การศึกษาเปรียบเทียบการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลบนอินเทอร์เน็ต เพื่อปรับปรุงประสิทธิภาพของเว็บแคช

นาย กัลย์ แก้วแก่น

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2543

ISBN 974-13-0945-7

COMPARATIVE STUDY OF COMPRESSION ALGORITHMS FOR DATA CONTENTS ON THE INTERNET FOR IMPROVING WEB CACHING PERFORMANCE

Mr. Kun Kaewken

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2000

ISBN 974-13-0945-7

หัวข้อวิทยานิพนธ์	การศึกษาเปรียบเทียบการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลบนอินเทอร์เน็ตเพื่อปรับปรุงประสิทธิภาพของเว็บแคช
โดย	นาย กัลย์ แก้วแก่น
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ บุญชัย โสวรรณวณิชกุล)

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

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

.....กรรมการ
(อาจารย์ ดร. ชัย พงศ์พันธุ์ภาณี)

กัลย์ แก้วแก่น : การศึกษาเปรียบเทียบการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลบนอินเทอร์เน็ตเพื่อปรับปรุงประสิทธิภาพของเว็บแคช (COMPARATIVE STUDY OF COMPRESSION ALGORITHMS FOR DATA CONTENTS ON THE INTERNET FOR IMPROVING WEB CACHING PERFORMANCE) อ.ที่ปรึกษา : ดร.ณัฐวุฒิ หนูไพโรจน์, 66 หน้า. ISBN 974-13-0945-7.

ในช่วงระยะหลายปีที่ผ่านมา การใช้เว็บมีอัตราการเพิ่มขึ้นอย่างรวดเร็วและต่อเนื่อง ก่อให้เกิดความคับคั่งบนระบบเครือข่าย ทำให้ผู้ให้บริการอินเทอร์เน็ตต้องเพิ่มแบนด์วิดท์ในการเชื่อมต่อสู่อินเทอร์เน็ต งานวิจัยด้านเว็บแคชที่ผ่านมายังไม่มีการศึกษาเรื่องการใช้ขั้นตอนวิธีในการบีบอัดข้อมูลมาใช้ในแคช ดังนั้นงานวิจัยนี้จึงได้ทำการศึกษาถึงการใช้ขั้นตอนวิธีการบีบอัดข้อมูลแบบต่างๆ มาใช้กับข้อมูลที่ปรากฏอยู่ในเว็บแคช โดยใช้ข้อมูลการใช้เว็บของจุฬาลงกรณ์มหาวิทยาลัยเป็นกรณีศึกษา

การวิเคราะห์หาประเภทข้อมูลที่มีการเรียกขอมาก พบว่า รูปภาพจิฟ เอกสารเอชทีเอ็มแอล และรูปภาพเจเพก มีการเรียกขอมากที่สุดเรียงตามลำดับ ข้อมูลที่สามารถทำการบีบอัดได้ที่นำมาทำการศึกษา คือ ข้อมูลตัวอักษร และรูปภาพเจเพก เมื่อแบ่งข้อมูลออกเป็นกลุ่มเท่าๆ กัน 8 กลุ่ม วัดค่าเวลาเฉลี่ยในการเรียกขอข้อมูล พบว่าเวลาที่ใช้เมื่อเกิดแคชมิสมากกว่าเวลาที่ใช้เมื่อเกิดแคชฮิท 5 วินาทีสำหรับข้อมูลประเภทตัวอักษร และ 8 วินาทีสำหรับข้อมูลรูปภาพเจเพก เมื่อทำการสุ่มข้อมูลแบบไม่ซ้ำจากอินเทอร์เน็ตมาทดลองกับขั้นตอนวิธีการบีบอัดข้อมูลแบบต่างๆ วัดค่าอัตราการบีบอัด เวลาที่ใช้ในการบีบอัดและเวลาที่ใช้ในการขยายข้อมูล และทำการเปรียบเทียบการตัดสินใจเลือกขั้นตอนวิธีการบีบอัด ใช้อัตราการบีบอัดเป็นเครื่องมือในการตัดสินใจ เนื่องจากเวลาที่ใช้ในการบีบอัดและเวลาในการขยายข้อมูล มีค่าน้อยกว่าค่าส่วนต่างของเวลาที่ได้จากการวิเคราะห์เวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล ผลการวิจัยสรุปว่า ข้อมูลประเภทตัวอักษรขนาดเล็กควรใช้ LZA ขนาดใหญ่ควรใช้ ZIP ข้อมูลรูปภาพเจเพกขนาดเล็กควรใช้ LZA ขนาดกลางควรใช้ ZIP ขนาดใหญ่ควรใช้ LZA

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2543 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

4070207221 : MAJOR COMPUTER SCIENCE

KEYWORD : WORLD WIDE WEB / WWW / CACHE / DATA COMPRESSION ALGORITHM

KUN KAEWKEN : COMPARATIVE STUDY OF COMPRESSION ALGORITHMS FOR DATA CONTENTS ON THE INTERNET FOR IMPROVING WEB CACHING PERFORMANCE.

THESIS ADVISOR: NATAWUT NUPAIROJ, Ph. D. 66 pp.

ISBN 974-13-0945-7.

During the past few years, Web usage has rapidly increased. This causes the bottleneck on network and ISPs have to extend the bandwidth through the Internet. There has been no research on the compression algorithm for web caching. In this thesis we study the compression algorithm for web contents based on web usage data of Chulalongkorn University.

Our study indicated that GIF, HTML and JPEG are the most requested content type. In our study, we examine the compression algorithm for TEXT and JPEG formats. Dividing data into 8 equal groups, we measured the average request time and found that cache miss time is 5 seconds and 8 seconds greater than cache hit time for TEXT and JPEG respectively. We randomly retrieved data from the Internet and measured the performance of data compression algorithms regrarding compression ratio, compression time, and decompression time. We use compression ratio as a metric for comparing the compression algorithms because both compression time and decompression time are less than the average request time from the analysis above. Our study concludes that the LZA is suitable for small-sized TEXT and the ZIP is suitable for large-sized TEXT. For small-, medium-, and large-sized JPEG, we recommend LZA, ZIP, and LZA respectively.

Department Computer Engineering . Student's signature .

Field of study Computer Science . Advisor's signature .

Academic year 2000 . Co-advisor's signature .

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ อ.ดร.ณัฐวุฒิ หนูไพโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ในการวิจัยมาด้วยดีตลอด

ขอขอบพระคุณมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร (Computer and Communication Education Foundation) และมูลนิธิเพื่อการศึกษาและวิจัยวิทยาศาสตร์คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่สนับสนุนทุนการศึกษาและทุนในการทำงานวิจัยชิ้นนี้

ขอขอบคุณ พี่ๆ เพื่อนๆ และน้องๆ ที่ได้ให้ความช่วยเหลือ

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณมารดา ครอบครัว และคุณเขมะ โชคชัยภักดี ซึ่งสนับสนุนและให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

กัลย์ แก้วแก่น

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญภาพ

หน้า

สารบัญตาราง

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมา

ในช่วงระยะเวลา 2-3 ปีที่ผ่านมา การใช้เวิลด์ไวด์เว็บ (WWW) เป็นที่นิยมกันอย่างแพร่หลายมากและมีอัตราการขยายตัวเพิ่มขึ้นอย่างรวดเร็วและต่อเนื่อง จะเห็นได้จากสัดส่วนของการจราจรในระบบเครือข่าย (Network Traffic) ที่เกิดขึ้นอันเนื่องมาจากการเรียกขอข้อมูลโดยใช้โพรโทคอลเอชทีทีพี (HTTP) เพิ่มขึ้นเป็นอย่างมาก ทำให้การทำงานของเซิร์ฟเวอร์ (Server Load) เพิ่มขึ้นมาก ก่อให้เกิดการคับคั่งบนระบบเครือข่าย (Network Congestion) ซึ่งส่งผลให้เวลาในการตอบสนองต่อผู้ใช้ (User Response Time) เพิ่มมากขึ้น และปริมาณช่องสัญญาณ (Bandwidth) ที่บริการอื่นๆ ใช้ร่วมกันบนอินเทอร์เน็ต เช่น FTP Telnet และ News เป็นต้น สามารถนำไปใช้ได้ลดลง

เพื่อแก้ปัญหาที่กล่าวมาข้างต้น จึงได้มีการนำเทคนิคเว็บแคช (Web cache) เข้ามาใช้ โดยหลักการของเว็บแคชคือ การนำข้อมูลที่ถูกเรียกขอในช่วงเวลาที่ผ่านมา เก็บเข้าสู่หน่วยความจำหลัก (Main Memory) หรือ ภายในเนื้อที่เก็บข้อมูล (Disk Space) ของเว็บแคชเซิร์ฟเวอร์ (Web Cache Server) เมื่อมีผู้ใช้รายอื่นเรียกขอข้อมูลเดียวกันนี้ เว็บแคชเซิร์ฟเวอร์ จะทำการส่งข้อมูลที่ถูกเก็บไว้ภายในแคช (Cache) ให้ผู้เรียกขอแทนการติดต่อกับเซิร์ฟเวอร์ต้นทางโดยตรง ซึ่งจะส่งผลให้ปริมาณการจราจรที่เกิดในระบบเครือข่ายลดลง และทำให้เวลาในการตอบสนองต่อผู้ใช้ลดลงด้วย

ถึงแม้ว่าการใช้เว็บแคชจะช่วยแก้ปัญหาในเรื่องความคับคั่งของปริมาณการจราจรในระบบเครือข่าย เว็บแคชยังมีข้อจำกัดอยู่หลายประการ กล่าวคือ

- ข้อมูลที่เก็บอยู่ภายในแคช ไม่เป็นข้อมูลที่เป็นปัจจุบัน ส่งผลให้ต้องทำการเรียกขอข้อมูลจากเซิร์ฟเวอร์โดยตรง

- ข้อมูลที่ต้องการเรียกขอยังไม่มีการถูกจัดเก็บอยู่ภายในแคช หรือถูกลบออกจากแคชไปก่อนที่จะมีการเรียกขอ (Cache Miss) ซึ่งจะทำให้เวลาการตอบสนองต่อผู้ใช้มากขึ้น เนื่องจากมีการเสียเวลาค้นหาข้อมูลภายในแคชเพิ่มเติมขึ้นมานอกเหนือจากการเรียกขอข้อมูลจากเซิร์ฟเวอร์

- ข้อมูลที่สามารถเก็บลงในแคชได้ต้องเป็นข้อมูลประเภทสแตติก (Static Document) เช่น เว็บเพจ เสียง หรือภาพเคลื่อนไหว เป็นต้น ไม่สามารถเก็บข้อมูลประเภทไดนามิก (Dynamic Document) เช่น ผลของการค้นหาข้อมูลจากฐานข้อมูล หรือเอกสารเว็บที่มีการใส่ป้ายระบุ (Tag) ประเภท No-Cache เป็นต้น ได้

นอกจากนี้ งานวิจัยของ Standidge และคณะ [1] พบว่าค่าฮิทเรโช (Hit Ratio) ซึ่งแสดงถึงประสิทธิภาพของการพบข้อมูลในเว็บแคช จะมีค่าสูงสุดประมาณ 30-50% ซึ่งนับว่าน้อยมากเมื่อเทียบกับค่าฮิทเรโชของแคชในซีพียูซึ่งจะมีค่ามากกว่า 80% ดังนั้นจึงได้มีงานวิจัยจำนวนมากเพื่อศึกษาแนวทางการเพิ่มประสิทธิภาพของการใช้แคช เช่น

- การศึกษาเพื่อทำการปรับปรุงขั้นตอนวิธีการแทนที่ (Replacement Algorithms) [2], [3], [4] เป็นการศึกษาเพื่อเพิ่มประสิทธิภาพของเว็บแคชโดยการปรับปรุงขั้นตอนวิธีการแทนที่ข้อมูลในเว็บแคชโดยเน้นให้เอกสารที่คาดว่าจะถูกเรียกขอในอนาคตมีโอกาสอยู่ในเว็บแคชมากที่สุด

- โทโปโลยีของระบบแคช (Cache Topology) [5] เป็นการศึกษาการจัดโครงสร้างของเว็บแคชให้มีประสิทธิภาพโดยการใช้การติดต่อในระดับเครือข่าย (Network Level) แทนการติดต่อในระดับโปรแกรม (Application Level) เพื่อปรับปรุงการส่งข้อมูลระหว่างเว็บแคชให้รวดเร็วขึ้น

- การดึงข้อมูลล่วงหน้า (Pre-Fetching) [6] เป็นการเพิ่มประสิทธิภาพของเว็บแคชโดยการคาดการณ์ถึงยูอาร์แอลที่จะเรียกใช้ในอนาคตและทำการเรียกขอข้อมูลจากยูอาร์แอลนั้นมาล่วงหน้า

- การใช้ระบบแคชแบบลำดับชั้น (Hierarchical Caching) [7] เป็นการใช้การค้นหาข้อมูลระหว่างเว็บแคชเพื่อค้นหาเอกสารที่ถูกเก็บอยู่ในต่างเว็บแคชเพื่อลดการติดต่อไปยังเซิร์ฟเวอร์

- การปรับปรุงโครงสร้างของ HTTP [8] เป็นการศึกษาการลดจำนวนครั้งของการติดต่อเพื่อทำการเรียกขอเอกสารจากเซิร์ฟเวอร์

ถึงแม้ว่างานวิจัยต่างๆ ที่กล่าวมาข้างต้นจะสามารถเพิ่มประสิทธิภาพของเว็บแคชได้เป็นอย่างดี แต่อย่างไรก็ตามประสิทธิภาพของเว็บแคชยังถูกจำกัดด้วยขนาดของเนื้อที่ของแคช ซึ่งงานวิจัยของ Baentsch และคณะ [6] แสดงให้เห็นว่า โดยปกติแล้ว ฮิทเรโชที่ได้จะมีค่าที่ค่อนข้างต่ำเนื่องมาจากปัจจัยประการหนึ่ง คือ มีเนื้อที่ที่ใช้ในการเก็บข้อมูลอยู่จำกัด จากวิทยานิพนธ์ “การวิเคราะห์เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการแทนที่ในพร็อกซีแคช” [9] พบว่าเมื่อเนื้อที่แคชเพิ่มขึ้นจาก 1% เป็น 30% ของขนาดพื้นที่ที่สามารถเก็บข้อมูลได้ทั้งหมดโดยไม่มีการแทนที่ (Infinite Cache Size) จะทำให้ฮิทเรโชเพิ่มขึ้นจาก 20% เป็น 50% ซึ่งแสดงว่าการเพิ่มความจุของพื้นที่เก็บข้อมูลให้มากขึ้น จะทำให้ฮิทเรโชของเว็บแคชมากยิ่งขึ้น

วิธีการหนึ่งที่จะเพิ่มความจุของพื้นที่เก็บข้อมูล สามารถทำได้โดยการใช้ขั้นตอนวิธีการบีบอัดข้อมูลเพื่อลดขนาดของข้อมูลที่จัดเก็บ อย่างไรก็ดีขั้นตอนวิธีการบีบอัดข้อมูลนั้นมีอยู่หลากหลาย และมีความแตกต่างกันทั้งในแง่เวลาที่ใช้และความสามารถในการบีบอัดข้อมูล และนอกจากนี้ข้อมูลที่เก็บในเว็บแคชยังมีหลายประเภทและหลายขนาด ดังนั้นในงานวิจัยนี้จะศึกษาความ

สามารถของขั้นตอนวิธีการบีบอัดข้อมูลแบบต่างๆ กับผลที่มีต่อข้อมูลประเภทต่างๆ ที่มักปรากฏอยู่ในเว็บแคช ซึ่งผลของงานวิจัยนี้จะสามารถบ่งบอกถึงขั้นตอนวิธีการบีบอัดข้อมูลที่มีความเหมาะสมที่จะนำไปใช้ในเว็บแคชต่อไป

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาความสามารถของขั้นตอนวิธีการบีบอัดข้อมูลแบบต่างๆ กับผลที่มีต่อข้อมูลประเภทต่าง ๆ ที่มักปรากฏอยู่ในเว็บแคช

1.2.2 วิเคราะห์หาความเหมาะสมของการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลประเภทต่าง ๆ ในเว็บแคชโดยวัดจาก

1.2.2.1 สัดส่วนการบีบอัดข้อมูล

1.2.2.2 ความเร็วที่ใช้ในการบีบอัดข้อมูล

1.2.2.3 ความเร็วที่ใช้ในการขยายข้อมูล

1.3 ขอบเขตการวิจัย

1.3.1 วิทยานิพนธ์นี้ เป็นการศึกษาการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลประเภทที่สามารถเก็บอยู่ภายในเว็บแคชได้เท่านั้น

1.3.2 รูปแบบข้อมูลที่ใช้ศึกษาต้องมีความเป็นมาตรฐาน ที่ได้รับการรับรอง

1.3.3 วิธีการบีบอัดข้อมูลที่ศึกษา จะทำการศึกษาเฉพาะวิธีการบีบอัดข้อมูลโดยกระบวนการทางซอฟต์แวร์เท่านั้น ไม่รวมถึงวิธีการทางฮาร์ดแวร์

1.3.4 ขั้นตอนวิธีการในการบีบอัดข้อมูลที่นำมาใช้ในการทำวิจัย จะประกอบด้วยขั้นตอนวิธีการอย่างน้อย 4 ขั้นตอนวิธี

1.4 วิธีดำเนินงานวิจัย

1.4.1 ศึกษาทฤษฎีและวิธีการที่เกี่ยวข้องของขั้นตอนวิธีการบีบอัดข้อมูล

1.4.2 ศึกษาและวิเคราะห์ลักษณะของข้อมูลที่อยู่บนอินเทอร์เน็ต และข้อมูลที่ถูกเก็บลงในเว็บแคช

1.4.3 ทำการแบ่งกลุ่มตัวอย่างข้อมูลตามขนาดของแฟ้มข้อมูล

1.4.4 คำนวณหาขนาดตัวอย่างที่นำมาใช้เป็นตัวแทนของข้อมูลในแต่ละกลุ่ม

1.4.5 ทำการเรียกขอข้อมูลตัวอย่างจากอินเทอร์เน็ต

1.4.6 วิเคราะห์ขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลตัวอย่างและทำการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูล

1.4.7 สรุปผล แนวคิดในการประยุกต์และข้อเสนอแนะ

1.4.8 จัดทำรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

นำเสนอขั้นตอนวิธีการบีบอัดข้อมูลที่จะนำมาประยุกต์ใช้กับพร็อกซีเว็บแคช ที่เหมาะสมกับสภาพการใช้เว็บของจุฬาลงกรณ์มหาวิทยาลัย ซึ่งสามารถนำไปใช้ในการเพิ่มประสิทธิภาพของเว็บแคชที่มีอยู่ เพื่อทำให้เว็บแคชสามารถเก็บข้อมูลเพิ่มขึ้น

บทที่ 2
ทฤษฎีและแนวความคิดที่ใช้

งานวิจัยนี้จะใช้แนวคิด และทฤษฎีทางด้านการสื่อสารข้อมูลในระบบเครือข่าย (Data Network Communication) ด้านขั้นตอนวิธีการบีบอัดข้อมูล (Data Compression Algorithm) และการประยุกต์ใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับเว็บแคช ดังนี้

2.1 ด้านการสื่อสารข้อมูลในระบบเครือข่าย

เว็บแคช (Web Cache) คือ อุปกรณ์ที่ช่วยในการเพิ่มประสิทธิภาพของการสื่อสารข้อมูลบนอินเทอร์เน็ต โดยอาศัยแนวความคิดที่ว่าความคับคั่งที่เกิดขึ้นในระบบเครือข่าย เป็นสาเหตุให้การสื่อสารข้อมูลบนอินเทอร์เน็ตมีความล่าช้า เมื่อเกิดความคับคั่งบนจุดเชื่อมต่อหรือที่ปลายทางข้อมูลจะถูกทิ้งไป ซึ่งทำให้ปริมาณข้อมูลที่ส่ง (Throughput) ลดลง การติดตั้งวงจร (Circuits) เราท์เตอร์ หรือสวิทช์ที่มีความเร็วสูงขึ้นสามารถช่วยลดการเกิดความคับคั่งได้ แต่ไม่สามารถลดระยะเวลาการเดินทางของข้อมูล (Round-Trip Time) ระหว่างจุดได้ เนื่องจากปัจจัยเรื่องความเร็วแสง ซึ่งเป็นข้อจำกัดพื้นฐานของความล่าช้าบนระบบเครือข่าย (Network Delay)

แคชชิง (Caching) คือ การทำสำเนาของข้อมูลที่เป็นที่นิยมจากเซิร์ฟเวอร์ต้นทางมาเก็บไว้ในที่ ๆ ใกล้กับผู้ใช้เพื่อลดระยะเวลาการเดินทางของข้อมูล โดยใช้แนวคิดที่นำมาจากการออกแบบคอมพิวเตอร์และระบบเครือข่าย เช่น ซีพียูใช้แคชเพื่อรับส่งชุดคำสั่งปฏิบัติงานกับหน่วยความจำระบบปฏิบัติการใช้แคชบัฟเฟอร์เพื่อรับส่งข้อมูลกับจานบันทึกข้อมูล เป็นต้น

การใช้พร็อกซีแคชเซิร์ฟเวอร์ (Proxy Cache Server) สามารถลดเวลาที่ใช้ในการเข้าถึงข้อมูลที่เป็นที่นิยมได้อย่างมาก จากข้อมูลในตารางที่ 2.1 การส่งข้อมูลอินเทอร์เน็ตผ่านมหาสมุทร (Transoceanic Internet Links) มีความล่าช้าของการส่งข้อมูลอยู่ระหว่าง 100-300 มิลลิวินาที แต่เวลาที่ใช้ในการเข้าถึงข้อมูลในหน่วยเก็บข้อมูลภายใน (Local Disk Storage) จะมีความล่าช้าน้อยกว่า 10 มิลลิวินาที และอีกประการหนึ่ง เว็บแคชสามารถลดปริมาณการใช้ปริมาณช่องสัญญาณที่ใช้ติดต่อไปยังอินเทอร์เน็ต โดยการเรียกขอข้อมูลจากที่ถูกเก็บอยู่ภายในเว็บแคช [10]

