การดึงข้อมูลล่วงหน้าบนเว็บโดยเปลี่ยนแปลงตามแบบวิดท์

นายภรศิษฐ์ ธนวุฒิวัฒน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2544

ISBN 974-03-0428-1

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

BANDWIDTH-BASED ADAPTIVE WEB PREFETCHING

Mr. Pornsit Tanavuttiwat

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2001

ISBN 974-03-0428-1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อวิทยานิพนธ์	การดึงข้อมูลล่วงหน้าบนเว็บโดยเปลี่ยนแปลงตามแบนวิดท์
โดย	นาย ภรศิษฐ์ ธนวุฒิวัฒน์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
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 วรุตม์ ทวีทรัพย์)

.....กรรมการ
(อาจารย์ ชยา ลิมจิตติ)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภรศิษฐ์ ธนวุฒิวัฒน์ : การดึงข้อมูลล่วงหน้าบนเว็บโดยเปลี่ยนแปลงตามแบนวิดท์. (BANDWIDTH-BASED ADAPTIVE WEB PREFETCHING) อ.ที่ปรึกษา : อ.ดร. ณัฐวุฒิ หนูไพโรจน์, 50 หน้า, ISBN 974-03-0428-1

ปัจจุบันเวิลด์ไวด์เว็บได้รับความนิยมอย่างมาก ทำให้ปริมาณข้อมูลที่เกิดขึ้นมีมากกว่าที่ระบบอินเตอร์เน็ตสามารถรองรับได้ ซึ่งส่งผลให้ผู้ใช้งานเว็บต้องใช้เวลานานในการรอคอยเอกสารจากอินเตอร์เน็ต จึงได้มีงานวิจัยจำนวนมาก ค้นคว้าหาวิธีลดเวลาการรอคอยเอกสารที่เกิดขึ้น การดึงข้อมูลล่วงหน้าเป็นวิธีหนึ่งที่ใช้ลดเวลาการรอคอยเอกสารได้อย่างมีประสิทธิภาพ โดยมีหลักการที่ว่าในขณะที่ผู้ใช้งานกำลังดูเอกสารอยู่ระบบจะทำการทำนายเอกสารที่ผู้ใช้งานร้องขอในอนาคตอันใกล้และดึงข้อมูลล่วงหน้ามาเก็บไว้ในแคช แต่การดึงข้อมูลล่วงหน้ามีจุดบกพร่องที่สำคัญอยู่ 2 ประการคือ การดึงข้อมูลล่วงหน้าจะมีการใช้แบนวิดท์จำนวนมากในการดึงเอกสารที่ได้ทำนายไว้ นอกจากนี้การควบคุมอัตราการดึงข้อมูลของระบบการดึงข้อมูลล่วงหน้าจะมีลักษณะตายตัวซึ่งไม่เหมาะสมกับระบบที่มีปริมาณการใช้งานเปลี่ยนแปลงอยู่ตลอดเวลา จุดบกพร่องที่เกิดขึ้นจะก่อให้เกิดปัญหากับระบบเครือข่ายที่สามารถใช้งานแบนวิดท์ได้จำกัด

ในงานวิทยานิพนธ์นี้ได้เสนอวิธีการดึงข้อมูลล่วงหน้าแบบใหม่ที่จะช่วยลดเวลาการรอคอยของเอกสารโดยสร้างผลกระทบต่อระบบเครือข่ายน้อย โดยระบบการดึงข้อมูลล่วงหน้าแบบใหม่จะวัดปริมาณการใช้งานแบนวิดท์ของระบบเพื่อหาปริมาณแบนวิดท์ที่ว่างในระบบ และทำการดึงข้อมูลล่วงหน้าโดยใช้แบนวิดท์ที่วัดได้เท่านั้น และวิทยานิพนธ์นี้ได้ทำการทดสอบระบบที่ออกแบบขึ้น เปรียบเทียบกับระบบดึงข้อมูลล่วงหน้าแบบเดิม การทดสอบจะใช้วิธีจำลองการทำงานของระบบเครือข่ายโดยใช้ข้อมูลการใช้เว็บที่ได้จากพร็อกซี เซิร์ฟเวอร์ของสำนักเทคโนโลยีสารสนเทศ จุฬาลงกรณ์มหาวิทยาลัย จากผลการทดลองพบว่าในสภาพแวดล้อมที่มีปริมาณแบนวิดท์จำกัด การดึงข้อมูลล่วงหน้าที่ออกแบบขึ้นสามารถเพิ่มฮิตเรทของพร็อกซีได้โดยที่ส่งผลกระทบต่อระบบน้อยมากในขณะที่การดึงข้อมูลล่วงหน้าแบบเดิมเพิ่มฮิตเรทได้มากกว่าแต่จะส่งผลกระทบต่อระบบมากกว่ามาก ทำให้การดึงข้อมูลล่วงหน้าที่ออกแบบขึ้นสามารถลดเวลาเข้าถึงเอกสารของผู้ใช้งานได้ดีกว่าการดึงข้อมูลล่วงหน้าแบบเดิม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่ออาจารย์ที่ปรึกษา	
ปีการศึกษา	2544	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม	

##4270473021 : MAJOR COMPUTER ENGINEERING
KEY WORD : PREFETCH / CACHE / WWW / PAIR WISE
PORNSIT TANAVUTTIWAT : BANDWIDTH-BASED ADAPTIVE WEB PREFETCHING, THESIS ADVISOR : DR. NATAWUT NUPAIROJ, 50 pp. ISBN 974-03-0428-1

The increasingly popularity of World Wide Web lead to the increasing of Internet bandwidth. As Internet has limited bandwidth, web users must wait for a long time to access web documents. To elevate this problem several researches have focused on the way to reduce user access time. Web Prefetching is one of the methods that are considered to reduce access time efficiently. The main idea of prefetching is to predict the next document that user will request it and fetch into cache while user is viewing a current document. But Prefetching has two major drawbacks, which are large bandwidth requirement and fixed prefetching control parameter that may not be suitable for Internet environments that are constantly changed. These disadvantages will impose serious limitations of using prefetching in the real world.

In this thesis we have studied a new prefetching scheme that can reduce the user access time while generates minimal effect to network system. Our scheme will measure the bandwidth usage in the network and perform prefetching using only available bandwidth. We have conduct experiments in order to study the performance of our prefetching scheme. We have simulated network system by using access logs from Proxy server of the office of Information Technology of Chulalongkorn University. The result of simulation shows that in bandwidth-limited network, our scheme can increase the hit rate of proxy while experiencing almost degradation of the quality of service of network. While old scheme can increase more hit rate but greatly degrade the performance of the quality of service. Consequently, our scheme can reduce the access time better than old scheme.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Computer Engineering Student's signature .....
Field of study Computer Engineering Advisor's signature .....
Academic year 2001 Co-advisor's signature .....
000

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งจาก อาจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ รวมทั้งสนับสนุนการทำงานวิจัยของข้าพเจ้ามาด้วยดีตลอด ขอขอบคุณ อาจารย์ ดร.ยรรยง เต็งอำนวย อาจารย์ ดร.ทวิตีย์ เสนีวงค์ ณ อยุธยา และอาจารย์ท่านอื่นๆ ที่ให้การดูแลและอบรมสั่งสอนข้าพเจ้าตลอดช่วงที่ทำงานวิจัย และขอขอบคุณเพื่อนๆ ทุกคนที่ได้ให้ความช่วยเหลือในการทำวิทยานิพนธ์ และเรื่องอื่นๆ เสมอมา

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณมารดา รวมถึงทุกคนในครอบครัว ซึ่งสนับสนุนและให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฌ

สารบัญรูปภาพ ..... ญ

บทที่

1 บทนำ 1

1.1 ความเป็นมาและปัญหา 1

1.2 วัตถุประสงค์ 3

1.3 ขอบเขตการวิจัย 3

1.4 ขั้นตอนในการดำเนินงาน 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

1.6 ผลงานตีพิมพ์ 4

1.7 โครงสร้างวิทยานิพนธ์ 4

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 ทฤษฎีที่เกี่ยวข้อง 5

2.2 งานวิจัยที่เกี่ยวข้อง 12

3 การทดสอบการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ 15

3.1 การทำงานของระบบ 15

3.2 โครงสร้างของระบบ 16

3.3 วิธีที่ใช้ในการทดสอบ 18

3.4 สมมุติฐานที่ใช้ในการทดสอบ 25

3.5 ข้อมูลที่ใช้ในการทดสอบ 25

3.6 สภาพแวดล้อมที่ใช้ในการจำลอง 26

4 ผลการทดสอบ 27

4.1 ผลกระทบที่เกิดขึ้นกับระบบเมื่อมีการใช้การดึงข้อมูลล่วงหน้าแบบจับคู่ 27

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ (ต่อ)

4.2 การทดสอบประสิทธิภาพการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ 30
4.3 การวิเคราะห์ 37
5 สรุปผลการวิจัยและข้อเสนอแนะ 39
5.1 สรุปผลการวิจัย.....39
5.2 ปัญหาและข้อจำกัดที่ได้พบจากการวิจัย.....40
5.3 ข้อเสนอแนะ.....41
รายการอ้างอิง.....42
ภาคผนวก.....44
ภาคผนวก ก โปรแกรมที่ใช้ในการจำลอง.....45
ภาคผนวก ข การดักจับคำร้องขอจากแลนและการวัดแบนวิดท์ในเครือข่าย.....46
ภาคผนวก ก โปรแกรมที่ใช้ในการจำลอง.....45
ภาคผนวก ค ผลงานตีพิมพ์ 48
ประวัติผู้เขียนวิทยานิพนธ์.....50

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ 3.1 ลักษณะของข้อมูลการใช้เว็บที่ใช้ในการทดสอบ 26
ตารางที่ 4.1 เปรียบเทียบการจำลองแบนวิดท์ในระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้ากับระบบที่ใช้การดึงข้อมูลล่วงหน้าที่มีขอบเขตการดึงข้อมูลต่ำ 30
ตารางที่ 4.2 เปรียบเทียบวิธีที่ใช้ควบคุมแบนวิดท์ในระบบจำลองที่มีแบนวิดท์ 6 เมกกะบิต 32
ตารางที่ 4.3 เปรียบเทียบวิธีที่ใช้ควบคุมแบนวิดท์ในระบบจำลองที่มีแบนวิดท์ 4 เมกกะบิต 34

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญรูปภาพ

รูปที่ 1.1 ขั้นตอนการทำงานของการดึงข้อมูลล่วงหน้า....2
รูปที่ 2.1 ตำแหน่งของพร็อกซีเซิร์ฟเวอร์....6
รูปที่ 2.2 ตำแหน่งของโปรแกรมพรีเฟทช์....7
รูปที่ 2.3 วิธีการหาค่าของ p(B/A)....9
รูปที่ 2.4 วิธีการทำนายแบบวิเคราะห์เนื้อความ....10
รูปที่ 2.5 โครงร่างของระบบที่ใช้วิธีแบบท๊อบเท็น....11
รูปที่ 3.1 โครงสร้างของระบบเครือข่ายที่นำระบบดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์มาใช้งาน....15
รูปที่ 3.2 โครงสร้างของระบบดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์....16
รูปที่ 3.3 การเปลี่ยนแปลงขนาดของเฟทช์วินโดว์....18
รูปที่ 3.4 โครงสร้างของโปรแกรมจำลอง....19
รูปที่ 3.5 การทำงานของโปรแกรมจำลอง....20
รูปที่ 3.6 รายละเอียดแคชโมดูล....21
รูปที่ 3.7 โครงสร้างและการทำงานของแบนวิดท์โมดูล....23
รูปที่ 3.8 โครงสร้างและการทำงานของพรีเฟทช์โมดูลแบบเดิม....24
รูปที่ 3.9 โครงสร้างและการทำงานของพรีเฟทช์โมดูลแบบเปลี่ยนแปลงตามแบนวิดท์....24
รูปที่ 4.1ปริมาณการใช้แบนวิดท์ของระบบก่อนและหลังการดึงข้อมูลล่วงหน้า....28
รูปที่ 4.2 ฮิตเรตและเวลาเข้าถึงเอกสารของการจำลองแต่ละแบบ....29
รูปที่ 4.3 อัตราถ่ายโอนและเวลาในการดึงเอกสารที่ไม่พบในแคชของการจำลองแต่ละแบบ....29
รูปที่ 4.4 ปริมาณการใช้แบนวิดท์ของการจำลองแต่ละแบบ....31
รูปที่ 4.5 กราฟแสดงการใช้งานแบนวิดท์ของระบบจำลองที่มีแบนวิดท์ 4 เมกกะบิต....33
รูปที่ 4.6 ปริมาณการใช้แบนวิดท์ของแบบจำลองแต่ละแบบในแบนวิดท์แต่ละระดับ....35
รูปที่ 4.7 ฮิตเรทของการจำลองแต่ละแบบในแบนวิดท์แต่ละระดับ....35
รูปที่ 4.8 แสดงค่าความน่าจะเป็นและอัตราความสำเร็จของการดึงข้อมูลล่วงหน้าแต่ละแบบ....36
รูปที่ 4.9 เวลาที่ใช้ในการดึงเอกสารที่ไม่พบในแคชของการจำลองแต่ละแบบ....37
รูปที่ 4.10 ภาพจำลองการใช้แบนวิดท์ที่ว่างอยู่ของระบบควบคุมแบนวิดท์ทั้ง 2 วิธี....38

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

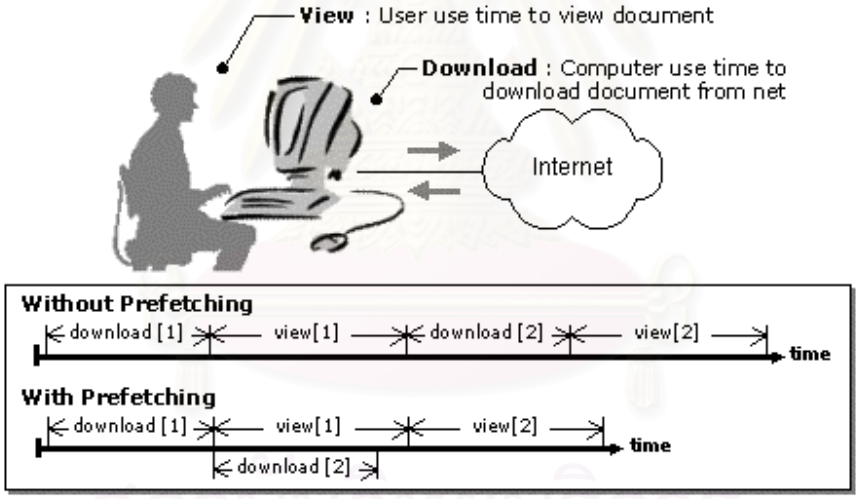
เนื่องจากในปัจจุบัน เวิลด์ไวด์เว็บได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้น ทั้งในด้านธุรกิจ การศึกษา และอื่นๆ ส่งผลให้มีผู้ใช้งานเพิ่มขึ้นอย่างรวดเร็ว ในขณะที่เครือข่าย (Network) ที่รองรับการใช้งานอินเทอร์เน็ต (Internet) ขยายตัวน้อยมาก เมื่อเทียบกับปริมาณข้อมูลจำนวนมหาศาลที่เกิดขึ้นเนื่องจากการขยายตัวของกลุ่มผู้ใช้งาน และการเปลี่ยนแปลงรูปแบบของเอกสาร (Document) ของเวิลด์ไวด์เว็บที่ถูกพัฒนาขึ้นจากเดิมซึ่งอยู่ในรูปของข้อความ (Text, HTML) ให้อยู่ในรูปของสื่อต่างๆ เช่น รูปภาพ (Image) เสียง (Wave) วีดีทัศน์ (Video) ซึ่งเป็นเอกสารที่มีขนาดใหญ่ ความแตกต่างนี้ทำให้เครือข่ายไม่สามารถรองรับการใช้งานในปัจจุบันได้ ทำให้บ่อยครั้งที่ผู้ใช้งานจะต้องใช้เวลานานเพื่อรับข้อมูลจากอินเทอร์เน็ต

จากสิ่งที่เกิดขึ้น ทำให้มีงานวิจัยจำนวนมากพยายามหาวิธีที่จะทำให้ผู้ใช้งานเวิลด์ไวด์เว็บ สามารถได้รับข้อมูลอย่างรวดเร็วผ่านทางเครือข่ายที่มีอยู่ในปัจจุบันได้ เช่น เว็บแคชชิง (Web Caching)[1] ซึ่งเป็นระบบที่เก็บเอกสารที่มีผู้ร้องขอจากอินเทอร์เน็ตไว้ เพื่อจะได้นำมาใช้อีกครั้งเมื่อมีผู้ร้องขออีกทำให้ไม่ต้องไปร้องขอจากอินเทอร์เน็ตใหม่ หรือ เรพลิเคท (Replicate) [2] ซึ่งจะดึงเอกสารจากเว็บเซิร์ฟเวอร์ (Web Server) มาเก็บไว้และทำการจ่ายเอกสารให้กับผู้ร้องขอที่อยู่ใกล้เคียง หรือ การดึงข้อมูลล่วงหน้าบนเว็บ (Web Prefetching) ซึ่งจะดึงเอกสารล่วงหน้า ก่อนที่ผู้ใช้งานจะร้องขอ หรือ การพัฒนาเว็บเซิร์ฟเวอร์ ให้สามารถจ่ายเอกสารให้ผู้ร้องขอได้รวดเร็วขึ้น [3] รวมทั้งการปรับปรุงโพรโทคอล เอ็ชทีทีพี (HTTP) [4] ให้มีประสิทธิภาพ เพื่อให้สามารถส่งข้อมูลได้รวดเร็วขึ้น

สำหรับการดึงข้อมูลล่วงหน้าบนเวิลด์ไวด์เว็บ จะช่วยลดเวลาในการรอคอยเอกสารของผู้ใช้งานโดยการดึงเอกสารที่คาดว่าผู้ใช้งานจะร้องขอเป็นตัวต่อไปมาเก็บไว้ในแคช การดึงข้อมูลล่วงหน้าบนเว็บมีหลักการ [5] ดังนี้คือ ในขณะที่ผู้ใช้งานกำลังอ่านเอกสารของเวิลด์ไวด์เว็บอยู่ ระบบจะทำนาย (Predict) ว่าเอกสารที่ผู้ใช้งานจะร้องขอต่อไปคือเอกสารใด แล้วดึงเอกสารมาเก็บไว้ในแคชล่วงหน้า เมื่อผู้ใช้งานทำการร้องขอเอกสารตัวต่อไปที่ระบบได้นำมาเก็บล่วงหน้าไว้แล้ว ผู้ใช้งานก็จะสามารถนำเอกสารจากแคชไปใช้ได้ทันที โดยไม่ต้องเสียเวลารอข้อมูลจากอินเตอร์เน็ตอีก ดังจะเห็นได้ในรูปที่ 1.1

อย่างไรก็ตามในการดึงเอกสารล่วงหน้าจะต้องใช้แบนวิดท์ (Bandwidth) เพิ่มขึ้นประมาณ 35 - 43 % จากการใช้งานตามปกติ [6] ทั้งนี้การที่ระบบทำนายเอกสารล่วงหน้าแล้วดึงเอกสารจากอินเทอร์เน็ตมาไว้ในแคชล่วงหน้าจำนวนหนึ่ง ซึ่งอาจจะทำนายได้ถูกต้องเพียงบางส่วนเท่านั้น กล่าวคือประมาณ 45 - 60 % [6] จะถูกส่งให้ผู้ใช้งาน ในกรณีที่ทำนายผิดพลาดจะทำให้เสียแบนวิดท์ที่ใช้ในการดึงเอกสารเหล่านั้นโดย

เปล่าประโยชน์ ดังนั้นในระบบของการดึงข้อมูลล่วงหน้าจึงมีตัวกำหนดว่าควรจะทำการดึงเอกสารล่วงหน้ามามากน้อยเพียงใดเรียกว่า ขอบเขตการดึงข้อมูลล่วงหน้า (Prefetch Threshold) ซึ่งเป็นค่าคงที่ โดยที่เมื่อส่วนทำนายของระบบได้ยูอาร์แอล (URL) ของเอกสารที่คาดว่าจะเป็นเอกสารที่ผู้ใช้จะร้องขอต่อไปมาแล้ว จะทำการคำนวณหาค่าความน่าจะเป็น (หรือค่าอื่นๆ ขึ้นอยู่กับวิธีการทำนายแต่ละวิธี) ให้กับแต่ละยูอาร์แอล ซึ่งยูอาร์แอลของเอกสารตัวใดที่มีค่าความน่าจะเป็นมากกว่าขอบเขตการดึงข้อมูลล่วงหน้า เอกสารนั้นก็จะถูกดึงมาเก็บที่แคช ดังนั้นถ้าตั้งขอบเขตการดึงข้อมูลล่วงหน้าให้มีค่าน้อย จะทำให้ระบบดึงเอกสารได้มากทำให้ผู้ใช้งานมีโอกาสพบเอกสารที่ต้องการมากขึ้นแต่ในขณะเดียวกันระบบจะใช้แบนวิดท์เพิ่มขึ้นเป็นจำนวนมากมาก แต่ถ้าตั้งค่าขอบเขตมีค่ามากจะทำให้ปริมาณการใช้แบนวิดท์ลดลงแต่จะทำให้โอกาสในการดึงเอกสารที่ตรงกับความต้องการของผู้ใช้ในอนาคตลดลงด้วยเช่นกัน โดยงานวิจัยส่วนใหญ่ [6,7,8,9,10,11,13] จะทำการวิจัยเพื่อหาวิธีทำนายที่มีประสิทธิภาพและจะทำการวิเคราะห์หาค่าขอบเขตการดึงข้อมูลล่วงหน้าที่เหมาะสมกับแต่ละวิธี ซึ่งค่าของขอบเขตการดึงข้อมูลล่วงหน้านี้จะอยู่ในรูปแบบต่างๆ กันเช่น จำนวนไฟล์ ค่าความน่าจะเป็น หรืออื่นๆ ขึ้นอยู่กับวิธีการทำนาย



รูปที่ 1.1 ขั้นตอนการทำงานของการดึงข้อมูลล่วงหน้า

อย่างไรก็ดีการศึกษาเหล่านี้จะทดลองในสภาพที่เครือข่ายมีแบนวิดท์ที่เชื่อมต่อไปยังอินเทอร์เน็ตไม่จำกัดจึงไม่เห็นผลกระทบที่เกิดขึ้นเมื่อเพิ่มปริมาณการใช้งานจำนวนมากเข้าไปในเครือข่ายที่มีแบนวิดท์จำกัด นอกจากนี้ ในสภาพความเป็นจริง การใช้งานแบนวิดท์จะมีการเปลี่ยนแปลงอยู่ตลอดเวลา ทำให้ภายในแต่ละวัน จะมีการใช้แบนวิดท์มากน้อยแตกต่างกันในแต่ละช่วงเวลา ดังนั้นค่าขอบเขตการดึงข้อมูลล่วงหน้าที่มีการกำหนดอย่างแน่นอนตายตัว จะไม่สามารถจะทำให้ระบบดึงข้อมูลล่วงหน้าทำงานได้เหมาะสมกับสภาพแวดล้อมของเครือข่ายที่เปลี่ยนแปลงอยู่ตลอดเวลาในแต่ละวัน เช่น ในสภาวะที่เครือข่ายมีการใช้งานแบน

วิดท์เต็มที่ การดึงข้อมูลล่วงหน้าอาจทำให้แบนวิดท์ไม่เพียงพอต่อการใช้งานโดยรวมของระบบได้ ในงานวิทยานิพนธ์นี้จึงได้นำเสนอระบบดึงข้อมูลล่วงหน้า ที่มีการปรับเปลี่ยนการทำงานไปตามสภาพของเครือข่าย โดยมุ่งเน้นการนำแบนวิดท์ที่ว่างอยู่มาใช้งาน และจำกัดการรบกวนการใช้งานอินเทอร์เน็ตตามปกติของผู้ใช้งานในเครือข่าย โดยจะสร้างโปรแกรมจำลองระบบจากข้อมูลการใช้งานเว็บจริง และทดลองโดยทำการปรับเปลี่ยนสภาวะการใช้งานแบนวิดท์ในรูปแบบต่างๆ เพื่อวิเคราะห์หาว่าระบบดึงข้อมูลล่วงหน้าที่พัฒนาขึ้นมาจะให้ผลแตกต่างกันอย่างไรภายใต้สภาวะแวดล้อมที่เปลี่ยนไป

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาระบบดึงข้อมูลล่วงหน้าที่ทำงานบนเว็บพร็อกซี ให้สามารถเปลี่ยนแปลงตามสภาพแวดล้อมของเครือข่ายและใช้งานแบนด์วิดท์เฉพาะส่วนที่เหลือจากการใช้งานตามปกติได้

1.3 ขอบเขตของการวิจัย

1. การปรับเปลี่ยนขอบเขตการดึงข้อมูลล่วงหน้าจะเปลี่ยนแปลงตามปริมาณแบนวิดท์ที่มีอยู่ในเครือข่าย
2. การทำการทำนายจะใช้วิธีทำนายเอกสารแบบจับคู่(Pair Wise) ในการจำลอง
3. การทดลองจะใช้แบบจำลองของระบบย่อยของการดึงข้อมูลล่วงหน้า (Prefetch Subsystem) ที่พัฒนาขึ้น
4. การทดลองจะมุ่งเน้นที่การวิเคราะห์ประสิทธิภาพของการดึงข้อมูลล่วงหน้าเท่านั้น

1.4 ขั้นตอนในการดำเนินงาน

1. ศึกษาการทำงานของระบบดึงข้อมูลล่วงหน้า การทำงานของพร็อกซีแคช และการวัดประสิทธิภาพของทั้งระบบดึงข้อมูลล่วงหน้าและพร็อกซีแคช
2. ทำการพัฒนาระบบดึงข้อมูลล่วงหน้าที่ขอบเขตสามารถเปลี่ยนแปลงตามสภาพการใช้งานแบนวิดท์ของเครือข่าย
3. เก็บรวบรวมข้อมูลการใช้เว็บ
4. พัฒนาโปรแกรมเพื่อใช้ในการจำลองระบบดึงข้อมูลล่วงหน้า
5. ทำการจำลองระบบดึงข้อมูลล่วงหน้าด้วยข้อมูลที่เก็บรวบรวมมา
6. ทดลองปรับเปลี่ยนสภาพแวดล้อมเครือข่ายและปรับปรุงระบบกำหนดขอบเขต
7. ศึกษาวิธีในการนำระบบที่จำลองขึ้นมาใช้งานจริง
8. วิเคราะห์และสรุปผล
9. จัดทำรายงาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

นำเสนอวิธีปรับปรุงพร็อกซีแคชให้มีประสิทธิภาพมากขึ้นด้วยการนำระบบดึงข้อมูลล่วงหน้าเข้ามาใช้งานซึ่งจะทำให้มีฮิตเรตเพิ่มขึ้น และผู้ใช้งานภายใต้พร็อกซีที่ใช้การดึงข้อมูลล่วงหน้าเสียเวลารอคอยเอกสารน้อยลง โดยที่การทำงานของระบบดึงข้อมูลล่วงหน้าไม่ไปรบกวนการใช้งานแบนวิดท์ตามปกติในเครือข่าย

1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของงานวิทยานิพนธ์นี้ได้ตีพิมพ์และนำเสนอในการประชุมวิชาการวิทยาการและวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 4 (The 4th National Computer Science and Engineering Conference (NCSEC 2000)) เมื่อวันที่ 16-17 พฤศจิกายน พ.ศ.2543 ในบทความเรื่อง Performance Evaluation of WWW Prefetching ผู้แต่งคือ ภรศิษฐ์ ธนวุฒิวัฒน์ และ อ.ดร. ณัฐวุฒิ หนูไพโรจน์

1.7 โครงสร้างวิทยานิพนธ์

ในบทต่อไปของวิทยานิพนธ์นี้จะกล่าวถึงทฤษฎีที่นำมาประยุกต์ใช้ในงานวิจัยและงานวิจัยที่เกี่ยวข้อง ส่วนในบทที่ 3 จะกล่าวถึงแนวคิดของงานวิทยานิพนธ์นี้รวมทั้งระบบจำลองที่วิทยานิพนธ์นี้ได้สร้างขึ้นเพื่อใช้ทดสอบงานวิจัย และรายละเอียดต่างๆในการทดสอบ ในบทที่ 4 จะกล่าวถึงผลที่ได้จากการทดสอบระบบที่ได้ออกแบบขึ้น และในบทสุดท้ายจะเป็นการสรุปผลของงานวิทยานิพนธ์และข้อเสนอแนะในการพัฒนาต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 เวลาที่ใช้ในการร้องขอเอกสาร (User Perceive Latency)

ในการร้องขอเอกสารจากเวิลด์ไวด์เว็บ กระบวนการที่เกิดขึ้นหลังจากที่ผู้ใช้งานได้สั่งให้โปรแกรมบราวเซอร์ (Browser) เรียกเอกสารเวิลด์ไวด์เว็บขึ้นมาดู คือ เครื่องไคลเอนท์ของผู้ใช้งานจะทำการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ผ่านโพรโทคอลทีซีพี (TCP) หลังจากติดต่อได้แล้ว ก็จะส่งคำร้องขอ (Request) ไปให้เว็บเซิร์ฟเวอร์ เมื่อได้รับคำร้องขอแล้วเว็บเซิร์ฟเวอร์ก็จะทำการประมวลผลและส่งผลลัพธ์ที่ได้กลับไปให้เครื่องไคลเอนท์ หลังจากนั้นเครื่องไคลเอนท์ก็จะทำการประมวลผลข้อมูลที่ได้ และแสดงข้อมูลออกบนหน้าจอของผู้ใช้งาน เวลาที่ถูกใช้ในขั้นตอนต่างๆ ที่กล่าวมานี้คือ เวลาที่ผู้ใช้งานจะต้องรอเอกสารจากอินเทอร์เน็ต หรือเวลาในการร้องขอเอกสาร (Retrieving Time) ซึ่ง เราสามารถแบ่งได้เป็น 2 ส่วนใหญ่ๆ [7] คือ เวลาที่ใช้ในการประมวลผล (Processing Time) ซึ่งก็คือเวลาที่เว็บเซิร์ฟเวอร์ใช้ในการประมวลผลเพื่อส่งข้อมูลกลับ และเวลาที่เครื่องไคลเอนท์ใช้ในการประมวลผลข้อมูลที่ได้รับให้ผู้ใช้งาน ส่วนที่ 2 คือเวลาที่ใช้ในการสื่อสาร (Communication Time) ซึ่งประกอบด้วย เวลาที่ใช้สร้างการเชื่อมต่อ (Propagation Time) กับเวลาที่ใช้ในการส่งข้อมูล (Transmission Time)

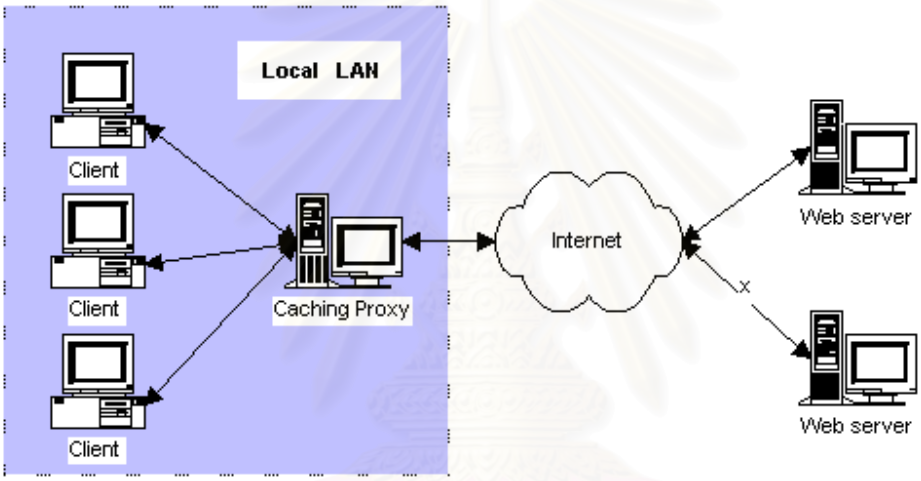
สำหรับเวลาที่ใช้ในการประมวลผลนั้นเราสามารถปรับปรุงได้โดยเพิ่มประสิทธิภาพของเครื่องทั้งด้านเว็บเซิร์ฟเวอร์และไคลเอนท์ แต่สำหรับเวลาที่ใช้ในการสื่อสารนั้น เวลาที่ใช้ในการส่งข้อมูลจะถูกปรับปรุงได้โดยเพิ่มประสิทธิภาพเครือข่ายอินเทอร์เน็ตซึ่งทำได้ยาก เนื่องจากต้องเสียค่ายใช้จ่ายสูง ส่วนเวลาที่ใช้สร้างการเชื่อมต่อจะเป็นเวลาที่ตายตัวไม่สามารถเปลี่ยนแปลงได้ ดังนั้นในการลดเวลาที่ใช้ในการสื่อสาร มักจะกระทำในรูปแบบของการพัฒนาวิธีการร้องขอและใช้งานเอกสารให้มีประสิทธิภาพ เช่น การทำเว็บแคชชิง การทำเรพลิเคท และ การดึงข้อมูลล่วงหน้า

การดึงข้อมูลล่วงหน้าจะช่วยลดเวลาในการรอคอยเอกสาร โดยการนำเอกสารมาเก็บไว้ล่วงหน้าในแคช ซึ่งอาจจะเป็นแคชภายในเครื่องของผู้ใช้งานหรือที่เว็บพร็อกซีก็ได้ขึ้นอยู่กับตำแหน่งของระบบดึงข้อมูลล่วงหน้า ซึ่งเวลาที่ใช้ในการดึงเอกสารจากแคชทั้งสองแบบจะน้อยกว่าเวลาที่ใช้ในการดึงเอกสารจากอินเทอร์เน็ตมาก

2.1.2 พร็อกซี แคชชิง(Proxy Caching)

พร็อกซี แคชชิง เป็นแนวความคิดที่จะให้มีเซิร์ฟเวอร์ อยู่ระหว่าง ผู้ใช้งานกับ เว็บเซิร์ฟเวอร์

ดังรูปที่ 2.1 เพื่อทำเว็บแคชชิงให้กับกลุ่มของผู้ใช้งานภายใต้พร็อกซีเซิร์ฟเวอร์นั้น โดย พร็อกซีเซิร์ฟเวอร์ จะทำหน้าที่เป็นเซิร์ฟเวอร์ที่คอยรับคำร้องขอจากผู้ใช้งาน เมื่อได้รับคำร้องขอแล้ว พร็อกซีเซิร์ฟเวอร์ก็จะไปเรียกเอกสารที่ผู้ใช้งานต้องการจากเว็บเซิร์ฟเวอร์ หลังจากนั้นก็จะทำการส่งเอกสารที่ได้ไปให้ผู้ใช้งานที่ร้องขอไว้และทำการเก็บเอกสารที่ได้มาเอาไว้ด้วย เมื่อผู้ใช้งานคนเดิมหรือคนอื่นร้องขอเอกสารตัวเดิมอีก พร็อกซีเซิร์ฟเวอร์ก็จะทำการส่งเอกสารที่เก็บไว้ให้ผู้ใช้งานเลย โดยไม่ต้องไปเอาเอกสารตัวเดิมจากเว็บเซิร์ฟเวอร์อีก จากการทำงานของพร็อกซีเซิร์ฟเวอร์นี้ทำให้ เว็บเซิร์ฟเวอร์ ไม่ต้องส่งเอกสารมาอีกครั้งซึ่งจะเป็นการลดภาระของเว็บเซิร์ฟเวอร์ลง และไม่ต้องสิ้นเปลืองแบนวิดท์ระหว่างพร็อกซีเซิร์ฟเวอร์ กับเว็บเซิร์ฟเวอร์อีก รวมทั้งผู้ใช้งานยังได้รับเอกสารด้วยความรวดเร็ว เนื่องจากระยะของ พร็อกซีเซิร์ฟเวอร์ อยู่ใกล้กับผู้ใช้งานมากกว่าเว็บเซิร์ฟเวอร์มากอีกด้วย

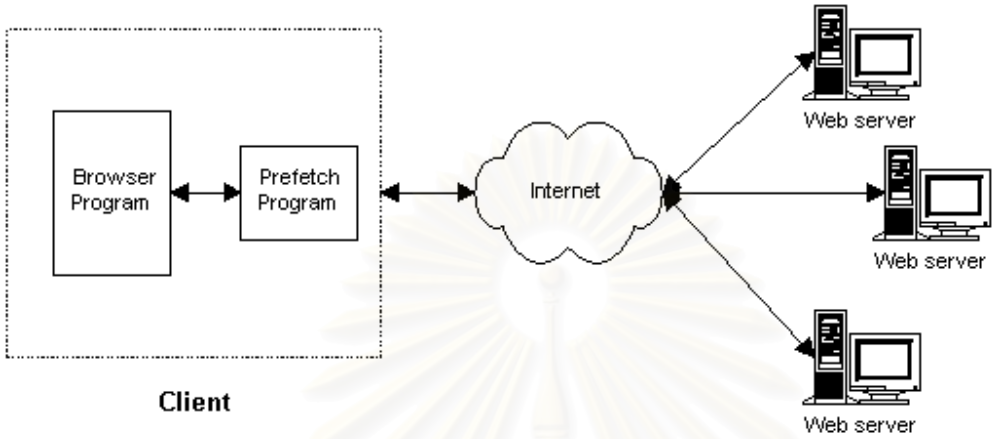


รูปที่ 2.1 ตำแหน่งของพร็อกซีเซิร์ฟเวอร์

2.1.3 **การดึงข้อมูลล่วงหน้าบนเว็บ** (Web Prefetching)

เมื่อผู้ใช้งานต้องการเอกสารจากเวิลด์ไวด์เว็บ ผู้ใช้งานก็จะร้องขอเอกสารผ่านโปรแกรมบราวเซอร์และรอให้บราวเซอร์ติดต่อกับเว็บเซิร์ฟเวอร์ และรับข้อมูลมาแสดงผลทางหน้าจอ หลังจากนั้นผู้ใช้งานก็จะใช้งานข้อมูลที่ได้จากเอกสาร และเมื่อพบเอกสารที่น่าสนใจก็จะทำการร้องขอเอกสารใหม่ จากขั้นตอนที่กล่าวมานี้จะสังเกตได้ว่าในช่วงที่ผู้ใช้งานกำลังดูเอกสารที่ได้มานั้น สายสัญญาณระหว่างเครื่องไคลเอนท์ของผู้ใช้งาน กับเครือข่ายอินเทอร์เน็ต ไม่ได้ถูกใช้งานเลย การดึงข้อมูลล่วงหน้า จะนำแบนวิดท์ (Bandwidth) ของสายสัญญาณที่ว่างอยู่นี้มาใช้ประโยชน์ โดยหลังจากที่โปรแกรมบราวเซอร์ได้รับเอกสารจากอินเทอร์เน็ต และแสดงให้ผู้ใช้งานแล้ว โปรแกรมดึงข้อมูลล่วงหน้าก็จะทำนาย (Prediction) ว่ามีเอกสารตัวใดที่ผู้ใช้งานน่าจะร้องขอต่อไปขึ้นมาจำนวนหนึ่ง โดยอาศัยข้อมูลจากสถิติที่โปรแกรมได้เก็บไว้ หรือจากเอกสารที่ผู้ใช้งานกำลังอ่านอยู่ แล้วก็จะทำการร้องขอเอกสารที่ทำนายไว้มาเก็บไว้ในแคช เมื่อผู้ใช้งานร้องขอเอกสารตัวต่อไปที่ตรง

กับเอกสารที่ได้ทำนายไว้ โปรแกรมก็จะนำเอกสารที่ถูกเก็บไว้ในแคชมาแสดงได้โดยทันที ไม่ต้องรอการส่งจากอินเทอร์เน็ต



รูปที่ 2.2 ตำแหน่งของโปรแกรมพรีเฟชท์

การดึงข้อมูลล่วงหน้า จะช่วยลดเวลาในการรอเอกสารของผู้ใช้งาน (User Perceived Latency) โดยใช้ช่วงเวลาที่ผู้ใช้งานกำลังอ่านเอกสารให้เป็นประโยชน์ แต่ในขณะเดียวกันการดึงข้อมูลล่วงหน้า ก็มีข้อเสียบางอย่างด้วย ในขั้นตอนที่โปรแกรมดึงข้อมูลล่วงหน้าทำการดึงเอกสารที่ได้ทำนายไว้ จะต้องใช้แบนวิดท์มากขึ้นกว่าปกติ และถ้าผู้ใช้งานร้องขอเอกสารตัวต่อไปที่ไม่ใช่เอกสารที่โปรแกรมได้ทำนายไว้จะทำให้เป็นการสิ้นเปลืองแบนวิดท์โดยเปล่าประโยชน์ นอกจากนี้ในขั้นตอนการทำนาย โปรแกรมจะต้องใช้ทรัพยากรของเครื่องเช่น ซีพียู (CPU) และหน่วยความจำ (Memory) ด้วย

จะเห็นได้ว่าถ้าการดึงข้อมูลล่วงหน้าถูกใช้อย่างไม่มีประสิทธิภาพจะทำให้สิ้นเปลืองทรัพยากรของระบบ ซึ่งประสิทธิภาพของการดึงข้อมูลล่วงหน้า จะขึ้นอยู่กับองค์ประกอบหลัก 2 ส่วนคือ วิธีการทำนาย (Prediction Algorithm) และ ขอบเขตการดึงข้อมูลล่วงหน้า (Prefetching Threshold) วิธีการทำนายจะเป็นตัวกำหนดที่มาของเอกสารที่จะถูกดึงล่วงหน้า ว่าจะนำมาจากไหน และเป็นตัวกำหนดว่า เอกสารแต่ละตัวจะมีความน่าจะเป็น (Probability) ที่จะถูกผู้ใช้งานร้องขอเท่าใด ส่วนขอบเขตการดึงข้อมูลล่วงหน้า จะเป็นตัวกำหนดไม่ให้มีการร้องขอเอกสารล่วงหน้ามากเกินไป โดยเอกสารที่มีค่าความน่าจะเป็นมากกว่าขอบเขตที่กำหนดไว้ ถึงจะถูกโปรแกรมร้องขอ นอกจากนี้ขอบเขตอาจจะเป็นตัวกำหนดถึงปริมาณไฟล์ และปริมาณแบนวิดท์ที่โปรแกรมจะร้องขอด้วย ในการสร้างระบบดึงข้อมูลล่วงหน้าที่มีประสิทธิภาพจะต้องอาศัยวิธีการทำนายที่แม่นยำ และมีการกำหนดขอบเขตที่เหมาะสมกับสภาพแวดล้อมของระบบที่จะนำการดึงข้อมูลล่วงหน้าไปใช้

นอกเหนือจากการนำระบบดึงข้อมูลล่วงหน้า มาใช้ในเครื่องไคลเอนท์ของผู้ใช้งานแล้ว เราสามารถนำระบบนี้ไปประยุกต์ใช้กับ แคชชิงพร็อกซีได้ เมื่ออยู่ในพร็อกซีเซิร์ฟเวอร์ เอกสารที่ถูกผู้ใช้งานคนใดคนหนึ่งดึงมาล่วงหน้า จะถูกเก็บอยู่ในแคช (Cache) ของพร็อกซีเซิร์ฟเวอร์ แทนที่จะอยู่ในเครื่องไคลเอนท์ เมื่อมีผู้ใช้งานคนอื่นร้องขอเอกสารตัวเดียวกันนี้ก็สามารถนำเอกสารที่ถูกเก็บไว้ไปใช้ได้ ทำให้ระบบสามารถใช้ประโยชน์จากเอกสารที่ได้ดึงมาล่วงหน้ามาเก็บไว้มากขึ้นด้วย นอกจากนี้ในกรณีที่ใช้วิธีการทำนายที่ทำนายโดยใช้สถิติการใช้งานเว็บของผู้ใช้ การดึงข้อมูลล่วงหน้าบนพร็อกซี จะทำให้สามารถทำนายเอกสารได้ถูกต้องมากขึ้นด้วยเพราะสถิติที่เก็บได้จะเป็นสถิติของผู้ใช้งานทั้งหมดในเครือข่ายนั้น

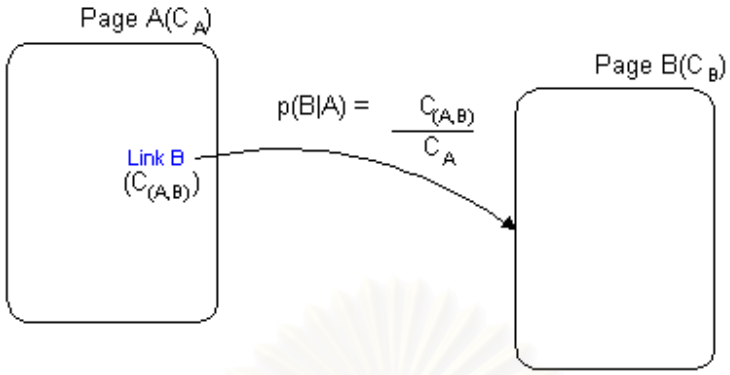
2.1.4 **วิธีการทำนาย** (Prediction Algorithm)

วิธีทำนายเอกสารของการดึงข้อมูลล่วงหน้ามีหลายวิธี แต่ละวิธีมีความแตกต่างกัน ทั้งในด้านการหารายชื่อของเอกสารที่ผู้ใช้งานจะร้องขอในอนาคต และการประเมินค่าความน่าจะเป็นของแต่ละเอกสาร ซึ่งจะขึ้นอยู่กับแนวความคิดของผู้วิจัยที่เสนอวิธีนั้นขึ้นมา ตัวอย่างเช่น

2.1.4.1 แบบจับคู่ (Pair Wise)

วิธีแบบจับคู่ [5,7,8,9,10] จะกำหนดตัวแปรชื่อ p(B|A) ขึ้นมา ซึ่งแสดงถึงความน่าจะเป็นที่เอกสาร B จะถูกผู้ใช้งานร้องขอต่อไปในอนาคตหลังจากที่ได้ทำร้องขอเอกสาร A ไปแล้ว โดยวิธีนี้จะอาศัยหลักการที่ว่า ปกติการดูเอกสารเวิลด์ไวด์เว็บ ผู้ใช้งานจะทำการคลิก (Click) จุดเชื่อมโยง (Link) ที่อยู่ในเอกสารที่กำลังอ่านอยู่เพื่อที่จะเลือกดูเอกสารหน้าใหม่ต่อไป หลังจากที่ได้ดูแล้วก็จะทำการคลิกเพื่อเลือกเอกสารใหม่จากหน้าที่กำลังดูนั้นอีก ซึ่งน้อยครั้งที่ผู้ใช้งานจะทำการพิมพ์ยูอาร์แอล (URL) โดยตรงเพื่อไปยังเอกสารหน้าต่อไป วิธีนี้จึงทำการเก็บสถิติจากการเลือกเอกสารของผู้ใช้งานไว้ เมื่อมีผู้ใช้งานร้องขอเอกสาร ระบบดึงข้อมูลล่วงหน้าก็จะทำนายเอกสารจากสถิติที่ได้เก็บไว้แล้วก็จะร้องขอเอกสารล่วงหน้ามาให้ผู้ใช้งานต่อไป

รูปที่ 2.3 แสดงถึงวิธีหาค่าของ p(B|A) ซึ่งจะต้องคำนวณจากประวัติการใช้งานของผู้ใช้งานแต่ละคนเพื่อให้ได้ข้อมูลที่ถูกต้อง การหาค่าของ p(B|A) จะต้องใช้ตัวนับ 2 ตัวคือ ตัวนับของเอกสาร (Page Counter) และตัวนับของการเชื่อมโยง (Link Counter) ยกตัวอย่างจากรูปที่ 3 ที่เอกสาร A จะมีตัวนับ C_A ซึ่งจะคอยนับทุกครั้งที่มีผู้ร้องขอเอกสาร A และถ้าในเอกสาร A มีจุดเชื่อมโยงไปยังหน้า B ก็จะเกิดตัวนับ C_{AB} ขึ้น ซึ่งจะเพิ่มขึ้นทุกครั้งที่มีการร้องขอเอกสาร B จากจุดเชื่อมโยงในเอกสาร A ค่าของ p(B|A) จะหาได้จากสูตร $p(B|A) = C_{AB} / C_A$ ซึ่งทุกครั้งที่มีการเรียกใช้เอกสาร A โปรแกรมดึงข้อมูลล่วงหน้าก็จะหารายชื่อเอกสารทั้งหมดที่เชื่อมโยงจากหน้า A และเลือกหน้าที่จะร้องขอล่วงหน้าโดยเปรียบเทียบกับค่าความน่าจะเป็นของแต่ละเอกสาร



รูปที่ 2.3 วิธีหาค่าของ p(B|A)

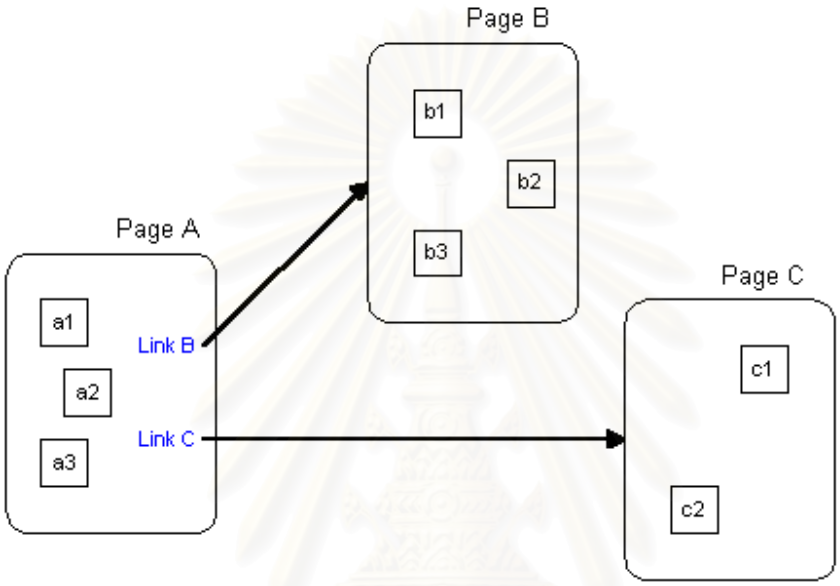
เราสามารถเก็บสถิติของการเชื่อมโยงเอกสารนี้ได้ 3 ที่คือ จากเครื่องไคลเอนท์ของผู้ใช้งาน จากพร็อกซีเซิร์ฟเวอร์ และจากเว็บเซิร์ฟเวอร์ ซึ่งแต่ละแหล่งจะได้สถิติที่มีรูปแบบที่แตกต่างกัน สถิติที่ได้จากเครื่องไคลเอนท์จะเป็นข้อมูลที่แสดงถึงความสนใจของผู้ใช้งานเครื่องไคลเอนท์เพียงคนเดียวเท่านั้น ส่วนสถิติที่ได้จากพร็อกซีเซิร์ฟเวอร์จะแสดงถึงแนวโน้มความสนใจของกลุ่มผู้ใช้งานที่ใช้พร็อกซีเซิร์ฟเวอร์นั้น ซึ่งจะครอบคลุมได้มากกว่า และถ้าเป็นสถิติที่ได้จากเว็บเซิร์ฟเวอร์ จะแสดงถึงรูปแบบการใช้งาน ของผู้ใช้งานทั้งหมดที่เข้ามาร้องขอเอกสารจากเว็บเซิร์ฟเวอร์นั้น แต่ในแบบสุดท้ายนี้จะไม่สามารถเก็บข้อมูลได้สมบูรณ์ เช่น ในกรณีที่จุดเชื่อมโยงภายในเอกสารนั้นเชื่อมโยงไปยังเอกสารของเว็บเซิร์ฟเวอร์อื่นๆ และในการที่จะใช้ข้อมูลจากเว็บเซิร์ฟเวอร์ จำเป็นที่จะต้องดัดแปลง โพรโทคอลเอชทีทีพี (HTTP) ที่ใช้ในปัจจุบันใหม่เพื่อให้เว็บเซิร์ฟเวอร์สามารถส่งรายชื่อเอกสารไปให้โปรแกรมดึงข้อมูลล่วงหน้าที่อยู่ปลายทางได้

2.1.4.2 แบบวิเคราะห์เนื้อความ (Content Parsing)

