การวัดและวิเคราะห์สมรรถภาพของโปรแกรมสควิดในระดับมอดูล

นางสาวนุชจรินทร์ อาจปรุ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2545

ISBN 974-17-1194-8

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

THE PERFORMANCE MEASUREMENT AND ANALYSIS OF SQUID AT MODULE-LEVEL

Miss. Nuchjarin Oajpru

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 2002
ISBN 974-17-1194-8

หัวข้อวิทยานิพนธ์	การวัดและวิเคราะห์สมรรถภาพของโปรแกรมสควิดในระดับมอดูล
โดย	นางสาวนุชจรินทร์ อาจปรุ
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

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

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

.....กรรมการ
(อาจารย์ จารุมาตร ปิ่นทอง)

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

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นุชจรินทร์ อาจปรุ : การวัดและวิเคราะห์สมรรถภาพของโปรแกรมสควิดในระดับมอดูล
(THE PERFORMANCE MEASUREMENT AND ANALYSIS OF SQUID AT MODULE-LEVEL)
อ.ที่ปรึกษา : อ.ดร.ณัฐวุฒิ หนูไพโรจน์, 104 หน้า. ISBN 974-17-1194 -8.

เนื่องจากความเจริญเติบโตของเครือข่ายอินเทอร์เน็ตที่เป็นไปอย่างรวดเร็ว เป็นผลให้ผู้ใช้มีจำนวนเพิ่มมากขึ้น ทำให้เครือข่ายต้องประสบกับปัญหาต่าง ๆ เช่นเว็บเซิร์ฟเวอร์มีขนาดเล็กไม่สามารถรองรับการร้องขอจากผู้ใช้จำนวนมากในเวลาเดียวกันได้ จึงได้มีการประยุกต์แนวความคิดของแคชมาช่วย หลักการของแคช คือ การจัดเก็บเอกสารที่ถูกเรียกใช้บ่อย ๆ หรือเป็นที่นิยมเอามาไว้ให้ใกล้กับผู้ใช้มากที่สุด จึงทำให้มีการพัฒนาซอฟต์แวร์แคชเกิดขึ้น มีทั้งที่เป็นฟรีแวร์ และซอฟต์แวร์ทางการค้า โปรแกรมสควิด เป็นโปรแกรมฟรีแวร์ทางด้านพร็อกซีเซิร์ฟเวอร์ตัวหนึ่งที่ได้รับความนิยมนำมาใช้กันอย่างแพร่หลายในหน่วยงานต่าง ๆ เพื่อช่วยเพิ่มประสิทธิภาพการให้บริการแก่ผู้ใช้ในการร้องขอเอกสารจากเครือข่ายอินเทอร์เน็ต แต่จากการทำเบนช์มาร์กของเบคออฟ พบว่าประสิทธิภาพการทำงานของสควิดทำได้ดีที่ 80 คำร้องขอต่อวินาที ถ้าหากมีการร้องขอในอัตราที่สูงกว่านั้นประสิทธิภาพการทำงานจะลดลงเรื่อย ๆ โดยที่ระยะเวลาในการตอบสนองเพิ่มมากขึ้น และอัตราการฮิตลดลง

การวิจัยนี้มีจุดประสงค์เพื่อศึกษาสมรรถภาพการทำงานของโปรแกรมสควิดในระดับมอดูลภายใต้สภาวะการร้องขอที่ต่างกัน ผลที่ได้จะนำไปเสนอแนะแนวทางในการปรับปรุงการใช้งานโปรแกรมสควิดให้มีประสิทธิภาพการทำงานที่ดีขึ้น โดยจะใช้หลักการในการพิจารณาปรับปรุงส่วนที่ใช้เวลาทำงานมากที่สุดก่อน เพื่อให้การปรับปรุงได้ประโยชน์สูงสุด ซึ่งจากการศึกษาและวิจัยพบว่าเวลาที่ใช้ในส่วนต่าง ๆ ของการทำงานของสควิดมีผลน้อยมากต่อเวลาการตอบสนองที่เพิ่มขึ้น เนื่องจากเวลาที่ใช้ไปส่วนใหญ่เกิดจากเวลาในส่วนการทำงานของเครือข่าย ดังนั้นในการปรับปรุงสมรรถภาพการทำงานของสควิดควรจะปรับปรุงสภาพการทำงานของเครือข่ายเป็นลำดับแรก และพิจารณาขนาดของแคชที่ใช้เก็บข้อมูล โดยให้เหมาะสมตามสภาพการใช้งานของเครือข่ายในแต่ละองค์กร และเมื่อพิจารณาปรับปรุงการทำงานในระดับมอดูลของสควิด ควรปรับปรุงส่วนการจัดการเครือข่ายเป็นลำดับแรก ส่วนการติดต่อเซิร์ฟเวอร์เป็นลำดับที่สอง ส่วนการจัดเก็บข้อมูลเป็นลำดับที่สามและส่วนการติดต่อไคลเอนต์เป็นลำดับสุดท้าย ซึ่งจะช่วยเพิ่มสมรรถภาพการทำงานของแคชได้ดียิ่งขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต .....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์	ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2545	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

4270392821 : MAJOR COMPUTER SCIENCE

KEY WORD: WEB CACHE/PERFORMANCE/SQUID

NUCHJARIN OAJPRU: THE PERFORMANCE MEASUREMENT AND ANALYSIS OF SQUID AT MODULE-LEVEL , THESIS ADVISOR : DR.NATAWUT NUPAIROJ, 104 pp. ISBN 974-17-1194-8.

The growth of the internet has resulted in an increasing number of users accessing large amount of data from server. This has resulted to the increasing of server loads. The development of web caching is an attempt to reduce server load by allowing users to obtain objects from the cache instead of going directly to the server. Squid is a high performance web caching widely used to increase web service performance and reduce internet connection cost. The bake-off benchmark indicates that performance of squid when there are more than 80 request/Sec. Many researches have been addressed squid's performance. However, they treat squid as a black-box which prevents them to explain how the time increases.

This thesis has focused on measuring the elapsed time of the client request spending inside squid and analyzing squid at module-level. The results can be used for improving the performance of squid.The results indicate each request spends large overhead in network management modules. Thus, to improve the squid performance, the network overhead must be addressed first. The overhead in server, storage management, and client management can be optimized after that.

.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Computer Engineering Student's signature .....

Field of study Computer Science Advisor's signature .....

Academic year 2002 Co-advisor's signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยความช่วยเหลือจาก ท่านอาจารย์ที่ปรึกษา อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ ที่ได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ ต่อการทำวิจัยในครั้งนี้ด้วยดีเสมอมา ตลอดจนบรรดาเพื่อน ๆ ที่ได้ให้ความช่วยเหลือ ให้คำแนะนำต่าง ๆ พร้อมทั้งให้กำลังใจแก่ผู้วิจัยเช่นกัน และขอขอบพระคุณท่าน อาจารย์ ดร.ยรรยง เต็งอำนวย อาจารย์ จารุมาตร ปิ่นทอง อาจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา ที่ได้ให้คำแนะนำในการจัดทำวิทยานิพนธ์ชุดนี้

ท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณ บิดา-มารดา ที่ได้ให้กำลังใจและสนับสนุนทุนทรัพย์ในการศึกษาแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญภาพ ..... ฎ

บทที่

1. บทนำ 1
    1.1 ความเป็นมาและความสำคัญของปัญหา 1
    1.2 วัตถุประสงค์ของงานวิจัย 2
    1.3 ขอบเขตของงานวิจัย 2
    1.4 ขั้นตอนและวิธีดำเนินงานวิจัย 2
    1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
    1.6 โครงสร้างวิทยานิพนธ์ 3
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4
    2.1 ทฤษฎีที่เกี่ยวข้อง 4
    2.2 งานวิจัยที่เกี่ยวข้อง 7
3. ศึกษาและวิเคราะห์การทำงานของโปรแกรมสควิด 10
    3.1 ลักษณะทั่ว ๆ ไปของโปรแกรมสควิด 10
    3.2 หลักการทำงานของโปรแกรมสควิด 10
    3.3 การทำงานในส่วนการรับคำร้องขอของโปรแกรมสควิด 10
    3.4 องค์ประกอบหลักของโปรแกรมสควิดในส่วนการรับคำร้องขอของสควิด 12
    3.5 แฟ้มชุดคำสั่งโปรแกรมสควิดที่เกี่ยวข้องในแต่ละส่วน 17
    3.6 ฟังก์ชันการทำงานของสควิดในแต่ละส่วนของการให้บริการการร้องขอข้อมูลจาก
        ไคลเอนต์ 18

สารบัญ (ต่อ)

หน้า

4. การทดลองการทำงานของโปรแกรมสควิด 27
4.1 แนวความคิดในการวัดประสิทธิภาพ 27
4.2 แนวทางการทดลอง 28
5. ผลการทดลองและการวิเคราะห์ผลการทดลอง 33
5.1 ผลการทดลองและวิเคราะห์การใช้เวลาในภาพรวมกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 33
5.2 ผลการทดลองในกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) โดยพิจารณาในระดับฟังก์ชัน 39
5.3 ผลการทดลองและวิเคราะห์การใช้เวลาแต่ละส่วนในภาพรวมกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ผ่านเครือข่าย) 48
5.4 ผลการทดลองในกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย) โดยพิจารณาในระดับฟังก์ชัน 50
6. สรุปผลการวิจัยและข้อเสนอแนะ 58
6.1 สรุปผลการวิจัย 58
6.2 ข้อเสนอแนะในการปรับแต่งสควิดในระดับมอดูล 59
6.3 ปัญหาและข้อจำกัดที่ได้พบจากการวิจัย 60
6.4 ข้อเสนอแนะ 61
รายการอ้างอิง 62
ภาคผนวก 64
ภาคผนวก ก การทำงานของสควิดส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์แต่ละส่วน 65
ภาคผนวก ข การทดลองของเบคออฟ 86
ภาคผนวก ค การใช้คำสั่งต่าง ๆ ในการทดลองประสิทธิภาพของโปรแกรมสควิด 91
ภาคผนวก ง โครงสร้างข้อมูลของโปรแกรมสควิดที่ใช้จัดเก็บข้อมูล 97
ภาคผนวก จ รูปแบบของแฟ้มบันทึกเข้าออกของโปรแกรมสควิด 102
ประวัติผู้เขียน 104

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงข้อมูลที่บันทึกใน Profiling Request in Squid 8
ตารางที่ 3.1 แสดงฟังก์ชันหลักในส่วนการติดต่อไคลเอนต์ 18
ตารางที่ 3.2 แสดงฟังก์ชันหลักในส่วนการจัดเก็บข้อมูล 20
ตารางที่ 3.3 แสดงฟังก์ชันหลักในส่วนการจัดการเครือข่าย 20
ตารางที่ 3.4 แสดงฟังก์ชันหลักในส่วนการติดต่อเซิร์ฟเวอร์ 21
ตารางที่ 4.1 แสดงรายการอุปกรณ์ที่ใช้ในการทดลอง..... 29
ตารางที่ 5.1 แสดงเวลา (ไมโครวินาที) รวมในแต่ละส่วนโดยพิจารณาจากไฟล์บันทึกเข้าออกของจีพร็อบโดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)..... 33
ตารางที่ 5.2 แสดงเปอร์เซ็นต์การใช้เวลาของแต่ละส่วนที่สควิดใช้ในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)..... 34
ตารางที่ 5.3 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการจัดการเครือข่ายโดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย).....39
ตารางที่ 5.4 แสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดการเครือข่ายกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 39
ตารางที่ 5.5 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการติดต่อเซิร์ฟเวอร์โดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)... 41
ตารางที่ 5.6 แสดงเปอร์เซ็นต์การใช้เวลาในส่วนการติดต่อเซิร์ฟเวอร์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 41
ตารางที่ 5.7 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการจัดเก็บข้อมูลโดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 43
ตารางที่ 5.8 แสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดเก็บข้อมูลกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)..... 43
ตารางที่ 5.9 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการติดต่อไคลเอนต์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 45
ตารางที่ 5.10 แสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการติดต่อไคลเอนต์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 45

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 5.11 แสดงการเปรียบเทียบใช้เวลา (ไมโครวินาที) ในส่วนต่าง ๆ ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีกรณีไม่มีเครือข่ายและกรณี 49
ตารางที่ 5.12 แแสดงเวลาที่ในส่วนของโอเวอร์เฮดของเครือข่าย 57
ตารางที่ ค1 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วนที่เป็น Flat Profile ของจีพร็อบ..... 93
ตารางที่ ค2 อธิบายความหมายของข้อมูลในแฟ้มบันทึกเข้าออกส่วนที่เป็น Flat Profile 93
ตารางที่ ค3 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วน Call Graph ที่เป็น Primary Line ของจีพร็อบ 94
ตารางที่ ค4 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วน Call Graph ที่เป็น Line for a function's Callers ของจีพร็อบ 94
ตารางที่ ค5 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วน ส่วน Call Graph ที่เป็น Line for a function's Subroutines ของจีพร็อบ 95
ตารางที่ ค6 อธิบายความหมายของข้อมูลในแฟ้มบันทึกเข้าออกส่วนที่เป็น Call Graph 95

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 2.1 แสดงการทำเว็บแคชในระดับต่าง ๆ 4
รูปที่ 2.2 แสดง Squid Network Diagram ในการทดสอบสมรรถภาพสควิดของเบคออฟ 5
รูปที่ 2.3 กราฟแสดงผลการทดสอบสมรรถภาพสควิดของเบคออฟ..... 6
รูปที่ 3.1 แสดงหลักการทำงานของโปรแกรมสควิด 10
รูปที่ 3.2 แสดงขั้นตอนการทำงานของสควิดในส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์..... 11
รูปที่ 3.3 แสดงองค์ประกอบหลักของสควิดในส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์ 13
รูปที่ 3.4 แสดงองค์ประกอบหลักและส่วนที่เกี่ยวข้องในการให้บริการการร้องขอข้อมูลจากไคลเอนต์ของสควิด..... 16
รูปที่ 3.5 แผนภาพกิจกรรมแสดงการทำงานในส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์และฟังก์ชันหลักที่เกี่ยวข้องในแต่ละองค์ประกอบ 22
รูปที่ 4.1 แสดงขั้นตอนหลักในการวิจัย..... 27
รูปที่ 4.2 แสดงรูปแบบการทดลองวัดสมรรถภาพของโปรแกรมสควิด..... 28
รูปที่ 5.1 กราฟแสดงการใช้เวลาในแต่ละส่วนของสควิดในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ 35
รูปที่ 5.2 แสดงการเปรียบเทียบการใช้เวลาในภาพรวมของแต่ละส่วนของการทำงานในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที และ 100 คำร้องขอต่อวินาที 36
รูปที่ 5.3 แสดงเปอร์เซ็นต์การใช้เวลาในภาพรวมของแต่ละส่วนในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที และ 100 คำร้องขอต่อวินาที 36
รูปที่ 5.4 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนการจัดการเครือข่ายในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที 40

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 5.5 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนเซิร์ฟเวอร์ในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที 42

รูปที่ 5.6 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนการจัดเก็บข้อมูลในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที 44

รูปที่ 5.7 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนการติดต่อไคลเอนต์ในการ ให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที 46

รูปที่ 5.8 กราฟแสดงการใช้เวลาในส่วนต่าง ๆ ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) 48

รูปที่ 5.9 กราฟเปรียบเทียบการใช้เวลา (ไมโครวินาที) ในภาพรวมของส่วนต่าง ๆ ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีในกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) และกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 50

รูปที่ 5.10 กราฟแสดงเปรียบเทียบเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดการเครือข่ายที่อัตรา 10 คำร้องขอต่อวินาทีในการทดลองกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) และกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 51

รูปที่ 5.11 กราฟแสดงการใช้เวลาของฟังก์ชันในส่วนการจัดการเครือข่ายของโปรแกรมสควิดในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ในการทดลองที่อัตราการร้องขอ10 คำร้องขอต่อวินาที กรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) และกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) 51

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 5.12 กราฟแสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการติดต่อเซิร์ฟเวอร์ที่อัตรา
การร้องขอ 10 คำร้องขอต่อวินาทีในการทดลองกรณีใช้เครื่องคอมพิวเตอร์
หนึ่งเครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) และกรณีเลียนแบบการ
ทดลองของเบคออฟ (ผ่านเครือข่าย) 53
รูปที่ 5.13 กราฟเปรียบเทียบการใช้เวลาของฟังก์ชันในส่วนการติดต่อเซิร์ฟเวอร์ที่อัตรา
การร้องขอ 10 คำร้องขอต่อวินาทีในการทดลองกรณีใช้เครื่องคอมพิวเตอร์
หนึ่งเครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) และกรณีเลียนแบบการ
ทดลองของเบคออฟ (ผ่านเครือข่าย) 53
รูปที่ 5.14 กราฟแสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดเก็บข้อมูล
ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีในการทดลองกรณีใช้เครื่องคอมพิวเตอร์
หนึ่งเครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) และกรณีเลียนแบบการทดลอง
ของเบคออฟ (ผ่านเครือข่าย) 54
รูปที่ 5.15 กราฟแสดงการใช้เวลาของฟังก์ชันในส่วนการจัดเก็บข้อมูลที่อัตรา
การร้องขอ 10 คำร้องขอต่อวินาทีในการทดลองกรณีใช้เครื่องคอมพิวเตอร์หนึ่ง
เครื่องในการจำลองการทำงาน (ไม่มีเครือข่าย) และกรณีเลียนแบบการทดลอง
ของเบคออฟ (ผ่านเครือข่าย) 54
รูปที่ 5.16 แสดงจำนวนคำร้องขอที่รับ-ส่งได้ในกรณีที่สภาพการทำงานของเครือข่ายมีความเร็ว
ในการส่งข้อมูลที่ต่างกัน..... 56
รูปที่ ก1 แสดงขั้นตอนการทำงานส่วนไคลเอนต์ในการให้บริการการร้องขอข้อมูลจาก
ไคลเอนต์ของโปรแกรมสควิด 66
รูปที่ ก2 แสดงขั้นตอนในการรอรับคำร้องขอ..... 67
รูปที่ ก3 แสดงขั้นตอนในการรับคำร้องขอ 68
รูปที่ ก4 ขั้นตอนในการตรวจสอบชื่อโฮสต์โดยใช้เลขที่อยู่ 69
รูปที่ ก5 แสดงขั้นตอนในการอ่านคำร้องขอ 70
รูปที่ ก6 ขั้นตอนในการตรวจสอบคำร้องขอ 71
รูปที่ ก7 ขั้นตอนในการจัดเก็บข้อมูลที่ร้องขอ..... 72

สารบัญภาพ (ต่อ)

หน้า

รูปที่ ก8 ขั้นตอนในการตรวจสอบสิทธิของไคลเอนต์ 73
รูปที่ ก9 ขั้นตอนในการประมวลผลคำร้องขอ 74
รูปที่ ก10 ขั้นตอนในการประมวลผลคำร้องขอ (ต่อ) 75
รูปที่ ก11 ขั้นตอนในการตรวจสอบข้อมูลว่าหมดอายุหรือมีการแก้ไขหรือไม่ 76
รูปที่ ก12 ขั้นตอนในการตรวจสอบข้อมูลว่าหมดอายุหรือมีการแก้ไขหรือไม่ (ต่อ) 77
รูปที่ ก13 ขั้นตอนในการจัดการข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์ 78
รูปที่ ก14 ขั้นตอนในการจัดการข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์ (ต่อ) 79
รูปที่ ก15 ขั้นตอนในการจัดการข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์ (ต่อ) 80
รูปที่ ก16 ขั้นตอนในกรณีที่มีข้อมูลที่จะจัดส่งให้กับไคลเอนต์เพิ่มเติม 81
รูปที่ ก17 ขั้นตอนในกรณีที่มีข้อมูลที่จะจัดส่งให้กับไคลเอนต์เพิ่มเติม (ต่อ) 82
รูปที่ ก18 ขั้นตอนในการดำเนินการในกรณีที่ข้อมูลที่ร้องขอไม่พบในแคช 83
รูปที่ ก19 ขั้นตอนในการดำเนินการในกรณีที่ข้อมูลที่ร้องขอไม่พบในแคช (ต่อ) 84
รูปที่ ก20 ขั้นตอนในการดำเนินการในกรณีที่ข้อมูลที่ร้องขอไม่พบในแคช 85

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากความเจริญเติบโตของเครือข่ายอินเทอร์เน็ตที่เป็นไปอย่างรวดเร็ว เป็นผลให้ผู้ใช้มีจำนวนเพิ่มมากขึ้น ทำให้เครือข่ายต้องประสบกับปัญหาต่าง ๆ เช่นเว็บเซิร์ฟเวอร์มีขนาดเล็กไม่สามารถรองรับการร้องขอจากผู้ใช้จำนวนมากในเวลาเดียวกันได้หรือช่องสัญญาณ (Bandwidth) ที่มีขนาดจำกัดทำให้ไม่เพียงพอต่อความต้องการและจำนวนของผู้ใช้ จึงได้มีการประยุกต์แนวความคิดของแคช [1] มาช่วย หลักการของแคช คือ การจัดเก็บเอกสารที่ถูกเรียกใช้บ่อยๆหรือเป็นที่นิยมเอามาไว้ให้ใกล้กับผู้ใช้มากที่สุด ซึ่งจะช่วยลดปริมาณของข้อมูลที่รับและส่งผ่านเครือข่าย ช่วยลดภาระของเซิร์ฟเวอร์ที่ต้องให้บริการการร้องขอแก่ผู้ใช้จำนวนมากในกรณีที่มีการร้องขอเอกสารชุดเดียวกัน และยังช่วยลดเวลาในการรับส่งข้อมูลอีกด้วย

ด้วยเหตุนี้ จึงทำให้มีการพัฒนาซอฟต์แวร์แคชเกิดขึ้น มีทั้งที่เป็นฟรีแวร์ เช่น Squid [2] และซอฟต์แวร์ทางการค้า เช่น DynaCache [3], CacheFlow [4], Cisco Cache Engine [5] ฯล ซึ่งจะแตกต่างกันในส่วนของฮาร์ดแวร์ที่ใช้และราคา โดยสควิดมีชุดคำสั่งโปรแกรมแบบเปิด (Open Source) กล่าวคือบุคคลทั่วไปสามารถนำชุดคำสั่งโปรแกรม (Source Code) ของสควิดมาพัฒนาและปรับปรุงการทำงานได้ ต่างจากซอฟต์แวร์ทางการค้าซึ่งผู้ใช้ไม่สามารถนำชุดคำสั่งโปรแกรมมาพัฒนาและปรับปรุงการทำงานได้ นอกจากนี้ซอฟต์แวร์ทางการค้าส่วนใหญ่จะเลียนแบบการทำงานมาจากสควิด โดยที่แคชเหล่านี้จะมีสมรรถภาพที่แตกต่างกัน จะเห็นได้จากการทดลองเบคออฟ (Bake-Off) [6] ซึ่งเป็นการทดลองเพื่อเปรียบเทียบสมรรถภาพการทำงานของแคชรุ่นต่าง ๆ พบว่าแต่ละชนิดมีสมรรถภาพไม่เท่ากัน และพบว่าเมื่อจำนวนการร้องขอเพิ่มมากขึ้นสมรรถภาพของแคชจะตกลงเรื่อย ๆ และระยะเวลาในการตอบสนองเพิ่มมากขึ้น งานวิจัยที่เกี่ยวข้องกับสควิด [7], [9], [10] มักจะกล่าวถึงสมรรถภาพการทำงานของสควิดเฉพาะในส่วนที่เป็นผลจากสภาพการทำงานจากภายนอกที่แตกต่างกัน โดยปกติแล้วพบว่าเวลาที่สควิดให้บริการเพิ่มขึ้น เมื่อมีความต้องการของผู้ใช้มากขึ้น อย่างไรก็ดีงานวิจัยเหล่านี้ยังไม่ได้พิจารณารายละเอียดส่วนของการทำงานในมอดูลต่าง ๆ ของสควิดโดยจะพิจารณาเป็นแบบกล่องดำ (Black Box) จึงทำให้ไม่สามารถทราบได้ว่าเวลาที่เพิ่มขึ้นนั้นเกิดจากการทำงานที่จุดใด และยังไม่มีการทำงานวิจัยที่เกี่ยวกับการศึกษาสมรรถภาพการทำงานของสควิดในระดับมอดูล จึงทำให้ขาดความเข้าใจถึงสมรรถภาพการทำงานในระดับมอดูล เมื่อมีสภาวะการร้องขอที่เปลี่ยนไป

งานวิจัยนี้จะทำการศึกษาการทำงานของสควิดในระดับมอดูล โดยมุ่งเน้นในส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์ ซึ่งจะวิเคราะห์การใช้เวลา (Elapsed Time) ในมอดูลต่าง ๆ ของสควิดที่รองรับการทำงานดังกล่าว โดยจะทำการบันทึกเวลาเริ่มต้นการทำงานและเวลาเสร็จสิ้นการทำงานของมอดูลต่าง ๆ ที่เกี่ยวข้องลงในแฟ้มบันทึกเข้าออก (Log File) จากนั้นนำข้อมูลที่บันทึกได้มาวิเคราะห์หาเวลาที่ใช้ในมอดูลต่าง ๆ ของสควิด ซึ่งผลที่ได้จากการศึกษาและวิเคราะห์การทำงานในส่วนดังกล่าวจะทำให้รู้เวลาการตอบสนองที่เพิ่มขึ้นว่าเสียเวลาที่มอดูลใด ซึ่งจะเป็นข้อมูลเพื่อเป็นแนวทางช่วยในการปรับปรุงการทำงานของสควิดได้ ตัวอย่างเช่น หลังจากการศึกษาและวิเคราะห์แล้วพบว่าเวลาที่ใช้โดยส่วนใหญ่เสียเวลาตรงส่วนที่เกี่ยวกับการอ่านบันทึกไฟล์ลงดิสก์ (Disk I/O) อาจจะพิจารณาเพิ่มดิสก์เพื่อช่วยให้การทำงานเร็วขึ้น เป็นต้น

1.2 วัตถุประสงค์ของการวิจัย

เพื่อทำการวัดและวิเคราะห์เวลาที่ใช้ในมอดูลต่าง ๆ ของสควิดที่ให้บริการการร้องขอข้อมูลแก่ผู้ใช้ ซึ่งจะเปลี่ยนแปลงไปตามจำนวนการร้องขอต่อหน่วยเวลา

1.3 ขอบเขตของการวิจัย

1. ใช้โปรแกรมเว็บโพลีกราฟสร้างอัตราการร้องขอ
2. การทดสอบจะใช้อัตราการร้องขอต่อหน่อยเวลาที่แตกต่างกัน โดยจะทำการทดลองที่อัตราการร้องขอต่างกันไม่น้อยกว่า 8 ค่า
3. ใช้สควิดเวอร์ชัน 2.4 เป็นอย่างน้อย
4. ทำการทดลองเพื่อวัดสมรรถภาพของแคชเดี่ยวเท่านั้น

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1. ศึกษาการทำงานของเว็บแคช
2. ศึกษาและวิเคราะห์การทำงานของสควิด ในส่วนของการให้บริการการร้องขอข้อมูลแก่ผู้ใช้
3. ศึกษาเครื่องมือที่ใช้ทดลองการทำงานของสควิด
4. ทดสอบการทำงานของสควิดโดยใช้โครงสร้างเดียวกับเบคออฟ ประกอบด้วยผู้ให้บริการ (Server) ผู้ใช้บริการและสควิด โดยจะใช้โปรแกรมเว็บโพลีกราฟเป็นตัวสร้างอัตราการร้องขอ
5. วิเคราะห์และสรุปผล
6. จัดทำรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ทำให้ทราบเวลาการตอบสนองที่ล่าช้าในกรณีที่มีจำนวนการร้องขอเพิ่มมากขึ้นจากผู้ใช้ว่าเกิดจากการใช้เวลาในส่วนของมอดูลใด ซึ่งจะเป็นแนวทางในการปรับปรุงการทำงานของสควิดให้เร็วขึ้นได้

1.6 โครงสร้างวิทยานิพนธ์

ในบทต่อไปของวิทยานิพนธ์นี้จะกล่าวถึงทฤษฎีที่นำมาประยุกต์ใช้ และงานวิจัยที่เกี่ยวข้อง ส่วนในบทที่ 3 จะกล่าวถึงโครงสร้าง การทำงานในส่วนการให้บริการการร้องข้อมูลของผู้ใช้ของสควิด ในบทที่ 4 จะกล่าวถึงแนวคิดในการวิจัย และส่วนประกอบที่ใช้จำลองการทำงานตลอดจนรายละเอียดต่างๆ ในการทดสอบการทำงานของสควิด ในบทที่ 5 กล่าวถึงผลการทดสอบการจำลองการทำงานของสควิด และในบทสุดท้ายจะเป็นการสรุปผลของงานวิทยานิพนธ์และข้อเสนอแนะในการปรับแต่งสควิดและการพัฒนาต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีและงานวิจัยที่เกี่ยวข้องที่นำมาเป็นแนวทางในการวิจัยครั้งนี้ ได้แก่

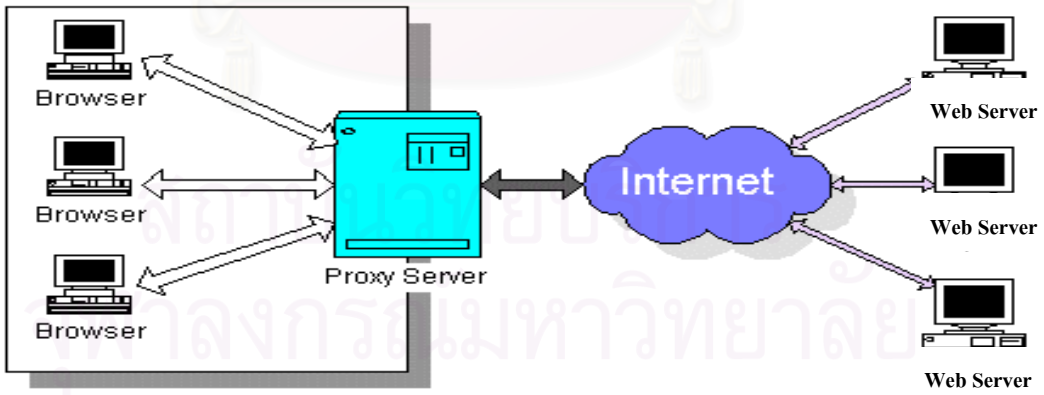
2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 เว็บแคช (Web Cache)

เว็บแคช เป็นซอฟต์แวร์หรือฮาร์ดแวร์ที่ถูกนำมาใช้ในระบบเครือข่าย เพื่อลดความคับคั่งของเครือข่าย และเพื่อลดความล่าช้าในการตอบสนองกับผู้ใช้ (Client) ของบริการเวิลด์ไวด์เว็บ (WWW) โดยจะเก็บสำเนาข้อมูลที่เป็นที่นิยม ผู้ใช้เรียกใช้บ่อยมาไว้ให้ใกล้ผู้ใช้มากที่สุดโดยไม่จำเป็นต้องไปนำเว็บเพจมาจากเซิร์ฟเวอร์ ซึ่งอาจจะอยู่ไกล จึงช่วยลดเวลาในการเดินทางของข้อมูลตลอดจนลดปริมาณของข้อมูลที่ส่งไปยังเครือข่ายอีกด้วย

การทำเว็บแคช มีหลายระดับ แสดงดังรูปที่ 2.1 ประกอบด้วย

- การทำเว็บแคชของโปรแกรมค้นผ่านเว็บ (Browser Cache)
- การทำเว็บแคชของตัวบริการแทน (Proxy Cache)
- การทำเว็บแคชของเซิร์ฟเวอร์ (Server Cache)



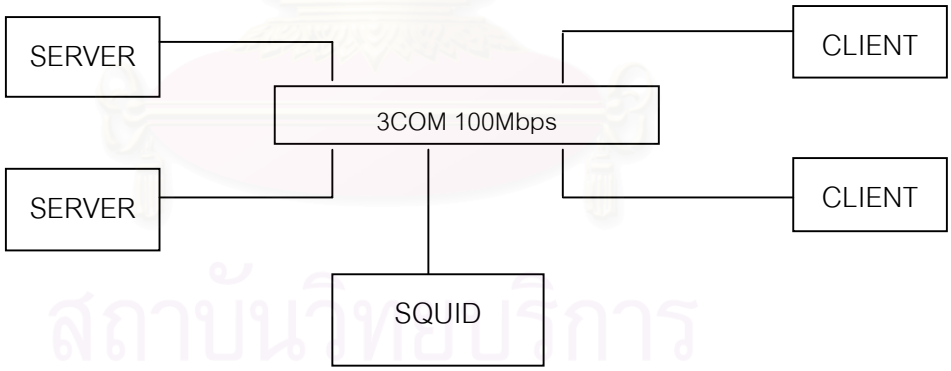
รูปที่ 2.1 แสดงการทำเว็บแคชในระดับต่าง ๆ

แคชแต่ละระดับมีหลักการทำงานที่คล้ายกัน คือ เก็บข้อมูลที่ถูกเรียกใช้บ่อย เป็นที่นิยม มาไว้ให้ใกล้กับปลายทางมากที่สุด ซึ่งช่วยลดภาระในการส่งข้อมูลสำหรับเว็บเซิร์ฟเวอร์ โดยในส่วนของเว็บแคชของโปรแกรมค้นผ่านเว็บและเว็บแคชของตัวบริการแทนจะช่วยลดปริมาณการติดต่อสื่อสารที่ออกสู่อินเทอร์เน็ต ส่วนเว็บแคชของเซิร์ฟเวอร์ช่วยลดปริมาณการติดต่อสื่อสารที่เข้ามายังเซิร์ฟเวอร์

2.1.2 เบคออฟ (Bake-off) [6]

เบคออฟ เป็นรูปแบบการทดสอบซึ่งจัดขึ้นโดยกลุ่มบุคคลที่ต้องการเปรียบเทียบสมรรถภาพการทำงานของแคชรุ่นต่าง ๆ เพื่อให้เกิดความเป็นกลางและความน่าเชื่อถือ ภายใต้สภาวะแวดล้อมและเงื่อนไขที่เหมือนกัน โดยการเข้าร่วมของบริษัทผู้ผลิตซอฟต์แวร์แคช ซึ่งเริ่มขึ้นเป็นครั้งแรกเมื่อวันที่ 8-10 พฤศจิกายน ค.ศ. 1998 สิ่งที่ใช้เปรียบเทียบสมรรถภาพการทำงานของแคชรุ่นต่าง ๆ จะดูจากปริมาณงาน (Throughput) เวลาการตอบสนอง (Response Time) อัตราการพบ (Hit Rate) เป็นต้น โดยใช้โปรแกรมเว็บโพลีกราฟ (Web Polygraph) [11] ซึ่งเป็นโปรแกรมที่พัฒนาขึ้นเพื่อทดสอบสมรรถภาพของพร็อกซีเซิร์ฟเวอร์โดยเฉพาะ สามารถจำลองการร้องขอเพื่อทดสอบสมรรถภาพของแคชได้

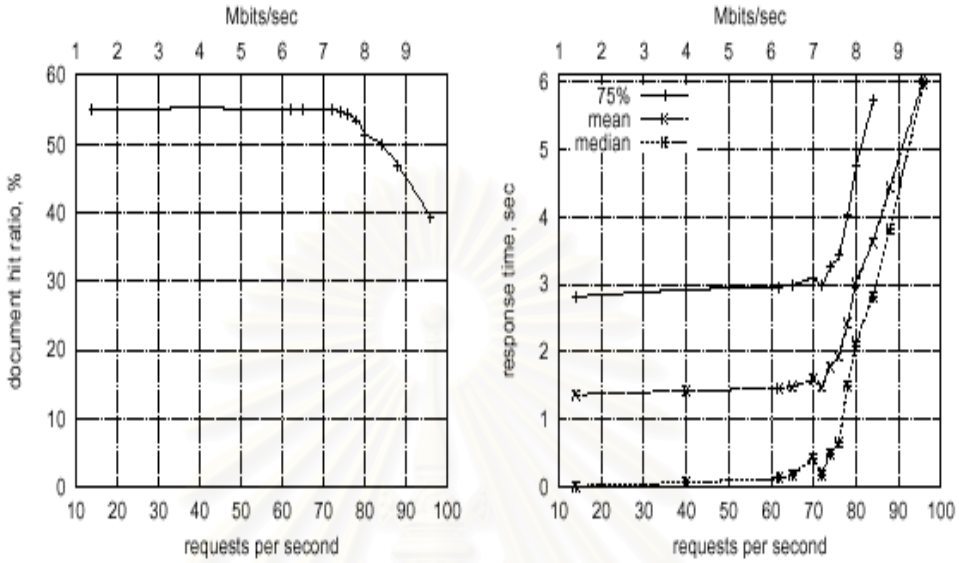
ในส่วนการทดสอบสควิด มีการจัดการทดสอบแสดงดังรูปที่ 2.2



รูปที่ 2.2 แสดง Squid Network Diagram ในการทดสอบสมรรถภาพสควิดของเบคออฟ

ในการทดสอบของเบคออฟ มีการทดสอบเว็บแคชโดยใช้อัตราการร้องขอข้อมูลจากผู้ใช้ในอัตราที่แตกต่างกันโดยมีอัตราการร้องขอเริ่มต้นที่ 14, 40, 62, 65, 70, 72, 74, 75, 76, 80, 84, 88, 96, 100, 124, 132 และ 138 คำร้องขอต่อวินาที จากผลการทดลอง พบว่าสควิดมีสมรรถภาพในการทำงานดีกว่าแคชรุ่นอื่น ๆ แต่ส่วนที่สำคัญคือส่วนที่แสดงให้เห็นถึงความสามารถใน

การรับการร้องขอข้อมูลในช่วงเวลาเดียวกัน กล่าวคือ เมื่อมีการร้องขอเข้ามามาก ทำให้แคชไม่สามารถรองรับการร้องขอจากผู้ใช้ทั้งหมดได้ เวลาที่ใช้ในการตอบสนองกับผู้ใช้จึงใช้เวลามากขึ้นกว่าเดิม ทำให้สมรรถภาพของแคชตกลง



รูปที่ 2.3 กราฟแสดงผลการทดสอบสมรรถภาพสควิดของเบคออฟ

ผลการทดสอบในส่วนของสควิด ที่ได้จากเบคออฟ แสดงดังรูปที่ 2.3 จากกราฟจะเห็นได้ว่าเมื่อการร้องขอมีจำนวนเข้าใกล้ 100 ครั้งต่อวินาที จะทำให้ค่าฮิตเรโชที่วัดได้มีค่าลดลงเรื่อย ๆ รวมทั้งเวลาที่แคชใช้ในการตอบสนองกับผู้ใช้เพิ่มขึ้นอย่างรวดเร็ว แสดงให้เห็นว่าเมื่อมีการร้องขอเข้ามายังแคชในช่วงเวลาเดียวกันมากเกินกว่าที่แคชจะสามารถรองรับได้ มีผลทำให้สมรรถภาพของแคชลดลงแต่ไม่สามารถระบุได้ว่าเวลาการตอบสนองที่เพิ่มขึ้นเกิดจากการทำงานที่จุดใด

2.1.3 สควิด (Squid) [2]

สควิดเป็นซอฟต์แวร์ทางด้านอินเทอร์เน็ตพร็อกซีเซิร์ฟเวอร์ ซึ่งเลียนแบบการทำงานมาจากฮาร์เวสโปรเจ็ค (Harvest Project) พัฒนาขึ้นโดยสถาบัน NLANR (The National Laboratory for Applied Network Research) เป็นที่นิยมใช้อย่างกว้างขวาง เนื่องจากถูกออกแบบมาเพื่อใช้กับระบบยูนิกซ์ สนับสนุนการทำงานในหลาย ๆ โพรโทคอล เช่น เอชทีทีพี (HTTP Protocol), เอฟทีพี (FTP Protocol), โกเฟอร์ (Gopher Protocol) เป็นต้น และเป็นฟรีแวร์ ซึ่งเปิดโอกาสให้บุคคลอื่น ๆ เข้ามามีส่วนร่วมในการพัฒนาชุดคำสั่งโปรแกรมได้ ปัจจุบันนี้พัฒนาถึงเวอร์ชัน 2.6 นอกจากนี้แล้วสควิดยังเป็นซอร์ฟแวร์ที่ขนาดของโปรแกรมมีขนาดเล็ก ทำงานได้เร็วและมีสมรรถภาพแต่จะมีข้อเสียในแง่ของการกำหนดและติดตั้งค่าต่าง ๆ ค่อนข้างยุ่งยาก เนื่องจากมีการกำหนดค่า

