สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคล

SIMULATION OF REMOTE MAIN MEMORY PAGING ON NETWORK OF PERSONAL COMPUTERS

TQ61032

ธีราภรณ์ วิศิษฐวาณิชย์

TEERAPORN WISITTAVANIT

เลขหมู่.....
เลขทะเบียน.....61032
วัน,เดือน,ปี..7...ก.ค...2549..

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2548

ISBN 974-15-1515-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b.11554502
i.....

SIMULATION OF REMOTE MAIN MEMORY PAGING ON NETWORK OF PERSONAL COMPUTERS

TEERAPORN WISITTAVANIT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

ISBN 974-15-1515-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2005

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	**การจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคล**
นักศึกษา	**นางสาวธีราภรณ์ วิศิษฐวาณิชย์**
รหัสประจำตัว	45064611
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.ศรัณย์ อินทโกสุม

บทคัดย่อ

หน่วยความจำเสมือนแบบสลับหน้าระยะไกลเป็นการเพิ่มเติมแนวความคิดของหน่วยความเสมือนแบบสลับหน้าปกติด้วยการใช้หน่วยความจำหลักของเครื่องคอมพิวเตอร์ที่ต่อกันภายในเครือข่ายแทนการใช้ฮาร์ดดิสก์ ซึ่งการทดลองที่ผ่านมาทำบนเครือข่ายของเครื่องเวิร์กสเตชันที่เครือข่ายและคอมพิวเตอร์ทุกเครื่องไม่มีการทำงานอื่นใด พบว่ามีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ ผู้วิจัยจึงมีสมมติฐานว่าวิธีดังกล่าวควรจะนำมาประยุกต์ใช้กับเครือข่ายคอมพิวเตอร์ส่วนบุคคลได้ทั้งกรณีที่เครือข่ายและคอมพิวเตอร์ทุกเครื่องมีและไม่มีการทำงานอื่นใด ดังนั้นงานวิจัยนี้จึงสร้างแบบจำลอง 3 แบบจำลอง คือ แบบจำลองหน่วยความจำเสมือนแบบปกติ แบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลที่เครือข่ายและคอมพิวเตอร์ทุกเครื่องไม่มีการทำงานอื่นใด และแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลที่เครือข่ายและคอมพิวเตอร์มีการทำงานอื่น ด้วยโปรแกรมอารีน่า เวอร์ชัน 5 ซึ่งผลการทดลองพบว่าหน่วยความจำเสมือนแบบสลับหน้าระยะไกลที่เครือข่ายและคอมพิวเตอร์ทุกเครื่องไม่มีการทำงานอื่นใดมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ ในขณะกรณีที่เครือข่ายและเครื่องคอมพิวเตอร์ทุกเครื่องมีการทำงานอื่นนั้นจะมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติในบางกรณี ซึ่งองค์กรที่สนใจจะนำวิธีหน่วยความจำเสมือนแบบสลับหน้าระยะไกลไปใช้งานจริงสามารถนำผลการทดลองและข้อสรุปจากงานวิจัยนี้มาประกอบการตัดสินใจได้ นอกจากนี้ตัวแปรควบคุมในแบบจำลองสามารถปรับเปลี่ยนเพื่อให้มีความใกล้เคียงกับสภาพแวดล้อมจริงขององค์กรได้ ยิ่งไปกว่านั้นงานวิจัยนี้ยังเป็นตัวอย่างของการจำลองแบบด้วยคอมพิวเตอร์เพื่อวัดประสิทธิของคอมพิวเตอร์อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Simulation of Remote Main Memory Paging on Network of Personal Computer.
Student	Ms. Teeraporn Wisittavanit
Student ID.	45064611
Degree	Master of Science
Programme	Computer Science
Year	2005
Thesis Advisor	Asst.Prof.Dr.Sarun Intakosum

ABSTRACT

Remote main memory paging extends the idea of ordinary virtual memory paging by using the facts that memories of connected computer in a network are always available so that they can be used in place of a local hard disk. Previous researches implement this technique on a network of workstation, and all computers and the network are solely use for that purpose, which is called idle network. The results of the experiment show that the performance of the remote virtual memory is better than the ordinary one. The hypothesis of this research is that the idea of remote virtual memory paging should be applied to a network of personal computers that may be in either idle or working condition. Three simulation models, ordinary virtual memory model, remote virtual memory model with idle network, and remote virtual memory with working condition, are developed using Arena version 5. The results of the simulations show that the performance of remote virtual memory model with idle network is always better than that of the ordinary one, while the working condition model is better under some criteria. The contributions of this research are that, an organization can use the simulation results as an information to decide whether to implement the remote virtual memory. In addition, the control variables in the models can be modified to reflect the real network condition of an organization. Moreover, this research can be considered as an example of how to apply computer simulation for performance evaluation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์นี้มิอาจจะสำเร็จลุล่วงไปได้ด้วยดี หากมิได้รับคำแนะนำ คำชี้แจง ความรู้ และความเอาใจใส่จาก ผศ.ดร. ศรัณย์ อินทโกสุม ผู้เป็นอาจารย์ที่ปรึกษา ซึ่งท่านได้สละเวลาให้กับข้าพเจ้าอย่างเต็มที่ จึงใคร่ขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ รศ.ดร. วีระ บุญจริง สำหรับคำแนะนำ และให้คำปรึกษาต่าง ๆ เกี่ยวกับการสร้างแบบจำลองด้วยดีมาโดยตลอด

ขอขอบพระคุณ ดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร ที่ให้คำชี้แจงเกี่ยวกับเครือข่ายและการรับส่งผ่านเครือข่ายเป็นอย่างดี

ขอขอบพระคุณ ผศ.ดร.จีรพร ศรีสวัสดิ์ ที่กรุณาตรวจสอบความถูกต้องทางด้านสถิติต่าง ๆ

ขอขอบพระคุณ มารดา ที่สนันสนุนให้ได้เรียนในระดับที่ได้ตั้งใจ อีกทั้งยังได้ดูแลเรื่องค่าใช้จ่ายต่าง ๆ ระหว่างการศึกษาเป็นอย่างดีอีกด้วย

ขอขอบพระคุณ น้า อา ทุกท่านที่คอยให้กำลังใจ และคอยช่วยเหลือด้านวัสดุ อุปกรณ์การทำงานต่าง ๆ

ขอขอบพระคุณ นายคุสิต นิยะโต และนายนคร สาแก้ว ที่คอยให้คำปรึกษาเกี่ยวกับวิทยานิพนธ์ และให้กำลังใจในการทำงานมาโดยตลอด

ขอขอบคุณ พี่ น้อง เพื่อน รุ่นน้อง ทุกคนที่คอยให้กำลังใจ ตักเตือน และผลักดันตลอดการทำงาน

สำหรับคุณงามความดีและประโยชน์อันใดที่เกิดขึ้นจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดา มารดา อาจารย์ทุกท่านซึ่งเป็นที่เคารพรักยิ่ง ตลอดจนญาติพี่น้อง และเพื่อน ๆ ทุกคน

ธีราภรณ์ วิศิษฐวาณิชย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ 2

1.3 สมมติฐานของการศึกษา 2

1.4 ขอบเขตการศึกษา 2

1.5 ขั้นตอนการศึกษา 3

1.6 คำจำกัดความที่ใช้ในการศึกษา 4

บทที่ 2 ทฤษฎีงานวิจัยที่เกี่ยวข้อง 5

2.1 แนวคิดเบื้องต้นเกี่ยวกับหน่วยความจำเสมือนแบบสลับหน้าระยะไกล 5

2.2 วิธีเพิ่มความน่าเชื่อถือของรีโมตเมนเมมโมรี 6

2.2.1 มิเรอร์ริง 6

2.2.2 พาริตี 7

2.2.3 พาริตีล็อกกิง 11

2.2.3.1 ล็อกกิงกรีดดี 14

2.2.3.2 ล็อกกิงเซต 14

2.2.4 ดีเลย์พาริตี 15

ผลการทดลองของหน่วยความจำเสมือนแบบสลับหน้าระยะไกล
แบบรีโมตเมนเมมโมรี 16

2.3 วิธีเพิ่มความน่าเชื่อถือของรีโมตแรมดิสก์ 16

2.3.1 พาริตีแคชชิง 16

2.3.2 อแดปทีฟพาริตีแคชชิง 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การสร้างแบบจำลอง 25

3.1 แบบจำลองปกติ 25

3.2 แบบจำลองระยะไกล 30

3.3 แบบจำลองระยะไกลแบบจำกัด 37

บทที่ 4 การทดลองและการวิเคราะห์ผล 43

4.1 การทดลองที่ 1 44

4.2 การทดลองที่ 2 52

บทที่ 5 สรุปและข้อเสนอแนะ 56

5.1 สรุปผลการทดลอง 56

5.2 ข้อเสนอแนะ 57

5.2.1 เกี่ยวกับแบบจำลอง 57

5.2.2 เกี่ยวกับการนำไปใช้จริง 58

เอกสารอ้างอิง 59

ภาคผนวก

ภาคผนวก ก ตัวแปรที่ใช้ในแบบจำลอง 60

ภาคผนวก ข แบบจำลองปกติ 67

ภาคผนวก ค แบบจำลองระยะไกลและแบบจำลองระยะไกลแบบจำกัด 81

ประวัติผู้เขียน 112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 กระบวนการย้ายเพจไปเครื่องผู้ให้บริการ 10

2.2 การส่งข้อมูลไปเครื่องผู้ให้บริการต่าง ๆ ของตัวอย่างที่ 2.8 ขั้นที่ 12 23

4.1 เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ และส่วนเบี่ยงเบนมาตรฐานของแบบจำลองปกติ 46

4.2 เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ และส่วนเบี่ยงเบนมาตรฐานของแบบจำลองระยะไกล เมื่อเครือข่ายมีจำนวนเครื่องผู้ให้บริการ3-9 เครื่องและหน่วยความจำหลักมีขนาด 10, 20 เปอร์เซ็นต์ของขนาดโปรแกรม 47

4.3 ค่าสถิติ Z เมื่อเครือข่ายมีจำนวนเครื่องผู้ให้บริการ 3-9 เครื่อง และหน่วยความจำหลักมีขนาด 10, 20 เปอร์เซ็นต์ของขนาดโปรแกรม 47

4.4 เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ เมื่อเครือข่ายมีจำนวนเครื่องผู้ให้บริการ 3-9 เครื่อง และหน่วยความจำหลักมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม 49

4.5 จำนวนเครื่องผู้ให้บริการน้อยที่สุดเมื่อมีสภาพแวดล้อมถูกจำกัด และหน่วยความจำหลักมีขนาด 10 เปอร์เซ็นต์ของขนาดโปรแกรม 53

4.6 จำนวนเครื่องผู้ให้บริการน้อยที่สุดเมื่อมีสภาพแวดล้อมถูกจำกัด และหน่วยความจำหลักมีขนาด 20 เปอร์เซ็นต์ ของขนาดโปรแกรม 53

ก.1 ตัวแปรแบบจำลองปกติ 61

ก.2 ตัวแปรแบบจำลองระยะไกลและแบบจำลองระยะไกลแบบจำกัด 63

ข.1 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 1 69

ข.2 การทำงานของโมดูลในโมเดลย่อย Global Variable 70

ข.3 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 2 71

ข.4 การทำงานของโมดูลในโมเดลย่อย Compute BlockN 72

ข.5 การทำงานของในโมเดลย่อยในส่วนที่ 3 73

ข.6 การทำงานของโมดูลในโมเดลย่อย Page in RAM 73

ข.7 การทำงานของโมเดลย่อยในส่วนที่ 4 74

ข.8 การทำงานของโมดูลในโมเดลย่อย Page in HD 75

ข.9 การทำงานของโมดูลในโมเดลย่อย Paging 76

ข.10 การทำงานของโมดูลในโมเดลย่อย RAM have free memory 77

ข.11 การทำงานของโมดูลในโมเดลย่อย Sent Page to HD 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

ข.12 การทำงานของโมดูลในโมเดลย่อย Put new block in RAM 79
ข.13 การทำงานของโมดูลในส่วนที่ 5 80
ค.1 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 1 83
ค.2 การทำงานของโมดูลในโมเดลย่อย Global Variable 84
ค.3 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 2 86
ค.4 การทำงานของโมดูลในโมเดลย่อย Compute BlockN 87
ค.5 การทำงานของในโมเดลย่อยในส่วนที่ 3 88
ค.6 การทำงานของโมดูลในโมเดลย่อย Page in RAM 88
ค.7 การทำงานของโมเดลย่อยในส่วนที่ 4 89
ค.8 การทำงานของโมดูลในโมเดลย่อย Page in HD 90
ค.9 การทำงานของโมเดลย่อยในส่วนที่ 5 90
ค.10 การทำงานของโมดูลในโมเดลย่อย Page in Server 91
ค.11 การทำงานของโมเดลย่อยในส่วนที่ 6 93
ค.12 การทำงานของโมดูลในโมเดลย่อย Paging 94
ค.13 การทำงานของโมดูลในโมเดลย่อย RAM have free memory 95
ค.14 การทำงานของโมดูลในโมเดลย่อย RAM don't have free mem 96
ค.15 การทำงานของโมดูลและโมเดลย่อยในโมเดลย่อย Adaptive 97
ค.16 การทำงานของโมดูลในโมเดลย่อย Check con 98
ค.17 การทำงานของโมดูลและโมเดลย่อยในโมเดลย่อย Don't continue in Stream 99
ค.18 การทำงานของโมดูลในโมเดลย่อย Put in Victim 100
ค.19 การทำงานของโมดูลในโมเดลย่อย Continue in Stream 101
ค.20 การทำงานของโมดูลในโมเดลย่อย Check full Stream 102
ค.21 การทำงานของโมดูลในโมเดลย่อย Stream full 103
ค.22 การทำงานของโมดูลในโมเดลย่อย Clear Stream Victim full twice 105
ค.23 การทำงานของโมดูลในโมเดลย่อย Have free ram in server 107
ค.24 การทำงานของโมดูลในโมเดลย่อย Clear stream full 108
ค.25 การทำงานของโมดูลในโมเดลย่อย Put new block in RAM 110
ค.26 การทำงานของโมดูลในส่วนที่ 7 111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 การทำงานของวิธีมิเรอร์ริง 7
2.2 เครื่องผู้ให้บริการของวิธีพาริตี 7
2.3 จำนวนเพจของวิธีพาริตี 7
2.4 การสร้างพาริตีเพจของวิธีพาริตี 8
2.5 การกู้คืนเพจของวิธีพาริตี 9
2.6 พาริตีล็อกกิง 12
2.7 การคำนวณพาริตีเพจของพาริตีล็อกกิง 12
2.8 พาริตีแคช 17
2.9 อแดปทีฟพาริตีแคชชิง 18
2.10 อแดปทีฟพาริตีแคช 18
2.11 ลำดับของบล็อกที่โปรแกรมต้องการใช้ในการประมวลผล 19
2.12 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 1 20
2.13 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 2-4 20
2.14 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 5 21
2.15 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 6-7 21
2.16 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 8 22
2.17 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 9 22
2.18 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 10-12 23
3.1 ความสัมพันธ์ของส่วนประกอบต่าง ๆ ของแบบจำลองปกติ 26
3.2 ความสัมพันธ์ของส่วนประกอบต่าง ๆ ของแบบจำลองระยะไกล 30
4.1 เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จการทดลองที่ 1 48
4.2 เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จการทดลองที่ 1 . 50
ข.1 แบบจำลองปกติ 68
ข.2 ส่วนที่ 1 69
ข.3 ภายในโมเดลย่อย Global Variable 69
ข.4 ส่วนที่ 2 71
ข.5 ภายในโมเดลย่อย Compute BlockN 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

ข.6 ส่วนที่ 3 73
ข.7 ภายในโมเดลย่อย Page in RAM 73
ข.8 ส่วนที่ 4 74
ข.9 ภายในโมเดลย่อย Page in HD 75
ข.10 ภายในโมเดลย่อย Paging 75
ข.11 ภายในโมเดลย่อย RAM have free memory 76
ข.12 ภายในโมเดลย่อย Sent Page to HD 78
ข.13 ภายในโมเดลย่อย Put new block in RAM 79
ข.14 ส่วนที่ 5 80
ค.1 แบบจำลองระยะไกลและแบบจำลองระยะไกลแบบจำกัด 82
ค.2 ส่วนที่ 1 83
ค.3 ภายในโมเดลย่อย Global Variable 83
ค.4 ส่วนที่ 2 85
ค.5 ภายในโมเดลย่อย Compute BlockN 86
ค.6 ส่วนที่ 3 87
ค.7 ภายในโมเดลย่อย Page in RAM 88
ค.8 ส่วนที่ 4 89
ค.9 ภายในโมเดลย่อย Page in HD 89
ค.10 ส่วนที่ 5 90
ค.11 ภายในโมเดลย่อย Page in Server 91
ค.12 ส่วนที่ 6 93
ค.13 ภายในโมเดลย่อย Paging 93
ค.14 ภายในโมเดลย่อย RAM have free memory 95
ค.15 ภายในโมเดลย่อย RAM don't have free mem 96
ค.16 ภายในโมเดลย่อย Adaptive 97
ค.17 ภายในโมเดลย่อย Check con 98
ค.18 ภายในโมเดลย่อย Don't continue in Stream 99
ค.19 ภายในโมเดลย่อย Put in Victim 100
ค.20 ภายในโมเดลย่อย Continue in Stream 101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

ค.21 ภายในโมเดลย่อย Check full Stream 101
ค.22 ภายในโมเดลย่อย Stream full 103
ค.23 ภายในโมเดลย่อย Clear Stream Victim full twice 104
ค.24 ภายในโมเดลย่อย Have free ram in server 107
ค.25 ภายในโมเดลย่อย Clear stream full 108
ค.26 ภายในโมเดลย่อย Put new block in RAM 109
ค.27 ส่วนที่ 7 111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การที่หน่วยความจำหลักมีขนาดเล็กกว่าขนาดของโปรแกรม ทำให้ไม่สามารถนำโปรแกรมทั้งหมดเข้าสู่หน่วยความจำหลักได้ วิธีแก้ปัญหาที่นิยมกันวิธีหนึ่ง คือ หน่วยความจำเสมือนแบบสลับหน้า โดยโปรแกรมจะถูกแบ่งออกเป็นส่วน ๆ เรียกว่า หน้าหรือเพจ (Page) ซึ่งมีขนาดเท่า ๆ กันและหน่วยความจำหลักถูกแบ่งเป็น เฟรม (Frame) ซึ่งมีขนาดเท่ากับเพจ จากนั้นระบบปฏิบัติการจะทำการโหลดข้อมูลเพียงบางส่วนเข้าสู่หน่วยความจำหลัก ส่วนข้อมูลอื่น ๆ จะเก็บไว้ที่ฮาร์ดดิสก์ ซึ่งทำให้หน่วยประมวลผลกลางไม่สามารถทำการประมวลผลได้อย่างต่อเนื่อง เพราะต้องทำการโหลดข้อมูลจากฮาร์ดดิสก์มายังหน่วยความจำหลัก และการที่เวลาในการรับส่งข้อมูลของฮาร์ดดิสก์ช้ากว่าหน่วยความจำหลักมากจึงทำให้เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จมากขึ้น จึงได้เกิดงานวิจัยที่ใช้หน่วยความจำหลักบางส่วนของเครื่องคอมพิวเตอร์อื่น ๆ ที่ต่อกันเป็นเครือข่ายทำหน้าที่เก็บข้อมูลแทนฮาร์ดดิสก์ซึ่ง เรียกว่า หน่วยความจำเสมือนแบบสลับหน้าระยะไกล (Remote Main Memory Paging) [7] อย่างไรก็ตามวิธีนี้ต้องทำงานผ่านระบบเครือข่ายซึ่งมีโอกาสที่จะเกิดความผิดพลาดได้มากกว่าการทำงานภายใน 1 เครื่อง จึงได้มีการพัฒนาวิธีเพิ่มความน่าเชื่อถือ [7,8,9,10,11] ซึ่งการทดลองที่ผ่านมาทำการทดลองจริงบนเครือข่ายของเครื่องเวิร์คสเตชันที่ทุกเครื่องไม่มีการทำงานอื่นใดนอกจากการทดลอง ผลเป็นที่ยอมรับว่าเมื่อใช้วิธีเพิ่มความน่าเชื่อถือทุกวิธีจะช่วยเพิ่มความน่าเชื่อถือให้กับระบบ และทำให้เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จน้อยกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติ แต่เมื่อเครื่องผู้ให้บริการมีการทำงานอื่นด้วยจะทำให้ประสิทธิภาพการทำงานของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลลดลง

ปัจจุบันคอมพิวเตอร์ส่วนบุคคลพัฒนาไปอย่างรวดเร็ว ทำให้มีประสิทธิภาพสูงและได้เข้ามาแทนที่เวิร์คสเตชันมากขึ้น จึงได้เกิดแนวความคิดที่จะนำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมาใช้กับเครือข่ายคอมพิวเตอร์ส่วนบุคคล แต่การพัฒนาโปรแกรมเพื่อทำการทดลองจริงนั้นจำเป็นต้องใช้งบประมาณและเวลามาก อีกทั้งยังไม่สามารถควบคุมสภาพแวดล้อมได้ตามที่ต้องการ ดังนั้นงานวิจัยนี้จึงสร้างแบบจำลองเพื่อทดสอบหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคล เพื่อให้ทราบถึงขอบเขตของสภาพแวดล้อมทางด้านเครือข่ายและด้านเครื่องผู้ให้บริการที่ทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังทำงานได้ดีกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ ซึ่งผลการทดลองนี้หน่วยงานหรือองค์กรที่สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะนำวิธีนี้ไปใช้เพื่อเพิ่มประสิทธิภาพให้กับระบบสามารถนำไปประกอบการพิจารณาได้ อีกทั้งผู้วิจัยที่สนใจวิธีหน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังสามารถนำผลการทดลองนี้ไปพัฒนาหน่วยความจำเสมือนแบบสลับหน้าระยะไกลให้มีประสิทธิภาพดียิ่งขึ้นได้

1.2 ความมุ่งหมายและวัตถุประสงค์

1.2.1 สร้างแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคลที่เครือข่ายทำการรับส่งข้อมูลด้วยความเร็วสูงสุด เครือข่ายมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ และทุกเครื่องไม่มีการทำงานอื่นใดนอกจากการทดลอง เพื่อวัดประสิทธิภาพว่าดีกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติ

1.2.2 สร้างแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคลที่เครือข่ายมีความเร็วในการรับส่งผ่านเครือข่ายกับประสิทธิภาพในเครือข่ายถูกจำกัด และเครื่องผู้ให้บริการทำงานอื่นนอกจากการทดลอง เพื่อให้ทราบถึงสภาพแวดล้อมที่หน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังทำงานได้ดีกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ

1.3 สมมุติฐานของการศึกษา

1.3.1 หน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคลที่เครือข่ายทำการรับส่งข้อมูลด้วยความเร็วสูงสุด เครือข่ายมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ และทุกเครื่องไม่มีการทำงานอื่นใดนอกจากการทดลองให้ประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติ

1.3.2 หน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคลจะมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติในขอบเขตสภาพแวดล้อมบางประการ

1.4 ขอบเขตการศึกษา

1.4.1 สร้างแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลที่ใช้วิธีเพิ่มความน่าเชื่อถือ คือ อแดปทีฟพาริตีแคชชิง

1.4.2 ตัวแปรที่สนใจศึกษา คือ ความเร็วในการรับส่งผ่านเครือข่าย ประสิทธิภาพของเครือข่าย ขนาดหน่วยความจำหลักของเครื่องผู้ให้บริการและเครื่องผู้ขอใช้บริการ เครื่องผู้ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานอื่นนอกจากการทดลอง ความเร็วในการรับส่งข้อมูลของฮาร์ดดิสก์ และจำนวนเครื่องผู้ให้บริการ

1.4.3 ผลลัพธ์ที่ได้จากแบบจำลอง คือ เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งเวลาดังกล่าวไม่รวมเวลาการเข้าใช้หน่วยประมวลผลกลาง เวลาในการค้นหาที่อยู่ของเพจในตารางเพจ เวลาที่จัดการสัญญาณขัดจังหวะ เวลาที่ใช้ในการคำนวณพาริตีเพจ และเวลาที่เครื่องผู้ให้บริการทำการประมวลผลคำร้องขอเพจที่ส่งจากเครื่องผู้ขอใช้บริการ

1.4.4 วิธีการเลือกเพจที่จะถูกแทนที่ใช้วิธีเข้าก่อนออกก่อน (First In First Out : FIFO)

1.4.5 ทุกครั้งที่จะส่งเพจกลับฮาร์ดดิสก์จะไม่มีการตรวจสอบว่าเพจมีการเปลี่ยนแปลงหรือไม่จะทำการเขียนทับทุกครั้ง

1.4.6 เครือข่ายมีจำนวนเครื่องคอมพิวเตอร์ส่วนบุคคลทั้งหมดไม่เกิน 10 เครื่อง

1.4.7 ใช้โพรโตคอลยูดีพี (User Datagram Protocol : UDP) ในการรับส่งข้อมูลจึงทำให้ไม่มีการเปิดช่องสื่อสารก่อนทำการรับส่ง และสมมติว่าประสบความสำเร็จทุกครั้งของการรับส่ง

1.4.8 หน่วยความจำหลักในเครื่องผู้ให้บริการที่จองไว้สำหรับการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลจะไม่มีการนำโปรแกรมอื่นมาแทนที่

1.4.9 คำร้องขอเพจจากเครื่องผู้ขอใช้บริการถือว่ามีความสำคัญสูงกว่าทุก ๆ คำร้อง ดังนั้นเมื่อเครื่องผู้ให้บริการได้รับคำร้องขอเพจจะต้องทำการประมวลผลทันที

1.4.10 ทำการทดลองกรณีที่หน่วยความจำหลักมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม

1.5 ขั้นตอนการศึกษา

1.5.1 ศึกษาการทำงานของหน่วยความจำเสมือนแบบสลับหน้าปกติและหน่วยความจำเสมือนแบบสลับหน้าระยะไกล

1.5.2 สร้างแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าปกติ

1.5.3 สร้างแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายของคอมพิวเตอร์ส่วนบุคคลที่เครือข่ายทำการรับส่งข้อมูลด้วยความเร็วสูงสุด เครือข่ายมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ และทุกเครื่องไม่มีการทำงานอื่นใดนอกจากการทดลอง

1.5.4 วิเคราะห์ผลการทดลอง เปรียบเทียบประสิทธิภาพของแบบจำลองในข้อ 1.5.2 กับ 1.5.3

1.5.5 สร้างแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคลที่เครือข่ายมีความเร็วในการรับส่งผ่านเครือข่ายกับประสิทธิภาพในเครือข่ายถูกจำกัด และเครื่องผู้ให้บริการทำงานอื่นนอกจากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5.6 วิเคราะห์ผลการทดลอง เปรียบเทียบประสิทธิภาพของแบบจำลองข้อ 1.5.2 กับ 1.5.5 เพื่อหาขอบเขตของสภาพแวดล้อมที่หน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังทำงานได้ดีกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ

1.6 คำจำกัดความที่ใช้ในการศึกษา

1.6.1 หน่วยความจำเสมือนแบบสลับหน้าระยะไกล คือ การใช้หน่วยความจำหลักบางส่วนของเครื่องคอมพิวเตอร์ที่ต่อกันเป็นเครือข่ายเป็นหน่วยความจำสำรองในการทำหน่วยความจำเสมือนแบบสลับหน้า

1.6.2 แบบจำลองปกติ คือ แบบจำลองหน่วยความจำเสมือนแบบสลับหน้าปกติ

1.6.3 แบบจำลองระยะไกล คือ เป็นแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายของคอมพิวเตอร์ส่วนบุคคลซึ่งประกอบด้วยเครื่องผู้ขอใช้บริการ 1 เครื่อง เครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน S เครื่อง และเครื่องผู้ให้บริการที่เก็บพาริตีจำนวน 1 เครื่อง โดยที่ทุกเครื่องในเครือข่ายไม่มีการทำงานอื่นใดนอกจากการทดลองทำให้เครือข่ายมีการรับส่งข้อมูลด้วยความเร็วสูงสุด และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์

1.6.4 แบบจำลองระยะไกลแบบจำกัด คือ เป็นแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายของคอมพิวเตอร์ส่วนบุคคลซึ่งประกอบด้วยเครื่องผู้ขอใช้บริการ 1 เครื่อง เครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน S เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีจำนวน 1 เครื่อง และเครื่องคอมพิวเตอร์อื่นนอกเหนือจากการทดลอง โดยที่เครือข่ายและเครื่องผู้ให้บริการกับเครื่องคอมพิวเตอร์อื่นนั้นมีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพถูกจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดเบื้องต้นเกี่ยวกับหน่วยความจำเสมือนแบบสลับหน้าระยะไกล

แนวคิดของหน่วยความจำเสมือนแบบสลับหน้าระยะไกล เกิดขึ้นเนื่องจากฮาร์ดดิสก์มีอัตรารับส่งข้อมูลช้ากว่าหน่วยความจำหลักมาก ทำให้มีแนวความคิดที่จะหาหน่วยความจำชนิดอื่นมาเป็นอุปกรณ์ในการทำหน่วยความจำเสมือนแบบสลับหน้าแทนฮาร์ดดิสก์ เพื่อให้สามารถรองรับระบบที่ต้องการประสิทธิภาพสูง ๆ ได้ จากการทดลองวัดปริมาณการใช้หน่วยความจำหลักทั้งหมดของเครื่องเวิร์คสเตชันที่ต่อกันเป็นเครือข่าย พบว่าหน่วยความจำหลักถูกใช้งานไม่เต็ม 100 เปอร์เซ็นต์ จึงมีงานวิจัยทำการทดลองใช้หน่วยความจำหลักบางส่วนของเครื่องเวิร์คสเตชันที่ต่อกันเป็นเครือข่ายทำหน้าที่สำรองข้อมูลแทนฮาร์ดดิสก์ [7,8,9,10,11] หรือเรียกว่าหน่วยความจำเสมือนแบบสลับหน้าระยะไกล ซึ่งมีพื้นฐานการทำงานแบบเครื่องผู้ให้บริการ (Server) และเครื่องผู้ขอใช้บริการ (Client) การทดลองดังกล่าวทำบนระบบเครือข่ายที่ใช้โพรโตคอลทีซีพี/ไอพี (TCP/IP) ซึ่งเครื่องผู้ขอใช้บริการจะติดต่อกับเครื่องผู้ให้บริการผ่านทางโปรแกรมที่สร้างขึ้นและใช้ซอคเก็ต (Socket) ในการรับส่งข้อมูลผ่านทางพอร์ต (Port) ที่กำหนดขึ้นพิเศษ

วิธีหลัก 2 วิธีของการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกล คือ รีโมตเมนเมมโมรี (Remote Main Memory) ที่มีการรับส่งทีละเพจ และรีโมตแรมดิสก์ (Remote Ram Disk) ที่มีการรับส่งทีละกลุ่มของเพจหรือบล็อก ซึ่งมีรายละเอียดดังต่อไปนี้

รีโมตเมนเมมโมรี มีหลักการคือ เมื่อหน่วยความจำหลักของเครื่องผู้ขอใช้บริการมีพื้นที่ไม่พอสำหรับเพจใหม่ ก็จะเลือกเพจที่จะถูกแทนที่ด้วยวิธีการแทนที่ต่าง ๆ เช่น แทนที่ส่วนที่ไม่ถูกใช้งานนานที่สุด (Least-Recently Used : LRU) [8] หรือ เข้าก่อนออกก่อน (First In First Out : FIFO) จากนั้นจะนำเพจที่จะถูกแทนที่ไปเก็บยังหน่วยความจำหลักบางส่วนของบนเครื่องผู้ให้บริการที่มีโปรแกรมทำการจองไว้ แต่ถ้าหน่วยความจำหลักของเครื่องผู้ให้บริการเต็มจะทำหน่วยความจำเสมือนแบบสลับหน้าปกติแทน

รีโมตแรมดิสก์ มีหลักการคือ เมื่อหน่วยความจำหลักของเครื่องผู้ขอใช้บริการมีพื้นที่ไม่พอสำหรับบล็อกใหม่ก็จะเลือกบล็อกที่จะถูกแทนที่ด้วยวิธีการแทนที่ต่าง ๆ จากนั้นจะนำบล็อกที่จะถูกแทนที่ไปเก็บที่แรมดิสก์ (RamDisk) ซึ่งแรมดิสก์ คือ หน่วยความจำหลักบางส่วนของเครื่องผู้ให้บริการที่ถูกโปรแกรมจัดการให้มีการทำงานเสมือนเป็นฮาร์ดดิสก์ก้อนหนึ่ง แต่ถ้าแรมดิสก์เต็มจะทำหน่วยความจำเสมือนแบบสลับหน้าปกติแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำรีโมตแรมดิสก์จะดีกว่ารีโมตเมนเมมโมรี ในกรณีที่โปรแกรมมีความต้องการเพจที่ต่อกัน เนื่องจากรีโมตแรมดิสก์มีการรับส่งทีละบล็อก แต่รีโมตเมนเมมโมรีรับส่งทีละเพจทำให้รีโมตเมนเมมโมรีต้องใช้เวลาในการเข้าถึงข้อมูลมากกว่ารีโมตแรมดิสก์เพื่อให้ได้ข้อมูลชุดเดียวกัน

เคาน์เตอร์ (Counter) [8] มีหลักการทำงานคือ นำเพจบางส่วนไปไว้ที่หน่วยความจำหลักของเครื่องผู้ให้บริการ และมีตัวแปรเคาน์เตอร์ซึ่งเก็บจำนวนการอ้างถึงของทุกเพจ (ค่าที่เหมาะสมที่สุด คือประมาณ 64-512 [8]) เมื่อมีการอ้างถึงเพจ ค่าเคาน์เตอร์ของเพจนั้นก็จะลดลง 1 และเมื่อเท่ากับ 0 ก็จะมีอินเตอร์รัปต์ (Interrupt) ไปยังเครื่องผู้ขอใช้บริการให้นำเพจดังกล่าวกลับมายังหน่วยความจำหลักของตน ทำให้เพจที่มีการอ้างถึงบ่อย ๆ จะอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ แต่เพจที่มีการอ้างถึงน้อย ๆ อยู่ที่เครื่องผู้ให้บริการ ซึ่งจะเป็นการลดการรับส่งข้อมูลภายในเครือข่ายได้เมื่อโดยเฉพาะโปรแกรมมีจำนวนการอ้างถึงของแต่ละเพจแตกต่างกัน แต่ในกรณีที่แต่ละเพจในโปรแกรมมีจำนวนการอ้างถึงเท่า ๆ กันก็จะทำให้แสดงประสิทธิภาพของวิธีนี้ลดลง นอกจากนี้วิธีนี้ยังต้องมี ตัวแปรจำนวนมากเพื่อเก็บจำนวนการอ้างถึงของแต่ละเพจอีกด้วย

เนื่องจากการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกล เป็นการทำงานผ่านเครือข่าย จึงทำให้มีโอกาสที่จะเกิดการผิดพลาดมากกว่าการทำงานภายในเครื่อง 1 เครื่อง จึงมีผู้วิจัย [7,8,9,10,11] เสนอวิธีเพิ่มความน่าเชื่อถือให้กับการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกล ดังต่อไปนี้

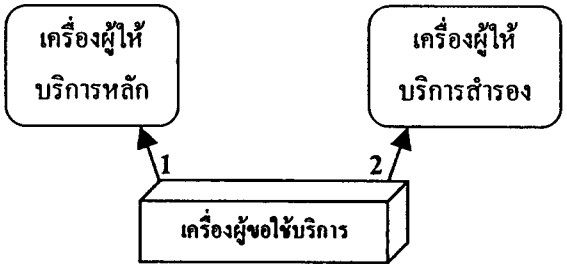
2.2 วิธีเพิ่มความน่าเชื่อถือของรีโมตเมนเมมโมรี

วิธีเพิ่มความน่าเชื่อถือของรีโมตเมนเมมโมรี 4 วิธี คือ มิเรอร์ริง (Mirroring), พาริตี (Parity), พาริตีล็อกกิง (Parity Logging) และดีเลย์พาริตี (Delay Parity) แต่ละวิธีมีรายละเอียดดังต่อไปนี้

2.2.1 มิเรอร์ริง [7]

เป็นการใช้หลักของความซ้ำซ้อน (Redundancy) ที่ง่ายที่สุด คือการสำรองข้อมูลเพิ่มอีก 1 ชุด ในการทดลองใช้เครื่องผู้ให้บริการอย่างน้อย 2 เครื่องที่มีข้อมูลเหมือนกัน และเมื่อหน่วยความจำหลักมีพื้นที่ไม่เพียงพอที่จะนำเพจใหม่เข้ามา ระบบปฏิบัติการจะเลือกเพจที่จะถูกแทนที่ และส่งเพจดังกล่าวไปยังเครื่องผู้ให้บริการหลักและเครื่องผู้ให้บริการสำรอง แสดงดังรูปที่ 2.1 เมื่อเกิดการผิดพลาดกับเครื่องผู้ให้บริการเครื่องใดก็ตามจะสามารถกู้คืนด้วยการคัดลอกข้อมูลจากเครื่องผู้ให้บริการอีกเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การทำงานของวิธีมิเรอร์ริง

ข้อดี

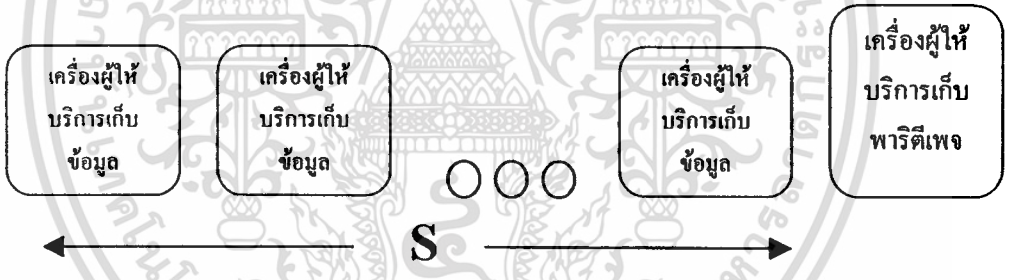
เป็นวิธีที่ง่ายต่อการจัดการ

ข้อเสีย

1. ต้องใช้หน่วยความจำในการสำรองข้อมูลเพิ่มอีก 1 เท่า
2. มีการรับส่งข้อมูลจำนวนมาก

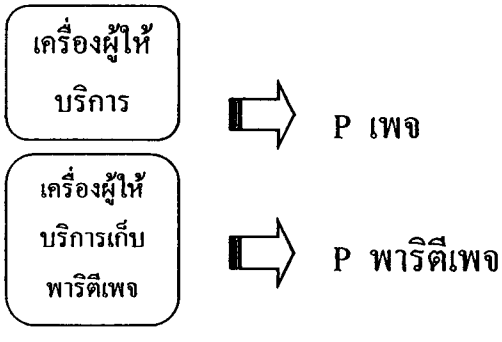
2.2.2 พาริตี [7]

หลักการคือ มีเครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน S เครื่อง และเครื่องผู้ให้บริการที่เก็บพาริตีเพจ (Parity Server) อีก 1 เครื่อง แสดงดังรูปที่ 2.2



รูปที่ 2.2 เครื่องผู้ให้บริการของวิธีพาริตี

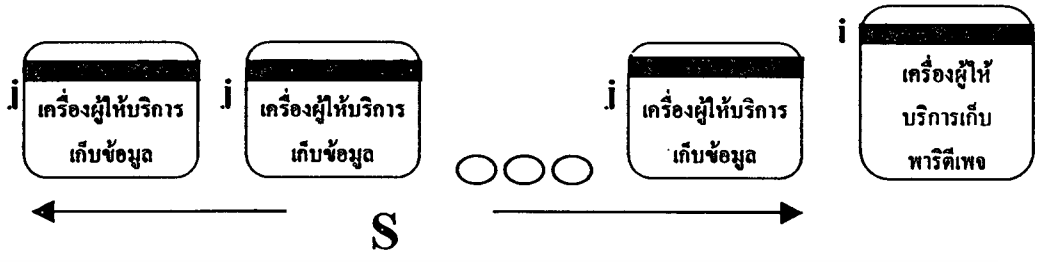
ในกรณีที่เครื่องผู้ให้บริการที่เก็บข้อมูลแต่ละเครื่องมี P เพจ เครื่องผู้ให้บริการที่เก็บพาริตีเพจจะมีพาริตีเพจจำนวน P เพจเช่นกัน แสดงดังรูปที่ 2.3



รูปที่ 2.3 จำนวนเพจของวิธีพาริตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่พาริตีเพจที่ j ถูกสร้างจากการนำข้อมูลเพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลทุกเครื่องทำเอ็กซ์คลูซีฟออร์ (Exclusive OR, XOR) กันแสดงดังรูปที่ 2.4 และตัวอย่างที่ 2.1



เพจที่ j ทั้งหมด เอ็กซ์คลูซีฟออร์กัน

รูปที่ 2.4 การสร้างพาริตีเพจของวิธีพาริตี

ตัวอย่างที่ 2.1

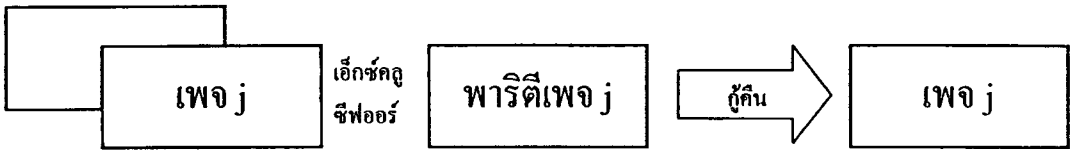
เครือข่ายมีเครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน 3 เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง และเครื่องผู้ขอใช้บริการจำนวน 1 เครื่อง

เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1 ข้อมูลเพจที่ j คือ 10110111
เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 ข้อมูลเพจที่ j คือ 00100110
เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 3 ข้อมูลเพจที่ j คือ 10101110

ข้อมูลเพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1-3 ทำการเอ็กซ์คลูซีฟออร์กันได้พาริตีเพจที่ j คือ 00111111

เมื่อเครื่องผู้ให้บริการใด ๆ เกิดการผิดพลาดก็จะสามารถกู้ข้อมูลคืนโดยทำเอ็กซ์คลูซีฟออร์ระหว่างเพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลทุกเครื่องยกเว้นเครื่องผู้ให้บริการที่เสียหายกับพาริตีเพจกลุ่มเดียวกันได้ แสดงดังรูปที่ 2.5 และ ตัวอย่างที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลทุกเครื่อง
<u>ยกเว้น</u> เครื่องที่เกิดความเสียหาย

เพจที่ได้รับความเสียหาย

รูปที่ 2.5 การกู้คืนเพจของวิธีพาริตี

ตัวอย่างที่ 2.2

เครือข่ายมีเครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน 3 เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง และเครื่องผู้ขอใช้บริการจำนวน 1 เครื่อง

เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1 ข้อมูลเพจที่ j คือ 10110111
เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 ข้อมูลเพจที่ j คือ 00100110
เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 3 ข้อมูลเพจที่ j คือ 10101110

สมมติ ข้อมูลเพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 เสียหาย สามารถกู้คืนได้โดยนำข้อมูลเพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1, 3 และพาริตีเพจมาทำการเอ็กซ์คลูซีฟออร์กัน ดังนี้

เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1 ข้อมูลเพจที่ j คือ 10110111
เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 3 ข้อมูลเพจที่ j คือ 10101110
ทำการเอ็กซ์คลูซีฟออร์กันได้ 00011001

00011001 ทำการเอ็กซ์คลูซีฟออร์กับพาริตีเพจ ได้ 00100110 ซึ่งก็คือ ข้อมูลเพจที่ j ของเครื่อง ผู้ให้บริการเก็บข้อมูลเครื่องที่ 2 นั่นเอง

ในการส่งเพจซึ่งจะถูกแทนที่ไปเก็บในหน่วยความจำหลักของเครื่องผู้ให้บริการที่เก็บข้อมูลสามารถเกิดขึ้นได้ 2 กรณี คือ

1. กรณีที่เพจนั้นยังไม่เคยนำไปวางบนเครื่องผู้ให้บริการที่เก็บข้อมูล จะนำเพจไปวางบนเครื่องผู้ให้บริการตามหลักการวนรอบ และทำการเปลี่ยนแปลงพาริตีเพจ แสดงดังตัวอย่างที่ 2.3

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรณีที่เพจนั้นเคยถูกนำไปวางบนเครื่องผู้ให้บริการที่เก็บข้อมูลแล้ว ต้องทำ 2 ขั้นตอนคือ

2.1 เครื่องผู้ขอใช้บริการส่งเพจซึ่งข้อมูลถูกเปลี่ยนแปลงไปให้เครื่องผู้ให้บริการที่เก็บข้อมูล จากนั้นเครื่องผู้ให้บริการที่เก็บข้อมูลทำการเอ็กซ์คลูซีฟออร์ระหว่างข้อมูลเดิมกับข้อมูลใหม่ ผลลัพธ์ส่งไปยังเครื่องผู้ให้บริการที่เก็บพาริตีเพจเพื่อทำการเปลี่ยนแปลงพาริตีเพจโดยการทำเอ็กซ์คลูซีฟออร์ระหว่างพาริตีเก่ากับผลลัพธ์

2.2 เครื่องผู้ให้บริการที่เก็บข้อมูลนำข้อมูลที่ได้รับมาแทนที่ข้อมูลเดิมแสดงดังตัวอย่างที่ 2.4

ตัวอย่างที่ 2.3

มีเครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน 3 เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง และเครื่องผู้ขอใช้บริการจำนวน 1 เครื่อง เครื่องผู้ขอใช้บริการทำการย้ายเพจไปวางบนเครื่องผู้ให้บริการที่เก็บข้อมูล แสดงดังตารางที่ 2.1

ตารางที่ 2.1 กระบวนการย้ายเพจไปเครื่องผู้ให้บริการ

เพจที่	ข้อมูลของเพจ	ตำแหน่งบนเครื่องผู้ให้บริการ	เครื่องผู้ให้บริการที่	พาริตีเพจที่ j	ข้อมูลพาริตีเพจ j
1	10110111	1	1	1	10110111
2	00100110	1	2	1	10010001
3	10101110	1	3	1	00111111
4	01100111	2	1	2	01100111
5	11100101	2	2	2	10000010
6	01000110	2	3	2	11000100

ตัวอย่างที่ 2.4

มีเครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน 3 เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง และเครื่องผู้ขอใช้บริการจำนวน 1 เครื่อง

เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1 ข้อมูลเพจที่ j คือ 10110111

เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 ข้อมูลเพจที่ j คือ 00100110

เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 3 ข้อมูลเพจที่ j คือ 10101110

ข้อมูลพาริตีเพจที่ j คือ 00111111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องผู้ขอใช้บริการส่งข้อมูลเพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 ซึ่งข้อมูลถูกเปลี่ยนแปลงเป็น 01100011

ขั้นที่ 1

ข้อมูลเก่า เพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 คือ 00100110

ข้อมูลใหม่ที่เครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 ได้รับมา คือ 01100011

ผลลัพธ์จากการทำเอ็กซ์คลูซีฟออร์ระหว่างข้อมูลเก่าและข้อมูลใหม่ คือ 01000101

ข้อมูลเก่า พาริตีเพจที่ j คือ 00111111

ทำการเอ็กซ์คลูซีฟออร์ระหว่าง พาริตีเพจกับผลลัพธ์ ดังนั้น ข้อมูลพาริตีเพจใหม่ คือ 01111010

ขั้นที่ 2

นำข้อมูลใหม่ที่ได้รับมาแทนที่ข้อมูลเดิม ดังนั้น เพจที่ j ของเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 คือ 01100011

ในกรณีที่หน่วยความจำหลักของเครื่องผู้ให้บริการที่เก็บข้อมูลเต็มจะมีข้อความจากเครื่องผู้ให้บริการที่เก็บข้อมูลส่งไปบอกเครื่องผู้ขอใช้บริการ หลังจากนั้นจะเปลี่ยนเป็นทำหน่วยความจำเสมือนแบบสลับหน้าปกติแทน

ข้อดี

แก้ปัญหา เรื่องการใช้เนื้อที่หน่วยความจำในการสำรองข้อมูลของวิธีมิเรอร์ริงได้

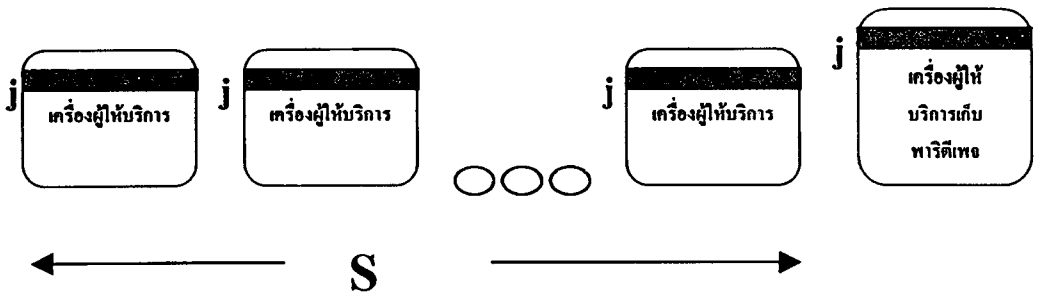
ข้อเสีย

1. มีการรับส่งข้อมูลจำนวนมาก เนื่องจากการปรับปรุงพาริตีเพจ
2. ไม่สามารถนำเพจอื่นไปทับเพจที่เคยอยู่ที่เครื่องผู้ให้บริการได้

2.2.3 พาริตีล็อกกิง [7,9]

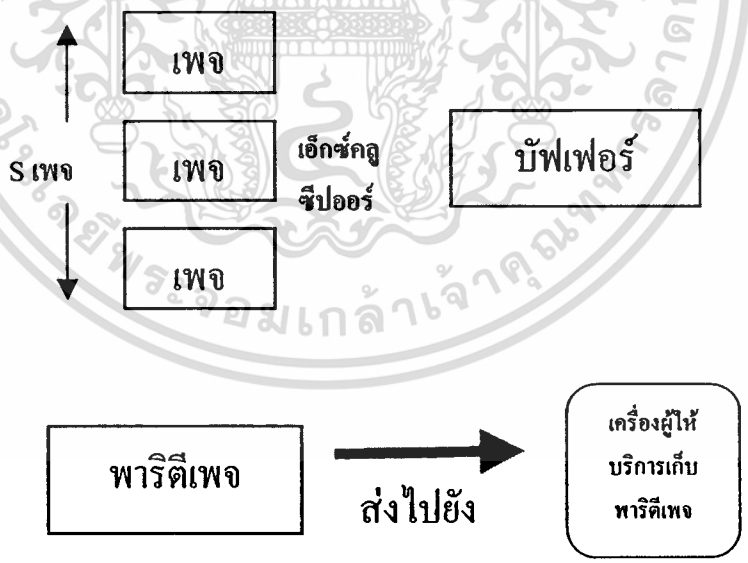
หลักการคล้ายของพาริตี คือ ใช้เครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน S เครื่องและเครื่องผู้ให้บริการที่เก็บพาริตีเพจอีก 1 เครื่อง แสดงดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 พาริตีล็อกกิง

แต่เมื่อเครื่องผู้ขอใช้บริการต้องการนำเพจที่จะถูกแทนที่ไปเก็บยังเครื่องผู้ให้บริการที่เก็บข้อมูล เพจนั้นไม่ถูกจำกัดว่าจะต้องอยู่เครื่องผู้ให้บริการที่เก็บข้อมูลและพาริตีเพจเดิม เนื่องจากเครื่องผู้ขอใช้บริการจะมีบัฟเฟอร์ในการสำรองข้อมูล ซึ่งทุกครั้งที่เครื่องผู้ขอใช้บริการนำเพจที่จะถูกแทนที่ไปเก็บยังเครื่องผู้ให้บริการที่เก็บข้อมูล เครื่องผู้ขอใช้บริการจะทำการเอ็กซ์คลูซีฟออร์ระหว่างเพจนั้นกับข้อมูลในบัฟเฟอร์ซึ่งมีค่าเริ่มต้นเป็น 0 ก่อน เมื่อทำการย้ายเพจครบ S เพจ ข้อมูลพาริตีเพจที่เก็บในบัฟเฟอร์ก็จะถูกส่งไปยังเครื่องผู้ให้บริการที่เก็บพาริตีเพจ ซึ่งจะถือเป็นพาริตีกลุ่มใหม่ แสดงดังรูปที่ 2.7 และตัวอย่างที่ 2.5



รูปที่ 2.7 การคำนวณพาริตีเพจของพาริตีล็อกกิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.5

มีเครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน 3 เครื่อง และเครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง

ข้อมูลเพจลำดับที่ 1 ที่จะถูกนำไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1 คือ 11001101

ข้อมูลเพจลำดับที่ 2 ที่จะถูกนำไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 คือ 00111011

ข้อมูลเพจลำดับที่ 3 ที่จะถูกนำไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 3 คือ 01110010

เริ่มต้นข้อมูลในบัฟเฟอร์คือ 00000000

เครื่องผู้ขอใช้บริการทำการเปลี่ยนแปลงข้อมูลในบัฟเฟอร์ ดังนี้

เมื่อข้อมูลเพจลำดับที่ 1 ถูกนำไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1 ข้อมูลในบัฟเฟอร์ คือ 11001101

เมื่อข้อมูลเพจลำดับที่ 2 ถูกนำไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 2 ข้อมูลในบัฟเฟอร์ คือ 11110110

เมื่อข้อมูลเพจลำดับที่ 3 ถูกนำไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 3 ข้อมูลในบัฟเฟอร์ คือ 10000100

เมื่อเพจถูกนำไปยังเครื่องผู้ให้บริการครบ 3 เพจ ข้อมูลพาริตีในบัฟเฟอร์จะถูกส่งไปยังเครื่องผู้ให้บริการที่เก็บพาริตีซึ่งมีข้อมูลคือ 10000100

เมื่อเครื่องผู้ให้บริการเครื่องใดเกิดความเสียหายจะสามารถกู้คืนได้เหมือนวิธีพาริตี และเมื่อเพจบนเครื่องผู้ให้บริการที่เก็บข้อมูลถูกนำกลับมายังเครื่องผู้ขอใช้บริการ พาริตีเพจกลุ่มที่มีเพจดังกล่าวจะระบุเพจที่ถูกนำกลับไปเครื่องผู้ขอใช้บริการนั้นเป็นอินแวลิด (Invalid) และเมื่อเพจทั้งหมดในพาริตีเพจเดียวกันเป็นอินแวลิด พื้นที่ที่เก็บเพจเหล่านี้บนเครื่องผู้ให้บริการที่เก็บข้อมูลและเครื่องผู้ให้บริการที่เก็บพาริตีเพจก็สามารถนำกลับมาใช้ใหม่ได้

ข้อดี

สามารถลดจำนวนการรับส่งข้อมูลจากการปรับปรุงพาริตีเพจได้

ข้อเสีย

พื้นที่หน่วยความจำหลักของเครื่องผู้ให้บริการที่เก็บเพจที่เป็นอินแวลิดจะสามารถแทนที่ได้ก็ต่อเมื่อเพจทั้งหมดในพาริตีเพจเดียวกันเป็นอินแวลิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เพจจะสามารถถูกแทนที่ได้ก็ต่อเมื่อเพจทั้งหมดในพาริตีเพจเดียวกันเป็นอินแวลิดทั้งหมด วิธี 2 วิธีที่สามารถนำมาแก้ปัญหานี้ คือ ล็อกกิงกรีดดี (Logging – Greedy) และล็อกกิงเซต (Logging – Set)

2.2.3.1 ล็อกกิงกรีดดี [9] พัฒนามาจากพาริตีล็อกกิงโดยเพิ่มหลักการดังนี้

1. เมื่อเครื่องผู้ขอใช้บริการนำเพจที่อยู่บนเครื่องผู้ให้บริการกลับเข้าสู่หน่วยความจำหลักของตน เพจนี้บนเครื่องผู้ให้บริการจะถูกระบุว่าเป็นอินแวลิด และหากเพจในพาริตีเพจเดียวกันเป็นอินแวลิดทั้งหมดแล้ว พื้นที่ที่เก็บข้อมูลส่วนนี้จะสามารถนำเพจมาแทนที่ได้ ซึ่งข้อมูลของการระบุว่าพื้นที่ใดว่างและบิทที่แสดงว่าเป็นอินแวลิดอยู่ที่เครื่องผู้ขอใช้บริการเพื่อลดการส่งข้อมูลโดยไม่จำเป็น

2. เมื่อหน่วยความจำหลักของเครื่องผู้ให้บริการมีพื้นที่ไม่เพียงพอ ระบบจะค้นหาพาริตีเพจที่มีจำนวนเพจที่ระบุว่าอินแวลิดมากที่สุด ต่อจากนั้นเมื่อเครื่องผู้ขอใช้บริการต้องการนำเพจใด ๆ มาวางที่เครื่องผู้ให้บริการ ก็จะนำเพจดังกล่าววางแทนที่เพจที่ถูกระบุว่าเป็นอินแวลิดของพาริตีเพจที่เลือกไว้

ข้อดี

สามารถนำเพจอื่นไปทับเพจที่เคยอยู่ที่เครื่องผู้ให้บริการได้โดยที่ไม่ต้องให้เพจทั้งหมดในพาริตีเพจเดียวกันเป็นอินแวลิดซึ่งวิธีพาริตีและพาริตีล็อกกิงทำไม่ได้

ข้อเสีย

เครื่องผู้ขอใช้บริการต้องมีพื้นที่ในการเก็บข้อมูลที่แสดงถึงสถานะของเพจ

2.2.3.2 ล็อกกิงเซต [9] พัฒนามาจากพาริตีล็อกกิง โดยเพิ่มหลักการคือ

1. เหมือนหลักการข้อ 1. ของ ล็อกกิงกรีดดี

2. เมื่อหน่วยความจำหลักของเครื่องผู้ให้บริการมีพื้นที่ไม่เพียงพอ ระบบจะค้นหาพาริตีเพจที่มีจำนวนเพจที่ระบุว่าอินแวลิดมากที่สุดและจะทำการย้ายเพจที่เป็นแวลิด (Valid) ไปยังพาริตีเพจที่ยังไม่เต็ม และสร้างพาริตีเพจใหม่ขึ้นมา

ข้อดี

สามารถหาพื้นที่ว่างสำหรับเพจที่จะนำเข้ามา โดยไม่ต้องรอให้ทุกเพจในพาริตีเพจเดียวกันเป็นอินแวลิดซึ่งวิธีพาริตีและพาริตีล็อกกิงทำไม่ได้

ข้อเสีย

เครื่องผู้ขอใช้บริการต้องมีพื้นที่ในการเก็บข้อมูลที่แสดงถึงสถานะของเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 ดีเลย์พาริตี [9]

จากการสังเกตการสร้างความน่าเชื่อถือแบบพาริตีนั้น พบว่าเพจที่อยู่ในพาริตีเพจเดียวกันมีความน่าจะเป็นที่จะถูกส่งออกไปเก็บยังเครื่องผู้ให้บริการพร้อม ๆ กัน ดังนั้นจึงเกิดหลักการทำงานใหม่ คือ เมื่อเครื่องผู้ขอใช้บริการส่งเพจที่อยู่ในพาริตีเพจเดียวกันไปยังเครื่องผู้ให้บริการ เครื่องผู้ขอใช้บริการจะคำนวณค่าพาริตีเพจและจะทำการส่งพาริตีเพจไปยังเครื่องผู้ให้บริการที่เก็บพาริตีเพจครั้งเดียว คือ เมื่อเพจสุดท้ายในพาริตีเพจนั้นถูกส่งออกไป จะไม่ทำการเปลี่ยนแปลงพาริตีเพจทุก ๆ ครั้งที่ส่งเพจไปเครื่องผู้ให้บริการเหมือนวิธีพาริตี ส่วนกรณีที่เพจในพาริตีเพจเดียวกันไม่ถูกส่งไปยังเครื่องผู้ให้บริการอย่างต่อเนื่องวิธีนี้จะมีประสิทธิภาพดีกว่าวิธีพาริตีในบางกรณี แสดงดังตัวอย่างที่ 2.6 และ 2.7

ตัวอย่างที่ 2.6

มีเครื่องผู้ให้บริการจำนวน 10 เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง และเครื่องผู้ขอใช้บริการจำนวน 1 เครื่อง เพจทั้งหมดเก็บไว้ที่เครื่องผู้ให้บริการ ดังนั้นจะมีเพจจำนวน 10 เพจที่อยู่ในพาริตีเดียวกัน

สมมติ ให้เครื่องผู้ขอใช้บริการส่งเพจที่ถูกเปลี่ยนแปลงกลับไปยังเครื่องผู้ให้บริการจำนวน 7 เพจจาก 10 เพจ

เครื่องผู้ขอใช้บริการทำการเปลี่ยนแปลงค่าพาริตีเพจของเพจกลุ่มนี้ จากการเอ็กซ์คลูซีฟออร์ แสดงดังตัวอย่างที่ 2.4 ด้วยเพจทั้ง 7 เพจที่ส่งออกไป และส่งพาริตีเพจที่ทำการเปลี่ยนแปลงเสร็จแล้วไปยังเครื่องผู้ให้บริการที่เก็บพาริตีเพจ พร้อมทั้งส่งข้อความไปบอกเครื่องผู้ให้บริการที่เหลือส่งเพจของตนที่อยู่ในพาริตีเพจเดียวกันนั้นไปให้เครื่องผู้ให้บริการที่เก็บพาริตีเพจเพื่อให้ปรับปรุงพาริตีเพจให้สมบูรณ์ ดังนั้น จำนวนการรับส่งข้อมูลทั้งหมด คือ 11 ครั้ง (จากการส่งเพจ 7 ครั้ง เครื่องผู้ให้บริการ 3 ครั้ง และส่งพาริตีเพจจากเครื่องผู้ขอใช้บริการ) ในขณะที่พาริตีจะต้องมีการรับส่งข้อมูล 14 ครั้ง (แต่ละเพจมีการส่งข้อมูล 2 ครั้ง ดังนั้น 7*2 เท่ากับ 14) จากการที่เครื่องผู้ขอใช้บริการส่งเพจไปยังเครื่องผู้ให้บริการ 7 เพจ

ตัวอย่างที่ 2.7

มีเครื่องผู้ให้บริการจำนวน 10 เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง และเครื่องผู้ขอใช้บริการจำนวน 1 เครื่อง เพจทั้งหมดเก็บไว้ที่เครื่องผู้ให้บริการ

เครื่องผู้ขอใช้บริการส่งเพจออกไปยังเครื่องผู้ให้บริการต่อเนื่องกัน 3 เพจ วิธีดีเลย์พาริตีจะมีการรับส่งเพจจำนวน 11 ครั้งเท่าเดิม แต่วิธีพาริตีจะทำการรับส่งเพจเพียง 6 ครั้งเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดี

ลดจำนวนการรับส่งเพจ และเวลาของการแสวงหาซึ่งวิธีนี้เหมาะกับระบบที่มีเครื่องผู้ให้บริการมาก ๆ และโปรแกรมมีความต้องการเพจที่ต่อเนื่องกัน

ข้อเสีย

กรณีที่เครื่องผู้ขอใช้บริการส่งเพจในพาริตีเดียวกันออกไปต่อเนื่องกันน้อยกว่าครึ่งหนึ่งของจำนวนเครื่องผู้ให้บริการจะทำให้มีการรับส่งเพจมากกว่าวิธีพาริตี

ผลการทดลองของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลแบบรีโมตเมนเมมโมรี

ผลการทดลองที่พบใน[7,8]มี 2 ประการหลัก ๆ คือ

1. เมื่อทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกล บนเครือข่ายที่มีเครื่องผู้ให้บริการทำงานอย่างอื่นควบคู่ไปด้วยจะทำให้ประสิทธิภาพลดลงจนมีค่าต่ำกว่าประสิทธิภาพของการทำหน่วยความจำเสมือนแบบสลับหน้าปกติ

2. เรื่องความเร็วในการรับส่งข้อมูลภายในเครือข่าย มีผู้วิจัยทำแบบจำลอง [7] เปรียบเทียบการทำงานระหว่างหน่วยความจำเสมือนแบบสลับหน้าระยะไกลกับการทำหน่วยความจำเสมือนแบบสลับหน้าปกติโดยเพิ่มความเร็วในการรับส่งข้อมูลของระบบเครือข่ายจาก 1 – 100 เมกะไบต์ต่อวินาที และฮาร์ดดิสก์ มีความเร็วคงที่คือ 5 เมกะไบต์ต่อวินาที ผลสรุปว่า

2.1 ถ้าเครือข่ายมีความเร็วในการรับส่งเท่ากับหรือมากกว่าฮาร์ดดิสก์ หน่วยความจำเสมือนแบบสลับหน้าระยะไกลจะมีเวลาเฉลี่ยในการเข้าถึงข้อมูลน้อยกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ

2.2 หน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายที่มีความเร็วในการรับส่งข้อมูลผ่านเครือข่ายประมาณ 25 – 100 เมกะไบต์ต่อวินาที จะใช้เวลาเฉลี่ยในการเข้าถึงข้อมูลใกล้เคียงกัน

2.3 วิธีเพิ่มความน่าเชื่อถือของรีโมตแรมดิสก์

วิธีเพิ่มความน่าเชื่อถือของรีโมตแรมดิสก์ 2 วิธี คือ พาริตีแคชชิง (Parity Caching) และอแดปทีฟพาริตีแคชชิง(Adaptive Parity Caching) แต่ละวิธีมีรายละเอียดดังต่อไปนี้

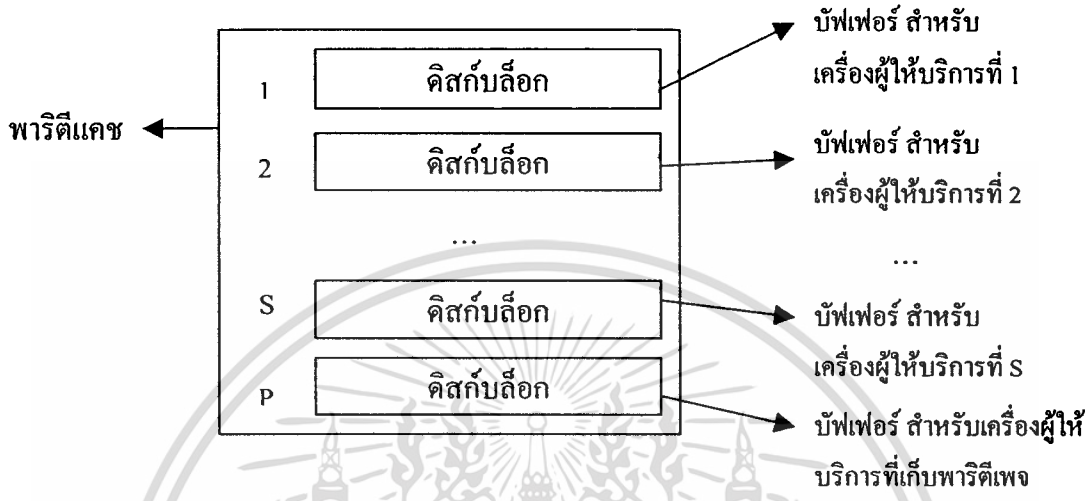
2.3.1 พาริตีแคชชิง [11]

มีหลักการคือ กำหนดให้หน่วยความจำหลักของเครื่องผู้ขอใช้บริการส่วนหนึ่งใช้ในการเก็บข้อมูลชั่วคราว เรียกว่าพาริตีแคช (Parity Cache) ซึ่งภายในประกอบด้วยดิสก์บล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

(Disk Blocks) ทำหน้าที่เสมือนเป็นบัฟเฟอร์สำหรับแต่ละเครื่องผู้ให้บริการ สมมติ เครือข่ายประกอบด้วยเครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน S เก็บพาริตีเพจอีก 1 เครื่อง จะมีพาริตีแคช แสดงดังรูปที่ 2.8



รูปที่ 2.8 พาริตีแคช

เมื่อเครื่องผู้ขอใช้บริการต้องการส่งบล็อกไปยังเครื่องผู้ให้บริการที่เก็บข้อมูล เครื่องผู้ขอใช้บริการจะบรรจุบล็อกลงบัฟเฟอร์สำหรับเครื่องผู้ให้บริการที่ 1 แล้วจึงใส่บัฟเฟอร์สำหรับเครื่อง ผู้ให้บริการที่ 2, 3 ตามหลักการแทนที่แบบวนรอบ เมื่อบัฟเฟอร์ S เต็ม เครื่องผู้ขอใช้บริการจะคำนวณพาริตีเพจ และทำการส่งข้อมูลของทุก ๆ บัฟเฟอร์ไปยังเครื่องผู้ให้บริการต่าง ๆ จากนั้นจะทำการลบข้อมูลในบัฟเฟอร์ทั้งหมด

ข้อดี

1. ลดจำนวนครั้งในการรับส่งข้อมูลภายในเครือข่าย
2. มีเวลาแฝงน้อยกว่าพาริตีล็อกกิง เพราะพาริตีแคชชิงจะรับส่งทีละบล็อกแต่พาริตีล็อกกิงรับส่งทีละเพจ
3. หากเครื่องผู้ให้บริการเกิดความเสียหาย ข้อมูลที่ยังไม่ได้ทำการเปลี่ยนแปลงก็ยังคงอยู่เพราะพาริตีแคชอยู่ที่เครื่องผู้ขอใช้บริการ

ข้อเสีย

ต้องใช้พื้นที่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการสำหรับพาริตีแคช

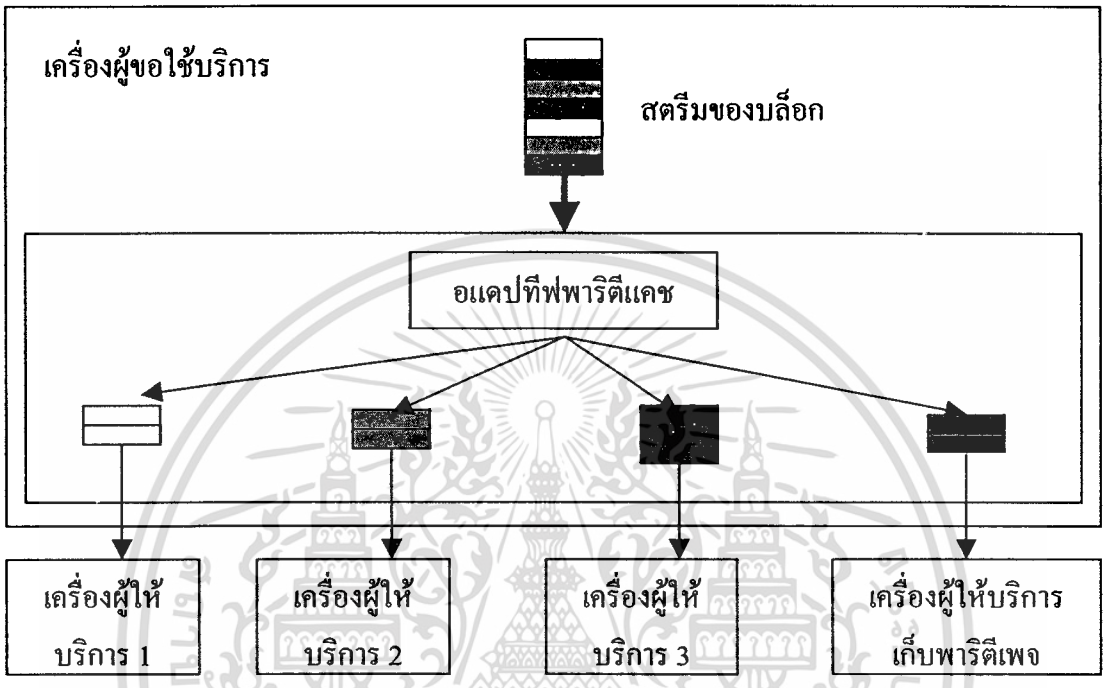
2.3.2 อแดปทีฟพาริตีแคชชิง [11]

จากความรู้ที่ว่า ระบบไฟล์จะพยายามจัดเก็บข้อมูลของไฟล์เดียวกันไว้ในพื้นที่ติดกัน ดังนั้นหน่วยความจำหลักของเครื่องผู้ให้บริการแต่ละเครื่องควรจะมีดิสก์บล็อกที่ต่อเนื่องกัน จึง

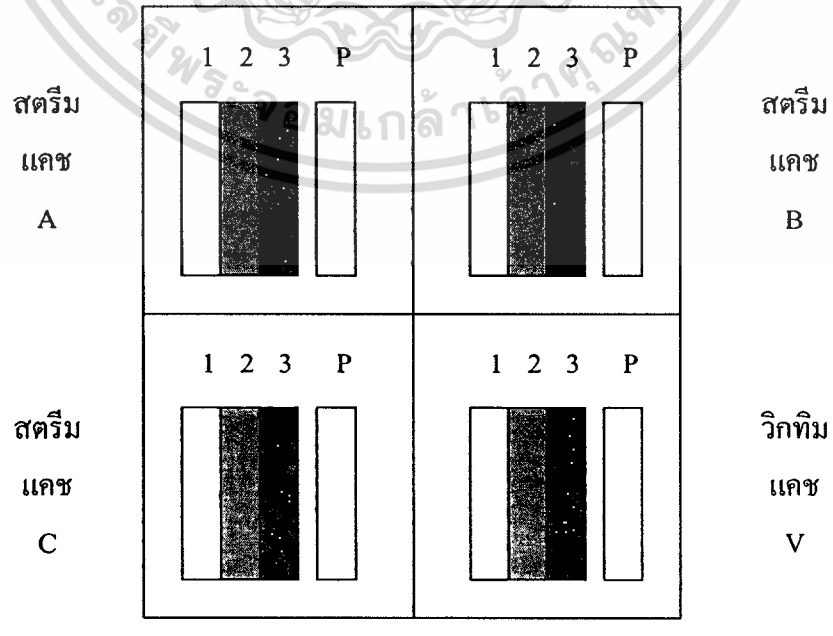
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

61032

สร้างอแดปทีฟพาริตีแคช (Adaptive Parity Cache) ซึ่งประกอบด้วย 3 สตรีมแคช (Stream Cache) และ 1 วิกทิมแคช (Victim Cache) 1 ที่เป็นกลุ่มของบล็อกที่ใช้ในการจัดเรียงบล็อกดิสก์ให้ต่อเนื่องกัน ก่อนส่งออกไปยังเครื่องผู้ให้บริการแต่ละเครื่อง แสดงดังรูปที่ 2.9 และ 2.10



รูปที่ 2.9 อแดปทีฟพาริตีแคชชิง



รูปที่ 2.10 อแดปทีฟพาริตีแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเครื่องผู้ขอใช้บริการสามารถเลือกวางบล็อกที่ต้องการส่งไปเก็บที่เครื่องผู้ให้บริการลงในแคชใด ๆ ตามเงื่อนไขต่อไปนี้

1. ถ้าสตรีมแคชทั้งหมดว่าง จะวางบล็อกดังกล่าวลงในบัฟเฟอร์แรกสุดของสตรีมแคชแรก

2. ถ้าสตรีมแคชไม่ว่างจะพิจารณาต่อดังนี้

2.1 ตรวจสอบสตรีมแคชที่ไม่ว่างทั้งหมด ว่าบล็อกดังกล่าวต่อเนื่องกับบล็อกเดิมที่มีอยู่ในสตรีมแคชใด ๆ หรือไม่ ถ้าบล็อกดังกล่าวต่อเนื่องกับบล็อกเดิมที่มีอยู่ในสตรีมแคชให้วางบล็อกดังกล่าวลงในบัฟเฟอร์ที่ว่างอันแรกของสตรีมแคชนั้น

2.2 ถ้าบล็อกดังกล่าวไม่ต่อเนื่องกับบล็อกเดิมในสตรีมแคชใด ๆ แต่มีสตรีมแคชอื่นที่บัฟเฟอร์ทั้งหมดว่าง ให้นำบล็อกดังกล่าวใส่ในบัฟเฟอร์แรกของสตรีมแคชนั้น

2.3 ถ้าบล็อกดังกล่าวไม่ต่อเนื่องกับบล็อกเดิมในสตรีมแคชใด ๆ และไม่มีสตรีมแคชอื่นที่บัฟเฟอร์ทั้งหมดว่าง ให้นำบล็อกดังกล่าวไปใส่บัฟเฟอร์ที่ว่างอันแรกในวิกทิมแคช

เมื่อวางบล็อกจนเต็มทุกบัฟเฟอร์ของแคชใด ๆ เครื่องผู้ขอใช้บริการจะทำการคำนวณหาพาริตีเพจของบล็อกในแคชดังกล่าว แล้วส่งบล็อกทั้งหมดในแคชนั้นไปยังเครื่องผู้ให้บริการต่าง ๆ จากนั้นจึงลบข้อมูลในบัฟเฟอร์ทั้งหมดในแคช แต่ในกรณีที่วิกทิมแคชเต็มติดต่อกัน 2 ครั้ง จะทำการเติม “0” ในบัฟเฟอร์ที่ว่างทั้งหมดของทุกสตรีมแคช แล้วทำการคำนวณพาริตีและลบข้อมูลในบัฟเฟอร์ของแคชทั้งหมด แสดงดังตัวอย่างที่ 2.8

ตัวอย่างที่ 2.8

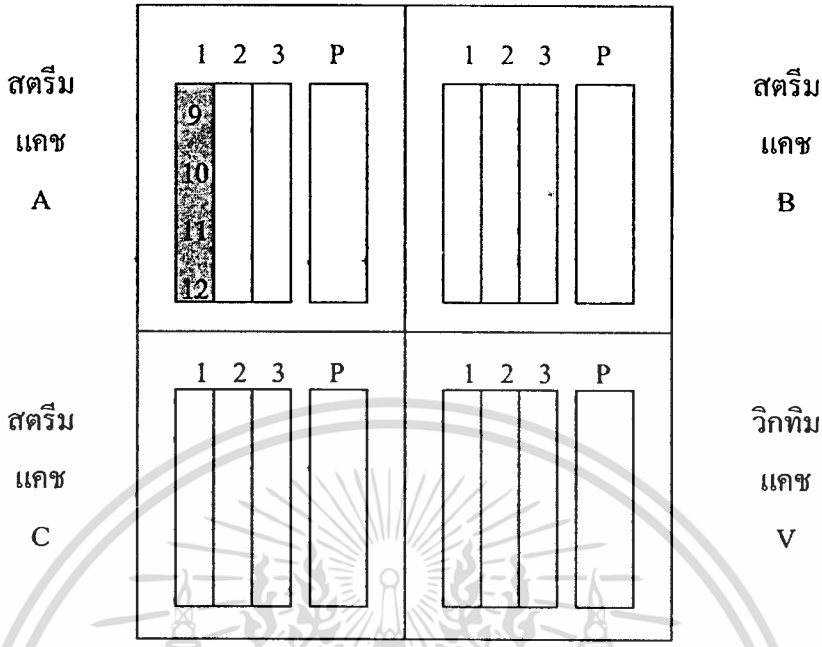
เครื่องผู้ให้บริการที่เก็บข้อมูล 3 เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีเพจ 1 เครื่อง เครื่องผู้ขอใช้บริการจำนวน 1 เครื่อง ลำดับของบล็อกเป็นแสดงดังรูปที่ 2.11 ขั้นตอนการจัดเรียงบล็อกเข้าสตรีมแสดงดังรูปที่ 2.12 ถึง 2.18

9	29	49	93	89	33	37	77	5	17	53	21
10	30	50	94	90	34	38	78	6	18	54	22
11	31	51	95	91	35	39	79	7	19	55	23
12	32	52	96	92	36	40	80	8	20	56	24

รูปที่ 2.11 ลำดับของบล็อกที่โปรแกรมต้องการใช้ในการประมวลผล

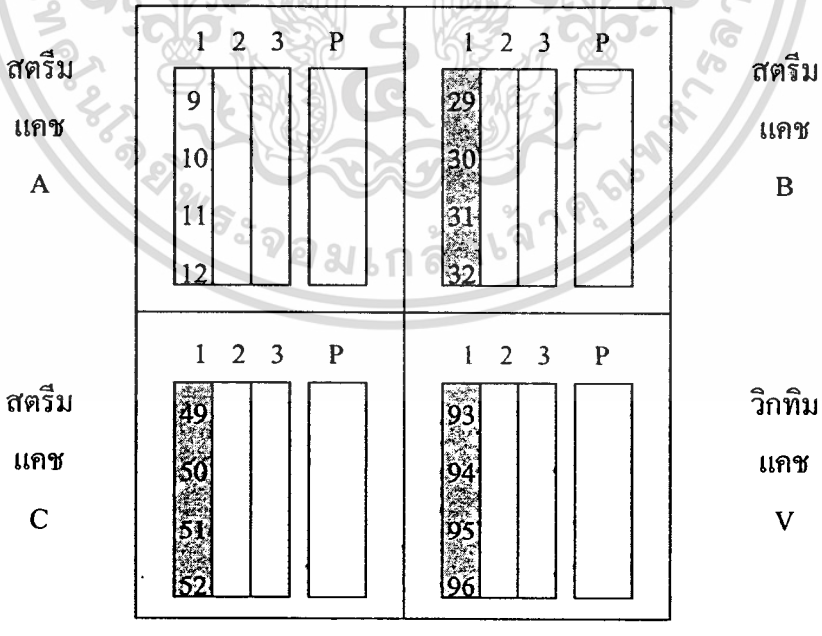
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 1 | 9 10 11 12 |



รูปที่ 2.12 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 1

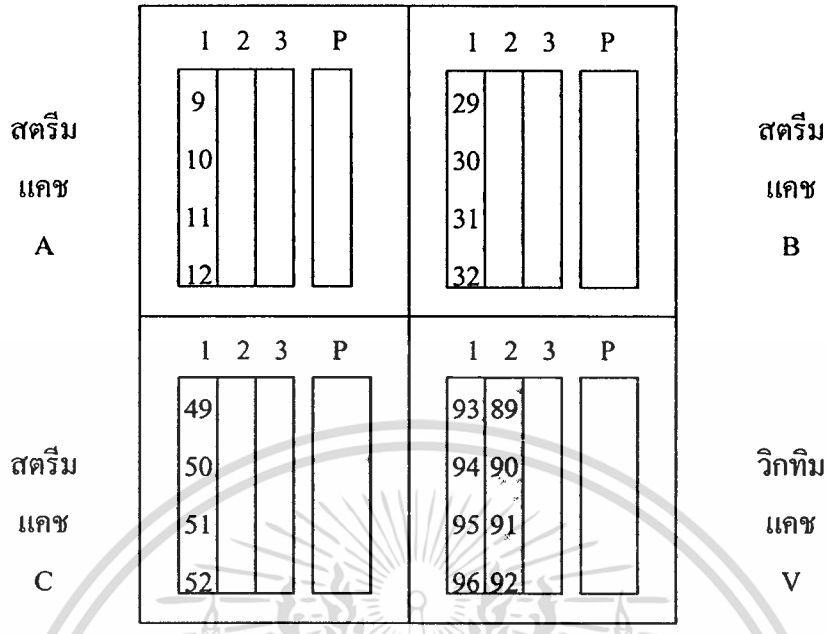
ขั้นที่ 2-4 | 29 30 31 32 | 49 50 51 52 | 93 94 95 96 |



รูปที่ 2.13 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 2-4

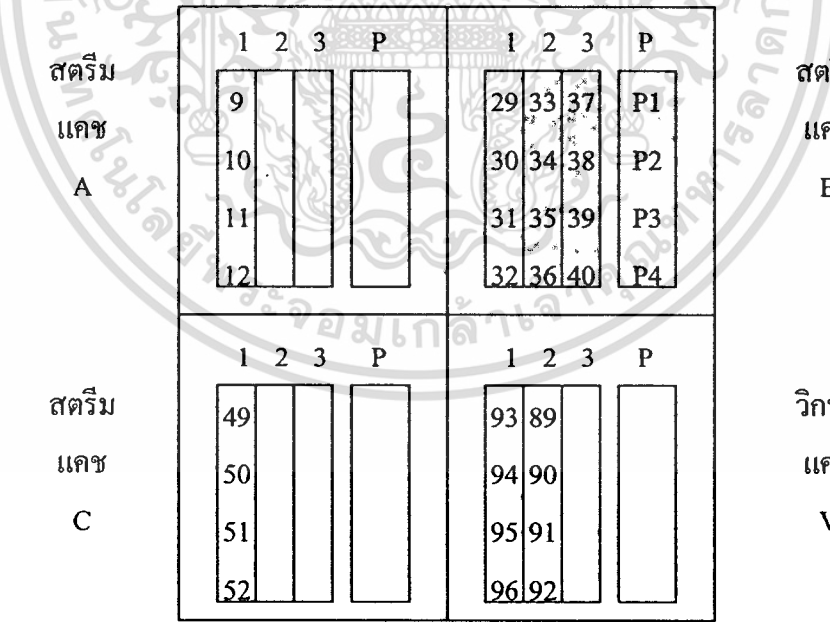
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 5 | 89 90 91 92 |



รูปที่ 2.14 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 5

ขั้นที่ 6-7 | 33 34 35 36 | | 37 38 39 40 |

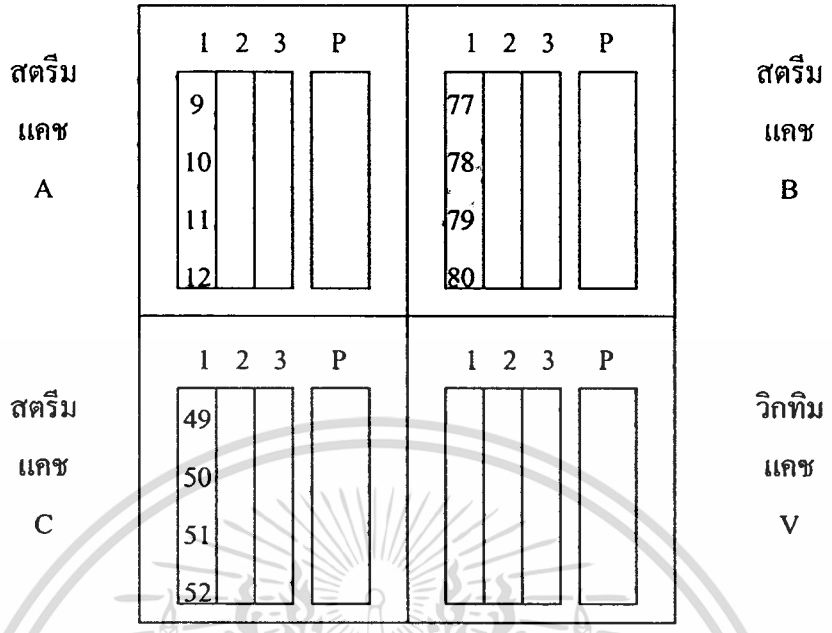


รูปที่ 2.15 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 6-7

ส่ง | 29 30 31 32 | ไปเครื่องผู้ให้บริการเครื่องที่ 1 | 33 34 35 36 | ไปเครื่องผู้ให้บริการเครื่องที่ 2
| 37 38 39 40 | ไปเครื่องผู้ให้บริการเครื่องที่ 3 | P1 P2 P3 P4 | ไปเครื่องผู้ให้บริการที่เก็บพาริตี

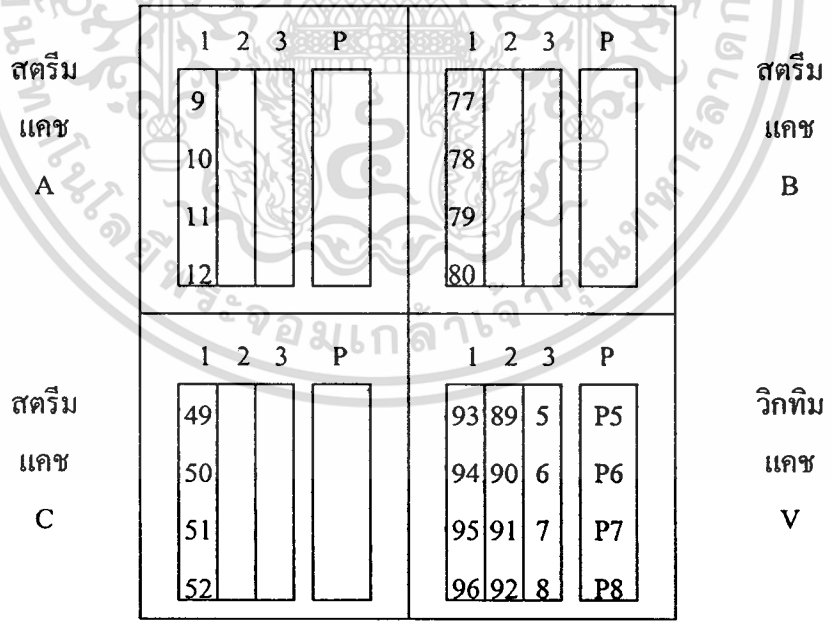
เอกสารนี้เป็นลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 8 | 77 78 79 80 |



รูปที่ 2.16 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 8

ขั้นที่ 9 | 5 6 7 8 |

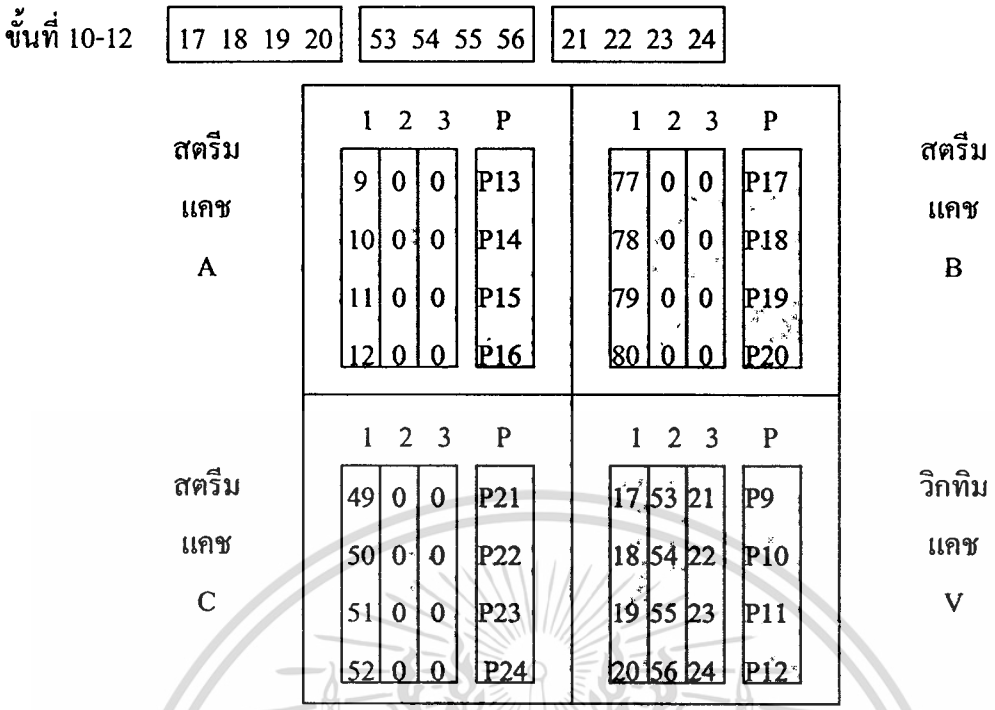


รูปที่ 2.17 แสดงการวางบล็อกตามหลักอแดปทีฟพาริตีแคชชิงขั้นที่ 9

ส่ง | 93 94 95 96 | ไปเครื่องผู้ให้บริการเครื่องที่ 1 | 89 90 91 92 | ไปเครื่องผู้ให้บริการเครื่องที่ 2

| 5 6 7 8 | ไปเครื่องผู้ให้บริการเครื่องที่ 3 | P5 P6 P7 P8 | ไปเครื่องผู้ให้บริการที่เก็บพาริตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงการวางบล็อกตามหลักอแคปทีฟพาริตีแคชชิงขั้นที่ 10-12

ใส่ 0 ลงในบัฟเฟอร์ว่างทั้งหมดในสตรีมแคชทั้งหมด คำนวณพาริตี ส่งข้อมูลไปเครื่องผู้ให้บริการต่าง ๆ แสดงดังตารางที่ 2.2

ตารางที่ 2.2 การส่งข้อมูลไปเครื่องผู้ให้บริการต่าง ๆ ของตัวอย่างที่ 2.8 ขั้นที่ 12

เครื่องผู้ให้บริการ เครื่องที่ 1	เครื่องผู้ให้บริการ เครื่องที่ 2	เครื่องผู้ให้บริการ เครื่องที่ 3	เครื่องผู้ให้บริการ เก็บพาริตีเพจ
9 10 11 12	0 0 0 0	0 0 0 0	P13 P14 P15 P16
77 78 79 80	0 0 0 0	0 0 0 0	P17 P18 P19 P20
49 50 51 52	0 0 0 0	0 0 0 0	P21 P22 P23 P24
17 18 19 20	53 54 55 56	21 22 23 24	P9 P10 P11 P12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดี

1. สามารถลดจำนวนการรับส่งข้อมูลผ่านเครือข่ายได้มากกว่าวิธีที่กล่าวมาข้างต้น

2. มีเวลาแฝงน้อยกว่าพาริตีล็อกกิง เพราะอแดปทีฟพาริตีแคชชิงจะรับส่งทีละบล็อกแต่พาริตีล็อกกิงรับส่งทีละเพจ

3. หากเครื่องผู้ให้บริการเกิดความเสียหาย ข้อมูลที่ยังไม่ได้ทำการเปลี่ยนแปลงก็ยังคงอยู่เพราะพาริตีแคชอยู่ที่เครื่องผู้ขอใช้บริการ

ข้อเสีย

ต้องใช้พื้นที่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการสำหรับสตรีมแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างแบบจำลอง

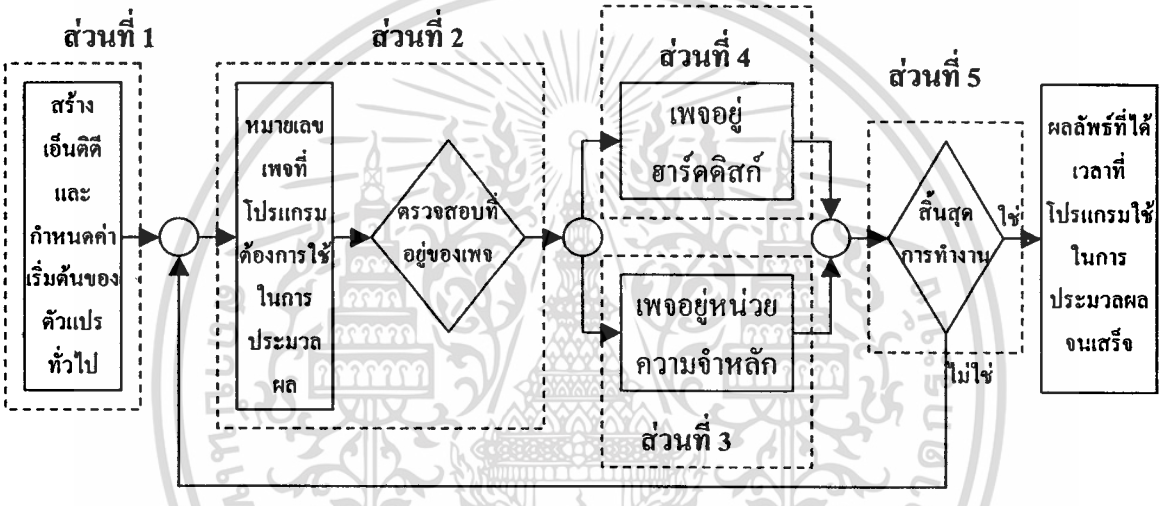
ระบบปฏิบัติการโดยทั่วไปมีการทำงานสำหรับจัดการโปรแกรมให้ได้รับการประมวลผล โดยการโหลดโปรแกรมบางส่วนที่จำเป็นในการประมวลผลเข้าสู่หน่วยความจำหลัก แล้วจัดการให้โปรแกรมได้เข้าทำงานในหน่วยประมวลผลกลาง จากนั้นจะเกิดกิจกรรมตามคำสั่งหรือเรียกว่าโพรเซส (process) ซึ่งในขณะที่โพรเซสกำลังทำงานอยู่หากต้องการข้อมูลใด ๆ ระบบปฏิบัติการจะหาข้อมูลดังกล่าวที่หน่วยความจำหลัก ถ้าไม่พบจะทำการโหลดข้อมูลนี้จากหน่วยความจำสำรอง ซึ่งถ้าเป็นหน่วยความจำเสมือนแบบสลับหน้าปกติ หน่วยความจำสำรอง คือ ฮาร์ดดิสก์ แต่ถ้าเป็นหน่วยความจำเสมือนแบบสลับหน้าระยะไกล หน่วยความจำสำรอง คือ ฮาร์ดดิสก์และหน่วยความจำหลักของเครื่องผู้ให้บริการ เมื่อนำข้อมูลที่ต้องการเข้ามายังหน่วยความจำหลักแล้ว โพรเซสจะทำงานต่อ ซึ่งจะเป็นเช่นนี้จนกว่าโพรเซสทำงานเสร็จ ดังนั้นงานวิจัยนี้ได้สร้างแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าปกติ และแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลด้วยโปรแกรมอารีน่า เวอร์ชัน 5 ซึ่งมี 3 แบบจำลอง คือ แบบจำลองปกติ แบบจำลองระยะไกล และแบบจำลองระยะไกลแบบจำกัด โดยที่ทุกแบบจำลองจะประกอบด้วยโมดูลต่าง ๆ ที่เรียงต่อกันตามลำดับการทำงานและมีเอ็นติตี (Entity) ซึ่งทำหน้าที่เป็นตัวดำเนินการ ดังนั้นในแบบจำลองที่สร้างขึ้นจะให้เอ็นติตีแทนโพรเซสของโปรแกรมในระบบจริง ซึ่งจะเคลื่อนที่ไปยังโมดูลต่าง ๆ ของแบบจำลองตามสถานการณ์ที่เกิดขึ้น ซึ่งรายละเอียดของแต่ละแบบจำลองมีดังต่อไปนี้

3.1 แบบจำลองปกติ

แบบจำลองปกติ เป็นแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าปกติที่มีการทำงานเกิดขึ้นภายในเครื่องคอมพิวเตอร์ส่วนบุคคล 1 เครื่อง โดยที่โมดูลในแบบจำลองปกติสามารถแบ่งได้ 5 ส่วนหลัก คือ ส่วนเริ่มต้นการทำงาน, ส่วนสร้างหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล และตรวจสอบที่อยู่ของเพจ, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลัก, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์ และส่วนตัดสินใจว่าสิ้นสุดการทำงาน รายละเอียดของโมดูลต่าง ๆ ในแต่ละส่วนประกอบสามารถดูได้ในภาคผนวก ข. ความสัมพันธ์ของทุกส่วนแสดงดังรูปที่ 3.1 โดยที่การทำงานของแบบจำลองนี้เริ่มขึ้นเมื่อเอ็นติตีถูกสร้างขึ้นมาในส่วนที่ 1 และเข้าสู่ส่วนที่ 2 ซึ่งเอ็นติตีจะได้รับหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล และจะถูกตรวจสอบว่าเพจดังกล่าวอยู่ที่ใด ถ้าเพจอยู่ที่หน่วยความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำหลักเอ็นติตีจะถูกส่งไปยังส่วนที่ 3 แต่ถ้าเพจอยู่ที่ฮาร์ดดิสก์เอ็นติตีจะถูกส่งไปยังส่วนที่ 4 และเมื่อผ่านส่วนที่ 3 หรือส่วนที่ 4 แล้วเอ็นติตีจะเข้าสู่ส่วนที่ 5 เพื่อตัดสินใจว่าสิ้นสุดการทำงานหรือยัง ถ้ายังก็จะวนกลับมายังส่วนที่ 2 อีกครั้งจนกว่าจะไม่ต้องการเพจในการประมวลผลอีกหรือเทียบเท่ากับโพรเซสสิ้นสุดการประมวลผลนั่นเอง ซึ่งถ้าสิ้นสุดการทำงานเอ็นติตีจะถูกทำลายและผลลัพธ์ที่ได้จากแบบจำลอง (Output) คือ เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จซึ่งเป็นเวลาที่เกิดจากการสะสมเวลาที่โปรแกรมใช้ในการประมวลของแต่ละครั้งที่โปรแกรมต้องการใช้เพจในการประมวลผลรายละเอียดของตัวแปรต่าง ๆ ในแบบจำลองสามารถดูได้จากตารางที่ ก.1 ในภาคผนวก ก. ส่วนรายละเอียดของส่วนประกอบหลัก 5 ส่วนในแบบจำลองปกติ มีดังนี้



รูปที่ 3.1 ความสัมพันธ์ของส่วนประกอบต่าง ๆ ของแบบจำลองปกติ

1. ส่วนเริ่มต้นการทำงาน เป็นส่วนที่สร้างเอ็นติตีขึ้นมาเพื่อดำเนินการต่าง ๆ ในแบบจำลองและกำหนดค่าเริ่มต้นให้กับตัวแปรทั่วไปในแบบจำลอง

2. ส่วนสร้างหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล และตรวจสอบที่อยู่ของเพจ เป็นส่วนที่ทำการสุ่มหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผลด้วยฟังก์ชันทางคณิตศาสตร์ จากนั้นก็จะคำนวณว่าเพจดังกล่าวอยู่ในบล็อกใด และตรวจสอบว่าเพจนั้นอยู่ที่หน่วยความจำหลักหรือฮาร์ดดิสก์

3. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลัก เป็นส่วนที่ทำการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ (ตัวแปร totalTime) ซึ่งคำนวณจากเวลาที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก (ตัวแปร ma)

= ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.1

totalTime = totalTime + ma วินาที (3.1)

4. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์ เป็นส่วนที่ทำการคำนวณเวลานำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก และตรวจสอบว่าหน่วยความจำหลักมีที่ว่างเพียงพอสำหรับบล็อกดังกล่าวหรือไม่ ซึ่งแบ่งเป็น 2 กรณี คือ หน่วยความจำหลักมีที่ว่างเพียงพอ และหน่วยความจำหลักมีที่ว่างไม่เพียงพอ รายละเอียดของ 2 กรณี มีดังต่อไปนี้

4.1 หน่วยความจำหลักมีที่ว่างเพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล จึงนำบล็อกดังกล่าวเข้าสู่หน่วยความจำหลัก และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่

(2) เวลาเฉลี่ยที่ใช้ในการนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก (ตัวแปร HDAccess) แสดงดังสมการที่ 3.3

(3) เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

การคำนวณเวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในฮาร์ดดิสก์ (Access time, ตัวแปร HDAccess) ประกอบด้วย 3 ส่วนดังนี้

1. เวลาแสวงหา (Seek Time, ตัวแปร seek) เป็นระยะเวลาที่หัวอ่านและหัวเขียนเคลื่อนที่มายังไซลินเดอร์ (Cylinder) ที่ต้องการ มีค่า 0.008 วินาที [1,12]

2. เวลาแฝง (Latency Time, ตัวแปร Latency) เป็นระยะเวลาที่เซคเตอร์ (Sector) ที่ต้องการหมุนมาถึงหัวอ่านและหัวเขียน คำนวณจากฮาร์ดดิสก์ในปัจจุบันที่มีอัตราการหมุน 7200 รอบต่อนาที ดังนั้นเวลาเฉลี่ยมีค่า 0.0042 วินาที [5]

3. เวลาในการรับส่งข้อมูล 1 บล็อกของฮาร์ดดิสก์ (Transfer Time, ตัวแปร TranHD) ซึ่งการคำนวณ แสดงดังสมการที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{เวลาในการรับส่งข้อมูล [6]} = \frac{\text{ขนาดข้อมูล}}{\text{ความเร็วในการรับส่งข้อมูล}} \text{ วินาที} \quad (3.2)$$

ความเร็วในการรับส่งข้อมูลของฮาร์ดดิสก์ คือ 5 เมกะไบต์ต่อวินาที [7,13] และขนาดของบล็อก คำนวณดังนี้

ขนาดบล็อก = จำนวนเพจใน 1 บล็อก * ขนาดเพจ

= 4*(1024*4) ไบต์

= 16384 ไบต์

ดังนั้น เวลาในการรับส่งข้อมูล $= \frac{16384}{5*1024*1024}$ วินาที

= 0.003125 วินาที

ดังนั้น เวลาที่ใช้ในการเข้าถึงข้อมูลที่อยู่ในฮาร์ดดิสก์ หรือตัวแปร HDAccess คำนวณได้จากการนำ เวลาแสวงหา เวลาแฝง และเวลาในการรับส่งข้อมูล มารวมกัน แสดงดังสมการที่ 3.3

HDAccess = เวลาแสวงหา + เวลาแฝง + เวลาในการรับส่งข้อมูล (3.3)

= 0.008 + 0.0042 + 0.003125

= 0.015325 วินาที

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2) + (3)

= ma + HDAccess + ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.4

totalTime = totalTime + (HDAccess + ma*2) วินาที (3.4)

4.2 หน่วยความจำหลักมีพื้นที่ว่างไม่เพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล จึงทำการเลือกบล็อกที่จะถูกแทนที่ และนำบล็อกนี้กลับลงฮาร์ดดิสก์ ซึ่งเมื่อหน่วยความจำหลักมีพื้นที่ว่างเพียงพอ จึงนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลเข้าสู่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่

(2) เวลาเฉลี่ยที่ใช้ในการนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก ดังแสดงในสมการที่ 3.3

(3) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อทำการเลือกบล็อกที่จะถูกแทนที่

(4) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในฮาร์ดดิสก์ เพื่อนำบล็อกที่จะถูกแทนที่กลับลงฮาร์ดดิสก์ ดังแสดงในสมการที่ 3.3

(5) เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2) + (3) + (4) + (5)

= ma + HDAccess + ma + HDAccess + ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.5

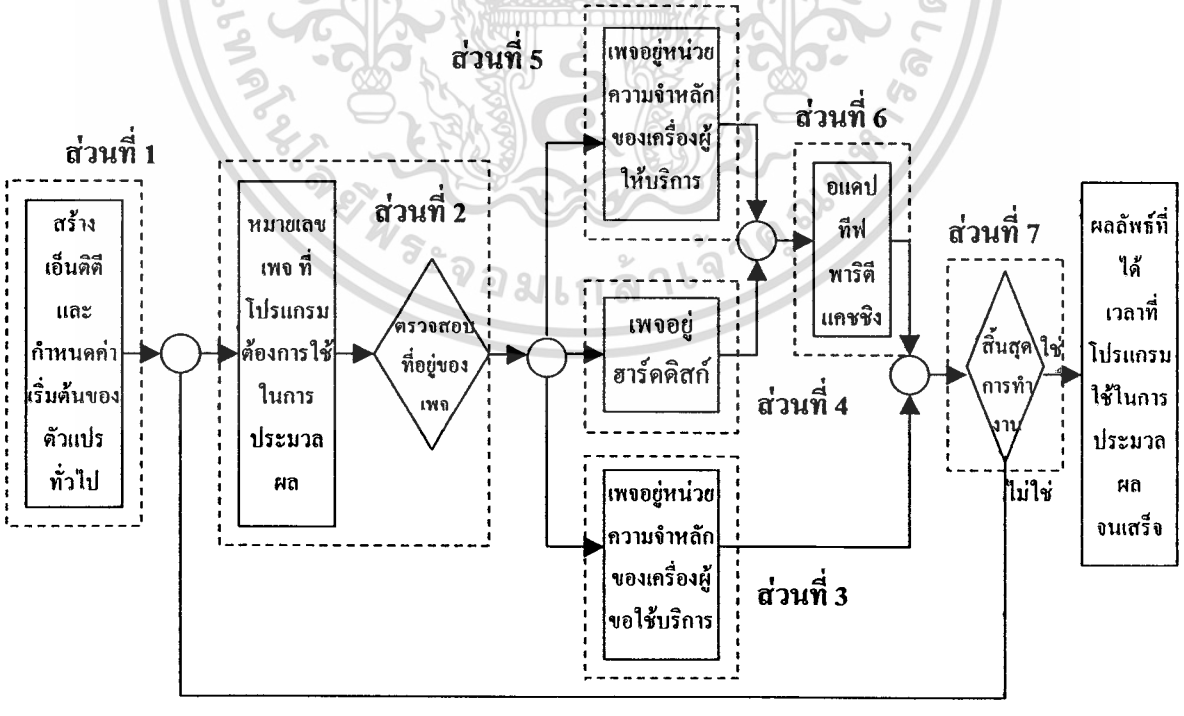
totalTime = totalTime + ((HDAccess*2) + (ma*3)) วินาที (3.5)

5. ส่วนตัดสินใจว่าสิ้นสุดการทำงาน จะเป็นส่วนที่ตรวจสอบว่าเอ็นติตียังต้องการเพจในการประมวลผลอีกหรือไม่ ถ้าใช่เอ็นติตีจะถูกส่งไปยังส่วนที่ 2 อีกครั้ง แต่ถ้าไม่ต้องการเพจแล้วก็จะถือว่าสิ้นสุดการทำงานและแสดงผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 แบบจำลองระยะไกล

แบบจำลองระยะไกล เป็นแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายของคอมพิวเตอร์ส่วนบุคคลซึ่งประกอบด้วยเครื่องผู้ขอใช้บริการ 1 เครื่อง เครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน S เครื่อง และเครื่องผู้ให้บริการที่เก็บพาริตีจำนวน 1 เครื่อง (จำนวนเครื่องผู้ให้บริการทั้งหมด คือ ตัวแปร Nserver) ซึ่งเครือข่ายและทุกเครื่องไม่มีการทำงานอื่นใดนอกจากการทดลองทำให้เครือข่ายมีการรับส่งข้อมูลด้วยความเร็วสูงสุด (ตัวแปร BW) และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ (ตัวแปร CE) โดยโมดูลในแบบจำลองระยะไกลสามารถแบ่งได้ 7 ส่วน คือ ส่วนเริ่มต้นการทำงาน, ส่วนสร้างหมายเลขเพจที่โปรแกรมใช้ในการประมวลผล และตรวจสอบที่อยู่ของเพจ, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อทำอแดปทีฟพาริตีแคชชิง และส่วนตัดสินใจว่าสิ้นสุดการทำงาน ซึ่งแต่ละส่วนภายในจะประกอบด้วยโมดูลต่าง ๆ ต่อกันตามลำดับการทำงาน รายละเอียดของโมดูลต่าง ๆ ในแต่ละส่วนประกอบสามารถดูได้ในภาคผนวก ค. ความสัมพันธ์ของส่วนทั้งหมด แสดงดังรูปที่ 3.2



รูปที่ 3.2 ความสัมพันธ์ของส่วนประกอบต่าง ๆ ของแบบจำลองระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่การทำงานของแบบจำลองนี้เริ่มขึ้นเมื่อเอ็นติตีถูกสร้างขึ้นมาจากส่วนที่ 1 และเข้าสู่ส่วนที่ 2 ซึ่งเอ็นติตีจะได้รับหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล และจะถูกตรวจสอบว่าเพจดังกล่าวอยู่ที่ใด ถ้าเพจอยู่ที่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการเอ็นติตีจะถูกส่งไปยังส่วนที่ 3 ต่อจากนั้นจะเข้าสู่ส่วนที่ 7 ถ้าเพจอยู่ที่ฮาร์ดดิสก์เอ็นติตีจะถูกส่งไปยังส่วนที่ 4 แต่ถ้าเพจอยู่หน่วยความจำหลักของเครื่องผู้ให้บริการเอ็นติตีจะถูกส่งไปยังส่วนที่ 5 ซึ่งเมื่อผ่านส่วนที่ 4 และ 5 แล้วก็จะเข้าสู่ส่วนที่ 6 ก่อน ต่อจากนั้นจะเข้าสู่ส่วนที่ 7 เพื่อตัดสินใจว่าสิ้นสุดการทำงานหรือยัง ถ้ายังก็จะวนกลับมายังส่วนที่ 2 อีกครั้งจนกว่าจะไม่ต้องการเพจในการประมวลผลอีกหรือเทียบเท่ากับโพรเซสสิ้นสุดการประมวลผลนั่นเอง ซึ่งถ้าสิ้นสุดการทำงานเอ็นติตีจะถูกทำลายและผลลัพธ์ที่ได้จากแบบจำลอง คือ เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จซึ่งเป็นเวลาที่เกิดจากการสะสมเวลาที่โปรแกรมใช้ในการประมวลของแต่ละครั้งที่โปรแกรมต้องการใช้เพจในการประมวลผล รายละเอียดของตัวแปรต่าง ๆ ในแบบจำลองสามารถดูได้จากตารางที่ ก.2 ในภาคผนวก ก. ส่วนรายละเอียดของส่วนประกอบหลัก 7 ส่วนในแบบจำลองระยะไกล มีดังนี้

1. ส่วนเริ่มต้นการทำงาน เป็นส่วนที่สร้างเอ็นติตีขึ้นมาเพื่อดำเนินการต่าง ๆ ในแบบจำลองและกำหนดค่าเริ่มต้นให้กับตัวแปรทั่วไปในแบบจำลอง

2. ส่วนสร้างหมายเลขเพจที่โปรแกรมใช้ในการประมวลผล และตรวจสอบที่อยู่ของเพจ เป็นส่วนที่ทำการสุ่มหมายเลขเพจที่โปรแกรมใช้ในการประมวลผลด้วยฟังก์ชันทางคณิตศาสตร์ จากนั้นก็จะคำนวณว่าเพจดังกล่าวอยู่ในบล็อกใด และตรวจสอบว่าเพจนั้นอยู่ที่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการ ฮาร์ดดิสก์ หรือหน่วยความจำหลักของเครื่องผู้ให้บริการ

3. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ ซึ่งมีการทำงานและคำนวณเหมือนกับแบบจำลองปกติ ดังแสดงในสมการที่ 3.1

4. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์ เป็นส่วนที่ทำการคำนวณเวลานำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลักและตรวจสอบว่าหน่วยความจำหลักมีที่ว่างเพียงพอสำหรับบล็อกดังกล่าวหรือไม่ ถ้าพอก็จะนำบล็อกที่มีเพจดังกล่าวอยู่เข้าสู่หน่วยความจำหลัก และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณเหมือนแบบจำลองปกติ ดังแสดงในสมการที่ 3.4 แต่ถ้าหน่วยความจำหลักมีพื้นที่ว่างไม่เพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล ก็จะทำการเลือกบล็อกที่จะถูกแทนที่และนำบล็อกนี้เข้าสู่ขั้นตอนของอเดปทีฟพาริตีแคชชิง (รายละเอียดสามารถดูได้ในบทที่ 2) เมื่อหน่วยความจำหลักมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นที่ว่างเพียงพอจึงนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลเข้าสู่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลักเพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่

(2) เวลาเฉลี่ยที่ใช้ในการนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก ดังแสดงในสมการที่ 3.3

(3) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อทำการเลือกบล็อกที่จะถูกแทนที่

(4) เวลาเฉลี่ยที่ใช้วางบล็อกที่จะถูกแทนที่ตามหลักอแดปทีฟพาริตีแคชชิง ซึ่งจะกล่าวรายละเอียดต่อไปในข้อ 5 จึงไม่รวมเวลาดังกล่าวในการคำนวณส่วนนี้

(5) เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2) + (3) + (5)

= ma + HDAccess + ma + ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.6

totalTime = totalTime + (HDAccess + (ma*3)) วินาที (3.6)

5. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ เป็นส่วนที่ทำการคำนวณเวลาที่นำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากเครื่องผู้ให้บริการกลับมายังหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ และตรวจสอบว่าหน่วยความจำหลักของเครื่องผู้ขอใช้บริการมีที่ว่างเพียงพอสำหรับบล็อกดังกล่าวหรือไม่ ซึ่งแบ่งเป็น 2 กรณี คือ หน่วยความจำหลักมีที่ว่างเพียงพอ และหน่วยความจำหลักมีที่ว่างไม่เพียงพอ รายละเอียดของ 2 กรณี มีดังต่อไปนี้

5.1 หน่วยความจำหลักมีที่ว่างเพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล จึงนำบล็อกดังกล่าวเข้าสู่หน่วยความจำหลัก และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลักเพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) เวลาที่เครื่องผู้ขอใช้บริการส่งคำร้องขอเพจที่ต้องการไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลของเพจที่ต้องการ

(3) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลมายังเครื่องผู้ขอใช้บริการ

(4) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งคำร้องไปยังเครื่องผู้ให้บริการที่เก็บพาริตีเพจเพื่อแก้ไขสถานะภาพเป็นอินแวลิดเพจ

(5) เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

โดยเวลาที่ (1) และ (4) เป็นการส่งคำร้องไม่มีการส่งข้อมูลบล็อก ดังนั้นขนาดของข้อมูลที่ส่งผ่านเครือข่ายจึงคิดแค่ข้อมูลส่วนหัว [6] ทำให้เวลาเฉลี่ยที่ใช้ในรับส่งคำร้องผ่านเครือข่ายแสดงดังสมการที่ 3.7

$$\text{เวลาเฉลี่ยที่ใช้ในการรับส่งคำร้อง} = \frac{\text{ขนาดคำร้อง}}{\text{ความเร็วในการรับส่งผ่านเครือข่าย}}$$

ขนาดคำร้อง = 64 * 8
= 512 บิต

ความเร็วในการรับส่งผ่านเครือข่าย = BW เมกะบิตต่อวินาที
(ตัวแปร BW) = BW*1048576 บิตต่อวินาที

ดังนั้น เวลาเฉลี่ยที่ใช้ในการรับส่งคำร้อง = 512 / (BW*1048576) วินาที (3.7)

เวลาที่ (3) เป็นการรับส่งบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล ดังนั้นการคำนวณเวลาเฉลี่ยที่ใช้ในการรับส่งผ่านเครือข่าย แสดงดังสมการที่ 3.8

$$\text{เวลาเฉลี่ยที่ใช้ในการรับส่งบล็อกผ่านเครือข่าย} = \frac{\text{ขนาดข้อมูล(ตัวแปร MsSize)}}{\text{ความเร็วในการรับส่งผ่านเครือข่าย}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดข้อมูล = ขนาดบล็อก + ขนาดข้อมูลส่วนหัว

= (4*1024*4*8) + (64*8)

= 131072 + 512

= 131584 บิต

ดังนั้น เวลาเฉลี่ยที่ใช้ในการรับส่งบล็อกผ่านเครือข่าย = 131584 / (BW*1048576) วินาที (3.8)

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2) + (3) + (4) + (5)

= ma + (512/(BW*1048576)) +
(131584/(BW*1048576)) +
(512/(BW*1048576)) + ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.9

totalTime = totalTime + ((ma*2) + (((512)/(BW*1048576))*2) +
(131584/(BW*1048576))) วินาที (3.9)

5.2 หน่วยความจำหลักมีที่ว่างไม่เพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล จึงทำการเลือกบล็อกที่จะถูกแทนที่และนำบล็อกนี้เข้าสู่ขั้นตอนของอแดปทีฟพาริตีแคชชิง (รายละเอียดสามารถดูได้ในบทที่ 2) เมื่อหน่วยความจำหลักมีพื้นที่ว่างเพียงพอจึงนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลเข้าสู่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่

(2) เวลาที่เครื่องผู้ขอใช้บริการส่งคำร้องขอเพจที่ต้องการไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลของเพจที่ต้องการ ดังแสดงในสมการที่ 3.7

(3) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลมายังเครื่องผู้ขอใช้บริการ ดังแสดงในสมการที่ 3.8

(4) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งคำร้องไปยังเครื่องผู้ให้บริการที่เก็บพาริตีเพจเพื่อแก้ไขสถานะภาพเป็นอินแวลิดเพจ ดังแสดงในสมการที่ 3.7

(5) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อทำการเลือกบล็อกที่จะถูกแทนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(6) เวลาเฉลี่ยที่ใช้วางบล็อกที่จะถูกแทนที่ตามหลักอแดปทีฟพาริตีแคชชิง ซึ่งจะกล่าวรายละเอียดต่อไปในข้อ 6 จึงไม่รวมเวลาดังกล่าวในการคำนวณส่วนนี้

(7) เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2) + (3) + (4) + (5) + (7)

= ma + (512/(BW*1048576)) +
(131584/(BW*1048576)) +
(512/(BW*1048576))+ ma + ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.10

totalTime = totalTime + ((ma*3) + (((512)/(BW*1048576))*2) +
(131584/(BW*1048576))) วินาที (3.10)

6. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อทำอแดปทีฟพาริตีแคชชิง เป็นส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผล เมื่อจัดวางบล็อกที่จะถูกแทนที่โดยใช้หลักการของอแดปทีฟพาริตีแคชชิง (รายละเอียดสามารถดูได้ในบทที่ 2) ซึ่งจะทำให้เกิดกรณีต่าง ๆ แบ่งเป็น 3 กรณี คือ

6.1 การนำบล็อกที่จะถูกแทนที่ไปใส่ในอแดปทีฟพาริตีแคช ซึ่งมี 3 สตรีมแคช และ 1 วิกทิมแคช และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งแยกเป็น 4 กรณี แสดงดังสมการที่ 3.11 ถึง 3.14

กรณีวางในสตรีมแคชที่ 1 totalTime = totalTime + ma วินาที (3.11)

กรณีวางในสตรีมแคชที่ 2 totalTime = totalTime + ma*2 วินาที (3.12)

กรณีวางในสตรีมแคชที่ 3 totalTime = totalTime + ma*3 วินาที (3.13)

กรณีวางในวิกทิมแคช totalTime = totalTime + ma*4 วินาที (3.14)

6.2 เมื่อจัดวางบล็อกที่จะถูกแทนที่ลงในสตรีมแคชหรือวิกทิมแคช แล้วพบว่าแคชนั้นเต็มและเครื่องผู้ให้บริการมีพื้นที่ว่างเพียงพอ จึงนำบล็อกทั้งหมดในแคชนั้นส่งไปยังเครื่องผู้ให้บริการต่าง ๆ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงหน่วยความจำหลักเพื่อทำการอ่านข้อมูลจำนวน 4 ครั้ง ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผลในขั้นนี้ แสดงดังสมการที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลัก จำนวน 4 ครั้ง

$$= \quad ma*4 \quad \text{วินาที} \quad (3.15)$$

(2) เวลาเฉลี่ยที่ใช้ในการส่งข้อมูล 1 บล็อก ไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1, 2, 3 และเครื่องผู้ให้บริการที่เก็บพาริตีเพจ รวมทั้งหมด 4 ครั้ง ซึ่งการคำนวณเวลาที่ใช้ในการรับส่งข้อมูล 1 บล็อก แสดงในสมการที่ 3.8 ดังนั้นการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลขั้นที่ 2 นี้ แสดงดังสมการที่ 3.16

เวลาเฉลี่ยที่ใช้ในการส่งบล็อก 1 บล็อกไปยังเครื่องผู้ให้บริการจำนวน 4 ครั้ง

$$= \quad (131584/(BW*1048576))*4 \quad \text{วินาที} \quad (3.16)$$

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2)

$$= \quad (ma*4)+((131584/(BW*1048576))*4) \quad \text{วินาที}$$

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.17

$$totalTime = totalTime+((ma*4)+((131584/(BW*1048576))*4)) \quad \text{วินาที} \quad (3.17)$$

6.3 เมื่อจัดวางบล็อกที่จะถูกแทนที่ลงในสตรีมแคชหรือวิกทิมแคช แล้วพบว่าสตรีมแคชหรือวิกทิมแคชนั้นเต็มและเครื่องผู้ให้บริการไม่มีพื้นที่ว่างเพียงพอจึงนำบล็อกที่เป็นข้อมูลไม่รวมพาริตีบล็อกในแคชนั้นกลับลงฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลักเพื่อทำการอ่านข้อมูลจำนวน 3 ครั้ง ดังนั้นเวลาที่โปรแกรมใช้ในการประมวลผลในขั้นนี้ แสดงดังสมการที่ 3.18

เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลัก จำนวน 3 ครั้ง

$$= \quad ma*3 \quad \text{วินาที} \quad (3.18)$$

(2) เวลาเฉลี่ยที่ใช้ในการรับส่งข้อมูล 1 บล็อกของฮาร์ดดิสก์ เพื่อเขียนข้อมูลทับข้อมูลเดิม จำนวน 3 ครั้ง ซึ่งเวลาเฉลี่ยที่ใช้ในการรับส่งข้อมูล 1 บล็อกมีการคำนวณเหมือนแบบจำลองปกติดังแสดงในสมการที่ 3.3 ดังนั้นการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลขั้นที่ 2 นี้ แสดงดังสมการที่ 3.19

เวลาที่ส่งข้อมูล 1 บล็อกไปเก็บยังฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการจำนวน 3 ครั้ง

$$= \quad HDAccess * 3 \quad \text{วินาที} \quad (3.19)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2)

= (ma*3) + (HDAccess*3) วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.20

totalTime = totalTime + ((ma*3) + (HDAccess*3)) วินาที (3.20)

7. ส่วนตัดสินใจว่าสิ้นสุดการทำงาน จะเป็นส่วนที่ตรวจสอบว่าเอ็นติตียังต้องการเพจในการประมวลผลอีกหรือไม่ ถ้าใช่เอ็นติตีจะถูกส่งไปยังส่วนที่ 2 อีกครั้ง แต่ถ้าไม่ต้องการเพจแล้วก็จะถือว่าสิ้นสุดการทำงานและแสดงผลลัพธ์

3.3 แบบจำลองระยะไกลแบบจำกัด

แบบจำลองระยะไกลแบบจำกัด เป็นแบบจำลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายของคอมพิวเตอร์ส่วนบุคคลซึ่งประกอบด้วยเครื่องผู้ขอใช้บริการ 1 เครื่อง เครื่องผู้ให้บริการที่เก็บข้อมูลจำนวน S เครื่อง เครื่องผู้ให้บริการที่เก็บพาริตีจำนวน 1 เครื่อง และเครื่องคอมพิวเตอร์อื่นนอกจากการทดลอง ซึ่งเครือข่ายและเครื่องคอมพิวเตอร์มีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพถูกจำกัด โดยที่โมดูลในแบบจำลองระยะไกลแบบจำกัดสามารถแบ่งได้ 7 ส่วน คือ ส่วนเริ่มต้นการทำงาน, ส่วนสร้างหมายเลขเพจที่โปรแกรมใช้ในการประมวลผล และตรวจสอบที่อยู่ของเพจ, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ, ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อทำอแคปทีฟพาริตีแคชชิง และส่วนตัดสินใจว่าสิ้นสุดการทำงาน ซึ่งแต่ละส่วนภายในจะประกอบด้วยโมดูลต่าง ๆ ต่อกันตามลำดับการทำงาน รายละเอียดของโมดูลต่าง ๆ ในแต่ละส่วนประกอบสามารถดูได้ในภาคผนวก ค. ความสัมพันธ์ของทุกส่วนประกอบและการทำงานของแบบจำลองนี้เหมือนกับแบบจำลองระยะไกล ดังแสดงในรูปที่ 3.2 ซึ่งผลลัพธ์ที่ได้จากแบบจำลอง คือ เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จซึ่งเป็นเวลาที่เกิดจากการสะสมเวลาที่โปรแกรมใช้ในการประมวลของแต่ละครั้งที่โปรแกรมต้องการใช้เพจในการประมวลผล รายละเอียดของตัวแปรต่าง ๆ ในแบบจำลองสามารถดูได้จากตารางที่ ก.2 ในภาคผนวก ก. รายละเอียดส่วนรายละเอียดของส่วนประกอบหลัก 7 ส่วนในแบบจำลองระยะไกลแบบจำกัด มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนเริ่มต้นการทำงาน เป็นส่วนที่สร้างเอ็นติตีขึ้นมาเพื่อดำเนินการต่าง ๆ ในแบบจำลองและกำหนดค่าเริ่มต้นให้กับตัวแปรทั่วไปในแบบจำลอง

2. ส่วนสร้างหมายเลขเพจที่โปรแกรมใช้ในการประมวลผลและตรวจสอบที่อยู่ของเพจ เป็นส่วนที่ทำการสุ่มหมายเลขเพจที่โปรแกรมใช้ในการประมวลผลด้วยฟังก์ชันทางคณิตศาสตร์ จากนั้นก็จะคำนวณว่าเพจดังกล่าวอยู่ในบล็อกใด และตรวจสอบว่าเพจดังกล่าวอยู่ที่อยู่ที่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการ ฮาร์ดดิสก์ หรือหน่วยความจำหลักของเครื่องผู้ให้บริการ

3. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ ซึ่งมีการทำงานและคำนวณเหมือนกับแบบจำลองปกติ ดังแสดงในสมการที่ 3.1

4. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์ เป็นส่วนที่ทำการคำนวณเวลานำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลักและตรวจสอบว่าหน่วยความจำหลักมีที่ว่างเพียงพอสำหรับบล็อกดังกล่าวหรือไม่ ถ้าพอก็จะนำบล็อกนั้นเข้าสู่หน่วยความจำหลัก และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณเหมือนแบบจำลองปกติ ดังแสดงในสมการที่ 3.4 แต่ถ้าหน่วยความจำหลักมีพื้นที่ว่างไม่เพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล ก็จะทำการเลือกบล็อกที่จะถูกแทนที่และนำบล็อกนี้เข้าสู่ขั้นตอนของอแคปทีฟพาริตีแคชชิง (รายละเอียดสามารถดูได้ในบทที่ 2) และเมื่อหน่วยความจำหลักมีพื้นที่ว่างเพียงพอ จึงนำบล็อกดังกล่าวเข้าสู่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณเหมือนแบบจำลองระยะไกล ดังแสดงในสมการที่ 3.6

5. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ เป็นส่วนที่ทำการคำนวณเวลาที่นำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากเครื่องผู้ให้บริการกลับมายังหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ และตรวจสอบว่าหน่วยความจำหลักของเครื่องผู้ขอใช้บริการมีที่ว่างเพียงพอสำหรับบล็อกดังกล่าวหรือไม่ ซึ่งแบ่งเป็น 2 กรณี คือ หน่วยความจำหลักมีที่ว่างเพียงพอ และหน่วยความจำหลักมีที่ว่างไม่เพียงพอ รายละเอียดของ 2 กรณี มีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 หน่วยความจำหลักมีที่ว่างเพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล จึงนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลเข้าสู่หน่วยความจำหลัก และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่

(2) เวลาที่เครื่องผู้ขอใช้บริการส่งคำร้องขอเพจที่ต้องการไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลของเพจที่ต้องการ

(3) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลมายังเครื่องผู้ขอใช้บริการ

(4) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งคำร้องไปยังเครื่องผู้ให้บริการที่เก็บพารีตีเพจเพื่อแก้ไขสถานะภาพเป็นอินแวลิดเพจ

(5) เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

โดยเวลาที่ (2), (3) และ (4) คำนวณเหมือนแบบจำลองระยะไกล ดังแสดงในสมการที่ 3.7, 3.8 และ3.7 ตามลำดับ แต่เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพของเครือข่ายถูกจำกัดซึ่งจะต้องมีเวลาที่รอให้สายสัญญาณว่าง (ตัวแปร TimeDelay) อีกทั้งเครื่องผู้ให้บริการทำงานอื่นนอกจากการทดลอง ทำให้ต้องมีเวลาทำคอนเท็กซ์สวิตซ์ (ตัวแปรTimeCS)

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2) + (3) + (4) + (5)

= ma + (((512)/(BW*1048576)) + TimeDelay + TimeCS) +
((131584/(BW*1048576)) + TimeDelay + TimeCS) +
(((512)/(BW*1048576)) + TimeDelay + TimeCS) + ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.21

totalTime = totalTime + ((ma*2) +
((((512)/(BW*1048576)) + TimeDelay + TimeCS) *2) +
(131584/(BW*1048576))+TimeDelay + TimeCS) วินาที (3.21)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณเวลาที่รอให้สายสัญญาณว่าง มีรายละเอียดแสดงดังสมการที่ 3.22

ประสิทธิภาพในการใช้ช่องสื่อสาร [5] (Channel Efficiency, ตัวแปร CE)

$$\text{ประสิทธิภาพในการใช้ช่องสื่อสาร} = \frac{P}{P + (2\pi/A)}$$

เมื่อ P คือ เวลาที่ใช้ในการรับส่งข้อความ

$$P = \frac{\text{ขนาดข้อความ}}{\text{ความเร็วในการรับส่ง}} \quad \text{วินาที}$$

A คือ ความน่าจะเป็นที่จะได้ส่งสัญญาณ

กำหนดให้ TimeDelay คือ เวลาที่รอให้สายสัญญาณว่าง $= 2\pi/A$

$$\text{ดังนั้น TimeDelay} = \frac{(1 - CE)*P}{CE} \quad \text{วินาที} \quad (3.22)$$

ตัวอย่างที่ 3.1

ความเร็วปกติของการรับส่ง 100 เมกะบิตต่อวินาที ขนาดข้อมูล 200 เมกะบิต ประสิทธิภาพเครือข่าย 80 % ดังนั้น

$$\text{TimeDelay} = \frac{(1 - 0.8) * (200/100)}{0.8} \quad \text{วินาที}$$

$$= \frac{0.2 * 1}{0.8} \quad \text{วินาที}$$

$$= 0.25 \quad \text{วินาที}$$

5.2 หน่วยความจำหลักมีที่ว่างไม่เพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล จึงทำการเลือกบล็อกที่จะถูกแทนที่และนำบล็อกนี้เข้าสู่ขั้นตอนของอแดปทีฟพาริตีแคชชิง (รายละเอียดสามารถดูได้ในบทที่ 2) เมื่อหน่วยความจำหลักมีพื้นที่ว่างเพียงพอ จึงนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลเข้าสู่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่

(2) เวลาที่เครื่องผู้ขอใช้บริการส่งคำร้องขอเพจที่ต้องการไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลของเพจที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลมายังเครื่องผู้ขอใช้บริการ

(4) เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งคำร้องไปยังเครื่องผู้ให้บริการที่เก็บพาริตีเพจเพื่อแก้ไขสถานะภาพเป็นอินแวลิดเพจ

(5) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อทำการเลือกเพจที่จะถูกแทนที่

(6) เวลาเฉลี่ยที่ใช้วางบล็อกที่จะถูกแทนที่ตามหลักอแดปทีฟพาริตีแคชชิง ซึ่งจะกล่าวรายละเอียดต่อไปในข้อ 6 จึงไม่รวมเวลาดังกล่าวในการคำนวณส่วนนี้

(7) เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ

โดยเวลาที่ (2), (3) และ (4) คำนวณเหมือนแบบจำลองระยะไกล ดังแสดงในสมการที่ 3.7, 3.8 และ3.7 ตามลำดับ แต่เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพของเครือข่ายถูกจำกัดซึ่งจะต้องมีเวลาที่รอให้สายสัญญาณว่าง อีกทั้งเครื่องผู้ให้บริการทำงานอื่นนอกจากการทดลองทำให้ต้องมีเวลาทำคอนเท็กซ์สวิตซ์

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2) + (3) + (4) + (5) + (7)

= ma + (512/(BW*1048576) + TimeDelay + TimeCS) +
(131584/(BW*1048576) + TimeDelay + TimeCS) +
(512/(BW*1048576) + TimeDelay + TimeCS) + ma + ma วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.23

totalTime = totalTime + ((ma*3) +
((512/(BW*1048576) + TimeDelay + TimeCS)*2) +
(131584/(BW*1048576) + TimeDelay + TimeCS)) วินาที(3.23)

6. ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผล เมื่อทำอแดปทีฟพาริตีแคชชิง เป็นส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผล เมื่อจัดวางบล็อกที่จะถูกแทนที่โดยใช้หลักการของอแดปทีฟพาริตีแคชชิง (รายละเอียดสามารถดูได้ในบทที่ 2) ซึ่งจะทำให้เกิดกรณีต่าง ๆ แบ่งเป็น 3 กรณี คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1 การนำเพจที่จะถูกแทนที่ไปใส่ในอแดปทีฟพาริตีแคชชิง ซึ่งมี 3 สตรีมแคช และ 1 วิกทิมแคช ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แยกเป็น 4 กรณี เหมือนแบบจำลองระยะไกล ดังแสดงในสมการที่ 3.11 ถึง 3.14

6.2 เมื่อจัดวางบล็อกที่จะถูกแทนที่ลงในสตรีมแคชหรือวิกทิกแคช แล้วพบว่าแคชนั้นเต็มและเครื่องผู้ให้บริการมีพื้นที่ว่างเพียงพอจึงนำบล็อกทั้งหมดในแคชนั้นส่งไปยังเครื่องผู้ให้บริการต่าง ๆ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณจาก

(1) เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อทำการอ่านข้อมูลจำนวน 4 ครั้ง

(2) เวลาที่ใช้ในการส่งข้อมูล 1 บล็อก ไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลเครื่องที่ 1, 2, 3 และเครื่องผู้ให้บริการที่เก็บพาริตีเพจ รวมทั้งหมด 4 ครั้ง ซึ่งการคำนวณเวลาที่ใช้ในการรับส่งข้อมูลขั้นนี้เหมือนกับแบบจำลองระยะไกล ดังแสดงในสมการที่ 3.15 แต่เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพของเครือข่ายถูกจำกัดซึ่งจะต้องมีเวลาที่รอให้สายสัญญาณว่าง อีกทั้งเครื่องผู้ให้บริการทำงานอื่นนอกจากการทดลองทำให้ต้องมีเวลาทำคอนเท็กซ์สวิตซ์

ดังนั้น เวลาที่โปรแกรมใช้ในการประมวลผล = (1) + (2)

= (ma*4) + (((131584/(BW*1048576)) + TimeDelay + TimeCS) *4) วินาที

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ แสดงดังสมการที่ 3.24

totalTime = totalTime + ((ma*4) +
(((131584/(BW*1048576))
+ TimeDelay + TimeCS) *4)) วินาที (3.24)

6.3 เมื่อจัดวางบล็อกที่จะถูกแทนที่ลงในสตรีมแคชหรือวิกทิมแคช แล้วพบว่าแคชนั้นเต็มและเครื่องผู้ให้บริการไม่มีพื้นที่ว่างเพียงพอจึงนำบล็อกที่เป็นข้อมูลไม่รวมพาริตีบล็อกในแคชนั้นกลับลงฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการ และคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ซึ่งคำนวณเหมือนแบบจำลองระยะไกล ดังแสดงในสมการที่ 3.20

7. ส่วนตัดสินใจว่าสิ้นสุดการทำงาน จะเป็นส่วนที่ตรวจสอบว่าเอ็นติตียังต้องการเพจในการประมวลผลอีกหรือไม่ ถ้าใช่เอ็นติตีจะถูกส่งไปยังส่วนที่ 2 อีกครั้ง แต่ถ้าไม่ต้องการเพจแล้วก็จะถือว่าสิ้นสุดการทำงานและแสดงผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและการวิเคราะห์ผล

การทดลองหน่วยความจำเสมือนแบบสลับหน้าระยะไกลแบ่งเป็น 2 กรณีหลัก ๆ คือ กรณีที่เครือข่ายและทุกเครื่องในเครือข่ายไม่มีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายมีความเร็วในการรับส่งข้อมูลด้วยความเร็วสูงสุด คือ 100 เมกะบิตต่อวินาที และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ อีกกรณีคือเครือข่ายมีเครื่องคอมพิวเตอร์นอกจากการทดลองและเครื่องผู้ให้บริการมีการทำงานอื่นด้วยทำให้การรับส่งข้อมูลผ่านเครือข่ายและประสิทธิภาพของเครือข่ายถูกจำกัด ซึ่งในกรณีแรกนั้นจะทำการทดลองเพื่อวัดประสิทธิภาพของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลว่ามีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ โดยสนใจสภาพแวดล้อมที่เป็นปัจจัยส่งผลกระทบกับประสิทธิภาพของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลนั้น คือ ความเร็วในการรับส่งข้อมูลผ่านเครือข่ายและฮาร์ดดิสก์ ประสิทธิภาพของเครือข่าย ขนาดของหน่วยความจำหลัก และจำนวนเครื่องผู้ให้บริการตั้งแต่ 3-9 เครื่อง (เนื่องจากแบบจำลองทั้ง 3 สร้างด้วยโปรแกรมอารีน่า เวอร์ชัน 5 สำหรับนักศึกษาซึ่งจำกัดด้านจำนวน ตัวแปร และจำนวนโมดูลทำให้จำนวนเครื่องคอมพิวเตอร์ที่กำหนดเป็นเครื่องผู้ให้บริการกับเครื่องผู้ขอใช้บริการรวมกันไม่เกิน 10 เครื่อง) เป็นการทดลองที่ 1 ส่วนกรณีที่เครือข่ายมีเครื่องคอมพิวเตอร์นอกจากการทดลองและเครื่องผู้ให้บริการมีการทำงานอื่นด้วยทำให้การรับส่งข้อมูลผ่านเครือข่ายและประสิทธิภาพของเครือข่ายถูกจำกัดนั้นจะเป็นการทดลองที่ 2 ซึ่งจะทำการหาจำนวนเครื่องผู้ให้บริการที่น้อยที่สุดที่ทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ โดยวิธีการเปรียบเทียบประสิทธิภาพของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลกับหน่วยความจำเสมือนแบบสลับหน้าปกติจะดูจากเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จที่ได้จากการทดลองแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การทดลองที่ 1

จุดประสงค์

เพื่อวัดประสิทธิภาพของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลว่ามีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ โดยที่เครือข่ายและทุกเครื่องไม่มีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายรับส่งข้อมูลด้วยความเร็วสูงสุด และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที หน่วยความจำหลักของแต่ละเครื่องมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม (ไม่รวมพื้นที่ที่ใช้เป็นอแดปทีฟพาริตีแคช) และเครื่องผู้ให้บริการมีจำนวน 3-9 เครื่อง

สมมติฐานการทดลอง

สมมติฐานหลัก : หน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพน้อยกว่าหรือเท่ากับหน่วยความจำเสมือนแบบสลับหน้าปกติ โดยที่เครือข่ายและทุกเครื่องไม่มีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายรับส่งข้อมูลด้วยความเร็วสูงสุด และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที หน่วยความจำหลักของแต่ละเครื่องมีขนาด i เปอร์เซ็นต์ของขนาดโปรแกรม และเครื่องผู้ให้บริการมีจำนวน j เครื่อง

สมมติฐานรอง : หน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ โดยที่เครือข่ายและทุกเครื่องไม่มีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายรับส่งข้อมูลด้วยความเร็วสูงสุด และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที หน่วยความจำหลักของแต่ละเครื่องมีขนาด i เปอร์เซ็นต์ของขนาดโปรแกรม และเครื่องผู้ให้บริการมีจำนวน j เครื่อง

เมื่อ i = 10, 20 และ j = 3, 4, ..., 9

สัญลักษณ์ที่ใช้

H_0 : $\mu_{Pi} - \mu_{Aij} \leq 0$

H_1 : $\mu_{Pi} - \mu_{Aij} > 0$

เมื่อ μ_{Pi} คือ เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จจากประชากรของหน่วยความจำเสมือนแบบสลับหน้าปกติ หน่วยความจำหลักของแต่ละเครื่องมีขนาด i เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

μ_{Aij} คือ เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จจากประชากรของหน่วยความจำเสมือนแบบสลับหน้าระยะไกล โดยที่เครือข่ายและทุกเครื่องไม่มีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายรับส่งข้อมูลด้วยความเร็วสูงสุด และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที หน่วยความจำหลักของแต่ละเครื่องมีขนาด i เปอร์เซ็นต์ของขนาดโปรแกรม และเครื่องผู้ให้บริการมีจำนวน j เครื่อง

เมื่อ i = 10, 20 และ j = 3, 4, ..., 9

สถิติที่ใช้ในการทดลอง

$$Z = \frac{(\overline{X}_P - \overline{X}_A) - (\mu_P - \mu_A)}{\sqrt{\frac{S_P^2}{n_P} + \frac{S_A^2}{n_A}}}$$

เมื่อ $\overline{X}_P$ คือ ค่าเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จจากแบบจำลองปกติ

$\overline{X}_A$ คือ ค่าเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จจากแบบจำลองระยะไกล

S_P^2 คือ ความแปรปรวนของแบบจำลองปกติ

S_A^2 คือ ความแปรปรวนของแบบจำลองระยะไกล

n_P คือ จำนวนตัวอย่างของแบบจำลองปกติ

n_A คือ จำนวนตัวอย่างของแบบจำลองระยะไกล

ที่ระดับนัยสำคัญ 0.05 อาณาเขตวิกฤต คือ $Z > 1.64$

ตัวแปรควบคุม

- ความเร็วในการรับส่งผ่านเครือข่าย 100 เมกะบิตต่อวินาที
- ประสิทธิภาพเครือข่าย 100 เปอร์เซ็นต์
- ทุกเครื่องในเครือข่ายไม่มีการทำงานอื่นใดนอกจากการทดลอง
- ฮาร์ดดิสก์มีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที

ตัวแปรที่พิจารณา

- จำนวนเครื่องผู้ให้บริการ 3 – 9 เครื่อง
- หน่วยความจำหลักของแต่ละเครื่องมีขนาด 10 และ 20 เปอร์เซ็นต์ ของขนาดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรวัด

เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ

วิธีการทดลอง

ทำการทดลองแบบจำลองปกติ และแบบจำลองระยะไกล โดยกำหนดค่าตัวแปรต่าง ๆ ให้กับแบบจำลองเพื่อให้มีสภาพแวดล้อมตรงตามต้องการ ซึ่งแต่ละสภาพแวดล้อมทำการทดลองจำนวน 100 ครั้ง จากนั้นคำนวณเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ และค่าสถิติต่าง ๆ

ผลการทดลอง

ตารางที่ 4.1 เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ และส่วนเบี่ยงเบนมาตรฐานของแบบจำลองปกติ

ขนาดหน่วยความจำหลัก (เปอร์เซ็นต์)	เวลาเฉลี่ยที่ใช้ในการประมวลผลจนเสร็จ (วินาที)	ส่วนเบี่ยงเบนมาตรฐาน
10	27.4670	0.2960
20	24.3868	0.4039

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ และส่วนเบี่ยงเบนมาตรฐานของแบบจำลองระยะไกล เมื่อเครือข่ายมีจำนวนเครื่องผู้ให้บริการ 3-9 เครื่องและหน่วยความจำหลักมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม

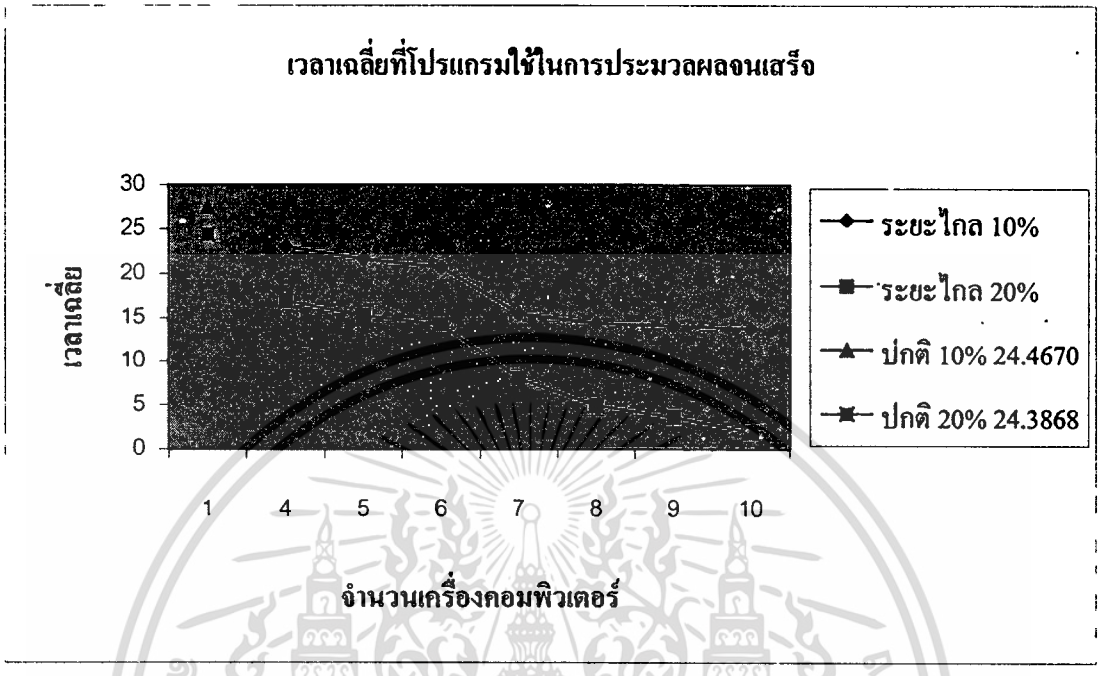
จำนวนเครื่องผู้ให้บริการ	ขนาดหน่วยความจำหลัก 10 เปอร์เซ็นต์		ขนาดหน่วยความจำหลัก 20 เปอร์เซ็นต์	
	เวลาเฉลี่ย (วินาที)	ส่วนเบี่ยงเบนมาตรฐาน	เวลาเฉลี่ย (วินาที)	ส่วนเบี่ยงเบนมาตรฐาน
3	23.0352	0.4175	16.9310	0.5018
4	21.7026	0.4580	15.3055	0.4652
5	20.7531	0.4469	14.2108	0.4931
6	15.5058	0.5455	7.9976	0.4661
7	14.4084	0.6829	4.9407	0.4500
8	14.1041	0.6352	3.6404	0.4664
9	13.7662	0.6829	1.9367	0.3049

ตารางที่ 4.3 ค่าสถิติ Z เมื่อเครือข่ายมีจำนวนเครื่องผู้ให้บริการ 3-9 เครื่อง และหน่วยความจำหลักมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม

จำนวนเครื่องผู้ให้บริการ	ค่า Z เมื่อขนาดหน่วยความจำหลัก 10 เปอร์เซ็นต์	ค่า Z เมื่อขนาดหน่วยความจำหลัก 20 เปอร์เซ็นต์
3	103.6554	115.7365
4	123.2340	147.5124
5	146.9297	159.7034
6	138.2267	265.8679
7	235.7594	321.8741
8	207.9845	336.1091
9	198.8340	540.8485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จจากตารางที่ 4.1 และ 4.2 แสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ การทดลองที่ 1

วิเคราะห์ผลการทดลอง

เมื่อพิจารณาค่าสถิติ Z ที่คำนวณได้ดังแสดงในตารางที่ 4.3 ในทุกกรณีพบว่ามากกว่า 1.64 หรืออยู่ในอาณาเขตวิกฤตจึงปฏิเสธสมมติฐานหลักยอมรับสมมติฐานรอง คือ หน่วยความจำเสมือนแบบสลับหน้าระยะไกลว่ามีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ ที่ระดับนัยสำคัญ 0.05 เมื่อมีสภาพแวดล้อมดังนี้ เครือข่ายและทุกเครื่องไม่มีการทำงานอื่นนอกจากการทดลองทำให้เครือข่ายรับส่งข้อมูลด้วยความเร็วสูงสุด และมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์ ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที เครื่องผู้ให้บริการมีจำนวน 3-9 เครื่อง และหน่วยความจำหลักมีขนาด 10 กับ 20 เปอร์เซ็นต์ของขนาดโปรแกรม

เมื่อพิจารณาเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลของหน่วยความจำเสมือนแบบสลับหน้าระยะไกล เมื่อขนาดหน่วยความจำหลักมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม เครือข่ายและทุกเครื่องไม่มีการทำงานอื่นใด ดังแสดงในตารางที่ 4.2 และรูปที่ 4.1 พบว่าเมื่อหน่วยความจำหลักมีขนาดใหญ่ขึ้นจะทำให้เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จมีแนวโน้มลดลง ดังนั้นเมื่อเพิ่มขนาดหน่วยความจำหลักให้มีขนาดใหญ่ขึ้น โดยที่เครือข่ายและทุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องในเครือข่ายไม่การทำงานอื่นนั้นจะทำให้การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากขึ้นด้วย

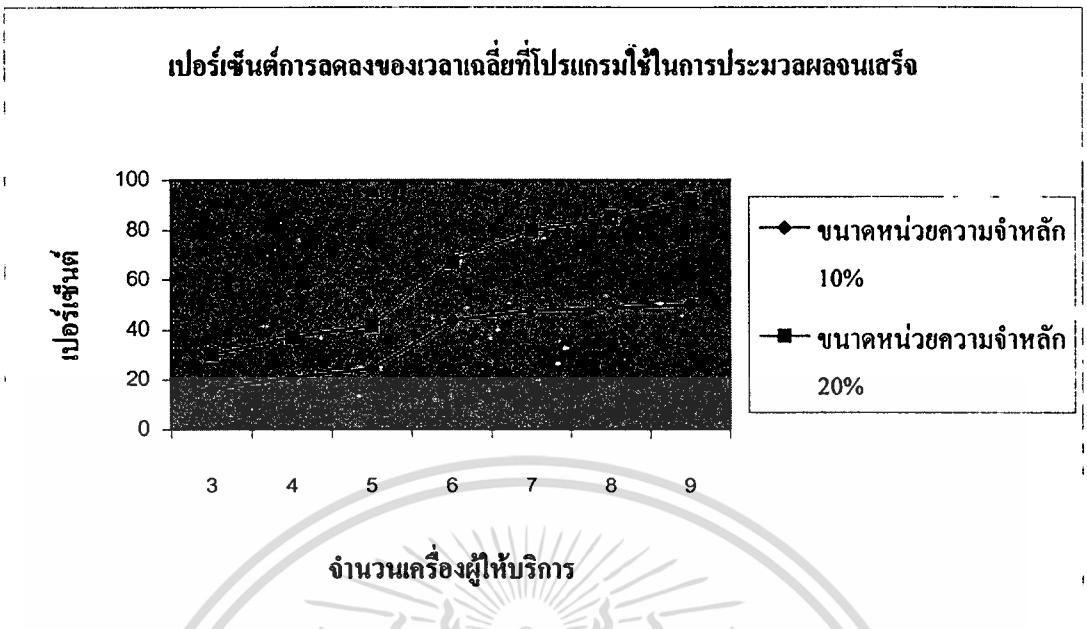
เพื่อแสดงให้เห็นถึงประสิทธิภาพของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลในกรณีที่เพิ่มจำนวนเครื่องผู้ให้บริการจาก 3-9 เครื่องได้ชัดเจนขึ้น จึงได้คำนวณเปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จโดยใช้ข้อมูลจากแบบจำลองปกติดังแสดงในตารางที่ 4.1 และแบบจำลองระยะไกลดังแสดงในตารางที่ 4.2 ซึ่งผลลัพธ์แสดงดังตารางที่ 4.4

ตารางที่ 4.4 เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ เมื่อเครือข่ายมีจำนวนเครื่องผู้ให้บริการ 3-9 เครื่อง และหน่วยความจำหลักมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม

จำนวนเครื่องผู้ให้บริการ	เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ	
	ขนาดหน่วยความจำหลัก 10 เปอร์เซ็นต์	ขนาดหน่วยความจำหลัก 20 เปอร์เซ็นต์
3	16.14	30.57
4	20.99	37.24
5	24.44	41.73
6	43.55	67.21
7	47.54	79.74
8	48.65	85.07
9	49.88	92.06

เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จจากตารางที่ 4.4 แสดงได้ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ การทดลองที่ 1

เมื่อพิจารณาข้อมูลในตารางที่ 4.4 และรูปที่ 4.2 จะเห็นได้อย่างชัดเจนว่าเมื่อเพิ่มจำนวนเครื่องผู้ให้บริการจะทำให้เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จของการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีแนวโน้มเพิ่มขึ้นเรื่อย ๆ โดยเฉพาะเมื่อเครื่องผู้ให้บริการมีจำนวนเพิ่มขึ้นจาก 5 เครื่อง เป็น 6 เครื่อง ทำให้เปอร์เซ็นต์การลดลงของเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จเพิ่มขึ้นอย่างชัดเจน ดังนั้นการเพิ่มจำนวนเครื่องผู้ให้บริการตั้งแต่ 3-9 เครื่อง ทำให้การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากขึ้นด้วย

เมื่อพิจารณาเวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในฮาร์ดดิสก์เทียบกับเวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลักที่เครื่องผู้ให้บริการเมื่อเครือข่ายและทุกเครื่องไม่มีการทำงานอื่นใดนอกจากการทดลอง ดังแสดงในสมการที่ 4.1 และ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในฮาร์ดดิสก์ เมื่อมีความเร็วในการรับส่งข้อมูล X เมกะไบต์ต่อวินาที

$$= \text{เวลาแสวงหา} + \text{เวลาแฝง} + \text{เวลาในการรับส่งข้อมูล}$$

$$= 0.008 + 0.0042 + \frac{131072}{X * 1024 * 1024 * 8}$$

$$= 0.0122 + \frac{0.015625}{X} \quad \text{วินาที} \qquad (4.1)$$

เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลักที่เครื่องผู้ให้บริการ

$$= \frac{\text{ขนาดข้อมูล}}{\text{ความเร็วในการรับส่งผ่านเครือข่าย}}$$

$$= \frac{\text{ขนาดบล็อก} + \text{ขนาดข้อมูลส่วนหัว}}{\text{ความเร็วในการรับส่งผ่านเครือข่าย}}$$

$$= \frac{131584}{100 * 1024 * 1024}$$

$$= 0.0012549 \quad \text{วินาที} \qquad (4.2)$$

พบว่าเวลาแสวงหาและเวลาแฝงรวมกันเท่ากับ 0.0122 ซึ่งมากกว่า 0.0012549 ซึ่งเป็นเวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความความจำหลักที่เครื่องผู้ให้บริการบนเครือข่ายที่ไม่มีการทำงานอื่นใดนอกจากการทดลองเสมอ ดังนั้นไม่ว่าฮาร์ดดิสก์จะมีความเร็วในการรับส่งข้อมูลเป็นเท่าใดการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายที่ไม่มีการทำงานอื่นใดนอกจากการทดลองยังคงมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าแบบปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองที่ 2

จุดประสงค์

เพื่อหาจำนวนเครื่องผู้ให้บริการน้อยที่สุดที่ทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ โดยที่เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพถูกจำกัด เครื่องผู้ให้บริการทุกเครื่องทำงานอื่นนอกจากการทดลอง ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที และหน่วยความจำหลักของทุกเครื่องมีขนาด 10 และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม (ไม่รวมพื้นที่ที่ใช้เป็นอแคปทีฟพาริตีแคช)

ตัวแปรควบคุม

- ความเร็วในการรับส่งผ่านเครือข่าย 10 เมกะบิตต่อวินาที เพิ่มขึ้นทีละ 10 เมกะบิตต่อวินาทีจนถึง 100 เมกะบิตต่อวินาที
- ประสิทธิภาพเครือข่าย 10 เปอร์เซ็นต์ เพิ่มขึ้นทีละ 10 เปอร์เซ็นต์ จนถึง 100 เปอร์เซ็นต์
- เครื่องผู้ให้บริการทุกเครื่องทำงานอื่นนอกจากการทดลอง
- ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที
- หน่วยความจำหลักของแต่ละเครื่องมีขนาด 10 และ 20 เปอร์เซ็นต์ ของขนาดโปรแกรม

ตัวแปรที่พิจารณา

- จำนวนเครื่องผู้ให้บริการ 3 – 9 เครื่อง

มาตรวัด

เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ

วิธีการทดลอง

ทำการค้นหาจำนวนเครื่องผู้ให้บริการน้อยที่สุดซึ่งทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติด้วยการค้นหาแบบฮิวริสติก (heuristic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

ตารางที่ 4.5 จำนวนเครื่องผู้ให้บริการน้อยที่สุดเมื่อมีสภาพแวดล้อมถูกจำกัด และหน่วยความจำหลักมีขนาด 10 เปอร์เซ็นต์ของขนาดโปรแกรม

ความเร็วในการรับส่งผ่านเครือข่าย	ประสิทธิภาพเครือข่าย									
	10	20	30	40	50	60	70	80	90	100
10	-	-	-	-	-	-	6	6	4	3
20	-	-	-	6	6	4	3	3	3	3
30	-	-	6	4	3	3	3	3	3	3
40	-	-	6	3	3	3	3	3	3	3
50	-	6	3	3	3	3	3	3	3	3
60	-	6	3	3	3	3	3	3	3	3
70	-	4	3	3	3	3	3	3	3	3
80	-	3	3	3	3	3	3	3	3	3
90	6	3	3	3	3	3	3	3	3	3
100	6	3	3	3	3	3	3	3	3	3

ตารางที่ 4.6 จำนวนเครื่องผู้ให้บริการน้อยที่สุดเมื่อมีสภาพแวดล้อมถูกจำกัด และหน่วยความจำหลักมีขนาด 20 เปอร์เซ็นต์ ของขนาดโปรแกรม

ความเร็วในการรับส่งผ่านเครือข่าย	ประสิทธิภาพเครือข่าย									
	10	20	30	40	50	60	70	80	90	100
10	-	-	-	-	-	-	7	6	6	
20	-	-	-	-	7	6	3	3	3	3
30	-	-	-	6	6	4	3	3	3	3
40	-	-	6	3	3	3	3	3	3	3
50	-	-	6	3	3	3	3	3	3	3
60	-	6	3	3	3	3	3	3	3	3
70	-	6	3	3	3	3	3	3	3	3
80	-	6	3	3	3	3	3	3	3	3
90	-	3	3	3	3	3	3	3	3	3
100	-	3	3	3	3	3	3	3	3	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ผลการทดลอง

ผลการทดลองในตารางที่ 4.5 ได้มาจากการนำจำนวนเครื่องผู้ให้บริการตั้งแต่ 3 เครื่อง ถึง 9 เครื่อง มาทำการค้นหาแบบฮิวริสติก เพื่อให้ทราบถึงจำนวนเครื่องผู้ให้บริการน้อยที่สุดซึ่งทำให้การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติ ซึ่งสภาพแวดล้อมของการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีดังนี้

- เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพถูกจำกัด
- เครื่องผู้ให้บริการทุกเครื่องทำงานอื่นนอกจากการทดลอง
- ฮาร์ดดิสก์เครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที
- หน่วยความจำหลักของทุกเครื่องมีขนาด 10 เปอร์เซ็นต์ของขนาดโปรแกรม

พิจารณาผลการทดลองดังแสดงในตารางที่ 4.5 พบว่าการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติในขอบเขตสภาพแวดล้อมถูกจำกัดบางประการ เช่น กรณีที่ความเร็วในการรับส่งผ่านเครือข่าย 80 เมกะบิตต่อวินาที เครือข่ายมีประสิทธิภาพ 10 เปอร์เซ็นต์ การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพน้อยกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติถึงแม้ว่าจะใช้เครื่องผู้ให้บริการถึง 9 เครื่องก็ตาม แต่เมื่อเพิ่มความเร็วในการรับส่งผ่านเครือข่ายเป็น 90 เมกะบิตต่อวินาที การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ แต่ต้องใช้จำนวนเครื่องผู้ให้บริการไม่น้อยกว่า 6 เครื่อง ซึ่งเมื่อเพิ่มความเร็วในการรับส่งข้อมูลผ่านเครือข่ายควบคู่ไปกับการเพิ่มประสิทธิภาพของเครือข่ายจะทำให้การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ โดยที่สามารถใช้เครื่องผู้ให้บริการน้อยที่สุดคือ 3 เครื่อง

ผลการทดลองในตารางที่ 4.6 ได้มาจากการนำจำนวนเครื่องผู้ให้บริการตั้งแต่ 3 เครื่อง ถึง 9 เครื่องมาทำการค้นหาแบบฮิวริสติก เพื่อให้ทราบถึงจำนวนเครื่องผู้ให้บริการน้อยที่สุดที่ทำให้การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติ ซึ่งสภาพแวดล้อมของการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีดังนี้

- เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพถูกจำกัด
- เครื่องผู้ให้บริการทุกเครื่องทำงานอื่นนอกจากการทดลอง
- ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการมีความเร็วในการรับส่งข้อมูล 5 เมกะไบต์ต่อวินาที
- หน่วยความจำหลักของทุกเครื่องมีขนาด 20 เปอร์เซ็นต์ของขนาดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาผลการทดลองดังแสดงในตารางที่ 4.6 พบว่า การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติในขอบเขตสภาพแวดล้อมถูกจำกัดบางประการ เช่น กรณีที่ความเร็วในการรับส่งผ่านเครือข่าย 40 เมกะบิตต่อวินาที และประสิทธิภาพของเครือข่าย 30 เปอร์เซ็นต์ การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลสามารถทำงานได้มีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ แต่ต้องใช้เครื่องผู้ให้บริการไม่น้อยกว่า 6 เครื่อง

เมื่อเปรียบเทียบข้อมูลในตารางที่ 4.5 และ 4.6 จะพบว่าการเพิ่มขนาดหน่วยความจำหลักในขณะที่เครือข่ายกับเครื่องผู้ให้บริการทำงานอื่น และเครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพถูกจำกัดบางประการทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพลดลง เช่น กรณีที่เครือข่ายมีความเร็วในการรับส่งข้อมูล 90 เมกะบิตต่อวินาทีขึ้นไป และประสิทธิภาพของเครือข่าย 10 เปอร์เซ็นต์ และขนาดหน่วยความจำหลักของทุกเครื่องมีขนาด 10 เปอร์เซ็นต์ของขนาดโปรแกรม การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ แต่เมื่อเราเพิ่มหน่วยความจำหลักของทุกเครื่องเป็น 20 เปอร์เซ็นต์ของขนาดโปรแกรม การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพน้อยกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ หรือในกรณีที่เครือข่ายมีความเร็วในการรับส่งข้อมูล 70 เมกะบิตต่อวินาที และประสิทธิภาพของเครือข่าย 20 เปอร์เซ็นต์ และขนาดหน่วยความจำหลักของทุกเครื่องมีขนาด 10 เปอร์เซ็นต์ของขนาดโปรแกรม ต้องใช้เครื่องผู้ให้บริการไม่น้อยกว่า 4 เครื่อง แต่เมื่อเราเพิ่มหน่วยความจำหลักของทุกเครื่องเป็น 20 เปอร์เซ็นต์ของขนาดโปรแกรม ต้องใช้เครื่องผู้ให้บริการไม่น้อยกว่า 6 เครื่อง ซึ่งเนื่องมาจากการที่เราเพิ่มหน่วยความจำหลักของเครื่องผู้ให้บริการทำให้เพจถูกนำไปเก็บอยู่ที่เครื่องผู้ให้บริการมากขึ้นส่งผลให้ต้องมีการรับส่งผ่านเครือข่ายมากขึ้น โดยในขณะที่เครือข่ายและเครื่องผู้ให้บริการมีการทำงานอื่นด้วยนั้นทำให้ต้องเสียเวลาในการรับส่งข้อมูลมากขึ้น ดังนั้นการเพิ่มขนาดหน่วยความจำหลักของเครื่องผู้ให้บริการให้มีขนาดใหญ่ขึ้นในกรณีที่เครือข่ายและเครื่องผู้ให้บริการมีการทำงานอื่นจะไม่ทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลจะมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

การทำหน่วยความจำเสมือนแบบสลับหน้าเป็นวิธีหนึ่งที่สามารถแก้ปัญหาหน่วยความจำหลักมีขนาดไม่เพียงพอกับความต้องการของโปรแกรม โดยการเก็บสำรองข้อมูลที่ยังไม่ใช้ในการประมวลผลไว้ที่ฮาร์ดดิสก์ ซึ่งเมื่อเทียบเวลาในการรับส่งข้อมูลระหว่างหน่วยความจำหลักและฮาร์ดดิสก์พบว่าฮาร์ดดิสก์ใช้เวลาในการรับส่งข้อมูลช้ากว่ามาก จึงได้มีแนวคิดที่นำเอาหน่วยความจำหลักของเครื่องคอมพิวเตอร์อื่นภายในเครือข่ายเดียวกันมาใช้สำรองข้อมูลแทนฮาร์ดดิสก์ ซึ่งวิธีนี้ต้องมีการรับส่งข้อมูลผ่านเครือข่ายจึงได้มีการพัฒนาวิธีเพิ่มความน่าเชื่อถือให้กับการทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลควบคู่กันไปด้วย โดยการทดลองที่ผ่านมาทำการทดลองจริงบนเครือข่ายเวิร์คสเตชันที่เครือข่ายและคอมพิวเตอร์ไม่มีการทำงานอื่น ผลการทดลองพบว่าวิธีดังกล่าวมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ ผู้วิจัยจึงมีสมมติฐานว่าวิธีดังกล่าวสามารถนำมาประยุกต์กับเครือข่ายคอมพิวเตอร์ส่วนบุคคลได้

งานวิจัยนี้มีจุดประสงค์เพื่อจำลองการทำงานหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคลโดยการจำลองแบบด้วยคอมพิวเตอร์ เพื่อทดสอบว่าการทำงานหน่วยความจำเสมือนแบบสลับหน้าระยะไกลบนเครือข่ายคอมพิวเตอร์ส่วนบุคคลที่เครือข่ายและเครื่องผู้ให้บริการไม่มีการทำงานอื่นใดมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ และทดลองหากรณีที่วิธีการนี้ยังคงทำงานได้อย่างมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติเมื่อความเร็วในการรับส่งข้อมูลผ่านเครือข่ายและประสิทธิภาพของเครือข่ายถูกจำกัดอีกทั้งเครื่องผู้ให้บริการมีการทำงานอื่น ดังนั้นจึงทำการสร้างแบบจำลอง 3 แบบจำลอง คือ แบบจำลองปกติ แบบจำลองระยะไกล แบบจำลองระยะไกลแบบจำกัด โดยทุกแบบจำลองมีผลลัพธ์ที่ได้ (Output) คือ เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ โดยจะทำการเปรียบเทียบประสิทธิภาพจากเวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ผลการทดลองพบว่า

1. หน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากกว่าการทำหน่วยความจำเสมือนแบบสลับหน้าปกติ ที่ระดับนัยสำคัญ 0.05 ภายใต้สภาพแวดล้อมดังนี้
   - เครือข่ายมีการรับส่งข้อมูลด้วยความเร็วสูงสุด
   - เครือข่ายมีประสิทธิภาพเต็มร้อยเปอร์เซ็นต์
   - ทุกเครื่องไม่มีการทำงานอื่นใดนอกจากการทดลอง
   - เครื่องผู้ให้บริการมีจำนวน 3 – 9 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำหลักของทุกเครื่อง (ไม่รวมพื้นที่ที่ใช้เป็นอแดปทีฟพาริตีแคช) มีขนาด 10 เปอร์เซ็นต์ และ 20 เปอร์เซ็นต์ของขนาดโปรแกรม

2. จากการทดสอบแบบจำลองที่สร้างขึ้นในขอบเขตสภาพแวดล้อมที่ถูกจำกัดด้วยเงื่อนไขต่าง ๆ เช่น ความเร็วในการรับส่งผ่านเครือข่าย ประสิทธิภาพของเครือข่าย จำนวนเครื่องผู้ให้บริการ พบว่าส่วนใหญ่การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังคงมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ

3. เมื่อเพิ่มจำนวนเครื่องผู้ให้บริการตั้งแต่ 3-9 เครื่อง จะทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากขึ้น ทั้งในกรณีที่เครือข่ายและเครื่องผู้ให้บริการไม่มีการทำงานอื่นและมีการทำงานอื่นด้วย

4. การเพิ่มขนาดหน่วยความจำหลักในกรณีที่เครือข่ายและเครื่องผู้ให้บริการไม่มีการทำงานอื่นจะทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพเพิ่มขึ้น แต่ในกรณีที่เครือข่ายและเครื่องผู้ให้บริการมีการทำงานอื่นด้วยนั้นจะไม่ทำให้หน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพเพิ่มขึ้น

5. เมื่อเพิ่มความเร็วในการรับส่งข้อมูลผ่านเครือข่ายควบคู่ไปกับการเพิ่มประสิทธิภาพของเครือข่ายจะทำให้การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ แม้จะใช้เครื่องผู้ให้บริการน้อยที่สุดคือ 3 เครื่อง

5.2 ข้อเสนอแนะ

ข้อเสนอแนะของงานวิจัยนี้แบ่งออกเป็น 2 ส่วน คือ เกี่ยวกับแบบจำลอง และเกี่ยวกับการนำไปใช้งานจริง ซึ่งมีรายละเอียดดังต่อไปนี้

5.2.1 เกี่ยวกับแบบจำลอง

1. แบบจำลองทั้ง 3 แบบจำลองไม่รวมเวลาการเข้าใช้หน่วยประมวลผลกลาง เวลาในการค้นหาที่อยู่ของเพจในตารางเพจ เวลาที่จัดการสัญญาณขัดจังหวะ เวลาที่ใช้ในการคำนวณพาริตีเพจ เวลาที่เครื่องผู้ให้บริการทำการประมวลผลคำร้องขอเพจที่ส่งจากเครื่องผู้ขอใช้บริการ และแบบจำลองระยะไกลกับแบบจำลองระยะไกลแบบจำกัดไม่รวมเวลาเปิดช่องการสื่อสาร เพราะยึดการทำงานของโพรโตคอลยูดีพี เครือข่ายมีความเร็วในการรับส่งข้อมูลและประสิทธิภาพคงที่ตลอดการทำงาน ซึ่งผู้ที่สนใจสามารถนำแบบจำลองทั้ง 3 แบบมาปรับปรุงให้รองรับการทำงานส่วนนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ในสภาพแวดล้อมถูกจำกัดด้านจำนวนเครื่องผู้ให้บริการ 3-9 เครื่อง ทำให้ในขอบเขตสภาพแวดล้อมบางประการ เช่น เครือข่ายที่มีความเร็วในการรับส่งข้อมูลและประสิทธิภาพต่ำ ๆ การทำหน่วยความจำเสมือนแบบสลับหน้าระยะไกลมีประสิทธิภาพน้อยกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ ซึ่งถ้าเพิ่มจำนวนเครื่องผู้ให้บริการก็อาจจะแสดงให้เห็นถึงประสิทธิภาพของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลได้ดีขึ้น

3. การนำเพจกลับลงฮาร์ดดิสก์จะไม่มีการตรวจสอบว่าเพจมีการเปลี่ยนแปลงหรือไม่จะนำเพจเขียนทับทุกครั้ง ถ้ามีการตรวจสอบการเปลี่ยนแปลงก่อนอาจจะทำให้เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จลดลง

4. คำร้องขอเพจจากเครื่องผู้ขอใช้บริการถือว่ามีความสำคัญสูงกว่าทุกคำร้องทำให้เครื่องผู้ให้บริการทำการประมวลผลคำร้องนี้ทันที หากคำร้องนี้มีความสำคัญเท่า ๆ กับคำร้องอื่น ๆ อาจจะทำให้เวลาเฉลี่ยที่โปรแกรมใช้ในการประมวลผลจนเสร็จของหน่วยความจำเสมือนแบบสลับหน้าระยะไกลเพิ่มขึ้น

5.2.2 เกี่ยวกับการนำไปใช้จริง

ผลการทดลองและความรู้เรื่องขอบเขตของสภาพแวดล้อมที่หน่วยความจำเสมือนแบบสลับหน้าระยะไกลยังสามารถทำงานได้อย่างมีประสิทธิภาพมากกว่าหน่วยความจำเสมือนแบบสลับหน้าปกติ จะเป็นประโยชน์กับหน่วยงานหรือองค์กรที่สนใจจะนำวิธีหน่วยความจำเสมือนแบบสลับหน้าระยะไกลไปใช้งานจริง และหากการทดลองกำหนดสภาพแวดล้อมไม่ตรงกับสภาพแวดล้อมจริงในองค์กร ก็สามารถนำแบบจำลองทั้ง 3 มาปรับเปลี่ยนให้ตรงกับสภาพแวดล้อมจริงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] มงคง อัศวโกวหกรณ์. **ระบบปฏิบัติการ (Operating System).** กรุงเทพมหานคร : บริษัทอินเตอร์เนชั่นเนลอินฟอร์เมติกส์ คอร์ปอเรชั่น จำกัด. 2535

[2] มูอัมหมัดอัฎฮา ยอดิง. **"ศัพท์เกี่ยวกับระบบปฏิบัติการ."** [Online]. Available : http://bala.riy.ac.th/~404559035/work3.html. 2546

[3] พงษ์ระพี. **คอมพิวเตอร์เข้าใจง่าย สไตล์ 3 มิติ.** พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน). 2539

[4] พิมลพรรณ ลีลาภัทรพันธุ์. **"ระบบปฏิบัติการ 1"** [Online]. Available : http://yala.riy.ac.th/%7Epimonpun/4121401-OS/4121401index.html. 2546

[5] พร้อมเลิศ หล่อวิจิตร. **"ประสิทธิภาพของฮาร์ดดิสก์"** [Online]. Available : http://www.provision.co.th/pcdirect/index.php?itemid=18&catid=5

[6] Andrew S. Tanenbaum. **Computer Networks.** Pearson Education Indochina LTD. 1999

[7] Evangelos P. M. and George D. **" Implementation of a Reliable Remote Memory Pager."** [Online]. Available : http://www.ics.forth.gr/carv/r-d-activities/telegraphos/usenix96/paper.html. 1996

[8] Evangelos P. Markatos. **" Using Remote Memory to avoid Disk Thrashing: A Simulation Study."** [Online]. Available : http://citeseer.nj.nec.com/markatos96using.html. 1996

[9] Evangelos P. Markatos. **" Issues in Reliable a Network Memory Paging."** [Online]. Available : http://citeseer.nj.nec.com/markatos96issues.html. 1996

[10] George D. and Evangelos P. M. **" Adaptive and Reliable Paging to Remote Main Memory."** Journal of Parallel and Distributed Computing 58. 1999. pp.357-388.

[11] Michail D. F. and Evangelos P. M. **" The Network RamDisk : Using Remote Memory on Heterogeneous NOWs."** [Online]. Available : http://citeseer.nj.nec.com/flouris99network.html. 1999

[12] Peter Baer G., ABRAHAM S. and GREG G. **Operating System Concepts. Sixth Edition.** New York : John Wiley & Sons Inc. 2003

[13] http://www.seagate.com/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ตัวแปรที่ใช้ในแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 ตัวแปรแบบจำลองปกติ

ที่	ชื่อตัวแปร	ความหมาย	ค่าตัวแปร	หน่วย
1.	Bsize	ขนาดของบล็อก	4	เพจ
2.	BlockN	เลขบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล	-	-
3.	BlockOut	เลขบล็อกที่จะถูกแทนที่เมื่อหน่วยความจำหลักเต็ม	-	-
4.	IDfreeRAM	ตำแหน่งที่ว่างในหน่วยความจำหลัก		
5.	HDAceess	เวลาที่ใช้ในการเข้าถึงข้อมูลที่อยู่ในฮาร์ดดิสก์	0.015325 ดูรายละเอียดสมการที่ 3.3	วินาที
6.	Latency	เวลาแฝงของฮาร์ดดิสก์	30/RPM	วินาที
7.	ma	เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลที่อยู่ในหน่วยความจำหลัก	0.000000001	วินาที
8.	numRam	จำนวนบล็อกที่อยู่ในหน่วยความจำหลัก	-	บล็อก
9.	PageIn	หมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล	สุ่มด้วยฟังก์ชัน UNIF(1,PGsize)	-
10.	PageRef	จำนวนครั้งของการอ้างถึงเพจที่โปรแกรมต้องการในการประมวลผลจนเสร็จ	1000	ครั้ง
11.	PageSize	ขนาดของเพจ	4	กิโลไบต์ (KByte)
12.	pt	ตารางที่จะใช้เก็บที่อยู่ของเพจทุกเพจของโปรแกรม	1 คืออยู่ฮาร์ดดิสก์ 2 คือ อยู่หน่วยความจำหลัก	-
13.	PGsize	ขนาดโปรแกรมที่ประมวลผลที่เครื่องผู้ขอใช้บริการ	4 - 320	เพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 (ต่อ)

ที่	ชื่อตัวแปร	ความหมาย	ค่าตัวแปร	หน่วย
14.	PTsize	ขนาดของตารางที่จะใช้เก็บที่อยู่ของเพจทุกเพจของโปรแกรม	PGsize/Bsize	บล็อก
15.	rc	ตารางเก็บเลขบล็อกที่อยู่ในหน่วยความจำหลัก	-	-
16.	RCsize	ขนาดของตาราง rc	Rsize / Bsize	บล็อก
17.	RPM	อัตราการหมุนของฮาร์ดดิสก์	7200	รอบต่อนาที
18.	seek	เวลาแสวงหาของฮาร์ดดิสก์	0.008	วินาที
19.	totalTime	เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ	-	-
20.	TranferHD	ความเร็วในการรับส่งของฮาร์ดดิสก์	5	เมกะไบต์ต่อวินาที
21.	TranHD	เวลาที่ใช้ในการรับส่งข้อมูล 1 บล็อกของฮาร์ดดิสก์	ขนาดบล็อก/ความเร็วในการรับส่งฮาร์ดดิสก์	วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.2 ตัวแปรแบบจำลองระยะไกลและแบบจำลองระยะไกลแบบจำกัด

ที่	ชื่อตัวแปร	ความหมาย	ค่าตัวแปร	หน่วย
1.	aa	ตัวแปรนับ	-	-
	Bsize	ขนาดของบล็อก	4	เพจ
2.	BlockN	เลขบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผล	-	-
3.	BlockOut	เลขบล็อกที่จะถูกแทนที่เมื่อหน่วยความจำหลักเต็ม	-	-
4.	BW	แบนด์วิดธ์	10-100	เมกะบิตต่อวินาที
5.	CE	ประสิทธิภาพการรับส่งในเครือข่าย	10-100	เปอร์เซ็นต์
6.	HDAccess	เวลาที่ใช้ในการเข้าถึงข้อมูลที่อยู่ในฮาร์ดดิสก์	0.015325 ดูรายละเอียดสมการที่ 3.3	วินาที
7.	IDcolumn	ตำแหน่งคอลัมน์	-	-
8.	IDcolumnPS	ตำแหน่งคอลัมน์ในตาราง ps	-	-
9.	IDfreeRAM	ตำแหน่งที่ว่างในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ	-	-
10.	IDfreeServer	ตำแหน่งที่ว่างในหน่วยความจำหลักของเครื่องผู้ให้บริการ	-	-
11.	IDfreeSt	ตำแหน่งที่ว่างในอเดปทีปพาริตีแคช	-	-
12.	IDrow	ตำแหน่งแถว	-	-
13.	Latency	เวลาแฝงของฮาร์ดดิสก์	30/RPM	วินาที
14.	ma	เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลที่อยู่ในหน่วยความจำหลัก	0.000000001	วินาที
15.	MsSize	ขนาดของข้อมูลที่ส่งผ่านเครือข่าย 1 ครั้ง	131584	บิต
16.	n	ตัวแปรนับ	-	-
17.	numInHD	จำนวนครั้งที่เพจอยู่ในฮาร์ดดิสก์	-	ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.2 (ต่อ)

ที่	ชื่อตัวแปร	ความหมาย	ค่าตัวแปร	หน่วย
18.	numInRam	จำนวนครั้งที่เพจอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ	-	ครั้ง
19.	numInServer	จำนวนครั้งที่เพจอยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ	-	ครั้ง
20.	numRam	จำนวนเพจที่อยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ	-	เพจ
21.	numSentHD	จำนวนครั้งที่ส่งเพจจากอเดปทีปพาริตีแคชไปยังฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการ	-	ครั้ง
22.	numTrans	จำนวนครั้งที่ส่งข้อมูลผ่านเครือข่ายในการอ้างถึงเพจ 1 รอบ	-	ครั้ง
23.	numVic	จำนวนครั้งที่ใส่เพจในวิกทิมแคช	-	ครั้ง
24.	Nstream	จำนวนสตรีมแคชรวมกับวิกทิมแคชในเครื่องผู้ขอใช้บริการ	4	สตรีม
25.	Nserver	จำนวนเครื่องผู้ให้บริการทั้งหมดในเครือข่ายซึ่งรวมทั้งเครื่องผู้ให้บริการที่เก็บข้อมูลและเก็บพาริตี้เพจ	3-9	เครื่อง
26.	Overhead	ขนาดของข้อมูลส่วนหัวที่เพิ่มขึ้นเมื่อส่งเพจผ่านเครือข่าย	64	ไบต์
27.	ps	ตารางที่เก็บว่าบล็อกใดอยู่ที่เครื่องผู้ให้บริการใด	-	-
28.	pt	ตารางเก็บที่อยู่ของบล็อก	1 คืออยู่ฮาร์ดดิสก์ 2 คือ อยู่หน่วยความจำหลักเครื่องผู้ขอใช้บริการ 3 คือ อยู่หน่วยความจำหลักเครื่องผู้ให้บริการ	-
29.	PageIn	หมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล	สุ่มด้วยฟังก์ชัน UNIF (1,PGsize)	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.2 (ต่อ)

ที่	ชื่อตัวแปร	ความหมาย	ค่าตัวแปร	หน่วย
30.	PageRef	จำนวนครั้งการอ้างถึงเพจที่โปรแกรมต้องการใช้จนจบการทำงาน	1000	ครั้ง
31.	PageSize	ขนาดของเพจ	4	กิโลไบต์
32.	PGsize	ขนาดโปรแกรมที่ประมวลผลที่เครื่องผู้ขอใช้บริการ	4 - 320	เพจ
33.	PTsize	ขนาดของตาราง pt	PGsize/Bsize	บล็อก
34.	rc	ตารางเก็บเลขบล็อกที่อยู่ในหน่วยความจำหลัก	-	-
35.	RCsize	ขนาดของตาราง rc	Rsize / Bsize	บล็อก
36.	Rsize	ขนาดหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ	4-40	เพจ
37.	RPM	อัตราการหมุนของฮาร์ดดิสก์	7200	รอบต่อนาที
38.	seek	เวลาแสวงหาของฮาร์ดดิสก์	0.008	วินาที
	st	ตารางเก็บเพจที่อยู่ในอเดปทีปพาริตีแคช	-	-
39.	Ssize	ขนาดหน่วยความจำหลักของเครื่องผู้ให้บริการ	4-40	เพจ
40	RSsize	ขนาดหน่วยความจำหลักของเครื่องผู้ให้บริการ	4-40	เพจ
41.	temp	ตัวแปรชั่วคราว	-	-
42.	tempPlus	ตัวแปรรวมชั่วคราว	-	-
43.	totalTime	เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ	-	วินาที
44.	TimeCS	เวลาในการทำคอนเท็กซ์สวิสซ์	0.0005	วินาที
45.	TimeDelay	เวลาที่รอให้สายสัญญาณว่าง	ดูรายละเอียดสมการที่ 3.22	วินาที
46.	TranferHD	ความเร็วในการรับส่งของฮาร์ดดิสก์	5	เมกะไบต์ต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.2 (ต่อ)

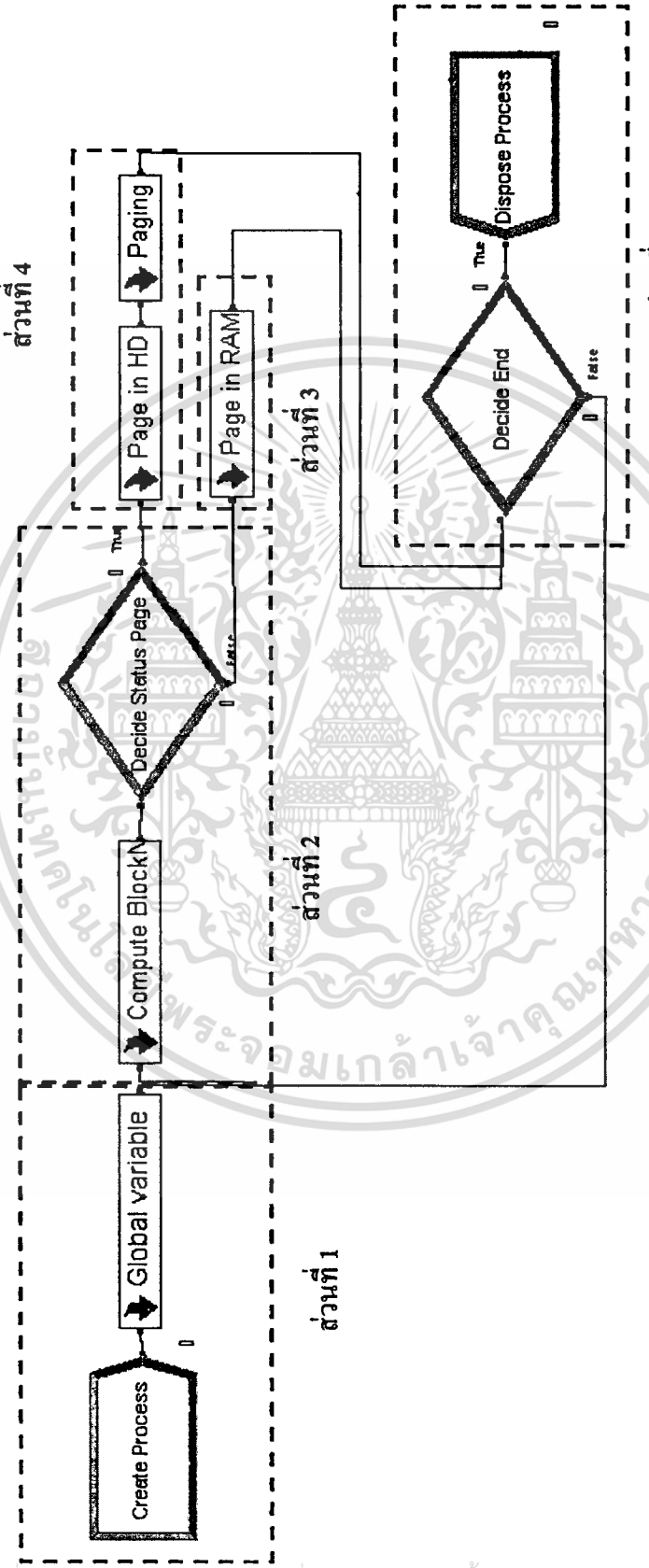
ที่	ชื่อตัวแปร	ความหมาย	ค่าตัวแปร	หน่วย
47.	TranHD	เวลาที่ใช้ในการรับส่งข้อมูล 1 บล็อก ของฮาร์ดดิสก์	ขนาดบล็อก/ ความเร็วในการ รับส่งฮาร์ดดิสก์	วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

แบบจำลองปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



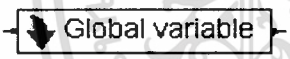
รูปที่ ข.1 แบบจำลองปกติ

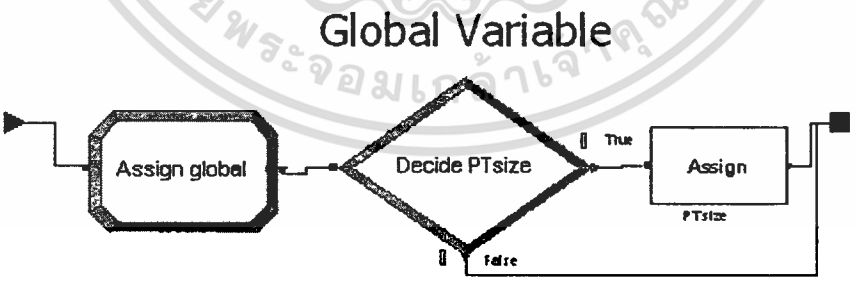
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 ส่วนเริ่มต้นการทำงาน เป็นส่วนที่สร้างเอ็นติตี ซึ่งกำหนดให้สร้างเพียง 1 เอ็นติตี เพื่อให้สอดคล้องกับสภาพการทำงานของเครื่องผู้ขอใช้บริการที่ไม่มีการทำงานอื่นนอกจากการทดลอง จากนั้นเอ็นติตีจะเข้าสู่โมเดลย่อย Global bariable ที่ทำการกำหนดค่าเริ่มต้นตัวแปรทั่วไปของแบบจำลอง ซึ่งส่วนที่ 1 นี้ประกอบด้วยโมดูลและโมเดลย่อยแสดงดังรูปที่ ข.2 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 1 แสดงดังตารางที่ ข.1

รูปที่ ข.2 ส่วนที่ 1

ตาราง ข.1 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 1

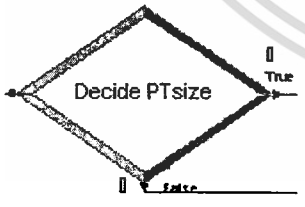
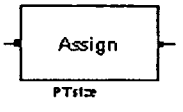
โมดูลและโมเดลย่อย	การทำงาน
	สร้างเอ็นติตีเพื่อให้เป็นดำเนินการของแบบจำลอง จำนวน 1 เอ็นติตี
	โมเดลย่อย Global variable ทำการกำหนดค่าตัวแปรทั่วไปของแบบจำลอง รายละเอียดภายในแสดงดังรูปที่ ข.3 และตารางที่ ข.2



รูปที่ ข.3 ภายในโมเดลย่อย Global Variable

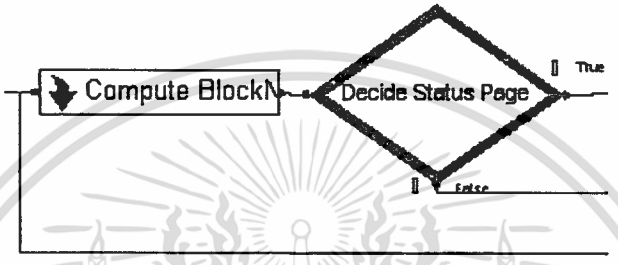
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ข.2 การทำงานของโมดูลในโมเดลย่อย Global Variable

โมดูล	การทำงาน
	-กำหนดค่าเริ่มต้นของตัวแปร totalTime = 0 -กำหนดค่าเริ่มต้นของตัวแปร PageRef = 1000 เพื่อใช้เป็นจำนวนครั้งของการอ้างถึงเพจที่โปรแกรมต้องการในการประมวลผลจนเสร็จ -กำหนดค่าเริ่มต้นของตัวแปร PTsize = PGsize/Bsize เพื่อคำนวณหาจำนวนแถวของตารางบล็อกซึ่งเป็นตารางที่จะใช้เก็บที่อยู่ของเพจทุกเพจของโปรแกรมหรือตาราง pt -กำหนดค่าเริ่มต้นของตัวแปร RCsize = Rsize/Bsize เพื่อคำนวณหาจำนวนคอลัมน์ของตารางที่ใช้เก็บเพจที่อยู่ในหน่วยความจำหลักหรือตาราง rc ซึ่งถ้าหารไม่ลงตัวจะตัดเศษทิ้งเนื่องจากจองพื้นที่ในหน่วยความจำหลักของเครื่องไว้แล้วไม่สามารถเพิ่มเติมพื้นที่ได้ -กำหนดค่าเริ่มต้นของตัวแปร Latency = 30/RPM เพื่อหาเวลาเฉลี่ยของเวลาแสวงหาของฮาร์ดดิสก์ -กำหนดค่าเริ่มต้นของตัวแปร TranHD=(Bsize * PageSize)/(TranferHD*1024) เพื่อเป็นเวลาที่ใช้ในการรับส่งข้อมูล 1 บล็อกของฮาร์ดดิสก์ -กำหนดค่าเริ่มต้นของตัวแปร HDAccess=seek + Latency + TranHD เพื่อใช้เป็นเวลาเฉลี่ยที่เข้าถึงข้อมูลในฮาร์ดดิสก์
	ตัดสินใจเพิ่มค่าตัวแปร PTsize เนื่องจากถ้าหารไม่ลงตัวจะต้องมีบล็อกเพิ่มอีก 1 บล็อกเพื่อให้สามารถเก็บที่อยู่ของทุกเพจได้
	เพิ่มค่าตัวแปร PTsize อีก 1

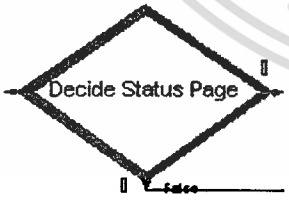
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 2 ส่วนสร้างหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล และตรวจสอบที่อยู่ของเพจ เมื่อเอ็นติตีเข้ามายังส่วนนี้จะได้รับหมายเลขเพจที่โปรแกรมต้องการจากการสุ่มด้วยฟังก์ชันทางคณิตศาสตร์ และคำนวณว่าเพจดังกล่าวอยู่ในบล็อกใด จากนั้นตรวจสอบที่อยู่ของบล็อกจากตารางบล็อกว่าอยู่ที่หน่วยความจำหลักหรือฮาร์ดดิสก์ ถ้าบล็อกอยู่ที่หน่วยความจำหลักก็จะส่งเอ็นติตีไปยังส่วนที่ 3 แต่ถ้าบล็อกอยู่ที่ฮาร์ดดิสก์จะส่งเอ็นติตีไปยังส่วนที่ 4 ซึ่งส่วนที่ 2 นี้ประกอบด้วยโมเดลย่อยและโมดูลแสดงดังรูปที่ ข.4 การทำงานของโมเดลย่อย และโมดูลแสดงดังตารางที่ ข.3

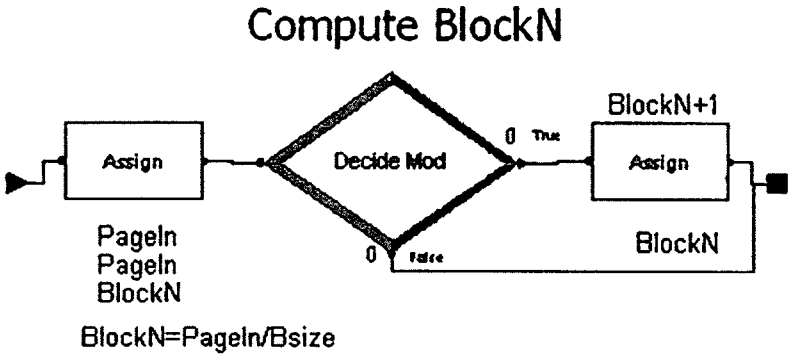


รูปที่ ข.4 ส่วนที่ 2

ตาราง ข.3 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 2

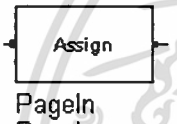
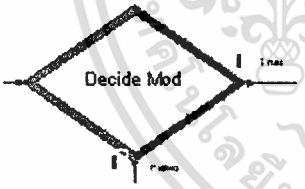
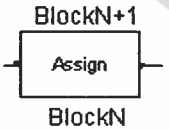
โมดูลและโมเดลย่อย	การทำงาน
	ทำการสุ่มหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผลและคำนวณว่าเพจดังกล่าวอยู่ในบล็อกใด รายละเอียดภายในแสดงดังรูปที่ ข.5 และตารางที่ ข.4
	ตรวจสอบที่อยู่ของเพจที่โปรแกรมต้องการว่าอยู่ที่หน่วยความจำหลักหรือฮาร์ดดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



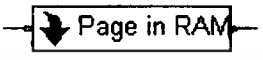
รูปที่ ข.5 ภายในโมเดลย่อย Compute BlockN

ตาราง ข.4 การทำงานของโมดูลในโมเดลย่อย Compute BlockN

โมดูล	การทำงาน
	-กำหนดค่าตัวแปร PageIn =หมายเลขเพจที่โปรแกรมต้องการที่ได้จากการสุ่มด้วยฟังก์ชันทางคณิตศาสตร์ UNIF (1,PGsize) ซึ่งจะได้ค่าออกมาเป็นเลขทศนิยมจึงต้องทำให้เป็นเลขจำนวนเต็ม PageIn =PageIn-AMOD(PageIn,1) -คำนวณว่าเพจที่โปรแกรมต้องการอยู่ที่บล็อกใด BlockN=(PageIn/Bsize)
	ตัดสินใจว่าตัวแปร BlockN ต้องบวก 1 หรือไม่ ในกรณีที่ PageIn หารด้วย Bsize ไม่ลงตัว
	กำหนดค่าตัวแปร BlockN = BlockN +1

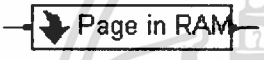
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

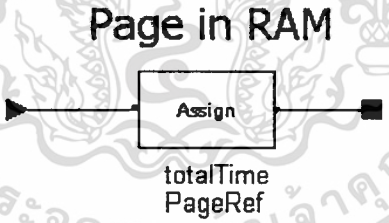
ส่วนที่ 3 ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลัก เมื่อเอ็นติตีเข้ามายังส่วนนี้จึงทำการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ เนื่องจากเพจที่โปรแกรมต้องการอยู่ในหน่วยความจำหลักเรียบร้อยแล้ว และลดจำนวนการอ้างถึงเพจที่โปรแกรมต้องการใช้ในการประมวลผลจนเสร็จ ซึ่งส่วนที่ 3 นี้ประกอบด้วยโมเดลย่อยแสดงดังรูปที่ ข.6 การทำงานของโมเดลย่อยแสดงดังตารางที่ ข.5



รูปที่ ข.6 ส่วนที่ 3

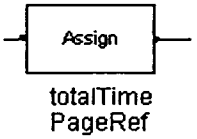
ตาราง ข.5 การทำงานของในโมเดลย่อยในส่วนที่ 3

โมเดล	การทำงาน
	คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลัก รายละเอียดภายในแสดงดังรูปที่ ข.7 และตารางที่ ข.6



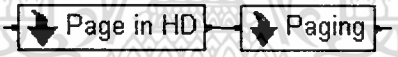
รูปที่ ข.7 ภายในโมเดลย่อย Page in RAM

ตาราง ข.6 การทำงานของโมดูลในโมเดลย่อย Page in RAM

โมดูล	การทำงาน
 totalTime PageRef	-คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ totalTime = totalTime +ma -ลดจำนวนครั้งของการอ้างถึงเพจลง 1 PageRef =PageRef-1

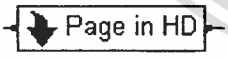
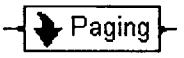
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 4 ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์ เมื่อเอ็นติตีเข้ามาที่ส่วนนี้จะทำการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จและลดจำนวนการอ้างถึงเพจที่โปรแกรมต้องการใช้ในการประมวลผลจนเสร็จ จากนั้นจึงทำการย้ายเพจจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก โดยต้องทำการตรวจสอบว่าหน่วยความจำหลักมีพื้นที่ว่างเพียงพอกับบล็อกใหม่หรือไม่ โดยดูจากจำนวนบล็อกที่อยู่หน่วยความจำหลัก ถ้าพอก็จะทำการย้ายเพจดังกล่าวเข้าสู่หน่วยความจำหลักโดยการเปลี่ยนแปลงค่าในตาราง rc และเปลี่ยนที่อยู่ของบล็อกในตารางบล็อก แต่ถ้าไม่พอก็จะทำการเลือกบล็อกที่จะถูกแทนที่โดยดูลำดับการเข้ามาของบล็อกในตาราง rc ซึ่งจะเลือกบล็อกที่มีลำดับการเข้ามาน้อยที่สุด จากนั้นทำการเปลี่ยนแปลงที่อยู่ของบล็อกที่จะถูกแทนที่ในตารางบล็อก เปลี่ยนแปลงข้อมูลในตาราง rc และลดจำนวนเพจที่อยู่ในหน่วยความจำหลัก ซึ่งจะทำให้เกิดที่ว่างเพียงพอสำหรับบล็อกใหม่ จากนั้นจึงนำบล็อกที่มีเพจที่โปรแกรมต้องการจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลักโดยการเปลี่ยนแปลงที่อยู่ของบล็อกที่มีเพจที่โปรแกรมต้องการในตารางบล็อก เปลี่ยนแปลงข้อมูลในตาราง rc และเพิ่มจำนวนบล็อกที่อยู่ในหน่วยความจำหลัก ซึ่งส่วนที่ 4 นี้ประกอบด้วยโมเดลย่อยแสดงดังรูปที่ ข.8 การทำงานของโมเดลย่อยในส่วนที่ 4 แสดงดังตารางที่ ข.7

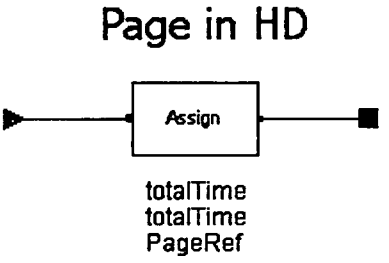


รูปที่ ข.8 ส่วนที่ 4

ตาราง ข.7 การทำงานของโมเดลย่อยในส่วนที่ 4

โมเดลย่อย	การทำงาน
	คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ เมื่อเพจที่โปรแกรมต้องการอยู่ในฮาร์ดดิสก์ รายละเอียดภายในแสดงดังรูปที่ ข.9 และตารางที่ ข.8
	ตรวจสอบว่าหน่วยความจำหลักมีที่ว่างเพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการหรือไม่ ถ้าพอก็คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ ถ้าไม่พอก็ทำการเลือกบล็อกที่จะถูกแทนที่และนำไปเก็บที่ฮาร์ดดิสก์ และนำบล็อกที่มีเพจที่โปรแกรมต้องการเข้าสู่หน่วยความจำหลัก รายละเอียดภายในแสดงดังรูปที่ ข.10 และตารางที่ ข.9

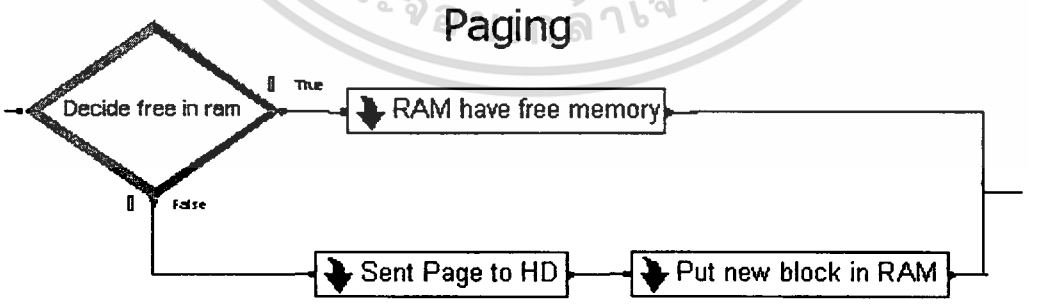
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.9 ภายในโมเดลย่อย Page in HD

ตาราง ข.8 การทำงานของโมดูลในโมเดลย่อย Page in HD

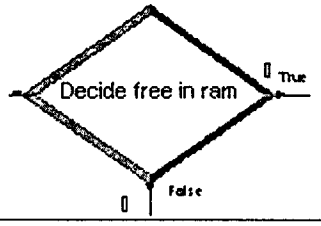
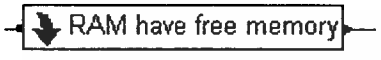
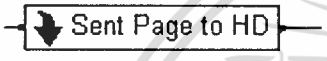
โมดูล	การทำงาน
totalTime totalTime PageRef	-คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่ +เวลาเฉลี่ยที่ใช้ในการนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก totalTime = totalTime + ma + HDAccess -ลดจำนวนครั้งของการอ้างถึงเพจ PageRef =PageRef-1

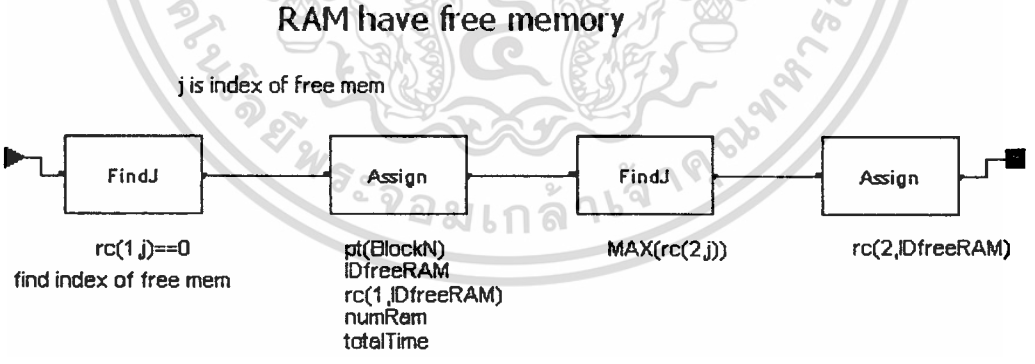


รูปที่ ข.10 ภายในโมเดลย่อย Paging

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ข.9 การทำงานของโมดูลในโมเดลย่อย Paging

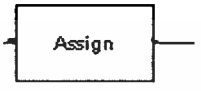
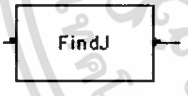
โมดูล	การทำงาน
	ตรวจสอบว่าหน่วยความจำหลักมีพื้นที่ว่างเพียงพอสำหรับบล็อกใหม่หรือไม่ ด้วยการดูจำนวนบล็อกที่อยู่ในหน่วยความจำหลัก ซึ่งถ้ามีที่ว่างเพียงพอจำนวนบล็อกต้องน้อยกว่าขนาดหน่วยความจำหลัก numRam < RCsize
	โมเดลย่อย RAM have free memory เป็นส่วนที่จัดการเมื่อหน่วยความจำหลักมีพื้นที่ว่างเพียงพอสำหรับเพจที่โปรแกรมต้องการ รายละเอียดภายในแสดงดังรูปที่ ข.11 และตารางที่ ข.10
	โมเดลย่อย Sent Page to HD เป็นส่วนที่จัดการนำเพจที่จะถูกแทนที่กลับลงฮาร์ดดิสก์ รายละเอียดภายในแสดงดังรูปที่ ข.12 และตารางที่ ข.11
	โมเดลย่อย Put new block in RAM เป็นส่วนที่จัดการนำเพจที่โปรแกรมต้องการจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก รายละเอียดภายในแสดงดังรูปที่ ข.13 และตารางที่ ข.12



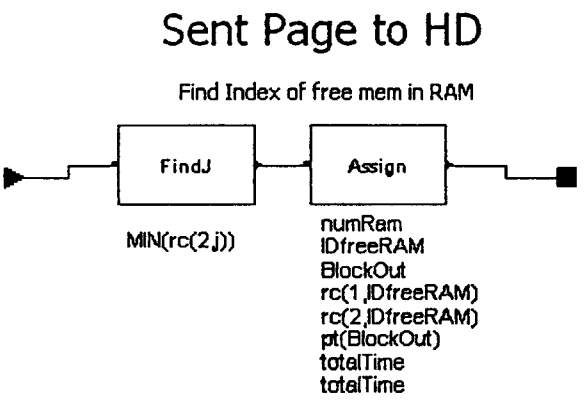
รูปที่ ข.11 ภายในโมเดลย่อย RAM have free memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ข.10 การทำงานของโมดูลในโมเดลย่อย RAM have free memory

โมดูล	การทำงาน
j is index of free mem rc(1,j)==0 find index of free mem	ค้นหาตำแหน่งที่ว่างในหน่วยความจำหลัก ซึ่งถ้าพบที่ว่างในหน่วยความจำหลักค่าตัวแปร j จะเป็นตำแหน่งที่ว่าง ถ้าไม่มีที่ว่างตัวแปร j =0
 pt(BlockN) IDfreeRAM rc(1,IDfreeRAM) numRam totalTime	-เปลี่ยนที่อยู่บล็อกในตารางบล็อก pt(BlockN)=2 ซึ่งหมายถึงอยู่ในหน่วยความจำหลัก -กำหนดค่าตัวแปร IDfreeRAM=j -นำบล็อกเข้าสู่หน่วยความจำหลักด้วยการเปลี่ยนแปลงค่าในตาราง rc(1,IDfreeRAM)= BlockN -เพิ่มจำนวนบล็อกที่อยู่ในหน่วยความจำหลัก numRam =numRam+1 -คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ totalTime = totalTime + ma
 MAX(rc(2,j))	ค้นหาลำดับสูงสุดของการเข้ามาอยู่ในหน่วยความจำหลักจากตาราง rc
 rc(2,IDfreeRAM)	กำหนดลำดับการเข้ามาของบล็อกใหม่ โดยใช้ลำดับสูงสุดที่หาได้จากโมดูลก่อนหน้ามาบวก 1 rc(2,IDfreeRAM)=rc(2,j)+1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

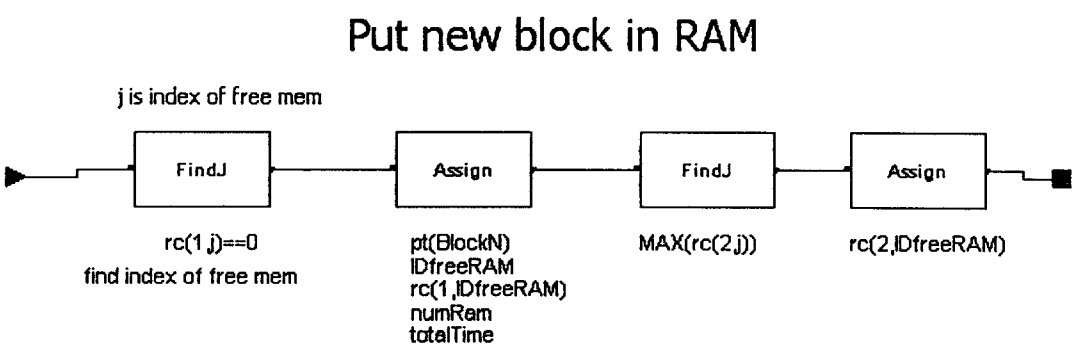


รูปที่ ข.12 ภายในโมเดลย่อย Sent Page to HD

ตาราง ข.11 การทำงานของโมดูลในโมเดลย่อย Sent Page to HD

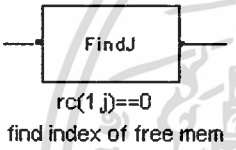
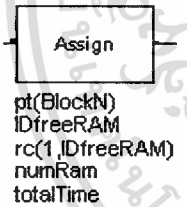
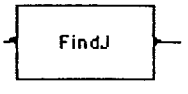
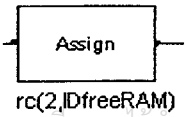
โมดูล	การทำงาน
FindJ MIN(rc(2,j))	ค้นหาลำดับน้อยสุดของการเข้ามาอยู่ในหน่วยความจำหลักของบล็อกจากตาราง rc เพื่อเลือกบล็อกที่จะถูกแทนที่
Assign numRam IDfreeRAM BlockOut rc(1,IDfreeRAM) rc(2,IDfreeRAM) pt(BlockOut) totalTime totalTime	-ลดจำนวนเพจในหน่วยความจำหลัก numRam =numRam-1 -กำหนดค่าตัวแปร IDfreeRAM=j ซึ่งเป็นลำดับน้อยสุดที่เข้ามาอยู่ในหน่วยความจำหลักของบล็อกจากโมดูลก่อนหน้า -เลือกบล็อกที่จะถูกแทนที่ และนำบล็อกออกจากหน่วยความจำหลัก BlockOut =rc(1,j) -เปลี่ยนแปลงข้อมูลในตาราง rc เพื่อให้เกิดที่ว่างจากการเลือกบล็อกที่จะถูกแทนที่rc(1,IDfreeRAM)=0 rc(2,IDfreeRAM)=0 -เปลี่ยนแปลงที่อยู่ของบล็อกที่จะถูกแทนที่ในตารางบล็อก pt(BlockOut) = 1 ซึ่งหมายถึงอยู่ที่ฮาร์ดดิสก์ -คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลักเพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่ +เวลาเฉลี่ยที่ใช้ในการนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก totalTime =totalTime+ma+HDAccess

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



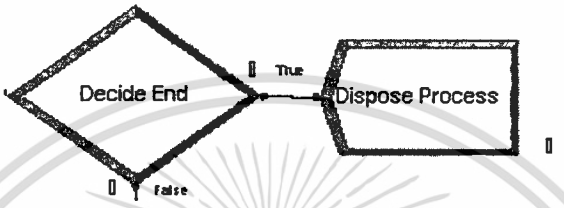
รูปที่ ข.13 ภายในโมเดลย่อย Put new block in RAM

ตาราง ข.12 การทำงานของโมดูลในโมเดลย่อย Put new block in RAM

โมดูล	การทำงาน
	ค้นหาตำแหน่งที่ว่างในหน่วยความจำหลัก จากตาราง rc ซึ่งเป็นที่ว่างที่เกิดจากนำบล็อกที่จะถูกแทนที่กลับลงฮาร์ดดิสก์แล้ว
	-เปลี่ยนสถานะบล็อกที่จะนำเข้าสู่หน่วยความจำหลัก pt(BlockN)=2 ซึ่งหมายถึงอยู่ในหน่วยความจำหลัก -กำหนดค่าตัวแปร IDfreeRAM =j ซึ่งเป็นตำแหน่งที่ว่างจากโมดูลก่อนหน้า -นำบล็อกใหม่เข้าสู่หน่วยความจำหลัก rc(1,IDfreeRAM)=BlockN -เพิ่มจำนวนบล็อกในหน่วยความจำหลัก numRam=numRam+1 -คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ= เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ totalTime = totalTime + ma
	ค้นหาลำดับสูงสุดของการเข้ามาอยู่ในหน่วยความจำหลัก
	กำหนดลำดับการเข้ามาในหน่วยความจำหลักของบล็อกใหม่ rc(2,IDfreeRAM)= rc(2,j)+1

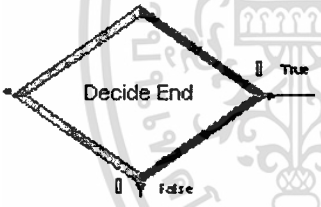
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 5 ส่วนตัดสินใจว่าสิ้นสุดการทำงานหรือไม่ เมื่อเอ็นติตีเข้ามายังส่วนนี้จะตรวจสอบว่าสิ้นสุดการทำงานหรือยัง โดยดูจากจำนวนการอ้างถึงเพจที่โปรแกรมต้องการใช้ในการประมวลผลจนเสร็จว่าเท่ากับ 0 หรือยัง ถ้าใช่แสดงว่าสิ้นสุดการทำงาน เอ็นติตีจะถูกทำลาย แต่ถ้าไม่ใช่เอ็นติตีจะถูกส่งไปยังส่วนที่ 2 เพื่อรับหมายเลขที่โปรแกรมต้องการใช้ในการประมวลผลครั้งต่อไป ซึ่งส่วนที่ 5 นี้ประกอบด้วยโมดูลต่าง ๆ แสดงดังรูปที่ ข.14 ซึ่งการทำงานของโมดูลแสดงดังตารางที่ ข.13



รูปที่ ข.14 ส่วนที่ 5

ตาราง ข.13 การทำงานของโมดูลในส่วนที่ 5

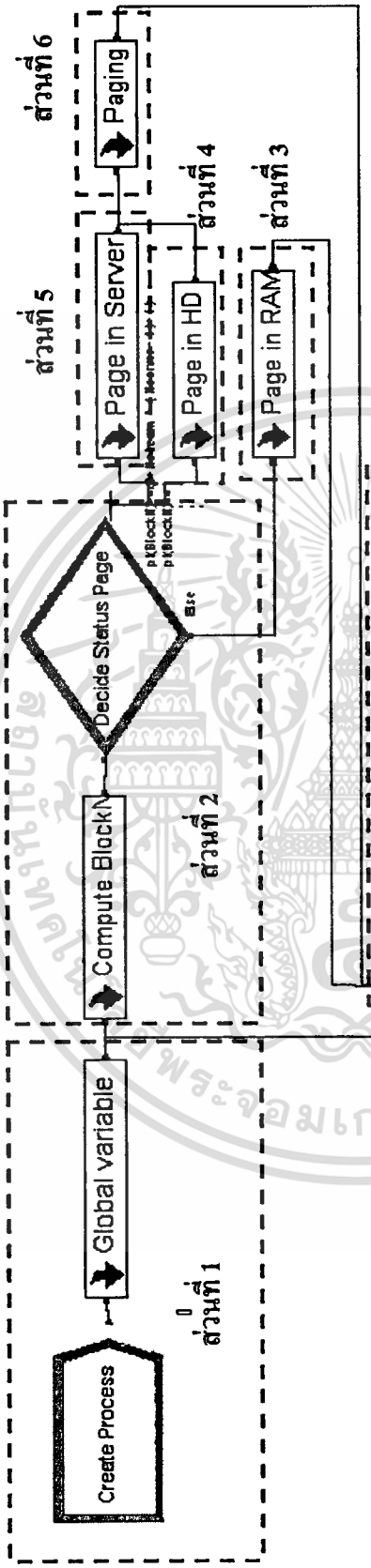
โมดูล	การทำงาน
	ตัดสินใจว่าสิ้นสุดการทำงานหรือไม่ ซึ่งตรวจสอบจาก PageRef == 0 หรือไม่ถ้าใช่ถือว่าสิ้นสุดการทำงาน เอ็นติตีจะถูกทำลาย ถ้าไม่ใช่จะนำเอ็นติตีกลับไปยังส่วนที่ 2 อีกครั้ง
	สิ้นสุดการทำงานทำลายเอ็นติตี และแสดงผลลัพธ์ที่ได้จากการทดลองซึ่งก็คือ เวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

แบบจำลองระยะไกลและแบบจำลองระยะไกลแบบจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



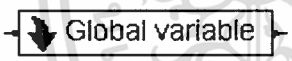
รูปที่ ค.1 แบบจำลองระยะใกล้และแบบจำลองระยะใกลแบบจำกัด

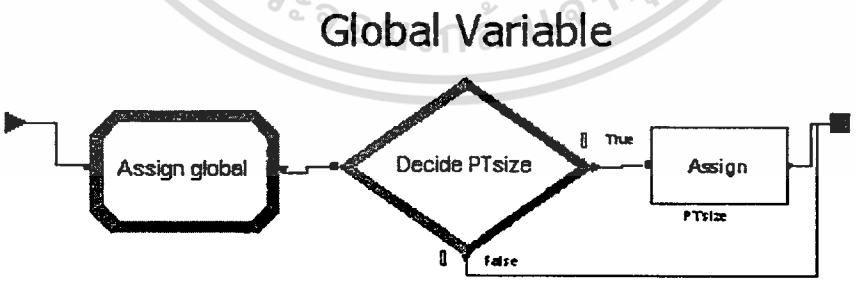
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 ส่วนเริ่มต้นการทำงาน เป็นส่วนที่สร้างเอ็นติตี ซึ่งกำหนดให้สร้างเพียง 1 เอ็นติตี เพื่อให้สอดคล้องกับสภาพการทำงานของเครื่องผู้ขอใช้บริการที่ไม่มีการทำงานอื่นนอกจากการทดลอง จากนั้นเอ็นติตีจะเข้าสู่โมเดลย่อย Global bariable ซึ่งเป็นส่วนที่กำหนดค่าเริ่มต้นตัวแปรทั่วไปของแบบจำลอง ซึ่งส่วนที่ 1 ประกอบด้วยโมดูลและโมเดลย่อยแสดงดังรูปที่ ค.2 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 1 แสดงดังตารางที่ ค.1

รูปที่ ค.2 ส่วนที่ 1

ตารางที่ ค.1 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 1

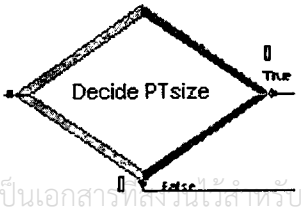
โมดูลและโมเดลย่อย	การทำงาน
	สร้างเอ็นติตีเพื่อให้เป็นดำเนินการของแบบจำลอง จำนวน 1 เอ็นติตี
	โมเดลย่อย Global variable ทำการกำหนดค่าตัวแปรทั่วไปของแบบจำลอง รายละเอียดภายในแสดงดังรูปที่ ค.3 และการทำงานของโมดูลแสดงดังตารางที่ ค.2



รูปที่ ค.3 ภายในโมเดลย่อย Global Variable

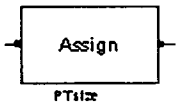
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.2 การทำงานของโมดูลในโมเดลย่อย Global Variable

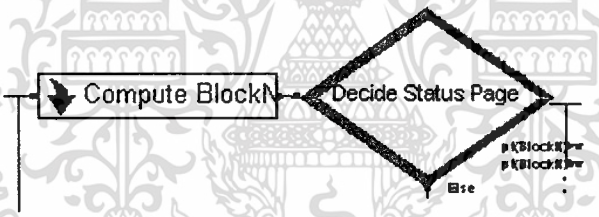
โมดูล	การทำงาน
	-กำหนดค่าเริ่มต้นของตัวแปร totalTime = 0 -กำหนดค่าเริ่มต้นของตัวแปร PageRef = 1000 เพื่อใช้เป็นจำนวนครั้งของการอ้างถึงเพจที่โปรแกรมต้องการในการประมวลผลจนเสร็จ -กำหนดค่าเริ่มต้นของตัวแปร PTsize = PGsize/Bsize เพื่อคำนวณหาขนาดของตารางบล็อกซึ่งเป็นตารางที่จะใช้เก็บที่อยู่ของเพจทุกเพจของโปรแกรมหรือตาราง pt -กำหนดค่าเริ่มต้นของตัวแปร RCsize = Rsize/Bsize เพื่อคำนวณหาจำนวนคอลัมน์ของตารางที่ใช้เก็บเพจที่อยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการหรือตาราง rc ซึ่งถ้าหารไม่ลงตัวจะตัดเศษทิ้งเนื่องจากจองพื้นที่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการไว้แล้วไม่สามารถเพิ่มเติมพื้นที่ได้ -กำหนดค่าเริ่มต้นของตัวแปร RSsize = Ssize/Bize เพื่อคำนวณหาจำนวนคอลัมน์ของตารางที่ใช้เก็บเพจที่อยู่หน่วยความจำหลักของเครื่องผู้ให้บริการ หรือตาราง ps ซึ่งถ้าหารไม่ลงตัวจะตัดเศษทิ้งเนื่องจากจองพื้นที่ในหน่วยความจำหลักของเครื่องผู้ให้บริการไว้แล้วไม่สามารถเพิ่มเติมพื้นที่ได้ -กำหนดค่าเริ่มต้นของตัวแปร Latency = 30/RPM เพื่อหาเวลาเฉลี่ยของเวลาแสวงหาของฮาร์ดดิสก์ -กำหนดค่าเริ่มต้นของตัวแปร TranHD=(Bsize * PageSize)/(TranferHD*1024) เพื่อเป็นเวลาที่ใช้ในการรับส่งข้อมูล 1 บล็อกของฮาร์ดดิสก์ -กำหนดค่าเริ่มต้นของตัวแปร HDAccess=seek + Latency + TranHD เพื่อใช้เป็นเวลาเฉลี่ยที่เข้าถึงข้อมูลในฮาร์ดดิสก์
	ตัดสินใจเพิ่มค่าตัวแปร PTsize เนื่องจากถ้าหารไม่ลงตัวจะต้องมีบล็อกเพิ่มอีก 1 บล็อกเพื่อให้เพจทุกเพจอยู่ครบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.2 (ต่อ)

โมดูล	การทำงาน
	เพิ่มค่าตัวแปร PTsize อีก 1

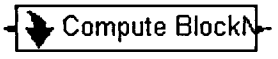
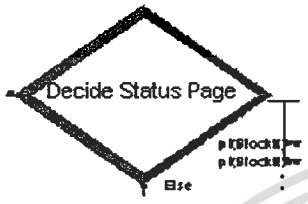
ส่วนที่ 2 ส่วนสร้างหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผล และตรวจสอบที่อยู่ของเพจ เมื่อเอ็นติตีเข้ามายังส่วนนี้จะได้รับหมายเลขเพจที่โปรแกรมต้องการจากการสุ่มด้วยฟังก์ชันทางคณิตศาสตร์ และคำนวณว่าเพจดังกล่าวอยู่ในบล็อกใด จากนั้นตรวจสอบที่อยู่ของบล็อกจากตารางบล็อกว่าอยู่ที่หน่วยความจำหลักหรือฮาร์ดดิสก์ ถ้าบล็อกอยู่ที่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการก็จะส่งเอ็นติตีไปยังส่วนที่ 3 แต่ถ้าบล็อกอยู่ที่ฮาร์ดดิสก์จะส่งเอ็นติตีไปยังส่วนที่ 4 และถ้าบล็อกอยู่ที่หน่วยความจำหลักของเครื่องผู้ให้บริการก็จะส่งเอ็นติตีไปยังส่วนที่ 5 ซึ่งส่วนที่ 2 นี้ประกอบด้วยโมเดลย่อยและโมดูลแสดงดังรูปที่ ค.4 การทำงานของโมเดลย่อย และโมดูลแสดงดังตารางที่ ค.3

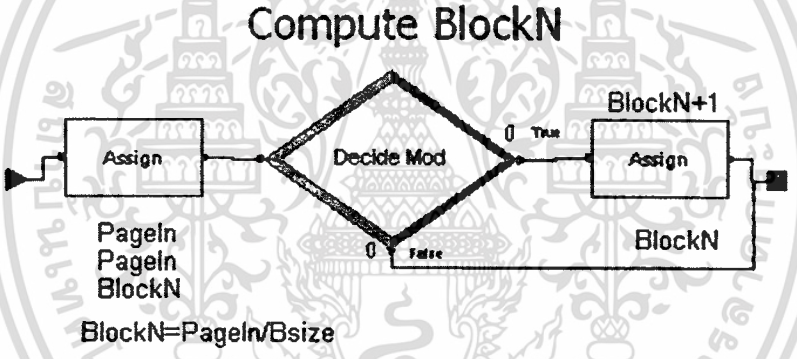


รูปที่ ค.4 ส่วนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.3 การทำงานของโมดูลและโมเดลย่อยในส่วนที่ 2

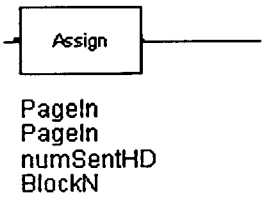
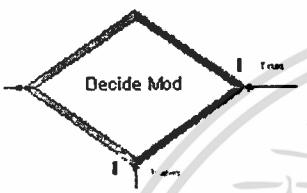
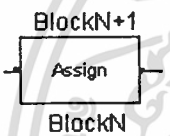
โมดูลและโมเดลย่อย	การทำงาน
	ทำการสุ่มหมายเลขเพจที่โปรแกรมต้องการใช้ในการประมวลผลและคำนวณว่าเพจดังกล่าวอยู่ในบล็อกใด รายละเอียดภายในแสดงดังรูปที่ ค.5 และตารางที่ ค.4
	ตรวจสอบที่อยู่ของเพจที่โปรแกรมต้องการว่าอยู่ที่หน่วยความจำหลักของเครื่องผู้ขอใช้บริการหรือฮาร์ดดิสก์หรือหน่วยความจำหลักของเครื่องผู้ให้บริการ



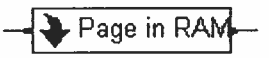
รูปที่ ค.5 ภายในโมเดลย่อย Compute BlockN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.4 การทำงานของโมดูลในโมเดลย่อย Compute BlockN

โมดูล	การทำงาน
	-กำหนดค่าตัวแปร PageIn =หมายเลขเพจที่โปรแกรมต้องการที่ได้จากการสุ่มด้วยฟังก์ชันทางคณิตศาสตร์ UNIF (1,PGsize) ซึ่งจะได้ค่าออกมาเป็นเลขทศนิยมจึงต้องทำให้เป็นเลขจำนวนเต็ม PageIn =PageIn-AMOD(PageIn,1) -คำนวณว่าเพจที่โปรแกรมต้องการอยู่ที่บล็อกใด BlockN=(PageIn/Bsize)
	ตัดสินใจว่าตัวแปร BlockN ต้องบวก 1 หรือไม่ ซึ่งจะต้องบวก 1 ในกรณีที่ PageIn หารด้วย Bsize ไม่ลงตัว
	กำหนดค่าตัวแปร BlockN = BlockN +1

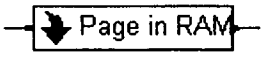
ส่วนที่ 3 ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ เมื่อเอ็นติตีเข้ามายังส่วนนี้จึงทำการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ เนื่องจากเพจที่โปรแกรมต้องการอยู่ในหน่วยความจำหลักเรียบร้อยแล้ว และลดจำนวนการอ้างถึงเพจที่โปรแกรมต้องการใช้ในการประมวลผลจนเสร็จ ซึ่งส่วนที่ 3 นี้ประกอบด้วยโมเดลย่อยแสดงดังรูปที่ ค.6 การทำงานของโมเดลย่อยแสดงดังตารางที่ ค.5

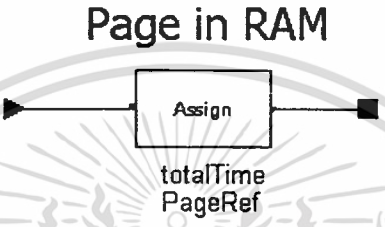


รูปที่ ค.6 ส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.5 การทำงานของในโมเดลย่อยในส่วนที่ 3

โมเดล	การทำงาน
	คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลัก รายละเอียดภายในแสดงดังรูปที่ ข.7และตารางที่ ข.6



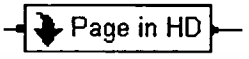
รูปที่ ค.7 ภายในโมเดลย่อย Page in RAM

ตารางที่ ค.6 การทำงานของโมดูลในโมเดลย่อย Page in RAM

โมดูล	การทำงาน
 totalTime PageRef	-คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ totalTime = totalTime +ma -ลดจำนวนครั้งของการอ้างถึงเพจลง 1 PageRef =PageRef-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 4 ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในฮาร์ดดิสก์ เมื่อเอ็นดิตีเข้ามาที่ส่วนนี้จะทำการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จและลดจำนวนการอ้างถึงเพจที่โปรแกรมต้องการใช้ในการประมวลผลจนเสร็จ ซึ่งส่วนที่ 4 นี้ประกอบด้วยโมเดลย่อยแสดงดังรูปที่ ค.8 การทำงานของโมเดลย่อยแสดงดังตารางที่ ค.7



รูปที่ ค.8 ส่วนที่ 4

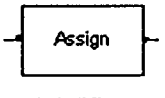
ตารางที่ ค.7 การทำงานของโมเดลย่อยในส่วนที่ 4

โมเดลย่อย	การทำงาน
	คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ เมื่อเพจที่โปรแกรมต้องการอยู่ในฮาร์ดดิสก์ รายละเอียดภายในแสดงดังรูปที่ ค.9 และตารางที่ ค.8

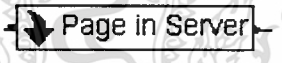
รูปที่ ค.9 ภายในโมเดลย่อย Page in HD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.8 การทำงานของโมดูลในโมเดลย่อย Page in HD

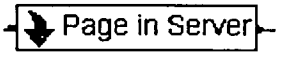
โมดูล	การทำงาน
 totalTime totalTime PageRef	-คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่ รวมกับเวลาเฉลี่ยที่ใช้ในการนำบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลจากฮาร์ดดิสก์เข้าสู่หน่วยความจำหลัก totalTime = totalTime + ma + HDAccess -ลดจำนวนครั้งของการอ้างถึงเพจ PageRef =PageRef-1

ส่วนที่ 5 ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อเพจที่โปรแกรมใช้ในการประมวลผลอยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ เมื่อเอ็นติตีเข้ามาที่ส่วนนี้จะทำการคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จและลดจำนวนการอ้างถึงเพจที่โปรแกรมต้องการใช้ในการประมวลผลจนเสร็จ ซึ่งส่วนที่ 5 นี้ประกอบด้วยโมเดลย่อยแสดงดังรูปที่ ค.10 การทำงานของโมเดลย่อยแสดงดังตารางที่ ค.9

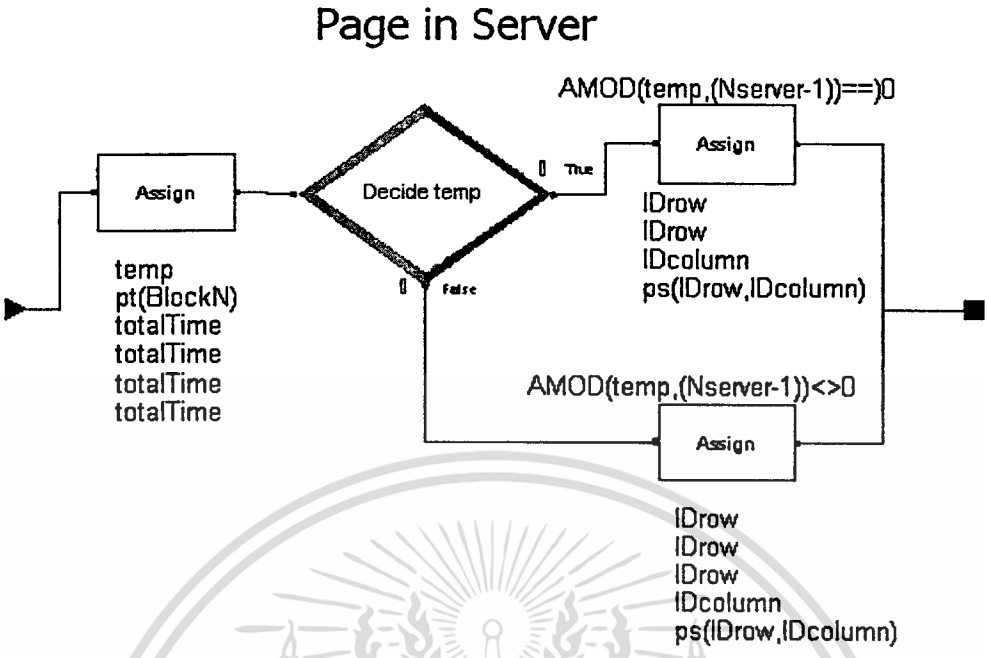


รูปที่ ค.10 ส่วนที่ 5

ตารางที่ ค.9 การทำงานของโมเดลย่อยในส่วนที่ 5

โมเดลย่อย	การทำงาน
	คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ เมื่อเพจที่โปรแกรมต้องการอยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ รายละเอียดภายในแสดงดังรูปที่ ค.11 และตารางที่ ค.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



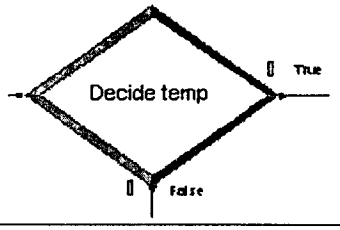
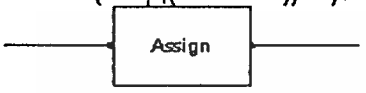
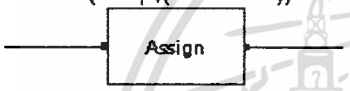
รูปที่ ค.11 ภายในโมเดลย่อย Page in Server

ตารางที่ ค.10 การทำงานของโมดูลในโมเดลย่อย Page in Server

โมดูล	การทำงาน
temp pt(BlockN) totalTime totalTime totalTime totalTime	-กำหนดค่าตัวแปร temp เพื่อคำนวณหาที่อยู่ของบล็อกที่โปรแกรมต้องการว่าอยู่ที่เครื่องผู้ให้บริการใด -เปลี่ยนสถานะที่อยู่ของบล็อก -คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลักเพื่อตรวจสอบว่าเพจที่โปรแกรมต้องการใช้ในการประมวลผลอยู่ในหน่วยความจำหลักหรือไม่+เวลาที่เครื่องผู้ขอใช้บริการส่งคำร้องขอเพจที่ต้องการไปยังเครื่องผู้ให้บริการที่เก็บข้อมูลของเพจที่ต้องการ+เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งบล็อกที่มีเพจที่โปรแกรมต้องการใช้ในการประมวลผลมายังเครื่องผู้ขอใช้บริการ+เวลาที่เครื่องผู้ให้บริการที่เก็บข้อมูลส่งคำร้องไปยังเครื่องผู้ให้บริการที่เก็บพารีตีเพจเพื่อแก้ไขสถานะภาพเป็นอินแวลิดเพจ totalTime = totalTime + (ma + (((512)/(BW*1048576))*2) + (131584/(BW*1048576)))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใ[illegible]ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.10 (ต่อ)

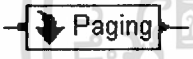
โมดูล	การทำงาน
	ตัดสินใจด้วยสมการ AMOD(temp,(Nserver-1))==0
AMOD(temp,(Nserver-1))==)0  IDrow IDrow IDcolumn ps(IDrow,IDcolumn)	เมื่อ AMOD(temp,(Nserver-1))==0 ทำการเปลี่ยนแปลงข้อมูลในตาราง ps ซึ่งเป็นตารางที่เก็บเลขเพจที่อยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ ps(IDrow,IDcolumn) =0 เพื่อแสดงว่าเพจถูกนำกลับมายังเครื่องผู้ขอใช้บริการแล้ว
AMOD(temp,(Nserver-1))<>0  IDrow IDrow IDrow IDcolumn ps(IDrow,IDcolumn)	เมื่อ AMOD(temp,(Nserver-1))<>0 ทำการเปลี่ยนแปลงข้อมูลในตาราง ps ซึ่งเป็นตารางที่เก็บเลขเพจที่อยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ ps(IDrow,IDcolumn) =0 เพื่อแสดงว่าเพจถูกนำกลับมายังเครื่องผู้ขอใช้บริการแล้ว

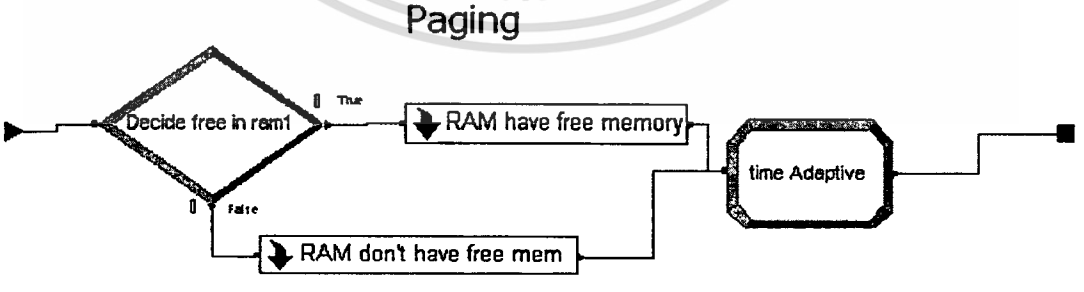
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 6 ส่วนคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลเมื่อทำอแคปทีฟพาริตีแคชชิง เมื่อเอ็นติตีเข้าสู่ส่วนนี้จะทำการนำบล็อกที่มีเพจที่โปรแกรมต้องการเข้าสู่หน่วยความจำหลัก ถ้าหน่วยความจำหลักมีที่ว่างไม่เพียงพอก็จะทำการเลือกเพจที่จะถูกแทนที่ จากนั้นนำเพจที่จะถูกแทนที่ส่งไปยังหน่วยความจำสำรองซึ่งได้แก่ หน่วยความจำหลักของเครื่องผู้ให้บริการหรือฮาร์ดดิสก์ ตามหลักการทำงานของอแคปทีฟพาริตีแคชชิงและคำนวณเวลาที่โปรแกรมใช้ในการประมวลผล ซึ่งส่วนที่ 6 นี้ประกอบด้วยโมเดลย่อยแสดงดังรูปที่ ค.12 การทำงานของโมดูลต่าง ๆ ในส่วนที่ 6 แสดงดังตารางที่ ค.11

รูปที่ ค.12 ส่วนที่ 6

ตารางที่ ค.11 การทำงานของโมเดลย่อยในส่วนที่ 6

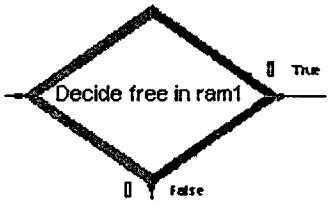
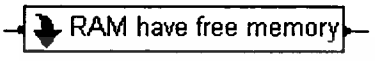
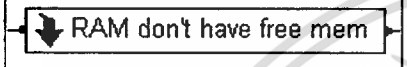
โมดูล	การทำงาน
	นำบล็อกที่มีเพจที่โปรแกรมต้องการเข้าสู่หน่วยความจำหลัก ถ้าหน่วยความจำหลักมีที่ว่างไม่เพียงพอก็จะทำการเลือกเพจที่จะถูกแทนที่ จากนั้นนำเพจที่จะถูกแทนที่ส่งไปยังหน่วยความจำสำรองตามหลักการทำงานของอแคปทีฟพาริตีแคช-ชิงและคำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ รายละเอียดภายในแสดงดังรูปที่ ค.13 และตารางที่ ค.12



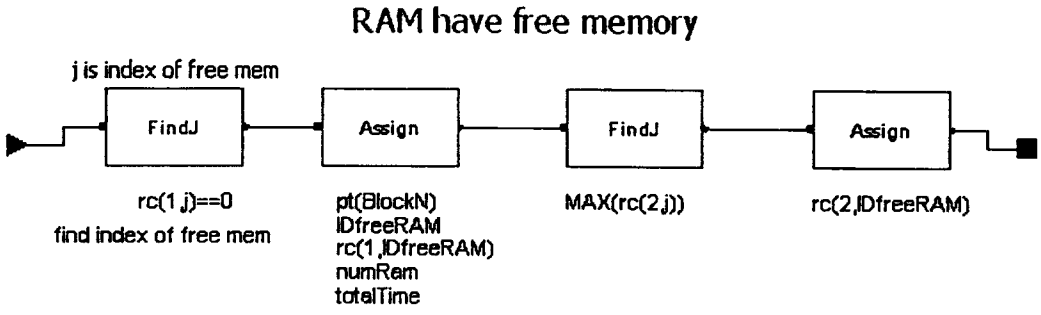
รูปที่ ค.13 ภายในโมเดลย่อย Paging

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.12 การทำงานของโมดูลในโมเดลย่อย Paging

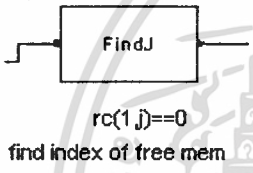
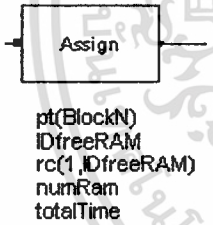
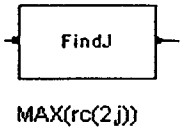
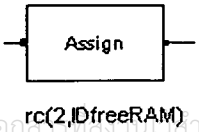
โมดูล	การทำงาน
	ตรวจสอบว่าหน่วยความจำหลักยังมีที่ว่างเพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการหรือไม่
	ส่วนที่จัดการเมื่อหน่วยความจำหลักมีพื้นที่ว่างเพียงพอสำหรับเพจที่โปรแกรมต้องการ รายละเอียดภายในแสดงดังรูปที่ ค.14 และตารางที่ ค.13
	ส่วนที่จัดการเมื่อหน่วยความจำหลักมีพื้นที่ว่างไม่เพียงพอสำหรับบล็อกที่มีเพจที่โปรแกรมต้องการ รายละเอียดภายในแสดงดังรูปที่ ค.15 และตารางที่ ค.14
	คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จเมื่อทำอแดปทีฟพาริตีแคชชิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

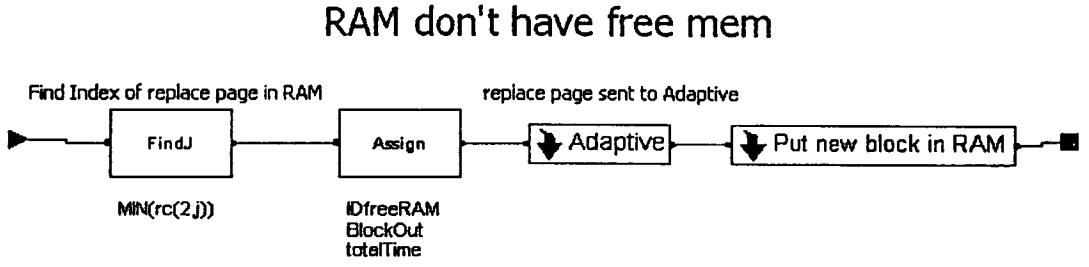


รูปที่ ค.14 ภายในโมเดลย่อย RAM have free memory

ตารางที่ ค.13 การทำงานของโมดูลในโมเดลย่อย RAM have free memory

โมดูล	การทำงาน
	ค้นหาตำแหน่งที่ว่างในหน่วยความจำหลัก
	-เปลี่ยนที่อยู่ของบล็อกที่มีเพจที่โปรแกรมต้องการอยู่ในหน่วยความจำหลัก pt(BlockN) =2 เพื่อแสดงว่าอยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ -กำหนดค่าตัวแปร IDfreeRAM = j -ย้ายบล็อกที่มีเพจที่โปรแกรมต้องการมาอยู่ในหน่วยความจำหลัก rc(1,IDfreeRAM) = BlockN -เพิ่มจำนวนบล็อกที่อยู่ในหน่วยความจำหลัก numRam = numRam+1 -คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ totalTime = totalTime+ma
	ค้นหาลำดับการเข้ามาหน่วยความจำหลักที่มากที่สุด
	กำหนดลำดับการเข้ามาหน่วยความจำหลักให้กับบล็อกที่เข้ามาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

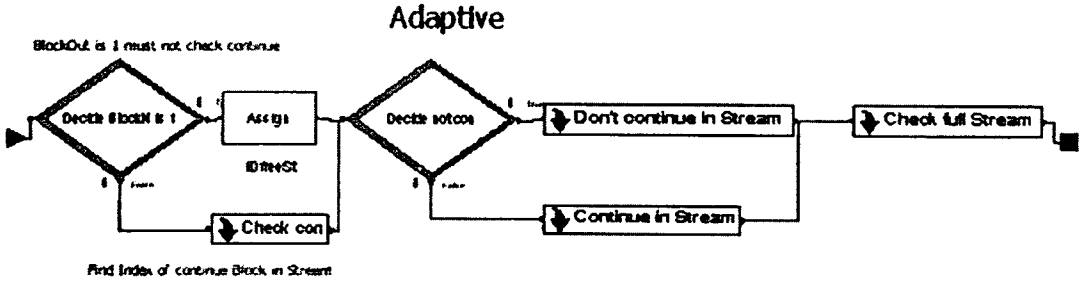


รูปที่ ค.15 ภายในโมเดลย่อย RAM don't have free mem

ตารางที่ ค.14 การทำงานของโมดูลในโมเดลย่อย RAM don't have free mem

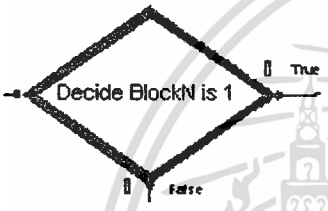
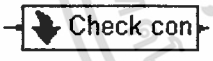
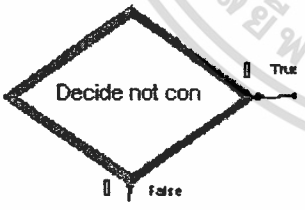
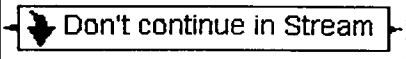
โมดูล	การทำงาน
Find Index of replace page in RAM FindJ MIN(rc(2,j))	ค้นหาบล็อกที่จะถูกแทนที่ซึ่งก็คือบล็อกที่เข้ามาก่อนหรือมีลำดับการเข้ามาน้อยสุด
Assign IDfreeRAM BlockOut totalTime	-กำหนดค่าตัวแปร IDfreeRAM = j คือตำแหน่งที่หน่วยความจำหลักจะว่าง -กำหนดค่าตัวแปร BlockOut เท่ากับเลขบล็อกที่จะถูกแทนที่ -คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำหลัก เพื่อทำการเลือกเพจที่จะถูกแทนที่ totalTime = totalTime+ma
Adaptive	เป็นส่วนที่จัดการกับบล็อกที่จะถูกแทนที่ตามหลักอแดปทีฟพาริตีแคชชิง รายละเอียดภายในแสดงดังรูปที่ ค.16 และตารางที่ ค.15
Put new block in RAM	เป็นส่วนที่จัดการวางบล็อกที่มีเพจที่โปรแกรมต้องการในหน่วยความจำหลัก รายละเอียดภายในแสดงดังรูปที่ ค.26 และตารางที่ ค.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.16 ภายในโมเดลย่อย Adaptive

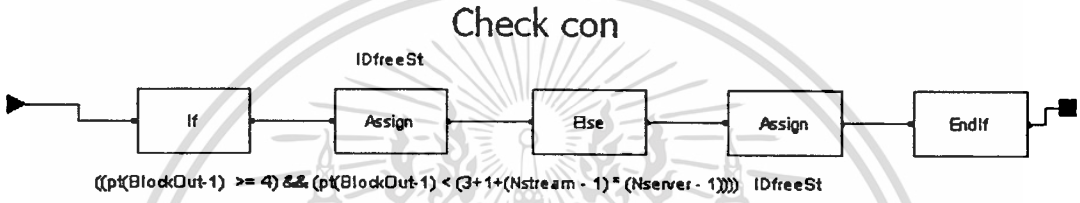
ตารางที่ ค.15 การทำงานของโมดูลและโมเดลย่อยในโมเดลย่อย Adaptive

โมดูลและโมเดลย่อย	การทำงาน
	ตรวจสอบว่าบล็อกที่จะถูกแทนที่เป็นบล็อกที่ 1 ของโปรแกรมหรือไม่
	เมื่อบล็อกที่จะถูกแทนที่เป็นบล็อกที่ 1 ของโปรแกรมจึงไม่ต้องตรวจสอบว่าต่อเนื่องในสตรีมแคชหรือวิกทิมแคช กำหนด IDfreeSt เป็น 0 เพื่อบ่งบอกว่าไม่ต่อเนื่องในแคช
	ตรวจสอบว่าบล็อกที่จะถูกแทนที่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคชใด รายละเอียดภายในแสดงดังรูปที่ ค.17 และตารางที่ ค.16
	ตรวจสอบว่า IDfreeSt ==0 หรือไม่ ถ้าเท่ากับ 0 แสดงว่าบล็อกที่จะถูกแทนที่ไม่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคช
	ดำเนินการในกรณีที่บล็อกจะถูกแทนที่ไม่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคช รายละเอียดภายในแสดงดังรูปที่ ค.18 และตารางที่ ค.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.15 (ต่อ)

โมดูล	การทำงาน
Continue in Stream	ดำเนินการในกรณีที่บล็อกที่มีเพจที่โปรแกรมต้องการต่อเนื่องในสตรีมแคชหรือวิกทิมแคช รายละเอียดภายในแสดงดังรูปที่ ค.20 และตารางที่ ค.19
Check full Stream	ดำเนินการในกรณีสตรีมแคชหรือวิกทิมแคชเต็ม รายละเอียดภายในแสดงดังรูปที่ ค.21 และตารางที่ ค.20

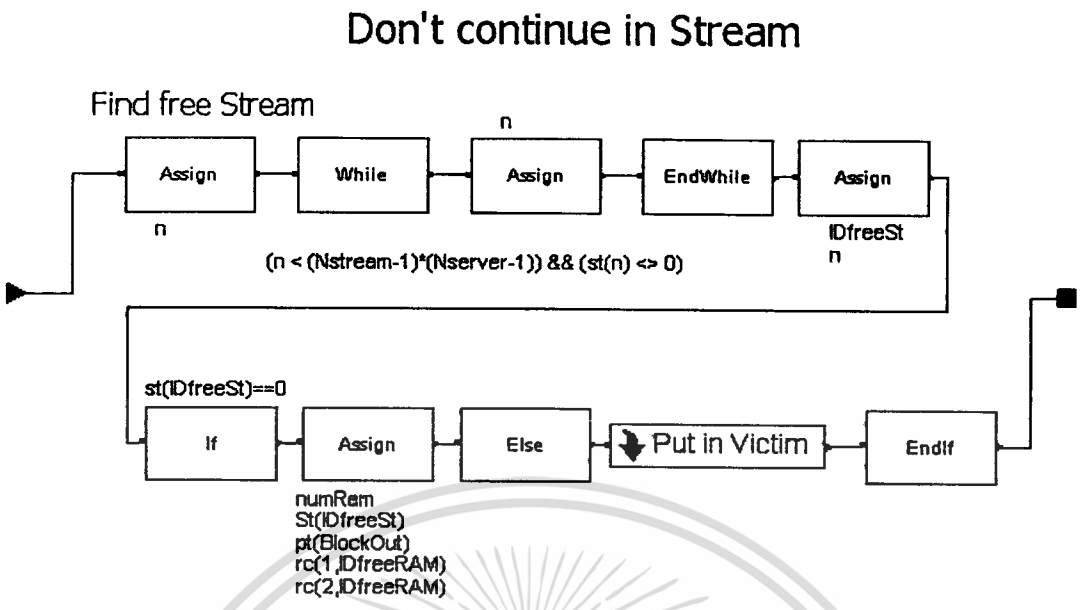


รูปที่ ค.17 ภายในโมเดลย่อย Check con

ตารางที่ ค.16 การทำงานของโมดูลในโมเดลย่อย Check con

โมดูล	การทำงาน
If	เงื่อนไขว่าบล็อกที่จะถูกแทนที่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคช
IDfreeSt Assign	กรณีที่บล็อกที่จะถูกแทนที่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคชจึงกำหนดตำแหน่งที่จะวางบล็อกดังกล่าวลงในสตรีมแคชหรือวิกทิมแคชนั้น
Else	เงื่อนไขว่าบล็อกที่จะถูกแทนที่ไม่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคช
Assign IDfreeSt	กรณีที่บล็อกที่จะถูกแทนที่ไม่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคชจึงกำหนดตำแหน่งที่จะวางเท่ากับศูนย์เพื่อแสดงว่าไม่ต่อเนื่องในสตรีมแคชหรือวิกทิมแคช
EndIf	สิ้นสุดเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



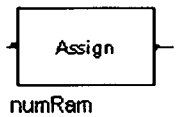
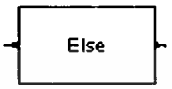
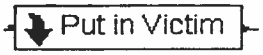
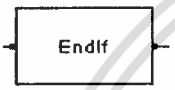
รูปที่ ค.18 ภายในโมเดลย่อย Don't continue in Stream

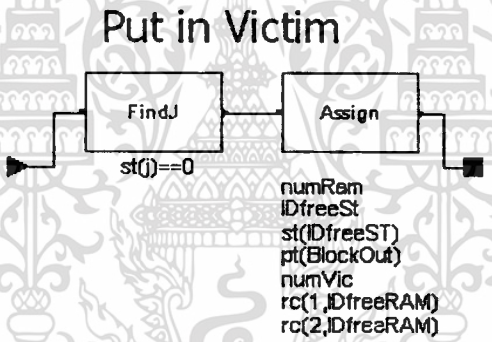
ตารางที่ ค.17 การทำงานของโมดูลและโมเดลย่อยในโมเดลย่อย Don't continue in Stream

โมดูลและโมเดลย่อย	การทำงาน
Assign n	กำหนดค่าตัวแปร n เป็นค่าเริ่มต้นการค้นหาสตรีมแคชที่ยังว่าง
While	วนลูปเพื่อหาสตรีมแคชที่บัฟเฟอร์ทั้งหมดว่าง
n Assign	กำหนดค่าตัวแปร n ในการค้นหารอบต่อไป
EndWhile	สิ้นสุดการวนลูปเพื่อค้นหาสตรีมแคชที่ยังว่าง
Assign IDfreeSt n	กำหนดค่าตัวแปร IDfreeSt เท่ากับตำแหน่งสตรีมแคชที่ว่าง กำหนดค่าตัวแปร n เป็นค่าเริ่มต้น
st(IDfreeSt)==0 If	เงื่อนไขมีสตรีมแคชว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

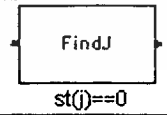
ตารางที่ ค.17 (ต่อ)

โมดูลและโมเดลย่อย	การทำงาน
	-ลดจำนวนบล็อกในหน่วยความจำหลัก -นำบล็อกที่จะถูกแทนที่ใส่ลงสตรีมแคชที่ว่าง -เปลี่ยนที่อยู่ของบล็อกที่จะถูกแทนที่ในตารางบล็อก -เปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก
	เงื่อนไขไม่มีสตรีมแคชว่าง
	วางบล็อกที่จะถูกแทนที่ลงวิกทิมแคช รายละเอียดภายในแสดงดังรูปที่ ค.19 และตารางที่ ค.18
	สิ้นสุดเงื่อนไข

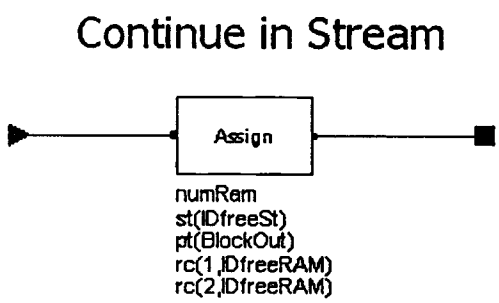


รูปที่ ค.19 ภายในโมเดลย่อย Put in Victim

ตารางที่ ค.18 การทำงานของโมดูลในโมเดลย่อย Put in Victim

โมดูล	การทำงาน
	ค้นหาที่ว่างในวิกทิมแคช
	-ลดจำนวนบล็อกในหน่วยความจำหลัก -กำหนดค่าตัวแปร IDfreeSt เท่ากับตำแหน่งว่างในวิกทิมแคช -นำบล็อกที่จะถูกแทนที่ลงในวิกทิมแคช -เปลี่ยนที่อยู่ของบล็อกที่จะถูกแทนที่ในตารางบล็อก -เพิ่มจำนวนบล็อกที่อยู่ในวิกทิกแคช -เปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก

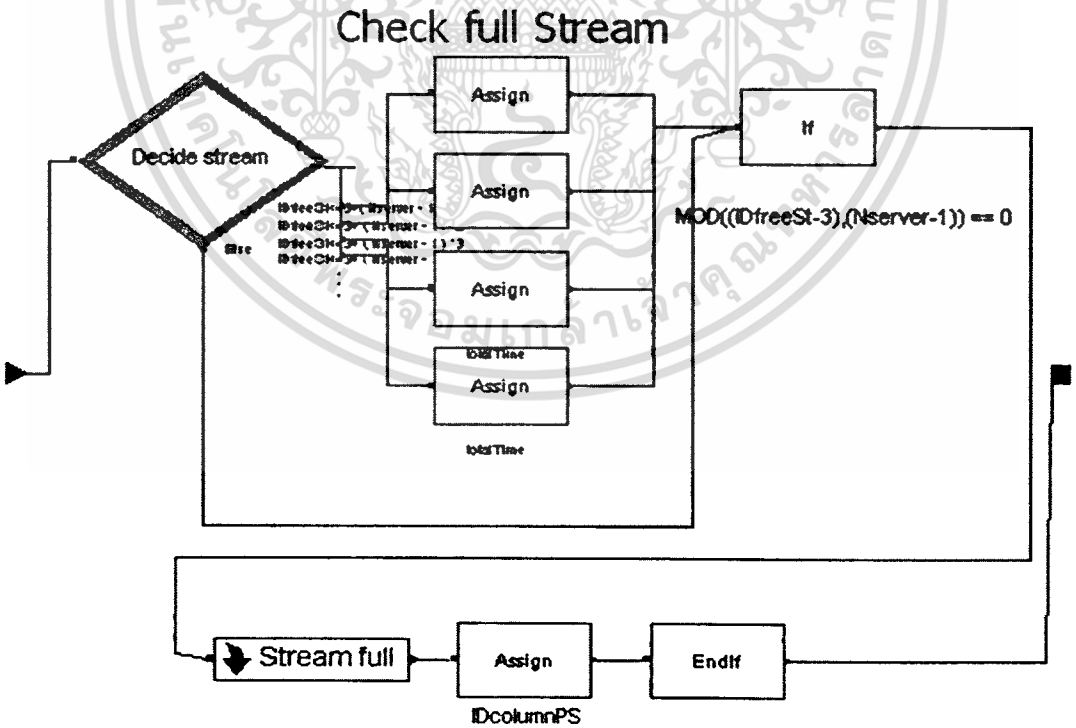
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.20 ภายในโมเดลย่อย Continue in Stream

ตารางที่ ค.19 การทำงานของโมดูลในโมเดลย่อย Continue in Stream

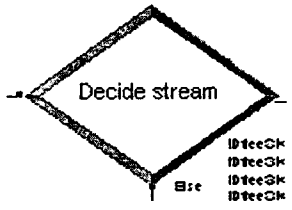
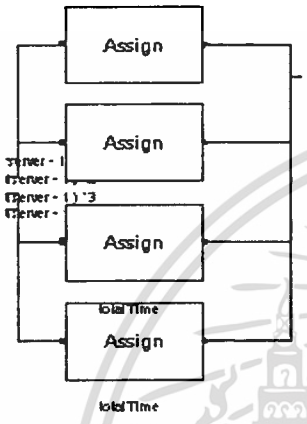
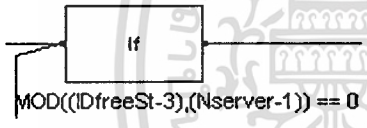
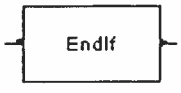
โมดูล	การทำงาน
Assign numRam st(IDfreeSt) pt(BlockOut) rc(1,IDfreeRAM) rc(2,IDfreeRAM)	-ลดจำนวนบล็อกในหน่วยความจำหลัก -นำบล็อกที่จะถูกแทนที่ใส่ลงสตรีมแคชที่ว่าง -เปลี่ยนที่อยู่ของบล็อกที่จะถูกแทนที่ในตารางบล็อก -เปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก



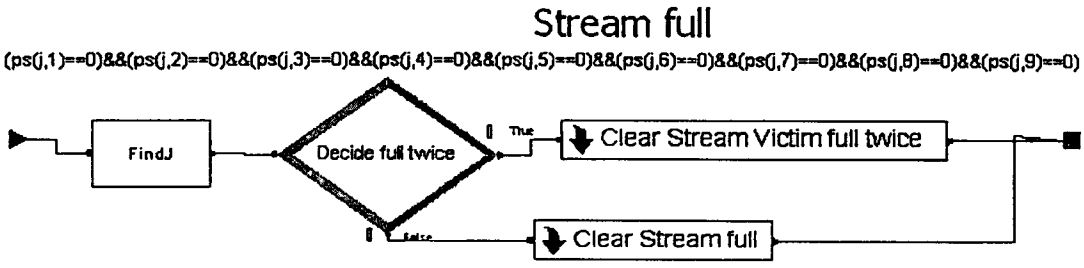
รูปที่ ค.21 ภายในโมเดลย่อย Check full Stream

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.20 การทำงานของโมดูลในโมเดลย่อย Check full Stream

โมดูล	การทำงาน
	ตรวจสอบว่านำบล็อกที่จะถูกแทนที่ใส่ลงสตรีมแคชหรือวิกทิมแคชใด
	คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จแบ่งเป็น 4 กรณี กรณีวางในสตรีมแคชที่ 1 totalTime=totalTime + ma กรณีวางในสตรีมแคชที่ 2 totalTime=totalTime + ma*2 กรณีวางในสตรีมแคชที่ 3 totalTime=totalTime + ma*3 กรณีวางในวิกทิมแคช totalTime=totalTime + ma*4
	เงื่อนไขเมื่อวางบล็อกที่จะถูกแทนที่แล้วทำให้สตรีมแคชหรือวิกทิมแคชเต็ม
	ดำเนินการในกรณีที่สตรีมแคชหรือวิกทิมแคชเต็ม รายละเอียดภายในแสดงดังรูปที่ ค.22 และตารางที่ ค.21
	กำหนดค่าเริ่มต้นให้กับตัวแปร IDcolumnPS
	สิ้นสุดเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

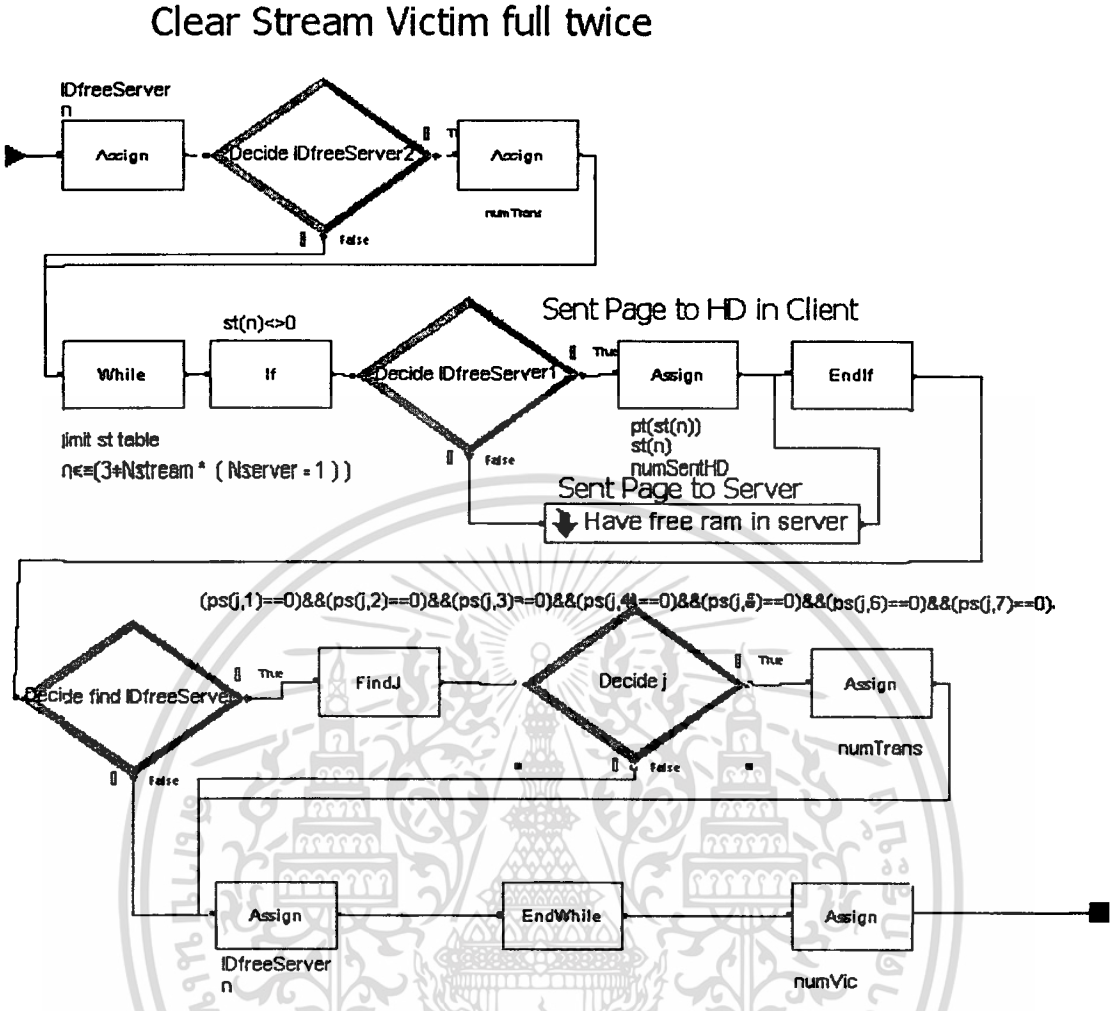


รูปที่ ค.22 ภายในโมเดลย่อย Stream full

ตารางที่ ค.21 การทำงานของโมดูลในโมเดลย่อย Stream full

โมดูล	การทำงาน
FindJ	หาที่ว่างในเครื่องผู้ให้บริการเพื่อจะนำข้อมูลในสตรีมแคช หรือวิกทิมแคชไปเก็บ
Decide full twice	ตรวจสอบว่าวิกทิมแคชเต็มเป็นครั้งที่ 2 หรือไม่
Clear Stream Victim full twice	ดำเนินการเมื่อวิกทิมแคชเต็มเป็นครั้งที่ 2 รายละเอียดภายในแสดงดังรูปที่ ค.23 และตารางที่ ค.22
Clear Stream full	ดำเนินการเมื่อสตรีมแคชหรือวิกทิมแคชเต็ม รายละเอียดภายในแสดงดังรูปที่ ค.25 และตารางที่ ค.24

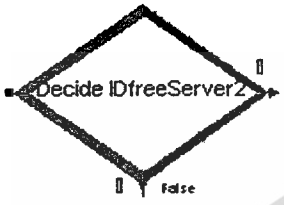
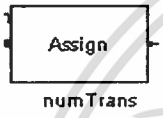
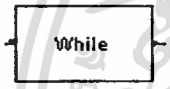
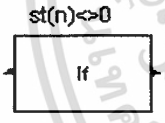
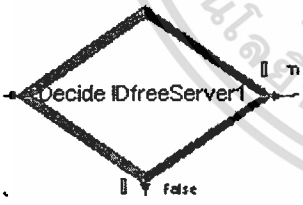
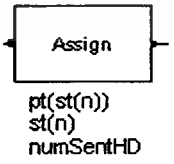
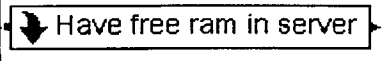
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.23 ภายในโมเดลย่อย Clear Stream Victim full twice

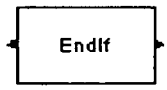
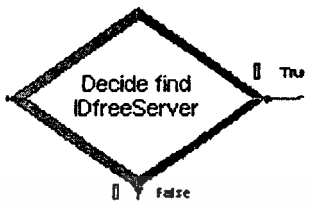
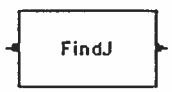
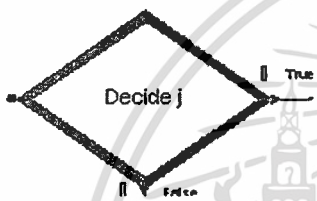
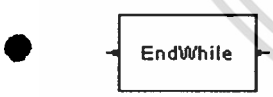
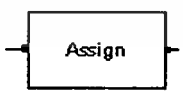
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.22 การทำงานของโมดูลในโมเดลย่อย Clear Stream Victim full twice

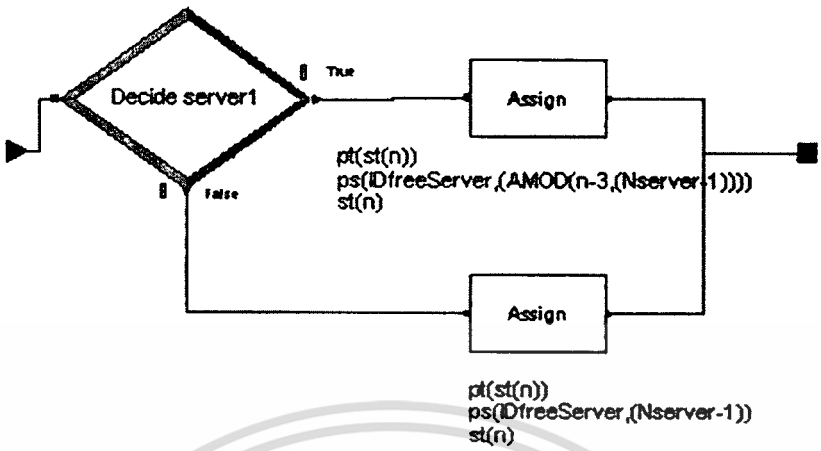
โมดูลและโมเดลย่อย	การทำงาน
	-กำหนดค่าตัวแปร IdfreeServer= jซึ่งเป็นตำแหน่งที่ว่างในเครื่องผู้ให้บริการ -กำหนดค่าเริ่มต้นตัวแปร n เพื่อการวนลูปส่งบล็อกในสตรีมแคชทั้งหมดและวิกทิมแคชไปเก็บยังหน่วยความจำสำรอง
	ตรวจสอบว่าเครื่องผู้ให้บริการมีที่ว่างสำหรับบล็อกในสตรีมแคชหรือวิกทิมแคชหรือไม่
	กรณีที่เครื่องผู้ให้บริการมีที่ว่างจะทำการเพิ่มจำนวนที่รับส่งข้อมูลผ่านเครือข่ายอีกเท่ากับจำนวนเครื่องผู้ให้บริการ
	การวนลูปเพื่อย้ายข้อมูลทั้งหมดในสตรีมแคชและวิกทิมแคชไปยังที่ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการหรือเครื่องผู้ให้บริการ
	เงื่อนไขถ้าข้อมูลในสตรีมแคชหรือวิกทิมแคชไม่เท่ากับ 0
	ตรวจสอบว่าจะส่งบล็อกไปยังที่ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการหรือเครื่องผู้ให้บริการ
	-เปลี่ยนที่อยู่ของบล็อกที่ถูกนำกลับฮาร์ดดิสก์ของเครื่องผู้ให้บริการในตารางบล็อก -เปลี่ยนแปลงข้อมูลในตารางเก็บเลขบล็อกในอแดปทีฟพาริตีแคช -เพิ่มจำนวนครั้งที่ส่งบล็อกกลับลงฮาร์ดดิสก์
	นำบล็อกที่อยู่ในสตรีมแคชส่งไปยังหน่วยความจำหลักของเครื่องผู้ให้บริการที่มีที่ว่างเพียงพอ รายละเอียดภายในแสดงดังรูปที่ ค.24 และตารางที่ ค.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.22 (ต่อ)

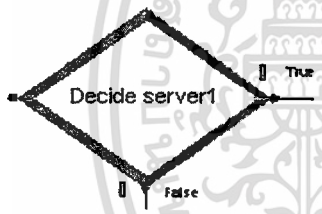
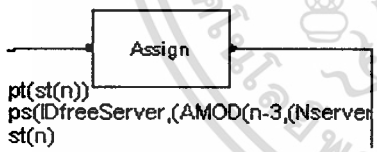
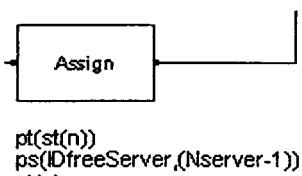
โมดูลและโมเดลย่อย	การทำงาน
	สิ้นสุดเงื่อนไขถ้าข้อมูลในสตรีมแคชหรือวิกทิมแคชไม่เท่ากับ 0
	ตรวจสอบว่าจะต้องทำการค้นหาที่ว่างที่เครื่องผู้ให้บริการอีกครั้งหรือไม่
	ค้นหาที่ว่างที่เครื่องผู้ให้บริการ
	ตัดสินใจว่าต้องเพิ่มจำนวนการส่งบล็อกผ่านเครือข่ายไปยังเครื่องผู้ให้บริการหรือไม่
	เพิ่มจำนวนการส่งบล็อกผ่านเครือข่ายไปยังเครื่องผู้ให้บริการ
	-กำหนดตำแหน่งที่ว่างในหน่วยความจำหลักที่เครื่องผู้ให้บริการ -เพิ่มค่าตัวแปร n เพื่อใช้ในการย้ายข้อมูล
	สิ้นสุดการวนลูปเพื่อย้ายข้อมูลทั้งหมดในสตรีมแคชและวิกทิมแคชไปยังที่ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการหรือเครื่องผู้ให้บริการ
	กำหนดค่าเริ่มต้นให้กับตัวแปร numVic เพื่อเริ่มนับจำนวนการเต็มของวิกทิมแคชใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

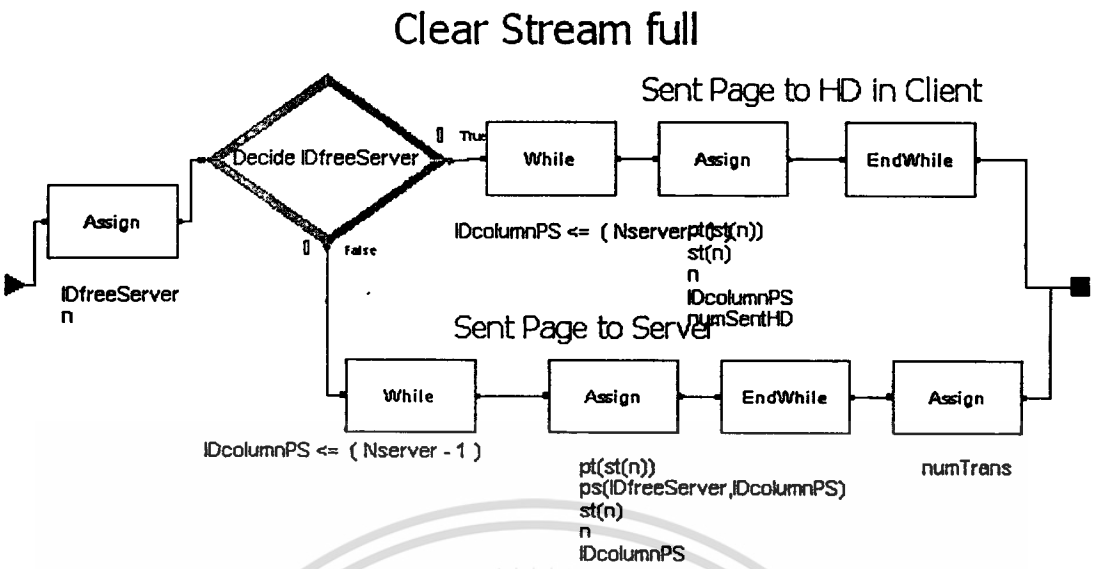


รูปที่ ค.24 ภายในโมเดลย่อย Have free ram in server

ตารางที่ ค.23 การทำงานของโมดูลในโมเดลย่อย Have free ram in server

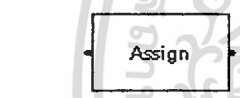
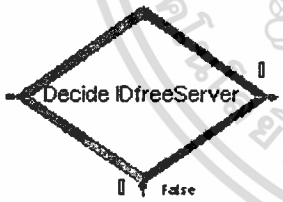
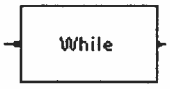
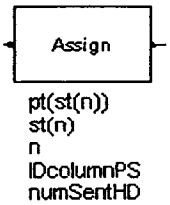
โมดูล	การทำงาน
	ตัดสินใจว่าไม่เป็นคอลัมน์สุดท้ายของตารางที่เก็บเลขบล็อกที่อยู่เครื่องผู้ให้บริการหรือไม่
	เมื่อไม่ใช่คอลัมน์สุดท้าย -เปลี่ยนที่อยู่ของบล็อกในตารางบล็อก -นำเลขบล็อกใส่ตารางที่เก็บเลขบล็อกที่อยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ -เปลี่ยนแปลงข้อมูลในตารางเก็บเลขบล็อกในอแดปทีฟพาริตีแคช
	เมื่อใช่คอลัมน์สุดท้าย -เปลี่ยนที่อยู่ของบล็อกในตารางบล็อก -นำเลขบล็อกใส่ตารางที่เก็บเลขบล็อกที่อยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ -เปลี่ยนแปลงข้อมูลในตารางเก็บเลขบล็อกในอแดปทีฟพาริตีแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



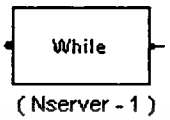
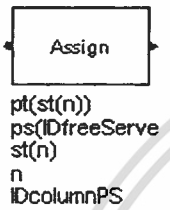
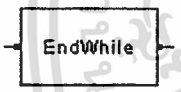
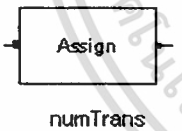
รูปที่ ค.25 ภายในโมเดลย่อย Clear stream full

ตารางที่ ค.24 การทำงานของโมดูลในโมเดลย่อย Clear stream full

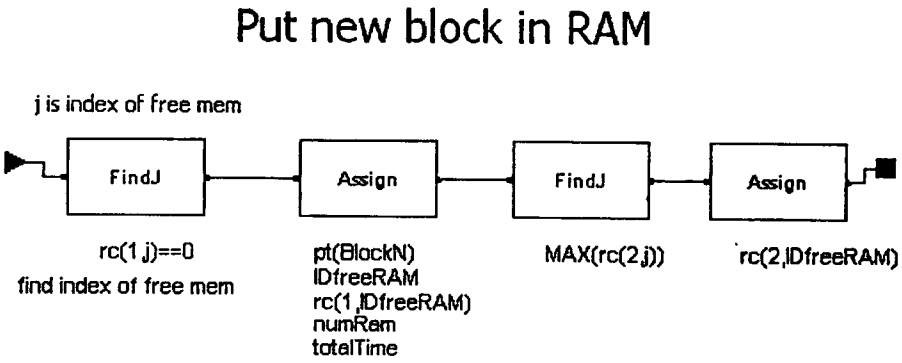
โมดูล	การทำงาน
	-กำหนดค่าตัวแปร IDfreeServer เท่ากับตำแหน่งที่ว่างในเครื่องผู้ให้บริการ -กำหนดค่าเริ่มต้นตัวแปร n เพื่อการวนลูปส่งบล็อกในสตรีมแคชทั้งหมดและวิกทิมแคชไปเก็บยังหน่วยความจำสำรอง
	ตรวจสอบว่าจะส่งบล็อกไปยังที่ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการหรือเครื่องผู้ให้บริการ
	การวนลูปเพื่อย้ายข้อมูลทั้งหมดในสตรีมแคชหรือวิกทิมแคชไปยังที่ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการ
	-เปลี่ยนที่อยู่ของบล็อกที่ถูกนำกลับฮาร์ดดิสก์ของเครื่องผู้ให้บริการในตารางบล็อก -เปลี่ยนแปลงข้อมูลในตารางเก็บเลขบล็อกในอแดปทีฟพาริตีแคช -เพิ่มค่าตัวแปร n และ IDcolumnPS -เพิ่มจำนวนครั้งที่ส่งบล็อกกลับลงฮาร์ดดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.24 (ต่อ)

โมดูล	การทำงาน
	สิ้นสุดการวนลูปเพื่อย้ายข้อมูลทั้งหมดในสตรีมแคชหรือวิกทิมแคชไปยังที่ฮาร์ดดิสก์ของเครื่องผู้ขอใช้บริการ
	การวนลูปเพื่อย้ายข้อมูลทั้งหมดในสตรีมแคชหรือวิกทิมแคชไปยังหน่วยความจำหลักที่เครื่องผู้ให้บริการ
	-เปลี่ยนที่อยู่ของบล็อกที่ถูกนำกลับฮาร์ดดิสก์ของเครื่องผู้ให้บริการในตารางบล็อก -นำเลขบล็อกใส่ตารางที่เก็บเลขบล็อกที่อยู่ในหน่วยความจำหลักของเครื่องผู้ให้บริการ -เปลี่ยนแปลงข้อมูลในตารางเก็บเลขบล็อกในอแดปทีฟพาร์ติชัน -เพิ่มค่าตัวแปร n และ IDcolumnPS
	สิ้นสุดการวนลูปเพื่อย้ายข้อมูลทั้งหมดในสตรีมแคชหรือวิกทิมแคชไปยังหน่วยความจำหลักที่เครื่องผู้ให้บริการ
	เพิ่มจำนวนการรับส่งข้อมูลผ่านเครือข่ายเพื่อนำบล็อกไปวางยังหน่วยความจำหลักของเครื่องผู้ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



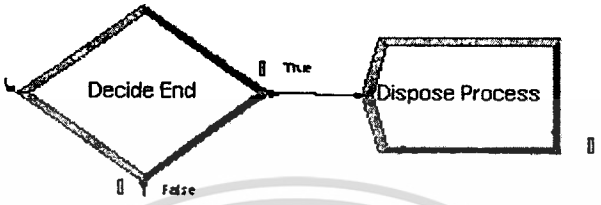
รูปที่ ค.26 ภายในโมเดลย่อย Put new block in RAM

ตารางที่ ค.25 การทำงานของโมดูลในโมเดลย่อย Put new block in RAM

โมดูล	การทำงาน
FindJ rc(1,j)==0	ค้นที่ตำแหน่งที่ว่างในหน่วยความจำหลักของเครื่องผู้ให้บริการ
Assign pt(BlockN) IDfreeRAM rc(1,IDfreeRAM) numRam totalTime	-เปลี่ยนที่อยู่ของบล็อกที่ถูกนำกลับฮาร์ดดิสก์ของเครื่องผู้ให้บริการในตารางบล็อก pt(BlockN)=2 -กำหนดค่าตัวแปร IDfreeRAM=j -เปลี่ยนแปลงตารางที่เก็บบล็อกที่อยู่ในหน่วยความจำหลักที่เครื่องผู้ขอใช้บริการ rc(1,IDfreeRAM)=BlockN -เพิ่มจำนวนบล็อกที่อยู่ในหน่วยความจำหลักของเครื่องผู้ขอใช้บริการ numRam = numRam+1 -คำนวณเวลาที่โปรแกรมใช้ในการประมวลผลจนเสร็จ = เวลาเฉลี่ยที่ใช้ในการเข้าถึงเพจในหน่วยความจำหลักที่โปรแกรมต้องการ totalTime = totalTime+ma
FindJ MAX(rc(2,j))	ค้นหาลำดับการเข้ามายังหน่วยความจำหลักที่มากที่สุด
Assign rc(2,IDfreeRAM)	กำหนดลำดับการเข้ามาของบล็อกที่มีเพจที่โปรแกรมต้องการ

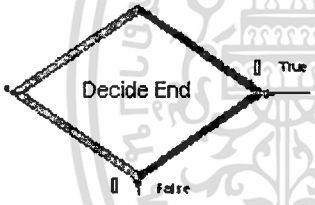
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 7 ส่วนตัดสินใจว่าสิ้นสุดการทำงาน จะเป็นส่วนที่ตรวจสอบว่ายังต้องการเพจในการประมวลผลอีกหรือไม่ ถ้าใช่จะทำส่วนที่ 2 อีกครั้ง แต่ถ้าไม่ต้องการเพจแล้วก็จะถือว่าสิ้นสุดการทำงานและแสดงผลลัพธ์ ซึ่งส่วนที่ 7 ประกอบด้วยโมดูลแสดงดังรูปที่ ค.27 การทำงานของโมดูลในส่วนที่ 7 แสดงดังตารางที่ ค.26



รูปที่ ค.27 ส่วนที่ 7

ตารางที่ ค.26 การทำงานของโมดูลในส่วนที่ 7

โมดูล	การทำงาน
	ตัดสินใจว่าสิ้นสุดการทำงานหรือไม่ ซึ่งตรวจสอบจาก PageRef == 0 หรือไม่ถ้าใช่ถือว่าสิ้นสุดการทำงาน เอ็นติตีจะถูกทำลาย ถ้าไม่ใช่จะนำเอ็นติตีกลับไปยังส่วนที่ 2 อีกครั้ง
	สิ้นสุดการทำงานทำลายเอ็นติตี และแสดงผลลัพธ์ที่ได้จากการทดลองซึ่งก็คือ เวลาที่โปรแกรมให้ในการประมวลผลจนเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ – สกุล	นางสาวธีราภรณ์ วิศิษฐวาณิชย์
วัน เดือน ปีเกิด	23 กรกฎาคม 2523 ที่สิงห์บุรี
ที่อยู่	61/16 หมู่ 5 หมู่บ้านเสนานิเวศน์ โครงการ 2
	แขวงจรเข้บัว เขตลาดพร้าว กรุงเทพฯ 10230 โทร. 0-2578-3546
ประวัติการศึกษา	2544 วิทยาศาสตรบัณฑิต สาขาวิชาสถิติประยุกต์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประวัติการทำงาน	
ตุลาคม 2547	วิทยากร อบรม ภาษาซี โครงการคอมพิวเตอร์โอลิมปิก ค่าย 1
	ประจำปี 2547 ณ โรงเรียนสามเสนวิทยาลัย
มีนาคม – เมษายน 2544	ฝึกงานที่ บริษัท ยูโนแคล ไทยแลนด์ จำกัด
	ณ อาคาร 3 ไทยพาณิชย์ปาร์คพลาซ่า
ทุนการศึกษา	
2542-2543	ทุนอุดหนุนการศึกษา (งบประมาณ)
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2544	ทุน บริษัท สหวนิชอินดัสตรี จำกัด
2445	ทุนเรียนดีแต่ขาดแคลน
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2546-2547	ทุนมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้