การปรับปรุงประสิทธิภาพของวินโดว์เอ็นทีเซิร์ฟเวอร์

PERFORMANCE TUNING ON WINDOWS NT SERVER

โดย

นางสาวกนกพร ภาวศุทธิกุล

นางสาวนพวรรณ สุรพานิช

วัน เดือน ปี 14 ก.ค. 2541

เลขทะเบียน 038953

เลขเรียกหนังสือ โ ม019น ก198ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

038953

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับปรุงประสิทธิภาพของวินโดว์เอ็นทีเซิร์ฟเวอร์

PERFORMANCE TUNING ON WINDOWS NT SERVER

โดย

นางสาวกนกพร ภาวศุทธิกุล 37014001

นางสาวนพวรรณ สุรพานิช 37014184

อาจารย์ที่ปรึกษา

ดร. วรวัฒน์ ลิ้มโภคา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การปรับปรุงประสิทธิภาพของวินโดว์เอ็นทีเซิร์ฟเวอร์

ผู้จัดทำ

1. นางสาวกนกพร ภาวศุทธิกุล
2. นางสาวนพวรรณ สุรพานิช

..... อาจารย์ที่ปรึกษา

(ดร. วรวัฒน์ ลิ้มโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับปรุงประสิทธิภาพของวินโดวส์เอ็นทีเซิร์ฟเวอร์

นางสาวกนกพร ภาวศุทธิกุล
นางสาวนพวรรณ สุรพานิช

อาจารย์ วรวัฒน์ ลิ้มโภคา
ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์นี้จัดทำขึ้นเพื่อวัดประสิทธิภาพขององค์ประกอบต่าง ๆ บนระบบวินโดวส์ เอ็นที เซิร์ฟเวอร์ (Windows NT Server) เพื่อให้องค์ประกอบต่าง ๆ ของวินโดวส์ เอ็นที เซิร์ฟเวอร์สามารถทำงานได้อย่างมีประสิทธิภาพสูงที่สุด

โดยได้ทำการตรวจจับคอขวด (bottleneck) ขององค์ประกอบต่าง ๆ บนระบบ เอ็นที เซิร์ฟเวอร์ โดยใช้แอปพลิเคชั่นโปรแกรม (application program) บนวินโดวส์ เอ็นที เซิร์ฟเวอร์ ที่มีชื่อว่า Performance Monitor ทำการตรวจจับคอขวดขององค์ประกอบต่าง ๆ บนระบบวินโดวส์ เอ็นที เซิร์ฟเวอร์ ซึ่งองค์ประกอบต่าง ๆ ได้แก่ โพรเซสเซอร์ (processor) , หน่วยความจำ (memory) , ดิสก์ (disk) , แคช (cache) และระบบเครือข่าย (network)

เมื่อพบคอขวดที่องค์ประกอบใดแล้วจึงทำการปรับปรุง (tune) องค์ประกอบนั้นให้มีประสิทธิภาพที่ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Performance Tuning on Windows NT Server

Kanokporn Pawasuttikul

Noppawan Surapanich

Dr. Worawat Limpoka

1997

Abstract

This thesis is implemented for measurement performance of many objects on Windows NT Server. With the intension of optimizing parts of the system to have the best performance.

We used "Performance Monitor" tool to detect bottlenecks. Performance Monitor is an excellent tool for optimizing computer performance on Windows NT.. When we used it to detect bottleneck on the server, such as, Processor Bottleneck, Memory Bottleneck, Disk Bottleneck, Cache Bottleneck and Network Bottleneck.

If we find bottleneck on any object, then we will tune that object to optimize it.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ..... II

สารบัญ..... III

สารบัญตาราง..... VI

สารบัญภาพ..... VII

บทที่ 1 บทนำ..... 1

1.1 ความสำคัญและที่มา..... 1

1.2 ขอบเขตของงานวิจัย..... 2

1.3 ประโยชน์ที่คาดว่าจะได้รับ..... 2

1.4 วิธีการดำเนินงาน..... 3

บทที่ 2 การตรวจจับปัญหาคอขวดที่เกิดจากโพรเซสเซอร์ (Processor Bottlenecks) 4

2.1 คอขวด คือเป้าที่เคลื่อนที่ได้..... 4

2.2 การวิเคราะห์ประสิทธิภาพของโพรเซสเซอร์..... 5

2.3 ทำไมโปรแกรม Performance Monitor จึงมีการใช้งานอุปกรณ์ต่าง ๆในระบบน้อยมาก..... 6

2.4 การจัดตารางการใช้งานของโพรเซสเซอร์ (Processor Scheduling) บน Windows NT..... 7

2.5 สิ่งที่ซ่อนอยู่ในความยาวของคิวแบบสลับฟันปลา (Sawtooth Queue Length)..... 7

2.6 ยูสเซอร์โหมด (user mode) และ พรีวิเลจโหมด (privileged mode)..... 7

2.7 ลักษณะของหลายกระบวนการ (Multiple Processes)และหลายเธร็ด (Multiple Threads)... 7

2.8 ปัญหาคอขวดที่อุปกรณ์ที่มีผลประโยชน์ต่ำ (Low utilization)..... 9

2.9 ระบบที่มีโพรเซสเซอร์หลายตัว..... 10

2.10 การแก้ไขปัญหาคอขวดที่เกิดจากโพรเซสเซอร์..... 11

2.10.1 การแก้ปัญหาโดยใช้ฮาร์ดแวร์ (hardware)..... 11

2.10.2 การแก้ปัญหาโดยใช้ซอฟต์แวร์ (software) 12

บทที่ 3 การตรวจจับปัญหาคอขวดที่เกิดจากดิสก์ (Disk Bottlenecks)..... 13

3.1 ดิสก์ที่มีธรูพุท (throughput) สูง..... 16

3.2 การแก้ไขปัญหาคอขวดที่เกิดจากดิสก์..... 16

3.2.1 การแก้ปัญหาโดยใช้ฮาร์ดแวร์ 16

3.2.2 การแก้ปัญหาโดยใช้ซอฟต์แวร์..... 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การตรวจจับปัญหาคอขวดที่เกิดจากหน่วยความจำ (Memory Bottlenecks)..... 19
4.1 การทำงานของระบบหน่วยความจำเวอร์ชวลของ Windows NT 19
4.2 ตัวอย่างของกิจกรรมของหน่วยความจำและการเพจจิ้ง..... 20
4.3 การเพจจิ้งที่มีหลายกระบวนการ..... 28
4.4 การดูค่าของนอนเพจพูล (Nonpaged Pool)..... 31
4.5 การแก้ไขปัญหาคอขวดที่เกิดจากหน่วยความจำ..... 32
4.5.1 การแก้ปัญหาโดยใช้ฮาร์ดแวร์ 32
4.5.2 การแก้ปัญหาโดยใช้ซอฟต์แวร์..... 32
บทที่ 5 การตรวจจับปัญหาคอขวดที่เกิดจากแคช (Cache Bottlenecks)..... 34
5.1 การทำงานของแคช..... 34
5.2 การทดลองเบื้องต้นเกี่ยวกับแคช 35
5.3 การอ่านและการเขียนแบบซีเควนเชียล..... 39
5.4 การอ่านและการเขียนแบบสุ่ม..... 44
5.5 การแมปไฟล์ลงในหน่วยความจำ..... 48
บทที่ 6 การตรวจจับปัญหาคอขวดที่เกิดจากเครือข่าย (Network Bottlenecks)..... 52
6.1 ธรรพุทของเครือข่าย 52
6.2 รูปแบบง่าย ๆ ของจุดคอขวดของเครือข่าย 55
6.3 หลักทั่ว ๆ ไปของการตรวจจับจุดคอขวดของเครือข่าย..... 55
6.4 การดูประสิทธิภาพของ TCP/IP 58
บทที่ 7 การคำนวณ การสร้าง การออกแบบ..... 63
7.1 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากโพรเซสเซอร์..... 63
7.1.1 %Processor Time..... 63
7.1.2 %Total Processor Time..... 63
7.1.3 Processor Queue Length..... 63
7.2 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากดิสก์..... 63
7.2.1%Disk Time..... 63
7.2.2Avg. Disk Bytes/Transfer..... 63
7.2.3Avg. Disk Sec/Transfer 63
7.2.4Disk Bytes/sec 64
7.3 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากหน่วยความจำ 64
7.3.1Pages/sec 64
7.3.2 Available Bytes 64
7.3.3 Commit Limit 64
7.3.4 Committed Bytes..... 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.5Pool Nonpaged Bytes 64
7.3.6 Logical Disk Object 64
7.4 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากแคช 65
7.4.1 Copy Read Hits% 65
7.4.2 Data Flush Pages/sec และ Data Flushes/sec 65
7.4.3 Data Map Hits% และ Data Maps/sec 65
7.4.4 รวมเคาน์เตอร์ของ Memory Object กับ Cache Object เข้าด้วยกัน
แล้วเทียบกับ Logical Disk Object 66
7.5 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากเครือข่าย 66
7.5.1 ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface)..... 66
7.5.2 ออบเจ็กต์ : ไอพี (Object : IP)..... 67
7.5.3 ออบเจ็กต์ : ทีซีพี (Object : TCP)..... 67
บทที่ 8 ผลการทดลอง..... 68
บทที่ 9 สรุปและวิจารณ์..... 146
9.1 วิเคราะห์โครงงานเทียบกับทฤษฎีที่ได้ศึกษามา 146
9.2 แนวทางในการพัฒนาต่อไป 146
9.3 สรุปผลโครงงาน 146
ภาคผนวก ก การใช้โปรแกรม Performance Monitor..... 147
ภาคผนวก ข WindowsNT Performance Counters..... 155
กิตติกรรมประกาศ..... 177
บรรณานุกรม..... 178

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
5-1	เวลาของไฟล์ I/O แบบใช้แคชและไม่ใช้แคช (หน่วยเป็นมิลลิวินาทีต่อเรคคอร์ด)	36
5-2	เวลาของ memory mapped เทียบกับไฟล์ I/O มีหน่วยเป็นมิลลิวินาทีต่อเรคคอร์ด	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ หน้า

2-1 ชาร์ตแสดงการใช้โพรเซสเซอร์และดิสก์..... 5
2-2 รายงานของการใช้โพรเซสเซอร์และดิสก์..... 5
2-3 การไม่มีโอเวอร์เฮด (overhead) ของ Windows NT Performance Monitor..... 6
2-4 การเปรียบเทียบความยาวของคิวของโพรเซสเซอร์..... 8
2-5 แสดงกระบวนการที่กำลังใช้โพรเซสเซอร์..... 8
2-6 เวลาการใช้โพรเซสเซอร์และความยาวของคิวซึ่งมีหลายเธร็ด 9
2-7 เวลาในการตอบสนองงานที่ใช้โพรเซสเซอร์แบบสุ่ม..... 10
3-1 ลักษณะการใช้โพรเซสเซอร์และดิสก์..... 13
3-2 ข้อมูลของโพรเซสเซอร์และระบบเมื่อมีการอ่านเรคคอร์ด (record) ขนาดเล็กจากดิสก์..... 13
3-3 ข้อมูลของดิสก์ในขณะอ่านเรคคอร์ดขนาดเล็กโดยทำการล็อคจากคนละที่..... 14
3-4 ความแตกต่างระหว่างอัตราการเคลื่อนย้าย (transfer) ข้อมูลของดิสก์..... 14
3-5 ข้อมูลระบบของ 5 กระบวนการที่กำลังอ่านเรคคอร์ดขนาดเล็กพร้อมกัน..... 15
3-6 ข้อมูลดิสก์ของ 5 กระบวนการที่กำลังอ่านเรคคอร์ดขนาดเล็กพร้อมกัน..... 15
3-7 การใช้ Registry Editor ในการเปลี่ยนค่าของระบบวินโดวส์เอ็นที..... 18
4-1 การทำ page fault บน Windows NT..... 20
4-2 กิจกรรมของโพรเซสเซอร์ในขณะเริ่มต้นคล็อก..... 21
4-3 การเกิด page fault ในขณะเริ่มต้นคล็อก..... 21
4-4 เพจที่ถูกนำเข้ามาขณะเริ่มต้นคล็อก..... 22
4-5 การอ่านเพจในขณะเริ่มต้นคล็อก..... 22
4-6 page fault โดยกระบวนการขณะเริ่มต้นคล็อก..... 23
4-7 รายงานสถิติของหน่วยความจำและดิสก์ในขณะเริ่มต้นคล็อก..... 24
4-8 ขนาดของ working set เพิ่มขึ้นเมื่อเริ่มต้นคล็อก..... 25
4-9 Available bytes ลดลงเมื่อเริ่มต้นคล็อก..... 26
4-10working set ลดลงโดยยูทิลิตี้ clearmem..... 27
4-11เพจที่มีการนำเข้ามาและส่งออกไปในระหว่างการเริ่มต้นคล็อก..... 28
4-12ปัญหาคอขวดที่มีผลกระทบต่อกันในระหว่างการเริ่มต้นคล็อก..... 29
4-13การใช้โพรเซสเซอร์และการจราจรของเพจภายใต้แรงกดดันของหน่วยความจำที่เพิ่มขึ้น..... 30
4-14การใช้ระบบขณะที่มีอัตราการเพจจิ้งมาก..... 30
4-15working set ของ Response Probe เมื่อแรงกดดันของหน่วยความจำเพิ่มขึ้น..... 31
5-1 การอ้างถึงแคชเมื่อไม่พบเพจที่ต้องการซึ่งจัดการโดยตัวจัดการหน่วยความจำ..... 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5-2 ประสิทธิภาพของดิสก์ของเครื่องตัวอย่าง 386SX/25..... 35
5-3 ลักษณะของระบบในขณะอ่านไฟล์ขนาดใหญ่โดยใช้แคช..... 37
5-4 สถิติของแคชในขณะอ่านไฟล์ขนาดใหญ่..... 37
5-5 Copy Read Hits% และ Copy Reads/Sec ในระหว่างการอ่านไฟล์ขนาดใหญ่..... 39
5-6 กิจกรรมของแคชและดิสก์ในการอ่านและเขียนไฟล์ขนาดใหญ่แบบซีเควนเชียล..... 40
5-7 สถิติของแคชในการอ่าน/เขียนไฟล์ขนาดใหญ่แบบซีเควนเชียล..... 40
5-8 สถิติของแคชและระบบสำหรับการอ่านและเขียนไฟล์ขนาดใหญ่แบบซีเควนเชียล..... 41
5-9 การเขียนแบบเลซี่ (lazy writing) โดยกระบวนการของระบบ..... 42
5-10 เธร็ดของกระบวนการของระบบที่ช่วยแบ่งงานของเลซี่ไรท์เตอร์..... 42
5-11 การตอบสนองของดิสก์ต่อกิจกรรมของแคชในระหว่างการอ่าน/เขียนแบบซีเควนเชียล..... 43
5-12 ตัวจัดการหน่วยความจำและตัวจัดการแคชทำงานประสานงานกัน..... 44
5-13 การอ่านและการเขียนแบบสุ่มที่ยุ่งมาก..... 45
5-14 แคชนำเพจที่ working set อื่นต้องการมาเป็นจำนวนมาก..... 45
5-15 สถิติของแคชสำหรับการอ่าน/เขียนโดยการแจกแจงแบบปกติ..... 46
5-16 สถิติของแคชสำหรับการอ่าน/เขียนโดยการแจกแจงแบบปกติ..... 46
5-17 ตัวจัดการหน่วยความจำและแคชโดยการแจกแจงแบบปกติ..... 47
5-18 สถิติของดิสก์ในการอ่าน/เขียนโดยการแจกแจงแบบปกติ..... 48
5-19 กระบวนการที่มีการแข่งขันกันในการอ่านแบบใช้แมปไฟล์และระบบไฟล์..... 49
5-20 กระบวนการที่มีการแข่งขันกันมากขึ้นในการอ่านแบบใช้แมปไฟล์และระบบไฟล์..... 50
5-21 เวลาการตอบสนองของกระบวนการที่ใช้การแมปไฟล์และบัฟเฟอร์..... 51
6-1 มุมมองทางด้านไคลเอ็นท์ของการทดสอบธรูพุทของเครือข่าย..... 52
6-2 มุมมองทางด้านเซิร์ฟเวอร์ของการทดสอบธรูพุทของเครือข่าย..... 53
6-3 มุมมองทางด้านไคลเอ็นท์ของการอ่านเรคคอร์ดขนาด 2048 ไบต์แบบไม่ใช้บัฟเฟอร์..... 54
6-4 มุมมองทางด้านไคลเอ็นท์ของการอ่านเรคคอร์ดขนาด 2048 ไบต์แบบไม่ใช้บัฟเฟอร์..... 54
6-5 มุมมองของไคลเอ็นท์ของการอ่านทีละ 14 เพจ..... 56
6-6 มุมมองของเซิร์ฟเวอร์ของการอ่านทีละ 14 เพจ..... 57
6-7 ชาร์ตแสดงธรูพุทของ TCP/IP..... 58
6-8 Network Interface ของ TCP/IP ในการอ่าน 2048 ไบต์ (ด้านเซิร์ฟเวอร์)..... 59
6-9 เคาน์เตอร์ของ TCP/IP ของเซิร์ฟเวอร์ในการอ่าน 2048 ไบต์..... 60
6-10 สถิติของ NBT ของเซิร์ฟเวอร์ในการอ่าน 2048 ไบต์..... 61
6-11 สถิติของ Network Interface ของเซิร์ฟเวอร์ในการเขียนขนาด 1024 ไบต์ไปยังเซิร์ฟเวอร์..... 62
6-12 สถิติของ UDP/IP ของเซิร์ฟเวอร์ในการเขียน 1024 ไบต์ไปยังเซิร์ฟเวอร์..... 62
8-1 แอปพลิเคชั่นที่เกี่ยวกับใบสั่งซื้อสินค้าทั้ง 2 รูปแบบ..... 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8-2 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 69
8-3 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 69
8-4 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 70
8-5 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 70
8-6 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 71
8-7 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน 73
8-8 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน 74
8-9 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 74
8-10 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 75
8-11 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 75
8-12 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน... 78
8-13 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน... 78
8-14 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 79
8-15 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 79
8-16 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 80
8-17 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที... 82
8-18 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที... 83
8-19 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที..... 83
8-20 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที..... 84
8-21 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที..... 84
8-22 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน
1วินาที..... 87
8-23 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน
1 วินาที..... 87
8-24 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน
1 วินาที..... 88
8-25 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน
1 วินาที..... 88
8-26 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน
1 วินาที..... 89
8-27 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 92
8-28 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 92
8-29 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8-30 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 93
8-31 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน..... 94
8-32 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 96
8-33 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 97
8-34 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 97
8-35 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 98
8-36 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 98
8-37 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน..... 99
8-38 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออร์เดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน
1 วินาที..... 101
8-39 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออร์เดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน
1 วินาที..... 102
8-40 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน .....
1 วินาที..... 102
8-41 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน .....
1 วินาที..... 103
8-42 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที..... 103
8-43 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 106
8-44 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 106
8-45 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 107
8-46 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 107
8-47 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 108
8-48 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 108
8-49 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน..... 112
8-50 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน..... 112
8-51 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน..... 113
8-52 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน..... 113
8-53 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน..... 114
8-54 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน..... 116
8-55 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟส สำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน..... 116
8-56 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 118
8-57 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 118
8-58 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 119

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8-59 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 119
8-60 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 121
8-61 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน 122
8-62 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 123
8-63 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 124
8-64 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 124
8-65 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 125
8-66 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 127
8-67 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 127
8-68 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดพร้อมกัน... 129
8-69 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 129
8-70 แสดงข้อมูลของไดร์ฟดีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 130
8-71 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 130
8-72 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดพร้อมกัน 132
8-73 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดพร้อมกัน..... 133
8-74 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์ โดยล็อกอินพร้อมกัน..... 134
8-75 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยล็อกอินพร้อมกัน..... 135
8-76 แสดงข้อมูลของไดร์ฟดีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยล็อกอินพร้อมกัน..... 135
8-77 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์ โดยล็อกอินพร้อมกัน..... 136
8-78 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยล็อกอินพร้อมกัน..... 138
8-79 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำเคาน์เตอร์ออเดอร์ โดยล็อกอินพร้อมกัน 138
8-80 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 140
8-81 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 140
8-82 แสดงข้อมูลของไดร์ฟดีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 141
8-83 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที 141
8-84 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 143
8-85 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดห่างกัน 1 วินาที..... 144

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันนี้การใช้ทรัพยากร (resource) และอุปกรณ์ต่าง ๆ สำหรับคอมพิวเตอร์ (computer) มีความต้องการที่จะให้เกิดประโยชน์สูงที่สุด เพื่อให้ประหยัดทั้งเงินและเวลาในการดำเนินงานและยังช่วยให้การใช้งานของอุปกรณ์ต่าง ๆ มีประสิทธิภาพมากที่สุด

ต่างจากในอดีตที่มนุษย์มักจะทำงานในลักษณะที่เป็นตัวคนเดียว ใช้อุปกรณ์ต่าง ๆ เพียงคนเดียว ไม่ได้มีการใช้อุปกรณ์ร่วม (share device) กับบุคคลอื่น อีกทั้งยังไม่มีการติดต่อสื่อสารส่งข้อมูลถึงคนอื่นอีกด้วย ทำให้มีการใช้ทรัพยากรแบบตัวใครตัวมันไม่มีการแบ่งทรัพยากรต่าง ๆ ที่มีอยู่ใช้กันอย่างคุ้มค่า ซึ่งหากมีผู้ต้องการใช้จำนวนมาก ๆ แล้วอุปกรณ์ต่าง ๆ ก็จะไม่เพียงพอกับความต้องการใช้งาน

ซึ่งในปัจจุบันนี้คอมพิวเตอร์ส่วนใหญ่มักจะทำการเชื่อมโยงถึงกันเป็นระบบแบบ ไคลเอ็นท์-เซิร์ฟเวอร์ (client-server) ซึ่งไม่ได้มีการใช้งานเครื่องเดียวโดด ๆ อย่างแต่ก่อนอีกต่อไป ซึ่งในระบบเชื่อมโยงคอมพิวเตอร์แบบไคลเอ็นท์-เซิร์ฟเวอร์จะมีการแบ่งทรัพยากรกันใช้ และทำให้มีการใช้ทรัพยากรกันอย่างประหยัดที่สุดอีกด้วย

โดยในที่นี้จะขอกล่าวถึงเฉพาะเซิร์ฟเวอร์เพราะเกี่ยวข้องกับปริญญานิพนธ์นี้ที่ได้ทำการวัดองค์ประกอบต่าง ๆ ทางด้านเซิร์ฟเวอร์ เซิร์ฟเวอร์ คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่ให้บริการแก่คอมพิวเตอร์อื่น ๆ ซึ่งเป็นลูกข่าย (client) โดยทั่วไปหน้าที่ของเซิร์ฟเวอร์มี 3 ประเภท คือ ให้บริการในการจัดเก็บข้อมูล เรียกว่า "File Server" , ให้บริการและควบคุมเครื่องพิมพ์ เรียกว่า " Print Server" และให้บริการควบคุมอุปกรณ์สื่อสาร เรียกว่า "Communication Server"

เมื่อมีผู้ต้องการใช้ทรัพยากรหรือบริการพร้อมกันมาก ๆ ในเวลาเดียวกัน องค์ประกอบต่าง ๆ บนระบบเซิร์ฟเวอร์อาจจะไม่สามารถให้บริการการใช้ทรัพยากรได้ในทันทีที่ต้องการหรืออาจจะทำให้ประสิทธิภาพการทำงานขององค์ประกอบต่าง ๆ บนเซิร์ฟเวอร์ลดลงได้

ดังนั้นจึงมีความจำเป็นที่จะต้องทำการวัดประสิทธิภาพการทำงานขององค์ประกอบต่าง ๆ บนเซิร์ฟเวอร์ที่ใช้วินโดวส์ เอ็นที ซึ่งวินโดวส์ เอ็นที เป็นเทคโนโลยีใหม่สำหรับระบบปฏิบัติการที่ถูกพัฒนาขึ้นมาจากบริษัทไมโครซอฟท์ (Microsoft) ซึ่งสามารถใช้งานได้ทั้งซิงเกิลยูสเซอร์ (single-user) , และถูกออกแบบสำหรับระบบปฏิบัติการที่สามารถใช้ได้หลาย ๆ งานในเวลาเดียวกัน เพื่อที่จะวิ่งบนเครื่องพีซี (PC) และเวิร์กสเตชัน (Workstation) ที่ต่าง ๆ กัน อีกทั้งยังมีระบบรักษาความปลอดภัยให้อีกด้วย (security mode)

ซึ่งในการวัดประสิทธิภาพขององค์ประกอบต่าง ๆ บนเซิร์ฟเวอร์นั้น เราจะใช้วิธีการตรวจจับคอขวดขององค์ประกอบต่าง ๆ บนเซิร์ฟเวอร์ ได้แก่ การตรวจจับคอขวดของโพรเซสเซอร์ (Detecting Processor Bottleneck), การตรวจจับคอขวดของหน่วยความจำ (Detecting Memory Bottleneck), การตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จับคอขวดของดิสก์ (Detecting Disk Bottleneck), การตรวจจับคอขวดของแคช (Detecting Cache Bottleneck) และการตรวจจับคอขวดของระบบเครือข่าย (Detecting Network Bottleneck)

คอขวดขององค์ประกอบต่าง ๆ ก็คือจุดที่องค์ประกอบนั้น ถูกใช้งานอย่างหนักจนทำให้ประสิทธิภาพการทำงานขององค์ประกอบนั้นช้าลงหรือทำงานได้น้อยลงแต่ใช้เวลาเพิ่มขึ้น ซึ่งเราสามารถทราบได้โดยดูที่เคาน์เตอร์ (counter) ขององ๕ประกอบนั้น ๆ ซึ่งเคาน์เตอร์ก็คือตัวบอกคุณลักษณะการทำงานในรูปแบบต่าง ๆ ขององค์ประกอบนั้น ๆ เช่น โพรเซสเซอร์ มีเคาน์เตอร์ต่าง ๆ มากมาย ในที่นี้จะขอกล่าวเพียง 2 เคาน์เตอร์ซึ่งก็คือ : % Processor Time คือ เปอร์เซ็นต์ของเวลาที่โพรเซสเซอร์ใช้ในการทำงาน และ Interrupts/sec คือ จำนวนของการอินเทอร์รัปต์ (Interrupt) โพรเซสเซอร์ต่อวินาที

โปรแกรมที่ใช้ในการตรวจจับก็คือ โปรแกรม "Performance Monitor" ซึ่งมีอยู่ในวินโดวส์ เอ็นที ซึ่งโปรแกรมนี้จะมีออบเจ็กต์ (object) ต่าง ๆ ให้เลือกเพื่อที่จะทำการตรวจจับ เช่น มี โพรเซสเซอร์ , หน่วยความจำ , ดิสก์ , แคช และอื่น ๆ อีกมากมายที่มีอยู่ในเครื่องที่จะทำการตรวจจับนั้น และในออบเจ็กต์แต่ละตัวจะมีเคาน์เตอร์เอาไว้ให้ดูการทำงานของออบเจ็กต์ ซึ่งควรจะดูเคาน์เตอร์ตัวไหนบ้างสำหรับออบเจ็กต์แต่ละตัว จะขอกล่าวในส่วนของการออกแบบ

และจากที่ได้กล่าวมาข้างต้นนั้น ผู้อ่านคงจะเห็นถึงความจำเป็นที่จะต้องทำการตรวจจับคอขวดขององค์ประกอบต่าง ๆ บนระบบวินโดวส์ เอ็นที เซิร์ฟเวอร์ และเมื่อทำการตรวจจับคอขวดได้แล้วก็จะทำการปรับปรุงองค์ประกอบนั้นให้มีการทำงานที่มีประสิทธิภาพมากขึ้นซึ่งทำอย่างไรนั้นจะกล่าวในบทต่อไป

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 ทำการศึกษาวิธีการใช้โปรแกรม Performance Monitor และสามารถใช้โปรแกรมในการตรวจจับคอขวดได้

1.2.2 สามารถทราบว่าถ้าทำการตรวจจับคอขวดที่องค์ประกอบต่าง ๆ แล้วองค์ประกอบนั้น ๆ ควรจะใช้เคาน์เตอร์ใดบ้างเป็นเกณฑ์ในการตรวจจับ

1.2.3 สามารถวิเคราะห์ข้อมูลที่ได้จากการบันทึกการตรวจจับว่าเกิดคอขวดที่องค์ประกอบใดบ้าง

1.2.4 สามารถทำการปรับปรุงองค์ประกอบนั้นให้มีประสิทธิภาพในการทำงานที่ดีขึ้นกว่าเดิมเมื่อเกิดคอขวด

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้ได้ทำการวัดองค์ประกอบต่าง ๆ ต่อไปนี้ คือ โพรเซสเซอร์, หน่วยความจำ, ดิสก์, แคช, และระบบเครือข่ายบนวินโดวส์ เอ็นที เซิร์ฟเวอร์ และบันทึกผลการวัดต่าง ๆ เอาไว้แล้วนำมาวิเคราะห์โดยใช้ทฤษฎีและหลักการต่าง ๆ ซึ่งทำการวิเคราะห์จากตัวเลขและกราฟของข้อมูลแล้วทำการสรุปว่าเกิดคอขวดที่องค์ประกอบใดบ้าง ซึ่งองค์ประกอบแต่ละตัวก็จะมีเกณฑ์ในการวัดดังที่ได้ออกแบบเอาไว้แล้ว

นอกจากนั้นยังมีการวัดที่เครื่องแอปพลิเคชั่นเซิร์ฟเวอร์ (application server) ว่าเกิดคอขวดจากการวิ่งแอปพลิเคชั่นใดบนเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพบคอขวดแล้วก็ให้ทำการปรับปรุงแก้ไข หรืออัพเกรด (upgrade) อุปกรณ์หรือองค์ประกอบนั้น ๆ แล้วทำการตรวจจับใหม่ว่าได้ผลดีขึ้นหรือไม่อย่างไร

1.4 วิธีการดำเนินงาน

การวิจัยในโครงงานนี้เริ่มต้นด้วยการศึกษาวิธีการใช้โปรแกรมที่ช่วยในการตรวจจับคอขวดซึ่งก็คือโปรแกรม Performance Monitor นั่นเอง โดยที่โปรแกรมตัวนี้เป็นโปรแกรมชั้นดีที่เขียนขึ้นมาเพื่อศึกษาถึงการทำงานภายในขององค์ประกอบต่าง ๆ จริง ๆ ซึ่งการเขียนโปรแกรมตัวนี้จะต้องเกี่ยวข้องกับระบบปฏิบัติการและเป็นผู้ที่ความรู้อย่างดีเยี่ยมจึงจะสามารถเขียนได้ ดังนั้นงานวิจัยนี้จึงไม่เกี่ยวข้องกับการเขียนโปรแกรมเพื่อตรวจจับคอขวด

โดยจะศึกษาโปรแกรมตัวนี้เกี่ยวกับการเก็บบันทึกข้อมูลข้อมูล ,การแสดงข้อมูลเป็นรูปกราฟต่าง ๆ และการแสดงข้อมูลเป็นตัวเลข ทั้งที่แสดงข้อมูล ณ เวลาปัจจุบันหรือแสดงข้อมูลจากแฟ้มข้อมูลที่บันทึกไว้เป็นล็อกไฟล์ (log file)

หลังจากการทำการศึกษาถึงวิธีการใช้โปรแกรม Performance Monitor แล้วก็ทำการศึกษาองค์ประกอบแต่ละตัวว่าถ้าจะตรวจจับคอขวด ควรจะพิจารณาที่เคาน์เตอร์ตัวใดบ้างและควรจะทราบค่าประมาณของเคาน์เตอร์แต่ละตัวที่ทำให้เกิดคอขวด และเคาน์เตอร์ตัวใดที่จะต้องดูควบคู่กับเคาน์เตอร์ใดบ้าง

เมื่อทราบถึงวิธีการใช้โปรแกรมตรวจจับคอขวดและรู้ทฤษฎีขององค์ประกอบแต่ละตัวแล้วก็ทำการบันทึกข้อมูลต่าง ๆ ลงในล็อกไฟล์ โดยข้อมูลต่าง ๆ ที่ได้ทำการบันทึกนั้นได้จากการทำงานจริงบนเครื่องเซิร์ฟเวอร์ของบริษัทสหอินโฟเทคโนโลยี จำกัด

ทำการวิเคราะห์ข้อมูลที่ได้ทำการบันทึกไว้ทั้งจากข้อมูลที่เป็นกราฟและจากข้อมูลที่เป็นตัวเลขแล้วสรุปผลว่าเกิดคอขวดที่องค์ประกอบใด แล้วให้ทำการปรับปรุงตัวเครื่องเซิร์ฟเวอร์ที่องค์ประกอบนั้นใหม่แล้วทำการวัดที่องค์ประกอบนั้น แล้ววิเคราะห์อีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การตรวจจับปัญหาคอขวดที่เกิดจากโพรเซสเซอร์

คอขวด (bottleneck) มีนิยามดังนี้

$$\frac{\text{util(Processor)}}{\text{util(Disk)}} = \frac{\text{demand(Processor)}}{\text{demand(Disk)}}$$

demand (Processor) คือ เวลาของโพรเซสเซอร์ที่ใช้ไปโดยการทำงานอย่างใดอย่างหนึ่ง (interaction)

demand (Disk) คือ เวลาของ ดิสก์ ที่ใช้ไปโดยการทำงานอย่างใดอย่างหนึ่ง

util (device) คือ ความสามารถในการใช้ประโยชน์ของอุปกรณ์ (ไม่ว่าจะเป็นดิสก์หรือโพรเซสเซอร์ในกรณีนี้)

util (device) จะมีค่าเสมอเป็นตัวเลขทั่ว ๆ ไปคือ 0 ถึง 1 ถ้าเป็นเปอร์เซ็นต์ 0 ถึง 100%

$$\text{max throughput(device)} = 1/\text{demand(device)}$$

ให้ maximum throughput = interaction/วินาที

ตัวอย่าง สมมติว่าการทำงานอย่างหนึ่ง (interaction) ต้องการ 0.3 วินาทีของเวลาโพรเซสเซอร์ และ 0.5 วินาทีของเวลาดิสก์ เพราะฉะนั้นโพรเซสเซอร์ จะมี 3.3 interaction/วินาที ขณะที่ ดิสก์ มี 2 interaction/วินาที ดังนั้นทั้งระบบจะมี 2 interaction/วินาที เท่านั้น ซึ่งจุดนี้ดิสก์จะอิ่มตัว (utilization = 1) และสามารถหา utilization ของโพรเซสเซอร์ ณ จุดนี้ จากสูตรได้คือ 0.3/0.5 = 0.6 หรือ 60%

$$\text{throughput} = \text{util(device)} / \text{demand(device)}$$

Processor Queue Length_ คือ การวัดจำนวนของเธร็ด (thread) เพื่อรอที่จะเอ็กซีคิว (execute) คำสั่งของโปรแกรมเมื่อไม่มีซีพียูว่าง

อุปกรณ์ ที่มี demand มากที่สุดจะเป็นคอขวดของระบบ

มันเป็นการไม่ยากนักที่จะดูว่ามีคอขวดที่ 2 ซ่อนอยู่ภายใต้อันที่ 1 โดยปกติแล้วจะเป็นอุปกรณ์ที่มีผลประโยชน์ (utilization) ต่ำรองลงมา ที่เราใช้คำว่า "โดยปกติ" ในที่นี้ก็เพราะว่าถ้าคุณกำจัดที่เป็นคอขวดอันแรก สิ่งหนึ่งบนลักษณะภายนอกอาจจะเปลี่ยนแปลงขึ้นอยู่กับว่าคอขวดอันแรก ถูกกำจัดอย่างไร สิ่งสำคัญที่ควรจำไว้การกำจัดคอขวดอันแรก เพียงแค่ 1 คอขวด ไม่ได้ช่วยให้คอขวดหมดไป บางครั้งคุณอาจจะต้องกำจัดคอขวดอื่น ๆ อีกด้วย

กฎต่าง ๆ ในการตรวจจับจุดคอขวด

กฎข้อที่ 1: ขณะเวลาหนึ่ง ๆ ควรทำการเปลี่ยนเพียงอย่างเดียวเท่านั้น

กฎข้อที่ 2: จุดคอขวดจุดหนึ่งอาจจะทำให้เกิดคอขวดที่จุดอื่นๆได้

กฎข้อที่ 3: การเกิดคอขวดขึ้นอยู่กับการมองของคุณ

กฎข้อที่ 4: ควรแก้ปัญหาที่ใหญ่ที่สุดก่อน

กฎข้อที่ 5: Heisenberg กล่าวว่า " เมื่อหาจุดคอขวดได้แล้ว ทำการเปลี่ยนมัน"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎข้อที่ 6: สิ่งใด ๆ ที่ค้นพบจะทำให้เกิดปัญหาใหม่ ๆ ขึ้น

กฎข้อที่ 7: อัตราส่วนของเคาน์เตอร์ (counter) ต่าง ๆ จะแสดงปัญหาคอขวด

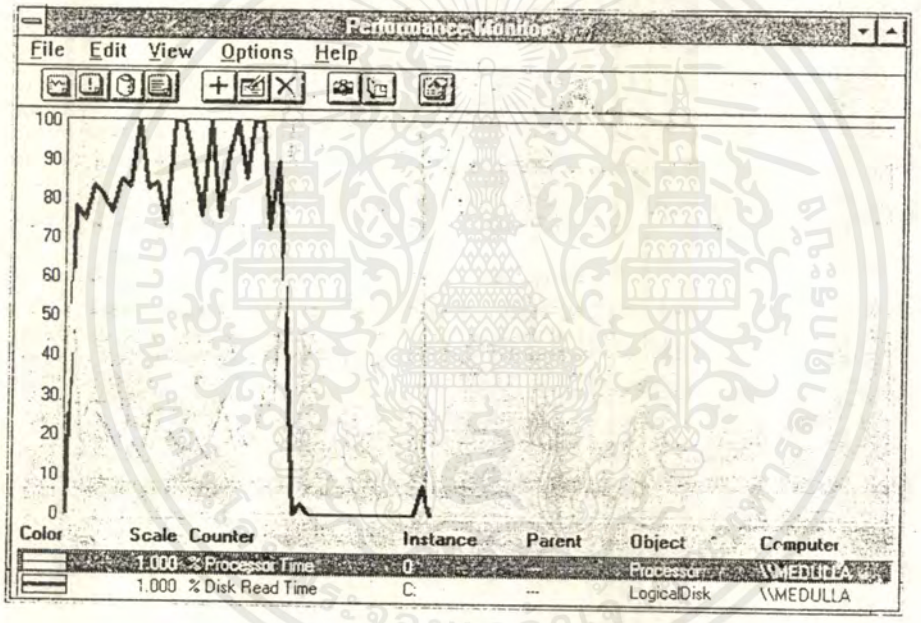
กฎข้อที่ 8: การใช้แบบสุ่มจะทำให้ความยาวของคิวเพิ่มขึ้นและการให้ผลประโยชน์ (utilization) ต่ำลง

กฎข้อที่ 9: ค่าของเคาน์เตอร์ต่าง ๆ ที่ใกล้เคียงกันสำหรับแต่ละออบเจ็กต์ (object)จะทำให้สามารถตรวจจับปัญหาคอขวดได้

กฎข้อที่ 10: ค่าเฉลี่ยต่าง ๆ จะแสดงค่าความจริงขณะที่มีร่องรอยของรายละเอียดต่าง ๆ ซ่อนอยู่

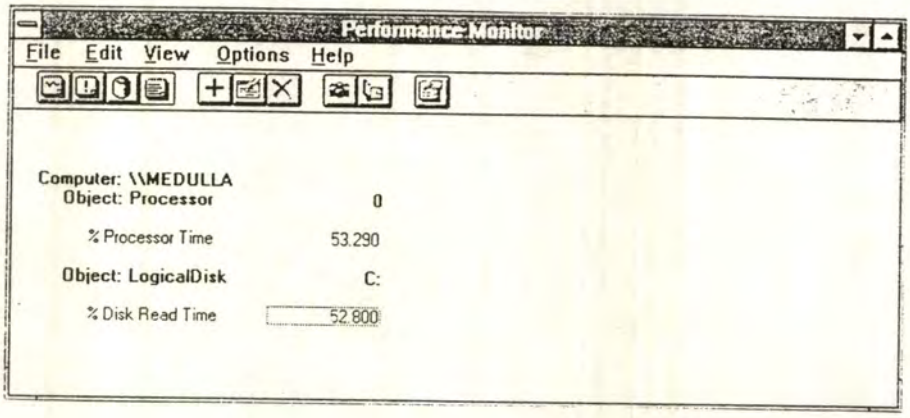
2.1 คอขวด คือเป้าที่เคลื่อนที่ได้

สิ่งหนึ่งที่ควรจะระลึกไว้เสมอว่าขณะดำเนินการตามขั้นตอนอยู่นั้น, จุดที่เป็นคอขวดอาจเคลื่อนที่ไปรอบ ๆ จากอุปกรณ์ชิ้นหนึ่งไปยังอุปกรณ์อีกชิ้นหนึ่ง, ในแต่ละวินาทีของการทำงานอาจจะมีคอขวดต่าง ๆ เกิดขึ้นถ้าตรวจสอบดู ถ้าคุณต้องการที่จะปรับปรุงระบบทั้งหมด คุณจะต้องตรวจสอบดูที่กราฟขนาดใหญ่และต้องอาศัยความชำนาญ



รูปที่ 2-1 ชาร์ตแสดงการใช้โพรเซสเซอร์และดิสก์

จากกราฟ จะมี 2 ช่วง คือช่วงแรกดิสก์เป็นคอขวด ส่วนช่วงที่ 2 ซีพียูเป็นคอขวด



รูปที่ 2-2 รายงานของการใช้โพรเซสเซอร์และดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดิสก์ให้ผลประโยชน์ ได้ 52.8% และซีพียู 53.3% แต่เราใช้เวลาทั้งหมด 44 วินาที หมายความว่า ใช้เวลา 23.23 วินาทีของเวลาดิสก์ (demand[disk]) และ 23.45 วินาที ของเวลาซีพียู (demand[processor]) ดิสก์มีธรูพุทมากที่สุดเป็น 1/23.23*3600=155 times/hour และโพรเซสเซอร์มีธรูพุทมากที่สุดเป็น 1/23.45*3600 = 153.5 times/hour

ดังนั้นโพรเซสเซอร์เป็นคอขวด แต่อันที่จริงแล้วทั้งซีพียูและดิสก์ มีค่าใกล้เคียงกันในอันที่จะกล่าวว่าอุปกรณ์ตัวไหนเป็นคอขวด

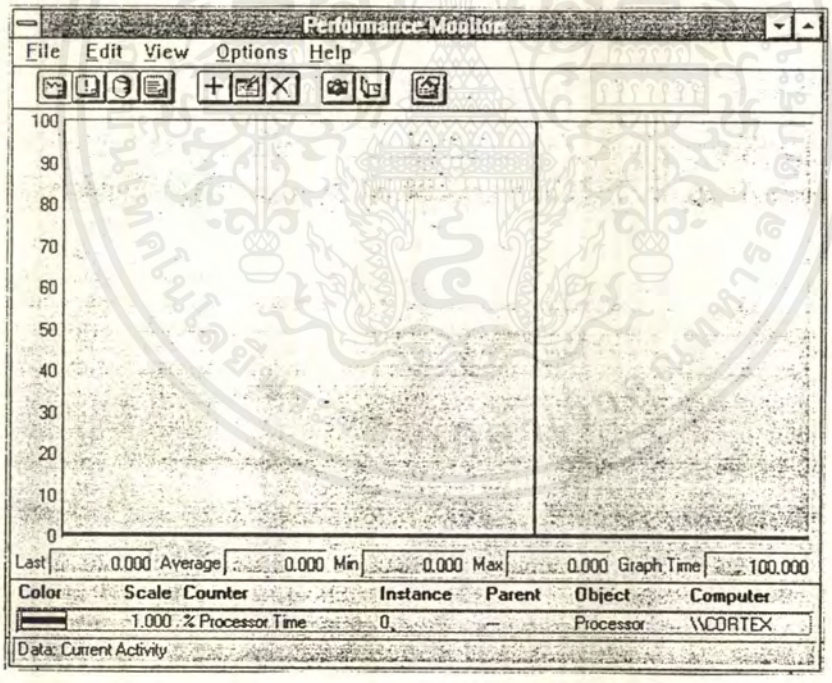
คุณอาจจะต้องการวัดระบบโดยใช้ Processor Queue Length นี่คือกุญแจสำคัญในการวัดความคับคั่งของโพรเซสเซอร์

2.2 การวิเคราะห์ประสิทธิภาพของโพรเซสเซอร์

เมื่อตรวจพบโพรเซสเซอร์ว่าเป็นคอขวด เราจะต้องหาต่อไปอีกว่ามันเป็นเพียงแค่กระบวนการ (process) เดียวหรือหลาย ๆ กระบวนการ ถ้ามี 1 กระบวนการ แล้วเป็นกระบวนการไหน แล้วมีแค่ 1 เธร็ดหรือหลายเธร็ด คำตอบจะบอกเราว่าเราจะต้องแก้ไขอะไรบ้าง

ดังนั้นขั้นต่อไปก็จะทำการตรวจสอบ System: Processor Queue Length เพื่อที่จะหาว่ามีจำนวนเธร็ดเท่าไร ที่กำลังแข่งขันช่วงชิงโพรเซสเซอร์กัน

2.3 ทำไมโปรแกรม Performance Monitor จึงมีการใช้งานอุปกรณ์ต่าง ๆ ในระบบน้อยมาก



รูปที่ 2-3 การไม่มีโอเวอร์เฮด (overhead) ของ Windows NT Performance Monitor

เป็นชาร์ตของโพรเซสเซอร์ที่ใช้ Windows NT Performance Monitor สังเกตดูจะเห็นว่า value bar ทั้งหมดเป็นศูนย์ เราจะคิดว่า Windows NT Performance Monitoring ไม่มีโอเวอร์เฮดจริง ๆ หรือ แต่ที่จริงแล้วมันมีโอเวอร์เฮดที่ต่ำมาก เพราะว่ามันใช้การปรับปรุงประสิทธิภาพ (tune) ด้วยตัวมันเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การจัดตารางการใช้งานของโพรเซสเซอร์ (Processor Scheduling) บน Windows NT

Windows NT กำหนดรายการของโพรเซสเซอร์ให้ใช้การประมวลผลแบบขนาน (symmetric multiprocessing) และการทำงานแบบพรีเอ็มทีฟ (preemtive multitasking)

อย่างแรกมาดูคอมพิวเตอร์ที่มีโพรเซสเซอร์ตัวเดียว ซึ่งกำหนดให้ใช้การทำงานแบบพรีเอ็มทีฟ หมายความว่าเธร็ดที่มีลำดับความสำคัญสูงที่สุด พร้อมที่จะถูกเอ็กซีคิวโดยโพรเซสเซอร์ , เธร็ดที่มีลำดับความสำคัญต่ำกว่าเธร็ดที่กำลังถูกเอ็กซีคิวจะต้องรอ , มีบางครั้งจะต้องมีการป้องกันการเกิดสตาร์เวชั่น (starvation) ถ้าลำดับความสำคัญมีค่าเท่ากันในการที่เข้าไปเอ็กซีคิวโพรเซสเซอร์ ดังนั้นเราจึงต้องมีการแบ่งโพรเซสเซอร์กันใช้ , ระบบจะมีการแบ่งใช้โพรเซสเซอร์เป็นช่วง ๆ สำหรับในเวลาหนึ่งให้เธร็ดหนึ่งเข้ามาใช้ และสลับจากเธร็ดที่เอ็กซีคิวไปแล้วช่วงหนึ่ง, ไปให้กับเธร็ดต่อไปตามลำดับ

2.5 สิ่งที่ซ่อนอยู่ในความยาวของคิวแบบสลับฟันปลา (Sawtooth Queue Length)

ขณะนี้เราทราบว่าการดูที่เวลาของโพรเซสเซอร์ อาจจะไม่ได้แสดงให้เห็นว่ามีคอขวด เหตุผลแรกก็คือ มีบางกระบวนการที่ใช้โพรเซสเซอร์เป็นเวลาเร็วมากจนกระทั่งไม่ได้มีการบันทึกว่ามีการใช้โพรเซสเซอร์เกิดขึ้น และอีกเหตุผลหนึ่งก็คือกระบวนการที่รออยู่ในคิว (queue) อาจจะไม่เคยได้ถูกเอ็กซีคิวเลยเพราะว่ามีกระบวนการที่มีลำดับความสำคัญสูงกว่าใช้โพรเซสเซอร์อยู่

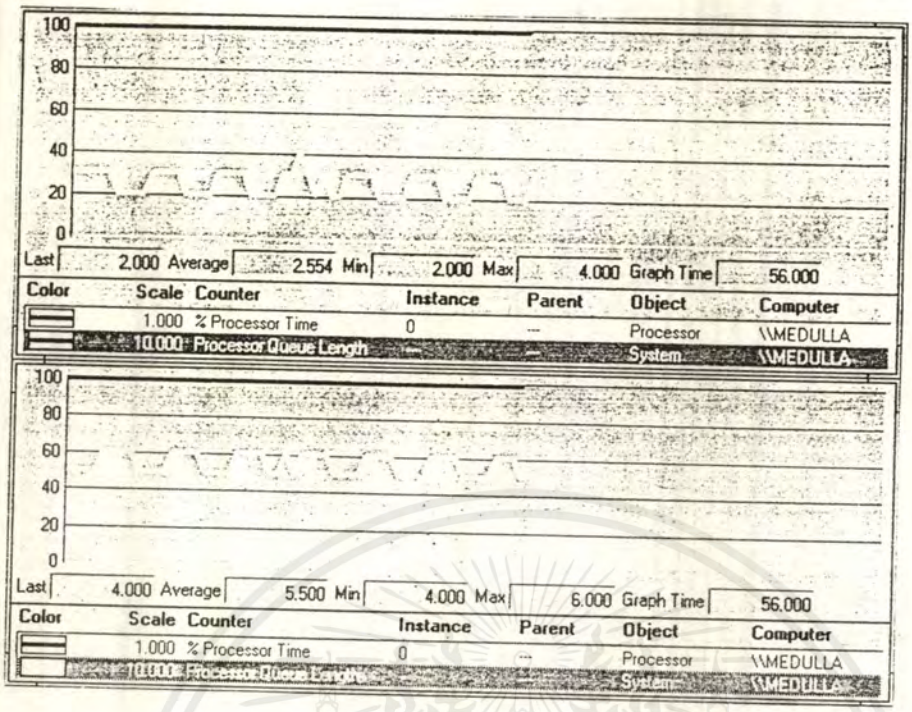
2.6 ยูสเซอร์โหมด (user mode) และ พรีวิเลจโหมด (privileged mode)

Windows NT มีกลไกในการป้องกัน 2 โหมดที่แตกต่างกัน, ในยูสเซอร์โหมด แอปพลิเคชั่นจะถูกจำกัดจำนวนในการใช้ มันไม่สามารถเข้าถึงภายนอกได้โดยตรงแต่มักจะเรียก Windows NT เพื่อที่จะทำการเอาหรือเปลี่ยนข้อมูลจากภายนอกทำให้มีความมั่นใจว่า Windows NT มีการป้องกันแอปพลิเคชั่น ไม่ให้ไปทำลายข้อมูลอื่น ๆ ของแอปพลิเคชั่นอื่น, เมื่อแอปพลิเคชั่นอยู่ในยูสเซอร์โหมด มันไม่สามารถที่จะอ่านหรือเปลี่ยนข้อมูลที่รักษาไว้โดย Windows NT ของตัวมันเองได้ เพื่อเป็นการป้องกันความผิดพลาดที่อาจจะเกิดจากความไม่ตั้งใจหรือตั้งใจ เมื่อแอปพลิเคชั่นต้องการให้ Windows NT ทำบางสิ่งบางอย่าง มันจะเรียก system routines, system routines นี้จะทำการเปลี่ยนให้ไปอยู่ในพรีวิเลจโหมด ดำเนินการตามที่ต้องการแล้วกลับคืนสู่ยูสเซอร์โหมด

2.7 ลักษณะของหลายกระบวนการ (Multiple Processes) และหลายเธร็ด (Multiple Threads)

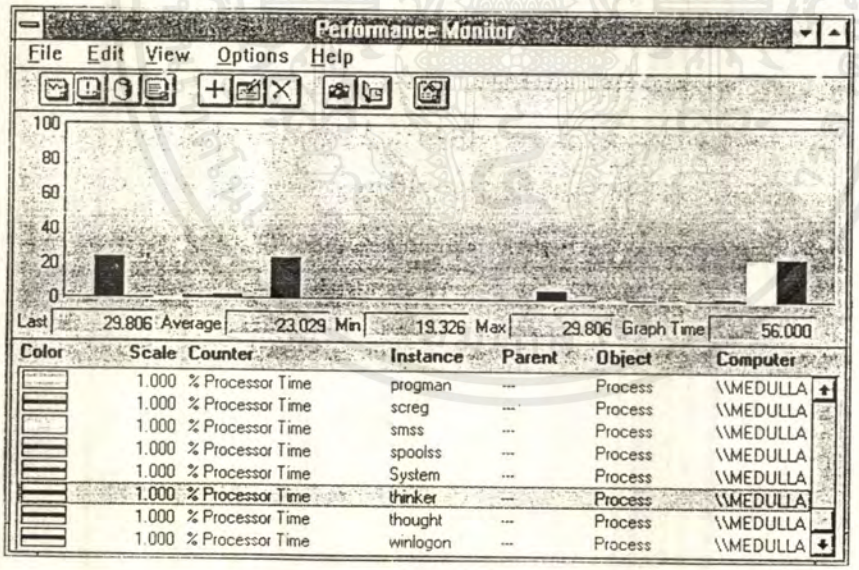
เมื่อตัวอย่างที่แล้ว เราได้ศึกษาระบบที่มีกระบวนการเดียว ที่ใช้โพรเซสเซอร์ตัวเดียว ถ้าเรามีกระบวนการมากกว่า 1 กระบวนการจะทำอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 การเปรียบเทียบความยาวของคิวของโพรเซสเซอร์

รูปนี้ได้ทำการเปรียบเทียบ Processor Queue Length ระหว่าง 2 กระบวนการ ค่าเฉลี่ยเป็น 2.5 และ 5.5 ดังนั้นเราอาจจะเดาว่ามีกระบวนการมากกว่า 3 กระบวนการในรูปล่าง

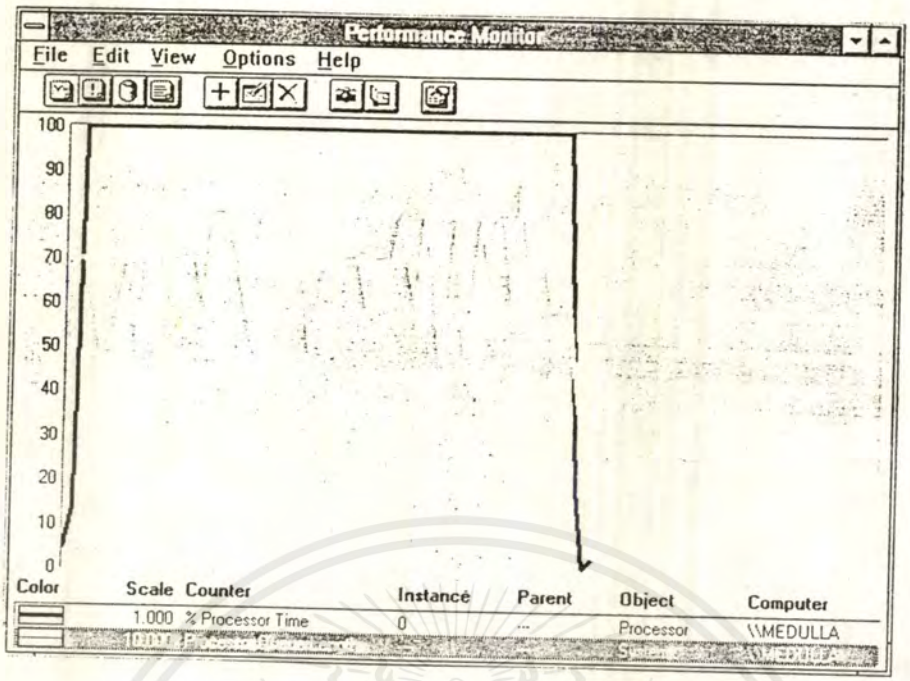


รูปที่ 2-5 แสดงกระบวนการที่กำลังใช้โพรเซสเซอร์

แสดงให้เห็นว่ามี 4 กระบวนการที่ใช้โพรเซสเซอร์

แล้วมันจะดูแตกต่างกันอย่างไร ถ้าเรามีเพียง 1 กระบวนการแต่มีหลายเธร็ด ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6 เวลาการใช้โพรเซสเซอร์และความยาวของคิวซึ่งมีหลายเธร็ด
(วัดจากระยะไกล)

Processor Queue Length จะมีความไม่สม่ำเสมอมากกว่าและมีค่าสูงกว่ากราฟที่แล้ว ถ้าเป็นแบบหลายเธร็ด

เราจะทราบอะไรบ้างถ้าเรามี 2 สิ่งใหม่นี้, เมื่อเราใช้การวัดระยะไกล (remote measurement) อย่างแรกคือเซิร์ฟเวอร์ของเครือข่าย , ซึ่งเราจะทราบเฉพาะตัวเลขเท่านั้น มันถูกเริ่มโดยไม่มีชื่อบอก, อย่างที่สองคือ Service Controller & Registry Process(SCREG)

ทั้งคู่ร่วมกันกำกับการร้องขอจากระยะไกล สำหรับการแสดงข้อมูล

คุณควรจำไว้อย่างหนึ่งว่าถึงแม้ว่าเราจะกล่าวถึงกระบวนการที่กำลังเอ็กซีคิวโปรแกรมบนระบบแต่จริง ๆ แล้วแล้วมันคือเธร็ดที่ใช้โพรเซสเซอร์ , ส่วนกระบวนการ คือ พื้นที่ที่กำหนดที่อยู่ (address space) และเธร็ดอยู่ภายในกระบวนการ มีการแบ่งกันใช้พื้นที่นี้ (address space) และมีการเอ็กซีคิวคำสั่งอย่างแท้จริง

2.8 ปัญหาคอขวดที่อุปกรณ์ที่มีผลประโยชน์ต่ำ (Low utilization)

อุปกรณ์ที่มีผลประโยชน์ 100% จะไม่เป็นคอขวดของระบบอย่างแน่นอน

มี 2 ประเด็นสำคัญที่สนับสนุนความสัมพันธ์ระหว่างความยาวของคิว (queue length) และผลประโยชน์ที่ได้รับ อย่างแรกคือ รูปแบบของการร้องขอบริการของอุปกรณ์ และอย่างที่สองคือ จำนวนรวมของงานที่อุปกรณ์ถูกร้องขอให้ทำในแต่ละการมาถึงของการร้องขอ

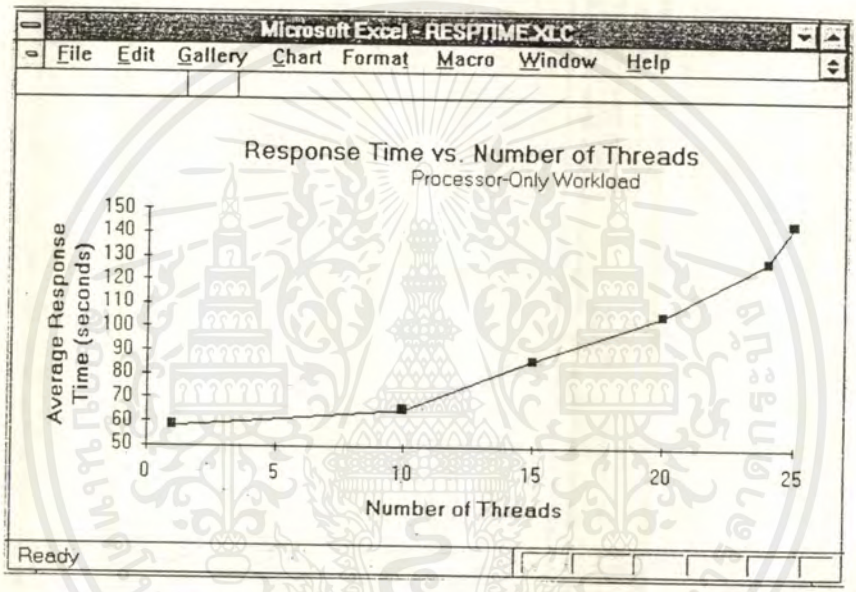
สมมติเรามี 10 เธร็ด แต่ละเธร็ดต้องการ 0.9 วินาที อย่างต่อเนื่องกันไปเป็นเวลา 10 วินาที เราลองจินตนาการดู 1 วินาทีแรก เธร็ดที่ 1 ก็ใช้ไป 0.9 วินาที และในวินาทีที่ 2 ก็ใช้ไป 0.9 วินาที เป็นอย่างนี้เรื่อยไปจนครบ 10 เธร็ด เพราะฉะนั้นโพรเซสเซอร์ จะยุ่ง (busy) 90% และไม่มีคิวเกิดขึ้นถ้าแต่ละเธร็ดใช้เวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.95 วินาที ของเวลาโพรเซสเซอร์ , โพรเซสเซอร์จะมีผลประโยชน์เท่ากับ 95% , ถ้าเป็น 0.999 ก็จะมี ผลประโยชน์ 99.9% ลักษณะนี้จะไม่มีการรอและไม่มีการแทรกระหว่างเธร็ด

ทฤษฎีของคิว คือ การมาถึงของการร้องขอที่คงที่และการแบ่งการกระจายของการบริการที่คงที่ อุปกรณ์ที่มีค่าใกล้เคียง 100% จะไม่มีการสร้างคิวขึ้น เราจะมีการแบ่งที่สมดุลกันอย่างไรระหว่างการมาถึงของการร้องขอและการบริการเพื่อให้มีความสามารถสูง

มันเป็นการไม่ยากเลยที่จะเข้าใจ, ถ้าเธร็ดที่ 2 มาถึงก่อนที่เธร็ดที่ 1 ซึ่งกำลังถูกเอ็กซีคิวอยู่ จะ เอ็กซีคิวได้มันก็จะต้องรอให้เธร็ดแรกเสร็จสมบูรณ์ก่อนที่เธร็ดที่ 2 จะสามารถเอ็กซีคิวได้ ในทำนองเดียวกัน ถ้าเธร็ดแรก ต้องการเวลาในการเอ็กซีคิวเพิ่ม ซึ่งจะไปกินเวลาของเธร็ดถัดไป ทำให้เธร็ดถัดไปต้องรอ, คิวของโพรเซสเซอร์ก็จะเริ่มขึ้น



รูปที่ 2-7 เวลาในการตอบสนองงานที่ใช้โพรเซสเซอร์แบบสุ่ม

จากรูปแสดงว่าเวลาในการตอบสนอง (response time) สามารถเพิ่มขึ้นตามจำนวนของเธร็ดที่เพิ่มสูงขึ้น ถ้าการมาถึงของการร้องขอและการร้องขอบริการอยู่ในปริมาณที่ต่ำอย่างสม่ำเสมอ เธร็ดนี้ค่อนข้างจะเป็นการสุ่มจำนวนรวมของเวลาโพรเซสเซอร์ หลังจากเกิดการหน่วงอย่างไม่สม่ำเสมอ, ผลประโยชน์จากโพรเซสเซอร์ใน 25 เธร็ด คือ 76% มีการหน่วงเวลาเกือบ 3 เท่าของเธร็ด หมายความว่า ความยาวของคิวของโพรเซสเซอร์มีค่าเกือบ 2

ตามทฤษฎีของคิว , ถ้ารูปแบบของการมาถึงของการร้องขอเป็นแบบสุ่มและรูปแบบของบริการก็เป็นแบบสุ่ม , ความยาวของคิวเป็น 2 เมื่อผลประโยชน์ของอุปกรณ์เป็น 66% หรือ 2-3 utilized

ดังนั้นถ้าจะดูว่าอุปกรณ์ถูกใช้อย่างไร ให้หาความยาวของคิว นั่นจะเป็นรูปแบบสำหรับการให้ผลประโยชน์ของอุปกรณ์ และควรจำไว้ว่า "ถ้ามีการใช้แบบสุ่มจะทำให้ความยาวของคิวเพิ่มขึ้นและผลประโยชน์ที่ได้รับก็จะต่ำ"

2.9 ระบบที่มีโพรเซสเซอร์หลายตัว

สมมติว่าเรามี 8 กระบวนการที่วิ่ง (run) อยู่ บนโพรเซสเซอร์ตัวเดียวกันจะเกิดอะไรขึ้น?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นแรกเราจะเห็นว่าโพรเซสเซอร์มีผลประโยชน์ 100% และในรูปที่ ต่อมาจะเห็นว่าโพรเซสเซอร์จะต้องมีการแบ่งกันใช้อย่างเท่า ๆ กันระหว่าง 8 กระบวนการ ดังนั้นความเร็วสูงสุดของโพรเซสเซอร์จะเป็น 12.5% เท่านั้น เราสามารถดูผลลัพธ์ได้ว่ามันช้าลงจาก Response Probe ซึ่งช้าลงไป 8.05 เท่า จากที่วัดได้จากกระบวนการเดียว ที่วิ่งบนโพรเซสเซอร์ตัวเดียว (response time) ขณะที่ทำงานชนิดเดียวกัน

ถ้าเรามี 8 กระบวนการ วิ่ง อยู่บน 8 โพรเซสเซอร์ แต่ละโพรเซสเซอร์ จะมีผลประโยชน์ 100% มีบางครั้งที่การเพิ่มโพรเซสเซอร์ ไม่ได้มีผลให้เวลาในการตอบสนองดีขึ้น ยกตัวอย่างเช่น ถ้าคอขวดไม่ได้อยู่ในที่โพรเซสเซอร์ การเพิ่มโพรเซสเซอร์ก็ไม่มีผล

มีบางปัญหาที่แฝงอยู่อาจจะปรากฏขึ้นได้ ซึ่งเกี่ยวข้องกับการแบ่งกันใช้กระบวนการ ในตัวอย่างที่แล้วนั้นกระบวนการนั้นมีความเห็นแก่ตัวอย่างที่สุด เพียงแค่มีการใช้โค้ด (code) ของกระบวนการเหล่านั้นร่วมกัน เพราะว่าโค้ดนี้อ่านได้เพียงอย่างเดียวแต่เขียนไม่ได้, แต่ละโพรเซสเซอร์ก็จะสามารถมีโค้ดสำรองทั้งในแคชตัวที่หนึ่ง (primary cache) และแคชตัวที่สอง (secondary cache) ขณะที่เอ็กซีคิว เขาก็จะไม่ต้องแบ่งการเข้าถึงไปยังหน่วยความจำ (RAM) , โปรแกรมก็จะถูกปฏิบัติโดยไม่ขึ้นต่อกัน

ทำไมหลายเธร็ด ใน 1 กระบวนการ จึงต่างจากหลายเธร็ด จากหลายกระบวนการ ? เพราะว่าหลายเธร็ด ใน 1 กระบวนการ จะอยู่ในกระบวนการเดียวกัน ที่มีการแบ่งกันใช้พื้นที่ (address space) เดียวกัน และกำลังถูกเขียนลงในหน่วยความจำที่เดียวกัน ซึ่งเป็นผลดีอย่างมากในระบบที่มีหลายโพรเซสเซอร์

2.10 การแก้ไขปัญหาคอขวดที่เกิดจากโพรเซสเซอร์

2.10.1 การแก้ปัญหาโดยใช้ฮาร์ดแวร์ (hardware)

- การควบคุมอินเทอร์รัปต์ที่มากเกินไป

สาเหตุทั่ว ๆ ไปที่ทำให้เกิดอินเทอร์รัปต์มากเกินไปก็คือ การออกแบบดีไวซ์ ไดร์ฟเวอร์ (device driver) ที่ไม่ดี คุณเคยลองวิ่งดีไวซ์ ไดร์ฟเวอร์บ้างหรือไม่ เราเคยลองวิ่งวิดีโอ ไดร์ฟเวอร์(video driver) ซึ่งปล่อยอินเทอร์รัปต์ออกมาเป็นพัน ๆ อินเทอร์รัปต์ต่อวินาที คุณสามารถทดสอบได้โดยวิ่ง standard VGA driver และเปรียบเทียบกับอินเทอร์รัปต์ก่อนและหลัง

- ใช้อแด็พเตอร์ (adapter) 32 บิต เมื่อมีการส่งข้อมูลกับหน่วยความจำหรือใช้ direct memory access (DMA)

สมมติว่าคุณมีเครื่อง 486 คุณอาจจะเกิดปัญหากับเครือข่าย (network) หรือดิสก์อแด็พเตอร์การ์ด (disk adapter cards) ที่คุณได้เลือกไว้ก็ได้ 8-บิต การ์ด จะใช้เวลาของโพรเซสเซอร์ มากกว่า 16-บิต หรือ 32-บิต การ์ด เสียอีกจำนวนบิตนี้หมายถึงจำนวนของข้อมูลที่ถูกเคลื่อนย้ายไปยังหน่วยความจำจากอแด็พเตอร์ในการส่งข้อมูลแต่ละครั้ง อแด็พเตอร์ จะไม่ใช้ memory-mapped buffer หรือ DMA จะต้องใช้คำสั่งของโพรเซสเซอร์ในการเคลื่อนย้ายข้อมูลและอาจจะทำให้โพรเซสเซอร์ยุ่งขึ้น DMA ใช้หน่วยความจำและทำให้โพรเซสเซอร์ช้าลงแต่ก็ยังมีประสิทธิภาพกว่าใช้คำสั่งแยกต่างหาก

- เพิ่มขนาดของหน่วยความจำสำรอง (secondary cache)

การเพิ่มขนาดของหน่วยความจำโดยปราศจากการเพิ่มขนาดของหน่วยความจำสำรองนั้นจะทำให้การทำงานของโพรเซสเซอร์แย่ลง ที่เป็นอย่างนี้เพราะว่าหน่วยความจำสำรองจะต้องแมป (map) กับหน่วยความจำที่ใหญ่ขึ้น ซึ่งผลลัพธ์ที่ได้ออกมาก็คือ จะเกิดอัตราการพบข้อมูล (hit rates) ที่ต่ำในแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(cache) และจะทำให้โปรแกรม processor-bound ช้าลง เพราะมันจะกระจายไปทั่วทั้งหน่วยความจำที่ถูกเพิ่มขนาดขึ้น ถ้าคุณสงสัยว่ามันจะช้าลงจริงหรือไม่ ลองทำการทดสอบ processor-bound ด้วย Response Probe ที่จะสัมผัสกับหน่วยความจำจำนวนมากแต่ปรับขนาดของหน่วยความจำเดิมไม่ให้สามารถรับการเพจ (page) ได้ ให้ทำการทดสอบก่อนและหลังการเพิ่มหน่วยความจำและคุณก็จะเห็นว่าการทดสอบนั้นช้าลงเมื่อหน่วยความจำเพิ่มขึ้น

- ซื้อโพรเซสเซอร์ที่มีความเร็วเพิ่มขึ้นอีกหรือเพิ่มโพรเซสเซอร์เป็นหลาย ๆ โพรเซสเซอร์

สิ่งสุดท้ายที่จะทำก็คือซื้อสิ่งที่เพิ่มอำนาจ power) ระบบแต่ก็ไม่ได้หมายความว่าคุณจะต้องใช้เงินมากมาย ให้จำไว้ว่ามี 2 วิธี ที่จะเพิ่มความเร็วให้กับระบบด้วยการจัดการเกี่ยวกับซีพียู (CPU) โดยวิธีแรก ซื้อซีพียูที่มีความเร็วขึ้น และอีกวิธีหนึ่งก็คือ การเพิ่มวีพียูให้กลายเป็นคอมพิวเตอร์ที่มีซีพียูหลายตัว คุณอาจจะพบว่า โพรเซสเซอร์ตัวที่สองอาจทำงานไม่ถึง 100 เปอร์เซ็นต์เต็มและเป็นที่น่าเสียใจสำหรับบางโปรแกรมที่โพรเซสเซอร์ตัวที่สองอ้างถึงนั้นไม่มีการปรับปรุงทั้งหมดที่เป็นอย่างนั้นก็เพราะว่าโปรแกรม NT ทั้งหมดไม่ได้ออกแบบมาสำหรับมัลติเธร็ด (multithreaded) ดังนั้นการเพิ่มโพรเซสเซอร์ก็ไม่มีประโยชน์ คุณสามารถที่จะทดสอบได้ว่าแอพพลิเคชันนี้เป็นซิงเกิลเธร็ดหรือไม่โดยดูที่ Process/% Processor Utilization และล็อก (log) กิจกรรมทั้งหมดของโพรเซสเซอร์ทุก ๆ ตัว ถ้าคุณมีระบบดิวอัล-โพรเซสเซอร์ (dual-processor) และมีโพรเซสเซอร์หนึ่งตัวที่ทำงานหนัก (สูงกว่า 50 % ยูทิลไลเซชัน) และโพรเซสเซอร์อีกตัวหนึ่งที่ไม่ได้ทำอะไรเลย ซึ่งสามารถพิสูจน์ได้ว่าแอพพลิเคชันนี้เป็นแบบซิงเกิลเธร็ด และคุณจำเป็นที่จะต้องร้องขอแอพพลิเคชันเดเวลอปเปอร์ (application developer)

2.10.2 การแก้ปัญหาโดยใช้ซอฟต์แวร์ (software)

- การควบคุมอินเทอร์รัปต์ที่มากเกินไป

และอีกสาเหตุหนึ่งที่ทำให้เกิดอินเทอร์รัปต์มากเกินไปก็คือ โปรแกรม ไทม์เมอร์-ไดร์ฟเวอร์ (timer-driver) ที่หนึ่งเครือข่ายจะจัดการให้มีอินเทอร์รัปต์ได้ 4000 อินเทอร์รัปต์ต่อวินาทีบนเครื่อง 486 ที่เป็นไฟล์เซิร์ฟเวอร์ (file server) หลังจากทำงานทุกอย่างกับระบบแล้วให้ทำการเปิด Schedule+ ใน Start Group และปิดมันลงแล้วอินเทอร์รัปต์ต่อวินาทีก็จะลดลงมาในอัตราที่ปกติ

- เคลื่อนย้ายบางโปรแกรมที่มีโหลดมาก ๆ ออกไป

การหยุดใช้ทรัพยากรที่มากเกินไปก็จะทำให้หยุดการร้องขอที่มากด้วยเช่นกัน เช่นถ้าคุณวิ่งโปรแกรม SQL Server 6.5 บนเครื่อง 25-MHz 486SX ด้วยแรม (RAM) 24 MB ทางที่จะช่วยได้ก็คือเคลื่อนย้าย SQL ออกไป

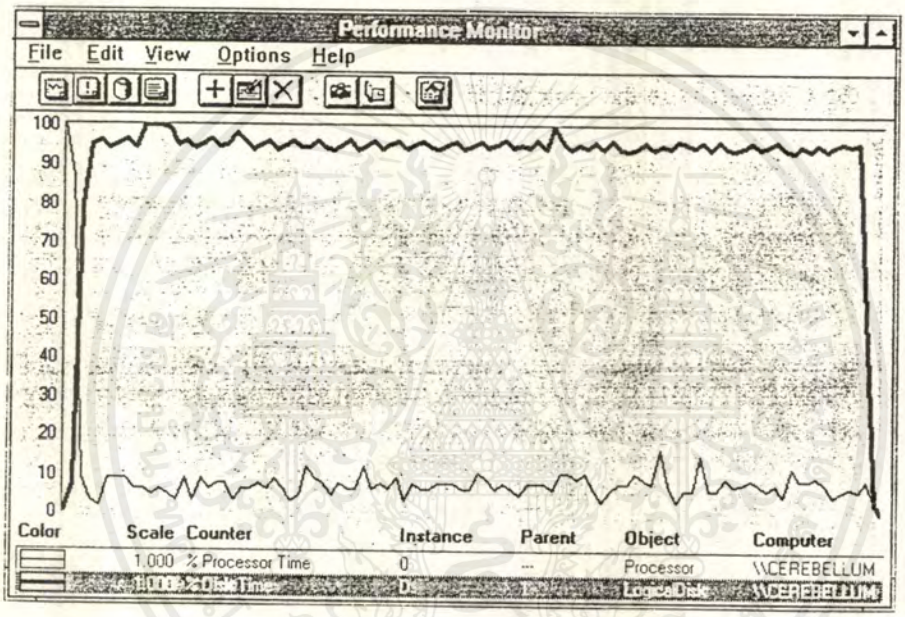
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การตรวจจับปัญหาคอขวดที่เกิดจากดิสก์

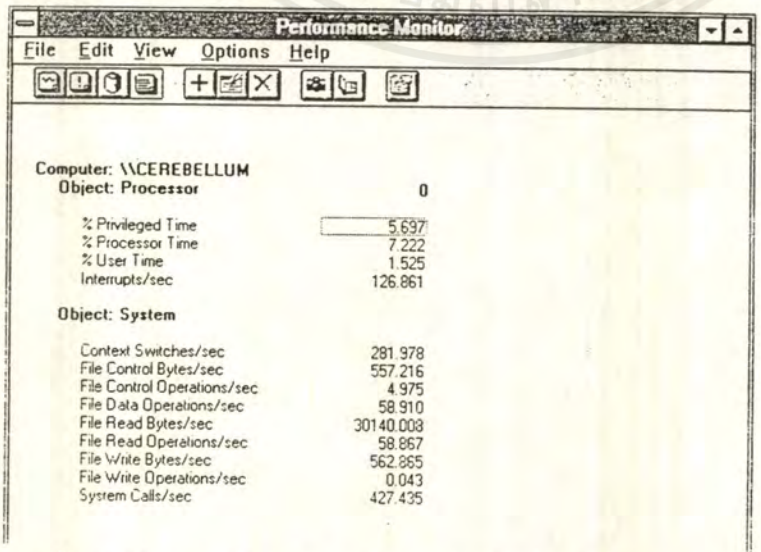
ดิสก์เก็บโปรแกรมและข้อมูลที่โปรแกรมดำเนินการเมื่อเรากำลังรอการตอบสนองของคอมพิวเตอร์ นั่นคือการเกิดปัญหาคอขวดของดิสก์

ความแตกต่างของฟิสิคอลดิสก์ (physical disk) กับลอจิคอลดิสก์ (logical disk) คือ ฟิสิคอลดิสก์คือจำนวนหน่วยของดิสก์ทางกายภาพอันเดียวเดี่ยว ๆ ขณะที่ลอจิคอลดิสก์ คือดิสก์ที่ทำพาร์ติชัน (partition) ยกตัวอย่างเช่นดิสก์ไดรว์เดี่ยวๆ อาจจะมี 2 พาร์ติชัน, ดิสก์เดี่ยว ๆ เป็นฟิสิคอลดิสก์ ตัวอย่างเช่น แทนฟิสิคอลดิสก์ด้วย 0 กับ 2 และลอจิคอลดิสก์ แทนด้วย C และ D



รูปที่ 3-1 ลักษณะการใช้โพรเซสเซอร์และดิสก์

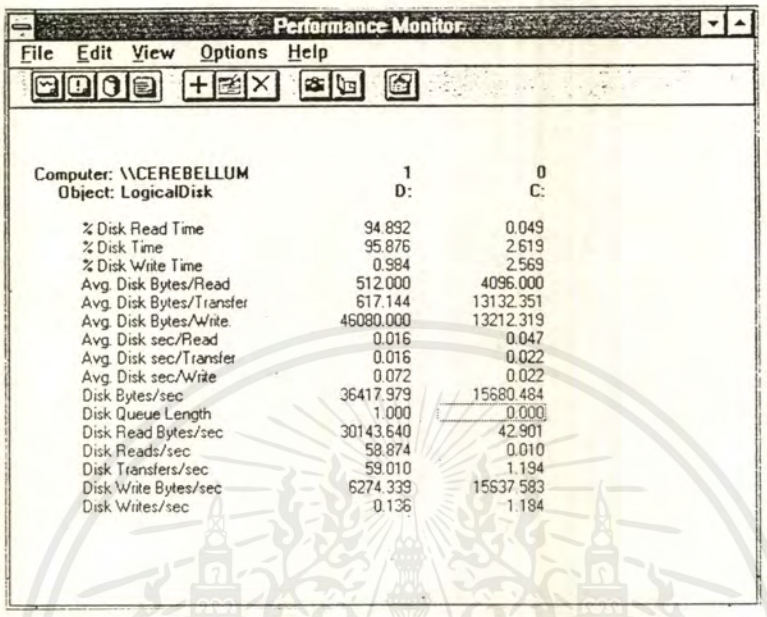
จากรูปนี้ โพรเซสเซอร์ไม่ได้เป็นคอขวดแน่นอน แต่ดิสก์เป็นคอขวด



รูปที่ 3-2 ข้อมูลของโพรเซสเซอร์และระบบเมื่อมีการอ่านเรคคอร์ด (record) ขนาดเล็กจากดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรเซสเซอร์ให้ผลประโยชน์เพียงแค่ 7.2% เท่านั้น แต่อัตราการอินเทอร์รัปมีค่าสูงกว่า 66.667 Interrupt/sec ถึง 60.194 Interrupt/sec เราคาดว่าน่าจะมาจากคล็อก (clock) ของระบบที่อยู่เฉย ๆ ที่แยกไว้ต่างหากจากเครือข่าย, หรือว่ามันจะสามารถมาจากดิสก์ ? เราจะทำการตรวจสอบต่อไป

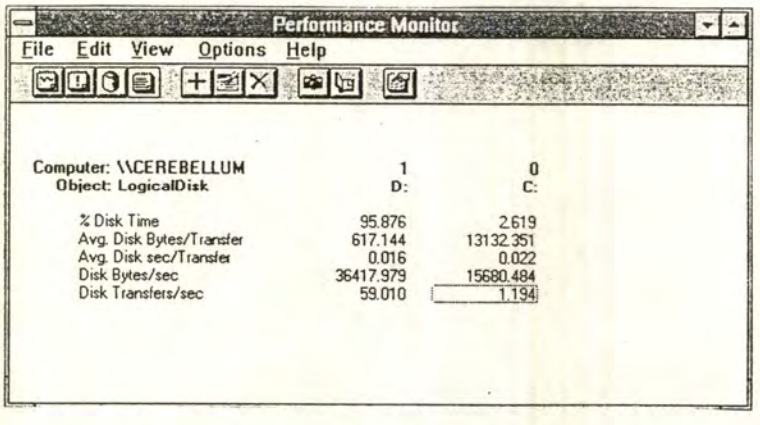


รูปที่ 3-3 ข้อมูลของดิสก์ในขณะอ่านเรคคอร์ดขนาดเล็กโดยทำการล็อคจากคนละที่

ทำการเปรียบเทียบระหว่างไดร์ฟ D กับไดร์ฟ C จากรูปที่ 3-2 ยังจำได้หรือไม่ว่า interrupt/sec ที่เกินมา 60.194 ถ้าเราลองเอา disk transfer/sec จากไดร์ฟ D บวกเข้ากับ disk transfer /sec ของไดร์ฟ C เราจะได้ค่าเท่ากับ 60.204 ซึ่งมีค่าใกล้เคียงกับ 60.194 เราสงสัยว่ามันน่าจะเป็นมากกว่าเหตุการณ์บังเอิญ, เราอาจจะคาดว่าเคาน์เตอร์ของระบบอาจจะมีส่วนเกี่ยวข้องบางส่วนกับเคาน์เตอร์ดิสก์ก็ได้

เนื่องจาก Performance Monitor แสดงค่าเฉลี่ยต่าง ๆ มากมาย ซึ่งค่าเฉลี่ยเหล่านี้ได้แสดงให้เห็นถึงหลักความจริงขั้นพื้นฐาน ซึ่งได้มีรายละเอียดที่สำคัญมากได้ถูกซ่อนเอาไว้

มีข้อแตกต่างที่น่าสนใจระหว่างข้อมูลบนไดร์ฟ D และ ไดร์ฟ C ดูรูปที่ 3-4 ซึ่งมีการแยกผลที่แสดงไว้ต่างหากระหว่างไดร์ฟ D และไดร์ฟ C



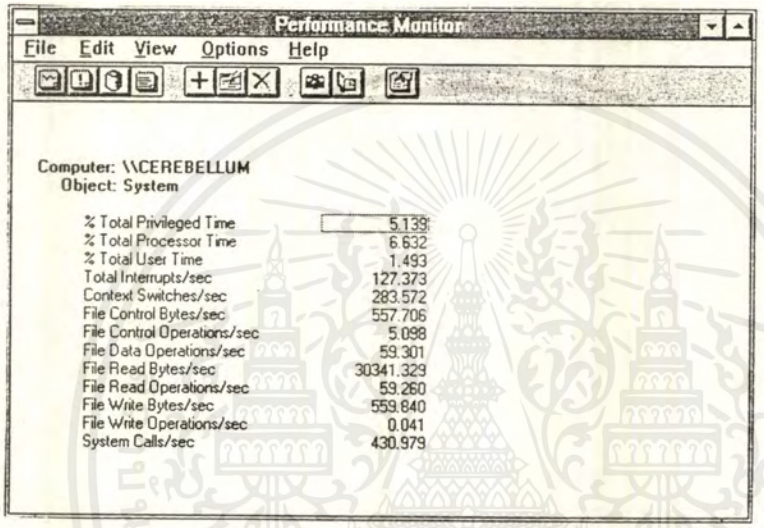
รูปที่ 3-4 ความแตกต่างระหว่างอัตราการโอนย้าย (transfer) ข้อมูลของดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

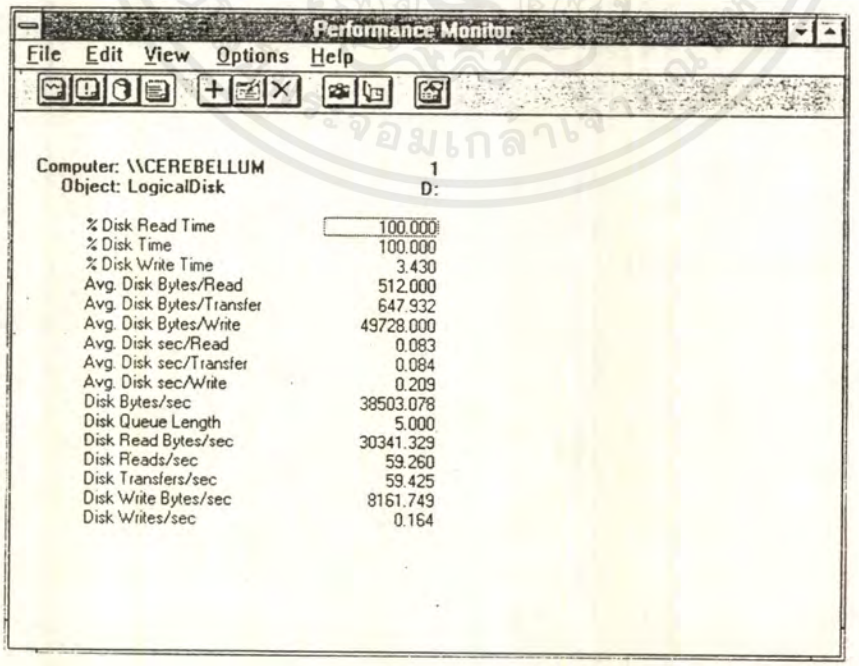
ไดร์ฟ C มีการดำเนินการที่มีประสิทธิภาพมากกว่าอย่างชัดเจน เหตุผลก็คือ ไดร์ฟ C มี Avg. Disk Bytes/Transfer ที่มีค่าสูงมากกว่า 20 เท่า ของไดร์ฟ D

ถ้าคุณพยายามที่จะหาตำแหน่งของคอขวดที่ดิสก์ หลังจากบันทึกกิจกรรมต่าง ๆ ของดิสก์แล้ว ให้ดูที่ขนาดเฉลี่ยของการเคลื่อนย้ายข้อมูล สิ่งนี้เป็นกุญแจสำคัญในการหาประสิทธิภาพของ ดิสก์

เปรียบรูปที่ 3-2 กับ 3-5 ในความรู้สึกแล้วรูปที่ 3-5 เป็นแบบอย่างที่ถูกต้องเนื่องจากว่าถ้าระบบมีหลายๆ โพรเซสเซอร์ รูปที่ 3-2 จะแสดงข้อมูลให้ทราบได้เพียง 1 โพรเซสเซอร์จากหลาย ๆ โพรเซสเซอร์ ซึ่งรูปที่ 3-5 จะมีการคำนวณผลรวมต่าง ๆ ไว้แล้ว ซึ่งจะง่ายในการเปรียบเทียบ



รูปที่ 3-5 ข้อมูลระบบของ 5 กระบวนการที่กำลังอ่านเรคคอร์ดขนาดเล็กพร้อมกัน



รูปที่ 3-6 ข้อมูลดิสก์ของ 5 กระบวนการที่กำลังอ่านเรคคอร์ดขนาดเล็กพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดสำคัญก็คือ สถิติที่ได้นี้มีค่าใกล้เคียงกับที่พิสูจน์ ถึงแม้ว่าเราจะเพิ่มกระบวนการไปอีกถึง 4 กระบวนการ ก็ไม่ได้ทำให้ค่าเปลี่ยนแปลงไปมากเลย ลักษณะอย่างนี้เองที่เราเรียกว่าคอขวด สถิติของดิสก์ ไม่เปลี่ยนแปลงก็เพราะว่าดิสก์ ถึงขีดสูงสุดแล้ว

Avg. Disk sec/Reads = %Disk time /Disk Reads/sec=1/59.260=0.16875 ผลลัพธ์ที่ได้คือ 16.875 มิลลิวินาที ซึ่งต่างจากรูปที่ 3-6 ซึ่งมีค่า Avg. Disk sec/Read เท่ากับ 0.083 เกิดอะไรขึ้น? 160875 มิลลิวินาที เป็นตัวเลขที่ใช้ในการรับเอาแต่ละการร้องขอจาก ดิสก์ แต่ค่าที่รายงานได้ 0.083 นี้เป็นเวลาที่จ่ายให้กับดิว บวกกับเวลาที่รับเอาข้อมูลดิสก์ ถ้าเรานำเอา 0.016875 ไปหาร 0.083 จะได้ค่าเท่ากับ 409 ซึ่งใช้เป็นค่าความยาวของคิวเฉลี่ยได้เป็นอย่างดี และดีกว่าค่าเคาน์เตอร์ที่ให้ค่า Disk Queue Length ที่เกิดขึ้นในช่วงเวลาสั้น ๆ

3.1 ดิสก์ที่มีธรูพุท (throughput) สูง

การสร้างไฟล์โดยใช้บัฟเฟอร์ (buffer) หรือใช้แคชเข้ามาเกี่ยวข้องนี้จะเร็วกว่าการเข้าถึงที่ไม่ใช้บัฟเฟอร์ ถ้ามีการสร้างไฟล์ จะตั้งให้มีการใช้บัฟเฟอร์

จะสามารถบอกได้ว่าดิสก์ยุ่งขนาดไหน โดยดูจากผลประโยชน์ที่ได้รับ แต่คุณจะไม่รู้เลยว่ามีงานมากมายเท่าไร เว้นเสียแต่ว่าคุณดูที่อัตราการเคลื่อนย้ายข้อมูล

ถ้าต้องการจะเพิ่มธรูพุทของดิสก์ ก็ให้เพิ่มขนาดของเรคคอร์ด จนกระทั่งถึง 64 K ในกรณีนี้เราสูญเสียการหมุนไปทุก ๆ ครั้ง และธรูพุทก็จะลดลงตาม ถ้าเราไม่เสริมกำลังการอ่านซ้ำของเรคคอร์ดซ้ำแล้วซ้ำอีก เราจะไม่รอให้ดิสก์หมุนจนครบรอบ เพื่อจะเริ่มต้นเรคคอร์ดในแต่ละเวลา เราจะใช้วิธีการเข้าถึงแบบสุ่มมากกว่า และ cost ของจำนวนรอบที่เพิ่มขึ้นมามีค่าน้อยมากยากที่จะสังเกต

การเพิ่มขนาดไฟล์เราควรจะดูด้วยว่าทำให้ธรูพุทลดลงหรือไม่ ขณะที่ดิสก์ใช้เวลาในการค้นหามากขึ้น

3.2 การแก้ไขปัญหาคอขวดที่เกิดจากดิสก์

3.2.1 การแก้ปัญหาโดยใช้ฮาร์ดแวร์

- ใช้ซีคไทม์ (seek time) ที่เร็วขึ้น

การวัดประสิทธิภาพของ ดิสก์มีสองประเด็นใหญ่ ๆ คือ ซีคไทม์และ การส่งข้อมูล (data transfer) ซีคไทม์เป็นคุณสมบัติเฉพาะตัวของแต่ละไดร์ฟที่คุณซื้อ ส่วนอัตราการส่งข้อมูลหาได้โดยชนิดของดิสก์โฮสต์อแดพเตอร์ (disk host adapter) ที่คุณซื้อดิสก์ที่เร็วกว่าก็จะสามารถหาข้อมูลได้โดยใช้เวลาที่น้อยกว่า ปัจจุบันนี้ไดร์ฟมีซีคไทม์ที่เป็นเพียงเลขหลักเดียวเท่านั้น ซึ่งควรจะซื้อไดร์ฟที่มีดิสก์ไทม์ตามนี้

- ใช้อัตราการส่งข้อมูลที่ดีกว่า

อัตราการส่งข้อมูลคือ ส่วนของดิสก์คอนโทรลเลอร์ (disk controller) หรือโฮสต์อแด็พเตอร์ ซึ่งต่อไปนี้คือ ลักษณะเล็กน้อยที่ควรจะจดจำไว้เมื่อซื้อดิสก์คอนโทรลเลอร์ใหม่

1. ซื้อ 32-บิต SCSI โฮสต์อแด็พเตอร์
2. ใช้บัสมาสเตอร์ริงโฮสต์อแด็พเตอร์ (bus mastering host adapter) มี 3 วิธีที่จะนำข้อมูลจากโฮสต์อแด็พเตอร์ไปยังหน่วยความจำของคอมพิวเตอร์: ใช้โปรแกรมอินพุท/เอาท์พุท (program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

input/output) (PIO), ใช้ไดเร็กเม็มโมรีแอ็คเซส (direct memory access) และ บัสมาสเตอร์ริง ซึ่งบัสมาสเตอร์ริงเป็นวิธีที่เร็วที่สุดใน 3 วิธีคุณสามารถที่จะซื้อ MCA, EISA หรือ PCI buses ก็ได้