ต่าง ๆ ในหลาย ๆ ส่วน ซึ่งผู้ใช้ต้องทำความเข้าใจ และกำหนดค่าให้ตรงตามสภาพทรัพยากรที่ใช้จึงจะทำงานได้สมรรถภาพสูงสุด

2.1.4 เว็บโพลีกราฟ (Web Polygraph) [11]

ในการวัดสมรรถภาพการทำงานของพร็อกซีแคชของเบคออฟจะใช้โปรแรมเว็บโพลีกราฟ ซึ่งเป็นโปรแกรมที่ใช้จำลองการทำงานการใช้เว็บระหว่างเซิร์ฟเวอร์กับไคลเอนต์ โปรแกรมส่วนที่จำลองการทำงานของเซิร์ฟเวอร์ เรียกว่า โพลีเซิร์ฟเวอร์ (Polysrv) และโปรแกรมส่วนที่จำลองการทำงานของไคลเอนต์ เรียกว่า โพลีไคลเอนต์ (Polyclt) ซึ่งสามารถกำหนดคุณลักษณะในการทำงานต่าง ๆ ที่ใกล้เคียงกับสภาพความเป็นจริงในการใช้เว็บ เนื่องจากผู้พัฒนาโปรแกรมได้อ้างอิงตามผลงานการวิจัย เช่น การวิจัยที่กล่าวถึงลักษณะการใช้เว็บ สำหรับตัวที่จะกำหนดคุณลักษณะการทำงานทั้งสองส่วนข้างต้น เรียกว่า เวิร์คโหลด (Workload) ซึ่งมีหลายตัวด้วยกัน ซึ่งบางส่วนจะพัฒนาปรับปรุงจากเวิร์คโหลดเดิมให้มีขีดความสามารถที่ดีกว่าเก่า หรือบางส่วนก็พัฒนาขึ้นมาใหม่เพื่อให้เหมาะกับสภาพแวดล้อมการใช้งานเว็บในปัจจุบันยิ่งขึ้น ซึ่งช่วยให้การวัดสมรรถภาพของพร็อกซีแคชได้ผลที่หลากหลายยิ่งขึ้น ตัวอย่างของเวิร์คโหลดสามารถดูได้ในส่วนของภาคผนวก ข

2.1.5 จีพร็อบ (Gprof) [12]

จีพร็อบเป็นคำสั่งในระบบยูนิกซ์ที่ใช้แสดงข้อมูลการเรียกใช้ฟังก์ชันและเวลาที่ใช้ในแต่ละฟังก์ชันของโปรแกรมที่เขียนด้วยภาษาซี (C) , ซีพลัสพลัส (C++) , ปาสคาล (Pascal) และภาษาฟอร์แทรน (FORTRAN) โดยจะอ่านข้อมูลจากไฟล์ชื่อว่า gmon.out ซึ่งได้จากการคอมไพล์โปรแกรมด้วยการใส่ออปชัน –pg โดยข้อมูลในการเรียกใช้ฟังก์ชันและเวลาที่ใช้ในแต่ละฟังก์ชันในการทำงานของโปรแกรมนั้น ๆ จะบันทึกไว้ในแฟ้มบันทึกเข้าออก แบ่งออกเป็นสองส่วน ได้แก่ ส่วนที่จัดเก็บรายชื่อฟังก์ชันที่ถูกเรียกใช้ในการทำงาน โดยเรียงจากรายชื่อฟังก์ชันที่ใช้เวลาในการทำงานมากไปยังรายชื่อฟังก์ชันที่มีการใช้เวลาในการทำงานน้อยเรียกว่า Flat Profile และส่วนที่จัดเก็บรายชื่อฟังก์ชันที่ถูกเรียกใช้ เรียกว่า Call Graph รูปแบบการใช้คำสั่งจีพร็อบและตัวอย่างข้อมูลที่จัดเก็บในแฟ้มบันทึกเข้าออกสามารถดูได้ในส่วนของภาคผนวก ค

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Performance Profiling Patch [7]

ทำการวิจัยเกี่ยวกับสมรรถภาพการทำงานของสควิดเพื่อศึกษาการเกิดปัญหาคอขวด (Bottleneck) ที่เกิดขึ้นในดิสก์ ซึ่งพิจารณาจากการติดต่อกันภายเครือข่าย (Network Connection) และวัดเวลาการทำงานในหกส่วน เริ่มจากการรับคำร้องขอจากผู้ใช้ (Client In) การส่งข้อมูลที่ร้องขอไปยังผู้ใช้ (Client Out) การส่งต่อคำร้องขอไปยังผู้ให้บริการ (Server Out)

ผู้ให้บริการรับคำร้องขอเข้ามา (Server In) และวัดเวลาในส่วนของการอ่าน/บันทึกไฟล์ลงดิสก์ เริ่มจากอ่านข้อมูลจากดิสก์มาไว้ที่หน่วยพักข้อมูล (Disk In) และการบันทึกข้อมูลจากหน่วยพักข้อมูลลงดิสก์ (Disk Out) โดยการทำแพทช์ (Patch) ซึ่งเป็นการเขียนโค้ดเพื่อเข้าไปวัดการทำงานในหกส่วนดังกล่าว ซึ่งในแต่ละส่วนมีการวัดเวลาการทำงานเป็นสามช่วง ประกอบด้วย

- เวลาที่เริ่มต้นของการเกิดการกระทำต่าง ๆ (The Start of The Activity) เช่น Client In วัดเวลาเริ่มต้น (Timestamp) ที่รับคำร้องขอเข้ามา
- เวลาเริ่มแรกที่เกิดความล่าช้า (The First Delay) เช่น Client In วัดเวลาที่ได้รับข้อมูลส่วนแรกจากผู้ใช้
- เวลารวมทั้งหมดที่เสร็จสิ้นการทำงานในแต่ละส่วน (Total Delay) เช่น Client In วัดเวลารวมที่ใช้เมื่อได้รับข้อมูลทั้งหมดจากผู้ใช้

พร้อมทั้งระบุว่ามอดูลที่ใช้ในส่วนของการวัดค่าต่าง ๆ อยู่ที่มอดูลใด บันทึกไว้เป็นโพรไฟล์ (Profile) [8] แสดงดังตารางที่ 2.1 ซึ่งสอดคล้องกับแนวคิดในการวิจัยในครั้งนี้ แต่งานวิจัยดังกล่าวเป็นการวัดเวลาการทำงานเฉพาะเพียงหกส่วนเท่านั้น ทำให้ไม่สามารถระบุเวลาที่แท้จริงว่าใช้ที่จุดใดบ้างในมอดูลที่เกี่ยวข้องทั้งหมด

ตารางที่ 2.1 แสดงข้อมูลที่บันทึกใน Profiling Request in Squid

Activity	Measurements	Code Location
Client In Accepting Request from a client	**Start:** The socket accepts timestamp	Icp.c: asciiHandle()
	First delay: The time it takes to receive the first data from a client	Icp.c: clientReadRequest()
	Total duration: The time it takes to receive all data from a client	Icp.c: clientReadRequest(), IcpStateFree()

2.2.2 A Performance Study of the Squid Proxy on HTTP/1.0 [9]

ทำการวิจัยเกี่ยวกับสมรรถภาพการทำงานของพร็อกซีเซิร์ฟเวอร์ โดยใช้สควิดภายใต้การทดลองที่มีเครื่องที่ใช้ ขนาดของแรม ขนาดพื้นที่ของดิสก์ ระบบปฏิบัติการที่ใช้ เวอร์ชันของสควิด และระดับของแคชแตกต่างกัน โดยวัดจาก ค่าฮิต (HIT) ค่ามิส (MISS) เวลาการตอบสนองการร้องขอ (Request Response Time) ขนาดของข้อมูลที่ร้องขอ ทราฟิก(Traffic) ของเครือข่ายที่เพิ่มขึ้น ทำให้ทราบเฉพาะสภาพการทำงานที่เหมาะสมกับสควิดเท่านั้น

2.2.3 Performance Issues of Enterprise Level Web Proxies [10]

ทำการวิจัยโดยวัดสมรรถภาพของพร็อกซีเซิร์ฟเวอร์ระหว่างเอ็ชทีทีพีดี (Httpd) ซึ่งพัฒนาขึ้นโดยสถาบัน CERN [13] กับสควิด โดยวัดจากการใช้ทรัพยากรของพร็อกซีเซิร์ฟเวอร์ คือ หน่วยประมวลผล (CPU) หน่วยความจำ (Memory) และการอ่าน/บันทึกไฟล์ลงดิสก์ ภายใต้ปริมาณการทำงาน (Workload) ระดับเดียวกันซึ่งจะมีเครื่องมือใช้วัดทุก ๆ 15 นาที เพื่อดูการใช้ทรัพยากรต่าง ๆ ของพร็อกซีแต่ละตัว งานวิจัยดังกล่าวเป็นการเปรียบเทียบการทำงานของพร็อกซีเซิร์ฟเวอร์ว่าซอฟต์แวร์ตัวใดทำงานได้ดีกว่ากันเท่านั้น ทำให้ไม่สามารถเข้าใจถึงสมรรถภาพการทำงานจริง ๆ ของพร็อกซีเซิร์ฟเวอร์ได้

2.3.4 Reducing Disk I/O of Web Proxy Server Cache [14]

ทำการวิจัยเพื่อเพิ่มสมรรถภาพการทำงานของพร็อกซีเซิร์ฟเวอร์ โดยเสนอขั้นตอนวิธี (Algorithm) ในการแคชข้อมูลของสควิด พบว่าลดการทำงานในส่วนของการอ่าน/บันทึกไฟล์ลงดิสก์ ได้ประมาณ 50 เปอร์เซ็นต์ งานวิจัยดังกล่าวเป็นการศึกษาสมรรถภาพการทำงานโดยไปเปลี่ยนแปลงขั้นตอนวิธีการในการทำงานบางส่วน ซึ่งยังไม่สามารถระบุถึงการทำงานในส่วนให้บริการการร้องขอที่เกี่ยวข้องกันทั้งหมดได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

ศึกษาและวิเคราะห์การทำงานของโปรแกรมสควิด

จากการศึกษาการทำงานของสควิด อธิบายได้ดังต่อไปนี้

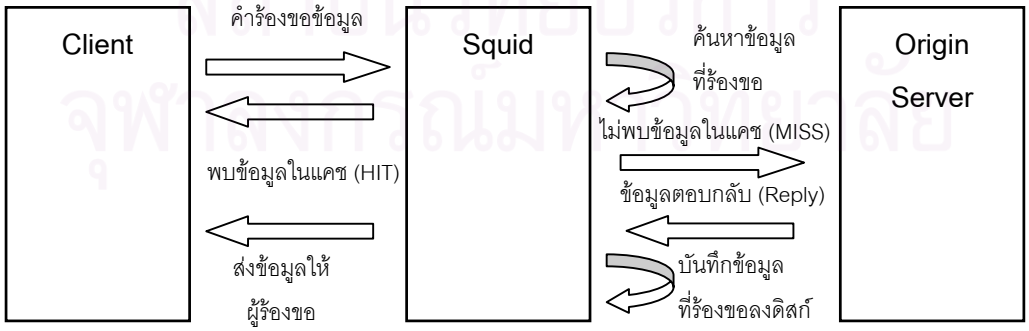
3.1 ลักษณะทั่ว ๆ ไปของโปรแกรมสควิด

สควิดเป็นโปรแกรมพร็อกซีเซิร์ฟเวอร์ที่พัฒนาขึ้นโดย Duane Wessels ซึ่งเป็นที่นิยมใช้กันอย่างแพร่หลาย เนื่องจากเป็นฟรีแวร์และโปรแกรมไฟล์มีขนาดเล็ก ทำงานในลักษณะจัดการคำร้องขอของไคลเอนต์โดยไม่มีการกระจายโพรเซส (Single Thread) และจัดการข้อมูลที่ร้องขอบ่อย ๆ ไว้ที่แรมทำให้ทำงานได้รวดเร็ว สนับสนุนการใช้งานกับโพรโทคอลที่หลากหลาย เช่น เอฟทีพี, โกเฟอร์ และเอชทีทีพี โดยทำงานภายใต้ระบบปฏิบัติการยูนิกซ์ เช่น Linux , FreeBSD SunOs/Solaris , และ AIX and OS/2 เป็นต้น

3.2 หลักการทำงานของโปรแกรมสควิด

หลักการทำงานของสควิด มีลักษณะคล้ายกับการทำงานของเว็บแคชดังที่กล่าวมาแล้วในบทที่ 2 กล่าวคือ จะทำหน้าที่เป็นเสมือนเซิร์ฟเวอร์รอรับการร้องขอข้อมูลจากไคลเอนต์และเก็บข้อมูลที่เป็นที่นิยมซึ่งไคลเอนต์ร้องขอเข้ามาบ่อย ๆ โดยไม่จำเป็นต้องไปติดต่อกับเซิร์ฟเวอร์ที่เก็บข้อมูลที่ร้องขอนั้นจริง ๆ ซึ่งจะช่วยลดเวลาการรอคอยข้อมูลของไคลเอนต์ ลดความคับคั่งของข้อมูลที่ส่งออกไปยังเครือข่าย และยังช่วยลดการทำงานของเซิร์ฟเวอร์อีกด้วย และในขณะเดียวกันก็จะทำหน้าที่เป็นไคลเอนต์เพื่อติดต่อขอข้อมูลจากเซิร์ฟเวอร์ ในกรณีที่ข้อมูลที่ไคลเอนต์ร้องขอเข้ามาไม่มีจัดเก็บไว้ในแคช หรือมีแต่ไม่สามารถอ่านข้อมูลได้ หรือตรวจสอบพบว่าข้อมูลเก่าแล้ว

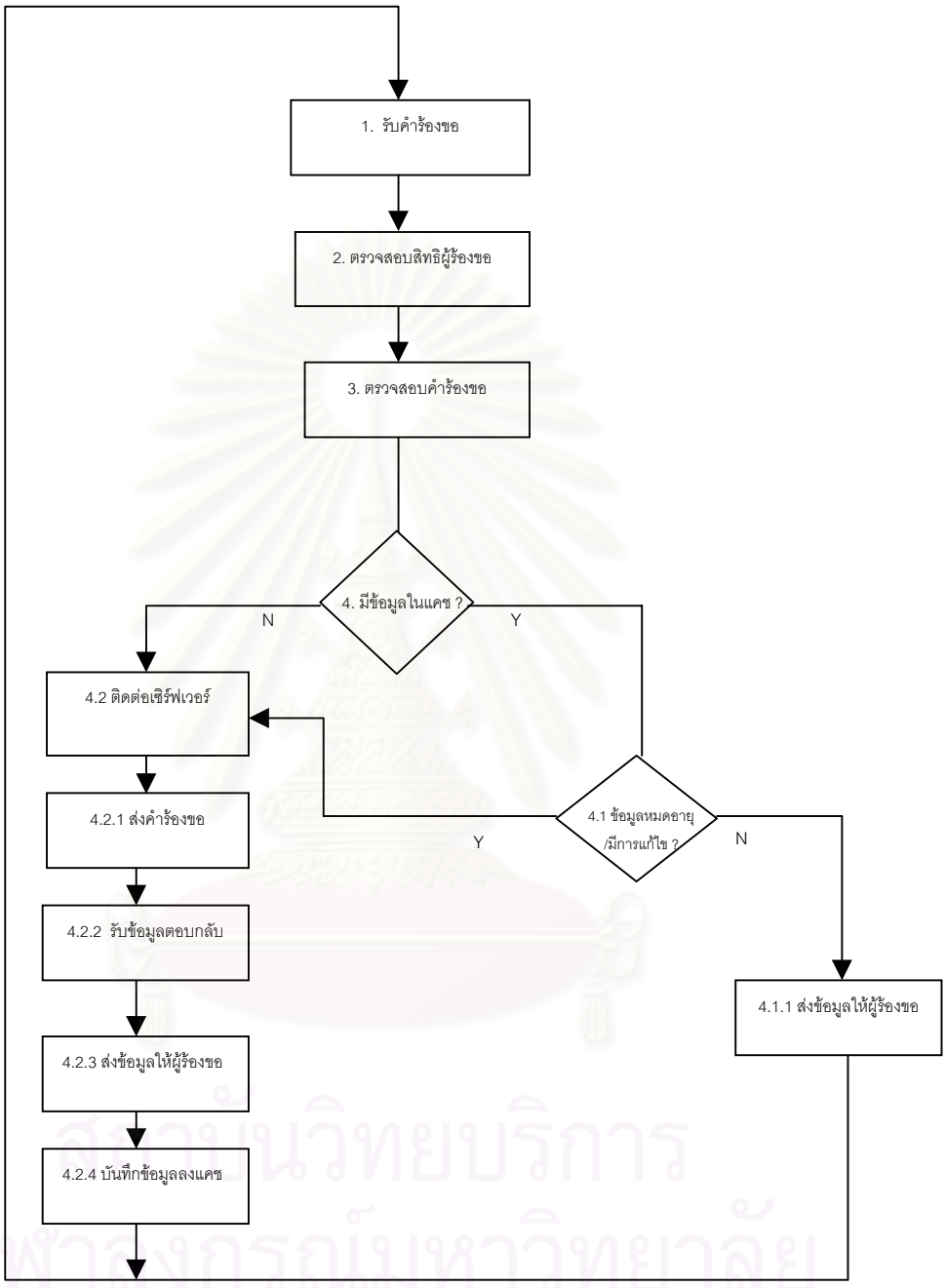
จากหลักการทำงานดังกล่าว สามารถนำเสนอได้ดังรูปภาพที่ 3.1



รูปที่ 3.1 แสดงหลักการทำงานของโปรแกรมสควิด

3.3 การทำงานในส่วนการรับคำร้องขอของโปรแกรมสควิด

การทำงานในส่วนการรับคำร้องขอของสควิด แสดงขั้นตอนได้ดังนี้



รูปที่ 3.2 แสดงขั้นตอนการทำงานของสควิดในส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จาก รูปที่ 3.2 แสดงขั้นตอนการทำงานของสควิดในส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์ อธิบายเพิ่มเติมได้ดังนี้

1. รับคำร้องขอจากไคลเอนต์
2. ตรวจสอบสิทธิของไคลเอนต์ว่าอยู่ในกลุ่มที่ได้รับอนุญาตให้สามารถเข้ามาใช้ข้อมูลในแคชได้หรือไม่
3. ตรวจสอบคำร้องขอที่ไคลเอนต์ส่งมาว่าถูกต้องตามรูปแบบที่กำหนดไว้หรือไม่
4. ค้นหาข้อมูลที่ไคลเอนต์ร้องขอมาในแคช ซึ่งแยกออกเป็นสองกรณี คือ

    4.1 กรณีมีข้อมูลในแคช สควิดจะตรวจสอบข้อมูลว่าหมดอายุหรือมีการแก้ไขหรือไม่ ซึ่งถ้าหากตรวจสอบแล้วพบว่าข้อมูลยังใหม่อยู่ และไม่มีการแก้ไข ก็จะจัดส่งข้อมูลให้ผู้ร้องขอ ตามการทำงานที่ 4.1.1 แต่ถ้าข้อมูลมีการแก้ไขก็จะทำการติดต่อเซิร์ฟเวอร์ที่จัดเก็บข้อมูลนั้น ตามการทำงานที่ 4.2

    4.2 กรณีไม่มีข้อมูลในแคช สควิดจะทำงานดังนี้

        4.2.1.1.1 ติดต่อไปยังเซิร์ฟเวอร์ที่จัดเก็บข้อมูลที่ไคลเอนต์ร้องขอเข้ามา

        4.2.1.1.2 ส่งคำร้องขอไปยังเซิร์ฟเวอร์เพื่อร้องขอข้อมูล

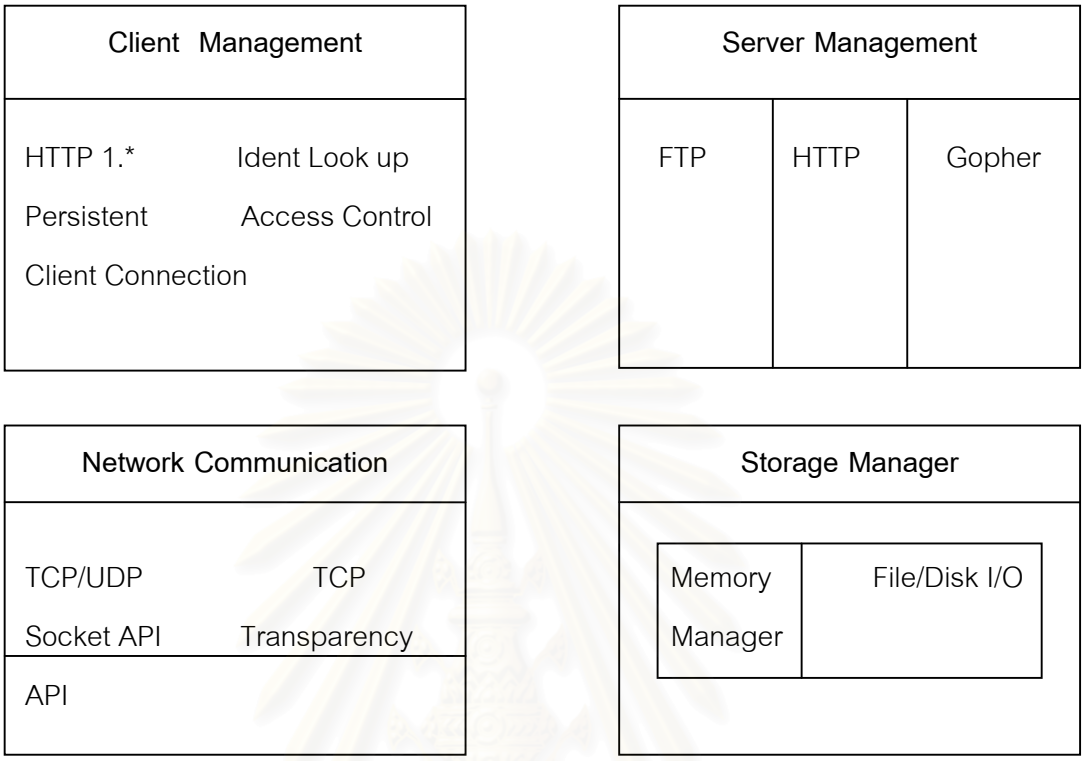
        4.2.1.1.3 รับข้อมูลตอบกลับจากเซิร์ฟเวอร์

        4.2.1.1.4 ส่งข้อมูลไปให้ไคลเอนต์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.4 องค์ประกอบหลักของโปรแกรมสควิดในส่วนการรับคำร้องขอของสควิด

จากการศึกษาการทำงานของสควิดแสดงองค์ประกอบหลักได้ดังรูปภาพที่ 3.3



รูปที่ 3.3 แสดงองค์ประกอบหลักของสควิดในส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์

เนื่องจากสควิดเป็นโปรแกรมพร็อกซีเซิร์ฟเวอร์ที่ทำหน้าที่เป็นทั้งเซิร์ฟเวอร์ที่เก็บข้อมูลไว้ให้บริการแก่ไคลเอนต์และในขณะเดียวกันก็ทำหน้าที่เป็นไคลเอนต์ในกรณีต้องติดต่อกับเซิร์ฟเวอร์ เพื่อร้องขอข้อมูลที่ไคลเอนต์ร้องขอเข้ามาแล้วไม่ได้จัดเก็บอยู่ในแคช ดังนั้นการทำงานหลัก ๆ ของสควิดจึงเกี่ยวข้องกับการติดต่อสื่อสารกันระหว่างสควิดกับไคลเอนต์และระหว่างสควิดกับเซิร์ฟเวอร์ การเก็บข้อมูล การให้บริการข้อมูลที่ไคลเอนต์ร้องขอ ด้วยเหตุนี้ ผู้วิจัยจึงได้พิจารณาองค์ประกอบหลัก ๆ ที่เกี่ยวข้องกับการให้บริการการร้องขอ แบ่งออกเป็น 4 ส่วน ดังนี้

1. **ส่วนการติดต่อไคลเอนต์** (Client Side) ส่วนนี้จะเกี่ยวข้องกับไคลเอนต์โดยตรงได้แก่ การรับคำร้องขอของไคลเอนต์ ภายใต้โพรโทคอลเอชทีทีพี โดยจะทำการตรวจสอบคำร้องขอ ตรวจสอบสิทธิของไคลเอนต์ และตรวจสอบข้อมูลที่ร้องขอว่ามีในแคชหรือไม่ มีองค์ประกอบที่เกี่ยวข้องคือ

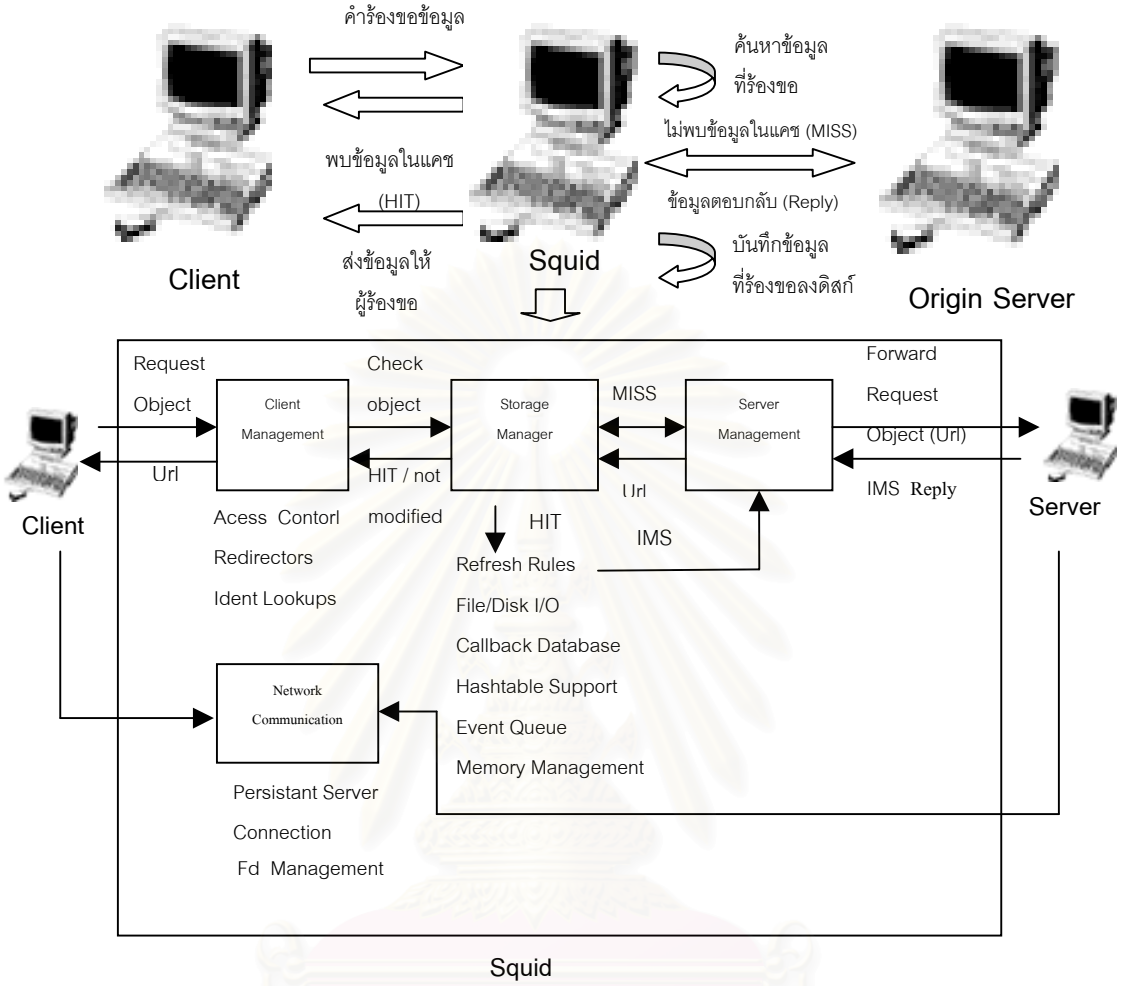
- **ส่วนการตรวจสอบสิทธิของไคลเอนต์** (Access Control) ข้อมูลที่ใช้ตรวจสอบ ได้แก่ เลขที่อยู่ของไคลเอนต์ ,โฮสต์เนมที่เก็บข้อมูลที่ไคลเอนต์ร้องขอ วิธีการร้องขอข้อมูล
- **ส่วนตรวจสอบรายชื่อไคลเอนต์** (Ident Lookups) ใช้ตรวจสอบรายชื่อของไคลเอนต์ที่ติดต่อเข้ามา เพื่อตรวจสอบสิทธิการเข้าใช้ข้อมูล
- **ส่วนการส่งคำร้องขอไปยังเซิร์ฟเวอร์โดยตรง** (Redirectors) หลังจากที่ตรวจสอบสิทธิของไคลเอนต์แล้ว อาจมีการส่งคำร้องขอไปยังเซิร์ฟเวอร์โดยตรง โดยไม่มีการเข้าไปตรวจสอบข้อมูลในแคช ซึ่งสควิดจะส่ง ยูอาร์แอล (Url) ใหม่ไปให้ไคลเอนต์
- **ส่วนการตรวจสอบความถูกต้องทันสมัยของข้อมูล** (Refresh Rules) เมื่อพบข้อมูลที่ไคลเอนต์ร้องขออยู่ในแคช และไคลเอนต์ส่งเงื่อนไขการร้องขอให้ตรวจสอบความถูกต้องทันสมัยของข้อมูล โดยสควิดจะใช้ ส่วนเฮดเดอร์ (HTTP Header) คำร้องขอ If-Modified-Since (IMS) ส่งไปตรวจสอบข้อมูลกับเซิร์ฟเวอร์ก่อน ถ้าหากข้อมูลไม่มีการเปลี่ยนแปลงแก้ไข สควิดจะทำการส่งข้อมูลที่อยู่ในแคชไปให้ไคลเอนต์

2. **ส่วนการจัดเก็บข้อมูล** (Storage Manager) ได้แก่ การจัดการพื้นที่หน่วยความจำ เพื่อรับ-ส่งข้อมูล และจัดเก็บข้อมูลลงดิสก์ การแทนที่ข้อมูลในกรณีที่ข้อมูลที่จัดเก็บนั้นเต็ม มีองค์ประกอบที่เกี่ยวข้องคือ
    - **ส่วนการจัดการหน่วยความจำ** (Memory Management) จัดการพื้นที่หน่วยความจำ เพื่อรับ-ส่งข้อมูล
    - **ส่วนการอ่านไฟล์และบันทึกไฟล์ลงดิสก์** (File/Disk I/O) ได้แก่ การอ่านไฟล์ข้อมูลจากดิสก์ การบันทึกไฟล์ข้อมูลลงดิสก์
    - **ส่วนการจัดการฟังก์ชันที่เรียกใช้หน่วยความจำ** (Callback Database) เพื่อตรวจสอบและป้องกันความผิดพลาดอันอาจเกิดขึ้นได้จากการเรียกใช้หน่วยความจำจากฟังก์ชันต่าง ๆ
    - **ส่วนการจัดเก็บข้อมูลในลักษณะการนำข้อมูลมาเข้าคีย์** (Hashtable Support) เพื่อเป็นการใช้หน่วยความจำให้เกิดประสิทธิภาพสูงสุดและมีความรวดเร็วในการค้นหาข้อมูลที่จัดเก็บ
    - **ส่วนการจัดเก็บฟังก์ชันที่จะทำงานในเวลาถัดไป** (Event Queue) ซึ่งจะจัดลำดับความสำคัญของฟังก์ชันที่จะทำงานภายใต้ระยะเวลาที่กำหนดไว้

3. **ส่วนการติดต่อเซิร์ฟเวอร์** (Server Management) ได้แก่การติดต่อกับเซิร์ฟเวอร์โดยตรง ในกรณีที่ไม่มีข้อมูลที่ไคลเอนต์ร้องขอมาเก็บไว้ในแคช หรือกรณีที่มีการตรวจสอบข้อมูลที่จัดเก็บว่ามีการเปลี่ยนแปลงแก้ไขหรือไม่ มีองค์ประกอบที่เกี่ยวข้องคือ
    - **ส่วนการส่งต่อคำร้องขอ** (Request Forwarding) ไปยังเซิร์ฟเวอร์ในกรณีที่ข้อมูลไม่มีอยู่ในแคชหรือตรวจสอบแล้วพบว่าข้อมูลนั้นหมดอายุ
4. **ส่วนการจัดการเครือข่าย** (Network Communication) ได้แก่การติดต่อกันระหว่างไคลเอนต์กับสควิด และระหว่างสควิดกับเซิร์ฟเวอร์ การเปิด-ปิดช่องทางการติดต่อ การยกเลิกการติดต่อการถ่ายโอนข้อมูล มีองค์ประกอบที่เกี่ยวข้องคือ
    - **ส่วนการจัดการซ็อกเก็ต** (File Descriptor Management) เพื่อใช้ตรวจสอบจำนวนไบต์ข้อมูลที่มีการอ่านและบันทึกในช่วงที่มีการติดต่อกันระหว่างไคลเอนต์กับสควิดหรือระหว่างสควิดกับเซิร์ฟเวอร์
    - **ส่วนการติดต่อของซ็อกเก็ต** (Persistent Server Connections) ระหว่างสควิดกับเซิร์ฟเวอร์ปลายทาง ซึ่งจะเก็บซ็อกเก็ตที่สามารถใช้ติดต่อกับเซิฟร์เวอร์ แต่จะกำหนดเวลาที่ซ็อกเก็ตนั้นว่างอยู่ (Idle) และสามารถใช้ติดต่อได้ ซึ่งถ้าหากเกินเวลาที่กำหนดแล้วซ็อกเก็ตนั้นก็จะทำการปิดการติดต่อทันที
    - **ส่วนการจัดเก็บชื่อโฮสต์เนมกับเลขที่อยู่และเลขที่อยู่กับโฮสต์เนม** (IP/FQDN Cache) เพื่อใช้ในการค้นหาชื่อโฮสต์เนมกับเลขที่อยู่ แทนการเรียกใช้โดเมนเนมเซิร์ฟเวอร์ (DNS Server) แต่ถ้าตรวจสอบแล้ว ชื่อโฮสต์หรือเลขที่อยู่ของโฮสต์ที่เก็บไว้หมดอายุ หรือตรวจสอบแล้วไม่พบชื่อโฮสต์หรือเลขที่อยู่ของโฮสต์นั้น สควิดจะติดต่อไปยังโดเมนเนมเซิร์ฟเวอร์และเก็บข้อมูลนั้นไว้ใช้ในครั้งต่อไปจนกว่าจะหมดอายุ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จากหลักการทำงานและองค์ประกอบหลักของสควิดในข้างต้นที่กล่าวมา นำเสนอในแต่ละส่วนที่เกี่ยวข้องกันได้ดังรูปที่ 3.4



รูปที่ 3.4 แสดงองค์ประกอบหลักและส่วนที่เกี่ยวข้องในการให้บริการการร้องขอข้อมูลจากไคลเอนต์ของสควิด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.5 **แฟ้มชุดคำสั่งโปรแกรมสควิดที่เกี่ยวข้องในแต่ละส่วน** พิจารณาได้ดังนี้

แฟ้มชุดคำสั่งของโปรแกรมสควิดที่เกี่ยวข้องในแต่ละส่วน พิจารณาได้ดังนี้

1. **ส่วนการติดต่อไคลเอนต์**

client.c http.cetag.c HttpStateLine.c

client_db.c HttpBody.c HttpHdrRange.c

HttpHdrCountRange.c

client_side.c HttpMsg.c HttpHeaderTool.c

HttpHeader.c HttpHdrCc.c HttpRequest.c

- ส่วนการตรวจสอบสิทธิของไคลเอนต์

  acl.c authenticate.c

- ส่วนการส่งคำร้องขอไปยังเซิร์ฟเวอร์โดยตรง

  redirect.c

- ส่วนการตรวจสอบความถูกต้องทันสมัยของข้อมูล

  refresh.c

2. **ส่วนการจัดเก็บข้อมูล**

store_client.c store.c s store_io_asyncufs.c

store_log.c store_dir.c

store_io.c store_rebuild.c

stmem.c leakfinder.c

- ส่วนการจัดการหน่วยความจำ (Memory Management)

  mem.c memBuf.c memPool.c

- ส่วนการอ่านไฟล์และบันทึกไฟล์ลงดิสก์

  filemap.c disk.c

- ส่วนการจัดการฟังก์ชันที่เรียกใช้หน่วยความจำ

  cbdata.c

- ส่วนการจัดเก็บข้อมูลในลักษณะการนำข้อมูลมาเข้าคีย์

  store_ key_md5.c

- ส่วนการจัดเก็บฟังก์ชันที่จะทำงานในเวลาถัดไป

  event.c

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3. **ส่วนการจัดการเครือข่าย**

comm.c comm_select.c

- ส่วนการจัดการซ็อกเก็ต

fd.c

4. **ส่วนการติดต่อเซิร์ฟเวอร์**

comm.c forward.c

- ส่วนการส่งต่อคำร้องขอ

fordward.c

- ส่วนการจัดเก็บชื่อโฮสต์เนมกับเลขที่อยู่และเลขที่อยู่กับโฮสต์เนม

ipcache.c fqdncache.c

3.6 ฟังก์ชันการทำงานของสควิดแต่ละส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์

จากการศึกษาการทำงานในส่วนของการให้บริการการร้องขอข้อมูลของไคลเอนต์ สรุปฟังก์ชันหลัก ๆ ที่เกี่ยวข้องในแต่ส่วนได้ดังนี้

ส่วนที่ 1 ส่วนการติดต่อไคลเอนต์

ตารางที่ 3.1 แสดงฟังก์ชันหลักในส่วนการติดต่อไคลเอนต์

ชื่อฟังก์ชัน	หน้าที่การทำงาน	ไฟล์ที่อยู่
clientHttpConnectionOpen	กำหนดช่องทางรอรับการติดต่อจากไคลเอนต์	client_side.c
httpAccept	รับการติดต่อจากไคลเอนต์และเก็บสถานะการติดต่อ	client_side.c
clientReadRequest	ตรวจสอบคำร้องขอ	client_side.c
clientAccessCheck	ตรวจสอบสิทธิการร้องขอของไคลเอนต์	
clientRedirectDone	ส่งคำร้องขอไปยังเซิร์ฟเวอร์ โดยไม่เข้าไปตรวจสอบข้อมูลในแคช	client_side.c
clientProcessRequest	ประมวลคำร้องขอเพื่อตรวจสอบว่าข้อมูลที่ร้องขอมาอยู่ในแคชหรือไม่	client_side.c
clientCacheHit	กรณีที่ข้อมูลที่ร้องขออยู่ในแคช	client_side.c
clientProcessMiss	กรณีที่ข้อมูลที่ร้องขอไม่อยู่ในแคช	client_side.c

ส่วนที่ 2 ส่วนการจัดเก็บข้อมูล

การจัดเก็บข้อมูลในสควิด

โครงสร้างข้อมูลที่สควิดใช้เพื่อการจัดเก็บข้อมูล ได้แก่ SwapDir ซึ่งจะกำหนดขนาดสูงสุดของข้อมูลที่เก็บได้ (Maximum Object Size) เก็บเส้นทาง (Path) ที่ระบุตำแหน่งของข้อมูล จัดเก็บวิธีการที่ใช้แทนที่ข้อมูลในกรณีที่แคชเต็ม และใช้โครงสร้างข้อมูล storeIoState (ดูได้จากภาคผนวก ง. เพื่อบันทึกสถานภาพของแต่ละข้อมูลแคช สควิดจะมีการจัดการข้อมูลในสองส่วน คือ ในหน่วยความจำเพื่อรับ-ส่งข้อมูลต่าง ๆ ที่ใช้ในการประมวลผลข้อมูล และจัดเก็บข้อมูลลงดิสก์เมื่อตรวจสอบแล้วข้อมูลที่ร้องขอมานั้นจัดเก็บได้ (ในกรณีเกิดการมิส) ซึ่งข้อมูลที่เก็บลงดิสก์จะเก็บเป็นสองส่วน ส่วนแรกได้แก่

