วิธีการค้นคืนข้อมูลอย่างมีประสิทธิภาพสำหรับระบบจัดการการเรียนการสอน

นายธนินทร์ ระเบียบโพธิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2550

AN OPTIMIZED DATA RETRIEVAL APPROACH FOR LEARNING MANAGEMENT SYSTEM

Thanin Rabiabpo

A Thesis Submitted in Partial Fulfillment of the Requirments for the Degree of Master of Engineering in Computer Engineering

Suranaree University of Technology

Academic Year 2007

วิธีการค้นคืนข้อมูลอย่างมีประสิทธิภาพสำหรับระบบจัดการการเรียนการสอน

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
ประธานกรรมการ

_____

(ผศ. ดร.คะชา ชาญศิลป์)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

_____

(อ.ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

_____

(ศ. ดร.ไพโรจน์ สัตยธรรม)
รองอธิการบดีฝ่ายวิชาการ

_____

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

ธนินทร์ ระเบียบโพธิ์ : วิธีการค้นคืนข้อมูลอย่างมีประสิทธิภาพสำหรับระบบจัดการการเรียนการสอน : (AN OPTIMIZED DATA RETRIEVAL APPROACH FOR LEARNING MANAGEMENT SYSTEM) อาจารย์ที่ปรึกษา: ผศ. ดร.คะชา ชาญศิลป์, 130 หน้า.

ในปัจจุบันงานด้านการพัฒนาสารสนเทศเพื่อการเรียนการสอน และการพัฒนาบทเรียนคอมพิวเตอร์ผ่านเครือข่าย (Web-Base Instruction : WBI) มีบทบาทสำคัญมาก เนื่องจากความต้องการของโรงเรียน หรือสถาบันการศึกษา ต้องการประหยัดเวลาในการสร้าง และการพัฒนาบทเรียน (Content Management System) รวมถึงการบริหารจัดการบทเรียนผ่านเครือข่าย จึงได้มีการพัฒนาซอฟต์แวร์ระบบจัดการบทเรียนแบบโอเพนซอร์ส (Open Learning Source Management System) เช่น Moodle, Atutor, Claroline, Learnloop, Splearn และ Vclass เป็นต้น ด้วยการใช้งานที่ง่าย สามารถจัดการเว็บไซต์บทเรียนได้แบบเบ็ดเสร็จ แก้ไข ปรับปรุง ดัดแปลง โปรแกรมได้ง่าย และสามารถทำเนื้อหาให้ทันสมัยอยู่เสมอโดยไม่ต้องอาศัยผู้เชี่ยวชาญด้านเทคนิค ช่วยแก้ไขปัญหาการขาดแคลนบุคลากรด้านการออกแบบเว็บ และโปรแกรมเมอร์ หรือผู้เชี่ยวชาญด้านการพัฒนาเว็บไซต์ ดังนั้น SUTinsServer 5103 Plus Moodle จึงเป็นระบบปฏิบัติการที่ได้นำมาทดสอบและวิจัย ซึ่งมีระบบจัดการบทเรียน Moodle ติดตั้งมาพร้อมกับระบบลินุกซ์เซิร์ฟเวอร์พร้อมใช้สำหรับนักพัฒนาจากมหาวิทยาลัยเทคโนโลยีสุรนารี

อย่างไรก็ตามการทำงานของระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตแบบเดิมนั้น ทุกครั้งที่มีการร้องขอข้อมูลจากเซิร์ฟเวอร์ ผู้ใช้จะต้องรอจนกว่าเซิร์ฟเวอร์จะประมวลผลข้อมูลทั้งหมดเสร็จแล้วข้อมูลจึงจะถูกส่งกลับมา ทำให้เกิดความล่าช้า การวิจัยนี้ ได้ออกแบบและพัฒนาเพื่อเพิ่มประสิทธิภาพของการค้นคืนข้อมูล โดยนำเทคโนโลยีการ Caching ข้อมูลของโปรแกรมภาษา PHP มาเป็นส่วนเสริม พร้อมทั้งนำเสนอผลการทดลองดังกล่าว

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2550 ลายมือชื่ออาจารย์ที่ปรึกษา_____

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม_____

THANIN RABIABPO : AN OPTIMIZED DATA RETRIEVAL APPROACH FOR LEARNING MANAGEMENT SYSTEM.

THESIS ADVISOR : ASST. PROF. KACHA CHANSILP, Ph.D., 130 PP.

CACHING OUTPUT/ RETRIEVAL/OPENSOURCE/LEARNING MANAGEMENT SYSTEM/LMS

At present, Web-Based Instruction plays an important role for learning performance. An educational institution need to save time to create a Content Management System via an internet network. The result is Open Learning Source Management System. The popular open source platforms are Moodle, Atutor, Claroline, Learnloop, Splearn and Vclass because they are easy to use, edit, improve and adapt the content without any assistance from a web master or programmer. Therefore, at SUT, there is the SUTinsServer 5103 Plus Moodle with has a content management system already installed on a Linux platform for software development.

However, its process is slow. When users request some recent data on web page, they have to wait until the server processes all data and refreshing all over the web page again. This research describes the design and development strategies by incorporating Caching PHP data programe to increase the performance. The results are also presented.

School of Computer Engineering

Academic Year 2007

Student's Signature_____

Advisor's Signature_____

CO-advisor's Signature_____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคลดังต่อไปนี้ ที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลือ อย่างดียิ่ง ทั้งในด้านวิชาการ และ ด้านการดำเนินงานวิจัย

- ผู้ช่วยศาสตราจารย์ ดร.คะชา ชาญศิลป์ อาจารย์ที่ปรึกษาวิทยานิพนธ์
- ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
- ผู้ช่วยศาสตราจารย์ สมพันธุ์ ชาญศิลป์ อาจารย์ผู้ให้คำแนะนำปรึกษา

ขอขอบคุณ อาจารย์ธรรมศักดิ์ เธียรนิเวศน์ ที่ได้ให้คำแนะนำและให้คำปรึกษาในการเขียนวิทยานิพนธ์ฉบับนี้ และขอขอบคุณเพื่อน ๆ บัณฑิตศึกษาสาขาวิชาวิศวกรรมคอมพิวเตอร์ทุกคนที่เป็นกำลังใจมาโดยตลอด

ท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา ที่ให้การเลี้ยงดูอบรมและส่งเสริมการศึกษาเป็นอย่างดีมาโดยตลอด จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตตลอดมา

ธนินทร์ ระเบียบโพธิ์

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ
บทที่
1 บทนำ 1
1.1 ความสำคัญและที่มาของปัญหาการวิจัย 1
1.2 วัตถุประสงค์การวิจัย 3
1.3 ขอบเขตการวิจัย 3
1.4 ประโยชน์ที่คาดว่าจะได้รับ 4
2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 5
2.1 เทคโนโลยีการค้นคืนข้อมูล 5
2.1.1 AJAX (Asynchronous JavaScript And XML) 6
2.1.2 ระบบไดนามิกเว็บแคช (Dynamic Web Caching System) 13
2.2 เทคโนโลยีเกี่ยวกับเว็บไซต์ 17
2.2.1 ประเภทของ Dynamic Web Page 18
2.2.2 เทคโนโลยีของเว็บไซต์ (Web Site Technology) 18
2.3 ระบบฐานข้อมูล MySQL 23
2.4 โปรแกรมภาษา PHP 24
2.5 มูเดิ้ล (Moodle) 26
2.6 งานวิจัยที่เกี่ยวข้อง 27
3 วิธีดำเนินการวิจัย 31
3.1 ระเบียบวิธีวิจัย 31

สารบัญ (ต่อ)

หน้า

3.2 ข้อมูลที่ใช้ในการวิจัย....32
3.3 เทคนิคการค้นคืนข้อมูลแบบวิธีการค้นคืนข้อมูลจากระบบ Caching ของเอาท์พุต....41
3.3.1 โครงสร้างของระบบ....42
3.3.2 กรณีทดสอบที่ใช้ทดสอบระบบ....46
3.3.3 ขั้นตอนการทดสอบ....47
3.4 การเปรียบเทียบประสิทธิภาพของอัลกอริธึมการค้นคืนข้อมูล....48
4 ผลการวิเคราะห์ข้อมูลและอภิปรายผล....53
4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาระบบการค้นคืนข้อมูล....53
4.2 ผลการทดสอบระบบ....54
4.2.1 การทดสอบระบบค้นคืนข้อมูลแบบเดิม....54
4.2.2 การทดสอบระบบการค้นคืนข้อมูลแบบที่ได้นำอัลกอริธึม การ Caching ข้อมูลของเอาท์พุตมาพัฒนาร่วม....73
4.3 อภิปรายผล....98
5 บทสรุป....97
5.1 สรุปผลการวิจัย....99
5.1.1 สรุปผลการทดสอบระบบการค้นคืนข้อมูลแบบเดิม....99
5.1.2 สรุปผลการทดสอบระบบการค้นคืนข้อมูลแบบการ Caching ข้อมูลของเอาท์พุต....99
5.2 ประโยชน์ของระบบการค้นคืนข้อมูลจากการ Caching ข้อมูลของเอาท์พุต....100
5.3 ข้อจำกัดของระบบการค้นคืนข้อมูลจากการ Caching ข้อมูลของเอาท์พุต....100
5.4 แนวทางในการพัฒนาต่อ....100
รายการอ้างอิง....101
ภาคผนวก
ภาคผนวก ก บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการทางคอมพิวเตอร์และเทคโนโลยีสารสนเทศ....103

สารบัญ (ต่อ)

หน้า

ภาคผนวก ข รูปแบบของการค้นคืนข้อมูลแบบ Cache โดยภาษา PHP
และการค้นคว้าเครื่องมือที่จะใช้ในการศึกษาวิจัย 111
ภาคผนวก ค วิธีการใช้งานโปรแกรม Webserver Stress Tool 125
ประวัติผู้เขียน 130

สารบัญตาราง

ตารางที่ **หน้า**

3.1 แสดงตารางฐานข้อมูลที่ใช้ในระบบ Moodle 33
3.2 การเลือกองค์ประกอบ Output Control 45
4.1 แสดงค่า Protocol Times for all URLs แบบเดิม 55
4.2 แสดงค่า Average User และ Server Bandwidth แบบเดิม 57
4.3 แสดงค่า Transferred Data, System Memory และ CPU Load แบบเดิม 60
4.4 แสดงค่า Open Requests และ Transferred Data แบบเดิม 63
4.5 แสดงค่า Click Time, User/s และ Hits/s for all URLs แบบเดิม 66
4.6 แสดงค่า Average Request Time และ Errors (per URL) แบบเดิม 69
4.7 แสดงค่า Protocol Times for all URLs แบบที่ใช้ระบบ CO-LMS 73
4.8 แสดงค่า Average User และ Server Bandwidth แบบที่ใช้ระบบ CO-LMS 76
4.9 แสดงค่า Transferred Data, System Memory และ CPU Load แบบที่ใช้ระบบ CO-LMS 79
4.10 แสดงค่า Open Requests และ Transferred Data แบบที่ใช้ระบบ CO-LMS 82
4.11 แสดงค่า Click Time, User/s และ Hits/s for all URLs แบบที่ใช้ระบบ CO-LMS 85
4.12 แสดงค่า Average Request Time และ Errors (per URL) แบบที่ใช้ระบบ CO-LMS 88
4.13 แสดงค่าผลที่ได้จากระบบแบบเดิม 92
4.14 แสดงค่าผลที่ได้จากระบบแบบที่ได้นำระบบ CO-LMS 93
4.15 แสดงค่าเฉลี่ยของ Clicks Hits Errors Average Click Time Bytes และ kbit/s ในการทดสอบระบบแบบเดิมกับระบบ CO-LMS 93
4.16 แสดงค่า Time Spent และ Avg. Click Time จากระบบแบบเดิม 94
4.17 แสดงค่า Time Spent และ Avg. Click Time จากระบบแบบที่ได้นำระบบ CO-LMS 95
4.18 แสดงการเปรียบเทียบค่าของเวลาที่ผู้ใช้ได้รับการตอบสนองจากระบบ 95
ข.1 แสดงการเปรียบเทียบการใช้โปรแกรม 123

สารบัญรูป

รูปที่ **หน้า**

2.1 แสดงการเปรียบเทียบการทำงานแบบเดิม กับ AJAX 7
2.2 การเปรียบเทียบระหว่างการสื่อสารแบบ Synchronous กับแบบ Asynchronous 9
2.3 AJAX Architecture 10
2.4 การทำงานแบบ Asynchronous และการเปลี่ยนแปลงหน้าเว็บแบบบางส่วน 11
2.5 Proxy-Cache เซิร์ฟเวอร์ Network Configuration 14
2.6 Proxy-Cache Network Configuration with Firewall 15
2.7 Multi-level Web Caching Organization 16
2.8 แสดงการเปรียบเทียบระหว่าง Static กับ Dynamic Web 19
2.9 แสดง Web services actors, objects และ operations 20
3.1 แสดงความสัมพันธ์ของฐานข้อมูลของโมดูล forum 38
3.2 แสดงความสัมพันธ์ของฐานข้อมูล block 39
3.3 แสดงความสัมพันธ์ของฐานข้อมูล excercise 39
3.4 แสดงความสัมพันธ์ของฐานข้อมูล cache_text 40
3.5 แสดงความสัมพันธ์ของฐานข้อมูล user 40
3.6 แสดงความสัมพันธ์ของฐานข้อมูล course 41
3.7 แสดงโครงสร้างของระบบโดยทั่วไป 42
3.8 แสดงส่วนการทำงานของระบบ CO-LMS 43
3.9 แสดง Flowchart ของระบบ CO-LMS 44
3.10 แสดงขั้นตอนการทำงานของระบบ CO-LMS 48
3.11 การทดสอบ Apache บน Windows 2003 และ SUTinsServer 5103 Plus Moodle 50
3.12 กราฟแสดงเวลาเฉลี่ยในการตอบสนองต่อผู้เข้าชมเว็บไซต์ 51
4.1 กราฟแสดงค่าของ Protocol Times for all URLs 57
4.2 กราฟแสดงค่าของ Average User and Server Bandwidth 60
4.3 กราฟแสดงค่าของ Transferred Data และ System Memory และ CPU Load 63
4.4 กราฟแสดงค่าของ Open Requests และ Transferred Data 66

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.5 กราฟแสดงค่าของ Click Time, Hits/s และ User/s for all URLs 69
4.6 กราฟแสดงค่าของ Average Request Time และ Errors (per URL) 72
4.7 กราฟแสดงค่า Spectrum of Click Times 72
4.8 กราฟแสดงค่าของ Protocol Times for all URLs 76
4.9 กราฟแสดงค่าของ Average User และ Server Bandwidth 79
4.10 กราฟแสดงค่าของ Transferred Data, System Memory และ CPU Load 82
4.11 กราฟแสดงค่าของ Open Requests และ Transferred Data 85
4.12 กราฟแสดงค่าของ Click Time, Hits/s และ User/s for all URLs 88
4.13 กราฟแสดงค่าของ Average Request Time และ Errors (per URL) 91
4.14 กราฟแสดงค่า Spectrum of Click Times 91
4.15 แสดงการเปรียบเทียบค่าเฉลี่ยของเวลาในการตอบสนองต่อผู้ใช้ 98
ข.1 แสดงหน้าจอโปรแกรม 116
ข.2 แสดง log file จากการทำงานของโปรแกรม 117
ข.3 แสดงหน้าจอโปรแกรม 117
ข.4 แสดงแสดงผลลัพธ์ เป็น Log file 118
ข.5 แสดง Server and User Bandwidth 119
ข.6 แสดง Test Client Health 119
ข.7 แสดง Open Requests and Transferred Data 120
ข.8 แสดงโปรแกรมระหว่างการทำงาน 120
ข.9 แสดงหน้าจอโปรแกรม 121
ข.10 แสดงหน้าจอโปรแกรม 122
ข.11 แสดงผลที่ได้จากการ Monitor ของโปรแกรม 123
ค.1 แสดงหน้าต่างโปรแกรม Webserver Stress Tool 126
ค.2 หน้าต่างโปรแกรมส่วนของ URL Platern 127
ค.3 หน้าต่างโปรแกรมส่วนของ Browser Simulation 128
ค.4 แสดงการกำหนดค่า Program Option และแสดงผลลัพธ์ เป็น Log และ html file 129

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ปัจจัยสำคัญอย่างหนึ่ง ที่จะสร้างความสามารถในการแข่งขันทางด้านเศรษฐกิจ คือ การประยุกต์ใช้ฐานความรู้ แต่หากสร้างความรู้ขึ้นมาแล้วไม่มีวิธีการเก็บรักษา หรือไม่มีวิธีการนำไปใช้ให้เกิดประโยชน์สูงสุดก็จะเป็นที่น่าเสียดายอย่างมาก ดังนั้นระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต (e-Learning) จึงได้ถูกนำมาพิจารณาในการเผยแพร่ความรู้ดังกล่าวสู่กลุ่มเป้าหมาย เนื่องจากระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตนี้ จะเอื้ออำนวยให้ผู้เรียนได้รับอิสระในการเลือกเวลาและสถานที่ตามต้องการได้ ทั้งยังประหยัด ง่าย และสะดวกต่อการปรับเปลี่ยนข้อมูลบทเรียน ดังนั้นบริษัทและองค์กรต่าง ๆ จึงพัฒนาระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตนี้ขึ้นมาเพื่อรองรับความต้องการดังกล่าวไว้แล้วข้างต้น

ปัจจุบัน ระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ได้รับการยอมรับว่าเป็นกลไกสำคัญที่สนับสนุนและส่งเสริมประสิทธิภาพการเรียนการสอนของนักศึกษา ทั้งในโลกปัจจุบันและโลกอนาคต ช่วยให้ผู้เรียนเข้าถึงแหล่งความรู้ได้ด้วยตนเองตามอัธยาศัย และสนับสนุนระบบการเรียนรู้แบบมีผู้เรียนเป็นศูนย์กลาง (Student-Centered Learning) ผู้เรียนสามารถควบคุมการเรียนรู้ได้ด้วยตัวเอง (Self-Directed Learning) และสนับสนุนการเรียนรู้แบบทุกคน ทุกเวลา ทุกสถานที่ (anyone-anywhere-anytime learning)

ระบบการจัดการการเรียนรู้ (Learning Management System : LMS) เป็นระบบที่สามารถจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ซึ่งประกอบด้วยโมดูลต่าง ๆ ที่ทำหน้าที่ในการจัดการฐานความรู้ เช่น การสร้างรายวิชา และแทรกเนื้อหาสาระ โดยในปัจจุบันมีระบบการจัดการการเรียนรู้ดังกล่าวขึ้นอย่างมากมาย ซึ่งแต่ละระบบต่างมีข้อดีและข้อเสียแตกต่างกันออกไป โดยระบบการจัดการการเรียนรู้ที่เป็นที่รู้จักและนิยมใช้งานในปัจจุบัน ได้แก่ Moodle, Atutor, Claroline, Learnloop, Splearn และ Vclass โดยเฉพาะระบบ Moodle (Modular Object Oriented Dynamic Learning Environment) ซึ่งถือได้ว่าเป็นระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่ได้รับความนิยมอย่างแพร่หลาย โดย Moodle ถูกพัฒนาขึ้นด้วยภาษา PHP และทำงานร่วมกับระบบฐานข้อมูล MySQL อีกทั้งยังเป็นโอเพนซอร์ส ซึ่งผู้ใช้สามารถนำมาใช้ได้ฟรีโดยไม่มีค่าใช้จ่าย และสามารถนำไปพัฒนาเพิ่มเติมได้ โดยที่ผู้พัฒนาไม่ต้องเสียค่าใช้จ่ายทางด้านลิขสิทธิ์สำหรับระบบ

ต้นฉบับ ในประเทศไทยมีกลุ่มผู้ใช้งาน Moodle อยู่หลายสถาบัน เช่น มหาวิทยาลัยบูรพา มหาวิทยาลัยเทคโนโลยีสุรนารี มหาวิทยาลัยวลัยลักษณ์ จุฬาลงกรณ์มหาวิทยาลัย มหาวิทยาลัยสงขลา มหาวิทยาลัยขอนแก่น มหาวิทยาลัยราชภัฏกว่า 20 สถาบัน กลุ่มโรงเรียนประถมและมัธยมศึกษาในสังกัดสำนักงานคณะกรรมการการศึกษาขั้นพื้นฐานกว่า 1,000 โรงเรียน

เนื่องจากการใช้งาน Moodle ในปัจจุบันได้รับความนิยมกันอย่างแพร่หลาย ทั้งในประเทศและต่างประเทศอีกกว่า 150 ประเทศทั่วโลก จึงได้มีการพัฒนาโมดูลการใช้งานต่าง ๆ ของ Moodle ก็เพิ่มขึ้นเรื่อย ๆ แต่อย่างไรก็ตาม ระบบที่มีการใช้งานเพิ่มมากขึ้น มักจะเกิดปัญหาต่าง ๆ ตามมา เช่น ความล่าช้าในการค้นคืนข้อมูลเพื่อแสดงผลข้อมูลออกมาทางเบราว์เซอร์ ซึ่งเกิดจากการเติบโตของข้อมูลที่ถูกเก็บไว้ในระบบฐานข้อมูลมีจำนวนมาก

เทคโนโลยีที่นำมาใช้เพื่อเพิ่มประสิทธิภาพให้กับการค้นคืนข้อมูลในปัจจุบัน มีมากมายหลายรูปแบบ ยกตัวอย่าง เช่น เทคโนโลยี AJAX (Asynchronous JavaScript And XML) และการทำระบบ Caching โดยเทคโนโลยี AJAX เป็นเทคนิคหนึ่งในการพัฒนาเว็บแอพพลิเคชันที่ช่วยเพิ่มประสิทธิภาพในการค้นคืนข้อมูลสำหรับเว็บเพจที่มีการเปลี่ยนแปลงเฉพาะบางส่วนบ่อย ๆ โดยไม่ต้องโหลดข้อมูลใหม่ทั้งหมด ในเว็บแอพพลิเคชันทั่วไป ๆ นั้น เมื่อผู้ใช้ร้องขอข้อมูลไปยังเซิร์ฟเวอร์และได้รับข้อมูลใหม่จากเซิร์ฟเวอร์ เว็บเพจจะต้องโหลดข้อมูลใหม่ทุกครั้งเพื่อแสดงผลการทำงาน แต่เมื่อนำเทคโนโลยี AJAX มาใช้จะช่วยลดปริมาณการติดต่อระหว่างเซิร์ฟเวอร์และไคลเอนท์ลงได้ เพราะว่าเทคโนโลยี AJAX นั้นจะทำการโหลดข้อมูลใหม่เฉพาะข้อมูลที่มีการเปลี่ยนแปลงบางส่วนของเว็บเพจเท่านั้น แต่ถ้าหากเว็บแอพพลิเคชันนั้น ๆ ต้องมีการเปลี่ยนแปลงหน้าเว็บเพจทั้งหมดทุกครั้งที่มีการร้องขอข้อมูลจากผู้ใช้ การนำเทคโนโลยี AJAX มาใช้ก็จะไม่เกิดประโยชน์แต่อย่างใด ส่วนอีกเทคโนโลยีหนึ่งที่ได้รับความสนใจคือ เทคนิคการทำระบบ Caching ร่วมกับการพัฒนาเว็บแอพพลิเคชัน ซึ่งระบบดังกล่าวจะสามารถตอบสนองต่อรูปแบบของเว็บแอพพลิเคชันได้ทุกรูปแบบ ไม่ว่าเว็บแอพพลิเคชันจะเปลี่ยนแปลงบางส่วนของหน้าเว็บ หรือเปลี่ยนแปลงไปทั้งหมดก็ตาม

ดังนั้นงานวิจัยนี้จึงมุ่งที่จะศึกษาค้นคว้า ตลอดจนพัฒนาเทคนิคการค้นคืนข้อมูลในระบบการจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต โดยใช้เทคโนโลยีระบบ Caching เพื่อเพิ่มประสิทธิภาพในการค้นคืนข้อมูลให้รวดเร็วมากขึ้น

1.2 วัตถุประสงค์การวิจัย

1.2.1 เพื่อศึกษาและวิจัยแนวทางการพัฒนาระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่สามารถค้นคืนข้อมูลได้อย่างมีประสิทธิภาพ

1.2.2 เพื่อเปรียบเทียบและค้นหาเทคนิคการค้นคืนข้อมูลที่เหมาะสมกับระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต

1.2.3 เพื่อศึกษาและพัฒนาเทคนิคที่จะช่วยเพิ่มประสิทธิภาพในการค้นคืนข้อมูลสำหรับ Moodle

1.2.4 เพื่อเป็นแนวทางสำหรับการพัฒนาเว็บแอพพลิเคชันที่สามารถค้นคืนข้อมูลได้อย่างมีประสิทธิภาพ

1.3 ขอบเขตการวิจัย

งานวิจัยนี้มุ่งเน้นที่จะศึกษาและพัฒนาเทคนิคการค้นคืนข้อมูล โดยการทำระบบ Caching ร่วมกับระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ซึ่งสรุปลักษณะสำคัญของเทคนิคการค้นคืนข้อมูลที่สนใจสำหรับงานวิจัย ดังต่อไปนี้

1.3.1 สามารถทำงานร่วมกับเว็บไซต์ที่มีการปรับปรุงเปลี่ยนแปลงเนื้อหาหรือรูปแบบของเว็บตลอดเวลา (Dynamic Web) และถูกพัฒนาขึ้นด้วยภาษา PHP ซึ่งในปัจจุบัน เว็บไซต์ส่วนใหญ่จะมีการพัฒนาเป็นแบบ Dynamic เพราะสามารถใช้งานโต้ตอบกับผู้ใช้ได้ทันที ซึ่งทำให้เว็บไซต์มีความน่าสนใจมากยิ่งขึ้น ลักษณะของเว็บไซต์ที่เป็นแบบ Dynamic นั้น ส่วนใหญ่จะมีการสร้างเว็บโดยใช้โปรแกรมภาษา PHP และฐานข้อมูล MySQL เป็นส่วนประกอบ

1.3.2 สามารถทำงานร่วมกับ Moodle ได้ เนื่องจาก Moodle เป็นระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่มีฟังก์ชันการทำงานที่เป็นที่ยอมรับ และได้รับความนิยมอย่างแพร่หลายจากผู้ใช้ทั่วโลก โดยงานวิจัยนี้จะนำ Moodle มาเป็นส่วนที่ใช้ทดสอบร่วมกับเทคนิคการค้นคืนที่พัฒนาขึ้น

1.3.3 ใช้การทำระบบ Caching ซึ่งเทคนิคดังกล่าวจะทำการเก็บข้อมูลหรือเอกสารที่มีผู้ใช้คนหนึ่งได้เรียกใช้งานแล้วให้อยู่ในตำแหน่งที่สามารถใช้ร่วมกันกับผู้ใช้คนอื่นได้ เมื่อผู้ใช้คนที่สองร้องขอข้อมูลหรือเอกสารชุดเดียวกัน ระบบ Caching ก็จะส่งข้อมูลดังกล่าวไปให้ผู้ใช้ โดยไม่ต้องร้องขอข้อมูลเดิมจากเซิร์ฟเวอร์ซ้ำอีก นั่นคือให้ผู้ใช้คนที่สองจะได้รับข้อมูลหรือเอกสารที่ถูกบันทึกไว้ตำแหน่งร่วม (Common Place) ของระบบ Caching แทน ซึ่งจะช่วยลดปริมาณการติดต่อระหว่างเซิร์ฟเวอร์และไคลเอนท์ และเพิ่มประสิทธิภาพในการค้นคืนข้อมูลได้

งานวิจัยนี้จะทำการศึกษาและพัฒนาเทคนิคการค้นคืนข้อมูลสำหรับระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต โดยการประยุกต์ใช้ระบบ Caching ซึ่งเป็นระบบที่ใช้ในการเก็บข้อมูลที่มีการเรียกใช้บ่อยเอาไว้ในระบบ เพื่อที่จะส่งข้อมูลดังกล่าวต่อไปให้กับผู้ที่ร้องขอข้อมูลคนต่อไปที่ต้องการข้อมูลเหมือนกัน โดยจะได้นำไปทดสอบการทำงานของเทคนิคที่พัฒนาขึ้นร่วมกับ Moodle บนระบบปฏิบัติการ SUTinsServer 5103 Plus Moodle (SUT Instant Server for Developer Plus Moodle) ซึ่งเป็นระบบปฏิบัติการลินุกซ์เซิร์ฟเวอร์พร้อมใช้สำหรับนักพัฒนา จากมหาวิทยาลัยเทคโนโลยีสุรนารี

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ทำให้เกิดการพัฒนาเทคนิคการค้นคืนข้อมูลที่มีประสิทธิภาพและมีความรวดเร็วขึ้น

1.4.2 สามารถนำเทคนิคดังกล่าว มาพัฒนาระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่สามารถค้นคืนข้อมูลอย่างมีประสิทธิภาพได้

1.4.3 สามารถนำเทคนิคดังกล่าวมาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพในการค้นคืนข้อมูลร่วมกับ Moodle ซึ่งเป็นระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่มีผู้ใช้เป็นจำนวนมาก

1.4.4 ทำให้การค้นคืนข้อมูลบนระบบที่มีข้อมูลขนาดใหญ่ เป็นไปได้ด้วยความรวดเร็วขึ้น เพราะไม่ต้องไปดึงข้อมูลจากระบบใหม่ทุกครั้ง

1.4.5 ทำให้การค้นคืนข้อมูลใช้เวลาน้อยลง คือ สามารถลดเวลาที่ใช้ในกระบวนการทั้งหมดลงได้

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะเป็นการนำเสนอวรรณกรรมและงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ โดยในหัวข้อ 2.1 จะกล่าวถึงเทคโนโลยีในการค้นคืนข้อมูลอย่างมีประสิทธิภาพจากระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต โดยนำเสนอเทคนิควิธีการค้นคืนข้อมูลที่ใช้เทคโนโลยี AJAX และเทคนิควิธีการค้นคืนข้อมูลแบบ Dynamic Web Cache ในหัวข้อ 2.2 จะกล่าวถึงคำจำกัดความและเครื่องมือในการพัฒนาเว็บไซต์โดยจะแนะนำเครื่องมือต่าง ๆ ที่ใช้ในการพัฒนาเว็บไซต์แบบไดนามิกที่ได้มีการพัฒนาเกี่ยวกับการค้นคืนข้อมูลให้กับระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ในหัวข้อ 2.3 จะกล่าวถึงระบบฐานข้อมูลที่นำมาใช้ในการทดลอง ในหัวข้อ 2.4 จะกล่าวถึงโปรแกรมภาษา PHP ซึ่งเป็นโปรแกรมภาษาที่ได้นำมาพัฒนาร่วมกับการทดลองนี้ ในหัวข้อ 2.5 จะกล่าวถึง Moodle ซึ่งเป็นระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ที่ได้นำมาทดลองร่วมกันกับระบบการค้นคืนข้อมูลดังกล่าว และในหัวข้อ 2.6 ได้กล่าวถึงงานวิจัยที่เกี่ยวข้อง

2.1 เทคโนโลยีการค้นคืนข้อมูล

ปัจจุบัน ลักษณะการทำงานแบบไคลเอนท์กับเซิร์ฟเวอร์ (Client - Server) ถูกนำมาใช้งานอย่างแพร่หลายในลักษณะการติดต่อสื่อสารผ่านทางเว็บเบราว์เซอร์ โดยไคลเอนท์จะร้องขอข้อมูลที่ต้องการจากเซิร์ฟเวอร์ เมื่อไคลเอนท์ได้รับการตอบสนองจากเซิร์ฟเวอร์แล้วจะต้องทำการรีเฟรชหน้าเว็บเพื่อแสดงข้อมูลที่รับเข้ามาบนหน้าเว็บนั้น ดังนั้นการโหลดข้อมูลและการรีเฟรชหน้าเว็บจึงเป็นสิ่งที่หลีกเลี่ยงไม่ได้ เป็นผลให้การทำงานทางด้านฝั่งไคลเอนท์ใช้เวลามากในการโหลดข้อมูลและการรีเฟรชหน้าเว็บ ดังนั้นจึงมีผู้คิดค้นและพัฒนาเทคนิคในการค้นคืนข้อมูลโดยจำกัดการโหลดข้อมูลและการรีเฟรชหน้าเว็บให้น้อยที่สุด เช่น เทคโนโลยี AJAX ซึ่งใช้เทคนิคการรีเฟรชข้อมูลเฉพาะจุดที่มีการเปลี่ยนแปลงบนหน้าเว็บเท่านั้น และเทคนิคการทำ Dynamic Web Cache ซึ่งจะเก็บข้อมูลที่เคยถูกร้องขอไว้ที่ระบบ Cache เมื่อมีผู้ใช้ร้องขอข้อมูลเดียวกันนี้ ระบบก็จะดึงข้อมูลที่เก็บไว้บนหน่วยความจำที่สามารถเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ และส่งไปให้กับผู้ใช้ได้ทันที โดยไม่ต้องทำการค้นหาข้อมูลเหล่านั้นจากฐานข้อมูลบนเซิร์ฟเวอร์ซ้ำอีก ซึ่งหากฐานข้อมูลมีขนาดใหญ่มาก การเข้าถึงข้อมูลก็จะยิ่งช้า และไม่ทันต่อการใช้งาน

2.1.1 AJAX (Asynchronous JavaScript And XML)

Jeese Jams Garett (www, 2005) เป็นผู้ตั้งชื่อและได้กล่าวว่า เอแจ็กซ์ (AJAX - Asynchronous JavaScript And XML) เป็นเทคนิคในการพัฒนาเว็บแอปพลิเคชันเพื่อให้ความสามารถโต้ตอบกับผู้ใช้ได้ดีขึ้น โดยการรับส่งข้อมูลในฉากหลัง ทำให้ทั้งหน้าไม่ต้องโหลดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ซึ่งช่วยทำให้เพิ่มการตอบสนอง ความรวดเร็ว และการใช้งานโดยรวม

AJAX นั้นไม่ใช่เทคโนโลยีใหม่ แต่เป็นเทคนิคที่ได้ใช้เทคโนโลยีหลายอย่างที่มีอยู่แล้วรวมกันดังต่อไปนี้

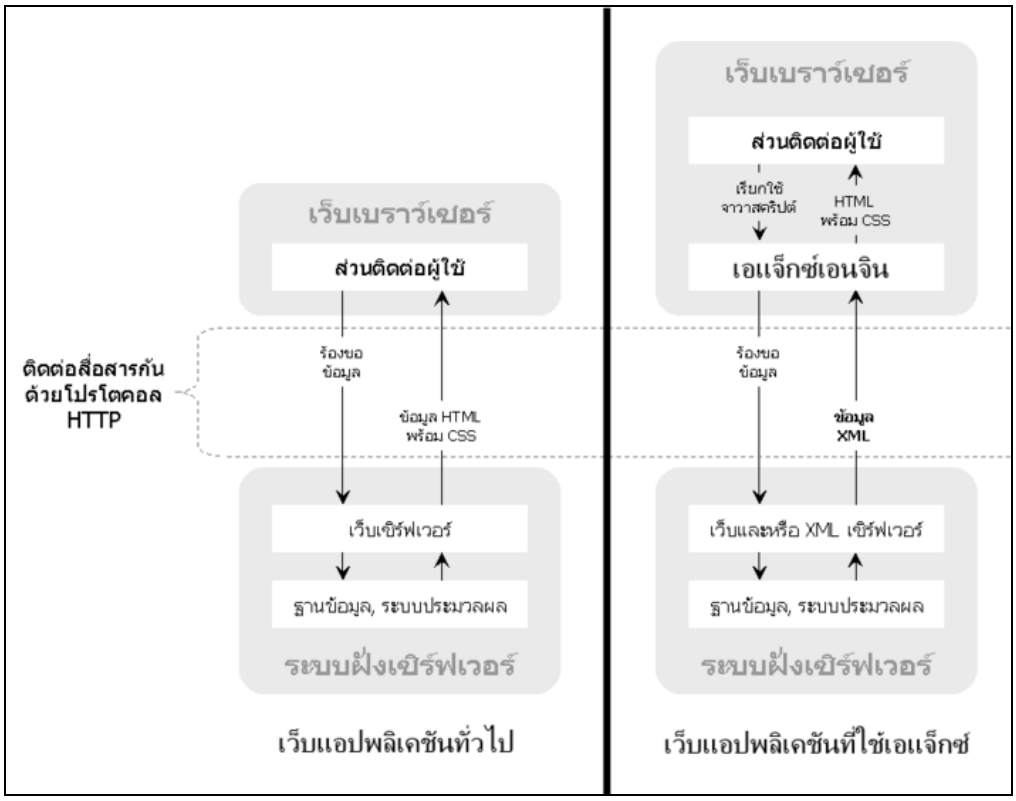
XHTML (หรือ HTML) และ CSS ใช้ในการแสดงผลลัพท์และรูปแบบข้อมูล

ECMA สคริปท์ เช่น จาวาสคริปท์ ในการเข้าถึง Document Object Model (DOM) โดยใช้ในการแสดงข้อมูลที่มีการเปลี่ยนแปลงหรือโต้ตอบกับผู้ใช้

XMLHttpRequest จะใช้ในการแลกเปลี่ยนข้อมูลแบบ Asynchronously กับเว็บเซิร์ฟเวอร์

XML เป็นรูปแบบที่ใช้ในการแลกเปลี่ยนข้อมูล สำหรับรูปแบบในการแลกเปลี่ยนข้อมูลอื่น ๆ ที่สามารถใช้ได้ เช่น HTML, JSON, EBML, หรือ ข้อความที่เป็น Text

AJAX Engine ทำหน้าที่เป็นตัวกลางระหว่างไคลเอนท์และเซิร์ฟเวอร์ ฉะนั้นเมื่อไคลเอนท์มีการร้องขอ (Request) แทนที่จะส่ง HTTP Request ไปยังเซิร์ฟเวอร์โดยตรง ไคลเอนท์จะส่ง JavaScript Call ไปยัง AJAX Engine เพื่อโหลดข้อมูลที่ผู้ใช้ต้องการ และหาก AJAX Engine ต้องการข้อมูลเพิ่มเติมในการตอบสนอง (Response) ต่อผู้ใช้ AJAX Engine ก็จะส่ง Request ไปยังเซิร์ฟเวอร์โดยใช้ XML ดังรูปที่ 2.1 แสดงการเปรียบเทียบการทำงานแบบเดิม กับ AJAX



รูปที่ 2.1 แสดงการเปรียบเทียบการทำงานแบบเดิม กับ AJAX (วิกิพีเดีย, www, 2550)

โดยส่วนประกอบจำเป็นขั้นพื้นฐานที่ขาดไม่ได้ใน AJAX ได้แก่ HTML/XHTML DOM และ JavaScript

วิกิพีเดีย (www, 2550) ได้มีผู้กล่าวถึงประวัติความเป็นมาของเทคโนโลยี AJAX โดยเริ่มต้นเมื่อปี ค.ศ. 2002 ไมโครซอฟท์ได้ทำการคิดค้น XMLHttpRequest ขึ้นมาเพื่อเป็นทางเลือกในการเขียนโปรแกรมบนเว็บเพจ เพื่อใช้ติดต่อกับเว็บเซิร์ฟเวอร์ ในการแลกเปลี่ยนข้อมูล ซึ่งในขณะนั้นมีแต่เพียงอินเทอร์เน็ตเอกซ์พลอเรอร์เท่านั้นที่มีความสามารถนี้ ต่อมาเว็บเบราว์เซอร์อื่น ๆ เช่นจาก มอซิลลาไฟร์ฟอกซ์ได้นำแนวคิดของ XMLHttpRequest ไปใส่ในเบราว์เซอร์ของตนด้วย จึงเริ่มทำให้มีการใช้อย่างกว้างขวางขึ้น จนปัจจุบันได้กลายเป็นมาตรฐานที่ทุกเว็บเบราว์เซอร์ต้องมี

ในตอนแรกนั้นไมโครซอฟท์เป็นผู้ที่ได้นำ XMLHttpRequest โดยใช้ใน Outlook Web Access ที่มาพร้อมกับ Microsoft Exchange Server 2000 ต่อมาเว็บไซต์อย่าง Google ได้เปิดบริการใหม่ชื่อ Gmail ซึ่งใช้ XMLHttpRequest เป็นหัวใจหลักในการดึงข้อมูลจากเว็บเซิร์ฟเวอร์ จึงทำให้แนวคิดและเทคนิคการพัฒนาเว็บแอปพลิเคชันด้วย AJAX เริ่มรู้จักเป็นที่กว้างขวางขึ้น จนปัจจุบันถือว่าเป็นหนึ่งในหัวใจหลักของแนวคิดเรื่อง Web 2.0 ไปด้วย

และต่อมา Google ได้ใช้การติดต่อสื่อสารแบบ Asynchronous เพื่อเป็นรากฐานที่ทำให้รู้จักกับ AJAX กันอย่างแพร่หลาย การทำงานแบบไคลเอนท์-เซิร์ฟเวอร์ ถูกนำมาใช้งานเป็นจำนวนมาก เช่น การติดต่อกับฐานข้อมูลที่เซิร์ฟเวอร์ หรือการให้บริการทางอินเทอร์เน็ต ซึ่ง Google เป็นผู้พัฒนาและการทดสอบ AJAX จึงสังเกตได้ว่า ผลผลิตใหญ่ของ Google ในช่วงต้นปี 2005 จึงเป็นการนำ AJAX มาประยุกต์ใช้งาน เช่น Gmail, GoogleMap, GoogleSuggest และ GoogleGroup เป็นต้น

เนื่องจากการประยุกต์ใช้งานในปัจจุบันนี้ มีหลักการที่ทำงานแล้วเกิดการสูญเสียเวลาและทรัพยากรของผู้ใช้ในการรอคอยการทำงานต่าง ๆ ทำให้ผู้ใช้ต้องหยุดรอ และทำให้การทำงานของผู้ใช้เป็นไปอย่างไม่ต่อเนื่อง โดยปัญหาที่เกิดขึ้นมีดังต่อไปนี้

1) แบบอย่างผลกระทบต่อผู้ใช้ หรือที่เรียกว่า "Click, wait, and refresh"

การที่เบราว์เซอร์ตอบสนองต่อการทำงานของผู้ใช้ โดยจะทิ้งหน้าเว็บที่แสดงอยู่ในขณะนั้น แล้วไปทำการส่ง HTTP Request กลับไปยังเซิร์ฟเวอร์แทน ซึ่งทำให้ผู้ใช้ไม่สามารถทำอะไรได้เลยในขณะนั้น นอกจากการรอคอย เมื่อเซิร์ฟเวอร์ทำการประมวลเสร็จก็จะส่งหน้า HTML กลับมายังเบราว์เซอร์ ต่อจากนั้นเบราว์เซอร์ก็จะรีเฟรชและแสดงหน้า HTML หน้าใหม่ ผู้ใช้จึงจะสามารถใช้งานต่อไปได้

จะเห็นว่าผู้ใช้ต้องหยุดรอสำหรับการประมวลผลของเซิร์ฟเวอร์ และการรีเฟรชหน้า HTML ใหม่ทั้งหน้า ซึ่งเป็นสิ่งที่ไม่มีประสิทธิภาพในเชิง Dynamic ของการทำงานบนเว็บแอพพลิเคชัน

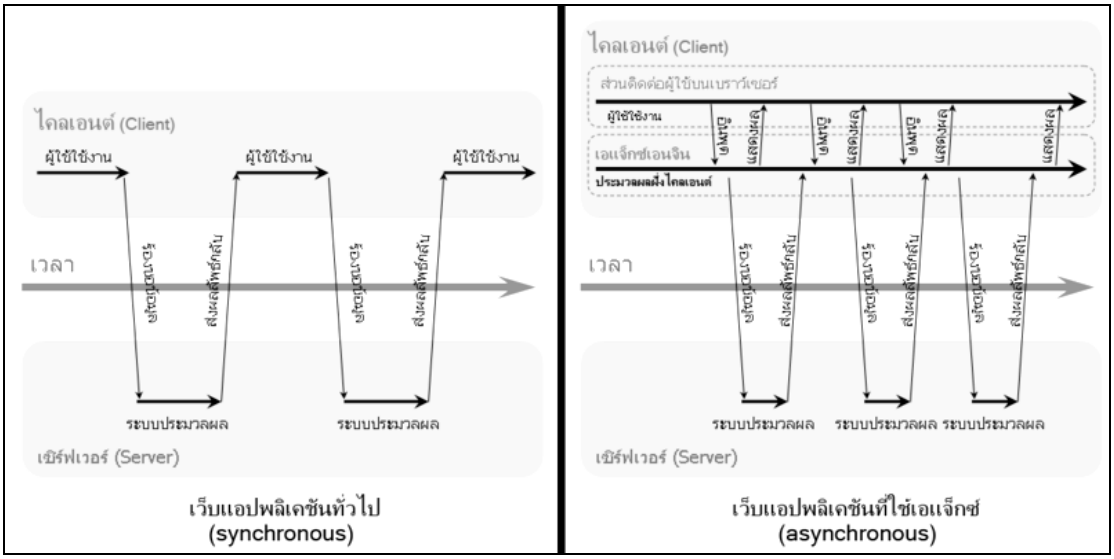
2) วิธีการติดต่อแบบ Synchronous หรือที่เรียกว่า "request/response"

การที่เบราว์เซอร์เริ่มทำการร้องขอข้อมูลจากเซิร์ฟเวอร์ และเซิร์ฟเวอร์จะทำการตอบสนองเฉพาะการร้องขอที่เบราว์เซอร์ร้องขอมาเท่านั้น โดยเซิร์ฟเวอร์จะไม่สามารถส่งข้อมูลได้ถ้าเบราว์เซอร์ไม่ได้ร้องขอข้อมูลในขณะนั้น ซึ่งถือเป็นการติดต่อสื่อสารแบบทิศทางเดียว วงจรแบบนี้เรียกว่า Request/Response แบบ Synchronous คือ การทำงานแบบประสานจังหวะระหว่างเบราว์เซอร์กับเซิร์ฟเวอร์ ทำให้เกิดความล่าช้าในการทำงานทำให้ผู้ใช้ทำอะไรไม่ได้อีก นอกจากการรอการตอบสนองกลับมาจากเซิร์ฟเวอร์ หลังจากเซิร์ฟเวอร์ประมวลผลเสร็จ ดังรูปที่ 2.2 แสดงการเปรียบเทียบระหว่างการสื่อสารแบบ Synchronous กับแบบ Asynchronous