3. ใช้โฮสต์อแด็พเตอร์ที่สนับสนุนอะซิงโครนัสอินพุท/เอาท์พุท (asynchronous input/output) SCSI-II หรือ SCSI-III โฮสต์อแด็พเตอร์ ส่วนใหญ่จะอนุญาตให้มีหลาย ๆ ไดร์ฟ เพื่อที่จะทำงานเป็นอิสระต่อกัน นั่นหมายความว่าคุณสามารถที่จะซื้อไดร์ฟหลาย ๆ ไดร์ฟ และทุก ๆ ไดร์ฟทำการซีคที่เวลาเดียวกัน ควรจำไว้ว่าโฮสต์อแด็พเตอร์ส่วนใหญ่จะไม่สนับสนุนอย่างนี้ นั่นหมายความว่า คุณสามารถมีโฮสต์อแด็พเตอร์ที่มี 5 ฮาร์ดดิสก์แต่มี 1 ไดร์ฟเท่านั้นที่ทำงานในเวลาใดเวลาหนึ่ง ดังนั้นการใช้หลาย ๆ ไดร์ฟก็จะไม่เกิดประโยชน์อะไรเลยต่อความเร็วของระบบคุณ

4. ซื้อ SCSI ซึ่งอีกสาเหตุหนึ่งก็คือ ATDISK และ ATAPI เป็นไดร์ฟเวอร์ที่ถูกสร้างขึ้นใน IDE และ EIDE ไดร์ฟ ซึ่งไม่สามารถสนับสนุนอะซิงโครนัสดิสก์ไอโอ (asynchronous disk I/O) นั้นหมายความว่าถ้าคุณมีคอมพิวเตอร์ที่มี EIDE ทั้ง 2 ฮาร์ดดิสก์ มันจะไม่วิ่งที่เวลาเดียวกัน ยกตัวอย่างเช่น สมมติไดร์ฟหนึ่งมีชื่อว่า C: และอีกไดร์ฟหนึ่งมีชื่อว่า D: และคุณก็สำเนาไฟล์จากไดร์ฟหนึ่งไปยังอีกไดร์ฟหนึ่ง ถ้าคุณสามารถเห็นการสำเนาข้อมูลเป็นแบบ มัลติเซ็คคัน-บาย-มัลติเซ็คคัน เบซิส (multisecond-by-multisecond basis) คุณจะเห็นทั้ง C: และ D: ไม่ได้วิ่งที่เวลาเดียวกัน (มันวิ่งสลับกัน) อย่างนี้จะทำให้เสียเวลามากซึ่งเป็นสาเหตุหนึ่งว่าทำไมคุณจึงต้องใช้อะซิงโครนัสไดร์ฟ

5. สร้างสไตร์ฟเซ็ท (stripe set) ด้วยหลาย ๆ ไดร์ฟ เพราะว่าสไตร์ฟเซ็ทเหล่านี้จะกระจายข้อมูลของดิสก์ไปทั่วทุกไดร์ฟ จะทำให้อ่านข้อมูลได้เร็วขึ้นเพราะว่าฟิสิคอลไดร์ฟ (physical drive) ทั้งหมดในสไตร์ฟเซ็ทสามารถที่จะทำงานแบบขนานได้

- แบ่งให้แต่ละเซิร์ฟเวอร์มีโหลดที่เท่ากัน

ถ้าคุณไม่สามารถซื้อฮาร์ดแวร์ที่ทำให้เร็วขึ้นได้ คุณก็ควรที่จะกระจายงานออกไปเท่า ๆ กัน ถ้าคุณมีเซิร์ฟเวอร์หลาย ๆ ตัว คุณก็ควรที่จะพยายามเคลื่อนย้ายแอปพลิเคชั่นจากเซิร์ฟเวอร์หนึ่งไปยังเซิร์ฟเวอร์หนึ่งให้แต่ละเซิร์ฟเวอร์มีโหลดเท่า ๆ กัน

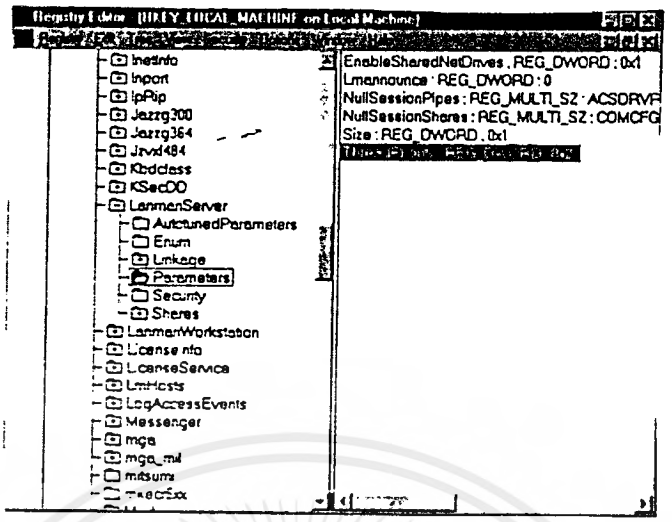
3.2.2 การแก้ปัญหาโดยใช้ซอฟต์แวร์

ถ้าวินโดวส์ NT 4.0 เซิร์ฟเวอร์ ใช้พรินต์เซิร์ฟเวอร์ร่วมกับไฟล์เซิร์ฟเวอร์ มีวิธีการที่จะเพิ่มลำดับความสำคัญ (priority) ของเธร็ดที่ควบคุมการให้บริการไฟล์ และลดลำดับความสำคัญของเธร็ดที่ควบคุมการให้บริการพิมพ์ข้อมูล การเพิ่มลำดับความสำคัญให้กับเธร็ดของไฟล์เซิร์ฟเวอร์จะทำให้การร้องขอไฟล์ควบคุมได้เร็วขึ้น โดยยอมเสียการให้บริการพิมพ์ข้อมูล

โดยทั่วไปแล้ว พรินต์เซิร์ฟเวอร์เธร็ดจะถูกเซ็ทเป็น 2 และไฟล์เซิร์ฟเวอร์เธร็ดจะถูกเซ็ทเป็น 1 ตัวเลขที่มากกว่าจะเป็นลำดับความสำคัญที่สูงกว่า การเปลี่ยนลำดับความสำคัญของไฟล์เซิร์ฟเวอร์เธร็ดด้วย Registry Editor มีขั้นตอนดังนี้

1. จาก start menu เลือก Run เพื่อที่จะเปิด Run dialog
2. ในกล่องของตัวหนังสือ, พิมพ์ "regedt32" แล้วกด OK เพื่อเริ่ม Registry Editor ดังรูปที่ 3-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-7 การใช้ Registry Editor ในการเปลี่ยนค่าของระบบวินโดวส์เอ็นที

3. เลือก HKEY_LOCAL_MACHINE
4. ขยายไดเร็กทอรีออกเป็น SYSTEM\CurrentControlSet\Services\Lanmanserver\Parameters
5. เลือก ThreadPriority entry

   ข้อควรจำ: ทุกระบบจะไม่มี ThreadPriority เสมอไป ถ้าระบบของคุณไม่มีเอ็นทริ (entry) นี้ ให้ใช้เมนู Edit แล้วเซ็ทค่าเริ่มต้นเป็น 2
6. เปลี่ยนจาก 1 เป็น 2 เพื่อเพิ่มลำดับความสำคัญให้กับไฟล์เซิร์ฟเวอร์เธร็ด
7. ปิด Registry Editor

ถ้าเครือข่ายของคุณประกอบไปด้วยวินโดวส์ NT เซิร์ฟเวอร์หลาย ๆ เครื่อง สามารถที่จะใช้เทคนิคที่เรียกว่า "load balancing" เพื่อกระจายโหลดของงานไปยังเซิร์ฟเวอร์ต่าง ๆ

ระบบ load balancing แต่ละเซิร์ฟเวอร์จะทำการหากิจกรรมที่ทำมากที่สุดที่พบ โดยจะมีการสำเนาตัวเองของไฟล์และโฟลเดอร์ (folder) และส่งไปให้ยังเซิร์ฟเวอร์ต่าง ๆ เพื่อให้ระบบเครือข่ายมีโหลดที่สมดุลย์กัน และยังช่วยลดการจราจรของเซิร์ฟเวอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับปัญหาคอขวดที่เกิดจากหน่วยความจำ

- หน่วยความจำทำให้เกิดประสิทธิภาพต่ำได้มากกว่าทรัพยากรอื่น ๆ เพราะหน่วยความจำ ทำให้คอมพิวเตอร์ต้องอ่านและเขียนจากดิสก์บ่อย ๆ และการเข้าถึงดิสก์ก็ช้ากว่าการเอ็กซีคิวคำสั่งในโพรเซสเซอร์มาก
- Windows NT เป็นระบบหน่วยความจำเวอร์ชวล (virtual memory) และใช้วิธีการเพจจิ้ง (paging) คือ เมื่อมันเอ็กซีคิวโปรแกรม มันจะไม่เก็บโปรแกรมทั้งหมดลงในหน่วยความจำ แต่จะเก็บบางส่วนเท่านั้น โดยแบ่งเป็นเพจ (page) แต่ละเพจก็จะอยู่ในหน่วยความจำในช่วงเวลาใดเวลาหนึ่ง เมื่อคำสั่งในโปรแกรมร้องขอเพจของโค้ดหรือข้อมูล ซึ่งไม่ได้อยู่ในหน่วยความจำ Windows NT ก็จะนำเพจนั้นมาจากดิสก์ ซึ่งเพียง 1 คำสั่งก็อาจทำให้เกิดการกระทำของ I/O จำนวน 1,2 เพจ หรือมากกว่านั้นก็ได้
- ในการเอ็กซีคิวคำสั่งนั้น โดยเฉลี่ยแล้วใช้เวลาในหน่วยของ 100 นาโนวินาที (nanoseconds) แต่การเข้าถึง ดิสก์จะใช้เวลาในหน่วย 10 มิลลิวินาที (milliseconds) จะเห็นว่า 1 เพจที่ไม่พบสำหรับ 1 คำสั่ง จะทำให้เครื่องวิ่งช้าลงกว่าปกติถึง 100,000 เท่า ด้วยเหตุนี้หน่วยความจำจึงเป็นสาเหตุสำคัญที่ทำให้เกิดคอขวด

4.1 การทำงานของระบบหน่วยความจำเวอร์ชวลของ Windows NT

Windows NT เป็นระบบปฏิบัติการชนิด 32 บิต ที่สามารถวิ่งแอปพลิเคชั่นได้ทั้งแบบ 16 บิต และ 32 บิต เมื่อระบบเรียกใช้แอปพลิเคชั่น 16 บิต มันก็จะถูกเปลี่ยนให้เป็น 32 บิต หมายความว่าโปรแกรมสามารถเห็นพื้นที่ขนาด 32 บิต ซึ่งแปลงเป็นหน่วยความจำเวอร์ชวล 4 จิกะไบต์ ครึ่งบนของหน่วยความจำนี้จะเป็นโค้ดและข้อมูลของระบบ ซึ่งกระบวนการอยู่ในพรีวิเลจโหมดเท่านั้นที่จะมองเห็นหน่วยความจำส่วนนี้ได้ ส่วนครึ่งล่าง (2 จิกะไบต์) จะถูกใช้โดยโปรแกรมของยูสเซอร์ (user program) ซึ่งอยู่ในยูสเซอร์โหมด

RAM ใน Windows NT แบ่งเป็น 2 ชนิด : นอนเพจ (nonpaged) และ เพจ (paged) ซึ่งโค้ดหรือข้อมูลนอนเพจ ต้องอยู่ในหน่วยความจำ ซึ่งไม่สามารถเขียนทับหรือดึงออกมาจากอุปกรณ์ต่อพ่วงต่าง ๆ ได้ (อุปกรณ์ต่อพ่วง เช่น ดิสก์, แลน, ซีดีรอม และอุปกรณ์อื่น ๆ) ส่วนหน่วยความจำแบบเพจนั้น สามารถใช้และนำหลาย ๆ เพจ จากอุปกรณ์ต่อพ่วงกลับมาใช้อีกได้ หน่วยความจำแบบเพจถูกแบ่งออกเป็น เพจเฟรม (paged เฟรมs) อีก ซึ่งเพจเฟรมนี้จะประกอบด้วยหลาย ๆ เพจ เหมือนกับกรอบรูปภาพที่ประกอบไปด้วยหลาย ๆ รูปภาพ

ขนาดของเพจจะต่างกันไปขึ้นกับชนิดของโพรเซสเซอร์ สำหรับ 386, 486 และเพนเทียม โพรเซสเซอร์ ขนาดเพจ = 4096 ไบต์ (4K) เช่นเดียวกับ MIPS® โพรเซสเซอร์ ส่วน DEC® Alpha โพรเซสเซอร์ มีขนาดเพจ = 8192 (8K) ด้วยความแตกต่างของขนาดของเพจ ทำให้

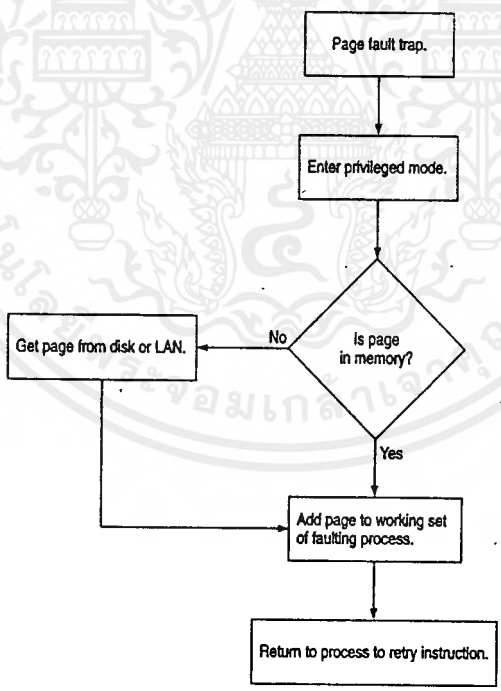
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038953

เคาน์เตอร์ของ Performance Monitor มีหน่วยเป็นไบต์ ดังนั้นข้อมูล 100 เพจ จะมีจำนวนข้อมูลไม่เท่ากันในทุก ๆ คอมพิวเตอร์

เมื่อระบบต้องการเพจของโค้ดหรือข้อมูลจากอุปกรณ์ต่อพ่วง ตัวจัดการหน่วยความจำ (memory manager) ของWindows NT ก็จะหาเพจเฟรมที่ว่าง เพื่อที่จะใส่เพจที่ต้องการลงไป ถ้าไม่มี เพจเฟรมใดว่าง ตัวจัดการหน่วยความจำก็จะเลือกเพจเฟรมใดก็ได้ที่ไม่ได้ถูกใช้มานานแล้วแทน แต่ เพจที่ถูกเปลี่ยนนี้จะต้องถูกเขียนกลับไปในพื้นที่เดิมบนอุปกรณ์ต่อพ่วงนั้น ๆ ก่อนที่เพจใหม่จะแทนที่ลงไป

โปรแกรมจะเอ็กซีคิวโดยการอ่าน (fetch) คำสั่งทีละคำสั่งจากโค้ดเพจ จนกระทั่งมีการเรียก (call) หรือกลับ (return) ไปยังเพจอื่น ๆ หรือมีการกระโดดไปยังเพจอื่น หรือมีการวิ่งไปจนหมดเพจปัจจุบันแล้วต้องการเพจถัดไป การย้ายการควบคุมคำสั่งไปในเพจใหม่นี้เรียกว่า " page fault " ถ้า เพจที่ต้องการไม่ได้อยู่ใน " working set " ปัจจุบันของกระบวนการ

- working set ของกระบวนการ คือ เซตของเพจที่สามารถมองเห็นได้ใน RAM
- สามารถแทนที่โค้ดเพจได้โดยไม่ต้องเขียนกลับสู่ดิสก์
- ความแตกต่างระหว่างเพจข้อมูล กับโค้ดเพจ คือ เพจข้อมูลจะถูกเปลี่ยนโดยกระบวนการที่มาเข้าถึง บ่อยกว่าโค้ดเพจ ดังนั้นตัวจัดการหน่วยความจำ จึงต้องระมัดระวังในการเขียนมันลงบนอุปกรณ์ต่อพ่วงก่อนที่จะแทนที่ด้วยเพจอื่น

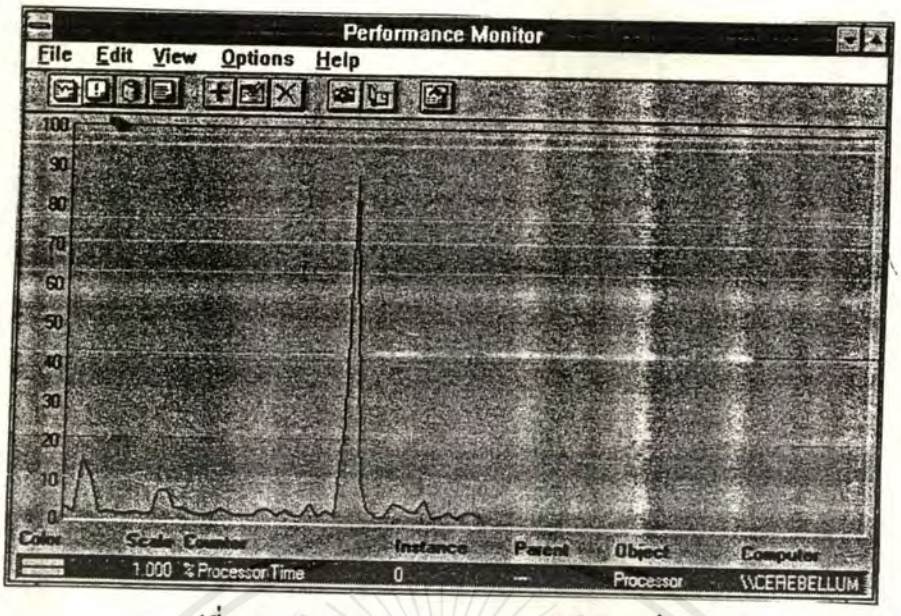


รูปที่ 4-1 การทำ page fault บน Windows NT

4.2 ตัวอย่างของกิจกรรมของหน่วยความจำและการเพจจิ้ง

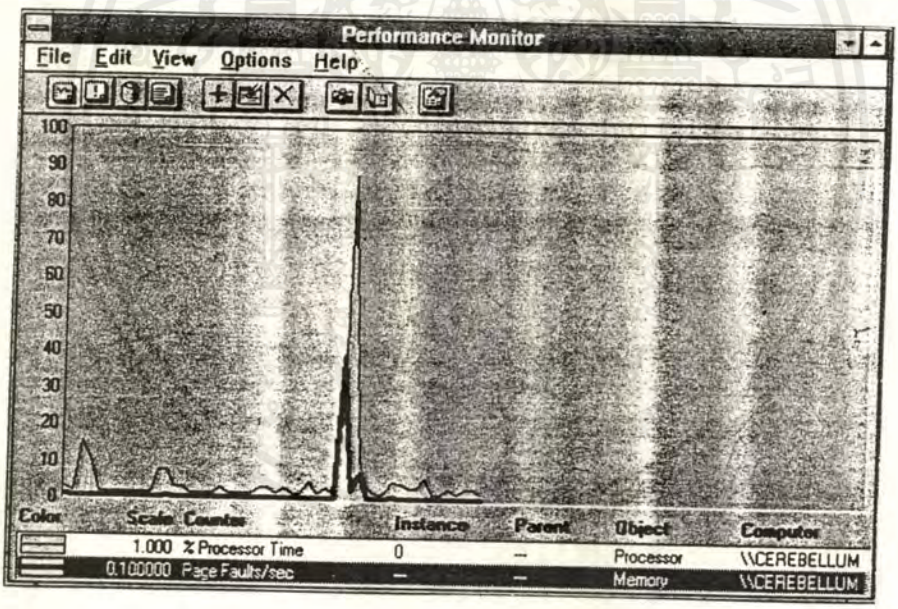
ดูตัวอย่างคลือกซึ่งเป็นแอคเซสซอรี่ (accessory) บน Windows NT และดูชนิดของกิจกรรมของหน่วยความจำที่เกิดขึ้นเวลาในการเริ่มต้นแอปพลิเคชั่นต่าง ๆ นั้นจะรวดเร็วมาก ดังนั้นจึงทำการล็อคเพื่อดูเหตุการณ์ที่เกิดขึ้นภายใน 1 วินาทีเท่านั้น โดยเลือกคลือกจากกลุ่มแอคเซสซอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-2 กิจกรรมของโพรเซสเซอร์ในขณะเริ่มต้นคล็อก

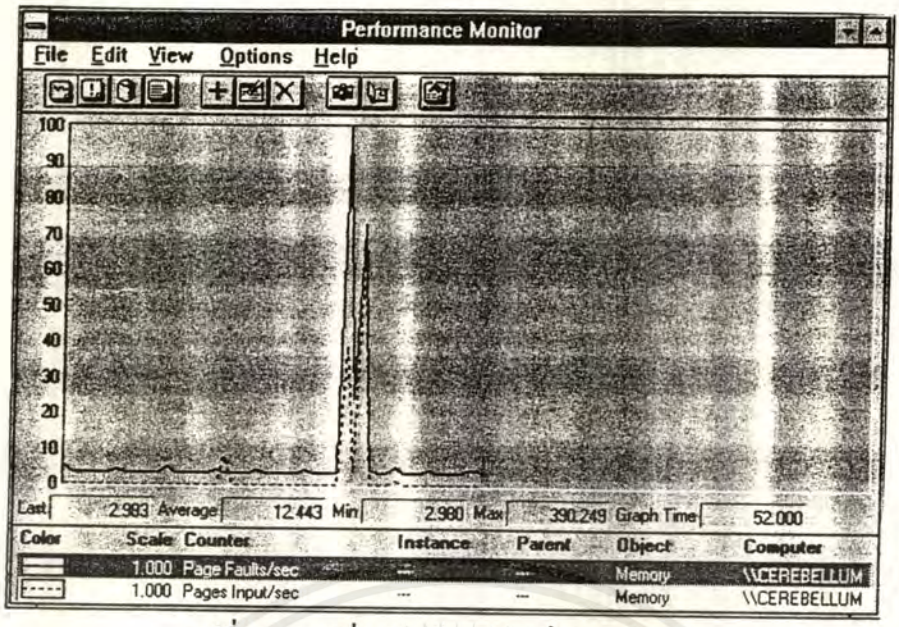
แล้วเพิ่ม Memory : Page Faults/sec ดังรูปที่ 4-3 ซึ่งเป็นเส้นหนาสีดำมี page fault จำนวน 2 ครั้ง โดยครั้งแรกจะมากกว่าครั้งที่สอง



รูปที่ 4-3 การเกิด page fault ในขณะเริ่มต้นคล็อก

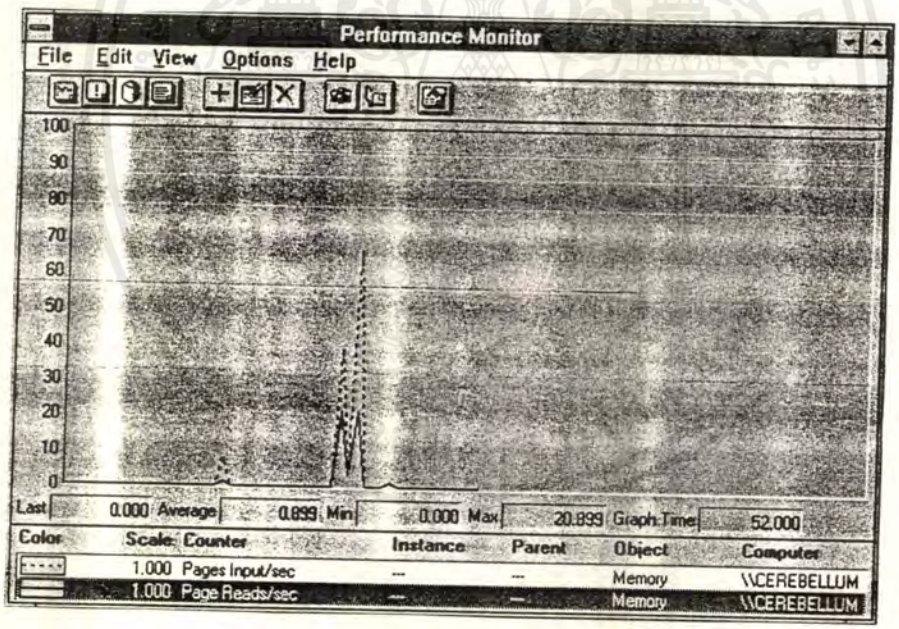
การเกิด page faults อาจจะไม่ได้หมายถึงการติดต่อกับอุปกรณ์ต่อพ่วงก็ได้ ดังนั้นต้องดูว่าทุกครั้งที่เกิด page fault ขึ้นนั้นเป็นการอ่าน เพจที่มาจากดิสก์จริง ๆ หรือไม่ จากรูปที่ 4-4 เส้นทึบสีดำแสดง Page Faults / sec และเส้นประแสดง Pages Input / sec ในพีค (peak) แรกนั้น มีการเกิด page fault จำนวนมากที่ไม่ได้มาจากดิสก์จริง ๆ แต่อย่างไรก็ตามในพีคที่สองนั้น ทุก ๆ การเกิด page fault จะเกิดจากการอ่านเพจใหม่จากดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-4 เพจที่ถูกนำเข้ามาขณะเริ่มต้นคล็อก

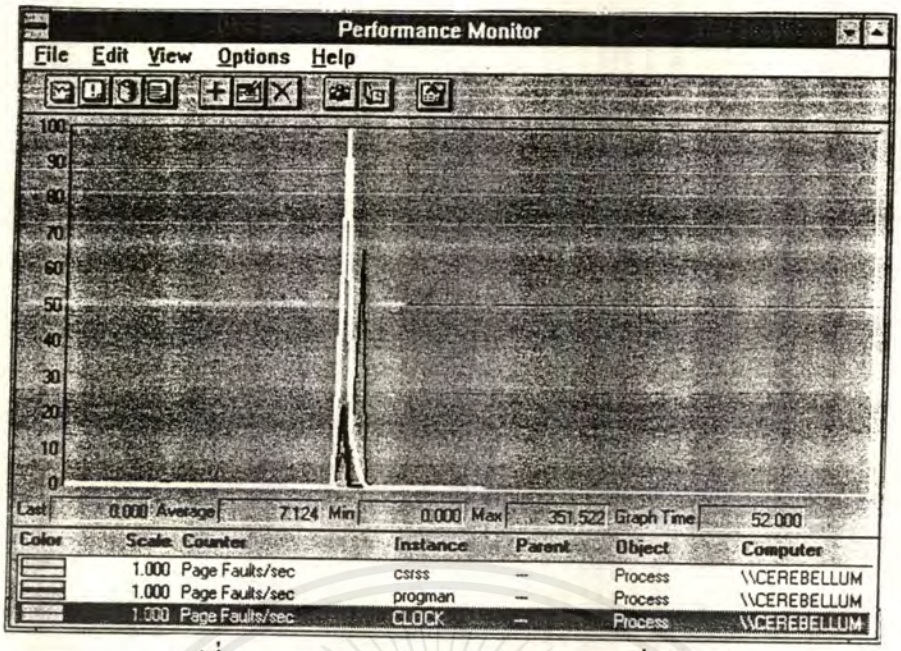
ในรูปถัดไปเราสามารถดูจำนวนครั้งต่อวินาทีที่ตัวจัดการหน่วยความจำร้องขอเพจใหม่จากดิสก์ไดรเวอร์ (disk driver) ได้ ซึ่งจะเห็นว่าจำนวนครั้งนี้มีค่าน้อยกว่าจำนวนเพจที่เข้ามา แสดงว่าตัวจัดการหน่วยความจำทำการร้องขอเพจจากดิสก์ไดรเวอร์ครั้งละหลาย ๆ เพจ



รูปที่ 4-5 การอ่านเพจในขณะเริ่มต้นคล็อก

อาจสรุปได้ว่าคล็อกเป็นเหตุให้เกิดทุก page fault ข้างต้น แต่เราก็ควรจะตรวจสอบให้แน่ใจด้วยโดยไปดูที่รายงานแล้วดู page fault ที่เกิดโดยทุก ๆ กระบวนการพบว่า มี 3 กระบวนการที่เกิด page fault ในเวลานี้ คือ : ตัวจัดการโปรแกรม (Program manager), คล็อกและ CSRSS จากรูปที่ 4-6 เส้นของจำนวน page fault ของคล็อกจะมีแถบสว่างอยู่, เส้นหนาสีดำจะแสดงจำนวน page fault ของตัวจัดการโปรแกรม และเส้นบางสีดำจะแสดงจำนวน page fault ของ CSRSS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



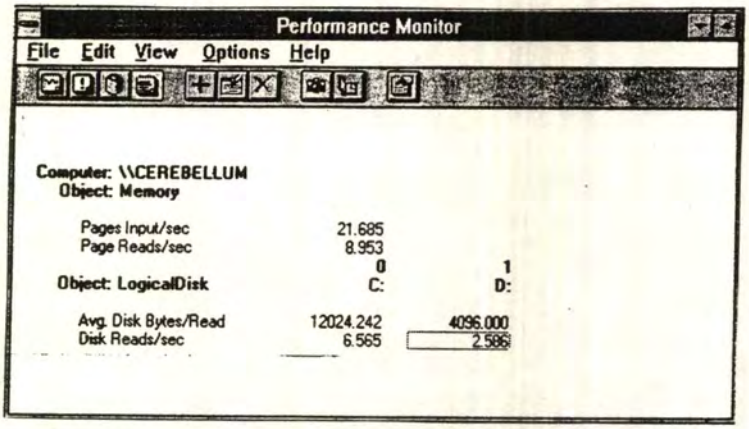
รูปที่ 4-6 page fault โดยกระบวนการขณะเริ่มต้นคลอก

จากรูปที่ 4-6 เส้นของ CSRSS ลูกแรกทางด้านซ้ายมือมีจำนวน page fault ที่น้อยมากเมื่อเทียบกับจำนวน page fault ของคลอก แต่ถ้าเราเน้นดูไปที่ตัวจัดการโปรแกรม ก็จะสามารถมองเห็นจำนวน page fault ของ CSRSS ด้านซ้ายนี้ได้ ดังนั้น ณ. จุดนี้ไม่ใช่ทุกๆ เพจใน working set ของ CSRSS ที่ถูกต้องในการเริ่มต้นคลอก และจากรูปจะเห็นได้ว่าในพีคแรกนั้น จำนวน page fault ที่เกิดขึ้นนั้น จะเกิดจากคลอกมากกว่าจากตัวจัดการโปรแกรม เพราะฉะนั้นในช่วงเวลานี้จะเห็นว่า เพจ ส่วนใหญ่จะอยู่ในหน่วยความจำอยู่แล้ว ซึ่งไม่ต้องไปอ่านเพจจากดิสก์เลย แสดงว่าคลอกใช้กระบวนการของระบบวินโดว์และกราฟฟิก (graphics) ซึ่งกระบวนการเหล่านี้ได้ถูกใช้โดยกระบวนการอื่น ๆ อยู่แล้ว

สำหรับ CSRSS จะเกิด page fault น้อยมาก ในช่วงพีคแรก แต่จำนวน page fault จะเพิ่มขึ้นมากในพีคที่สอง เนื่องจากเราทราบว่าจำนวนของ page fault และเพจที่นำเข้ามาในพีคที่สองนั้นมีจำนวนที่ใกล้เคียงกัน จึงสามารถสรุปได้ว่าเพจที่ CSRSS ต้องการในการเริ่มต้นคลอกนั้น ไม่ได้อยู่ในหน่วยความจำ เพราะคลอกใช้ฟอนต์ (font) ขนาดใหญ่ในการแสดงตัวเลข และฟอนต์ขนาดใหญ่นี้ไม่ได้อยู่ในหน่วยความจำ

ต่อไปจะดูกิจกรรมของดิสก์บ้าง ที่เกิดจากกิจกรรมเหล่านี้ เราจะพิจารณาในจุดที่แคบลง คือ พิจารณาช่วงของ active paging และดูสถิติบางอย่างของหน่วยความจำและดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-7 รายงานสถิติของหน่วยความจำและดิสก์ในขณะเริ่มต้นคล็อก

จากรูปที่ 4-7 โดยการเพิ่มค่าของ LogicalDisk : Disk Read / sec จากทั้ง 2 ไดร์ฟ เมื่อรวมค่าจาก 2 ไดร์ฟเข้าด้วยกัน (6.565 + 2.586 ~ 9) จะมีค่ามากกว่าค่าของ Memory : Page Reads / sec เพียงเล็กน้อย (8.953) บนไดร์ฟ C อ่านได้เกือบ 3 เพจ (12 K) ต่อการร้องขอการอ่าน 1 ครั้ง

เพราะฉะนั้น ไดร์ฟ C ~ (6 Read / sec) x (3 pages / read)

~ 18 + pages / sec

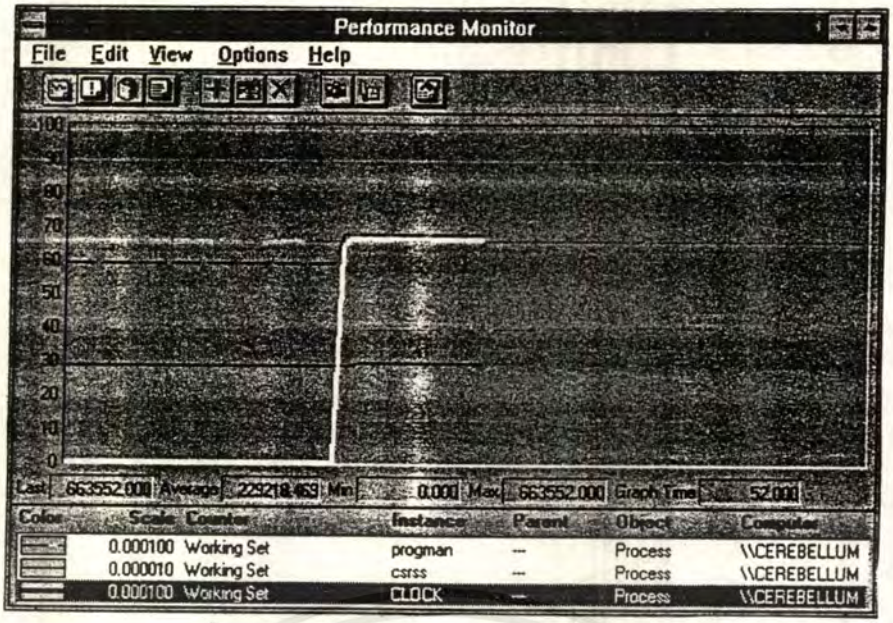
ไดร์ฟ D ~ (2 Reads / sec) x (1 pages / read)

~ 2 + pages / sec

เพราะฉะนั้น Pages Input / sec ของดิสก์ (ไดร์ฟ C, D) ~ (18+) + (2+) ~ 21 + pages / sec ดังนั้นสถิติการเพจจิ้งของดิสก์และหน่วยความจำจึงสัมพันธ์กันและมีค่าใกล้เคียงกัน ทำให้สามารถตรวจจับปัญหาคอขวดได้จากกฎข้อที่ 9 ในที่นี้เรากำหนดให้เวลาที่ window ไม่ทำงานเท่ากับ 5 วินาที ดังนั้นจึงมีทั้งหมด 105 เพจ (21 x 5)

ต่อไปจะเป็นการดูขนาดของ working set ของกระบวนการต่าง ๆ ถ้าเกิด page fault เป็นจำนวนมาก กระบวนการเหล่านี้อาจเพิ่ม working set ของมัน รูปที่ 4-8 แสดงขนาดของ working set ของแต่ละกระบวนการที่ทำให้เกิด page fault

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 ขนาดของ working set เพิ่มขึ้นเมื่อเริ่มต้นคลีอก

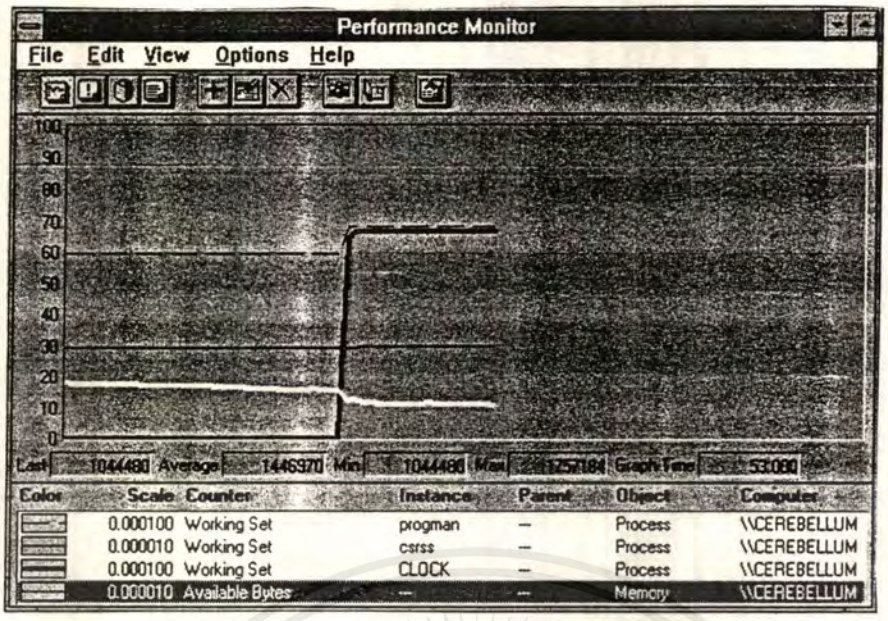
จากรูปที่ 4-8 ให้ระมัดระวังหน่วยของสเกล (scale) ของชาร์ตนี้ ทั้งตัวจัดการโปรแกรมและคลีอก มีขนาด working set เพิ่มขึ้นถึงประมาณ 600 K (60 / 0.0001) แต่ CSRSS เพิ่มขึ้นถึงเกือบ 3 MB (30 / 0.00001) ทำให้มั่นใจได้ว่าขนาดของ working set จะเพิ่มขึ้นตามสมมุติฐานที่ตั้งไว้ คือ เพิ่มตามจำนวน page fault ที่เกิดมากขึ้นในตอนเริ่มต้นนั้นตัวจัดการโปรแกรม จะมีจำนวนเพจ ที่อยู่ในหน่วยความจำอยู่พอสมควร และสิ่งแรกที่เกิดขึ้น คือ ตัวจัดการโปรแกรมจะนำเพจเข้ามาใน working set ที่เพิ่มขึ้น คือ 84 K (21 เพจ) ซึ่งสามารถคำนวณได้โดยลบค่าที่น้อยที่สุดออกจากค่าที่มากที่สุดบน value bar เมื่อเลือกตัวจัดการโปรแกรม

คลีอกซึ่งเริ่มต้นที่ขนาดศูนย์ จะนำเพจเข้ามาใน working set ของมัน ซึ่งเพจเหล่านี้ส่วนใหญ่จะอยู่ในหน่วยความจำแล้ว เพียงแค่เพิ่มลงไปใน working set ของคลีอกเท่านั้น เพื่อที่คลีอกจะสามารถแบ่งกันใช้เพจได้ หรืออาจเป็นเพจของข้อมูลใหม่ที่คลีอกต้องการ แต่ไม่จำเป็นต้องมาจาก ดิสก์

เมื่อคลีอกเริ่มใช้ CSRSS ในการวาดตัวเลขขนาดใหญ่บนหน้าปัดนาฬิกา CSRSS ก็จะเริ่มนำเพจเข้ามาใน working set จากรูปอาจดูเหมือนว่า CSRSS ไม่เพิ่มขึ้นมากนัก แต่จริง ๆ แล้วหน่วยของสเกลของ CSRSS จะมีค่าน้อยกว่า 10 เท่า คือ ขนาดของ CSRSS จะเพิ่มขึ้น 88 K หรือ 22 เพจ จากรูปที่ 4-4 การเกิด page fault ส่วนใหญ่ของ CSRSS จะมาจากเพจจริง ๆ จาก ดิสก์ เนื่องจากเราทราบว่าจะเกิด page fault รวมทั้งหมด 105 เพจ และทราบว่าเป็นของ CSRSS 22 เพจ, เป็นของตัวจัดการโปรแกรม 21 เพจ จึงทำให้ทราบได้ว่าเกิดจากคลีอกเองเป็นจำนวน 62 เพจ

กระบวนการเหล่านี้ต้องการพื้นที่นี้จริง ๆ หรือไม่ ? อาจจะไม่ในช่วงเวลาหนึ่งก็ได้ ซึ่งตัวจัดการหน่วยความจำจะปล่อยให้ working set ของกระบวนการเหล่านี้เพิ่มขึ้นเรื่อย ๆ จนกระทั่งเกิด แรงกดดันกับหน่วยความจำขึ้น ซึ่งสังเกตได้จากการลดของเคาน์เตอร์ที่สำคัญตัวหนึ่ง คือ Memory : Available Bytes ในรูปถัดไปเราจะเพิ่ม Available Bytes ลงในชาร์ตข้างต้น เพื่อดูว่ามีหน่วยความจำที่ว่างอยู่เท่าใด ทั้งก่อนและหลังการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



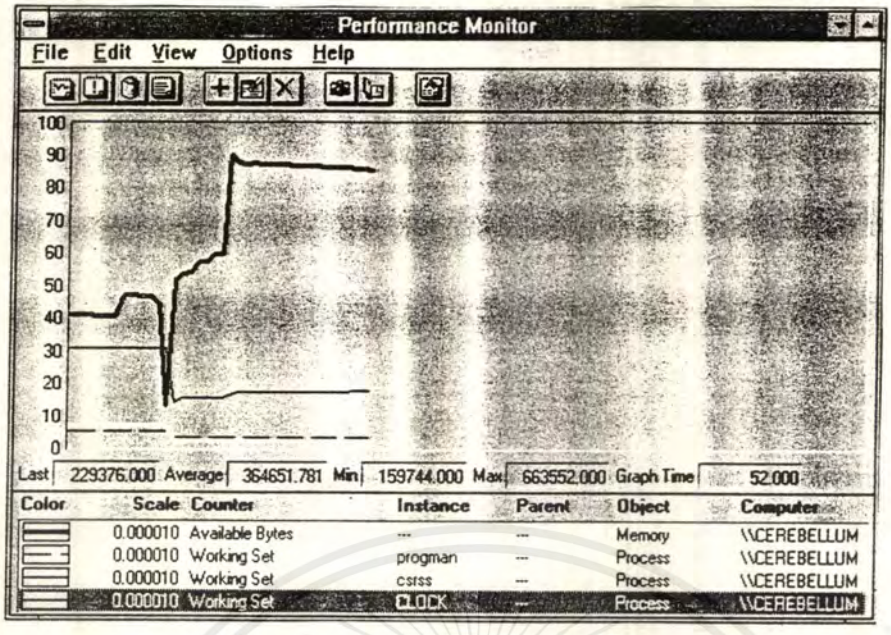
รูปที่ 4-9 Available bytes ลดลงเมื่อเริ่มต้นคล็อก

จาก value bar พบว่า Memory : Available Bytes เริ่มต้นที่ 1.7 MB และสิ้นสุดที่ประมาณ 1 MB เมื่อ Memory : Available Bytes น้อยลงมาก ๆ ตัวจัดการหน่วยความจำก็จะนำเพจ มาจาก working set ของกระบวนการที่ไม่แอคทีฟ (inactive) แล้ว หรืออาจมีทางเลือกอื่น คือ แทนที่จะให้ working set ของเพจเพิ่มขึ้นเรื่อย ๆ และใช้หน่วยความจำที่เหลือ มันก็จะนำเพจมาจากส่วนอื่นของ working set ของกระบวนการนั้น

สิ่งนี้เปลี่ยนจากการแทนที่เพจแบบโกลบอล (global) เป็นโลคอล (local) แทน เมื่อมีเนื้อที่เพียงพอแล้ว ตัวจัดการหน่วยความจำก็จะเปลี่ยนเป็นแบบโกลบอลอีก ในบางขอบเขตเราสามารถที่จะปรับได้เมื่อเกิดการเปลี่ยนแปลงเช่นนี้ขึ้น โดยเลือก Network option ใน Control Panel แล้วเลือก Server ใน Installation Network Software List แล้วเลือก configure ซึ่งสามารถเลือก option ได้มากมาย แต่สำหรับระบบโดยปกติแล้วให้ใช้ Balanced option เราทำการทดลองนี้บนระบบเซิร์ฟเวอร์ที่ถูกปรับเป็นให้มี Throughput มากที่สุดสำหรับแบ่งใช้ไฟล์

เมื่อแรงกดดันที่เกิดขึ้นกับหน่วยความจำเป็นสิ่งสำคัญ และขนาดของ working set เพิ่มมากขึ้น และคุณไม่สามารถบอกจากการดูมันได้ว่าต้องการเนื้อที่จำนวนเท่าไร แต่คุณสามารถสร้างแรงกดดันสำหรับหน่วยความจำได้ และเราสามารถทำหน่วยความจำว่างให้มีประสิทธิภาพมากขึ้น โดยใช้ ยูทิลิตี้ (utility) ที่เรียกว่า “clearmem “ โดย clearmem นี้จะกำหนดขนาดของ RAM และใส่ข้อมูลอย่างเพียงพอลงในนั้น แล้วอ้างถึงข้อมูลนี้ให้เร็วที่สุดเท่าที่จะเร็วได้ ซึ่งจะเป็นการนำเพจอื่น ๆ ที่ไม่เกี่ยวข้องออกจากหน่วยความจำด้วย และมันจะเข้าถึงไฟล์โดยการเคลียร์ (clear) แคชด้วย ต่อไปให้ วิ่ง clearmen บนระบบหลังจากการเริ่มต้นคล็อก และสังเกตขนาดของ working set หลังจากที่นำ เพจที่ไม่ใช้ออกไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



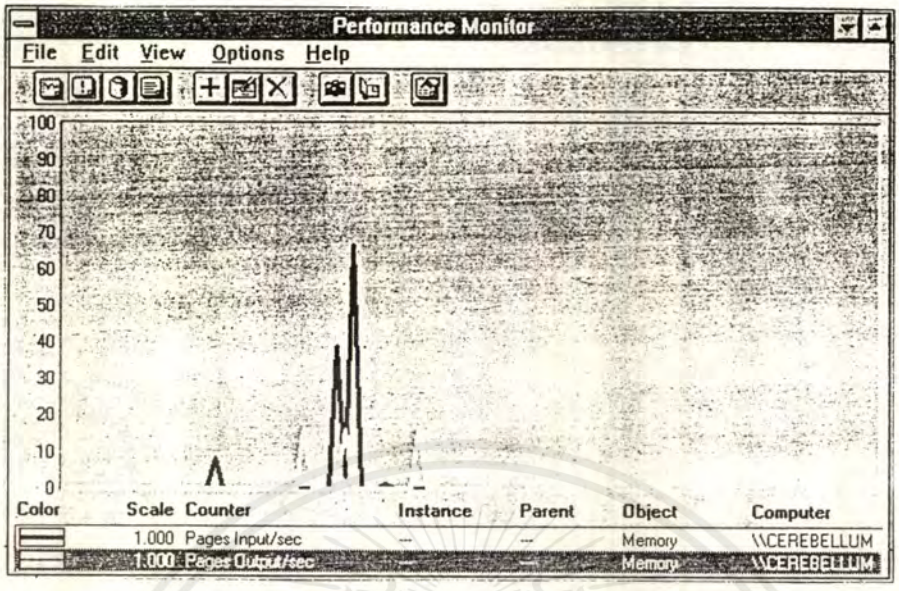
รูปที่ 4-10 working set ลดลงโดยยูทิลิตี้ clearmem

จากรูปจะเห็นว่า Available Bytes ซึ่งเป็นเส้นหนาสีดำนั้น โดยเป็นผลจาก clearmem เส้นคล็อกซึ่งเป็นแถบสว่าง ลดลงจาก 648 K เป็น 224 K ซึ่งแตกต่างกันมาก ตัวจัดการโปรแกรมก็เช่นเดียวกัน ส่วน CSRSS นั้นลดลงประมาณครึ่งหนึ่ง คือ 1.7 MB

ดังนั้นเราจะเห็นได้ว่า working set มีขนาดใหญ่เกินกว่าความต้องการในการวิ่งคล็อก (และ Performance Monitor) ซึ่งถือว่ามีประสิทธิภาพ ตัวจัดการหน่วยความจำทำให้เกิดความสมดุลระหว่างการใช้วัฏจักรของโพรเซสเซอร์ในการปรับแต่ง working set, และไม่ใช้วัฏจักรเหล่านั้นเมื่อไม่จำเป็น สามารถเห็นจากรูปได้ว่า เมื่อมีความต้องการใช้หน่วยความจำจะทำให้กระบวนการในการปรับแต่ง working set มีความเร็วสูง

ตอนนี้กลับไปดูตอนเริ่มต้นคล็อก ได้มีการนำเพจเข้ามามากมาย ซึ่งดูเหมือนว่าเพจที่เข้ามานั้นเข้ามาอยู่ในเพจเฟรมที่ว่าง เพราะ Available Bytes ลดลง แต่เราดูเพียงแค่เพจที่เข้ามาในหน่วยความจำเท่านั้น ไม่ดูไปถึงเพจที่ตัวจัดการหน่วยความจำนำออกไปด้วย รูปที่ 4-11 การจราจรของเพจ คือ การเข้าและออกในขณะเริ่มต้นคล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-11 เพจที่มีการนำเข้ามาและส่งออกไปในระหว่างการเริ่มต้นคล็อก

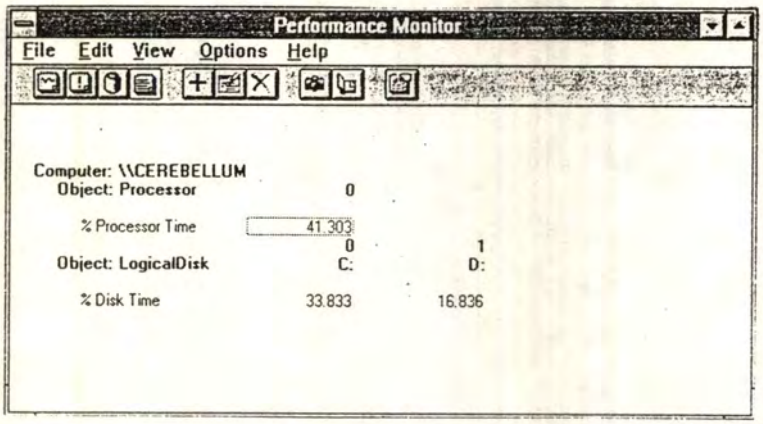
จากรูปเพจที่นำเข้ามาเป็นเส้นสีดำ และเพจที่ถูกเขียนออกไปนั้นเป็นสีขาว ดูเหมือนว่าบาง เพจจะถูกเขียนเพื่อตอบสนองการเริ่มต้นคล็อก ดังเช่นเส้นสีขาว ลูกกลางของรูปภาพ แต่จะมีบางเพจ ที่ถูกเขียนก่อนและหลังการเริ่มคล็อก ด้วยเพจที่ถูกเปลี่ยนแปลงใน RAM แต่ยังไม่ได้ถูกเปลี่ยนแปลง ที่อุปกรณ์ต่อพ่วง เรียกว่า " dirty page " เมื่อเพจที่ถูกเปลี่ยนแปลงไปนั้นถูกนำออกจาก working set เมื่อมีแรงกดดันกับหน่วยความจำไม่มากนัก ตัวจัดการหน่วยความจำอาจจะไม่เขียนมันกลับสู่ดิสก์ ทันที แต่จะนำเข้าไว้ในรายการของเพจที่ถูกเปลี่ยนแปลงแล้ว (modified page list) แทน ซึ่งจะมีตัวเขียนเพจที่ถูกเปลี่ยนแปลง (modified page writer) คอยตรวจสอบรายการของเพจที่ถูกเปลี่ยนแปลงแล้วเป็นระยะ ๆ และทำการเขียนบางเพจออกไป เพื่อให้พื้นที่ว่าง โดยวิธีนี้ทำให้ไม่ต้องเขียนเพจ กลับสู่ดิสก์เป็นจำนวนมากเพื่อความรวดเร็วในการนำเพจนั้นกลับมาใช้ใหม่อีก เมื่อพื้นที่เหลือน้อยตัวเขียนเพจที่ถูกเปลี่ยนแปลงก็จะถูกกระตุ้นอยู่บ่อย ๆ

ในการดูการจราจรของเพจทั้ง 2 ทิศทางนั้นใช้ Memory : Pages / sec เคาน์เตอร์นี้จะระบุการเกิดเพจจิ้งในทุกระดับ และที่สำคัญอย่างยิ่งที่จะดูทั้งเพจที่เข้ามาและเพจที่ออกไป แม้ว่าการเกิด page fault นั้นจะเกิดเฉพาะเพจที่เข้าก็ตาม เพราะในกระบวนการหนึ่งของระบบ อาจจะมี dirty page อยู่มาก (clearmem ทำงาน) และทำให้ตัวจัดการหน่วยความจำต้องปรับเปลี่ยน working set ของ กระบวนการอื่น ๆ อีกหลายกระบวนการ

4.3 การเพจจิ้งที่มีหลายกระบวนการ

ในรูปที่ 4-12 ได้ลดเวลาที่วินโดว์จะเริ่มต้นคล็อก ให้เหลือ 3 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



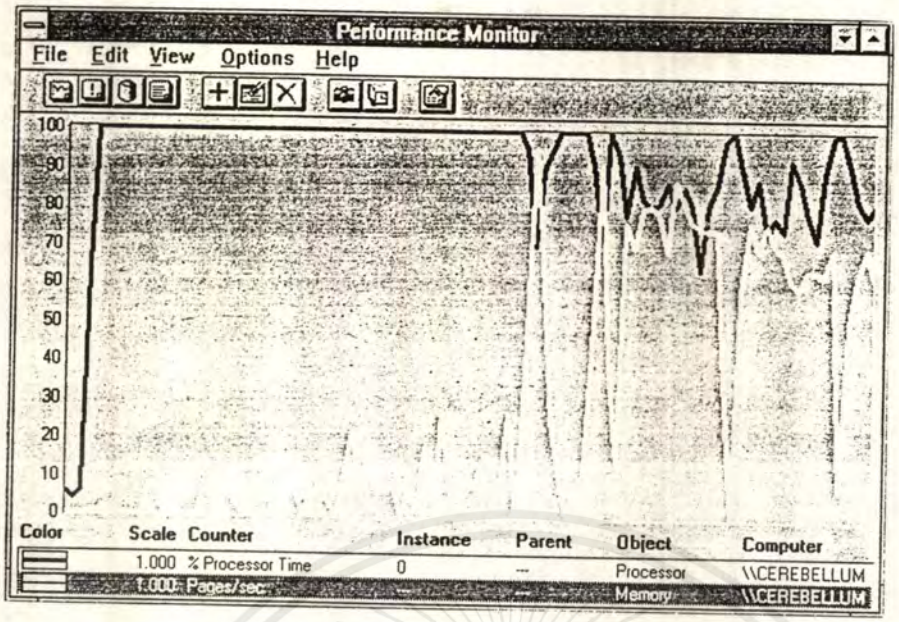
รูปที่ 4-12 ปัญหาคอขวดที่มีผลกระทบต่อกันในระหว่างการเริ่มต้นคล็อก

เมื่อรวมกิจกรรมของดิสก์ของทั้งสองไดร์ฟ (C,D) เข้าด้วยกันแล้วจะได้ 50.669 % (33.883 + 16.836) ในขณะที่ของโพรเซสเซอร์เป็น 41.303 % แสดงว่าดิสก์เป็นจุดคอขวด ระบบนี้ค่อนข้างสมดุล และถ้าเราให้ระบบของดิสก์มีความเร็วเพิ่มขึ้น ก็จะทำให้โพรเซสเซอร์เป็นจุดคอขวด เราใช้เวลาของดิสก์ = 3 x 0.50669 = 1.520 วินาที และเวลาของโพรเซสเซอร์ = 3 x 0.41303 = 1.239 วินาที เวลาที่ใช้ไปจริง ๆ ในการเริ่มต้นคล็อก คือ ผลรวมของทั้งสองจำนวนนี้ คือ 2.759 วินาที ซึ่งเวลาของดิสก์จะมากกว่าครึ่งหนึ่งของจำนวนนี้เล็กน้อย

การเพจจิ้งทำให้ระบบใช้หน่วยความจำอย่างมีเหตุผล, ช่วยโปรแกรมเมอร์ในงานจัดการกับหน่วยความจำ ซึ่งถ้าจัดการกับงานนี้ไม่ดี อาจทำให้โปรแกรมผิดพลาดหรือเกิดความไม่น่าเชื่อถือขึ้นได้ แต่ก็มีบางครั้งที่ไม่มีหน่วยความจำเพียงพอสำหรับทุก ๆ เพจที่จำเป็นต้องใช้จึงเกิดปัญหาขึ้น

ต่อไปดูกรณีที่มีปัญหามากขึ้น โดยจะไม่ใช้ Response Probe แต่จะทำการเพิ่มจำนวนกระบวนการเข้าไป และจะตั้งให้แต่ละกระบวนการเขียนลงบนหน่วยความจำ 1 เมกะไบต์ (megabyte)ในการแจกแจงแบบปกติ รูปที่ 4-13 แสดงการใช้โพรเซสเซอร์และการจราจรของเพจในระหว่างการทดลอง

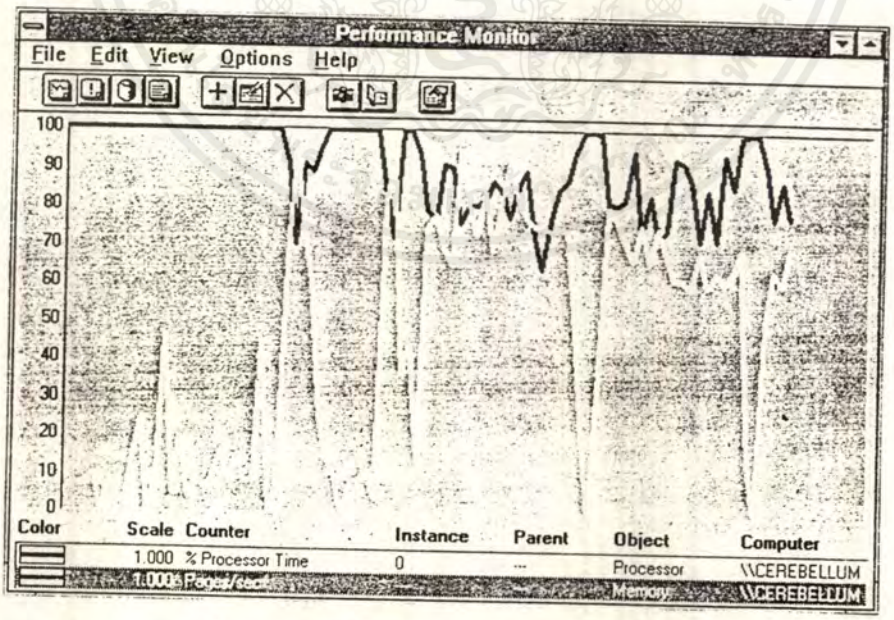
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-13 การใช้โพรเซสเซอร์และการจราจรของเพจภายใต้แรงกดดันของหน่วยความจำที่เพิ่มขึ้น

จะเห็นว่าการใช้โพรเซสเซอร์นั้นมีค่าลดลง และ Memory : Pages/sec มีค่าเพิ่มขึ้น แต่แนวโน้มนี้จะเป็นไปอย่างไม่สม่ำเสมอ เพราะถ้ามีกระบวนการหนึ่งมีจำนวนเพจเพียงพอในหน่วยความจำ ในการทำให้กระบวนการนั้นก้าวหน้าได้ มันจะดึงโพรเซสเซอร์ไว้และทำการเอ็กซีคิวชั่วขณะ จากรูปโดยรวมแล้ว ทางครึ่งซ้ายเป็นระบบที่ดี ส่วนทางครึ่งขวาเป็นระบบที่ยังไม่ดี

ทางด้านขวา จะเห็นเส้นการจราจรของการเพจจิ้งเป็นยอดแหลมตกลงมา ในรูปต่อไป เราจะพิจารณากิจกรรมตรงส่วนนี้เป็นพิเศษ

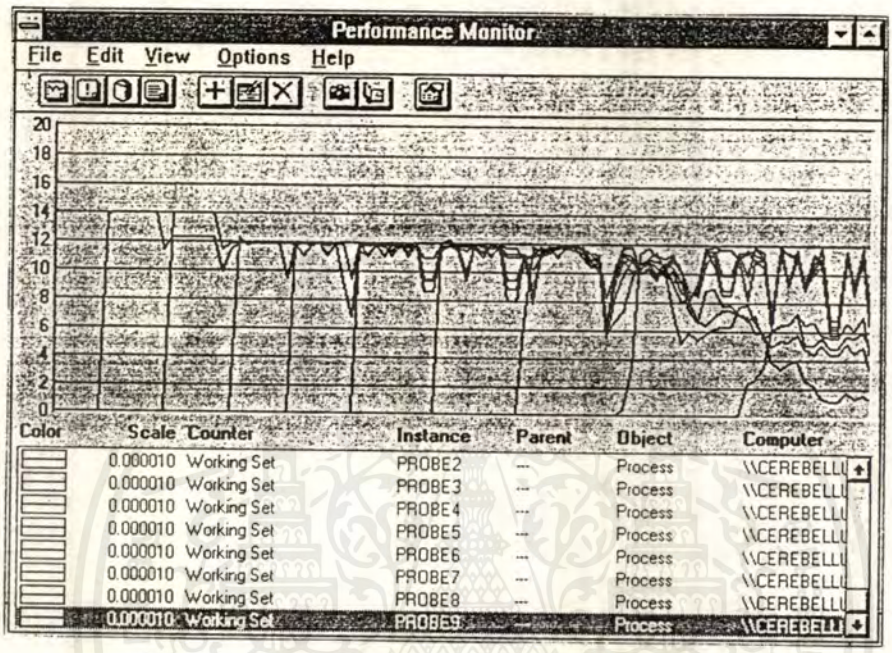


รูปที่ 4-14 การใช้ระบบขณะที่มีอัตราการเพจจิ้งมาก

ในรูปถัดไป แสดง working set ของกระบวนการที่ถูกเพิ่มเข้าไปเพื่อเพิ่มแรงกดดันให้กับระบบ เมื่อมีกระบวนการ ที่ 3 เข้ามาดูเหมือนว่า working set จะลดลง เหลือแต่ที่ต้องการเท่านั้น คือ ประมาณ 1.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MB. ทุกอย่างก็ดูเหมือนว่าจะเป็นไปด้วยดี แต่เมื่อมี กระบวนการ ที่ 8 เข้ามา ทุกอย่างก็ดูเหมือนจะลดลงทางด้านขวาของชาร์ตนั้นบางกระบวนการจะถูกบังคับให้ออกไปแล้วมีบางเพจเข้าแทนที่ ถ้าเราปล่อยให้มีการแย่งชิงกันเช่นนี้ ก็จะทำให้ได้จำนวนที่เท่ากันในแต่ละกระบวนการ แต่ความจริงก็คือ ไม่มีพื้นที่เพียงพอสำหรับทุกกระบวนการ จากรูปก่อนหน้านี้พบว่าค่า Pages/sec เพิ่มขึ้นเป็น 70s ซึ่งมีความเร็วเหมือนกับว่าเครื่องนี้สามารถเอ็กซีคิวการเคลื่อนย้ายเพจได้



รูปที่ 4-15 working set ของ Response Probe เมื่อแรงกดดันของหน่วยความจำเพิ่มขึ้น

4.4 การดูค่าของนอนเพจพูล (Nonpaged Pool)

พูล (Pools) คือ ที่เก็บข้อมูลสำหรับระบบปฏิบัติการและส่วนประกอบของมัน และนอกจากนี้โครงสร้างข้อมูลที่แทนออบเจ็กต์ของระบบที่ถูกสร้างและใช้โดยแอปพลิเคชั่นต่างๆ นั้น ก็อยู่ในพูลด้วย จะสามารถเข้าถึงพูลได้ในพรีวิเลจโหมดเท่านั้น

เพจพูล (paged pool) คือ ที่ที่ระบบใส่ข้อมูลที่สามารถเขียนลงบนดิสก์ได้ ส่วนใน นอนเพจพูลนั้น ไม่สามารถนำเพจออกจากหน่วยความจำได้ การที่จะได้พื้นที่มานั้นโครงสร้างข้อมูลที่ถูกเก็บอยู่จะต้องถูกเข้าถึงได้โดยกระบวนการอินเทอร์รัปต์หรืออยู่ในสปินล็อคคริติคอลเซ็คชั่น (spinlock critical sections) ซึ่งป้องกันการทำงานพร้อมกันของโพรเซสเซอร์หลายตัว, เพจเหล่านี้จึงยังคงอยู่ในหน่วยความจำ เพราะการเกิด page fault ไม่ได้รับอนุญาตให้เกิดในอินเทอร์รัปต์ หรือสปินล็อค

การเพิ่มขึ้นของพื้นที่เพจพูล ที่ไม่สามารถควบคุมได้เป็นจุดบกพร่องชนิดหนึ่งที่เราต้องพิจารณาโดยตรวจสอบขนาดของนอนเพจพูล ซึ่งจะแตกต่างกันไปบ้างเล็กน้อยสำหรับคอมพิวเตอร์แต่ละเครื่องขึ้นอยู่กับการใช้บริการของระบบ เราควรบันทึกขนาดของนอนเพจพูลในตอนเริ่มต้นระบบ แล้วเปรียบเทียบกับค่าปัจจุบัน ซึ่งมันไม่ควรเพิ่มขึ้นในระหว่างการกระทำใด ๆ ในระบบ ถึงแม้ว่าโปรแกรมจะสร้างออบเจ็กต์ใหม่ที่ใช้พื้นที่ของนอนเพจพูลก็ตาม การเพิ่มขึ้นทีละเล็กทีละน้อยของพื้นที่นอนเพจพูลนั้นเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่า " pool leak " ซึ่งไม่เหมือนกับสระน้ำโดยทั่วไป คือ เมื่อเกิดการรั่วจะทำให้มีข้อมูลเข้าไม่ใช่ออก โดยปกติสาเหตุของ pool leak คือ การเปิดไฟล์ของแอปพลิเคชั่นซ้ำกันโดยไม่ได้ตั้งใจ

Memory : Nonpaged Pool Allocs จะแสดงถึงจำนวนรวมของ allocations ปัจจุบันที่มีอยู่ในพูลถ้านำจำนวนไบต์ทั้งหมดของนอนเพจพูล หารด้วยจำนวนรวมของ allocations นี้ จะได้ขนาดเฉลี่ยของ allocation คือ 211 ไบต์ (1204224 / 5696) ถ้าเกิด pool leak ขึ้น ตัวเลขของ allocation จะเพิ่มขึ้น แต่จะรู้ได้ว่าเกิด pool leak ขึ้น ก็ต้องจับตาดูค่านี้นานเป็นชั่วโมง

แต่ละกระบวนการมีเคาน์เตอร์สำหรับ Pool Nonpaged Bytes และ Pool paged Bytes แต่ละค่าเหล่านี้เป็นการประมาณค่าจากตัวจัดการออบเจ็กต์ของระบบ ซึ่งจะเป็นตัวกำหนดพื้นที่ให้ กระบวนการ ในการครอบครองออบเจ็กต์ที่ถูกสร้างขึ้นมา และมีการจัดการกับออบเจ็กต์ ซึ่งเป็นการจัดการที่เป็นการจำลองแบบมา สิ่งสำคัญ คือ ถ้าเกิด pool leak ขึ้น เราสามารถหาได้ว่าเกิดกับ กระบวนการใดโดยดูที่สถิติของกระบวนการของพูล

เพราะฉะนั้นถ้าไม่จำเป็นก็ไม่ควรเพิ่มโปรโตคอลและไดร์ฟเวอร์ ลงไปในคอมพิวเตอร์ เพราะแม้แต่โปรโตคอลที่อยู่เฉย ๆ ยังต้องการใช้พื้นที่พูล

4.5 การแก้ไขปัญหาคอขวดที่เกิดจากหน่วยความจำ

4.5.1 การแก้ปัญหาโดยใช้ฮาร์ดแวร์

- เพิ่มความเร็วของหน่วยความจำ

คุณไม่สามารถเพิ่มความเร็วของหน่วยความจำได้โดยตรง แต่คุณสามารถเพิ่มความเร็วของดิสก์ได้โดยทำเป็นหน่วยความจำเวอร์ชวล (virtual memory) โดยที่ดิสก์สามารถใช้ไดร์ฟและไดร์ฟคอนโทรลเลอร์ที่เร็วและใช้หลาย ๆ ไดร์ฟและกระจายเพจไฟล์ (page file) ไปยังไดร์ฟต่าง ๆ แต่จะต้องไม่วางเพจไฟล์บนสไตร์ฟไดร์ฟเพราะจะทำให้ไม่เป็นการปรับปรุงประสิทธิภาพเลยถ้าทำอย่างนั้น

- เพิ่มหน่วยความจำให้กับระบบ
- ตัดดีไวซ์ไดร์ฟเวอร์ที่ไม่จำเป็นออกไป
- ลงฟิกซ์ดิสก์ (fixed disk) ที่เร็วขึ้นเพื่อลดเวลาในการตอบสนอง (response time) ของการเพจ

4.5.2 การแก้ปัญหาโดยใช้ซอฟต์แวร์

- ลดความต้องการของหน่วยความจำลง

ทางที่ดีที่สุดที่จะปรับปรุงความต้องการของหน่วยความจำของ NT ก็คือการตัดคุณลักษณะบางอย่างที่คุณใช้ออกไปหรือเคลื่อนย้ายแอปพลิเคชั่นตัวที่มีความต้องการหน่วยความจำมีลักษณะดังต่อไปนี้

- RAS
- TCP/IP support
- RAID

สิ่งอื่น ๆ ที่กินหน่วยความจำเป็นจำนวนมากก็คือแอปพลิเคชั่นที่คุณวิ่งบน NT เซิร์ฟเวอร์ก็คือแอปพลิเคชั่นเซิร์ฟเวอร์นั่นเองอย่างเช่น SQL เซิร์ฟเวอร์: ซึ่งต้องการหน่วยความจำ 32 เมกกะไบต์บนเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกสิ่งหนึ่งของหน่วยความจำก็คือการให้บริการต่าง ๆ , NT จะเริ่มทำงานด้วยการให้บริการต่าง ๆ มากมายที่คุณไม่ต้องการและไม่จำเป็น ถ้าเซิร์ฟเวอร์ของคุณไม่มีเครื่องพิมพ์ ดังนั้นในสปูลเลอร์เซอร์วิส (Spooler Service) ก็ไม่มีความจำเป็นที่จะต้องมี ถ้าสตอเร็จดีไวซ์ (storage device) ทั้งหมดเป็น SCSI ให้ตรวจดูถ้า ATDISK เป็นการเชื่อมต่อแบบ IDE ในกรณีทั้งคู่นี้คุณสามารถที่จะปิดตัวใดตัวหนึ่ง และควรจะกำจัดโปรโตคอลที่ไม่จำเป็นออกไปซึ่งมันเป็นตัวที่กินหน่วยความจำ

- หาตำแหน่งของหน่วยความจำที่เกิดเมมโมรีลีค (memory leak)

บางครั้งคุณพบว่าส่วนของหน่วยความจำที่ว่าง ๆ ในระบบของคุณจะลดลงแล้วก็ลดลงเรื่อย ๆ ถึงแม้ว่าคุณจะไม่ได้ทำอะไรทั้งนั้น หรือคุณอาจออกจากระบบในเย็นของวันศุกร์และหยุดพักระบบในวันหยุด และกลับมาอีกทีในวันจันทร์คุณจะพบข้อความบนจอภาพเกี่ยวกับการขาดแคลนหน่วยความจำทั้ง ๆ ที่คุณไม่ได้ทำอะไรในวันหยุดสุดสัปดาห์และก็เป็นไปไม่ได้ที่จะมีใครมาทำงานที่สำนักงานในวันเสาร์และวันอาทิตย์ซึ่งได้หมายความว่าเกิดเมมโมรีลีค

เมมโมรีลีคเกิดจากแอปพลิเคชั่นที่มีข้อบกพร่องในตัวมันแล้วมันก็จะร้องขอหน่วยความจำเพิ่มขึ้นแล้วก็เพิ่มขึ้นเรื่อย ๆ จากระบบปฏิบัติการ ซึ่งมันไม่ได้ทำอะไรกับหน่วยความจำหรือว่ามันอาจจะใช้หน่วยความจำแต่ว่าไม่ได้มีการจัดการที่ดี คุณสามารถที่จะหาตำแหน่งของเมมโมรีลีค โดยการบันทึกออบเจ็กต์ของโพรเซส (logging Process object) แล้วใช้เคาน์เตอร์ของโพรเซสที่ชื่อว่าเวิร์กกิงเซ็ท (working set) เป็นตัววัดของหน่วยความจำที่แอปพลิเคชั่นใช้งานไป แล้วลีคเกอร์ (leaker) ก็จะแสดงตัวออกมาแล้วจึงติดต่อกับบริษัทที่เขียนแอปพลิเคชั่น ให้ทำการปรับปรุงแอปพลิเคชั่นตามข้อมูลที่วัดออกมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การตรวจจับปัญหาคอขวดที่เกิดจากแคช

ในระบบเอ็มเอส-ดอส (MS-DOS) พารามิเตอร์แรกในการปรับปรุงระบบ คือ ขนาดของแคช (cache) ในเอ็มเอสดอสและวินโดว์เวอร์ชั่นปัจจุบันได้ลดความต้องการในการดึงพารามิเตอร์นี้ลง โดยเพิ่มอีก 1 บิตให้กับขนาดของหน่วยความจำในเครื่อง แต่ก็ยังมียูสเซอร์บางคนที่ยังใช้การปรับขนาดของแคชในการเพิ่มประสิทธิภาพให้กับคอมพิวเตอร์

ส่วนแคชของ Windows NT นั้นจะปรับตัวเองโดยอัตโนมัติให้เข้ากับขนาดของหน่วยความจำและแรงกดดันในคอมพิวเตอร์ และมีการควบคุมการปรับเพียงเล็กน้อย

5.1 การทำงานของแคช

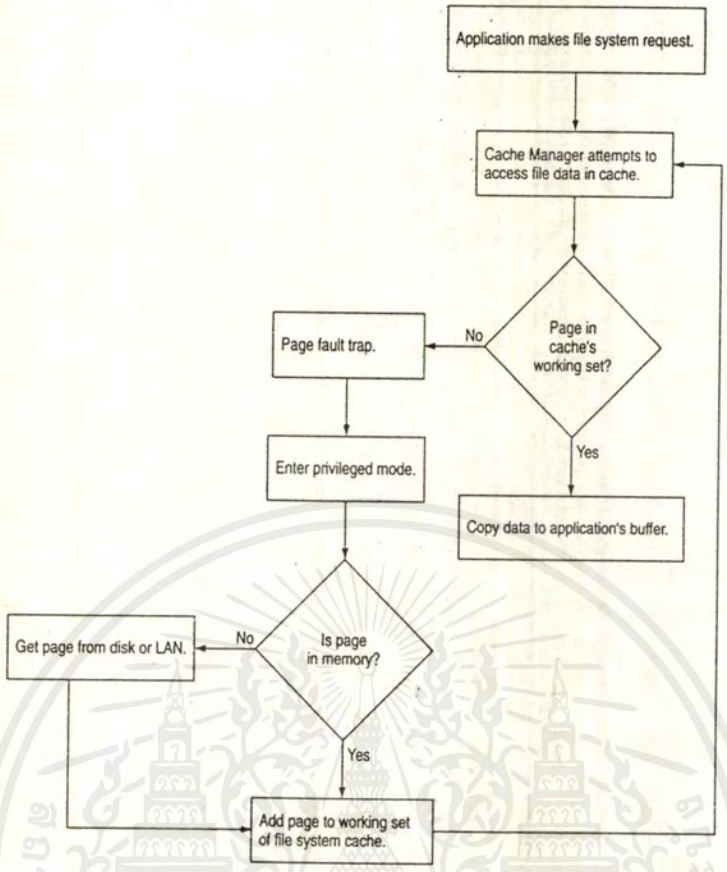
แคช คือ บัฟเฟอร์ที่เก็บข้อมูลที่มาจากหรือไปยังดิสก์, แลน (LANs) และอุปกรณ์ต่อพ่วงอื่น ๆ (เช่น ไดร์ฟซีดีรอม) Windows NT ใช้แคชตัวเดียวสำหรับทุกอุปกรณ์ประกอบอื่น ๆ ที่สามารถเก็บข้อมูลใน แคชได้

ถ้าแอปพลิเคชั่นไม่ได้กำหนดพารามิเตอร์ FILE_FLAG_NO_BUFFERING ในการเรียกของแอปพลิเคชั่น ตอนเปิดไฟล์แคชก็จะถูกใช้ในการเข้าถึงดิสก์ เมื่ออ่านข้อมูลมาจากอุปกรณ์ ข้อมูลนั้นจะถูกเก็บเข้าไว้ในแคชก่อน ส่วนในการเขียนก็เช่นเดียวกันข้อมูลจะถูกเก็บไว้ที่แคชก่อนที่จะถูกเขียนลงดิสก์

เนื่องจากการมีบัฟเฟอร์จะมีส่วนช่วยอย่างมากในการปรับปรุงประสิทธิภาพของระบบ ดังนั้นจึงเป็นเรื่องไม่ปกตินักในการเปิดไฟล์โดยไม่มีบัฟเฟอร์ แอปพลิเคชั่นที่ทำเช่นนี้โดยปกติจะเป็นแอปพลิเคชั่นของเซิร์ฟเวอร์ (เช่น SQL เซิร์ฟเวอร์) ที่จัดการกับบัฟเฟอร์ของตนเอง ในบทนี้เราจะพิจารณาทุกกิจกรรมของไฟล์ที่ถูกเก็บไว้ในบัฟเฟอร์โดย

เมื่อ Windows NT เปิดไฟล์ครั้งแรก แคชก็จะทำการแมป (map) ไฟล์นั้นไปยังพื้นที่ของมัน และสามารถอ่านไฟล์นั้นได้ แม้ว่ามันจะเป็นอาร์เรย์ของเรคคอร์ด (array of records) ในหน่วยความจำ เมื่อแอปพลิเคชั่นร้องขอข้อมูลของไฟล์ ระบบไฟล์ก็จะดูในแคชว่ามีข้อมูลนั้นหรือไม่ แล้วแคชก็จะทำการคัดลอกข้อมูลนั้นไปยังบัฟเฟอร์ของแอปพลิเคชั่น ถ้าเพจของข้อมูลนั้นไม่ได้อยู่ใน working set ของ แคชก็จะเกิด page fault ขึ้น ดังรูปที่ 5-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



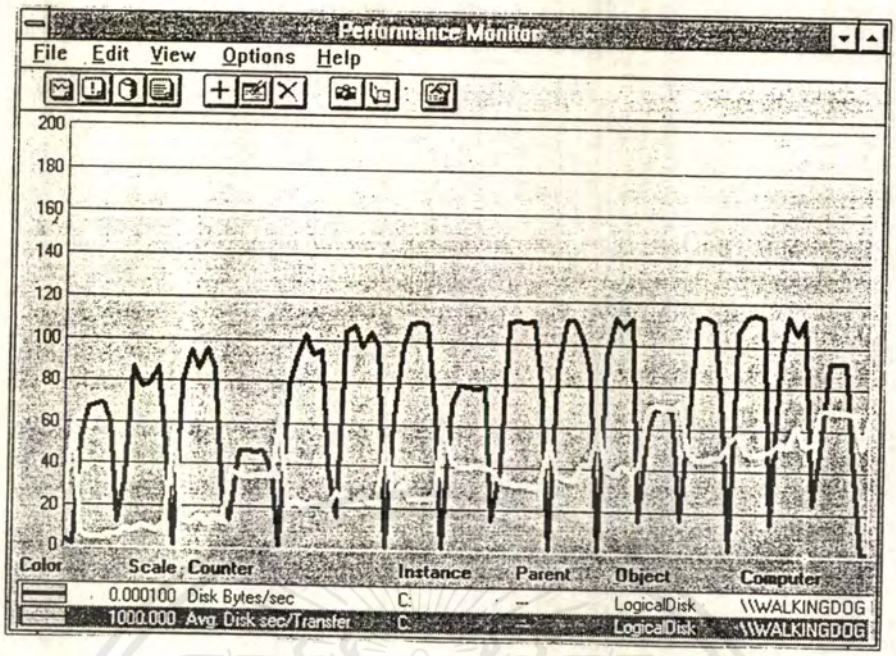
รูปที่ 5-1 การอ้างถึงแคชเมื่อไม่พบเพจที่ต้องการซึ่งจัดการโดยตัวจัดการหน่วยความจำ

ถ้าเพจนั้นอยู่ในหน่วยความจำข้อมูลก็จะถูกแมป (ไม่ใช่คัดลอก) ไปยัง working set ของแคช ถ้าเพจไม่ได้อยู่ในหน่วยความจำ ตัวจัดการหน่วยความจำก็จะนำเพจมาจากไฟล์บนอุปกรณ์ต่อพ่วง จึงสรุปได้ว่าตัวจัดการแคช (cache manager) ใช้ตัวจัดการหน่วยความจำ ในการนำเพจเข้ามาให้กับตน

5.2 การทดลองเบื้องต้นเกี่ยวกับแคช

การทดลองในบทนี้ใช้เครื่อง 386sx/25 มี RAM 25 MB และฮาร์ดไดร์ฟ 120 MB. สิ่งแรกคือดูว่าดิสก์ จะแสดงออกมาอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-2 ประสิทธิภาพของดิสก์ของเครื่องตัวอย่าง 386SX/25

ต่อไปจะดูว่าทำไมเราจึงต้องการแคช โดยทดสอบดังนี้ วิ่ง clearmem เพื่อให้แน่ใจว่าในแคชนั้นไม่มีข้อมูลอยู่ ขั้นแรกจะเป็นการอ่านไฟล์ขนาด 1 MB. แบบซีเควนเชียล (sequential) โดยใช้เรคคอร์ดขนาด 4096 ไบต์ ขั้นที่สองจะเป็นการเขียนไฟล์แบบซีเควนเชียลเช่นกัน ต่อไปเป็นขั้นที่สาม จะอ่านไฟล์แบบสุ่ม และการทดสอบสุดท้ายจะเป็นการเขียนและอ่านไฟล์แบบสุ่ม

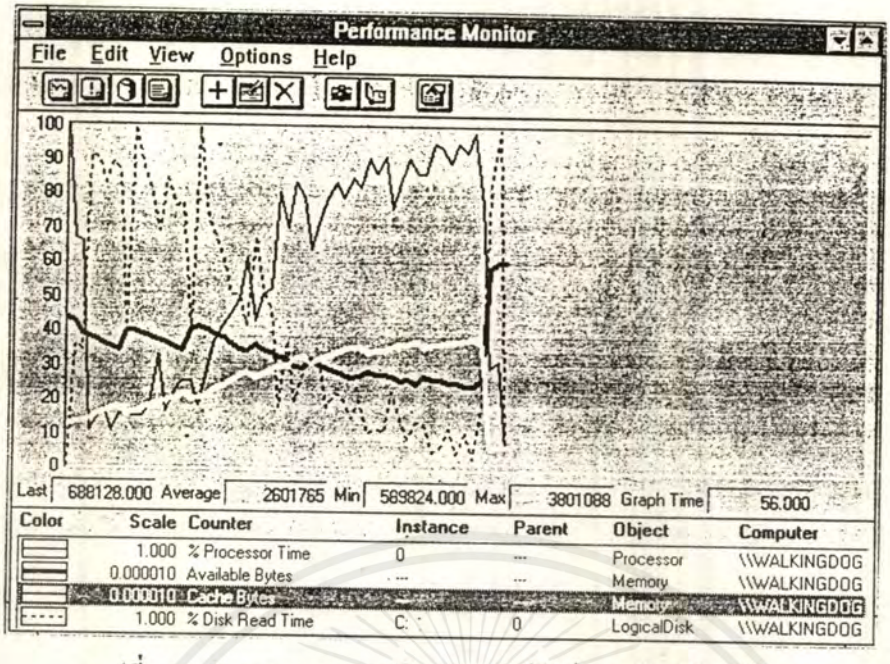
ตารางที่ 5-1 เวลาของไฟล์ I/O แบบใช้แคชและไม่ใช้แคช (หน่วยเป็นมิลลิวินาทีต่อเรคคอร์ด)

Type of file access	Operation	Non-cached time	Cached time
Sequential	Read	6.58	1.32
Sequential	Write	22.91	1.70
Random	Read	20.45	1.51
Random	Read/Write	40.66	3.16

จากตารางจะเห็นว่าการใช้แคชนั้น มีประสิทธิภาพดีกว่า

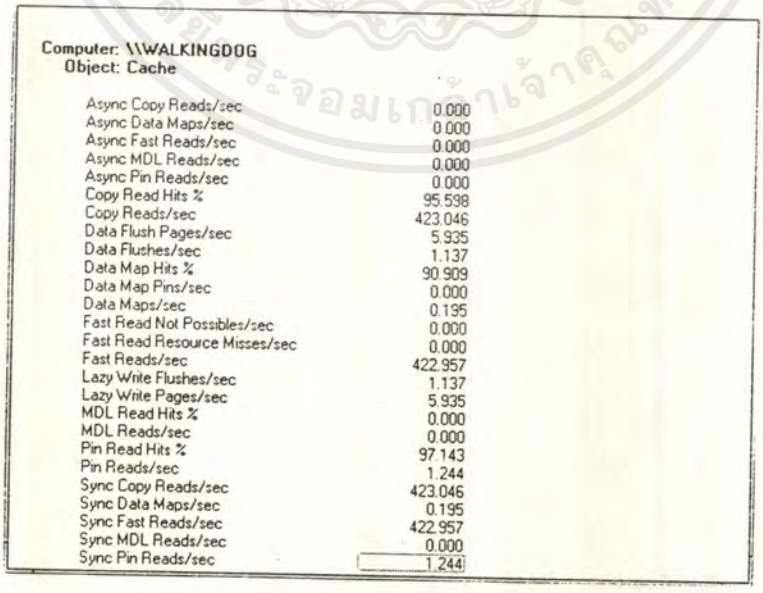
ต่อไปจะสร้างไฟล์ขนาด 10 MB. และจะอ่านเป็นเรคคอร์ดขนาด 8192 ไบต์ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-3 ลักษณะของระบบในขณะอ่านไฟล์ขนาดใหญ่โดยใช้แคช

ในตอนเริ่มต้นผลประโยชน์ของโพรเซสเซอร์จะต่ำ และของดิสก์จะสูง ในขณะที่แคชมีค่าเพิ่มขึ้น เมื่อแคชมีค่าขึ้นถึง 4 MB. เส้นของดิสก์ และโพรเซสเซอร์จะตรงกันข้ามกับตอนเริ่มต้น คือ กิจกรรมของดิสก์จะลดลงและของโพรเซสเซอร์จะเพิ่มขึ้น ส่วนเส้นสีดำเข้มนั้น คือ Memory:Available Bytes เมื่อ Available Bytes มีค่าต่ำกว่า 4 MB. ตัวจัดการหน่วยความจำก็จะทำการปรับแต่ง working set โดยหาจากเพจที่ในปัจจุบันนั้นยังไม่ได้ใช้ การเพิ่มขึ้นของ Available Bytes นั้น เป็นผลจากเพจที่ได้รับการปรับแต่งให้สามารถใช้ได้ แคชก็จะกู้เพจที่ปรับแต่งแล้วกลับคืนมาอย่างรวดเร็ว เพราะ เพจเหล่านี้แอคทีฟอยู่ ทำให้ Available Bytes เพิ่มขึ้นเรื่อย ๆ จนกระทั่งแคชมีขนาดที่เหมาะสม มันจึงจำกัดขนาดของ Available Bytes ให้มีค่าประมาณ 2.75 MB. และระบบก็อยู่ในเสถียรภาพตั้งแต่จุดนี้ไปจนกระทั่งสิ้นสุดการทดลอง



รูปที่ 5-4 สถิติของแคชในขณะอ่านไฟล์ขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

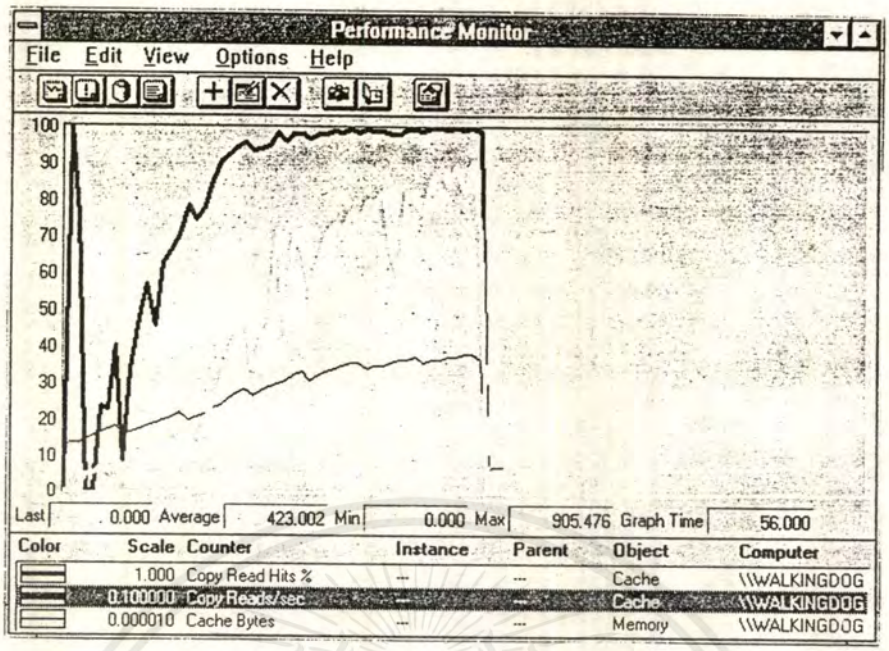
จากรูปเคาน์เตอร์ตัวแรก คือ Copy Read Hits % ซึ่งบอกถึงการคัดลอกข้อมูลจากแคชไปยังบัฟเฟอร์ของแอปพลิเคชั่น การฮิต (hit) นี้จะเกิดขึ้นเมื่อมีการร้องขอจากระบบไฟล์ และข้อมูลนั้นก็ต้องอยู่ในแคชด้วย ซึ่งจากรูปจะพบว่ามีอัตราการฮิตที่สูง และมีจำนวนการเกิดการอ่านต่อวินาทีที่สูงเช่นกัน นอกจากนี้ ยังมีกิจกรรมที่นำข้อมูลออกของแคช ที่ระบุโดยเคาน์เตอร์ดาต้าฟลัช (Data Flush) 2 ตัว ดาต้าฟลัชนี้จะเกิดขึ้นเมื่อตัวจัดการแคช ถูกบอกให้หาพื้นที่ในการเขียนเพจที่ถูกเปลี่ยนแปลงแล้ว ไปยังอุปกรณ์ต่อพ่วง มีตัวบ่งบอกจำนวนหนึ่งที่ก่อให้เกิดดาต้าฟลัชคือ

- เลซี่ไรท์เตอร์ (lazy writer) ของตัวจัดการแคช จะคอยตรวจสอบเพจของแคชที่ถูกเปลี่ยนแปลงเป็นระยะ ๆ ในการฟลัช (flush) ให้กับดิสก์
- แมปเพจไรท์เตอร์ (mapped page writer) ของตัวจัดการหน่วยความจำ สามารถทำให้เกิดการฟลัชข้อมูลได้ ถ้าหน่วยความจำไม่มีพื้นที่เพียงพอ หรือถ้าจำนวนของเพจที่ถูกเปลี่ยนแปลงแล้ว ที่ถูกแมปไปยังแคชนั้นมีขนาดมากเกินไป
- แอปพลิเคชั่นสามารถทำให้ระบบไฟล์ทำการฟลัชแคช สำหรับไฟล์ใดไฟล์หนึ่งโดยเฉพาะได้ ทั้งหมดนี้เรียกว่าการฟลัชข้อมูล ซึ่งจะเป็นตัวทำให้กระบวนการของตัวจัดการหน่วยความจำ mทำการสร้างการร้องขอให้นำข้อมูลออก สำหรับระบบไฟล์ในการเขียนข้อมูลลงบนอุปกรณ์ต่อพ่วง

ต่อไปพิจารณาที่ % ของ Data Map Hits ซึ่งมีค่าสูงมาก ซึ่งดูเหมือนว่าเป็นอัตราการฮิตของแคชที่ดี แต่เมื่อดูที่เคาน์เตอร์ของโอเปอร์เรชั่น (operation) คือ Data Maps/sec จะมีค่าน้อยมาก ดังนั้นจึงเป็นสิ่งสำคัญในการดูเคาน์เตอร์ของโอเปอร์เรชั่นด้วย ในการวิเคราะห์อัตราการฮิต ค่าดาต้าแมป (Data Map) นี้จะถูกใช้ในการแมปในระบบไฟล์ เช่น ไดเรกทอรี่ (directory) , FAT , Fnodes ใน HPFS หรือ Master File Table ใน NTFS ถ้าค่านี้มีค่าสูงก็จะเป็นภาระหนักกับโอเปอร์เรชั่นของไดเรกทอรี่ ซึ่งอาจจะเป็นการคัดลอกไฟล์ขนาดเล็กจำนวนมากก็ได้

เพื่อที่จะเน้นถึงความสำคัญของอัตราการฮิตและความถี่ในการเกิดโอเปอร์เรชั่น ในภาพถัดไปจะแสดงความสัมพันธ์ระหว่าง Copy Read Hits% และ Copy Reads/sec ทางด้านซ้ายของรูปจะมีเส้น Copy Read Hits% สูงเป็นยอดแหลม แต่อัตราของโอเปอร์เรชั่นจะต่ำ ทำให้อัตราฮิตที่สูงนั้นไม่เกิดประโยชน์อะไรขึ้นมา ต่อมาขนาดของแคชเพิ่มมากขึ้นเพื่อที่จะจัดพื้นที่ให้กับไฟล์ ตัวนับทั้งสองตัวนี้ก็จะเพิ่มขึ้นด้วย ซึ่งเป็นผลให้การจราจรของดิสก์ลดน้อยลง และมีผลประโยชน์ของโพรเซสเซอร์เพิ่มขึ้น ดังรูปที่ 5-3 ทางครึ่งขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-5 Copy Read Hits% และ Copy Reads/Sec ในระหว่างการอ่านไฟล์ขนาดใหญ่

การร้องขอของ multiple data list (MDL) เป็นวิธีของระบบไฟล์ ในการส่งข้อมูลจำนวนมากของแคช โดยใช้ Direct Memory Access (DMA) MDL จะให้พื้นที่หน่วยความจำแบบฟิสิคอล สำหรับแต่ละเพจที่จะใช้ในการส่งข้อมูล บางครั้งกระบวนการของ Windows NT เซิร์ฟเวอร์ จะใช้วิธีนี้ในการรับส่งข้อมูลขนาดใหญ่จากแคช

ในโอเปอร์เรชั่น Pin Reads นั้น แคชจะอ่านข้อมูลโดยมีวัตถุประสงค์เพื่อที่จะเขียนข้อมูลไปยังเพจเฉพาะ โดยแคชต้องอ่านเพจทั้งหมดเข้ามาจากอุปกรณ์ต่อพ่วงก่อน แล้วเพจนั้นก็จะถูกฝังไว้ในหน่วยความจำ จนกระทั่งมีการเขียนเกิดขึ้น การฮิตจะเกิดขึ้นเมื่อมีข้อมูลอยู่ในแคช ในเวลาเดียวกับที่มีการร้องขอการอ่าน

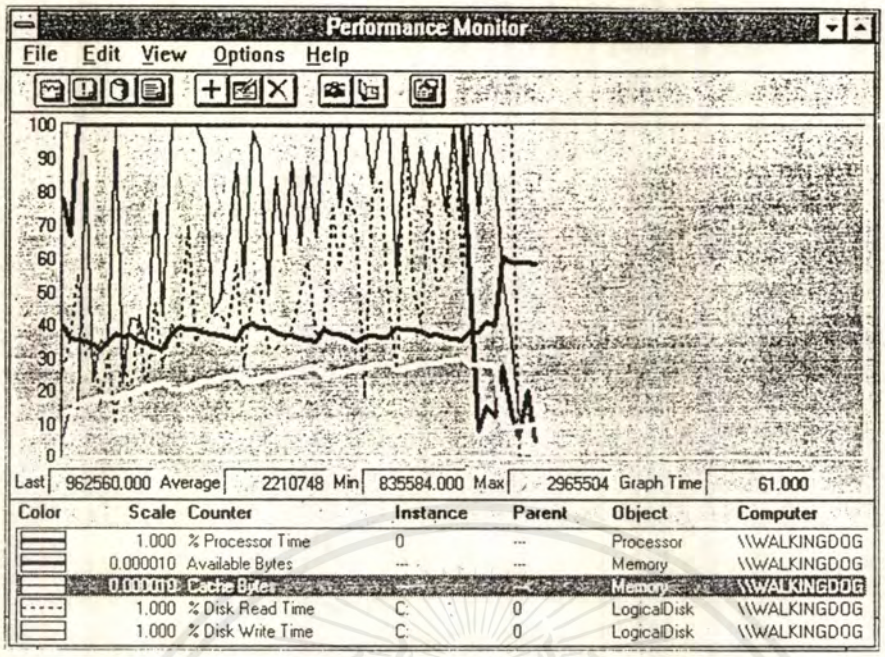
เคาน์เตอร์ซิงค์ (Sync) ใช้ในการบอกว่าการร้องขอใดเป็นซิงโครนัส (synchronous) หรืออะซิงโครนัส (asynchronous) ซึ่งไม่สำคัญนัก

สรุปการดูโอเปอร์เรชั่นการอ่านของไฟล์โดยปกติจะต้องดู Copy Read สำหรับโอเปอร์เรชั่นการเขียนของไฟล์โดยปกติจะต้องดูดาต้าฟลัช และจะดูกิจกรรมของไดเรกทอรี่หรือของไฟล์หลาย ๆ ไฟล์ จะต้องดูดาต้าแมป

5.3 การอ่านและการเขียนแบบซีเควนเชียล

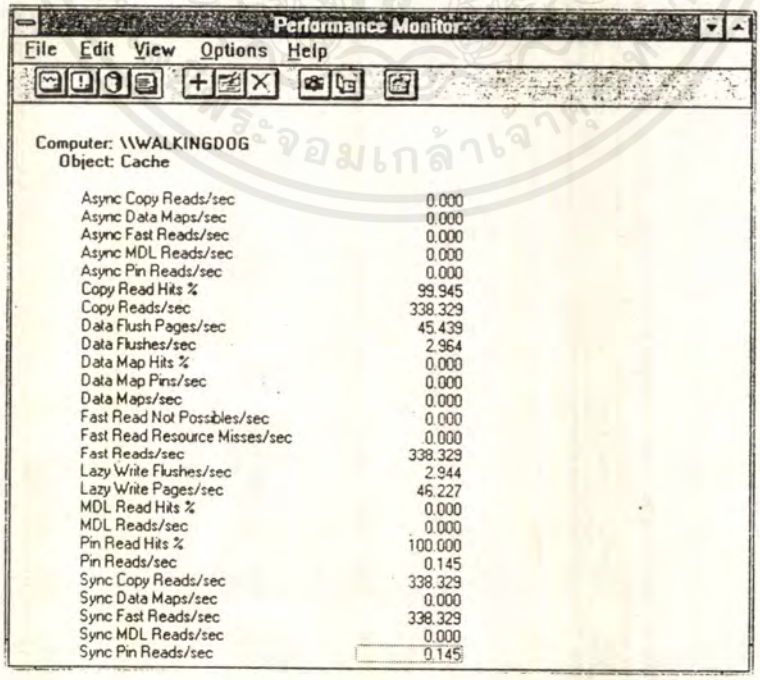
ต่อไปจะทำการทดสอบโดยการอ่านและเขียนไฟล์แบบซีเควนเชียลโดยตั้งให้ขนาดของเรคคอร์ดเท่ากับ 512 ไบต์ และไฟล์มีขนาด 10 MB.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-6 กิจกรรมของแคชและดิสก์ในการอ่านและเขียนไฟล์ขนาดใหญ่แบบซีเควนเชียล

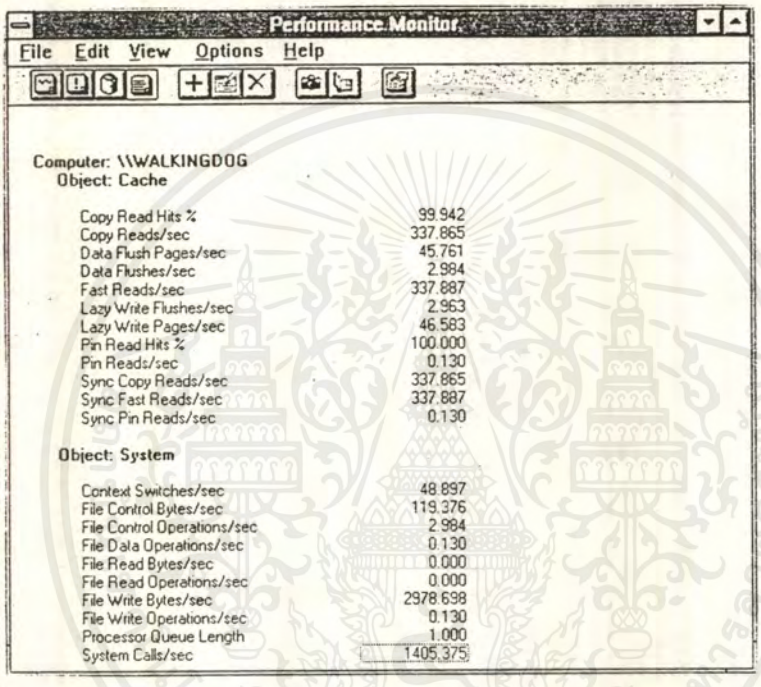
ผลประโยชน์ของโพรเซสเซอร์ (เส้นสีดำเข้มทางด้านบนของชาร์ต) มีค่าเป็น 100% จากรูปจะเห็นว่าดิสก์มีทั้งการอ่าน (เส้นประ) และการเขียน (เส้นสีดำเส้นเล็ก) ที่ยุ่งมาก แต่ค่าของแคชก็ไม่สูงขึ้นมาก แม้ว่าเราจัดการกับไฟล์ทั้งหมด 10 MB. ซึ่งมีข้อมูลมากกว่าตัวอย่างที่แล้ว เพราะตัวจัดการแคชตรวจจับได้ว่า ไฟล์ได้ถูกอ่านเข้ามาแบบซีเควนเชียล ทำให้การเก็บข้อมูลมาก ๆ ไว้ในแคชก็ไม่ได้ช่วยให้อะไรดีขึ้นมา เพราะข้อมูลในไฟล์จะไม่ถูกอ้างถึงอีกถ้าข้อมูลนั้นได้ถูกอ้างถึงไปแล้ว รูปต่อไปจะเป็นสถิติของแคชสำหรับกรณีนี้



รูปที่ 5-7 สถิติของแคชในการอ่าน/เขียนไฟล์ขนาดใหญ่แบบซีเควนเชียล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ Copy Reads/sec ซึ่งมีค่าสูงในตัวอย่างที่แล้ว ขณะนี้ได้ลดลงแล้ว มี 45 Data Flush Pages/sec แต่เกิดการ flush 2.9 ครั้งต่อวินาที แสดงว่าเราได้ส่งข้อมูลออกเท่ากับ 45/2.9 ~ 15 เพจต่อการฟลัช 1 ครั้ง จากข้อมูลเหล่านี้ ทำให้เราทราบว่า ตัวจัดการแคชได้พบว่ามีการเข้าถึงไฟล์แบบซีเควนเชียล และได้แบ่งเพจเป็นกลุ่ม กลุ่มละหลาย ๆ เพจ เลซี่ไรท์เตอร์ต้องการเขียนข้อมูลแบบซีเควนเชียล เป็น 64K ส่วน แต่อย่างไรก็ตามเลซี่ไรท์เตอร์ไม่ได้กระทำเช่นนี้ เพราะค่า Data Flush/sec มีค่ามากกว่า Lazy Write Flush/sec อยู่เพียงเล็กน้อย หมายความว่าแมปเพจไรท์เตอร์ ทำงานเกี่ยวข้องกับหน่วยความจำ และทำการเขียน เพจออกไปเอง สิ่งนี้ทำให้รบกวนการเขียนแบบซีเควนเชียลของเลซี่ไรท์เตอร์ และลดจำนวนเพจในการเขียนแต่ละครั้ง



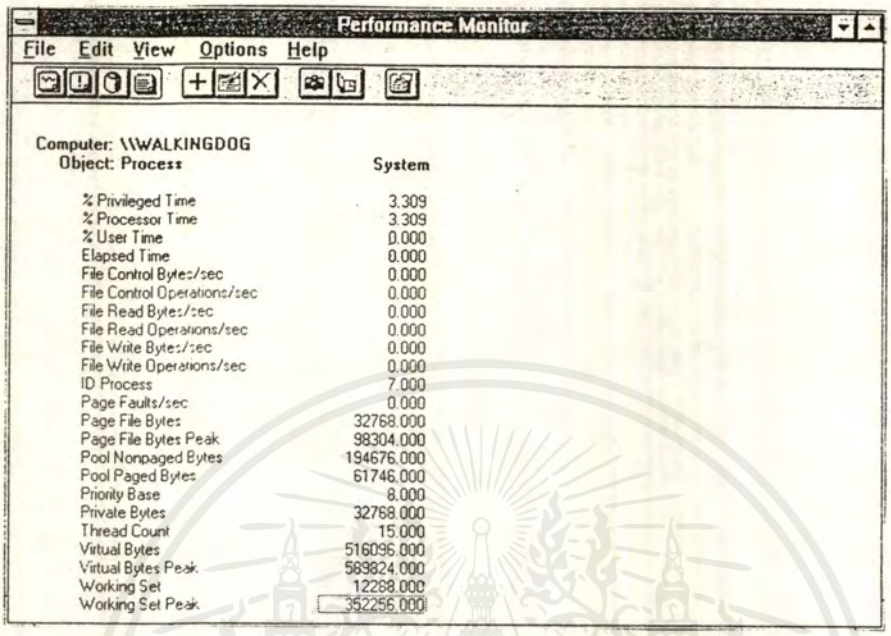
รูปที่ 5-8 สถิติของแคชและระบบสำหรับการอ่านและเขียนไฟล์ขนาดใหญ่แบบซีเควนเชียล

จากรูปที่ 5-8 จะเห็นได้ว่าเรากำลังอยู่ในช่วงการอ่านแบบเร็ว เพราะเคาน์เตอร์ของโอเปอร์เรชั่นของไฟล์ในออบเจ็กต์ System มีค่าเกือบจะเป็นศูนย์ หมายความว่าตัวจัดการไอโอ (I/O manager) หันไปร้องขอจากแคชมากกว่าที่จะร้องขอการเขียนอ่านจากระบบไฟล์ เราพบว่ามีจำนวน 1405 การร้องขอจากระบบ (system call) ทุก ๆ การอ่าน 338 ครั้ง แสดงว่ามี 4 การร้องขอจากระบบ ต่อการอ่าน 1 ครั้ง เราจึงทราบว่ามีการเขียน 1 ครั้ง ทุก ๆ การอ่านแต่ละครั้ง มันไม่ยากเลยที่จะดูว่าทำไมจึงเกิด 4 การร้องขอจากระบบต่อการอ่าน 1 ครั้ง โดยใช้เครื่องมือ (tool) ชื่อ WAP ซึ่งเครื่องมือนี้จะบอกถึงกิจกรรมของแอปพลิเคชั่นไฟล์

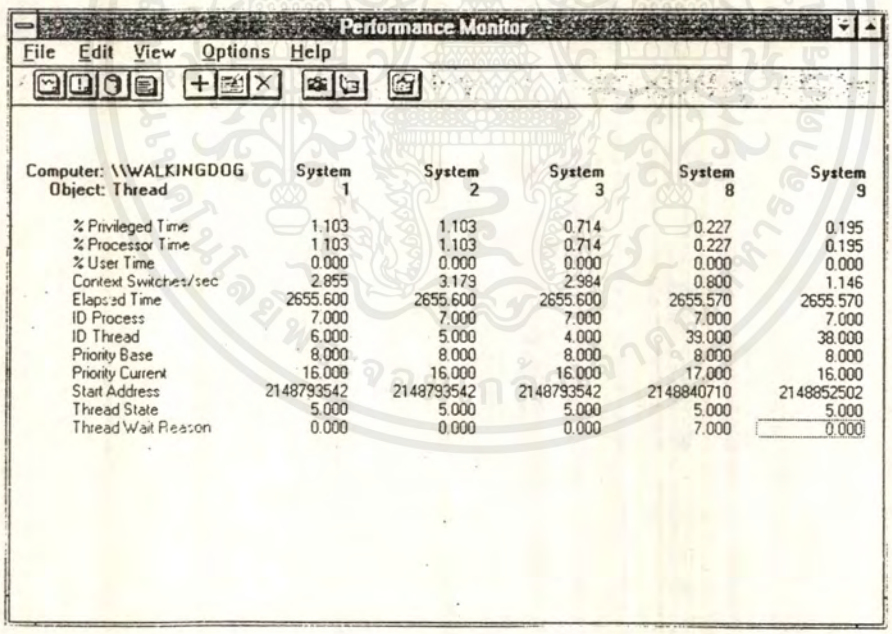
ต่อไปพิจารณาประสิทธิภาพของการฟลัชข้อมูล แม้ว่าจะมีการอ่านและการเขียนอย่างละเกือบ 338 ครั้งต่อวินาที เลซี่ไรท์เตอร์ถูกกระตุ้นเพียง 3 ครั้งต่อวินาทีเท่านั้น และแต่ละครั้งจะทำการเขียนจำนวน 15 เพจ กระบวนการ System ใช้เวลาเพียงแค่ 3.3% ในการทำทั้งหมดนี้ รูปต่อไปแสดงถึงเธร็ดของกระบวนการของระบบ ซึ่งกระบวนการนี้ใช้เวลาของโพรเซสเซอร์น้อยมากในการนำเพจออก เธร็ดที่กล่าวถึงในที่นี้ ส่วนใหญ่ คือ เธร็ดของเลซี่ไรท์เตอร์ , เธร็ดของแมปเพจไรต์เตอร์ และเธร็ดของตัวเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพจที่ถูกเปลี่ยนแปลงแล้ว ถ้าระบบกำลังสร้างไฟล์ดีมานซีโรเธร็ด (demand zero thread) จะสร้างเพจเฟรมที่มีแต่ค่าศูนย์ ถ้าหน่วยความจำเต็ม เธร็ดของตัวจัดการ working set ก็จะทำการปรับแต่ง working set เพื่อให้มีพื้นที่มากขึ้น



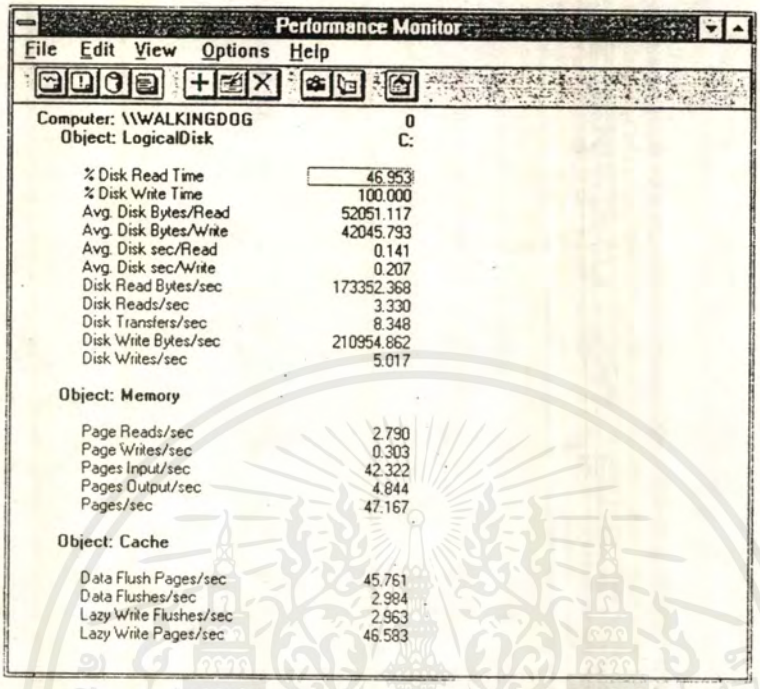
รูปที่ 5-9 การเขียนแบบเลซี่ (lazy writing) โดยกระบวนการของระบบ



รูปที่ 5-10 เธร็ดของกระบวนการของระบบที่ช่วยแบ่งงานของเลซี่ไรท์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปดูการเป็นไปของดิสก์ภายใต้ความกดดันนี้ รูปที่ 5-11 ดูที่การนำเพจออก คือทำการบวกแคช Data Flush Pages/sec กับ Memory:Output Pages/sec จะได้ 50.605/sec แล้วคูณด้วย 4096 bytes/page จะได้ 207208 bytes/sec ซึ่งมีค่าใกล้เคียงกับ 210955 Write Bytes/sec ของดิสก์ไดร์ฟ

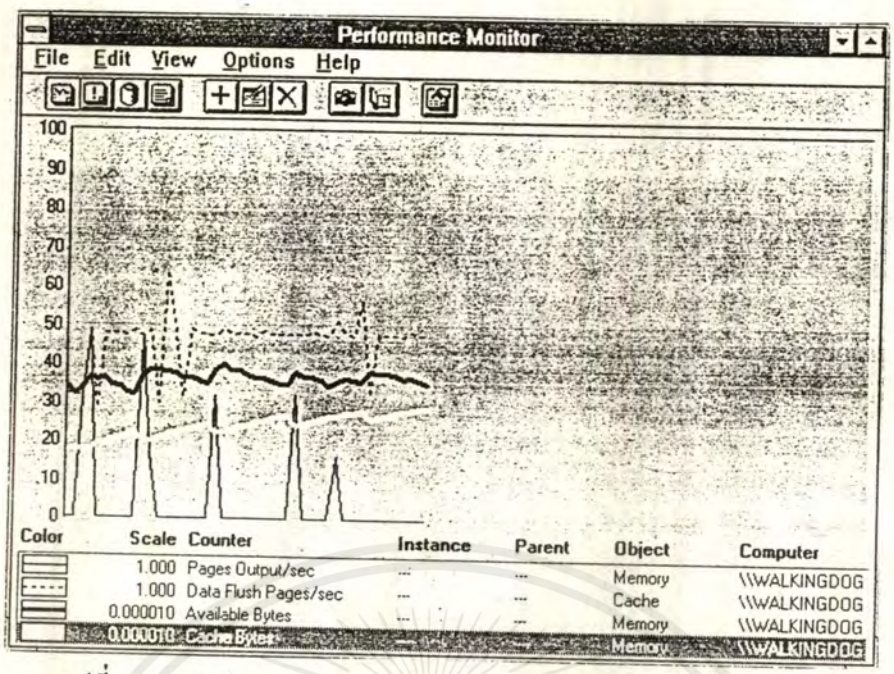


รูปที่ 5-11 การตอบสนองของดิสก์ต่อกิจกรรมของแคชในระหว่างการอ่าน/เขียนแบบซีเควนเชียล

ต่อไปพิจารณาการอ่านของ Memory:Pages Input/sec = 42.322 แล้วคูณด้วยขนาดของเพจจะได้ 173392 bytes input /sec ซึ่งใกล้เคียงกับค่า 173352 Disk Read Bytes/sec (ดูที่กฎข้อที่9)

ดูที่ Avg. Disk Bytes/Read และ Avg. ดิสก์ Bytes/Write จะเห็นว่าเป็นตัวเลขที่สูงพอสมควร ซึ่งดีแต่เนื่องจากเลซี่ไรท์เตอร์พยายามที่จะเขียนออกไป 64K ส่วน แบบซีเควนเชียล จึงเป็นสิ่งที่น่าอายที่ตัวเลขของ Avg. ดิสก์ Bytes/Write ไม่สูงขนาดนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-12 ตัวจัดการหน่วยความจำและตัวจัดการแคชทำงานประสานงานกัน

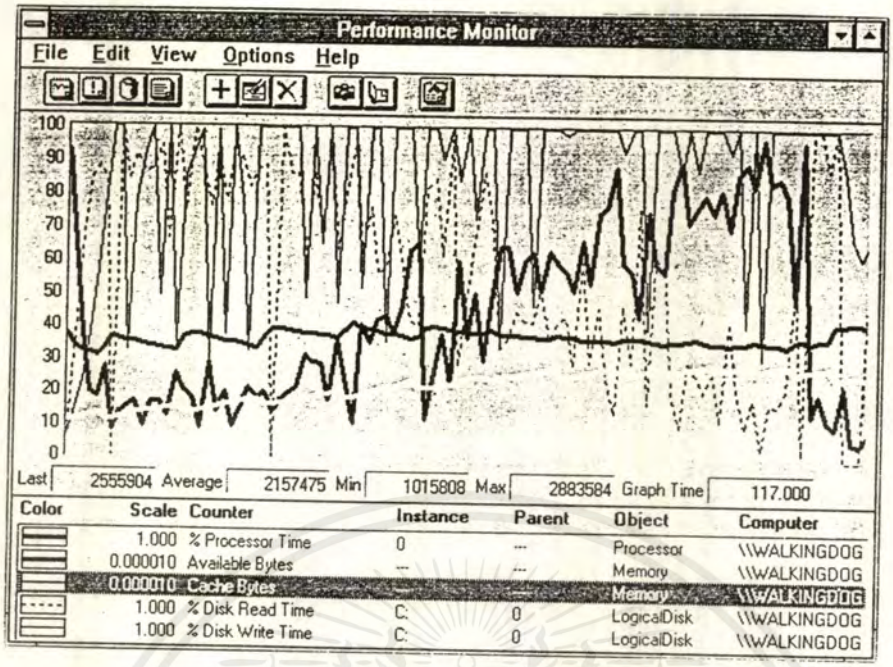
เส้น Memory:Pages Output/sec (เส้นบางสีดำ) ซึ่งเป็นงานของตัวจัดการหน่วยความจำนั้น มียอดแหลม 5 ยอด คือ ยอดแรกแสดงว่ามีเพจถูกนำออก 48 เพจ ซึ่งเป็นผลบวกของ Available Bytes และแคช แต่ละวินาทีตัวจัดการแคชจะพยายามเขียนข้อมูล 48 เพจ แต่หลังจากที่ตัวจัดการหน่วยความจำได้เขียนข้อมูลทั้ง 48 เพจแล้ว ตัวจัดการแคชจะเขียน 30 เพจต่อวินาที ที่ตำแหน่งยอดที่ 2 ตัวจัดการหน่วยความจำมีการเขียนข้อมูลเพิ่มขึ้น โดยนำบางเพจมาจากแคช แต่เรารู้ว่ามีการนำเพจมาจาก working set อื่นด้วย เพราะการเพิ่มขึ้นของ Available Bytes มีค่ามากกว่าการลดจำนวนไบต์ของแคช ตัวจัดการแคชจึงทำการเขียนน้อยกว่า 48 เพจ และเป็นอย่างนี้ไปเรื่อย ๆ ใน 3 วินาทีถัดไป

ในยอดที่ 3 ตัวจัดการหน่วยความจำเขียนข้อมูลจำนวน 32 เพจ ขณะนี้แคช ได้ให้ Available Bytes เป็นส่วนใหญ่ ในยอดที่ 4 เขียนข้อมูลจำนวน 32 เพจเช่นกัน แต่ในจำนวนนี้เกือบทั้งหมดมาจากแคช แล้วตัวจัดการหน่วยความจำก็พบว่าไม่มีความก้าวหน้าอะไรเกิดขึ้น แต่ก็จะพยายามเป็นครั้งสุดท้าย โดยนำ 16 เพจออกมาจากแคช และ 2-3 นาทีต่อมาตัวจัดการแคช จึงเขียนเพจด้วยจำนวนที่น้อยกว่าลงดิสก์

5.4 การอ่านและการเขียนแบบสุ่ม

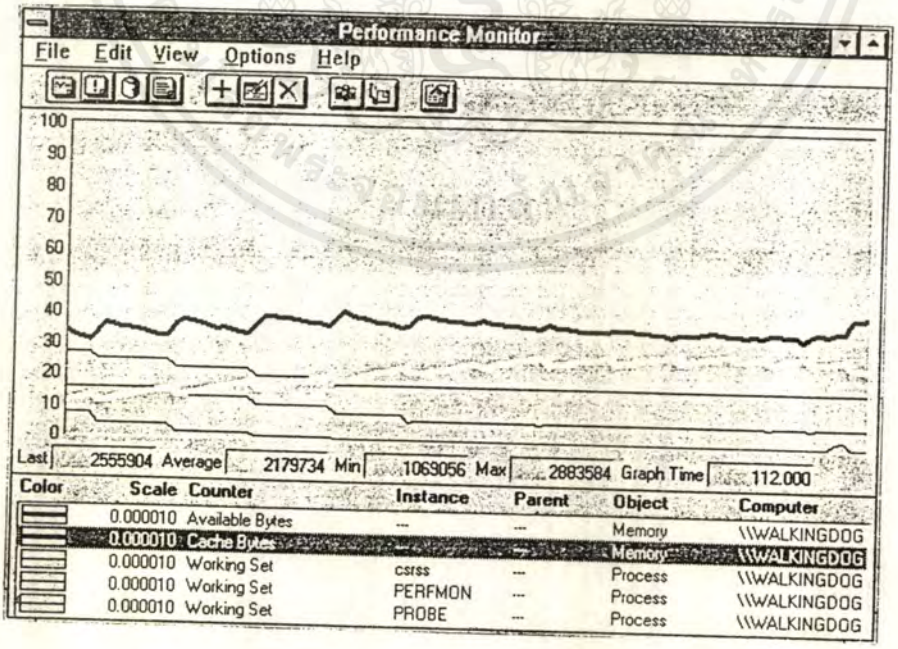
ดูการทำงานของแคชเมื่อเราเข้าถึงไฟล์ขนาด 3 MB. ด้วยการแจกแจงแบบปกติ อันดับแรกจะอ่านเรคคอร์ดขนาด 4096 ไบต์ เข้ามาก่อนแล้วจึงเขียนออกไป ดูรูปที่ 5-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



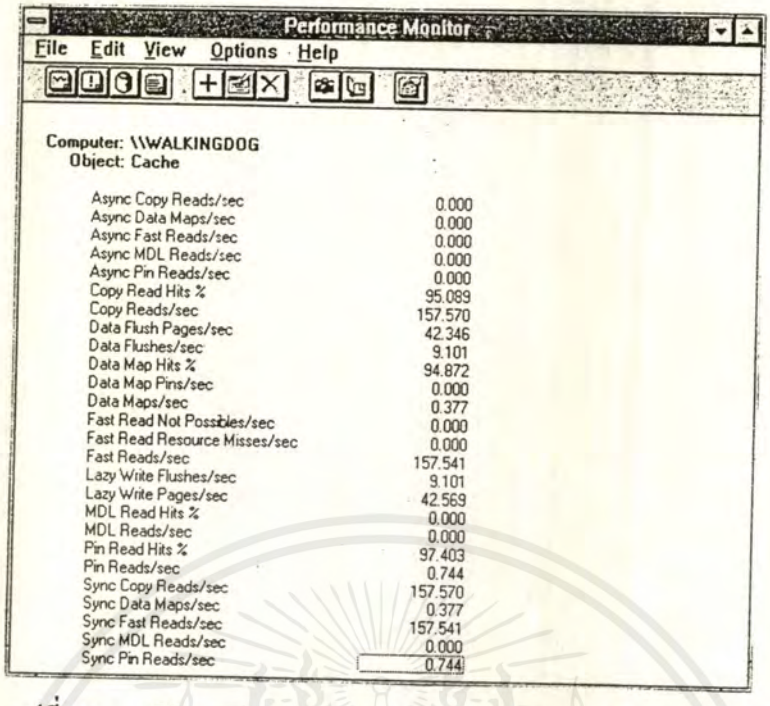
รูปที่ 5-13 การอ่านและการเขียนแบบสุ่มที่ยุ่งมาก

เส้นประ แสดง %Disk Read Time ซึ่งอยู่ในระดับที่ยุ่งพอสมควร ส่วนเส้นบางสีดำแสดง % Disk Write Time มันใช้เวลาส่วนใหญ่เกือบ 100% เพราะยิ่งใช้เวลาน้อยในการอ่านข้อมูลจากดิสก์ จะทำให้เขียนได้เร็วขึ้นและมากขึ้นด้วย เส้นสีดำเส้นใหญ่แสดง % Processor Time ที่เพิ่มขึ้นอย่างสม่ำเสมอ เส้นสีดำเส้นใหญ่อีกเส้นหนึ่ง แสดง Available Bytes ซึ่งมีค่าค่อนข้างคงที่ แสดงว่าแคชมีการนำเพจใหม่มาจาก working sets ที่ไม่แอคทีฟแล้ว ดังแสดงในชาร์ตถัดไป



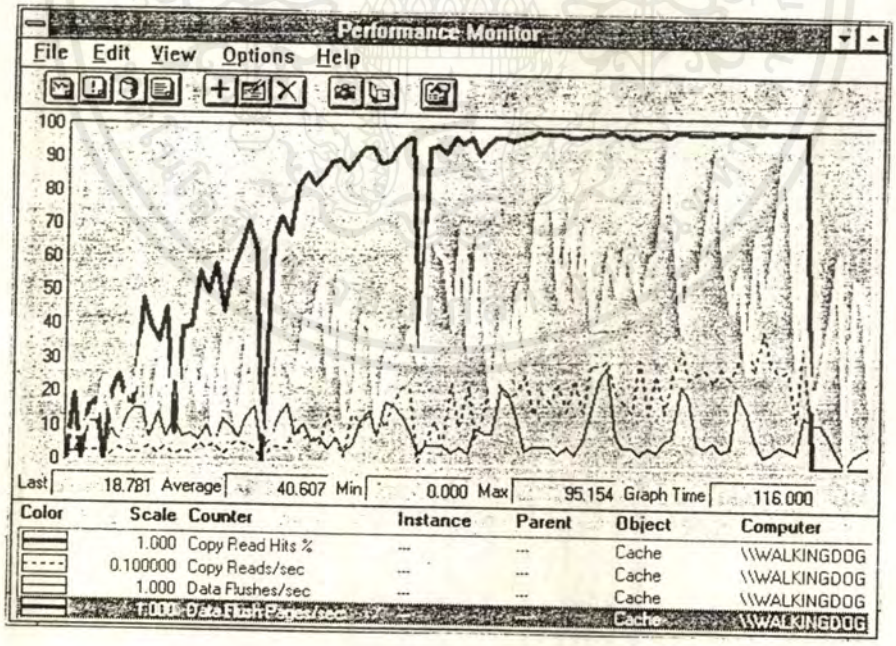
รูปที่ 5-14 แคชนำเพจที่ working set อื่นต้องการมาเป็นจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-15 สถิติของแคชสำหรับการอ่าน/เขียนโดยการแจกแจงแบบปกติ

จากรูป ค่าของ Copy Reads/sec จะมีค่าต่ำกว่ากรณีอ่านแบบซีเควนเชียล เพราะตัวจัดการแคชสามารถคาดล่วงหน้าได้ว่าจะมีการร้องขอการอ่านมาเมื่อไร รูปที่ 5-15 เป็นตัวอย่างของกฎข้อที่ 10 ซึ่งเป็นกฎพื้นฐานแต่มีรายละเอียดที่สำคัญซ่อนอยู่ ซึ่งรูปที่ 5-16 จะแสดงว่าทำไมจึงเป็นเช่นนี้

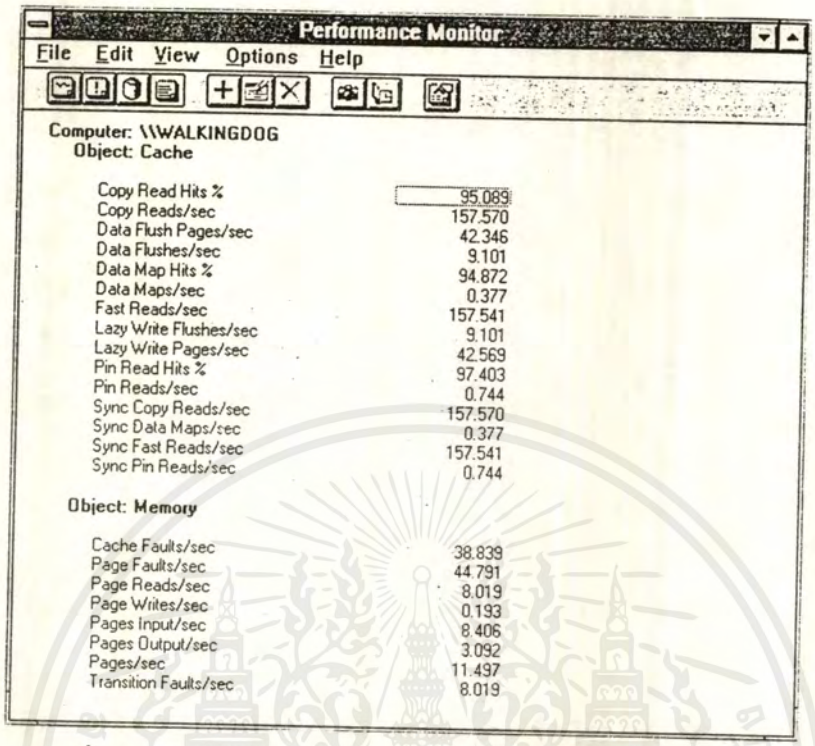


รูปที่ 5-16 สถิติของแคชสำหรับการอ่าน/เขียนโดยการแจกแจงแบบปกติ

เส้นสีดำเส้นใหญ่ แสดง Copy Read Hits % มีค่าเพิ่มขึ้นเรื่อย ๆ ซึ่งเป็นการดี ส่วน Copy Reads/sec ตอนเริ่มต้นจะมีค่าค่อนข้างต่ำ แต่ก็มีค่าเพิ่มขึ้นเมื่ออัตราฮิตเพิ่มขึ้น เช่นเดียวกับ Data Flush

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pades/sec จะมีค่าเพิ่มขึ้นทางด้านขวามือของชาร์ต ค่าต่าง ๆ เหล่านี้ จะมีแนวโน้มเป็นเช่นนี้ไปเรื่อย ๆ จนกระทั่งจบการทดสอบ เมื่อเลซี่ไรท์เตอร์ได้ทำการเคลียร์ dirty data ของแคช

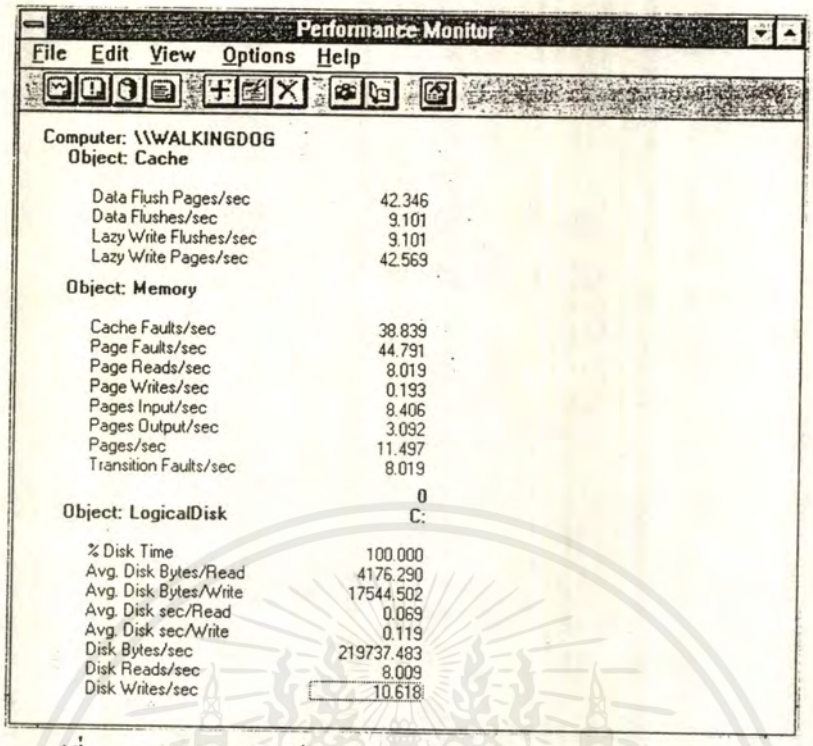


รูปที่ 5-17 ตัวจัดการหน่วยความจำและแคชโดยการแจกแจงแบบปกติ

รูปที่ 5-17 แสดงสถิติของตัวจัดการหน่วยความจำเมื่อเทียบกับแคช ค่า Cache Faults/sec เป็นสับเซต (subset) ของ Page Faults/sec เพราะฉะนั้นจึงสามารถบอกได้ว่าขนาดของการจัดการกับหน่วยความจำขึ้นอยู่กับกิจกรรมของแคช เพราะ Cache Faults/sec มีค่า 38.8 ซึ่งเป็นค่าส่วนใหญ่ของ 44.8 Page Faults/sec แต่ในจำนวนนี้มี 2-3 เพจ ที่เป็นซอร์ฟฟอลต์ (soft fault) เพราะจำนวนของ Page Read/sec มีค่าเพียง 8.0 นอกจากนี้การอ่านแบบซีเควนเชียลก็น้อยลง คือ มีเพียง 8.4 Pages Input/sec มีเพจจำนวนไม่มากที่ได้มาจากการอ่านแต่ละครั้ง

ต่อไปพิจารณาที่การนำข้อมูลออก Data Flush Pages / sec มีค่า 42.3 และ Pages Output / sec มีค่า 3.1 แสดงว่า แคชจะเป็นตัวส่งข้อมูลออกเป็นส่วนใหญ่ ในความเป็นจริงแล้วเราได้พบเหตุการณ์เช่นนี้มาก่อนแล้ว : เพราะ Page Write / sec มีค่าเพียง 0.2 และ Pages Output / sec มีค่า 3.1 แสดงว่ามี 3.1 / 0.2 = 15.5 เพจ ต่อ 1 การจัดการนำข้อมูลออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-18 สถิติของดิสก์ในการอ่าน/เขียนโดยการแจกแจงแบบปกติ

5.5 การแมปไฟล์ลงในหน่วยความจำ

ในแอปพลิเคชั่นมีวิธีที่จะเข้าถึงไฟล์ได้เร็วกว่าการใช้แคช โดยการแมปไฟล์เข้าไปในพื้นที่ของมันโดยเฉพาะ เพื่อที่จะเข้าถึงข้อมูลเหมือนเป็นอาร์เรย์และไม่ต้องมีการเรียกระบบไฟล์ด้วย ทำให้ลด overhead ของการเรียกระบบไฟล์และการค้นหาของแคช ในตารางถัดไปจะแสดงประสิทธิภาพของเครื่องที่มีการเข้าถึงไฟล์โดยการแมปด้วย

ตารางที่ 5-2 เวลาของ memory mapped เทียบกับไฟล์ I/O มีหน่วยเป็นมิลลิวินาทีต่อเรคคอร์ด

Type of file access	Operation	Non-cached time	Cached time	Mapped time
Sequential	Read	6.58	1.32	0.75
Sequential	Write	22.91	1.70	0.64
Random	Read	20.45	1.51	0.97
Random	Read/Write	40.66	3.16	1.31

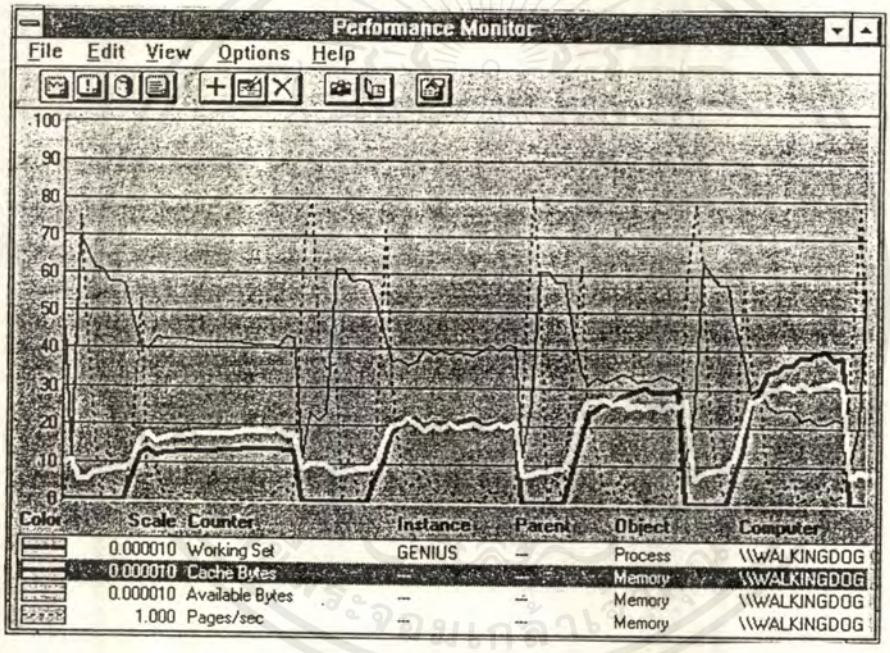
จากตารางดูเหมือนการแมปไฟล์เป็นสิ่งที่ดี แต่ก็ไม่แนะนำเพราะจะต้องเขียนโค้ดของแอปพลิเคชั่น ซึ่งมีอยู่แล้วใหม่ เพื่อที่จะไม่ให้มีการเรียกระบบไฟล์แบบเก่า ถึงแม้ว่าการเขียนโค้ดจะเป็นสิ่งที่ง่าย แต่ก็เป็นการเสียเวลา เพราะมีโค้ดเดิมอยู่แล้ว และอีกอย่างก็คือ การเข้าถึงไฟล์แบบซีเควนเชียลนั้น แคชจะใช้หน่วยความจำน้อยมากในการอ่านไฟล์ และการใช้การแมปหน่วยความจำนี้ หมายถึง ไม่สามารถเข้าถึงระบบไฟล์ในโหมดซิงโครนัสได้ เช่น file locking เป็นต้น หมายความว่า การเขียนลงไฟล์หลาย ๆ ครั้งพร้อมกัน จะต้องสัมพันธ์กันโดยใช้มิวเทกซ์ (mutexes) และถ้ามีหลาย ๆ กระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ต้องการเขียนลงไฟล์พร้อม ๆ กัน ซึ่งกระบวนการเหล่านี้มาจากคอมพิวเตอร์ต่างเครื่องกัน จะต้องมีการสร้างกลไกการซิงโครไนซ์ระหว่างกระบวนการซึ่งเป็นการลดประสิทธิภาพของการแมปหน่วยความจำลง

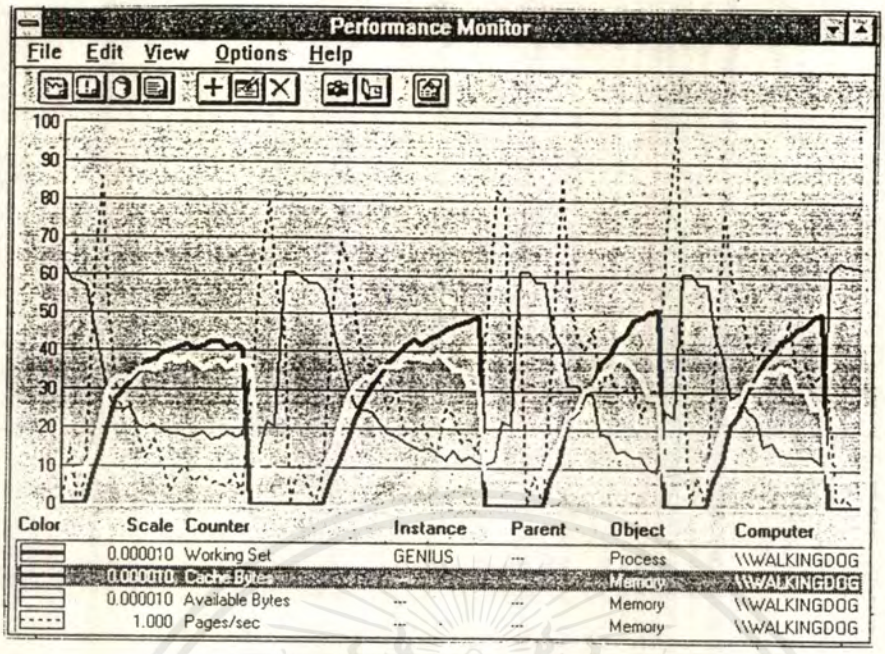
ความแตกต่างของระบบระหว่างการใช้แมปไฟล์และไม่ได้แมป คือ working set ของแมปไฟล์จะอยู่ที่กระบวนการและบัฟเฟอร์ไฟล์ (ไฟล์ที่ไม่มีการแมป) working set จะอยู่ที่แคชเมื่อกระบวนการทำการแมปไฟล์เข้าไปในพื้นที่ของมัน มันจะใช้ส่วนของ RAM เพียงเล็กน้อยในการเก็บไฟล์ แต่สำหรับมุมมองของตัวจัดการหน่วยความจำนั้นไม่มีความแตกต่างอะไรระหว่าง working set ของกระบวนการหรือของแคช

ต่อไปทดสอบโดยใช้ 2 กระบวนการ ซึ่งแต่ละการเข้าถึงไฟล์จะเป็นการเข้าถึงแบบมีการแจกแจงปกติ โดยการแจกแจงจะครอบคลุม 1MB. , 2MB. , 3MB. ไปจนถึง 8MB. กระบวนการหนึ่งใช้การเรียกระบบไฟล์ และอีกกระบวนการหนึ่งใช้การแมปไฟล์ ผลจะแสดงไว้ใน 2 ชาร์ตถัดไป รูปที่ 5-19 แสดงการทดลองที่มี working set 1 - 4 MB และรูปที่ 5-20 แสดง working set จาก 5 - 8 MB



รูปที่ 5-19 กระบวนการที่มีการแข่งขันกันในการอ่านแบบใช้แมปไฟล์และระบบไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

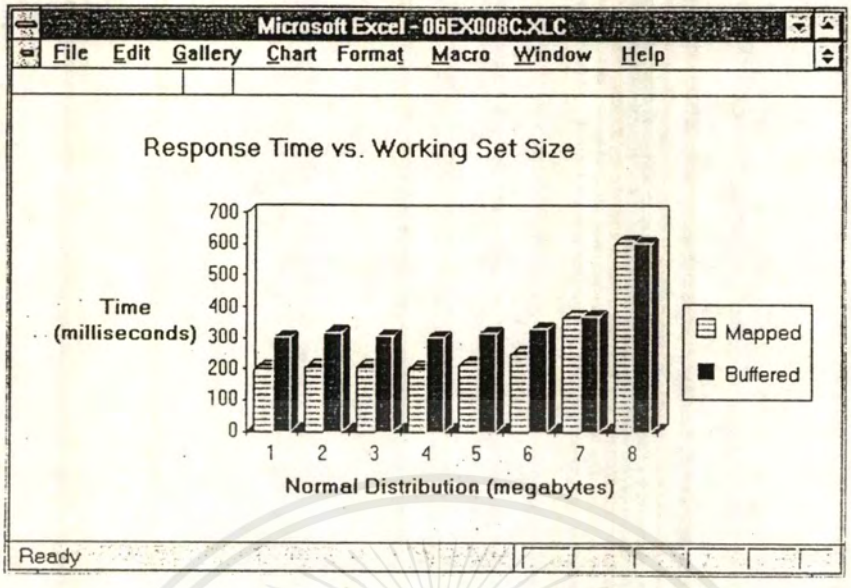


รูปที่ 5-20 กระบวนการที่มีการแข่งขันกันมากขึ้นในการอ่านแบบใช้แมปไฟล์และระบบไฟล์

ในตอนเริ่มต้นเส้นแคช ซึ่งมีแถบสว่างอยู่นั้นมีค่าค่อนข้างน้อย แต่ก็มีค่าเพิ่มขึ้นเมื่อขนาดของ working set ของกระบวนการเพิ่มขึ้น เส้นเล็กสีดำแสดงถึง Available Bytes ซึ่งมีค่าลดลง และเส้นประแสดงถึง Pages/sec ซึ่งเพิ่มขึ้นเมื่อการทดลองได้ก้าวหน้าขึ้น เมื่อการแจกแจงปกติได้ครอบคลุมถึง 8MB. อัตราการเพจจิ้งก็มากขึ้นในทันที

ชาร์ตนี้ดูเหมือนจะบ่งบอกว่าตัวจัดการหน่วยความจำสนับสนุน working set ของกระบวนการมากกว่าแคช ซึ่งในบางแง่มุมสิ่งนี้เป็นสิ่งที่ถูกต้อง โดยปกติโค้ดและข้อมูลที่ถูกอ้างถึงโดยตรงโดยกระบวนการนั้นมีความสำคัญต่อประสิทธิภาพของแอปพลิเคชั่นมากกว่าข้อมูลไฟล์ในแคช แคชมีแนวโน้มที่จะได้พื้นที่ที่กระบวนการไม่ต้องการ แต่ในรูปต่อไปจะแสดงว่าการสนับสนุนกระบวนการนั้นก็ไม่ได้ดีเหนือกว่าแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-21 เวลาการตอบสนองของกระบวนการที่ใช้การแมปไฟล์และบัฟเฟอร์

รูปที่ 5-21 แสดงเวลาการตอบสนองของแต่ละกระบวนการ ซึ่งวัดโดย Response Probe ในตอนเริ่มต้นการเข้าถึงแบบแมปจะเร็วกว่า ดังในตารางที่ 5-2 แต่เมื่ออัตราการเพจจิ้งเพิ่มขึ้น และเวลาการเข้าถึงดิสก์กลายเป็นส่วนสำคัญของเวลาการตอบสนอง ทำให้ประสิทธิภาพของทั้ง 2 กระบวนการขึ้นมาเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

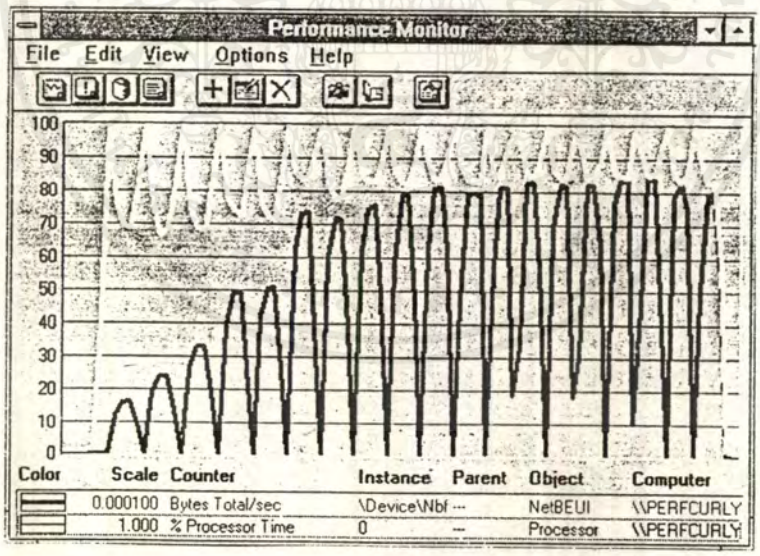
บทที่ 6

การตรวจจับปัญหาคอขวดที่เกิดจากเครือข่าย

สิ่งสำคัญในการหาจุดคอขวดของเครือข่าย คือ ทุกสิ่งทุกอย่างที่ได้กล่าวถึงในบทก่อนหน้านี้ เช่น เซิร์ฟเวอร์ที่มีดิสก์เป็นคอขวด เพราะหน่วยความจำไม่พอก็คือ คอมพิวเตอร์เครื่องหนึ่งที่มีดิสก์เป็นปัญหาคอขวด ที่มีหน่วยความจำไม่พอเช่นกัน แต่เซิร์ฟเวอร์จะมีผลกระทบต่อคนหลาย ๆ คน

6.1 ธรูพุทของเครือข่าย

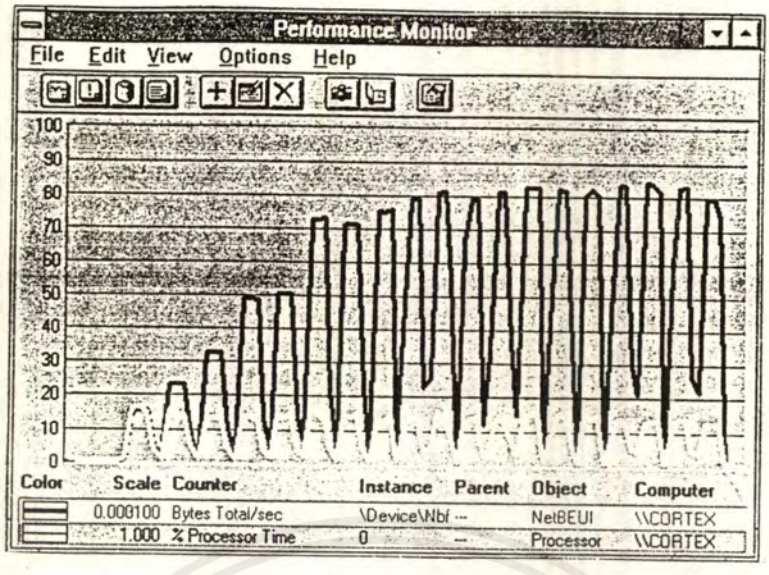
เราใช้อ่านจากไฟล์แบบไม่ใช้บัฟเฟอร์ ดังนั้นไม่ต้องพิจารณาแคชบนไคลเอ็นท์ แต่ให้พิจารณาแคชบนเซิร์ฟเวอร์ เพราะเมื่อเราทำการอ่านเรคคอร์ดเดิมซ้ำ ๆ เป็นการฮิตแคชในทุก ๆ ครั้ง โดยการเพิ่มขนาดของเรคคอร์ด เราจะสามารถเห็นได้ชัดว่ามีข้อมูลจำนวนเท่าไรที่ ไคลเอ็นท์ตัวเดียวสามารถส่งข้ามเครือข่ายนี้ โดยคุณสมบัติของเซิร์ฟเวอร์ ที่ใช้มีดังนี้ 100 - MHz. MIPS 4000, แคช 1 MB, memory bus 50 - MHz., RAM 32 MB และมี Sonic DMA Ethernet controller แบบ on - board ส่วนคุณสมบัติของไคลเอ็นท์ที่ใช้ คือ 486 / 33 มี RAM 16 MB และ 16 bit memory mapped network adapter สายสื่อสารเป็นเคเบิล Ethernet และมีโปรโตคอล คือ NetBEUI เนื่องจากเราใช้การอ่านแบบไม่ใช้บัฟเฟอร์ เราจึงต้องอ่านเซกเตอร์ (sector) ของดิสก์หลาย ๆ ขนาด เริ่มจาก 512 - , 1024 - , 2048 - และ 4096 - ไบต์ แล้วเพิ่มทีละ 4096 ไบต์ ไปจนกระทั่งถึง 64 K เรคคอร์ด ผลของ NetBEUI : Byte Total / sec จากทางด้านไคลเอ็นท์ แสดงดังรูปที่ 6-1



รูปที่ 6-1 มุมมองทางด้านไคลเอ็นท์ของการทดสอบธรูพุทของเครือข่าย

สามารถวัดธรูพุทได้โดยดู NetBEUI : Bytes Total / sec ซึ่งเพิ่มขึ้นตามขนาดของเรคคอร์ด ต่อไปดูทางด้านเซิร์ฟเวอร์บ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



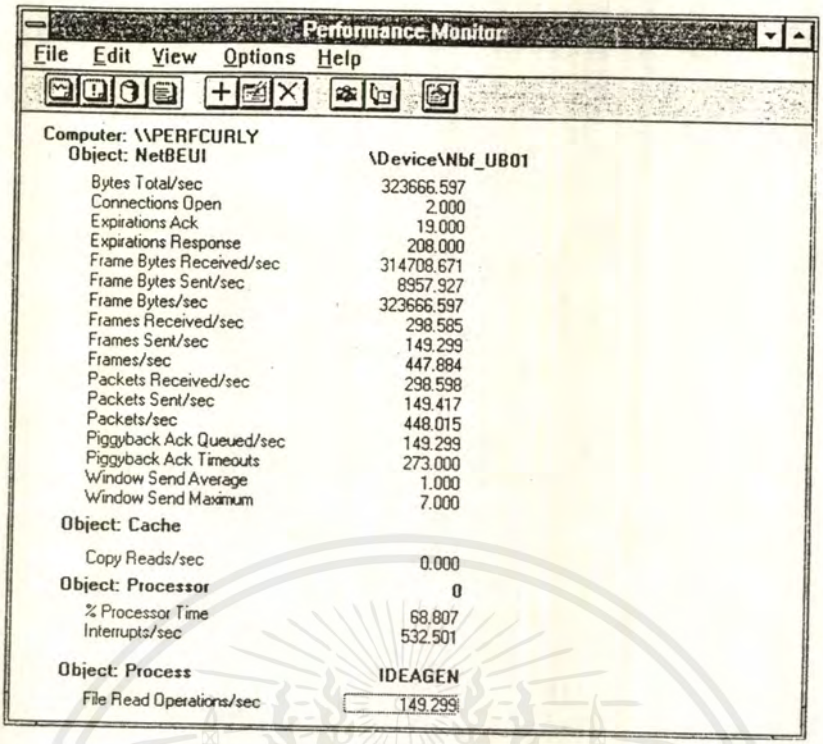
รูปที่ 6-2 มุมมองทางด้านเซิร์ฟเวอร์ของการทดสอบธรูพุทของเครือข่าย

ธรูพุทสัมพันธ์กันทางด้านไคลเอ็นท์ แต่ผลประโยชน์ของโพรเซสเซอร์ต่ำกว่ามาก ทางด้านไคลเอ็นท์ประโยชน์ของโพรเซสเซอร์ค่อนข้างคงที่

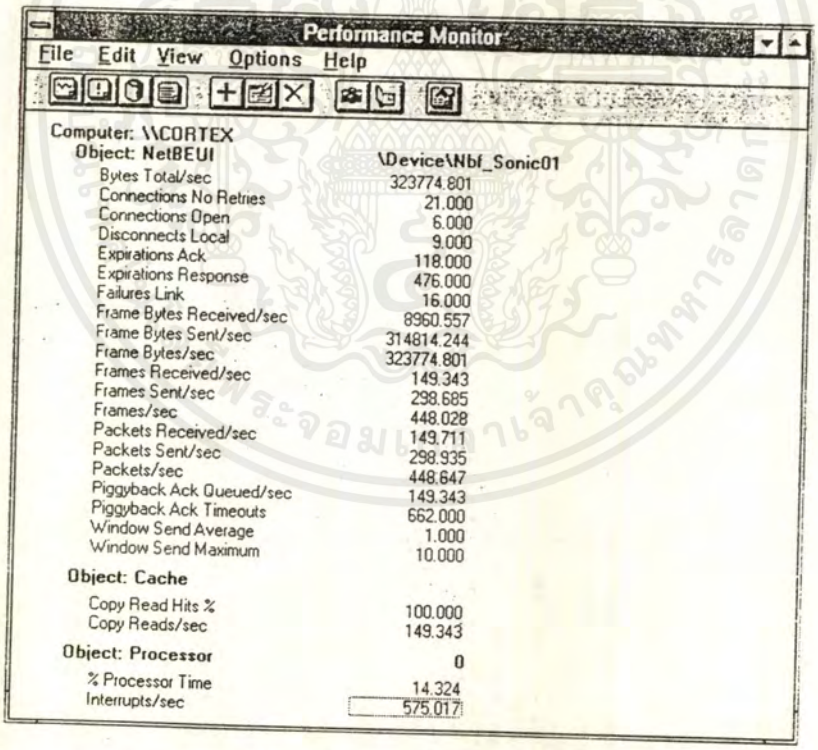
ธรูพุทเพิ่มขึ้นเกือบเป็นเส้นตรงจนถึงจุดที่ขนาดของ เรคคอร์ด = 4096 ไบต์ แต่เมื่อถึง 8192 ไบต์ จะมีธรูพุทมากกว่าที่ 4096 ไบต์ เพราะว่า Redirector file system มองเหมือนว่าการร้องขอของ 8192 ไบต์ เป็นการร้องขอขนาด 4096 ไบต์ 2 ครั้ง

NetBEUI : Bytes Total / sec ประกอบด้วยกิจกรรมที่ใช้เฟรมและดาต้าแกรม (datagram) เป็นพื้นฐาน (เมื่อเฟรมถูกส่งผ่านไปในเครือข่ายและคาดว่าจะได้รับสัญญาณว่าได้รับเฟรมนั้นแล้วจากผู้รับ และสามารถส่งเฟรมนั้นกลับไปใหม่ได้ ถ้าไม่ได้รับสัญญาณจากผู้รับ ส่วนดาต้าแกรมนั้นจะไม่รอสัญญาณใด ๆ ทั้งสิ้น และไม่สามารถส่งข้อมูลนั้นกลับไปใหม่ได้ ถ้าเกิดข้อผิดพลาดขึ้นมา) เคาน์เตอร์ของดาต้าแกรมเป็นตัวสำคัญในการบอกกิจกรรมได้ โดยเพียงรู้ว่าแอปพลิเคชั่นตัวใดที่ใช้ดาต้าแกรม ก็จะรู้ว่าอะไรเป็นเหตุให้เกิดกิจกรรมบนเครือข่าย รูปที่ 6-3 ไม่มีกระบวนการใดใช้ดาต้าแกรมจึงไม่มีเคาน์เตอร์ของดาต้าแกรมแสดงให้ดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-3 มุมมองทางด้านไคลเอ็นท์ของการอ่านเรคคอร์ดขนาด 2048 ไบต์แบบไม่ใช้บัฟเฟอร์



รูปที่ 6-4 มุมมองทางด้านเซิร์ฟเวอร์ของการอ่านเรคคอร์ดขนาด 2048 ไบต์แบบไม่ใช้บัฟเฟอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 รูปแบบง่าย ๆ ของจุดคอขวดของเครือข่าย

รูปที่ 6-3 และ 6-4 จะตรงกับยอดที่สามทางด้านซ้ายมือของรูปที่ 6-1 และ 6-2 ตามลำดับ ถ้าเราเพิ่มขนาดของเรคคอร์ด ทำให้ธรูพุทเพิ่มขึ้น

ต่อไปจะเป็นการกำหนดว่ามีอะไรเกิดขึ้นบ้างในการอ่าน 1 ครั้ง นำเวลาทั้งหมดในการอ่าน 1 ครั้งหารด้วย Process : File Read Operations / sec จะได้ 0.00668 วินาที และโดยปกติจะมีสิ่งต่าง ๆ เกิดขึ้นดังนี้ :

- เวลาของโพรเซสเซอร์บนไคลเอ็นท์
- เวลาของสื่อในการส่งการร้องขอข้อมูลมายังเซิร์ฟเวอร์
- เวลาของโพรเซสเซอร์บนเซิร์ฟเวอร์
- เวลาของสื่อในการส่งข้อมูล 2108 ไบต์กลับมา
- เวลาของโพรเซสเซอร์บนไคลเอ็นท์ ที่ใช้ในการนำข้อมูลไปยังบัฟเฟอร์ของแอปพลิเคชั่น

สมมุติว่าไม่มีเวลาคาบเกี่ยวกันระหว่างเวลาของสื่อและโพรเซสเซอร์ ดังนั้นเวลาในการอ่าน 1 ครั้ง จึงเป็นเวลาของโพรเซสเซอร์บนไคลเอ็นท์ + เวลาของสื่อ + เวลาของโพรเซสเซอร์บนเซิร์ฟเวอร์ เวลาของโพรเซสเซอร์บนเซิร์ฟเวอร์ที่ใช้ใน 1 วินาที คือ ค่า Processor : % Processor Time ซึ่งมีค่าระหว่าง 0 กับ 1 ในที่นี้เท่ากับ 0.14324 วินาที ส่วนบนไคลเอ็นท์เท่ากับ 0.68807 วินาที แล้วหารแต่ละค่าข้างต้นด้วยจำนวนของการอ่านใน 1 วินาที จะได้เวลาของโพรเซสเซอร์บนเซิร์ฟเวอร์ ต่อการอ่าน 1 ครั้ง เท่ากับ 0.000959 วินาที และบนไคลเอ็นท์เท่ากับ 0.004609 วินาที

การอ่านแต่ละครั้งจะส่งข้อมูล 2168 ไบต์ (ได้จากการหาร Bytes Total / sec ด้วย Frames Total / sec) สื่อ (ในที่นี้ คือ Ethernet) ส่ง 800 nanoseconds / byte แล้วคูณด้วย จำนวน bytes / read จะได้ 0.001734 sec / read แล้วรวมเวลาของโพรเซสเซอร์บน เซิร์ฟเวอร์, บน ไคลเอ็นท์ และเวลาของสื่อเข้าด้วยกัน จะได้ 0.007302 วินาที ซึ่งมีค่ามากกว่าเวลาที่เราคิดไว้ข้างต้นสำหรับแต่ละเรคคอร์ด อยู่ 604 ไมโครวินาที แสดงว่าที่เราสมมุติไว้ข้างต้นว่าไม่มีเวลาที่คาบเกี่ยวนั้นค่อนข้างไม่แน่นอน

6.3 หลักทั่ว ๆ ไปของการตรวจจับจุดคอขวดของเครือข่าย

สมมุติว่าอินเทอร์รัปต์ มักจะเกิดตอนที่โพรเซสเซอร์อยู่ในสถานะพัก การสมมุตินี้ไม่ใช่เป็นการสมมุติที่ดีในทุกกรณี แต่ในการทดลองนี้จะใช้สมมุติฐานเช่นนี้

ทางด้านเซิร์ฟเวอร์นั้นผลประโยชน์ของโพรเซสเซอร์รวม เท่ากับ 14.324 % และผลประโยชน์ของโพรเซสเซอร์ของกระบวนการของเซิร์ฟเวอร์เท่ากับ 9.370 % ทำการลบค่าผลประโยชน์ของโพรเซสเซอร์รวม ด้วยค่าผลประโยชน์ของโพรเซสเซอร์ของเซิร์ฟเวอร์ จะได้ว่าการเกิดอินเทอร์รัปต์ ใช้เวลา 0.04954 วินาที และเนื่องจากมี 575.017 อินเทอร์รัปต์ต่อวินาที เมื่อเราทำการหาร 0.04954 ด้วยจำนวนนี้แล้วจะได้ 0.00008615 วินาที หรือ 86.15 ไมโครวินาทีต่อการอินเทอร์รัปต์ 1 ครั้ง

ทางด้านของไคลเอ็นท์นั้นพบว่ามีผลประโยชน์ของโพรเซสเซอร์เท่ากับ 68.807 % , มีของกระบวนการ Ideagen 22.006 % ทำให้ได้เวลาในการเกิดอินเทอร์รัปต์เท่ากับ 0.46801 วินาที แล้วทำการคำนวณเหมือนเดิมจะได้ 878.89 ไมโครวินาทีต่อการอินเทอร์รัปต์ 1 ครั้ง เหตุที่ค่านี้มีค่ามากเพราะทาง

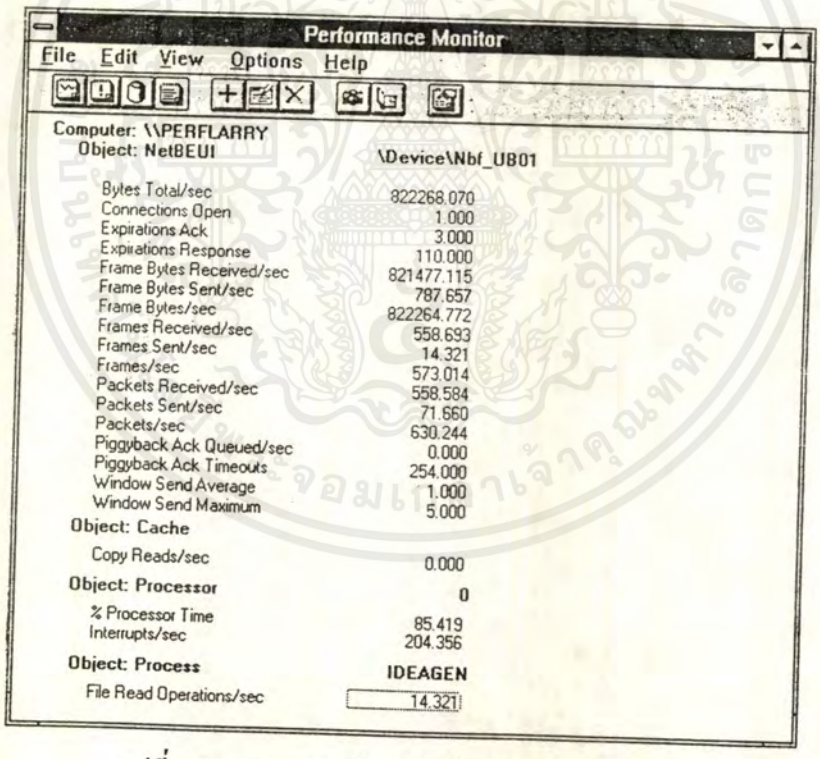
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านไคลเอ็นท์นั้น ข้อมูลจะต้องถูกคัดลอกไปยังบัฟเฟอร์ของแอปพลิเคชั่น ในขณะที่ทางเซิร์ฟเวอร์ข้อมูลสามารถถูกอ่านไปบนเครือข่ายได้โดยตรงจากแคช

เวลาในการเกิดอินเทอร์รัปต์นี้ส่วนใหญ่ไม่ได้เกิดในอินเทอร์รัปต์แฮนเดรอร์ (interrupt handler) ซึ่งทำให้อินเทอร์รัปต์ที่มีลำดับความสำคัญต่ำไม่ได้รับการตอบรับเป็นเวลานาน ในสถาปัตยกรรมการอินเทอร์รัปต์ของ Windows NT จะอนุญาตให้ทำงานได้ในระหว่างอินเทอร์รัปต์แฮนเดรอร์ โดยจัดงานนั้นอยู่ในระดับอีกระดับหนึ่งระหว่างอินเทอร์รัปต์และเธร็ด เรียกว่า deferred procedure call หรือ ระดับ DPC interrupt handler จะนำงานนั้นไปเข้าคิวเป็น DPC packet

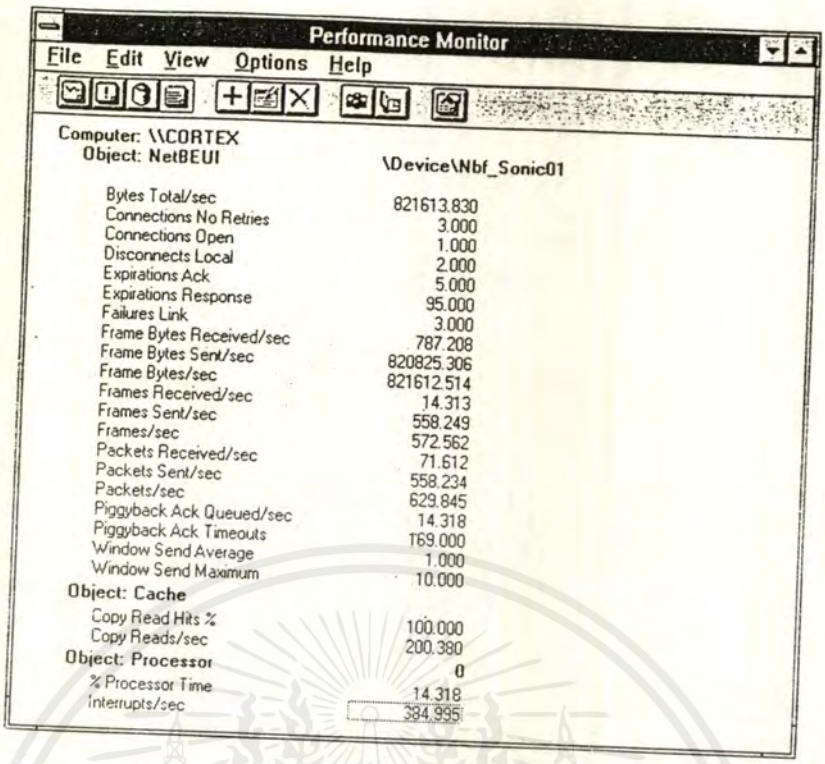
เมื่อไม่มีอินเทอร์รัปต์เหลือที่จะให้บริการแล้ว ระบบก็จะมุ่งไปที่ DPCs เพื่อที่จะทำการเอ็กซีคิวต่อไป โดยการเอ็กซีคิว DPC นี้จะมีลำดับความสำคัญต่ำกว่าอินเทอร์รัปต์ ดังนั้นในระหว่างเอ็กซีคิว DPCs อยู่ก็อนุญาตให้มีอินเทอร์รัปต์เกิดขึ้นได้ แต่ก็ไม่มีการเอ็กซีคิวเธร็ดจนกว่าจะเอ็กซีคิว DPCs ที่เหลือจนหมดก่อน การออกแบบนี้ทำให้ Windows NT มีระบบตอบสนองต่อการอินเทอร์รัปต์ที่ดี ทำให้อัตราการเกิดอินเทอร์รัปต์มีค่าสูงขึ้น

ต่อไปดูที่ทางด้านขวาของรูปที่ 6-1 และดูว่าผลลัพธ์ได้เปลี่ยนแปลงไปอย่างไร เราได้เลือกการส่งข้อมูลแบบ 14 เพจเพราะมีธรูพุทสูงสุด แม้ว่าทุกกรณีของทางด้านขวาในรูป 6-1 จะใกล้เคียงกับค่าสูงสุด



รูปที่ 6-5 มุมมองของไคลเอ็นท์ของการอ่านทีละ 14 เพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-6 มุมมองของเซิร์ฟเวอร์ของการอ่านทีละ 14 เพจ

ดูที่ไคลเอ็นท์ File Read Operations / sec จะได้ 14.321 reads / sec ในทางกลับกันจะได้ 0.069828 seconds / read

ทางด้านเซิร์ฟเวอร์ใช้โพรเซสเซอร์ 14.318 % หารจำนวนนี้ด้วย File Read Operations / sec ของไคลเอ็นท์จะได้ 0.0099979 seconds / read ส่วนทางด้านไคลเอ็นท์นั้นใช้โพรเซสเซอร์ 85.419 % หรือ 0.059646 seconds / read ต่อไปคิดเวลาของสื่อในเครือข่าย ทำการหาร NetBEUI : Bytes Total / sec ของไคลเอ็นท์ด้วย 14.321 ได้ 57417 bytes / read ซึ่งมากกว่า 57344 requested / read อยู่ 73 byte และคูณ 57417 ด้วย เวลาในการส่งข้อมูลของ Ethernet ที่ได้กล่าวถึงก่อนหน้านี้ จะได้ 0.0459336 seconds / read สุดท้ายทำการรวมเวลาของไคลเอ็นท์, เซิร์ฟเวอร์ และสื่อจะได้ 0.115578 วินาที ซึ่งมีค่ามากกว่าค่าที่เราคำนวณจากการกลับเศษส่วนค่า reads / sec คือ 0.069828 เพราะโปรโตคอลที่มีประสิทธิภาพ คือ มีหลายแพ็คเก็ต (packet) ที่จะส่งต่อการอ่าน 1 ครั้ง ที่อนุญาตให้มีเวลาคาบเกี่ยวกันระหว่างเวลาการคำนวณบนเซิร์ฟเวอร์ , เวลาในการส่งข้อมูล และเวลาในการคำนวณของไคลเอ็นท์ ดังนั้นการส่งผ่านข้อมูลขณะนี้จะแยกออกเป็น 558.693 / 14.321 = 39.012 เฟรม (ให้นึกถึงกฎข้อที่ 9 และคิดเป็น 39 เฟรม)

ต่อไปพิจารณาจากการส่งข้อมูลที่มีขนาดใหญ่กว่ามีผลกระทบอย่างไรกับเวลาต่อการอินเทอร์รัปต์ 1 ครั้ง โดยตั้งสมมุติฐานเหมือนเดิมว่าการอินเทอร์รัปต์เกิดขึ้นเมื่อเซิร์ฟเวอร์และกระบวนการ Ideagen กำลังอยู่ในสถานะพักบนเซิร์ฟเวอร์ มี 384.995 interrupts / sec และรู้ว่าใช้โพรเซสเซอร์ไป 14.318 % ลบด้วย 6.232 % ที่ใช้ไปในกระบวนการของเซิร์ฟเวอร์จะได้ 8.086 % ในการอินเทอร์รัปต์แล้วหารค่านี้ด้วยค่า interrupt / sec จะได้ 0.00021003 วินาที ซึ่งมากกว่า 2 เท่าของ 0.00008615 วินาที ซึ่งเป็นเวลาของการอินเทอร์รัปต์ของเซิร์ฟเวอร์ ในการส่งข้อมูลทีละ 2048 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางด้านไคลเอ็นท์มีเวลาของโพรเซสเซอร์ 85.419 % ซึ่งรวม 3.892 % ของกระบวนการ Ideagen ด้วยจะได้ 0.81527 วินาที ของแต่ละวินาทีที่อยู่ใน interrupt handler เนื่องจากไคลเอ็นท์มี 204.356 Interrupt / sec ทำการคำนวณเหมือนเดิม จะได้ 0.0039895 หรือ 3.99 milliseconds / interrupt ซึ่งมากกว่าค่า 878.891 microseconds / interrupt ในกรณีของ 2048 ไบต์ ถึง 4.5 เท่า

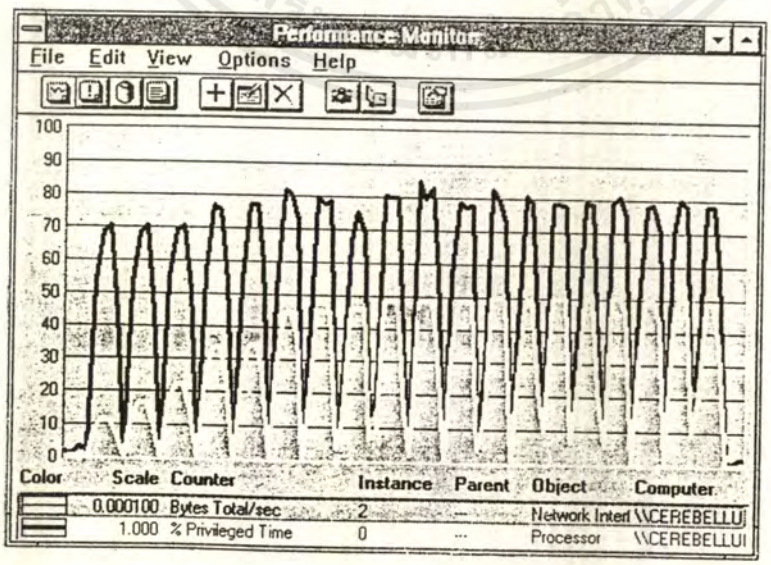
6.4 การดูประสิทธิภาพของ TCP/IP

ถ้าเป็นโปรโตคอล TCP/IP จะต้องทำการติดตั้ง SNMP ก่อนจึงจะเห็นเคาน์เตอร์ของ TCP/IP ใน Performance Monitor ถ้ามีการติดตั้งหลายโปรโตคอลแล้วไคลเอ็นท์จะต้องมี TCP/IP เป็นโปรโตคอลที่มีลำดับความสำคัญสูงที่สุด หรือจะต้องเป็นเพียงโปรโตคอลเดียวบนเซิร์ฟเวอร์ โปรโตคอลที่มีลำดับความสำคัญสูงที่สุดนั้นจะเป็นโปรโตคอลแรกที่เครื่องเวอร์กสเตชั่น (workstation) ใช้ เมื่อมันพยายามที่จะทำการติดต่อ

การตั้งลำดับความสำคัญของโปรโตคอล

1. ใน Control Panel , เลือก Network option
2. ใน Network Settings dialog box , เลือก Bindings
3. ใน Network Bindings dialog box , เลือก Workstations ใน Show Bindings For box.
   โปรโตคอลปัจจุบันที่กำลังใช้อยู่บนเวอร์กสเตชั่นจะมีรายการอยู่ในกรอบสี่เหลี่ยมใหญ่โดยโปรโตคอลบนสุดจะมีลำดับความสำคัญสูงที่สุด
4. ในการเปลี่ยนลำดับความสำคัญของโปรโตคอล ให้เลือกชื่อโปรโตคอลนั้นและใช้ปุ่มลูกศรทางขวาของ dialog box ในการเคลื่อนย้ายมันขึ้นหรือลงภายในรายการ
5. เมื่อเสร็จเรียบร้อยแล้วให้เลือก OK

ในรูป 6-11 เป็นชาร์ตแสดงธรูพุท ของ TCP/IP จากมุมมองของไคลเอ็นท์ และอีกครั้งเมื่อเราข้ามผ่านแคชไป บนด้านของไคลเอ็นท์ และกำลังอ่านจำนวน 512- , 1024- ,2048- และ 4096-ไบต์ และเมื่อขนาดเพจเพิ่มโดยเพิ่มขนาดเรคคอร์ด จนถึง 16 เพจ

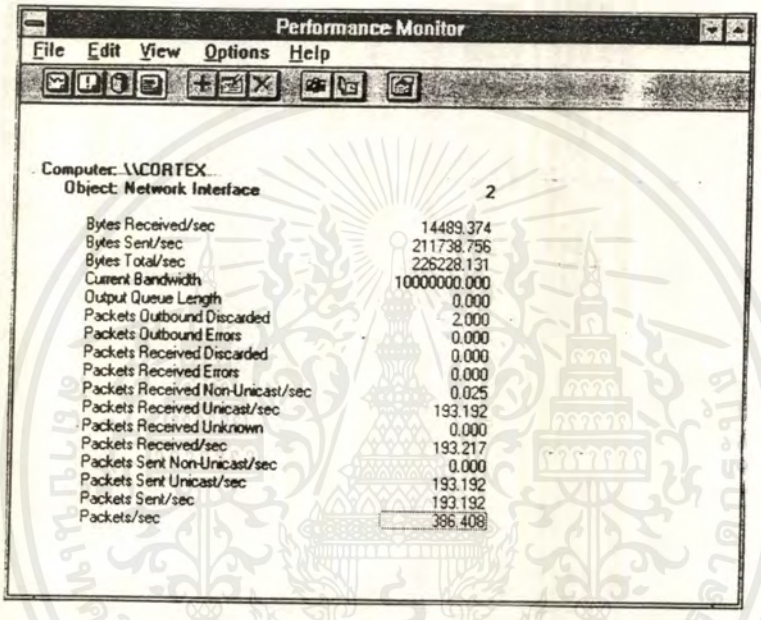


รูปที่ 6-7 ชาร์ตแสดงธรูพุทของ TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราไม่ได้ธรูพุทของ TCP/IP เหมือนกับ NetBEUI ในรูปที่ 6-1 TCP/IP จะใช้การส่งแบบต่อเนื่อง (stream) ซึ่งอนุญาตให้มีการพัฒนาโปรโตคอลใหม่ที่สามารถเคลื่อนย้าย (portable) ได้ แม้ว่าจะเกิดโอเวอร์เฮดของโพรเซสเซอร์ สิ่งนี้เป็นความสะดวกของการติดต่อสื่อสารโดยใช้ TCP/IP โดยไม่ต้องเสียเวลาหลายปีในการรอการพัฒนา , การแก้ไข (debug) และการติดตั้ง (implementation)

เคาน์เตอร์ของ TCP/IP ใน Performance Monitor สนับสนุน Management Information Base II (MIB II) สำหรับใช้กับโปรโตคอลในอินเตอร์เน็ตที่ใช้ TCP/IP เป็นพื้นฐาน ดังนั้นเราจึงมีความคิดที่จะดูในรายละเอียดของการอ่าน 2048-ไบต์ (เหมือนกับที่ได้ทำกับ NetBEUI) อันดับแรกเราดูที่ Network Interface ซึ่งเป็นสื่อที่ใกล้เคียงที่สุดที่เราสามารถดูเคาน์เตอร์ของ TCP/IP ได้

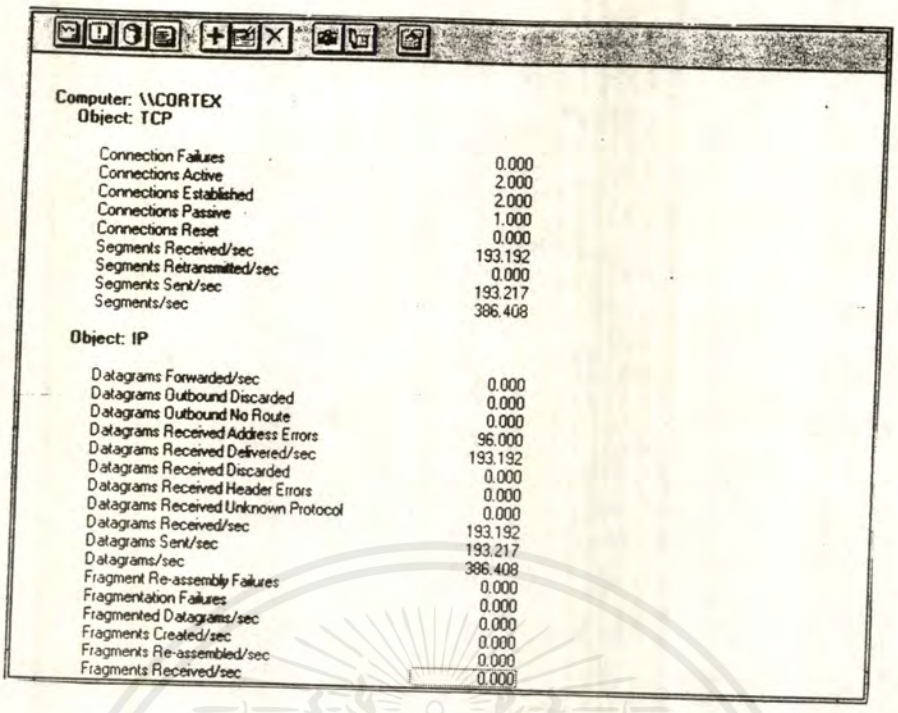


รูปที่ 6-8 Network Interface ของ TCP/IP ในการอ่าน 2048 ไบต์ (ด้านเซิร์ฟเวอร์)

ทำการหาร Bytes Received/sec และ Bytes Sent/sec ด้วย Packets Sent/sec พบว่าแต่ละแพ็คเก็ตมี 75 ไบต์ทางด้านรับ และ 1096 ไบต์ทางด้านส่ง ค่าของ Bytes Total/sec ต่ำกว่าค่า Bytes Total/sec สำหรับ NetBEUI ค่า Current Bandwidth นั้นคงที่สำหรับ Ethernet ซึ่งวัดในหน่วย bits/sec คำว่า "Unicast" หมายความว่า แพ็คเก็ตนั้นถูกส่งมายังคอมพิวเตอร์นี้โดยเฉพาะ (ซึ่งตรงกันข้ามกับ broadcast คือ ส่งไปยังทุกเครื่องบนเครือข่ายย่อย หรือ multicast จะส่งไปยังหลาย ๆ เครื่องบนเครือข่ายย่อย) บางครั้งโปรโตคอลจะทิ้งแพ็คเก็ตไป แม้ว่าจะไม่มีความผิดพลาดเกิดขึ้น แต่ต้องการทำให้บัฟเฟอร์ว่าง เป็นต้น errors หมายถึง แพ็คเก็ตที่มีปัญหาไม่สามารถส่งไปยังโปรโตคอลระดับสูงกว่าได้ (หมายถึง IP) Output Queue Length หมายถึงจำนวนแพ็คเก็ตปัจจุบันที่ค้างอยู่และคอยการส่งออกไป

จากรูป 6-11 การใช้โพรเซสเซอร์ สำหรับ TCP/IP นั้นสูงกว่า NetBEUI ซึ่งเราก็คาดไว้ว่าจะเป็นเช่นนี้ เพราะเป็นการส่งแบบต่อเนื่อง ส่วนอัตราการอินเทอร์รัปต์อยู่ในระดับปานกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Computer: \\CORTEX
Object: TCP

Connection Failures	0.000
Connections Active	2.000
Connections Established	2.000
Connections Passive	1.000
Connections Reset	0.000
Segments Received/sec	193.192
Segments Retransmitted/sec	0.000
Segments Sent/sec	193.217
Segments/sec	386.408

Object: IP

Datagrams Forwarded/sec	0.000
Datagrams Outbound Discarded	0.000
Datagrams Outbound No Route	0.000
Datagrams Received Address Errors	96.000
Datagrams Received Delivered/sec	193.192
Datagrams Received Discarded	0.000
Datagrams Received Header Errors	0.000
Datagrams Received Unknown Protocol	0.000
Datagrams Received/sec	193.192
Datagrams Sent/sec	193.217
Datagrams/sec	386.408
Fragment Re-assembly Failures	0.000
Fragmentation Failures	0.000
Fragmented Datagrams/sec	0.000
Fragments Created/sec	0.000
Fragments Re-assembled/sec	0.000
Fragments Received/sec	0.000

รูปที่ 6-9 เคาน์เตอร์ของ TCP/IP ของเซิร์ฟเวอร์ในการอ่าน 2048 ไบต์

ชั้นที่สูงขึ้นไปเหนือ Network Interface คือ IP (Internet Protocol) โดยชั้นนี้จะเห็นเฉพาะดาต้าแกรม ส่วนโปรโตคอลระดับสูงกว่านี้ (ในกรณีนี้ คือ TCP) จะมีการตรวจสอบความถูกต้องระหว่างจุดปลายด้วย เพื่อให้มั่นใจว่าไม่มีการผิดลำดับของแพ็คเก็ตที่ชั้นนี้ จะมีการนับการกำหนดดาต้าแกรมที่แตกต่างกันมากมาย ถ้ามีปัญหาเรื่องความถูกต้องของข้อมูล ก็สามารถตรวจจับมันได้ที่นี่

Datagrams Forwarded/sec คือ อัตราที่แสดงว่าโหนด (node) นี้เป็นไอพีเกตเวย์ (IP gateway) ที่ได้รับแพ็คเก็ตที่ส่งมาแต่ไม่ได้รับการให้ที่อยู่ ที่โหนดนี้ และรวมถึงการไม่มีเส้นทางและการทิ้งดาต้าแกรมด้วย (DataGrams Outbound No Route)

Transmission Control Protocol (TCP) มีการติดต่อระหว่างจุดปลายโดยใช้ IP และเพื่อให้มั่นใจว่าทุกแพ็คเก็ตได้ถูกส่งไปแล้ว ถ้ามันไม่ถูกส่งจะมีการส่งไปอีกครั้งหนึ่ง ออบเจ็กต์นี้จะเป็นการดูจำนวนแพ็คเก็ตที่ถูกส่งไปและได้รับมา

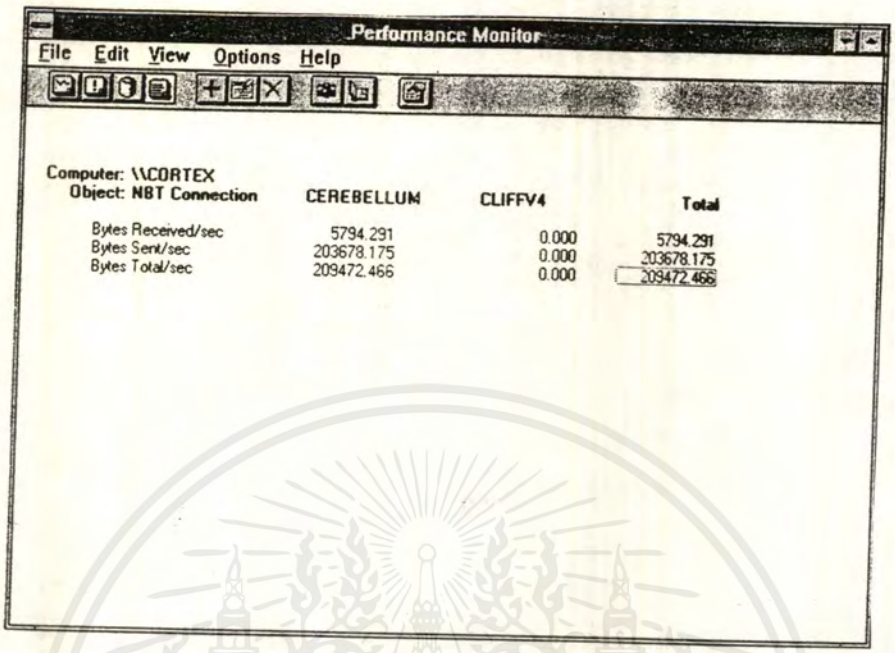
User Datagram Protocol (UDP) ให้ข้อมูลการติดต่อโดยตรงและอย่างรวดเร็วกับ IP โดยไม่ต้องมีการร้องขอการติดต่อกับด้านรับก่อน อย่างไรก็ตามการส่งแพ็คเก็ตนั้นอาจจะไม่เป็นลำดับ หรือมีการส่งซ้ำ หรือแพ็คเก็ตอาจสูญหายได้ โดยปกติแล้วแอปพลิเคชั่นจะใช้ TCP หรือ UDP ก็ได้เพื่อติดต่อสื่อสารกับ IP ดังนั้นเคาน์เตอร์เหล่านี้อาจจะเป็นศูนย์ทั้งหมดก็ได้ในการทดลองนี้