- ข้อมูลส่วนต้น (Metadata) ซึ่งได้แก่ข้อมูลที่เป็นคีย์ (Cache Key) ซึ่งใช้วิธีการของ MD5 (วิธีการเข้ารหัสข้อมูลวิธีหนึ่ง) ข้อมูลบางส่วนที่มาจาก StoreEntry เช่น วันหมดอายุของข้อมูล (Expire), วันที่มีการเปลี่ยนแปลงแก้ไขข้อมูลครั้งล่าสุด (Last Modified), วันที่มีการเรียกใช้ข้อมูลครั้งล่าสุด (Last Reference), ขนาดพื้นที่ชั่วคราวที่ใช้เก็บไฟล์ (Swapfile Size), จำนวนครั้งที่มีการเรียกใช้ข้อมูลนั้น ๆ (Reference Count) เป็นต้น
- ส่วนที่เป็นข้อมูล (Object data) ซึ่งรวมทั้งเฮดเดอร์ส่วนตอบกลับ (Http Reply) ทั้งที่เป็นส่วนเฮดเดอร์และส่วนเนื้อหาด้วย

กระบวนการที่เกี่ยวข้องกับการให้บริการการร้องขอข้อมูลของไคลเอนต์

สำหรับกระบวนการที่เกี่ยวข้องในส่วนการให้บิรการการร้องขอข้อมูลของไคลเอนต์ในส่วนจัดเก็บข้อมูล คือ

- การสร้างพื้นที่ในการจัดเก็บข้อมูลเมื่อมีข้อมูลเข้ามาใหม่ (กรณีที่เกิดการมิสและข้อมูลถูกส่งมาจากเซิร์ฟเวอร์และสควิดตรวจสอบแล้วว่าข้อมูลนั้นจัดเก็บได้) หรือข้อมูลมีการเปลี่ยนแปลง (จากการตรวจสอบความถูกต้องทันสมัยของข้อมูล)
- การอ่านไฟล์ข้อมูลจากดิสก์ ในกรณีที่ข้อมูลที่ร้องขออยู่ในแคชและข้อมูลนั้นไม่มีการปรับปรุงแก้ไข
- การบันทึกไฟล์ข้อมูลลงดิสก์
- การลบไฟล์ข้อมูลออกจากดิสก์ โดยใช้วีการแทนที่ข้อมูลที่กำหนดเพื่อคัดเลือกข้อมูลที่จะลบออกจากแคชในกรณีที่ต้องการพื้นที่ในการจัดเก็บข้อมูลตัวใหม่ที่เข้ามา

สรุปฟังก์ชันหลัก ๆ ที่เกี่ยวข้องกับการให้บริการการร้องขอข้อมูลของไคลเอนต์ ในส่วนการจัดเก็บข้อมูลได้ดังนี้

ตารางที่ 3.2 แสดงฟังก์ชันหลักในส่วนการจัดเก็บข้อมูล

ชื่อฟังก์ชัน	หน้าที่การทำงาน	ไฟล์ที่อยู่
storeCreateEntry	กำหนดพื้นที่ในการจัดเก็บข้อมูลที่ร้องขอมา	store.c
storeAppend	จัดเก็บข้อมูลที่ส่งมาจากเซิร์ฟเวอร์เพิ่มเติม	store.c
storeComplete	เสร็จสิ้นการจัดเก็บข้อมูลลงแคช	store.c
storeGet	ค้นหาข้อมูลในแคชโดยใช้คีย์ ในการค้นหา	store.c
storeClientCopy	ส่งข้อมูลที่ร้องขอจากสควิดไปยังไคลเอนต์	store_client.c
storeSwapOut	บันทึกข้อมูลลงดิสก์	store_swapout.c
storeSwapOutAble	ตรวจสอบว่าสามารถที่จะบันทึกข้อมูลลงดิสก์ได้หรือไม่	store_swapout.c

ส่วนที่ 3 **ส่วนการจัดการเครือข่าย**

สรุปฟังก์ชันหลัก ๆ ที่เกี่ยวข้องกับการให้บริการการร้องขอข้อมูลของไคลเอนต์ ในส่วนการจัดการเครือข่าย ได้ดังนี้

ตารางที่ 3.3 แสดงฟังก์ชันหลักในส่วนการจัดการเครือข่าย

ชื่อฟังก์ชัน	หน้าที่การทำงาน	ไฟล์ที่อยู่
clientdbEstablished	กำหนดสถานะการติดต่อระหว่างไคลเอนต์กับสควิดเมื่อมีการรับคำร้องขอของไคลเอนต์นั้น ๆ แล้ว	client_db.c
commConnectStart	ทำการติดต่อไปยังเซิร์ฟเวอร์เพื่อส่งคำร้องขอ	comm.c
commConnectHandle	ดูแลและจัดการการติดระหว่างสควิดกับไคลเอนต์และระหว่างสควิดกับเซิร์ฟเวอร์	comm.c
comm_write	บันทึกข้อมูลการติดต่อระหว่างสควิดกับเซิร์ฟเวอร์	comm.c

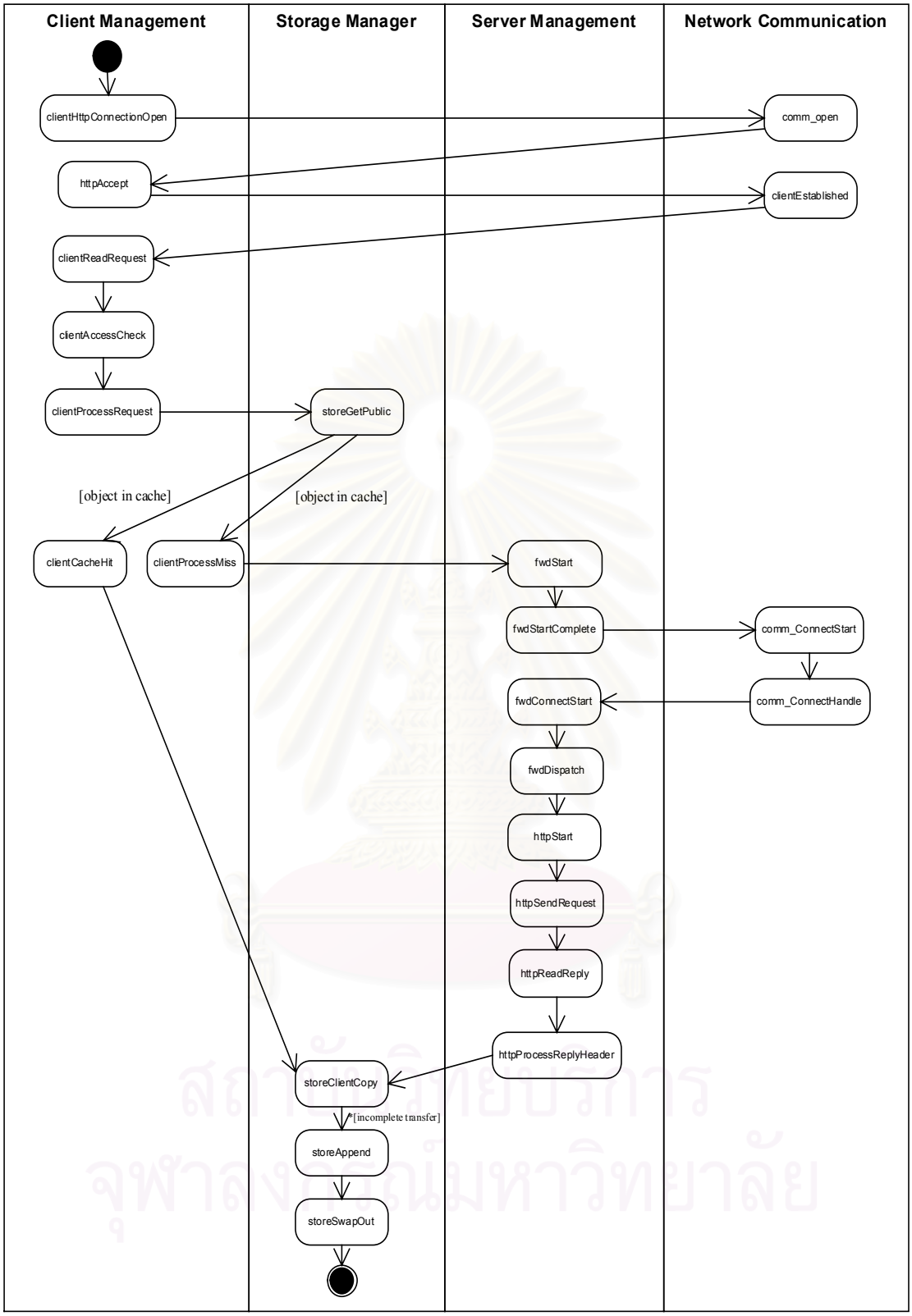
ส่วนที่ 4 ส่วนการติดต่อเซิร์ฟเวอร์

สรุปฟังก์ชันหลัก ๆ ที่เกี่ยวข้องกับการให้บริการการร้องขอข้อมูลของไคลเอนต์ ในส่วนของการติดต่อเซิร์ฟเวอร์ ได้ดังนี้

ตารางที่ 3.4 แสดงฟังก์ชันหลักในส่วนการติดต่อเซิร์ฟเวอร์

ชื่อฟังก์ชัน	หน้าที่การทำงาน	ไฟล์ที่อยู่
fwdStart	ส่งคำร้องขอไปยังเซิร์ฟเวอร์	forward.c
fwdStartComplete	ตรวจสอบการส่งคำร้องขอไปยังเซิร์ฟเวอร์	forward.c
fwdConnectStart	เริ่มทำการติดต่อไปยังเซิร์ฟเวอร์	forward.c
fwdDispatch	ส่งคำร้องขอ	forward.c
httpStart	เริ่มส่งคำร้องขอ	http.c
httpSendRequestPrefix	ส่งคำร้องขอส่วนแรกไปยังเซิร์ฟเวอร์	http.c
httpBuildRequestHeader	ควบคุมและจัดการการสร้างคำร้องขอเพื่อส่งไปยังเซิร์ฟเวอร์	http.c
httpSendComplete	ตรวจสอบความเรียบร้อยในการส่งคำร้องขอไปยังเซิร์ฟเวอร์และเตรียมการเพื่อรอรับข้อมูล	http.c
HttpReadReply	รับส่วนตอบกลับและข้อมูลจากเซิร์ฟเวอร์ปลายทางที่สควิดติดต่อไปเพื่อร้องขอข้อมูล	http.c
httpProcessReplyHeader	ประมวลผลข้อมูลส่วนตอบกลับจากเซิร์ฟเวอร์ปลายทางเพื่อตรวจสอบข้อมูลว่าสามารถจัดเก็บลงแคชได้หรือไม่	http.c

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.5 แสดงแผนภาพกิจกรรมการทำงานและฟังก์ชันที่เกี่ยวข้องในแต่ละส่วนในการให้บริการการร้องขอข้อมูลแก่ผู้ใช้ของโปรแกรมสควิด

จากรูปที่ 3.5 แสดงแผนภาพกิจกรรมการทำงานและฟังก์ชันที่เกี่ยวข้องในแต่ละส่วนในการให้บริการการร้องขอข้อมูลแก่ผู้ใช้ของโปรแกรมสควิด อธิบายกระบวนการและฟังก์ชันที่เกี่ยวข้องดังกล่าว ได้ดังนี้

ขั้นที่ 1. เป็นกระบวนการเริ่มต้นในการเปิดการติดต่อเพื่อรอรับซ็อกเก็ตที่จะติดต่อเข้ามาจากผู้ใช้ซึ่งจะต้องมีการกำหนดช่องทางไว้เพื่อรอรับการติดต่อเข้ามา ตลอดจนตรวจสอบการรับการติดต่อเข้ามาไม่ให้เกินกว่าค่าที่กำหนดรับได้

ฟังก์ชันหลักที่เกี่ยวข้องได้แก่

- clientHttpConnectionOpen()
- comm_open()

ขั้นที่ 2. เป็นกระบวนการรับคำร้องขอจากผู้ใช้ โดยจะมีการกำหนดค่าซ็อกเก็ตตัวใหม่ให้กับผู้ใช้ที่ส่งคำร้องขอเข้ามา และมีการกำหนดสถานะการติดต่อระหว่างกันเกิดขึ้น เพื่อให้ทราบว่าขณะนี้ได้มีการรับการติดต่อจากผู้ใช้คนนั้น ๆ อยู่

ฟังก์ชันหลักที่เกี่ยวข้องได้แก่

- httpAccept()
- clientEstablished()

ขั้นที่ 3. เป็นกระบวนการในการตรวจสอบคำร้องขอที่ผู้ใช้ส่งเข้ามาว่าถูกต้องหรือไม่ โดยจะมีการตรวจสอบในแต่ละส่วนของเฮดเดอร์ที่ส่งมาว่าแต่ละองค์ประกอบครบถ้วนถูกต้อง ซึ่งประกอบไปด้วย วิธีการร้องขอ โพรโทคอลที่ใช้ติดต่อเข้ามา ยูอาร์แอลหรือแหล่งที่อยู่ข้อมูลที่ร้องขอ นอกจากนี้ยังมีการตรวจสอบสิทธิของผู้ที่ส่งคำร้องขอเข้ามาด้วย

ฟังก์ชันหลักที่เกี่ยวข้องได้แก่

- clientReadRequest()

  -parseHttpRequest()

  -urlParse()

  -httpRequestParseHeader()

  -httpHeaderParse()

  -urlCheckRequest()

  -clientCheckContentLength()

  -clientAccessCheck()

  -clientAccessCheckDone()

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นที่ 4. เป็นกระบวนการตรวจสอบข้อมูลที่ผู้ใช้ร้องขอมาว่ามีอยู่ในแคชหรือไม่ซึ่งถ้าหากมีในแคชก็จะส่งค่าฮิต กลับไป แต่ถ้าหากไม่พบในแคชจะส่งค่ามิส กลับไป

ฟังก์ชันหลักที่เกี่ยวข้อง ได้แก่

- clientProcessRequest ()

  -storeGetPublic()

ขั้นที่ 4.1 กรณีที่พบมูลในแคช ซึ่งก็จะมีกระบวนตรวจสอบข้อมูลที่จัดเก็บนั้นเก่าหรือไม่ถ้ายังไม่เก่า ก็จะต้องตรวจสอบอีกว่ามีการเปลี่ยนแปลงแก้ไขข้อมูลหรือไม่ โดยการเช็คจากค่า IMS (If-Modified-Since)

ฟังก์ชันหลักที่เกี่ยวข้อง ได้แก่

- clientCacheHit()

  -refreshCheckHttp()

  -clientProcessExpired()

  -modifiedSince()

ขั้นที่ 4.2 กรณีที่ไม่พบข้อมูลที่ร้องขอในแคช จะมีการตรวจสอบว่าข้อมูลที่ร้องขอสามารถติดต่อขอได้แหล่งใดอีกบ้างนอกเหนือจากเซิร์ฟเวอร์ที่เก็บข้อมูลนั้นจริง ๆ (ในกรณีที่มีการติดตั้งแคชแบบหลายระดับ ไม่ได้มีแคชเพียงตัวเดียว) มีกระบวนการจัดเตรียมพื้นที่เพื่อใช้ในการจัดเก็บข้อมูล

ฟังก์ชันหลักที่เกี่ยวข้อง ได้แก่

- clientProcessMiss()

  -clientCreateStoreEntry()

  -storeCreateEntry()

ขั้นที่ 5. เป็นกระบวนการส่งคำร้องขอต่อไปยังเซิร์ฟเวอร์ที่มีข้อมูลที่ผู้ใช้ร้องขอเข้ามา เมื่อพิจารณาได้แล้วว่าจะส่งคำร้องขอไปยังเซิร์ฟเวอร์ตัวใด ซึ่งจะมีกระบวนการในการควบคุมดูแลการติดต่อตลอดจนดูแลการติดต่อนั้นให้เป็นไปอย่างสมบูรณ์ กล่าวคือสามารถติดต่อได้และได้รับข้อมูลที่ร้องขอไปตอบกลับมาโดยเรียบร้อย

ฟังก์ชันหลักที่เกี่ยวข้องได้แก่

- fwdStart()
- fwdStartComplete()

ขั้นที่ 6. เป็นกระบวนการเริ่มทำการติดต่อไปยังเซิร์ฟเวอร์ที่เก็บข้อมูลที่ผู้ใช้ร้องขอเข้ามา ซึ่งจะมีกระบวนการในการติดต่อตลอดจนดูแลการติดต่อนั้นให้เป็นไปได้ด้วยดีในตลอดระยะเวลาการติดต่อนั้น ๆ

ฟังก์ชันหลักที่เกี่ยวข้องได้แก่

- fwdConnectStart()

  -commConnectStart()

  -commConnectHandle() -

ขั้นที่ 7. เป็นกระบวนการที่เกี่ยวข้องกับการส่งคำร้องขอไปยังเซิร์ฟเวอร์ เมื่อทำการติดต่อกับเซิร์ฟเวอร์ที่เก็บข้อมูลที่ผู้ใช้ร้องขอได้แล้ว จะทำการส่งคำร้องขอไปยังเซิร์ฟเวอร์นั้น ๆ และคอยตรวจสอบว่าได้ส่งคำร้องขอไปเรียบร้อยแล้วหรือไม่

ฟังก์ชันหลักที่เกี่ยวข้อง ได้แก่

- fwdDispatch()

  -httpStart()

  -httpSendRequest()

  -httpSendRequestPrefix()

  -httpBuildRequestHandler()

  -comm_Write()

  -commHandleWrite()

  -commWriteStateCallBackAndFree()

  -httpSendComplete()

ขั้นที่ 8. เป็นกระบวนการอ่านข้อมูลตอบกลับจากเซิร์ฟเวอร์ที่สควิดส่งคำร้องขอข้อมูลไปยังเซิร์ฟเวอร์นั้น ๆ ซึ่งเมื่อส่งคำร้องขอไปเรียบร้อยแล้วก็จะมีการรับข้อมูลที่ตอบกลับมาจากเซิร์ฟเวอร์ ซึ่งก็จะมีกระบวนการในการตรวจสอบข้อมูลที่ตอบกลับมาทั้งในส่วนที่เป็นเฮดเดอร์และส่วนที่เป็นข้อมูล ตรวจสอบว่าข้อมูลที่รับมานั้นจะสามารถจัดเก็บลงแคชได้หรือไม่ มีกระบวนการตรวจสอบการรับข้อมูลที่เซิร์ฟเวอร์จัดส่งมาจนกว่าที่ได้รับข้อมูลจนเสร็จสมบูรณ์

ฟังก์ชันหลักที่เกี่ยวข้อง ได้แก่

- httpReadReply()

  -httpProcessReplyHeader()

  -httpReplyParse()

  -httpCacheableReply()

  -storeAppend()

-storeSwapOut() → ขั้นที่ 10

-httpConnTransferDone()

ขั้นที่ 9. เป็นกระบวนการในการส่งข้อมูลที่ร้องขอไปยังผู้ใช้ เมื่อตรวจสอบข้อมูลเรียบร้อยแล้ว ก็จะทำการจัดส่งข้อมูลนั้น ๆ ไปให้แก่ผู้ที่ร้องขอเข้ามา จนกระทั่งผู้ร้องขอได้รับข้อมูลที่ร้องขอนั้นเสร็จสมบูรณ์

ฟังก์ชันหลักที่เกี่ยวข้อง ได้แก่

- storeClientCopy

  -storeClientCopy2()

  -storeClientCopy3()

  -storeClientCallBack()

  -clientSendMoreData()

  -clientWriteComplete()

ขั้นที่ 10. เป็นกระบวนการบันทึกข้อมูลที่ร้องขอจากเซิร์ฟเวอร์ลงดิสก์ เมื่อส่งข้อมูลไปให้แก่ผู้ใช้ที่ร้องขอข้อมูลนั้นเรียบร้อยแล้ว ก็จะทำการจัดเก็บข้อมูลนั้นลงดิสก์ ซึ่งก็จะมีกระบวนการคอยตรวจสอบว่าสามารถจัดเก็บได้หรือไม่

ฟังก์ชันหลักที่เกี่ยวข้อง ได้แก่

- storeSwapOut()

  -storeWrite()

  -storeSwapOutAble()

  -storeSwapOutObjectBytesOnDisk()

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การทดลองการทำงานของโปรแกรมสควิด

4.1 แนวความคิดในการวัดประสิทธิภาพ

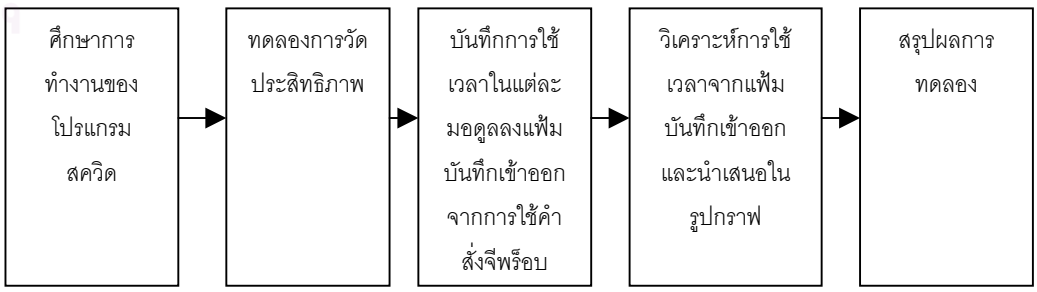
เนื่องจากการวัดประสิทธิภาพของสควิดในงานวิจัยนี้ ใช้วัดจากเวลาที่สควิดใช้ในแต่ละมอดูล โดยมอดูลที่วัดจะเกี่ยวข้องกับงานในส่วนการให้บริการการร้องขอข้อมูลแก่ไคลเอนต์ กล่าวโดยสรุปคือ ดูเวลาที่ใช้ทั้งหมด ในแต่ละมอดูลที่ทำงานตั้งแต่รับคำร้องขอเข้ามาจนกระทั่งส่งข้อมูลกลับไปให้ยังไคลเอนต์

4.1.1 แนวทางการศึกษา

จากแนวความคิดดังกล่าว จึงเข้าสู่กระบวนการศึกษาดังนี้

1. ศึกษาการทำงานของโปรแกรมสควิด โดยศึกษาองค์ประกอบต่าง ๆ ที่เกี่ยวข้องกับการทำงานโดยภาพรวมเพื่อศึกษาขั้นตอนในการรับคำร้องขอข้อมูลจากไคลเอนต์และส่งข้อมูลกลับไปยังไคลเอนต์
2. ศึกษาเส้นทางการทำงานของโปรแกรมในส่วนการให้บริการการร้องขอข้อมูลของสควิดโดยใช้คำสั่งจีพร็อบช่วยในการแสดงเส้นทางการเรียกใช้มอดูลของโปรแกรมสควิด
3. สรุปมอดูลหลัก ๆ ที่เกี่ยวข้องกับการทำงานในส่วนการให้บริการการร้องขอข้อมูลของไคลเอนต์
4. วัดประสิทธิภาพการทำงานโดยการจำลองการทำงานจากการทดสอบประสิทธิภาพพร็อกซีของเบคออฟ ซึ่งก็จะได้เวลาที่ใช้ในแต่ละมอดูลโดยใช้คำสั่งจีพร็อบในการจัดเก็บข้อมูลลงแฟ้มบันทึกเข้าออก
5. นำเวลาที่ได้จากแฟ้มบันทึกเข้าออกมาวิเคราะห์นำเสนอในรูปกราฟ
6. สรุปผลการทดลองที่ได้

4.1.2 สรุปขั้นตอนหลัก ๆ ในการทำวิจัย



รูปที่ 4.1 แสดงขั้นตอนหลักในการทำวิจัย

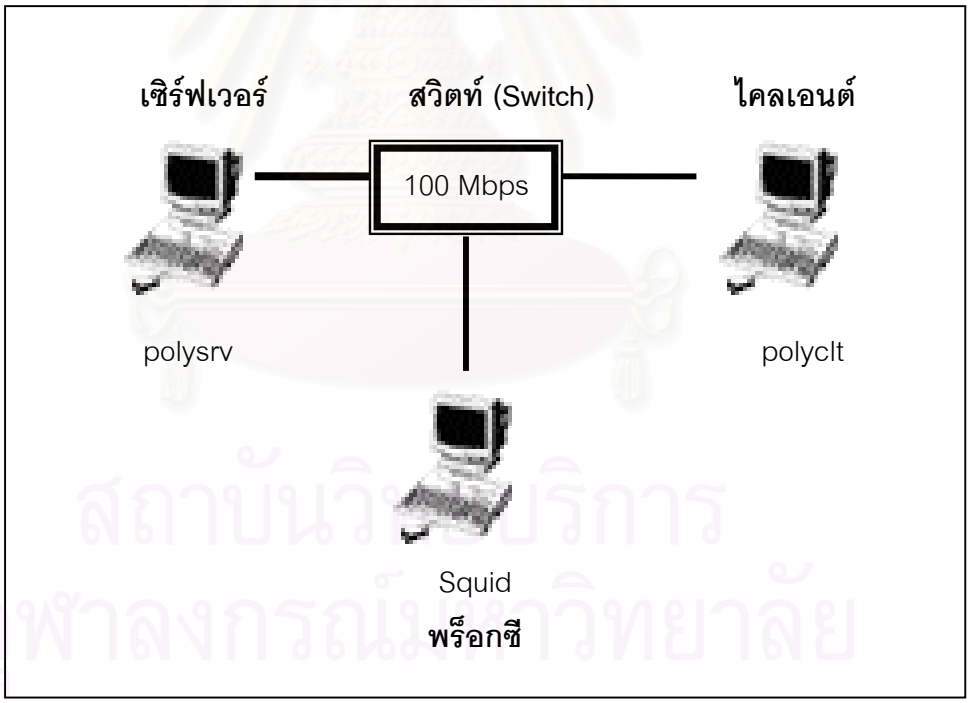
4.2 แนวทางการทดลอง

สำหรับการทดลองการทำงานของสควิด สิ่งที่ผู้วิจัยพิจารณา เพื่อใช้เป็นแนวทางในการทดลอง ดังนี้

4.2.1 รูปแบบการทดลอง โดยเลียนแบบการทดลองจากเบคออฟ

รูปแบบการทดลองวัดประสิทธิภาพโปรแกรมสควิดในครั้งนี้ผู้วิจัยได้เลียนแบบการวัดประสิทธิภาพของเบคออฟ ซึ่งประกอบด้วย

- เครื่องคอมพิวเตอร์สำหรับใช้รันโปรแกรมเว็บโพลีกราฟส่วนที่เป็นผู้ให้บริการ (Polysrv) จำนวน 1 เครื่อง
- เครื่องคอมพิวเตอร์สำหรับรันโปรแกรมสควิด ซึ่งเป็นพร็อกซีเซิร์ฟเวอร์ จำนวน 1 เครื่อง
- เครื่องคอมพิวเตอร์สำหรับรันโปรแกรมเว็บโพลีกราฟส่วนของผู้ใช้ที่ร้องขอข้อมูล (Polyclt) จำนวน 1 เครื่อง



รูปที่ 4.2 แสดงรูปแบบการทดลองวัดสมรรถภาพของโปรแกรมสควิด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.2 เครื่องมืออุปกรณ์ที่ใช้ในการทดลอง

ตารางที่ 4.1 แสดงรายการอุปกรณ์ที่ใช้ในการทดลอง

	เครื่องคอมพิวเตอร์	ระบบปฏิบัติการ	ซอฟแวร์
พร็อกซีเซิร์ฟเวอร์	Pentium III Ram 256 MB HD 20 GB	Red Hat Linux version 7.1	squid 2.4 stable3
ไคลเอนต์	Pentium III Ram 128 MB HD 7 GB	Freebsd 4.3 TMF	Web polygraph 2.7.6 stable
เซิร์ฟเวอร์	Pentium III Ram 64 MB HD 4 GB	Freebsd 4.3 TMF	Web polygraph 2.7.6 stable

4.2.3 การกำหนดค่าต่าง ๆ ในการทดลอง

- ในการทดลองวัดสมรรถภาพพร็อกซีของเบคออฟในส่วนของโปรแกรมสควิด มีการกำหนดค่า ในไฟล์ squid .config ซึ่งเปลี่ยนแปลงจากค่าปกติที่ถูกกำหนดไว้แล้ว ดังนี้

  -max_disk_open_fd 200

  -memory_pools off

  -half_closed_clients_off

  -cache_mem 128 MB

- โปรแกรมเว็บโพลีกราฟ

  -ใช้ตัวกำหนดลักษณะการทำงานทั้งในส่วนที่เป็นไคลเอนต์และเซิร์ฟเวอร์หรือเรียกว่าเวิร์คโหลดที่เบคออฟใช้ เป็นตัวจำลองอัตราการร้องขอในการทดลอง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.4 ข้อมูลที่ใช้ในการทดลอง

การทดลองการทำงานของสควิดจะเลียนแบบการทดลองจากเบคออฟ โดยมีรูปแบบการทดลองดังกล่าวข้างต้น และกำหนดค่าต่าง ๆ ในการทดลองดังนี้

1. จำนวนการร้องขอต่อหน่วยเวลาที่ใช้ในการทดลองใช้อัตราการร้องขอที่ใช้ในการทดลองของเบคออฟ โดยเริ่มที่ 10, 20, 30, 40, 50, 60, 70, 80, 90 และ 100 คำร้องขอต่อวินาที โดยมีการจำลองไอพีแอดเดรสของเครื่องทั้งส่วนที่เป็นเซิร์ฟเวอร์และไคลเอนต์จากเครื่องมือที่มีอยู่ในโปรแกรมเว็บโพลีกราฟ เรียกว่า aka tool
2. เวลาที่ใช้ในการทดลองต่อจำนวนการร้องขอชุดหนึ่ง ๆ 2.30 ชั่วโมง เนื่องจากเวลาที่ใช้กำหนดในการทดลองที่แนะนำไว้ของเบคออฟอย่างน้อยไม่ควรต่ำกว่า 150 นาทีขึ้นไปเพื่อให้สามารถเก็บสถิติการทำงานในเฟสต่าง ๆ ของแคชได้
3. ข้อมูลที่ใช้กำหนดลักษณะการทำงานทั้งในส่วนที่เป็นไคลเอนต์และส่วนที่เป็นเซิร์ฟเวอร์ ได้แก่ตัวเวิร์คโหลดที่เรียกว่า polymix-2 เนื่องจากมีการพัฒนาเพิ่มเติมมาจากการทดลองของเบคออฟในครั้งแรก โดยมีการเพิ่มเติมในส่วนของอายุของข้อมูลที่จัดเก็บ ทำให้สามารถตรวจสอบความถูกต้องตรงกันของข้อมูลได้ เหมาะกับการนำมาทดสอบประสิทธิภาพการทำงานของพร็อกซียิ่งขึ้น การทดลองจะมีทั้งหมด 9 เฟส ได้แก่

   3.1 เฟสเริ่มต้น (Warm Pharse)

   3.2 เฟสเพิ่มการทำงานของพร็อกซีในช่วงที่หนึ่ง (Inc1 Pharse)

   3.3 เฟสเพิ่มการทำงานสูงสุดของพร็อกซีในช่วงที่หนึ่ง (Top1 Pharse)

   3.4 เฟสลดการทำงานของพร็อกซีในช่วงที่หนึ่ง (Dec1 Pharse)

   3.5 เฟสช่วงว่างจากการทำงานของพร็อกซี (Idle Pharse)

   3.6 เฟสเพิ่มการทำงานของพร็อกซีในช่วงที่สอง (Inc2 Pharse)

   3.7 เฟสเพิ่มการทำงานสูงสุดของพร็อกซีในช่วงที่สอง (Top2 Pharse)

   3.8 เฟสลดการทำงานของพร็อกซีในช่วงที่สอง (Dec2 Pharse)

   3.9 เฟสการเตรียมตัวเพื่อหยุดการทำงาน (Cool Pharse)

   ในส่วนของรายละเอียดในแต่ละเฟส สามารถดูได้จากภาคผนวก ข

4.2.5 การเก็บผลการทดลอง

ผลการทดลองที่ได้จัดเก็บเป็นแฟ้มบันทึกเข้าออกโดยใช้คำสั่งจีพร็อบ ซึ่งแฟ้มบันทึกเข้าออกที่จัดเก็บประกอบด้วยสองส่วน ได้แก่

1. Flat Profile

แสดงรายชื่อฟังก์ชันที่เรียกใช้โดยเรียงตามลำดับเวลาที่ใช้ในแต่ละฟังก์ชันจากมากไปน้อย สำหรับการทำงานของโปรแกรมสควิดในครั้งหนึ่ง ๆ จนกระทั่งจบการทำงาน เวลาที่ใช้แสดงในหน่วย ไมโครวินาที ตัวอย่าง Flat Profile สามารถดูได้จากภาคผนวก ค

2. Call Graph

แสดงเวลาที่ใช้ในแต่ละฟังก์รวมถึงเวลาที่ใช้ในฟังก์ชันย่อย ๆ ที่ฟังก์ชันนั้น ๆ เรียกใช้สำหรับการทำงานครั้งหนึ่ง ๆ ของโปรแกรมสควิดจนกระทั่งจบโปรแกรม

Call Graph แบ่งออกเป็น 3 ส่วน ได้แก่

1. Primary Line แสดงถึงฟังก์ชันนั้น ๆ ไปเรียกใช้ฟังก์ชันย่อย ๆ ใดบ้าง สำหรับการทำงาน
   ครั้งหนึ่ง ๆ ของโปรแกรมสควิดจนกระทั่งจบโปรแกรม
2. Line for a function's Callers แสดงถึงฟังก์ชันนั้น ๆ ถูกฟังก์ชันใดเรียกใช้สำหรับการทำงานครั้งหนึ่ง ๆ ของโปรแกรมสควิดจนกระทั่งจบโปรแกรม
3. Line for a function's Subroutines แสดงถึงฟังก์ชันย่อย ๆ ที่ถูกฟังก์ชันนั้น ๆ เรียกใช้สำหรับการทำงานครั้งหนึ่ง ๆ ของโปรแกรมสควิดจนกระทั่งจบโปรแกรม ตัวอย่าง Call Graph ในส่วนต่าง ๆ สามารถดูได้จากภาคผนวก ค.

4.2.4 วิธีการคำนวณเวลาที่ใช้ในแต่ละส่วน

ในการวัดประสิทธิภาพการทำงานของสควิดในส่วนการให้บริการการร้องขอข้อมูลของผู้ใช้ พิจารณาเป็นสี่ส่วน ได้แก่ ส่วนไคลเอนต์, ส่วนการจัดเก็บข้อมูล, ส่วนเซิร์ฟเวอร์ และส่วนการจัดการเครือข่าย โดยการคำนวณเวลาที่ใช้ในแต่ละส่วนของสควิดจะพิจารณาจากไฟล์บันทึกเข้าออก จากสองส่วนด้วยกัน คือ

- ไฟล์บันทึกเข้าออกจากจีพร็อบ ซึ่งจัดเก็บรายชื่อฟังก์ชันที่ถูกเรียกใช้และเวลาที่ใช้ในแต่ละฟังก์ชัน
- ไฟล์บันทึกเข้าออกจากสควิด ซึ่งจะได้เวลาที่ใช้ไปทั้งหมดทั้งในกรณีที่เกิดการมิสและฮิต

การคำนวณเวลาที่ใช้ในแต่ละส่วน

1. คำนวณหาเวลาเฉลี่ยต่อหนึ่งคำร้องขอของแต่ละฟังก์ชันในแต่ละส่วน (จากไฟล์บันทึกเข้าออกของจีพร็อบ) โดยนำเวลาที่ใช้ไปทั้งหมดของการทดลองครั้งนั้น ๆ หารด้วยจำนวนคำร้องขอ (ได้จากแฟ้มบันทึกเข้าออกของสควิด) ทั้งหมดที่รับได้

   การหาค่าเฉลี่ยของฟังก์ชันในแต่ละส่วน

   - ส่วนไคลเอนต์หารด้วยจำนวนคำร้องขอทั้งหมดที่รับมา
   - ส่วนการจัดเก็บข้อมูลหารด้วยจำนวนคำร้องขอทั้งหมดที่รับมา
   - ส่วนเซิร์ฟเวอร์หารด้วยจำนวนคำร้องขอทั้งหมดที่รับมากรณีมิส
   - ส่วนการจัดการเครือข่ายหารด้วยจำนวนคำร้องขอทั้งหมดที่รับมา
2. คำนวณหาเวลารวมที่ใช้ในแต่ละส่วน โดยนำเวลาที่ใช้ของแต่ละฟังก์ชันโดยเฉลี่ยต่อหนึ่งคำร้องขอในแต่ละส่วน (เวลาในข้อ 1) มารวมกัน
3. คำนวณหาเวลาทั้งหมดที่ใช้เฉลี่ยต่อหนึ่งคำร้องขอ โดยนำเวลาในแต่ละส่วน (เวลาที่ได้ในข้อ 2) มารวมกัน
4. คำนวณเวลาที่ใช้ในแต่ละฟังก์ชันของแต่ละส่วนคิดเป็นเปอร์เซ็นต์ โดยคำนวณจาก

$$\%\ \text{ของเวลาที่ใช้ในแต่ละฟังก์ชัน} = \frac{\text{เวลาที่ได้ในข้อหนึ่ง}}{\text{เวลาที่ได้ในข้อสอง}} \times 100$$

5. คำนวณเวลาที่ใช้ในแต่ละส่วนคิดเป็นเปอร์เซ็นต์ โดยคำนวณจาก

$$\%\ \text{ของเวลาที่ใช้ในแต่ละส่วน} = \frac{\text{เวลาที่ได้ในข้อสอง}}{\text{เวลาที่ได้ในข้อสาม}} \times 100$$

4.2.5 การวิเคราะห์ผลที่ได้และรูปแบบการนำเสนอ

เพื่อให้ง่ายต่อการศึกษาผลการทดลองการวัดประสิทธิภาพของโปรแกรมสควิด จึงจะนำเสนอข้อมูลที่วิเคราะห์แล้วในรูปแบบกราฟ ทั้งที่เป็นกราฟในลักษณะรูปแท่ง รูปเส้น รูปวงกลม ทั้งนี้ขึ้นอยู่กับความเหมาะสมและง่ายต่อการทำความเข้าใจ

บทที่ 5

ผลการทดลองและการวิเคราะห์ผลการทดลอง

จากการทดลองการวัดสมรรถภาพของสควิดในรูปแบบดังที่กล่าวมาแล้วในบทที่ 4 ซึ่งจัดเก็บไว้ในแฟ้มบันทึกข้อมูลเข้าออกจากการใช้คำสั่งจีพร็อบ จะทำการพิจารณาผลการทดลองในส่วนต่าง ๆ ออกเป็นสองกรณีคือ การทดลองกรณีผ่านเครือข่าย ซึ่งเป็นลักษณะการทดลองที่เลียนแบบการทดลองของเบคออฟ และการทดลองในกรณีไม่ผ่านเครือข่ายโดยใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงานส่วนที่เป็นไคลเอนต์, เซิร์ฟเวอร์และสควิด ในระดับอัตราการร้องขอที่ 10 คำร้องขอต่อวินาที จากการทดลองในทั้งสองกรณี จะพิจารณาเวลาที่ได้จากแฟ้มบันทึกเข้าออกของจีพร็อบซึ่งในส่วนนี้จะได้เวลาที่ใช้ในแต่ละฟังก์ชัน มีหน่วยเวลาเป็นไมโครวินาที โดยจะทำการพิจารณาเสนอผลในภาพรวมก่อน จากนั้นจะทำการพิจารณาผลการทดลองการใช้เวลาของฟังก์ชันในส่วนที่มีผลต่อการทำงานของโปรแกรมสควิด เพื่อดูการใช้เวลาของฟังก์ชันการทำงานของส่วนนั้น ๆ ซึ่งจะพิจารณาผลในกรณีผ่านเครือข่ายเป็นหลักและพิจารณาในกรณีไม่ผ่านเครือข่ายเพื่อเป็นการเปรียบเทียบ เสนอผลการทดลองได้ดังต่อไปนี้

5.1 ผลการทดลองและวิเคราะห์การใช้เวลาในภาพรวมกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

ผลการทดลองที่ได้ในแต่ละส่วนเมื่อพิจารณาโดยภาพรวมในกรณีนี้แสดงได้ดังตารางที่ 5.1 และ ตารางที่ 5.2

ตารางที่ 5.1 แสดงเวลา (ไมโครวินาที) รวมในแต่ละส่วนโดยพิจารณาจากไฟล์บันทึกเข้าออกของจีพร็อบโดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

