

รายการอ้างอิง

- [1] Ari Luotonen ,CERN, Kevin Altis, Intel. World – Wide Web Proxies. (n.p.):April 1994.
- [2] Tim Berners-Lee, CERN. Universal Resource indentifiers in WWW. (n.p.) 1994.
- [3] Wessels D., Claffy K. ICP and the Squid Web Cache, National Laboratory for Applied Network Research. Available from: <<http://www.nlanr.net/~wessels/Papers/icp-squid.ps.gz>>
- [4] Wessels D. The Squid Internet Object Cache, National Laboratory for Applied Network Research Available from: <<http://squid.nlanr.net/Squid/>>
- [5] Duane Wessel., Squid Programmer Guide. Available from: <<http://squid.nlanr.net/Doc/Prog-Guide/>>
- [6] R. Rivest. The MD5 Message-Digest Algorithm, RFC 1321. MIT Laboratory for Computer Science and RSA Data Security. Inc. 1992.
- [7] Yuping Yang, Neelam Soundarajan, Mukesh Singhal. Department of Computer and Information Science. The Ohio State University. Suitability of Signature Indexing Over the -World Wide Web. 1999.
- [8] William Stallings. Data and Computer Communications. Fifth Edition U.S.A: Prentice-Hall, 1997.
- [9] Richard W. Hamming. Coding and Information Theory. Second Edition. U.S.A: Prentice-Hall, 1986.
- [10] Jean – Paul Tremblay, Paul G. Sorenson. An Introduction to Data Structures with Application Second Edition. U.S.A: McGraw-Hill International Editions, 1984.
- [11] Donald E. Knuth. The Art of Computer Programming, Vol. 3/Sorting and Searching. Addison-Wesley Publishing Company, 1973
- [12] B. Kaliski. The MD2 Message-Digest Algorithm, RFC 1319. RSA Laboratories. 1992.
- [13] The MD4 Message-Digest Algorithm, RFC 1320. R. Rivest MIT Laboratory for Computer Science and RSA Data Security. 1992.

- [14] SECURE HASH STANDARD". FIPS PUB 180-1, The Federal Information Processing Standards Publication Series of the National Institute of Standards and Technology (NIST)1993.
- [15] W. W. Chang, H. J. Schek. A Signature Access Method for the Starburst Database System Proc. of the 15th Int'l Conf. on Very Large Data Bases, 1989.
- [16] C. Faloutsos, Signature-based text retrieval methods: A survey, Data Engineering Bulletin, Vol. 13, No. 1, March 1990.
- [17] Y. Ishikawa, H. Kitagawa, N. Ohbo, Evaluation of Signature Files as Set Access Facilities in OODBs, Proc. of the 1993 SIGMOD Conference, Washington, DC, June 1993.
- [18] L. Fan, P. Cao, J. Almeida, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol", Technical Report CS-TR-98-1361, University of Wisconsin – Madison, 1998. In Proceedings of ACM SIGCOMM'98, September, 1998.
- [19] Y. Yang, M. Singhal, Signature Cache: A Light Weight Web Cache Indexing Structure, Special Issue on World Wide Web, Journal of the Brazilian Computer Society, Vol. 5, No.2, November 1998.
- [20] Sharp Corp., Super Proxy Script Available from: <http://naragw.sharp.co.jp/sps/>

ภาคผนวก

ภาคผนวก ก

รูปแบบข้อมูลการเรียกขอแบบสควิดที่พร้อมใช้แคชบันทึกไว้

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

time	เวลาที่บันทึกการเรียกขอ
elapsed	เวลาที่ใช้ในการรับข้อมูลจากเว็บเซิร์ฟเวอร์ หน่วยเป็นมิลลิวินาที
remotehost	หมายเลขไอพีของเว็บเซิร์ฟเวอร์
code/status	รหัสการตอบกลับจากเว็บเซิร์ฟเวอร์
bytes	ปริมาณข้อมูลที่ได้รับมาจากเว็บเซิร์ฟเวอร์ หน่วยเป็นไบต์
method	วิธีการเรียกขอข้อมูล
URL	ที่อยู่ของข้อมูลที่เรียกขอ
rfc931	ชื่อของผู้เรียกขอ โดยปกติจะไม่เก็บข้อมูลนี้ไว้
peerstatus/peerhost	วิธีส่งต่อการเรียกขอ/ไอพีแอดเดรสของปลายทางที่การเรียกขอถูกส่งไป

ภาคผนวก ข

โปรแกรมที่พัฒนาขึ้นมาเพื่อใช้ประกอบงานวิจัย

โปรแกรมที่ใช้คัดเลือกเฉพาะยูอาร์แอลออกจากเพิ่มการเรียกขอที่อยู่ในรูปแบบสควิด

urls.pl

```
#!/usr/bin/perl
#
# This script is use to extract URLs from Squid Log file.
#
while (<>) {
    my($url)=(split)[6];
    print "$url\n";
}
```

รูปแบบการเรียกใช้งานโปรแกรม

\$ urls.pl < <ชื่อเพิ่มข้อมูลการเรียกขอ> > <ชื่อเพิ่มข้อมูลที่ต้องการบันทึกผลลัพธ์>

โปรแกรมที่ใช้คัดเลือกยูอาร์แอลที่สามารถแคชได้เท่านั้น

cacheable_urls.pl

```
#!/usr/bin/perl

# For the comment of sign that presented in URLs and notifying
# the non-cacheable URLs, '?' is the problem. Cut it out!
#

my $url;
my $nohost;

while(<>) {
    $url = $_;
    $nohost = $_;
    if (length($nohost) >=7) {
        $nohost =~ s/^[^/]*\\/[^\/*]*\\//;
        unless ($nohost =~ /cgi-bin|\.cgi|\.jsp|\.exe|\.pl|\.asp|\.php|\.dll|
            \.cfm|\.class|\\?/) {
            print "$url";
        }
    }
}
```

รูปแบบการเรียกใช้งานโปรแกรม

\$ cacheable_urls.pl < [ชื่อเพิ่มข้อมูลยูอาร์แอล] > [ชื่อเพิ่มข้อมูลที่ต้องการบันทึกผลลัพธ์]

โปรแกรมที่ใช้วิเคราะห์ความยาวยูอาร์แอล

URLlen.pl

```
#!/usr/bin/perl
# The diffence between this file with URLlen3.pl is
# this script is not open any file. It use redirection in UNIX for
# input data
#
```

```

# Known Bugs:
#   Finding the length that has maximum repeat times, if the lengths
#   that has maximum repeat times are more than 1 length. The report
#   will list only the last one.
my %url;
my $len;
my $lensqr;
my $u;

my $lines=0;
my $maxlen=0;
my @urlllen;
my $total=0;
my $totalchar=0;

while(<>) {
  chomp($_);
  $len = length($_);
  $lensqr += $len * $len;
  if ($len >= $maxlen) { $maxlen = $len; }
  if (exists $url{$len}) {
    $url{$len}++;
  }
  else {
    $url{$len} = 1;
  }
  $lines++;
  $totalchar += $len;
}

@urlllen = sort { $a <=> $b } keys (%url);
$max_repeat = 0;
print "Chars\tTimes\n";
print "=====\t=====\n";
foreach $url_length (@urlllen) {
  print $url_length."\t".$url{$url_length}."\n";
  if ( $url{$url_length} >= $max_repeat ) {
    $max_repeat = $url{$url_length};
    $maxtimes = $url_length;
  }
  $total = $total+$url{$url_length};
}
print "Lines processed: $lines\n";
printf("URL average length: %f\n",$totalchar/$lines);
print "Max Length = ".$maxlen."\n";
print "Max times length = ".$maxtimes."\n";
print "Max repeat times = ".$max_repeat."\n";
$u = $totalchar/$lines;
print "Standard Diviation =" . sqrt(($lensqr/$lines) - ($u * $u));

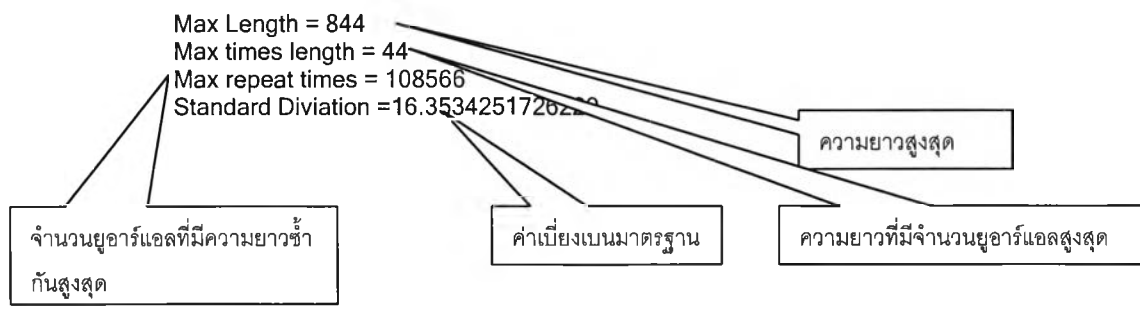
```

รูปแบบการเรียกใช้งานโปรแกรม

\$ URLlen.pl < [ชื่อเพิ่มข้อมูลยูอาร์แอล] > [ชื่อเพิ่มข้อมูลที่ต้องการบันทึกผลลัพธ์]

ผลในแฟ้มบันทึกผลลัพธ์

Chars	Times	
6	9	ความยาวยูอาร์แอล หน่วย ตัวอักษร
7	116	จำนวนยูอาร์แอลที่มีความยาวดังตัวเลขข้างหน้า
9	1	
...		
616	1	จำนวนยูอาร์แอลที่ถูกประมวลผล
648	1	
844	1	
Lines processed: 2271128		ความยาวเฉลี่ยยูอาร์แอล
URL average length: 49.681380		



ภาคผนวก ค

โปรแกรมที่ใช้ในการเข้ารหัส

โปรแกรมในกลุ่ม MD (Message Digest) ใช้โปรแกรมไดรเวอร์เดียวกัน แตกต่างกันเฉพาะในส่วนของการเข้ารหัส มีรายชื่อไฟล์ดังต่อไปนี้

โปรแกรมเข้ารหัสในกลุ่ม MD

global.h	เฮดเดอร์ไฟล์ กำหนดค่าต่างๆ ใช้กับทุกๆ MD
md2.h	เฮดเดอร์ไฟล์ กำหนดค่าต่างๆ ใช้งานกับโปรแกรมเข้ารหัส MD2 เท่านั้น
md4.h	เฮดเดอร์ไฟล์ กำหนดค่าต่างๆ ใช้งานกับโปรแกรมเข้ารหัส MD4 เท่านั้น
md5.h	เฮดเดอร์ไฟล์ กำหนดค่าต่างๆ ใช้งานกับโปรแกรมเข้ารหัส MD5 เท่านั้น
md2c.c	โปรแกรมเข้ารหัส MD2
md4c.c	โปรแกรมเข้ารหัส MD4
md5c.c	โปรแกรมเข้ารหัส MD5
md401.c	โปรแกรมไดรเวอร์สำหรับเข้ารหัสในกลุ่ม md

โปรแกรมเข้ารหัส SHA-1 ใช้ไฟล์เพียง 2 ไฟล์โดยโปรแกรมไดรเวอร์รวมอยู่ในตัวแล้ว และมีลักษณะเหมือนกับโปรแกรมไดรเวอร์ของโปรแกรมเข้ารหัสในกลุ่ม MD

โปรแกรมเข้ารหัส SHA-1

sha.h	เฮดเดอร์ไฟล์ กำหนดค่าต่างๆ ใช้กับโปรแกรมเข้ารหัส SHA-1
sha.c	โปรแกรมเข้ารหัส SHA-1

โปรแกรมเข้ารหัสในกลุ่ม CRC นั้น CRC-16 และ CRC-CCITT เป็นโปรแกรมเดียวกัน ต่างกันเพียงการกำหนดตัวแปรบางตัว และส่วนของการเรียกใช้โปรแกรมเข้ารหัสเท่านั้น นอกจากนั้น ส่วนของโปรแกรมนอกจากส่วนที่เรียกโปรแกรมย่อยเพื่อเข้ารหัส จะมีลักษณะเดียวกับโปรแกรมไดรเวอร์ของโปรแกรมในกลุ่ม MD และ SHA

โปรแกรมเข้ารหัส CRC-16 CRC-CCITT

crcfastdrv.c	โปรแกรมเข้ารหัส CRC-16
crcttfast.c	โปรแกรมเข้ารหัส CRC-CCITT

ส่วนโปรแกรม CRC-32 แตกต่างจาก CRC-16 และ CRC-CCITT ซึ่งตามทฤษฎีแตกต่างกันเพียงตัวหาร เนื่องจาก CRC-32 ทำงานแบบ 32 บิต จึงมีการเขียนโปรแกรมให้สามารถทำงานได้ดีกว่ากับสถาปัตยกรรมคอมพิวเตอร์ปัจจุบัน

โปรแกรมเข้ารหัส CRC

crc32.h	เฮดเดอร์ไฟล์ กำหนดค่าต่างๆ ใช้กับโปรแกรมเข้ารหัส CRC-32
crc32drv.c	โปรแกรมไดรเวอร์สำหรับเข้ารหัส CRC-32
crc32.c	โปรแกรมสำหรับเข้ารหัส CRC-32

โปรแกรมในกลุ่ม Simple Hash Function ทุกโปรแกรมเป็นโปรแกรมที่มีไดรเวอร์ในตัว ไม่มีการแยกโปรแกรมออกเป็นส่วนของไดรเวอร์และส่วนเข้ารหัสดังเช่นโปรแกรมในกลุ่ม MD และมีบางโปรแกรมที่มีการกำหนดขนาดของรหัส ซึ่งโปรแกรมหากล่าวจะมีความแตกต่างกันเพียงในส่วนการกำหนดความยาวรหัสนั้น

digitXX.c	โปรแกรมเข้ารหัส Digit Analysis Method โดย XX เป็นขนาดของรหัสที่กำหนด คือ 02, 04, 08 และ 16 ไบต์
foldingXX.c	โปรแกรมเข้ารหัส Folding Method โดย XX เป็นขนาดของรหัสที่กำหนด คือ 02, 04, 08 และ 16 ไบต์
division03.c	โปรแกรมเข้ารหัส Division Method
midsquare.c	โปรแกรมเข้ารหัส Midsquare Method

โปรแกรมสุดท้ายจะเป็นโปรแกรมที่ใช้เข้ารหัสแบบฮัฟแมน (Huffman Coding) โปรแกรมแบ่งออกเป็น 2 ส่วนตามลักษณะเฉพาะของอัลกอริทึม กล่าวคือ โปรแกรมแรกจะเป็นโปรแกรมที่ใช้ในการวิเคราะห์ความถี่ของการปรากฏตัวอักษร และอีกโปรแกรมจะเป็นโปรแกรมซึ่งทำการเข้ารหัสจริงโดยอาศัยข้อมูลความถี่ตัวอักษรจากโปรแกรมแรก

huff_analyse_101.c	โปรแกรมวิเคราะห์ความถี่ของการปรากฏตัวอักษร
myhuff53.c	โปรแกรมเข้ารหัสแบบฮัฟแมน (Huffman Coding)

โปรแกรมทั้งหมดสามารถดูรายละเอียดได้ในซีดีรอมประกอบวิทยานิพนธ์

การคอมไพล์โปรแกรม

โปรแกรมเข้ารหัสบางโปรแกรมมีการคอมไพล์ที่ต้องกำหนดมาโคร มีรายละเอียดดังนี้

การคอมไพล์โปรแกรมในกลุ่ม MD จะต้องกำหนดค่ามาโคร ดังนี้

```
$ cc -o <ชื่อโปรแกรมที่ต้องการ> md401.c mdXc.c -DMD=X
```

โดย X มีค่าเป็น 2,4 หรือ 5

การคอมไพล์โปรแกรม CRC-32

```
$ cc -o <ชื่อโปรแกรมที่ต้องการ> crc32drv.crc32.c
```

การคอมไพล์โปรแกรม Division Method

```
$ cc -o <ชื่อโปรแกรมที่ต้องการ> division03.c -DDIV_LEN=X
```

โดย X มีค่าเป็น 2 หรือ 4 เพื่อกำหนดขนาดความยาวของตัวหาร

ประวัติผู้วิจัย

นายประเสริฐ วิชาอุภาส เกิดเมื่อวันจันทร์ที่ 2 กุมภาพันธ์ พ.ศ. 2513 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ ในปีการศึกษา 2536 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2541 ปัจจุบันทำงานที่ แผนกติดตามและประเมินผลโครงการ กองประสานงานโครงการ ฝ่ายปฏิบัติการด้านการใช้ไฟฟ้า การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย ในตำแหน่ง วิศวกรระดับ 6