The Internet Control Message Protocol (ICMP) เป็นชั้นของโปรโตคอลที่อยู่เหนือ IP มันจัดการกับงานการส่งข้อความพิเศษ คือ :

- ทำการส่งข้อความเพื่อตรวจสอบว่าการติดต่อนั้นใช้ได้ (ใช้ **ping** utility)
- ทำการรีไดเร็กโหนด (redirect node) เพื่อจะได้ใช้เส้นทางที่เหมาะสม
- แนะทิศทางให้โหนด เพื่อลดอัตราการส่งข้อมูล เพื่อบรรเทาความคับคั่งบนเครือข่าย (source quench)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการส่งข้อความว่าไม่สามารถไปถึงปลายทางได้ ถ้าไม่สามารถส่งดาต้าแกรมตามที่ถูกร้องขอมาได้ เคาน์เตอร์เหล่านี้เป็นศูนย์ทั้งหมดในกรณีนี้ ดังนั้นเราจึงไม่ได้แสดงไว้ในที่นี้

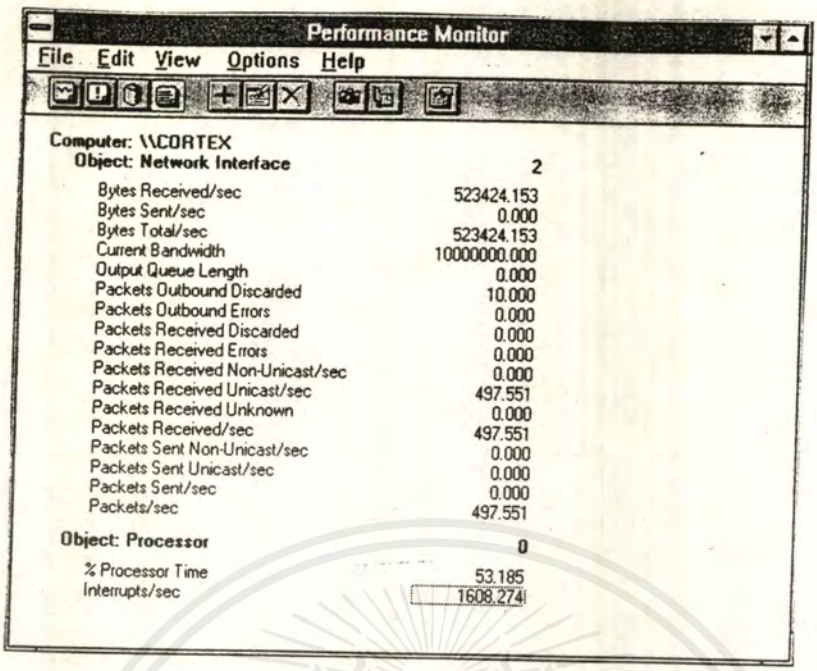


รูปที่ 6-10 สถิติของ NBT ของเซิร์ฟเวอร์ในการอ่าน 2048 ไบต์

ออบเจ็กต์ NetBEUI TCP/IP Connection (NBT connection) นั้นจะบันทึกการติดต่อสื่อสารไปยังทุก ๆ จุด และเป็นประโยชน์มากบนเซิร์ฟเวอร์ที่ยุ่งมาก ๆ NBT connection จะอยู่บนชั้น TCP ในสแตก (stack) ของโปรโตคอล NBT จะนับอัตราของไบต์ที่เข้ามาและที่ส่งออกไป และมี 1 อินสแตนซ์ สำหรับแต่ละติดต่อ และมีค่ารวมสำหรับทุกการติดต่อ สามารถใช้มันในการดูว่าการติดต่อใดเป็นแหล่งของโหลด (load) และเป็นไปในทิศทางใด

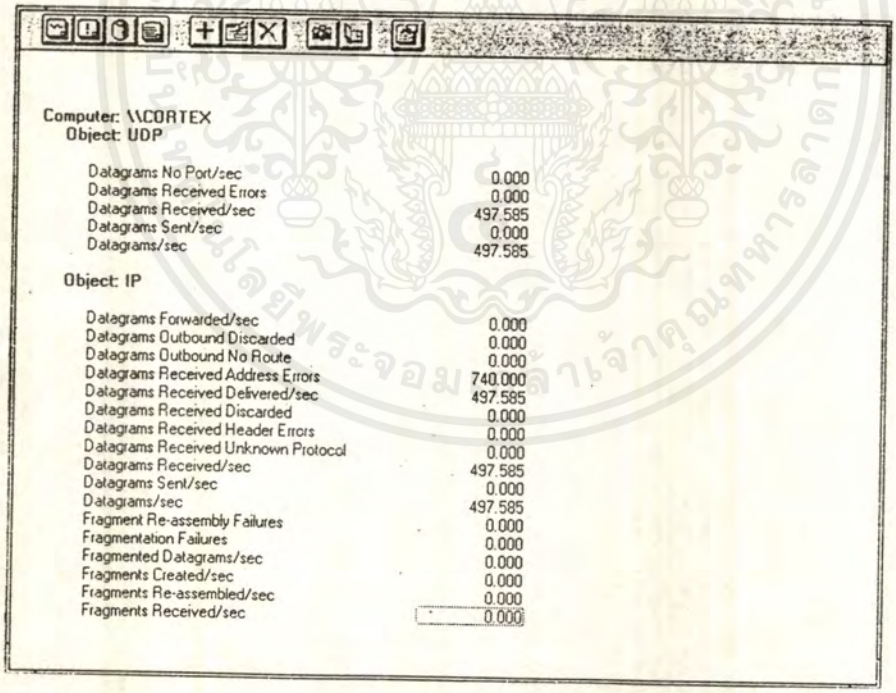
เราจะทำการสร้างให้มีการรับส่ง UDP ขึ้นมาเล็กน้อย เพื่อจะได้พอมีข้อมูลอยู่บ้าง ใน 2 รูปถัดไป แสดงการกระทำของ UDP จากมุมมองทางด้านเซิร์ฟเวอร์ เรากำลังส่งการเขียน 1024-ไบต์ จากกระบวนการบนไคลเอ็นท์ ไปยังกระบวนการบนเซิร์ฟเวอร์ โดยใช้ Windows Sockets (WinSock) ในการติดต่อ UDP ในกรณีนี้ WinSock จะให้ความถูกต้องของแพ็คเก็ตระหว่างจุดปลายและดังนั้นสามารถใช้ UDP ซึ่งไม่น่าเชื่อถือได้ สิ่งนี้ไม่ได้อ้างถึงกระบวนการบนเซิร์ฟเวอร์ทั้งหมด แต่กระทำเหมือนกับแอปพลิเคชั่น เช่น SQL เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-11 สถิติของ Network Interface ของเซิร์ฟเวอร์ในการเขียนขนาด 1024 ไบต์ไปยังเซิร์ฟเวอร์

มันเร็วมากสำหรับเรคคอร์ดขนาดเท่านี้ และอัตราการอินเทอร์รัปต์นั้นสูงมากเช่นกัน ซึ่งมีผลทำให้มีการใช้โพรเซสเซอร์มากด้วย



รูปที่ 6-12 สถิติของ UDP/IP ของเซิร์ฟเวอร์ในการเขียน 1024 ไบต์ไปยังเซิร์ฟเวอร์

จากรูปไม่ต้องกังวลกับ 740 IP : Datagrams Received Address Errors มันเกิดขึ้นในระหว่างการเตรียมการทดลองของการทดลองนี้ และเพราะว่ามันเป็นการนับ ณ เวลาหนึ่งของจำนวนข้อผิดพลาดปัจจุบัน เราจึงต้องดูชาร์ตในระหว่างการทดลอง ถ้ามันมีค่าไม่เปลี่ยนแปลงในขณะทำการทดลอง แสดงว่าข้อผิดพลาดที่เกิดขึ้นนั้นเกิดก่อนการทดลองนี้ เมื่อเราตรวจสอบแล้ว ก็เป็นเช่นนั้นจริง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

หลักการวิเคราะห์

7.1 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากโพรเซสเซอร์ มีดังนี้

7.1.1 % Processor Time

เพื่อที่จะดูปริมาณการใช้โพรเซสเซอร์ในกระบวนการว่ามีการทำงานเกิดขึ้นมากน้อยเพียงใด ถ้ามีงานที่กำลังทำอยู่และโพรเซสเซอร์ถูกหยุดพักอยู่ ก็สามารถมั่นใจได้ว่ามีออบเจ็กต์อื่น ๆ สามารถที่จะทำให้เกิดความล่าช้า เสียเวลา

ถ้า % Processor Time มีค่ามากกว่า 80% แสดงว่า โปรเซสเซอร์มีการใช้งานสูง

7.1.2 % Total Processor Time

ถ้าเป็นระบบที่มีโพรเซสเซอร์หลายตัว ควรจะวัดค่านี้ด้วย เนื่องจากว่าค่า % Total Processor Time นี้ก็คือค่าของ % Processor Time ในการใช้โพรเซสเซอร์เอ็กซีคิวงานของแต่ละตัว แล้วนำมารวมกันเพื่อหาค่าเฉลี่ย

7.1.3 Processor Queue Length (ในออบเจ็กต์ System)

เนื่องจากเราพบว่า % Processor Time อาจไม่ได้แสดงจุดคอขวด ซึ่งมี 2 เหตุผล คือ

- มีบางกระบวนการซึ่งอาจมีการเอ็กซีคิวที่เร็วมากจนไม่ได้มีการบันทึกการใช้โพรเซสเซอร์
- มีบางกระบวนการในคิว ซึ่งอาจไม่เคยได้รับการเอ็กซีคิวเลย เนื่องจากว่ามีบางกระบวนการ ที่มีลำดับความสำคัญสูงกว่า ใช้โพรเซสเซอร์อยู่

Processor Queue Length เป็นเคาน์เตอร์ที่บ่งบอกว่ามีเทรดจำนวนเท่าไรในคิวของโพรเซสเซอร์ ที่กำลังแข่งขันกันเพื่อจะให้ได้ใช้โพรเซสเซอร์

7.2 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากดิสก์ มีดังนี้

7.2.1 % Disk Time (ทั้งฟิสิคอลดิสก์และลอจิคอลดิสก์)

เคาน์เตอร์ตัวนี้จะเป็นตัวบ่งบอกว่าระบบดิสก์นั้นมีความคล่องแคล่วว่องไวหรือมีความคล่องตัวหรือไม่อย่างไร ถ้ามีการเพจจิ้งมากเกินไป เคาน์เตอร์ตัวนี้ก็จะแสดงค่าการใช้ดิสก์ที่มีค่าสูงมาก และยังแสดงค่าเวลาที่สูญเสียไปในการบริการการร้องขอการอ่านหรือการเขียน

7.2.2 Avg. Disk Bytes/Transfer

ซึ่งแสดงขนาดของค่าเฉลี่ยในการโอนย้ายข้อมูลในแต่ละครั้ง ถ้าเคาน์เตอร์ตัวนี้มีค่าสูงก็จะแสดงว่าการเคลื่อนย้ายข้อมูลมีประสิทธิภาพสูงด้วย

7.2.3 Avg. Disk sec/Transfer

เพื่อใช้แสดงเวลาในการโอนย้ายข้อมูล ซึ่งถ้าใช้เวลาน้อยก็จะแสดงว่าดิสก์นี้มีประสิทธิภาพ แต่ค่านี้ก็จะขึ้นอยู่กับขนาดของเรคคอร์ดในการเคลื่อนย้ายข้อมูลด้วย ซึ่งถ้าเรคคอร์ดนี้มีขนาดใหญ่ก็จะแสดงว่ามีประสิทธิภาพจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.4 Disk Bytes/sec

ค่านี้นับว่าเป็นธรูพุทในการโอนย้ายข้อมูลของดิสก์ เพื่อดูว่ามีการโอนย้ายข้อมูลจำนวนกี่ไบต์ใน 1 วินาที ซึ่งค่านี้ก็จะต้องดูควบคู่กันไปกับ Avg. Disk sec/Transfer ด้วย

7.3 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากหน่วยความจำ มีดังนี้

7.3.1 Pages/sec

ค่า Pages/sec คือ ผลรวมของ Pages Input/sec และ Pages Output/sec ค่า Pages/sec นี้จะบอกถึงปริมาณเพจที่จะต้องเขียนหรืออ่านจากดิสก์และบรรจุเข้าไปในหน่วยความจำฟิสิคอล ค่านี้จะบ่งบอกว่าระบบต้องการหน่วยความจำฟิสิคอลเพิ่มขึ้นหรือไม่

ถ้าค่านี้มีค่าเกิน 5 pages/sec สำหรับดิสก์อันเดียว ก็หมายความว่าอาจจะเกิดคอขวดที่หน่วยความจำได้

7.3.2 Available Bytes

ค่า Available Bytes หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ถ้าค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ Windows NT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอปพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวลไว้

ถ้าค่านี้มีค่าต่ำกว่า 1MB อาจจะทำให้ประสิทธิภาพของหน่วยความจำต่ำลง

7.3.3 Commit Limit

Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิต (commit) ได้โดยไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับ ขนาดของ Pagingfile.sys รวมกับขนาดของ RAM ที่สามารถกลับไปยังดิสก์ได้ (หรือจำนวนของ RAM ที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)

7.3.4 Committed Bytes

Committed Bytes คือ จำนวนของหน่วยความจำ (ไบต์) ที่แอปพลิเคชั่นโปรแกรมต่าง ๆ ใช้ในการวิ่งโปรแกรม

ค่า Committed Bytes นี้ควรมีค่าน้อยกว่า ค่า Commit Limit

7.3.5 Pool Nonpaged Bytes

Pool Nonpaged Bytes คือ จำนวนของไบต์ในนอนเพจพูล, ค่านี้จะหามาได้จากการที่งานต่าง ๆ ได้ทำงานเสร็จสิ้นลงแล้ว ค่านอนเพจนี้จะไม่สามารถดึงออกไปเป็นค่าของเพจจิ้งไฟล์ (paging file) ได้ แต่จะยังเป็นค่าที่เก็บรักษาไว้ในหน่วยความจำหลักเท่ากับจำนวนที่มันได้จองไว้

ค่านี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอลลบด้วยจำนวน 4 MB

7.3.6 Logical Disk object

มักจะดูเทียบกับออบเจ็กต์นี้เพราะถ้ามีการติดต่อกับดิสก์มาก มักจะแสดงว่ามีหน่วยความจำไม่เพียงพอ

7.3.6.1 ดูจำนวนเพจที่อ่านได้ต่อวินาทีของหน่วยความจำและลอจิคอลดิสก์ ถ้ามีค่าใกล้เคียงกันแสดงว่าเราสามารถตรวจจับจุดคอขวดได้ ตามกฎข้อที่ 9 โดยค่าเคาน์เตอร์ที่ต้องดูมีดังนี้

Object : Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pages Input/sec

Object : Logical Disk

Avg. Disk Bytes/Read

Disk Reads/sec

จากค่า Avg. Disk Bytes/Read เราสามารถหาจำนวนเพจต่อการอ่าน 1 ครั้งได้ เพราะเราทราบว่า 1 เพจมี 4K ไบต์ เมื่อได้จำนวนเพจต่อการอ่าน 1 ครั้งแล้ว นำมาคูณกับจำนวนของการอ่านต่อวินาที (Disk Reads/sec) จะได้จำนวนเพจที่อ่านได้ใน 1 วินาที ของลอจิคอลดิสก์ แล้วนำไปเทียบกับ Pages Input/sec ของหน่วยความจำ ถ้ามีค่าใกล้เคียงกันก็จะสรุปได้ว่าเกิดปัญหาคอขวดขึ้น ตามกฎข้อที่ 9

7.3.6.2 ดูค่า Page Faults/sec เทียบกับค่า Transition Faults/sec ถ้าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำแต่กำลังถูกเขียนลงบนดิสก์ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)
ถ้าค่าทั้งสองตัวนี้มีค่าใกล้เคียงกัน อาจสรุปได้ว่ามีปัญหาคอขวดเกิดขึ้น

7.3.6.3 อาจดูค่าเทียบกันเป็นคู่ ๆ ดังนี้

- ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec
- ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec
- Page Inputs/sec * 4096 bytes/page = bytes/read เทียบกับค่า
  Page Reads/sec
  Avg. Disk Bytes/Read
- Page Outputs/sec * 4096 bytes/page = bytes/write เทียบกับค่า
  Page Writes/sec
  Avg. Disk Bytes/Write

ถ้าค่าในแต่ละคู่นั้นมีค่าเท่ากัน ก็แสดงว่าสามารถตรวจจับจุดคอขวดได้ตามกฎข้อที่ 9

7.4 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากแคช มีดังนี้

7.4.1 Copy Read Hits %

เปอร์เซ็นต์การฮิตเกิดขึ้นเมื่อมีการร้องขอข้อมูลจากระบบไฟล์และข้อมูลก็อยู่ในแคชเรียบร้อยแล้ว ซึ่งอัตราการฮิตสูง อาจเป็นตัวบอกได้ว่าขนาดของแคชมีเพียงพอ แต่อย่างไรก็ตามก็ต้องดูปริมาณการอ่านต่อวินาทีด้วย คือ ดูค่า **Copy Reads/sec** เพราะถ้ามีอัตราฮิตสูง แต่อัตราที่อ่านได้นั้นมีค่าต่ำ ก็จะไม่ได้หมายความว่ามีประสิทธิภาพสูง

7.4.2 Data Flush Pages/sec และ Data Flushes/sec

ซึ่งเป็นการดูกิจกรรมการนำข้อมูลออกของแคช ซึ่งได้รับการร้องขอจากระบบไฟล์ให้เขียนข้อมูลออกไปยังอุปกรณ์ต่อพ่วงโดยตรง

7.4.3 Data Map Hits% และ Data Maps/sec

ที่ต้องดู 2 ค่าควบคู่กันไป ก็ด้วยเหตุผลเดียวกับข้อ 7.4.1 คือ ต้องดูปริมาณของงานที่ทำได้ใน 1 วินาทีด้วย เพราะอัตราฮิตที่สูง แต่ปริมาณงานที่ทำได้นั้นต่ำ ก็อาจทำให้ประสิทธิภาพต่ำด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า Data Map นี้ จะบอกถึงกิจกรรมเกี่ยวกับไดเรกทอรี่ หรือ กิจกรรมที่เกี่ยวกับไฟล์จำนวนมาก ถ้าค่าเคาน์เตอร์นี้สูง แสดงว่ามีปัญหาเกี่ยวกับไดเรกทอรี่ อาจหมายถึง การคัดลอก ไฟล์ขนาดเล็กจำนวนมาก เป็นต้น

7.4.4รวมเคาน์เตอร์ของ Memory object กับ Cache object เข้าด้วยกัน แล้วเทียบกับ LogicalDisk object เช่น

(Memory : Pages Output/sec) + (Cache : Data Flush Pages/sec)

จะได้จำนวนเพจที่เขียนออกไปยังดิสก์ต่อวินาที แล้วคูณด้วย bytes/page (4K) จะได้จำนวน bytes/sec แล้วนำไปเทียบกับ Disk Write Bytes/sec ถ้าได้ค่าที่ใกล้เคียงกัน ก็จะสามารถตรวจจับจุดคอขวดได้ตามกฎข้อ 9 อย่างไรก็ตาม ไม่จำเป็นต้องเทียบค่าของการเขียนอย่างเดียว อาจเทียบค่าของการอ่านได้ โดยใช้วิธีเดียวกัน

7.5 เคาน์เตอร์ที่ควรพิจารณาในการตรวจจับจุดคอขวดที่เกิดจากเครือข่าย มีดังนี้

การตรวจจับจดคอขวดบนระบบแลน (LAN) จะใช้เทคนิคที่เหมือนกับระบบแวน (WAN) และออบเจ็กต์ที่ต้องทำการตรวจจับก็คือออบเจ็กต์ต่าง ๆ ที่ได้กล่าวมาแล้วข้างต้น มีออบเจ็กต์และเคาน์เตอร์ต่างๆ ที่ต้องพิจารณา ดังต่อไปนี้

7.5.1. ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface) เคาน์เตอร์ที่ควรพิจารณามีดังนี้

- Bytes Total/sec

เนื่องจากเครือข่ายที่ใช้ทดลองนี้เป็นเครือข่ายอีเทอร์เน็ต (Ethernet) ดังนั้นจึงมีอัตราการส่งเป็น 10 เมกะบิตต่อวินาที (10 Mbps.) และสามารถคิดเป็น %Utilization ของการส่งข้อมูลเป็นบิท จากค่าของ Bytes Total/sec ได้ดังนี้

จำนวน Bytes Total/sec * 8 = จำนวน Bits Total/sec

เพราะฉะนั้น $\%\text{Bit Utilization} = \frac{\text{จำนวน Bits Total/sec} * 100}{10\text{Mbps.}}$

- Packets/sec

เราสามารถหาจำนวนแพ็คเก็ตต่อการส่งใน 1 วินาทีของอีเทอร์เน็ตได้ดังนี้

จำนวนไบต์ต่อแพ็คเก็ต = 512 Bytes/packet = 512 * 8 Bits/packet

ดังนั้น $\text{Packets/sec} = \frac{10\text{ Mbps}}{512 * 8\text{ Bits/packet}} = \frac{10 * 10^6}{512 * 8}$ Packets/sec

เพราะฉะนั้น $\%\text{Packet Utilization} = \frac{\text{จำนวน Packets/sec} * 512 * 8 * 100}{10 * 10^6}$

เมื่อพิจารณาจากค่า %Utilization ทั้ง 2 ค่าควรจะมีค่าที่ไม่ใกล้เคียง 100% จึงจะไม่เกิดคอขวดที่เครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5.2. ออบเจ็กต์ : ไอพี (Object : IP) ควรพิจารณาที่เคาน์เตอร์ดังต่อไปนี้

- พิจารณาค่า Datagrams Received Delivered/sec เทียบกับค่า Datagrams Received/sec

ค่า Datagrams Received/sec เพื่อดูปริมาณจำนวน datagrams ที่ได้รับมา เทียบกับค่า Datagrams Received Delivered/sec เพื่อแสดงว่ามีการส่งดาต้าแกรมที่ได้รับมาออกไปได้จำนวนกี่ดาต้าแกรม ค่าทั้งสองนี้ควรจะมีค่าที่ใกล้เคียงกันเพื่อที่จะได้เกิดความผิดพลาดที่น้อยที่สุด

7.5.3. ออบเจ็กต์ : ทีซีพี (Object : TCP) ควรพิจารณาที่เคาน์เตอร์ดังต่อไปนี้

- Connections Failures

เพื่อดูจำนวนครั้งของความผิดพลาดทั้งการรับและส่งข้อมูลในการทำการติดต่อ

- Connections Active

Connection Active คือ จำนวนครั้งของการติดต่อ TCP เพื่อทำการส่งข้อมูล

- Connections Passive

Connection Passive คือ จำนวนครั้งของการติดต่อ TCP ในการรับข้อมูล

พิจารณาเปอร์เซ็นต์การติดต่อ (connection) ที่ไม่สำเร็จ โดยคิดเทียบกับจำนวนการติดต่อทั้งหมด (คือ ทั้งด้านรับและด้านส่ง) และจำนวนการติดต่อทั้งหมดคิดได้จาก

ค่า Connections Active (ด้านส่งข้อมูล) + ค่า Connections Passive (ด้านรับข้อมูล)

เพราะฉะนั้นเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จ $= \frac{\text{Connection Failures}}{\text{Connections Active + Connection Passive}} * 100$

ถ้าเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จนั้นน้อยมาก แสดงว่าการรับส่งข้อมูลของชั้นทีซีพีนี้มีความผิดพลาดน้อยมาก จึงไม่เกิดปัญหาขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

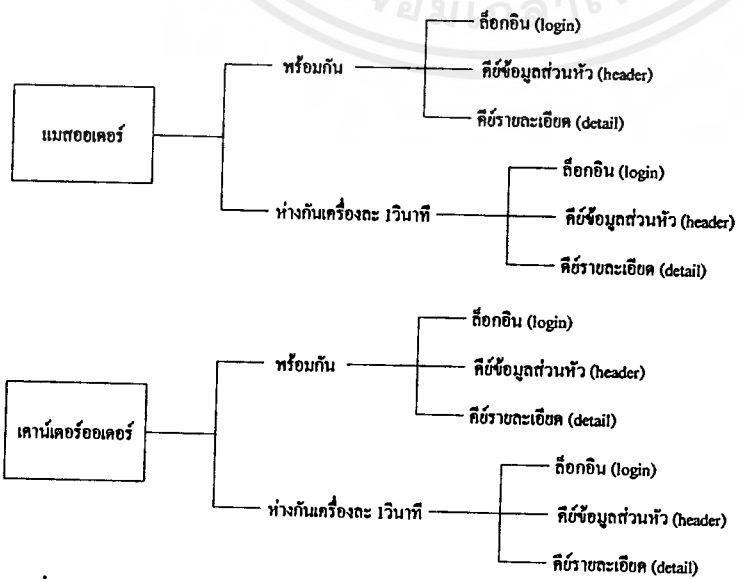
ผลการทดลอง

การทดลองนี้จะทำการตรวจจับคอขวดที่เซิร์ฟเวอร์ 2 เครื่อง คือ ไฟล์เซิร์ฟเวอร์ (File Server) และดาต้าเบสเซิร์ฟเวอร์ (Database Server) โดยการใส่โหลด (load) เป็น 2 รูปแบบ คือ วิ่งแอปพลิเคชั่นที่เกี่ยวกับใบสั่งซื้อสินค้า 2 รูปแบบ ดังนี้

1. แมสออเดอร์ (Mass Order) คือ การสั่งซื้อสินค้าเป็นจำนวนมาก เป็นการขายขาดให้กับผู้ซื้อเลย
2. เคาน์เตอร์ออเดอร์ (Counter Order) คือ การสั่งซื้อสินค้าย่อยที่จะนำไปวางจำหน่ายตามเคาน์เตอร์ในห้างสรรพสินค้าหรือร้านค้าย่อย การสั่งซื้อชนิดนี้สิทธิในสินค้านั้นจะยังเป็นของบริษัทผู้ขายอยู่จนกระทั่งมีการขายสินค้าได้จริง ๆ จึงจะมีการคิดเงินกับตัวแทนจำหน่าย

จะมีการวัดประสิทธิภาพของระบบตามขั้นตอนการคีย์ออเดอร์ 3 ขั้นตอนดังนี้

- ขั้นตอนการล็อกอิน (login) เป็นการคีย์ชื่อผู้ใช้และชื่อของคลังข้อมูลลงไป แล้วโปรแกรมจะมีการตรวจสอบสิทธิของผู้ใช้ว่ามีสิทธิเข้าใช้ฐานข้อมูลของสินค้าหรือไม่ และสามารถใช้ฐานข้อมูลได้ในขอบเขตมากน้อยเพียงใด ขั้นตอนนี้จะแบ่งเป็น 2 รูปแบบ คือ
  - เครื่องไคลเอ็นท์แต่ละเครื่องมีการล็อกอินพร้อมกัน
  - เครื่องไคลเอ็นท์จำนวน 6 เครื่องมีการล็อกอินห่างกันเครื่องละ 1 วินาที
- ขั้นตอนการคีย์ข้อมูลส่วนหัว (header) ของใบสั่งซื้อสินค้า ขั้นตอนนี้จะแบ่งเป็น 2 รูปแบบเช่นกัน คือ
  - เครื่องไคลเอ็นท์แต่ละเครื่องมีการคีย์ข้อมูลส่วนหัวพร้อมกัน
  - เครื่องไคลเอ็นท์จำนวน 6 เครื่องมีการคีย์ข้อมูลส่วนหัวห่างกันเครื่องละ 1 วินาที
- ขั้นตอนการคีย์รายละเอียด (detail) ในใบสั่งซื้อสินค้านั้น ขั้นตอนนี้จะแบ่งเป็น 2 รูปแบบเช่นกัน คือ
  - เครื่องไคลเอ็นท์แต่ละเครื่องมีการคีย์รายละเอียดของใบสั่งซื้อพร้อมกัน
  - เครื่องไคลเอ็นท์จำนวน 6 เครื่องมีการคีย์รายละเอียดของใบสั่งซื้อห่างกันเครื่องละ 1 วินาที

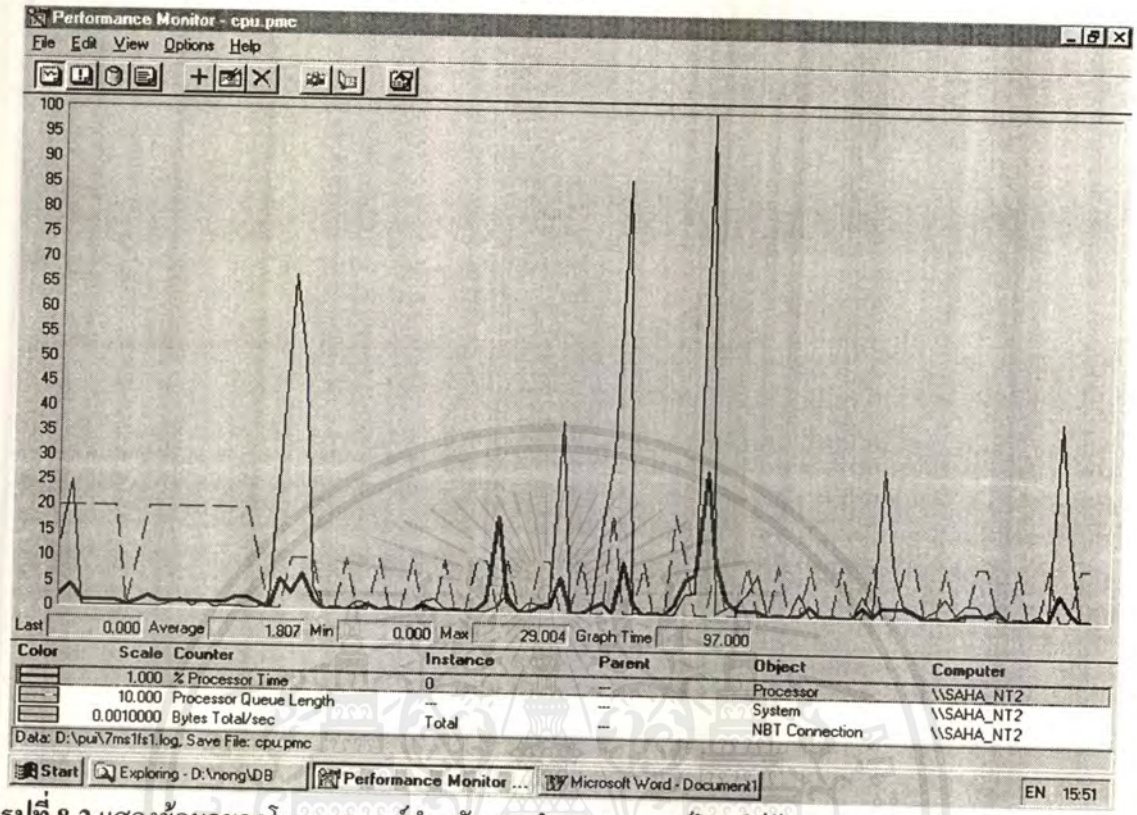


รูปที่ 8-1 แอปพลิเคชั่นที่เกี่ยวกับใบสั่งซื้อสินค้าทั้ง 2 รูปแบบ

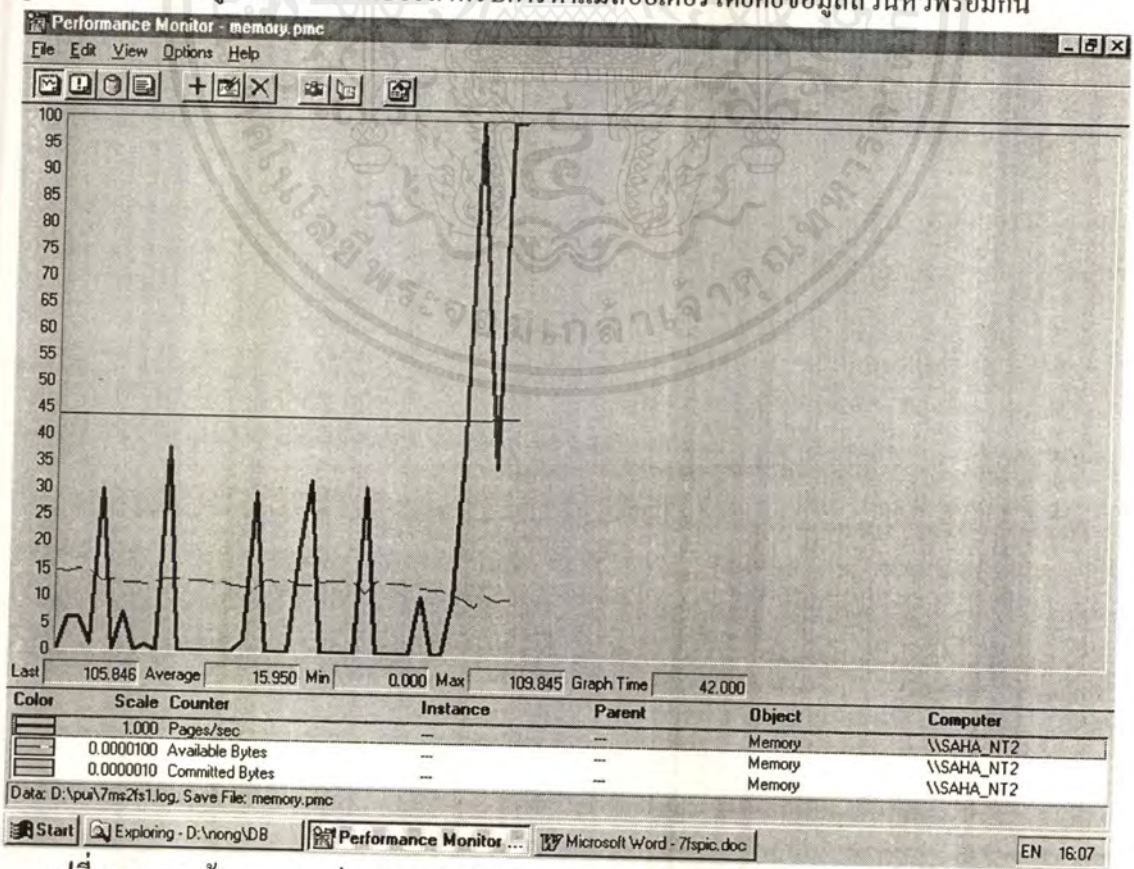
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์เซิร์ฟเวอร์

1. แมสออเดอร์ คีย์ข้อมูลส่วนหัวพร้อมกัน ได้ผลดังนี้

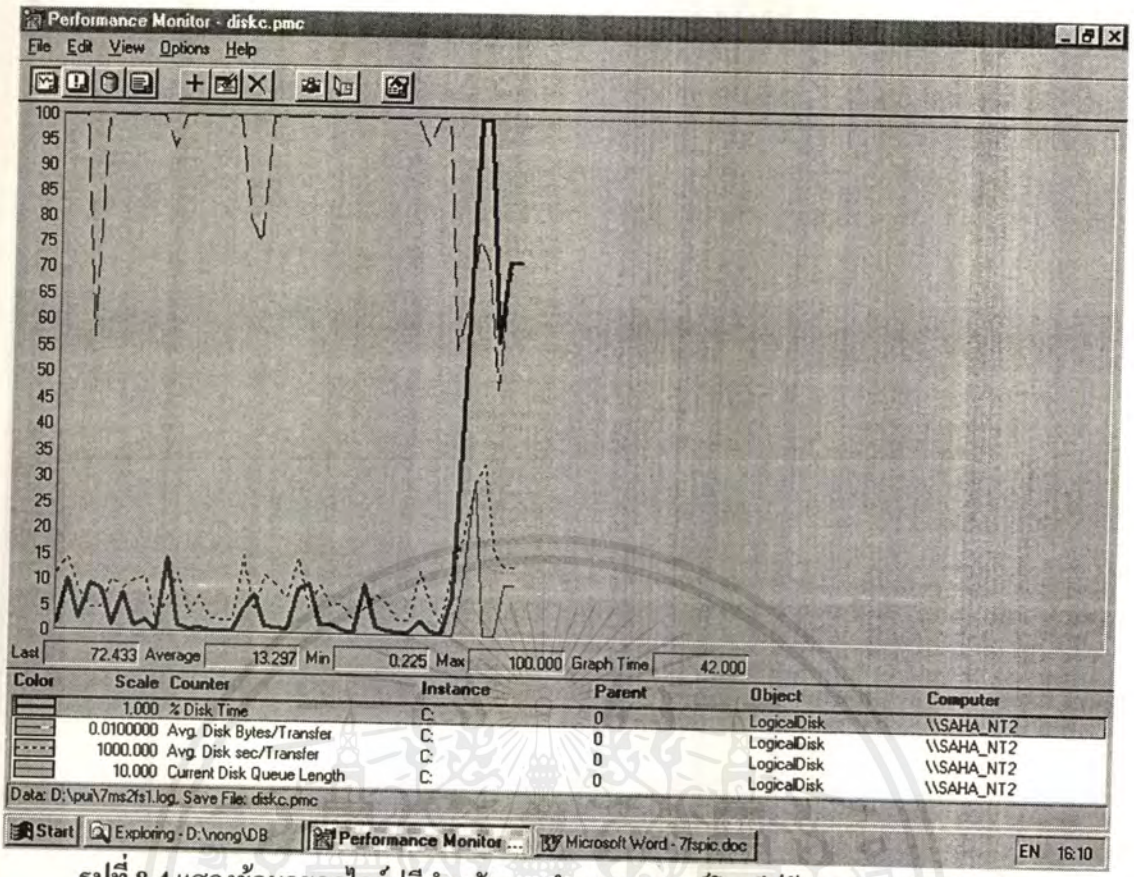


รูปที่ 8-2 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

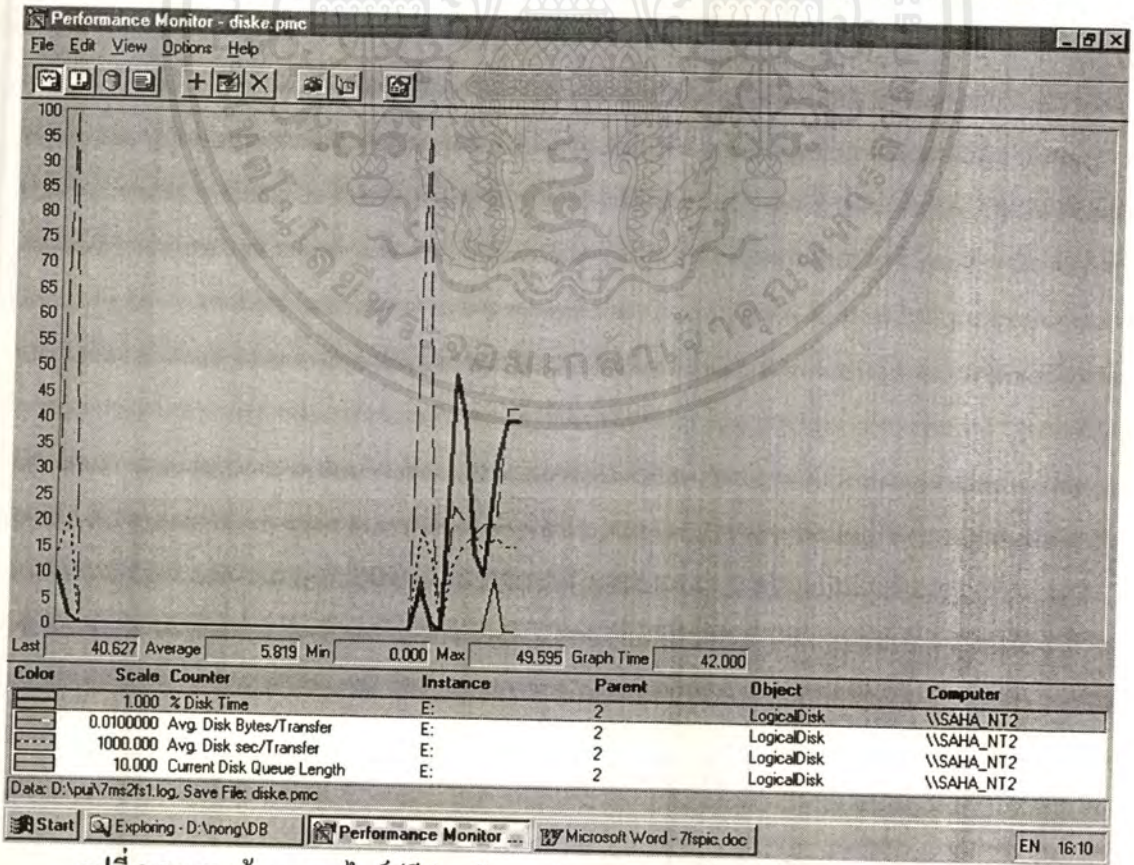


รูปที่ 8-3 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

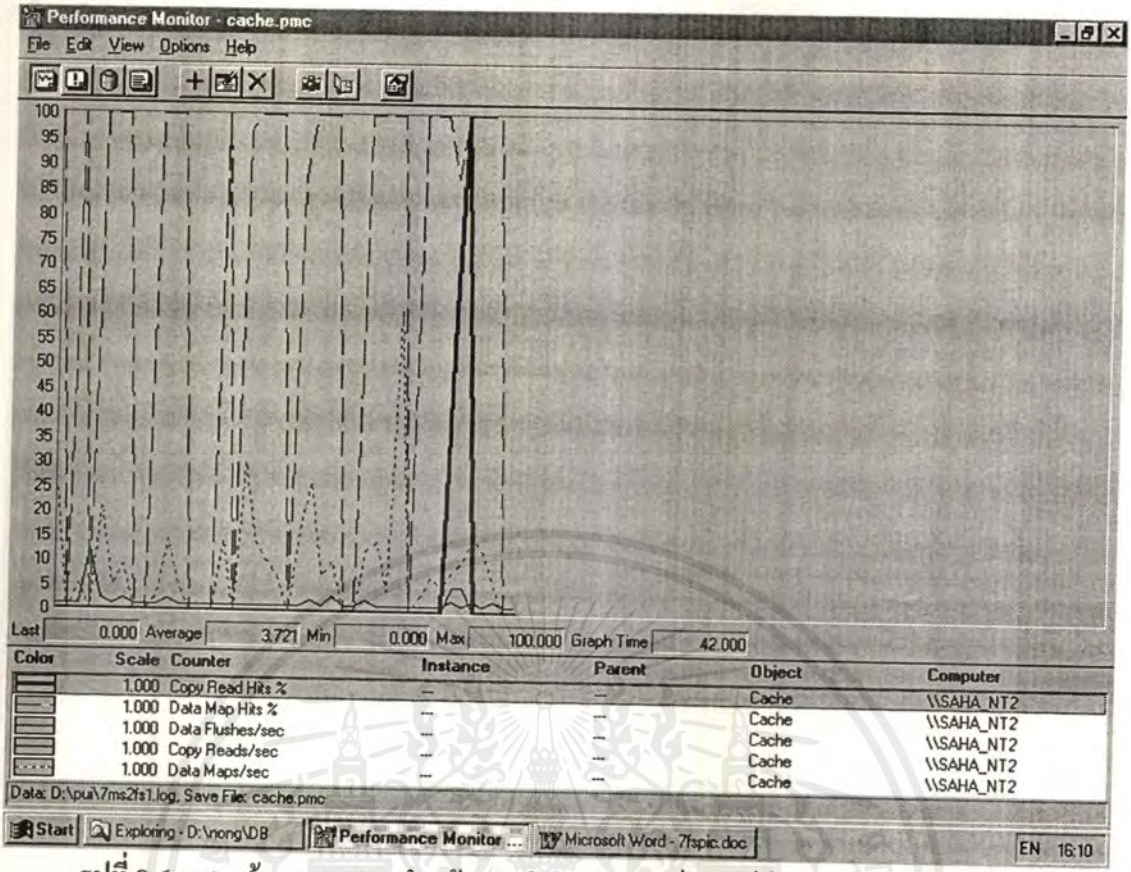


รูปที่ 8-4 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน



รูปที่ 8-5 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-6 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. %Processor Time มีค่าเฉลี่ย 7.977 และ ค่ามากที่สุด 33.997 ซึ่งถือว่ามีค่าค่อนข้างต่ำ
2. Processor Queue Length จากกราฟมีคิวเกิดขึ้นตลอดช่วงของการล็อค ส่วนใหญ่จะมี Queue Length เป็น 1, มีค่ามากที่สุด เป็น 2 และค่าเฉลี่ยเป็น 0.512 จึงไม่น่าจะมีปัญหาโพรเซสเซอร์เกิดขึ้น เมื่อดูจากค่าทั้งสองข้างต้น โพรเซสเซอร์ไม่น่าจะเป็นคอขวด

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยที่ 15.95 แสดงว่าปริมาณเพจที่อ่านเข้ามาและเขียนออกไปยังดิสก์จริง ๆ นั้น มากพอสมควร เพราะค่านี้ควรจะมีค่าน้อยกว่า 5 pages/sec
2. Available Bytes มีค่าเฉลี่ยประมาณ 1MB. หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ available bytes มีค่าประมาณเท่ากับเทรสโฮล ที่กำหนดคือ 1MB เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ WindowsNT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอพพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวลไว้ซึ่งอาจจะทำให้ประสิทธิภาพของหน่วยความจำต่ำลง
3. Commit Limit มีค่าประมาณ 57 MB ซึ่ง Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิทได้ โดยไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์ ซึ่งค่าCommit Limitนี้มีค่าเท่ากับขนาดของpaging file.sys รวมกับ ขนาด RAM ที่สามารถเปลี่ยนไปยังดิสก์ได้ (หรือจำนวนของแรมที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่าเฉลี่ยประมาณ 44 MB.ซึ่งถือว่าไม่มีปัญหาเพราะมีค่าไม่เกินขนาดของ Commit Limit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Page Faults/sec มีค่า 24.916 เมื่อเทียบกับค่า Transition Faults/sec ซึ่งเท่ากับ 2.544 จะเห็นว่าทั้ง 2 ค่าไม่ใกล้เคียงกัน จึงไม่ใช่เป็นคอขวดที่หน่วยความจำ
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB ในที่นี้หน่วยความจำฟิสิคอลเท่ากับ 64 MB. เพราะฉะนั้นค่า Pool Nonpaged Bytes ควรมีค่า 60 MB. ขึ้นไป
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของไดร์ฟต่าง ๆ ได้ว่าค่า Page Reads/sec มีค่าเท่ากับ Disk Reads/sec ของ ไดร์ฟซี
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้ไม่ใกล้เคียงกันเลย
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}}$ * 4096 bytes/page = 5762 bytes/read เทียบกับค่า

   Avg. Disk Bytes/Read ได้ว่ามีค่าใกล้เคียงกับ ไดร์ฟซี
10. $\frac{\text{Page Outputs/sec}}{\text{Page Writes/sec}}$ * 4096 bytes/page = 10464 bytes/write เทียบกับค่า

    Avg. Disk Bytes/Write ซึ่งมีค่าไม่ใกล้เคียงกับ ไดร์ฟใดเลย

    สรุปว่าเกิดคอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่ดิสก์

1. %Disk Time

- ไดร์ฟซี: มีค่าเฉลี่ย 13.297 ซึ่งถือว่าต่ำมาก แสดงว่ามีการเข้าถึงดิสก์ไดร์ฟซีน้อยมาก
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการเข้าถึงไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าเฉลี่ย 5.819 ซึ่งถือว่าต่ำมากเช่นกัน เพราะฉะนั้นการใช้ไดร์ฟอีจึงน้อยมาก และเมื่อดูจากกราฟมีการเข้าถึงไดร์ฟอีในช่วงเกือบท้าย ๆ ของการล็อคเพียงช่วงเดียวเท่านั้น

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีค่าประมาณ 21K แสดงว่าการโอนย้ายข้อมูลมีประสิทธิภาพดี เพราะค่านี้ควรจะมีค่ามากกว่า 20K
- ไดร์ฟดี: มีค่าประมาณ 0K แสดงว่าไม่มีการโอนย้ายข้อมูลที่ ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าประมาณ 1K ประสิทธิภาพการโอนย้ายข้อมูลน้อยมาก และเมื่อดูจากกราฟจะเห็นว่ามีการ โอนย้ายข้อมูลในช่วงเกือบท้าย ๆ ของการล็อคเพียงช่วงเดียวเช่นกัน

3. Avg. Disk Sec/transfer

- ไดร์ฟซี: มีค่า 0.009 ซึ่งค่อนข้างต่ำแสดงว่าดิสก์ใช้เวลาในการโอนย้ายข้อมูลน้อยเพราะถ้าค่านี้มีค่ามากกว่า 0.3 sec. แสดงว่าตัวควบคุมดิสก์พยายามที่จะเข้าถึงดิสก์ซ้ำ ๆ เนื่องจากการอ่านหรือการเขียนเกิดผิดพลาด
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการโอนย้ายข้อมูลที่ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่า 0.004 ซึ่งค่าต่ำเช่นกัน

4. Disk Queue Length

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

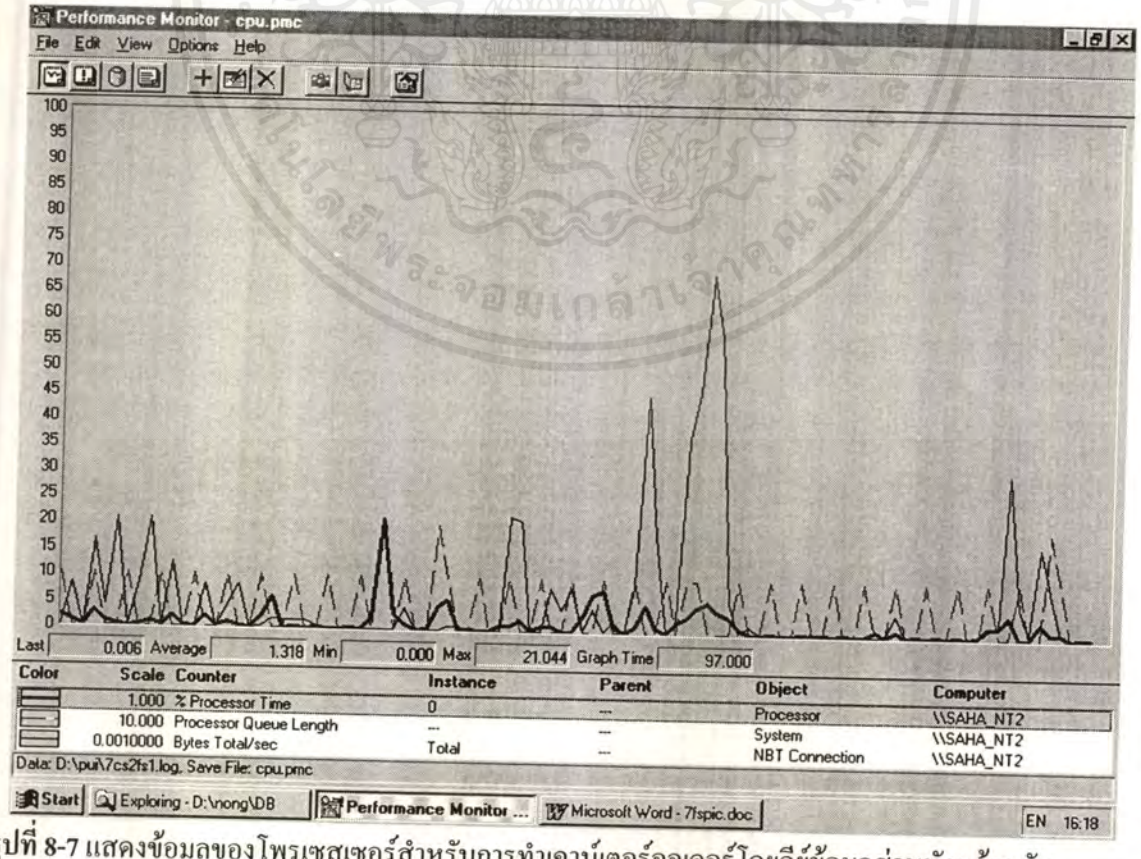
- ไดร์ฟซี: จากกราฟจะเห็นว่ามีคิวเกิดขึ้นเพียงช่วงท้ายของการล็อคเท่านั้น และเกิดค่า Queue Length มากที่สุดเป็น 3 แต่มีค่าเฉลี่ยเพียง 0.14 จึงไม่น่าจะเป็นปัญหาอะไร เพราะถ้าค่า Disk Queue Length มากกว่า 2 จึงจะถือว่าเกิดคอขวดที่ดิสก์ขึ้น
- ไดร์ฟดี: เป็นศูนย์ จึงไม่น่าจะมีปัญหาอะไร
- ไดร์ฟอี: จากกราฟมีคิวเกิดขึ้นเพียงช่วงท้ายของการล็อคเท่านั้น และมีค่ามากที่สุด เป็น 1 และค่าเฉลี่ยเพียง 0.047 จึงไม่น่าจะมีปัญหาเกิดขึ้นที่ไดร์ฟอี

สรุปในการทำแมสออเดอร์ โดยคีย์ข้อมูลส่วนหัวพร้อมกันนี้ไม่เกิดคอขวดที่ดิสก์ขึ้น การโอนย้ายข้อมูลของไดร์ฟซี มีประสิทธิภาพดี แต่ของไดร์ฟอื่น ๆ นั้นยังไม่มีประสิทธิภาพเท่าที่ควร และเป็นที่น่าสังเกตว่าไดร์ฟซี มีอัตราการโอนย้ายข้อมูลที่สูง แต่ใช้เวลาการโอนย้ายข้อมูลน้อยมาก เมื่อเทียบกับไดรว์อี ซึ่งอัตราการ โอนย้ายข้อมูลต่ำกว่าไดร์ฟซีมากแต่ใช้เวลามากกว่าน้อยกว่าไดร์ฟซี เพียงเล็กน้อยเท่านั้น

การตรวจจับจุดคอขวดที่แคช

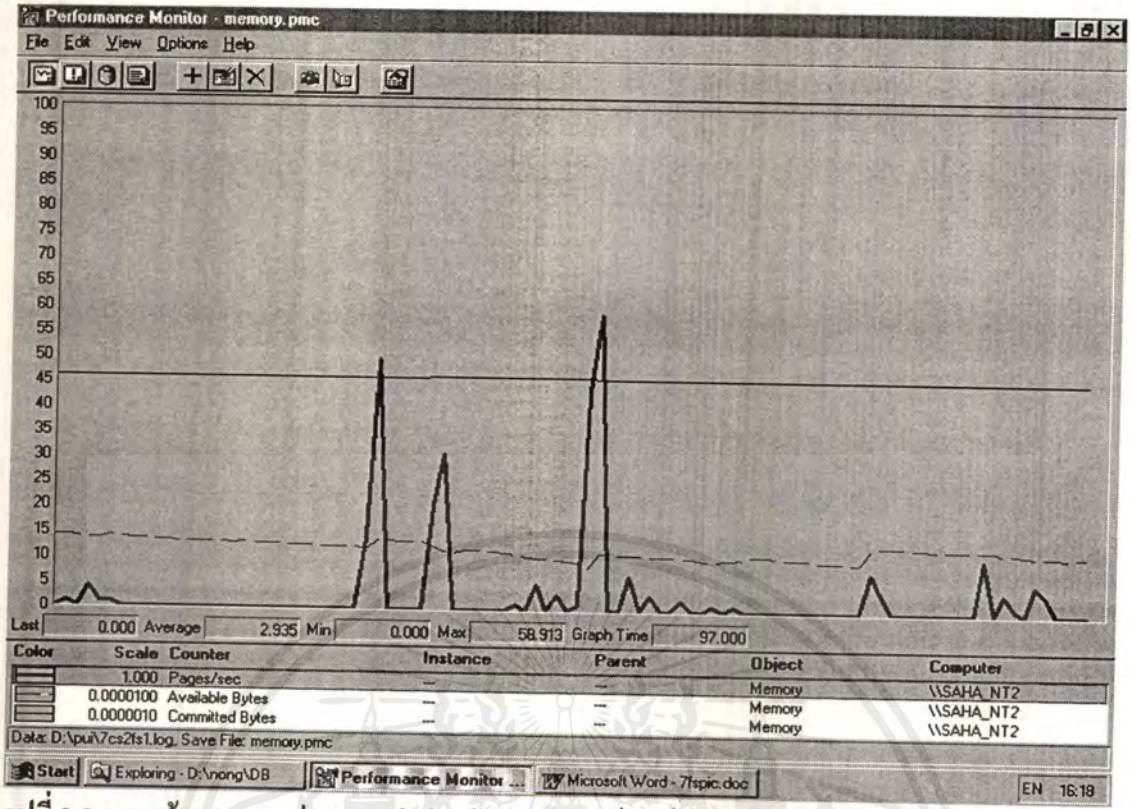
1. Copy Read Hits% อัตราการฮิตจะสูงขึ้นจนถึง100%ช่วงท้ายของการล็อคเท่านั้น แต่ส่วนใหญ่แล้วจะเป็น 0
2. Data Flushes/sec มีค่า 1.486 ซึ่งหมายถึงแคชมีการนำข้อมูลออก โดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วง เพียง 1.486
3. Data Map Hits% จากกราฟจะเห็นว่า กิจกรรมเกี่ยวกับไดเรกทอรี่หรือกิจกรรมที่เกี่ยวกับไฟล์จำนวนมากนั้นมีค่าสูงเป็นระยะ ๆ ตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดการล็อค

2. เคาน์เตอร์ออเดอร์ คีย์ข้อมูลส่วนหัว พร้อมกัน ได้ผลดังนี้

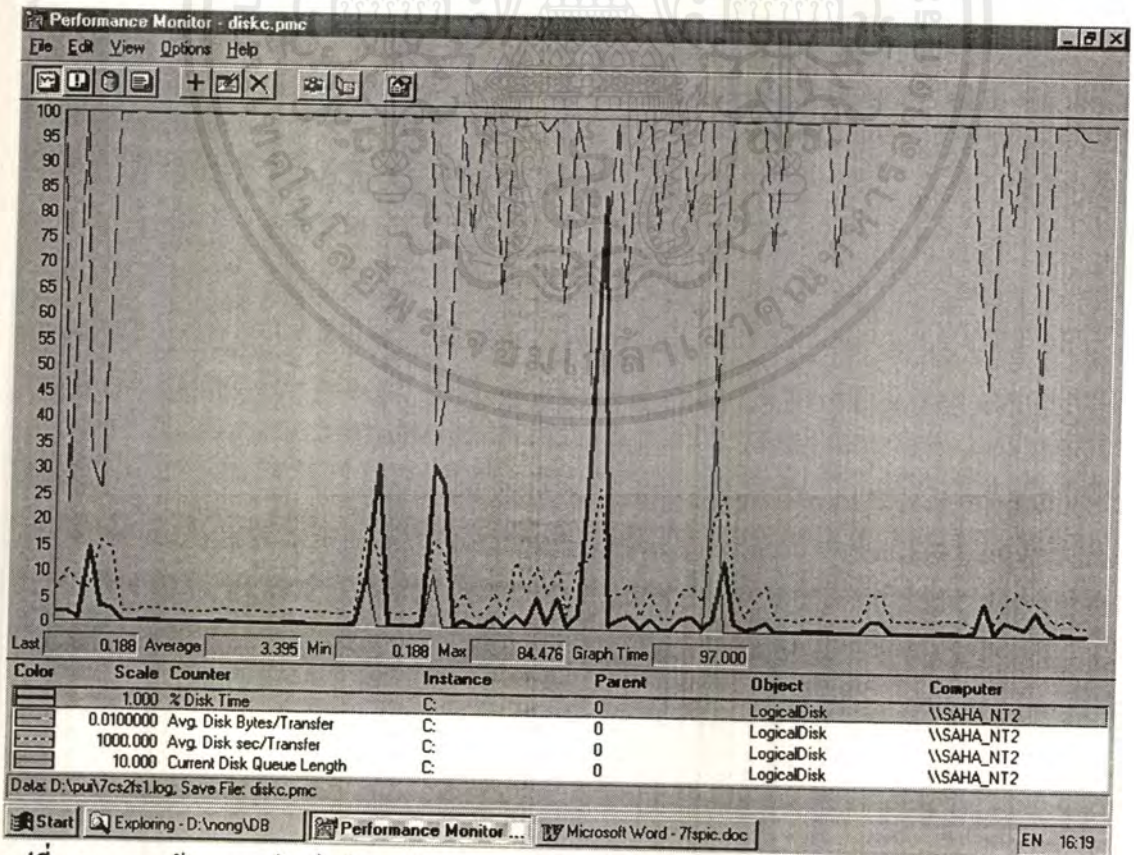


รูปที่ 8-7 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

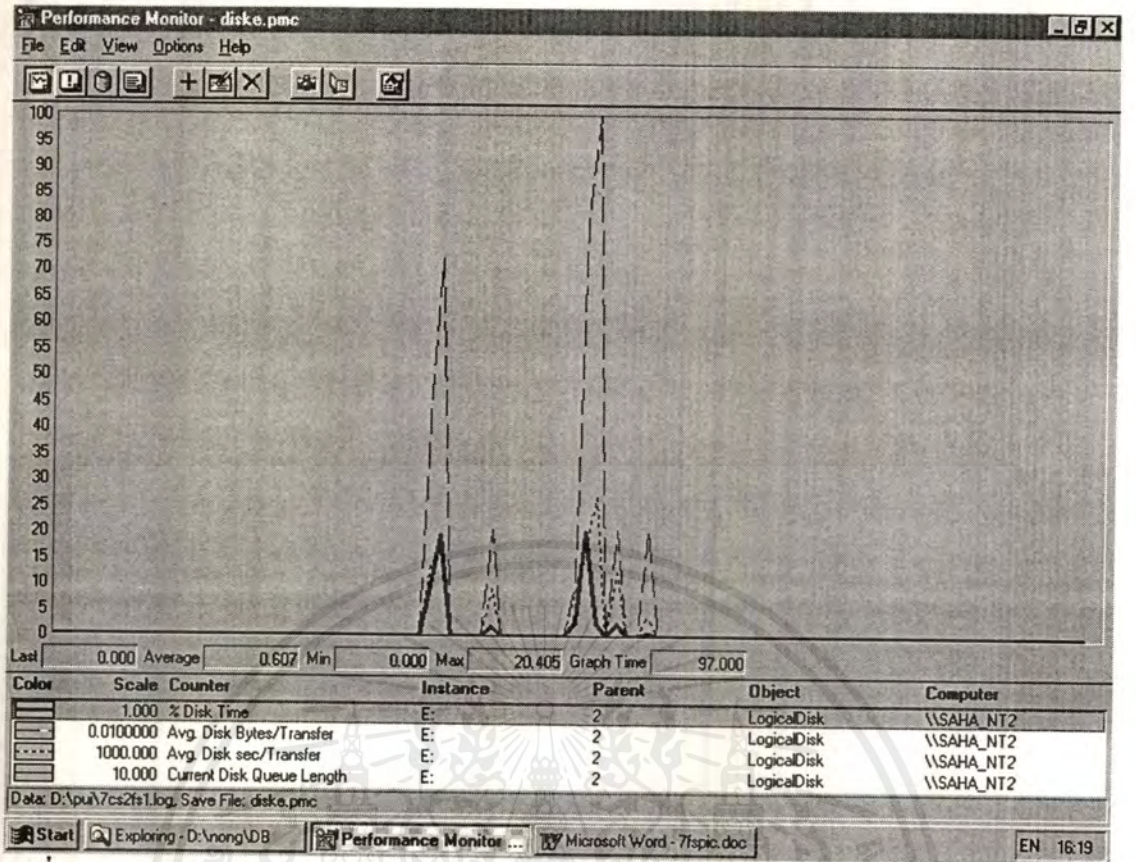


รูปที่ 8-8 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

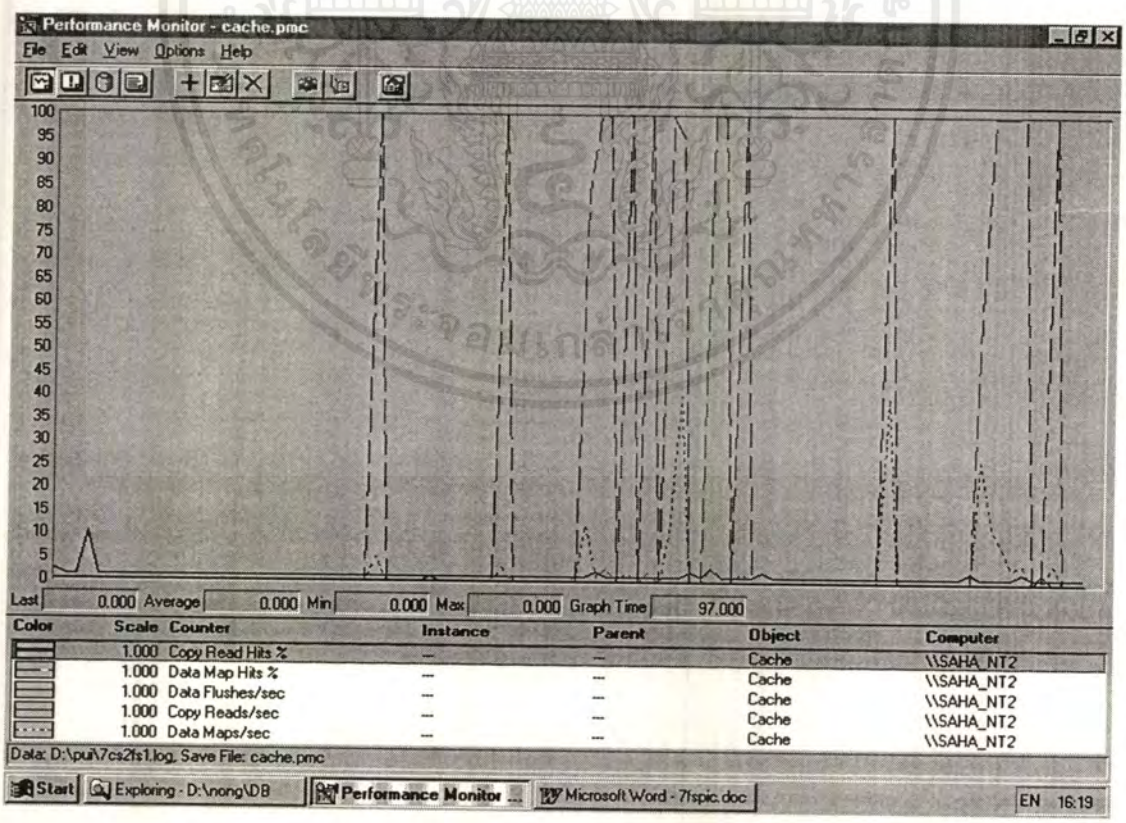


รูปที่ 8-9 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-10 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน



รูปที่ 8-11 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. %Processor Time มีค่าเฉลี่ย 1.318 และ ค่ามากที่สุด ที่ 21.044 ซึ่งถือว่ามีค่าค่อนข้างต่ำ
2. Processor Queue Length จากกราฟมีคิวเกิดขึ้นตลอดการล็อค ส่วนใหญ่จะมี Queue Length เป็น 1 , มีค่า ค่ามากที่สุด เป็น 2 และค่า เฉลี่ย เป็น 1.318 จึงไม่น่าจะมีปัญหาโพรเซสเซอร์เกิดขึ้น เมื่อดูจากค่าทั้งสองข้างต้นโพรเซสเซอร์ไม่น่าจะเป็น คอขวด

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยที่ 2.935 แสดงว่าปริมาณเพจที่อ่านเข้ามาและเขียนออกไปยังดิสก์ จริง ๆ นั้น ไม่มากนัก เพราะค่านี้ควรจะมีค่าน้อยกว่า 5 pages/sec

2. Available Bytes มีค่าเฉลี่ยประมาณ 1MB. หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าประมาณเท่ากับเทรสโฮลที่กำหนดคือ 1MB เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ WindowsNT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอพพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวลไว้ ซึ่งอาจจะทำให้ประสิทธิภาพของหน่วยความจำต่ำลง

1. Commit Limit มีค่าประมาณ 57 MB ซึ่ง Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิทได้ โดยไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ paging file.sys รวมกับขนาด RAM ที่สามารถเปลี่ยนไปยังดิสก์ได้ (หรือจำนวนของแรมที่เหลือจากการใช้เป็น ระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่าเฉลี่ยประมาณ 46 MB. ซึ่งถือว่าไม่มีปัญหาเพราะมีค่าไม่เกินขนาดของ Commit Limit
5. Page Faults/sec มีค่า 10.356 เมื่อเทียบกับค่า Transition Faults/sec ซึ่งเท่ากับ 4.087 จะเห็นว่าทั้ง 2 ค่าไม่ใกล้เคียงกัน จึงไม่ใช่เป็น หน่วยความจำ คอขวด
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB ในที่นี้หน่วยความจำฟิสิคอล เท่ากับ 64 MB. เพราะฉะนั้นค่า Pool Nonpaged Bytes ควรมีค่า 60 MB. ขึ้นไป
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของ ไดร์ฟต่าง ๆ ได้ว่าค่า Page Reads/sec มีค่าเท่ากับ Disk Reads/sec ของ ไดร์ฟซี
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดรว์ต่าง ๆ ซึ่งค่าที่อ่านได้ไม่ใกล้เคียงกันเลย
9. $\frac{\text{Page Inputs/sec} * 4096 \text{ bytes/page}}{\text{Page Reads/sec}} = 6445 \text{ bytes/read}$ เทียบกับค่า

   Avg. Disk Bytes/Read ได้ว่ามีค่าไม่ใกล้เคียงกับ ไดร์ฟใดเลย
10. $\frac{\text{Page Outputs/sec} * 4096 \text{ bytes/page}}{\text{Page Writes/sec}} = 9784 \text{ bytes/write}$ เทียบกับค่า

    Avg. Disk Bytes/Write ซึ่งมีค่าไม่ใกล้เคียงกับ ไดร์ฟใดเลย

    สรุปว่าไม่เกิดคอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่ดิสก์

1. %Disk Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไดร์ฟซี: มีค่าเฉลี่ย 3.395 ซึ่งถือว่าต่ำมาก แสดงว่ามีการเข้าถึงดิสก์ไดร์ฟซีน้อยมาก
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการเข้าถึงไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าเฉลี่ย 0.607 ซึ่งถือว่าต่ำมากเช่นกัน เพราะฉะนั้นการใช้ไดร์ฟอีจึงน้อยมาก และเมื่อดูจากกราฟมีการเข้าถึงไดร์ฟอีเพียงช่วงกลางของการล็อคเท่านั้น

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีค่าประมาณ 13K แสดงว่าการโอนย้ายข้อมูลยังไม่มีประสิทธิภาพเท่าที่ควร เพราะค่านี้ควรจะมีค่ามากกว่า 20K จึงจะถือว่าการโอนย้ายข้อมูลมีประสิทธิภาพดี
- ไดร์ฟดี: มีค่าประมาณ 0K แสดงว่าไม่มีการโอนย้ายข้อมูลที่ ไดร์ฟดี เลย
- ไดร์ฟอี: มีค่าประมาณ 444 bytes แสดงว่าประสิทธิภาพการโอนย้ายข้อมูลต่ำมาก

3. Avg. Disk Sec/transfer

- ไดร์ฟซี: มีค่า 0.006 ซึ่งค่อนข้างต่ำ แสดงว่าดิสก์ใช้เวลาในการโอนย้ายข้อมูลน้อย เพราะถ้าค่านี้มีค่ามากกว่า0.3 sec. แสดงว่าตัวควบคุมดิสก์พยายามที่จะเข้าถึงดิสก์ซ้ำๆเนื่องจากการอ่านหรือการเขียนเกิดผิดพลาด
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการโอนย้ายข้อมูลที่ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่า 0.001 ซึ่งค่าต่ำเช่นกัน

4. Disk Queue Length

- ไดร์ฟซี: จากกราฟจะเห็นว่ามีคิวเกิดขึ้นประมาณช่วงกลางของการล็อค และเกิด ค่า Queue Length มากที่สุด เท่ากับ 4 แต่มีค่าเฉลี่ยเพียง 0.061 จึงไม่น่าจะเป็นปัญหาอะไร เพราะถ้าค่า Disk Queue Length มากกว่า 2 จึงจะถือว่าเกิดคอขวดที่ดิสก์ขึ้น
- ไดร์ฟดี: เป็นศูนย์ จึงไม่น่าจะมีปัญหาอะไร
- ไดร์ฟอี: เป็นศูนย์เช่นกัน

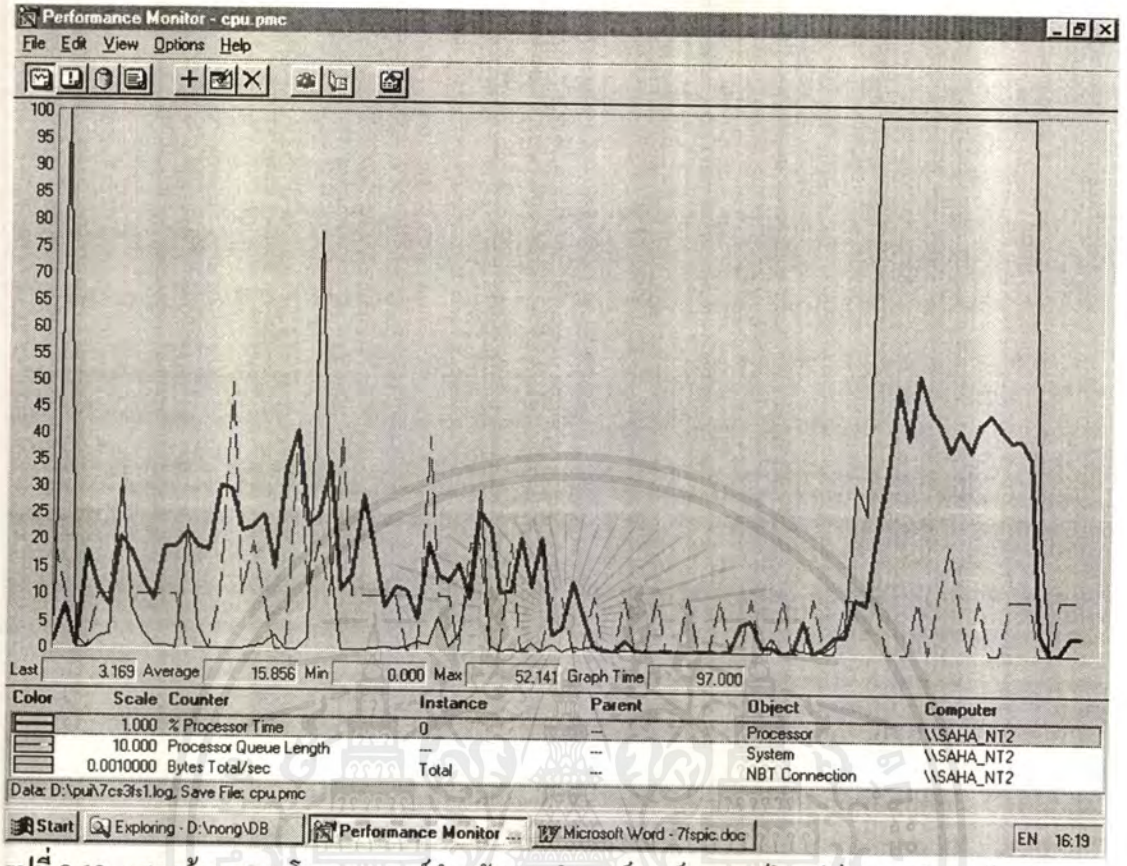
สรุปในการทำเคาน์เตอร์ออเดอร์ โดยดีย์ข้อมูลส่วนหัวพร้อมกัน. นี้ไม่เกิดคอขวดที่ดิสก์ขึ้น แต่การโอนย้ายข้อมูลของดิสก์นั้นยังไม่มีประสิทธิภาพเท่าที่ควร

การตรวจจับจุดคอขวดที่แคช

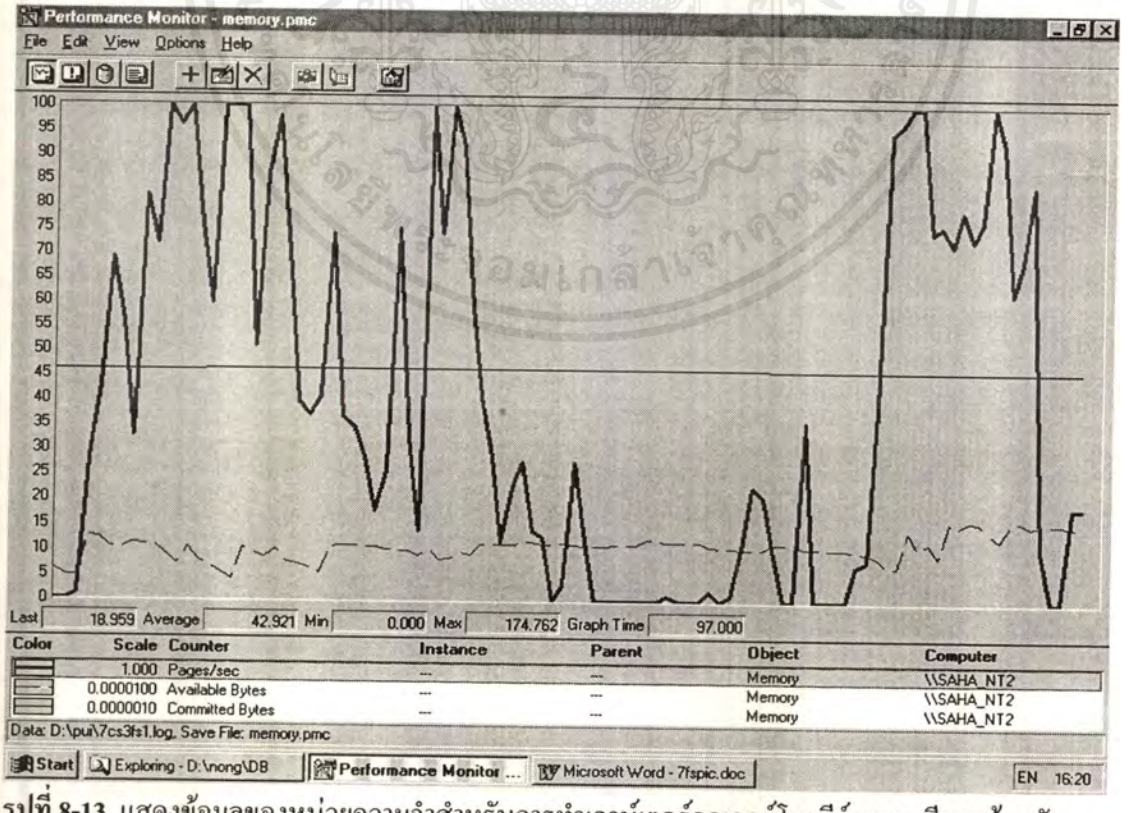
1. Copy Read Hits% เป็นศูนย์ตลอดช่วงของการล็อค
2. Data Flushes/sec มีค่า 1.172 ซึ่งหมายถึงแคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วง เพียง 1.172
3. Data Map Hits% จากกราฟจะเห็นว่า กิจกรรมเกี่ยวกับไดเรกทอรี่หรือกิจกรรมที่เกี่ยวกับไฟล์จำนวนมากนั้นมีค่าสูงเป็นระยะ ๆ ตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดการล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เคาน์เตอร์ออเดอร์ คีย์รายละเอียดพร้อมกัน ได้ผลดังนี้

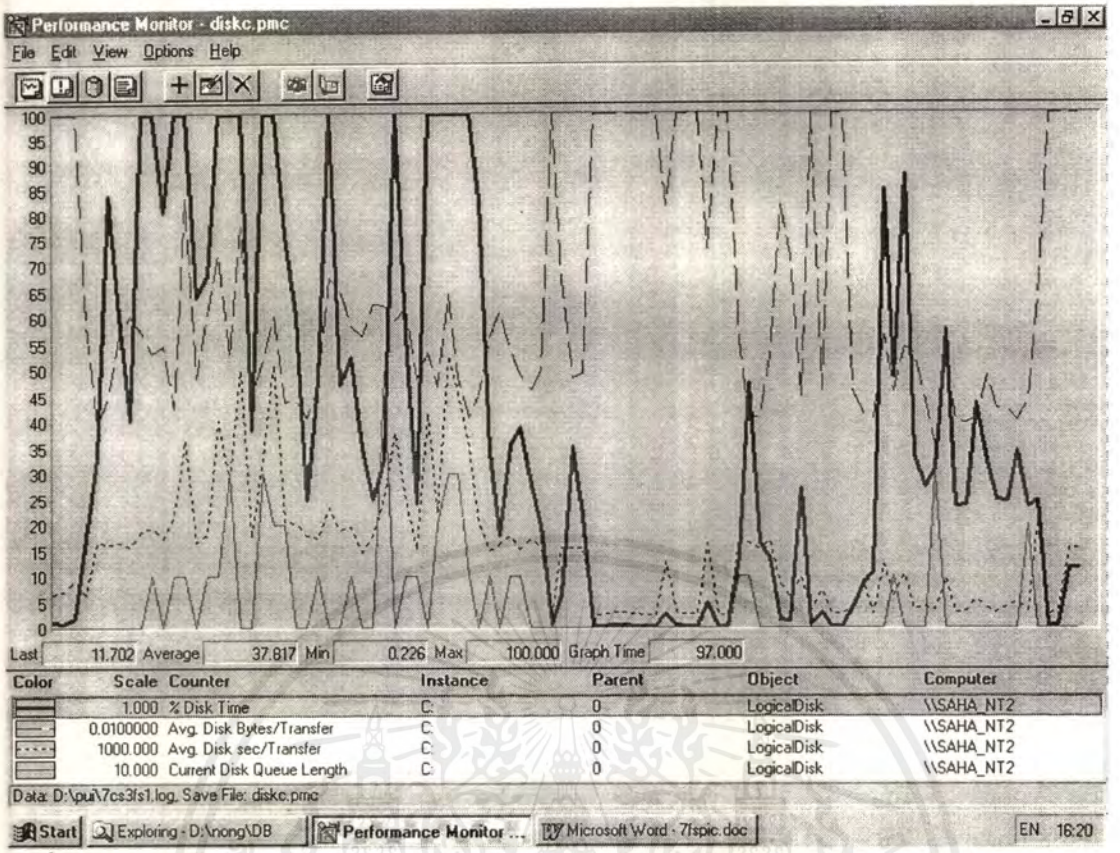


รูปที่ 8-12 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

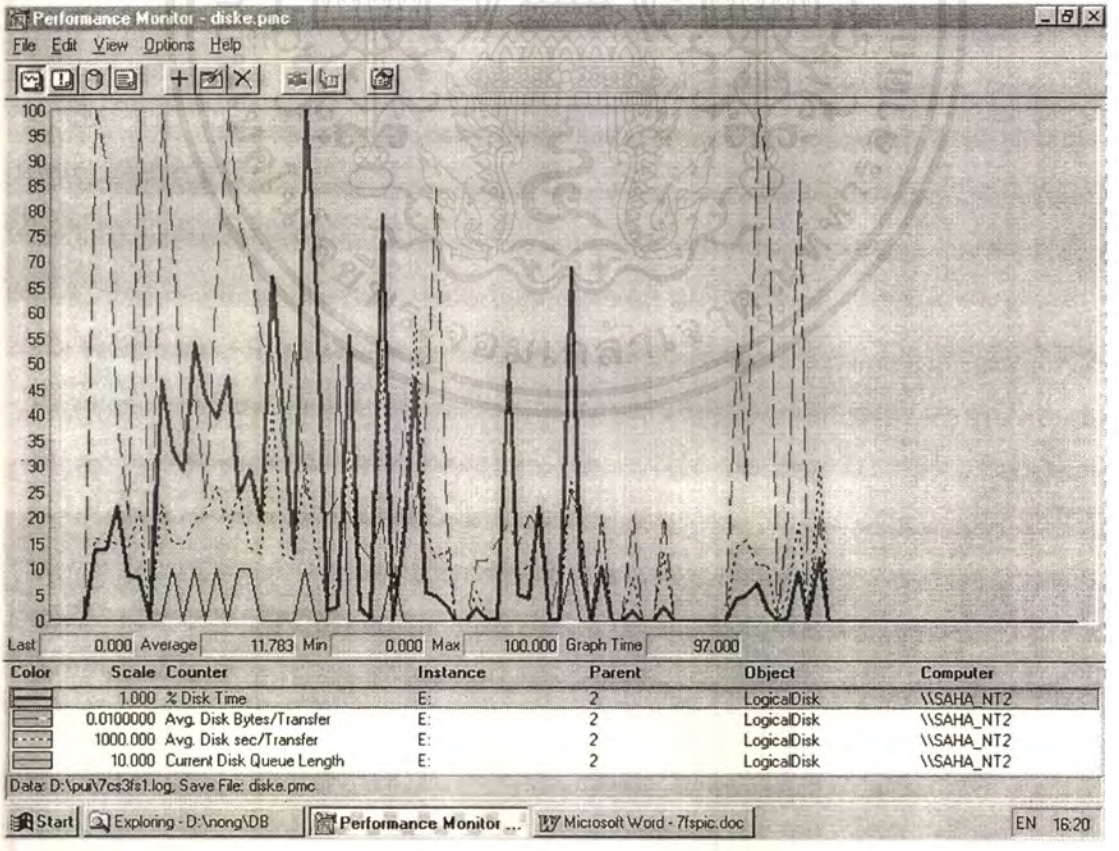


รูปที่ 8-13 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

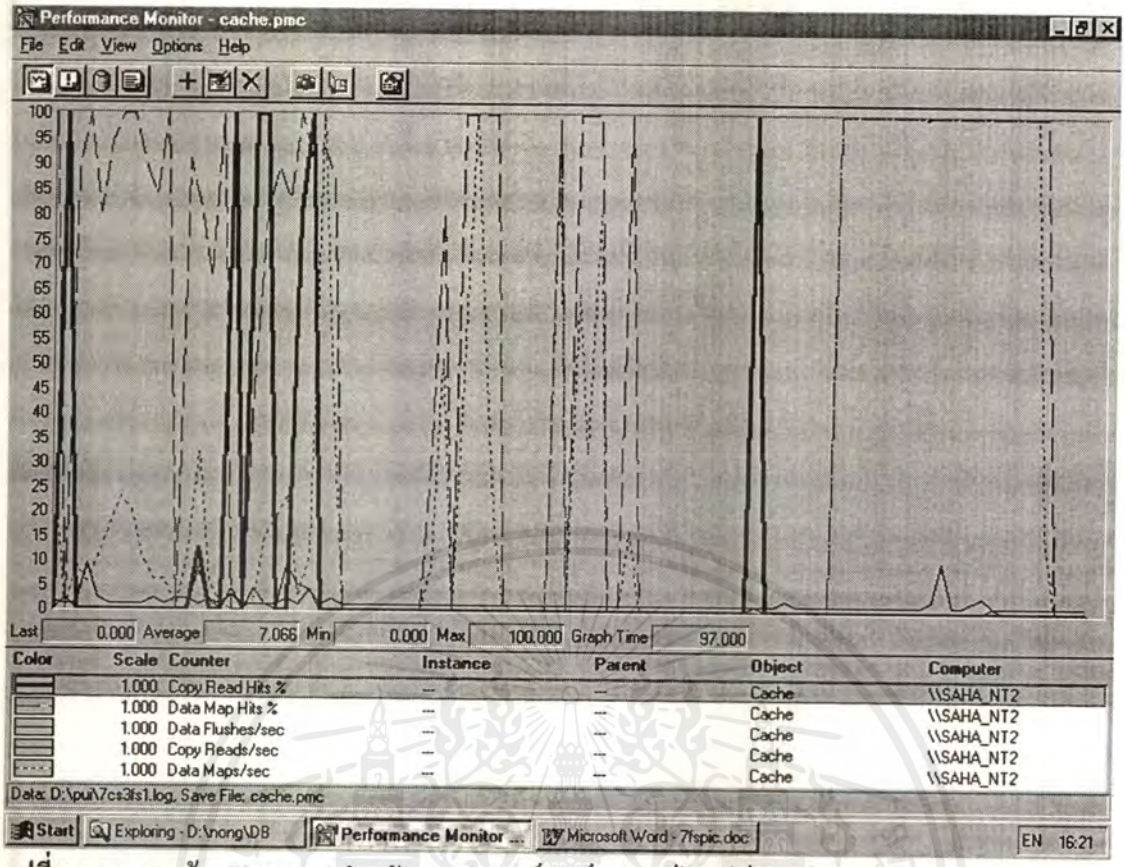


รูปที่ 8-14 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน



รูปที่ 8-15 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-16 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. %Processor Time มีค่าเฉลี่ย 15.856 และ ค่ามากที่สุด ที่ 52.141 ซึ่งถือว่ามีค่าปานกลาง
2. Processor Queue Length มีคิวเกิดขึ้นตลอดการล็อค มีค่ามากที่สุดเป็น5,ค่าเฉลี่ยเป็น 0.796จึงไม่น่ามีปัญหาโพรเซสเซอร์เกิดขึ้น แต่เมื่อเทียบกับการทดลองที่ผ่านมาจะพบว่าโพรเซสเซอร์มีคิวมากขึ้น เมื่อดูจากค่าทั้งสองข้างต้นโพรเซสเซอร์ ไม่น่าจะเป็น คอขวด

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยที่ 42.921 แสดงว่าปริมาณเพจที่อ่านเข้ามาและเขียนออกไปยัง ดิสก์ จริง ๆ นั้นมากพอสมควร เพราะค่านี้ควรจะมีค่าน้อยกว่า 5 pages/sec
2. Available Bytes มีค่าเฉลี่ยประมาณ 1MB. หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าประมาณเท่ากับ เทรสโฮล ที่กำหนดคือ 1MB เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ WindowsNT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอพพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวลไว้ ซึ่งอาจจะทำให้ ประสิทธิภาพของหน่วยความจำต่ำลง
3. Commit Limit มีค่าประมาณ 57 MB ซึ่ง Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิทได้ โดยไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ paging file.sys รวมกับขนาด RAM ที่สามารถเปลี่ยนไปยังดิสก์ได้ (หรือจำนวนของแรมที่เหลือจากการใช้เป็น ระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่าเฉลี่ยประมาณ 46 MB. ซึ่งถือว่าไม่มีปัญหาเพราะมีค่าไม่เกินขนาดของ Commit Limit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Page Faults/sec มีค่า 55.670 เมื่อเทียบกับค่า Transition Faults/sec ซึ่งเท่ากับ 22.870 จะเห็นว่าทั้ง 2 ค่าไม่ใกล้เคียงกัน จึงไม่ใช่เป็นคอขวดที่หน่วยความจำ แต่เป็นที่น่าสังเกตว่า เมื่อเทียบกับรูปแบบที่ผ่าน ๆ มาแล้ว ค่า Page Faults/sec และค่า Transition Faults/sec มีค่าเพิ่มขึ้นมาก
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB ในที่นี้หน่วยความจำฟิสิคอล เท่ากับ 64 MB. เพราะฉะนั้นค่า Pool Nonpaged Bytes ควรมีค่า 60 MB. ขึ้นไป
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของ ไดร์ฟต่าง ๆ ได้ว่าค่า Page Reads/sec มีค่าเท่ากับ Disk Reads/sec ของไดร์ฟซี
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดรว์ต่างๆ ซึ่งค่าที่อ่านได้ใกล้เคียงกับไดรว์ซี
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}}$ * 4096 bytes/page = 5131 bytes/read เทียบกับค่า

   Avg. Disk Bytes/Read ได้ว่ามีค่าไม่ใกล้เคียงกับ ไดร์ฟใดเลย
10. $\frac{\text{Page Outputs/sec}}{\text{Page Writes/sec}}$ * 4096 bytes/page = 197315 bytes/write เทียบกับค่า

    Avg. Disk Bytes/Write ซึ่งมีค่าไม่ใกล้เคียงกับ ไดร์ฟใดเลย

    สรุปว่าเกิดคอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่ดิสก์

1. %Disk Time

- ไดร์ฟซี: มีค่าเฉลี่ย 37.817 ซึ่งถือว่าค่อนข้างต่ำ แต่มีค่ามากที่สุด เป็น 100% และจากกราฟ %Disk Time จะมีค่าถึง 100% ในช่วงแรก ๆ ของการล็อค
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการเข้าถึงไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าเฉลี่ย 11.783 แต่มีค่ามากที่สุด เป็น 100% และจากกราฟ จะมี %Disk Time ขึ้นสูงในช่วงแรก ๆ ของการล็อคเช่นเดียวกับไดร์ฟซี

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีค่าประมาณ 7K แสดงว่าการโอนย้ายข้อมูลยังไม่มีประสิทธิภาพเท่าที่ควร เพราะค่านี้ควรจะมีค่ามากกว่า 20K จึงจะถือว่าการโอนย้ายข้อมูลมีประสิทธิภาพดี
- ไดร์ฟดี: มีค่าประมาณ 0K แสดงว่าไม่มีการโอนย้ายข้อมูลที่ ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าประมาณ 2K แสดงว่าประสิทธิภาพการโอนย้ายข้อมูลต่ำมาก

3. Avg. Disk Sec/transfer

- ไดร์ฟซี: มีค่า 0.014 ซึ่งค่อนข้างต่ำแสดงว่าดิสก์ใช้เวลาในการโอนย้ายข้อมูลน้อย เพราะถ้าค่านี้มีค่ามากกว่า 0.3 sec. แสดงว่า ตัวควบคุมดิสก์พยายามที่จะเข้าถึงดิสก์ซ้ำ ๆ เนื่องจากการอ่านหรือการเขียนเกิดผิดพลาด
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการโอนย้ายข้อมูลที่ไดร์ฟดีเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไดร์ฟอี: มีค่า 0.009 ซึ่งค่าต่ำเช่นกัน

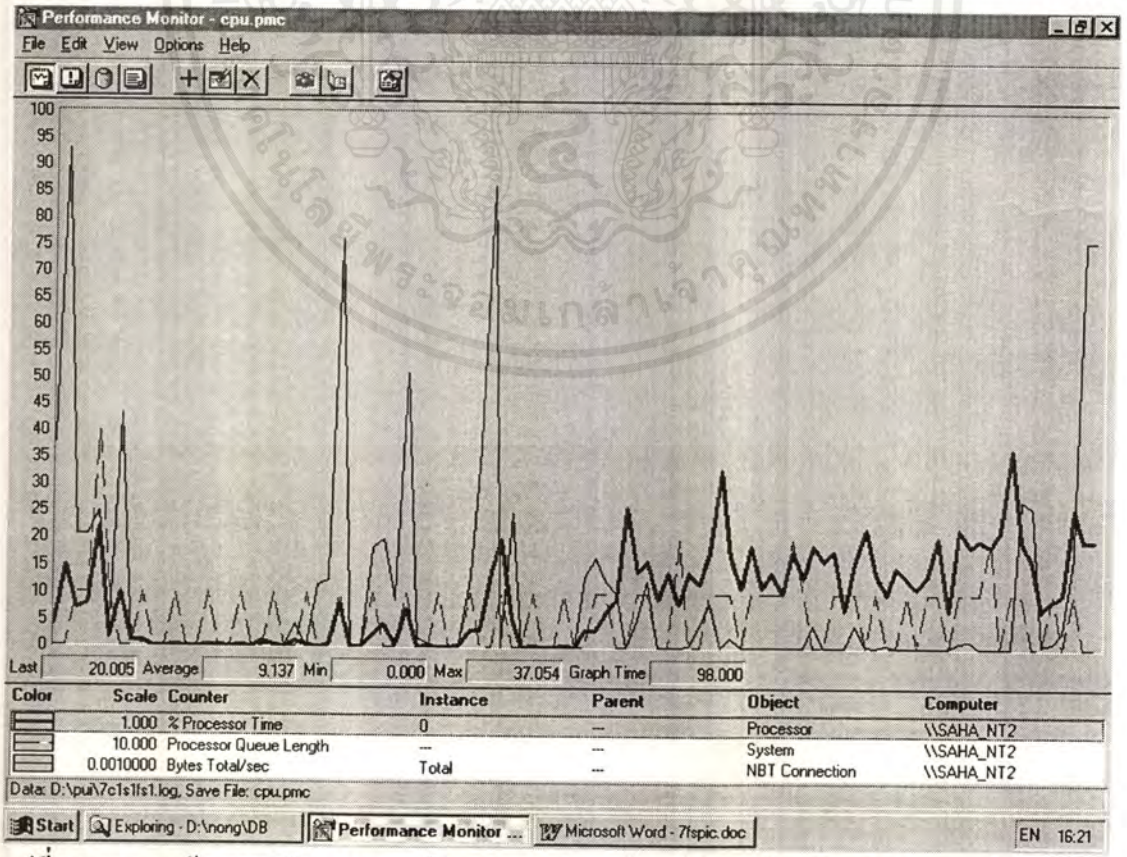
4. Disk Queue Length

- ไดร์ฟซี: จากกราฟจะเห็นว่ามีคิวเกิดขึ้นตลอดช่วงของการล็อค และเกิดค่า Queue Length มากที่สุด เป็น 5 และมีค่าเฉลี่ย 0.449 จึงไม่น่าจะเป็นปัญหาอะไร เพราะถ้าค่า Disk Queue Length มากกว่า 2 จึงจะถือว่าเกิดคอขวดที่ดิสก์ขึ้น
- ไดร์ฟดี: เป็นศูนย์ จึงไม่น่าจะมีปัญหาอะไร
- ไดร์ฟอี: มี ค่า Queue Length มากที่สุด เป็น 5 และมีค่าเฉลี่ยที่ 0.133 ดังนั้นจึงไม่น่าจะเป็นปัญหา สรุปในการทำเคาน์เตอร์ออเดอร์ โดยคีย์รายละเอียดพร้อมกันนี้ไม่เกิดคอขวดที่ดิสก์ขึ้น แต่การโอนย้ายข้อมูลของดิสก์นั้นยังไม่มีประสิทธิภาพเท่าที่ควร

การตรวจจับจุดคอขวดที่แคช

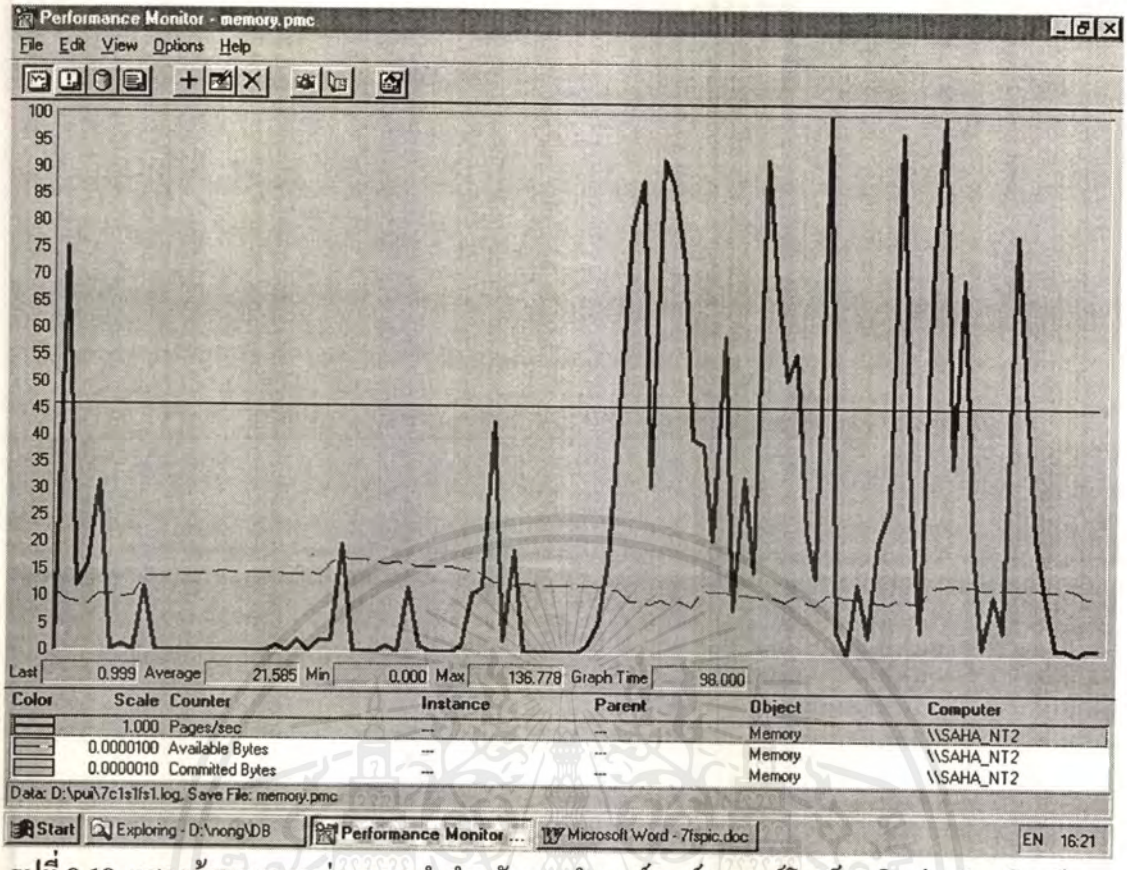
1. Copy Read Hits% เมื่อดูจากกราฟ อัตราการ ฮิต จะสูงขึ้นจนถึง 100% ในช่วงที่มีการใช้ ไดรว์ซี และไดรว์อีมาก แต่ส่วนใหญ่แล้วจะมีค่าเป็นศูนย์
2. Data Flushes/sec มีค่า 1.351 ซึ่งหมายถึง แคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยัง อุปกรณ์ต่อพ่วง เพียง 1.351
3. Data Map Hits% จากกราฟจะเห็นว่า กิจกรรมเกี่ยวกับไดเรกทอรี่หรือกิจกรรมที่เกี่ยวกับไฟล์จำนวนมากนั้นมีค่าสูงเป็นระยะ ๆ ตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดการล็อค

4. เคาน์เตอร์ออเดอร์ ล็อกอินห่างกัน 1 วินาที ได้ผลดังนี้

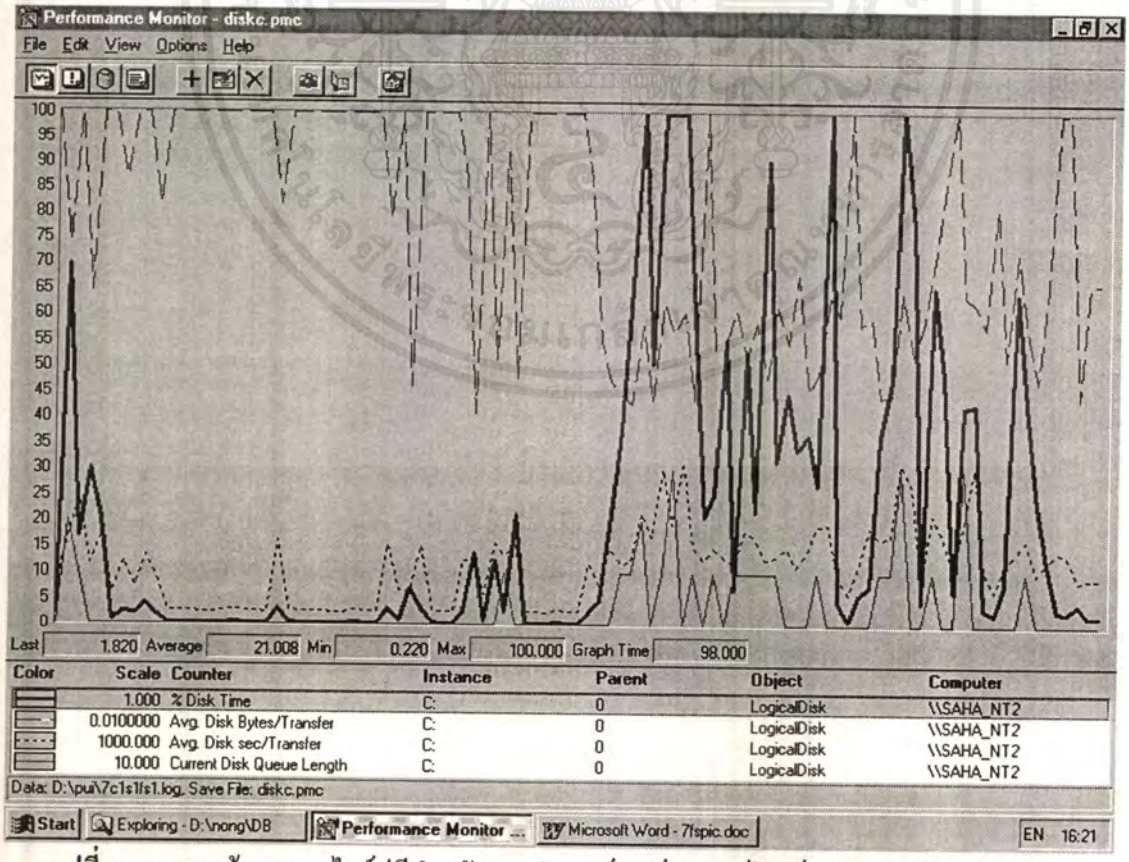


รูปที่ 8-17 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

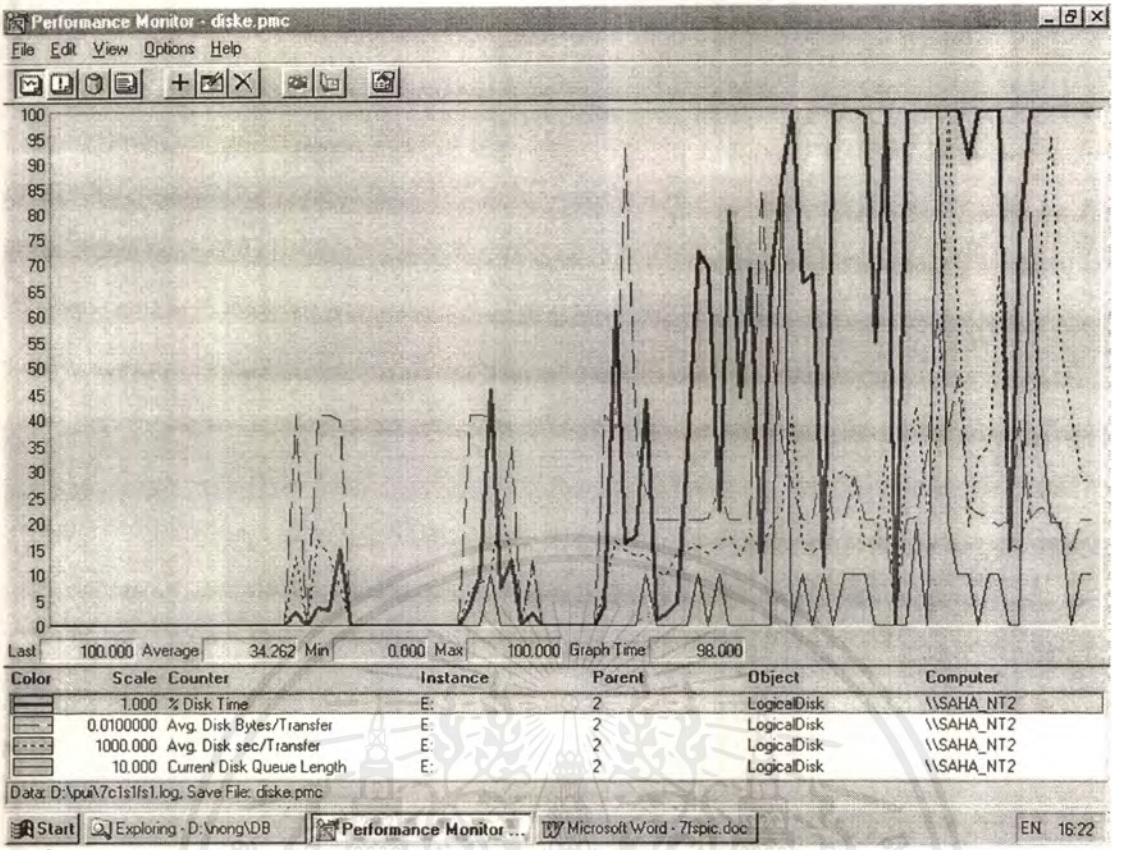


รูปที่ 8-18 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1วินาที

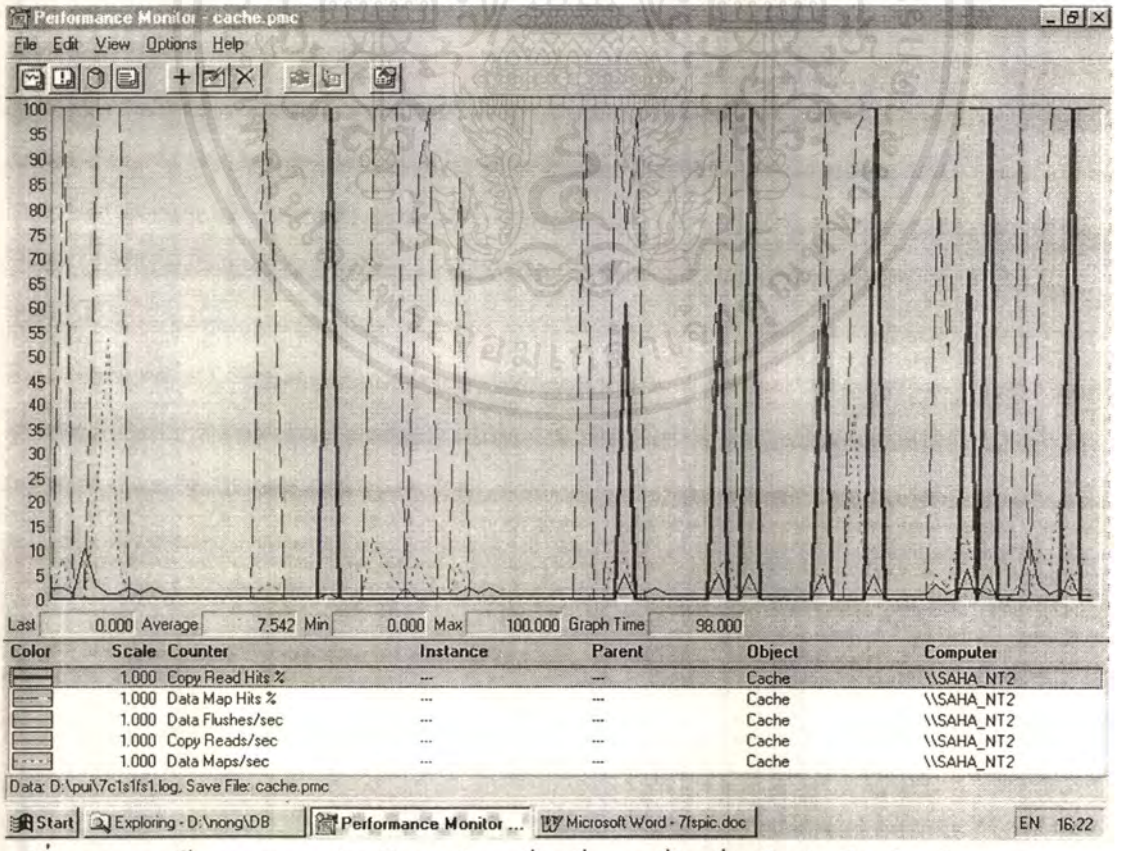


รูปที่ 8-19 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-20 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1วินาที



รูปที่ 8-21 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. %Processor Time มีค่าเฉลี่ย 9.137 และ ค่ามากที่สุด ที่ 37.054 ซึ่งถือว่ามีค่าปานกลาง
2. Processor Queue Length จากกราฟมีคิวเกิดขึ้นตลอดการล็อค ส่วนใหญ่แล้วมี Queue Length เป็น 1 , มีค่า ค่ามากที่สุด เป็น 4 และค่าเฉลี่ย เป็น 0.535 จึงไม่น่าจะมีปัญหาโพรเซสเซอร์เกิดขึ้น เมื่อดูจากค่าทั้งสองข้างต้นโพรเซสเซอร์ ไม่น่าจะเป็นคอขวด

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยที่ 21.585 แสดงว่าปริมาณเพจที่อ่านเข้ามาและเขียนออกไปยังดิสก์ จริง ๆ นั้น มากพอสมควร เพราะค่านี้ควรจะมีค่าน้อยกว่า 5 pages/sec และจากกราฟพบว่ามีอัตราการเพจจิ้งในช่วงท้ายของการล็อค
3. Available Bytes มีค่าเฉลี่ยประมาณ 1MB. หมายถึงค่าของจำนวน หน่วยความจำเวอร์ชวล ที่ยังสามารถใช้ได้ซึ่งในที่นี้ค่าavailable bytes มีค่าประมาณเท่ากับเทรสโฮลที่กำหนดคือ1MB เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ WindowsNT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอพพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวลไว้ ซึ่งอาจจะทำให้ ประสิทธิภาพของหน่วยความจำต่ำลง
4. Commit Limit มีค่าประมาณ 57 MB ซึ่ง Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิทได้โดยไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ paging file.sys รวมกับขนาด RAM ที่สามารถเปลี่ยนไปยังดิสก์ได้ (หรือจำนวนของแรมที่เหลือจากการใช้เป็น ระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่าเฉลี่ยประมาณ 45 MB. ซึ่งถือว่าไม่มีปัญหาเพราะมีค่าไม่เกินขนาดของ Commit Limit
5. Page Faults/sec มีค่า 29.649 เมื่อเทียบกับค่า Transition Faults/sec ซึ่งเท่ากับ 12.793 จะเห็นว่าทั้ง 2 ค่าไม่ใกล้เคียงกัน จึงไม่ใช่เป็นคอขวดที่หน่วยความจำ
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB ในที่นี้หน่วยความจำฟิสิคอล เท่ากับ 64 MB. เพราะฉะนั้นค่า Pool Nonpaged Bytes ควรมีค่า 60 MB. ขึ้นไป
7. ดูค่า Page Reads/sec เทียบกับค่าDisk Reads/sec ของไดร์ฟต่างๆ มีค่าไม่ใกล้เคียงกับของไดร์ฟใดเลย
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่างๆ ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}}$ * 4096 bytes/page = 5383 bytes/read เทียบกับค่า

   Avg. Disk Bytes/Read ได้ว่ามีค่าไม่ใกล้เคียงกับ ไดร์ฟใดเลย
10. $\frac{\text{Page Outputs/sec}}{\text{Page Writes/sec}}$ * 4096 bytes/page = 9668 bytes/write เทียบกับค่า

    Avg. Disk Bytes/Write ซึ่งมีค่าไม่ใกล้เคียงกับ ไดร์ฟใดเลย

สรุปว่าเกิดคอขวดที่หน่วยความจำขึ้นในช่วงท้ายของการล็อค

การตรวจจับจุดคอขวดที่ดิสก์

1. %Disk Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไดร์ฟซี: มีค่าเฉลี่ย 21.008 ซึ่งถือว่าค่อนข้างต่ำ แต่มีค่ามากที่สุด เป็น 100% และจากกราฟ %Disk Time จะมีค่าถึง 100% ในช่วงท้ายของการล็อคซึ่งสัมพันธ์กับกราฟของหน่วยความจำ
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการเข้าถึงไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าเฉลี่ย 34.262 แต่มีค่า ค่ามากที่สุด เป็น 100% และจากกราฟ จะมี %Disk Time ขึ้นสูงในช่วงท้ายของการล็อคเช่นเดียวกับ ไดร์ฟซี และสัมพันธ์กับกราฟของ หน่วยความจำ เช่นกัน

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีค่าประมาณ 9K แสดงว่าการโอนย้ายข้อมูลยังไม่มีประสิทธิภาพเท่าที่ควร เพราะค่านี้ควรจะมีค่ามากกว่า 20K จึงจะถือว่าการโอนย้ายข้อมูลมีประสิทธิภาพดี
- ไดร์ฟดี: มีค่าประมาณ 0K แสดงว่าไม่มีการโอนย้ายข้อมูลที่ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าประมาณ 1K แสดงว่าประสิทธิภาพการโอนย้ายข้อมูลต่ำมาก

3. Avg. Disk Sec/transfer

- ไดร์ฟซี: มีค่า 0.11 ซึ่งค่อนข้างต่ำ แสดงว่า ดิสก์ใช้เวลาในการโอนย้ายข้อมูลน้อย เพราะถ้าค่านี้มีค่ามากกว่า 0.3 sec. แสดงว่าตัวควบคุมดิสก์พยายามที่จะเข้าถึงดิสก์ซ้ำๆเนื่องจากการอ่านหรือการเขียนเกิดผิดพลาด
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการโอนย้ายข้อมูลที่ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่า 0.015 ซึ่งค่าต่ำเช่นกัน

4. Disk Queue Length

- ไดร์ฟซี: จากกราฟจะเห็นว่าส่วนใหญ่เกิดคิวในช่วงที่มีการใช้หน่วยความจำ และ ดิสก์ และเกิด ค่า Queue Length มากที่สุด เป็น 3 และมีค่าเฉลี่ย 0.303 จึงน่าจะเกิด คอขวด ที่ช่วงท้ายของการล็อค
- ไดร์ฟดี: เป็นศูนย์ จึงไม่น่าจะมีปัญหาอะไร
- ไดร์ฟอี: มี ค่า Queue Length มากที่สุด เป็น 10 และมีค่าเฉลี่ยที่ 0.545 และจากกราฟจะเกิดคิวส่วนใหญ่ในช่วงท้ายของการล็อคซึ่งสัมพันธ์กับช่วงที่มีการใช้หน่วยความจำและดิสก์มาก ดังนั้นจึงน่าจะมีปัญหาในช่วงท้ายของการล็อค

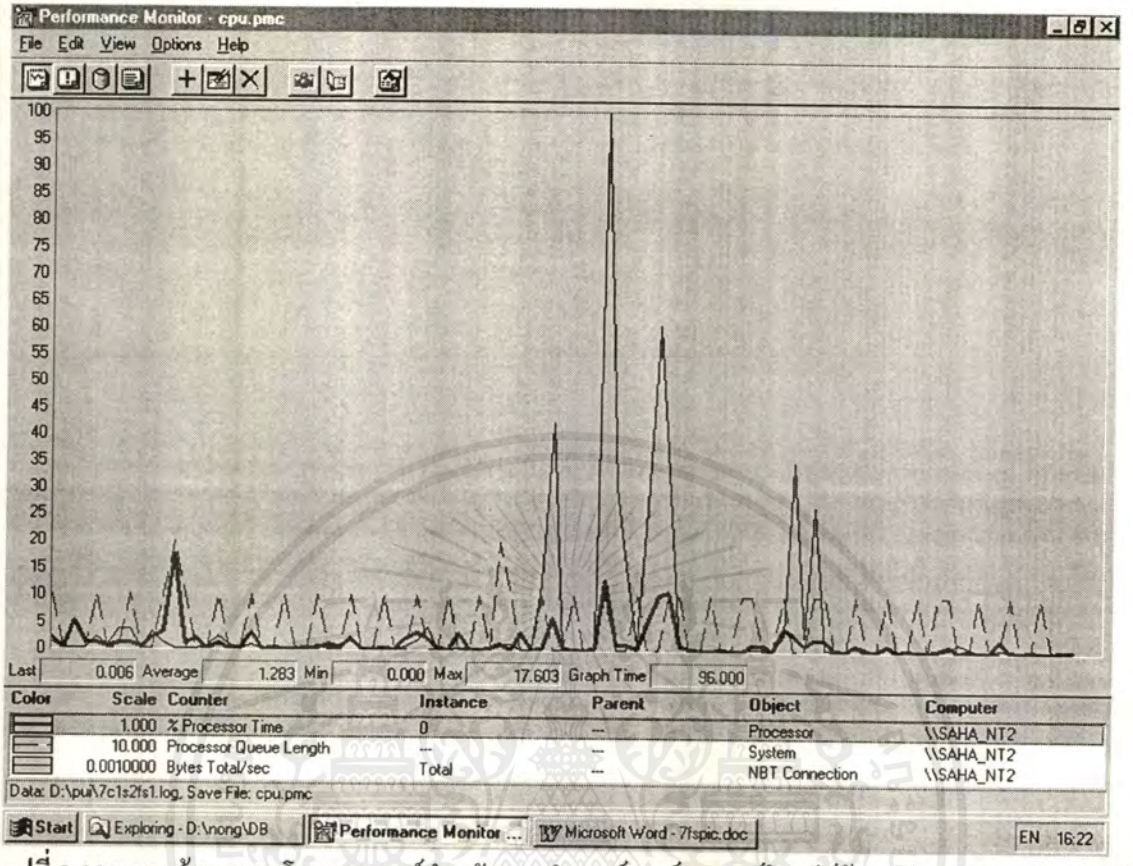
  สรุปในการทำเคาน์เตอร์ออเดอร์ โดยล็อคอิน ห่างกัน 1 วินาทีนี้เกิดคอขวดที่ดิสก์ขึ้นในช่วงท้ายของการล็อค เนื่องจากการเกิดคอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่แคช

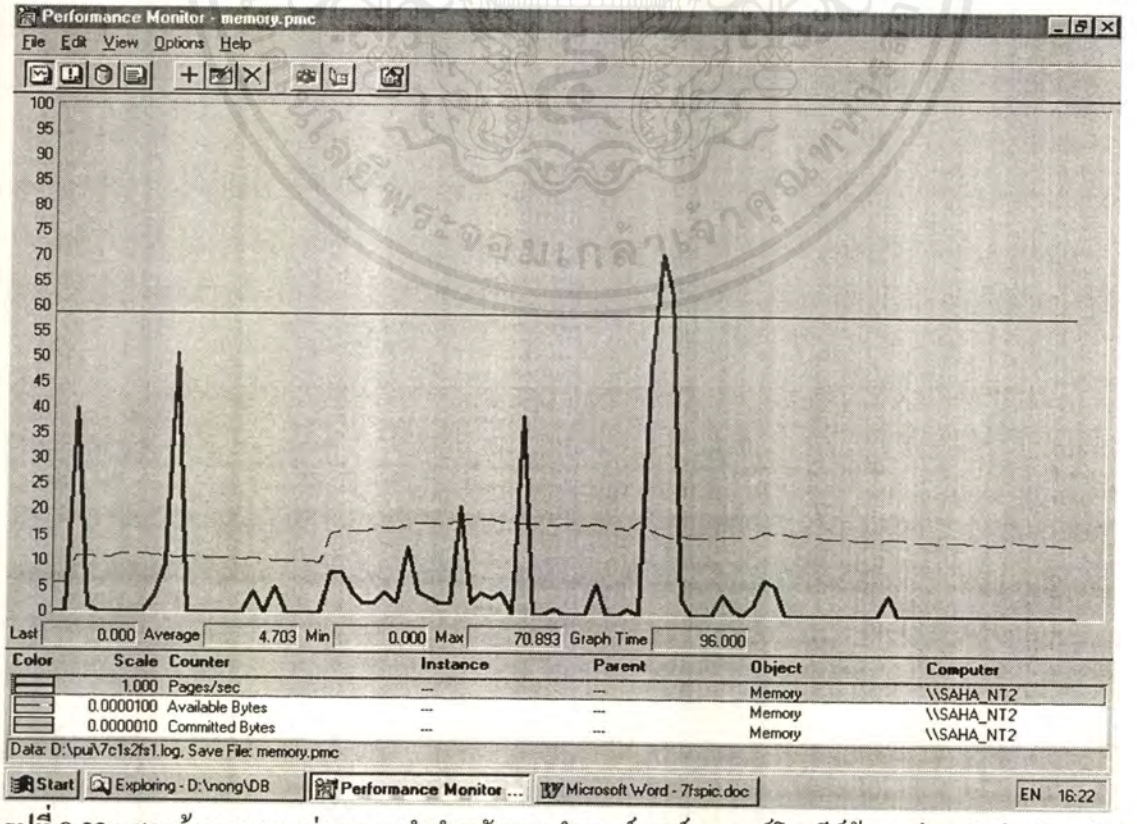
1. Copy Read Hits% เมื่อดูจากกราฟ อัตราการ ฮิต จะสูงขึ้นจนถึง 100% ในช่วงที่มีการใช้ หน่วยความจำ และ ดิสก์ มาก
2. Data Flushes/sec มีค่า 1.37 ซึ่งหมายถึง แคชมีการนำข้อมูลออก โดยได้รับการร้องขอจาก ระบบไฟล์ ให้นำข้อมูลออกไปยัง อุปกรณ์ต่อพ่วง เพียง 1.37
3. Data Map Hits% จากกราฟจะเห็นว่า กิจกรรมเกี่ยวกับไดเรกทอรี่ หรือกิจกรรมที่เกี่ยวกับไฟล์จำนวนมากนั้นมีค่าสูงเป็นระยะ ๆ ตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดการล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เคาน์เตอร์ออเดอร์ คีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที ได้ผลดังนี้

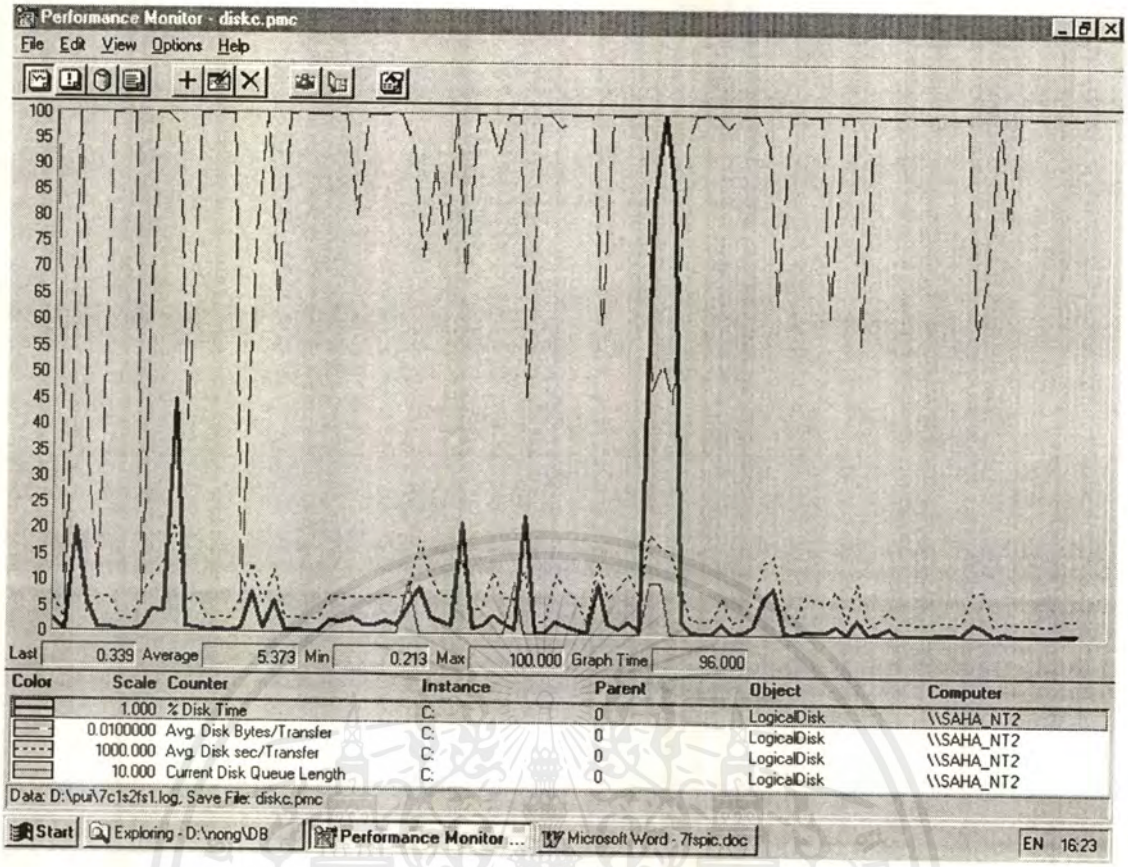


รูปที่ 8-22 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1วินาที

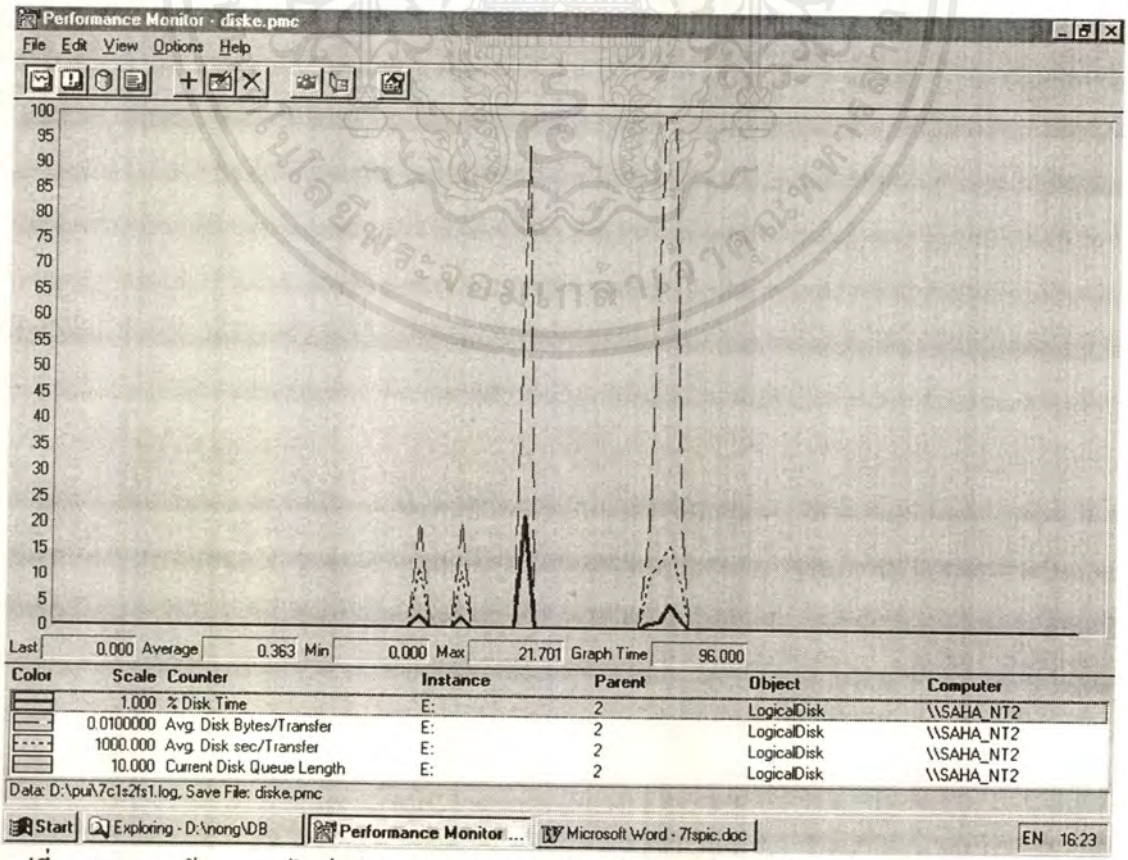


รูปที่ 8-23 แสดงข้อมูลของหน่วยความจำสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

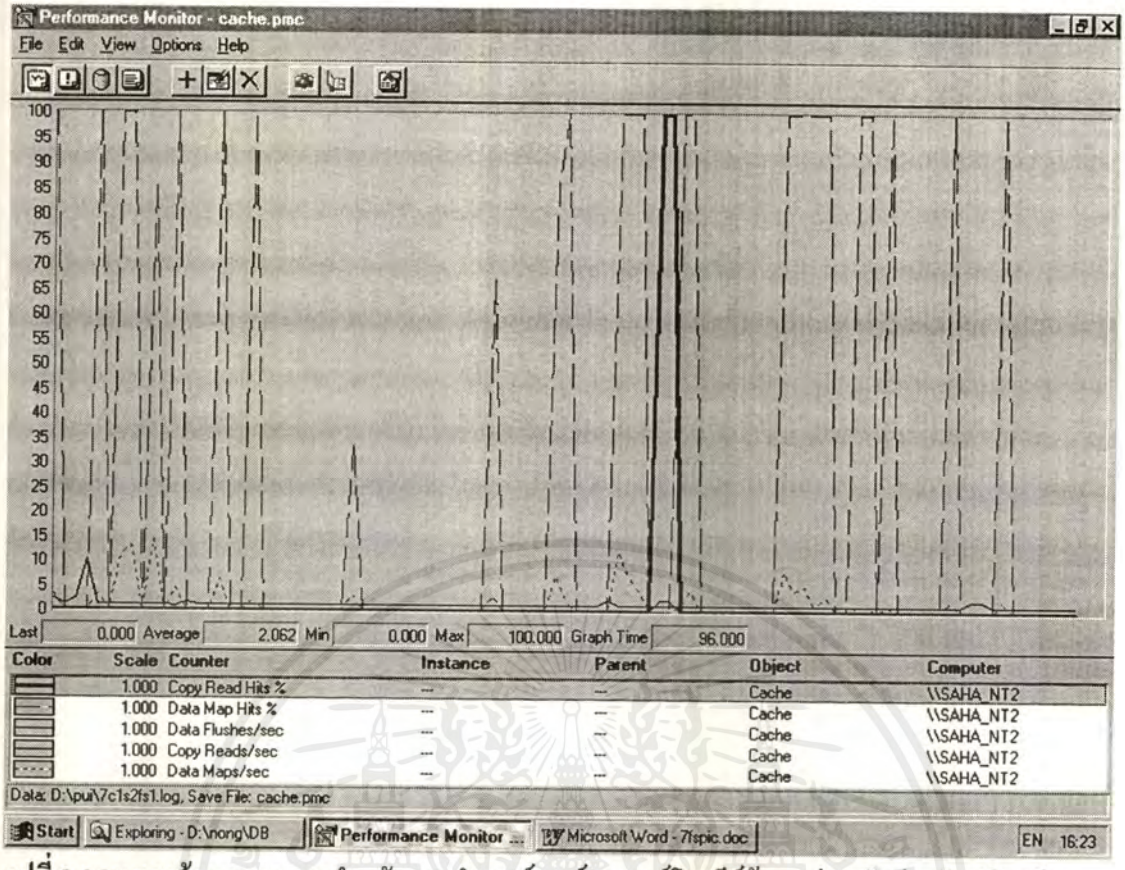


รูปที่ 8-24 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1วินาที



รูปที่ 8-25 แสดงข้อมูลของไดร์ฟอีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-26 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1วินาที

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. %Processor Time มีค่าเฉลี่ย 1.283 และ ค่ามากที่สุด ที่ 17.603 ซึ่งถือว่ามีค่าไม่มากนัก
2. Processor Queue Length จากกราฟมีคิวเกิดขึ้นตลอดการล็อค ส่วนใหญ่แล้วมี Queue Length เป็น 1 , มีค่ามากที่สุด เป็น 2 และค่าเฉลี่ย เป็น 0.402 จึงไม่น่าจะมีปัญหาโพรเซสเซอร์ เกิดขึ้น เมื่อดูจากค่าทั้งสองข้างต้น โพรเซสเซอร์ ไม่น่าจะเป็น คอขวด

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยที่ 4.703 แสดงว่าปริมาณเพจที่อ่านเข้ามาและเขียนออกไปยังดิสก์ จริง ๆ นั้น ไม่มากนัก เพราะค่านี้ควรจะมีค่าน้อยกว่า 5 pages/sec และจากกราฟพบว่ามีอัตราการเพจจิ้งบางช่วงจะมาก ถึงประมาณ 40-70 pages/sec
2. Available Bytes มีค่าเฉลี่ยประมาณ 1MB. หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ค่าavailable bytes มีค่าประมาณเท่ากับเทรสโฮลที่กำหนดคือ 1MB เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ WindowsNT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอพพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวลไว้ ซึ่งอาจจะทำให้ ประสิทธิภาพของหน่วยความจำ ต่ำลง
3. Commit Limit มีค่าประมาณ 59 MB ซึ่ง Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิทได้ โดยไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ paging file.sys รวมกับขนาด RAM ที่สามารถเปลี่ยนไปยังดิสก์ได้ (หรือจำนวนของแรมที่เหลือจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้เป็นระบบปฏิบัติการแล้ว)

4. Committed Bytes มีค่าเฉลี่ยประมาณ 58 MB. ซึ่งถือว่าไม่มีปัญหาเพราะมีค่าไม่เกินขนาดของ Commit Limit
5. Page Faults/sec มีค่า 13.69 เมื่อเทียบกับค่า Transition Faults/sec ซึ่งเท่ากับ 6.99 จะเห็นว่าทั้ง 2 ค่าไม่ใกล้เคียงกัน จึงไม่ใช่เป็นคอขวดที่หน่วยความจำ
5. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB ในที่นี้หน่วยความจำฟิสิคอลเท่ากับ 64 MB. เพราะฉะนั้นค่า Pool Nonpaged Bytes ควรมีค่า 60 MB. ขึ้นไป
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของ ไดรว์ต่าง ๆ มีค่าใกล้เคียงกับของ ไดรว์ซี
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่างๆซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}}$ * 4096 bytes/page = 5382 bytes/read เทียบกับค่า

   Avg. Disk Bytes/Read ได้ว่ามีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
10. $\frac{\text{Page Outputs/sec}}{\text{Page Writes/sec}}$ * 4096 bytes/page = 10679 bytes/write เทียบกับค่า

    Avg. Disk Bytes/Write ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย

    สรุปว่าเกิดคอขวดที่หน่วยความจำ ขึ้นในช่วงที่ pages/sec ประมาณ 70

การตรวจจับจุดคอขวดที่ดิสก์

1. %Disk Time

- ไดร์ฟซี: มีค่าเฉลี่ย 5.373 ซึ่งถือว่าค่อนข้างต่ำ แต่มีค่า ค่ามากที่สุด เป็น 100% และจากกราฟ %Disk Time จะมีค่าถึง 100% ตรงกับช่วงที่มี pages/sec เท่ากับ 70
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการเข้าถึง ไดร์ฟดี เลย
- ไดร์ฟอี: มีค่าเฉลี่ย 0.363 แต่มีค่ามากที่สุด เพียง 21.701 จึงไม่น่าจะเป็นปัญหาอะไร

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีค่าประมาณ 12K แสดงว่าการโอนย้ายข้อมูลยังไม่มีประสิทธิภาพเท่าที่ควร เพราะค่านี้ควรจะมีค่ามากกว่า 20K จึงจะถือว่าการโอนย้ายข้อมูลมีประสิทธิภาพดี
- ไดร์ฟดี: มีค่าประมาณ 0K แสดงว่าไม่มีการโอนย้ายข้อมูลที่ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่าประมาณ 441 bytes แสดงว่าประสิทธิภาพการโอนย้ายข้อมูลต่ำมาก

3. Avg. Disk Sec/transfer

- ไดร์ฟซี: มีค่า 0.007 ซึ่งค่อนข้างต่ำ แสดงว่าดิสก์ใช้เวลาในการโอนย้ายข้อมูลน้อย เพราะถ้าค่านี้มีค่ามากกว่า 0.3 sec.แสดงว่าตัวควบคุมดิสก์พยายามที่จะเข้าถึงดิสก์ซ้ำๆเนื่องจากการอ่านหรือการเขียนเกิดผิดพลาด
- ไดร์ฟดี: มีค่า 0 และเมื่อดูจากกราฟแสดงว่าไม่มีการโอนย้ายข้อมูลที่ไดร์ฟดีเลย
- ไดร์ฟอี: มีค่า 0.001 ซึ่งค่าต่ำเช่นกัน

4. Disk Queue Length

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไดร์ฟซี: จากกราฟจะเห็นว่ามีคิวเกิดขึ้นน้อยมาก และ ค่า Queue Length มากที่สุด เพียง 1 เท่านั้น และมีค่าเฉลี่ย 0.041 จึงไม่น่าจะเกิดคอขวดขึ้น
- ไดร์ฟดี: เป็นศูนย์ จึงไม่น่าจะมีปัญหาอะไร
- ไดร์ฟอี: มี ค่า Queue Length มากที่สุด เป็น 10 และมีค่าเฉลี่ยที่ 0.545 และจากกราฟจะเกิดคิวส่วนใหญ่ในช่วงท้ายของการล็อคซึ่งสัมพันธ์กับช่วงที่มีการใช้หน่วยความจำและดิสก์มาก ดังนั้นจึงน่าจะมีปัญหาในช่วงท้ายของการล็อค

  สรุปในการทำเคาน์เตอร์ออเดอร์ โดยคีย์ข้อมูลส่วนหัว ห่างกัน 1 วินาทีนี้เกิดคอขวดที่ดิสก์ เนื่องจากการเกิด คอขวดที่หน่วยความจำ

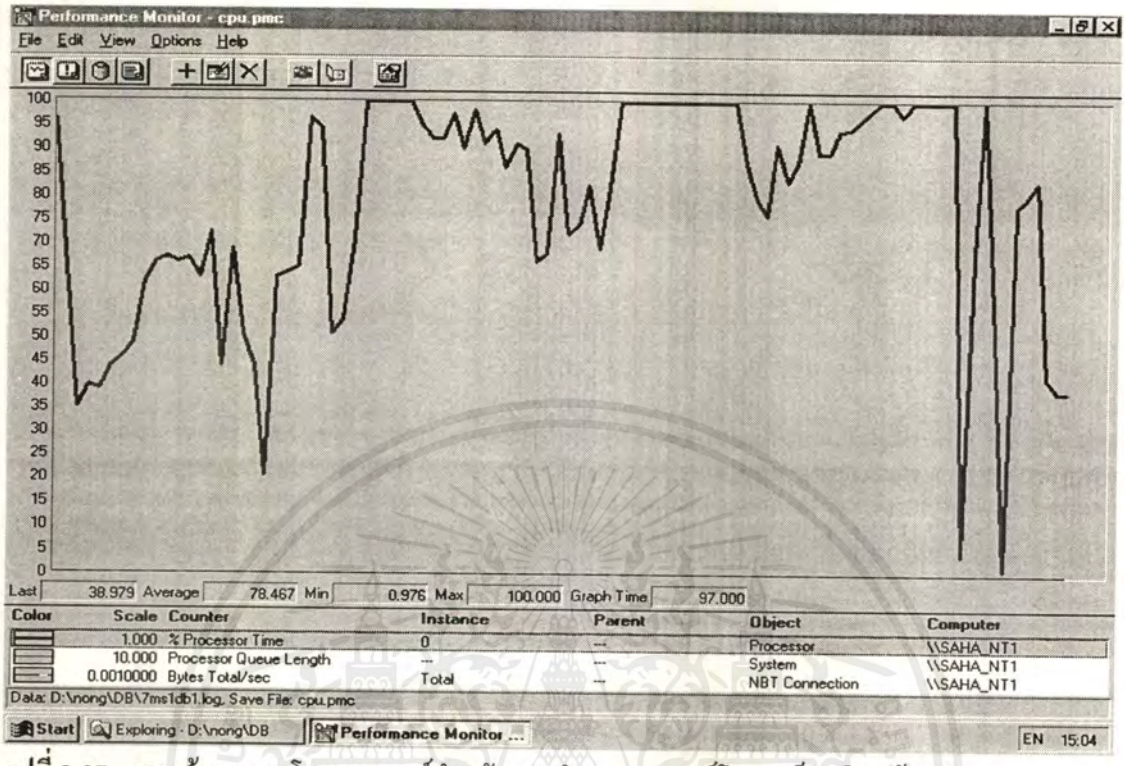
การตรวจจับจุดคอขวดที่แคช

1. Copy Read Hits% เมื่อดูจากกราฟอัตราการฮิตจะสูงขึ้นจนถึง 100% เพียงครั้งเดียวในช่วงกลาง ๆ ของการล็อค
2. Data Flushes/sec มีค่า 1.147 ซึ่งหมายถึง แคช มี การนำข้อมูลออก โดยได้รับการร้องขอจาก ระบบไฟล์ ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วง เพียง 1.147
3. Data Map Hits% จากกราฟจะเห็นว่า กิจกรรมเกี่ยวกับไดเรกทอรี่หรือกิจกรรมที่เกี่ยวกับไฟล์จำนวนมากนั้นมีค่าสูงเป็นระยะ ๆ ตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดการล็อค

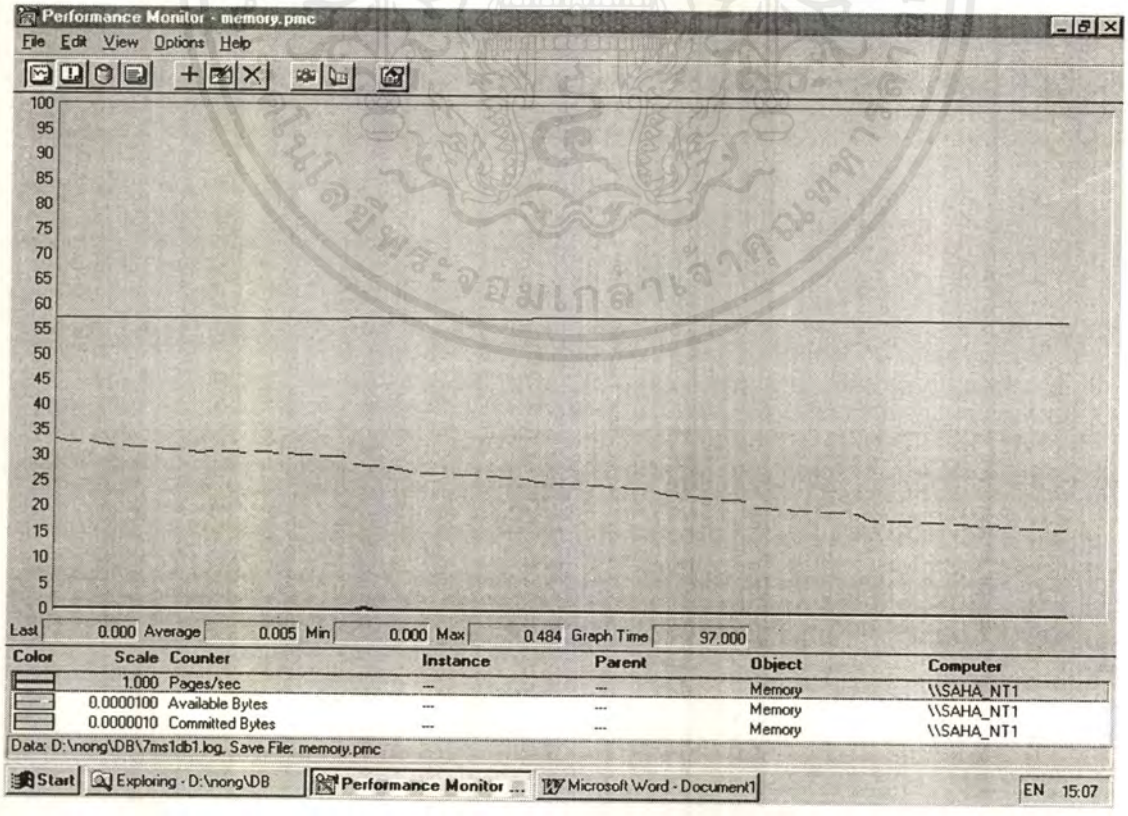
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดาต้าเบสเซิร์ฟเวอร์

1. แมสออเดอร์ กดล็อกอินพร้อมกัน ได้ผลดังนี้

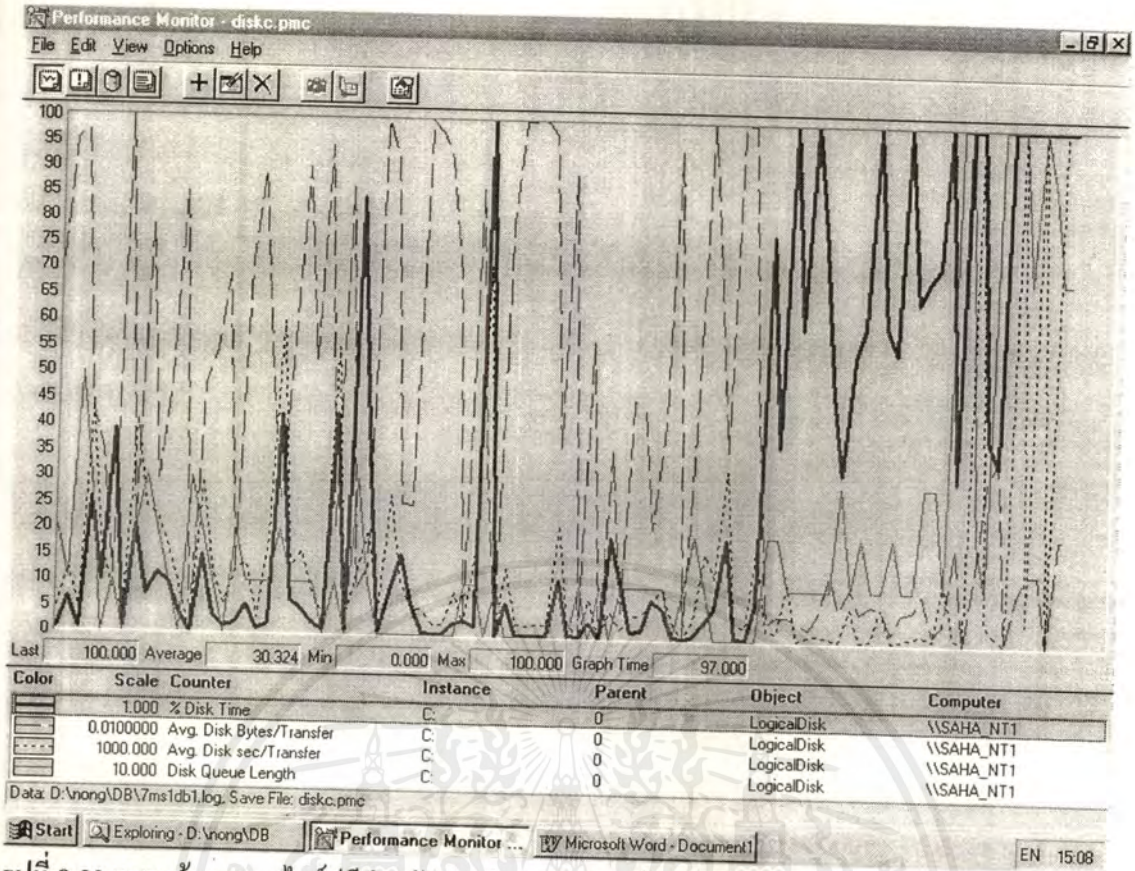


รูปที่ 8-27 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยกดล็อกอินพร้อมกัน

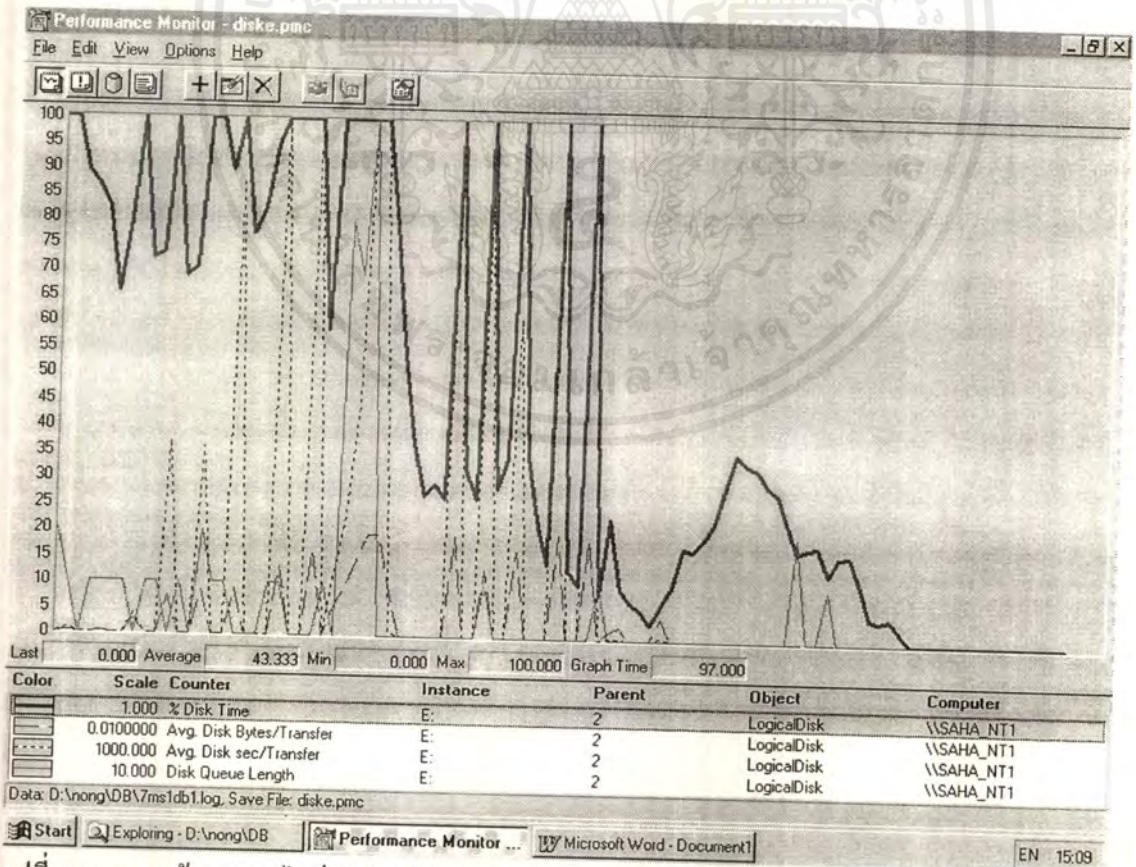


รูปที่ 8-28 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยกดล็อกอินพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

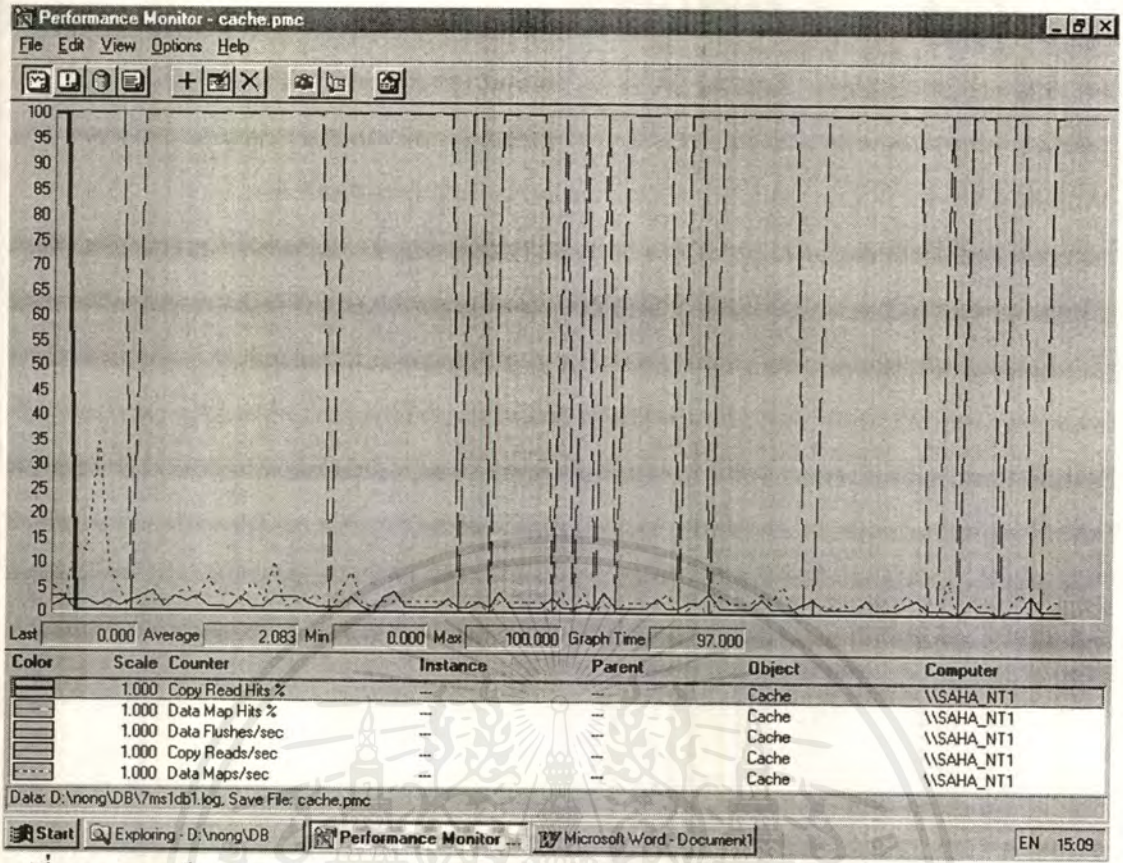


รูปที่ 8-29 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยกดล็อกอินพร้อมกัน



รูปที่ 8-30 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยกดล็อกอินพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-31 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยกดล็อกอินพร้อมกัน

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. % Processor Time มีค่าเฉลี่ย 79.112% เพื่อดูปริมาณการใช้โพรเซสเซอร์ในกระบวนการทำงาน
2. Processor Queue Length มีค่าที่เวลาหนึ่งเป็น 0 แสดงว่าไม่มีเทรดในคิวของโพรเซสเซอร์ที่กำลังรอการใช้โพรเซสเซอร์อยู่

สรุปไม่เกิดคอขวดที่โพรเซสเซอร์

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเพียง 0.010 เพจ ซึ่งหมายความว่ามีเพียง 0.010 เพจที่จะต้องเขียนหรืออ่านจาก disk และบรรจุเข้าไปใน หน่วยความจำฟิสิคอล ค่านี้จะบ่งบอกว่าในระบบของคุณจะต้องการ หน่วยความจำฟิสิคอล เพิ่มขึ้นหรือไม่ ถ้าค่านี้มีค่าเกิน 5 pages/sec สำหรับดิสก์เดียว ก็หมายความว่าอาจจะเกิด คอขวดที่หน่วยความจำ
2. Available Bytes มีค่า 1.36 MB หมายถึงค่าของจำนวน หน่วยความจำเวอร์ชวล ที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่า เทรสโฮล ที่กำหนดคือ 1MB. เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ Windows NT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอพพลิเคชั่น มาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวลไว้ ซึ่งอาจจะทำให้ประสิทธิภาพของหน่วยความจำต่ำลง
3. Commit Limit มีค่าเท่ากับ 329.14 MB ,Commit Limit คือ ขนาดของ หน่วยความจำเวอร์ชวลที่สามารถ คอมมิทได้ โดยไม่ต้องเพิ่มขนาดของ เพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับ ขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ paging file.sys รวมกับ ขนาดของ RAM ที่สามารถเปลี่ยนไปยังดิสก์ ได้ (หรือจำนวนของ RAM ที่ว่าง ที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)

4. Committed Bytes มีค่า 55.3 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์
5. ดูค่า Page Faults/sec ซึ่งมีค่า 4.421 page faults เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.021 page faults ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำแต่กำลังถูกเขียนลงบนดิสก์ ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)
6. Pool Nonpaged Bytes มีค่าเท่ากับ 1.87 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของ ไดร์ฟต่าง ๆ ซึ่งมีค่าไม่ใกล้เคียงกันเลย
8. ดูค่า Page Writes/secเทียบกับค่า DiskWrites/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้เป็น0ใกล้เคียงกับไดร์ฟดี
9. Page Inputs/sec * 4096 bytes/page = 4096 bytes/read เทียบกับค่า
   Page Reads/sec
   Avg. Disk Bytes/Read ซึ่งมีค่าใกล้เคียงกับ ไดร์ฟซี
10. Page Outputs/sec * 4096 bytes/page = 0 bytes/write เทียบกับค่า
    Page Writes/sec
    Avg. Disk Bytes/Write ซึ่งใกล้เคียงกับ ไดร์ฟดี

สรุปไม่เกิด คอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่ดิสก์

1. % Disk Time

- ไดร์ฟซี: มี % Disk Time สูงถึง 100%
- ไดร์ฟดี: มี % Disk Time เท่ากับ 0
- ไดร์ฟอี: มี % Disk Time สูงถึง 100% เท่ากับ ไดร์ฟซี

2. Avg. Disk Bytes/Transfer

- ไดร์ฟซี: มีจำนวน ไบต์ในการ โอนย้าย 4531 bytes ซึ่งค่านี้นับว่าต่ำมากควรจะมีการโอนย้ายข้อมูลประมาณ 20KB
- ไดร์ฟดี: ไม่มีจำนวนไบต์ในการโอนย้าย
- ไดร์ฟอี: มีจำนวนในการโอนย้ายข้อมูลที่ใกล้เคียงกับ ไดร์ฟซี คือ มีค่าเท่ากับ4610.095 bytes

3. Avg. Disk sec/transfer

- ไดร์ฟซี: มีเวลาการโอนย้ายข้อมูลใน 1 ครั้ง คือ 0.129 sec ซึ่งเป็นอัตราการ โอนย้าย ข้อมูลที่เร็ว เพราะมีค่าต่ำกว่า 0.3 วินาที
- ไดร์ฟดี: มีเวลาการโอนย้าย เท่ากับ 0 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไดร์ฟอี: มีเวลาการโอนย้ายข้อมูลเท่ากับ 0.071 วินาที

4. Disk Queue Length

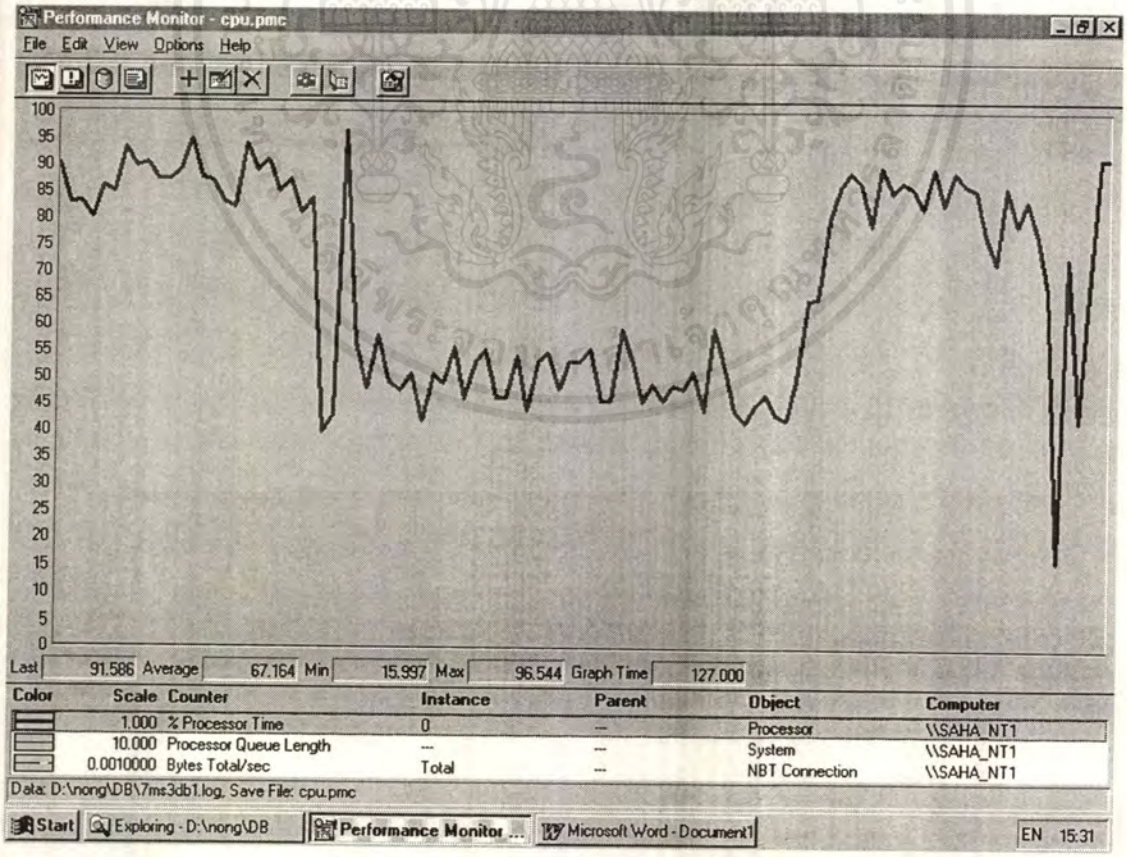
- ไดร์ฟซี: มีค่าเฉลี่ยของคิวสูง2.104 คิว และมีค่า คิวสูงสุดถึง 17 คิว ซึ่งอาจจะสรุปได้ว่าเกิด คอขวดที่ดิสก์ ขึ้นที่ ไดรว์ซี เพราะมีค่าของคิวสูงเกิน 2 คิว
- ไดร์ฟดี: ไม่มีคิวใน Disk Queue Length
- ไดร์ฟอี: มีค่าเฉลี่ยของคิวเป็น 0.865 ซึ่งมีค่าต่ำมาก และช่วงตรงกลางของกราฟก็มีค่าสูงสุดของคิวถึง 45 คิว

สรุปเกิดคอขวดที่ดิสก์ขึ้นที่ไดร์ฟซี และ เป็นบางช่วงที่ไดร์ฟอี

การตรวจจับจุดคอขวดที่แคช

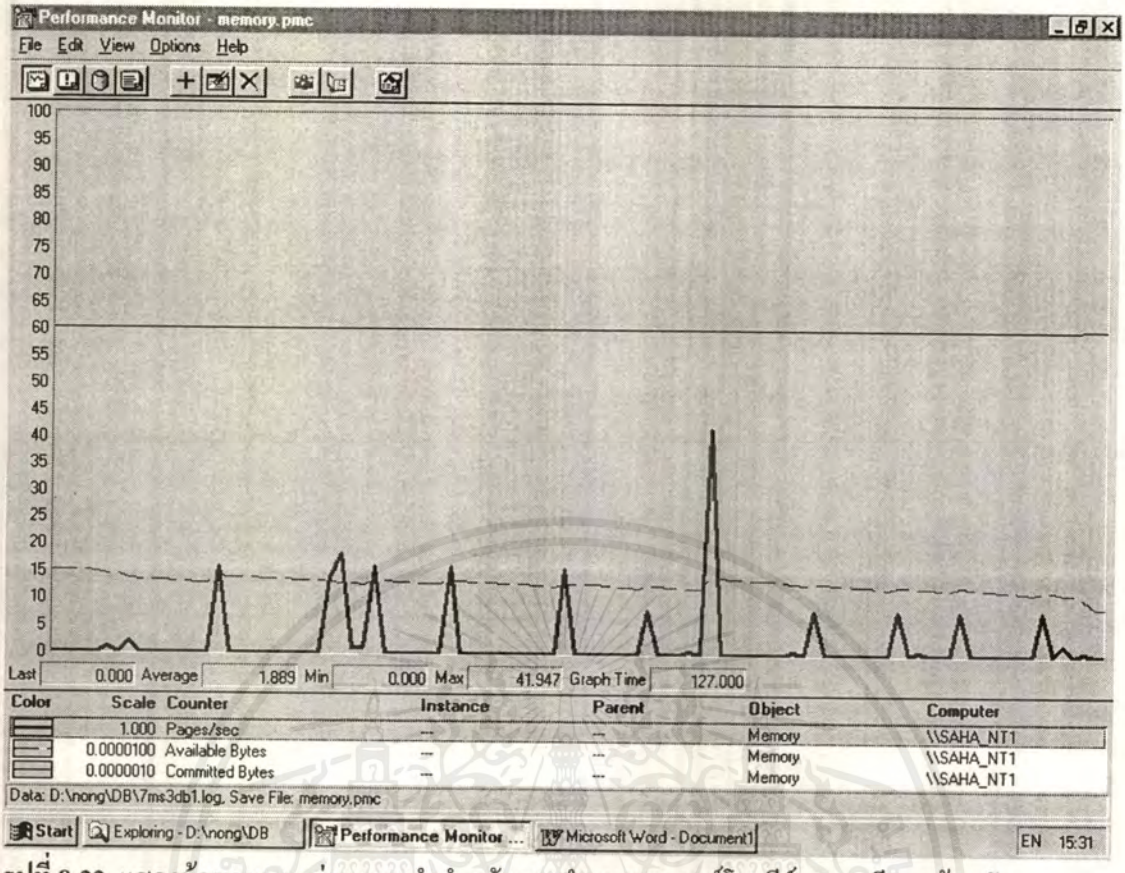
1. Copy Read Hits % ดูเทียบกับค่า Copy Reads/sec ซึ่งค่า Copy Read Hits % มีค่าสูงถึง 100% เมื่อเทียบกับ Copy Reads/sec ซึ่งมีค่าเพียง 0.062 คือมีอัตราการฮิตที่สูงแต่อัตราที่อ่านได้ต่ำ
2. Data Flushes/sec มีค่า 1.628 ซึ่งหมายถึงแคชมี การนำข้อมูลออก โดยได้รับการร้องขอจาก ระบบไฟล์ ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วง เพียง 1.628
3. ค่า Data Map Hits % มีค่าสูงถึง 100% แต่ค่า Data Maps/sec มีค่าต่ำเพียง 2.824 อาจจะทำให้ ประสิทธิภาพไม่ดีก็ได้ถ้าอัตราการฮิตสูงแต่อัตราการแมปต่ำ

2. แมสออเดอร์ คีย์รายละเอียดพร้อมกัน ได้ผลดังนี้

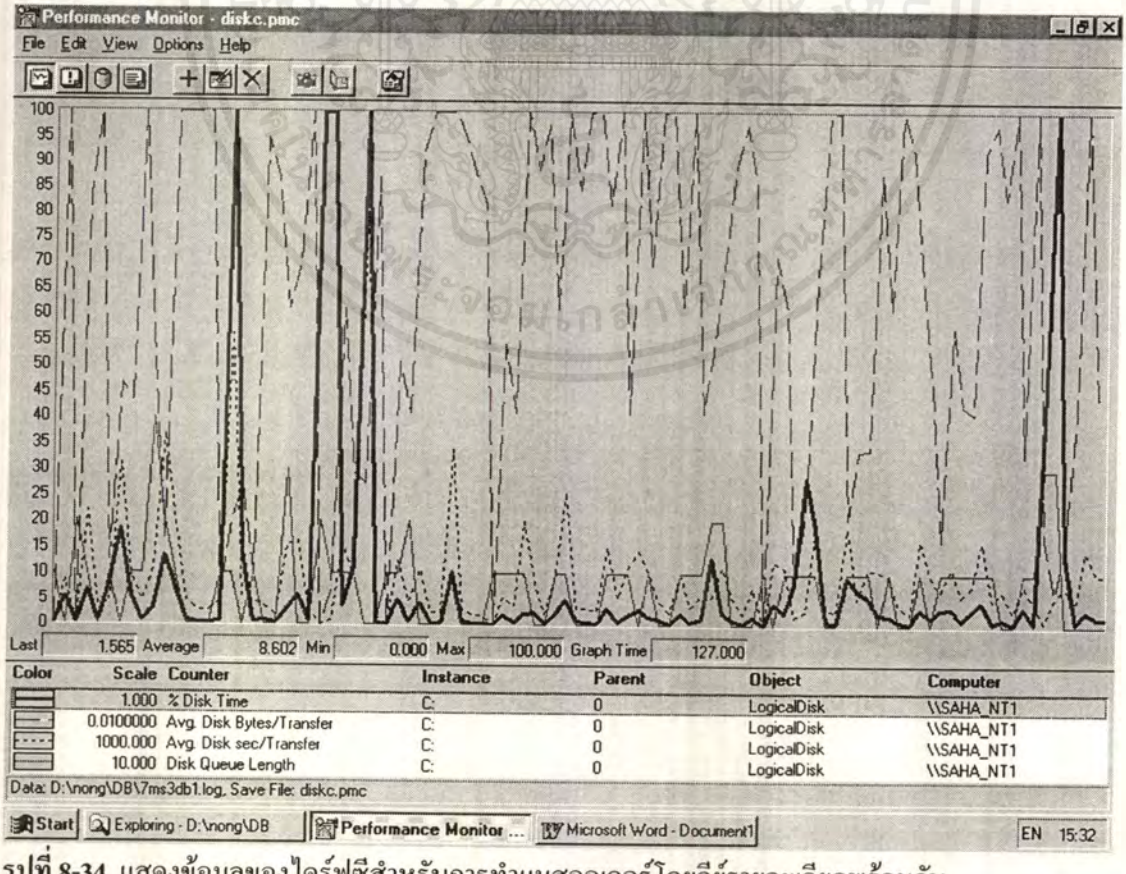


รูปที่ 8-32 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

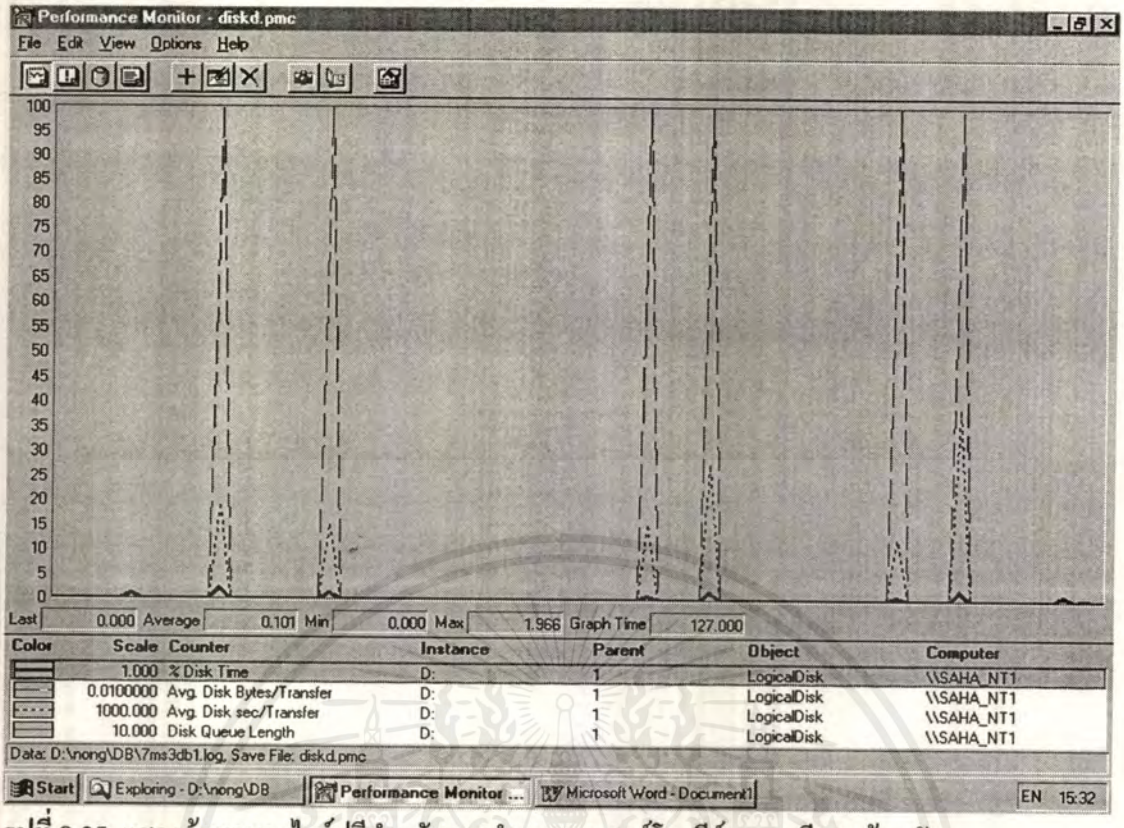


รูปที่ 8-33 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน

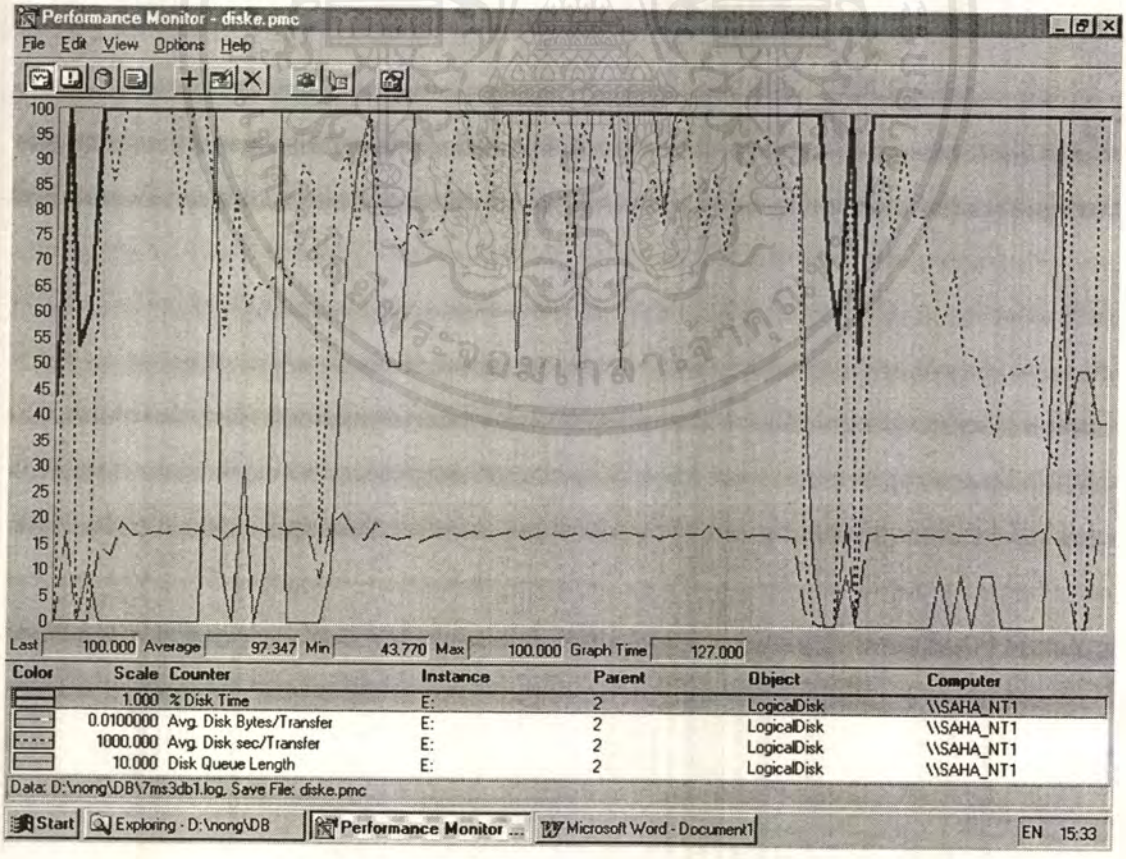


รูปที่ 8-34 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

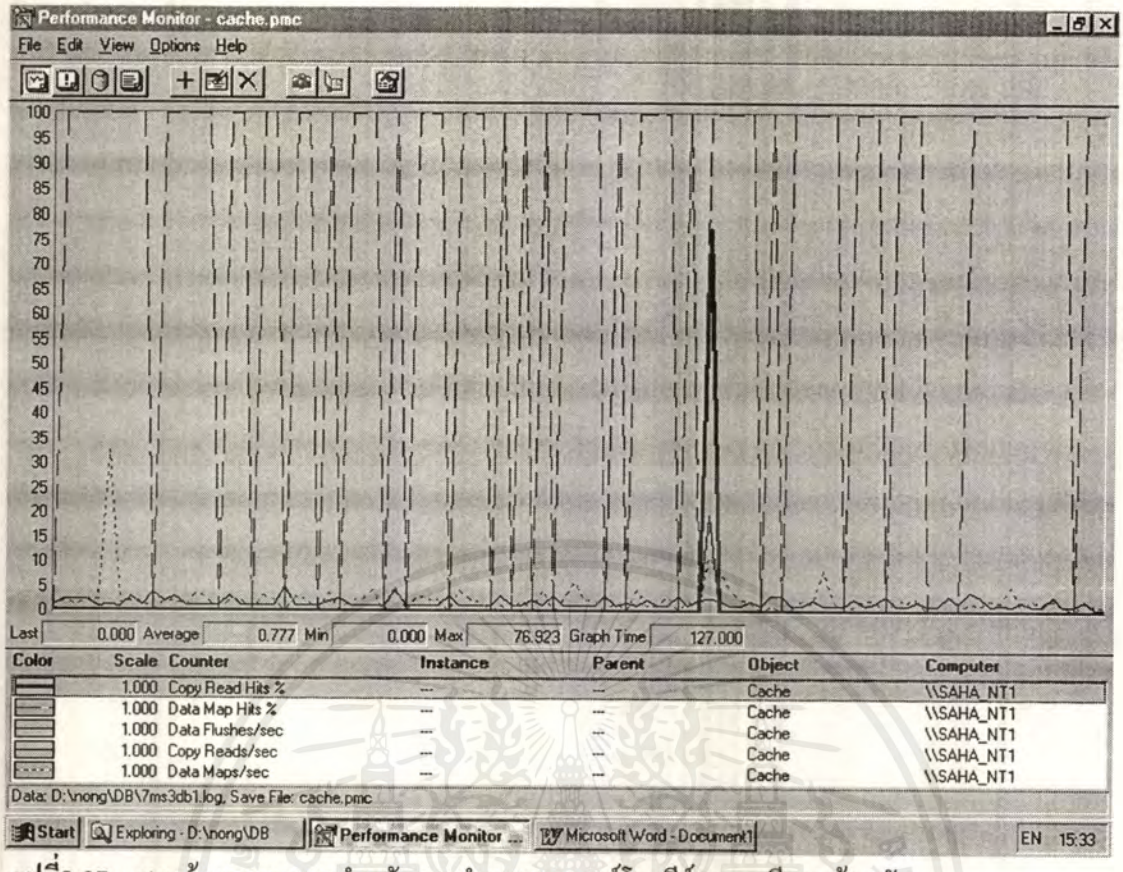


รูปที่ 8-35 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน



รูปที่ 8-36 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่8-37 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดพร้อมกัน

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. % Processor Time มีค่าเฉลี่ย 67.937% เพื่อดูปริมาณการใช้ โพรเซสเซอร์ ในกระบวนการทำงาน
2. Processor Queue Length มีค่าที่เวลาหนึ่งเป็น 0 แสดงว่าไม่มีเทรดในคิวของโพรเซสเซอร์ ที่กำลังรอการใช้โพรเซสเซอร์อยู่

สรุปไม่เกิด คอขวดที่โพรเซสเซอร์

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเพียง 3.308 เพจ ซึ่งหมายความว่ามีเพียง 3.308 เพจ ที่จะต้องเขียนหรืออ่านจากดิสก์ และบรรจุเข้าไปใน หน่วยความจำฟิสิคอล ค่านี้จะบ่งบอกว่าในระบบของคุณจะต้องการ หน่วยความจำฟิสิคอล เพิ่มขึ้นหรือไม่ ถ้าค่านี้มีค่าเกิน 5 pages/sec สำหรับ ดิสก์เดียว ก็หมายความว่าอาจจะเกิด คอขวดที่หน่วยความจำ
2. Available Bytes มีค่า 1.36 MB หมายถึงค่าของจำนวน หน่วยความจำเวอร์ชวล ที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่า เทรสโฮล ที่กำหนดคือ 1MB. เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ Windows NT ก็จะค่อย ๆ นำเอาหน่วยความจำจากการวิ่งแอพพลิเคชั่น มาเพื่อรักษาค่าต่ำสุดของ หน่วยความจำเวอร์ชวล ไว้ ซึ่งอาจจะทำให้ ประสิทธิภาพของหน่วยความจำต่ำลง
3. Commit Limit มีค่าเท่ากับ 329.14 MB ,Commit Limit คือ ขนาดของ หน่วยความจำเวอร์ชวล ที่สามารถ คอมมิทได้ โดยไม่ต้องเพิ่มขนาดของ เพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับ ขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ paging file.sys รวมกับ ขนาดของ RAM ที่สามารถ เปลี่ยน ไปยัง ดิสก์ ได้ (หรือจำนวนของ RAM ที่ว่างที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)

4. Committed Bytes มีค่า 57.84 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของเพจจิ้งไฟล์
5. ดูค่า Page Faults/sec ซึ่งมีค่า 4.603 page faults เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.292 page faults ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำแต่กำลังถูกเขียนลงบน ดิสก์ ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะ เปลี่ยนแปลงแก้ไข ข้อมูล)
6. Pool Nonpaged Bytes มีค่าเท่ากับ 1.87 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของ หน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec ที่อ่านได้มีค่าใกล้เคียง 0 เทียบกับค่า Disk Reads/sec ของ ไดร์ฟต่าง ๆ ซึ่งมีค่าใกล้เคียงกับ ไดร์ฟดี ที่สุด
8. ดูค่า Page Writes/sec ที่อ่านได้มีค่าใกล้เคียง 0 เทียบกับค่า Disk Writes/sec ของ ไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าน้อยมากใกล้เคียงกับ 0 และใกล้เคียงกับ ไดร์ฟดี
9. Page Inputs/sec * 4096 bytes/page = 4614 bytes/read เทียบกับค่า
   Page Reads/sec
   Avg. Disk Bytes/Read ซึ่งมีค่าใกล้เคียงกับไดร์ฟดี
10. Page Outputs/sec * 4096 bytes/page = 65498 bytes/write เทียบกับค่า
    Page Writes/sec
    Avg. Disk Bytes/Write ซึ่งมีค่าใกล้เคียงกับไดร์ฟดี

สรุปไม่เกิด คอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่ดิสก์

1. % Disk Time

- ไดร์ฟซี: มี % Disk Time เท่ากับ 11.455%
- ไดร์ฟดี: มี % Disk Time เท่ากับ 0.415%
- ไดร์ฟอี: มี % Disk Time เท่ากับ 100%

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีจำนวน ไบต์ในการโอนย้าย 5792.398 ไบต์ ควรจะมีการโอนย้ายข้อมูลได้ประมาณ 20KB
- ไดร์ฟดี: มีจำนวน ในการโอนย้าย 43593.145 ไบต์ ซึ่งการโอนย้ายของไดร์ฟนี้อยู่ในเกณฑ์ที่น่าพอใจ
- ไดร์ฟอี: มีจำนวนในการโอนย้าย ข้อมูล มีค่าเท่ากับ 3142.399 ไบต์

3. Avg. Disk sec/transfer

- ไดร์ฟซี: มีเวลาการโอนย้าย ข้อมูลใน 1 ครั้ง คือ 0.030 วินาที ซึ่งเป็นอัตราการ โอนย้าย ข้อมูลที่เร็ว เพราะมีค่าต่ำกว่า 0.3 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไดร์ฟดี: มีเวลาการ โอนย้าย เท่ากับ 0.038 วินาที
- ไดร์ฟอี: มีเวลาการ โอนย้าย ข้อมูลเท่ากับ 0.097 วินาที

4. Disk Queue Length

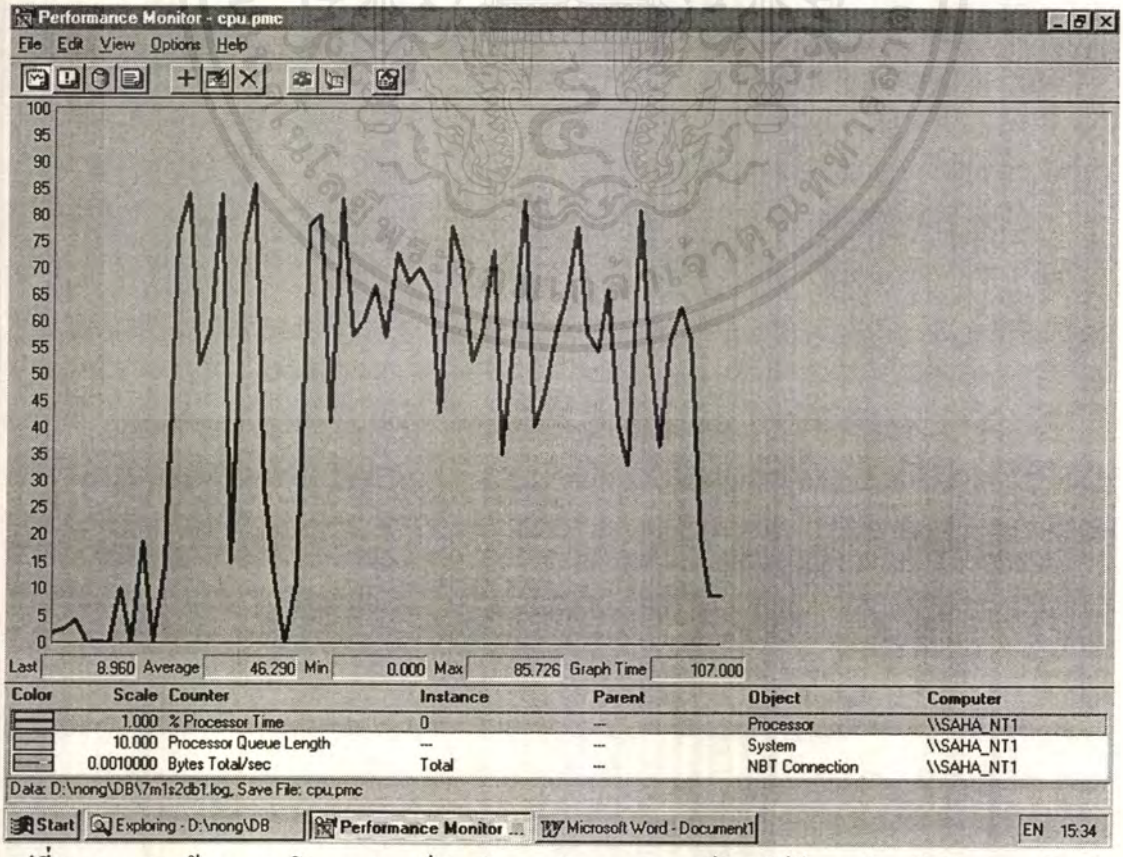
- ไดร์ฟซี: มีค่าเฉลี่ยของคิว 0.717 คิว และมีค่าคิวสูงสุด 4 คิว
- ไดร์ฟดี: ไม่มีคิวใน Disk Queue Length
- ไดร์ฟอี: มีค่าเฉลี่ยของคิว 11.889 คิว และมีค่าคิวสูงสุด 79 คิว

สรุปเกิด คอขวดที่ดิสก์ ขึ้นที่ ไดร์ฟอี

การตรวจจับจุดคอขวดที่แคช

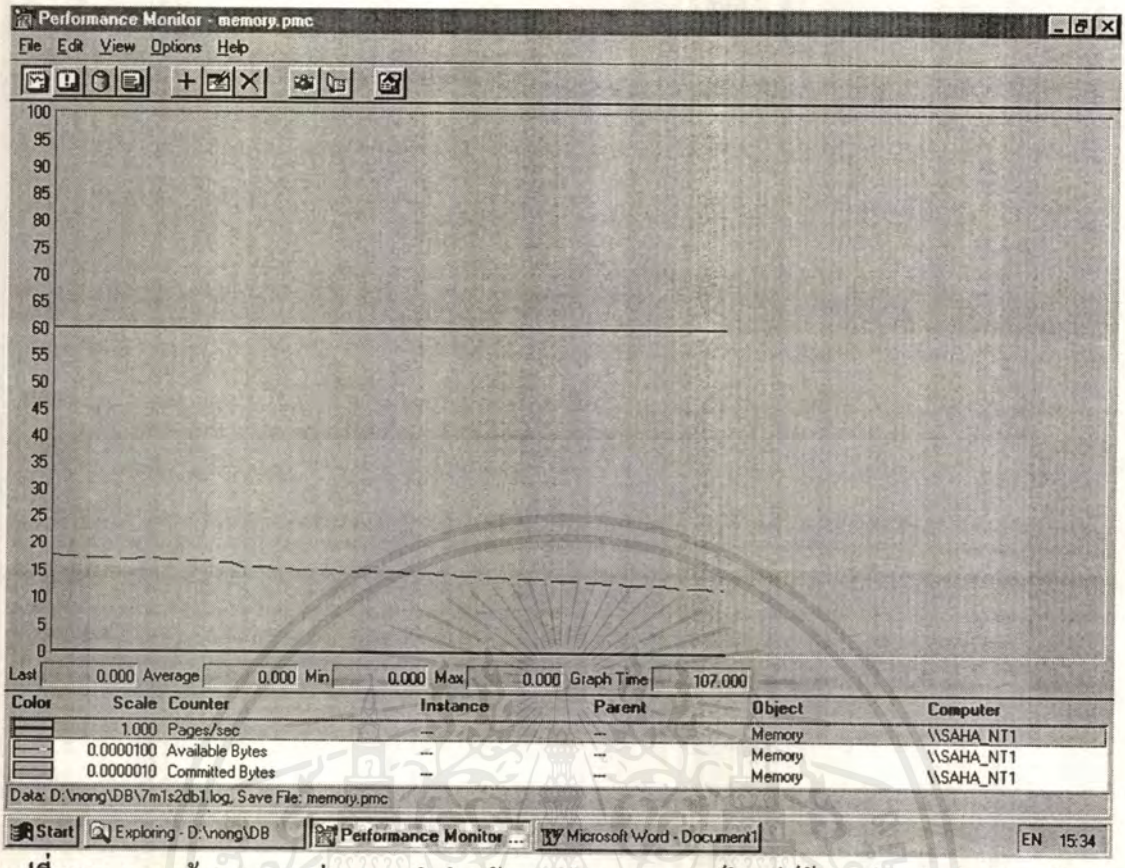
1. Copy Read Hits % คูเทียบกับค่า Copy Reads/sec ซึ่งค่า Copy Read Hits % มีค่าสูงถึง 76.923% เมื่อเทียบกับ Copy Reads/sec ซึ่งมีค่าเพียง 0.308 คือมีอัตราการฮิตที่สูงแต่อัตราที่อ่านได้ต่ำ
2. Data Flushes/sec มีค่า 1.579 ซึ่งหมายถึง แคชมีการนำข้อมูลออก โดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยัง อุปกรณ์ต่อพ่วง เพียง 1.628
3. ค่า Data Map Hits % มีค่าสูงถึง 100% แต่ค่า Data Maps/sec มีค่าต่ำเพียง 2.053 อาจจะทำให้ ประสิทธิภาพ ไม่ดีก็ได้ถ้าอัตราการฮิตสูง แต่อัตราการแมปต่ำ

3. แมสออเดอร์ คีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที ได้ผลดังนี้

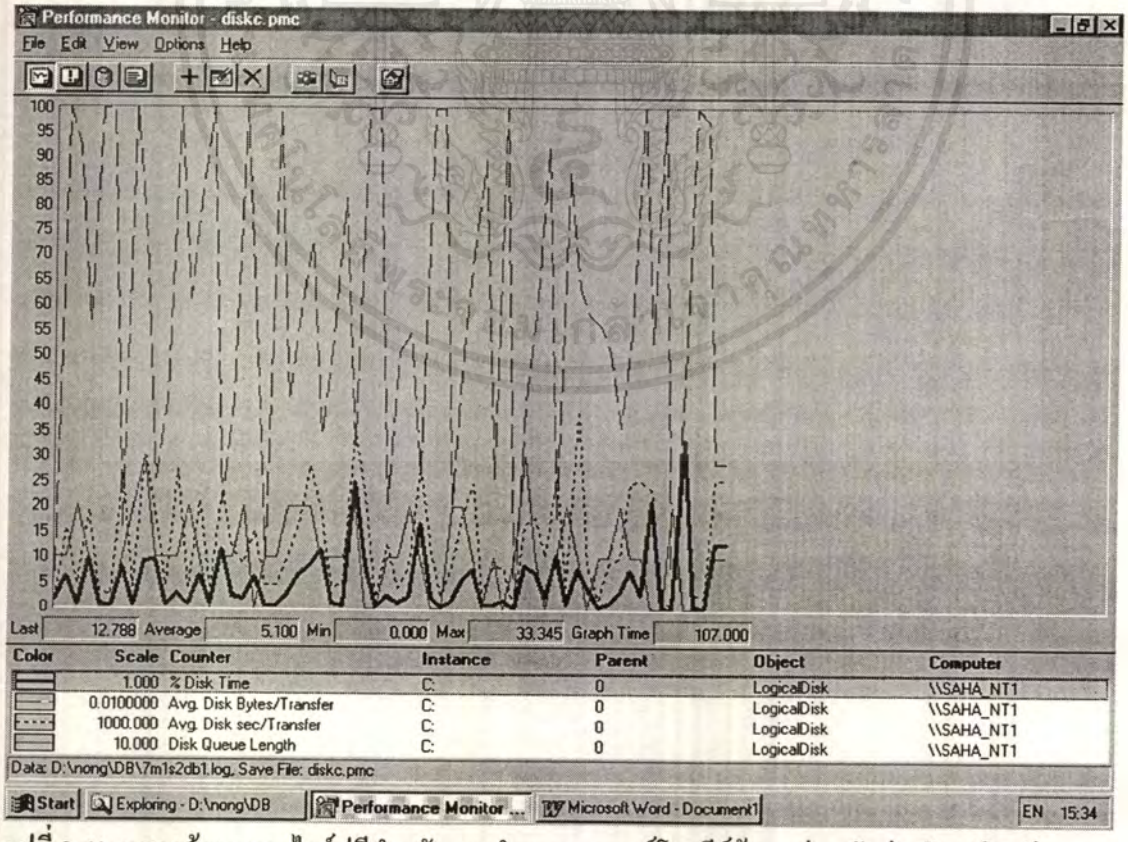


รูปที่ 8-38 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

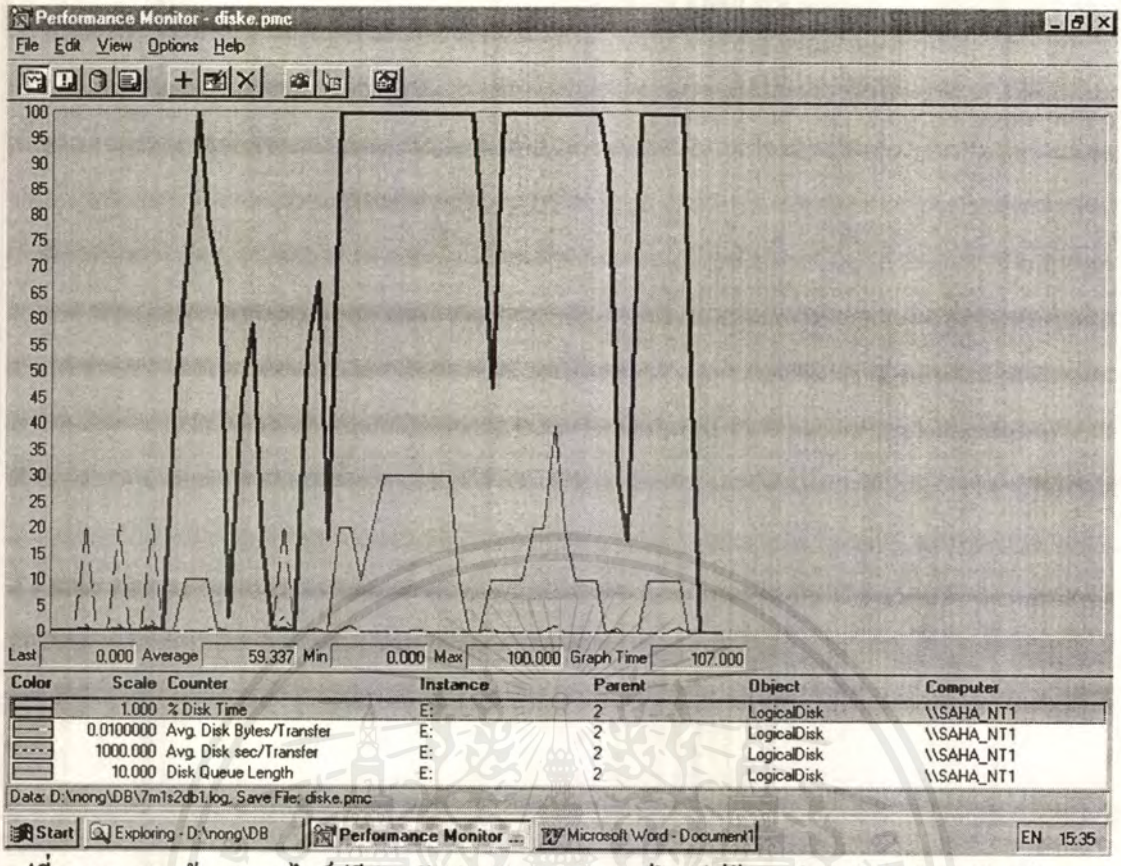


รูปที่ 8-39 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที

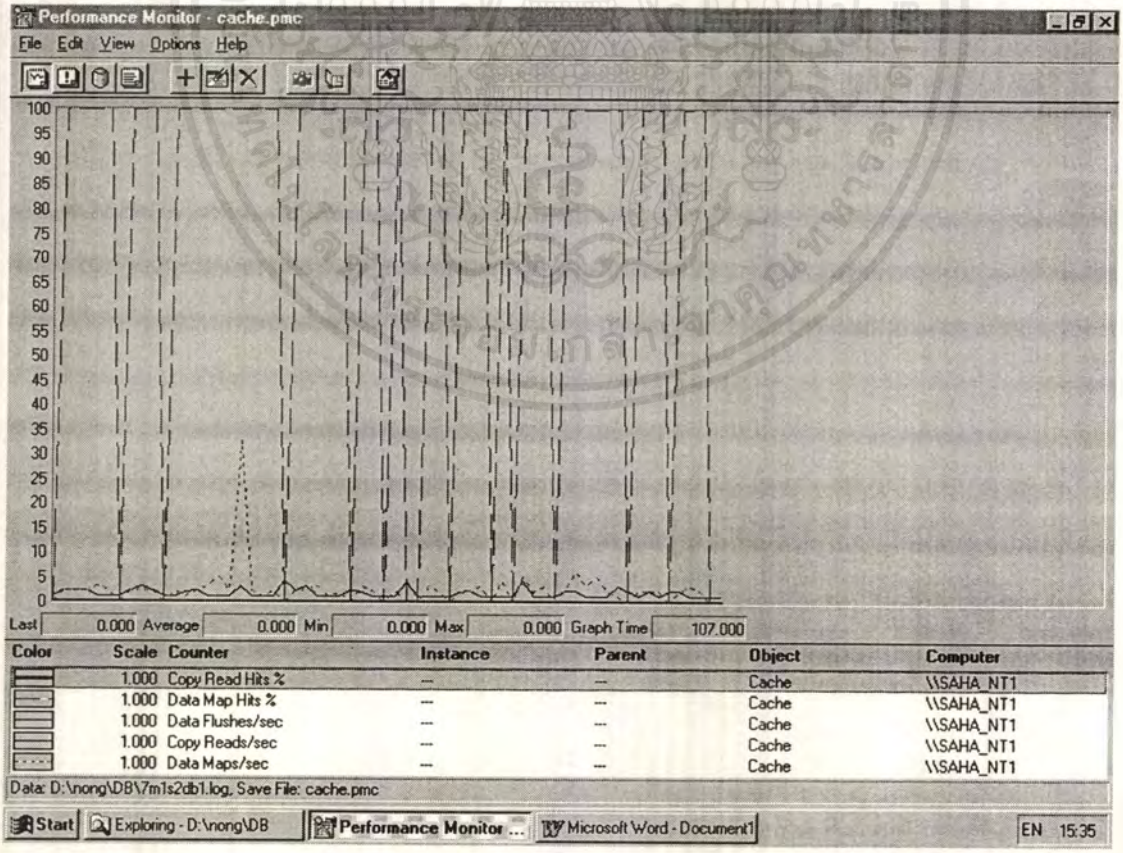


รูปที่ 8-40 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-41 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที



รูปที่ 8-42 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์ข้อมูลส่วนหัวห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. % Processor Time มีค่าเฉลี่ย 46.889% เพื่อดูปริมาณการใช้โพรเซสเซอร์ในกระบวนการทำงาน
2. Processor Queue Length มีค่าที่เวลาหนึ่งเป็น 0 แสดงว่าไม่มีเทรดในคิวของโพรเซสเซอร์ที่กำลังรอการใช้โพรเซสเซอร์อยู่

สรุปไม่เกิด คอขวดที่โพรเซสเซอร์

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเพียง 0 เพจ ซึ่งหมายความว่าไม่มีเพจที่จะต้องเขียนหรืออ่านจาก ดิสก์ และบรรจุเข้าไปใน หน่วยความจำฟิสิคอล ค่านี้จะบ่งบอกว่าในระบบของคุณจะต้องการ หน่วยความจำฟิสิคอลเพิ่มขึ้นหรือไม่ ถ้าค่านี้มีค่าเกิน 5 pages/sec สำหรับ ดิสก์เดียว ก็หมายความว่าอาจจะเกิด คอขวดที่หน่วยความจำ
2. Available Bytes มีค่า 1.14 MB หมายถึงค่าของจำนวน หน่วยความจำเวอร์ชวล ที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่า เทรสโฮล ที่กำหนดคือ 1MB. เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ Windows NT ก็จะค่อย ๆ นำเอา หน่วยความจำจากการวิ่งแอพพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของ หน่วยความจำเวอร์ชวล ไว้ ซึ่งอาจจะทำให้ ประสิทธิภาพของหน่วยความจำ ต่ำลง
3. Commit Limit มีค่าเท่ากับ 329.14 MB ,Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถ คอมมิทได้ โดยไม่ต้องเพิ่มขนาดของ เพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับ ขนาดของ paging file.sys รวมกับ ขนาดของ RAM ที่สามารถ เปลี่ยน ไปยัง ดิสก์ ได้ (หรือจำนวนของ RAM ที่ว่าง ที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 57.6 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ เพจจิ้งไฟล์
5. ดูค่า Page Faults/sec ซึ่งมีค่า 2.609 page faults เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.032 page faults ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำ เพจมาจาก transition page (เพจ ที่อยู่ใน หน่วยความจำ แต่กำลังถูกเขียนลงบน ดิสก์ ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะ เปลี่ยนแปลงแก้ไข ข้อมูล)
6. Pool Nonpaged Bytes มีค่าเท่ากับ 1.87 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของ หน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec ที่อ่านได้มีค่าใกล้เคียง 0 เทียบกับค่า Disk Reads/sec ของ ไดร์ฟต่าง ๆ ซึ่งมีค่าใกล้เคียงกับ ไดร์ฟดี
8. ดูค่า Page Writes/sec ที่อ่านได้มีค่าใกล้เคียง 0 เทียบกับค่า Disk Writes/sec ของ ไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าใกล้เคียงกับ 0 และใกล้เคียงกับ ไดร์ฟดี
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}} * 4096 \text{ bytes/page} = 0 \text{ bytes/read}$ เทียบกับค่า Avg. Disk Bytes/Read ซึ่งมีค่าใกล้เคียงกับ ไดร์ฟดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Page Outputs/sec * 4096 bytes/page = 0 bytes/write เทียบกับค่า
    Page Writes/sec
    Avg. Disk Bytes/Write ซึ่งมีค่าใกล้เคียงกับ ไดร์ฟดี
    สรุปไม่เกิด คอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่ดิสก์

1. % Disk Time

- ไดร์ฟซี: มี % Disk Time เท่ากับ 4.971%
- ไดร์ฟดี: มี % Disk Time เท่ากับ 0%
- ไดร์ฟอี: มี % Disk Time เท่ากับ 100%

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีจำนวนไบต์ในการโอนย้าย 5965.396 ไบต์ ควรจะมีการโอนย้ายข้อมูลได้ประมาณ 20KB
- ไดร์ฟดี: ไม่มีการโอนย้ายของ ไดรว์ดี เพราะการโอนย้ายเป็น 0
- ไดร์ฟอี: มีจำนวนในการโอนย้ายข้อมูล มีค่าเท่ากับ 13KB

3. Avg. Disk sec/transfer

- ไดร์ฟซี: มีเวลาการ โอนย้าย ข้อมูลใน 1 ครั้ง คือ 0.018 วินาที ซึ่งเป็นอัตราการ โอนย้าย ข้อมูลที่เร็ว เพราะมีค่าต่ำกว่า 0.3 วินาที
- ไดร์ฟดี: มีเวลาการ โอนย้าย เท่ากับ 0 วินาที
- ไดร์ฟอี: มีเวลาการ โอนย้าย ข้อมูลเท่ากับ 0.023 วินาที

4. Disk Queue Length

- ไดร์ฟซี: มีค่าเฉลี่ยของคิว 1.016 คิว และมีค่าคิวสูงสุด 3 คิว
- ไดร์ฟดี: ไม่มีคิวใน Disk Queue Length
- ไดร์ฟอี: มีค่าเฉลี่ยของคิว 0.841 คิว และมีค่าคิวสูงสุด 4 คิว

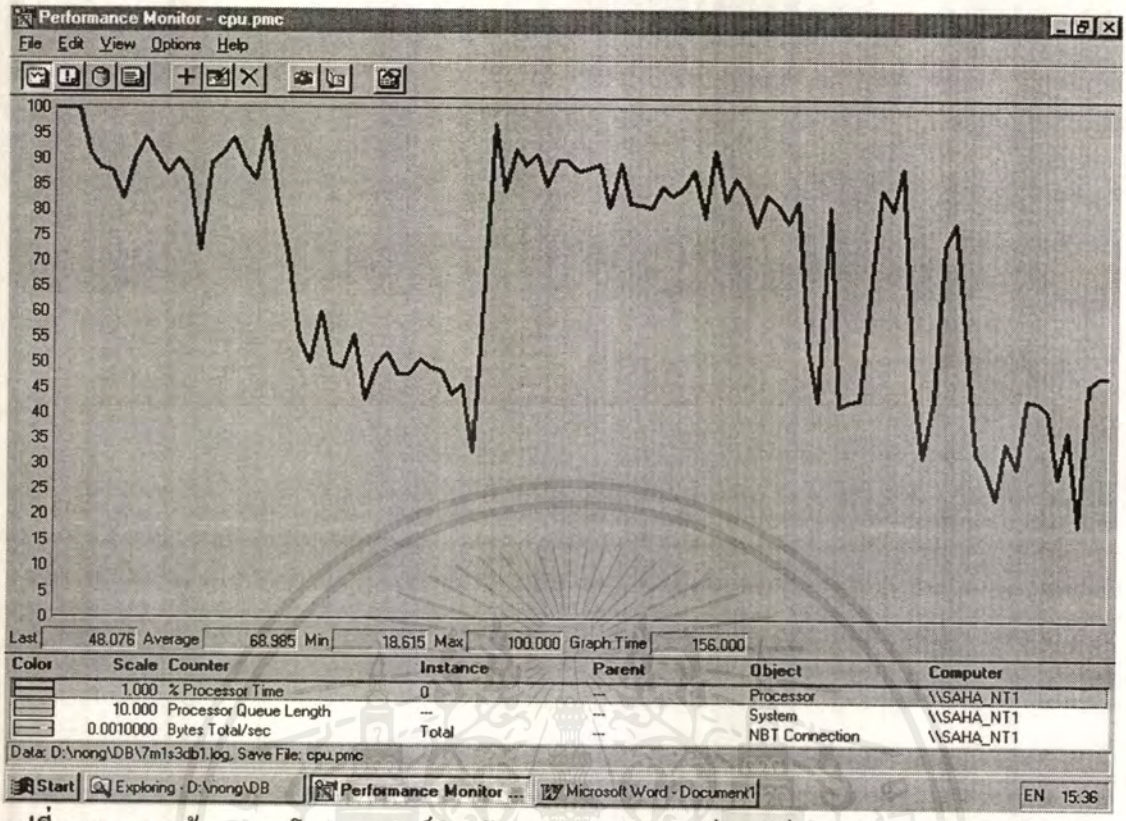
สรุปเกิด คอขวดที่ดิสก์ ขึ้นที่ ไดร์ฟอี

การตรวจจับจุดคอขวดที่แคช

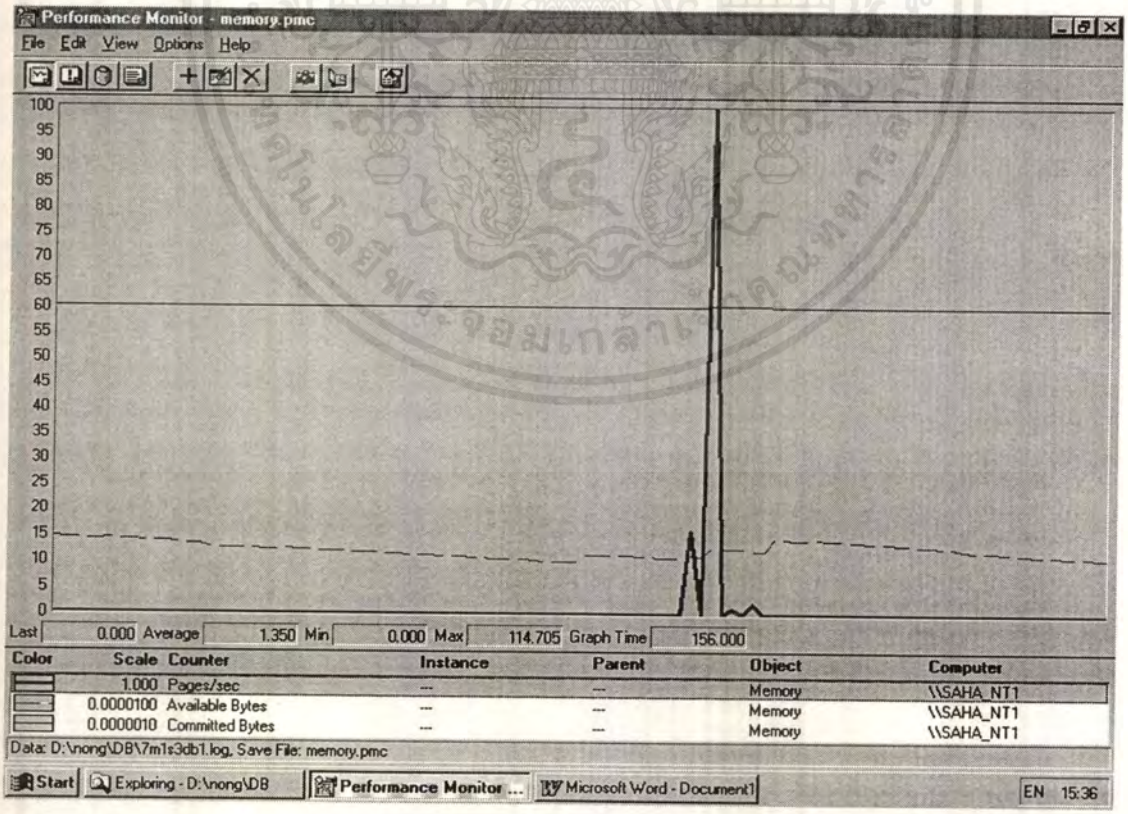
1. Copy Read Hits % ดูเทียบกับค่า Copy Reads/sec ซึ่งค่า Copy Read Hits % มีค่า 0% เมื่อเทียบกับ Copy Reads/sec ซึ่งมีค่าเท่ากับ 0 คือมีค่าเท่ากัน แสดงว่าไม่มีการ ฮิต และ Copy Read ข้อมูลจากแคช
2. Data Flushes/sec มีค่า 1.562 ซึ่งหมายถึงแคชมี การนำข้อมูลออก โดยได้รับการร้องขอจาก ระบบไฟล์ ให้นำข้อมูลออกไปยัง อุปกรณ์ต่อพ่วง เพียง 1.562
3. ค่า Data Map Hits % มีค่าสูงถึง 100% แต่ค่า Data Maps/sec มีค่าต่ำเพียง 2.335 อาจจะทำให้ ประสิทธิภาพ ไม่ดีก็ได้ถ้าอัตราการ ฮิต สูง แต่อัตราการแมป ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แมสออเดอร์ คีย์รายละเอียด ห่างกัน 1 วินาที ได้ผลดังนี้

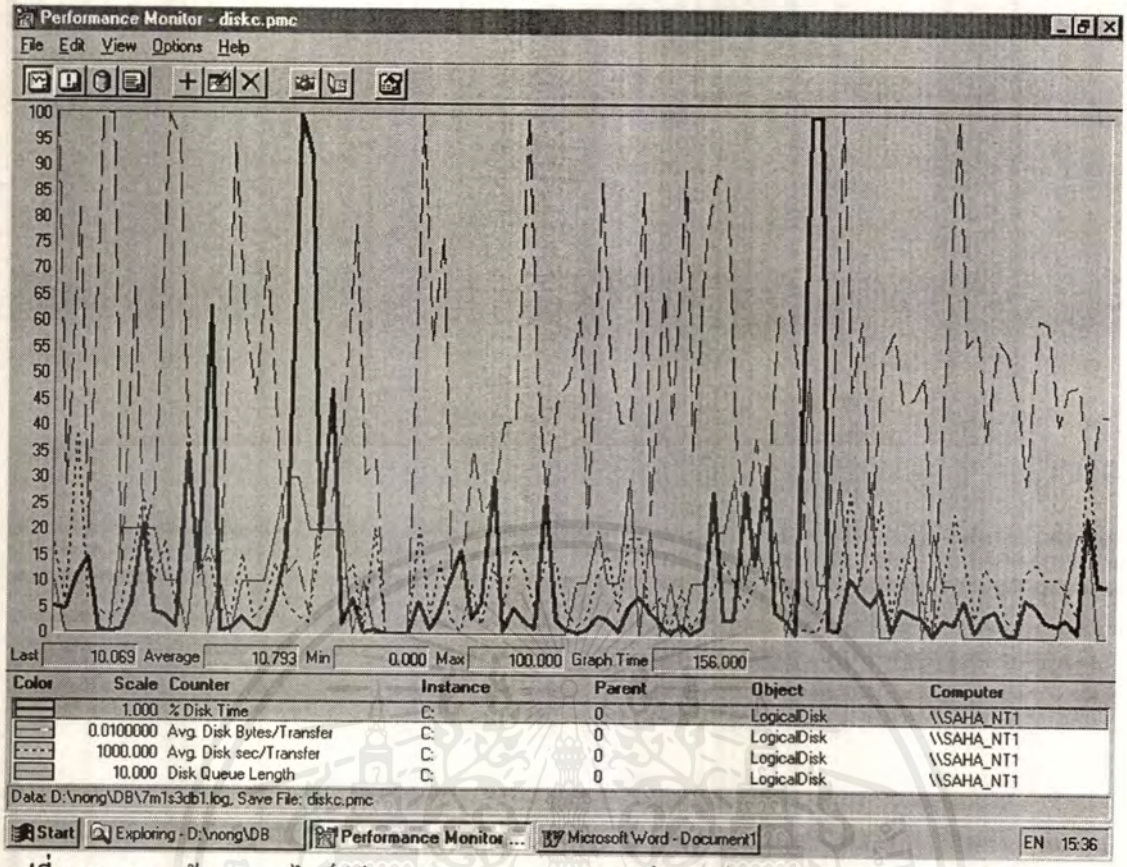


รูปที่ 8-43 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

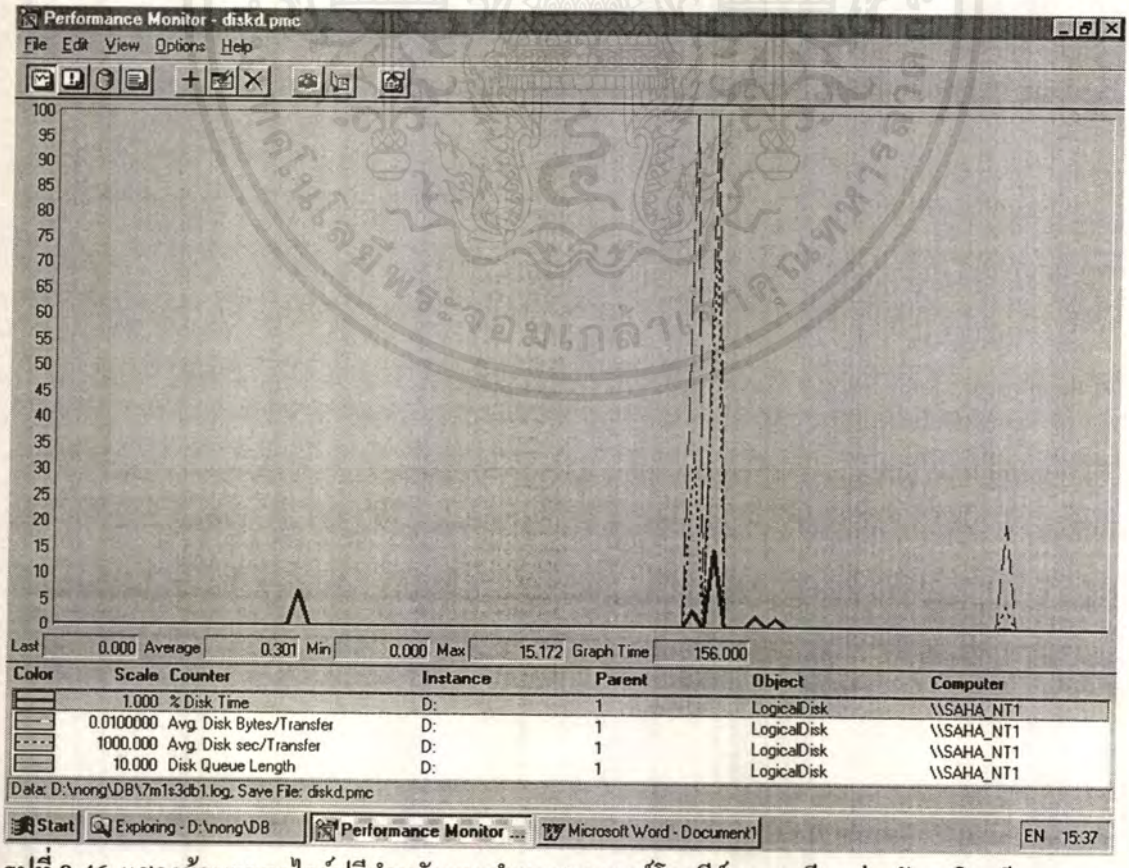


รูปที่ 8-44 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

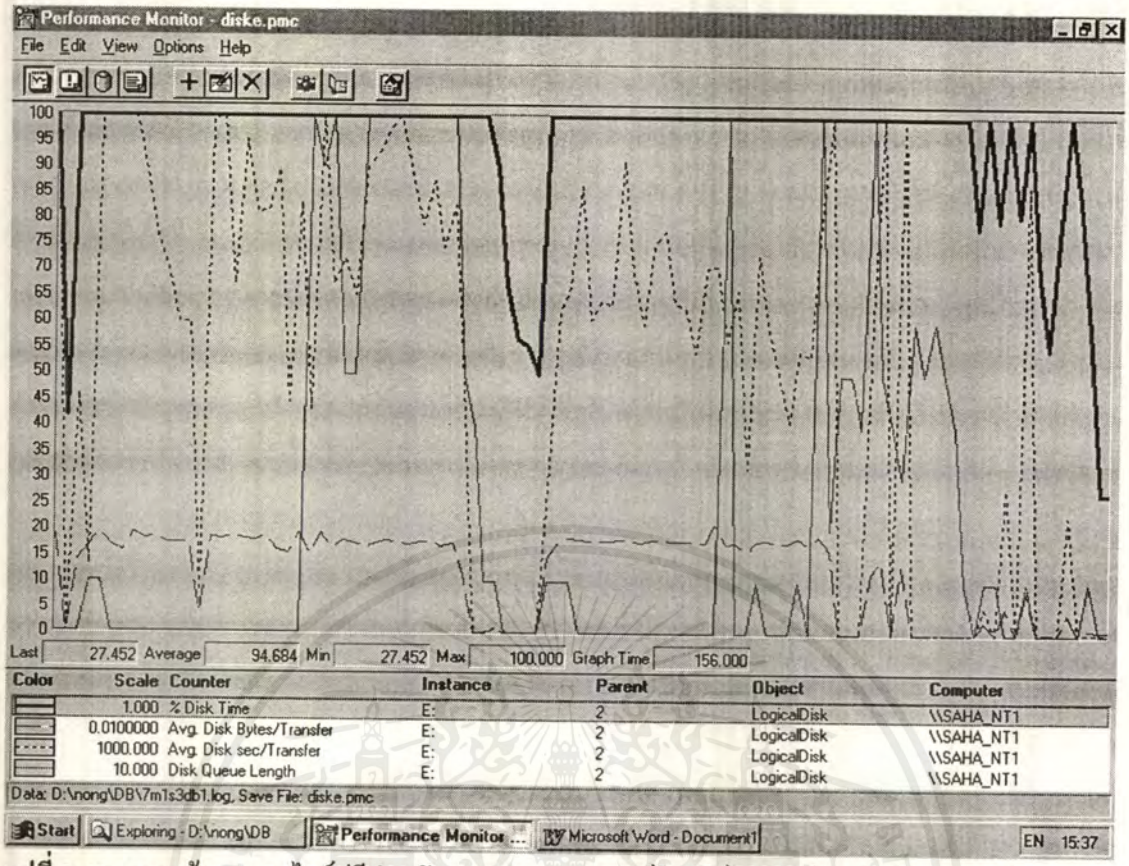


รูปที่ 8-45 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

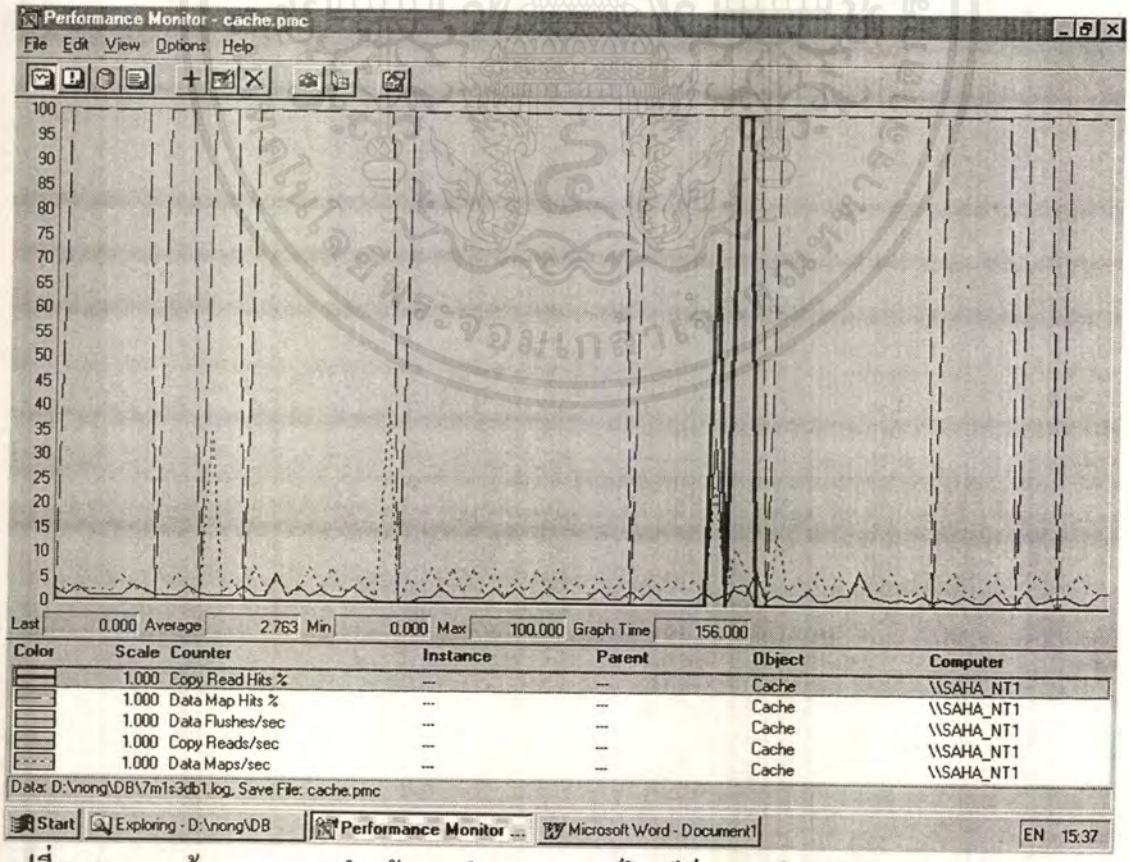


รูปที่ 8-46 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-47 แสดงข้อมูลของไดร์ฟอีสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที



รูปที่ 8-48 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับจุดคอขวดที่โพรเซสเซอร์

1. % Processor Time มีค่าเฉลี่ย 62.251% เพื่อดูปริมาณการใช้โพรเซสเซอร์ในกระบวนการทำงาน
2. Processor Queue Length มีค่าที่เวลาหนึ่งเป็น 0 แสดงว่าไม่มีเทรดในคิวของโพรเซสเซอร์ที่กำลังรอการใช้โพรเซสเซอร์อยู่

สรุปไม่เกิด คอขวดที่โพรเซสเซอร์

การตรวจจับจุดคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเพียง 0.858 เพจ ซึ่งหมายความว่ามีเพจที่จะต้องเขียนหรืออ่านจากดิสก์และบรรจุเข้าไปในหน่วยความจำฟิสิคอล ค่านี้จะบ่งบอกว่าในระบบของคุณจะต้องการหน่วยความจำฟิสิคอลเพิ่มขึ้นหรือไม่ ถ้าค่านี้มีค่าเกิน 5 pages/sec สำหรับ ดิสก์เดียว ก็หมายความว่าอาจจะเกิด คอขวดที่หน่วยความจำ
2. Available Bytes มีค่า 1 MB หมายถึงค่าของจำนวน หน่วยความจำเวอร์ชวล ที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าเท่ากับ เทรสโฮล ที่กำหนดคือ 1MB. เมื่อค่านี้มีค่าต่ำกว่าขอบเขตที่กำหนดไว้ Windows NT ก็จะค่อย ๆ นำเอา หน่วยความจำจากการวิ่งแอพพลิเคชั่นมาเพื่อรักษาค่าต่ำสุดของหน่วยความจำเวอร์ชวล ไว้ ซึ่งอาจจะทำให้ ประสิทธิภาพ ของ หน่วยความจำ ต่ำลง
3. Commit Limit มีค่าเท่ากับ 329.14 MB ,Commit Limit คือ ขนาดของ หน่วยความจำเวอร์ชวล ที่สามารถ คอมมิทได้ โดยไม่ต้องเพิ่มขนาดของ เพจจิ้งไฟล์ ซึ่งค่า Commit Limit นี้มีค่าเท่ากับ ขนาดของ paging file.sys รวมกับ ขนาดของ RAM ที่สามารถ เปลี่ยน ไปยัง ดิสก์ ได้ (หรือจำนวนของ RAM ที่ว่าง ที่เหลือจากการใช้เป็น ระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 57.6 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ เพจจิ้งไฟล์
5. ดูค่า Page Faults/sec ซึ่งมีค่า 3.054 page faults เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.314 page faults ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำ เพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำแต่กำลังถูกเขียนลงบน ดิสก์ ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะ เปลี่ยนแปลงแก้ไข ข้อมูล)
6. Pool Nonpaged Bytes มีค่าเท่ากับ 1.87 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของ หน่วยความจำฟิสิคอล ลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec ที่อ่านได้มีค่าใกล้เคียง 0 เทียบกับค่า Disk Reads/sec ของ ไดร์ฟต่าง ๆ ซึ่งมีค่าใกล้เคียงกับ ไดร์ฟดี
8. ดูค่า Page Writes/sec ที่อ่านได้มีค่าใกล้เคียง 0 เทียบกับค่า Disk Writes/sec ของ ไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าใกล้เคียงกับ 0 และใกล้เคียงกับ ไดร์ฟดี
9. Page Inputs/sec * 4096 bytes/page = 4733.9 bytes/read เทียบกับค่า
   Page Reads/sec
   Avg. Disk Bytes/Read ซึ่งมีค่าไม่ใกล้เคียงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Page Outputs/sec * 4096 bytes/page = 65262.93 bytes/write เทียบกับค่า
Page Writes/sec
Avg. Disk Bytes/Write ซึ่งมีค่าไม่ใกล้เคียงกันเลย

สรุปไม่เกิด คอขวดที่หน่วยความจำ

การตรวจจับจุดคอขวดที่ดิสก์

1. % Disk Time

- ไดร์ฟซี: มี % Disk Time เท่ากับ 8.743%
- ไดร์ฟดี: มี % Disk Time เท่ากับ 0.192%
- ไดร์ฟอี: มี % Disk Time เท่ากับ 100%

2. Avg. Disk Bytes/transfer

- ไดร์ฟซี: มีจำนวน ไบต์ในการ โอนย้าย 4131.556 ไบต์ควรจะมีการโอนย้ายข้อมูลได้ประมาณ 20KB
- ไดร์ฟดี: มีการโอนย้ายเป็น 23210.666 ไบต์ ซึ่งนับว่ามีการโอนย้ายที่สูงมาก
- ไดร์ฟอี: มีจำนวนในการโอนย้ายข้อมูล มีค่าเท่ากับ 3445.191

3. Avg. Disk sec/transfer

- ไดร์ฟซี: มีเวลาการโอนย้ายข้อมูลใน 1 ครั้ง คือ 0.024 วินาที ซึ่งเป็นอัตราการโอนย้ายข้อมูลที่เร็ว เพราะมีค่าต่ำกว่า 0.3 วินาที
- ไดร์ฟดี: มีเวลาการ โอนย้าย เท่ากับ 0.033 วินาที
- ไดร์ฟอี: มีเวลาการ โอนย้าย ข้อมูลเท่ากับ 0.093 วินาที

4. Disk Queue Length

- ไดร์ฟซี: มีค่าเฉลี่ยของคิว 0.899 คิว และมีค่าคิวสูงสุด 5 คิว
- ไดร์ฟดี: ไม่มีคิวใน Disk Queue Length
- ไดร์ฟอี: มีค่าเฉลี่ยของคิว 7.717 คิว และมีค่าคิวสูงสุด 147 คิว

สรุปเกิด คอขวดที่ดิสก์ ขึ้นที่ ไดร์ฟอี

การตรวจจับจุดคอขวดที่แคช

1. Copy Read Hits % ดูเทียบกับค่า Copy Reads/sec ซึ่งค่า Copy Read Hits % มีค่า 78.571% เมื่อเทียบกับ Copy Reads/sec ซึ่งมีค่าเท่ากับ 0.269 ซึ่งมีอัตราการของข้อมูลในแคชที่ค่อนข้างสูง แต่อัตราการอ่านค่อนข้างต่ำ
2. Data Flushes/sec มีค่า 1.236 ซึ่งหมายถึงแคชมี การนำข้อมูลออก โดยได้รับการร้องขอจาก ระบบไฟล์ ให้นำข้อมูลออกไปยัง อุปกรณ์ต่อพ่วง เพียง 1.236
3. ค่า Data Map Hits % มีค่าสูงถึง 100% แต่ค่า Data Maps/sec มีค่าต่ำเพียง 2.990 อาจจะทำให้ ประสิทธิภาพ ไม่ดีก็ได้ถ้าอัตราการ ฮิต สูง แต่อัตราการแมปต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองนี้พบว่าส่วนใหญ่จะเกิดจุดคอขวดที่หน่วยความจำและดิสก์ของทั้งไฟล์เซิร์ฟเวอร์และดาต้าเบสเซิร์ฟเวอร์ จึงทำการปรับปรุงประสิทธิภาพของเครื่องเซิร์ฟเวอร์ทั้ง 2 เครื่อง ดังนี้

เครื่องดาต้าเบสเซิร์ฟเวอร์ (ชื่อเครื่อง SAHA_NT1)

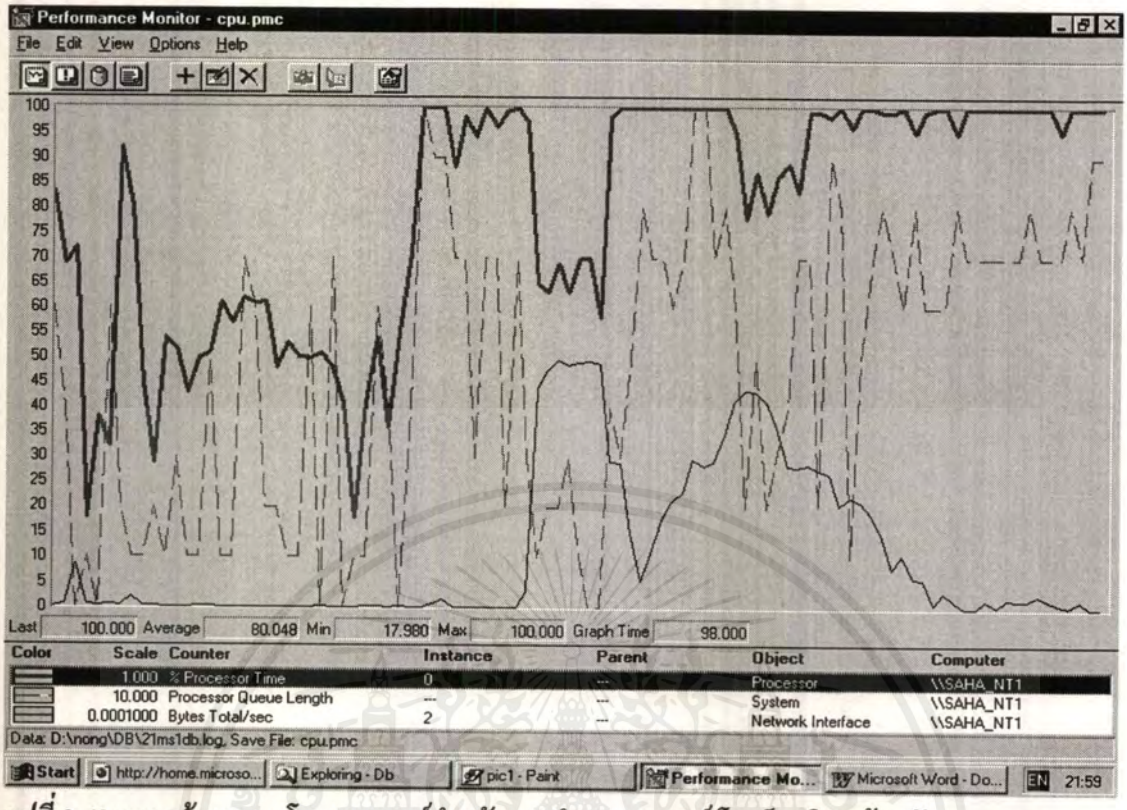
1. เพิ่มหน่วยความจำ จากเดิมมี 64 MB. เพิ่มเป็น 96 MB.
2. เพิ่มเอสจีเอ (SGA:System Global Area) ซึ่งเป็นแคชอย่างหนึ่งของออราเคิล (Oracle) จากเดิมมี 25 MB. เพิ่มเป็น 35 MB. เนื่องจากว่าโปรแกรมที่ใช้ในการทดลองนั้นส่วนใหญ่จะเป็นการดึงข้อมูลจากฐานข้อมูลโดยใช้ออราเคิล จึงจำเป็นที่จะต้องเพิ่มพื้นที่แคชให้กับโปรแกรม โดยแคชนี้จะเป็นพื้นที่จริงจากหน่วยความจำหลักของวินโดว์เอ็นที คือ ในที่นี้มีหน่วยความจำหลัก 96 MB. จะถูกใช้ไปเป็นแคชของออราเคิลจำนวน 35 MB.
3. ทำการแบ่งพาร์ติชั่นดิสก์ใหม่ จากเดิมมีฟิสิคอลดิสก์ 3 ดิสก์ แบ่งเป็น 3 ลอจิคอลดิสก์ คือ C:, D:, E: โดยแบ่งใหม่เป็นลอจิคอลดิสก์ C:, D:, E:, F:
4. ทำการย้ายฐานข้อมูล (load balancing) ครึ่งหนึ่งไปไว้ที่เครื่องไฟล์เซิร์ฟเวอร์

เครื่องไฟล์เซิร์ฟเวอร์ (ชื่อเครื่อง SAHA_NT2)

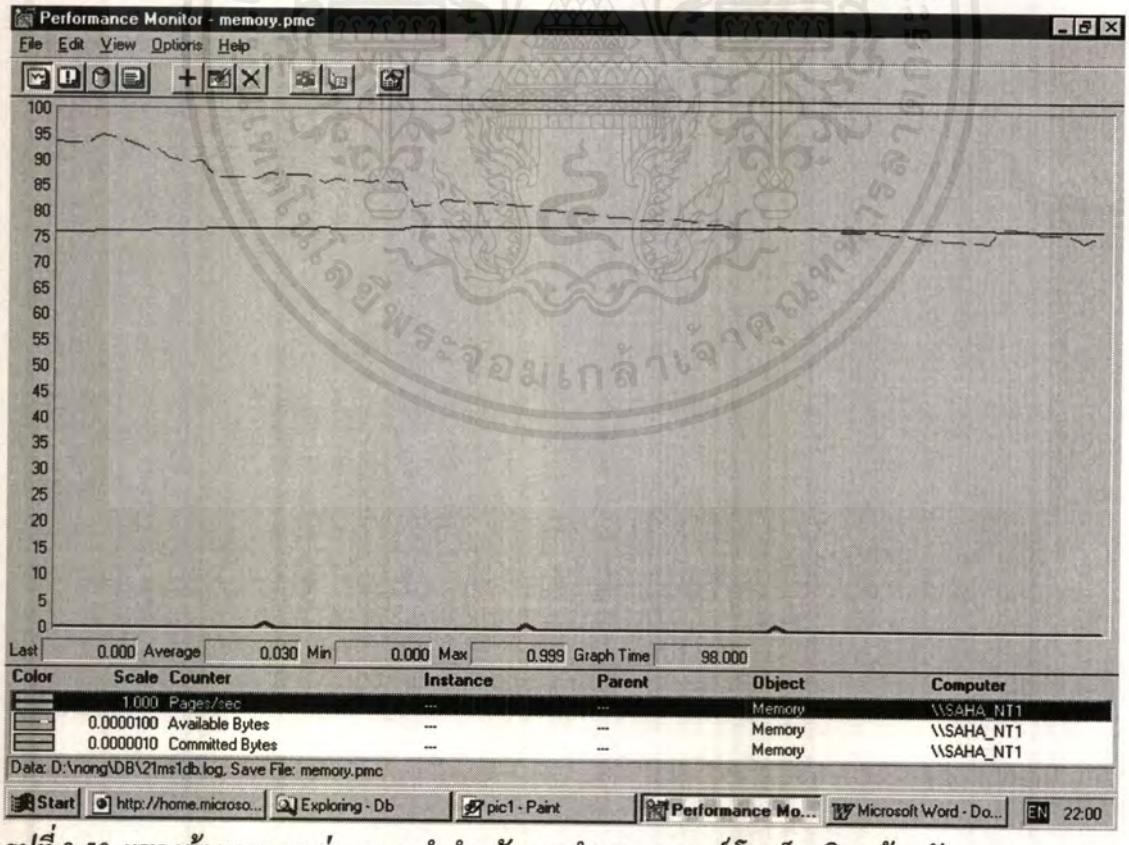
1. เพิ่มหน่วยความจำ จากเดิมมี 32 MB. เพิ่มเป็น 96 MB.
2. เนื่องจากเครื่อง SAHA_NT2 นี้ได้รับการแบ่งฐานข้อมูลมาจากเครื่อง SAHA_NT1 จึงต้องมีการติดตั้งโปรแกรมออราเคิล และให้มีเอสจีเอหรือแคชของออราเคิลจำนวน 35 MB. เช่นเดียวกับเครื่อง SAHA_NT1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แมสออเดอร์ ล็อกอินพร้อมกัน

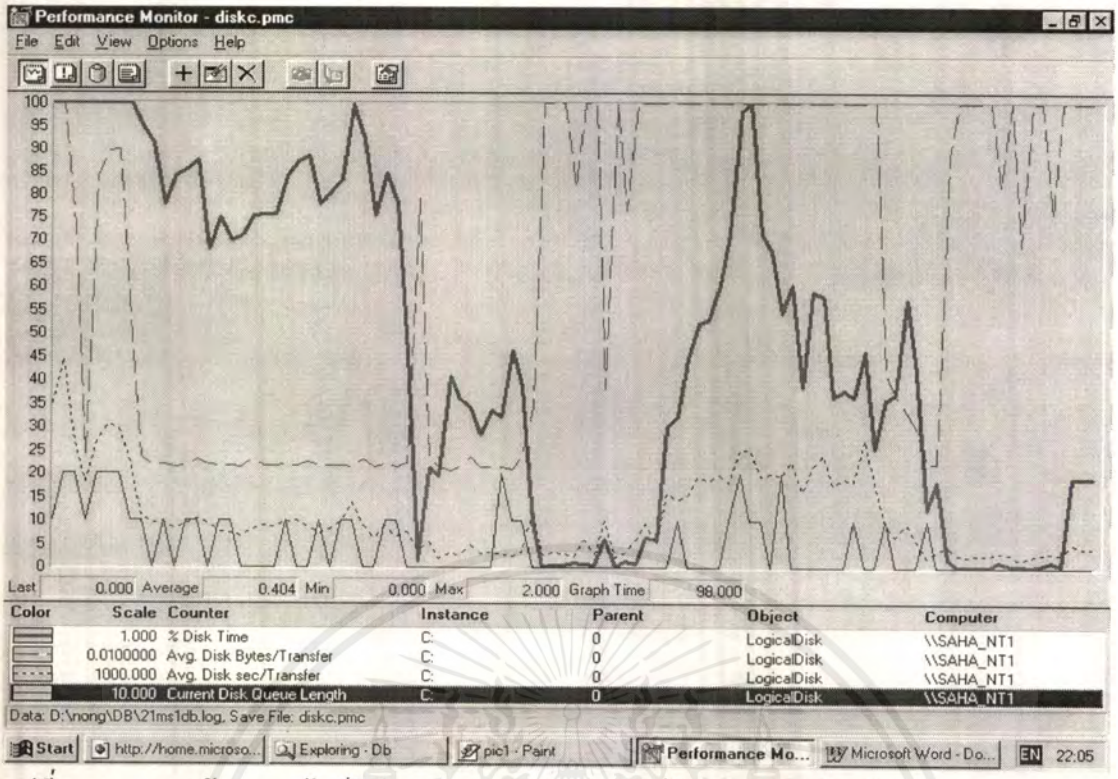


รูปที่ 8-49 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน

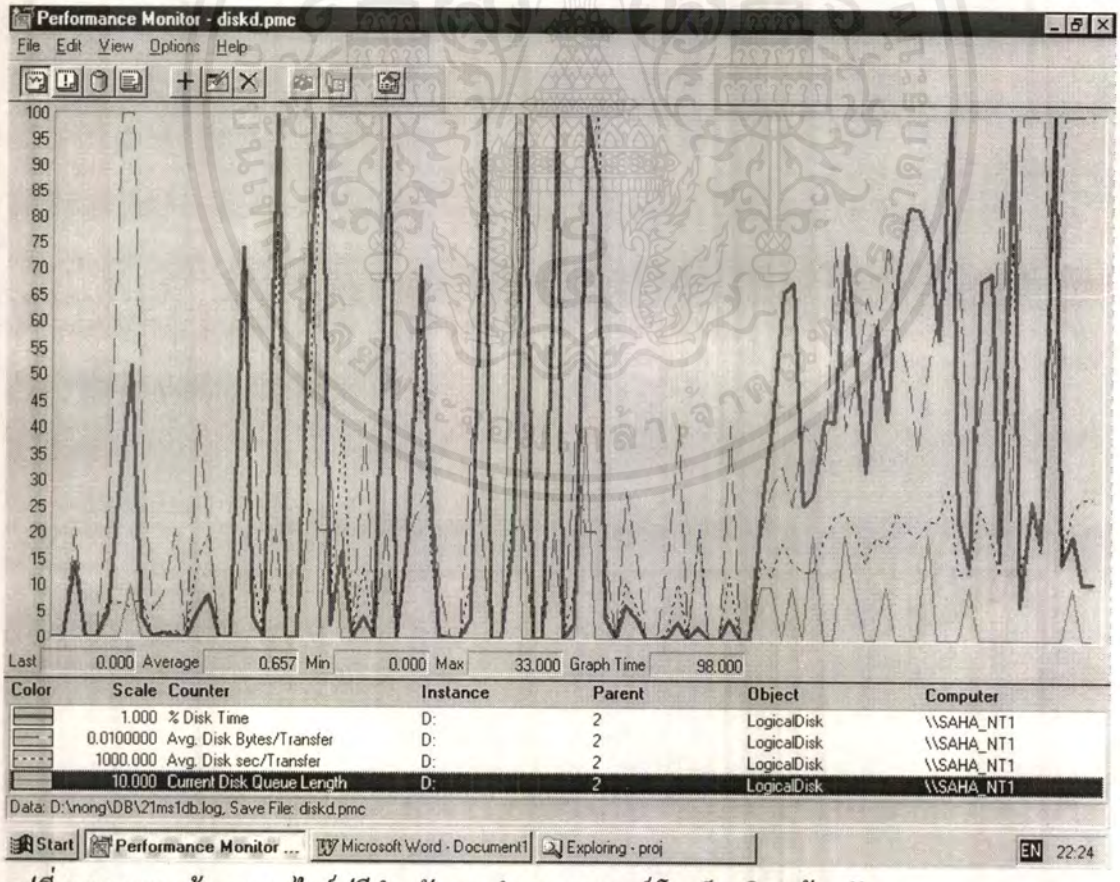


รูปที่ 8-50 แสดงข้อมูลของหน่วยความจำสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

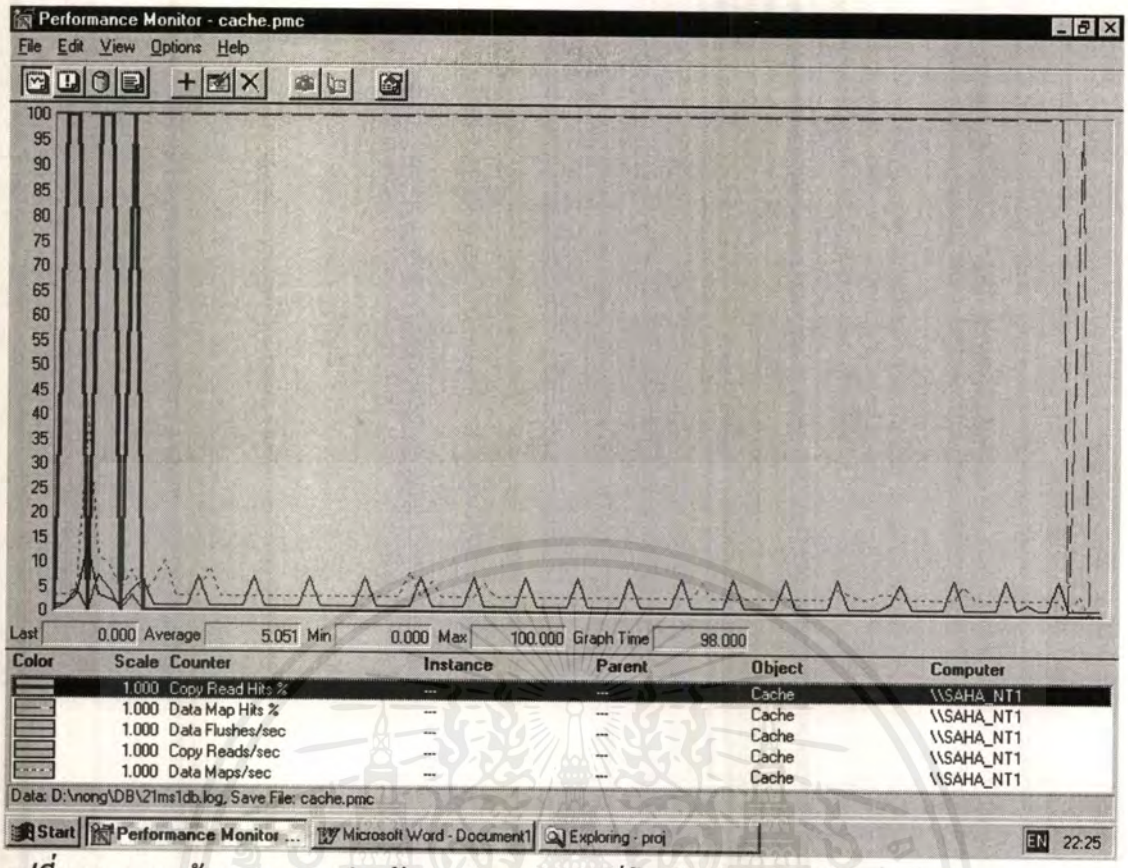


รูปที่ 8-51 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน



รูปที่ 8-52 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-53 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน

การตรวจจับคอขวดที่โพรเซสเซอร์

1. %Processor Time มีค่าเฉลี่ยที่ 80.048 % แสดงว่ามีการใช้โพรเซสเซอร์ในกระบวนการทำงานมาก
2. เมื่อดูที่ Processor Queue Length พบว่ามีค่าค่อนข้างสูงตลอดการล็อค โดยมีค่าเฉลี่ยที่ 4.778 และมีค่าสูงที่สุด คือ 10 จึงสรุปได้ว่าเกิดจุดคอขวดที่โพรเซสเซอร์แน่นอน

การตรวจจับคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเพียง 0.03 เพจ แสดงว่ามีอัตราการเพจจิ้งน้อยมาก
2. Available Bytes มีค่าประมาณ 8 MB หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่าเทรสโฮลที่กำหนดคือ 1MB
3. Commit Limit มีค่าประมาณ 400 MB ,Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิตได้ โดยไม่ต้องเพิ่มขนาดของ paging file ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ Paging file.sys รวมกับ ขนาดของแรมที่สามารถเปลี่ยนกลับไปยังดิสก์ได้ (หรือจำนวนของแรมที่ว่างที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 77 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ Paging file
5. ดูค่า Page Faults/sec ซึ่งมีค่า 54.289 เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.224 ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ แต่กำลังถูกเขียนลงบนดิสก์ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)

6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอลลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของไดร์ฟต่าง ๆ ซึ่งมีค่าไม่ใกล้เคียงกันเลย
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้ไม่ใกล้เคียงกัน
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}}$ * 4096 bytes/page = 4096 bytes/read เทียบกับค่า

   Avg. Disk Bytes/Read ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
10. $\frac{\text{Page Outputs/sec}}{\text{Page Writes/sec}}$ * 4096 bytes/page = 0 bytes/write เทียบกับค่า

    Avg. Disk Bytes/Write ซึ่งใกล้เคียงกับไดร์ฟดี

สรุปไม่เกิดคอขวดที่หน่วยความจำ

การตรวจจับคอขวดที่ดิสก์

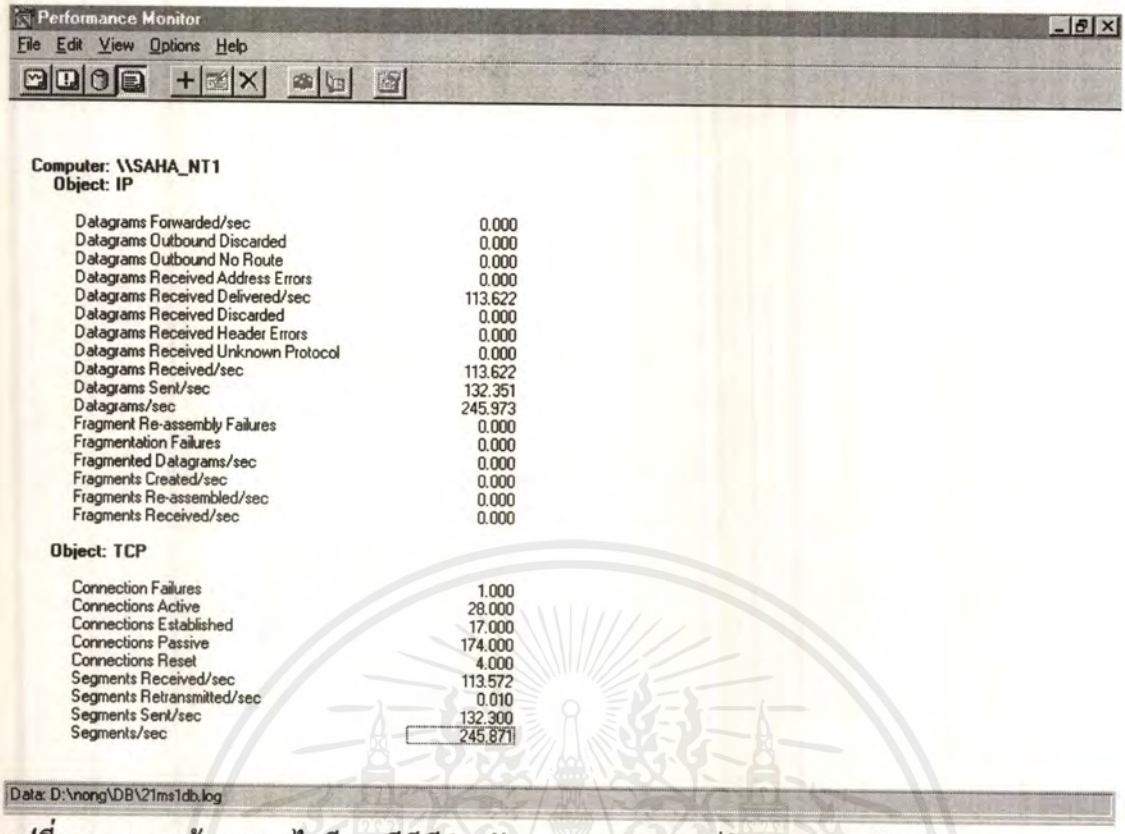
1. ที่ไดร์ฟซี เกิดคอขวดเฉพาะตรงช่วงแรกของการล็อคเท่านั้น โดยช่วงแรกนี้มี %Disk Time สูงถึง 85 % ส่วนคิวของดิสก์นั้นมีค่าเฉลี่ยเป็น 0.424 โดยจะเกิดคิวมากในช่วงที่เกิดคอขวดที่ดิสก์
2. ที่ไดร์ฟดีนั้นไม่เกิดคอขวด เพราะเปอร์เซ็นต์การใช้ดิสก์ไม่มากนัก ส่วนคิวนั้นมีค่าเฉลี่ยเพียง 0.675
3. ที่ไดร์อีและเอฟนั้นไม่มีการส่งข้อมูลจากดิสก์หรือไปยังดิสก์เลย

สรุป ไม่เกิดคอขวดที่ดิสก์

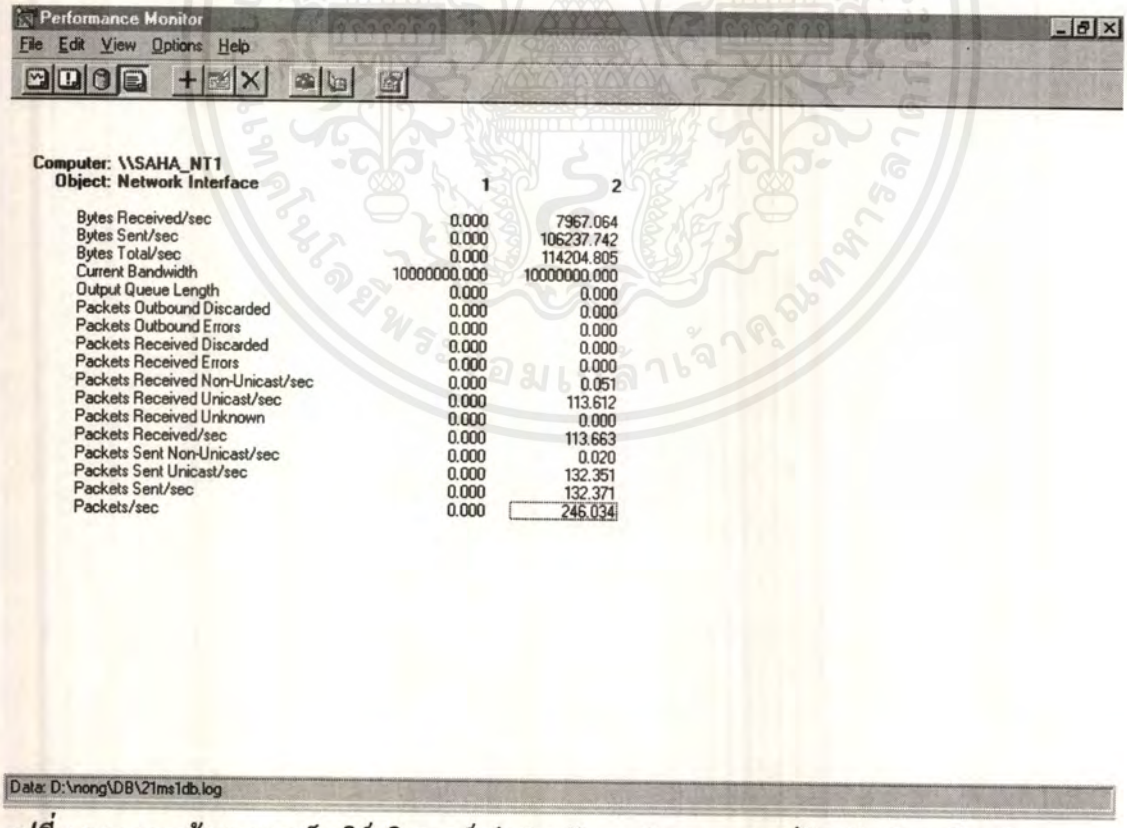
การตรวจจับคอขวดที่แคช

1. Copy Read Hits % และ Copy Reads/sec มีค่าค่อนข้างต่ำ
2. Data Flushes/sec มีค่า 2.3 ซึ่งหมายถึงแคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วงเพียง 2.3
3. ค่า Data Map Hits % มีค่าสูงถึง 97.98 % แต่ค่า Data Maps/sec มีค่าต่ำเพียง 3.914 อาจจะทำให้ประสิทธิภาพไม่ดีก็ได้ถ้าอัตราการฮิตสูง แต่อัตราการแมปต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-54 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน



รูปที่ 8-55 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำแมสออเดอร์ โดยล็อกอินพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับคอขวดที่เครือข่าย

1. ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface)

เนื่องจากเครือข่ายที่ใช้ทดลองนี้เป็นเครือข่ายอีเทอร์เน็ต (Ethernet) ดังนั้นจึงมีอัตราการส่งเป็น 10 เมกะบิตต่อวินาที (10 Mbps.) และสามารถคิดเป็น %Utilization ของการส่งข้อมูลเป็นไบต์ จากค่าของ Bytes Total/sec ได้ดังนี้

114204.805 Bytes Total/sec = 114204.805 * 8 Bits/sec

เพราะฉะนั้น %Byte Utilization = $\frac{\text{จำนวน Bits/sec}}{\text{10Mbps.}} * 100$

$$= \frac{114204.805 * 8}{10 * 10^6} * 100 = 9.136\%$$

และสามารถหาจำนวนแพ็คเก็ตต่อการส่งใน 1 วินาทีของอีเทอร์เน็ตได้ดังนี้

จำนวนไบต์ต่อแพ็คเก็ต = 512 Bytes/packet = 512 * 8 Bits/packet

ดังนั้น $\text{Packets/sec} = \frac{\text{10 Mbps}}{\text{512 * 8 Bits/packet}} = \frac{10 * 10^6}{512 * 8}$ Packets/sec

เพราะฉะนั้น %Packet Utilization $= \frac{246.034 * 512 * 8}{10 * 10^6} * 100$

$= 10.078\ \%$

เมื่อพิจารณาจากค่า %Utilization ทั้ง 2 ค่าแล้วจะเห็นว่ามีค่าค่อนข้างต่ำ แสดงว่าการรับส่งข้อมูลมีค่าน้อยมากเมื่อเทียบกับค่าที่มันสามารถรองรับได้

2. ออบเจ็กต์ : ไอพี (Object : IP)

เมื่อดูจากค่า Datagrams Received Delivered/sec พบว่ามีค่าเท่ากับ Datagrams Received/sec แสดงว่ามีการส่งดาต้าแกรมที่ได้รับมา ออกไปสำเร็จทุกดาต้าแกรม

3.ออบเจ็กต์ : ทีซีพี (Object : TCP)

พิจารณาเปอร์เซ็นต์การติดต่อ (connection) ที่ไม่สำเร็จ โดยคิดเทียบกับจำนวนการติดต่อทั้งหมด (คือ ทั้งด้านรับและด้านส่ง) และจำนวนการติดต่อทั้งหมดคิดได้จาก

ค่า Connections Active (ด้านส่งข้อมูล) + ค่า Connections Passive (ด้านรับข้อมูล)

เพราะฉะนั้นเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จ $= \frac{\text{Connection Failures}}{\text{Connections Active + Connection Passive}} * 100$

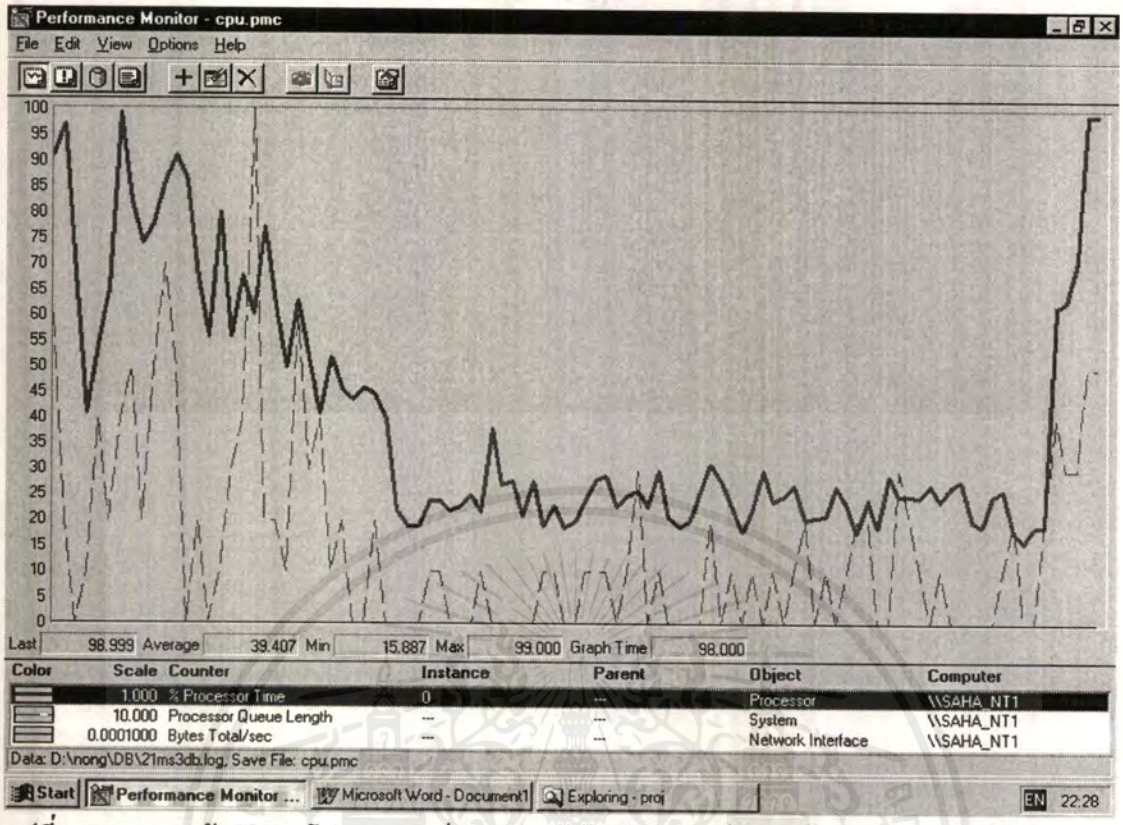
$$= \frac{1}{28+174} * 100 = 0.495\%$$

จะเห็นว่าเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จนั้นน้อยมาก แสดงว่าการรับส่งข้อมูลของชั้นทีซีพีนี้มีความผิดพลาดน้อยมาก จึงไม่เกิดปัญหาขึ้น

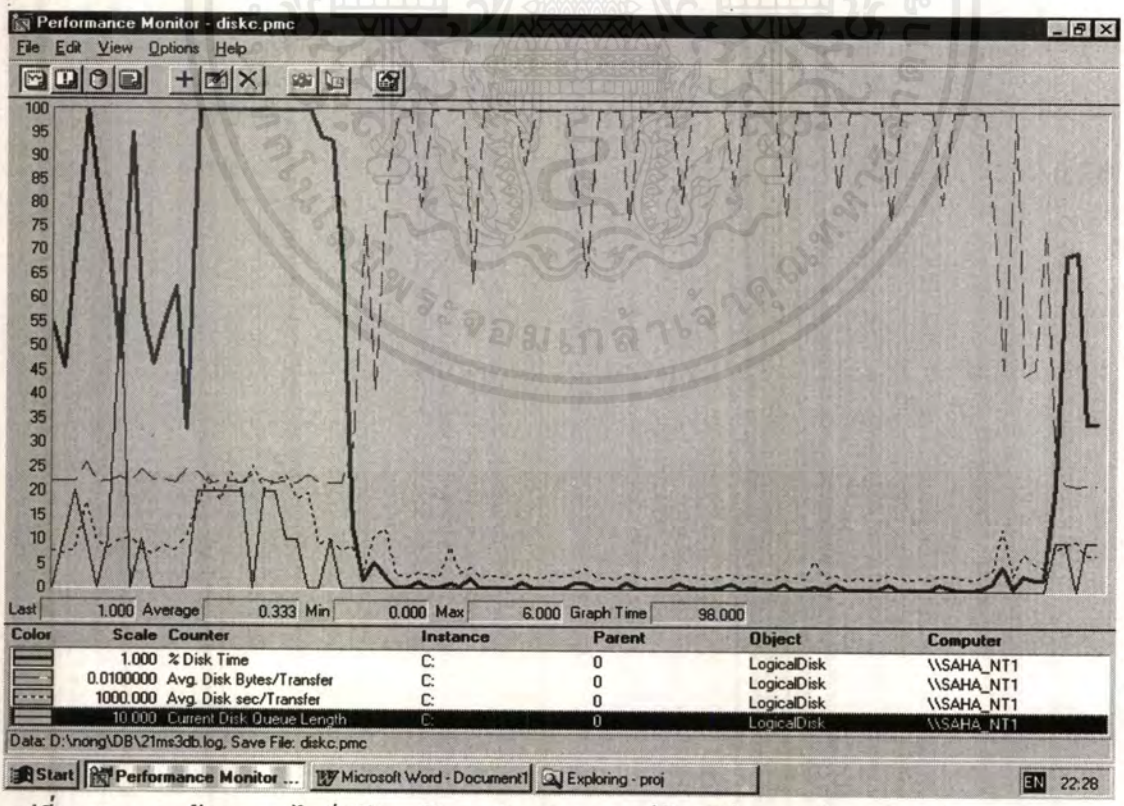
สรุป ไม่เกิดคอขวดที่เครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แมสออเดอร์ คีย์รายละเอียดพร้อมกัน

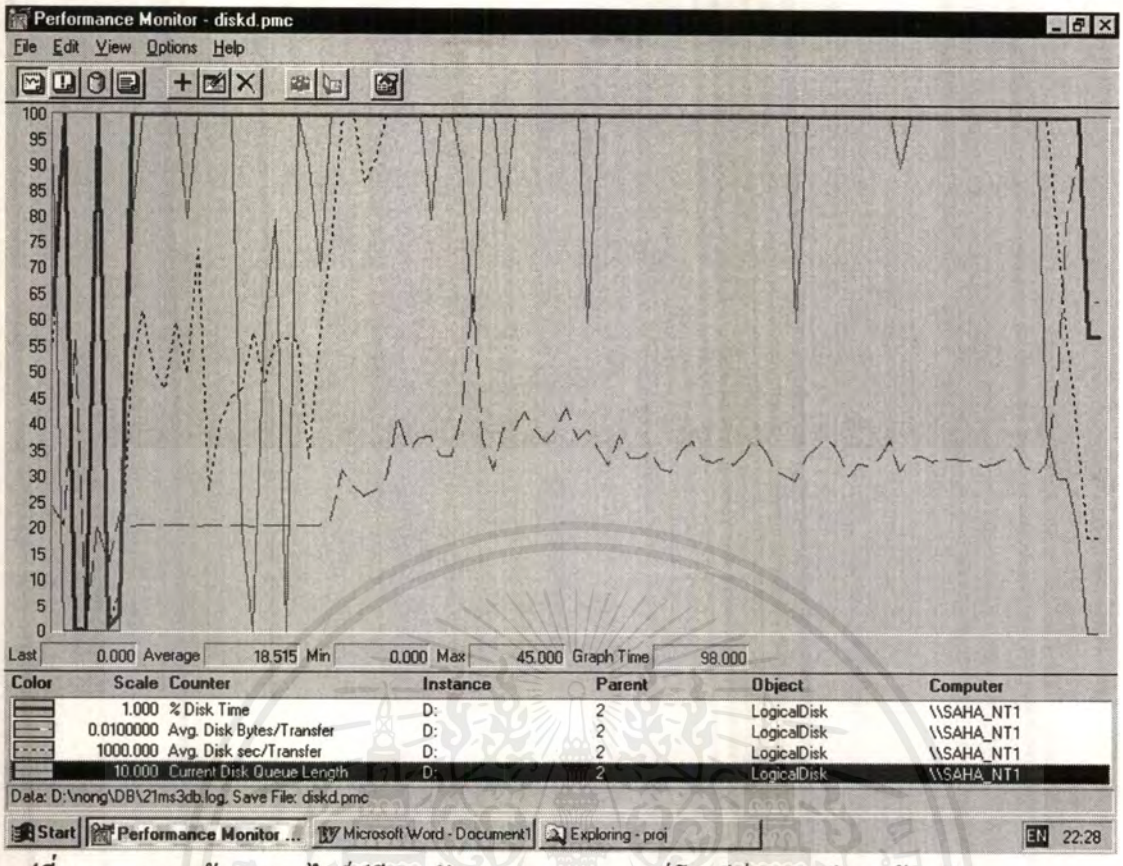


รูปที่ 8-56 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน

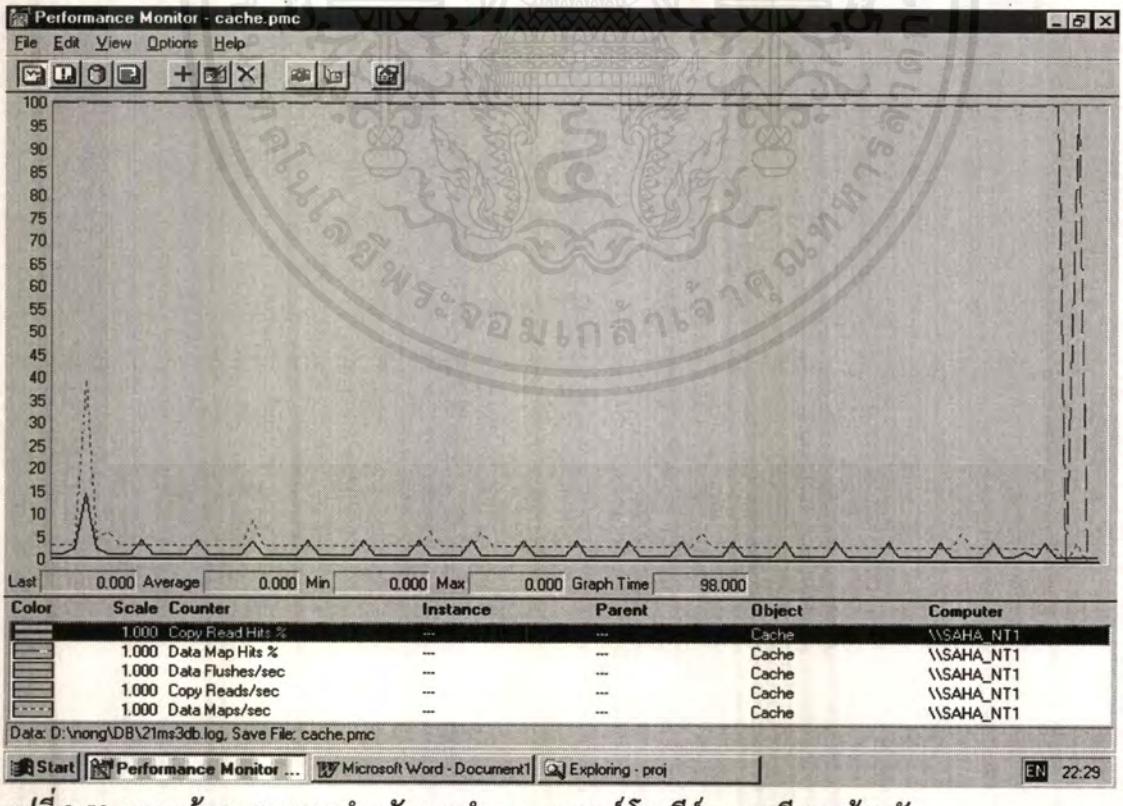


รูปที่ 8-57 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-58 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน



รูปที่ 8-59 แสดงข้อมูลของแคชสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับคอขวดที่โพรเซสเซอร์

1. %Processor Time มีค่าเฉลี่ยที่ 39.417 % แสดงว่ามีการใช้โพรเซสเซอร์ในกระบวนการทำงานไม่มาก
2. เมื่อดูที่ Processor Queue Length พบว่ามีค่าไม่มากนัก โดยมีค่าเฉลี่ยเพียง 1.485 และมีค่าสูงที่สุด คือ 11 สรุปได้ว่าไม่เกิดคอขวดที่โพรเซสเซอร์

การตรวจจับคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเพียง 0.031 เพจ แสดงว่ามีอัตราการเพจจิ้งน้อยมาก
2. Available Bytes มีค่าประมาณ 7 MB หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่าเทรสโฮลที่กำหนดคือ 1MB
3. Commit Limit มีค่าประมาณ 400 MB ,Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิตได้ โดยไม่ต้องเพิ่มขนาดของ paging file ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ Paging file.sys รวมกับ ขนาดของแรมที่สามารถเปลี่ยนกลับไปยังดิสก์ได้ (หรือจำนวนของแรมที่ว่างที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 78 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ Paging file
5. ดูค่า Page Faults/sec ซึ่งมีค่า 21.046 เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.101 ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำ แต่กำลังถูกเขียนลงบนดิสก์ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอลลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของไดร์ฟต่าง ๆ ซึ่งมีค่าไม่ใกล้เคียงกันเลย
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าเป็นศูนย์ ซึ่งเท่ากับไดร์ฟอีและเอฟ
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}}$ * 4096 bytes/page = 4096 bytes/read เทียบกับค่า Avg. Disk Bytes/Read ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
10. $\frac{\text{Page Outputs/sec}}{\text{Page Writes/sec}}$ * 4096 bytes/page = 0 bytes/write เทียบกับค่า Avg. Disk Bytes/Write ซึ่งเท่ากับไดร์ฟอีและเอฟ

สรุปไม่เกิดคอขวดที่หน่วยความจำ

การตรวจจับคอขวดที่ดิสก์

1. ที่ไดร์ฟซี มีการใช้ดิสก์ไดร์ฟซีมากตรงช่วงแรก คือมี %Disk Time 73.941% แต่ก็ยังไม่ถึงกับเป็นคอขวดของระบบ และเมื่อดูเปอร์เซ็นต์โดยเฉลี่ยแล้ว มีเพียง 24.908% เท่านั้น ส่วนคิวของดิสก์นั้นมีค่าเฉลี่ยเป็น 0.323 เท่านั้น จึงทำให้สรุปได้ว่าดิสก์ไดร์ฟซีไม่เป็นคอขวด

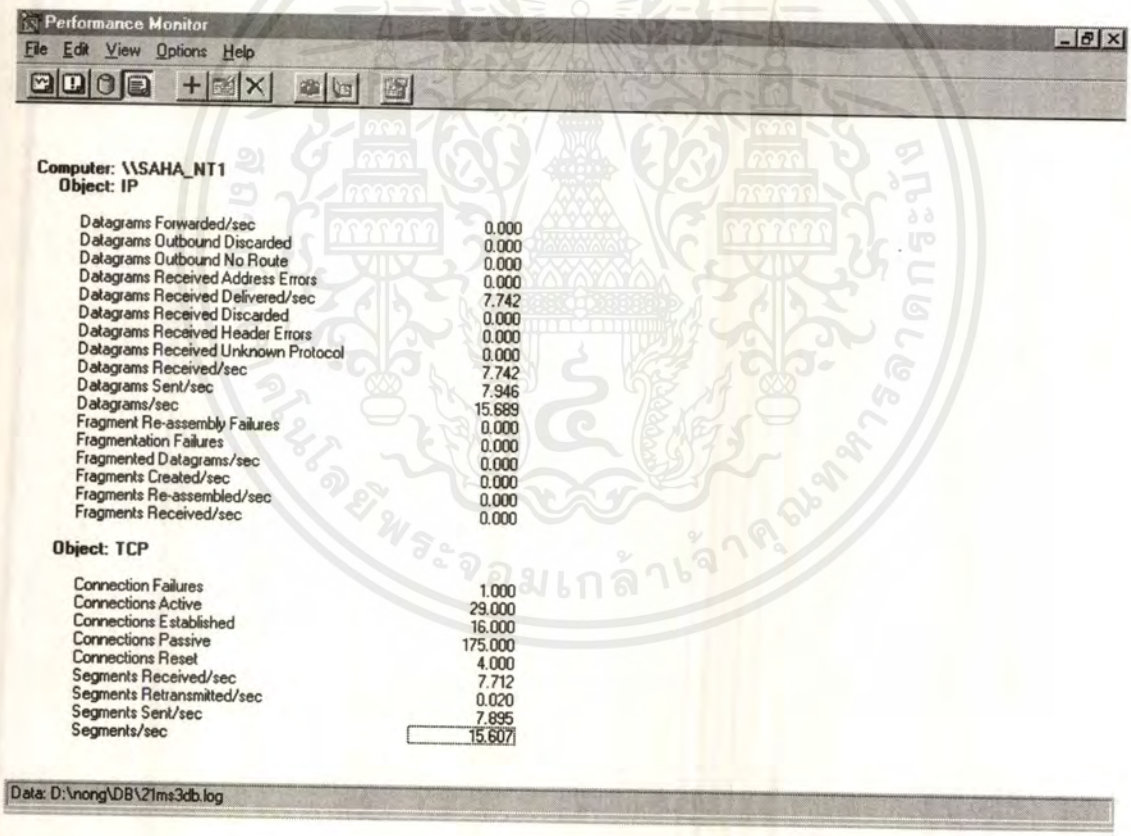
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ที่ไดร์ฟดีนั้นมีการใช้ดิสก์มาก คือมีการใช้มากโดยเฉลี่ยเป็น 94.2% ส่วนคิวก็มีค่ามากเช่นกัน โดยมีค่าเฉลี่ยสูงถึง 18.515 และค่ามากที่สุด คือ 45 คิว ดังนั้นจึงสรุปได้ว่าเกิดคอขวดที่ดิสก์ไดร์ฟดี
3. ที่ไดร์อีและเอฟนั้นไม่มีการส่งข้อมูลจากดิสก์หรือไปยังดิสก์เลย

สรุปเกิดคอขวดที่ดิสก์ไดร์ฟดี

การตรวจจับคอขวดที่แคช

1. Copy Read Hits % และ Copy Reads/sec มีค่าเป็นศูนย์
2. Data Flushes/sec มีค่า 1.713 ซึ่งหมายถึงแคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วงเพียง 1.713
3. ค่า Data Map Hits % มีค่าสูงถึง 97.98 % แต่ค่า Data Maps/sec มีค่าต่ำเพียง 3.537 อาจจะทำให้ประสิทธิภาพไม่ดีก็ได้ถ้าอัตราการฮิตสูง แต่อัตราการแมปต่ำ



รูปที่ 8-60 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Computer: \\SAHA_NT1
Object: Network Interface

	1	2
Bytes Received/sec	0.000	2413.042
Bytes Sent/sec	0.000	2035.018
Bytes Total/sec	0.000	4448.060
Current Bandwidth	10000000.000	10000000.000
Output Queue Length	0.000	0.000
Packets Outbound Discarded	0.000	0.000
Packets Outbound Errors	0.000	0.000
Packets Received Discarded	0.000	0.000
Packets Received Errors	0.000	0.000
Packets Received Non-Unicast/sec	0.000	0.041
Packets Received Unicast/sec	0.000	7.773
Packets Received Unknown	0.000	0.000
Packets Received/sec	0.000	7.814
Packets Sent Non-Unicast/sec	0.000	0.031
Packets Sent Unicast/sec	0.000	7.926
Packets Sent/sec	0.000	7.956
Packets/sec	0.000	15.770

Data: D:\nong\DB\21ms3db.log

รูปที่ 8-61 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียดพร้อมกัน

การตรวจจับคอขวดที่เครือข่าย

1. ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface)

เนื่องจากเครือข่ายที่ใช้ทดลองนี้เป็นเครือข่ายอีเทอร์เน็ต (Ethernet) ดังนั้นจึงมีอัตราการส่งเป็น 10 เมกะบิตต่อวินาที (10 Mbps.) และสามารถคิดเป็น %Utilization ของการส่งข้อมูลเป็นไบต์ จากค่าของ Bytes Total/sec ได้ดังนี้

4448.060 Bytes Total/sec = 4448.060 * 8 Bits/sec

เพราะฉะนั้น $\%\text{Byte Utilization} = \frac{\text{จำนวน Bits/sec} * 100}{10\text{Mbps.}}$

$$= \frac{4448.060 * 8 * 100}{10 * 10^6} = 0.356\%$$

และสามารถหาจำนวนแพ็คเก็ตต่อการส่งใน 1 วินาทีของอีเทอร์เน็ตได้ดังนี้

จำนวนไบต์ต่อแพ็คเก็ต = 512 Bytes/packet = 512 * 8 Bits/packet

ดังนั้น $\text{Packets/sec} = \frac{10\text{ Mbps}}{512 * 8\text{ Bits/packet}} = \frac{10 * 10^6}{512 * 8}$ Packets/sec

เพราะฉะนั้น $\%\text{Packet Utilization} = \frac{15.770 * 512 * 8}{10 * 10^6} * 100$

$= 0.646\ \%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาจากค่า %Utilization ทั้ง 2 ค่าแล้วจะเห็นว่ามีค่าค่อนข้างต่ำ แสดงว่าการรับส่งข้อมูลมีค่าน้อยมากเมื่อเทียบกับค่าที่มันสามารถรองรับได้

2. ออบเจ็กต์ : ไอพี (Object : IP)

เมื่อดูจากค่า Datagrams Received Delivered/sec พบว่ามีค่าเท่ากับ Datagrams Received/sec แสดงว่ามีการส่งดาต้าแกรมที่ได้รับมา ออกไปสำเร็จทุกดาต้าแกรม

3.ออบเจ็กต์ : ทีซีพี (Object : TCP)

พิจารณาเปอร์เซ็นต์การติดต่อ (connection) ที่ไม่สำเร็จ โดยคิดเทียบกับจำนวนการติดต่อทั้งหมด (คือ ทั้งด้านรับและด้านส่ง) และจำนวนการติดต่อทั้งหมดคิดได้จาก

ค่า Connections Active (ด้านส่งข้อมูล) + ค่า Connections Passive (ด้านรับข้อมูล)

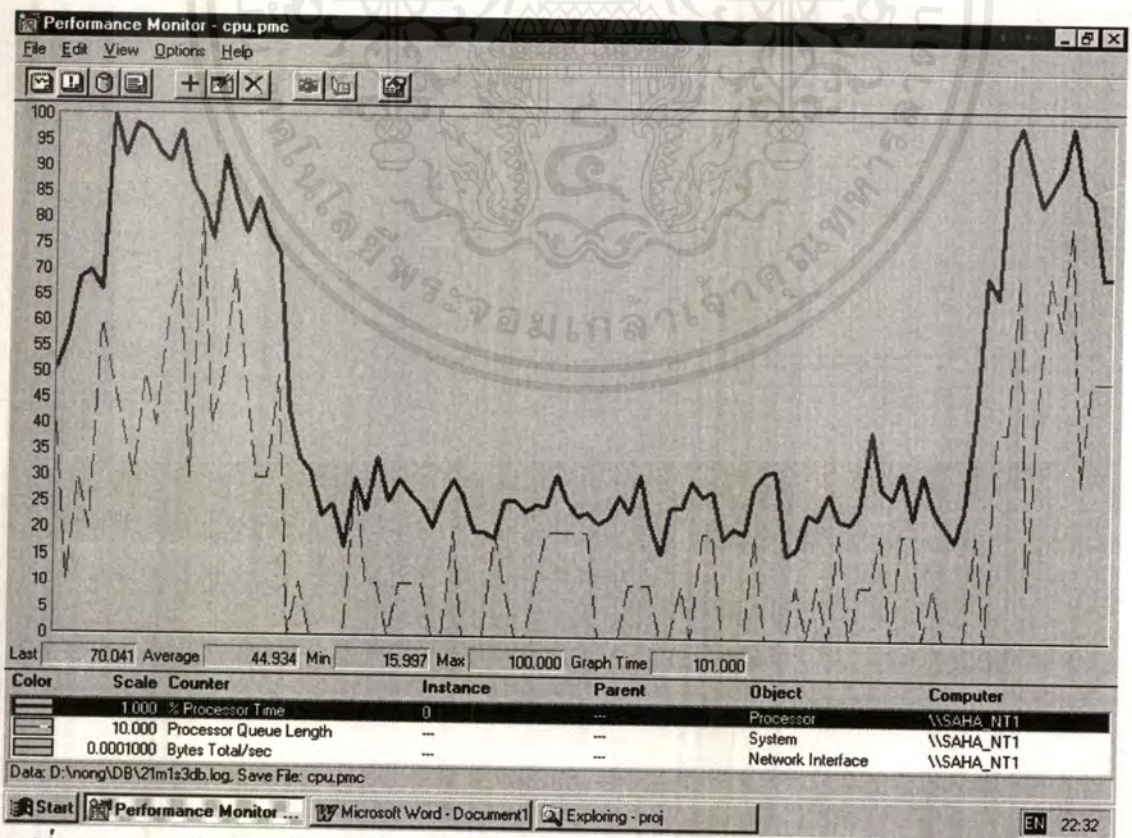
เพราะฉะนั้นเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จ $= \frac{\text{Connection Failures}}{\text{Connections Active + Connection Passive}} * 100$

$= \frac{1}{29+175} * 100 = 0.49\%$

จะเห็นว่าเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จนั้นน้อยมาก แสดงว่าการรับส่งข้อมูลของชั้นทีซีพีนี้มีความผิดพลาดน้อยมาก จึงไม่เกิดปัญหาขึ้น

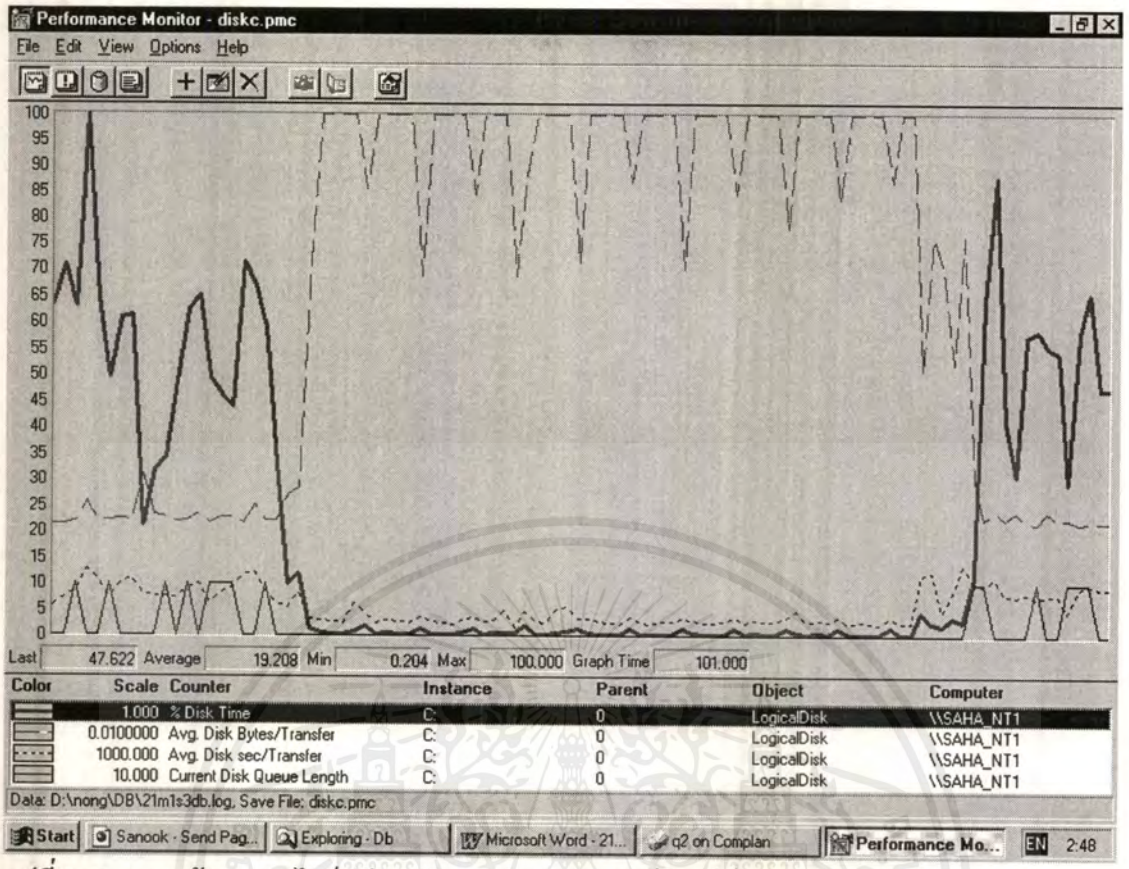
สรุป ไม่เกิดคอขวดที่เครือข่าย

3. แมสออเดอร์ คีย์รายละเอียด ห่างกัน 1 วินาที

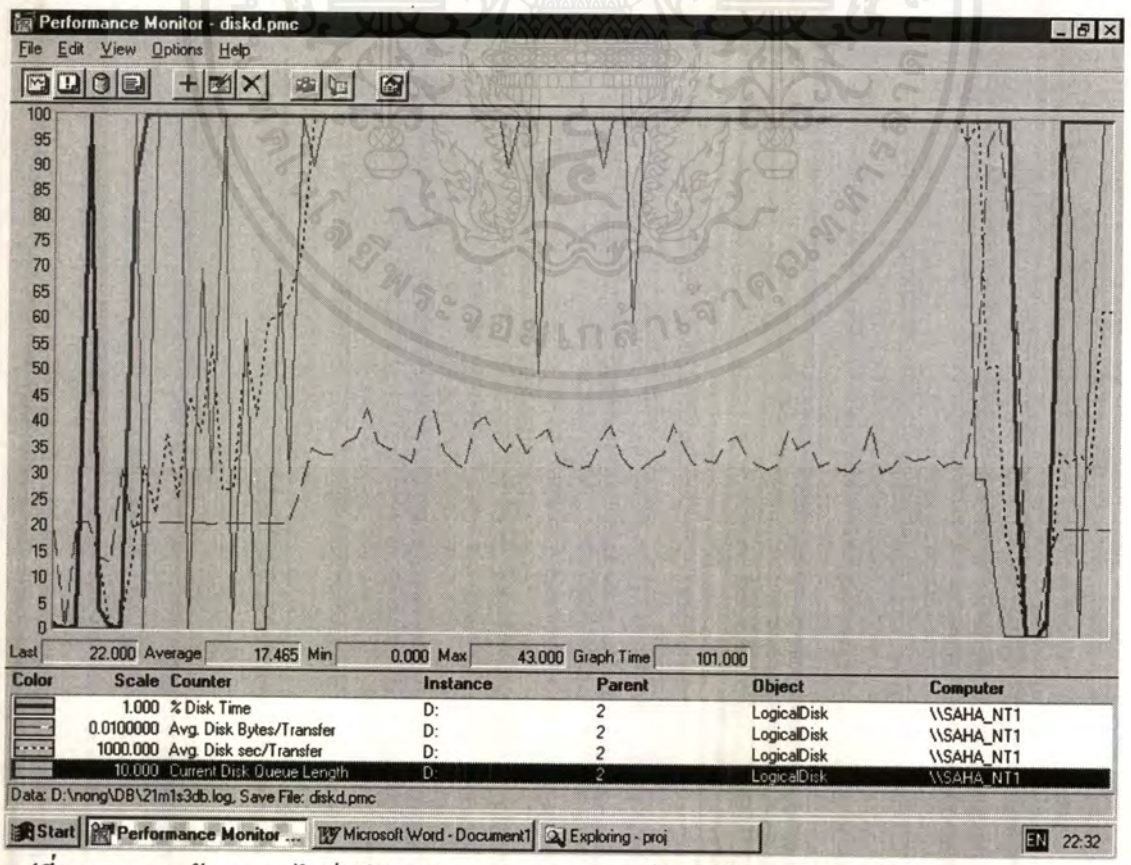


รูปที่ 8-62 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียด ห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

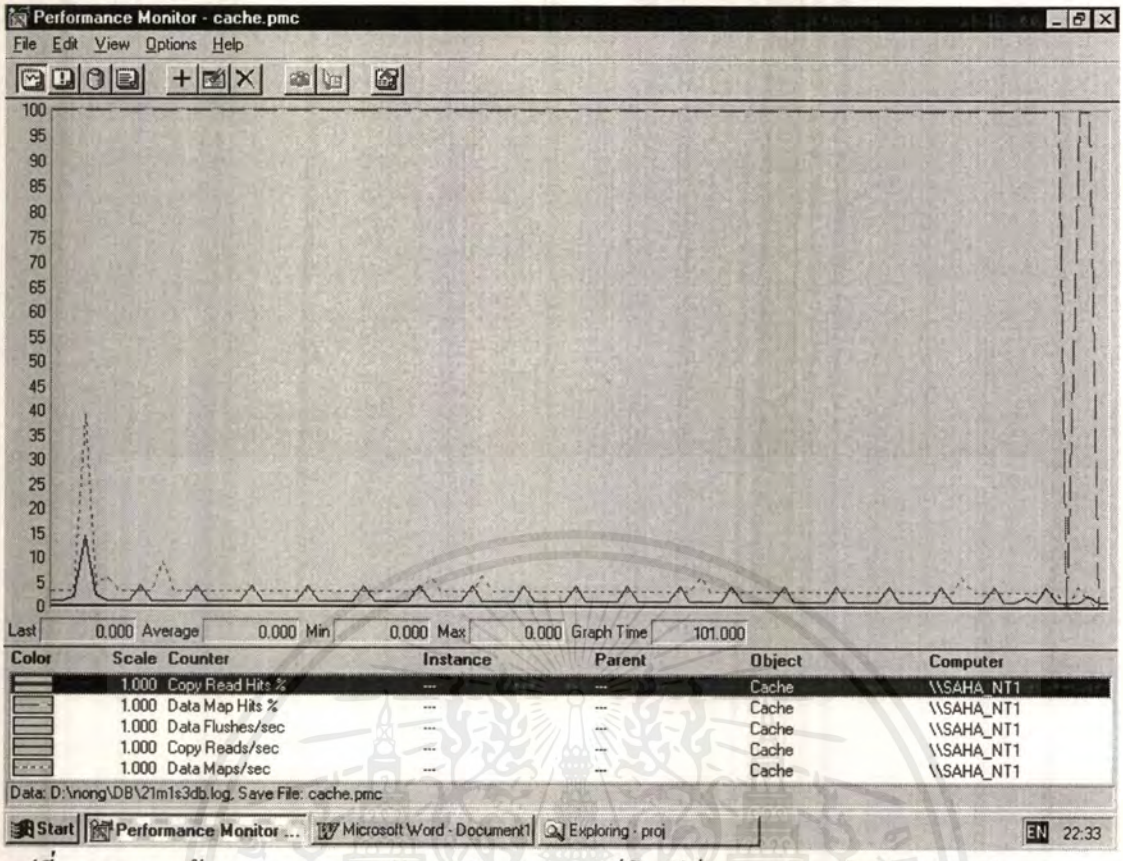


รูปที่ 8-63 แสดงข้อมูลของไดร์ฟซีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียด ห่างกัน 1 วินาที



รูปที่ 8-64 แสดงข้อมูลของไดร์ฟดีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียด ห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-65 แสดงข้อมูลของแคชสำหรับการทำแมลออเดอร์ โดยคีย์รายละเอียด ห่างกัน 1 วินาที

การตรวจจับคอขวดที่โพรเซสเซอร์

1. %Processor Time มีการใช้โพรเซสเซอร์มากตรงช่วงเริ่มต้นและช่วงท้าย ทั้ง 2 ช่วงนี้มีเปอร์เซ็นต์การใช้โพรเซสเซอร์ประมาณ 60-70% คือมีค่าไม่มากจนเกินไป แต่ค่าเฉลี่ยรวมตลอดการล็อคเท่ากับ 44.379 % แสดงว่ามีการใช้โพรเซสเซอร์ในกระบวนการทำงานไม่มากนัก
2. เมื่อดูที่ Processor Queue Length พบว่ามีค่าไม่มากนัก โดยมีค่าเฉลี่ย 1.96 และมีค่าสูงที่สุด คือ 8 สรุปได้ว่าไม่เกิดคอขวดที่โพรเซสเซอร์

การตรวจจับคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเพียง 0.03 เพจ แสดงว่ามีอัตราการเพจจิ้งน้อยมาก
2. Available Bytes มีค่าประมาณ 9 MB หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่าเทรสโฮลที่กำหนดคือ 1MB
3. Commit Limit มีค่าประมาณ 400 MB ,Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิตได้ โดยไม่ต้องเพิ่มขนาดของ paging file ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ Paging file.sys รวมกับ ขนาดของแรมที่สามารถเปลี่ยนกลับไปยังดิสก์ได้ (หรือจำนวนของแรมที่ว่างที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 77 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ Paging file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ดูค่า Page Faults/sec ซึ่งมีค่า 20.001 เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.108 ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำ แต่กำลังถูกเขียนลงบนดิสก์ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอลลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของไดร์ฟต่าง ๆ ซึ่งมีค่าไม่ใกล้เคียงกันเลย
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าเป็นศูนย์ ซึ่งเท่ากับไดร์ฟอีและเอฟ
9. Page Inputs/sec * 4096 bytes/page = 4096 bytes/read เทียบกับค่า
   Page Reads/sec
   Avg. Disk Bytes/Read ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
10. Page Outputs/sec * 4096 bytes/page = 0 bytes/write เทียบกับค่า
    Page Writes/sec
    Avg. Disk Bytes/Write ซึ่งเท่ากับไดร์ฟอีและเอฟ

สรุปไม่เกิดคอขวดที่หน่วยความจำ

การตรวจจับคอขวดที่ดิสก์

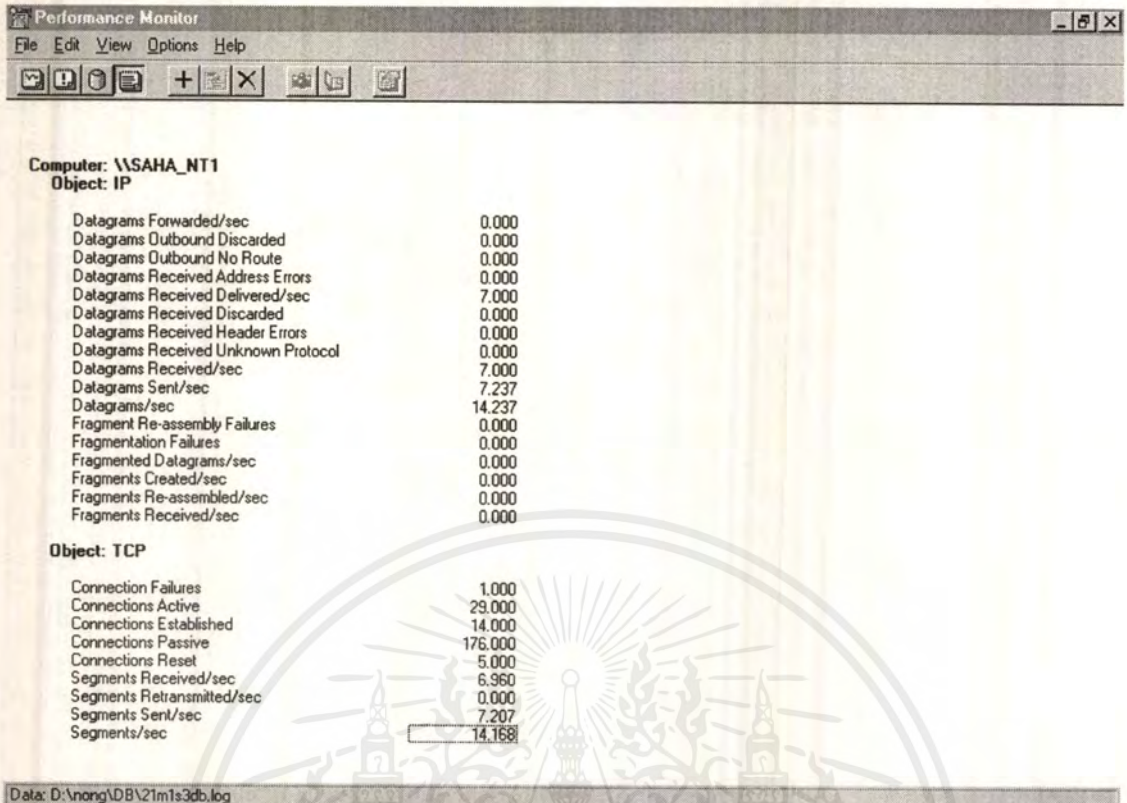
1. ที่ไดร์ฟซี มีการใช้ดิสก์ไดร์ฟซีค่อนข้างน้อย คือมี %Disk Time เพียง 18.839% ส่วนคิวของดิสก์นั้นมีค่าเฉลี่ยเป็น 0.131 เท่านั้น จึงทำให้สรุปได้ว่าดิสก์ไดร์ฟซีไม่เป็นคอขวด
2. ที่ไดร์ฟดีนั้น จากกราฟมีการใช้ดิสก์มาก มีเปอร์เซ็นต์การใช้ดิสก์ถึง 89.113% และมีค่าสูงสุดถึง 100% เกือบตลอดช่วงการล็อค ส่วนคิวนั้นมีค่ามากเช่นกัน โดยมีค่าเฉลี่ยถึง 17.323 และค่ามากที่สุดคือ 43 คิว และจากกราฟจะเห็นว่าเมื่อมีการใช้ไดร์ฟดีมาก เวลาที่ใช้ในการโอนย้ายข้อมูลก็มากไปด้วย ดังนั้นจึงสรุปได้ว่าเกิดคอขวดที่ดิสก์ไดร์ฟดี
3. ที่ไดร์อีและเอฟนั้นไม่มีการส่งข้อมูลจากดิสก์หรือไปยังดิสก์เลย

สรุปว่าเกิดคอขวดที่ดิสก์ไดร์ฟดี

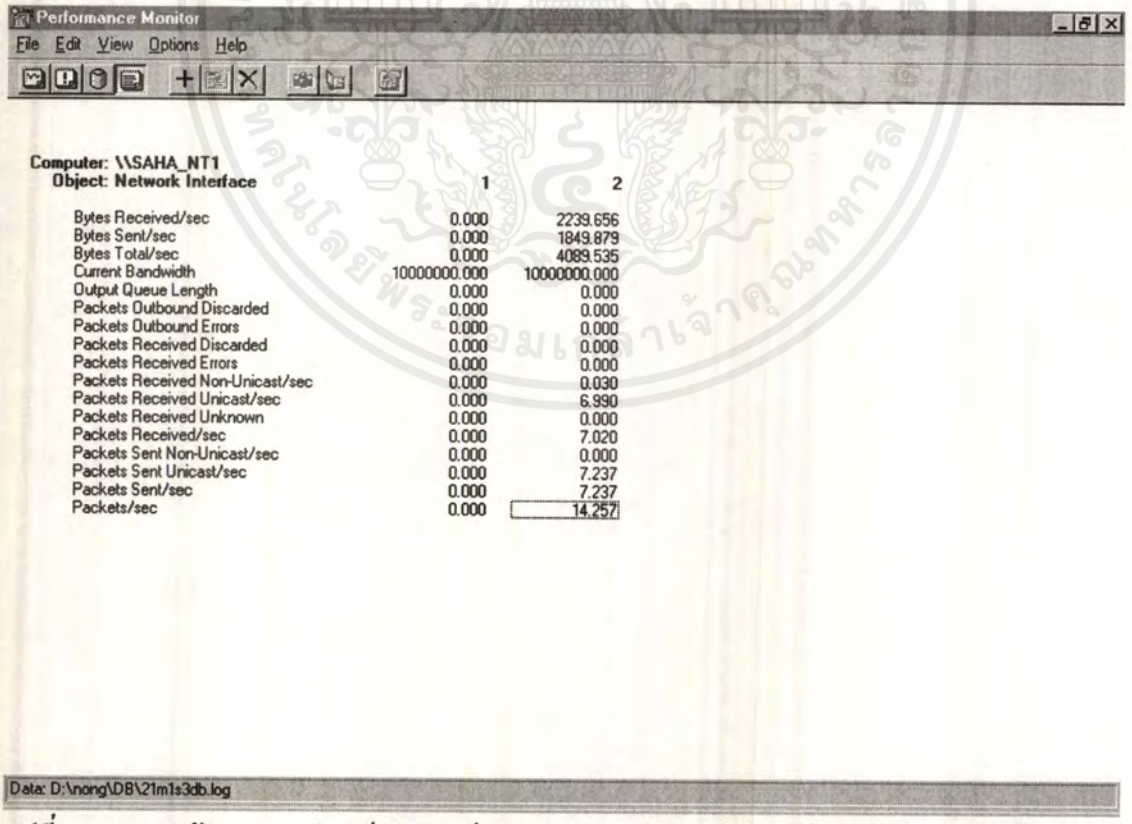
การตรวจจับคอขวดที่แคช

1. Copy Read Hits % และ Copy Reads/sec มีค่าเป็นศูนย์
2. Data Flushes/sec มีค่า 1.746 ซึ่งหมายถึงแคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วงเพียง 1.746
3. ค่า Data Map Hits % มีค่าสูงถึง 98.99 % แต่ค่า Data Maps/sec มีค่าต่ำเพียง 3.552 อาจจะทำให้ประสิทธิภาพไม่ดีก็ได้ถ้าอัตราการฮิตสูง แต่อัตราการแมปต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-66 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียด ห่างกัน 1 วินาที



รูปที่ 8-67 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำแมสออเดอร์ โดยคีย์รายละเอียด ห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับคอขวดที่เครือข่าย

1. ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface)

เนื่องจากเครือข่ายที่ใช้ทดลองนี้เป็นเครือข่ายอีเทอร์เน็ต (Ethernet) ดังนั้นจึงมีอัตราการส่งเป็น 10 เมกะบิตต่อวินาที (10 Mbps.) และสามารถคิดเป็น %Utilization ของการส่งข้อมูลเป็นไบต์ จากค่าของ Bytes Total/sec ได้ดังนี้

4089.535 Bytes Total/sec = 4089.535 * 8 Bits/sec

เพราะฉะนั้น %Byte Utilization = $\frac{\text{จำนวน Bits/sec}}{\text{10Mbps.}} * 100$

$$= \frac{4089.535 * 8}{10 * 10^6} * 100 = 0.327\%$$

และสามารถหาจำนวนแพ็คเก็ตต่อการส่งใน 1 วินาทีของอีเทอร์เน็ตได้ดังนี้

จำนวนไบต์ต่อแพ็คเก็ต = 512 Bytes/packet = 512 * 8 Bits/packet

ดังนั้น Packets/sec = $\frac{10 \text{ Mbps}}{512 * 8 \text{ Bits/packet}} = \frac{10 * 10^6}{512 * 8}$ Packets/sec

เพราะฉะนั้น %Packet Utilization $= \frac{14.257 * 512 * 8}{10 * 10^6} * 100$

$= 0.584\ \%$

เมื่อพิจารณาจากค่า %Utilization ทั้ง 2 ค่าแล้วจะเห็นว่ามีค่าค่อนข้างต่ำ แสดงว่าการรับส่งข้อมูลมีค่าน้อยมากเมื่อเทียบกับค่าที่มันสามารถรองรับได้

2. ออบเจ็กต์ : ไอพี (Object : IP)

เมื่อดูจากค่า Datagrams Received Delivered/sec พบว่ามีค่าเท่ากับ Datagrams Received/sec แสดงว่ามีการส่งดาต้าแกรมที่ได้รับมา ออกไปสำเร็จทุกดาต้าแกรม

3.ออบเจ็กต์ : ทีซีพี (Object : TCP)

พิจารณาเปอร์เซ็นต์การติดต่อ (connection) ที่ไม่สำเร็จ โดยคิดเทียบกับจำนวนการติดต่อทั้งหมด (คือ ทั้งด้านรับและด้านส่ง) และจำนวนการติดต่อทั้งหมดคิดได้จาก

ค่า Connections Active (ด้านส่งข้อมูล) + ค่า Connections Passive (ด้านรับข้อมูล)

เพราะฉะนั้นเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จ = $\frac{\text{Connection Failures}}{\text{Connections Active + Connection Passive}} * 100$

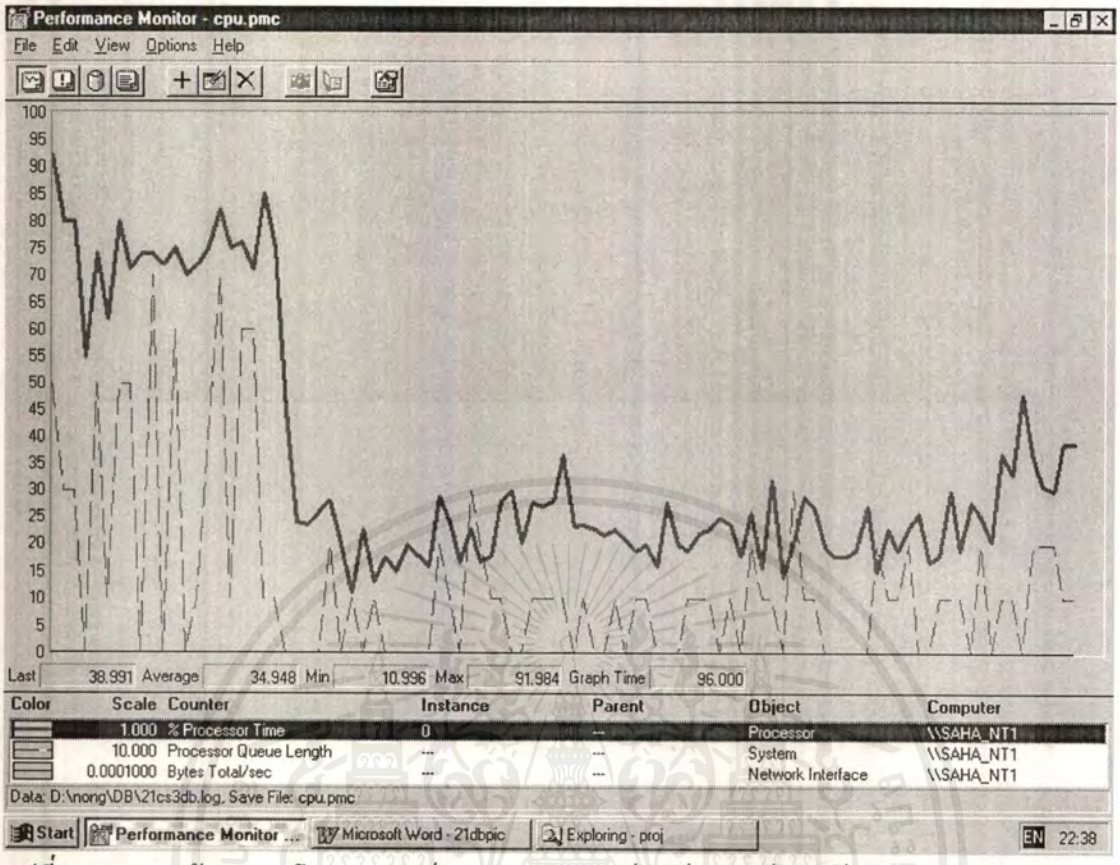
$$= \frac{1}{29+176} * 100 = 0.488\%$$

จะเห็นว่าเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จนั้นน้อยมาก แสดงว่าการรับส่งข้อมูลของชั้นทีซีพีนี้มีความผิดพลาดน้อยมาก จึงไม่เกิดปัญหาขึ้น

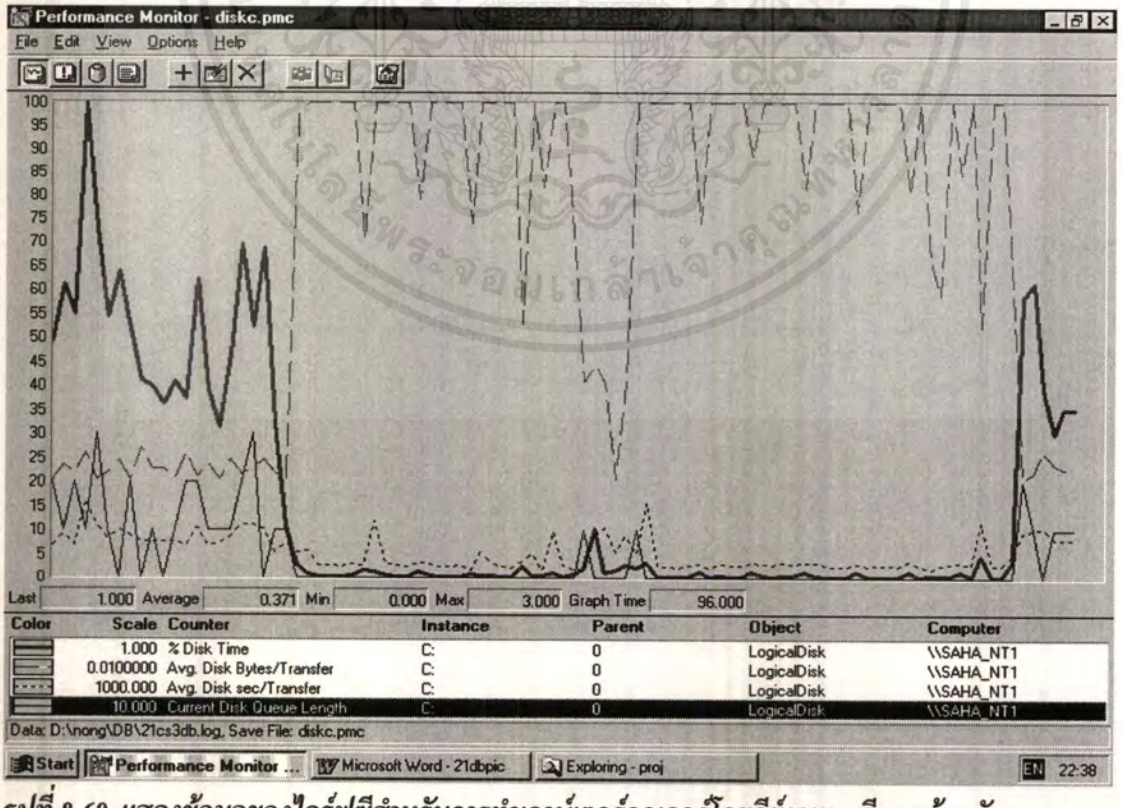
สรุป ไม่เกิดคอขวดที่เครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เคาน์เตอร์ออเดอร์ คีย์รายละเอียดพร้อมกัน

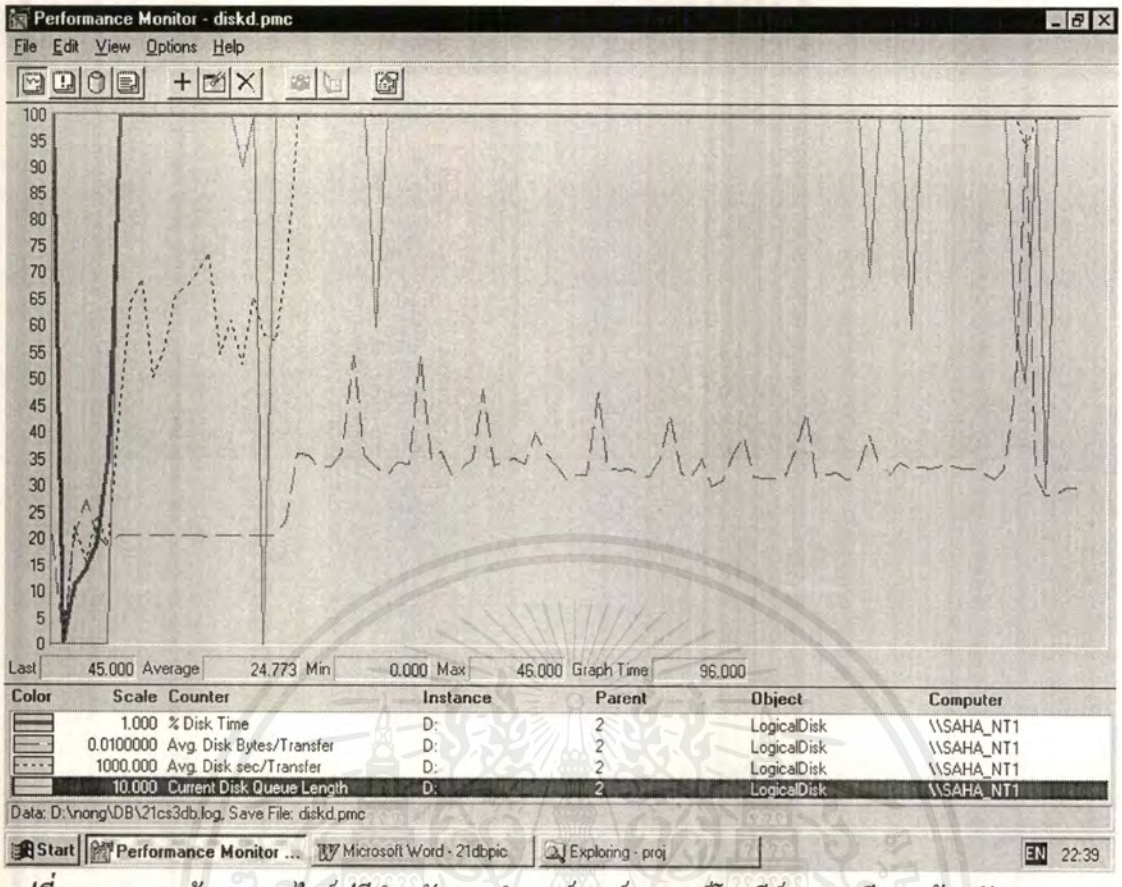


รูปที่ 8-68 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

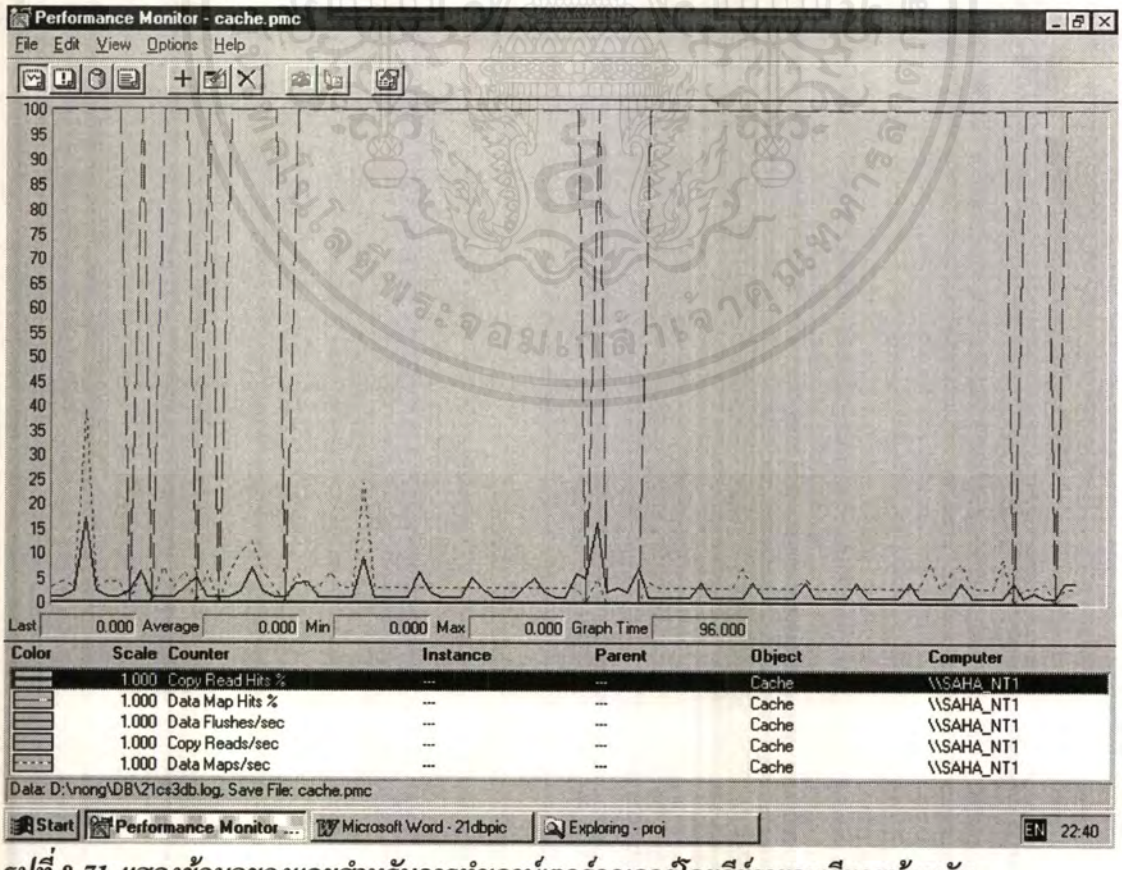


รูปที่ 8-69 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-70 แสดงข้อมูลของไดร์ฟดีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน



รูปที่ 8-71 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับคอขวดที่โพรเซสเซอร์

1. %Processor Time มีการใช้โพรเซสเซอร์ค่อนข้างน้อย คือมีค่าเฉลี่ยเพียง 34.948% แสดงว่ามีการใช้โพรเซสเซอร์ในกระบวนการทำงานค่อนข้างน้อย
2. เมื่อดูที่ Processor Queue Length พบว่ามีค่าน้อยเช่นกัน โดยมีค่าเฉลี่ย 1.268 และมีค่าสูงที่สุด คือ 7 สรุปได้ว่าไม่เกิดคอขวดที่โพรเซสเซอร์

การตรวจจับคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเท่ากับ 0.031 เพจ แสดงว่ามีอัตราการเพจจิ้งน้อยมาก
2. Available Bytes มีค่าประมาณ 6 MB หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่าเทรสโฮลที่กำหนดคือ 1MB
3. Commit Limit มีค่าประมาณ 400 MB ,Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิตได้ โดยไม่ต้องเพิ่มขนาดของ paging file ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ Paging file.sys รวมกับ ขนาดของแรมที่สามารถเปลี่ยนกลับไปยังดิสก์ได้ (หรือจำนวนของแรมที่ว่างที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 79 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ Paging file
5. ดูค่า Page Faults/sec ซึ่งมีค่า 11.057 เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.042 ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำ แต่กำลังถูกเขียนลงบนดิสก์ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอลลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของไดร์ฟต่าง ๆ ซึ่งมีค่าไม่ใกล้เคียงกันเลย
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าเท่ากับศูนย์ ซึ่งเท่ากับไดร์ฟอีและเอฟ
9. Page Inputs/sec * 4096 bytes/page = 4096 bytes/read เทียบกับค่า
   Page Reads/sec
   Avg. Disk Bytes/Read ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
10. Page Outputs/sec * 4096 bytes/page = 0 bytes/write เทียบกับค่า
    Page Writes/sec
    Avg. Disk Bytes/Write ซึ่งเท่ากับไดร์ฟอีและเอฟ

สรุปไม่เกิดคอขวดที่หน่วยความจำ

การตรวจจับคอขวดที่ดิสก์

1. ที่ไดร์ฟซี มีการใช้ดิสก์ไดร์ฟซีน้อยมาก คือมี %Disk Time เพียง 14.886% ส่วนคิวของดิสก์นั้นมีค่าเฉลี่ยน้อยเช่นกัน คือ 0.371 เท่านั้น จึงทำให้สรุปได้ว่าดิสก์ไดร์ฟซีไม่เป็นคอขวด

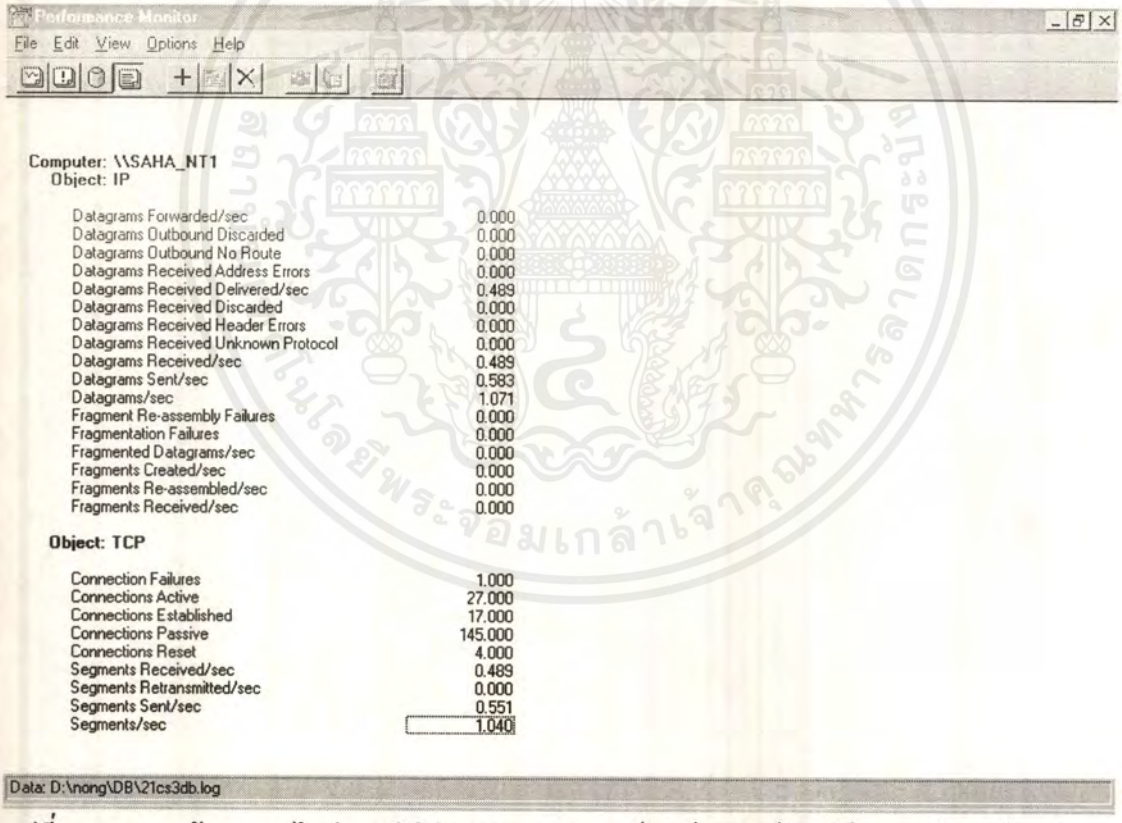
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **ที่ไดร์ฟดีนั้น มีการใช้ดิสก์ไดร์ฟดีมาก คือ สูงถึง 95.666%** จากกราฟจะเห็นว่ามีการใช้ดิสก์มากที่สุดถึง 100% เกือบตลอดช่วงของการล็อค ส่วนคิวนั้นมีค่ามากเช่นกัน โดยมีค่าเฉลี่ยถึง 24.773 และค่ามากที่สุดถึง คือ 46 คิว ดังนั้นจึงสรุปได้ว่าเกิดคอขวดที่ดิสก์ไดร์ฟดี
3. ที่ไดร์อีและเอฟนั้นไม่มีการส่งข้อมูลจากดิสก์หรือไปยังดิสก์เลย

สรุปว่าเกิดคอขวดที่ดิสก์ไดร์ฟดี

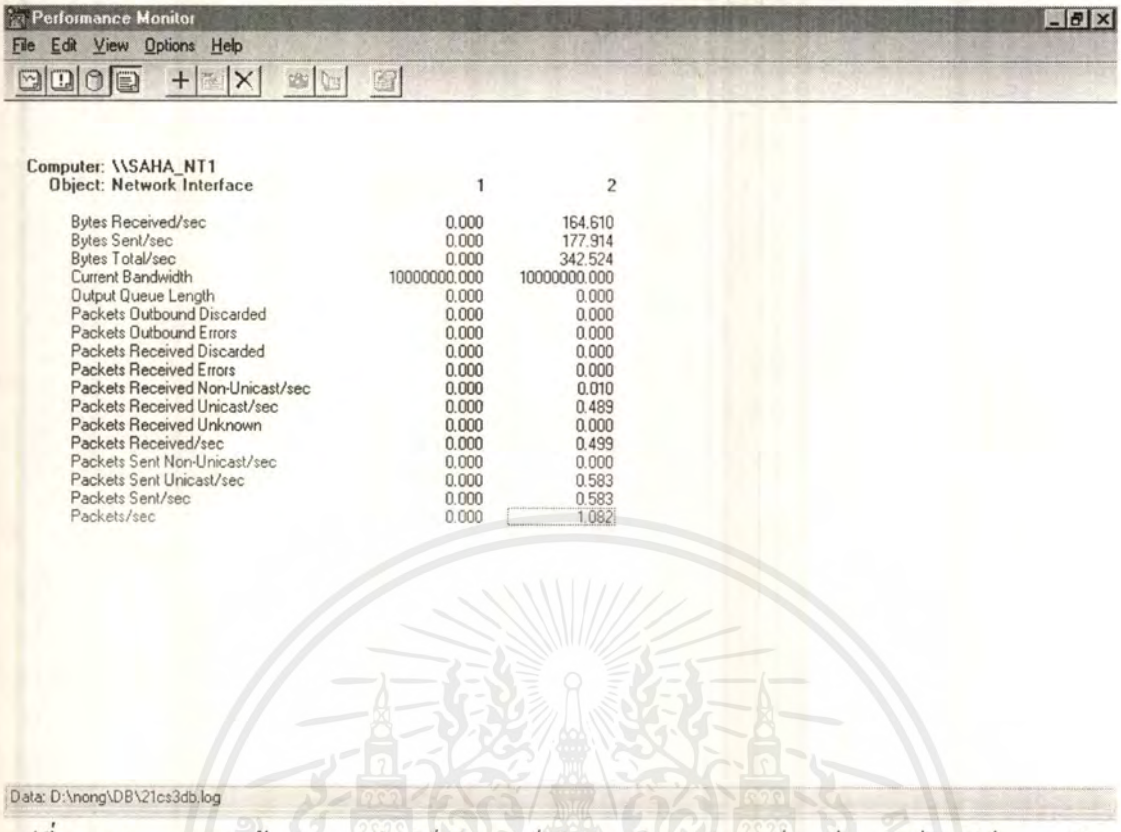
การตรวจจับคอขวดที่แคช

1. Copy Read Hits % และ Copy Reads/sec มีค่าเป็นศูนย์
2. Data Flushes/sec มีค่า 2.368 ซึ่งหมายถึงแคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วงเพียง 2.368
3. ค่า Data Map Hits % มีค่า 87.629 % แต่ค่า Data Maps/sec มีค่าต่ำเพียง 3.964 อาจจะทำให้ประสิทธิภาพไม่ดีก็ได้ถ้าอัตราการฮิตสูง แต่อัตราการแมปต่ำ



รูปที่ 8-72 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-73 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดพร้อมกัน

การตรวจจับคอขวดที่เครือข่าย

1. ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface)

เนื่องจากเครือข่ายที่ใช้ทดลองนี้เป็นเครือข่ายอีเทอร์เน็ต (Ethernet) ดังนั้นจึงมีอัตราการส่งเป็น 10 เมกะบิตต่อวินาที (10 Mbps.) และสามารถคิดเป็น %Utilization ของการส่งข้อมูลเป็นไบต์ จากค่าของ Bytes Total/sec ได้ดังนี้

342.524 Bytes Total/sec = 342.524 * 8 Bits/sec

เพราะฉะนั้น $\%\text{Byte Utilization} = \frac{\text{จำนวน Bits/sec} * 100}{10\text{Mbps.}}$

$$= \frac{342.524 * 8 * 100}{10 * 10^6} = 0.027\%$$

และสามารถหาจำนวนแพ็คเก็ตต่อการส่งใน 1 วินาทีของอีเทอร์เน็ตได้ดังนี้

จำนวนไบต์ต่อแพ็คเก็ต = 512 Bytes/packet = 512 * 8 Bits/packet

ดังนั้น $\text{Packets/sec} = \frac{10\text{ Mbps}}{512 * 8\text{ Bits/packet}} = \frac{10 * 10^6}{512 * 8}$ Packets/sec

เพราะฉะนั้น $\%\text{Packet Utilization} = \frac{1.082 * 512 * 8 * 100}{10 * 10^6}$

$= 0.044\ \%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาจากค่า %Utilization ทั้ง 2 ค่าแล้วจะเห็นว่ามีค่าค่อนข้างต่ำ แสดงว่าการรับส่งข้อมูลมีค่าน้อยมากเมื่อเทียบกับค่าที่มันสามารถรองรับได้

2. ออบเจ็กต์ : ไอพี (Object : IP)

เมื่อดูจากค่า Datagrams Received Delivered/sec พบว่ามีค่าเท่ากับ Datagrams Received/sec แสดงว่ามีการส่งดาต้าแกรมที่ได้รับมา ออกไปสำเร็จทุกดาต้าแกรม

3.ออบเจ็กต์ : ทีซีพี (Object : TCP)

พิจารณาเปอร์เซ็นต์การติดต่อ (connection) ที่ไม่สำเร็จ โดยคิดเทียบกับจำนวนการติดต่อทั้งหมด (คือ ทั้งด้านรับและด้านส่ง) และจำนวนการติดต่อทั้งหมดคิดได้จาก

ค่า Connections Active (ด้านส่งข้อมูล) + ค่า Connections Passive (ด้านรับข้อมูล)

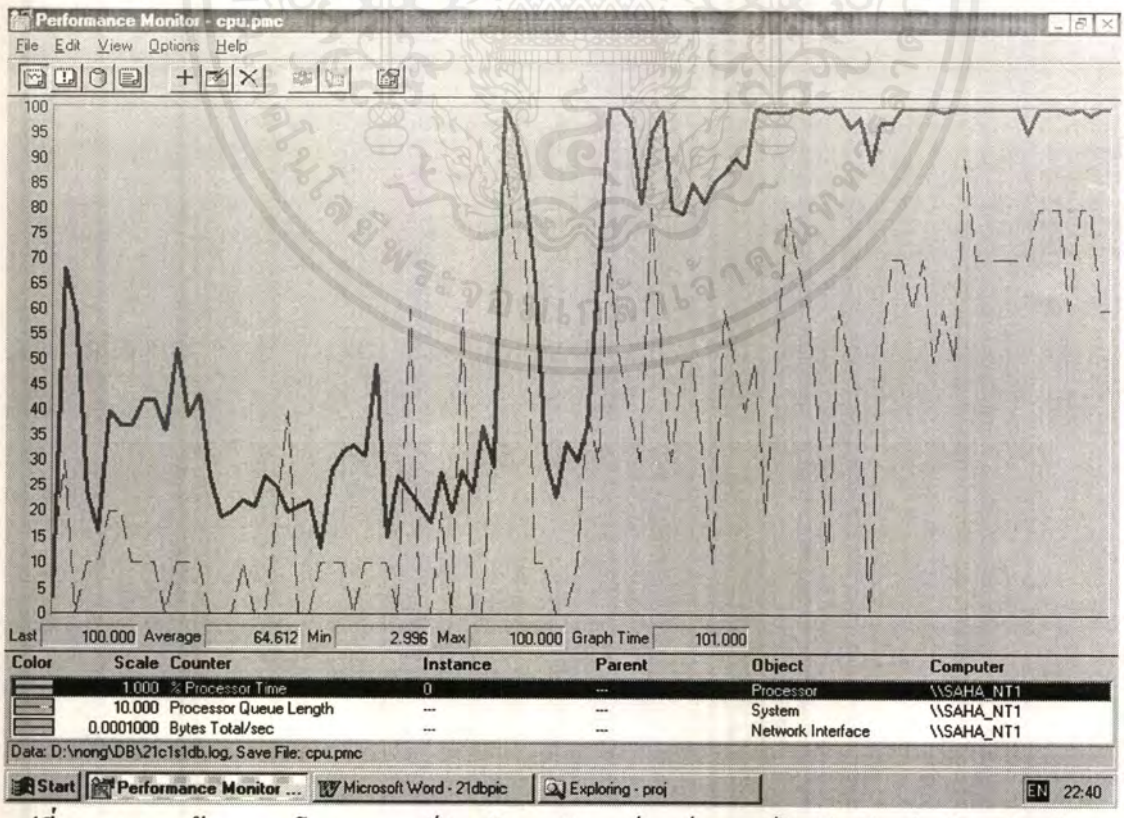
เพราะฉะนั้นเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จ $= \frac{\text{Connection Failures}}{\text{Connections Active + Connection Passive}} * 100$

$= \frac{1}{27+145} * 100 = 0.581\%$

จะเห็นว่าเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จนั้นน้อยมาก แสดงว่าการรับส่งข้อมูลของชั้นทีซีพีนี้มีความผิดพลาดน้อยมาก จึงไม่เกิดปัญหาขึ้น

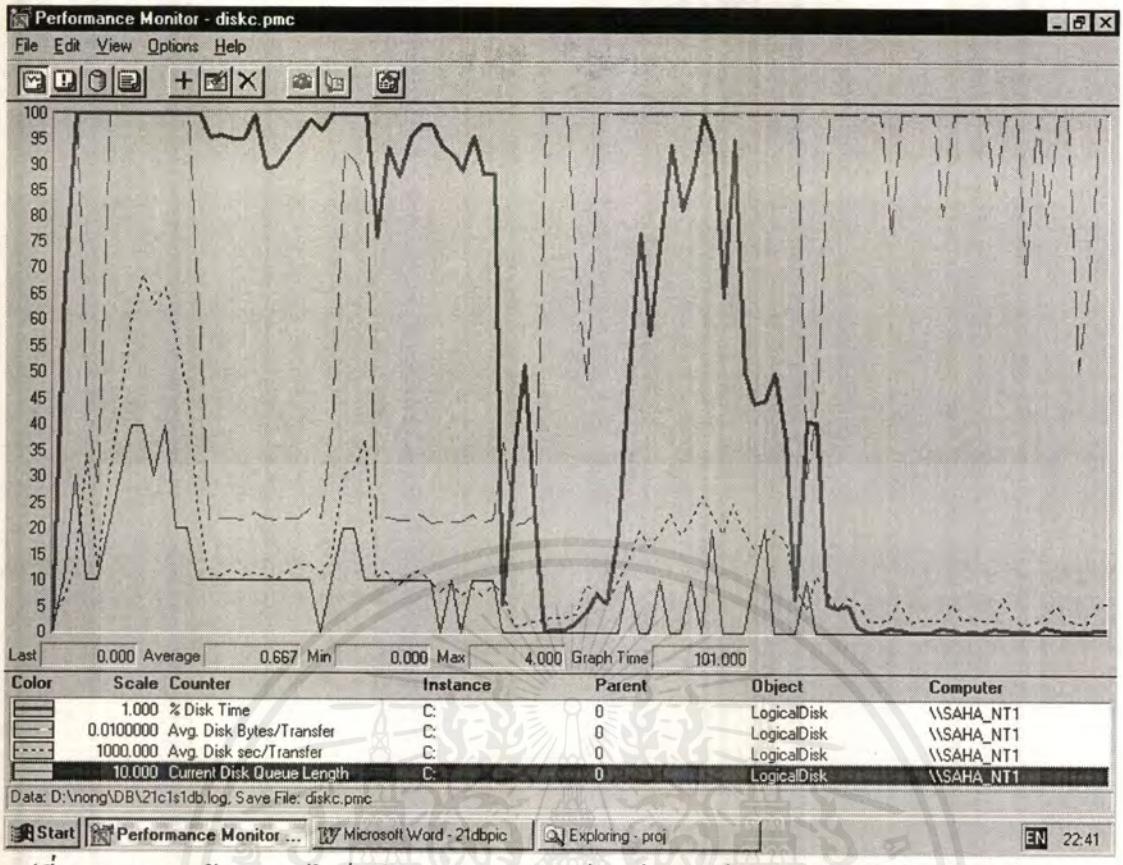
สรุป ไม่เกิดคอขวดที่เครือข่าย

5. เคาน์เตอร์ออเดอร์ ล็อกอิน ห่างกัน 1 วินาที

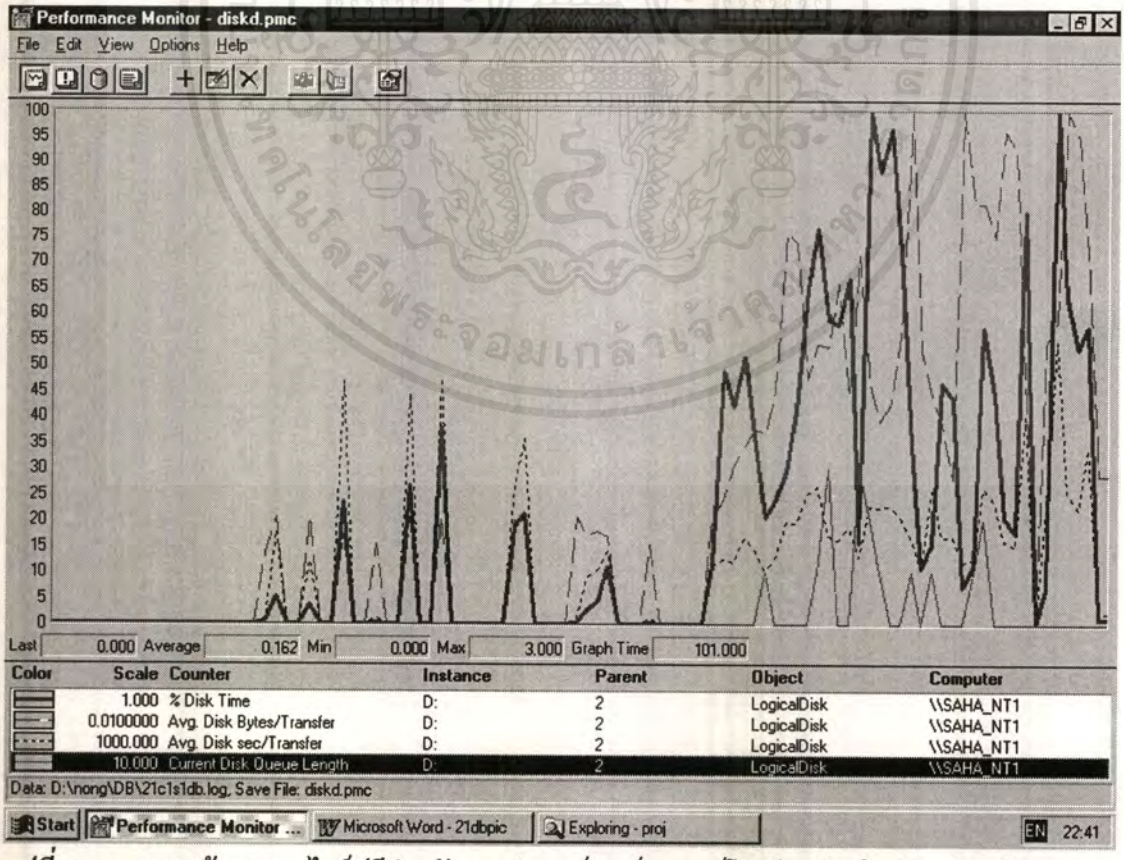


รูปที่ 8-74 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

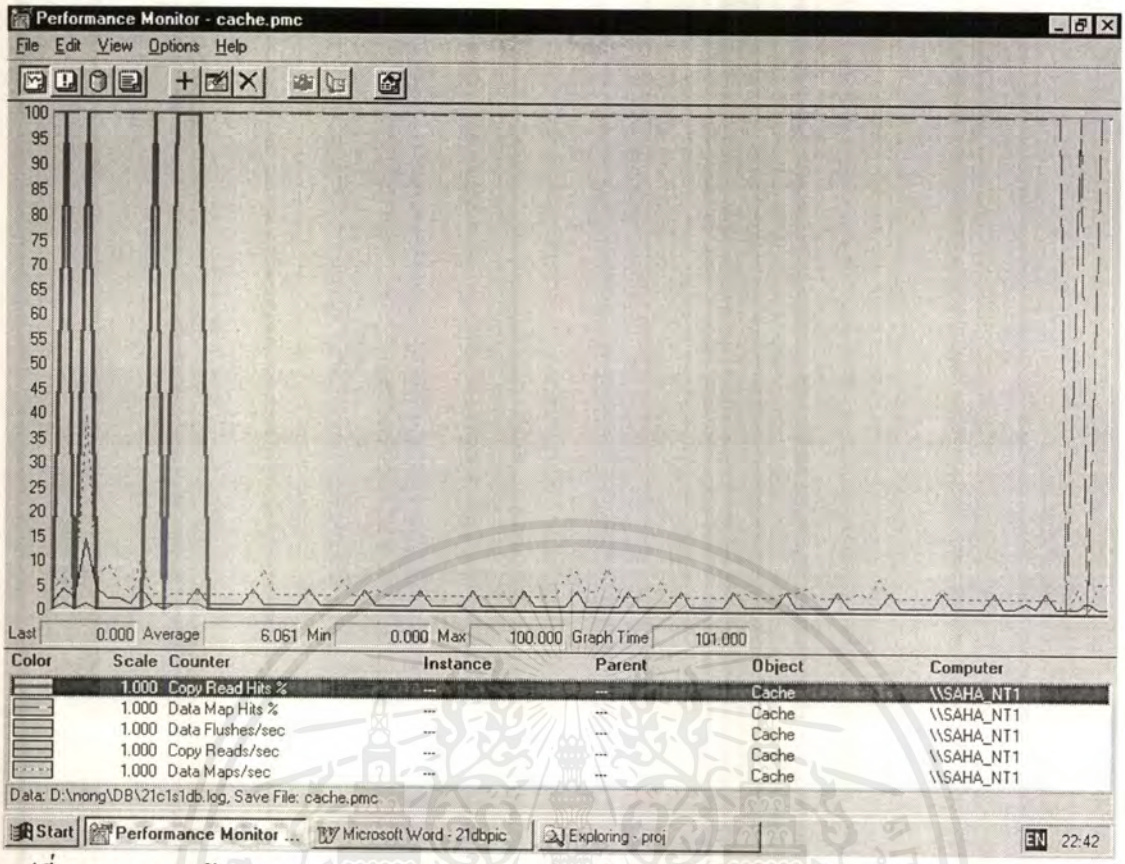


รูปที่ 8-75 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที



รูปที่ 8-76 แสดงข้อมูลของไดร์ฟดีสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-77 แสดงข้อมูลของแคชสำหรับการทำคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที

การตรวจจับคอขวดที่โพรเซสเซอร์

1. %Processor Time มีการใช้โพรเซสเซอร์มากในช่วงท้ายของการล็อค คือ มีการใช้โพรเซสเซอร์ประมาณ 90% ในช่วงท้าย เพราะฉะนั้นเกิดจุดคอขวดที่โพรเซสเซอร์ในช่วงท้าย แต่ค่าเฉลี่ยรวมเพียง 63.632 % เท่านั้น
2. เมื่อดูที่ Processor Queue Length พบว่ามีมากเช่นกัน โดยมีค่าเฉลี่ยถึง 3.465 และมีค่าสูงที่สุด คือ 9 สรุปได้ว่าเกิดคอขวดที่โพรเซสเซอร์ โดยเฉพาะในช่วงท้าย

การตรวจจับคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเท่ากับ 0.03 เพจ แสดงว่ามีอัตราการเพจจิ้งน้อยมาก
2. Available Bytes มีค่าประมาณ 8 MB หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่าเทรสโฮลที่กำหนดคือ 1MB
3. Commit Limit มีค่าประมาณ 400 MB ,Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิตได้ โดยไม่ต้องเพิ่มขนาดของ paging file ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ Paging file.sys รวมกับ ขนาดของแรมที่สามารถเปลี่ยนกลับไปยังดิสก์ได้ (หรือจำนวนของแรมที่ว่างที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 76 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ Paging file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ดูค่า Page Faults/sec ซึ่งมีค่า 43.599 เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.529 ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำ แต่กำลังถูกเขียนลงบนดิสก์ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอลลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของไดร์ฟต่าง ๆ ซึ่งมีค่าไม่ใกล้เคียงกันเลย
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าเท่ากับศูนย์ ซึ่งเท่ากับไดร์ฟอีและเอฟ
9. $\frac{\text{Page Inputs/sec}}{\text{Page Reads/sec}}$ * 4096 bytes/page = 4096 bytes/read เทียบกับค่า

   Avg. Disk Bytes/Read ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
10. $\frac{\text{Page Outputs/sec}}{\text{Page Writes/sec}}$ * 4096 bytes/page = 0 bytes/write เทียบกับค่า

    Avg. Disk Bytes/Write ซึ่งเท่ากับไดร์ฟอีและเอฟ

สรุปไม่เกิดคอขวดที่หน่วยความจำ

การตรวจจับคอขวดที่ดิสก์

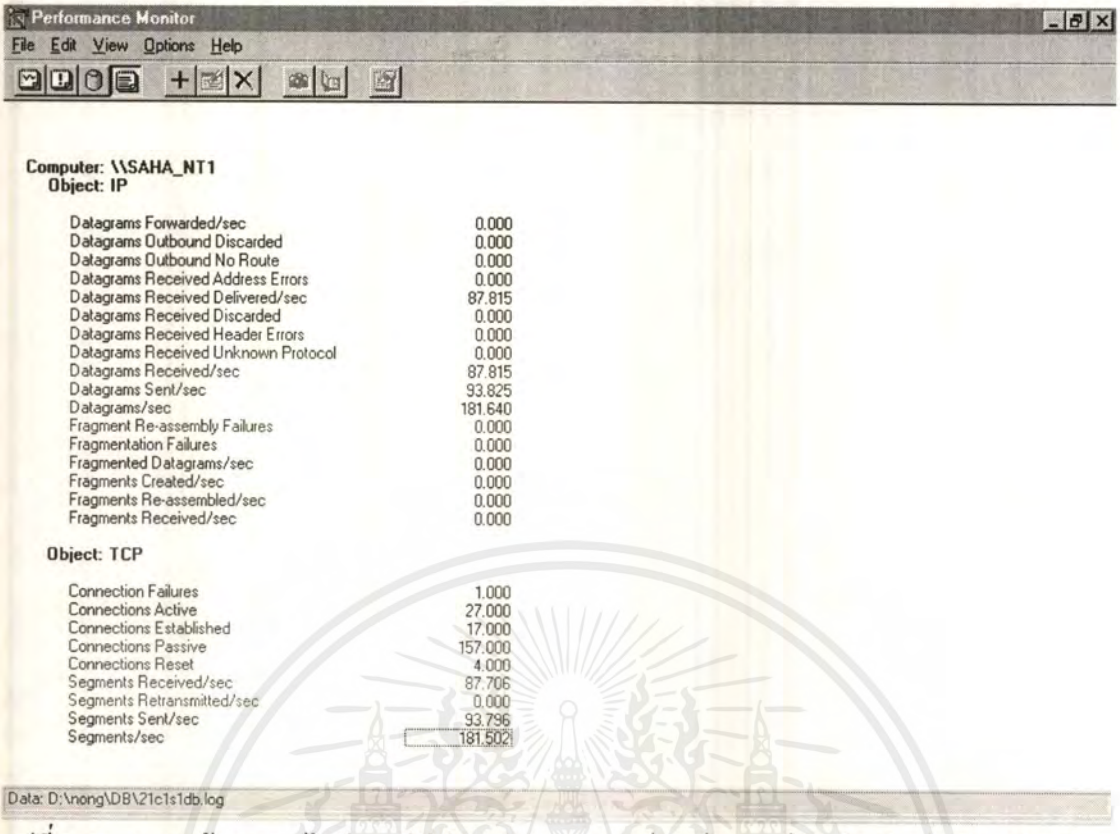
1. ที่ไดร์ฟซี มีการใช้ดิสก์ไดร์ฟซีปานกลาง คือมี %Disk Time 52.355% ส่วนคิวของดิสก์นั้นมีค่าเฉลี่ยคือ 0.667 เท่านั้น จึงทำให้สรุปได้ว่าดิสก์ไดร์ฟซีไม่เป็นคอขวด
2. ที่ไดร์ฟดีนั้น มีการใช้ดิสก์ไดร์ฟดีค่อนข้างน้อย คือ 18.322% ส่วนคิวนั้นมีค่าเฉลี่ย 0.162 และค่ามากที่สุด คือ 3 คิว ดังนั้นจึงสรุปได้ว่าไม่เกิดคอขวดที่ดิสก์ไดร์ฟดี
3. ที่ไดร์อีและเอฟนั้นไม่มีการส่งข้อมูลจากดิสก์หรือไปยังดิสก์เลย

สรุปว่าไม่เกิดคอขวดที่ดิสก์ไดร์ฟใดเลย

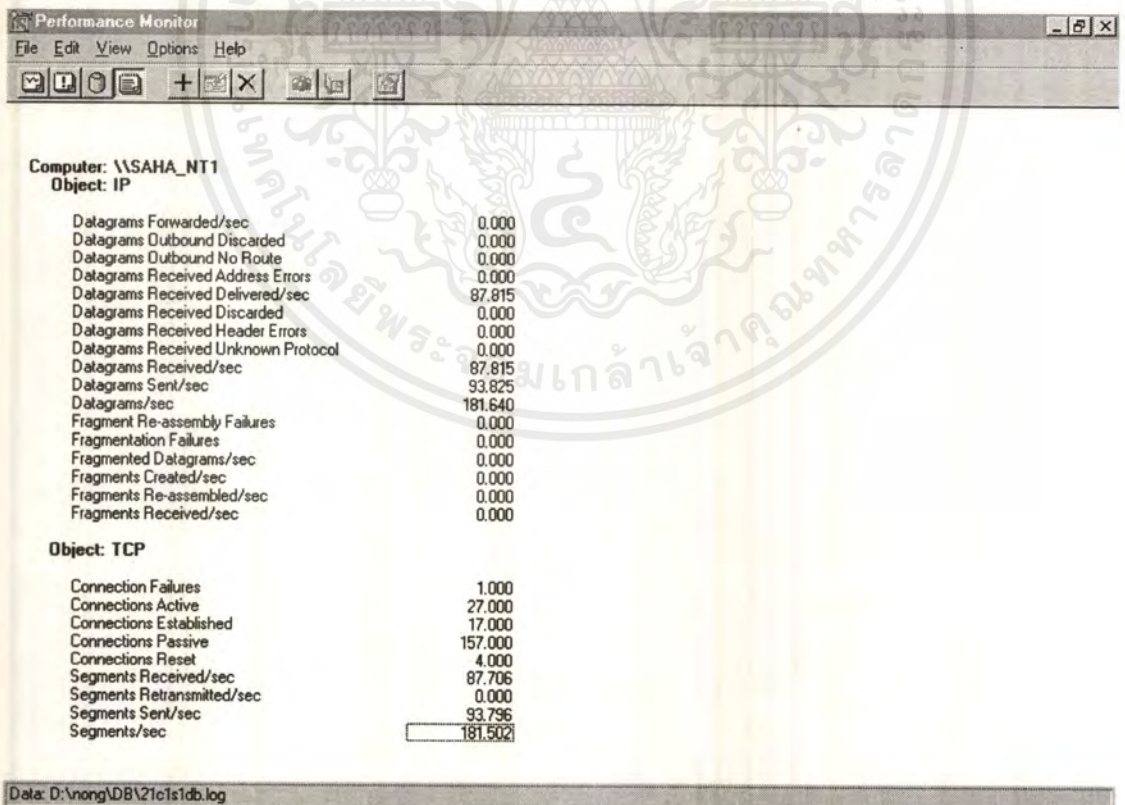
การตรวจจับคอขวดที่แคช

1. Copy Read Hits % และ Copy Reads/sec มีค่าค่อนข้างต่ำ
2. Data Flushes/sec มีค่า 1.815 ซึ่งหมายถึงแคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วงเพียง 1.815
3. ค่า Data Map Hits % มีค่า 97.98 % แต่ค่า Data Maps/sec มีค่าต่ำเพียง 3.948 อาจจะทำให้ประสิทธิภาพไม่ดีก็ได้ถ้าอัตราการฮิตสูง แต่อัตราการแมปต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-78 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที



รูปที่ 8-79 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำเคาน์เตอร์ออเดอร์โดยล็อกอินห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับคอขวดที่เครือข่าย

1. ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface)

เนื่องจากเครือข่ายที่ใช้ทดลองนี้เป็นเครือข่ายอีเทอร์เน็ต (Ethernet) ดังนั้นจึงมีอัตราการส่งเป็น 10 เมกะบิตต่อวินาที (10 Mbps.) และสามารถคิดเป็น %Utilization ของการส่งข้อมูลเป็นไบต์ จากค่าของ Bytes Total/sec ได้ดังนี้

83339.266 Bytes Total/sec = 83339.266 * 8 Bits/sec

เพราะฉะนั้น %Byte Utilization = $\frac{\text{จำนวน Bits/sec}}{\text{10Mbps.}} * 100$

$$= \frac{83339.266 * 8}{10 * 10^6} * 100 = 6.667\%$$

และสามารถหาจำนวนแพ็คเก็ตต่อการส่งใน 1 วินาทีของอีเทอร์เน็ตได้ดังนี้

จำนวนไบต์ต่อแพ็คเก็ต = 512 Bytes/packet = 512 * 8 Bits/packet

ดังนั้น $\text{Packets/sec} = \frac{10 \text{ Mbps}}{512 * 8 \text{ Bits/packet}} = \frac{10 * 10^6}{512 * 8}$ Packets/sec

เพราะฉะนั้น %Packet Utilization $= \frac{181.739 * 512 * 8}{10 * 10^6} * 100$

$= 7.444\ \%$

เมื่อพิจารณาจากค่า %Utilization ทั้ง 2 ค่าแล้วจะเห็นว่ามีค่าค่อนข้างต่ำ แสดงว่าการรับส่งข้อมูลมีค่าน้อยมากเมื่อเทียบกับค่าที่มันสามารถรองรับได้

2. ออบเจ็กต์ : ไอพี (Object : IP)

เมื่อดูจากค่า Datagrams Received Delivered/sec พบว่ามีค่าเท่ากับ Datagrams Received/sec แสดงว่ามีการส่งดาต้าแกรมที่ได้รับมา ออกไปสำเร็จทุกดาต้าแกรม

3.ออบเจ็กต์ : ทีซีพี (Object : TCP)

พิจารณาเปอร์เซ็นต์การติดต่อ (connection) ที่ไม่สำเร็จ โดยคิดเทียบกับจำนวนการติดต่อทั้งหมด (คือ ทั้งด้านรับและด้านส่ง) และจำนวนการติดต่อทั้งหมดคิดได้จาก

ค่า Connections Active (ด้านส่งข้อมูล) + ค่า Connections Passive (ด้านรับข้อมูล)

เพราะฉะนั้นเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จ $= \frac{\text{Connection Failures}}{\text{Connections Active + Connection Passive}} * 100$

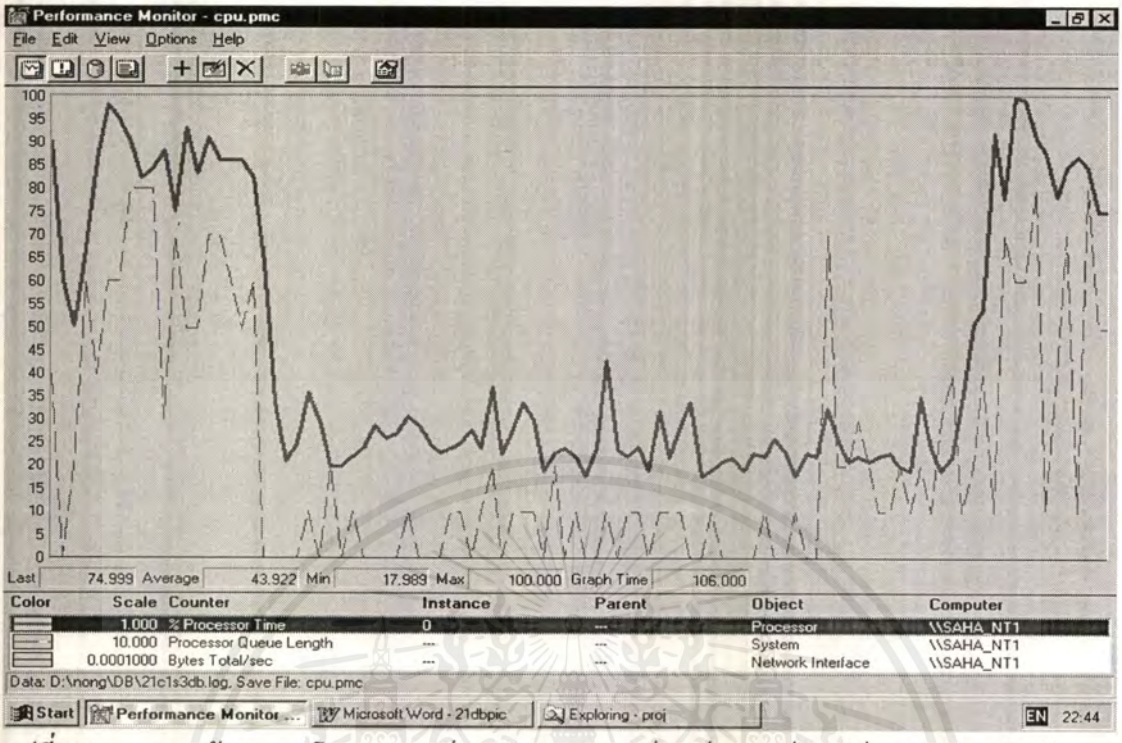
$$= \frac{1}{27+157} * 100 = 0.543\%$$

จะเห็นว่าเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จนั้นน้อยมาก แสดงว่าการรับส่งข้อมูลของชั้นทีซีพีนี้มีความผิดพลาดน้อยมาก จึงไม่เกิดปัญหาขึ้น

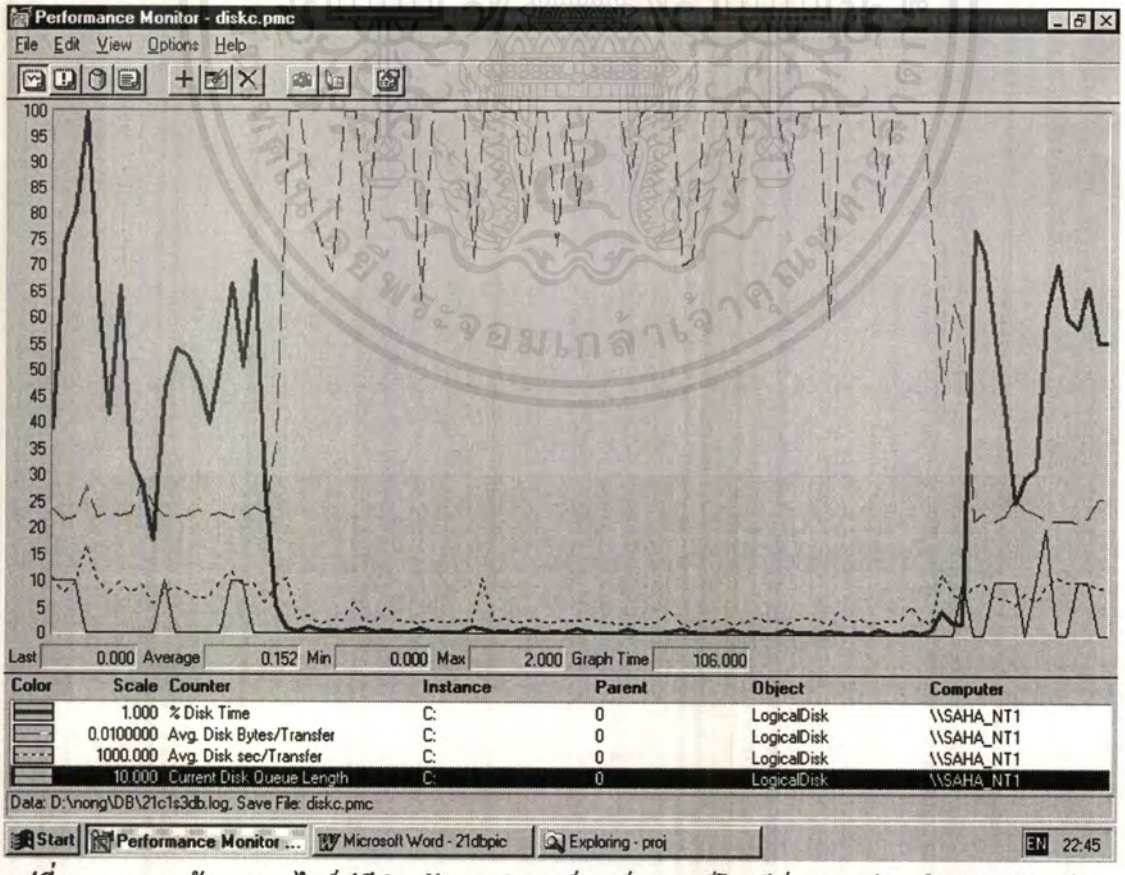
สรุป ไม่เกิดคอขวดที่เครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เคาน์เตอร์ออเดอร์ คีย์รายละเอียด ห่างกัน 1 วินาที

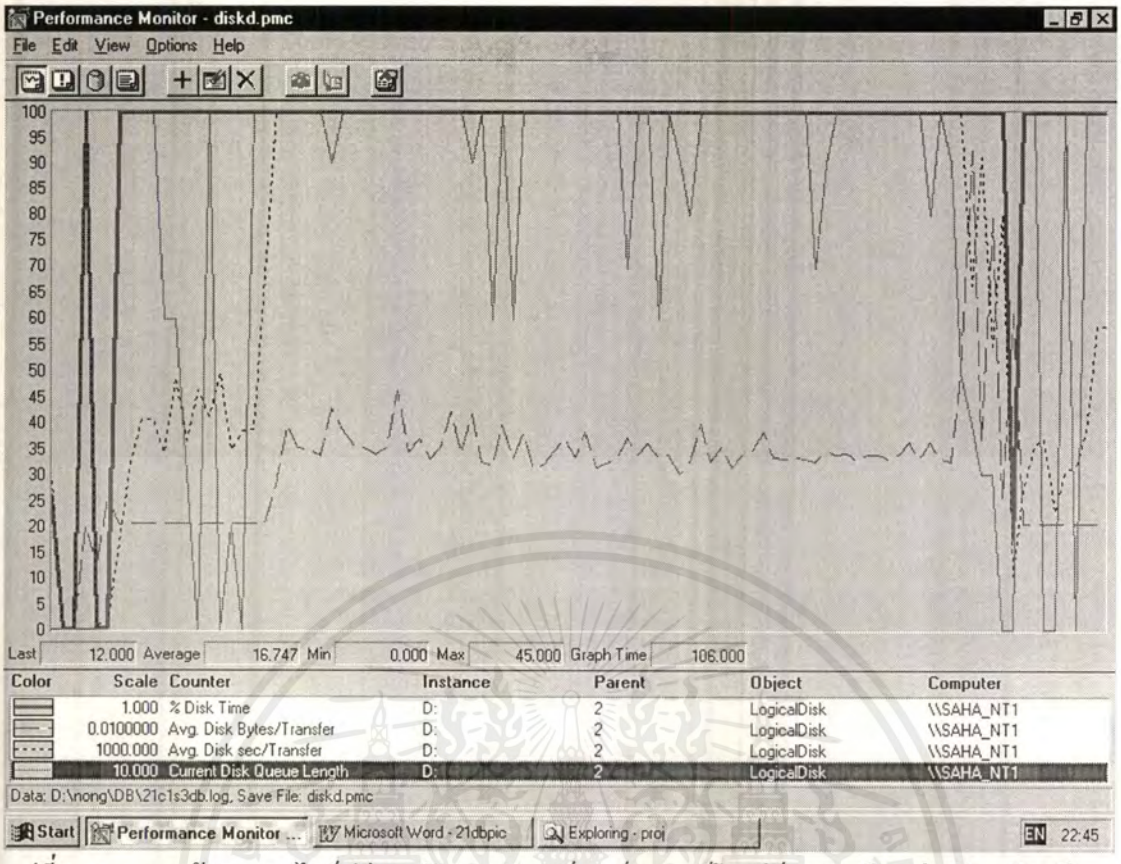


รูปที่8-80 แสดงข้อมูลของโพรเซสเซอร์สำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

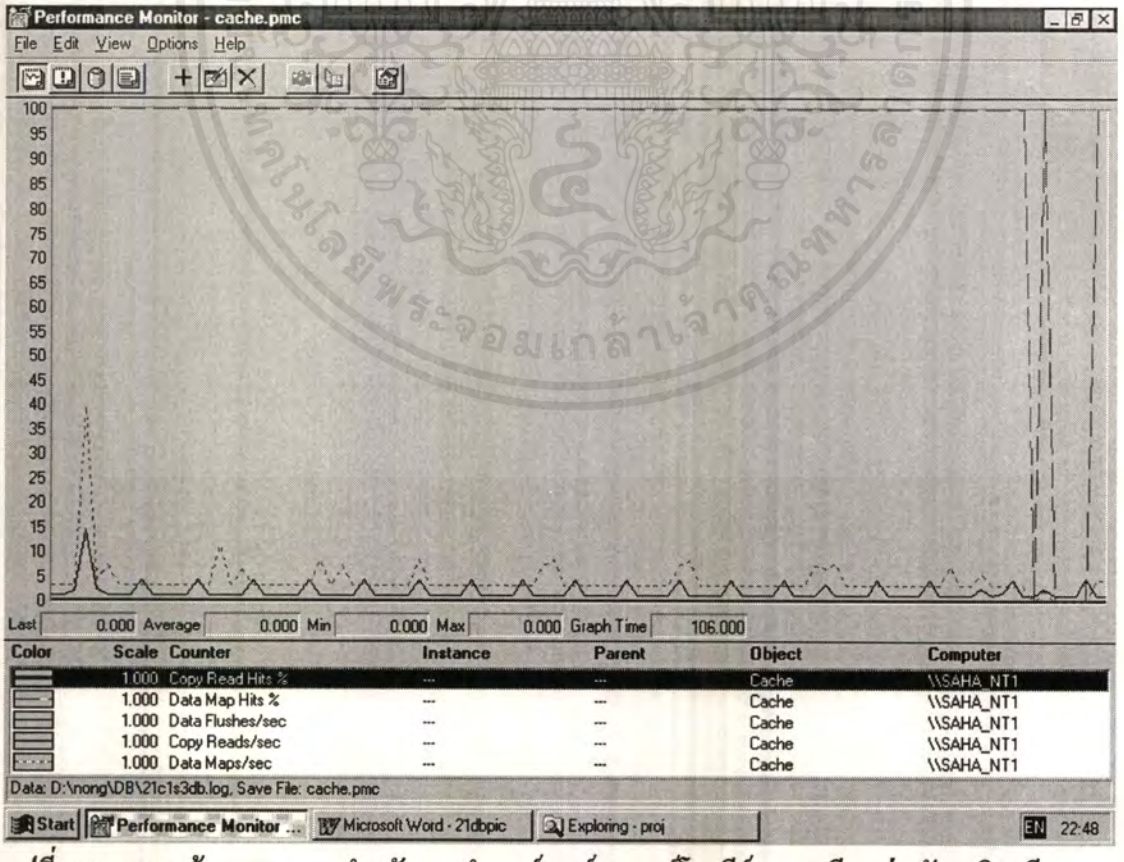


รูปที่ 8-81 แสดงข้อมูลของไดร์ฟซีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-82 แสดงข้อมูลของไดร์ฟดีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที



รูปที่ 8-83 แสดงข้อมูลของแคชสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจจับคอขวดที่โพรเซสเซอร์

1. %Processor Time มีการใช้โพรเซสเซอร์ประมาณ 43.628% จากกราฟจะพบว่ามีการใช้โพรเซสเซอร์มากในช่วงแรกและช่วงสุดท้ายของการล็อค จึงสรุปว่าเกิดจุดคอขวดที่โพรเซสเซอร์ ในช่วงแรกและช่วงสุดท้าย
2. เมื่อดูที่ Processor Queue Length พบว่ามีมีค่าเฉลี่ย 2.182 และมีค่าสูงที่สุด คือ 8 และจากกราฟพบว่ามีคิวเกิดขึ้นมากเมื่อมีการใช้โพรเซสเซอร์มาก สรุปได้ว่าเกิดคอขวดที่โพรเซสเซอร์

การตรวจจับคอขวดที่หน่วยความจำ

1. Pages/sec มีค่าเฉลี่ยเท่ากับ 0.03 เพจ แสดงว่ามีอัตราการเพจจิ้งน้อยมาก
2. Available Bytes มีค่าประมาณ 6 MB หมายถึงค่าของจำนวนหน่วยความจำเวอร์ชวลที่ยังสามารถใช้ได้ ซึ่งในที่นี้ ค่า available bytes มีค่าสูงกว่าเทรสโฮลที่กำหนดคือ 1MB
3. Commit Limit มีค่าประมาณ 400 MB ,Commit Limit คือ ขนาดของหน่วยความจำเวอร์ชวลที่สามารถคอมมิตได้ โดยไม่ต้องเพิ่มขนาดของ paging file ซึ่งค่า Commit Limit นี้มีค่าเท่ากับขนาดของ Paging file.sys รวมกับ ขนาดของแรมที่สามารถเปลี่ยนกลับไปยังดิสก์ได้ (หรือจำนวนของแรมที่ว่างที่เหลือจากการใช้เป็นระบบปฏิบัติการแล้ว)
4. Committed Bytes มีค่า 78 MB ซึ่งยังอยู่ในขอบเขตของ Commit Limit แสดงว่าเราไม่ต้องเพิ่มขนาดของ Paging file
5. ดูค่า Page Faults/sec ซึ่งมีค่า 14.515 เทียบกับ Transition Faults/sec ซึ่งมีค่า 0.279 ถ้าค่าทั้งสองนี้มีค่าใกล้เคียงกันแสดงว่าการเกิด page faults นั้นส่วนใหญ่นำเพจมาจาก transition page (เพจที่อยู่ในหน่วยความจำ แต่กำลังถูกเขียนลงบนดิสก์ในขณะเดียวกับที่เกิด page fault ขึ้น เพื่อที่จะเปลี่ยนแปลงแก้ไขข้อมูล)
6. Pool Nonpaged Bytes มีค่าประมาณ 2 MB ,ค่า Pool Nonpaged Bytes นี้ควรจะมีค่ามากกว่าขนาดของหน่วยความจำฟิสิคอลลบด้วยจำนวน 4 MB
7. ดูค่า Page Reads/sec เทียบกับค่า Disk Reads/sec ของไดร์ฟต่าง ๆ ซึ่งมีค่าไม่ใกล้เคียงกันเลย
8. ดูค่า Page Writes/sec เทียบกับค่า Disk Writes/sec ของไดร์ฟต่าง ๆ ซึ่งค่าที่อ่านได้มีค่าเท่ากับศูนย์ ซึ่งเท่ากับไดร์ฟอีและเอฟ
9. Page Inputs/sec * 4096 bytes/page = 4096 bytes/read เทียบกับค่า
   Page Reads/sec
   Avg. Disk Bytes/Read ซึ่งมีค่าไม่ใกล้เคียงกับไดร์ฟใดเลย
10. Page Outputs/sec * 4096 bytes/page = 0 bytes/write เทียบกับค่า
    Page Writes/sec
    Avg. Disk Bytes/Write ซึ่งเท่ากับไดร์ฟอีและเอฟ

