

การวิเคราะห์เปรียบเทียบกลวิธีในการลดความยาวของ URL

นายประเสริฐ วิชชุโอภาส



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2544

ISBN 974-03-0381-1

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

# A COMPARATIVE STUDY OF ALGORITHMS FOR REDUCING URL LENGTH

Mr. Prasert Vitthu-o-pas

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

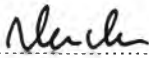
Academic Year 2001

ISBN 974-03-0381-1

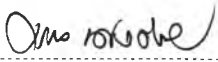
หัวข้อวิทยานิพนธ์ การวิเคราะห์เปรียบเทียบกลวิธีในการลดความยาวของ URL  
โดย นายประเสริฐ วิชชุโสภาส  
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์  
อาจารย์ที่ปรึกษา อาจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์


---

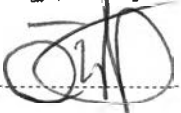
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

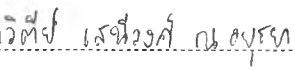
  
..... คณบดีคณะวิศวกรรมศาสตร์  
(ศาสตราจารย์ ดร. สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

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

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

  
..... กรรมการ  
(รองศาสตราจารย์ ดร. วันชัย รั้วไพบูลย์)

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

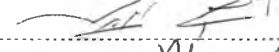
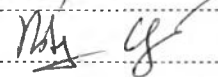
ประเสริฐ วิชาอุภาส : การวิเคราะห์เปรียบเทียบกลวิธีในการลดความยาวของยูอาร์แอล (THE COMPARATIVE STUDY OF ALGORITHMS FOR REDUCING URL LENGTH)

อ. ที่ปรึกษา : ดร. ณัฐวุฒิ หนูไพโรจน์, 68 หน้า, ISBN 974-03-0381-1

การทำงานของพรีอ็อกซีแคชเกี่ยวข้องกับการประมวลผลยูอาร์แอลเป็นจำนวนมาก ทั้งการระบุว่ายูอาร์แอลที่เครื่องลูกข่ายต้องการอยู่ในแคชของตนหรือไม่ และการสอบถามข้อมูลระหว่างพรีอ็อกซีแคชด้วยกัน ล้วนแล้วแต่ต้องประมวลผลยูอาร์แอลทั้งสิ้น ถ้าหากสามารถเพิ่มประสิทธิภาพโดยรวมของการประมวลผลยูอาร์แอลก็จะเป็นการเพิ่มประสิทธิภาพพรีอ็อกซีแคชด้วยเช่นกัน จากการศึกษาในเบื้องต้นพบว่า การเข้ารหัสยูอาร์แอลที่ใช้งานในการทำงานพรีอ็อกซีแคช มีความต้องการคุณสมบัติของวิธีการเข้ารหัสที่แตกต่างกัน พรีอ็อกซีแคชในปัจจุบันมีการใช้วิธีการเข้ารหัสที่ชื่อว่า MD5 ในการย่อยูอาร์แอลก่อนนำไปประมวลผลหรือสอบถามข้อมูลระหว่างพรีอ็อกซีแคชด้วยกัน แต่ยังไม่มียานวิจัยใดที่ทำการศึกษาวิธีการเข้ารหัสอื่นๆ ดังนั้นในงานวิจัยนี้จึงได้เลือกวิธีการเข้ารหัสแบบต่างๆ ขึ้นมาจำนวนหนึ่ง ทำการเข้ารหัสยูอาร์แอลเพื่อเปรียบเทียบ เวลาที่ใช้ในการเข้ารหัส ความยาวรหัส และปริมาณการชนกันของรหัส ของแต่ละวิธีการ และนำมาใช้ในการเสนอแนะแนวทางในการเลือกใช้วิธีการเข้ารหัสยูอาร์แอลที่มีประสิทธิภาพ สำหรับพรีอ็อกซีแคช

ข้อมูลยูอาร์แอลที่ใช้ในการทดสอบนำมาจากข้อมูลการใช้เว็บของจุฬาลงกรณ์มหาวิทยาลัย ข้อมูลนี้ถูกนำมาเข้ารหัสด้วยวิธีการเข้ารหัสต่างๆ ซึ่งเลือกขึ้นมา 12 วิธีการ โดยวิธีการเข้ารหัสเหล่านั้นเป็นที่รู้จักกันดี รวมทั้งวิธีการ MD5 ผลการทดลองทำให้ทราบว่าวิธีการเข้ารหัสที่มีความซับซ้อนน้อยใช้เวลาในการเข้ารหัสน้อยกว่าวิธีการที่มีความซับซ้อนมากและมีโอกาสที่เกิดการชนของรหัสมากกว่า วิธีการเข้ารหัสที่มีขนาดของรหัสที่สั้นมีโอกาสเกิดการชนกันของผลลัพธ์มากกว่า วิธีการที่เลือกมาทดสอบในการวิจัยไม่มีวิธีการเข้ารหัสใดที่ดีที่สุด การพิจารณาว่าวิธีการใดเหมาะสมจะพิจารณาจากความต้องการของแอปพลิเคชันเป็นหลัก โดยแอปพลิเคชันที่ต้องการความเร็วในการเข้ารหัสและรหัสที่สั้น ควรเลือกวิธีการ CRC-16, Digit Analysis method, Folding method แอปพลิเคชันที่ต้องการความเร็วในการเข้ารหัสและไม่ต้องการให้มีการชนกันของรหัสเลย ควรเลือกวิธีการในกลุ่ม MD หรือ Huffman Coding แต่ถ้าหากยอมให้มีการชนกันของรหัสได้บ้าง ควรเลือกวิธีการ CRC-32 หรือ Folding method และสำหรับแอปพลิเคชันที่ต้องการรหัสที่สั้นและมีปริมาณการชนกันของรหัสน้อย ควรเลือกวิธีการ CRC-32, Division method และ Folding method นอกจากนี้ ยังพบว่าวิธีการ CRC-32 และ Folding method ใช้เวลาในการเข้ารหัสน้อย รหัสที่ได้มีขนาดสั้น และมีการชนกันของรหัสน้อยมาก จึงอาจปรับปรุงโดยการรวมวิธีการทั้งสองเข้าด้วยกันเพื่อหาวิธีการใหม่ที่มีประสิทธิภาพมากขึ้นได้

ภาควิชา วิศวกรรมคอมพิวเตอร์.....  
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์.....  
ปีการศึกษา 2544.....

ลายมือชื่อนิสิต   
ลายมือชื่ออาจารย์ที่ปรึกษา   
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

## 4171455821 : MAJOR COMPUTER SCIENCE

KEY WORD: WWW / PROXY CACHING / MESSAGE DIGEST ALGORITHM / URL

PRASERT VITCHU-O-PAS : THE COMPARATIVE STUDY OF ALGORITHMS FOR REDUCING  
URL LENGTH. THESIS ADVISOR : NATAWUT NUPAIROJ, Ph. D. 68 pp.

ISBN 974-03-0381-1

Webs Proxy Caches usually process large amounts of URLs to identify if requested pages are in the caches and to communicate among shared Web Caches. Improving URL processing can greatly enhance the performance of Web Caches. The preliminary study of this thesis shows that the requirements of URL processing for web caching application are varied.

Typically, many Web Proxy Caches use a hashing function, MD5, to encode URL. However, there is no study of a suitable algorithm for URL. In this study, we choose some well-known encoding or hashing algorithms, use them to encode URL in web access log and compare their encoding time, encoded length, and number of collisions. The comparison of the result is used as a guideline to select suitable URL encoding algorithm.

In our experiment, 12 well known coding and hashing algorithms include MD5 are selected to encode the cacheable URL from web access log of Office of Information Technology, Chulalongkorn University. We found that complicated algorithms use more time to encode URL but their encoded keys have less number of collisions. In addition, we found that algorithms that generate shorter length encoded keys lead to more number of collisions. Our studies indicate that there is no algorithm that can be considered the best for every Web Cache Applications. The algorithm that is suitable for each Web Cache Application is determined by their core requirements. For example, applications that speed and key length are critical should use CRC-16, Digit Analysis method, or Folding method as their main coding algorithm. Applications that require no collision and focus on fast speed should use a MD2, MD4, or MD5 algorithm. However, if some collisions are allowed, CRC-32 and Folding method are suitable. For the applications that require short length key and tolerate some collisions, CRC-32, Division method and Folding method should be selected. We also found that the CRC-32 and Folding method are faster than other algorithms and their encoded keys have very low collision rates. Therefore, combining these 2 algorithms may create a new powerful algorithm for Web Cache Application.

Department Computer Engineering .....

Field of study Computer Science .....

Academic year 2001 .....

Student's signature  .....

Advisor's signature  .....

Co-advisor's signature .....



## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความช่วยเหลืออย่างดียิ่งของ ดร. ณัฐวุฒิ หนูไพโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆในการวิจัยมาด้วยดีตลอด

ทำยนี้ ผู้วิจัยใคร่ขอขอบพระคุณ บิดา – มารดา เพื่อนร่วมงาน ซึ่งให้ข้อคิดเห็น และสนับสนุน ผู้วิจัยจนสำเร็จการศึกษา

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ .....	ฉ
สารบัญ.....	ช
สารบัญตาราง .....	ญ
สารบัญรูปภาพ.....	ฎ
บทที่	
1 บทนำ .....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 ความสำคัญของยูอาร์แอล .....	2
1.3 ปัญหาในการวิเคราะห์ยูอาร์แอล .....	3
1.4 แนวคิดในการวิจัย.....	5
1.5 งานวิจัยที่เกี่ยวข้อง.....	6
1.6 ขอบเขตการวิจัย.....	7
1.7 วัตถุประสงค์.....	8
1.8 ประโยชน์ที่จะได้รับ .....	8
1.9 โครงสร้างวิทยานิพนธ์.....	8
2 แนวความคิดที่ใช้และทฤษฎีที่เกี่ยวข้อง.....	10
2.1 การเข้ารหัสแบบ Cyclic Redundancy Check .....	11
2.2 การเข้ารหัสแบบฮัฟแมน (Huffman Coding) .....	11
2.3 ฟังก์ชันแฮช (Hashing Function).....	12
2.3.1 ฟังก์ชันแฮชคืออะไร .....	12
2.3.2 อัลกอริทึมในกลุ่ม MD (Message Digest Algorithm) .....	14
2.3.3 อัลกอริทึมแฮชอย่างง่าย.....	15

## สารบัญ ( ต่อ )

บทที่	หน้า
3 แนวทางการวิจัย.....	17
3.1 ขั้นตอนในการทำวิจัย .....	18
3.2 การพิจารณาเลือกใช้ข้อมูล .....	19
3.2.1 ลักษณะเฉพาะของยูอาร์แอล.....	19
3.2.2 การกำจัดข้อมูลที่มีการซ้ำกัน.....	19
3.2.3 การคัดเลือกยูอาร์แอล.....	20
3.3 วิธีทำการทดลอง.....	20
3.4 การพิจารณาเลือกภาษาโปรแกรม.....	23
3.5 การทำการทดลองซ้ำ .....	23
3.6 ผลการทดลอง.....	23
4 ผลการทดลอง .....	25
4.1 สภาพแวดล้อมในการทดลอง .....	25
4.2 ยูอาร์แอลในการทดลอง.....	25
4.2.1 จำนวนและความยาวยูอาร์แอล .....	26
4.2.2 ฮิสโตแกรมของความยาวยูอาร์แอล.....	27
4.3 ผลการทดลองเข้ารหัสยูอาร์แอล แยกตามอัลกอริทึม .....	29
4.3.1 ผลการทดลองเข้ารหัสของอัลกอริทึมในกลุ่ม MD .....	29
4.3.2 ผลการทดลองเข้ารหัสยูอาร์แอลด้วยอัลกอริทึมในกลุ่ม CRC .....	31
4.3.3 ผลการทดลองเข้ารหัสยูอาร์แอลด้วยฟังก์ชันแฮชอย่างง่าย .....	33
4.3.3.1 Digit Analysis Method.....	33
4.3.3.2 Folding Method.....	35
4.3.3.3 Midsquare Method.....	36
4.3.3.4 Division Method .....	37
4.3.4 ผลการทดลองเข้ารหัสยูอาร์แอลด้วยอัลกอริทึม Huffman Coding.....	39
5 วิเคราะห์ผลการทดลอง.....	40
5.1 ความยาวและลักษณะของยูอาร์แอล.....	40
5.2 ความเร็วในการเข้ารหัส .....	41
5.3 ความยาวรหัสและประสิทธิภาพในการลดขนาดยูอาร์แอล .....	44



## สารบัญ ( ต่อ )

บทที่	หน้า
5.4 การชนกันของผลลัพท์.....	48
5.5 สรุปผลการทดลอง .....	49
6 สรุปการวิจัย.....	55
6.1 สรุปผลการวิจัย .....	55
6.2 ปัญหาและข้อจำกัดที่พบในการวิจัย .....	57
6.3 ข้อเสนอแนะ .....	57
รายการอ้างอิง .....	58
ภาคผนวก .....	60
ภาคผนวก ก.....	61
ภาคผนวก ข.....	62
ภาคผนวก ค .....	65
ประวัติผู้วิจัย.....	68

## สารบัญตาราง

	หน้า
ตารางที่ 3.1 ปริมาณการเรียกขอของข้อมูลที่นำมาใช้ในการวิจัย.....	19
ตารางที่ 4.1 ผลการวิเคราะห์ความยาวยูอาร์แอล.....	26
ตารางที่ 4.2 เวลาที่ใช้ในการเข้ารหัสด้วยอัลกอริทึม MD2.....	30
ตารางที่ 4.3 เวลาที่ใช้ในการเข้ารหัสด้วยอัลกอริทึม MD4.....	30
ตารางที่ 4.4 เวลาที่ใช้ในการเข้ารหัสด้วยอัลกอริทึม MD5.....	31
ตารางที่ 4.5 เวลาที่ใช้ในการเข้ารหัสด้วยอัลกอริทึม SHA-1.....	31
ตารางที่ 4.6 ผลการเข้ารหัสด้วยอัลกอริทึม CRC-16.....	32
ตารางที่ 4.7 ผลการเข้ารหัสด้วยอัลกอริทึม CRC-CCITT.....	32
ตารางที่ 4.8 ผลการเข้ารหัสด้วยอัลกอริทึม CRC-32.....	32
ตารางที่ 4.9 ผลการเข้ารหัสด้วยอัลกอริทึม Digit Analysis ที่มีขนาดรหัส 2 ไบต์.....	33
ตารางที่ 4.10 ผลการเข้ารหัสด้วยอัลกอริทึม Digit Analysis ที่มีขนาดรหัส 4 ไบต์.....	33
ตารางที่ 4.11 ผลการเข้ารหัสด้วยอัลกอริทึม Digit Analysis ที่มีขนาดรหัส 8 ไบต์.....	34
ตารางที่ 4.12 ผลการเข้ารหัสด้วยอัลกอริทึม Digit Analysis ที่มีขนาดรหัส 16 ไบต์.....	34
ตารางที่ 4.13 ผลการเข้ารหัสด้วยอัลกอริทึม Folding Method ที่มีขนาดรหัส 2 ไบต์.....	35
ตารางที่ 4.14 ผลการเข้ารหัสด้วยอัลกอริทึม Folding Method ที่มีขนาดรหัส 4 ไบต์.....	35
ตารางที่ 4.15 ผลการเข้ารหัสด้วยอัลกอริทึม Folding Method ที่มีขนาดรหัส 8 ไบต์.....	35
ตารางที่ 4.16 ผลการเข้ารหัสด้วยอัลกอริทึม Folding Method ที่มีขนาดรหัส 16 ไบต์.....	36
ตารางที่ 4.17 ผลการเข้ารหัสด้วยอัลกอริทึม Midsquare Method.....	36
ตารางที่ 4.18 ผลการวิเคราะห์ความยาวรหัสที่ได้จาก Midsquare Method.....	37
ตารางที่ 4.19 ผลการเข้ารหัสด้วยอัลกอริทึม Division Method ขนาดตัวหาร 2 ไบต์.....	37
ตารางที่ 4.20 ผลการวิเคราะห์ความยาวรหัสที่ได้จาก Division Method ขนาดตัวหาร 2 ไบต์..	38
ตารางที่ 4.21 ผลการเข้ารหัสด้วยอัลกอริทึม Division Method ขนาดตัวหาร 4 ไบต์.....	38
ตารางที่ 4.22 ผลการวิเคราะห์ความยาวรหัสที่ได้จาก Division Method ขนาดตัวหาร 4 ไบต์..	38
ตารางที่ 4.23 ผลการเข้ารหัสด้วยอัลกอริทึม Huffman Coding.....	39
ตารางที่ 4.24 ผลการวิเคราะห์ความยาวรหัสที่ได้จาก Huffman Coding.....	39
ตารางที่ 5.1 ตารางแสดงการเปรียบเทียบเวลาที่ใช้ในการเข้ารหัสแต่ละอัลกอริทึม.....	41
ตารางที่ 5.2 ตารางแสดงการเปรียบเทียบความยาวรหัสแต่ละอัลกอริทึม.....	45
ตารางที่ 5.3 ตารางแสดงการเปรียบเทียบประสิทธิภาพในการลดขนาดยูอาร์แอล.....	46
ตารางที่ 5.4 ตารางแสดงการเปรียบเทียบปริมาณการชนกันของรหัสแต่ละอัลกอริทึม.....	48