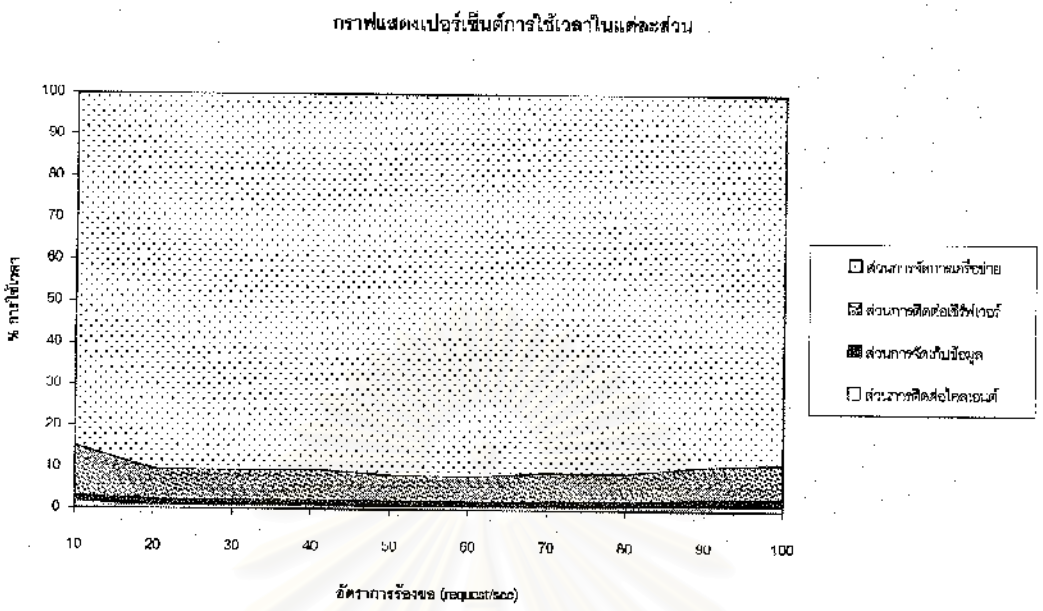
อัตราการร้องขอ / **ส่วนการทำงาน**	10	20	30	40	50	60	70	80	90	100
ส่วนการติดต่อไคลเอนต์	81.15	62.75	84.75	87.45	92.98	88.41	88.27	88.45	90.33	89.25
ส่วนการจัดเก็บข้อมูล	61.07	92.32	104.06	110.41	108.99	105.06	111.77	104.60	108.58	115.50
ส่วนการติดต่อเซิร์ฟเวอร์	554.07	490.03	622.48	615.12	614.12	576.70	574.63	547.11	560.67	571.76
ส่วนการจัดการเครือข่าย	3899.52	6082.51	7764.74	7555.04	8955.58	8671.78	7772.38	7519.95	6306.08	6071.83

ตารางที่ 5.2 แสดงเปอร์เซ็นต์การใช้เวลาของแต่ละส่วนที่สควิดใช้ในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / **ส่วนการทำงาน**	10	20	30	40	50	60	70	80	90	100
ส่วนการติดต่อไคลเอนต์	1.77	0.93	0.99	1.05	0.95	0.94	1.03	1.07	1.28	1.30
ส่วนการจัดเก็บข้อมูล	1.33	1.37	1.21	1.32	1.12	1.11	1.31	1.27	1.54	1.69
ส่วนการติดต่อเซิร์ฟเวอร์	12.06	7.28	7.26	7.35	6.28	6.11	6.72	6.62	7.94	8.35
ส่วนการจัดการเครือข่าย	84.85	90.41	90.54	90.28	91.65	91.84	90.94	91.04	89.25	88.66

พิจารณาผลของข้อมูลจากตารางที่ 5.1 และ ตารางที่ 5.2 นำเสนอเป็นกราฟเพื่อแสดงภาพโดยรวมการใช้เวลาของแต่ละส่วนในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของสควิดแสดงได้ดังกราฟรูปที่ 5.1 และเพื่อความชัดเจนในการพิจารณายิ่งขึ้นได้นำเสนอกราฟ โดยพิจารณาเปรียบเทียบข้อมูลการใช้เวลาและเปอร์เซ็นต์การใช้เวลาในแต่ละส่วนโดยภาพรวมในระดับอัตราการร้องขอที่น้อยที่สุดและระดับอัตราการร้องขอที่มากที่สุด ซึ่งในที่นี้ได้แก่ ระดับอัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ100 คำร้องขอต่อวินาที แสดงได้ดังกราฟรูปที่ 5.2 และกราฟรูปที่ 5.3

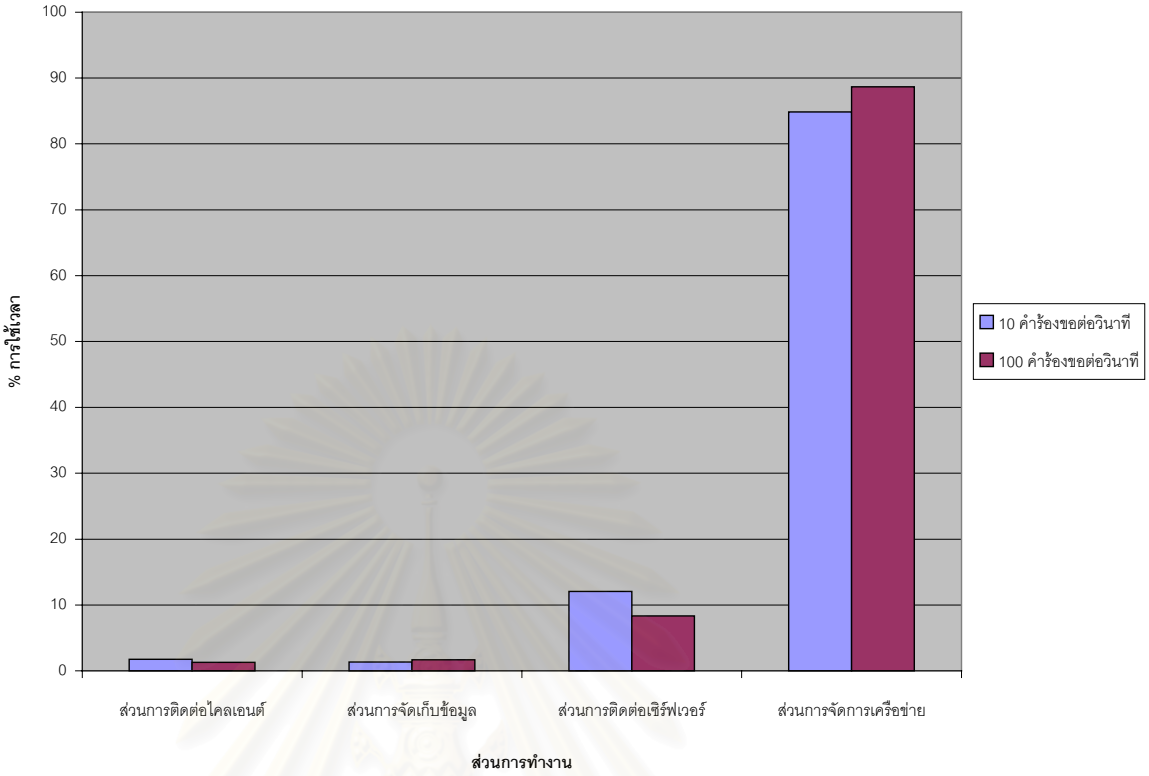
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



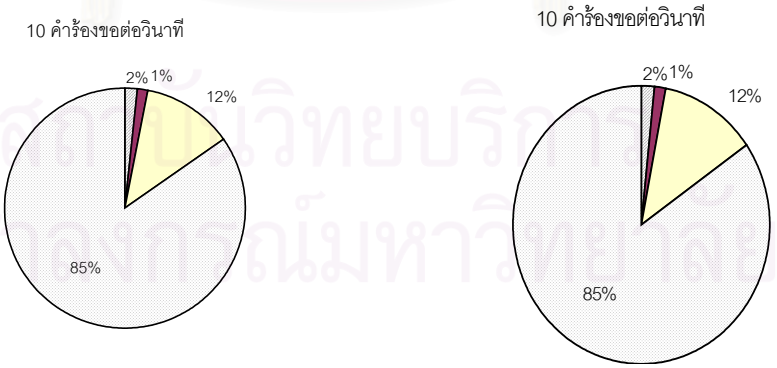
รูปที่ 5.1 กราฟแสดงการใช้เวลาในแต่ละส่วนของสควิดในการให้บริการรับคำร้องขอ ข้อมูลจากไคลเอนต์ในการทดลองกรณีเลียนแบบการทดลองซองเบคออฟ (ผ่านเครือข่าย)

จากกราฟรูปที่ 5.1 แสดงให้เห็นการใช้เวลาในแต่ละส่วนโดยภาพรวม เมื่อพิจารณาแล้วจะเห็นได้ว่าส่วนการจัดการเครือข่ายเป็นส่วนที่มีการใช้เวลามากที่สุด รองลงมาได้แก่ส่วนเซิร์ฟเวอร์, ส่วนการจัดเก็บข้อมูลและส่วนไคลเอนต์เป็นลำดับสุดท้าย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.2 แสดงการเปรียบเทียบการใช้เวลาในภาพรวมของแต่ละส่วนของการทำงานในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที และ 100 คำร้องขอต่อวินาที



รูปที่ 5.3 แสดงเปอร์เซ็นต์การใช้เวลาในภาพรวมของแต่ละส่วนในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที และ 100 คำร้องขอต่อวินาที

จากกราฟรูปที่ 5.2 และ กราฟรูปที่ 5.3 แสดงเวลาและเปอร์เซ็นต์การใช้เวลาในภาพรวมกรณีมีเครือข่าย ซึ่งโดยเฉลี่ยแล้ว ส่วนการจัดการเครือข่ายมีการใช้เวลาและเปอร์เซ็นต์การใช้เวลาในแต่ละอัตราการร้องขอมากกว่าส่วนอื่น ๆ และมีแนวโน้มเพิ่มขึ้นเมื่อมีอัตราการร้องขอเพิ่มมากขึ้น และมีส่วนเซิร์ฟเวอร์เป็นส่วนที่มีการใช้เวลาและเปอร์เซ็นต์การใช้เวลาเป็นลำดับรองลงมาตามด้วยส่วนการจัดเก็บข้อมูล และส่วนไคลเอนต์เป็นลำดับสุดท้าย

เมื่อพิจารณาหน้าที่การทำงานในส่วนดังกล่าวแล้ว อธิบายได้ดังนี้

ส่วนการจัดการเครือข่าย เมื่อพิจารณาในส่วนนี้ ซึ่งทำหน้าที่เกี่ยวกับการติดต่อระหว่างไคลเอนต์กับสควิด และสควิดกับเซิร์ฟเวอร์ปลายทาง โดยมีการสร้างส่วนติดต่อระหว่างกัน คือ สร้างซ็อกเก็ตเพื่อใช้ในการติดต่อ จำกัดการติดต่อโดยตรวจสอบไม่ให้เกินกว่าที่กำหนดไว้ ยกเลิกการติดต่อเมื่อการทำงานเสร็จสิ้นหรือมีการยกเลิกการติดต่อระหว่างกัน เมื่อพิจารณากระบวนการทำงานดังกล่าว ซึ่งจะต้องมีการติดต่อกันตลอดระยะเวลาในการส่งผ่านข้อมูลต่าง ๆ ระหว่างกัน มีการอ่าน/บันทึกข้อมูลที่ใช้ติดต่อกัน ซึ่งถือได้ว่าเป็นส่วนหลักในการติดต่อสื่อสารกันภายในเครือข่าย ซึ่งในแต่ละขั้นตอนต้องใช้เวลาการทำงานเพื่อจัดการกับข้อมูลส่วนที่รับเข้ามาใหม่และจัดการกับข้อมูลที่ยังมีการติดต่ออยู่ เมื่อมีการติดต่อเข้ามามากในช่วงระยะเวลาเดียวกัน ส่วนนี้ก็จะต้องทำงานเพิ่มขึ้นเพื่อรับการติดต่อที่เพิ่มจำนวนขึ้นในเวลาเดียวกัน ในขณะที่ลักษณะการทำงานก็ยังคงมีขั้นตอนการทำงานเช่นเดิม กล่าวคือจะต้องมีการสร้างส่วนการติดต่อ ต้องคอยตรวจเช็ค, ควบคุมการติดต่อให้เป็นไปด้วยความเรียบร้อยจนกว่าจะมีการยุติการติดต่อ ซึ่งก็หมายถึง ผู้ใช้ได้รับข้อมูลที่ร้องขอเป็นที่เรียบร้อยแล้ว งานนั้นจึงจะถือว่าเสร็จสมบูรณ์

ส่วนการติดต่อเซิร์ฟเวอร์ เป็นส่วนที่มีกระบวนการไปติดต่อกับเซิร์ฟเวอร์ปลายทาง ต้องมีกระบวนการต่าง ๆ เริ่มตั้งแต่ พิจารณาว่าจะส่งไปยังเซิร์ฟเวอร์ปลายทางเครื่องใด แล้วจึงเริ่มกระบวนการติดต่อ, การส่งคำร้องขอ, การรอรับการตอบกลับ, การรับข้อมูล จากเซิร์ฟเวอร์ปลายทาง ซึ่งในกรณีที่ไม่พบข้อมูลที่ร้องขอในแคช ส่วนนี้จะต้องทำหน้าที่รับผิดชอบส่งคำร้องขอต่อไปยังเซิร์ฟเวอร์ปลายทาง ซึ่งเมื่อพิจารณาจากหน้าที่การทำงานแล้ว ในส่วนนี้จะกระทำก็ต่อเมื่อข้อมูลที่ไคลเอนต์ร้องขอเข้ามาไม่พบในแคช ดังนั้นเมื่อมีการร้องขอเข้ามาในช่วงระยะเวลาเดียวกันเป็นจำนวนมาก โอกาสที่ไม่พบข้อมูลในแคชจึงมีมากกว่า ทั้งนี้ก็เนื่องจากรับคำร้องขอไม่ไหว ดังจะเห็นได้จากที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที ส่วนนี้ใช้เวลาน้อยกว่าที่อัตราการร้องขอ 100 คำร้องขอต่อวินาที แต่ในส่วนนี้ก็จะไม่ใช้เวลามากกว่าส่วนการจัดการเครือข่าย เนื่องจากจะทำงานเมื่อคำร้องขอข้อมูลของไคลเอนต์นั้น ๆ ไม่พบข้อมูลในแคช แต่ในขณะที่ส่วนการจัดการเครือข่ายจะต้องควบคุมดูแลทุก ๆ คำร้องขอที่รับเข้ามา

ส่วนการจัดเก็บข้อมูล ส่วนนี้จะเกี่ยวข้องกับกระบวนการอ่านข้อมูลจากดิสก์ การจัดเก็บข้อมูลลงดิสก์ การจัดการพื้นที่เพื่อใช้จัดเก็บข้อมูล เมื่อพิจารณาในส่วนนี้ซึ่งน่าจะเป็นส่วนที่เกิดกระบวนการคอขวดได้มากกว่าส่วนอื่น ๆ แต่พิจารณาจากการจัดเก็บข้อมูล การถ่ายโอนข้อมูลแล้วสควิดจะจัดเก็บข้อมูลบางส่วนลงดิสก์ ส่วนข้อมูลที่มีการเรียกใช้บ่อย ๆ จะเก็บไว้ที่แรม แบ่งการจัดเก็บข้อมูลในลักษณะเป็นไดเรกทอรีย่อย ๆ ข้อมูลที่จัดเก็บผ่านแฮชฟังก์ชันซึ่งทำให้มีขนาดเล็กลง ค้นหาข้อมูลได้รวดเร็ว และในกระบวนการจองพื้นที่หน่วยความจำใช้ฟังก์ชันที่พัฒนาขึ้นใหม่คือ xmalloc() จากเดิมใช้ ฟังก์ชัน mallloc() ซึ่งทำให้ทำงานมีประสิทธิภาพมากกว่า กล่าวคือ มีการคืนหน่วยความจำในกรณีที่เลิกใช้แล้วได้ดีกว่า ช่วยลดปัญหาการไม่ยอมคืนหน่วยความจำ (Memory Leak)

ส่วนการติดต่อไคลเอนต์ พิจารณาจากหน้าที่หลัก ๆ จะเกี่ยวข้องกับไคลเอนต์โดยตรง คือ จะรับคำร้องขอที่ส่งเข้ามา แล้วพิจารณาคำร้องขอ ตรวจสอบสิทธิของไคลเอนต์ที่ส่งคำร้องขอและทำการประมวลผลข้อมูลที่ร้องขอเข้ามา พิจารณาจากหน้าที่หลัก ๆ จะเกี่ยวข้องกับไคลเอนต์โดยตรง คือ จะรับคำร้องขอที่ส่งเข้ามา แล้วพิจารณาคำร้องขอ ตรวจสอบสิทธิของไคลเอนต์ที่ส่งคำร้องขอ และทำการประมวลผลข้อมูลที่ร้องขอเข้ามา ถึงแม้ว่าในส่วนนี้จะพิจารณาคำร้องขอทั้งหมดที่ส่งเข้ามา แต่เปอร์เซ็นต์การใช้เวลาก็ไม่มากกว่าส่วนอื่น ทั้งนี้เนื่องจากการทำงานในกระบวนการต่าง ๆ ถือได้ว่าเป็นจุดเริ่มต้นของการให้บริการ จึงต้องมีการตรวจสอบข้อมูลอย่างรวดเร็วเพื่อที่จะรับการติดต่อของแต่ละไคลเอนต์ จึงมีการจัดเก็บข้อมูลบางส่วนไว้ ซึ่งถ้าหากมีการติดต่อเข้ามาใหม่ แต่เป็นไคลเอนต์คนเดิม ก็สามารถตรวจสอบข้อมูลได้อย่างรวดเร็วถึงแม้อัตราการร้องขอจะเพิ่มขึ้นก็ตาม ดังจะเห็นได้จากเวลาที่ใช้ในระดับการร้องขอ 10 คำร้องขอต่อวินาที ไม่แตกต่างมากนักกับระดับอัตราการร้องขอ 100 คำร้องขอต่อวินาที อาจกล่าวได้ว่าทำงานในส่วนนี้ไม่มีผลต่อการทำงานของโปรแกรมสควิดมากนัก เมื่อเทียบกับการทำงานในส่วนอื่น ๆ เนื่องจากเวลาที่ใช้น้อยมากเมื่อเทียบกับส่วนอื่น ๆ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5.2 ผลการทดลองในกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) โดยพิจารณาในระดับฟังก์ชัน

จากการพิจารณาในภาพรวมของส่วนการทำงานที่กล่าวมาแล้วนั้น ในส่วนนี้จะพิจารณาการใช้เวลาของแต่ละฟังก์ชันในส่วนดังกล่าว เพื่อจะได้ทราบว่าฟังก์ชันใดที่มีการใช้เวลามากที่สุด ข้อมูลการทดลองแสดงได้ดังตารางที่ 5.3 และ 5.4

- **ข้อมูลการทดลองส่วนการจัดการเครือข่ายกรณีผ่านเครือข่าย**

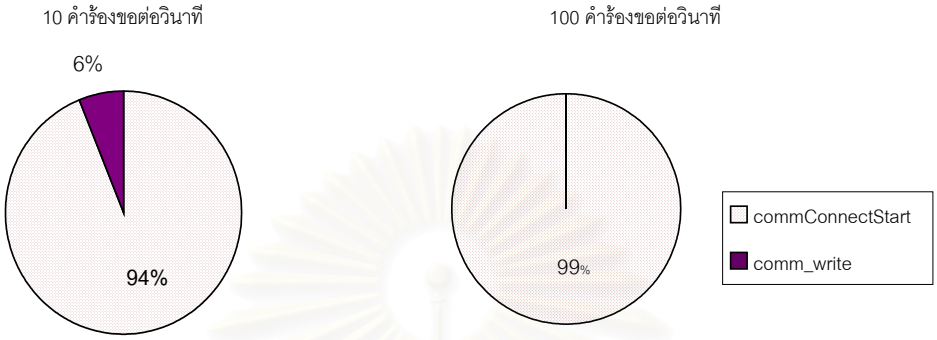
ตารางที่ 5.3 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการจัดการเครือข่ายโดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / ฟังก์ชันการทำงาน	10	20	30	40	50	60	70	80	90	100
comm_open	1.14	1.22	1.59	1.53	1.68	1.39	1.43	1.36	1.44	1.26
clientdb Established	0.21	0.14	0.18	0.20	0.08	0.20	0.19	0.17	0.20	0.31
commConnect Start	3657.04	5463.01	6531.63	5838.21	7045.84	7100.41	7023.95	7316.30	6304.25	6069.66
commConnect Handle	0.27	0.24	0.35	0.32	0.44	0.34	0.33	0.24	0.13	0.14
comm_write	240.87	617.91	1230.99	1714.79	1907.56	1569.46	746.49	201.89	0.07	0.46

ตารางที่ 5.4 แสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดการเครือข่ายกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / ฟังก์ชันการทำงาน	10	20	30	40	50	60	70	80	90	100
comm_open	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
clientdbEstablished	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
commConnectStart	93.78	89.82	84.12	77.28	78.68	81.88	90.37	97.29	99.97	99.96
commConnectHandle	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
comm_write	6.18	10.16	15.85	22.70	21.30	18.10	9.60	2.68	0.00	0.01

จากข้อมูลในตารางที่ 5.4 พิจารณาเปอร์เซ็นต์การใช้เวลาในอัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที นำเสนอเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันในส่วนการจัดการเครือข่าย ได้ดังกราฟรูปที่ 5.4



รูปที่ 5.4 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนการจัดการเครือข่ายในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟผ่านเครือข่ายที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที และ 100 คำร้องขอต่อวินาที

จากกราฟรูปที่ 5.4 แสดงให้เห็นว่าในส่วนการจัดการเครือข่ายนั้น โดยเฉลี่ยแล้วฟังก์ชัน commConnectStart จะมีการใช้เวลามากกว่าฟังก์ชันอื่น ๆ ตั้งแต่อัตราการร้องขอในระดับแรกและในระดับอัตราการร้องขอที่เพิ่มขึ้น ทั้งนี้เมื่อพิจารณาหน้าที่การทำงานของฟังก์ชันนี้ จะพบว่าเป็นฟังก์ชันที่ทำหน้าที่ในการควบคุมการติดต่อระหว่างสควิดกับไคลเอนต์และสควิดกับเซิร์ฟเวอร์ปลายทาง โดยในการติดต่อกันนั้นจะคงอยู่ไปจนกว่าจะเสร็จสิ้นกระบวนการทำงานต่าง ๆ จึงส่งผลให้ฟังก์ชันนี้มีอัตราการใช้เวลามากกว่าฟังก์ชันอื่น ๆ ในส่วนนี้

- **ข้อมูลการทดลองส่วนการติดต่อเซิร์ฟเวอร์กรณีผ่านเครือข่าย**

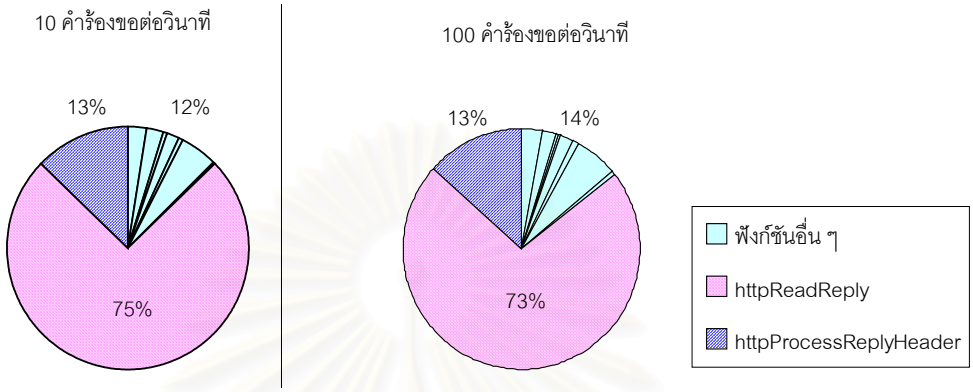
ตารางที่ 5.5 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการติดต่อเซิร์ฟเวอร์โดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / ฟังก์ชันการทำงาน	10	20	30	40	50	60	70	80	90	100
fwdStart	13.12	12.91	14.17	13.99	14.76	13.43	14.58	13.73	14.87	15.29
fwdStartComplete	0.34	0.16	0.19	0.19	0.23	0.31	0.19	0.22	0.10	0.22
fwdConnectStart	12.41	11.97	13.37	13.27	13.75	12.77	12.79	12.15	12.29	12.22
fwdDispatch	0.34	0.66	1.27	0.81	1.12	1.04	0.70	0.87	1.21	1.02
httpStart	2.50	1.78	2.09	2.29	2.41	2.08	2.23	2.57	2.25	2.12
httpBuildRequestPrefix	10.48	10.31	11.67	11.66	12.05	11.01	11.40	10.74	11.44	10.96
httpSendRequest	3.03	3.48	4.11	4.56	4.38	4.43	4.43	4.50	4.79	4.49
httpBuildRequestHeader	26.36	32.77	32.93	36.97	35.00	35.70	34.90	32.65	33.98	33.59
httpSendComplete	1.71	2.23	2.06	2.58	2.38	2.85	2.48	2.88	2.74	2.78
httpReadReply	413.45	343.66	467.28	457.04	453.41	417.64	415.73	393.32	401.09	412.98
httpProcessReplyHeader	70.32	69.82	73.34	71.76	74.62	75.43	75.20	73.48	75.90	76.09

ตารางที่ 5.6 แสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการติดต่อเซิร์ฟเวอร์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / ฟังก์ชันการทำงาน	10	20	30	40	50	60	70	80	90	100
fwdStart	2.37	2.63	2.28	2.27	2.40	2.33	2.54	2.51	2.65	2.68
fwdStartComplete	0.06	0.03	0.03	0.03	0.04	0.05	0.03	0.04	0.02	0.04
fwdConnectStart	2.24	2.44	2.15	2.16	2.24	2.22	2.23	2.22	2.19	2.14
fwdDispatch	0.06	0.20	0.20	0.13	0.18	0.18	0.12	0.16	0.22	0.18
httpStart	0.45	0.36	0.34	0.37	0.39	0.36	0.39	0.47	0.40	0.37
httpBuildRequestPrefix	1.89	2.10	1.87	1.90	1.96	1.91	1.98	1.96	2.04	1.92
httpSendRequest	0.55	0.71	0.66	0.74	0.71	0.77	0.77	0.82	0.86	0.78
httpBuildRequestHeader	4.76	6.69	5.29	6.01	5.70	6.19	6.07	5.97	6.06	5.88
httpSendComplete	0.31	0.46	0.33	0.42	0.39	0.49	0.43	0.53	0.49	0.49
httpReadReply	74.62	70.13	75.07	74.30	73.83	72.42	72.35	71.89	71.54	72.23
httpProcessReplyHeader	12.69	14.25	11.78	11.67	12.15	13.08	13.09	13.43	13.54	13.31

จากข้อมูลในตารางที่ 5.6 พิจารณาเปอร์เซ็นต์การใช้เวลาในอัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที นำเสนอเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันในส่วนเซิร์ฟเวอร์ ได้ดังกราฟรูปที่ 5.5



รูปที่ 5.5 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนการติดต่อเซิร์ฟเวอร์ในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที

จากกราฟรูปที่ 5.5 แสดงให้เห็นถึงเปอร์เซ็นต์การใช้เวลาในฟังก์ชันการทำงานส่วนการติดต่อเซิร์ฟเวอร์ จะเห็นได้ว่าฟังก์ชัน httpReadReply โดยเฉลี่ยแล้วมีการใช้เวลามากกว่าฟังก์ชันอื่น ๆ ตั้งแต่ในระดับแรกของอัตราการร้องขอและในระดับอัตราการร้องขอที่เพิ่มขึ้น เมื่อพิจารณาหน้าที่การทำงานของฟังก์ชันนี้ จะพบว่าเป็นฟังก์ชันที่ทำหน้าที่ในการรับคำตอบกลับจากเซิร์ฟเวอร์ปลายทาง กล่าวคือ เมื่อมีการส่งคำร้องขอไปยังเซิร์ฟเวอร์ปลายทางเพื่อร้องขอข้อมูลที่ไม่พบในแคช ต้องทำการตรวจสอบเซิร์ฟเวอร์ที่จะติดต่อเพื่อร้องขอข้อมูลก่อน ซึ่งเมื่อทราบว่าจะติดต่อกับเซิร์ฟเวอร์เครื่องใดแล้ว ก็จะทำการติดต่อไปยังเซิร์ฟเวอร์ตัวนั้นก่อน และเมื่อติดต่อได้แล้วจะต้องมีกระบวนการอ่านข้อมูลที่ส่งมาจนกระทั่งเสร็จสิ้นการรับ-ส่งข้อมูลที่ร้องขอไป ซึ่งจะต้องมีการตรวจสอบส่วนเฮดเดอร์ที่จัดส่งมา, ตรวจสอบว่าข้อมูลที่ส่งมาจัดเก็บได้หรือไม่ และตรวจรับข้อมูลที่จะจัดส่งมาทั้งหมดจนเสร็จเรียบร้อย ทั้งนี้ต้องอยู่ภายในเวลาที่กำหนดไว้ด้วย ทำให้ฟังก์ชันนี้ใช้เวลามากกว่าฟังก์ชันอื่น ๆ

- **ข้อมูลการทดลองส่วนการจัดเก็บข้อมูลกรณีผ่านเครือข่าย**

ตารางที่ 5.7 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการจัดเก็บข้อมูลโดยเฉลี่ยต่อหนึ่งคำร้องขอกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

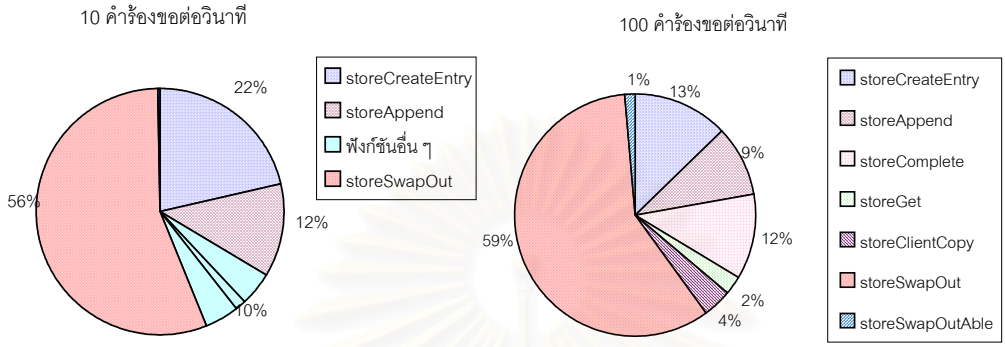
อัตราการร้องขอ / ฟังก์ชันการทำงาน	10	20	30	40	50	60	70	80	90	100
storeCreateEntry	13.14	13.84	12.83	12.95	13.69	14.39	14.32	14.11	14.51	14.79
StoreAppen	7.44	12.18	14.76	14.96	13.97	11.36	11.40	8.06	10.49	10.78
storeComplete	2.72	4.70	5.85	7.82	8.49	9.20	11.55	12.33	12.60	13.35
storeGet	0.83	0.92	1.27	1.66	1.71	1.73	2.00	2.14	2.53	2.78
storeClientCopy	2.66	5.08	9.00	5.90	5.41	4.27	4.76	4.69	4.38	4.39
storeSwapOut	34.11	54.27	59.32	65.17	64.32	62.89	66.48	62.58	63.11	67.94
storeSwapOutAble	0.19	1.32	1.04	1.93	1.39	1.24	1.28	0.70	0.96	1.50

ตารางที่ 5.8 แสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดเก็บข้อมูลกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / ฟังก์ชันการทำงาน	10	20	30	40	50	60	70	80	90	100
storeCreateEntry	21.51	14.99	12.33	11.73	12.56	13.70	12.81	13.49	13.364	12.81
storeAppend	12.18	13.19	14.18	13.55	12.82	10.81	10.20	7.70	9.658	9.34
storeComplete	4.45	5.09	5.62	7.08	7.79	8.76	10.33	11.79	11.605	11.56
storeGet	1.36	0.99	1.22	1.50	1.57	1.64	1.79	2.05	2.328	2.40
storeClientCopy	4.36	5.50	8.65	5.36	4.97	4.06	4.25	4.48	4.038	3.80
storeSwapOut	55.84	58.78	57.00	59.03	59.02	59.86	59.48	59.83	58.126	58.82
storeSwapOutAble	0.31	1.45	0.99	1.74	1.28	1.18	1.15	0.66	0.882	1.30

จุฬาลงกรณ์มหาวิทยาลัย

จากข้อมูลในตารางที่ 5.8 พิจารณาเปอร์เซ็นต์การใช้เวลาในอัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที นำเสนอเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันในส่วนการจัดเก็บข้อมูล ได้ดังกราฟรูปที่ 5.6



รูปที่ 5.6 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนการจัดเก็บข้อมูลในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ10 คำร้องขอต่อวินาที และ 100 คำร้องขอต่อวินาที

จากกราฟรูปที่ 5.6 แสดงให้เห็นว่าในส่วนการจัดเก็บข้อมูลนั้น โดยเฉลี่ยแล้วฟังก์ชัน storeSwapOut มีเปอร์เซ็นต์การใช้เวลามากกว่าฟังก์ชันอื่น ๆ ในส่วนนี้ เริ่มตั้งแต่อัตราการร้องขอในระดับแรกและมีแนวโน้มการใช้เวลาเพิ่มขึ้นในระดับการร้องขอที่เพิ่มขึ้น ซึ่งที่ระดับอัตราการร้องขอ 10 คำร้องขอต่อวินาที ฟังก์ชันดังกล่าวมีเปอร์เซ็นต์ในการใช้เวลาคิดเป็น 56 เปอร์เซ็นต์ และที่ระดับอัตราการร้องขอที่ 100 คำร้องขอต่อวินาทีมีเปอร์เซ็นต์ในการใช้เวลาในฟังก์ชันดังกล่าวเพิ่มขึ้นเป็น 59 เปอร์เซ็นต์ ทั้งนี้เมื่อพิจารณาหน้าที่การทำงานของฟังก์ชันนี้ จะพบว่าเป็นฟังก์ชันที่ทำหน้าที่ในการบันทึกข้อมูลลงดิสก์ กล่าวคือ เมื่อมีการส่งข้อมูลมาจากเซิร์ฟเวอร์ปลายทาง และสควิดส่งข้อมูลให้กับผู้ใช้แล้ว สควิดจะทำการบันทึกข้อมูลลงดิสก์ ซึ่งถ้าในกรณีที่มีจำนวนข้อมูลที่ร้องขอไม่พบในแคชทำให้ต้องมีการจัดเก็บข้อมูลนั้น ซึ่งต้องมีการตรวจสอบข้อมูลเพื่อจัดหาพื้นที่สำหรับจัดเก็บข้อมูลดังกล่าว ทำให้ต้องใช้เวลาในส่วนนี้มากกว่าส่วนอื่น ๆ ซึ่งในส่วนนี้ช่วยสนับสนุนงานวิจัยการเกิดปัญหาคอขวดในส่วนของการอ่านและบันทึกข้อมูลลงดิสก์ [7] ควรที่จะเพิ่มขนาดของแรมหรือหน่วยพักข้อมูลให้มีขนาดเพิ่มมากขึ้น เพื่อที่จะสามารถเข้าถึงข้อมูลได้เร็วขึ้นและจัดเก็บข้อมูลได้มากขึ้น

- **ข้อมูลการทดลองส่วนการติดต่อไคลเอนต์กรณีผ่านเครือข่าย**

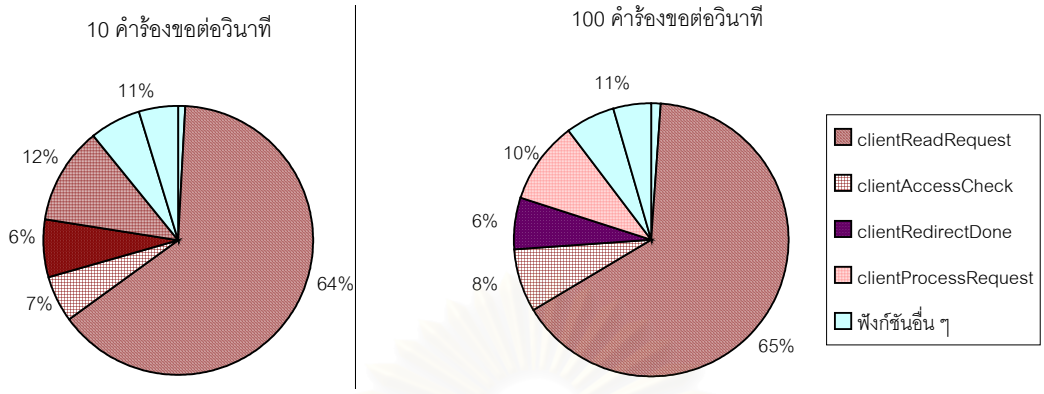
ตารางที่ 5.9 แสดงการใช้เวลา (ไมโครวินาที) ของฟังก์ชันในส่วนการติดต่อไคลเอนต์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / ฟังก์ชันการทำงาน	10	20	30	40	50	60	70	80	90	100
clientHttpConnectionOpen	0.00006	0.00003	0.00003	0.00002	0.00002	0.00001	0.00001	0.00001	0.00001	0.00001
httpAccept	0.58	1.13	0.86	0.90	0.86	0.93	0.83	0.93	0.94	0.95
clientReadRequest	52.12	30.67	56.79	59.44	62.35	58.69	58.76	57.89	59.71	58.32
clientAccessCheck	4.59	7.83	4.54	4.56	5.43	5.34	5.63	5.92	6.47	6.70
clientRedirectDone	5.53	6.11	5.94	5.67	5.99	5.29	5.25	5.76	5.85	5.45
clientProcessRequest	9.63	6.86	8.47	8.02	8.63	8.83	8.39	8.2	8.65	8.70
clientCacheHit	4.97	4.99	4.34	4.23	6.37	5.78	5.23	6.03	5.12	5.26
clientProcessMiss	3.72	5.16	3.81	4.63	3.35	3.55	4.18	3.71	3.59	3.87

ตารางที่ 5.10 แสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการติดต่อไคลเอนต์กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

อัตราการร้องขอ / ฟังกชันการทำงาน	10	20	30	40	50	60	70	80	90	100
clientHttpConnection Open	0.00007	0.00005	0.00003	0.00002	0.00001.	0.00001	0.00001	0.00001	0.00001	0.00001
httpAccept	0.72	1.80	1.02	1.03	0.93	1.05	0.94	1.05	1.04	1.06
clientReadRequest	64.23	48.87	67.01	67.97	67.06	66.39	66.57	65.46	66.10	65.34
clientAccessCheck	5.66	12.48	5.36	5.21	5.84	6.04	6.38	6.69	7.16	7.51
clientRedirectDone	6.82	9.73	7.01	6.48	6.44	5.98	5.95	6.51	6.48	6.11
clientProcess Request	11.87	10.93	9.99	9.17	9.28	9.99	9.50	9.27	9.58	9.75
clientCacheHit	6.13	7.96	5.12	4.84	6.85	6.54	5.92	6.82	5.67	5.90
clientProcessMiss	4.58	8.23	4.50	5.29	3.60	4.02	4.74	4.20	3.97	4.34

จากข้อมูลในตารางที่ 5.10 พิจารณาเปอร์เซ็นต์การใช้เวลาในอัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที นำเสนอเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันในส่วนการติดต่อไคลเอนต์ ได้ดังกราฟรูปที่ 5.7



รูปที่ 5.7 แสดงการเปรียบเทียบเปอร์เซ็นต์การใช้เวลาในแต่ละฟังก์ชันของส่วนการติดต่อไคลเอนต์ในการ ให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิดกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีและ 100 คำร้องขอต่อวินาที

จากกราฟรูปที่ 5.7 แสดงให้เห็นว่าที่อัตราการร้องขอที่เพิ่มมากขึ้นในส่วนการติดต่อไคลเอนต์ โดยฟังก์ชัน clientReadRequest มีเปอร์เซ็นต์การใช้เวลามากกว่าฟังก์ชันอื่น ๆ ในส่วนนี้ ซึ่งจะเห็นได้ชัดโดยเริ่มตั้งแต่อัตราการร้องขอในระดับแรกและมีแนวโน้มเพิ่มขึ้นในระดับการร้องขอที่เพิ่มขึ้น ซึ่งพิจารณาได้จากการทดลองที่ระดับอัตราการร้องขอ 10 คำร้องขอต่อวินาทีและที่ระดับคำร้องขอ 100 คำร้องขอต่อวินาที มีเปอร์เซ็นต์การใช้เวลาเพิ่มขึ้น จาก 64 เปอร์เซ็นต์ เป็น 65 เปอร์เซ็นต์ ทั้งนี้เมื่อพิจารณาหน้าที่การทำงานของฟังก์ชันนี้ จะพบว่าเป็นฟังก์ชันที่ทำหน้าที่หลักในส่วนนี้ กล่าวคือ เมื่อมีการรับคำร้องขอเข้ามาจะต้องมีการตรวจสอบคำร้องขอของผู้ใช้ว่าถูกต้องหรือไม่ ซึ่งจะต้องมีการตรวจสอบโพรโทคอลที่ใช้ วิธีการที่ร้องขอ ยูอาร์แอลที่ร้องขอ ตรวจสอบสิทธิของผู้ใช้ โดยจะตรวจสอบจากเลขที่อยู่ ชื่อเครื่องว่าอยู่ในกลุ่มที่ได้รับสิทธิหรือไม่ มีกระบวนการไปเรียกใช้ฟังก์ชันการทำงานค่อนข้างมากกว่าฟังก์ชันอื่น ๆ เพื่อตรวจเช็คค่าต่าง ๆ จึงทำให้ใช้เวลามากกว่าฟังก์ชันอื่น ๆ ซึ่งหากจะช่วยลดเวลาการทำงานในส่วนนี้ อาจจะมีการเพิ่มการแคชข้อมูลที่มีการใช้บ่อย ๆ ในกระบวนการทำงานที่ซ้ำ ๆ คงเดิม เช่น ในกรณีที่ไคลเอนต์คนเดิมร้องขอข้อมูลเดิมเข้ามา

จากผลการทดลองโดยภาพรวมกรณีทดลองผ่านเครือข่ายดังกล่าวข้างต้นนั้น เมื่อพิจารณาเวลาที่ใช้โดยเฉลี่ยต่อหนึ่งคำร้องขอจากแฟ้มบันทึกเข้าออกของสควิดเปรียบเทียบกับเวลารวมโดยเฉลี่ยของการทำงานทั้งสี่ส่วนที่ได้จากจากแฟ้มบันทึกเข้าออกของจีพร็อบพบว่าเวลาที่ใช้ในการทำงานของสควิดนั้นน้อยมาก เวลาส่วนที่เหลือจึงน่าจะเป็นเวลาการรอคอย (Wait time) การทำงานในส่วนของเครือข่ายหรือเรียกว่าเป็นโอเวอร์เฮดของเครือข่าย โดยโอเวอร์เฮดของเครือข่ายจะแบ่งเป็นส่วนต่าง ๆ ดังนี้ โดยพิจารณาโอเวอร์เฮดที่เกิดจากชั้นการสื่อสารข้อมูลของเครือข่ายซึ่งโดยมาตรฐานแล้วจะแบ่งออกเป็น 7 ชั้นการสื่อสาร [15] แต่ในที่นี่โพรโทคอลที่นิยมใช้ในเครือข่ายอินเทอร์เน็ตได้แก่โพรโทคอลทีซีพีไอพีนั้นได้จัดแบ่งชั้นการสื่อสารในการติดต่อสื่อสารระหว่างเครื่องออกเป็น 4 ชั้นการสื่อสาร [15] โดยมีการรวมชั้นการสื่อสารของรูปแบบมาตรฐานการติดต่อสื่อสารบางชั้นเข้าด้วยกัน เพื่อให้การทำงานง่ายขึ้นและไม่ซับซ้อน สามารถเชื่อมต่อระหว่างระบบที่มีความแตกต่างกันได้ ชั้นการสื่อสารที่มี ได้แก่

- ชั้นสื่อสารการประยุกต์ (Application Layer) ทำหน้าที่ติดต่อกับโปรแกรมประยุกต์ที่ใช้ และโปรแกรมในชั้นที่เหลือของเครื่องที่ติดต่อระหว่างกันเพื่อให้เกิดความเข้าใจตรงกันในการติดต่อสื่อสาร ทำให้ผู้ใช้สามารถติดต่อไปใช้อีกเครื่องหนึ่งได้
- ชั้นสื่อสารนำส่งข้อมูล (Transport Layer) ซึ่งจะทำหน้าที่ควบคุมการไหลของข้อมูลไม่ให้ผู้ส่งส่งเร็วเกินกว่าที่ผู้รับจะทำงานได้ทัน และตรวจสอบข้อมูลที่จัดส่งว่าถูกต้องและครบถ้วนหรือไม่
- ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer) ทำหน้าที่ตรวจสอบที่อยู่ที่จะส่งข้อมูล จัดหาเส้นทางเพื่อจัดส่งข้อมูลไปยังเครื่องที่ระบุไว้ตามที่อยู่
- ชั้นติดต่ออุปกรณ์เครือข่าย (Network Interface Layer) จัดการส่งข้อมูลไปยังเครื่องที่ระบุไว้ในเส้นทางเดินข้อมูล มีการแปลงข้อมูล จัดรูปแบบข้อมูล และติดต่อกับอุปกรณ์สื่อสารข้อมูลโดยตรง เพื่อส่งผ่านข้อมูลถึงกัน

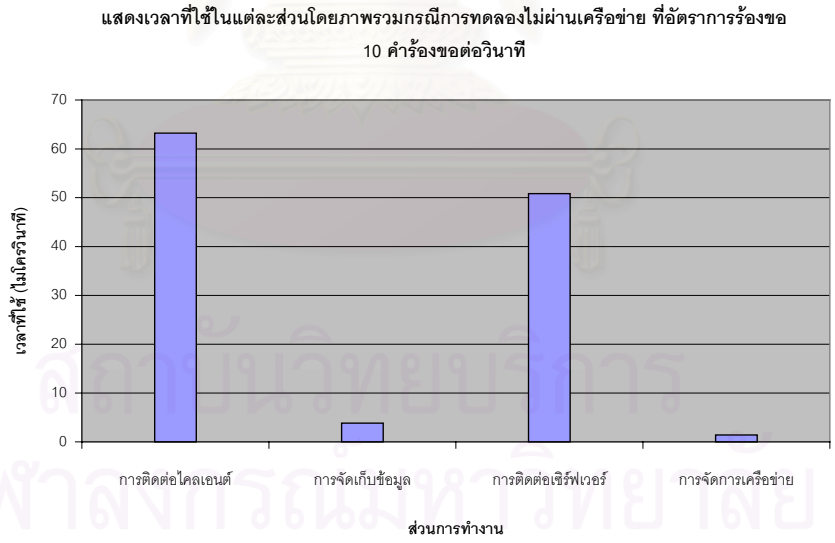
ในการติดต่อกันระหว่างเครื่องหรือเรียกว่าติดต่อกันผ่านเครือข่ายนี้ จะต้องมีการติดต่อกันตามชั้นสื่อสารดังที่กล่าว โดยผ่านโปรแกรมการทำงานซึ่งก็จะมีทั้งโปรแกรมส่วนที่ไคลเอนต์หรือเซิร์ฟเวอร์ใช้ และโปรแกรมส่วนที่เป็นระบบปฏิบัติการที่จะต้องคอยควบคุมการทำงานของอุปกรณ์ที่ใช้เพื่อติดต่อสื่อสารกันภายในเครือข่าย นอกจากนี้ยังมีโอเวอร์เฮดจากกระบวนการติดต่อสื่อสารของโพรโตคอลเอชทีทีพีที่ใช้สำหรับส่งคำร้องขอข้อมูลระหว่างไคลเอนต์กับเซิร์ฟเวอร์ที่เรียกว่า Three-Way Handshake ซึ่งหมายถึง เมื่อไคลเอนต์จะทำการติดต่อไปยังเซิร์ฟเวอร์ ในขั้นต้นนั้นไคลเอนต์จะต้องส่งสัญญาณไปยังเซิร์ฟเวอร์ก่อนเพื่อเป็นการบอกให้เซิร์ฟเวอร์ทราบว่าตนเองจะติดต่อไป จากนั้นไคลเอนต์ต้องรอสัญญาณตอบกลับจากเซิร์ฟเวอร์เพื่อแสดงว่าเซิร์ฟเวอร์ได้รับข้อมูลการติดต่อจากไคลเอนต์แล้ว ไคลเอนต์จะส่งสัญญาณการตอบกลับพร้อมทั้งคำร้องขอไปยัง

เซิร์ฟเวอร์ ขณะเดียวกันฝั่งเซิร์ฟเวอร์เองก็ต้องได้รับสัญญาณการตอบกลับจากไคลเอนต์เช่นกัน เพื่อเป็นการยืนยันว่าได้รับสัญญาณตอบกลับจากเซิร์ฟเวอร์แล้ว ในส่วนการส่งข้อมูลนั้นเซิร์ฟเวอร์ก็จะทำการส่งข้อมูลไปเรื่อย ๆ จนข้อมูลที่จัดส่งครบเรียบร้อยแล้ว และก็จะรอสัญญาณตอบรับจากไคลเอนต์ให้ครบในทุก ๆ ชุดข้อมูลที่ส่งไป เพื่อเป็นการยืนว่าไคลเอนต์ได้รับข้อมูลที่ส่งไปครบเรียบร้อยแล้ว จึงจะทำการยกเลิกการติดต่อกัน ดังนั้นในการติดต่อสื่อสารกันภายในเครือข่ายแล้วจะต้องมีการสูญเสียเวลาส่วนหนึ่งในกระบวนการทำงานดังกล่าวเสมอ

และจากการทดลองข้างต้นที่กล่าวมาพบว่ากระบวนการของการติดต่อภายในเครือข่ายมีผลต่อการทำงานของสควิด และเพื่อเป็นการตรวจสอบกระบวนการดังกล่าว ว่าเวลาที่ใช้มากขึ้นนั้น ส่วนหนึ่งเกิดจากเวลาของโอเวอร์เฮดของเครือข่าย จึงได้ทำการทดลองโดยเรียกว่าการทดลองกรณีไม่ผ่านเครือข่ายโดยการนำเครื่องคอมพิวเตอร์หนึ่งเครื่อง จำลองส่วนของไคลเอนต์, สควิด และเซิร์ฟเวอร์ในเครื่องเดียวกัน และทดลองโดยใช้อัตราการร้องขอที่ 10 คำร้องขอต่อวินาที เพื่อนำผลดังกล่าวมาเปรียบเทียบกับผลการทดลองในกรณีการทดลองผ่านเครือข่าย

5.3 ผลการทดลองและวิเคราะห์การใช้เวลาแต่ละส่วนในภาพรวมกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย)

พิจารณาผลการทดลองในภาพรวมแต่ละส่วน ในกรณีนี้ แสดงผลได้ดังกราฟรูปที่ 5.8



รูปที่ 5.8 กราฟแสดงการใช้เวลาในส่วนต่าง ๆ ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีกรณีการทดลองโดยใช้เครื่องคอมพิวเตอร์หนึ่งเครื่อง

จากกราฟรูปที่ 5.8 ในภาพรวมแสดงให้เห็นว่าส่วนการติดต่อไคลเอนต์เป็นส่วนที่มีการใช้เวลามากที่สุด รองลงมาได้แก่ส่วนการติดต่อเซิร์ฟเวอร์, ส่วนการจัดเก็บข้อมูล และส่วนการจัดการเครือข่ายซึ่งจะเห็นได้ว่าในกรณีไม่ผ่านเครือข่ายนี้ ส่วนที่เกี่ยวข้องกับการจัดการเครือข่ายจะมี

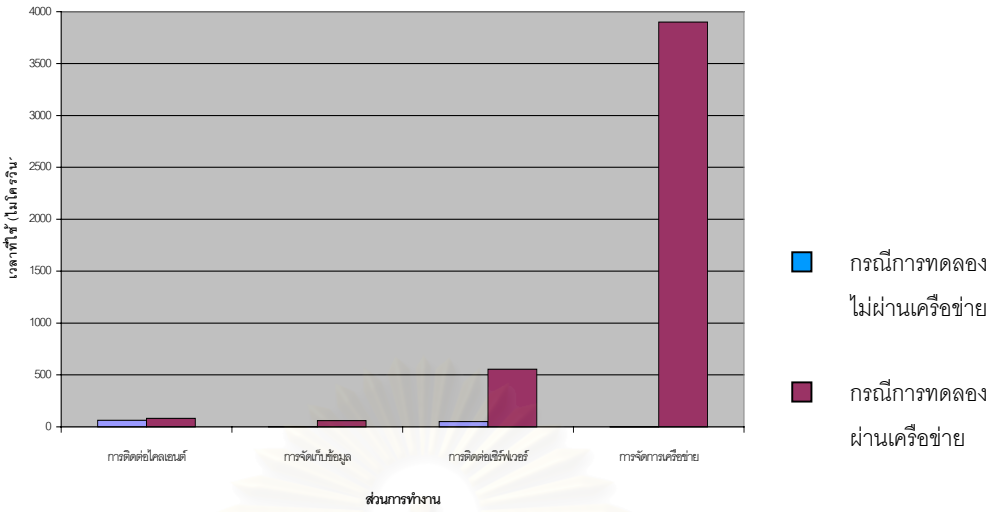
เปอร์เซ็นการใช้เวลาที่น้อยมาก เนื่องจากในการสร้างส่วนการติดต่อ การควบคุมการติดต่อต่าง ๆ ไม่ต้องส่งข้ามไปยังอีกเครื่องหนึ่ง ระยะเวลาในการรอในส่วนของเครือข่ายจึงลดลง เวลาที่ใช้ในส่วนการจัดการเครือข่ายจึงมีค่าน้อย ซึ่งชี้ให้เห็นว่าการทำงานของเครือข่ายมีผลต่อการใช้เวลาในการทำงานของส่วนต่าง ๆ ที่ต้องเกี่ยวข้องกับเครือข่าย แสดงข้อมูลเปรียบเทียบการใช้เวลาในภาพรวมของส่วนต่าง ๆ ดังกล่าวได้ดังตารางที่ 5.11

ตารางที่ 5.11 แสดงการเปรียบเทียบใช้เวลา (ไมโครวินาที) ในส่วนต่าง ๆ ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีกรณีการทดลองโดยใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย) และกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

ส่วนการทำงาน	**เวลาที่ใช้กรณีเลียนแบบการทดลองของเบคออฟ (ไมโครวินาที)**	**เวลาที่ใช้กรณีใช้เครื่องหนึ่งเครื่องในการจำลองการทำงาน (ไมโครวินาที)**
การติดต่อไคลเอนต์	63.21	81.15
การจัดเก็บข้อมูล	3.82	61.07
การติดต่อเซิร์ฟเวอร์	50.84	554.07
การจัดการเครือข่าย	1.41	3899.52

จากตารางที่ 5.11 นำข้อมูลที่ได้เสนอดังกราฟรูปที่ 5.9

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

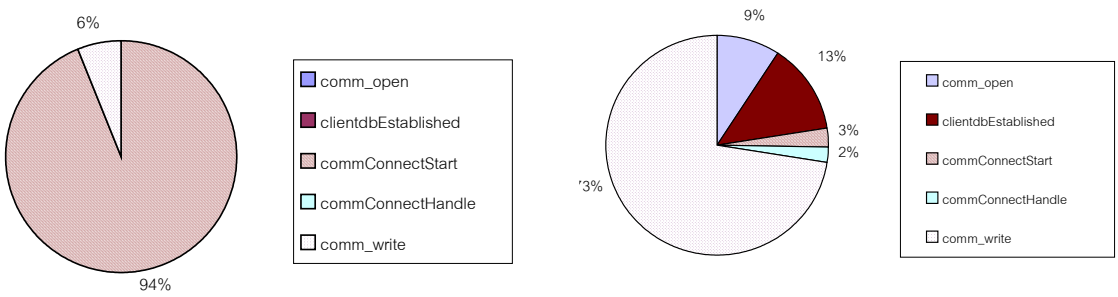


รูปที่ 5.9 กราฟเปรียบเทียบการใช้เวลา (ไมโครวินาที) ในภาพรวมของส่วนต่าง ๆ ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีในกรณีการทดลองโดยใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย) และกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

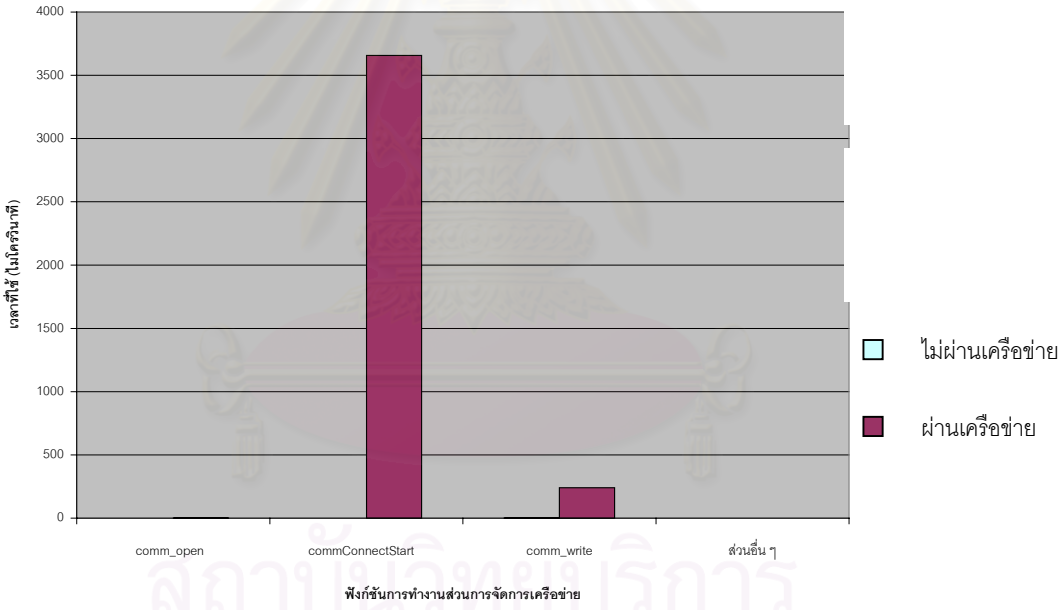
จากกราฟรูปที่ 5.9 แสดงให้เห็นว่าถึงแม้ในภาพรวมกรณีการทดลองผ่านเครือข่าย ส่วนการติดต่อไคลเอนต์จะเป็นส่วนที่มีการใช้เวลามากที่สุด แต่เมื่อเทียบส่วนนี้กับกรณีการทดลองไม่ผ่านเครือข่ายที่อัตราการร้องขอเดียวกันแล้ว พบว่าเวลาที่ใช้โดยเฉลี่ยแล้วแตกต่างกันไม่มากนัก แต่ในขณะที่ส่วนอื่น ๆ มีการใช้เพิ่มมากขึ้นอย่างเห็นได้ชัด ไม่ว่าจะเป็นส่วนการติดต่อเซิร์ฟเวอร์, ส่วนการจัดเก็บข้อมูล และส่วนการจัดการเครือข่าย โดยเฉพาะส่วนการจัดการเครือข่าย กรณีการทดลองไม่ผ่านเครือข่าย มีการใช้เวลาโดยเฉลี่ยเพียง 1.41 ไมโครวินาที ส่วนในกรณีการทดลองผ่านเครือข่ายใช้เวลาโดยเฉลี่ยถึง 3899.52 ไมโครวินาที ซึ่งแสดงให้เห็นว่าการทำงานของเครือข่ายมีผลต่อการทำงานของสควิด

5.4 ผลการทดลองในกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย) โดยพิจารณาในระดับฟังก์ชัน

เมื่อพิจารณาการใช้เวลาในระดับฟังก์ชันของส่วน การจัดการเครือข่าย, การติดต่อเซิร์ฟเวอร์และส่วนการจัดเก็บข้อมูลในกรณีนี้เทียบกับในกรณีการทดลองผ่านเครือข่ายแล้ว พบว่าฟังก์ชันที่มีการใช้เวลามากที่สุด มีลักษณะเป็นไปในทางเดียวกัน จะแตกต่างกันเฉพาะส่วนของการจัดการเครือข่ายเท่านั้นเอง ซึ่งเปรียบเทียบให้เห็นได้ดังกราฟต่อไปนี้

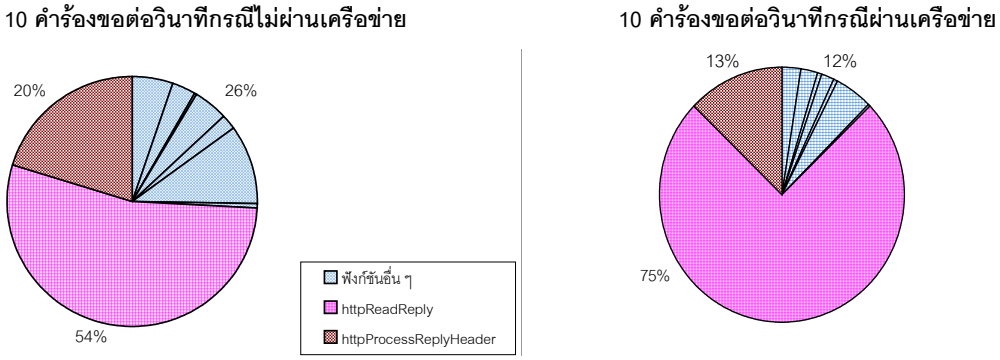


รูปที่ 5.10 กราฟแสดงเปรียบเทียบเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดการเครือข่ายที่อัตรา 10 คำร้องขอต่อวินาทีในการทดลองกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) และกรณีใช้คอมพิวเตอร์เครื่องเดียวในการจำลองการทำงาน (ไม่ผ่านเครือข่าย)



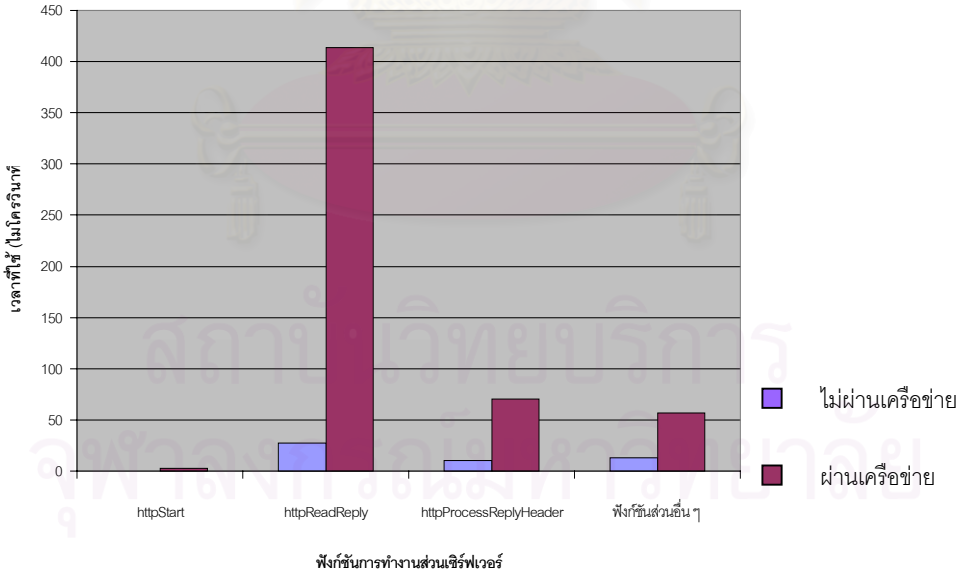
รูปที่ 5.11 กราฟแสดงการใช้เวลาของฟังก์ชันในส่วนการจัดการเครือข่ายของโปรแกรมสควิดในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ในการทดลองที่อัตราการร้องขอ10 คำร้องขอต่อวินาที กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) และกรณีใช้คอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย)

จากกราฟรูปที่ 5.10 แสดงให้เห็นว่าการทำงานของเครือข่ายมีผลต่อฟังก์ชันการทำงานในส่วนนี้ ดังจะเห็นได้จากเปอร์เซ็นต์การใช้เวลาที่เพิ่มขึ้นอย่างมาก เนื่องจากในกรณีการทดลองผ่านเครือข่าย จะต้องมีการการสร้างส่วนการติดต่อเกิดขึ้น การควบคุมการติดต่อต่าง ๆ ระหว่างเครื่อง ซึ่งในกรณีกรณีการทดลองไม่ผ่านเครือข่ายในฟังก์ชันเหล่านั้นใช้เวลาน้อย ระยะเวลาในการรอในส่วนของเครือข่ายจึงลดลง ส่งผลให้การใช้เวลาในส่วนนี้น้อย ดังจะเห็นได้จาก ฟังก์ชันในส่วนนี้ที่มีเปอร์เซ็นต์การใช้เวลามากที่สุด ได้แก่ comm_write ซึ่งเป็นฟังก์ชันที่เกี่ยวข้องกับการอ่าน/บันทึกข้อมูลส่วนที่ใช้ในการติดต่อ แต่ในขณะที่มีเครือข่ายฟังก์ชันที่มีการใช้มีการใช้เวลามากที่สุดได้แก่ commConnectStart เนื่องจากฟังก์ชันนี้จะต้องทำการติดต่อไปยังเซิร์ฟเวอร์อีกเครื่องหนึ่ง จึงต้องตรวจสอบชื่อของเซิร์ฟเวอร์ที่จะทำการติดต่อก่อน จากนั้นจึงกำหนดช่องทางการติดต่อไปยังเซิร์ฟเวอร์นั้น ๆ ซึ่งถ้าหากมีการติดต่อจากไคลเอนต์เข้ามาในเวลาเดียวกันเป็นจำนวนมาก และไม่พบข้อมูลที่ร้องขอ ก็จะต้องมีการส่งคำร้องขอต่อไปยังเซิร์ฟเวอร์ปลายทางมากขึ้น ทำให้ต้องมีการตรวจสอบชื่อของเซิร์ฟเวอร์ที่จะต้องส่งคำร้องขอเพิ่มมากขึ้นเช่นกัน และถึงแม้สควิดจะมีการจัดเก็บชื่อเซิร์ฟเวอร์เหล่านั้นไว้เพื่อใช้ตรวจสอบก็ตาม แต่ถ้าหากข้อมูลนั้นหมดเวลาในการจัดเก็บ ก็จะต้องมีการตรวจสอบใหม่อีกครั้ง ซึ่งก็ต้องใช้เวลาไปในส่วนนี้ด้วยเช่นกัน จึงทำให้ฟังก์ชันนี้ใช้เวลาค่อนข้างมากกว่าฟังก์ชันอื่น ๆ เมื่ออัตราการร้องขอเพิ่มมากขึ้น และในขณะเดียวกันฟังก์ชัน comm_write ก็มีการใช้เวลาเพิ่มขึ้นเช่นกัน พิจารณาได้ดังกราฟรูปที่ 5.11 ซึ่งเป็นฟังก์ชันที่เกี่ยวข้องกับการติดต่อกันระหว่างเครื่อง และเมื่อมีการติดต่อกันระหว่างเครื่องเกิดขึ้น เวลาการรอคอยต่าง ๆ จึงเพิ่มขึ้น จึงส่งผลให้ ไม่ว่าจะเป็นส่วนการจัดเก็บข้อมูล ซึ่งจะต้องมีการตรวจสอบข้อมูล, การจัดเตรียมพื้นที่ในจัดเก็บข้อมูลเพิ่มขึ้น ส่วนเซิร์ฟเวอร์ ซึ่งจะต้องทำการติดต่อไปยังเครื่องเซิร์ฟเวอร์ปลายทาง คอยรับ-ส่งข้อมูล, ตรวจสอบข้อมูลที่ส่งมาจนกว่าจะยุติการติดต่อ หรือส่วนการจัดการเครือข่ายที่มีการใช้เวลาที่เพิ่มขึ้นค่อนข้างมาก ทั้งนี้เพราะจะต้องมีการสร้างส่วนการติดต่อ การควบคุมการติดต่อระหว่างกันเกิดขึ้นเพื่อให้เป็นไปด้วยความเรียบร้อยและจนกว่าจะยกเลิกการติดต่อระหว่างกัน ทั้งระหว่างไคลเอนต์กับสควิดและระหว่างสควิดกับเซิร์ฟเวอร์ปลายทางเอง และจากกราฟดังกล่าวช่วยสนับสนุนว่าส่วนการจัดการเครือข่ายมีผลต่อการทำงานของโปรแกรมสควิดมากที่สุด



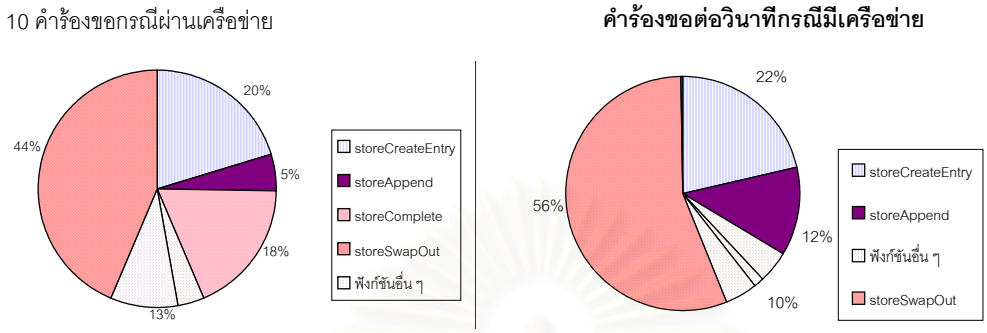
รูปที่ 5.12 กราฟแสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการติดต่อเซิร์ฟเวอร์ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีในการทดลองกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) และกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย)

จากกราฟรูปที่ 5.12 แสดงให้เห็นว่าฟังก์ชันที่ใช้เวลาในส่วนนี้ในกรณีไม่ผ่านเครือข่ายกับผ่านเครือข่ายโดยรวมแล้วก็ยังคงเป็นไปในลักษณะเดียวกัน ซึ่งฟังก์ชัน httpReadReply ก็ยังคงเป็นฟังก์ชันที่มีการใช้เวลามากที่สุด



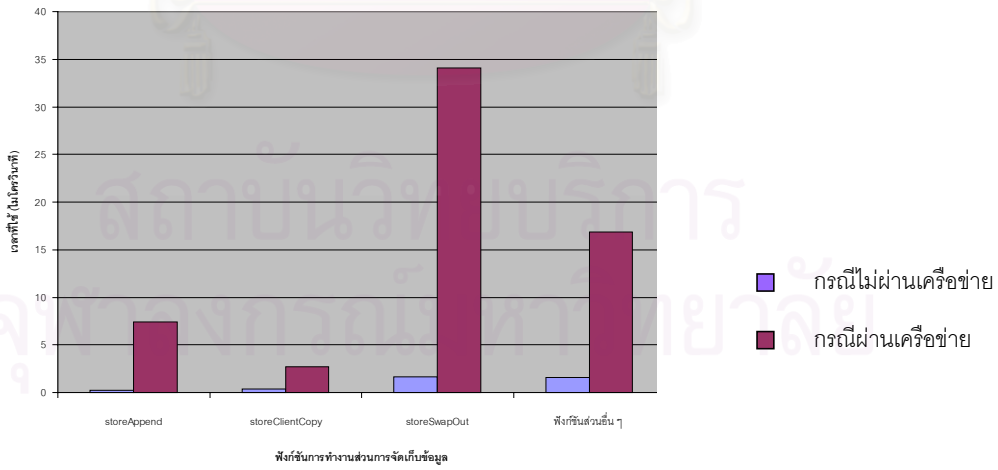
รูปที่ 5.13 กราฟเปรียบเทียบการใช้เวลาของฟังก์ชันในส่วนการติดต่อเซิร์ฟเวอร์ที่อัตราการร้องขอ 10 คำ ร้องขอต่อวินาทีในการทดลองกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) และกรณีใช้เครื่องคอมพิวเตอร์หนึ่งแครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย)

จากกราฟรูปที่ 5.13 แสดงให้เห็นว่าส่วนเครือข่ายมีผลต่อการทำงานอย่างมากซึ่งจะเห็นได้จากเวลาที่เพิ่มมากขึ้นในแต่ละฟังก์ชัน ถึงแม้จะมีอัตราการร้องขอในระดับเดียวกันก็ตาม



รูปที่ 5.14 กราฟแสดงเปอร์เซ็นต์การใช้เวลาของฟังก์ชันในส่วนการจัดเก็บข้อมูลที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีในการทดลองกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) และกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย)

จากกราฟรูปที่ 5.14 แสดงให้เห็นว่าฟังก์ชันที่มีการใช้เวลาในส่วนการจัดเก็บข้อมูลทั้งสองกรณีมีลักษณะการใช้เวลาที่เป็นไปในลักษณะเดียวกันโดยส่วนใหญ่ โดยที่ฟังก์ชัน storeSwapOut ยังคงเป็นฟังก์ชันที่มีการใช้เวลามากที่สุดเช่นกัน



รูปที่ 5.15 กราฟแสดงการใช้เวลาของฟังก์ชันในส่วนการจัดเก็บข้อมูลที่อัตราการร้องขอ 10 คำร้องขอต่อวินาทีในการทดลองกรณีใช้เครื่องคอมพิวเตอร์หนึ่งเครื่องในการจำลองการทำงาน (ไม่ผ่านเครือข่าย) และกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย)

จากกราฟรูปที่ 5.15 แสดงให้เห็นว่าส่วนของเครือข่ายนั้นมีผลต่อการทำงานเป็นอย่างยิ่ง ซึ่งจะเห็นได้จากเวลาในฟังก์ชันบางฟังก์ชันเพิ่มขึ้นถึงแม้จะมีอัตราการร้องขอที่ระดับเดียวกัน

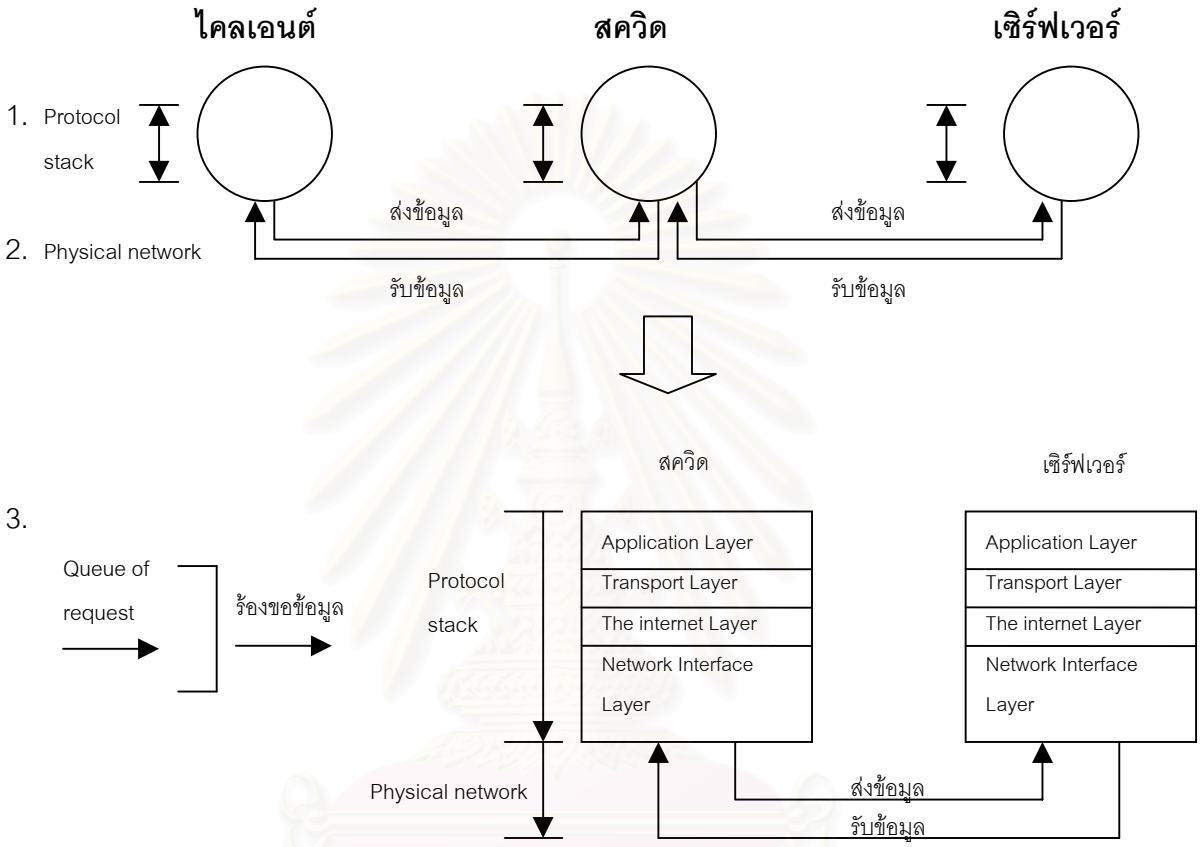
กล่าวโดยสรุปแล้ว เมื่อพิจารณาโดยภาพรวมในการทำงานของส่วนต่าง ๆ ของโปรแกรมสควิดในการให้บริการรับคำร้องขอข้อมูลจากไคลเอนต์ ส่วนที่มีการใช้เวลามากที่สุดเรียงตามลำดับได้ดังนี้

- ส่วนการจัดการเครือข่าย ซึ่งพิจารณาเปรียบเทียบได้กับผลการทดลองในภาพรวมกรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที จะเห็นได้ว่าส่วนนี้มีเปอร์เซ็นต์การใช้เวลาน้อยกว่าส่วนอื่น โดยคิดเป็นเปอร์เซ็นต์เพียง 1.19% ในขณะที่ในการทดลองที่อัตราการร้องขอเดียวกันในกรณีใช้เครื่องคอมพิวเตอร์เครื่องเดียวในการจำลองการทำงาน (ไม่ผ่านเครือข่าย) ส่วนนี้มีเปอร์เซ็นต์การใช้เวลาคิดเป็น 89% ซึ่งชี้ให้เห็นว่าส่วนการจัดการเครือข่ายมีผลต่อการทำของโปรแกรมสควิดมากที่สุด
- ส่วนการติดต่อเซิร์ฟเวอร์ ซึ่งพิจารณาเปรียบเทียบได้กับผลการทดลองในภาพรวมกรณีใช้เครื่องเดียวในการทดลอง (ไม่ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที ในส่วนนี้มีการใช้เวลาโดยเฉลี่ยเพิ่มขึ้นอย่างเห็นได้ชัด ซึ่งแสดงให้เห็นว่าการทำงานของเครือข่ายมีผลต่อการทำงานในส่วนนี้ด้วยเช่นกัน
- ส่วนการจัดเก็บข้อมูล ซึ่งพิจารณาเปรียบเทียบได้กับผลการทดลองในภาพรวมกรณีใช้เครื่องเดียวในการทดลอง (ไม่ผ่านเครือข่าย) ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที ในส่วนนี้ก็มีเปอร์เซ็นต์การใช้เวลาโดยเฉลี่ยเพิ่มมากขึ้นเช่นกัน
- ส่วนการติดต่อไคลเอนต์ ซึ่งส่วนสุดท้ายนี้เมื่อเทียบกับส่วนอื่น ๆ แล้วถือได้ว่าไม่มีผลต่อการทำงานของสควิดเท่าไหร่นัก โดยพิจารณาได้จากการทดลองที่อัตราการร้องขอเดียวกันในทั้งสองกรณี จะเห็นว่าเวลาที่ใช้ไม่แตกต่างกันมากนัก ทั้งนี้เนื่องจากในกระบวนการทำงานก็ยังคงลักษณะการทำงานเหมือนเดิม กรณีเลียนแบบการทดลองของเบคออฟ (ผ่านเครือข่าย) จึงไม่มีผลต่อการทำงานในส่วนนี้มากนัก แต่จะไปมีผลต่อการทำงานในส่วนอื่น ๆ มากกว่าที่ได้กล่าวมาแล้ว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิเคราะห์การใช้เวลาในส่วนโอเวอร์เฮดของเครือข่าย

จากผลการทดลองที่ได้ พบว่าเวลาที่ใช้ในแต่ละส่วนการทำงานของสควิดมีผลไม่มากนัก และเวลาที่เพิ่มขึ้นนั้นส่วนหนึ่งแล้วจะเป็นเวลาที่ต้องใช้ไปในส่วนของเครือข่ายหรือเป็นเวลาที่ใช้ไปกับโอเวอร์เฮดของเครือข่าย โดยพิจารณาการเกิดโอเวอร์เฮดของเครือข่าย ได้จากรูปที่ 5.16



รูปที่ 5.16 แสดงการเกิดโอเวอร์เฮดของเครือข่ายที่เกิดขึ้นจากการร้องขอข้อมูลจากไคลเอนต์ไปยังสควิดและจากสควิดไปยังเซิร์ฟเวอร์