สรุปไม่เกิดคอขวดที่หน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

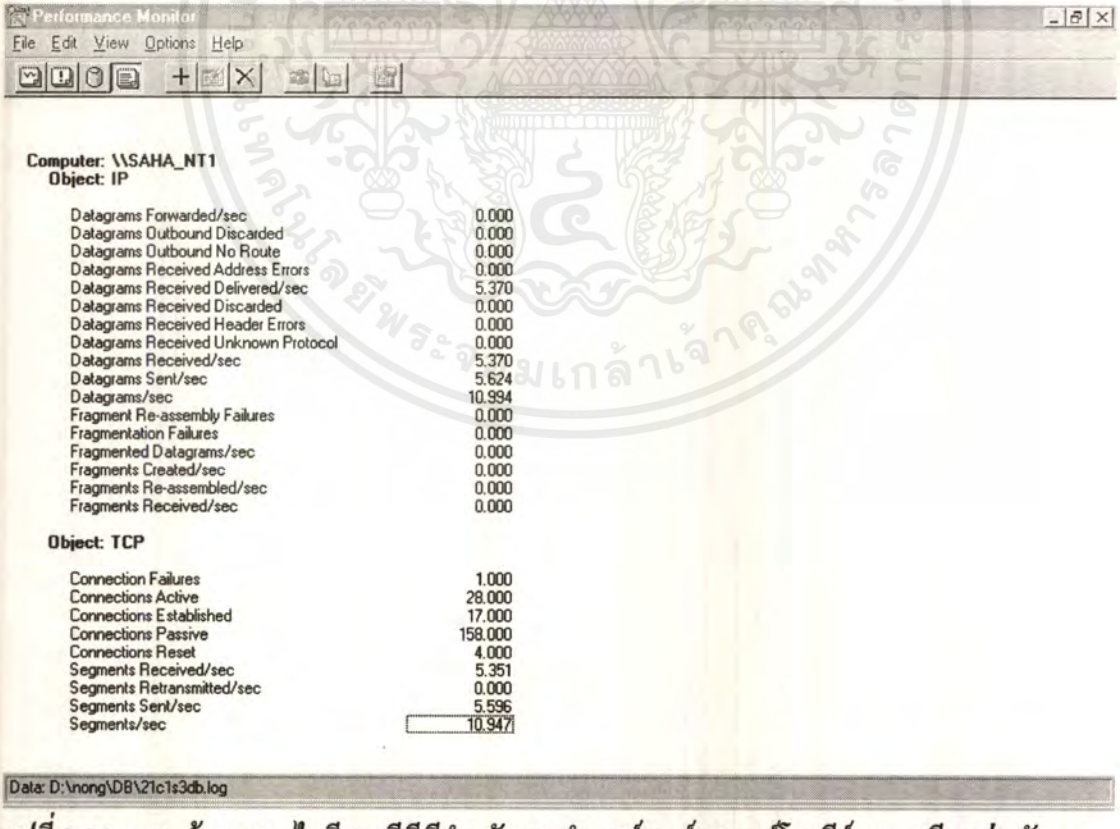
การตรวจจับคอขวดที่ดิสก์

1. ที่ไดร์ฟซี มีการใช้ดิสก์ไดร์ฟซีน้อยมาก คือมี %Disk Time เพียง 18.353% ส่วนคิวของดิสก์นั้นมีค่าเฉลี่ยคือ 0.162 เท่านั้น จึงทำให้สรุปได้ว่าดิสก์ไดร์ฟซีไม่เป็นคอขวด
2. ที่ไดร์ฟดีนั้น มีการใช้ดิสก์ไดร์ฟดีมาก คือ 93.285% และจากกราฟ ปริมาณการใช้ไดร์ฟดีมีค่ามากที่สุด 100% เกือบตลอดการล็อค ส่วนคิวนั้นมีค่าเฉลี่ยถึง 17.354 และค่ามากที่สุดถึง 45 คิว ดังนั้นจึงสรุปได้ว่าเกิดคอขวดที่ดิสก์ไดร์ฟดี
3. ที่ไดร์อีและเอฟนั้นไม่มีการส่งข้อมูลจากดิสก์หรือไปยังดิสก์เลย

สรุปว่าไม่เกิดคอขวดที่ดิสก์ไดร์ฟใดเลย

การตรวจจับคอขวดที่แคช

1. Copy Read Hits % และ Copy Reads/sec มีค่าเป็นศูนย์
2. Data Flushes/sec มีค่า 1.769 ซึ่งหมายถึงแคชมีการนำข้อมูลออกโดยได้รับการร้องขอจากระบบไฟล์ให้นำข้อมูลออกไปยังอุปกรณ์ต่อพ่วงเพียง 1.769
3. ค่า Data Map Hits % มีค่า 94.949 % แต่ค่า Data Maps/sec มีค่าต่ำเพียง 3.905 อาจจะทำให้ประสิทธิภาพไม่ดีก็ได้ถ้าอัตราการฮิตสูง แต่อัตราการแมปต่ำ



รูปที่ 8-84 แสดงข้อมูลของไอพีและทีซีพีสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Computer: \\SAHA_NT1
Object: Network Interface

	1	2
Bytes Received/sec	0.000	1745.169
Bytes Sent/sec	0.000	1481.771
Bytes Total/sec	0.000	3226.941
Current Bandwidth	10000000.000	10000000.000
Output Queue Length	0.000	0.000
Packets Outbound Discarded	0.000	0.000
Packets Outbound Errors	0.000	0.000
Packets Received Discarded	0.000	0.000
Packets Received Errors	0.000	0.000
Packets Received Non-Unicast/sec	0.000	0.047
Packets Received Unicast/sec	0.000	5.351
Packets Received Unknown	0.000	0.000
Packets Received/sec	0.000	5.398
Packets Sent Non-Unicast/sec	0.000	0.000
Packets Sent Unicast/sec	0.000	5.633
Packets Sent/sec	0.000	5.633
Packets/sec	0.000	11.031

Data: D:\nong\DB\21c1s3db.log

รูปที่ 8-85 แสดงข้อมูลของเน็ตเวิร์กอินเตอร์เฟสสำหรับการทำเคาน์เตอร์ออเดอร์โดยคีย์รายละเอียดห่างกัน 1 วินาที

การตรวจจับคอขวดที่เครือข่าย

1. ออบเจ็กต์ : เน็ตเวิร์กอินเตอร์เฟส (Object : Network Interface)

เนื่องจากเครือข่ายที่ใช้ทดลองนี้เป็นเครือข่ายอีเทอร์เน็ต (Ethernet) ดังนั้นจึงมีอัตราการส่งเป็น 10 เมกะบิตต่อวินาที (10 Mbps.) และสามารถคิดเป็น %Utilization ของการส่งข้อมูลเป็นไบต์ จากค่าของ Bytes Total/sec ได้ดังนี้

3226.941 Bytes Total/sec = 3226.941 * 8 Bits/sec

เพราะฉะนั้น $\%\text{Byte Utilization} = \frac{\text{จำนวน Bits/sec} * 100}{10\text{Mbps.}}$

$$= \frac{3226.941 * 8 * 100}{10 * 10^6} = 0.258\%$$

และสามารถหาจำนวนแพ็คเก็ตต่อการส่งใน 1 วินาทีของอีเทอร์เน็ตได้ดังนี้

จำนวนไบต์ต่อแพ็คเก็ต = 512 Bytes/packet = 512 * 8 Bits/packet

ดังนั้น $\text{Packets/sec} = \frac{10\text{ Mbps}}{512 * 8\text{ Bits/packet}} = \frac{10 * 10^6}{512 * 8}$ Packets/sec

เพราะฉะนั้น $\%\text{Packet Utilization} = \frac{11.031 * 512 * 8}{10 * 10^6} * 100$

$= 0.452\ \%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาจากค่า %Utilization ทั้ง 2 ค่าแล้วจะเห็นว่ามีค่าค่อนข้างต่ำ แสดงว่าการรับส่งข้อมูลมีค่าน้อยมากเมื่อเทียบกับค่าที่มันสามารถรองรับได้

2. ออบเจ็กต์ : ไอพี (Object : IP)

เมื่อดูจากค่า Datagrams Received Delivered/sec พบว่ามีค่าเท่ากับ Datagrams Received/sec แสดงว่ามีการส่งดาต้าแกรมที่ได้รับมา ออกไปสำเร็จทุกดาต้าแกรม

3.ออบเจ็กต์ : ทีซีพี (Object : TCP)

พิจารณาเปอร์เซ็นต์การติดต่อ (connection) ที่ไม่สำเร็จ โดยคิดเทียบกับจำนวนการติดต่อทั้งหมด (คือ ทั้งด้านรับและด้านส่ง) และจำนวนการติดต่อทั้งหมดคิดได้จาก

ค่า Connections Active (ด้านส่งข้อมูล) + ค่า Connections Passive (ด้านรับข้อมูล)

เพราะฉะนั้นเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จ $= \frac{\text{Connection Failures}}{\text{Connections Active + Connection Passive}} * 100$

$= \frac{1}{28+158} * 100 = 0.538\%$

จะเห็นว่าเปอร์เซ็นต์การติดต่อที่ไม่สำเร็จนั้นน้อยมาก แสดงว่าการรับส่งข้อมูลของชั้นทีซีพีนี้มีความผิดพลาดน้อยมาก จึงไม่เกิดปัญหาขึ้น

สรุป ไม่เกิดคอขวดที่เครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
บทวิจารณ์และสรุป

9.1 วิเคราะห์โครงงานเทียบกับทฤษฎีที่ได้ศึกษามา

1. การเก็บบันทึกผลข้อมูลลงล็อกไฟล์ เพื่อนำไปทำการวิเคราะห์ในภายหลังทำให้ได้ผลการวิเคราะห์ที่เที่ยงตรงแม่นยำมากยิ่งขึ้น

2. การวัดประสิทธิภาพขององค์ประกอบต่าง ๆ บนวินโดวส์เอ็นทีเซิร์ฟเวอร์ควรที่จะทำการตรวจจับตลอดเวลาเนื่องจากแต่ละเวลาบนเครื่องเซิร์ฟเวอร์จะรับโหลดไม่เท่ากันเมื่อรับโหลดมาก ๆ อาจทำให้เกิดคอขวด

3. ในการวัดแต่ละองค์ประกอบควรทราบว่าแต่ละองค์ประกอบจะต้องพิจารณาที่เคาน์เตอร์ตัวใดบ้างเป็นสำคัญเพื่อให้ผลการวิเคราะห์ที่ถูกต้องยิ่งขึ้น

4. การวิเคราะห์นั้นอาจจะไม่ขึ้นกับค่าที่กำหนดให้สำหรับแต่ละเคาน์เตอร์ตัวหนึ่งตัวใดโดยเฉพาะแต่อาจจะต้องพิจารณาประกอบกับเคาน์เตอร์ตัวอื่น ๆ ด้วย

5. เคาน์เตอร์บางตัวที่ใช้ในการวิเคราะห์ไม่มีเกณฑ์ที่แน่นอนลงไปว่ามีค่าตั้งแต่ช่วงไหนเป็นต้นไปจึงจะเกิดคอขวดจึงทำให้ยากแก่การวิเคราะห์

6. การปรับปรุงประสิทธิภาพบนวินโดวส์เอ็นทีนั้น ควรจะต้องมีผลการวิเคราะห์ที่ถูกต้องแม่นยำสามารถระบุได้ว่าองค์ประกอบใดบ้างที่เกิดคอขวดและทราบว่าควรที่จะปรับปรุงอย่างไร เพื่อที่จะปรับปรุงประสิทธิภาพได้ดีขึ้นและประหยัดทั้งเวลาและทรัพยากร

9.2 แนวทางในการพัฒนาต่อไป

โครงงานปรับปรุงประสิทธิภาพบนวินโดวส์เอ็นทีเซิร์ฟเวอร์ สามารถที่จะพัฒนาขีดความสามารถได้อีกดังนี้ คือ ทำการปรับปรุงประสิทธิภาพของออราเคิล (Oracle) เนื่องจากปัจจุบันนี้การใช้งานขององค์กรต่าง ๆ มักจะเป็นงานที่เกี่ยวข้องกับฐานข้อมูลขนาดใหญ่ จึงควรที่จะทำการปรับปรุงประสิทธิภาพของโปรแกรมที่เกี่ยวกับฐานข้อมูลที่วิ่งอยู่บนวินโดวส์เอ็นทีเซิร์ฟเวอร์

9.3 สรุปผลโครงงาน

โครงงานปรับปรุงประสิทธิภาพบนวินโดวส์เอ็นทีเซิร์ฟเวอร์นี้สามารถปรับปรุงองค์ประกอบต่าง ๆ ในระบบให้ทำงานได้มีประสิทธิภาพมากยิ่งขึ้น คือ สามารถเพิ่มความเร็วในการตอบสนองต่อการร้องขอต่างๆ จากเครื่องไคลเอ็นท์ได้ตามการปรับปรุงประสิทธิภาพขององค์ประกอบและจุดประสงค์ที่ตั้งไว้ และโปรแกรมที่ใช้ในการตรวจจับก็นับว่าเป็นโปรแกรมที่ถูกพัฒนาขึ้นมาอย่างดีเลิศแล้ว สามารถที่จะนำไปใช้ในการตรวจจับงานออราเคิลและงานอื่น ๆ ที่เกี่ยวข้องต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การใช้โปรแกรม Performance Monitor

Performance Monitor สามารถแสดงค่าเคาน์เตอร์ต่าง ๆ ได้ในหลายรูปแบบดังนี้

- ชาร์ต (Chart) แสดงเคาน์เตอร์หลาย ๆ ตัวได้ในเวลาเดียวกัน โดยแสดงออกมาในรูปกราฟหรือฮิสโทแกรม (histogram) โดยแกนนอนเป็นเวลา
- รายงาน (Report) สามารถแสดงค่าเคาน์เตอร์ทุก ๆ ตัวได้ใน 1 รายงาน
- การเตือน (Alert) สามารถตั้งค่าให้ทำการเตือนได้ ถ้าเคาน์เตอร์ตัวใดตัวหนึ่งมีค่าถึงที่กำหนดไว้
- การล็อค (Log) การล็อคนี้ ทำให้ข้อมูลต่าง ๆ ของเคาน์เตอร์ถูกบันทึกไว้บนดิสก์ เพื่อที่จะทำการวิเคราะห์ต่อไป โดย Performance Monitor นั้นสามารถแสดงชาร์ต, รายงาน และการเตือนจากข้อมูลในล็อคไฟล์ได้

การเลือกออบเจ็ก (Objects)

เมื่อเลือกคำสั่ง Add แล้ว จะเห็นรายการของออบเจ็กต่างๆ ที่เรียงตามลำดับตัวอักษรในเครื่องคอมพิวเตอร์ของ ซึ่งสามารถที่จะเลือกออบเจ็กต่าง ๆ เหล่านี้แล้วนำไปใช้ในการวัดได้ เครื่องคอมพิวเตอร์ที่ต่างกันก็จะมีออบเจ็กที่ต่าง ๆ กันไป ซึ่งขึ้นอยู่กับฮาร์ดแวร์และซอร์ฟแวร์ที่ทำการติดตั้ง

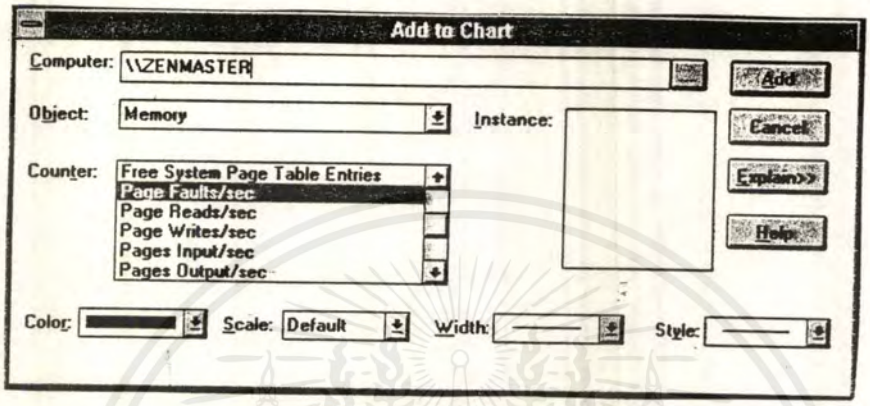
ในทุกๆระบบของ Windows NT จะมีออบเจ็กหลัก ๆ ดังต่อไปนี้เสมอ

ชื่อออบเจ็ก	คุณสมบัติ
Cache	เป็นระบบไฟล์แบบใช้แคชเป็นบัฟเฟอร์เก็บข้อมูล
Logical Disk	เป็นดิสก์ที่ถูกทำพาร์ติชั่น
Memory	เป็นหน่วยความจำแบบ Random-Access ที่ใช้ในการเก็บโค้ดและข้อมูล
Objects	เป็นออบเจ็กของซอร์ฟแวร์ต่างๆในระบบ
Paging File	เป็นไฟล์ที่ใช้ back up ข้อมูลขณะที่ทำหน่วยความจำเวอร์ชวล
Physical Disk	เป็นฮาร์ดแวร์ดิสก์
Process	เป็นออบเจ็กของซอร์ฟแวร์ต่าง ๆ ในระบบที่แสดงให้เห็นว่ากำลังทำการวิ่งโปรแกรมอยู่
Processor	เป็นฮาร์ดแวร์ที่ทำหน้าที่เอ็กซีคิวคำสั่งของโปรแกรม
System	เป็นเคาน์เตอร์ที่ประยุกต์มาใช้สำหรับทุก ๆ ระบบฮาร์ดแวร์และซอร์ฟแวร์
Thread	เป็นออบเจ็กของซอร์ฟแวร์ต่าง ๆ ในระบบที่อยู่ในกระบวนการที่กำลังใช้โปรเซสเซอร์อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกเคาน์เตอร์ (Counters)

ถ้าอยู่ในมุมมองของชาร์ต, รายงาน และการเตือน ก็สามารถที่จะเลือกเคาน์เตอร์ที่จะวัดได้จาก Countr box ในแต่ละออบเจ็ก ถ้าเคาน์เตอร์ตัวใดมีความสำคัญมากที่สุด จะถูกตั้งให้เป็นดีฟอลต์ (default) และเคาน์เตอร์จะถูกเรียงไว้ตามลำดับตัวอักษร สามารถเลือกเคาน์เตอร์ ได้มากกว่า 1 ตัวภายใน 1 ครั้ง โดยการกดคีย์ CTRL ค้างไว้ แล้วคลิกชื่อเคาน์เตอร์ แล้วกดปุ่ม Add



รูปที่ ก-1 การเลือกเคาน์เตอร์สำหรับการวัด

สามารถเลือกทุก ๆ เคาน์เตอร์สำหรับออบเจ็กหนึ่ง ๆ ได้ในมุมมองของรายงานโดยการใช้เมาส์ลากชื่อเคาน์เตอร์ แล้วกดคีย์ Home แล้วตามด้วย SHIFT+END

การเลือกอินสแตนซ์ (Instances)

มีออบเจ็กต่าง ๆ มากมายที่มีอินสแตนซ์ซึ่งเป็นอินสแตนซ์ของแต่ละชนิดของออบเจ็ก ยกตัวอย่าง เช่นถ้าในระบบของคุณมีดิสก์มากกว่า 1 พาร์ติชั่น คุณก็จะมี 1 อินสแตนซ์ สำหรับแต่ละพาร์ติชั่น ของดิสก์ หรือค่าเฉลี่ยของ Processor : % Processor Time สำหรับแต่ละอินสแตนซ์ของโปรเซสเซอร์ บนระบบที่มีหลายโปรเซสเซอร์

ในแต่ละ อินสแตนซ์จะมีพาเร้นท์ (parent) ของอินสแตนซ์ เรียกว่า *parent instance* เช่น ลอจิคอลดิสก์อินสแตนซ์ จะมีพาเร้นท์อินสแตนซ์ ของมันคือฟิสิคอลดิสก์ ซึ่งสามารถแสดงอินสแตนซ์กับพาเร้นท์ของมันโดยใช้สัญลักษณ์

parent instance==>child instance

ใน Instance box ของ Add to dialog box อินสแตนซ์จะถูกตั้งให้เป็น 1 อินสแตนซ์ เพราะเราไม่สามารถเดาได้ว่าออบเจ็กหรือเคาน์เตอร์ตัวไหนที่คุณสนใจ คุณอาจจะต้องการเลือกอินสแตนซ์ที่แตกต่างกันออกไปหรือหลาย ๆ อินสแตนซ์ เช่น ถ้าคุณต้องการเลือกหลาย ๆ อินสแตนซ์ คุณสามารถเลือกได้โดยกดคีย์ SHIFT หรือ CTRL และคลิกอินสแตนซ์อย่างต่อเนื่องหรือไม่ต่อเนื่องก็ได้ตามอินสแตนซ์ที่คุณต้องการจะเลือก

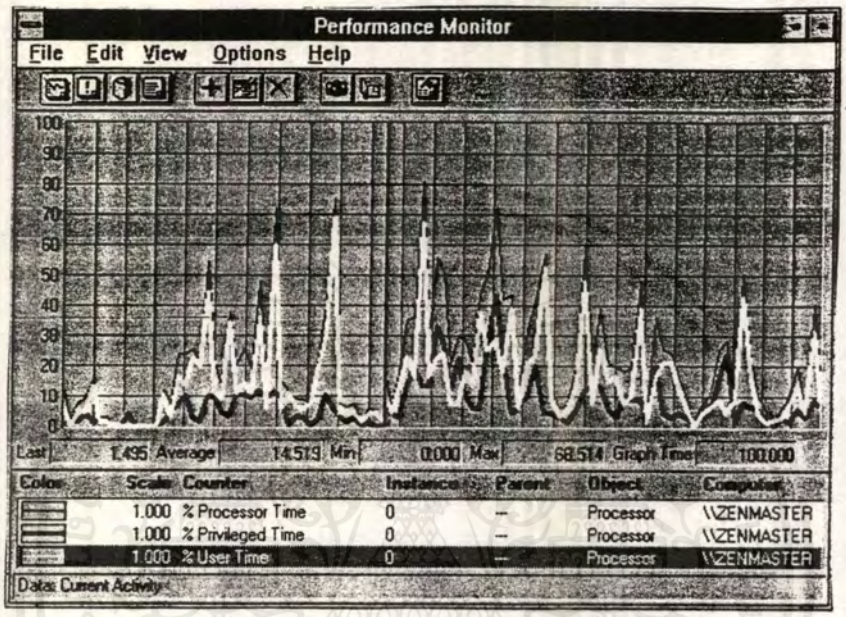
Performance Monitor มีมุมมอง 4 มุมมองดังนี้

1. มุมมองของชาร์ต (Chart View)

สามารถดูเคาน์เตอร์ของระบบซึ่งตอบสนองแบบ real time ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเลือกออปชั่น (option) ของชาร์ตที่จะปรากฏบนจอภาพนั้น สามารถใช้คำสั่ง Chart จากเมนู Options ได้ และจาก dialog box ของ Chart Options นั้นมีรูปแบบของชาร์ตที่สามารถเลือกได้ 2 รูปแบบ คือ กราฟ และฮิสโทแกรม โดยกราฟจะมีประโยชน์สำหรับดูค่าเคาน์เตอร์ตัวใดตัวหนึ่งในช่วงเวลาหนึ่งได้ ซึ่งฮิสโทแกรมจะไม่สามารถดูค่าเป็นช่วงเวลาได้ แต่ฮิสโทแกรมก็มีประโยชน์สำหรับดูค่าเคาน์เตอร์หลาย ๆ ตัว ณ เวลาหนึ่ง



รูปที่ ก-2 ชาร์ตของกิจกรรมปัจจุบัน (current activity) ในรูปแบบกราฟ

จากรูป Performance Monitor ชาร์ต จะแสดงระหว่าง 0 ถึง 100 data points สำหรับแต่ละเคาน์เตอร์ ที่แสดงอยู่ และมีค่าต่างที่แสดงบน value bar ดังนี้

Last คือ ค่าสุดท้ายของเคาน์เตอร์ที่มีแถบสว่างอยู่

Average คือ ค่าเฉลี่ยของเคาน์เตอร์ที่มีแถบสว่างอยู่

Min คือ ค่าที่น้อยที่สุดของเคาน์เตอร์ที่มีแถบสว่างอยู่

Max คือ ค่าที่มากที่สุดของเคาน์เตอร์ที่มีแถบสว่างอยู่

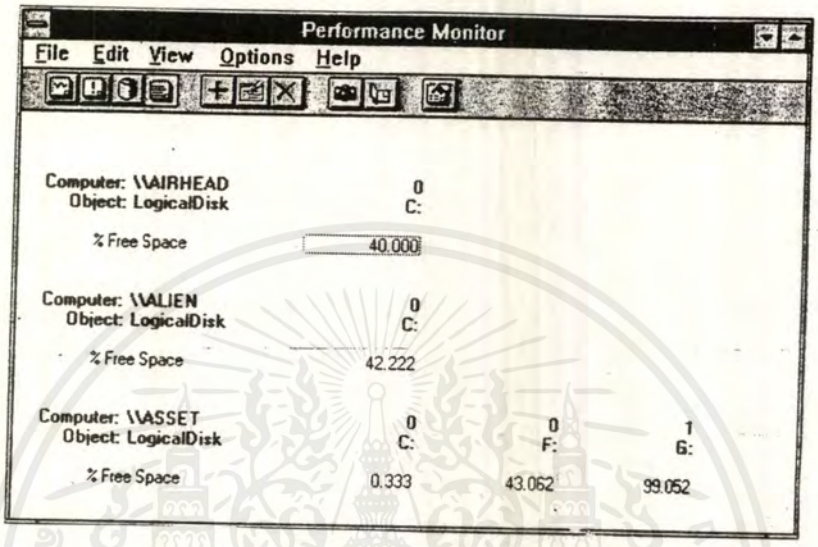
Graph Time คือ เวลาเป็นวินาทีที่ใช้ในการแสดงกราฟบนหน้าต่างโดย 1 หน้าจอจะแสดงได้ 100 data points เพราะฉะนั้นเวลาล็อคไฟล์ก็นำค่า time interval คูณด้วย 100 จะได้เวลาที่ใช้ในการล็อคเพื่อที่จะแสดงชาร์ตได้ 1 หน้าจอพอดี

- ถ้าล็อคไฟล์นั้นมีน้อยกว่า 100 data points ก็จะเห็นทุก ๆ จุดบนกราฟ โดยที่เส้นกราฟจะไม่ถึงขอบทางด้านขวาของหน้าต่าง
- ถ้าล็อคไฟล์นั้นมีมากกว่า 100 data points เช่นล็อคได้ 1000 data points ทุก ๆ 10 จุดก็จะถูกแสดงบนกราฟ ถ้าต้องการเห็นทุกจุด ก็ให้เลือกคำสั่ง Time Window จากเมนู Edit หรืออาจใช้การ export ข้อมูลได้จากเมนู File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **มุมมองของรายงาน (Report View)**

รายงาน มีประโยชน์ในการดูค่าเคาน์เตอร์หลาย ๆ ตัวในเวลาเดียวกัน ซึ่งมีส่วนช่วยในการตัดสินใจว่าจะเลือกเคาน์เตอร์ตัวใดให้แสดงบนชาร์ต รูปต่อไปแสดงรายงานที่แสดงค่าเคาน์เตอร์จากเครื่องคอมพิวเตอร์หลาย ๆ เครื่อง



รูปที่ ก-3 รายงานแสดงเคาน์เตอร์จากคอมพิวเตอร์หลายเครื่อง

สำหรับรายงานเราสามารถเลือกเคาน์เตอร์หลาย ๆ ตัวได้ และเลือกอินสแตนซ์หลาย ๆ ตัวได้เช่นเดียวกันกับชาร์ต ส่วนออปชั่นของรายงานนั้นมีเพียงออปชั่นเดียว คือ time interval ซึ่งตั้งค่าไว้คือ 5 วินาที

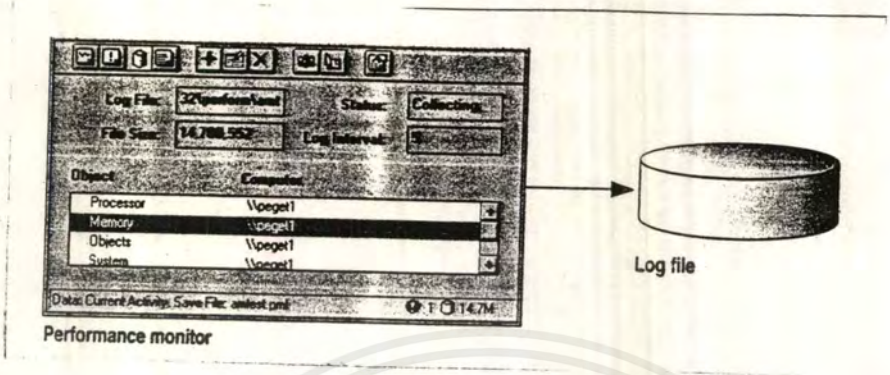
3. **มุมมองของการเตือน (Alert View)**

ใช้ในการดูเคาน์เตอร์หลาย ๆ ตัว แต่มี overhead ต่ำ และมีประโยชน์สำหรับดูคอมพิวเตอร์หลายๆเครื่องบนเครือข่ายโดยสามารถเพิ่มเคาน์เตอร์ ในมุมมองของการเตือนได้เหมือนกับในชาร์ตและในรายงาน แต่มุมมองของการเตือนจะมีแอททริบิว (attributes) บางอย่างที่เป็นเอกลักษณ์ คือ สำหรับแต่ละเคาน์เตอร์ที่เราต้องการให้มีการเตือน จะต้องมีการใส่ค่าจุดเริ่มต้นซึ่งเคาน์เตอร์ส่วนใหญ่ จะต้องการให้เตือนเมื่อมีค่ามากเกินค่าที่ตั้งไว้ และมีส่วนน้อยที่ต้องการให้เตือนเมื่อมีค่าน้อยกว่าที่ตั้งไว้ แต่ในโครงงานนี้ไม่ได้ใช้มุมมองของการเตือน จึงไม่ขอกล่าวในรายละเอียด

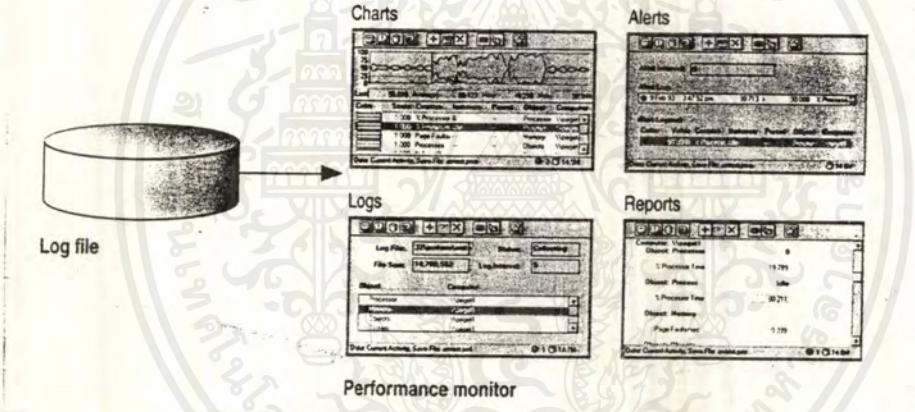
4. **มุมมองของการล็อก (Log View)**

ใช้เมื่อต้องการดูประสิทธิภาพของแอพพลิเคชั่น โดยการล็อคข้อมูลไว้ การล็อคนี้ทำให้สามารถดูข้อมูลเวลาใดก็ได้ และสามารถวิเคราะห์ข้อมูลได้อย่างถี่ถ้วน มากกว่าการดูข้อมูลแบบ real time บนจอภาพในรูปที่ ก-4, ก-5 แสดงการทำงานของการล็อค คือ ใช้คำสั่งให้ Performance Monitor ทำการเขียนข้อมูลลงล็อคไฟล์ เมื่อเขียนข้อมูลลงล็อคไฟล์เรียบร้อยแล้ว ก็สามารถอ่านข้อมูลที่ล้อคนั้น จากล็อคไฟล์ได้อีก ซึ่งสามารถแสดงข้อมูลในรูปชาร์ต, รายงาน, การเตือน หรือแม้กระทั่งล็อคใหม่จากล็อคไฟล์เดิมอีกก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



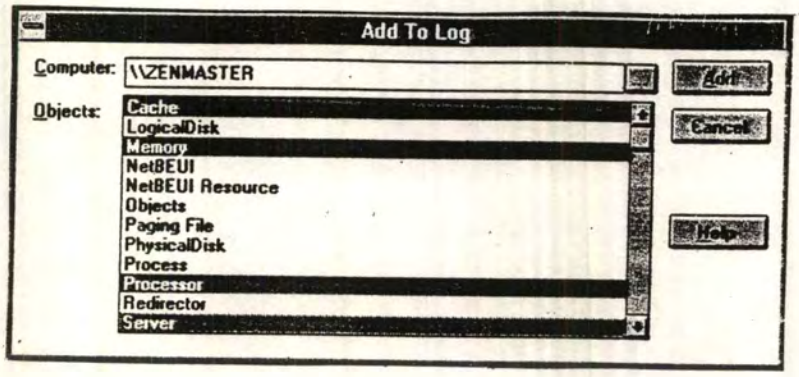
รูปที่ ก-4 การสร้างล็อคไฟล์



รูปที่ ก-5 การวิเคราะห์ข้อมูลจากล็อคไฟล์

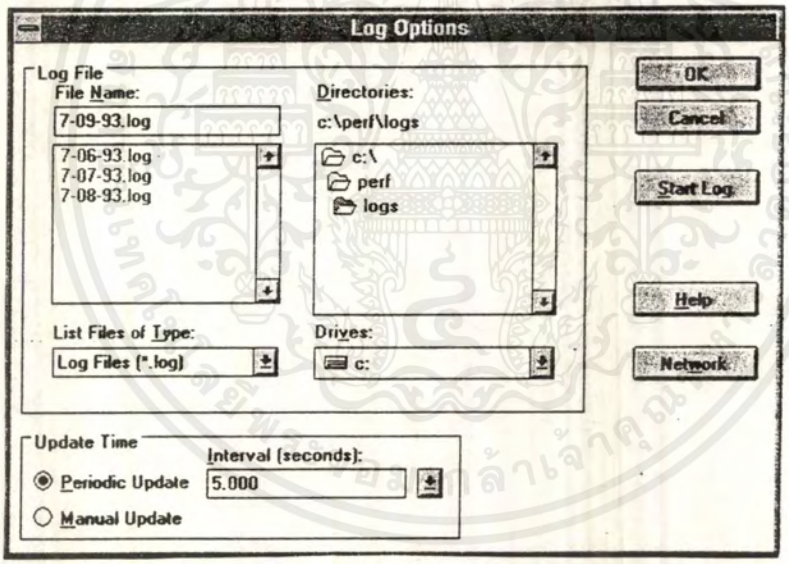
ในการสร้างล็อคไฟล์ให้เลือกไปที่มุมมองของล็อคแล้วเลือกคำสั่ง add to log จากเมนู edit แล้วเลือกออบเจ็กที่จะทำการล็อค ใน add to log dialog box นั้ นสามารถล็อคข้อมูลจากเครื่องคอมพิวเตอร์หลาย ๆ เครื่องลงในล็อคไฟล์ ไฟล์เดียวกันได้เพียงใส่ชื่อของเครื่องคอมพิวเตอร์ที่ต้องการจะดูเท่านั้น โดยเลือกชื่อจาก Select Computer Dialog Box และสามารถเลือกล็อคออบเจ็ก 1 ออบเจ็ก หรือมากกว่าก็ได้ โดยใช้คีย์ SHIFT และ CTRL เหมือนกับการเลือกเคาน์เตอร์ หรืออินสแตนซ์ และทุก ๆ เคาน์เตอร์สำหรับทุก ๆ อินสแตนซ์ของออบเจ็ก ที่ถูกเลือกก็จะถูกล็อคด้วยทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-6 Add To Log dialog box

เมื่อเลือกคอมพิวเตอร์และออบเจ็กแล้ว สามารถเลือกคำสั่ง log จากเมนู Options ได้ ซึ่งเมื่อเลือกแล้วจะปรากฏ Log Options dialog box ขึ้นมา ซึ่งสามารถกำหนดชื่อของล็อคไฟล์ได้ โดยสามารถใช้ปุ่ม Network ได้ ถ้าต้องการให้ ล็อคไฟล์ลง remote computer



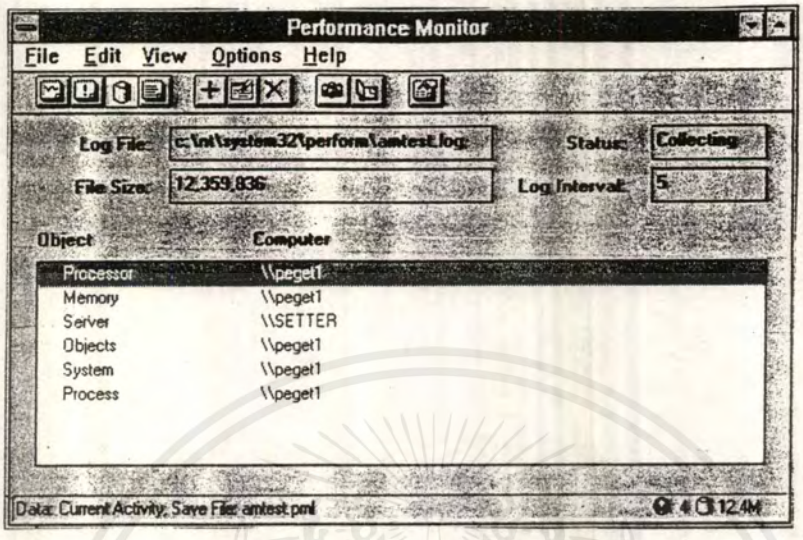
รูปที่ ก-7 Log Options dialog box

สามารถใช้ Log Options dialog box ในการเปลี่ยนค่า time interval ได้ ซึ่งค่าที่ตั้งไว้คือ 15 วินาที ซึ่งสามารถไปเลือกที่โหมด Manual Update ได้ แต่ในกรณีนี้จะไม่มีข้อมูลอะไรถูกล็อคไว้เลย ถ้าไม่ใช้คำสั่ง Update Now จกเมนู Options หรือใช้ไอคอน (icon) ที่เป็นรูปกล้องถ่ายรูปก็ได้ และต้องคลิกที่ปุ่ม Start Log เพื่อเริ่มต้นการล็อคด้วย ถ้าทุกอย่างได้ถูกตั้งค่าไว้อย่างถูกต้องสถานะของมุมมองของล็อคก็จะเป็น collecting ถ้าเพียงแต่คลิกปุ่ม OK สถานะก็จะเป็น Closed ซึ่งต้องทำการล็อคโดย

- เลือกออบเจ็กอย่างน้อย 1 ออบเจ็ก ในการล็อค
- ใส่ชื่อล็อคไฟล์
- คลิกปุ่ม Start Log

เอกสารนี้จัดทำขึ้นเพื่อไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการหยุดล็อค ให้เลือกที่ปุ่ม Stop Log ใน Log Options dialog box

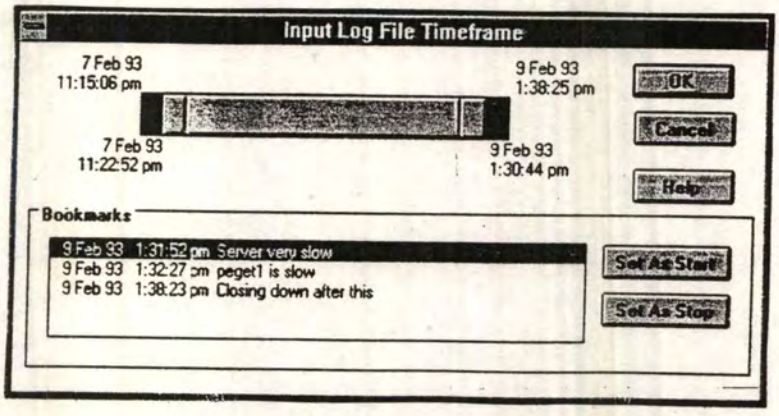


รูปที่ ก-8 มุมมองของล็อคในระหว่างการเก็บข้อมูล

การ Load และการดู ล็อคไฟล์

ในการดู ล็อคไฟล์ นั้นให้ใช้คำสั่ง Data From ในเมนู Options แล้วเลือก ล็อคไฟล์ และพิมพ์ชื่อล็อคไฟล์ หรือเลือกชื่อ ล็อคไฟล์ จากปุ่มที่มีลูกศรชี้ลงจาก Option ล็อคไฟล์ dialog box

ในการดูข้อมูลจากล็อคไฟล์นั้น เหมือนกับการดูกิจกรรมปัจจุบัน คือ สามารถดู ชาร์ต, รายงาน, การเตือน หรือทำการล็อคไฟล์อีกครั้งหนึ่ง โดยสามารถเลือกออบเจ็ก, เคาน์เตอร์ และ อินสแตนซ์ สำหรับการดู ชาร์ต เหมือนดูกิจกรรมปัจจุบันแต่เวลาในการแสดงชาร์ต นั้นจะแตกต่างกันออกไป Performance Monitor นั้น พยายามที่จะแสดงชาร์ต 100 จุด เพื่อให้เต็ม 1 หน้าต่างพอดี ถ้าข้อมูลในไฟล์นั้นมีน้อยกว่า 100 snapshots ก็จะเห็นทุก ๆ จุดบนกราฟ โดยที่เส้นกราฟจะไม่ถึงขอบทางด้านขวาของหน้าต่าง ถ้ามีมากกว่า 100 snapshot ในล็อคไฟล์ เช่นมี 1000 data points กราฟจะแสดงทุก ๆ 10 จุด ถ้าต้องการเห็นทุกจุดก็ให้เลือกคำสั่ง Time Window จากเมนู edit ดังรูป



รูปที่ ก-9 Input Log File Time Frame dialog box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการใช้ slide bar ใน dialog box เพื่อเปลี่ยน time window ที่แสดงบนชาร์ตซึ่งสามารถเปลี่ยนเวลาเริ่มต้นและเวลาสิ้นสุดของ time window โดยการปรับ slide bar เวลาที่แสดงอยู่เหนือ slide bar คือ เวลาเริ่มต้นและสิ้นสุดของทุกข้อมูลในล็อคไฟล์ ส่วนเวลาที่แสดงอยู่ใต้ slide bar คือ เวลาเริ่มต้นและสิ้นสุด ณ ปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

Windows NT Performance Counters

รายการ counter แต่ละตัว จะแสดงในรูปแบบต่อไปนี้

Counter Name *Complexity*

Explain text

Default Scale : Scale Factor

ตาราง ข.1 Performance Counter Reference Information

Field	Explanation
Counter Name	ชื่อ counter
Complexity	ระดับความยากในการใช้ counter โดยเริ่มจากระดับ Novice ซึ่งเป็นระดับง่ายที่สุด ไปจนถึงระดับ Wizard ซึ่งเป็นระดับของ Windows NT programmer ขั้นสูง
Explain text	คำอธิบาย counter
Scale Factor	ตัวเลขที่ Performance Monitor ใช้คูณ scale ในแนวตั้ง

Cache Object

Object : **Cache** *Advanced*

cache จะจัดการ memory สำหรับการเข้าถึงไฟล์แบบรวดเร็ว ไฟล์บน Windows NT จะถูกเก็บไว้ที่ cache memory ในหน่วยของ page Cache จะเก็บ page ของไฟล์ไว้ให้นานที่สุดเท่าที่จะนานได้เพื่ออนุญาตให้ file system เข้าถึงข้อมูลได้โดยไม่ต้องเข้าถึง disk

Async Copy Reads/sec *Wizard*

ความถี่ในการอ่าน page จาก Cache ซึ่งทำให้เกิดการคัดลอกข้อมูลจาก Cache ไปยัง buffer ของ application และ application ก็จะได้รับการควบคุมกลับมาทันที แม้ว่า disk จะต้องถูกเข้าถึงเพื่อที่จะนำ page กลับคืนมา

Default Scale : 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Async Data Maps/sec *Wizard*

ความถี่ที่ application ใช้ file system เช่น NTFS และ HPFS ในการ map page เข้าไปใน Cache เพื่อที่จะอ่าน page นั้น และไม่รอ Cache ในการนำ page นั้นกลับคืนมา ในกรณีที่ page นั้นไม่อยู่ใน main memory

Default Scale : 1

Async Fast Reads/sec *Wizard*

ความถี่ในการอ่าน page จาก Cache ซึ่งข้ามการเรียกใช้ file system ไป และนำข้อมูลมาจาก Cache โดยตรง ถ้าข้อมูลอยู่ใน Cache แต่ถ้าข้อมูลไม่อยู่ใน Cache การร้องขอจาก application ก็จะไม่รอจนกระทั่งได้ข้อมูลมาจาก disk แต่จะทำการควบคุมอย่างรวดเร็ว

Default Scale : 0.1

Async MDL Reads/sec *Wizard*

ความถี่ในการอ่าน page จาก Cache โดยใช้ Memory Descriptor List (MDL) ในการเข้าถึง page MDL จะประกอบด้วย physical address ในการ transfer โดยใช้ Direct Memory Access (DMA) ถ้า page นั้นไม่ได้อยู่ใน main memory การร้องขอของ application นั้นก็จะไม่รอ page ที่จะถูกอ่านเข้ามาจาก disk

Default Scale : 1

Async Pin Reads/sec *Wizard*

ความถี่ในการอ่านข้อมูลเข้ามาใน Cache เพื่อเตรียมที่จะเขียนข้อมูลนั้นกลับไปยัง disk page ที่ถูกอ่านในกรณีนี้จะถูกยึดไว้อยู่ที่ memory จนกระทั่งกระบวนการในการอ่านเสร็จสมบูรณ์ file system จะได้รับการควบคุมกลับมาทันที แม้ว่า disk จะถูกเข้าถึงเพื่อนำ page กลับคืนมาขณะที่ถูกยึดอยู่ใน memory นั้น physical address ของ page จะไม่เปลี่ยนแปลง

Default Scale : 1

Copy Read Hits % *Expert*

เปอร์เซ็นต์การร้องขอ Copy Read จาก Cache ซึ่งมีข้อมูลนั้นอยู่ใน Cache เรียกว่าเกิดการ hit Copy Read เป็น operation ในการอ่านไฟล์ ซึ่งคัดลอก page จาก Cache ไปยัง buffer ของ Application,LAN Redirector ใช้วิธีนี้ในการนำข้อมูลมาจาก Cacheเช่นเดียวกับ LAN Server ซึ่งใช้ในการ transfer ข้อมูลขนาดเล็ก นอกจากนี้disk file system ก็ใช้วิธีนี้ด้วยเช่นกัน

Default Scale : 1

Copy Reads/sec *Expert*

ความถี่ในการอ่าน page ของ Cache ซึ่งเป็นการคัดลอกข้อมูลจาก Cache ไปยัง buffer ของ application LAN Redirector ใช้วิธีนี้ในการนำข้อมูลมาจาก Cache เช่นเดียวกับ LAN Server ซึ่งใช้ในการ transfer ข้อมูลขนาดเล็ก นอกจากนี้ disk file system ก็ใช้วิธีนี้ด้วยเช่นกัน

Default Scale : 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Flush Pages/sec *Advanced*

จำนวน page ที่ Cache เขียนออกไปยัง disk ซึ่งในการเขียนแต่ละครั้งนั้น สามารถเขียนได้มากกว่า 1 page

Default Scale : 1

Data Flushes/sec *Wizard*

ความถี่ที่ Cache เขียนข้อมูลออกไปยัง disk ซึ่งในการเขียนแต่ละครั้งนั้น สามารถเขียนได้มากกว่า 1 page

Default Scale : 1

Data Map Hits % *Wizard*

เปอร์เซ็นต์ของ Data Map ใน Cache ที่ไม่ต้องนำ page มาจาก disk เพราะ page นั้นอยู่ใน physical memory อยู่แล้ว

Default Scale : 1

Data Map Pins/sec *Wizard*

ความถี่ของ Data Map ใน Cache ที่เป็นผลให้เกิดการยึด page ไว้ใน main memory การกระทำนี้เป็นการเตรียมการเขียนไปยัง disk ในขณะที่ page ถูกยึดไว้ physical address ของ page ใน main memory และ virtual address ใน Cache จะไม่เปลี่ยนแปลง

Default Scale : 1

Data Maps/sec *Expert*

ความถี่ที่ file system เช่น NTFS หรือ HPFS ทำการ map page ของไฟล์ไปยัง Cache เพื่อที่จะอ่าน page นั้น

Default Scale : 1

Fast Read Not Possibles/sec *Wizard*

ความถี่ที่ function call ของ application program interface (API) พยายามดึงข้อมูลจาก Cache โดยไม่ผ่าน file system แต่ผลสุดท้ายจะไม่ได้รับข้อมูล ถ้าไม่มีการร้องขอผ่าน file system

Default Scale : 1

Fast Read Resource Misses/sec *Wizard*

ความถี่ของการไม่พบข้อมูลใน Cache (Cache miss) ซึ่งเป็นผลจากการขาด resource ในการสนับสนุนการร้องขอ

Default Scale : 1

Fast Reads/sec *Wizard*

ความถี่ในการอ่าน page จาก Cache ซึ่งข้ามการเรียกใช้ file system ไป และนำข้อมูลมาจาก Cache โดยตรง ถ้าข้อมูลอยู่ใน Cache แต่ถ้าข้อมูลไม่อยู่ใน Cache ก็จะไม่มีการร้องขอผ่าน file system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default Scale : 0.1

Lazy Write Flushes/sec *Wizard*

ความถี่ที่ Lazy Write thread ของ Cache ทำการเขียนไปยัง disk การเขียนของ Lazy Write นี้เป็นขบวนการในการ update disk หลังจากที่ page ของข้อมูลได้ถูกเปลี่ยนแปลงใน memory ดังนั้น application ที่ทำให้ไฟล์เปลี่ยนแปลงนั้นไม่ต้องรอให้การเขียนไปยัง disk เสร็จสมบูรณ์ ก็สามารถดำเนินการต่อไปได้ สำหรับ operation การเขียนนั้นสามารถเขียนแต่ละครั้งได้มากกว่า 1 page

Default Scale : 1

Lazy Write Pages/sec *Advanced*

จำนวน page ที่ Lazy Write thread ของ Cache ทำการเขียนไปยัง disk การเขียนของ Lazy Write นี้เป็นขบวนการในการ update disk หลังจากที่ page ของข้อมูลได้ถูกเปลี่ยนแปลงใน memory ดังนั้น application ที่ทำให้ไฟล์เปลี่ยนแปลงนั้นไม่ต้องรอให้การเขียนไปยัง disk เสร็จสมบูรณ์ ก็สามารถดำเนินการต่อไปได้ สำหรับ operation การเขียนนั้นสามารถเขียนแต่ละครั้งได้มากกว่า 1 page

Default Scale : 1

MDL Read Hits % *Expert*

เปอร์เซ็นต์การร้องขอการอ่านของ Cache Memory Descriptor List (MDL) ที่ hit Cache

Default Scale : 1

MDL Reads/sec *Expert*

ความถี่ในการอ่าน page ของ Cache ที่ใช้ Memory Descriptor List (MDL) ในการเข้าถึงข้อมูล MDL ประกอบด้วย physical address ของแต่ละ page ที่ถูกอ้างถึงในการ transfer และสามารถใช้ hardware แบบ Direct Memory Access (DMA) ในการ copy LAN Server ก็ใช้วิธีนี้ในการ transfer ข้อมูลขนาดใหญ่ออกจาก server

Default Scale : 1

Pin Read Hits % *Expert*

เปอร์เซ็นต์ของการร้องขอ Cache Pin Read ที่เกิดการ hit Cache เมื่อ page ถูกยึดไว้ใน Cache physical address ของ page นั้นจะไม่เปลี่ยนแปลง LAN Redirector ใช้วิธีนี้ในการนำข้อมูลมาจาก Cache เช่นเดียวกับ LAN Server ซึ่งใช้ในการ transfer ข้อมูลขนาดเล็ก นอกจากนี้ disk file system ก็ใช้วิธีนี้ด้วยเช่นกัน

Default Scale : 1

Pin Reads/sec *Expert*

ความถี่ ของการอ่านข้อมูลเข้าไปใน Cache เพื่อเตรียมที่จะเขียนข้อมูลกลับไปยัง disk page ที่ถูกอ่านในกรณีนี้จะถูกยึดอยู่ที่ memory จนกระทั่งกระบวนการในการอ่านเสร็จสมบูรณ์ ขณะที่ถูกยึดอยู่ใน memory นั้น physical address ของ page จะไม่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default Scale : 1

Read Aheads/sec *Advanced*

ความถี่ในการอ่านของ Cache เมื่อ Cache ตรวจจับได้ว่ามีการ access ไฟล์แบบ sequential read aheads อนุญาตให้ทำการ transfer ข้อมูลเป็น block ขนาดใหญ่กว่า block ที่ถูกร้องขอโดย application เพื่อลด overhead ต่อ 1 access

Default Scale : 1

Sync Copy Reads/sec *Wizard*

ความถี่ในการอ่าน page จาก Cache ซึ่งทำให้เกิดการคัดลอกข้อมูลจาก Cache ไปยัง buffer ของ application file system จะไม่ได้รับการควบคุมกลับคืนมาจนกระทั่ง operation การ copy ได้เสร็จสมบูรณ์แล้ว แม้ว่า disk จะต้องถูกเข้าถึงเพื่อที่จะนำ page กลับคืนมา

Default Scale : 1

Sync Data Maps/sec *Wizard*

ความถี่ที่ file system เช่น NTFS หรือ HPFS ทำการ map page เข้าไปใน Cache เพื่อที่จะอ่าน page นั้น และรอ Cache ในการนำ page นั้นกลับคืนมาในกรณีที่ page นั้นไม่อยู่ใน main memory

Default Scale : 1

Sync Fast Reads/sec *Wizard*

ความถี่ในการอ่าน page จาก Cache ซึ่งข้ามการเรียกใช้ file system ไป และนำข้อมูลมาจาก Cache โดยตรง ถ้าข้อมูลอยู่ใน Cache แต่ถ้าข้อมูลไม่อยู่ใน Cache การร้องขอจาก application ก็จะรอจนกระทั่งได้ข้อมูลมาจาก disk

Default Scale : 0.1

Sync MDL Reads/sec *Wizard*

ความถี่ในการอ่าน page จาก Cache โดยใช้ Memory Descriptor List (MDL) ในการเข้าถึง page MDL จะประกอบด้วย physical address ในการ transfer โดยใช้ Direct Memory Access (DMA) ถ้า page นั้นไม่ได้อยู่ใน main memory การร้องขอของ application นั้นก็จะรอ page ที่จะถูกอ่านเข้ามาจาก disk

Default Scale : 1

Sync Pin Reads/sec *Wizard*

ความถี่ในการอ่านข้อมูลเข้ามาใน Cache เพื่อเตรียมที่จะเขียนข้อมูลนั้นกลับไปยัง disk page ที่ถูกอ่านในกรณีนี้จะถูกยึดไว้อยู่ที่ memory จนกระทั่งกระบวนการในการอ่านเสร็จสมบูรณ์ file system จะไม่ได้รับการควบคุม จนกระทั่ง page ถูกยึดไว้อยู่ใน Cache ถ้า disk ต้องถูกเข้าถึงเพื่อที่จะได้ page นั้นกลับคืนมา ใน ขณะที่ถูกยึดอยู่ใน Cache นั้น physical address ของ page ใน Cache นั้นจะไม่เปลี่ยนแปลง

Default Scale : 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LogicalDisk Object

Object : **LogicalDisk** *Novice*

A Logical Disk object type คือ การทำ partition บน hard disk หรือ fixed disk และเมื่อทำการแบ่ง disk เป็นส่วนๆแล้วก็จะทำการกำหนด disk ที่ถูกแบ่งแล้วเป็นตัวอักษร เช่น drive C, disk ที่สามารถทำเป็น partition จะมีส่วนต่างๆที่สามารถ store file, program และ page data, disk ถูกอ่านเพื่อดึงข้อมูลต่างๆ และถูกเขียนเพื่อบันทึกการเปลี่ยนแปลงสิ่งเหล่านั้น

% Disk Read Time *Novice*

Disk Read Time คือ เปอร์เซ็นต์เวลาที่สูญเสียไปของ disk drive ที่ถูกเลือกให้บริการการร้องขอการอ่าน

Default Scale : 1

% Disk Time *Novice*

Disk Time คือ เปอร์เซ็นต์เวลาที่สูญเสียไปของ disk drive ที่ถูกเลือกให้บริการการร้องขอการอ่านหรือการเขียน

Default Scale : 1

% Disk Write Time *Novice*

Disk Write Time คือ เปอร์เซ็นต์เวลาที่สูญเสียไปของ disk drive ที่ถูกเลือกการให้บริการการร้องขอการเขียน

Default Scale : 1

% Free Space *Novice*

Percent Free Space คือ อัตราส่วนของเนื้อที่ที่ยังมีค่าของ logical disk ที่ยังไม่ได้ใช้ต่อพื้นที่ทั้งหมดของ logical disk ที่ถูกเลือก

Default Scale : 1

Avg. Disk Bytes/Read *Expert*

Avg. Disk Bytes/Read คือ ค่าเฉลี่ยของจำนวน bytes ที่ transfer จาก disk ระหว่างทำการ read

Default Scale : 1

Avg. Disk Bytes/Transfer *Expert*

Avg. Disk Bytes/Transfer คือ ค่าเฉลี่ยของจำนวน bytes ที่ transfer ไปยังหรือจาก disk ระหว่างการเขียนหรือการอ่าน

Default Scale : 1

Avg. Disk Bytes/Write *Expert*

Avg. Disk Bytes/Write คือ ค่าเฉลี่ยของจำนวน bytes ที่ transfer ไปยัง disk ระหว่างทำการ write

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default Scale : 1

Avg. Disk sec/Read *Advanced*

Avg. Disk sec/Read คือ ค่าเฉลี่ยของเวลาในหน่วยวินาทีของการอ่านข้อมูลจาก disk

Default Scale : 1000

Avg. Disk sec/Transfer *Advanced*

Avg. Disk sec/Transfer คือ ค่าเฉลี่ยของเวลาในหน่วยวินาทีในการ transfer disk

Default Scale : 1

Avg. Disk sec/Write *Advanced*

Avg. Disk sec/Write คือ ค่าเฉลี่ยของเวลาในหน่วยวินาทีของการเขียนข้อมูลไปยัง disk

Default Scale : 1

Disk Bytes/sec *Advanced*

Disk Bytes/sec คือ จำนวนอัตรา bytes ที่ถูก transfer ไปยังหรือจาก disk ระหว่างการเขียนหรือการอ่าน

Default Scale : 0.0001

Disk Queue Length *Novice*

Disk Queue Length คือ จำนวนของการ request ที่ยังต้องทำบน disk ขณะเวลาที่การกระทำ data ถูกสะสม, มันรวมไปถึงการ request การบริการขณะเวลาของ snapshot Disk Queue Length เป็นความยาวของคิวที่เกิดขึ้นในช่วงเวลาสั้นๆ ไม่ได้เป็นค่าเฉลี่ยของทั้งหมดของ time interval

Default Scale : 10

Disk Read Bytes/sec *Advanced*

Disk Read Bytes/sec คือ อัตราของ bytes ที่ถูก transfer จาก disk ระหว่างการอ่าน

Default Scale : 0.0001

Disk Reads/sec *Novice*

Disk Read/sec คือ อัตราของการอ่านบน disk

Default Scale : 1

Disk Transfer/sec *Novice*

Disk Transfer/sec คือ อัตราของการอ่านหรือเขียนบน disk

Default Scale : 1

Disk Writes/sec *Novice*

Disk Writes/sec คือ อัตราของการเขียนบน disk

Default Scale : 1

Disk Write Bytes/sec *Advanced*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Disk Write Bytes/sec คืออัตราของ bytes ที่ถูก transfer ไปยัง disk ระหว่างการเขียน

Default Scale : 0.0001

Free Megabytes *Novice*

Free Megabytes แสดง space ที่ยังไม่ได้ถูก allocate บน disk drive ในหน่วย Megabytes

Default Scale : 1

IP Object

Object : **IP** *Advanced*

The IP object type ประกอบด้วยเคาน์เตอร์ที่อธิบายถึงอัตราที่ IP datagrams ได้รับและส่งไปโดยคอมพิวเตอร์ที่ใช้โปรโตคอล IP และยังอธิบายถึงความผิดพลาดต่าง ๆ ที่ใช้โปรโตคอล IP

Datagrams Forwarded/sec *Advanced*

Datagrams Forwarded/sec คือ อัตราของ input datagrams ที่ไม่ใช่ส่วนสุดท้ายของ IP destination เพื่อใช้ในการหาเส้นทางสำหรับส่งดาต้าแกรมส่วนหน้าไปยังจุดหมาย ค่านี้ไม่ได้แสดงตัวเป็น IP Gateway ค่านี้จะเป็นเพียงค่าที่รวมไปถึงแพ็คเก็ตที่เป็น Source-Route เท่านั้น

Default Scale : 0.1

Datagrams Outbound Discarded *Advanced*

Datagrams Outbound Discarded คือ จำนวนของ output IP datagrams ที่ไม่มีปัญหาซึ่งถูกเผชิญให้ป้องกันในการส่งข้อมูลไปยังจุดหมายแต่จะถูกตัดทิ้ง เช่น พื้นที่ของบัฟเฟอร์ไม่เพียงพอซึ่งค่านี้อาจจะรวมไปถึงค่า Datagrams Forwarded ถ้าค่าแพ็คเก็ตที่พบอยู่ในดุลยพินิจของช่วงวิกฤติที่จะตัดทิ้ง

Default Scale : 1

Datagrams Received Delivered/sec *Advanced*

Datagrams Received Delivered/sec คือ อัตราของ input datagramsที่ส่งสำเร็จโดยใช้โปรโตคอล IP(รวมไปถึง ICMP)

Default Scale : 0.1

Datagrams Received/sec *Advanced*

Datgrams Received/sec คือ อัตราที่ IP datagrams ได้รับจากการเชื่อมต่อรวมไปถึงความผิดพลาดต่าง ๆ ด้วย

Default Scale : 0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Datagrams Sent/sec *Advanced*

Datagrams Sent/sec คือ อัตราที่ IP datagrams ถูกสนับสนุนจาก IP สำหรับการส่งโดยใช้โปรโตคอล IP (รวมถึง ICMP) ค่านี้จะไม่รวมค่า Datagrams Forwarded

Default Scale : 0.1

Datagrams/sec *Advanced*

Datagrams/sec คืออัตราที่ IP datagrams ได้รับจากหรือส่งไปยังการเชื่อมต่อรวมไปถึงค่าความผิดพลาดและค่านี้จะไม่รวมค่า Forwarded Datagrams

Default Scale : 0.1

Memory Object

Object : **Memory** *Novice*

Memory object ประกอบด้วย counter ที่อธิบายถึงพฤติกรรมของ real และ virtual memory บนคอมพิวเตอร์ โดย real memory ถูก allocate ในหน่วยของ page และ virtual อาจมีขนาดมากกว่า real memory ทำให้เกิด page traffic ขึ้นเมื่อ virtual pages ถูกย้ายไปมาระหว่าง disk และ real memory

Available Bytes *Expert*

แสดงขนาดของ virtual memory ที่ปัจจุบันอยู่บน Zeroed , Free และ Standby lists Zeroed และ Free memory นั้นพร้อมที่จะใช้แล้ว โดย Zeroed memory จะถูกเคลียร์โดยใส่ศูนย์ลงไป ส่วน Standby memory เป็น memory ที่ถูกเอาออกมาจาก working set ของ process ที่ยังใช้ได้อยู่ ซึ่งค่านี้เป็นค่าที่นับขณะใดขณะหนึ่ง ไม่ใช่ค่าเฉลี่ยในขณะ time interval

Default Scale : 0.00001

Cache Bytes *Advanced*

จำนวน bytes ที่ถูกใช้อยู่ในปัจจุบันโดย system Cache system Cache ถูกใช้ในการเก็บข้อมูลที่นำมาจาก disk หรือ LAN system Cache ใช้ memory ที่ไม่อยู่ในระหว่างการใช้โดย active processes ในคอมพิวเตอร์

Default Scale : 0.00001

Cache Bytes Peak *Advanced*

จำนวน bytes สูงสุดที่ถูกใช้โดย system Cache system Cache ถูกใช้ในการเก็บข้อมูลที่นำมาจาก disk หรือ LAN system Cache ใช้ memory ที่ไม่อยู่ในระหว่างการใช้โดย active processes ในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default Scale : 0.00001

Cache Faults/sec *Wizard*

Cache Faults เกิดขึ้นเมื่อ Cache manager ไม่สามารถหา page ของไฟล์ได้ใน Cache และต้องร้องขอ memory manager ให้นำ page มาจาก memory หรือมาจาก disk เพื่อทำการ load page นั้นลง Cache

Default Scale : 0.1

Demand Zero Faults/sec *Wizard*

จำนวนของ page faults ซึ่งเป็น page ที่ถูกใส่ด้วยค่าศูนย์ก่อนที่จะเกิดการ fault ถ้า Zeroed list ไม่ empty คือ มี page อยู่ การ fault นี้จะแก้ได้โดยนำ page ออกมาจาก Zeroed list

Default Scale : 0.1

Free System Page Table Entries *Wizard*

จำนวนของ Page Table Entries ที่ปัจจุบันยังไม่ถูกใช้โดยระบบ

Default Scale : 0.01

Page Faults/sec *Novice*

จำนวน page faults ใน processor page fault เกิดขึ้นเมื่อ process อ้างถึง page ของ virtual memory ที่ไม่ได้อยู่ใน Working Set ของมันใน main memory การเกิด page fault นี้ไม่จำเป็นต้องนำ page มาจาก disk ถ้า page นั้นอยู่บน standby list และอยู่ใน main memory เรียบร้อยแล้ว หรือ page นั้นจะถูกใช้อยู่โดย process อื่น ซึ่งสามารถ share page กันใช้ได้

Default Scale : 0.1

Page Reads/sec *Expert*

จำนวนครั้งของการอ่าน page เข้ามาจาก disk เพื่อแก้การเกิด page fault ของ virtual memory สามารถอ่าน page เข้ามาได้หลาย page ต่อการอ่าน 1 ครั้ง

Default Scale : 1

Page Writes/sec *Expert*

จำนวนครั้งที่ page ถูกเขียนไปยัง disk เพราะ page นั้นได้ถูกเปลี่ยนแปลงไปจากตอนที่อ่าน page นั้นเข้ามา การเขียน 1 ครั้งสามารถเขียนได้หลาย page

Default Scale : 1

Pages Input/sec *Novice*

จำนวน page ที่อ่านเข้ามาจาก disk เพื่อแก้การที่ memory อ้างถึง page ที่ไม่ได้อยู่ใน memory ขณะที่อ้างถึงนั้น counter นี้รวมถึง paging traffic ของ system Cache ในการเข้าถึงไฟล์ของ application counter นี้เป็นตัวสำคัญในการสังเกตว่าเกิด memory pressure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นหรือไม่

Default Scale : 1

Pages Output/sec *Advanced*

จำนวน page ที่ถูกเขียนไปยัง disk เพราะ page นั้นได้ถูกเปลี่ยนแปลงใน main memory

Default Scale : 1

Pages/sec *Novice*

จำนวน page ที่ถูกอ่านจาก disk หรือถูกเขียนไปยัง disk เพื่อแก้การที่ memory อ้างถึง page ที่ไม่ได้อยู่ใน memory ขณะที่อ้างถึงนั้น ค่านี้เป็นผลรวมของ Pages Input/sec กับ Pages Output/sec counter นี้รวมถึง paging traffic ของ system Cache ในการเข้าถึงไฟล์ของ application counter นี้เป็นขั้นต้นในการสังเกตดูว่าเกิด memory pressure ขึ้นหรือไม่

Default Scale : 1

Pool Paged Alloc *Wizard*

จำนวนการ call เพื่อจะ allocate space ใน system Paged Pool Paged Pool คือพื้นที่ของ system memory ที่ OS. Component สามารถนำ space จากพื้นที่นี้ได้ เมื่อมันทำงานที่ได้รับมอบหมายสำเร็จ page ของ Paged Pool สามารถถูกเขียนออกไปยัง paging file เมื่อมันไม่ถูกเข้าถึงโดยระบบในเวลาที่กำหนด

Default Scale : 0.01

Pool Paged Bytes *Advanced*

จำนวน bytes ใน paged pool ซึ่งก็คือ พื้นที่ของ system memory ที่ OS. Component สามารถนำ space จากพื้นที่นี้ได้ เมื่อมันทำงานที่ได้รับมอบหมายสำเร็จ page ของ Paged Pool สามารถถูกเขียนออกไปยัง paging file เมื่อมันไม่ถูกเข้าถึงโดยระบบในเวลาที่กำหนด

Default Scale : 0.00001

Pool Nonpaged Allocs *Wizard*

จำนวนการ call เพื่อที่จะ allocate space ใน system Nonpaged Pool

Nonpaged Pool คือ พื้นที่ของ system memory ที่ OS. Component สามารถนำ space จากพื้นที่นี้ได้ เมื่อมันทำงานที่ได้รับมอบหมายสำเร็จ page ของ Nonpaged Pool ไม่สามารถถูกเขียนออกไปยัง paging file แต่จะยังอยู่ใน main memory ตราบเท่าที่มันยังถูก allocate อยู่

Default Scale : 0.01

Pool Nonpaged Bytes *Advanced*

จำนวน bytes ใน Nonpaged Pool Nonpaged Pool คือ พื้นที่ของ system memory ที่ OS. Component สามารถนำ space จากพื้นที่นี้ได้ เมื่อมันทำงานที่ได้รับมอบหมายสำเร็จ page ของ Nonpaged Pool ไม่สามารถถูกเขียนออกไปยัง paging file แต่จะยังอยู่ใน main memory ตราบเท่าที่มันยังถูก allocate อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default Scale : 0.00001

Transition Faults/sec *Wizard*

จำนวน page fault ที่แก้ไขโดยนำ page ที่อยู่ในระหว่างการ transition ซึ่งก็คือ page ที่กำลังถูกเขียนไปยัง disk ในขณะเกิด page fault โดย page นั้นถูกนำกลับมาโดยไม่ได้เกิด disk activity เพิ่ม

Default Scale : 0.1

Network Interface Object

Object : **Network Intrface** *Advanced*

The Network Interfac object tye ประกอบด้วยค่าที่อะบายถึงอัตราของไบต์และแพ็คเก็ตที่ได้รับและถูกส่งไปในเน็ทเวิร์กที่ใช้การติดต่อแบบ TCP/IP ซึ่งรวมไปถึงความผิดพลาดต่าง ๆ สำหรับการติดต่อที่เหมือน ๆ กัน

Bytes Received/sec *Advanced*

Bytes Received/sec คือ อัตราไบต์ที่ได้รับระหว่างการเชื่อมต่อรวมถึง framing characters

Default Scale : 0.0001

Bytes Sent/sec *Advanced*

Bytes Sent /sec คือ อัตราไบต์ที่ถูกส่งระหว่างการเชื่อมต่อรวมถึง framing characters

Default Scale : 0.0001

Bytes Total/sec *Advanced*

Bytes Total/sec คือ อัตราไบต์ที่ถูกส่งและได้รับระหว่างการเชื่อมต่อรวมไปถึง Framing Characters

Default Scale : 0.0001

Current Bandwidth *Advanced*

Current Bandwidth คือ การประมาณค่าของช่วงความกว้างปัจจุบันของบิทในการเชื่อมต่อ (bps) สำหรับการเชื่อมต่อนั้นจะไม่เปลี่ยนแปลงไปตามแถบความกว้างหรือการประมาณที่ไม่แน่นอนซึ่งค่านี้เป็นเพียงชื่อของแถบความกว้าง

Default Scale : 0.000001

Output Queue Length *Advanced*

Output Queue Length คือ ค่าความยาวของ output packet ที่รออยู่ในคิว ถ้าค่านี้ยาวเกิน 2 อาจทำให้เกิด delay และเกิดคอขวดขึ้นได้และควรกำจัดมันถ้าเป็นไปได้ ตั้งแต่มีการร้องขอและถูกเข้าคิวโดย NDIS ในการดำเนินการนี้ค่านี้จะเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Default Scale : 1

Packets Received/sec *Advanced*

Packets Received/sec คือ อัตราของแพ็คเก็ตที่ได้รับบนเครือข่ายการเชื่อมต่อ

Default Scale : 0.1

Packets Sent/sec *Advanced*

Packets Sent/sec คือ อัตราของแพ็คเก็ตที่ส่งไปบนเครือข่ายที่เชื่อมต่อ

Default Scale : 0.1

Packets/sec *Advanced*

Packets/sec คือ อัตราของแพ็คเก็ตที่ได้รับและส่งไปบนเครือข่ายที่เชื่อมต่อ

Default Scale : 0.1

PhysicalDisk Object

Object : **PhysicalDisk** *Advanced*

A Physical Disk object type คือ hard หรือ fixed disk drive มันอาจจะมี 1 หรือมากกว่า 1 logical partitions Disks จะถูก store file, program และ paging data. disk ถูกอ่านเพื่อที่จะดึงสิ่งต่างๆเหล่านี้ออกมา และถูกเขียนเพื่อที่จะบันทึกการเปลี่ยนแปลงลงในสิ่งเหล่านี้

% Disk Read Time *Novice*

Disk Read Time คือ เปอร์เซ็นต์เวลาที่สูญเสียไปของ disk drive ที่ถูกเลือกให้บริการการร้องขอการอ่าน

Default Scale : 1

% Disk Time *Novice*

Disk Time คือ เปอร์เซ็นต์เวลาที่สูญเสียไปของ disk drive ที่ถูกเลือกให้บริการการร้องขอการอ่านหรือการเขียน

Default Scale : 1

% Disk Write Time *Novice*

Disk Write Time คือ เปอร์เซ็นต์เวลาที่สูญเสียไปของ disk drive ที่ถูกเลือกการให้บริการการร้องขอการเขียน

Default Scale : 1

% Free Space

Percent Free Space คือ อัตราส่วนของเนื้อที่ที่ยังมีค่าของ logical disk ที่ยังไม่ได้ใช้ต่อพื้นที่ทั้งหมดของ logical disk ที่ถูกเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Avg. Disk Bytes/Read *Expert*

Avg. Disk Bytes/Read คือ ค่าเฉลี่ยของจำนวน bytes ที่ transfer จาก disk ระหว่างทำการ read

Default Scale : 0.01

Avg. Disk Bytes/Transfer *Expert*

Avg. Disk Bytes/Transfer คือ ค่าเฉลี่ยของจำนวน bytes ที่ transfer ไปยังหรือจาก disk ระหว่างการเขียนหรือการอ่าน

Default Scale : 1

Avg. Disk Bytes/Write *Expert*

Avg. Disk Bytes/Write คือ ค่าเฉลี่ยของจำนวน bytes ที่ transfer ไปยัง disk ระหว่างทำการ write

Default Scale : 1

Avg. Disk sec/Read *Advanced*

Avg. Disk sec/Read คือ ค่าเฉลี่ยของเวลาในหน่วยวินาทีของการอ่านข้อมูลจาก disk

Default Scale : 1

Avg. Disk sec/Transfer *Advanced*

Avg. Disk sec/Transfer คือ ค่าเฉลี่ยของเวลาในหน่วยวินาทีในการ transfer disk

Default Scale : 1

Avg. Disk sec/Write *Advanced*

Avg. Disk sec/Write คือ ค่าเฉลี่ยของเวลาในหน่วยวินาทีของการเขียนข้อมูลไปยัง disk

Default Scale : 1

Disk Bytes/sec *Advanced*

Disk Bytes/sec คือ จำนวนอัตรา bytes ที่ถูก transfer ไปยังหรือจาก disk ระหว่างการเขียนหรือการอ่าน

Default Scale : 0.0001

Disk Queue Length *Novice*

Disk Queue Length คือ จำนวนของการ request ที่ยังต้องทำบน disk ขณะเวลาที่การกระทำ data ถูกสะสม, มันรวมไปถึงการ request การบริการขณะเวลาของ snapshot Disk Queue Length เป็นความยาวของคิวที่เกิดขึ้นในช่วงเวลาสั้นๆไม่ได้เป็นค่าเฉลี่ยของทั้งหมดของ time interval

Default Scale : 10

Disk Read Bytes/sec *Advanced*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Disk Read Bytes/sec คือ อัตราของ bytes ที่ถูก transfer จาก disk ระหว่างการอ่าน

Default Scale : 0.0001

Disk Reads/sec *Novice*

Disk Read/sec คือ อัตราของการอ่านบน disk

Default Scale : 1

Disk Transfer/sec *Novice*

Disk Transfer/sec คือ อัตราของการอ่านหรือเขียนบน disk

Default Scale : 1

Disk Writes/sec *Novice*

Disk Writes/sec คือ อัตราของการเขียนบน disk

Default Scale : 1

Disk Write Bytes/sec *Advanced*

Disk Write Bytes/sec คืออัตราของ bytes ที่ถูก transfer ไปยัง disk ระหว่างการเขียน

Default Scale : 1

Process Object

Object : **Process** *Novice*

The Process object type ถูกสร้างขึ้นมาเมื่อ program ถูก run ทุกๆ thread ใน process มีการ share address space เดียวกันและมีการเข้าถึงข้อมูลเดียวกัน

% Privileged Time *Advanced*

Privileged Time คือ เปอร์เซ็นต์ของเวลาที่สูญเสียไปของ thread ของ processor นี้ ได้ใช้จ่ายไปในการ execute code ใน Privileged Mode เมื่อ Windows NT system service ถูกเรียก service จะ run ใน Privileged Mode เพื่อเพิ่มการเข้าถึงข้อมูลของแต่ละระบบ ดังนั้นข้อมูลจะถูกป้องกันจากการเข้าถึงโดย thread ที่ execute ใน User Mode

Default Scale : 1

% Processor Time *Novice*

% Processor Time คือ เปอร์เซ็นต์ของเวลาที่สูญเสียไปสำหรับทุก thread ของ process นั้นๆ ที่ใช้ processor ในการ execute คำสั่งและ process คือ object ที่ถูกสร้างขึ้นเมื่อ program ถูก run

Default Scale : 1

% User Time *Advanced*

% User Time คือ เปอร์เซ็นต์ของเวลาที่สูญเสียไปของ thread ของ process ที่ได้ใช้จ่ายไปใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ execute code ใน User Mode . Applications execute ใน User Mode. Code ที่ทำการ execute ใน User Mode จะไม่สามารถไปทำลายความถูกต้องของ Windows NT Executive, Kernel และ device drivers

Default Scale : 1

Elapsed Time *Advanced*

เวลาที่ใช้ไปทั้งหมดในหน่วยวินาทีของ process ขณะที่ทำการ run

Default Scale : 0.0001

File Read Bytes/sec *Expert*

File Read Bytes/sec คือ อัตราของจำนวน bytes ที่ transfer โดยการอ่านที่แจกจ่ายโดย thread ของ process นี้ บน file system devices

Default Scale : 0.0001

File Read Operations/sec *Advanced*

File Read Operations/sec คือ อัตนาของการอ่านบน file system devices ที่แจกจ่ายโดย thread ใน process นี้

Default Scale : 1

File Write Bytes/sec *Expert*

File Write Bytes/sec คือ อัตราของจำนวน bytes ที่ transfer โดยกระบวนการเขียนที่แจกจ่ายโดย threads ใน processนี้ ไปยัง file system device

Default Scale : 0.0001

File Write Operations/sec *Advanced*

File Write Operations/sec คือ อัตราของการเขียนบน file system device ที่แจกจ่ายโดย thread ใน process นี้

Default Scale : 1

Processor Object

Object : **Processor** *Novice*

Processor object type รวมไปถึง processors ทั้งหมดที่อยู่บนคอมพิวเตอร์. processor เป็นส่วนหนึ่งของคอมพิวเตอร์ที่แสดงการคำนวณทั้งเลขคณิตและตรรกศาสตร์ และรับอุปกรณ์ภายนอกเข้ามาดำเนินการ มันจะทำการ execute program บนคอมพิวเตอร์

% Privileged Time *Novice*

Privileged Time คือ เปอร์เซ็นต์ของเวลา processor ที่จ่ายให้กับ Privileged Mode ใน non-Idle threads ใน service layer ของ Windows NT ,the Executive routines และ Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kernel ทำการ execute ใน Privileged Mode . Device drivers สำหรับ device อื่นๆส่วนใหญ่จะ execute ใน Privileged Mode

Default Scale : 1

% Processor Time *Novice*

Processor Time ได้ถูกแสดงเป็นเปอร์เซ็นต์ของเวลาที่ processor ใช้ไปในการกำลัง execute thread ที่ไม่ได้หยุดว่างอยู่ มันสามารถมองเห็นได้ในอัตราส่วนของเวลาที่ใช้ไปในการทำงาน, แต่ละ process ถูกกำหนดให้กับแต่ละ Idle thread ใน Idle process ซึ่งใช้วัฏจักรของ processor ที่เปล่าประโยชน์ จะไม่ใช้โดย thread อื่นๆ

Default Scale : 1

% User Time *Advanced*

User Time คือ เปอร์เซ็นต์ของเวลา processor ที่จ่ายให้กับ User Mode ใน non-Idle threads ทั้ง application code และ subsystem code ทั้งหมดจะ execute ใน User Mode. The graphics engine ,graphic device drivers, printer device drivers และ window manager ก็จะ execute ใน User Mode ด้วย การ execute code ใน User Mode ไม่สามารถจะไปทำลายความถูกต้องของ Windows NT Executive , Kernel และ device Drivers ได้ Windows NT ใช้กระบวนการกำหนดขอบเขตสำหรับการป้องกัน subsystem ในการเพิ่มเข้าไปในการป้องกันของ User และ Privileged Modes

Default Scale : 1

Interrupt/sec *Novice*

Interrupt/sec คือ จำนวน interrupt ของ device ที่ทำการ interrupt processor. device จะ interrupt processor เมื่อ processor ทำงานเสร็จแล้วหรือเมื่อมันต้องการ. thread ที่ทำการ execute อยู่จะหยุดพักเมื่อเกิดการ interrupt และ interrupt อาจเป็นสาเหตุให้ processor switch ไปงานอื่นซึ่งเป็น thread ที่มี priority สูงกว่า, clock interrupt จะ create background ของกิจกรรมการ interrupt

Default Scale : 0.01

Server Object

Object : **Server** *Novice*

Server เป็นขบวนการในการ interface ของบริการจาก local computer ไปยังบริการของ network

Blocking Requests Rejected *Advanced*

จำนวนครั้งที่ Server ปฏิเสธการ block SMBs เพราะการนับจำนวนงานที่ว่างไม่เพียงพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งค่านี้เป็นสัญญาณบอกว่า ควรจะมีการปรับพารามิเตอร์ของ Server คือ maxworkitem หรือ minfreeworkitems หรือไม่

Default Scale : 1

Bytes Received/sec *Advanced*

จำนวน bytes ที่ server ได้รับจาก network ซึ่งเป็นตัวบอกว่า server ยุ่งขนาดไหน

Default Scale : 0.0001

Bytes Total/sec *Novice*

จำนวน bytes ที่ server ส่งไปและได้รับจาก network ค่านี้เป็นค่ารวมที่บอกว่า server ยุ่งขนาดไหน

Default Scale : 0.0001

Bytes Transmitted/sec *Advanced*

จำนวน bytes ที่ server ส่งออกไปยัง network ซึ่งเป็นตัวบอกว่า server ยุ่งขนาดไหน

Default Scale : 0.0001

Context Block Queue Time *Novice*

เวลาเฉลี่ยในหน่วย milliseconds ที่ work context block อยู่ในคิว FSP ของ server ซึ่งรอการกระทำบน request

Default Scale : 1

Context Blocks Queued/sec *Novice*

อัตราของ work context blocks ที่ถูกวางไว้ในคิว FSP ของ server เพื่อที่จะรอการกระทำของ server

Default Scale : 0.1

Errors Access Permissions *Novice*

จำนวนครั้งของการเปิดในนามของ client ที่ fail ด้วย STATUS_ACCESS_DENIED ค่านี้เป็นตัวระบุว่ามีผู้ใดพยายามที่จะเข้าถึงไฟล์ เพื่อที่จะได้รับสิ่งใดสิ่งหนึ่งที่ไม่ได้รับการป้องกันไว้อย่างดี หรือไม่

Default Scale : 1

Error Granted Access *Advanced*

จำนวนครั้งที่ถูกปฏิเสธของการเข้าถึงไฟล์ที่ถูกเปิด ค่านี้บ่งบอกถึงความพยายามในการเข้าถึงไฟล์ โดยปราศจากการอนุญาตในการเข้าถึงที่ถูกต้อง

Default Scale : 1

Errors Logon *Advanced*

จำนวนครั้งที่ความพยายาม logon บน server ไม่สำเร็จ ค่านี้เป็นตัวบ่งบอกว่าโปรแกรมการเดารหัสผ่าน ถูกใช้ในการทำลายระบบรักษาความปลอดภัยบน server หรือไม่

Default Scale : 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Errors System *Expert*

จำนวนครั้งที่ตรวจพบความผิดพลาดภายในของ server ความผิดพลาดที่ไม่ได้คาดคิดที่เกิดขึ้นนั้นเป็นตัวบ่งบอกถึงปัญหาที่เกิดขึ้นกับ server

Default Scale : 1

File Directory Searches *Advanced*

จำนวนการค้นหาไฟล์ที่ปัจจุบัน active อยู่ใน server ค่านี้บอกถึงกิจกรรมปัจจุบันของ server

Default Scale : 1

Files Open *Novice*

จำนวนไฟล์ที่ถูกเปิดในปัจจุบันบน server ค่านี้บอกถึงกิจกรรมปัจจุบันของ server

Default Scale : 1

Files Opened Total *Novice*

จำนวนของความพยายามในการเปิดที่สำเร็จ ซึ่งกระทำโดย server แต่ทำในนามของ client มีประโยชน์ในการหาปริมาณ file I/O , หา overhead สำหรับ path-based operations , หาประสิทธิภาพของ oplocks

Default Scale : 0.001

Pool Nonpaged Bytes *Expert*

จำนวน bytes ของ memory แบบ non-paged ที่ server กำลังใช้อยู่ในปัจจุบัน ค่านี้ช่วยในการหาค่าสำหรับพารามิเตอร์ MaxNonpagedMemoryUsage

Default Scale : 0.0001

Pool Nonpaged Failures *Wizard*

จำนวนครั้งที่การ allocate จาก nonpaged pool นั้น ทำไม่ได้ ค่านี้บ่งบอกว่า physical memory ของคอมพิวเตอร์นั้นเล็กเกินไป

Default Scale : 1

Pool Nonpaged Peak *Expert*

จำนวน bytes ที่มากที่สุดของ nonpaged pool ที่ server ใช้ ค่านี้บ่งบอกว่าคอมพิวเตอร์ควรจะมี physical memory มากน้อยเพียงใด

Default Scale : 0.0001

Pool Paged Bytes *Expert*

จำนวน bytes ของ memory ที่สามารถนำ page ออกได้ ที่ server กำลังใช้อยู่ ค่านี้สามารถช่วยในการหาค่าที่ถูกต้องของพารามิเตอร์ MaxPagedMemoryUsage

Default Scale : 0.0001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pool Paged Failures *Wizard*

จำนวนครั้งของการ allocate paged pool ที่ไม่สำเร็จ ตัวนับนี้บ่งบอกว่า physical memory ของ pagefile มีขนาดเล็กเกินไป

Default Scale : 1

Pool Paged Peak *Advanced*

จำนวน bytes สูงสุดของ paged pool ที่ server ได้ทำการ allocate ซึ่งเป็นตัวบอกขนาดที่เหมาะสมของ Page File(s) และ physical memory

Default Scale : 0.0001

TCP Object

Object : **TCP** *Advance*

The TCP object type ได้รวมเคาน์เตอร์ที่อธิบายอัตราต่าง ๆ ที่ TCP Segments ได้รับและส่งโดยใช้โปรโตคอล TCP นอกจากนั้นยังได้รวมคำอธิบายของจำนวนครั้งในการติดต่อ TCP ในแต่ละสถานะที่เป็นไปได้ของการติดต่อ TCP

Connection Failures *Advance*

Connection Failures คือจำนวนครั้งของการติดต่อ TCP ในการส่งโดยตรงจากสถานะ SYN-SENT หรือจากสถานะ SYN-RCVD ไปเป็นสถานะ CLOSED บวกด้วยจำนวนครั้งในการติดต่อ TCP โดยตรงจากสถานะ SYN-RCVD ไปเป็นสถานะ LISTEN

Default Scale : 1

Connections Active *Advance*

Connection Active คือ จำนวนครั้งในการติดต่อ TCP โดยตรงจากสถานะ CLOSED ไปเป็นสถานะ SYN-SENT

Default Scale : 1

Connections Established *Advance*

Connections Established คือ จำนวนของการติดต่อ TCP สำหรับสถานะปัจจุบันไม่ว่าจะเป็น ESTABLISHED หรือ CLOSED-WAIT

Default Scale : 1

Connections Passive *Advance*

Connections Passive คือจำนวนของการติดต่อ TCP โดยตรงจากสถานะ LISTEN ไปเป็นสถานะ SYN-RCVD

Default Scale : 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connections Reset *Advanced*

Connections Reset คือจำนวนครั้งในการติดต่อ TCP โดยตรงไม่ว่าจะเป็นจากสถานะ ESTABLISHED หรือ สถานะ CLOSED-WAIT ไปยังสถานะ CLOSED

Default Scale : 1

Segments Received/sec *Advance*

Segments Received/sec คือ อัตรา segments ที่ได้รับรวมไปถึง segments ที่ได้รับมาแต่เกิดความผิดพลาดด้วยซึ่งค่านี้รวมไปถึงการได้รับ segments ขณะเริ่มการติดต่อในช่วงปัจจุบันด้วย

Default Scale : 0.1

Segments Retransmitted/sec *Advance*

Segments Retransmitted/sec คือ อัตราของ segments ที่ถูกส่งอีกครั้งหนึ่งซึ่ง segments ที่ส่งนั้นบรรจุไปด้วยหนึ่งหรือหลาย ๆ ไบต์ที่ส่งไปก่อนหน้านี้

Default Scale : 0.1

Segments Sent/sec *Advance*

Segments Sent/sec คือ อัตราของ segments ที่ถูกส่งไปรวมไปถึงการติดต่อในขณะปัจจุบันแต่ไม่รวมจำนวนไบต์ที่ทำการส่งอีกครั้งหนึ่ง

Default Scale : 0.1

Segments/sec *Advance*

Segments/sec คือ อัตราที่ TCP segments ถูกส่งหรือรับโดยใช้โปรโตคอล TCP

Default Scale : 0.1

Thread Object

Object : **Thread** *Novice*

The Thread object type คือ object พื้นฐานที่ execute คำสั่งใน processor , ทุกๆ process ที่กำลัง run อยู่จะมีอย่างน้อย 1 thread

% Privileged Time *Advanced*

Privileged Time คือ เปอร์เซ็นต์ของเวลาที่สูญเสียไปของ thread นี้ ได้ใช้จ่ายไปในการ execute code ใน Privileged Mode เมื่อ Windows NT system service ถูกเรียก service จะ run ใน Privileged Mode เพื่อเพิ่มการเข้าถึงข้อมูลของแต่ละระบบ ดังนั้นข้อมูลจะถูกป้องกันจากการเข้าถึงโดย thread ที่ execute ใน User Mode

Default Scale : 1

% Processor Time *Novice*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

% Processor Time คือ เปอร์เซ็นต์ของเวลาที่ใช้ processor ไปในการ execute คำสั่งของ thread

Default Scale : 1

% User Time *Advanced*

% User Time คือ เปอร์เซ็นต์ของเวลาที่สูญเสียไปของ thread ที่ได้ใช้จ่ายไปในการ execute code ใน User Mode . Applications execute ใน User Mode. Code ที่ทำการ execute ใน User Mode จะไม่สามารถไปทำลายความถูกต้องของ Windows NT Executive, Kernel และ device drivers

Default Scale : 1

Context Switches/sec *Advanced*

Context Switches/sec คือ อัตราของการ switches จาก thread หนึ่งไปยัง thread อื่น, thread สามารถ switch กันภายใน process หรือข้าม process ก็ได้

Default Scale : 0.01

Elapsed Time *Advanced*

เวลาในหน่วยวินาทีที่ใช้ไปทั้งหมดขณะที่ thread กำลัง run

Default Scale : 0.0001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำขอขอบพระคุณอาจารย์วรวัฒน์ ลิ้มโภคา, อาจารย์บุญธีร์ เครือตราชู, พี่ศิราณี จรัสวชิระกุล, พี่วิชัย อัจฉริยบุญยงค์, พี่ไกรมน มณีศิลป์, พี่ศิริวรรณ คุณาศยัน และพี่ ๆ บริษัทสหอินโฟเทคโนโลยี จำกัด เป็นอย่างมากที่ได้ให้คำปรึกษา แนะนำ ให้ความช่วยเหลือตลอดมา และขอขอบคุณ คุณฆนานาท ลิขิตธนานันท์ ที่ได้ให้ความช่วยเหลือในด้านของการทำเอกสารของโครงงาน จึงทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Russ Blake (1993): "Optimizing Windows NT Vol.3", Microsoft Press, 1993

[2] Mark Minasai, Christa Anderson, Elizabeth Creegan: "Mastering Windows NT server 4, 3rd edition", Singapore: Tech, 1996, pp.817-872

[3] Roger Jennings, Kate Gregory: "Using Windows NT server 4, special edition", Indianapolis, IN : Que, 1996, pp.502-526

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้