วิธีนี้ [6,11] จะใช้หลักการเดียวกับแบบจับคู่คือ ผู้ใช้งานส่วนใหญ่จะเรียกดูเอกสารหน้าใหม่จากจุดเชื่อมโยงภายในเอกสารที่กำลังอ่านอยู่ แต่วิธีนี้ไม่ได้ใช้ข้อมูลจากการเก็บสถิติเหมือนวิธีแรกแต่ใช้วิธีวิเคราะห์เนื้อความของเอกสารว่าในเอกสารนั้นมีจุดเชื่อมโยงอะไรบ้าง แล้วนำจุดเชื่อมโยงเหล่านี้ไปใช้เปรียบเทียบต่อไป

ด้วยวิธีนี้ การทำนายจะเกิดขึ้นโดยโปรแกรมดึงข้อมูลล่วงหน้า จะทำการอ่าน (Parse) เอกสารที่ผู้ใช้ได้ร้องขอมา เพื่อหาจุดเชื่อมต่อของเอกสารถัดไป โดยจะแบ่งรายชื่อของเอกสารที่ได้ออกเป็น 2 ประเภท คือ รูปภาพของเอกสาร (In-line Image) ซึ่งเป็นรูปภาพที่ถูกแสดงภายในเอกสารนั้น กับเอกสารเชื่อมโยง (Referenced Page) ซึ่งเป็นเอกสารที่จะถูกร้องขอต่อไปถ้าผู้ใช้งานคลิกจุดเชื่อมโยงในเอกสารนั้น โดยโปรแกรม จะทำการร้องขอเฉพาะเอกสารเชื่อมโยง เท่านั้น เพราะหลังจากที่บราวเซอร์ได้รับเอกสารหน้าแรกแล้ว ตัวบราวเซอร์ก็จะทำการร้องขอเอกสารรูปภาพเองอยู่แล้ว และเมื่อโปรแกรมดึงข้อมูลล่วงหน้า ได้เอกสารเชื่อมโยงมาแล้วโปรแกรมก็จะทำการดึงรูปภาพของเอกสารนั้นมาด้วย ยกตัวอย่างจากรูปที่ 2.4 เมื่อมีผู้ใช้งานร้องขอเอกสาร A หลังจากที่ได้รับเอกสาร A แล้ว โปรแกรมก็จะทำการอ่านเอกสาร A และหา

รายชื่อเอกสารที่จะร้องขอล่วงหน้า ซึ่งจะได้จุดเชื่อมโยงของเอกสาร a1, a2, a3, B และ C ซึ่งตัวโปรแกรมก็จะร้องขอเฉพาะเอกสาร B และ C และหลังจากที่ได้เอกสาร B และ C และ โปรแกรมก็จะทำการอ่านไฟล์ทั้ง 2 เพื่อหารูปของเอกสารทั้งสองหน้าและจะได้จุดเชื่อมโยง b1, b2, b3, ... ที่เป็นรูปภาพและจะทำการร้องขอรูปภาพเหล่านี้มาเก็บไว้ล่วงหน้าเช่นกัน



รูปที่ 2.4 วิธีการทำนายของแบบวิเคราะห์เนื้อความ

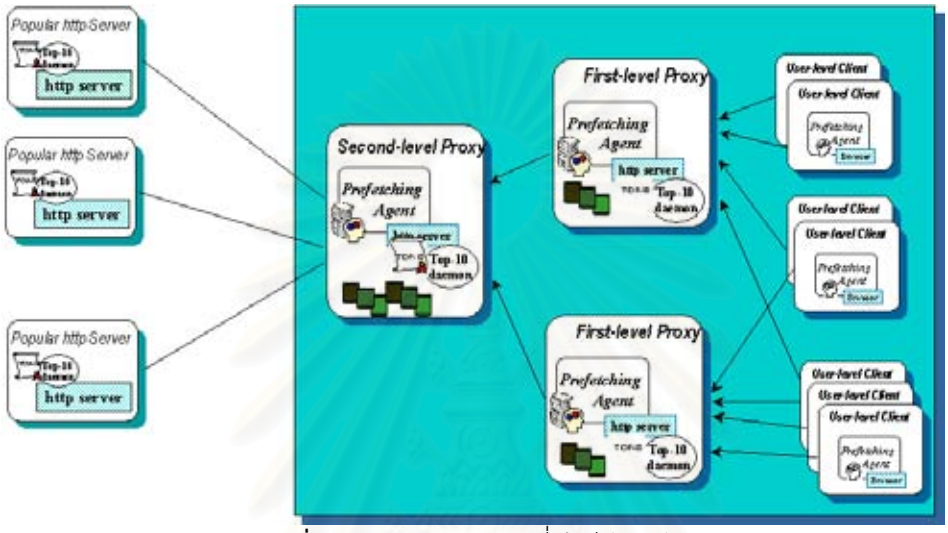
วิธีวิเคราะห์เนื้อความนี้จะใช้การประมวลผลค่อนข้างมากเพราะต้องค้นหาจุดเชื่อมโยงภายในเอกสารที่จะทำการดึงล่วงหน้า แต่จะไม่มีการเก็บข้อมูลเพื่อใช้ในการทำนายจึงไม่เปลืองหน่วยความจำของระบบ ซึ่งแตกต่างกับวิธีจับคู่ที่ใช้หน่วยความจำปริมาณมากในการเก็บสถิติการใช้งานเว็บของผู้ใช้ แต่วิธีวิเคราะห์แบบเนื้อความนี้ยังมีการศึกษาเกี่ยวกับวิธีในการเลือกเอกสารค่อนข้างน้อยและวิธีที่มีอยู่ยังไม่มีประสิทธิภาพดีพอ ซึ่งแตกต่างจากวิธีจับคู่ที่ถูกใช้ในงานวิจัยจำนวนมาก ทำให้มีวิธีเลือกเอกสารที่หลากหลายและมีประสิทธิภาพมากกว่า

2.1.4.3 แบบท็อปเทน (TOP-10)

วิธีท็อปเทน [12] จะอาศัยหลักการว่าเอกสารที่ได้รับความนิยม (มีการเรียกบ่อย) มีโอกาสที่จะถูกร้องขอมาก และส่วนที่จะรู้ว่าเอกสารใดได้รับความนิยมบ้างคือเซิร์ฟเวอร์ที่เป็นผู้ให้บริการ ด้วยวิธีนี้เซิร์ฟเวอร์จะเก็บข้อมูลว่าเอกสารใดในเซิร์ฟเวอร์ที่ถูกเรียกใช้มาก เมื่อมีผู้ใช้งานมาร้องขอเอกสาร เซิร์ฟเวอร์ก็จะส่ง

เอกสารที่ผู้ใช้งานต้องการให้ และส่งข้อมูลแนะนำ (Hint) ไปด้วยว่ามีเอกสารใดที่ได้รับความนิยมบ้าง เพื่อให้ผู้ใช้งานได้ทำการดึงเอกสารล่วงหน้าได้

E.P. Markatos และ C.E. Chronaki [12] ได้ใช้วิธีนี้ในการดึงข้อมูลล่วงหน้าโดยมีโครงร่างดังรูปที่ 2.5



รูปที่ 2.5 โครงร่างของระบบที่ใช้วิธีท็อปเท็น

จากรูปที่ 2.5 [12] จะเห็นได้ว่าในระบบจะประกอบด้วยโปรแกรม 2 ชุดคือ ท็อปเทนดีมอน (TOP-10 Daemon) กับ พรีเฟทช์เอเจนท์ (Prefetch Agent) โดย ท็อปเทนดีมอนจะทำงานอยู่ที่เซิร์ฟเวอร์โดยจะเก็บข้อมูลว่าเอกสารตัวใดบ้างที่ได้รับความนิยม และหลังจากที่มีการร้องขอเอกสารมายังเซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำการส่งเอกสารไปให้และจะส่งข้อมูลแนะนำไปให้ว่ามีเอกสารใดที่ได้รับความนิยมบ้าง พรีเฟทช์เอเจนท์ที่อยู่บนเครื่องของผู้ร้องขอ เมื่อพรีเฟทช์เอเจนท์ได้รับข้อมูลแล้วก็จะทำการดึงข้อมูลล่วงหน้าจากเซิร์ฟเวอร์มาเก็บไว้ที่แคชเพื่อให้ผู้ใช้งานได้ใช้ข้อมูลเมื่อร้องขอครั้งต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.1.1.5 ค่าที่ใช้ในการวัดประสิทธิภาพ

สำหรับการเปรียบเทียบประสิทธิภาพการดึงข้อมูลล่วงหน้า ตัวแปรที่งานวิทยานิพนธ์นี้ใช้เป็นหลักคือ แบนวิดท์ อัตราความสำเร็จ (Success Rate) ฮิตเรท (Hit Rate) และ ไบท์ฮิตเรท (Byte Hit Rate) ช่วงเวลาเข้าถึง (Access Time) และ อัตราถ่ายโอน (Transfer rate) โดย

แบนวิดท์ = ปริมาณข้อมูลที่ระบบได้ร้องขอจากอินเทอร์เน็ต

$$\text{ฮิตเรท} = \frac{\text{จำนวนคำร้องขอที่พบในพร็อกซี}}{\text{จำนวนคำร้องขอทั้งหมดที่ส่งมายังพร็อกซี}}$$

$$\text{ไบท์ฮิตเรท} = \frac{\text{ปริมาณข้อมูลของคำร้องขอที่พบในพร็อกซี}}{\text{ปริมาณข้อมูลของคำร้องขอทั้งหมด}}$$

$$\text{อัตราความสำเร็จ} = \frac{\text{จำนวนเอกสารที่ทำนายถูกต้อง}}{\text{จำนวนเอกสารที่ทำการดึงมาล่วงหน้าทั้งหมด}}$$

ช่วงเวลาเข้าถึง = ช่วงเวลาตั้งแต่ผู้ใช้งานร้องขอจนได้รับเอกสารทั้งหมด

อัตราถ่ายโอน = อัตราการส่งข้อมูล(ไบท์ต่อวินาที)ของแต่ละคำร้องขอ

เนื่องจากระบบดึงข้อมูลล่วงหน้าทำให้เกิดการใช้งานแบนวิดท์เพิ่มขึ้นจากการใช้งานตามปกติทำให้แบนวิดท์ที่ถูกระบบดึงข้อมูลล่วงหน้าใช้เป็นปัจจัยที่สำคัญตัวหนึ่งที่ใช้วัดประสิทธิภาพการทำงาน และในงานวิทยานิพนธ์นี้ ระบบดึงข้อมูลล่วงหน้าจะถูกนำมาใช้ในพร็อกซีแคช ดังนั้นฮิตเรท และไบท์ฮิตเรท ซึ่งเป็นตัววัดประสิทธิภาพของพร็อกซี จึงถูกนำมาใช้เพื่อบอกว่าระบบจะสามารถเพิ่มประสิทธิภาพของพร็อกซีได้มากขึ้นเพียงใด อัตราความสำเร็จจะเป็นค่าที่ใช้บอกความแม่นยำโดยตรงของวิธีการทำนายแต่ละวิธี รวมถึงการกำหนดขอบเขตการดึงข้อมูลล่วงหน้าที่เหมาะสมด้วย ส่วนช่างเวลาเข้าถึง และอัตราถ่ายโอนจะเป็นตัวแปรที่บอกถึงผลกระทบของการดึงข้อมูลล่วงหน้าที่มีต่อระบบ

2.2 งานวิจัยที่เกี่ยวข้อง

ในปัจจุบันมีงานวิจัยจำนวนมากที่ทำการศึกษาเกี่ยวกับวิธีการดึงข้อมูลล่วงหน้าบนเว็บ ทำให้เกิดวิธีในการดึงข้อมูลล่วงหน้าที่แตกต่างกันทั้งในด้านแหล่งข้อมูล ที่ตั้งของระบบ และวิธีทำนาย บางงานวิจัยจะให้เว็บเซิร์ฟเวอร์เป็นแหล่งข้อมูล งานวิจัย [7,8,9] จะให้เว็บเซิร์ฟเวอร์เก็บสถิติการใช้งานที่เข้ามายังเว็บเซิร์ฟเวอร์ แล้วใช้วิธีที่ดัดแปลงจากวิธีทำนายแบบจับคู่ ทำนายแล้วส่งคำแนะนำ (Hint) ไปบอกยังพร็อกซีหรือเครื่องไคลเอนท์ ว่าควรจะดึงเอกสารใดล่วงหน้า ส่วน [12] จะให้เว็บเซิร์ฟเวอร์เป็นผู้แนะนำเช่นกัน แต่จะส่งรายชื่อของเอกสารที่ถูกเรียกใช้มากที่สุดไปแทน งานวิจัย [13] นำระบบดึงข้อมูลล่วงหน้ามาใช้กับโพรโทคอลเอฟทีพี (Ftp Protocol) โดยทำนายจากตำแหน่งของเอกสารที่อยู่ในสารบบ (Directory) ซึ่งเซิร์ฟเวอร์จะเป็นผู้ทำนายเอกสารและเป็นผู้ส่งไฟล์ไปให้ผู้ใช้งานเลย แต่บางงานวิจัยเสนอให้เก็บข้อมูลที่เว็บพร็อกซีและให้พร็อกซีเป็นตัวดึงเอกสารล่วงหน้าเอง เช่น [6,11,15] โดย [11] จะใช้วิธีวิเคราะห์เนื้อความในการทำนายเอกสาร ส่วน [6] ใช้ ทั้งวิธีจับคู่และวิเคราะห์เนื้อความร่วมกันในการทำนายเอกสาร ส่วนงานวิจัยที่ดึงข้อมูลล่วงหน้าโดยควบคุมปริมาณการดึงเอกสารเพื่อไม่ให้ส่งผลกระทบต่อแบนวิดท์ของระบบมีดังนี้

2.2.1 The Network Effects of Prefetching [15]

งานวิจัยนี้ทำการศึกษาถึงผลกระทบของการดึงข้อมูลล่วงหน้าที่มีต่อเครือข่ายของระบบโดยค่าที่ใช้ในการวัดสภาพของระบบคือ ค่ากลางของขนาดคิว (Mean Queue Length) ของอุปกรณ์จัดเส้นทาง (Router) ของเครือข่าย โดยใช้สมมติฐานว่าถ้าขนาดของคิวมีมากหมายความว่าแบนด์วิดท์ไม่เพียงพอทำให้แต่ละกลุ่มข้อมูล (Packet) ที่ถูกส่งผ่านไปยังอุปกรณ์จัดเส้นทางจะต้องรออยู่ในคิวนานทำให้ผู้ใช้งานได้รับข้อมูลช้าลง ผู้วิจัยได้ทำการทดสอบโดยการจำลองระบบขึ้นมาซึ่งได้ผลว่าการดึงข้อมูลล่วงหน้าแบบปกติทำให้ขนาดของคิวเพิ่มขึ้นมากซึ่งส่งผลให้ระบบช้าลง และผู้วิจัยได้เสนอระบบดึงข้อมูลล่วงหน้าใหม่เรียกว่า เรท คอนโทรลพรีเฟทชิง (Rate-Controlled Prefetching) ซึ่งระบบนี้จะอาศัยช่วงเวลาการใช้งานของผู้ใช้งานที่แบ่งออกเป็น 2 ช่วงคือช่วงที่มีการถ่ายโอน (Transfer) เอกสาร(ON Time Duration) กับช่วงใช้งานเอกสาร(OFF Time Duration) ซึ่งไม่มีการใช้แบนด์วิดท์ ระบบนี้จะควบคุมไม่ให้การดึงข้อมูลล่วงหน้าใช้แบนวิดท์เต็มที่ แต่จะจำกัดอัตราการส่งกลุ่มข้อมูลให้กระจายไปตลอดช่วงเวลาของช่วงใช้งานเอกสารโดยจะกำหนดขนาดวินโดว์ของทีซีพี (TCP Window Size) จากจำนวนกลุ่มข้อมูล เวลาที่ใช้ในการถ่ายโอนแต่ละกลุ่มข้อมูล และเวลาช่วงใช้งานเอกสาร ซึ่งจากการทดสอบจะพบว่า ระบบใหม่นี้สามารถลดค่ากลางของขนาดคิวของอุปกรณ์จัดเส้นทางได้เนื่องจากจะเกิดการดึงเอกสารล่วงหน้าก่อนการร้องขอข้อมูลจริงโดยใช้อัตราการถ่ายโดนข้อมูลต่ำ

งานวิจัยนี้คล้ายคลึงกับวิทยานิพนธ์ในด้านการควบคุมการดึงข้อมูลล่วงหน้าไม่ให้มีการใช้แบนด์วิดท์มากเกินไป แต่ต่างในด้านของจุดที่พัฒนากับวิธีการควบคุม โดยในงานวิจัยนี้จะทำงานที่โปรแกรมบราวเซอร์ของผู้ใช้งาน และจะทำการควบคุมอัตราการส่งข้อมูลแต่ละคำร้องขอ แต่ในวิทยานิพนธ์นี้ระบบที่สร้างขึ้นจะทำงานร่วมกับพร็อกซีเซิร์ฟเวอร์ และจะควบคุมปริมาณการส่งคำร้องขอแทน โดยมีการวัดสภาพแวดล้อมของเครือข่ายเพื่อใช้ในการควบคุมด้วย

2.2.2 On Bandwidth Smoothing [14]

งานวิจัยนี้มีแนวความคิดว่าในระบบเครือข่ายขนาดใหญ่จะมีปริมาณการใช้แบนวิดท์แต่ละช่วงจะไม่เท่ากันโดยจะแบ่งเป็นช่วงที่มีการใช้งานมาก (Peak Period) และช่วงที่มีการใช้งานน้อย (Off-peak Period) และจะลดปริมาณการใช้แบนวิดท์ในช่วงที่มีการใช้งานมากโดยจะทำการดึงข้อมูลล่วงหน้าในช่วงที่มีการใช้งานน้อยและเก็บไว้ในแคชก่อน ในงานวิจัยนี้จะแบบเอกสารที่สามารถดึงข้อมูลล่วงหน้าออกเป็น 3 ประเภท คือ เอกสารที่ผู้ใช้งานเคยพบ (Seen Prefetchable object) หรือเอกสารที่ผู้ใช้งานมีการเรียกใช้ในช่วงเวลานั้น และเอกสารที่พร็อกซี เซิร์ฟเวอร์เคยพบ (Seen-server Prefetchble object) หรือเอกสารที่มีอยู่ในข้อมูลการใช้เว็บของช่วงเวลาในอดีต และ เอกสารที่ผู้ใช้งานและเซิรฟ์เวอร์ไม่เคยพบ (Unseen-server prefetchable object) หรือเอกสารที่ไม่เคยถูกเรียกเข้ามาที่พร็อกซีเซิร์ฟเวอร์เลย ซึ่งในงานวิจัยนี้จะวิเคราะห์อัตราส่วนของเอกสารทั้ง 3 แบบ เมื่อทำการดึงข้อมูลล่วงหน้า โดยในงานวิจัยนี้จะใช้วิธีของกลไกการเรียนรู้ (

Machine Learning) โดยจะทำการฝึก (train) ส่วนทำนายโดยใช้ข้อมูลการใช้เว็บที่ผ่านๆ มาโดยข้อมูลที่ใช้ในการเรียนรู้เพื่อตัดสินใจจะใช้ อายุของเอกสาร ชนิดของเอกสาร ขนาด ซื่อเซิร์ฟเวอร์ ความลึกของตำแหน่งเอกสาร เป็นต้น และจะนำมาใช้งานโดยจะทำนายเอกสารจากข้อมูลการใช้เว็บของช่วงที่มีการใช้งานมากช่วงที่ผ่านมา และใช้ช่วงเวลาที่มีการใช้งานน้อยเพื่อทำการดึงข้อมูลมากเก็บไว้ในแคชเพื่อจะลดปริมาณการใช้งานของช่วงที่มีการใช้งานมากช่วงถัดไป

งานวิจัยนี้จะคล้ายคลึงกับวิทยานิพนธ์ในแง่ที่มีการนำการดึงข้อมูลล่วงหน้ามาใช้ร่วมกับแคช และมีการคำนึงถึงสภาพแวดล้อมของเครือข่าย แต่ต่างกันในด้านของวิธีการทำนายและช่วงเวลาการทำนาย โดยในงานวิจัยนี้จะทำนายเอกสารจากข้อมูลที่ผ่านไปแล้ว แต่ในวิทยานิพนธ์นี้จะทำนายเอกสารจากคำร้องขอที่ผู้ใช้งานร้องของมาที่พร็อกซีเซิร์ฟเวอร์

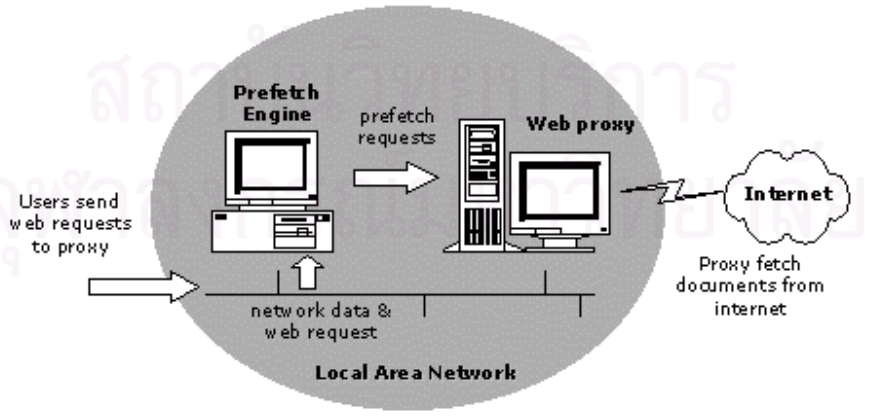
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การทดสอบระบบดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์

ในบทที่ 2 ได้กล่าวถึงปัญหาของการดึงข้อมูลล่วงหน้าไว้ว่า ถ้ากำหนดขอบเขตการพรีเฟทช์ได้ไม่เหมาะสมจะทำให้ระบบสิ้นเปลืองแบนวิดท์โดยเปล่าประโยชน์ ซึ่งในความเป็นจริงระบบเครือข่ายมีแบนวิดท์จำกัด ดังนั้นคำร้องขอจำนวนมากที่ถูกดึงข้อมูลล่วงหน้าจะส่งผลให้ระบบเครือข่ายต้องรับภาระในการจัดการคำร้องขอที่เพิ่มขึ้นมากและปริมาณการใช้งานแบนวิดท์ที่เพิ่มขึ้นมากซึ่งจะทำให้ระบบมีแบนวิดท์ไม่เพียงพอต่อการให้บริการตามปกติ จะส่งผลให้ประสิทธิภาพโดยรวมของระบบลดลง นอกจากนี้ในกรณีที่ปริมาณแบนวิดท์ที่ใช้ในการดึงข้อมูลล่วงหน้ากับขนาดของแบนวิทด์ของเครือข่ายแตกต่างกันมาก ประสิทธิภาพของระบบอาจจะต่ำกว่าระบบที่ไม่ได้นำการดึงข้อมูลล่วงหน้าไปใช้ได้

นอกจากปัญหาของการใช้งานแบนวิดท์แล้ว อีกปัญหาหนึ่งที่สำคัญคือ ปริมาณการใช้งานแบนวิดท์ที่เปลี่ยนแปลงอยู่ตลอดเวลา การเปลี่ยนแปลงนี้จะทำให้ขอบเขตการพรีเฟทช์ซึ่งเป็นค่าคงที่ ไม่สามารถควบคุมการดึงข้อมูลล่วงหน้าให้ทำงานได้อย่างเหมาะสมในแต่ละช่วงเวลาได้ ดังนั้นในวิทยานิพนธ์นี้ เราได้พัฒนารูปแบบการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ซึ่งจะวัดปริมาณแบนวิดท์ที่ถูกใช้งานในระบบและนำแบนวิดท์ส่วนที่ว่างอยู่มาใช้ในการดึงข้อมูลล่วงหน้า ทำให้สามารถลดเวลาการรอคอยเอกสารที่ทำนายได้โดยไม่ทำให้ประสิทธิภาพโดยรวมของระบบลดลง โดยในบทนี้จะกล่าวถึงรายละเอียดของรูปแบบการดึงข้อมูลล่วงหน้าที่พัฒนาขึ้นมา รวมทั้งการทดสอบประสิทธิภาพเปรียบเทียบกับระบบที่มีอยู่เดิมในสภาวะแวดล้อมต่างๆ

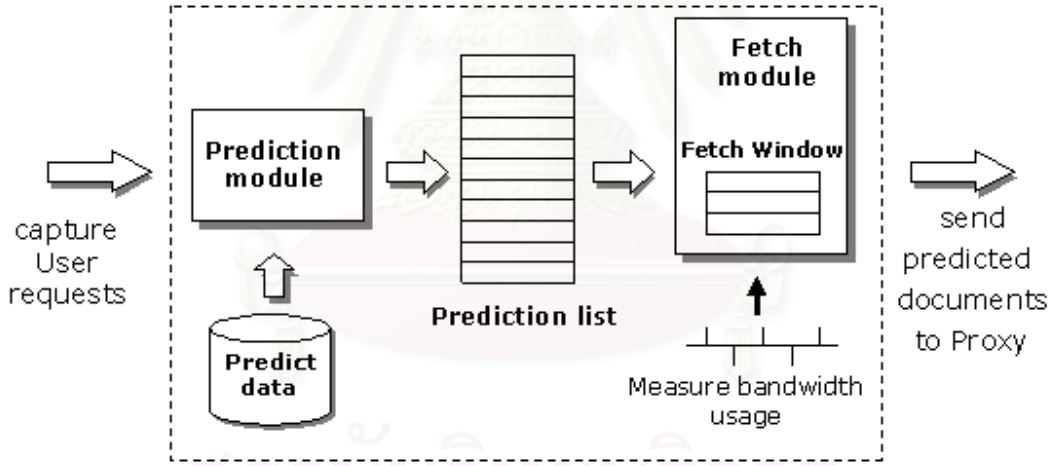
3.1 การทำงานของระบบ



รูปที่ 3.1 โครงสร้างของระบบเครือข่ายที่นำระบบดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์มาใช้งาน

จากรูปที่ 3.1 จะเห็นได้ว่าพรีเฟทช์เอนจิน (Prefetch Engine) หรือเครื่องที่ใช้ดึงข้อมูลล่วงหน้าจะอยู่ในเครือข่ายแลน (LAN) เดียวกับเว็บพร็อกซี เพราะระบบที่พัฒนาขึ้นจะต้องวัดปริมาณการใช้งานแบนวิดท์ของเครือข่ายตลอดเวลา และ จะดักจับคำร้องขอที่ผู้ใช้งานส่งมายังพร็อกซี เพื่อใช้ในการทำนายเอกสาร หลังจากที่ได้ ยูอาร์แอลของเอกสารที่จะทำการดึงข้อมูลล่วงหน้าแล้ว พรีเฟทช์เอนจินร้องขอเอกสารไปยังพร็อกซี เพื่อให้พร็อกซีนำเอกสารจากอินเตอร์เน็ตมาเก็บยังแคช เมื่อผู้ใช้งานทำการร้องขอเอกสารที่ได้ทำนายไว้ พร็อกซีก็จะนำเอกสารในแคชไปให้ผู้ใช้งานได้ทันที การแยกระบบดึงข้อมูลล่วงหน้าออกจากเว็บพร็อกซี เป็นการปรับปรุงข้อเสียของการดึงข้อมูลล่วงหน้าที่ใช้ทรัพยากรของเครื่อง (ซีพียู, หน่วยความจำ) เป็นจำนวนมาก ซึ่งทำให้เซิร์ฟเวอร์ที่มีระบบดึงข้อมูลล่วงหน้าทำงานอยู่มีประสิทธิภาพลดลง นอกจากนี้วิธีนี้ยังสามารถนำไปใช้งานได้ง่ายเพราะไม่ต้องปรับปรุง โพรโทคอลเอชทีทีพี (HTTP Protocol) หรือ องค์ประกอบอื่นๆ (เว็บเซิร์ฟเวอร์, พร็อกซีเซิร์ฟเวอร์, โปรแกรมบราวเซอร์) ในเครือข่าย เพราะพรีเฟทช์เอนจินจะทำงานเหมือนผู้ใช้งานคนหนึ่งในระบบเครือข่าย

3.2 โครงสร้างของระบบ



รูปที่ 3.2 โครงสร้างของระบบดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์

จากรูปที่ 3.2 เราจะเห็นโครงสร้างและขั้นตอนการทำงานของระบบ เมื่อได้รับยูอาร์แอลที่ได้จากการดักจับคำร้องขอของผู้ใช้งานแล้ว พรีดิกชันโมดูล (Prediction Module) ก็จะทำนายยูอาร์แอลของเอกสารจากข้อมูลสถิติที่เก็บไว้ และจะส่งยูอาร์แอลที่ทำนายได้ไปไว้ยัง พรีดิกชันลิสต์ (Prediction List) ซึ่งยูอาร์แอลที่อยู่ในพรีดิกชันลิสต์จะเรียงลำดับตามค่าความน่าจะเป็นที่จะถูกร้องขอเป็นตัวต่อไปของแต่ละยูอาร์แอล และแต่ละยูอาร์แอลจะถูกกำหนดอายุไว้ด้วย โดยจะกำหนดให้มีค่าเท่ากับเวลาที่ผู้ใช้งานใช้ดูเอกสาร (จากการเก็บสถิติการใช้งานเว็บของวิทยานิพนธ์นี้ เวลาเฉลี่ยในการดูเอกสารแต่ละครั้งของผู้ใช้งานคือ 72 วินาที)

ถ้ายูอาร์แอลใดที่หมดอายุก็จะไม่ถูกดึงข้อมูลล่วงหน้า เพราะเอกสารที่ดึงล่วงหน้าจะใช้ได้ผลในช่วงเวลาก่อนที่ผู้ใช้งานจะอ่านเอกสารถัดไปเท่านั้น

เฟทช์โมดูล (Fetch Module) จะเป็นตัวควบคุมการดึงข้อมูล โดยจะดึงข้อมูลที่มีค่าความเป็นไปได้มากที่สุดจากพรีดิกชันลิสต์ และร้องขอไปยังพร็อกซี เฟทช์โมดูลจะควบคุมปริมาณการส่งคำร้องขอด้วยเฟทช์วินโดว์ (Fetch window) เฟทช์วินโดว์จะเป็นตัวกำหนดจำนวนไฟล์ที่เฟทช์โมดูลสามารถดึงข้อมูลจากอินเตอร์เน็ตพร้อมกันได้ โดยขนาดของเฟทช์วินโดว์จะขึ้นอยู่กับปริมาณคำร้องขอที่สามารใช้งานได้ (Available Request : Ra) โดยค่า Ra สามารถหาได้จากปริมาณการใช้งานแบนวิดท์ที่เฟทช์โมดูลวัดจากระบบเครือข่ายและคำนวนจากสูตร

$$Ra = \frac{(Mbw - Cbw)}{Tr} \quad (1)$$

โดย

Ra คือ ปริมาณคำร้องขอที่ระบบดึงข้อมูลล่วงหน้าสามารถใช้ได้

Mbw คือ ปริมาณแบนวิดท์สูงสุดที่ระบบใช้งานได้ (Maximum Bandwidth)

Cbw คือ ปริมาณการใช้งานแบนวิดท์ที่วัดจากระบบเครือข่าย (Current Bandwidth)

Tr คือ อัตราการถ่ายโอนข้อมูลเฉลี่ยของระบบเครือข่าย (Transfer Rate)

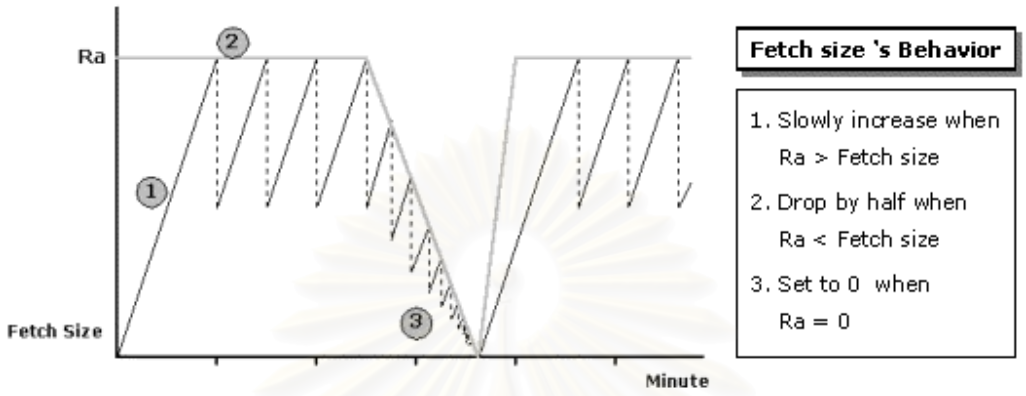
Ra จะเป็นตัวบ่งบอกว่ามีปริมาณแบนวิดท์ในระบบว่างมากน้อยเพียงใดในช่วงเวลาที่วัด ซึ่งเฟทช์โมดูลจะใช้ค่านี้ในการกำหนดขนาดของเฟทช์วินโดว์ แต่เฟทช์โมดูลจะไม่กำหนดให้ขนาดของเฟทช์วินโดว์มีค่าเท่ากับ Ra เลยเพราะ ในระบบเครือข่ายจะมีการใช้งานแบนวิดท์ตลอดเวลา ถ้าเฟทช์โมดูลใช้แบนวิดท์ที่ว่างอยู่ทั้งหมด จะทำให้ระบบมีแบนวิดท์ไม่เพียงพอที่จะให้บริการอื่นๆ ได้ ในงานวิทยานิพนธ์นี้จะทดลองใช้วิธีในการกำหนดขนาดของเฟทช์วินโดว์ 2 วิธี คือ

1) คอนเจสชัน คอนโทรล (Congestion Control)

วิธีนี้จะนำแนวคิดของ สโลวสตาร์ท (Slow Start) กับ คอนเจสชันอวอยแดนซ์ (Congestion Avoidance) ใน สไลด์ดิงวินโดว์ (Sliding Window) ของ ทีซีพีโพรโทคอล(TCP Protocol) [16] มาใช้ในการควบคุม

รูปที่ 3.3 แสดงให้เห็นถึงวิธีการควบคุมการขนาดของเฟทช์วินโดว์ ที่จุด 1: ขนาดของเฟทช์วินโดว์จะเริ่มที่ 0 เมื่อเริ่มต้นระบบและจะค่อยๆเพิ่มทีละ 10% ของค่าความแตกต่างระหว่าง Ra กับ ขนาดเดิมของเฟทช์วินโดว์ ที่จุด 2: เมื่อขนาดของเฟทช์วินโดว์มากกว่าหรือเท่ากับค่า Ra ที่วัดได้ ซึ่งหมายความว่าระบบดึงข้อมูลล่วงหน้ากำลังใช้แบนวิดท์ของเครือข่ายที่ว่างอยู่ทั้งหมด ขนาดของเฟทช์วินโดว์จะลดลงครึ่งหนึ่งของขนาดเดิมเพื่อที่จะคืนแบนวิดท์ให้เครือข่าย และที่จุด 3: เมื่อแบนวิดท์ในระบบถูกใช้เต็มที่จะส่งผลให้ Ra มีค่า

เป็น 0 ขนาดของเฟทช์วินโดว์ก็จะถูกกำหนดเป็น 0 ด้วย เพื่อหยุดการทำงานของระบบดึงข้อมูลล่วงหน้า และจะเริ่มทำงานต่อเมื่อมีแบนวิดท์ในเครือข่ายว่างอีกครั้งซึ่งค่า Ra จะมีค่ามากกว่า 0



รูปที่ 3.3 การเปลี่ยนแปลงขนาดของเฟทช์วินโดว์

ด้วยวิธีนี้จะทำให้ระบบดึงข้อมูลล่วงหน้าไม่ใช้แบนวิดท์ที่ว่างอยู่ทั้งหมดในครั้งเดียว แต่จะค่อยๆ เพิ่มปริมาณการใช้เพื่อไม่ให้ส่งผลกะทบต่อเครือข่าย และจะลดปริมาณการใช้อย่างรวดเร็ว เมื่อเครือข่ายต้องการแบนวิดท์

2) เลเวล คอนโทรล (Level Control)

วิธีนี้จะควบคุมโดยการตั้งอัตราส่วนที่แน่นอนระหว่างขนาดของเฟทช์วินโดว์ กับค่าของ Ra โดยในวิทยานิพนธ์นี้จะใช้ค่าเท่ากับ 0.9 ซึ่งหมายความว่า ขนาดของเฟทช์วินโดว์จะมีค่าเป็น 90% ของ ค่า Ra ที่หาได้จากการวัดเครือข่าย วิธีนี้จะทำให้การดึงข้อมูลล่วงหน้าไม่ใช้แบนวิดท์ที่มีอยู่ทั้งหมดทำให้มีแบนวิดท์เหลือสำหรับการใช้งานตามปกติอยู่ตลอดเวลา และจะสามารถใช้แบนวิดท์ของเครือข่ายที่ว่างอยู่ได้มากกว่าแบบคอนเจสชันคอนโทรล เนื่องจากมีการปรับขนาดของเฟทช์วินโดว์ตามขนาดของ Ra ทันทีโดยไม่ต้องเสียเวลาเหมือนแบบแรก แต่ในวิธีนี้จะเกิดปัญหาขึ้นเมื่อระบบเครือข่ายมีแบนวิดท์ที่ว่างเหลืออยู่น้อย การควบคุมแบนวิดท์ของวิธีนี้จะทำให้ปริมาณแบนวิดท์ที่ว่างอยู่ของระบบน้อยลงไปอีก ทำให้มีโอกาสที่จะเกิดผลกระทบกับระบบได้มากกว่าวิธีแรก

3.3 วิธีที่ใช้ในการทดสอบ

ในวิทยาพนธ์นี้จะทดสอบประสิทธิภาพของระบบที่ได้ออกแบบขึ้นโดยการใช้โปรแกรมจำลองการทำงานของระบบเครือข่ายที่ได้พัฒนาขึ้นมาเอง โดยจะจำลองการทำงานของระบบภายใต้สภาพแวดล้อมที่แตกต่างกันเปรียบเทียบกับระบบเดิม เพื่อทดสอบประสิทธิภาพและผลที่ได้รับจากการใช้ระบบที่ได้พัฒนาขึ้น โปรแกรมจำลองที่วิทยานิพนธ์นี้สร้างขึ้นมาใช้ภาษาเพิร์ล (Perl) ในการพัฒนา โดยตัวโปรแกรมจะทำงาน

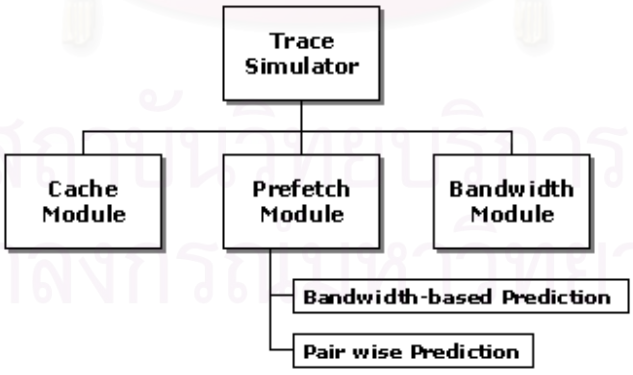
แบบ การจำลองแบบติดตาม (Trace-driven Simulation) ซึ่งเป็นการจำลองที่จะต้องใช้ข้อมูลการใช้เว็บ (log) โดยจะจำลองการทำงานไปตามเหตุการณ์ที่เกิดขึ้นในข้อมูลการใช้เว็บ

โปรแกรมจำลองจะทำงานไปตามบันทึกการใช้งานภายใต้สภาพแวดล้อมที่ได้กำหนดไว้ในการทดสอบแต่ละแบบ โดยจะบันทึกสภาพแวดล้อมของระบบผลที่เกิดขึ้นในการจำลอง และทำการวัดประสิทธิภาพของการดึงข้อมูลล่วงหน้าในแต่ละการทดสอบ

3.3.1 หลักการจำลอง

โปรแกรมจำลองระบบที่สร้างขึ้นมาจะประกอบด้วยส่วนหลักๆ 4 ส่วน ซึ่งเราจะเห็นได้จากรูปที่ 3.4 คือ

- โปรแกรมจำลอง (Trace Simulator) ส่วนนี้จะควบคุมการจำลองทั้งหมดและบันทึกสภาพแวดล้อมของระบบที่สร้างขึ้นระหว่างการจำลองรวมทั้งสรุปผลที่ได้เมื่อสิ้นสุดการทำงาน
- แคชโมดูล (Cache Module) ทำหน้าที่เป็นพร็อกซีแคชของระบบจำลอง โดยจัดการระบบเอกสารที่ถูกเก็บอยู่ภายในแคช
- แบนวิดท์โมดูล (Bandwidth Module) ทำหน้าที่ควบคุมแบนวิดท์ของระบบจำลอง และจำลองการทำงานของคำร้องขอที่สร้างขึ้นมาในระบบ
- พรีเฟทช์โมดูล (Prefetch Module) จะจำลองการทำงานของระบบพรีเฟทช์ ในส่วนนี้จะแบ่งออกเป็น 2 โมดูล คือ โมดูลที่ใช้ในการจำลองระบบดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ กับโมดูลที่ใช้จำลองระบบดึงข้อมูลล่วงหน้าแบบจับคู่ เพื่อใช้ในการเปรียบเทียบประสิทธิภาพในการทดสอบแต่ละแบบ

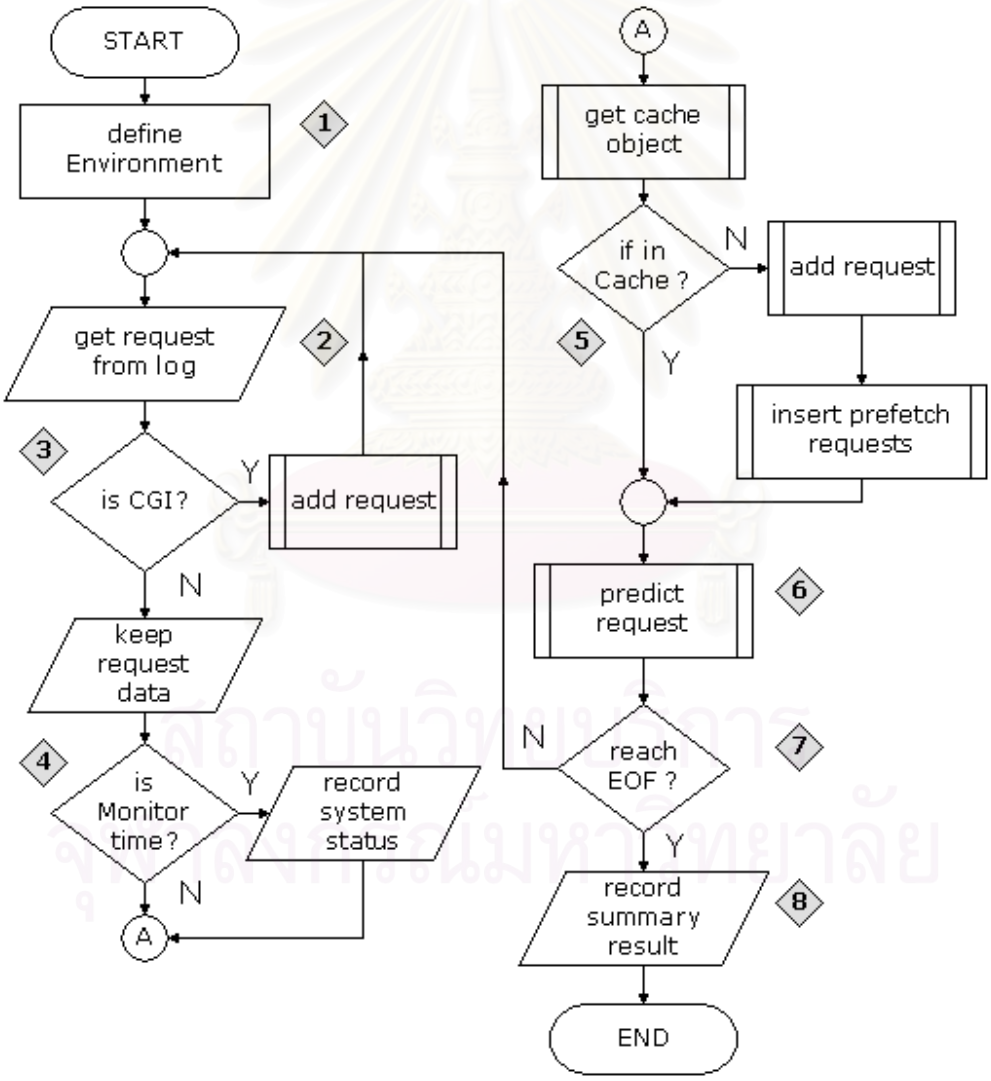


รูปที่ 3.4 โครงสร้างของโปรแกรมจำลอง

3.3.2 ขั้นตอนการทำงานของโปรแกรมจำลอง

รูปที่ 3.5 แสดงผังงาน (Flow Chart) ของโปรแกรมจำลองซึ่งอธิบายได้ตามลำดับดังนี้

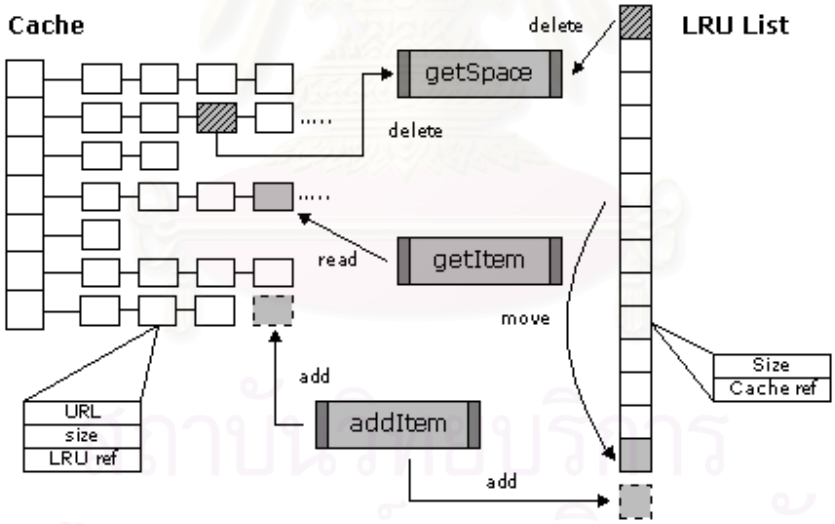
1) กำหนดสภาพแวดล้อมของระบบจำลอง โดยพารามิเตอร์ต่างๆ เช่น ขนาดของแคช ความจุแบนวิดท์ของระบบ วิธีการพรีเฟทช์ ซึ่งจะขึ้นอยู่กับผู้ทดสอบว่าต้องการให้ระบบทำงานอยู่ในสภาพแวดล้อมแบบใด
2) ดึงข้อมูลจากข้อมูลการใช้เว็บ (log) เพื่อใช้ในการจำลองคำร้องขอที่ผู้ใช้งานร้องขอมาที่ระบบ
3) ถ้าคำร้องขอเป็นเอกสารแบบซีจีไอ (CGI) โปรแกรมจะเพิ่มคำร้องขอไปในระบบแบนวิดท์ และกลับไปดึงข้อมูลใหม่อีกครั้ง เพราะเอกสารที่เป็นซีจีไอจะไม่ถูกเก็บอยู่ในแคช และไม่เหมาะที่นำไปดึงข้อมูลล่วงหน้า แต่ถ้าเป็นเอกสารอื่น เช่น เอกสารเอชทีเอ็มแอล เอกสารที่เป็นรูปภาพ ก็จะเก็บข้อมูลของคำร้องขอและทำงานต่อไป



รูปที่ 3.5 การทำงานของโปรแกรมจำลอง

4) โปรแกรมจำลองจะบันทึกสภาพของระบบที่ได้จำลองขึ้นอยู่ตลอดเวลาเพื่อศึกษาผลกระทบที่มีต่อระบบในแต่ละช่วงของวัน เมื่อถึงช่วงเวลาที่ตั้งไว้โปรแกรมก็จะทำการเก็บสภาพของระบบที่จำลองไว้ ซึ่งในวิทยานิพนธ์นี้จะตั้งให้โปรแกรมเก็บข้อมูลทุกๆ 1 นาที
5) โปรแกรมจะทำการค้นหาเอกสารภายในแคช ถ้าไม่พบก็จะทำเพิ่มคำร้องขอไปในระบบแบนวิดท์เพื่อจำลองการร้องขอไปยังเว็บเซิร์ฟเวอร์ ซึ่งทุกๆครั้งที่มีการเพิ่มคำร้องขอไปในระบบแบนวิดท์โปรแกรมก็จะแทรกคำร้องขอที่ระบบดึงข้อมูลล่วงหน้าสร้างไว้ลงไปด้วย เนื่องจากระบบดึงข้อมูลล่วงหน้าจะทำตัวเสมือนผู้ใช้งานคนหนึ่งในเครือข่าย
6) โปรแกรมจำลองจะส่งคำร้องขอของผู้ใช้งานไปให้ส่วนดึงข้อมูลล่วงหน้าทำนายซึ่งจะสร้างคำร้องขอที่ทำนายได้ออกสู่ระบบ
7) โปรแกรมจะวนอ่านข้อมูลและทำการจำลองต่อไปจนกว่าข้อมูลการใช้งานจะหมด
8) เมื่อจำลองเสร็จแล้ว โปรแกรมจะทำการบันทึกผลสรุปที่ได้จากการจำลอง

3.3.3 รายละเอียดของแคชโมดูล



รูปที่ 3.6 โครงสร้างการทำงานของแคชโมดูล

แคชโมดูลจะจำลองการทำงานของพร็อกซีในระบบ การทดสอบในวิทยานิพนธ์นี้จะให้พร็อกซีใช้วิธีแทนที่เอกสาร (Replacement Policy) แบบ แอลอาร์ยู (LRU: least recently use) ซึ่งเป็นวิธีที่นิยมใช้ที่สุด โดยเมื่อพร็อกซีมีพื้นที่ไม่เพียงพอที่จะเก็บเอกสารที่เข้ามาใหม่ พร็อกซีก็จะทำการลบเอกสารที่อยู่ในแคชก่อนหน้านี้ออกไป วิธีนี้จะเลือกเอกสารในแคชที่ไม่ได้ถูกเรียกงานนานที่สุดออกไป จนกว่าแคชมีพื้นที่พอที่จะเก็บ

เอกสารใหม่ รูปที่ 3.6 แสดงถึงโครงสร้างและการทำงานของแคชโมดูล แคชโมดูลจะประกอบด้วย 2 ส่วนหลักๆ คือ ออบเจ็กต์ Cache ซึ่งเป็นกลุ่มของรายการโยง (Link List) ที่ถูกแยกด้วยแฮชฟังก์ชัน (hash function) โดย ออบเจ็กต์ Cache จะใช้เก็บข้อมูลของเอกสารที่อยู่ในแคช และ รายการ LRU ซึ่งเป็นรายการโยงที่ใช้ในการแทนที่เอกสารแบบแอลอาร์ยูในแคช

เมื่อเอกสารถูกเพิ่มเข้ามาในแคชด้วยฟังก์ชัน addItem เอกสารจะถูกกำหนดโดยแฮชฟังก์ชันว่าจะอยู่ในรายการโยงไหน แล้วจะถูกนำไปต่อท้ายของรายการโยงนั้น และฟังก์ชัน addItem จะเพิ่มข้อมูลของเอกสารไปยังท้ายของรายการ LRU เพื่อบอกถึงลำดับการถูกเรียกใช้งานของเอกสาร

เมื่อมีการเรียกใช้งานเอกสารในแคช ฟังก์ชัน getItem จะหาเอกสารในออบเจ็กต์ Cache ถ้าไม่พบก็จะแจ้งว่าไม่พบเอกสาร แต่ถ้าพบฟังก์ชัน getItem จะคืนข้อมูลของเอกสารให้ และทำการเปลี่ยนตำแหน่งของเอกสารในรายการ LRU ไปไว้ที่ท้ายรายการ

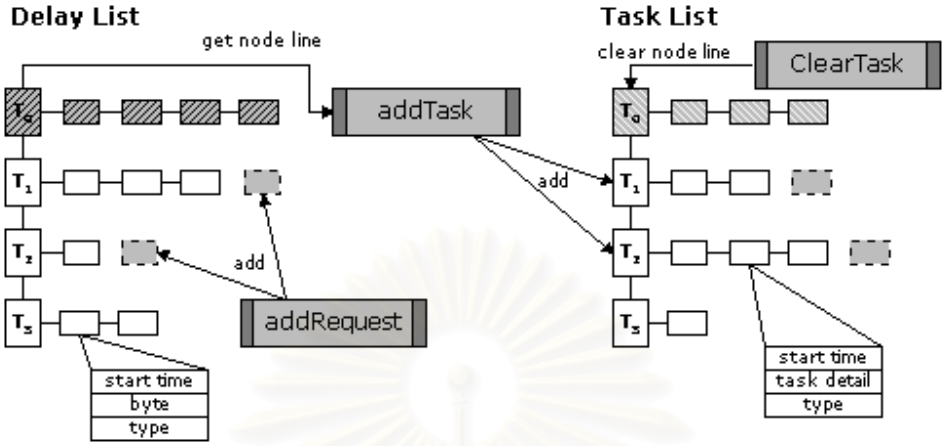
ฟังก์ชัน getspace จะถูกเรียกใช้เมื่อแคชมีพื้นที่ไม่เพียงพอที่จะเก็บเอกสารใหม่ ฟังก์ชัน getspace จะลบเอกสารที่อยู่ที่ต้นรายการของรายการ LRU ซึ่งเป็นเอกสารที่ไม่ถูกเรียกใช้นานที่สุด โดยฟังก์ชัน getspace จะลบข้อมูลของเอกสารทั้งใน รายการ LRU และ ออบเจ็กต์ Cache เพื่อเพิ่มพื้นที่ว่างให้กับแคชจนกว่าจะมีเนื้อที่พอให้เอกสารใหม่

3.3.4 รายละเอียดของแบนวิดท์โมดูล

แบนวิดท์โมดูลจะมีหน้าที่รับคำร้องขอทั้งหมดที่เกิดขึ้นในระบบและจำลองสภาพของแบนวิดท์ที่เกิดขึ้นเนื่องจากคำร้องขอเหล่านั้น โดยรายละเอียดของโมดูลจะถูกแสดงในรูปที่ 3.7 Delay List จะใช้ในการจำลองเวลาที่เกิดขึ้นจากการสร้างเชื่อมต่อ (Propagation Time) และเวลาที่เกิดขึ้นในคิวของพร็อกซีเซิร์ฟเวอร์และเครือข่าย ส่วน Task List จะใช้ในการจำลองช่วงเวลาส่งข้อมูล (Transmission Time) และการใช้งานแบนวิดท์ของระบบ

เมื่อมีการเพิ่มคำร้องขอเข้ามาในระบบ ฟังก์ชัน addRequest จะคำนวณเวลาที่เกิดขึ้นจากการเชื่อมต่อของคำร้องขอที่เข้ามาและจะเพิ่มคำร้องขอเข้าไปในรายการโยงของ Delay List ที่จะหมดเวลาในช่วงนั้น ส่วนฟังก์ชัน addTask จะถูกใช้งานเมื่อมีรายการโยงใน Delay List หมดเวลา โดยจะดึงคำร้องขอที่อยู่ในรายการโยงนั้นไปเพิ่มใน Task List เพื่อเข้าสู่ช่วงการถ่ายโอนข้อมูล (Transfer Process) ของคำร้องขอ แต่ถ้าในขณะนั้นระบบไม่สามารถรับคำร้องขอเพิ่มได้ คำร้องขอจะต้องรออยู่ใน Delay List ต่อไป

เมื่อถูกเพิ่มเข้ามาใน Task List คำร้องขอจะใช้ค่าอัตราถ่ายโอนของระบบในขณะนั้นในการคำนวณเวลาที่ใช้ในการถ่ายโอนข้อมูลและปริมาณแบนวิดท์ที่เพิ่มขึ้นในระบบ และจะถูกเพิ่มเข้าไปในรายการโยงที่มีเวลาสิ้นสุดงานตรงกัน แต่เมื่อมีการเปลี่ยนแปลงอัตราถ่ายโอนของระบบเกิดขึ้น แบนวิดท์โมดูลจะคำนวณเวลาสิ้นสุดงานของทุกคำร้องขอใน Task List ใหม่และจะปรับตำแหน่งของคำร้องขอให้ไปอยู่ในรายการโยงที่ตรงกันอีกครั้ง



รูปที่ 3.7 โครงสร้างและการทำงานของแบนวิดท์โมดูล

เมื่อรายการโยงใน Task List หมดเวลาลง ฟังก์ชัน ClearTask ก็จะลบคำร้องขอที่อยู่ในรายการนั้นและคืนแบนวิดท์ที่แต่ละคำร้องขอใช้อยู่ให้กับระบบ โดยการนำอัตราการถ่ายโอนที่คำร้องขอใช้อยู่ไปบวกให้กับตัวแปร CurrentBandwidth ซึ่งเป็นตัวแปรที่เก็บค่าปริมาณการใช้งานแบนวิดท์ของทั้งระบบ

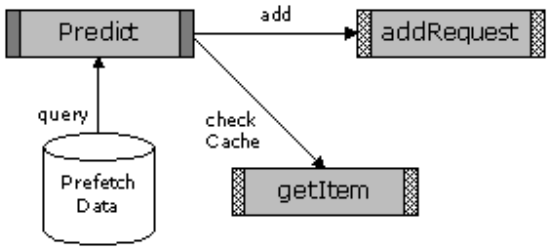
3.3.5 รายละเอียดของพรีเฟทช์โมดูล

โปรแกรมจำลองจะใช้พรีเฟทช์โมดูล 2 แบบเพื่อใช้จำลองการดึงข้อมูลล่วงหน้าแบบเดิมและการดึงข้อมูลล่วงหน้าแบบที่เปลี่ยนแปลงตามแบนวิดท์ ซึ่งจะมีรายละเอียดดังนี้คือ

3.3.5.1 พรีเฟทช์โมดูลของการดึงข้อมูลล่วงหน้าแบบจับคู่

โมดูลนี้จะใช้ในการจำลองการดึงข้อมูลล่วงหน้าแบบเดิมที่ใช้วิธีการทำนายแบบจับคู่ โดยเมื่อได้ยูอาร์แอลของคำร้องขอที่ผู้ใช้งานส่งมา ฟังก์ชัน Predict ก็จะค้นหายูอาร์แอลของเอกสารถัดไปจากฐานข้อมูลและจะนำมาเปรียบเทียบกับขอบเขตการดึงข้อมูลล่วงหน้า ถ้ายูอาร์แอลของเอกสารใดที่มีความน่าจะเป็นมากกว่าขอบเขต ฟังก์ชัน Predict ก็จะค้นหาเอกสารที่ต้องการในแคชก่อน แต่ถ้าไม่พบก็จะทำการร้องขอเอกสารไปยังระบบ ดังจะเห็นได้ในรูปที่ 3.8

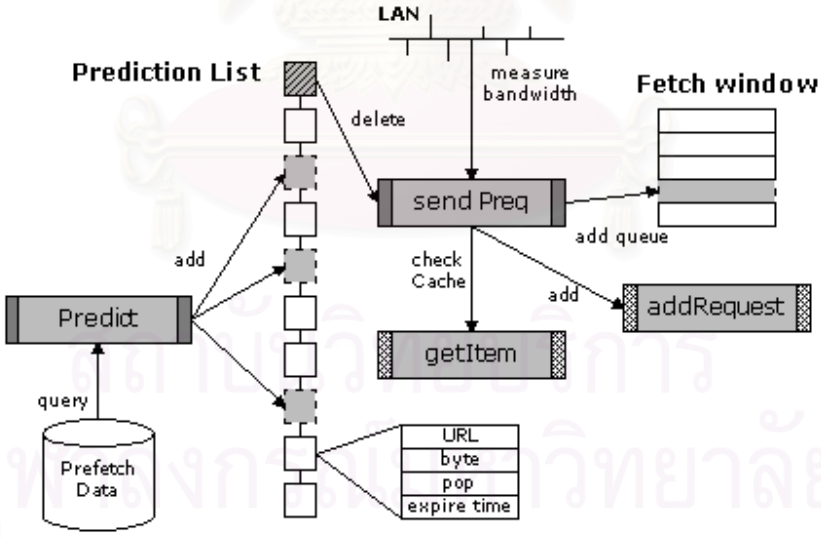
จากรูปที่ 3.9 เราจะเห็นโครงสร้างของพรีเฟทช์โมดูลที่จำลองการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ โดยจะมีส่วนที่เพิ่มเข้ามาคือ Prediction List ที่ใช้เก็บยูอาร์แอลของเอกสารที่ทำนายได้ และเฟทช์วินโดว์ (Fetch window) ที่ใช้จำกัดปริมาณการดึงข้อมูลล่วงหน้า หลังจากค้นข้อมูลจากฐานข้อมูลแล้ว ฟังก์ชัน Predict ก็จะนำยูอาร์แอลที่ได้มาใส่ไว้ใน Prediction List โดยยูอาร์แอลใน Prediction List จะเรียงลำดับตามค่าความน่าจะเป็นของเอกสารจากมากไปน้อย



รูปที่ 3.8 โครงสร้างและการทำงานของพรีเฟทช์โมดูลแบบเดิม

3.3.5.2 พรีเฟทช์โมดูลของการดึงข้อมูลล่วงหน้าแบบที่เปลี่ยนแปลงตามแบนวิดท์

ฟังก์ชัน sendPreq จะดึงยูอาร์แอลที่มีความน่าจะเป็นมากที่สุดจากต้น Prediction List โดยจะเช็คก่อนว่ายูอาร์แอล์ที่ดึงมาหมดอายุหรือไม่ ถ้ายังไม่หมดอายุก็จะทำการค้นหาเอกสารในแคช ในกรณีที่ไม่พบในแคช ฟังก์ชัน sendPreq ก็จะร้องขอเอกสารของยูอาร์แอลนั้นไปยังระบบเพื่อทำการดึงข้อมูลล่วงหน้าโดย จำนวนของคำร้องขอที่ฟังก์ชัน sendPreq จะร้องขอได้พร้อมๆ กันจะมีได้ไม่เกินขนาดของเฟทช์วินโดว์ ซึ่งจะถูกกำหนดจากปริมาณแบนวิดท์ในเครือข่ายอีกที การกำหนดขนาดของเฟทช์วินโดว์จะขึ้นอยู่กับวิธีที่ใช้ในการควบคุมที่ผู้ทำการจำลองได้ตั้งค่าไว้ว่าจะใช้การควบคุมแบบ เลเวล คอนโทรล หรือ คอนเจสชัน คอนโทรล



รูปที่ 3.9 โครงสร้างและการทำงานของพรีเฟทช์โมดูลแบบเปลี่ยนแปลงตามแบนวิดท์

3.4 สมมุติฐานในการทดสอบ

ในการทดสอบของวิทยานิพนธ์นี้พารามิเตอร์ (Parameter) ของสภาพแวดล้อมในระบบที่สร้างขึ้นจะถูกตั้งให้เป็นค่าคงที่จำนวนหนึ่ง เพื่อความสะดวกในการจำลองและความชัดเจนในการวัดประสิทธิภาพ ซึ่งมีดังนี้คือ

ค่าที่ได้จากการเก็บสถิติจากข้อมูลการใช้งานเว็บ

- ช่วงเวลาเข้าถึง (Access Time) ของเอกสารที่พบในแคช จะมีค่าเป็น 0.029 วินาที
- ปริมาณคำร้องขอที่ระบบเครือข่ายสามารถรองรับได้ต่อวินาที มีค่าเป็น 270 คำร้องขอต่อวินาที

พารามิเตอร์เหล่านี้จะได้การเก็บสถิติจากข้อมูลการเว็บที่ใช้ในการทดสอบระบบในหัวข้อ 3.5 โดยทำการสุ่มตัวอย่างขึ้นมา 3 วันแล้วหาค่าเฉลี่ยของพารามิเตอร์ที่ต้องการวัด

ค่าที่อ้างอิงจากงานวิจัยอื่น

- เวลาที่ใช้สร้างการเชื่อมต่อ (propagation time) ของแต่ละคำร้องขอ จะมีค่าเป็น 1.13 วินาที เนื่องจากสภาพของข้อมูลที่นำมาใช้ในการทดสอบมีการกระจายของช่วงเวลาเข้าถึงมากจึงไม่สามารถหาเวลาที่ใช้ในการทดสอบได้ ในงานวิทยานิพนธ์นี้จึงอ้างอิงเวลาที่ใช้สร้างการเชื่อมต่อที่ [7] ได้วิเคราะห์ไว้

ค่าที่กำหนดขึ้นเพื่อใช้ในการวัดประสิทธิภาพ

- อัตราถ่ายโอนต่ำสุด (Minimum Transfer Rate) มีค่าเป็น 500 ไบต์ต่อวินาที
- อัตราถ่ายโอนสูงสุด (Maximum Transfer Rate) มีค่าเป็น 4000 ไบต์ต่อวินาที

ในสภาพความเป็นจริงอัตราถ่ายโอนจะขึ้นกับสภาพเครือข่ายของผู้ใช้งาน กับสภาพเครือข่ายของเซิร์ฟเวอร์ที่ผู้ใช้งานร้องขอเอกสารไป ซึ่งจะมีค่าไม่แน่นอนทำให้ยากต่อการวัดประสิทธิภาพ ในวิทยานิพนธ์นี้จึงกำหนดให้อัตราถ่ายโอนขึ้นอยู่กับสภาพเครือข่ายของผู้ใช้งานเท่านั้น และกำหนดของเขตของการถ่ายโอนเพื่อให้ง่ายต่อการวัดประสิทธิภาพ

นอกจากนี้ในระบบจำลองยังกำหนดให้เอกสารที่ถูกเก็บอยู่ในแคชไม่มีการหมดอายุ เพื่อลดความซับซ้อนในการจำลอง และขนาดของคิวของพร็อกซีเซิร์ฟเวอร์และอุปกรณ์จัดเส้นทางของเครือข่ายมีขนาดไม่จำกัด ซึ่งจะทำให้ไม่มีกรณีที่คำร้องขอถูกปฏิเสธจากระบบ และจะส่งผลให้เวลาเฉลี่ยที่คำร้องขอใช้ในการรอคอยอยู่ในคิวมากขึ้นถ้าเกิดกรณีที่มีคำร้องขอมากเกินกว่าที่ระบบจริงรองรับได้

3.5 ข้อมูลที่ใช้ในการทดสอบ

ข้อมูลที่ใช้ในการทดสอบ คือ ข้อมูลการใช้เว็บจาก พร็อกซีเซิร์ฟเวอร์ของสำนักเทคโนโลยีสารสนเทศ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งรองรับการใช้งานของอาจารย์และนักศึกษาภายในมหาลัยทั้งหมด โดยข้อมูลที่ได้มาจะกรอง (Filter) เอาคำร้องขอที่ผิดพลาด และคำร้องขอที่มาจากพร็อกซีภายในองค์กรตัวอื่นๆ ออกเพื่อให้ทำการจำลองได้ถูกต้องซึ่งข้อมูลที่ได้จากกรองจะมีลักษณะดังตาราง 3.1

ตาราง 3.1 ลักษณะของข้อมูลการใช้เว็บที่ใช้ในการทดสอบ

จำนวนคำร้องขอ	3,376,878 ครั้ง
จำนวนผู้ใช้งาน	4,961 คน
จำนวนเอกสารที่ไม่ซ้ำกัน	980,924 เอกสาร
ขนาดรวมของเอกสารที่ไม่ซ้ำกัน	10,326,965,814 ไบต์
ขนาดของเอกสารโดยเฉลี่ย	10,527.70 ไบต์

และข้อมูลที่นำมาใช้ในการสร้างสถิติในการทำนายแบบจับคู่คือข้อมูลการใช้เว็บ จำนวน 15 วันจากที่เดียวกันซึ่งเก็บข้อมูล 3,470,782 ลิงค์ (Link) ของ 396,153 เอกสาร

3.6 สภาพแวดล้อมของเครื่องที่ใช้ในการจำลอง

ในวิทยานิพนธ์นี้ใช้คอมพิวเตอร์ 2 เครื่องในการจำลองระบบรูปแบบต่างๆ ที่ใช้ในการทดสอบ ด้วยโปรแกรมจำลองที่ได้พัฒนาขึ้น ซึ่งแต่ละเครื่องมีรายละเอียดดังนี้

1. เครื่องแรก
   - คอมพิวเตอร์แบบพีซี Pentium celeron 300 เมกกะเฮิรตซ์
   - หน่วยความจำ 416 เมกกะไบต์
   - ฮาร์ดดิสก์ 3 กิกะไบต์
   - ใช้ระบบปฏิบัติการ Linux RedHat 6.2
2. เครื่องที่สอง
   - คอมพิวเตอร์แบบพีซี AMD duron 600 เมกกะเฮิรตซ์
   - หน่วยความจำ 512 เมกกะไบต์
   - ฮาร์ดดิสก์ 10 กิกะไบต์
   - ใช้ระบบปฏิบัติการ Linux RedHat 7.1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

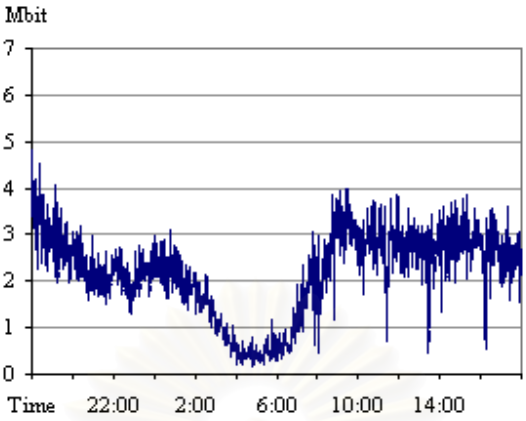
ผลการทดสอบ

ในบทนี้จะกล่าวถึงผลการทดสอบประสิทธิภาพของการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์โดยใช้วิธีการทดสอบตามที่ได้กล่าวไปแล้วในบทที่ 3 แต่ก่อนหน้าที่จะทำการทดสอบระบบที่ใช้การดึงข้อมูลล่วงหน้าที่ได้ออกแบบขึ้น การทดสอบในบทนี้ได้จำลองการทำงานของระบบการดึงข้อมูลล่วงหน้าที่ใช้ขอบเขตการดึงข้อมูลเท่ากับ 0.01 เพื่อศึกษาผลที่ได้รับและผลกระทบที่เกิดขึ้นกับระบบเครือข่ายเมื่อมีการนำการดึงข้อมูลล่วงหน้าที่มีขอบเขตต่ำที่ทำให้เกิดคำร้องขอจำนวนมากมาใช้ในระบบ หลังจากนั้นวิทยานิพนธ์นี้จะจำลองระบบที่ใช้การดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ และนำผลที่ได้มาเปรียบเทียบระบบที่ใช้การดึงข้อมูลแบบเดิม ในสภาวะที่ระบบมีการใช้งานแบนวิทด์ที่แตกต่างกัน เพื่อศึกษาประสิทธิภาพและผลกระทบที่มีต่อระบบ และเปรียบเทียบการทำงานของวิธีควบคุมแบนวิทด์ทั้ง 2 วิธี

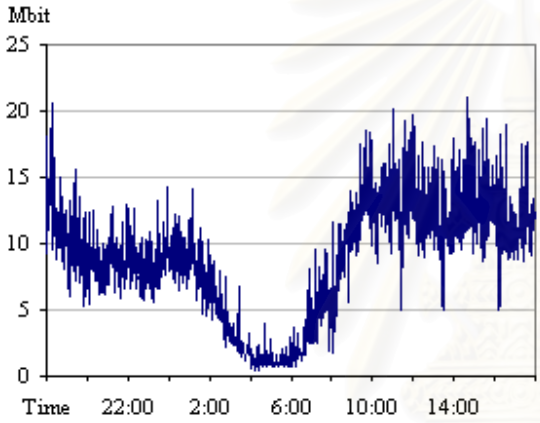
4.1 ผลกระทบที่เกิดขึ้นกับระบบเมื่อใช้การดึงข้อมูลล่วงหน้าแบบจับคู่

จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่างานวิจัยส่วนใหญ่จะทดสอบระบบโดยจำลองระบบที่มีแบนวิดท์ที่ไม่จำกัด โดยจะสนใจว่าการดึงข้อมูลล่วงหน้าที่พัฒนาขึ้นจะเพิ่มฮิตเรทได้มากขึ้นแค่ไหนและจะเพิ่มปริมาณการใช้งานแบนวิดท์เท่าไหร่ แต่ในความเป็นจริงในแต่ละเครือข่ายจะสามารถใช้แบนวิดท์ได้ในปริมาณที่จำกัดซึ่งปริมาณการใช้งานแบนวิดท์ที่มากเกินกว่าที่เครือข่ายจะรองรับได้จะก่อให้เกิดผลกระทบกับระบบ ในหัวข้อนี้จึงได้ทำการศึกษาผลกระทบที่เกิดขึ้นเมื่อมีปริมาณคำร้องขอจำนวนมากที่เกิดจากการดึงข้อมูลล่วงหน้าเข้าสู่ระบบ โดยจะทดสอบระบบการดึงข้อมูลล่วงหน้าแบบจับคู่โดยใช้ขอบเขตการดึงข้อมูลล่วงหน้าเท่ากับ 0.01 ซึ่งเป็นการขอบเขตการดึงข้อมูลล่วงหน้าที่ต่ำ ทำให้ระบบสามารถเพิ่มฮิตเรตได้มาก แต่ในขณะเดียวกันก็ใช้แบนวิดท์จำนวนมากด้วยเช่นกัน ในการทดสอบจะจำลองให้ระบบอยู่ในสภาวะที่มีแบนวิดท์ 6 เมกกะบิต และ สภาวะที่ไม่จำกัดแบนวิดท์ เพื่อเปรียบเทียบผลที่เกิดขึ้นจากการดึงข้อมูลในระบบที่ทำการจำลองแบบไม่มีข้อจำกัด กับระบบที่จำลองตามสภาพความเป็นจริง

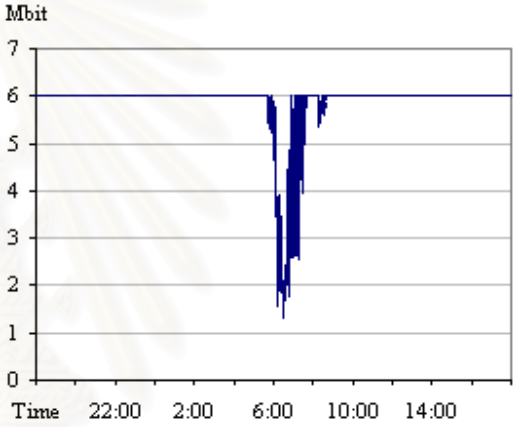
ในการวัดประสิทธิภาพของการดึงข้อมูลล่วงหน้า พารามิเตอร์สำคัญที่ใช้ในการวัดผลกระทบที่เกิดขึ้นกับระบบคือ อัตราถ่ายโอน (Transfer rate) กับเวลาดึงเอกสารที่ไม่พบในแคช (Fetch Time) ซึ่งจะเป็นพารามิเตอร์ที่แสดงถึงปริมาณทรัพยากรที่ผู้ใช้งานได้รับจากระบบเครือข่าย ถ้าอัตราถ่ายโอนลดลง หรือผู้ใช้งานใช้เวลาในการดึงเอกสารจากอินเตอร์เน็ตนานขึ้น แสดงว่าระบบดึงข้อมูลล่วงหน้าทำให้ประสิทธิภาพในการให้บริการของระบบเครือข่ายลดลง สำหรับเวลาเข้าถึงเอกสาร (access time) เป็นเวลาเฉลี่ยของทุกคำร้องขอที่เกิดขึ้นในระบบเครือข่ายซึ่งรวมทั้ง คำร้องขอที่พบในแคช และคำร้องขอที่ไม่พบในแคชซึ่งต้องดึงข้อมูลจากอินเตอร์เน็ต ดังนั้นเวลาเข้าถึงเอกสารจะเป็นตัววัดผลของการดึงข้อมูลล่วงหน้าที่ผู้ใช้งานได้รับ



ก) ระบบที่ไม่ได้ใช้การดึงข้อมูลล่วงหน้า



ข) ระบบที่ดึงข้อมูลล่วงหน้าโดยไม่จำกัดแบนวิดท์



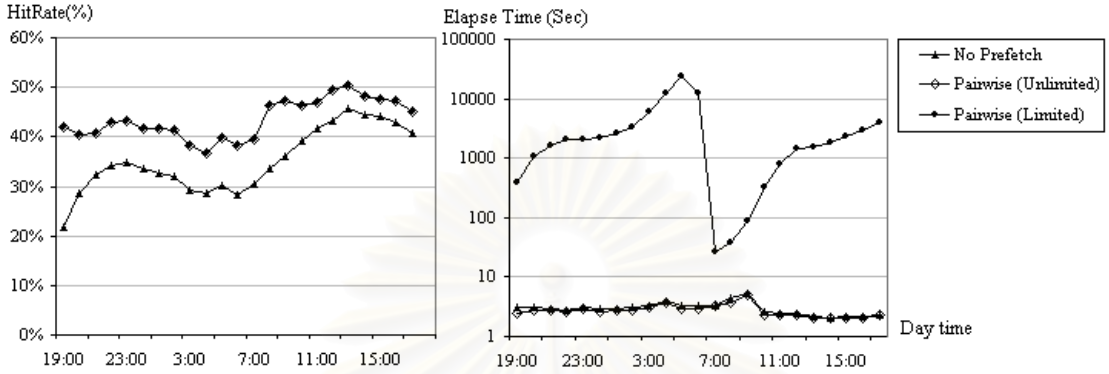
ค) ระบบที่ดึงข้อมูลล่วงหน้าโดยจำกัดแบนวิดท์

รูปที่ 4.1 ปริมาณการใช้แบนวิดท์ของระบบก่อนและ หลังการดึงข้อมูลล่วงหน้า

ในรูปที่ 4.1ข เราจะเห็นได้ว่าระบบที่นำการดึงข้อมูลล่วงหน้ามาใช้มีปริมาณการใช้งานแบนวิดท์เพิ่มขึ้นมากกว่าระบบที่ไม่ได้ใช้การดึงข้อมูล (รูปที่ 4.1ก) มาก และเมื่อทำการจำลองการดึงข้อมูลล่วงหน้าในสภาพที่มีแบนวิดท์จำกัด (รูปที่ 4.1ค) จะเห็นว่าระบบมีการใช้งานแบนวิดท์เต็มที่อยู่ตลอดเวลา โดยเราจะเห็นผลกระทบที่เกิดขึ้นจากรูปที่ 4.2 และ 4.3

จากรูปที่ 4.2ก จะเห็นได้ว่าฮิตเรทของระบบที่ใช้ดึงข้อมูลล่วงหน้าแบบจับคู่จะมากกว่าระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า ซึ่งจากตารางที่ 4.1 จะเห็นได้ว่าฮิตเรทโดยรวมจะเพิ่มขึ้น 7.56% ซึ่งในการจำลองระบบที่ใช้การดึงข้อมูลล่วงหน้าในสภาวะที่ไม่จำกัดแบนวิดท์ ฮิตเรทที่เพิ่มขึ้นจะทำให้เวลาเข้าถึงเอกสารลดลงจาก 2.6 วินาทีเป็น 2.3 วินาที แต่ในรูปที่ 4.2ข จะเห็นความแตกต่างไม่ชัดเจนนักเนื่องจากมีการเปรียบเทียบการจำลองระบบที่ใช้การดึงข้อมูลล่วงหน้าที่จำกัดแบนวิดท์ที่มีเวลาเข้าถึงเอกสารมากถึง 2245.4 วินาที สาเหตุที่ระบบที่ใช้การดึงข้อมูลล่วงหน้าที่จำกัดแบนวิดท์มีเวลาเข้าถึงเอกสารเพิ่มขึ้นมาก ทั้งๆ ที่สามารถเพิ่ม

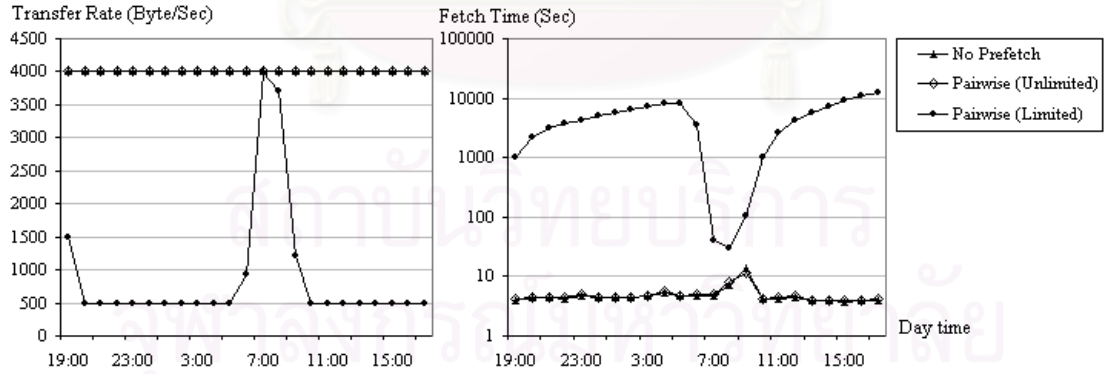
ฮิตเรทได้เท่ากับการดึงข้อมูลล่วงหน้าที่ไม่จำกัดแบนวิดท์ เกิดจากการเพิ่มปริมาณคำร้องขอจำนวนมากของการดึงข้อมูลล่วงหน้าส่งผลกระทบต่อระบบเครือข่ายที่มีแบนวิดท์จำกัด



ก) ฮิตเรทของการจำลองแต่ละแบบ ข) เวลาเข้าถึงเอกสารของการจำลองแต่ละแบบ

รูปที่ 4.2 ฮิตเรทและเวลาเข้าถึงเอกสารของการจำลอง แบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า (No Prefetch), แบบที่ใช้การดึงข้อมูลล่วงหน้าแบบจับคู่ในสภาวะที่ไม่จำกัดแบนวิดท์ (Pairwise (Unlimited)) และ แบบที่ใช้การดึงข้อมูลล่วงหน้าแบบจับคู่ในสภาวะที่จำกัดแบนวิดท์ (Pairwise (Limited))

รูปที่ 4.3ก แสดงถึงอัตราถ่ายโอนข้อมูลของการจำลองแต่ละแบบ เราจะเห็นว่าอัตราถ่ายโอนของการจำลองแบบดึงข้อมูลล่วงหน้าในสภาวะที่จำกัดแบนวิดท์จะอยู่ในระดับต่ำสุดเป็นส่วนใหญ่ เนื่องจากปริมาณคำร้องขอที่เกิดขึ้นมีมากกว่าที่ระบบเครือข่ายรองรับได้ ในตารางที่ 4.1 เราจะพบว่าเวลาที่ใช้ในการ



ก) อัตราถ่ายโอนของการจำลองแต่ละแบบ ข) เวลาที่ใช้ดึงเอกสารที่ไม่พบในแคชของการจำลองแต่ละแบบ

รูปที่ 4.3 อัตราถ่ายโอนและเวลาที่ใช้ดึงเอกสารที่ไม่พบในแคชของการจำลองแบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า (No Prefetch), แบบที่ใช้การดึงข้อมูลล่วงหน้าแบบจับคู่ในสภาวะที่ไม่จำกัดแบนวิดท์(Pairwise (Unlimited)) และ แบบที่ใช้การดึงข้อมูลล่วงหน้าแบบจับคู่ในสภาวะที่จำกัดแบนวิดท์ (Pairwise (Limited))

ดึงเอกสารที่ไม่พบในแคชของการจำลองแบบดึงข้อมูลล่วงหน้าที่จำกัดแบนวิดท์มีค่า 5461.485 วินาทีซึ่งมีสาเหตุมาจากอัตราการถ่ายโอนข้อมูลที่ต่ำ และปริมาณคำร้องขอที่มีมากจนต้องเสียเวลารออยู่ในคิวของพร็อกซีเซิร์ฟเวอร์ ซึ่งในความเป็นจริงการร้องขอข้อมูลไม่สามารถใช้เวลาได้มากเท่ากับค่าที่วัดได้จากการจำลอง เพราะจะเกิดกรณีที่บัฟเฟอร์ (Buffer) ของพร็อกซีเต็มและจะปฏิเสธคำร้องขอที่ไม่สามารถรองรับได้ไป และจะเกิดกรณีที่เว็บเซิร์ฟเวอร์ปฏิเสธคำร้องขอที่ส่งออกไปเนื่องจากใช้เวลาส่งข้อมูลนานเกินไป (Connection Timeout)

ตารางที่ 4.1 เปรียบเทียบการจำลองแบนวิดท์ในระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้ากับระบบที่ใช้การดึงข้อมูลล่วงหน้าที่มีขอบเขตการดึงข้อมูลต่ำ

	ฮิตเรท	เวลาดึงเอกสารที่ไม่พบในแคช	เวลาเข้าถึงเอกสาร	อัตราถ่ายโอนต่อคำร้องขอ
แบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า	37.65%	4.16 sec	2.60 sec	3396.11 byte/sec
แบบจับคู่ (0.01) ที่ไม่จำกัดแบนวิดท์	45.21%	4.30 sec	2.37 sec	4000 byte/sec
แบบจับคู่ (0.01) ที่จำกัดแบนวิดท์	45.21%	5461.49 sec	2245.43 sec	867.29 byte/sec

จากผลการทดสอบจะเห็นได้ว่าการดึงข้อมูลล่วงหน้าที่ใช้ขอบเขตต่ำไม่ได้มีข้อเสียที่ใช้แบนวิดท์จำนวนมากเท่านั้น แต่คำร้องขอและแบนวิดท์ที่การดึงข้อมูลล่วงหน้าสร้างขึ้นมาจะแย่งทรัพยากรของระบบจากผู้ใช้งานทำให้คุณภาพของบริการที่ผู้ใช้งานได้รับลดลง

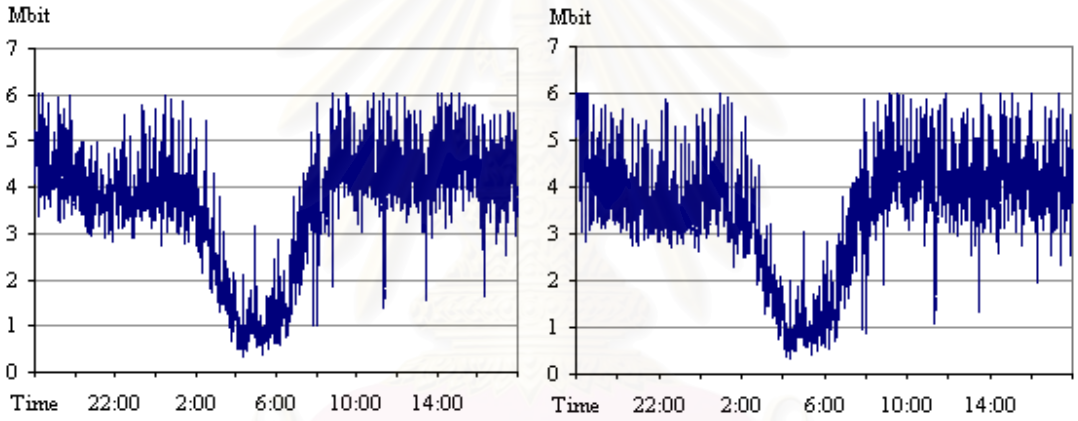
4.2 การทดสอบประสิทธิภาพของการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์

ในการทดสอบต่อไป วิทยานิพนธ์นี้จะทำการทดสอบประสิทธิภาพของการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ โดยจะทำการจำลองระบบที่มีแบนวิดท์ 6 เมกกะบิต ซึ่งอยู่ในสภาวะที่มีแบนวิดท์เพียงพอต่อการดึงข้อมูลล่วงหน้า และจะทำการจำลองระบบที่ 4 เมกกะบิต ซึ่งอยู่ในสภาวะที่มีแบนวิดท์เหลืออยู่น้อย เพื่อเปรียบเทียบสภาพการทำงานของวิธีควบคุมแบนวิดท์ทั้ง 2 วิธี หลังจากนั้นในหัวข้อ 4.2.3 เราจะทำการจำลองระบบในสภาพที่มีแบนวิดท์ 9 เมกกะบิต และในสภาพที่มีแบนวิดท์ไม่จำกัด เพื่อสังเกตประโยชน์ที่ได้รับและผลกระทบที่เกิดขึ้น เมื่อสภาพของแบนวิดท์เปลี่ยนไป โดยในการทดสอบจะจำลองการดึงข้อมูลล่วงหน้าแบบจับคู่ ที่ใช้ขอบเขต 0.01 ด้วย เพื่อเปรียบเทียบการดึงข้อมูลแบบใหม่ กับแบบเดิม โดย

ในการทดสอบ การดึงข้อมูลล่วงหน้าที่ใช้วิธีควบคุมทั้ง 2 แบบจะใช้วิธีทำนายแบบจับคู่ในการทำนายเอกสารเช่นกัน

4.2.1 การทดสอบประสิทธิภาพที่ 6 เมกกะบิต

ในหัวข้อนี้เราได้จำลองการทำงานของระบบที่ออกแบบขึ้นโดยใช้วิธีควบคุมแบนวิดท์ทั้ง 2 วิธี โดยจะจำลองในสภาวะที่ระบบมีแบนวิดท์ 6 เมกกะบิต ซึ่งเป็นสภาวะที่มีแบนวิดท์เพียงพอต่อการดึงข้อมูลล่วงหน้า และได้จำลองระบบที่ใช้การดึงข้อมูลล่วงหน้าแบบจับคู่ที่ใช้ของเขต 0.01 ในระบบที่มีแบนวิดท์ 6 เมกกะบิต ด้วยเพื่อเปรียบเทียบผลที่เกิดขึ้นกับระบบการดึงข้อมูลล่วงหน้าแบบเดิม เราจะเห็นปริมาณการใช้งานของแบนวิดท์ที่เกิดขึ้นของการดึงข้อมูลล่วงหน้าแบบควบคุมแบนวิดท์แต่ละแบบในรูปที่ 4.4 และข้อมูลที่ได้จากการทดสอบในตาราง 4.2



ก) ระบบที่ควบคุมแบบเลเวล คอนโทรล ข) ระบบที่ควบคุมแบบคอนเจสชัน คอนโทรล

รูปที่ 4.4 ปริมาณการใช้งานแบนวิดท์ของการจำลองแต่และแบบ

ในรูปที่ 4.4 จะเห็นได้ว่าระบบใหม่มีการใช้งานแบนวิดท์เพิ่มขึ้นมากกว่าแบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า (รูปที่ 4.1ก) มาก แต่ปริมาณการใช้งานแบนวิดท์จะไม่เพิ่มจนเต็มระดับการใช้งานแบนวิดท์ของระบบทำให้ทั้ง 2 แบบสร้างผลกระทบต่อระบบค่อนข้างน้อย ซึ่งจะเห็นได้ในตารางที่ 4.2 ว่าการดึงข้อมูลล่วงหน้าแบบใหม่ทำให้เวลาที่ใช้ในการดึงเอกสารที่ไม่พบในแคชเพิ่มขึ้นน้อยกว่า 1.5% ซึ่งต่างกับระบบจับคู่ที่ส่งผลกระทบจนระบบไม่สามารถทำงานได้ เราจะสังเกตได้ว่าในกราฟรูป 4.4ข เราจะเห็นว่าปริมาณการใช้งานแบนวิดท์จะเกิดการเพิ่มขึ้นและลดลงบ่อยครั้งเนื่องจากกลไกในการควบคุมแบนวิดท์ของการควบคุมแบบคอนเจสชัน คอนโทรล

ตารางที่ 4.2 **เปรียบเทียบการดึงข้อมูลล่วงหน้าแบบต่าง ๆในระบบจำลองที่มีแบนวิดท์ 6 เมกกะบิต**

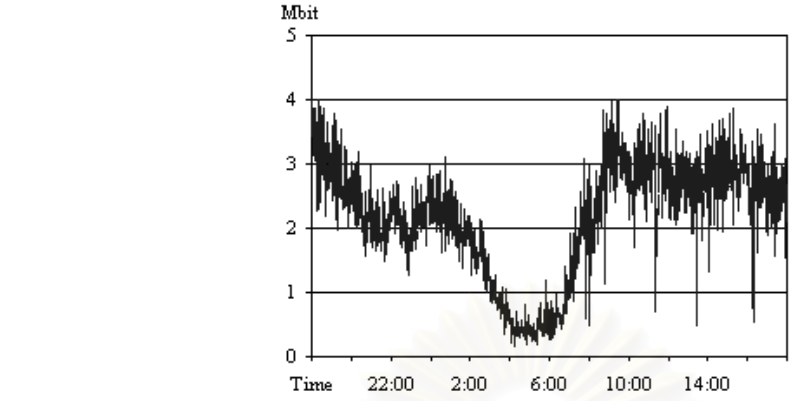
	แบบไม่ดึงข้อมูลล่วงหน้า	แบบ เลเวล คอนโทรล	แบบ คอนเจสชันคอนโทรล	แบบจับคู่ (0.01)
ฮิตเรต	37.65%	43.21%	42.42%	45.21%
เวลาดึงเอกสาร ที่ไม่พบในแคช	4.16 sec	4.22 sec	4.20 sec	5461.49 sec
อัตราถ่ายโอนต่อคำร้องขอ	3996.11 byte/sec	3985.83 byte/sec	3980.56 byte/sec	867.29 byte/sec
เวลาเข้าถึงเอกสาร	2.61 sec	2.41 sec	2.43 sec	2245.43 sec
ปริมาณแบนวิดท์ที่ใช้	18,105,813,099 byte	31,975,040,270 byte (176.6% ของระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า)	30,245,180,013 byte (167.1% ของระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า)	62,206,930,971 byte (343.6% ของระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า)

จากตารางที่ 4.2 จะเห็นได้ว่าแบบควบคุมแบนวิดท์ทั้ง 2 แบบจะให้ฮิตเรทน้อยกว่าแบบจับคู่ปกติภายใต้สภาวะเดียวกันประมาณ 3% แต่ส่งผลกระทบต่อระบบน้อยกว่ามากโดยเพิ่มเวลาดึงเอกสารที่ไม่พบในแคชเพียง 1.5% และทำให้อัตราถ่ายโอนข้อมูลลดลง 0.32% เท่านั้น

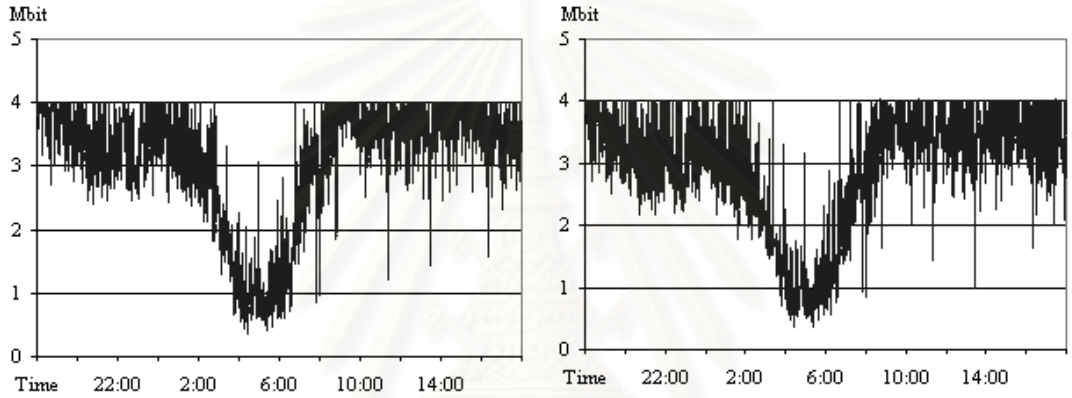
เมื่อเปรียบเทียบการทำงานของวิธีการควบคุมแบนวิดท์ทั้ง 2 วิธีในระบบที่มีแบนวิดท์ 6 เมกกะบิตซึ่งเป็นสภาพเครือข่ายที่มีแบนวิดเพียงพอต่อการดึงข้อมูลล่วงหน้าแล้วพบว่าสามารถทำงานได้ประสิทธิภาพใกล้เคียงกัน แต่แบบเลเวล คอนโทรลจะเพิ่มฮิตเรตได้มากกว่า ในขณะที่แบบคอนเจสชัน คอนโทรลจะส่งผลกระทบต่อระบบน้อยกว่า ซึ่งผลลัพธ์ที่ได้จะแสดงให้เห็นที่เวลาในการเข้าถึงเอกสาร ซึ่งพบว่าแบบเลเวล คอนโทรลสามารถลดเวลาเข้าถึงเอกสารได้มากกว่า

4.2.2 การทดสอบประสิทธิภาพที่ 4 เมกกะบิต

จากการทดสอบในหัวข้อที่ 4.2.1 พบว่าในระบบที่มีแบนวิดท์เพียงพอการควบคุมแบนวิดท์แบบเลเวล คอนโทรลจะให้ผลรวมที่ดีกว่าแบบคอนเจสชัน คอนโทรล โดยสามารลดเวลาเข้าถึงเอกสารได้มากกว่า 0.84% ในการทดสอบนี้เราจะจำลองระบบให้มีปริมาณแบนวิดท์ที่ระดับ 4 เมกกะบิต ซึ่งเป็นสภาพที่ระบบมีแบนวิดท์เหลืออยู่น้อยเพื่อศึกษาการทำงานของระบบควบคุมแบนวิดท์ทั้ง 2 วิธีเมื่อยู่ในสภาพแวดล้อมที่มีแบนวิดท์ไม่เพียงพอต่อการดึงข้อมูลล่วงหน้า



ก) ระบบที่ไม่ได้ดึงข้อมูลล่วงหน้า



ข) ระบบที่ควบคุมแบบเลเวล คอนโทรล ค) ระบบที่ควบคุมแบบคอนเจสชัน คอนโทรล

รูปที่ 4.5 กราฟแสดงการใช้งานแบนวิดท์ของระบบจำลองที่มีแบนวิดท์ 4 เมกกะบิต

รูปที่ 4.5 จะแสดงการใช้งานแบนวิดท์ของระบบที่จำลองขึ้น เราจะเห็นได้ว่าเมื่อเรากำหนดระดับแบนวิดท์อยู่ที่ 4 เมกกะบิตซึ่งมีปริมาณแบนวิดท์เหลืออยู่น้อย การดึงข้อมูลล่วงหน้าที่ใช้วิธีควบคุมทั้ง 2 แบบ จะเพิ่มการใช้งานแบนวิดท์ขึ้นจนถึงระดับแบนวิดท์ที่กำหนดไว้แต่จะไม่ใช้แบนวิดท์ที่เหลืออยู่ทั้งหมดทำให้เวลาดึงเอกสารที่ไม่พบในแคชเพิ่มขึ้นไม่มากนัก จากตาราง 4.3 เราจะเห็นได้ว่าเวลาดึงเอกสารของเพิ่มขึ้น 4.6% - 7.2% เท่านั้น

จากตารางที่ 4.3 เราพบว่าแบบเลเวล คอนโทรลจะเพิ่มฮิตเรทได้ 3.6% แต่ทำให้เวลาดึงเอกสารที่ไม่พบในแคชเพิ่มขึ้น 7.2% ทำให้เวลาเข้าถึงเอกสารเพิ่มขึ้นมากกว่าระบบที่ไม่ได้ใช้การดึงข้อมูลล่วงหน้า ในขณะที่แบบคอนเจสชัน คอนโทรลจะเพิ่มฮิตเรทได้ 3% แต่จะเพิ่มเวลาดึงเอกสารที่ไม่พบในแคชแค่ 4.6% ทำให้สามารถลดเวลาเข้าถึงเอกสารได้ จากการทดสอบนี้แสดงให้เห็นว่าในระบบที่มีแบนวิดท์เหลืออยู่น้อย การดึงข้อมูลล่วงหน้าที่ใช้การควบคุมแบบคอนเจสชัน คอนโทรล จะเหมาะสมกว่าแบบเลเวล คอนโทรล

ตารางที่ 4.3 **เปรียบเทียบการดึงข้อมูลล่วงหน้าแบบต่าง ๆในระบบจำลองที่มีแบนวิดท์ 4 เมกกะบิต**

	แบบไม่ดึงข้อมูลล่วงหน้า	แบบ เลเวล คอนโทรล	แบบ คอนเจสชัน คอนโทรล	แบบจับคู่ (0.01)
ฮิตเรท	37.65%	41.25%	40.61%	45.21%
เวลาดึงเอกสารที่ไม่พบในแคช	4.16 sec	4.46 sec	4.35 sec	12175.69 sec
อัตราถ่ายโอนต่อคำร้องขอ	3996.1 byte/sec	3866.56 byte/sec	3910.49 byte/sec	574.38 byte/sec
เวลาเข้าถึงเอกสาร	2.61 sec	2.63 sec	2.60 sec	1,1618.41 sec
ปริมาณแบนวิดท์ที่ใช้	18,105,813,099 byte	26,915,617,624 byte (148.66% ของระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า)	25,245,267,606 byte (139.43% ของระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า)	50,376,603,952 byte (343.6% ของระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้า)

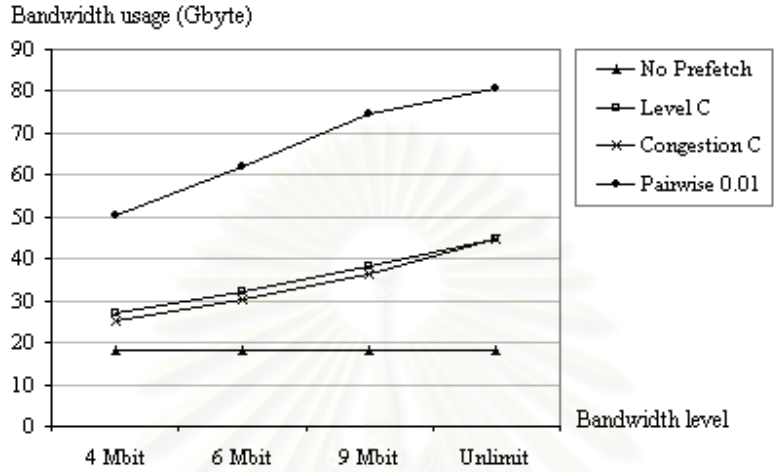
ในหัวข้อต่อไปเราจะทำการจำลองระบบที่ใช้การดึงข้อมูลล่วงหน้าของวิธีควบคุมทั้ง 2 แบบ ในระดับแบนวิดท์ที่แตกต่างกันเพื่อทดสอบถึงการเปลี่ยนแปลงของประสิทธิภาพและผลกระทบที่เกิดขึ้นของทั้ง 2 แบบเมื่อทำงานในสภาพแวดล้อมที่เปลี่ยนไป โดยเราจะทำการจำลองการดึงข้อมูลล่วงหน้าแบบเดิมที่ใช้ขอบเขต 0.01 ในแบนวิดท์แต่ละระดับด้วยเพื่อเปรียบเทียบผลที่เกิดขึ้น

4.2.3 การทดสอบประสิทธิภาพของระบบเมื่อเปลี่ยนแปลงระดับแบนวิดท์ของเครือข่าย

ในหัวข้อนี้ เราจะทำการเปรียบเทียบประสิทธิภาพของการดึงข้อมูลล่วงหน้าแบบเดิมและแบบควบคุมแบนวิดท์ โดยจะจำลองในระบบเครือข่ายที่มีระดับแบนวิดท์เท่ากับ 4 เมกกะบิต, 6 เมกกะบิต, 9 เมกกะบิต และ ที่ปริมาณแบนวิดท์ไม่จำกัด เพื่อเปรียบเทียบประสิทธิภาพของระบบควบคุมแบนวิดท์เมื่อทำงานในระบบที่มีปริมาณการใช้งานแบนวิดท์เปลี่ยนไป

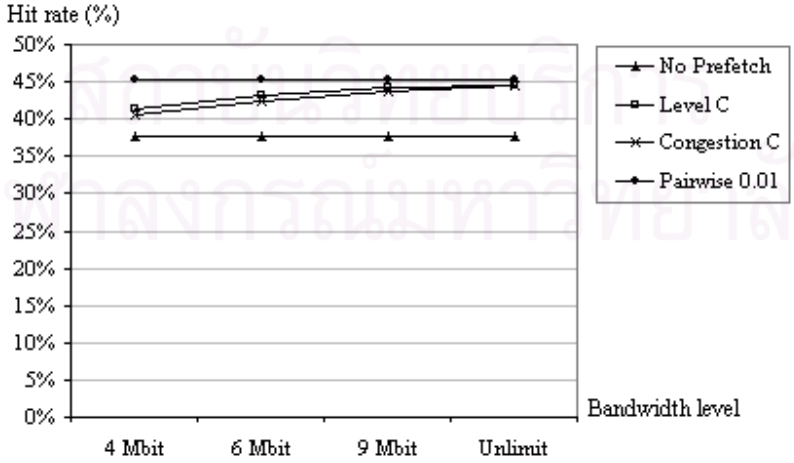
จากรูปที่ 4.6 จะเห็นได้ว่าที่ปริมาณการใช้งานแบนวิดท์ของการดึงข้อมูลล่วงหน้าแบบควบคุมแบนวิดท์ทั้ง 2 แบบจะค่อยๆ เพิ่มขึ้นตามระดับแบนวิดท์ที่เพิ่มขึ้นเนื่องจากระบบมีแบนวิดท์มากขึ้นแต่มีการใช้งานเท่าเดิมทำให้มีแบนวิดท์ที่ว่างเหลืออยู่ให้ระบบใช้ในการดึงข้อมูลล่วงหน้ามากขึ้น โดยแบบเลเวล คอนโทรลจะใช้แบนวิดท์ 139.40% ของระบบที่ไม่ได้ดึงข้อมูลล่วงหน้า ที่ 4 เมกกะบิตและเพิ่มขึ้นจนถึง 247.32% ในการจำลองที่ไม่จำกัดแบบวิดท์ ส่วนแบบคอนเจสชัน คอนโทรลจะใช้แบนวิดท์ตั้งแต่ 148.66% จนถึง

247.32% โดยที่ระบบการดึงข้อมูลล่วงหน้าแบบเดิมที่ใช้ขอบเขต 0.01 จะใช้แบนวิดท์ถึง 278.23% ตั้งแต่ที่ระดับ 4 เมกกะบิต และเพิ่มขึ้นเรื่อยๆเมื่อเพิ่มระดับแบนวิดท์ ซึ่งตามสมมุติฐานแล้วปริมาณการใช้แบนวิดท์



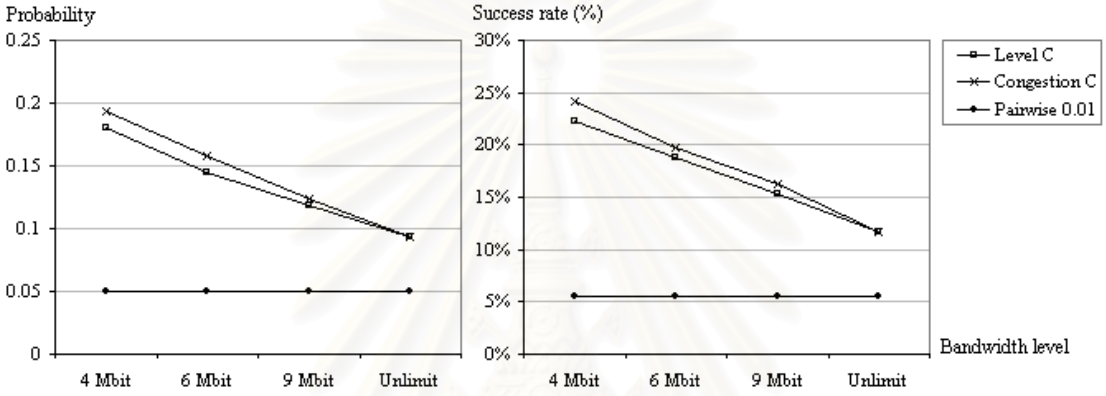
รูปที่ 4.6 ปริมาณการใช้แบนวิดท์ของการจำลองแต่ละแบบในแบนวิดท์แต่ละระดับ

ของการดึงข้อมูลล่วงหน้าแบบเดิมจะต้องค่าคงที่ในทุกระดับแบบวิดท์ แต่ในการจำลองแบบที่ใช้ขอบเขต 0.01 จะมีคำร้องขอค้างอยู่ในระบบมากในขณะที่โปรแกรมสิ้นสุดการจำลองซึ่งทำให้ปริมาณแบนวิดท์ที่ระบบถ่ายโอนได้มีค่าไม่เท่ากัน จากกราฟเราจะเห็นได้ว่าที่แบนวิดท์ระดับเดียวกัน แบบเลเวล คอนโทรลจะใช้แบนวิดท์มากกว่าแบบคอนเจสชัน คอนโทรลเล็กน้อย แต่จะมีค่าเท่ากันในระบบจำลองที่ไม่จำกัดแบนวิดท์เนื่องจากการควบคุมแบนวิดท์จะไม่มีผลในกรณีที่เครือข่ายมีแบนวิดท์ว่างอยู่มากพอที่ระบบดึงข้อมูลล่วงหน้าสามารถใช้ดึงเอกสารที่ทำนายล่วงหน้าได้ทั้งหมด



รูปที่ 4.7 ฮิตเรทของการจำลองแต่ละแบบในแบนวิดท์แต่ละระดับ

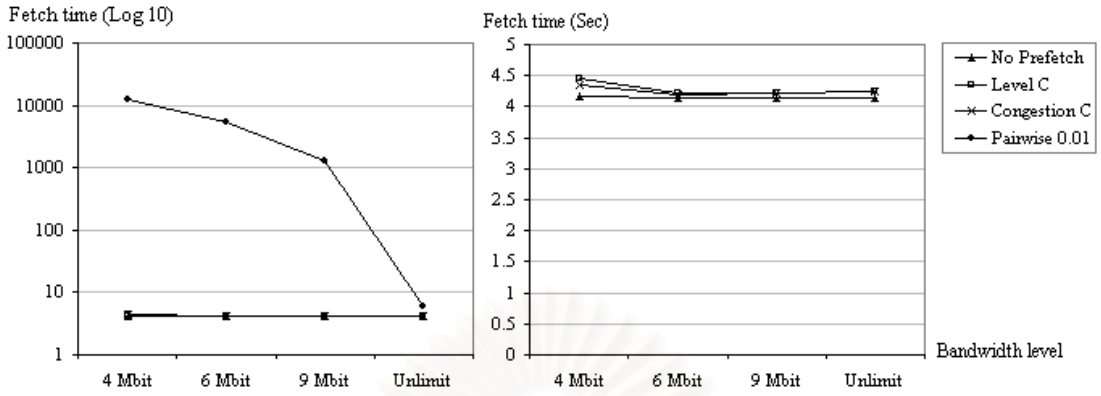
เราจะเห็นการเปลี่ยนแปลงของฮิตเรทของแต่ละแบบได้ในรูปที่ 4.7 ซึ่งค่าฮิตเรทของระบบที่ไม่ใช้การดึงข้อมูลล่วงหน้าคงที่โดยจะมีค่า 37.65% ในการจำลองทุกแบบ ส่วนฮิตเรทของระบบที่ใช้การดึงข้อมูลล่วงหน้าแบบเดิมที่ใช้ขอบเขต 0.01 จะมีค่าอยู่ที่ 45.21% ในแบนวิดท์ทุกระดับเนื่องจากการดึงข้อมูลล่วงหน้าถูกกำหนดขอบเขตตายตัว แต่ฮิตเรทของแบบควบคุมแบนวิดท์ทั้ง 2 แบบจะเพิ่มขึ้นเรื่อยๆ จนมีค่าเท่ากับ 44.59% ซึ่งใกล้เคียงกับแบบที่ใช้ขอบเขต 0.01



ก) ความน่าจะเป็นของการจำลองแต่ละแบบ ข) อัตราความสำเร็จของการจำลองแต่ละแบบ

รูปที่ 4.8 แสดงค่าความน่าจะเป็นและอัตราความสำเร็จของการดึงข้อมูลล่วงหน้าแต่ละแบบ

รูป 4.8 จะแสดงถึงการเปลี่ยนแปลงที่เกิดขึ้นกับค่าเฉลี่ยของค่าความน่าจะเป็นของเอกสารที่ถูกดึงข้อมูลล่วงหน้า และอัตราความสำเร็จของการดึงข้อมูลล่วงหน้าแต่ละแบบ โดยค่าความน่าจะเป็นของเอกสารของการดึงข้อมูลล่วงหน้าแบบเดิมจะคงที่อยู่ที่ 0.5 และอัตราความสำเร็จจะมีค่าอยู่ที่ 5.5% ในทุกระดับแบนวิดท์ แต่ในระบบที่ใช้การดึงข้อมูลล่วงหน้าแบบควบคุมแบนวิดท์ ค่าความน่าจะเป็นของเอกสารจะมีค่า 0.18 ในแบบเลเวล คอนโทรล และ0.19 ในแบบคอนเจสชัน คอนโทรล และอัตราความสำเร็จของทั้ง 2 แบบจะเป็นมีค่าเป็น 22.32% และ 24.12% ตามลำดับ และจะลดลงจนมีค่าเท่ากับ 11.64% ที่ความน่าจะเป็น 0.09 ในการจำลองที่ไม่จำกัดแบดวิดท์ สาเหตุที่ค่าความน่าจะเป็นและอัตราความสำเร็จของแบบควบคุมทั้ง 2 แบบลดลงเนื่องจากเมื่อระดับของแบนวิดท์สูงขึ้นจะทำให้มีแบนวิดท์ที่ว่างมากขึ้น ซึ่งการดึงข้อมูลล่วงหน้าแบบควบคุมแบนวิดท์จะเพิ่มปริมาณการดึงเอกสารล่วงหน้าทำให้เอกสารที่มีค่าความน่าจะเป็นน้อยมีโอกาสถูกร้องขอมาไว้ในแคช ซึ่งจะทำให้ค่าเฉลี่ยของความน่าจะเป็นของเอกสาร และอัตราความสำเร็จโดยเฉลี่ยลดต่ำลง แต่การเอกสารที่ถูกร้องขอมาไว้ในแคชมากขึ้นทำให้ฮิตเรทสูงขึ้น



ก)แสดงการจำลองทุกแบบในรูปของล็อก10 ข) แสดงการจำลองทุกแบบยกเว้นแบบที่ใช้ขอบเขต 0.01

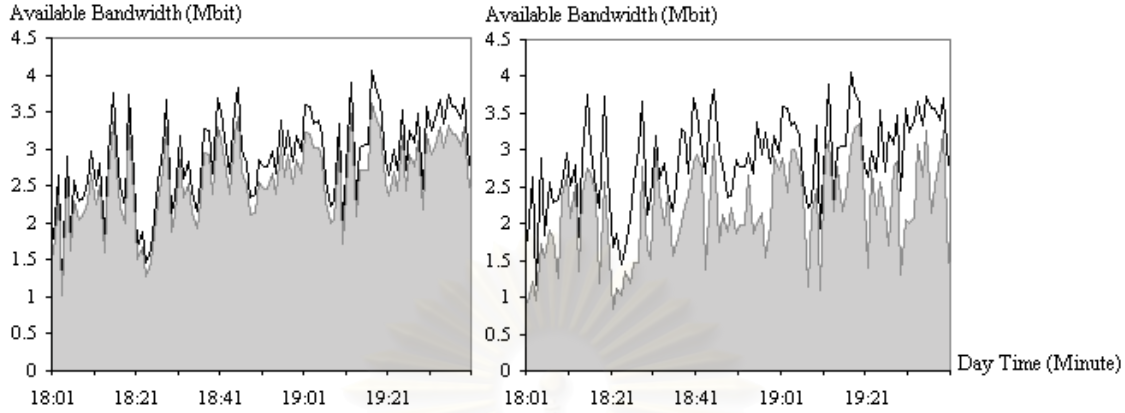
รูปที่ 4.9 เวลาที่ใช้ในการดึงเอกสารที่ไม่พบในแคชของการจำลองแต่ละแบบ

กราฟในรูปที่ 4.9 แสดงให้เห็นว่าแบบที่ใช้ขอบเขต 0.01 ทำให้เวลาที่ใช้ดึงข้อมูลของระบบเพิ่มขึ้นมาก ในขณะที่แบบควบคุมแบนวิดท์ทั้ง 2 แบบจะทำให้เวลาดึงข้อมูลของระบบเพิ่มขึ้นเล็กน้อยโดยจะเพิ่มขึ้น 4.5% - 7% ที่ระดับแบนวิดท์ 4 เมกกะบิตซึ่งเป็นสภาพที่มีแบนวิดท์น้อย และเพิ่มขึ้น 1% - 1.6% ในระดับแบนวิดท์ที่ 6 เมกกะบิต และ 9 เมกกะบิต ซึ่งจะเห็นได้ว่าและผลกระทบที่เกิดกับระบบจะลดน้อยลงเมื่อจำลองระบบในระดับแบนวิดท์ที่สูงขึ้น แต่ในการจำลองที่ไม่จำกัดแบนวิดท์เวลาในการดึงเอกสารจะเพิ่มขึ้น 2.19% ซึ่งมากกว่าแบบที่จำกัดแบนวิดท์เนื่องจากจำนวนคำร้องขอที่เพิ่มขึ้นมากทำให้เวลาที่คำร้องขอต้องรอในคิวของพร็อกซีมีผลต่อเวลาดึงเอกสารโดยรวมของระบบ

4.3 การวิเคราะห์

จากการทดสอบในหัวข้อ 4.1 เราจะพบว่า ในระบบที่มีแบนวิดท์จำกัดการดึงข้อมูลล่วงหน้าที่ใช้ขอบเขตต่ำจะไม่เพียงเพิ่มปริมาณการใช้งานของแบนวิดท์ในระบบเท่านั้น แต่คำร้องขอจำนวนมากที่เกิดขึ้นจากการดึงข้อมูลล่วงหน้าจะทำให้พร็อกซีและระบบเครือข่ายต้องรับภาระในการจัดการคำร้องขอเหล่านี้ ส่งผลให้บริการที่ผู้ใช้งานได้รับจากเครือข่ายมีคุณภาพลดลง ซึ่งผลกระทบที่เกิดขึ้นนี้จะทำให้เวลาในการเข้าถึงเอกสารของผู้ใช้งานสูงขึ้น ทั้งๆ ที่ระบบการดึงข้อมูลล่วงหน้าช่วยเพิ่มฮิตเรทในระบบแล้วก็ตาม และในกรณีที่ปริมาณแบนวิดท์ที่เกิดขึ้นมีมากกว่าที่ระบบจะรองรับได้มาจะทำให้ระบบเครือข่ายแทบไม่สามารถให้บริการผู้ใช้งานได้เลย

จากการทดสอบการดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์ในหัวข้อ 4.2 พบว่าในสภาพแวดล้อมที่มีแบนวิดท์จำกัด วิธีใหม่นี้จะเพิ่มฮิตเรทได้น้อยกว่าการดึงข้อมูลล่วงหน้าแบบเดิม แต่จะให้ผลกระทบต่อระบบน้อยกว่ามากทำให้สามารถเพิ่มประสิทธิภาพโดยรวมได้ดีกว่าการดึงข้อมูลล่วงหน้าแบบเดิม และเมื่อมีแบนวิทด์ในระบบว่างมากขึ้น การดึงข้อมูลล่วงหน้าแบบใหม่จะเพิ่มปริมาณการดึงข้อมูลมากขึ้นส่งผลให้ฮิตเรทของระบบเพิ่มมากขึ้น โดยสร้างผลกระทบที่เกิดขึ้นกับระบบน้อยลงด้วย



ก) แบบเลเวล คอนโทรล ข) แบบคอนเจสชัน คอนโทรล

รูปที่ 4.10 กราฟแสดงการคำนวนปริมาณการใช้แบนวิดท์ที่ว่างอยู่ของระบบของวิธีควบคุมแบนวิดท์ทั้ง 2 วิธี

จากการทดสอบเราจะพบว่าวิธีควบคุมแบนวิดท์ทั้ง 2 แบบจะให้ผลที่ใกล้เคียงกันแต่แบบเลเวล คอนโทรล แต่แบบเลเวล คอนโทรล สามารถเพิ่มฮิตเรตได้มากกว่า แต่จะส่งผลกระทบต่อระบบมากกว่าแบบ คอนเจสชัน คอนโทรลเนื่องจากกลไกการควบคุมแบนวิดท์ที่แตกต่างกันของทั้ง 2 แบบ โดยเราจะสามารถ อธิบายได้ด้วยรูปที่ 4.10 โดยพื้นที่แรเงาจะหมายถึงแสดงถึงแนวโน้มของปริมาณการใช้งานแบนวิดท์ของการ ดึงข้อมูลล่วงหน้าแต่ละวิธี และจุดที่พื้นที่แรเงาทับกับกราฟจะหมายถึงช่วงเวลาที่การดึงข้อมูลล่วงหน้าใช้ แบนวิดท์ของระบบเต็มที่ซึ่งจะก่อให้เกิดผลกระทบกับระบบ จากรูปที่ 4.10ก เราจะเห็นได้ว่ากลไกการควบคุม แบนวิดท์ของแบบเลเวล คอนโทรลจะทำให้มีการดึงข้อมูลล่วงหน้าสามารถใช้แบนวิดท์ได้มากกว่าแบบคอน เจสชัน คอนโทรล แต่เนื่องจากปริมาณการใช้งานแบนวิดท์ของระบบมีการเปลี่ยนแปลงขึ้นลงอยู่ตลอดเวลา ทำให้เกิดจุดที่กราฟตัดกับพื้นที่แรเงาของแบบเลเวล คอนโทรลมากกว่า แสดงว่าผลกระทบที่เกิดกับระบบจะ มีมากกว่าแบบคอนเจสชัน คอนโทรลในรูปที่ 4.10ข ที่มีการปรับปริมาณการใช้แบนวิดท์ขึ้นลงอยู่ตลอดเวลา ทำให้เกิดกรณีที่มีการใช้แบนวิดท์ทั้งหมดน้อย แต่จะทำให้ใช้ปริมาณแบนวิดท์ที่ว่างอยู่ได้น้อยกว่าเช่นกัน

จากการจำลองการทำงานของทั้ง 2 วิธีในระดับแบนวิดท์ต่างๆ เราสามารถสรุปได้ว่าการควบคุม แบนวิดท์แบบเลเวลคอนโทรลจะให้ผลที่ดีกว่าในกรณีที่มีแบนวิดท์เพียงพอที่จะทำการดึงข้อมูลล่วงหน้า แต่ ในเครือข่ายที่สามารถใช้งานแบนวิดท์ได้น้อย การควบคุมแบนวิดท์แบบคอนเจสชัน คอนโทรลจะเหมาะสม กว่าเนื่องจากจะสร้างผลกระทบต่อระบบน้อยกว่าแบบเลเวล คอนโทรล

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

จากการที่ได้ศึกษาเกี่ยวกับการดึงข้อมูลล่วงหน้าพบว่าการดึงข้อมูลล่วงหน้ามีปัญหาที่สำคัญคือ การใช้งานแบนวิดท์จำนวนมาก การควบคุมปริมาณการดึงเอกสารที่แน่นอนตายตัว และการใช้ทรัพยากรจำนวนมากในการทำนายทำให้เซิร์ฟเวอร์ที่ระบบดึงข้อมูลล่วงหน้าทำงานอยู่ไม่สามารถทำงานได้เต็มที่ ซึ่งงานวิจัยในด้านนี้ส่วนใหญ่จะพัฒนาวิธีในการทำนายเอกสารล่วงหน้าให้มีความแม่นยำโดยทดสอบประสิทธิภาพในสภาวะที่มีแบนวิดท์ไม่จำกัด ซึ่งในระบบเครือข่ายจริงจะมีแบนวิดท์จำกัดและปริมาณแบนวิดท์ที่เพิ่มขึ้นจากการดึงข้อมูลล่วงหน้าจะส่งผลกระทบต่อระบบ ในงานวิทยานิพนธ์นี้ได้เสนอวิธีการดึงข้อมูลล่วงหน้าแบบใหม่ ที่จะวัดปริมาณการใช้งานจากเครือข่ายก่อนแล้วใช้แบนวิดท์ที่ว่างในการดึงข้อมูลล่วงหน้าภายใต้ระบบควบคุมแบนวิดท์ที่สร้างขึ้นเพื่อไม่ให้ปริมาณแบนวิดท์ที่ใช้ในการดึงข้อมูลล่วงหน้ามากจนรบกวนผู้ใช้งานในเครือข่ายและทำให้สามารถควบคุมการใช้งานแบนวิดท์ให้เปลี่ยนแปลงตามสภาวะแวดล้อมของเครือข่าย และได้ออกแบบให้ระบบดึงข้อมูลล่วงหน้าตั้งอยู่ในระบบเครือข่ายเดียวกับพร็อกซีและทำนายโดยการดักคำร้องขอที่ส่งมายังพร็อกซีและส่งคำร้องขอของเอกสารที่ได้ทำนายไว้ให้กับพร็อกซีเหมือนกับเป็นผู้ใช้งานคนหนึ่ง เพื่อแก้ปัญหาการใช้ทรัพยากรจำนวนมากในการทำนาย

เพื่อเป็นการทดสอบสมมุติฐานที่ได้คิดขึ้น ในวิทยานิพนธ์นี้ได้สร้างโปรแกรมจำลองการทำงานของระบบเครือข่ายขึ้น และใช้ข้อมูลการใช้เว็บที่ได้จากพร็อกซีของสำนักงานเทคโนโลยีสารสนเทศ จุฬาลงกรณ์มหาวิทยาลัย โดยเราได้จำลองระบบที่ได้คิดขึ้นเปรียบเทียบกับระบบดึงข้อมูลล่วงหน้าแบบเดิมในสภาวะแวดล้อมต่างๆกัน ซึ่งสามารถสรุปผลที่ได้จากการวิจัย ปัญหาและข้อจำกัดที่พบ รวมไปถึงการเสนอแนะแนวทางการพัฒนาต่อไปได้ดังนี้

5.1 สรุปผลการวิจัย

1. ในระบบเครือข่ายที่มีแบนวิดท์จำกัด ปริมาณแบนวิดท์จำนวนมากที่เกิดจากการดึงข้อมูลล่วงหน้าที่มีขอบเขตต่ำทำให้แบนวิดท์ในระบบเครือข่ายไม่เพียงพอต่อการใช้งาน ส่งผลให้เวลาเข้าถึงเอกสารของผู้ใช้งานเพิ่มขึ้นทั้งๆ ที่การดึงข้อมูลล่วงหน้าสามารถเพิ่มฮิตเรทของระบบ
2. การดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์สามารถเพิ่มฮิตเรทได้น้อยกว่าการดึงข้อมูลล่วงหน้าแบบเดิม แต่ว่าจะส่งผลกระทบต่อระบบน้อยกว่าทำให้สามารถลดเวลาเข้าถึงข้อมูลได้ดีกว่าการดึงข้อมูลล่วงหน้าแบบเดิม โดยในการทดลองที่ 4.2.1 จะเห็นได้ว่า ในเครือข่ายที่จำกัดแบนวิดท์ 6 เมกกะบิต การดึงข้อมูลล่วงหน้าแบบควบคุมแบนวิดท์มีค่าฮิตเรทเท่ากับ 42.42% ซึ่งน้อยกว่าการดึงข้อมูลล่วงหน้าแบบเดิมที่สามารถเพิ่มฮิตเรตได้เป็น 45.21% แต่ว่าจะสามารถลดเวลาเข้าถึงเอกสารได้ 7.28% ในขณะ

ที่การดึงข้อมูลล่วงหน้าแบบเดิมทำให้เวลาเข้าถึงเอกสารเพิ่มขึ้นมากจนระบบไม่สามารถทำงานได้ แต่เมื่อมีระบบมีแบนวิดท์ว่างมากขึ้น การดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์สามารถเพิ่มฮิตเรทขึ้นมาจนใกล้เคียงกับการดึงข้อมูลล่วงหน้าแบบเดิมโดยจะเพิ่มขึ้นเป็น 43.83% เมื่อเพิ่มระดับแบนวิดท์เป็น 9 เมกกะบิต ทำให้สามารถลดเวลาเข้าถึงเอกสารได้ 8.65% ในขณะที่การดึงข้อมูลล่วงหน้าแบบเดิมจะให้ค่าฮิตเรทเท่าเดิม

3. เมื่อเปรียบเทียบวิธีควบคุมแบนวิดท์ทั้ง 2 วิธีพบว่าแบบเลเวล คอนโทรลสามารถเพิ่มฮิตเรทได้มากกว่าแบบคอนเจสชันคอนโทรล แต่จะส่งผลกระทบต่อระบบมากกว่า ซึ่งในระบบที่มีแบนวิดท์เหลืออยู่มาก วิธีแบบเลเวล คอนโทรลจะให้สามารถลดเวลาเข้าถึงเอกสาร ได้มากกว่าวิธีคอนเจสชัน คอนโทรล แต่ในระบบที่มีแบนวิดท์เหลืออยู่น้อย เช่น การจำลองระบบที่มีระดับแบนวิดท์ 4 เมกกะบิต ในการทดลอง 4.2.2 พบว่าเลเวลคอนโทรล จะสามารถลดเวลาเข้าถึงเอกสารได้น้อยกว่าแบบคอนเจสชัน คอนโทรล เนื่องผลกระทบต่อระบบของแบบเลเวล คอนโทรลสร้างขึ้นมีมากกว่าผลที่ได้รับจากฮิตเรทที่เพิ่มขึ้น

จากที่กล่าวมาข้างต้นเราสามารถสรุปได้ว่า การดึงข้อมูลล่วงหน้าที่เปลี่ยนแปลงตามแบนวิดท์สามารถเพิ่มฮิตเรทของพร็อกซีเซิร์ฟเวอร์ในขณะที่ส่งผลกระทบต่อระบบเครือข่ายเพียงเล็กน้อยเท่านั้น และระบบสามารถเปลี่ยนแปลงปริมาณการดึงข้อมูลล่วงหน้าได้ตามปริมาณแบนวิดท์ที่ว่างของระบบได้ โดยวิธีควบคุมแบนวิดท์แบบเลเวลคอนโทรลจะเหมาะสมกับระบบเครือข่ายที่มีแบนวิดท์เหลืออยู่มาก ส่วนวิธีควบคุมแบนวิดท์แบบคอนเจสชัน คอนโทรลจะเหมาะสมกับระบบเครือข่ายที่มีแบนวิดท์เหลืออยู่น้อย

5.2 ปัญหาและข้อจำกัดที่ได้พบจากการวิจัย

1. ข้อมูลสถิติที่ใช้ในการทำนายโดยวิธีแบบจับคู่มีค่าเฉลี่ยโดยรวมของความน่าจะเป็นค่อนข้างต่ำทำให้การเปรียบเทียบผลที่ได้จากการทดสอบไม่ชัดเจนเท่าที่ควร
2. เนื่องจากโปรแกรมจำลองที่สร้างขึ้นค่อนข้างซับซ้อนและต้องใช้หน่วยความจำจำนวนมากในการจำลองแคชของระบบ ประกอบกับข้อมูลการใช้เว็บที่ใช้ในการทดสอบมีขนาดใหญ่มาก (ประมาณ 3-4 ล้านคำร้องขอ) ทำให้การจำลองต้องใช้เครื่องที่มีประสิทธิภาพค่อนข้างสูงในการจำลอง และใช้เวลานานในการจำลองแต่ละครั้ง
3. ข้อมูลการใช้เว็บที่ใช้ในการจำลองเป็นข้อมูลที่ได้จากระบบที่มีแบนวิดท์จำกัดทำให้อัตราการถ่ายโอนและเวลาที่ใช้ในการสร้างการเชื่อมต่อไม่แน่นอนทำให้ไม่สามารถหาค่าที่จะใช้ในการจำลองได้ จึงต้องอ้างอิงจากงานวิจัยอื่น หรือทำการหาอย่างคร่าวๆ จากข้อมูลการใช้เว็บ

5.3 ข้อเสนอแนะ

1. ระบบดึงข้อมูลที่วิทยานิพนธ์นี้ได้เสนอไว้สามารถนำไปประยุกต์ใช้การทำนายแบบใดก็ได้ ถ้ามีการนำระบบทำนายที่มีประสิทธิภาพเข้ามาใช้จะสามารถเพิ่มประสิทธิภาพของระบบได้
2. การทดสอบในวิทยานิพนธ์นี้ได้ทำสอบประสิทธิภาพของการเปลี่ยนแปลงแบนวิดท์โดยใช้แคชที่มีขนาดคงที่ คือ 2 กิกกะไบท์ ซึ่งมีขนาดเท่ากับ 19.37% ของปริมาณข้อมูลทั้งหมดของเอกสารที่สามารถเก็บในแคชได้ และในงานวิจัยส่วนใหญ่จะทำการทดลองโดยสมมุติให้แคชมีขนาดไม่จำกัด จึงควรจะมีการทดสอบว่าควรใช้แคชขนาดเท่าไหร่เมื่อมีการนำการดึงข้อมูลล่วงหน้ามาใช้งานจึงจะมีประสิทธิภาพมากที่สุด
3. ในวิทยานิพนธ์นี้ได้เสนอแนวทางในการเลือกวิธีควบคุมแบนวิดท์ที่เหมาะสมระบบเครือข่าย ควรจะมีการพัฒนาระบบให้สามารถปรับวิธีการควบคุมแบนวิดท์ให้เหมาะสมกับสภาพเครือข่ายโดยอัตโนมัติ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Wang, J. A Survey of Web Caching Schemes for the Internet. ACM Computer Communication Review (October 1999) : 36 - 46.

[2] Baentsch, M., et al. Enhancing the Web 's Infrastructure: From Caching to Replication. IEEE Communications Magazine (March-April 1997).

[3] Homedalh, V., Smith, B. and Yang, T. Cooperative Caching of Dynamic Content on Distributed Web Server. IEEE Journal (1998) : 243 - 250.

[4] Padmanabhan, V., N. and Mogul, J., C. Improving WWW Latency. Proceedings of *Second International World Wide Web Conference* Chicago, Illinois, USA,1994.

[5] Wang, Z. and Croweroft, J. Prefetching in World Wide Web. Proceedings of Global Telecommunications Conference (1996) : 28 - 32.

[6] Horng, Y.,Lin, W. and Mei, H. Hybrid Prefetching for WWW Proxy Servers. Proceeding of Parallel and Distributed Systems (1998) : 541 – 548.

[7] Padmanabhan, V., N. and Mogul, J., C. Using Predictive Prefetching to Improve World Wide Web Latency. ACM Computer Communication Review 26 (1996).

[8] Jiang, Z. and Kleinrock, L. An Adaptive Network Prefetch Scheme. IEEE Journal on Selected Areas in Communications (April 1998) : 358 – 368.

[9] Cohen, E., Krishanamurthy, B. and Rexford, J. Efficient Algorithms for Predicting Requests to Web Servers. Proceeding of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (1999): 284 - 293.

[10] Zhao, Z. and Wang, S. Measurement Based Intelligent Prefetch and Cache Technique in web. Proceeding of IEEE Canadian Conference (1999) : 168 - 173.

[11] Chinen, K. and Yamaguchi, S. An Interactive Prefetching Proxy Server for Improvement of WWW Latency. Proceeding of INET '97 Conference. Kuala Lumpur, Malaysia,1997.

[12] Markatos, E., P. and Chronaki, C., E. A Top-10 Approach to Prefetching the Web. Proceeding *of INET '98 Conference*. Geneva, Switzerland, July 1998.

[13] Toucha, J., D. and Farber, D., J. An Experiment in Latency Reduction. Proceedings of INFOCOM '94, Networking for Global Communications (1994): 175 - 181.

[14] Maltzahn, C. and Richardson, K. On Bandwidth Smoothing. Proceedings of 4th International Web Caching Workshop. San Diego, California, 1999.

[15] Crovella, M. and Barford, P. The network effects of prefetching. Proceedings of Seventeenth Annual Joint Conference of IEEE Computer and Communication Societies (1998) : 1232 - 1239.

[16] Stevens, W., R. and Wright, G. R. TCP/IP illustrated. Vol. I: Protocols. Mass: Addison-Wesley,1994.

[17] Kirch, O. Linux Network Administrator' s Guide. Mass: O' Really & Associates,1995

[18] Snort. Available from : http://www.snort.org

[19] Stalling, W. SNMP,SNMPv2, and RMON: Practical Network Management Mass: Addison-Wesley,1996.

[20] MTRG (Multi Router Traffic Grapher). Available from : http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

โปรแกรมที่ใช้ในการจำลอง

1. รูปแบบของการเรียกใช้โปรแกรม

   presimv63.pl <ชื่อไฟล์ข้อมูลการใช้เว็บ> <ชื่อของแฟ้มข้อมูลที่ใช้เก็บผลลัพธ์>

   ตัวอย่าง

   presimv63.pl ./cuproxy-31-07-01-162.txt ./result31/

2. ตัวอย่างของข้อมูลการใช้เว็บที่ใช้ในโปรแกรม

   " 979552882.905 8583 161.200.96.192 TCP_MISS/0 14426 GET

   http://www.soccersuck.com/sslink/eye.gif - DIRECT/www.soccersuck.com binary/* "

3. ไฟล์ที่ใช้ร่วมกับโปรแกรม

   - parselog2.pp - ใช้ในการแยกฟิลด์ของข้อมูลการใช้เว็บ
   - Mybw53.pm - โมดูลที่ใช้ควบคุมแบนวิดท์
   - Mycachelruv3.pm - โมดูลที่ใช้จำลองการทำงานของแคช
   - Mypref63.pm - โมดูลที่ใช้จำลองการดึงข้อมูลล่วงหน้าแบบคอนเจสชัน คอนโทรล
   - Mypref63le.pm - โมดูลที่ใช้จำลองการดึงข้อมูลล่วงหน้าแบบเลเวล คอนโทรล
   - Mypref63ag.pm - โมดูลที่ใช้จำลองการดึงข้อมูลล่วงหน้าแบบเดิมที่ใช้วิธีจับคู่
   - LinkList.pm - โมดูลที่ใช้สร้างรายการโยงในระบบ

   ทั้งนี้ในการทำงานระบบจะต้องมีโมดูล Digest::MD5 ที่ใช้ในการเข้ารหัสแบบ MD5 และโมดูล DBI ในการเชื่อมต่อกับฐานข้อมูลที่เก็บสถิติการใช้งานเว็บด้วย

4. รูปแบบของผลลัพธ์ - โปรแกรมจำลองจะสร้างไฟล์ที่เก็บข้อมูลจากการจำลองออกมา 6 ไฟล์ คือ

   - trace file - บันทึกสภาพการทำงานระบบระหว่างการจำลอง
   - user time histogram - เก็บเวลาดึงเอกสารของผู้ใช้งานแบบฮิสโตแกรม
   - prefetch time histogram - เก็บเวลาดึงเอกสารของการดึงข้อมูลล่วงหน้าแบบฮิสโตแกรม
   - total elapse histogram - เก็บเวลาดึงเอกสารโดยรวมแบบฮิสโตแกรม
   - hit rate file - บันทึกค่าฮิตเรทโดยเฉลี่ยของแต่ละชั่วโมง
   - result - เก็บผลสรุปที่ได้จากการจำลอง

ภาคผนวก ข

การดักจับคำร้องขอจากแลน (LAN) และ การวัดแบนวิดท์ในระบบเครือข่าย

การดักจับคำร้องขอจากแลน

การดักจับคำร้องขอจะทำได้โดยดักจับกลุ่มข้อมูล (Packet) ที่วิ่งอยู่ภายในแลน ซึ่งโดยปกติการ์เน็ตเวิร์คของเครื่องจะทำการกรองกลุ่มข้อมูลที่ไม่ได้ระบุที่อยู่มายังเครื่องคอมพิวเตอร์ออก จึงต้องทำการตั้งค่าให้การ์เน็ตเวิร์คทำงานแบบโพรมิคิวอัส (Promiscuous mode)[17] ซึ่งจะรับทุกกลุ่มข้อมูลที่ผ่านเข้ามา โดยในระบบปฏิบัติการลีนุกซ์ ผู้ดูแลระบบ (Root) จะสามารถตั้งค่าโดยใช้คำสั่ง

```
# /sbin/ifconfig eth0 promisc
```

แต่เนื่องจากกลุ่มข้อมูลที่ถูกส่งผ่านภายในแลนมีหลายประเภทจึงต้องทำการกรองเอาเฉพาะกลุ่มข้อมูลที่เกิดจากคำร้องขอที่ผู้ใช้งานส่งมาและต้องแยกข้อมูลที่ต้องการออกจากกลุ่มข้อมูล ในวิทยานิพนธ์นี้จะทดลองใช้โปรแกรม Snort [18] ซึ่งเป็นโปรแกรมตรวจสอบผู้บุกรุก (Intruder Detection) โดยนำมาประยุกต์ให้ดักจับกลุ่มข้อมูลที่ส่งคำร้องขอผ่านพอร์ท 80 ออกไปนอกระบบ ซึ่งก็คือคำร้องขอที่ผู้ใช้งานร้องขอนั้นเอง โดยเราจะตั้งกฎการดักจับ (Snort Rules) ไว้ดังนี้คือ

```
alert tcp any any -> any 80 (session:printable;content:"HTTP";)
```

ซึ่งโปรแกรม Snort จะทำงานตามกฎคือ จะทำการดักจับกลุ่มข้อมูลที่ใช้โปรโตคอล ทีซีพี จากทุกเครื่องในระบบแลนที่ส่งไปยังเครื่องใดก็ได้ที่เปิดพอร์ต 80 อยู่ และข้อมูลจะต้องมีคำว่า "HTTP" ซึ่งตัวอย่างของผลที่ได้จากการดักจับจะเป็นดังนี้

```
GET /us.yimg.com/p/nm/20010911/amdf50834.jpg HTTP/1.1
Accept: */*
Referer: http://www.yahoo.com/
Accept-Language: th
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)
Host: us.news2.yimg.com
Connection: Keep-Alive
```

จากกลุ่มข้อมูลข้างต้นจะอยู่ในรูปแบบของโพรโทคอลเอชทีทีพี ซึ่งข้อความบรรทัดแรกของกลุ่มข้อมูลจะบอกถึงที่อยู่ของเอกสารที่ผู้ใช้งานต้องการและข้อมูลบรรทัดที่ 7 จะระบุชื่อเซิร์ฟเวอร์ที่ผู้ใช้งานติดต่อ จากข้อมูลทั้ง 2 ส่วนจะทำให้เราได้ยูอาร์แอลของเอกสารที่ผู้ใช้งานร้องขอจากระบบแลนได้

ในการนำมาใช้กับระบบดึงข้อมูลล่วงหน้าที่ได้ออกแบบขึ้นอาจจะต้องปรับปรุงกฎการดักจับขึ้นใหม่ โดยให้ดักจับกลุ่มข้อมูลที่ส่งไปยังปลายทางที่ระบุเลขที่อยู่ไอพี (IP Address) ของพร็อกซี เซิร์ฟเวอร์ที่พอร์ท 8080 ซึ่งจะเขียนได้ดังนี้

```
alert tcp any any -> [proxy IP]/32 8080 (session:printable;content:"HTTP";)
```

การวัดแบนวิดท์ในระบบเครือข่าย

การวัดแบนวิดท์ในระบบเครือข่ายจะทำได้โดยอาศัย โพรโทคอลเอสเอ็นเอ็มพี (SNMP : Simple Network Management Protocol) [19] ซึ่งเป็นโพรโทคอลที่ใช้ในการจัดการระบบเครือข่าย โพรโทคอลเอสเอนเอ็มพี จะกำหนดให้อุปกรณ์ในเครือข่ายสร้างชุดข้อมูลที่เก็บข้อมูลของตัวเองไว้เรียกว่า เอ็มไอบี (MIB : Management Information Base) ซึ่งจะเก็บข้อมูลต่างๆ ของตัวอุปกรณ์ทั้งหมด ซึ่งสามารถเรียกใช้งานได้โดยผ่านโพรโทคอลเอสเอ็นเอ็มพี จากการวิเคราะห์โปรแกรมเอ็มทีจีอาร์ (MRTG : Multi Router Traffic Grahher)[20] ซึ่งเป็นโปรแกรมที่ใช้วัดปริมาณข้อมูลที่รับส่งผ่านอุปกรณ์จัดเส้นทาง (Router) พบว่าโปรแกรมเอ็มทีจีอาร์จะใช้ข้อมูลจากออบเจ็กต์ ifInOctets กับ ifOutOctets ซึ่งอยู่ในกลุ่มข้อมูล interfaces ในชุดข้อมูลเอ็มไอบี เป็นตัวคำนวณปริมาณข้อมูลที่ถูกส่งผ่านอุปกรณ์จัดเส้นทาง โดยออบเจ็คต์ ifInOctets จะบอกถึงจำนวนข้อมูล (Octet) ที่ผ่านเข้ามาในอุปกรณ์ ส่วนออบเจ็คต์ ifOutOctets จะเป็นตัวบอกปริมาณข้อมูลที่ถูกส่งออกจากอุปกรณ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

ผลงานตีพิมพ์

การทดสอบประสิทธิภาพของการดึงข้อมูลล่วงหน้าบนเวิลด์ไวด์เว็บ

Performance Evaluation of WWW Prefetching

พรศิษฐ์ อรัญสมาน[1] และณัฐวุฒิ หนูไพโรจน์[2]

Pornsit Arunsamarn and Natawut Nupairoj

Department of Computer Engineering, Faculty of Engineering

Chulalongkorn University, Bangkok, Thailand 10330

E-mail: g42par[1], natawut[2]@cp.eng.chula.ac.th

บทคัดย่อ : ในปัจจุบันมีผู้ใช้งานเวิลด์ไวด์เว็บเพิ่มขึ้นเป็นจำนวนมาก ซึ่งก่อให้เกิดปัญหาว่า ระบบเครือข่ายที่มีอยู่ไม่สามารถรองรับการใช้งานที่เพิ่มขึ้นมานี้ได้ ทำให้ผู้ใช้งานจำนวนมากต้องใช้เวลานานในการเข้าถึงเอกสารเวิลด์ไวด์เว็บ จึงได้มีการค้นคว้าหาวิธีที่จะทำให้ผู้ใช้งานสามารถใช้งานระบบเวิลด์ไวด์เว็บได้เร็วขึ้นบนระบบเครือข่ายที่มีอยู่เดิม การดึงข้อมูลล่วงหน้าเป็นวิธีหนึ่งที่จะช่วยลดเวลาการเข้าถึงเอกสารได้อย่างมีประสิทธิภาพ ในบทความนี้ เราจะประเมินประสิทธิภาพระบบเมื่อนำวิธีนี้เข้ามาใช้งานกับเครือข่ายในประเทศไทย

คำสำคัญ : พรีเฟทช์, เวิลด์ไวด์เว็บ, เวลาการเข้าถึงเอกสาร, ประสิทธิภาพ

Abstract : The number of internet users has been increased drastically recently. As the bandwidth of the network is limited, users are suffered from long delay while they are requesting WWW documents. To address the problem, many studies have been focused on the WWW prefetching which is one of efficient method to reduce retrieval time. In this paper we evaluated the potential of using this method when it was implemented in Thailand network environment.

Keyword : prefetch, WWW, retrieval time, performance

1. บทนำ

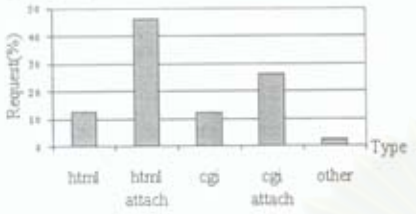
ในปัจจุบันมีงานวิจัยจำนวนมากที่หาวิธีที่จะทำให้ผู้ใช้งานเวิลด์ไวด์เว็บสามารถได้รับข้อมูลอย่างรวดเร็วผ่านทางเครือข่ายที่มีอยู่ในปัจจุบันได้ เว็บพรีเฟทชิ่ง หรือ การดึงข้อมูลล่วงหน้าบนเวิลด์ไวด์เว็บ เป็นวิธีหนึ่งที่ถูกนำมาใช้ โดยมีหลักการดังนี้คือ ในขณะที่ผู้ใช้งานกำลังอ่านเอกสารของเวิลด์ไวด์เว็บอยู่ ระบบก็จะทำการคาดเดา(Predict) ว่าเอกสารที่ผู้ใช้งานจะร้องขอต่อไปคืออะไร แล้วทำการดึงเอกสารล่วงหน้า(Prefetch) มาเก็บไว้ในแคช(cache) เมื่อผู้ใช้งานทำการร้องขอเอกสารตัวต่อไปที่ระบบได้นำมาเก็บล่วงหน้าไว้แล้ว ผู้ใช้งานก็จะสามารถนำเอกสารจากแคชไปใช้ได้ทันที โดยไม่ต้องเสียเวลารอข้อมูลจากอินเตอร์เน็ตอีก

2. การประเมินประสิทธิภาพของการทำพรีเฟทช์

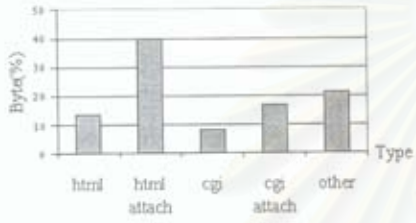
เราได้วัดประสิทธิภาพโดยรวมของการทำพรีเฟทชิ่งในสภาพแวดล้อมในประเทศไทยเพื่อที่จะประเมินถึงผลกระทบของระบบเมื่อนำการทำพรีเฟทช์มาใช้งาน โดยวิเคราะห์เพิ่มบันทึกการใช้งาน(access log) ของพร็อกซีแคช จากสำนักเทคโนโลยีสารสนเทศ ของจุฬาลงกรณ์มหาวิทยาลัย ในช่วงวันที่ 19 - 21 มกราคม ปี 2543 ซึ่งมี ปริมาณคำร้องขอ(request) 5,984,578 ครั้ง และจำนวนผู้ใช้งาน 5,497 คน โดยได้ตัดการร้องขอที่มาจากพร็อกซีภายในองค์กรออก

จากการวิเคราะห์เราจะเห็นถึงอัตราส่วนของคำร้องขอแต่ละแบบซึ่งมี ไฟล์เอกสาร(html), ไฟล์ที่ถูกนำมาใช้ในเอกสาร(html attach), เอกสารซีจีไอ(CGI) ซึ่งเป็นเอกสารที่ถูกสร้างโดยโปรแกรมซีจีไอที่เว็บเซิร์ฟเวอร์, ไฟล์ที่ติดมากับเอกสารซีจีไอ(cgi attach) และไฟล์อื่นๆ (other) เช่น ไฟล์โปรแกรมหรือไฟล์ข้อมูลต่างๆ โดยรูป 1 เป็นกราฟที่แสดงถึงสัดส่วนของจำนวน คำร้อง

แต่ละแบบ ส่วนรูปที่ 2 จะแสดงถึงสัดส่วนของปริมาณข้อมูล(byte) ของแต่ละแบบ



รูปที่ 1 :แสดงอัตราส่วนของคำร้องแต่ละประเภท



รูปที่ 2: แสดงอัตราส่วนของข้อมูลแต่ละประเภท

เนื่องจากเอกสารซีจีไอเป็นเอกสารที่จะถูกสร้างเมื่อผู้ใช้งานร้องขอซึ่งส่วนใหญ่จะต้องใช้ข้อมูลบางอย่างที่ผู้ร้องขอส่งมาด้วย เราจึงไม่สามารถทำนายได้ว่าผู้ใช้งานจะส่งข้อมูลอะไรไป ดังนั้นโปรแกรมพรีเฟทช์จะสามารถทำดึงเอกสารปกติ(html) และไฟล์ที่ติดมากับเอกสาร (html attach) เท่านั้น จากที่กล่าวมานี้จะเห็นได้ว่าสัดส่วนของไฟล์เอกสารที่สามารถพรีเฟทช์ได้จะมีประมาณ 58.98 % ของคำร้องขอทั้งหมด และเป็น 53.37 % ของปริมาณข้อมูลทั้งหมด ส่วนเอกสารอื่นๆ (other) จากกราฟที่ได้จะเห็นว่าเป็นไฟล์ที่มีขนาดใหญ่ ไม่เหมาะกับการพรีเฟทช์ และจากการวิเคราะห์นี้เราได้พบอีกว่าผู้ใช้งานใช้เวลาในการดูเอกสารแต่ละหน้าโดยเฉลี่ยประมาณ 70 วินาที แสดงว่าโปรแกรมพรีเฟทช์ที่จะนำมาใช้กับผู้ใช้งานกลุ่มนี้จะต้องจำกัดว่า ควรจะทำการพรีเฟทช์ภายใน 70 วินาทีหลังจากที่ผู้ใช้งานได้ร้องขอมา

จากข้อจำกัดในเรื่องประเภทของเอกสารที่ได้กล่าวมา เราได้จำลองการทำพรีเฟทช์ จากเพิ่มบันทึกการใช้งานตัวเดิม โดยโปรแกรมจำลองจะพรีเฟทช์เฉพาะไฟล์เอกสารปกติกับไฟล์ที่ติดมาด้วยเท่านั้น และเอกสารที่พรีเฟทช์ได้จะถือว่าเวลาที่ผู้ใช้งานใช้ในการรอเป็น 0 เพราะเอกสารถูกเก็บอยู่ในแคชแล้ว ซึ่งได้ผลดังนี้

เอกสารที่พรีเฟทช์ได้	605,964 หน้า (45.2%)
คำร้องที่พรีเฟทช์ได้	3,340,620 ครั้ง (55.8%)
ข้อมูลที่พรีเฟทช์ได้	23,314,537,304 ไบท์ (51.7%)
เวลาที่ลดลง	13,428,536,714 วินาที (40.5%)

ตารางที่ 1 แสดงถึงผลที่ได้จากจำลองการพรีเฟทช์

ตารางที่ 1 แสดงถึงประสิทธิภาพที่โปรแกรมพรีเฟทช์ แถวที่ 1 ถึง 3 จะแสดงถึงคำร้อง เอกสาร และปริมาณข้อมูล ที่โปรแกรมทำได้การร้องขอล่วงหน้าได้เปรียบเทียบกับปริมาณการร้องขอทั้งหมด ส่วนแถวที่ 4 บอกถึงเวลาโดยรวมที่โปรแกรมสามารถลดได้ จะเห็นได้ว่าในกรณีที่คาดเดาเอกสารถูกต้องทั้งหมดโปรแกรมพรีเฟทช์จะสามารถลดเวลาในการรอได้ถึง 40.52% และทำให้อัตราการพบเอกสาร(hit rate) เพิ่มขึ้นถึง 55.82 %

แต่อย่างไรก็ตามทดลองที่ได้กล่าวมาทั้งหมดนี้เป็นการจำลองในกรณีที่โปรแกรมพรีเฟทช์สามารถคาดเดาเอกสารได้ถูกต้อง 100% ซึ่งในความเป็นจริงประสิทธิภาพที่ได้จากโปรแกรมพรีเฟทช์ที่ใช้งานจริง จะน้อยกว่าผลลัพธ์ที่แสดงไว้ซึ่งจะขึ้นอยู่กับวิธีการคาดเดาเอกสารของแต่ละโปรแกรม และการทำเว็บพรีเฟทช์ จะมีข้อเสียในเรื่องของการสิ้นเปลืองแบนวิดท์ เพราะยิ่งขยายขอบเขตการพรีเฟทช์(Prefetching threshold) มากขึ้นแล้วโปรแกรมก็จะใช้แบนวิดท์มากขึ้นเช่นกัน และในกรณีที่โปรแกรมคาดเดาเอกสารผิดพลาดก็จะทำให้สิ้นเปลืองแบนวิดท์โดยเปล่าประโยชน์

สรุป

เราได้ทำการจำลองการทำงานของเว็บพรีเฟทช์ ในสภาพแวดล้อมของเครือข่ายในประเทศไทยเพื่อวัดศักยภาพของเว็บพรีเฟทช์ ซึ่งพบว่าพรีเฟทช์จะช่วยลดเวลารับเอกสาร ได้ถึง 40.52% และมีอัตราการพบเอกสารถึง 55.82 % แต่ในการทำพรีเฟทช์จะทำให้ต้องใช้ทรัพยากรของเครื่อง และแบนวิดท์เพิ่มขึ้นมาก ซึ่งปริมาณที่ใช้นี้จะขึ้นอยู่กับวิธีการคาดเดาแต่ละวิธี และขนาดของขอบเขตการพรีเฟทช์ที่กำหนดไว้ ซึ่งจะถูกใช้เป็นตัวแปรในการทดลองของงานวิจัยครั้งต่อไป

ประวัติผู้เขียนวิทยานิพนธ์

นายภรศิษฐ์ ธนวุฒิวัฒน์ เกิดเมื่อวันพุธที่ 8 ธันวาคม พุทธศักราช 2519 ที่กรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาตรีวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยี พระจอมเกล้าฯ เจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2541 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2542

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย