การใช้แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮลสองระดับในการทำนายการร้องขอเว็บเพจ

AN EFFICIENT TWO-LEVEL THRESHOLDS PAIRWISE ALGORITHM FOR PREDICTING REQUEST TO WEB SERVER

วิศิษฏ์ พัฒน์ชู
WISIT PHATCHOO

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

ISBN 974-9709-72-1

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การใช้แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮลสองระดับในการทำนายการร้องขอเว็บเพจ

AN EFFICIENT TWO-LEVEL THRESHOLDS PAIRWISE ALGORITHM FOR PREDICTING REQUEST TO WEB SERVER

วิศิษฎ์ พัฒน์ชู

WISIT PHATCHOO

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

ISBN 974-9709-72-1

เลขหมู่.....
เลขทะเบียน 51616
วัน,เดือน,ปี 26 ก.ค. 2547

b.....
i.....

AN EFFICIENT TWO-LEVEL THRESHOLDS PAIRWISE ALGORITHM FOR PREDICTING REQUEST TO WEB SERVER.

WISIT PHATCHOO

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

ISBN 974-9709-72-1

COPYRIGHT 2004

SCHOOL OF GRADUATE STUDIEDS

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การใช้แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮลสองระดับในการทำนายการร้องขอเว็บเพจ

AN EFFICIENT TWO-LEVEL THRESHOLDS PAIRWISE ALGORITHM FOR PREDICTION REQUEST TO WEB SERVER

ชื่อนักศึกษา นายวิศิษฏ์ พัฒน์ชู

รหัสประจำตัว 43067004

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ดร.โชติพัชร์ ภรณวลัย

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.โชติพัชร์	ภรณวลัย	
รศ.ดร.รัตติกร	วรากูลศิริพันธุ์	
ผศ.ดร.จันทร์บูรณ์	สถิตวิริยวงศ์	
ผศ.ดร.วรพจน์	กรีสุระเดช	
ผศ.อัครินทร์	คุณกิตติ	

วัน/เดือน/ปี ที่สอบ 18 พฤษภาคม 2547 เวลา 11.00 น. เป็นต้นไปไป

สถานที่สอบ ณ ห้อง M 23 (ชั้นลอย) อาคารเรียนรวมและปฏิบัติการคณะเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัยรับรองแล้ว

(ผศ.ดร.จารุวัตร เจริญสุข)

คณบดีบัณฑิตวิทยาลัย

วันที่ 28 **เดือน** พฤษภาคม **พ.ศ.** 2547

หัวข้อวิทยานิพนธ์	การใช้แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮลสองระดับในการทำนายการร้องขอเว็บเพจ
นักศึกษา	นาย วิศิษฎ์ พัฒน์ชู
รหัสประจำตัว	43067004
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการสารสนเทศ
พ.ศ.	2547
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร. โชติพัชร์ ภรณวลัย

บทคัดย่อ

พรีเฟรชชิ่ง (Pre-fetching) เป็นเทคนิคที่ใช้ในการลดเวลาในการเข้าถึงเว็บเพจ (webpage) หรือรีซอร์ส (resource) ต่างๆบนเครือข่ายอินเตอร์เนต โดยจะทำการทำนายเว็บเพจหรือรีซอร์สที่ผู้ใช้จะเข้าถึงในอนาคต และจะทำการดึงเว็บเพจหรือรีซอร์สที่ทำนายมาเก็บไว้ที่พร็อกซีเซิร์ฟเวอร์ (Proxy server) หรือแคช (Cache) ก่อนที่ผู้ใช้จะเข้าถึง และแพร์ไวส์อัลกอริทึม (Pairwise Algorithm) ก็เป็นหนึ่งในพรีเฟรชชิ่งเทคนิคที่จะใช้ข้อมูลในการเข้าถึงเวบเพจหรือรีซอร์สต่างๆของผู้ใช้มาทำการสร้างเป็นฮิ้น (Hint) ที่จะใช้ในการบอกว่าเว็บเพจหรือรีซอร์สที่ผู้ใช้จะเข้าถึงในอนาคต เพื่อทำการดึงมาเก็บไว้ก่อน

ในวิทยานิพนธ์นี้นำเสนอเทคนิคที่ใช้ในการปรับปรุงประสทธิภาพของแพร์ไวส์อัลกอริทึม โดยเทคนิคที่นำเสนอนี้เรียกว่าแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ ซึ่งจะมีการรวมเอาฮิ้นที่จะจากข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สของผู้ใช้เฉพาะคนและจากข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สของผู้ใช้ทั้งหมด ซึ่งผลจากการนำฮิ้นทั้งสองมารวมกันจะทำให้ได้ฮิ้นที่มีอัตราความถูกต้อง (Precision) และอัตราการนำฮิ้นกลับมาใช้ (Recall) ที่สูงกว่าแพร์ไวส์อัลกอริทึมเดิม

Thesis Title	An Effcient Two-Level Thresholds Pairwise Algorithm for Predicting Request to Web Server.
Student	Mr. Wisit Phatchoo
Student ID.	43067004
Degree	Master of Science
Programme	Information Science
Year	2004
Thesis Advisor	Asst. Prof. Dr. Chotipat Pornavalai

ABSTRACT

Prefetching is the technique to reduce time or latency of users resource access on internet network. The prefetching will predict the webpages or resources that user will access in future and then it pull predicted webpages or resources and keep them on proxy server or cache. Pairwise algorithm is one of prefetching technique that use resource access information of clients to construct hint volumes. The hint volumes are predicted webpage or resource.

This thesis represents the extend technique that improve the efficiency of pairwise algorithm. Our extend technique, we call two-level thresholds pairwise algorithm. It will construct hint volume from the combination of hint volumes that use resource access of specific clients and hint volumes that use resource access information of all clients. After that, we will get more efficiency hint volumes from both resource access information. The performance of hint volumes from our extend technique has high normalized precision rate and high normalized recall rate than original pairwise algorithm.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ ที่ปรึกษา ผศ.ดร.โชติพัชร์ ภรณวัลย์ ที่ให้ความช่วยเหลือให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณ รศ.ดร.รัตติกร วรากูลศิริพันธุ์, ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์, ผศ.ดร. วรพจน์ กรีสุระเดช, ผศ. อัครินทร์ คุณกิตติ กรรมการสอบหัวข้อและโครงร่างวิทยานิพนธ์ที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะจนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบคุณเพื่อนๆ และน้อง ๆ ของข้าพเจ้าที่ได้ช่วยเหลือในการหาข้อมูลเกี่ยวกับล็อกไฟล์และเนื้อหาต่างๆ

สุดท้ายต้องขอขอบคุณพ่อและแม่ของข้าพเจ้า นายวิรัตน์และนางสาลี่ พัฒน์ชู ที่เป็นกำลังให้ตลอดเวลาที่ทำวิทยานิพนธ์ฉบับนี้

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดาซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ..... II

กิตติกรรมประกาศ..... III

สารบัญ..... IV

สารบัญรูป..... VI

สารบัญตาราง..... XIII

บทที่ 1 บทนำ..... 1

1.1 ที่มาและเหตุจูงใจในการทำวิจัย..... 1

1.2 อธิบายถึงปัญหาที่กำลังศึกษาอยู่ของวิทยานิพนธ์..... 1

1.3 ทำไมปัญหานี้ถึงน่าสนใจ..... 1

1.4 หลักการใหม่อะไรที่นำเสนอ..... 2

1.5 เปรียบเทียบกับหลักการที่มีอยู่เดิม..... 2

1.6 รายละเอียดในวิทยานิพนธ์..... 3

บทที่ 2 ความรู้พื้นฐานของการเข้าถึงเว็บเพจบนเครือข่ายอินเตอร์เนต..... 4

2.1 องค์ประกอบของอินเตอร์เนต..... 4

2.2 องค์ประกอบพื้นฐานของเวิลด์ไวด์เว็บ..... 10

2.3 ขั้นตอนการเข้าถึงเว็บเพจ..... 22

บทที่ 3 ระบบการทำงานของพรีเฟรชชิ่งและแพร์ไวส์อัลกอริทึม..... 47

3.1 กล่าวนำ..... 47

3.2 รูปแบบการทำงานแบบต่างๆของการพรีเฟรชชิ่ง 47

3.3 รูปแบบต่างๆ ของล็อกไฟล์..... 50

3.4 รูปแบบต่างๆของระบบพรีเฟรชชิ่ง 54

บทที่ 4 ทฤษฎีและการทำงานของแพร์ไวส์อัลกอริทึม..... 57

4.1 ทฤษฎีของแพร์ไวส์อัลกอริทึม..... 57

4.2 การทำงานของแพร์ไวส์อัลกอริทึม..... 58

4.3 ประสิทธิภาพของแพร์ไวส์อัลกอริทึม..... 62

4.4 วิเคราะห์ประสิทธิภาพของแพร์ไวส์อัลกอริทึม..... 67

บทที่ 5 การทดลองและวิเคราะห์หาประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ..... 70
5.1 ทฤษฎีของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ..... 70
5.2 การวัดประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ..... 75
5.3 เปรียบเทียบประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้โทรสโฮล 2 ระดับ กับแพร์ไวส์อัลกอริทึมแบบเดิม..... 76
5.4 การปรับปรุงแพร์ไวส์อัลกอริทึมแบบใช้โทรสโฮล 2 ระดับ..... 84

บทที่ 6 บทสรุปและวิจารณ์ผลการทดลอง..... 100
6.1 สรุปและวิเคราะห์ผลการทดลอง 100
6.2 ทรัพยากรที่ใช้ในการทำงานของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ ... 101
6.3 การเลือกใช้เทรสโฮลในส่วนการทำงานแต่ละส่วน 103
6.4 ข้อเสนอแนะ 104
6.5 การนำแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลไปใช้งานจริง 104

เอกสารอ้างอิง..... 106

ภาคผนวก ก 107

ภาคผนวก ข 108

ประวัติผู้เขียน.....116

สารบัญรูป

รูปที่		หน้า
2.1	แสดงลักษณะทำงานของโปรโตคอลของมนุษย์เมื่อเปรียบเทียบกับโปรโตคอลบนคอมพิวเตอร์	4
2.2	แสดงลำดับชั้นต่างๆของอินเตอร์เน็ตโปรโตคอล	5
2.3	แสดงกระบวนการร้องขอและการส่งกลับของเว็บเพจระหว่างเว็บไคลเอนด์และเว็บเซิร์ฟเวอร์	11
2.4	ลักษณะทั่วไปของข้อมูลการร้องขอแบบ HTTP (HTTP Request Message)	14
2.5	โครงสร้างของ HTTP Request Message	15
2.6	ลักษณะทั่วไปของการตอบกลับแบบ HTTP HTTP Response Message	16
2.7	โครงสร้างของข้อมูลการตอบกลับแบบ HTTP (HTTP response message)	17
2.8	ตัวอย่างของข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) ทั่วๆไป	20
2.9	ตัวอย่างของข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) ทั่วๆไป	21
2.10	ข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) ที่มีการใช้ CONDITIONAL GET	21
2.11	ข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) ที่มีตอบกลับด้วย CONDITIONAL GET	22
2.12	โครงสร้างในการเข้าถึงเว็บเพจในเวิลด์ไวด์เว็บ	23
2.13	ตัวอย่างที่พร็อกซี่มีการทำงานเป็นประตูสู่เครือข่ายภายนอก (Gateway) ไปยัง FTP, HTTP และ Mail	30
2.14	ขั้นตอนที่เกิดขึ้นในกระบวนการร้องขอและตอบกลับแบบ HTTP ที่มีการใช้งานเว็บพร็อกซี่	31
2.15	แสดงรูปแบบการทำงานอย่างง่ายๆ เป็นลำดับของพร็อกซี่	34
2.16	แสดงตัวอย่างของการส่งตัวแปร (Parameter) ผ่านแท็ก <FORM> ในเว็บเพจ	40
2.17	รูปแบบการ Hosting แบบต่างๆ	45
3.1	แสดงตัวอย่างของล็อกไฟล์	52
4.1	รูปแบบการทำนายของแพร์ไวส์อัลกอริทึม	57
4.2	ขั้นตอนอ่านและประมวลผลเรคคอร์ดในล็อกไฟล์	59
4.3	สรุปขั้นตอนของแพร์ไวส์อัลกอริทึม	59
4.4	ขั้นตอนการนับของแพร์ไวส์อัลกอริทึม	61

สารบัญรูป(ต่อ)

รูปที่		หน้า
4.5	ขั้นตอนการหาความน่าจะเป็นของการขึ้นต่อกันของแพร์ไวส์อัลกอริทึม	60
4.6	ขั้นตอนการสร้างฮิ้นโวลลุ่มของแพร์ไวส์อัลกอริทึม	61
4.7	อัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30	64
4.8	อัตรานอมัลไลซ์พรีซีชั่นของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30	65
4.9	ขนาดเฉลี่ยของฮิ้นของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30	66
4.10	อัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30 ของผู้ใช้ที่มีการร้องขอ 3 อันดับแรก	68
4.11	อัตราพรีซีชั่นของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30 ของผู้ใช้ที่มีการร้องขอ 3 อันดับแรก	69
5.1	ขั้นตอนทั้งหมดของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮล	71
5.2	การรวมฮิ้นของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮล	72
5.3	แสดงอัตรานอมัลไลซ์รีคอลที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 203.107.128.249	77
5.4	แสดงอัตรานอมัลไลซ์รีคอลที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 203.148.232.254	77
5.5	แสดงอัตรานอมัลไลซ์รีคอลที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.44.10.80	78
5.6	แสดงอัตรานอมัลไลซ์พรีซิชั่นจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 203.107.128.249	79
5.7	แสดงอัตรานอมัลไลซ์พรีซิชั่นที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.28.22.254	79
5.8	แสดงอัตรานอมัลไลซ์พรีซิชั่นที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.44.10.80	80
5.9	แสดงขนาดฮิ้นเฉลี่ยจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 203.107.128.249	81

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.10 แสดงขนาดฮิ้นเฉลี่ยจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.28.22.254 81

5.11 แสดงขนาดฮิ้นเฉลี่ยจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.44.10.80 82

5.12 แสดงอัตรานอมัลไลซ์รีคอลเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮล ของผู้ใช้ทั้งหมด 82

5.13 แสดงอัตรานอมัลไลซ์พริซิชั่นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮล ของผู้ใช้ทั้งหมด 83

5.14 แสดงอัตราขนาดของฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮล ของผู้ใช้ทั้งหมด 83

5.15 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$ 85

5.16 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$ 85

5.17 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$ 86

5.18 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$ 86

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.19 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$ 87

5.20 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$ 87

5.21 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$ 88

5.22 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$ 88

5.23 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$ 89

5.24 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$ 89

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.25 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$ 90

5.26 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้งและมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$ 90

5.27 แสดงกระจายตัวของจำนวนครั้งของการร้องขอที่มาจากไคลเอนท์ของล็อกไฟล์ฝั่งเซิร์ฟเวอร์.....91

5.28 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$ 92

5.29 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$ 92

5.30 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$ 93

5.31 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$ 93

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.32 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$ 94

5.33 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$ 94

5.34 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$ 95

5.35 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$ 95

5.36 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$ 96

5.37 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$ 96

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.38 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$ 97

5.39 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้งและมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$ 97

5.40 แสดงกระจายตัวของจำนวนครั้งของการร้องขอที่มาจากไคลเอนท์.....98

6.1 แสดงเวลาการทำงานที่ใช้ของแพร์ไวส์อัลกอริทึม, แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงของข้อมูลที่ได้จากทางด้านเซิร์ฟเวอร์ 102

6.2 แสดงเวลาการทำงานที่ใช้ของแพร์ไวส์อัลกอริทึม, แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงของข้อมูลที่ได้จากทางด้านไคลเอนต์ 102

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดง MIME Type ต่างๆ 25
3.1 แสดงข้อมูลทั่วไปของที่มีอยู่ในล็อกไฟล์ 51
3.2 แสดงข้อมูลเพิ่มเติมที่อาจมีในล็อกไฟล์ 53
4.1 ตัวอย่างของแพร์ไวส์อัลกอริทึม 61
5.1 ตัวอย่างของแพร์ไวส์อัลกอริทึมที่มีการใช้เทรสโฮล 2 ระดับ 73

บทที่ 1
บทนำ

1.1 ที่มาเหตุจูงใจในการทำวิจัย

ความแพร่หลายของอินเตอร์เน็ตและเวิลด์ไวด์เว็บมีการเติบโตอย่างมาก เวิลด์ไวด์เว็บเป็นแหล่งข้อมูลขนาดมหาศาล ผู้คนสามารถหาข้อมูลได้เกือบทุกประเภทบนเวิลด์ไวด์เว็บไม่ว่าจะเป็นข้อมูลทางด้านการแพทย์ การสื่อสาร ข้อมูลสัตว์ป่า ข้อมูลข่าว ฯลฯ มีเว็บไซต์หลายร้อยเว็บไซต์เกิดขึ้นทุกวัน เว็บไซต์ชื่อดังต่างๆ สามารถดึงดูดให้มีผู้คนเข้ามาดูได้เป็นจำนวนมาก ซึ่งมีผลทำให้เกิดปริมาณข้อมูลที่อยู่บนเครือข่ายอินเตอร์เน็ตมีปริมาณที่สูงมากอีกทั้งจำนวนผู้ใช้เครือข่ายอินเตอร์เน็ตก็เพิ่มสูงขึ้นกว่าเมื่อก่อน ด้วยปัจจัยข้างต้นที่ได้กล่าวมาแล้ว เป็นผลให้การเข้าถึงเว็บไซต์ต่างๆในปัจจุบันใช้เวลานานขึ้นกว่าเดิม แม้จะมีการพัฒนาและเพิ่มประสิทธิภาพของอินเตอร์เน็ตให้ดีขึ้นไม่ว่าจะเป็นการเพิ่มความเร็วของเครือข่าย การเพิ่มประสิทธิภาพของเซิรฟ์เวอร์ผู้ให้บริการ แต่ก็ยังไม่เพียงพอกับการเติบโตที่รวดเร็วของเวิลด์ไวด์เว็บ

1.2 อธิบายถึงปัญหาที่กำลังศึกษาและวิจัย

แพร์ไวส์อัลกอริทึมเป็นหนึ่งในวิธีการพรีเฟรชชิ่งที่ใช้ในการช่วยลดเวลาในการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆ บนเครือข่ายอินเตอร์เน็ต โดยจะอาศัยข้อมูลในการเข้าถึงของผู้ใช้ในอดีตเพื่อทำนายเว็บเพจหรือรีซอร์สถัดไปในช่วงเวลาหนึ่งจากเว็บเพจหรือรีซอร์สปัจจุบันที่ผู้ใช้เข้าถึงอยู่ ซึ่งวิธีการแพร์ไวส์อัลกอริทึมนี้จะมีอัตราความถูกต้อง (precision) และอัตราการนำฮิ้นกลับมาใช้ (recall) ยังไม่ดีเท่าที่ควร ซึ่งทำให้การเกิดการเสียประโยชน์และช่องสัญญาณในการพรีเฟรชชิ่ง วิทยานิพนธ์ฉบับนี้จึงได้ทำการวิเคราะห์วิธีการแพร์ไวส์อัลกอริทึมที่มีการใช้งานอยู่ และนำเสนอเทคนิคเพิ่มเติมที่นำไปใช้ในการช่วยทำให้วิธีการแพร์ไวส์อัลกอริทึมมีความถูกต้องและมีประสิทธิภาพเพิ่มมากขึ้น ซึ่งได้ทำการวิเคราะห์และผลลัพธ์ที่แสดงถึงความถูกต้องและประสิทธิภาพที่เพิ่มขึ้นเมื่อเปรียบเทียบกับวิธีการแพร์ไวส์อัลกอริทึมเดิมกับวิธีการแพร์ไวส์อัลกอริทึมที่ถูกปรับปรุงด้วยการนำเอาเทคนิคที่วิทยานิพนธ์นี้เสนอ

1.3 ทำไมปัญหานี้ถึงน่าสนใจ

เมื่อไม่กี่ปีที่ผ่านมาได้มีการนำเสนอวิธีการหลายวิธีในการช่วยลดเวลาในการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆบนเครือข่ายอินเตอร์เน็ต เช่น พร็อกซี-แคช (proxy-cache) ที่มีการเก็บเอาเว็บเพจหรือรีซอร์สที่ผู้ใช้คนหนึ่งเคยดูแล้วไว้เมื่อผู้ใช้คนอื่นต้องการเข้าถึงหรือดูเว็บเพจเดียวกันนี้ก็สามารถนำจาก พร็อกซี-แคชได้เลยไม่ต้องมีการดึงจากเซิรฟ์เวอร์ต้นทางใหม่ ว้อดเน็ซ(What-

next) ที่มีการใช้เอ็นแกรม (n-gram) อัลกอริทึมในการทำนายหาเว็บเพจหรือรีซอร์สที่ผู้ใช้ต้องการเข้าถึงล่วงหน้า พรีเฟรชชิ่ง (Pre-fetching) จะทำการทำนายและดึงเอาเว็บเพจหรือรีซอร์สที่ผู้ใช้จะเข้าถึงในอนาคตมาเก็บไว้ที่พร็อกซีหรือแคชก่อนเพื่อที่เมื่อผู้ใช้ต้องการเข้าถึงเว็บเพจหรือรีซอร์สนั้นก็สามารถเอาจากพร็อกซีหรือแคชได้เลย

สำหรับเทคนิคการพรีเฟรชชิ่ง (Pre-fetching) ได้มีการนำเสนอวิธีการปรับปรุงให้มีประสิทธิภาพมากขึ้น เช่นการนำเอาโครงข่ายประสาทเทียม (neural net) ที่จะใช้การทำงานแบบ context-specific แทนการใช้ url-specific ในการทำนายเว็บเพจหรือรีซอร์สที่จะถูกเข้าถึง Dns-Tcp connection prefetching ที่มีการเพิ่มกระบวนการ pre-resolving, pre-connecting และ pre-warming http ที่จะช่วยลดเวลาของพรีเฟรชชิ่ง การทำงานร่วมกันของ coherence-prefetching ที่จะใช้การทำงานร่วมกันระหว่าง cache coherence และ pre-fetching ในการปรับปรุงประสิทธิภาพของ pre-fetching และการควบคุมการพรีเฟรชชิ่งที่มีการเพิ่มกลไกในการควมคุมการพรีเฟรชชิ่ง

1.4 มีการนำเสนอหลักการใหม่อย่างไร

เทคนิคที่นำมาใช้ในวิทยานิพนธ์นี้คือ วิธีการแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ (Two-Level Thesholds Pairwise Algoritmh) ที่จะนำไปใช้ปรับปรุงประสิทธิภาพของแพร์ไวส์อัลกอริทึม โดยในแพร์ไวส์อัลกอริทึมจะมีการนำข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างในอดีตของผู้ใช้ทั้งหมดนำมาใช้เป็นฮิ้น (Hint) ที่ใช้บอกเว็บเพจหรือรีซอร์สใดที่จะถูกเข้าถึงในอนาคต แต่วิธีการแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับจะเพิ่มเติมจากแพร์ไวส์อัลกอริทึม โดยจะเพิ่มการเก็บข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆ ของผู้ใช้เป็นคนๆไป จากนั้นจะนำข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สของผู้ใช้ที่เก็บเป็นคนๆ นั้นนำมาสร้างเป็นฮิ้นที่เหมาะกับผู้ใช้แต่ละคน และก็จะนำฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมเดิมมาเพิ่มเข้าไปด้วย ดังนั้นฮิ้นที่ส่งไปให้ผู้ใช้ใช้ในการพรีเฟรชชิ่งจากประกอบด้วยฮิ้นที่สร้างจากข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆในอดีตของผู้ใช้เฉพาะคนนั้นๆกับฮิ้นที่เกิดจากข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สของผู้ใช้ทั้งหมด ซึ่งผลจากการรวมฮิ้นจากสร้างจากข้อมูลทั้งสองแหล่งทำให้ได้ฮิ้นที่มีอัตราความถูกต้อง(precision) และการนำฮิ้นกลับมาใช้ (recall) ที่สูงขึ้น

1.5 เปรียบเทียบกับหลักการเดิมที่มีอยู่

ประสิทธิภาพของการพรีเฟรชชิ่งนั้นจากวัดในรูปของอัตราความถูกต้อง (precision), อัตราการนำฮิ้นกลับมาใช้ใหม่ (recall), ขนาดของฮิ้นเฉลี่ย (average hintsize) ซึ่งวิธีการแพรไวส์อัลกอริทึมเดิมเมื่อมีการใช้ค่าพรอบพาบริติ้เทรสโฮลล์ (Probabilities Threshold , p_t) ที่สูงแล้วอัตราการนำฮิ้นกลับมาใช้จะเริ่มลดลงอย่างรวดเร็วในขณะที่อัตราความถูกต้องยังคงเพิ่มขึ้นเล็กน้อย

จนถึงค่าพรอบพาบริตี้เทรสโฮลด์ค่าหนึ่งจากนั้นอัตราความถูกต้องจะเริ่มลดลงในขณะที่อัตราการนำฮิ้นกลับมาใช้จะต่ำมาก ซึ่งในวิทยานพนธ์นี้ที่การนำวิธีการแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับมาปรับปรุงประสิทธิภาพของแพร์ไวส์อัลกอริทึม ทำให้ได้ฮิ้นที่มีประสิทธิภาพมากขึ้น มีอัตราความถูกต้องและอัตราการนำฮิ้นกลับมาใช้ที่สูงขึ้นเมื่อเพิ่มค่าพรอบพาบริตี้เทรสโฮลด์ และเมื่อค่าพรอบพาบริตี้เทรสโฮลด์สูงๆ อัตราความถูกต้องและอัตราการนำฮิ้นกลับมาใช้ก็ยังคงอยู่ในระดับสูงอยู่ ในขณะที่ขนาดของฮิ้นเฉลี่ยที่สูงขึ้นจากวิธีการแพร์ไวส์เดิมเล็กน้อย

1.6 รายละเอียดของวิทยานิพนธ์

ในวิทยานิพนธ์นี้ประกอบด้วย 6 บทด้วยกัน กล่าวคือ

บทที่ 1 จะกล่าวถึงเหตุจูงใจในการทำวิจัยนี้ วัตถุประสงค์ของการทำวิจัย ตลอดถึงปัญหาที่น่าสนใจที่ศึกษาในวิทยานิพนธ์นี้ ขอบเขตของการทำวิจัย นอกจากนี้ยังเสนอหลักการใหม่เมื่อเปรียบเทียบกับหลักการเดิม

บทที่ 2 จะกล่าวถึงความรู้พื้นฐานของการเข้าถึงเว็บเพจบนเครือข่ายอินเตอร์เน็ต ซึ่งจะอธิบายองค์ประกอบต่างของการเข้าถึงเวบเพจในปัจจุบัน การทำงานที่จะที่เกิดขึ้นเมื่อมีการร้องขอหรือเข้าถึงเว็บเพจ

บทที่ 3 จะกล่าวถึงระบบการทำงานของการพรีเฟรชชิ่ง องค์ประกอบ รูปแบบการทำงานที่มีการใช้หรือวิจัยอยู่ในปัจจุบัน ข้อดีและข้อด้อยของการพรีเฟรชชิ่ง

บทที่ 4 ในบทนี้จะกล่าวถึงวิธีการแพร์ไวส์อัลกอริทึมที่เป็นหนึ่งในวิธีการที่ใช้ในการพรีเฟรชชิ่ง องค์ประกอบและการทำงานของแพร์ไวส์อัลกอริทึม และจะกล่าวถึงแพร์ไวส์อัลกอริทึมที่มีการใช้เทรสโฮล 2 ระดับที่เป็นแพร์ไวส์แบบปรับปรุงให้มีประสิทธิภาพที่ได้ดีกว่าแพร์ไวส์อัลกอริทึมเดิม

บทที่ 5 จะกล่าวถึงวิธีการทดลองหาประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบเดิมและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ และทำการวิเคราะห์ผลการทดลองของวิธีทั้งสอง

บทที่ 6 ในบทนี้จะทำการสรุปผลและวิจารณ์การทดลองที่เกิดขึ้นรวมถึงข้อเสนอแนะและปัญหาที่เกิดขึ้นในการทดลอง

บทที่ 2
ความรู้พื้นฐานของการเข้าถึงเว็บเพจบนเครือข่ายอินเตอร์เน็ต

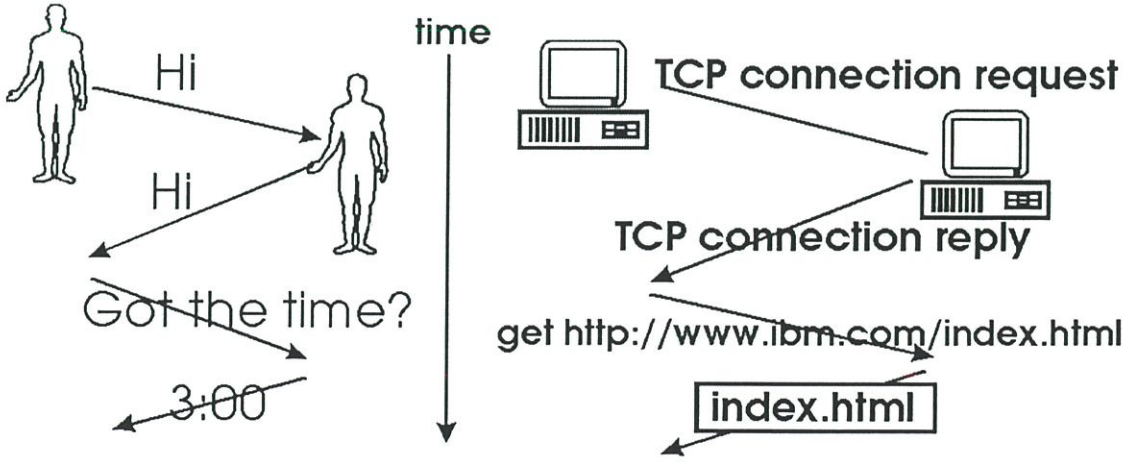
เวิลด์ไวด์เว็บเกิดขึ้นครั้งแรกในปี 1989 โดยการนำเสนอของ ทิม เบอร์เนอร์ลี (Tim Berner-Lee) และต่อมาก็กลายเป็นแหล่งข้อมูลขนาดมหาศาลบนเครือข่ายคอมพิวเตอร์ การทำงานที่มีส่วนการติดต่อกับผู้ใช้งานแบบกราฟฟิกทำให้ผู้ใช้งานสามารถที่จะดูหรือท่องไปตามกลุ่มของเว็บเพจด้วยการคลิกที่ลิงก์ที่อยู่ในเว็บเพจโดยที่ผู้ใช้ไม่จำเป็นต้องสนใจรูปแบบหรือตำแหน่งของเว็บเพจเลย และในเวิลด์ไวด์เว็บนั้นผู้ใช้งานไม่เพียงแต่จะสามารถดูเว็บเพจได้เท่านั้น ยังสามารถที่จะค้นหาข้อมูล รับและส่งอิเลคทรอนิกส์เมล์ และสามารถที่จะทำธุรกรรมโดยผ่านทางเวิลด์ไวด์เว็บได้

เวิลด์ไวด์เว็บนั้นทำงานบนเครือข่ายอินเตอร์เน็ต โดยใช้โปรโตคอล HTTP ในการสื่อสารระหว่างเว็บบราวเซอร์และเว็บเซิร์ฟเวอร์ และข้อมูลของเว็บเพจนั้นจะอยู่ในรูปของภาษา HTML ที่จะใช้ในการบอกว่าให้เว็บบราวเซอร์ให้แสดงเว็บเพจอย่างไร ในบทนี้จะอธิบายถึงการทำงานของอินเตอร์เน็ต องค์ประกอบต่างๆ และการทำงานของเวิลด์ไวด์เว็บ และองค์ประกอบต่างของเวิลด์ไวด์เว็บ

2.1 องค์ประกอบของอินเตอร์เน็ต

2.1.1 โปรโตตอล

โดยเริ่มต้นจากการอธิบายโปรโตคอลของมนุษย์ก่อนโดยพิจารณาจากรูปที่ 2.1



รูปที่ 2.1 แสดงลักษณะทำงานของโปรโตคอลของมนุษย์เมื่อเปรียบเทียบกับโปรโตคอลบนคอมพิวเตอร์

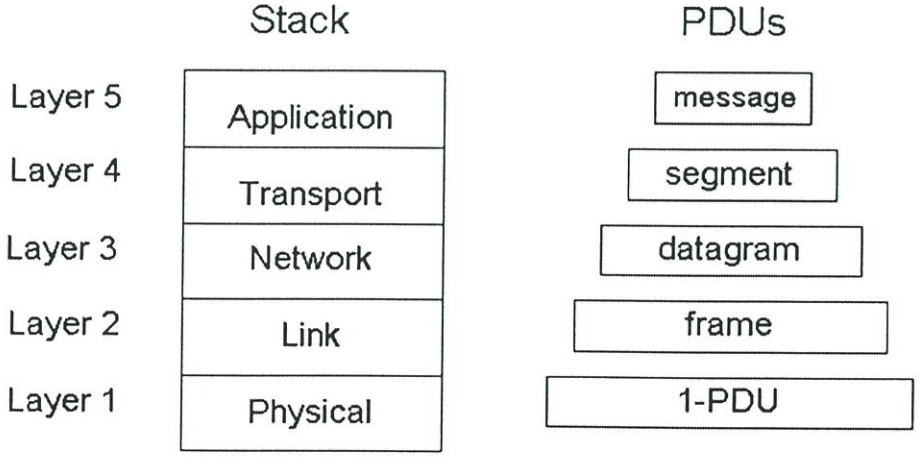
จากรูปที่ 2.1 นั้นเริ่มต้นคนทั่วไปจะเริ่มทักทายด้วยการ "สวัสดี" จากนั้นคนที่สองก็จาก "สวัสดี" ตอบ การตอบของคนที่สองเป็นการบอกเป็นนัยว่าคนที่สองนั้นได้รับรู้การ "สวัสดี" ของคนแรกแล้ว จากนั้นคนแรกก็ถามว่า "ตอนนี้เวลาอะไร" ไปยังคนที่สอง คนที่สองก็ตอบกลับไปว่ า "ตอนนี้เวลา 15.00" ซึ่งในความเป็นจริงนั้นคนที่สองอาจจะไม่ตอบอะไรเลยก็ได้ หรืออาจจะตอบอย่างอื่นกลับมาก็ได้ ดังนั้นสำหรับโปรโตคอลของมนุษย์นั้นคือ "ข้อความอย่างใดอย่างหนึ่งที่มีการส่งเมื่อเกิดกระทำอย่างใดอย่างหนึ่งเกิดขึ้น และการตอบรับอย่างใดอย่างหนึ่งที่รองรับข้อความหรือการกระทำที่มีการส่งมา"

และสำหรับโปรโตคอลของระบบเครือข่ายนั้นจะคล้ายกับโปรโตคอลของมนุษย์ยกเว้นเพียงข้อความที่แลกเปลี่ยนกันนั้นจะถูกทำโดยฮาร์ดแวร์หรือซอฟแวร์แทน การติดต่อสื่อสารต่างๆอาจจะเกี่ยวข้องกับสองส่วนประกอบของระบบเครือข่ายหรือมากกว่านั้นก็ได้ ดังนั้นนิยามของโปรโตคอลของระบบเครือข่ายคือ

"รูปแบบและลำดับของข้อความหรือการกระทำที่มีการแลกเปลี่ยนระหว่างสองหรือมากกว่าส่วนประกอบที่มีการติดต่อกัน"

2.1.2 อินเตอร์เน็ตโปรโตคอล

อินเตอร์เน็ตโปรโตคอลประกอบด้วย 5 เลเยอร์ (layers) นั่นคือ ฟิสิกส์คอลเลเยอร์, เดต้าลิงก์เลเยอร์, เน็ตเวิร์คเลเยอร์, ทรานสปอตเลเยอร์, แอพลิเคชั่นเลเยอร์ การแลกเปลี่ยนข้อมูลระหว่างเลเยอร์จะอยู่ในรูปของ PDU-n โดยจะมีชื่อเรียกเฉพาะสำหรับ 4 เลเยอร์บน นั่นคือ Frame, Datagram, Segment, Message สำหรับในส่วนของฟิสิกส์คอลเลเยอร์นั้นจะไม่มีชื่อเรียกเฉพาะ เพราะข้อมูลจะถูกส่งไปเป็นบิทๆ(bit) ลักษณะของข้อมูลในชั้นเลเยอร์ต่างเป็นดังรูปที่ 2.2



รูปที่ 2.2 แสดงลำดับชั้นต่างๆของอินเตอร์เน็ตโปรโตคอล

โปรโตคอลเลเยอร์ต่างๆ จะถูกใส่เข้าไปให้ทำงานที่ทั้งซอฟแวร์, ฮาร์ดแวร์ หรือทำงานในส่วนทั้งฮาร์ดแวร์และซอฟแวร์ก็ได้ ตัวอย่างแอพลิเคชั่นเลเยอร์เช่น HTTP ซึ่งจะทำงานในซอฟแวร์ที่เครื่องคอมพิวเตอร์ที่เรียกว่าเว็บไคลเอนต์ (Web Client) หรือเครื่องเซิร์ฟเวอร์ต่างๆที่เรียกว่าเว็บเซิร์ฟเวอร์ และเช่นเดียวกับทรานสปอตเลเยอร์ แต่สำหรับเดต้าลิงก์เลเยอร์และฟิสิกส์คอลเลเยอร์นั้นใช้ในการติดต่อสื่อสารระหว่างลิงก์ใดๆ โดยทั้งสองเลเยอร์นั้นจะถูกใส่เข้าไปทำงานในเน็ตเวิร์กการ์ดเลย สำหรับเน็ตเวิร์กเลเยอร์จะทำงานในส่วนร่วมกันระหว่างซอฟแวร์และฮาร์ดแวร์

สำหรับหน้าที่ต่างๆของเลเยอร์แต่ละเลเยอร์ของอินเตอร์เน็ตโปรโตคอลนั้นสามารถสรุปได้ดังนี้

- แอพลิเคชั่นเลเยอร์ (Application Layer) : มีหน้าที่ในการให้การสนับสนุนเน็ตเวิร์คแอพลิเคชั่นต่างๆ โดยแอพลิเคชั่นเลเยอร์นั้นจะรวมทั้งโปรโตคอลต่างๆเช่น HTTP ที่มีหน้าที่ในด้านการเข้าถึงเว็บเพจ SMTP ที่มีหน้าที่ในการส่งอีเมล์ และ FTP ที่มีหน้าที่ในการรับส่งไฟลต่างๆ
- ทรานสปอตเลเยอร์ (Transport Layer) : มีหน้าที่ในการส่งข้อมูลจากแอพลิเคชั่นเลเยอร์หรือแมสเสจ (Message) จากแอพลิเคชั่นเลเยอร์ระหว่างแอพลิเคชั่นของเครื่องไคลเอนต์และแอพลิเคชั่นของเซิร์ฟเวอร์ ในอินเตอร์เน็ตนั้นจะมี 2 ทรานสปอตโปรโตคอลนั้นก็คือ TCP และ UDP โดย TCP จะให้บริการแบบรับการประกันการถึงข้อมูล (Connection-Oriented) และ TCP จะทำการแบ่งข้อมูลจากแอพลิเคชั่นเลเยอร์ (Message) ออกเป็นเซกเมนต์ (Segment) ย่อยๆ ก่อนที่จะส่งข้อมูล สำหรับ UDP จะเป็นโปรโตคอลแบบไม่รับประกันการถึงของข้อมูล (Connectionless) ซึ่งจะกล่าวในรายละเอียดต่อไป การทำงานของทรานสปอตเลเยอร์มีส่วนสำคัญอย่างมากในการควบคุมการส่งข้อมูลให้ถึงเป้าหมาย ไม่ให้เกิดความผิดพลาด ไม่ให้มีการส่งข้อมูลที่ได้ส่งไปแล้วซ้ำ
- เน็ตเวิร์คเลเยอร์ (Network Layer) : มีหน้าที่ในการค้นหาเส้นทางที่จะใช้ในการส่งข้อมูลของทรานสปอตเลเยอร์หรือเดต้าแกรม (Datagram) จากเครื่องโฮลต์หรือคอมพิวเตอร์เครื่องหนึ่งไปยังเครื่องโฮลต์อีกเครื่องหนึ่ง โดยส่วนประกอบหลักๆของเน็ตเวิร์กเลเยอร์มี 2 ส่วนคือ ส่วนแรกเป็นส่วนในการนิยามฟิลด์ต่างในเดต้าแกรม (Datagram) ที่จะใช้ในการจำแนกเครื่องคอมพิวเตอร์เครื่องต่างๆในเครือข่ายอินเตอร์เน็ต และส่วนที่สองคือส่วนที่จะใช้ในการค้นและรักษาข้อมูลที่จะใช้ในการพิจารณาว่าจะส่งข้อมูลไปทางใดเพื่อให้ถึงปลายทาง ซึ่งส่วนประกอบทั้งสองจะอยู่ในโปรโตคอลที่ใช้ในชั้นเน็ตเวิร์กเลเยอร์นั่นก็คือไอพี(IP) โปรโตคอล ในอินเตอร์เน็ตนั้นเครื่องทุกเครื่องและอุปกรณ์ทุกชนิดต่างต้องมีการทำงานของไอพีโปรโตคอลเพื่อให้ติดต่อสื่อสารได้
- เดต้าลิงก์เลเยอร์ (Datalink Layer) : สำหรับเน็ตเวิร์กเลเยอร์นั้นจะใช้ในการส่งข้อมูลไปตามเส้นทางที่จะประกอบอุปกรณ์ต่างๆในอินเตอร์เน็ตระหว่างเครื่องต้นทางและเครื่อง

ปลายทาง ซึ่งในการเคลื่อนย้ายหรือส่งข้อมูลในเครือข่ายอินเตอร์เน็ตนั้นจะต้องส่งข้อมูลไปเป็นฮอฟ (Hop) หรือเป็นลิงก์ไป ในแต่ละฮอฟ (hop) นั้นก็จะคือเครื่องคอมพิวเตอร์หรืออุปกรณ์เน็ตเวิร์คที่อยู่ระหว่างเครื่องต้นทางและเครื่องปลายทาง โดยระหว่างเครื่องต้นทางและเครื่องปลายทางนั้นก็จะประกอบด้วยหลายๆ ฮอฟหรือหลายๆลิงก์ ซึ่งเดต้าลิงก์เลเยอร์จะทำหน้าที่ในส่วนนี้ โดยเครื่องคอมพิวเตอร์หรืออุปกรณ์เน็ตเวิร์คในแต่ละฮอฟจะส่งเดต้าแกรมไปยังเดต้าลิงก์เลเยอร์ของเครื่องคอมพิวเตอร์หรืออุปกรณ์เน็ตเวิร์คนั้น จากนั้นเดต้าลิงก์เลเยอร์จะทำการเพิ่มส่วนหัว (Header) และส่วนหาง (Trailer) เข้าไปในเดต้าแกรมที่มาจากเน็ตเวิร์กเลเยอร์ซึ่งจะเรียกข้อมูลในส่วนนี้ว่าเป็นเฟรม (Frame) ก่อนส่งให้ฟิสิกส์คอลเลเยอร์ต่อไป

- ฟิสิกส์คอลเลเยอร์ (Physical Layer) : มีหน้าที่ในการส่งข้อมูลที่เป็นบิทๆจากเฟรมของข้อมูลที่ถูกส่งมาจากเดต้าลิงก์เลเยอร์จากฮอฟ (hop) หนึ่งไปยังอีกฮอฟหนึ่ง โดยสำหรับการทำงานของเลเยอร์นี้จะต้องเกี่ยวข้องกับสื่อหรือสายสัญญาณที่ใช้ด้วยเช่นถ้าใช้สายสัญญาณเป็นสายสัญญาณแบบเส้นไยแก้วนำแสง (Fiber Optic) ก็ต้องมีการแปลงข้อมูลที่เป็นบิทๆให้อยู่ในรูปของแสงก่อนจะส่งไปยังสายไยแก้วนำแสงนั้น

2.1.3 อินเตอร์เน็ต

ถ้าจะนิยามอินเตอร์เน็ต[5] นั้นก็สามารถนิยามได้ว่าเป็นระบบเครือข่ายของระบบเครือข่าย นั่นคืออินเตอร์เน็ตเป็นการเชื่อมต่อกันของระบบเครือข่ายต่างๆทั่วโลก ที่ประกอบด้วยเครือข่ายแบบปิดและเครือข่ายแบบเปิด ซึ่งทุกๆระบบเครือข่ายที่เชื่อมต่อกับอินเตอร์เน็ตจะต้องมีการทำงานบนโปรโตคอลไอพี

อินเตอร์เน็ตเกิดขึ้นมาเมื่อ 20 ปีก่อน ด้วยความพยายามของกองทัพสหรัฐอเมริกา (Department of defense) ในงานวิจัยทางการทหารเพื่อต่อต้านศัตรู โดยได้ทำการทดลองเครือข่ายคอมพิวเตอร์ซึ่งเรียกว่าอาร์ปาเนต (ARPAnet) เป็นการออกแบบตามความต้องการในขั้นต้น ที่จะใช้ในการรับ-ส่งข้อความระหว่างเครื่องคอมพิวเตอร์ โดยใช้ไอพี (IP) โปรโตคอล

จากอดีตที่คิดว่าไม่น่าจะเป็นไปได้แต่ก็ได้พิสูจน์แล้วว่าเป็นความจริง จากการติดต่อสื่อสารในการรับส่งข้อความระหว่างเครื่องคอมพิวเตอร์หนึ่งกับเครื่องคอมพิวเตอร์หนึ่ง กลายมาเป็นเครื่องคอมพิวเตอร์หนึ่งกับเครื่องคอมพิวเตอร์อื่นอีกหลายๆเครื่องได้ อินเตอร์เน็ตได้ถูกพัฒนาขึ้นในสหรัฐอเมริกา อังกฤษ และสแกนดิเนียเวีย ซึ่งภายหลังรัฐบาล, มหาวิทยาลัยและประชาชนทั่วไปได้ให้ความสนใจมากขึ้น มีการผลิตซอฟร์แวร์หรือแอพลิเคชั่นเพื่อรองรับการใช้งานบนเครือข่ายอินเตอร์เน็ตแตกต่างกันไป และทุกคนสามารถหาซื้อเครื่องคอมพิวเตอร์ที่สามารถสื่อสารกับเครื่องคอมพิวเตอร์อื่นบนเครือข่ายได้ตามความต้องการ

ในขณะเดียวกัน ประมาณพ.ศ.2526 ระบบเครือข่ายแบบท้องถิ่น (Local Area Network:LAN) ได้ถูกพัฒนาขึ้นอย่างรวดเร็วซึ่งสามารถใช้ได้บนเครื่องคอมพิวเตอร์เวิร์คสเตชั่นซึ่งนิยมใช้ระบบยูนิกซ์(Unix) ของ Berkeley ซึ่งในยูนิกซ์นี้ได้รวมโปรแกรมในส่วนการทำงานของไอพีที่มีรูปแบบการใช้งานใกล้เคียงกับระบบยูนิกซ์หน่วยงานอื่นๆ ทำให้สามารถทำงานร่วมกันได้ จึงได้มีการเริ่มสร้างระบบเครือข่ายของตนเองโดยเชื่อว่าถ้าติดต่อภายในระบบของตนเองได้ ก็ย่อมสามารถติดต่อกับเครือข่ายอื่นๆ ที่นอกเหนือภายในระบบตนเองได้เช่นกัน ระบบเครือข่ายที่สำคัญอีกเครือข่ายหนึ่งคือ NSFnet หรือมูลนิธิวิทยาศาสตร์แห่งชาติของสหรัฐอเมริกา (National Science Foundation) NSF ได้จัดตั้งศูนย์ซุปเปอร์คอมพิวเตอร์ (Supercomputer) 5 แห่งเพื่อสนับสนุนงานวิจัย ครั้งแรก NSF ได้พยายามที่จะใช้อาร์ปาเนต (ARPAnet) ในการติดต่อสื่อสารซึ่งกันและกัน แต่ไม่เป็นผลสำเร็จเพราะเป็นการทำงานของระบบราชการ ดังนั้น NSF จึงตัดสินใจสร้างระบบเครือข่ายของตนเองบนพื้นฐานของเครือข่ายอาร์ปาเนตที่ใช้โปรโตคอลทีซีพี-ไอพีซึ่งสามารถติดต่อศูนย์ต่างๆได้ด้วยความเร็ว 56 กิโลบิต/วินาที (56 Kbps) และได้พัฒนาโดยการขยายเครือข่ายออกสู่ภูมิภาคของประเทศ ในปี พ.ศ.2530 NSF ได้พัฒนาระบบเพื่อรองรับการใช้งานที่เพิ่มขึ้นความหวังที่สำคัญของการพัฒนาเครือข่าย NSF ก็คือเพื่อให้ทุกคนได้ใช้ระบบเครือข่ายได้ในระดับหนึ่งไม่จำกัดเฉพาะหน่วยงานของรัฐบาล หรือนักวิจัยของสถาบันเท่านั้น NSF ยังได้พัฒนาและสนับสนุนทางด้านการศึกษาโดยการให้ทุนแก่มหาวิทยาลัยในการติดตั้งระบบเครือข่าย เพื่อให้ทุกคนสามารถใช้ได้ทั่วถึงกัน การใช้อินเตอร์เน็ตได้แพร่หลายมากขึ้นเรื่อยๆ มีการพัฒนาปรับปรุงเทคโนโลยีต่างๆ ภายในระบบงานของแต่ละเครือข่ายของหน่วยงานต่างๆ โดยชื่อที่ใช้เรียกเครือข่ายมีการปรับเปลี่ยนไปเรื่อยตั้งแต่ชื่อ อาร์พาเนต (ARPAnet) เป็นเฟเดอรัลรีเสิร์ชอินเตอร์เน็ต (Federal Research Internet) บ้างหรือทีซีพี/ไอพี บ้างจนในที่สุดกลายเป็นที่รู้จักกันในปัจจุบันคือ อินเตอร์เน็ต (Internet)

ลักษณะเครือข่ายของอินเตอร์เน็ตจะอยู่ในรูปลำดับชั้นที่ไม่มีโครงสร้างที่แน่นอน แต่จะอยู่รูปล่างไปยังบน (Hierarchical) ที่ประกอบด้วยระบบที่เป็นปลายสุด(เครื่องคอมพิวเตอร์ที่ใช้งานตามบ้านหรือระบบเครือข่ายขององค์กรใดๆ) ที่จะเชื่อมต่อไปยังผู้ให้บริการอินเตอร์เน็ตท้องถิ่นจากนั้นผู้ให้บริการท้องถิ่นก็จะเชื่อมต่อไปยังผู้ให้บริการภูมิภาค และผู้ให้บริการส่วนภูมิภาคก็จะเชื่อมต่อไปยังผู้ให้บริการระดับประเทศและผู้ให้บริการระดับประเทศก็จะเชื่อมต่อกันเป็นระบบเครือข่ายขนาดมหาศาล

สำหรับรูปแบบของแอพลิเคชั่นที่ใช้งานบนเครือข่ายอินเตอร์เน็ตนั้นส่วนใหญ่จะมีการทำงานที่เรียกว่า “ไคลเอนต์ – เซิร์ฟเวอร์” โดยไคลเอนต์จะเป็นเครื่องที่ต้องการร้องขอการให้บริการจากเครื่องเซิร์ฟเวอร์ และเครื่องเซิร์ฟเวอร์เป็นเครื่องใช้ในการให้บริการกับเครื่องไคลเอนต์ โดยการทำงานของแอพลิเคชั่นบนเครือข่ายอินเตอร์เน็ตจะมีสองแบบคือการทำงานแบบรับประกันการถึง

ของข้อมูล (CONNECTION-ORIENTD) และการทำงานแบบไม่รับประกันการถึงของข้อมูล (CONNECTIONLESS)

2.1.4 รูปแบบการให้บริการของอินเตอร์เน็ต

การให้บริการบนอินเตอร์เน็ตมี 2 แบบคือ

2.1.4.1 การให้บริการแบบรับประกันการถึงของข้อมูล (CONNECTION-ORIENTED)

การให้บริการแบบรับประกันการถึงของข้อมูล (CONNECTION-ORIENTED) นั้นเป็นการติดต่อกับระหว่างเครื่องไคลเอนต์และเครื่องเซิร์ฟเวอร์จะต้องมีการสร้างการเชื่อมต่อระหว่างกันก่อนที่จะมีการส่งข้อมูลระหว่างกัน และอย่างที่กล่าวแล้วว่าอินเตอร์เน็ตจะใช้โปรโตคอลทีซีพี/ไอพี (TCP/IP) ในการทำงาน ดังนั้นจะเรียกระบวนการสร้างการเชื่อมต่อก่อนการส่งของข้อมูลนี้ว่า "การตกลงการเชื่อมต่อสามทาง (Three-way Handshake)"

โดยโปรโตคอลทีซีพี (TCP) ที่เป็นโปรโตคอลในส่วนทรานสปอตน์เลเยอร์ในการให้บริการแบบรับประกันการถึงของข้อมูล (CONNECTION-ORIENTED) และการให้บริการแบบนี้ยังสามารถให้บริการแบบอื่นเพิ่มเติมได้อีกเช่น

- ความถูกต้องของข้อมูล (Reliable Data Transfer) จะมีการให้บริการที่ทำให้แน่ใจได้ว่าข้อมูลที่มีการส่งระหว่างไคลเอนต์และเซิร์ฟเวอร์นั้นจะไม่มีการสูญหาย จะใช้กระบวนการที่เรียกว่า Acknowledgment และ Retransmission ในการให้บริการแบบนี้
- การควบคุมการส่งข้อมูล (Flow control) จะมีการให้บริการที่ใช้ในการควบคุมการส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ให้แต่ละฝั่งสามารถรับข้อมูลได้ทัน ป้องกันไม่ให้ฝั่งหนึ่งส่งข้อมูลเร็วกว่าอีกฝั่งที่จะรับได้ทัน
- การควบคุมความคับคั่งของข้อมูล (Congestion control) จะมีการให้บริการที่ใช้ในการควบคุมไม่ให้ระบบเครือข่ายเกิดความคับคั่งของข้อมูล เพราะเมื่อมีการเกิดความคับคั่งของข้อมูลแล้ว การควบคุมความคับคั่งของข้อมูลจะทำการลดความเร็วในการส่งลงเพื่อไม่ให้ไปเพิ่มความคับคั่งของระบบเครือข่ายอีก

2.1.4.2 การให้บริการแบบไม่รับประกันการถึงของข้อมูล (CONNECTIONLESS)

การให้บริการแบบไม่รับประกันการถึงของข้อมูล (CONNECTIONLESS) นั้นจะไม่มีการสร้างการเชื่อมต่อก่อนที่จะเริ่มส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ การส่งข้อมูลระหว่างกันนั้นไม่ว่าจะเป็นฝั่งไคลเอนต์หรือฝั่งเซิร์ฟเวอร์จะเพียงเมื่อต้องการจะส่งข้อมูลก็ทำการสร้างแมสเสจ (Message) ที่จะส่งแล้วก็ส่งออกไปเลย จะไม่มีกระบวนการตรวจสอบว่าข้อมูลที่ส่งไปนี้ทางอีกฝั่งได้รับหรือไม่ และไม่มีกระบวนการควบคุมการส่งใดๆเลย

การให้บริการแบบ CONNECTIONLESS นั้นจะใช้โปรโตคอลในชั้นทรานสปอตน์เลเยอร์คือ UDP

2.2 องค์ประกอบพื้นฐานของเวิลด์ไวด์เว็บ

2.2.1 UNIFORM RESOURCE IDENTIFIER (URI)

การเข้าถึงและการจัดการรีซอร์สต่างๆ ที่กระจายอยู่บนเวิลด์ไวด์เว็บจะต้องมีวิธีการที่ใช้ในการบ่งบอกหรือแสดงตำแหน่งของรีซอร์สนั้นๆ ซึ่งในเวิลด์ไวด์เว็บจะใช้ UNIFORM RESOURCE IDENTIFIER(URI) [6] ในการบ่งบอกหรือแสดงตำแหน่งของรีซอร์ส โดย URI จะทำงานคล้ายกับตัวชี้ตำแหน่งไปยังรีซอร์สนั้นๆ URI จะประกอบด้วย 3 ส่วน คือ โปรโตคอลที่ใช้ในการติดต่อกับเครื่องเซิร์ฟเวอร์หรือเว็บเซิร์ฟเวอร์, UNIFORM RESOURCE LOCALTOR (URL) คือชื่อของเครื่องเซิร์ฟเวอร์หรือเว็บเซิร์ฟเวอร์หรือตำแหน่งของรีซอร์สและ UNIFORM RESOURCE NAME (URN) คือชื่อของรีซอร์สนั้นๆ เช่น http://www.foo.com/coolpic.gif ในตัวอย่างนี้โปรโตคอลที่ใช้ในการติดต่อกับเครื่องแม่ข่าย (http), ชื่อของเครื่องเซิร์ฟเวอร์(www.foo.com), และชื่อของรีซอร์สนั้นๆบนเครื่องแม่ข่าย (coolpic.gif)

2.2.2 HYPERTEXT MARKUP LANGUAGE (HTML)

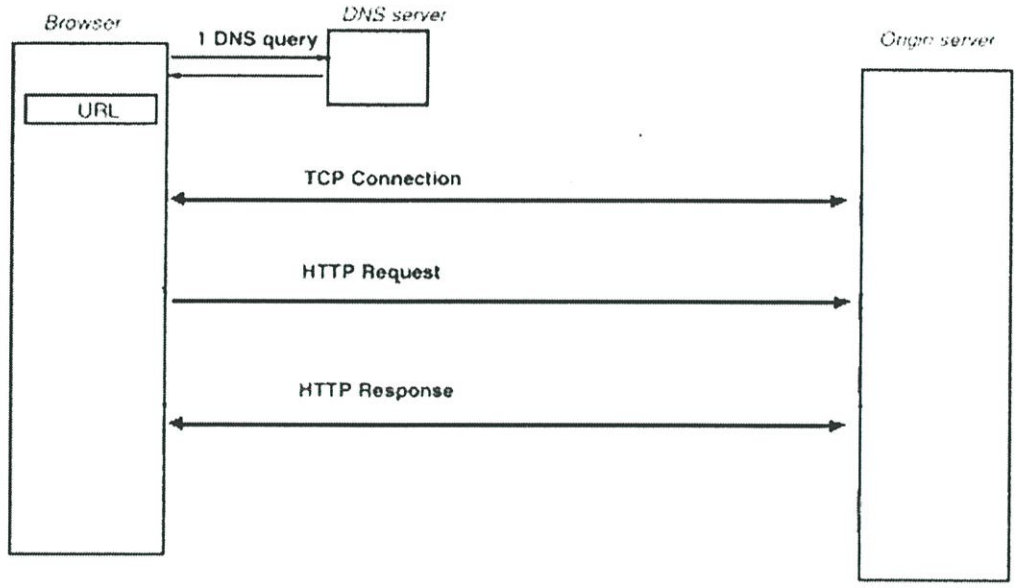
Hypertext Markup Language [7] คือภาษามาตรฐานในการแสดงตัวอักษรหรือรูปแบบของตัวอักษรบนเว็บเพจ ซึ่ง HTML เป็นภาษาที่ได้รับการสืบทอดจาก Standard Generalized Markup Langauage (SGML) โดย HTML สามารถให้ผู้สร้างเว็บเพจสามารถที่จะจัดรูปแบบของข้อความแสดงรูปภาพ การสร้างลิงก์ต่างๆบนเว็บเพจ โครงสร้างของ HTML เป็นรูปแบบง่ายๆในการทำความเข้าใจและการเรียนรู้ ปัจจุบันมีการพัฒนา HTML ไปถึงเวอร์ชั่นที่ 4 แล้ว

การทำงานของ HTML ก็คือเมื่อผู้ใช้ร้องขอเว็บเพจจากเครื่องเซิร์ฟเวอร์แล้วทางเครื่องเซิร์ฟเวอร์จะส่งเว็บเพจนั้นกลับมาให้ผู้ใช้ โดยเว็บเพจนั้นจะอยู่ในรูปของโครงสร้าง HTML เมื่อเครื่องของผู้ใช้ได้รับแล้วตัวโปรแกรมที่ทำหน้าที่ในการแปลความหมายและแสดงผลเว็บเพจหรือเว็บไคลเอนต์ก็คือ โปรแกรมเว็บบราวเซอร์ ก็จะทำงานเพื่อแสดงเว็บเพจนั้นๆออกมาตามโครงสร้าง HTML

2.2.3 HYPERTEXT TRANSFER PROTOCOL (HTTP)

Hypertext Transfer Protocol เป็นโปรโตคอลระดับแอพลิเคชั่นเลเยอร์ที่เป็นส่วนสำคัญมากในเครือข่ายเวิลด์ไวด์เว็บ โดย HTTP จะทำงานเป็น 2 ฝั่งคือฝั่งเครื่องไคลเอนต์และฝั่งเครื่องเซิร์ฟเวอร์ การติดต่อระหว่าง 2 ฝั่งจะใช้ HTTP ในการติดต่อสื่อสารกัน

HTTP จะนิยามว่าทางโปรแกรมฝั่งไคลเอนต์นั่นก็คือเว็บไคลเอนต์หรือเว็บบราวเซอร์จะร้องขอเว็บเพจจากเครื่องเซิร์ฟเวอร์อย่างไร และทางโปรแกรมฝั่งเครื่องเซิร์ฟเวอร์หรือเว็บเซิร์ฟเวอร์จะส่งเว็บเพจกลับไปอย่างไร การทำงานของการจะเป็นรูปแบบดังรูป



รูปที่ 2.3 แสดงกระบวนการร้องขอและการส่งกลับของเว็บเพจระหว่างเว็บไคลเอนต์และเว็บเซิร์ฟเวอร์

ก่อนอื่นต้องขออธิบายก่อนว่าในเว็บเพจหนึ่งจะประกอบด้วยส่วนประกอบย่อย (objects) เช่นตัวเว็บเพจหรือตัวโครงสร้าง HTML, รูปภาพที่มีในเว็บเพจนั้น หรือส่วนประกอบอื่นๆ ไม่ว่าจะเป็นแฟ้มข้อมูลแบบเสียงที่มีในเว็บเพจ เมื่อผู้ใช้ทางฝั่งเครื่องไคลเอนต์ร้องขอเว็บเพจโปรแกรมเว็บไคลเอนต์หรือเว็บบราวเซอร์จะส่ง การร้องขอข้อมูลแบบ HTTP (HTTP Request) สำหรับส่วนประกอบต่างๆ (objects) แต่ละอันในเว็บเพจไปยังเว็บเซิร์ฟเวอร์ เมื่อเครื่องเว็บเซิร์ฟเวอร์ได้รับการร้องขอจากผู้ใช้ทางฝั่งเครื่องไคลเอนต์ก็จะตอบกลับด้วยการตอบกลับของการร้องขอข้อมูล HTTP (HTTP Response) ที่ประกอบด้วยส่วนประกอบที่ร้องขอมาอยู่ในเนื้อหา (body) ของการตอบกลับนั้นด้วย ในช่วงปีตั้งแต่ 1997 เว็บเซิร์ฟเวอร์หรือเว็บบราวเซอร์จะทำงานโดยใช้ HTTP เวอร์ชั่น 1.0 [8] และตั้งแต่ปี 1998 เป็นต้นมาเว็บเซิร์ฟเวอร์และเว็บบราวเซอร์ก็เริ่มทำงานโดยใช้ HTTP เวอร์ชั่น 1.1 [9] โดย HTTP เวอร์ชั่น 1.1 จะยังคงสนับสนุน HTTP เวอร์ชั่น 1.0 อยู่ เว็บเซิร์ฟเวอร์ที่ทำงานด้วย HTTP เวอร์ชั่น 1.0 สามารถที่จะติดต่อกับเว็บบราวเซอร์ที่ทำงานด้วย HTTP

เวอร์ชั่น 1.1 ได้และเว็บบราวเซอร์ที่ทำงานด้วย HTTP เวอร์ชั่น 1.0 ก็สามารถติดต่อสื่อสารกับเว็บเซิร์ฟเวอร์ที่ทำงานด้วย HTTP เวอร์ชั่น 1.1 ได้

ทั้ง HTTP/1.1 และ HTTP/1.0 จะใช้ทีซีพี (TCP) เป็นโปรโตคอลในชั้นทรานสปอตต์เลเยอร์ ในการทำงานทางฝั่งของเครื่องไคลเอนต์ที่มีโปรแกรมเว็บไคลเอนต์หรือเว็บบราวเซอร์ทำงานอยู่ เริ่มต้นจะทำการสร้างการเชื่อมต่อแบบทีซีพี (TCP Connection) กับเว็บเซิร์ฟเวอร์และเมื่อการเชื่อมต่อเรียบร้อยแล้ว เว็บบราวเซอร์และเว็บเซิร์ฟเวอร์ก็จะทำการติดต่อกันโดยผ่าน Socket Interface ทางเว็บบราวเซอร์ก็จะส่งข้อมูลการร้องแบบ HTTP ไปยัง Socket Interface และขณะเดียวกันก็จะรับข้อมูลการตอบกลับแบบ HTTP ผ่านทาง Socket Interface เช่นกัน และทางเว็บเซิร์ฟเวอร์ก็เช่นเดียวกัน เมื่อมีการส่งข้อมูลการร้องขอแบบ HTTP จากเว็บบราวเซอร์ไปยังเว็บเซิร์ฟเวอร์หรือการส่งข้อมูลการตอบกลับแบบ HTTP จากเว็บเซิร์ฟเวอร์ไปยังเว็บบราวเซอร์แล้วจะส่งต่อให้เป็นหน้าที่ของทีซีพีในชั้นทรานสปอตเลเยอร์ในการที่จะรับผิดชอบในการทำให้ข้อมูลการร้องแบบ HTTP ส่งถึงเว็บเซิร์ฟเวอร์ หรือข้อมูลการตอบกลับแบบ HTTP ให้ถึงเว็บบราวเซอร์

เว็บเซิร์ฟเวอร์จะส่งเว็บเพจที่ทางเว็บบราวเซอร์ของเครื่องไคลเอนต์ร้องขอโดยไม่มีการเก็บข้อมูลหรือสถานะใดๆเกี่ยวกับทางเครื่องไคลเอนต์ ดังนั้นถ้าทางฝั่งเครื่องไคลเอนต์ร้องขอเว็บเพจที่เคยร้องขอมาแล้ว ทางเว็บเซิร์ฟเวอร์ก็จะไม่มีการบอกว่าเว็บเพจนี้ได้ส่งมาแล้ว แต่เว็บเซิร์ฟเวอร์ก็จะส่งเว็บเพจเดิมกลับมาให้ทางเครื่องไคลเอนต์ นั่นคือจะไม่มีการจดหรือเก็บสถานะใดๆเลยในเว็บเซิร์ฟเวอร์ ซึ่งการทำงานแบบนี้จะเรียก HTTP ว่าเป็นโปรโตคอลที่มีการทำงานเป็น Stateless Protocol

2.2.3.1 การเชื่อมต่อแบบ NON-PERSISTENT และ PERSISTENT

HTTP สามารถใช้ทั้งการเชื่อมต่อแบบไม่คงตัว (Non-Persistent) และการเชื่อมต่อแบบคงตัว (Persistent) โดย การเชื่อมต่อแบบไม่คงตัว (Non-Persistent) เป็นโหมดมาตรฐานของ HTTP/1.0 และการเชื่อมต่อแบบคงตัว (Persistent) นั้นเป็นโหมดเริ่มต้นของ HTTP/1.1

<u>การเชื่อมต่อแบบไม่คงตัว (Non-Persistent)</u>

การทำงานในการสื่อสารระหว่างเว็บบราวเซอร์กับเว็บเซิร์ฟเวอร์ในกรณีที่มีการทำงานแบบไม่คงตัว (Non-Persistent) จะมีขั้นตอนดังนี้

สมมติว่ามีการร้องขอเว็บเพจ *"www.someschool.edu/somedepartment/home.index"* ที่ประกอบด้วยรูป 10 รูปในเว็บเพจนี้

- เว็บบราวเซอร์ที่ทางเครื่องไคลเอนต์จะทำการเริ่มต้นการสร้างการเชื่อมต่อแบบทีซีพี ไปยังเว็บเซิร์ฟเวอร์ของ "www.someschool.edu" ที่ port หมายเลข 80 ที่เป็น port มาตราฐานของเว็บเซิร์ฟเวอร์ที่ใช้ในการรอรับการร้องขอจากทางเครื่องไคลเอนต์

- เว็บบราวเซอร์ที่เครื่องไคลเอนต์ส่งข้อมูลการร้องขอแบบ HTTP เข้าไปใน Socket ที่มีการเชื่อมต่อกับทางเว็บเซิร์ฟเวอร์ที่ได้มีการสร้างการเชื่อมต่อแล้วในขั้นตอนแรก ภายในข้อมูลการร้องขอแบบ HTTP จะประกอบด้วยพาธ "/somedepartment/home.index" ที่ทำการร้องขอ
- เว็บเซิร์ฟเวอร์ได้รับการร้องขอแบบ HTTP จาก Socket ที่มีการเชื่อมต่อกับเครื่องไคลเอนต์แล้ว จากนั้นเว็บเซิร์ฟเวอร์ก็จะทำการดึงเอา "/somedepartment/home.index" จากเนื้อที่ในดิสก์ของเว็บเซิร์ฟเวอร์ที่เก็บอยู่ เอามาแล้วใส่เข้าไปในในส่วนของเนื้อหา(body) ในข้อมูลการตอบกลับแบบ HTTP แล้วส่งกลับไปให้ทางเครื่องไคลเอนต์
- จากนั้นเว็บเซิร์ฟเวอร์จะทำการปิดหรือยกเลิกการเชื่อมต่อกับเว็บบราวเซอร์ของเครื่องไคลเอนต์
- เว็บบราวเซอร์ของเครื่องลูกข่ายได้รับการตอบกลับแบบ HTTP จากเว็บเซิร์ฟเวอร์แล้วทำการแปลข้อมูลของเว็บเพจที่ได้รับจากข้อมูลการตอบกลับแบบ HTTP และทำการหาว่าในเว็บเพจมีส่วนประกอบ (Object) อื่นๆอีกหรือไม่
- จากนั้นเว็บบราวเซอร์ก็จะเริ่มกระบวนข้างต้นอีกครั้งเพื่อทำการร้องขอไฟล์รูปภาพหรือส่วนประกอบอื่นๆ ที่มีอยู่ในเว็บเพจนี้

กระบวนการทั้งหมดข้างต้นเป็นการทำงานการเชื่อมต่อแบบไม่คงตัว (NON-PERSISTENT) เพราะว่าแต่ละ การเชื่อมต่อนั้นจะถูกปิดหรือยกเลิกภายหลังที่เว็บเซิร์ฟเวอร์ส่งส่วนประกอบอันนั้นๆ กลับมาแล้ว การเชื่อมต่อที่สร้างขึ้นจะไม่ถูกใช้สำหรับส่วนประกอบอื่นๆ ดังนั้นจากตัวอย่างข้างต้นจะต้องมีการสร้างการเชื่อมต่อทั้งหมด 11 การเชื่อมต่อในการดูเว็บเพจนี้ (1 การเชื่อมต่อสำหรับโครงสร้าง HTML และ 10 การเชื่อมต่อสำหรับรูปภาพทั้ง 10 ภาพ)

<u>การเชื่อมต่อแบบคงตัว (PERSISTENT)</u>

การเชื่อมต่อแบบคงตัว (PERSISTENT) นั้น เว็บเซิร์ฟเวอร์จะทำการคงการเชื่อมต่อแบบทีซีพีไว้หลังจากที่ได้มีการส่งส่วนประกอบ (Objects) ใดๆ กลับไปให้ทางเครื่องไคลเอนต์แล้ว และการร้องขอส่วนประกอบอื่นๆจากเครื่องไคลเอนต์เครื่องเดียวกันสามารถที่จะถูกส่งผ่านทางการเชื่อมต่อเดิมได้ ซึ่งการเชื่อมต่อแบบคงตัวนี้จะช่วยลดเวลาที่ใช้ในการสร้างการเชื่อมต่อทีซีพี และในการเชื่อมต่อแบบคงตัวนี้ (PERSISTENT) นั้นยังแบ่งเป็นการเชื่อมต่อแบบคงตัวที่มีการส่งข้อมูลแบบตามลำดับซึ่งมีการทำงานที่ทางเครื่องไคลเอนต์จะรอให้ได้รับการตอบกลับแบบ HTTP จากทางเว็บเซิร์ฟเวอร์ก่อนถึงจะทำการร้องขอส่วนประกอบถัดไปและการเชื่อมต่อแบบคงตัวที่มีการส่งข้อมูลแบบขนานที่จะทำการสร้างการเชื่อมต่อทีซีพีหลายๆอันแล้วทำการร้องขอแต่ละส่วนประกอบพร้อมๆกัน ซึ่งจำนวนการเชื่อมต่อแบบทีซีพีที่จะทำการสร้างก็ขึ้นอยู่กับเว็บ

บราวเซอร์ว่ากำหนดให้สร้างได้กี่การเชื่อมต่อในคราวเดียว การเชื่อมต่อแบบคงตัว(PERSISTENT) จะถูกกำหนดเป็นโหมดมาตรฐานของ HTTP/1.1

2.2.3.2 โครงสร้างรูปแบบของ HTTP (HTTP Message Format)

HTTP/1.0 และ HTTP/1.1 จะแบ่งเป็น HTTP Request Message และ Response Message

ข้อมูลการร้องขอแบบ (HTTP Request Message)

ข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) มีรูปแบบดังนี้

GET /somefir/page.html HTTP/1.1

Connection: close

User-agent: Mozilla/4.0

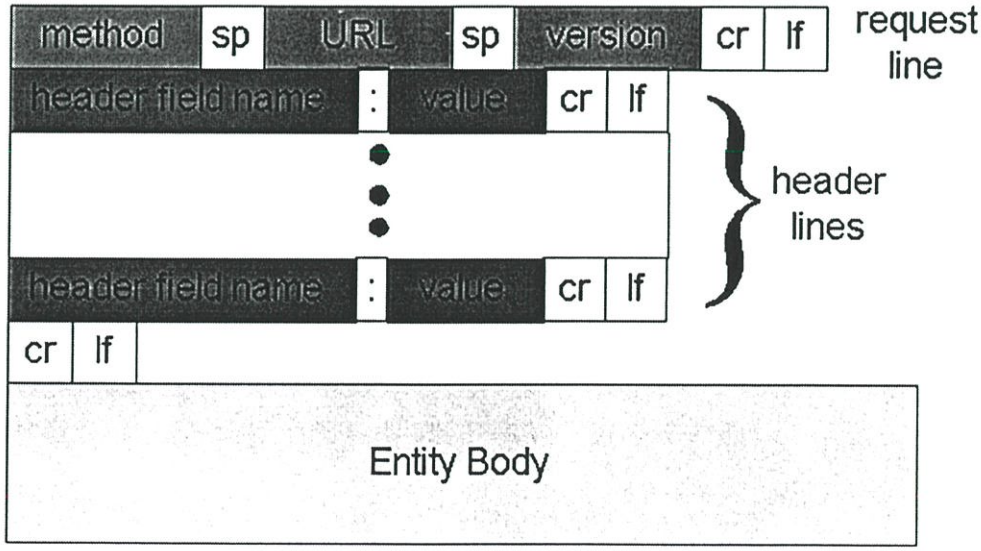
Accept: text/html, image/gif, image/jpeg

Accept-language: fr

รูปที่ 2.4 ลักษณะทั่วไปของข้อมูลการร้องขอแบบ HTTP (HTTP Request Message)

รูปแบบของข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) จะอยู่ในรูปของรหัสแอสกี้ (ASCII code) ธรรมดาดังนั้นคนทั่วๆไปสามารถที่จะอ่านได้ ในข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) ตามรูปที่ 2.4 นั้นตัวอย่างจะประกอบด้วย 5 บรรทัด แต่ละบรรทัดจะต้องจบด้วยอักษรจบบรรทัด (Carriage Return) และอักษรเริ่มต้นบรรทัดใหม่ (Line Feed) และในบรรทัดสุดท้ายจะมีการเพิ่มอักษรจบบรรทัด (Carriage Return) และอักษรเริ่มต้นบรรทัดใหม่ (Line Feed) ไปอีกหนึ่งครั้งเป็น 2 ครั้งเฉพาะในบรรทัดสุดท้าย ในบรรทัดแรกของข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) จะเรียกว่า "Request Line" ซึ่งบรรทัดหลังจากนี้จะเรียกว่า "Header Line" ใน Request Line จะประกอบด้วย 3 ฟิลด์คือ ฟิลด์ Method, ฟิลด์ URL และ ฟิลด์ HTTP Version ที่ใช้และใน ฟิลด์ Method จะสามารถมีได้หลายค่าคือ GET, POST หรือ HEAD การใช้งานส่วนใหญ่ของข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) จะใช้ GET Method ซึ่งการใช้ GET Method จะเป็นการที่บราวเซอร์หรือเว็บไคลเอนต์ที่เครื่องลูกข่ายร้องขอส่วนประกอบใดๆในเว็บเพจ (ซึ่งจะระบุในฟิลด์ URL และในตัวอย่างคือ "/somedir/page.html") จากเว็บเซิร์ฟเวอร์และจะบอกเวอร์ชั่นของ HTTP ที่ใช้ในฟิลด์ HTTP version (ซึ่งในตัวอย่างคือเวอร์ชั่น 1.1)

สำหรับในส่วนของ Header Line ในตัวอย่างจะประกอบด้วย *Connection: close* Header Line เป็นการที่บราวเซอร์หรือเว็บไคลเอนต์บอกทางเว็บเซิร์ฟเวอร์ว่าไม่ต้องการใช้การเชื่อมต่อแบบคงตัว (PERSISTENT) นั่นก็คือต้องการให้ทางเว็บเซิร์ฟเวอร์ปิดการเชื่อมต่อหลังจากที่ได้มีการส่งส่วนประกอบใดๆ (Object) ของเว็บเพจที่ร้องขอแล้ว ที่ต้องบอกเพราะว่าใน Request Line ทางบราวเซอร์บอกทางเว็บเซิร์ฟเวอร์ว่าใช้ HTTP/1.1 ซึ่งถ้าไม่บอกใน *Connection: close* ใน Header Line แล้วทางเว็บเซิร์ฟเวอร์จะใช้การเชื่อมต่อแบบคงคงตัว PERSISTENT (เพราะเป็นโหมดมาตรฐานของ HTTP เวอร์ชั่น 1.1) สำหรับบรรทัดถัดไป *User-agent:* Header Line จะเป็นการบอกว่าบราวเซอร์ที่ทำการร้องขอนั่นเป็นบราวเซอร์อะไร (ในตัวอย่างคือ *Mozilla/4.0* นั่นก็คือ NETSCAPE บราวเซอร์) และ *Accept:* Header Line จะเป็นการบอกเว็บเซิร์ฟเวอร์ว่าทางบราวเซอร์สามารถรองรับส่วนประกอบของเว็บเพจแบบไหนบ้าง (ซึ่งในตัวอย่างคือ html ไฟล์, รูปภาพแบบ jpeg, gif) ถ้าใน "/somedir/page.html" มีไฟล์ประเภทอื่นเช่น จาวา แล้วทางเว็บเซิร์ฟเวอร์ก็จะไม่ส่งไฟล์นั้นให้ทางบราวเซอร์ที่เครื่องไคลเอนต์และสุดท้าย *Accept-language:* Header Line นั้นเป็นการบอกว่าทางผู้ใช้ที่เครื่องไคลเอนต์นั้นอยากที่จะรับส่วนประกอบต่างๆ (Objects) ของเว็บเพจที่เป็นภาษาฝรั่งเศส ซึ่งถ้าทางเว็บเซิร์ฟเวอร์มีส่วนประกอบต่างๆ (Objects) ที่เป็นแบบฝรั่งเศสก็ให้ส่งกลับมา แต่ถ้าไม่มีการจะส่งในแบบที่เป็นมาตราฐานกลับมา



รูปที่ 2.5 โครงสร้างของ HTTP Request Message

จากรูปหลังจากข้อมูลในส่วน Header Line จะเป็นส่วนที่เรียกว่า "Entity Body" ซึ่ง Entity Body จะไม่ใช้กับ GET Method แต่จะใช้กับ POST Method การใช้ POST Method ทางบราวเซอร์จะใช้ในการส่งข้อมูลบางอย่างไปให้กับเว็บเซิร์ฟเวอร์ เช่นการค้นหาบนเว็บ ทางผู้ใช้ที่เครื่องไคลเอนต์ก็จะต้องส่งข้อความที่จะต้องการค้นหาไปให้เว็บเซิร์ฟเวอร์ และข้อมูลส่วน Entity Method จะส่งถูกส่งผ่านฟอร์มบนเว็บเพจ และสุดท้ายคือ HEAD Method จะใช้ในการตรวจสอบแก้ไขความผิดพลาดของ HTTP Request Message

ข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message)

รูปแบบทั่วไปของข้อมูลการตอบกลับแบบ HTTP (HTTP response message) เป็นดังรูปที่ 2.6

HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
Data data data data data ...

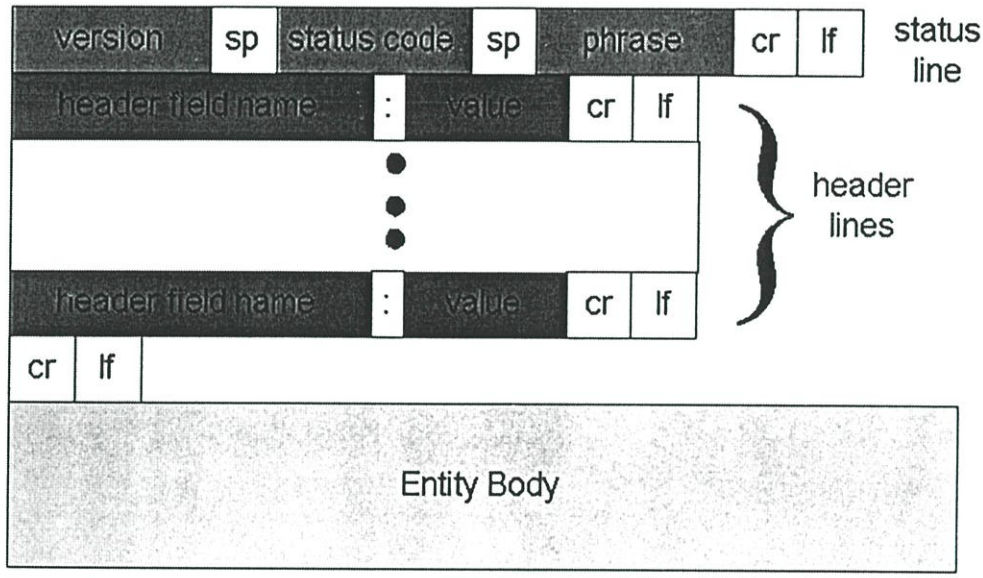
รูปที่ 2.6 ลักษณะทั่วไปของการตอบกลับแบบ HTTP Response Message

ในข้อมูลการตอบกลับแบบ HTTP (HTTP response message) ตามตัวอย่างประกอบด้วย 3 ส่วนคือ Status Line, 6 Header Line และ Entity Body ในส่วนของ Entity Body จะเป็นข้อมูลของ Object ที่ทางผู้ใช้ร้องขอจากเว็บเซิร์ฟเวอร์(ส่วนของ *Data data data data data ...* ในรูป 2.4) ใน Status Line จะประกอบด้วย 3 ฟิลด์คือ ฟิลด์ Protocol Version, ฟิลด์ Status Code และ ฟิลด์ Corresponding Status Code ในตัวอย่าง Protocol version คือ HTTP/1.1 และ Status Code แสดงการส่งข้อมูลถูกต้อง (200 OK)

สำหรับในส่วนของ Header Line ทางเว็บเซิร์ฟเวอร์จะใช้ *Connection: close* Header Line ที่จะบอกกับทางเว็บบราวเซอร์ของผู้ใช้ว่าจะมีการปิดการเชื่อมต่อหลังจากที่ได้ส่งข้อมูลของส่วนประกอบใดๆ (Objects) ของเว็บเพจกลับมาให้แล้ว ส่วน *Date:* Header Line เป็นการบอกวันและเวลาที่ข้อมูลการตอบกลับแบบ HTTP (HTTP response message) ได้ถูกสร้างขึ้นแล้วส่งกลับไปให้ทางเว็บบราวเซอร์ของผู้ใช้ ส่วน *Server:* Header Line จะใช้บอกว่าข้อมูลการตอบกลับแบบ

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

HTTP (HTTP response message) นี้ถูกสร้างจากเว็บเซิร์ฟเวอร์อะไร (ซึ่งในตัวอย่างคือ APACHE WEB SEVER) สำหรับ *Last-Modified:* Header Line จะเป็นการบอกว่าส่วนประกอบใดๆ (Objects) ของเว็บเพจที่ได้ส่งกลับมาให้นี้ได้มีการแก้ไขครั้งล่าสุดเมื่อไหร่ สำหรับ *Content-Length:* Header Line คือขนาดเป็นไบต์ของส่วนประกอบของเว็บเพจที่ส่งกลับมา และสุดท้าย *Content-Type:* Header Line เป็นการบอกว่าใน Entity Body นั้นเป็นส่วนประกอบของเว็บเพจแบบใด(จากในตัวอย่างก็เป็นการบอกว่าใน Entity Body เป็นข้อมูลแบบไฟล์ HTML)



รูปที่ 2.7 โครงสร้างของข้อมูลการตอบกลับแบบ HTTP (HTTP response message)

รูปที่ 2.7 เป็นรูปแบบมาตรฐานของข้อมูลการตอบกลับแบบ HTTP (HTTP response message) และในส่วนของ Status Code จะมีได้หลายค่าตามการเชื่อมต่อของการตอบกลับแบบ HTTP สำหรับตัวอย่างและความหมายของบาง Status Code และ Corresponding Status Code คือ

- 200 OK : แสดงว่า Request ของผู้ใช้ถูกต้อง และได้ส่งข้อมูลกลับมาให้แล้ว
- 301 Moved Permanently : บ่งบอกว่าส่วนประกอบของเว็บเพจที่ทางผู้ใช้ร้องขอได้มีการเปลี่ยนที่อยู่แล้วและตำแหน่งที่อยู่ใหม่ของส่วนประกอบของเว็บเพจ (Objects) จะระบุใน *Location:* Header แล้วบราวเซอร์ของทางผู้ใช้จะทำการร้องขอไปที่ URL ตาม *Location:* Header โดยอัตโนมัติ

51616

- 401 Bad Request เป็นการบอกว่าการร้องขอแบบ HTTP ที่ทางผู้ใช้ส่งมามีรูปแบบไม่ถูกต้อง
- 404 Not Found เป็นการบอกว่า Object ที่ทางผู้ใช้ร้องขอไม่มีอยู่บนเว็บเซิร์ฟเวอร์
- 505 HTTP Version Not Supported เป็นการบอกว่าเว็บเซิร์ฟเวอร์ไม่รองรับ HTTP Version ที่ระบุมาใน Request Message การติดต่อกันระหว่างเครื่องไคลเอนต์และเครื่องเซิร์ฟเวอร์

2.2.3.3 การติดต่อกันระหว่างเครื่องไคลเอนต์และเครื่องเซิร์ฟเวอร์ที่มีการใช้การยืนยันตัวเอง (AUTHENTICATION) และการใช้คุกกี้ (COOKIES)

HTTP มีการทำงานแบบไม่เก็บสถานะ (Stateless) เกี่ยวกับผู้ใช้งานไว้ แต่อย่างไรก็ดีเว็บไซต์ต่างๆ อาจจะจำเป็นต้องมีการจดจำหรือต้องการจำแนกผู้ใช้เพราะจุดประสงค์เกี่ยวกับการจำกัดการเข้าถึงเว็บไซต์หรือต้องการที่จะเก็บข้อมูลบางอย่างที่ใช้กับแต่ละคน ซึ่งใน HTTP ก็ได้มีกลไกที่จะใช้ช่วยในการทำงานลักษณะนี้คือ การยืนยันตัวเอง (AUTHENTICATION) และการใช้คุกกี้ (COOKIES)

<u>การยืนยันตัวเอง (AUTHENTICATION)</u>

บางเว็บไซต์อาจต้องการให้ผู้ใช้งานใส่ชื่อบัญชีผู้ใช้ (Username) และรหัสลับ (Password) ในการที่เข้าถึงเอกสารหรือเว็บเพจของเว็บไซต์นั้นๆ ซึ่งทำให้ต้องการกลไกที่เรียกว่า กระบวนการยืนยันตัวเอง (AUTHENTICATION) โดย HTTP จะมีการกำหนด Status Code และ Header แบบพิเศษที่จะช่วยให้เว็บไซต์สามารถที่จะทำกระบวนการยืนยันตัวเองได้ ตัวอย่างในการทำงานของกระบวนการยืนยันตัวเองมีดังนี้

- ผู้ใช้งานที่เครื่องลูกข่ายส่งข้อมูลการร้องขอเว็บเพจแบบ HTTP (HTTP Request Message) ไปยังเว็บเซิร์ฟเวอร์
- ทางเว็บเซิร์ฟเวอร์จะตอบกลับด้วยข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) ที่จะไม่มีส่วนประกอบใดๆของเว็บเพจใน Entity Body ส่งกลับมาด้วยและจะตอบด้วย Status Code ว่า 401 Authorization Required และในส่วนของ Header จะมีค่าเป็น WWW-Authentication Header เพื่อที่จะบ่งบอกว่าทางเว็บเซิร์ฟเวอร์กำหนดให้มีการยืนยันตัวเอง (AUTHENTICATION) ในการเข้าถึงเว็บเพจนั้นๆ
- ทางเว็บบราวเซอร์ของผู้ใช้งานที่เครื่องไคลเอนต์ได้รับการตอบกลับแบบ HTTP ที่ตอบกลับมา ก็จะมีการเตือนให้ใส่ชื่อบัญชีผู้ใช้ (Username) และรหัสลับ (Password) หลังจากที่ผู้ใช้งานใส่ชื่อบัญชีผู้ใช้และรหัสลับแล้ว เว็บบราวเซอร์หรือเว็บไคลเอนต์ก็จะส่งข้อมูลการร้องขอแบบ HTTP ไ ปยังเว็บเซิร์ฟเวอร์อีกครั้งซึ่งในการร้องขอแบบ HTTP ครั้งนี้จะมีส่วน

ของ Header เป็น Authorization: Header และมีชื่อบัญชีผู้ใช้ (Username) และรหัสลับ (Password) อยู่ในภายในส่วนของ Header ด้วย

หลังจากที่ได้รับส่วนประกอบของเว็บเพจ(Objects) แรกแล้วทางเว็บบราวเซอร์หรือเว็บไคลเอนต์ที่เครื่องไคลเอนต์ก็จะส่งการร้องขอแบบ HTTP ของส่วนประกอบของเว็บ (Objects) ที่เหลือซึ่งในการร้องขอครั้งนี้และครั้งต่อๆไป นั้นก็จะส่งชื่อบัญชีผู้ใช้ (Username) และรหัสลับ (Password) อยู่ภายในการร้องขอแบบ HTTP นั้นๆ ด้วย ซึ่งชื่อบัญชีผู้ใช้และรหัสลับนั้นจะถูกเก็บไว้ที่แคช (Cache) ทำให้ผู้ใช้งานไม่จำเป็นต้องใส่ชื่อบัญชีผู้ใช้และรหัสลับทุกครั้ง ชื่อบัญชีผู้ใช้และรหัสลับจะยังคงอยู่ในแคช (Cache) ภายในช่วงระยะเวลาหนึ่งเท่านั้น

คุกกี้ (COOKIES)

คุกกี้ (COOKIES) เป็นกลไกอีกอย่างหนึ่งสำหรับเว็บเซิร์ฟเวอร์ในการที่เก็บข้อมูลบางอย่างของผู้ใช้ไว้ โดยจะถูกกำหนดไว้ใน RFC 2101 [10] ตัวอย่างในการใช้งานคุกกี้มีดังนี้ สมมติผู้ใช้งานต้องการเข้าถึงเว็บไซต์หนึ่งและในเว็บไซต์นี้มีการใช้งานคุกกี้จากนั้นทางเว็บเซิร์ฟเวอร์ก็จะตอบกลับด้วยข้อมูลการตอบกลับแบบ HTTP ที่มี *Set-Cookie*: อยู่ใน Header ซึ่ง Header นี้จะประกอบด้วยหมายเลขที่ใช้ในการจำแนกผู้ใช้โดยหมายเลขนี้จะถูกสร้างจากทางเว็บเซิร์ฟเวอร์สมมติให้เป็น

Set-cookie: 1678453

เมื่อทางผู้ใช้งานได้รับการตอบกลับนี้แล้วเว็บบราวเซอร์ของเครื่องไคลเอนต์ก็จะพบว่ามี *Set-Cookie:* Header และมีหมายเลขของคุกกี้ในส่วนของ Header จากนั้นเว็บบราวเซอร์ก็จะเพิ่มข้อมูลของคุกกี้นี้ในไฟล์ที่เว็บบราวเซอร์ใช้ในการเก็บข้อมูลของคุกกี้ไว้บนเครื่องไคลเอนต์โดยในไฟล์นั้นก็จะเก็บชื่อเครื่องเซิร์ฟเวอร์ (HOSTNAME) ของเว็บเซิร์ฟเวอร์และหมายเลขที่ทางเว็บเซิร์ฟเวอร์ส่งมาไว้ ดังนั้นการร้องขอในครั้งถัดๆไปในส่วนของ Header ของข้อมูลการร้องแบบ HTTP (HTTP Request Message) ที่ทางเว็บบราวเซอร์ของผู้ใช้จะส่งให้ทางเว็บเซิร์ฟเวอร์ก็จะมี *Cookies:* Header และก็บอกหมายเลขนั้นกับทางเว็บเซิร์ฟเวอร์ด้วยดังเช่น

Cookie: 1678453

ซึ่งในกรณีนี้ทางเว็บเซิร์ฟเวอร์จะไม่รู้ชื่อบัญชีผู้ใช้ และรหัสลับของผู้ใช้งานแต่ทางเว็บเซิร์ฟเวอร์จะสามารถรู้ได้ว่าผู้ใช้งานคนนี้เป็นคนเดียวกับผู้ใช้งานที่เคยเข้ามาแล้ว นอกจากนี้เว็บเซิร์ฟเวอร์สามารถใช้คุกกี้ในการทำงานหลายๆอย่างเช่น

- ถ้าทางเว็บเซิร์ฟเวอร์ต้องการการยืนยันตัวเอง (AUTHENTICATION) ของผู้ใช้แต่ไม่ต้องการให้ทางผู้ใช้ใส่ชื่อบัญชีผู้ใช้และรหัสลับทุกครั้งก็สามารถใช้คุกกี้ได้
- ถ้าทางเว็บเซิร์ฟเวอร์ต้องการจำข้อมูลที่ใช้กับผู้ใช้แต่ละคน ก็สามารถใช้คุกกี้ได้
- ถ้าผู้ใช้มีการซื้อสินค้าผ่านทางเว็บไซต์ เว็บเซิร์ฟเวอร์ก็สามารถใช้คุกกี้ในการเก็บข้อมูลสินค้าที่ผู้ใช้ซื้อไว้ เพื่อใช้ในการสร้าง Shopping Cart ได้

แต่อย่างไรก็ดีคุกกี้ (Cookie) ก็ยังมีปัญหาในกรณีที่ผู้ใช้งานคนเดิมแต่ใช้เครื่องไคลเอนต์คนละเครื่องกับที่เคยเข้าถึงเว็บไซต์ เว็บเซิร์ฟเวอร์จะไม่สามารถจำแนกหรือจดจำข้อมูลผู้ใช้คนนี้ได้

2.2.3.4 CONDITIONAL GET

เว็บแคช (Web Cache) นั้นเป็นการเก็บส่วนประกอบต่างๆของเว็บเพจ (Objects) ที่เคยมีการเข้าถึงแล้วนั้นไว้เพื่อใช้ในการลดเวลาในการเข้าถึงส่วนประกอบต่างๆของเว็บเพจเดิมนั้น โดย เว็บแคช (Web Cache) สามารถทำที่เครื่องไคลเอนต์หรือเครื่องใดๆ ที่ทำหน้าแคชส่วนประกอบนั้นๆ (Object) เฉพาะก็ได้ (Proxy Server or Cache Server)

แม้ว่าเว็บแคช (Web Cache) จะสามารถช่วยลดเวลาในการเข้าถึงส่วนประกอบต่างๆของเว็บเพจใดๆ ที่อยู่บนเว็บเซิร์ฟเวอร์ได้ แต่ก็จะมีปัญหาที่เกิดขึ้นมาอีกก็คือ จะรู้ได้อย่างไรว่าส่วนประกอบของเว็บเพจนั้นที่ได้มีการเก็บหรือแคชไว้ในเครื่องไคลเอนต์หรือที่พร็อกซี่เซิร์ฟเวอร์ (Proxy Server) นั้นเก่าไปหรือได้มีการแก้ไขแล้วที่เว็บเซิร์ฟเวอร์แล้วหรือยัง ดังนั้น HTTP จึงมีกลไกที่ใช้ในการตรวจสอบกับทางเว็บเซิร์ฟเวอร์ว่าส่วนประกอบของเว็บเพจใดๆ ที่ได้มีการเก็บหรือแคชไว้ได้มีการแก้ไขแล้วหรือไม่ โดยกลไกนี้จะเรียกว่า CONDITIONAL GET ซึ่งการร้องขอแบบ HTTP แบบนี้จะถูกเรียกว่า "การร้องขอแบบ CONDITIONAL GET (CONDITIONAL GET Message)" ซึ่งในการร้องขอแบบ HTTP นี้จะมีการใช้ GET Method และใน Header มีการระบุ *If-Modified-Since:* Header ไว้

GET /fruit/kiwi.gif HTTP/1.0

User-agent: Mozilla/4.0

Accept: text/html, image/gif, image/jpeg

รูปที่ 2.8 ตัวอย่างของข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) ทั่วๆไป

และทางเว็บเซิร์ฟเวอร์ก็จะตอบกลับด้วยข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) นี้

HTTP/1.0 200 OK

Date: Wed, 12 Aug 1998 15:39:29

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 09:23:24

Content-Type: image/gif

data data data data data .....

รูปที่ 2.9 ตัวอย่างของข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) ทั่วๆไป

จากนั้นทางเว็บบราวเซอร์ของเครื่องไคลเอนต์ก็จะแสดงผลส่วนประกอบของเว็บเพจนั้นและก็จะทำการเก็บส่วนประกอบของเว็บเพจนี้ไว้ในแคชและจะเก็บ Last-Modified ของส่วนประกอบนี้ไว้ด้วย สมมติหลังจากนี้ 3 อาทิตย์ผ่านไปผู้ใช้ที่เครื่องไคลเอนต์เดียวกันนี้ต้องการร้องขอเว็บเพจเดียวกันนี้และส่วนประกอบของเว็บเพจนี้ยังอยู่ในแคชและด้วยเหตุที่ว่าส่วนประกอบนี้อาจมีการเปลี่ยนแปลงเกิดขึ้นที่เว็บเซิร์ฟเวอร์ ดังนั้นเว็บบราวเซอร์ก็จะทำการตรวจสอบว่าเกิดการแก้ไขส่วนประกอบของเว็บเพจนี้ที่เว็บเซิร์ฟเวอร์หรือไม่ด้วย CONDITIONAL GET โดยจะส่งการร้องขอที่มีการข้อมูลการร้องขอแบบ HTTP ดังนี้

GET /fruit/kiwi.gif HTTP/1.0

User-agent: Mozilla/4.0

Accept: text/html, image/gif, image/jpeg

If-modified-since: Mon, 22 Jun 1998 09:23:24

รูปที่ 2.10 ข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) ที่มีการใช้ CONDITIONAL GET

โดยค่าของ *If-modified-since*: header จะเป็นค่าที่ได้จาก *Last-Modified:* header ที่ได้รับจากเว็บเซิร์ฟเวอร์เมื่อตอนที่ได้มีการเก็บหรือแคชส่วนประกอบของเว็บเพจหรือตัวเว็บเพจนี้ไว้ CONDITIONAL GET นี้จะบอกกับทางเว็บเซิร์ฟเวอร์ว่าให้ส่งส่วนประกอบที่ร้องขอของเว็บเพจนี้กลับมาถ้าส่วนประกอบนี้ได้มีการแก้ไขเกิดขึ้นหลังจากวันที่ระบุใน *If-modified-since* : Header แล้ว ถ้าไม่มีการแก้ไขเกิดขึ้นหลังจากวันที่ระบุ ทางเว็บเซิร์ฟเวอร์ก็จะตอบกลับด้วยการตอบกลับที่มีข้อมูลการตอบกลับดังนี้

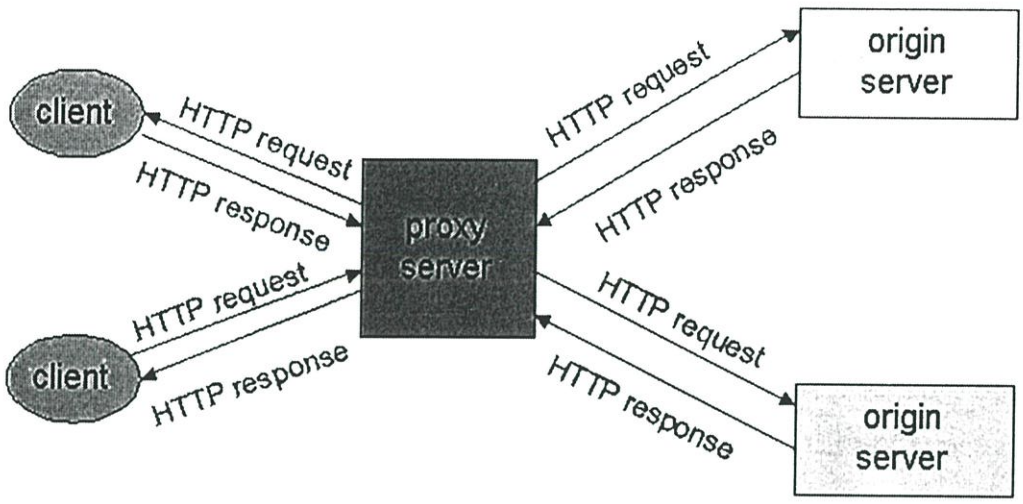
HTTP/1.0 304 Not Modified
Date: Wed, 19 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
(empty entity body)

รูปที่ 2.11 ข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) ที่มีตอบกลับด้วย CONDITIONAL GET

จากรูปที่ 2.11 ทางเว็บเซิร์ฟเวอร์จะยังคงส่งข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) กลับมา แต่จะไม่ได้ส่งส่วนประกอบนั้นกลับมา เพื่อแต่ส่งข้อมูลการตอบกลับแบบ HTTP ที่มี Status Code 304 Not Modified เพื่อบอกว่าส่วนประกอบ (Objects) นั้นไม่ได้รับการแก้ไขหลังจากวันที่ระบุใน *If-modified-since* header ในข้อมูลการร้องขอแบบ HTTP ที่ร้องขอไป ดังนั้นทางเว็บบราวเซอร์ก็สามารถใช้ส่วนประกอบชิ้นนั้นของเว็บเพจนั้นที่ได้เก็บหรือแคชไว้แสดงผลได้เลย

2.3 ขั้นตอนการเข้าถึงเว็บเพจ

การเข้าถึงเว็บเพจบนเวิลด์ไวด์เว็บโดยทั่วไปที่มีการใช้งานอยู่ในปัจจุบันจากเป็นดังรูปที่ 2.12 ซึ่งการทำงานในการเข้าถึงเว็บเพจบางที่ของผู้ใช้อาจจะประกอบด้วยส่วนโครงสร้างต่างๆที่แตกต่างจากรูปที่ 2.12 ก็ได้



รูปที่ 2.12 โครงสร้างในการเข้าถึงเว็บเพจในเวิลด์ไวด์เว็บ

สำหรับรูปที่ 2.12 นั้นจะอธิบายได้ดังนี้

2.3.1 เว็บไคลเอนต์

เว็บไคลเอนต์จะเป็นส่วนของโปรแกรมที่ทำงานที่เครื่องไคลเอนต์ซึ่งก็คือเว็บบราวเซอร์นั่นเอง เว็บบราวเซอร์เป็นโปรแกรมที่ผู้ใช้ใช้ในการดูเว็บเพจบนเวิลด์ไวด์เว็บจะทำงานในระดับแอพลิเคชั่นเลเยอร์ (Application Layer) หน้าที่หลักของเว็บบราวเซอร์มีดังนี้

2.3.1.1 ส่งการร้องขอเว็บเพจ

เว็บบราวเซอร์จะทำการสร้างข้อมูลการร้องขอแบบ HTTP (HTTP Request Message) แล้วส่งไปยังเว็บเซิร์ฟเวอร์เมื่อผู้ใช้คลิกที่ลิงค์บนเว็บเพจหรือเมื่อมีการพิมพ์ URL โดยตรง นอกจากนี้เว็บบราวเซอร์จะทำการสร้างและส่งการร้องขอแบบ HTTP เพื่อร้องขอส่วนประกอบต่างๆที่มีในเว็บเพจด้วย

2.3.1.2 จัดการกับการตอบกลับของเว็บเซิร์ฟเวอร์

เว็บบราวเซอร์จะทำหน้าที่ในการจัดการกับการตอบกลับด้วยข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) ที่ได้รับเว็บเซิร์ฟเวอร์โดยเว็บบราวเซอร์จะทำการแปลหรือเรียกว่า "Parse" ข้อมูลของเว็บเพจที่อยู่ในข้อมูลการตอบกลับแบบ HTTP แล้วก็จะแสดงผลเว็บเพจนั้น จากนั้นก็จะทำการร้องขอส่วนประกอบต่างๆ ที่อยู่ภายในเว็บเพจนั้นเพื่อใช้ในการแสดงผลเว็บเพจนี้ให้ครบถ้วน นอกจากนี้ ถ้ามีส่วนของโปรแกรมที่ทำงานที่ฝ่ายเครื่องไคลเอนต์ที่ต้องใช้ในการ

แสดงผลเว็บเพจด้วย เว็บบราวเซอร์ก็จะทำงานในส่วนของโปรแกรมนั้นด้วย ตัวอย่างของโปรแกรมที่ทำงานที่ฝ่ายเครื่องลูกข่ายเช่น การใช้ Javascript ในเว็บเพจ

2.3.1.3 จัดการเกี่ยวกับการแคชชิ่ง (Caching)

การแคชชิ่งจะเป็นการเก็บส่วนประกอบของเว็บเพจ (Object) หรือเอกสารที่เคยร้องขอมาแล้วจากเว็บเซิร์ฟเวอร์ไว้ในเนื้อที่ส่วนหนึ่งในเครื่องไคลเอนต์ของผู้ใช้ ในการเข้าถึงส่วนประกอบของเว็บเพจหรือตัวเว็บเพจนั้นอีกครั้งภายหลังก็สามารถดึงจากแคชได้เลย การแคชชิ่งจะช่วยลดเวลาในการเข้าถึงเว็บเพจต่างและยังช่วยในการลดความคับคั่งที่เกิดบนเวิลด์ไวด์เว็บอีกด้วย แต่อย่างไรก็ตามการแคชชิ่งจะต้องมีการใช้กลไกบางอย่างในการทำงานเพื่อใช้ในการตรวจสอบว่าส่วนประกอบใดๆของเว็บเพจที่ได้มีการแคชไว้ได้มีการแก้ไขที่เว็บเซิร์ฟเวอร์หรือไม่ ซึ่งกลไกตรงนี้เรียกว่า "การตรวจสอบข้อมูลที่แคชไว้ (Cache Revalidation)" และทำงานร่วมกับโปรโตคอล HTTP ด้วย ดังที่กล่าวในหัวข้อ HTTP (CONDITIONAL GET)

2.3.1.4 การจัดเก็บการตั้งค่าต่างๆ

เว็บบราวเซอร์นั้นจะมีการทำงานเป็นแบบที่มีส่วนตอบโต้กับผู้ใช้งาน (Interactive Software) ดังนั้นจึงสามารถที่จะปรับแต่งการทำงานของเว็บบราวเซอร์ได้ การปรับแต่งนั้นจะแบ่งเป็น

2.3.1.4.1 การแสดงผลทางกายภาพ

ผู้ใช้งานสามารถที่จะปรับแต่งการแสดงผลเว็บเพจได้ตามความต้องการของผู้ใช้เอง ขอบเขตในการปรับแต่งนั้นก็ขึ้นกับเว็บบราวเซอร์เองว่ามีความสามารถในการปรับแต่งได้มากน้อยเพียง ซึ่งจะจำแนกได้เป็น

- การแสดงผลภายนอก : การปรับแต่งในส่วนนี้จะเป็นการปรับแต่งการรูปแบบของเว็บบราวเซอร์เอง ไม่ว่าจะเป็นขนาดของเว็บบราวเซอร์ รูปแบบของปุ่มหรือแถบเลื่อนต่างๆ
- แสดงผลรูปภาพต่างๆที่มีในเว็บเพจ : ในเว็บเพจนั้นสามารถมีได้ทั้งตัวอักษรแล้วรูปภาพต่างๆ ดังนั้นเว็บบราวเซอร์ก็อาจจะสามารถที่ปรับแต่งให้มีการดึงและแสดงผลรูปภาพที่มีในเว็บเพจด้วยหรือไม่แสดงก็ได้ เพราะในกรณีที่มีรูปขนาดใหญ่อยู่ในเว็บเพจอาจจะทำให้ใช้เวลาในการดึงรูปนั้นนานได้
- ลักษณะตัวอักษร: เว็บบราวเซอร์สามารถที่จะปรับเปลี่ยนรูปแบบตัวอักษรที่จะใช้แสดงผลได้ไม่ว่าจะเป็นขนาดรูปแบบของตัวอักษร ขนาดของตัวอักษร
- สีต่างๆ : การปรับแต่งในส่วนนี้เป็นการปรับแต่งในส่วนของสีพื้นหลังของเว็บบราวเซอร์เองหรือจะเป็นสีของตัวอักษรต่างๆในเว็บเพจเช่น ข้อความที่เป็นลิงค์ไปยังเว็บเพจอื่นที่ยังไม่เคยดูเป็นสีแดง ขณะที่ส่วนของข้อความที่เป็นลิงค์ไปยังเว็บเพจอื่นที่เคยดูมาแล้วเป็นสีเขียว

2.3.1.4.2 การจัดการส่วนที่ไม่นิยามในโปรโตคอล HTTP

เว็บเพจในปัจจุบันไม่ได้มีเพียงลิงค์ไปยังเว็บเพจต่างๆเท่านั้น ยังมีการลิงค์ไปยังเอกสารที่เป็นรูปแบบต่างๆ เช่น PDF, PS, DOC ซึ่งถ้าเป็นเว็บเพจหรือรูปภาพแล้วเว็บบราวเซอร์สามารถที่จะจัดการในการแสดงผลได้ แต่ถ้าเป็นไฟล์รูปแบบอื่นๆแล้วเว็บบราวเซอร์จะต้องอาศัยโปรแกรมภายนอก (Helper Application) ในการช่วยแสดงผล เช่นถ้าผู้ใช้งานต้องการร้องขอไฟล์ PDF บนเว็บเพจ เว็บบราวเซอร์จะต้องดึงข้อมูลของไฟล์ PDF นั้นจากเว็บบราวเซอร์แล้วทำการเรียกให้โปรแกรมที่สามารถแสดงผลไฟล์นั้นขึ้นมาทำงานและแสดงผลไฟล์นั้น การที่เว็บบราวเซอร์สามารถรู้ได้ว่าจะต้องเรียกใช้โปรแกรมภายนอก (Helper Application) อะไรโดยดูจาก MIME Type ในข้อมูลการตอบกลับแบบ HTTP ที่ได้รับจากเว็บเซิร์ฟเวอร์ ตัวอย่างของ MIME Type และโปรแกรมภายนอกที่ใช้เป็นดังตาราง

ตารางที่ 2.1 แสดง MIME Type ต่างๆ

Content type	MIME type	Helper application
Zip compressed data	application/x-zip-compressed	*gunzip*/WINZIP32
PostScript document	application/postscript	*ghostview*/GSVIEW32
Word document	application/msword	*catdoc*/WINWORD
PDF document	application/pdf	*acroread*/ACRORD32
audio/video	video/x-mpeg-2	*raplayer*/MPLAYER2

ในตารางแสดงว่าถ้า Object มี MIME Type เป็น application/postscript ให้ทำการเรียกใช้โปรแกรมภายนอกคือ ghostview

แต่อย่างไรก็ดีเว็บบราวเซอร์สามารถที่จะเรียกในงานโปรแกรมภายนอกบางอย่างโดยอัตโนมัติโดยที่ผู้ใช้ไม่สามารถรู้ได้เลยเพราะก็แสดงผลจะอยู่เว็บบราวเซอร์เลย เช่น เว็บบราวเซอร์ สามารถเรียกใช้ Acroread ในการแสดงผลไฟล์ PDF format และแสดงผลในตัวเว็บบราวเซอร์ได้เลย ซึ่งโปรแกรมภายนอกที่มีความสามารถแบบนี้จะเรียกว่า Plug-in ซึ่งการเรียกใช้งาน Plug-in ใดก็อาจดูที่ MIME Type หรือดูที่แท็กพิเศษใช้ข้อมูลของเว็บเพจ

2.3.1.5 การจัดการด้านการรักษาความปลอดภัย

เนื่องจากในปัจจุบันมีเว็บไซต์เกิดขึ้นมากมาย ในแต่ละเว็บไซต์ก็ประกอบด้วยเว็บเพจแบบต่างๆ ทั้งแบบไม่เปลี่ยนแปลง(static) หรือแบบมีการเปลี่ยนแปลงได้(dynamic) นอกจากนี้ยังสามารถที่จะประกอบด้วยส่วนการทำงานที่ทำให้เกิดความเสี่ยงต่อพวกไวรัสหรือการลักลอบเข้าถึงเครื่องของผู้ใช้ได้ เช่น Javascript, ActiveX, Perl Script ทำให้เว็บบราวเซอร์เองนั้นจะต้องมีการกลไกที่ใช้ในการป้องกันการเข้าถึงเครื่องของผู้ใช้และรวมถึงป้องกันการทำงานของพวกไวรัสได้ด้วย ซึ่งส่วนการทำงานที่ทำให้เกิดปัญหาได้นี้ส่วนใหญ่จะมาจาก

- JAVA : JAVA เป็นภาษาโปรแกรมที่จะใช้ในการเขียนโปรแกรมที่ทำงานบนเว็บเพจ เรียกว่า Java Applet โดยเว็บบราวเซอร์จะทำการดึงส่วนประกอบของเว็บเพจที่เป็น Java Applet มาจากเว็บซิร์ฟเวอร์ก่อนถึงทำการประมวลผลเพื่อแสดงผล Java Applet นั้น การประมวลผล Java Applet นั้นจะต้องมีการเรียกใช้งาน ตัวแปลภาษา Java ขึ้นมาแปลแล้วถึงจะสามารถแสดงผลได้ แม้ว่า Java Applet นั้นไม่สามารถที่จะลบข้อมูลบนเครื่องของผู้ใช้ได้ แต่ Java Applet ก็มีความสามารถในการที่จะเขียน Java Applet ให้มีการสื่อสารบนเครือข่ายได้ ดังนั้นก็เป็นช่องทางหนึ่งในการที่ทำให้การความเสี่ยงในด้านความปลอดภัยได้
- JAVASCRIPT : JAVASCRIPT จะไม่เหมือน Java เพราะสามารถเขียนขึ้นมาได้ง่ายกว่าและ Javascript เป็นส่วนเพิ่มเติมของ HTML ดังนั้นเว็บบราวเซอร์สามารถที่จะประมวลผล Javascript ได้โดยตรง ปัญหาของ Javascript คือ Javascript สามารถที่เรียกใช้งานคำสั่งบางอย่างโดยที่ผู้ใช้อาจจะไม่รู้เลยก็ได้ว่ามีการเรียกใช้ เช่น Javascript สามารถที่จะใช้ในการส่งอีเมล์ได้ และ Javascript สามารถที่เข้าถึงไฟล์ข้อมูลในเครื่องของผู้ใช้ได้ ดังนั้นก็สามารถเป็นช่องทางในการเกิดความเสี่ยงด้านความปลอดภัยได้
- ACTIVEX : ACTIVEX Control จะมีการทำงานคล้ายกับ Java Applet แต่การทำงานของ ActiveX นั้นจะแสดงหน้าต่างแสดงข้อมูลของ ActiveX นั้น ซึ่งเรียกว่ากลไก Certificate โดยการ Certificate นั้นจะมีข้อมูลส่วนของผู้ใช้เขียน ActiveX นั้นแสดงขึ้นมาแสดงด้วย แต่เพราะ ActiveX มีความสามารถที่เหมือนกัน Java applet ดังนั้นก็ทำให้เกิดปัญหาในรูปแบบเดียวกัน

จากปัญหาในด้านความปลอดภัยที่เกิดขึ้นกับเว็บบราวเซอร์ดังนั้นทางผู้พัฒนาเว็บบราวเซอร์จึงต้องมีการแก้ไขและปรับปรุงการทำงานของเว็บบราวเซอร์ให้มีความปลอดภัยมากยิ่งๆขึ้น

2.3.1.6 จัดการเกี่ยวกับคุกกี้ (COOKIES)

เว็บบราวเซอร์จะมีหน้าที่ในการเก็บคุกกี้ที่ได้รับจากเว็บไซต์แต่ละแห่ง ซึ่งรายละเอียดเกี่ยวกับคุกกี้ได้อธิบายโดยละเอียดในส่วนโปรโตคอล HTTP แล้วการเก็บคุกกี้นั้นเว็บบราวเซอร์ต่างๆ ก็มี

การเก็บที่แตกต่าง เช่น อาจมีการเก็บคุกกี้ทั้งหมดไว้ในไฟล์ข้อมูลเพียงไฟล์เดียวเลย หรือ อาจจะเก็บแยกเป็นไฟล์ๆสำหรับแต่ละเว็บไซต์ ผู้ใช้สามารถที่จะลบคุกกี้ที่มีอยู่ในเครื่องได้

2.3.1.7 สรุป

เว็บไคลเอนต์จะมีโปรแกรมเว็บบราวเซอร์ทำงานอยู่ โดยเว็บบราวเซอร์จะใช้ในการสร้างและส่งการร้องขอแบบ HTTP (HTTP Request Message) ของเว็บเพจที่ผู้ใช้ต้องการดูไปยังเว็บเซิร์ฟเวอร์ ต่อจากนั้นก็จะทำหน้าที่ในการแปลและแสดงผลข้อมูลเว็บเพจที่ได้รับจากข้อมูลการตอบกลับแบบ HTTP (HTTP Response Message) ที่ส่งกลับมาจากเว็บเซิร์ฟเวอร์ซึ่งในการแสดงผลนั้นอาจจะต้องมีการเรียกใช้งานโปรแกรมภายนอก (Helper Application) เพื่อประมวลผลในส่วนที่เว็บบราวเซอร์ไม่สามารถทำงานได้ การเข้าถึงเว็บไซต์ต่างๆ อาจจะเกิดปัญหาเกี่ยวกับความปลอดภัยของเครื่องที่ผู้ใช้ใช้งานอยู่หรือปัญหาเกี่ยวกับไวรัส ซึ่งปัญหานี้ส่วนมากอาจจะเกิดจาก Java Applet, ActiveX, Javascript ที่มีอยู่ในเว็บเพจได้

2.3.2 เว็บพร็อกซี่ (Web Proxy)

การทำงานรูปแบบไคลเอนต์-เซิร์ฟเวอร์นั้นจะมีการเชื่อมต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยตรงจะไม่มีกล่าวถึงเครื่องเซิร์ฟเวอร์ใดๆระหว่างไคลเอนต์และเซิร์ฟเวอร์เช่นเดียวกับการเข้าถึงเว็บเพจต่าง การร้องขอของผู้ใช้จะถูกส่งไปยังเว็บเซิร์ฟเวอร์โดยตรงและการตอบกลับจากเว็บเซิร์ฟเวอร์ก็จะถูกส่งกลับไปยังเครื่องของผู้ใช้โดยตรง ต่อในระยะเวลาต่อมาได้มีการนำเครื่องเซิร์ฟเวอร์ที่ทำงานเป็นตัวกลางในการเชื่อมต่อระหว่างเครื่องของผู้ใช้และเว็บเซิร์ฟเวอร์ ซึ่งเรียกเครื่องเซิร์ฟเวอร์ นี้ว่า "PROXY" และสำหรับการใช้พร็อกซี่ในเวิลด์ไวด์เว็บนั้นจะเรียกว่า "WEB PROXY"

เว็บพร็อกซี่นั้นจะเป็นเครื่องหรืออุปกรณ์ที่มีการทำงานอยู่ระหว่างเว็บไคลเอนต์และเว็บเซิร์ฟเวอร์โดยจะมีการทำงานคล้ายกับเป็นเว็บเซิร์ฟเวอร์เสมือน (Virtual Web Server) ให้กับเว็บไคลเอนต์ต้นทางเพราะจะทำหน้าที่ในการรับการร้องขอจากเว็บไคลเอนต์ต้นทางและในทางตรงข้ามก็จะทำงานคล้ายกับเป็นเว็บไคลเอนต์เสมือน (Virtual Web Client) ให้กับเว็บเซิร์ฟเวอร์ปลายทางด้วยเพราะจะทำการรับการตอบรับของเว็บเซิร์ฟเวอร์ปลายทางด้วย

จุดประสงค์ของการนำเว็บพร็อกซี่มาใช้ก็เพราะว่าเพื่อลดการเชื่อมต่อที่ไม่จำเป็นระหว่างเครื่องไคลเอนต์และเว็บเซิร์ฟเวอร์ อย่างเช่น นำมาทำงานเกี่ยวกับแคชชิ่งเว็บเพจต่างและส่งเว็บเพจนั้นกลับไปให้ผู้ใช้โดยที่อาจจะไม่ต้องมีการติดต่อกับเครื่องเว็บเซิร์ฟเวอร์เลยก็ได้ อีกทั้งยังช่วยลดระยะเวลาในการร้องขอเว็บเพจจากเว็บเซิร์ฟเวอร์ด้วย และยังลดความคับคั่งของเครือข่ายภายในที่เชื่อมต่อไปยังเครือข่ายภายนอกด้วย

2.3.2.1 เว็บพร็อกซี่ชนิดต่างๆ

รูปแบบการทำงานของพร็อกซี่แบบเป็น 4 แบบ คือ

- Regular Proxy : เป็นพร็อกซี่ที่มีการทำงานเพียงแค่รับ การร้องขอจากเว็บไคลเอนต์ที่เครื่องผู้ใช้และทำการส่งการร้องขอนั้นไปยังเว็บเซิร์ฟเวอร์ และในทางกลับกันการทำการรับการตอบกลับจาก เว็บเซิร์ฟเวอร์ และส่งการตอบกลับนั้นกลับไปยังผู้ใช้
- Caching Proxy : ความแตกต่างระหว่าง Regular Proxy และ Caching proxy คือ Caching proxy จะทำการเก็บข้อมูลการตอบกลับต่างๆที่ได้รับจากเว็บเซิร์ฟเวอร์ไว้ด้วย และเมื่อ Caching proxy ได้รับการร้องขอเว็บเพจหรือรีซอร์สที่มีอยู่ในพร็อกซี่เองก็สามารถทำการเอาเว็บเพจหรือรีซอร์สที่แคชชิ่งไว้ส่งกลับไปให้ผู้ใช้ได้เลย
- Transparent Proxy : การทำงานที่โดยจะไม่มีการแก้ไขข้อมูลต่างๆที่ผ่าน อีกทั้งยังสามารถทำงานแบบ Caching Proxy ได้ด้วย
- Non-transparent Proxy : จะมีการทำงานคล้ายกับ Transparent Proxy แต่ Non-Transparent Proxy จะสามารถแก้ไขข้อมูลที่วิ่งผ่านได้ โดยอาจจะมีการเพิ่มหรือนำข้อมูลบางอย่างออกจากการร้องขอของเว็บไคลเอนต์หรือการตอบกลับจากเว็บเซิร์ฟเวอร์ได้

2.3.2.2 การทำงานของเว็บพร็อกซี่

ในหัวข้อนี้จะกล่าวถึงฟังก์ชั่นการทำงานที่มีใน proxy

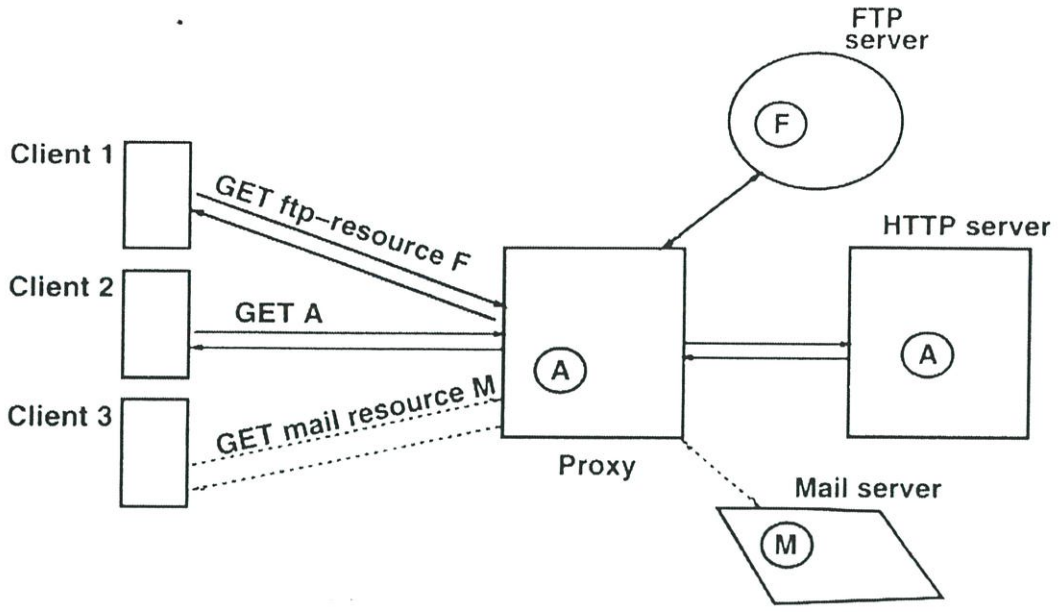
2.3.2.2.1 ความสามารถในการใช้การเชื่อมต่อไปยังเวิลด์ไวด์เว็บร่วมกัน

เว็บพร็อกซี่นั้นจะทำงานเสมือนเป็นเครื่องที่แทนเครื่องในเครือข่ายทั้งหมด และจะทำให้เครื่องต่างๆในเครือข่ายสามารถที่จะเข้าถึงเครือข่ายเวิลด์ไวด์เว็บได้ เว็บพร็อกซี่สามารถที่จะสร้างการเชื่อมต่อเพียงการเชื่อมต่อเดียวไปยังเว็บเซิร์ฟเวอร์เพื่อจัดการการร้องขอจากหลายๆผู้ใช้ได้ เพราะการร้องขอและการตอบรับต่างๆจะต้องผ่านที่เว็บพร็อกซี่ก่อน จากนั้นเว็บพร็อกซี่ค่อยทำการร้องขอไปยังเว็บเซิร์ฟเวอร์ต่อไป

2.3.2.2.2 แคชชิ่งข้อมูลการตอบกลับจากเว็บเซิร์ฟเวอร์

หนึ่งในการทำงานที่สำคัญของเว็บพร็อกซี่ก็คือการแคชชิ่งเว็บเพจหรือรีซอร์สที่ได้รับจากเว็บเซิร์ฟเวอร์ แล้วเว็บเพจหรือรีซอร์สนั้นกลับมาใช้กับผู้ใช้คนอื่นได้ เว็บพร็อกซี่จะทำการเว็บเพจหรือรีซอร์สที่ได้ทำการแคชไว้กลับไปให้ผู้ใช้เมื่อเว็บพร็อกซี่พิจารณาแล้วว่าเว็บเพจหรือรีซอร์สนั้นยังไม่ได้รับการแก้ไข(การทำงานตรงนี้ เว็บพร็อกซี่จะต้องมีการส่งการร้องเพื่อทำการ Cache Revalidation ที่มี Header เป็น If-Modified-Since ไปยังเว็บเซิร์ฟเวอร์ (ตามที่ได้อธิบายไว้ในหัวข้อ HTTP Protocol) นอกเหนือจากนั้นยังมีหน้าที่คือ

- ซ่อนข้อมูลของเว็บไคลเอนต์ในเครือข่าย (Anonymizing Clients) : เว็บพร็อกซี่จะทำหน้าที่อีกอย่างที่สำคัญคือ จะทำการปิดปังหรือซ่อนผู้ใช้ต่างๆของเครือข่ายภายในไม่ให้เครือข่ายภายนอกสามารถเห็นได้ เพราะการร้องขอและการตอบรับต่างๆจะต้องผ่านเว็บพร็อกซี่อย่างเดียวเท่านั้นถึงจะถูกส่งต่อไปยังผู้ใช้หรือเว็บเซิร์ฟเวอร์อีกที ดังนั้นเครื่องต่างๆในเครือข่ายภายนอกจะไม่รู้ได้เลยว่าเครื่องที่อยู่ในเครือข่ายนั้นมีกี่เครื่อง มีไอพีอะไรบ้าง
- แปลงหรือทำการแก้ไขการร้องขอหรือการตอบกลับบนโปรโตคอล HTTP (Transforming Requests and Responses) : เว็บพร็อกซี่อาจจะทำการแก้ไขข้อมูลการร้องขอแบบ HTTP หรือข้อมูลการตอบกลับแบบ HTTP ได้ในหลายๆกรณี เช่น เว็บ พร็อกซี่กับเว็บเซิร์ฟเวอร์อาจจะมีการทำงานที่ HTTP เวอร์ชั่นใหม่กว่าเว็บบราวเซอร์ของผู้ใช้ดังนั้นเว็บพร็อกซี่สามารถที่จะแปลงการร้องขอแบบ HTTP ของผู้ใช้ให้เป็นเวอร์ชั่นที่ใหม่ที่เว็บเซิร์ฟเวอร์รองรับแล้วส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์แล้วเมื่อเว็บเซิร์ฟเวอร์ตอบกลับก็ทำการแปลงเป็น HTTP เวอร์ชั่นที่เว็บบราวเซอร์ของผู้ใช้รองรับอีกทีหนึ่ง ที่ต้องมีการแปลงเพราะการร้องขอหรือการตอบกลับอาจจะใช้ประโยชน์จาก HTTP เวอร์ชั่นที่ใหม่กว่าได้เช่นการใช้ Header "If-Modified-Since" เป็นต้น หรือ เว็บพร็อกซี่กับเว็บเซิร์ฟเวอร์อาจจะมีการบีบอัดข้อมูลระหว่างกัน ซึ่งการทำงานตรงนี้ผู้ใช้จะไม่เกี่ยวข้องเลยเพราะเมื่อเว็บพร็อกซี่ได้รับการการร้องขอจากเว็บเซิร์ฟเวอร์แล้วเว็บพร็อกซี่ก็จะทำการขยายข้อมูลที่บีบอัดและส่งเป็นการตอบกลับแล้วส่งกลับไปให้ผู้ใช้ได้ โดยที่ผู้ใช้ไม่รู้เลยว่ามีการบีบอัดข้อมูลเกิดขึ้น
- เป็นประตูสู่เครือข่ายภายนอก (Gateway to non-HTTP system) : เว็บพร็อกซี่สามารถที่จะมีบทบาทในการเชื่อมต่อระหว่างระบบอื่นๆที่ไม่ได้มีการติดต่อกันโดยใช้โปรโตคอล HTTP เช่น เว็บพร็อกซี่อาจจะเป็นตัวกลางระหว่างเว็บไคลเอนต์และเอฟทีพีเซิร์ฟเวอร์ (FTP Server) ซึ่งการร้องของขอผู้ใช้จะถูกส่งไปยังเอฟทีพีเซิร์ฟเวอร์ (FTP Server) และการตอบกลับจากเอฟทีพีเซิร์ฟเวอร์ (FTP Server) ก็จะถูกส่งกลับไปยังผู้ใช้ แต่เพราะเว็บไคลเอนต์และเอฟทีพีเซิร์ฟเวอร์ใช้โปรโตคอลที่แตกต่างกันในการติดต่อ ดังนั้นเว็บพร็อกซี่ต้องทำการแปลงการร้องขอแบบ HTTP ที่มาจากเว็บไคลเอนต์ให้เป็นการร้องขอแบบ FTP ก่อน และจะต้องแปลงการตอบกลับแบบ FTP ให้เป็นการตอบกลับแบบ HTTP ก่อนส่งกลับไปยังเว็บไคลเอนต์ดังกรณีนี้ เว็บพร็อกซี่จะทำหน้าที่เหมือนกับเป็น "ประตูสู่เครือข่ายภายนอก (Gateway)" ด้วย



รูปที่ 2.13 ตัวอย่างที่พร็อกซี่มีการทำงานเป็นประตูสู่เครือข่ายภายนอก (Gateway) ไปยัง FTP, HTTP และ Mail

ตัวอย่างของการที่เว็บพร็อกซี่ทำตัวเป็นประตูสู่เครือข่ายภายนอก (Gateway) เช่น สมมติเว็บไคลเอนต์มีการร้องขอไปยังเอฟทีพีเซิร์ฟเวอร์ (FTP Server) โดยร้องขอ ftp://ftp.kmitl.ac.th/F ไปยังเว็บพร็อกซี่เมื่อเว็บพร็อกซี่ได้รับการร้องขอแล้ว เว็บพร็อกซี่ก็จะทำการแปลงการร้องขอแบบ HTTP ของเว็บไคลเอนต์ไปเป็นการร้องขอแบบ FTP แล้วถึงทำการร้องขอไปยังเอฟทีพีเซิร์ฟเวอร์ (FTP Server) ต่อไป แต่อย่างไรก็ดีในกรณีนี้เว็บพร็อกซี่จะต้องทำการสร้างการเชื่อมต่อไปยัง เอฟทีพีเซิร์ฟเวอร์ (FTP Server) และเว็บพร็อกซี่จะทำงานคล้ายเป็นเอฟทีพีเซิร์ฟเวอร์ (FTP Client) เลย นอกเหนือจากนั้นการทำงานของโปรโตคอล FTP จะมีการส่งข้อมูลกับการร้องขอจะส่งโดยใช้การเชื่อมต่อที่แตกต่างกัน(ข้อมูลทาง port 20, การร้องขอทาง port 21) ดังนั้นเว็บพร็อกซี่ต้องทำการดึงข้อมูลจากช่องการเชื่อมต่อของข้อมูลในการตอบกลับของเอฟทีพีเซิร์ฟเวอร์ (FTP Server) แล้วทำการแปลงเป็นการตอบกลับแบบ HTTP ส่งกลับไปเว็บไคลเอนต์ต่อไป

2.3.2.2.3 กรองและตรวจสอบการร้องขอจากเว็บไคลเอนต์หรือการตอบกลับจากเว็บเซิร์ฟเวอร์

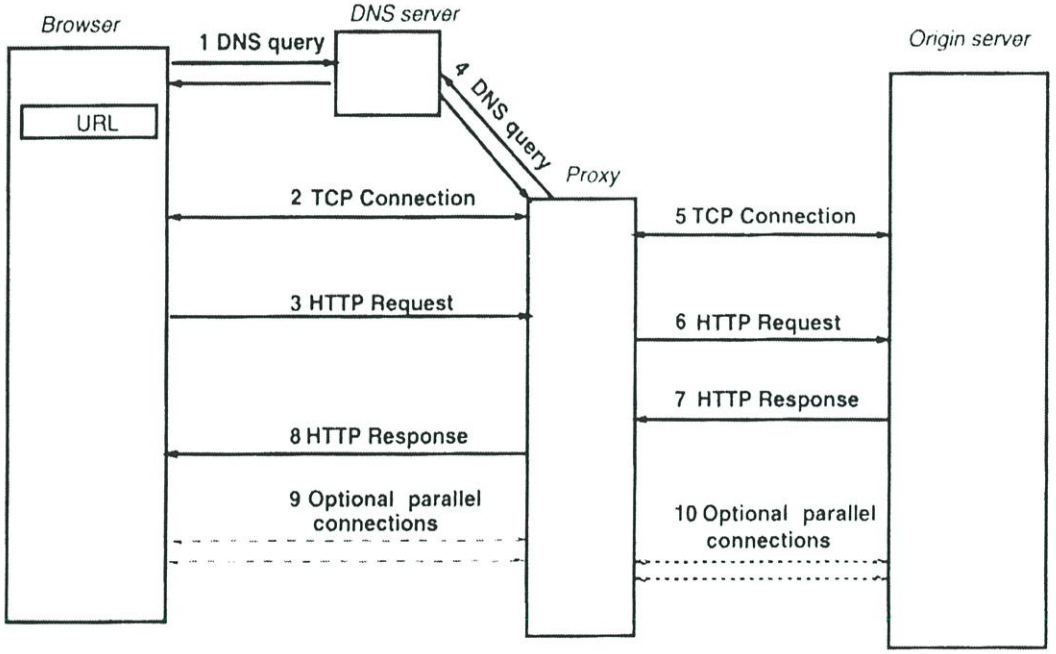
เว็บพร็อกซี่สามารถทำหน้าที่ในการกรองหรือป้องกันการร้องขอแบบ HTTP หรือการตอบกลับแบบ HTTP ที่ไม่ถูกต้อง โดยการกรอง (Filter) ของเว็บพร็อกซี่นั้นอาจจะกรองโดยการดูจากเว็บไซต์ปลายทางว่าถ้าเป็นเว็บไซต์นี้จะไม่ให้มีการร้องขอเกิดขึ้น หรือจะเป็นการกรองในกรณีของการตอบกลับแบบ HTTP ว่าถ้ามีขนาดหรือมีรูปแบบของข้อมูลเป็นตามเงื่อนไขใดเงื่อนไขหนึ่งก็ไม่ให้ทำการส่งการตอบกลับแบบ HTTP นั้นกลับไปให้ผู้ใช้ได้ นอกเหนือจากนี้การกรองของเว็บ

พร็อกซี่ยังรวมไปถึงการนำหรือการใส่ข้อมูลเพิ่มลงไปในข้อมูลของการร้องขอแบบ HTTP หรือ – ข้อมูลการตอบกลับแบบ HTTP ได้ด้วย

2.3.2.3 บทบาทของเว็บพร็อกซี่เกี่ยวกับการจัดการการร้องขอและการตอบรับการร้องขอ

ในหัวข้อนี้จะกล่าวถึงการทำงานโดยรายละเอียดในการที่เว็บพร็อกซี่จะจัดการกับการร้องขอแบบ HTTP ที่ส่งมาจากผู้ใช้และการตอบกลับแบบ HTTP ที่ส่งมาจากเว็บเซิร์ฟเวอร์

2.3.2.3.1 ขั้นตอนในการแลกเปลี่ยนของข้อมูลการร้องขอแบบ HTTP และข้อมูลการตอบกลับแบบ HTTP ของเว็บพร็อกซี่



รูปที่ 2.14 ขั้นตอนที่เกิดขึ้นในกระบวนการร้องขอและตอบกลับแบบ HTTP ที่มีการใช้งานเว็บพร็อกซี่

ในรูปที่ 2.14 จะแสดงขั้นตอนที่เกี่ยวข้องทั้งหมดเมื่อมีการใช้งานเว็บพร็อกซี่ในการเป็นตัวกลางระหว่างเว็บไคลเอนต์และเว็บเซิร์ฟเวอร์

เมื่อผู้ใช้ทำการร้องขอเว็บเพจหรือรีซอร์สใดๆ เว็บบราวเซอร์ของผู้ใช้จะมีการสร้างการเชื่อมต่อ (TCP Connection) ไปยังเว็บพร็อกซี่นั่นก็เพราะมีการกำหนดให้เว็บบราวเซอร์ของผู้ใช้มีการสร้างการเชื่อมต่อผ่านเว็บพร็อกซี่ ซึ่งเว็บบราวเซอร์อาจจะต้องทำการค้นหาไอพีของเว็บพร็อกซี่ด้วย

ดังนั้นเว็บราวเซอร์ต้องทำการส่งข้อมูลเพื่อสอบถามไอพีของพร็อกซี (DNS Query) ไปยังดีเอ็นเอสเซิร์ฟเวอร์ (DNS Server) ของผู้ใช้ด้วย (ขั้นตอนที่ 1 และ 2) จากนั้นเมื่อรู้ไอพีของเว็บพร็อกซี่แล้ว เว็บบราวเซอร์ก็จะทำการส่งการร้องขอแบบ HTTP ไปยังเว็บพร็อกซี่ (ขั้นตอนที่ 3) เมื่อเว็บพร็อกซี่ได้รับการร้องขอแบบ HTTP จากผู้ใช้แล้ว เว็บพร็อกซี่ก็จะต้องทำการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ปลายทางโดยเว็บพร็อกซี่จะต้องทำการค้นหาไอพีของเว็บเซิร์ฟเวอร์ปลายทางเช่นกันโดยจะทำการส่งข้อมูลเพื่อสอบถามไอพีของพร็อกซี (DNS Query)ไปยังดีเอ็นเอสเซิร์ฟเวอร์ (DNS Server) (ขั้นตอนที่ 4) ซึ่งดีเอ็นเอสเซิร์ฟเวอร์ของเว็บพร็อกซี่อาจจะต้องตัวเดียวหรือคนละตัวกับผู้ใช้ก็ได้ เมื่อรู้ไอพีของเว็บเซิร์ฟเวอร์ปลายทางแล้วเว็บพร็อกซี่ก็จะทำการสร้างการเชื่อมต่อ(TCP Connection) ไปยังเว็บเซิร์ฟเวอร์ปลายทางและทำการส่งการร้องขอแบบ HTTP ไปยังเว็บเซิร์ฟเวอร์ (ขั้นตอนที่ 5 และ 6) จากนั้นเมื่อเว็บเซิร์ฟเวอร์ส่งการตอบกลับแบบ HTTP กลับมายังเว็บพร็อกซี่ เว็บพร็อกซี่ก็จะทำการส่งต่อข้อมูลการตอบกลับแบบ HTTP นั้นกลับไปให้เว็บบราวเซอร์ของผู้ใช้ต่อไป ซึ่งระหว่างกระบวนการทั้งหมดนี้เว็บบราวเซอร์อาจจะมีการสร้างการร้องขอเพิ่มเติมไปยังเว็บพร็อกซี่ก็ได้ และเว็บพร็อกซี่ก็ต้องทำการร้องขอเว็บเพจหรือรีซอร์สต่างๆเพิ่มเติมตามที่ได้รับการร้องขอจากผู้ใช้ด้วย(ขั้นตอนที่ 9 และ 10)

2.3.2.3.2 การจัดการข้อมูลของการร้องขอแบบ HTTP และการตอบกลับแบบ HTTP

ตามที่ได้กล่าวมาแล้วว่าเว็บพร็อกซี่อาจจะมีการเพิ่มหรือลบข้อมูลบางอย่างไปในข้อมูลของการร้องขอแบบ HTTP หรือข้อมูลการตอบกลับแบบ HTTP ได้ซึ่งการเพิ่มหรือการลบต่างๆที่เกิดขึ้นอาจจะเป็นเกิดเพราะกรณีดังนี้

- แสดงตัวเองของเว็บพร็อกซี่ (Identification) : เว็บพร็อกซี่สามารถที่จะแสดงตัวเองใน HTTP Header ได้เพื่อให้เซิร์ฟเวอร์หรือส่วนประกอบอื่นๆเช่น เราเตอร์สามารถที่จะรู้ได้ว่าการร้องขอแบบ HTTP หรือการตอบกลับแบบ HTTP นี้ถูกส่งมาจากเว็บพร็อกซี่นี้ หนึ่งในจุดประสงค์ที่ทำการเพิ่ม Header นี้ก็เพราะใช้ในการแสดงว่ามีการเกิดลูปในการส่งการร้องขอแบบ HTTP หรือการตอบกลับแบบ HTTP หรือไม่
- แปลงเวอร์ชั่นของ HTTP (Version number change) : เว็บพร็อกซี่สามารถที่จะเปลี่ยน Request Line ในข้อมูลการร้องขอแบบ HTTP ได้เช่น HTTP Request Line ที่ถูกส่งมาจากผู้ใช้อาจจะเป็น

GET /foo/bar.html HTTP/1.0

อาจจะถูกแก้ไขเป็น

GET /foo/bar.html HTTP/1.1

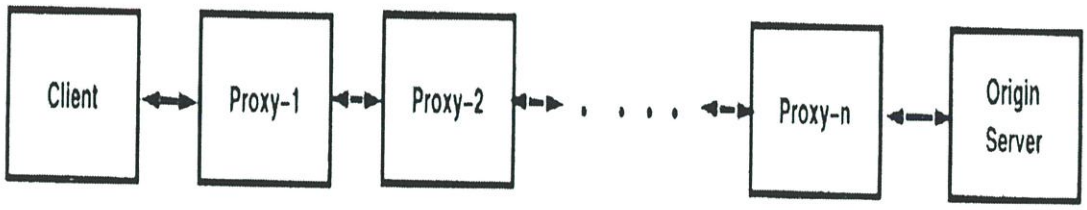
ก่อนที่จะถูกส่งไปยังเว็บเซิร์ฟเวอร์ เพราะเว็บพร็อกซี่นั้นอาจจะต้องการทำงานหรือใช้ความสามารถของ HTTP ในเวอร์ชั่นที่ใหม่กว่า ซึ่งเว็บบราวเซอร์ของผู้ใช้ไม่สนันสนุนได้ และในกรณีที่เว็บบราวเซอร์ของผู้ใช้ทำงานด้วย HTTP เวอร์ชั่นใหม่กว่าและก็เช่นเดียวกันเว็บพร็อกซี่ก็สามารถแปลงเป็น HTTP เวอร์ชั่นที่เว็บเซิร์ฟเวอร์สนับสนุนได้

- เพิ่มข้อมูลบางอย่างเกี่ยวกับการตอบกลับที่ส่งกลับไป (Adding obligatory information about the resource) : ในการส่งการตอบกลับแบบ HTTP กลับไปยังผู้ใช้โดยใช้เว็บเพจหรือรีซอร์สที่ทำการแคชไว้นั้น เว็บพร็อกซี่อาจจะต้องการเพิ่มข้อมูลบางอย่างเกี่ยวกับเว็บเพจหรือรีซอร์สนั้นเข้าไป เช่น มีการเพิ่มข้อมูลบอกช่วงระยะเวลาที่เว็บเพจหรือรีซอร์สนั้นมีการตรวจสอบการแก้ไขกับเว็บเซิร์ฟเวอร์
- จัดการเกี่ยวกับความล่าช้าและการสำรองข้อมูล (Handling delay and Buffering) : เว็บพร็อกซี่นั้นจะทำงานโดยเป็นตัวกลางระหว่างเว็บไคลเอนต์และเว็บเซิร์ฟเวอร์ นอกจากนั้นยังจัดเก็บเว็บเพจหรือรีซอร์สต่างๆที่ไดมีการร้องขอไว้ในแคชแล้วยังทำให้หน้าที่ในการส่งเว็บเพจหรือรีซอร์สที่ได้มีการแคชไว้ส่งกลับไปให้ทางผู้ใช้ด้วย
- รักษาสถานะและข้อกำหนดต่างของการเชื่อมต่อ (State maintenance and policy issue) : เว็บพร็อกซี่จะต้องมีการรักษาและดูแลสถานะของแต่ล่ะการเชื่อมต่อที่เข้ามาที่เว็บ พร็อกซี่และการเชื่อมต่อที่เว็บพร็อกซี่ทำการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ด้วยเช่น การปิดการเชื่อมต่อกับบราวเซอร์ของผู้ใช้ในกรณีการมีการยกเลิกการร้องขอเว็บเพจจากเว็บเซิร์ฟเวอร์ แต่สำหรับการเชื่อมต่อกับเว็บเซิร์ฟเวอร์นั้นเว็บพร็อกซี่อาจจะไม่ทำการยกเลิกการเชื่อมต่อถ้าผู้ใช้ยกเลิกการเชื่อมต่อ เพราะเว็บพร็อกซี่อาจจะต้องการแคชเว็บเพจหรือรีซอร์สนั้นไว้ก่อนซึ่งก็ขึ้นอยู่กับนโยบายที่กำหนดไว้ให้กับเว็บพร็อกซี่ด้วย
- จัดการเกี่ยวกับคุกกี้ (Handling cookies) : อย่างที่ได้กล่าวไปแล้วว่าเว็บไคลเอนต์จะต้องมีการใช้งาน คุกกี้ (Cookies) ด้วย ซึ่งคุกกี้นั้นจะถูกส่งมาจากเว็บเซิร์ฟเวอร์และเพราะเว็บพร็อกซี่นั้นจะทำงานคล้ายกับเป็นทั้งเว็บไคลเอนต์และเว็บเซิร์ฟเวอร์เพราะเว็บพร็อกซี่นั้นจะได้รับทั้งคุกกี้ที่อยู่ในข้อมูลการร้องขอแบบ HTTP และคุกกี้ที่ได้รับจากการตอบกลับแบบ HTTP ซึ่งเว็บพร็อกซี่จะต้องทำการส่งต่อไปยังเว็บเซิร์ฟเวอร์หรือเว็บไคลเอนต์ต่อไป ตัวอย่างเช่น เว็บไคลเอนต์ทำการร้องขอแบบ HTTP ไปยังเว็บเซิร์ฟเวอร์ S โดยผ่านเว็บพร็อกซี่ P จากนั้นเว็บเซิร์ฟเวอร์ S ก็ได้ส่งคุกกี้ โดยการใช้ Set-Cookie Header ในการตอบกลับแบบ HTTP กลับไปให้เว็บพร็อกซี่เพราะเว็บ พร็อกซี่จะถูกมองว่าเป็นไคลเอนต์ เมื่อเว็บพร็อกซี่ได้รับข้อมูลการตอบกลับแบบ HTTP ที่มี Set-Cookie แล้วก็ทำการส่งการตอบกลับแบบ HTTP นี้กลับไปให้เว็บไคลเอนต์ต่อไป และครั้งต่อไปที่เว็บไคลเอนต์ต้นทางทำการร้องขอเว็บเพจ

หรือรีซอร์ส จากเว็บเซิร์ฟเวอร์ S อีกก็จะทำการส่งคุกกี้ที่ได้เก็บไว้เว็บเซิร์ฟเวอร์ S โดยผ่านเว็บ พร็อกซี่ P และเว็บพร็อกซี่ P ก็จะทำการส่งการร้องขอแบบ HTTP ที่มีคุกกี้นี้ต่อไปให้เว็บเซิร์ฟเวอร์ต่อไป ซึ่งเว็บพร็อกซี่ P อาจจะมีการเพิ่มเอาคุกกี้ของตัวเองเข้าไปได้ด้วย ไม่ว่าจะเป็นการกำหนดให้เว็บไคลเอนต์เก็บคกุกี้ของตัวเอง(คุกกี้ของพร็อกซี่)หรือส่ง คุกกี้ที่ตัวเองเก็บไว้ส่งให้กับเว็บเซิร์ฟเวอร์

2.3.2.4 ลำดับชั้นและการเชื่อมต่อเป็นลูกโซ่ของเว็บพร็อกซี่

ในการทำงานจริงๆของเว็บพร็อกซี่นั้นอาจจะมีการทำงานของเว็บพร็อกซี่หลายตัวระหว่าง เว็บไคลเอนต์ และเว็บเซิร์ฟเวอร์ ตัวอย่างเช่นสมมติว่า มหาวิทยาลัยหนึ่งมีเว็บพร็อกซี่ทำงานอยู่หนึ่งตัว และแต่ละคณะก็สามารถที่จะมีเว็บพร็อกซี่เป็นของตัวเองในเครือข่ายของแต่ละคณะได้อีก และมหาวิทยาลัยนี้อาจจะเชื่อมต่อกับไอเอสพี (ISP: Internet Service Provide) ซึ่งอาจจะมีเว็บพร็อกซี่ของไอเอสพีอีก และท้ายที่สุดที่เว็บเซิร์ฟเวอร์ปลายทางอาจจะมีเว็บพร็อกซี่อยู่อีกได้เช่นกัน ดังนั้นการทำงานจะเป็นดังรูป



รูปที่ 2.15 แสดงรูปแบบการทำงานอย่างง่ายๆ เป็นลำดับของพร็อกซี่

ดังนั้นโดยสรุปการร้องขอแบบ HTTP ของเว็บไคลเอนต์นั้นอาจจะผ่านเว็บพร็อกซี่หลายๆ ตัวกว่าจะถึงเว็บเซิร์ฟเวอร์ปลายทางและในความเป็นจริงอาจจะไม่ทำงานเป็นเส้นตรงดังรูปก็เป็นได้ โดยอาจจะมีการการติดต่อของเว็บพร็อกซี่เป็นลำดับขั้นได้เช่นกัน เช่น เว็บพร็อกซี่ของมหาวิทยาลัยหลายแห่งอาจจะเชื่อมต่อกับเว็บพร็อกซี่ของไอเอสพีและเว็บพร็อกซี่ของไอเอสพีทั้งหมดของประเทศหนึ่งอาจจะเชื่อมต่อกับเว็บพร็อกซี่ของทวีปอีกชั้นหนึ่งก็ได้

2.3.2.5 สรุป

เว็บพร็อกซี่เป็นเครื่องหรืออุปกรณ์ที่มีการทำงานระหว่างเว็บไคลเอนต์และเว็บเซิร์ฟเวอร์ ซึ่งเว็บพร็อกซี่มีความสามารถในการทำให้ผู้ใช้ต่างๆในเครือข่ายขององค์กรหนึ่งสามารถที่เข้าถึงเครือข่ายเวิลด์ไวด์เว็บร่วมกันได้ เว็บพร็อกซี่สามารถทำหน้าที่ในการกรองการร้องขอแบบ HTTP

หรือการตอบกลับแบบ HTTP ที่ผ่านได้ และเว็บพร็อกซี่ยังทำหน้าที่ในการแคชเว็บเพจหรือรีซอร์สไว้ใช้กับผู้ใช้งานคนอื่นๆได้ การนำเว็บพร็อกซี่มาใช้ยังมีประโยชน์ในการช่วยลดระยะเวลาในการเข้าถึงเว็บเพจหรือรีซอร์สต่างด้วย

2.3.3 เว็บเซิร์ฟเวอร์ (Web server)

เว็บเซิร์ฟเวอร์เป็นโปรแกรมที่จะใช้ในการสร้างและส่งการตอบกลับแบบ HTTP กลับไปยังผู้ใช้ที่มีการร้องขอเว็บเพจหรือรีซอร์สใดบนเว็บเซิร์ฟเวอร์

2.3.3.1 ความแตกต่างระหว่างเว็บไซต์และเว็บเซิร์ฟเวอร์

เว็บไซต์จะประกอบด้วยกลุ่มของเว็บเพจที่อยู่ภายใต้ชื่อเฉพาะชื่อหนึ่ง แต่เว็บเซิร์ฟเวอร์จะเป็นโปรแกรมที่จะใช้ในการติดต่อสื่อสารกับเว็บไคลเอนต์ที่เครื่องของผู้ใช้ที่มีการร้องขอเว็บเพจหรือรีซอร์สใดบนเว็บไซต์ ซึ่งในหัวข้อนี้จะอธิบายลักษณะการทำงานแบบต่างๆของเว็บไซต์ รูปแบบการทำงานของเว็บเซิร์ฟเวอร์ และการจำแนกรูปแบบของเว็บเซิร์ฟเวอร์ แบบต่างๆ

2.3.3.1.1 เว็บไซต์

รูปแบบต่างของเว็บไซต์ในปัจจุบันสามารถแบ่งง่ายๆเป็น

- เว็บไซต์สถาบันการศึกษา (University) : เว็บไซต์แบบนี้ส่วนใหญ่จะเป็นเว็บไซต์ที่มีการให้ข้อมูลเกี่ยวกับหลักสูตรการศึกษาหรือข้อมูลการวิจัยรวมถึงรูปและข้อมูลที่ใช้ในการติดต่อกับทางสถาบันการศึกษา ในเว็บไซต์อาจจะมีลิงค์ไปยังเว็บไซต์ของแต่ละส่วนการศึกษาที่แสดงรายชื่อของนักศึกษาและอาจารย์ผู้สอนรวมถึงโฮมเพจในแต่ล่ะส่วนการศึกษานั้นๆด้วย
- เว็บไซต์ในรูปแบบการให้ข้อมูลจากธุรกิจถึงผู้บริโภค (Business-to-consumer) : บริษัททั่วไปอาจจะมีการสร้างเว็บไซต์ที่ใช้ในการให้ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ขายหรือจะบอกตำแหน่งงานที่ว่างอยู่ ภายในเว็บไซต์อาจจะให้ลูกค้าสามารถที่จะดูข้อมูลรายละเอียดของผลิตภัณฑ์ ราคา ข้อมูลการสั่งซื้อ อาจจะรวมถึงสามารถดูข้อมูลสินค้าที่มีอยู่ในคลังสินค้าได้ด้วย ลูกค้าอาจจะสามารถค้นหาข้อมูลได้ด้วย ลักษณะของเว็บไซต์ที่มีการทำงานอย่างนี้จะเรียกว่าเว็บไซต์แบบ "B2C"
- เว็บไซต์ภายในขององค์กร (Corporate intranet) : บริษัทในปัจจุบันหลายบริษัทได้มีการสร้างเว็บไซต์ที่ใช้เฉพาะภายในองค์กรเท่านั้นที่เรียกว่า "อินทราเน็ตเว็บไซต์" โดยเว็บไซต์นี้จะให้ข้อมูลต่างๆที่จำเป็นต้องใช้ในการทำงานกับพนักงานในองค์กร อีกทั้งยังใช้ในการติดต่อระหว่างพนักงานในองค์กรเอง การเข้าถึงเว็บไซต์นี้จะต้องมีการใส่ บัญชีรายชื่อ (Username) และรหัสผ่าน (Password) ก่อน
- เว็บไซต์จากกลุ่มธุรกิจหนึ่งไปยังอีกกลุ่มธุรกิจหนึ่ง (Business-to-busines)s : การทำงานในบริษัทหนึ่งอาจจะต้องมีการขึ้นกับอีกบริษัทหนึ่ง ดังนั้นบริษัทหนึ่งอาจจะมีการสร้างเว็บไซต์

ที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างทั้งสองบริษัทขึ้นมาเพื่อให้อีกบริษัทหนึ่งสามารถที่จะดึงหรือค้นหาข้อมูลของอีกบริษัทหนึ่งได้ ลักษณะของเว็บไซต์ที่มีการทำงานอย่างนี้จะเรียกว่าเว็บไซต์แบบ “B2B”

- เว็บไซต์เหตุการณ์เฉพาะ (Special Event) : บางเว็บไซต์ประเภทนี้จะเป็นเว็บไซต์แบบที่มีการตั้งขึ้นชั่วคราวเฉพาะในเวลาหนึ่งๆ เช่นเว็บไซต์มหกรรมกีฬาต่างๆ ภายในไซต์จะแสดงข้อมูลการรายงานต่างๆของเหตุการณ์นั้น โดยข้อมูลจะมีการเปลี่ยนแปลงอย่างรวดเร็วตามเหตุการณ์นั้นๆ เว็บไซต์แบบนี้จะมีผู้ใช้เข้าถึงเป็นจำนวนมากในช่วงระยะเวลาหนึ่ง
- เว็บท่า (Portal) : เว็บไซต์ท่าจะถูกสร้างขึ้นเพื่อใช้ในการสร้างความสัมพันธ์กับผู้ใช้งานที่จะเป็นศูนย์กลางข้อมูลทุกอย่างที่ผู้ใช้ต้องการจะเข้าถึง เช่น เว็บไซต์แบบ Portal อาจมีการแสดงข้อมูลภูมิอากาศ รายการโทรทัศน์ ข่าว ซึ่งเว็บไซต์แบบนี้จะสามารถให้ผู้ใช้งานสามารถที่ปรับเปลี่ยนข้อมูลที่จะแสดงได้ เช่นผู้ใช้อาจจะต้องการดูข้อมูลเกี่ยวภาพยนต์ที่เข้าฉายอยู่ เป็นต้น ในเว็บไซต์จะมีการจัดการให้มีลิงค์ไปยังเว็บไซต์ต่างๆที่ผู้ใช้สนใจเพื่อให้ง่ายในการเข้าถึงของผู้ใช้งาน
- เว็บไซต์ที่ใช้ค้นหาข้อมูลในเครือข่ายอินเตอร์เน็ต (Search) : เว็บไซต์แบบนี้จะเป็นเว็บไซต์ที่จะให้ผู้ใช้งานสามารถที่จะค้นหาข้อมูลที่ผู้ใช้สนใจจากเว็บไซต์ๆ โดยผู้ใช้จะต้องใส่คำที่ต้องการจะหา (Keyword) หรือข้อมูลที่ต้องการค้นหาแล้วทางเว็บไซต์จะทำการค้นหาให้แสดงรายการเว็บไซต์ที่มีข้อมูลที่อาจจะตรงหรือใกล้เคียงกับข้อมูลที่ผู้ใช้ต้องการจะค้นหาซึ่งความถูกต้องของการค้นหานั้นก็ขึ้นการรูปแบบหรือเทคโนโลยีที่เว็บไซต์ที่ให้บริการค้นหาข้อมูลใช้
- เว็บไซต์ที่เป็นประตูสู่บริการด้านอื่น (Gateway to other services) : เว็บไซต์หลายๆแห่งจะให้บริการอื่นเช่น กลุ่มข่าว (Newsgroups), จดหมายอิเลคทรอนิคส์ (Email), ห้องสนทนา (Chatroom) เช่นเว็บไซต์สำหรับกลุ่มข่าวเกี่ยวกับระบบเครือข่ายก็อาจจะให้ผู้ใช้งานสามารถที่จะลงทะเบียนเป็นสมาชิกเพื่อรับข้อมูลข่าวสารเกี่ยวกับระบบเครือข่ายได้

2.3.3.1.2 เว็บเซิร์ฟเวอร์

เว็บเซิร์ฟเวอร์เป็นโปรแกรมที่ใช้ในการจัดการการร้องขอแบบ HTTP ที่ถูกส่งมาจากผู้ร้องขอเว็บเพจหรือรีซอร์สบนเว็บไซต์ โดยเว็บเซิร์ฟเวอร์จะต้องมีการเชื่อมต่อกับระบบเครือข่ายและจะต้องมี Memory, Harddisk, I/O Device รวมถึงโปรแกรมที่ใช้ในการจัดการการร้องขอแบบ HTTP ซึ่งเว็บเซิร์ฟเวอร์จะต้องมีความสามารถในการรองรับการร้องขอแบบ HTTP จำนวนมากจากผู้ใช้ต่างๆได้ ซึ่งจะรองรับได้เพียงใดก็ขึ้นกับโปรแกรมและเครื่องที่ใช้ ภายในเว็บเซิร์ฟเวอร์อาจจะมีการทำงานบริการอื่นๆได้เช่น Email, FTP

2.3.3.2 การจัดการการร้องขอจากผู้ใช้

2.3.3.2.1 การจัดการกับการร้องขอของผู้ใช้งาน

- อ่านและแปลงข้อมูลการร้องแบบ HTTP : เว็บเซิร์ฟเวอร์จะอ่านข้อมูลการร้องขอใน การร้องขอแบบ HTTP ที่ส่งมา โดย Header Control จะประกอบด้วย Request LINE เช่น GET และ URL ที่ทำการร้องขอเช่น www.it.kmitl.ac.th อีกทั้งจะทำการดึงเอาข้อมูลของการร้องขออื่นๆออกมาจากข้อมูลการร้องขอแบบ HTTP ที่ส่งมาด้วย เช่น Cookies, If-Modified-Since เป็นต้น
- ทำการแปลง URL ให้เป็นในรูปของชื่อของไฟล์ : เว็บเซิร์ฟเวอร์จะทำการแปลง URL ให้เป็นชื่อของไฟล์ที่มีอยู่ในเครื่องเว็บเซิร์ฟเวอร์โดยการแปลงอาจจะทำการแปลงแบบตรงๆเลย เช่น Directory เริ่มต้นของเว็บไซต์ www.it.kmitl.ac.th คือ C:\IT ถ้ามีการร้องขอ www.it.kmitl.ac.th/index.html ก็จะทำการแปลงเป็นการร้องของ C:\IT\index.html
- พิจารณาว่าการร้องขอนั้นจำเป็นต้องมีการตรวจสอบสิทธิของผู้ใช้หรือไม่ (authorized): ก่อนที่จะทำการส่งการตอบกลับแบบ HTTP กลับไปให้ผู้ใช้ เว็บเซิร์ฟเวอร์จะต้องมีการตรวจสอบก่อนว่าผู้ใช้งานนั้นๆมีสิทธิในการเข้าถึงหรือร้องขอเว็บเพจหรือรีซอร์สนั้นๆ หรือไม่ ถ้ามีเว็บเซิร์ฟเวอร์จะต้องมีการส่งการตอบกลับแบบ HTTP ที่มี Header บอกทางเว็บไคลเอนต์เพื่อให้มีการแสดงกล่องข้อความว่าให้ผู้ใช้งานแสดงชื่อผู้ใช้งาน (Username) แ ละรหัสผ่านก่อน (Password) ของผู้ที่สิทธิใช้งานก่อน
- ส่งข้อมูลของเว็บเพจหรือรีซอร์สนั้นๆกลับไปให้ผู้ใช้ : เว็บเซิร์ฟเวอร์จะทำการสร้าง –ข้อมูลการตอบกลับแบบ HTTP ที่จะมีข้อมูล Status Code ของการตอบกลับแบบ HTTP รวมถึงข้อมูลของเว็บเพจหรือรีซอร์สนั้นส่งกลับไปให้ยังผู้ใช้

ในการจัดการการร้องขอบางครั้งเว็บเซิร์ฟเวอร์อาจจะต้องมีการเก็บบันทึกข้อมูลของการร้องขอแบบ HTTP หรือการตอบกลับแบบ HTTP ไว้ด้วย เช่น เว็บเซิร์ฟเวอร์อาจจะมีการเก็บไอพีแอดเดรสของผู้ใช้, URL ที่ผู้ใช้ร้องขอ, เวลาที่มีการร้องขอไว้ใช้ไฟล์ไฟล์หนึ่งที่เรียกว่า "ล็อกไฟล์ (Log File)"

ในข้อมูลการตอบกลับแบบ HTTP ที่ถูกส่งจากเว็บเซิร์ฟเวอร์นั้นไม่จำเป็นจะต้องมีข้อมูลของเว็บเพจหรือรีซอร์สเสมอไปอย่างเช่นในกรณีที่การร้องขอแบบ HTTP ของผู้ใช้ส่งมาเป็น If-Modified-Since Header ถ้าเว็บเพจหรือรีซอร์สที่ทำการร้องขอไม่ได้มีการเปลี่ยนแปลงในเวลาของ If-Modified-Since Header แล้วเว็บเซิร์ฟเวอร์ก็จะตอบกลับด้วย Not-Modified Header ในข้อมูลการตอบกลับแบบ HTTP เท่านั้น

2.3.3.2.2 การควบคุมการเข้าถึงเว็บเซิร์ฟเวอร์

เว็บเซิร์ฟเวอร์อาจจะต้องมีการจำกัดการเข้าถึงของผู้ใช้ที่เข้ามาใช้งานแต่ละคน กระบวนการควบคุมการเข้าถึง(Access Control) จะเป็นการรวมกระบวนการ "การยืนยันตัวเองของผู้ใช้ (Authentication)" ที่ใช้ในการบ่งบอกว่าใครเป็นผู้ร้องขอเว็บเพจหรือรีซอร์สนั้นและ "การบอกสิทธิของผู้ใช้ (Authorization)" ซึ่งใช้ในการตรวจสอบว่าผู้ใช้มีสิทธิในการเข้าถึงเว็บเพจหรือรีซอร์สนั้นหรือไม่

<u>การยืนยันตัวเอง (AUTHENTICATION)</u>

การทำงานแบบไคลเอนต์-เซิร์ฟเวอร์นั้นการยืนยันตัวเองจะเป็นการถามชื่อผู้เข้าใช้ (Username) และรหัสผ่าน (Password) ของผู้ใช้งาน ในการทำงานแบบนี้เว็บเซิร์ฟเวอร์จะทำการเก็บรักษารหัสผ่านของผู้ใช้งานไว้โดยจะประกอบด้วยชื่อผู้เข้าใช้ (Username) และรหัสผ่าน (Password) ของผู้ใช้ที่ต้องมีการยืนยันตัวเอง การเก็บรหัสผ่านอาจจะเก็บในรูปแบบที่มีการเข้ารหัสหรือไม่ก็ได้ ในการสร้างเซสชั่น (Session) เพื่อร้องขอเว็บเพจหรือรีซอร์สใดของผู้ใช้งานจะต้องมีการให้ทางผู้ใช้ส่งชื่อผู้เข้าใช้และรหัสผ่านมาด้วยแล้วทางเว็บเซิร์ฟเวอร์จะตรวจสอบว่าชื่อผู้เข้าใช้และรหัสผ่านถูกต้องหรือไม่ ถ้าถูกต้องเซสชั่นระหว่างเว็บบราวเซอร์ผู้ใช้และเว็บเซิร์ฟเวอร์ก็จะถูกสร้างขึ้น ถ้ามีการยกเลิกเซสชั่นแล้ว ผู้ใช้งานจะต้องมีการส่งชื่อผู้เข้าใช้และรหัสผ่านอีกครั้งในการสร้างเซสชั่นใหม่

เซสชั่น(Session) ก็คือการเชื่อมต่อในการส่งเว็บเพจหรือรีซอร์สหนึ่งๆ ดังนั้นเว็บเซิร์ฟเวอร์จะต้องมีการยืนยันตัวเองในทุกๆ เซสชั่นซึ่งทำให้ผู้ใช้งานจะต้องทำการส่งชื่อผู้เข้าใช้และรหัสผ่านให้ทุกครั้งดังนั้นเว็บบราวเซอร์ของผู้ใช้จะมีความสามารถการเก็บชื่อผู้เข้าใช้และรหัสผ่านที่ได้มีการส่งไปให้กับเว็บเซิร์ฟเวอร์ในครั้งแรกไว้ ดังนั้นการร้องขอต่างๆที่ถูกส่งภายหลัง ผู้ใช้งานก็ไม่จำเป็นต้องมีการใส่ชื่อผู้เข้าใช้และรหัสผ่านอีกครั้งเพราะเว็บบราวเซอร์จะทำการใส่ชื่อผู้เข้าใช้และรหัสผ่านไว้ใน Request Header ให้เอง และเว็บบราวเซอร์จะเก็บชื่อผู้เข้าใช้และรหัสผ่านไว้จนกว่าจะมีการปิดเว็บบราวเซอร์นั้น

สำหรับรูปแบบในการยืนยันตัวเองนั้นก็มีหลายแบบเช่น Basic Authentication, Digest Access Authentication เป็นต้น

<u>การตรวจสอบสิทธิ (AUTHORIZATION)</u>

การตรวจสอบสิทธิเป็นการที่จะทำให้เว็บเซิร์ฟเวอร์ใช้ในการตรวจสอบว่าการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆ เพราะเว็บเพจหรือรีซอร์สต่างๆในเว็บเซิร์ฟเวอร์จะมีการควบคุมการเข้าถึงอยู่ เช่น Admin.asp อาจจะสามารถเข้าถึงได้เฉพาะผู้ใช้งานที่มีสิทธิเป็น Superuser เท่านั้น ผู้ใช้ทั่วไปไม่สามารถเข้าถึงได้ โดยสิทธิในการเข้าถึงรีซอร์สต่างๆในเว็บเซิร์ฟเวอร์จะถูกเก็บไว้เรียกว่า

Access Control List ที่จะแสดงว่าผู้ใช้แต่ละคนมีสิทธิในการเข้าถึงรีซอร์สใดบ้าง ซึ่งรายละเอียดเกี่ยวกับการสร้างและการเก็บรักษา Access Control List นั้นจะขึ้นอยู่กับแต่ละโปรแกรมเว็บเซิร์ฟเวอร์

การตรวจสอบสิทธินั้นส่วนมากจะทำเป็น Directory มากกว่าที่จะเป็นรีซอร์สๆหนึ่ง สำหรับการตรวจสอบสิทธินั้นจะทำให้ประสิทธิภาพของเว็บเซิร์ฟเวอร์ลดลงมากเพราะเว็บเซิร์ฟเวอร์จะต้องมีการตรวจสอบผู้เข้าใช้และรหัสผ่านทุกครั้งที่มีการเข้าถึงดังนั้นควรกำหนดให้มีการตรวจสอบสิทธิ เฉพาะรีซอร์สหรือ Directory ที่จำเป็นเท่านั้น

2.3.3.2.3 การสร้างการตอบกลับแบบไม่คงที่ (Dynamically generated responses)

การทำงานโดยทั่วไปของเว็บเซิร์ฟเวอร์จะมีการส่งข้อมูลเว็บเพจที่เป็นแบบคงที่ (Static) แต่ต่อมาก็ได้มีการทำงานของเว็บเพจที่เป็นแบบไม่คงที่ (Dynamic) เกิดขึ้น โดยเว็บเพจแบบไม่คงที่จะมีการทำงานอื่นร่วมอยู่ด้วย ข้อมูลของเว็บเพจแบบไม่คงที่นั้นจะถูกสร้างขึ้นเมื่อมีการร้องขอจากผู้ใช้งาน ดังนั้นผู้ใช้งานที่เข้าถึงเว็บเพจแบบไม่คงที่แต่ละคนอาจจะได้รับข้อมูลของเว็บเพจที่ไม่เหมือนกัน โดยเว็บเพจแบบไม่คงที่สามารถถูกสร้างได้จากหลายๆรูปแบบ เช่นการใช้ Server-Side Includes/Scripts สำหรับภาษาโปรแกรมที่ใช้ในการสร้าง Server-side ก็เช่น Active Server Page(ASP), Professional Homepage(PHP), Perl, Java Server Page(JSP)

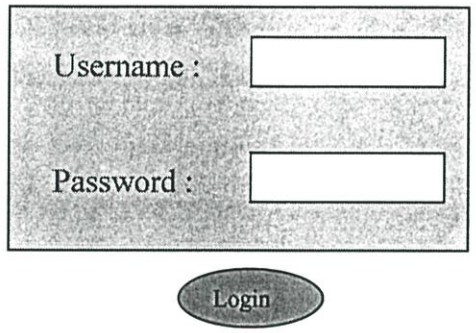
สำหรับ Server-Side Include/Script นั้นเมื่อมีการร้องขอเว็บเพจที่เป็นแบบ Server-Side Include/Script แล้ว เว็บเซิร์ฟเวอร์จะทำการประมวลผล Server-Side Include/Script เว็บเพจนั้นเมื่อได้เป็นผลลัพธ์ออกมาแล้วก็จะทำการส่งเว็บเพจผลลัพธ์นั้นกลับไปให้ผู้ร้องขอ ในการประมวลผลนั้นอาจจะต้องมีการใช้ตัวแปร (Parameter) บางอย่างที่ทางผู้ร้องขอส่งมาด้วย

การส่งตัวแปร (Parameter) ในการร้องขอ Server-Side Include/Script นั้นสามารถส่งได้ 2 แบบ คือการส่งผ่าน URL ของเว็บเพจที่เป็น Server-Side กับส่งโดยการผ่านแท็ก <FORM> ของเว็บเพจ โดยการส่งผ่าน URL นั้นจะส่งให้รูปที่มี ? เป็นตัวบอกว่ามีการส่งตัวแปร (Parameter) ให้กับเว็บเพจนั้นๆ และแต่ละตัวแปร (Parameter) จะคั่นด้วย & เช่น มีการร้องขอ

"www.it.kmitl.ac.th/login.asp?cost=10&name=wisit"

จากการร้องขอข้างบนจะเป็นการส่งตัวแปร (Parameter) ที่เรียกว่าเป็นการส่งแบบ GET Method เพราะตัวแปร (Parameter) จะส่งมาใน URL เลยจากตัวอย่างเป็นการส่งตัวแปร (Parameter) ที่ชื่อ Cost ซึ่งมีค่า 10 และส่งตัวแปร (Parameter) ที่ชื่อ Name มีค่า wisit ไปให้กับ เว็บเพจ Login.asp ซึ่งเว็บเซิร์ฟเวอร์ก็จะทำการประมวลผล login.asp โดยใช้ค่าตัวแปร (Parameter) ตามที่ส่งมา

สำหรับการส่ง Parameter อีกแบบจะเรียกว่าส่งแบบ POST Method โดยจะส่งผ่านแท็ก <FORM> ในเว็บเพจที่เป็น Server-Side Include/Script เช่นในเว็บเพจมีฟอร์มดังรูป ซึ่งเมื่อ Submit แล้วส่งจะมีการส่งค่า Parameter ไปยัง Login.asp



รูปที่ 2.16 แสดงตัวอย่างของการส่งตัวแปร (Parameter) ผ่านแท็ก <FORM> ในเว็บเพจ

การส่งตัวแปร (Parameter) แบบนี้ข้อมูลตัวแปร (Parameter) จะอยู่ในส่วนของ Entity Body ของข้อมูลการร้องขอแบบ HTTP ที่ส่งไปให้ทางเว็บเซิร์ฟเวอร์ เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอ Login.asp แล้วก็จะทำการดึงค่าตัวแปร (Parameter) ที่อยู่ใน Entity Body ของข้อมูลการร้องขอแบบ HTTP ออกมาเพื่อใช้ในการประมวลผล

การทำงานของเว็บเซิร์ฟเวอร์ที่มีการสร้างเว็บเพจแบบไม่คงที่นั้นจะทำให้เว็บเซิร์ฟเวอร์มีการทำงานมากขึ้น ดังนั้นจึงทำให้ประสิทธิภาพการทำงานของเว็บเซิร์ฟเวอร์ลดลงกว่าเว็บเซิร์ฟเวอร์ที่ให้บริการเฉพาะเว็บเพจแบบคงที่ อีกทั้งยังมีความเสี่ยงเกี่ยวกับด้านความปลอดภัยของ เว็บเซิร์ฟเวอร์ด้วยเพราะอาจมีการสร้าง Server-Side Include/Script ที่มีการเข้าถึงไฟล์หรือรีซอร์สต่างๆ บนเว็บเซิร์ฟเวอร์โดยตรงได้ ทำให้ Server-Side Include/Script สามารถลบหรือแก้ไขไฟล์หรือรีซอร์สต่างๆได้ ดังนั้นจะต้องมีการตั้งค่าการทำงานเกี่ยวกับการเข้าถึงรีซอร์สต่างๆอย่างระมัดระวัง

2.3.3.2.4 การสร้างและใช้งานคุกกี้

คุกกี้ (Cookies) นั้นเป็นหนทางหนึ่งที่ทำให้เว็บเซิร์ฟเวอร์สามารถที่จะเก็บข้อมูลบางอย่างเกี่ยวกับผู้ใช้งานได้ ซึ่งเมื่อเว็บเซิร์ฟเวอร์ต้องการที่จะให้เว็บบราวเซอร์ของผู้ใช้งานสร้าง Cookie นั้นเว็บเซิร์ฟเวอร์จะมีการส่งการตอบกลับแบบ HTTP ที่มี Header เป็น “Set-Cookie” พร้อมกับชื่อและค่าของคุกกี้นั้นให้กับเว็บบราวเซอร์จากนั้นเว็บบราวเซอร์จะทำการสร้างคุกกี้นั้นและเก็บไว้ที่เครื่องของผู้ใช้ และการร้องขอครั้งถัดไปของผู้ใช้ก็จะมีคุกกี้นี้ส่งมาด้วย

2.3.3.3 การใช้ข้อมูลในการสร้างการร้องขอแบบ HTTP ร่วมกัน

แม้ว่าเว็บเซิร์ฟเวอร์จะสามารถจัดการการร้องขอแบบ HTTP ได้โดยอิสระ โดยจะมีการสร้างโพรเซส (Process) ขึ้นมาจัดการกับการร้องขอแบบ HTTP แต่ละอัน ซึ่งการทำงานอย่างนี้ทำให้เว็บเซิร์ฟเวอร์มีการทำงานที่หนัก ดังนั้นก็มีวิธีที่จะช่วยลดการทำงานนี้

2.3.3.3.1 การใช้การร้องขอแบบ HTTP และการตอบกลับ HTTP ร่วมกัน

ในการที่เว็บเซิร์ฟเวอร์จะสร้างข้อมูลการตอบกลับแบบ HTTP แล้วส่งกลับไปให้ทางผู้ร้องขอนั้น เว็บเซิร์ฟเวอร์จะต้องมีการอ่านข้อมูลจากแฟ้มข้อมูลบนดิสก์ จากนั้นก็ดึงมาเก็บไว้ที่หน่วยความจำ จากนั้นถึงทำการสร้างเป็นการร้องขอแบบ HTTP ได้ ซึ่งการทำงานแบบนี้ก็ทำให้มีความล่าช้าในการทำงานมาก เพราะถ้ามีการร้องขอเป็นจำนวนมากเว็บเซิร์ฟเวอร์ก็ต้องมีการทำงานดังที่กล่าวมาแล้วซ้ำแล้วซ้ำอีกในแต่ละการร้องขอ ทั้งๆที่เป็นเว็บเพจเดียวกัน ดังนั้นจึงมีการนำเสนอการทำงานที่เรียกว่า "แคชชิ่ง(caching)" บนเว็บเซิร์ฟเวอร์ขึ้น

การแคชชิ่งบนเว็บเซิร์ฟเวอร์จะไม่เหมือนกับการแคชชิ่งบนเว็บบราวเซอร์เพราะการแคชชิ่งบนเว็บเซิร์ฟเวอร์นั้นสำหรับเว็บเพจแบบคงที่จะทำการเก็บข้อมูลของเว็บเพจนั้นที่ได้มีการส่งไปให้ยังผู้ใช้คนก่อนๆ ไว้ในหน่วยความจำ จากนั้นเมื่อมีผู้ใช้คนอื่นร้องขอเว็บเพจเดียวกันนี้เว็บเซิร์ฟเวอร์ก็สามารถนำข้อมูลของเว็บเพจนั้นจากหน่วยความจำมาสร้างเป็นข้อมูลการตอบกลับแบบ HTTP แล้วส่งกลับไปให้ผู้ใช้ได้เลย แต่การแคชชิ่งของเว็บเพจแบบคงที่นั้นเว็บเซิร์ฟเวอร์จะต้องมีการตรวจสอบด้วยว่าเว็บเพจที่ได้มีการแคชไว้นั้นได้รับการแก้ไขหรือไม่ ถ้าได้รับการแก้เว็บเซิร์ฟเวอร์ต้องมีการดึงเอาข้อมูลของเว็บเพจล่าสุดมาแคชชิ่งไว้

สำหรับการแคชเว็บเพจแบบไม่คงที่นั้นก็สามารถทำได้ โดยเว็บเซิร์ฟเวอร์จะทำการเก็บผลลัพธ์ของการประมวลผลของเว็บเพจแบบไม่คงที่ที่มีการร้องขอบ่อยไว้ เมื่อมีการร้องขอเว็บเพจแบบไม่คงที่ที่มีตัวแปร (Parameter) แบบเดิมก็สามารถที่จะดึงเอาผลลัพธ์ที่ได้ประมวลผลไว้แล้วส่งกลับไปให้ผู้ใช้ได้เลย แต่การทำงานแคชชิ่งแบบนี้ก็ต้องมีการตรวจสอบก่อนด้วยว่าผลลัพธ์ของการประมวลผลเว็บเพจแบบไม่คงที่นั้นได้มีการเปลี่ยนแปลงหรือไม่ เช่นถ้าเว็บเพจแบบไม่คงที่ต้องมีการติดต่อกับฐานข้อมูล เมื่อข้อมูลในฐานข้อมูลมีการเปลี่ยนแปลงผลลัพธ์ที่มีการแคชชิ่งไว้ก็ไม่สามารถใช้ได้

2.3.3.3.2 การใช้งาน METADATA ของการร้องขอแบบ HTTPร่วมกัน

อย่างที่ได้กล่าวมาแล้วว่าเว็บเซิร์ฟเวอร์จะต้องมีการสร้างโพรเซสเพื่อใช้ในการจัดการกับการร้องของผู้ใช้ที่ได้รับ ดังนั้นเว็บเซิร์ฟเวอร์สามารถที่จะแคชชิ่งข้อมูลแบบอย่างของโพรเซสที่ใช้จัดการการร้องขอเว็บเพจหนึ่งไว้ เช่น พอยเตอร์ (Pointer) ที่ใช้ชี้ยังข้อมูลของเว็บเพจนั้นบนดิสก์, ข้อมูลการแปลงจาก URL ไปเป็นชื่อไฟล์, Header ของการตอบกลับแบบ HTTP ของเว็บเพจนี้ไว้ เมื่อมีการร้องขอเว็บเพจเดียวกันนี้จากผู้ใช้อื่นๆ เว็บเซิร์ฟเวอร์ก็สามารถสร้างโพรเซสที่ใช้ในการ

จัดการการร้องขอโดยนำข้อมูลของโพรเซสก่อนหน้านี้มาใช้ได้เลย ซึ่งการจะสามารถช่วยลดเวลาในการจัดการการร้องขอผู้ใช้ได้

2.3.3.4 รูปแบบต่างๆ ของเว็บเซิร์ฟเวอร์

ในหัวข้อนี้จะกล่าวถึงรูปแบบโมเดลต่างๆที่เว็บเซิร์ฟเวอร์ใช้ในการจัดการกับรีซอร์สของเว็บเซิร์ฟเวอร์เพื่อรองรับการร้องขอของผู้ใช้

2.3.3.4.1 เว็บเซิร์ฟเวอร์แบบ EVENT-DRIVEN

เป็นวิธีการที่ง่ายที่สุดคือจะมี 1 โพรเซสในการจัดการเพียง 1 การร้องขอทั้งหมด ณ เวลาหนึ่งๆ โดยสิ่งที่จะต้องทำก็คือ โพรเซสจะต้องทำยอมรับการร้องขอ, สร้างข้อมูลการตอบกลับแบบ HTTP และส่งกลับไปยังผู้ร้องขอ จากนั้นก็จะถึงไปจัดการกับการร้องขอถัดไป EVENT-DRIVEN จะใช้หลักการนี้ในการทำงานแต่ EVENT-DRIVEN จะทำการแบ่งการทำงานออกเป็นงานย่อยๆ คือ การยอมรับการร้องขอ, อ่านและแปลการร้องขอแบบ HTTP, ดึงเว็บเพจหรือรีซอร์สที่ร้องขอ, ส่งเป็นการตอบกลับแบบ HTTP กลับไปให้ผู้ร้องขอ ซึ่งทำการที่โพรเซสทำงานส่วนย่อยๆนี้จะมีช่วงเวลาที่จะต้องรอหรือมีการทำงานเกี่ยวกับ I/O Device เช่น Diskdrive ที่จะใช้เวลาในการจัดการหลังจากที่มีการร้องขอ ดังนั้นช่วงเวลาที่โพรเซส กำลังรองานเกี่ยวกับ I/O อยู่โพรเซสก็สามารถที่จะปรับเปลี่ยนไปจัดการกับการร้องขออันถัดไปได้ นอกเหนือจากนั้นการทำงานแบบนี้ยังเป็นการควบคุมความเป็นลำดับ (Serializing) ของการร้องขอของแต่ละผู้ร้องขอได้ เช่น 2 ผู้ใช้งานอาจจะต้องการเขียนหรือแก้ไขเว็บเพจหรือรีซอร์สเดียวกัน EVENT-DRIVEN จะสามารถทำให้แน่ใจได้ว่าการแก้ไขนั้นจะเป็นไปตามลำดับที่ถูกต้อง อีกทั้งถ้าเป็นการร้องขอรีซอร์สเดียวกันยังสามารถใช้ประโยชน์จากการแคชชิ่งได้อีกด้วย

แต่การทำงานแบบ EVENT-DRIVEN นั้นจะต้องมีการจัดการเกี่ยวกับ Non-Blocking และ Blocking System เช่น พวก I/O ต่างตามที่ได้กล่าวมาแล้ว ดังนั้นโปรแกรมเว็บเซิร์ฟเวอร์จะต้องการติดต่อกับระบบปฏิบัติการณ์ที่ยุ่งยากมากขึ้นๆ ดังนั้นเว็บเซิร์ฟเวอร์ส่วนใหญ่จะไม่ใช้การทำงานแบบ EVENT-DRIVEN

2.3.3.4.2 เว็บเซิร์ฟเวอร์แบบ PROCESS-DRIVEN

ในการแบบนี้จะใช้โพรเซส 1 โพรเซสในการจัดการ 1 การร้องขอซึ่งจะแตกต่างจาก EVENT-DRIVEN ที่จะใช้เพียง 1 โพรเซสในการจัดการการร้องขอทั้งหมด ดังนั้นถ้าสมมติโพรเซสหนึ่งที่มีการจัดการการร้องขออยู่เกิดต้องมีการรอเกี่ยวกับ I/O Device โปรแกรมเว็บเซิร์ฟเวอร์ก็สามารถที่เปลี่ยนไปประมวลผลโพรเซสที่จัดการการร้องขออื่น

เว็บเซิร์ฟเวอร์แบบ PROCESS-DRIVEN นั้นจะมีโพรเซสหนึ่งที่เป็นโพรเซสหลักเมื่อมีการร้องขอจากผู้ใช้ส่งมายังเว็บเซิร์ฟเวอร์แล้วโพรเซสหลัก จะทำการสร้างโพรเซสย่อยขึ้นมาจัดการการ

ร้องขอนั้น แล้วเมื่อมีการจัดการการร้องขอเรียบร้อยแล้วโพรเซสย่อยที่ถูกสร้างขึ้นมานั้นก็จะถูกยกเลิก แต่โพรเซสหลักอาจจะต้องการลดการทำงานที่ลดประสิทธิภาพของเว็บเซิร์ฟเวอร์ในการที่จะต้องสร้างโพรเซสขึ้นใหม่ทุกครั้งที่มีการร้องขอเข้ามา โดยโพรเซสหลักอาจจะทำการสร้างโพรเซสย่อยๆไว้ก่อนเลยหลายๆ โพรเซสไว้เลยแต่กำหนดสถานะของโพรเซสย่อยนั้นไว้ว่าอยู่ในภาวะรอการทำงาน จากนั้นเมื่อมีการร้องขอส่งมาโพรเซสหลักก็กำหนดสถานะของโพรเซสย่อยหนึ่งๆที่ได้มีการสร้างไว้แล้วเป็นทำงาน แล้วให้โพรเซสนั้นจัดการกับการร้องขอที่เข้ามา

แต่การทำงานแบบ PROCESS-DRIVEN นั้นก็มีข้อเสียคือ อย่างแรกคือ ประสิทธิภาพที่ลดลงเมื่อมีการต้องสับเปลี่ยนไปทำงานโพรเซสต่างๆ เพราะอาจจะต้องมีการจัดเก็บข้อมูลของโพรเซสปัจจุบันเช่นเก็บค่า Address ของ Memory ต่างๆ, ค่า Counter ต่างไว้ จากนั้นก็ต้องทำการดึงข้อมูลของโพรเซสใหม่ขึ้นมาใช้ประมวลผล ซึ่งใน EVENT-DRIVEN นั้นใช้โพรเซสเดียวดังนั้นค่า Address ต่างๆจึงไม่เปลี่ยนแปลง อีกทั้งยังเกิดปัญหาในการที่โพรเซส 2โพรเซสต้องมีการใช้รีซอร์สร่วมกัน ดังนั้นต้องเสียการทำงานในการดึงข้อมูลเข้าแล้วออกไปมาในการใช้งาน

2.3.3.4.3 เว็บเซิร์ฟเวอร์แบบ HYBRID

เว็บเซิร์ฟเวอร์แบบ HYBRID นั้นจะเป็นการรวมเอาข้อดีของทั้ง EVENT-DRIVEN และ PROCESS-DRIVEN เข้าด้วยกัน ซึ่งในการรวมการทำงานทั้งสองอย่างนี้โดย Hybrid server ก็มีการทำงานหลายๆแบบ

- แบบแรกจะมีการแบ่งส่วนการทำงานที่อิสระย่อยที่เรียกว่า “Thread” แต่ละ Thread จะมีการทำงานที่อิสระในโพรเซสดังนั้นเว็บเซิร์ฟเวอร์สามารถที่จะกำหนดให้แต่ละ Thread ในโพรเซสสามารถที่จะจัดการกับการร้องขอของผู้ใช้ที่แตกต่างกันได้ ซึ่งส่วนนี้เป็นการทำงานที่คล้ายกับ EVENT-DRIVEN แต่เพราะแต่ละ Thread มีการทำงานที่อิสระ ดังนั้นเมื่อ Thread ใด Thread หนึ่งในโพรเซสต้องมีการรอเกี่ยวกับ I/O แล้ว Thread อื่นในโพรเซสเดียวกันก็สามารถที่ทำงานต่อไปได้ แล้วประสิทธิภาพในการเปลี่ยนไปทำงานที่แต่ละ Thread ยังดีกว่าในการปรับเปลี่ยนการทำงานระหว่างโพร เซสเพราะ Thread อยู่ภายใต้โพรเซสเดียวกันทำให้ใช้ Address ของ Memory ร่วมกัน อีกทั้งการใช้การทำงานแบบ Thread ยังช่วยในการแก้ปัญหาในการจัดการเกี่ยวกับรีซอร์สที่มีการใช้งานร่วมกันด้วย
- แบบที่สอง เว็บเซิร์ฟเวอร์นั้นอาจจะจัดการเกี่ยวกับการร้องขอที่ไม่มีการติดต่อกับ I/O เช่นมีความผิดพลาดเช่นการร้องขอเว็บเพจหรือรีซอร์สที่ไม่มีอยู่บนเว็บเซิร์ฟเวอร์ ในกรณีนี้ ถ้าเว็บเซิร์ฟเวอร์มีการทำงานแบบ PROCESS-DRIVEN ก็จะทำให้เกิดการเสียเวลาในการสร้างโพรเซสเพื่อเพียงแค่ตอบกลับเป็นการตอบกลับแบบ HTTP ที่แสดงความผิดพลาดเท่านั้น ดังนั้นในกรณีนี้เว็บเซิร์ฟเวอร์ควรที่จะมีการทำงานแบบ EVENT-DRIVEN เพราะไม่ต้องเสียเวลาในการสร้างโพรเซสใหม่และยังสามารถใช้ประโยชน์จาก Header ของข้อมูลการ

ตอบกลับแบบ HTTP ที่ได้มีการแคชชิ่งไว้ด้วย ดังนั้นเว็บเซิร์ฟเวอร์แบบ HYBRID นี้จะมีโพรเซสหลักที่มีการทำงานแบบ EVENT-DRIVEN แล้วเมื่อมีการร้องขอเข้ามีที่ต้องมีการติดต่อกับ I/O แล้วโพรเซสหลักก็จะทำการสร้างโพรเซสลูกขึ้นมาจัดการ แต่ถ้าการร้องขอนั้นไม่ต้องมีการทำงานเกี่ยวกับ I/O โพรเซสก็สามารถจัดการการร้องขอนั้นได้เลย

2.3.3.5 การตั้งเว็บไซต์บนเว็บเซิร์ฟเวอร์ (SERVER HOSING)

2.3.3.5.1 การตั้งหลายๆเว็บไซต์บนเว็บเซิร์ฟเวอร์เครื่องเดียว (Multiple Website on single web server)

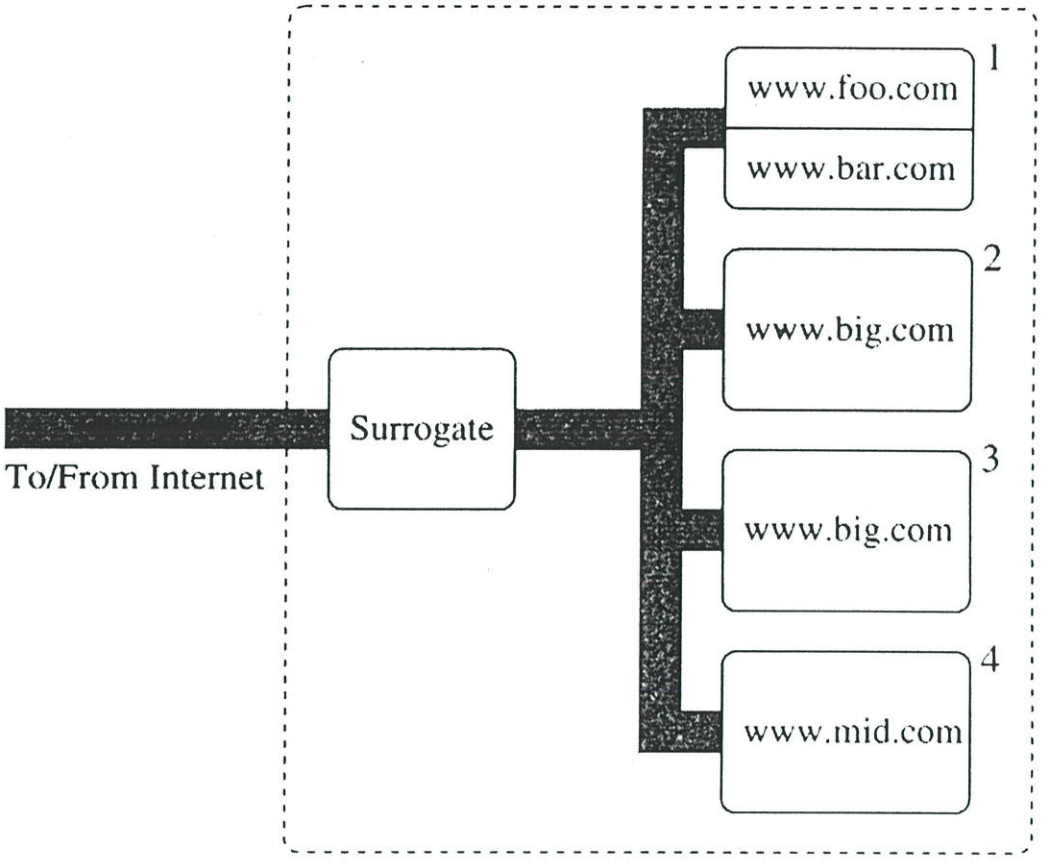
รูปแบบการของเว็บเซิร์ฟเวอร์นี้จะมีการเอาเว็บไซต์หลายๆ เว็บไซต์ลงไปในเครื่องเว็บเซิร์ฟเวอร์เครื่องเดียวเท่านั้น ดังนั้นเว็บเซิร์ฟเวอร์เครื่องเดียวนี้จะมีหน้าที่ใช้การจัดการการร้องขอของเว็บไซต์ทั้งหมดที่มีอยู่ ซึ่งเว็บเซิร์ฟเวอร์นี้จะให้บริการโดย Hosting Company ที่จะให้บริการเช่าเครื่องเว็บเซิร์ฟเวอร์ในการสร้างเว็บไซต์ขององค์กรต่างๆ ซึ่งการทำงานแบบ Multiple Website นั้นจะมีสองแบบ

- โฮสต์เสมือน(Virtual host) : เว็บเซิร์ฟเวอร์แบบนี้จะใช้ไอพีแอดเดรส (IP Address) เดียวกันในแต่ละ เว็บไซต์แต่จะใช้โดเมนเนมในการแยกความแตกต่างระหว่างเว็บไซต์ซึ่งการทำงานแบบนี้โปรแกรมเว็บเซิร์ฟเวอร์จะต้องสนับสนุนการทำงานของโฮสต์เสมือนด้วยและจะใช้บน HTTP/1.1 เท่านั้น
- การใช้ไอพีเดียว(Dedicate IP) : เว็บเซิร์ฟเวอร์แบบนี้แต่ละเว็บไซต์จะมีไอพีแอดเดรส (IP address) เป็นของตนเอง

2.3.3.5.2 การตั้งเว็บไซต์เดียวบนเว็บเซิร์ฟเวอร์หลายๆเครื่อง (Multiple machine for single web site)

สำหรับเว็บไซต์ที่มีจำนวนผู้เข้ามาใช้งานจำนวนมาก การใช้เครื่องเพียงหนึ่งเครื่องเป็นเว็บเซิร์ฟเวอร์ในการจัดการกับการร้องขอจำนวนมากนั้น อาจจะทำให้ใช้เวลาในการจัดการการร้องขอนาน ดังนั้นจึงมีการสร้างหรือจัดกลุ่มของเครื่องเว็บเซิร์ฟเวอร์ให้จัดการเว็บไซต์เดียวกัน โดยจะต้องมีกลไกในการจัดการเกี่ยวกับการแปลงจากโดเมนเนมของเว็บไซต์เป็นไอพีแอดเดรส ถ้าสมมติ www.foo.com มีการใช้ 2 เครื่องเว็บเซิร์ฟเวอร์ที่มีไอพีแอดเดรสเป็น 10.1.1.1 กับ 10.1.1.2 ถ้ามีการร้องขอการแปลงโดนเมนเนม “www.foo.com” มาที่โดเมนเนมเซิร์ฟเวอร์แล้วโดเมนเนมเซิร์ฟเวอร์อาจจะส่งไอพีแอดเดรส กลับไปเป็น 10.1.1.1 หรือ 10.1.1.2 ก็ได้ซึ่งอาจจะเป็นแบบสุ่มหรือตั้งค่าไว้ก็ได้

ส่วนสำคัญอีกอย่างหนึ่งคือจะต้องมีการสำเนาข้อมูลในเครื่องทุกเครื่องของเว็บไซต์เดียวกันให้มีข้อมูลที่เหมือนกันๆด้วย ซึ่งตรงนี้อาจจะใช้ Diskdrive ร่วมกันสำหรับเครื่องทุกเครื่องหรือใช้การทำงานของ RAID เข้ามาช่วยได้



รูปที่ 2.17 รูปแบบการ Hosting แบบต่างๆ

2.3.3.6 สรุป

เว็บเซิร์ฟเวอร์มีหน้าในการสร้างและส่งการตอบกลับแบบ HTTP กลับไปให้ไคลเอนต์ เมื่อมีการส่งการร้องขอมาเว็บเซิร์ฟเวอร์จะใช้ URL เพื่อทำการแปลงเป็นชื่อของไฟล์ที่อยู่ในดิสก์ของเครื่องเว็บเซิร์ฟเวอร์เอง และทำการดึงไฟล์นั้นออกมาแล้วส่งกับไปเป็นการตอบกลับแบบ HTTP เว็บเซิร์ฟเวอร์สามารถมีกลไปในการร้องการยืนยันตัวเองจากผู้ใช้ได้ นอกจากนั้นเว็บเซิร์ฟเวอร์ยังมีกลไกในการตรวจสอบสิทธิในการเข้าถึงเว็บเพจหรือรีซอร์สใดๆของผู้ใช้ด้วย เว็บเพจที่เว็บเซิร์ฟเวอร์ส่งกลับไปให้ผู้ใช้นั้นมีทั้งเป็นแบบคงและเป็นแบบไม่คงที่ รูปแบบของเว็บเซิร์ฟเวอร์นั้น

มีทั้งเป็นแบบ EVENT-DRIVEN หรือ PROCESS-DRIVEN หรือเป็น HYBRID ที่เป็นการทำงานร่วมกันระหว่าง EVENT-DRIVEN และ PROCESS-DRIVEN ได้

บทที่ 3
ระบบการทำงานของพรีเฟรชชิ่ง

3.1 กล่าวนำ

ความแพร่หลายของเวิลด์ไวด์เว็บทำให้มีผู้ใช้ในการเข้างานอินเตอร์เน็ตเพิ่มขึ้นมาก ซึ่งทำให้การจราจรบนเครือข่ายอินเตอร์เน็ตมีความคับคั่งของข้อมูลสูงมากขึ้น และเมื่อการจราจรบนเครือข่ายอินเตอร์เน็ตคับคั่งมาก จะทำให้การเข้าถึงเว็บเพจหรือรีซอร์สใดบนอินเตอร์เน็ตมีแนวโน้มว่าจะใช้เวลาเพิ่มหรือว่านานขึ้น เพราะว่าการเข้าถึงเว็บเพจหรือรีซอร์สใดบนอินเตอร์เน็ตจะต้องมีการสร้างการเชื่อมต่อ (TCP Connection) ไปยังเซิร์ฟเวอร์จากนั้นทางแอพลิเคชั่นทางด้านไคลเอนต์ (Web Client) หรือเว็บบราวเซอร์ ก็จะทำการส่งข้อมูลการร้องขอแบบ HTTP (HTTP Request) ของเว็บเพจหรือรีซอร์สใดที่ต้องการเข้าถึงไปยังเว็บเซิร์ฟเวอร์ (Web Server) เมื่อเว็บเซิร์ฟเวอร์ได้รับข้อมูลการร้องขอแบบ HTTP แล้วก็จะทำการดึงเอาเว็บเพจหรือรีซอร์สนั้นออกมาแล้วส่งกลับไปด้วยข้อมูลการตอบกลับแบบ HTTP (HTTP Response) และส่งกลับไปยังไคลเอนต์ซึ่งเวลาส่วนใหญ่ของกระบวนการร้องขอและการตอบกลับระหว่างเซิร์ฟเวอร์และไคลเอนต์จะเป็นช่วงเวลาที่ข้อมูลของการส่งการร้องขอหรือการตอบกลับเดินทางอยู่ในเครือข่ายซึ่งถ้าเครือข่ายมีความคับคั่งแล้วจะทำให้เวลาโดยรวมของการเข้าถึงเว็บเพจหรือรีซอร์สใดเพิ่มสูงขึ้น

มีวิธีการต่างๆที่ถูกนำเสนอในการช่วยลดการเข้าถึงรีซอร์สต่างๆ บนอินเตอร์เน็ตเช่น Cache Replacement[1,2] , Speculative Prefecthing[3-4] ซึ่งวิธีการต่างๆเหล่านี้ก็มีประสิทธิภาพต่างกันไป แต่วิธีที่มีการใช้กันอย่างแพร่หลายในปัจจุบันคือการใช้พร็อกซี่เซิร์ฟเวอร์(Proxy Server) นั่นก็คือการใช้พร็อกซี่หรือในการเก็บเว็บเพจหรือรีซอร์สต่างๆที่ได้มีการเข้าถึงโดยผู้ใช้คนหนึ่งแล้วไว้ เมื่อมีผู้ใช้คนอื่นๆต้องการเข้าถึงเว็บเพจหรือรีซอร์สเดียวกันนี้ พร็อกซี่ก็สามารถที่จะส่งเว็บเพจหรือรีซอร์สนั้นกลับไปให้ผู้ใช้ได้เลย ไม่จำเป็นต้องมีการร้องขอจากเว็บเซิร์ฟเวอร์ปลายทางตามรายละเอียดของพร็อกซี่ที่ได้กล่าวถึงในบทที่ 2 แล้ว นอกจากนี้ยังสามารถนำพร็อกซี่มาใช้ในกระบวนการที่เรียกว่า “พรีเฟรชชิ่ง(PRE-FETCHING)” ซึ่งเป็นกระบวนที่สามารถช่วยในการลดเวลาในการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆ เช่นเดียวกัน

3.2 ระบบพรีเฟรชชิ่ง

สำหรับคำนิยามของการพรีเฟรชชิ่งก็คือ เป็นระบบที่จะทำการดึงการทำงานบางอย่างที่จะต้องเกิดขึ้นในข้อมูลการร้องขอแบบ HTTP ของผู้ใช้ในอนาคต แล้วทำงานส่วนนั้นล่วงหน้าก่อนที่

จะต้องมีการทำงานส่วนนั้นจริง ซึ่งการทำงานอย่างนี้จะทำให้ลดเวลาในการทำงานส่วนนั้นเมื่อต้องใช้งานจริงเพราะได้มีการทำงานส่วนนั้นไว้แล้ว การทำงานที่จะทำล่วงหน้านั้นก็จะจำแนกได้เป็นประเภท ซึ่งจะกล่าวในภายหลัง

ระบบพรีเฟรชชิ่งนั้นจะต้องอาศัยข้อมูลการเข้าถึงของผู้ใช้ในอดีตเพื่อใช้ในการทำนายการเข้าถึงที่จะเกิดขึ้นในอนาคตเพื่อใช้ในการทำงานล่วงหน้าก่อนที่จะเกิดการร้องขอจริง โดยข้อมูลหลักที่จะนำมาใช้เป็นข้อมูลในการทำนายมีดังนี้

3.2.1 เซิร์ฟเวอร์ล็อกไฟล์ (SERVER LOG)

โดยทั่วไปเว็บเซิร์ฟเวอร์จะทำการสร้างบันทึกการร้องขอที่เรียกว่า "ล็อก" ของเว็บบราวเซอร์ของผู้ใช้ไว้โดยกระบวนการนี้จะเป็นส่วนหนึ่งในการจัดการกับการร้องขอที่ได้รับจากเว็บไคลเอนต์ แต่ละเรคคอร์ดของล็อกจะแสดงการร้องขอแต่ละครั้งที่เกิดขึ้น เซิร์ฟเวอร์ล็อกไฟล์นั้นสามารถนำมาใช้ในการวิเคราะห์รูปแบบการเข้าถึงของผู้ใช้เพื่อใช้ในการทำนายการเข้าถึงในอนาคตได้ แต่อย่างไรก็ดีการร้องขอที่มีในเซิร์ฟเวอร์นั้นอาจจะไม่ถูกต้องทั้งหมด เพราะเซิร์ฟเวอร์ล็อก จะเก็บข้อมูลจากการร้องขอที่ถูกส่งเข้ามาที่เว็บเซิร์ฟเวอร์เท่านั้นดังนั้นการร้องขอที่เกิดจากผู้ใช้แต่ไม่ได้ถูกส่งมาจากเว็บเซิร์ฟเวอร์เพราะการร้องขอนั้นอาจจะถูกจัดการโดยพร็อกซี่หรือแคชของเว็บบราวเซอร์ของผู้ใช้เอง

สำหรับข้อมูลในแต่ละเรคคอร์ดในล็อกไฟล์นั้นส่วนใหญ่จะประกอบด้วย IP Address หรือโฮสต์เนมของไคลเอนต์ (Hostname), เวลาที่มีการร้องขอส่งเข้ามา (Access time), ชื่อของเว็บเพจหรือรีซอร์สที่ทำการร้องและเวอร์ชั่นของ HTTP ที่ใช้ในการร้องขอ

3.2.2 พร็อกซี่ล็อกไฟล์ (PROXY LOG)

พร็อกซี่นั้นจะทำงานคล้ายกับเป็นเว็บเซิร์ฟเวอร์ดังนั้นก็จะมีส่วนการทำงานที่ใช้ในการสร้างและบันทึกการร้องขอจากไคลเอนต์เช่นเดียวกับเว็บเซิร์ฟเวอร์ พร็อกซี่ล็อกนั้นจะให้ข้อมูลที่มีประโยชน์กว่าเซิร์ฟเวอร์ล็อก เพราะการร้องขอต่างๆ จะต้องผ่านที่พร็อกซี่ก่อนถึงจะมีการส่งการร้องขอนั้นไปยังเว็บเซิร์ฟเวอร์ปลายทาง ดังนั้นการใช้พร็อกซี่ล็อกในการวิเคราะห์รูปแบบการเข้าถึงของผู้ใช้เพื่อใช้ในการทำนายการร้องขอจึงมีความถูกต้องมากกว่า อีกทั้งพร็อกซี่ล็อกยังเก็บการร้องขอจากผู้ใช้ไปยังเว็บเซิร์ฟเวอร์ต่างๆ มากกว่าหนึ่งแห่ง ทำให้มีข้อมูลมากกว่าในการทำนายเมื่อเปรียบเทียบกับเซิร์ฟเวอร์ล็อกที่มีการบันทึกการร้องขอของเว็บเซิร์ฟเวอร์นั้นเท่านั้น

ยิ่งไปกว่านั้นพร็อกซี่ล็อกยังรวมไปถึงการร้องขอที่ถูกจัดการโดยพร็อกซี่เองด้วย ซึ่งตรงนี้เซิร์ฟเวอร์ล็อกจะไม่ได้มีการบันทึกไว้เพราะไม่มีการร้องขอส่งไปถึง แต่อย่างไรก็ดียังมีข้อเสียบางอย่างของพร็อกซี่ล็อกเช่นเดียวกับเซิร์ฟเวอร์ล็อก เช่นการร้องขอที่ได้ถูกจัดการโดยแคชของเว็บบราวเซอร์ที่ไคลเอนต์เองนั้น พร็อกซี่ล็อกจะไม่ได้มีการบันทึกไว้ และจากที่ได้กล่าวไว้แล้วในบท

ที่ 2 ว่าพร็อกซี่นั้นสามารถที่การทำงานเป็นลูกโซ่ได้ดังนั้นพร็อกซี่ล็อกของพร็อกซี่ที่อยู่ในใกล้ผู้ใช้มากที่สุดจะให้ข้อมูลรูปแบบการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆของผู้ใช้ได้ดีที่สุดเพราะพร็อกซี่ล็อกของพร็อกซี่ตัวห่างจากผู้ใช้ในลูกโซ่นั้นจะไม่สามารถจำแนกผู้ใช้ต่างๆได้เพราะจะถูกมองเป็นไคลเอนต์เดียวจากพร็อกซี่ลำดับก่อนหน้านั้น

3.2.3 ไคลเอนต์ล็อกไฟล์ (CLIENT LOG)

ไคลเอนต์ล็อกจะเป็นการบันทึกการร้องขอต่างๆที่เกิดขึ้นจากไคลเอนต์ ดังนั้นทุกๆการร้องขอจะถูกบันทึกไว้ในไคลเอนต์ล็อกรวมไปถึงการร้องขอที่ถูกจัดการโดยแคชของเว็บบราวเซอร์เองด้วย ซึ่งส่วนการบันทึกจะให้ข้อมูลที่ดีกว่าเซิร์ฟเวอร์ล็อกและพร็อกซี่ล็อก อีกทั้งไคลเอนต์ล็อกยังให้ข้อมูลเกี่ยวกับการร้องขอที่มีการยกเลิกจากผู้ใช้ด้วย ซึ่งส่วนตรงนี้เซิร์ฟเวอร์ล็อกและพร็อกซี่ล็อกไม่สามารถบันทึกได้ แต่ไคลเอนต์ล็อกก็มีข้อเสียในแง่ที่ว่าไม่สามารถให้ข้อมูลรูปแบบการเข้าถึงของผู้ใช้ในกรณีที่ผู้ใช้ไม่เคยเข้าเว็บนั้นๆเลย เพราะจะไม่มีข้อมูลรูปแบบการร้องขอของผู้ใช้อยู่เลย แต่ในพร็อกซี่ล็อกหรือเซิร์ฟเวอร์ล็อกจะสามารถพอที่จะทำนายการร้องขอสำหรับเว็บเพจที่ผู้ใช้ไม่เคยเข้าถึงได้โดยดูจากรูปแบบการเข้าถึงเว็บเพจนั้นของผู้ใช้คนอื่นๆ

3.2.4 ข้อมูลจากการเฝ้าดูระบบเครือข่าย (NETWORK MONITORING)

ไคลเอนต์, พร็อกซี่และเซิร์ฟเวอร์ล็อกนั้นจะต้องมีส่วนการทำงานของซอฟแวร์ที่เพิ่มเข้ามาเพื่อใช้ในการสร้างล็อกของการร้องขอที่เกิดขึ้น และข้อมูลในล็อกนั้นจะเป็นข้อมูลที่เกิดจากระดับแอพลิเคชั่นดังนั้นจึงไม่มีข้อมูลเกี่ยวกับการทำงานในระดับเน็ตเวิร์กเช่น TCP และ UDP เลย การเฝ้าดูระบบเครือข่ายเป็นการเก็บ และการติดตามข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆรูปแบบใหม่และเก็บเป็นล็อกไฟล์ไว้ โดยจะไม่มีผลกระทบการโปรแกรมเว็บเซิร์ฟเวอร์เลย แต่ส่วนการทำงานนี้จะทำงานอยู่ที่อุปกรณ์เครือข่ายเช่น เราเตอร์ แทน

การเฝ้าดูระบบเครือข่ายจะให้ข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆ ได้ถึงในระดับเน็ตเวิร์กคือ ไอพีเลเยอร์ (IP Layer) การเฝ้าดูสามารถดูเวลาของแต่ละแพคเกจของการร้องขอบนแต่ละลิงก์ของระบบเครือข่ายได้เลย ซึ่งเมื่อเปรียบเทียบกับเซิร์ฟเวอร์, พร็อกซี่และไคลเอนต์ล็อกแล้วเวลาที่สามารถดูได้จะเป็นเวลาที่มีค่าเดียวคือ เวลาที่การร้องถูกส่งมาถึงที่เซิร์ฟเวอร์หรือพร็อกซี่หรือที่เกิดจากไคลเอนต์เท่านั้น เวลาที่ได้จากการเฝ้าดูระบบเครือข่ายแต่ละลิงก์สามารถนำมาใช้พิจารณาในแง่ของความล่าช้าที่เกิดขึ้นในแต่ละลิงก์ได้ และด้วยการเฝ้าดูระบบเครือข่ายนั้นจะไม่เกี่ยวข้องกับโปรแกรมของเว็บเซิร์ฟเวอร์ดังนั้นจึงสามารถเก็บข้อมูลของการร้องขอได้อย่างเต็มที่เช่น อาจจะเก็บข้อมูลของ HTTP Header ได้ทั้งหมด

แต่อย่างไรก็ดีการเฝ้าดูระบบเครือข่ายนั้นก็ยังมีข้อเสียเช่นเดียวกับ เซิร์ฟเวอร์และพร็อกซี่ล็อกคือไม่สามารถที่จะเก็บข้อมูลการร้องขอที่ถูกจัดการโดยแคชของเว็บบราวเซอร์ที่ไคลเอนต์ได้ และ

ยังมีข้อเสียในกรณีที่มีการใช้การเข้ารหัสข้อมูลก่อนส่งระหว่างไคลเอนต์และเซิร์ฟเวอร์ การเฝ้าดูระบบเครือข่ายจะไม่สามารถเก็บข้อมูลได้ และมีข้อจำกัดในเรื่องของความเร็วในการเฝ้าดูข้อมูล เพราะถ้าลิงก์ที่ถูกใช้ในการเฝ้าดูนั้นมีความเร็วสูง การเฝ้าดูข้อมูลการร้องขอต่างๆจะต้องเร็วตามไปด้วยเพื่อให้สามารถเฝ้าดูได้ทันทำให้ต้องใช้เครื่องและอุปกรณ์ที่มีความเร็วสูงและสามารถรองรับความเร็วในการส่งข้อมูลได้

ทั้งหมดเป็นแหล่งข้อมูลที่จะใช้นำมาสร้างเป็นรูปแบบการเข้าถึงของผู้ใช้ที่เครื่องไคลเอนต์ต่างๆ แล้วนำรูปแบบการเข้าถึงนั้นมาใช้ทำนายการร้องขอเว็บเพจหรือรีซอร์สที่จะเกิดขึ้นในอนาคต ซึ่งแต่ละแหล่งข้อมูลก็มีข้อดีข้อเสียแตกต่างกันไป การใช้ไคลเอนต์ล็อกนั้นรูปแบบการเข้าถึงที่ได้ก็จะเหมาะกับไคลเอนต์ใดไคลเอนต์หนึ่งเท่านั้นและในกรณีที่ไคลเอนต์นั้นไม่เคยมีการเข้าถึงเว็บไซต์หนึ่งเลยก็จะไม่สามารถสร้างรูปแบบการเข้าถึงของไคลเอนต์นั้นได้ แต่ถ้ามีการใช้พร็อกซี่ล็อกนั้นก็จะไม่สามารถบันทึกการร้องขอที่เกิดขึ้นได้ทั้งหมด แต่การใช้พร็อกซี่ล็อกก็จะดีในส่วนที่ถ้าไคลเอนต์ใดๆไม่เคยเข้าถึงเว็บไซต์หนึ่งเลยก็สามารถเอารูปแบบการเข้าถึงของไคลเอนต์อื่นมาใช้ในการทำนายได้อีกทั้งพร็อกซี่ล็อกยังได้มาจากไคลเอนต์หลายๆไคลเอนต์ดังนั้นดังนั้นจึงมีข้อมูลในการเข้าถึงเว็บเซิร์ฟเวอร์ต่างๆมากกว่าไคลเอนต์ล็อก ส่วนเซิร์ฟเวอร์ล็อกนั้นจะมีการบันทึกการร้องขอที่ถูกส่งมาจากไคลเอนต์หรือพร็อกซี่เท่านั้น ข้อมูลในเซิร์ฟเวอร์ล็อกนั้นจะเป็นของเซิร์ฟเวอร์เดียวเท่านั้นในการนำมาสร้างเป็นรูปแบบการเข้าถึงของผู้ใช้ที่ไคลเอนต์ต่างๆจึงเหมาะสมสำหรับเว็บเซิร์ฟเวอร์เดียวเท่านั้น ส่วนการใช้ข้อมูลจากการเฝ้าดูระบบเครือข่ายนั้นจะได้เป็นข้อมูลที่มีรายละเอียดมากเพราะสามารถดูได้ทั้งในรูป TCP หรื UDP หรือ HTTP ซึ่งการใช้แหล่งข้อมูลนี้ไม่ค่อยนิยมใช้กันเพราะมีความเกี่ยวข้องกับนโยบายความเป็นส่วนตัว และมีความยุ่งยากในการนำมาสร้างรูปแบบการเข้าถึงของไคลเอนต์ด้วย

3.3 รูปแบบต่างๆ ของล็อกไฟล์

3.3.1 รูปแบบทั่วไปของล็อกไฟล์

ในแต่ละเรคคอร์ดของล็อกไฟล์จะแตกต่างกันไปขึ้นอยู่กับแอพลิเคชั่นที่ใช้บนเว็บเซิร์ฟเวอร์หรือพร็อกซี่เซิร์ฟเวอร์ แต่โดยทั่วๆไปจะประกอบด้วยข้อมูลหลัก 7 อย่างซึ่งแสดงดังตารางที่อยู่ข้างล่าง

ตารางที่ 3.1 แสดงข้อมูลทั่วไปของที่มีอยู่ในล็อกไฟล์

Field	Meaning
Remote host	Hostname or IP address of requesting client
Remote identity	Account associated with connection on client machine
Authenticated user	Name provided by user for authentication
Time	Date/time associated with the request
Request	Request method, Request-URI, and protocol version
Response code	Three-digit HTTP response code
Content length	Number of bytes associated with the response

- ชื่อโฮสต์ (Remote Host) : จะแสดงไอพีแอดเดรส(IP Address) ของเครื่องไคลเอนต์หรือจะเป็นชื่อของโฮสต์ก็ได้เช่น 10.245.131.2 หรือ Chaokhun.kmitl.ac.th เครื่องเซิร์ฟเวอร์สามารถที่จะหาไอพีแอดเดรสของเครื่องไคลเอนต์ได้โดยตรงจาก Socket ที่ใช้ในการส่งข้อมูลการร้องขอแบบ HTTP (HTTP Request) ได้เลย แต่อย่างไรก็ดีสำหรับการหาชื่อโฮสต์ของเครื่องไคลเอนต์นั้น จะต้องมีการส่งการแปลงจากไอพีแอดเดรสเป็นชื่อโฮลต์ (DNS Lookup) ไปยังดีเอนเอสเซิร์ฟเวอร์(DNS Server) เพื่อทำแปลงจากย้อนกลับจากไอพีเอดเดรสเป็นชื่อโอสต์(Reverse DNS Lookup) หาชื่อโฮลต์ของเครื่องไคลเอนต์ แต่โดยทั่วไปเว็บเซิร์ฟเวอร์จะไม่มีใช้การล็อกชื่อโฮสต์ เพราะต้องเสียเวลาในการส่งแปลงจากไอพีแอดเดรส เป็นชื่อโฮสต์ต่างๆ เพื่อไม่ทำให้ประสิทธิภาพของเว็บเซิร์ฟเวอร์ลดลง
- บัญชีผู้ใช้(Remote Identity) : จะแสดงชื่อของผู้ใช้ที่กำลังใช้เครื่องไคลเอนต์เชื่อมต่อ TCP เข้ามาที่เซิร์ฟเวอร์ โดยการเชื่อมต่อ TCP นี้จะถูกใช้ในการติดต่อสื่อสารระหว่างแอพลิเคชั่นที่ทำงานอยู่ที่เครื่องไคลเอนต์กับเครื่องเซิร์ฟเวอร์ แต่โดยทั่วไปฟิลด์นี้จะไม่มีการบันทึกไว้เพราะเว็บเซิร์ฟเวอร์ส่วนใหญ่ไม่มีความจำเป็นต้องเก็บข้อมูลส่วนนี้ ดังนั้นฟิลด์นี้จะปรากฏเป็น "-" ในล็อกไฟล์ส่วนใหญ่
- ข้อมูลของผู้ใช้ที่ได้รับสิทธิเข้ามา(Authenticated User) : จะเป็นส่วนที่แสดงชื่อของผู้ใช้ที่ได้มีการยืนยันตัวเองโดยผ่านกระบวนยืนยันการใช้งานบนโปรโตคอล HTTP ด้วย การใช้หัวข้อการยืนยันตัวเอง(Authentication Header) ในข้อมูลการร้องขอแบบ HTTP ที่จะถูกส่วนมาจากเครื่องไคลเอนต์ในกรณีที่ต้องมีการใช้ ชื่อผู้ใช้(User) และ รหัสลับ (Password) ในการเข้าถึงเว็บเพจหรือรีซอร์สใดๆ ในเซิร์ฟเวอร์ โดยทั่วไปฟิลด์นี้จะปรากฏเป็น "-" ถ้าไม่มีการใช้การแสดงสิทธิ หรือกรณีที่เซิร์ฟเวอร์ไม่มีการเก็บข้อมูลส่วนนี้

- เวลา(Time) : ในล็อกไฟล์นั้นจะต้องมีการเก็บเวลาที่ได้รับการร้องขอ โดยจะมีหน่วยเป็นวินาที, นาทีและชั่วโมงตามลำดับ แต่อย่างไรก็ดีในส่วนของฟิลด์นี้จะไม่ได้มีการกำหนดรูปแบบของเวลาที่มีการบันทึกไว้แน่นอน จะขึ้นกับแต่ละเซิร์ฟเวอร์ โดยเวลาที่มีการบันทึกนั้นจะสามารถใช้เป็นเวลา 4 แบบ อาจจะเป็นเวลาที่ได้รับการร้องขอ หรือจะเป็นเวลาที่เริ่มการส่งการตอบกลับไปยังไคลเอนต์ เวลาที่ส่งการตอบกลับเสร็จแล้วหรือจะเป็นเวลาที่มีการเริ่มสร้างข้อมูลการตอบกลับแบบ HTTP (HTTP Response)
- เว็บเพจหรือรีซอร์สที่ร้องขอ (Request) : ส่วนของฟิลด์นี้จะใช้เก็บรีซอร์สที่มีการร้องขอมาจากไคลเอนต์ โดยจะประกอบด้วย URI ของรีซอร์สและโปรโตคอลเวอร์ชั่นที่มีการส่งมา
- รหัสของข้อมูลการตอบกลับแบบ HTTP (Response Code) : เป็นส่วนที่จะแสดงรหัสในการตอบกลับไปแบบ HTTP ซึ่งจะอยู่ในรูปตัวเลข 3 ตำแหน่งเช่น 200 OK เป็นต้น
- ขนาดของข้อมูล (Content Length) : เป็นฟิลด์ที่ใช้ในการเก็บจำนวนของไบต์ที่ได้มีการส่งไปในข้อมูลการตอบกลับแบบ HTTP และก็ขึ้นกับเว็บเซิร์ฟเวอร์เช่นกันในการเก็บข้อมูลส่วนนี้ โดยอาจจะเป็นขนาดของข้อมูลในส่วนของ Entity body ในข้อมูลการตอบกลับแบบ HTTP หรือจะเป็นขนาดของการตอบกลับแบ HTTP ทั้งหมดก็ได้ ถ้าเซิร์ฟเวอร์ไม่มีการเก็บหรือไม่รู้ในส่วนนี้ก็จะเก็บเป็น "-" หรือเป็น 0 สำหรับในส่วนของการตอบกลับที่ไม่มีข้อมูลในส่วนของ Entity Body(304 Not Modified)

```
10.245.131.2 - - [15/Oct/2000:00:00:25 -0400] "GET /img/logo.gif
   HTTP/1.0" 304 0
10.245.131.2 - - [15/Oct/2000:00:00:26 -0400] "GET /img/site.gif
    HTTP/1.0" 304 0
10.3.2.16 - - [15/Oct/2000:00:00:38 -0400] "GET /est/bybhi.html
    HTTP/1.0" 304 0
10.16.202.34 - - [15/Oct/2000:00:00:45 -0400] "GET /img/tday.gif
    HTTP/1.0" 200 3699
```

รูปที่ 3.1 แสดงตัวอย่างของล็อกไฟล์

รูปที่ 3.1 แสดง 4 เรคคอร์ดในล็อกไฟล์ที่มีรูปแบบทั่วไป ในแต่ละเรคคอร์ดฟิลด์ที่สองและฟิลด์ที่สามจะเป็น "-" เพราะเซิร์ฟเวอร์ไม่มีการบันทึกบัญชีผู้ใช้และบัญชีผู้ใช้ที่ได้มีการยืนยันตัวเองมาก ส่วนในฟิลด์เวลา(Time) จะมีรูปแบบเป็น วัน เดือน ปี และตามด้วยเวลาท้องถิ่นของเซิร์ฟเวอร์ ส่วน "-0400" แสดงว่าเวลาท้องถิ่นของเซิร์ฟเวอร์เป็น 4 ชม.หลังจากเวลากลางกรีนนิช (Greenwich Mean Time)

3.3.2 รูปแบบเพิ่มเติมจากรูปแบบทั่วไป (Extended Common Log Format)

การบันทึกล็อกไฟล์ของเซิร์ฟเวอร์นั้นอาจจะมีสามารถเก็บข้อมูลเพิ่มเติมจากรูปแบบทั่วไปได้อีก โดยข้อมูลที่สามารถจัดเก็บได้จะเป็นดังตารางที่ 3.2

ตารางที่ 3.2 แสดงข้อมูลเพิ่มเติมที่อาจมีในล็อกไฟล์

Field	Meaning
User agent	Information on user agent software
Referer	URI from which Request-URI was obtained
Request processing time	Time spent processing the request
Request header size	Number of bytes in the request header
Request body size	Number of bytes in the request body
Remote response code	Response code from the server
Remote content length	Size of the response from server
Remote response header size	Size of the response header sent by server
Proxy request header size	Size of the request header sent to server
Proxy response header size	Size of the response header sent to client

- เว็บแอพลิเคชั่นที่ไคลเอนต์ใช้ (User Agent) : ในส่วนนี้จะเก็บบันทึกค่าของ User-Agent ที่อยู่ในหัวข้อของ User-agent ในข้อมูลการร้องของแบบ HTTP ที่ส่งมาจากไคลเอนต์ โดยจะแสดงชื่อและเวอร์ชั่นของแอพลิเคชั่นที่ได้ทำการส่งข้อมูลการร้องของแบบ HTTP มา ซึ่งส่วนนี้อาจจะมีข้อมูลเพิ่มเติมได้อีกเช่น ชื่อของระบบปฏิบัติการณ์ที่ทำงานอยู่ที่เครื่องไคลเอนต์
- เว็บเพจอ้างอิง(Referer) : ในส่วนนี้จะเก็บค่าของ Referer ที่อยู่ในหัวข้อ Referer ในข้อมูลการร้องขอแบบ HTTP ในกรณีถ้ามีการส่งข้อมูลส่วนนี้มากับการร้องของแบบ HTTP ที่มาจากไคลเอนต์
- เวลาที่ใช้ในการกระบวนผลการร้องขอแบบ HTTP (Request Processing Time) : เป็นฟิลด์ที่จะใช้บอกระยะเวลาที่มีการสร้างการตอบกลับแบบ HTTP (HTTP Response) ที่จะส่งกลับไปให้ไคลเอนต์ โดยมีหน่วยเป็นวินาที
- ขนาดของข้อมูลในส่วนหัวข้อที่มีมาในข้อมูลการร้องแบบ HTTP (Request Header Size) : ส่วนนี้จะเก็บบันทึกจำนวนไบต์ของหัวข้อของ HTTP ที่ได้รับจากข้อมูลการร้องแบบ HTTP (HTTP Request) ที่ส่งมาจากไคลเอนต์
- ขนาดของข้อมูลของที่มีใช้ส่วนเนื้อหา (Request Body Size) : ส่วนนี้จะบันทึกจำนวนไบต์ของส่วนเนื้อหา(Body) ของการร้องขอแบบ HTTP ที่ได้รับจากไคลเอนต์

โดยทั่วไป HTTP Request จะไม่มีข้อมูลในส่วนของ Body ยกเว้นกรณีที่ทางไคลเอนต์มีการส่งแบบ PUT หรือ POST

สำหรับข้อมูลส่วนที่เหลือในตาราง 3.2 จะเกี่ยวข้องกับพร็อกซี่เซิร์ฟเวอร์ ซึ่งเป็นส่วนที่อาจจะมีการบันทึกไว้ในพร็อกซี่ล็อก ประกอบด้วย

- รหัสของการตอบกลับปลายทาง (Remote Response Code) : ส่วนของฟิลด์นี้จะบันทึกในรูปของตัวเลข 3 ตำแหน่งที่จะประกอบด้วยสถานะของการตอบกลับแบบ HTTP (HTTP Response) ที่ได้รับมาจากเว็บเซิร์ฟเวอร์ปลายทางมายังพร็อกซี่นี้ ซึ่งรหัสการตอบกลับนี้นี้อาจจะแตกต่างจากรหัสการตอบกลับฟิลด์ (Response Code) ที่มีใช้ข้อมูลการตอบกลับแบบ HTTP ที่ส่งจากพร็อกซี่ไปยังไคลเอนต์ก็ได้
- ขนาดของข้อมูลปลายทาง(Remote Content Length) : จะเป็นจำนวนของไบต์ในข้อมูลการตอบกลับแบบ HTTP ที่ส่งมากจากเว็บเซิร์ฟเวอร์ปลายทางมายังพร็อกซี่ ซึ่งขนาดของข้อมูลปลายทาง นี้อาจจะแตกต่างจาก Content Length ฟิลด์ของ HTTP Response ที่ส่งจากพร็อกซี่ไปยังไคลเอนต์ก็ได้
- ขนาดของหัวข้อในข้อมูลการตอบกลับแบบ HTTP ปลายทาง(Remote Response Header Size) : เป็นขนาดที่เป็นจำนวนไบต์ของหัวข้อในข้อมูลการตอบกลับแบบ HTTP ที่ส่งมาจากเว็บเซิร์ฟเวอร์ปลายทางมายังพร็อกซี่
- ขนาดของหัวข้อในการร้องขอจากพร็อกซี่(Proxy Request Header Size) : เป็นจำนวนไบต์ของในส่วนหัวข้อของการร้องขอแบบ HTTP ที่ส่งจากพร็อกซี่ไปยังเว็บเซิร์ฟเวอร์ปลายทาง ซึ่งอาจจะแตกต่างจากขนาดของการส่วนของหัวข้อในการร้องแบบ HTTP ที่ส่งมาจากไคลเอนต์มายังพร็อกซี่ก็ได้
- ขนาดของส่วนหัวข้อในการตอบกลับแบบ HTTP ของพร็อกซี่(Proxy Response Header Size) : เป็นจำนวนไบต์ของส่วนหัวข้อที่มีในการตอบกลับแบบ HTTP ที่ส่งจากพร็อกซี่ไปยังไคลเอนต์

3.4 รูปแบบต่างๆของระบบพรีเฟรชชิ่ง

3.4.1 การพรีเฟรชชิ่งชื่อโฮสต์ (DNS Prefetching)

ก่อนที่จะมีการสร้างการเชื่อมต่อไปยังเว็บเซิร์ฟเวอร์นั้น ทางด้านแอพลิเคชั่นของผู้ใช้งานนั้นจะต้องมีการแปลงจากส่วนที่เป็นโฮสต์ของ URL ที่ถูกร้องขอให้ไปเป็นไอพีแอดเดรสก่อน ซึ่งกระบวนการนี้อาจจะต้องมีการติดต่อกับดีเอนเอสเซิร์ฟเวอร์ท้องถิ่น (Local DNS Server) หรือ

อาจจะต้องมีการติดต่อกับดีเอนเอสเซิร์ฟเวอร์ตัวอื่นในระดับที่สูงขึ้นไป ซึ่งทำให้มีการเสียเวลาในขั้นตอนนี้พอสมควร

เพื่อหลีกเลี่ยงหรือลดเวลาในกระบวนการแปลงจากชื่อโฮลต์เป็นไอพีแอดเดรสนั้น นั่นก็โดยจะทำการแปลงจากชื่อโฮลต์เป็นไอพีแอดเดรสล่วงหน้าก่อนที่ผู้ใช้จะมีการร้องขอเว็บเพจหรือรีซอร์สใดๆจากโฮลต์นั้น ซึ่งการทำงานแบบนี้ก็จะช่วยลดเวลาในกระบวนการที่เกี่ยวกับการแปลงชื่อหรือ URL เป็นไอพีแอดเดรส (DNS Service) และการทำงานของระบบพรีเฟรชชิ่งแบบนี้จะใช้ได้ดีและมีประสิทธิภาพ ถ้าในกระบวนการแปลงชื่อโฮลต์เป็นไอพีแอดเดรสของผู้ใช้ต้องมีการติดต่อกับดีเอสเอสเซิร์ฟเวอร์ในหลายๆระดับเพราะจะเสียเวลาในการกระบวนการนี้เยอะ ทำให้เกิดความล่าช้าในการเข้าถึงเว็บเพจหรือรีซอร์สต่างบนเวิลด์ไวด์เว็บ

3.4.2 การพรีเฟรชชิ่งการเชื่อมต่อแบบ TCP (Connection prefetching)

ในกระบวนการจัดการข้อมูลการร้องขอแบบ HTTP นั้นแอพลิเคชั่นทางด้านผู้ใช้หรือเว็บไคลเอนต์นั้น จะต้องมีการสร้างการเชื่อมต่อแบบ TCP ไปยังเว็บเซิร์ฟเวอร์หรือเซิร์ฟเวอร์ที่ให้บริการในการเข้าถึงเว็บเพจเช่น พร็อกซี่เซิร์ฟเวอร์ การเชื่อมต่อแบบ TCP นี้จะต้องมีการเริ่มต้นด้วยกระบวนการสร้างการเชื่อมต่อสามทาง (Three-way handshake) ก่อนระหว่างไคลเอนต์กับเซิร์ฟเวอร์ก่อน ซึ่งกระบวนการนี้การช่วงเวลาที่จะต้องใช้ไปในการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆ การลดช่วงเวลาในการสร้างการเชื่อมต่อแบบ Three-way handshake นี้ก็สามารถทำได้โดยการสร้างการการเชื่อมต่อแบบ TCP ไว้ก่อนเลยล่วงหน้าเลย ดังนั้นเมื่อผู้ใช้มีการคลิกลิงก์ต่างๆ บนเว็บเพจแอพลิเคชั่นของผู้ใช้ก็สามารถจะส่งการร้องขอแบบ HTTP ไปได้เลย ไม่ต้องเสียเวลาในการสร้างการเชื่อมต่อแบบ TCP อีก

ดังที่ได้กล่าวแล้วว่าการพรีเฟรชชิ่งการเชื่อมต่อแบบ TCP คือกระบวนการที่จะพรีเฟรชชิ่งการเชื่อมต่อ TCP ล่วงหน้านั่นเองซึ่งสมมติว่าหน้าเว็บเพจหน้าหนึ่งมีลิงก์ไปยังเว็บเซิร์ฟเวอร์ต่างๆ ในกระบวนการพรีเฟรชชิ่งแบบนี้ก็จะเลือกที่จะทำการสร้างการเชื่อมต่อแบบ TCP ไปยังเว็บเซิร์ฟเวอร์ต่างๆ ไว้ก่อนเมื่อดังนั้นเมื่อเกิดการคลิกที่ลิงก์ขึ้น เว็บแอพลิเคชั่นก็สามารถที่จะส่ง HTTP Request ไปยังเว็บเซิร์ฟเวอร์ได้เลย

ข้อจำกัดบางอย่างที่จะต้องพิจารณาด้วยสำหรับการทำงานของการพรีเฟรชชิ่งการเชื่อมต่อแบบ TCP นั่นก็คือการที่มีการสร้างการเชื่อมต่อแบบ TCP ทิ้งไว้ล่วงหน้านั้น ในกรณีถ้าการเชื่อมต่อนี้ไม่ได้มีการคลิกเกิดขึ้นแม้ว่าการเชื่อมต่อนั้นถูกปิดไปเอง แต่ก็จะทำให้สูญเสียทรัพยากรของเว็บเซิร์ฟเวอร์ไปเปล่าๆ

3.4.3 การพรีเฟรชชิ่งการร้องขอแบบ HTTP (HTTP prefetching)

การดูเว็บเพจหรือรีซอร์สต่างๆ นั้นจะอยู่ในรูปแบบที่ว่าผู้ใช้จะคลิกลิงก์ต่างๆบนเว็บเพจเพื่อใช้ในการดูเว็บเพจหน้าต่อไปหรือเว็บเพจอื่น ดังที่ได้กล่าวในบทก็หน้านี้ว่าเมื่อมีการคลิกลิงก์บนเว็บเพจเว็บแอพลิเคชั่นหรือเว็บบราวเซอร์จะมีการสร้างการเชื่อมต่อแบบ TCP ไปยังเว็บเซิร์ฟเวอร์ จากนั้นก็จะทำการส่งการร้องขอแบบ HTTP ไปยังเว็บเซิร์ฟเวอร์ ซึ่งระยะเวลาในการที่จะได้รับการตอบกลับแบบ HTTP นั้นก็ขึ้นอยู่ปัจจัยหลายอย่าง ไม่ว่าจะเป็นความเร็วของการประมวล ผลของเซิร์ฟเวอร์ปลายทาง, ปริมาณข้อมูลหรือความคับคั่งของเครือข่าย ณ เวลานั้น, ขนาดของเว็บเพจหรือรีซอร์สที่มีการร้องขอ ซึ่งถึงแม้ในกรณีของรีซอร์สที่มีการแคชไว้แล้วนั้นก็อาจจะยังคงต้องมีการส่งการร้องขอแบบ HTTP ไปยังเว็บเซิร์ฟเวอร์อยู่ดีเพื่อตรวจสอบว่ารีซอร์สที่ได้มีการแคชไว้นั้นมีการแก้ไขหรือไม่ที่เว็บเซิร์ฟเวอร์

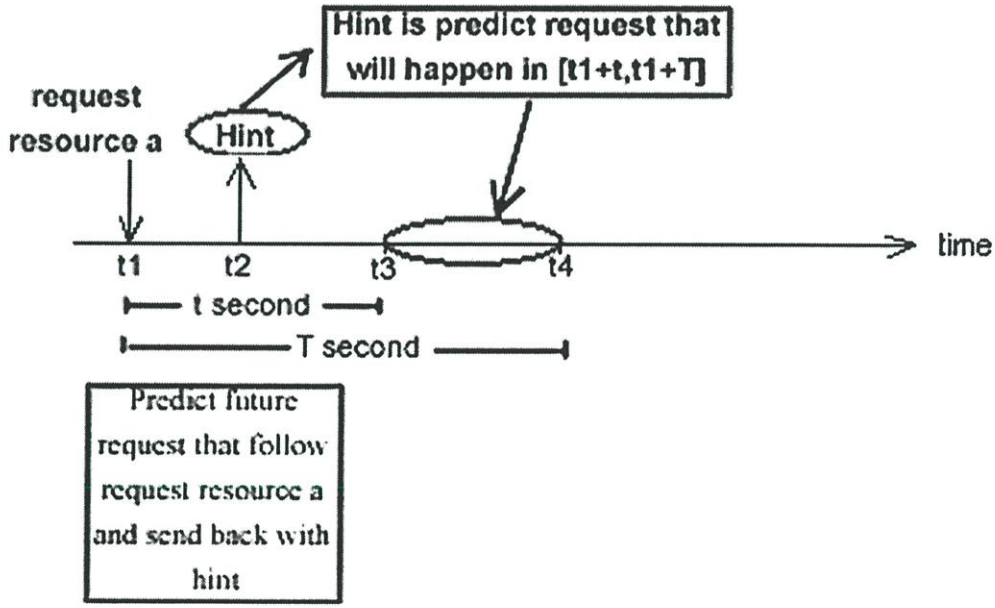
ในช่วงเวลาที่ผู้ใช้กำลังดูหรืออ่านเว็บเพจอยู่นั้นจะเป็นช่วงเวลาที่ช่องสัญญาณของผู้ใช้นั้นว่างเพราะไม่ได้มีการส่งการร้องขอแบบ HTTP ออกไป ดังนั้นในช่วงเวลานี้ถ้ามีการส่งการร้องขอแบบ HTTP เว็บเพจหรือรีซอร์สที่ถูกพิจารณาแล้วว่าผู้ใช้ต้องมีการร้องขอต่อไป เมื่อมีการได้รับการตอบกลับแบบ HTTP ก็จะทำการเก็บเว็บเพจหรือรีซอร์สนั้นไว้ที่แคชก่อน จากนั้นถ้าผู้ใช้มีการเข้าถึงเว็บเพจหรือรีซอร์สนั้นต่อไปจริงๆ เว็บแอพลิเคชั่นหรือเว็บบราวเซอร์ก็สามารถนำเว็บเพจหรือรีซอร์สนั้นๆมาแสดงจากแคชได้เลย สำหรับการทำงานแบบนี้ยังสามารถใช้กับในกรณีที่ทำการตรวจสอบว่าเว็บเพจหรือรีซอร์สใดที่ได้แคชไว้มีการแก้ไขที่เว็บเซิร์ฟเวอร์ได้อีกด้วย ที่กล่าวมาข้างต้นนั่นก็คือกระบวนการพรีเฟรชชิ่งการร้องขอแบบ HTTP (HTTP prefetching) ที่จะทำการพรีเฟรชชิ่งเว็บเพจหรือรีซอร์สต่างๆล่วงหน้าก่อนที่จะมีการร้องขอจากผู้ใช้

สำหรับวิธีการนี้จุดสำคัญอยู่ที่กระบวนการที่จะใช้ในการพิจารณาว่าเว็บเพจหรือรีซอร์สใดจะถูกเข้าถึงต่อไปหรือล่วงหน้า ซึ่งยิ่งสามารถพิจารณาได้ถูกต้องมากเท่าไหร่การพรีเฟรชชิ่งแบบนี้ก็จะมีประสิทธิภาพเพิ่มขึ้นตามไปด้วย ซึ่งในวิทยานิพนธ์นี้จะนำเสนอวิธีการที่มีประสิทธิภาพมากวิธีการหนึ่งที่จะนำมาใช้ในการทำนายหรือพิจารณาเว็บเพจหรือรีซอร์สที่จะมีการเข้าถึงจากผู้ใช้ล่วงหน้า

บทที่ 4
ทฤษฎีและการทำงานของแพร์ไวส์อัลกอริทึม

4.1 ทฤษฎีของแพร์ไวส์อัลกอริทึม

แพร์ไวส์อัลกอริทึมเป็นอัลกอริทึมที่ใช้ในการทำนายการร้องเว็บเพจหรือรีซอร์สใดๆ ที่จะเกิดขึ้นในอนาคตของผู้ใช้ที่เครื่องไคลเอนต์โดยจะทำการประมวลผลเซิร์ฟเวอร์ล็อกไฟล์เพื่อใช้ในการสร้างรูปแบบการเข้าถึงของผู้ใช้ และนำข้อมูลนั้นมาใช้ในการทำนายการร้องในอนาคตของผู้ใช้ รูปแบบการทำนายของแพร์ไวส์อัลกอริทึมจะเป็นดังรูป



รูปที่ 4.1 รูปแบบการทำนายของแพร์ไวส์อัลกอริทึม

จากรูปที่ 4.1 สามารถอธิบายได้ว่า ถ้าสมมติเมื่อเวลา t1 เว็บเซิร์ฟเวอร์ได้รับการร้องขอแบบ HTTP (HTTP Request) ที่เป็นการร้องขอรีซอร์ส A จากเว็บบราวเซอร์ของผู้ใช้ที่เครื่องไคลเอนต์ ทางเว็บเซิร์ฟเวอร์ใช้เวลาในการดึงข้อมูลรีซอร์ส A และทำการสร้างเป็นข้อมูลการตอบกลับแบบ HTTP (HTTP Response) และทำการดึงเอากลุ่มของ URL ของเว็บเพจหรือรีซอร์สที่เว็บเซิร์ฟเวอร์ทำนายว่าผู้ใช้จะต้องเข้าถึงหลังจากเข้าถึงรีซอร์ส A ภายในช่วงระยะเวลา t3 ถึง t4 (นั่นก็คือช่วงเวลา t ถึง T

วินาทีจากเวลา t1) โดยจะเรียกกลุ่มของ URL นี้ว่า "ฮิ้น(Hint)" จากนั้นก็ทำการส่งฮิ้นกลับมาพร้อมกับข้อมูลของรีซอร์ส A ไปยังเว็บบราวเซอร์ของผู้ใช้ จากนั้นเว็บบราวเซอร์ก็จะทำการแสดงผลรีซอร์ส A และระหว่างที่ผู้ใช้กับอ่านหรือดูรีซอร์ส A อยู่นั้น เว็บบราวเซอร์ก็จะทำการพรีเฟรช(Pre-fetch) เว็บเพจหรือรีซอร์สตาม URL ที่มีในฮิ้นที่ได้รับจากเว็บเซิร์ฟเวอร์ เมื่อพรีเฟรชแล้วจะทำการเก็บไว้ที่แคช ถ้าการทำงานถูกต้องผู้ใช้ต้องการเข้าถึงเว็บเพจหรือรีซอร์สตามที่ได้ทำนายไว้ เว็บบราวเซอร์ก็สามารถที่จะนำเว็บเพจหรือรีซอร์สนั้นจากแคชขึ้นมาแสดงผลได้เลย ไม่ต้องเสียเวลาในการร้องขอจากเว็บเซิร์ฟเวอร์

กระบวนการข้างต้นนั้นทางฝั่งไคลเอนต์ไม่จำเป็นต้องเป็นเว็บบราวเซอร์เสมอไป การทำงานดังกล่าวสามารถเอามาใช้กับพร็อกซี่ได้เช่นกัน โดยพร็อกซี่จะทำงานเช่นเดียวกับเว็บบราวเซอร์แต่ถ้ามีการทำงานที่พร็อกซี่นั้นอาจจะได้เปรียบกว่าการทำงานที่เว็บบราวเซอร์เพราะถ้ามีการแคชที่พร็อกซี่แล้วผู้ใช้คนอื่นของพร็อกซี่จะได้รับประโยชน์จากการพรีเฟรชด้วย

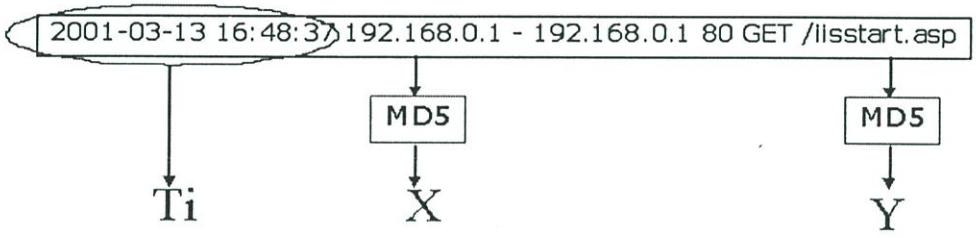
4.2 การทำงานของแพร์ไวส์อัลกอริทึม

ส่วนประกอบที่สำคัญต่างๆ ของแพร์ไวส์อัลกอริทึมมีดังนี้

- รีซอร์ส A หมายถึง เว็บเพจหรือรีซอร์สใดๆ ที่มีการเข้าถึงโดยผู้ใช้ที่เครื่องไคลเอนต์
- รีซอร์ส B หมายถึง เว็บเพจหรือรีซอร์สใดๆ ที่มีการเข้าถึงโดยผู้ใช้ที่เครื่องไคลเอนต์หลังจากเข้าถึงรีซอร์ส A แล้ว ภายในไปเกินช่วง [t,T]
- เคาเตอร์ C_a เป็นเคาเตอร์ที่จะใช้นับจำนวนครั้งที่เกิดการร้องขอรีซอร์ส A ของผู้ใช้ทุกๆคนที่มีการร้องขอมาที่เว็บเซิร์ฟเวอร์ในล็อกไฟล์
- เคาเตอร์ $C_{b|a}$ เป็นเคาเตอร์ที่จะใช้นับจำนวนครั้งที่เกิดการร้องขอรีซอร์ส B ใดๆของผู้ใช้ทุกๆคนที่มีการร้องขอมาที่เว็บเซิร์ฟเวอร์ที่เกิดหลังจากเข้าถึงรีซอร์ส A แล้วและถูกร้องขอจากผู้ใช้คนเดียวกันด้วยในล็อกไฟล์
- เคาเตอร์ $P_{b|a}$ เป็นความน่าจะเป็นที่ผู้ใช้จะมีการร้องขอรีซอร์ส B ภายหลังจากรีซอร์ส A แล้วไม่น้อยว่า t วินาทีแต่ไม่เกิน T วินาที โดยสามารถหา $P_{b|a} = C_{b|a} / C_a$
- ความน่าจะเป็นเทรสโฮส (P_t) เป็นความน่าจะเป็นที่จะใช้เป็นเกณฑ์ในการพิจารณาว่าเว็บเพจหรือรีซอร์ส B ใดบ้างจะถูกใส่เข้าไปใน V_a

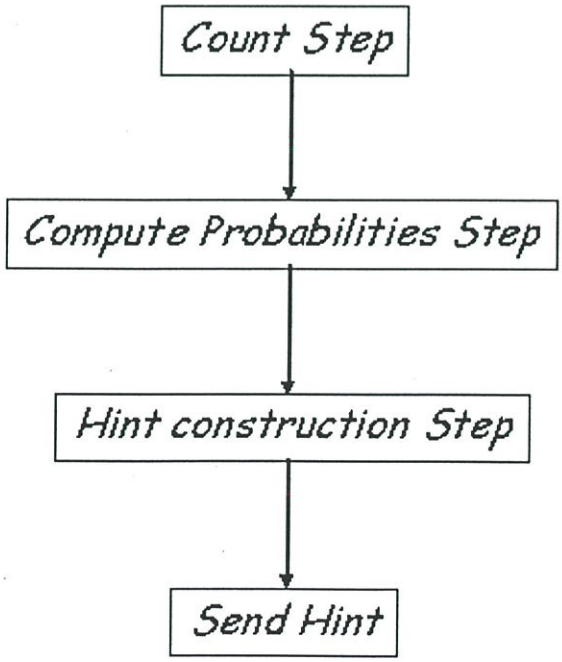
แพร์ไวส์อัลกอริทึมจะทำการประมวลผลล็อกไฟล์เพื่อทำงานสร้างความน่าจะเป็นที่รีซอร์ส B ใดๆ ขึ้นกับรีซอร์ส A ใด แต่ก่อนจะเริ่มการทำงานของแพร์ไวส์ต้องมีกระบวนการ "Clean" ล็อกไฟล์ก่อนดังนี้

1. จะทำการลบหรือนำเรคคอร์ดภายในล็อกไฟล์ที่เป็นการร้องขอเว็บเพจหรือรีซอร์สที่เป็น CGI, ASP, JSP ออกไปแล้ว
2. จากนั้นก็เอาล็อกไฟล์ที่ผ่านจากขั้นตอนแรกมาอ่านทีละเรคคอร์ด และจะเก็บเฉพาะไอพีแอดเดรส (IP Address) ของไคลเอนต์,รีซอร์สที่มีการร้องขอและเวลาที่มีการร้องขอ แล้วทำการเข้ารหัสไอพีแอดเดรสและชื่อของรีซอร์สที่มีการร้องขอด้วยอัลกอริทึม MD5 ในแต่ละเรคคอร์ด โดยจะได้เป็น {Ti, X, Y} เมื่อ Ti เป็นเวลาที่มีการร้องขอ, X เป็นไอพีแอดเดรสของไคลเอนต์และ Y เป็นเว็บเพจหรือรีซอร์สที่มีการร้องขอ ดังรูปที่ 4.2



รูปที่ 4.2 ขั้นตอนอ่านและประมวลผลเรคคอร์ดในล็อกไฟล์

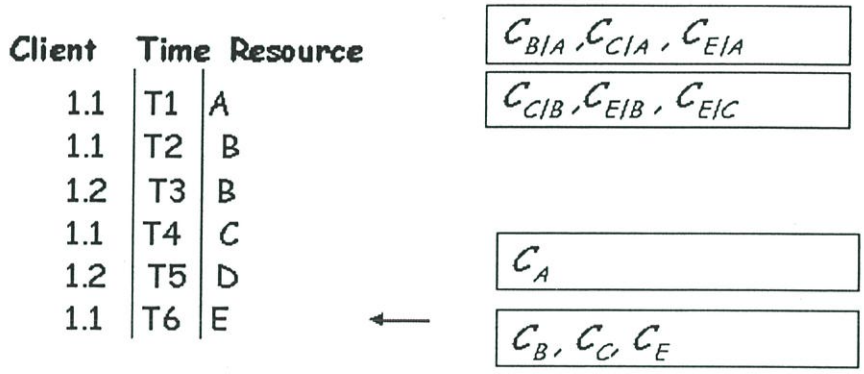
ขั้นตอนการทำงานของแพร์ไวส์อัลกอริทึมจะประกอบด้วย 4 ขั้นตอนคือ



รูปที่ 4.3 สรุปขั้นตอนของแพร์ไวส์อัลกอริทึม

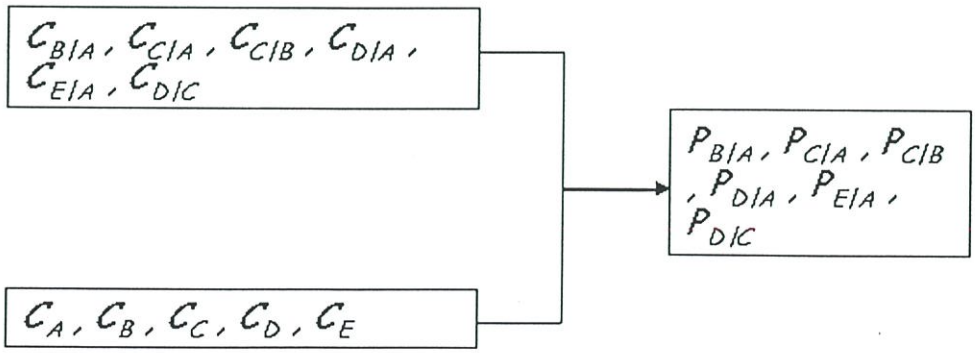
รายละเอียดของแต่ละขั้นตอนเป็นดังนี้

1. ทำการนับจำนวนการร้องขอเว็บเพจหรือรีซอร์ส A (C_a) และนับจำนวนการร้องขอเว็บเพจหรือรีซอร์ส B ที่หลังจากการร้องขอเว็บหรือรีซอร์ส A ($C_{b|a}$) แล้วในช่วงไม่น้อยกว่า t วินาทีและไม่เกิน T วินาที([t,T]) หลังจากได้รับการร้องขอเว็บเพจหรือรีซอร์ส A แล้ว (ดู flowchart การทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 1)



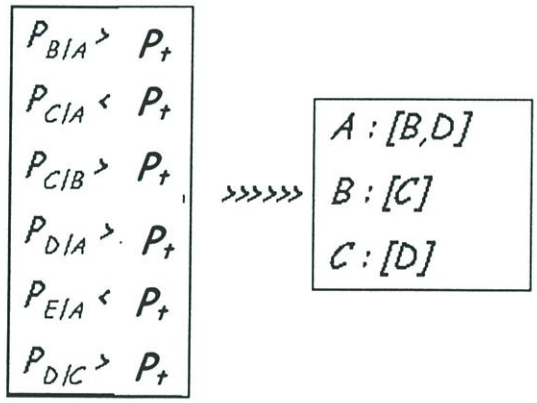
รูปที่ 4.4 ขั้นตอนการนับของแพร์ไวส์อัลกอริทึม

2. เมื่อทำตามขั้นตอนที่ 1 แล้วก็จะได้กลุ่มของ $C_{b|a}$ และกลุ่มของ C_a จากนั้นก็นำมาทำการหาความน่าจะเป็นของการขึ้นต่อกันของเว็บเพจหรือรีซอร์ส B ใดๆ กับเว็บเพจหรือรีซอร์ส A ใด($P_{b|a}$) ตามสูตรที่ได้กล่าวมาแล้ว เมื่อเสร็จขั้นตอนนี้แล้วก็จะได้เป็นกลุ่มของ $P_{b|a}$ (ดู flowchart การทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 2)



รูปที่ 4.5 ขั้นตอนการหาความน่าจะเป็นของการขึ้นต่อกันของแพร์ไวส์อัลกอริทึม

3. ขั้นตอนนี้ก็จะพิจารณาว่า $P_{b|a}$ ใดที่มีค่ามากกว่าหรือเท่ากับ P_t ก็จะนำเว็บเพจหรือรีซอร์ส B นั้นใส่เข้าไปเป็นโวลลุ่มของเว็บเพจหรือรีซอร์ส A (V_a) และทำอย่างนี้กับทุก $P_{b|a}$ ที่ได้จากขั้นตอนที่ 4 (ดู flowchart การทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 3)



รูปที่ 4.6 ขั้นตอนการสร้างฮิ้นโวลลุ่มของแพร์ไวส์อัลกอริทึม

4. เป็นขั้นตอนสุดท้ายในการใช้งาน คือเมื่อมีการร้องขอเว็บเพจหรือรีซอร์ส A มาที่เว็บเซิร์ฟเวอร์แล้ว ทางเว็บเซิร์ฟเวอร์จะไปดูว่าโวลลุ่มของเว็บเพจหรือรีซอร์ส A นี้มีอะไรบ้าง จากนั้นก็ส่งโวลลุ่มนั้นกลับไปพร้อมกับเว็บเพจหรือรีซอร์ส A เพื่อให้ทางไคลเอนต์ทำการพรีเฟรชต่อไป (ดู flowchart การทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 4)

สำหรับอัลกอริทึมของแพร์ไวส์อัลกอริทึมจะเขียนได้ดังตารางที่ 4.1

ตารางที่ 4.1 ตัวอย่างของแพร์ไวส์อัลกอริทึม

```
// ขั้นตอนการลบข้อมูลส่วนที่เป็น CGI, ASP, JSP
สำหรับแต่ละเรคคอร์ดในล็อกไฟล์
    ถ้ารูปแบบของไฟล์ไม่เป็น .CGI หรือ .ASP หรือ .JSP หรือ .PHP
        ทำการเก็บค่า Ti,ค่า MD5 ของไอพีของผู้ใช้, ค่า MD5 ของเว็บเพจที่ถูกเข้าถึง
```

ตารางที่ 4.1 (ต่อ)

```
// นับจำนวนการร้องขอ a แล้วตามด้วย b
สำหรับผู้ใช้คนที่ i ของผู้ใช้ที่มีการร้องขอมาทั้งหมด
  สำหรับเว็บเพจ a ที่ผู้ใช้ i
    C_a = C_a+1
    สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a ของผู้ใช้คนที่ i
      C_b|a = C_b|a + 1
// คำนวนค่า P_b|a ของแต่เว็บเพจ a
สำหรับเว็บเพจ a ที่ถูกร้องขอมา
  สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a
    P_b|a = C_b|a / C_a

// สร้างฮิ้น
สำหรับเว็บเพจ a ที่ถูกร้องขอมา
  สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a
    ถ้า P_b|a ใดมีค่ามากกว่า P_t แล้วให้เพิ่มเว็บเพจ b เข้าไปในฮิ้นของเว็บเพจ a

ผลลัพธ์ที่ได้ : กลุ่มของฮิ้นที่จะใช้กับผู้ใช้ทุกคน
```

4.3 ประสิทธิภาพของแพร์ไวส์อัลกอริทึม

ในการวัดประสิทธิภาพของแพร์ไวส์อัลกอริทึมนั้นจะใช้เซิร์ฟเวอร์ล็อกไฟล์จากเว็บเซิร์ฟเวอร์ที่เป็นที่นิยมแห่งหนึ่งในประเทศไทยในการสร้างรูปแบบการเข้าถึงของผู้ใช้ต่างๆที่มีการร้องขอจากเว็บเซิร์ฟเวอร์นี้เพื่อใช้ในการสร้างฮิ้น โดยจะแบ่งส่วนของล็อกไฟล์ออกเป็นสองส่วน ส่วนหนึ่งขนาดประมาณ 200,000 เรคคอร์ด และส่วนที่สองขนาดประมาณ 100,000 เรคคอร์ดในการทดสอบประสิทธิภาพของฮิ้นที่สร้างขึ้นมาได้

จะกำหนดให้ N_sendhint เป็นจำนวนครั้งที่มีการส่งฮิ้นโวลลุ่ม และแต่ละฮิ้นในฮิ้นโวลลุ่มจะมีการกำหนดค่า Hint_status ที่จะแสดงว่าฮิ้นนั้นมีการร้องขอจริงจากไคลเอนต์หรือไม่ ส่วน Hited-

number เป็นจำนวนครั้งที่มีการร้องขอเว็บเพจที่ตรงกับฮิ้นในฮิ้นโวลลุ่มที่ส่งไป ส่วน N_hinturl เป็นจำนวนฮิ้นในฮิ้นโวลลุ่ม สำหรับการวัดประสิทธิภาพของแพร์ไวส์อัลกอริทึมจะวัดในรูป 3 แบบคือ

- นอมัลไลซ์รีคอล (Normalized Recall) เป็นอัตราที่เว็บเพจหรือรีซอร์สที่ผู้ใช้กำลังเข้าถึงอยู่ในปัจจุบันถูกทำนายด้วยฮิ้นหรือไม่ เช่น ถ้าสมมติมีส่งฮิ้นไปว่าเมื่อผู้ใช้มีการเข้าถึงเว็บเพจ A แล้ว ผู้ใช้จะเข้าถึงเว็บเพจ B, C, D ด้วย และการเข้าถึงจริงของผู้ใช้เป็น A, B, E

$$Normalized\ Recall = \frac{\sum_{j=1}^{n_sendhint}\left(\frac{Hited-number\ of\ hinturl\ of\ a}{n_after\ \ a}\right)}{n_sendhint}$$

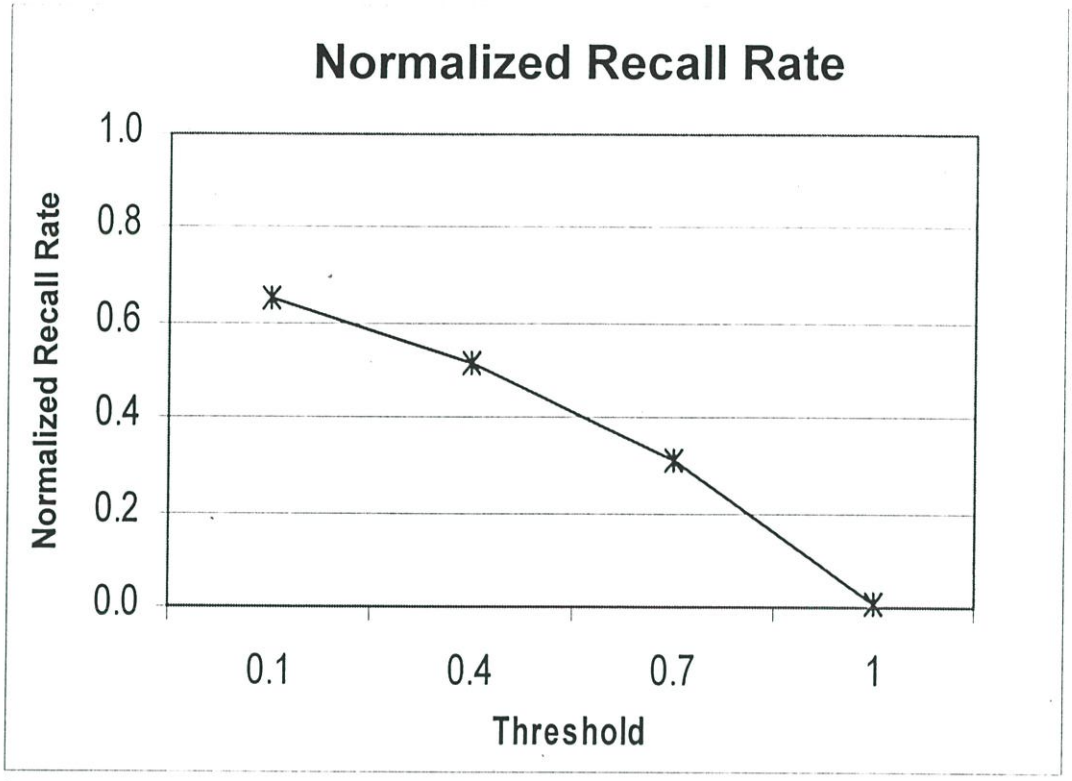
- นอมัลไลซ์พรีซีชั่น(Normalized Precision) เป็นอัตราที่เว็บเพจหรือรีซอร์สที่ถูกทำนายว่าผู้ใช้จะเข้าถึงในช่วงเวลา [t,T] ข้างหน้าแล้ว ผู้ใช้มีการเข้าถึงเว็บเพจหรือรีซอร์สเหล่านั้นถูกต้องเท่าไหร่

$$Normalized\ Precision = \frac{\sum_{j=1}^{n_sendhint}\left(\frac{\sum_{i=1}^{N_hinturl\ of\ a_i} Hited-status\ of\ hinturl\ of\ a_i}{n_hinturl}\right)}{n_sendhint}$$

- ขนาดเฉลี่ยของฮิ้น (Average Hintsize) เป็นขนาดเฉลี่ยของฮิ้นที่ได้มีการสร้างขึ้นทั้งหมด

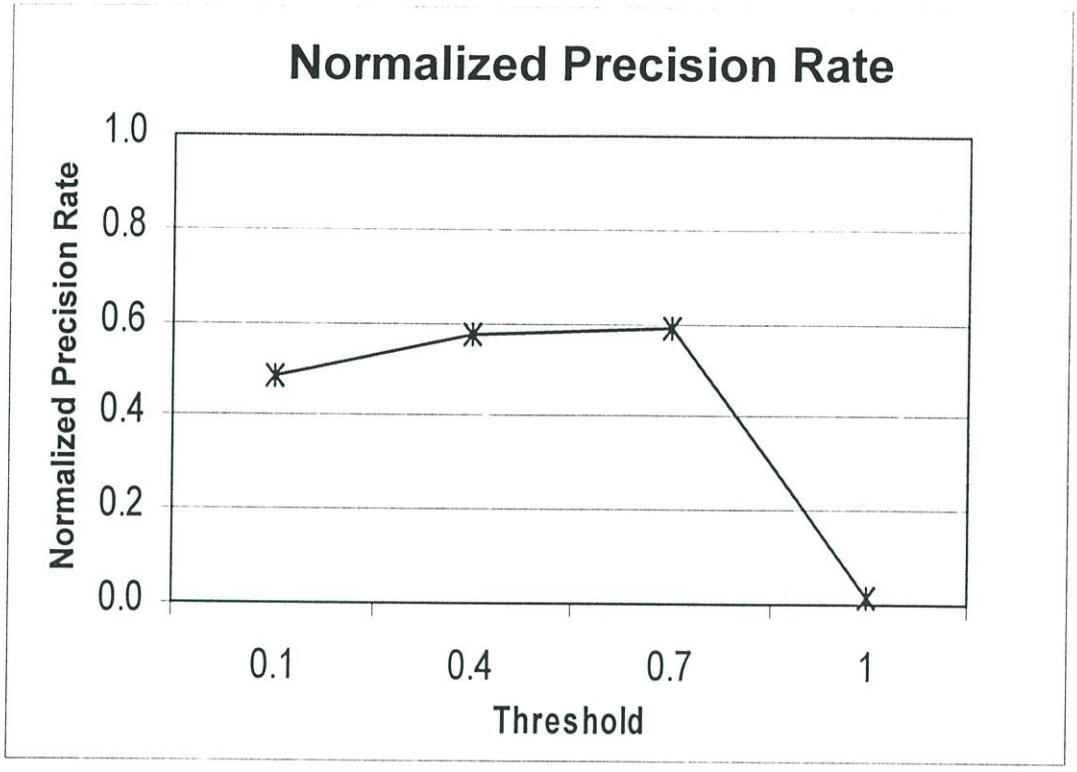
$$Average\ Hintsize\ = \frac{\sum_{i=1}^{n_sendhint} n_hinturl\ of\ a_i}{n_sendhint}$$

ละในการทดลองหาประสิทธิภาพจะมีการใช้ค่า P_t หลายค่าจากค่า 0.1 ถึง 1.0 และจะมีการใช้ช่วงระยะเวลาที่จะทำนายการเข้าถึงของผู้ใช้นั่นคือค่า t และ T หลายค่าเช่นกันโดยค่า t=0 คงที่แต่ค่า T จะมีค่าเป็น 30 ประสิทธิภาพของแพร์ไวส์อัลกอริทึมเป็นดังรูปข้างล่าง



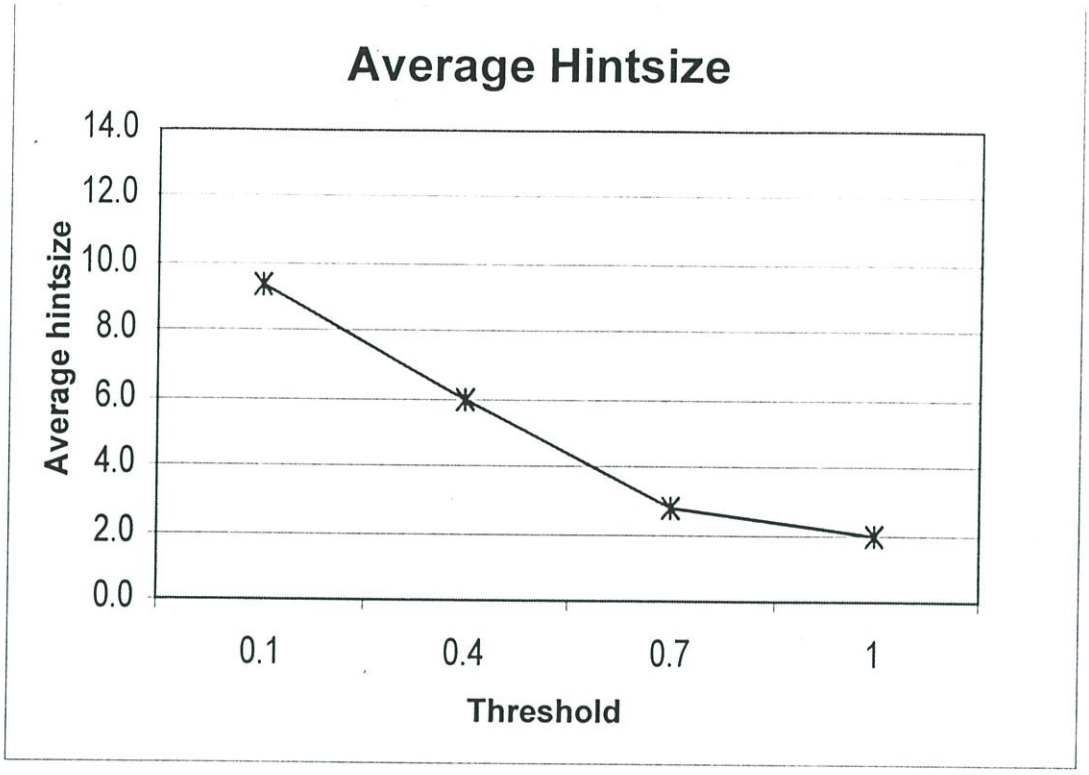
รูปที่ 4.7 อัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30

จากรูปที่ 4.7 แสดงอัตรานอมัลไลซ์รีคอลจะลดลงตามค่า P_t ที่เพิ่มขึ้นนั่นก็เพราะฮิ้นที่ได้มีขนาดเล็กลงเมื่อมีการใช้ค่า P_t ที่สูงทำให้อัตรานอมัลไลซ์รีคอลลดลงแปรผันตามค่า P_t



รูปที่ 4.8 อัตรานอมัลไลซ์พรีซีชั่นของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30

ส่วนรูปที่ 4.8 แสดงอัตรานอมัลไลซ์พรีซีชั่นที่จะเพิ่มขึ้นตามค่า P_t ที่เพิ่มขึ้นโดยอัตรานอมัลไลซ์พรีซีชั่นจะเพิ่มขึ้นเรื่อยจนถึงค่าของ P_t ค่าหนึ่ง แล้วอัตรานอมัลไลซ์พรีซีชั่นจะเริ่มลดลง แม้ว่าจะเพิ่มค่า P_t แล้วก็ไม่ได้ทำให้อัตรานอมัลไลซ์พรีซีชั่นเพิ่มขึ้นอีกแล้ว



รูปที่ 4.9 ขนาดเฉลี่ยของฮิ้นของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30

จากรูปที่ 4.9 แสดงความขึ้นต่อกันของขนาดเฉลี่ยของฮิ้นและค่าความน่าจะเป็นเทรสโฮลจะเห็นว่าเมื่อมีการใช้ค่าความน่าจะเป็นเทรสโฮลที่ค่าต่ำ จะได้ขนาดของฮิ้นที่มีขนาดใหญ่ ซึ่งจะทำให้ค่าอัตรานอมัลไลซ์รีคอลมีอัตราสูงเพราะยิ่งฮิ้นมีขนาดใหญ่ยิ่งทำให้มีโอกาสที่การร้องถัดไปจะอยู่ในฮิ้นมาก ในขณะที่ถ้ามีการใช้ค่าความน่าจะเป็นเทรสโฮลที่สูง จะได้ฮิ้นที่มีขนาดเล็กซึ่งทำให้อัตรานอมัลไลซ์รีคอลมีอัตราลดลงแต่ฮิ้นที่ได้จะมีความถูกต้องมากขึ้นเพราะมาจากข้อมูลจำนวนมากครั้ง ซึ่งจะเห็นว่าอัตรานอมัลไลซ์รีคอลกับอัตรานอมัลไลซ์พรีซีชั่นค่อนข้างเป็นส่วนกลับกัน ดังนั้นการหาค่าที่เหมาะสมระหว่างค่าความน่าจะเป็นเทรสโฮลที่ดีที่สุดที่ให้อัตรานอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พรีซีชั่นที่ดีที่สุดจึงเป็นสิ่งสำคัญมาก และการใช้ค่าช่วงเวลาที่ใช้ในการสร้างฮิ้น (Time Interval) นั้นจะขึ้นกับว่ามีการใช้กับโปรแกรมประยุกต์ใด ซึ่งโปรแกรมประยุกต์ใด หนึ่งอาจจะต้องใช้ช่วงเวลา 0-300 วินาทีถึงจะดีแต่กับอีกโปรแกรมประยุกต์อีกอันหนึ่งอาจจะใช้แค่ 0-30 วินาทีก็พอแล้ว

4.4 วิเคราะห์ประสิทธิภาพของแพร์ไวส์อัลกอริทึม

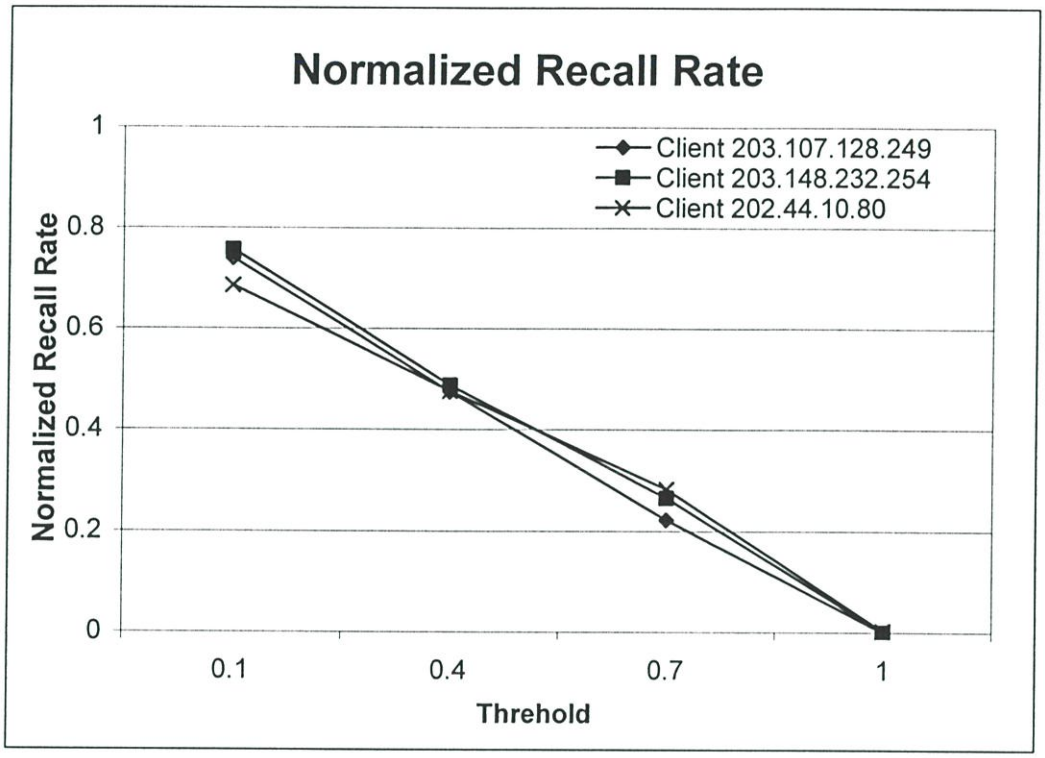
ซึ่งตามที่ได้กล่าวไปแล้วจะทำให้เมื่อมีการใช้ค่า P_t ที่สูงแล้วอัตรานอมัลไลซ์รีคอลที่ได้ควรที่จะน่าจะดีขึ้นเรื่อย กลับมีการลดลง จากรูปที่ 4.7 และจะเห็นได้ว่าอัตรานอมัลไลซ์พรีซีชั่นจะเพิ่มขึ้นตามค่า P_t ที่เพิ่มขึ้นนั่นก็เพราะว่าเมื่อมีการใช้ค่า P_t ที่สูงนั้น ฮิ้นที่ได้ก็จะมีความถูกต้องสูงเพราะถ้าค่า $P_{b|a}$ สูงแล้วนั่นก็หมายถึงว่ามีจำนวนผู้ใช้จำนวนมากที่ร้องขอเว็บเพจ B ต่อจากเว็บเพจ A แต่เมื่อมีใช้ค่า P_t ที่สูงมากๆใกล้ค่า 1 แล้วฮิ้นที่ได้นั้นมีขนาดเล็กๆมากหรือไม่ได้เลย ทำให้ค่านอมัลไลซ์พรีซีชั่นที่ได้จะลดลงอย่างเร็วมากซึ่งแสดงให้เห็นว่าการใช้ค่า P_t ที่สูงๆมากแล้วไม่ได้ทำให้นอมัลไลซ์อัตราพรีซีชั่นดีตามไปด้วย จากรูปที่ 4.8 ซึ่งแสดงให้เห็นถึงว่าฮิ้นที่ได้ไม่ค่อยมีประสิทธิภาพนัก

และสำหรับแพร์ไวส์อัลกอริทึมนั้นจะใช้ข้อมูลการร้องขอในอดีตมาใช้ในการสร้างรูปแบบการเข้าถึงเว็บเพจหรือรีซอร์สต่างของผู้ใช้ทุกคน ซึ่งตรงนี้นั้นเมื่อมีการใช้ค่า P_t ที่สูงแล้วก็จะให้ได้ฮิ้นที่มาจากรูปแบบการร้องขอของผู้ใช้ที่มีการร้องขอมากครั้งหรือผู้ใช้ที่มีอัตราการร้องขอต่อเว็บเพจทั้งหมดที่ร้องขอมีค่ามากเท่านั้น เพราะแพร์ไวส์อัลกอริทึมเองจะใช้ข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างบนเครื่องแม่ข่ายของผู้ใช้ทั้งหมดร่วมกันและนำข้อมูลนั้นมาสร้างเป็นกลุ่มของฮิ้นที่จะนำมาใช้งานกับผู้ใช้ทุกๆคน

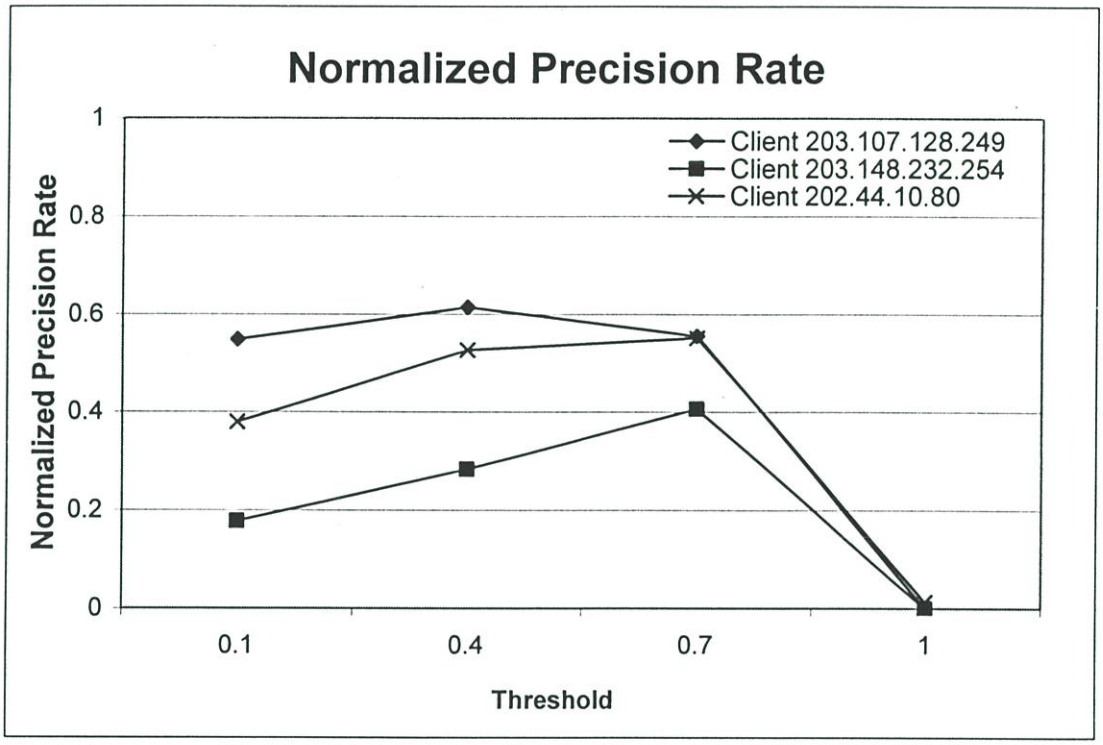
ดังนั้นในวิทยานิพนธ์นี้จึงได้ใช้ฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมมาจำลองสถานการณ์ตามที่ได้กล่าวไปแล้วแต่จะคำนวณอัตรานอมัลไลซ์รีคอล, นอมัลไลซ์พรีซิชั่นและขนาดของฮิ้นเฉลี่ยเป็นผู้ใช้คนๆไป เพื่อให้เห็นว่าเมื่อคิดค่าเฉลี่ยอัตรานอมัลไลซ์รีคอล, อัตรานอมัลไลซ์พรีซิชั่นของฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมเป็นคนๆ แล้ว ฮิ้น ของแพร์ไวส์อัลกอริทึมนั้นจะแสดงความไม่มีประสิทธิภาพเท่าที่ควร ความไม่มีประสิทธิภาพของฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมนั้นมาจากการใช้ข้อมูลร่วมกันของผู้ใช้ทั้งหมดนำมาสร้างเป็นกลุ่มของฮิ้นเพื่อนำมาใช้กับผู้ใช้คนต่างๆ นั่นเอง

จากรูปที่ 4.10 และรูปที่ 4.11 ซึ่งจะแสดงอัตรานอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พรีซิชั่นของผู้ใช้ที่มีการร้องขอมามากที่สุด 3 อันดับแรก จะเห็นได้ว่าแม้ว่าผู้ใช้ทั้งสามคนจะมีการร้องขอเว็บเพจหรือรีซอร์สต่างๆมากครั้งที่สุด แต่อัตรานอมัลไลซ์พรีซิชั่นและอัตรานอมัลไลซ์รีคอลก็ไม่ได้สูงตามไปด้วย ทั้งที่ผู้ใช้ทั้งสามน่าจะมีอัตราดังกล่าวในระดับสูงเพราะมีข้อมูลการร้องขอของผู้ใช้ทั้งสามมาก นั่นก็เพราะฮิ้นที่ได้นั้นมีส่วนของข้อมูลที่ได้จากผู้ใช้คนอื่นๆด้วยทำให้ฮิ้นดังกล่าวไม่ได้แสดงถึงรูปแบบหรือลักษณะการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆ ของผู้ใช้ทั้งสามได้ดีพอ

]



รูปที่ 4.10 อัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30 ของผู้ใช้ที่มีการร้องขอ 3 อันดับแรก



รูปที่ 4.11 อัตรานอมัลไลซ์พรีซีชั่นของแพร์ไวส์อัลกอริทึมเมื่อมีการใช้ P_t จาก 0.1 – 1 และใช้ t=0, T=30 ของผู้ใช้ที่มีการร้องขอ 3 อันดับแรก

ในบทถัดจะกล่าวถึงการปรับปรุงประสิทธิภาพของแพร์ไวส์อัลกอริทึมให้ดีขึ้น และสามารถสร้างฮิ้นที่มีประสิทธิภาพที่ให้อัตรานอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พรีซีชั่นที่ดีขึ้น

บทที่ 5
แพร์ไวส์อัลกอริทึมแบบใช้ 2 โทรสโฮล

จากบทที่แล้วได้มีการกล่างถึงข้อด้อยต่างๆของแพร์ไวส์อัลกอริทึมจึงได้มีการนำเสนอแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลเพื่อเพิ่มประสิทธิภาพของแพร์ไวส์อัลกอริทึมเดิม

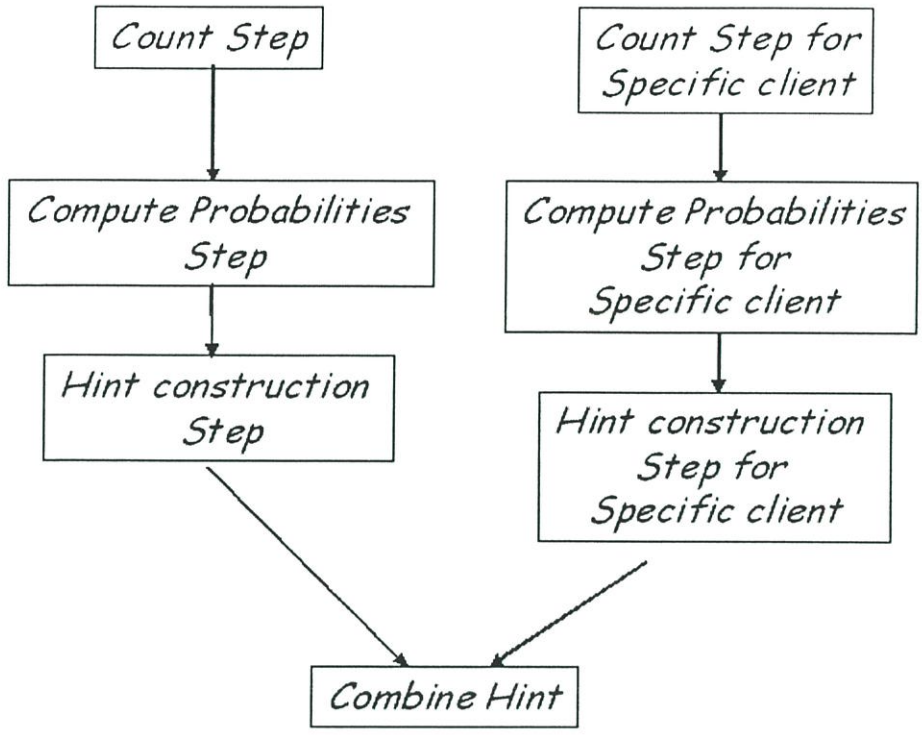
5.1 ทฤษฎีของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮล

แพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลนั้นจะมีการทำงานแบ่งได้เป็น 2 ส่วนที่ทำงานร่วม กัน คือ

1 . ส่วนที่มีการสร้างฮิ้นจากข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆของผู้ใช้ทั้งหมด การสร้างฮิ้นแบบนี้จะไม่มีการสนใจว่าผู้ใช้แต่ละคนเข้าถึงเว็บเพจหรือรีซอร์สอะไรบ้าง เพียงแต่สนใจว่าเซสชั่นในการเข้าถึงเว็บเพจหรือรีซอร์สของผู้ใช้นั้นเข้าถึงอะไรบ้าง แล้วนำข้อมูลนั้นมาใช้ในการสร้างฮิ้น จะเห็นได้ว่าการทำงานในส่วนนี้ก็คือเป็นแพร์ไวส์อัลกอริทึมเดิมนั่นเองและในการกล่าวถึงฮิ้นในส่วนนี้จะเรียกว่า “Old hint”

2. ส่วนที่มีการสร้างฮิ้นจากข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆของผู้ใช้แต่ละคน การสร้างฮิ้นแบบนี้คือจะเป็นส่วนที่เพิ่มเติมเข้ามาจากแพร์ไวส์อัลกอริทึมเดิม โดยจะมีการเก็บด้วยว่าผู้ใช้แต่ละคนนั้นเข้าถึงเว็บเพจหรือรีซอร์สอะไรบ้าง แล้วนำข้อมูลนั้นมาใช้ในการสร้างเป็นฮิ้นและในการกล่าวถึงฮิ้นในส่วนนี้จะเรียกว่า “New hint” ซึ่งฮิ้นที่ได้จะมีความเหมาะสมสำหรับผู้ใช้คนหนึ่งๆ

จากส่วนการทำงานทั้งสองส่วนจะได้ว่าฮิ้นที่ได้จากการทำงานส่วนแรกจะเป็นฮิ้นที่เหมาะสมกับผู้ใช้ทุกคน เพราะใช้ข้อมูลการเข้าถึงจากผู้ใช้ทั้งหมดในการสร้าง แต่ฮิ้นในส่วนที่สองนั้นเป็นฮิ้นที่เพิ่มจากแพร์ไวส์อัลกอริทึมเดิมจะเป็นฮิ้นที่เหมาะสมกับผู้ใช้เป็นคนๆไปเท่านั้น เพราะสร้างจากข้อมูลการเข้าถึงของผู้ใช้แต่ละคน หลังจากทำงานในทั้งสองส่วนแล้วเมื่อต้องการการทำการส่งฮิ้นก็จะทำการรวมเอาฮิ้นจากทั้งสองส่วนแล้วจะส่งไปให้ทางไคลเอนด์ต่อไป ขั้นตอนทั้งหมดสามารถสรุปได้ดังรูปที่ 5.1 และจากการที่ส่วนการทำงานแรกจะเหมือนแพร์ไวส์อัลกอริทึมเดิมตามที่ได้อธิบายไว้ในบทที่ 4 แล้ว ดังนั้นจะอธิบายแต่ส่วนการทำงานที่สองเท่านั้น



รูปที่ 5.1 ขั้นตอนทั้งหมดของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮล

5.1.1 ส่วนที่มีการสร้างฮิ้นจากข้อมูลการเข้าถึงเว็บเพจหรือรีซอร์สต่างๆของผู้ใช้แต่ละคน

ส่วนการทำงานนี้จะเป็นขั้นตอนที่คล้ายกับแพร์ไวส์อัลกอริทึมเดิม แต่จะแยกเก็บค่าเคาเตอร์ต่างๆของผู้ใช้เป็นคนๆไป รายละเอียดของแต่ขั้นตอนเป็นดังนี้รูปที่ 5.2 สำหรับคำนิยามต่างๆ จะเป็นเหมือนที่ใช้ในแพร์ไวส์อัลกอริทึมเดิมในบทที่ 4 แต่จะขอกำหนดความน่าจะเป็นเทรสโฮลของส่วนที่ 1 (แพร์ไวส์อัลกอริทึมเดิม) เป็น P_{t1} และส่วนความน่าจะเป็นเทรสโฮลของส่วนที่ 2 จะให้เป็น P_{t2}

5.1.1.1 การเก็บจำนวนครั้งของการขึ้นต่อกันของเว็บเพจ ($C_{b|a}$) ที่ผู้ใช้ต่อละคนเข้าถึง

ขั้นตอนนี้จะทำการนับจำนวนครั้งของการขึ้นต่อกันของเว็บเพจเช่นเดียวกับที่ทำในแพร์ไวส์อัลกอริทึมเดิม แต่จะมีการทำงานที่เพิ่มเติมก็คือจะแยกเก็บเคาเตอร์ต่างๆ แยกเป็นผู้ใช้คนๆ ไป ที่ต้อง ทำอย่างนี้ก็เพราะต้องการใช้เคาเตอร์ต่างๆของการขึ้นต่อกันของเว็บเพจต่างๆ ของแต่ละผู้ใช้ในการคำนวณหาความน่าจะเป็นของการที่ผู้ใช้คนใดคนหนึ่งร้องขอเว็บ A แล้วตามด้วยเว็บเพจ B ($C_{b|a}$) (ดู การทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 5)

5.1.1.2 การคำนวณหาความน่าจะเป็นของการขึ้นต่อกันของเว็บเพจ ($P_{b|a}$) ที่ผู้ใช้แต่ละคนเข้าถึง

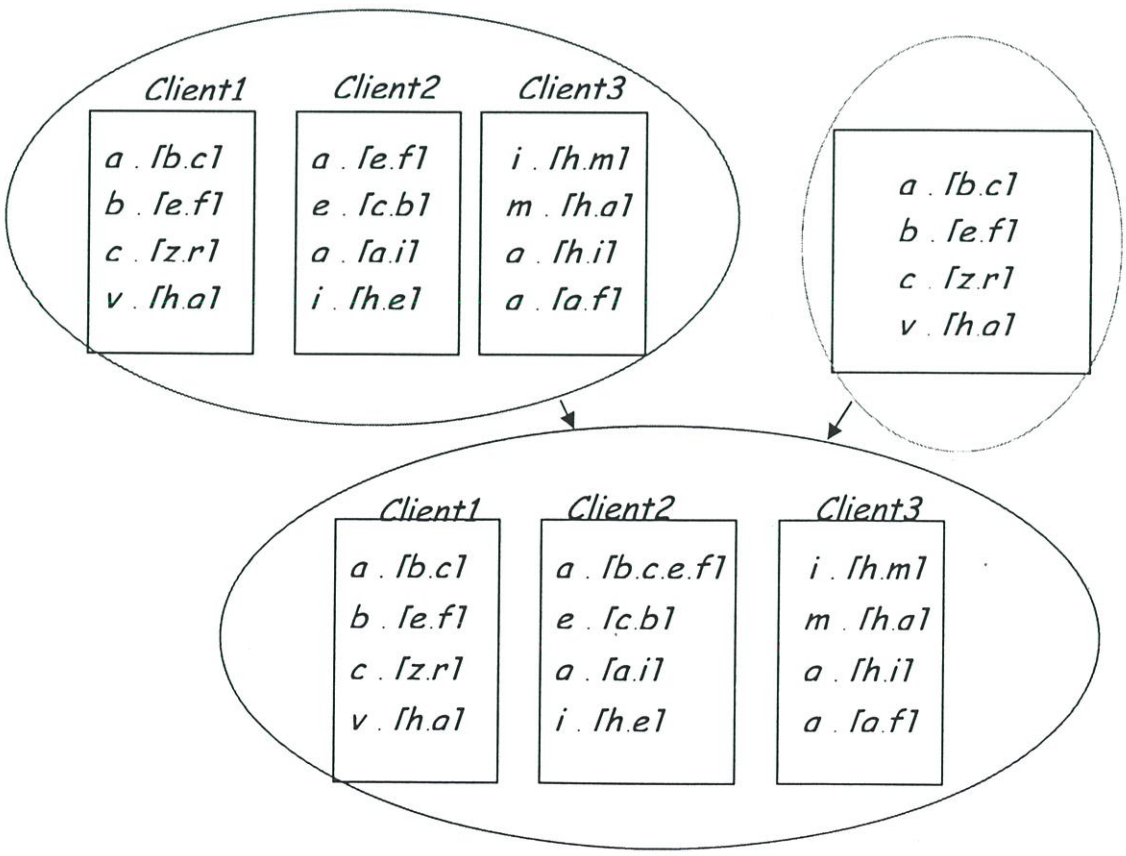
ขั้นตอนนี้จะทำเช่นเดียวกับแพร์ไวส์อัลกอริทึมเดิม แต่ที่แตกต่างคือจะแยกเก็บความน่าจะเป็นของการขึ้นต่อกันของเว็บเพจเป็นผู้ใช้คนๆ ไป(ดูการทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 6)

5.1.1.3 สร้างฮิ้นสำหรับผู้ใช้แต่ละคน

ขั้นตอนนี้ก็เป็นเช่นเดียวกับแพร์ไวส์อัลกอริทึมเดิม คือจะพิจารณาว่าความน่าจะเป็นของการขึ้นต่อกับของเว็บเพจB ใดๆขึ้นต่อ A ใดๆ ($P_{b|a}$) ถ้ามีค่ามากกว่าความน่าจะเป็นเทรสโฮล(P_{t2}) ก็จะนำเอาเว็บเพจ B นั้นใส่เข้าไปเป็นฮิ้นของเว็บเพจ A ของผู้ใช้คนหนึ่งๆ โดยที่ค่าความน่าจะเป็นเทรสโฮลนี้ (P_{t2}) อาจจะเป็นค่าเดียวกับค่า P_{t1} ของการทำงานในส่วนแรก (แพร์ไวส์อัลกอริทึมเดิม) ก็ได้ (ดู การทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 7)

5.1.1.4 การรวมฮิ้นที่ได้จะทั้ง 2 ส่วนการทำงาน

ขั้นตอนนี้จะทำการรวมฮิ้นที่ได้จากข้อมูลที่การเข้าถึงเว็บเพจของผู้ใช้ทุกๆคนมาสร้างฮิ้นและข้อมูลการเข้าถึงเว็บเพจของผู้ใช้แต่ละคนแยกกันมาสร้างเป็นฮิ้น มารวมกัน โดยจะเลือกเอาฮิ้นของทั้งสองส่วนที่ซ้ำจะเอามาแค่หนึ่งรวมเป็นฮิ้นสุดท้ายที่จะใช้ในการส่งไปให้เว็บไคลเอนต์เพื่อใช้ในการพรีเฟรชชิ่งต่อไป ดังรูปที่ 5.2 (ดู การทำงานได้ใน ภาคผนวก ข. Flowchart ที่ 8)



รูปที่ 5.2 การรวมฮิ้นของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮล

จากรูปที่ 5.2 สำหรับกลุ่มของฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลส่วนแรก(แพร์ไวส์อัลกอริทึมเดิม) คือกลุ่มที่อยู่วงกลมด้านบนขวา และกลุ่มของฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลส่วนที่สองจะเป็นส่วนที่อยู่ในวงกลมด้านบนซ้าย จะเห็นได้ว่าจะมีกลุ่มของฮิ้นเฉพาะของแต่ละผู้ใช้ไปภายในกลุ่มของฮิ้นทั้งหมด และสุดท้ายส่วนของฮิ้นที่เป็นที่อยู่ด้านล่างจะเป็นกลุ่มของฮิ้นที่ได้จากการรวมฮิ้นจากทั้งสองส่วนแล้วจะเห็นได้ว่าในส่วนของผู้ใช้คนที่ 2 จะได้ฮิ้นเพิ่มเติมจากการทำงานในส่วนแรกเข้าไปด้วย และฮิ้นสุดท้ายจะมีสำหรับผู้ใช้แต่คนไป เช่นถ้าผู้ใช้มีการร้องเว็บเพจ E มา ทางเว็บเซิร์ฟเวอร์ก็จะส่งฮิ้นของเว็บเพจ E ที่อยู่ในส่วนของฮิ้นของผู้ใช้คนที่ 2 ในกลุ่มของฮิ้นสุดท้ายที่ได้จากการรวมกันของฮิ้นกลับไป จากตัวอย่างที่ 5.2 นั่นก็คือ C, B

สำหรับอัลกอริทึมของแพร์ไวส์อัลกอริทึมที่มีการใช้เทรสโฮล 2 ระดับจะเป็นดังตารางที่ 5.1

ตารางที่ 5.1 ตัวอย่างของแพร์ไวส์อัลกอริทึมที่มีการใช้เทรสโฮล 2 ระดับ

```
// ขั้นตอนการลบข้อมูลส่วนที่เป็น CGI, ASP, JSP
สำหรับแต่ละเรคคอร์ดในล็อกไฟล์
    ถ้ารูปแบบของไฟล์ไม่เป็น .CGI หรือ .ASP หรือ .JSP หรือ .PHP
        ทำการเก็บค่า Ti,ค่า MD5 ของไอพีของผู้ใช้, ค่า MD5 ของเว็บเพจที่ถูกเข้าถึง

// นับจำนวนการร้องขอ a แล้วตามด้วย b (ส่วนที่หนึ่ง)
สำหรับผู้ใช้คนที่ i ของผู้ใช้ที่มีการร้องขอมาทั้งหมด
  สำหรับเว็บเพจ a ที่ผู้ใช้ i
    C_a = C_a+1
    สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a ของผู้ใช้คนที่ i
      C_b|a = C_b|a + 1
```

ตารางที่ 5.1(ต่อ)

// คำนวนค่า $P_{b|a}$ ของแต่เว็บเพจ a
สำหรับเว็บเพจ a ที่ถูกร้องขอมา
  สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a
    $P_{b|a} = C_{b|a} / C_a$

// สร้างฮิ้น
สำหรับเว็บเพจ a ที่ถูกร้องขอมา
  สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a
    ถ้า $P_{b|a}$ ใดมีค่ามากกว่า P_t แล้วให้เพิ่มเว็บเพจ b เข้าไปในฮิ้นของเว็บเพจ a

ผลลัพธ์ที่ได้ : กลุ่มของฮิ้นที่จะใช้กับผู้ใช้ทุกคน
---------- จบส่วนการทำงานที่หนึ่ง ----------

// นับจำนวนการร้องขอ a แล้วตามด้วย b ของผู้ใช้แต่คน
สำหรับผู้ใช้คนที่ i ของผู้ใช้ที่มีการร้องขอมาทั้งหมด
  สำหรับเว็บเพจ a ที่ผู้ใช้ i
    $C_a = C_a+1$ (C_a ของผู้ใช้คนที่ i)
    สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a ของผู้ใช้คนที่ i
      $C_{b|a} = C_{b|a} + 1$ ($C_{b|a}$ ของผู้ใช้คนที่ i)

// คำนวนค่า $P_{b|a}$ ของแต่เว็บเพจ a ของผู้ใช้แต่ละคน
สำหรับเว็บเพจ a ที่ถูกร้องขอมา
  สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a
    $P_{b|a} = C_{b|a} / C_a$ ($P_{b|a}$ ของผู้ใช้คนที่ i)

ตารางที่ 5.1(ต่อ)

```
// สร้างฮิ้นสำหรับแต่ละคน
สำหรับผู้ใช้คนที่ i ของผู้ใช้ทั้งหมด
   สำหรับเว็บเพจ a ที่ถูกร้องขอมาของผู้ใช้คนที่ i
      สำหรับเว็บเพจ b ที่ถูกร้องขอตามหลังเว็บเพจ a ของผู้ใช้คนที่ i
         ถ้า P_b|a ใดของผู้ใช้คนที่ I มีค่ามากกว่า P_t2 แล้วให้เพิ่มเว็บเพจ b เข้าไปในฮิ้นของ
         เว็บเพจ a ของผู้ใช้คนที่ i

ผลลัพธ์ที่ได้ : กลุ่มของฮิ้นที่จะใช้กับผู้ใช้เฉพาะคนๆไป
---------- จบส่วนการทำงานที่สอง ----------

//เวลาส่งฮิ้นกลับไปให้เว็บไคลเอนต์ของผู้ใช้ในกรณีที่ร้องขอเว็บเพจ a
สำหรับกลุ่มของเว็บเพจที่เป็นฮิ้นของเว็บเพจ a ของผู้ใช้คนที่ร้องขอมา (ส่วนการทำงานที่สอง)
    สำหรับเว็บเพจ b ใดๆที่เป็นฮิ้นของเว็บเพจ a (จากส่วนการทำงานที่หนึ่ง)
        ถ้าเว็บเพจ b ไม่อยู่ในฮิ้นของเว็บเพจ a ของผู้ใช้คนที่ร้องขอมา
            เพิ่มเว็บเพจ b เข้าไปเป็นฮิ้นของเว็บเพจ a ของผู้ใช้คนที่ i

ผลลัพธ์สุดท้าย : กลุ่มของฮิ้นที่ได้จากการรวมกันของฮิ้นทั้งส่วน ซึ่งเป็นกลุ่มของฮิ้นเฉพาะของ
ผู้ใช้แต่ละคนไปส่งกลับไปให้เว็บไคลเอนท์ของผู้ใช้
```

5.2 การวัดประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮล

การวัดประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลนั้นจะวัดในแบบเช่นเดียวกับที่ใช้วัดประสิทธิภาพของแพร์ไวส์อัลกอริทึมเดิมดังที่ได้กล่าวแล้วในบทที่ 4 ก็คือจะใช้หลักในการวัดเดียวกัน ซึ่งจะวัดในรูปของอัตรานอมัลไลซ์รีคอล, นอมัลไลซ์พรีซีชั่นและขนาดของฮิ้นโดยเฉลี่ยเช่นเดียวกับในบทที่ 4 แต่จะวัดเป็นฮิ้นของผู้ใช้เป็นคนๆไป เมื่อได้อัตราทั้งสามอย่างจากฮิ้นของแต่ละคนแล้วจึงมาหาค่าเฉลี่ยจากผู้ใช้ทั้งหมดอีกครั้งหนึ่ง ดังสูตรที่ 1, 2, 3

Normalized Recall :

$$\sum_{K=1}^{\text{Number of Client}} \frac{\left(\sum_{j=1}^{n_sendhint \text{ of Client } k} \left(\frac{Hited - number\ of\ hinturl\ a}{n_after\ a_i} \right) \right)}{n_sendhint} \Bigg/ Number\ of\ all\ client$$

Normalized Precision :

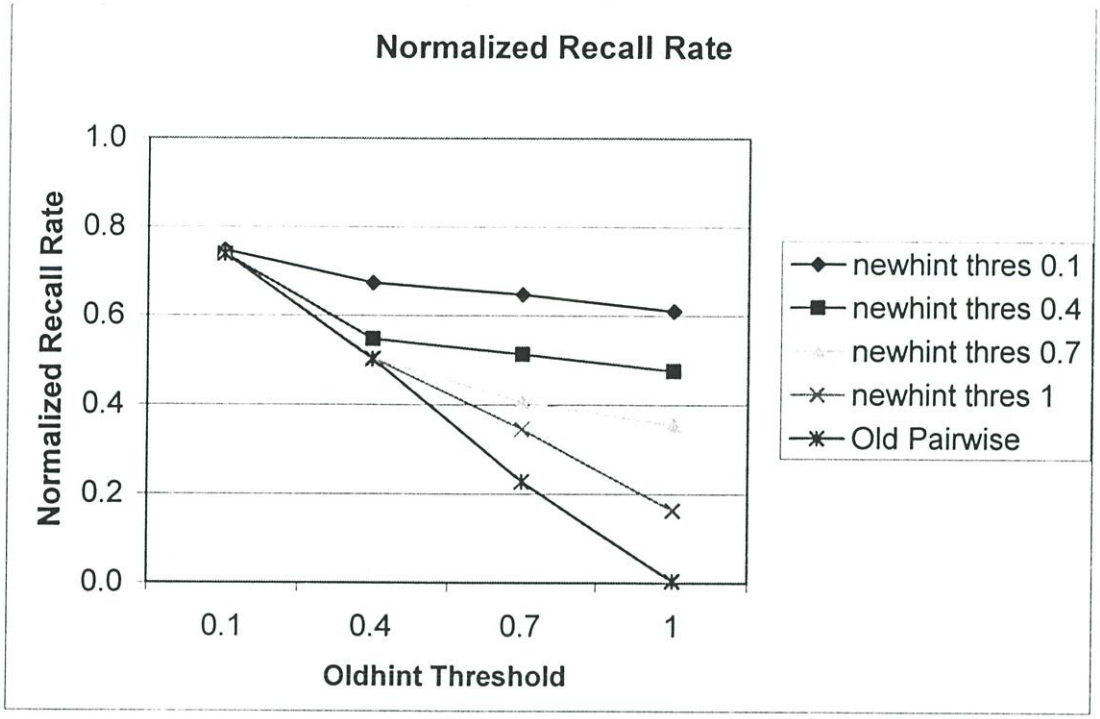
$$\sum_{K=1}^{\text{Number of Client}} \left(\frac{\left(\sum_{j=1}^{n_sendhint \text{ of Client } k} \left(\frac{\sum_{i=1}^{N_hinturl\ of\ a_i \text{ of Client } k} Hited - status\ of\ hinturl\ of\ a_i}{n_hinturl\ of\ a_i \text{ of Client } k} \right) \right)}{n_sendhint} \right) \Bigg/ Number\ of\ All\ Client$$

Average Hintsize :

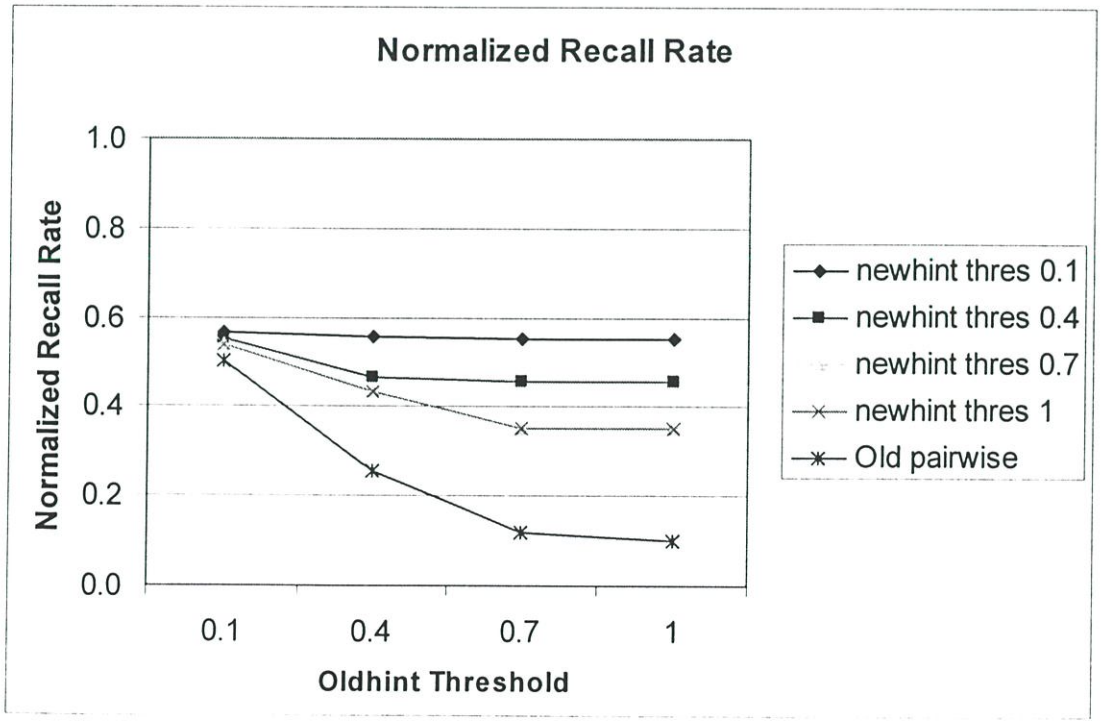
$$\sum_{K=1}^{Number\ of\ Client} \left(\frac{\sum_{i=1}^{n_sendhint\ of\ a_i \text{ of Client } k} n_hinturl\ of\ a_i}{n_sendhint} \right) \Bigg/ Number\ of\ All\ Client$$

5.3 ประสิทธิภาพของแพร์ไวส์อัลกอริทึมที่มีการใช้เทรสโฮล 2 ระดับ

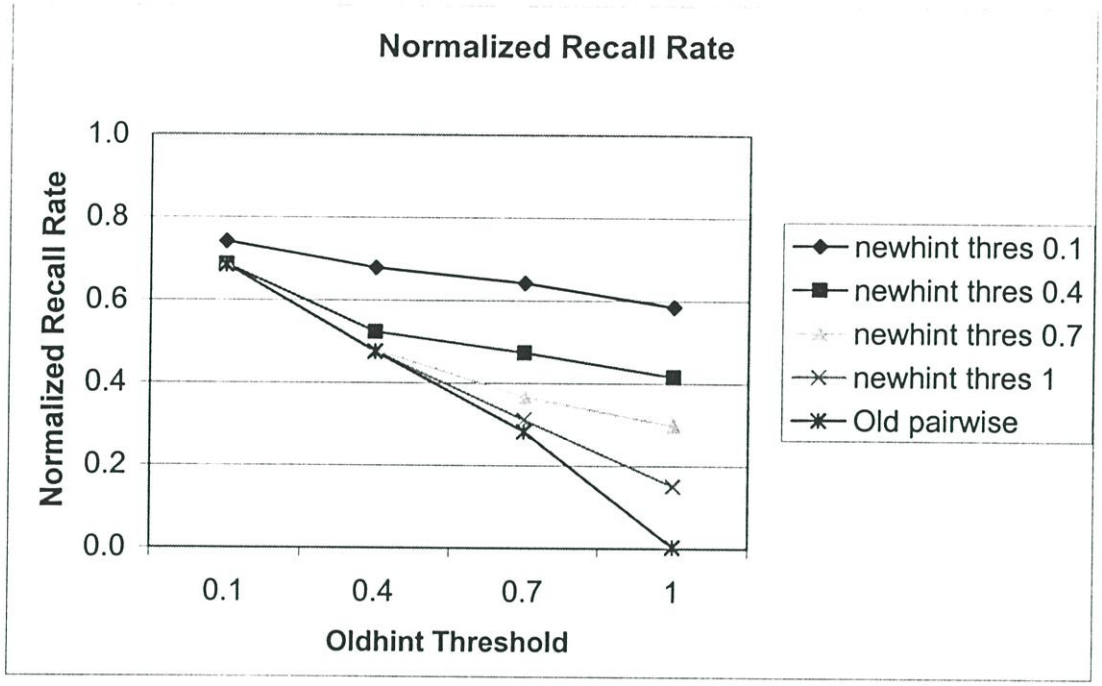
ในการจำลองสถานการณ์จะใช้ข้อมูลชุดเดียวกับที่ใช้ในการจำลองสถานการณ์ของแพร์ไวส์อัลกอริทึมเดิมดังในบทที่ 4 (*ในการจำลองสถานการณ์เพื่อวัดประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับจะจำลองสถานการณ์ในอยู่บนสมมติฐานที่ว่าเครื่องเซิร์ฟเวอร์ที่ใช้ในการสร้างฮิ้นนั้นมีขนาดของเมมโมรีขนาดใหญ่และมีโพรเซสเซอร์ที่มีความเร็วสูง เพราะการทำงานของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับจะต้องใช้เมมโมรีสูง*) ผลการจำลองสถานการณ์การทำงานแพร์ไวส์อัลกอริทึมแบบมีการใช้ เทรสโฮล 2 ระดับของผู้ใช้ 3 อันดับแรกที่มีการเข้าถึงเว็บเซิร์ฟเวอร์มากที่สุดเป็นดังรูปที่ 5.3 – 5.11



รูปที่ 5.3 แสดงอัตรานอมัลไลซ์รีคอลที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 203.107.128.249



รูปที่ 5.4 แสดงอัตรานอมัลไลซ์รีคอลที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.28.22.254

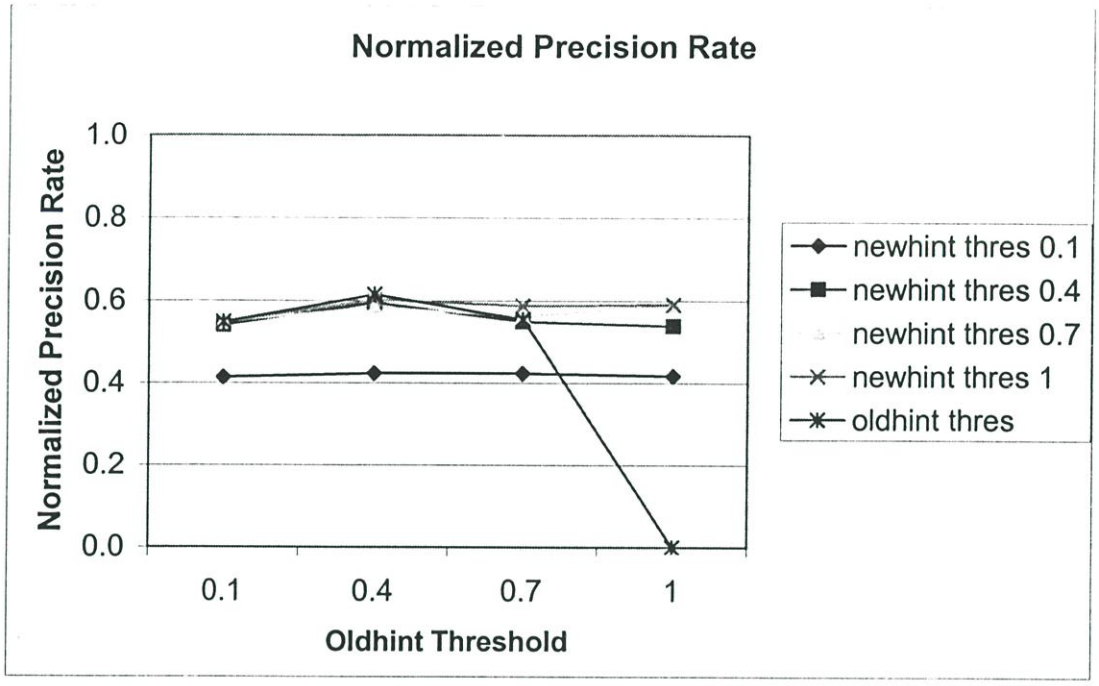


รูปที่ 5.5 แสดงอัตรานอมัลไลซ์รีคอลที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.44.10.80

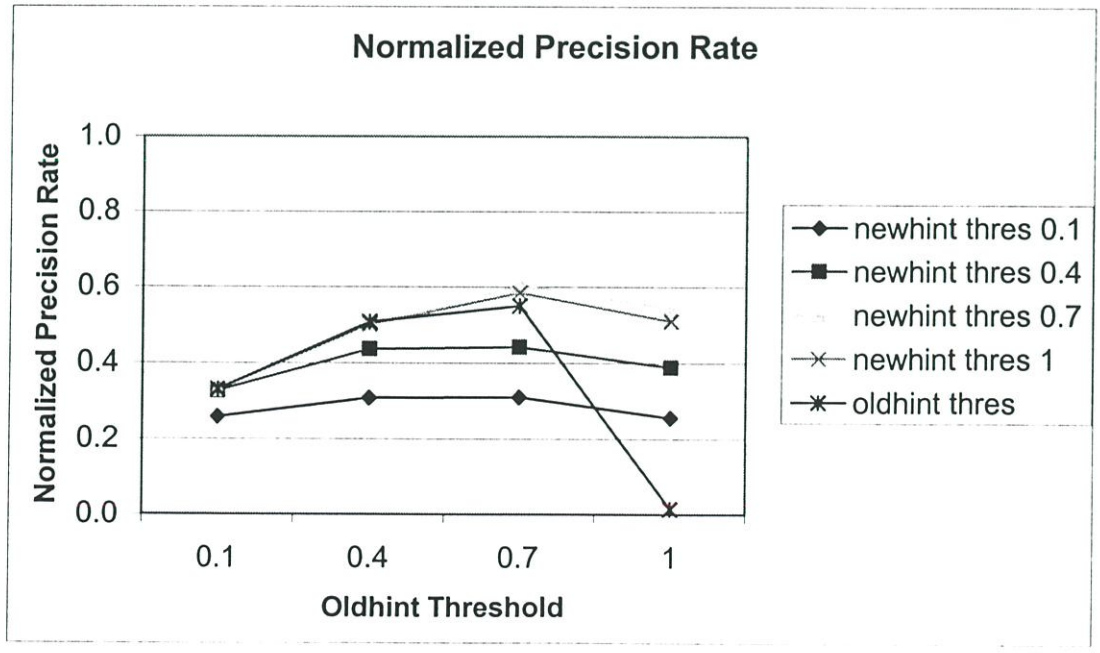
จากรูปที่ 5.3 – 5.5 นั้นจะแสดงอัตรานอมัลไลซ์รีคอลของผู้ใช้ที่มีการร้องขอมาที่เว็บเซิร์ฟเวอร์มากที่สุด 3 อันดับแรก โดยในแกน X จะแทนค่า P_{t1} นั่นก็คือค่าความน่าจะเป็นเทรสโฮลของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลในส่วนที่หนึ่ง (แพร์ไวส์อัลกอริทึมเดิม) ส่วนแต่ละเส้นของกราฟจะแสดงค่า P_{t2} ที่ใช้ นั่นก็คือค่าความน่าจะเป็นเทรสโฮลของแพร์ไวส์อัลกอรึทึมแบบใช้ 2 เทรสโฮล ส่วนการทำงานที่สองที่มีการใช้ข้อมูลเฉพาะแต่ละคนในการสร้างฮิ้น

อัตรานอมัลไลซ์รีคอลของผู้ใช้ทั้ง 3 คนมีแนวโน้มที่คล้ายกันนั่นก็คือจะมีอัตรานอมัลไลซ์รีคอลที่สูงเมื่อมีการใช้ค่า P_{t1} และ P_{t2} ที่ต่ำ แต่ก็จะลดลงตามลำดับเมื่อมีการใช้ค่า P_{t1} หรือ P_{t2} ที่เพิ่มขึ้น เมื่อพิจารณาที่ค่า P_{t2} ค่าหนึ่งและเปลี่ยนค่า P_{t1} ไปหลายค่าจาก 0.1- 1.0 จะเห็นว่าอัตราการลดลงของค่านอมัลไลซ์รีคอลจะน้อยกว่าแพร์ไวส์อัลกอริทึมเดิม (ดูผลลัพธ์จากในบทที่ 4) มากซึ่งก็ทำให้แสดงถึงประสิทธิภาพที่ดีกว่าแพร์ไวส์อัลกอริทึมเดิม

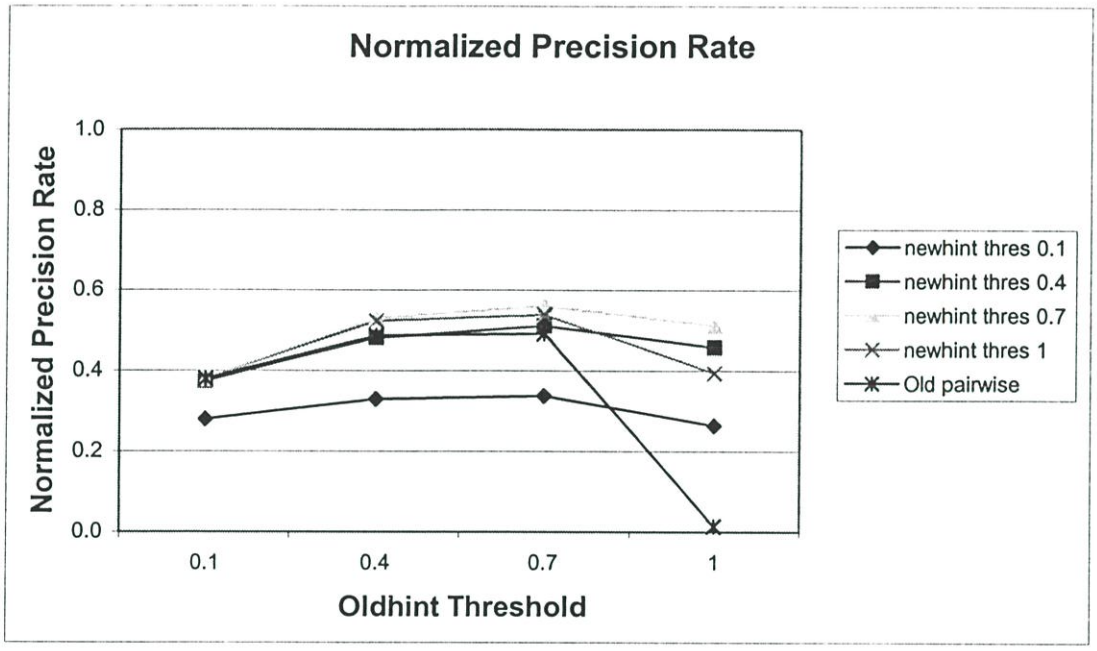
และสำหรับอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับนั้นจะเป็นดังรูปที่ 5.6 – 5.8



รูปที่ 5.6 แสดงอัตรานอมัลไลซ์พรีซิชั่นจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 203.107.128.249



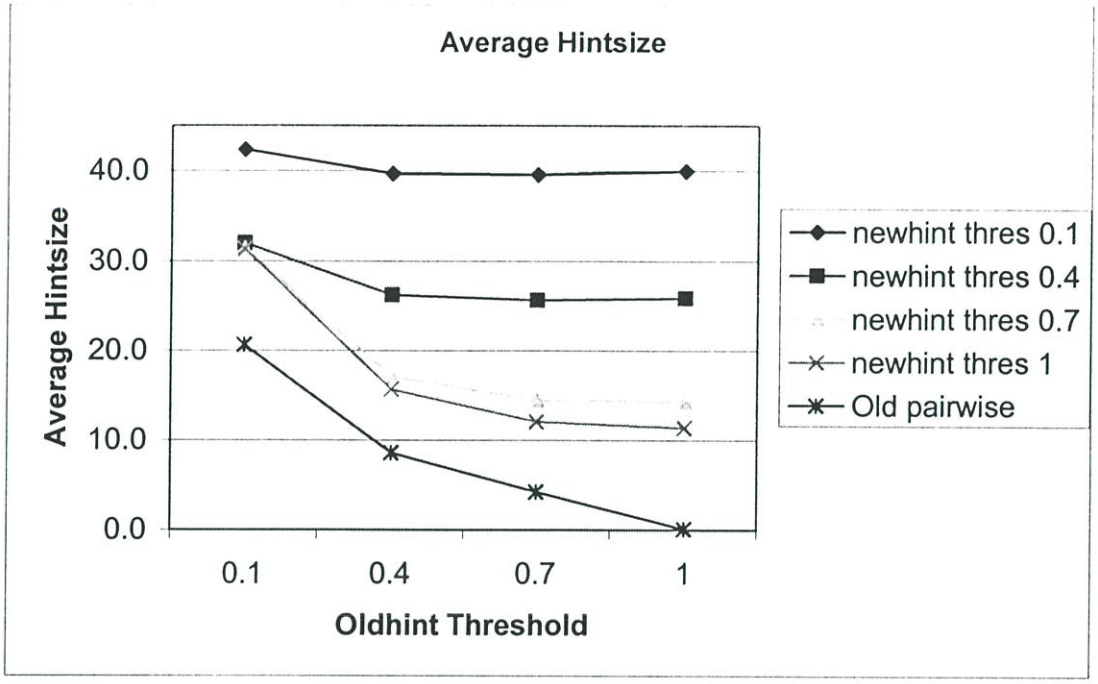
รูปที่ 5.7 แสดงอัตรานอมัลไลซ์พรีซิชั่นจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.28.22.254



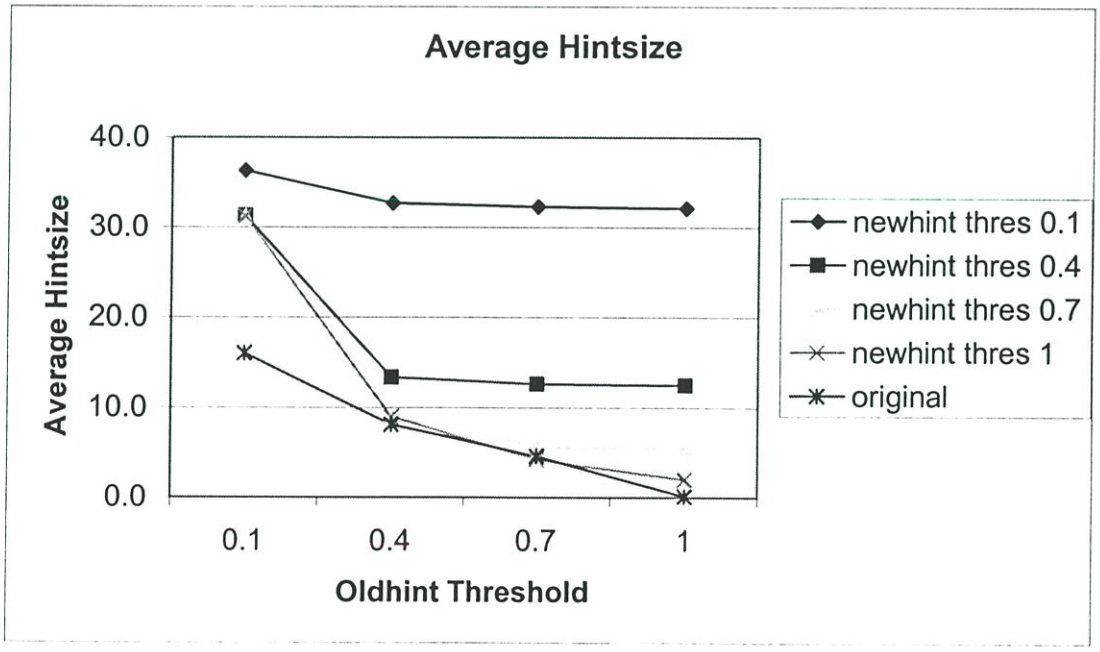
รูปที่ 5.8 แสดงอัตรานอมัลไลซ์พรีซิชั่นที่ได้จากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.44.10.80

สำหรับอัตรานอมัลไลซ์พรีซิชั่นของผู้ใช้ทั้งสามคน จะเห็นได้ว่าอัตรานอมัลไลซ์พรีซิชั่นส่วนใหญ่จะมีค่าที่สูงกว่าแพร์ไวส์อัลกอริทึมเดิม อีกทั้งการเพิ่มขึ้นของอัตรานอมัลไลซ์พริซีชั่นยังเพิ่มขึ้นหรือคงที่และถึงจะมีการตกลงของอัตรานอมัลไลซ์พรีซิชั่นก็มีการลดลงเพียงเล็กน้อยเท่านั้น เมื่อเปรียบเทียบกับแพร์ไวส์อัลกอริทึมเดิมที่มีการลด ลงอย่างรวดเร็ว

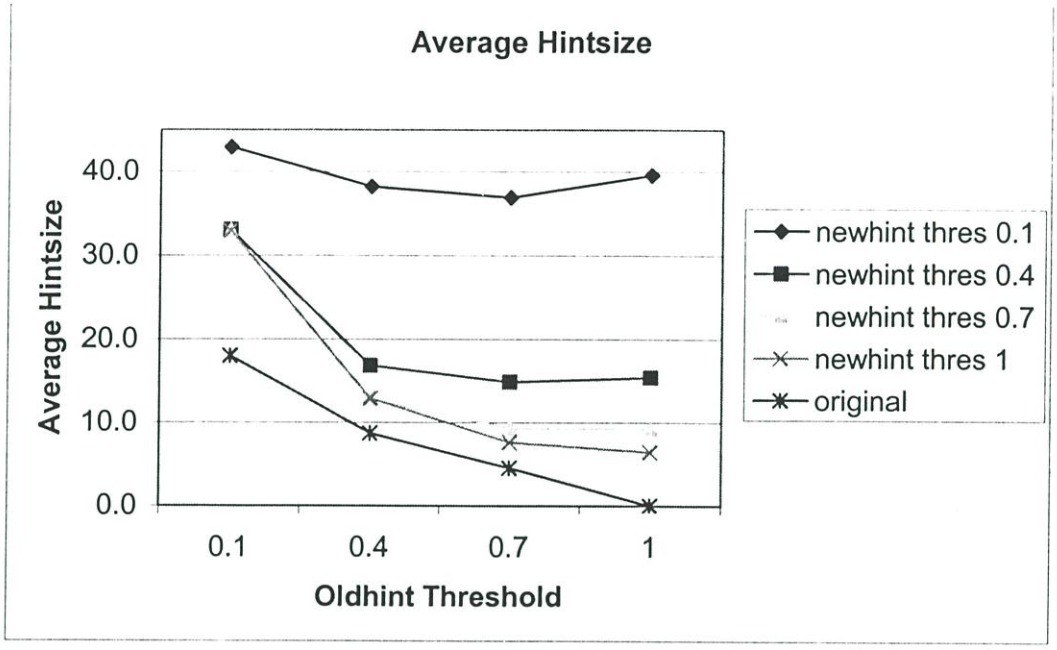
สำหรับขนาดของฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมเดิมเมื่อเทียบกับแพร์ไวส์อัลกอริทึมที่มีการใช้เทรสโฮลแบบ 2 ระดับจะเป็นดังรูปที่ 5.9 – 5.11 ซึ่งจะเห็นได้ว่าขนาดของฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลนั้นค่อนข้างที่จะมีขนาดใหญ่กว่าแพร์ไวส์อัลกอริทึมเดิมเมื่อมีการใช้ P_{t2} ที่มีค่าต่ำ แต่เมื่อมีการใช้ค่า P_{t2} ที่มีค่าสูงๆแล้ว ขนาดของฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลจะมีขนาดใหญ่กว่าแพร์ไวส์อัลกอริทึมเดิมไม่มากเท่านั้นแต่ก็ทำให้ประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลมีมากกว่าแพร์ไวส์อัลกอริทึมเดิม ดังที่ได้กล่าวไปแล้วในส่วนของอัตรานอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พริซิชั่น



รูปที่ 5.9 แสดงขนาดฮิ้นเฉลี่ยจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 203.107.128.249

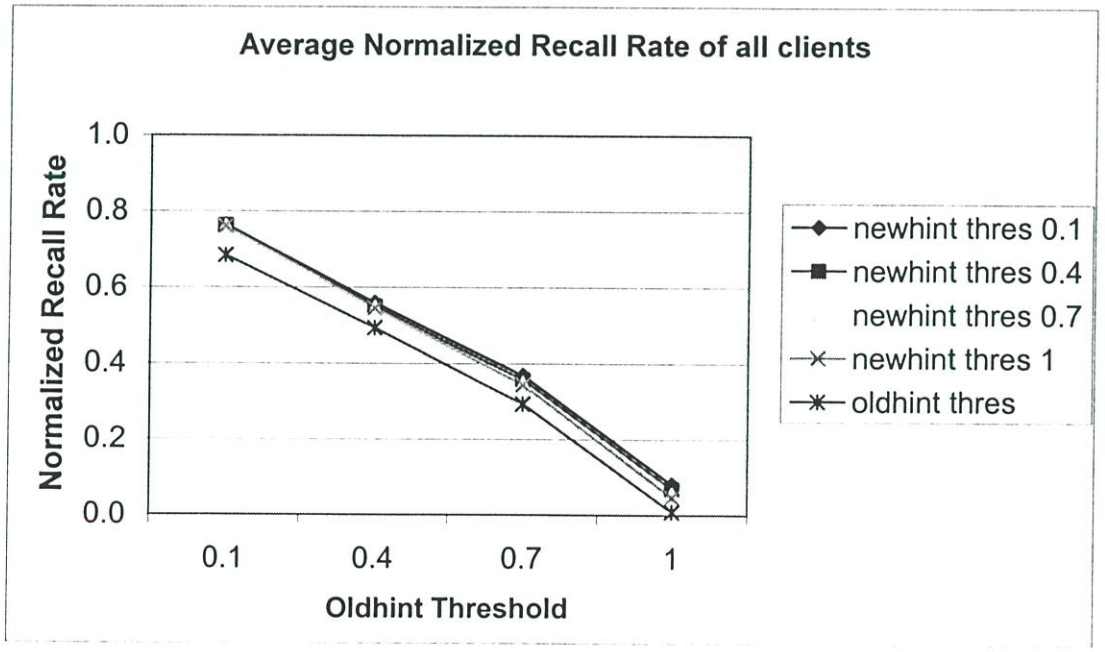


รูปที่ 5.10 แสดงขนาดฮิ้นเฉลี่ยจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.28.22.254

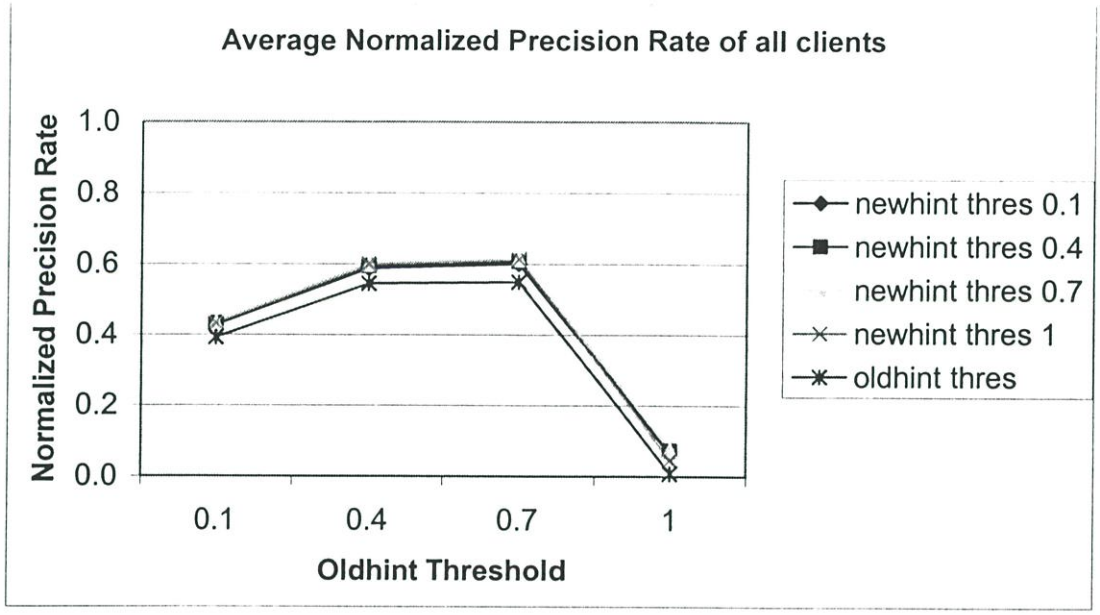


รูปที่ 5.11 แสดงขนาดฮิ้นเฉลี่ยจากแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ 202.44.10.80

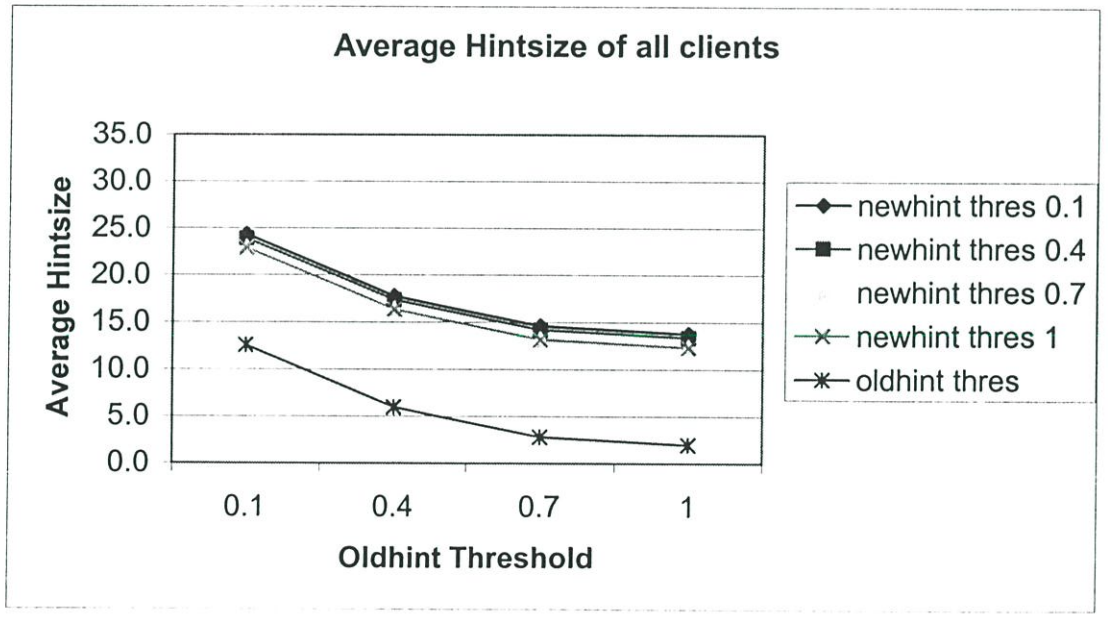
สำหรับเมื่อเปรียบเทียบแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลกับแพร์ไวส์อัลกอริทึมเดิมของผู้ใช้ทั้งหมดจะได้เป็นดังรูป



รูปที่ 5.12 แสดงอัตรานอมัลไลซ์รีคอลเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด



รูปที่ 5.13 แสดงอัตรานอมัลไลซ์พริซิชั่นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด



รูปที่ 5.14 แสดงอัตราขนาดของฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด

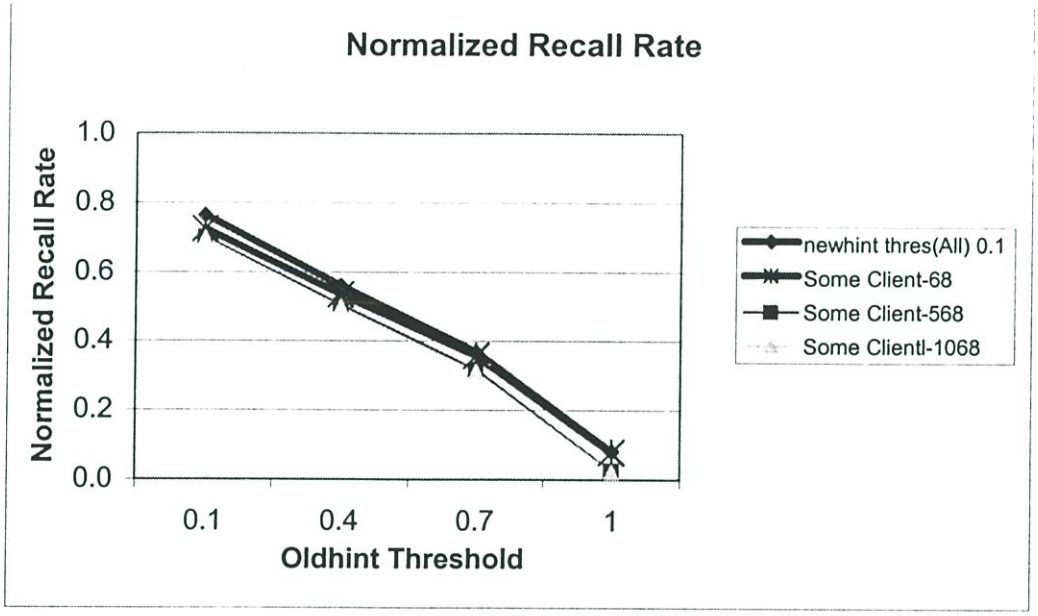
จากรูปที่ 5.12 – 5.14 แสดงผลสรุปของประสิทธิภาพของแพร์ไวส์อัลกอริทึมที่มีการใช้เทรสโฮล 2 ระดับของผู้ใช้ทั้งหมดเมื่อเปรียบเทียบกับแพร์ไวส์อัลกอริทึมเดิม จะเห็นได้ว่าอัตรา

นอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พรีซิชั่น ระหว่างแพร์ไวส์อัลกอริทึมเดิมกับแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับนี้มีค่าที่ไม่แตกต่างกันนัก นั่นก็เพราะว่าข้อมูลชุดนี้เป็นข้อมูลของล็อกไฟล์ (log file) ที่อยู่ด้านเซิร์ฟเวอร์นั่นเอง ทำให้การหารูปแบบการร้องขอที่นำมาสร้างเป็นฮิ้นเฉพาะนั้นไม่ค่อยดีพอ เพราะที่ทางด้านเซิร์ฟเวอร์นั้นไอพีแอดเดรสของไคลเอนต์ที่พร็อกซีของเซิร์ฟเวอร์เห็นอาจจะเป็นไอพีแอดเดรสที่มีการผ่านการแปลงไอพีแอดเดรสจากทางด้านไคลเอนต์แล้ว (Network Address Translation: NAT [11]) ซึ่งทำให้ข้อมูลที่ได้ไม่ได้แสดงถึงรูปแบบการร้องขอของไคลเอนต์จริง แต่ถึงอย่างไรก็ตามอัตรานอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับของไคลเอนต์ที่มีการร้องขอมามากครั้งจริงๆก็ยังคงสูงกว่าแพร์ไวส์อัลกอริทึมเดิมดังเช่นรูปที่ 5.3 – 5.5 และรูปที่ 5.6 – 5.8 เพราะข้อมูลที่ได้นำมาสร้างเป็นฮิ้นได้เอาส่วนของฮิ้นที่มาเฉพาะจากไคลเอนต์นั้นๆ ครอบคลุมการร้องส่วนใหญ่ของไคลเอนต์นั้นๆ ส่วนขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบใช้ 2 ระดับจะมีขนาดใหญ่กว่าแพร์ไวส์อัลกอริทึมเดิมไม่มากเมื่อใช้ค่า P_{t2} สูงๆแต่ก็ได้มาซึ่งประสิทธิภาพที่สูงขึ้น

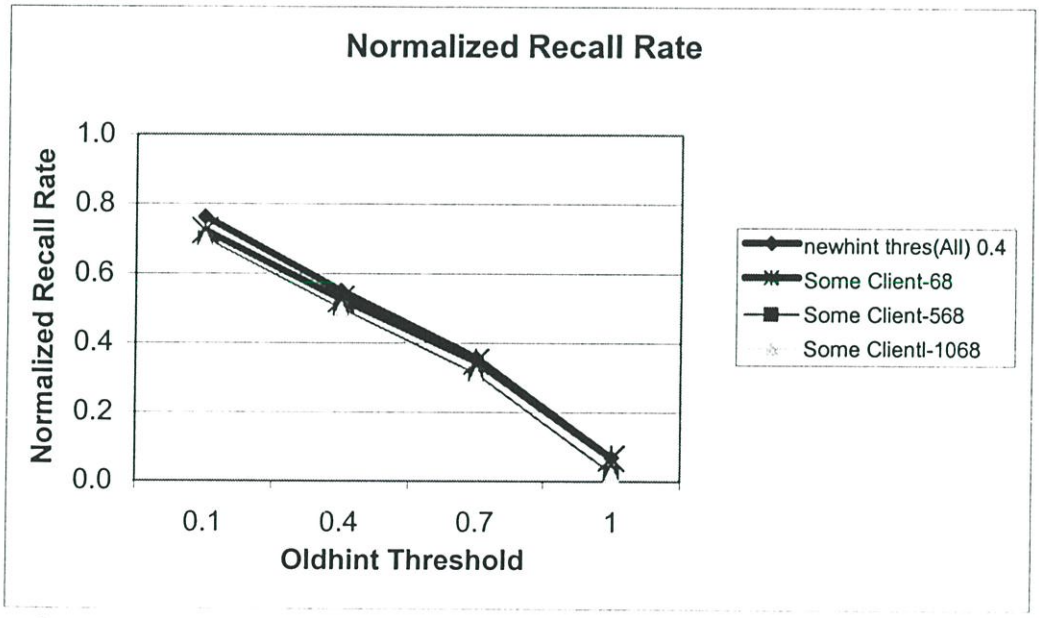
5.4 การปรับปรุงแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

แพร์ไวส์อัลกอริทึมแบบใช้โทรสโอล 2 ระดับนั้นจะสร้างฮิ้นจากข้อมูลสองส่วนคือ ข้อมูลการร้องขอเว็บเพจในอดีตของผู้ทั้งหมด และข้อมูลการร้องขอเว็บในอดีตของผู้ใช้แต่ละคน ซึ่งการทำงานในส่วนที่สองอาจจะได้ฮิ้นที่มีจำนวนมากเพราะนำมาจากข้อมูลของผู้ใช้คนๆเดียวเท่านั้น ซึ่งข้อมูลหรือลำดับการร้องขอของผู้ใช้คนนั้นอาจจะไม่ได้แสดงถึงรูปแบบการร้องขอของผู้ใช้คนนั้นได้ดีพอได้ดีพอ

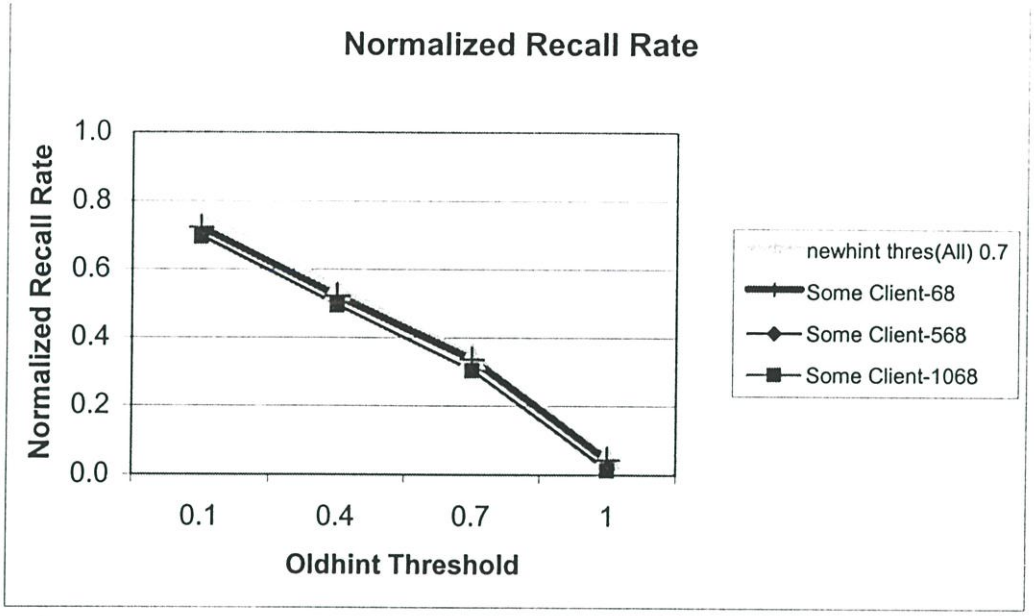
การปรับปรุงประสิทธิภาพแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับคือ จะทำการสร้างฮิ้นในส่วนที่สองเฉพาะข้อมูลการร้องขอของผู้ใช้ที่มีจำนวนครั้งในการร้องขอมามากครั้งพอ เพราะถ้าผู้ใช้มีการร้องขอไม่มากครั้งพอ เมื่อนำมาสร้างเป็นฮิ้นก็จะได้ฮิ้นที่ไม่มีประสิทธิภาพพอ และถึงแม้จะใช้ค่า P_{t2} ที่ใกล้ๆ 1 แล้วก็อาจยังคงได้ฮิ้นที่ไม่มีประสิทธิภาพพอเพราะเหตุผลที่กล่าวข้างต้น แต่ถ้าข้อมูลมาจากการร้องขอเว็บเพจของผู้ใช้ที่มีการร้องขอมามากครั้ง ข้อมูลนั้นก็จะมีประสิทธิภาพพอที่จะนำมาสร้างเป็นฮิ้นได้ ในการจำลองสถานการณ์จะใช้จำนวนครั้งที่มากพอในการนำข้อมูลการร้องขอเว็บเพจของผู้ใช้เฉพาะคนมาสร้างเป็นฮิ้นในส่วนที่สองคือ 68 ครั้ง ซึ่งเป็นค่าเฉลี่ยของจำนวนการร้องขอของผู้ใช้ทั้งหมด และยังได้ลองกับค่าที่มากกว่าค่าเฉลี่ยอีกสองถึงสามค่าเพื่อเปรียบเทียบประสิทธิ ภาพกับด้วย ซึ่งได้ผลเป็นดังนี้



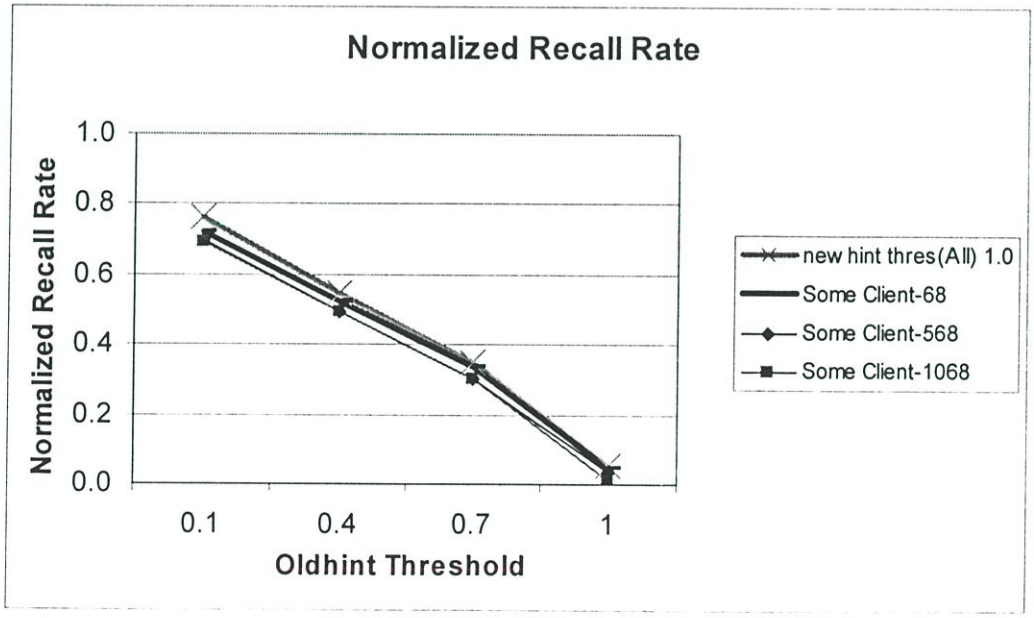
รูปที่ 5.15 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$



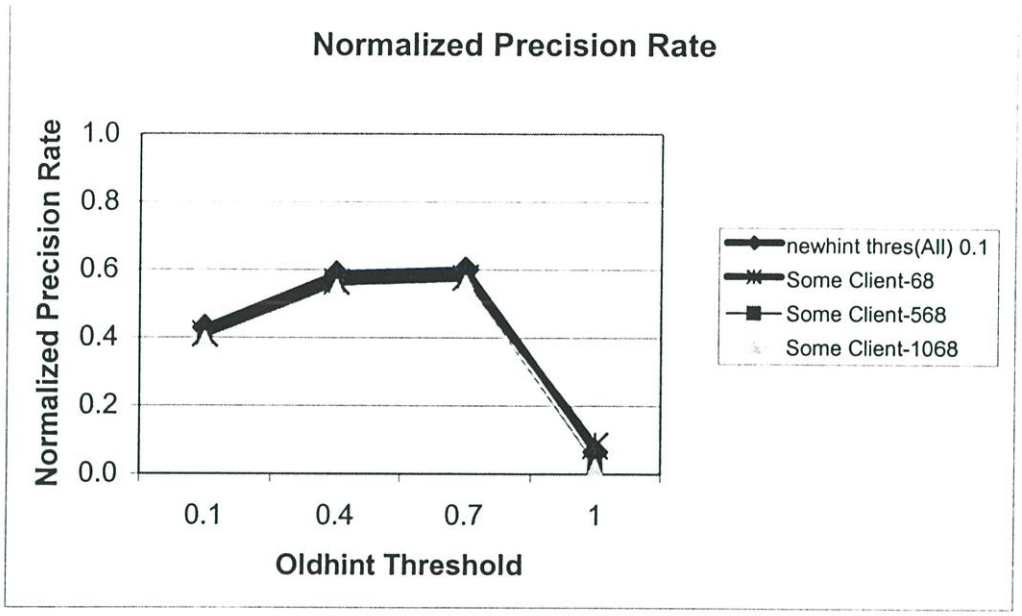
รูปที่ 5.16 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$



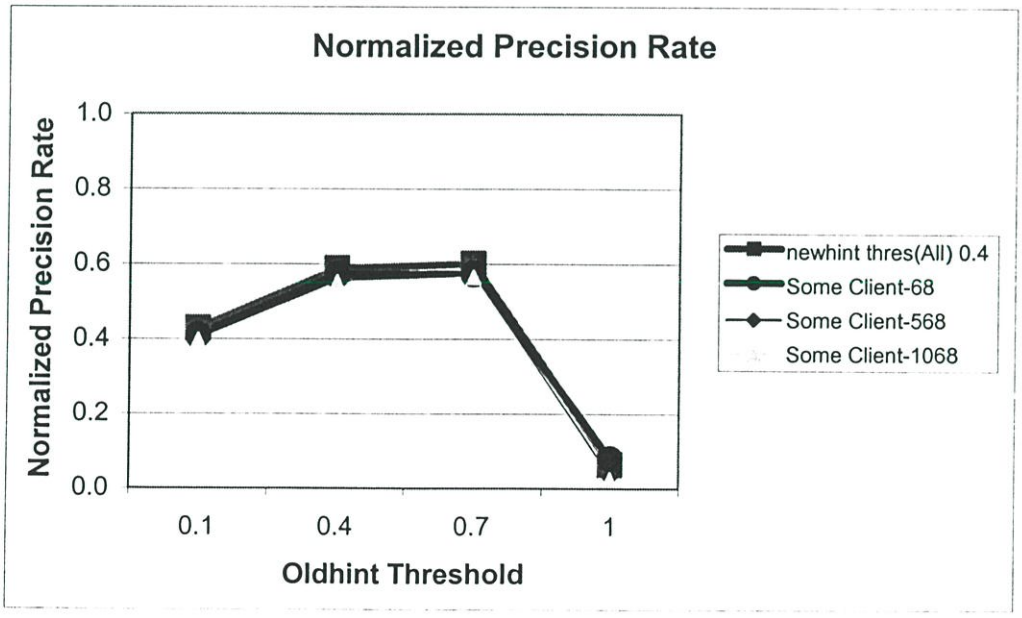
รูปที่ 5.17 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$



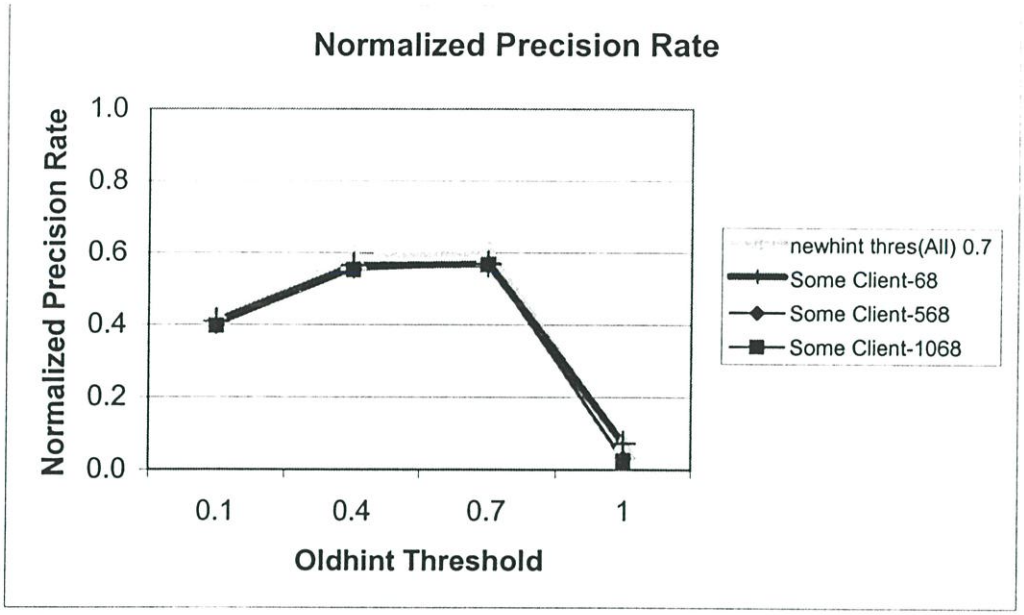
รูปที่ 5.18 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$



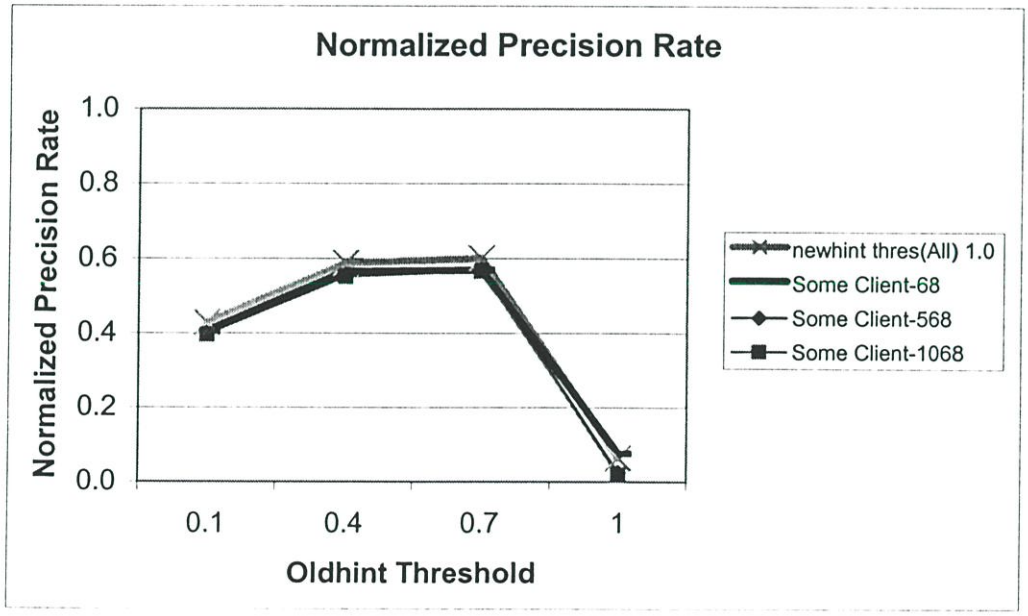
รูปที่ 5.19 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$



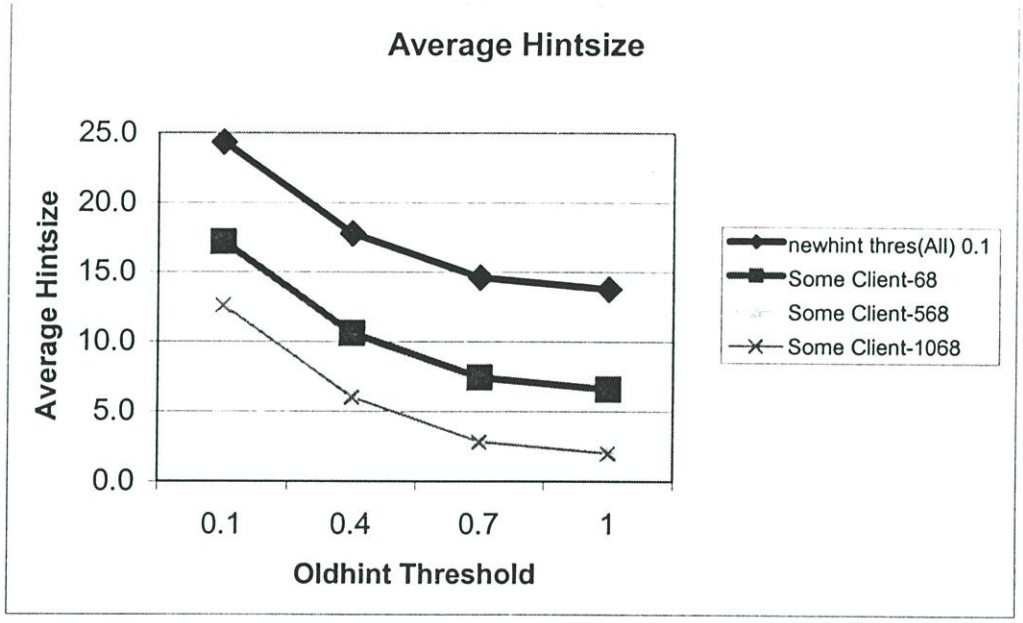
รูปที่ 5.20 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$



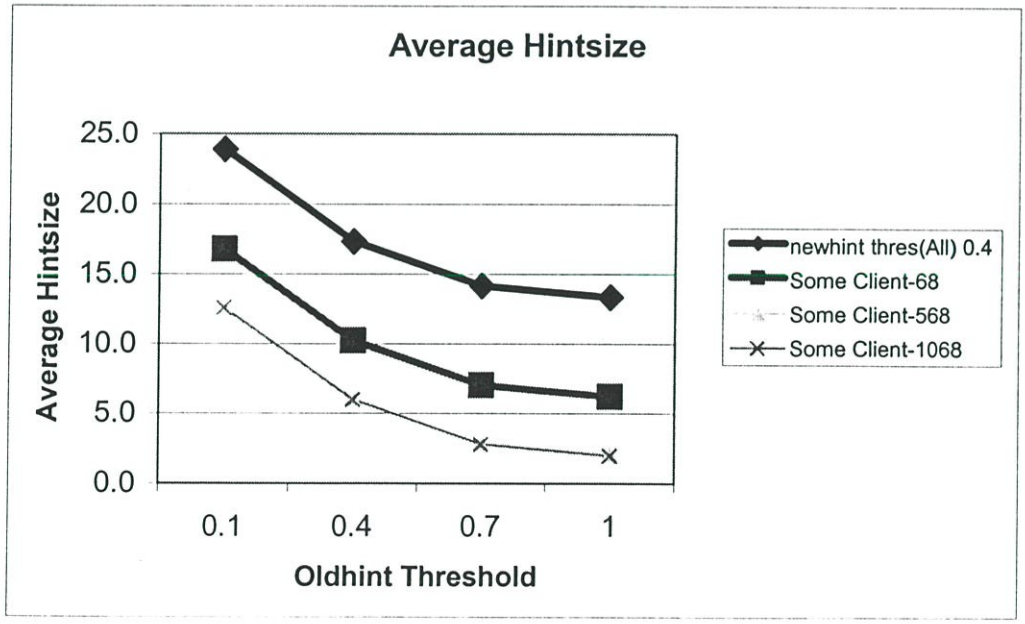
รูปที่ 5.21 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$



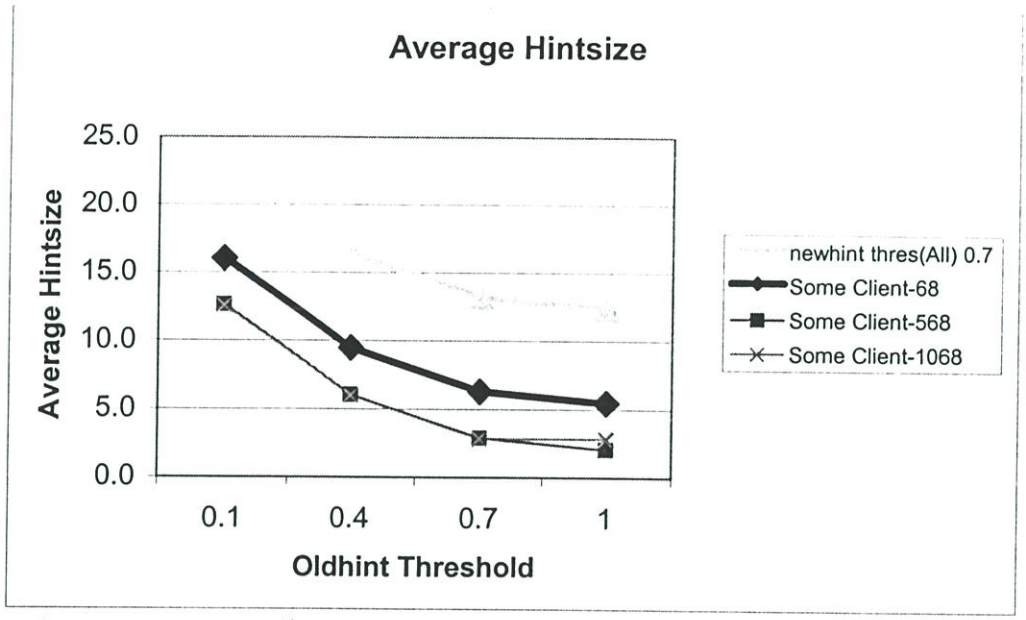
รูปที่ 5.22 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$



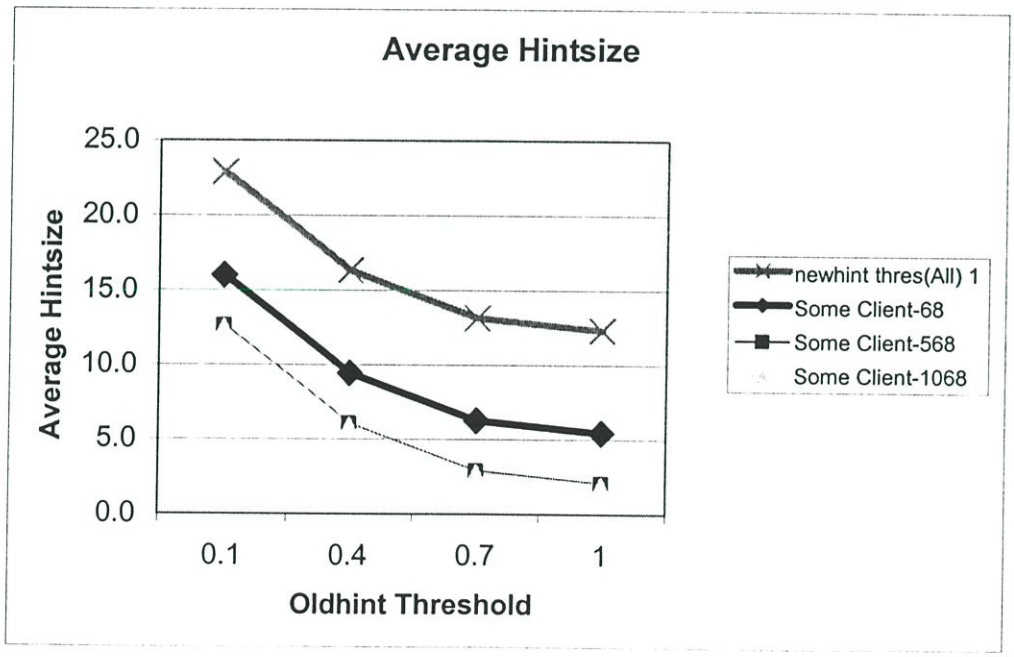
รูปที่ 5.23 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$



รูปที่ 5.24 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$

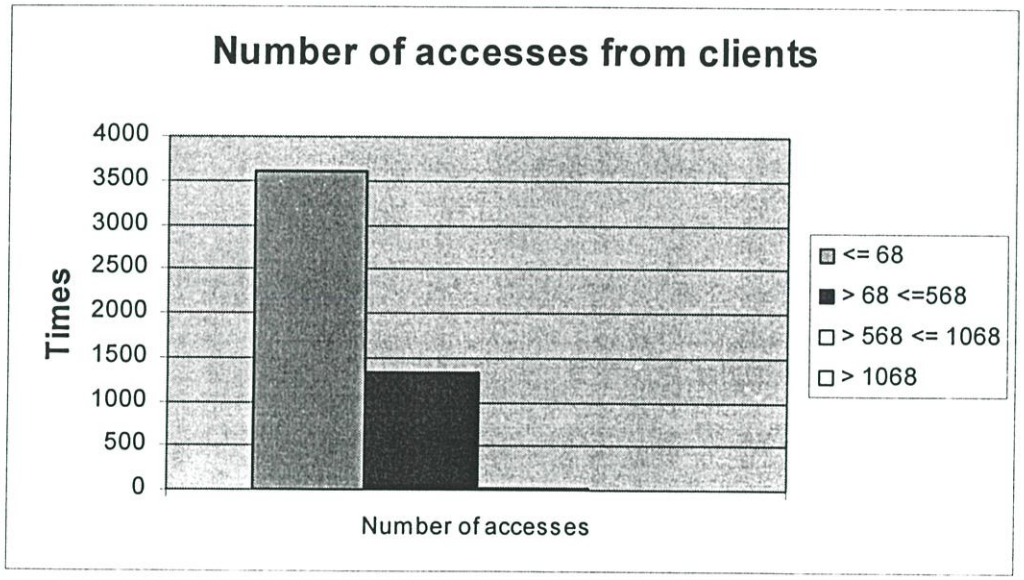


รูปที่ 5.25 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$



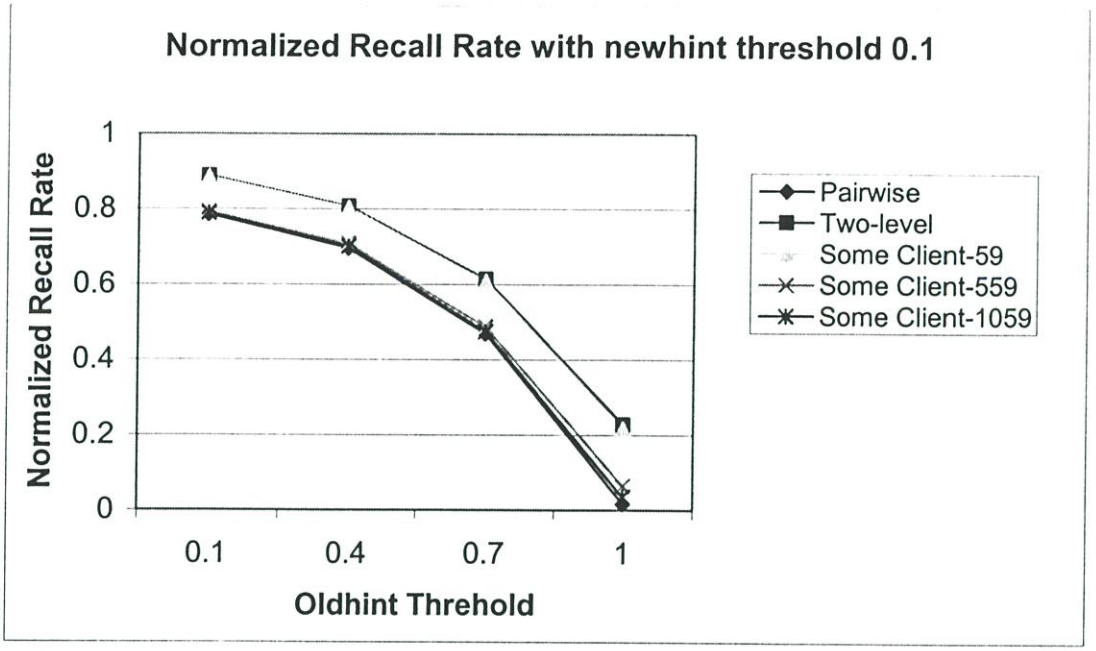
รูปที่ 5.26 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 68 ครั้ง, มากกว่า 568 ครั้ง และมากกว่า 1068 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$

จะเห็นได้ว่าเมื่อได้มีการปรับปรุงประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับด้วยการกำหนดจำนวนครั้งของผู้ใช้ที่มีการร้องขอมากพอ ถึงจะนำข้อมูลการเข้าถึงของผู้ใช้นั้นมาสร้างเป็นฮิ้นเฉพาะของผู้ใช้นั้น (ฮิ้นในส่วนการทำงานที่สอง) ซึ่งทำให้แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่มีการเพิ่มส่วนการทำงานนี้มีประสิทธิภาพยังคงใกล้เคียงกับแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับเดิม แต่ขนาดของฮิ้นเฉลี่ยจะมีขนาดเล็กลง ซึ่งก็ทำให้การใช้ทรัพยากรในการเก็บฮิ้นลดลงเป็นอย่างมาก ซึ่งจากรูปที่ 5.15 – 5.26 จะเห็นว่าเมื่อมีการสร้างฮิ้นเฉพาะในส่วนการทำงานที่สองของไคลเอนต์ที่มีการร้องขอมากกว่า 68 ครั้งจะทำให้ฮิ้นโดยรวมมีขนาดลดลงอย่างมาก แต่เมื่อมีการเพิ่มจำนวนครั้งของไคลเอนต์ที่มีการร้องขอมากพอถึงจะนำมาสร้างเป็นฮิ้นส่วนที่สองขึ้นไม่ได้ทำให้มีผลกับขนาดของฮิ้นหรืออัตราพรีซิชั่นและรีคอลมากนัก ทำให้เห็นว่าจำนวนครั้งของการร้องขอส่วนใหญ่ของไคลเอนต์อยู่ในช่วงไม่เกินค่าเฉลี่ย การกระจายตัวของจำนวนครั้งของการร้องขอของไคลเอนต์เป็นดังรูป 5.27

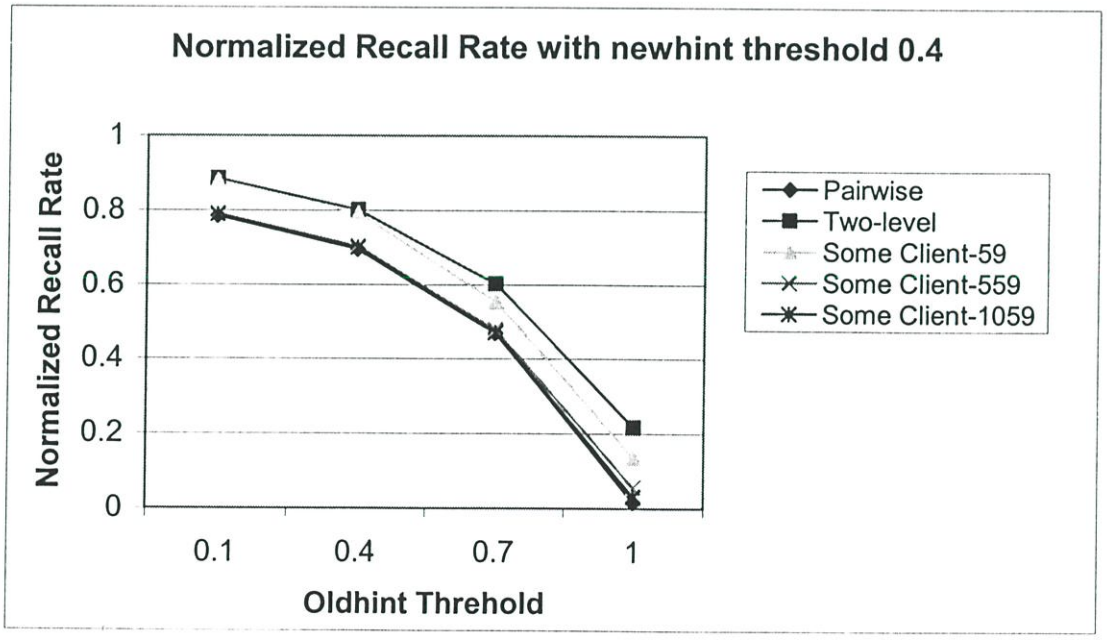


รูปที่ 5.27 แสดงกระจายตัวของจำนวนครั้งของการร้องขอที่มาจากไคลเอนต์ของล็อกไฟล์ฝั่งเซิร์ฟเวอร์

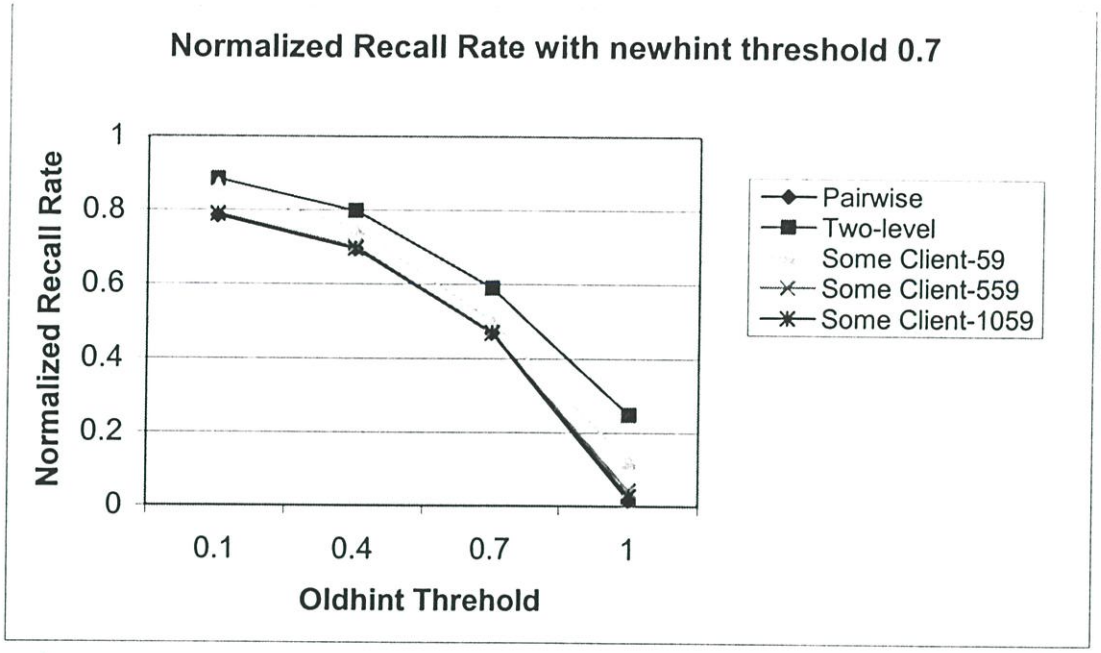
สำหรับการทดลองจำลองสถานการณ์การใช้งานแพร์ไวส์อัลกอริทึมแบบธรรมดา การใช้แพร์ไวสอัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและการใช้แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรังปรุงแล้ว กับข้อมูลการร้องขอของล็อกไฟล์ของพร็อกซี่ทางด้านไคลเอนต์ของบริษัทขนาดใหญ่แห่งหนึ่ง นอกเหนือจากข้อมูลในบทที่ 4 ซึ่งข้อมูลชุดนี้มีค่าเฉลี่ยของจำนวนครั้งของการร้องขอของผู้ใช้ที่ใช้ ซึ่งใช้เป็นเทรสโฮลของการสร้างฮิ้นเฉพาะมีค่าเป็น 59 ครั้งเป็นดังนี้



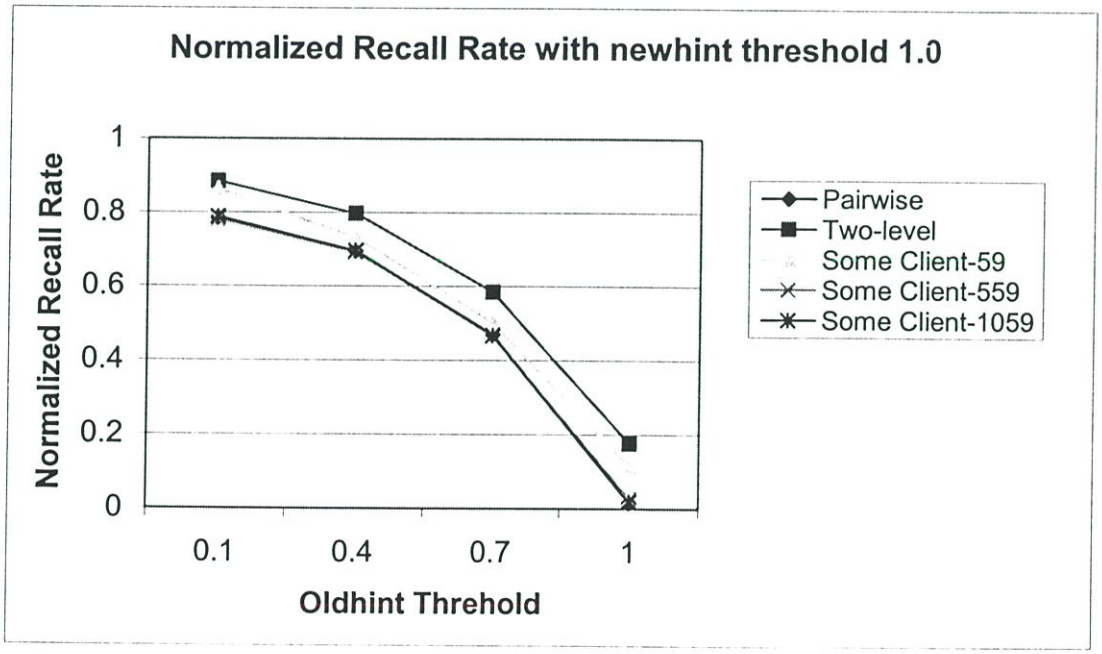
รูปที่ 5.28 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$



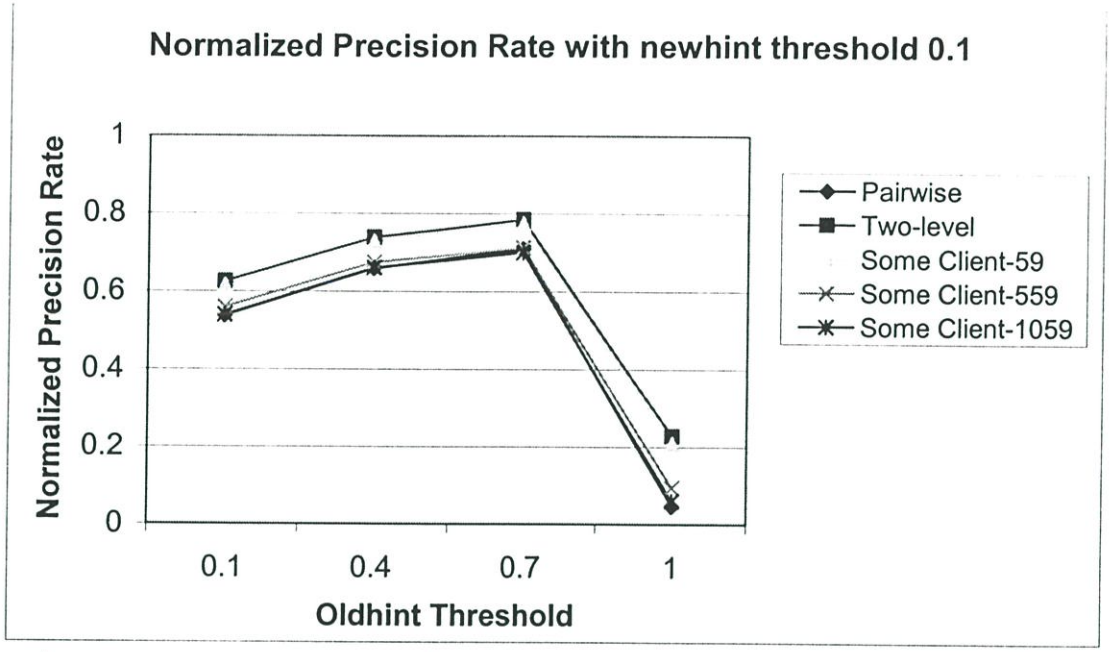
รูปที่ 5.29 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$



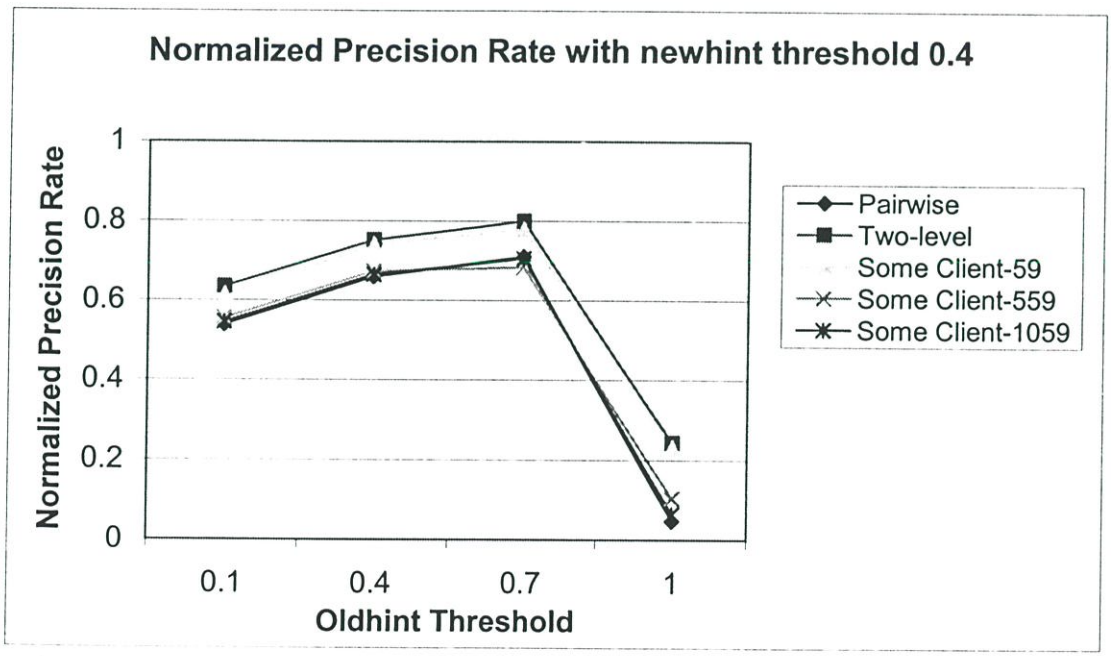
รูปที่ 5.30 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$



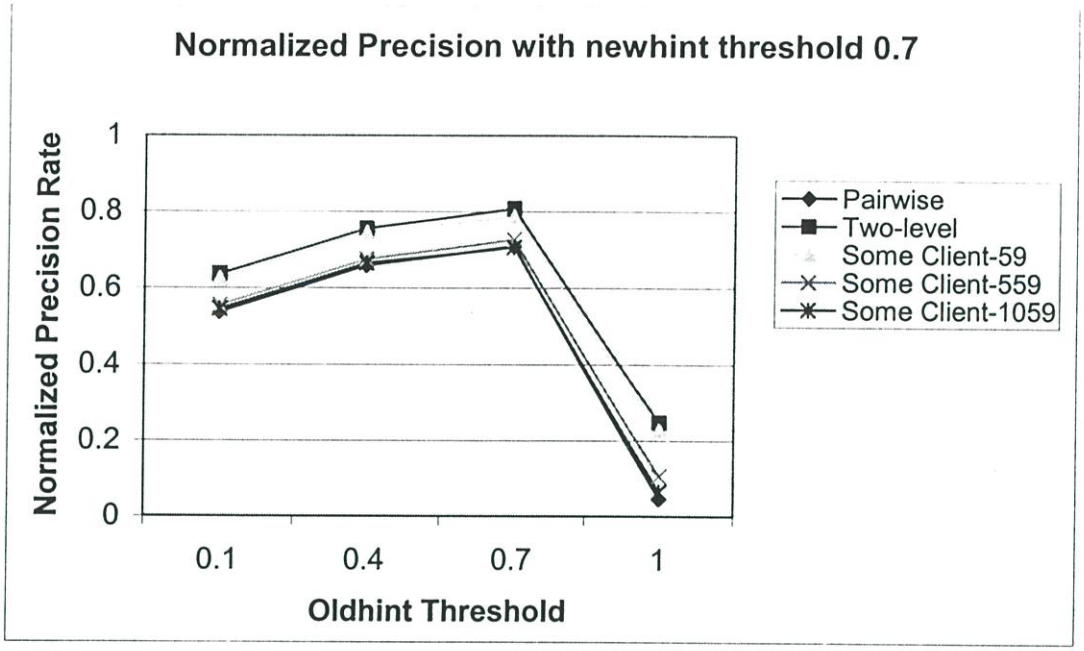
รูปที่ 5.31 แสดงอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$



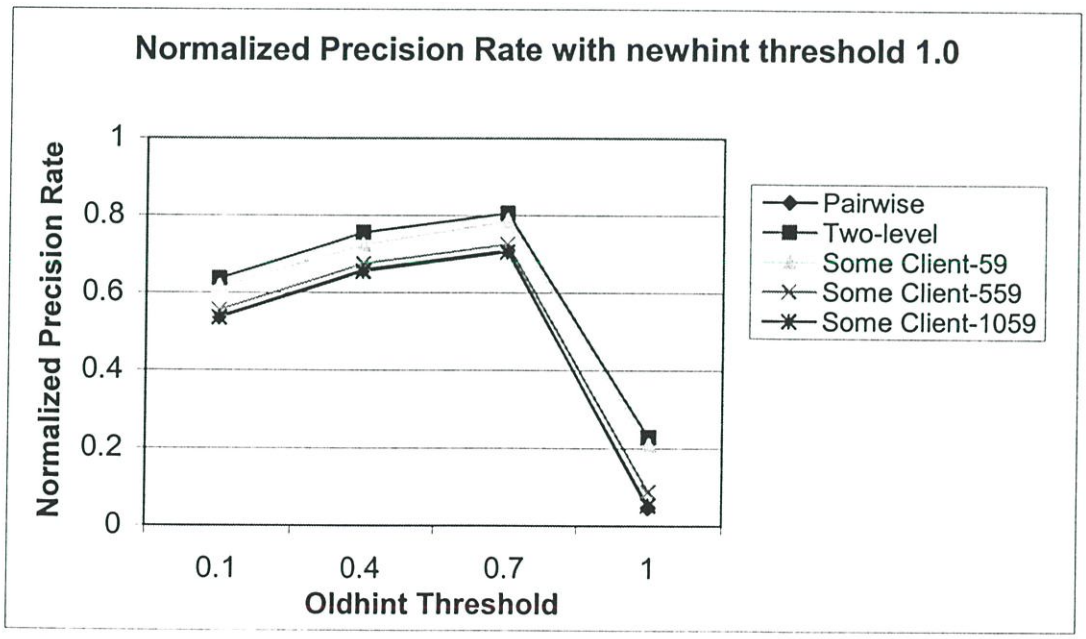
รูปที่ 5.32 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$



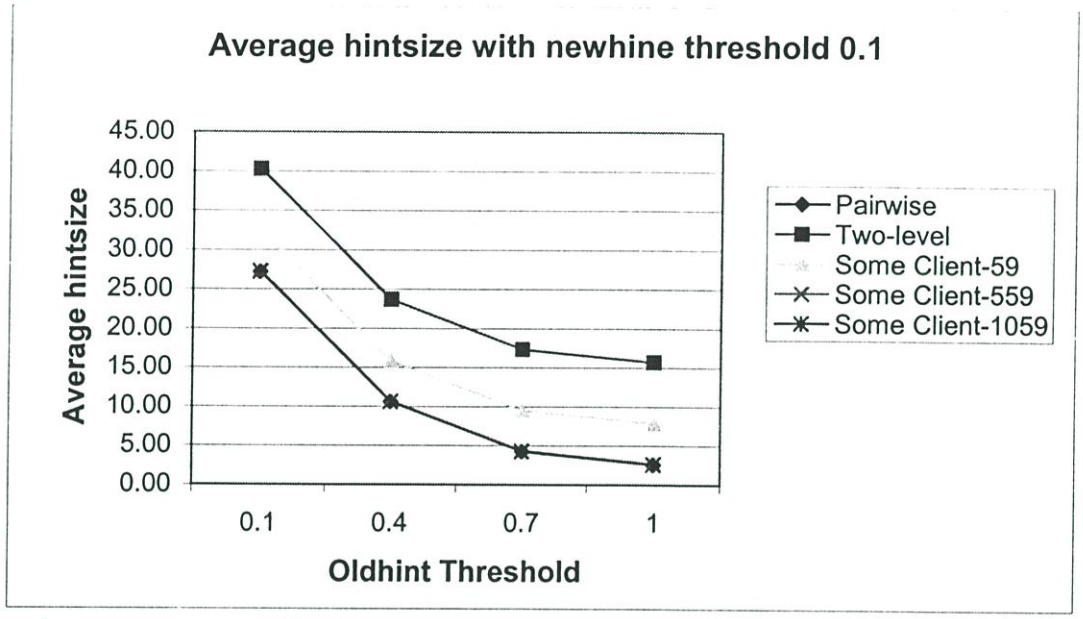
รูปที่ 5.33 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$



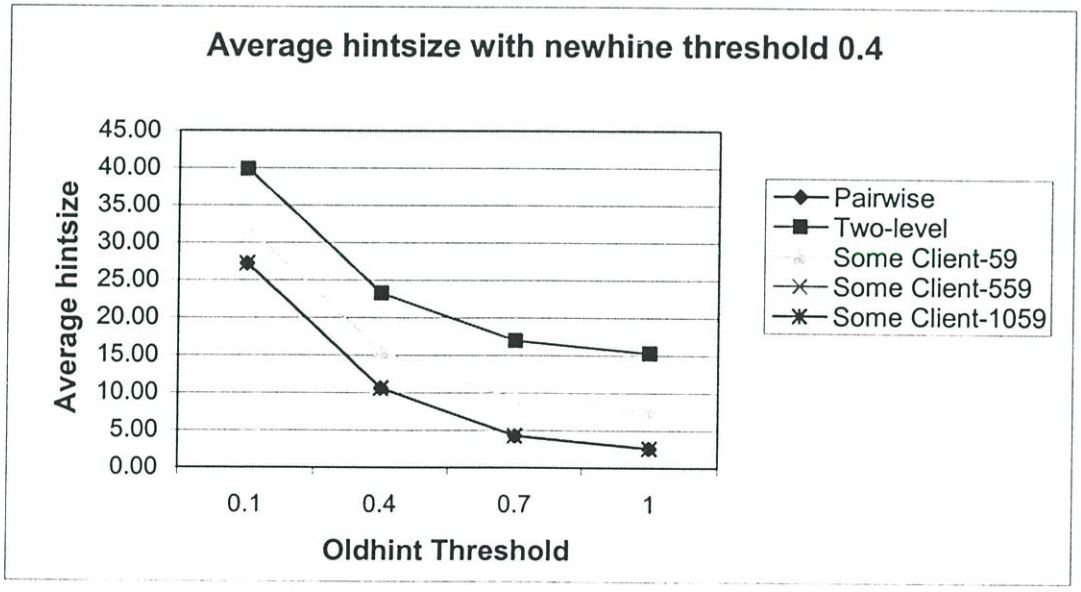
รูปที่ 5.34 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$



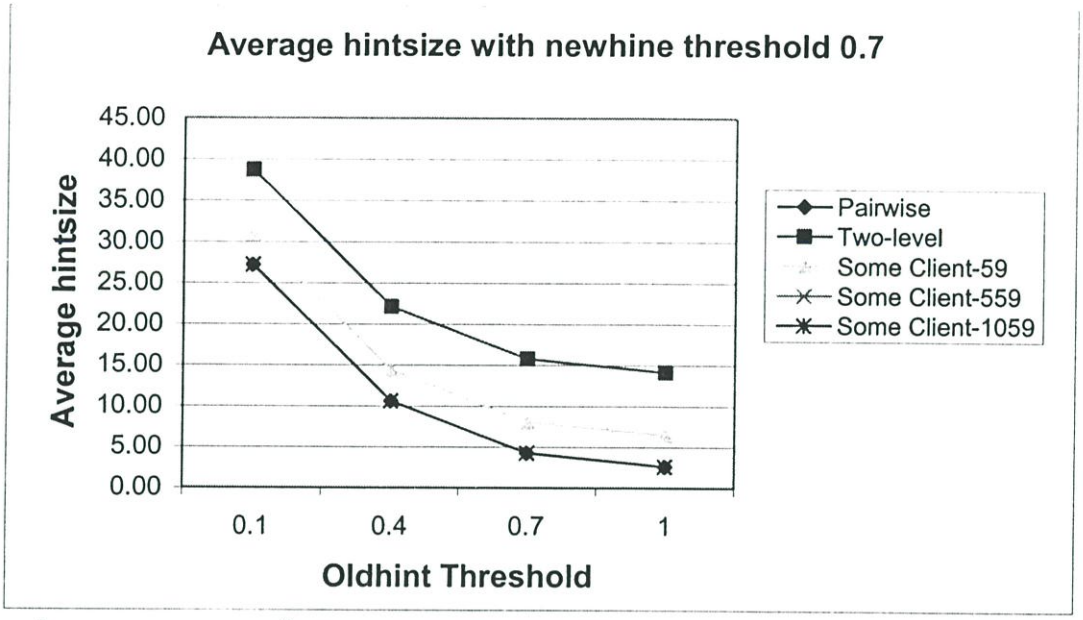
รูปที่ 5.35 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมด เปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$



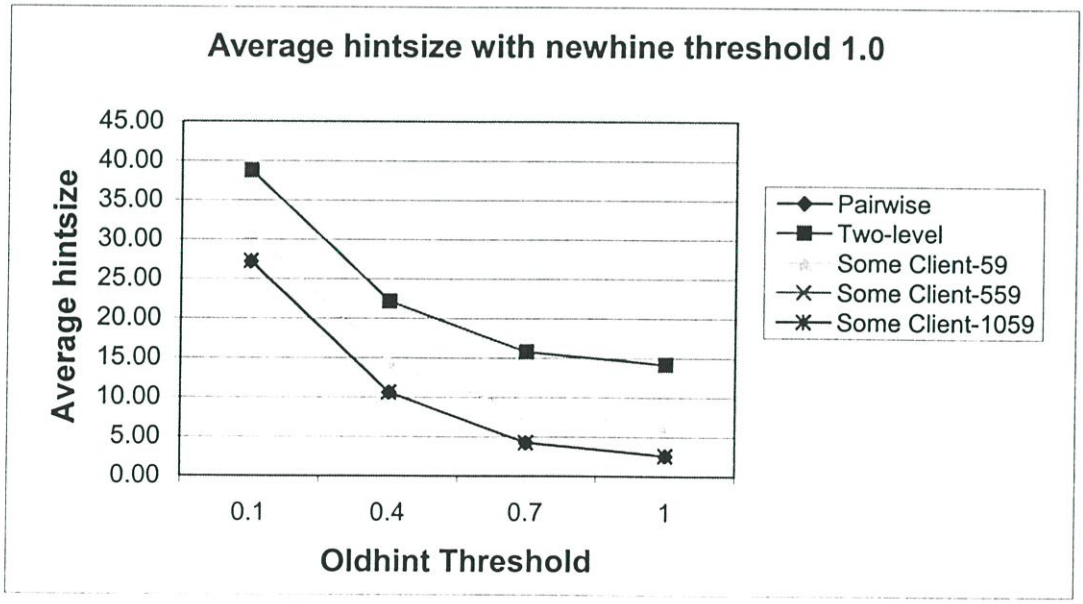
รูปที่ 5.36 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.1$



รูปที่ 5.37 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.4$



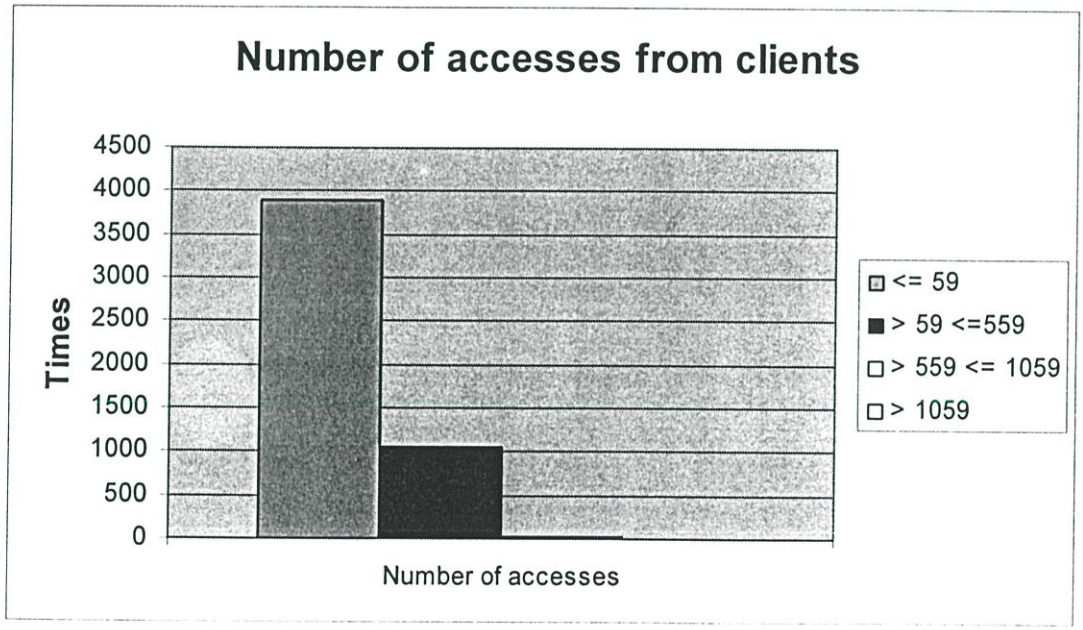
รูปที่ 5.38 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 0.7$



รูปที่ 5.39 แสดงขนาดฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบ 2 เทรสโฮลของผู้ใช้ทั้งหมดเปรียบเทียบระหว่างการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้แต่ละคนทั้งหมดกับการทำงานส่วนที่สองที่ใช้ข้อมูลของผู้ใช้ที่มีการร้องมามากกว่า 59 ครั้ง, มากกว่า 559 ครั้ง และมากกว่า 1059 ครั้ง เมื่อใช้ค่า $P_{t2} = 1.0$

จากรูปที่ 5.28 – 5.31 แสดงให้เห็นประสิทธิภาพที่เพิ่มขึ้นของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับเมื่อเปรียบเทียบกับแพร์ไวส์อัลกอริทึมเดิม อัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับมีค่าที่สูงกว่าแพร์ไวส์อัลกอริทึมเดิม เมื่อเปรียบเทียบกันเมื่อใช้ค่า P_{t2} ค่าหนึ่งค่าใด อัตราแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับจะมีค่าสูงกว่าแพร์ไวส์อัลกอริทึมเดิมในทุกๆค่า สำหรับรูปที่ 5.32 – 5.35 แสดงอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมเดิมเมื่อเปรียบเทียบกับแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับ จะเห็นได้ว่าอัตรานอมัลไลซ์พรีซิชั่นของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับจะมีค่าที่ดีกว่าแพร์ไวส์อัลกอริทึมเดิมเช่นกัน

สำหรับรูปที่ 5.36 – 5.39 แสดงอัตราขนาดของฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับกับแพร์ไวส์อัลกอริทึมเดิม ซึ่งแนวโน้มก็เป็นดังเช่นผลการจำลองสถานการณ์ที่ผ่านมา คือขนาดของฮิ้นเฉลี่ยของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับยังมีขนาดที่ใหญ่กว่าแพร์ไวส์อัลกอริทึมเดิมพอสมควร



รูปที่ 5.40 แสดงกระจายตัวของจำนวนครั้งของการร้องขอที่มาจากไคลเอนต์ของล็อกไฟล์ฝั่งไคลเอนต์

สำหรับการกระจายตัวของจำนวนการร้องขอของไคลเอนต์ของข้อมูลชุดนี้เป็นดังรูปที่ 5.40 และจากรูปที่ 5.28 – 5.31 ก็ได้แสดงถึงอัตรารีคอลของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับที่ได้รับการปรับปรุงแล้วคือจะนำข้อมูลการร้องขอของไคลเอนต์ที่มีการร้องขอมากครั้งพอถึง

จะนำมาสร้างเป็นฮิ้นในส่วนที่สอง ซึ่งค่าที่น่าจะเหมาะสมที่สุดก็คือจำนวนครั้งเฉลี่ยของไคลเอนต์ทั้งหมดนั่นเอง เพราะทำให้อัตรานอมัลไลซ์รีคอลยังคงอยู่ในระดับสูงและลดลงจากแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับมาตรฐานเล็กน้อย

เช่นกันจากรูปที่ 5.32 – 5.35 ก็แสดงถึงอัตรานอมัลไลซ์พรีซิชั่นที่ลดลงเล็กน้อยเมื่อได้มีการปรับปรุงแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับ ซึ่งผลที่ได้ก็คล้ายกับอัตรานอมัลไลซ์รีคอลที่ได้กล่าวไปแล้ว ส่วนขนาดของฮิ้นที่ได้ลดลงอย่างมากเมื่อได้มีการปรับปรุงแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮส 2 ระดับแล้ว ดังรูปที่ 5.36 – 5.39 ทำให้สูญเสียทรัพยากรของระบบในการเก็บข้อมูลฮิ้นค่อนข้างมาก

จากผลการทดลองข้างต้นก็ทำให้เห็นความแตกต่างของประสิทธิภาพของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับเมื่อมีการใช้ข้อมูลล็อกไฟล์ระหว่างล็อกไฟล์จากพร็อกซี่ทางด้านเซิร์ฟเวอร์และล็อกไฟล์จากพร็อกซี่ทางด้านไคลเอนต์ ข้อมูลล็อกไฟล์จากทางด้านไคลเอนต์จะให้ผลการจำลองสถานการณ์ของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ดีกว่าเพราะข้อมูลที่นำมาจะมีความถูกต้องมากกว่า ดังผลจากรูปที่ 5.28 – 5.39

จากข้อมูลทางด้านเซิร์ฟเวอร์โดยสรุปจะเห็นว่าอัตรานอมัลไลซ์พรีซิชั่นและอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงจะเพิ่มขึ้นจากแพร์ไวส์อัลกอริทึมเดิมอยู่ในช่วง 7-10 % ทั้งสองวิธีในขณะที่ขนาดของฮิ้นจะเพิ่มขึ้นประมาณ 2-2.5 เท่าและ 1 เท่าตามลำดับ

ส่วนข้อมูลที่ได้จากทางด้านไคลเอนต์จะเห็นว่าอัตรานอมัลไลซ์พรีซิชั่นและอัตรานอมัลไลซ์รีคอลของแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงจะเพิ่มขึ้นจากแพร์ไวส์อัลกอริทึมเดิมอยู่ในช่วง 10-15 % และ 8-15 % ตามลำดับ ในขณะที่ขนาดของฮิ้นจะเพิ่มขึ้นประมาณ 1.5-2 เท่าและ 0.5-1 เท่าตามลำดับ

บทที่ 6
บทสรุปและวิเคราะห์ผลการทดลอง

6.1 สรุปและวิเคราะห์ผลการทดลอง

จากผลการจำลองสถานการณ์ในบทที่ 4 จะเห็นได้ว่าแพร์ไวส์อัลกอริทึมนั้นเป็นอัลกอริทึมที่มีประสิทธิภาพในการทำนายเว็บเพจหรือรีซอร์สใดๆ ที่จะถูกเข้าถึงในช่วงระยะเวลาหนึ่งหลังจากที่ได้มีการร้องขอเว็บเพจหรือรีซอร์สหนึ่งๆ ได้ดีพอสมควร โดยแพร์ไวส์อัลกอริทึมจะใช้วิธีการทางสถิติมาประยุกต์ใช้หาความน่าจะเป็นของการขึ้นต่อกันของเว็บเพจใดๆ จากนั้นก็จะนำกลุ่มของเว็บเพจใดๆที่ขึ้นกับเว็บเพจหนึ่งมาสร้างเป็นฮิ้น โดยฮิ้นจะเป็นกลุ่มของเว็บเพจที่น่าจะถูกร้องขอหลังจากที่ได้มีการร้องขอเว็บเพจที่เว็บเพจหลักของฮิ้นกลุ่มนั้นแล้วการทำงานของแพร์ไวส์อัลกอริทึมจะใช้ข้อมูลในอดีตในการประมวลผล เพื่อใช้ในการสร้างเป็นฮิ้นของเว็บเพจต่างประสิทธิภาพของแพร์ไวส์อัลกอริทึมนั้น ค่าประสิทธิภาพที่ได้จะขึ้นกับคุณภาพของฮิ้นที่ถูกสร้างขึ้นมา โดยฮิ้นที่ถูกสร้างขึ้นมาโดยแพร์ไวส์อัลกอริทึมนั้นจะได้เป็นกลุ่มของฮิ้นเดียว (Hints Pool) แล้วจะนำฮิ้นที่ได้นั้นมาใช้กับผู้ใช้งานทุกๆคนที่มีการร้องขอมา และจากการจำลองสถานการณ์ในบทที่ 4 นั้นแสดงให้เห็นว่าฮิ้นที่ได้จากแพร์ไวส์อัลกอริทึมนั้น ค่อนข้างที่จะไม่ได้แสดงพฤติกรรมของผู้ใช้แต่ละคนได้ดีนักเพราะสร้างจากข้อมูลการเข้าถึงเว็บเพจในอดีตของผู้ใช้ทุกๆคน ทำให้ฮิ้นที่ได้เหมาะสมกับผู้ใช้บางคนเท่านั้นดังผลที่ได้ในบทที่ 4

วิธีการแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับซึ่งจะแบ่งข้อมูลของผู้ใช้ออกเป็น 2 ส่วนในการใช้สร้างเป็นฮิ้น ส่วนแรกจะแยกข้อมูลการร้องขอของผู้ใช้แต่ละคนแล้วนำข้อมูลของแต่ละคนนั้นมาสร้างเป็นฮิ้น ซึ่งฮิ้นที่ได้ก็ย่อมแสดงพฤติกรรมของผู้ใช้คนนั้นๆจริงๆ เพราะข้อมูลที่นำมาใช้สร้างเป็นฮิ้นนั้นมาจากเฉพาะผู้ใช้คนนั้นๆเท่านั้น ส่วนที่สองจะทำการสร้างฮิ้นโดยใช้ข้อมูลผู้ใช้ทั้งหมดไม่มีการแยกเป็นผู้ใช้คนไหน ซึ่งการทำงานส่วนนี้ก็เหมือนกับการทำงานของแพร์ไวส์อัลกอริทึมเดิมนั่นเอง ที่ยังคงต้องมีส่วนการทำงานของแพร์ไวส์อัลกอริทึมเดิมอยู่นั้นเพราะต้องการนำการขึ้นต่อกันของเว็บเพจที่มีความน่าจะเป็นสูงๆของผู้ใช้ส่วนใหญ่มาสร้างเป็นฮิ้นด้วยเพื่อเพิ่มประสิทธิภาพของฮิ้นของแพร์ไวส์อัลกอริทึมขึ้น เมื่อได้ฮิ้นจากทั้งสองส่วนการทำงานแล้ว เมื่อมีการร้องขอเว็บเพจใดๆมาที่เว็บเซิร์ฟเวอร์ ทางเซิร์ฟเวอร์ก็จะทำการรวมฮิ้นของเว็บเพจที่ถูกร้องขอจากทั้งสองส่วนเข้าด้วยกันในลักษณะยูเนี่ยน (U) ก็ได้กลุ่มของฮิ้นที่จะใช้ส่งไปยังเว็บไคลเอนต์ของผู้ใช้เพื่อใช้ในการพรีเฟรชชิ่งต่อไป

จากการจำลองสถานการณ์ในบทที่ 5 จะเห็นได้ว่าประสิทธิของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับจะมีอัตรานอมัลไลซ์รีคอลที่สูงกว่าแพร์ไวส์อัลกอริทึมเดิมอีกทั้งยังมีอัตราการ

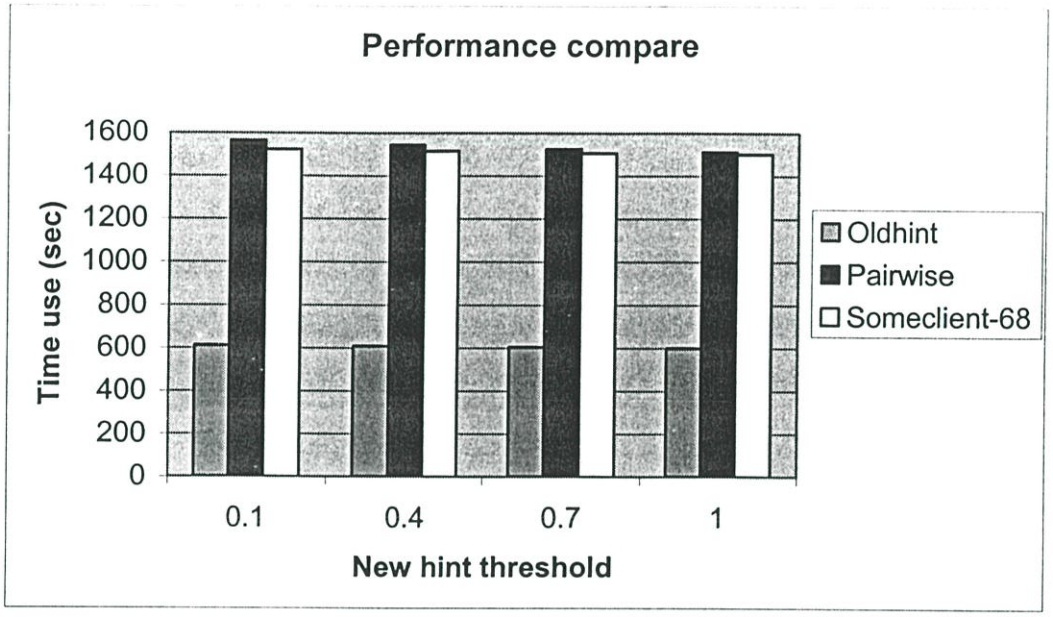
ลดลงของอัตรานอมัลไลซ์รีคอลที่น้อยกว่าแพร์ไวส์อัลกอริทึมเดิม ส่วนนอมัลไลซ์อัตราพรีซิชั่นนั้น แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับนั้นจะมีระดับที่สูงกว่าแพร์ไวส์อัลกอริทึมเดิมเช่นกัน แต่ข้อด้อยของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับนั้นคือฮิ้นที่ได้จะมีขนาดค่อนข้างที่จะใหญ่กว่าแพร์ไวส์อัลกอริทึมเดิมพอสมควร เพราะเกิดจากการนำเอาฮิ้นจากส่วนการทำงานสองส่วน ซึ่งในบทที่ 5 ก็ได้เสนอวิธีการที่จะลดขนาดของฮิ้นของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับนั้นจะทำโดยการเลือกที่จะสร้างฮิ้นของส่วนการทำงานที่หนึ่ง (ฮิ้นที่ได้จากข้อมูลการร้องขอของผู้ใช้เฉพาะคน) สำหรับข้อมูลการร้องขอของผู้ใช้คนใดๆที่มีจำนวนครั้งของการเข้าถึงมากพอเท่านั้น เพราะถ้าผู้ใช้มีการร้องขอมาน้อยครั้งแล้วนำข้อมูลนั้นมาสร้างเป็นฮิ้นเฉพาะสำหรับผู้ใช้คนนั้นย่อมที่จะได้ฮิ้นที่ไม่ดีพอ เพราะฮิ้นนั้นไม่ได้แสดงถึงพฤติกรรมของผู้ใช้ได้ดีพอดังนั้นจึงสามารถลดขนาดของฮิ้นเฉลี่ยได้มากพอสมควร ในขณะที่ยังคงระดับอัตรานอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พรีซิชั่นที่ใกล้เคียงกับแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับมาตราฐาน ผลการจำลองสถานการณ์ของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับกับแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้มีกา รปรับปรุงเพื่อลดขนาดของฮิ้นนั้นจะเห็นว่าอัตรานอมัลไลซ์รีคอลและอัตรานอมัลไลซ์พรีซิชั่นลดลงเพียงเล็กน้อยเท่านั้นแต่สามารถที่จะลดขนาดของฮิ้นได้มากเลยทีเดียว

แต่สิ่งสำคัญอีกอย่างหนึ่งที่ต้องคำนึงถึงคือ แพร์ไวส์อัลกอริทึมแบบใช้เทรสโอส 2 ระดับค่อนข้างที่จะไม่เหมาะสมกับข้อมูลการร้องขอของไคลเอนต์ที่ได้จากปลายทางหรือเซิร์ฟเวอร์ปลายทาง เพราะข้อมูลที่ได้นั้นไอพีแอดเดรสอาจจะไม่ใช้ไอพีแอดเดรสจริงๆของผู้ใช้ก็ได้ อีกทั้งถ้าเป็นข้อมูลล็อกไฟล์จากเซิร์ฟเวอร์ปลายทางแล้ว ข้อมูลส่วนที่สองซึ่งก็คือส่วนการทำงานที่มีการสร้างฮิ้นเฉพาะของแต่ละไคลเอนต์จะไม่ค่อยมีผลกับอัตรานอมัลไลซ์รีคอลหรืออัตรานอมัลไลซ์พรีซิชั่นมากนั้นเพราะรูปแบบ การร้องขอของไคลเอนต์ส่วนใหญ่ที่มาถึงเซิร์ฟเวอร์น่าจะมีรูปแบบที่คล้ายๆกัน เพราะรีซอร์สที่อยู่บนเซิร์ฟเวอร์มีจำกัด ทำให้การกระจายตัวของรูปแบบการร้องขอของไคลเอนต์ไม่ได้แตกต่างกันนั้นแต่สำหรับข้อมูลที่ได้จากต้นทางทางด้านไคลเอนต์ เช่น พร็อกซี่ทางด้านไคลเอนต์แพร์ไวส์อัลกอริทึมเหมาะสมมากที่จะนำมาใช้ เพราะไอพีแอดเดรสที่ได้เป็นไอพีแอดเดรสจริงๆของไคลเอนต์อีกทั้งการกระจายตัวของการร้องขอของไคลเอนต์มีมากกว่าข้อมูลการร้องขอที่ได้จากเซิร์ฟ เวอร์ปลายทางแน่นอน ด้วยเหตุผลที่ได้กล่าวมาแล้วทั้งหมดก็เพียงพอที่จะกล่าวได้ว่าแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับเป็นอัลกอริทึมหนึ่งที่เหมาะสมในการนำมาใช้ในกระบวนการพรีเฟรชชิ่งได้ดี

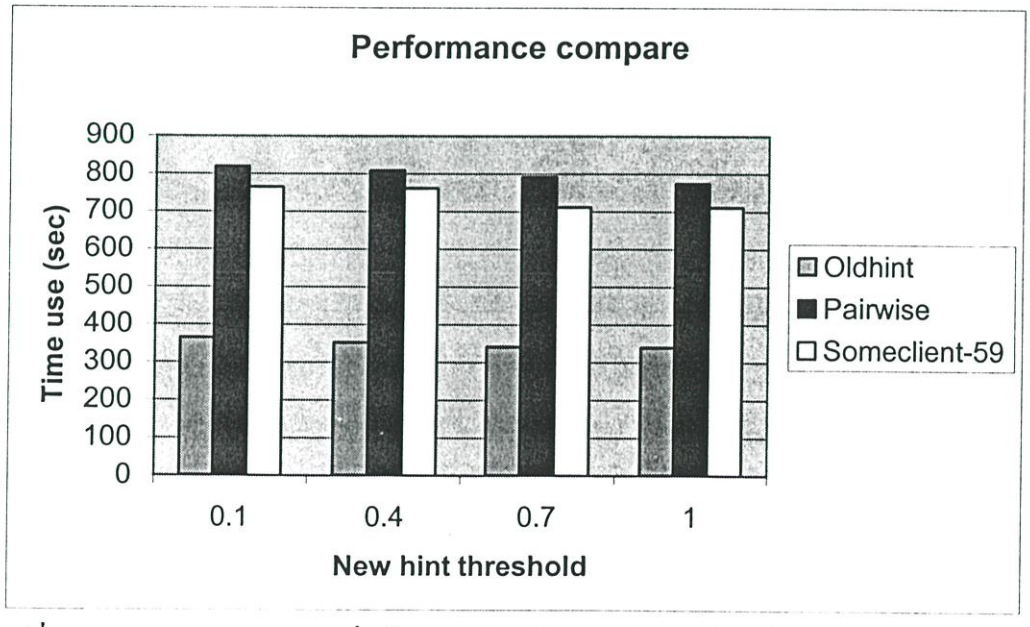
6.2 ทรัพยากรที่ใช้ในการทำงานของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

การทำงานของแพร์ไวส์อัลกอริทึม, การทำงานแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

และการทำงานแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุง ต้องมีการใช้ทรัพยากรในการทำงานที่แตกต่างกัน และก็ใช้เวลาในการทำงานที่แตกต่างกันด้วย ดังรูปที่ 6.1



รูปที่ 6.1 แสดงเวลาการทำงานที่ใช้ของแพร์ไวส์อัลกอริทึม, แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงของข้อมูลที่ได้จากทางด้านเซิร์ฟเวอร์



รูปที่ 6.2 แสดงเวลาการทำงานที่ใช้ของแพร์ไวส์อัลกอริทึม, แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงของข้อมูลที่ได้จากทางด้านไคลเอนต์

จะเห็นว่าในการทำงานของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและจะเห็นว่าในการทำงานของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงแล้ว จะใช้เวลาในการสร้างฮิ้นที่ค่อนข้างนานกว่าแพร์ไวส์อัลกอริทึมเดิม เพราะต้องมีการคำนวณในส่วนการทำงานที่สองเพิ่มขึ้นมาก

สำหรับขนาดของเมมโมรี่ (Memory) ที่ใช้นั้นแพร์ไวส์อัลกอริทึมเดิม, แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุง จะใช้เมมโมรี่สูงสุดประมาณ 447 MB, 978 MB และ 897 MB ตามลำดับ และสำหรับข้อมูลจากทางด้านเซิร์ฟเวอร์ในบทที่ 4 และ 5 สำหรับข้อมูลจากทางด้านไคลเอนต์จากบทที่ 4 และ 5 นั้นแพร์ไวส์อัลกอริทึมเดิม, แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับและแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุง จะใช้เมมโมรี่สูงสุด 421 MB, 945 MB และ 891 MB ตามลำดับ

จะเห็นว่าการทำงานแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงจะช่วยลดขนาดเมมโมรี่ที่ใช้ได้ได้ประมาณ 10 % และช่วยลดเวลาในการในการทำงานของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับประมาณ 7-10 % แต่ก็ยังคงใช้เมมโมรี่และเวลาที่สูงกว่าแพร์ไวส์อัลกอริทึมเดิมประมาณเกือบๆ 2 เท่าทั้งในขนาดของเมมโมรี่ ส่วนด้านเวลาแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับที่ได้รับการปรับปรุงแล้วก็ยังคงใช้เวลาในการทำงานนานกว่าแพร์ไวส์อัลกอริทึมเดิมประมาณ 2 เท่า และการใช้เมมโมรี่, เวลาในการทำงานก็จะเพิ่มขึ้นถ้ามีข้อมูลที่นำมาสร้างฮิ้นที่เพิ่มขึ้น

แม้ว่าแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับจะทำให้มีการใช้เมมโมรี่และเวลาในการทำงานที่เพิ่มขึ้นแต่สำหรับองค์กรขนาดกลางที่มีจำนวนผู้ใช้ไม่มากนัก แพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับก็เป็นวิธีการน่าสนใจในการลดเวลาในการเข้าถึงของผู้ใช้วิธีการหนึ่ง

6.3 การเลือกใช้เทรสโฮลในส่วนการทำงานแต่ละส่วน

สำหรับการเลือกใช้เทรสโฮลสำหรับส่วนการทำงานแต่ละส่วนของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับนั้นส่วนใหญ่จะขึ้นกับข้อมูลที่นำมาใช้ในการสร้างฮิ้น ซึ่งจากผลการจำลองสถานการณ์ที่ได้กล่าวถึงในบทนี้ ก็สามารถสรุปการเลือกใช้เทรสโฮลต่างๆได้ดังนี้ ถ้าใช้ข้อมูลการร้องขอของผู้ใช้ที่ได้จากฝั่งไคลเอนต์นั้นควรที่จะเลือกใช้ค่าเทรสโฮลของส่วนการทำงานแรก (ส่วนของแพร์ไวส์อัลกอริทึมเดิม, P_{t1}) ที่ค่าสูง 0.7 – 1.0 เพราะต้องการฮิ้นที่เกิดจากการร้องขอมากครั้งจากผู้ใช้ทั้งหมด ส่วนค่าเทรสโฮลในส่วนการทำงานที่สอง (ส่วนที่เพิ่มเติม, P_{t2}) ก็ควรเลือกใช้ค่าเทรสโฮลที่อยู่ในช่วง 0.5 – 1.0 ขึ้นอยู่กับค่าอัตรานอมัลไลซ์รีคอล (Normalized Recall Rate) และอัตรานอมัลไลซ์พรีซิชั่น (Normalized Precision Rate) ที่ต้องการ ถ้าต้องการค่าอัตรานอมัลไลซ์รีคอล (Normalized Recall Rate) ที่สูงการอาจจะเลือกค่า P_{t2} ที่ 0.5 ซึ่งให้ค่าอัตรานอมัลไลซ์รีคอล

(Normalized Recall Rate) ที่สูงในขณะที่ค่าอัตรานอมัลไลซ์พรีซิชั่น (Normalized Precision Rate)ที่ยังคงสูงอยู่ แต่ถ้าต้องการค่าอัตรานอมัลไลซ์พรีซิชั่น (Normalized Precision Rate) ที่สูงๆ ก็อาจจะเลือกใช้ค่า P_{t2} ที่ 0.7 ซึ่งจะให้ค่าอัตรานอมัลไลซ์พรีซิชั่น (Normalized Precision Rate) ที่สูงและยังคงให้ค่าอัตรานอมัลไลซ์รีคอล (Normalized Recall Rate) ที่สูงในระดับหนึ่ง

สำหรับถ้าใช้ข้อมูลที่มาจากด้านเซิร์ฟเวอร์ก็คือเลือกใช้ค่า P_{t1} ที่สูงเช่นกันอาจจะอยู่ในช่วง 0.7 – 1.0 เพราะต้องการฮิ้นที่ได้จากการร้องขอมากๆ ครั้งของผู้ใช้ทั้งหมดเท่านั้นในขณะที่ควรใช้ค่า P_{t2} นั้นควรที่จะใช้ค่าที่ไม่สูงมากเพราะต้องการที่ได้ฮิ้นที่ครอบคลุมกับการร้องขอผู้ใช้คนใดๆคนหนึ่ง (ซึ่งอาจจะเป็นผู้ใช้หลายๆคนที่มาจาก 1 ไอพีแอดเดรสก็ได้) อาจจะอยู่ในค่า 0.3 – 0.7 เพราะถ้าใช้ค่า P_{t2} ที่สูงมากไปฮิ้นสำหรับส่วนการทำงานที่สองของข้อมูลที่ได้จากด้านเซิร์ฟเวอร์จะเป็นฮิ้นที่ค่อนข้างไม่ครอบคลุมการร้องขอของไอพีแอดเดรสหนึ่งไอพีนั้น

6.4 ข้อเสนอแนะ

จากการทำงานแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับนั้นในส่วนการทำงานที่สองจะต้องมีการค้นหาเกี่ยวกับเคาเตอร์ต่างๆของผู้ใช้เฉพาะคน ซึ่งเวลาในการค้นหาเคาเตอร์ต่างๆสามารถปรับปรุงได้โดยอาจจะใช้วิธีการค้นหาใหม่ที่สามารถใช้เวลาในการค้นหาที่ลดลง ซึ่งจะช่วยให้เวลาในการทำงานโดยรวมของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับลดลงได้

6.5 การนำแพร์ไวส์อัลกอริทึมแบบใช้ 2 เทรสโฮลไปใช้งานจริง

การนำแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับไปใช้งานจริงนั้นสิ่งที่ต้องคำนึงอย่างแรกคือฮาร์ดแวร์ที่จะใช้เป็นเครื่องที่จะใช้ในการสร้างฮิ้น ควรที่จะมีขนาดเมมโมรีที่ค่อนข้างสูง เพราะในการสร้างฮิ้นเฉพาะสำหรับผู้ใช้คนหนึ่ง (ส่วนการทำงานที่สอง) ต้องมีการเก็บข้อมูลเฉพาะของผู้ใช้แต่ละคน สิ่งที่ต้องคำนึงต่อไปคือตำแหน่งที่ฮิ้นจะถูกสร้างและเก็บไว้ซึ่งถ้าเป็นฮิ้นที่สร้างจากข้อมูลทางด้านเซิร์ฟเวอร์ฮิ้นก็จะถูกเก็บไว้ที่เซิร์ฟเวอร์แล้วจะถูกส่งกลับไปให้ทางผู้ใช้ทำการพรีเฟรช อีกครั้งหนึ่ง ส่วนในการสร้างฮิ้นที่ได้จากข้อมูลด้านไคลเอนต์ควรที่จะใช้เครื่องที่เป็นพร็อกซี่ทางด้านไคลเอนต์เป็นเครื่องที่ทำการสร้างฮิ้น เพราะฮิ้นที่สร้างก็ไม่จำเป็นต้องส่งไปให้ใคร พร็อกซี่จะเป็นเครื่องที่ทำการพรีเฟรชอีก วิธีการยังมีข้อดีอีกอย่างคือไม่ต้องสูญเสียช่องสัญญาณของข้อมูลส่วนที่เป็นฮิ้นที่ส่งกลับมาให้ทางผู้ใช้ (พร็อกซี่ของผู้ใช้ทำการพรีเฟรช)

การทำการสร้างฮิ้นนำควรที่จะทำงานในแบบออฟไลน์ เพราะต้องมีการประมวลผลที่สูงไม่ว่าจะใช้แพร์ไวส์อักลอริทึมแบบเดิมหรือแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ และค่อนข้างใช้เวลานานในการสร้างฮิ้น

เอกสารอ้างอิง

[1] J. C. Mogul, "Hinted caching in the Web," in *Proc. ACM SIGOPS European Workshop*, 1996.

[2] E.Cohen, B. Krishnamurthy, and J. Rexford, "Evaluating server-assisted cache replacement in the web," in *Proc. European Symposium on Algorithms*, August 1998.

[3] V.N. Padmanabhan and J. C. Mogul, "Using predictive prefetching to improve World Wide Web Latency," *Computer Communication* Review, vol 26, no. 3,pp. 22-36, 1996

[4] M. Crovella and P. Barford, "The network effects of prefetching," in *Proc IEEE INFOCOM*, April 1998.

[5] James F. Kurose, "Computer Networking: A Top-Down Approach Featuring the Internet", Addison Wesley Publisher, History of Computer Networking and the Internet section, Chapter 1.

[6] Balachander Krishnamurthy and Jennifer Rexford, "Web protocols and Practice", Addison Wesley Publisher, Chapter 2. page 176 – 210.

[7] ชูเกียรติ นาคพิทักษ์กุลและวิทยา ต่อศรีเจริญ, "*เริ่มต้นสร้างโฮมเพจด้วย HTML*", สำนักพิมพ์ WITTY Group, หน้า 26 – 27.

[8] James F. Kurose, "Computer Networking: A Top-Down Approach Featuring the Internet", Addison Wesley Publisher, The Web and HTTP section, Chapter 2., page 84-87.

[9] James F. Kurose, "Computer Networking: A Top-Down Approach Featuring the Internet", Addison Wesley Publisher, The Web and HTTP section, Chapter 2, page 88-96.

[10] "RFC 2101", www.scit.wlv.ac.uk/rfc/rfc21xx/RFC2101.html .

[11] Network Address Translation : NAT, http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci214107,00.html.

ภาคผนวก ก.
ผลงานวิจัยที่ได้รับการตีพิมพ์

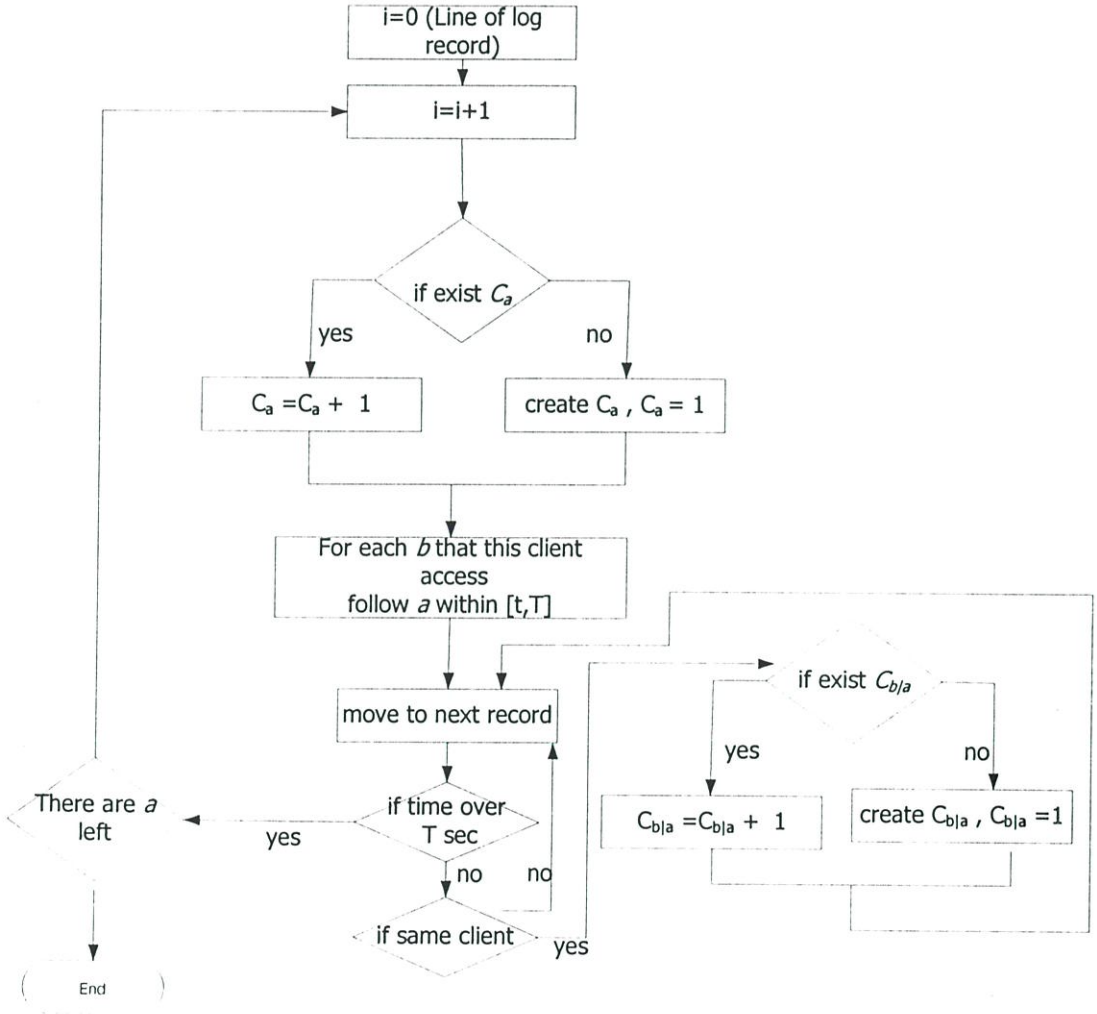
[1] Chotipat Pornavalai and Wisit Phatchoo, "An Efficient Two-Level Thresholds Pairwise Alogrithm for Predicting Request to Web Server", The Second International Symposium On Communication and Information Technology, 23-25 October 2002.

ภาคผนวก ข.

Flowchart 1 แสดงขั้นตอนการทำงานในส่วน Count Step ของแพร์ไวส์อัลกอริทึม

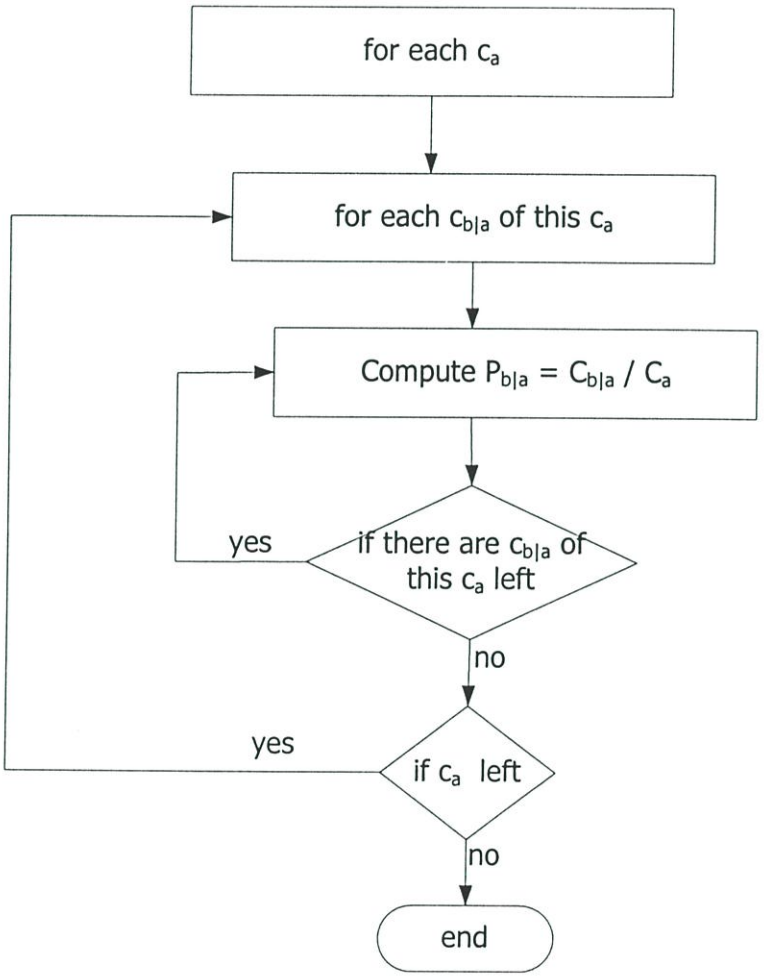
นิยาม C_a เป็นจำนวนครั้งที่มีการร้องขอเว็บเพจ a ใดๆ

$C_{b|a}$ เป็นจำนวนครั้งที่มีการร้องเว็บเพจ b ใดๆ ต่อจากเว็บเพจ a ใดๆ



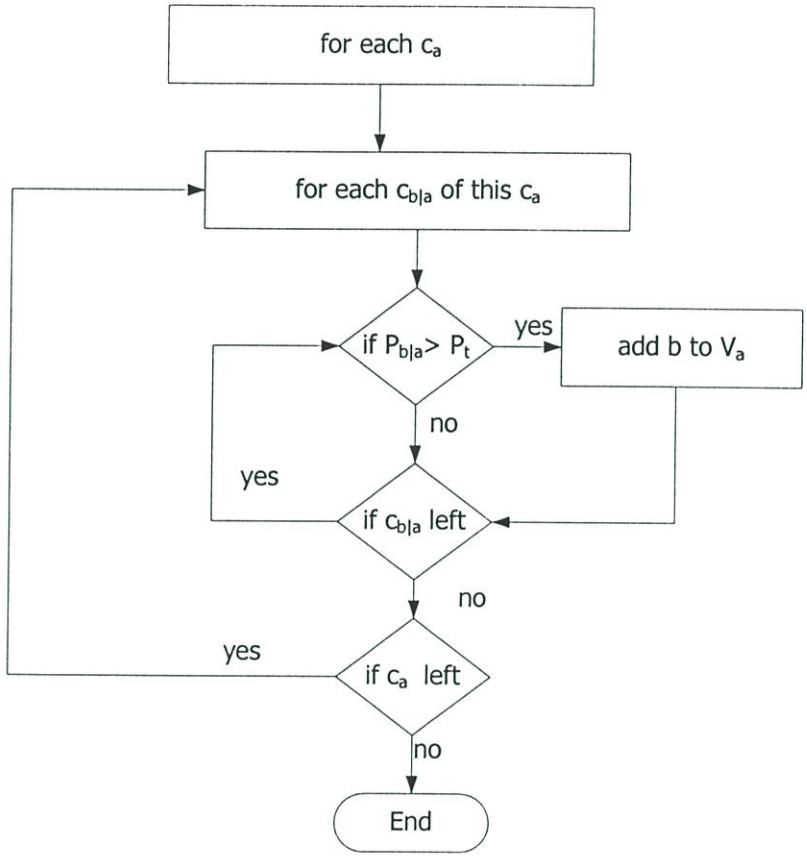
Flowchart 2 แสดงขั้นตอนการทำงานในส่วน Compute Probabilities Step ของแพร์ไวส์อัลกอริทึม

นิยาม $P_{b|a}$ เป็นความน่าจะเป็นที่เว็บเพจ b ใดๆขึ้นกับเว็บพจ a ใดๆ



Flowchart 3 แสดงขั้นตอนการทำงานในส่วน Hint Construction Step ของแพร์ไวส์อัลกอริทึม

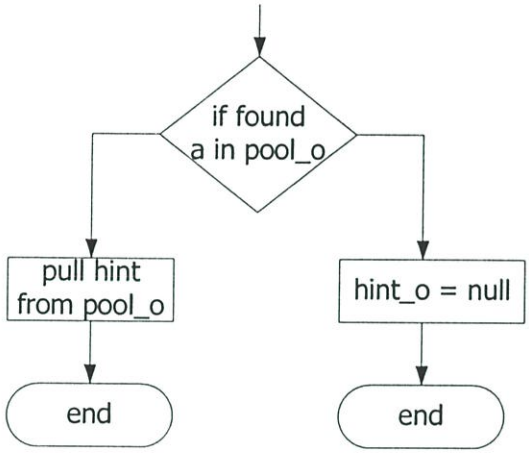
นิยาม P_t เป็นความน่าจะเป็นเทรสโฮล



Flowchart 4 แสดงขั้นตอนการทำงานในส่วน Send hint Step ของแพร์ไวส์อัลกอริทึม

นิยาม *Pool_o* เป็นกลุ่มของฮิ้นโวลลุ่มที่ได้จากแพร์ไวส์อัลกอริทึม

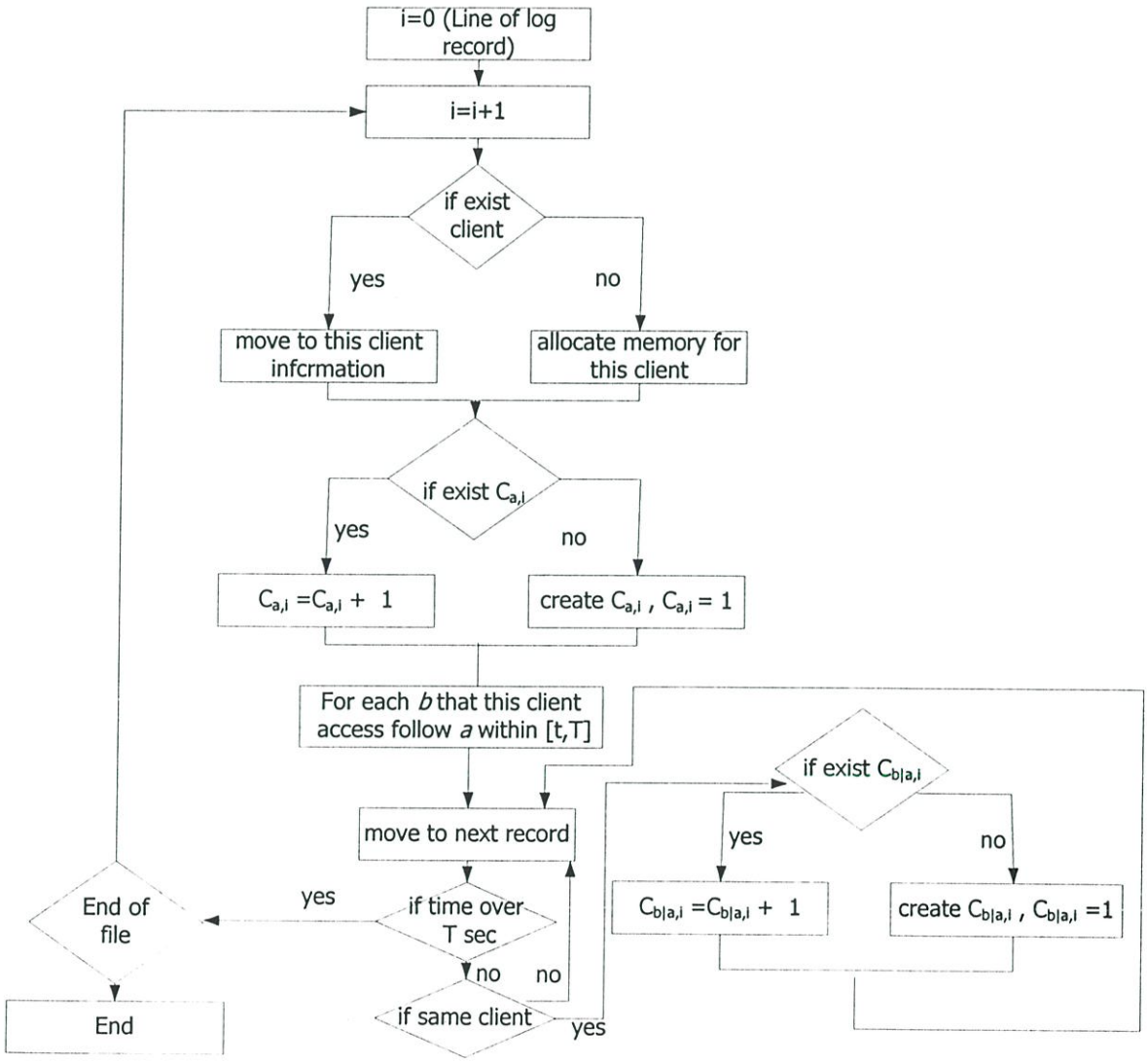
Hint_o เป็นฮิ้นที่จะถูกส่งกลับไปให้ผู้ใช้ทำการพรีเฟรช



Flowchart 5 แสดงขั้นตอนการทำงานในส่วน Count Step for specific client ของแพร์ไวส์อัลกอริทึม แบบใช้เทรสโฮล 2 ระดับ

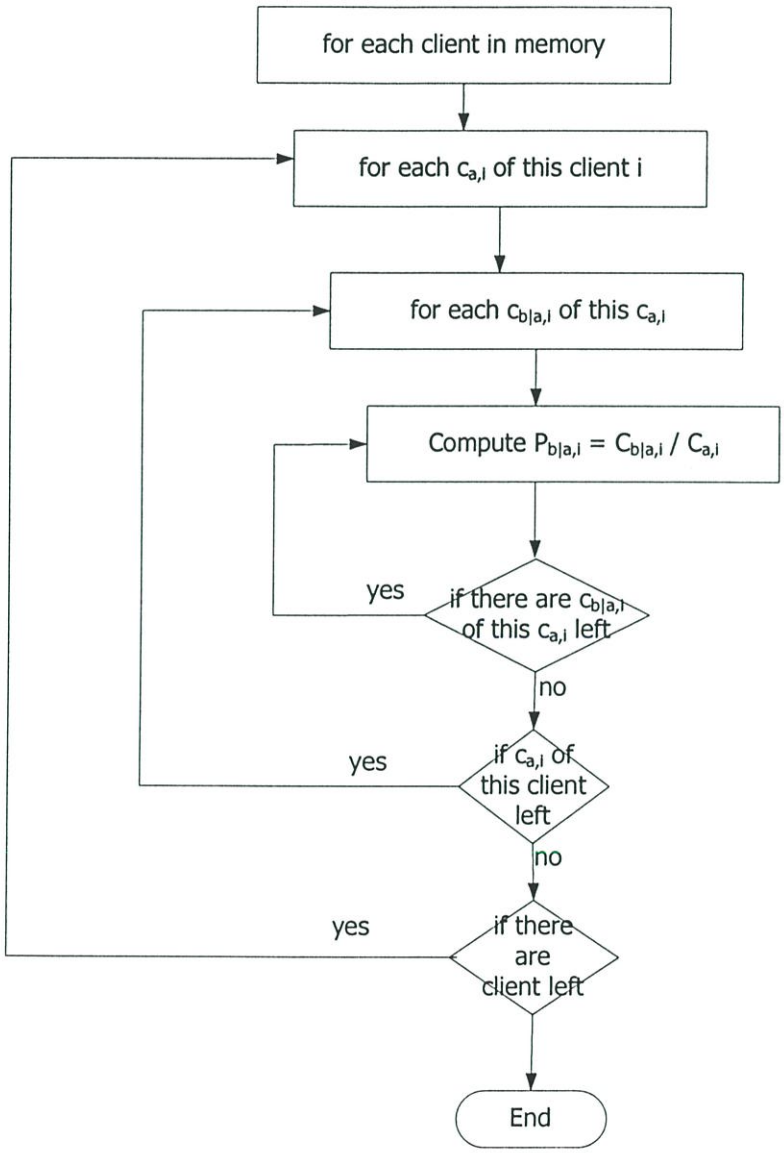
นิยาม $C_{a,i}$ เป็นจำนวนครั้งที่ผู้ใช้คนที่ i ร้องขอเว็บเพจ a ใดๆ

$C_{b|a,i}$ เป็นจำนวนครั้งที่ผู้ใช้คนที่ i มีการร้องเว็บเพจ b ใดๆ ต่อจากเว็บเพจ b ใดๆ



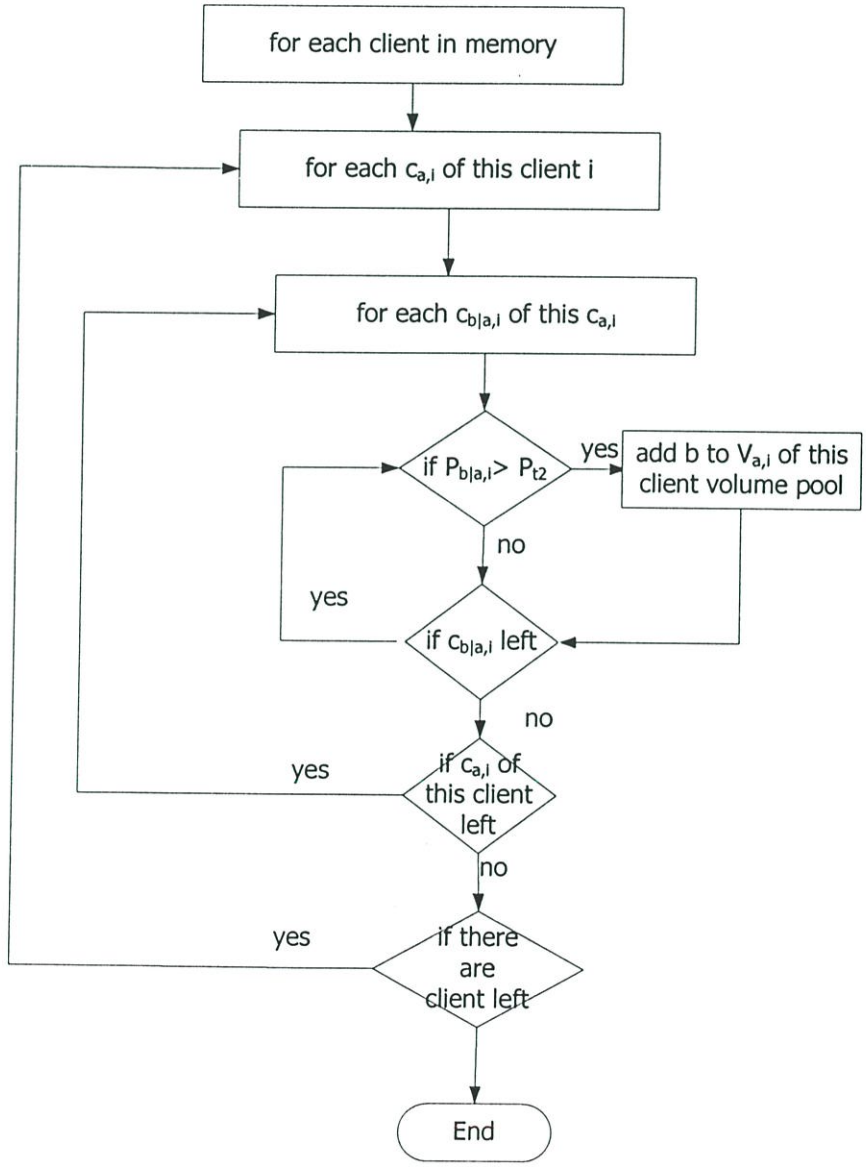
Flowchart 6 แสดงขั้นตอนการทำงานในส่วน Compute Probabilities Step for Specific client ของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

นิยาม $P_{b|a,i}$ เป็นความน่าจะเป็นของผู้ใช้คนที่ i ที่ร้องขอเว็บเพจ b ขึ้นกับเว็บเพจ a ใดๆ



Flowchart 7 แสดงขั้นตอนการทำงานในส่วน Hint Construction Step for Specific client ของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

นิยาม P_{t2} เป็นความน่าจะเป็นเทรสโฮลของส่วนการทำงานที่สองของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ



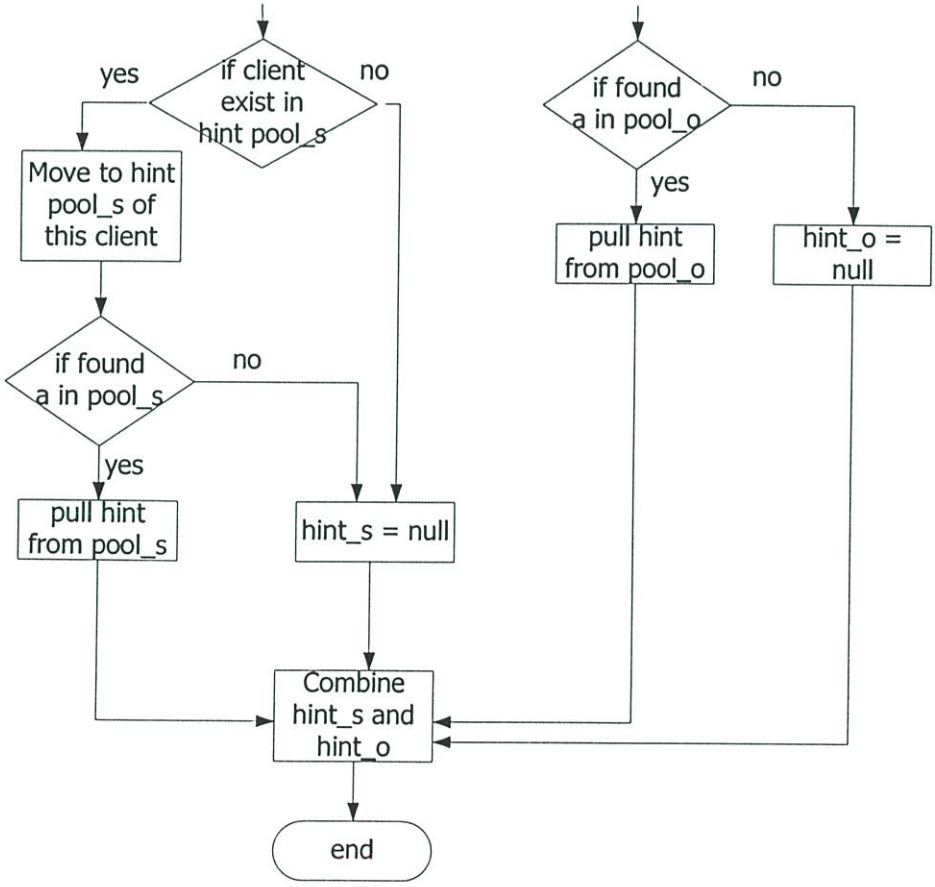
Flowchart 8 แสดงขั้นตอนการทำงานในส่วน Send Hint Step for Specific client ของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

นิยาม *Pool_o* คือกลุ่มของฮิ้นโวลลุ่มที่ได้จากส่วนการทำงานที่แรกของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

Pool_s คือกลุ่มของฮิ้นโวลลุ่มที่ได้จากส่วนการทำงานที่สองของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

Hint_o เป็นฮิ้นที่ได้จากส่วนการทำงานแรกของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ

Hint_s เป็นฮิ้นที่ได้จากส่วนการทำงานสองของแพร์ไวส์อัลกอริทึมแบบใช้เทรสโฮล 2 ระดับ



ประวัติผู้เขียน

ชื่อ-นามสกุล	นายวิศิษฎ์ พัฒน์ชู
วัน เดือน ปีเกิด	23 สิงหาคม 2521 ที่กรุงเทพมหานคร
ที่อยู่	52 ซอยพึ่งมี 38 ถนนสุขุมวิท 93 แขวงบางจาก เขตพระโขนง กรุงเทพ 10260 โทร. 0-2311-0371
ประวัติการศึกษา	2539 วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ความชำนาญเฉพาะด้าน

1.) ระบบเครือข่ายคอมพิวเตอร์

2.) เว็บโปรแกรมด้วย ASP, JSP

ประสบการณ์การทำงานและผลงานวิจัย

พ.ศ. 2543 – ปัจจุบัน	ตำแหน่งผู้เชี่ยวชาญด้านเครือข่ายบริษัทดาต้าโปรคอมพิวเตอร์ซิสเต็มส์
	- ให้คำปรึกษาด้านระบบเครือข่ายคอมพิวเตอร์
	- ออกแบบและแก้ไขปัญหาของระบบเครือข่าย