ตารางที่ 2.1 การเปรียบเทียบเวลาที่ใช้ในการเรียกข้อมูลโดยตรงจากเครื่องบริการข้อมูลกับที่มีการใช้พร็อกซีแคชเซิร์ฟเวอร์ [6]

	0/direct access	1 Squid proxy	1 CERN proxy	2 proxies	3 proxies
LAN measurements					
Duration (cold caches)	240.42	274.10	335.65	359.15	602.78
Degradation	-	14.0%	39.6%	49.4%	150.7%
Duration (warm caches)	240.42	150.99	181.18	240.41	334.49
Speedup	-	37.2%	24.4%	0.0%	-39.1%
WAN measurements					
Duration (cold caches)	2443.66	2616.20	2985.12	2922.54	4048.50
Degradation	-	7.06%	22.16%	19.59%	65.67%
Duration (warm caches)	2443.66	91.55	123.23	140.85	147.89
Speedup	-	2669.21%	1983.00%	1734.94%	1652.35%

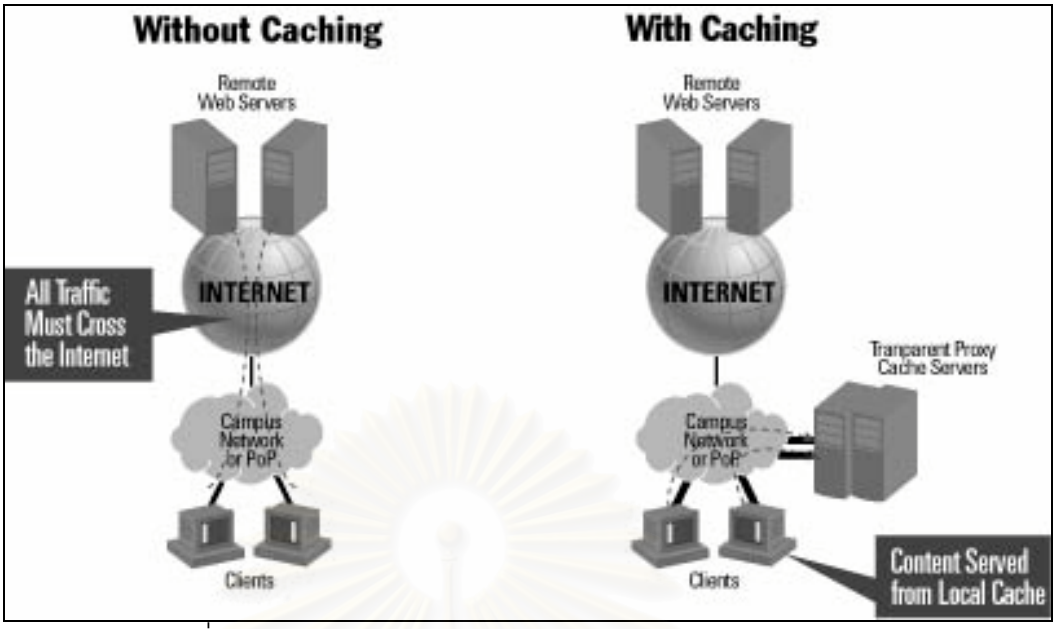
ซึ่งประโยชน์ที่เกิดขึ้น คือ

1. ผู้ใช้งานเว็บจะได้รับข้อมูลจากการเรียกขอ ในเวลาที่สั้นลง
2. ปริมาณการจราจรของข้อมูลที่ออกสู่อินเทอร์เน็ตลดลง

ประเภทของแคช มีดังนี้

1. แคชในเว็บเบราว์เซอร์ (Web Browser) ของผู้ใช้ ซึ่งสามารถใช้ได้เพียงคนเดียว

2. แคชภายในเครือข่าย หรือที่เรียกว่า พร็อกซีแคชเซิร์ฟเวอร์ (Proxy Cache Server) ซึ่งผู้ใช้หลายคน สามารถใช้งานร่วมกันได้ ซึ่งเป็นประเภทที่มีความสำคัญต่อระบบเครือข่าย และเป็นประเภทที่จะทำการวิจัย



ภาพที่ 2.1 การเปรียบเทียบการส่งผ่านข้อมูลในระบบอินเทอร์เน็ต

(ก.) ไม่มีการใช้เว็บแคช (ข.) มีการใช้เว็บแคช

การทำงานของพร็อกซีแคชเซิร์ฟเวอร์

1. รับการเรียกขอข้อมูลจากเครื่องผู้ใช้ต่างๆ ที่เรียกขอไปยังปลายทาง (URL) และเมื่อเกิดการแคชฮิท (Cache Hit) คือข้อมูลที่เรียกขอพบอยู่ภายในแคช พร็อกซีแคชเซิร์ฟเวอร์จะทำการส่งข้อมูลที่มีอยู่ไปยังเครื่องผู้ใช้

2. ในกรณีที่แคชมิส (Cache Miss) คือข้อมูลที่ถูกเรียกขอไม่พบอยู่ภายในแคช พร็อกซีแคชเซิร์ฟเวอร์จะทำการเรียกขอข้อมูลไปยังปลายทาง แล้วนำข้อมูลนั้นส่งต่อไปยังเครื่องผู้ใช้ และทำการเก็บข้อมูลลงในแคชเพื่อการเรียกใช้ในภายหลัง

การวัดประสิทธิภาพของเว็บแคช [10] คำนวณจากค่าฮิทเรโช (Hit Ratio)

$$\text{ฮิทเรโช} = \frac{\text{จำนวนแคชฮิท}}{\text{จำนวนการเรียกขอข้อมูลทั้งหมด}}$$

2.2 ขั้นตอนวิธีการบีบอัดข้อมูล

การบีบอัดข้อมูล (Data Compression) คือ กระบวนการในการลดขนาดของข้อมูลให้มีขนาดเล็กลง โดยการขจัดข้อมูลส่วนที่มีการซ้ำซ้อนออก [11]

วิธีการบีบอัดข้อมูล สามารถแบ่งออกได้เป็น 2 ประเภท คือ

2.2.1 การบีบอัดข้อมูลแบบมีการสูญเสียข้อมูล (Lossy Data Compression)

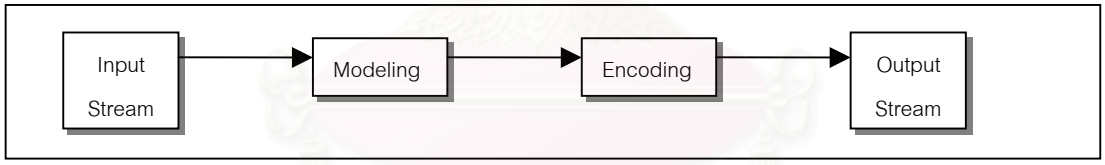
เป็นวิธีการบีบอัดข้อมูลที่ยอมให้ข้อมูลก่อนและหลังผ่านกระบวนการ สามารถมีความผิดเพี้ยนเกิดขึ้นได้ โดยมีการยอมรับความสูญหายที่เกิดขึ้นเพื่อแลกเปลี่ยนกับอัตราการบีบอัดที่เพิ่มขึ้น วิธีนี้มีประสิทธิภาพเมื่อนำมาใช้กับรูปภาพกราฟฟิก หรือเสียงที่ได้จากการแปลงค่าเป็นตัวเลข (Digitized) เนื่องจากธรรมชาติของการใช้การแปลงค่าเป็นตัวเลข เพื่อเป็นการแทนค่าแอนะล็อกซึ่งไม่มีความสมบูรณ์แบบตั้งแต่ตอนเริ่ม ทำให้ข้อมูลเข้าและออกที่ได้เกิดความผิดเพี้ยนขึ้น เราสามารถจะให้มีการผิดเพี้ยนได้มากเท่าที่ต้องการตราบเท่าที่คุณภาพของข้อมูลยังเป็นที่ยอมรับได้ โดยวิธีนี้สามารถปรับเปลี่ยนระดับคุณภาพได้ เมื่อต้องการความถูกต้องสูงขึ้น ประสิทธิภาพในการบีบอัดจะลดลง

2.2.2 การบีบอัดข้อมูลแบบไม่มีการสูญเสียข้อมูล (Lossless Data Compression)

เป็นวิธีการบีบอัดข้อมูลที่รับรองว่าข้อมูลก่อนและหลังผ่านกระบวนการบีบอัดหรือขยายข้อมูลที่ถูกบีบอัดกลับคืนมา จะให้ผลที่ได้ถูกต้องอย่างสมบูรณ์ วิธีนี้ถูกนำมาใช้กับข้อมูลประเภทตัวอักษร ข้อมูลที่อยู่ในฐานข้อมูล หรือแผ่นตารางทำการ (Spreadsheet) เป็นต้น

2.3 หลักการทำงานของการบีบอัดข้อมูลแบบไม่มีการสูญเสียข้อมูล

โดยทั่วไปแล้ว รูปแบบในการบีบอัดข้อมูลที่ใช้อยู่ในปัจจุบัน มีรูปแบบแสดงในภาพที่ 2.2



ภาพที่ 2.2 รูปแบบการทำการบีบอัดข้อมูลแบบไม่มีการสูญเสียข้อมูล

เมื่อโปรแกรมที่ทำการบีบอัดข้อมูล อ่านข้อมูลเข้ามาจะทำการสร้างโครงสร้าง (Model) ข้อมูลนำเข้า ซึ่งส่วนใหญ่จะถูกสร้างโครงสร้างในรูปแบบของความน่าจะเป็นของตัวอักษร (Symbol) ของข้อมูลนำเข้า โดยจะนับความถี่ของแต่ละตัวอักษร เมื่อตัวอักษรถูกอ่านเข้ามาจะถูกนำไปทำการปรับเปลี่ยนโครงสร้าง จากนั้นโครงสร้างข้อมูลที่อยู่ในรูปแบบของความน่าจะเป็นจะถูกนำไปใช้ เพื่อการเข้ารหัส โดยตัวเข้ารหัส (Encoder) และได้ผลลัพธ์เป็นข้อมูลที่ถูกบีบอัดแล้ว

2.4 ขั้นตอนวิธีการบีบอัดข้อมูลที่ใช้ในงานวิจัย

ขั้นตอนวิธีการบีบอัดข้อมูลที่ใช้ในงานวิจัยนี้ เป็นแบบไม่มีการสูญเสียข้อมูล มีดังนี้

2.4.1 แบบจำลองแบบสถิติ (Statistical Modeling) เป็นการสร้างโครงสร้างของข้อมูลโดยการใช้สถิติของการปรากฏของตัวอักษรในข้อมูลทั้งหมด และทำการจัดลำดับของการปรากฏ แล้วจึงทำการแปลงข้อมูลที่ปรากฏมากที่สุดเป็นข้อมูลที่ใช้บิตในการเป็นตัวแทนให้สั้นที่สุด ขั้นตอนวิธีการบีบอัดข้อมูลในกลุ่มนี้ คือ

- Huffman Coding โดยใช้โปรแกรมที่เขียนขึ้นโดย Shaun Case [12] แนวความคิดของขั้นตอนวิธีการนี้ คิดค้นโดย Huffman โดยมีแนวคิดว่า ค่าบิตที่ใช้แทนตัวอักษรที่มีความน่าจะเป็นต่ำจะมีจำนวนบิตมาก และค่าบิตที่ใช้แทนตัวอักษรที่มีความน่าจะเป็นสูงจะมีจำนวนบิตน้อย โดยค่าบิตที่ใช้แทนตัวอักษรจะต่างกันในแต่ละตัวอักษร

หลักการทำงาน คือ จะทำการนับจำนวนที่ปรากฏของแต่ละตัวอักษร แล้วทำการเรียงลำดับตัวอักษรโดยเรียงจากตัวอักษรที่มีจำนวนที่ปรากฏจากมากสุดไปยังตัวอักษรที่มีจำนวนที่ปรากฏน้อยที่สุด ทำการสร้างโครงสร้างต้นไม้โดยรวมจำนวนที่ปรากฏของตัวอักษรในด้านที่ปรากฏน้อยที่สุดคราวละ 2 ตัวอักษร ทำการปรับสมดุลของโครงสร้างต้นไม้ ทำซ้ำจนกระทั่งครบทุกตัวอักษร จะได้โครงสร้างต้นไม้ที่ด้านซ้ายของโครงสร้างต้นไม้จะเป็นตัวอักษรที่ปรากฏมากกว่าตัวอักษรที่อยู่ทางด้านขวาของโครงสร้างต้นไม้

เมื่อเริ่มที่ระดับบนสุดของโครงสร้างข้อมูลแบบต้นไม้ในการเข้ารหัสนี้ และทำการท่องไปในโครงสร้างเพื่อหาตัวอักษรที่จะทำการเข้ารหัส ถ้าไปทางด้านซ้ายจะได้ค่า “0” และไปทางด้านขวาจะได้ค่า “1”

ในการถอดรหัส ตัวถอดรหัสจะต้องรู้จักโครงสร้างข้อมูลแบบต้นไม้ในการเข้ารหัส จึงจะสามารถทำการถอดรหัสข้อมูลได้ถูกต้อง

2.4.2 แบบจำลองที่มีพื้นฐานจากพจนานุกรม (Dictionary Based Modeling) เป็นการสร้างโครงสร้างของข้อมูลโดยการใช้ตารางพจนานุกรมของคำเป็นค่าเปรียบเทียบว่าข้อมูลนั้น จะมีบิตตัวแทนเป็นอย่างไร ขั้นตอนวิธีการบีบอัดข้อมูลในกลุ่มนี้ คือ

- ZIP โดยใช้โปรแกรมที่พัฒนาขึ้นโดยบริษัท Pkware เวอร์ชัน 2.04g [13] แนวความคิดคือ การสร้างโครงสร้างข้อมูลจะทำงาน 2 รอบ

ในรอบแรกจะใช้ขั้นตอนวิธีการ LZ ที่คิดค้นโดย Lempel และ Ziv โดยจะทำการแบ่งข้อมูลเป็น 2 ส่วน คือส่วนแรกจะเป็นส่วนของข้อมูลที่ถูกเข้ารหัสแล้ว หรือเรียกว่าพจนานุกรม (Dictionary) ส่วนที่สองคือข้อมูลที่จะถูกเข้ารหัส ส่วนการทำงานจะทำการอ่านข้อมูล

เก็บเข้าส่วนที่สอง แล้วตรวจสอบว่ามีส่วนของข้อมูลที่ยาวที่สุดที่เข้ากับข้อมูลในส่วนแรก และทำการเข้ารหัสเป็นคู่ของตำแหน่งกับความยาวและอักษรตัวถัดไปตัวแรกที่ไม่เข้ากับข้อมูลในส่วนแรก

ในรอบที่สองจะใช้ขั้นตอนวิธีการของ Huffman

2.4.3 แบบจำลองแบบมีการปรับแต่ง (Adaptive Modeling) เป็นการสร้างโครงสร้างของข้อมูลที่มีความซับซ้อนมากกว่า 2 กลุ่มข้างต้น คือ มีการใช้หลักการสร้างโครงสร้างข้อมูลมากกว่า 1 หลักการขึ้นไป ขั้นตอนวิธีการบีบอัดข้อมูลในกลุ่มนี้ คือ

- LZSS โดยใช้โปรแกรมที่เขียนขึ้นโดย Haruhiko Okumura [14] ซึ่งใช้แนวความคิดของ Ziv และ Lempel และมีการดัดแปลงโดย Storer และ Szymanski โดยมีการประยุกต์ใช้โครงสร้างต้นไม้แบบทวิภาค (Binary Tree) ขั้นตอนวิธีการคือ เก็บบัฟเฟอร์ ซึ่งตั้งต้นด้วยค่าอักษรว่าง (Space) อ่านตัวอักษรจากแฟ้มข้อมูลเข้าสู่บัฟเฟอร์ แล้วทำการค้นหาสายอักขระที่ยาวที่สุดภายในบัฟเฟอร์ ที่เข้ากับ (Match) ตัวอักษรที่เพิ่งทำการอ่านเข้ามา และส่งค่าความยาวและตำแหน่งภายในบัฟเฟอร์ ถ้าขนาดบัฟเฟอร์เท่ากับ 4096 ไบต์ จะสามารถทำการเข้ารหัสของตำแหน่งในจำนวน 12 บิต ถ้าทำการแทนค่าความยาวที่เข้ากันด้วย 4 บิต คู่ของตำแหน่งกับความยาว จะมีขนาด 2 ไบต์ ถ้าค่าที่เข้ากันที่ยาวที่สุดมีขนาดน้อยกว่า 2 ตัวอักษร จะทำการส่งค่าตัวอักษร 1 ตัวไปโดยไม่มีการเข้ารหัส และจะทำการเริ่มกระบวนการนี้กับตัวอักษรถัดไป จะต้องมีการส่งบิตพิเศษจำนวน 1 บิต ไปทุกครั้งเพื่อเป็นการบอกตัวถอดรหัสว่ามีการส่งคู่ของตำแหน่งกับความยาว หรือตัวอักษรที่ไม่ได้ถูกเข้ารหัส

- LZARI โดยใช้โปรแกรมที่เขียนขึ้นโดย Haruhiko Okumura [14] แนวความคิดคือ การสร้างโครงสร้างข้อมูลจะทำงาน 2 รอบ

ในรอบแรกเป็นการใช้ขั้นตอนวิธีการบีบอัดของ LZ

ในรอบที่สองใช้แบบจำลองแบบเรขาคณิต คือใช้หลักการที่ว่าจะทำการเข้ารหัสตัวอักษรที่พบบ่อย ด้วยจำนวนบิตที่น้อย และทำการเข้ารหัสตัวอักษรที่พบน้อย ด้วยจำนวนบิตที่มาก โดยการเรียงลำดับตัวอักษรตามความน่าจะเป็นที่จะพบตัวอักษร แล้วกำหนดช่วงของความน่าจะเป็นให้กับแต่ละตัวอักษร

- LZHUF โดยใช้โปรแกรมที่เขียนขึ้นโดย Haruhiko Okumura [14] แนวความคิดคือ การสร้างโครงสร้างข้อมูลจะทำงาน 2 รอบ

ในรอบแรกเป็นการใช้ขั้นตอนวิธีการบีบอัดของ LZ

ในรอบที่สองใช้แบบจำลอง Huffman ที่มีการปรับแต่ง (Adaptive Huffman) โดย LZHUF จะทำการเข้ารหัส 6 บิตที่สำคัญที่สุดของตำแหน่งภายในบัฟเฟอร์ขนาด 4096 ไบต์ โดยการค้นจากตาราง ตำแหน่งที่เพิ่งมีการอ้างถึงหรือมีแนวโน้มที่จะมีการอ้างถึง จะมีรหัสที่มีจำนวนบิตน้อยกว่า และ 6 บิตที่เหลือจะถูกส่งไปตามเดิม

- LZO โดยใช้โปรแกรมที่เขียนขึ้นโดย Markus Oberhumer [15] แนวความคิดเป็นเช่นเดียวกับ LZ แต่ปรับปรุงในส่วนความเร็วที่ใช้ในการบีบอัดข้อมูลและความเร็วที่ใช้ในการขยายข้อมูล ซึ่งผู้เขียนโปรแกรมไม่ได้ระบุให้ทราบว่ามีการใช้วิธีการใด

2.5 การวัดประสิทธิภาพของการบีบอัดข้อมูล

ประสิทธิภาพของการบีบอัดข้อมูล สามารถวัดได้จากปัจจัย ดังนี้

2.5.1 อัตราการบีบอัด (Compression Ratio) [11]

$$\text{อัตราการบีบอัด} = \frac{\text{ขนาดแฟ้มข้อมูลก่อนทำการบีบอัดข้อมูล}}{\text{ขนาดแฟ้มข้อมูลหลังทำการบีบอัดข้อมูล}}$$

ถ้าอัตราการบีบอัดของขั้นตอนวิธีการบีบอัดใดมาก แสดงว่ามีความสามารถในการลดขนาดของแฟ้มข้อมูลได้มาก

2.5.2 เวลาที่ใช้ในการบีบอัดข้อมูล เป็นเวลาทั้งหมดที่ขั้นตอนวิธีการบีบอัดข้อมูลใช้ในการบีบอัดข้อมูล เวลาที่ใช้น้อยแสดงว่ามีประสิทธิภาพสูงในด้านความเร็ว สามารถบีบอัดได้เร็ว

2.5.3 เวลาที่ใช้ในการขยายข้อมูล เป็นเวลาทั้งหมดที่ขั้นตอนวิธีการบีบอัดข้อมูลใช้ในการขยายข้อมูลกลับสู่ข้อมูลในลักษณะเดิม เวลาที่ใช้น้อยแสดงว่ามีประสิทธิภาพสูงในด้านความเร็ว สามารถขยายข้อมูลได้เร็ว

2.6 การประยุกต์ใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับเว็บแคช

การประยุกต์ใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับเว็บแคช แสดงในภาพที่ 2.3

2.6.1 กรณีที่เกิดแคชฮิท จะเกิดการทำงานตามขั้นตอน ดังนี้

2.6.1.1 เว็บแคชทำการดึงข้อมูลที่ถูกเรียกขอ จากส่วนเก็บข้อมูล

2.6.1.2 ทำการขยายข้อมูล จากที่ถูกบีบอัดกลับสู่ประเภทข้อมูลเดิม

2.6.1.3 ส่งข้อมูลไปยังผู้เรียกขอ

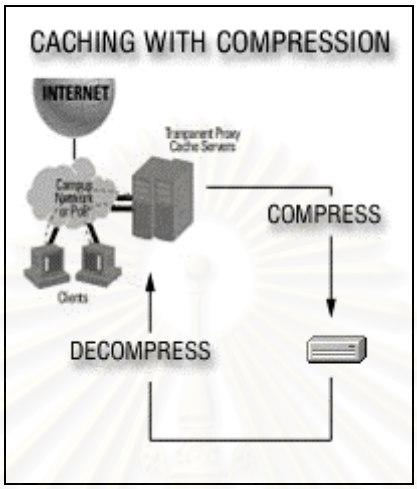
2.6.2 กรณีที่เกิดแคชมิส จะเกิดการทำงานตามขั้นตอนการนำข้อมูลเข้าสู่ส่วนเก็บข้อมูลของพร็อกซีเว็บแคช และทำการบีบอัดข้อมูล จะมีขั้นตอนดังนี้

2.6.2.1 เว็บแคชทำการติดต่อขอข้อมูล จากเครื่องบริการข้อมูล

2.6.2.2 ส่งข้อมูลไปยังผู้เรียกขอ และพิจารณาประเภทของข้อมูลที่จะนำเข้า

2.6.2.3 เลือกใช้ขั้นตอนวิธีที่เหมาะสมที่ได้จากการวิเคราะห์เปรียบเทียบประสิทธิภาพการบีบอัด เพื่อทำการบีบอัดข้อมูล

2.6.2.4 เก็บข้อมูลที่ผ่านกระบวนการบีบอัด ลงสู่ส่วนเก็บข้อมูล



ภาพที่ 2.3 กระบวนการเก็บข้อมูลเข้าและนำข้อมูลออกจากส่วนเก็บข้อมูล

2.7 ปัจจัยที่คำนึงถึงสำหรับการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับเว็บแคช

ปัจจัยที่คำนึงถึงสำหรับการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับเว็บแคช คือ จะพยายามรักษาเวลาในการตอบสนองต่อผู้ใช้ ในกรณีที่เกิดแคชฮิทและแคชมิส ไม่ให้มากกว่าการใช้เว็บแคชแบบปกติที่ไม่มีการใช้ขั้นตอนวิธีการบีบอัดข้อมูล โดยพิจารณาจากความสัมพันธ์

$$\text{User Response Time} = (\text{Hit Ratio} * \text{Hit Cost}) + (\text{Miss Ratio} * \text{Miss Cost})$$

โดย User Response Time คือ เวลาในการตอบสนองต่อผู้ใช้

Hit Ratio คือ ค่าอัตราส่วนในการค้นพบเอกสารภายในเว็บแคช

Hit Cost คือ จำนวนเวลาทั้งหมดที่ใช้ในการเกิดแคชฮิท คือเวลาที่ใช้ในการค้นหาเอกสารในเว็บแคช บวกเวลาที่ใช้ในการนำเอกสารจากจานบันทึกหรือหน่วยความจำ บวกด้วยเวลาที่ใช้ในการส่งเอกสารไปยังผู้ร้องขอ

Miss Ratio คือ ค่าอัตราส่วนในการค้นไม่พบเอกสารภายในเว็บแคช ซึ่งเท่ากับ (1 – Hit Ratio)

Miss Cost คือ จำนวนเวลาทั้งหมดที่ใช้ในการเกิดแคชมิส คือเวลาที่ใช้ในการค้นหาเอกสารในเว็บแคช บวกเวลาที่ใช้ในการขอเอกสารจากเซิร์ฟเวอร์ บวกด้วยเวลาที่ใช้ในการส่งเอกสารไปยังผู้ร้องขอ

เมื่อพิจารณาจากความสัมพันธ์ดังกล่าว

- ในกรณีแคชฮิท เวลาในการตอบสนองต่อผู้ใช้จะขึ้นกับเวลาที่ใช้ในการค้นหาเอกสารในแคชและเวลาที่ใช้ในการนำเอกสารจากจานบันทึกหรือหน่วยความจำ
- ในกรณีแคชมิส เวลาในการตอบสนองต่อผู้ใช้จะขึ้นกับเวลาที่ใช้ในการขอเอกสารจากเซิร์ฟเวอร์

ดังนั้นในการที่จะรักษาเวลาในการตอบสนองต่อผู้ใช้เมื่อมีการใช้วิธีการบีบอัดข้อมูลให้เวลาในการตอบสนองไม่มากไปกว่าการใช้เว็บแคชปกติจึงต้องใช้ปัจจัยต่อไปนี้เข้ามาประกอบด้วย คือ

2.7.1 อัตราการบีบอัดข้อมูล ถ้ามีอัตราการบีบอัดสูงจะสามารถเก็บข้อมูลลงในส่วนเก็บข้อมูลได้มากขึ้น ทำให้เกิดฮิทเรโชสูงขึ้น

2.7.2 เวลาที่ใช้ในการบีบอัดข้อมูล ถ้าเวลาที่ใช้ในการบีบอัดมากจะทำให้พร็อกซีแคชเซิร์ฟเวอร์มีประสิทธิภาพในการทำงานลดลง

2.7.3 เวลาที่ใช้ในการขยายข้อมูล ถ้าเวลาที่ใช้ในการขยายข้อมูลมาก จะทำให้เวลาที่ใช้ในการนำข้อมูลออกจากจานบันทึกเพื่อเตรียมส่งไปยังผู้เรียกขอมากขึ้น จะส่งผลให้เวลาในการตอบสนองต่อผู้ใช้สูงขึ้น

2.8 แนวคิดในการทดสอบ

ในงานวิจัยนี้ มีแนวทางในการทดสอบ ดังนี้ ทดสอบอัตราการบีบอัดข้อมูล เวลาที่ใช้ในการบีบอัดข้อมูล และเวลาที่ใช้ในการขยายข้อมูล ทำโดยการคัดเลือกกลุ่มตัวอย่างของข้อมูลประเภทต่าง ๆ ที่มักปรากฏภายในเว็บแคช มาทำการบีบอัดข้อมูล และขยายข้อมูล โดยใช้ขั้นตอนวิธีต่าง ๆ กัน แล้วทำการวัดอัตราการบีบอัดข้อมูล เวลาที่ใช้ในการบีบอัดข้อมูล และเวลาที่ใช้ในการขยายข้อมูล ทำการจัดอันดับของขั้นตอนวิธีในการบีบอัดข้อมูล

บทที่ 3
การวิเคราะห์สภาพการใช้เว็บ

ในบทนี้ จะเป็นการวิเคราะห์สภาพการใช้เว็บของ จุฬาลงกรณ์มหาวิทยาลัย เพื่อศึกษาถึงลักษณะการใช้งานอินเทอร์เน็ตของนิสิต อาจารย์ และบุคลากรของจุฬาลงกรณ์มหาวิทยาลัย ว่าเว็บแคชมีการทำการเรียกขอข้อมูลจากอินเทอร์เน็ต ในลักษณะข้อมูลประเภทใด ขนาดข้อมูลที่เรียกขอมีขนาดเท่าใด ซึ่งจะทำให้ทราบถึงว่าขั้นตอนการบีบอัดข้อมูลจะต้องมีความสามารถในการจัดการบีบอัดกับข้อมูลประเภทใด และขนาดข้อมูลเท่าใด จึงจะมีประสิทธิภาพในการบีบอัดสูงที่สุด

การวิเคราะห์สภาพการใช้เว็บ ทำโดยการวิเคราะห์จากสถิติการใช้งานอินเทอร์เน็ตในจุฬาลงกรณ์มหาวิทยาลัย ที่ถูกบันทึกอยู่ในแฟ้มข้อมูลบันทึกการใช้งาน (access.log) ของสควิดเว็บแคช โดยช่วงเวลาของข้อมูลที่นำมาทำการวิเคราะห์อยู่ในช่วงระหว่างวันที่ 3 ตุลาคม พ.ศ. 2542 ถึงวันที่ 1 พฤศจิกายน พ.ศ. 2542 ซึ่งรวบรวมโดยสำนักเทคโนโลยีสารสนเทศ จุฬาลงกรณ์มหาวิทยาลัย

3.1 แฟ้มบันทึกการใช้งานของโปรแกรมสควิดเว็บแคช (Access.log)

แฟ้มบันทึกการใช้งานของโปรแกรมสควิดเว็บแคช [16] เวอร์ชัน 2 ขึ้นไป มีการเก็บข้อมูลการใช้งานอินเทอร์เน็ตต่างๆ ไว้ โดยจะเก็บบันทึกการใช้งานเป็นหนึ่งการใช้งานต่อหนึ่งบรรทัด มีรูปแบบดังนี้

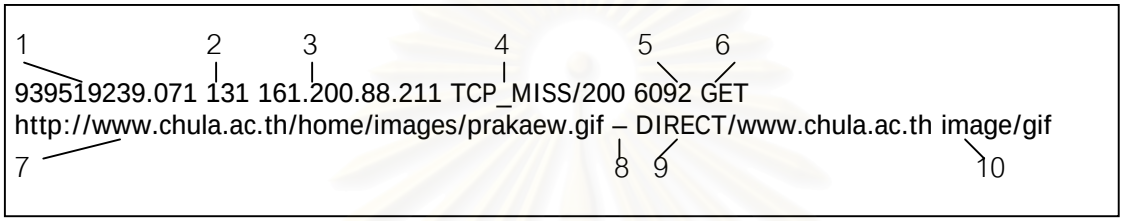
time elapsed remotehost code/status bytes method URL rfc931 peerstatus/peerhost type

โดยมีคำอธิบายดังนี้

1. time	เวลาที่บันทึกการเรียกขอ
2. elapsed	เวลาที่ใช้ในการรับข้อมูลจากเว็บเซิร์ฟเวอร์ หน่วยเป็นมิลลิวินาที
3. remotehost	เลขไอพีของเครื่องลูกข่ายที่เรียกขอ
4. code/status	รหัสการตอบกลับจากเว็บเซิร์ฟเวอร์
5. bytes	ปริมาณข้อมูลที่รับจากเว็บเซิร์ฟเวอร์ หน่วยเป็นไบต์
6. method	วิธีการเรียกขอข้อมูล

7. URL	ยูอาร์แอล
8. rfc931	ชื่อของผู้เรียกขอ โดยปกติจะไม่มีการเก็บข้อมูลส่วนนี้
9. peerstatus/peerhost	วิธีการส่งต่อการเรียกขอ/ไอพีปลายทางที่การเรียกขอถูกส่งต่อไป
10. type	ประเภทของข้อมูล

ตัวอย่างบันทึกการเรียกขอ



3.2 ขั้นตอนการวิเคราะห์สภาพการใช้เว็บ

การวิเคราะห์สภาพการใช้เว็บ ทำโดยการใช้โปรแกรมวิเคราะห์บันทึกการใช้งานสควิดเว็บแคช ชื่อ คาลามาริส (Calamaris) [17] โดยโปรแกรมคาลามาริสเป็นโปรแกรมที่ใช้ในการวิเคราะห์ข้อมูลบันทึกการใช้งานของสควิดเว็บแคช

โดยมีรูปแแบบการใช้งาน ดังนี้

```
$./calamaris –a –w access.log > result.html
```

โดย access.log คือแฟ้มข้อมูลบันทึกการใช้งาน

result.html คือผลการวิเคราะห์ในรูปแบบเอกสารเอชทีเอ็มแอล

ข้อมูลทั้งหมดที่ได้จากการวิเคราะห์โดยโปรแกรมคาลามาริส แสดงในภาคผนวก ก. แต่ข้อมูลที่สนใจในการวิเคราะห์นี้ คือ ประเภทของข้อมูลและสัดส่วนของประเภทข้อมูลที่มีการเรียกขอ เนื่องจากต้องการทราบถึงลักษณะการใช้งานในจุฬาลงกรณ์มหาวิทยาลัย ว่ามีการเรียกใช้ข้อมูลประเภทใดเป็นสัดส่วนเท่าใด

3.3 ผลการวิเคราะห์สภาพการใช้เว็บ

ผลการวิเคราะห์สภาพการใช้งานโดยโปรแกรมคาลามาริส แสดงในภาพที่ 3.1

Requested content-type

content-type	request	%	kByte	%	hit-%
image/gif	2228159	49.63	7084085	24.00	74.00
text/html	883947	19.69	5937614	20.11	16.28
image/jpeg	547702	12.20	5483615	18.57	49.75
<unknown>	381065	8.49	77280	0.26	20.58
<error>	297737	6.63	445073	1.51	8.29
text/plain	42317	0.94	916817	3.11	13.00
application/x-javascript	29975	0.67	58231	0.20	31.22
text/css	18101	0.40	22640	0.08	69.98
application/octet-stream	16594	0.37	3845962	13.03	51.27
<secure>	13728	0.31	35930	0.12	0.00
application/zip	5396	0.12	1634697	5.54	10.04
audio/midi	3067	0.07	60179	0.20	49.43
audio/x-midi	1780	0.04	19842	0.07	38.82
audio/x-pn-realaudio	1643	0.04	214948	0.73	43.64
image/pjpeg	1617	0.04	45942	0.16	11.69
application/x-shockwave-flash	1541	0.03	73659	0.25	51.78
application/pdf	1182	0.03	451135	1.53	11.34
image/x-xbitmap	1107	0.02	1118	0.00	0.27
application/postscript	1016	0.02	581166	1.97	6.20
application/cache-digest	908	0.02	147216	0.50	100.00
other: 190 content-types	10992	0.24	2385968	8.08	29.18
Sum	4489574	100.00	29523128	100.00	49.30

ภาพที่ 3.1 ผลการวิเคราะห์สภาพการใช้งาน โดยโปรแกรมคาลามาริสของข้อมูลระหว่างวันที่ 3 ต.ค. ถึง 1 พ.ย. 2542 จากสำนักเทคโนโลยีสารสนเทศ จุฬาลงกรณ์มหาวิทยาลัย

จากภาพที่ 3.1 แสดงถึงการใช้งานอินเทอร์เน็ตในจุฬาลงกรณ์มหาวิทยาลัย จำนวน 4,489,574 การเรียกใช้ ซึ่งในช่วงวันที่ 3 ตุลาคม ถึงวันที่ 1 พฤศจิกายน พ.ศ. 2542 เป็นช่วงเวลาการศึกษาภาคการศึกษาปลายของปีการศึกษา 2542 ซึ่งสามารถใช้เป็นตัวแทนของข้อมูลการใช้งานของจุฬาลงกรณ์มหาวิทยาลัยในช่วงภาคการศึกษาได้

เมื่อพิจารณาถึงผลการวิเคราะห์ในส่วนของจำนวนการเรียกใช้ข้อมูลโดยจำแนกตามประเภทข้อมูล แสดงให้เห็นว่าข้อมูลประเภทที่มีการเรียกขอมากที่สุด คือ รูปภาพแบบจิฟ 49.63% ตัวอักษรแบบเอชทีเอ็มแอล 19.69% รูปภาพแบบเจเพก 12.20% ส่วนที่เหลือที่สามารถระบุประเภทข้อมูลได้มีอยู่ 6.04% ไม่ทราบประเภท 8.49%

เมื่อพิจารณาถึงผลการวิเคราะห์ในส่วนของจำนวนไบต์ข้อมูลที่มีการเรียกใช้ข้อมูลโดยจำแนกตามประเภทข้อมูล แสดงให้เห็นว่าข้อมูลประเภทที่มีการเรียกขอมากที่สุด คือ รูปภาพแบบจิฟ 24.0% ตัวอักษรแบบเอชทีเอ็มแอล 20.11% รูปภาพแบบเจเพก 18.57% ส่วนที่เหลือที่สามารถระบุประเภทข้อมูลได้มีอยู่ 35.45% ไม่ทราบประเภท 0.26%

จะเห็นได้ว่าประเภทข้อมูลที่มีการเรียกขอมากที่สุด สามารถแบ่งออกได้เป็น

1. รูปภาพแบบจิฟ
2. ตัวอักษรแบบเอชทีเอ็มแอล
3. รูปภาพแบบเจเพก
4. ประเภทอื่นๆ

แต่เนื่องจากข้อมูลประเภทอื่นๆ นั้นประกอบด้วยประเภทข้อมูลที่ต่างกันหลายๆ ประเภท จึงไม่สามารถทำการวิเคราะห์ได้ทั้งหมด จึงเลือกเฉพาะประเภทข้อมูลอันดับที่ 1 ถึงอันดับที่ 3 มาทำการศึกษารูปแบบแฟ้มข้อมูล (File Format) และจากการศึกษาถึงรูปแบบแฟ้มข้อมูลของรูปภาพแบบจิฟพบว่ามีการนำขั้นตอนวิธีการบีบอัดข้อมูลแบบ LZW [11] มาประยุกต์ใช้ในการเก็บข้อมูลแล้ว จึงไม่สามารถทำการบีบอัดข้อมูลได้เพิ่มเติมอีก

ดังนั้น ประเภทข้อมูลที่สามารถทำการใช้การบีบอัดข้อมูล ได้มีดังนี้

1. ข้อมูลตัวอักษร ซึ่งในที่นี้จะรวมถึงเอกสารเอชทีเอ็มแอล (HTML) ข้อความธรรมดา (Plain Text) เอกสารรูปแบบซีเอสเอส (CSS) และเอกสารที่มีการระบุประเภทเป็นตัวอักษรอื่นๆ ทั้งหมด

2. ข้อมูลรูปภาพเจเพก

3.4 การแบ่งกลุ่มข้อมูล

เมื่อได้ประเภทข้อมูลที่จะทำการวิเคราะห์แล้ว ขั้นตอนต่อมาคือการพิจารณาถึงการกระจายของขนาดข้อมูล เพื่อให้ทราบถึงลักษณะข้อมูลที่เรียกขอว่าส่วนใหญ่มีขนาดแฟ้มข้อมูลเท่าใด เนื่องจากมีการตั้งสมมติฐานเบื้องต้นว่าในขั้นตอนวิธีการบีบอัดข้อมูลหนึ่งจะมีความสามารถในการบีบอัดข้อมูลต่อประเภทข้อมูล และขนาดข้อมูลที่ไม่เท่ากัน จึงต้องมีการวิเคราะห์หาประสิทธิภาพการบีบอัดข้อมูลกับข้อมูลในช่วงขนาดข้อมูลที่ต่างกัน จึงมีการทำการแบ่งข้อมูลออกเป็นส่วนๆ ตามขนาดของแฟ้มข้อมูล โดยในขั้นตอนนี้ได้ทำการสุ่มเลือกแฟ้มข้อมูลบันทึกการใช้งานมา ได้แฟ้มข้อมูลบันทึกการใช้งานของวันที่ 20 ตุลาคม พ.ศ. 2542 เพื่อใช้ในการวิเคราะห์ข้อมูลขั้นต่อไป

3.4.1 การแบ่งกลุ่มข้อมูลของข้อมูลประเภทตัวอักษร

ขั้นตอนที่ใช้ในการแบ่งกลุ่มข้อมูลของข้อมูลประเภทตัวอักษร มีดังนี้

3.4.1.1 คัดเลือกข้อมูลที่มีการระบุประเภทเป็น “text” จากแฟ้มข้อมูลบันทึกการใช้งาน โดยการใช้โปรแกรม awk โดยมีรูปแแบบคำสั่งดังนี้

```
$awk '$10 ~ "text" {print}' access.log > text.log
```

โดย $10 คือ ข้อมูลในคอลัมน์ที่ 10 ของแฟ้มข้อมูลบันทึกการใช้งาน ซึ่งเป็นตัวระบุประเภทข้อมูล

ผลที่ได้คือ แฟ้ม text.log ซึ่งจะมีข้อมูลเฉพาะประเภทตัวอักษรเท่านั้น

3.4.1.2 คัดเลือกข้อมูลเฉพาะที่มีรหัสการตอบกลับที่มีการติดต่อสมบูรณ์ (200) เนื่องจากเป็นรหัสการตอบกลับที่จะระบุขนาดของแฟ้มข้อมูลที่ถูกต้อง โดยการใช้โปรแกรม awk โดยมีรูปแบบคำสั่งดังนี้

```
$awk '$4 ~ "200" {print}' cuttext.log > text200.log
```

โดย $4 คือ ข้อมูลในคอลัมน์ที่ 4 ของแฟ้มข้อมูลบันทึกการใช้งาน ซึ่งเป็นตัวระบุรหัสการตอบกลับ

ผลที่ได้คือ แฟ้ม text200.log ซึ่งจะมีข้อมูลเฉพาะประเภทตัวอักษรที่มีรหัสการตอบกลับที่มีการติดต่อสมบูรณ์เท่านั้น

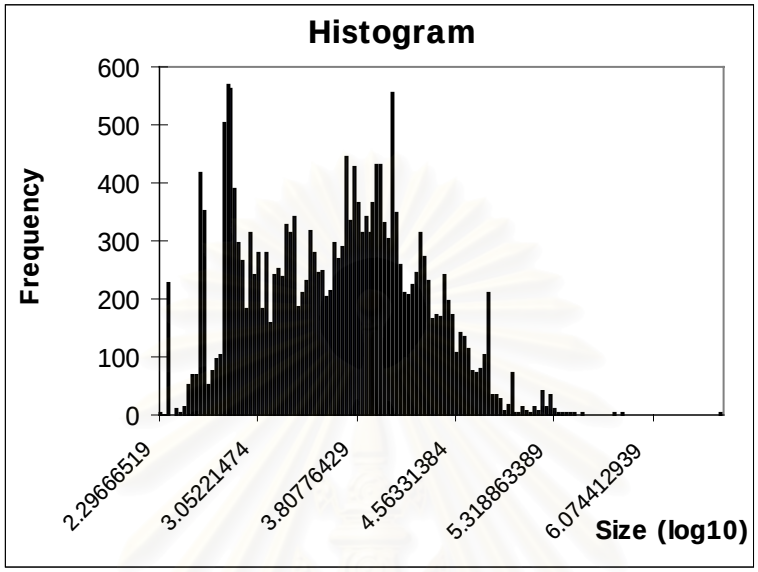
3.4.1.3 คัดเลือกข้อมูลเฉพาะที่สามารถเก็บลงในเว็บแคชได้ โดยการคัดยูอาร์แอลที่มีส่วนประกอบด้วย “cgi” หรือ “?” ออก โดยการใช้โปรแกรม awk โดยมีรูปแแบบคำสั่งดังนี้

```
$awk '$7 !~ "cgi | ?" {print}' text200.log > text200cache.log
```

โดย $7 คือ ข้อมูลในคอลัมน์ที่ 7 ของแฟ้มข้อมูลบันทึกการใช้งาน ซึ่งเป็นตัวระบุยูอาร์แอล

ผลที่ได้คือ แฟ้ม text200cache.log ซึ่งจะมีข้อมูลเฉพาะประเภทตัวอักษรที่มีรหัสการตอบกลับที่มีการติดต่อสมบูรณ์และเป็นข้อมูลที่สามารถเก็บอยู่ในเว็บแคชได้เท่านั้น

3.4.1.4 นำข้อมูลที่ได้ในส่วนของขนาดแฟ้มข้อมูล หน่วยเป็นไบต์ มาทำการวัดการกระจายข้อมูล ดังแสดงในภาพที่ 3.2



ภาพที่ 3.2 การกระจายของขนาดข้อมูลประเภทตัวอักษร

จากภาพที่ 3.2 ซึ่งแสดงการกระจายของขนาดข้อมูลประเภทตัวอักษร พบว่ามีการกระจายตัวหนาแน่นในช่วงขนาดข้อมูล 700 ไบต์ ถึง 100,000 ไบต์ ส่วนข้อมูลที่มีขนาดที่ใหญ่กว่า 100,000 ไบต์ จะมีจำนวนน้อยมาก การที่จะแบ่งข้อมูลออกเป็นกลุ่มๆ โดยที่ในแต่ละกลุ่มมีข้อมูลจำนวนเท่าๆ กัน จึงใช้ค่าเปอร์เซ็นไตล์ เป็นเกณฑ์ในการแบ่งกลุ่มข้อมูลออกเป็นส่วนๆ โดยในการทดลองนี้ใช้ค่าเปอร์เซ็นไตล์ที่ 0-12.5, 12.5-25, 25-37.5, 37.5-50, 50-62.5, 62.5-75, 75-87.5 และ 87.5-100 โดยการแบ่งกลุ่มโดยใช้ค่าเปอร์เซ็นไตล์จะทำให้ได้กลุ่มของข้อมูล 8 กลุ่มข้อมูล ได้ผลตามตารางที่ 3.1

ตารางที่ 3.1 ผลการแบ่งกลุ่มข้อมูลโดยใช้ค่าเปอร์เซ็นไตล์ของขนาดแฟ้มข้อมูลของข้อมูลประเภทตัวอักษร

ช่วงเปอร์เซ็นไตล์	0-12.5	12.5-25	25-37.5	37.5-50	50-62.5	62.5-75	75-87.5	87.5-100
กลุ่มข้อมูลที่	1	2	3	4	5	6	7	8
จำนวนข้อมูล	2889	2861	2866	2862	2864	2862	2871	2829
ขนาดแฟ้มข้อมูลที่เล็กที่สุด (ไบต์)	1	640	1148	2071	4413	7121	11423	21190
ขนาดแฟ้มข้อมูลที่ใหญ่ที่สุด (ไบต์)	639	1147	2070	4412	7120	11422	21188	655743

3.4.2 การแบ่งกลุ่มข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ขั้นตอนที่ใช้ในการแบ่งกลุ่มข้อมูลของข้อมูลประเภทรูปภาพเจเพก มีดังนี้

3.4.2.1 คัดเลือกข้อมูลที่มีการระบุประเภทเป็น “image/jpeg” จากแฟ้มข้อมูลบันทึกการใช้ งาน โดยการใช้โปรแกรม awk โดยมีรูปแแบบคำสั่งดังนี้

```
$awk '$10 ~ "image/jpeg" {print}' access.log > jpg.log
```

โดย $10 คือ ข้อมูลในคอลัมน์ที่ 10 ของแฟ้มข้อมูลบันทึกการใช้งาน ซึ่งเป็นตัวระบุประเภทข้อมูล

ผลที่ได้ คือ แฟ้ม jpg.log ซึ่งจะมีข้อมูลเฉพาะประเภทรูปภาพเจเพกเท่านั้น

3.4.2.2 คัดเลือกข้อมูลเฉพาะที่มีรหัสการตอบกลับที่มีการติดต่อสมบูรณ์ (200) เนื่องจากเป็นรหัสการตอบกลับที่จะระบุขนาดของแฟ้มข้อมูลที่ถูกต้อง โดยการใช้โปรแกรม awk โดยมีรูปแบบคำสั่งดังนี้

```
$awk '$4 ~ "200" {print}' jpg.log > jpg200.log
```

โดย $4 คือ ข้อมูลในคอลัมน์ที่ 4 ของแฟ้มข้อมูลบันทึกการใช้งาน ซึ่งเป็นตัวระบุรหัสการตอบกลับ

ผลที่ได้ คือ แฟ้ม jpg200.log ซึ่งจะมีข้อมูลเฉพาะประเภทรูปภาพเจเพกที่มีรหัสการตอบกลับที่มีการติดต่อสมบูรณ์เท่านั้น

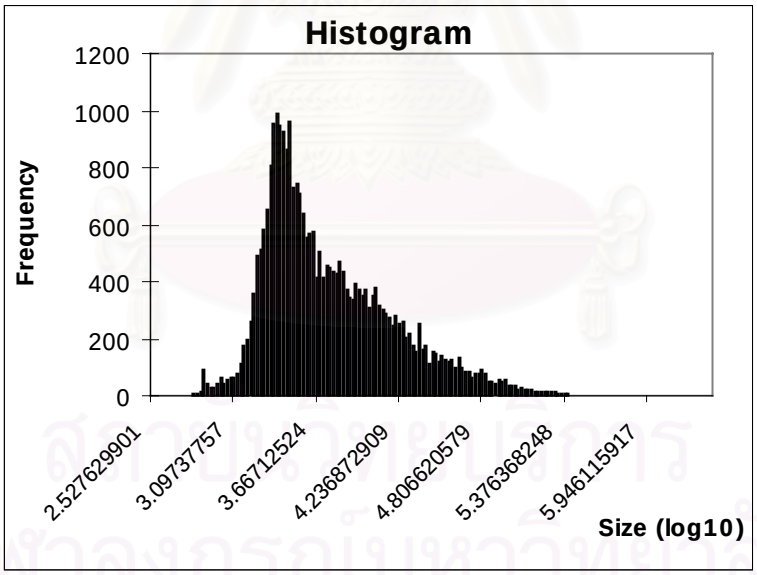
3.4.2.3 คัดเลือกข้อมูลเฉพาะที่สามารถเก็บลงในเว็บแคชได้ โดยการคัดยูอาร์แอลที่มีส่วนประกอบด้วย “cgi” หรือ “?” ออก โดยการใช้โปรแกรม awk โดยมีรูปแบบคำสั่งดังนี้

```
$awk '$7 !~ "cgi | ?" {print}' jpg200.log > jpg200cache.log
```

โดย $7 คือ ข้อมูลในคอลัมน์ที่ 7 ของแฟ้มข้อมูลบันทึกการใช้งาน ซึ่งเป็นตัวระบุยูอาร์แอล

ผลที่ได้คือ แฟ้ม jpg200cache.log ซึ่งจะมีข้อมูลเฉพาะประเภทรูปภาพเจเพกที่มีรหัสการตอบกลับที่มีการติดต่อสมบูรณ์และเป็นข้อมูลที่สามารถเก็บอยู่ในเว็บแคชได้เท่านั้น

3.4.2.4 นำข้อมูลที่ได้ในส่วนของขนาดแฟ้มข้อมูล หน่วยเป็นไบต์ มาทำการวัดการกระจายข้อมูล ดังแสดงในภาพที่ 3.3



ภาพที่ 3.3 การกระจายของขนาดข้อมูลประเภทรูปภาพเจเพก

จากภาพที่ 3.3 ซึ่งแสดงการกระจายของขนาดข้อมูลประเภทรูปภาพเจเพก พบว่ามีการกระจายตัวหนาแน่นในช่วงขนาดข้อมูล 3,000 ไบต์ ถึง 100,000 ไบต์ ส่วนข้อมูลที่มีขนาดที่ใหญ่กว่า 100,000 ไบต์จะมีจำนวนน้อยมาก การที่จะแบ่งข้อมูลออกเป็นกลุ่มๆ โดยที่ในแต่ละกลุ่มมีข้อมูล

จำนวนเท่าๆ กัน จึงใช้ค่าเปอร์เซ็นไตล์ เป็นเกณฑ์ในการแบ่งกลุ่มข้อมูลออกเป็นส่วนๆ โดยในการทดลองนี้ใช้ค่าเปอร์เซ็นไตล์ที่ 0-12.5, 12.5-25, 25-37.5, 37.5-50, 50-62.5, 62.5-75, 75-87.5 และ 87.5-100 โดยการแบ่งกลุ่มโดยใช้ค่าเปอร์เซ็นไตล์จะทำให้ได้กลุ่มของข้อมูล 8 กลุ่มข้อมูล ได้ผลตามตารางที่ 3.2

ตารางที่ 3.2 ผลการแบ่งกลุ่มข้อมูลโดยใช้ค่าเปอร์เซ็นไตล์ของขนาดแฟ้มข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ช่วงเปอร์เซ็นไตล์	0-12.5	12.5-25	25-37.5	37.5-50	50-62.5	62.5-75	75-87.5	87.5-100
กลุ่มข้อมูลที่	1	2	3	4	5	6	7	8
จำนวนข้อมูล	3687	3656	3659	3658	3653	3654	3655	3627
ขนาดแฟ้มข้อมูลที่เล็กที่สุด (ไบต์)	1	2055	2530	3140	4206	6346	10309	19929
ขนาดแฟ้มข้อมูลที่ใหญ่ที่สุด (ไบต์)	2054	2529	3139	4205	6344	10306	19925	2394000

3.5 การกำหนดขนาดตัวอย่างโดยวิธีการทางสถิติโดยใช้ค่าเฉลี่ย

การกำหนดขนาดตัวอย่างโดยวิธีการทางสถิติโดยใช้ค่าเฉลี่ย [18] มีขั้นตอนดังต่อไปนี้

3.5.1 กำหนดระดับความคลาดเคลื่อน (D) ในกรณีใช้ค่า 5% ของค่าเฉลี่ย ซึ่งหมายถึงต้องการให้ขนาดของแฟ้มข้อมูลที่เลือกมามีความถูกต้องในช่วงผิดพลาดไม่เกิน 5% ของค่าเฉลี่ย

3.5.2 กำหนดระดับความเชื่อมั่น กำหนดที่ 95%

3.5.3 หา Z ที่สอดคล้องกับระดับความเชื่อมั่นที่ต้องการ ที่ระดับความเชื่อมั่น 95% ความน่าจะเป็นที่ค่าเฉลี่ยจะอยู่นอกช่วง คือ 0.05/2 = 0.025 เมื่อนำไปหาค่าจากตาราง Z จะได้เท่ากับ 1.96

3.5.4 คำนวณค่าส่วนเบี่ยงเบนมาตรฐานของประชากร (Standard Deviation: SD)

3.5.5 คำนวณขนาดตัวอย่าง ได้จากสมการ

$$n = \left(\frac{SD * Z}{D}\right)^2$$

เมื่อการสุ่มตัวอย่างเป็นแบบไม่ใส่คืน ขนาดตัวอย่าง คือ

$$nc = \frac{nN}{N + n - 1}$$

ตัวอย่างเช่น การคำนวณขนาดตัวอย่างของข้อมูลประเภทตัวอักษร ในกลุ่มที่ 1 ที่เปอร์เซ็นไตล์ที่ 0 – 12.5

จำนวนข้อมูล	2889 ข้อมูล
ค่าเบี่ยงเบนมาตรฐาน	145.3994
ค่าเฉลี่ย	467.1381
ค่าระดับความคลาดเคลื่อน (5% ของค่าเฉลี่ย)	23.35691
ค่า Z จากตาราง	1.96

จำนวนตัวอย่าง

$$n = \left(\frac{145.3994 * 1.96}{23.35691}\right)^2$$

= 148.8698

จำนวนตัวอย่าง

$$n_c = \frac{148.8698 * 2889}{2{,}889 + 148.8698 - 1}$$

= 142

นั่นคือการที่จะสุ่มตัวอย่างจากข้อมูล 2,889 ข้อมูล เพื่อให้ขนาดของแฟ้มข้อมูลที่สุ่มได้มีความผิดพลาดไม่เกิน 5% ของค่าเฉลี่ยที่ระดับความเชื่อมั่น 95% จะต้องใช้ตัวอย่างจำนวน 142 ตัวอย่าง

ผลการคำนวณหาขนาดกลุ่มตัวอย่างของข้อมูลประเภทตัวอักษร แสดงในตารางที่ 3.3

ตารางที่ 3.3 ผลการคำนวณจำนวนกลุ่มตัวอย่างของข้อมูลประเภทตัวอักษร

ช่วงเปอร์เซ็นไตล์	0-12.5	12.5-25	25-37.5	37.5-50	50-62.5	62.5-75	75-87.5	87.5-100
กลุ่มข้อมูลที่	1	2	3	4	5	6	7	8
จำนวนข้อมูล	2889	2861	2866	2862	2864	2862	2871	2829
ค่าเบี่ยงเบนมาตรฐาน	145.3994	151.2532	288.8145	678.4618	796.0331	1193.812	3025.934	45675.52
ค่าเฉลี่ย	467.1381	834.2716	1516.424	3115.577	5789.086	9010.089	15483.81	49112.72
ค่าความแม่นยำ	23.35691	41.71358	75.82118	155.7789	289.4543	450.5045	774.1906	2455.636
ขนาดตัวอย่าง n	148.8698	50.50871	55.74025	72.86951	29.05457	26.97653	58.6861	1329.08
ขนาดตัวอย่าง nc	141.6211	49.64954	54.69557	71.08447	28.77273	26.73389	57.53017	904.4731
ขนาดตัวอย่างเมื่อมีการปัดเศษ	142	50	55	72	29	27	58	905

จากการคำนวณหากลุ่มขนาดตัวอย่างของข้อมูลประเภทตัวอักษร จากข้อมูลทั้งหมด 22,904 ต้องใช้ขนาดตัวอย่างจำนวน 1,338 ข้อมูล ซึ่งคิดเป็น 5.8% ของข้อมูลทั้งหมด โดยกลุ่มตัวอย่างสำหรับข้อมูลในกลุ่มที่ 8 มีจำนวนมากเนื่องจากข้อมูลในกลุ่มนี้มีความแปรปรวนมาก เพราะขนาดข้อมูลที่มีขนาดใหญ่มีการเรียกขอน้อย

ผลการคำนวณหาขนาดกลุ่มตัวอย่างของข้อมูลประเภทรูปภาพเจเพก แสดงในตารางที่ 3.4

จากการคำนวณหากลุ่มขนาดตัวอย่างของข้อมูลประเภทรูปภาพเจเพก จากข้อมูลทั้งหมด 29,249 ต้องใช้ขนาดตัวอย่างจำนวน 1,434 ข้อมูล ซึ่งคิดเป็น 5% ของข้อมูลทั้งหมด โดยกลุ่มตัวอย่างสำหรับข้อมูลในกลุ่มที่ 8 มีจำนวนมากเนื่องจากข้อมูลในกลุ่มนี้มีความแปรปรวนมาก เพราะขนาดข้อมูลที่มีขนาดใหญ่มีการเรียกขอน้อย

ในการวัดการกระจายของขนาดข้อมูล และการแบ่งกลุ่มข้อมูล ไม่มีการคัดยูอาร์แอลที่ซ้ำกันทิ้ง เนื่องจากในการวิเคราะห์ส่วนนี้ต้องการทราบถึงลักษณะการกระจายของข้อมูลที่มีการเรียกใช้งานทั้งหมดที่แท้จริง เพื่อนำมาใช้พิจารณาว่าในขนาดข้อมูลใดจะต้องมีการทำการขยายข้อมูลบ่อยเพียงใด

ตารางที่ 3.4 ผลการคำนวณจำนวนกลุ่มตัวอย่างของข้อมูลประเภทรูปภาพเจเพก

ช่วงเปอร์เซ็นไตล์	0-12.5	12.5-25	25-37.5	37.5-50	50-62.5	62.5-75	75-87.5	87.5-100
กลุ่มข้อมูลที่	1	2	3	4	5	6	7	8
จำนวนข้อมูล	3687	3656	3659	3658	3653	3654	3655	3627
ค่าเบี่ยงเบนมาตรฐาน	359.6307	134.3288	172.2177	307.0048	623.4704	1167.088	2682.308	54137.2491
ค่าเฉลี่ย	1617.239	2299.485	2816.771	3612.85	5172.704	8116.571	14139.43	49372.6661
ค่าความแม่นยำ	80.86196	114.9742	140.8385	180.6425	258.6352	405.8286	706.9714	2468.63331
ขนาดตัวอย่าง n	75.98662	5.243838	5.744138	11.0959	22.32382	31.77123	55.30006	1847.52933
ขนาดตัวอย่าง nc	74.472	5.237758	5.7367	11.06536	22.19427	31.50591	54.49053	1224.25376
ขนาดตัวอย่างเมื่อมีการปัดเศษ	75	6	6	12	23	32	55	1225

3.6 การสุ่มตัวอย่างข้อมูล

เมื่อได้จำนวนข้อมูลตัวอย่างที่ต้องการใช้เป็นตัวแทนของข้อมูลในแต่ละกลุ่มของข้อมูล นำมาทำการสุ่มตัวอย่างแบบไม่ซ้ำ โดยสุ่มจากแฟ้มข้อมูลบันทึกการใช้งานเพื่อหายูอาร์แอลที่ต้องการ ในกรณีที่เกิดการซ้ำกันของยูอาร์แอล จะทำการคัดค่ายูอาร์แอลที่ซ้ำทิ้ง และทำการสุ่มให้ได้ข้อมูลครบตามจำนวนข้อมูลตัวอย่างที่คำนวณไว้

การสุ่มตัวอย่างในขั้นตอนนี้เป็นการสุ่มตัวอย่างแบบไม่ซ้ำ เนื่องจากขั้นตอนที่จะวิเคราะห์ต่อไปเป็นการทดสอบกับการใช้ขั้นตอนวิธีการบีบอัดข้อมูล การทำการบีบอัดข้อมูลโดยใช้วิธีการเดียวกับข้อมูลเดียวกัน จะได้ผลเหมือนกัน จึงทำการคัดยูอาร์แอลที่ซ้ำกันทิ้ง ซึ่งส่งผลให้ระดับความเชื่อมั่นสูงขึ้นกว่าที่คำนวณไว้ในขั้นต้น

3.7 การเรียกขอข้อมูลตัวอย่างจากอินเทอร์เน็ต

เมื่อได้ยูอาร์แอลของข้อมูลตัวอย่างที่ต้องการแล้ว นำไปทำการเรียกขอข้อมูลจากอินเทอร์เน็ตโดยใช้โปรแกรม Wget ซึ่งเป็นโปรแกรมที่มีอยู่ในระบบปฏิบัติการลินุกซ์ (Linux) เพื่อนำข้อมูลจากอินเทอร์เน็ตมาสู่เครื่องที่เรียกขอ โดยมีรูปแบบคำสั่งดังนี้

$wget url

เช่น ดังภาพที่ 3.4 แสดงการเรียกขอข้อมูลยูอาร์แอล “http://www.chula.ac.th/index.html” ได้ข้อมูลบันทึกลงในจานบันทึกชื่อแฟ้ม “index.html” ขนาด 482 ไบต์

ภาพที่ 3.4 การทำงานของโปรแกรม Wget

โดยข้อมูลที่ได้รับ เมื่อเทียบกับยูอาร์แอลที่เรียกขอ ปรากฏว่ามีบางข้อมูลที่มีขนาดแตกต่างไปจากยูอาร์แอลที่เรียกขอ หรือบางยูอาร์แอลที่เรียกขอไม่มีข้อมูลอยู่ ในการนี้จึงได้ทำการแก้ไขโดยการทำการสุ่มยูอาร์แอลเพิ่มเติม และทำการเรียกขอข้อมูลจากอินเทอร์เน็ตอีกครั้ง เพื่อให้ได้จำนวนตัวอย่างเท่ากับขนาดจำนวนตัวอย่างที่ได้คำนวณไว้ในหัวข้อ 3.5

ข้อมูลที่ได้รับจากการเรียกขอ ประเภทข้อมูลตัวอักษร รวมทั้งสิ้น 1,338 ไฟล์ เป็นขนาด 20,587,434 ไบต์ ประเภทข้อมูลรูปภาพเจเพก รวมทั้งสิ้น 1,434 ไฟล์ เป็นขนาด 40,403,441 ไบต์ จะถูกเก็บไว้ที่เครื่องที่ทำการเรียกขอ เพื่อนำไปทำการวิเคราะห์กับขั้นตอนวิธีการบีบอัดข้อมูลต่อไป

3.8 การวิเคราะห์หาเวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล

การวิเคราะห์หาเวลาที่ใช้ในการเรียกขอข้อมูลในแต่ละกลุ่มข้อมูล เพื่อต้องการทราบว่าโดยเฉลี่ยแล้วเวลาที่ใช้ในการเรียกขอข้อมูลจนกระทั่งได้รับข้อมูลทั้งในกรณีแคชฮิท และแคชมิสเป็นเท่าใด เพื่อหาค่าความแตกต่างระหว่างเวลาที่ใช้กรณีแคชฮิทและแคชมิส เพื่อใช้ในการพิจารณาเวลาที่ใช้ในการบีบอัดข้อมูล และเวลาในการขยายข้อมูล ว่าจะมีค่ามากที่สุดได้เป็นเท่าใด

3.8.1 ขั้นตอนการวิเคราะห์หาเวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล

3.8.1.1 การหาเวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล เมื่อเกิดแคชฮิทมีขั้นตอนดังนี้

3.8.1.1.1 รวมเวลาของการเรียกขอข้อมูลของการเกิดแคชฮิท จากแฟ้มข้อมูลบันทึกการใช้งาน ในคอลัมน์ที่ 2 ซึ่งเป็นส่วนเก็บค่าเวลาที่ใช้ในการเรียกขอข้อมูล

3.8.1.1.2 คำนวณเวลาที่ใช้ในการเรียกขอเอกสาร จากสมการ

$$\text{เวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูลเมื่อเกิดแคชฮิท} = \frac{\text{เวลาที่ใช้ในการเรียกขอข้อมูลรวมทั้งหมดเมื่อเกิดแคชฮิท}}{\text{จำนวนการเรียกขอทั้งหมดเมื่อเกิดแคชฮิท}}$$

3.8.1.2 การหาเวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล เมื่อเกิดแคชมิสมีขั้นตอนดังนี้

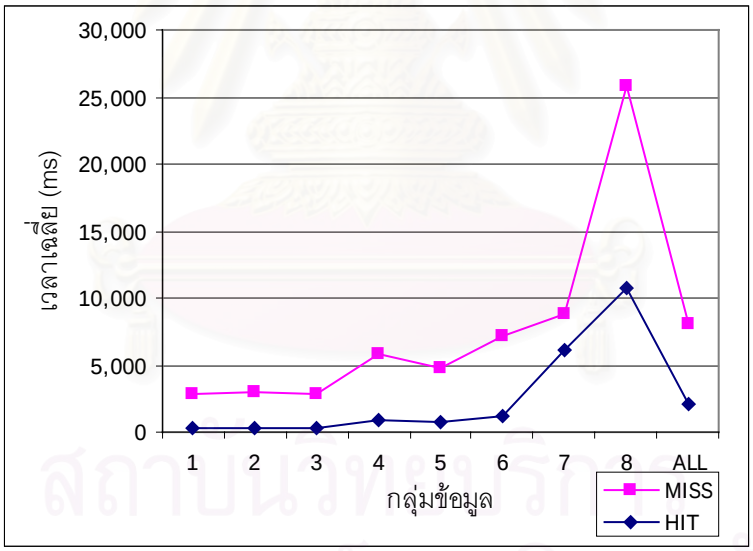
3.8.1.2.1 รวมเวลาของการเรียกขอข้อมูลของการเกิดแคชมิส จากแฟ้มข้อมูลบันทึกการใช้งาน ในคอลัมน์ที่ 2 ซึ่งเป็นส่วนเก็บค่าเวลาที่ใช้ในการเรียกขอข้อมูล

3.8.1.2.2 คำนวณเวลาที่ใช้ในการเรียกขอเอกสาร จากสมการ

$$\text{เวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูลเมื่อเกิดแคชมิส} = \frac{\text{เวลาที่ใช้ในการเรียกขอข้อมูลรวมทั้งหมดเมื่อเกิดแคชมิส}}{\text{จำนวนการเรียกขอทั้งหมดเมื่อเกิดแคชมิส}}$$

3.8.2 ผลการวิเคราะห์หาเวลาเฉลี่ย

3.8.2.1 ผลการวิเคราะห์หาเวลาเฉลี่ยของข้อมูลประเภทตัวอักษร

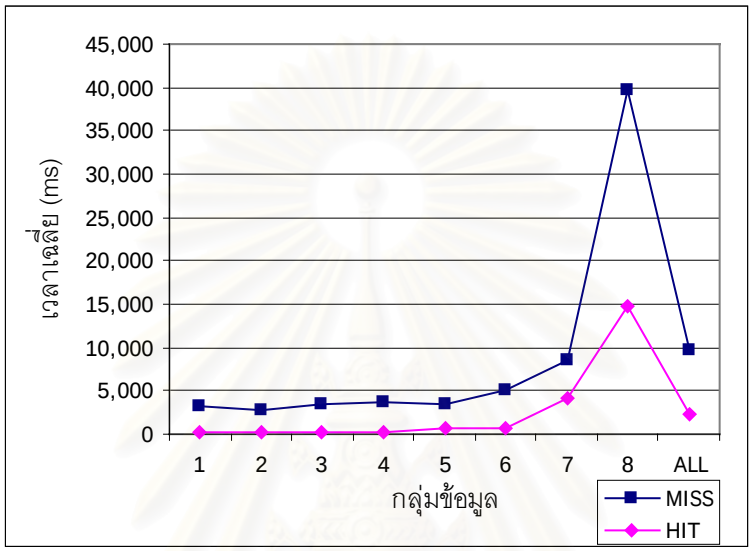


ภาพที่ 3.5 เวลาเฉลี่ยที่ใช้ในการเรียกขอเอกสารประเภทตัวอักษร เมื่อเกิดแคชฮิทและแคชมิส

จากภาพที่ 3.5 จะพบว่าเวลาเฉลี่ยที่ใช้ในการเรียกขอเอกสารในแต่ละกลุ่มข้อมูลจะมีค่าเพิ่มขึ้นเมื่อขนาดเอกสารมีขนาดใหญ่ขึ้น โดยเมื่อเกิดแคชมิส เวลาเฉลี่ยที่ใช้ในการเรียกขอเอกสารจะมีค่าที่มากกว่าเวลาที่ใช้ในการเรียกขอเอกสารเมื่อเกิดแคชฮิทอยู่ประมาณ 5 วินาที นั่นคือในการ

เลือกขั้นตอนวิธีการบีบอัดข้อมูล เวลาที่ใช้ในการขยายข้อมูลควรจะต่ำกว่า 5 วินาทีจึงจะไม่ส่งผลต่อเวลาในการตอบสนองต่อผู้ใช้

3.8.2.2 ผลการวิเคราะห์หาเวลาเฉลี่ยของข้อมูลประเภทรูปภาพเจเพก



ภาพที่ 3.6 เวลาเฉลี่ยที่ใช้ในการเรียกขอเอกสารประเภทรูปภาพเจเพก เมื่อเกิดแคชฮิทและแคชมิส

จากภาพที่ 3.6 จะพบว่าเวลาที่ใช้ในการเรียกขอเอกสารในแต่ละกลุ่มข้อมูลจะมีค่าเพิ่มขึ้นเมื่อขนาดเอกสารมีขนาดใหญ่ขึ้น โดยเมื่อเกิดแคชมิส เวลาเฉลี่ยที่ใช้ในการเรียกขอเอกสารเมื่อเกิดแคชมิส จะมีค่าที่มากกว่าเวลาที่ใช้ในการเรียกขอเอกสารเมื่อเกิดแคชฮิทอยู่ประมาณ 8 วินาที นั่นคือในการเลือกขั้นตอนวิธีการบีบอัดข้อมูล เวลาที่ใช้ในการขยายข้อมูลควรจะต่ำกว่า 8 วินาทีจึงจะไม่ส่งผลต่อเวลาในการตอบสนองต่อผู้ใช้

3.9 สรุปผลการวิเคราะห์สภาพการใช้เว็บ

สัดส่วนของประเภทข้อมูลตามการใช้งานของจุฬาลงกรณ์มหาวิทยาลัย คือ รูปภาพจิฟ ข้อมูลตัวอักษรเอชทีเอ็มแอล รูปภาพเจเพก และข้อมูลประเภทอื่น ประเภทข้อมูลที่นำมาวิเคราะห์กับขั้นตอนวิธีบีบอัดข้อมูล คือ ข้อมูลตัวอักษร และรูปภาพเจเพก

เมื่อทำการวิเคราะห์การกระจายของขนาดข้อมูลกับจำนวนแฟ้มข้อมูล พบว่าขนาดแฟ้มข้อมูลทั้งข้อมูลประเภทตัวอักษร และรูปภาพเจเพก มีการกระจายตัวอยู่ในช่วงขนาดไม่เกิน 100,000 ไบต์ เมื่อขนาดข้อมูลใหญ่กว่า 100,000 ไบต์แล้ว จะมีจำนวนแฟ้มข้อมูลน้อยลงมากอย่างเห็นได้ชัดเจน การแบ่งข้อมูลออกเป็นกลุ่ม จึงใช้ค่าเปอร์เซ็นไตล์ที่ 12.5, 25, 37.5, 50, 62.5, 75, 87.5 และ 100 เพื่อแบ่งข้อมูลออกเป็น 8 กลุ่ม

เมื่อแบ่งข้อมูลออกเป็นกลุ่มแล้ว ขนาดข้อมูลในแต่ละกลุ่มมีจำนวนมาก จึงเป็นการเสียเวลาที่จะทำการวิเคราะห์กับข้อมูลทั้งหมด จึงใช้กลุ่มตัวอย่างเพื่อเป็นตัวแทนของข้อมูลในกลุ่ม จากผลการคำนวณได้จำนวนตัวอย่างที่จะใช้เป็นตัวแทน ประมาณ 5% จากข้อมูลทั้งหมด

จากนั้นจึงทำการสุ่มเลือกยูอาร์แอล แบบไม่ซ้ำ และเรียกขอยูอาร์แอลจากอินเทอร์เน็ต ได้ข้อมูลประเภทตัวอักษรขนาดทั้งหมด 20 เมกะไบต์ และข้อมูลประเภทรูปภาพเจเพกขนาดทั้งหมด 40 เมกะไบต์ เพื่อนำไปวิเคราะห์กับขั้นตอนวิธีการบีบอัดข้อมูล

ในการวิเคราะห์หาค่าเวลาเฉลี่ยที่ใช้ในการเรียกขอเอกสาร พบว่าสำหรับข้อมูลประเภทตัวอักษร มีค่าประมาณ 5 วินาที และสำหรับข้อมูลประเภทรูปภาพเจเพก มีค่าประมาณ 8 วินาที โดยค่าเวลาเฉลี่ยนี้จะถูกใช้เป็นเกณฑ์ที่ใช้ในการเลือกขั้นตอนวิธีการบีบอัดที่เหมาะสม เพราะขั้นตอนวิธีการบีบอัดที่เลือก ควรจะมีค่าเวลาที่ใช้ในการบีบอัดข้อมูลและเวลาที่ใช้ในการขยายข้อมูลน้อยกว่าเวลาเฉลี่ยที่ใช้ในการเรียกขอเอกสารของเว็บแคชที่ไม่มีการนำขั้นตอนวิธีการบีบอัดข้อมูลมาใช้ จึงจะไม่ทำให้ค่าเวลาตอบสนองต่อผู้ใช้เพิ่มขึ้น

บทที่ 4
การวิเคราะห์เปรียบเทียบขั้นตอนวิธีการบีบอัดข้อมูล

ขั้นตอนวิธีการบีบอัดข้อมูลแต่ละชนิดมีลักษณะการทำงานที่แตกต่างกัน การที่จะระบุได้ว่าขั้นตอนวิธีการบีบอัดข้อมูลใด มีประสิทธิภาพและความเหมาะสมที่จะนำไปประยุกต์ใช้กับการบีบอัดข้อมูลในเว็บแคช ต้องทำการวิเคราะห์เปรียบเทียบในปัจจัยต่างๆ เพื่อหาขั้นตอนวิธีการบีบอัดข้อมูลที่เหมาะสม

4.1 ขั้นตอนวิธีการบีบอัดข้อมูล

ขั้นตอนวิธีการบีบอัดข้อมูลที่ใช้ในการวิเคราะห์ มีอยู่ 6 ขั้นตอนวิธีการดังนี้

- Huffman Coding
- ZIP
- LZARI
- LZHUF
- LZO
- LZSS

โดยขั้นตอนวิธีทั้งหมด เลือกระดับการบีบอัดแบบปกติ

4.2 การทดสอบประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูล

การทดสอบประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูลแต่ละชนิด เพื่อเปรียบเทียบประสิทธิภาพของการบีบอัดที่ทำกับข้อมูล โดยการวัดจากปัจจัย 3 ประการ ดังนี้

4.2.1 อัตราการบีบอัดข้อมูล

อัตราการบีบอัดข้อมูล เป็นค่าที่บอกความสามารถในการบีบอัดข้อมูล อัตราการบีบอัดข้อมูลยิ่งมาก จะยิ่งส่งผลดีกับเว็บแคช เนื่องจากเว็บแคชใช้เนื้อที่ในการเก็บข้อมูลน้อยลง

4.2.2 เวลาที่ใช้ในการบีบอัดข้อมูล

เวลาที่ใช้ในการบีบอัดข้อมูล เป็นเวลาทั้งหมดที่ขั้นตอนวิธีการบีบอัดข้อมูลใช้ในการบีบอัดข้อมูล วัดโดยการใช้โปรแกรมจับเวลาการทำงาน Ultra Precision Command Timer (UPCT) [19] โดยเวลาที่ใช้ในการบีบอัดข้อมูลที่น้อย จะส่งผลดีต่อเว็บแคช คือเวลาในการทำงานโดยรวมของเว็บแคชจะไม่ถูกกระทบกระเทือนให้ช้าลง

4.2.3 เวลาที่ใช้ในการขยายข้อมูล

เวลาที่ใช้ในการขยายข้อมูล เป็นเวลาทั้งหมดที่ขั้นตอนวิธีการบีบอัดข้อมูลใช้ในการขยายข้อมูล วัดโดยการใช้โปรแกรมจับเวลาการทำงาน Ultra Precision Command Timer โดยเวลาที่ใช้ในการขยายข้อมูลที่น้อย จะส่งผลดีต่อเว็บแคช คือเวลาในการทำงานเพื่อนำข้อมูลกลับเข้ารูปแบบเดิมสั้น ทำให้ข้อมูลสามารถถูกส่งไปยังผู้เรียกขอได้เร็วขึ้น

4.3 ขั้นตอน วิธีการ และสภาพแวดล้อมที่ใช้ในการวิเคราะห์

4.3.1 สภาพแวดล้อมของการวิเคราะห์

เนื่องจากการวิเคราะห์เพื่อเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูลนี้ ต้องการสภาพแวดล้อมของระบบการทำงานที่มีสภาพคงที่ จึงพยายามกำจัดปัจจัยด้านสภาพแวดล้อมอื่นๆ เช่น การแบ่งซีพียูไปทำงานอื่นของระบบปฏิบัติการ ที่อาจส่งผลต่อการทำงานของขั้นตอนวิธีการบีบอัดข้อมูล จึงทำการวิเคราะห์บนระบบปฏิบัติการดอส (DOS) ซึ่งเป็นระบบปฏิบัติการที่ไม่มีการทำงานแบบหลายภารกิจ (Multitasking) จึงทำให้ค่าเวลาที่วัดได้ใกล้เคียงกับค่าที่เป็นจริงมากที่สุด เครื่องคอมพิวเตอร์ที่ใช้ในการทดลองการบีบอัดข้อมูล ในวิทยานิพนธ์นี้ มีรายละเอียดดังนี้

4.3.1.1 ฮาร์ดแวร์

4.3.1.1.1 คอมพิวเตอร์แบบพีซี หน่วยประมวลผลกลาง Pentium ความเร็ว 100 เมกะเฮิรตซ์

4.3.1.1.2 หน่วยความจำ 64 เมกะไบต์

4.3.1.1.3 จานบันทึกแบบแข็ง ความจุ 1.7 กิกะไบต์

4.3.1.2 ซอฟต์แวร์

4.3.1.2.1 ระบบปฏิบัติการดอส (DOS) เวอร์ชัน 7 โดยมีการปรับแต่งไฟล์ Config.sys ดังนี้

```
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE RAM
FILESHIGH=50
BUFFERSHIGH=30
DOS=UMB
DOS=HIGH
SHELL=C:\COMMAND.COM C:\ /E:800 /p
```

4.3.1.2.2 โปรแกรมจับเวลา Ultra Precision Command Timer เวอร์ชัน 1.5 โดย Erik de Neve

4.3.2 ขั้นตอนการทำงาน

ขั้นตอนการทำการวิเคราะห์เพื่อวัดประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูล มีดังนี้

4.3.2.1 ส่วนการบีบอัดข้อมูล นำข้อมูลจากกลุ่มข้อมูลตัวอย่างที่ได้จากการคัดเลือกจากบทที่ 3 นำมาทำการวิเคราะห์ โดยการใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลคราวละ 1 แฟ้มข้อมูล โดยมีลักษณะคำสั่งดังนี้

c:\upct ขั้นตอนวิธีการบีบอัดข้อมูล แฟ้มข้อมูลที่ทดลอง

แล้วทำการวัดค่าเวลาที่ใช้ในการบีบอัดข้อมูลและขนาดแฟ้มข้อมูลที่ได้หลังการถูกบีบอัด

4.3.2.2 ส่วนการขยายข้อมูล นำข้อมูลที่ถูกบีบอัดจากข้อ 4.3.2.1 มาทำการขยายข้อมูล โดยมีลักษณะคำสั่งดังนี้

c:\upct ขั้นตอนวิธีการบีบอัดข้อมูล แฟ้มข้อมูลที่ถูกบีบอัด

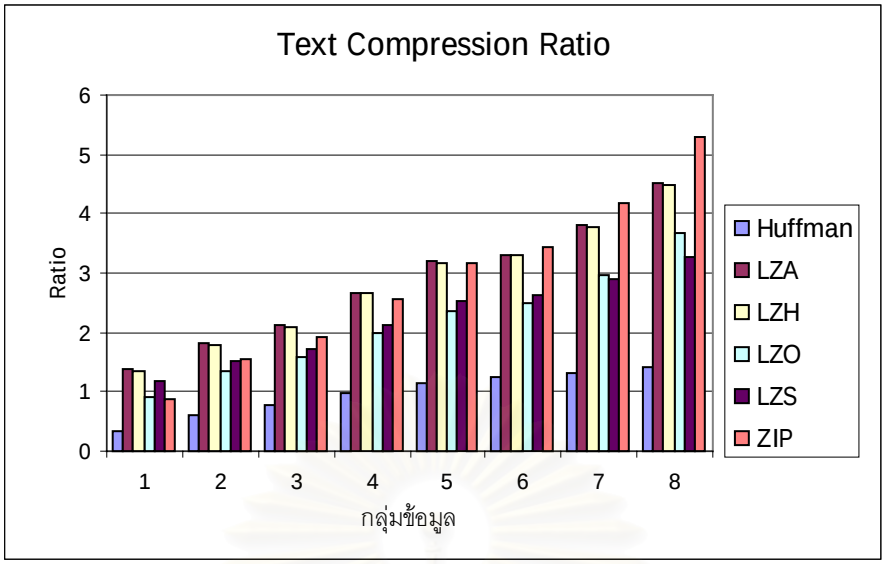
แล้วทำการวัดค่าเวลาที่ใช้ในการขยายข้อมูล

4.4 ผลการทดลอง

ผลการวิเคราะห์การเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูล คือ

4.4.1 ข้อมูลประเภทตัวอักษร

4.4.1.1 อัตราการบีบอัดข้อมูล แสดงเปรียบเทียบในภาพที่ 4.1



ภาพที่ 4.1 ผลการทดลองอัตราการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

ตารางที่ 4.1 สรุปอันดับอัตราการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

กลุ่มข้อมูล	ขั้นตอนวิธีการบีบอัดข้อมูล					
	HUFF	LZA	LZH	LZO	LZSS	ZIP
1	6	1	2	4	3	5
2	6	1	2	5	4	3
3	6	1	2	5	4	3
4	6	1	2	5	4	3
5	6	1	3	5	4	2
6	6	2	3	5	4	1
7	6	2	3	4	5	1
8	6	2	3	4	5	1

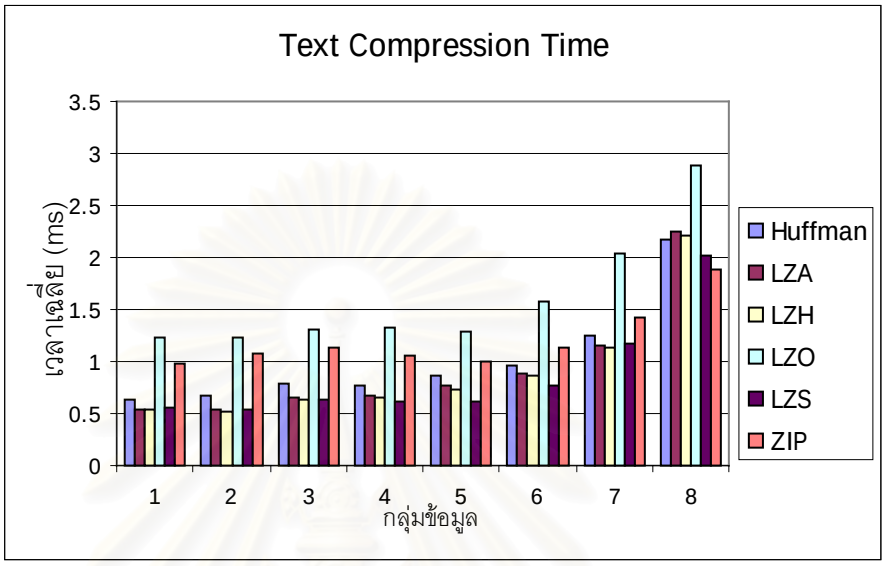
จากการทดลองเพื่อเปรียบเทียบประสิทธิภาพ โดยใช้อัตราการบีบอัดข้อมูล เป็นปัจจัยในการพิจารณา พบว่า

ขั้นตอนวิธี LZA มีประสิทธิภาพในการบีบอัดข้อมูลสูงสุด ในข้อมูลในกลุ่มที่ 1 ถึง 5 หรือขนาดข้อมูลน้อยกว่า 7,120 ไบต์ โดยมีอัตราการบีบอัดประมาณ 1.5 - 3.2

ขั้นตอนวิธี ZIP มีประสิทธิภาพในการบีบอัดข้อมูลสูงสุด ในกลุ่มที่ 6 ถึงกลุ่มที่ 8 หรือขนาดข้อมูลใหญ่กว่า 7,121 ไบต์ โดยมีอัตราการบีบอัดประมาณ 3.5 – 5.4

สังเกตได้ว่าเมื่อขนาดของข้อมูลประเภทตัวอักษรใหญ่ขึ้น อัตราการบีบอัดข้อมูลจะเพิ่มขึ้น

4.4.1.2 เวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูล



ภาพที่ 4.2 ผลการทดลองเวลาที่ในการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

ตารางที่ 4.2 สรุปอันดับเวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

กลุ่มข้อมูล	ขั้นตอนวิธีการบีบอัดข้อมูล					
	HUFF	LZA	LZH	LZO	LZSS	ZIP
1	4	2	1	6	3	5
2	4	2	1	6	3	5
3	4	3	2	6	1	5
4	4	3	2	6	1	5
5	4	3	2	6	1	5
6	4	3	2	6	1	5
7	4	2	1	6	3	5
8	3	5	4	6	2	1

จากการทดลองเพื่อเปรียบเทียบประสิทธิภาพ โดยใช้เวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูล เป็นปัจจัยในการพิจารณา พบว่า

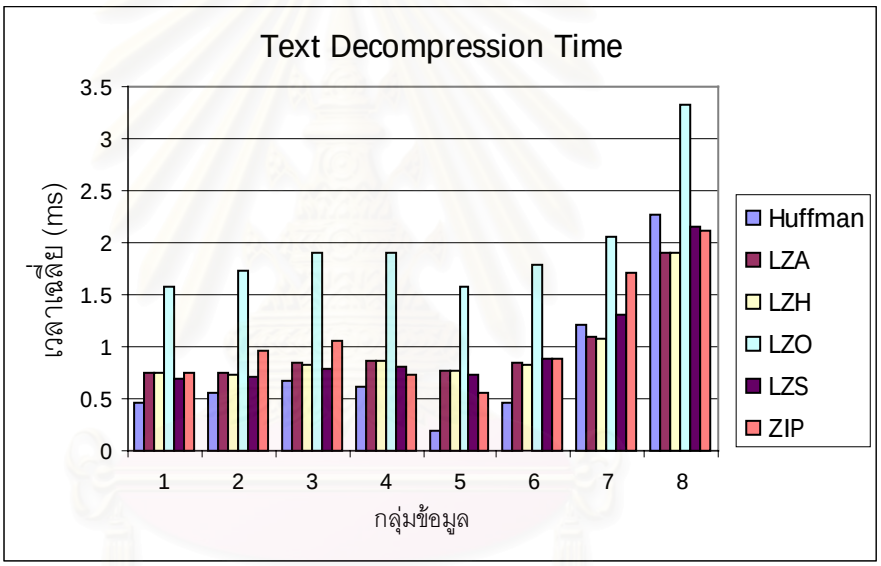
ขั้นตอนวิธี LZH ใช้เวลาในการบีบอัดข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 1, 2 และ 7 หรือขนาดข้อมูลเล็กกว่า 1,147 ไบต์ และขนาดข้อมูลอยู่ในช่วง 11,423-21,188 ไบต์ โดยใช้เวลาประมาณ 0.5 – 1.2 วินาที

ขั้นตอนวิธี LZSS ใช้เวลาในการบีบอัดข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 3 ถึง 6 หรือขนาดข้อมูลในช่วง 1,148-11,422 ไบต์ โดยใช้เวลาประมาณ 0.6 – 0.7 วินาที

ขั้นตอนวิธี ZIP ใช้เวลาในการบีบอัดข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มข้อมูลที่ 8 หรือขนาดข้อมูลใหญ่กว่า 21,190 ไบต์ โดยใช้เวลาประมาณ 1.9 วินาที

สังเกตได้ว่า เมื่อขนาดของข้อมูลประเภทตัวอักษรใหญ่ขึ้น เวลาที่ใช้ในการบีบอัดข้อมูลจะเพิ่มขึ้น

4.4.1.3 เวลาเฉลี่ยที่ใช้ในการขยายข้อมูล



ภาพที่ 4.3 ผลการทดลองเวลาที่ในการขยายข้อมูลของข้อมูลประเภทตัวอักษร

จากการทดลองเพื่อเปรียบเทียบประสิทธิภาพ โดยใช้เวลาที่ใช้ในการขยายข้อมูลเป็นปัจจัยในการพิจารณา พบว่า

ขั้นตอนวิธี HUFF ใช้เวลาในการขยายข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 1 ถึงกลุ่ม 6 หรือขนาดข้อมูลเล็กกว่า 11,422 ไบต์ โดยใช้เวลาประมาณ 0.5 วินาที

ขั้นตอนวิธี LZH ใช้เวลาในการขยายข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 7 หรือขนาดข้อมูลในช่วง 11,423 – 21,188 ไบต์ โดยใช้เวลาประมาณ 1.1 วินาที

ขั้นตอนวิธี LZA ใช้เวลาในการขยายข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มข้อมูลที่ 8 หรือขนาดข้อมูลใหญ่กว่า 21,190 ไบต์ โดยใช้เวลาประมาณ 1.9 วินาที

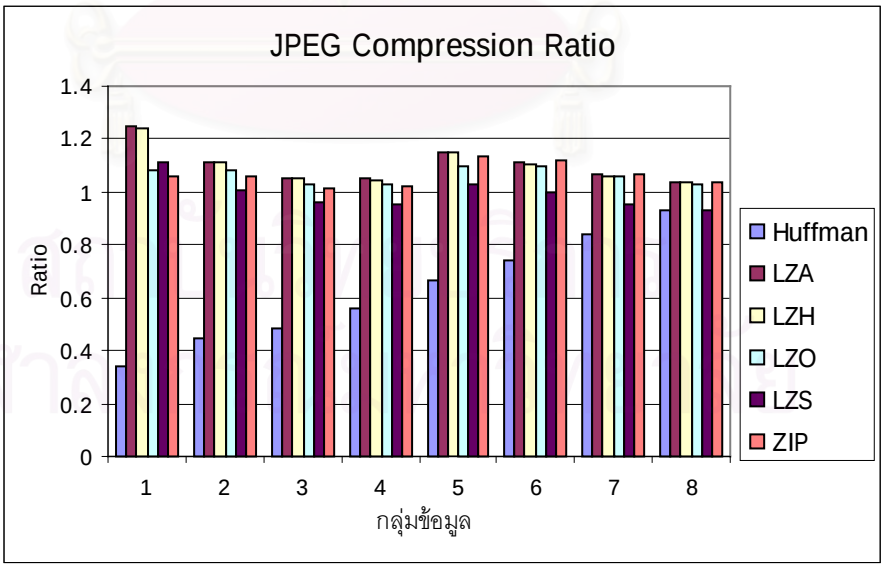
ตารางที่ 4.3 สรุปอันดับเวลาเฉลี่ยที่ใช้ในการขยายข้อมูลของข้อมูลประเภทตัวอักษร

กลุ่มข้อมูล	ขั้นตอนวิธีการบีบอัดข้อมูล					
	HUFF	LZA	LZH	LZO	LZSS	ZIP
1	1	4	3	6	2	5
2	1	4	3	6	2	5
3	1	4	3	6	4	5
4	1	5	4	6	3	2
5	1	5	4	6	3	2
6	1	3	2	6	4	5
7	3	2	1	6	4	5
8	5	1	2	6	4	3

สังเกตได้ว่า เมื่อขนาดของข้อมูลประเภทตัวอักษรใหญ่ขึ้น เวลาที่ใช้ในการขยายข้อมูลมีแนวโน้มจะเพิ่มขึ้น

4.4.2 ข้อมูลประเภทรูปภาพเจเพก

4.4.2.1 อัตราการบีบอัดข้อมูล แสดงเปรียบเทียบในภาพที่ 4.4



ภาพที่ 4.4 ผลการทดลองอัตราการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ตารางที่ 4.4 สรุปอันดับอัตราการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

กลุ่มข้อมูล	ขั้นตอนวิธีการบีบอัดข้อมูล					
	HUFF	LZA	LZH	LZO	LZSS	ZIP
1	6	1	2	4	3	5
2	6	1	2	3	5	4
3	6	1	2	3	5	4
4	6	1	2	3	5	4
5	6	1	2	4	5	3
6	6	2	3	4	5	1
7	6	2	3	4	5	1
8	5	1	3	4	6	2

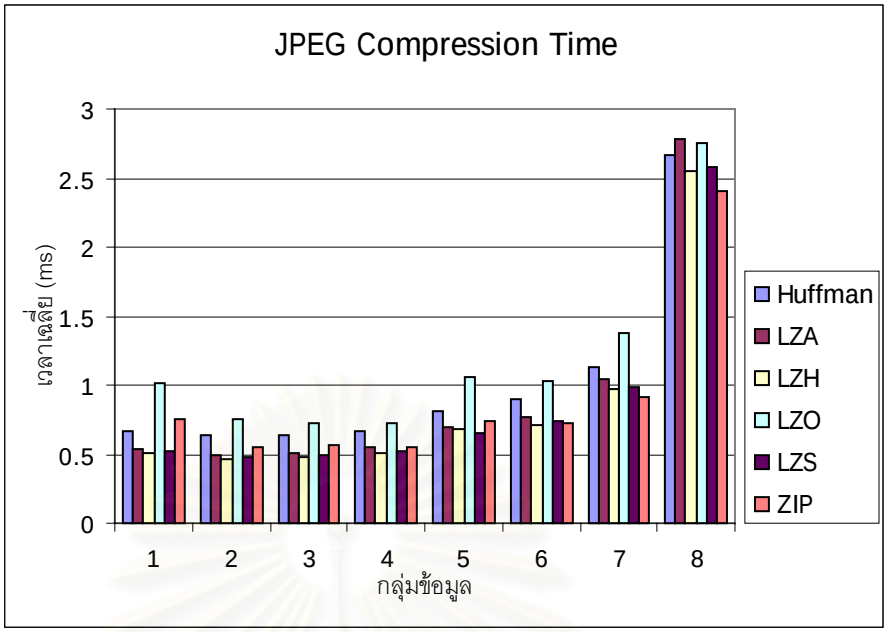
จากการทดลองเพื่อเปรียบเทียบประสิทธิภาพ โดยใช้อัตราการบีบอัดข้อมูล เป็นปัจจัยในการพิจารณา พบว่า

ขั้นตอนวิธี LZA มีประสิทธิภาพในการบีบอัดข้อมูลสูงสุด ในข้อมูลในกลุ่มที่ 1 ถึง 5 และกลุ่มที่ 8 หรือขนาดข้อมูลน้อยกว่า 6,344 ไบต์ และขนาดข้อมูลที่ใหญ่กว่า 19,929 โดยมีอัตราการบีบอัดประมาณ 1.1 – 1.25

ขั้นตอนวิธี ZIP มีประสิทธิภาพในการบีบอัดข้อมูลสูงสุด ในกลุ่มที่ 6 และกลุ่มที่ 7 หรือขนาดข้อมูลอยู่ในช่วง 6,346-19,925 ไบต์ โดยมีอัตราการบีบอัดประมาณ 1.1

สังเกตได้ว่า ขนาดของข้อมูลประเภทรูปภาพเจเพก ไม่มีความสัมพันธ์กับอัตราการบีบอัดข้อมูล ยกเว้นขั้นตอนวิธี Huffman ที่มีอัตราการบีบอัดเพิ่มขึ้นเมื่อขนาดข้อมูลใหญ่ขึ้น

4.4.2.2 เวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูล



ภาพที่ 4.5 ผลการทดลองเวลาเฉลี่ยที่ในการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ตารางที่ 4.5 สรุปอันดับเวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

กลุ่มข้อมูล	ขั้นตอนวิธีการบีบอัดข้อมูล					
	HUFF	LZA	LZH	LZO	LZSS	ZIP
1	4	3	1	6	2	5
2	5	3	1	6	2	4
3	5	3	1	6	2	4
4	5	3	1	6	2	4
5	5	3	2	6	1	4
6	5	4	1	6	3	2
7	5	4	2	6	3	1
8	4	6	2	5	3	1

จากการทดลองเพื่อเปรียบเทียบประสิทธิภาพ โดยใช้เวลาที่ใช้ในการบีบอัดข้อมูลเป็นปัจจัยในการพิจารณา พบว่า

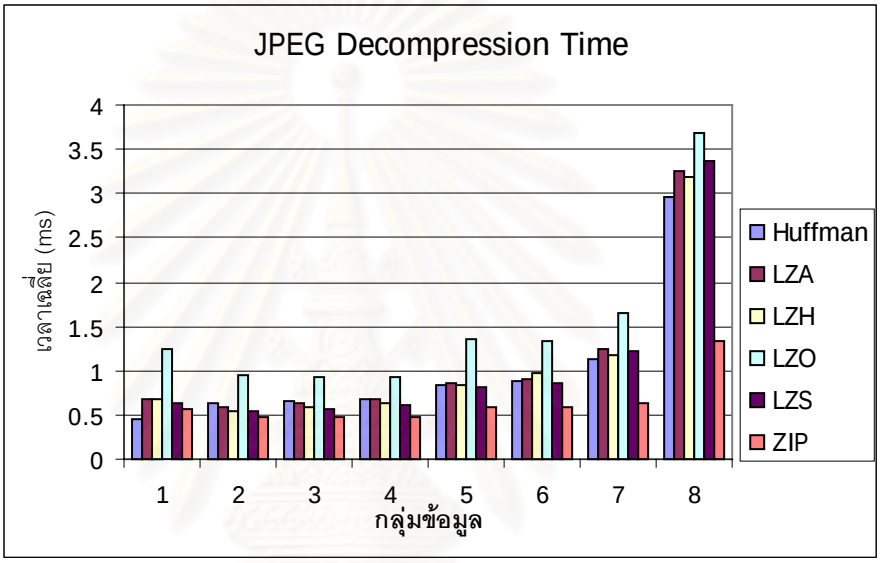
ขั้นตอนวิธี LZH ใช้เวลาในการบีบอัดข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 1 ถึง 4 และกลุ่มที่ 6 หรือขนาดข้อมูลเล็กกว่า 4,205 ไบต์ และขนาดข้อมูลอยู่ในช่วง 6,346-10,306 ไบต์ โดยใช้เวลาประมาณ 0.5 – 0.7 วินาที

ขั้นตอนวิธี LZSS ใช้เวลาในการบีบอัดข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 5 หรือขนาดข้อมูลในช่วง 4,206-6,344 ไบต์ โดยใช้เวลาประมาณ 0.7 วินาที

ขั้นตอนวิธี ZIP ใช้เวลาในการบีบอัดข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มข้อมูลที่ 7 และ 8 หรือขนาดข้อมูลใหญ่กว่า 10,309 ไบต์ โดยใช้เวลาประมาณ 0.9 – 2.4 วินาที

สังเกตได้ว่า เมื่อขนาดของข้อมูลประเภทรูปภาพเจเพกใหญ่ขึ้น เวลาที่ใช้ในการบีบอัดข้อมูลจะเพิ่มขึ้น

4.4.2.3 เวลาเฉลี่ยที่ใช้ในการขยายข้อมูล



ภาพที่ 4.6 ผลการทดลองเวลาที่ในการขยายข้อมูลของข้อมูลประเภทรูปภาพเจเพก

จากการทดลองเพื่อเปรียบเทียบประสิทธิภาพ โดยใช้เวลาที่ใช้ในการขยายข้อมูลเป็นปัจจัยในการพิจารณา พบว่า

ขั้นตอนวิธี HUFFMAN ใช้เวลาในการขยายข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 1 หรือขนาดข้อมูลเล็กกว่า 2,054 ไบต์ โดยใช้เวลาประมาณ 0.5 วินาที

ขั้นตอนวิธี ZIP ใช้เวลาในการขยายข้อมูลน้อยที่สุดกับข้อมูลในกลุ่มที่ 2 ถึง 8 หรือขนาดข้อมูลใหญ่กว่า 2,055 ไบต์ โดยใช้เวลาประมาณ 0.5 – 1.4 วินาที

สังเกตได้ว่า เมื่อขนาดของข้อมูลประเภทรูปภาพเจเพกใหญ่ขึ้น เวลาที่ใช้ในการขยายข้อมูลจะเพิ่มขึ้น

ตารางที่ 4.6 สรุปอันดับเวลาเฉลี่ยที่ใช้ในการขยายข้อมูลของข้อมูลประเภทรูปภาพเจเพก

กลุ่มข้อมูล	ขั้นตอนวิธีการบีบอัดข้อมูล					
	HUFF	LZA	LZH	LZO	LZSS	ZIP
1	1	5	4	6	3	2
2	5	4	2	6	3	1
3	5	4	3	6	2	1
4	5	4	3	6	2	1
5	3	5	4	6	2	1
6	3	4	5	6	2	1
7	2	5	3	6	4	1
8	2	4	3	6	5	1

4.5 สรุปผลการวิเคราะห์เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูล

จากการวิเคราะห์ประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูล โดยใช้ปัจจัยๆ เดียวเป็นตัวตัดสินใจ จะพบว่า

4.5.1 ข้อมูลประเภทตัวอักษร

4.5.1.1 อัตราการบีบอัดข้อมูล

ตารางที่ 4.7 สรุปผลอัตราการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

ขนาดข้อมูล (ไบต์)	ขั้นตอนวิธีการบีบอัดข้อมูล	อัตราการบีบอัดข้อมูล
< 7,120	LZA	1.5 – 3.2
> 7,120	ZIP	3.5 – 5.4

โดยอัตราการบีบอัดข้อมูลมีความสัมพันธ์ในทิศทางเดียวกันกับขนาดของข้อมูล

4.5.1.2 เวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูล

ตารางที่ 4.8 สรุปผลเวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

ขนาดข้อมูล (ไบต์)	ขั้นตอนวิธีการบีบอัดข้อมูล	เวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูล (วินาที)
< 1,147	LZH	0.5
1,148-11,422	LZSS	0.6 – 0.7
11,423-21,188	LZH	1.2
>21,190	ZIP	1.9

โดยเวลาที่ใช้ในการบีบอัดข้อมูลมีความสัมพันธ์ในทิศทางเดียวกันกับขนาดของข้อมูล

4.5.1.3 เวลาเฉลี่ยที่ใช้ในการขยายข้อมูล

ตารางที่ 4.9 สรุปผลเวลาเฉลี่ยที่ใช้ในการขยายข้อมูลของข้อมูลประเภทตัวอักษร

ขนาดข้อมูล (ไบต์)	ขั้นตอนวิธีการบีบอัดข้อมูล	เวลาเฉลี่ยที่ใช้ในการขยายข้อมูล (วินาที)
<11,422	HUFF	0.5
11,423-21,188	LZH	1.1
>21,190	LZA	1.9

โดยเวลาที่ใช้ในการขยายข้อมูลมีแนวโน้มความสัมพันธ์ในทิศทางเดียวกันกับขนาดของข้อมูล

4.5.2 ข้อมูลประเภทรูปภาพเจเพก

4.5.2.1 อัตราการบีบอัดข้อมูล

ตารางที่ 4.10 สรุปผลอัตราการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ขนาดข้อมูล (ไบต์)	ขั้นตอนวิธีการบีบอัดข้อมูล	อัตราการบีบอัดข้อมูล
< 6,344	LZA	1.1 – 1.25
6,346-19,925	ZIP	1.1
> 19,929	LZA	1.05

โดยขั้นตอนวิธีการบีบอัดข้อมูลส่วนใหญ่ไม่มีความสัมพันธ์ระหว่างอัตราการบีบอัดข้อมูลกับขนาดข้อมูล นอกจากขั้นตอนวิธี Huffman ที่มีอัตราการบีบอัดเพิ่มขึ้นเมื่อขนาดข้อมูลใหญ่ขึ้น

4.5.2.2 เวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูล

ตารางที่ 4.11 สรุปผลเวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ขนาดข้อมูล (ไบต์)	ขั้นตอนวิธีการบีบอัดข้อมูล	เวลาเฉลี่ยที่ใช้ในการบีบอัดข้อมูล (วินาที)
< 4,205	LZH	0.5
4,206-6,344	LZSS	0.6
6,346-10,306	LZH	0.7
>10,309	ZIP	0.9 - 2.4

โดยขนาดของข้อมูลประเภทรูปภาพเจเพกมีความสัมพันธ์ในทิศทางเดียวกันกับเวลาที่ใช้ในการบีบอัดข้อมูล

4.5.2.3 เวลาเฉลี่ยที่ใช้ในการขยายข้อมูล

ตารางที่ 4.12 สรุปผลเวลาเฉลี่ยที่ใช้ในการขยายข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ขนาดข้อมูล (ไบต์)	ขั้นตอนวิธีการบีบอัดข้อมูล	เวลาเฉลี่ยที่ใช้ในการขยายข้อมูล (วินาที)
<2,054	HUFFMAN	0.5
>2,055	ZIP	0.5 – 1.4

โดยขนาดของข้อมูลประเภทรูปภาพเจเพกมีความสัมพันธ์ในทิศทางเดียวกันกับเวลาที่ใช้ในการขยายข้อมูล

แต่เมื่อพิจารณาถึงการนำไปประยุกต์ใช้ในเว็บแคช ปัจจัยที่ต้องคำนึงถึงเป็นสิ่งแรกคือ อัตราการบีบอัดข้อมูล สาเหตุที่ไม่ให้ความสำคัญกับเวลาที่ใช้ในการบีบอัดข้อมูล และเวลาที่ใช้ในการขยายข้อมูล เนื่องจากผลของการวิเคราะห์เวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล ซึ่งพบว่าเวลาที่แตกต่างในการเรียกขอข้อมูลกรณีแคชมิสกับการเรียกขอกรณีแคชฮิทของข้อมูลประเภทตัว

อักษรและข้อมูลรูปภาพเจเพกเป็น 5 วินาทีและ 8 วินาทีตามลำดับ ซึ่งมีค่าสูงกว่าเวลาที่ใช้ในการบีบอัดข้อมูลและเวลาที่ใช้ในการขยายข้อมูลที่ใช้เวลาไม่เกิน 2.5 วินาที

ดังนั้นสำหรับข้อมูลประเภทตัวอักษร เมื่อข้อมูลมีขนาดน้อยกว่า 7KB ควรใช้ขั้นตอนวิธี LZA และขนาดข้อมูลมากกว่า 7KB ควรใช้ขั้นตอนวิธี ZIP

สำหรับข้อมูลประเภทรูปภาพเจเพก เมื่อข้อมูลมีขนาดน้อยกว่า 6KB ควรใช้ขั้นตอนวิธี LZA ขนาดข้อมูลระหว่าง 6-20KB ควรใช้ขั้นตอนวิธี ZIP และขนาดข้อมูลมากกว่า 20KB ควรใช้ขั้นตอนวิธี LZA

จากการทดลองพบว่าขั้นตอนวิธีการบีบอัดข้อมูลที่ให้อัตราการบีบอัดข้อมูลที่สูงที่สุด คือ LZA และ ZIP ซึ่งทั้งสองขั้นตอนวิธีการมีการใช้แบบจำลองข้อมูลของ LZ เหมือนกันในการทำการสร้างโครงสร้างข้อมูลในขั้นที่หนึ่ง จะต่างกันในส่วนการทำงานในขั้นที่สอง ซึ่ง LZA ใช้แบบจำลองแบบเรขาคณิตเพื่อกำหนดรหัส ซึ่งใช้การจัดลำดับของความน่าจะเป็นที่จะพบตัวอักษรหรือสัญลักษณ์นั้น ซึ่งจะเหมาะกับข้อมูลที่มีขนาดเล็ก แต่ ZIP ใช้แบบจำลองแบบ Huffman ซึ่งเป็นการสร้างโครงสร้างต้นไม้ ซึ่งการใช้โครงสร้างต้นไม้ เมื่อขนาดข้อมูลมีขนาดใหญ่ การเข้ารหัสโดยวิธีการนี้จะมีประสิทธิภาพดีกว่า

บทที่ 5
สรุปผลการวิจัย การประยุกต์ใช้งาน และ ข้อเสนอแนะ

จากผลของการวิเคราะห์สภาพการใช้เว็บของจุฬาลงกรณ์มหาวิทยาลัย และการวิเคราะห์เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลที่มักปรากฏในเว็บแคช สามารถสรุปผลที่ได้จากการวิจัย เสนอแนวคิดในการประยุกต์ใช้งานขั้นตอนวิธีการบีบอัดข้อมูลกับเว็บแคชเพื่อปรับปรุงประสิทธิภาพของเว็บแคชในส่วนการเพิ่มเนื้อที่เก็บข้อมูล และเสนอแนะแนวทางในการวิจัยต่อไป ได้ดังนี้

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้ได้ทำการวิเคราะห์สภาพการใช้เว็บของจุฬาลงกรณ์มหาวิทยาลัย เพื่อศึกษาลักษณะประเภทข้อมูลที่มีการเรียกขอ และอยู่ในเว็บแคช เมื่อทราบถึงประเภทข้อมูลที่มีการเรียกขอมาก ทำการวิเคราะห์หาการกระจายของขนาดข้อมูล เพื่อทำการแบ่งกลุ่มข้อมูลออกเป็นกลุ่มที่เท่ากัน เพื่อนำไปวิเคราะห์กับขั้นตอนวิธีการบีบอัดข้อมูล แต่เนื่องจากข้อมูลมีจำนวนมาก จึงทำการคำนวณหาขนาดกลุ่มข้อมูลตัวอย่างเพื่อใช้เป็นตัวแทนของข้อมูลทั้งหมดในแต่ละกลุ่ม โดยกำหนดระดับความเชื่อมั่นของการเลือกกลุ่มตัวอย่างไว้ที่ 95% ทำการสุ่มเลือกยูอาร์แอลแบบไม่ซ้ำ เพื่อนำไปทำการเรียกขอข้อมูลจากอินเทอร์เน็ต เมื่อได้ข้อมูลตัวอย่างจากอินเทอร์เน็ตแล้วนำมาทำการวิเคราะห์กับขั้นตอนวิธีการบีบอัดข้อมูล 6 ขั้นตอนวิธี คือ Huffman Coding, LZA, LZH, LZO, LZSS และ ZIP ทำการวัดอัตราการบีบอัดข้อมูล เวลาที่ใช้ในการบีบอัดข้อมูล และเวลาที่ใช้ในการขยายข้อมูล นำค่าที่วัดได้มาทำการเปรียบเทียบเพื่อเลือกขั้นตอนวิธีการบีบอัดข้อมูลที่เหมาะสมกับการประยุกต์ใช้กับเว็บแคช

จากการวิเคราะห์สภาพการใช้งานอินเทอร์เน็ตของจุฬาลงกรณ์มหาวิทยาลัย โดยใช้แฟ้มบันทึกการใช้งาน ของช่วงวันที่ 3 ตุลาคม ถึง 1 พฤศจิกายน พ.ศ. 2542 จากสำนักเทคโนโลยีสารสนเทศ จุฬาลงกรณ์มหาวิทยาลัย พบว่าประเภทข้อมูลที่มีการเรียกขอสูงสุดคือ รูปภาพจิฟ 24% ข้อมูลตัวอักษรเอชทีเอ็มแอล 20% และรูปภาพเจเพก 19% แต่ข้อมูลรูปภาพจิฟมีการใช้ขั้นตอนวิธีการบีบอัดในรูปแบบแฟ้มข้อมูลแล้ว จึงไม่นำมาวิเคราะห์กับขั้นตอนวิธีการบีบอัด

- ในการวิเคราะห์การกระจายของขนาดข้อมูล พบว่าข้อมูลที่พบส่วนมากจะมีขนาดเล็กกว่า 10 กิโลไบต์ ส่วนข้อมูลที่มีขนาดใหญ่กว่า 10 กิโลไบต์จะมีอยู่น้อยมาก

- การแบ่งกลุ่มข้อมูลออกเป็นส่วนเท่าๆ กัน ใช้ค่าเปอร์เซ็นไตล์ของขนาดเอกสารที่ 12.5, 25, 37.5, 50, 62.5, 75, 87.5 และ 100 แบ่งข้อมูลออกเป็นกลุ่มได้ 8 กลุ่ม

- การคำนวณขนาดกลุ่มตัวอย่าง ได้ขนาดข้อมูลประมาณ 5% ของข้อมูลทั้งหมดที่จะนำมาใช้เป็นตัวแทน

- นำจำนวนข้อมูลตัวอย่างที่ต้องการมาทำการสุ่มยูอาร์แอล และทำการเรียกขอข้อมูลจากอินเทอร์เน็ต ได้ข้อมูลประเภทตัวอักษร 20 เมกะไบต์ และข้อมูลประเภทรูปภาพเจเพก 40 เมกะไบต์

- การวิเคราะห์ค่าเวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูลเมื่อเกิดแคชฮิทและแคชมิสของข้อมูลประเภทตัวอักษรและรูปภาพเจเพก พบว่ามีระยะเวลาที่ต่างกันอยู่ประมาณ 5 วินาที และ 8 วินาทีตามลำดับ

- เมื่อทดลองวัดประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูลกับข้อมูลประเภทที่สามารถทำการบีบอัดได้ คือ ข้อมูลตัวอักษร และรูปภาพเจเพก จึงให้ความสำคัญกับอัตราการบีบอัดข้อมูลเพียงปัจจัยเดียว เนื่องจากเวลาที่ใช้ในการบีบอัด และขยายข้อมูลมีค่าต่ำกว่า 5 วินาที ผลสรุปที่ได้คือ

ข้อมูลตัวอักษรที่มีขนาดเล็กกว่า 7KB ควรใช้ขั้นตอนวิธี LZA ถ้ามีขนาดใหญ่กว่า 7KB ควรใช้ขั้นตอนวิธี ZIP

ข้อมูลรูปภาพเจเพกที่มีขนาดเล็กกว่า 6KB ควรใช้ขั้นตอนวิธี LZA ถ้ามีขนาดกลาง 6KB-20KB ควรใช้ขั้นตอนวิธี ZIP ถ้ามีขนาดใหญ่กว่า 20 KB ควรใช้ขั้นตอนวิธี LZA

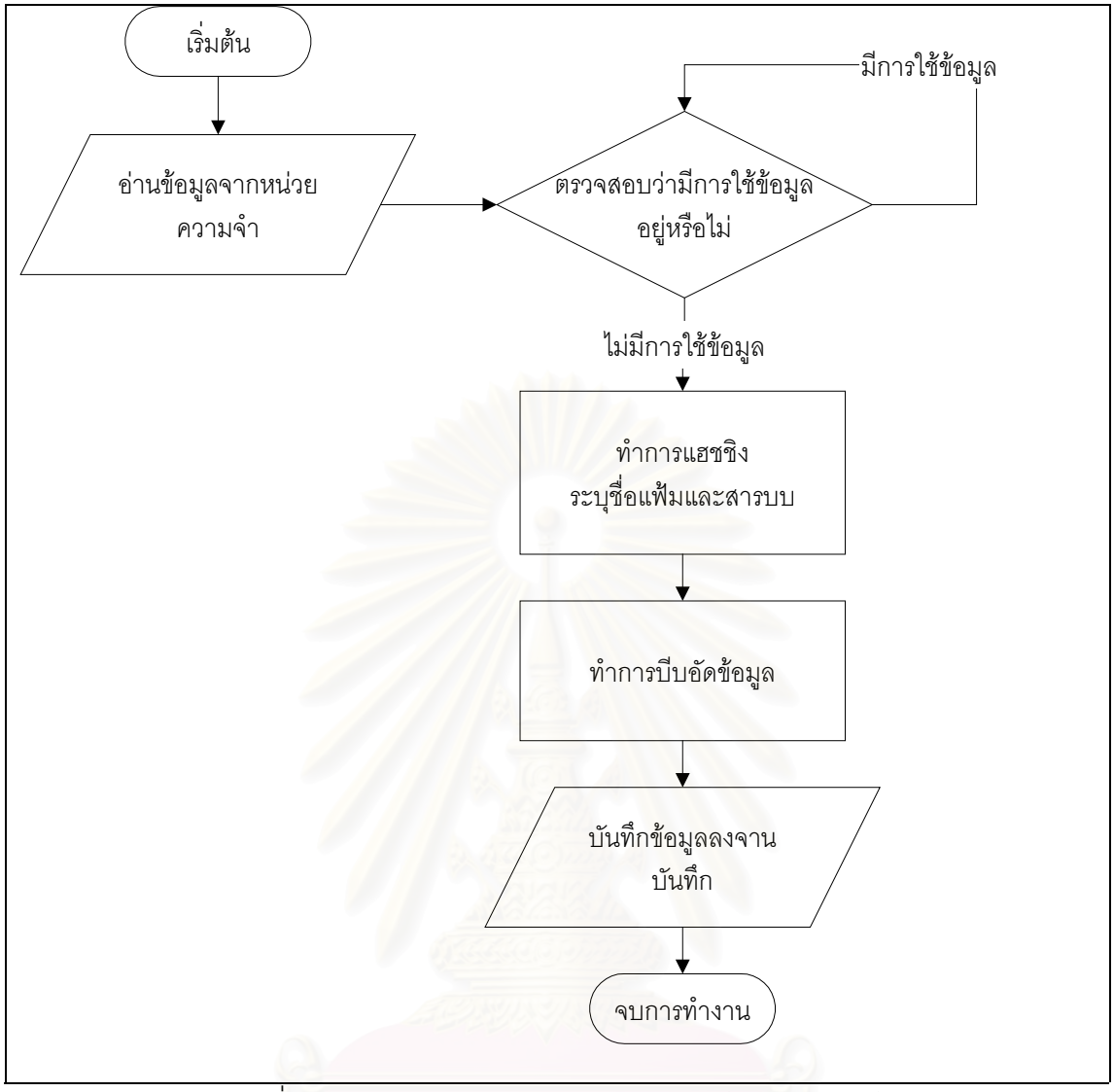
ขั้นตอนวิธีการบีบอัดข้อมูลที่ให้อัตราการบีบอัดข้อมูลที่สูงที่สุด คือ LZA และ ZIP ซึ่งทั้งสองขั้นตอนวิธีการมีการใช้แบบจำลองข้อมูลของ LZ เหมือนกันในการทำการสร้างโครงสร้างข้อมูลในขั้นที่หนึ่ง จะต่างกันในส่วนการทำงานในขั้นที่สอง ซึ่ง LZA ใช้แบบจำลองแบบเรขาคณิตเพื่อกำหนดรหัส ซึ่งใช้การจัดลำดับของความน่าจะเป็นที่จะพบตัวอักษรหรือสัญลักษณ์นั้น ซึ่งจะเหมาะกับข้อมูลที่มีขนาดเล็ก แต่ ZIP ใช้แบบจำลองแบบ Huffman ซึ่งเป็นการสร้างโครงสร้างต้นไม้ ซึ่งเหมาะเมื่อขนาดข้อมูลมีขนาดใหญ่ การเข้ารหัสโดยวิธีการนี้จะมีประสิทธิภาพดีกว่า

5.2 การประยุกต์ใช้งาน

การประยุกต์ใช้งานนั้น ผู้ใช้งานสามารถนำไปประยุกต์ใช้กับเว็บแคชที่มีอยู่ในปัจจุบันได้กับทุกผลิตภัณฑ์ ในกรณีนี้จะยกตัวอย่าง สควิดเว็บแคช

แนวคิดในการประยุกต์ใช้ขั้นตอนวิธีการบีบอัดข้อมูลกับสควิด

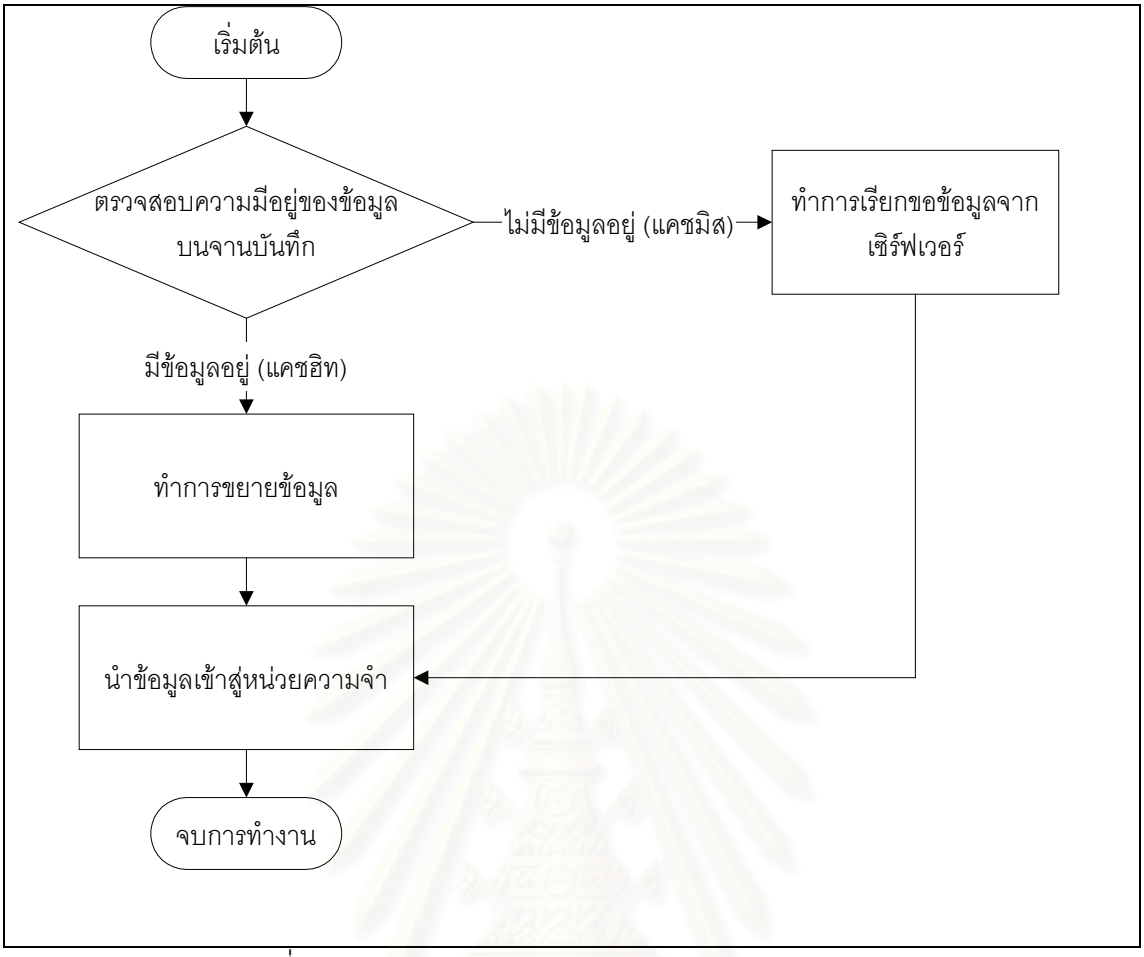
1. ในส่วนการเกิดแคชมิส นำเอกสารเว็บที่อยู่ภายในหน่วยความจำมาทำการบีบอัดข้อมูลแล้วจึงทำการบันทึกสู่จานบันทึก



ภาพที่ 5.1 การทำงานในส่วนการนำข้อมูลบันทึกลงจานบันทึก

หลังจากที่มีการทำแฮชชิง เพื่อระบุชื่อแฟ้มและสารบบแล้ว ระบบที่ดัดแปลงจะทำการตรวจขนาดของแฟ้มข้อมูลและประเภทข้อมูล เมื่อทำการเปรียบเทียบกับขั้นตอนวิธีที่เหมาะสมกับขนาดและประเภทที่ได้จากการทดลองจากบทที่ 4 จึงทำการบีบอัดข้อมูล แล้วจึงบันทึกลงจานบันทึก

2. ในส่วนการเกิดแคชฮิท ทำการตรวจสอบหาเอกสารที่ต้องการ ทำการขยายข้อมูล แล้วจึงทำการนำข้อมูลเข้าสู่หน่วยความจำ เพื่อส่งเอกสารเว็บไปยังเบราว์เซอร์ที่ทำการเรียกขอ



ภาพที่ 5.2 การทำงานในส่วนการนำข้อมูลเข้าสู่หน่วยความจำ

หลังจากที่ระบบเว็บแคชตรวจสอบว่าเอกสารมีอยู่ในแคชหรือไม่ ถ้ามีอยู่ในแคช ระบบที่ดัดแปลงจะทำการตรวจสอบว่าแฟ้มข้อมูลนั้นถูกบีบอัดโดยขั้นตอนวิธีการบีบอัดข้อมูลใดอยู่ ทำการขยายข้อมูลกลับเป็นแบบปกติ และทำการส่งข้อมูลไปยังเครื่องผู้ร้องขอ ถ้าไม่มีข้อมูลอยู่ในแคช เว็บแคชจะทำการขอข้อมูลจากเซิร์ฟเวอร์ ส่งไปยังเครื่องผู้ร้องขอ และทำการจัดเก็บไว้ในแคชตามวิธีการในข้อที่ 1 ข้างต้น

5.3 ข้อเสนอแนะในการทำวิจัย

ขั้นตอนวิธีการบีบอัดข้อมูลที่ใช้ในการวิเคราะห์ในงานวิจัยนี้ ใช้เพียง 6 ขั้นตอนวิธี ยังมีขั้นตอนวิธีอีกมากมาย เช่น ขั้นตอนวิธีบีบอัดข้อมูลที่ใช้แบบจำลองแบบเรขาคณิต (Arithmetic Coding) หรือขั้นตอนวิธีที่ดัดแปลงจากขั้นตอนวิธีการบีบอัดแบบพื้นฐาน เช่น แบบจำลองฮัฟแมนที่มีการปรับแต่ง (Adaptive Huffman Coding) ที่สามารถนำมาวิเคราะห์ประสิทธิภาพการบีบอัดเพื่อคัดเลือกหาขั้นตอนวิธีการบีบอัดที่มีความเหมาะสมที่จะนำมาประยุกต์ใช้กับเว็บแคช

นอกเหนือจากขั้นตอนวิธีการบีบอัดที่เป็นแบบไม่สูญเสียข้อมูลที่กล่าวข้างต้น อาจศึกษาเพิ่มเติมในส่วนของขั้นตอนวิธีการบีบอัดข้อมูลแบบสูญเสียข้อมูลบางส่วน (Lossy Data Compression) เพื่อนำมาประยุกต์ใช้กับข้อมูลประเภทที่ยอมรับการสูญเสียข้อมูลบางส่วน เช่น รูปภาพที่สามารถลดคุณภาพของรูปภาพให้ด้อยลง เพื่อให้ขนาดของข้อมูลเล็กลง ตัวอย่างเช่น รูปภาพเจเพก เป็นรูปภาพที่มีการลดทอนคุณภาพอยู่แล้ว การใช้ขั้นตอนบีบอัดข้อมูลแบบสูญเสียข้อมูลบางส่วนจะทำให้ภาพที่ได้มีคุณภาพต่ำลงอีก แต่จะทำให้อัตราการบีบอัดข้อมูลสูงกว่าที่ได้ทำการวิเคราะห์ไว้ในงานวิจัยนี้

รายการอ้างอิง

[1] Ghaleb Abdulla, Marc Abrams, Edward A. Fox, Chales R. Standridge, and Stephen Williams. Caching Proxies : Limitations and potentials. Proceedings of 4th International World Wide Web Conference (1995).

[2] Cao, P., and Irani, S., Cost-Aware WWW Proxy Caching Algorithms. Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS) (December 1997).

[3] Alex Rousskov, Valery Soloviev, and Igor Tatarinov. Static Caching : Position Paper. Proceedings of 2nd International Web Caching Workshop (1997).

[4] Alex Rousskov, Valery Soloviev, and Igor Tatarinov. Static Caching in Web Servers. Proceedings of 6th IEEE International Conference on Computer Communications and Networks (1997).

[5] Ulana Legedza, John Guttag. Using Network-level Support to Improve Cache Routing. Proceedings of 3rd International Web Caching Workshop (1998).

[6] Michel Baentsch, L. Baum, Georg Molter, S. Rothkugel, and P. Sturm. World Wide Web Caching : The Application-Level View of the Internet. IEEE Communications Magazine June 1997.

[7] Berger, D., Lorch, L., and Radhika Malpani. Making World Wide Web Caching Servers Cooperate. Proceedings of 4th International World Wide Web Conference (1995).

[8] Hari Balakrishnan, Randy. H. Katz, Venkata. N. Padmanabhan, Srinivassan. Seshan, Mark Stemm. TCP behavior of a busy internet server: Analysis and improvements. Proceedings of IEEE Infocom (1998).

[9] พรทวี วัฒนวิทูกูร. การวิเคราะห์เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการแทนที่ในพร็อกซีแคช. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, (2542).

[10] Duane Wessels. Information Resource Caching FAQ. Available from: http://ircache.nlanr.net/Cache/FAQ/ircache-faq.html.

[11] Mark Nelson. The Data Compression Book : Featuring fast, efficient data compression techniques in C. U.S.A. : Prentice-Hall International, Inc., 1991.

[12] Shaun Case. Available from: mail://atman%ecst.csuchico.edu@RELAY.CS.NET.

[13] Pkware, Available from: http://www.pkware.com.

[14] Haruhiko Okumura. Available from: mail://okumura@matsusaka-u.ac.jp.

[15] LZO, Available from: http://www.infosys.tuwien.ac.at/Staff/lux/marco/lzo.html.

[16] Squid Web Cache, Available from: http://www.squid-cache.org.

[17] Calamaris, Available from: http://cord.de/tools/squid/calamaris.

[18] ประชุม สุวัตถี. การวิเคราะห์เชิงสถิติ เล่ม 1. คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์. (ม.ป.ป).

[19] Erik de Neve. Ultra Precision Command Timer [Computer program].

ภาคผนวก

ภาคผนวก ก

ผลการวิเคราะห์สภาพการใช้เว็บ

จากแฟ้มข้อมูลบันทึกการใช้งานวันที่ 3 ตุลาคม ถึง 1 พฤศจิกายน พ.ศ. 2542 รวบรวมโดยสำนักเทคโนโลยีสารสนเทศ จุฬาลงกรณ์มหาวิทยาลัย โดยโปรแกรมวิเคราะห์แฟ้มบันทึกการใช้งาน คาลามาริส โดยมีการตัดทอนผลในส่วน Incoming TCP-requests by host และ Performance in 60 minute steps

Proxy Report

03.Oct 99 00:00:10 – 01.Nov 99 00:00:02

Summary

lines parsed:	4998675
invalid lines:	0
unique hosts/users:	656
parse time (sec):	3523

Incoming requests by method

method	Request	%	kByte	%	sec	kB/sec
GET	4385977	87.74	29081463	98.38	8	0.79
ICP_QUERY	509101	10.18	35840	0.12	0	703.99
POST	85850	1.72	404144	1.37	15	0.31
CONNECT	13783	0.28	36005	0.12	16	0.16
HEAD	3816	0.08	1269	0.00	6	0.05
NONE	148	0.00	245	0.00	0	9.94
Sum	4998675	100.00	29558968	100.00	7	0.77

Incoming UDP-requests by status

status	request	%	kByte	%	msec	kB/sec
HIT	127477	25.04	8554	23.87	0	671.05
UDP_HIT	127477	25.04	8554	23.87	0	671.05
MISS	381624	74.96	27285	76.13	0	715.00
UDP_MISS	381624	74.96	27285	76.13	0	715.00
Sum	509101		35840		0	703.99

Incoming TCP-requests by status

status	request	%	kByte	%	sec	kB/sec
HIT	2213282	49.30	9149502	30.99	1	2.14
TCP_HIT	912746	20.33	6482107	21.96	3	2.32
TCP_IMS_HIT	883588	19.68	395052	1.34	0	7.07
TCP_REFRESH_HIT	343900	7.66	2133809	7.23	4	1.51
TCP_MEM_HIT	48327	1.08	115875	0.39	0	33.37
TCP_NEGATIVE_HIT	24721	0.55	22656	0.08	0	24.26

MISS	2073366	46.18	20041280	67.88	16	0.59
TCP_MISS	2026301	45.13	19624623	66.47	16	0.59
TCP_REFRESH_MISS	46965	1.05	416535	1.41	18	0.48
TCP_SWAPFAIL_MISS	100	0.00	122	0.00	5	0.21
ERROR	202926	4.52	332345	1.13	0	22.97
NONE	199567	4.45	329426	1.12	0	22.97
TCP_DENIED	1912	0.04	2066	0.01	0	23.99
TCP_MISS	1018	0.02	327	0.00	0	12.21
TCP_REFRESH_MISS	417	0.01	515	0.00	0	36.24
TCP_SWAPFAIL_MISS	12	0.00	9	0.00	0	92.27
Sum	4489574		29523128		8	0.77

Outgoing requests by status

status	request	%	kByte	%	sec	kB/sec
DIRECT Fetch from Source	2390807	98.91	21893132	98.73	14	0.62
DIRECT	2026472	83.83	17759908	80.09	13	0.63
TIMEOUT_DIRECT	364335	15.07	4133223	18.64	18	0.60
HIT on Sibling or Parent Cache	26459	1.09	281957	1.27	8	1.27
SIBLING_HIT	20121	0.83	216867	0.98	9	1.12
CACHE_DIGEST_HIT	6338	0.26	65090	0.29	4	2.26
PARENT	0	0.00	0	0.00	0	0.00
Sum	2417266		22175090		14	0.63

Outgoing requests by destination

neighbor type	request	%	kByte	%	sec	kB/sec
DIRECT	2390807	98.91	21893132	98.73	14	0.62
161.200.255.162	16058	0.66	177519	0.80	0	0.99

SIBLING_HIT	16058	0.66	177519	0.80	11	0.99
proxy.acc.chula.ac.th	6842	0.28	68247	0.31	0	2.31
CACHE_DIGEST_HIT	6338	0.26	65090	0.29	4	2.26
SIBLING_HIT	504	0.02	3157	0.01	1	4.04
proxy.eng.chula.ac.th	3559	0.15	36190	0.16	0	2.57
SIBLING_HIT	3559	0.15	36190	0.16	3	2.57
Sum	2417266		22175090		14	0.63

Request-destinations by 2ndlevel-domain

destination	request	%	kByte	%	hit-%
<error>	298363	6.65	447162	1.51	8.30
***.chula.ac.th**	267624	5.96	1473364	4.99	57.10
***.inet.co.th**	150237	3.35	537699	1.82	54.38
***.sanook.com**	149435	3.33	829436	2.81	91.61
***.geocities.com**	115275	2.57	1867038	6.32	35.83
***.msn.com**	99187	2.21	850851	2.88	27.96
***.hunsa.com**	84229	1.88	226894	0.77	49.37
***.yahoo.com**	63812	1.42	315945	1.07	3.56
***.phonelink.net**	63542	1.42	140256	0.48	86.50
***.yimg.com**	55023	1.23	146003	0.49	74.79
***.icq.com**	45110	1.00	196879	0.67	68.68
***.doubleclick.net**	42326	0.94	111736	0.38	24.07
***.hypermart.net**	40027	0.89	366071	1.24	61.60
***.hitbox.com**	36914	0.82	69603	0.24	30.94
***.imgis.com**	35655	0.79	63346	0.21	23.01
***.bluemountain.com**	35542	0.79	122875	0.42	66.46
***.mthai.com**	35333	0.79	241102	0.82	69.45
***.xoom.com**	35201	0.78	943029	3.19	50.89
***.thaimail.com**	33319	0.74	89531	0.30	56.71

***.dynamix.net**	32711	0.73	349233	1.18	69.70
other: 17619 2nd-level-domains	2770709	61.71	20135065	68.20	51.06
Sum	4489574	100.00	29523128	100.00	49.30

Request-destinations by toplevel-domain

destination	request	%	kByte	%	hit-%
***.com**	2665170	59.36	19037674	64.48	50.31
***.th**	718882	16.01	3669222	12.43	61.66
***.net**	391005	8.71	2185231	7.40	55.41
<error>	298363	6.65	447162	1.51	8.30
<unresolved>	194311	4.33	1486266	5.03	58.27
***.org**	33982	0.76	366608	1.24	39.08
***.edu**	32290	0.72	349912	1.19	37.26
***.jp**	30950	0.69	365588	1.24	21.80
***.uk**	20243	0.45	133933	0.45	43.76
***.to**	14025	0.31	64142	0.22	39.18
***.sg**	12916	0.29	56752	0.19	72.39
***.de**	10804	0.24	140672	0.48	11.57
***.it**	7478	0.17	63499	0.22	31.43
***.dk**	6138	0.14	43941	0.15	29.62
***.gov**	5918	0.13	109785	0.37	25.79
***.nu**	4968	0.11	35746	0.12	34.66
***.au**	4468	0.10	57628	0.20	15.78
***.fr**	3394	0.08	148258	0.50	12.43
***.tw**	3001	0.07	60249	0.20	26.79
***.se**	2658	0.06	44452	0.15	21.71
other: 201 top-level-domains	28610	0.64	656398	2.22	26.86
Sum	4489574	100.00	29523128	100.00	49.30

TCP-Request-protocol

protocol	request	%	kByte	%	hit-%
http:	4174370	92.98	27547393	93.31	52.41
<error>	298363	6.65	447162	1.51	8.30
<secure>	13728	0.31	35930	0.12	0.00
ftp:	3108	0.07	1492573	5.06	28.73
gopher:	5	0.00	68	0.00	0.00
Sum	4489574	100.00	29523128	100.00	49.30

Requested content-type

content-type	request	%	kByte	%	hit-%
image/gif	2228159	49.63	7084085	24.00	74.00
text/html	883947	19.69	5937614	20.11	16.28
image/jpeg	547702	12.20	5483615	18.57	49.75
<unknown>	381065	8.49	77280	0.26	20.58
<error>	297737	6.63	445073	1.51	8.29
text/plain	42317	0.94	916817	3.11	13.00
application/x-javascript	29975	0.67	58231	0.20	31.22
text/css	18101	0.40	22640	0.08	69.98
application/octet-stream	16594	0.37	3845962	13.03	51.27
<secure>	13728	0.31	35930	0.12	0.00
application/zip	5396	0.12	1634697	5.54	10.04
audio/midi	3067	0.07	60179	0.20	49.43
audio/x-midi	1780	0.04	19842	0.07	38.82
audio/x-pn-realaudio	1643	0.04	214948	0.73	43.64
image/pjpeg	1617	0.04	45942	0.16	11.69
application/x-shockwave-flash	1541	0.03	73659	0.25	51.78

application/pdf	1182	0.03	451135	1.53	11.34
image/x-xbitmap	1107	0.02	1118	0.00	0.27
application/postscript	1016	0.02	581166	1.97	6.20
application/cache-digest	908	0.02	147216	0.50	100.00
other: 190 content-types	10992	0.24	2385968	8.08	29.18
Sum	4489574	100.00	29523128	100.00	49.30

Requested extensions

extensions	request	%	kByte	%	hit-%
gif	2208355	49.19	6383141	21.62	73.90
<dynamic>	630159	14.04	2748070	9.31	3.00
jpg	557053	12.41	5075687	17.19	48.87
<error>	298363	6.65	447162	1.51	8.30
<none>	211179	4.70	1534395	5.20	24.66
html	171429	3.82	1352945	4.58	37.41
htm	119722	2.67	520363	1.76	34.99
GIF	65820	1.47	261440	0.89	77.23
shtml	31460	0.70	452033	1.53	0.44
pl	29265	0.65	88585	0.30	0.01
JPG	26176	0.58	301658	1.02	44.16
css	21108	0.47	25898	0.09	69.15
asp	14970	0.33	164476	0.56	0.06
js	14692	0.33	41859	0.14	59.07
<secure>	13728	0.31	35930	0.12	0.00
cgi	9201	0.20	43208	0.15	2.45
class	7152	0.16	160076	0.54	53.55
zip	6466	0.14	1988415	6.74	12.31
mid	5543	0.12	83819	0.28	44.60
exe	3037	0.07	2401747	8.14	21.11

other: 1185 extensions	44696	1.00	5412211	18.33	30.05
Sum	4489574	100.00	29523128	100.00	49.30

Incoming UDP-requests by host

host	request	hit-%	kByte	hit-%	msec	kB/sec
student.eng.chula.ac.th	325821	38.06	22765	36.56	0	698.70
Phoenix.acc.chula.ac.th	183280	1.90	13075	1.77	0	713.41
Sum	509101	25.04	35840	23.87	0	703.99

Incoming TCP-requests by host

host	request	hit-%	kByte	hit-%	sec	kB/sec
Phoenix.acc.chula.ac.th	271882	69.19	1620777	51.61	5	1.14
161.200.58.11	148274	47.70	674309	36.17	4	1.05
student.eng.chula.ac.th	121501	99.74	1013304	99.83	0	19.38
161.200.118.223	96137	48.68	430194	29.75	4	0.99
161.200.144.33	49390	58.96	170688	51.01	4	0.82
161.200.192.92	36031	50.13	154735	24.98	7	0.61
161.200.144.47	32294	51.46	131764	46.88	7	0.54
161.200.49.32	31457	65.31	100100	41.55	4	0.67
mail.sc.chula.ac.th	1	0.00	0	0.00	32	0.01
nontri.ku.ac.th	1	0.00	1	0.00	0	23.12
67_54.acc.chula.ac.th	1	100.00	0	100.00	0	23.00
203.149.11.94	1	0.00	1	0.00	0	38.04
Sum	4489574	49.30	29523128	30.99	8	0.77

Performance in 60 minute steps

			incomin	hit	miss	direct	sibling	fetch
date	**request**	**MByte**	**kB/sec**	**kB/sec**	**kB/sec**	**kB/sec**	**kB/sec**	**kB/sec**
03.Oct 99 00:00	12451	76	1.81	3.21	1.52	1.54	1.85	
03.Oct 99 01:00	14808	105	1.38	2.89	1.12	1.16	1.22	
03.Oct 99 02:00	11631	90	1.04	2.29	0.86	0.88	1.05	
03.Oct 99 03:00	7022	75	1.16	2.07	1.00	1.07	0.79	
31.Oct 99 22:00	9187	55	1.28	2.96	1.05	1.10	1.53	
31.Oct 99 23:00	9522	62	1.29	2.50	0.95	1.07	0.86	
01.Nov 99 00:00	7	0	2.36	19.30	0.44	0.44		
overall	4489574	28831	0.77	2.14	0.59	0.62	1.27	

ภาคผนวก ข
ผลการวิเคราะห์เวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูลกรณีที่เกิดแคชฮิท และแคชมิส

1. ข้อมูลประเภทตัวอักษร

ตารางที่ ข.1 ค่าเวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล กรณีที่เกิดแคชฮิทและแคชมิสของข้อมูลประเภทตัวอักษร

กลุ่มข้อมูล	เวลาเฉลี่ยของการเรียกขอเอกสาร เมื่อแคชฮิท	เวลาเฉลี่ยของการเรียกขอเอกสาร เมื่อแคชมิส
1	283.85	2,823.79
2	282.11	2,998.59
3	244.04	2,828.24
4	895.08	5,829.51
5	722.94	4,816.46
6	1,188.65	7,140.96
7	6,085.10	8,810.90
8	10,770.50	25,817.50
ALL	2,137.17	7,994.08

2. **ข้อมูลประเภทรูปภาพเจเพก**

ตารางที่ ข.2 ค่าเวลาเฉลี่ยที่ใช้ในการเรียกขอข้อมูล กรณีที่เกิดแคชฮิทและแคชมิสของข้อมูลประเภทรูปภาพเจเพก

กลุ่มข้อมูล	เวลาเฉลี่ยของการเรียกขอเอกสาร เมื่อแคชฮิท	เวลาเฉลี่ยของการเรียกขอเอกสาร เมื่อแคชมิส
1	283.07	3,337.90
2	169.45	2,796.17
3	263.57	3,536.11
4	340.02	3,631.64
5	583.52	3,498.71
6	776.68	5,052.49
7	4,262.45	8,562.54
8	14,826.70	39,786.40
ALL	2,230.36	9,719.51

ภาคผนวก ค
ผลการวิเคราะห์ประสิทธิภาพของขั้นตอนวิธีการบีบอัดข้อมูล

1. ผลการวิเคราะห์ประสิทธิภาพขั้นตอนวิธีการบีบอัดข้อมูลในส่วนอัตราการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

ขั้นตอนวิธีการบีบอัดข้อมูล	กลุ่มข้อมูล							
	1	2	3	4	5	6	7	8
Huffman	0.3372	0.6174	0.7662	0.9857	1.1361	1.2444	1.3242	1.4113
LZA	1.3751	1.8194	2.1294	2.6751	3.1976	3.3162	3.7924	4.5162
LZH	1.3432	1.794	2.1063	2.6514	3.1746	3.2952	3.7721	4.4923
LZO	0.9176	1.3365	1.5826	1.9793	2.3683	2.5074	2.9584	3.6598
LZS	1.1933	1.5106	1.7195	2.1375	2.542	2.6335	2.9045	3.2795
ZIP	0.8858	1.5585	1.9273	2.5574	3.1808	3.4496	4.1651	5.3044

2. ผลการวิเคราะห์ประสิทธิภาพขั้นตอนวิธีการบีบอัดข้อมูลในส่วนเวลาที่ใช้ในการบีบอัดข้อมูลของข้อมูลประเภทตัวอักษร

ขั้นตอนวิธีการบีบอัดข้อมูล	กลุ่มข้อมูล							
	1	2	3	4	5	6	7	8
Huffman	0.6296	0.6656	0.7902	0.7743	0.8638	0.9602	1.2491	2.182
LZA	0.5477	0.5307	0.6575	0.6708	0.7667	0.8752	1.1495	2.2463
LZH	0.5426	0.5192	0.6416	0.6579	0.7349	0.8634	1.1405	2.2128
LZO	1.2219	1.2212	1.3092	1.3363	1.2965	1.5731	2.048	2.8853
LZS	0.5513	0.5438	0.6411	0.6145	0.6135	0.7779	1.1748	2.0185
ZIP	0.9832	1.0857	1.126	1.0652	1.0034	1.1373	1.4299	1.8791

3. ผลการวิเคราะห์ประสิทธิภาพขั้นตอนวิธีการบีบอัดข้อมูลในส่วนเวลาที่ใช้ในการขยายข้อมูลของข้อมูลประเภทตัวอักษร

ขั้นตอนวิธีการบีบอัดข้อมูล	กลุ่มข้อมูล							
	1	2	3	4	5	6	7	8
Huffman	0.4696	0.5635	0.6824	0.6177	0.1949	0.4591	1.2128	2.2639
LZA	0.7514	0.747	0.8478	0.8571	0.766	0.8441	1.095	1.9029
LZH	0.7432	0.7305	0.8277	0.8561	0.762	0.8205	1.0814	1.9046
LZO	1.578	1.7249	1.9064	1.9106	1.5833	1.7936	2.067	3.321
LZS	0.6889	0.7049	0.7923	0.8083	0.7267	0.888	1.3164	2.1516
ZIP	0.7582	0.9679	1.0652	0.7214	0.5596	0.8907	1.7024	2.114

4. ผลการวิเคราะห์ประสิทธิภาพขั้นตอนวิธีการบีบอัดข้อมูลในส่วนอัตราการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ขั้นตอนวิธีการบีบอัดข้อมูล	กลุ่มข้อมูล							
	1	2	3	4	5	6	7	8
Huffman	0.3398	0.4434	0.4872	0.558	0.6646	0.7413	0.8377	0.9321
LZA	1.2485	1.1129	1.0542	1.0493	1.1521	1.1113	1.0636	1.039
LZH	1.2375	1.1089	1.05	1.0453	1.1473	1.1072	1.0604	1.0363
LZO	1.0816	1.0793	1.0313	1.0329	1.0976	1.0998	1.0562	1.0277
LZS	1.1124	1.009	0.9579	0.9538	1.0286	0.9975	0.9527	0.9271
ZIP	1.0576	1.0561	1.0153	1.0236	1.1358	1.1207	1.0647	1.0388

5. ผลการวิเคราะห์ประสิทธิภาพขั้นตอนวิธีการบีบอัดข้อมูลในส่วนเวลาที่ใช้ในการบีบอัดข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ขั้นตอนวิธีการบีบอัดข้อมูล	กลุ่มข้อมูล							
	1	2	3	4	5	6	7	8
Huffman	0.6725	0.6394	0.6415	0.6661	0.8188	0.9018	1.1323	2.6723
LZA	0.5314	0.4989	0.5115	0.5442	0.6962	0.7674	1.0409	2.7891
LZH	0.5063	0.468	0.481	0.5083	0.6762	0.7131	0.9662	2.5543
LZO	1.0118	0.7498	0.7312	0.7226	1.0579	1.0324	1.3745	2.7564
LZS	0.5223	0.4794	0.4865	0.5159	0.655	0.732	0.9898	2.5779
ZIP	0.7476	0.5566	0.5585	0.5552	0.7403	0.7297	0.9148	2.4079

6. ผลการวิเคราะห์ประสิทธิภาพขั้นตอนวิธีการบีบอัดข้อมูลในส่วนเวลาที่ใช้ในการขยายข้อมูลของข้อมูลประเภทรูปภาพเจเพก

ขั้นตอนวิธีการบีบอัดข้อมูล	กลุ่มข้อมูล							
	1	2	3	4	5	6	7	8
Huffman	0.4516	0.6285	0.6464	0.6792	0.8321	0.886	1.135	2.9672
LZA	0.6848	0.5906	0.6382	0.6709	0.8635	0.9119	1.2448	3.2566
LZH	0.6699	0.5343	0.5855	0.6346	0.836	0.9714	1.1819	3.1754
LZO	1.2433	0.941	0.9159	0.9177	1.3527	1.3242	1.653	3.692
LZS	0.6366	0.5355	0.5598	0.6035	0.8237	0.8667	1.2308	3.3717
ZIP	0.5637	0.4663	0.4737	0.4818	0.5882	0.5787	0.6427	1.3425

ประวัติผู้เขียนวิทยานิพนธ์

นายกัลย์ แก้วแก่น เกิดเมื่อวันเสาร์ที่ 1 กันยายน พ.ศ. 2516 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตรบัณฑิต ภาควิชาปฐพีวิทยา คณะเทคโนโลยีการเกษตร สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อปี พ.ศ. 2537 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2540