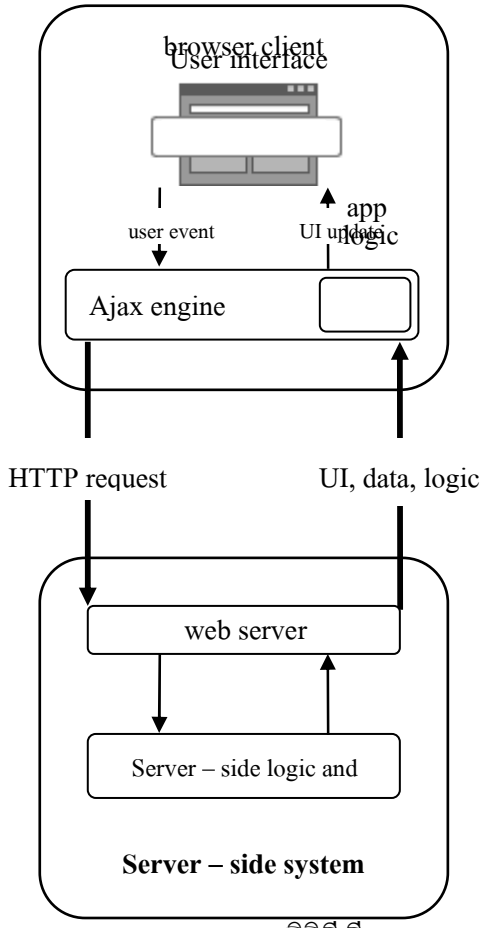
มุมมองของโครงสร้างทาง Software ของ AJAX ต่างจากเว็บแอพลิเคชันในปัจจุบันนี้ เนื่องจากมีการเพิ่ม Engine ทางฝั่งไคลเอนท์

จากรูปที่ 2.2 AJAX Engine นี้ จะอยู่ระหว่าง ส่วนติดต่อผู้ใช้กับเซิร์ฟเวอร์ ซึ่งจะมองว่าเป็นการทำงานที่ไคลเอนท์ การทำงานต่าง ๆ ของผู้ใช้โปรแกรมจะไปเรียก AJAX Engine ตัว

นี้ขึ้นมา แทนที่การร้องขอหน้าเว็บจากเซิร์ฟเวอร์โดยตรง และจะใช้โครงสร้างข้อมูลแบบ XML ในการขนย้ายข้อมูลระหว่างเซิร์ฟเวอร์กับ AJAX Engine เมื่อเบราว์เซอร์ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์



รูปที่ 2.2 การเปรียบเทียบระหว่างการสื่อสารแบบ Synchronous กับแบบ Asynchronous (วิกิพีเดีย, www, 2550)



รูปที่ 2.3 AJAX Architecture (วิกิพีเดีย, www, 2550)

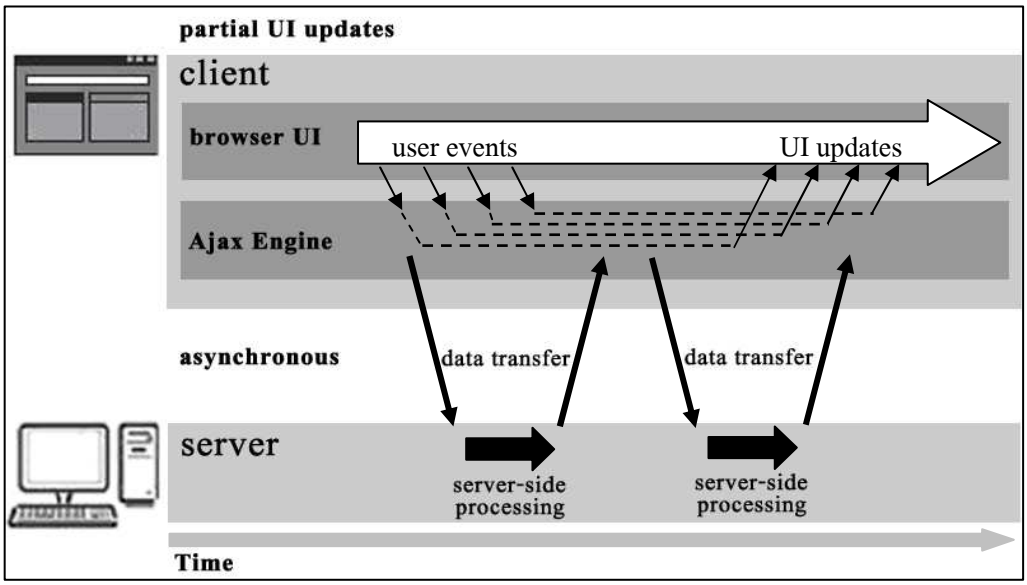
AJAX สามารถช่วยลดการติดต่อระหว่างไคลเอนท์กับเซิร์ฟเวอร์ ในช่วงของการโหลดหน้าเว็บ เบราว์เซอร์จะโหลดข้อมูลจาก AJAX Engine แทนการร้องขอข้อมูลจากเซิร์ฟเวอร์โดยตรง ดังนั้น AJAX จะทำหน้าที่ทั้งการนำส่งส่วนติดต่อกับผู้ใช้และติดต่อไปยังเซิร์ฟเวอร์แล้ว AJAX Engine จะอนุญาตให้การกระทำต่าง ๆ ในเว็บแอพพลิเคชันเป็นแบบ Asynchronous คือความเป็นอิสระในการติดต่อไปยังเซิร์ฟเวอร์นั่นเอง ดังนั้นผู้ใช้จะไม่พบกับเบราว์เซอร์หน้าว่างเปล่าในขณะที่กำลังรอข้อมูลจากเซิร์ฟเวอร์อีกต่อไป ดังนั้นวัตถุประสงค์ในการใช้เทคโนโลยี AJAX จึงถูกคิดค้นขึ้นเพื่อแก้ไขปัญหาต่าง ๆ ต่อไปนี้

1) แบบอย่างผลกระทบต่อผู้ใช้ หรือ "Partial screen update" replaces the "click, wait, and refresh"

ร้องขอไปยังเซิร์ฟเวอร์จะถูกจัดเป็นข้อมูลใหม่เมื่อถูกเปลี่ยนแปลงแล้ว การหยุดชะงักของส่วนติดต่อกับผู้ใช้จึงไม่เกิดขึ้น เพราะหน้าเว็บนั้นยังคงถูกแสดงอยู่และสามารถใช้งานได้ โดยปราศจากการหยุดชะงักการทำงานของผู้ใช้ การเปลี่ยนแปลงหน้าเว็บบางส่วนสามารถทำให้หน้าเว็บทำงานต่อไปได้ ถึงจะไม่ใช่ทั้งหมด แต่อย่างน้อยก็ทำให้การทำงานไม่หยุดชะงักได้

2) การติดต่อแบบ Asynchronous แทนที่แบบ "synchronous request/response"

สำหรับ AJAX การ Request/Response จะทำแบบ Asynchronous ซึ่งคือการติดต่อสื่อสารกับเซิร์ฟเวอร์แบบอิสระโดยทำการลดการติดต่อระหว่างเบราว์เซอร์กับเซิร์ฟเวอร์ ผลที่ได้ก็คือผู้ใช้สามารถใช้งานเว็บแอพลิเคชันได้ ในขณะที่ไคลเอนท์ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์อยู่เบื้องหลัง (เป็นการทำงานแบบพร้อ มกัน โดยมองเป็น 2 ฝั่ง เช่น หน้าร้านกับหลังร้าน) เมื่อข้อมูลเดินทางมาถึงเบราว์เซอร์ ก็จะเปลี่ยนแปลงหน้าเว็บในส่วนที่ผู้ใช้ร้องขอข้อมูลใหม่ ในขณะที่ข้อมูลส่วนอื่น ๆ ยังคงแสดงต่อไป (วิชา ศิริธรรมจักร์, 2547)



รูปที่ 2.4 การทำงานแบบ Asynchronous และการเปลี่ยนแปลงหน้าเว็บแบบบางส่วน (วิกิพีเดีย, www, 2550)

Greg Murray (2006) ได้เปรียบเทียบถึงข้อดี และข้อเสียของ AJAX ไว้ดังนี้

ข้อดีของ AJAX

1) เป็นเทคโนโลยีใหม่ที่ตอบสนองต่อผู้ใช้ได้อย่างรวดเร็วเนื่องจากการเปลี่ยนแปลงแบบบางส่วน

2) ผู้ใช้ไม่ต้องหยุดรอการประมวลผลของเซิร์ฟเวอร์ เนื่องจากเป็นการติดต่อแบบ Asynchronous

3) รองรับกับเบราว์เซอร์หลัก ๆ ที่สามารถใช้ JavaScript ได้

4) ทำให้การประมวลผลที่เซิร์ฟเวอร์มีความรวดเร็วมากขึ้น เนื่องจากการประมวลผลที่เซิร์ฟเวอร์ลดลง

5) ไม่ต้องทำการติดตั้ง หรือใช้ Plugs-in

6) ไม่ยึดติดกับ Platform หรือภาษาที่ใช้ในการเขียนโปรแกรม

ข้อเสียของ AJAX

1) เกิดจากความเข้าใจผิดของผู้ใช้ เนื่องจากความเข้าใจผิดของปุ่มย้อนกลับ (Back/Button) โดยปกติแล้วผู้ใช้มักจะใช้ปุ่มย้อนกลับเพื่อกลับไปหน้าเดิม แต่ใน AJAX จะไม่เป็นอย่างนั้น นักพัฒนาเว็บได้ช่วยกันหาทางออกกับปัญหานี้ โดยจะใช้ IFRAMEs เพื่อทำให้เกิดการจดจำหน้าที่แล้วโดยเบราว์เซอร์ อีกวิธีหนึ่งคือ การทำ Bookmark ของผู้ใช้ เนื่องจากเบราว์เซอร์จะจดจำหน้าเว็บที่ Bookmark ผ่านทาง URL เท่านั้น ทางออกส่วนใหญ่อยู่ที่การผสม URL (URL Fragment Identifier) การเติมบางส่วนของ URL ตามหลังเครื่องหมาย # ตามมาตรฐาน rfc2396 และ rfc3896 เพื่อให้ URL เปลี่ยนไปทุกครั้งที่มีการคลิ๊ก วิธีนี้ทำให้ผู้ใช้สามารถย้อนกลับมาที่หน้าเว็บนั้นได้อีกครั้งตามต้องการ โดยเบราว์เซอร์ที่ใช้จะต้องอนุญาตให้ JavaScript นั้นทำการปรับปรุงค่าในช่อง URL ทางออกเช่นนี้ยังช่วยแก้ปัญหาความเข้าใจผิดของผู้ใช้ในเรื่องปุ่มย้อนกลับได้อีกด้วย

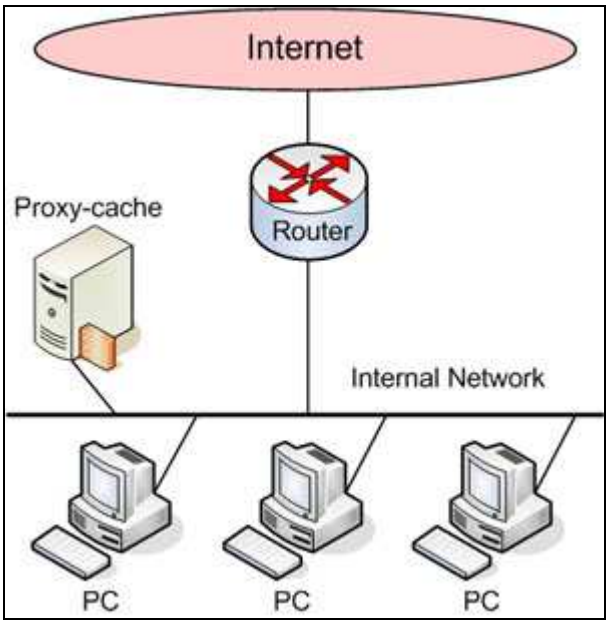
2) การตอบสนอง หรืออาการหน่วงของเซิร์ฟเวอร์ คือระยะเวลาตั้งแต่ผู้ใช้ร้องขอจนถึงเซิร์ฟเวอร์ตอบรับ ต้องพิจารณาอย่างรอบคอบเมื่อมีการพัฒนาโปรแกรมร่วมกับ AJAX ถ้าเบราเซอร์ไม่มีความสามารถในการแสดงผล และการเขียนรับ XMLHttpRequest Object อย่างดีแล้ว ผู้ใช้อาจจะพบการตอบสนองที่เชื่องช้าของหน้าเว็บ ที่การแสดงผลให้ผู้ใช้เห็นว่าขณะนี้ระบบกำลังประมวลผลอะไรอยู่ โดยอาจแสดงผลข้อมูลบางส่วนออกมาก่อน

3) ความน่าเชื่อถือของ JavaScript ถึงแม้ว่า AJAX ไม่ต้องการการติดตั้งอะไรเพิ่มจากเบราว์เซอร์ แต่ผู้ใช้จำเป็นต้องเปิดให้เบราว์เซอร์นั้นยอมรับ JavaScript ส่วนผู้ใช้ IE6 หรือเก่ากว่านั้นจะต้องเปิด ActiveX ในเบราว์เซอร์อีกด้วย เพราะว่า XMLHttpRequest นั้นถูกกำหนดให้ทำการประมวลผลคำสั่งด้วย ActiveX บน IE สำหรับบน IE7 นั้นได้แก้ปัญหานี้แล้ว เนื่องจากเป็นเว็บประเภท DHTML ทำให้ผู้พัฒนาเว็บที่ใช้เทคโนโลยี AJAX ต้องทำการทดสอบอย่างดีเพื่อไม่ให้พบปัญหาการแสดงผลที่แตกต่างกันเมื่อทำงานบนเบราว์เซอร์และระบบปฏิบัติการที่แตกต่างกัน จึงมีการพัฒนาไลบรารีหลายตัวเพื่อช่วยให้ AJAX นั้นมีความสามารถสูงขึ้น และช่วยลดความยุ่งยากใน

การแก้ปัญหานี้ได้ รวมทั้งการพัฒนาเทคนิคต่าง ๆ ที่ช่วยลดความยุ่งยากในการออกแบบโปรแกรม และเพิ่มทางเลือกอื่น ๆ นอกเหนือจากการใช้ JavaScript เพียงอย่างเดียว

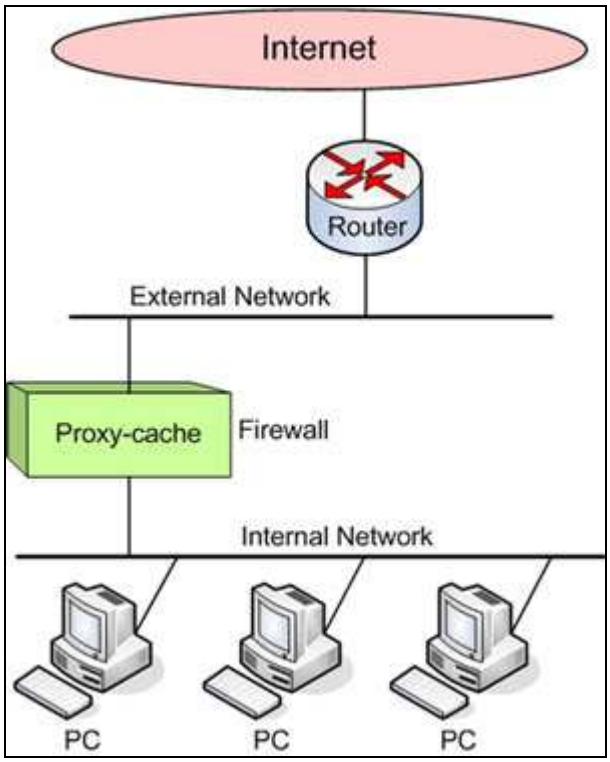
2.1.2 ระบบไดนามิกเว็บแคช (Dynamic Web Caching System)

David Guerrero (www, 1999) ได้กล่าวว่า Web Caching เป็นการปรับปรุงการใช้งานอินเทอร์เน็ตให้เร็วขึ้นและประหยัดแบนด์วิดธ์ที่เรียกไปยังอินเทอร์เน็ต ด้วยการใช้พร็อกซี่เซิร์ฟเวอร์ (Proxy Server) ที่ทำหน้าที่เก็บเว็บที่เคยเรียกแล้วไว้ใน Proxy Server แม้ว่าหลายคนจะไม่รู้จักคำว่า Web Caching แต่จริง ๆ แล้วเคยมีการใช้งาน Web Caching ที่จัดการโดยเบราว์เซอร์แล้ว ซึ่งเบราว์เซอร์ส่วนใหญ่จะมีวิธีการเก็บเอกสารและข้อมูลต่าง ๆ ที่ได้มาจากเว็บไว้ในหน่วยความจำ ในแต่ละครั้งที่มีการคลิ๊กปุ่มย้อนกลับของเบราว์เซอร์เพื่อกลับไปยังหน้าเว็บที่เคยผ่านมาแล้ว หน้าเว็บดังกล่าวจะถูกแสดงผลโดยดึงข้อมูลจากหน่วยความจำบน Proxy Server ซึ่งไม่จำเป็นต้องไปดึงมาจากเว็บเซิร์ฟเวอร์อีก ลักษณะแบบนี้เป็นการทำ Caching ระดับแรก (First Level of Caching) แนวคิดพื้นฐานของการทำ Caching คือการเก็บเอกสารที่ผู้ใช้คนหนึ่งได้เรียกใช้งานแล้วให้อยู่ในตำแหน่งที่สามารถใช้ร่วมกันได้ เพื่อไม่ให้ผู้ใช้คนที่สองที่ต้องการเรียกใช้งานเอกสารชุดเดียวกันไปเรียกโดยตรงจากเซิร์ฟเวอร์อีก นั่นคือให้ผู้ใช้คนที่สองสามารถดึงเอกสารมาจากตำแหน่งร่วม (Common Place) แทน ซึ่งลักษณะดังกล่าวนี้เป็นสิ่งที่จำเป็นมาก ยกตัวอย่างเช่น เมื่อองค์กรขนาดใหญ่แห่งหนึ่งซึ่งตั้งอยู่ในทวีปยุโรป และมีปริมาณการใช้งานขาเข้าส่วนใหญ่มาจากอีกด้านหนึ่งของแอตแลนติกด้วยระบบเครือข่ายที่มีความเร็วต่ำ เมื่อประยุกต์ใช้วิธีดังกล่าว จะสามารถเรียกใช้งานเอกสารที่ต้องการจากเว็บได้อย่างรวดเร็ว และลดปริมาณการใช้งานที่จะออกไปยังอินเทอร์เน็ต ซึ่งองค์กรขนาดกลางที่มีผู้ใช้ขนาด 50 ถึง 100 นั้น การทำ Caching สามารถให้บริการได้ถึง 60 เปอร์เซ็นต์ของ URL ที่ร้องขอจาก Local Cache ความแตกต่าง Browser-Cache กับเซิร์ฟเวอร์ Proxy-Cache คือ Browser-Cache จะเป็นการ Cache สำหรับผู้ใช้คนเดียวและอยู่ในตำแหน่งที่เป็น Final User Workstation ในขณะที่ Proxy-Cache เป็นโปรแกรมที่ทำหน้าที่เป็นตัวแทนจำนวนมากของเว็บเบราว์เซอร์ไคลเอนท์ โดยอนุญาตให้ไคลเอนท์เครื่องหนึ่งอ่านเอกสารที่เคยถูกร้องขอโดยคนอื่นก่อนหน้านี้ ซึ่ง Proxy-Cache เซิร์ฟเวอร์ที่มักจะวางอยู่ระหว่าง Local Network กับอินเทอร์เน็ต โดยเบราว์เซอร์ทั้งหมดจะร้องขอเอกสารจากเซิร์ฟเวอร์ Proxy รูปที่ 2.5 แสดงโครงสร้างของ Proxy-Cache เซิร์ฟเวอร์ ชนิดนี้



รูปที่ 2.5 Proxy-Cache เซิร์ฟเวอร์ Network Configuration (IDESIGN.IN.TH, www, 2005)

Proxy-Cache ไม่เป็นเพียงทางออกเฉพาะสำหรับการแก้ปัญหาในเรื่องของแบนด์วิดธ์เท่านั้น แต่ยังเป็นสิ่งที่จำเป็นเมื่อองค์กรต้องการไฟร์วอลล์เพื่อประกันความปลอดภัยของเครือข่าย ซึ่งในกรณีนี้ Proxy-Cache จะอยู่บนเซิร์ฟเวอร์ที่เบราว์เซอร์ของทุกไคลเอนท์สามารถเข้าถึงได้ แต่จะเป็นตัวแยกเครื่องไคลเอนท์ทั้งหมดออกจากเครือข่ายอินเทอร์เน็ต โดยเซิร์ฟเวอร์แบบนี้จะต้องมี 2 จุดเชื่อมต่อสำหรับเครือข่าย ซึ่งที่ทำหน้าที่ในการเชื่อมต่อกับเครือข่ายภายในและเชื่อมต่อกับเครือข่ายภายนอก โดยการเข้ามาจากเครือข่ายอินเทอร์เน็ตสามารถเข้าถึงได้เฉพาะเครือข่ายภายนอกเท่านั้น การใช้งานเซิร์ฟเวอร์ Proxy-Cache แบบนี้ ควรจะอนุญาตให้เฉพาะเครือข่ายภายในเท่านั้นที่สามารถใช้งาน Caching ได้

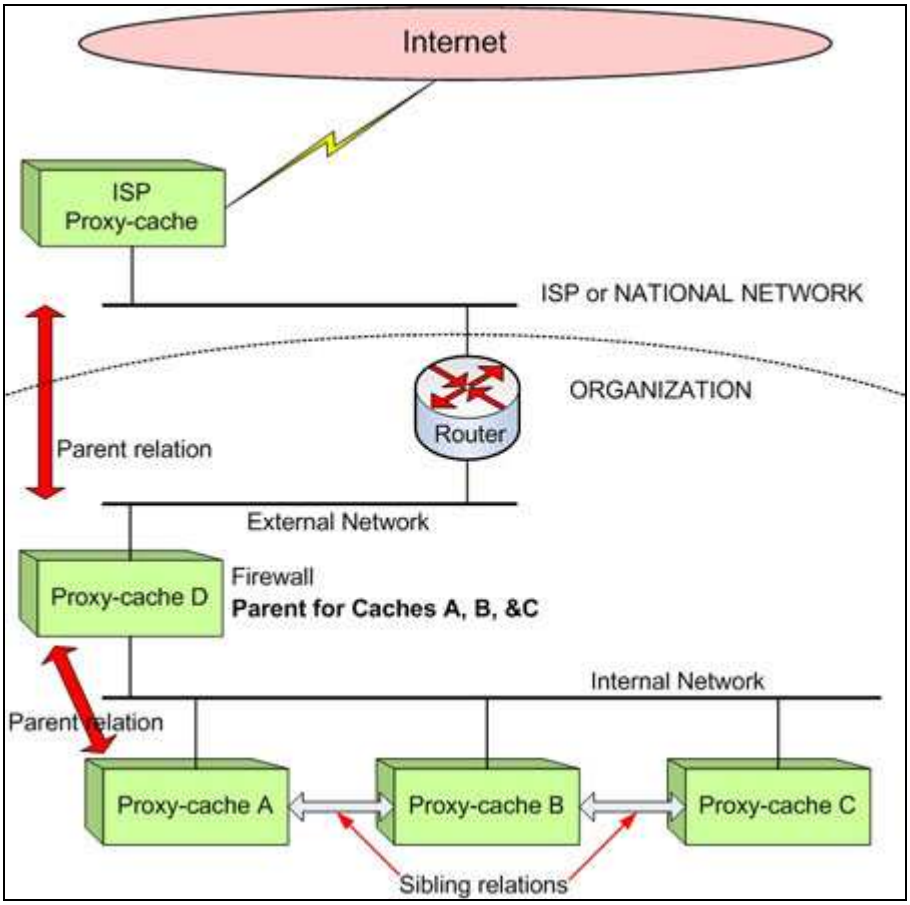


รูปที่ 2.6 Proxy-Cache Network Configuration with Firewall (IDESIGN.IN.TH, www, 2005)

David Guerrero (1999) ได้ให้ความหมายของ Multi-Level Web Caching ว่าเป็นอีกก้าวหนึ่งของความก้าวหน้าในการทำ Web Caching คือการทำ Cache แบบลำดับชั้น (Cache Hierarchy) ซึ่งในการเชื่อมต่อใช้งานอาจจะมีการเชื่อมต่อที่เป็นลำดับชั้นและมีเซิร์ฟเวอร์ Proxy-Cache ตั้งแต่ 2 ตัวขึ้นไป โดย Proxy-Cache หนึ่ง ๆ อาจจะมีบทบาทหน้าที่ที่แตกต่างกันสองแบบในแต่ละชั้น โดยจะขึ้นอยู่กับรูปแบบของเครือข่าย (Network Topology) นโยบายของ ISP และทรัพยากรระบบ ซึ่ง Cache ที่เป็นแบบข้างเคียง (Neighbor Cache หรือ Sibling Cache) จะให้บริการเฉพาะเอกสารที่มีอยู่ในเครื่องตัวเองแล้วเท่านั้น ส่วน Cache ที่อยู่ในลำดับเหนือกว่า (Parent Cache) สามารถให้บริการได้ทั้งจากการดึงมาจาก Cache ตัวอื่น ๆ ที่อยู่ในลำดับสูงกว่าหรือจากแหล่งกำเนิด (Source) ขึ้นอยู่กับว่า Cache ตัวนี้มีการเชื่อมต่อไปยัง Parent Cache ตัวอื่นอีกหรือไม่ หรือมี Sibling Cache อยู่ในระดับเดียวกันหรือไม่ โดย Parent Cache ควรจะถูกใช้เมื่อไม่สามารถที่จะดึงเอกสารจาก Cache ในระดับเดียวกันได้แล้ว

การเลือก Cache Topology ที่ดีเป็นสิ่งที่สำคัญมาก เพราะสามารถลดปริมาณการใช้งานบนเครือข่ายได้ องค์กรหนึ่ง ๆ สามารถที่จะเลือกให้มีหลาย Sibling Cache ในแต่ละหน่วยงานย่อย และให้มี Parent Cache เพียงหนึ่ง Parent Cache ไว้ในส่วนงานที่อยู่ใกล้กับเครือข่าย

อินเทอร์เน็ต ซึ่ง Parent Cache ดังกล่าวนี้สามารถจะตั้งค่าให้มีการร้องขอเอกสารจาก Parent Cache ตัวอื่นที่อยู่เหนือกว่าได้ เช่น ที่ ISP ข้อตกลงในการจัดตั้ง Sibling Cache หรือ Parent Cache ระหว่างผู้ใช้บริการกับ ISP สามารถลดปริมาณการใช้งานที่มีมากเกิน (Overload) ในส่วนของการเชื่อมต่อได้ หรือจัดทำปริมาณการใช้งานที่เป็นเว็บผ่านเส้นทางอื่นที่ไม่ใช่ IP Traffic ทั่วไป การทำ Web Caching ควรจะพิจารณาในส่วนของ Application-Level วิธีการทำเราติ้ง ซึ่งใช้ ICP (Internet Cache Protocol) เป็นโปรโคคอลหลัก



รูปที่ 2.7 Multi-level Web Caching Organization (IDESIGN.IN.TH, www, 2005)

ICP เป็นโปรโตคอลที่ใช้สำหรับการสื่อสารระหว่าง Web Cache เป็น Lightweight Protocol ที่อยู่ในส่วนบนของ UDP โดย ICP จะถูกใช้เพื่อหาตำแหน่งของ Web Object ใน Cache ที่อยู่ข้างเคียง การถ่ายโอน Object ระหว่าง Cache ส่วนใหญ่จะทำด้วย TCP-Based HTTP Protocol

ในการหาตำแหน่งที่ดีที่สุดสำหรับการดาวน์โหลด Object นั้น ระบบ Cache จะส่ง ICP Request Packet ไปยัง Sibling Cache และ Parent Cache ทุกตัว แล้ว Cache เหล่านี้ก็จะส่ง ICP ตอบกลับมาด้วย HIT Code หรือ MISS Code ซึ่ง HIT หมายถึง Cache ตัวนี้มี Object และตกลงที่จะให้บริการ ส่วน MISS หมายถึงไม่มี Object ที่ต้องการ ดังนั้น Cache ตัวที่เป็นผู้ร้องขอก็จะรู้ว่าใครมี Object ที่ตัวเองต้องการและมันจะเอาข้อมูลนี้รวมกับตัวประกอบอื่น ๆ เช่น Round-Trip Time หรือการตอบสนองในแต่ละครั้ง เพื่อที่จะเลือกได้ว่าจะร้องขอผ่าน HTTP ไปยัง Cache ตัวใด ถ้า Cache ตอบกลับมาเป็น MISS Packet ตัว Cache ที่เป็นผู้ร้องขอก็จะทำการร้องขอเอกสารจาก Parent Cache กระบวนการร้องขอและตอบกลับ ICP (ICP Request/Reply Exchange) ควรจะเกิดขึ้นในช่วงสั้น ๆ มิฉะนั้นจะทำให้เพิ่มเวลาแฝง (Latency Time) ให้กับเบราว์เซอร์ แต่เวลาตรงนี้ผู้ใช้มักจะไม่สามารถสังเกตได้เพราะมีระยะเวลาไม่มากนักเมื่อเทียบกับเวลาที่แสดงผล

ถ้า Object ที่ได้ร้องขอผ่าน ICP มีขนาดเล็กเพียงพอ Object นี้ก็สามารถถูกรวมอยู่ใน ICP HIT Reply เหมือนกับ HTTP Redirect แต่การทำแบบนี้ไม่ค่อยมีกันมากนัก การใช้งาน ICP ที่กล่าวมานี้จะใช้เมื่อระบบเครือข่ายเป็นแบบ Multi-Level Cache ที่ประกอบด้วย Sibling Cache และ Parent Cache เท่านั้น ซึ่งการใช้งาน ICP ไม่มีความจำเป็นสำหรับเครือข่ายในรูปที่ 1 และ 2 ในกรณีที่มี Cache เพียงตัวเดียวหรือในกรณีที่ Cache มีการร้องขอ Document จาก Higher Lever Cache ที่เหมือนกัน สิ่งที่ ICP ไม่ต้องการก็เฉพาะ Overhead เท่านั้น

2.2 เทคโนโลยีเกี่ยวกับเว็บไซต์

ซีเอ็มเอสไทยแลนด์ (www, 2550)ได้สรุปไว้ว่า เว็บเพจแบบสแตติก (Static Web Page) คือ เว็บเพจทั่วไป โดยเนื้อหาของเว็บเพจเหล่านี้จะไม่มีการเปลี่ยนแปลงใด ๆ จนกว่าจะมีการนำมาทำการแก้ไขเนื้อหาของเว็บเพจนั้นใหม่จากหน้าเว็บที่เครื่องเซิร์ฟเวอร์ ตัวอย่างเช่น ถ้าต้องการให้ผู้เยี่ยมชมเว็บไซต์ทราบเวลาของเซิร์ฟเวอร์ในขณะที่ผู้ใช้กำลังเยี่ยมชม เว็บเพจแบบสแตติกจะไม่สามารถแสดงเวลาได้สำเร็จด้วยภาษา HTML ธรรมดา หรือถ้าต้องการให้ผู้ใช้ต้องการทราบเวลาที่แท้จริง ก็จำเป็นต้องอัพโหลดหน้าเว็บไซต์ทุกนาที หรือทุกวินาที เพื่อให้เวลาในหน้าเว็บดังกล่าวเดินไปเรื่อยๆ หรือให้เวลาดังกล่าว ตรงกับเวลาของเครื่อง เซิร์ฟเวอร์

เว็บเพจแบบไดนามิก (Dynamic Web Page) คือ เว็บเพจที่ทำให้เนื้อหาของเว็บไซต์เปลี่ยนไปได้เรื่อย ๆ ตามเวลา ตามวัน หรืออาจจะมีเนื้อหาเปลี่ยนไปตามที่ผู้ใช้ร้องขอ (ซีเอ็มเอสไทยแลนด์, www, 2550) สำหรับการเขียนเว็บเพจแบบนี้ จะมีความยุ่งยากมากกว่าการเขียนเว็บเพจแบบสแตติกบ้าง แต่การพัฒนาหน้าเว็บแบบไดนามิกนี้ จะช่วยลดงานในการเปลี่ยนแปลงเว็บ หรือกล่าวอีกนัยหนึ่งคือ ไม่จำเป็นต้องทำการเปลี่ยนแปลงเว็บไซต์บ่อย ๆ เพียงแค่เตรียมเนื้อหาที่จะ

แสดงในอนาคตไว้ แล้วทำการเขียนโปรแกรมให้หน้าเว็บนำเนื้อหาเหล่านั้นมาแสดงในเวลาที่กำหนด หรือตามคำร้องขอของผู้ใช้

2.2.1 ประเภทของ Dynamic Web Page

ประเภทของ Dynamic Web Page สามารถแบ่งตามหน้าที่การทำงานออกได้ 3 ประเภท ดังนี้

1) สคริปท์ฝั่งไคลเอนท์ คือ หน้าเว็บแบบไดนามิกที่มีคำสั่งซึ่งทำงานบนเครื่องไคลเอนท์ เมื่อผู้ใช้เข้าไปยังเว็บเพจนั้น ๆ เซิร์ฟเวอร์จะส่งเนื้อหาทั้งหมด รวมทั้งโค้ดทั้งหมดไปยังเครื่องไคลเอนท์ที่เรียกดูหน้าเว็บดังกล่าว แล้วโค้ดทั้งหมดจะถูกประมวลผลที่เครื่องไคลเอนท์

2) สคริปท์ฝั่งเซิร์ฟเวอร์ คือ หน้าเว็บแบบไดนามิกที่มีคำสั่งซึ่งทำงานบนเครื่องเซิร์ฟเวอร์เมื่อผู้ใช้เข้าไปยังเว็บเพจแบบนี้ เซิร์ฟเวอร์จะทำอ่านไฟล์ดังกล่าวก่อน แล้วทำการแปลโค้ดที่ได้เขียนไว้ในหน้าเว็บนั้นทั้งหมดเพื่อให้ได้ผลลัพธ์ออกมา หลังจากนั้นจึงส่งผลลัพธ์ซึ่งเป็น HTML เพียงอย่างเดียวไปให้เครื่องผู้ใช้

3) เป็นทั้งสคริปท์ฝั่งเซิร์ฟเวอร์ และสคริปท์ฝั่งไคลเอนท์ คือมีลักษณะเหมือนกับแบบที่ 1 และแบบที่ 2 รวมกัน คือมีโค้ดบางส่วนถูกแปลคำสั่งที่เครื่องเซิร์ฟเวอร์ก่อน และโค้ดบางส่วนที่จะต้องถูกแปลโดยเครื่องไคลเอนท์ (siamdev.net, www, 2548)

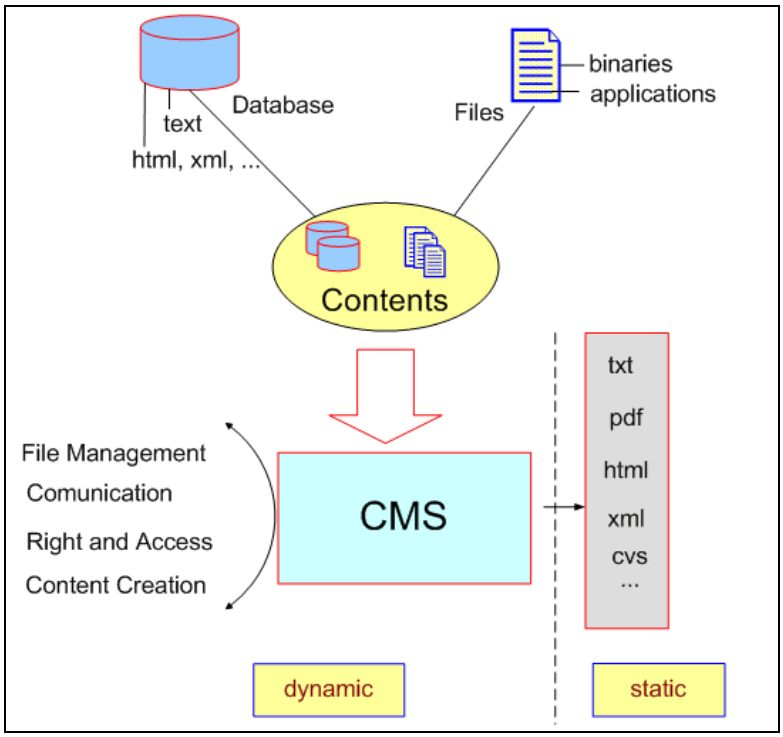
2.2.2 เทคโนโลยีของเว็บไซต์ (Web Site Technology)

ในปัจจุบันเทคโนโลยีของเว็บไซต์ได้ก้าวล้ำไปมาก หากจะเปรียบเทียบกับยุคการพัฒนาเว็บเพจในยุคแรก ๆ การศึกษาการเขียนโปรแกรมบนเว็บจะเป็นแบบ 1-2-3-4 ก็คือผู้เรียนต้องศึกษาเป็นขั้น ๆ กว่าจะพัฒนาเว็บไซต์ได้ ต้องเรียนรู้การใช้เครื่องมือหลายตัวด้วยกัน และใช้เวลายาวนาน แต่ในปัจจุบัน ผู้ที่จะศึกษาหรือเขียนเว็บ สามารถทำเว็บได้ด้วยเวลาอันรวดเร็ว เพราะมีเครื่องมือสำเร็จรูปให้เลือกใช้งานมากมาย และมีซอร์สโค้ดให้ดาวน์โหลดมากมายผ่านทางเครือข่ายอินเทอร์เน็ต ซึ่งเว็บที่พัฒนาขึ้นมาเพื่อการเรียนการสอน ก็เป็นอีกแบบหนึ่งที่มีให้ผู้ใช้สามารถดาวน์โหลดมาใช้งานได้โดยที่ผู้ใช้ไม่จำเป็นต้องมีความรู้ในการเขียนเว็บดังกล่าว

สำหรับเทคโนโลยีในการพัฒนาเว็บนั้น เราสามารถแบ่งออกได้เป็น 3 ยุคด้วยกัน

ยุคที่ 1 Static Web เป็นการเขียนเว็บไซต์แบบธรรมดาใช้เบราว์เซอร์เรียกเว็บเพจที่สร้างด้วยภาษา HTML อย่างเดียว หรืออาจจะมีสคริปท์ทางฝั่งไคลเอนท์ (Client-side Script) อย่าง JavaScript, VBScript หรือใช้ภาษา Java Applet ซึ่ง Static web ส่วนมากนิยมในกลุ่มของนักเรียนนักศึกษา หรือผู้เริ่มศึกษาการสร้างเว็บ อาจใช้ในการสร้างโฮมเพจส่วนตัว หรือเว็บไซต์เสนอเรื่องราวต่าง ๆ เช่น บทความวิชาการ เป็นต้น เมื่อทำเสร็จและก็อัปโหลดข้อมูลไปยังเว็บโฮสติ้งที่ให้บริการพื้นที่เก็บเว็บฟรี

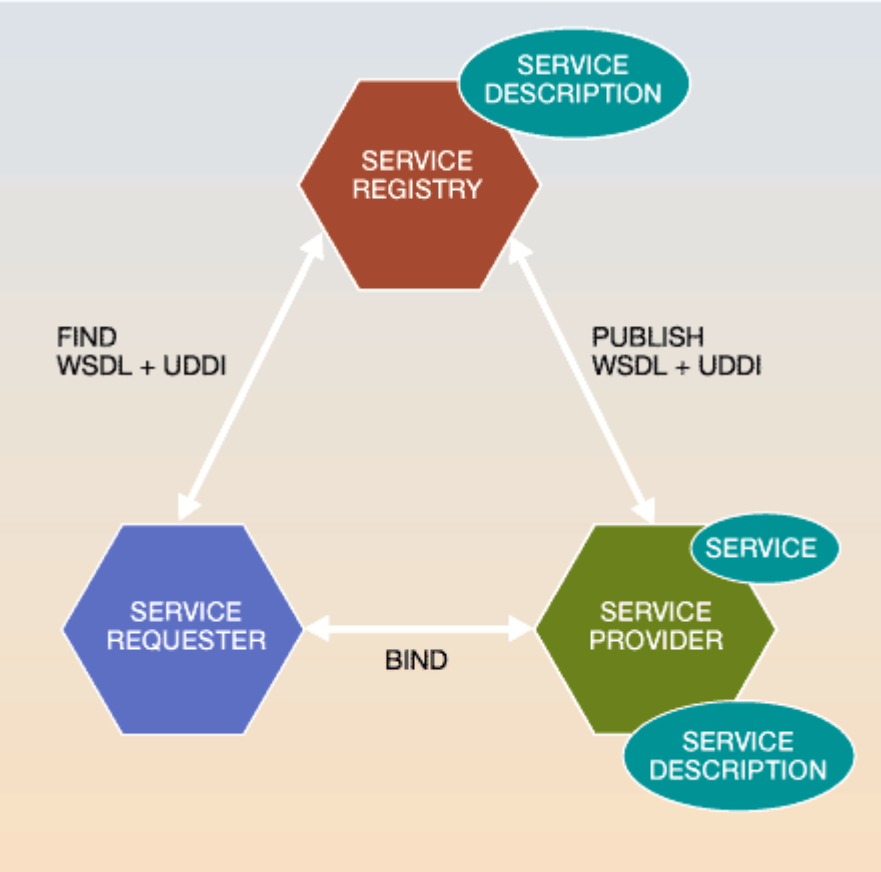
ยุคที่ 2 Dynamic Web พัฒนาต่อมาจากยุคที่ 1 มีการใช้สคริปท์ทางฝั่งเซิร์ฟเวอร์ (Server-side Script) มาช่วยในการเพิ่มความสามารถของเอกสาร HTML ในการติดต่อกับองค์ประกอบอื่นๆ ของทางฝั่งเซิร์ฟเวอร์ เช่น ฐานข้อมูล หรืออาศัยความสามารถในการประมวลผลของเว็บเซิร์ฟเวอร์เพื่อทำงานบางอย่าง เช่น Search, Webboard, Guestbook, Chat room, Webmail เป็นต้น โดยที่เทคโนโลยีในการพัฒนาเว็บแบบ Dynamic ในช่วงแรกคือ CGI (Common Gateway Interface) ภาษาที่ใช้เขียนสคริปท์นี้เช่น ภาษา C, Perl เป็นต้น ต่อมาได้มีการพัฒนาเทคโนโลยีที่มีการทำงานคล้าย ๆ CGI เพื่อทำงานทางฝั่งเซิร์ฟเวอร์หลายภาษา เช่น ASP (Microsoft), PHP (Open Source) และ JSP (Sun Microsystem) เป็นต้น



รูปที่ 2.8 แสดงการเปรียบเทียบระหว่าง Static กับ Dynamic Web
(ซีเอ็มเอสไทยแลนด์, www, 2550)

ยุคที่ 3 เว็บเซอร์วิส (Web Service) เป็นรูปแบบการบริการยุคใหม่ในวงการเว็บ ตัวอย่าง เว็บเซอร์วิสที่เห็นได้ชัด เช่น Microsoft Passport ที่ให้บริการตรวจสอบความเป็นตัวตนจริง (Authentication) ผ่านเว็บ ภาษาที่ใช้เป็นแกนกลางในการพัฒนาเว็บเซอร์วิสคือ XML (Extensive Markup Language) โดย Gottschalk, Graham, Kreger และ Snell (2002) ได้นิยามความหมายอย่าง

เป็นทางการของเว็บเซอร์วิสว่า เว็บเซอร์วิส คือ เว็บแอพพลิเคชันยุคใหม่ที่ประกอบด้วยส่วนย่อย ๆ มีความสมบูรณ์ในตัวเอง สามารถติดตั้ง ค้นหา หรือทำงานได้ผ่านเว็บ เว็บเซอร์วิสสามารถทำอะไรก็ได้ตั้งแต่งานง่าย ๆ เช่นดึงข้อมูล จนถึงกระบวนการทางธุรกิจที่ซับซ้อน เมื่อเว็บเซอร์วิสตัวใดตัวหนึ่งเริ่มทำงานเว็บเซอร์วิสตัวอื่นก็สามารถรับรู้และเริ่มทำงานได้อีกด้วย (ซีเอ็มเอสไทยแลนด์, www, 2550) ซึ่งสามารถแสดงถึงความสัมพันธ์ที่เกี่ยวข้องกับเว็บเซอร์วิสได้ดังรูป 2.9



รูปที่ 2.9 แสดง Web services actors, objects และ operations (researchweb, www, 2002)

ระบบ LMS จัดอยู่ในยุคของ Dynamic Web Site ในปัจจุบัน บางตัวก็เริ่มก้าวเข้าสู่ยุคที่ 3 ซึ่งเป็นเว็บเซอร์วิส โดยมีบริการต่าง ๆ แบบอัตโนมัติในตัวเอง สำหรับตัวอย่างเครื่องมือในการสร้างและตกแต่งรูปแบบบนเว็บไซต์มีดังนี้

1) เครื่องมือในการสร้างเว็บไซต์

- HTML เป็นภาษาแรกใช้ในการพัฒนาเว็บไซต์ จะใช้ Editor ตัวใดเขียนก็ได้ จากนั้นทำการบันทึกงานเป็นนามสกุล .htm หรือ .html

- EditPlus เป็นเครื่องมือช่วยในการเขียนสคริปท์ต่าง ๆ อาทิ HTML, PHP, Java และ XML เป็นต้น ซึ่งนิยมในกลุ่มของนักพัฒนาโปรแกรมมาก โดยโปรแกรมถูกพัฒนาจากบริษัท ES-Computing

- Dreamweaver เครื่องมือช่วยสร้างเว็บไซต์ ลักษณะ WYSIWYG โดยโปรแกรมถูกพัฒนาจากบริษัท Macromedia

- FrontPage เครื่องมือช่วยสร้างเว็บไซต์ ลักษณะ WYSIWYG โดยโปรแกรมถูกพัฒนาจากบริษัท Microsoft

- Namo Web Editor เครื่องมือช่วยสร้างเว็บไซต์ โดยโปรแกรมถูกพัฒนาจากบริษัท SJ NAMO

- PostNuke เครื่องมือบริหารและจัดการสร้างเว็บไซต์แบบสำเร็จรูป (Content Management System) โดยโปรแกรมถูกพัฒนาจากบริษัท PostNuke Software Foundation

- Moodle เครื่องมือบริหารและจัดการสร้างเว็บไซต์แบบเรียนสำเร็จรูป (Learning Management System)

2) เครื่องมือในการเพิ่มลูกเล่นบนเว็บไซต์

- JavaScript เครื่องมือในการเสริมแต่งเว็บ เช่น ข้อความวิ่ง หรือใส่เหตุการณ์ต่างๆ บนหน้าเว็บ

- CSS (Cascading Style Sheets) นิยมนำมาใช้ควบคุมการแสดงผลของข้อความในเว็บไซต์

3) เครื่องมือในการสร้างรูปภาพกราฟิกและภาพเคลื่อนไหว สำหรับใช้บนเว็บ

- Photoshop & Image Ready โฟโต้ชอบใช้ในการตกแต่งภาพ ส่วนอิมเมจรีดดี้ ใช้ในการลดขนาดภาพแล้วทำภาพเคลื่อนไหว โดยโปรแกรมถูกพัฒนาจากบริษัท Adobe

- Firework ใช้ในการสร้างภาพและตกแต่งภาพ สำหรับเว็บโดยเฉพาะ โปรแกรมถูกพัฒนาขึ้นจากบริษัท Macromedia แต่ในปัจจุบันได้เปลี่ยนมาเป็นบริษัท Adobe

- Flash ใช้ในการสร้างภาพเคลื่อนไหวสำหรับเว็บ โดยโปรแกรมถูกพัฒนาขึ้นจากบริษัท Macromedia แต่ในปัจจุบันได้เปลี่ยนมาเป็นบริษัท Adobe

- SWiSH เป็นโปรแกรมที่มีขนาดเล็กและทำงานคล้าย Flash สามารถสร้างภาพเคลื่อนไหวได้อย่างง่าย โดยโปรแกรมถูกพัฒนาจากบริษัท SWiSHzone

- GIF Animator สร้างภาพเคลื่อนไหวแบบง่าย ๆ โดยโปรแกรมถูกพัฒนาจากบริษัท Creabit Development

4) เครื่องมือในการพัฒนาแอพพลิเคชันบนเว็บ

- PHP (PHP Hypertext Preprocessor) เป็นฟรีแวร์ ที่ได้รับความนิยมอย่างมากในปัจจุบัน

- ASP (Active Server Page) เป็นผลิตภัณฑ์ของไมโครซอฟท์ ปัจจุบัน ASP จะถูกใส่เข้าไปในโปรแกรมที่เป็นเว็บเซิร์ฟเวอร์ของบริษัทไมโครซอร์ฟ ดังนั้นไม่จำเป็นต้องทำการติดตั้งโปรแกรม ASP อีก

- ASP.NET คือแนวความคิดหนึ่งที่ไมโครซอฟท์ได้พัฒนาขึ้น โดย .NET ไม่ได้เกี่ยวข้องกับโดเมนเนมของเว็บใดๆทั้งสิ้น แต่ .NET หมายถึง การนำเอาอุปกรณ์ทุกอย่างบนโลกมาเชื่อมโยงต่อกันเหมือนตาข่าย (Net = ตาข่าย)

- JSP (Java Server Page) เป็นสคริปท์อีกภาษาหนึ่ง ซึ่งเป็นทางเลือกที่น่าสนใจสำหรับการพัฒนาเว็บแอพพลิเคชัน จุดเด่นที่สำคัญของ JSP อยู่ที่การใช้ภาษา Java ซึ่งเป็นภาษาเชิงวัตถุที่ช่วยให้สามารถพัฒนาแอปพลิเคชันขนาดใหญ่และซับซ้อน ได้อย่างสะดวกและรวดเร็ว

- Perl ได้ถูกนิยามในยุคแรก ๆ ของเว็บ โดย Doug Sheppard (www, 2000) ได้ให้ความหมายว่า Perl ย่อมาจาก Practical Extraction and Report Language ถูกพัฒนาขึ้นในปี 1980 โดยนาย Larry Wall ซึ่งเป็นภาษาที่ใช้ทำ CGI ภาษาหนึ่ง มีรากฐานการพัฒนามาจากภาษา C ดังนั้นผู้ที่เคยศึกษาภาษา C มาบ้างแล้ว สามารถศึกษาภาษา Perl ได้โดยไม่ยาก

- Python เป็นภาษาระดับสูงภาษาหนึ่ง ที่มีความสามารถสูงถูกสร้างขึ้น โดย Guido van Rossum (www, 1989) ซึ่งภาษานี้พัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows NT/2000/XP/2003, FreeBSD หรือแม้แต่ระบบ MAC OS X, Palm, Nokia Mobile อีกอย่างหนึ่งเป็นภาษาลักษณะ Open Source ที่แจกจ่ายให้ใช้งานฟรี

5) โปรแกรมฐานข้อมูลที่นิยมใช้บนเว็บ

- MySQL เป็นฐานข้อมูลแบบโอเพนซอร์สที่ได้รับความนิยมในการใช้งานสูงสุดโปรแกรมหนึ่งบนเครื่องให้บริการ มีความสามารถในการจัดการกับฐานข้อมูลด้วยภาษา SQL (Structures Query Language) อย่างมีประสิทธิภาพ มีความรวดเร็วในการทำงาน รองรับการทำงานจากผู้ใช้หลาย ๆ คนและหลาย ๆ งานได้ในขณะเดียวกัน

- PostgreSQL คือ ระบบฐานข้อมูลที่นำเอาคำสั่งต่าง ๆ ของภาษา SQL มารวมกันในการโปรแกรม ทำให้คำสั่งเหล่านั้นสามารถทำงานได้อย่างมีประสิทธิภาพ และตรงตามความต้องการของผู้ใช้

- Microsoft Access เป็นอีกโปรแกรมหนึ่งของชุด Microsoft Office โปรแกรม Access นี้เป็นโปรแกมฐานข้อมูล เหมาะสำหรับเก็บข้อมูล และนำมาประมวลผล หรือนำเสนอในรูปแบบรายงาน สามารถนำฐานข้อมูล Access เพื่อเป็นฐานข้อมูลของโปรแกรมอื่น ๆ ได้ทั่วไป ไม่ว่าจะใช้พัฒนาร่วมกับ Visual Basic, C++ ก็ได้

2.3 ระบบฐานข้อมูล MySQL

MySQL (มายเอสคิวแอล) สร้างขึ้นโดยชาวสวีเดน ชื่อ David Axmark และ Allan Larsson และชาวฟินแลนด์ชื่อ Michael "Monty" Widenius (MySQL.com, www, 2008)โดยระบบฐานข้อมูล MySQL เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน สามารถแบ่งออกได้ 3 แบบ คือเวอร์ชันที่เป็นของฟรี (คอมมิวนิตี้ เวอร์ชัน) เวอร์ชันการค้า (คอมเมอร์เชียลเวอร์ชัน) และ เวอร์ชันที่สนับสนุนกับผลิตภัณฑ์ SAP (MAX DB) โดยมีความแตกต่างคือ เวอร์ชันคอมมิวนิตี้นั้นสามารถนำไปใช้งานได้ฟรีแต่ขาดการสนับสนุนหรือการช่วยเหลือเมื่อมีปัญหาเกิดขึ้น ส่วนเวอร์ชันที่เป็นคอมเมอร์เชียลนั้นให้บริการด้านความสนับสนุนเมื่อมีปัญหา (ซื้อบริการ) และเวอร์ชันสุดท้าย สนับสนุนและใช้งานกับโปรแกรม SAP ในระบบฐานข้อมูล MySQL เวอร์ชันล่าสุดคือ เวอร์ชัน 5.0 มีความสามารถหลายอย่างที่สำคัญสำหรับระบบฐานข้อมูลขนาดใหญ่ (EnterPrise Feature) เช่น Store Procedure, trigger, view และ schema เป็นต้น ปัจจุบันอยู่ที่เวอร์ชัน 6.0.4 alpha (MySQL.com, www, 2008) ซึ่งได้มีการปรับเพิ่มประสิทธิภาพในส่วนของตารางและ Index ขึ้นมาอีก

MySQL เป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ เช่น มีเดียวิกิ และ phpBB และนิยมใช้งานร่วมกับภาษาโปรแกรม PHP จะเห็นได้จากคู่มือคอมพิวเตอร์ต่าง ๆ ที่จะสอนการใช้งาน MySQL และ PHP ควบคู่กันไป นอกจากนี้ หลายโปรแกรมภาษาที่สามารถทำงานร่วมกับฐานข้อมูล MySQL ซึ่งรวมถึง ภาษาซี ซีพลัสพลัส ปาสคาล ซีชาร์ป ภาษาจาวา เพิร์ล PHP ไพทอน รูบี และภาษาอื่น ใช้งานผ่าน API สำหรับโปรแกรมที่ติดต่อผ่าน ODBC เช่น เอเอสพี สามารถเรียกใช้ MySQL ผ่านทาง MyODBC

ส่วนการจัดการฐานข้อมูล MySQL สามารถใช้โปรแกรมแบบ Command-line (โดยใช้คำสั่ง : mysql และ mysqladmin) หรือจะดาวน์โหลดโปรแกรมจัดการฐานข้อมูลแบบกราฟิค (Graphic User Interface : GUI) จากเว็บไซต์ของ MySQL ซึ่งคือโปรแกรม MySQL Administrator และ MySQL Query Browser

ส่วนติดต่อ (GUI) เพื่อเชื่อมต่อกับภาษาในการพัฒนาอื่นๆ เพื่อให้เข้าถึงฟังก์ชันการทำงานกับฐานข้อมูล MySQL ได้เช่น ODBC (Open Database Connector) อันเป็นมาตรฐานกลางที่กำหนดมาเพื่อให้ใช้เป็นสะพานในการเชื่อมต่อกับโปรแกรมหรือระบบอื่น ๆ เช่น MyODBC อันเป็นไดรเวอร์เพื่อใช้สำหรับการเชื่อมต่อในระบบปฏิบัติการวินโดว์, JDBC คลาสส่วนเชื่อมต่อสำหรับ Java เพื่อใช้ในการติดต่อกับ MySQL และมี API (Application Programming Interface) ต่าง ๆ มีให้เลือกใช้มากมายในการที่เข้าถึง MySQL โดยไม่ขึ้นอยู่กับภาษาในการพัฒนาภาษาใดภาษาหนึ่งและยังมีโปรแกรมเว็บแอพพลิเคชันอีกตัวหนึ่งชื่อว่า phpMyAdmin เป็นโปรแกรมบริหารการจัดการฐานข้อมูลในรูปแบบกราฟิค ซึ่งพัฒนาขึ้นโดย Olivier Mueller และ Marc Delisle (sourceforge.net, www, 2001) ซึ่งเป็นทีมงานของ The phpMyAdmin Project และได้รับความนิยมในการใช้งานกันอย่างแพร่หลาย

2.4 โปรแกรมภาษา PHP

ภาษาพีเอชพี ในชื่อภาษาอังกฤษว่า PHP ซึ่งใช้เป็นคำย่อแบบกล่าวซ้ำ จากคำว่า PHP Hypertext Preprocessor หรือชื่อเดิม Personal Home Page หรือในอีกความหมาย PHP คือภาษาคอมพิวเตอร์ที่มีลักษณะการทำงานฝั่งของเซิร์ฟเวอร์แบบเซิร์ฟเวอร์-ไซต์สคริปท์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษา PHP ใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจาก ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล โดย ภาษา PHP ง่ายต่อการเรียนรู้ และเหมาะสำหรับนักพัฒนาเว็บไซต์เพื่อใช้ในการเขียนสคริปท์สำหรับเว็บเพจที่ต้องการการตอบโต้สำหรับผู้ใช้กับเซิร์ฟเวอร์ (กิตติ ภักดีวัฒนกุล, 2547)

การแสดงผลของ PHP จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่ PHP แตกต่างจากภาษาในลักษณะไคลเอนต์-ไซต์สคริปท์ เช่น ภาษาจาวาสคริปท์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้ PHP ยังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถการประมวลผลหลักของ PHP ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะ CGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (Command Line Scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปท์ PHP ทำงานผ่าน PHP พาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราเซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ในยูนิกซ์หรือลีนุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปท์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

การแสดงผลของ PHP ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผล HTML แต่ยังสามารถสร้าง XHTML หรือ XML ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่าง ๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แฟลช (โดยใช้ Libswf และ Ming) PHP มีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก POSIX Extended หรือ รูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลงและเข้าสู่เอกสาร XML เรารองรับมาตราฐาน SAX และ DOM สามารถใช้รูปแบบ XSLT ของเราเพื่อแปลงเอกสาร XML เมื่อใช้ PHP ในการทำอีคอมเมิร์ซ สามารถทำงานร่วมกับโปรแกรมอื่น เช่น Cybercash payment, CyberMUT, VeriSign Payflow Pro และ CCVS functions เป็นต้น เพื่อใช้ในการสร้างโปรแกรมทำธุรกรรมทางการเงิน

คำสั่งของ PHP สามารถสร้างผ่านทางโปรแกรมแก้ไขข้อความทั่วไป เช่น notepad หรือ vi ซึ่งทำให้การทำงาน PHP สามารถทำงานได้ในระบบปฏิบัติการหลักเกือบทั้งหมด และสามารถประมวลผลบนเว็บเซิร์ฟเวอร์ต่าง ๆ เช่น Apache, Microsoft Internet Information Server (IIS), Personal Web Server, Netscape, iPlanet Servers, Oreilly Website Pro Servers, Caudium, Xitami, OmniHTTPd เป็นต้น สำหรับส่วนหลักของ PHP ยังมีโมดูลในการรองรับ CGI มาตรฐาน ซึ่ง PHP สามารถทำงานเป็นตัวประมวลผล CGI ได้ ทั้งยังสามารถทำงานร่วมกับระบบปฏิบัติการและเว็บเซิร์ฟเวอร์ได้ทุกระบบ นอกจากนี้ยังสามารถใช้สร้างโปรแกรมโครงสร้าง สร้างโปรแกรมเชิงวัตถุ (OOP) หรือสร้างโปรแกรมที่รวมทั้งสองอย่างเข้าด้วยกัน แม้ว่าความสามารถของคำสั่ง OOP มาตรฐานในเวอร์ชันนี้ยังไม่สมบูรณ์ แต่ตัวไลบรารีทั้งหลายของโปรแกรม และตัวโปรแกรมประยุกต์ (รวมถึง PEAR library) ได้ถูกเขียนขึ้นโดยใช้รูปแบบการเขียนแบบ OOP เท่านั้น

PHP สามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับได้แก่ Oracle, dBase, PostgreSQL, IBM DB2, MySQL และ Informix ODBC เป็นต้น โครงสร้างของฐานข้อมูลแบบ DBX ซึ่งทำให้ PHP ใช้กับฐานข้อมูลที่รองรับรูปแบบนี้ และ PHP ยังรองรับ ODBC (Open Database Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อฐานข้อมูลที่ใช้กันแพร่หลายอีกด้วย ซึ่งสามารถเชื่อมต่อกับฐานข้อมูลต่าง ๆ ที่รองรับมาตรฐานโลกนี้ได้

PHP ยังสามารถรองรับการสื่อสารกับการบริการในโปรโตคอลต่างๆ เช่น LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (บนวินโดวส์) และอื่นๆ อีกมากมาย โดยสามารถเปิด Socket บนเครื่อข่ายโดยตรงและตอบโต้โดยใช้โปรโตคอลใด ๆ ก็ได้ PHP มีการรองรับสำหรับการแลกเปลี่ยนข้อมูลแบบ WDDX Complex กับ Web Programing อื่นๆ ทั่วไปได้ กล่าวถึงในส่วน Interconnection โดย PHP มีการรองรับสำหรับ Java Objects ให้เปลี่ยนเป็น PHP Object แล้วใช้งาน ยังสามารถใช้รูปแบบ CORBA เพื่อเข้าสู่ Remote Object ได้เช่นกัน

เว็บเพจที่ใช้ PHP เป็นโครงสร้างหลัก โดยเว็บเพจส่วนมากที่ใช้ PHP เป็นโครงสร้างหลักนั้น จะมีลักษณะรูปแบบเว็บเพจเป็นแบบ Dynamic ซึ่งสามารถโต้ตอบกับผู้ใช้ได้ดี จึงทำให้ได้รับความนิยมกันอย่างแพร่หลาย อีกทั้งภาษา PHP ยังเป็น Open Source อีกด้วย จึงทำให้มีผู้นำไปพัฒนากันอย่างแพร่หลาย เช่น ระบบการจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ก็เป็นอีกระบบหนึ่งที่มีการนำภาษา PHP ไปใช้งานด้วยเช่นกัน ตัวอย่างระบบที่ใช้ภาษา PHP เช่น Joomla, Moodle, PHPBB และ Mambo เป็นต้น โดยเฉพาะ Moodle เป็นระบบที่ได้รับความนิยมเป็นอย่างมาก ซึ่งจะได้อธิบายรายละเอียดของ Moodle ในหัวข้อถัดไป

2.5 Moodle (มูเดิ้ล)

Moodle คือ ชุดของ เซิร์ฟเวอร์-ไซต์สคริปท์ สำหรับสถาบันการศึกษา หรือ ครู อาจารย์ เพื่อเตรียมแหล่งข้อมูล กิจกรรม และเผยแพร่แบบเรียนออนไลน์ผ่านอินเทอร์เน็ต หรืออินทราเน็ต สามารถนำไปใช้ได้ทั้งองค์กรในระดับ มหาวิทยาลัย โรงเรียน สถาบัน หรือครูสอนพิเศษ ผู้พัฒนาโปรแกรมคือ Martin Dougiamas โปรแกรมชุดนี้เป็น Open Source ภายใต้ข้อตกลงของ gnu.org (General Public License) สามารถดาวน์โหลดและใช้งานได้ฟรี สำหรับผู้ดูแลระบบ (Admin) ที่จะนำโปรแกรมไปติดตั้ง ต้องมีเว็บเซิร์ฟเวอร์ที่บริการ PHP และ MySQL ด้วย เพราะโปรแกรมดังกล่าวถูกพัฒนาขึ้นมาจากภาษา PHP และระบบฐานข้อมูลที่ใช้ได้พัฒนามาจาก MySQL (Moodle.org, www, 2005)

Moodle เป็นโอเพนซอร์สที่ได้รับการยอมรับเป็นอย่างมาก ซึ่งเห็นได้จากจำนวนเว็บไซต์ 13544 เว็บบีบีHห์ไซต์จาก 158 ประเทศทั่วโลก (moodle.org, www, 2004) สามารถเป็นได้ทั้ง CMS (Course Management System) และ LMS (Learning Management System) ที่สามารถช่วยรวบรวมวิชาเป็นหมวดหมู่ เผยแพร่เนื้อหาของผู้สอน พร้อมบริการให้นักเรียนเข้ามาศึกษา และบันทึกกิจกรรมของนักเรียน สามารถสร้างแหล่งข้อมูลใหม่หรือเผยแพร่เอกสารที่ทำไว้ เช่น Microsoft Office, Web Page, PDF หรือ Image เป็นต้น มีเอกสารที่รวบรวมไว้สามารถส่งเข้าไปเผยแพร่ได้โดยง่าย มีระบบติดต่อสื่อสารระหว่างนักเรียน เพื่อนร่วมชั้น และผู้สอน เช่น chat หรือ webboard เป็นต้น มีระบบแบบทดสอบ รับการบ้าน และกิจกรรม ที่รองรับระบบให้คะแนนที่หลากหลาย ให้ส่งงาน ให้ทำแบบฝึกหัด ตรวจให้คะแนนแล้ว export ไป excel สำรองข้อมูลเป็น .zip แฟ้มเดียว ในอนาคตสามารถนำไปกู้คืนลงไปในเครื่องใดก็ได้ สำหรับแหล่งข้อมูลหรือกิจกรรม (Resource and Activities) ที่ถูกพัฒนาและนำมาใช้งานนั้น สามารถแบ่งออกได้ดังนี้

- SCORM (แหล่งข้อมูล ที่รวม Content จากภายนอก ที่เป็นมาตรฐาน)
- Wiki (สารานุกรม ที่ยอมให้ผู้เรียนเข้ามาแก้ไข)

- อภิธานศัพท์ (Glossary : รวมคำศัพท์ จัดหมวดหมู่ สามารถสืบค้นได้)
- ห้องสนทนา (Chat : ห้องที่สามารถนัดเวลาสนทนาระหว่างครู และนักเรียน)
- กระดานเสวนา (Forum : กระดานให้ครู และนักเรียนเข้ามาฝากความคิดเห็น)
- การบ้าน (Assignment : ที่นักเรียนพิมพ์งานแล้วนำมา upload ส่งครู)
- ห้องปฏิบัติการ (Workshop : ที่นักเรียนทำงานแล้วส่ง ซึ่งประเมินได้หลายแบบ)
- ป้ายประกาศ (Label : แสดงข้อความ เพื่อประกาศให้ทราบ)
- แบบทดสอบ (Quiz : สร้างคลังข้อสอบแล้วเลือกมาให้ทำบางส่วน ระบบสามารถทำงานได้อัตโนมัติ)
- โพลล์ (Poll : แสดงความคิดเห็นตามตัวเลือก)
- แหล่งข้อมูล (Resources : text, html, upload, weblink, webpage, program)
- กิจกรรมของผู้สอน (Teacher Activities) (elearning.pharmacy.psu.ac.th, www, 2550)

2.6 งานวิจัยที่เกี่ยวข้อง

Iyengar, Challenger and Dantzig (1999) ได้เสนอแนวความคิดในการออกแบบระบบที่ใช้ในการค้นคืนข้อมูลจากเว็บที่เขียนแบบ Dynamic โดยได้เสนอการค้นคืนข้อมูลจากระบบ Caching และได้ทำการออกแบบการทดลองระบบดังกล่าว พร้อมทั้งนำเสนอข้อมูล โดยระบบที่ได้ทำการทดสอบนี้สามารถเพิ่มประสิทธิภาพในการค้นคืนข้อมูลได้ดียิ่งขึ้น

Iyengar, Challenger, Dias and Dantzig (2000) นำเสนอแนวความคิดในการพัฒนาอัลกอริธึมใหม่ที่เรียกว่า Data Update Propagation (DUP) เพื่อตรวจสอบว่าหน้าเว็บของ Cache ใดเก่าแล้วจากข้อมูลใหม่โดย DUP จะพิจารณาว่าหน้าเว็บเพจของ Cache ใดมีการเปลี่ยนแปลง ถ้ากลุ่มของหน้าเว็บเพจของ Cache ถูกสร้างขึ้นจาก Table ที่อยู่ในฐานข้อมูลของ Cache นั้น จะต้องเหมือนกันกับข้อมูลในฐานข้อมูลจึงจะถือว่าข้อมูลหน้านั้นไม่มีการเปลี่ยนแปลง ซึ่งสามารถทำให้ระบบทำงานได้ดีขึ้น

Chang, Lin and Ting (2004) ได้เสนอวิธีการค้นคืนข้อมูลในระบบ Cache โดยได้ทำการทดลองกับ Tomcat Web เซิร์ฟเวอร์ และเว็บเพจที่เป็นแบบชนิด Dynamic โดยโปรแกรมภาษา JSP (Java Servlet Pages) และยังได้เสนอขั้นตอนการออกแบบระบบ และขั้นตอนการทำงานดังกล่าว โดยระบุว่าระบบที่ได้ทำการออกแบบนั้น สามารถช่วยปรับปรุงระบบให้มีประสิทธิภาพดีขึ้นได้ถึง 290 เปอร์เซ็นต์

จักรพันธ์ มาดีตระกูล และ สุขุมาล กิติสิน (2547) ได้เสนอแนวคิดใหม่ โดยการใช้งานของเว็บแคชในปัจจุบันเป็นแบบศูนย์รวม (Centralized Web Cache) ซึ่งใช้พร็อกซี่แคชเป็นศูนย์กลางใน

การให้บริการแก่ไคลเอนท์ในองค์กร ซึ่งมีข้อจำกัดคือเมื่อไคลเอนต์มีจำนวนเพิ่มมากขึ้น Proxy-cache จะต้องมีการปรับปรุงประสิทธิภาพเพื่อให้สามารถรองรับการขยายตัวได้ตามจำนวนของไคลเอนท์ โดยได้เปลี่ยนรูปแบบการให้บริการจากแบบรวมศูนย์มาเป็นแบบกระจาย (Decentralized Web Cache) ที่ใช้เครื่องไคลเอนท์ทำหน้าที่แคชกันเองในระบบ ทำให้สามารถแก้ข้อจำกัดดังกล่าวได้ อย่างไรก็ตามเว็บแคชแบบกระจายที่ได้มีการนำเสนอใช้วิธีการจัดวางข้อมูลโดยการนำ URL มาแฮชเพื่อระบุเครื่องไคลเอนท์ที่ต้องทำหน้าที่แคชข้อมูล ประกอบกับการใช้วิธีการแทนที่แบบ LRU (Least Recently Used) เพียงอย่างเดียวเท่านั้น โดยในงานวิจัยนี้ได้ทำการศึกษาเปรียบเทียบวิธีการแทนที่แบบต่าง ๆ และประกอบกับการจัดวางข้อมูลโดยการใช้อัลกอริทึม MD5 โดยในการจำลองการทำงานของเว็บแคชแบบกระจาย ได้ใช้ข้อมูลการใช้เว็บของมหาวิทยาลัยเกษตรศาสตร์ในการทดสอบ เพื่อเป็นการศึกษาความเป็นไปได้ในการนำเว็บแคชแบบกระจายมาใช้งานจริงในมหาวิทยาลัยที่มีจำนวนเครื่องไคลเอนท์เพิ่มขึ้นอย่างต่อเนื่อง ผลการทดลองพบว่าวิธีการแทนที่ที่เหมาะสมที่สุดคือแบบ LFU (Least Frequency Used) ด้วยการคัดเลือกข้อมูลที่มีขนาดใหญ่ที่สุดออกหากความถี่ที่น้อยที่สุดมีมากกว่า 1 ตัว จากการวัดประสิทธิภาพทั้งในแง่ของอัตราการพบข้อมูลและภาระงานการแทนที่ของไคลเอนท์ในระบบ

ศุภวัฒน์ แก้วมงคล และ มารอง ผดุงสิทธิ์ (2549) ได้นำเสนอแนวความคิดเกี่ยวกับไดนามิกเว็บ (DynamicWeb) ว่าสามารถส่งผลให้เว็บเซิร์ฟเวอร์ (Web Server) นั้นมีประสิทธิภาพลดลง เนื่องจากมีการประมวลผลสคริปท์ทางฝั่งเซิร์ฟเวอร์ (Server-Side Script) เทคนิคทั่วไปที่ใช้ในการบรรเทาปัญหาดังกล่าวคือการแคชหน้าไดนามิกเว็บ (Dynamic Web Caching) ซึ่งส่งผลให้การร้องขอหน้าเว็บเดิมในคราวถัดไปไม่จำเป็นต้องมีการประมวลผลสคริปท์ซ้ำอีก เนื่องด้วยแคชนั้นมักจะถูกเก็บไว้ในหน่วยความจำคอมพิวเตอร์ไว้เป็นระยะเวลานาน จึงทำให้มีโอกาสเกิดความผิดพร่อง (Fault) ในหน่วยความจำและอาจส่งผลให้มีการสร้างหน้าเว็บที่ผิดพลาดสำหรับเว็บแอพพลิเคชัน (Web Application) บางประเภทที่ต้องการความน่าเชื่อสูง (Reliability) จำเป็นจะต้องมีการหลีกเลี่ยงไม่ให้เกิดความผิดพร่องในระหว่างการสร้างหน้าเว็บ หรือทำให้สามารถทนทานต่อความผิดพร่อง (Fault-tolerance) ดังกล่าวได้ เพื่อให้ไดนามิกเว็บสามารถทนทานต่อความผิดพร่องได้ ผู้วิจัยจึงได้นำเสนอเทคนิคซึ่งใช้หลักการเพิ่มส่วนที่ซ้ำซ้อนของข้อมูลข่าวสาร (Information Redundancy) เข้ากับโปรแกรมประยุกต์ที่ใช้ในการแคชไดนามิกเว็บซึ่งมีอยู่แล้ว โดยผลที่ได้ปรากฏว่าสามารถทนทานต่อความผิดพร่องได้ โดยไม่ส่งผลกระทบต่อประสิทธิภาพความเร็วในการสร้างหน้าเว็บมากนัก

อภิเศรษฐ์ ทนันชัย และ สมชาย นำประเสริฐชัย (2545) ได้ทำการศึกษาเรื่องของสควิดหรือพร็อกซี่แคชเซิร์ฟเวอร์ที่มีประสิทธิภาพสูงที่ทำงานบนระบบปฏิบัติการยูนิกซ์ ข้อมูลแต่ละอี

อบเจกต์ที่สามารถแคชได้จะถูกเก็บลงเป็นแฟ้มข้อมูลของระบบปฏิบัติการ ระบบแฟ้มข้อมูลทั่วไปถูกออกแบบมาให้ทำการเก็บและค้นหาชื่อแฟ้มข้อมูลด้วยวิธีการเชิงเส้น ทำให้เวลาที่ใช้เป็นสัดส่วนโดยตรงกับจำนวนแฟ้มข้อมูลที่มีปัญหาคอขวด จึงเกิดขึ้นในส่วนของการเข้าถึงแฟ้มข้อมูลของสควิดเนื่องจากมีแฟ้มข้อมูลเป็นจำนวนมาก ปัญหานี้ทำให้การเข้าใช้งานแฟ้มข้อมูลส่วนใหญ่ของสควิดใช้เวลาค่อนข้างมาก บทความนี้จะได้กล่าวถึงแนวทางปรับปรุง การออกแบบ และพัฒนาวิธีเก็บอ็อบเจกต์ใหม่สำหรับสควิดเพื่อลดปัญหาข้างต้น ผลลัพธ์ที่ได้แสดงให้เห็นถึงประสิทธิภาพบางส่วนที่ดีขึ้นเมื่อเทียบกับวิธีการในระบบแบบเดิม

ศุภชานันท์ วนภู (2550) ได้ทำการศึกษาวิเคราะห์ระบบฐานข้อมูล และฟังก์ชันการทำงานต่าง ๆ ที่เกี่ยวข้องกับข้อมูลการเข้าใช้บทเรียนของผู้เรียน ของระบบ Moodle โดยวิเคราะห์จากข้อมูลจริงที่ได้จากระบบ SUT-LMS โดยได้ทำการศึกษาและวิจัยเกี่ยวกับพฤติกรรมการเข้าใช้บทเรียน e-Learning จากผลการวิจัยพบว่า สามารถนำเอาข้อมูลที่ได้บันทึกการเข้าใช้บทเรียนที่จัดเก็บอยู่ในฐานข้อมูลขนาดใหญ่ของระบบจัดการการเรียนการสอนผ่านเครื่อข่ายอินเทอร์เน็ตนั้นมาผ่านกระบวนการวิเคราะห์ สังเคราะห์ เพื่อให้เกิดเป็นสารสนเทศที่เป็นประโยชน์ต่อไปได้จึงเป็นที่มาของการออกแบบและพัฒนาโปรแกรม Edi-Mo

สมพันธุ์ ชาญศิลป์ (2551) ได้นำเสนอผลงานการวิจัยและพัฒนาระบบลินุกซ์เซิร์ฟเวอร์พร้อมใช้ ในงานโครงการขับเคลื่อนอุตสาหกรรมซอฟต์แวร์ด้วยโอเพนซอร์ส 2551 โดยได้ทำการนำเสนอ SUT Instant Server for Developer Plus Moodle ซึ่งเป็นระบบลินุกซ์เซิร์ฟเวอร์ที่ได้พัฒนาขึ้น โดยนำ Moodle 3 เวอร์ชัน คือ Moodle1.6.6, Moodle1.8.4 และ Moodle2.0 มาพัฒนาร่วมด้วย

จากที่ได้กล่าวมาทั้งหมดแล้วนั้น จะเห็นได้ว่าเทคโนโลยีที่ใช้ในการค้นคืนข้อมูลมีอยู่หลายรูปแบบ เช่น การใช้เทคโนโลยี AJAX ในการพัฒนาการเขียนเว็บเพจเพื่อช่วยลดภาระการทำงานของเซิร์ฟเวอร์ทำให้มีการตอบสนองต่อผู้ใช้ได้ดีขึ้น หรือระบบไดนามิกเว็บแคชที่ทำงานอยู่ในระบบ Proxy Server เป็นต้น ระบบเหล่านี้มีความสามารถในการทำงานที่แตกต่างกันออกไปแล้วแต่ความเหมาะสมในการนำไปใช้งาน ซึ่งเว็บไซต์ที่พัฒนาขึ้นในปัจจุบันนี้ก็มีอยู่หลายรูปแบบสามารถแบ่งออกเป็นเว็บเพจแบบสแตติกมีลักษณะเป็นเว็บเพจแบบทั่วไป โดยเนื้อหาของเว็บจะไม่มีการเปลี่ยนแปลงจนกว่าผู้ดูแลจะนำมาแก้ไข ส่วนเว็บเพจอีกแบบคือเว็บเพจแบบไดนามิก ซึ่งมีลักษณะเป็นเว็บเพจแบบที่เนื้อหาของเว็บจะมีการเปลี่ยนแปลงตลอดเวลา อาจจะเกิดการเปลี่ยนแปลงโดยผู้ดูแลระบบเองหรือเกิดจากผู้ใช้คนอื่น ๆ ได้ แต่การสร้างเว็บเพจแบบไดนามิกนั้น ก็มีความยุ่งยากกว่าการสร้างเว็บเพจแบบสแตติกอยู่บ้าง เพราะการสร้างเว็บเพจแบบไดนามิกต้องมีการเขียนโปรแกรมระบบขึ้นมา โดยในระบบที่สร้างขึ้นมานั้นจะประกอบไปด้วยส่วนของโปรแกรมภาษาที่

ใช้ในการติดต่อระหว่างไคลเอนท์กับเซิร์ฟเวอร์ซึ่งภาษาที่ได้รับความนิยมมากที่สุดคือ PHP และในส่วนของการเก็บข้อมูลหรือที่เรียกว่าฐานข้อมูล โดยระบบฐานข้อมูลที่ใช้งานกันมากคือ MySQL ทำให้ในปัจจุบันนี้การแข่งขันทางด้านการพัฒนาเว็บไซต์มีมาก โดยเว็บไซต์ที่พัฒนาขึ้นได้มีการนำไปใช้งานตามที่ผู้พัฒนาสร้างขึ้นมา เช่น ด้านการศึกษา ด้านการค้า ด้านความบันเทิง เป็นต้น ระบบ Moodle เป็นอีกระบบหนึ่งที่มีผู้พัฒนาขึ้นสำหรับการศึกษา ซึ่งมีรูปแบบการพัฒนาเป็นแบบไดนามิกโดยใช้ภาษา PHP เป็นภาษาในการพัฒนาร่วมกับระบบฐานข้อมูล MySQL นั่นเอง ระบบ Moodle นั้นเป็นระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต มีโมดูลในการใช้งานหลากหลาย ทำให้มีการเก็บบันทึกข้อมูลไว้มาก เมื่อทำการค้นคืนข้อมูลแต่ละครั้งจึงต้องใช้เวลา การค้นคืนข้อมูลเป็นงานที่ต้องใช้เทคนิคที่ซับซ้อน และมักต้องใช้ความชำนาญในการเรียนรู้ระบบ เพื่อการนำมาทดสอบระบบให้มีความรวดเร็วในการค้นคืนข้อมูล ซึ่งได้มีการนำเสนอและวิเคราะห์ระบบต่าง ๆ ออกมาหลายระบบ แต่ละระบบก็มีข้อดีข้อเสียแตกต่างกันออกไป ตัวอย่างเช่น ระบบที่พัฒนาโดยใช้เทคโนโลยี AJAX ก็เป็นอีกเทคนิควิธีหนึ่งที่สามารถช่วยให้การค้นคืนข้อมูลระหว่างไคลเอนท์และเซิร์ฟเวอร์ทำงานได้ดีขึ้น และเทคนิควิธีระบบ Caching Output ก็เป็นอีกวิธีที่ช่วยเพิ่มประสิทธิภาพให้กับการค้นคืนข้อมูลได้อย่างมีประสิทธิภาพเช่นกัน ดังนั้น จึงจำเป็นจะต้องนำเทคนิคต่าง ๆ ที่มีอยู่แล้วมาทำการทดสอบเพื่อหาจุดเด่นของแต่ละเทคนิค และเพื่อเป็นการเริ่มต้นที่ดีสำหรับการพัฒนาเทคนิคการค้นคืนข้อมูลที่ดีขึ้นต่อไป โดยขั้นตอนและวิธีดำเนินการวิจัยจะได้กล่าวถึงในบทต่อไป

บทที่ 3
วิธีดำเนินการวิจัย

งานวิจัยนี้มีจุดประสงค์เพื่อศึกษาและเปรียบเทียบเทคนิคการค้นคืนข้อมูล ตลอดจนพัฒนาเทคนิคการค้นคืนข้อมูลที่มีประสิทธิภาพสูงขึ้นเพื่อใช้ในงานการค้นคืนข้อมูลในเว็บเซิร์ฟเวอร์ ที่มีขนาดใหญ่ สำหรับเนื้อหาในบทนี้จะได้นำเสนอขั้นตอนและวิธีดำเนินการวิจัย โดยเริ่มจากระเบียบวิธีวิจัย ดังแสดงในหัวข้อที่ 3.1 และหัวข้อที่ 3.2 จะนำเสนอข้อมูลที่ใช้ในการวิจัย ในหัวข้อที่ 3.3 จะนำเสนอเทคนิคการค้นคืนข้อมูลแบบวิธีการค้นคืนข้อมูลจากระบบ Caching System ของเอาต์พุต ซึ่งเป็นเทคนิคที่ผู้วิจัยได้พัฒนาขึ้นสำหรับงานการค้นคืนข้อมูลโดยเฉพาะ และในหัวข้อที่ 3.4 จะเป็นขั้นตอนการเปรียบเทียบประสิทธิภาพของอัลกอลิธึมการค้นคืนข้อมูล

3.1 ระเบียบวิธีวิจัย

3.1.1 ศึกษาค้นคว้าและรวบรวมงานวิจัยที่เกี่ยวข้อง

3.1.2 ศึกษาการใช้งานระบบ Moodle เพื่อใช้ในการทดสอบระบบในการค้นคืนข้อมูล

3.1.3 รวบรวมและพัฒนาอัลกอริธึมการค้นคืนข้อมูลที่มีคุณสมบัติเหมาะสมสำหรับระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่มีนักวิจัยท่านอื่นเคยเสนอไว้แล้ว โดยอัลกอริธึมที่เลือกใช้ในการวิจัยนี้คืออัลกอริธึม Caching Output

3.1.4 พัฒนาเครื่องมือที่จำเป็นเพื่อช่วยในการทดลอง ประกอบไปด้วย

1) SUTinsServer 5103 Plus Moodle เพื่อใช้ในการทดสอบเป็นระบบเว็บเซิร์ฟเวอร์

2) Moodle เพื่อใช้ในการเป็นระบบการทดสอบเทคนิคการค้นคืนข้อมูล และMoodle ที่ใช้ในการทดสอบนี้มี 3 เวอร์ชัน คือ Moodle1.6.6, Moodle1.8.4 และ Moodle2.0

3) Webserver Stress Tools เพื่อใช้ในการเปรียบเทียบการค้นคืนข้อมูลระหว่างระบบที่มีการค้นคืนข้อมูลแบบเดิม กับระบบที่ได้เพิ่มอัลกอริธึมการค้นคืนข้อมูลแบบ Caching Output ได้

3.1.5 ทดสอบประสิทธิภาพของอัลกอริธึมในการค้นคืนข้อมูลแบบต่าง ๆ เช่น อัลกอริธึมที่ใช้เทคโนโลยี AJAX อัลกอริธึมที่ใช้เทคนิคการค้นคืนข้อมูลจาก Cache Memory และอัลกอริธึมจากการ Caching Output เป็นต้น เพื่อวิเคราะห์หาลักษณะเด่นของแต่ละอัลกอริธึม ที่จะนำมาใช้ประกอบการออกแบบอัลกอริธึมใหม่

3.1.6 ออกแบบและพัฒนาอัลกอริธึมการค้นคืนข้อมูลแบบใหม่ที่มีประสิทธิภาพสูงขึ้น

3.1.7 ทดสอบประสิทธิภาพของอัลกอริธึมการค้นคืนข้อมูลแบบใหม่ที่พัฒนาขึ้น

3.1.8 วิเคราะห์และสรุปผลการวิจัย

3.2 ข้อมูลที่ใช้ในการวิจัย

สำหรับข้อมูลที่ใช้ในการวิจัยนี้ เป็นชุดข้อมูลที่ได้จากระบบ Moodle ซึ่งเป็นระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่ได้รับความน่าสนใจเป็นอย่างมากในปัจจุบัน โดยใน Moodle นั้นจะเป็นชุดของสคริปส์ที่มีการทำงานอยู่ฝั่งเซิร์ฟเวอร์ โดยจะใช้โปรแกรมภาษา PHP ในการพัฒนา ซึ่งเหมาะสำหรับสถาบันการศึกษาเพื่อใช้เตรียมแหล่งข้อมูล กิจกรรม อีกทั้งยังเผยแพร่แบบออนไลน์ผ่านเครือข่ายอินเทอร์เน็ตหรืออินทราเน็ต และสามารถนำไปใช้ได้ทั้งองค์กรระดับมหาวิทยาลัย โรงเรียน สถาบัน หรือครูสอนพิเศษ โปรแกรมชุดนี้เป็นโอเพนซอร์สภายใต้ข้อตกลงของ GNU (General Public License) สำหรับผู้ดูแลระบบ (Admin) ที่จะนำโปรแกรมไปติดตั้ง ต้องมีเว็บเซิร์ฟเวอร์ที่ให้บริการ PHP และ MySQL

ระบบ Moodle เป็นระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่มีผู้นิยมนำไปใช้ในการเรียนการสอนมากที่สุด อีกทั้งยังมีโมดูลอีกมากมายที่ใช้สำหรับการเรียนการสอน โมดูลหลักที่มีมากับระบบ เช่น Assignment, Chat , Choice, Data, Forum, Glossary, Hotpot, Journal, Label, Leson, Quiz, Resource, Scorm, Servey, Wiki, Workshop, Blog และอื่นๆ ส่วนโมดูลเสริมที่มีมาให้ได้ทดลองใช้ เช่น โมดูลกิจกรรม Audio record, Book, Dialogue, Certificate, Flash, Gallery, Ipodcast, Object, Project, Questionaire, Portfolio, WebQuest, Link to Learning Object และ Block เพิ่มเติม หากไม่ต้องการใช้งานโมดูลใด ๆ สามารถยกเลิกการใช้งานได้ตามต้องการ

จากการที่มีจำนวนโมดูลการใช้งานที่มีอยู่เป็นจำนวนมากนั้น จึงทำให้ฐานข้อมูลที่มีอยู่ในระบบ Moodle จึงมีมากตามไปด้วย ซึ่งทำให้เมื่อระบบ Moodle ถูกเรียกใช้งาน จะต้องทำการค้นหาข้อมูลต่าง ๆ จากระบบฐานข้อมูลที่มีมาทั้งหมด ทำให้เกิดความล่าช้าในการค้นคืนข้อมูลของระบบดังกล่าว ดังแสดงตารางฐานข้อมูล

ตารางที่ 3.1 แสดงตารางฐานข้อมูลที่ใช้ในระบบ Moodle

ตาราง	ระเบียน	ขนาด (กิโลไบต์)
adodb_logsql	0	1.0
mdl_assignment	0	1.0
mdl_assignment_submissions	0	1.0
mdl_backup_config	0	1.0
mdl_backup_courses	0	1.0
mdl_backup_files	0	1.0
mdl_backup_ids	0	1.0
mdl_backup_log	0	1.0
mdl_block	20	2.5
mdl_block_instance	5	4.2
mdl_block_rss_client	0	1.0
mdl_cache_filters	0	1.0
mdl_cache_text	58	5.5
mdl_chat	0	1.0
mdl_chat_messages	0	1.0
mdl_chat_users	0	1.0
mdl_choice	0	1.0
mdl_choice_answers	0	1.0
mdl_choice_options	0	1.0
mdl_config	171	15.7
mdl_config_plugins	0	1.0
mdl_course	1	5.1
mdl_course_categories	1	3.0
mdl_course_display	0	1.0
mdl_course_meta	0	1.0
mdl_course_modules	0	1.0
mdl_course_sections	0	1.0
mdl_enrol_authorize	0	1.0

ตารางที่ 3.1 แสดงตารางฐานข้อมูลที่ใช้ในระบบ Moodle (ต่อ)

ตาราง	ระเบียน	ขนาด (กิโลไบต์)
mdl_enrol_paypal	0	1.0
mdl_event	0	1.0
mdl_forum	0	1.0
mdl_forum_discussions	0	1.0
mdl_forum_posts	0	1.0
mdl_forum_queue	0	1.0
mdl_forum_ratings	0	1.0
mdl_forum_read	0	1.0
mdl_forum_subscriptions	0	1.0
mdl_forum_track_prefs	0	1.0
mdl_glossary	0	1.0
mdl_glossary_alias	0	1.0
mdl_glossary_categories	0	1.0
mdl_glossary_comments	0	1.0
mdl_glossary_entries	0	1.0
mdl_glossary_entries_categories	0	1.0
mdl_glossary_formats	0	1.0
mdl_glossary_ratings	0	1.0
mdl_grade_category	0	1.0
mdl_grade_exceptions	0	1.0
mdl_grade_item	0	1.0
mdl_grade_letter	0	1.0
mdl_grade_preferences	0	1.0
mdl_groups	0	1.0
mdl_groups_members	0	1.0
mdl_hotpot	0	1.0
mdl_hotpot_attempts	0	1.0

ตารางที่ 3.1 แสดงตารางฐานข้อมูลที่ใช้ในระบบ Moodle (ต่อ)

ตาราง	ระเบียน	ขนาด (กิโลไบต์)
mdl_hotpot_questions	0	1.0
mdl_hotpot_responses	0	1.0
mdl_hotpot_strings	0	1.0
mdl_journal	0	1.0
mdl_journal_entries	0	1.0
mdl_label	0	1.0
mdl_lesson	0	1.0
mdl_lesson_answers	0	1.0
mdl_lesson_attempts	0	1.0
mdl_lesson_branch	0	1.0
mdl_lesson_default	0	1.0
mdl_lesson_grades	0	1.0
mdl_lesson_high_scores	0	1.0
mdl_lesson_pages	0	1.0
mdl_lesson_timer	0	1.0
mdl_log	177	24.8
mdl_log_display	94	4.4
mdl_message	0	1.0
mdl_message_contacts	0	1.0
mdl_message_read	0	1.0
mdl_modules	15	4.4
mdl_quiz	0	1.0
mdl_quiz_answers	0	1.0
mdl_quiz_attemptonlast_datasets	0	1.0
mdl_quiz_attempts	0	1.0
mdl_quiz_calculated	0	1.0
mdl_quiz_categories	0	1.0

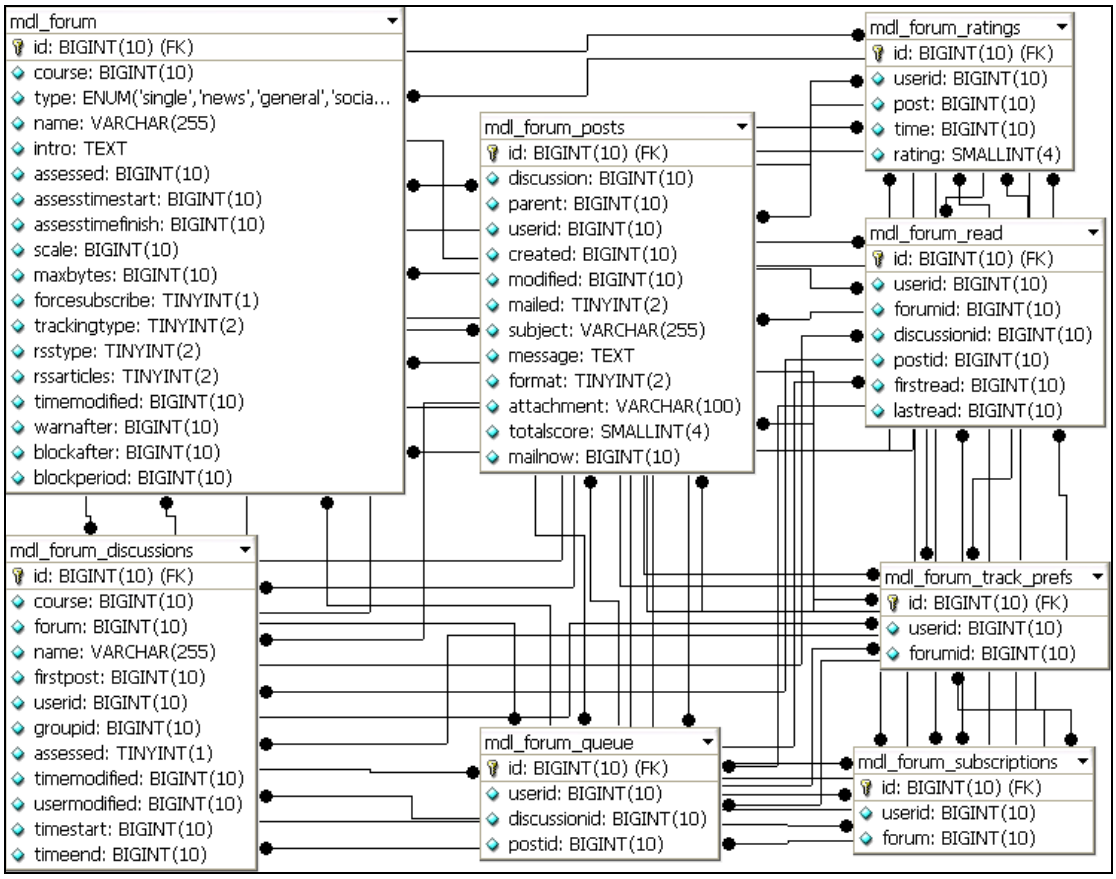
ตารางที่ 3.1 แสดงตารางฐานข้อมูลที่ใช้ในระบบ Moodle (ต่อ)

ตาราง	ระเบียน	ขนาด (กิโลไบต์)
mdl_quiz_dataset_definitions	0	1.0
mdl_quiz_dataset_items	0	1.0
mdl_quiz_match	0	1.0
mdl_quiz_match_sub	0	1.0
mdl_quiz_multianswers	0	1.0
mdl_quiz_multichoice	0	1.0
mdl_quiz_newest_states	0	1.0
mdl_quiz_numerical	0	1.0
mdl_quiz_numerical_units	0	1.0
mdl_quiz_question_datasets	0	1.0
mdl_quiz_question_instances	0	1.0
mdl_quiz_question_versions	0	1.0
mdl_quiz_questions	0	1.0
mdl_quiz_randomsamatch	0	1.0
mdl_quiz_rqp	0	1.0
mdl_quiz_rqp_servers	0	1.0
mdl_quiz_rqp_states	0	1.0
mdl_quiz_rqp_types	0	1.0
mdl_quiz_shortanswer	0	1.0
mdl_quiz_states	0	1.0
mdl_quiz_truefalse	0	1.0
mdl_resource	0	1.0
mdl_scale	0	1.0
mdl_scorm	0	1.0
mdl_scorm_scoes	0	1.0
mdl_scorm_scoes_track	0	1.0

ตารางที่ 3.1 แสดงตารางฐานข้อมูลที่ใช้ในระบบ Moodle (ต่อ)

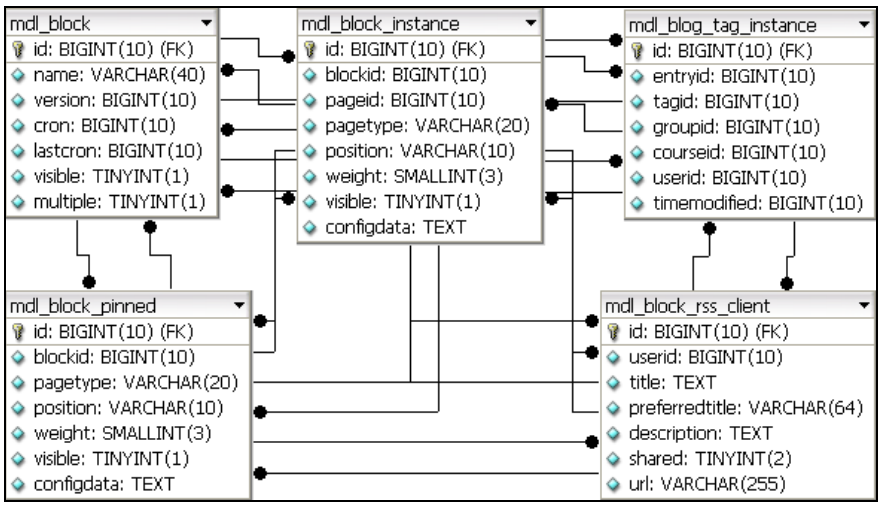
ตาราง	ระเบียน	ขนาด (กิโลไบต์)
mdl_sessions	0	1.0
mdl_survey	5	3.3
mdl_survey_analysis	0	1.0
mdl_timezone	0	1.0
mdl_user	2	15.3
mdl_user_admins	1	4.0
mdl_user_coursecreators	0	1.0
mdl_user_preferences	2	4.1
mdl_user_students	0	1.0
mdl_user_teachers	1	6.0
mdl_wiki	0	1.0
mdl_wiki_entries	0	1.0
mdl_wiki_pages	0	1.0
mdl_workshop	0	1.0
mdl_workshop_assessments	0	1.0
mdl_workshop_comments	0	1.0
mdl_workshop_elements	0	1.0
mdl_workshop_grades	0	1.0
mdl_workshop_rubrics	0	1.0
mdl_workshop_stockcomments	0	1.0
mdl_workshop_submissions	0	1.0

โดยข้อมูลในตารางฐานข้อมูลนั้นสามารถแสดงความสัมพันธ์ของฐานข้อมูลดังตัวอย่างได้ เช่น ความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่แสดงอยู่ในตารางความสัมพันธ์เกี่ยวกับโมดูล forum

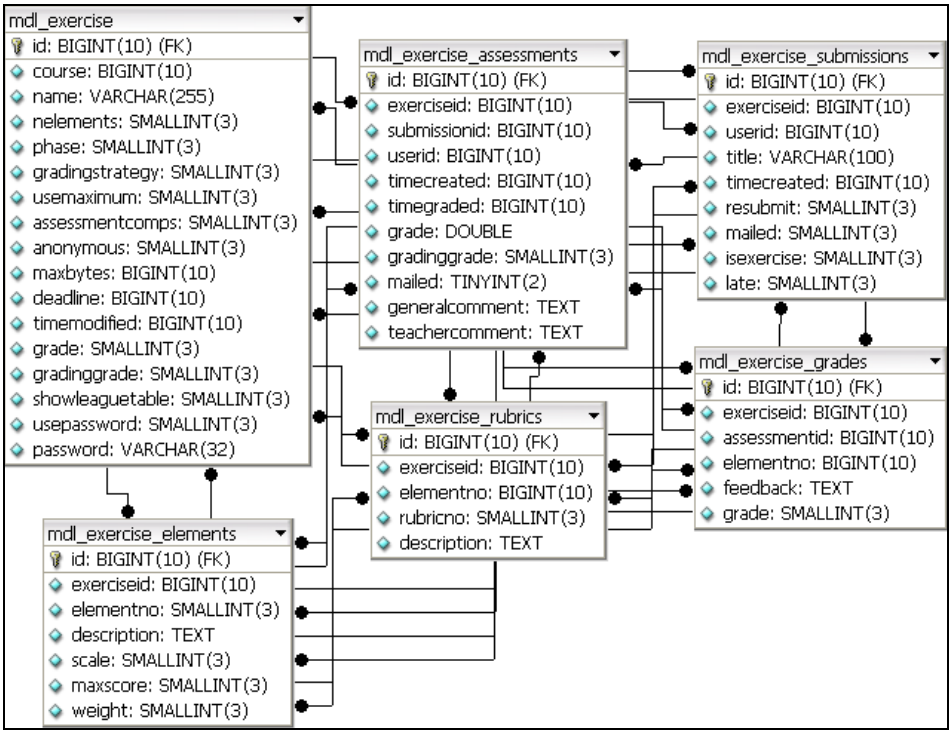


รูปที่ 3.1 แสดงความสัมพันธ์ของฐานข้อมูลของโมดูล forum

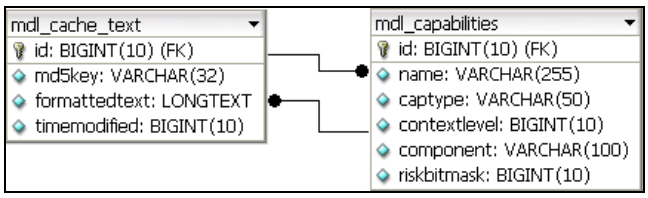
จากความสัมพันธ์ของตารางในฐานข้อมูลในโมดูล forum นั้นจะเห็นได้ว่าเมื่อนำมาแสดงเป็นความสัมพันธ์แบบ E-R Diagram แล้ว ตารางในฐานข้อมูลดังกล่าวนั้นมีความสัมพันธ์กันค่อนข้างซับซ้อน ซึ่งหากเมื่อผู้ใช้มีการเรียกใช้งานโมดูล forum ดังกล่าว จึงทำให้ Moodle ต้องทำการค้นหา หรือเรียกใช้ข้อมูลในส่วนต่าง ๆ ที่มีความัมพันธ์กัน ซึ่งนอกจากโมดูล furum ที่มีความสัมพันธ์กันแล้ว โมดูลอื่น ๆ ที่ใช้งานในระบบ Moodle นั้นยังมีอีกมากมาย ดังแสดงในรูปแบบ E-R Diagram ของโมดูลต่าง ๆ



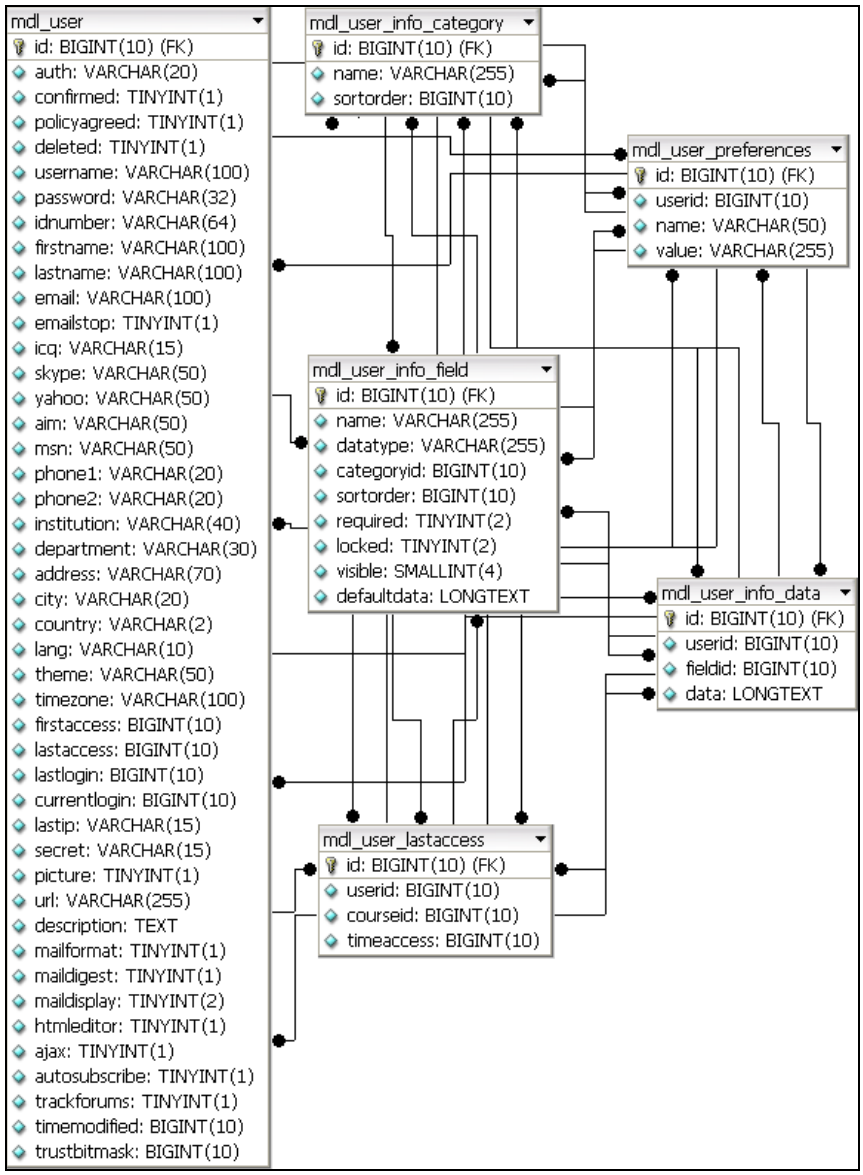
รูปที่ 3.2 แสดงความสัมพันธ์ของฐานข้อมูล block



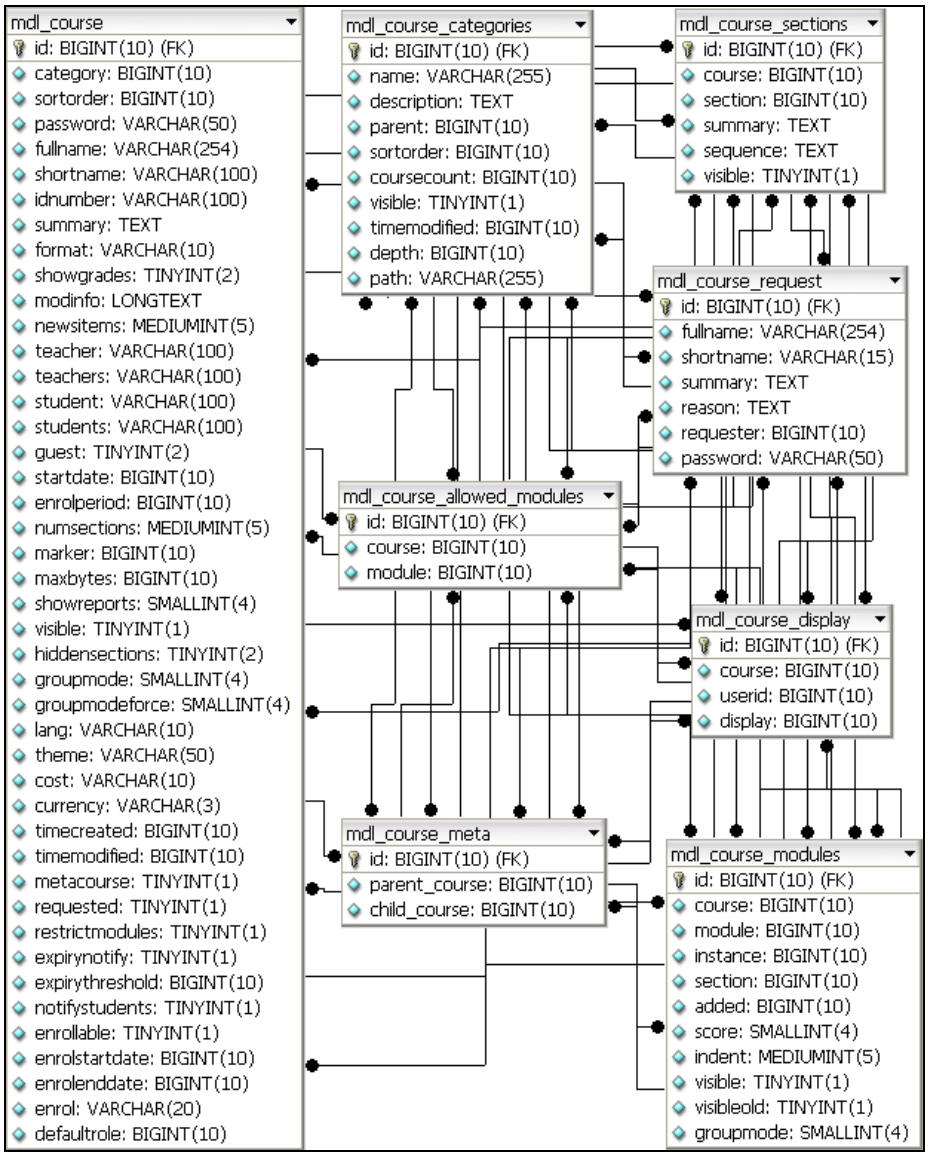
รูปที่ 3.3 แสดงความสัมพันธ์ของฐานข้อมูล excercise



รูปที่ 3.4 แสดงความสัมพันธ์ของฐานข้อมูล cache_text



รูปที่ 3.5 แสดงความสัมพันธ์ของฐานข้อมูล user



รูปที่ 3.6 แสดงความสัมพันธ์ของฐานข้อมูล course

3.3 เทคนิคการค้นคืนข้อมูลแบบวิธีการค้นคืนข้อมูลจากระบบ Caching ของเอาท์พุต

ในส่วนนี้จะเป็นการนำเสนอเทคนิคการค้นคืนข้อมูล แบบวิธีการค้นคืนข้อมูลจากระบบ Caching ของเอาท์พุตของข้อมูล (A Caching Output in Learning Management System : CO-LMS) ซึ่งเป็นเทคนิคที่พัฒนาขึ้น สำหรับงานการค้นคืนข้อมูลในระบบการจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตโดยเฉพาะ ลักษณะที่สำคัญของอัลกอริธึมมีดังต่อไปนี้

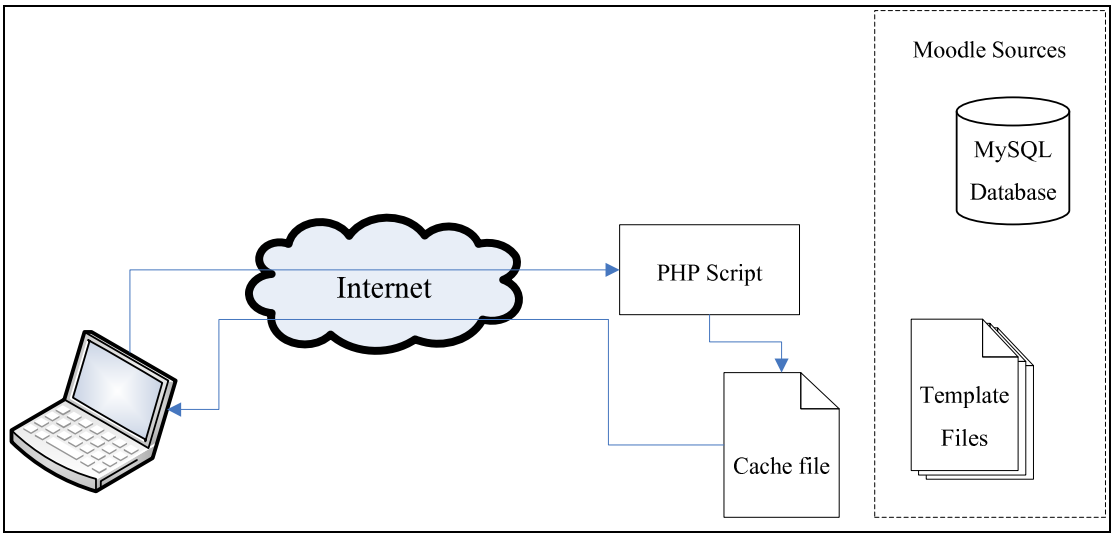
1) สามารถค้นคืนข้อมูลจากเว็บเซิร์ฟเวอร์ได้ ซึ่งจะนำข้อมูลที่มีผู้เข้าชมเว็บไซต์มาเก็บในระบบ Caching และส่งต่อให้กับผู้เข้าชมคนต่อไป

2) สามารถรองรับกับระบบของเว็บไซต์ที่มีรูปแบบแตกต่างกันออกไปได้

สำหรับเทคนิคการค้นคืนข้อมูลแบบใหม่ที่พัฒนาขึ้นนั้น (CO-LMS) เป็นเทคนิคที่ใช้โครงสร้างหลักของฟังก์ชันในโปรแกรมภาษา PHP ซึ่งถือได้ว่าเป็นโปรแกรมภาษาที่ได้รับความนิยมเป็นอย่างมากในปัจจุบัน เนื่องจากโปรแกรมภาษา PHP นี้เป็นโอเพนซอร์ส ทำให้ง่ายต่อการพัฒนา

3.3.1 โครงสร้างของระบบ CO-LMS

โครงสร้างการทำงานโดยทั่วไปของระบบ CO-LMS นั้น เมื่อผู้ใช้ร้องขอข้อมูลผ่านระบบอินเทอร์เน็ตมาที่ฝั่งเซิร์ฟเวอร์ สคริปต์ของ PHP จะทำงานและตรวจสอบข้อมูลที่ทำการร้องขอมา จากนั้นจะทำการส่งข้อมูลที่ร้องขอกลับไปที่ไคลเอนท์ ระหว่างนั้นระบบ CO-LMS จะทำการบันทึกข้อมูลต่าง ๆ ที่มีการทำงานนั้น ตั้งแต่เริ่มต้นการค้นข้อมูล จนกระทั่งถึงกระบวนการคืนข้อมูลกลับไปที่ไคลเอนท์ เมื่อกระบวนการสิ้นสุด ระบบจะทำการบันทึกข้อมูลดังกล่าวในรูปแบบ html และ Text ไฟล์ แล้วนำไปเก็บไว้ที่ระบบ เมื่อไคลเอนท์ร้องขอข้อมูลมาอีกครั้ง ระบบจะทำการตรวจสอบว่าข้อมูลดังกล่าวตรงกับข้อมูลที่มีอยู่ในระบบหรือไม่ ถ้าหากมีข้อมูลที่ตรงกันระบบก็จะทำการส่งข้อมูลดังกล่าวไปให้กับไคลเอนท์ทันที ทำให้ระบบไม่ต้องทำการค้นข้อมูลจากเซิร์ฟเวอร์ใหม่ทุกครั้ง

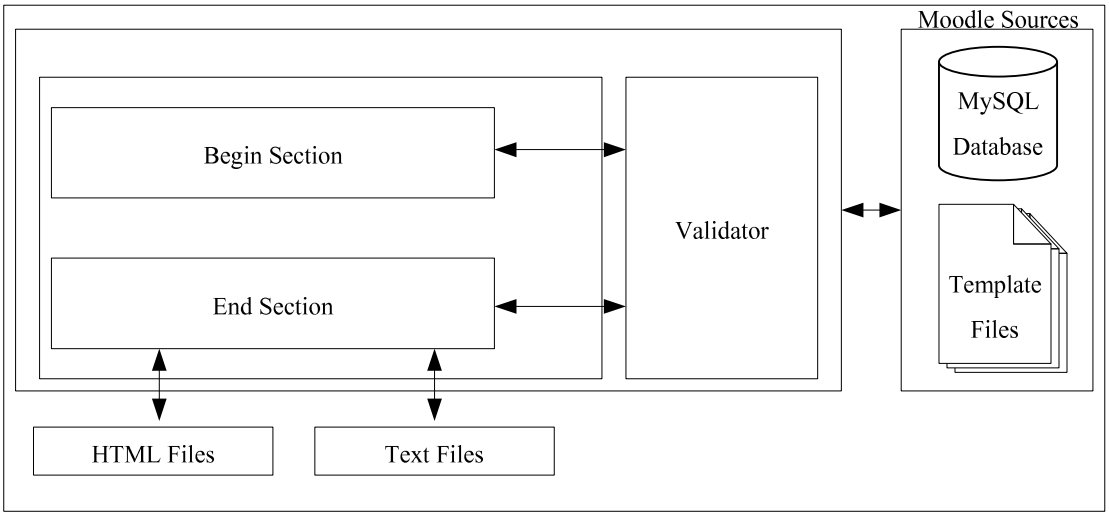


รูปที่ 3.7 แสดงโครงสร้างของระบบโดยทั่วไป

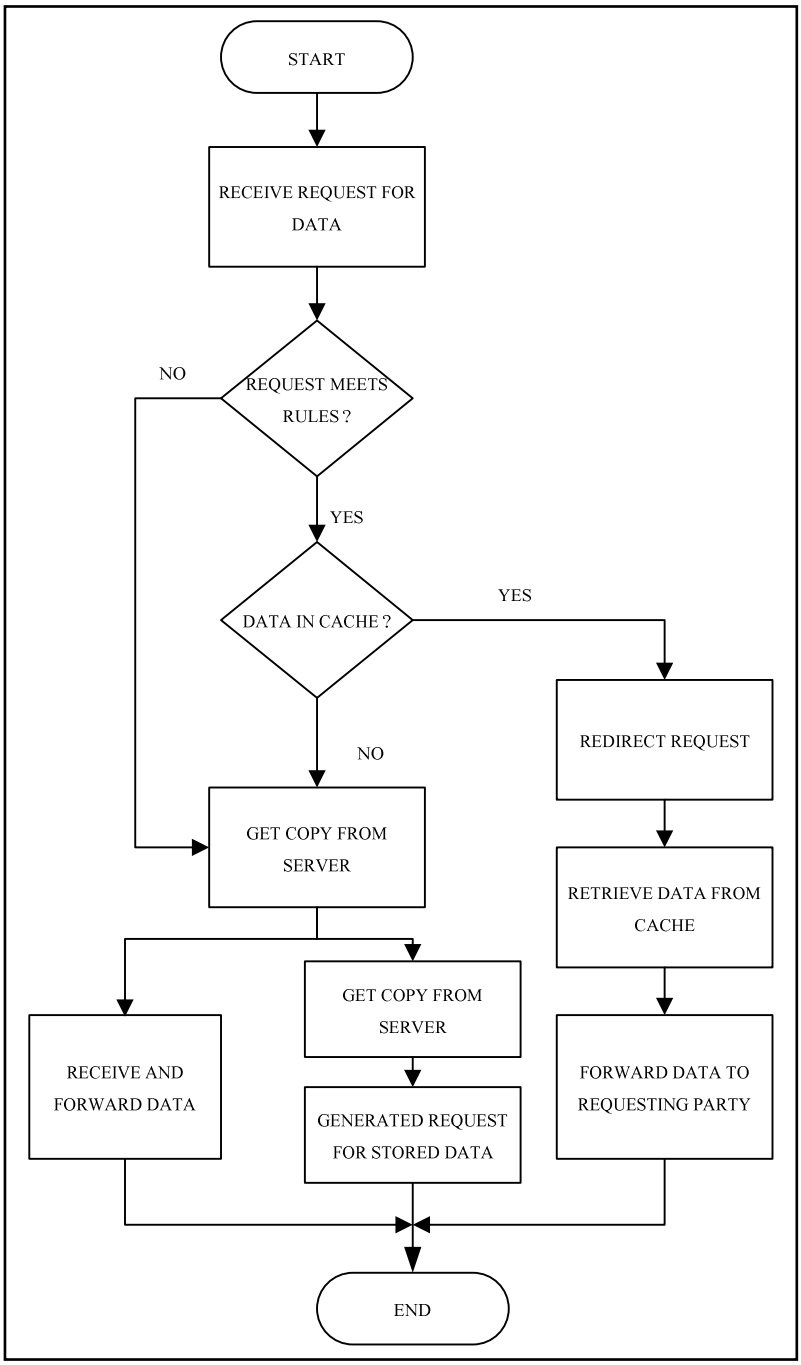
ระบบที่พัฒนาขึ้นนี้ประกอบด้วยส่วนหลักอยู่ 2 ส่วน คือส่วนเริ่มต้น (Beginning Section) และ ส่วนสุดท้าย (Ending Section) โดยมีโครงสร้างของระบบดังรูปที่ 3.7 ซึ่งการ Caching เอาท์พุตใน PHP นั้น จะเป็นการใช้งานบัฟเฟอร์ของฟังก์ชันที่มีมากับ PHP เวอร์ชัน 4 ขึ้นไป โดยในแต่ละส่วนจะถูกเรียกใช้เมื่อมีการร้องขอข้อมูลจากผู้ใช้ ซึ่งมีรายละเอียดต่าง ๆ ดังนี้

1) ส่วนเริ่มต้น เป็นส่วนโมดูลเริ่มแรกที่มีการทำงานของระบบ โดยเมื่อผู้ใช้ทำการร้องขอข้อมูลมาที่เว็บเซิร์ฟเวอร์ดังกล่าว เว็บเซิร์ฟเวอร์จะทำการประมวลผลคำสั่งและเรียกใช้งานระบบ CO-LMS และเมื่อระบบเริ่มทำงาน โมดูลส่วนเริ่มต้นจะเป็นส่วนแรกที่ถูกเรียกใช้งาน โดยในส่วนนี้จะทำหน้าที่ในการตรวจสอบว่าไฟล์ PHP Script ที่ถูกเรียกใช้งานอยู่นั้น มีเก็บไว้ในระบบหรือไม่ โดยจะทำการเปรียบเทียบโดยการตรวจสอบกับค่าของรหัส MD5 ของไฟล์ที่ถูกเก็บไว้ในส่วนของ Cache ในระบบ ซึ่งถ้าหากไฟล์ในระบบมีค่าของรหัส MD 5 ที่ตรงกันกับไฟล์ที่ถูกเรียกใช้งาน ระบบ CO-LMS ก็จะทำการส่งไฟล์ที่มีในระบบไปให้กับผู้ใช้แทน แต่ถ้าหากยังไม่มีระบบ CO-LMS ก็จะทำการอ่านค่ารหัสต่าง ๆ ของไฟล์ PHP Script ที่แสดงอยู่ในหน้าเว็บเพจที่ผู้ใช้ต้องการ จากนั้นจะทำการบันทึกเป็นไฟล์ในรูปแบบภาษา HTML หรือไฟล์ข้อความต่อไป

2) ส่วนสุดท้าย ซึ่งเป็นส่วนของโมดูลที่ทำหน้าที่ในการสิ้นสุดกระบวนการการทำงานของระบบ เมื่อระบบได้ทำงานมาถึงส่วนนี้ระบบจะทำการบันทึกไฟล์ PHP Script นั้นเก็บเอาไว้ และเมื่อมีผู้ใช้คนต่อไปทำการร้องขอข้อมูลทีมีค่าของรหัส MD5 ตรงกับไฟล์ที่ได้บันทึกนั้น ระบบก็จะส่งไฟล์ที่มีอยู่นั้นไปให้แทน ทำให้เว็บเซิร์ฟเวอร์ไม่ต้องทำการค้นหาข้อมูลจากฐานข้อมูลจริง หรือจากระบบจริงได้ ซึ่งจะเป็นการช่วยลดภาระในการค้นคืนข้อมูลให้กับเว็บเซิร์ฟเวอร์ได้มาก



รูปที่ 3.8 แสดงส่วนการทำงานของระบบ CO-LMS



ฟังก์ชัน Output Control เป็นฟังก์ชันที่มีอยู่ในไลบรารีของภาษา PHP จึงไม่จำเป็นต้องใช้ไลบรารีจากภายนอกเพื่อขยายการทำงาน และไม่จำเป็นต้องทำการติดตั้งเพิ่มเติม ซึ่งเมื่อต้องการให้ฟังก์ชันนี้ทำงาน จะต้องมีการตั้งค่าในไฟล์ php.ini เมื่อฟังก์ชันทำงานจะทำให้สามารถควบคุม

เอาท์พุตที่ถูกส่งออกมาจากสคริปท์ซึ่งถือได้ว่าเป็นประโยชน์อย่างมากในหลาย ๆ กรณี โดยเฉพาะเมื่อต้องการจะส่งเฮดเดอร์ไปยังเบราเซอร์ หลังจากที่สคริปท์เริ่มต้นทำการส่งข้อมูลออกไป โดยที่ฟังก์ชัน Output Control จะไม่ส่งผลต่อเฮดเดอร์ที่ส่งออกโดยใช้ฟังก์ชัน header() หรือ ฟังก์ชัน setcookie() โดยเฉพาะฟังก์ชัน echo() และข้อมูลระหว่างบล็อกของโค้ด PHP ในการตั้งค่าให้กับฟังก์ชันจะต้องมีการเลือกองค์ประกอบดังแสดงในตาราง

ตารางที่ 3.2 การเลือกองค์ประกอบ Output Control

Name	Default	Changeable	Changelog
output_buffering	0	PHP_INI_PERDIR	
output_handler	NULL	PHP_INI_PERDIR	Available since PHP 4.0.4.
implicit_flush	0	PHP_INI_ALL	PHP_INI_PERDIR in PHP <= 4.2.3.

ฟังก์ชัน Output Control สามารถทำให้เอาท์พุตบัฟเฟอร์ทำงานสำหรับทุก ๆ ไฟล์โดยที่ทำการกำหนดการตั้งค่าให้เป็น ‘on’ ถ้าอยากให้จำกัดขอบเขตขนาดของบัฟเฟอร์ให้แน่นอนสามารถใช้ค่าตัวเลขสูงสุดของไบต์แทนที่ ‘on’ ซึ่งเป็นค่าที่กำหนด เช่น output_buffering=4096 เหมือนกับ PHP 4.3.5 ที่ถูกกำหนดให้เป็น Off ตลอดเวลาใน PHP-CLI ในการกำหนดค่าเพิ่มเติมจะไม่มีการกำหนดค่าทรัพยากรที่คงที่ และไม่มีการกำหนดค่าคงที่ใด ๆ สำหรับตัวอย่างในการใช้งานฟังก์ชันมีดังนี้

```
<?php
      ob_start();
      echo"Hello\n";
      setcookie("cookiename","cookiedata");
      ob_end_flush();
?>
```

จากตัวอย่างดังกล่าว สิ่งที่นำออกมาจาก echo() จะถูกตั้งอยู่ในบริเวณเก็บข้อมูลชั่วคราว สิ่งที่นำออกจนกระทั่ง ob_end_flush() ในความหมายของเวลา การ setcookie() จะประสบความสำเร็จขึ้นอยู่กับ cookie จะปราศจากการเป็นสาเหตุให้เกิดข้อผิดพลาด (โดยปกติแล้วจะไม่สามารถส่ง header หลังข้อมูลได้ถูกส่งไปแล้ว

เมื่อเกิดการเปลี่ยนใหม่จาก PHP เวอร์ชัน 4.1 (และเวอร์ชัน 4.2) เป็นเวอร์ชัน 4.3 จุดบกพร่องที่เกิดขึ้นในเวอร์ชันแรก ๆ จะต้องแน่ใจได้ว่า implict_flush จะถูกยกเลิกใน php.ini

มิฉะนั้นสิ่งที่ถูกส่งออกไปกับ ob_start() จะไม่ถูกซ่อนจากสิ่งที่นำออก

ตัวอย่างฟังก์ชันและการทำงาน

flush -- บริเวณไหลข้อมูลชั่วคราวของเอาท์พุต

ob_clean -- ทำความสะอาด (ลบ) บริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_end_clean -- ทำความสะอาด (ลบ) บริเวณเก็บข้อมูลที่นำออกชั่วคราวและปิดการบริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_end_flush -- Flush (ส่ง) บริเวณเก็บข้อมูลที่นำออกชั่วคราวและปิดการบริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_flush -- Flush (ส่ง) บริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_get_clean -- รับสิ่งที่บรรจุบริเวณเก็บข้อมูลชั่วคราวในปัจจุบันและลบบริเวณเก็บข้อมูลที่นำออกชั่วคราวในปัจจุบัน

ob_get_contents -- คืนสิ่งที่บรรจุของบริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_get_flush -- Flush บริเวณเก็บข้อมูลที่นำออกชั่วคราว, คืนค่าที่เป็นข้อความและปิดการบริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_get_length -- คืนความยาวของบริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_get_level -- คืนระดับที่ที่อยู่ของกลไกที่บริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_get_status -- รับสถานะของบริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_gzhandler -- ob_start เรียกกลับเพื่อ gzip บริเวณเก็บข้อมูลที่นำออกชั่วคราว

ob_implicit_flush -- รายการ implicit flush เปิดหรือปิด

ob_list_handlers -- แสดงรายการสิ่งที่นำออกที่ทุกคนอยู่ในการใช้

ob_start -- เปิดบริเวณเก็บข้อมูลที่นำออกชั่วคราว

output_add_rewrite_var -- เพิ่มค่าผู้เขียนใหม่ URL

output_reset_rewrite_vars -- คืนค่าผู้เขียนใหม่ URL

3.3.2 กรณีทดสอบที่ใช้ทดสอบระบบ

เป็นการสร้างกรณีทดสอบสำหรับระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต โดยได้ทำการทดสอบกับระบบ Moodle โดยระบบที่นำมาทดสอบนั้น ได้ทำการติดตั้งมาพร้อมกับระบบปฏิบัติการ SUTinsServer 5103 Plus Moodle โดยเป็นระบบลินุกซ์เซิร์ฟเวอร์พร้อมใช้ สำหรับพัฒนาระบบจัดการการเรียนรู้ผ่านเครือข่ายอินเทอร์เน็ต ซึ่งมี 3 เวอร์ชัน คือ Moodle1.6.6, Moodle1.8.4 และ Moodle2.0

3.3.3 ขั้นตอนการทดสอบ

การทดสอบระบบการค้นคืนข้อมูลโดยวิธี Caching Output โดยใช้โปรแกรมภาษา PHP นั้น ต้องทำการกำหนดไฟล์ขึ้นมา 2 ไฟล์ ดังนี้

1) begin_caching.php

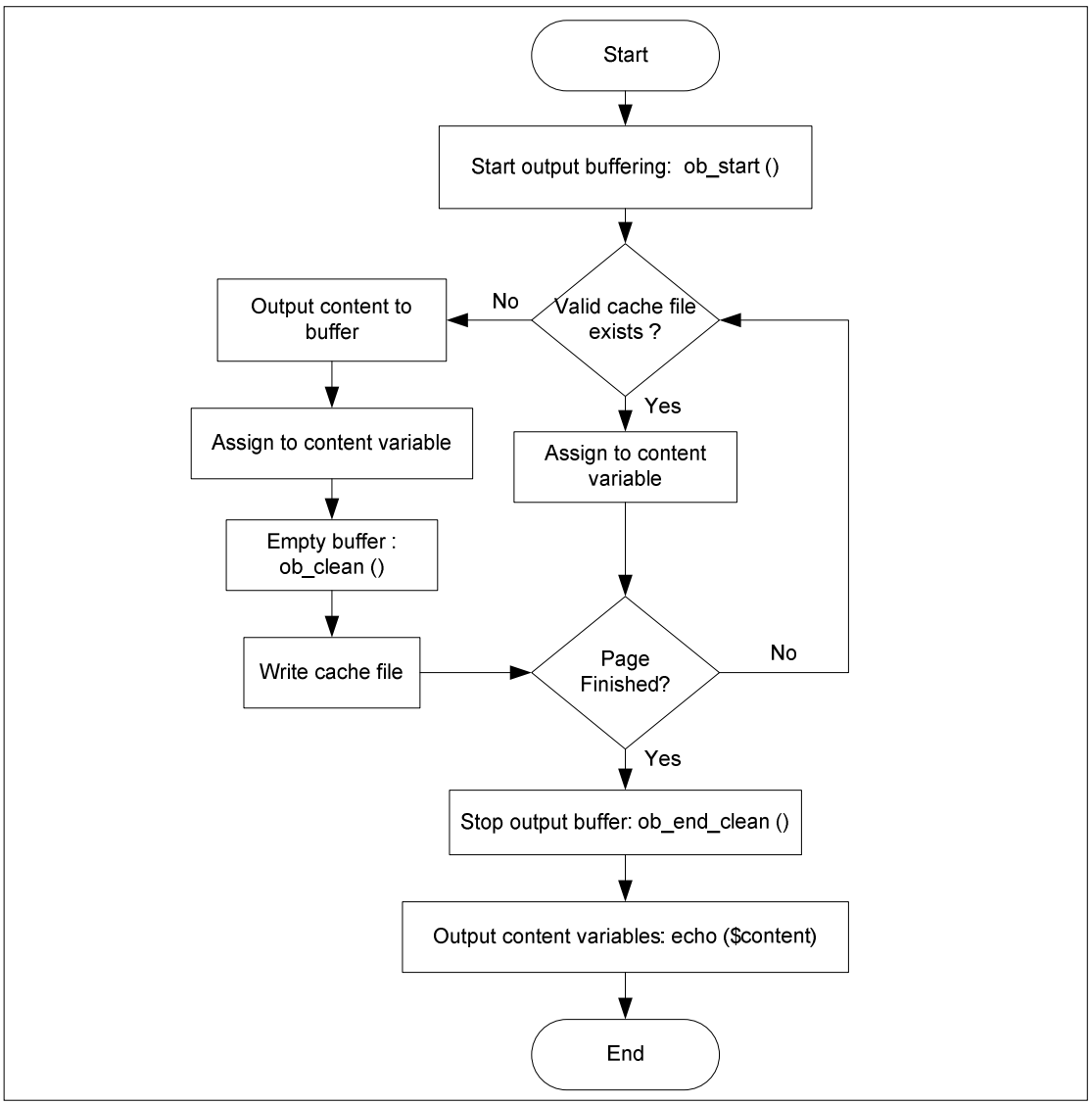
2) end_caching.php

เมื่อได้สร้างไฟล์สองไฟล์ขึ้นมาแล้ว ให้ทำการบันทึกไฟล์ทั้งสองไฟล์ไปเก็บไว้ในส่วนของเซิร์ฟเวอร์ที่ทำการเก็บเว็บเพจเอาไว้ ซึ่งในการทำวิจัยนี้ได้ใช้เซิร์ฟเวอร์ลินุกซ์ในการทดสอบ ดังนั้นจึงต้องทำการบันทึกไฟล์ดังกล่าวไปเก็บไว้ที่ /var/www/html ซึ่งเป็นส่วนที่ได้มีการระบุเอาไว้ว่าให้เป็นส่วนที่มีการรันข้อมูลเว็บเพจทุกครั้งที่มีการร้องขอมาจากไคลแอนท์

จากนั้นให้กำหนดการทำงานของระบบการค้นคืนข้อมูลจากการ Caching Output ของเว็บเซิร์ฟเวอร์ด้วยโปรแกรมภาษา PHP ซึ่งในขั้นตอนนี้มีวิธีการกำหนดการทำงานของระบบโดยแบ่งออกได้เป็น 2 วิธีคือ

1) ใช้ฟังก์ชัน include() โดยทำการกำหนดให้กับทุกสคริปท์ หรือทุกไฟล์ของเว็บเพจที่ต้องการให้ระบบทำงาน

2) ทำการกำหนดให้กับระบบทั้งระบบ ซึ่งเมื่อมีการร้องขอมาจากไคลแอนท์ให้ทำการเรียกใช้ระบบทุกครั้ง โดยการกำหนดดังกล่าวคือ การแก้ไขไฟล์ php.ini ซึ่งเป็นไฟล์ที่เมื่อมีการทำงานของโปรแกรมภาษา PHP ไฟล์ดังกล่าวจะทำการตรวจสอบข้อมูลหรือทำงานในการติดต่อระหว่างเซิร์ฟเวอร์และไคลแอนท์ ซึ่งเป็นตัวกลางในการแปลข้อมูลให้กับทั้งสองฝั่งนั่นเอง ข้อมูลที่ต้องทำการแก้ไขนั้นให้กำหนดในส่วนของไฟล์ begin_caching.php เริ่มทำงานทุกครั้งที่มีการร้องขอข้อมูลมาจากไคลแอนท์หรือที่เรียกว่า ส่วน header และในส่วนของไฟล์ end_caching.php นั้นกำหนดในส่วนสุดท้ายของการทำงานหรือที่เรียกว่า ส่วน footer นั่นเอง



รูปที่ 3.10 แสดงขั้นตอนการทำงานของระบบ CO-LMS

3.4 การเปรียบเทียบประสิทธิภาพของอัลกอริธึมการค้นคืนข้อมูล

ในหัวข้อนี้จะกล่าวถึงการเปรียบเทียบการค้นคืนข้อมูลจากระบบปฏิบัติการ Windows Server 2003 และระบบปฏิบัติการ SUTinsServer 5103 Plus Moodle ซึ่งเป็นระบบปฏิบัติการลินุกซ์ และได้นำโปรแกรม Webserver Stress Tool ใช้ในการทดสอบเพื่อทำการหาประสิทธิภาพของระบบปฏิบัติการทั้งสองแบบดังกล่าว

โปรแกรม Webserver Stress Tool คือ ประเภทของซอฟต์แวร์ซึ่งทำหน้าที่ในการทดสอบประสิทธิภาพการทำงานของเซิร์ฟเวอร์ด้วยการจำลองเป็นผู้เยี่ยมชมเว็บ โดยใช้ Webserver Stress Tool ในการตรวจสอบข้อมูลดังนี้

1) หา Maximize Uptime เพื่อต้องการทราบว่าเซิร์ฟเวอร์ สามารถรองรับจำนวนผู้เข้าชมได้มากน้อยเท่าไหร่ ซึ่งจะนำไปสู่การแก้ปัญหาก่อนเซิร์ฟเวอร์จะหยุดการทำงาน

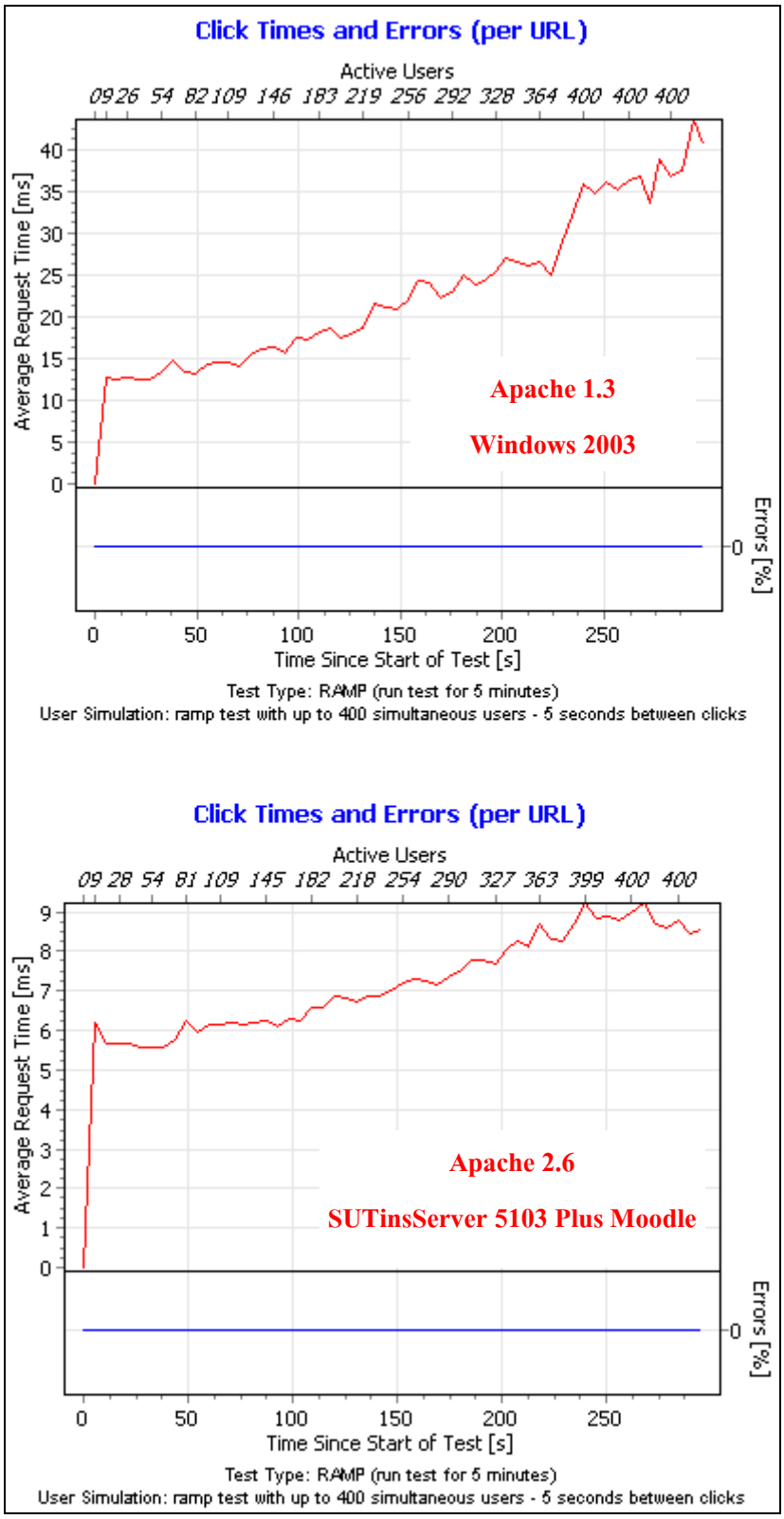
2) หา Maximize Performance เพื่อต้องการทราบว่าเซิร์ฟเวอร์จะให้ประสิทธิภาพสูงสุดและมีคุณภาพสูงสุด ณ จุดใด (หมายถึง จุดที่เซิร์ฟเวอร์จะสามารถทำงานได้อย่างมีประสิทธิภาพและประสิทธิผลสูงสุด) เพื่อเป็นการรับประกันว่าผู้เข้าชมเว็บไซต์สามารถได้รับข้อมูลได้อย่างมีคุณภาพ รวดเร็ว ภายใต้เงื่อนไขต่าง ๆ ที่ได้กำหนดเอาไว้

3) หา Maximize ROI โดยที่เซิร์ฟเวอร์ที่มีประสิทธิภาพดี มักจะมีราคาที่สูง หากใช้ในการรองรับผู้เข้าชม 10,000 คน ต่อ หนึ่งนาที ต้องแน่ใจว่าสิ่งที่ลงทุนจะคุ้มค่า

4) จุดเด่น ของโปรแกรม Webserver Stress Tool

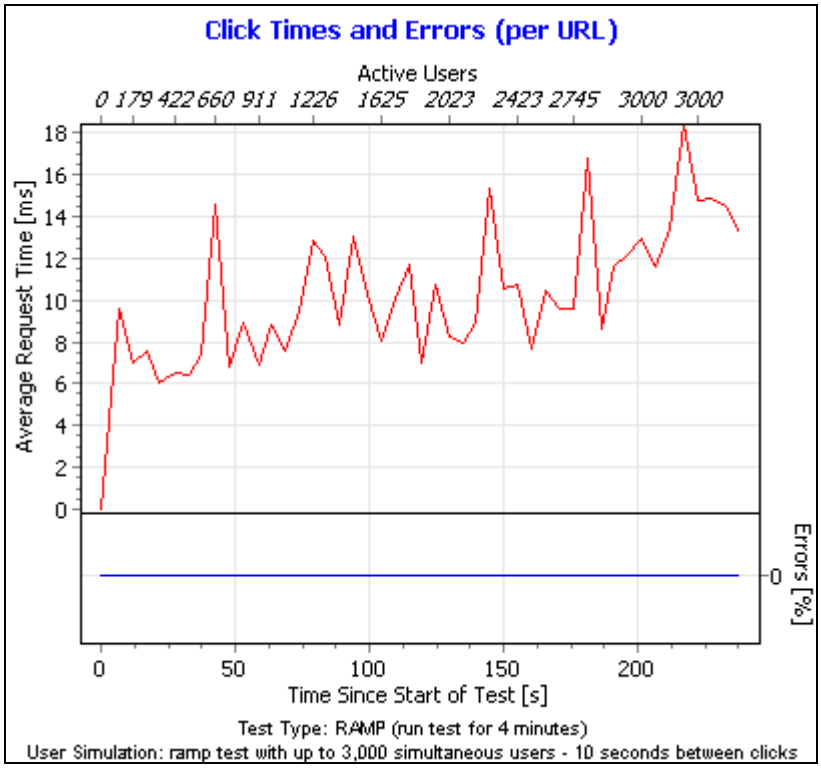
- สามารถรองรับโหลดบนเครือข่ายได้ 500 MBit/s
- สามารถแสดงผลหน้าเว็บเพจได้ 1.000.000 หน้าต่อชั่วโมง
- ผู้ใช้สามารถใช้งานได้ 10.000 คนได้ในเวลาเดียวกัน
- ออกรายงานเป็น html, word

ตัวอย่างในการใช้โปรแกรม Webserver Stress Tool ในการทดสอบ Apache บน Windows 2003 และ Linux ด้วยการเปิดสคริปท์ phpinfo.php โดยการจำลอง user ตั้งแต่ 1 ถึง 400 คน ด้วยการสั่งสคริปท์ทุก ๆ 5 วินาที ผลเป็นดังนี้



รูปที่ 3.11 การทดสอบ Apache บน Windows 2003 และ SUTinsServer 5103 Plus Moodle

ตัวอย่างการเข้าชมเว็บไซต์จากผู้เข้าชม 3,000 คน ซึ่งเข้ามาเยี่ยมชมทุก 10 วินาที ได้ผลดังนี้



รูปที่ 3.12 กราฟแสดงเวลาเฉลี่ยในการตอบสนองต่อผู้เข้าชมเว็บไซต์

ซึ่งจากเหตุผลดังกล่าว จึงทำให้ผู้วิจัยได้เลือกใช้เว็บเซิร์ฟเวอร์ที่เป็นระบบปฏิบัติการลินุกซ์ เพราะว่าสามารถตอบสนองต่อผู้ใช้ได้เร็วขึ้น นอกจากนั้นแล้วยังเป็นโอเพนซอร์สซึ่งสามารถพัฒนาต่อได้โดยไม่มีปัญหาใด ๆ โดยระบบปฏิบัติการลินุกซ์ที่นำมาใช้เป็นเซิร์ฟเวอร์คือ ลินุกซ์เซิร์ฟเวอร์พร้อมใช้สำหรับนักพัฒนาจากมหาวิทยาลัยเทคโนโลยีสุรนารี มีความสามารถต่าง ๆ ดังนี้

- มาพร้อมกับความเสถียรภาพและปลอดจากการรบกวนของไวรัส
- เป็นเวอร์ชัน 2 ภาษา โดยเริ่มตั้งแต่การลงระบบ ถ้าเลือกเมนูการลงระบบด้วยภาษาใด (ไทยหรืออังกฤษ) ก็จะได้ระบบการใช้งานของภาษานั้น
- ขั้นตอนการติดตั้งสามารถใช้เมนูภาษาไทย มีโปรแกรมแบ่งพาร์ทิชันไว้ให้ (ใช้เวลาติดตั้งไม่นานก็จะสามารถมีลินุกซ์เซิร์ฟเวอร์พร้อมใช้งานได้)
- สามารถ อ่าน-เขียน ไดร์ฟของวินโดวส์ได้ทันที ทั้งไดร์ฟที่เป็น FAT32 และ NTFS
- สามารถ อ่าน-เขียน แฟรชไดร์ฟ ซีดีรอม และ โอนถ่ายรูปจากกล้องดิจิตอล
- เหมาะสำหรับการเซ็ตอัพให้เป็นเครื่องเซิร์ฟเวอร์ต่างๆ เช่น Web (PHP และ MySQL),

DNS, DHCP, FTP, SSH, Proxy และ Webmail เป็นต้น

- มี Remote admin ที่ใช้งานง่าย

- มีเครื่องมือช่วยสร้างเว็บหลาย ๆ เว็บ คือเป็น Multi-web ที่แต่ละเว็บมีเนื้อที่เป็นของตนเอง และสามารถส่งเนื้อหาขึ้นเว็บด้วยวิธี FTP เหมือนเว็บที่ให้เช่าเนื้อที่อื่น ๆ

- มีเครื่องมือต่าง ๆ ให้ เช่น สามารถดูหนังฟังเพลง เล่น CD/DVD มีอีดิเตอร์สำหรับการพัฒนาหน้าเพจ มี PHPMyadmin สำหรับจัดการฐานข้อมูล มีโปรแกรมเขียน CD/DVD มีโปรแกรมแก้ไขตกแต่งรูปภาพเป็นต้น

- มีเครื่องมือช่วยให้ทำลินุกซ์เซิร์ฟเวอร์ที่เซ็ตอัพ เปลี่ยนแปลงโปรแกรมและรูปแบบไว้เรียบร้อยแล้ว ให้เป็น Distribution ของตัวเอง และเมื่อเขียนสิ่งที่ได้ลงบนแผ่นซีดีแล้ว จะได้เป็นเซิร์ฟเวอร์เคลื่อนที่พร้อมใช้เป็นของตัวเอง และเมื่อนำไปติดตั้งลงเครื่องใด ๆ ก็จะได้เซิร์ฟเวอร์พร้อมใช้งานของตัวเองทันที

- มีคู่มือการเซ็ตอัพต่าง ๆ เช่น การเซ็ตอัพและการใช้งาน SUTinsServer บนโปรแกรม Vmware, การเซ็ตอัพ SSH, การเซ็ตอัพ VSFTP Server, การเซ็ตอัพ DHCP Server, การเซ็ตอัพ DNS Server, การเซ็ตอัพ Vhost ใน Apache, การเซ็ตอัพโปรแกรมเมล์ Postfix/Dovecot, การเซ็ตอัพ Webmail, การใช้งานเป็น Remote Admin, การทำแผ่นลินุกซ์เป็นของตัวเอง และ การเซ็ตอัพเป็น Proxy Server ไว้เรียบร้อยแล้ว

บทที่ 4
ผลการวิเคราะห์ข้อมูลและอภิปรายผล

การพัฒนาระบบการค้นคืนข้อมูลโดยใช้เทคนิคการ Caching ข้อมูลของเอาท์พุตในครั้งนี้ ได้แบ่งการทดลองออกเป็นสองส่วนคือ การทดสอบประสิทธิภาพการค้นคืนข้อมูลจากระบบแบบเดิม และการทดสอบประสิทธิภาพการค้นคืนข้อมูลที่ได้นำอัลกอริธึมที่พัฒนาขึ้นมาพัฒนาร่วม โดยในการทดลองแต่ละส่วนจะทำการศึกษาคุณสมบัติในการค้นคืนข้อมูลจากระบบของอัลกอริธึมนั้น ๆ ด้วย เกณฑ์ในการพิจารณาความเหมาะสมของแต่ละเทคนิคประกอบด้วย เวลาที่ใช้ในโปรโตคอล (Protocol Times) ค่าของแบนด์วิดท์ (Bandwidth) ค่าของการส่งข้อมูล (Transferred Data) เวลาที่ใช้ในระบบของหน่วยความจำและหน่วยประมวลผลกลาง (System Memory and CPU Load) ตลอดจนเวลาที่ใช้ในการร้องขอข้อมูลของผู้ใช้ (Click Time) และเวลาที่ระบบทำการตอบสนองต่อผู้ใช้ (Hits rate) ในการทดลองนี้จะใช้อัลกอริธึม Caching Output ที่มีอยู่ในโปรแกรมภาษา PHP

สำหรับเนื้อหาที่ได้นำเสนอในบทนี้ จะเป็นการนำเสนอการทดสอบระบบการค้นคืนข้อมูลที่มีประสิทธิภาพโดยมีรายละเอียดดังนี้ หัวข้อ 4.1 อธิบายถึงสภาพแวดล้อมที่ใช้ในการพัฒนาระบบการค้นคืนข้อมูล หัวข้อ 4.2 อธิบายถึงผลการทดสอบระบบแบบเดิม กับการทดสอบระบบแบบที่ใช้ระบบ CO-LMS มาพัฒนาร่วม และหัวข้อ 4.3 เป็นการอภิปรายสรุปผลการเปรียบเทียบการทดสอบระบบทั้งสองระบบ

4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาระบบการค้นคืนข้อมูล

เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนามีรายละเอียดดังนี้

เครื่องให้บริการ (Server)

ฮาร์ดแวร์

- คอมพิวเตอร์แบบพีซี Pentium IV 3.2 กิกะเฮิร์ต
- หน่วยความจำ 512 เมกกะไบต์
- ฮาร์ดดิสก์ 80 กิกะไบต์

ซอฟต์แวร์

- ระบบปฏิบัติการลินุกซ์ SUTinsServer 5103 Plus Moodle

- ระบบ LMS ซึ่งได้ใช้ Moodle เป็นระบบในการทดสอบ

เครื่องรับบริการ (Client)

ฮาร์ดแวร์

- คอมพิวเตอร์แบบพีซี Pentium IV 3.2 กิกะเฮิร์ต
- หน่วยความจำ 512 เมกกะไบต์
- ฮาร์ดดิสก์ 80 กิกะไบต์

ซอฟต์แวร์

- ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ เอ็กซ์พี
- โปรแกรมทดสอบอัตราในการค้นคืนข้อมูล Webserver Stress Tool

4.2 ผลการทดสอบระบบ

การทดสอบระบบการค้นคืนข้อมูลอย่างมีประสิทธิภาพนั้น จะทำการทดสอบระบบการค้นคืนข้อมูลแบบเดิม กับระบบการค้นคืนข้อมูลแบบที่ใช้ระบบ CO-LMS มาพัฒนาร่วม โดยที่ในระบบแบบเดิมนั้นจะใช้การทดสอบกับระบบที่ใช้งานในการทำระบบเว็บเซิร์ฟเวอร์ทั่วไปซึ่งไม่ได้มีการทำเป็น Proxy เซิร์ฟเวอร์ และระบบ CO-LMS มาพัฒนาร่วมกับระบบแบบเดิมเช่นกัน สำหรับ URLs ที่ใช้ในการทดสอบระบบ คือ http://www.co-lms.net/elearning/

4.2.1 การทดสอบระบบค้นคืนข้อมูลแบบเดิม

ตารางที่ 4.1 ถึง 4.6 แสดงผลการทดสอบระบบการค้นคืนแบบเดิม โดยจะแสดงค่าเวลาที่ใช้สำหรับโปรโตคอล (Protocol Times) ค่าของแบนด์วิดท์ (Bandwidth) ค่าของการส่งข้อมูล (Transferred Data) ค่าเวลาที่ใช้ในระบบของหน่วยความจำและหน่วยประมวลผลกลาง (System Memory and CPU Load) ตลอดจนเวลาที่ใช้ในการร้องขอข้อมูลของผู้ใช้ (Click Time) และเวลาที่ระบบทำการตอบสนองต่อผู้ใช้ (Hits Rate) เป็นต้น

1) ค่าเวลาที่ใช้สำหรับโปรโตคอล (Protocol Times) จะเป็นค่าเวลาที่ระบบใช้ในการส่งข้อมูลสำหรับการโอนถ่าย (Transmission) ข้อมูลหรือการแลกเปลี่ยนข้อมูลระหว่างต้นทางกับปลายทาง โดยข้อมูลที่ได้จากการทดลองนี้เป็นค่าเวลาที่ทำงานอยู่ระหว่างโปรโตคอล ซึ่งจะได้ค่าเวลาที่ผู้ใช้ใช้ในการร้องขอ (Click Time), เวลาที่ใช้ส่งข้อมูลไบต์แรก (Time to First Byte), เวลาที่ใช้ในการเชื่อมต่อ (Time to Connect), เวลาสำหรับ DNS (Time for DNS) และเวลาสำหรับ Local Socket (Time for Local Socket)

ตารางที่ 4.1 แสดงค่า Protocol Times for all URLs แบบเดิม

Time [ms]	Click Time [s]	Time to First Byte [s]	Time to Connect [s]	Time for DNS [s]	Time for local socket [s]
0	0	0	0	0	0
5.84	4868.43	1688.71	53.71	18.00	5.78
11.01	4497.25	754.17	39.50	12.75	0.57
16.12	4694.64	664.43	45.07	15.50	0.66
21.30	0.00	0.00	0.00	0.00	1.97
26.70	8643.08	1326.58	160.75	69.58	3.90
31.97	4362.43	544.00	50.43	18.00	0.65
37.33	5010.75	746.75	176.55	79.70	1.22
42.51	5356.94	902.00	217.50	98.50	0.72
47.73	3773.00	466.54	40.31	10.77	0.71
52.92	4136.08	509.33	48.83	14.92	0.51
58.11	5550.40	688.90	37.10	10.30	0.84
63.34	4520.35	637.29	94.59	36.82	1.49
68.56	7247.75	917.25	45.33	14.50	1.06
74.03	3807.22	577.22	115.94	48.00	1.31
79.20	4658.67	645.56	69.44	26.67	1.47
84.53	8895.23	1441.85	262.85	110.69	0.74
89.94	9283.55	2524.85	136.80	59.90	0.70
95.09	2720.86	335.86	38.57	10.86	0.99
100.30	6025.42	878.05	141.95	59.21	0.82
105.48	6192.20	731.40	39.20	9.80	2.65
111.09	5450.10	970.33	276.05	117.33	0.86
116.34	5300.79	686.00	69.21	24.36	0.72
121.66	6124.44	875.89	106.44	41.00	1.09
126.91	4934.42	1760.89	115.95	55.05	0.62
132.16	4074.92	556.92	60.38	19.08	0.69

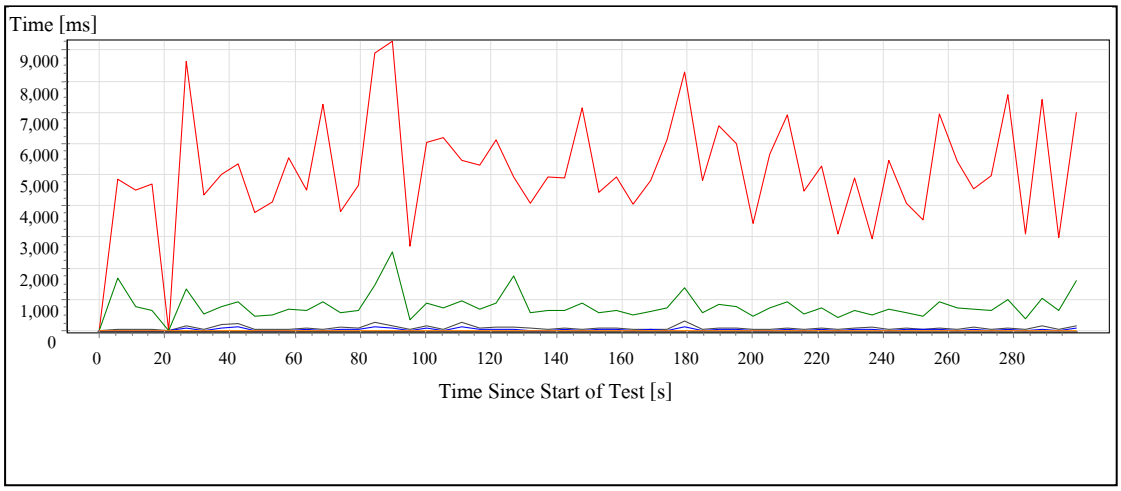
ตารางที่ 4.1 แสดงค่า Protocol Times for all URLs แบบเดิม (ต่อ)

Time [ms]	Click Time [s]	Time to First Byte [s]	Time to Connect [s]	Time for DNS [s]	Time for local socket [s]
142.59	4907.19	644.94	67.38	24.00	0.53
147.81	7159.00	866.64	40.09	11.73	0.69
153.06	4420.13	585.56	65.13	20.94	1.14
158.26	4931.33	656.61	80.11	29.67	0.81
163.45	4044.00	496.38	39.75	12.38	0.74
168.69	4804.28	611.39	54.72	19.39	1.90
173.87	6106.00	717.17	37.17	10.67	4.25
179.33	8277.15	1383.75	287.35	110.25	0.71
184.51	4828.88	589.63	48.38	16.63	0.83
189.76	6586.33	850.42	70.08	24.33	0.90
195.01	5984.89	758.83	81.78	27.22	0.56
200.22	3432.00	453.83	47.25	14.33	0.68
205.42	5661.23	743.69	54.54	17.85	0.91
210.61	6900.13	914.94	94.88	32.69	0.59
215.78	4468.36	543.45	46.91	15.27	1.29
221.03	5256.12	713.24	67.29	24.88	0.84
226.20	3105.45	427.73	52.64	15.00	1.07
231.39	4886.00	640.72	90.33	33.94	0.99
236.59	2944.08	479.67	103.67	39.67	0.55
241.78	5465.55	678.82	46.00	15.00	0.85
246.98	4093.24	582.65	84.88	29.82	0.71
252.17	3549.67	476.67	55.67	20.00	0.96
257.41	6956.56	922.06	74.89	28.28	0.52
262.59	5416.33	713.78	39.67	11.33	2.42
267.84	4545.00	693.13	120.06	46.44	0.83
273.00	4982.27	651.27	51.36	15.55	0.81
278.26	7564.68	993.05	78.95	30.58	0.54

ตารางที่ 4.1 แสดงค่า Protocol Times for all URLs แบบเดิม (ต่อ)

Time [ms]	Click Time [s]	Time to First Byte [s]	Time to Connect [s]	Time for DNS [s]	Time for local socket [s]
283.50	3089.14	382.71	45.57	13.43	0.98
288.69	7412.75	1041.40	140.10	55.90	0.56
293.89	2964.50	646.25	40.25	14.75	2.04
299.14	6996.28	1590.11	144.06	65.33	0.83

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์ สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.1



รูปที่ 4.1 กราฟแสดงค่าของ Protocol Times for all URLs

2) ค่าแบนด์วิดท์เฉลี่ยของผู้ใช้ และค่าแบนด์วิดท์ของเซิร์ฟเวอร์ (Average User and Server Bandwidth) โดยค่าแบนด์วิดท์เฉลี่ยของผู้ใช้จะเป็นค่าของแบนด์วิดท์ในขณะที่ผู้ใช้ทำการส่งและรับข้อมูลทั้งหมดไปที่เซิร์ฟเวอร์ ส่วนค่าแบนด์วิดท์ของเซิร์ฟเวอร์จะเป็นค่าของแบนด์วิดท์ที่เซิร์ฟเวอร์ทำการส่งและรับข้อมูลไปที่ผู้ใช้

ตารางที่ 4.2 แสดงค่า Average User และ Server Bandwidth แบบเดิม

Time [ms]	Avg. User Bandwidth [kbit/s]	Server Bandwidth [kbit/s]
5.84	35.49	238.11

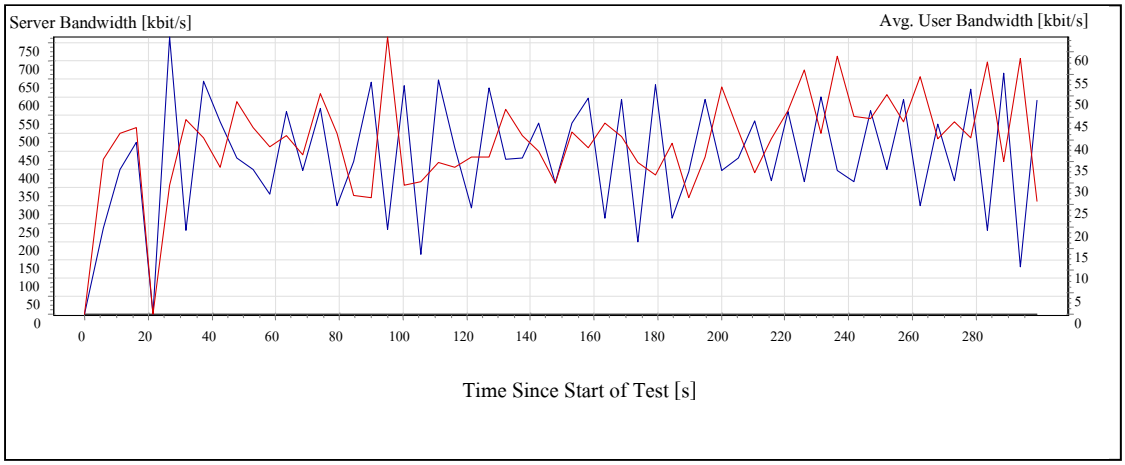
ตารางที่ 4.2 แสดงค่า Average User และ Server Bandwidth แบบเดิม (ต่อ)

Time [ms]	Avg. User Bandwidth [kbit/s]	Server Bandwidth [kbit/s]
11.01	41.51	400.87
16.12	42.65	473.67
21.30	0.00	0.00
26.70	29.43	765.24
31.97	44.52	229.92
37.33	40.35	643.67
42.51	33.61	531.96
47.73	48.75	430.28
52.92	42.75	399.27
58.11	38.45	332.23
63.34	40.98	560.57
68.56	36.47	397.17
74.03	50.39	568.84
79.20	41.50	299.61
84.53	27.22	421.89
89.94	26.74	638.92
95.09	63.44	234.16
100.30	29.43	630.76
105.48	30.32	166.12
111.09	34.68	645.73
116.34	33.66	460.44
121.66	35.90	292.30
126.91	35.98	625.12
132.16	46.76	427.71
137.34	40.94	432.22
142.59	37.37	526.80
147.81	30.00	362.95
153.06	41.73	527.27

ตารางที่ 4.2 แสดงค่า Average User และ Server Bandwidth แบบเดิม (ต่อ)

Time [ms]	Avg. User Bandwidth [kbit/s]	Server Bandwidth [kbit/s]
158.26	38.06	596.66
163.45	43.78	266.18
168.69	40.64	593.99
173.87	34.72	199.36
179.33	31.87	633.39
184.51	39.03	265.99
189.76	26.71	394.23
195.01	36.09	592.65
200.22	51.96	398.07
205.42	42.05	430.60
210.61	32.43	532.76
215.78	40.18	367.66
221.03	46.69	558.85
226.20	56.02	366.59
231.39	41.38	599.35
236.59	59.08	397.48
241.78	45.32	366.55
246.98	44.92	563.09
252.17	50.32	399.27
257.41	44.11	594.46
262.59	54.25	299.67
267.84	40.25	526.03
273.00	43.91	367.38
278.26	40.45	622.59
283.50	57.62	231.42
288.69	35.00	665.45
293.89	58.54	132.49
299.14	25.92	591.31

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์ สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.2



รูปที่ 4.2 กราฟแสดงค่าของ Average User and Server Bandwidth

3) ค่าของการส่งข้อมูล (Transferred Data), หน่วยความจำที่ใช้ในระบบ (System Memory) และโหลดที่ใช้ในหน่วยประมวลผลกลาง (CPU Load) โดยค่าของการส่งข้อมูลคือปริมาณของข้อมูลที่ผู้ใช้ได้ส่งและรับข้อมูลไปที่เซิร์ฟเวอร์ ซึ่งค่าที่ผู้ใช้ส่งข้อมูลไปที่เซิร์ฟเวอร์ เช่น การบรรจุข้อมูลขึ้นไปในระบบ FTP หรือการบรรจุข้อมูลของเว็บต่าง ๆ ผ่านเว็บเบราว์เซอร์เป็นต้น ส่วนการรับข้อมูลจากเซิร์ฟเวอร์นั้น เช่นการเรียกใช้งานของผู้ใช้หรือการเปิดเว็บเบราว์เซอร์ เป็นต้น

ตารางที่ 4.3 แสดงค่า Transferred Data System Memory และ CPU Load แบบเดิม

Time [ms]	Transferred Data [kbit/s]	Available System Memory [MB]	Local CPU Load [%]
5.96	238.11	148.00	15.65
11.05	400.87	148.00	22.68
16.21	473.67	147.00	33.69
21.54	0.00	147.00	33.59
26.83	765.24	146.00	32.45
32.23	229.92	145.00	28.39
37.44	643.67	144.00	22.45

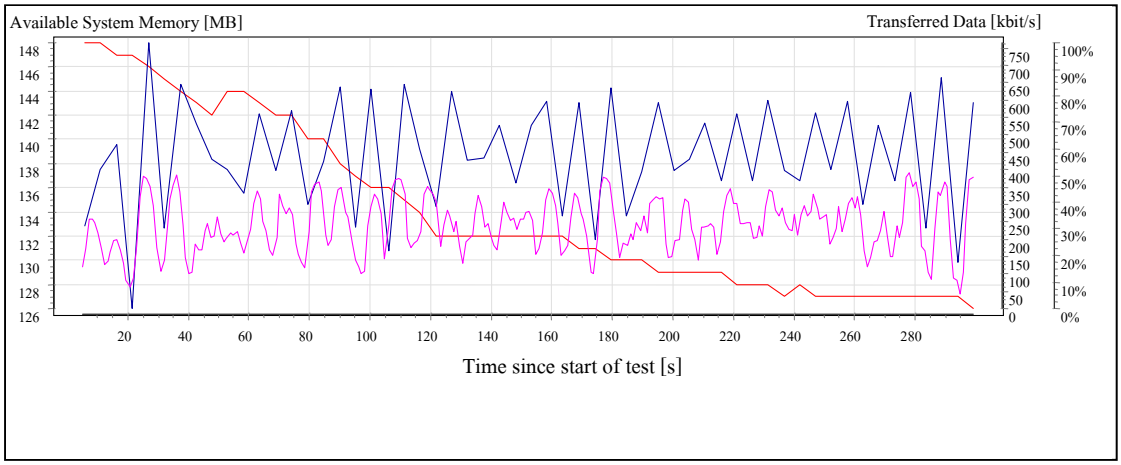
ตารางที่ 4.3 แสดงค่า Transferred Data System Memory และ CPU Load แบบเดิม (ต่อ)

Time [ms]	Transferred Data [kbit/s]	Available System Memory [MB]	Local CPU Load [%]
42.63	531.96	143.00	16.72
47.84	430.28	142.00	17.67
58.24	332.23	144.00	25.71
63.46	560.57	143.00	25.80
68.91	397.17	142.00	22.01
74.14	568.84	142.00	16.96
79.47	299.61	140.00	10.71
84.71	421.89	140.00	7.98
90.05	638.92	138.00	11.31
95.21	234.16	137.00	23.37
100.43	630.76	136.00	42.24
106.02	166.12	136.00	49.75
111.30	645.73	135.00	48.82
116.57	460.44	134.00	45.99
121.78	292.30	132.00	38.83
127.07	625.12	132.00	22.84
132.28	427.71	132.00	14.23
137.50	432.22	132.00	18.26
142.72	526.80	132.00	31.64
148.01	362.95	132.00	41.43
153.18	527.27	132.00	46.28
158.40	596.66	132.00	50.23
163.59	266.18	132.00	43.67
168.81	593.99	131.00	31.65
174.27	199.36	131.00	19.25
179.47	633.39	130.00	13.17
184.69	265.99	130.00	13.53

ตารางที่ 4.3 แสดงค่า Transferred Data System Memory และ CPU Load แบบเดิม (ต่อ)

Time [ms]	Transferred Data [kbit/s]	Available System Memory [MB]	Local CPU Load [%]
189.90	394.23	130.00	24.08
195.16	592.65	129.00	22.30
200.38	398.07	129.00	22.16
205.56	430.60	129.00	29.29
210.74	532.76	129.00	31.96
215.92	367.66	129.00	26.70
221.16	558.85	128.00	27.42
226.34	366.59	128.00	34.32
231.53	599.35	128.00	28.21
236.73	397.48	127.00	24.91
241.95	366.55	128.00	26.48
247.15	563.09	127.00	28.57
252.33	399.27	127.00	27.84
257.55	594.46	127.00	28.96
262.72	299.67	127.00	24.47
267.98	526.03	127.00	20.98
273.21	367.38	127.00	25.69
278.46	622.59	127.00	29.69
283.63	231.42	127.00	39.63
288.84	665.45	127.00	44.11
294.03	132.49	127.00	41.41
299.28	591.31	126.00	32.87

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์ สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.3



รูปที่ 4.3 กราฟแสดงค่าของ Transferred Data และ System Memory และ CPU Load

4) ค่าของ Open Requests และ Transferred Data จากการทดลองนี้จะแสดงค่าแบนด์วิดท์ที่ใช้ (Bandwidth), ค่า Open Requests, ค่าเวลาที่ใช้ในการรับข้อมูล (Received Requests) และค่าเวลาที่ใช้ในการส่งข้อมูล (Sent Requests)

ตารางที่ 4.4 แสดงค่า Open Requests และ Transferred Data แบบเดิม

Time [ms]	Bandwidth [kbit/s]	Open Requests	Received Requests/s	Sent Requests/s
5.96	238.11	80.00	13.95	15.52
11.05	400.87	24.00	21.31	19.38
16.21	473.67	41.00	5.26	9.20
21.54	0.00	55.00	31.21	28.37
26.83	765.24	81.00	14.75	17.51
32.23	229.92	43.00	26.04	22.76
37.44	643.67	69.00	24.66	23.50
42.63	531.96	82.00	19.94	19.94
47.84	430.28	24.00	18.65	19.04

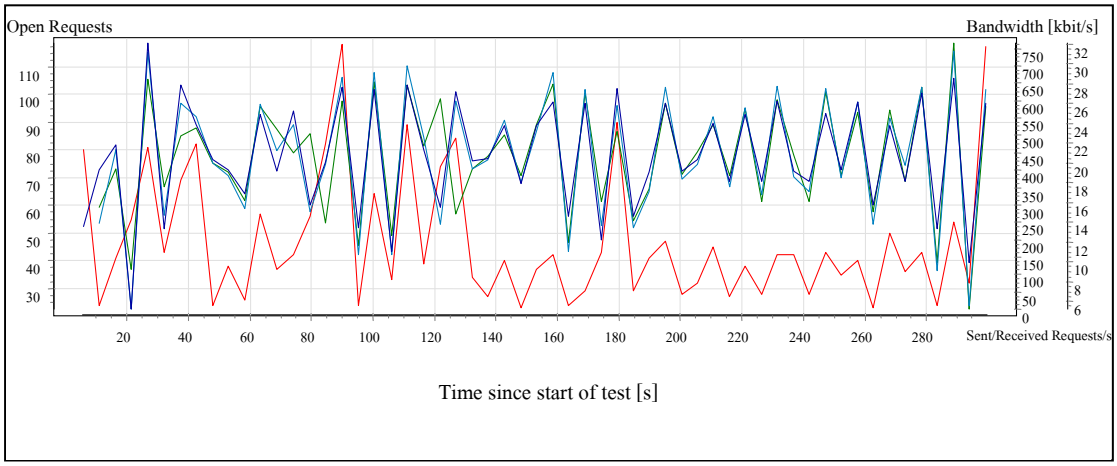
ตารางที่ 4.4 แสดงค่า Open Requests และ Transferred Data แบบเดิม (ต่อ)

Time [ms]	Bandwidth [kbit/s]	Open Requests	Received Requests/s	Sent Requests/s
53.04	399.27	38.00	15.39	16.16
58.24	332.23	26.00	25.86	25.67
63.46	560.57	57.00	21.16	23.35
74.14	568.84	42.00	15.02	22.91
79.47	299.61	56.00	19.82	13.91
84.71	421.89	82.00	28.66	26.22
90.05	638.92	118.00	10.66	11.63
95.21	234.16	24.00	29.12	28.16
100.43	630.76	64.00	10.72	12.69
106.02	166.12	33.00	29.77	27.87
111.30	645.73	89.00	22.61	21.66
116.57	460.44	39.00	13.81	26.46
121.78	292.30	74.00	26.29	14.75
127.07	625.12	84.00	19.39	19.39
132.28	427.71	34.00	20.31	20.69
137.50	432.22	27.00	24.32	22.78
142.72	526.80	40.00	17.96	18.72
148.01	362.95	23.00	23.21	23.80
153.18	527.27	37.00	29.12	27.97
158.40	596.66	42.00	10.98	11.95
163.59	266.18	24.00	27.39	27.01
168.81	593.99	29.00	13.71	16.09
174.27	199.36	43.00	25.82	23.12
179.47	633.39	90.00	13.39	14.16
184.69	265.99	29.00	17.08	17.27
189.90	394.23	41.00	27.59	26.06
195.16	592.65	47.00	18.40	18.79

ตารางที่ 4.4 แสดงค่า Open Requests และ Transferred Data แบบเดิม (ต่อ)

Time [ms]	Bandwidth [kbit/s]	Open Requests	Received Requests/s	Sent Requests/s
200.38	398.07	28.00	19.89	21.05
205.56	430.60	32.00	24.67	23.90
210.74	532.76	45.00	17.58	18.74
215.92	367.66	27.00	25.58	25.39
226.34	366.59	28.00	27.74	26.39
231.53	599.35	42.00	18.63	20.74
236.73	397.48	42.00	17.06	16.10
241.95	366.55	28.00	27.52	27.13
247.15	563.09	43.00	18.50	18.69
252.33	399.27	35.00	26.08	25.12
257.55	594.46	40.00	13.73	15.08
262.72	299.67	23.00	24.55	25.32
267.98	526.03	50.00	19.69	18.16
273.21	367.38	36.00	27.63	27.63
278.46	622.59	43.00	9.08	9.85
283.63	231.42	24.00	31.31	32.08
288.84	665.45	54.00	5.59	5.20
294.03	132.49	32.00	27.40	25.69
299.28	591.31	117.00	27.63	25.32

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์ สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.4



รูปที่ 4.4 กราฟแสดงค่าของ Open Requests และ Transferred Data

5) ค่าเวลาที่ใช้ในการร้องขอข้อมูล (Click Time) และค่าเวลาที่ระบบทำการตอบสนอง (Hits/s) ต่อผู้ใช้ จากการทดลองนี้จะได้ค่าความสัมพันธ์ต่าง ๆ ระหว่างที่ผู้ใช้ได้ทำการร้องขอข้อมูลมายังระบบ

ตารางที่ 4.5 แสดงค่า Click Time User/s และ Hits/s for all URLs แบบเดิม

Time [ms]	Click Time [ms]	Clicks/s	Hits/s
5.84	4868.43	1.38	11.04
11.02	4497.25	2.32	18.58
16.13	4694.64	2.74	21.95
21.30	0.00	0.00	0.00
26.70	8643.08	4.43	35.47
31.97	4362.43	1.33	10.66
37.33	5010.75	3.73	29.83
42.52	5356.94	3.08	24.65
47.73	3773.00	2.49	19.94
52.92	4136.08	2.31	18.50

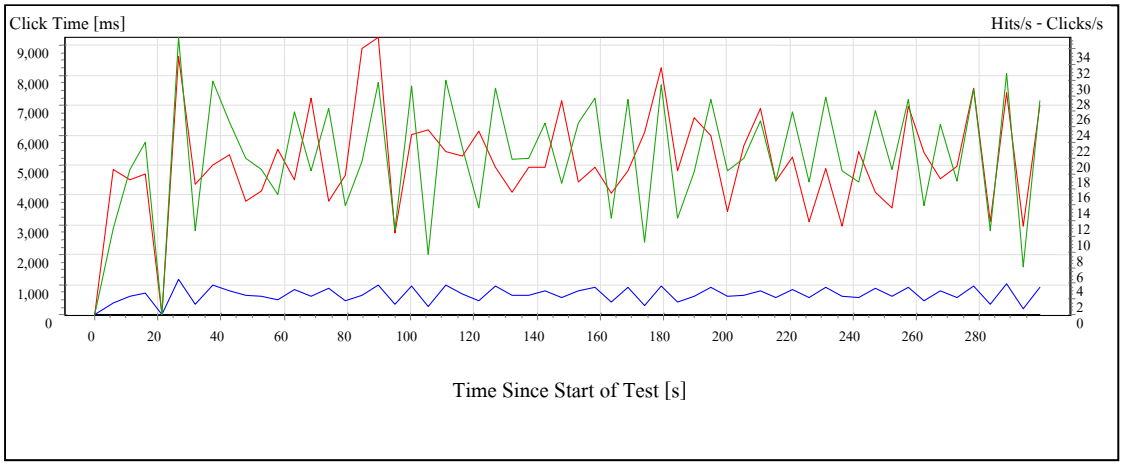
ตารางที่ 4.5 แสดงค่า Click Time User/s และ Hits/s for all URLs แบบเดิม (ต่อ)

Time [ms]	Click Time [ms]	Clicks/s	Hits/s
58.11	5550.40	1.92	15.40
63.34	4520.35	3.25	25.98
68.56	7247.75	2.30	18.41
74.03	3807.22	3.30	26.36
79.20	4658.67	1.74	13.89
84.53	8895.23	2.44	19.55
89.94	9283.55	3.70	29.61
95.09	2720.86	1.36	10.85
100.30	6025.42	3.65	29.23
105.48	6192.20	0.96	7.70
111.09	5450.10	3.74	29.93
116.34	5300.79	2.67	21.34
121.66	6124.44	1.69	13.55
126.91	4934.42	3.62	28.97
132.16	4074.92	2.48	19.82
137.34	4917.62	2.50	20.03
142.59	4907.19	3.05	24.42
147.81	7159.00	2.10	16.82
153.06	4420.13	3.05	24.44
158.27	4931.33	3.46	27.65
163.45	4044.00	1.54	12.34
168.69	4804.28	3.44	27.53
173.88	6106.00	1.15	9.24
179.33	8277.15	3.67	29.35
184.52	4828.88	1.54	12.33
189.77	6586.33	2.28	18.27
195.02	5984.89	3.43	27.47
200.22	3432.00	2.31	18.45

ตารางที่ 4.5 แสดงค่า Click Time User/s และ Hits/s for all URLs แบบเดิม (ต่อ)

Time [ms]	Click Time [ms]	Clicks/s	Hits/s
205.42	5661.23	2.49	19.96
210.61	6900.13	3.09	24.69
215.78	4468.36	2.13	17.04
221.03	5256.12	3.24	25.90
226.20	3105.46	2.12	16.99
231.39	4886.00	3.47	27.78
236.59	2944.08	2.30	18.42
241.78	5465.55	2.12	16.99
246.98	4093.24	3.26	26.10
252.17	3549.67	2.31	18.50
257.41	6956.56	3.44	27.55
262.59	5416.33	1.74	13.89
267.84	4545.00	3.05	24.38
273.00	4982.27	2.13	17.03
278.27	7564.68	3.61	28.85
283.50	3089.14	1.34	10.73
288.69	7412.75	3.86	30.84
293.89	2964.50	0.77	6.14
299.14	6996.28	3.43	27.40

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์ สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.5



รูปที่ 4.5 กราฟแสดงค่าของ Click Time, Hits/s และ User/s for all URLs

6) ค่าเฉลี่ยของเวลาที่ได้จากการร้องขอข้อมูลจากระบบ (Average Request Time) และค่าความผิดพลาดเมื่อทำการร้องขอข้อมูลจากระบบ (Errors)

ตารางที่ 4.6 แสดงค่า Average Request Time และ Errors (per URL) แบบเดิม

Time [ms]	Average Request Time [ms]	Average Request Time [ms] All Images	Errors [%]	
5.84	4868.36	1525.43	0	0%
11.02	4497.30	404.65	0	0%
16.13	4694.63	355.39	0	0%
21.30	0.00	93.75	0	0%
31.97	4362.41	281.09	0	0%
37.33	5010.85	507.66	0	0%
42.52	5356.94	611.42	0	0%
47.73	3773.12	120.36	0	0%
52.92	4136.17	178.76	0	0%
58.11	5550.52	146.60	0	0%

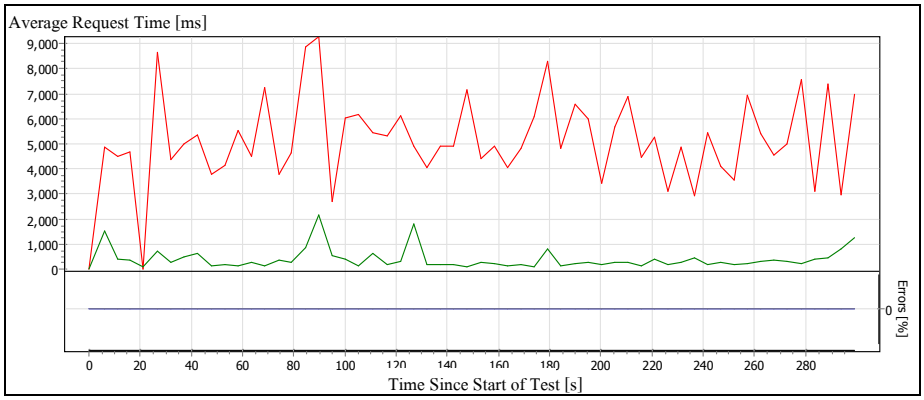
ตารางที่ 4.6 แสดงค่า Average Request Time และ Errors (per URL) แบบเดิม (ต่อ)

Time [ms]	Average Request Time [ms]	Average Request Time [ms] All Images	Errors [%]	
63.34	4520.27	289.17	0	0%
68.56	7247.79	125.45	0	0%
74.03	3807.29	364.96	0	0%
79.20	4658.60	282.35	0	0%
84.53	8895.21	859.00	0	0%
89.94	9283.53	2142.07	0	0%
95.09	2721.05	526.75	0	0%
100.30	6025.33	426.97	0	0%
105.48	6192.46	118.32	0	0%
111.09	5450.18	650.47	0	0%
116.34	5300.80	193.66	0	0%
121.66	6124.36	337.90	0	0%
126.91	4934.38	1809.80	0	0%
132.16	4075.00	158.55	0	0%
137.34	4917.60	191.66	0	0%
142.59	4907.17	177.63	0	0%
147.81	7159.01	109.64	0	0%
153.06	4420.21	263.20	0	0%
158.27	4931.19	231.88	0	0%
163.45	4044.12	139.22	0	0%
168.69	4804.25	187.16	0	0%
173.88	6105.84	102.62	0	0%
179.33	8277.20	793.88	0	0%
184.52	4828.76	136.52	0	0%
189.77	6586.35	230.10	0	0%
195.02	5984.88	263.43	0	0%
200.22	3431.87	192.13	0	0%

ตารางที่ 4.6 แสดงค่า Average Request Time และ Errors (per URL) แบบเดิม (ต่อ)

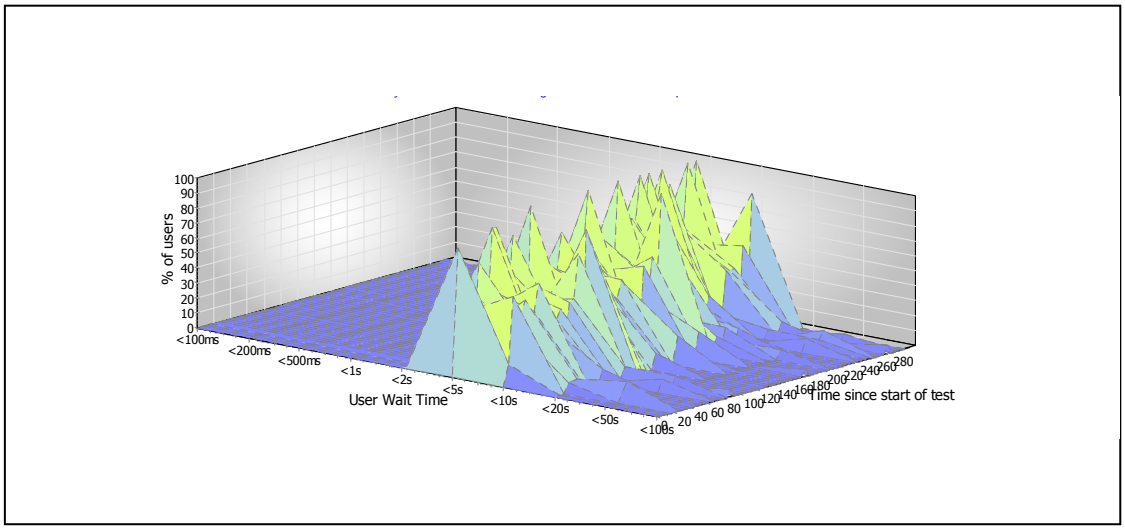
Time [ms]	Average Request Time [ms]	Average Request Time [ms] All Images	Errors [%]	
205.42	5661.06	270.30	0	0%
210.61	6900.01	261.30	0	0%
215.78	4468.37	124.73	0	0%
221.03	5256.16	384.33	0	0%
226.20	3105.38	182.48	0	0%
231.39	4886.06	267.49	0	0%
236.59	2944.19	472.03	0	0%
241.78	5465.51	190.96	0	0%
246.98	4093.14	269.85	0	0%
252.17	3549.85	190.44	0	0%
257.41	6956.57	213.65	0	0%
262.59	5416.50	314.39	0	0%
267.84	4544.97	352.15	0	0%
273.00	4982.23	315.20	0	0%
278.27	7564.72	237.04	0	0%
283.50	3089.15	394.91	0	0%
288.69	7412.82	434.56	0	0%
293.89	2964.38	812.82	0	0%
299.14	6996.17	1256.45	0	0%

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์ สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.6



รูปที่ 4.6 กราฟแสดงค่าของ Average Request Time และ Errors (per URL)

เมื่อนำค่าเวลาที่ผู้ใช้ทำการรอระหว่างการร้องขอข้อมูลจากระบบ (User Wait Time) ค่าเวลาที่ใช้ในการทดสอบ (Time Since Start of Test) และเปอร์เซ็นต์ค่าความผิดพลาดเมื่อผู้ใช้ร้องขอข้อมูลจากระบบ (% of User) มาทำการแสดงผลเป็นแบบ Spectrum จะทำให้สามารถวิเคราะห์ได้ว่า เมื่อผู้ใช้ทำการร้องขอข้อมูลจากระบบแบบเดิมนั้น เวลาที่ผู้ใช้ร้องขอข้อมูลจากระบบเป็นไปกระจัดกระจายและไม่ต่อเนื่อง



รูปที่ 4.7 กราฟแสดงค่า Spectrum of Click Times

4.2.2 การทดสอบระบบการค้นคืนข้อมูลแบบที่ได้นำระบบ CO-LMS มาพัฒนาร่วม

ตารางที่ 4.7 ถึง 4.12 แสดงผลการทดสอบระบบการค้นคืนแบบที่ได้นำอัลกอริธึมการ Caching ข้อมูลของเอาท์พุตมาพัฒนาร่วม โดยจะแสดงค่าเวลาที่ใช้สำหรับโปรโตคอล (Protocol Times) ค่าของแบนด์วิดท์ (Bandwidth) ค่าของการส่งข้อมูล (Transferred Data) ค่าเวลาที่ใช้ในระบบของหน่วยความจำและหน่วยประมวลผลกลาง (System Memory and CPU Load) ตลอดจนเวลาที่ใช้ในการร้องขอข้อมูลของผู้ใช้ (Click Time) และเวลาที่ระบบทำการตอบสนองต่อผู้ใช้ (Hits rate) เป็นต้น

1) ค่าเวลาที่ใช้สำหรับโปรโตคอล (Protocol Times) ซึ่งเป็นค่าเวลาที่ระบบใช้ในการส่งข้อมูลสำหรับการโอนถ่าย (Transmission) ข้อมูลหรือการแลกเปลี่ยนข้อมูลระหว่างต้นทางกับปลายทาง โดยข้อมูลที่ได้จากการทดลองนี้เป็นค่าเวลาที่ทำงานอยู่ระหว่างโปรโตคอล ซึ่งจะได้ค่าเวลาที่ผู้ใช้ใช้ในการร้องขอ (Click Time) เวลาที่ใช้ส่งข้อมูลไบท์แรก (Time to First Byte) เวลาที่ใช้ในการเชื่อมต่อ (Time to Connect) เวลาสำหรับ DNS (Time for DNS) และเวลาสำหรับ Local Socket (Time for Local Socket) โดยแสดงในตารางที่ 4.7

ตารางที่ 4.7 แสดงค่า Protocol Times for all URLs แบบที่ใช้ระบบ CO-LMS

Time [ms]	Click Time [s]	Time to First Byte [s]	Time to Connect [s]	Time for DNS [s]	Time for local socket [s]
0.00	0.00	0.00	0.00	0.00	0.00
5.89	4203.00	148.00	13.52	273.00	706.00
11.28	5448.96	263.11	3.34	546.70	1317.07
16.36	2770.57	46.52	1.63	135.86	296.29
21.95	2858.59	62.18	1.66	163.55	336.45
27.37	2765.78	43.09	1.84	140.09	300.00
32.94	3020.62	73.00	2.87	233.52	475.24
38.37	3471.35	136.90	3.91	347.00	654.35
43.69	2864.14	68.91	1.73	197.18	390.50
49.05	2750.48	40.65	1.20	126.87	282.96
54.31	2818.96	71.22	2.28	196.57	372.74
59.70	2869.65	57.65	2.10	166.12	365.65
65.28	3069.00	78.48	1.94	215.89	461.04

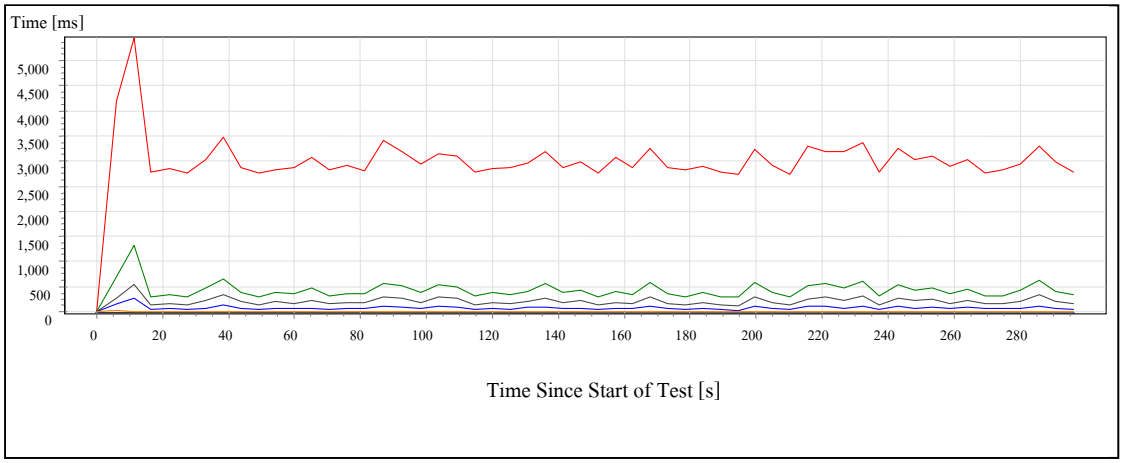
ตารางที่ 4.7 แสดงค่า Protocol Times for all URLs แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Click Time [s]	Time to First Byte [s]	Time to Connect [s]	Time for DNS [s]	Time for local socket [s]
70.64	2821.00	50.78	1.92	145.91	311.00
75.84	2918.16	58.89	1.70	172.21	353.05
81.13	2812.05	68.90	2.94	185.81	350.57
92.64	3179.93	93.18	2.24	276.50	509.36
98.16	2935.00	73.76	2.25	182.53	380.65
103.70	3145.39	104.87	5.02	287.65	529.43
109.45	3093.83	99.46	1.33	259.00	492.00
114.72	2771.43	43.43	1.34	139.38	314.86
119.95	2856.39	66.43	2.66	189.87	373.70
125.22	2875.84	49.53	2.54	148.47	325.42
130.84	2950.79	86.08	2.10	210.21	414.83
136.16	3175.95	86.24	3.73	269.52	566.95
141.39	2872.80	59.40	2.99	178.95	371.60
146.67	2993.17	76.58	2.37	219.54	431.50
151.97	2747.52	42.14	1.34	138.24	301.00
157.38	3070.83	67.56	2.40	185.94	396.94
162.56	2870.04	58.78	1.86	163.26	345.30
167.94	3240.64	113.59	4.26	300.32	577.09
173.33	2876.86	56.86	1.50	165.76	362.38
178.61	2825.68	45.64	2.62	130.55	297.27
183.94	2902.54	68.83	1.97	189.46	375.46
189.25	2773.62	39.76	3.20	124.95	294.10
194.51	2740.67	31.71	1.84	110.81	281.43
199.75	3231.39	109.09	2.89	291.65	574.61
205.08	2905.77	64.73	2.03	185.14	377.32
210.31	2745.58	40.05	1.60	130.21	301.84

ตารางที่ 4.7 แสดงค่า Protocol Times for all URLs แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Click Time [s]	Time to First Byte [s]	Time to Connect [s]	Time for DNS [s]	Time for local socket [s]
215.64	3287.15	108.50	6.37	256.35	512.10
221.03	3190.84	110.63	2.22	291.74	550.74
226.59	3189.00	75.20	3.20	226.75	464.15
232.20	3353.07	120.00	3.02	314.30	595.52
242.98	3257.96	106.57	2.23	275.70	544.39
248.14	3025.24	72.86	1.79	222.62	432.90
253.41	3101.62	93.05	1.76	238.43	465.38
258.70	2885.38	62.57	2.34	166.24	354.71
264.03	3037.68	83.68	3.92	225.53	442.68
269.45	2762.30	56.30	1.72	149.48	311.96
274.94	2816.32	56.82	1.97	160.18	320.09
280.25	2944.16	60.11	3.36	204.63	418.84
285.72	3305.41	122.82	2.58	332.82	636.36
290.91	2986.23	73.77	1.91	196.50	401.09
296.20	2787.65	55.43	1.89	163.26	338.26

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.8



รูปที่ 4.8 กราฟแสดงค่าของ Protocol Times for all URLs

2) ค่าแบนด์วิดท์เฉลี่ยของผู้ใช้ และค่าแบนด์วิดท์ของเซิร์ฟเวอร์ (Average User and Server Bandwidth) โดยค่าแบนด์วิดท์เฉลี่ยของผู้ใช้จะเป็นค่าของแบนด์วิดท์ในขณะที่ผู้ใช้ทำการส่งและรับข้อมูลทั้งหมดไปที่เซิร์ฟเวอร์ ส่วนค่าแบนด์วิดท์ของเซิร์ฟเวอร์จะเป็นค่าของแบนด์วิดท์ที่เซิร์ฟเวอร์ทำการส่งและรับข้อมูลไปที่ผู้ใช้

ตารางที่ 4.8 แสดงค่า Average User และ Server Bandwidth แบบที่ใช้ระบบ CO-LMS

Time [ms]	User Bandwidth [kbit/s]	Server Bandwidth [kbit/s]
5.89	40.35	32.19
11.28	35.34	854.63
16.36	61.26	702.29
21.95	59.63	665.75
27.37	61.38	719.31
32.94	56.50	641.48
38.37	50.38	622.94
43.69	59.42	702.62
49.05	61.71	727.84

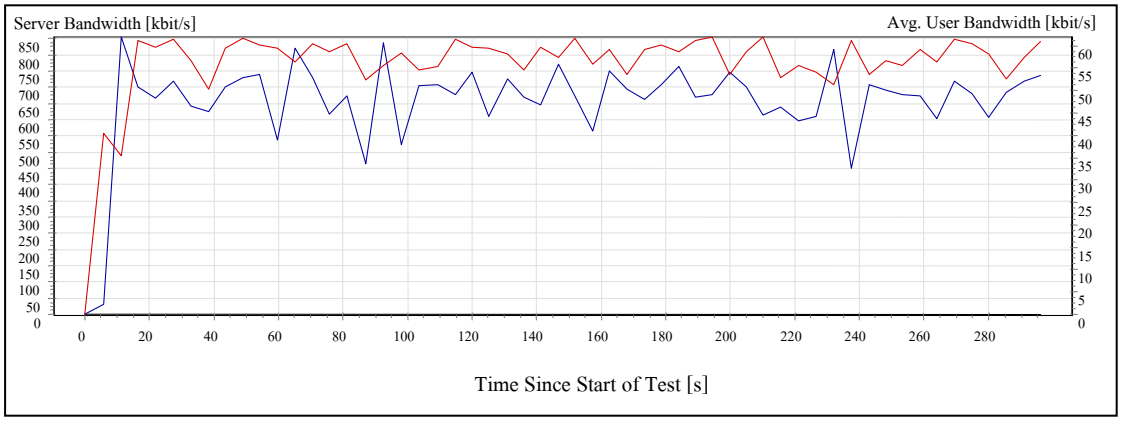
ตารางที่ 4.8 แสดงค่า Average User และ Server Bandwidth แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	User Bandwidth [kbit/s]	Server Bandwidth [kbit/s]
54.31	60.28	739.43
59.70	59.30	535.81
65.28	56.40	819.51
70.64	60.37	729.43
75.84	58.53	618.46
81.13	60.49	673.98
86.95	52.38	465.47
92.64	55.65	835.73
98.16	58.40	522.19
103.70	54.66	703.22
109.45	55.34	707.52
114.72	61.40	676.76
119.95	59.68	745.92
125.22	59.32	611.41
128.76	58.69	735.46
146.67	57.33	771.41
151.97	61.79	672.40
157.38	55.98	564.36
162.56	59.27	751.37
167.94	53.57	693.81
173.33	59.15	661.48
178.61	60.13	706.66
183.94	58.73	764.18
189.25	61.21	669.88
194.51	61.93	676.22
199.75	53.56	745.58
205.08	58.68	700.68

ตารางที่ 4.8 แสดงค่า Average User และ Server Bandwidth แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	User Bandwidth [kbit/s]	Server Bandwidth [kbit/s]
210.31	61.84	614.11
215.64	52.91	637.64
221.03	55.61	597.34
226.59	54.20	609.18
232.20	51.32	816.14
237.47	61.08	451.55
242.98	53.50	707.31
248.14	56.72	689.84
253.41	55.53	675.97
258.70	59.11	673.17
264.03	56.40	604.67
269.45	61.46	718.94
274.94	60.39	681.56
280.25	58.05	605.53
285.72	52.49	682.12
290.91	57.47	720.33

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.9



รูปที่ 4.9 กราฟแสดงค่าของ Average User และ Server Bandwidth

3) ค่าของการส่งข้อมูล (Transferred Data), หน่วยความจำที่ใช้ในระบบ (System Memory) และโหลดที่ใช้ในหน่วยประมวลผลกลาง (CPU Load) โดยค่าของการส่งข้อมูลคือปริมาณของข้อมูลที่ผู้ใช้ได้ส่งและรับข้อมูลไปที่เซิร์ฟเวอร์ ซึ่งค่าที่ผู้ใช้ส่งข้อมูลไปที่เซิร์ฟเวอร์ เช่น การบรรจุข้อมูลขึ้นไปในระบบ FTP หรือการบรรจุข้อมูลของเว็บต่าง ๆ ผ่านเว็บเบราว์เซอร์ เป็นต้น ส่วนการรับข้อมูลจากเซิร์ฟเวอร์นั้น เช่นการเรียกใช้งานของผู้ใช้ หรือการเปิดเว็บเบราว์เซอร์เป็นต้น โดยในตารางที่ 4.9 แสดงค่าที่ได้จากการทดสอบกับระบบ CO-LMS

ตารางที่ 4.9 แสดงค่า Transferred Data System Memory และ CPU Load แบบที่ใช้ระบบ CO-LMS

Time [ms]	Transferred Data [kbit/s]	Available System Memory [MB]	CPU Load [%]
6.24	32.19	167	53.93
11.33	854.63	167	69.51
16.70	702.29	167	81.43
22.15	665.75	167	91.73
27.68	719.31	166	76.94
33.12	641.48	166	62.87
38.54	622.94	166	65.39
43.94	702.62	166	62.71

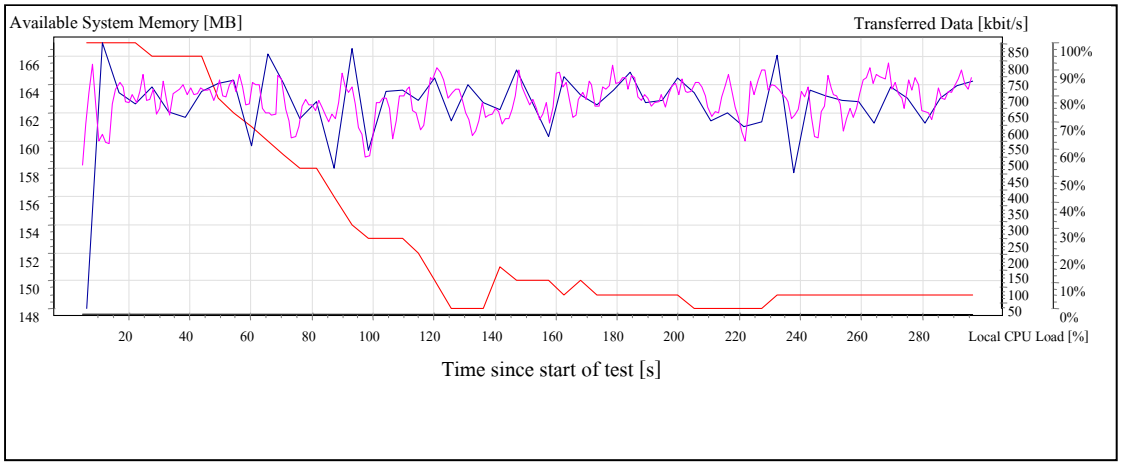
ตารางที่ 4.9 แสดงค่า Transferred Data System Memory และ CPU Load
แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Transferred Data [kbit/s]	Available System Memory [MB]	CPU Load [%]
54.54	739.43	162	77.05
60.23	535.81	161	82.64
65.52	819.51	160	84.98
70.77	729.43	159	83.52
76.00	618.46	158	77.72
81.66	673.98	158	77.50
87.47	465.47	156	80.51
92.95	835.73	154	77.28
98.62	522.19	153	81.83
104.20	703.22	153	88.25
109.70	707.52	153	78.19
114.89	676.76	152	78.68
120.14	745.92	150	82.44
125.76	611.41	148	73.13
131.11	723.89	148	75.34
136.34	670.27	148	85.39
141.55	646.90	151	78.71
146.83	771.41	150	72.92
152.13	672.40	150	81.05
157.55	564.36	150	81.82
162.75	751.37	149	82.75
168.12	693.81	150	84.43
173.49	661.48	149	80.49
178.89	706.66	149	82.90
184.20	764.18	149	80.41
189.42	669.88	149	80.76

ตารางที่ 4.9 แสดงค่า Transferred Data System Memory และ CPU Load
แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Transferred Data [kbit/s]	Available System Memory [MB]	CPU Load [%]
194.68	676.22	149	73.86
199.91	745.58	149	76.32
205.24	700.68	148	87.47
215.98	637.64	148	80.52
221.50	597.34	148	79.98
227.19	609.18	148	77.22
232.38	816.14	149	66.82
237.95	451.55	149	71.34
243.15	707.31	149	75.18
248.34	689.84	149	72.03
253.57	675.97	149	75.50
258.98	673.17	149	81.43
264.30	604.67	149	85.99
269.70	718.94	149	86.82
275.24	681.56	149	90.81
280.70	605.53	149	87.77
285.93	682.12	149	86.50
291.20	720.33	149	74.72
296.42	735.10	149	75.50

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.10



รูปที่ 4.10 กราฟแสดงค่าของ Transferred Data, System Memory และ CPU Load

4) ค่าของ Open Requests และ Transferred Data จากการทดลองนี้จะแสดงค่าแบนด์วิดท์ที่ใช้ (Bandwidth), ค่า Open Requests, ค่าเวลาที่ใช้ในการรับข้อมูล (Received Requests) และค่าเวลาที่ใช้ในการส่งข้อมูล (Sent Requests) โดยแสดงในตารางที่ 4.10

ตารางที่ 4.10 แสดงค่า Open Requests และ Transferred Data แบบที่ใช้ระบบ CO-LMS

Time [ms]	Bandwidth [kbit/s]	Open Requests	Received Requests/s	Sent Requests/s
6.24	32.19	134	38.45	16.09
11.33	854.63	131	32.74	35.54
16.70	702.29	45	35.28	31.23
22.15	665.75	50	33.26	34.89
27.68	719.31	41	26.81	26.44
33.12	641.48	59	29.35	27.88
38.54	622.94	72	32.85	35.07
43.94	702.62	50	35.48	34.93
49.32	727.84	44	33.31	32.55

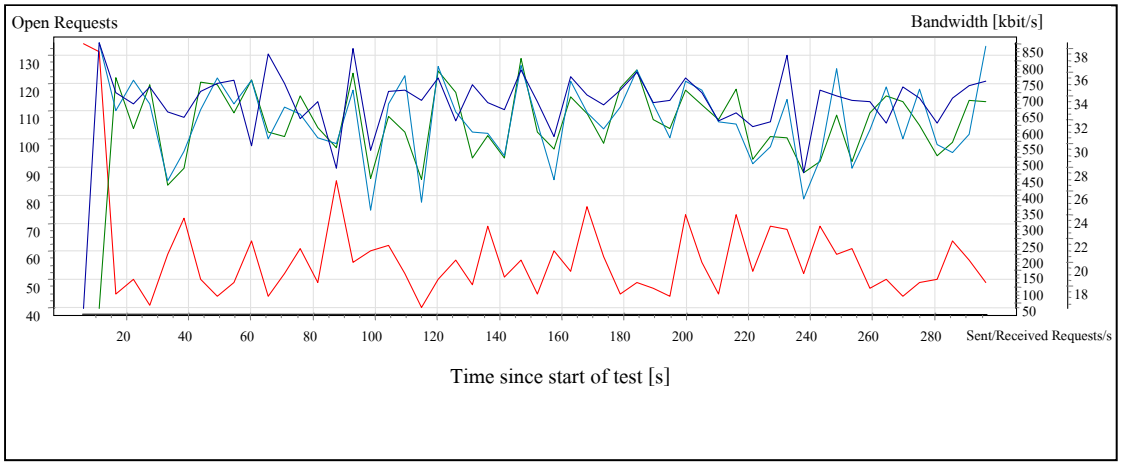
ตารางที่ 4.10 แสดงค่า Open Requests และ Transferred Data แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Bandwidth [kbit/s]	Open Requests	Received Requests/s	Sent Requests/s
54.54	739.43	49	35.35	35.35
60.23	535.81	64	30.35	30.92
65.52	819.51	44	32.98	30.50
70.77	729.43	52	32.47	33.99
76.00	618.46	61	30.42	31.31
81.66	673.98	49	29.94	29.59
87.47	465.47	85	34.46	35.92
92.95	835.73	56	24.35	27.00
98.62	522.19	60	33.31	32.23
104.20	703.22	62	35.69	30.96
109.70	707.52	52	25.02	26.95
114.89	676.76	40	36.42	36.04
120.14	745.92	50	32.62	34.22
125.76	611.41	57	30.96	28.71
131.11	723.89	48	30.79	30.60
136.34	670.27	69	28.94	28.75
141.55	646.90	51	36.58	37.14
146.83	771.41	57	31.69	30.94
152.13	672.40	45	26.93	29.52
157.55	564.36	60	35.23	33.89
162.75	751.37	53	32.60	32.41
168.12	693.81	76	31.24	29.94
173.49	661.48	58	32.98	34.65
178.89	706.66	45	36.18	36.18
184.20	764.18	49	33.28	31.94
189.42	669.88	47	30.44	31.20

ตารางที่ 4.10 แสดงค่า Open Requests และ Transferred Data แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Bandwidth [kbit/s]	Open Requests	Received Requests/s	Sent Requests/s
194.68	676.22	44	35.19	34.43
199.91	745.58	73	34.49	33.18
205.24	700.68	56	31.78	31.96
210.59	614.11	45	31.57	34.54
215.98	637.64	73	28.28	28.64
221.50	597.34	53	29.69	30.57
227.19	609.18	69	33.71	30.44
232.38	816.14	68	25.30	27.46
237.95	451.55	52	28.66	28.47
243.15	707.31	69	36.23	32.39
248.34	689.84	59	27.85	28.43
253.57	675.97	61	31.08	32.56
258.98	673.17	47	34.72	33.97
264.30	604.67	50	30.35	33.49
269.70	718.94	44	34.53	31.46
275.24	681.56	49	29.84	28.92
280.70	605.53	50	29.26	30.03
285.93	682.12	64	30.75	33.60
291.20	720.33	57	38.12	33.53
296.42	735.10	49	38.65	33.15

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.11



รูปที่ 4.11 กราฟแสดงค่าของ Open Requests และ Transferred Data

5) ค่าเวลาที่ใช้ในการร้องขอข้อมูล (Click Time) และค่าเวลาที่ระบบทำการตอบสนอง (Hits/s) ต่อผู้ใช้ โดยที่จากการทดลองนี้จะได้ค่าความสัมพันธ์ต่าง ๆ ระหว่างที่ผู้ใช้ได้ทำการร้องขอข้อมูลมายังระบบ โดยในตารางที่ 4.11 แสดงค่าที่ได้จากการทดสอบกับระบบ CO-LMS

ตารางที่ 4.11 แสดงค่า Click Time User/s และ Hits/s for all URLs แบบที่ใช้ระบบ CO-LMS

Time [ms]	Click Time [ms]	Clicks/s	Hits/s
5.89	4203.00	0.19	1.52
11.28	5448.96	5.01	40.10
16.36	2770.57	4.14	33.13
27.37	2765.78	4.24	33.93
32.94	3020.62	3.78	30.26
38.37	3471.35	3.67	29.39
43.69	2864.14	4.14	33.14
49.05	2750.48	4.29	34.33
54.31	2818.96	4.36	34.88
59.70	2869.65	3.16	25.28

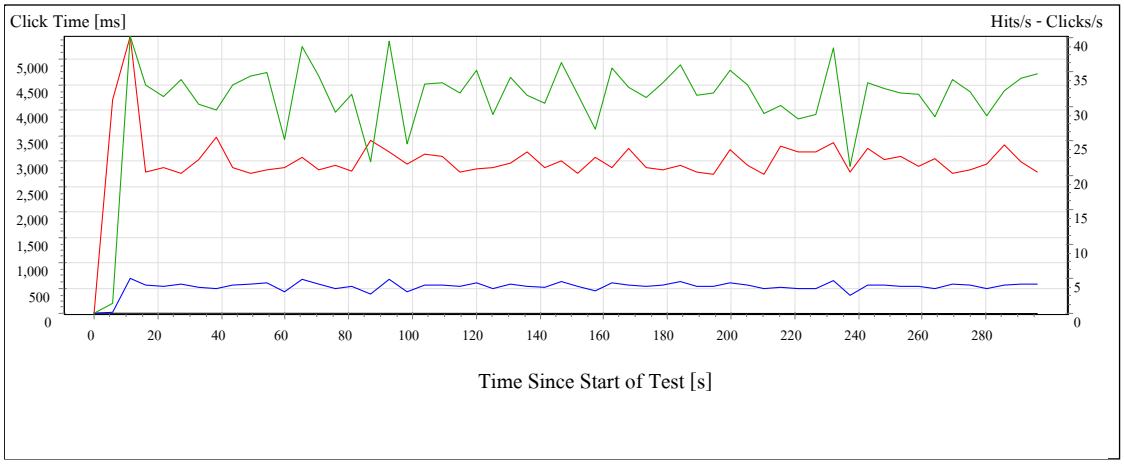
ตารางที่ 4.11 แสดงค่า Click Time User/s และ Hits/s for all URLs แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Click Time [ms]	Clicks/s	Hits/s
65.28	3069.00	4.83	38.66
70.64	2821.00	4.30	34.41
75.84	2918.16	3.65	29.17
81.13	2812.05	3.97	31.79
86.95	3413.13	2.74	21.96
92.64	3179.93	4.93	39.42
98.16	2935.00	3.08	24.63
103.70	3145.39	4.15	33.17
109.45	3093.83	4.17	33.38
114.72	2771.43	3.99	31.92
119.95	2856.39	4.40	35.19
125.22	2875.84	3.61	28.84
130.84	2950.79	4.27	34.15
136.16	3175.95	3.95	31.62
141.39	2872.80	3.81	30.52
146.67	2993.17	4.55	36.39
151.97	2747.52	3.96	31.72
157.38	3070.83	3.33	26.62
162.56	2870.04	4.43	35.44
167.94	3240.64	4.09	32.73
173.33	2876.86	3.90	31.20
178.61	2825.68	4.17	33.33
183.94	2902.54	4.51	36.05
189.25	2773.62	3.95	31.60
194.51	2740.67	3.99	31.90
199.75	3231.39	4.40	35.17
205.08	2905.77	4.13	33.05

ตารางที่ 4.11 แสดงค่า Click Time User/s และ Hits/s for all URLs แบบที่ใช้ระบบ CO-LMS (ต่อ)

Time [ms]	Click Time [ms]	Clicks/s	Hits/s
210.31	2745.58	3.62	28.97
215.64	3287.15	3.76	30.08
221.03	3190.84	3.52	28.18
226.59	3189.00	3.59	28.74
232.20	3353.07	4.81	38.50
237.47	2778.21	2.66	21.30
242.98	3257.96	4.17	33.37
248.14	3025.24	4.07	32.54
253.41	3101.62	3.99	31.89
258.70	2885.38	3.97	31.75
264.03	3037.68	3.57	28.52
269.45	2762.30	4.24	33.91
274.94	2816.32	4.02	32.15
280.25	2944.16	3.57	28.56
285.72	3305.41	4.02	32.18
290.91	2986.23	4.25	33.98
296.20	2787.65	4.33	34.68

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.12



รูปที่ 4.12 กราฟแสดงค่าของ Click Time, Hits/s และ User/s for all URLs

6) ค่าเฉลี่ยของเวลาที่ได้จากการร้องขอข้อมูลจากระบบ (Average Request Time) และค่าความผิดพลาดเมื่อทำการร้องขอข้อมูลจากระบบ (Errors) โดยแสดงในตารางที่ 4.12

ตารางที่ 4.12 แสดงค่า Average Request Time และ Errors (per URL) แบบที่ใช้ระบบ CO-LMS

Time [ms]	Average Request Time [ms]	Average Request Time [ms] All Images	Errors [%]	
5.89	4203.23	830.74	0	0%
11.28	5449.03	1599.53	0	0%
16.36	2770.56	557.48	0	0%
21.95	2858.58	645.17	0	0%
27.37	2765.81	598.27	0	0%
32.94	3020.52	736.39	0	0%
38.37	3471.33	1069.22	0	0%
43.69	2864.19	674.79	0	0%
49.05	2750.49	640.36	0	0%
54.31	2819.02	618.64	0	0%

ตารางที่ 4.12 แสดงค่า Average Request Time และ Errors (per URL)

แบบที่ใช้ระบบ CO-LMS (ต่อ)

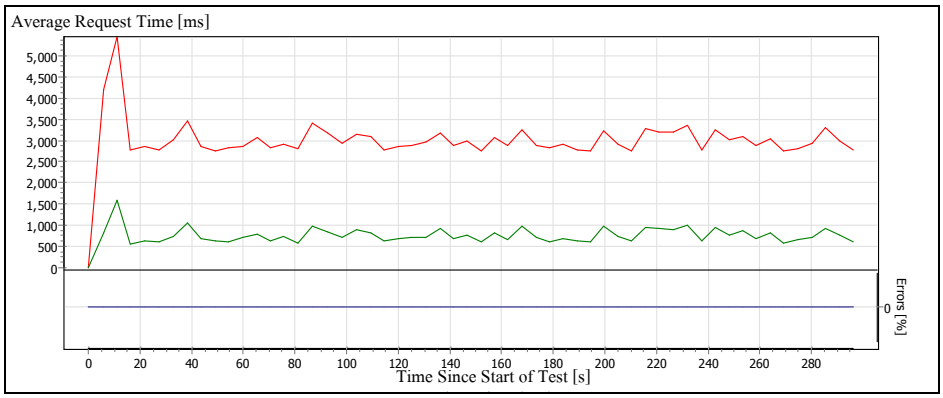
Time [ms]	Average Request Time [ms]	Average Request Time [ms] All Images	Errors [%]	
59.70	2869.63	703.64	0	0%
65.28	3068.99	800.42	0	0%
70.64	2820.97	630.55	0	0%
75.84	2918.10	750.55	0	0%
81.13	2812.10	586.79	0	0%
86.95	3413.18	983.04	0	0%
92.64	3179.94	839.22	0	0%
98.16	2935.12	701.33	0	0%
103.70	3145.47	895.91	0	0%
109.45	3093.83	824.10	0	0%
114.72	2771.40	631.24	0	0%
119.95	2856.43	683.94	0	0%
128.34	2870.10	672.12	0	0%
125.22	2875.87	713.41	0	0%
130.84	2950.81	703.83	0	0%
136.16	3175.94	933.07	0	0%
141.39	2872.77	700.15	0	0%
146.67	2993.15	776.43	0	0%
151.97	2747.53	612.49	0	0%
157.38	3070.80	827.34	0	0%
162.56	2870.10	672.12	0	0%
167.94	3240.55	968.64	0	0%
173.33	2876.83	702.31	0	0%
178.61	2825.60	614.73	0	0%
183.94	2902.64	698.05	0	0%
189.25	2773.67	623.77	0	0%

ตารางที่ 4.12 แสดงค่า Average Request Time และ Errors (per URL)

แบบที่ใช้ระบบ CO-LMS (ต่อ)

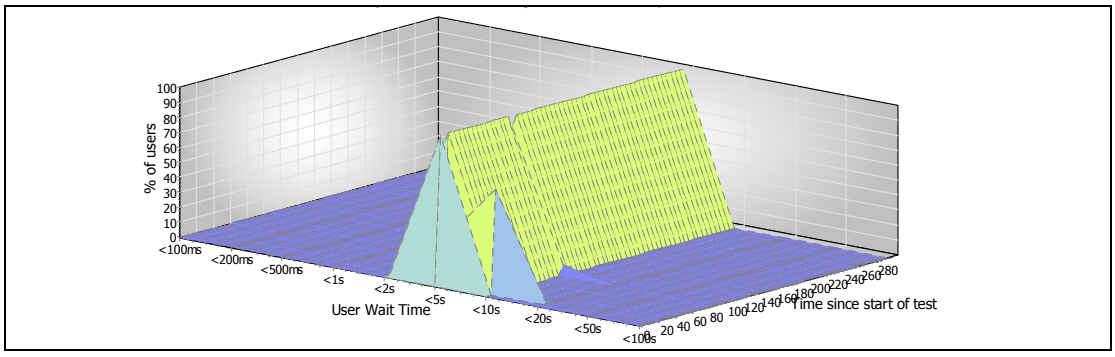
Time [ms]	Average Request Time [ms]	Average Request Time [ms] All Images	Errors [%]	
194.51	2740.63	616.72	0	0%
199.75	3231.32	965.72	0	0%
205.08	2905.82	752.72	0	0%
210.31	2745.60	638.84	0	0%
215.64	3287.20	941.94	0	0%
221.03	3190.74	919.72	0	0%
226.59	3189.12	894.88	0	0%
232.20	3353.05	1008.83	0	0%
237.47	2778.16	632.16	0	0%
242.98	3257.99	957.77	0	0%
248.14	3025.23	767.68	0	0%
253.41	3101.67	874.95	0	0%
258.70	2885.36	700.72	0	0%
264.03	3037.57	810.55	0	0%
269.45	2762.36	580.08	0	0%
274.94	2816.27	651.30	0	0%
280.25	2944.17	718.32	0	0%
285.72	3305.48	936.40	0	0%
290.91	2986.21	758.02	0	0%

เมื่อนำค่าที่ได้ในตารางมาวิเคราะห์สามารถแสดงผลได้เป็นกราฟดังรูปที่ 4.13



รูปที่ 4.13 กราฟแสดงค่าของ Average Request Time และ Errors (per URL)

เมื่อนำเวลาที่ผู้ใช้ต้องรอระหว่างการร้องขอข้อมูลจากระบบ (User Wait Time) กับเวลาที่ใช้ในการทดสอบ (Time Since Start of Test) และเปอร์เซ็นต์ค่าความผิดพลาดเมื่อผู้ใช้ร้องขอข้อมูลจากระบบ (% error of User) มาทำการแสดงผลเป็นแบบ Spectrum จะทำให้สามารถวิเคราะห์ได้ว่า เมื่อผู้ใช้ทำการร้องขอข้อมูลจากระบบที่นำระบบ CO-LMS ข้อมูลของเอาท์พุตมาพัฒนาร่วมนั้น เวลาที่ผู้ใช้ร้องขอข้อมูลจากระบบเป็นไปอย่างต่อเนื่องคงที่สม่ำเสมอไม่กระจัดกระจาย



รูปที่ 4.14 กราฟแสดงค่า Spectrum of Click Times

โดยสรุปแล้ว เมื่อทำการทดสอบระบบแบบเดิมกับระบบ CO-LMS มาพัฒนาร่วมนั้น จะได้ค่าความสัมพันธ์ต่าง ๆ ที่เกิดขึ้นมา ซึ่งเมื่อนำมาวิเคราะห์ผลแล้วจะทำให้ทราบได้ว่าระบบ

แบบใดที่จะมีประสิทธิภาพดีกว่า การที่จะสามารถทดสอบได้ว่าระบบนั้นมีประสิทธิภาพมากขึ้นหรือไม่นั้น จะสามารถวิเคราะห์ได้จากค่าต่าง ๆ ได้จาก จำนวนผู้ใช้งานระบบ (Clicks) ค่าตอบสนองของระบบต่อผู้ใช้ (Hits) ค่าเวลาที่ระบบตอบสนองต่อผู้ใช้ (Avg. Click Time) ขนาดข้อมูลที่ระบบส่งให้ผู้ใช้ (Bytes) และขนาดข้อมูลที่ส่งไปในระบบ (kbit/s) โดยแสดงผลได้ดังนี้

ตารางที่ 4.13 แสดงค่าผลที่ได้จากระบบแบบเดิม

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s
1	43	344	0	4,993	927,811	34.57
2	46	360	0	4,567	970,965	37.79
3	40	320	0	5,815	863,080	29.69
4	43	336	0	4,823	906,234	35.79
5	27	216	0	9,661	582,579	17.87
6	32	256	0	7,281	690,464	23.71
7	35	280	0	6,430	755,195	26.85
8	42	328	0	5,021	884,657	34.38
9	35	280	0	6,689	755,195	25.81
10	34	272	0	7,199	733,618	23.98
11	42	336	0	5,165	906,234	33.42
12	40	312	0	5,574	841,503	30.97
13	43	336	0	5,103	906,234	33.83
14	43	336	0	5,019	906,234	34.40
15	38	304	0	5,856	819,926	29.48
16	46	368	0	4,717	992,542	36.59
17	36	288	0	6,005	776,772	28.75
18	40	312	0	5,794	841,503	29.79
19	43	344	0	4,815	927,811	35.85
20	39	312	0	5,588	841,503	30.89

ตารางที่ 4.14 แสดงค่าผลที่ได้จากระบบแบบที่ได้นำระบบ CO-LMS

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s
1	64	512	0	2,936	1,357,116	57.78
2	62	496	0	3,125	1,314,718	54.28
3	62	488	0	2,921	1,293,139	58.06
4	58	464	0	3,147	1,229,922	53.91
5	56	448	0	2,968	1,187,524	57.17
6	56	448	0	3,080	1,187,144	55.07
7	60	480	0	2,965	1,271,940	57.20
8	59	464	0	3,073	1,229,542	55.19
9	65	512	0	2,948	1,356,736	57.52
10	59	464	0	3,125	1,229,922	54.29
11	59	464	0	2,977	1,229,542	56.97
12	59	472	0	3,115	1,251,121	54.46
13	63	496	0	3,001	1,314,338	56.52
14	57	448	0	3,137	1,187,144	54.06
15	62	488	0	2,919	1,293,139	58.11
16	55	432	0	3,194	1,144,746	53.10
17	63	504	0	2,947	1,335,917	57.57
18	55	432	0	3,151	1,144,746	53.83
19	61	480	0	2,933	1,272,320	57.85
20	60	480	0	2,994	1,271,940	56.64

เมื่อนำผลที่ได้จากระบบทั้งสองมาหาค่าเฉลี่ยทั้งหมด จะทำให้ได้ความสัมพันธ์ต่าง ๆ ออกมาดังนี้

ตารางที่ 4.15 แสดงค่าเฉลี่ยของ Clicks Hits Errors Average Click Time Bytes และ kbit/s ในการทดสอบระบบแบบเดิมกับระบบ CO-LMS

System Test	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s
ระบบ CO-LMS	60	473.6	0	3032.8	1255133	55.979
ระบบเดิม	60	468	0	8709	1262254	46.08075

จากตารางที่ 4.15 จะทำให้ทราบค่า Clicks ซึ่งเป็นค่าของจำนวนที่ผู้ใช้ใช้ในการคลิกระบบ โดยค่าที่ได้จะเป็นค่าเฉลี่ย และค่า Hits คือจำนวนไฟล์ที่เซิร์ฟเวอร์ทำการดาวน์โหลดมาจากเว็บเซิร์ฟเวอร์ โดยในที่นี้ความหมายของไฟล์ก็คือ หน้าของเว็บไซต์ และรูปภาพต่าง ๆ เป็นต้น ตัวอย่างเช่น ถ้ามีผู้ดาวน์โหลด 1 หน้าเว็บไซต์ที่ประกอบไปด้วยรูปภาพ 4 รูป ซึ่งจะมี Hits ทั้งหมดจำนวน 5 Hits เพราะ Hits ครั้งที่ 1 มาจากการดาวน์โหลดหน้าเว็บไซต์ และที่เหลืออีก 4 ครั้งจากการดาวน์โหลดรูปภาพ 4 รูปภาพนั่นเอง ด้วยเหตุนี้จึงทำให้ Hits เป็นตัวชี้วัดที่ไม่ดีนัก เพราะมีจำนวนสูงมากเนื่องจากปกติแล้วหนึ่งหน้าเว็บไซต์จะประกอบไปด้วยหลายไฟล์ ส่วนค่า Errors จะเป็นค่าผิดพลาดที่ได้จากระบบโดยเมื่อระบบมีปัญหาในการติดต่อกับผู้ใช้ จะทำให้เกิดค่าผิดพลาดขึ้นได้ และค่า Average Click Time เป็นค่าเวลาเฉลี่ยที่ผู้ใช้ทำการติดต่อไปยังเซิร์ฟเวอร์ ส่วนค่า Bytes จะเป็นขนาดข้อมูลที่ผู้ใช้ทำการ โหลดจากเซิร์ฟเวอร์ และค่า kbit/s จะเป็นค่าที่เซิร์ฟเวอร์ส่งข้อมูลให้กับเซิร์ฟเวอร์ ซึ่งเมื่อทำการวิเคราะห์แล้วจะเห็นว่าจำนวนการคลิกของผู้ใช้ที่เท่ากัน ค่าของ Hits ในระบบ CO-LMS จะมากกว่าเพราะอาจจะมีการสร้างไฟล์ให้ผู้ใช้โหลดจำนวนมาก แต่ค่าเวลาเฉลี่ยในการตอบสนองการคลิกนั้นน้อยกว่ามาก ทำให้เมื่อผู้ใช้โหลดข้อมูลจากเซิร์ฟเวอร์จึงน้อยกว่าระบบแบบเดิม ซึ่งทำให้การส่งข้อมูลจากเซิร์ฟเวอร์ให้กับผู้ใช้นั้นได้มากขึ้นอีกด้วย

และผลที่ได้จากการทดสอบระบบจะทำให้ได้ค่าเวลาที่ใช้ในการอ่านข้อมูล (Time Spent) และเวลาที่ใช้ตอบสนองต่อผู้ใช้ (Average Click Time) ต่อ URL ดังนี้

ตารางที่ 4.16 แสดงค่า Time Spent และ Avg. Click Time จากระบบแบบเดิม

URL No.	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	767	0	0.00	4,235,841	5,523

ตารางที่ 4.17 แสดงค่า Time Spent และ Avg. Click Time จากระบบแบบที่ได้นำระบบ CO-LMS

URL No.	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	1,163	0	0.00	3,530,673	3,036

จากค่าต่าง ๆ ที่ได้ในตารางข้างต้นนี้ สามารถนำมาหาความสัมพันธ์ระหว่างระบบทั้งสองแบบ เช่น ค่าของ Time Spent คือค่าของเวลาที่ผู้ใช้เรียกใช้งานหน้าเว็บเพจนั้นเป็นเวลานานเท่าไหร่หากค่าเวลามาก แสดงว่าผู้ใช้เรียกใช้หน้าเว็บเพจดังกล่าวเป็นเวลานาน เป็นต้น ซึ่งจะทำให้สามารถวิเคราะห์ถึงประสิทธิภาพในการค้นคืนข้อมูลที่ชัดเจนมากขึ้นว่าระบบแบบใด ที่จะสามารถช่วยเพิ่มประสิทธิภาพให้กับระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตได้ ดังแสดงในตารางที่ 4.17 ซึ่งระบบแบบเดิมนั้นจะต้องใช้เวลามากในการตอบสนองกับผู้ใช้และเป็นไปโดยไม่สม่ำเสมอ จะมีการเปลี่ยนแปลงเวลาค่อนข้างมาก แต่ระบบ CO-LMS นั้นจะมีการตอบสนองต่อผู้ใช้ที่รวดเร็วกว่า ซึ่งจะมีเพียงช่วงแรกเท่านั้นที่ผู้ใช้จะต้องใช้เวลาค่อนข้างเยอะเพื่อรอการตอบสนอง แต่เมื่อมีการทำการ Caching ข้อมูลเก็บไว้ในระบบเรียบร้อยแล้วนั้น เมื่อผู้ใช้ทำการร้องขอข้อมูลเดิมอีกครั้ง ระบบ CO-LMS นั้นก็จะทำการตอบสนองได้เร็วขึ้น และเป็นไปอย่างสม่ำเสมออีกด้วย

ตารางที่ 4.18 แสดงการเปรียบเทียบค่าของเวลาที่ผู้ใช้ได้รับการตอบสนองจากระบบ

Time [ms]	Average Request Time [ms] ของระบบ CO-LMS	Average Request Time [ms] ของระบบแบบเดิม
0	0	0
5.89	4203.23	4868.36
11.28	5449.03	4497.3
16.36	2770.56	4694.63
21.95	2858.58	0
27.37	2765.81	8643.08
32.94	3020.52	4362.41
38.37	3471.33	5010.85
43.69	2864.19	5356.94

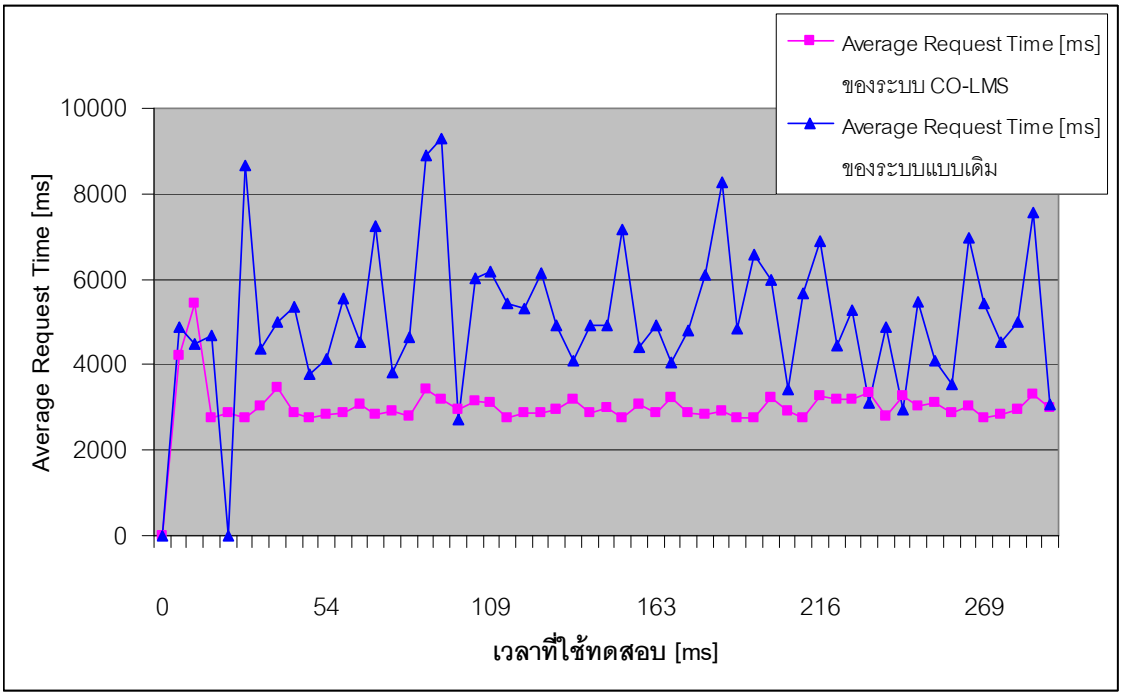
ตารางที่ 4.18 แสดงการเปรียบเทียบค่าของเวลาที่ผู้ใช้ได้รับการตอบสนองจากระบบ (ต่อ)

Time [ms]	Average Request Time [ms] ของระบบ CO-LMS	Average Request Time [ms] ของระบบแบบเดิม
49.05	2750.49	3773.12
54.31	2819.02	4136.17
59.7	2869.63	5550.52
65.28	3068.99	4520.27
70.64	2820.97	7247.79
75.84	2918.1	3807.29
81.13	2812.1	4658.6
86.95	3413.18	8895.21
92.64	3179.94	9283.53
98.16	2935.12	2721.05
103.7	3145.47	6025.33
109.45	3093.83	6192.46
114.72	2771.4	5450.18
119.95	2856.43	5300.8
122.89	2787.53	7149.01
125.22	2875.87	6124.36
130.84	2950.81	4934.38
136.16	3175.94	4075
141.39	2872.77	4917.6
146.67	2993.15	4907.17
151.97	2747.53	7159.01
157.38	3070.8	4420.21
162.56	2870.1	4931.19
167.94	3240.55	4044.12

ตารางที่ 4.18 แสดงการเปรียบเทียบค่าของเวลาที่ผู้ใช้ได้รับการตอบสนองจากระบบ (ต่อ)

Time [ms]	Average Request Time [ms] ของระบบ CO-LMS	Average Request Time [ms] ของระบบแบบเดิม
173.33	2876.83	4804.25
178.61	2825.6	6105.84
183.94	2902.64	8277.2
189.25	2773.67	4828.76
194.51	2740.63	6586.35
199.75	3231.32	5984.88
205.08	2905.82	3431.87
210.31	2745.6	5661.06
215.64	3287.2	6900.01
221.03	3190.74	4468.37
226.59	3189.12	5256.16
232.2	3353.05	3105.38
237.47	2778.16	4886.06
242.98	3257.99	2944.19
248.14	3025.23	5465.51
253.41	3101.67	4093.14
258.7	2885.36	3549.85
264.03	3037.57	6956.57
269.45	2762.36	5416.5
274.94	2816.27	4544.97
280.25	2944.17	4982.23
285.72	3305.48	7564.72
290.91	2986.21	3089.15

ซึ่งสามารถนำมาวิเคราะห์ออกมาเป็นกราฟได้ดังรูปที่ 4.15



รูปที่ 4.15 แสดงการเปรียบเทียบค่าเฉลี่ยของเวลาในการตอบสนองต่อผู้ใช้

4.3 อภิปรายผล

ในการทดสอบระบบการค้นคืนข้อมูลนี้ ระบบสามารถค้นคืนข้อมูลได้อย่างมีประสิทธิภาพ และมีลำดับขั้นตอนในการทำงานที่ไม่ยุ่งยาก สามารถตอบสนองต่อความต้องการของผู้ใช้ได้ ทำให้ผู้ใช้สามารถใช้งานการค้นคืนข้อมูลต่าง ๆ ได้รวดเร็วขึ้น ซึ่งเมื่อนำค่าที่ได้จากการทดสอบระบบทั้งสองแบบนั้น จะสามารถทำให้ทราบได้ว่าระบบแบบที่ได้นำระบบ CO-LMS ข้อมูลของเอาท์พุตมาพัฒนาร่วมนั้น สามารถช่วยเพิ่มประสิทธิภาพให้กับระบบเซิร์ฟเวอร์ หรือระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตได้ดียิ่งขึ้น

บทที่ 5
บทสรุป

จากการศึกษา วิจัย และพัฒนาระบบการค้นคืนข้อมูลนั้น สามารถสรุปได้ว่า ในการค้นคืนข้อมูลเพื่อให้เกิดประสิทธิภาพมากที่สุด จะต้องใช้เวลาในการตอบสนองต่อผู้ใช้ได้อย่างรวดเร็ว ถูกต้องที่สุด ซึ่งได้มีแนวทางและวิธีการหลากหลายที่จะสามารถช่วยให้ระบบการค้นคืนข้อมูลนั้นมีประสิทธิภาพมากขึ้น เช่น การใช้เทคโนโลยีเอ-แจค (AJAX) การทำระบบ Proxy Server และการทำ Load Balancing เหล่านี้เป็นต้น แต่เทคนิคและวิธีดังกล่าวอาจจะต้องใช้ความรู้ความสามารถของผู้ใช้งานค่อนข้างเยอะ อีกทั้งยังเป็นการสิ้นเปลืองค่าใช้จ่ายมากอีกด้วย

ในการศึกษางานวิจัยนี้ จึงมุ่งเน้นในเรื่องของการค้นคืนข้อมูลจากระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต เพื่อให้มีประสิทธิภาพเพิ่มมากขึ้นจากระบบแบบเดิมที่ยังไม่ได้มีการนำเทคโนโลยีใด ๆ มาใช้ร่วมด้วย ในกรณีที่เว็บเพจมีการดึงข้อมูลจากฐานข้อมูลมาแสดงเป็นจำนวนมาก เช่นการเก็บข่าวไว้ในฐานข้อมูลแล้วแสดง Link Topic ในหน้าแรก หากผู้ใช้แต่ละคนเข้ามาที่เว็บเซิร์ฟเวอร์ในแต่ละครั้งแล้วทำการดึงข้อมูลทุกครั้งหรือมีการ Concurrent มาก ๆ ก็จะทำให้ระบบทำงานล่าช้าจนอาจทำให้ระบบเสียหายได้ ดังนั้นในภาษา PHP มีฟังก์ชันในการเก็บ Output Cache หรือ Data Cache เอาไว้ได้ เช่นมีการ Query ข้อมูลใหม่เฉพาะเมื่อข้อมูลในฐานข้อมูลเปลี่ยนไป หากข้อมูลเหมือนเดิมก็ให้ทำการดึงข้อมูลจาก Cache ที่เก็บไว้นั้นไปแสดงผลแทนโดยที่ไม่ต้อง Query ซ้ำ

5.1 สรุปผลการวิจัย

5.1.1 สรุปผลการทดสอบระบบการค้นคืนข้อมูลแบบเดิม

การค้นคืนข้อมูลระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตแบบเดิมนั้น จะตอบสนองต่อผู้ใช้ค่อนข้างช้า อีกทั้งยังไม่สามารถช่วยให้ทรัพยากรของระบบทำงานได้อย่างมีประสิทธิภาพ เป็นการเพิ่มภาระให้ระบบในการทำงาน ซึ่งอาจจะส่งผลต่อการทำงานอื่น ๆ ในระบบได้อีกด้วย

5.1.2 สรุปผลการทดสอบระบบการค้นคืนข้อมูลแบบการ Caching ข้อมูลของเอาท์พุต

ส่วนการค้นคืนข้อมูลระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต โดยใช้การ Caching ข้อมูลของเอาท์พุตนั้นสามารถช่วยเพิ่มประสิทธิภาพให้กับระบบได้มากขึ้น โดยการ

ตอบสนองต่อความต้องการของผู้ใช้ได้รวดเร็วขึ้น อีกทั้งยังช่วยลดภาระการทำงานของทรัพยากรของระบบให้ทำงานได้อย่างมีประสิทธิภาพมากขึ้นอีกด้วย

5.2 ประโยชน์ของระบบการค้นคืนข้อมูลจากการ Caching ข้อมูลของเอาท์พุต

1) ช่วยให้ผู้ใช้งานระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ได้รับการตอบสนองที่รวดเร็วขึ้น

2) ช่วยลดภาระการใช้งานทรัพยากรของระบบ ทำให้ระบบสามารถทำงานคำสั่งต่าง ๆ ได้รวดเร็วขึ้น

5.3 ข้อจำกัดของระบบการค้นคืนข้อมูลจากการ Caching ข้อมูลของเอาท์พุต

1) ระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่ใช้งานอยู่นั้น ทำงานได้ไม่ดีสำหรับเว็บที่เป็นแบบเว็บเซอร์วิสและมีการเปลี่ยนแปลงตลอดเวลาไม่เหมาะกับการใช้อัลกอริธึมนี้ หากข้อมูลในระบบมีการเปลี่ยนแปลงอยู่ตลอดเวลา ระบบการ Caching ข้อมูลของเอาท์พุตอาจจะไม่ทำให้เกิดผลได้ดีมากนัก

2) ความสามารถของผู้ดูแลระบบ อาจจะต้องมีความรู้พื้นฐานในการเขียนโปรแกรมภาษา PHP และยังต้องรู้และเข้าใจในการกำหนดค่าต่าง ๆ ในระบบปฏิบัติการลินุกซ์

5.4 แนวทางในการพัฒนาต่อ

ระบบ CO-LMS สามารถตอบสนองต่อความต้องการของผู้ใช้ได้ในระดับหนึ่ง ซึ่งเมื่อนำระบบดังกล่าวมาพัฒนาร่วมกับการค้นคืนข้อมูลให้กับระบบเว็บไซต์ที่เป็นแบบไดนามิก และระบบนั้นมีการเปลี่ยนแปลงอยู่ตลอดเวลา ก็อาจจะทำให้ไม่สามารถใช้งานได้เต็มที่เท่าไร ดังนั้นเพื่อเป็นการเพิ่มประสิทธิภาพให้ระบบสามารถทำงานได้มากยิ่งขึ้นนั้น อาจจะต้องมีการเทคนิควิธีอื่น ๆ มาทำงานร่วมด้วย เช่น การใช้เทคนิคการบัฟเฟอร์ของ gzip ซึ่งจะทำการบีบอัดข้อมูลในส่วนที่ไม่ได้ทำการ Caching ข้อมูลเอาไว้ ส่งไปให้ผู้ใช้แทนซึ่งก็จะทำให้ระบบมีประสิทธิภาพเพิ่มมากขึ้นอีก

รายการอ้างอิง

Arun Iyengar, Jim Challengar, Daniel Dias และ Paul Dantzig. (2000). **High-Performance Web Site Design Techniques**. IBM T.J. Watson Research Center, IEEE Internet Computing.

Arun Iyengar, Jim Challenger และ Paul Dantzig. (1999). **A Scalable System for Consistently Caching Dynamic web Data**. INFOCOM 1999: 294-303.

David Guerrero. (1999). **Caching The Web : Improve your users browsing and save your bandwidth by using proxy servers to cache web pages**. [On-line]. Available: http://www.david-guerrero.com/papers/squid/squid.htm.

Doug Sheppard. (2000). **Beginner's Introduction to Perl**. [On-line]. Available: http://www.perl.com/pub/a/2000/10/begperl1.html.

Greg Murray. (2006). **Asynchronous JavaScript Technology and XML (Ajax) With the Java Platform**. [On-line]. Available: http://java.sun.com/developer/technicalArticles/J2EE/AJAX/

Guido van Rossum. (1989). **What's New in Python**. [On-line]. Available: http://www.python.org/doc/essays/ppt/stanford2003/Stanford.pdf

IDESIGN.IN.TH. (2005). **Web Caching and Multi-level Web Caching**. [On-line]. Available: http://tutorials.idesign.in.th/component/content/article/34-2008-03-16-18-18-11/45-web-caching-and-multi-level-web-caching.html

Jesse James Garrett. (2005). **Ajax: A New Approach to Web Applications**. [On-line]. Available: http://www.adaptivepath.com/ideas/essays/archives/000385.php

K. Gottschalk, S. Graham, H. Kreger และ J. Snell. (2002). **Introduction to Web services architecture**. [On-line]. Available: http://researchweb.watson.ibm.com/journal/sj/412/gottschalk.html.

Moodle. (2008). **Moodle**. [On-line]. Available: http://moodle.org/.

MySQL. (2008). **MySQL 6.0 DOWNLOADS**. [On-line]. Available: http://dev.mysql.com/downloads/mysql/6.0.html

MySQL. (2008). **New CEO at MySQL AB**. [On-line]. Available: http://www.mysql.fr/news-and-events/press-release/release_2001_1.html

Olivier Mueller and Marc Delisle. (2001). **phpMyAdmin**. [On-line]. Available: http://sourceforge.net/projects/phpmyadmin/

Yeim-Kuan Chang, Yu-Ren Lin and Yi-Wei Ting. (2004). **Caching Personalized and Database-related Dynamic Web Pages**. IEEE, NAS '06. International Workshop: 5-9

กิตติ ภักดีวัฒนะกุล. (2547)**. คัมภีร์ PHP**. พิมพ์ครั้งที่ 2. สำนักพิมพ์ เคทีพี

คณะเภสัชศาสตร์ มหาวิทยาลัยสงขลานครินทร์. (2550). **Moodle คืออะไร**. [ออนไลน์]. ได้จาก: http://elearning.pharmacy.psu.ac.th/index.php?option=com_content&task=view&id=6 2 &Item id=34

จักรพันธ์ มาดีตระกูล และ สุขุมาล กิติสิน. (2547). **Decenttralized Web Cache:Replacent Algorithms and Object:Placement Using MD5**. ThCSC The 1st Thailand Computer Science Conference มหาวิทยาลัยเกษตรศาสตร์

ซีเอ็มเอสไทยแลนด์. (2550). **เทคโนโลยีเว็บไซต์**. [ออนไลน์]. ได้จาก: http://www.cmsthailand.com/web45-47/print.php?sid=68

ซีเอ็มเอสไทยแลนด์. (2550). **ยุคของการพัฒนาเว็บแอพพลิเคชัน**. [ออนไลน์]. ได้จาก: http://www.cmsthailand.com/ws/

วิกิพีเดีย. (2550). **เอแจ็กซ์**. [ออนไลน์]. ได้จาก: http://th.wikipedia.org/wiki/เอแจ็กซ์

วิชา ศิริธรรมจักร์. (2547). **Web Programming ด้วย AJAX และ PHP**. พิมพ์ครั้งที่ 1. สำนักพิมพ์ เคทีพี

ศุภชานันท์ วนภู. (2550). **การออกแบบและพัฒนาโปรแกรม Edi-Mo**. National e-Learning Conference 2007

ศุภวัฒน์ แก้วมงคล และ มารอง ผดุงสิทธิ์. (2549). **Fault-tolerant Dynamic Web Using Cache**. 4th International Joint Conference on Computer Science and Software Engineering

สยามเดเวอลอบเปอร์เน็ทเวิร์ค. (2548). **Dynamic Web Page คืออะไร**. [ออนไลน์]. ได้จาก: http://www.siamdev.net/node/76

สมพันธุ์ ชาญศิลป์.(2551) . **SUT Instant Server for Developer Plus Moodle**. Seminar on Driving Open Source Using in Software Industry 2008

อภิเศรษฐ์ ทนันชัย และ สมชาย นำประเสริฐชัย. (2545). **การพัฒนาระบบจัดเก็บข้อมูลของสควิด**, วิศวกรรมสาร มหาวิทยาลัยเกษตรศาสตร์

ภาคผนวก ก

บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการทางคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

CIT2007
การประชุมวิชาการทางคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
Conference on Computer Information Technologies 2007 (CIT2007)
เมนูหลัก
หน้าหลัก
รายละเอียดโครงการ
หัวข้อบทความวิจัย
รูปแบบการเขียนบทความวิจัย
คณะกรรมการ
Download Call For Paper
กำหนดการ
การส่งบทความวิจัย
ติดต่อสอบถาม
หน่วยงานผู้สนับสนุน
CIT2007 & WUNCA 16th
31 January - 3 February 2550
Rajamangala University of Technology Srivijaya
รายละเอียด
Clickที่นี่ค่ะ
การประชุมวิชาการทางคอมพิวเตอร์และเทคโนโลยีสารสนเทศ"
(Conference on Computer Information Technologies 2007 หรือ CIT2007)
จัดการประชุมโดย สำนักงานบริหารเทคโนโลยีสารสนเทศเพื่อพัฒนาการศึกษา (UNINET)ร่วมกับ ศูนย์อิเลคทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) และมหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย คณะวิทยาศาสตร์และเทคโนโลยีการประมง จังหวัดตรัง
รายละเอียดเพิ่มเติม
ข่าวประชาสัมพันธ์
NEW ประกาศผลการพิจารณาคัดเลือกบทความวิจัย CIT2007
NEW สำหรับเจ้าของบทความวิจัยที่ผ่านการคัดเลือกแล้ว สามารถตรวจสอบรายละเอียดที่พักและรายละเอียดการจัดงาน ได้ที่นี่ค่ะ
NEW ในวันที่ 31 ม.ค. 50 ขอเชิญผู้เข้าร่วมนำเสนอบทความวิจัย รับประกาศนียบัตร ในงานเลี้ยงอาหารค่ำ
ประกาศรายชื่อบทความวิจัย(Paper CIT2007)
ECTI-21
NECTEC
สำนักงานบริหารเทคโนโลยีสารสนเทศเพื่อพัฒนาการศึกษา(UniNet)
เลขที่ 328 ชั้น 9 อาคารสำนักงานคณะกรรมการการอุดมศึกษา
แขวง ทุ่งพญาไท เขต ราชเทวี กรุงเทพ ฯ 10400
โทร : 0-2354-5678
โทรสาร:: 0-2354-5678 ต่อ 5012

วิธีการค้นคืนข้อมูลอย่างมีประสิทธิภาพสำหรับระบบจัดการการเรียนการสอน
AN OPTIMIZED DATA RETRIEVAL APPROACH FOR LEARNING MANAGEMENT SYSTEM

ธนินทร์ ระเบียบโพธิ์, คะชา ชาญศิลป์ และ สมพันธุ์ ชาญศิลป์
สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
tanin@linux.sut.ac.th, kacha@sut.ac.th and sompan@sut.ac.th

Abstract

Learning Management System (LMS) via internet network is turn key solution to support and improve learning performance. The popular LMSs via internet network on open-source platform are Moodle and ATutor. However, its process is slow. When users request some recent data on web page, they have to wait until the server processes all data and refreshing all over the web page again. This research describes the design and development strategies by incorporating reading and writing data in Dynamic Web Cache Memory technology to increase the performance. The results are also presented.

Keywords: Learning Management System, LMS, open source, retrieval, Cache Memory

บทคัดย่อ

ระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต ได้รับการยอมรับว่าเป็นกลไกสำคัญอีกอย่างที่สนับสนุนส่งเสริม ประสิทธิภาพการเรียนการสอน ระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตที่เป็นแบบระบบเปิดที่ได้รับความนิยมในปัจจุบัน ได้แก่ Moodle และ Atutor เป็นต้น อย่างไรก็ตาม การทำงานของระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตแบบเดิมนั้น ทุกครั้งที่มีการร้องขอข้อมูลจาก เซิร์ฟเวอร์ ผู้ใช้จะต้องรอจนกว่าเซิร์ฟเวอร์จะประมวลผลข้อมูลจากฐานข้อมูลที่อยู่ในฮาร์ดดิสก์ทั้งหมดเสร็จแล้ว ข้อมูลจึงจะถูกส่งกลับมา ทำให้เกิดความล่าช้า การวิจัยชิ้นนี้ ได้ออกแบบและพัฒนาเพื่อเพิ่มประสิทธิภาพการค้นคืนข้อมูล โดยนำเทคโนโลยีในการเขียนและอ่านข้อมูลใน Cache หน่วยความจำของหน่วยประมวลผลบนเซิร์ฟเวอร์แบบพลวัต (Dynamic) มาเป็นส่วนเสริม พร้อมทั้งนำเสนอผลการทดลองดังกล่าว

คำสำคัญ ระบบจัดการการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ต, ระบบเปิด, มูเดิ้ล, แคช หน่วยความจำ, การค้นคืนข้อมูล

1. บทนำ

เว็บไซต์ในโลกมีอยู่มากมาย โดยแต่ละเว็บไซต์ก็จะมีความแตกต่างกันในการพัฒนา [1] ซึ่งมีทั้งแบบสถิตย์ (Static) คือ เว็บไซต์ที่มีเนื้อหาคงที่ไม่มีการเปลี่ยนแปลงเนื้อหาหรือรูปแบบของเว็บได้ และแบบพลวัต (Dynamic) คือเว็บไซต์ที่สามารถปรับปรุงเปลี่ยนแปลงเนื้อหาหรือรูปแบบของเว็บได้ตลอด ซึ่งในปัจจุบัน เว็บไซต์ส่วนใหญ่จะมีการพัฒนาเป็นแบบ Dynamic เพราะสามารถใช้งานโต้ตอบกับผู้ใช้ได้ทันที จึงทำให้เว็บไซต์มีความน่าสนใจมากยิ่งขึ้น ลักษณะของเว็บไซต์ที่เป็น

แบบ Dynamic นั้น ส่วนใหญ่จะมีการสร้างเว็บโดยใช้โปรแกรมภาษา PHP และฐานข้อมูล เช่น MySQL เป็นส่วนประกอบ ซึ่งเว็บไซต์ที่สร้างขึ้นมาเป็นแบบ Dynamic นี้มีการนำไปใช้งานที่หลากหลายมากขึ้น ทั้งงานในเชิงพาณิชย์ และงานทางด้านการจัดการการเรียนการสอน เป็นต้น โดยเฉพาะงานทางด้านการพัฒนาสารสนเทศเพื่อการเรียนการสอน และการพัฒนาบทเรียนคอมพิวเตอร์ผ่านเครือข่าย (Web-Base Instruction : WBI) มีบทบาทสำคัญมากต่อการพัฒนาระบบการเรียนการสอน จึงได้มีการพัฒนาซอฟต์แวร์ระบบจัดการบทเรียนแบบ โอเพนซอร์ส (Open Learning Source Management System) เช่น Moodle, Atutor, Clarolin, Learnloop, Splearn และ Vclass เป็นต้น ด้วยการใช้งานที่ง่าย และสามารถจัดการเว็บไซต์บทเรียนได้แบบเบ็ดเสร็จ ซึ่งเอื้ออำนวยให้ผู้ใช้สามารถแก้ไข ดัดแปลง และปรับปรุงเนื้อหาให้ทันสมัยอยู่เสมอ โดยไม่ต้องอาศัยผู้เชี่ยวชาญทางด้านเทคนิค ซึ่งจะช่วยแก้ไขปัญหาการขาดแคลนบุคคลากรด้านการพัฒนาเว็บไซต์ และโปรแกรมเมอร์ เพื่อพัฒนาระบบดังกล่าว ดังนั้น Moodle จึงเป็นทางออกหนึ่ง ที่เป็นเครื่องมือช่วยพัฒนาเนื้อหาบทเรียนคอมพิวเตอร์

อย่างไรก็ตาม การทำงานของระบบการเรียนการสอนผ่านเครือข่ายอินเทอร์เน็ตแบบเดิมนั้น ทุกครั้งที่มีการร้องขอข้อมูลจากเซิร์ฟเวอร์ ผู้ใช้จะต้องรอจนกว่าเซิร์ฟเวอร์จะประมวลผลข้อมูลทั้งหมดเสร็จ ข้อมูลจึงจะถูกส่งกลับมา จึงทำให้เกิดความล่าช้า ในปัจจุบันได้มีเทคโนโลยีการค้นคืนข้อมูล ที่สามารถทำให้การติดต่อรับข้อมูลจากเครื่องเซิร์ฟเวอร์ มีความเร็วเพิ่มมากขึ้น เช่น เทคโนโลยี เอแจค (AJAX: Asynchonous Javascript and XML) แต่ในการสร้างเว็บที่ใช้เทคโนโลยีดังกล่าวนั้น จะต้องอาศัยโปรแกรมเมอร์ที่มีความรู้ความสามารถมาก เพราะต้องเป็นผู้ที่เข้าใจในการเขียนโปรแกรมภาษาจาวา (Javascript Language) และ XML โดยเฉพาะอย่างยิ่งการเขียนเว็บที่ใช้เทคโนโลยีนี้ ยังต้องสร้างรูปแบบเว็บที่คงที่ หากหน้าเว็บเพจเกิดการเปลี่ยนแปลงรูปแบบไปทั้งหน้า เทคโนโลยีดังกล่าวจะค่อนข้างยากที่จะนำมาใช้ร่วม สำหรับงานวิจัยนี้ได้นำเทคโนโลยีการเขียนข้อมูลลงบนแคช (Cache) หน่วยความจำของหน่วยประมวลผลกลาง ในเครื่องเซิร์ฟเวอร์มาใช้เพื่อเพิ่มประสิทธิภาพในการค้นคืนข้อมูล ซึ่งการดึงข้อมูลจากแคชในหน่วยความจำนั้น จะทำให้ระบบไม่จำเป็นต้องไปดึงข้อมูลใหม่มาจากฐานข้อมูลทุกครั้ง จึงทำให้สามารถค้นคืนข้อมูลได้เร็วยิ่งขึ้น พร้อมทั้งนำเสนอและเปรียบเทียบผลการทดลองดังกล่าว

2. ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 หน่วยความจำที่ใช้กับ Cache

[2] แคช (Cache) คือ หน่วยความจำขนาดเล็กที่มีความเร็วสูงซึ่งเก็บข้อมูล หรือคำสั่งที่ถูกเรียกใช้บ่อยๆ ข้อมูลและคำสั่งที่เก็บอยู่ใน Cache ซึ่งใช้ SRAM (Static RAM) จะถูกดึงไปใช้งานได้เร็วกว่าการดึงข้อมูลจากหน่วยความจำหลัก (Main Memory) ซึ่งใช้ DRAM (Dynamic RAM) หลายเท่าตัว

2.1.1 การทำงานของ Cache

โปรแกรมต่าง ๆ ที่ทำงานโดยผ่านหน่วยประมวลผลกลาง (Central Processing Unit : CPU) ได้ทำการเรียกข้อมูลหรือรหัสที่ CPU จำเป็นต้องใช้ RAM Cache ซึ่งเป็นส่วนหนึ่งของ

วงจรหลักในเครื่องคอมพิวเตอร์ ได้รับสัญญาณการเรียกข้อมูลในขณะที่คำสั่งการเรียกข้อมูลกำลังเดินทางไปยังRAM และ Cache จะทำการค้นหาข้อมูลจาก RAM และส่งต่อข้อมูลไปยัง CPU ในการค้นหาข้อมูลครั้งแรกอาจจะใช้เวลานานโดยที่ตัว CPU ไม่สามารถทำงานอย่างอื่นได้ในเวลานั้น ในขั้นตอนการค้นหาข้อมูลนี้ Cache จะทำการบันทึกข้อมูลที่ค้นพบไว้ใน High-speed Memory Chips ที่มีเฉพาะภายใน Cache ในทันทีที่ Cache ตรวจสอบพบว่า CPU ได้ทำงานเสร็จสิ้นและกำลังว่างอยู่ Cache จะทำการค้นหาข้อมูลหรือรหัสของโปรแกรม ซึ่งอยู่ใกล้เคียงกับตำแหน่งของข้อมูลที่ทางโปรแกรมได้เรียกใช้ก่อนหน้านี้จาก Memory Address และจัดเก็บข้อมูลไว้ใน High-speed Memory Chips จากนั้นเมื่อโปรแกรมถามหาข้อมูลจากทาง CPU อีกครั้ง Cache จะตรวจสอบดูว่าข้อมูลที่โปรแกรมต้องการมีอยู่ใน High-speed Memory Chips แล้วหรือยัง ถ้ามีอยู่แล้ว Cache จะส่งข้อมูลไปให้ CPU ได้โดยไม่จำเป็นต้องผ่านหน่วยความจำหลัก ซึ่งมีการทำงานที่ช้ากว่ามาก ทำให้ CPU สามารถทำงานได้มีประสิทธิภาพมากขึ้น เมื่อ CPU ต้องการเปลี่ยนข้อมูลบางอย่างที่มีอยู่ใน CPU อยู่แล้ว Cache จะตรวจสอบดูก่อนว่าข้อมูลที่โปรแกรมต้องการจะเปลี่ยน มีการจัดเก็บอยู่ใน High-speed Memory Chips แล้วหรือยัง ถ้ามีอยู่แล้ว Cache จะเปรียบเทียบข้อมูลที่มีอยู่เดิมกับข้อมูลใหม่ที่เปลี่ยนไป และจะส่งข้อมูลไปเฉพาะ Memory Address ในหน่วยความจำหลัก ที่มีการเปลี่ยนแปลงข้อมูลจากข้อมูลเดิมใน High-speed Memory Chips ซึ่งจะเร็วกว่าการเปลี่ยนแปลงข้อมูลทั้งหมด [2]

2.1.2 ตัวอย่างการทำงานของ Cache

[3] เมื่อ CPU ทำการอ่านข้อมูล จะตรวจสอบว่าข้อมูลที่ต้องการเรียกใช้นั้นมีอยู่ใน Cache หรือไม่ ถ้ามี ข้อมูลก็จะถูกถ่ายทอดไปยัง CPU อย่างรวดเร็ว เพราะไม่ต้องเรียกข้อมูลจากหน่วยความจำหลัก ซึ่งจะใช้เวลานาน แต่ถ้าตรวจสอบแล้วไม่มีข้อมูลอยู่ใน Cache CPU จะดึงข้อมูลจากหน่วยความจำหลัก จากนั้นจะทำสำเนาและเก็บข้อมูลไว้ใน Cache ด้วย ถ้ามีการเรียกใช้ข้อมูลนี้อีก ข้อมูลจะถูกดึงจาก Cache โดยตรง ซึ่งการเรียกใช้ข้อมูลแบบนี้จะเรียกว่า HIT สำหรับประสิทธิภาพการทำงานของ Cache จะขึ้นอยู่กับอัตราการพบข้อมูล Cache ที่มีประสิทธิภาพ คือ Cache ที่จำข้อมูลที่ถูกเรียกใช้บ่อยได้ดี เช่น [4] บนบอร์ด 100 MHz ของ Intel ต้องใช้เวลาถึง 180 Nanosecond เพื่อนำข้อมูลจากหน่วยความจำหลัก ในขณะที่ใช้เพียง 45 Nanosecond ในการนำข้อมูลจาก Cache ด้วยประสิทธิภาพที่ไม่น่าเชื่อของ Cache ทำให้น่าจะนำ Cache มาใช้เป็นหน่วยความจำหลักแทน แต่หน่วยความจำของ Cache ใช้ SRAM ซึ่งมีราคาสูงกว่า DRAM ที่ใช้ในหน่วยความจำหลัก ถึงประมาณ 6 เท่า ฉะนั้นทำให้การนำ Cache มาใช้แทนหน่วยความจำหลักนั้นไม่คุ้มค่าเพราะทำให้สิ้นเปลืองสูง โดยอาจจะเปรียบเทียบได้กับ การลงทุนทำร้านขายของชำเพื่อที่จะเก็บสินค้าทุกประเภท ซึ่งเป็นการลงทุนที่ไม่คุ้มค่า แต่การนำวิธีการค้นคืนข้อมูลโดยใช้การเก็บและดึงข้อมูลที่ถูกเรียกใช้งานบ่อยๆจาก Cache มาใช้ จะเป็นวิธีหนึ่งที่เหมาะสม และสามารถค้นคืนประสิทธิภาพการค้นคืนข้อมูลได้อย่างดีเยี่ยม

2.2 งานวิจัยที่เกี่ยวข้อง

ปัจจุบัน งานวิจัยที่เกี่ยวกับการเพิ่มประสิทธิภาพการค้นคืนข้อมูลระหว่างเครื่องไคลเอนต์กับเครื่องเซิร์ฟเวอร์ ให้สามารถทำงานร่วมกันได้เร็วขึ้นนั้น ได้รับความสนใจจากนักวิจัยอย่างกว้างขวาง [1] Arun Iyengar และ Jim Challenger ได้เสนอแนวความคิดในการออกแบบระบบที่ใช้ในการค้นคืนข้อมูลจากเว็บที่เขียนแบบ Dynamic โดยได้เสนอการค้นคืนข้อมูลจากระบบ Cache ของ CPU และได้ทำการออกแบบการทดลองระบบดังกล่าว พร้อมทั้งนำเสนอข้อมูล โดยระบบที่ได้ทำการทดสอบนี้สามารถเพิ่มประสิทธิภาพในการค้นคืนข้อมูลได้ดียิ่งขึ้น

[5] Arun Iyengar, Jim Challenger, Daniel Dias, และ Paul Dantzig นำเสนอแนวความคิดในการพัฒนาอัลกอริธึมใหม่ที่เรียกว่า Data Update Propagation (DUP) เพื่อตรวจสอบว่าหน้าเว็บเพจของ Cache ใดเก่าแล้วจากข้อมูลใหม่โดย DUP จะพิจารณาว่าหน้าเว็บเพจของ Cache ใดมีการเปลี่ยนแปลง ถ้ากลุ่มของหน้าเว็บเพจของ Cache ถูกสร้างขึ้นจาก Table ที่อยู่ในฐานข้อมูลของ Cache นั้น จะต้องเหมือนกันกับข้อมูลในฐานข้อมูลจึงจะถือว่าข้อมูลหน้านั้นไม่มีการเปลี่ยนแปลง ซึ่งสามารถทำให้ระบบทำงานได้ดีขึ้น

[6] Yein-Kuan Chang, Yu-Ren Lin และ Yi-Wei Ting ได้เสนอวิธีการค้นคืนข้อมูลในระบบ Cache โดยได้ทำการทดลองกับ Tomcat Web Server และเว็บเพจที่เป็นแบบชนิด Dynamic โดยโปรแกรมภาษา JSP (Java Servlet Pages) และยังได้เสนอขั้นตอนการออกแบบระบบ และขั้นตอนการทำงานดังกล่าว โดยระบุว่าระบบที่ได้ทำการออกแบบนั้น สามารถช่วยปรับปรุงระบบให้มีประสิทธิภาพดีขึ้นได้ถึง 290%

3. วิธีดำเนินการวิจัย

ปัจจุบันนี้ ลักษณะการทำงานแบบไคลเอนต์ - เซิร์ฟเวอร์ เริ่มถูกนำมาใช้งานอย่างแพร่หลาย ในลักษณะการติดต่อ สื่อสารผ่านทาง Web Browser ซึ่งการทำงานแบบนี้ จะมีการทำงานโดยไคลเอนต์ จะร้องขอข้อมูลบางอย่างจากเซิร์ฟเวอร์ ดังนั้นการโหลดและการรีเฟรชหน้าจอจึงเป็นสิ่งที่หลีกเลี่ยงไม่ได้ เป็นผลให้ผู้ใช้งานทางฝั่งไคลเอนต์ต้องหยุดรอการโหลดและการรีเฟรชหน้าจอ ซึ่งถือว่าเป็นการทำงานที่ไม่มีประสิทธิภาพ

3.1 ทดสอบระบบการทำงานแบบเดิม

ในการทดสอบระบบแบบเดิมนั้น ได้ทำการออกแบบโปรแกรมที่สามารถตรวจสอบได้ว่าทุกครั้งที่มีการร้องขอข้อมูลจากเครื่องไคลเอนต์นั้น เซิร์ฟเวอร์จะทำการประมวล ผลข้อมูลจากฐานข้อมูลใหม่ทุกครั้ง แล้วถึงจะส่งข้อมูลออกมาให้กับเครื่องไคลเอนต์ ซึ่งโปรแกรมสามารถจับเวลาในการส่งข้อมูลแต่ละครั้ง เพื่อใช้เป็นข้อมูลสำหรับการวิเคราะห์ต่อไป

3.2 ทดสอบระบบการทำงานแบบใหม่

ส่วนการทดสอบระบบแบบใหม่ ได้ทำการออกแบบโปรแกรมที่สามารถตรวจสอบ การร้องขอข้อมูลจากเครื่องไคลเอนต์ เช่นเดียวกันกับการทดสอบระบบการทำงานแบบเดิม แต่จะมีความแตกต่างกันตรงที่ ข้อมูลที่ถูกส่งกลับมาใหม่ทุกครั้งนั้น ไม่ได้ไปดึงมาจากฐานข้อมูลโดยตรง แต่จะไปดึงข้อมูลจาก Cache ของหน่วยประมวลผลกลางที่อยู่บนเครื่องเซิร์ฟเวอร์แทน โดยข้อมูลเหล่านั้นจะถูกเก็บในรูปแบบของ Text File ซึ่งสามารถจับเวลา

ในการส่งแต่ละครั้งได้ออกมาเป็นข้อมูล เพื่อการวิเคราะห์เปรียบเทียบประสิทธิภาพของระบบทั้ง 2 ระบบต่อไป

3.3 วิธีการทดลอง

งานวิจัยนี้จะทำการทดสอบระบบการทำงานแบบเดิมกับระบบการทำงานแบบใหม่ ที่ได้ประยุกต์ใช้วิธีการดึงข้อมูลจาก Cache ของหน่วยความจำ โดยการทดสอบการค้นคืนข้อมูลร่วมกับเว็บการสอบออนไลน์ ที่ได้จำลองขึ้นสำหรับงานวิจัยนี้โดยเฉพาะ ในการทดลองนี้ได้ทำการทดสอบเปรียบเทียบประสิทธิภาพการค้นคืนข้อมูลระหว่างเครื่องไคลเอนต์และเซิร์ฟเวอร์ ซึ่งรายละเอียดของเครื่องเซิร์ฟเวอร์และชุดทดลอง มีดังต่อไปนี้

ตารางที่ 1 รายละเอียดชุดทดลองที่ 1

Intel Celeron CPU 2.40 GHz 130 nm	
Cache Type	L1 8 KB, L2 128 KB
Memory	DDR 512 MB
OS	SUT-MEU 1.1 (Linux)
Cache Size	128 KB
Number of Simulated Click	150
Number of Clients	20

ตารางที่ 2 รายละเอียดชุดทดลองที่ 2

Intel Pentium 4 CPU 2.80 GHz	
Cache Type	L1 8 KB, L2 512 KB 8-way set, 64 B/line
Memory	DDR 256 MB
OS	SUT-MEU 1.1 (Linux)
Cache Size	512 KB
Number of Simulated Click	150
Number of Clients	20

4. ผลการวิเคราะห์ข้อมูลและการอภิปรายผล

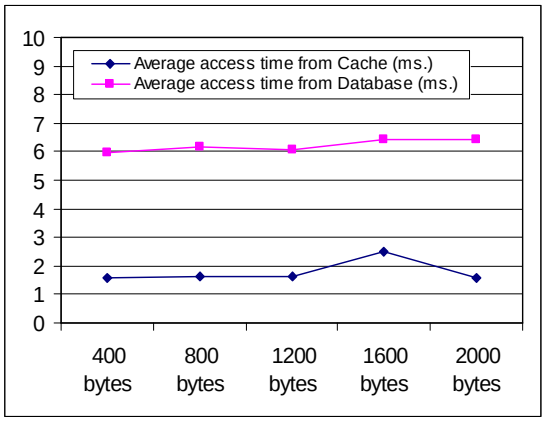
จากการทดลอง และได้ทำการวิจัยทดสอบระบบดังกล่าวแล้วนั้น จะเห็นได้ว่าข้อมูลที่เครื่องไคลเอนต์มีการร้องขอไปยังเซิร์ฟเวอร์ ในระบบแบบเดิมจะต้องมีการค้นคืนข้อมูลที่เก็บอยู่ในฐานข้อมูล ซึ่งเมื่อเปรียบเทียบกับระบบใหม่ที่ได้ทำการออกแบบ โดยอาศัยเทคโนโลยีการเก็บข้อมูลใน Cache ของหน่วยความจำของเครื่องเซิร์ฟเวอร์ ซึ่งจากทั้ง 2 การทดลอง ดังตารางที่ 3 และตารางที่ 4 นั้น เมื่อนำมาทำการเปรียบเทียบผลดังกล่าว (ดังรูปที่ 1 และรูปที่ 2) จะเห็นได้ว่าการค้นคืนข้อมูลจากฐานข้อมูล จะใช้เวลามากกว่าการค้นคืนข้อมูลจาก Cache ถึง 3.5 เท่า ทำให้ได้ระบบมีประสิทธิภาพที่ดียิ่งขึ้น

ตารางที่ 3 การเข้าถึงข้อมูลที่มีขนาด Cache size เท่ากับ 128 KB

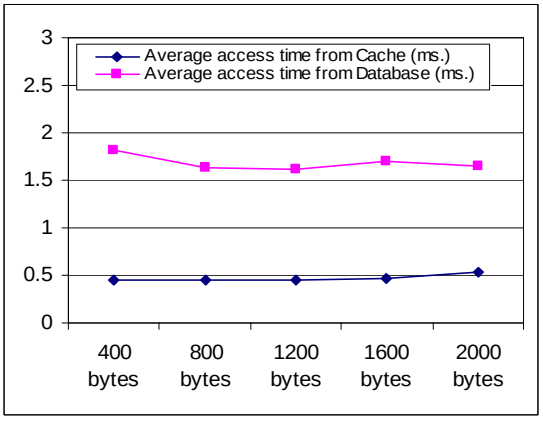
File size (bytes)	Average access time from Cache (ms.)	Average access time from Database (ms.)
400	1.56388712	5.94800991
800	1.6118571	6.19116917
1200	1.64641825	6.08104062
1600	2.50772277	6.42845932
2000	1.59610004	6.41205416

ตารางที่ 4 การเข้าถึงข้อมูลที่มีขนาด Cache size เท่ากับ 512 KB

File size (bytes)	Average access time from Cache (ms.)	Average access time from Database (ms.)
400	0.44471494	1.81189888
800	0.4526618	1.63058406
1200	0.45688939	1.61358984
1600	0.46310366	1.70644158
2000	0.53137386	1.64867937



รูปที่ 1 แสดงการเปรียบเทียบการเข้าถึงข้อมูลของชุดทดลองที่ 1



รูปที่ 2 แสดงการเปรียบเทียบการเข้าถึงข้อมูลของชุดทดลองที่ 2

5. บทสรุป

จากผลการทดลองดังกล่าว จะเห็นได้ว่าเมื่อมีการนำเทคโนโลยีการค้นคืนโดยนำวิธีการเก็บข้อมูลในรูปแบบของ Text File ไว้ที่ Cache หน่วยความจำของหน่วยประมวลผล จะช่วยให้สามารถเพิ่มประสิทธิภาพให้กับระบบการค้นคืนข้อมูลระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์ได้ถึง 3.5 เท่า แต่อย่างไรก็ตาม เทคโนโลยีต่างๆ ทางด้านสารสนเทศในปัจจุบัน มีการพัฒนาเปลี่ยนแปลงไปอย่างรวดเร็ว จึงจำเป็นที่จะต้องมีการคิดค้นวิธีการใหม่ๆ ตลอดเวลา เพื่อที่จะนำมาช่วยเพิ่มประสิทธิภาพของระบบให้ดียิ่งขึ้น

6. เอกสารอ้างอิง

[1] Arun Iyengar and Jim Challenger, *"Improving Web Server Performance by Caching Dynamic Data"*, Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, California,1997.

[2] ชยา ลิมจิตติ, *"มารู้จักแคชกันเถอะ"*, บิซิเนส คอมพิวเตอร์แมกกาซีน ปีที่ 6 ฉบับที่ 67, กันยายน 2537, หน้า 80-88.

[3] Computer Kids, *"Cache สิ่งจำเป็นสำหรับคอมพิวเตอร์ของคุณ"*, คอมพิวเตอร์กูเดย์แมกกาซีน ปีที่ 4 ฉบับที่ 43, มกราคม 2538, หน้า 97-104.

[4] "*GigaHertz Processors - Getting Bang for the Buck*" http://www.csse.monash.edu.au/~carlo/SYSTEMS/GHz-CPU-Performance-0801.htm, 2001.

[5] Arun Iyengar, Jim Challengar, Daniel Dias, and Paul Dantzig, "*High-Performance Web Site Design Techniques*", IBM T.J. Watson Research Center, IEEE Internet Computing, 2000, pp. 17-26.

[6] Yeim-Kuan Chang, Yu-Ren Lin and Yi-Wei Ting, *"Caching Personalized and Database-related Dynamic Web Pages"*, IEEE, NAS '06. International Workshop, pp. 5-9.

ภาคผนวก ข

รูปแบบของการค้นคืนข้อมูลแบบ Cache โดยภาษา PHP และการค้นคว้าเครื่องมือที่จะใช้ในการศึกษาวิจัย

รูปแบบของการค้นคืนข้อมูลแบบ Cache โดยภาษา PHP

ในการวิจัยนี้ ได้มีการเขียนโปรแกรมขึ้นมาเพื่อทดสอบการทำงานโดยวิธีการ การ Caching Output ของเว็บเซิร์ฟเวอร์ด้วยโปรแกรมภาษา PHP ซึ่งจากการศึกษาและค้นคว้าหาข้อมูลมานั้น ผู้วิจัยได้ทำการสร้างระบบการค้นคืนข้อมูลโดยวิธีการดังกล่าวขึ้นมา

การทำระบบการค้นคืนข้อมูลโดยวิธี Caching Output ในโปรแกรมภาษา PHP นั้น ในขั้นตอนแรกต้องทำการสร้างไฟล์ขึ้นมา 2 ไฟล์ โดยทำการตั้งชื่อไฟล์แรกว่า begin_caching.php ซึ่งมีรายละเอียดของ Source Code ดังต่อไปนี้

```
<?php

// Settings
$cachedir = '/tmp/cache/'; // Directory to cache files in (keep
outside web root)
$cachetime = 600; // Seconds to cache files for
$cacheext = 'cache'; // Extension to give cached files (usually
cache, htm, txt)

// Ignore List
$ignore_list = array(
'http://www.co-lms.net/elearning/rss.php',
'http://www.co-lms.net/elearning/search/',
'http://www.co-lms.net/elearning/login/'
);

// Script
$page = 'http://' . $_SERVER['HTTP_HOST'] .
$_SERVER['REQUEST_URI'].serialize($_GET).serialize($_POST); //
Requested page

$cachefile = $cachedir . md5($page) . '.' . $cacheext; // Cache
file to either load or create


$ignore_page = false;
for ($i = 0; $i < count($ignore_list); $i++) {
//$ignore_page = (strpos($page, $ignore_list[$i]) !== false) ?
true : $ignore_page;
$ignore_page = (strpos($page, $ignore_list[$i]) !== false) ?
$ignore_page : true;
}

$cachefile_created = ((@file_exists($cachefile)) and ($ignore_page
=== false)) ? @filemtime($cachefile) : 0;
@clearstatcache();
```

```
// Show file from cache if still valid
if (time() - $cachetime < $cachefile_created) {

//ob_start('ob_gzhandler');
@readfile($cachefile);

//ob_end_flush();
exit();

}

// If we're still here, we need to generate a cache file

ob_start();

?>
```

และทำการสร้างไฟล์ที่ 2 ขึ้นมา โดยตั้งชื่อว่า end_caching.php ซึ่งมีรายละเอียดของ source code ดังต่อไปนี้

```
<?php

// Now the script has run, generate a new cache file
$fp = @fopen($cachefile, 'w');

// save the contents of output buffer to the file
@fwrite($fp, ob_get_contents());
@fclose($fp);

ob_end_flush();

?>
```

เมื่อได้สร้างไฟล์ทั้งสองไฟล์ขึ้นมาเรียบร้อยแล้ว ให้ทำการบันทึกไฟล์ทั้งสองไฟล์ไปเก็บไว้ในส่วนของเซิร์ฟเวอร์ที่ทำการเก็บเว็บเพจเอาไว้ ซึ่งในการทำวิจัยนี้ได้ใช้เซิร์ฟเวอร์ลินุกซ์ในการทดสอบ ดังนั้นจึงต้องทำการบันทึกไฟล์ดังกล่าวไปเก็บไว้ที่ /var/www/html ซึ่งเป็นส่วนที่ได้มีการระบุเอาไว้ว่าให้เป็นส่วนที่มีการรันข้อมูลเว็บเพจทุกครั้งที่มีการร้องขอมาจากไคลแอนท์

จากนั้นให้กำหนดการทำงานของระบบการค้นคืนข้อมูลจากการ Caching Output ของเว็บเซิร์ฟเวอร์ด้วยโปรแกรมภาษา PHP ซึ่งในขั้นตอนนี้มีวิธีการกำหนดการทำงานของระบบโดยแบ่งออกได้เป็น 2 วิธีคือ

1) ใช้ฟังก์ชัน include() โดยทำการกำนดให้กับทุกสคริปท์ หรือทุกไฟล์ของเว็บเพจที่ต้องการให้ระบบทำงาน

2) ทำการกำหนดให้กับระบบทั้งระบบ ซึ่งเมื่อมีการร้องขอมาจากไคลแอนท์ให้ทำการเรียกใช้ระบบทุกครั้ง โดยการกำหนดดังกล่าวคือ การแก้ไขให้กับไฟล์ php.ini ซึ่งเป็นไฟล์ที่เมื่อมีการทำงานของโปรแกรมภาษา PHP ไฟล์ดังกล่าวจะทำการตรวจสอบข้อมูลหรือทำงานในการติดต่อระหว่างเซิร์ฟเวอร์และไคลแอนท์ ซึ่งเป็นตัวกลางในการแปลข้อมูลให้กับทั้งสองฝั่งนั่นเอง ข้อมูลที่ต้องทำการแก้ไขนั้นให้กำหนดในส่วนของไฟล์ begin_caching.php เริ่มทำงานทุกครั้งที่มีการร้องขอข้อมูลจากไคลแอนท์หรือที่เรียกว่า ส่วน header และในส่วนของไฟล์ end_caching.php นั้น กำหนดในส่วนสุดท้ายของการทำงานหรือที่เรียกว่า ส่วน footer นั่นเอง

ดังนั้นถ้าต้องการให้ระบบทำงานได้สมบูรณ์โดยที่ไม่ต้องทำการแก้ไขเว็บเพจทุกไฟล์ ผู้วิจัยจึงใช้วิธีในการกำหนดให้ระบบทำงานทุกครั้งเมื่อมีการร้องขอข้อมูลจากไคลแอนท์ ซึ่งต้องทำการกำหนดในไฟล์ php.ini ดังนี้

```
; Magic quotes
;

; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = Off

; Magic quotes for runtime-generated data, e.g. data from SQL,
from exec(), etc.
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ' with '' instead of \').
magic_quotes_sybase = Off

; Automatically add files before or after any PHP document.
auto_prepend_file = /var/www/html/begin_caching.php
auto_append_file = /var/www/html/end_caching.php

; As of 4.0b4, PHP always outputs a character encoding by default
in
; the Content-type: header.  To disable sending of the charset,
simply
; set it to be empty.
;
```

เมื่อทำการกำนดข้อมูลต่าง ๆ เรียบร้อยแล้ว ให้ทำการสร้างโฟวเดอร์เพื่อเก็บข้อมูลการ Caching โดยในโปรแกรมได้ทำการกำหนดให้ข้อมูลที่ได้ Cache เอาไว้ในส่วนของ /tmp/cache นั่นเอง ทำให้เมื่อมีการร้องขอข้อมูลมาจากไคลแอนท์ในทุก ๆ ครั้งจะมีการ Caching ข้อมูลดังกล่าว

ไปเก็บไว้ จากนั้นเมื่อมีผู้ใช้คนต่อไปมาร้องขอข้อมูลดังกล่าว ระบบจะทำการตรวจสอบว่ามีข้อมูลมีอยู่ใน cache หรือไม่ หากมีระบบก็จะทำการตรวจสอบอีกว่าข้อมูลดังกล่าวมีการเปลี่ยนแปลงไปหรือไม่ ถ้าข้อมูลไม่มีการเปลี่ยนแปลงไป ระบบก็จะทำการส่งข้อมูลไปให้กับผู้ใช้ที่ร้องขอข้อมูลดังกล่าวทันที แต่ถ้าหากข้อมูลที่ร้องขอมีการเปลี่ยนแปลงไป ระบบจะทำการบันทึกข้อมูลใหม่เข้าไปเก็บไว้ที่ cache และทำการส่งข้อมูลใหม่ไปให้กับผู้ใช้คนดังกล่าวทันที

การค้นคว้าเครื่องมือที่จะใช้ในการศึกษาวิจัย

จากการศึกษาค้นคว้าและหาเครื่องมือในการตรวจสอบการทำงานของเว็บเซิร์ฟเวอร์ นั้น ผู้วิจัยได้คัดเลือกเครื่องมือในการตรวจสอบการทำงานที่สอดคล้องกับความต้องการมาดังนี้

- 32Bit Server Monitor
- WebstressTools
- ExpertWebmaster
- IPCheck
- ServerMonitorpro

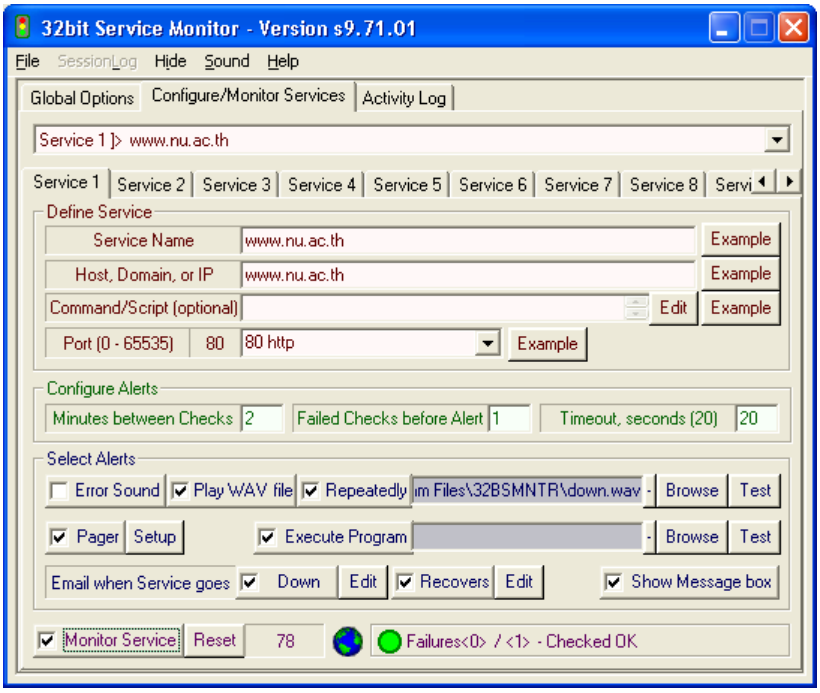
1. ดำเนินการติดตั้งโปรแกรมและทดลองใช้งาน

เมื่อได้ทำการค้นคว้าเครื่องมือแล้วและได้ดำเนินการรทดลองใช้งานและ ติดตั้งระบบดังนี้

1) ทำการติดตั้งโปรแกรม 32Bit Server Monitor
2) ทำการติดตั้งโปรแกรม WebstressTools
3) ทำการติดตั้งโปรแกรม Expert Webmaster
4) ทำการติดตั้งโปรแกรม IPCheck
5) ทำการติดตั้งโปรแกรม ServerMonitorpro
6) ทดสอบการใช้งานโปรแกรมต่าง ๆ

หลังจากที่ได้ทำการติดตั้งโปรแกรมทั้งหมดเรียบร้อยแล้ว จะเป็นการทดสอบการใช้งานโปรแกรมต่าง ๆ และบันทึกผลการทำงานของโปรแกรมเพื่อเปรียบเทียบข้อดีข้อเสียดังนี้

2. ผลการทดสอบการใช้งาน 32Bit Server Monitor



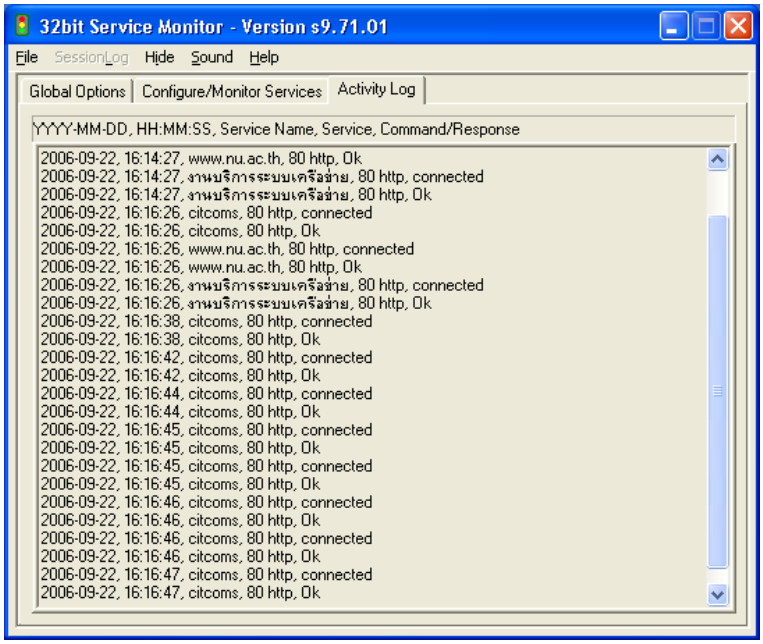
รูปที่ ข.1 แสดงหน้าจอโปรแกรม

ความสามารถของโปรแกรม และผลที่ได้

1) เครื่อง Client 1 เครื่องสามารถทำการ Monitor ได้ 20 URL
2) การทำงานของโปรแกรมทำงานแบบเรียลไทม์ ตรวจสอบตลอดเวลา
3) ตามเวลาที่ผู้ใช้ได้กำหนด
4) มี Log file ให้สามารถตรวจสอบได้ย้อนหลังได้
5) มีการแจ้งเตือน (Alerts) หลายๆรูปแบบที่สามารถทำให้ผู้ดูแลระบบทราบได้ทันที

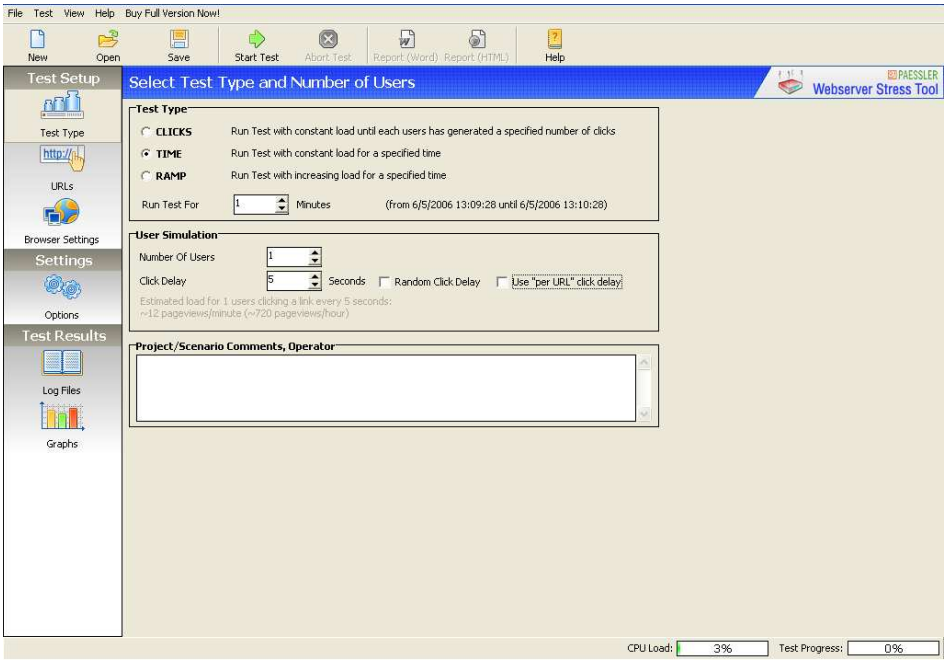
- Sound Alert เมื่อระบบตรวจสอบได้ว่า เว็บเซิร์ฟเวอร์ ไม่สามารถเข้าถึงได้หรือ หรือ Service อาจจะมีการ Respond ช้าไปโปรแกรมจะส่งเสียงเตือน (Service is down) เพื่อให้ผู้ดูแลระบบทราบและสามารถเข้าตรวจสอบได้อย่างทันที

- Email Alert Service เมื่อระบบตรวจสอบได้ว่า เว็บเซิร์ฟเวอร์ ไม่สามารถเข้าถึงได้หรือ หรือ Service อาจจะมีการ Respond ช้าไปโปรแกรมจะส่ง Email แจ้งไปยัง Email ของผู้ดูแลระบบทันที



รูปที่ ข.2 แสดง log file จากการทำงานของโปรแกรม

3. ผลการทดสอบการใช้งาน Webserver stress tool7



รูปที่ ข.3 แสดงหน้าจอโปรแกรม

ความสามารถของโปรแกรม และผลที่ได้

1) การทำงานของโปรแกรมเป็นแบบ Manual

2) บอกสถานะ การใช้ CPU และ Performance ของเครื่อง

3) สามารถทดสอบการทำงานของ user 1 คนต่อ 1 เว็บจากจำนวนการ Click

4) Ramp จะทดสอบการโหลดที่เพิ่มของเวลานั้น ๆ

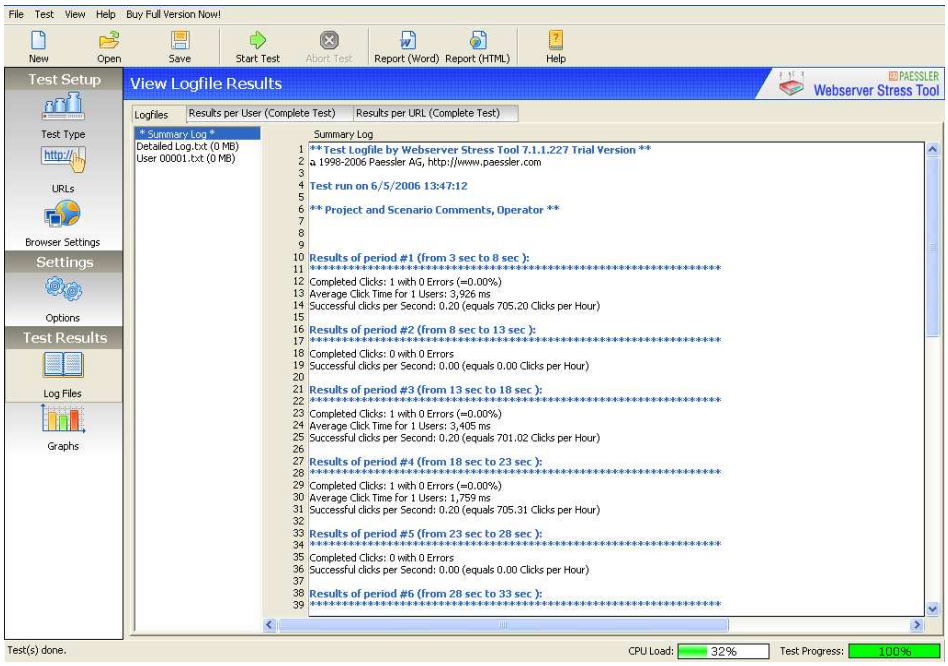
5) Time จะทดสอบการทำงานในการโหลดในช่วงเวลานั้น ๆ สามารถกำหนด Run Test กี่นาที

6) Number of Users กำหนดผู้ใช้กี่คน

7) Old Delay กำหนดเวลาย้อนกลับเป็นวินาที ในที่นี้กำหนด 5 วินาที

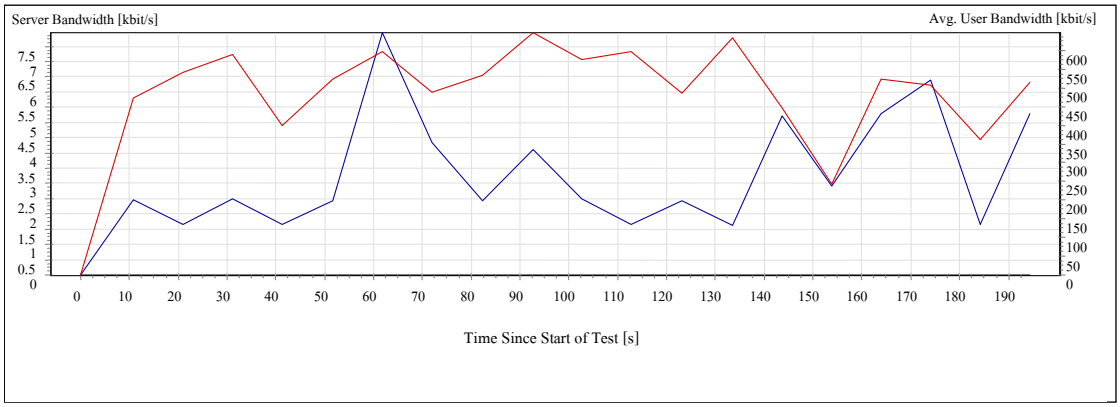
8) ค่าที่ได้จากการสั่ง Start Run จะเก็บไว้ใน Log Files และแสดงเป็น Graphs

9) เก็บเป็น Log Files ทั้งหมด



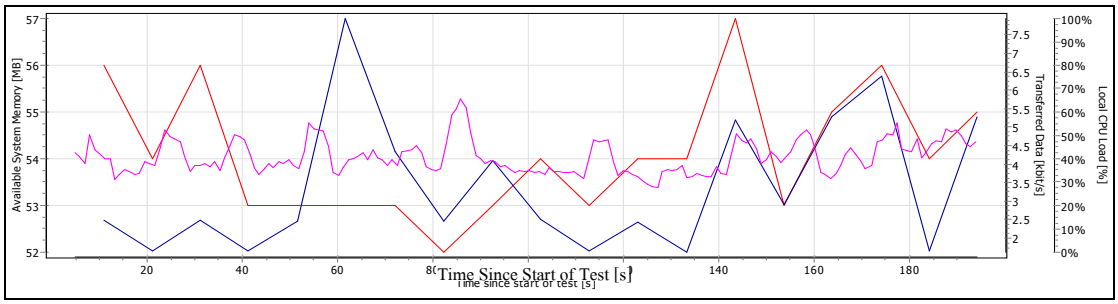
รูปที่ ข.4 แสดงแสดงผลลัพธ์ เป็น Log file

ตัวอย่างจากการตรวจสอบการทำงานของเว็บเซิร์ฟเวอร์จาก Option web server load Performance ที่ได้



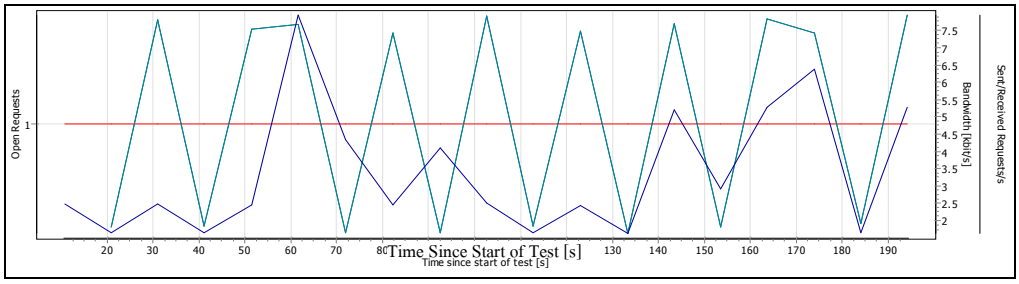
รูปที่ ข.5 แสดง Server and User Bandwidth

จากรูป เป็นอัตราการ Load การทำงานของเว็บเซิร์ฟเวอร์ต่อการเรียกใช้งานที่เพิ่มขึ้นของ User แต่การทำงานของเซิร์ฟเวอร์ก็ยังสามารถตอบสนองได้ไม่โหลดหนักมากจนเกินไปมีเพียงบางช่วงเวลา (ช่วง 60-70) จะมีอัตราการโหลดค่อนข้างสูงแต่โดยรวมแล้วยังปกติ



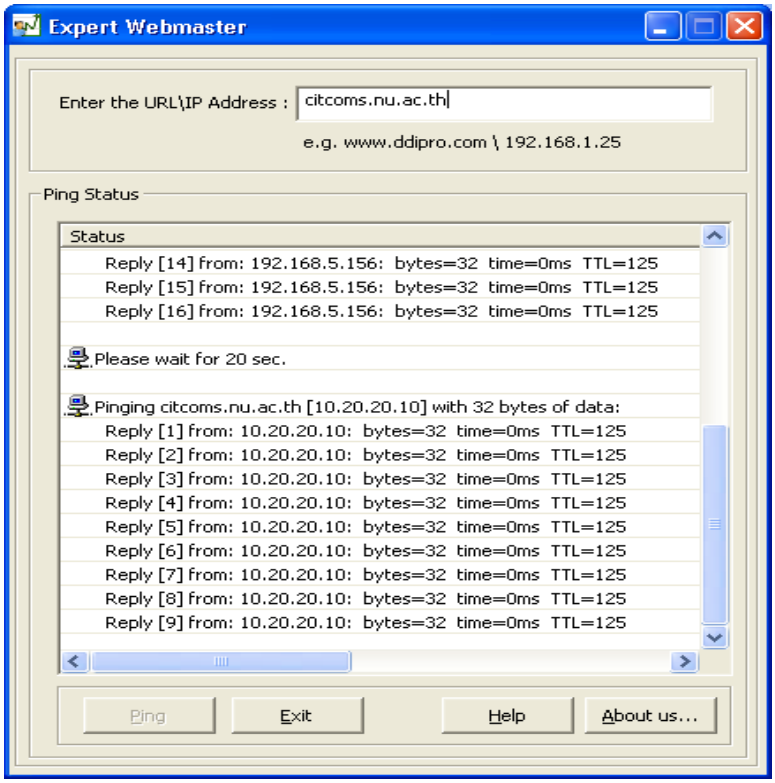
รูปที่ ข.6 แสดง Test Client Health

จากรูป แสดงอัตราการใช้ Memory และ CPU ของการรับส่งข้อมูลเมื่อ User เรียกใช้งานเว็บไซต์



รูปที่ ข.7 แสดง Open Requests and Transferred Data

4. ผลการทดสอบการใช้งานโปรแกรม Expert Webmaster



รูปที่ ข.8 แสดงโปรแกรมระหว่างการทำงาน

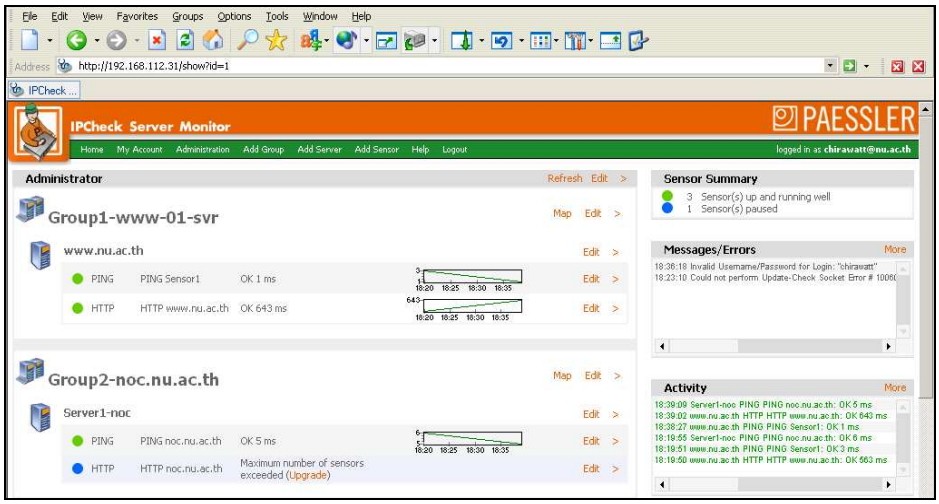
ผลการทดสอบการใช้งานโปรแกรม Expert Webmaster

หลังจากติดตั้งเสร็จแล้วก็สามารถใช้งานได้เลยโดยการพิมพ์ URL ของเว็บไซต์ที่เราต้องการตรวจสอบลงในช่อง URL/IP Address

ความสามารถของโปรแกรม และผลที่ได้

1) การทำงานของโปรแกรมทำงานแบบเรียลไทม์
2) สามารถตรวจสอบได้ทีละหลายๆ URL ไม่จำกัด
3) ไม่มี log file

5. การทดสอบการใช้งาน IPcheck Server Monitor



รูปที่ ข.9 แสดงหน้าจอโปรแกรม

ความสามารถของโปรแกรม และผลที่ได้

1) Login เข้ามาแล้วจะได้ดังรูปที่ ข.9
2) การทำงานของโปรแกรมทำงานแบบเรียลไทม์
3) มีการรายงานผลการตรวจสอบไปยัง Email ของผู้ดูแลระบบทุกวันเพื่อสามารถนำไปเป็นสถิติในการตรวจสอบสถานะ down time และ uptime ได้
4) การแสดงผลการตรวจสอบเป็นรูปแบบของ Graph ผลการจากการตรวจสอบการทำงาน

6. การทดสอบการใช้งาน PA Server Monitor Pro

รูปที่ ข.10 แสดงหน้าจอโปรแกรม

ความสามารถของโปรแกรม และผลที่ได้

1) การทำงานของโปรแกรมทำงานแบบเรียลไทม์ ตรวจสอบตลอดเวลาตามเวลาที่ผู้ใช้ได้กำหนด

2) มี Log file ให้สามารถตรวจสอบได้ย้อนหลังได้

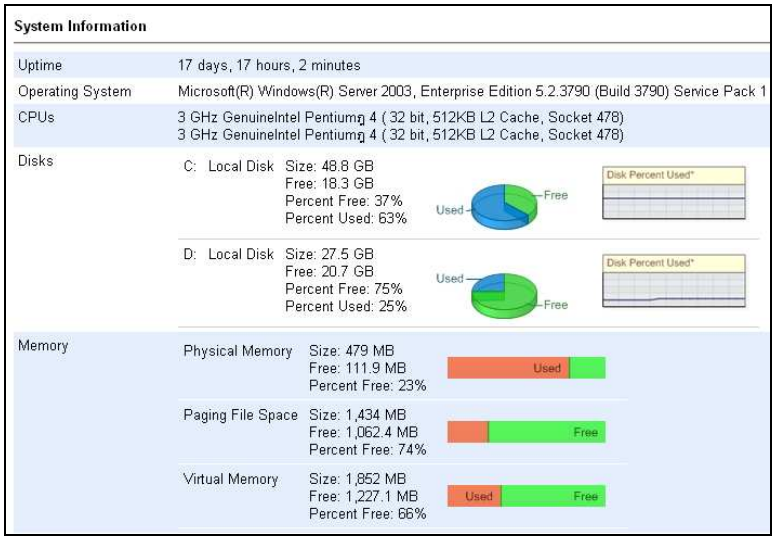
3) มีการแสดง System information ทั้งหมด ผ่าน Browser เป็นรูปแบบกราฟ และ Log file ได้

4) แสดงเวลา Uptime ทั้งหมด

5) การตรวจสอบพื้นที่ Disk spaces สามารถกำหนด Low disk space เพื่อแจ้งให้เราทราบได้

6) แสดงการใช้ Memory ของเครื่องที่ให้บริการเกี่ยวกับเซิร์ฟเวอร์ เช่น Physical Memory ,virtual Memory

7) มีการแจ้งเตือน (Alerts) ผ่านทาง e-mail



รูปที่ ข.11 แสดงผลที่ได้จากการ Monitor ของโปรแกรม

7. ทำการประเมินผลและสรุปผล

จากการทดสอบการใช้งานโปรแกรมและได้บันทึกผลการทดลองเก็บสถิติต่าง ๆ จากการรายงานผลการทำงานของโปรแกรมแล้วพบว่า แต่ละโปรแกรมมีความสามารถที่แตกต่างกันออกไปดังนี้

ตารางที่ ข.1 แสดงการเปรียบเทียบการใช้โปรแกรม

ความสามารถ	**โปรแกรมที่ใช้**				
	ติดตั้งเครื่อง Client			ติดตั้งที่เครื่อง Server	
	32BitService	Expert Webmaster	Webstress Tools	IPCheck	Server Monitorpro
Realtime	/	/	/	/	/
e-mail Alert	/			/	/
Sound Alert	/				
log file	/	/	/	/	/
แสดงผลแบบกราฟ			/	/	/
check Disk space					/
Memory Use			/		/

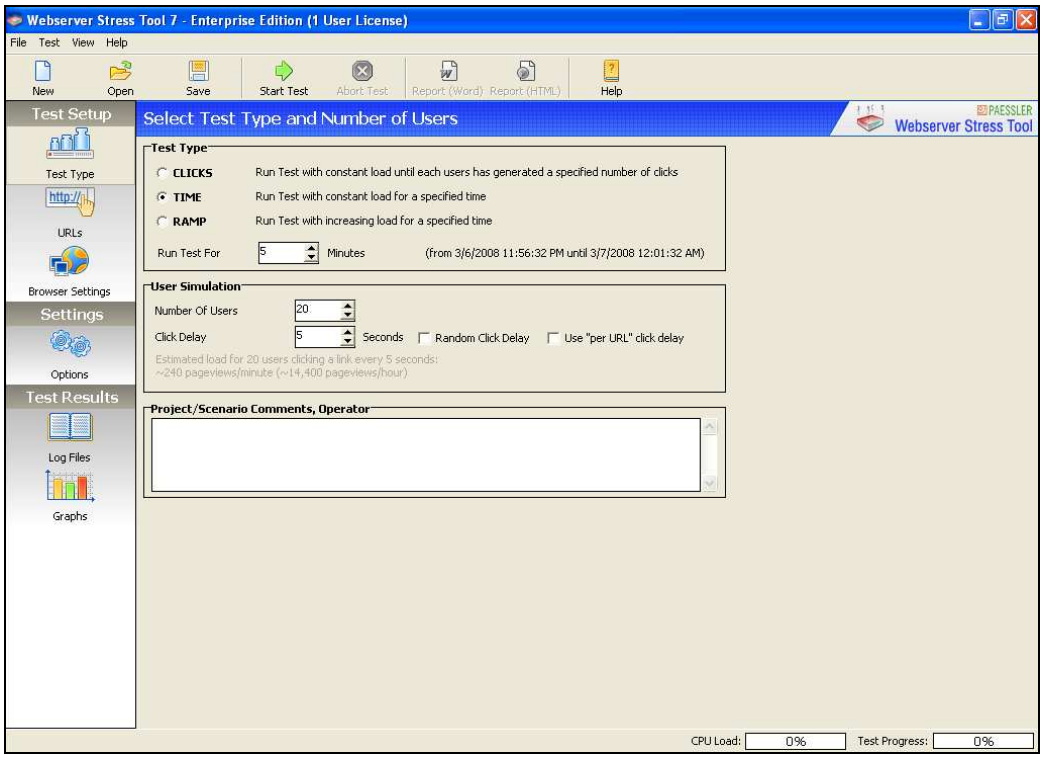
จากตารางรายละเอียดความสามารถของแต่ละโปรแกรม และจากข้อมูลผลการทดสอบการใช้งานโปรแกรมต่าง ๆ ในขั้นต้นนั้น ผู้วิจัยขอสรุปและเลือก Tool ที่มีความสามารถทั้งในส่วนของโปรแกรมที่ใช้ติดตั้งที่เครื่องไคลเอนท์และทำงานบนเครื่องเซิร์ฟเวอร์ได้อย่างมีประสิทธิภาพ คือ โปรแกรม Webserver Stress Tool ซึ่งมีคุณสมบัติที่จะช่วยให้ผู้ดูแลระบบได้ทราบถึงสถานะการทำงานของเว็บเซิร์ฟเวอร์ได้ และเพื่อช่วยเป็นเครื่องมือในการตรวจสอบการทำงานของเว็บเซิร์ฟเวอร์ เพื่อให้ผู้ดูแลระบบได้มีข้อมูลการทำงานของเครื่องให้บริการเว็บเซิร์ฟเวอร์ และจะได้เป็นข้อมูลในการนำเสนอต่อไป

ภาคผนวก ค

วิธีการใช้งานโปรแกรม Webserver Stress Tool

วิธีการใช้งานโปรแกรม Webserver Stress Tool 7

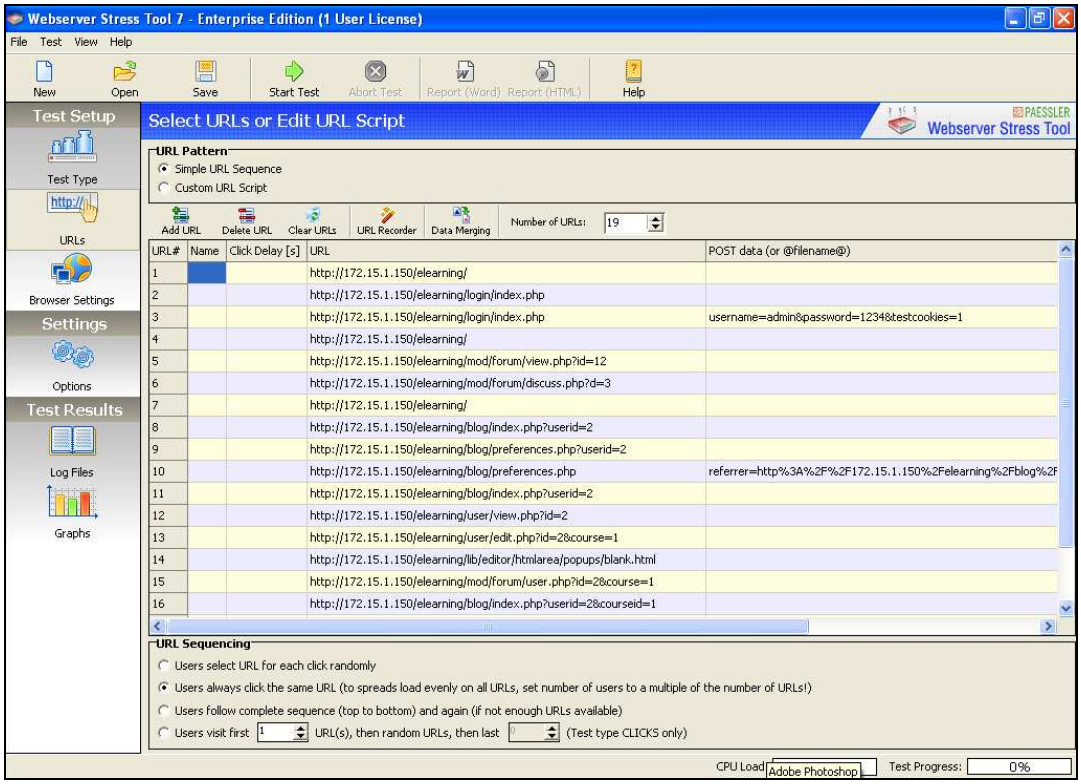
เลือก File -> New Scenario ก็จะปรากฏหน้าจอดังรูป



รูปที่ ค.1 แสดงหน้าต่างโปรแกรม Webserver Stress Tool

Option การทดสอบจะมีให้เลือก 3 รูปแบบด้วยกันคือ

- Clicks จะทดสอบการทำงานของผู้ใช้ 1 คนต่อ 1 เว็บจากจำนวนการ Click
- Ramp จะทดสอบการโหลดที่เพิ่มของเวลานั้น ๆ
- Timeจะทดสอบการทำงานในการโหลดในช่วงเวลานั้น ๆ สามารถกำหนดให้ Run Test กี่นาที
- กำหนดชื่อ URLs ที่จะใช้ในการทดสอบ
- ตัวอย่างที่ใช้ทดสอบคือ http://clms.sut.ac.th/elearning และยังสามารถทดสอบเว็บไซต์หลาย ๆ เว็บไปพร้อมกันได้

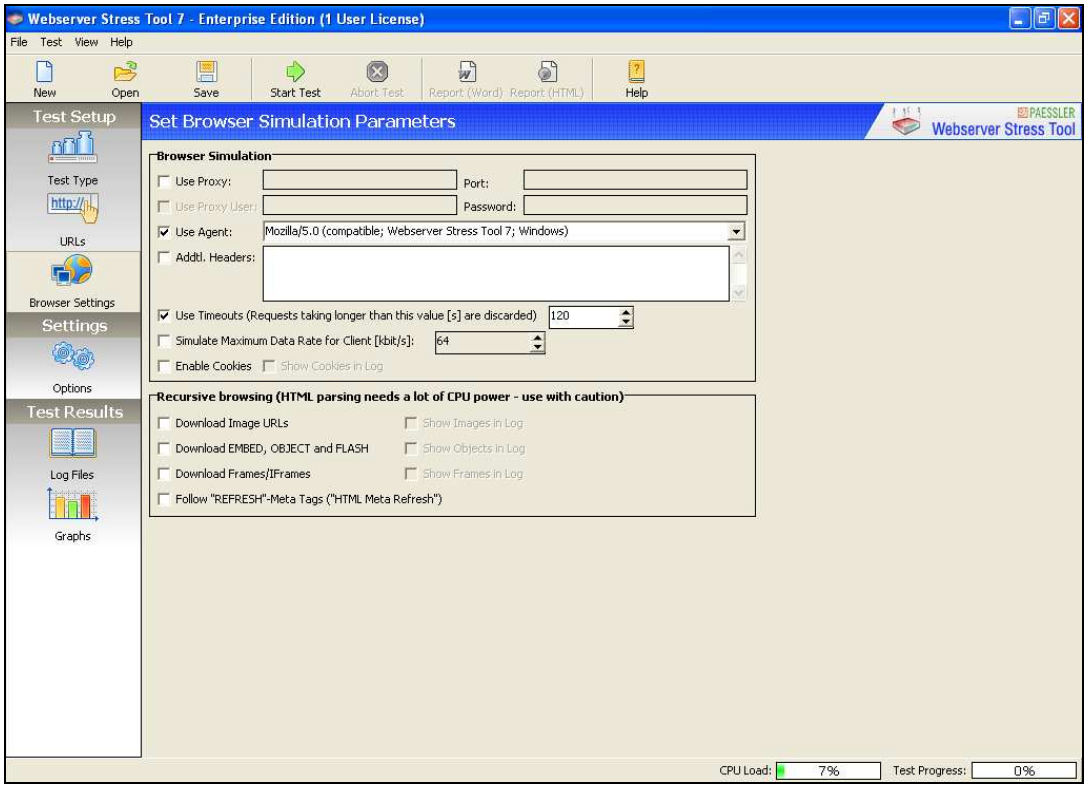


รูปที่ ค.2 หน้าต่างโปรแกรมส่วนของ URL Platern

กำหนดค่า Browser Settings

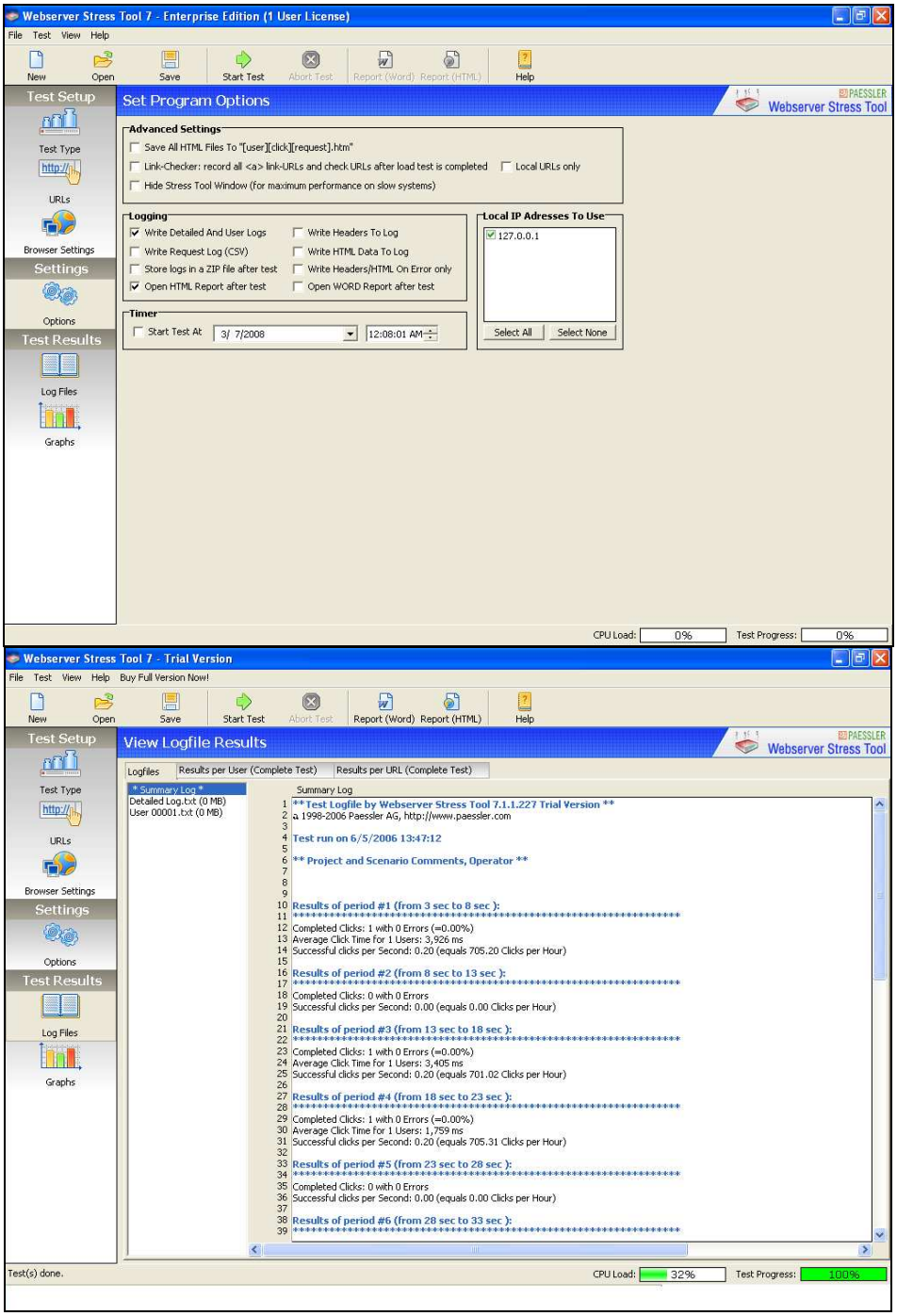
Use proxy : ใส่ตามต้องการ

Use proxy User : Username & Password



รูปที่ ค.3 หน้าต่างโปรแกรมส่วนของ Browser Simulation

Default ที่กำหนดไว้แล้วคือ write details and user logs กับ open HTML Report after test และ Local IP Addresses ที่ใช้ในการทดสอบ คือ 192.168.150 และเมื่อทำการกำหนดค่าต่าง ๆ เสร็จแล้ว ให้สั่ง Start Test จะแสดงผลลัพธ์ออกมาเป็นหน้า html



รูปที่ ค.4 แสดงการกำหนดค่า Program Option และแสดงผลลัพธ์ เป็น Log และ html file

ประวัติผู้เขียน

นายธนินทร์ ระเบียบโพธิ์ เกิดเมื่อวันที่ 25 เมษายน พ.ศ. 2522 ที่จังหวัดนครราชสีมา เข้าศึกษาในระดับประถมศึกษาที่โรงเรียนบึงพญาปราบ ชั้นมัธยมศึกษาที่โรงเรียนราชสีมาวิทยาลัย และเข้าศึกษาในระดับปริญญาตรีในปีการศึกษา 2540 ที่สาขาวิศวกรรมไฟฟ้า-โทรคมนาคม มหาวิทยาลัยเทคโนโลยีราชมงคล อีสาน (ชื่อเดิมคือสถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคตะวันออกเฉียงเหนือ นครราชสีมา) และสำเร็จการศึกษาเมื่อปี พ.ศ. 2544 ภายหลังสำเร็จการศึกษาได้เข้ารับการทำงานในตำแหน่งอาจารย์อัตราจ้าง ในแผนกวิชาเทคนิคคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคล อีสาน ด้วยความสนใจระบบงานทางด้านเว็บไซต์และเน็ตเวิร์คเป็นพิเศษ จึงทำให้เกิดแรงจูงใจที่จะศึกษาต่อในระดับที่สูงขึ้น โดยได้เข้าศึกษาต่อในระดับปริญญาโทที่สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีการศึกษา 2548

ในระหว่างการศึกษาได้รับความอนุเคราะห์อย่างดียิ่งจากคณาจารย์ในสาขาวิชา โดยได้รับความไว้วางใจให้เป็นผู้ช่วยวิจัยและผู้สอนปฏิบัติการของสาขาวิชาวิศวกรรมคอมพิวเตอร์ จำนวน 5 รายวิชา คือ (1) วิชา Web Application (2) วิชา Computer Security (3) วิชา Computer Network (4) วิชา Operating System และ (5) วิชา Computer Programming เป็นระยะเวลา 3 ปี