การออกแบบเว็บเซิร์ฟเวอร์แบบกระจายโดยการใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา

นางสาวนุจรีย์ ศรีเพ็ชรดานนท์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2545

ISBN 974-17-1208-1

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

THE DESIGN OF DISTRIBUTED WEB SERVERS USING DYNAMIC LOAD BALANCING AND CONTENT-BASED ROUTING

Miss Nujaree Sripetchdanon

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 2002
ISBN 974-17-1208-1

หัวข้อวิทยานิพนธ์	การออกแบบเว็บเซิร์ฟเวอร์แบบกระจายโดยการใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหา
โดย	นางสาวนุจรีย์ ศรีเพ็ชรดานนท์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

..... กรรมการ
(อาจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... กรรมการ
(อาจารย์ ดร.วีระ เหมืองสิน)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นุจรีย์ ศรีเพ็ชรดานนท์ : การออกแบบเว็บเซิร์ฟเวอร์แบบกระจายโดยการใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา. (THE DESIGN OF DISTRIBUTED WEB SERVERS USING DYNAMIC LOAD BANLANCING AND CONTENT-BASED ROUTING) อ.ที่ปรึกษา : อ.ดร.ณัฐวุฒิ หนูไพโรจน์, 73 หน้า. ISBN 974-17-1208-1.

ปัจจุบัน ความนิยมในการใช้อินเทอร์เน็ตเพิ่มขึ้นอย่างรวดเร็ว การใช้เว็บเซิร์ฟเวอร์ตัวเดียวอาจจะไม่สามารถรองรับการร้องขอที่เข้ามาพร้อมกันจำนวนมากได้ ดังนั้นจึงมีการใช้เว็บเซิร์ฟเวอร์หลายตัวในการให้บริการ เพื่อเพิ่มความสามารถในการทำงาน และเสถียรภาพของระบบ แต่วิธีการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส ซึ่งเป็นวิธีที่นิยมใช้กันมากนั้นยังไม่สามารถแก้ปัญหาได้เต็มที่ เนื่องจากไคลเอนต์จากเครือข่ายเดียวกัน จะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ตัวเดียวกัน ซึ่งอาจทำให้เว็บเซิร์ฟเวอร์บางตัวมีสภาพภาระงานมาก จนไม่สามารถให้บริการการร้องขอได้อีก

ในวิทยานิพนธ์นี้ได้เสนอวิธีการกระจายการร้องขอแบบใหม่ โดยแต่ละเว็บเซิร์ฟเวอร์จะเป็นผู้กระจายการร้องขอเอง โดยพิจารณาจากเนื้อหาของการร้องขอ และสภาพภาระงานของเว็บเซิร์ฟเวอร์เพื่อส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่มีข้อมูลเอกสารนั้นอยู่ในบัฟเฟอร์ ซึ่งจะช่วยลดเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ อันส่งผลให้เวลาการรอคอยเอกสารของไคลเอนต์ลดลง วิทยานิพนธ์นี้ได้ทำการทดสอบระบบที่ออกแบบขึ้นเปรียบเทียบกับระบบที่กระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส ซึ่งการทดสอบจะใช้วิธีการจำลองการทำงานของระบบด้วยโปรแกรมจำลองเอนเอสทู โดยใช้ข้อมูลการใช้เว็บของ www.pantip.com และ www.thaisecondhand.com จากผลการทดสอบพบว่าวิธีการกระจายการร้องขอที่เสนอไว้ในวิทยานิพนธ์นี้ช่วยเพิ่มฮิตเรตให้กับเว็บเซิร์ฟเวอร์ ลดเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ ลดเวลาตอบสนอง และเพิ่มความสามารถของเว็บเซิร์ฟเวอร์ในการบริการการร้องขอได้มากกว่าระบบที่ใช้การกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต .....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์	ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2545	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

4370367221 : MAJOR COMPUTER SCIENCE
KEY WORD: DISTRIBUTED WEB SERVERS / DYNAMIC LOAD BALANCING / CONTENT-BASED ROUTING

NUJAREE SRIPETCHDANON : THE DESIGN OF DISTRIBUTED WEB SERVERS USING DYNAMIC LOAD BALANCING AND CONTENT-BASED ROUTING, THESIS ADVISOR : DR. NATAWUT NUPAIROJ, 73 pp. ISBN 974-17-1208-1.

With the increasing popularity of the Internet, a web site with a single web server is inadequate for handling all of the incoming simultaneous requests. To overcome this problem, multiple web servers are used to improve performance and stability of the systems. However, the most popular request distribution, DNS Round Robin scheme, is not efficient enough to support a very large number of requests because clients from the same network will send requests to the same server. This may cause the server to be overloaded and no longer able to serve other requests.

In this thesis, we have studied a new request distribution scheme, which allows the web servers to use the content of the requests and the load information of the other web servers to redirect the requests to the web servers that have the requested document in the buffer. This will decrease disk access time and reduce clients' waiting time. We have used NS2 to simulate network environment for comparing the performance of the servers that use our scheme and the servers that use DNS Round Robin under usage patterns from the logs of www.pantip.com and www.thaisecondhand.com. The result of the simulation shows that our scheme can increase hit rate of web servers and decrease disk access time as well as response time. Moreover, it can improve web servers' capability to serve more requests than DNS Round Robin scheme.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Computer Engineering	Student's signature
Field of study Computer Science	Advisor's signature
Academic year 2002	Co-advisor's signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณา และความช่วยเหลืออย่างดียิ่ง จากอาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ รวมทั้งสนับสนุนการทำงานวิจัยของข้าพเจ้ามาด้วยดีตลอด

ขอขอบคุณผู้บริหารเว็บไซท์ www.pantip.com และ www.thaisecondhand.com ที่กรุณาให้ข้อมูลสำหรับใช้ทดสอบในวิทยานิพนธ์นี้

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ที่ได้ให้คำแนะนำ และกำลังใจแก่ข้าพเจ้าตลอดเวลาที่ศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์แห่งนี้

ท้ายที่สุด ข้าพเจ้าใคร่ขอกราบขอบพระคุณบิดา มารดา และทุกคนในครอบครัว ซึ่งสนับสนุนและให้กำลังใจแก่ข้าพเจ้าเสมอมาจนสำเร็จการศึกษา

นุจรีย์ ศรีเพ็ชรดานนท์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญภาพ ..... ญ

บทที่

1. บทนำ 1
    1.1 ความเป็นมาและความสำคัญของปัญหา 1
    1.2 วัตถุประสงค์ของงานวิจัย 5
    1.3 ขอบเขตของงานวิจัย 5
    1.4 ขั้นตอนและวิธีดำเนินงานวิจัย 6
    1.5 ประโยชน์ที่คาดว่าจะได้รับ 6
    1.6 โครงสร้างวิทยานิพนธ์ 6
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 7
    2.1 ทฤษฎีที่เกี่ยวข้อง 7
    2.2 งานวิจัยที่เกี่ยวข้อง 13
3. ระบบเว็บเซิร์ฟเวอร์โดยใช้การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา 17
    3.1 การทำงานของระบบ 17
4. การทดสอบและผลการทดสอบ 23
    4.1 การจำลองระบบเพื่อใช้ในการทดสอบ 23
    4.2 เครื่องมือที่ใช้ในการทดสอบ 29
    4.3 สมมติฐานที่ใช้ในการทดสอบ 30
    4.4 ข้อมูลที่ใช้ในการทดสอบ 31
    4.5 สภาพแวดล้อมที่ใช้ในการทดสอบ 32
    4.6 วิธีการทดสอบ 32

สารบัญ (ต่อ)

หน้า

4.7 ผลการทดสอบ 33
5. การวิเคราะห์ผลการทดลอง 54
6. สรุปผลการวิจัยและข้อเสนอแนะ 58
6.1 สรุปผลการวิจัย 58
6.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย 58
6.3 ข้อเสนอแนะ 59
รายการอ้างอิง 60
ภาคผนวก 62
ภาคผนวก ก สรุปคำสั่ง และตัวอย่างสคริปต์ไฟล์ 63
ภาคผนวก ข การแก้ไขโปรแกรมจำลองเอนเอสทู 69
ประวัติผู้เขียนวิทยานิพนธ์ 73

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 2.1 รหัสเลขฐาน 16 ของวิธีการร้องขอ 9
ตารางที่ 3.1 ตัวอย่างการเข้ารหัสเอกสาร 20
ตารางที่ 4.1 ลักษณะข้อมูลการใช้เว็บ 32
ตารางที่ 5.1 ปัจจัยที่มีผลต่อการให้บริการ..... 57
ตารางที่ ข.1 เปรียบเทียบโปรแกรมจำลองเอนเอสทูกับความต้องการของวิทยานิพนธ์ 70

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 1.1 การกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส 2
รูปที่ 1.2 ตัวกลางกระจายการร้องขอโดยพิจารณาจากเนื้อหาของการร้องขอ..... 4
รูปที่ 2.1 HTTP Redirect 7
รูปที่ 2.2 การแทนค่าในข้อมูลสรุปบัฟเฟอร์..... 10
รูปที่ 2.3 การกระจายการร้องโดยพิจารณาจากเนื้อหาของการร้องขอ 11
รูปที่ 2.4 การเปลี่ยนแปลงสถานะของเซิร์ฟเวอร์..... 12
รูปที่ 3.1 การทำงานและการให้บริการการร้องขอของเซิร์ฟเวอร์..... 17
รูปที่ 3.2 การแลกเปลี่ยนข้อมูลรายชื่อเอกสารที่มีในบัฟเฟอร์ 19
รูปที่ 3.3 การแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์..... 19
รูปที่ 3.4 การตรวจสอบสภาพภาระงานเพื่อแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบ..... 21
รูปที่ 4.1 โครงสร้างของระบบ..... 23
รูปที่ 4.2 ขั้นตอนการทำงานของโปรแกรมจำลอง..... 25
รูปที่ 4.3 ขั้นตอนการกระจายการร้องขอ 27
รูปที่ 4.4 เปอร์เซ็นต์ซ้ำเมื่อคำนวณข้อมูลสรุปบัฟเฟอร์..... 31
รูปที่ 4.5 เวลาตอบสนอง เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์ 35
รูปที่ 4.6 ฮิตเรต เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์ 35
รูปที่ 4.7 เปอร์เซ็นต์รีไดเร็กท์ เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์ 36
รูปที่ 4.8 โอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์..... 36
รูปที่ 4.9 จำนวนการร้องขอของเว็บเซิร์ฟเวอร์ต่อวินาที 37
รูปที่ 4.10 เปอร์เซ็นต์การใช้งานของเว็บเซิร์ฟเวอร์..... 38
รูปที่ 4.11 เวลาตอบสนอง เมื่อเพิ่มจำนวนไคลเอนต์ 39
รูปที่ 4.12 เปอร์เซ็นต์รีไดเร็กท์ เมื่อเพิ่มจำนวนไคลเอนต์ 40
รูปที่ 4.13 โอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อเพิ่มจำนวนไคลเอนต์ 41
รูปที่ 4.14 เวลาตอบสนอง เมื่อเพิ่มขนาดบัฟเฟอร์..... 42
รูปที่ 4.15 เปอร์เซ็นต์รีไดเร็กท์ เมื่อเพิ่มขนาดบัฟเฟอร์ 43
รูปที่ 4.16 โอกาสพบเอกสารจากการ รีไดเร็กท์เมื่อเพิ่มขนาดบัฟเฟอร์..... 43
รูปที่ 4.17 เปอร์เซ็นต์รีไดเร็กท์ เมื่อปรับช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ 44

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 4.18 โอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อปรับช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ 45
รูปที่ 4.19 เวลาตอบสนอง เมื่อปรับช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ 45
รูปที่ 4.20 ปริมาณการร้องขอ เมื่อใช้ดีเลย์ 500 มิลลิวินาที 46
รูปที่ 4.21 เวลาตอบสนอง เมื่อใช้ดีเลย์ 500 มิลลิวินาที 47
รูปที่ 4.22 เปอร์เซ็นต์รีไดเร็กท์ เมื่อใช้ดีเลย์ 500 มิลลิวินาที 48
รูปที่ 4.23 โอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อใช้ดีเลย์ 500 มิลลิวินาที 49
รูปที่ 4.24 การไม่รับการร้องขอ เมื่อใช้ดีเลย์ 500 มิลลิวินาที 49
รูปที่ 4.25 เปอร์เซ็นต์การใช้งานของเว็บเซิร์ฟเวอร์ เมื่อกำหนดดีเลย์ 500 มิลลิวินาที 50
รูปที่ 4.26 เวลาตอบสนอง เมื่อใช้ข้อมูลการใช้เว็บของ www.thaisecondhand.com 51
รูปที่ 4.27 เปอร์เซ็นต์รีไดเร็กท์ เมื่อใช้ข้อมูลของ www.thaisecondhand.com 52
รูปที่ 4.28 แสดงโอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อใช้ข้อมูลของ www.thaisecondhand.com 52

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

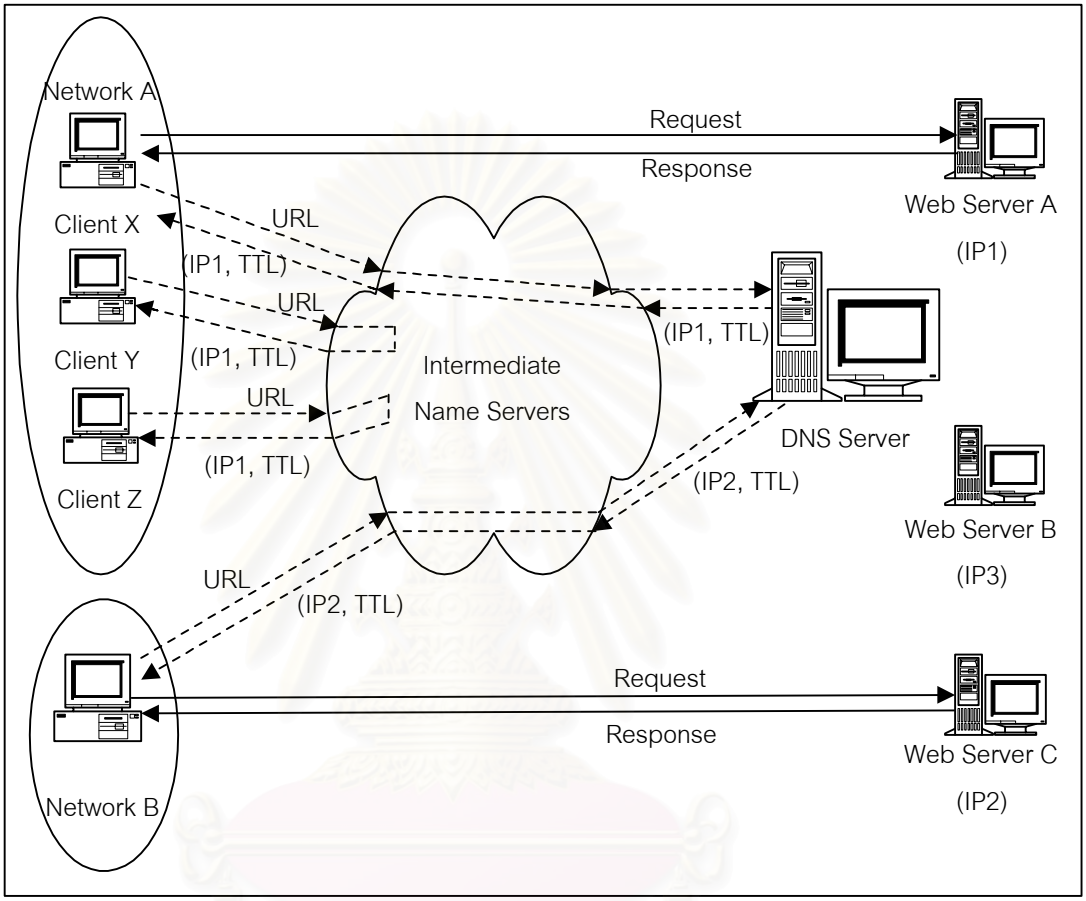
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในระยะเวลาที่ผ่านมาหลายปีนี้ ความนิยมในการใช้อินเทอร์เน็ตเพิ่มขึ้นอย่างรวดเร็ว ทำให้ปริมาณข้อมูลที่ส่งผ่านในเครือข่ายอินเทอร์เน็ตเพิ่มขึ้นเกินกว่าที่จะสามารถรองรับได้ เช่น www.yahoo.com ในเดือนกรกฎาคม 2001 มีจำนวนเอกสารที่มีผู้ร้องขอเฉลี่ยประมาณ 1.2 พันล้านหน้าต่อวัน [17] ดังนั้น ถ้ามีเว็บเซิร์ฟเวอร์เพียงตัวเดียวทำหน้าที่ให้บริการการร้องขอของไคลเอนต์ จะส่งผลให้ความสามารถในการทำงานของระบบ และเสถียรภาพของระบบลดลง ต่อมาจึงเกิดความคิดที่จะใช้เว็บเซิร์ฟเวอร์หลาย ๆ ตัวเพื่อให้บริการที่เหมือนกันแทนการใช้เว็บเซิร์ฟเวอร์เพียงตัวเดียว จึงมีผู้เสนอเทคนิคที่ใช้ในการกระจายการร้องขอของไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์แต่ละตัว โดยแบ่งตามเอนทิตีที่ทำหน้าที่ในการกระจายการร้องขอดังนี้ คือ

1. Client-based [10,13] คือ ไคลเอนต์สามารถตัดสินใจเลือกได้ว่าจะใช้บริการจากเว็บเซิร์ฟเวอร์ใด ข้อเสียของวิธีนี้ คือ ไคลเอนต์ต้องมีความสามารถที่จะเลือกเว็บเซิร์ฟเวอร์ได้ และต้องรู้ด้วยว่าเว็บไซท์นั้นใช้เว็บเซิร์ฟเวอร์หลายตัวในการให้บริการการร้องขอของไคลเอนต์
2. DNS-based [4,10,13] คือ ดีเอนเอสทำหน้าที่ในการกระจายการร้องขอของไคลเอนต์ โดยเมื่อไคลเอนต์ต้องการติดต่อไปยังเว็บเซิร์ฟเวอร์ก็จะต้องติดต่อมายังดีเอนเอส แล้วดีเอนเอสก็จะกระจายการร้องขอโดยการตอบกลับเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ที่จะให้ไคลเอนต์ติดต่อได้ ซึ่งวิธีนี้จะมีข้อเสียคือ ไคลเอนต์จากเครือข่ายเดียวกันก็จะติดต่อไปยังเว็บเซิร์ฟเวอร์เดียวกัน และหากมีการร้องขอส่งไปยังเว็บเซิร์ฟเวอร์นั้นพร้อม ๆ กันจำนวนมาก ก็จะส่งผลให้เว็บเซิร์ฟเวอร์นั้นมีสภาพภาระงานมากได้ ดังรูปที่ 1.1 คือ เมื่อไคลเอนต์ X ต้องการร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์ จะต้องทำการเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพี โดยถามไปยังดีเอนเอสที่ใกล้ที่สุด ถ้าดีเอนเอสนั้นไม่รู้จักยูอาร์แอลนี้ ยังไม่มีข้อมูลเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ก็จะถามไปยังดีเอนเอสถัดไป จนกระทั่งถึงดีเอนเอสปลายทาง เมื่อดีเอนเอสปลายทางได้รับจึงตอบกลับเป็นเลขที่อยู่ไอพีหนึ่งของเว็บเซิร์ฟเวอร์ตัวหนึ่งในเว็บเซิร์ฟเวอร์ทั้งหมดโดยใช้วิธีวนรอบ ในที่นี้ดีเอนเอสเลือกเว็บเซิร์ฟเวอร์ A ดีเอนเอสจึงตอบกลับมาเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ A ดีเอนเอสที่ถามไปก็จะจดจำไว้ว่ายูอาร์แอลนี้คือเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ A จนกว่าจะหมดช่วงอายุทีทีแอล และดีเอน

เอสจะตอบกลับเป็นเลขที่อยู่ไอพีนี้ส่งกลับมายังดีเอนเอสที่ถามไป จนกระทั่งถึงดีเอนเอสแรก ดีเอนเอสก็จะตอบไคลเอนต์ X เป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ A โดยไคลเอนต์จะจดจำเลขที่อยู่ไอพีนี้ไว้ และทุกการร้องขอของไคลเอนต์ X จะส่งไปยังเลขที่อยู่ไอพีนี้จนกระทั่งหมดช่วงอายุทีทีแอลจึงจะถามไปยังดีเอนเอสอีกครั้ง



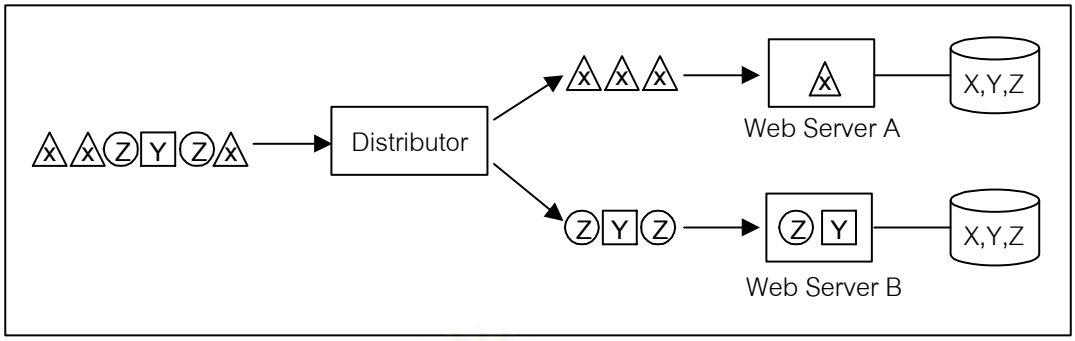
รูปที่ 1.1 การกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส

และเมื่อไคลเอนต์ Y ซึ่งอยู่ในเครือข่ายเดียวกันต้องการร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์ ไคลเอนต์ก็จะต้องทำการเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพี โดยถามไปยังดีเอนเอสที่ใกล้ที่สุดซึ่งก็คือดีเอนเอสตัวเดียวกับที่ไคลเอนต์ X ติดต่อไป เมื่อดีเอนเอสได้รับการร้องขอแล้วทำการตรวจสอบพบว่าเคยมีคนทำการเปลี่ยนยูอาร์แอลนี้เป็นเลขที่อยู่ไอพีแล้ว ซึ่งก็คือเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ A ดีเอนเอสก็จะนำข้อมูลที่มีอยู่ตอบกลับให้ไคลเอนต์ Y ดังนั้นไคลเอนต์ Y ก็จะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ A และทุกการร้องขอของไคลเอนต์ Y ก็จะส่งไปยังเว็บเซิร์ฟเวอร์ A จนกว่าจะหมดช่วงอายุทีทีแอล นั่นแสดงให้เห็นว่าทุกการร้องขอของไคลเอนต์ในเครือข่ายเดียวกัน ก็จะ

ส่งไปยังเว็บเซิร์ฟเวอร์ตัวเดียวกัน ซึ่งถ้าหากว่าการร้องขอจากเครือข่ายนี้มีจำนวนมาก ก็จะส่งผลกระทบให้เว็บเซิร์ฟเวอร์นี้อาจมีสภาพภาระงานมากเกินไป จนไม่สามารถให้บริการการร้องขอได้อีก

3. Dispatcher-based [10,13] คือ ดิสแพทเชอร์ (Dispatcher) ซึ่งเป็นอุปกรณ์ตัวกลาง ทำหน้าที่ในการรับการร้องขอจากไคลเอนต์ และกระจายการร้องขอโดยการตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่จะให้บริการการร้องขอนั้น แล้วส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่เลือก นั่นคือทุกครั้งที่ไคลเอนต์ต้องการติดต่อกับเว็บเซิร์ฟเวอร์ก็จะต้องส่งการร้องขอผ่านดิสแพทเชอร์ ทำให้เกิดสภาวะคอขวดที่ดิสแพทเชอร์ และหากไคลเอนต์ไม่สามารถติดต่อดิสแพทเชอร์ได้ ก็หมายความว่าไคลเอนต์จะไม่สามารถใช้บริการจากเว็บเซิร์ฟเวอร์ได้
4. Server-based [8,10,13] คือ เว็บเซิร์ฟเวอร์แต่ละตัวสามารถตัดสินใจได้ว่าจะให้บริการการร้องขอนั้นเอง หรือจะกระจายการร้องขอไปให้เว็บเซิร์ฟเวอร์ตัวอื่น โดยการพิจารณาจากสภาพภาระงานของเว็บเซิร์ฟเวอร์ ซึ่งแต่ละเว็บเซิร์ฟเวอร์จะต้องแลกเปลี่ยนข้อมูลสภาพภาระงานระหว่างกันทุกช่วงเวลาที่กำหนด การแลกเปลี่ยนข้อมูลสภาพภาระงานนี้ทำให้เกิดข้อเสีย คือ ทำให้เว็บเซิร์ฟเวอร์มีสภาพภาระงานเพิ่มขึ้น

อย่างไรก็ตาม วิธีการแก้ปัญหาข้างต้นก็ยังไม่สามารถรองรับการร้องขอของไคลเอนต์ที่เข้ามาพร้อม ๆ กันจำนวนมากได้ เนื่องมาจากการเกิดสภาวะคอขวดที่เว็บเซิร์ฟเวอร์หรือดิสแพทเชอร์ จึงไม่สามารถให้บริการการร้องขอของไคลเอนต์ได้อย่างรวดเร็ว ต่อมา จึงเกิดความคิดขึ้นมาว่ากรณีที่ไคลเอนต์ส่งการร้องขอที่มีเนื้อหาเหมือนกันหรือเอกสารเดียวกัน ก็ควรจะส่งไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน [15] เพราะเมื่อเว็บเซิร์ฟเวอร์เคยประมวลผลการร้องขอนั้น ถ้ามีการร้องขอเข้ามาและสามารถพบข้อมูลในบัฟเฟอร์แล้ว เว็บเซิร์ฟเวอร์ก็จะสามารถตอบการร้องขอนั้นได้ทันที จากงานวิจัยในปัจจุบันนั้น [3,9,12] พบว่า ได้มีการออกแบบให้มีการกระจายการร้องขอที่มีเนื้อหาเหมือนกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน โดยมีตัวกระจาย (Distributor) ซึ่งเป็นเซิร์ฟเวอร์พิเศษทำหน้าที่ตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่จะให้บริการการร้องขอของไคลเอนต์ โดยการพิจารณาจากเนื้อหาของการร้องขอและสภาพภาระงานของเว็บเซิร์ฟเวอร์ ดังรูปที่ 1.2



รูปที่ 1.2 ตัวกลางกระจายการร้องขอโดยพิจารณาจากเนื้อหาของการร้องขอ

จากรูปที่ 1.2 มีเว็บเซิร์ฟเวอร์ A และ B มี D เป็นตัวกระจาย และมีการร้องขอ X,Y และ Z เมื่อไคลเอนต์ส่งการร้องขอเข้ามา ทุกการร้องขอจะส่งไปที่ D ก่อนเพื่อตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่จะเป็นผู้ให้บริการโดย D จะส่งการร้องขอที่มีเนื้อหาเหมือนกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน ดังนั้นทุกการร้องขอ X จะส่งไปที่เว็บเซิร์ฟเวอร์ A และการร้องขอ Y และ Z จะส่งไปยังเว็บเซิร์ฟเวอร์ B จากการศึกษาพบว่า การใช้ตัวกลางเพื่อทำหน้าที่ในการตัดสินใจนั้น จะทำให้เกิดปัญหาตามมา คือ เมื่อมีการร้องขอเข้ามาจำนวนมากก็จะทำให้เกิดสภาวะคอขวดที่ตัวกลางได้ นอกจากนี้ยังทำให้เกิดความเสียหายที่เกิดจากจุด ๆ เดียว (Single Point of Failure) คือ ถ้าตัวกลางนี้ไม่สามารถทำงานได้ เช่น เครื่องค้าง (Hang) ทรัพยากรของเครื่องไม่เพียงพอ ฮาร์ดแวร์เสีย หรือกระบวนการกระจายการร้องขอหยุดทำงาน หากมีการร้องขอจากไคลเอนต์เข้ามา ก็จะไม่มีผู้ทำหน้าที่ในการตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่จะให้บริการ ทำให้ทั้งระบบไม่สามารถให้บริการแก่ไคลเอนต์ได้

ในงานวิจัยนี้จะทำการแก้ปัญหาการมีตัวกลางที่ทำหน้าที่ในการตัดสินใจ โดยจะกระจายการตัดสินใจให้เว็บเซิร์ฟเวอร์สามารถตัดสินใจเลือกได้ว่า แต่ละการร้องขอนั้นควรจะส่งให้เว็บเซิร์ฟเวอร์ใดเป็นผู้ให้บริการ โดยการพิจารณาจากเนื้อหาของการร้องขอและสภาพภาระงานของเว็บเซิร์ฟเวอร์ คือ จะส่งการร้องขอที่มีเนื้อหาเหมือนกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน และหากเว็บเซิร์ฟเวอร์ที่ควรจะเป็นผู้ให้บริการนั้นมีสภาพภาระงานมาก ก็จะส่งการร้องขอไปให้เว็บเซิร์ฟเวอร์ที่มีสภาพภาระงานน้อยกว่าเป็นผู้ให้บริการแทน ซึ่งเป็นวิธีการของ การสมดุลภาระงานแบบกระจาย (Distributed Load Balancing) ซึ่งงานวิจัยนี้จะให้ประโยชน์คือ เว็บเซิร์ฟเวอร์แต่ละตัวนั้นสามารถตัดสินใจได้เองว่าควรจะกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์ใด ไม่ต้องมีเซิร์ฟเวอร์ตัวกลางทำหน้าที่ตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่จะเป็นผู้ให้บริการ นอกจากนี้ สิ่งที่ต้องคำนึงถึงคือ ข้อมูลที่แลกเปลี่ยนกันระหว่างเว็บเซิร์ฟเวอร์ ได้แก่ ข้อมูลการร้องขอที่มีอยู่ในบัฟเฟอร์

และข้อมูลสภาพภาระงานของแต่ละเว็บเซิร์ฟเวอร์ ซึ่งข้อมูลทั้งสองนี้ควรมีขนาดเล็ก เพราะถ้าข้อมูลมีขนาดใหญ่ และเว็บเซิร์ฟเวอร์มีสภาพภาระงานมาก ก็จะยิ่งทำให้สภาพภาระงานของเว็บเซิร์ฟเวอร์เพิ่มขึ้น การให้บริการแก่ไคลเอนต์ก็จะช้าลง

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาความเป็นไปได้ในการกระจายการร้องขอ ไปยังเว็บเซิร์ฟเวอร์ต่าง ๆ โดยการพิจารณาจากเนื้อหาของการร้องขอและสภาพภาระงานของเว็บเซิร์ฟเวอร์ โดยกระจายการตัดสินใจให้เว็บเซิร์ฟเวอร์แต่ละตัวนั้นสามารถตัดสินใจได้ว่าควรจะให้บริการการร้องขอนั้นเอง หรือจะกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่นเป็นผู้ให้บริการ

1.3 ขอบเขตของการวิจัย

1. ทดสอบกระจายการร้องขอโดยวิธีการจำลอง (Simulation)
2. เว็บเซิร์ฟเวอร์จะกระจายการร้องขอเฉพาะกรณีเนื้อหาสถิต (Static Content)
3. ทุก ๆ เว็บเซิร์ฟเวอร์จะให้บริการข้อมูลที่เหมือน ๆ กัน
4. พารามิเตอร์ที่ปรับเปลี่ยนค่าได้ คือ จำนวนเว็บเซิร์ฟเวอร์ที่ให้บริการ, จำนวนไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน, ความถี่ในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์, ขนาดของบัฟเฟอร์
5. การวัดผลจะวัดเป็น
    - จำนวนการร้องขอที่เว็บเซิร์ฟเวอร์ให้บริการต่อวินาที (Server Throughput)
    - เวลาตอบสนอง (Response Time) เฉลี่ยของเว็บเซิร์ฟเวอร์
    - เปอร์เซ็นต์ของการรีไดเร็กท์ (Redirect)
    - เปอร์เซ็นต์การไม่รับการร้องขอ (Disconnection)
    - เปอร์เซ็นต์การใช้งาน (Utilization) ของเว็บเซิร์ฟเวอร์
6. การแทนที่ข้อมูล (Replacement Policy) ในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ จะใช้วิธีการแทนที่แบบแอลอาร์ยู (LRU: Least Recently Used)
7. ออกแบบกลยุทธ์ในการเลือกเว็บเซิร์ฟเวอร์ที่เหมาะสม
8. พิจารณาเฉพาะเวลาตอบสนองของเว็บเซิร์ฟเวอร์เท่านั้น
9. ทำการทดลองและเปรียบเทียบกับการสมดุลภาระงานแบบอื่น

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1. ศึกษาการทำงานด้วยวิธีวนรอบของดีเอนเอส
2. ศึกษาการทำงานของ HTTP Redirect
3. ศึกษาวิธีการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์และข้อมูลสภาพภาระงานระหว่างเว็บเซิร์ฟเวอร์
4. ศึกษาโปรแกรมจำลองเอนเอสทู (NS2: Network Simulations2)
5. พัฒนาโปรแกรมการกระจายการร้องขอ เพิ่มเติมในเอนเอสทู
6. ทดสอบกระจายการร้องขอโดยการปรับเปลี่ยนพารามิเตอร์ต่าง ๆ
7. วิเคราะห์และสรุปผล
8. จัดทำรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

นำเสนอแนวทางในการกระจายการร้องขอของไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์ โดยการพิจารณาจากเนื้อหาของการร้องขอและสภาพภาระงานของเว็บเซิร์ฟเวอร์ โดยเว็บเซิร์ฟเวอร์สามารถตัดสินใจได้เองว่าควรจะส่งการร้องไปให้เว็บเซิร์ฟเวอร์ใดเป็นผู้ให้บริการแก่การร้องขอนั้น ซึ่งสามารถนำไปประยุกต์ใช้ในงานจริงได้

1.6 โครงสร้างวิทยานิพนธ์

ในบทต่อไปของวิทยานิพนธ์นี้จะกล่าวถึงทฤษฎีที่นำมาประยุกต์ใช้ และงานวิจัยที่เกี่ยวข้อง ส่วนในบทที่ 3 จะกล่าวถึงแนวคิดของงานวิจัยนี้ ในบทที่ 4 จะกล่าวถึงระบบจำลองที่ได้สร้างขึ้นเพื่อใช้ในการทดสอบ และผลที่ได้จากการทดสอบระบบที่ได้ออกแบบขึ้น ในบทที่ 5 จะกล่าวถึงการวิเคราะห์ผลการทดลอง และในบทสุดท้ายจะเป็นการสรุปผลของงานวิทยานิพนธ์และข้อเสนอแนะในการพัฒนาต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

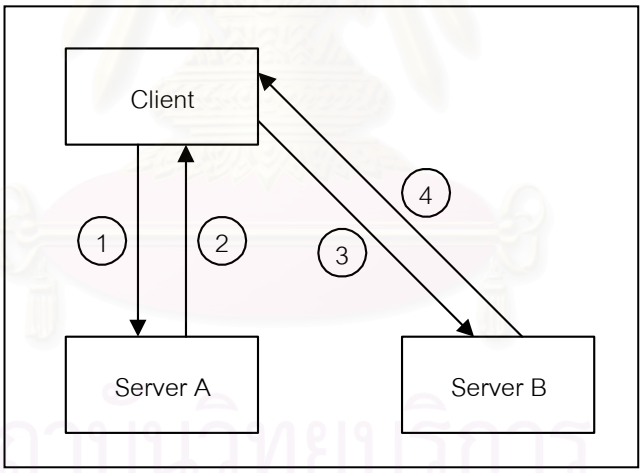
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 HTTP Redirect

เมื่อไคลเอนต์ส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ตัวหนึ่ง และเว็บเซิร์ฟเวอร์นั้นต้องการส่งการร้องขอนั้นไปยังเว็บเซิร์ฟเวอร์ตัวอื่น เว็บเซิร์ฟเวอร์ก็จะตอบกลับมาโดยมีรหัสตอบกลับ (Status Code) ในช่วง 300 – 399 เช่น 301 คือ เอกสารนั้นเปลี่ยนที่อยู่ ซึ่งชี้ให้เห็นว่าไคลเอนต์จะต้องติดต่อไปยังเว็บเซิร์ฟเวอร์อีกตัวหนึ่ง โดยจะระบุเว็บเซิร์ฟเวอร์ที่จะเป็นผู้ให้บริการไว้ในเฮดเดอร์ส่วนที่บอกตำแหน่ง (Location Header) นั่นคือ เมื่อไคลเอนต์พบว่ารหัสตอบกลับเป็น 301 แล้ว ไคลเอนต์ก็จะดูจากเฮดเดอร์ส่วนที่บอกตำแหน่ง และส่งการร้องขอนั้นไปไปยังยูอาร์แอลที่ระบุไว้ในเฮดเดอร์ส่วนที่บอกตำแหน่งนั้น



รูปที่ 2.1 HTTP Redirect

การรีไดเร็กท์ ประกอบด้วย 4 ขั้นตอน ดังรูปที่ 2.1

1. ไคลเอนต์ส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ A
2. เว็บเซิร์ฟเวอร์ A ส่งยูอาร์แอลของเว็บเซิร์ฟเวอร์ B กลับมายังไคลเอนต์
3. ไคลเอนต์ส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ B
4. เว็บเซิร์ฟเวอร์ B ตอบการร้องขอส่งกลับให้ไคลเอนต์

2.1.2 การสมดุลภาระงานแบบกระจาย

1. การกระจายภาระงานแบ่งได้เป็น
    - Static Load Distribution คือ การกระจายการร้องขอไปยังเซิร์ฟเวอร์ต่าง ๆ โดยไม่คำนึงถึงสภาพภาระงานในปัจจุบันของเซิร์ฟเวอร์ เช่น วิธีวนรอบ หรือวิธีสุ่ม
    - Dynamic Load Distribution คือ การกระจายการร้องขอไปยังเซิร์ฟเวอร์ต่าง ๆ โดยพิจารณาจากสภาพภาระงานของเซิร์ฟเวอร์
    - Adaptive Load Distribution คือ การกระจายการร้องขอ ซึ่งจะมีการปรับเปลี่ยนนโยบายในการกระจายการร้องขอโดยขึ้นอยู่กับสภาพภาระงานของระบบในขณะนั้น
2. นโยบายที่ใช้ในขั้นตอนวิธีกระจายภาระงาน
    - Location Policy คือ การเลือกว่าควรจะโอนงานไปยังเซิร์ฟเวอร์ตัวใด หรือรับงานจากเซิร์ฟเวอร์ตัวใด ซึ่งอาจจะใช้วิธีสุ่ม หรือเป็น State Polling
    - Information Policy คือ การพิจารณาว่าเมื่อไรควรจะมีการแลกเปลี่ยนข้อมูลสภาพภาระงานของเซิร์ฟเวอร์ ซึ่งอาจแลกเปลี่ยนข้อมูลกันแบบทุกช่วงเวลาที่กำหนด หรือตามความต้องการ
    - Transfer Policy คือ การพิจารณาว่าควรจะโอนงานให้เซิร์ฟเวอร์อื่นหรือไม่เมื่อไรควรจะโอนงาน
    - Selection Policy คือ การเลือกว่าจะโอนงานใดไปให้เซิร์ฟเวอร์ตัวอื่น

2.1.3 ข้อมูลสรุปบัฟเฟอร์ (Buffer Summarization)

ข้อมูลสรุปบัฟเฟอร์ คือ ข้อมูลผลสรุปของเนื้อหาของการร้องขอที่มีอยู่ในบัฟเฟอร์ของแต่ละเซิร์ฟเวอร์โดยจะจัดเก็บในรูปแบบอะเรย์ของบิต แต่ละเซิร์ฟเวอร์จะนำข้อมูลเนื้อหาของการร้องขอที่มีอยู่ในบัฟเฟอร์มาสร้างเป็นข้อมูลสรุปบัฟเฟอร์ และแลกเปลี่ยนกันระหว่างเซิร์ฟเวอร์เพื่อใช้ในการตรวจสอบว่าเซิร์ฟเวอร์ใดน่าจะมีข้อมูลการร้องขอที่มีเนื้อหาตรงกับที่ต้องการ โดยอาศัยหลักการของ Summary Cache [7] และ Cache Digests [1]

1. การสร้างข้อมูลสรุปบัฟเฟอร์

ข้อมูลสรุปบัฟเฟอร์นี้จะเกิดจากการนำยูอาร์แอลและวิธีการร้องขอมาเข้าฟังก์ชันแฮชชิงได้เป็นค่าดรรชนีบอกตำแหน่งของข้อมูลสรุปบัฟเฟอร์ที่จะต้องกำหนดค่าให้เป็น 1 โดยเริ่มแรกจะต้องกำหนดค่าเริ่มต้นในทุกบิตเป็น 0 ทั้งหมดก่อน จากนั้นจึงนำข้อมูลในบัฟเฟอร์มาสร้างเป็นข้อมูลสรุปบัฟเฟอร์ดังนี้ คือ

- นำวิธีการร้องขอและยูอาร์แอลมาแปลงเป็นรหัสเลขฐาน 16 ซึ่งวิธีการร้องขอต่าง ๆ จะกำหนดค่าดังตารางที่ 2.1

ตารางที่ 2.1 รหัสเลขฐาน 16 ของวิธีการร้องขอ

วิธีการร้องขอ	รหัสเลขฐาน 16
GET	01
POST	02
PUT	03
HEAD	04
CONNECT	05
TRACE	06
PURGE	07

ตัวอย่างเช่น GET http://www.w3.org

แปลงได้เป็น 01 68 74 3a 2f 2f 77 77 77 2e 77 33 2e 6f 72 67 2f

- คำนวณไดเจสท์ (Digest) โดยใช้ MD5

ได้เป็น e06a56257d8879d9e968e83f2ded3df7

- แบ่งไดเจสท์ที่ได้เป็น N ส่วน (ในที่นี้ N = 4)

Temp_Key[0] = 0xe06a5625

Temp_Key[1] = 0x7d8879d9

Temp_Key[2] = 0xe968e83f

Temp_Key[3] = 0x2ded3df7

- คำนวณค่าดรรชนีของข้อมูลสรุปบัฟเฟอร์ (ในที่นี้ Digest_Size = 128)

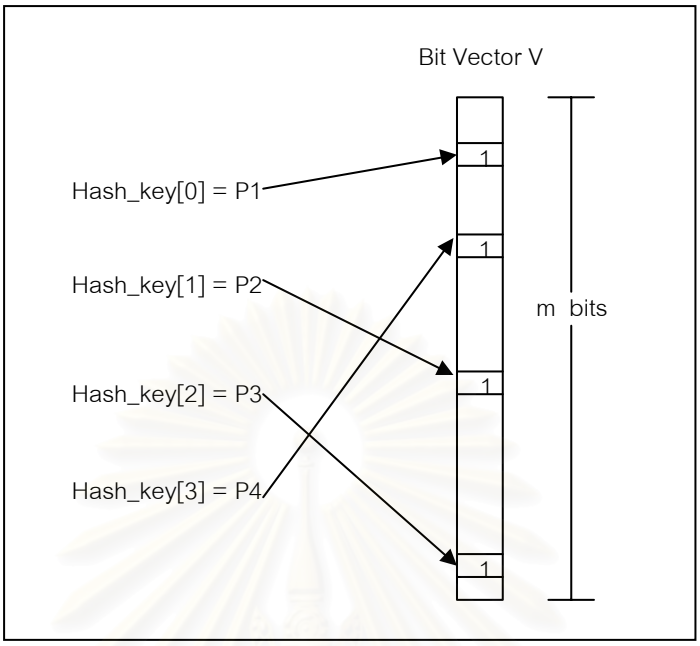
Hash_Key[0] = Temp_Key[0] % (Digest_Size x 8) = 0x05

Hash_Key[1] = Temp_Key[1] % (Digest_Size x 8) = 0x29

Hash_Key[2] = Temp_Key[2] % (Digest_Size x 8) = 0x5f

Hash_Key[3] = Temp_Key[3] % (Digest_Size x 8) = 0x17

- กำหนดค่า ณ ตำแหน่งดรรชนีที่ได้เป็น 1 ดังรูปที่ 2.2



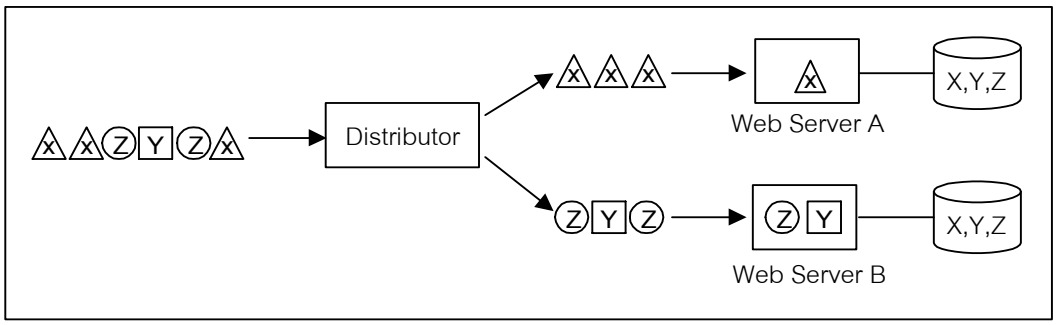
รูปที่ 2.2 การแทนค่าในข้อมูลสรุปบัฟเฟอร์

2. การตรวจสอบข้อมูลการร้องขอในข้อมูลสรุปบัฟเฟอร์

การตรวจสอบว่ามีข้อมูลการร้องขอในข้อมูลสรุปบัฟเฟอร์หรือไม่นั้น จะใช้วิธีเดียวกับการสร้างข้อมูลสรุปบัฟเฟอร์ จนกระทั่งได้ค่าดรรชนีของข้อมูลสรุปบัฟเฟอร์ จากนั้นจึงเปรียบเทียบว่า ณ ตำแหน่งดรรชนีนั้นของข้อมูลสรุปบัฟเฟอร์มีการกำหนดค่าให้เป็น 1 หรือไม่ ถ้าทุกตำแหน่งดรรชนีที่ได้ มีการกำหนดค่าเป็น 1 นั่นคือน่าจะมีข้อมูลเอกสารนั้นในบัฟเฟอร์ แต่ถ้า ณ ตำแหน่งดรรชนีใดได้ค่าเป็น 0 นั่นก็คือไม่มีข้อมูลการร้องขอนั้นในข้อมูลสรุปบัฟเฟอร์

2.1.4 Content-based Routing [3,9,12]

Content-based Routing คือ การกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์ต่าง ๆ โดยพิจารณาจากเนื้อหาของการร้องขอ และกระจายการร้องขอที่มีเนื้อหาเหมือนกันไปยังเว็บเซิร์ฟเวอร์เดียวกัน ซึ่งวิธีนี้จะให้ผลดีคือ สามารถนำข้อมูลที่มีอยู่ในบัฟเฟอร์ตอบกลับให้ไคลเอนต์ได้ทันที โดยอาศัยหลักการที่ว่าการเข้าถึงข้อมูลเอกสารในบัฟเฟอร์น่าจะเร็วกว่าการเข้าถึงข้อมูลเอกสารในดิสก์ ดังรูปที่ 2.3



รูปที่ 2.3 การกระจายการร้องโดยพิจารณาจากเนื้อหาของการร้องขอ

จากรูปที่ 2.3 เมื่อไคลเอนต์ส่งการร้องขอเข้ามาแล้ว ตัวกระจายจะทำหน้าที่รับการร้องขอของไคลเอนต์ และกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์แต่ละตัวโดยพิจารณาจากเนื้อหาของการร้องขอ เพื่อส่งการร้องขอที่มีเนื้อหาเดียวกันไปประมวลผลที่เซิร์ฟเวอร์ตัวเดียวกัน ดังนั้น ทุกการร้องขอ X จะส่งไปยังเซิร์ฟเวอร์ A และทุกการร้องขอ Y, Z จะส่งไปยังเซิร์ฟเวอร์ B

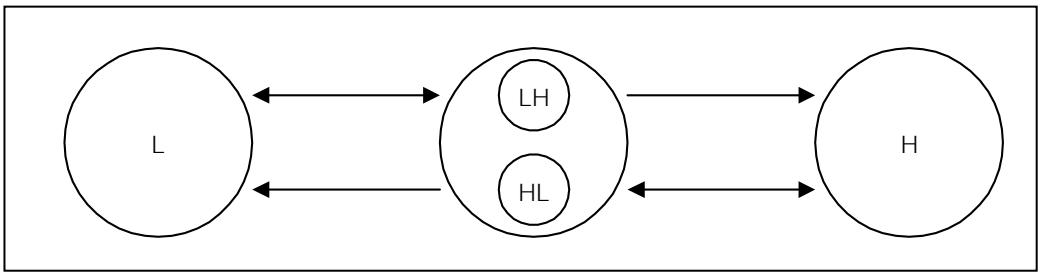
2.1.5 การรวบรวมและแลกเปลี่ยนข้อมูลสภาพภาระงานระหว่างเซิร์ฟเวอร์

เซิร์ฟเวอร์แต่ละตัวจะต้องแลกเปลี่ยนข้อมูลสภาพภาระงานกัน ดังนั้นข้อมูลที่จะแลกเปลี่ยนนั้นจะต้องไม่ทำให้สภาพภาระงานของเซิร์ฟเวอร์เพิ่มขึ้นมากนัก จึงควรมีการแลกเปลี่ยนข้อมูลกันเท่าที่จำเป็นเท่านั้น โดยอาศัยหลักการแก้ปัญหาการกระจายภาระงานตามแบบการสมดุลภาระงานแบบกระจาย [5]

กำหนดให้เซิร์ฟเวอร์มีสภาพภาระงานที่แตกต่างกัน 3 สถานะ คือ สภาพภาระงานน้อย, สภาพภาระงานปานกลาง และสภาพภาระงานมาก โดยการกำหนดค่าขีดแบ่ง (Threshold) 2 ค่า คือ TL (ค่าน้อย) และ TH (ค่ามาก) ในการแบ่งสถานะทั้งสาม และมี LI คือ ความยาวแถวคอยที่รอการประมวลผลและความยาวแถวคอยที่กำลังประมวลผลอยู่ ซึ่งการกำหนดสภาพภาระงานของเซิร์ฟเวอร์จะกำหนดตามเงื่อนไขดังนี้

- เซิร์ฟเวอร์อยู่ในสภาพภาระงานน้อย เมื่อ LI น้อยกว่าหรือเท่ากับ TL
- เซิร์ฟเวอร์อยู่ในสภาพภาระงานปานกลาง เมื่อ LI มากกว่า TL และ LI น้อยกว่าหรือเท่ากับ TH
- เซิร์ฟเวอร์อยู่ในสภาพภาระงานมาก เมื่อ LI มากกว่า TH

สถานะของของเซิร์ฟเวอร์นั้นจะเปลี่ยนแปลงไปตามทิศทางดังรูปที่ 2.4 ให้ L แทนสภาพภาระงานน้อย H แทนสภาพภาระงานมาก ดังนั้น การเปลี่ยนแปลงจากสถานะจากสภาพภาระงานน้อยเป็นสภาพภาระงานปานกลางจะแทนด้วย LH และการเปลี่ยนแปลงสถานะจากสภาพภาระงานมากเป็นสภาพภาระงานปานกลางจะแทนด้วย HL



รูปที่ 2.4 การเปลี่ยนแปลงสถานะของเซิร์ฟเวอร์

จากรูปที่ 2.4 แสดงให้เห็นว่าการเปลี่ยนแปลงสภาพภาระงานที่เป็นไปได้ทั้งหมด คือ L->LH, LH->L, LH->H, H->HL, HL->H, HL->L อย่างไรก็ตาม งานวิจัยนี้จะสนใจสภาพภาระงานที่เปลี่ยนแปลง 2 สถานะเท่านั้น คือ

- LH->H เมื่อสถานะของเซิร์ฟเวอร์เปลี่ยนจากสภาพภาระงานปานกลางเป็นสภาพภาระงานมาก โดยเกิดจากการเปลี่ยนสถานะจากสภาพภาระงานน้อย และสภาพภาระงานเพิ่มขึ้น ๆ จนมีสภาพภาระงานมาก ทำให้เซิร์ฟเวอร์มีสภาพภาระงานที่ค่อนข้างมากแล้ว
- HL->L เมื่อสถานะของเซิร์ฟเวอร์เปลี่ยนจากสภาพภาระงานปานกลางเป็นสภาพภาระงานน้อย โดยเกิดจากการเปลี่ยนสถานะจากสภาพภาระงาน และสภาพภาระงานลดลง จนมีสภาพภาระงานน้อย ทำให้เซิร์ฟเวอร์มีสถานะค่อนข้างว่างแล้ว

นั่นคือ เมื่อเซิร์ฟเวอร์ใดเกิดการเปลี่ยนแปลงสถานะใน 2 สถานะข้างต้นแล้ว ก็จะต้องมีการแจ้งให้เซิร์ฟเวอร์อื่นรู้ถึงสถานะของตัวเอง เพื่อใช้เป็นข้อมูลในการตัดสินใจกระจายภาระงานไปยังเซิร์ฟเวอร์ตัวอื่นต่อไป

2.1.6 **เอนเอสทู** (NS-2 : Network Simulator version 2) [16]

เอนเอสทูเป็นซอฟต์แวร์ที่ใช้ในโปรเจควินท์ (VINT) เพื่อใช้จำลองการทำงานของระบบเครือข่าย โดยเป้าหมายของโปรเจควินท์ คือเพื่อที่จะทำให้เกิดความง่ายในการวิจัยทางด้านระบบเครือข่าย โดยทั้งหมดใช้ภาษาซีพลัสพลัส และใช้ภาษาโอทีซีแอล (OTcl) ซึ่งเป็นภาษาเชิงวัตถุทั้งคู่ ซึ่งมีข้อดีคือ การนำกลับมาใช้ใหม่ และการปรับปรุงแก้ไขในภายหลัง แต่ข้อเสียคือ อาจจะทำงานได้ช้าและเปลืองหน่วยความจำ และต้องมีการวางแผนสำหรับแต่ละส่วนประกอบอย่างระมัดระวังในการเปรียบเทียบประสิทธิภาพการจำลอง

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 A Scalable HTTP Server: The NCSA Prototype [4]

ในงานวิจัย [4] ได้เสนอแนวทางในการกระจายการร้องขอของไคลเอนต์ไปยังเซิร์ฟเวอร์ต่าง ๆ โดยมีดีเอนเอสทำหน้าที่ในการกระจายการร้องขอ โดยไคลเอนต์จถามไปยังดีเอนเอส จากนั้นดีเอนเอสก็จะเปลี่ยนจากยูอาร์แอลเป็นที่อยู่ไอพีของเซิร์ฟเวอร์ตัวหนึ่งในเซิร์ฟเวอร์ทั้งหมด โดยวิธีการเลือกที่อยู่ไอพีของเซิร์ฟเวอร์นั้นจะใช้วิธีวนรอบ (Round Robin)

2.2.2 Scalable Web Server Architectures [2]

ในงานวิจัย [2] ชี้ให้เห็นว่า การกำหนดให้เซิร์ฟเวอร์แต่ละตัว เพื่อให้บริการข้อมูลที่ต่างกันนั้น ทำให้ เซิร์ฟเวอร์แต่ละตัวให้บริการข้อมูลปริมาณที่น้อยลง จึงเสนอให้มีเซิร์ฟเวอร์ตัวกลางที่เก็บข้อมูลว่าแต่ละการร้องขอนั้นจะสามารถใช้บริการได้จากเซิร์ฟเวอร์ใด และเมื่อไคลเอนต์ต้องการร้องขอข้อมูลก็จะต้องส่งการร้องขอมายังเซิร์ฟเวอร์ตัวกลางก่อน จากนั้นเซิร์ฟเวอร์ตัวกลางก็จะเลือกว่าการร้องขอนั้นจะสามารถใช้บริการได้จากเซิร์ฟเวอร์ใด แล้วจึงส่งการร้องขอนั้นไปยังเซิร์ฟเวอร์นั้นโดยใช้ HTTP Redirect

2.2.3 Network Dispatcher: a Connection Router for Scalable Internet Services [6]

ในงานวิจัย [6] ได้เสนอเน็ทเวิร์กดิสแพทเชอร์ คือ เร้าเตอร์ (Router) จะคอยตรวจสอบสภาพภาระงานของแต่ละเซิร์ฟเวอร์ เมื่อมีการร้องขอจากไคลเอนต์เข้ามายังเน็ทเวิร์กดิสแพทเชอร์แล้ว เน็ทเวิร์กดิสแพทเชอร์ก็จะเลือกเซิร์ฟเวอร์ที่มีสภาพภาระงานน้อยที่สุดเพื่อเป็นผู้ให้บริการ จากนั้นก็จะส่งการร้องขอไปยังเซิร์ฟเวอร์นั้นโดยใช้วิธี IP Packet Forwarding

2.2.4 High Performance Web-Server Systems [13]

ในบทความ [13] ชี้ให้เห็นว่าเว็บไซท์ที่ได้รับความนิยมมากนั้นไม่สามารถที่จะรองรับการร้องขอจำนวนมาก ๆ ได้แม้จะใช้เซิร์ฟเวอร์ที่มีความสามารถมาก ๆ หรือการทำ Mirror Sites ก็ตาม ในบทความนี้จึงได้สรุปวิธีการในการสมดุลภาระงาน รวมทั้งวิเคราะห์ข้อดีและข้อจำกัดของแต่ละวิธีด้วย

2.2.5 Redirection Algorithms for Load Sharing in Distributed Web-Server Systems [11]

ในงานวิจัย [11] ได้เสนอสถาปัตยกรรมเว็บเซิร์ฟเวอร์ดังนี้คือ ดีเอนเอสเก็บข้อมูลการเข้าถึงจากทุก ๆ เซิร์ฟเวอร์ในการกำหนดว่าในแต่ละเครือข่ายของไคลเอนต์ควรจะใช้บริการจากเซิร์ฟเวอร์ใด ซึ่ง ดีเอนเอสก็จะใช้ข้อมูลนี้ในการกระจายการร้องขอไปยังแต่ละเซิร์ฟเวอร์ นอกจากนี้ดีเอนเอสก็จะส่งข้อมูลนี้ให้แต่ละเซิร์ฟเวอร์ เพื่อกระจายตัดสินใจให้แต่ละเซิร์ฟเวอร์

ตัดสินใจได้ว่าเซิร์ฟเวอร์นั้นจะต้องให้บริการการร้องขอนั้นหรือควรจะส่งการร้องขอนั้นไปยังเซิร์ฟเวอร์ตัวอื่น

2.2.6 Dynamic Load Balancing on Web-Server Systems [10]

ในบทความ [10] ได้สรุปสถาปัตยกรรมเว็บเซิร์ฟเวอร์แบบกระจาย ซึ่งจะแบ่งตามเอนทิตีที่ทำหน้าที่ในการกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์ คือ Client-based, DNS-based, Dispatcher-based, Server-based และยังได้ประเมินถึงข้อจำกัดของแต่ละวิธี

2.2.7 Load Balancing a Cluster of Web Servers Using Distributed Packet Rewriting [8]

ในงานวิจัย [8] ได้เสนอให้มีดีเอนเอสทำหน้าที่กระจายการร้องขอของไคลเอนต์โดยวิธีวนรอบ เพื่อให้ทุก ๆ เซิร์ฟเวอร์สามารถรับการร้องขอจากไคลเอนต์ได้โดยตรง และเมื่อไคลเอนต์ส่งการร้องขอเข้ามายังเซิร์ฟเวอร์แล้วเซิร์ฟเวอร์นั้นอาจจะให้บริการเอง แต่หากเซิร์ฟเวอร์มีภาระงานก็จะส่งการร้องขอไปให้เซิร์ฟเวอร์อื่นให้เป็นผู้บริการแทนโดยใช้เทคนิค Distributed Packet Rewriting

2.2.8 Load Distribution via Static Scheduling and Client Redirection for Replicated Web Servers [14]

ในงานวิจัย [14] เสนอว่าควรใช้ข้อมูลประวัติการเข้าถึงของไคลเอนต์จากแต่ละเครือข่ายในการคาดคะเนการเข้าถึงในอนาคตที่จะเกิดขึ้น เพื่อกำหนดว่าเซิร์ฟเวอร์ใดจะต้องให้บริการแก่ไคลเอนต์ที่มาจากเครือข่ายใดบ้าง

นั่นคือ เมื่อไคลเอนต์ต้องการติดต่อไปยังเซิร์ฟเวอร์ก็จะต้องติดต่อไปยังดีเอนเอสก่อน แล้วดีเอนเอสก็จะตอบกลับเป็นเลขที่อยู่ไอพีของเซิร์ฟเวอร์ตัวหนึ่งในเซิร์ฟเวอร์ทั้งหมด จากนั้นไคลเอนต์จึงส่งการร้องขอไปยังเซิร์ฟเวอร์ตัวนั้น เซิร์ฟเวอร์ก็จะพิจารณาว่าไคลเอนต์นั้นควรจะใช้บริการจากเซิร์ฟเวอร์ใดก็จะส่งการร้องขอไปยังเซิร์ฟเวอร์นั้น และหากเซิร์ฟเวอร์ที่จะต้องให้บริการการร้องขอนั้นมีสภาพภาระงาน ก็จะสามารถส่งการร้องขอไปให้เซิร์ฟเวอร์อื่นเป็นผู้ให้บริการแทนได้

นอกจากนี้หากเครือข่ายของไคลเอนต์นั้นยังไม่ถูกกำหนดว่าจะให้เซิร์ฟเวอร์ใดเป็นผู้ให้บริการ เซิร์ฟเวอร์ที่เป็นผู้รับการร้องขอนั้นมา ก็จะกำหนดเซิร์ฟเวอร์ที่จะให้บริการแก่ไคลเอนต์เอง ซึ่งจะกำหนดโดยใช้วิธีวนรอบ

2.2.9 Locality-Aware Request Distribution in Web Server Clusters [12]

ในงานวิจัย [12] เสนอให้มี ฟรอนท์เอนด์ (Front End) ทำหน้าที่กระจายการร้องขอไปยังเซิร์ฟเวอร์แต่ละตัว โดยการพิจารณาจากเนื้อหาของการร้องขอและสภาพภาระงานของเซิร์ฟเวอร์

เพื่อเลือกเซิร์ฟเวอร์ที่จะให้บริการการร้องขอนั้น โดยจะส่งการร้องขอที่มีเนื้อหาเหมือนกันไปประมวลผลที่เซิร์ฟเวอร์เดียวกัน ทำให้เซิร์ฟเวอร์สามารถนำข้อมูลในหน่วยความจำหลักมาใช้ในการตอบการร้องขอของไคลเอนต์ได้ แต่การกระจายการร้องขอโดยพิจารณาจากเนื้อหาของการร้องขอนั้นจะต้องมีการสร้างการเชื่อมต่อกับฟรอนท์เอนด์ก่อน จึงจะสามารถรู้ได้ว่าเนื้อหาของการร้องขอนั้นคืออะไร จากนั้นฟรอนท์เอนด์จึงส่งการร้องขอนั้นไปยังเซิร์ฟเวอร์ที่จะเป็นผู้ให้บริการ

2.2.10 Efficient Support for Content-based Routing in Web Server Clusters [3]

ในงานวิจัย [3] เสนอว่าควรมีการกระจายการร้องขอของไคลเอนต์ไปยังเซิร์ฟเวอร์ต่าง ๆ โดยพิจารณาจากเนื้อหาของการร้องขอ คือ เมื่อไคลเอนต์ส่งการร้องขอเข้ามาและสร้างการเชื่อมต่อกับตัวกระจายแล้ว ตัวกระจายจะตัดสินใจเลือกเซิร์ฟเวอร์ที่จะให้บริการ โดยพิจารณาจากเนื้อหาของการร้องขอและสภาพภาระงานของเซิร์ฟเวอร์ เพื่อส่งการร้องขอที่มีเนื้อหาเหมือนกันไปประมวลผลที่เซิร์ฟเวอร์เดียวกัน นอกจากนี้ตัวกระจายยังมีความสามารถในการกระจายการร้องขอที่มีเนื้อหาพลวัต (Dynamic Content) ไปยังเซิร์ฟเวอร์คนละตัวกับที่ให้บริการการร้องขอที่มีเนื้อหาสถิต อีกทั้งยอมให้เซิร์ฟเวอร์แต่ละตัวนั้นเก็บข้อมูลที่ต่างกันได้

2.2.11 Scalable Content-aware Request Distribution in Cluster-based Network Servers [9]

ในงานวิจัย [9] ได้เสนอสถาปัตยกรรมที่ปรับขนาดได้ สำหรับการกระจายการร้องขอโดยพิจารณาจากเนื้อหาของการร้องขอ กล่าวคือ เมื่อไคลเอนต์ส่งการร้องขอเข้ามา จะมี Layer 4 Switch ทำหน้าที่ในการรับการร้องขอ และกระจายการร้องขอไปยังเซิร์ฟเวอร์แต่ละตัว จากนั้นแต่ละเซิร์ฟเวอร์นั้นจะถามไปยังดิสแพทเชอร์ เพื่อให้ตัดสินใจเลือกเซิร์ฟเวอร์ที่จะให้บริการการร้องขอโดยพิจารณาจากเนื้อหาของการร้องขอ เมื่อดิสแพทเชอร์ ตอบกลับว่าควรใช้บริการจากเซิร์ฟเวอร์ใด ก็จะส่งการร้องขอนั้นไปยังเซิร์ฟเวอร์นั้น

2.2.12 Summary Cache [7]

ในงานวิจัย [7] เสนอว่าควรจะมีการใช้ข้อมูลการร้องขอร่วมกันระหว่างพรอกซี โดยจะเก็บและแลกเปลี่ยนข้อมูลที่อยู่ในรูปของซัมมารีแคช (Summary Cache) ซึ่งก็คือ ข้อมูลผลสรุปของข้อมูลการร้องขอในแคชของแต่ละพรอกซี เพื่อใช้ในการตรวจสอบก่อนว่ามีข้อมูลที่ต้องการอยู่ในแคชของพรอกซีนั้นหรือไม่ก่อนที่จะร้องขอข้อมูลจากพรอกซีนั้น ซึ่งซัมมารีแคชนี้จะมีการเปลี่ยนแปลงทุกช่วงเวลาที่กำหนดเท่านั้น นอกจากนี้ซัมมารีแคชยังมีข้อดีคือ จัดเก็บเป็นข้อมูลขนาดเล็ก ซึ่งจะช่วยลดจำนวนข้อมูลที่แลกเปลี่ยนกันระหว่างพรอกซี

2.2.13 Cache Digests [1]

ในงานวิจัย [1] เสนอการทำแคชไดเจสท์ (Cache Digests) สำหรับใช้ในเว็บแคช ซึ่งแต่ละพรอกซีจะนำข้อมูลที่มีอยู่ในแคชมาสร้างเป็นแคชไดเจสท์ที่มีขนาดเล็ก เพื่อให้พรอกซีตัวอื่นใช้แคชไดเจสท์ในการตรวจสอบได้ว่าพรอกซีใดน่าจะมีข้อมูลของเอกสารที่ต้องการ

2.2.14 An Adaptive Load Balancing Algorithm Using Simple Prediction [5]

ในงานวิจัย [5] เสนอขั้นตอนวิธีในการกระจายภาระงานของเซิร์ฟเวอร์ให้เท่ากัน โดยใช้วิธีการทำนายแบบง่าย คือ แต่ละเซิร์ฟเวอร์จะมีการเก็บข้อมูลสภาพภาระงานของเซิร์ฟเวอร์และแลกเปลี่ยนกันระหว่างเซิร์ฟเวอร์ โดยจะแลกเปลี่ยนข้อมูลกันเท่าที่จำเป็น กล่าวคือ ถ้ามีเซิร์ฟเวอร์ใดโอนงานมาให้นั่นคือเซิร์ฟเวอร์นั้นมีสภาพภาระงานมาก และเมื่อสภาพภาระงานของเซิร์ฟเวอร์ลดลงจากสภาพภาระงานมากเป็นสภาพภาระงานน้อยแล้วก็จะต้องแจ้งให้เซิร์ฟเวอร์ตัวอื่นรู้ ซึ่งเซิร์ฟเวอร์สามารถใช้ข้อมูลนี้ในการทำนายสภาพภาระงานของเซิร์ฟเวอร์ตัวอื่นได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

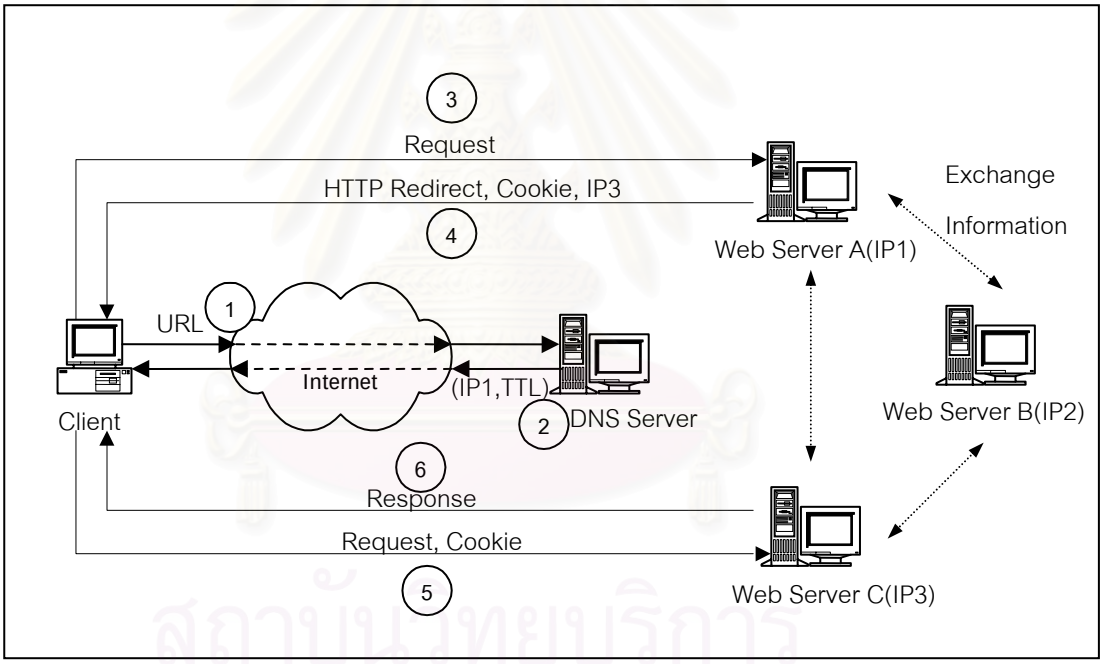
บทที่ 3

ระบบเว็บเซิร์ฟเวอร์โดยใช้การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา

ในบทนี้จะกล่าวถึงวิธีการทำงานของเว็บเซิร์ฟเวอร์ที่ใช้วิธีกระจายการร้องขอ โดยใช้การสมดุลภาระแบบพลวัต และกระจายการร้องขอโดยพิจารณาจากเนื้อหา

3.1 การทำงานของระบบ

การทำงานของเว็บเซิร์ฟเวอร์ จะประกอบด้วยการให้บริการการร้องขอข้อมูล การกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่น และการแลกเปลี่ยนข้อมูลที่ใช้ในการตัดสินใจกระจายการร้องขอ



รูปที่ 3.1 การทำงานและการให้บริการการร้องขอของเซิร์ฟเวอร์

จากรูปที่ 3.1 แสดงขั้นตอนการทำงานและการให้บริการการร้องขอของเว็บเซิร์ฟเวอร์ ดังนี้

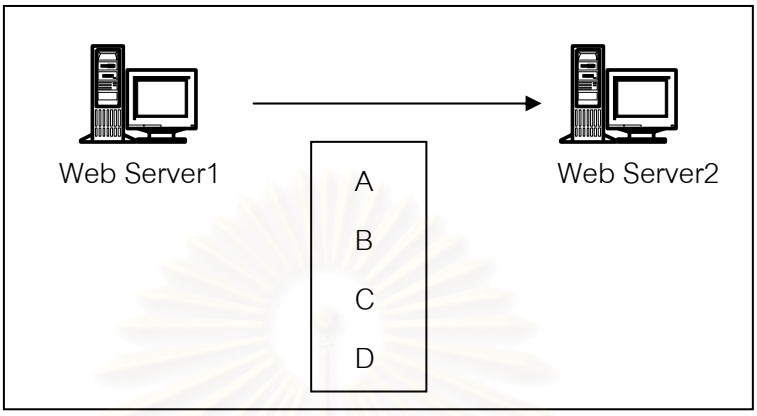
1. ไคลเอนต์เปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ (Domain Name Mapping) โดยการสอบถาม (DNS Query) ไปยังดีเอนเอส

2. ดีเอนเอสจะตอบกลับไปยังไคลเอนต์ เป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ตัวหนึ่งในเว็บเซิร์ฟเวอร์ทั้งหมด โดยดีเอนเอสจะตอบกลับมาโดยใช้วิธีวนรอบ (DNS Round Robin) ในที่นี้คือ เลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ A
3. ไคลเอนต์จะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ตามเลขที่อยู่ไอพีที่ได้จากดีเอนเอส ซึ่งก็คือ เว็บเซิร์ฟเวอร์ A
4. เว็บเซิร์ฟเวอร์ A ได้รับการร้องขอจากไคลเอนต์ก็จะตรวจสอบก่อนว่ามีการกำหนดค่าคุกกี้มาพร้อมกับการร้องขอนั้นหรือไม่ ในที่นี้พบว่าไม่มีการกำหนดค่าคุกกี้ เว็บเซิร์ฟเวอร์ A ก็จะทำการกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่น่าจะมีข้อมูลเอกสารที่ร้องขอนี้ในบัฟเฟอร์ โดยพิจารณาจากข้อมูลสรุปบัฟเฟอร์ และข้อมูลสภาพภาระงาน ในที่นี้เมื่อตรวจสอบจากข้อมูลสรุปบัฟเฟอร์ พบว่ามีข้อมูลเอกสารที่ร้องขออยู่ในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ C และเว็บเซิร์ฟเวอร์ C มีสภาพภาระงานปานกลาง เว็บเซิร์ฟเวอร์ A จึงเลือกที่จะกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์ C โดยกำหนดค่าคุกกี้และส่งค่าคุกกี้ พร้อมตอบการร้องขอไปยังไคลเอนต์ โดยกำหนดรหัสตอบกลับเป็นรหัสของการรีไดเร็กท์ และกำหนดในเฮดเดอร์ส่วนที่บอกตำแหน่งเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ C
5. ไคลเอนต์ส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ตามที่ระบุไว้ในเฮดเดอร์ส่วนที่บอกตำแหน่ง ซึ่งในที่นี้ก็คือ เว็บเซิร์ฟเวอร์ C โดยส่งค่าคุกกี้ไปพร้อมกับการร้องขอ
6. เว็บเซิร์ฟเวอร์ C ได้รับการร้องขอจากไคลเอนต์ ตรวจพบว่าการร้องขอนั้นมีการกำหนดค่าคุกกี้ ดังนั้นเว็บเซิร์ฟเวอร์ C จะต้องเป็นผู้ให้บริการการร้องขอนั้น โดยส่งข้อมูลเอกสารให้กับไคลเอนต์

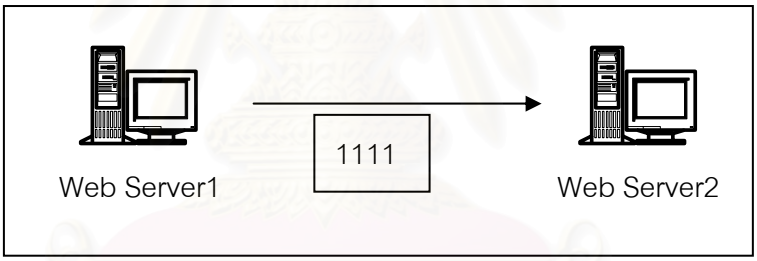
จากขั้นตอนการทำงานข้างต้น เว็บเซิร์ฟเวอร์จะต้องมีการแลกเปลี่ยนข้อมูลกันเพื่อใช้ในการตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่จะให้บริการการร้องขอ ซึ่งข้อมูลดังกล่าวได้แก่

1. ข้อมูลสรุปบัฟเฟอร์ เว็บเซิร์ฟเวอร์จะต้องมีการแลกเปลี่ยนข้อมูลกันเพื่อให้รู้ว่าแต่ละเว็บเซิร์ฟเวอร์มีข้อมูลเอกสารใดในบัฟเฟอร์บ้าง โดยทั่วไปจะใช้การแลกเป็นรายชื่อของเอกสารที่มีทั้งหมด ดังรูปที่ 3.2 ซึ่งถ้าเอกสารในบัฟเฟอร์มีจำนวนมาก รายชื่อเอกสารก็จะมีขนาดใหญ่ ใช้เวลาในการแลกเปลี่ยนมาก งานวิจัยนี้จึงใช้หลักการของ Summary Cache [7] และ Cache Digests [1] ในการนำเอกสารในบัฟเฟอร์มาคำนวณสร้างเป็นข้อมูลผลสรุป แทนเอกสารทั้งหมดที่มีในบัฟเฟอร์ เรียกว่า ข้อมูลสรุปบัฟเฟอร์ และใช้ข้อมูลนี้แลกเปลี่ยนกับเว็บเซิร์ฟเวอร์อื่น ดังรูปที่ 3.3 โดยเว็บ

เซิร์ฟเวอร์จะกำหนดช่วงเวลาในการนำข้อมูลการร้องขอที่มีอยู่ในบัฟเฟอร์ มาคำนวณเป็นข้อมูลสรุปบัฟเฟอร์แล้วแลกเปลี่ยนกับเว็บเซิร์ฟเวอร์ตัวอื่น



รูปที่ 3.2 การแลกเปลี่ยนข้อมูลรายชื่อเอกสารที่มีในบัฟเฟอร์



รูปที่ 3.3 การแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์

ตัวอย่างการพิจารณาว่ามีเอกสารที่ต้องการนั้นในบัฟเฟอร์หรือไม่ โดยการตรวจสอบจากข้อมูลสรุปบัฟเฟอร์ ทำได้โดยการนำเอกสารไปคำนวณได้เป็นเลขฐานสิบหก แล้วนำค่าที่ได้ไปเปรียบเทียบกับข้อมูลสรุปบัฟเฟอร์ด้วยวิธีการออร์ (OR Operation) ถ้าพบว่าได้ค่าตรงกับข้อมูลสรุปบัฟเฟอร์ นั่นคือ น่าจะมีข้อมูลเอกสารนั้นในบัฟเฟอร์ แต่ถ้าค่าที่ได้ไม่ตรงกับข้อมูลสรุปบัฟเฟอร์ นั่นคือ ไม่มีข้อมูลเอกสารนั้นในบัฟเฟอร์ แต่อย่างไรก็ตาม บางครั้งการนำค่าไปเปรียบเทียบกับข้อมูลสรุปบัฟเฟอร์นั้นได้ค่าเท่ากัน คือน่าจะมีข้อมูลเอกสารในบัฟเฟอร์ แต่ความเป็นจริงไม่มีข้อมูลเอกสารนั้นในบัฟเฟอร์

ตารางที่ 3.1 ตัวอย่างการเข้ารหัสเอกสาร

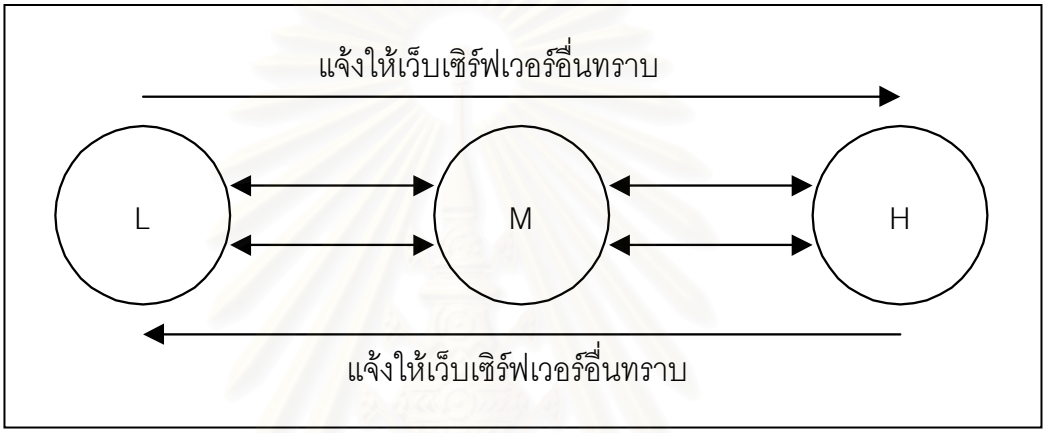
เอกสาร	เลขฐานสิบหก
A	1001
B	0011
C	1100
D	1010

จากตารางที่ 3.1 แสดงผลจากการนำเอกสารมาคำนวณเป็นเลขฐานสิบหก ถ้ากำหนดให้บัฟเฟอร์มีข้อมูลเอกสาร A และเอกสาร B นำมาคำนวณเป็นข้อมูลสรุปบัฟเฟอร์ด้วยวิธีการออร์ ได้เป็น 1011 ดังนั้น เมื่อมีการร้องขอเอกสาร ก็จะต้องทำการตรวจสอบจากข้อมูลสรุปบัฟเฟอร์ ดังนี้

- เอกสาร A คำนวณเป็นเลขฐานสิบหก ได้เป็น 1001 เมื่อนำไปเปรียบเทียบกับข้อมูลสรุปบัฟเฟอร์ และได้ค่าเดียวกันกับข้อมูลสรุปบัฟเฟอร์ นั่นคือ น่าจะมีข้อมูลเอกสารในบัฟเฟอร์ ซึ่งผลจากการตรวจสอบนี้ถูกต้อง
- เอกสาร B คำนวณเป็นเลขฐานสิบหก ได้เป็น 0011 เมื่อนำไปเปรียบเทียบกับข้อมูลสรุปบัฟเฟอร์ และได้ค่าเดียวกันกับข้อมูลสรุปบัฟเฟอร์ นั่นคือ น่าจะมีข้อมูลเอกสารในบัฟเฟอร์ ซึ่งผลจากการตรวจสอบนี้ถูกต้อง
- เอกสาร C คำนวณเป็นเลขฐานสิบหก ได้เป็น 1100 เมื่อนำไปเปรียบเทียบกับข้อมูลสรุปบัฟเฟอร์ ไม่ได้ค่าเดียวกันกับข้อมูลสรุปบัฟเฟอร์ นั่นคือ ตรวจสอบพบว่าไม่มีข้อมูลเอกสารในบัฟเฟอร์ ซึ่งผลจากการตรวจสอบนี้ถูกต้อง
- เอกสาร D คำนวณเป็นเลขฐานสิบหก ได้เป็น 0011 เมื่อนำไปเปรียบเทียบกับข้อมูลสรุปบัฟเฟอร์ และได้ค่าเดียวกันกับข้อมูลสรุปบัฟเฟอร์ นั่นคือ จากการตรวจสอบพบน่าจะมีข้อมูลเอกสารในบัฟเฟอร์ แต่ความเป็นจริงไม่มีข้อมูลเอกสาร D ในบัฟเฟอร์ ซึ่งผลจากการตรวจสอบนี้มีความผิดพลาด

จากที่กล่าวมาข้างต้น แสดงให้เห็นว่าข้อมูลสรุปบัฟเฟอร์นี้ไม่ถูกต้องทั้ง 100 เปอร์เซ็นต์ อาจจะเกิดข้อผิดพลาดขึ้นได้ แต่เราจะยอมรับถึงความผิดพลาดนี้ เพื่อให้ข้อมูลที่ใช้ในการตรวจสอบเอกสารในบัฟเฟอร์ที่มีการแลกเปลี่ยนกันระหว่างเว็บเซิร์ฟเวอร์ มีขนาดเล็กลง

2. ข้อมูลสภาพภาระงาน เว็บเซิร์ฟเวอร์จะกำหนดช่วงเวลาในการตรวจสอบสภาพภาระงาน ณ ขณะนั้น กล่าวคือ ถ้าพบว่าสภาพภาระงานเปลี่ยนแปลงจากสภาพภาระงานน้อยเป็นสภาพภาระงานมาก ก็จะแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบว่าตนเองมีสภาพภาระงานมาก ไม่สามารถแบ่งเบาภาระงานจากเว็บเซิร์ฟเวอร์อื่นได้ และถ้าสภาพภาระงานเปลี่ยนแปลงจากสภาพภาระงานมากเป็นสภาพภาระงานน้อย ก็จะแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบว่าตนเองมีสภาพภาระงานน้อย พร้อมที่จะรับภาระงานจากเว็บเซิร์ฟเวอร์อื่นได้ ดังรูปที่ 3.4



รูปที่ 3.4 การตรวจสอบสภาพภาระงานเพื่อแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบ

จากรูปที่ 3.4 ให้ L แทนสภาพภาระงานน้อย M แทนสภาพภาระงานปานกลาง และ H แทนสภาพภาระงานมาก เมื่อสภาพภาระงานของเว็บเซิร์ฟเวอร์มีการเปลี่ยนแปลง เว็บเซิร์ฟเวอร์ก็จะพิจารณาว่าจะแจ้งให้กับเว็บเซิร์ฟเวอร์อื่นทราบหรือไม่ โดยพิจารณาดังนี้

- กรณี L -> M -> H เมื่อมีการเปลี่ยนแปลงสภาพภาระงานจากสภาพภาระงานน้อยเพิ่มขึ้นเป็นสภาพภาระงานปานกลางจนเป็นสภาพภาระงานมาก จะแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบ เพื่อให้เว็บเซิร์ฟเวอร์อื่นทราบว่าตนเองมีสภาพภาระงานมาก ไม่รับการกระจายภาระงานจากเว็บเซิร์ฟเวอร์อื่น
- กรณี H -> M -> L เมื่อมีการเปลี่ยนแปลงสภาพภาระงานจากสภาพภาระงานมากลดลง เป็นสภาพภาระงานปานกลาง จนเป็นสภาพภาระงานน้อย จะแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบ เพื่อให้เว็บเซิร์ฟเวอร์อื่นทราบว่าตนเองมีสภาพภาระงานน้อย สามารถที่จะรับการกระจายภาระงานจากเว็บเซิร์ฟเวอร์อื่นได้

- กรณีที่สภาพภาระงานมีการเปลี่ยนแปลงไปมาระหว่างสภาพภาระงานที่ติดกัน คือ เปลี่ยนแปลงไปมาระหว่างสภาพภาระงานน้อยกับสภาพภาระงานปานกลาง หรือ เปลี่ยนแปลงระหว่างสภาพภาระงานปานกลางกับสภาพภาระงานมาก เช่น L -> M -> L, M -> L -> M, M -> H -> M, H -> M -> H ถ้าสภาพภาระงานมีการเปลี่ยนแปลงดังนี้ ก็จะไม่แจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบ จนกว่าจะเป็นการเปลี่ยนแปลง L -> M -> Hs, H -> M -> L จึงจะแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

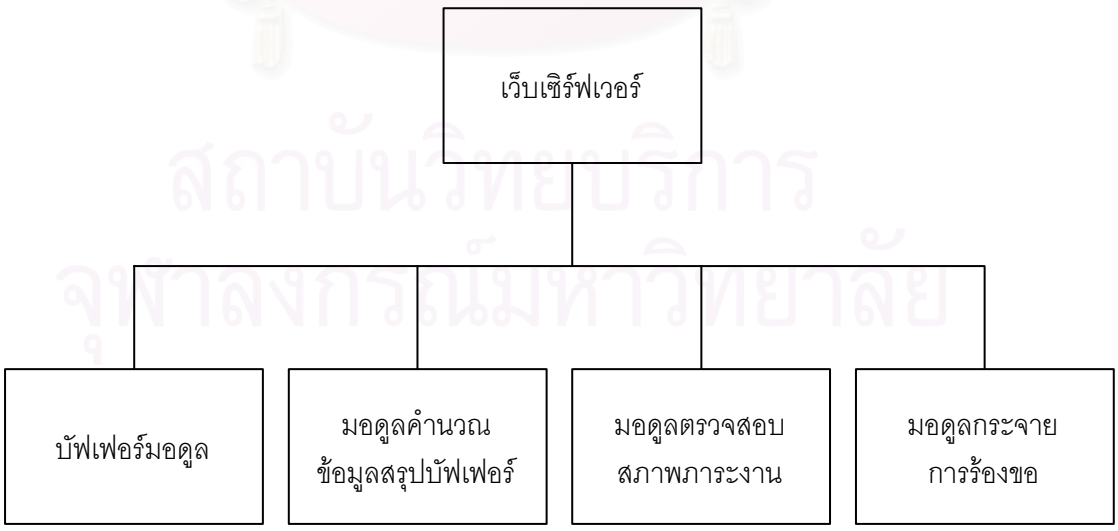
บทที่ 4

การทดสอบ และผลการทดสอบ

ในบทนี้จะกล่าวถึงระบบจำลองที่ใช้ในการการทดสอบ และผลการทดสอบประสิทธิภาพของการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา โดยจะทำการทดลองเปรียบเทียบกับการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส และการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยพิจารณาจากข้อมูลสภาพภาระงาน เพื่อเปรียบเทียบประสิทธิภาพของระบบ

4.1 การจำลองระบบเพื่อใช้ในการทดสอบ

ในวิทยานิพนธ์นี้จะทำการทดสอบประสิทธิภาพของระบบที่ได้ออกแบบขึ้นสำหรับทดลองการกระจายการร้องขอ โดยจะจำลองการทำงานของระบบภายใต้สภาพแวดล้อมที่แตกต่างกัน เปรียบเทียบกับการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยพิจารณาจากข้อมูลสภาพภาระงาน และการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส เพื่อทดสอบประสิทธิภาพของระบบที่ได้พัฒนาขึ้น ซึ่งวิทยานิพนธ์นี้ใช้โปรแกรมจำลองเอนเอสทูในการพัฒนา โดยใช้วิธีการทำงานแบบการจำลองแบบติดตาม (Trace-driven Simulation) ซึ่งเป็นการจำลองที่จะต้องใช้ข้อมูลการใช้เว็บ (Log) ในการจำลองการทำงานไปตามเหตุการณ์ที่เกิดขึ้นในข้อมูลการใช้เว็บ โดยเว็บเซิร์ฟเวอร์ที่ใช้ในการกระจายการร้องขอมีโครงสร้างการทำงาน ดังรูปที่ 4.1



รูปที่ 4.1 โครงสร้างของระบบ

จากรูปที่ 4.1 เว็บเซิร์ฟเวอร์ของระบบ ประกอบด้วยส่วนหลัก ๆ 4 ส่วน คือ

1. บัฟเฟอร์มอดูล
2. มอดูลคำนวณข้อมูลสรุปบัฟเฟอร์
3. มอดูลตรวจสอบสภาพภาระงาน
4. มอดูลกระจายการร้องขอ

4.1.1 หลักการจำลอง

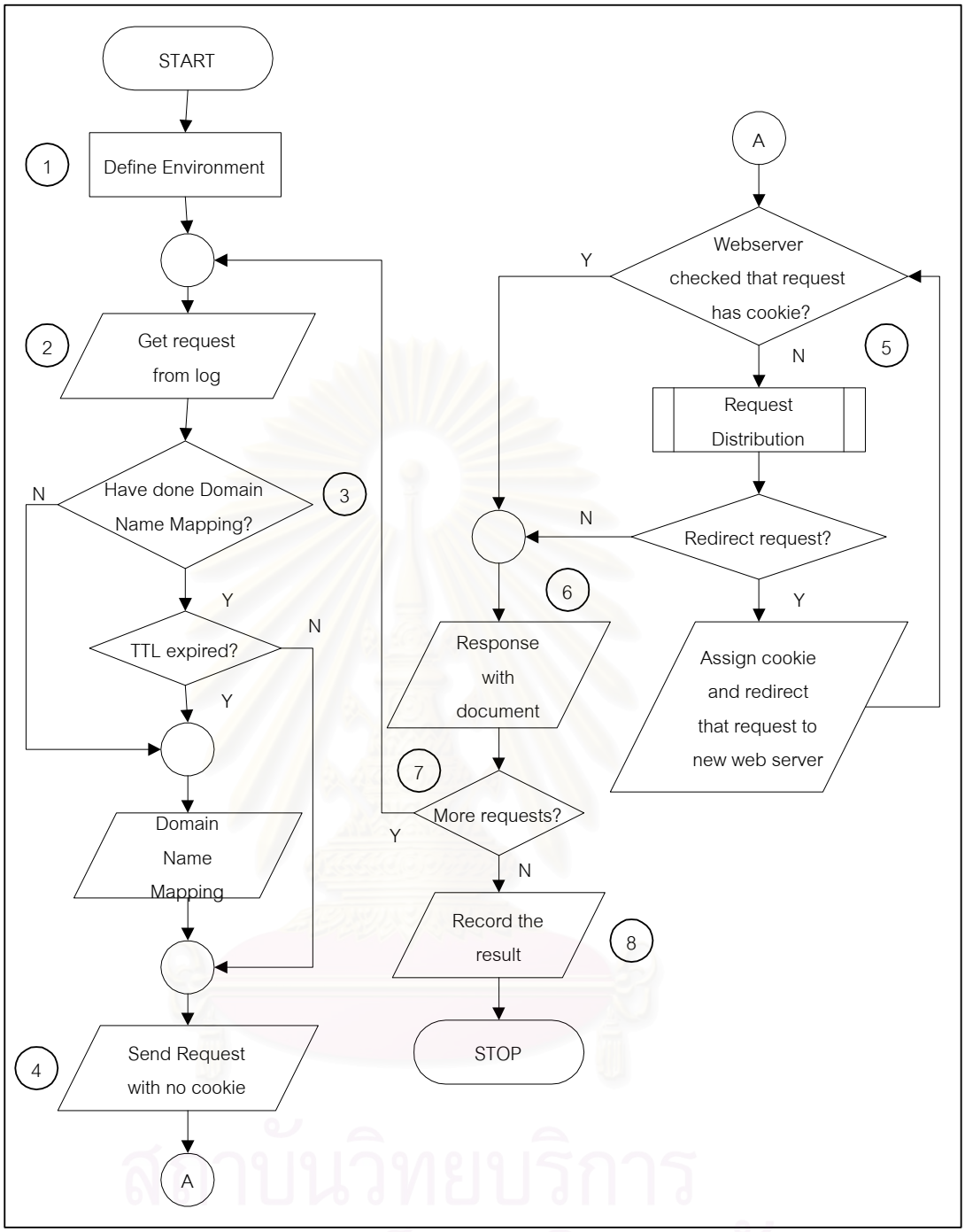
โปรแกรมจำลองระบบที่สร้างขึ้น ประกอบด้วยส่วนหลัก ๆ 5 ส่วน คือ

1. โปรแกรมจำลอง (Trace Simulator) ทำหน้าที่ควบคุมการจำลองทั้งหมด และสรุปผลที่ได้เมื่อสิ้นสุดการทำงาน โดยใช้ข้อมูลการใช้เว็บในการจำลอง
2. บัฟเฟอร์มอดูล ทำหน้าที่จัดการจัดเก็บเอกสารในบัฟเฟอร์
3. มอดูลกระจายการร้องขอ ทำหน้าที่ตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่ควรจะเป็นผู้ให้บริการในแต่ละการร้องขอ
4. มอดูลคำนวณข้อมูลสรุปบัฟเฟอร์ ทำหน้าที่นำข้อมูลเอกสารที่มีอยู่ในบัฟเฟอร์มาสร้างเป็นข้อมูลสรุปบัฟเฟอร์ และยังใช้ตรวจสอบจากข้อมูลสรุปบัฟเฟอร์ว่าน่าจะมีข้อมูลเอกสารในเว็บเซิร์ฟเวอร์ใดหรือไม่ นอกจากนี้ยังทำหน้าที่ในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ระหว่างเว็บเซิร์ฟเวอร์ด้วย
5. มอดูลตรวจสอบสภาพภาระงาน ทำหน้าที่ตรวจสอบสภาพภาระงานของเว็บเซิร์ฟเวอร์ เพื่อแจ้งถึงสภาพภาระงานที่เปลี่ยนไปให้เว็บเซิร์ฟเวอร์อื่นทราบ

4.1.2 ขั้นตอนการทำงานของโปรแกรมจำลอง

โปรแกรมจำลองที่ใช้ในงานวิทยานิพนธ์ มีขั้นตอนการทำงาน ดังนี้

1. กำหนดสภาพแวดล้อมของระบบ
2. อ่านข้อมูลจากข้อมูลการใช้เว็บ เพื่อใช้ในการจำลองการร้องขอที่มีผู้ใช้งานร้องขอเข้ามาในระบบ
3. ถ้าไคลเอนต์ไม่เคยทำการเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์หรือเคยทำแต่หมดช่วงอายุทีทีแอลที่ดีเอนเอสกำหนดแล้ว ไคลเอนต์จะส่งการร้องขอถามไปยังดีเอนเอส แล้วดีเอนเอสจะตอบกลับมาเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์หนึ่งในเว็บเซิร์ฟเวอร์ทั้งหมด
4. ไคลเอนต์ส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ โดยการร้องขอใหม่ที่อ่านจากข้อมูลการใช้เว็บ จะไม่กำหนดค่าคุกกี้



รูปที่ 4.2 ขั้นตอนการทำงานของโปรแกรมจำลอง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5. เว็บเซิร์ฟเวอร์ได้รับการร้องขอ แล้วตรวจสอบว่าการร้องขอนั้นมีการกำหนดค่าคุกกี้หรือไม่ ถ้ามีการกำหนดค่าคุกกี้ เว็บเซิร์ฟเวอร์ก็จะให้บริการต่อการร้องขอนั้น แต่ถ้าไม่มีการกำหนดค่าคุกกี้ เว็บเซิร์ฟเวอร์ก็จะตัดสินใจเลือกเว็บเซิร์ฟเวอร์ที่จะเป็นผู้ให้บริการโดยทำการกำหนดค่าคุกกี้ แล้วส่งคุกกี้ไปพร้อมกับการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่เลือกโดยใช้ Http Redirect
6. ไคลเอนต์ได้รับข้อมูลเอกสารที่ร้องขอไป
7. โปรแกรมจะวนอ่านข้อมูลการร้องขอและทำการจำลองต่อไปจนกว่าจะครบตามช่วงเวลาการจำลองที่ได้กำหนดไว้
8. เมื่อจำลองเสร็จแล้ว โปรแกรมจะบันทึกผลสรุปที่ได้จากการทดลอง

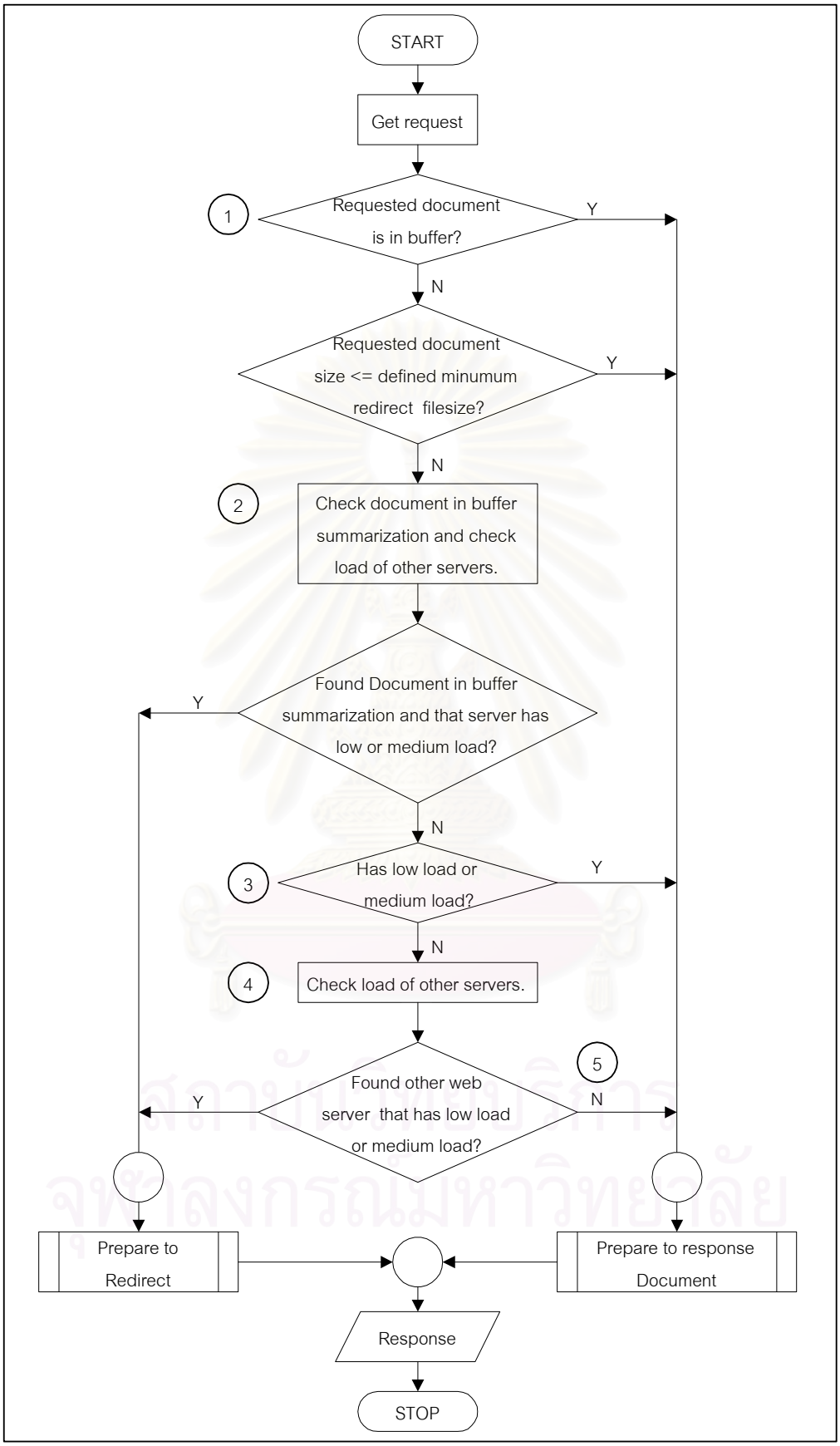
4.1.3 **รายละเอียดของบัฟเฟอร์มอดูล**

บัฟเฟอร์มอดูลในวิทยานิพนธ์นี้จะใช้วิธีแทนที่เอกสารแบบแอลอาร์ยู ซึ่งโครงสร้างของบัฟเฟอร์มอดูล จะประกอบด้วย 2 ส่วนหลัก ๆ คือ ออบเจ็กต์ PageList และ ออบเจ็กต์ LruList โดยที่ออบเจ็กต์ PageList เป็นกลุ่มของรายการโยง (Linked List) ที่ถูกแยกด้วยฟังก์ชันแฮช (Hash Entry) จะใช้จัดเก็บข้อมูลเอกสารที่มีอยู่ในบัฟเฟอร์ ส่วนออบเจ็กต์ LruList เป็นรายการโยงสำหรับจัดเก็บข้อมูลเอกสารที่มีอยู่ในบัฟเฟอร์ โดยจัดเรียงตามลำดับการร้องขอเอกสารตามวิธีแทนที่เอกสารแบบแอลอาร์ยู

เมื่อไคลเอนต์ร้องขอข้อมูลเอกสารจากเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ก็จะตรวจสอบก่อนว่ามีเอกสารในออบเจ็กต์ PageList หรือไม่ ถ้ามีก็จะนำข้อมูลเอกสารนั้นส่งให้ไคลเอนต์ และทำการเปลี่ยนตำแหน่งของเอกสารในออบเจ็กต์ LruList ไปยังตำแหน่งสุดท้ายของรายการโยง แต่ถ้าไม่มีเอกสารในออบเจ็กต์ PageList ก็จะอ่านข้อมูลเอกสารนั้นจากดิสก์แล้วจัดเก็บในออบเจ็กต์ PageList และ LruList โดยก่อนที่จะเพิ่มข้อมูลเอกสารนั้น จะตรวจสอบก่อนว่าบัฟเฟอร์มีพื้นที่เพียงพอที่จะจัดเก็บเอกสารนั้นหรือไม่ ถ้ามีพื้นที่ไม่เพียงพอก็จะลบข้อมูลเอกสารที่มีอยู่ในบัฟเฟอร์ที่ไม่ถูกร้องขอนานที่สุด โดยจะลบข้อมูลเอกสารที่ต้น LruList ก่อนและลบใน PageList จนกว่าบัฟเฟอร์จะมีพื้นที่เพียงพอ แล้วจึงค่อยจัดเก็บข้อมูลเอกสารใหม่ใน PageList และตำแหน่งสุดท้ายของ LruList

4.1.4 **รายละเอียดของมอดูลกระจายการร้องขอ**

มอดูลกระจายการร้องขอจะพิจารณาตัดสินใจว่าการร้องขอแต่ละการร้องขอนั้น ควรจะให้เว็บเซิร์ฟเวอร์ใดเป็นผู้ให้บริการ โดยจะกระจายการร้องขอเฉพาะการร้องขอเอกสารที่มีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ ซึ่งการตัดสินใจนั้นจะพิจารณาจากเนื้อหาของการร้องขอและสภาพภาระงานของแต่ละเว็บเซิร์ฟเวอร์



รูปที่ 4.3 ขั้นตอนการกระจายการร้องขอ

จากรูปที่ 4.3 แสดงกลยุทธ์ในการกระจายการร้องขอที่ใช้ในวิทยานิพนธ์นี้ ซึ่งมีขั้นตอนดังนี้ คือ

1. ตรวจสอบพบว่ามีข้อมูลการร้องขอในบัฟเฟอร์ หรือเอกสารที่ร้องขอนั้นมีขนาดไม่เกินขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ เว็บเซิร์ฟเวอร์ที่รับการร้องขอก็จะให้บริการการร้องขอนั้นเอง
2. แต่ถ้าเอกสารมีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ และไม่พบข้อมูลเอกสารในบัฟเฟอร์ เว็บเซิร์ฟเวอร์ก็จะนำเนื้อหาของการร้องขอนั้นไปตรวจสอบกับข้อมูลสรุปบัฟเฟอร์ของเว็บเซิร์ฟเวอร์อื่น และเลือกเว็บเซิร์ฟเวอร์ที่มีเนื้อหาการร้องขอในข้อมูลสรุปบัฟเฟอร์ โดยที่เว็บเซิร์ฟเวอร์นั้นมีสภาพภาระงานไม่มาก โดยรีไดเร็กท์ไปยังเว็บเซิร์ฟเวอร์ที่พบในข้อมูลสรุปบัฟเฟอร์
3. ถ้าพบว่าไม่มีเว็บเซิร์ฟเวอร์ใดเป็นไปตามเงื่อนไขข้างต้น ก็จะพิจารณาจากสภาพภาระงานของตนเอง ถ้าตนเองมีสภาพภาระงานไม่มากก็จะให้บริการการร้องขอนั้นเอง
4. ถ้าตนเองมีสภาพภาระงานมากก็จะเลือกเว็บเซิร์ฟเวอร์อื่นที่มีสภาพภาระงานไม่มากเป็นผู้ให้บริการ โดยรีไดเร็กท์ไปยังเว็บเซิร์ฟเวอร์ที่พบในข้อมูลสรุปบัฟเฟอร์
5. แต่ถ้าไม่มีเว็บเซิร์ฟเวอร์ใดให้บริการได้ ตนเองก็จะเป็นผู้ให้บริการการร้องขอนั้นเอง

4.1.5 **รายละเอียดของมอดูลคำนวณข้อมูลสรุปบัฟเฟอร์**

มอดูลคำนวณข้อมูลสรุปบัฟเฟอร์นั้นจะแบ่งการทำงานเป็น 3 ส่วน คือ

1. นำข้อมูลเอกสารที่มีอยู่ในบัฟเฟอร์มาคำนวณเป็นข้อมูลสรุปบัฟเฟอร์เพื่อใช้แลกเปลี่ยนกันระหว่างเว็บเซิร์ฟเวอร์ โดยเอกสารที่จะนำมาคำนวณนั้น จะสนใจเฉพาะเอกสารที่มีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์เท่านั้น เพื่อเพิ่มความถูกต้องให้กับข้อมูลสรุปบัฟเฟอร์
2. ตรวจสอบว่ามีข้อมูลเอกสารอยู่ในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ใดหรือไม่ โดยนำเนื้อหาของการร้องขอไปตรวจสอบจากข้อมูลสรุปบัฟเฟอร์ ถ้าตรวจสอบพบ นั่นคือเว็บเซิร์ฟเวอร์นั้นน่าจะมีเอกสารที่ร้องขอนั้นอยู่ในบัฟเฟอร์
3. นำข้อมูลสรุปบัฟเฟอร์ที่ได้จากคำนวณ แลกเปลี่ยนกันระหว่างเว็บเซิร์ฟเวอร์ เพื่อใช้ข้อมูลนี้ในการตัดสินใจกระจายการร้องขอ

4.1.6 รายละเอียดของมอดูลตรวจสอบสภาพภาระงาน

มอดูลตรวจสอบสภาพภาระงาน จะทำหน้าที่ตรวจสอบว่า ณ ขณะนั้นเว็บเซิร์ฟเวอร์มีสภาพภาระงานเป็นอย่างไร และถ้ามีการเปลี่ยนแปลงสภาพภาระงานจากสภาพภาระงานมากเป็นสภาพภาระงานน้อย หรือเปลี่ยนจากสภาพภาระงานน้อยเป็นสภาพภาระงานมาก ก็จะแจ้งให้กับเว็บเซิร์ฟเวอร์ตัวอื่นทราบ เพื่อใช้เป็นข้อมูลในการตัดสินใจสำหรับการกระจายภาระงานต่อไป โดยจะตรวจสอบทุกช่วงเวลาที่กำหนด

4.2 เครื่องมือที่ใช้ในการทดสอบ

งานวิจัยนี้ได้จำลองการทำงานการทดลองกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหา โดยใช้ข้อมูลการใช้เว็บตามความเป็นจริงในการทดสอบ โดยเครื่องมือที่ใช้ก็คือ โปรแกรมจำลองเอนเอสทู เวอร์ชั่น ns-2.1b8a โดยได้นำโค้ดของโปรแกรมจำลองเอนเอสทูนั้นมาแก้ไขให้สามารถทำงานได้ตามแนวทางของงานวิทยานิพนธ์ ซึ่งได้แก้ไขในเรื่องต่าง ๆ ดังนี้

1. ไคลเอนต์จะต้องติดต่อไปยังดีเอนเอส เพื่อเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์หนึ่งในเว็บเซิร์ฟเวอร์ทั้งหมดโดยใช้วิธีวนรอบ แล้วไคลเอนต์จึงส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ตามเลขที่อยู่ไอพีที่ได้จากดีเอนเอส และเมื่อหมดช่วงอายุทีทีแอลที่ได้จากดีเอนเอส ไคลเอนต์ก็จะต้องสอบถามไปยังดีเอนเอส เพื่อเปลี่ยนยูอาร์แอลเป็นเลขที่อยู่ไอพีอีกครั้งก่อนที่จะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์
2. เว็บเซิร์ฟเวอร์สามารถตัดสินใจได้ว่าจะให้บริการการร้องขอเอง หรือจะกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่นโดยใช้ข้อมูลสรุปบัฟเฟอร์และข้อมูลสภาพภาระงานในการตัดสินใจ ซึ่งการกระจายการร้องขอทำได้โดยการรีไดเร็กท์ โดยจะกำหนดค่าคุกกี้ส่งไปพร้อมกับการร้องขอ เพื่อให้เว็บเซิร์ฟเวอร์ที่รับการรีไดเร็กท์รู้ว่าจะต้องเป็นผู้ให้บริการการร้องขอนั้น
3. ในการส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ ถ้าเป็นการร้องขอที่เกิดจากการรีไดเร็กท์ ไคลเอนต์ก็จะต้องส่งค่าคุกกี้ที่ได้จากเว็บเซิร์ฟเวอร์เดิม ไปยังเว็บเซิร์ฟเวอร์ใหม่ด้วย
4. เว็บเซิร์ฟเวอร์จะต้องมีการกำหนดขนาดบัฟเฟอร์ และมีการจัดการเอกสารในบัฟเฟอร์โดยใช้วิธีแทนที่เอกสารแบบแอลอาร์ยู
5. กรณีที่เป็นเอกสารใหม่ที่ยังไม่มีในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ จะต้องมีการจำลองเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์
6. เว็บเซิร์ฟเวอร์จะนำเอกสารในบัฟเฟอร์มาคำนวณสร้างเป็นข้อมูลสรุปบัฟเฟอร์ และแลกเปลี่ยนกับเว็บเซิร์ฟเวอร์อื่น

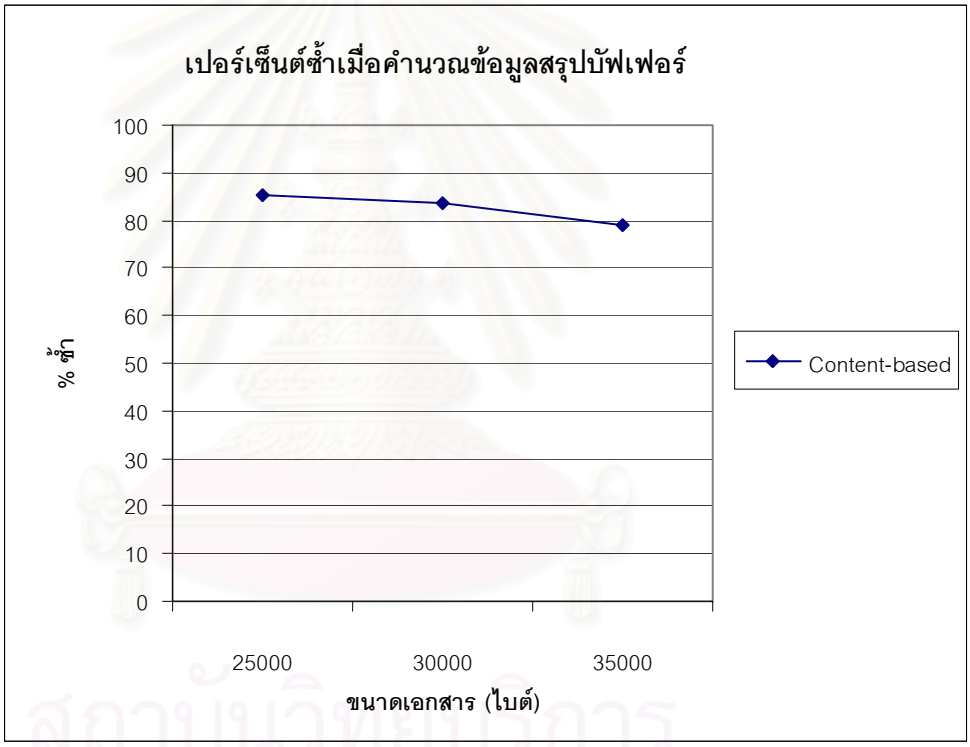
7. เว็บเซิร์ฟเวอร์จะต้องตรวจสอบสภาพภาระงานทุกช่วงเวลาที่กำหนด และถ้ามีการเปลี่ยนแปลงจากสภาพภาระงานน้อยเป็นสภาพภาระงานมาก หรือเปลี่ยนแปลงจากสภาพภาระงานมากเป็นสภาพภาระงานน้อย ก็จะต้องแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบเพื่อใช้เป็นข้อมูลในการตัดสินใจกระจายการร้องขอ

4.3 **สมมติฐานที่ใช้ในการทดสอบ**

ในการทดสอบ ได้กำหนดค่าต่าง ๆ เพื่อใช้ในการทดสอบ ดังนี้

1. วิทยานิพนธ์นี้ให้ความสำคัญและถือว่าเวลาในการอ่านข้อมูลเอกสารจากดิสก์ มีผลต่อการให้บริการของเว็บเซิร์ฟเวอร์ มากกว่าเวลาในเครือข่าย (Network Delay)
2. วิทยานิพนธ์นี้ใช้คุกกี้ในการพิจารณาถึงการกระจายการร้องขอ แต่ถ้าเว็บเบราว์เซอร์ปิดไม่ให้รับคุกกี้ ก็สามารถใช้วิธีการของ URL Rewriting แทน
3. วิทยานิพนธ์นี้ใช้คุกกี้ในการทดลองเท่านั้น
4. ลิงค์ที่เชื่อมต่อระหว่างเว็บเซิร์ฟเวอร์กับไคลเอนต์เป็นลิงค์แบบสื่อสารสองทาง (Duplex-link) มีแบนวิดท์เป็น 45 เมกะบิต (เทียบเท่ากับการเชื่อมต่อขนาด T3) โดยมีดีเลย์ 100 มิลลิวินาที (ได้จากการทดสอบโดยการ ping จากผู้ให้บริการอินเทอร์เน็ตในประเทศไทยไปยัง www.pantip.com และ www.thaisecondhand.com ได้ค่าเฉลี่ยเป็น 100 มิลลิวินาที)
5. เวลาในการอ่านข้อมูลเอกสารในดิสก์ ประกอบด้วย เวลาในการเข้าถึงข้อมูลในดิสก์ คือ 28 มิลลิวินาที เวลาในการถ่ายโอนข้อมูล คือ 410 มิลลิวินาทีสำหรับทุก ๆ 4 กิโลไบต์ และเวลา Rotation Latency เมื่อไฟล์มีขนาดมากกว่า 44 กิโลไบต์ จะใช้เวลา 14 มิลลิวินาทีสำหรับทุก ๆ 44 กิโลไบต์ ในวิทยานิพนธ์นี้ได้อ้างถึงเวลาในการอ่านข้อมูลเอกสารในดิสก์ตามงานวิจัย [12]
6. ความยาวแถวคอยที่เว็บเซิร์ฟเวอร์สามารถให้บริการการร้องขอพร้อม ๆ กัน ณ ขณะหนึ่ง เท่ากับ 1000 การร้องขอ
7. ค่าขีดแบ่งสภาพภาระงานที่ใช้พิจารณาว่าควรแจ้งสภาพภาระงานให้เว็บเซิร์ฟเวอร์อื่นทราบ มี 2 ค่า ได้แก่ ค่าขีดแบ่งมาก คือ 70 เปอร์เซ็นต์ของความยาวแถวคอย และค่าขีดแบ่งน้อย คือ 50 เปอร์เซ็นต์ของความยาวแถวคอย (จากการทดสอบเปลี่ยนแปลงค่าสภาพภาระงาน พบว่าได้ผลการทดลองที่ใกล้เคียงกัน จึงเลือกค่านี้มาใช้ในการทดลอง)
8. ค่าทีทีแอลที่ได้จากการเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์โดยดีเอนเอส คือ 20 นาที
9. การร้องขอแต่ละการร้องขอ จะมีการรีไดเร็กท์ได้อย่างมาก 1 ครั้ง

10. ข้อมูลสรุปบัฟเฟอร์ที่ได้จากการคำนวณมีขนาด 1,024 บิต
11. ขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ โดยจะพิจารณาว่าเว็บเซิร์ฟเวอร์ที่รับการร้องขอนั้น จะเป็นผู้ให้บริการการร้องขอนั้นเอง หรือจะกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่มีข้อมูลเอกสารนั้นในบัฟเฟอร์ โดยเมื่อทำการทดลองโดยกำหนดขนาดเอกสารขั้นต่ำสำหรับการรีไดเร็กท์เป็น 25000, 30000, 35000 ไบต์ จากนั้นจะพิจารณาว่าเมื่อนำเอกสารที่มีขนาดมากกว่าเอกสารที่กำหนด มาคำนวณเป็นข้อมูลสรุปบัฟเฟอร์แล้วทำให้ค่าในตำแหน่งของข้อมูลสรุปบัฟเฟอร์ซ้ำกันมากน้อยเท่าไร ซึ่งจากการทดลองได้ผลดังรูปที่ 4.4 วิทยานิพนธ์นี้จึงเลือกขนาดเอกสารที่ทำให้โอกาสซ้ำนี้ไม่เกิน 85 เปอร์เซ็นต์ ดังนั้นขนาดเอกสารที่เลือกใช้ในการทดลอง คือ 30000 ไบต์



รูปที่ 4.4 เปอร์เซ็นต์ซ้ำเมื่อคำนวณข้อมูลสรุปบัฟเฟอร์

4.4 ข้อมูลที่ใช้ในการทดสอบ

ข้อมูลที่ใช้ในการทดสอบ คือ ข้อมูลการใช้เว็บจริงจากเว็บไซท์ www.pantip.com ในช่วงวันที่ 14-20 กุมภาพันธ์ พ.ศ. 2545 และข้อมูลการใช้เว็บจริงจากเว็บไซท์ www.thaisecondhand.com ในช่วงวันที่ 17 มกราคม - 16 กุมภาพันธ์ พ.ศ. 2545 โดยข้อมูลการใช้เว็บมีลักษณะข้อมูล ดังตารางที่ 4.1

ตารางที่ 4.1 ลักษณะข้อมูลการใช้เว็บ

	ข้อมูล www.pantip.com	ข้อมูล www.thaisecondhand.com
จำนวนคำร้องขอ	3,058,709 การร้องขอ	2,842,422 การร้องขอ
จำนวนผู้ใช้งาน	27,861 คน	11,158 คน
จำนวนเอกสารทั้งหมด	107,601 เอกสาร	10,298 เอกสาร
จำนวนเอกสารที่มีขนาดมากกว่า 30000 ไบต์	26,947 เอกสาร	1,454 เอกสาร

4.5 **สภาพแวดล้อมที่ใช้ในการทดสอบ**

ในงานวิจัยนี้ใช้คอมพิวเตอร์ 2 เครื่องในการทดสอบจำลองระบบในรูปแบบต่าง ๆ โดยทั้งสองเครื่องจะใช้ในการจำลองที่เหมือนกัน ซึ่งแต่ละเครื่องมีรายละเอียดดังนี้

1. เครื่องแรก
   - คอมพิวเตอร์แบบพีซี Pentium III 1 กิกะเฮิรตซ์
   - หน่วยความจำ 2 กิกะไบต์
   - ฮาร์ดดิสก์ 20 กิกะไบต์
   - ใช้ระบบปฏิบัติการ Linux RedHat 7.2
2. เครื่องที่สอง
   - คอมพิวเตอร์แบบพีซี Pentium4 2.1 กิกะเฮิรตซ์
   - หน่วยความจำ 512 เมกะไบต์
   - ฮาร์ดดิสก์ 20 กิกะไบต์
   - ใช้ระบบปฏิบัติการ Linux RedHat 7.2

4.6 **วิธีการทดสอบ**

การทดสอบทำโดยเขียนสคริปต์ไฟล์สำหรับจำลองการกระจายการร้องขอ ซึ่งในแต่ละการจำลองจะประกอบด้วยเว็บเซิร์ฟเวอร์ ไคลเอนต์ และดีเอนเอส โดยทำการทดสอบโดยการเปลี่ยนแปลงจำนวนเว็บเซิร์ฟเวอร์ที่ให้บริการ จำนวนไคลเอนต์ ขนาดของบัฟเฟอร์ ช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์

4.7 **ผลการทดสอบ**

การทดสอบ ได้ทำการทดสอบต่าง ๆ ดังนี้

- การทดสอบประสิทธิภาพโดยการเพิ่มจำนวนเว็บเซิร์ฟเวอร์ที่ให้บริการการร้องขอ เพื่อทดสอบว่าเมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น ส่งผลให้บริการของเว็บเซิร์ฟเวอร์ดีขึ้นหรือไม่
- การทดสอบประสิทธิภาพโดยการเพิ่มจำนวนไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน เพื่อทดสอบว่าเมื่อมีจำนวนการร้องขอส่งเข้าไปยังเว็บเซิร์ฟเวอร์มากขึ้น เว็บเซิร์ฟเวอร์ยังสามารถให้บริการได้ดีหรือไม่
- การทดสอบประสิทธิภาพโดยการเพิ่มขนาดบัฟเฟอร์ เพื่อทดสอบว่าเมื่อขนาดบัฟเฟอร์เพิ่มขึ้น มีเนื้อที่ในการจัดเอกสารมากขึ้น มีผลต่อเวลาในการให้บริการของเว็บเซิร์ฟเวอร์หรือไม่
- การทดสอบประสิทธิภาพโดยการปรับช่วงเวลาในการแลกเปลี่ยนช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ เพื่อทดสอบว่าช่วงเวลาในการแลกเปลี่ยน มีผลต่อเวลาในการให้บริการมากน้อยเพียงใด
- การทดสอบประสิทธิภาพโดยการกำหนดดีเลย์ในการเชื่อมต่อระหว่างเว็บเซิร์ฟเวอร์กับไคลเอนต์เป็น 500 มิลลิวินาที เนื่องจากว่าการใช้การใช้ดีเลย์ 100 มิลลิวินาทีนั้นสำหรับการติดต่อไปยังเว็บเซิร์ฟเวอร์ภายในประเทศ แต่การติดต่อไปยังเว็บเซิร์ฟเวอร์ในต่างประเทศนั้นมีดีเลย์ที่มากกว่า จึงได้ทำการทดลอง ping จากผู้ให้บริการอินเทอร์เน็ตในประเทศไทยไปยัง www.yahoo.com ได้ค่าดีเลย์ประมาณ 500 มิลลิวินาที และนำค่าดีเลย์นี้มาใช้ในการทดสอบ เพื่อทดสอบดูว่าเมื่อดีเลย์มากขึ้นการเชื่อมต่อมากขึ้น การรีไดเร็กท์จะมีผลกระทบต่อบริการของเว็บเซิร์ฟเวอร์หรือไม่อย่างไร
- การทดสอบประสิทธิภาพโดยใช้ข้อมูลของ www.thaisecondhand.com เนื่องจากว่าข้อมูลของ www.pantip.com นั้น ลักษณะเอกสารมีความหลากหลายเหมือนกับข้อมูลเว็บไซท์ทั่วไป จึงทำการทดสอบโดยใช้ข้อมูลของ www.thaisecondhand.com ซึ่งเป็นเว็บไซท์เกี่ยวกับการซื้อขายสินค้า ซึ่งขนาดเอกสารส่วนมากจะมีขนาดเล็ก เพื่อทดสอบดูว่าลักษณะข้อมูลอย่างนี้จะส่งผลต่อบริการอย่างไรบ้าง การกระจายการร้องขอวิธีการของวิทยานิพนธ์นี้เหมาะสมกับข้อมูลลักษณะนี้หรือไม่

4.7.1 การทดสอบประสิทธิภาพโดยการเพิ่มจำนวนเว็บเซิร์ฟเวอร์

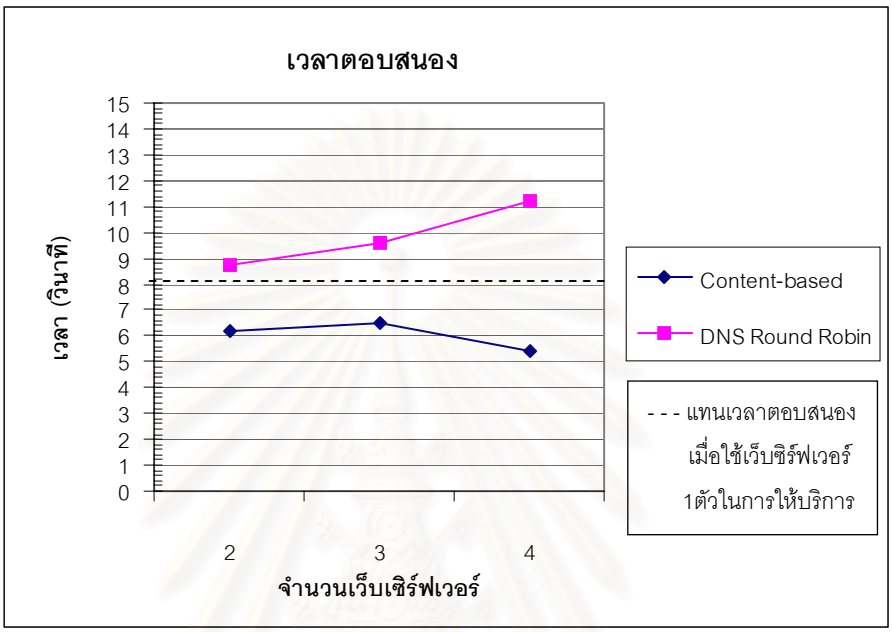
การทดลองนี้ กำหนดให้มีไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน 100 ไคลเอนต์ แต่ละเว็บเซิร์ฟเวอร์มีบัฟเฟอร์ขนาด 512 เมกะไบต์ มีการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ทุก ๆ 1 นาที โดยการทดลองนี้กำหนดดีเลย์ในการเชื่อมต่อระหว่างไคลเอนต์และเว็บเซิร์ฟเวอร์เป็น 100 มิลลิวินาที โดยจะทำการเปรียบเทียบกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหา กับการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส โดยที่จะไม่เปรียบเทียบกับการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยพิจารณาจากสภาพภาระงาน เนื่องจากการทดลองนี้ไม่เกิดเหตุการณ์ที่เว็บเซิร์ฟเวอร์มีสภาพภาระงานมาก ทำให้การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยพิจารณาจากสภาพภาระงานไม่มีการกระจายการร้องขอของเว็บเซิร์ฟเวอร์เกิดขึ้น ผลที่ได้จากการทดลองจึงเทียบเท่ากับการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส

จากการทดสอบโดยการเพิ่มจำนวนเว็บเซิร์ฟเวอร์ที่ให้บริการ พบว่าเมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น ระบบที่กระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น เว็บเซิร์ฟเวอร์สามารถให้บริการได้เร็วกว่ามีเวลาตอบสนองที่ดีกว่าระบบที่กระจายการร้องขอโดยวิธีวนรอบของดีเอนเอสดังรูปที่ 4.5 นั่นเป็นเพราะในการให้บริการการร้องขอเอกสารที่มีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ เว็บเซิร์ฟเวอร์ที่รับการร้องขอจะพยายามส่งการร้องขอนั้นไปยังเว็บเซิร์ฟเวอร์ที่น่าจะมีข้อมูลเอกสารนั้นในบัฟเฟอร์ ทำให้ฮิตเรตของระบบเพิ่มขึ้น ดังรูปที่ 4.6 เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลง และเวลารอคอยเอกสารลดลง ดังนั้น ระบบที่ใช้การกระจายร้องขอด้วยวิธีนี้จึงมีเวลาตอบสนองที่ดีกว่า

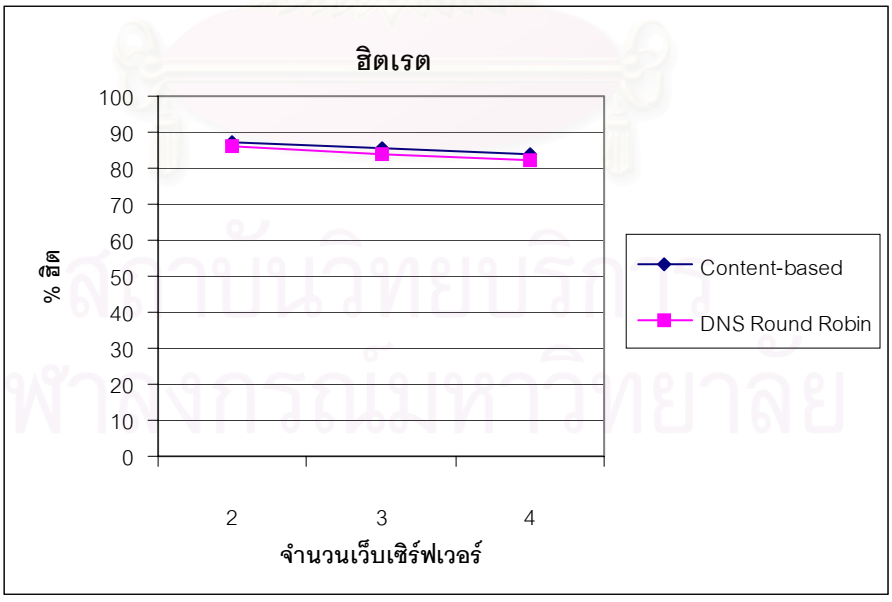
ในระบบที่ใช้การกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส จะพบว่าเมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองมีแนวโน้มเพิ่มขึ้น ทำให้ดูเหมือนว่าไม่มีประโยชน์เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์ แต่การเพิ่มจำนวนเว็บเซิร์ฟเวอร์นี้ทำให้เว็บเซิร์ฟเวอร์สามารถให้บริการการร้องขอของไคลเอนต์ได้มากขึ้น โดยเมื่อใช้เว็บเซิร์ฟเวอร์เพียงตัวเดียวในการให้บริการนั้นพบว่ามีการเกิดการไม่รับการร้องขอมากถึง 7.85 เปอร์เซ็นต์

ในระบบที่ใช้การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองมีแนวโน้มลดลง เนื่องจากว่าการกระจายการร้องขอวิธีนี้ บัฟเฟอร์ของเว็บเซิร์ฟเวอร์นั้นเสมือนเท่ากับบัฟเฟอร์ของทั้งระบบ เพราะสามารถใช้ข้อมูลเอกสารที่จัดเก็บไว้ในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์อื่นได้ ดังนั้น เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น ขนาดบัฟเฟอร์เสมือนว่าใหญ่ขึ้นด้วย เมื่อบัฟเฟอร์มีขนาดใหญ่ขึ้นจึงทำให้สามารถจัดเก็บเอกสารได้มากขึ้น และตรวจสอบพบเอกสารจาก

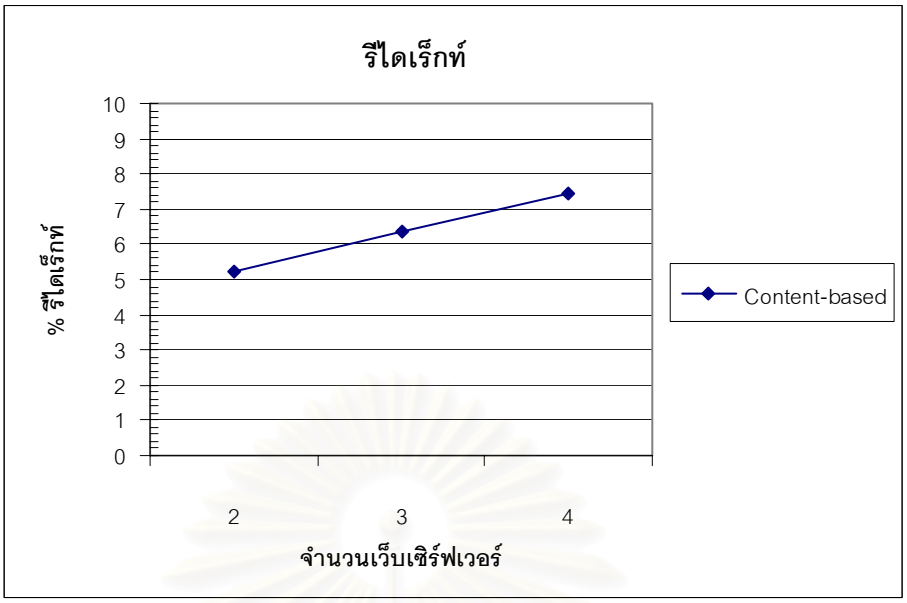
ข้อมูลสรุปบัฟเฟอร์ได้มากขึ้น จึงมีการรีไดเร็กท์มากขึ้น ดังรูปที่ 4.7 และโดยที่รีไดเร็กท์ไปแล้วก็จะมีโอกาสพบเอกสารในบัฟเฟอร์มากขึ้นด้วย ดังรูปที่ 4.8 ซึ่งเป็นเพราะบัฟเฟอร์ที่มีขนาดใหญ่ขึ้น การแทนที่เอกสารลดลง ทำให้ค้นพบเอกสารในบัฟเฟอร์ได้มากขึ้น ซึ่งช่วยลดเวลารอคอยเอกสาร ดังนั้น เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองจึงลดลง



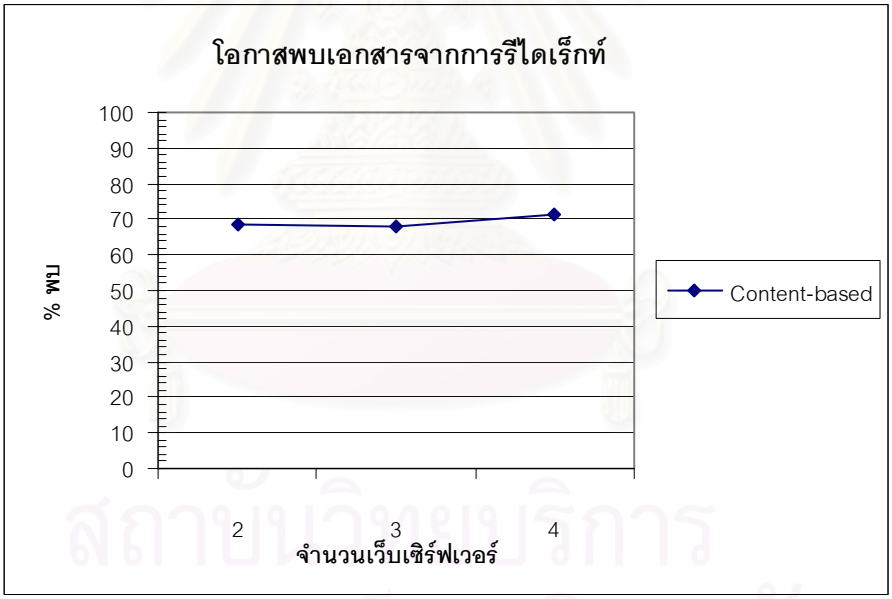
รูปที่ 4.5 เวลาตอบสนอง เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์



รูปที่ 4.6 ฮิตเรต เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์

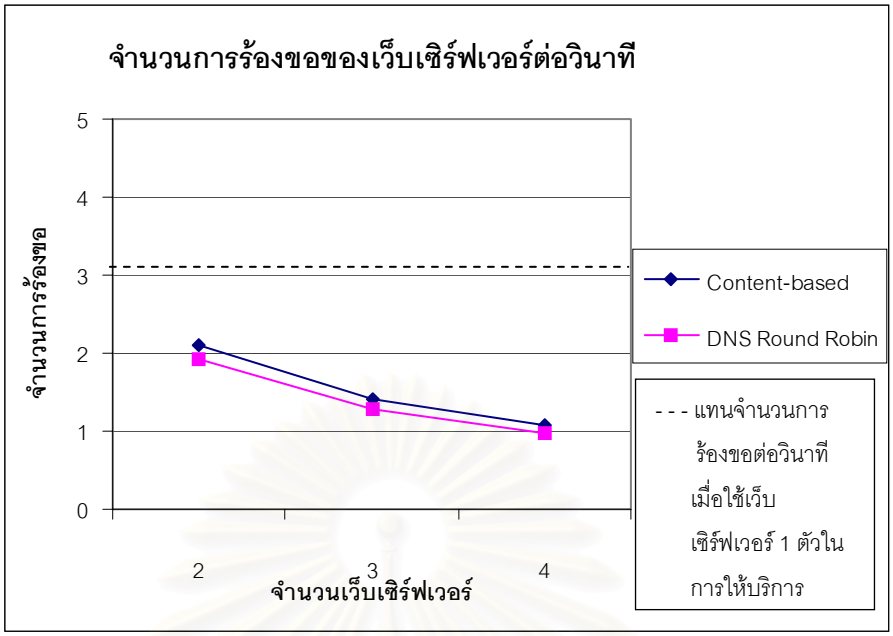


รูปที่ 4.7 เปอร์เซ็นต์รีไดเร็กท์ เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์



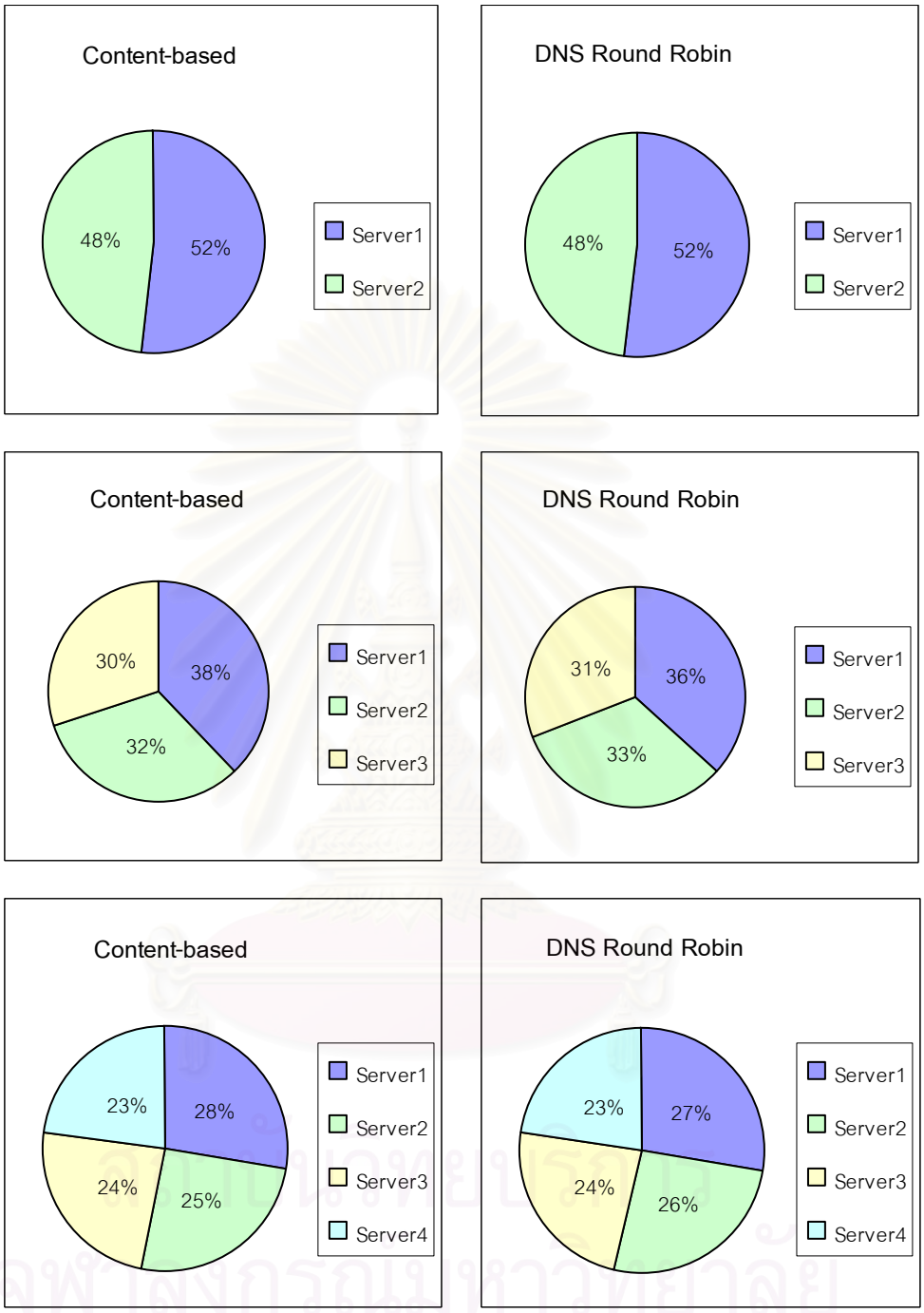
รูปที่ 4.8 โอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อเพิ่มจำนวนเว็บเซิร์ฟเวอร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.9 จำนวนการร้องขอของเว็บเซิร์ฟเวอร์ต่อวินาที

นอกจากนี้ ในการทดลองนี้กำหนดจำนวนไคลเอนต์ 100 ไคลเอนต์ ร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์ตามข้อมูลการใช้เว็บ เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น ใน 1 วินาทีเว็บเซิร์ฟเวอร์ให้บริการจำนวนการร้องขอน้อยลง ดังรูปที่ 4.9 แต่การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา สามารถให้บริการการร้องขอต่อวินาทีได้มากกว่าการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส อีกทั้งถ้าเปรียบเทียบการกระจายภาระงานระหว่างเว็บเซิร์ฟเวอร์แต่ละตัว โดยใช้วิธีการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอสเป็นหลักในการเปรียบเทียบ จะพบว่าการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา ทำให้มีการกระจายภาระงานจากเว็บเซิร์ฟเวอร์หนึ่งไปยังอีกเว็บเซิร์ฟเวอร์หนึ่ง ดังรูปที่ 4.10 แต่ไม่เห็นความแตกต่างมากนัก เนื่องจากว่าในการทดสอบนี้ จะกระจายการร้องขอของไคลเอนต์แต่ละไคลเอนต์โดยใช้วิธีวนรอบของดีเอนเอส ไม่ได้มีการจำลองเป็นลักษณะของไคลเอนต์จากเครือข่ายเดียวกัน ทำให้การทดลองนี้มีการกระจายการร้องขอไปยังทุก ๆ เว็บเซิร์ฟเวอร์ใกล้เคียงกัน

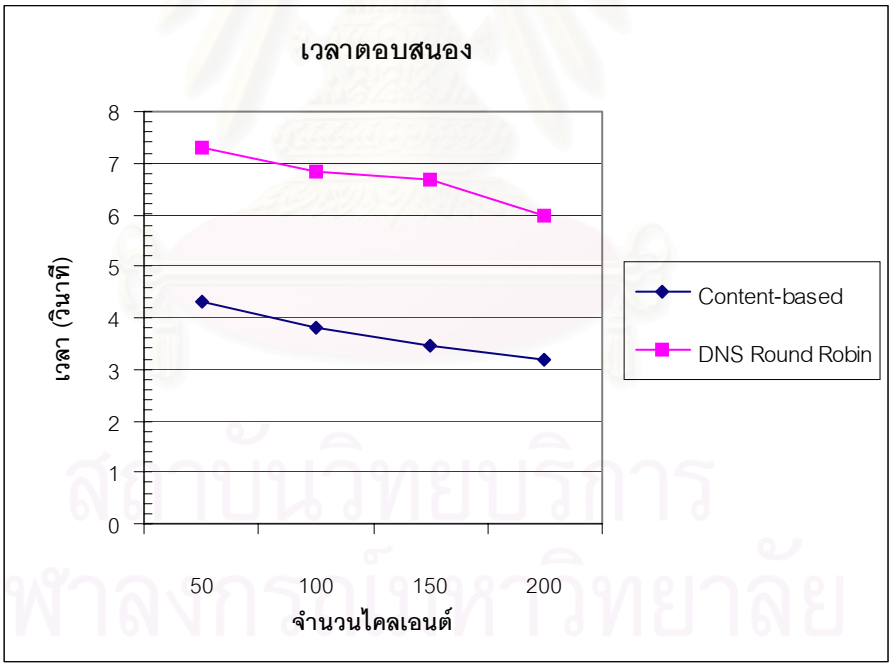


4.10 เปอร์เซ็นต์การใช้งานของเว็บเซิร์ฟเวอร์

4.2.2 การทดสอบประสิทธิภาพโดยเพิ่มจำนวนไคลเอนต์

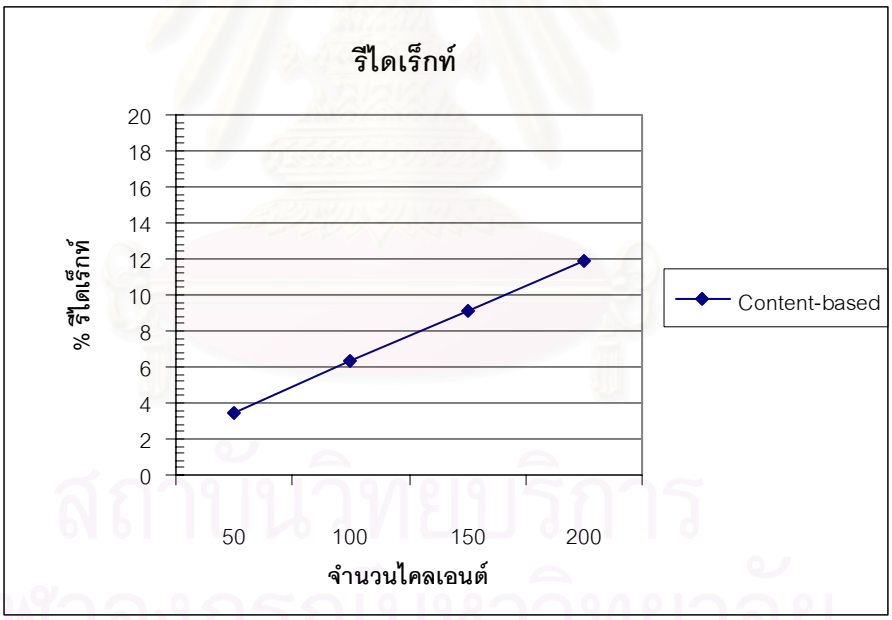
การทดลองนี้ กำหนดให้มีเว็บเซิร์ฟเวอร์ที่ให้บริการการร้องขอจำนวน 3 เว็บเซิร์ฟเวอร์ โดยที่แต่ละเว็บเซิร์ฟเวอร์มีบัฟเฟอร์ขนาด 512 เมกะไบต์ มีการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ ทุก ๆ 1 นาที โดยเมื่อจำนวนไคลเอนต์เพิ่มขึ้น เอกสารที่ร้องขอก็จะมีความหลากหลายมากขึ้น แต่ก็มีโอกาสร้องขอเอกสารซ้ำกันมากขึ้นด้วยเช่นกัน จึงมีโอกาสพบในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์มากขึ้น

จากการทดลองโดยเพิ่มจำนวนไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน เมื่อจำนวนไคลเอนต์เพิ่มขึ้น จำนวนการร้องขอเพิ่มขึ้น จำนวนเอกสารที่ร้องขอก็มีมากขึ้น เวลาในการเข้าถึงข้อมูลเอกสารจึงเพิ่มขึ้น แต่การกระจายการร้องขอของเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้นใช้เวลาเข้าถึงข้อมูลเอกสารในดิสก์น้อยกว่าการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส ซึ่งเป็นเพราะในการกระจายการร้องขอโดยวิธีแรกนั้นจะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่น่าจะมีข้อมูลเอกสารนั้นในบัฟเฟอร์ เพื่อลดเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ ลดเวลารอคอยเอกสาร ไคลเอนต์จะได้รับเอกสารเร็วขึ้น จึงทำให้การกระจายการร้องขอวิธีนี้มีเวลาตอบสนองน้อยกว่า ดังรูปที่ 4.11

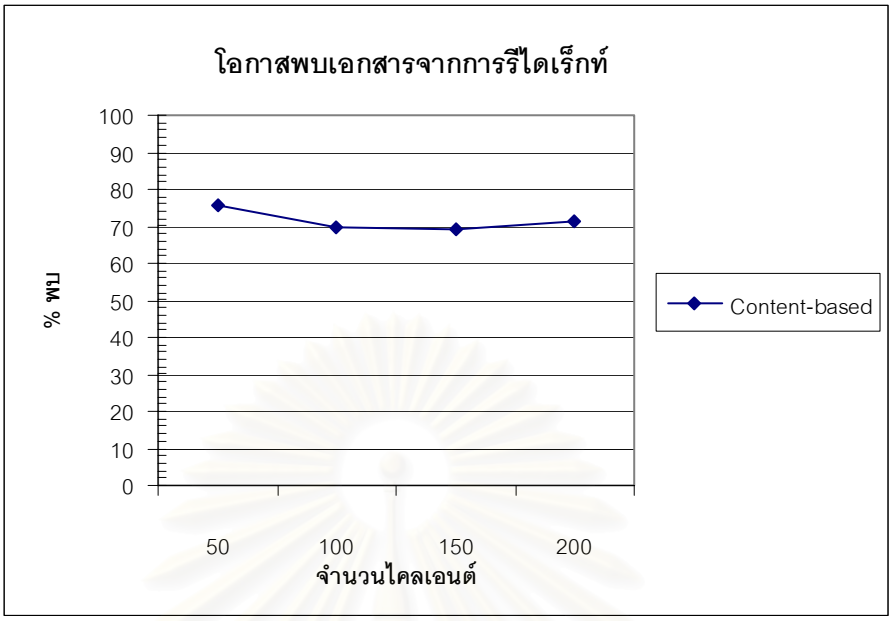


รูปที่ 4.11 เวลาตอบสนอง เมื่อเพิ่มจำนวนไคลเอนต์

จากรูปที่ 4.11 จะพบว่าเมื่อจำนวนไคลเอนต์เพิ่มขึ้น เวลาตอบสนองมีค่าลดลง นั่นเป็นเพราะเมื่อจำนวนไคลเอนต์มากขึ้น เอกสารที่ร้องขอมีโอกาสซ้ำกันมากขึ้น จึงมีโอกาสพบเอกสารในบัฟเฟอร์มากขึ้น เว็บเซิร์ฟเวอร์สามารถนำข้อมูลเอกสารในบัฟเฟอร์ตอบการร้องขอได้ทันที ทำให้เวลารอคอยเอกสารลดลง เวลาตอบสนองจึงลดลง แต่การกระจายการร้องขอของเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา มีเวลาตอบสนองน้อยกว่าการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส ซึ่งเป็นเพราะในการกระจายการร้องขอโดยวิธีแรกนั้นมีการส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่มีข้อมูลเอกสารนั้น ซึ่งช่วยลดเวลาในการเข้าถึงเอกสาร ทำให้การกระจายการร้องขอด้วยวิธีนี้มีเวลาตอบสนองที่ดีกว่า โดยจะพบว่าเมื่อจำนวนไคลเอนต์เพิ่มขึ้น เว็บเซิร์ฟเวอร์ตรวจสอบพบข้อมูลเอกสารจากข้อมูลสรุปบัฟเฟอร์มากขึ้น การรีไดเร็กท์จึงมากขึ้น ดังรูปที่ 4.12 โดยที่การรีไดเร็กท์นี้มีโอกาสพบเอกสารในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ที่รับการรีไดเร็กท์ใกล้เคียงกัน ดังรูปที่ 4.13 ซึ่งการรีไดเร็กท์แล้วพบเอกสารในบัฟเฟอร์นี้ช่วยเพิ่มฮิตเรตให้กับระบบ ทำให้เวลารอคอยเอกสารน้อยลง เวลาตอบสนองจึงลดลงด้วย ดังนั้น เมื่อจำนวนไคลเอนต์เพิ่มขึ้น เวลาตอบสนองจึงลดลง



รูปที่ 4.12 เปอร์เซ็นต์รีไดเร็กท์ เมื่อเพิ่มจำนวนไคลเอนต์

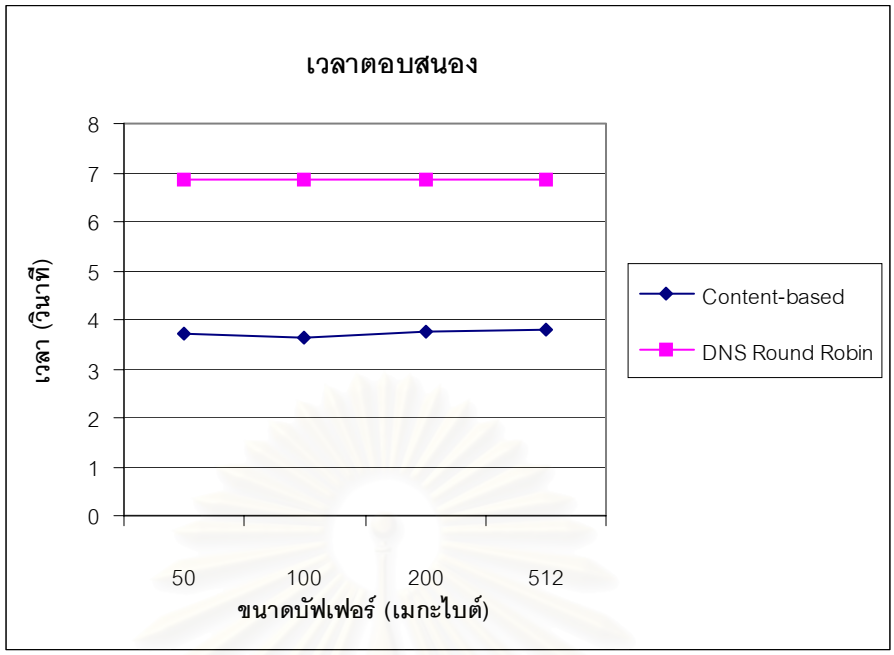


รูปที่ 4.13 โอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อเพิ่มจำนวนไคลเอนต์

4.2.3 การทดสอบประสิทธิภาพโดยการเพิ่มขนาดบัฟเฟอร์

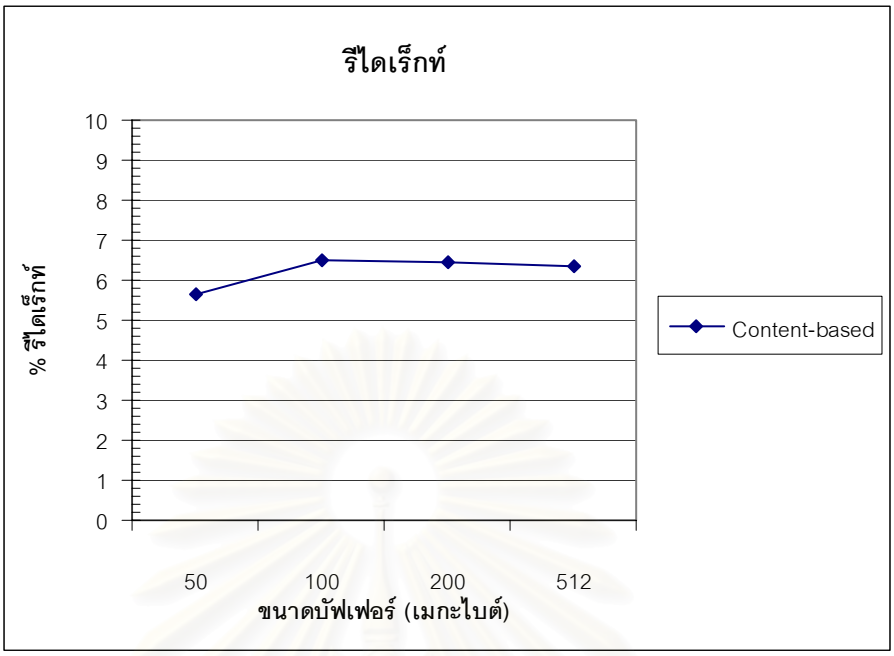
การทดลองนี้ กำหนดให้มีไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน 100 ไคลเอนต์ มีเว็บเซิร์ฟเวอร์ที่ให้บริการ 3 เว็บเซิร์ฟเวอร์ โดยที่แต่ละเว็บเซิร์ฟเวอร์มีบัฟเฟอร์ขนาด 512 เมกะไบต์ มีการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ทุก ๆ 1 นาที

จากการทดสอบโดยการเพิ่มขนาดบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ เมื่อบัฟเฟอร์มีขนาดเพิ่มขึ้น บัฟเฟอร์ก็จะสามารถจัดเก็บข้อมูลเอกสารได้มากขึ้น การแทนที่เอกสารในบัฟเฟอร์ลดลง สามารถค้นพบเอกสารในบัฟเฟอร์ได้มากขึ้น จึงทำให้ฮิตเรตของระบบเพิ่มขึ้น และเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลง โดยระบบที่กระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น จะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่มีข้อมูลเอกสารนั้นในบัฟเฟอร์ ซึ่งช่วยลดเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ จึงทำให้การกระจายการร้องขอด้วยวิธีนี้ ใช้เวลารอคอยเอกสารน้อยกว่าการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส และมีเวลารอคอยเอกสารน้อยกว่า จึงมีเวลาตอบสนองดีกว่าการกระจายการร้องขอด้วยวิธีที่สอง

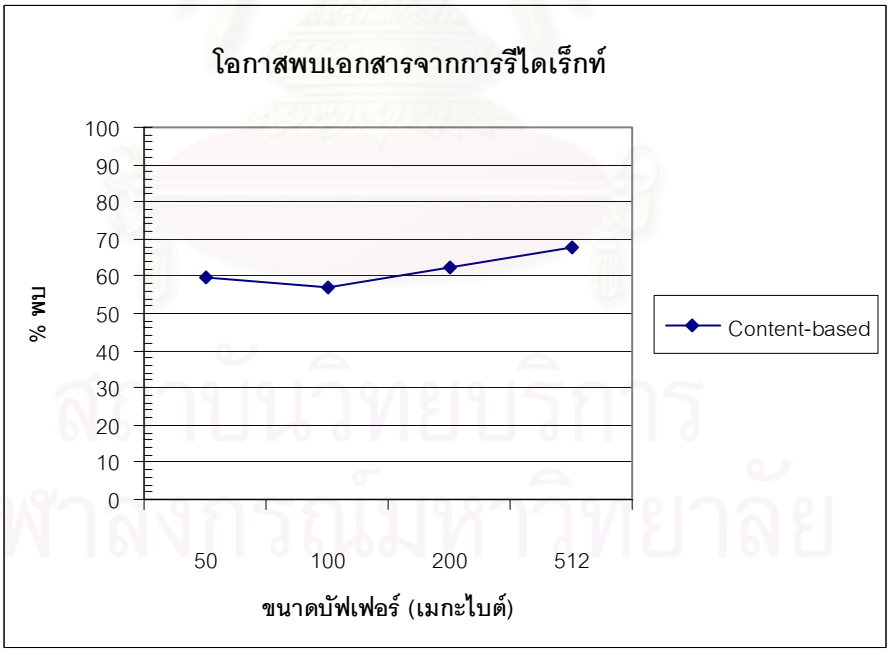


รูปที่ 4.14 เวลาตอบสนอง เมื่อเพิ่มขนาดบัฟเฟอร์

เมื่อบัฟเฟอร์มีขนาดเพิ่มขึ้น เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลง น่าจะส่งผลให้เวลาตอบสนองลดลงด้วย แต่จากรูปที่ 4.14 จะพบว่าเมื่อบัฟเฟอร์มีขนาดเพิ่มขึ้น เวลาตอบสนองมีค่าใกล้เคียงกัน โดยที่เมื่อบัฟเฟอร์มีขนาดเพิ่มขึ้นจาก 50 เมกะไบต์เป็น 100 เมกะไบต์ มีการรีไดเร็กท์เพิ่มขึ้น ดังรูปที่ 4.15 โดยที่รีไดเร็กท์ไปแล้วมีโอกาสพบเอกสารในบัฟเฟอร์ลดลง ดังรูปที่ 4.16 ซึ่งเป็นเพราะบัฟเฟอร์มีขนาดไม่มากนัก เมื่อบัฟเฟอร์มีขนาด 50 เมกะไบต์ จำนวนเอกสารที่นำมาคำนวณข้อมูลสรุปบัฟเฟอร์นั้นมีไม่มากเท่าเมื่อบัฟเฟอร์มีขนาด 100 เมกกะไบต์ ความผิดพลาดจึงน้อยกว่าเมื่อใช้บัฟเฟอร์ขนาด 100 เมกะไบต์ และเมื่อบัฟเฟอร์มีขนาดเพิ่มขึ้นจาก 100 เมกะไบต์เป็น 200 เมกะไบต์ และ 512 เมกะไบต์นั้น มีการรีไดเร็กท์ใกล้เคียงกัน แต่มีโอกาสพบเอกสารในบัฟเฟอร์เพิ่มขึ้น เนื่องจากบัฟเฟอร์มีขนาดใหญ่ สามารถจัดเก็บข้อมูลเอกสารได้มาก จึงมีโอกาสพบเอกสารในบัฟเฟอร์มากขึ้น แต่อย่างไรก็ตามจำนวนการร้องขอที่ใช้ในการทดลองอาจจะไม่มากนัก จึงไม่มีผลต่อเวลาตอบสนองมากนักเมื่อเพิ่มขนาดบัฟเฟอร์



รูปที่ 4.15 เปอร์เซ็นต์รีไดเร็กท์ เมื่อเพิ่มขนาดบัฟเฟอร์

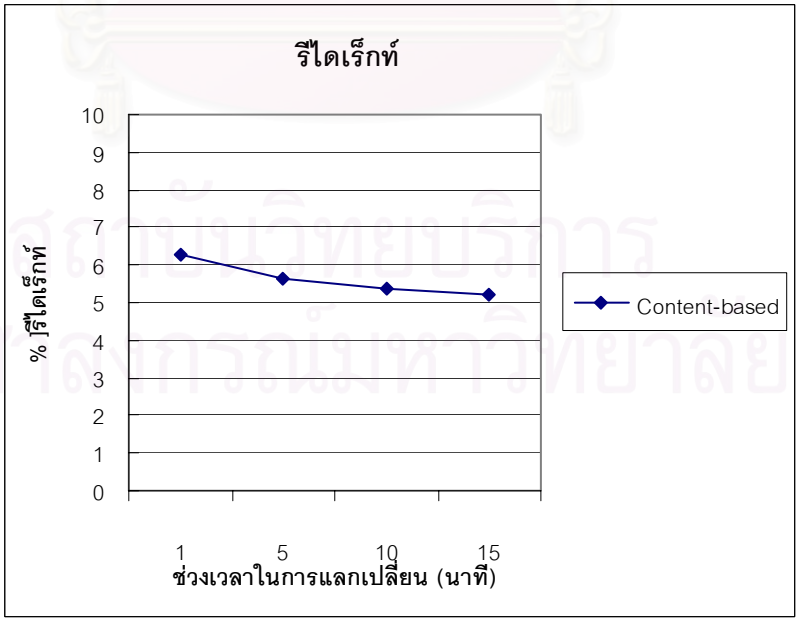


รูปที่ 4.16 โอกาสพบเอกสารจากการรีไดเร็กท์เมื่อเพิ่มขนาดบัฟเฟอร์

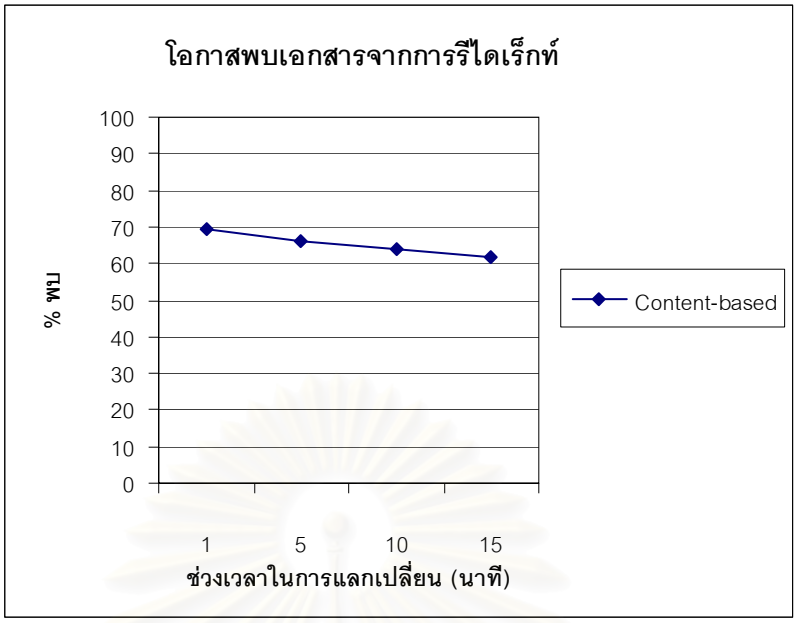
4.2.4 การทดสอบประสิทธิภาพโดยปรับช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์

การทดลองนี้กำหนดให้มีไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน 100 ไคลเอนต์ มีเว็บเซิร์ฟเวอร์ที่ให้บริการ 3 เว็บเซิร์ฟเวอร์ โดยที่แต่ละเว็บเซิร์ฟเวอร์มีบัฟเฟอร์ขนาด 512 เมกะไบต์

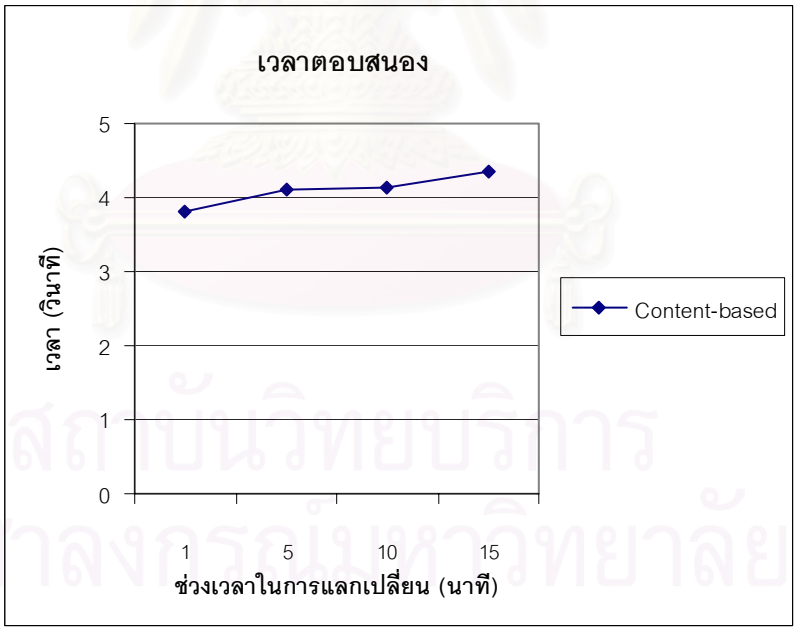
จากการทดสอบโดยการปรับเปลี่ยนช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ พบว่า เมื่อช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ยิ่งเพิ่มขึ้น การตรวจสอบพบเอกสารจากข้อมูลสรุปบัฟเฟอร์ว่าน่าจะมีเอกสารอยู่ในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์อื่นจะลดลง การรีไดเร็กท์ไปยังเว็บเซิร์ฟเวอร์อื่นจะลดลงดังรูปที่ 4.17 และโอกาสที่จะพบเอกสารหลังการรีไดเร็กท์จะลดลงด้วย ดังรูปที่ 4.18 ซึ่งเป็นเพราะเมื่อช่วงเวลาในการแลกเปลี่ยนเพิ่มขึ้น ความทันสมัยของข้อมูลสรุปบัฟเฟอร์ลดลง ทำให้เกิดความผิดพลาด คือ เมื่อตรวจพบเอกสารจากข้อมูลสรุปบัฟเฟอร์แล้วรีไดเร็กท์ไปยังเว็บเซิร์ฟเวอร์ที่น่าจะมีข้อมูลเอกสารนั้น แต่ไม่พบเอกสารในบัฟเฟอร์ เพราะข้อมูลเอกสารถูกแทนที่ไปแล้ว จึงทำให้เสียเวลาในช่วงของการรีไดเร็กท์ ทั้งที่เว็บเซิร์ฟเวอร์แรกที่เป็นผู้รับการร้องขอนั้นควรจะเป็นผู้ให้บริการโดยการเข้าถึงข้อมูลเอกสารในดิสก์ แต่เมื่อมีการรีไดเร็กท์เวลาในการรอคอยจึงเพิ่มขึ้นในส่วนของเวลารีไดเร็กท์ ทำให้เวลาตอบสนองเพิ่มขึ้น หรือในกรณีที่ตรวจสอบจากข้อมูลสรุปบัฟเฟอร์แล้วไม่พบว่ามีข้อมูลเอกสารในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์อื่น ทั้งที่เว็บเซิร์ฟเวอร์อื่นมีข้อมูลเอกสารในบัฟเฟอร์ จึงต้องอ่านข้อมูลเอกสารจากดิสก์เอง เวลารอคอยเอกสารเพิ่มขึ้น เวลาตอบสนองจึงเพิ่มขึ้น นั่นแสดงให้เห็นว่า ช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์เพิ่มขึ้น ความทันสมัยของข้อมูลจะลดลง เวลาตอบสนองก็จะเพิ่มขึ้น ดังรูปที่ 4.19



รูปที่ 4.17 เปอร์เซ็นต์รีไดเร็กท์ เมื่อปรับช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์



รูปที่ 4.18 โอกาสพบเอกสารจากการรีไดเร็กท์
เมื่อปรับช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์



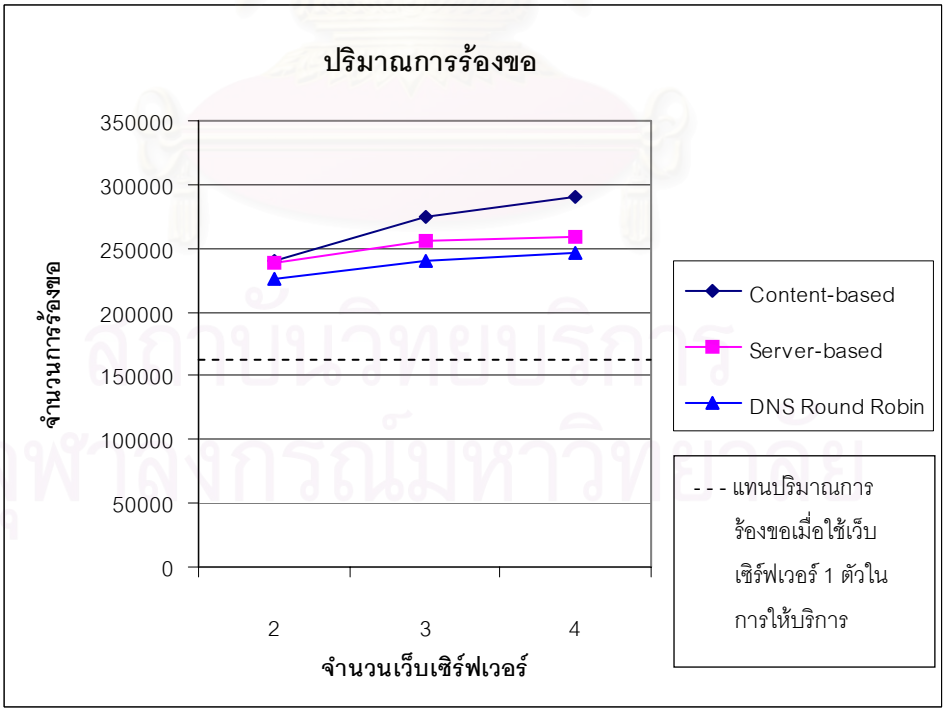
รูปที่ 4.19 เวลาตอบสนอง เมื่อปรับช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

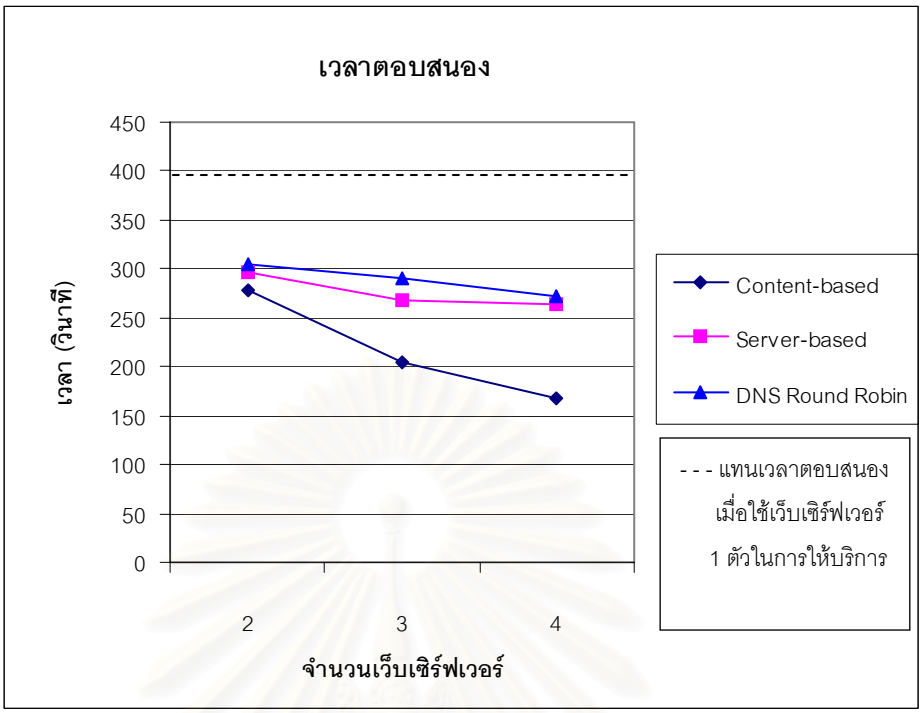
4.2.5 การทดสอบประสิทธิภาพเมื่อกำหนดค่าดีเลย์ 500 มิลลิวินาที

การทดลองนี้กำหนดให้มีไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน 100 ไคลเอนต์ มีเว็บเซิร์ฟเวอร์ที่ให้บริการ 3 เว็บเซิร์ฟเวอร์ โดยที่แต่ละเว็บเซิร์ฟเวอร์มีบัฟเฟอร์ขนาด 512 เมกะไบต์ มีการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ทุก ๆ 1 นาที โดยใช้ดีเลย์ 500 มิลลิวินาที (ทดสอบจากการ ping จากผู้ให้บริการอินเทอร์เน็ตในประเทศไทยไปยัง www.yahoo.com ได้ค่าเฉลี่ยเป็น 500 มิลลิวินาที) เพื่อเปรียบเทียบประสิทธิภาพของระบบที่ใช้การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหา การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยพิจารณาจากข้อมูลสภาพภาระงาน และการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส

จากการทดลองโดยการกำหนดค่าดีเลย์ 500 มิลลิวินาที พบว่าการให้บริการของเว็บเซิร์ฟเวอร์นั้น ระบบที่ใช้การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหา เว็บเซิร์ฟเวอร์สามารถให้บริการการร้องขอได้มากที่สุด รองลงมาคือ การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยพิจารณาจากข้อมูลสภาพภาระงาน และการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส เว็บเซิร์ฟเวอร์สามารถให้บริการการร้องขอได้น้อยที่สุด ดังรูปที่ 4.20



รูปที่ 4.20 ปริมาณการร้องขอ เมื่อใช้ดีเลย์ 500 มิลลิวินาที

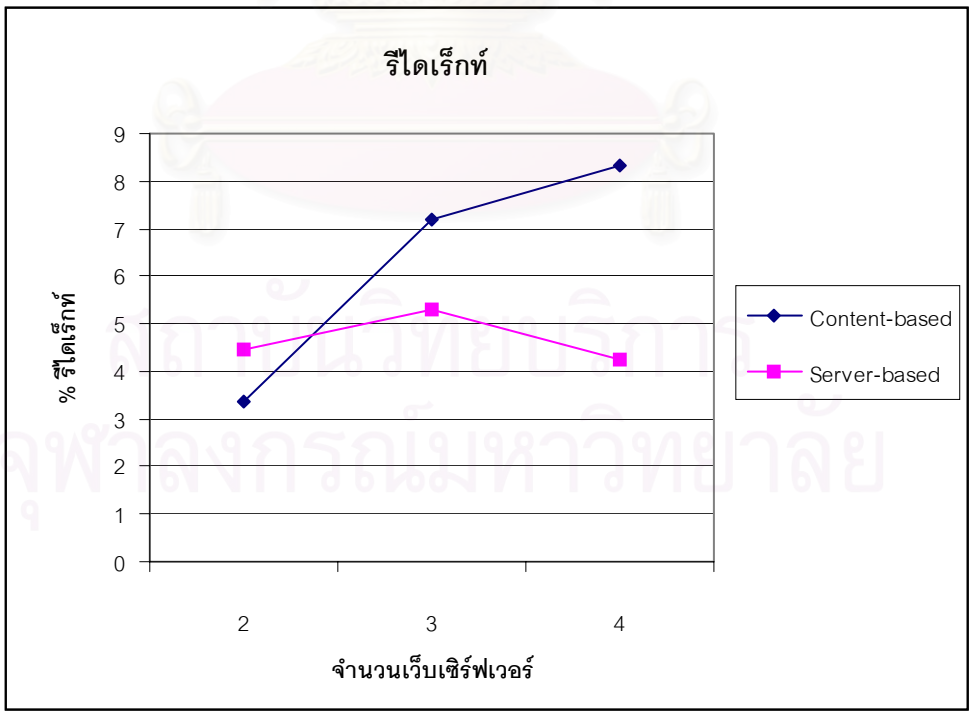


รูปที่ 4.21 เวลาตอบสนอง เมื่อใช้ดีเลย์ 500 มิลลิวินาที

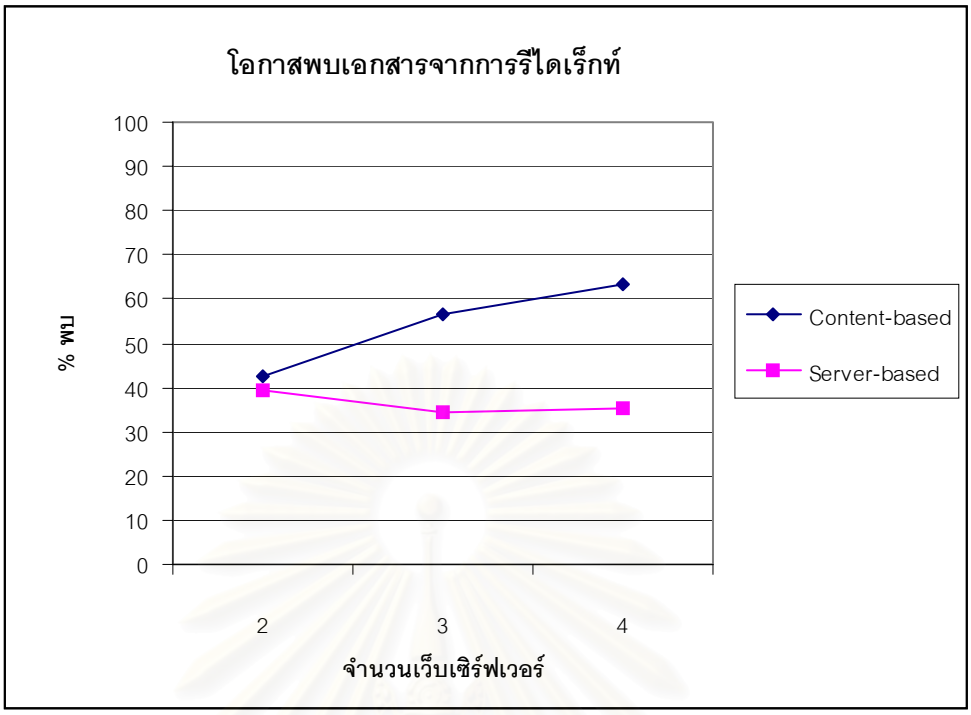
จากรูปที่ 4.21 ระบบที่ใช้การกระจายการร้องขอโดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้นมีเวลาตอบสนองน้อยที่สุด รองลงมาคือ การกระจายการร้องขอโดยพิจารณาจากสภาพภาระงาน และการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอสนั้นมีเวลาตอบสนองมากที่สุด โดยที่เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองลดลง

เมื่อเปรียบเทียบเวลาตอบสนองของการกระจายการร้องขอโดยพิจารณาจากสภาพภาระงาน กับการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส จะพบว่าการกระจายการร้องขอโดยวิธีแรกนั้นมีเวลาตอบสนองที่ดีกว่า เนื่องจากเมื่อเว็บเซิร์ฟเวอร์มีสภาพภาระงานมาก ก็จะกระจายภาระงานไปยังเว็บเซิร์ฟเวอร์ที่มีสภาพภาระงานน้อยถึงปานกลาง ทำให้สามารถบริการการร้องขอได้เร็วขึ้น เวลาตอบสนองจึงมีค่าน้อยกว่า และเมื่อเปรียบเทียบเวลาตอบสนองของการกระจายการร้องขอทั้งสองวิธี เมื่อใช้เว็บเซิร์ฟเวอร์ 2 และ 4 ตัวในการให้บริการพบว่า เวลาตอบสนองของทั้งสองวิธีมีค่าใกล้เคียงกันมาก แต่เมื่อใช้เว็บเซิร์ฟเวอร์ 3 ตัว เวลาตอบสนองของทั้งสองวิธีมีค่าต่างกัน ซึ่งเป็นเพราะเมื่อใช้เว็บเซิร์ฟเวอร์ 2 และ 4 ตัว การรีไดเร็กท์มีค่าใกล้เคียงกัน แต่เมื่อใช้เว็บเซิร์ฟเวอร์ 3 ตัว มีการรีไดเร็กท์มากกว่า ดังรูปที่ 4.22 เมื่อการรีไดเร็กท์มากขึ้น การกระจายภาระงานจากเว็บเซิร์ฟเวอร์ที่มีภาระงานมากกว่าไปยังเว็บเซิร์ฟเวอร์ที่มีภาระงานน้อยกว่าก็มีมากขึ้น ทำให้การให้บริการทำได้ดีขึ้น เวลาตอบสนองจึงมีค่าแตกต่างกัน

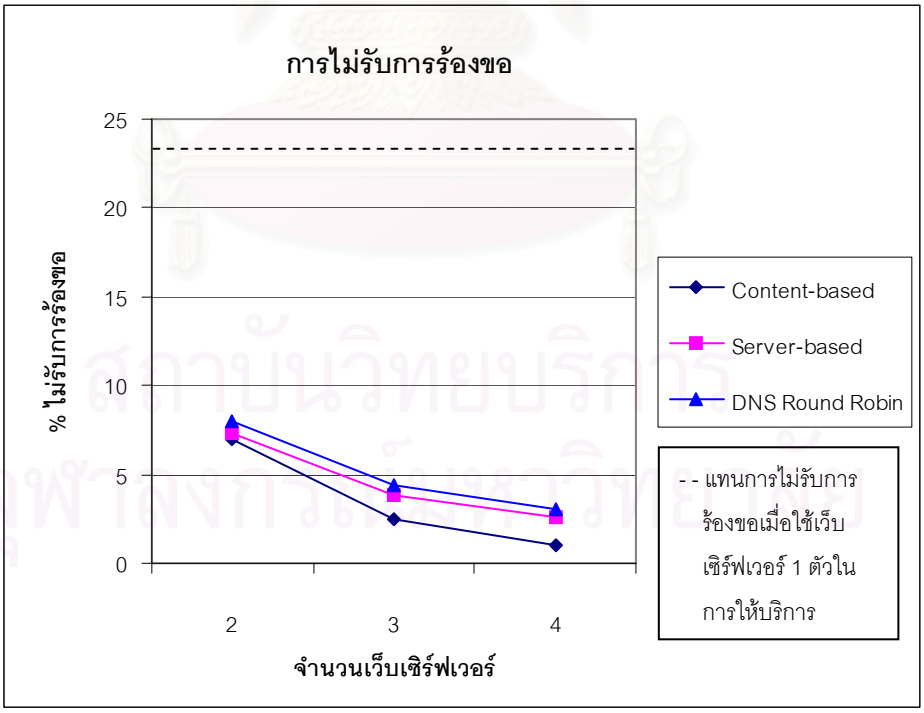
เมื่อเปรียบเทียบเวลาตอบสนองของการกระจายการร้องขอโดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหา กับการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส จะพบว่าการกระจายการร้องขอโดยวิธีแรกนั้นมีเวลาตอบสนองที่ดีกว่า เนื่องจากจะมีการส่งการร้องขอเอกสารเดียวกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน ซึ่งช่วยลดเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ ทำให้เวลาในการรอคอยเอกสารลดลง ดังนั้นเวลาตอบสนองจึงน้อยกว่าวิธีที่สอง เมื่อเปรียบเทียบเวลาตอบสนองของการกระจายการร้องขอทั้งสองวิธี เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น ผลต่างของเวลาตอบสนองของทั้งสองวิธีจะมากขึ้น โดยที่เมื่อใช้เว็บเซิร์ฟเวอร์ 2 ตัวในการให้บริการ เวลาตอบสนองของการกระจายการร้องขอทั้งสองวิธีจะไม่แตกต่างกันมาก แต่เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองของทั้งสองวิธีแตกต่างกันมากขึ้น โดยที่การกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส จะมีเวลาตอบสนองลดลงไม่มาก แต่การกระจายการร้องขอโดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองลดลงมาก ที่เป็นเช่นนี้เนื่องจากเมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น การตรวจสอบพบเอกสารจากข้อมูลสรุปบัฟเฟอร์เพิ่มขึ้น การรีไดเร็กท์การร้องขอจากเว็บเซิร์ฟเวอร์หนึ่งไปยังอีกเว็บเซิร์ฟเวอร์หนึ่งก็เพิ่มขึ้นดังรูปที่ 4.22 โดยเมื่อรีไดเร็กท์ไปแล้วก็มีโอกาสพบเอกสารในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ที่รับการร้องขอมากขึ้น ดังรูปที่ 4.23 ทำให้เวลารอคอยเอกสารลดลง ดังนั้นเวลาตอบสนองจึงลดลงด้วย



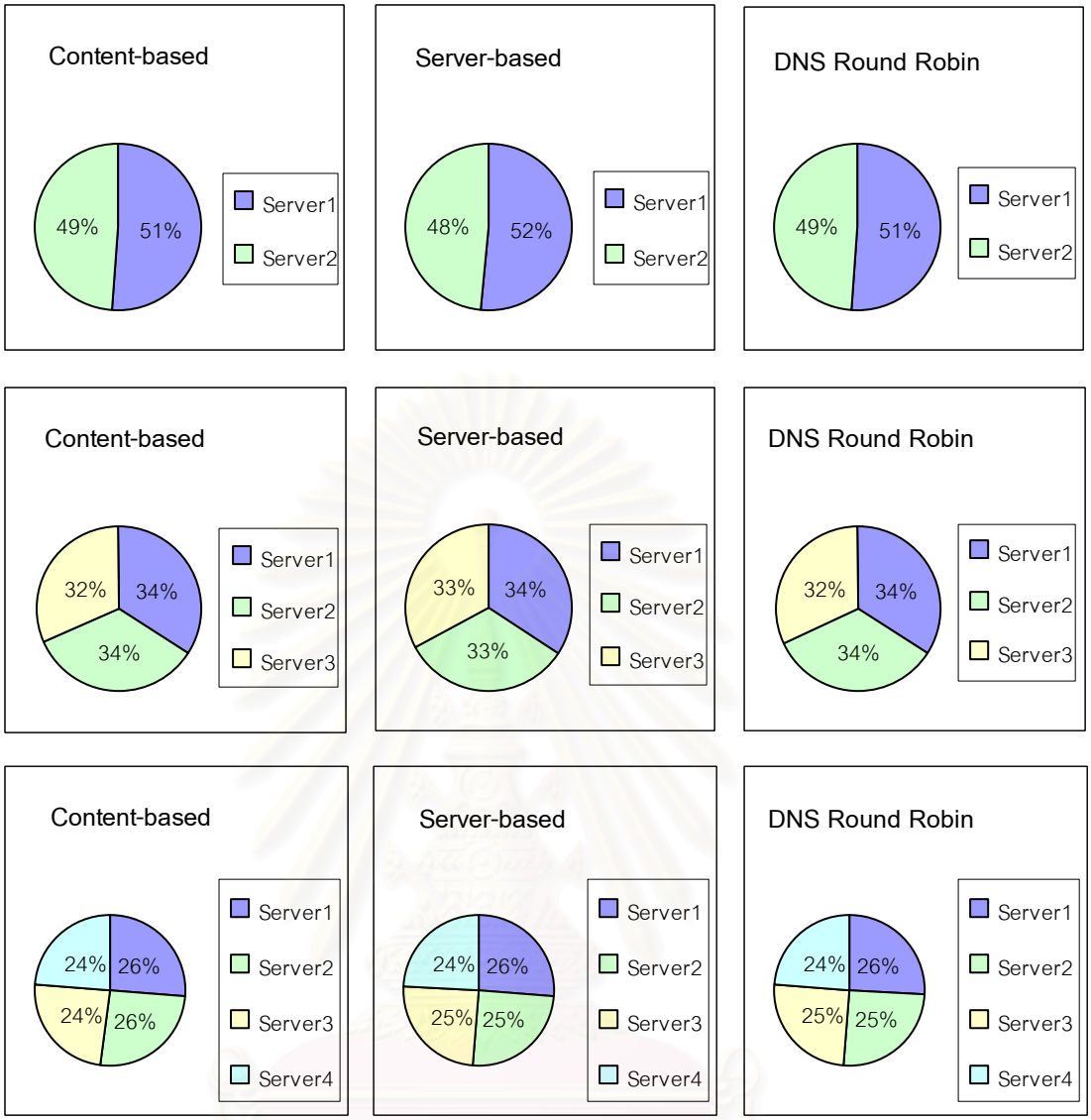
รูปที่ 4.22 เปอร์เซ็นต์รีไดเร็กท์ เมื่อใช้ดีเลย์ 500 มิลลิวินาที



รูปที่ 4.23 โอกาสพบเอกสารจากการรีไดเร็กท์ เมื่อใช้ดีเลย์ 500 มิลลิวินาที



รูปที่ 4.24 การไม่รับการร้องขอ เมื่อใช้ดีเลย์ 500 มิลลิวินาที



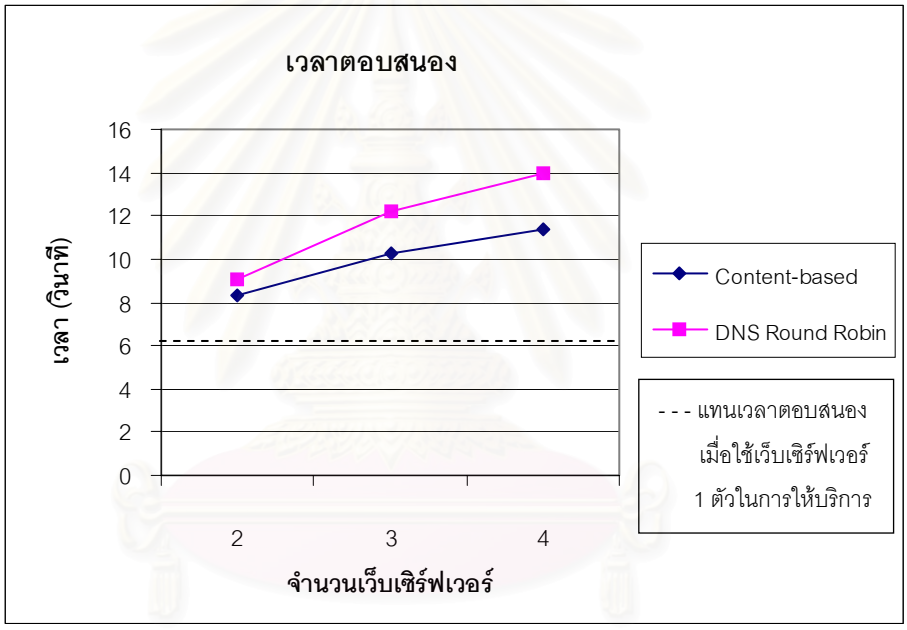
รูปที่ 4.25 เปอร์เซ็นต์การใช้งานของเว็บเซิร์ฟเวอร์ เมื่อกำหนดดีเลย์ 500 มิลลิวินาที

จากรูปที่ 4.24 จะพบว่าในระบบที่ใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา จะส่งการร้องขอเอกสารเดียวกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน ทำให้มีการกระจายการร้องขอมาก จึงเกิดการไม่รับการร้องขอน้อยที่สุด รองลงมา คือ การกระจายการร้องขอโดยพิจารณาจากข้อมูลสภาพภาระงาน จะกระจายการร้องขอเมื่อเว็บเซิร์ฟเวอร์มีสภาพภาระงานมากโดยการส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่มีสภาพภาระงานน้อยถึงปานกลาง ซึ่งทำจึงเกิดการไม่รับการร้องขอมากเป็นอันดับสอง และการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส เว็บเซิร์ฟเวอร์จะต้องเป็นผู้ให้บริการการร้องขอจากไคลเอนต์ทั้งหมด เมื่อเว็บเซิร์ฟเวอร์มีภาระงานมากจะไม่มีการกระจายภาระงานไปยังเว็บเซิร์ฟเวอร์ตัวอื่น ดังนั้นจึงมีโอกาสเกิดการไม่รับการร้องขอก็มากด้วย ทำให้วิธีนี้มีโอกาสเกิดการไม่รับการร้องขอมากที่สุด

และจากรูปที่ 4.25 พบว่าการกระจายภาระงานของเว็บเซิร์ฟเวอร์สำหรับทุก ๆ วิธีกระจายร้องขอ เว็บเซิร์ฟเวอร์มีการให้บริการการร้องขอปริมาณที่ใกล้เคียงกัน เนื่องมาจากการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส จะกระจายการร้องขอของไคลเอนต์ไปยังทุก ๆ เว็บเซิร์ฟเวอร์โดยใช้วิธีวนรอบ โดยไม่มีการจำลองลักษณะของไคลเอนต์จากเครือข่ายเดียวกัน ทำให้ผลการทดลองที่ได้ทุกเว็บเซิร์ฟเวอร์ให้บริการการร้องขอในปริมาณที่ใกล้เคียงกัน

4.2.6 การทดสอบประสิทธิภาพโดยใช้ข้อมูลการใช้เว็บของ www.thaisecondhand.com

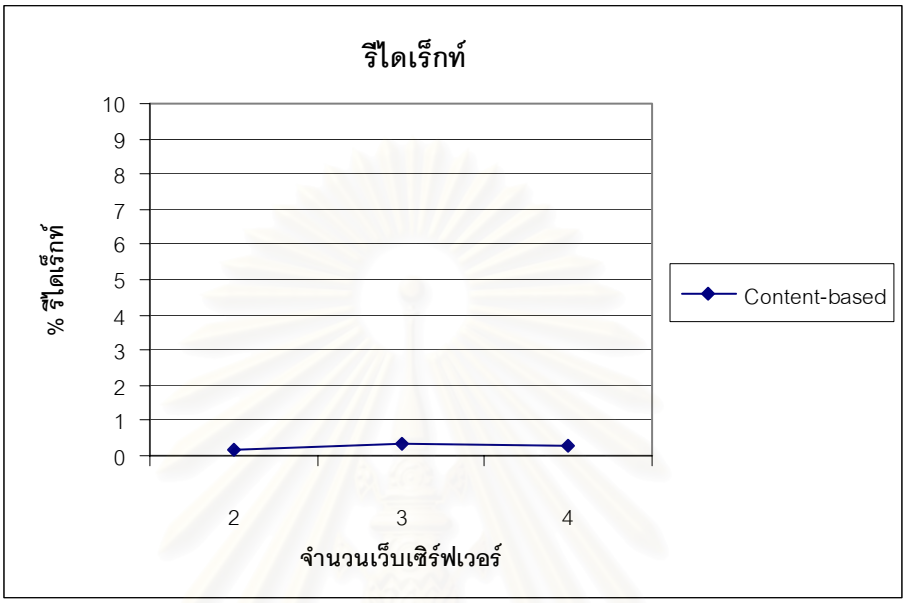
การทดลองนี้ กำหนดให้มีไคลเอนต์ที่ร้องขอข้อมูลพร้อม ๆ กัน 100 ไคลเอนต์ แต่ละเซิร์ฟเวอร์มีบัฟเฟอร์ขนาด 512 เมกะไบต์ มีการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ทุก ๆ 1 นาที โดยใช้ข้อมูลการใช้เว็บของ www.thaisecondhand.com



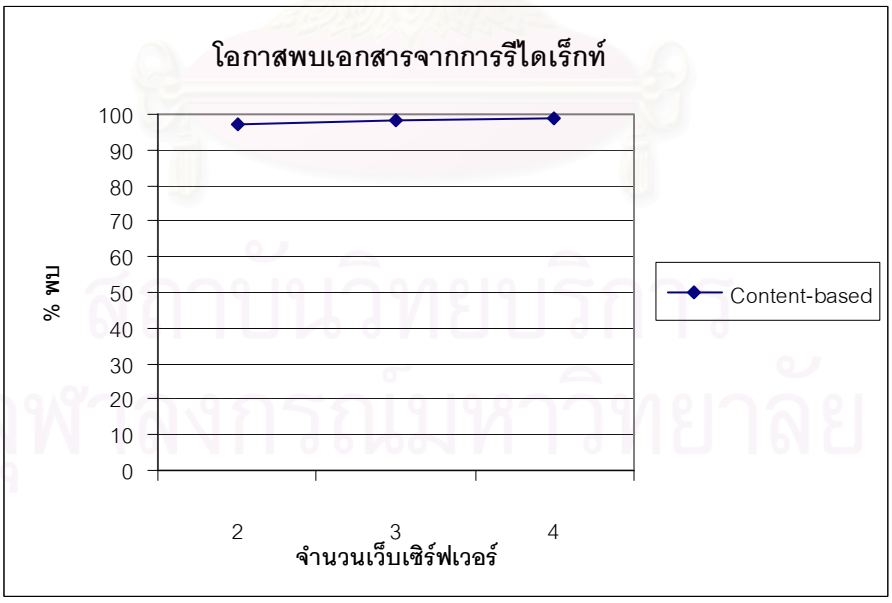
รูปที่ 4.26 เวลาตอบสนอง เมื่อใช้ข้อมูลการใช้เว็บของ www.thaisecondhand.com

จากการทดสอบโดยการเพิ่มจำนวนเว็บเซิร์ฟเวอร์ที่ให้บริการ เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น ระบบก็จะสามารถให้บริการการร้องขอได้ใกล้เคียงกัน แต่ในระบบที่กระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น เว็บเซิร์ฟเวอร์สามารถให้บริการได้เร็วกว่า มีเวลาตอบสนองที่ดีกว่าระบบที่กระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส ดังรูปที่ 4.26 นั่นเป็นเพราะการกระจายการร้องขอโดยวิธีแรกนั้น มีการกระจายการร้องขอกรณีที่เป็นการร้องขอเอกสารที่มีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ เพื่อส่งการร้องขอเอกสารเดียวกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน โดยรีไดเร็กท์ส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่น่าจะมีข้อมูลเอกสารนั้นอยู่ในบัฟเฟอร์

แต่จากรูปที่ 4.27 พบว่ามีการรีไดเร็กท์เกิดขึ้นน้อยมากไม่ถึง 1 เปอร์เซ็นต์ ซึ่งเป็นเพราะข้อมูลเอกสารส่วนมากนั้นเป็นข้อมูลขนาดเล็ก มีขนาดไม่เกินขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ ทำให้มีการตรวจสอบจากข้อมูลสรุปบัฟเฟอร์น้อยครั้ง การรีไดเร็กท์จึงเกิดขึ้นน้อยมาก



รูปที่ 4.27 เปอร์เซ็นต์รีไดเร็กท์ เมื่อใช้ข้อมูลของ www.thaisecondhand.com



รูปที่ 4.28 แสดงโอกาสพบเอกสารจากการรีไดเร็กท์
เมื่อใช้ข้อมูลของ www.thaisecondhand.com

และจะพบว่าเมื่อรีไดเร็กท์แล้วมีโอกาสพบเอกสารในบัฟเฟอร์ที่รับการรีไดเร็กท์เกือบ 100 เปอร์เซ็นต์ ดังรูปที่ 4.28 ซึ่งเป็นเพราะเอกสารส่วนมากมีขนาดเล็ก และมีจำนวนเอกสารไม่มากนัก ทำให้จำนวนเอกสารที่จะนำมาใช้คำนวณเป็นข้อมูลสรุปบัฟเฟอร์นั้นมีน้อยมาก ดังนั้นการใช้ข้อมูลสรุปบัฟเฟอร์ในการตัดสินใจกระจายการร้องขอจึงมีโอกาสผิดพลาดน้อยมาก ทำให้การรีไดเร็กท์นั้นมีโอกาสพบข้อมูลเอกสารในบัฟเฟอร์สูงมาก ทำให้เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลง ดังนั้น ระบบที่กระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัตและการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น จึงมีเวลาตอบสนองที่ดีกว่าระบบที่กระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส

แต่จะพบว่าเมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองมีค่าเพิ่มขึ้น ซึ่งเป็นเพราะเอกสารส่วนมากมีขนาดเล็ก ทำให้เกิดการรีไดเร็กท์น้อย เกิดการกระจายการร้องขอน้อยมาก เว็บเซิร์ฟเวอร์ที่รับการร้องขอนั้นจะต้องเป็นผู้ให้บริการการร้องขอเกือบทั้งหมด และเมื่อไม่พบข้อมูลเอกสารในบัฟเฟอร์ก็จะต้องอ่านข้อมูลเอกสารจากดิสก์เอง เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ก็เพิ่มขึ้นด้วย เวลาการรอคอยเอกสารจึงเพิ่มขึ้น ดังนั้น เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองจึงเพิ่มขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การวิเคราะห์ผลการทดลอง

เมื่อเปรียบเทียบการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา กับการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส จะเห็นว่าทั้งสองวิธีมีขั้นตอนวิธีที่เหมือนกัน คือ เมื่อไคลเอนต์ทำการสอบถามไปยังดีเอนเอส เพื่อเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ โดยดีเอนเอสจะตอบกลับมาเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ตัวหนึ่งในเว็บเซิร์ฟเวอร์ทั้งหมด ซึ่งทำให้มีการกระจายการร้องขอของไคลเอนต์ไปยังทุก ๆ เว็บเซิร์ฟเวอร์ และการร้องขอจากไคลเอนต์เดียวกันจะส่งไปยังเว็บเซิร์ฟเวอร์ตัวเดียวกันจนกว่าจะหมดช่วงอายุทีทีแอลที่ดีเอนเอสกำหนด ซึ่งทั้งสองวิธีจะมีขั้นตอนนี้ที่เหมือนกัน แต่การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น จะมีขั้นตอนที่มากกว่า คือ เว็บเซิร์ฟเวอร์สามารถกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่นได้ โดยใช้ข้อมูลสรุปบัฟเฟอร์และข้อมูลสภาพภาระงานของเว็บเซิร์ฟเวอร์ที่แลกเปลี่ยนกันระหว่างเว็บเซิร์ฟเวอร์ในการตัดสินใจกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่น ซึ่งจากผลการทดลองในบทที่ 4 จะพบว่าระบบที่ใช้การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหานั้น มีเวลาตอบสนองดีกว่า ซึ่งแสดงให้เห็นว่า การใช้ข้อมูลสรุปบัฟเฟอร์ในการตัดสินใจกระจายการร้องขอด้วยการรีไดเร็กท์ ช่วยให้บริการของเว็บเซิร์ฟเวอร์ดีขึ้น กล่าวคือ เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอจากไคลเอนต์ ก็จะทำการตรวจสอบ ถ้าพบว่าเอกสารนั้นมีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ และไม่พบในบัฟเฟอร์ ก็จะโดยเมื่อตรวจสอบจากข้อมูลสรุปบัฟเฟอร์ เพื่อส่งการร้องขอเอกสารเดียวกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน ถ้าตรวจสอบพบเอกสารจากข้อมูลสรุปบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ใด ก็จะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์นั้น ซึ่งจะพบว่า ยิ่งพบเอกสารจากข้อมูลสรุปบัฟเฟอร์มาก การรีไดเร็กท์ไปยังเว็บเซิร์ฟเวอร์อื่นก็มาก โดยที่รีไดเร็กท์ไปแล้วมีโอกาสพบข้อมูลเอกสารในบัฟเฟอร์ค่อนข้างมาก ทำให้เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลง เพราะจะใช้เวลาเข้าถึงข้อมูลเอกสารในดิสก์ครั้งแรกที่ได้รับการร้องขอเอกสารนั้น และเมื่อมีการรีไดเร็กท์มาแล้วพบเอกสารในบัฟเฟอร์ เว็บเซิร์ฟเวอร์ก็สามารถที่ตอบการร้องขอได้ทันที ทำให้เวลารอคอยเอกสารลดลง เวลาตอบสนองจึงดีขึ้น

จากการทดลองในบทที่ 4 จะพบว่า ปัจจัยที่มีผลต่อการให้บริการของเว็บเซิร์ฟเวอร์ มีดังนี้

1. จำนวนเว็บเซิร์ฟเวอร์

เมื่อจำนวนเว็บเซิร์ฟเวอร์ที่ให้บริการเพิ่มขึ้น บัฟเฟอร์รวมของระบบก็มีขนาดเพิ่มขึ้น ทำให้สามารถจัดเก็บข้อมูลเอกสารได้มากขึ้น มีการกระจายของเอกสารขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาการรีไดเร็กท์มากขึ้น ดังนั้น จำนวนเอกสารที่จะนำมาคำนวณข้อมูลสรุปบัฟเฟอร์ก็จะลดลง ทำให้ข้อมูลสรุปบัฟเฟอร์มีความถูกต้องมากขึ้น เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น การรีไดเร็กท์การร้องขอเพิ่มขึ้น และมีโอกาสพบเอกสารหลังการรีไดเร็กท์เพิ่มขึ้น อีกทั้งทำให้ระบบมีการกระจายภาระงานมากขึ้น และเมื่อมีการกระจายภาระงานเพื่อส่งการร้องขอเอกสารเดียวกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกัน ทำให้เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลงเพราะจะเข้าถึงข้อมูลเอกสารครั้งแรกครั้งเดียว ครั้งต่อไปก็สามารถนำข้อมูลในบัฟเฟอร์มาตอบการร้องขอได้ทันที แต่ในกรณีที่ใช้ข้อมูลของ www.thaisecondhand.com ในการทดสอบ จะพบว่าเมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น เวลาตอบสนองมีค่าเพิ่มขึ้นด้วย ซึ่งเป็นเพราะเอกสารมีปริมาณไม่มาก และเอกสารส่วนมากมีขนาดเล็ก การกระจายการร้องขอเกิดขึ้นน้อยมาก เว็บเซิร์ฟเวอร์ที่รับการร้องขอจะต้องเป็นผู้ให้บริการการร้องขอนั้นเองเกือบทั้งหมด โดยเมื่อไม่พบเอกสารในบัฟเฟอร์ก็จะต้องอ่านข้อมูลเอกสารจากดิสก์ แล้วจึงตอบการร้องขอของไคลเอนต์ เมื่อจำนวนเว็บเซิร์ฟเวอร์เพิ่มขึ้น ทำให้เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์เพิ่มขึ้น เวลาตอบสนองจึงเพิ่มขึ้น ซึ่งในกรณีนี้การเพิ่มจำนวนเว็บเซิร์ฟเวอร์จะไม่ช่วยให้บริการของเว็บเซิร์ฟเวอร์ดีขึ้น

2. ขนาดบัฟเฟอร์

เมื่อบัฟเฟอร์มีขนาดเพิ่มขึ้น บัฟเฟอร์รวมของระบบก็มีขนาดเพิ่มขึ้นด้วย ทำให้สามารถจัดเก็บเอกสารได้มากขึ้น และมีการแทนที่เอกสารในบัฟเฟอร์ลดลง เว็บเซิร์ฟเวอร์สามารถตรวจสอบพบเอกสารจากข้อมูลสรุปบัฟเฟอร์ได้มากขึ้น จึงเกิดการรีไดเร็กท์เพิ่มขึ้น โดยที่รีไดเร็กท์แล้วมีโอกาสพบเอกสารในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์ที่รับการรีไดเร็กท์เพิ่มขึ้น ทำให้เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลง เวลาตอบสนองจะยิ่งดีขึ้น

3. ช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์

เมื่อช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์มีค่าน้อย คือ มีการแลกเปลี่ยนข้อมูลบ่อย ข้อมูลจะมีความทันสมัย ทำให้ตรวจพบเอกสารจากข้อมูลสรุปบัฟเฟอร์มาก จึงมีการส่งการร้องขอเอกสารเดียวกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกันมากขึ้น ทำให้เวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ลดลง เวลารอคอยเอกสารลดลง เวลาตอบสนองจึงน้อย แต่เมื่อช่วงเวลาในการแลกเปลี่ยนมีค่ามาก คือ มีการแลกเปลี่ยนน้อย ข้อมูลไม่ทันสมัย เว็บเซิร์ฟเวอร์ไม่สามารถรู้ได้ว่ามีข้อมูลเอกสารนั้นในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์อื่น จึงต้องอ่านข้อมูลเอกสารจากดิสก์เอง ทั้งที่

ควรจะส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่มีข้อมูลเอกสารนั้นซึ่งจะสามารถตอบการร้องขอได้ทันที จึงทำให้เวลารอคอยเอกสารเพิ่มขึ้น เวลาตอบสนองจึงเพิ่มขึ้นด้วย

จากผลการทดลอง เมื่อมีการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์บ่อย เวลาตอบสนองจะยิ่งดี นั่นเป็นเพราะการทดลองนี้ไม่ได้พิจารณาถึงเวลาที่ใช้ในการคำนวณข้อมูลสรุปบัฟเฟอร์ เนื่องจากเป็นข้อจำกัดของโปรแกรมจำลองทำให้ไม่สามารถจำลองเวลาในการคำนวณข้อมูลสรุปบัฟเฟอร์ได้ แต่ในการใช้งานจริง การแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์บ่อย ก็จะต้องใช้เวลาในการคำนวณข้อมูลสรุปบัฟเฟอร์ ซึ่งอาจจะส่งผลต่อการให้บริการของเว็บเซิร์ฟเวอร์ ทำให้บริการของเว็บเซิร์ฟเวอร์ทำได้ช้าลง

4. ลักษณะข้อมูลเอกสาร

ข้อมูลที่ใช้ในการทดลองนั้น มาจากแหล่งข้อมูล 2 แหล่งที่แตกต่างกัน คือ ข้อมูลของ www.pantip.com ซึ่งเอกสารมีขนาดที่ไม่แน่นอน มีตั้งแต่ขนาดเล็กจนถึงขนาดใหญ่ เอกสารมีปริมาณมาก การกระจายการร้องขอเมื่อเอกสารมีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ก็มีมาก ทำให้มีการรีไดเร็กท์กระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่น มากกว่าเมื่อใช้ข้อมูลของ www.thaisecondhand.com ซึ่งข้อมูลมีขนาดค่อนข้างเล็ก ปริมาณเอกสารไม่มาก เอกสารที่มีขนาดมากกว่าขนาดเอกสารขั้นต่ำสำหรับการรีไดเร็กท์มีน้อยมาก การรีไดเร็กท์เพื่อกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์จึงน้อยมาก ซึ่งจะพบว่า เมื่อเอกสารส่วนมากมีขนาดเล็ก การกระจายการร้องขอโดยเว็บเซิร์ฟเวอร์ก็น้อยมาก ทำให้การบริการนั้นเสมือนกับเป็นการกระจายการร้องขอโดยวิธีวนรอบของดีเอนเอส

จากปัจจัยที่ได้กล่าวมาแล้ว พบว่าการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา เหมาะสมกับข้อมูลของ www.pantip.com มากกว่าข้อมูลของ www.thaisecondhand.com เพราะจะมีการกระจายการร้องขอ เมื่อเป็นการร้องขอเอกสารที่มีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ ซึ่งช่วยลดเวลารอคอยเอกสาร ทำให้เวลาตอบสนองลดลง โดยที่จะต้องมีการกำหนดจำนวนเว็บเซิร์ฟเวอร์และขนาดบัฟเฟอร์ให้เหมาะสมกับลักษณะเอกสาร และปริมาณการร้องขอ และเลือกช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ให้เหมาะสม เพื่อให้เว็บเซิร์ฟเวอร์สามารถให้บริการได้ดีที่สุด ซึ่งสามารถสรุปได้ดังตารางที่ 5.1

ตารางที่ 5.1 ปัจจัยที่มีผลต่อการให้บริการ

ปัจจัย	ลักษณะ/ความต้องการ
จำนวนเว็บเซิร์ฟเวอร์	ปานกลาง
เวลาในอ่านข้อมูลเอกสารในดิสก์	มาก
ขนาดบัฟเฟอร์	มาก
ความทันสมัยของข้อมูลสรุปบัฟเฟอร์	มาก
ลักษณะข้อมูล	เอกสารขนาดใหญ่

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

สรุปผลการวิจัย และข้อเสนอแนะ

จากการศึกษาเกี่ยวกับการกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา พบว่า การกระจายการร้องขอวิธีนี้นั้น เหมาะสำหรับกรณีที่ข้อมูลมีขนาดค่อนข้างใหญ่ คือ มีขนาดมากกว่าขนาดเอกสารขั้นต่ำที่พิจารณาเพื่อการรีไดเร็กท์ เพื่อให้เกิดการกระจายการร้องขอโดยเว็บเซิร์ฟเวอร์ ซึ่งจะช่วยลดเวลาในการเข้าถึงข้อมูลเอกสารในดิสก์ อันส่งผลให้เวลาในการให้บริการของเว็บเซิร์ฟเวอร์ดีขึ้น เวลาตอบสนองลดลง โดยงานวิทยานิพนธ์นี้ได้ทำการทดสอบว่าการใช้ข้อมูลสรุปบัฟเฟอร์นี้เป็นปัจจัยสำคัญที่ทำให้เว็บเซิร์ฟเวอร์สามารถให้บริการการร้องขอได้ดีกว่าระบบที่กระจายการร้องขอด้วยวิธีวนรอบของดีเอนเอส ซึ่งสามารถสรุปผลที่ได้จากการวิจัย ปัญหาและข้อจำกัดที่พบ รวมไปถึงข้อเสนอแนะแนวทางการพัฒนาต่อไป ดังนี้

6.1 สรุปผลการวิจัย

1. การรีไดเร็กท์เพื่อส่งการร้องขอเอกสารเดียวกันไปประมวลผลที่เว็บเซิร์ฟเวอร์เดียวกันนั้น ช่วยลดเวลารอคอยเอกสาร ทำให้เวลาตอบสนองลดลง
2. ความทันสมัยของข้อมูลสรุปบัฟเฟอร์ ทำให้รีไดเร็กท์แล้วมีโอกาสพบเอกสารมากขึ้น ทำให้เวลาตอบสนองลดลง
3. การกระจายการร้องขอของเว็บเซิร์ฟเวอร์โดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา มีความเหมาะสมสำหรับระบบที่ข้อมูลเอกสารมีขนาดใหญ่ เพราะจะมีการกระจายการร้องขอเพิ่มขึ้น ทำให้เวลาตอบสนองลดลง

6.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

1. โปรแกรมจำลองเอนเอสทู เป็นโปรแกรมจำลองที่ค่อนข้างซับซ้อน และเป็นการจำลองในเครื่องคอมพิวเตอร์เครื่องเดียว ทำให้ต้องใช้หน่วยความจำขนาดใหญ่ในการจำลอง และเครื่องที่ใช้ต้องมีประสิทธิภาพสูง โดยในการจำลองบางครั้งเกิดปัญหาหน่วยความจำหลักไม่เพียงพอแม้จะใช้หน่วยความจำหลักมากถึง 2 กิกะไบต์ ก็ยังเกิดปัญหาเรื่องหน่วยความจำ ทำให้ทำการจำลองไม่สำเร็จ
2. โปรแกรมจำลองเอนเอสทู ไม่สามารถจำลองเวลาในการคำนวณข้อมูลสรุปบัฟเฟอร์ เวลาในการเข้าถึงข้อมูลในบัฟเฟอร์ เวลาในการกระจายการร้องขอได้

3. ในการจำลองนั้น ไคลเอนต์ที่ใช้ในการจำลอง จะแบ่งตามเลขที่อยู่ไอพีของไคลเอนต์ที่ร้องขอตามข้อมูลการใช้เว็บ ทำให้การร้องขอของบางไคลเอนต์มีจำนวนมาก อีกทั้งรูปแบบการร้องขอของแต่ละไคลเอนต์นั้นจะส่งมาทีละหลาย ๆ การร้องขอ หรือส่งการร้องขอเข้ามาติด ๆ กัน ซึ่งจะต้องมีใช้แบนวิดท์ร่วมกัน ดังนั้นเวลาตอบสนองต่อการร้องขอก็จะเพิ่มขึ้นด้วย
4. กรณีที่เอกสารที่จะใช้ในการคำนวณข้อมูลสรุปบัฟเฟอร์มีจำนวนมาก ความถูกต้องของข้อมูลสรุปบัฟเฟอร์จะลดลง การรีไดเร็กท์การร้องขอไปยังเว็บเซิร์ฟเวอร์อื่น มีโอกาสที่จะพบเอกสารในบัฟเฟอร์ของเว็บเซิร์ฟเวอร์นั้นลดลง นั่นคือ ถ้าเอกสารที่จะนำมาใช้ในการคำนวณข้อมูลสรุปบัฟเฟอร์มีจำนวนมาก ความผิดพลาดของข้อมูลสรุปบัฟเฟอร์ก็จะเพิ่มขึ้นด้วย

6.3 ข้อเสนอแนะ

1. วิธีการกระจายการร้องขอโดยใช้การสมดุลภาระแบบพลวัต และการกระจายการร้องขอโดยพิจารณาจากเนื้อหา สามารถนำไปประยุกต์ใช้กับเว็บเซิร์ฟเวอร์ เพื่อใช้ในการกระจายการร้องขอระหว่างเว็บเซิร์ฟเวอร์ได้
2. กรณีเอกสารที่จะนำมาใช้ในการคำนวณข้อมูลสรุปบัฟเฟอร์มีจำนวนมาก ควรจะหาวิธีที่จะช่วยลดความผิดพลาดของข้อมูลสรุปบัฟเฟอร์ หรือเลือกขนาดของข้อมูลสรุปบัฟเฟอร์ที่เหมาะสม เพื่อเพิ่มความถูกต้องให้กับข้อมูลสรุปบัฟเฟอร์
3. พิจารณาถึงขนาดเอกสารที่เหมาะสม สำหรับใช้พิจารณาเพื่อการรีไดเร็กท์ เพื่อให้เกิดความคุ้มค่าที่ใช้การรีไดเร็กท์
4. การกระจายการร้องขอจากเว็บเซิร์ฟเวอร์หนึ่งไปยังอีกเว็บเซิร์ฟเวอร์หนึ่ง อาจใช้เทคนิค Distributed Packet Rewriting ในการส่งการร้องขอโดยตรงจากเว็บเซิร์ฟเวอร์หนึ่งไปยังอีกเว็บเซิร์ฟเวอร์หนึ่งโดยตรง ซึ่งน่าจะส่งผลให้การให้บริการดีกว่าการรีไดเร็กท์
5. การพัฒนาส่วนของการกระจายการร้องขอโดยการใช้ URL Rewriting แทนการใช้คุกกี้ เพื่อแก้ปัญหากรณีเว็บเบราเซอร์ไม่เปิดรับคุกกี้
6. เปรียบเทียบเวลาในการรีไดเร็กท์กับเวลาในการอ่านข้อมูลจากดิสก์ว่าค่าใดมีผลต่อการให้บริการของเว็บเซิร์ฟเวอร์มากกว่ากัน
7. การจำลองในงานวิทยานิพนธ์นี้ ไม่ได้คำนึงถึงเวลาในการคำนวณข้อมูลสรุปบัฟเฟอร์ เวลาในการเข้าถึงข้อมูลในบัฟเฟอร์ เวลาในการกระจายการร้องขอได้ ดังนั้น ควรมีการจำลองค่าพวกนี้ เพราะอาจจะส่งผลต่อเวลาในการให้บริการของเว็บเซิร์ฟเวอร์ได้

รายการอ้างอิง

[1] Alex Rousskov and Duane Wessels. Cache Digests. Proceedings of the 3rd International WWW Caching Workshop, Manchester, England , June 1998. TERENA, Trans-European Research and Education Networking Association.

[2] Antoine Mourad and Huiqun Liu. Scalable Web Server Architectures. Proceedings of the IEEE Symposium on Computers and Communications (ISCC'97), Alexandria, Egypt, July 1997.

[3] Chu – Sing Yang and Mon–Yen Luo. Efficient Support for Content–Based Routing in Web Server Clusters. USENIX Symposium on Internet Technologies and Systems, Boulder, Colorado, USA, October 1999.

[4] Eric Dean Katz, Michelle Butler and Robert McGrath. A Scalable HTTP Server: The NCSA Prototype. Computer Networks and ISDN Systems, 1994.

[5] Gil-Haeng Lee, Wang-Don Woo and Byeong-Nam Yoon. An Adaptive Load Balancing Algorithm Using Simple Prediction Mechanism. Proceedings of the 9th International Database and Expert Systems Applications Workshop, 496-501, 1998.

[6] Guerney D.H. Hunt, German S. Goldszmidt, Richard P. King and Rajat Mukherjee. Network Dispatcher: a connection router for scalable Internet services. Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, April 1998.

[7] Li Fan, Pei Cao, Jussara Almeida and Andrei Z. Border. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. IEEE/ACM Transactions on Networking, 8(3):281-293, June 2000.

[8] Luis Aversa and Azer Bestavros. Load Balancing a Cluster of Web Servers Using Distributed Packet Rewriting. Proceedings of the IEEE International Performance, Computing, and Communications Conference, Phoenix, AZ USA, February 2000.

[9] Mohit Aron et al, Darren Sanders, Peter Druschel and Willy Zwaenepoel. Scalable Content–Aware Request Distribution in Cluster–based Network Servers. Proceedings of the 2000 USENIX Annual Technical Conference, San Diego, CA USA, June 2000.

[10] Valeria Cardellini, Michele Colajanni and Philip S. Yu. Dynamic Load Balancing on Web-Server Systems. IEEE Internet Computing, 3(3):28-39, May-June 1999.

[11] Valeria Cardellini, Michele Colajanni and Philip S. Yu. Redirection Algorithms for Load Sharing in Distributed Web-Server Systems. Proceedings of the 19th International Conference on Distributed Computing Systems, Austin, Texas, June 1999.

[12] Vivek S. Pai, Mohit Aron, Gauray Banga, Michael Svendsen, Peter Druschel, Willy Zwaenepoel and Erich Nahum. Locality–Aware Request Distribution in Cluster–based Network Servers. Proceedings of the 8th ACM Conference on Architectural Support for Programming Languages and Operating Systems, Sanjose, CA USA, October 1998.

[13] Valeria Cardellini, Michele Colajanni and Philip S. Yu. High Performance Web-server Systems. Proceedings of the 13th International Symposium on Computer and Information Sciences, Ankara, Turkey, October 1998.

[14] Wenting Tang and Matt W. Mutka. Load Distribution via Static Scheduling and Client Redirection for Replicated Web Servers. Proceedings of the International Workshop on Scalable Web Services (in conjunction with ICPP'00), Toronto, Ontario, Canada, August 2000.

[15] Alteon Personal Content Director [Online]. Available from : http://www.westcon.uk/alteon/alteonprodinfoPCD.htm [2001, Sep 17]

[16] Network Simulation2 [Online]. Available from : http://www.isi.edu/nsnam/ns [2002, Jan 10]

[17] Yahoo [Online]. Available from : http://www.yahoo.com [2001, Aug 14]

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

สรุปคำสั่ง และตัวอย่างสคริปต์ไฟล์

สรุปคำสั่ง

1. คำสั่งพื้นฐานทั่วไปที่ใช้ในสคริปต์

- ns <otclfile> <arg> <arg> ...
  คำสั่งพื้นฐานที่ใช้รันสคริปต์ที่ใช้จำลองการทำงาน
- set ns_ [new Simulator]
  คำสั่งสร้างอินสแตนท์ของอ็อบเจกต์ตัวจำลองการทำงาน
- set now [$ns_ now]
  คำสั่งนี้จะส่งค่าเวลาปัจจุบันของการจำลองการทำงานกลับมา
- $ns_ run
  คำสั่งเริ่มการทำงาน
- $ns_ at <time> <event>
  คำสั่งกำหนดเวลาที่จะให้เกิดเหตุการณ์ต่างๆ ขึ้น
- $ns_ cancel <event>
  คำสั่งยกเลิกเหตุการณ์ได้กำหนดไว้
- $ns_ flush-trace
  คำสั่งเขียนข้อมูลการติดตามอ็อบเจกต์ทั้งหมดลงบัฟเฟอร์
- set agent [new Agent/AgentType]
  คำสั่งสร้าง agent ต่างๆ ตามรูปแบบ AgentType
- $ns_ attach-agent <node> <agent>
  คำสั่งใช้ติด <agent> บน <node>
- $ns duplex-link $node0 $node1 <bandwidth> <delay> <queue_type>
  คำสั่งการสร้างลิงค์เพื่อเชื่อมต่อระหว่างโหนดโดยมีรูปแบบ <queue_type> ต่างๆ ดังนี้ DropTail, RED, CBQ, FQ, SFQ, DRR
- set node [$ns node]
  คำสั่งการสร้าง node

2. **คำสั่งที่ใช้ในการจำลองการทำงานของเว็บเซิร์ฟเวอร์**

- set server [new Http/Server <sim> <s-node>]
  คำสั่งสร้างอินสแตนท์ของ Http Server บน <s-node> ที่กำหนด และใช้อินสแตนท์ของตัวจำลองการทำงาน <sim> เป็นตัวส่งผ่านอาร์กิวเมนต์
- set client [new Http/Client <sim> <c-node>]
  คำสั่งสร้างอินสแตนท์ของ Http Client บน <c-node>
- set dns [new Http/Dns <sim> <e-node>]
  คำสั่งสร้างอินสแตนท์ของ Http Dns บน <e-node>
- set pgp [new PagePool/ProxyTrace]
  คำสั่งสร้าง ProxyTrace PagePool ใช้สำหรับจัดการการร้องขอจากข้อมูลจริง
- $pgp set-reqfile "reqlog"
  คำสั่งกำหนดไฟล์ที่ใช้สำหรับการร้องขอ
- $pgp set-pagefile "pglog"
  คำสั่งกำหนดไฟล์ที่ใช้เป็นข้อมูลของเอกสาร
- $pgp set-client-num <num>
  คำสั่งกำหนดจำนวนผู้เรียกขอในการจำลองการทำงานของเว็บแคช
- $pgp bimodal-ratio <ratio>
  คำสั่งกำหนดค่าของเอกสารที่มีการเปลี่ยนแปลงบ่อย โดยมีค่าเป็น <ratio>*10 เปอร์เซ็นต์
- $pgp ranvar-dp <rv>
  คำสั่งกำหนดช่วงเวลาการเปลี่ยนแปลงของเอกสารสำหรับเอกสารที่มีการเปลี่ยนแปลงบ่อย
- $pgp ranvar-sp <rv>
  คำสั่งกำหนดช่วงเวลาการเปลี่ยนแปลงของเอกสารสำหรับเอกสารที่ไม่ค่อยมีการเปลี่ยนแปลง
- $server set-page-generator <pgp>
  คำสั่งกำหนด PagePool สำหรับ Server
- $server log <handle-to-log-file>
  คำสั่งกำหนด log file สำหรับ Server

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- $client set-page-generator <pgp>
  คำสั่งกำหนด PagePool สำหรับ Client
- $client log <handle-to-log-file>
  คำสั่งกำหนด log file สำหรับ Client
- $client connect <server>
  คำสั่งเชื่อมต่อ Client กับ Server
- $client start-session <server> <server>
  คำสั่งเริ่มต้นส่งการร้องขอจาก Client ไปยัง Server
- $server set-method <method>
  คำสั่งกำหนดวิธีกระจายการร้องขอ
- $server set-buffersize <buffer-size>
  คำสั่งกำหนดขนาดบัฟเฟอร์
- $server set-replacement-style “LRU”
  คำสั่งกำหนดวิธีการแทนที่เอกสาร
- $server set-update-summary-interval <summary-interval>
  คำสั่งกำหนดช่วงเวลาในการแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์
- $server set-check-load-interval <load-interval>
  คำสั่งกำหนดช่วงเวลาในตรวจสอบสภาพภาระงาน
- $server set-load-threshold <queue-length> <th> <tl>
  คำสั่งกำหนดค่าขีดแบ่งสภาพภาระงาน
- $server register-server $dns
  คำสั่งลงทะเบียนให้ DNS รู้ว่ามี Server อยู่ในระบบ
- $server($i) connect-server $server($j)
  คำสั่งในการแจ้งให้ Server ทั้งสอง Server รู้ว่ามี Server อีกตัวที่ให้บริการอยู่ เพื่อใช้ในการกระจายการร้องขอ และแลกเปลี่ยนข้อมูล
- $client set-dns <dns>
  คำสั่งกำหนด DNS ที่ไคลเอนต์จะใช้สำหรับทำการเปลี่ยนยูอาร์แอลเป็นที่อยู่ไอพีของเว็บเซิร์ฟเวอร์
- $client connect-dns
  คำสั่งเชื่อมต่อ Client กับ DNS

- $client disconnect-dns

  คำสั่งยกเลิกการเชื่อมต่อ Client กับ DNS

- $client disconnect <server>

  คำสั่งสิ้นสุดการเชื่อมต่อ Client กับ Server

- $server unregister-server <dns>

  คำสั่งยกเลิกการลงทะเบียนของ Server

ตัวอย่างสคิปต์ไฟล์

```
set ns [new Simulator]
set node(d) [$ns node]
set numc 100
for {set i 1} {$i <= $numc} {incr i} {
        set node(c$i) [$ns node]
        $ns duplex-link $node(d) $node(c$i) 100Mb 100ms DropTail
}
set nums 4
for {set i 1} {$i <= $nums} {incr i} {
        set node(s$i) [$ns node]
        $ns duplex-link $node(d) $node(s$i)  10Mb  5ms DropTail
}
for {set i 1} {$i <= $nums} {incr i} {
        for {set j 1} {$j <= $numc} {incr j} {
                $ns duplex-link $node(s$i) $node(c$j) 45Mb 500ms DropTail
        }
}

$ns rtproto Session

# Output log
set log [open "16.log" w]

# Input log
set pgp [new PagePool/ProxyTrace]
$pgp set-reqfile "/home/g43nsr/pantip/16/reqlog16"
$pgp set-pagefile "/home/g43nsr/pantip/16/pglog16"

#set number of client to use this reqfile, pagefile
$pgp set-client-num 28538

# Set the ratio of hot pages in all pages. Because no page modification
# data is available in most traces, we assume a bimodal page age distribution
$pgp bimodal-ratio 0.1


# Dynamic (hot) page age generator
set tmp [new RandomVariable/Exponential] ;# Age generator
$tmp set avg_ 5 ;# average page age
$pgp ranvar-dp $tmp
```

```
# Static page age generator
set tmp [new RandomVariable/Constant]
$tmp set val_ 1000
$pgp ranvar-sp $tmp

#-------------- DNS ------------
set dns [new Http/Dns $ns $node(d)]
#$dns register-server $server

#---------------- server ------------------
set interval [expr 1 * 60]
set buffersize 512000000
set checkloadinterval 1
set queuelength 1000
set th 0.7
set tl 0.3
#method 0 -> dns round robin  method4 -> content-based
set method 4
for {set i 1} {$i <= $nums} {incr i} {
        set n $node(s$i)
        set server($i) [new Http/Server $ns $n]
        $server($i) set-page-generator $pgp
        $server($i) log $log
        $server($i) set-buffer-size $buffersize
        $server($i) set-replacement-style "LRU"
        $server($i) set-method $method
        if {$method > 0} {
                $server($i) set-update-summary-interval $interval
        }
        $server($i) set-check-load-interval $checkloadinterval
        $server($i) set-load-threshold $queuelength $th $tl
}

#---------------- Client ---------------
for {set i 1} {$i <= $numc} {incr i} {
        set n $node(c$i)
        set client($i) [new Http/Client $ns $n]
        $client($i) set-page-generator $pgp
        $client($i) log $log
        $client($i) set-dns $dns
}
set startTime 1 ;# simulation start time
set finishTime 86400 ; # simulation end time
$ns at $startTime "start-connection"
$ns at $finishTime "finish"

proc start-connection {} {
        global ns server dns client numc nums method
        for {set i 1} {$i <= $nums} {incr i} {
                $server($i) register-server $dns
        }
        if {$method > 0} {
                for {set i 1} {$i <= $nums} {incr i} {
                        for {set j 1} {$j <= $nums} {incr j} {
                                if {$i != $j} {
                                        $server($i) connect-server $server($j)
                                }
```

```
                        }
                }
        }
        for {set i 1} {$i <= $numc} {incr i} {
                for {set j 1} {$j <= $nums} {incr j} {
                        $client($i) connect $server($j)
                }
        }
        set j 1
        for {set i 1} {$i <= $numc} {incr i} {
                $client($i) connect-dns
                $client($i) start-session $server($j) $server($j)
        }
}

proc finish {} {
        global ns log server client dns nums numc
        puts ""
        for {set i 1} {$i <= $numc} {incr i} {
                $client($i) print
        }
        puts ""
        for {set i 1} {$i <= $nums} {incr i} {
                $server($i) print
        }
        puts ""
        for {set i 1} {$i <= $numc} {incr i} {
                for {set j 1} {$j <= $nums} {incr j} {
                        $client($i) disconnect $server($j)
                }
                $client($i) disconnect-dns
        }
        for {set i 1} {$i <= $nums} {incr i} {
                $server($i) unregister-server $dns
        }
        $ns flush-trace
        flush $log
        close $log
        exit 0
}
$ns run
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

การแก้ไขโปรแกรมจำลองเอนเอสทู

เนื่องจากวิทยานิพนธ์นี้มีขั้นตอนการทำงานดังนี้ คือ เมื่อไคลเอนต์ต้องการส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ ก็จะต้องติดต่อไปยังดีเอนเอส เพื่อทำเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ แล้วจึงส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่ดีเอนเอสกำหนด โดยที่การร้องขอที่ส่งไป จะไม่กำหนดค่าคุกกี้ เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอจากไคลเอนต์ ก็จะตรวจสอบว่ามีการกำหนดค่าคุกกี้มากับการร้องขอนั้นหรือไม่ ถ้ามีการกำหนดค่าคุกกี้ เว็บเซิร์ฟเวอร์ก็จะให้บริการการร้องขอนั้นเอง แต่ถ้าไม่มีการกำหนดค่าคุกกี้ เว็บเซิร์ฟเวอร์ก็จะทำการเลือกเว็บเซิร์ฟเวอร์ที่น่าจะมีการข้อมูลเอกสารที่ร้องขออยู่ในบัฟเฟอร์ โดยตรวจสอบจากข้อมูลสรุปบัฟเฟอร์และข้อมูลสภาพภาระงานของเว็บเซิร์ฟเวอร์ เพื่อกระจายการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่น และเมื่อต้องการกระจายการร้องขอ ก็จะทำการกำหนดในค่าคุกกี้ แล้วทำการตอบการร้องขอด้วย Http Redirect พร้อมกับคุกกี้นั้น ซึ่งเว็บเซิร์ฟเวอร์จะต้องมีการนำข้อมูลเอกสารในบัฟเฟอร์มาคำนวณเป็นข้อมูลสรุปบัฟเฟอร์ และแลกเปลี่ยนกันระหว่างเว็บเซิร์ฟเวอร์ นอกจากนี้จะต้องมีการแลกเปลี่ยนข้อมูลสภาพภาระงานของเว็บเซิร์ฟเวอร์ด้วย

จากการศึกษาโปรแกรมเอนเอสทู จะพบว่า โครงสร้าง และการทำงานของเอนเอสทูไม่สอดคล้องกับความต้องการของวิทยานิพนธ์ ดังตารางที่ ข.1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.1 เปรียบเทียบโปรแกรมจำลองเอนเอสทูกับความต้องการของวิทยานิพนธ์

เอนเอสทู	วิทยานิพนธ์
1. โครงสร้างระบบจะประกอบด้วย เว็บเซิร์ฟเวอร์ และไคลเอนต์	1. โครงสร้างระบบ จะประกอบด้วย เว็บเซิร์ฟเวอร์ ไคลเอนต์ และดีเอนเอส นั่นคือ ต้องพัฒนาในส่วนของดีเอนเอส เพื่อทำหน้าที่ในการเปลี่ยนยูอาร์แอลเป็นเลขที่อยู่ไอพีโดยใช้วิธีวนรอบ พัฒนาเพิ่มคลาส Http/DNS ใน http-dns.tcl
2. เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอ ก็จะตอบเป็นข้อมูลเอกสาร	2. เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอ อาจจะตอบการร้องขอเป็นข้อมูลเอกสาร หรือกระจายการร้องขอโดยใช้ Http Redirect ดังนั้น ต้องพัฒนาเพิ่มในส่วนของการรับการร้องขอ ให้มีการตรวจสอบจากค่าคุกกี้ เพื่อตรวจสอบว่าจะตอบเป็นข้อมูลเอกสาร หรือจะกระจายการร้องขอ แก้ไขในคลาส Http/Server ใน http-server.tcl
3. ไม่มีการกำหนดขนาดบัฟเฟอร์ ไม่มีการแทนที่เอกสาร แต่จะเก็บข้อมูลทุก ๆ เอกสารไว้ในบัฟเฟอร์	3. มีการกำหนดขนาดบัฟเฟอร์ และมีการแทนที่เอกสารแบบแอลอาร์ยูแก้ไขในคลาส ClientPagePool ใน pagepool.cc 4. มีการคำนวณข้อมูลสรุปบัฟเฟอร์ และแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์ระหว่างเว็บเซิร์ฟเวอร์ แก้ไขในคลาส Http/Server ใน http-server.tcl 5. มีการตรวจสอบสภาพภาระงาน และแลกเปลี่ยนข้อมูลสภาพภาระงาน เมื่อสภาพภาระงานเปลี่ยนแปลงจากสภาพภาระงานน้อย เป็นสภาพภาระงานมาก หรือสภาพภาระงานมากเปลี่ยนเป็นสภาพภาระงานน้อย ก็จะแจ้งให้เว็บเซิร์ฟเวอร์อื่นทราบ แก้ไขในคลาส Http/Server ใน http-server.tcl 6. จำลองเวลาเข้าถึงข้อมูลเอกสารในดิสก์ แก้ไขในคลาส Http/Server ใน http-server.tcl

ดังนั้นจึงต้องนำโปรแกรมเอนเอสทูมาพัฒนาเพิ่มเติม เพื่อให้สามารถทำงานได้สอดคล้องตามแนวทางของวิทยานิพนธ์ ซึ่งส่วนที่ต้องแก้ไข และพัฒนาเพิ่มเติม มีดังนี้

1. คลาส ClientPagePool ใน pagepool.cc
   1.1 การเพิ่มเอกสารในบัฟเฟอร์ จะต้องมีการตรวจสอบว่าบัฟเฟอร์มีเนื้อที่เพียงพอที่จะเพิ่มเอกสารใหม่หรือไม่ ถ้ามีเนื้อที่ไม่เพียงพอ ก็จะต้องทำการแทนที่เอกสาร โดยการลบเอกสารที่ไม่มีการเข้าถึงนานที่สุดก่อน จนกว่าบัฟเฟอร์จะมีเนื้อที่เพียงพอที่จะเก็บเอกสารใหม่
   1.2 การร้องขอเอกสารในบัฟเฟอร์ จะต้องมีการจัดลำดับของเอกสารตามวิธีแทนที่เอกสารแบบแอลอาร์ยู
   1.3 นำข้อมูลเอกสารในบัฟเฟอร์มาคำนวณเป็นข้อมูลสรุปบัฟเฟอร์
   1.4 ตรวจสอบว่ามีเอกสารในข้อมูลสรุปบัฟเฟอร์หรือไม่ เพื่อเลือกเว็บเซิร์ฟเวอร์ที่จะให้บริการการร้องขอ
   1.5 เลือกเว็บเซิร์ฟเวอร์ที่จะให้บริการ เพื่อกระจายการร้องขอ โดยพิจารณาจากขนาดเอกสาร ข้อมูลในบัฟเฟอร์ ข้อมูลสรุปบัฟเฟอร์และข้อมูลสภาพภาระงานของเว็บเซิร์ฟเวอร์อื่น
   1.6 จัดเก็บข้อมูลสรุปบัฟเฟอร์ และข้อมูลสภาพภาระงานของเว็บเซิร์ฟเวอร์อื่น
2. คลาส Http ใน http.cc

   คลาส Http เป็นคลาสกลางเพื่อให้คลาส Http/Server เรียกใช้ฟังก์ชันในคลาส ClientPagePool จึงมีการรับคำสั่งจากคลาส Http/Server แล้วเรียกใช้ฟังก์ชันในคลาส ClientPagePool
3. คลาส Http/Client ใน http-client.tcl
   3.1 สอบถามไปยังดีเอนเอส เพื่อเปลี่ยนจากยูอาร์แอลเป็นเลขที่อยู่ไอพีของเว็บเซิร์ฟเวอร์ และเมื่อหมดช่วงอายุทีทีแอลที่ดีเอนเอสกำหนด ไคลเอนต์ก็จะต้องสอบถามไปยังดีเอนเอสอีกครั้ง
   3.2 ส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์ที่ดีเอนเอสกำหนด โดยส่งค่าคุกกี้ไปด้วย
   3.3 รับการตอบการร้องขอที่เป็น Http Redirect เพื่อทำการส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์อื่นต่อไป
4. คลาส Http/Server ใน http-server.tcl
   4.1 ลงทะเบียนให้ดีเอนเอสรู้ว่ามีเว็บเซิร์ฟเวอร์นี้อยู่ในระบบ
   4.2 รับการร้องขอจากไคลเอนต์ ให้มีการตรวจสอบค่าคุกกี้ ถ้ามีการกำหนดค่าคุกกี้เว็บเซิร์ฟเวอร์ก็จะตอบเป็นข้อมูลเอกสาร แต่ถ้าไม่มีการกำหนดค่าคุกกี้ก็จะ

กระจายการร้องขอเพื่อเลือกเว็บเซิร์ฟเวอร์ที่จะให้บริการการร้องขอ โดยเรียกใช้ฟังก์ชันในคลาส ClientPagePool ถ้าเว็บเซิร์ฟเวอร์ที่จะเป็นผู้ให้บริการเป็นเว็บเซิร์ฟเวอร์ตัวอื่น ก็จะกำหนดค่าคุกกี้ และตอบเป็น Http Redirect

4.3 จำลองเวลาในการอ่านข้อมูลเอกสารจากดิสก์

4.4 เรียกใช้ฟังก์ชันในคลาส ClientPagePool เพื่อคำนวณข้อมูลสรุปบัฟเฟอร์ และแลกเปลี่ยนข้อมูลสรุปบัฟเฟอร์กับเว็บเซิร์ฟเวอร์อื่น

4.5 ตรวจสอบสภาพภาระงานของตนเอง และแลกเปลี่ยนข้อมูลสภาพภาระงานกับเว็บเซิร์ฟเวอร์อื่น

4.6 เรียกใช้ฟังก์ชันในคลาส ClientPagePool เพื่อจัดเก็บข้อมูลสรุปบัฟเฟอร์และสภาพภาระงานของเว็บเซิร์ฟเวอร์อื่น

5. คลาส Http/DNS ใน http-dns.tcl

5.1 รับการลงทะเบียนจากเว็บเซิร์ฟเวอร์

5.2 เปลี่ยนยูอาร์แอลเป็นเลขที่อยู่ไอพีหนึ่งของเว็บเซิร์ฟเวอร์ทั้งหมด โดยกำหนดอายุทีทีแอลด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวนุจรีย์ ศรีเพ็ชรดานนท์ เกิดเมื่อวันที่ 12 เมษายน พ.ศ. 2521 สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ เมื่อปีการศึกษา 2542 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2543

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย