ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การออกแบบระบบเครือข่ายพาณิชย์อิเล็กทรอนิกส์ที่มีความสามารถในการขยายขีดความสามารถ

Designing a Highly Available and Scalable E-Commerce Infrastructure System

H002874

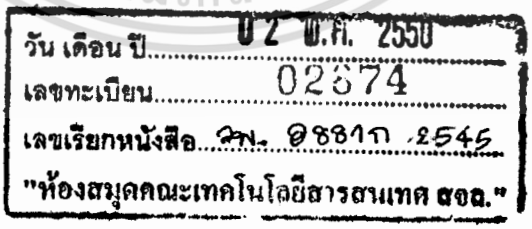
โดย

นายเอกราช กลิ่นบุบผา

รหัส 43067273

อาจารย์ที่ปรึกษา

ดร.จันทร์บูรณ์ สถิตวิริยวงศ์



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการศึกษากรณีพิเศษ

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2545

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อการ	การออกแบบระบบเครือข่ายพาณิชย์อิเล็กทรอนิกส์ที่มีความสามารถในการขยายขีดความสามารถ
นักศึกษา	นายเอกราช กลิ่นบุบผา
อาจารย์ที่ปรึกษา	ดร.จันทร์บูรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	การจัดการเทคโนโลยีสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

การออกแบบระบบที่มีความสามารถในการขยายขีดความสามารถ เป็นการศึกษาเพื่อหาว่าเทคโนโลยีใดในปัจจุบันที่มีความเหมาะสมที่สุด ในแง่ของการนำมาออกแบบระบบโครงสร้างพื้นฐานพาณิชย์อิเล็กโทรนิก จากการศึกษาพบว่าเทคโนโลยีการใช้เครื่องแม่ข่ายหลายเครื่องทำงานร่วมกันโดยมีอุปกรณ์ Intelligent Load Balancer ทำหน้าที่ในการกระจายคำเรียกขอใช้งานระบบที่ส่งจากเครื่องลูกข่ายไปยังเครื่องแม่ข่ายแต่ละเครื่องเท่าๆกัน จากการทดสอบพบว่า เมื่อมีการใช้เครื่องแม่ข่ายหลายเครื่องทำงานร่วมกันโดยมีโปรแกรม Network Load Balancing ของบริษัท Microsoft Corporation ทำหน้าที่เป็น Intelligent Load Balancer ระบบจะสามารถเพิ่มขีดความสามารถได้เมื่อมีการเพิ่มเครื่องแม่ข่ายเข้าไปในระบบเรื่อยๆ โดยดูได้จากค่า Successful Return และ ASP/second ที่เครื่องแม่ข่ายสามารถตอบสนองต่อเครื่องลูกข่ายได้ภายในเวลาที่กำหนด ซึ่งในระบบที่ทำการทดสอบซึ่งประกอบไปด้วยเครื่องแม่ข่ายจำนวนต่างกันตั้งแต่ 1, 2, 4, และ 6 เครื่องพบว่า ค่า ASP/second ที่ระบบสามารถให้บริการได้คือ 86.41 ASP/second, 166.75 ASP/second, 359.27 ASP/second, และ 527.83 ASP/second ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Designing a Highly Available and Scalable E-Commerce Infrastructure system
Student	Mr.Ekkarat Klinbubpa
Advisor	Dr. Chanboon Satidviriyavong
Level of Study	Master of Science in Information Technology
Major	Information Technology Management
Academic Year	2002

ABSTRACT

This special project called "Designing a highly available and scalable e-commerce infrastructure system" was designed find the most appropriate technology available in the market today in order to design a highly available and highly scalable e-commerce infrastructure system. The study showed that a distributed server architecture using an intelligent load balancer to distribute workload to multiple web servers in the system is well suit to this requirement. The test result from using Microsoft's Network Load Balancing Service to distribute workload to multiple web servers showed that this system scaled well in proportion to the number of web servers added to the test system. The test system's performance which was measured in terms of Successful Return and ASP/second exhibit the capability of the system to response to client's request with in a limited time frame and the result ranged from 86.41 ASP/second, 166.75 ASP/second, 359.27 ASP/second, and 527.83 ASP/second on a system running 1, 2, 4, and 6 web servers respectively.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทคัดย่อภาษาไทย		
บทคัดย่อภาษาอังกฤษ		
สารบัญ		
บทที่ 1 บทนำ		
1.1	ความเป็นมา	1
1.2	แนวโน้มที่สำคัญของการใช้งานระบบอินเตอร์เน็ตและอินทราเน็ต	1
1.3	วัตถุประสงค์ในการศึกษา	3
1.4	วัตถุประสงค์ในการทดสอบ	3
1.5	การวิเคราะห์ผลการศึกษา	4
1.6	ผลการทดลองที่คาดว่าจะได้รับ	4
บทที่ 2 เทคโนโลยีการออกแบบระบบโครงสร้างพื้นฐานพาณิชย์อิเล็กโทรนิก		5
2.1 การออกแบบระบบให้บริการเว็บไซต์ (Web Server)		5
2.2 แนวทางการเลือกใช้งาน Intelligent Load Balancer		9
2.3 อัลกอริธึมการทำงานของ Load Balancer ที่สำคัญ		9
บทที่ 3 วิธีการศึกษา		11
3.1	วิธีการศึกษา	11
3.2	วัตถุประสงค์ในการทดสอบ	12
3.3	ขั้นตอนการทำการศึกษา	12
บทที่ 4 ผลการศึกษาที่ได้		29
4.1	ผลการทดสอบประสิทธิภาพในการขยายความสามารถในการทำงาน	29
บทที่ 5 บทสรุปและข้อเสนอแนะ		45
5.1	สรุปผลการทดสอบความสามารถในการขยายความสามารถของระบบ	45
5.2	ข้อเสนอแนะในการทดลอง	46
บรรณานุกรม		47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 4.1 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 1 เครื่อง 30

ตารางที่ 4.2 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 2 เครื่อง 33

ตารางที่ 4.3 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 4 เครื่อง 36

ตารางที่ 4.4 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 6 เครื่อง 39

ตารางที่ 4.5 แสดงการเปรียบเทียบประสิทธิภาพของระบบที่มีจำนวนเครื่องแม่ข่ายแตกต่างกัน 41

ตารางที่ 4.6 แสดงประสิทธิภาพของระบบที่เครื่องแม่ข่ายทำงานร่วมกันแบบ Load-Balance เมื่อเปรียบเทียบกับระบบที่เครื่องแม่ข่ายแต่ละเครื่องทำงานแยกกันอย่างเป็นอิสระ 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 1.1มูลค่าการซื้อขายบนอินเตอร์เน็ตระหว่างปี 1998-2002	2
รูปที่ 1.2ความสำคัญของระบบที่ใช้ในอินเตอร์เน็ต เอ็กซ์ทราเน็ต และอินทราเน็ตที่มีต่อผู้บริหารองค์กร	2
รูปที่ 1.3 รูปแบบการพัฒนาและออกแบบระบบพาณิชย์อิเล็กทรอนิกส์ที่นิยมใช้ในปัจจุบัน	3
รูปที่ 2.1 แสดงการทำงานของ Windows NT 4.0 Server Enterprise Edition แบบ Cluster	6
รูปที่ 2.2 แสดงการทำงานของ DNS Round Robin	7
รูปที่ 2.3 แสดงรูปแบบการทำงานโดยใช้ Intelligent Load Balancer	8
รูปที่ 3.1 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 1 เครื่องต่อเชื่อมเข้ากับระบบเครือข่าย และมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ	13
รูปที่ 3.2 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 2 เครื่องต่อเชื่อมเข้ากับระบบเครือข่าย และมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ	14
รูปที่ 3.3 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 4 เครื่องต่อเชื่อมเข้ากับระบบเครือข่าย และมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ	15
รูปที่ 3.4 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 6 เครื่องต่อเชื่อมเข้ากับระบบเครือข่าย และมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ	16
รูปที่ 3.5 เครื่องแม่ข่ายรุ่นโปรไลแอนท์ บีแอล 10 อี (Proliant BL 10e) ภาพล่าง และตู้บรรจุเครื่องแม่ข่าย (server enclosure – รูปบน)	17
รูปที่ 3.6 Diagnostic Adapter ที่ใช้ต่อ VGA, keyboard, และ mouse เข้ากับเครื่องแม่ข่าย ProLiant BL10e	18
รูปที่ 3.7 แสดงการติดตั้งเครื่องแม่ข่าย ProLiant BL10e ผ่านทางโปรแกรม ProLiant Essential – Rapid Deployment Pack	19
รูปที่ 3.8 แสดงการเลือก Properties ของ Local Area Connection เพื่อติดตั้ง Network Load Balancing Service	20
รูปที่ 3.9 แสดงการเริ่มต้นติดตั้ง Network Load Balancing Service	21
รูปที่ 3.10 แสดงการกำหนดค่าต่างๆในส่วนของ Cluster Parameters ของ Network Load Balancing Service ในเครื่องแม่ข่ายเครื่องที่ 1	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 3.11 แสดงการกำหนดค่าต่างๆในส่วนของ Host Parameters ของ Network Load Balancing Service ในเครื่องแม่ข่ายเครื่องที่ 1 23

รูปที่ 3.12 แสดงการกำหนดค่าต่างๆในส่วนของ Port Rules ของ Network Load Balancing Service ในเครื่องแม่ข่ายเครื่องที่ 1 24

รูปที่ 3.13 แสดงการปรับตั้งค่าโปรแกรม InetMonitor เพื่อใช้ในการทดสอบ 27

รูปที่ 4.1 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 1 เครื่อง 31

รูปที่ 4.2 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 1 เครื่อง 31

รูปที่ 4.3 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 2 เครื่อง 34

รูปที่ 4.4 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 2 เครื่อง 34

รูปที่ 4.5 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 4 เครื่อง 37

รูปที่ 4.6 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 4 เครื่อง 37

รูปที่ 4.7 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 6 เครื่อง 40

รูปที่ 4.8 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 6 เครื่อง 40

รูปที่ 4.9 แสดงความสามารถของระบบที่มีจำนวนเครื่องแม่ข่ายแตกต่างกัน 42

รูปที่ 4.10 แสดงร้อยละของ Succssful Return ในแต่ละระบบเมื่อเทียบกับจำนวน Request ทั้งหมดที่เกิดขึ้นในการทดสอบ 42

รูปที่ 4.11 แสดงร้อยละของ Timeout Request ในแต่ละระบบเมื่อเทียบกับจำนวน Request ทั้งหมดที่เกิดขึ้นในการทดสอบ 43

รูปที่ 4.12 แสดงการเปรียบเทียบประสิทธิภาพของระบบที่เครื่องแม่ข่ายทำงานร่วมกันแบบ Load-Balance เมื่อเปรียบเทียบกับระบบที่เครื่องแม่ข่ายแต่ละเครื่องทำงานแยกกันอย่างเป็นอิสระ 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมา

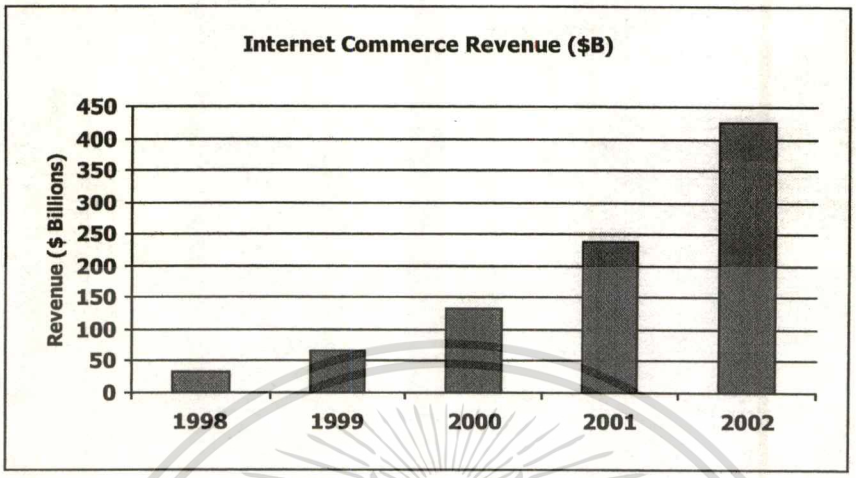
การขยายตัวของเทคโนโลยีอินเตอร์เน็ตอย่างรวดเร็วได้ส่งผลกระทบอย่างชัดเจนต่อการเขียนและติดตั้งระบบและโปรแกรมคอมพิวเตอร์ต่างๆ สิ่งหนึ่งที่เห็นได้ชัดก็คือในอดีตเว็บไซต์ส่วนใหญ่จะมีข้อมูลในลักษณะที่ตายตัวและมีการเปลี่ยนแปลงน้อย แต่เมื่อเทคโนโลยีใหม่ๆเพิ่มมากขึ้น เว็บไซต์ดังกล่าวก็ค่อยๆมีการพัฒนารูปแบบมาเป็นลักษณะที่สามารถเปลี่ยนแปลงรูปแบบได้หลากหลาย (Dynamic page) ในบริษัทและองค์กรขนาดใหญ่ก็เริ่มมีการปรับปรุงโปรแกรมเดิมหรือมีการพัฒนาโปรแกรมใหม่ๆเพื่อนำมาใช้ในระบบอินทราเน็ตกันมากขึ้น ดังนั้นเมื่อมีการใช้งานอินเตอร์เน็ตเพิ่มมากขึ้นเรื่อยๆเช่นนี้จึงทำให้เกิดคำถามขึ้นว่า ทำอย่างไรเราจึงจะสามารถพัฒนาระบบโครงสร้างอินเตอร์เน็ตเพื่อการพาณิชย์หรือที่เรียกกันว่าระบบพาณิชย์อิเล็กทรอนิกส์ (e-commerce system) ให้มีความสามารถในการขยายขีดความสามารถ (scalable)

1.2 แนวโน้มที่สำคัญของการใช้งานระบบอินเตอร์เน็ตและอินทราเน็ต

จากการศึกษาของบริษัทที่ทำการศึกษาด้านการตลาดชื่อ IDC ได้ทำการพยากรณ์ไว้ว่ามูลค่าการซื้อขายบนเครือข่ายอินเตอร์เน็ตจะเพิ่มจากประมาณ 67 พันล้านเหรียญสหรัฐฯ ขึ้นไปเป็น 400 พันล้านเหรียญสหรัฐฯภายในปี2002 ซึ่งแนวโน้มนี้แสดงให้เห็นว่าโปรแกรมหรือระบบคอมพิวเตอร์ที่ใช้ในเครือข่ายอินเตอร์เน็ตจะกลายเป็นระบบที่มีความสำคัญต่อองค์กร (mission critical) มากขึ้นเรื่อยๆ ดังนั้นในการออกแบบระบบดังกล่าว จึงต้องคำนึงถึงความสามารถในการขยายขีดความสามารถเพื่อรองรับงานที่จะเพิ่มมากขึ้นในอนาคตเป็นเรื่องสำคัญด้วย

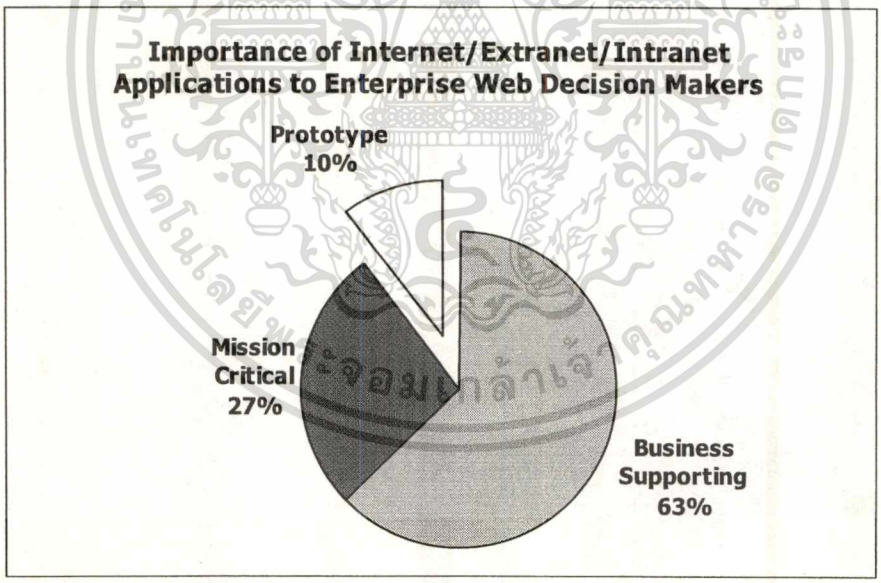
นอกจากนี้หากมองภาพภายในบริษัทหรือองค์กรขนาดใหญ่ในช่วงที่ผ่านมาจะพบว่าในบรรดาระบบคอมพิวเตอร์ที่มีการพัฒนาและติดตั้งเพื่อใช้ในอินทราเน็ต เอ็กซ์ทราเน็ต และอินเตอร์เน็ตขององค์กรหนึ่งๆ จะมีระบบที่จัดว่ามีความสำคัญต่อการทำงานและการตัดสินใจของบริษัทหรือองค์กรนั้นๆ (mission critical) อยู่ในสัดส่วนที่สูงพอสมควร ซึ่งระบบเหล่านี้จำเป็นที่จะต้องมีการออกแบบที่ระเอียดรอบคอบและสามารถทำงานได้อย่างมีประสิทธิภาพอยู่ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 มูลค่าการซื้อขายบนอินเตอร์เน็ตระหว่างปี 1998-2002

ที่มา: International Data Corporation, 1998

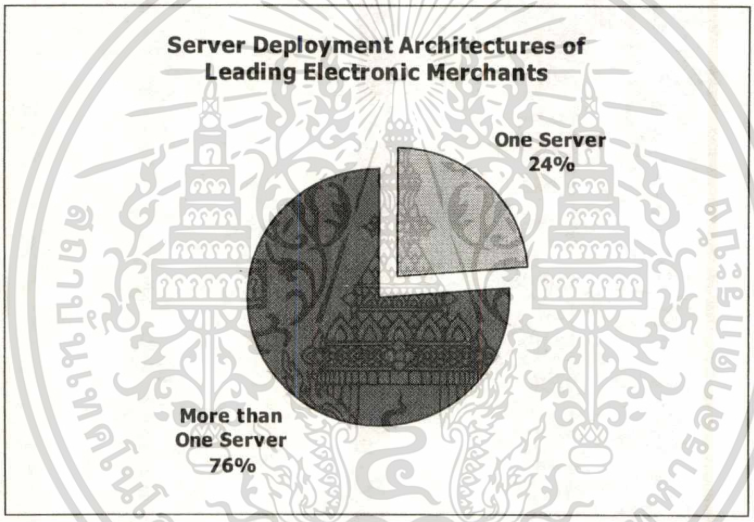


รูปที่ 1.2 ความสำคัญของระบบที่ใช้ในอินเตอร์เน็ต เอ็กซ์ทราเน็ต อินทราเน็ตที่มีต่อผู้บริหารองค์กร

ที่มา: ประยุกต์มาจากการสำรวจของ IDC Web Decision Maker Survey, ธันวาคม 1997

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นแนวโน้มที่สำคัญของนักพัฒนาและการออกแบบระบบเครือข่ายพาณิชย์อิเล็กทรอนิกส์ในปัจจุบันจึงมีแนวโน้มที่จะมุ่งเน้นการออกแบบและพัฒนาเพื่อให้ระบบมีความสามารถในการขยายขีดความสามารถ (scalable) และสามารถทำงานได้อย่างต่อเนื่องตลอดเวลา (available) ให้สูงที่สุดภายใต้งบประมาณที่เหมาะสม โดยที่รูปแบบหนึ่งที่กำลังได้รับความนิยมมากขึ้นก็คือการออกแบบระบบในลักษณะที่มีเครื่องคอมพิวเตอร์แม่ข่าย (server) หลายๆเครื่องและกระจายการทำงาน (distributed) ออกเป็นหลายๆชั้น (multi-tier) แทนที่จะใช้เครื่องคอมพิวเตอร์ขนาดใหญ่เพียงเครื่องเดียวทำงานทุกอย่างทั้งหมดพร้อมๆกัน ดังในรูปที่ 3 แสดงให้เห็นว่านักออกแบบระบบพาณิชย์อิเล็กทรอนิกส์ในปัจจุบันให้การยอมรับและใช้หลักการการออกแบบในลักษณะนี้กันมากขึ้น



รูปที่ 1.3 รูปแบบการพัฒนาและออกแบบระบบพาณิชย์อิเล็กทรอนิกส์ที่นิยมใช้ในปัจจุบัน

ที่มา: ประยุกต์มาจากการสำรวจของ Intermarket Group eMerchant ปี 1998

1.3 วัตถุประสงค์ในการศึกษา

1. เพื่อทำการศึกษาว่ามีเทคโนโลยีใดบ้างในปัจจุบันที่มีความเหมาะสมในการนำมาออกแบบเพื่อระบบพาณิชย์อิเล็กทรอนิกส์ที่มีความสามารถในการเพิ่มขีดความสามารถ
2. เพื่อนำเทคโนโลยีที่ทำการศึกษาได้มาสร้างแบบจำลองระบบพาณิชย์อิเล็กทรอนิกส์และทดสอบความสามารถในการเพิ่มขีดความสามารถ

1.4 วัตถุประสงค์ในการทดสอบ

ในการทดสอบแบบจำลอง มีวัตถุประสงค์ดังต่อไปนี้:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เพื่อประเมินคุณสมบัติของระบบในรูปแบบต่างๆโดยวัดจากความสามารถในการขยายขีดความสามารถ
2. เพื่อหาข้อสรุปและข้อเสนอแนะรวมทั้งข้อดีข้อเสียต่างๆที่ต้องคำนึงถึงเมื่อจะทำการติดตั้งระบบในแต่ละรูปแบบที่ทำการศึกษา

1.5 การวิเคราะห์ผลการศึกษา

ในการวิเคราะห์ผลที่ได้จากการศึกษาทดสอบจะใช้ความสามารถในการขยายขีดความสามารถเป็นเกณฑ์ในการตัดสิน ในการวัดความสามารถในการขยายขีดความสามารถของระบบ จะใช้ค่าความสามารถในการตอบรับการเรียกใช้ (throughput) ของ Active Server Pages ในเวลา 1 วินาที (ASP/sec) ของระบบนั้นๆเมื่อเทียบกับระบบมาตรฐาน (เครื่อง stand alone ที่ 1 หน่วยประมวลผล)

1.6 ผลการทดลองที่คาดว่าจะได้รับ

จากการพิจารณาคุณลักษณะของระบบต่างๆโดยใช้หลักการพิจารณาข้างต้น เราอาจประมาณการผลการทดลองที่คาดว่าจะได้รับได้ดังนี้

1.6.1 ระบบที่ใช้เครื่องแม่ข่ายเพียงเครื่องเดียวจะไม่สามารถทำงานต่อได้หากเกิดข้อผิดพลาดขึ้นกับโปรแกรมที่ใช้งานอยู่

ถึงแม้ว่าเราจะสามารถเลือกใช้เครื่องแม่ข่ายที่มีการใช้อุปกรณ์ที่สามารถป้องกันข้อผิดพลาดของเครื่องเช่น Error Checking and Correcting (ECC) RAM, Redundant Power Supply, Redundant Fan, RAID Array ก็ตาม อุปกรณ์ที่กล่าวมาทั้งหมดนี้จะไม่สามารถช่วยอะไรได้เลยหากปัญหาที่เกิดขึ้นนั้นมีสาเหตุมาจากโปรแกรมที่ใช้งานอยู่ในเครื่อง เช่นข้อผิดพลาดในระบบปฏิบัติการ (bug in Operating System) หรือไวรัสเป็นต้น

1.6.2 ระบบที่ใช้เครื่องแม่ข่ายหลายเครื่องทำงานร่วมกับ Intelligent Load Balancer สามารถขยายขีดความสามารถได้อย่างคงที่และเป็นสัดส่วนโดยตรงกับจำนวนเครื่องแม่ข่าย

เนื่องจากงานที่เรียกจากเครื่องลูกข่ายมาหาเครื่องที่เป็น Web Server จะถูกกระจายไปสู่ Web Server หลายๆตัวด้วยการทำงานของเครื่อง IP Load Balancer ดังนั้นการเพิ่มปริมาณ Web Server หรือ Application Server เข้าไปในระบบจึงเปรียบเสมือนการเพิ่มปริมาณแรงงานในการทำงาน หากเครื่องแม่ข่าย 1 เครื่องสามารถรองรับเครื่องลูกข่ายได้ 500 เครื่อง การเพิ่มเครื่องแม่ข่ายเครื่องที่ 2 เข้าไปในระบบก็ย่อมจะทำให้ระบบสามารถรองรับลูกข่ายเพิ่มขึ้นได้อีก 500 เครื่องเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

เทคโนโลยีการออกแบบระบบโครงสร้างพื้นฐานพาณิชย์อิเล็กโทรนิก

2.1 การออกแบบระบบให้บริการเว็บไซต์ (Web Server)

ในปัจจุบันมีแนวทางการออกแบบระบบ Web Server มีด้วยกัน 4 วิธีหลักคือ

- การใช้เครื่องที่มีหน่วยประมวลผลหลายหน่วยในเครื่องเดียว (Stand Alone Multi-processor Server)
- การใช้เครื่องที่มีหน่วยประมวลผลหลายหน่วยสองเครื่องหรือมากกว่าทำงานร่วมกันแบบ Cluster (Clustered Multi-processor Server)
- การกระจายงานให้หลายเครื่องทำงานร่วมกันด้วย DNS Server (Round Robin DNS)
- การกระจายงานให้หลายเครื่องทำงานร่วมกันด้วยอุปกรณ์พิเศษ (Intelligent Load Balancing)

2.1.1 การใช้เครื่องที่มีหน่วยประมวลผลหลายหน่วยในเครื่องเดียว (Standalone Multi-processor Servers)

ในช่วงแรกที่เริ่มมีการสร้าง web site หรือ e-commerce site นั้นนิยมออกแบบระบบโดยให้มีเครื่องแม่ข่ายเพียงเครื่องเดียวทำงานเป็นทั้ง web server และเครื่องเก็บข้อมูลในเวลาเดียวกัน หากเครื่องมีการเรียกใช้งานมากขึ้นและต้องการขยายประสิทธิภาพเพื่อให้สามารถรองรับผู้ใช้งานได้จำนวนมากขึ้น ก็จะใช้วิธีการขยายทรัพยากรที่สำคัญของเครื่องให้มากขึ้น เช่นการเพิ่มจำนวนหน่วยประมวลผล (CPU) และหน่วยความจำหลัก (RAM) อย่างไรก็ดี วิธีการเพิ่มทรัพยากรระบบให้มากขึ้นนี้จะช่วยเพิ่มประสิทธิภาพในระดับหนึ่งเท่านั้น ทั้งนี้เพราะโปรแกรมบางโปรแกรมไม่สามารถขยายความสามารถเป็นสัดส่วนโดยตรงกับทรัพยากรที่เพิ่มขึ้นได้ ในขณะเดียวกันเครื่องที่ใช้ก็มีข้อจำกัดในการขยายทรัพยากรดังกล่าวด้วย ยกตัวอย่างเช่น เครื่องมาเครื่องสามารถรองรับการขยายหน่วยความจำได้เพียง 4 GB ดังนั้นเมื่อใส่หน่วยความจำจนเต็มแล้ว ก็จะไม่สามารถขยายได้อีกต่อไป

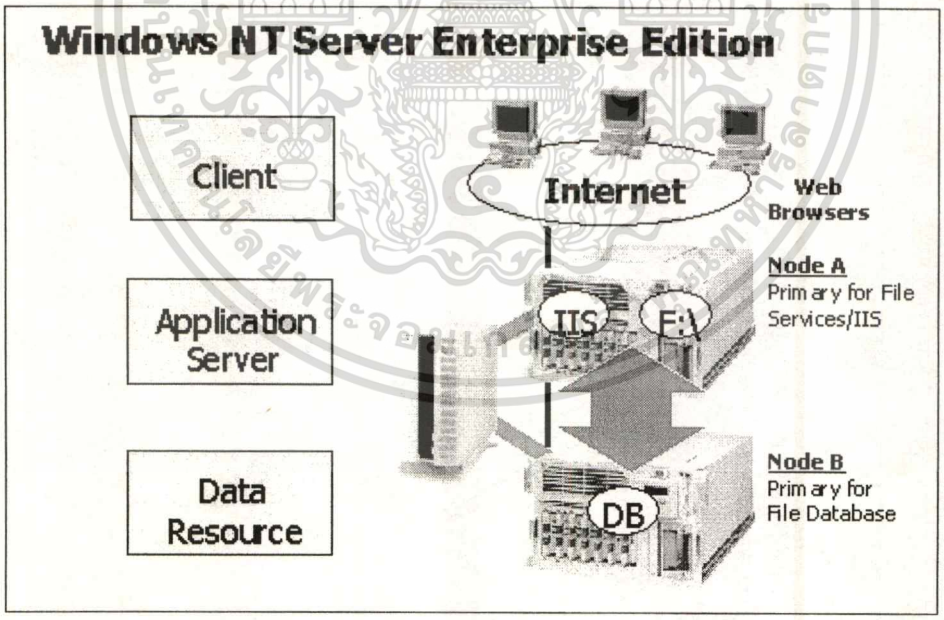
ส่วนกรณีของความสามารถในการทำงานอย่างต่อเนื่องของระบบนั้น เครื่องแม่ข่ายสามารถใช้เทคโนโลยีหลายๆประการเช่น Hard Disk Array, ECC memory, แหล่งจ่ายไฟฟ้าสำรอง,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัดลมระบายความร้อนสำรอง และอื่นๆ เพื่อช่วยป้องกันหรือลดความเสี่ยงที่ระบบจะเกิดข้อผิดพลาดจนไม่สามารถทำงานต่อได้หรือประสิทธิภาพลดลง

2.1.2 การใช้เครื่องที่มีหน่วยประมวลผลหลายหน่วยสองเครื่องหรือมากกว่าทำงานร่วมกันแบบ (Clustered Multi-processor Servers)

เทคโนโลยี Clustering เป็นเทคโนโลยีที่มาช่วยเสริมความสามารถในการทำงานอย่างต่อเนื่องของระบบให้กับโปรแกรมต่างๆ โดยที่เทคโนโลยีนี้จะช่วยให้โปรแกรมสามารถโอนย้ายการทำงาน (fail-over) จากเครื่องที่มีปัญหาไปยังเครื่องที่เตรียมพร้อมไว้ได้โดยอัตโนมัติ ยกตัวอย่างการทำงานของโปรแกรม Internet Information Server ที่ทำงานอยู่บนเครื่องแม่ข่าย โดยที่มีการเก็บข้อมูลอยู่บนตู้เก็บข้อมูลภายนอก (external storage) ในขณะทำงาน โปรแกรมจะทำงานอยู่บนเครื่องแม่ข่ายเครื่องหนึ่งและมีแม่ข่ายอีกเครื่องหนึ่งซึ่งต่ออยู่กับตู้เก็บข้อมูลภายนอกเช่นกันเตรียมพร้อมที่จะทำงานอยู่ (stand-by) หากเครื่องที่โปรแกรมทำงานอยู่เกิดข้อผิดพลาดขึ้นจนไม่สามารถทำงานต่อได้ โปรแกรมก็จะถูกย้ายไปทำงานที่เครื่อง stand-by โดยอัตโนมัติและไม่เกิดผลเสียต่อผู้ใช้งาน



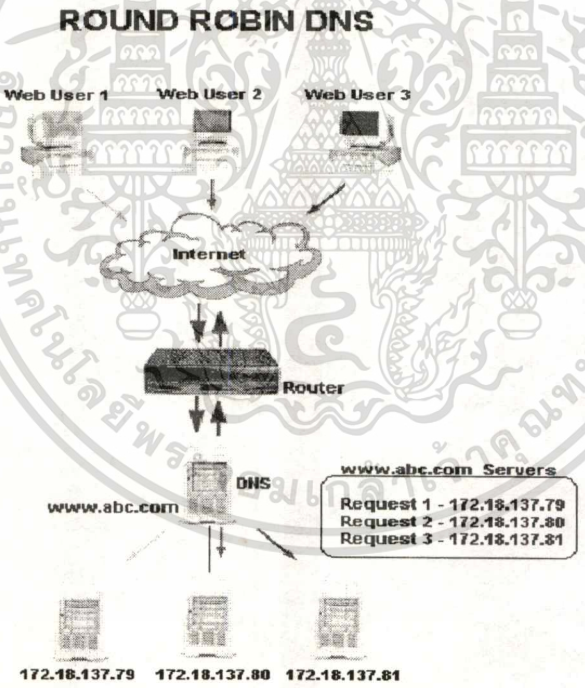
รูปที่ 2.1 แสดงการทำงานของ Windows NT 4.0 Server Enterprise Edition แบบ Cluster

2.1.3 การกระจายงานให้หลายเครื่องทำงานร่วมกันด้วย DNS Server (Round Robin DNS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานแบบ Round Robin DNS เป็นวิธีการกำหนดให้เครื่องแม่ข่ายที่ทำหน้าที่เป็น Domain Name Service (DNS) เพื่อให้เครื่อง DNS Server สามารถกระจายคำเรียกขอใช้บริการมาที่ชื่อเครื่อง (host name) ชื่อหนึ่งไปยังเครื่องแม่ข่ายหลายๆเครื่องที่มี IP address ที่แตกต่างกันได้ ดังนั้นผู้ใช้งานแต่ละคนที่ส่งคำขอเข้ามาใช้งานระบบ จะถูกส่งไปหาเครื่องแม่ข่ายเครื่องใดเครื่องหนึ่งในระบบเครือข่ายซึ่งถูกกำหนดไว้เสมอ

วิธีการใช้ Round Robin DNS นี้จัดเป็นวิธีที่ทำการติดตั้งง่ายและมีค่าใช้จ่ายต่ำ อย่างไรก็ดี ข้อจำกัดอย่างหนึ่งที่สำคัญของวิธีนี้ก็คือ DNS Server จะไม่มีทางรู้เลยว่าในขณะหนึ่งๆ เครื่องแม่ข่ายแต่ละตัวมีงานค้างอยู่ (Work load) เท่าใด ดังนั้นหากในขณะนั้นเครื่องแม่ข่ายดังกล่าวกำลังอยู่ในสภาพ over load หรือไม่สามารถทำงานได้อยู่ ผู้ใช้ระบบที่ถูกส่งไปหาเครื่องแม่ข่ายเครื่องดังกล่าวก็จะไม่สามารถใช้งานระบบได้เลยจนกว่าผู้ดูแลระบบจะลบ IP Address ของเครื่องนั้นออกจากรายการของ DNS server



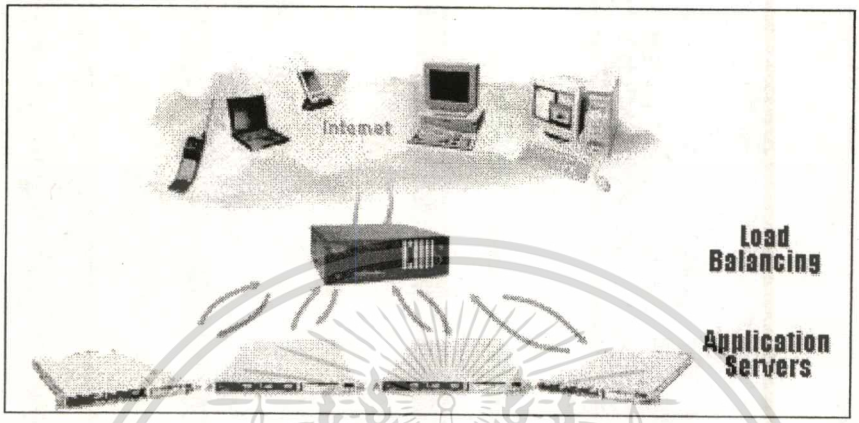
รูปที่ 2.2 แสดงการทำงานของ DNS Round Robin

2.1.4 การกระจายงานให้หลายเครื่องทำงานร่วมกันด้วยอุปกรณ์พิเศษ (Intelligent Load Balancing)

เนื่องจากการทำงานแบบ Round Robin DNS มีข้อจำกัดในการทำงานอย่างที่กล่าวมาข้างต้น ดังนั้นจึงได้มีการคิดค้นอุปกรณ์พิเศษเพื่อมาช่วยในการกระจายงานโดยที่มีความสามารถในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวนหาความสามารถของระบบในขณะนั้นและกระจายงานอย่างเหมาะสม พร้อมทั้งยังเพิ่มความสามารถในการทำงานอย่างต่อเนื่องของระบบได้อีกด้วย



รูปที่ 2.3 แสดงรูปแบบการทำงานโดยใช้ Intelligent Load Balancer

ข้อดีประการแรกของการใช้ Intelligent Load Balancer ก็คือ ในขณะที่ระบบทำงานอุปกรณ์ที่เรียกว่า Intelligent Load Balancer จะคอยทำการตรวจวัดการทำงานของเครื่องแม่ข่ายทั้งหมดที่กำหนดไว้ หากเครื่อง Intelligent Load Balancer ตรวจพบว่ามีเครื่องแม่ข่ายเครื่องใดเครื่องหนึ่งเกิดปัญหาจนไม่สามารถทำงานต่อได้ เครื่อง Intelligent Load Balancer ก็จะทำการตัดเครื่องนั้นออกจากรายการที่มีการกระจายงานไปให้โดยอัตโนมัติ และส่งคำขอการใช้งานจากผู้ใช้ที่ส่งเข้ามาไปหาเครื่องที่เหลือทันที

ข้อดีประการที่สองคือการใช้ Intelligent Load Balancer สามารถช่วยให้ผู้ดูแลระบบสามารถเพิ่มความสามารถในการทำงานของระบบได้อย่างรวดเร็ว เมื่อผู้ดูแลระบบพบว่าความสามารถของระบบที่มีอยู่ขณะนั้นไม่เพียงพอกับความต้องการของผู้เรียกใช้งาน ผู้ดูแลระบบก็เพียงแต่ทำการติดตั้งเครื่องแม่ข่ายเพิ่มขึ้นอีกตามจำนวนที่คาดว่าน่าจะเพียงพอกับการเรียกใช้ จากนั้นจึงทำการแก้ไขข้อมูลที่ Intelligent Load Balancer เพื่อให้เพิ่ม IP Address ของเครื่องแม่ข่ายที่ติดตั้งใหม่เข้าไปในรายการเครื่องที่จะมีการกระจายงานไปให้ เมื่อดำเนินการเสร็จเรียบร้อยแล้ว Intelligent Load Balancer ก็จะทำการกระจายงานไปให้เครื่องที่ติดตั้งใหม่โดยอัตโนมัติ โดยพิจารณาจากความสามารถและงานของแต่ละเครื่องที่ทำอยู่เป็นหลัก

2.2 แนวทางการเลือกใช้งาน Intelligent Load Balancer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปัจจุบันมี Intelligent Load Balancer ให้เลือกใช้อยู่สองกลุ่มหลักๆคือ Software-base Load Balancer ซึ่งประกอบด้วยโปรแกรมคอมพิวเตอร์ที่มีหน้าที่ทำงานในการกระจายงานและเครื่องแม่ข่ายซึ่งเป็นเครื่องแม่ข่ายมาตรฐานทั่วไป ตัวอย่างของ Intelligent Load Balancer ได้แก่ โปรแกรม Network Load Balancing Service ที่มีมาพร้อมกับโปรแกรม Windows 2000 Advance Server ของบริษัทไมโครซอฟท์ และโปรแกรม Local Dispatch และ Global Dispatch ของบริษัท Resonate เป็นต้น

อีกกรณีหนึ่งคือ Hardware-base Load Balancer เป็นอุปกรณ์สำเร็จรูปซึ่งออกแบบมาเพื่อทำหน้าที่เป็น Load Balancer โดยเฉพาะ ส่วนใหญ่เมื่อทำการติดตั้งเข้าไปในระบบแล้วจะต้องทำการปรับแต่งระบบอีกเพียงเล็กน้อยก็จะสามารถทำงานได้ทันทีและโดยทั่วไปแล้วมักจะมีประสิทธิภาพสูงกว่า Software-base Load Balancing เนื่องจากถูกออกแบบมาให้ทำงานนี้โดยเฉพาะ ตัวอย่างของอุปกรณ์ในกลุ่มนี้ได้แก่ Radware Web Server Director, Cisco Local Director, และ F5 Big IP Load Balancer เป็นต้น

ไม่ว่าจะเป็น Software-base Load Balancer หรือ Hardware-base Load Balancer ก็ตาม อุปกรณ์ทั้งสองแบบต่างมีหลักการทำงานที่คล้ายคลึงกัน กล่าวคืออุปกรณ์นี้จะทำหน้าที่สร้าง virtual IP address ขึ้นมาหมายเลขหนึ่งเพื่อให้ผู้ใช้งานที่อยู่ในเครือข่ายอินเตอร์เน็ตเรียกเข้ามาใช้งาน เมื่อมีการเรียกขอใช้งานระบบจากผู้ใช้ส่งเข้ามาที่ Intelligent Load Balancer เครื่องจะทำการแปลง virtual IP address ดังกล่าวไปเป็น IP address ของเครื่องแม่ข่ายที่มีอยู่จริงในระบบด้วยเทคนิกที่เรียกว่า Network Address Translation (NAT) ซึ่งเครื่องแม่ข่ายนี้มักจะมีอยู่มากกว่าหนึ่งเครื่องเพื่อให้ Intelligent Load Balancer สามารถกระจายงานไปให้หลายๆเครื่องได้

ด้วยวิธีนี้ เมื่อมีการเรียกขอใช้งานระบบจำนวนมากส่งเข้ามาพร้อมๆกัน Intelligent Load Balancer ก็จะทำการกระจายคำขอใช้งานไปยังเครื่องต่างๆในปริมาณที่เหมาะสม ในเวลาเดียวกันเครื่องก็จะคอยตรวจสอบเครื่องแม่ข่ายทั้งหมดด้วยว่ามีเครื่องใดมีปัญหาหรือไม่ หากพบปัญหาเกิดขึ้น Intelligent Load Balancer ก็จะหยุดส่งคำขอไปที่เครื่องนั้นแต่จะส่งไปที่เครื่องอื่นๆที่เหลือแทน ทำให้ผู้ใช้สามารถใช้งานระบบได้อย่างต่อเนื่อง

2.3 อัลกอริธึมการทำงานของ Load Balancer ที่สำคัญ

อัลกอริธึมหรือวิธีการทำงานในการกระจายงานของ Intelligent Load Balancer มีอยู่ด้วยกันหลายวิธีและต่างก็มีข้อดีและความเหมาะสมในการใช้งานที่แตกต่างกัน ตัวอย่างของอัลกอริธึมที่สำคัญได้แก่ Round Robin, Least-Connections, และ Response Time

2.3.1 อัลกอริธึมแบบ Round Robin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการกระจายงานไปยังเครื่องแม่ข่ายหลายๆเครื่องเท่าๆกันตามลำดับที่ได้กำหนดไว้แล้ว แม้ว่าวิธีนี้จะคล้ายคลึงกับวิธี Round Robin DNS ที่กล่าวถึงไปแล้วข้างต้นแต่ก็มีข้อดีกว่าคือ Intelligent Load Balancer จะสามารถตรวจสอบได้ว่าเครื่องแม่ข่ายแต่ละเครื่องยังทำงานเป็นปกติอยู่หรือไม่ หากเครื่องไหนไม่สามารถทำงานได้ก็จะถูกตัดออกจากระบบโดยอัตโนมัติเพื่อให้ระบบทำงานได้อย่างต่อเนื่องและไม่เกิดผลกระทบต่อผู้ใช้งาน นอกจากนี้ผู้ดูแลระบบยังสามารถกำหนดน้ำหนักการกระจายงานไปยังเครื่องต่างๆได้ในกรณีที่เครื่องแม่ข่ายที่ใช้มีความสามารถแตกต่างกัน ทั้งนี้ก็เพื่อป้องกันไม่ให้เครื่องใดเครื่องหนึ่งรับภาระงานหนักจนเกินไปและไม่สามารถทำงานต่อได้

2.3.2 อัลกอริธึมแบบ Least Connections

เป็นอัลกอริธึมที่มีการวัดจำนวนการเชื่อมต่อของผู้ใช้งานกับเครื่องแม่ข่ายที่เปิดค้างอยู่ (open connection) ก่อนที่ Intelligent Load Balancer จะมีการจ่ายงานไปยังเครื่องที่มีการเชื่อมต่อค้างอยู่น้อยที่สุด (least connections) ซึ่งจะทำให้เครื่องสามารถตอบรับการคำขอนั้นได้ดีที่สุดเมื่อเทียบกับเครื่องอื่นๆในขณะเดียวกัน นอกจากนี้ผู้ดูแลระบบสามารถที่จะกำหนดน้ำหนักการจ่ายงานไปให้เครื่องแต่ละเครื่องได้เช่นกัน เพื่อให้เครื่องแต่ละเครื่องได้รับงานมาอย่างเหมาะสมกับความสามารถของเครื่องนั้นๆได้เช่นเดียวกับวิธี Round Robin อีกด้วย

2.3.3 อัลกอริธึม Response Time

เป็นการเลือกกระจายงานไปยังเครื่องที่สามารถตอบสนองต่อ Intelligent Load Balancer ได้เร็วที่สุดเมื่อเปรียบเทียบกับเครื่องอื่นๆในขณะนั้น ทำให้ได้ข้อดีเช่นเดียวกันกับวิธี least connection คือจะได้เครื่องที่ทำงานได้ดีที่สุดในขณะนั้นมาทำการตอบรับคำขอของผู้ใช้งาน นอกเหนือจากวิธีการที่ได้กล่าวถึงมาแล้วข้างต้น ยังมีวิธีการกระจายงานอีกหลากหลายวิธีซึ่งอาจแตกต่างกันไปตามผู้ผลิตอุปกรณ์ต่างๆจะจัดเตรียมไว้ให้ เช่นการวัดจากการใช้งานหน่วยประมวลผลกลาง (CPU utilization) หรือค่าทางสถิติอื่นๆที่วัดได้จากเครื่องแม่ข่ายซึ่งอาจเรียกรวมๆว่า resource-based scheduling นอกจากนี้ในบางครั้งยังมีการใช้เทคนิคหลายๆแบบผสมผสานกันเพื่อให้ได้ผลลัพท์ที่ดีที่สุดอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการศึกษา

3.1 วิธีการศึกษา

จากข้อมูลที่ได้ทำการศึกษาและรวบรวมมาจากแหล่งต่างๆพบว่า แนวทางการออกแบบระบบ web server แต่ละแบบต่างก็มีข้อดีข้อเสียที่แตกต่างกันไป การใช้เครื่องแม่ข่ายเพียงเครื่องเดียวมีข้อจำกัดในแง่ของความสามารถในการทำงานอย่างต่อเนื่องเพราะหากเครื่องแม่ข่ายเกิดมีปัญหาจนไม่สามารถทำงานต่อได้ นั่นก็หมายความว่าระบบโดยรวมจะต้องหยุดทำงานไปโดยปริยาย

การทำงานแบบใช้เครื่องแม่ข่ายสองตัวทำงานแบบ clustering เป็นทางเลือกที่ดีในแง่ของการรองรับการทำงานได้อย่างต่อเนื่องของระบบ อย่างไรก็ดีเนื่องจากในช่วงเวลาใดเวลาหนึ่งจะมีเครื่องแม่ข่ายทำงานอยู่เพียงเครื่องเดียว และอีกเครื่องหนึ่งเป็นเครื่องที่อยู่ในสถานภาพเตรียมพร้อม (stand-by) เท่านั้น วิธีนี้จึงเป็นวิธีที่มีการลงทุนในแง่ของอุปกรณ์ค่อนข้างสูง อีกทั้งยังไม่สามารถรองรับการขยายความสามารถของระบบได้ดีเท่าที่ควรด้วย

เนื่องจากการทำออกแบบระบบด้วยวิธี Round Robin DNS และวิธีการใช้ Intelligent Load Balancer ร่วมกับเครื่องแม่ข่ายหลายๆตัวที่ทำงานร่วมกันแบบ Load Balance เป็นวิธีที่มีความน่าสนใจและน่าจะมีความสามารถในการขยายประสิทธิภาพ รวมทั้งมีความสามารถในการทำงานอย่างต่อเนื่องได้ดีที่สุด แต่ต่างกันที่วิธีการใช้ Intelligent Load Balancer จะมีข้อได้เปรียบกว่าเนื่องจากอุปกรณ์ Intelligent Load Balancer จะสามารถช่วยให้การกระจายงานไปยังเครื่องแม่ข่ายแต่ละเครื่องมีประสิทธิภาพสูงขึ้น อีกทั้งยังช่วยป้องกันไม่ให้เครื่องใดเครื่องหนึ่งรับงานมากเกินไปจนไม่สามารถทำงานต่อได้ (system overload)

ประกอบกับข้อจำกัดในแง่ของระยะเวลาดำเนินการศึกษา รวมทั้งอุปกรณ์ที่จำเป็นต้องนำมาใช้ในการทดสอบซึ่งมีราคาสูงและมีปริมาณจำกัด ดังนั้นในการศึกษาครั้งนี้ จึงได้เลือกทำการทดสอบสมมุติฐานการออกแบบระบบในรูปแบบที่สี่ คือวิธีการกระจายงานให้หลายเครื่องแม่ข่ายหลายเครื่องทำงานร่วมกันด้วยอุปกรณ์พิเศษ (Intelligent Load Balancing) โดยมีวัตถุประสงค์ในการดำเนินการศึกษาดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วัตถุประสงค์ในการทดสอบ

1) เพื่อทำการทดสอบสมมติฐานว่า การใช้งาน Intelligent Load Balancer ร่วมกับเครื่องแม่ข่ายที่ให้บริการ web server จะช่วยให้ระบบ web server มีความสามารถมากขึ้นและเป็นสัดส่วนกับจำนวนเครื่องแม่ข่ายที่ใช้ในการทดสอบ

3.3 ขั้นตอนการทำการศึกษา

ในการศึกษาจะแบ่งวิธีการศึกษาออกเป็น 3 ขั้นตอนหลักคือ

1) ขั้นตอนการออกแบบและติดตั้งระบบเพื่อเตรียมพร้อมทำการทดสอบ
2) ขั้นตอนการดำเนินการทดสอบและวัดผลการทดสอบที่ได้รับ
3) ขั้นตอนการประเมินและสรุปผลการทดลองที่ได้รับ

3.3.1 ขั้นตอนการออกแบบและติดตั้งระบบเพื่อเตรียมพร้อมทำการทดสอบ

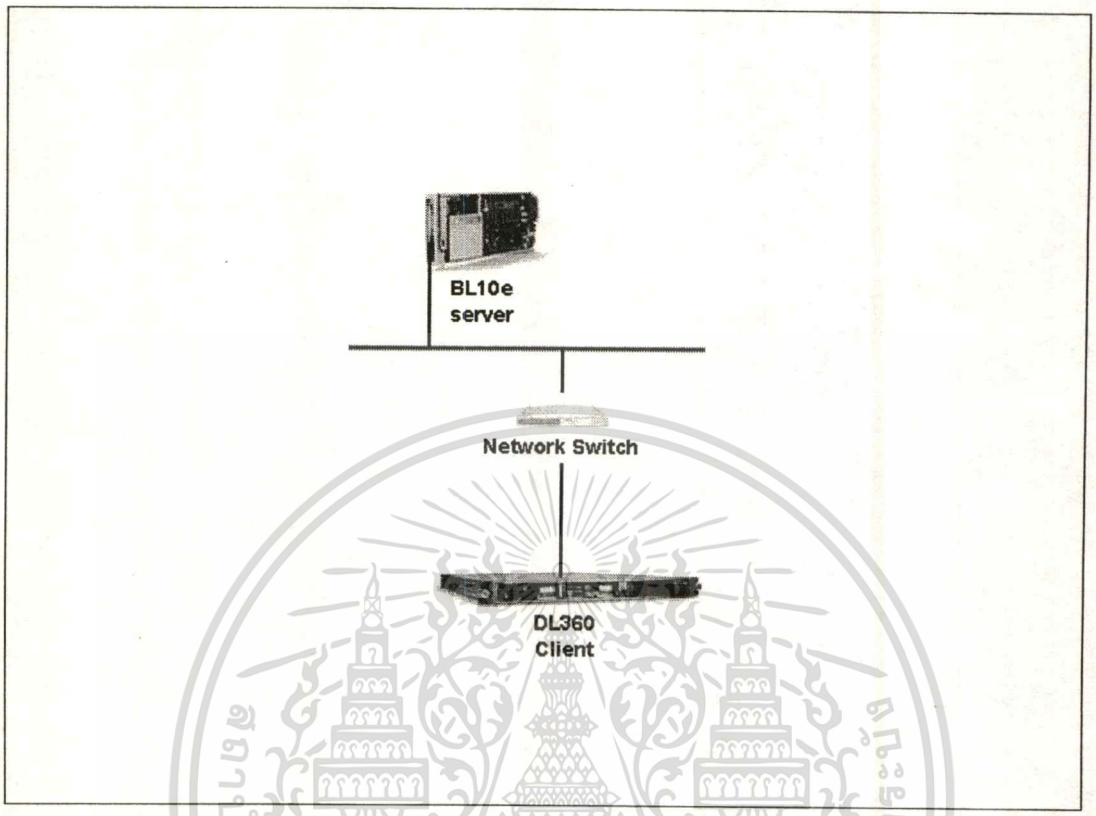
ในการศึกษาจะทำการออกแบบการทดลองและแบ่งระบบที่ทำการทดสอบออกเป็น 4 รูปแบบคือ

1) ระบบที่ใช้เครื่องแม่ข่าย 1 เครื่องทำงานโดยไม่ต้องใช้ Intelligent Load Balancer
2) ระบบที่ใช้เครื่องแม่ข่าย 2 เครื่องทำงานร่วมกันโดยใช้ Intelligent Load Balancer
3) ระบบที่ใช้เครื่องแม่ข่าย 4 เครื่องทำงานร่วมกันโดยใช้ Intelligent Load Balancer
4) ระบบที่ใช้เครื่องแม่ข่าย 6 เครื่องทำงานร่วมกันโดยใช้ Intelligent Load Balancer

รูปแบบที่ 1 ระบบที่ใช้เครื่องแม่ข่าย 1 เครื่องทำงานโดยไม่ต้องใช้ Intelligent Load Balancer

แบบแรกเป็นแบบที่ใช้เครื่องแม่ข่ายเพียงเครื่องเดียวโดยที่เครื่องนี้จะมีหน่วยประมวลผลเพียง 1 เท่านั้น เครื่องแม่ข่ายเครื่องนี้จะเป็นเครื่องที่มีโปรแกรมที่เกี่ยวข้องทั้งหมดติดตั้งอยู่ซึ่งได้แก่ Internet Information Server, Volcano Web Site และ Network Load Balancing Service (เพื่อทำหน้าที่เป็น Intelligent Load Balancer) โดยวัตถุประสงค์หลักของระบบนี้ก็เพื่อใช้เป็นบรรทัดฐานในการวัดเทียบกับระบบอื่นๆ อย่างไรก็ดีรูปแบบนี้ยังไม่มีการทำ Load Balancing เนื่องจากมีเครื่องแม่ข่ายเพียงเครื่องเดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

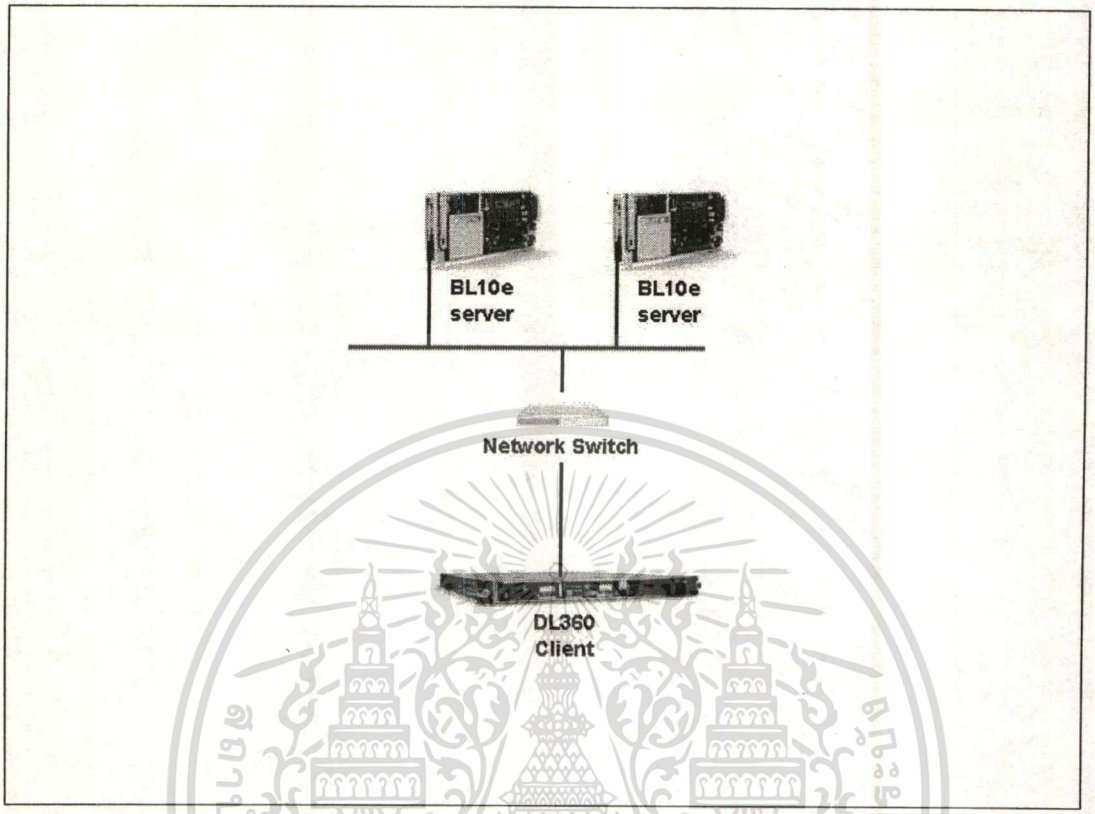


รูปที่ 3.1 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 1 เครื่องต่อเชื่อมเข้ากับระบบเครือข่ายและมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ

รูปแบบที่ 2 ระบบที่ใช้เครื่องแม่ข่าย 2 เครื่องทำงานร่วมกันโดยใช้ Intelligent Load Balancer

เป็นรูปแบบที่ใช้เครื่องแม่ข่าย 2 เครื่องและติดตั้งโปรแกรม Internet Information Server, Volcano Web Site และ Network Load Balancing Service เหมือนกันกับระบบที่มีเครื่องแม่ข่าย 1 เครื่อง อย่างไรก็ดี รูปแบบนี้จะมีการตั้งค่าของโปรแกรม Network Load Balancing Service (เพื่อทำหน้าที่เป็น Intelligent Load Balancer) เพื่อให้เครื่องแม่ข่ายมีการกระจายการร้องขอรับบริการจากเครื่องลูกข่ายไปยังเครื่องแม่ข่ายทั้งสองเครื่องเท่าๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

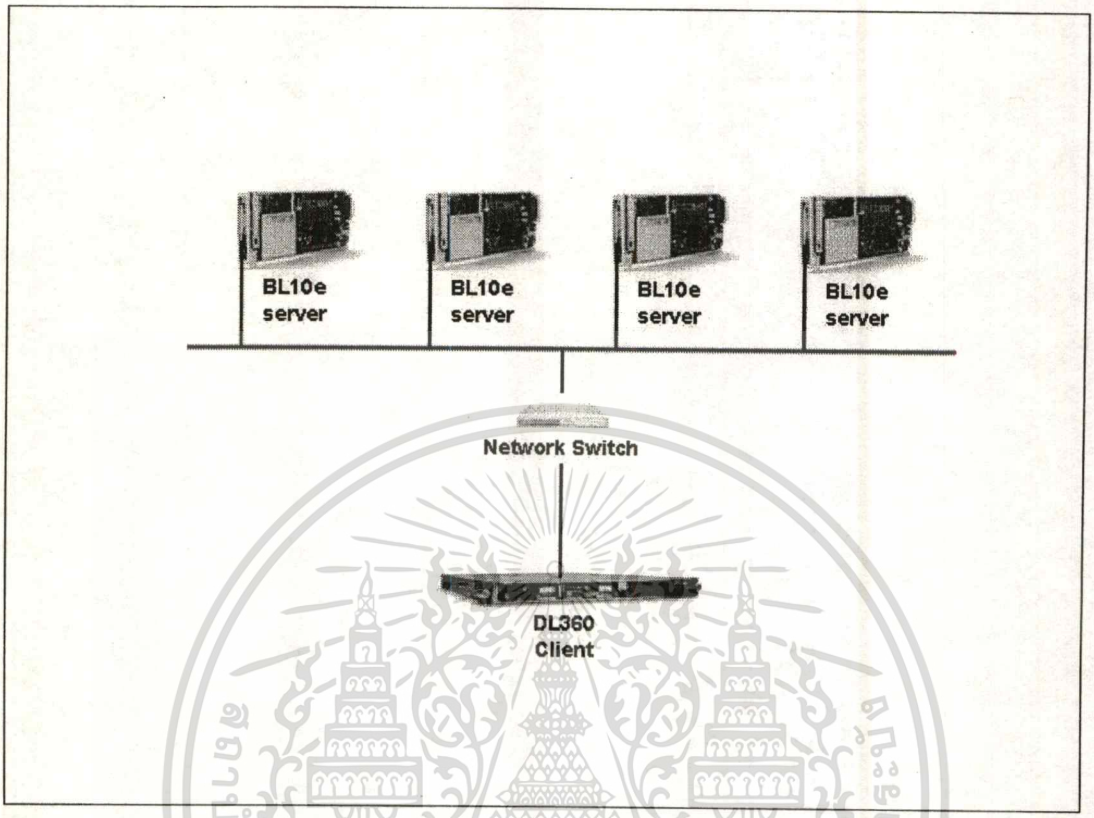


รูปที่ 3.2 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 2 เครื่องต่อเชื่อมเข้ากับระบบเครือข่ายและมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ

รูปแบบที่ 3 ระบบที่ใช้เครื่องแม่ข่าย 4 เครื่องทำงานร่วมกันโดยใช้ Intelligent Load Balancer

เป็นรูปแบบที่ใช้เครื่องแม่ข่าย 4 เครื่องและติดตั้งโปรแกรม Internet Information Server, Volcano Web Site และ Network Load Balancing Service เหมือนกันกับระบบที่มีเครื่องแม่ข่าย 1 เครื่อง เช่นเดียวกับระบบที่มีเครื่องแม่ข่าย 2 เครื่อง รูปแบบนี้จะมีการตั้งค่าของโปรแกรม Network Load Balancing Service (เพื่อทำหน้าที่เป็น Intelligent Load Balancer) เพื่อให้เครื่องแม่ข่ายมีการกระจายการร้องขอรับบริการจากเครื่องลูกข่ายไปยังเครื่องแม่ข่ายทั้งสี่เครื่องเท่าๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

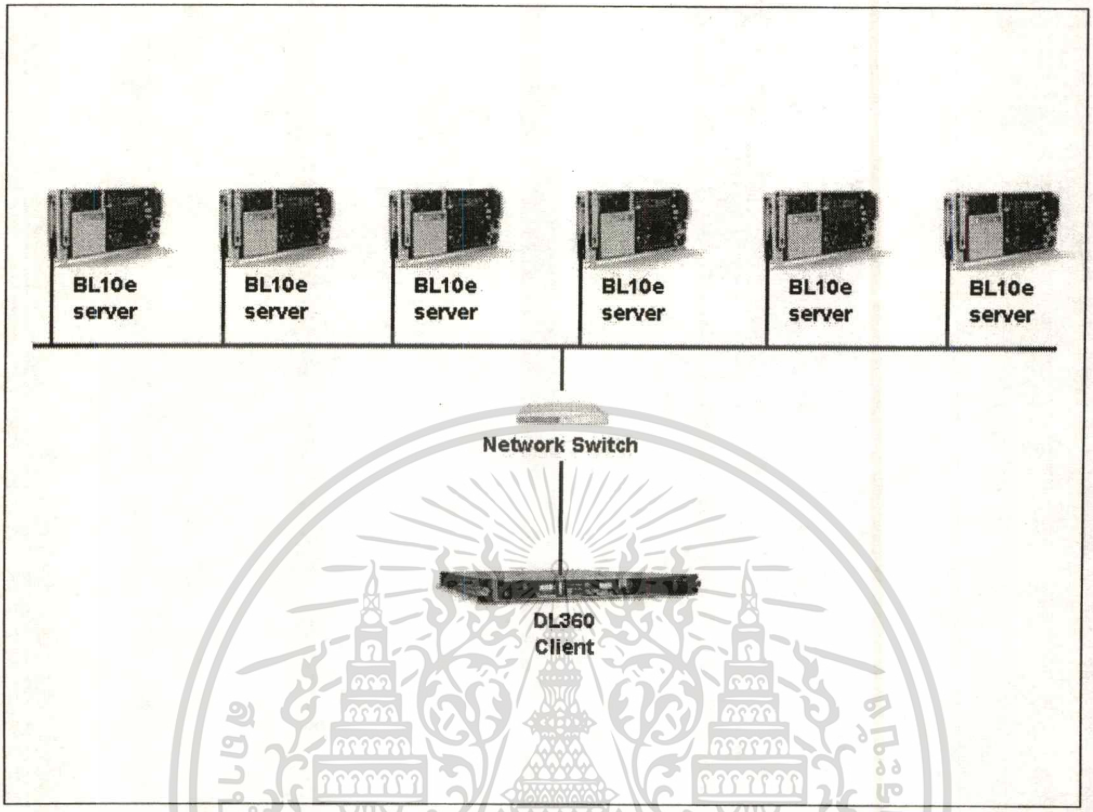


รูปที่ 3.3 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 4 เครื่องต่อเชื่อมเข้ากับระบบเครือข่ายและมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ

รูปแบบที่ 4 ระบบที่ใช้เครื่องแม่ข่าย 6 เครื่องทำงานร่วมกันโดยใช้ Intelligent Load Balancer

เป็นรูปแบบที่ใช้เครื่องแม่ข่าย 6 เครื่องและติดตั้งโปรแกรม Internet Information Server, Volcano Web Site และ Network Load Balancing Service เหมือนกันกับระบบที่มีเครื่องแม่ข่าย 1 เครื่อง เช่นเดียวกับระบบที่มีเครื่องแม่ข่าย 2 เครื่อง รูปแบบนี้จะมีการตั้งค่าของโปรแกรม Network Load Balancing Service (เพื่อทำหน้าที่เป็น Intelligent Load Balancer) เพื่อให้เครื่องแม่ข่ายมีการกระจายการร้องขอรับบริการจากเครื่องลูกข่ายไปยังเครื่องแม่ข่ายทั้งหกเครื่องเท่าๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.4 แสดงรูปแบบการจัดเครื่องแม่ข่ายจำนวน 6 เครื่องต่อเชื่อมเข้ากับระบบเครือข่ายและมีเครื่องลูกข่าย 1 เครื่องเป็นตัวจำลองการทำงานของผู้ใช้ระบบ

ขั้นตอนการติดตั้งระบบที่จะทำการทดสอบ

ขั้นตอนการติดตั้งเครื่องแม่ข่ายและส่วนประกอบอื่น ๆ ที่เกี่ยวข้อง

เครื่องแม่ข่าย

เครื่องแม่ข่ายที่นำมาใช้ในการทดสอบเป็นเครื่องแม่ข่ายรุ่น โปรไลแอนท์ บีแอล 10 อี (ProLiant BL10e) จำนวนหกเครื่อง ซึ่งเครื่องแม่ข่ายรุ่นนี้เป็นเครื่องแม่ข่ายที่ออกแบบมาเพื่อทำงานในลักษณะของเครื่องแม่ข่ายที่ให้บริการ web server โดยเฉพาะและแต่ละเครื่องมีคุณลักษณะดังต่อไปนี้

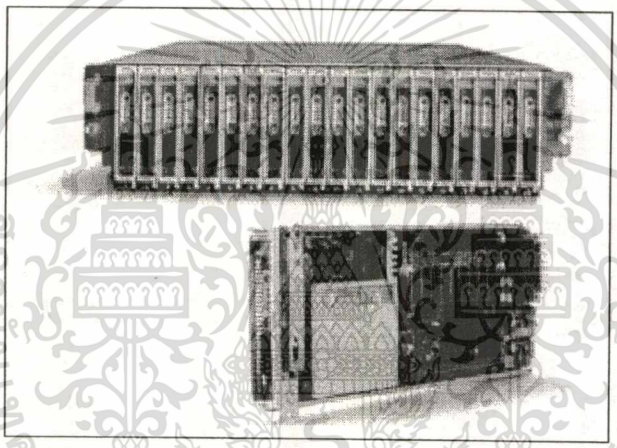
- ProLiant BL10e
- 800-MHz Pentium III Processor with 512KB Cache
- 1024 MB RAM
- 40-GB Ultra ATA/100 4,200 rpm Non-Hot Plug 2.5" hard drive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

- Integrated dual-channel Compaq NC3163 Fast Ethernet NICs 10/100 TX UTP controller

เนื่องจากเครื่องแม่ข่ายรุ่น ProLiant BL10e เป็นเครื่องแม่ข่ายที่ถูกออกแบบมาให้มีขนาดเล็กเป็นพิเศษและสามารถติดตั้งในตู้บรรจุเครื่องแม่ข่าย (server enclosure) ที่ออกแบบมาพิเศษเช่นเดียวกัน เพื่อให้สามารถบรรจุเครื่องแม่ข่ายได้ถึง 20 เครื่องในตู้บรรจุเครื่องแม่ข่ายที่สูงเพียง 3U (U คือหน่วยความสูงของตู้ Rack ซึ่ง 1U = 1.75 นิ้ว) เครื่องแม่ข่ายแต่ละเครื่องจึงไม่มี Floppy Disk Drive หรือ CD-ROM Drive ติดตั้งอยู่ภายในเครื่อง ดังนั้นการติดตั้งระบบปฏิบัติการลงในเครื่องแม่ข่ายแต่ละเครื่องจะต้องกระทำผ่านโปรแกรมพิเศษชื่อ ProLiant Essential – Rapid Deployment Pack



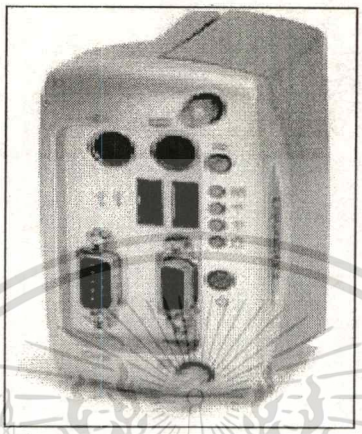
รูปที่ 3.5 เครื่องแม่ข่ายรุ่นโปรไลแอนท์ บีแอล 10 อี (ProLiant BL10e) ภาพล่าง และตู้บรรจุเครื่องแม่ข่าย (server enclosure – รูปบน) ที่ใช้ในการทดสอบ

เนื่องจากเครื่องแม่ข่าย ProLiant BL10e เป็นเครื่องที่ออกแบบมาพิเศษให้มีขนาดเล็กมาก จึงไม่มี VGA port, Mouse Port, และ Keyboard port ในตัวเอง หากผู้ดูแลระบบต้องการจะเข้าไปใช้งานหน้าจอของเครื่องแม่ข่ายแต่ละเครื่องโดยตรงจะต้องใช้วิธีพิเศษซึ่งมีด้วยกัน 2 วิธีคือ

1. ใช้โปรแกรม Web Browser เรียกเข้าไปยังเครื่องแม่ข่ายแต่ละเครื่องเพื่อขอใช้บริการ Terminal Service ซึ่งจะทำการจำลองหน้าจอของเครื่องแม่ข่ายลงมายังเครื่องลูกข่ายที่ใช้งานอยู่ ณ ขณะนั้น โดยการเรียกใช้ก็เพียงแต่ใช้โปรแกรม web browser จากเครื่องลูกข่ายเรียกเข้าไปยัง http://<*servername*>/tsweb (<servername> หมายถึงชื่อเครื่องแม่ข่ายที่ต้องการเรียกใช้เช่น BL10e_01) แล้วทำตามข้อแนะนำบนหน้าจอเพื่อเริ่มการทำงาน)
2. ใช้ Diagnostic Adapter ที่เป็นอุปกรณ์พิเศษที่ให้มากับตู้บรรจุเครื่องแม่ข่าย (Server Enclosure) ต่อเข้าไปยังเครื่องแม่ข่าย ProLiant BL10e ตัวที่ต้องการใช้งาน แล้วจึงต่อสาย VGA, keyboard,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ mouse มาที่ Diagnostic Adapter อีกทีหนึ่ง ด้วยวิธีนี้จะทำให้ผู้ใช้สามารถควบคุมการทำงานเครื่องแม่ข่าย ProLiant BL10e ได้เหมือนการใช้งาน Keyboard และ mouse บนเครื่องแม่ข่ายปกติ



รูปที่ 3.6 Diagnostic Adapter ที่ใช้ต่อ VGA, keyboard, และ mouse เข้ากับเครื่องแม่ข่าย ProLiant BL10e

ระบบเครือข่ายที่ใช้ในการทดสอบ

ระบบทั้งหมดอันได้แก่เครื่องแม่ข่ายและเครื่องลูกข่ายทั้งหมดที่ใช้ในการทดสอบจะเชื่อมต่อกันอยู่บนเครือข่าย Ethernet โดยใช้โปรโตคอล TCP/IP เป็นหลักในการสื่อสารภายในระบบที่ทดสอบโดยใช้ Compaq Netelligent 2824 Dual-Speed Hub (24-port 10/100) เป็นอุปกรณ์สำคัญในการเชื่อมต่อ

โปรแกรมที่ใช้บนเครื่องแม่ข่าย

ในเครื่องแม่ข่ายแต่ละเครื่องที่ใช้ในการทดสอบ จะทำการติดตั้งระบบปฏิบัติการและโปรแกรมต่างๆดังต่อไปนี้

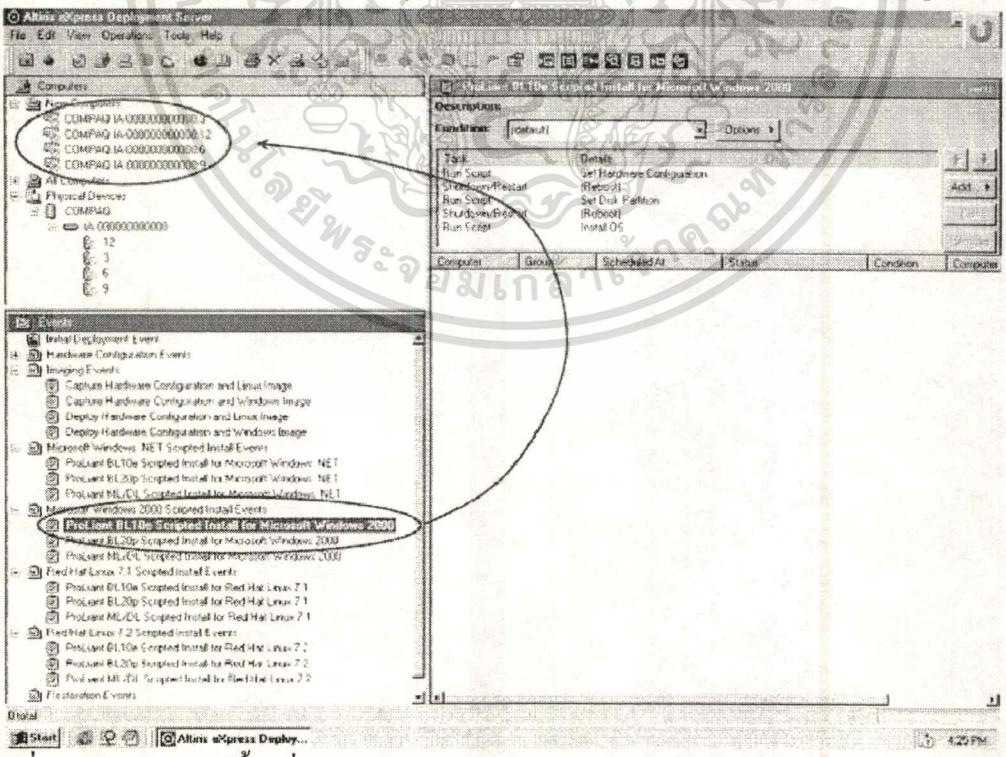
- ระบบปฏิบัติการ Windows 2000 Advance Server
- โปรแกรม Internet Information Server ซึ่งทำหน้าที่เป็น Web Server
- โปรแกรม Network Load Balancing Service เพื่อทำหน้าที่เป็นตัวกระจายการเรียกใช้งานที่ส่งมาจากเครื่องลูกข่ายไปยังเครื่องแม่ข่ายที่ทำการทดสอบอยู่ตาม algorithm แบบ least connection ซึ่งเป็นแบบมาตรฐานของโปรแกรมนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งระบบปฏิบัติการ Windows 2000 Advance Server และโปรแกรม Internet Information Server

การติดตั้งระบบปฏิบัติการลงในเครื่องแม่ข่ายแต่ละเครื่องจะต้องกระทำผ่านโปรแกรมพิเศษชื่อ ProLiant Essential – Rapid Deployment Pack ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

1. บรรจุเครื่องแม่ข่าย ProLiant BL10e ทั้ง 6 เครื่องลงในตู้บรรจุเครื่องแม่ข่าย
2. ติดตั้งสายเน็ตเวิร์กที่บริเวณด้านหลังของตู้บรรจุเครื่องแม่ข่ายให้ตรงกับ network port ของเครื่องแต่ละเครื่องทั้ง 6 เครื่อง
3. เปิดสวิทช์ไฟฟ้าของตู้บรรจุเครื่องแม่ข่ายและของเครื่องแม่ข่ายทั้ง 6 เครื่อง
4. จากหน้าจอของโปรแกรม ProLiant Essential – Rapid Deployment Pack (ซึ่งถูกติดตั้งไว้ก่อนหน้าทำการทดลองเพื่อใช้ในการติดตั้ง ProLiant BL10e โดยเฉพาะ) จะมองเห็นเครื่องแม่ข่ายทั้ง 6 เครื่องแสดงอยู่บนหน้าจอด้านมุมซ้ายบน โดยแต่ละเครื่องจะแสดงสถานะภาพว่ายังไม่มีระบบปฏิบัติการติดตั้งอยู่
5. จากหน้าจอทางด้านล่างของโปรแกรม ให้เลือกที่ "ProLiant BL10e Scripted Install for Microsoft Windows 2000" แล้วกดเม้าส์ที่ปุ่มซ้ายค้างไว้ จากนั้นลาก icon ดังกล่าวไปแปะที่เครื่องแม่ข่ายเครื่องที่หนึ่ง ที่บริเวณมุมซ้ายบนของจอแล้วปล่อยเมาส์ (Drag-and-Drop)



รูปที่ 3.7 แสดงการติดตั้งเครื่องแม่ข่าย ProLiant BL10e ผ่านทางโปรแกรม ProLiant Essential – Rapid Deployment Pack

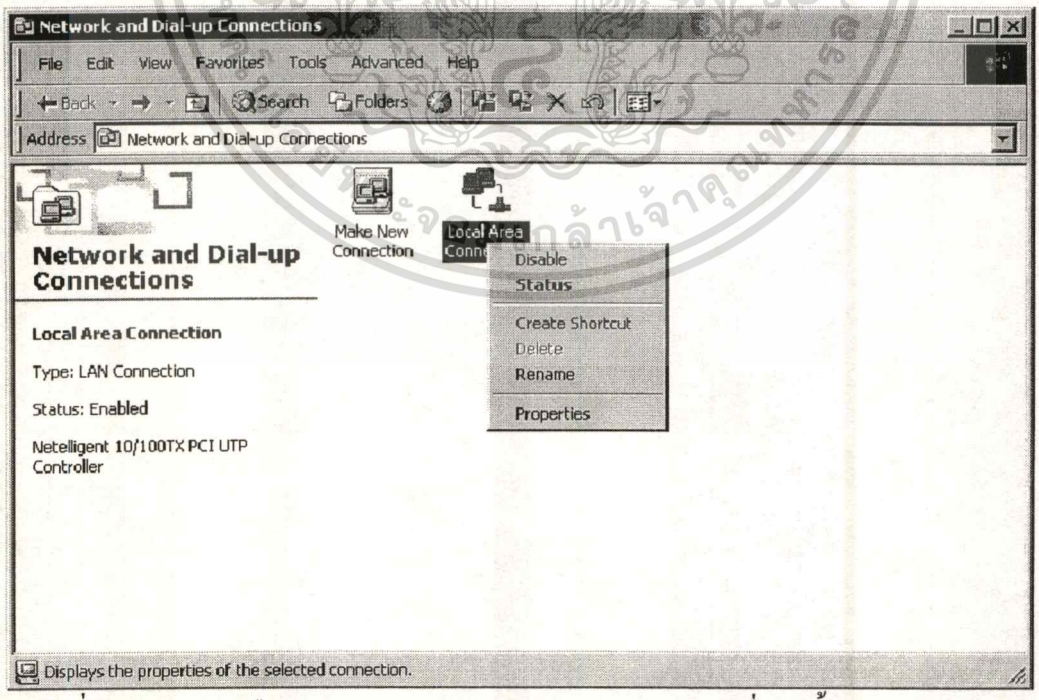
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. โปรแกรมจะทำการติดตั้งระบบปฏิบัติการ Windows 2000 Advance Server ให้กับเครื่องแม่ข่ายดังกล่าวพร้อมทั้งโปรแกรม Internet Information Server 5.0 ซึ่งเป็นหนึ่งในโปรแกรมมาตรฐานที่ให้มาพร้อมกับ Windows 2000 Advance Server โดยอัตโนมัติ
7. ทำขั้นตอนที่ 5 และ 6 ซ้ำสำหรับเครื่องแม่ข่ายเครื่องที่ 2-6 เพื่อติดตั้งระบบปฏิบัติการและโปรแกรม Internet Information Server 5.0 ในลักษณะเดียวกัน

การติดตั้งโปรแกรม Network Load Balancing Service บนเครื่องแม่ข่าย

ในการติดตั้งโปรแกรม Network Load Balancing Service จะใช้วิธีการต่อ Diagnostic Adapter เข้ากับ ProLiant BL10e โดยตรงดังที่ได้อธิบายไว้ข้างต้นแล้วเพื่อความสะดวกในการทำงาน โดยขั้นตอนการติดตั้งมีดังต่อไปนี้

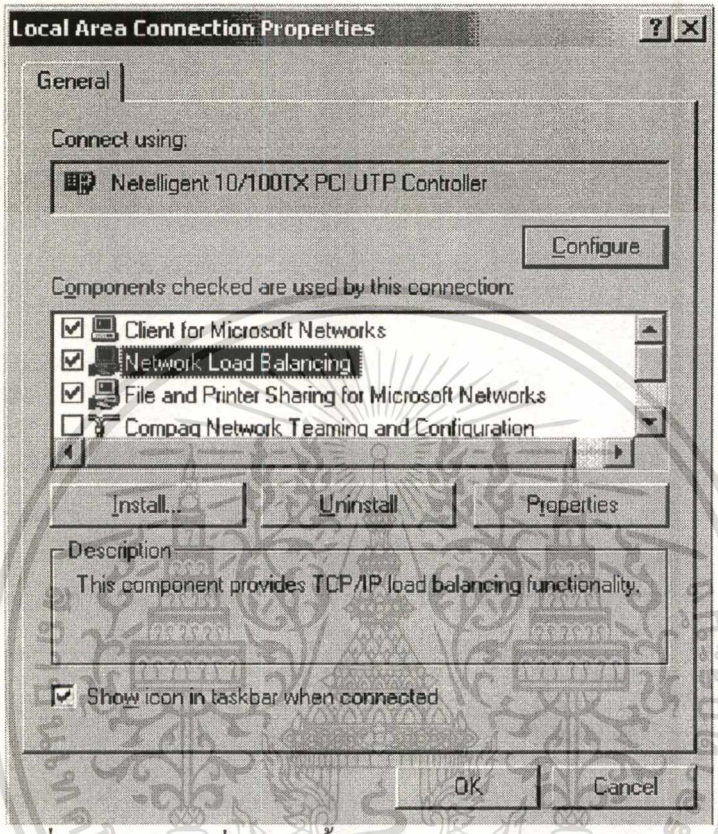
1. ต่อ Diagnostic Adapter รวมทั้งสาย VGA, keyboard, และ mouse เข้ากับเครื่องแม่ข่าย ProLiant BL10e จากนั้นจะสามารถควบคุมการทำงานของเครื่องได้เหมือนเครื่องแม่ข่ายปกติ
2. ทำการติดตั้งโปรแกรม Network Load Balancing Service ใน Windows 2000 Advance Server โดยเปิดไปที่เมนู Start --> Control Panel --> Network and Dial-up Connections
3. เมื่อหน้าจอ Network and Dial-up Connections เปิดขึ้นมา ให้คลิ๊กเมาส์ขวาที่ไอคอน Local Area Connection แล้วเลือก Properties



รูปที่ 3.8 แสดงการเลือก Properties ของ Local Area Connection เพื่อติดตั้ง Network Load Balancing Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในหน้าจอ Local Area Connection Properties ให้เช็คเครื่องหมายถูกที่หน้าหัวข้อ Network Load Balancing แล้วกดปุ่ม Properties



รูปที่ 3.9 แสดงการเริ่มต้นติดตั้ง Network Load Balancing Service

5. ทำการกำหนดค่าต่างๆดังต่อไปนี้

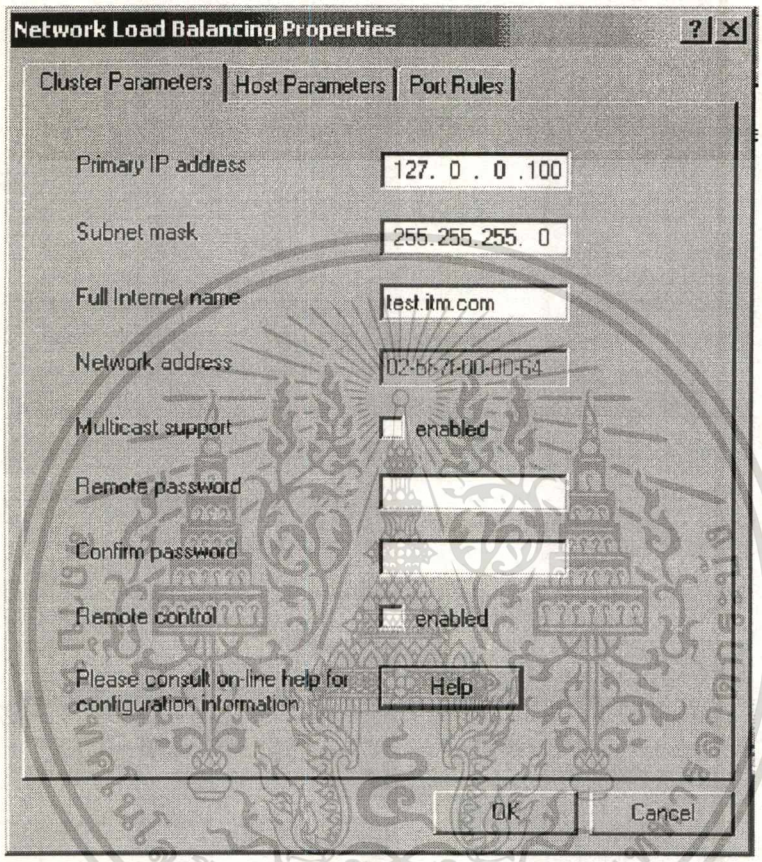
Cluster	*Parameters*
Primary IP address:	192.168.0.100
Subnet mask:	255.255.255.0
Full Internet name:	test.itm.com
Host	*Parameters*
Dedicated IP address:	192.168.0.1
Subnet mask:	255.255.255.0

Port Rules

(ไม่จำเป็นต้องทำการเปลี่ยนแปลงค่าใด ๆ)

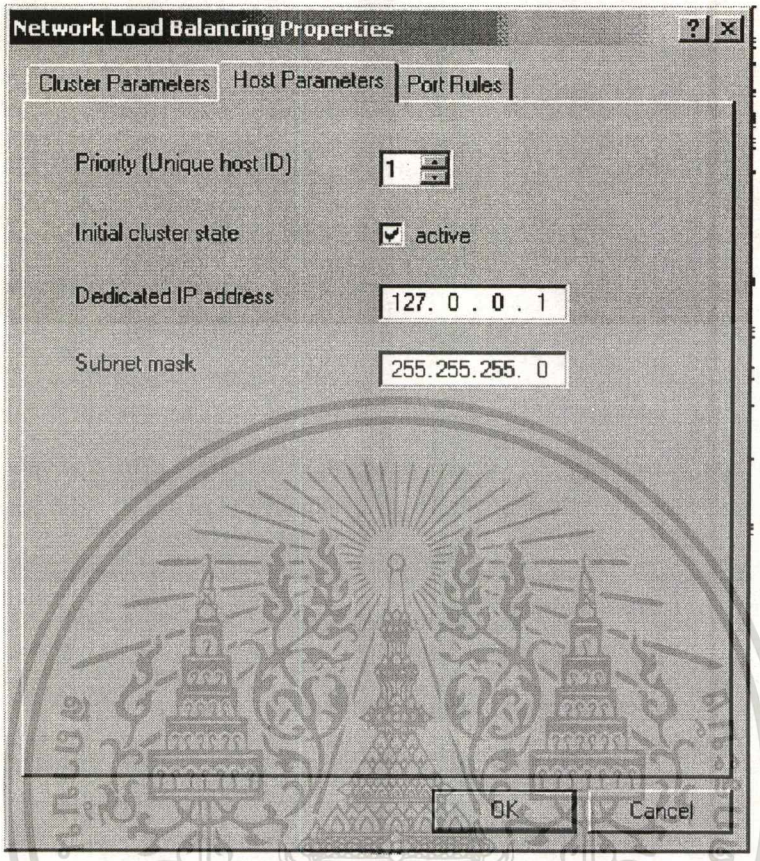
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นกดปุ่ม OK สองครั้งเพื่อจบการกำหนดค่าของ Network Load Balancing ในเครื่องแม่ข่ายเครื่องที่ 1



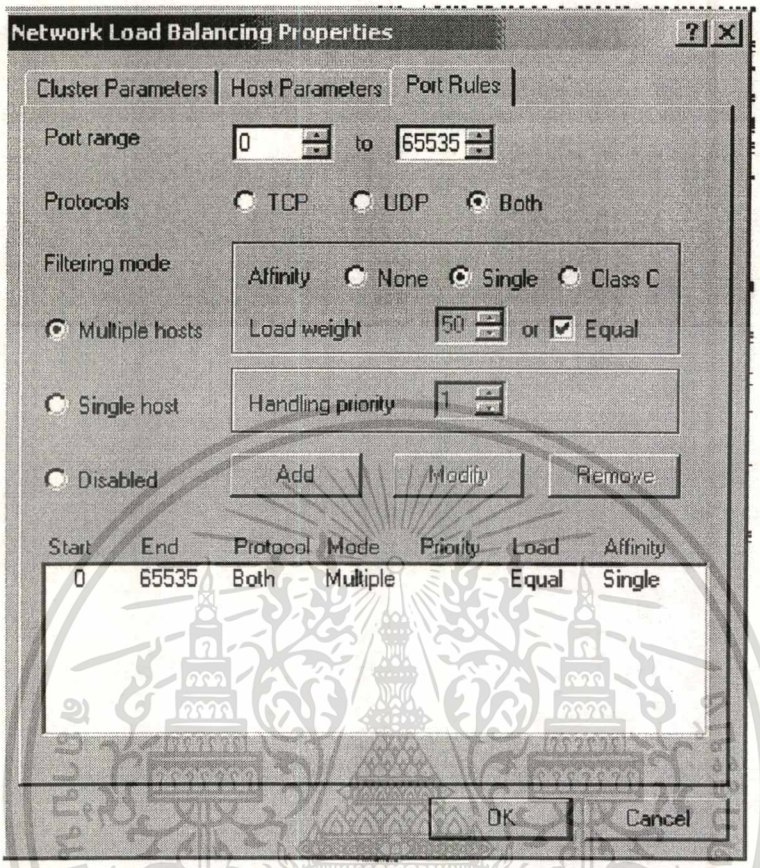
รูปที่ 3.10 แสดงการกำหนดค่าต่างๆในส่วนของ Cluster Parameters ของ Network Load Balancing Service ในเครื่องแม่ข่ายเครื่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงการกำหนดค่าต่างๆในส่วนของ Host Parameters ของ Network Load Balancing Service ในเครื่องแม่ข่ายเครื่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงการกำหนดค่าต่างๆในส่วนของ Port Rules ของ Network Load Balancing Service ในเครื่องแม่ข่ายเครื่องที่ 1

6. ทำการกำหนดค่าการใช้งาน Network Load Balancing Service ในเครื่องแม่ข่ายทั้ง 5 เครื่องที่เหลือตามวิธีทำข้อที่ 1-5 โดยมีการเปลี่ยนแปลงค่าของ Dedicated IP address ของแต่ละเครื่องเป็น 192.168.0.2 – 192.168.0.6 ตามลำดับ

การติดตั้งเว็บไซต์ตัวอย่าง Volcano Coffee บนเครื่องแม่ข่ายเพื่อใช้ในการทดสอบ

Volcano Coffee เป็นเว็บไซต์ตัวอย่างที่บริษัท Microsoft Corporation จัดทำไว้บนเว็บไซต์ของบริษัทเพื่อให้ผู้ใช้สามารถดาวน์โหลดไปใช้ได้ฟรี ไฟล์ที่สามารถดาวน์โหลดได้จาก web คือ VCTurbo.exe ซึ่งเมื่อผู้ใช้ดับเบิ้ลคลิ้กที่ไฟล์ดังกล่าวแล้ว โปรแกรมจะทำการติดตั้ง Volcano Coffee ลงไปให้บนเครื่องโดยอัตโนมัติภายใต้โฟล์เดอร์ \inetpub\wwwroot\ ซึ่งเป็นที่เก็บข้อมูลเว็บไซต์หลักที่โปรแกรม Internet Information Server 5.0ใช้ในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้งเครื่องลูกข่ายและส่วนประกอบอื่นๆที่เกี่ยวข้อง

เครื่องลูกข่าย

เครื่องลูกข่ายที่นำมาใช้ในการทดสอบเป็นเครื่องแม่ข่ายรุ่น โปรไลน์แอนท์ ดีแอล 360 (ProLiant DL360) มาใช้ทำหน้าที่เป็นเครื่องลูกข่ายในการทดสอบแทนที่จะใช้เครื่อง Personal Computer (PC) ทั่วไป ทั้งนี้ก็เพื่อหลีกเลี่ยงไม่ให้ประสิทธิภาพของเครื่องลูกข่าย กลายมาเป็นปัญหาคอขวด (bottleneck) ของการทดสอบ โดยเครื่องโปรไลน์แอนท์ ดีแอล 360 ที่ใช้มีคุณลักษณะดังต่อไปนี้

- ProLiant DL360
- 1.4GHz Pentium III Processor with 512KB Cache
- 128 MB RAM
- 18.2GB Wide Ultra3 SCSI 10,000 rpm Hot Plug hard drive
- Integrated dual-channel Compaq NC3163 Fast Ethernet NICs 10/100 TX UTP controller

โปรแกรมที่ใช้บนเครื่องลูกข่าย

ในเครื่องแม่ลูกข่ายที่ใช้ในการทดสอบ จะทำการติดตั้งระบบปฏิบัติการและโปรแกรมต่างๆดังต่อไปนี้

- ระบบปฏิบัติการ Windows 2000 Server
- โปรแกรม iNetMonitor ที่ทำหน้าที่ผู้จำลองลักษณะการใช้งาน web site ที่จะส่งเข้าไปเรียกใช้งานที่ตัวเครื่องแม่ข่าย

ขั้นตอนการติดตั้งระบบปฏิบัติการ Windows 2000 Server

การติดตั้งระบบปฏิบัติการ Windows 2000 Server บนเครื่องลูกข่ายกระทำโดยแบ่งเป็น 2 ขั้นตอนดังต่อไปนี้

1. การติดตั้งเครื่องขั้นต้นก่อนติดตั้งระบบปฏิบัติการ เป็นการเตรียมความพร้อมของเครื่องและตั้งค่าเบื้องต้นของอุปกรณ์ต่างๆเช่น Bios และ System Partition โดยในการติดตั้งจะทำการบูตเครื่องจากแผ่นซีดี Smart Start Utility ซึ่งให้มาพร้อมกับเครื่อง ProLiant DL360 ที่ใช้ และปล่อยให้โปรแกรมดำเนินการติดตั้งอุปกรณ์ต่างๆโดยอัตโนมัติ ซึ่งเป็นวิธีการที่แนะนำโดยบริษัทผู้ผลิตเครื่อง
2. การติดตั้งระบบปฏิบัติการ Windows 2000 Server ทำโดยการบูตเครื่องหลังจากที่ Smart Start Utility ติดตั้งเสร็จแล้ว ในขั้นตอนการติดตั้งจะใช้ค่ามาตรฐานที่ Windows แนะนำทั้งสิ้น และมีการกำหนดค่า IP Address เป็น 192.168.0.11 และ Subnet mask เป็น 255.255.255.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้งโปรแกรม iNetMonitor

InetMonitor 3.0 (InetMon) เป็นโปรแกรมที่บริษัท Microsoft Corporation พัฒนาขึ้นเพื่อใช้เป็นเครื่องมือในการตรวจสอบประสิทธิภาพและจุดที่เป็นปัญหาคอขวดของระบบ โดยตัวโปรแกรมจะทำการสร้าง request ส่งไปยังเครื่องแม่ข่ายเพื่อขอใช้บริการ web ตามที่กำหนดไว้ใน test script ซึ่งในกรณีของการทดสอบกับเว็บตัวอย่าง Volcano Coffee นี้สามารถทำการดาวน์โหลดทั้งตัวโปรแกรม iNetMonitor และตัวอย่าง test script ได้ฟรีจากเว็บไซต์ของไมโครซอฟท์โดยตรง นอกจากนี้โปรแกรมยังสามารถตรวจวัดประสิทธิภาพการทำงานของระบบในส่วนที่สำคัญๆในระหว่างที่ทำการทดสอบและสร้างเป็นรายงานออกมาได้อีกด้วย

โปรแกรม InetMonotor ที่ทำการดาวน์โหลดมาจากเว็บของไมโครซอฟท์ (inetmon30.exe) สามารถทำการติดตั้งได้ง่ายเพียงแค่ดับเบิ้ลคลิ้กที่ไฟล์แล้วทำการติดตั้งโดยใช้ค่าพื้นฐานที่กำหนดเท่านั้น

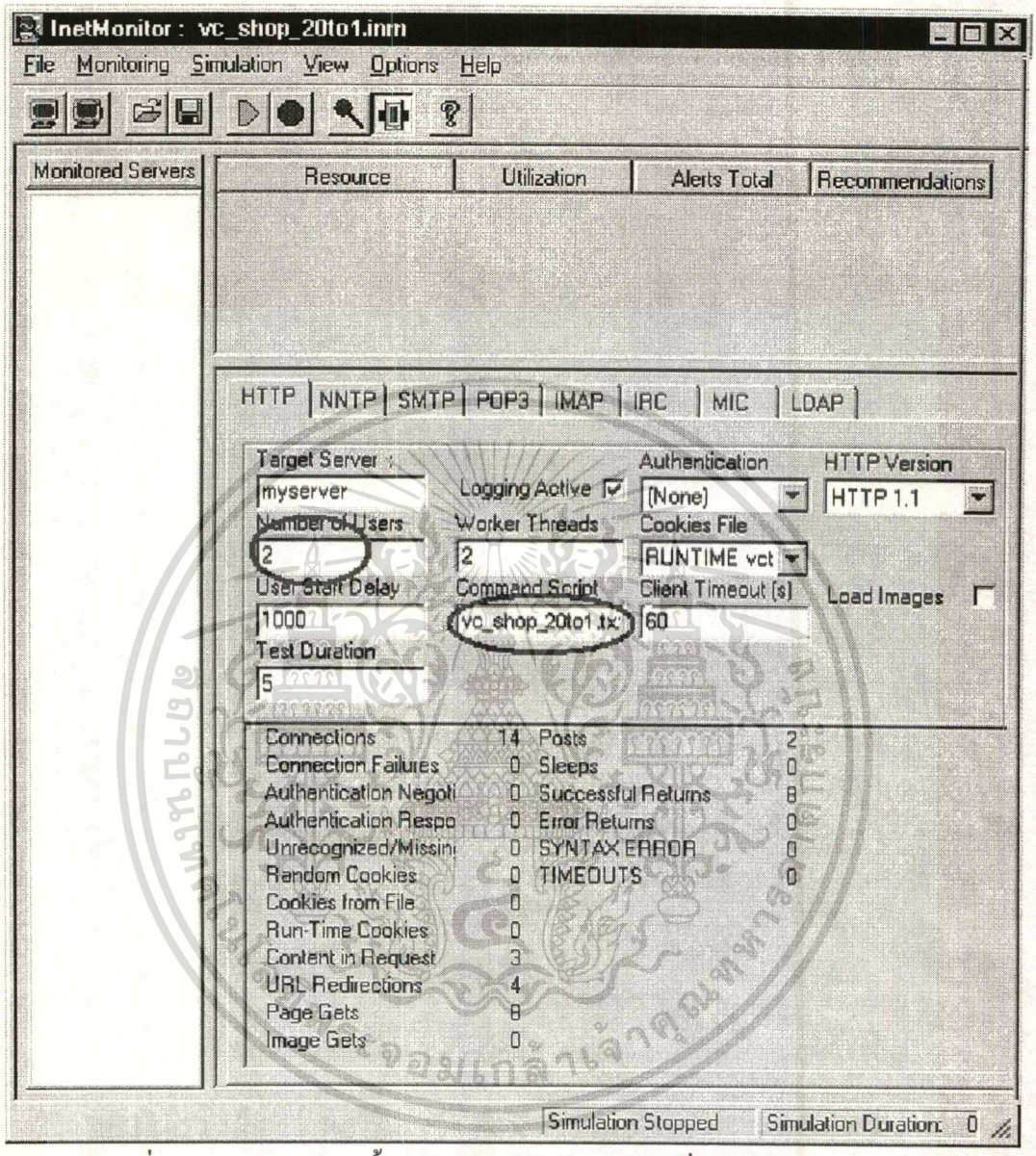
3.3.2 ขั้นตอนการดำเนินการทดสอบและวัดผลการทดสอบที่ได้รับ

การจำลองปริมาณงานการใช้ระบบพาณิชย์อิเล็คโทรนิกที่จะเกิดขึ้นกับระบบจะใช้โปรแกรมชื่อ InetMonitor ซึ่งทำหน้าที่จำลองการใช้งาน Web Site ของผู้ใช้งานผ่านเครื่องลูกข่ายที่ต่ออยู่ในระบบเช่นเดียวกัน โดยเครื่องลูกข่ายจะทำหน้าที่เปรียบเสมือนเป็นผู้ใช้งานหลายๆคนที่มีลักษณะการใช้งานเหมือนๆกันและเข้าเรียกใช้งานเครื่องแม่ข่ายในเวลาเดียวกัน ในการทดสอบจริงจะใช้โปรแกรมดังกล่าวเพื่อจำลองการเรียกใช้งานเข้าไปที่เครื่องแม่ข่ายและเพิ่มจำนวนผู้ใช้ขึ้นเรื่อยๆทีละ 50 คนและบันทึกค่า successful return และ timeout request ไปเรื่อยๆจนกระทั่งความสามารถระบบถึงจุดอิ่มตัวซึ่งสามารถสังเกตได้จากการที่อัตราการเพิ่มของ successful return จะลดลงอย่างเห็นได้ชัดรวมทั้งปริมาณของ timeout request ก็จะเพิ่มขึ้นด้วย จากนั้นจึงนำค่าที่บันทึกได้ไปสร้างแผนภูมิเพื่อหาค่าประสิทธิภาพของระบบ ณ จุดอิ่มตัวต่อไป

การทดสอบและวัดผลจะแบ่งเป็นขั้นตอนดังต่อไปนี้

1. เริ่มดำเนินการทดลองโดยใช้รูปแบบการทดลองที่มีเครื่องแม่ข่ายเพียง 1 เครื่องก่อน ส่วนเครื่องแม่ข่ายที่เหลือให้ปิดเครื่องไว้ก่อน
2. ขั้นตอนการจำลองการเรียกใช้งาน web ส่งไปยังเครื่องแม่ข่าย
    a. ทำการปรับตั้งค่าของโปรแกรม iNetMonitor ให้มี user เริ่มต้นที่ 50 user และ Client timeout (s) เป็น 60 วินาที และระบุ IP address ของเครื่องแม่ข่าย (virtual IP address ที่กำหนดไว้ใน Network Load Balancing Service 192.168.0.100) ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงการปรับตั้งค่าโปรแกรม iNetMonitor เพื่อใช้ในการทดสอบ

b. สั่งให้โปรแกรมเริ่มทำการสร้าง request ส่งไปยังเครื่องแม่ข่าย และเริ่มจับเวลาการทดสอบ

c. เมื่อครบกำหนดเวลา 5 นาที (300 วินาที) ให้หยุดโปรแกรม InetMonitor แล้วทำการบันทึกค่า successful return และ timeout request

d. เริ่มต้นสั่งให้โปรแกรมทำงานอีกครั้งโดยเพิ่มจำนวน user ขึ้นอีก 50 user แล้วทำการทดลองซ้ำตามข้อ 2 จนกระทั่งระบบถึงจุดอิ่มตัว ซึ่งสังเกตได้จากอัตราการเพิ่มของ successful return ที่ลดลง หรือจำนวน timeout request ที่เพิ่มขึ้นอย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

e. ทำการทดลองต่อจากจุดที่คาดว่าน่าจะเป็นจุดอิ่มตัวของระบบต่อไปอีกอย่างน้อย 3 จุด เพื่อเป็นการยืนยันผลการทดลอง

f. คำนวณค่า ASP/second ซึ่งระบบสามารถทำงานได้ในแต่ละจุด ซึ่งคำนวณได้จาก

page/second = Successful Returns/300

(300 คือระยะเวลาที่ใช้ในการทดสอบ = 5 นาที = 300 วินาที)

3. ทำการทดสอบในข้อที่ 1-5 อีกครั้งแต่เปลี่ยนเป็นใช้เครื่องแม่ข่าย 2, 4, และ 6 เครื่องตามลำดับ แล้วทำการบันทึกผลที่ได้

3.3.3 ขั้นตอนการประเมินและสรุปผลการทดลองที่ได้รับ

การประเมินระดับความสามารถของระบบที่เพิ่มขึ้นจะวัดจากปริมาณ successful return ที่ระบบสามารถตอบสนองการขอใช้บริการจากเครื่องลูกข่าย รวมถึงค่า timeout request ที่เกิดขึ้นในระหว่างเวลา 5 นาทีที่ทำการทดสอบ จากนั้นจึงนำค่าที่วัดได้มาคำนวนเป็นค่า ASP/second และ % successful return และ % timeout request ตามลำดับ

ค่าที่วัดและคำนวนได้จะถูกนำมาสร้างเป็นแผนภูมิเส้นเพื่อหาจุดที่ระบบมีประสิทธิภาพสูงที่สุดในการทดสอบแต่ละครั้ง แล้วนำค่าประสิทธิภาพที่วัดได้ ณ จุดนั้นไปสร้างแผนภูมิเส้นเปรียบเทียบกันระหว่างระบบต่างๆเพื่อดูความสามารถในการขยายประสิทธิภาพของระบบเมื่อมีเครื่องแม่ข่ายมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการศึกษาที่ได้

จากการทำการศึกษาตามวิธีที่กำหนดดังได้กล่าวไปแล้วในบทที่ 3 ได้ผลการศึกษาการประเมินความสามารถของระบบดังต่อไปนี้

4.1 ผลการทดสอบประสิทธิภาพการขยายความสามารถในการทำงาน

ในการวัดประสิทธิภาพในการขยายความสามารถของระบบซึ่งมีเครื่องแม่ข่ายจำนวนแตกต่างกัน ค่าที่บันทึกได้คือจำนวนครั้งที่เครื่องแม่ข่ายสามารถตอบกลับมายังเครื่องลูกข่ายได้ ภายในเวลาไม่เกิน 60 วินาที (successful return) จำนวนครั้งที่เครื่องแม่ข่ายไม่สามารถตอบกลับได้ในเวลาที่กำหนด (timeout) แล้วนำมาคำนวนหาค่า ASP/second, % Successful Return, และ % Timeout ซึ่งผลการทดลองที่ได้รับ มีดังต่อไปนี้

4.1.1 ผลการทดลองในระบบที่มีเครื่องแม่ข่าย 1 เครื่อง

ในระบบที่มีเครื่องแม่ข่ายที่ใช้ในการทดลอง 1 เครื่อง เมื่อทำการทดสอบแล้วได้ผลการทดลองดังแสดงในตารางที่ 4.1 เมื่อนำผลการทดลองดังกล่าวมาสร้างเป็นแผนภูมิเส้นแสดงความสัมพันธ์ระหว่าง successful return กับจำนวน user ซึ่งในการทดลองจะพบว่า เมื่อจำนวน user เพิ่มขึ้น ค่า successful return ก็จะเพิ่มขึ้นในอัตราส่วนที่ค่อนข้างคงที่จนกระทั่งมีจำนวน user ที่ใช้ในการทดสอบถึง 700 user และหลังจากนั้นจำนวน successful return ที่วัดได้ก็จะยังคงเพิ่มขึ้นแต่มีอัตราการเพิ่มน้อยลง

เช่นเดียวกัน เมื่อนำค่า timeout request กับจำนวน user มาสร้างเป็นแผนภูมิเส้นจะพบว่าจำนวน timeout request ที่วัดได้จะมีค่าต่ำมากเมื่อเทียบกับจำนวน request ทั้งหมดที่เกิดขึ้นในระบบและค่า timeout request ที่วัดได้นี้จะค่อยๆเพิ่มขึ้นทีละน้อยจนกระทั่งระบบมี user ที่ใช้ในการทดสอบ 700 user และหลังจากนั้นค่า timeout request ที่วัดได้ก็จะเพิ่มขึ้นอย่างรวดเร็วและเห็นได้ชัด

จากลักษณะดังกล่าวแสดงให้เห็นว่า ระบบที่ใช้ในการทดสอบ มีความสามารถในการรองรับการเรียกขอใช้งานระบบจากเครื่องลูกข่ายได้ดีที่สุดเมื่อมีจำนวน user ไม่เกิน 700 user แต่เมื่อมี user มากขึ้นไปกว่านั้น ถึงแม้ว่าระบบจะยังคงรองรับการเรียกขอใช้บริการจากเครื่องลูกข่ายได้ แต่จำนวนเครื่องลูกข่ายที่ไม่สามารถเรียกขอใช้บริการระบบได้ในเวลาที่กำหนด

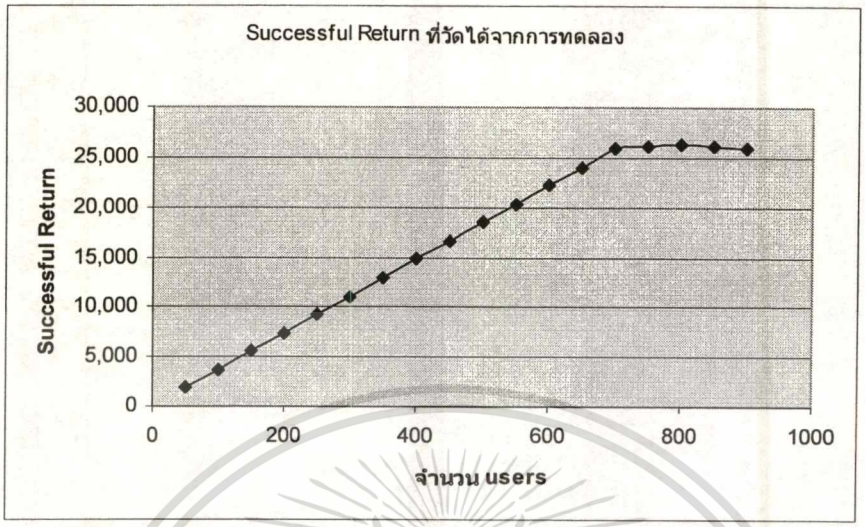
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(timeout request) เพิ่มมากขึ้นเรื่อยๆ ดังนั้นในการทดสอบนี้จึงถือว่า ณ จุดที่ระบบสามารถรองรับ user ได้ 700 user เป็นจุดที่ระบบนี้มีประสิทธิภาพดีที่สุด

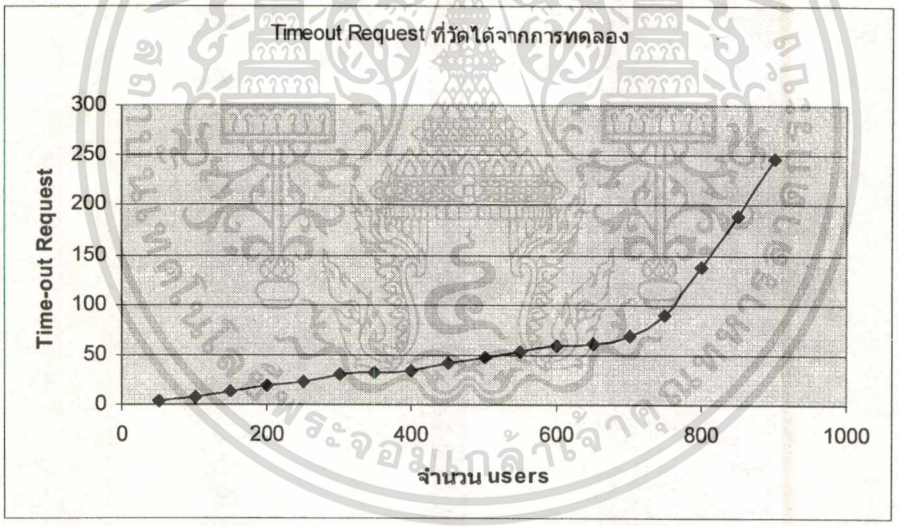
จำนวน users	Total Request	Suc. Return	ASP/sec	Time Out	% Time Out
50	1,856	1,853	6.18	3	0.16%
100	3,708	3,701	12.34	7	0.19%
150	5,566	5,552	18.51	14	0.25%
200	7,425	7,406	24.69	19	0.26%
250	9,276	9,254	30.85	22	0.24%
300	11,134	11,103	37.01	31	0.28%
350	12,991	12,958	43.19	33	0.25%
400	14,849	14,814	49.38	35	0.24%
450	16,709	16,667	55.56	42	0.25%
500	18,564	18,517	61.72	47	0.25%
550	20,419	20,366	67.89	53	0.26%
600	22,278	22,219	74.06	59	0.26%
650	24,133	24,071	80.24	62	0.26%
700	25,991	25,922	86.41	69	0.27%
750	26,243	26,153	87.18	90	0.34%
800	26,401	26,263	87.54	138	0.52%
850	26,316	26,127	87.09	189	0.72%
900	26,252	26,005	86.68	247	0.94%

ตารางที่ 4.1 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 1 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 1 เครื่อง



รูปที่ 4.2 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 1 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดลองในระบบที่มีเครื่องแม่ข่าย 2 เครื่อง

ในระบบที่มีเครื่องแม่ข่ายที่ใช้ในการทดลอง 2 เครื่อง เมื่อทำการทดสอบแล้วได้ผลการทดลองดังแสดงในตารางที่ 4.2 เมื่อนำผลการทดลองดังกล่าวมาสร้างเป็นแผนภูมิเส้นแสดงความสัมพันธ์ระหว่าง successful return กับจำนวน user ซึ่งในการทดลองจะพบว่า เมื่อจำนวน user เพิ่มขึ้น ค่า successful return ก็จะเพิ่มขึ้นในอัตราส่วนที่ค่อนข้างคงที่จนกระทั่งมีจำนวน user ที่ใช้ในการทดสอบถึง 1,350 user และหลังจากนั้นจำนวน successful return ที่วัดได้ก็จะยังคงเพิ่มขึ้นแต่มีอัตราการเพิ่มน้อยลง

เช่นเดียวกัน เมื่อนำค่า timeout request กับจำนวน user มาสร้างเป็นแผนภูมิเส้นจะพบว่าจำนวน timeout request ที่วัดได้จะมีค่าต่ำมากเมื่อเทียบกับจำนวน request ทั้งหมดที่เกิดขึ้นในระบบและค่า timeout request ที่วัดได้นี้จะค่อยๆเพิ่มขึ้นทีละน้อยจนกระทั่งระบบมี user ที่ใช้ในการทดสอบ 1,350 user และหลังจากนั้นค่า timeout request ที่วัดได้ก็จะเพิ่มขึ้นอย่างรวดเร็วและเห็นได้ชัด

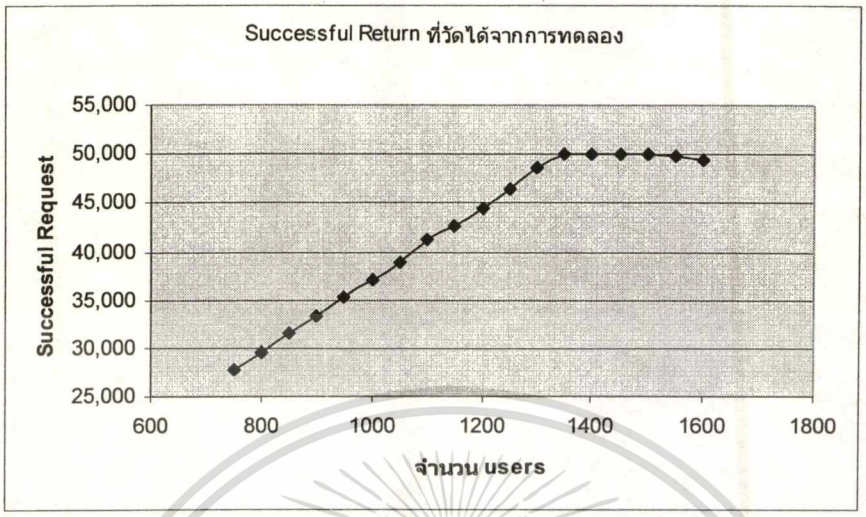
จากลักษณะดังกล่าวแสดงให้เห็นว่า ระบบที่ใช้ในการทดสอบ มีความสามารถในการรองรับการเรียกขอใช้งานระบบจากเครื่องลูกข่ายได้ดีที่สุดเมื่อมีจำนวน user ไม่เกิน 1,350 user แต่เมื่อมี user มากขึ้นไปกว่านั้น ถึงแม้ว่าระบบจะยังคงรองรับการเรียกขอใช้บริการจากเครื่องลูกข่ายได้ แต่จำนวนเครื่องลูกข่ายที่ไม่สามารถเรียกขอใช้บริการระบบได้ในเวลาที่กำหนด (timeout request) เพิ่มมากขึ้นเรื่อยๆ ดังนั้นในการทดสอบนี้จึงถือว่า ณ จุดที่ระบบสามารถรองรับ user ได้ 1,350 user เป็นจุดที่ระบบนี้มีประสิทธิภาพดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

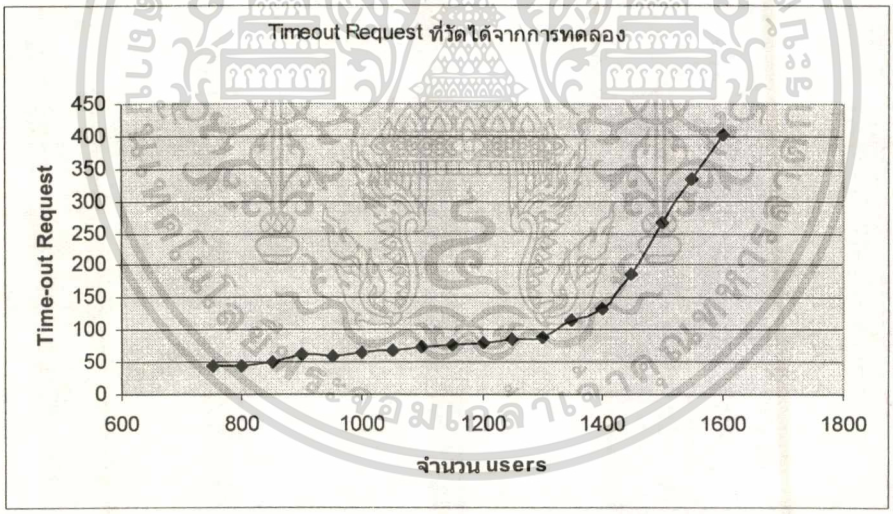
จำนวน users	Total Request	Suc. Return	ASP/sec	Time Out	% Time Out
750	27,828	27,783	92.61	45	0.16%
800	29,692	29,649	98.83	43	0.14%
850	31,575	31,526	105.09	49	0.16%
900	33,430	33,367	111.22	63	0.19%
950	35,305	35,245	117.48	60	0.17%
1000	37,131	37,065	123.55	66	0.18%
1050	38,968	38,900	129.67	68	0.17%
1100	41,439	41,365	137.88	74	0.18%
1150	42,676	42,598	141.99	78	0.18%
1200	44,545	44,466	148.22	79	0.18%
1250	46,623	46,538	155.13	85	0.18%
1300	48,819	48,730	162.43	89	0.18%
1350	50,142	50,026	166.75	116	0.23%
1400	50,148	50,016	166.72	132	0.26%
1450	50,189	50,002	166.67	187	0.37%
1500	50,250	49,983	166.61	267	0.53%
1550	50,185	49,850	166.17	335	0.67%
1600	49,840	49,438	164.79	402	0.81%

ตารางที่ 4.2 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 2 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 2 เครื่อง



รูปที่ 4.4 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 2 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ผลการทดลองในระบบที่มีเครื่องแม่ข่าย 4 เครื่อง

ในระบบที่มีเครื่องแม่ข่ายที่ใช้ในการทดลอง 4 เครื่อง เมื่อทำการทดสอบแล้วได้ผลการทดลองดังแสดงในตารางที่ 4.3 เมื่อนำผลการทดลองดังกล่าวมาสร้างเป็นแผนภูมิเส้นแสดงความสัมพันธ์ระหว่าง successful return กับจำนวน user ซึ่งในการทดลองจะพบว่า เมื่อจำนวน user เพิ่มขึ้น ค่า successful return ก็จะเพิ่มขึ้นในอัตราส่วนที่ค่อนข้างคงที่จนกระทั่งมีจำนวน user ที่ใช้ในการทดสอบถึง 2,900 user และหลังจากนั้นจำนวน successful return ที่วัดได้ก็จะยังคงเพิ่มขึ้นแต่มีอัตราการเพิ่มน้อยลง

เช่นเดียวกัน เมื่อนำค่า timeout request กับจำนวน user มาสร้างเป็นแผนภูมิเส้นจะพบว่าจำนวน timeout request ที่วัดได้จะมีค่าต่ำมากเมื่อเทียบกับจำนวน request ทั้งหมดที่เกิดขึ้นในระบบและค่า timeout request ที่วัดได้นี้จะค่อยๆเพิ่มขึ้นทีละน้อยจนกระทั่งระบบมี user ที่ใช้ในการทดสอบ 2,900 user และหลังจากนั้นค่า timeout request ที่วัดได้ก็จะเพิ่มขึ้นอย่างรวดเร็วและเห็นได้ชัด

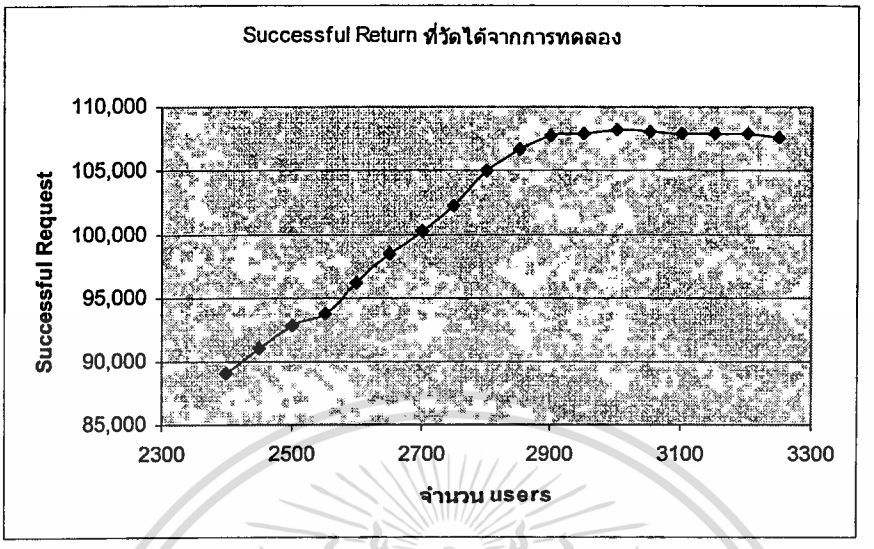
จากลักษณะดังกล่าวแสดงให้เห็นว่า ระบบที่ใช้ในการทดสอบ มีความสามารถในการรองรับการเรียกขอใช้งานระบบจากเครื่องลูกข่ายได้ดีที่สุดเมื่อมีจำนวน user ไม่เกิน 2,900 user แต่เมื่อมี user มากขึ้นไปกว่านั้น ถึงแม้ว่าระบบจะยังคงรองรับการเรียกขอใช้บริการจากเครื่องลูกข่ายได้ แต่จำนวนเครื่องลูกข่ายที่ไม่สามารถเรียกขอใช้บริการระบบได้ในเวลาที่กำหนด (timeout request) เพิ่มมากขึ้นเรื่อยๆ ดังนั้นในการทดสอบนี้จึงถือว่า ณ จุดที่ระบบสามารถรองรับ user ได้ 2,900 user เป็นจุดที่ระบบนี้มีประสิทธิภาพดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

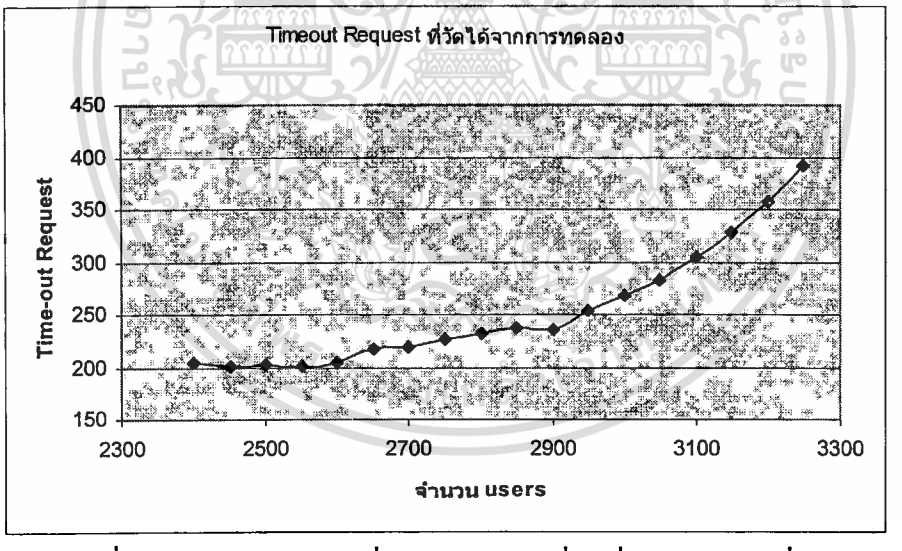
จำนวน users	Total Request	Suc. Return	ASP/sec	Time Out	% Time Out
2400	89,349	89,145	297.15	204	0.23%
2450	91,244	91,043	303.48	201	0.22%
2500	93,119	92,916	309.72	203	0.22%
2550	94,030	93,829	312.76	201	0.21%
2600	96,409	96,204	320.68	205	0.21%
2650	98,700	98,482	328.27	218	0.22%
2700	100,502	100,283	334.28	219	0.22%
2750	102,549	102,322	341.07	227	0.22%
2800	105,259	105,027	350.09	232	0.22%
2850	106,830	106,593	355.31	237	0.22%
2900	108,015	107,780	359.27	235	0.22%
2950	108,135	107,881	359.60	254	0.23%
3000	108,486	108,218	360.73	268	0.25%
3050	108,286	108,003	360.01	283	0.26%
3100	108,195	107,891	359.64	304	0.28%
3150	108,152	107,823	359.41	329	0.30%
3200	108,268	107,910	359.70	358	0.33%
3250	107,917	107,525	358.42	392	0.36%

ตารางที่ 4.3 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 4 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 4 เครื่อง



รูปที่ 4.6 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 4 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 ผลการทดลองในระบบที่มีเครื่องแม่ข่าย 6 เครื่อง

ในระบบที่มีเครื่องแม่ข่ายที่ใช้ในการทดลอง 6 เครื่อง เมื่อทำการทดสอบแล้วได้ผลการทดลองดังแสดงในตารางที่ 4.4 เมื่อนำผลการทดลองดังกล่าวมาสร้างเป็นแผนภูมิเส้นแสดงความสัมพันธ์ระหว่าง successful return กับจำนวน user ซึ่งในการทดลองจะพบว่า เมื่อจำนวน user เพิ่มขึ้น ค่า successful return ก็จะเพิ่มขึ้นในอัตราส่วนที่ค่อนข้างคงที่จนกระทั่งมีจำนวน user ที่ใช้ในการทดสอบถึง 4,200 user และหลังจากนั้นจำนวน successful return ที่วัดได้ก็จะยังคงเพิ่มขึ้นแต่มีอัตราการเพิ่มน้อยลง

เช่นเดียวกัน เมื่อนำค่า timeout request กับจำนวน user มาสร้างเป็นแผนภูมิเส้นจะพบว่าจำนวน timeout request ที่วัดได้จะมีค่าต่ำมากเมื่อเทียบกับจำนวน request ทั้งหมดที่เกิดขึ้นในระบบและค่า timeout request ที่วัดได้นี้จะค่อยๆเพิ่มขึ้นทีละน้อยจนกระทั่งระบบมี user ที่ใช้ในการทดสอบ 4,200 user และหลังจากนั้นค่า timeout request ที่วัดได้ก็จะเพิ่มขึ้นอย่างรวดเร็วและเห็นได้ชัด

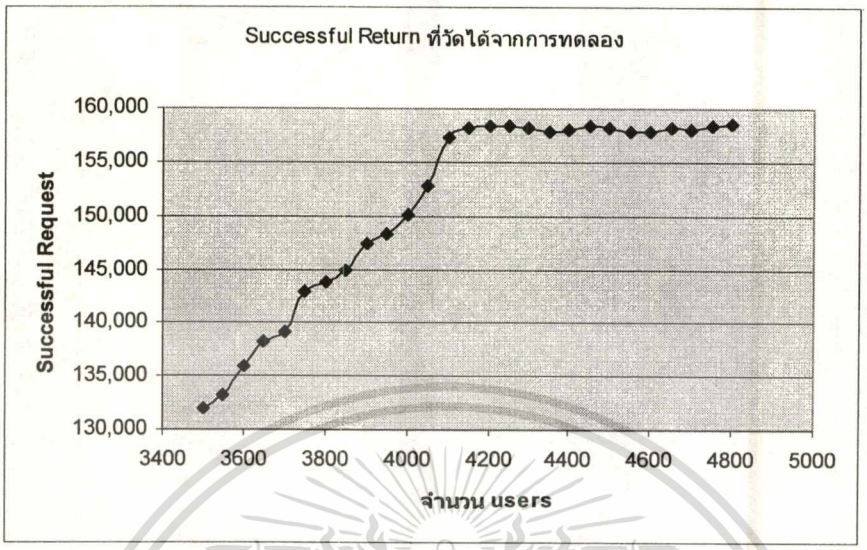
จากลักษณะดังกล่าวแสดงให้เห็นว่า ระบบที่ใช้ในการทดสอบ มีความสามารถในการรองรับการเรียกขอใช้งานระบบจากเครื่องลูกข่ายได้ดีที่สุดเมื่อมีจำนวน user ไม่เกิน 4,200 user แต่เมื่อมี user มากขึ้นไปกว่านั้น ถึงแม้ว่าระบบจะยังคงรองรับการเรียกขอใช้บริการจากเครื่องลูกข่ายได้ แต่จำนวนเครื่องลูกข่ายที่ไม่สามารถเรียกขอใช้บริการระบบได้ในเวลาที่กำหนด (timeout request) เพิ่มมากขึ้นเรื่อยๆ ดังนั้นในการทดสอบนี้จึงถือว่า ณ จุดที่ระบบสามารถรองรับ user ได้ 4,200 user เป็นจุดที่ระบบนี้มีประสิทธิภาพดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

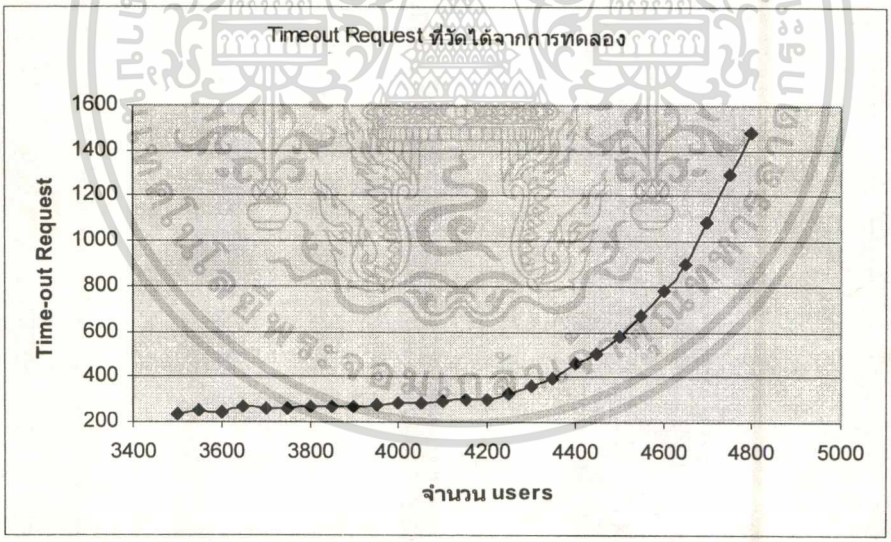
จำนวน users	Total Request	Suc. Return	ASP/sec	Time Out	% Time Out
3500	132,173	131,938	439.79	235	0.18%
3550	133,542	133,294	444.31	248	0.19%
3600	136,236	135,993	453.31	243	0.18%
3650	138,560	138,294	460.98	266	0.19%
3700	139,496	139,239	464.13	257	0.18%
3750	143,245	142,982	476.61	263	0.18%
3800	144,089	143,823	479.41	266	0.18%
3850	145,093	144,823	482.74	270	0.19%
3900	147,656	147,385	491.28	271	0.18%
3950	148,603	148,329	494.43	274	0.18%
4000	150,346	150,059	500.20	287	0.19%
4050	153,120	152,832	509.44	288	0.19%
4100	157,683	157,392	524.64	291	0.18%
4150	158,556	158,259	527.53	297	0.19%
4200	158,650	158,349	527.83	301	0.19%
4250	158,636	158,312	527.71	324	0.20%
4300	158,572	158,214	527.38	358	0.23%
4350	158,309	157,917	526.39	392	0.25%

ตารางที่ 4.4 ประสิทธิภาพที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 6 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 Successful Return ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 6 เครื่อง



รูปที่ 4.8 Timeout Request ที่วัดได้จากระบบที่มีเครื่องแม่ข่าย 6 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 การเปรียบเทียบประสิทธิภาพระหว่างระบบที่มีเครื่องแม่ข่ายจำนวนต่างกัน

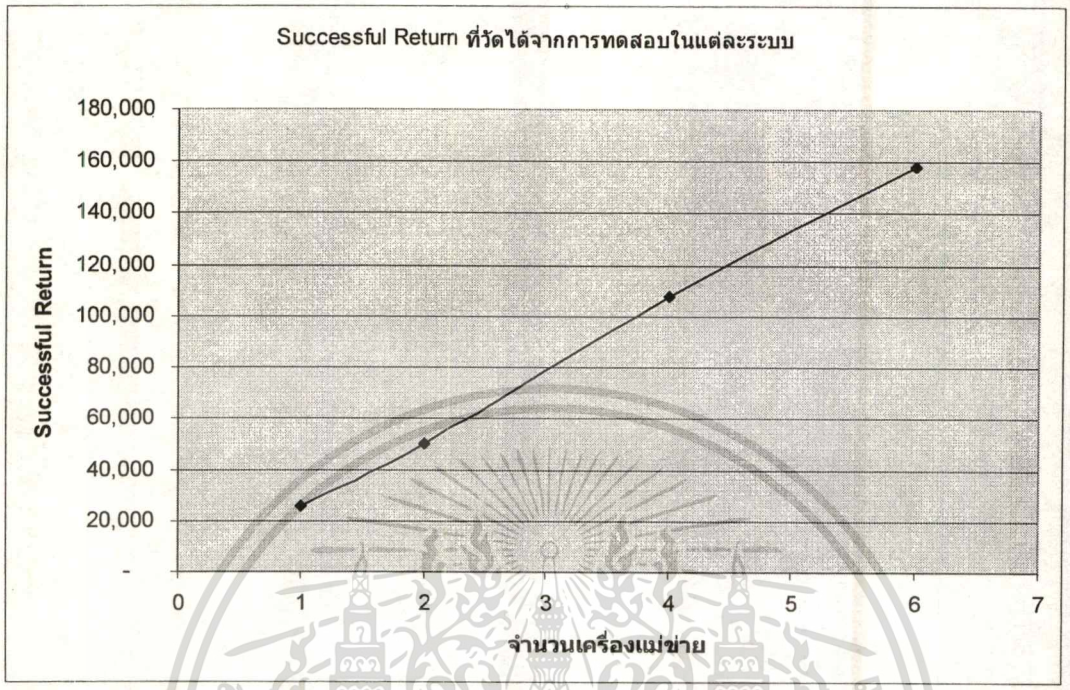
เมื่อเสร็จสิ้นการทดลองทั้ง 4 ระบบแล้วพบว่า ในระบบที่มีเครื่องแม่ข่าย 1, 2, 4, และ 6 เครื่องจะมีประสิทธิภาพในการทำงานดีที่สุดเมื่อมีจำนวน user ที่ 700, 1,350, 2,900 และ 4,250 user ตามลำดับ จากข้อมูลดังกล่าวเมื่อนำมาสร้างเป็นแผนภูมิเส้นแสดงความสัมพันธ์ระหว่างจำนวน successful return กับจำนวนเครื่องแม่ข่ายที่มีในระบบจะพบว่า ค่า successful return จะแปรผันตรงกับจำนวนเครื่องแม่ข่ายอย่างชัดเจนดังที่แสดงให้เห็นในภาพที่ 4.9

นอกจากนี้ หากนำจำนวน successful request และ timeout request มาเทียบเป็นร้อยละของจำนวน request ที่เกิดขึ้นทั้งหมดระหว่างการทดสอบ (total request) แล้วสร้างเป็นแผนภูมิเส้นเพื่อหาความสัมพันธ์ระหว่างค่าดังกล่าวกับจำนวนเครื่องแม่ข่ายในระบบดังที่แสดงไว้ในภาพที่ 4.10 และ 4.11 จะพบว่าเมื่อระบบที่ทำการทดสอบมีเครื่องแม่ข่ายจำนวนมากขึ้น ระบบมีแนวโน้มที่จะสามารถรองรับการเรียกใช้งานจากเครื่องลูกข่ายได้ในอัตราส่วนที่ดียิ่งขึ้น ซึ่งสามารถเห็นได้ชัดเจนจาก % successful return ที่เพิ่มขึ้นเล็กน้อย และ % timeout request ที่ลดลงเล็กน้อยเมื่อระบบมีเครื่องแม่ข่ายจำนวนมากขึ้น

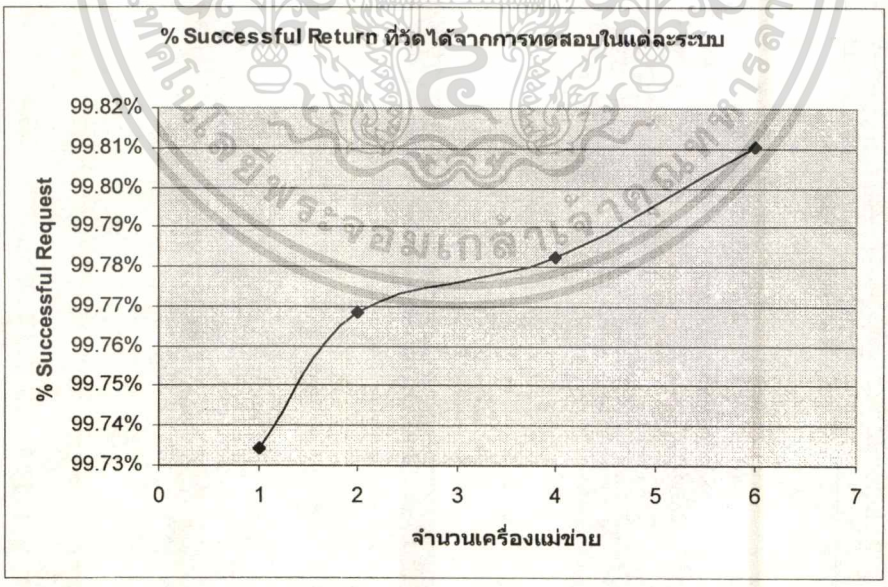
จำนวนเครื่องแม่ข่าย	1	2	4	6
Total Request	25,991	50,142	108,015	158,650
จำนวน users	700	1,350	2,900	4,250
Successful Return	25,922	50,026	107,780	158,349
ASP/sec	86.41	166.75	359.27	527.83
% Success	99.73%	99.77%	99.78%	99.81%
Time Out	69	116	235	301
% Time Out	0.27%	0.23%	0.22%	0.19%

ตารางที่ 4.5 แสดงการเปรียบเทียบประสิทธิภาพของระบบที่มีจำนวนเครื่องแม่ข่ายแตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

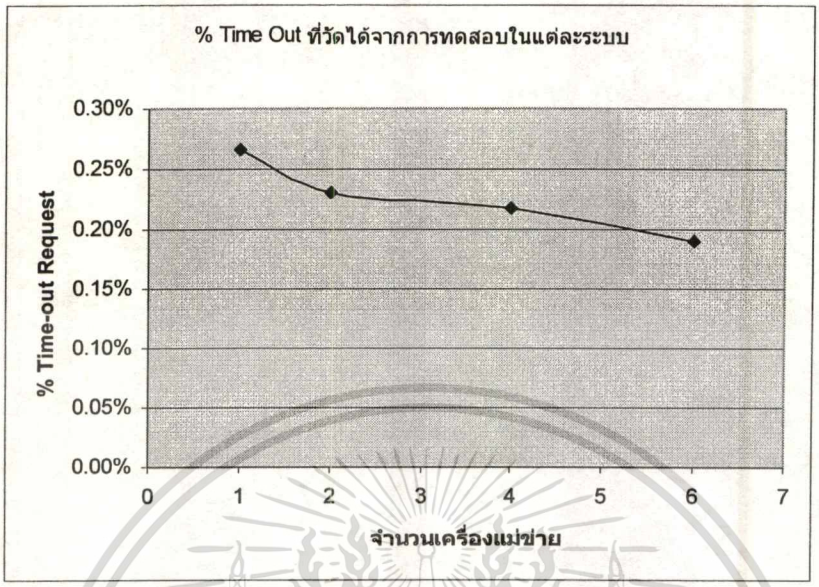


รูปที่ 4.9 แสดงความสามารถของระบบที่มีจำนวนเครื่องแม่ข่ายแตกต่างกัน



รูปที่ 4.10 แสดงร้อยละของ Successful Return ในแต่ละระบบเมื่อเทียบกับจำนวน Request ทั้งหมดที่เกิดขึ้นในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงร้อยละของ Timeout Request ในแต่ละระบบเมื่อเทียบกับจำนวน Request ทั้งหมดที่เกิดขึ้นในการทดสอบ

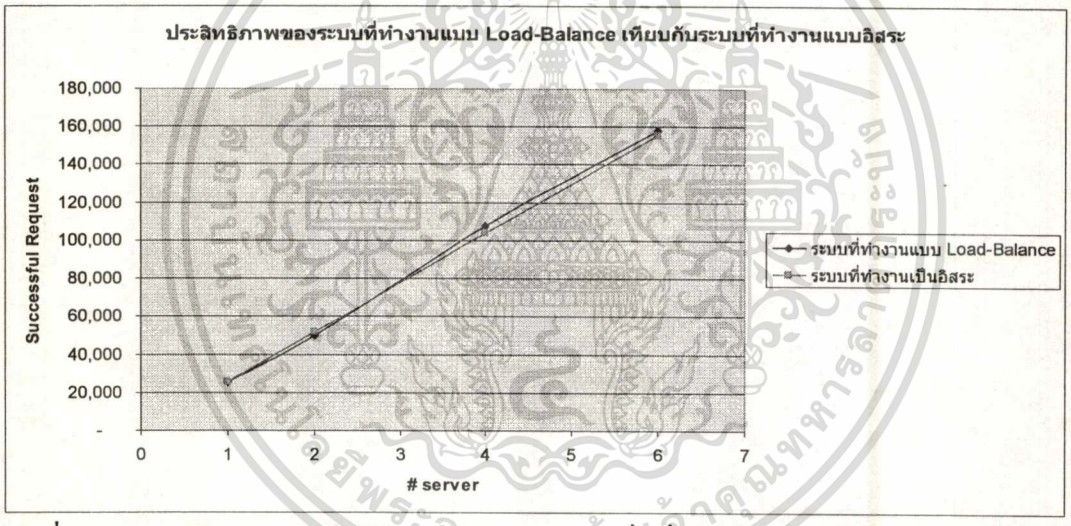
เพื่อเป็นการเปรียบเทียบประสิทธิภาพระหว่างระบบที่เครื่องแม่ข่ายแต่ละเครื่องทำงานเป็นอิสระต่อกันกับระบบที่เครื่องแม่ข่ายทั้งหมดทำงานร่วมกันในลักษณะ load-balance จึงได้มีการสร้างตารางที่ 4.6 และภาพที่ 4.12 ขึ้นเพื่อเปรียบเทียบระหว่างค่า successful return ที่วัดได้จริงในระบบที่ทำงานแบบ load-balance กับค่า successful return ของระบบที่ทำงานแบบอิสระซึ่งได้จากการคำนวนโดยใช้ประสิทธิภาพการทำงานของระบบที่มีเครื่องแม่ข่าย 1 เครื่องเป็นเกณฑ์

จากตารางที่ 4.6 และภาพที่ 4.12 แสดงให้เห็นว่าระบบที่ทำงานแบบ load-balance สามารถทำงานได้ดีกว่าระบบที่เครื่องแม่ข่ายแต่ละเครื่องทำงานเป็นอิสระต่อกันอยู่เล็กน้อย ลักษณะดังกล่าวอธิบายได้จากการที่ระบบที่ทำงานแบบ load-balance มี intelligent load balancer เข้ามาช่วยในการกระจายคำเรียกขอใช้งานระบบจากเครื่องลูกข่าย ทำให้เครื่องแม่ข่ายแต่ละเครื่องสามารถทำงานได้เต็มประสิทธิภาพโดยที่ไม่เกิดการ overload ขึ้นที่เครื่องใดเครื่องหนึ่ง และส่งผลให้ประสิทธิภาพโดยรวมของระบบดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนเครื่องแม่ข่าย	1	2	4	6
ระบบที่เครื่องแม่ข่ายทำงานร่วมกันแบบ Load-Balance (Request)	25,922	50,026	107,780	158,349
ระบบที่เครื่องแม่ข่ายทำงานแยกกันเป็นอิสระ (Request)	25,922	51,844	103,688	155,532

ตารางที่ 4.6 แสดงประสิทธิภาพของระบบที่เครื่องแม่ข่ายทำงานร่วมกันแบบ Load-Balance เมื่อเปรียบเทียบกับระบบที่เครื่องแม่ข่ายแต่ละเครื่องทำงานแยกกันอย่างเป็นอิสระ



รูปที่ 4.12 แสดงการเปรียบเทียบประสิทธิภาพของระบบที่เครื่องแม่ข่ายทำงานร่วมกันแบบ Load-Balance เมื่อเปรียบเทียบกับระบบที่เครื่องแม่ข่ายแต่ละเครื่องทำงานแยกกันอย่างเป็นอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดสอบความสามารถในการขยายความสามารถของระบบ

จากการทดสอบและผลการทดสอบที่ได้แสดงให้เห็นถึงความสัมพันธ์ระหว่างประสิทธิภาพของระบบที่เพิ่มขึ้นกับจำนวนเครื่องแม่ข่ายที่เพิ่มขึ้นอย่างชัดเจน และสามารถสรุปผลการทดสอบได้ดังนี้

1) ประสิทธิภาพของระบบโดยรวมจะแปรผันตรงกับจำนวนเครื่องแม่ข่ายที่ใช้ในการทำการทดสอบ โดยเมื่อมีการเพิ่มเครื่องแม่ข่ายเข้าไปในระบบมากขึ้น ระบบจะสามารถรองรับการขอใช้บริการจากเครื่องลูกข่ายได้เพิ่มขึ้นเป็นสัดส่วนเดียวกันกับจำนวนเครื่องแม่ข่ายที่เพิ่มขึ้นด้วย ลักษณะดังกล่าวแสดงให้เห็นว่า การใช้ Intelligent Load Balancer มาช่วยในระบบ ทำให้การเพิ่มขีดความสามารถของระบบทำได้ง่ายขึ้น และได้ประสิทธิภาพเพิ่มขึ้นเต็มที่ตามที่ต้องการ
2) เมื่อนำค่า % successful return และ % timeout request ที่วัดได้ ณ จุดอิ่มตัวของแต่ละระบบมาสร้างเป็นแผนภูมิเส้น ดังที่แสดงไว้ในภาพที่ 4.10 และ 4.11 ตามลำดับจะพบว่า เมื่อระบบมีเครื่องแม่ข่ายมากขึ้น ประสิทธิภาพโดยรวมของระบบมีแนวโน้มที่จะเพิ่มขึ้นเล็กน้อย ลักษณะดังกล่าวคาดว่าเป็นผลมาจากการใช้ Intelligent Load Balancer มาช่วยในการกระจายคำเรียกขอใช้บริการระบบที่ส่งมาจากผู้ใช้ไปยังเครื่องแม่ข่ายโดยพิจารณาจากประสิทธิภาพการทำงานของเครื่องแม่ข่ายแต่ละเครื่องในขณะนั้นเป็นเกณฑ์ ทำให้คำเรียกขอใช้บริการถูกส่งไปยังเครื่องที่มีประสิทธิภาพดีที่สุดในช่วงเวลานั้นๆเสมอ เป็นผลให้ระบบโดยรวมทำงานได้ดีขึ้นอีกเล็กน้อย
3) เมื่อนำค่า successful return ที่วัดได้ ณ จุดอิ่มตัวของแต่ละระบบมาสร้างเป็นแผนภูมิเส้น เปรียบเทียบกับประสิทธิภาพของระบบที่ใช้เครื่องแม่ข่ายแต่ละเครื่องทำงานแยกกันเป็นอิสระตั้งแต่ 1, 2, 4, และ 6 เครื่องซึ่งคำนวนได้จาก *(ประสิทธิภาพของระบบที่มีเครื่องแม่ข่าย 1 เครื่อง) x (จำนวนเครื่องในระบบ)* จะพบว่า ระบบที่มีเครื่องแม่ข่ายทำงานร่วมกันแบบ Load-Balance จะมีประสิทธิภาพโดยรวมสูงกว่าระบบที่มีเครื่องแม่ข่ายทำงานแยกกันเป็นอิสระอยู่เล็กน้อย ซึ่งสามารถอธิบายได้ว่าเป็นผลจากการกระจายคำเรียกขอใช้บริการจากเครื่องลูกข่ายที่มีประสิทธิภาพดีขึ้น ดังที่ได้อธิบายไว้แล้วในข้อ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะในการทดลอง

จากการทำการทดลองที่ได้พบว่ามีจุดที่สำคัญที่ควรทำการศึกษาเพิ่มเติมหรือพัฒนาปรับปรุงให้ดีขึ้นในประเด็นต่างดังต่อไปนี้

1. การใช้ Network Load Balancing Service ใน Windows 2000 Advance Server เพื่อเป็นตัวช่วยในการกระจาย request ที่ส่งมาจาก user ไปยังเครื่องแม่ข่ายหลายๆเครื่องนั้น ควรมีการใช้ Intelligent Load Balancer ชนิดอื่นๆเช่น Hardware-base Load Balancer ร่วมในการทดสอบด้วย เพื่อที่จะได้เป็นข้อเปรียบเทียบว่ามีประสิทธิภาพต่างกันเพียงใด
2. Network Load Balancing Service มีการกระจาย request โดยใช้อัลกอริธึมแบบ Fully Distributed Filtering ซึ่งเครื่องแม่ข่ายแต่ละตัวที่อยู่ในกลุ่ม Network Load Balance เดียวกันจะช่วยกันตัดสินใจว่าเครื่องแม่ข่ายเครื่องไหนจะจัดการกับ request ใดและจะทำงานเป็นอิสระต่อกัน วิธีดังกล่าวเป็นวิธีที่เหมาะกับระบบเครือข่ายที่มี user จำนวนมากที่มีการส่ง request ขนาดไม่ใหญ่นักแต่มีจำนวนมากเข้ามาหาเครื่องแม่ข่ายอยู่เสมอ แต่หากผู้ใช้งานต้องการที่จะทำ Load Balance ในเครือข่ายที่มีลักษณะแตกต่างออกไป ก็ควรจะพิจารณาทดสอบ Intelligent Load Balancer ประเภทอื่นๆด้วย
3. เนื่องจากการทดสอบครั้งนี้ มุ่งเน้นที่การทดสอบว่าเทคโนโลยีที่ทำการศึกษาสามารถนำมาออกแบบและพัฒนาระบบพาณิชย์อิเล็คโทรนิกที่มีความสามารถในการขยายขีดความสามารถได้จริงหรือไม่ ดังนั้นการทดสอบจึงมุ่งเน้นศึกษาในแง่ของปริมาณงานที่เครื่องแม่ข่ายหรือระบบจะสามารถรองรับได้ต่อหนึ่งหน่วยเวลาเท่านั้น ส่วนในแง่ของคุณภาพของงานที่ผู้ใช้จะได้รับ เช่น system response time นั้นจะถูกกำหนดให้เป็นตัวแปรคงที่เพื่อไม่ให้มีตัวแปรในการทดสอบมากจนเกินไป เช่น response ของเครื่องแม่ข่ายที่สามารถถือได้ว่าเป็น successful response นั้นต้องมีค่า response time ไม่เกิน 60 วินาที อย่างไรก็ดี ในการนำเทคโนโลยีที่ใช้ในการทดลองนี้ไปประยุกต์ใช้ในระบบจริง ควรคำนึงถึงลักษณะของเนื้อหาของ web site เป็นปัจจัยสำคัญด้วย เช่น web site ที่มีการใช้ ASP หรือ Java Script เป็นส่วนใหญ่และมีการดึงข้อมูลจากฐานข้อมูลบ่อยๆ มีแนวโน้มที่จะมี response time สูงกว่า web site ที่ข้อมูลส่วนใหญ่เป็น static content ดังนั้นการทดสอบก็ต้องมีการกำหนด response time ที่ต้องการให้เหมาะสมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Berg, A. 2000. **P2P, or Not P2P?.** Information Security. New York: McGraw-Hill.

Edwards, J. 2001. **Not Just for Music Anymore.** CIO Magazine. New York: n.p.

Ellis, R. 2000. **How to Stream your Media Files.** [Online]. Available: www.washington.edu/computer/windows/issue24/file.htm.

Enterprise Management Associate. 2001. **Managing Multi-Tiered E-Business Applications.** [Online]. Available: Http://www.enterprisemanagement.com/.

Internet and E-Commerce Solutions. 2002. **Load Balancing Considerations for Dynamic Internet Solutions Architecture Environments**. [Online]. Available: Http://www.compaq.com/activeanswers.

Internet and E-Commerce Solutions. 2002. **Overview of the Compaq Dynamic Internet Solutions Architecture (DISA).** [Online]. Available: Http://www.compaq.com/activeanswers.

King, S. et al. 2000. **The Case for IPv6**. Internet Architect Board: n.p.

Legard, D. 2000. **IDC : Instant Messaging to See Explosive Growth.** Infoworld. n.p.

Loshin, P. 1997. **Extranet Design and Implementation**. San Francisco: Sybex Network Press.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microsoft Corporation. 2000. **Network Load Balancing Technical Overview**. USA. n.p.

Pffafenberger, B. 1998. **Building a Strategic Internet**. Foster City: California.

Shirky, C. 2000. **What is P2P ...and What Isn't.** [Online]. Available: www.oreillynet.com.

Shirky, C. et. al. 2001. **The Emergent P2P Platform of Presence, Identity and Edge Resources**. Sebastopol, California: O'Reilly & Associates.

Small, P. 2001. **The Ultimate Game of Strategy**. London: FT.Com.

Treese, G. and Stewart, L. 1998. **Designing Systems for Internet Commerce**. Massachusetts: Addison-Wesley.

Viken, A. 2001. "Streaming: Past, Present and Future." PH.D. Thesis of Royal Institute of Technology. Stockholm, Sweden.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้