จากรูปที่ 5.16 พิจารณาส่วนที่เป็นโอเวอร์เฮดของเครือข่าย ได้ดังนี้

1. เวลาที่ใช้ไปในส่วนโอเวอร์เฮดที่เกิดจากกระบวนการของโพรโทคอลที่ใช้ในการติดต่อสื่อสารระหว่างกัน (Protocol Stack) ก่อนที่จะมีการส่งข้อมูลออกไปยังอีกเครื่องหนึ่ง ซึ่งคิดจากการรับและส่งข้อมูล เพราะฉะนั้นเวลาดังกล่าวจะคำนวณได้ทั้งหมด 6 ครั้ง กล่าวคือ ผ่านโพรโทคอลของไคลเอนต์สองครั้ง ผ่านโพรโทคอลของสควิดสองครั้ง และผ่านโพรโทคอลของเซิร์ฟเวอร์สองครั้ง

2. เวลาที่ใช้ไปในส่วนของการส่งข้อมูลในระดับชั้นกายภาพของเครือข่าย (Physical Network) ซึ่งจะเกี่ยวกับระดับการทำงานของอุปกรณ์ต่าง ๆ ในการเชื่อมต่อ ซึ่งการวัดเวลาดังกล่าวได้จากการทดลองโดยทำการวัดเวลานับตั้งแต่เริ่มมีการติดต่อระหว่างกันจนกระทั่งไคลเอนต์ได้รับข้อมูลตอบกลับเรียบร้อยและมีการยกเลิกการติดต่อ (Propergation Delay) โดยเฉลี่ยต่อหนึ่งคำร้องขอ ที่อัตราการร้องขอ10 คำร้องขอต่อวินาที

3. เวลาที่ใช้ในส่วนการเข้าคิวเพื่อรอรับการประมวลผลการร้องขอของสควิด เวลาในส่วนนี้หมายถึงส่วนของเวลารอคอยการให้บริการรับคำร้องขอข้อมูลของสควิด สำหรับไคลเอนต์ที่มีการร้องขอเข้ามา สาเหตุที่ต้องมีการเข้าคิวดังกล่าวของไคลเอนต์ เนื่องจากสควิดจะทำงานในลักษณะซิงเกิลโพรเซส กล่าวคือ มีตัวที่คอยจัดการให้บริการเพียงตัวเดียว เพราะฉะนั้นในแต่ละไคลเอนต์ที่ส่งคำร้องขอเข้ามาจะต้องเข้าคิวเพื่อรอรับการให้บริการ ซึ่งเวลาในส่วนดังกล่าวพิจารณาจาก เวลาในส่วนที่เหลือจากเวลาในส่วนที่กล่าวในข้อหนึ่งและข้อสองรวมกัน โดยเทียบจากเวลาที่ได้จากผลการทดลองที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที ซึ่งการทำงานในส่วนนี้ควรจะปรับปรุงให้มีการทำงานในลักษณะเทรด (Thread) กล่าวคือ มีการทำงานในลักษณะกระจายโพรเซสในการให้การบริการรับคำร้องขอเป็นหลาย ๆ โพรเซส เพื่อให้สามารถให้บริการรับคำร้องขอได้ในคราวเดียวกัน ซึ่งได้มีการกล่าวถึงการพัฒนาให้สควิดสามารถทำงานได้ในลักษณะดังกล่าวในเว็บสควิด

โดยผลการทดลองจะเปรียบเทียบกับเวลาโอเวอร์เฮดของเครือข่ายที่คำนวณได้จากการทดลองที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที ซึ่งเวลาดังกล่าวที่วัดได้ประมาณ 1.29 วินาที โดยรายละเอียดของโอเวอร์เฮดดังกล่าวแสดงผลการทดลองได้ดังตารางที่ 5.12

ตารางที่ 5.12 แสดงเวลาที่ในส่วนของโอเวอร์เฮดของเครือข่าย

โอเวอร์เฮดของเครือข่าย	เวลาที่ใช้ (วินาที)
1.Protocol stack	0.8
2.Physical network	0.23
3.Queue of request	0.26 วินาที

ดังนั้นจะเห็นได้ว่าในการติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์แล้วจะต้องมีการใช้เวลาไปกับส่วนโอเวอร์เฮดของเครือข่ายดังกล่าวส่วนหนึ่ง

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

จากงานวิทยานิพนธ์นี้ เมื่อพิจารณาดูลักษณะการใช้เวลาในแต่ละส่วนโดยรวมแล้ว การทำงานในมอดูลไม่มีผลมากนักต่อเวลาการตอบสนองที่ช้าลง จะเห็นได้จากการใช้เวลาของแต่ละส่วนเพิ่มขึ้นไม่มากนัก มีลักษณะการใช้เวลาที่ค่อนข้างใกล้เคียงกันระหว่างอัตราการร้องขอในระดับต่ำกับอัตราการร้องขอในระดับที่เพิ่มมากขึ้น และพบว่าเวลาที่ใช้ส่วนหนึ่งนั้นจะใช้ไปกับส่วนที่เป็นโอเวอร์เฮดของเครือข่าย ซึ่งได้แก่โอเวอร์เฮดในระดับชั้นกายภาพของเครือข่าย โอเวอร์เฮดในส่วนของการทำงานของโพรโทคอล นอกจากนี้สภาพการทำงานของเครือข่ายก็มีผลต่อการทำงานของสควิดเช่นกัน ดังจะเห็นได้จากการทดลองการทำงานของเครือข่ายที่ใช้อุปกรณ์ในการเชื่อมต่อสำหรับการส่งข้อมูลได้ในอัตรา 10 เมกะบิตต่อวินาที พบว่าสควิดให้บริการรับคำร้องขอได้น้อย และเครือข่ายที่ใช้อุปกรณ์ในการเชื่อมต่อสำหรับการส่งข้อมูลได้ในอัตรา 100 เมกะบิตต่อวินาที พบว่าอัตราในการให้บริการคำร้องขอของสควิดเพิ่มขึ้น

ถ้าพิจารณาการใช้เวลาของส่วนการทำงานในแต่ละส่วนโดยไม่รวมเวลาในการรอคอยเครือข่ายแล้ว สามารถสรุปได้ว่าเมื่อมีจำนวนการร้องขอเพิ่มมากขึ้นส่วนที่ใช้เวลาในการทำงานมากที่สุด ได้แก่

- ส่วนการจัดการเครือข่าย โดยเฉพาะฟังก์ชันส่วนที่ทำการติดต่อไปยังเซิร์ฟเวอร์ปลายทาง เนื่องจากจะต้องมีการตรวจสอบชื่อเซิร์ฟเวอร์ปลายทางที่จะส่งไปติดต่อก่อน ถึงแม้จะมีการจัดเก็บข้อมูลเหล่านั้นไว้ใช้ก็จะมีการกำหนดระยะเวลาในการใช้ ซึ่งถ้าข้อมูลที่จัดเก็บไว้หมดเวลาก็ต้องทำการตรวจสอบใหม่ ทำให้ต้องใช้เวลาในส่วนนี้เพิ่มขึ้น เป็นผลให้ฟังก์ชันส่วนดังกล่าวใช้เวลามากกว่าฟังก์ชันอื่น ๆ
- ส่วนเซิร์ฟเวอร์ โดยเฉพาะฟังก์ชันส่วนการรอรับข้อมูลการตอบกลับจากเซิร์ฟเวอร์ปลายทาง เนื่องจากจะต้องมีการตรวจสอบข้อมูลส่วนเฮดเดอร์ที่ส่งตอบกลับมา ต้องทำการตรวจสอบข้อมูลว่าจัดเก็บได้หรือไม่ และทำการรอรับ-ส่งข้อมูลที่ระบุว่าจะจัดส่งมาจนเสร็จเรียบร้อย ซึ่งทำให้ฟังก์ชันส่วนดังกล่าวใช้เวลามากกว่าฟังก์ชันอื่น ๆ
- ส่วนการจัดเก็บข้อมูล โดยเฉพาะฟังก์ชันส่วนการจัดเก็บข้อมูลลงดิสก์ ทั้งนี้จะต้องมีการจัดเตรียมพื้นที่เพื่อใช้ในการจัดเก็บข้อมูล และจัดเตรียมข้อมูลบางส่วนเพื่อจัดให้อยู่ในรูปแบบที่จะจัดเก็บลงดิสก์ มีการอ่านข้อมูลจากหน่วยพักข้อมูลจัดเก็บลงดิสก์

และคอยตรวจสอบว่าการจัดเก็บว่าถูกต้องและเรียบร้อยดีหรือไม่ ทำให้ฟังก์ชันส่วนดังกล่าวใช้เวลามากกว่าฟังก์ชันอื่น ๆ

สำหรับส่วนไคลเอนต์ถือได้ว่ามีผลต่อการทำงานของสควิดน้อยมาก ซึ่งการทดลองในกรณีไม่ผ่านครือข่ายช่วยสนับสนุนเหตุผลดังกล่าว กล่าวคือ เมื่อพิจารณาเวลาที่ใช้ที่อัตราการร้องขอ 10 คำร้องขอต่อวินาที ในทั้งสองกรณี พบว่าเวลาที่ใช้ไม่แตกต่างกันมากนัก กล่าวคือ กรณีมีเครือข่ายโดยเฉลี่ยแล้วใช้เวลาประมาณ 63.21 ไมโครวินาที และในกรณีไม่มีเครือข่ายใช้เวลาโดยเฉลี่ยแล้วประมาณ 81.15 ไมโครวินาที ซึ่งเมื่อเทียบกับการทำงานส่วนอื่น ๆ แล้วจะมีการใช้เวลาที่เพิ่มขึ้นอย่างเห็นได้ชัด

6.2 ข้อเสนอแนะในการปรับแต่งสควิดในระดับมอดูลในการให้บริการการร้องขอข้อมูลจากไคลเอนต์

จากการทดลองเวลาที่ใช้ในแต่ละส่วนอาจมีผลไม่มากนักต่อการปรับแต่งการทำงานของสควิดให้เร็วขึ้น โดยเฉพาะในกรณีที่ถ้าหากสภาพการทำงานของระบบเครือข่ายมีปัญหา เกิดความล่าช้าอยู่ก่อนแล้ว การปรับปรุงการทำงานของสควิดในระดับมอดูลก็จะไม่ส่งผลให้สควิดทำงานได้ดีขึ้น ดังนั้นในการพิจารณาเพื่อปรับปรุงสมรรถภาพการทำงานของสควิดให้ดีขึ้น สิ่งที่ต้องพิจารณาเป็นอันดับแรกคือปรับปรุงในส่วนการทำงานของเครือข่าย และลำดับต่อมาควรจะพิจารณาเรื่องขนาดของแคช ซึ่งขนาดของแคชนั้นก็จะมีผลต่อการเพิ่มสมรรถภาพในการทำงานของสควิดได้เช่นกัน เพราะเมื่อขนาดของแคชเพิ่มขึ้นอัตราการพบข้อมูลก็จะเพิ่มขึ้น ทำให้สควิดรับคำร้องขอได้มากขึ้น แต่ทั้งนี้ก็จะช่วยให้ทำงานได้ดีในระดับหนึ่งเนื่องด้วยข้อจำกัดของอัตราการพบข้อมูลที่มีอยู่ นอกจากนี้ขนาดของแคชที่เพิ่มยังต้องพิจารณาให้เหมาะสมกับสภาพการใช้เว็บขององค์กรนั้น ๆ ดังจะเห็นได้จากงานวิจัยวิเคราะห์เปรียบเทียบสมรรถภาพของขั้นตอนวิธีการแทนที่ข้อมูลในพร็อกซีแคช [16] ซึ่งชี้ให้เห็นว่าขนาดของแคชควรจะเลือกให้เหมาะสมกับสภาพการใช้เว็บ ในขณะเดียวกันขนาดของแคชที่ใช้ก็ต้องเลือกขั้นตอนวิธีการแทนที่ให้เหมาะสมด้วย จึงจะทำให้สมรรถภาพการทำงานของแคชดีขึ้น เมื่อพิจารณาส่วนที่จะต้องทำการปรับปรุงร่วมกับการทำงานของสควิดแล้ว ในการปรับแต่งการทำงานของสควิดในระดับมอดูล ควรจะพิจารณาส่วนการทำงาน ดังนี้

- ส่วนการจัดการเครือข่าย เป็นลำดับแรก โดยพิจารณาฟังก์ชันที่ทำหน้าที่ในส่วนของการทำการติดต่อไปยังเซิร์ฟเวอร์ปลายทาง เพื่อร้องขอข้อมูลที่ผู้ใช้ร้องขอเข้ามา ซึ่งจะต้องมีการตรวจสอบชื่อเซิร์ฟเวอร์ที่จะต้องทำการติดต่อด้วย โดยสควิดจะมีการจัดเก็บข้อมูลส่วนนี้ไว้เพื่อใช้ในการตรวจสอบรายชื่อเซิร์ฟเวอร์ที่ไคลเอนต์ระบุข้อมูลที่ร้องขอเข้ามา และมีการกำหนดวันหมดอายุของข้อมูลที่จัดเก็บ จากการทำงานดังกล่าวจึงควรที่จะขยายวันหมดอายุออกไป เพื่อจะได้นำ

มาเป็นข้อมูลในการตรวจสอบได้นานขึ้น ทั้งนี้เนื่องจากชื่อของเซิร์ฟเวอร์มีการเปลี่ยนแปลงไม่บ่อยนัก

- ส่วนเซิร์ฟเวอร์ เป็นลำดับที่สอง โดยพิจารณาฟังก์ชันในส่วนที่มีการรับและอ่านข้อมูลที่จะตอบกลับมาจากเซิร์ฟเวอร์ปลายทาง ซึ่งจะต้องมีการตรวจสอบข้อมูลส่วนเฮดเดอร์ทุก ๆ ครั้งที่มีการตอบกลับมา อาจจะหาวิธีการลดการตรวจสอบส่วนเฮดเดอร์ที่ส่งกลับมาโดยไม่ต้องตรวจสอบทุก ๆ ครั้งถ้าเป็นเฮดเดอร์ที่ส่งมาจากที่เดียวกัน

- ส่วนการจัดเก็บข้อมูล เป็นลำดับที่สาม โดยพิจารณาฟังก์ชันส่วนที่มีการบันทึกข้อมูลลงดิสก์ โดยอาจให้มีการเพิ่มหน่วยความจำให้มีขนาดมากขึ้นทั้งหน่วยความจำหลักและหน่วยความจำรอง และเพิ่มหน่วยพักข้อมูลให้สามารถรองรับข้อมูลได้มากขึ้น ซึ่งจะช่วยลดการติดต่อกับดิสก์ ไม่ว่าจะเป็นการอ่านข้อมูลหรือบันทึกข้อมูลต่าง ๆ น้อยลง ทำให้เวลาที่จะต้องใช้ในส่วนนี้ลดลง และในขณะเดียวกันการปรับแต่งขนาดพื้นที่ของหน่วยความจำหลักและหน่วยความจำรองหรือแม้แต่หน่วยพักข้อมูลนั้นจะต้องพิจารณาให้เหมาะสมกับขนาดขององค์กรหรือหน่วยงานที่ใช้ตลอดจนการกำหนดค่าต่าง ๆ ของสควิดให้เหมาะสมกับทรัพยากรที่มีอยู่ ซึ่งก็จะช่วยปรับปรุงสมรรถภาพของพร็อกซีได้ในระดับหนึ่ง

6.3 ปัญหาและข้อจำกัดที่ได้พบจากการวิจัย

1. ในการจำลองการทำงานของสควิด เครื่องที่ใช้ในการรันโพลีไคลเอนต์ หยุดทำงานเนื่องจากพื้นที่ในการทำงานไม่เพียงพอที่ระดับคำร้องขอ
   - 200 คำร้องขอ/วินาที ทดสอบ โดยยังไม่ใช้พร็อกซี
   - 150 คำร้องขอ/วินาที ทดสอบกับพร็อกซี

   ทำให้ไม่สามารถจำลองการทำงานของพร็อกซีที่มีปริมาณการร้องขอเป็นจำนวนมากได้ ดังนั้นการทดลองการทำงานในครั้งนี้สามารถทำการจำลองการร้องขอได้สูงสุดไม่เกิน 150 คำร้องขอ/วินาที

2. ระดับคำร้องขอที่ได้ต่ำกว่าค่าที่ควรจะเป็น เนื่องจากข้อจำกัดที่เกิดจากอุปกรณ์ที่ใช้ในการจำลองการทำงานทำงานได้ไม่เต็มสมรรถภาพ

3. ในการทดลองโดยใช้ระบบปฏิบัติการลีนุกซ์ (Red Hat Linux) จะมีปัญหาในเรื่องของการสร้างไฟล์สำหรับการสร้างการติดต่อและรับการติดต่อระหว่างกันมีค่าจำกัด ถึงแม้จะมีวิธีแก้ปัญหาดังกล่าว แต่วิธีการแก้ไขค่อนข้างยุ่งยาก ต้องมีการแก้ไขในระดับเคอเนล ทำให้เกิดความผิดพลาดได้ง่าย

6.4 ข้อเสนอแนะ

มอดูลที่ได้ในส่วนการให้บริการการร้องขอข้อมูลผู้วิจัยพิจารณาเพียงมอดูลหลัก ๆ และพิจารณาในกรณีที่ทำงานในลักษณะเป็นแคชเดี่ยวเท่านั้น ซึ่งอาจจะศึกษาและทดลองเพิ่มเติมในส่วนที่ทำงานในลักษณะเป็นแคชหลายระดับ เพื่อเป็นประโยชน์ต่อการทำงานของแคชในหลาย ๆ ระดับต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Jia Wang, A Surway of Web Cacheing Schemes for the Internet,1999.

[2] Squid Internet Object Cache. [online] Available from: http://www.squid-cache.org/ [2000, June 20]

[3] DynaCache. [online] Available from: http://www.cacheflow.com/products/dynacache/overview.html. [2000, June 22]

[4] Cacheflow. [online] Available from: http://www.infolibria.com/products/index.cfm. [2000, June 25]

[5] Cisco's Cache Engine series. [online] Available from: http://www.cisco.com/warp/public/cc/pd/cxsr500.[2000, June 30]

[6] Alex Rousskov, Duane Wessls, and Glenn Chisholm, The First IRCache Web Cache Bake-Off. [online] Available from: http://cacheoff.ircache.net/N01/. [2000, July 3]

[7] Alex Rousskov, Performance Profiling Patch.1997. [online] Available from: http://www.cs.ndsu.nodak.edu/~rousskov/ research/cache/squid/profiling/patch.html.[2000, July 9]

[8] Alex Rousskov, Squid profiling statistics. [online] Available from: http://www.cs.ndsu.nodak.edu/~rousskov/research/ cache/squid/profiling/stats/.[2000, July 9]

[9] Alex Rousskov, A Performance Study of the Squid Proxy on HTTP/1.0,1999. [online] Available from: http://www.cs.ndsu.nodak.edu/rousskov99performance.html.[2000, July 3]

[10] Carlos Maltzahn, Kathy j. Richardson, and Dirk Grunwald, "Performance Issues of Enterprise Level Web Proxies", In Proc. Of the ACM SIGMETRICS Intl. Conference, Seattle, WA, June 1997. [online] Available from: http://citeseer.nj.nec.com/maltzahn97performance.html.[2000, August 2]

[11] Web Polygraph proxy performance benchmark. [online] Available from: http://polygraph.ircache.net.[2002, May 5]

[12] GNU gprof. [online] Available from: http://sources.redhat.com/binutils/docs- 2.10/gprof.html.[2002, Jaunary 9]

[13] Ari Luotonen, Henrik Frystyk Nielsen, and Tim Berners-Lee. Cern httpd 3.0A, July 1996. [online] Available from: http://www.w3.org/pub/WWW/Daemon. [2000, August 6]

[14] Carlos Maltazahn, Kathy j. Richardson, and Reducing, “Disk I/O of Web Proxy Server Caches”, in Proc. Of the USENIX Annual Technical Conference, Monterey, CA, June 6-11, 1999 . [online] Available from: http://citeseer.nj.nec.com/maltzahn99reducing.html.[2000, September 10]

[15] Douglas e. comer.Internetworking with TCP/IP.New Jersey: Prentice Hall, 1995.

[16] พรทวี วัฒนวิทูกูร. การวิเคราะห์เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการแทนที่ในพร็อกซีแคช. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2542.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

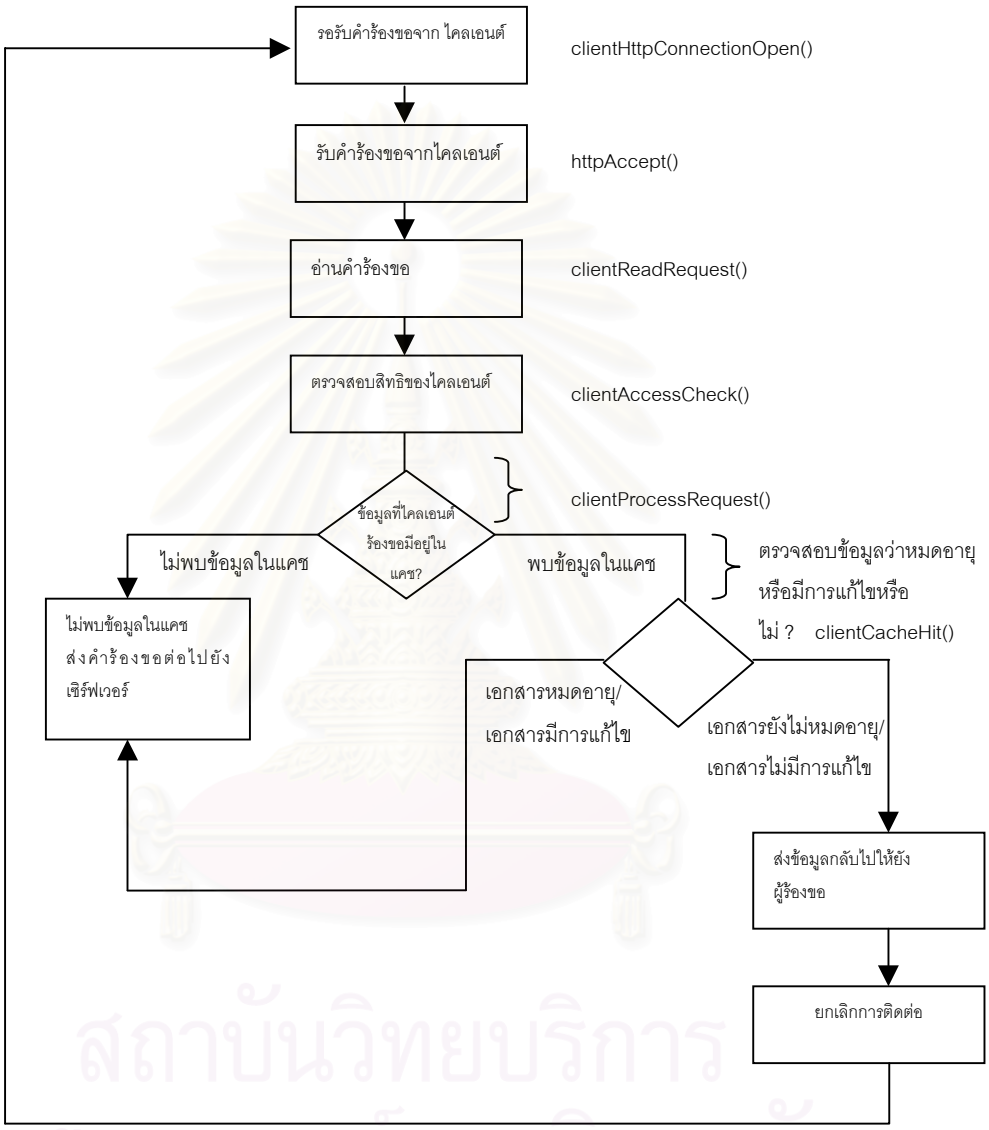
ภาคผนวก ก

การทำงานของสควิดส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์แต่ละส่วน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การทำงานของสควิดส่วนของการให้บริการการร้องขอข้อมูลจากไคลเอนต์แต่ละส่วน

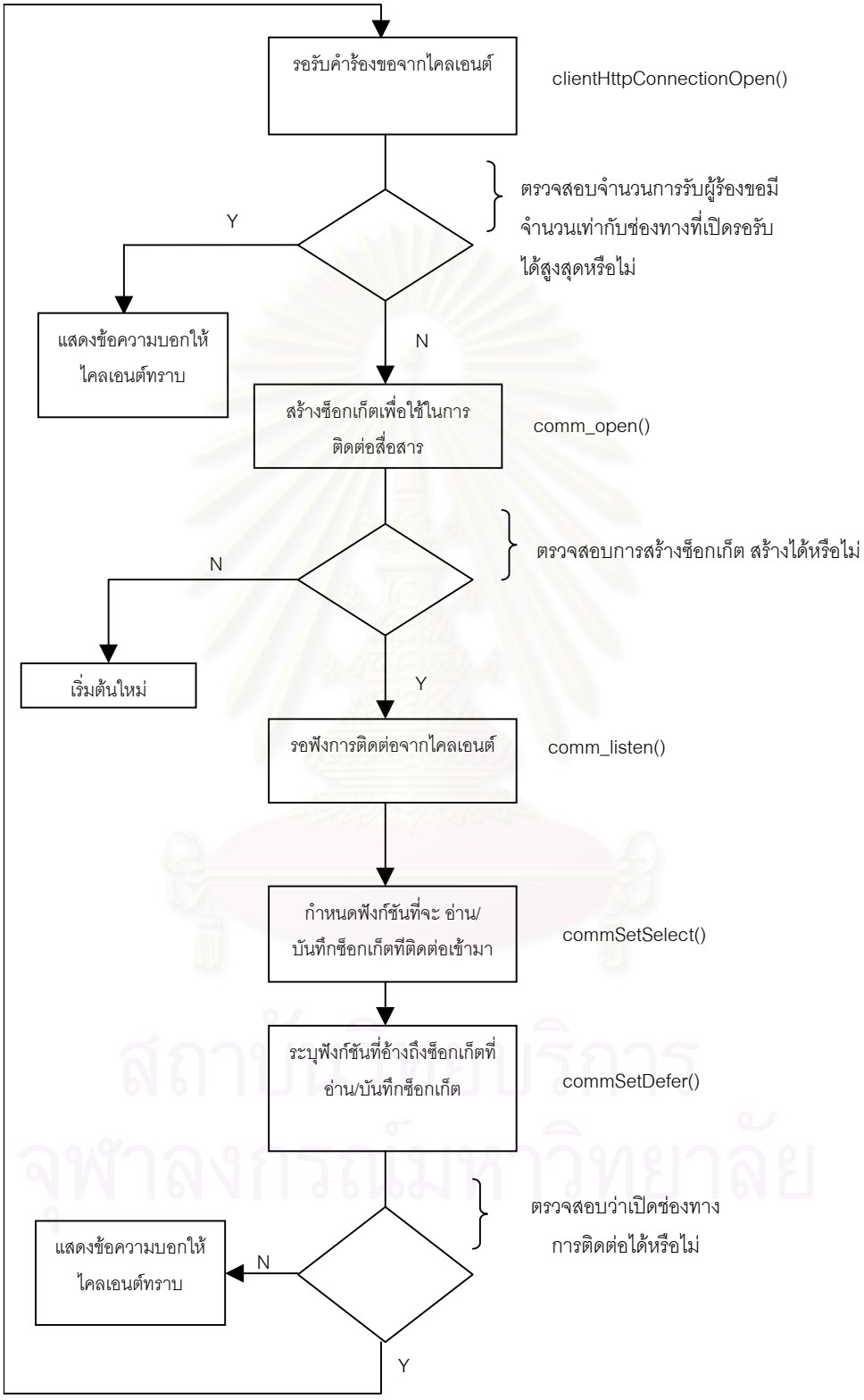
จากฟังก์ชันที่เกี่ยวข้องในแต่ละส่วนนำมาแสดงเป็นขั้นตอนการทำงานได้ดังโฟลว์ต่อไปนี้



รูปที่ ก1 แสดงขั้นตอนการทำงานส่วนไคลเอนต์ในการให้บริการการร้องขอข้อมูลจากไคลเอนต์ของโปรแกรมสควิด

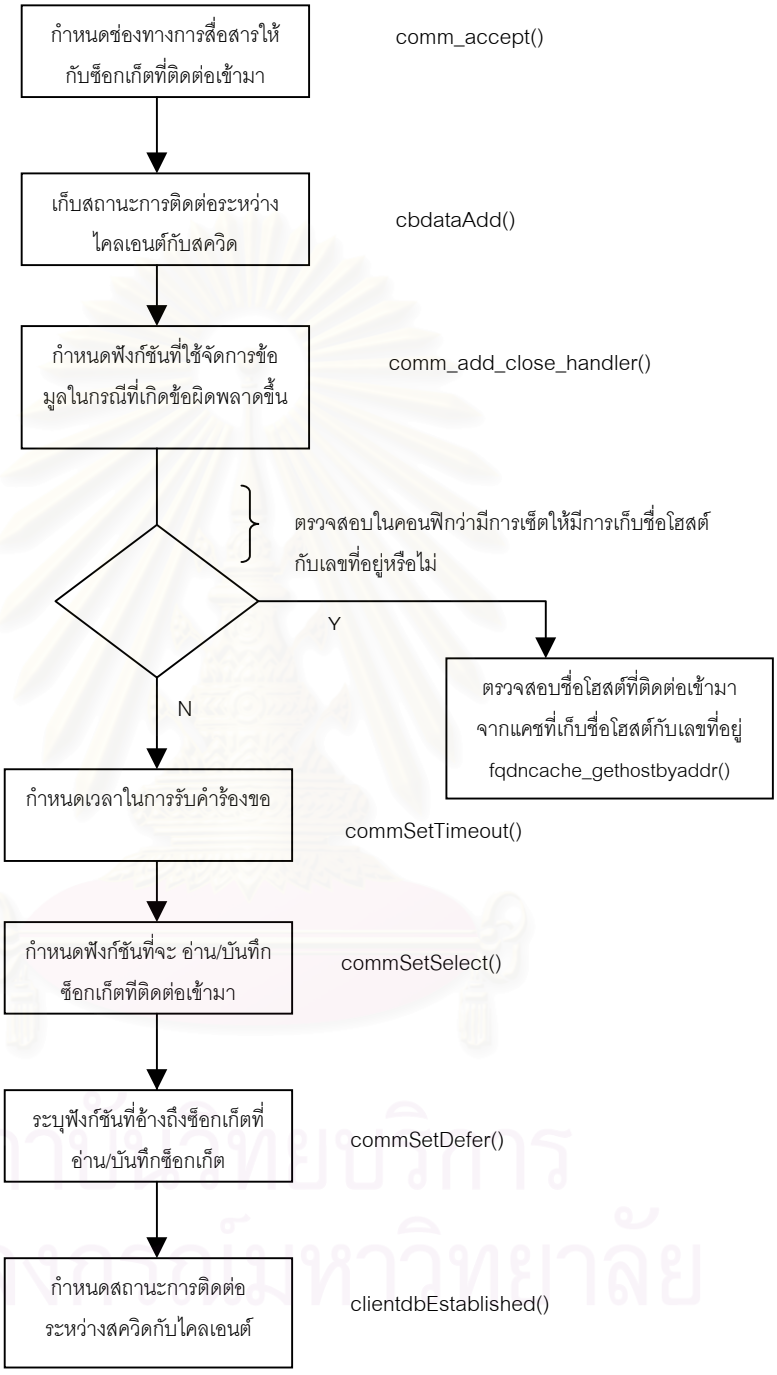
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนในการรอรับคำร้องขอและมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้



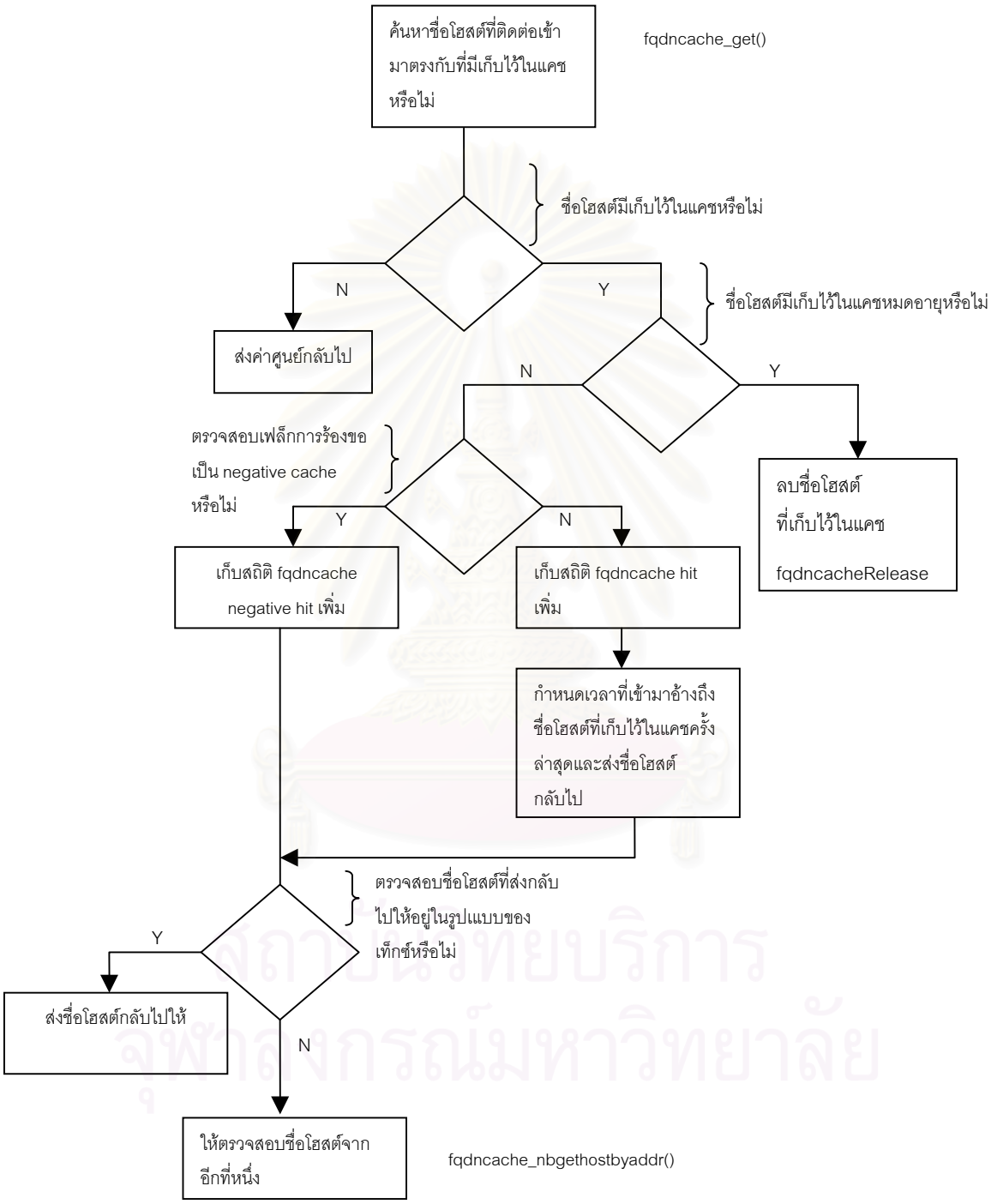
รูปที่ ก2 แสดงขั้นตอนในการรอรับคำร้องขอ

ขั้นตอนในการรับคำร้องขอและมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้



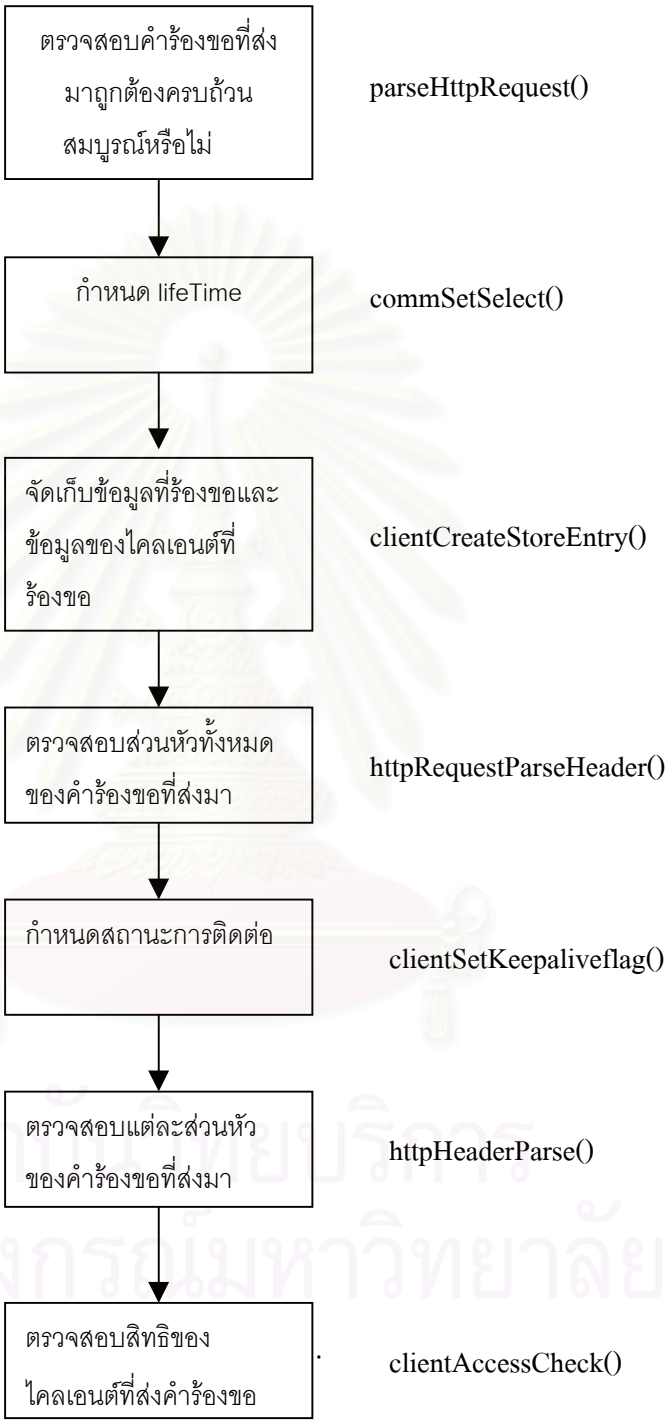
รูปที่ ก3 แสดงขั้นตอนในการรับคำร้องขอ

ขั้นตอนในการตรวจสอบชื่อโฮสต์โดยใช้เลขที่อยู่และมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้



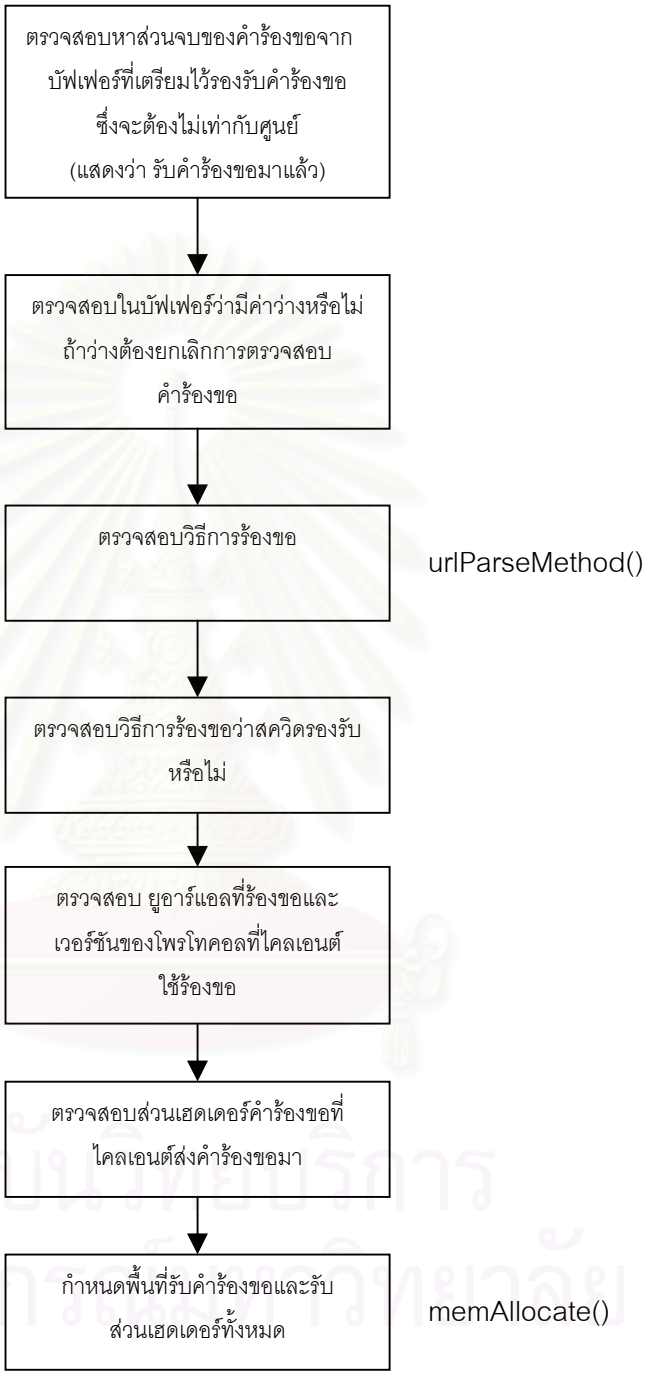
รูปที่ ก4 ขั้นตอนในการตรวจสอบชื่อโฮสต์โดยใช้เลขที่อยู่

ขั้นตอนในการอ่านคำร้องขอ clientReadRequest() และมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้



รูปที่ ก5 แสดงขั้นตอนในการอ่านคำร้องขอ

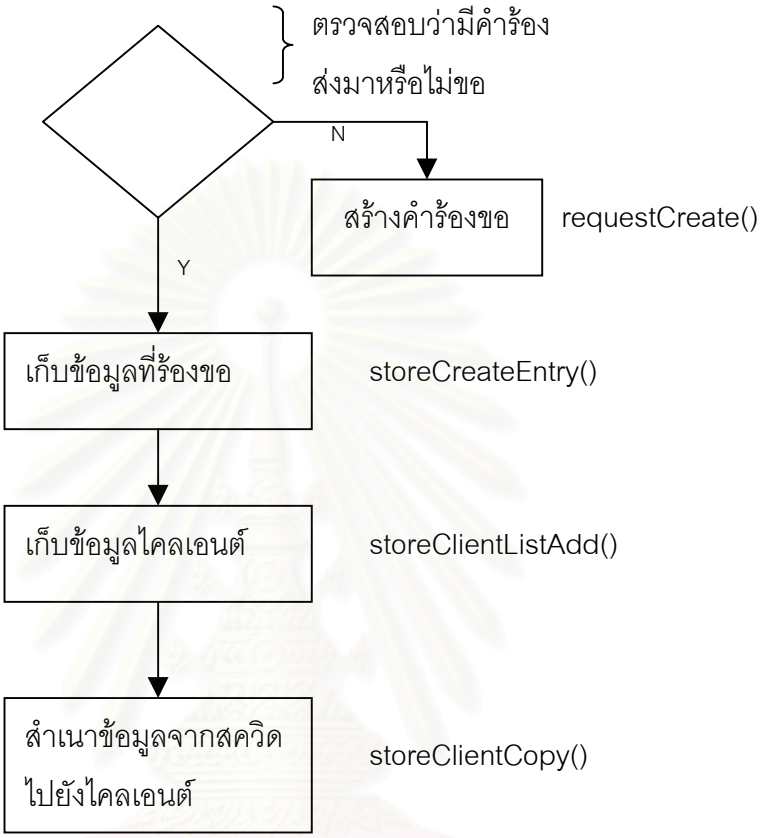
ขั้นตอนในการตรวจสอบคำร้องขอที่ส่งมาถูกต้องครบถ้วนสมบูรณ์หรือไม่ parseHttpRequest() **และมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้**



รูปที่ ก6 ขั้นตอนในการตรวจสอบคำร้องขอ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนในการจัดเก็บข้อมูลที่ร้องขอและข้อมูลของไคลเอนต์ที่ร้องขอ clientcreateStoreEntry() **และมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้**

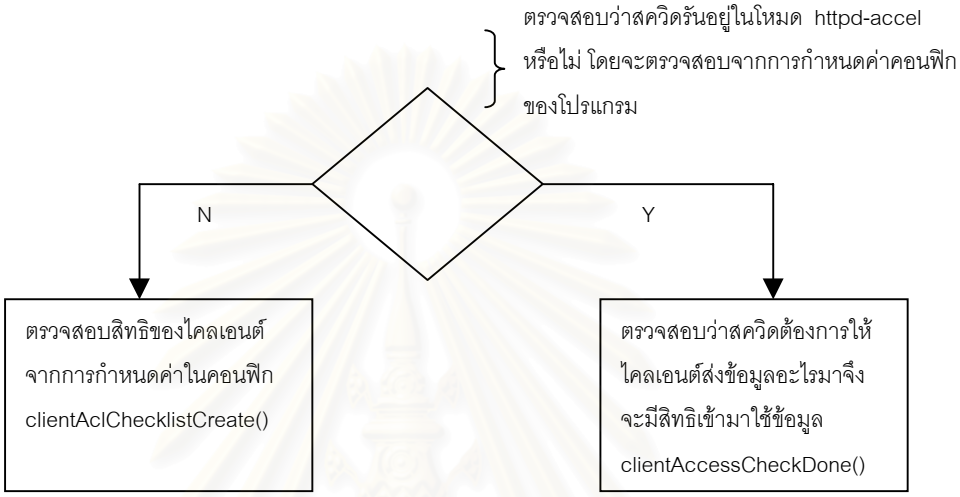


รูปที่ ก7 ขั้นตอนในการจัดเก็บข้อมูลที่ร้องขอ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนในการตรวจสอบสิทธิของไคลเอนต์ clientAccessCheck() **และมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้**

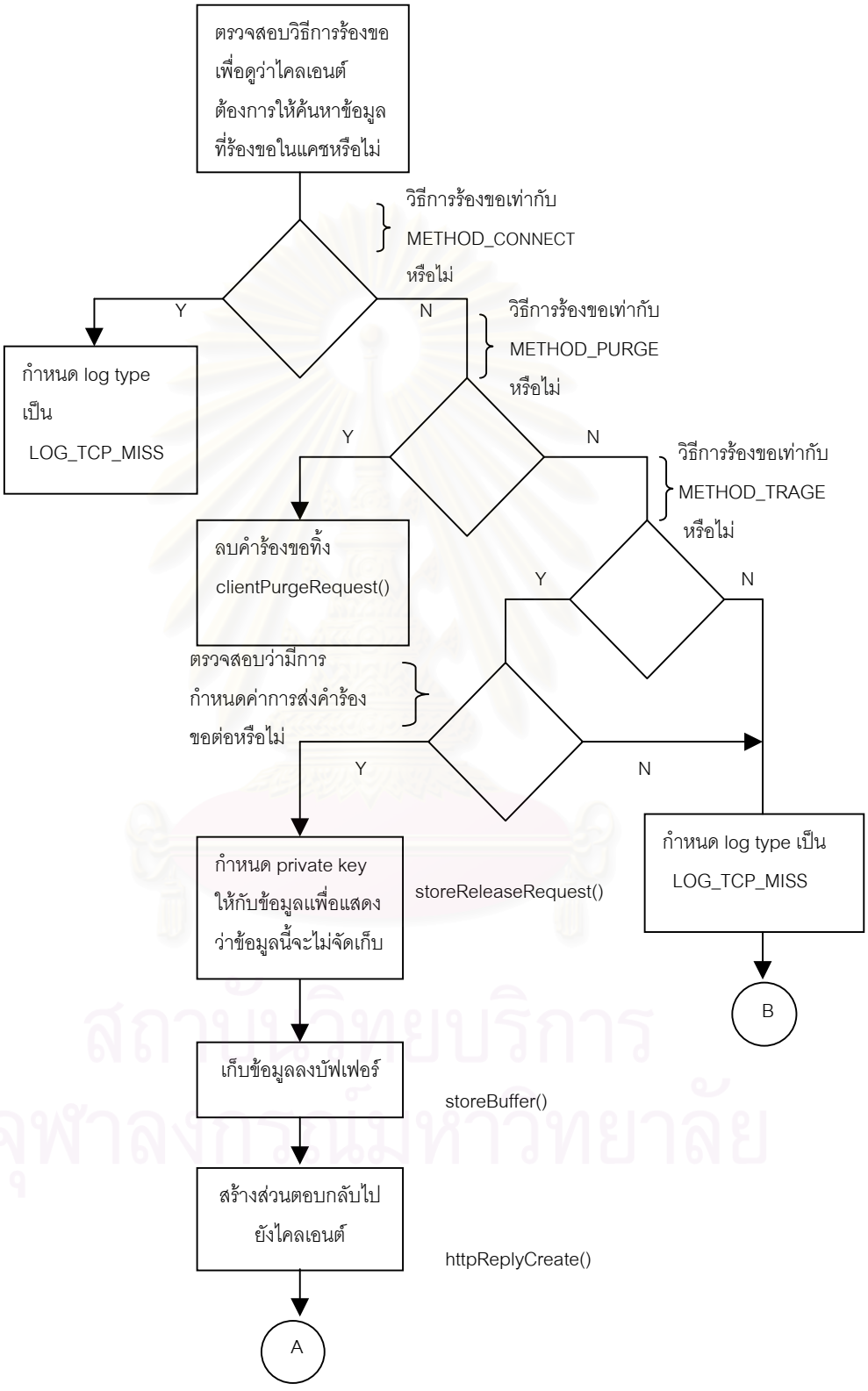
สควิดจะทำการตรวจสอบก่อนว่ารันในโหมด accel หรือไม่โดยจะตรวจสอบจากการกำหนดค่าในคอนฟิก ซึ่งถ้ามีไคลเอนต์บางคนส่งคำร้องขอมาในรูปแบบ proxy request ซึ่งจะถูกปฏิเสธทันทีสำหรับการรันในโหมดนี้



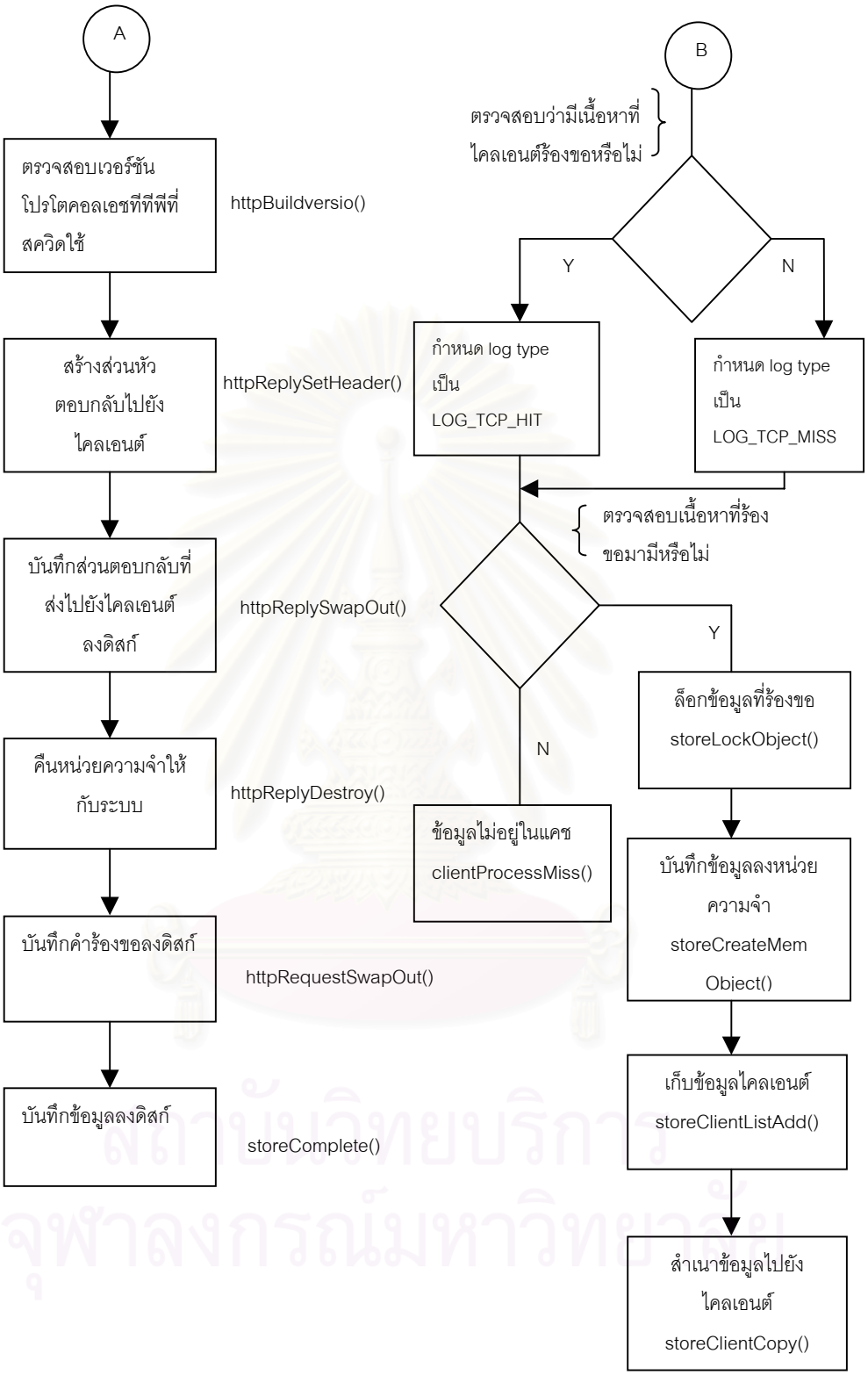
รูปที่ ก8 ขั้นตอนในการตรวจสอบสิทธิของไคลเอนต์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนในการประมวลผลคำร้องขอ clientProcessRequest() และมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้



รูปที่ ก9 ขั้นตอนในการประมวลผลคำร้องขอ

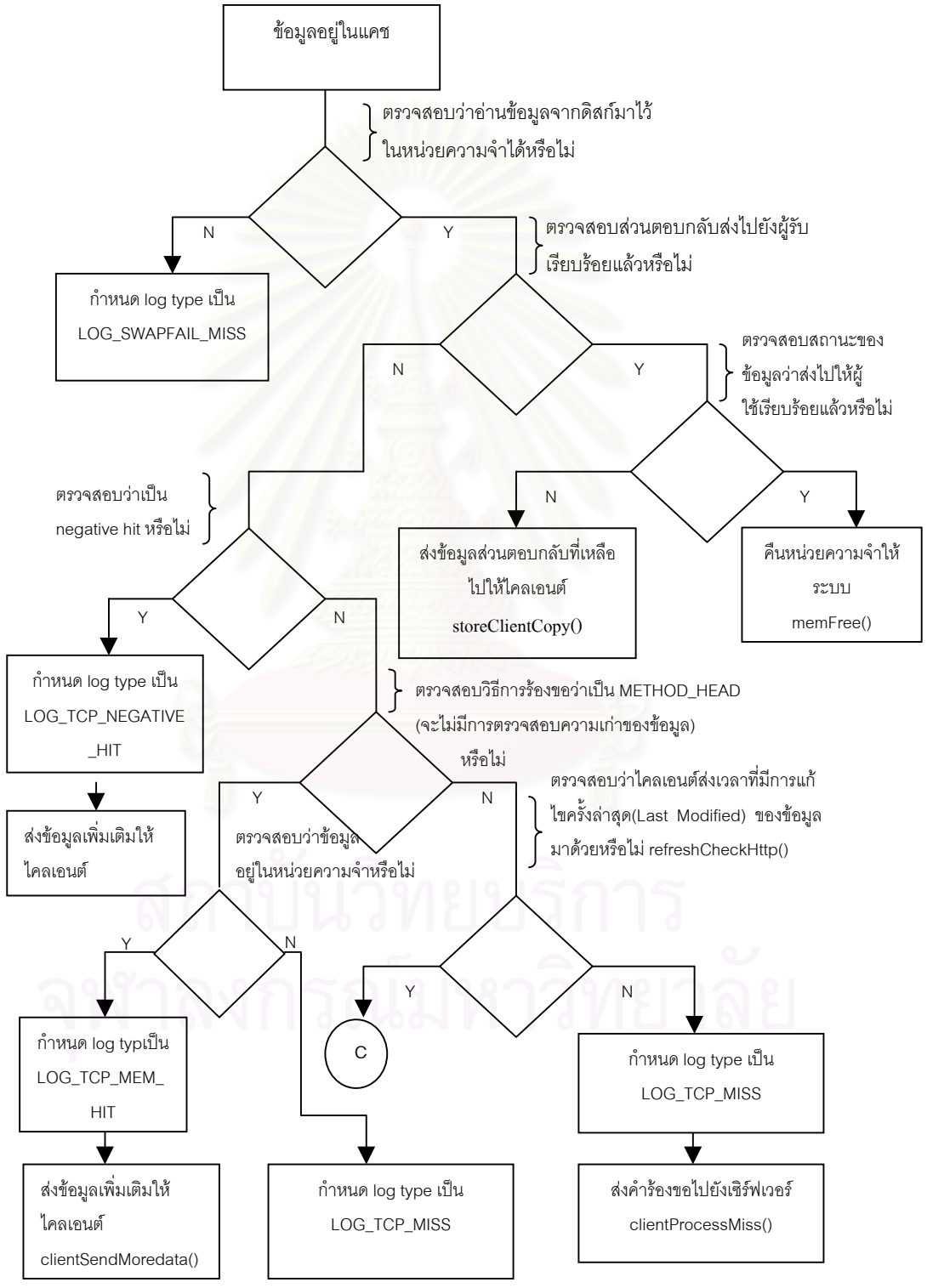


รูปที่ ก10 ขั้นตอนในการประมวลผลคำร้องขอ (ต่อ)

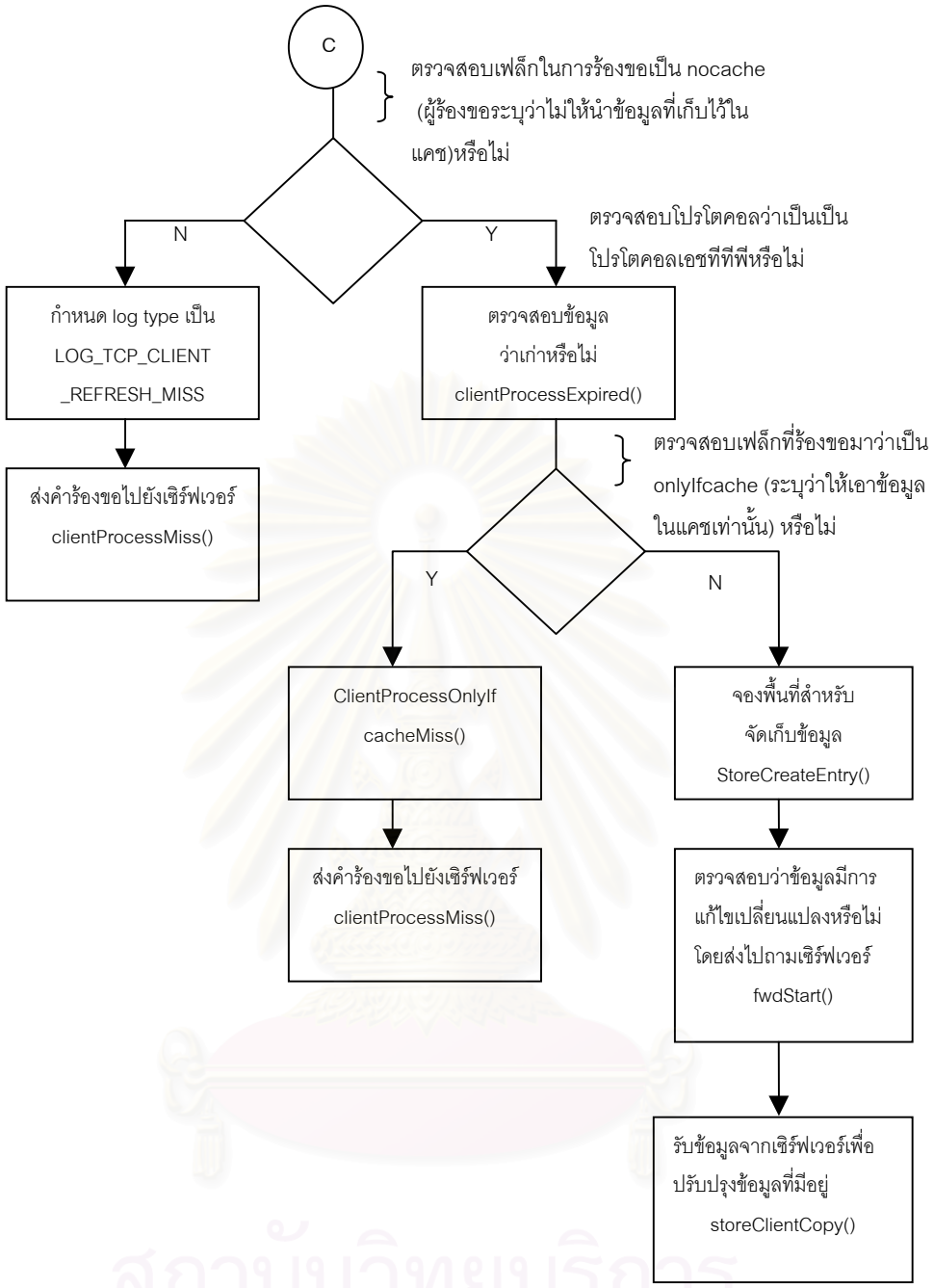
ขั้นตอนในการตรวจสอบข้อมูลว่าหมดอายุหรือมีการแก้ไขหรือไม่

clientCacheHit() **กรณีที่ข้อมูลที่ร้องขออยู่ในแคชพร้อมทั้งมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้**

จะดำเนินการเมื่อมีการส่งส่วนตอบกลับไปยังผู้ร้องขอเรียบร้อยแล้ว



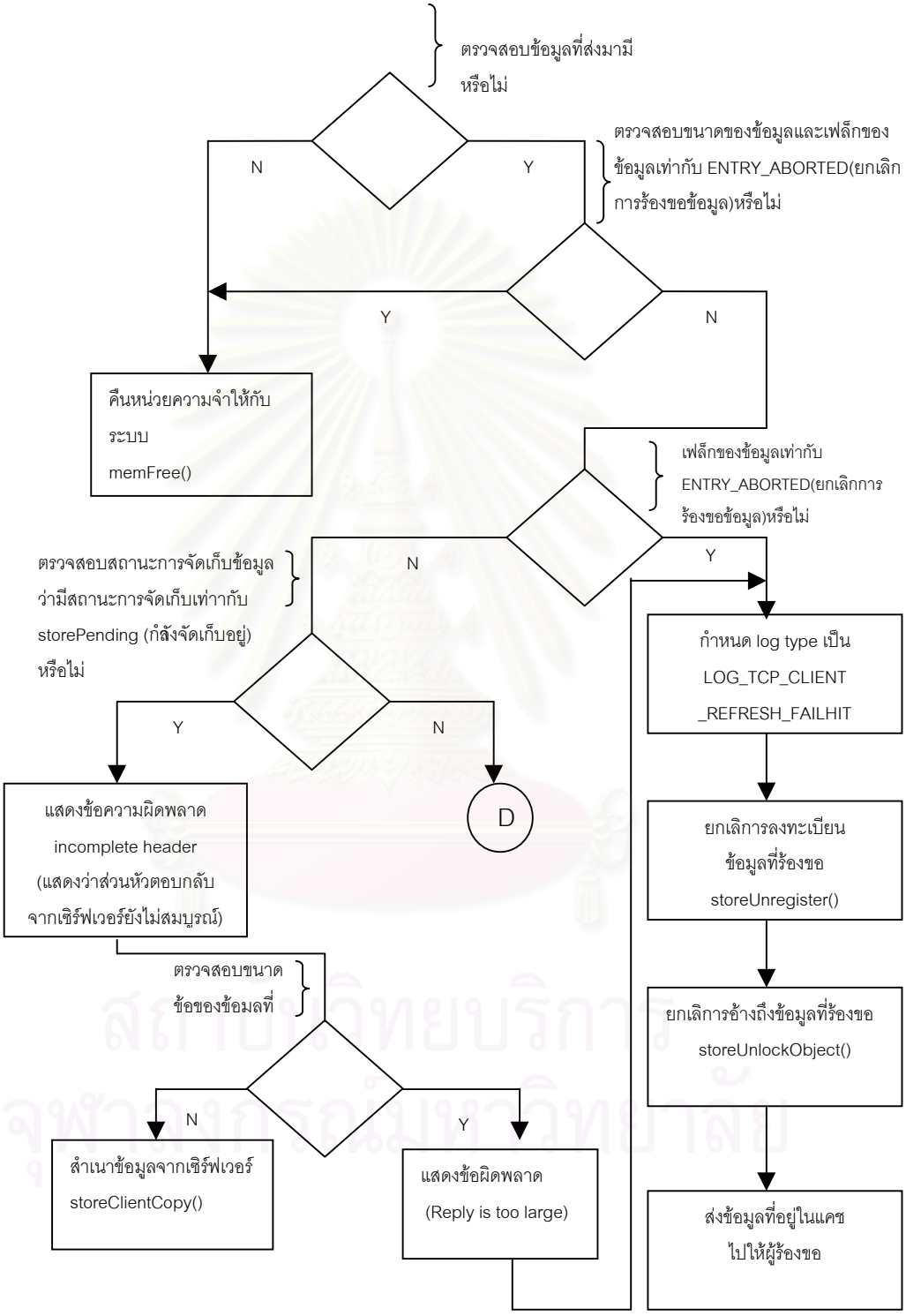
รูปที่ ก11 ขั้นตอนในการตรวจสอบข้อมูลว่าหมดอายุหรือมีการแก้ไขหรือไม่



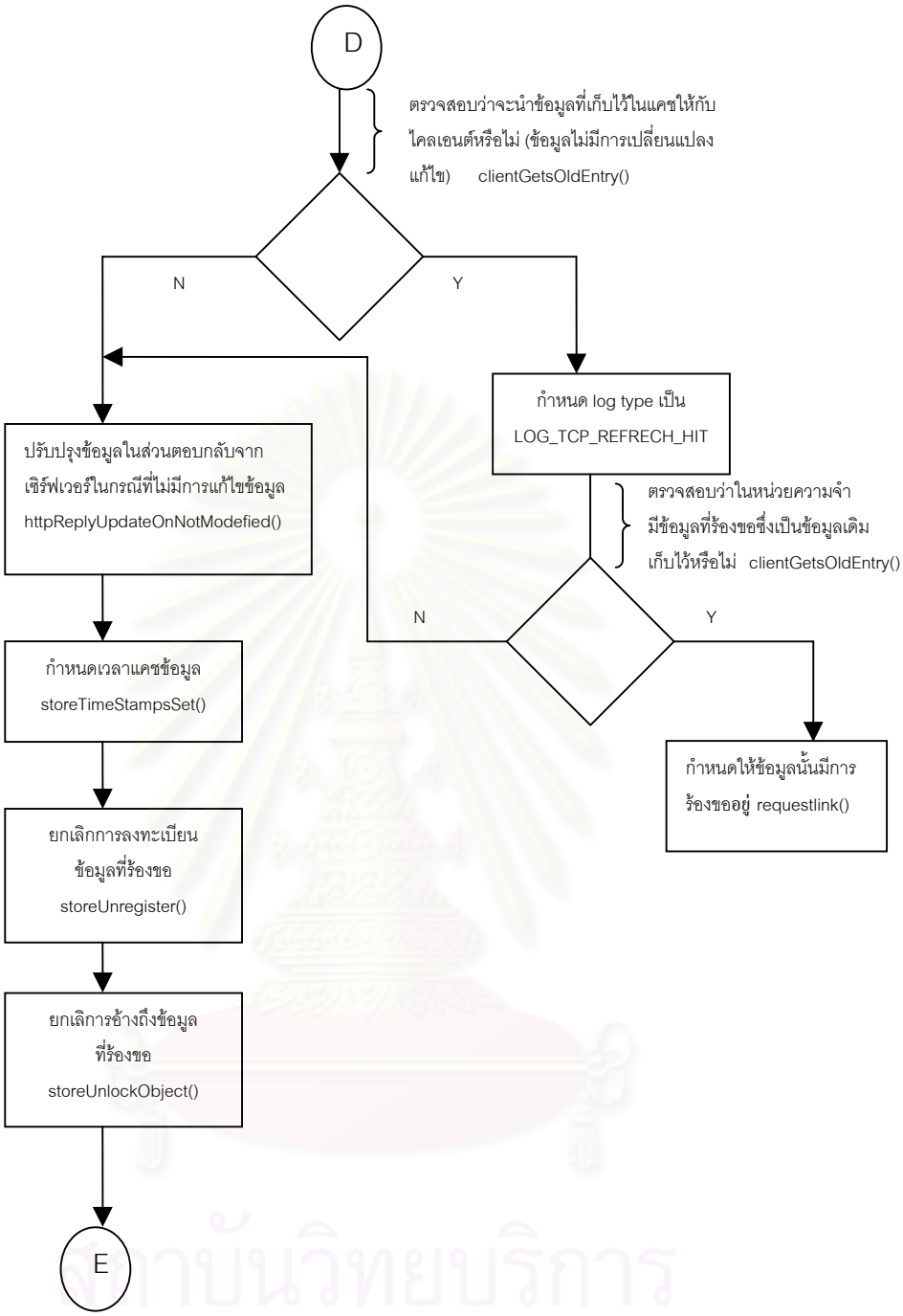
รูปที่ ก12 ขั้นตอนในการตรวจสอบข้อมูลว่าหมดอายุหรือมีการแก้ไขหรือไม่ (ต่อ)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

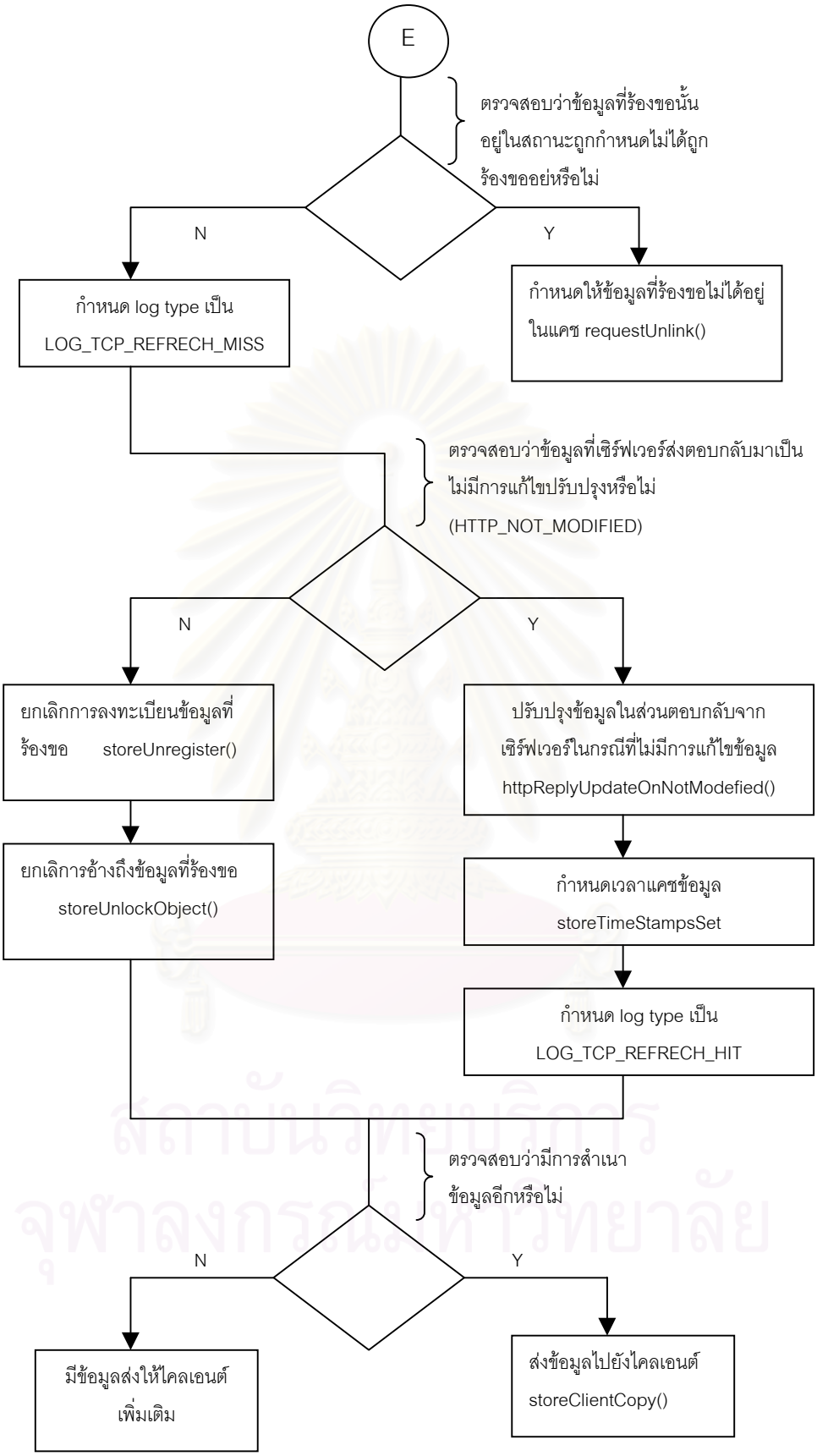
ขั้นตอนในการจัดการข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์ (clientHandleIMSReply()) **ในกรณีที่ส่งไปถามเซิร์ฟเวอร์ว่าข้อมูลมีการเปลี่ยนแปลงแก้ไข** (If modified since)**หรือไม่ พร้อมทั้งมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้**



รูปที่ ก13 ขั้นตอนในการจัดการข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์



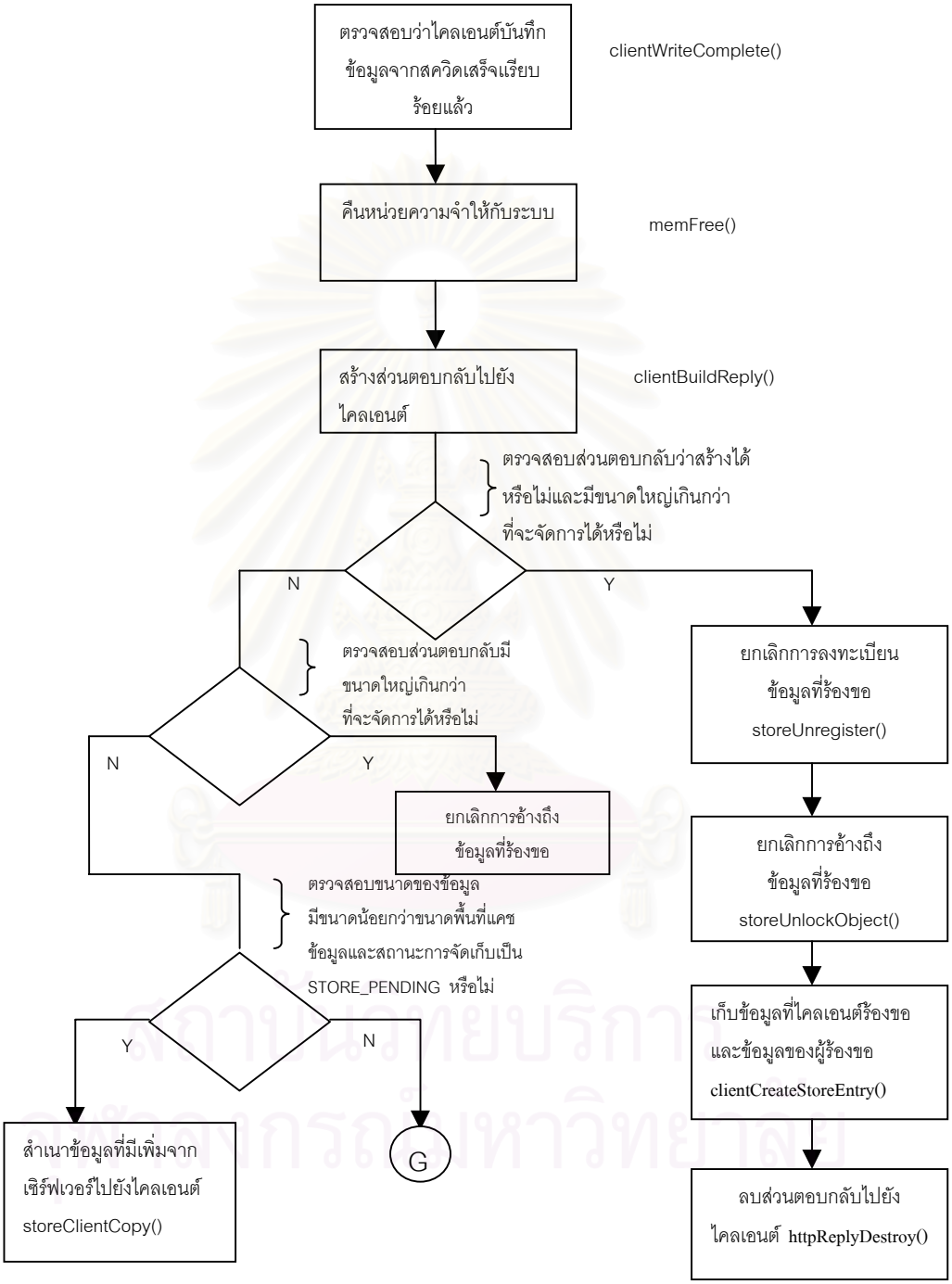
รูปที่ ก14 ขั้นตอนในการจัดการข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์ (ต่อ)



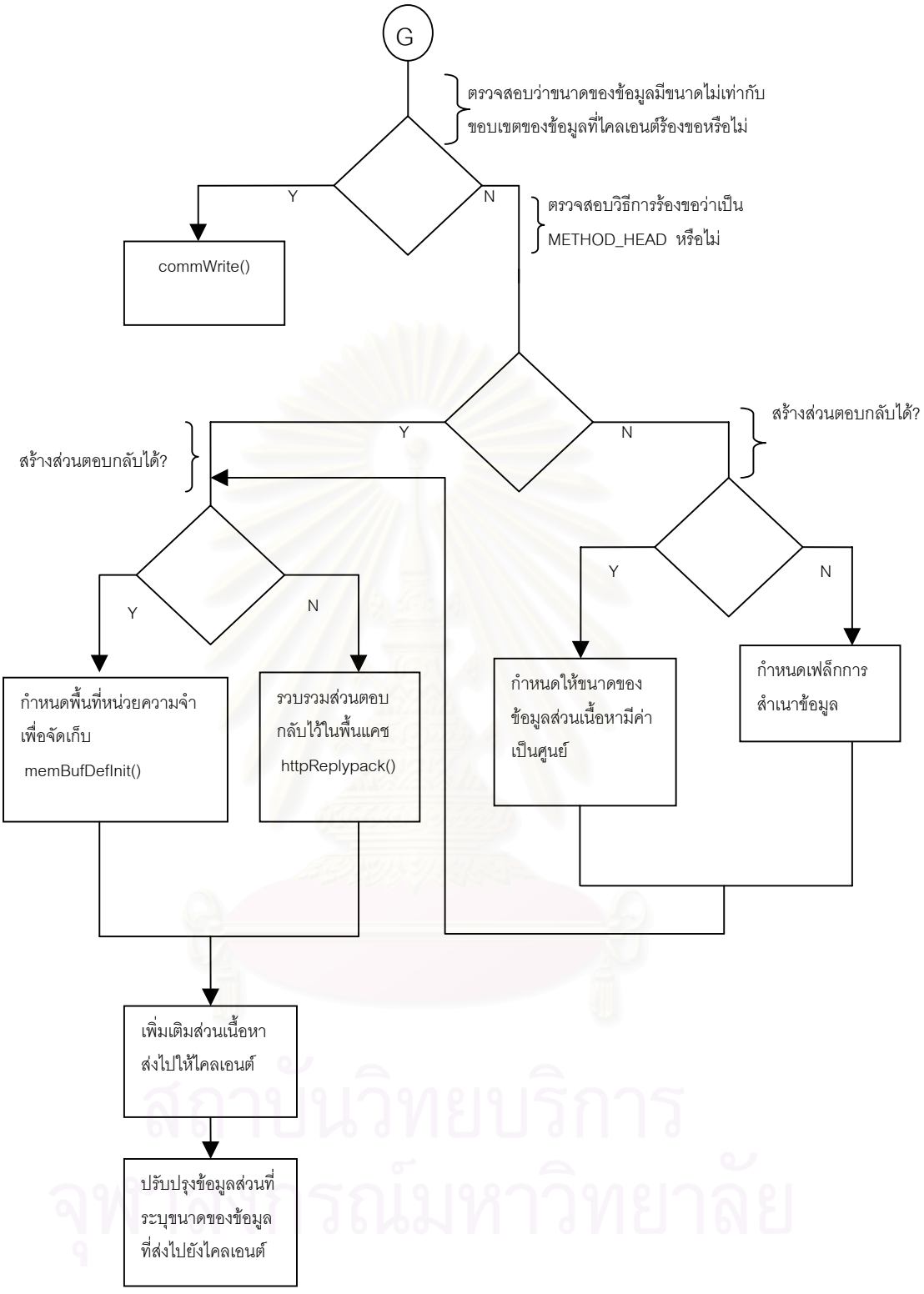
รูปที่ ก15 ขั้นตอนในการจัดการข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์ (ต่อ)

ขั้นตอนในกรณีที่มีข้อมูลที่จะจัดส่งให้กับไคลเอนต์เพิ่มเติม (clientSendMoreData ()) **พร้อมทั้งมอดูลที่เกี่ยวข้อง**

แสดงได้ดังโฟลว์ต่อไปนี้



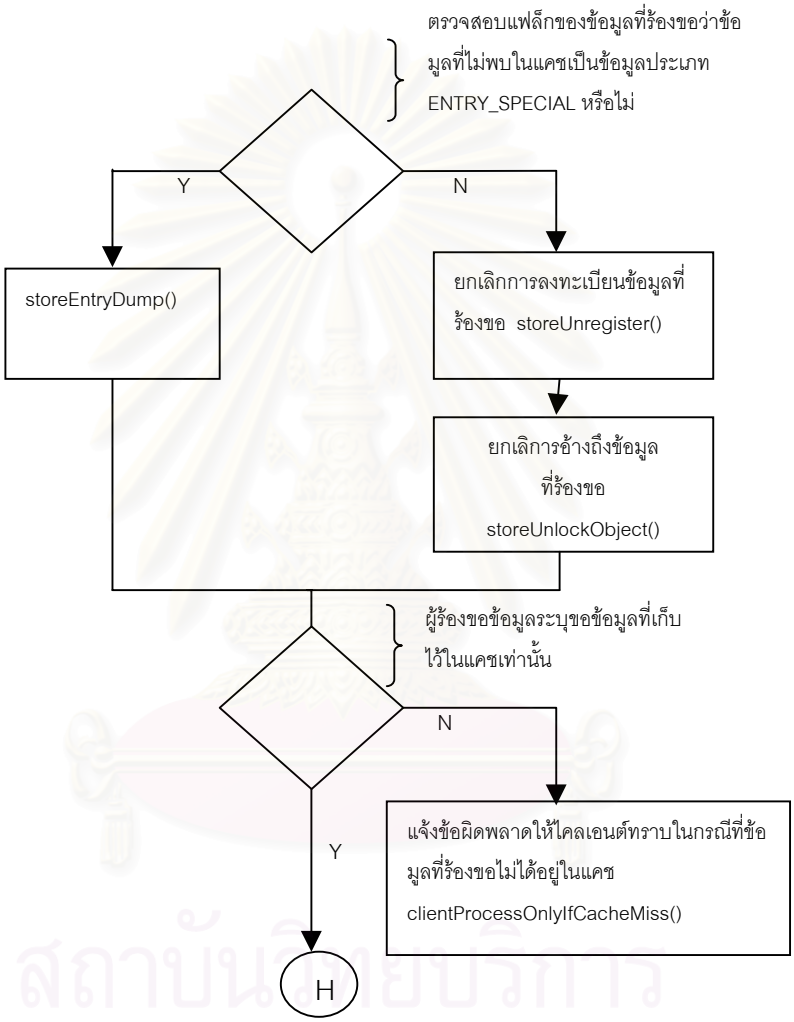
รูปที่ ก16 ขั้นตอนในกรณีที่มีข้อมูลที่จะจัดส่งให้กับไคลเอนต์เพิ่มเติม



รูปที่ ก17 ขั้นตอนในกรณีที่มีข้อมูลที่จะจัดส่งให้กับไคลเอนต์เพิ่มเติม (ต่อ)

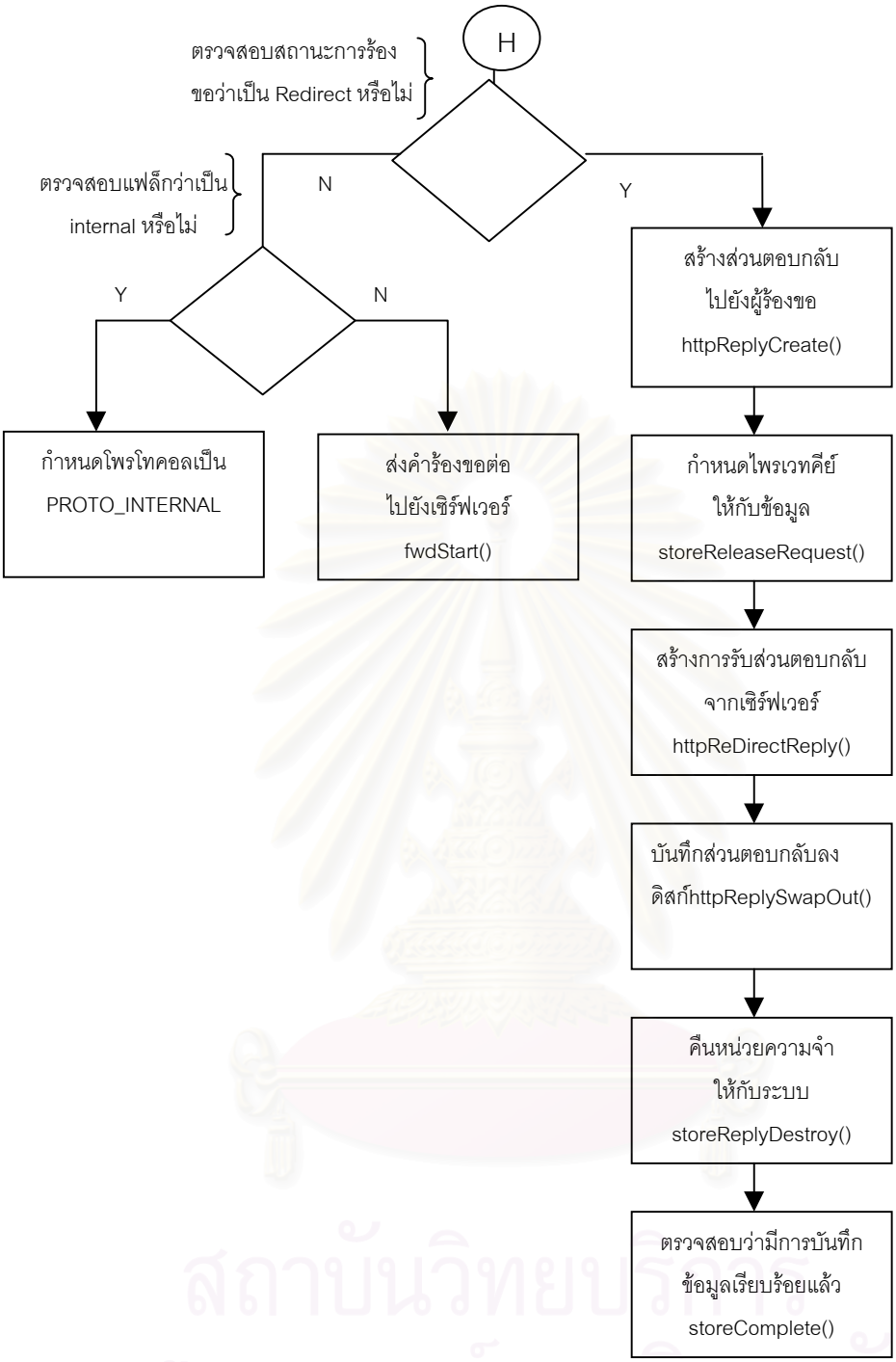
ขั้นตอนในการดำเนินการในกรณีที่ข้อมูลที่ร้องขอไม่พบในแคช clientProcessMiss() หรือข้อมูลหมดอายุแล้วพร้อมทั้งมอดูลที่เกี่ยวข้องแสดงได้ดังโฟลว์ต่อไปนี้

เพื่อเตรียมการไปนำข้อมูลที่ไคลเอนต์ร้องขอเข้ามาแล้วไม่พบในแคช โดยมีการตรวจสอบว่าจะไปติดต่อแหล่งข้อมูลได้จากเซิร์ฟเวอร์ใดบ้าง



รูปที่ ก18 ขั้นตอนในการดำเนินการในกรณีที่ข้อมูลที่ร้องขอไม่พบในแคช

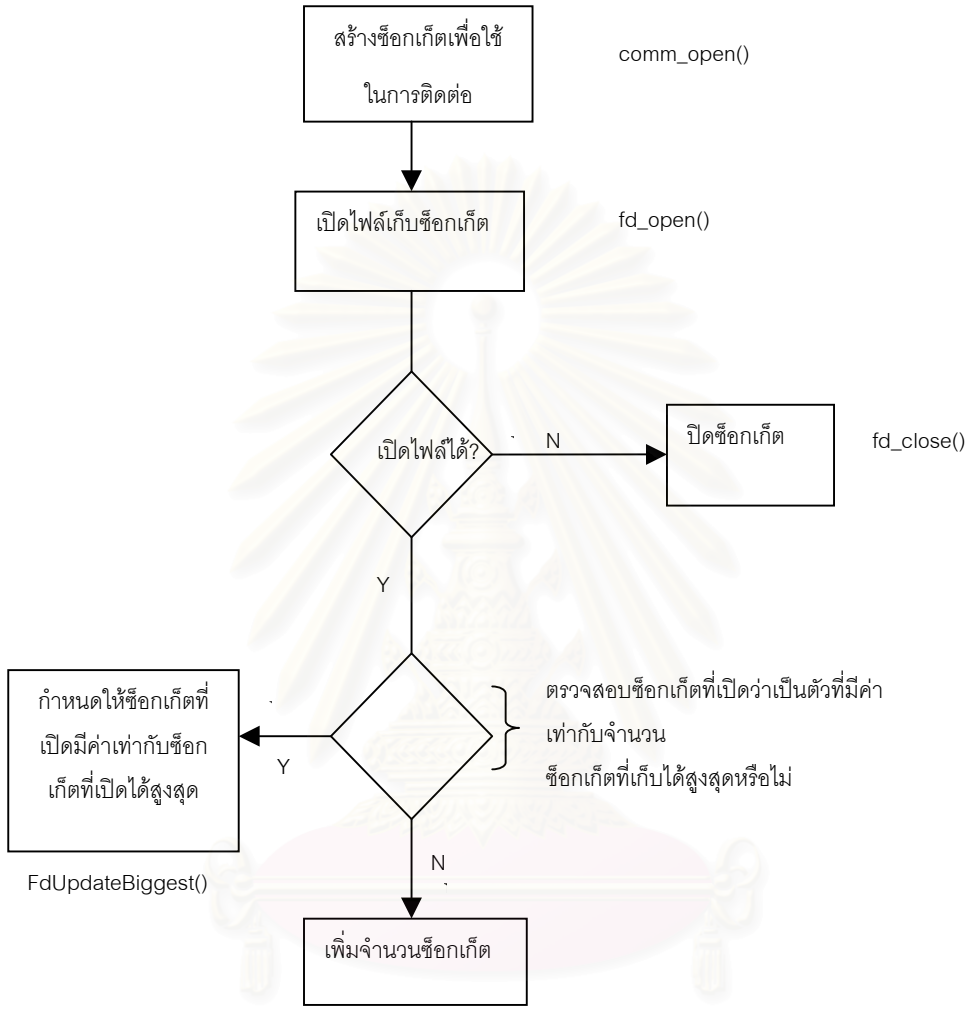
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก19 ขั้นตอนในการดำเนินการในกรณีที่ข้อมูลที่ร้องขอไม่พบในแคช (ต่อ)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แสดงขั้นตอนในการดำเนินการสร้างการติดต่อระหว่างสควิดกับไคลเอนต์ดังโฟลว์ต่อไปนี้



รูปที่ ก20 ขั้นตอนในการดำเนินการสร้างการติดต่อระหว่างสควิดกับไคลเอนต์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข
การทดลองของเบคออฟ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การทดลองของเบคออฟ

การทดลองที่ใช้จะเลียนแบบลักษณะของการใช้งานเว็บจริงที่เกิดขึ้น ได้แก่

- ข้อมูลที่ร้องขอ มีทั้งส่วนที่เป็นข้อความ รูปภาพ ภาพเคลื่อนไหว เสียง เป็นต้น
- มีการทดสอบการทำงานที่หลากหลายช่วง เช่น ช่วงเริ่มต้นการทำงานของแคช ช่วงเพิ่มการทำงาน ช่วงการทำงานสูงสุด ช่วงพักการทำงาน เป็นต้น
- มีการกำหนดช่วงอายุของข้อมูล เช่น วันหมดอายุ วันที่มีการแก้ไขข้อมูลครั้งล่าสุด เป็นต้น
- กำหนดการติดต่อโดยไม่ต้องทำการเปิดการติดต่อใหม่
- มีการสูญหายของข้อมูล
- มีส่วนการตอบกลับ
- มีการหน่วงเวลาในส่วนการติดต่อกับเซิร์ฟเวอร์ปลายทาง
- มีการกำหนดทั้งในแบบฮิตและมิส
- มีการกำหนดส่วนการตอบกลับในส่วนของข้อมูลที่ร้องขอ ทั้งแบบที่มีการระบุให้สามารถเก็บลงแคชได้กับแบบที่มีการระบุไม่ให้จัดเก็บลงแคช
- มีการกำหนดข้อมูลที่เป็นที่นิยม
- มีการกำหนดอัตราการร้องขอ
- มีการกำหนด Embedded อ็อบเจกต์และเลียนแบบพฤติกรรมการใช้เว็บ
- มีการจำลองอ็อบเจกต์ที่หลากหลายรูปแบบและไม่จำกัดจำนวนเพื่อใช้ในการทดลอง

รูปแบบปริมาณการใช้เว็บของโพลีกราฟ (Polygraph Traffic Model) ที่ใช้ในการทดลอง ประกอบด้วย

- เวลาที่ใช้ในการค้นหาเครื่องบริการชื่อโดเมน
- เนื้อหาของข้อมูลที่ร้องขอ มีทั้งข้อความ รูปภาพ เป็นต้น
- การยกเลิกการร้องขอ
- การตรวจสอบความถูกต้องทันสมัยของข้อมูล
- มีการบังคับให้แคชตรวจสอบความถูกต้องทันสมัยของข้อมูลด้วยวิธีการ Reload
- มีการหน่วงเวลาในส่วนไคลเอนต์
- มีการจำกัดแบนวิธด์
- ไม่มีการส่งปริมาณการร้องขอของเอชทีทีพี

ในการวัดสมรรถภาพการทำงานของพร็อกซีแคชของเบคออฟจะใช้โปรแกรมเว็บโพลีกราฟ ซึ่งเป็นโปรแกรมที่ใช้จำลองการทำงานการใช้เว็บระหว่างเซิร์ฟเวอร์กับไคลเอนต์ สำหรับตัวที่จะกำหนดคุณลักษณะการทำงานทั้งในส่วนไคลเอนต์และส่วนเซิร์ฟเวอร์ ได้แก่ เวิร์คโหลด

ตัวอย่างเวิร์คโหลดที่ใช้วัดสมรรถภาพพร็อกซีแคช

1. ชื่อเวิร์คโหลด : Polymix-1
   โพลีกราฟ เวอร์ชัน : 1.0p5 -1.0p7
   การกำหนดคุณลักษณะต่าง ๆ : ทั้งเซิร์ฟเวอร์และไคลเอนต์ ใช้การพิมพ์คำสั่ง (Command line) เช่น ต้องมีการกำหนดช่องทางในการรับการติดต่อของเซิร์ฟเวอร์ จะต้องระบุ ดังนี้ ./polysrv --ports 80 ซึ่งถ้าหากไม่ระบุ โปรแกรมก็จะไม่สามารถทำงานได้
   พารามิเตอร์ของเวิร์คโหลด : อัตราการร้องขอ (Request rate)
   วัตถุประสงค์ : พัฒนาขึ้นเพื่อใช้กับการทดลองวัดสมรรถภาพพร็อกซีแคชครั้งแรก (First cache –off)
   ข้อจำกัด : ไม่มีการกำหนดวันที่มีการแก้ไขข้อมูลครั้งล่าสุด (Last-Modified) และวันหมดอายุของข้อมูลที่ร้องขอ (Expires)
2. ชื่อเวิร์คโหลด : Polymix-2
   โพลีกราฟ เวอร์ชัน : 2.2.9 ขึ้นไป
   การกำหนดคุณลักษณะต่าง ๆ : ทั้งเซิร์ฟเวอร์และไคลเอนต์ ใช้การพิมพ์คำสั่ง เช่นกัน แต่มีการลดการพิมพ์คำสั่งบางส่วนลง เช่น ไม่ต้องกำหนดช่องทางในการรับการติดต่อของเซิร์ฟเวอร์ แต่มีการระบุบางส่วนเพิ่มเติม เช่น จะต้องระบุตำแหน่งของเวิร์คโหลดที่ใช้ เช่น ./polysrv --config /usr/local/workloads/polymix-2.pg ซึ่งถ้าหากไม่ระบุ โปรแกรมก็จะไม่สามารถทำงานได้
   พารามิเตอร์ของเวิร์คโหลด : อัตราการร้องขอสูงสุด (Peak request rate)
   สภาพแวดล้อมของพร็อกซี (Proxy environment)
   วัตถุประสงค์ : ใช้กับการทดลองวัดสมรรถภาพพร็อกซีแคชครั้งที่สอง (Second cache –off) ซึ่งพัฒนาขึ้นมาโดยปรับปรุงจาก Polymix-1 เพื่อลดข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นและเพิ่มคุณสมบัติใหม่ ๆ เพื่อให้ใกล้เคียงกับการทำงานจริง ๆ เช่น มีการเพิ่มรูปแบบในการร้องขอ เช่น การร้องขอที่กำหนดให้มีการตรวจสอบข้อมูลว่ามีการเปลี่ยนแปลงแก้ไขหรือไม่มี

If-Modified-Since การกำหนดการหมดอายุของข้อมูล การกำหนด รูปแบบการทำงานของพร็อกซี โดยเพิ่มระยะการทำงาน (Phase) ให้กับพร็อกซี เป็นช่วง ๆ เช่น มีช่วงเริ่มต้นการทำงาน (Warm up) การเพิ่มระดับการทำงาน (Increase) ช่วงการทำงานถึงระดับสูงสุด (Top) ช่วงการลดระดับการทำงาน (Decrease) ช่วงพักการทำงาน (Idle)

3. ชื่อเวิร์คโหลด : Polymix-3

โพลีกราฟ เวอร์ชัน : 2.5.4 ขึ้นไป

การกำหนดคุณลักษณะต่าง ๆ : ทั้งเซิร์ฟเวอร์และไคลเอนต์ ใช้การพิมพ์คำสั่งเช่นเดียวกับ Polymix-2.pg เช่น

./polysrv --config /usr/local/workloads/polymix-3.pg

ซึ่งถ้าหากไม่ ระบุ โปรแกรมก็จะไม่สามารถทำงานได้

พารามิเตอร์ของเวิร์คโหลด : ขนาดของแคช (Cache Size) , การเก็บคำร้องขอ (Fill request rate) , อัตราการร้องขอสูงสุด (Peak request rate)

วัตถุประสงค์ : ใช้กับการทดลองวัดสมรรถภาพพร็อกซีแคชครั้งที่สาม (Third cache –off) ซึ่งพัฒนาขึ้นมาโดยปรับปรุงจาก Polymix-2 เพื่อลดข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นและเพิ่มคุณสมบัติใหม่ ๆ เพื่อให้ตรงกับจุดประสงค์ในการวัดประสิทธิภาพของแคชมากยิ่งขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สำหรับการทดลองวัดสมรรถภาพของพร็อกซีที่เบคออฟใช้ในการทดลองจะมีทั้งหมด 9 เฟส ได้แก่

1. เฟสเริ่มต้น จะใช้เวลาประมาณ 10% ของเวลาที่ทดลองทั้งหมด เฟสนี้จะไม่มีผลต่อการทำงานของพร็อกซีมากนักเนื่องจากเป็นเฟสสำหรับการปรับตัวหาจุดหรือสภาวะที่พร้อมที่สุดในการทำงานของพร็อกซีก่อน
2. เฟสเพิ่มการทำงานของพร็อกซีในช่วงที่หนึ่ง เป็นเฟสเริ่มต้นการทำงานของพร็อกซีในช่วงที่หนึ่ง โดยจะค่อย ๆ เพิ่มระดับการทำงานให้กับพร็อกซี
3. เฟสเพิ่มการทำงานสูงสุดของพร็อกซีในช่วงที่หนึ่ง เป็นเฟสที่เพิ่มการทำงานของพร็อกซีสูงสุดในช่วงที่หนึ่ง
4. เฟสลดการทำงานของพร็อกซีในช่วงที่หนึ่ง เป็นเฟสเริ่มต้นลดระดับการทำงานของพร็อกซีลงในช่วงที่หนึ่ง
5. เฟสช่วงว่างจากการทำงานของพร็อกซี เป็นเฟสที่พร็อกซีได้พักจากการทำงาน
6. เฟสเพิ่มการทำงานของพร็อกซีในช่วงที่สอง เป็นเฟสเริ่มต้นการทำงานของพร็อกซีในช่วงที่สองโดยจะค่อย ๆ เพิ่มระดับการทำงานให้กับพร็อกซี
7. เฟสเพิ่มการทำงานสูงสุดของพร็อกซีในช่วงที่สอง เป็นเฟสการทำงานสูงสุดของพร็อกซีในช่วงที่สองโดยจะค่อย ๆ เพิ่มระดับการทำงานให้กับพร็อกซี
8. เฟสลดการทำงานของพร็อกซีในช่วงที่สอง เป็นเฟสเริ่มต้นลดระดับการทำงานของพร็อกซีลงในช่วงที่สอง
9. เฟสการเตรียมตัวเพื่อหยุดการทำงาน เป็นเฟสในช่วงระยะเวลาสุดท้ายของการทดลองเพื่อเตรียมที่จะหยุดการทำงานเนื่องจากครบกำหนดระยะเวลาที่ตั้งไว้ในการทดลอง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค
การใช้คำสั่งต่าง ๆ ในการทดลองสมรรถภาพของโปรแกรมสควิด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- **การใช้คำสั่งจีพร็อบ**

gprof เป็นคำสั่งในระบบยูนิกซ์ที่ใช้แสดงข้อมูลการเรียกใช้ฟังก์ชันของโปรแกรมที่เขียนด้วยภาษาซี (C) , ซีพลัสพลัส (C++) , ปาสคาล (Pascal) และภาษาฟอร์แทรน (FORTRAN) โดยจะ อ่านข้อมูลจากไฟล์ชื่อว่า gmon.out ซึ่งได้จากการคอมไพล์โปรแกรมด้วยการใส่ออปชัน –pg

รูปแบบการใช้คำสั่ง gprof

gprof [-abcsz] [-e|name] [-f|-F name] [objfile[gmon.out]

ตัวอย่าง เช่น ต้องการทราบว่าโปรแกรม test.cc มีการเรียกใช้ฟังก์ชันใดในการทำงานบ้าง

ขั้นตอน

1. คอมไพล์เพื่อให้ได้ไฟล์ gmon.out โดยใช้คำสั่งดังนี้

gcc -o -pg test test.cc

-o เป็นการระบุ ให้เปลี่ยนชื่อไฟล์ test.cc ไปเป็น test เมื่อคอมไพล์เรียบร้อยแล้วซึ่งถ้าหากไม่ระบุโปรแกรมจะถูกเก็บไว้ในชื่อ a.out

-pg เป็นการระบุให้มีการสร้างไฟล์เพื่อจัดเก็บข้อมูลในการทำงานของโปรแกรมว่ามีการเรียกใช้ฟังก์ชันใดและใช้เวลาเท่าไรในการทำงาน

2. ใช้คำสั่ง gprof เพื่อนำข้อมูลมาแสดงเก็บไว้ในไฟล์ โดยใช้คำสั่ง

gprof -c test gmon.out > time1 (ระบุชื่อไฟล์ที่จะเก็บผล)

หรือ gprof -c a.out gmon.out > time1 (ในกรณีที่ไม่ได้เปลี่ยนชื่อไฟล์ขณะสั่งคอมไพล์โปรแกรม)

3. เมื่อต้องการดูผล อาจจะใช้คำสั่ง pico หรือใช้คำสั่ง vi หรือคำสั่ง cat ก็ได้

เช่น vi time1 หรือ pico time1 หรือ pico time1

ซึ่งข้อมูลที่แสดงจากไฟล์ time2 จะประกอบด้วยข้อมูลสองส่วน ดังนี้

1. Flat Profile

แสดงรายชื่อฟังก์ชันที่เรียกใช้โดยเรียงตามลำดับเวลาที่ใช้ในแต่ละฟังก์ชันจากมากไปน้อยสำหรับการทำงานของโปรแกรมในครั้งหนึ่ง ๆ จนกระทั่งจบการทำงาน เวลาที่ใช้แสดงในหน่วย ไมโครวินาที

ตัวอย่าง Flat Profile

ตารางที่ ค1 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วนที่เป็น Flat Profile ของจีพร็อบ

% time	cumulative seconds	Self seconds	Called	self us/call	total us/call	name
11.44	20.01	20.01	249462	80.21	682.32	main [1]
7.70	33.48	13.47	6237468	2.16	2.85	create [2]
4.93	42.10	8.62	21201687	0.41	0.79	report [7]

คำอธิบายความหมายส่วนหัวของไฟล์บันทึกเข้าออกส่วน flat profile

ตารางที่ ค2 แสดงการอธิบายความหมายของข้อมูลในแฟ้มบันทึกเข้าออกส่วนที่เป็น Flat Profile

ชื่อข้อมูลที่เก็บ	คำอธิบาย
% time	หมายถึง ร้อยละของเวลาที่โปรแกรมสควิดใช้สำหรับการทำงานครั้งหนึ่ง ๆ ของแต่ละฟังก์ชัน
Comulative seconds	หมายถึง เวลารวมทั้งหมดที่ใช้จากฟังก์ชันลำดับต้น ๆ จนถึงฟังก์ชันนั้น ๆ ที่โปรแกรมสควิดใช้สำหรับการทำงานครั้งหนึ่ง ๆ จนกระทั่งจบโปรแกรม มีหน่วยเป็นวินาที
self seconds	หมายถึง เวลาที่ใช้เฉพาะฟังก์ชันนั้น ๆ โดยไม่รวมถึงเวลาที่เรียกใช้ฟังก์ชันย่อย ๆ ของฟังก์ชันนั้น มีหน่วยเป็นวินาที ซึ่งไฟล์บันทึกเข้าออกจะใช้เป็นตัวเรียงลำดับการใช้เวลาของแต่ละฟังก์ชันจากมากไปน้อย
calls	หมายถึง จำนวนครั้งที่แต่ละฟังก์ชันถูกเรียกสำหรับการทำงานของโปรแกรมสควิดจนกระทั่งจบโปรแกรม
Self us/call	หมายถึง เวลาเฉลี่ยในหน่วยไมโครวินาทีต่อการเรียกใช้ฟังก์ชันนั้น ๆ หนึ่งครั้งสำหรับการทำงานของโปรแกรมสควิดจนกระทั่งจบโปรแกรม
total us/call	หมายถึง เวลาเฉลี่ยในหน่วยไมโครวินาทีต่อการเรียกใช้ฟังก์ชันนั้น ๆ หนึ่งครั้งซึ่งรวมถึงเวลาที่ใช้ในฟังก์ชันย่อยเมื่อถูกฟังก์ชันนั้น ๆ เรียกใช้ของโปรแกรมสควิดสำหรับการทำงานในครั้งหนึ่งจนกระทั่งจบโปรแกรม
name	หมายถึง ชื่อของฟังก์ชันที่ถูกเรียกใช้สำหรับการทำงานของโปรแกรมสควิดจนกระทั่งจบโปรแกรม

2. Call Graph

แสดงเวลาที่ใช้ในแต่ละฟังก์รวมถึงเวลาที่ใช้ในฟังก์ชันย่อย ๆ ที่ฟังก์ชันนั้น ๆ เรียกใช้สำหรับการทำงานครั้งหนึ่ง ๆ ของโปรแกรมจนกระทั่งจบโปรแกรม

Call Graph แบ่งออกเป็น 3 ส่วน ได้แก่

1. Primary Line แสดงถึงฟังก์ชันนั้น ๆ ไปเรียกใช้ฟังก์ชันย่อย ๆใดบ้าง สำหรับการทำงานครั้งหนึ่ง ๆ ของโปรแกรมจนกระทั่งจบโปรแกรม
2. Line for a function's Callers แสดงถึงฟังก์ชันนั้น ๆ ถูกฟังก์ชันใดเรียกใช้ สำหรับการทำงานครั้งหนึ่ง ๆ ของโปรแกรมจนกระทั่งจบโปรแกรม
3. Line for a function's Subroutines แสดงถึงฟังก์ชันย่อย ๆ ที่ถูกฟังก์ชันนั้น ๆ เรียกใช้สำหรับการทำงานครั้งหนึ่ง ๆ ของโปรแกรมจนกระทั่งจบโปรแกรม

ตัวอย่าง Call Graph

- Primary Line

ตารางที่ ค3 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วนที่เป็น Primary Line ของข้อมูลที่เก็บในแฟ้มบันทึกเข้าออกส่วนที่เป็น Call Graph

index	% time	self	Children	called	name
[1]	100.0	0.38	174.60		main [1] **(Primary Line)**
		20.01	150.20	249462/249462	comm_poll [2]
		0.69	3.32	249462/249462	eventRun [46]

- Line for a function's Callers

ตารางที่ ค4 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วนที่เป็น Line for a function's Callers ของข้อมูลที่เก็บในแฟ้มบันทึกเข้าออกส่วนที่เป็น Call Graph

index	% time	self	Children	called	name
		20.01	150.20	249462/249462	main [1] **(Callers)**
[2]		20.01	150.20	249462	comm_poll [2]

- Line for a function's Subroutines

ตารางที่ ค5 แสดงตัวอย่างแฟ้มบันทึกเข้าออกส่วนที่เป็น Line for a function's Sibroutines ของข้อมูลที่เก็บในแฟ้มบันทึกเข้าออกส่วนที่เป็น Call Graph

index	% time	self	Children	called	name
[2]	97.3	20.01	150.20	249462	comm_poll [2]
		0.72	52.34	149184/149184	commHandleWrite [4]
		0.32	48.61	50910/50910	httpReadReply [5]

Subroutines

คำอธิบายความหมายส่วนหัวของไฟล์บันทึกเข้าออกส่วน Call Graph

ตารางที่ ค6 แสดงการอธิบายความหมายของข้อมูลในแฟ้มบันทึกเข้าออกส่วนที่เป็น Call Graph

ชื่อข้อมูลที่เก็บ	**คำอธิบาย**
index	หมายถึง แสดงถึงหมายเลขชื่อฟังก์ชันเพื่อช่วยในการอ้างถึงและพิจารณาในกรณีที่เป็นฟังก์ชันย่อยซึ่งถูกฟังก์ชันอื่น ๆ เรียกใช้ หรือเป็นฟังก์ชันที่เรียกใช้ฟังก์ชันอื่น ๆ
% time	หมายถึง ร้อยละของเวลาที่ใช้ทั้งหมดในแต่ละฟังก์ชันซึ่งรวมถึงเวลาที่ใช้ในฟังก์ชันย่อย ๆ ที่ฟังก์ชันนั้น ๆ เรียกใช้ด้วย
self	หมายถึง เวลาที่ใช้ทั้งหมดในฟังก์ชันนั้น ๆ โดยมีหน่วยเป็นวินาที
children	หมายถึง เวลาที่ใช้ทั้งหมดในฟังก์ชันย่อยๆ ที่ฟังก์ชันนั้น ๆ เรียกใช้
called	หมายถึง จำนวนครั้งที่ฟังก์ชันนั้น ๆ ถูกเรียกใช้ และถ้าหากฟังก์ชันนั้น ๆ มีการเรียกใช้ตัวมันเอง (Recursive) จะมีจำนวนตัวเลขสองตัวแบ่งคั่นโดยเครื่องหมาย “+” หมายเลขตัวแรก แสดงถึงจำนวนครั้งที่ฟังก์ชั้นนั้น ๆ ถูกเรียกใช้ ส่วนหมายเลขตัวที่สองแสดงถึงจำนวนครั้งที่มีการเรียกใช้ฟังก์ชันตัวมันเอง
name	หมายถึง ขื่อของฟังก์ชัน

- คำสั่งในการรันสควิด

  ./squid

- คำสั่งการรันเว็บโพลีกราฟส่วนผู้ให้บริการ

  ./polysrv --config /usr/local/polygraph/workloads/polymix-2.pg --verb_lvl 10 --
  cfg_dirs /usr/local/polygraph/workloads/include --log /tmp/s20.log

- คำสั่งการรันเว็บโพลีกราฟส่วนผู้ใช้

  ./polyclt --config /usr/local/polygraph/workloads/polymix-2.pg --verb_lvl 10 –
  cfg_dirs /usr/local/polygraph/workloads/include --log /tmp/c5.log --proxy
  161.200.93.168:3128

- คำสั่งการจำลองไอพีแอดเดรส โดยใช้ aka tool ของเว็บโพลีกราฟ

  ./aka interface <หมายเลขไอพีแอดเดรสที่จำลอง>

  เช่น ./aka eth0 10.167.129-131.1-50

  eth0 เป็นอินเทอร์เฟสการ์ดแลนด์

- **การกำหนดไอพีแอดเดรส มีลักษณะการกำหนดดังนี้**

รูปแบบ เซิร์ฟเวอร์ 10.x.129-131.x-x

ไคลเอนต์ 10.x.x-x.x-x

x หมายถึงตัวเลขที่ให้ผู้ใช้กำหนดเอง ส่วน 129-131 จะเป็นช่วงค่าที่กำหนดไว้สำหรับการจำลองไอพีแอดเดรสในส่วนที่เป็นเซิร์ฟเวอร์

- **สูตรการคำนวณหาจำนวนของไคลเอนต์และเซิร์ฟเวอร์เพื่อการจำลองไอพีแอดเดรสในการสร้างอัตราการร้อง**
  - จำนวนเซิร์ฟเวอร์ = 500 + Req_Rate/10
  - จำนวนไคล์เอนต์ = 2.5 * Req_Rate

    request_rate หมายถึงจำนวนคำร้องขอต่อวินาทีที่ผู้ใช้ต้องการจำลอง เช่น

    - 10 คำร้องขอ/วินาที ผู้ใช้จะต้องทำการจำลองไอพีในส่วนที่เป็นเซิร์ฟเวอร์เท่ากับ

    จำนวนเซิร์ฟเวอร์ = 500

    จำนวนไคล์เอนต์ = 25

ภาคผนวก ง
โครงสร้างข้อมูลของโปรแกรมสควิดที่ใช้จัดเก็บข้อมูล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- **โครงสร้างข้อมูลของโปรแกรมสควิดที่ใช้จัดเก็บข้อมูล**

1. StoreEntry

```
struct _StoreEntry {
    hash_link hash;                 /* must be first */
    MemObject *mem_obj;
    RemovalPolicyNode repl;
    time_t timestamp;
    time_t lastref;
    time_t expires;
    time_t lastmod;
    size_t swap_file_sz;
    u_short refcount;
    u_short flags;
    sfileno swap_filen:25;
    sdirno swap_dirn:7;
    u_short lock_count;             /* Assume < 65536! */
    mem_status_t mem_status:3;
    ping_status_t ping_status:3;
    store_status_t store_status:3;
    swap_status_t swap_status:3;
```

2. storeIOState

```
struct _storeIOState {
    sdirno swap_dirn;
    sfileno swap_filen;
    StoreEntry *e;                  /* Need this so the FS layers can play god */
    mode_t mode;
    size_t st_size;                 /* do stat(2) after read open */
    off_t offset;               /* current on-disk offset pointer */
    STFNCB *file_callback;      /* called on delayed sfileno assignments */
    STIOCB *callback;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    void *callback_data;
    struct {
        STRCB *callback;
        void *callback_data;
    } read;
    struct {
        unsigned int closing:1; /* debugging aid */
    } flags;
    void *fsstate;
};
```

3. SwapDir

```
struct _SwapDir {
    char *type;
    int cur_size;
    int low_size;
    int max_size;
    char *path;
    int index;              /* This entry's index into the swapDirs array */
    int suggest;
    ssize_t max_objsize;
    RemovalPolicy *repl;
    int removals;
    int scanned;
    struct {
        unsigned int selected:1;
        unsigned int read_only:1;
    } flags;
    STINIT *init;           /* Initialise the fs */
    STNEWFS *newfs;         /* Create a new fs */
    STDUMP *dump;           /* Dump fs config snippet */
    STFREE *freefs;         /* Free the fs data */
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
STDBLCHECK *dblcheck;   /* Double check the obj integrity */
STSTATFS *statfs;       /* Dump fs statistics */
STMAINTAINFS *maintainfs;/* Replacement maintainence */
STCHECKOBJ *checkobj;   /* Check if the fs will store an object */
/* These two are notifications */
STREFOBJ *refobj;       /* Reference this object */
STUNREFOBJ *unrefobj;   /* Unreference this object */
STCALLBACK *callback;   /* Handle pending callbacks */
STSYNC *sync;           /* Sync the directory */
struct {
        STOBJCREATE *create;
        STOBJOPEN *open;
        STOBJCLOSE *close;
        STOBJREAD *read;
        STOBJWRITE *write;
        STOBJUNLINK *unlink;
} obj;
struct {
        STLOGOPEN *open;
        STLOGCLOSE *close;
        STLOGWRITE *write;
        struct {
            STLOGCLEANSTART *start;
            STLOGCLEANNEXTENTRY *nextentry;
            STLOGCLEANWRITE *write;
            STLOGCLEANDONE *done;
            void *state;
        } clean;
        int writes_since_clean;
} log;
struct {
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        int blksize;
    } fs;
    void *fsdata;
};
```


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก จ
รูปแบบของแฟ้มบันทึกเข้าออกของโปรแกรมสควิด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- **รูปแบบของแฟ้มบันทึกเข้าออกของโปรแกรมสควิด**

Time Elapsed RemoteHost Code/Status Bytes Method URL rfc931 Type

Time	เวลาที่บันทึกการร้องขอ
Elapsed	เวลาที่ใช้ในการในการดึงข้อมูลมาให้ผู้ใช้ มีหน่วยเป็นมิลลิวินาที
RemoteHost	เลขที่อยู่ของเครื่องที่ผู้ใช้ทำการร้องขอเอกสาร
Code	รหัสแสดงสถานะการประมวลผลเอกสารที่ได้ของแคช
Status	รหัสการตอบกลับ
Bytes	ขนาดของข้อมูล
Method	วิธีการร้องขอข้อมูล
URL	ที่อยู่ของเอกสารที่ทำการร้องขอ
rfc931	ชื่อแสดงผู้ใช้ที่ทำการร้องขอข้อมูล
Type	ชนิดของข้อมูลที่ทำการร้องขอ

- **วิธีการร้องขอของผู้ใช้** (Client Request Methods)

1. GET เป็นวิธีการที่ระบุให้นำข้อมูลจากเซิร์ฟเวอร์และส่งข้อมูลไปยังผู้ร้องขอ
2. HEAD คล้ายกับคำสั่ง GET แต่ระบุให้เซิร์ฟเวอร์ส่งข้อมูลเฉพาะที่เป็นส่วนหัวเท่านั้น
3. POST เป็นการส่งข้อมูลจากผู้ใช้ไปยังผู้ให้บริการ
4. DELETE เป็นการระบุให้เซิรฟเวอร์ลบยูอาร์แอลที่กำหนดไว้ออกจากเซิร์ฟเวอร์
5. PUT เป็นรูปแบบการร้องขอคล้ายกับ POST แต่ไม่เป็นที่นิยมใช้
6. TRACE เป็นรูปแบบการร้องขอเพื่อทำการเรียกดูข้อความของผู้ใช้ในการร้องขอ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวนุชจรินทร์ อาจปรุ เกิดเมื่อวันที่ 25 พฤษภาคม พ.ศ. 2519 สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี สถาบันราชภัฏเทพสตรี เมื่อปีการศึกษา 2542 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2542

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย