สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบเว็บเซิร์ฟเวอร์หลายเครื่องแบบบาลานซ์โหลด

A Design of Balance Load Web Server

นายรักวิทย์ นันทปาลียอง
นายรังสรรค์ ปันทุราภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่
เลขทะเบียน 37055
วัน, เดือน, ปี 30 ส.ค. 2543

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบเว็บเซิร์ฟเวอร์หลายเครื่องแบบบาลานซ์โหลด

A Design of Balance Load Web Server

โดย

นายรักวิทย์ นันทปาลียอง

นายรังสรรค์ ปันทุราภรณ์

อาจารย์ที่ปรึกษา

อาจารย์ นวพร วรรณวิมลศรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบเว็บเซิร์ฟเวอร์หลายเครื่องแบบบาลานซ์โหลด

A Design of Balance Load Web Server

ผู้จัดทำ

1. นายรักวิทย์ นันทปาลียอง รหัสประจำตัว 40013263
2. นายรังสรรค์ ปันทุราภรณ์ รหัสประจำตัว 40013264

อาจารย์ที่ปรึกษา

(อาจารย์ นวพร วรรณวิมลศรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบเว็บเซิร์ฟเวอร์หลายเครื่องแบบบาลานซ์โหลด

นายรักวิทย์ นันทปาลียอง
นายรังสรรค์ ปันทุราภรณ์
อาจารย์ นวพร วรรณวิมลศรี อาจารย์ที่ปรึกษา
ปีการศึกษาที่ 2542

บทคัดย่อ

โครงงานนี้ลักษณะโดยรวมจะเป็นการออกแบบและทำการทดลองในโมเดลต่างๆ ซึ่งโมเดลต่างๆนั้นมีทั้งโมเดลที่มีใช้งานกันอย่างแพร่หลายแล้วและโมเดลที่ออกแบบขึ้นมาใหม่ โดยการทำการทดลองนั้นจะกำหนดเงื่อนไขและตัวแปรต่างๆอย่างชัดเจน ตัวอย่างเช่น ทำการทดลองในเน็ตเวิร์กระบบปิด, ลักษณะของชุดไฟล์ที่ทดสอบเป็นต้น การทดลองส่วนใหญ่จะเป็นการเปรียบเทียบการนำรีเวิร์สพร็อกซีเข้ามาช่วยในการเพิ่มประสิทธิภาพของเว็บไซต์ ในเงื่อนไขและตัวแปรต่างๆที่เรากำหนดขึ้น ซึ่งจากผลการทดลองส่วนใหญ่ปรากฏว่าในสภาวะแวดล้อม,ตัวแปรและเงื่อนไขต่างๆที่เรากำหนดสำหรับการทดลองในครั้งนี้ การนำรีเวิร์สพร็อกซีเข้ามาช่วย ไม่ได้ทำให้ประสิทธิภาพการทำงานโดยรวมดีขึ้นแต่อย่างใด และผลการทดลองที่ได้สามารถนำไปเป็นแนวทางสำหรับการจัดสภาวะแวดล้อม,เงื่อนไขและตัวแปรต่างๆในการทดสอบในอนาคตได้ เพื่อหาสภาวะแวดล้อม,เงื่อนไขและตัวแปรที่เหมาะสมสำหรับการนำรีเวิร์สพร็อกซีเข้าไปช่วยเพิ่มประสิทธิภาพการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A Design of Balance Load Web Server

Rukwit Nuntapaleeyong
Rungsant Phunturaporn
Navaporn Wanvimonsri Advisor

Abstract

The objectives of this thesis are to design and experiment with the webserver models. These webserver models may be widely used or just created. The strong conditions and variables declaration ,such as the environment in closed system or the characteristics of tested files, are essential for this experiment. Most of experiment are the performance comparisons by using the reverse proxy in the different conditions and variables. Its result shows using the reverse proxy does not increase the performance under the conditions which be set. This is able to be the tendency for finding the environment, conditions, and variables that increase the efficiency in more advance experiments.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ นวพร วรรณวิมลศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ อาจารย์ธนา หงษ์สุวรรณ และอาจารย์อัครเดช วัชระภูพงษ์ ที่ให้ใช้สถานที่ห้อง ISAG เป็นสถานที่ในการทำการทดลองเป็นส่วนใหญ่

ขอขอบใจเพื่อนๆชาว ISAG ทุกคนที่ช่วยเหลือในการทำงานและแก้ไขอุปสรรคต่างๆในการทำงานให้ผ่านพ้นไปด้วยดี เป็นที่ปรึกษาและกำลังใจที่ดีเสมอมา

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

รักวิทย์ นันทปาลียอง
รังสรรค์ ปันทุราภรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

1.1. ความสำคัญและที่มา

ปัจจุบันการใช้งานอินเทอร์เน็ตได้รับความนิยมอย่างกว้างขวาง ซึ่งบริการหนึ่งที่นิยมใช้งานกันมากคือ บริการเกี่ยวกับเอกสารอิเล็กทรอนิกส์ (HTML) ซึ่งบริการให้โดยโปรแกรมเว็บเซิร์ฟเวอร์ แน่นอนว่าเมื่อเว็บไซต์ใดที่ได้รับความนิยมสูง จำนวนของผู้เข้าเยี่ยมชมก็จะมากตามไปด้วย ส่งผลถึงภาระงานที่เพิ่มมากขึ้นของเซิร์ฟเวอร์ ตัวอย่างของเว็บไซต์ที่มีภาระงานมากๆ เช่น ไซต์ของบริษัทไมโครซอฟต์ เป็นต้น ทำให้ผู้เข้าเยี่ยมชมเว็บไซต์ดังกล่าวรู้สึกว่าเว็บไซต์ดังกล่าวทำงานช้า และอาจทำให้ความนิยมในการเข้าเยี่ยมชมเว็บไซต์ดังกล่าวน้อยลง กว่าที่ควรจะเป็น

จากปัญหาดังกล่าวข้างต้น สามารถที่จะแยกพิจารณาได้ 2 ประเด็นคือ

1.1.1 ปัญหาอยู่ที่ระบบเครือข่าย(Network)

เป็นไปได้ว่าขนาดแบนด์วิดธ์ของการเชื่อมสัญญาณจากอินเทอร์เน็ตสู่เว็บเซิร์ฟเวอร์มีขนาดที่น้อยเกินไป ไม่เพียงพอต่อความต้องการที่แท้จริง

1.1.2 ปัญหาอยู่ที่ความเร็วของเว็บเซิร์ฟเวอร์

เป็นไปได้ว่าประสิทธิภาพการทำงานของเครื่องเซิร์ฟเวอร์มีไม่เพียงพอ เช่น ความเร็วของสัญญาณนาฬิกาที่ช้าไป, จำนวนหน่วยความจำหลักมีน้อยเกินไป, ประสิทธิภาพการทำงานของอุปกรณ์อินพุต/เอาต์พุต

จากความต้องการที่จะเพิ่มประสิทธิภาพการทำงานให้กับเว็บไซต์ ให้สามารถที่จะรองรับจำนวนของผู้ร้องขอเข้ามาให้ได้มากที่สุด ซึ่งปรกติตามความเข้าใจของคนทั่วไปคือ หนึ่งเว็บไซต์ก็จะมีเครื่องเซิร์ฟเวอร์ทำงานรองรับเว็บไซต์นั้นอยู่เพียงหนึ่งเครื่อง แต่เมื่อเว็บไซต์เป็นที่นิยมของคนทั่วไปแล้ว จะมีสัญญาณร้องขอเข้ามาอย่างมากมาย เกินที่เครื่องเซิร์ฟเวอร์เครื่องเดียวจะรับได้ ซึ่งจำเป็นต้องนำเทคนิคการทำบาลานซ์โหลด มาช่วยกันในการให้บริการข้อมูลให้กับเว็บไซต์ หรือไม่เช่นนั้นก็ต้อง เปลี่ยนเครื่องเซิร์ฟเวอร์ให้เป็นเครื่องที่มีประสิทธิภาพที่สูงกว่าเครื่องเดิม

จึงเป็นที่มาของการทำโครงงานนี้ เพื่อศึกษารายละเอียดและเทคนิคการทำบาลานซ์โหลด เว็บเซิร์ฟเวอร์แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2. วัตถุประสงค์ของงาน

1.2.1 เพื่อศึกษาการหลักการทำงานของโดเมนเนมเซิร์ฟเวอร์

1.2.2 สามารถปรับแต่งการทำงานของโดเมนเนมเซิร์ฟเวอร์ได้

1.2.3 เพื่อศึกษาหลักการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์

1.2.4 สามารถปรับแต่งการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ได้

1.2.5 เพื่อศึกษาหลักการทำงานของโปรแกรมพรอกซี่เซิร์ฟเวอร์

1.2.6 สามารถปรับแต่งการทำงานของโปรแกรมพร็อกซีเซิร์ฟเวอร์ได้

1.2.7 สามารถปรับปรุงและออกแบบตัวอย่างของการเพิ่มประสิทธิภาพได้

1.3. ขอบเขตของงานวิจัย

การทดลองและวิจัยของโครงงานครั้งนี้ขอบเขตอยู่ที่สามารถสรุปผลการทดลองได้ ในเงื่อนไขและตัวแปรต่างๆของการทดลองที่กำหนดขึ้น การทดลองทั้งหมดมี 4 โมเดล โดยลักษณะของแต่ละโมเดลจะมีตั้งแต่โมเดลที่ง่ายในการทดลองไปจนถึงโมเดลที่ยากในการติดตั้งและทดลอง

1.4. ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 นักศึกษาสามารถเข้าใจถึงแนวความคิดของการต่อใช้งานเครื่องเว็บเซิร์ฟเวอร์ร่วมกัน

1.4.2 นักศึกษาสามารถเปรียบเทียบประสิทธิภาพของการปรับปรุงของแต่ละตัวอย่าง

1.4.3 นักศึกษาสามารถที่จะเสนอแนะ เป็นแนวทางปฏิบัติในการปรับปรุงประสิทธิภาพของเครื่องเว็บเซิร์ฟเวอร์ได้

1.4.4 ตัวอย่างใหม่ที่นักศึกษาเป็นผู้คิดค้นขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5. วิธีการดำเนินงาน

ขั้นตอนการดำเนินงานจะแบบออกเป็น 4 ส่วนงานใหญ่ คือ

1.5.1 ขั้นตอนการศึกษาหาข้อมูลและออกแบบ

เป็นการศึกษาร่วมรวมข้อมูลต่างๆ ส่วนมากจะได้ข้อมูลมาจากอินเทอร์เน็ต เพราะเรื่องดังกล่าวนี้เป็นเรื่องที่ข้องข้างใหม่ที่จะหาเอกสารภาษาไทยมาศึกษา แม้กระทั้งหนังสือในห้องสมุดแทบจะไม่มีเลยก็ว่าได้

1.5.2 ขั้นตอนการติดตั้งเซิร์ฟเวอร์ต่างๆ

เป็นขั้นตอนของการติดตั้งเซิร์ฟเวอร์ต่างๆ ที่จำเป็นต้องใช้งานในการทำโครงงานนี้ มีดังนี้

เซิร์ฟเวอร์ที่ต้องติดตั้ง

- Domain Name Server
- Apache Web Server
- Squid Web Server
- Linux Server

1.5.3 ขั้นตอนทดลองและปรับแต่ง

เป็นขั้นตอนของการทำการทดลองและปรับแต่งประสิทธิภาพการทำงานของแต่ละเซิร์ฟเวอร์ให้มีประสิทธิภาพการทำงานสูงที่สุด เท่าที่ประสิทธิภาพของเครื่องจะอำนวย โดยการทดลองจะแบบการทดลองออกเป็นแต่ละการทดลองโดยแต่ละการทดลองจะมีการนำค่ามาเปรียบเทียบกับประสิทธิภาพของเซิร์ฟเวอร์มาตราฐานที่เรากำหนดขึ้น

1.5.4. ขั้นตอนการสรุปผล

เป็นขั้นตอนของการสรุปผล โดยเปรียบเทียบชี้ประเด็นให้เป็นความแตกต่างของแต่ละตัวอย่างและทรัพยากรที่ต้องใช้ในการปรับปรุงแต่ละตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

2.1. เทคโนโลยีอินเทอร์เน็ต

จุดกำเนิดของอินเทอร์เน็ตมาจากความคิดเชิงยุทธศาสตร์ทางการทหารในช่วงปี พ.ศ. 2512 สงครามเย็นระหว่างสหรัฐฯกับรัสเซียยังก่อตัวอย่างเงียบๆ ทางกระทรวงกลาโหมของสหรัฐฯ ถือว่าการสื่อสารของคอมพิวเตอร์เป็นหัวใจหลัก หากคอมพิวเตอร์จุดใดจุดหนึ่งถูกบอมบ์ขึ้นมา การสื่อสารจะถูกตัดขาดซึ่งจะทำให้เสียเปรียบเชิงยุทธศาสตร์เป็นอย่างมาก ทางกระทรวงกลาโหมจึงเจียดงบประมาณมาพัฒนาเครือข่ายคอมพิวเตอร์แบบใหม่ขึ้น ซึ่งจะต้องมีคุณสมบัติ "อึด" ต่อการถูกทำลาย กล่าวคือเมื่อคอมพิวเตอร์จุดใดจุดหนึ่งถูกทำลายไป คอมพิวเตอร์ส่วนอื่นๆ จะสามารถหาเส้นทางใหม่เพื่อสื่อสารได้ โครงการคอมพิวเตอร์นี้ชื่อว่า ARPANET (Advanced Research Projects Agency Network)

ก้าวแรกของ ARPANET ประกอบด้วยเครื่องคอมพิวเตอร์ 4 เครื่อง คือ คอมพิวเตอร์ของมหาวิทยาลัยยูทาห์ มหาวิทยาลัยแคลิฟอร์เนียที่ซานตาบาบารา มหาวิทยาลัยแคลิฟอร์เนียที่ลอสแองเจลิส และสถาบันวิจัยของมหาวิทยาลัยสแตนฟอร์ด หลังจากมีการทดสอบการใช้งาน ARPANET ไปได้ระยะหนึ่ง ซึ่งผลทดสอบเป็นที่น่าพอใจ กระทรวงกลาโหมของสหรัฐฯ จึงมีการขยายเครือข่ายของ ARPANET ออกไปอีก โดยเชื่อมต่อกับคอมพิวเตอร์ของมหาวิทยาลัยและสถาบันวิจัยต่างๆ รวม 50 แห่ง โดยใช้โพรโตคอล (Protocol) ที่มีชื่อว่า NCP (Network Control Protocol)

แต่ต่อมาเมื่อมีการขยายเครือข่าย ARPANET ออกไปอีก ก็พบว่าโพรโตคอล NCP นี้มีข้อจำกัดอยู่มาก จึงมีการพัฒนาโพรโตคอลตัวใหม่ขึ้นมา ซึ่งมีชื่อว่า TCP/IP (Transmission Control Protocol/Internet Protocol) ซึ่งจุดเด่นของโพรโตคอลตัวใหม่นี้ก็คือ การทำให้เครื่องคอมพิวเตอร์แต่ละแบบสามารถสื่อสารกันได้ ต่อมาผู้พัฒนาระบบปฏิบัติการ Unix ได้นำเอาโพรโตคอลนี้เข้าเป็นส่วนหนึ่งของระบบปฏิบัติการด้วย และจากจุดนี้เองทำให้เครือข่ายคอมพิวเตอร์ ARPANET เติบโตขึ้นมาจาก 500 เครื่องในปีนี้ พ.ศ. 2525 มาเป็น 1,000 เครื่องในปีถัดมา

ต่อมาในปี พ.ศ. 2529 มูลนิธิวิทยาศาสตร์แหล่งชาติสหรัฐฯ หรือ NSF (National Science Foundation) ได้สร้างเครือข่ายคอมพิวเตอร์อีกระบบหนึ่งขึ้นมาเรียกว่า NSFNET (National Science Foundation Network) ซึ่งประกอบไปด้วยซูเปอร์คอมพิวเตอร์ 5 เครื่องเชื่อมต่อเข้าด้วยกัน โดยใช้โพรโตคอล TCP/IP ในการสื่อสาร สำหรับจุดมุ่งหมายของเครือข่ายนี้ ก็เพื่อใช้ในการวิจัยและการศึกษา ต่อมามีการเชื่อมต่อกับเครือข่ายนี้จำนวนมาก เนื่องจากต้องการใช้งานซูเปอร์คอมพิวเตอร์ และในที่สุดเครือข่าย ARPANET ก็มีการขอเชื่อมต่อกับ NSFNET ด้วย นอกจากเครือข่ายอื่นไม่ว่าจะเป็น UUNET, UUCP, BITNET หรือ CSNET ก็ของเชื่อมต่อเข้ากับ NSFNET ด้วยจนในปี พ.ศ. 2532 มีคอมพิวเตอร์ที่เชื่อมต่อกว่า 100,000 เครื่องทีเดียว

แต่เนื่องจากเจ้าของ NSFNET เป็นองค์กรไม่แสวงหากำไรและมีงบประมาณอยู่จำกัดทาง NSFNET จึงได้ผลักภาระการดำเนินงานเครือข่ายนี้ไปให้บริษัท Advanced Network and Service (ANS) แทน ซึ่งเป็นบริษัทร่วมทุนระหว่างบริษัท MCI, IBM และ มหาวิทยาลัยมิชิแกน และก็มีการเปลี่ยนชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NSFNET เป็นอินเทอร์เน็ตจนถึงปัจจุบัน และ ณ วันนี้ มีอินเทอร์เน็ตเซิร์ฟมากกว่า 10 ล้านเครื่อง และประมาณกันว่ามีผู้ใช้งานอินเทอร์เน็ตทั่วโลกมากกว่า 100 ล้านคน

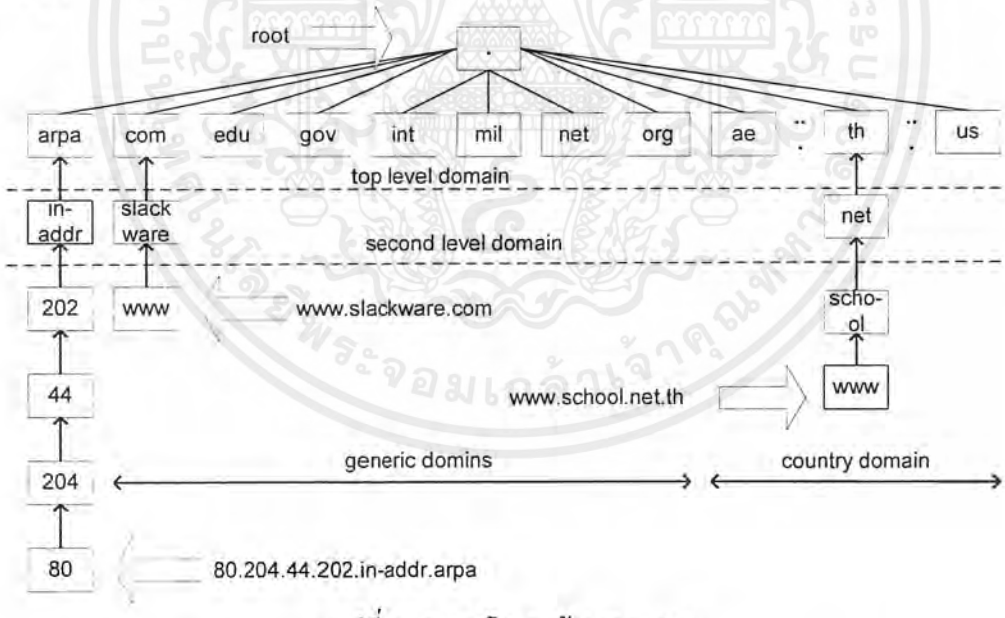
2.2. โดเมนเนมเซิร์ฟเวอร์

2.2.1. พื้นฐาน DNS

เครื่องคอมพิวเตอร์และอุปกรณ์สื่อสารต่างๆ ที่อยู่บนเครือข่ายอินเทอร์เน็ตนั้น แต่ล่ะตัวจะต้องมีเลขหมายเลขประจำเครื่อง(IP Address) ที่ต้องไม่ซ้ำกัน การติดต่อสื่อสารระหว่างเครื่องต่างๆ จะใช้หมายเลขประจำเครื่องเป็นหลัก แต่บางท่านอาจสงสัยว่าเหตุใดเราจึงเสามารถดูเว็บไซต์ต่างๆ โดยการพิมพ์ชื่อ เช่น www.cnn.com ก็สามารถไปยังเว็บไซต์ของสำนักข่าว CNN ได้ โดยไม่ต้องใส่หมายเลขประจำเครื่อง เหตุที่เป็นดังนี้ เพราะว่าเรามีระบบ DNS

ระบบโดเมนเนม (Domain Name System หรือ DNS) เป็นระบบการแปลงกลับไปมาระหว่างชื่อโฮสต์ (host) ให้เป็น หมายเลขประจำเครื่องโดยการเปลี่ยนจากชื่อโฮสต์เป็นหมายเลขประจำเครื่องจะเรียกว่า Forward mapping ส่วนการเปลี่ยนจากหมายเลขประจำเครื่องเป็นชื่อโฮสต์จะเรียกว่า การแปรงกลับ (Reverse mapping)

โครงสร้างของ DNS จะมีลักษณะคล้ายกับโครงสร้างไฟล์ของระบบ UNIX(Unix File System) ดังรูปที่ 2.1



รูปที่ 2.1 แสดงโครงสร้างของ DNS

จากรูปที่ 2.1 จุดบนสุดคือ รูต(Root) บางทีจะเรียกว่า . {อ่านว่าดอต(dot)}แล้วจะมีการแยกย่อยออกมาเป็น top level domain และ second level domain ลงมาเรื่อยๆ แต่ละโหนด(node) สามารถมีชื่อได้ 63 ตัวอักษรจะใช้ตัวอักษรใหญ่หรือเล็กก็ได้ไม่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดเมนเนมของโหนดใดๆ จะเริ่มที่โหนดนั้นๆ และเลื่อนไปจนกระทั่งถึงรูต โดยใช้ดอตเป็นตัวแยกชื่อโดเมนเนมเดียวกัน

ระดับของโดเมนสามารถแบ่งได้ดังนี้

1. arpa เป็นโดเมนพิเศษซึ่งใช้สำหรับการแปลงกลับ (Reverse Mapping)หมายเลขประจำเครื่อง เป็น ชื่อโดเมนเนม
2. โดเมนที่มีตัวอักษร 3 ตัวซึ่งมีอยู่ 7 โดเมนเรียกว่า **generic domain** ซึ่งส่วนมากจะใช้ในสหรัฐอเมริกา

ชื่อโดเมน	คำอธิบาย
1. Com	องค์กรธุรกิจ
2. Edu	สถาบันการศึกษา
3. Gov	หน่วยงานราชการ
4. Int	องค์กรระหว่างประเทศ
5. Mil	หน่วยงานทหาร
6. Net	ผู้ให้บริการเครือข่าย
7. Org	องค์กรอื่นๆ

ตารางที่ 2.1 แสดงการแบ่งแยกโดเมน

3. โดเมนที่มีตัวอักษร 2 ตัว เรียกว่า **country domain** ซึ่งเป็นตัวอักษรใช้แทนชื่อประเทศต่างๆ

หน้าที่หลักของโดเมนเนม คือ การกระจายความรับผิดชอบภายใน DNS ซึ่งจะมีหน่วยงานที่เรียกว่า NIC (Network Information Center) เป็นผู้กระจายความรับผิดชอบจากระดับ Top-Level Domain ไปสู่ระดับอื่นๆ ในแต่ละพื้นที่ในประเทศไทยมี หน่วยงาน THNIC เป็นผู้ดูแล Top-Level Domain ของประเทศไทย และผู้ที่ดูแลโดเมน nectec.or.th ก็คือ NECTEC จะเห็นว่า THNIC ได้กระจายความรับผิดชอบโดเมนนี้มาให้กับ NECTEC เป็นผู้ดูแล

2.2.2. Name Server

ผู้รับผิดชอบพื้นที่ของตัวเองก็จะต้องจัดหา Name Server ซึ่งเป็นเครื่องคอมพิวเตอร์ที่มีซอฟต์แวร์เพื่อทำหน้าที่เก็บฐานข้อมูลสำหรับการแปลงชื่อและหมายเลขประจำเครื่อง

Name Server แบ่งออกเป็น 3 ประเภทคือ

- Primary Name Server
- Secondary Name Server
- Cache Name Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Primary และ Secondary Name Server

Primary และ Secondary Name Server จะมีหน้าที่ในการเก็บฐานข้อมูล โดเมนเนม ส่วนที่รับผิดชอบ ผู้ดูแลระบบจะต้องจัดหา Primary Name Server 1 เครื่องและ Secondary Name Server อย่างน้อย 1 เครื่องโดยที่เครื่อง Name Server ทั้งสองประเภทจะต้องเป็นคนละเครื่องกันเพื่อป้องกันปัญหาว่า หากเครื่อง Primary Name Serverไม่สามารถที่จะให้บริการได้เครื่อง Secondary Name Server ก็จะให้บริการในการแปลงข้อมูลแทนได้

ความแตกต่างระหว่าง Primary และ Secondary Name Server คือเครื่องที่ทำหน้าที่เป็น Primary จะเก็บข้อมูลทั้งหมดสำหรับพื้นที่ที่ดูแลอยู่ ดังนั้นการแก้ไขต่างๆ จะทำที่ Primary Name Server เท่านั้น ส่วนเครื่องที่เป็น Secondary จะทำการดึงข้อมูลจากเครื่องที่เป็น Primary คือจะทำการโอนถ่ายข้อมูลจากเครื่อง Primary ซึ่งเรียกว่า Zone Transfer มายังเครื่องของตนเพื่อเก็บเป็นฐานข้อมูล ทุกๆ ช่วงเวลาหนึ่งๆ

เมื่อมีโฮส์ตใหม่ต้องการที่จะเพิ่มเข้าไปในพื้นที่ผู้ดูแลระบบจะต้องทำการเพิ่มข้อมูล คือชื่อโฮส์ตและหมายเลขประจำเครื่อง ของโฮส์ตนั้นลงไปยังฐานข้อมูลของ Primary Name Server จากนั้นเครื่อง Primary ก็จะทำการอ่านไฟล์ใหม่เรื่อยๆ และทำให้สามารถทำการตอบคำถามจากผู้ใช้ว่าโฮส์ตดังกล่าวมีหมายเลขประจำเครื่อง อะไร และถ้าเครื่อง Primary มีข้อมูลใหม่เกิดขึ้นจะทำให้เครื่อง Secondary มีข้อมูลใหม่ไปด้วย

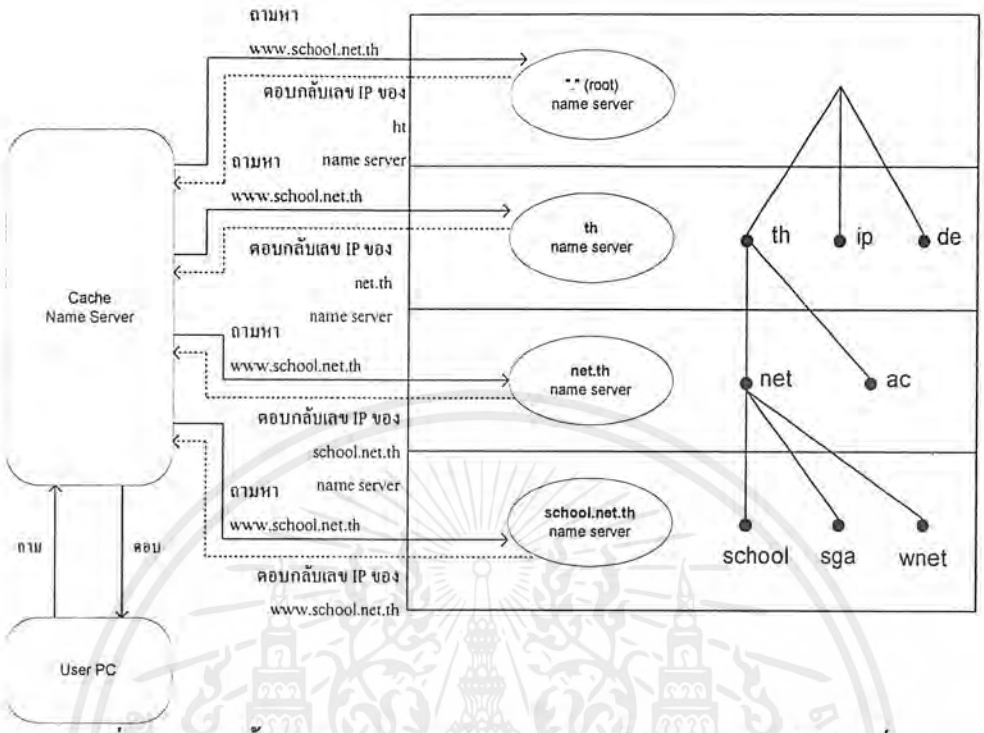
Cache Name Server

Cache Name Server จะทำหน้าที่ในการรับคำสั่งจากเครื่องคอมพิวเตอร์ของผู้ใช้ เช่นขอทราบ IP จากชื่อ หรือขอทราบชื่อจาก IP แล้วไปค้นหามาให้ Cache Name Serverจะเป็น DNS Server ที่เราตั้งในเครื่องพีซีนั้นเองการที่ Cache Name Server หยุดการทำงานจะทำให้ผู้ใช้ไม่สามารถที่จะใช้งานอินเทอร์เน็ตไม่ได้นอกจากจะใช้หมายเลขประจำเครื่องแทนชื่อส่วน Primary และ secondaryจะรับการร้องขอจาก Cache Name Server อีกที เครื่อง Cache Name Server จะมีฐานข้อมูลสำหรับเก็บชื่อ Root Name Server เวลาที่ Cache Name Serverได้รับคำขอทราบหมายเลขประจำเครื่องจากชื่อมันจะไล่ถามผู้รับผิดชอบในแต่ละ พื้นที่เริ่มจาก Root Name Server ไปจนกว่าจะเจอตัวที่รับผิดชอบจริงๆ คำถามใดที่เคยออกไปค้นหาและตอบแล้วนั้นจะถูกเก็บไว้ในหน่วยความจำ เพื่อที่ว่าครั้งต่อไปจะได้ตอบได้ในทันทีโดยไม่ต้องไปค้นมาอีก จนกว่าจะถึงเวลาที่ต้อง Update ข้อมูลใหม่ ตามค่า Time to Live

ข้อดีของ Cache Name Server คือมีความสามารถในการทำ caching

จากรูปที่ 2.2 เมื่อมีผู้ต้องการทราบหมายเลขประจำเครื่องของ www.school.net.th เครื่องจะทำการส่งคำถามมายัง Cache Name Server ซึ่ง Cache Name Server จะส่งคำถามนี้ต่อไปยัง Root Name Server จากนั้น Root Name Server จะตอบกลับมาว่าผู้รับผิดชอบพื้นที่ของ th อยู่คือใคร พร้อมทั้งส่งชื่อและหมายเลขประจำเครื่อง ของ ns.thnic.net ซึ่งเป็น Primary Name Server ของพื้นที่ th นั้นกลับมา แล้ว Name Server ของเราจะส่งคำถามนี้ไปยัง ns.thnic.net แล้วก็ส่งต่อไปเรื่อยๆ ดังแสดงในรูปจนกระทั่งทราบว่าผู้ดูแลพื้นที่ school.net.th คือ ns1.nectec.or.th จากนั้นก็ส่งคำถามไปยัง ns1.nectec.og.th แล้วจะได้คำตอบออกมาว่าหมายเลขประจำเครื่องของ www.school.net.th คือ 202.44.204.80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 2. แสดงขั้นตอนการทำงานของ DNS ในการค้นหาหมายเลขประจำเครื่อง

2.2.3. Time to Live

Name Server ไม่สามารถที่จะทำการเก็บข้อมูลที่ไปค้นหามาได้ตลอดเวลาเนื่องจากผู้ดูแลระบบจะมีการเปลี่ยนแปลงข้อมูลอยู่เสมอดังนั้นจึงต้องมี Time to Live หรือ TTL สำหรับข้อมูลของตน และ Name Server จะต้องเก็บเวลานั้นไว้ด้วย และเมื่อถึงเวลาดังกล่าวก็แสดงว่าข้อมูลที่ได้มาหมดอายุ Name Server จะต้องยกเลิกข้อมูลเดิมแล้วไปนำข้อมูลใหม่จากผู้รับผิดชอบในพื้นที่นั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3. Apache Web Server

2.3.1. Web Server

เว็บเซิร์ฟเวอร์เป็นเซิร์ฟเวอร์โปรแกรมประเภทหนึ่งที่ทำงานอยู่เบื้องหลังของบริการ World Wide Web โปรแกรมดังกล่าวจะรอรับสัญญาณการร้องขอจากเครื่องลูกข่าย เช่น โปรแกรมบราวเซอร์ซึ่งได้แก่ โปรแกรมเน็ตสเคป หรือไม่ก็เป็นโปรแกรม Internet Explorer เมื่อโปรแกรมเว็บเซิร์ฟเวอร์ ได้รับสัญญาณร้องขอเรียบร้อย โปรแกรมจะทำการจัดส่งข้อมูลบางประเภทไปยังเครื่องลูกข่ายที่ได้ส่งสัญญาณร้องขอเข้ามา ข้อมูลดังกล่าวอาจจะเป็นข้อมูลประเภทเท็กซ์ หรือว่าจะเป็นข้อมูลประเภทไฟล์กราฟิก เมื่อโปรแกรมเว็บบราวเซอร์ได้รับข้อมูลก็จะเรียบเรียงข้อมูลที่ได้รับมาแล้วค่อยนำเสนอให้กับผู้ใช้งานต่อไป ในเชิงคอนเซ็ปต์แล้วโปรแกรมเว็บเซิร์ฟเวอร์เป็นโปรแกรมที่มีลักษณะการทำงานที่ง่ายไม่ซับซ้อนอะไร คือ รอรับสัญญาณร้องขอ แล้วทำการส่งกลับข้อมูลที่มีการร้องขอเข้าเข้ามาไปยังเครื่องลูกข่ายอย่างถูกต้อง

โปรแกรมเว็บเซิร์ฟเวอร์จะทำการติดต่อกับเครื่องลูกข่ายต่างๆหรือติดต่อกับโปรแกรมเว็บบราวเซอร์ด้วยโพรโตคอลที่มีชื่อว่า Hypertext Transfer Protocol (HTTP) ซึ่งโพรโตคอล ดังกล่าวเป็นโพรโตคอลที่มีลักษณะโครงสร้างที่ง่าย และมีความเป็นมาตราฐานในตัวเอง จึงตัดปัญหาความการทำงานที่เข้ากันในของแต่ละบริษัทผู้ผลิตฮาร์ดแวร์

เอกสารส่วนมากที่ใช้ในการติดต่อสื่อสารกัน ส่วนมากจะอยู่ในรูปแบบของภาษา HTML (Hypertext Markup Language) ซึ่งภาษาดังกล่าวเป็นส่วนย่อยของภาษาหนึ่ง ซึ่งมีชื่อเรียกว่า SGML (Standard General Markup Language) ซึ่งถูกใช้งานส่วนอย่างแพร่หลายในองค์กรต่างๆของสหรัฐ

HTML เป็นเอกสารรูปแบบหลักที่ใช้งานในเว็บ HTML มีลักษณะเด่นหลายประการคือ ง่ายในการจัดเรียงและ ทำการสร้างรูปแบบของเอกสารในลักษณะเป็นเท็กซ์โปรแกรมบราวเซอร์จะทำการแปลงเอกสารในรูปแบบ HTML ก่อนแล้วจึงทำการเรียบเรียงอีกทีหนึ่ง ลักษณะเด่นอีกประการหนึ่งของ HTML คือ อนุญาตให้มีการทำลิงก์เชื่อมต่อกันระหว่างเอกสารต่างๆได้

HTML อนุญาตให้ผู้ใช้งานสามารถอ้างอิงถึงเอกสารอื่นที่เก็บไว้ในเครื่องคอมพิวเตอร์อื่นได้และ HTML อนุญาตให้เอกสารหรือข้อมูลมีลักษณะดูคล้ายเป็น 3 มิติได้ คือ เราไม่จำเป็นต้องอ่านแบบต่อเนื่องกันก็ได้ เราสามารถที่จะอ่านแบบกระโดดไปกระโดดมาก็ได้

2.3.2. Apache Web Server

เป็นโปรแกรมประเภทเว็บเซิร์ฟเวอร์ประเภทหนึ่ง ซึ่งมีลักษณะเด่นหลายประการกล่าวคือ

1. เป็นโปรแกรมที่แจกให้ใช้ฟรีบนอินเทอร์เน็ต
2. สามารถทำการติดตั้งได้ง่าย
3. สามารถทำการกำหนดค่าเริ่มต้นการใช้งานง่าย

จากข้อดีที่ได้กล่าวมาข้างต้นทำให้ Apache Server เป็นโปรแกรมเว็บเซิร์ฟเวอร์ที่เป็นที่นิยมใช้งานกันมาในปัจจุบันซึ่งจากตัวเลขของ Netcraft (www.netcraft.co.uk/survey/report) ได้สำรวจตัวเลขออกมาได้ค่าตัวเลขการใช้งานโปรแกรมดังกล่าวคิดเป็นร้อยละ 35.68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการแยกตัวออกมาสร้างกลุ่มเพื่อพัฒนาโปรแกรมอย่างอิสระของ Rob McCool ซึ่งเป็นหนึ่งในผู้ร่วมพัฒนาโปรแกรมของ NCSA (National Center for Supercomputing Applications) หลังจากการแยกตัวออกมาได้มารวมกลุ่มกับนักพัฒนาโปรแกรมอิสระ ในยุกต์แรก ๆ กลุ่มของนักพัฒนามีอยู่ด้วยกัน 8 คนหลัก ๆ คือ

- Brian Behlendorf
- Roy T. Fielding
- Rob Hartill
- David Robinson
- Cliff Skolnick
- Randy Terbush
- Robert S. Thau
- Andrew Wilson

ต่อมามีเข้ามาร่วมทีมอีกสามคน ได้แก่ Eric Hagberg, Frank Peters, Nicolas Pioch ในการพัฒนาใช้โปรแกรม NCSA httpd เวอร์ชั่น 1.3 เป็นพื้นฐานในการพัฒนาต่อ และได้ออกเวอร์ชั่นแรกในปี 1995 เป็นเวอร์ชั่น 0.6.2 และนับจากนั้นโปรแกรม Apache Web Server ได้ความนิยมเป็นอย่างสูง จึงได้มีการพัฒนาต่อมาจนถึงเวอร์ชั่นปัจจุบัน คือ 1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4. Squid Proxy Server

ในบทนี้จะกล่าวถึงภาพรวมของความเป็นมาในการพัฒนา พรอกซี่เซิร์ฟเวอร์จากนั้นจะตามด้วยรายละเอียดซึ่งจะลงลึกในเรื่องของค่าพิเศษเฉพาะบางค่าของพรอกซี่ซึ่งปกติแล้ว พรอกซี่เซิร์ฟเวอร์นั้นจะมีอยู่หลายชนิดมากซึ่งแต่ละแบบก็มีจุดประสงค์สำหรับใช้กับงานเฉพาะด้านนั้นๆ ในเครื่อง firewall proxy server ที่ใช้กันโดยทั่วไปนั้นถูกออกแบบให้มีค่า throughput ที่สูง(ซึ่งได้จากการทำ caching) และมีความสามารถทางด้าน firewall อีกด้วย ส่วน Department firewall นั้นจะเป็นพรอกซี่ขนาดเล็กลงมาซึ่งมักมีฟังก์ชันการทำงานต่างๆ เหมือนกับ firewall proxy และในตอนท้ายของบทนี้จะพูดถึง พรอกซี่เซิร์ฟเวอร์ที่ออกแบบมาสำหรับงานเฉพาะงาน

ทั้งหมดในบทนี้จะช่วยให้เข้าใจได้มากยิ่งขึ้นถึงพรอกซี่ต่างๆ ที่มีใช้อยู่ในปัจจุบัน คำว่า "พรอกซี่" จะใช้ในการรวมเอางานทั้งหมดที่สามารถทำงานด้วยกันได้เข้าด้วยกัน และในสายตาของพรอกซี่เซิร์ฟเวอร์แล้วก็คือจะทำงานทางด้านการการร้องขอต่างๆ ในฐานะตัวแทนของไคลเอ็นต์ซึ่งรวมทั้งการทำ filtering และ monitoring และฟังก์ชันทางด้านประสิทธิภาพอื่นๆ ด้วย

2.4.1. ประวัติความเป็นมาของ Proxy Server

ในปี 1990 นั้น proxy servers นั้นเดิมทีถูกมองว่าเป็น gateways ซึ่ง WWW gateway ตัวแรกนั้นถูกเขียนขึ้นโดยทีม WWW ที่ CERN ซึ่งนำโดย Tim Berners-Lee ซึ่งเขาเป็นคนที่สร้าง World Wide Web ขึ้นมาด้วย

คำว่า gateway นั้นมักใช้เรียกอุปกรณ์ที่ทำหน้าที่ forward แพกเก็ตระหว่างเครือข่าย และในบางครั้งจะต้องทำหน้าที่ในการแปลงโพรโตคอลหนึ่งให้เป็นอีกโพรโตคอลหนึ่งอีกด้วย ในปี 1993 ได้ใช้คำว่า "Web proxy servers" แทนอุปกรณ์ที่ทำหน้าที่เป็น Web gateway สาเหตุที่ทำดังกล่าวเนื่องจากต้องการให้เกิดความแตกต่างกันระหว่างคำว่า "Internet/Firewall gateways" และคำว่า "Information gateway" โดยที่คำว่า Internet/firewall gateways หรือพรอกซี่นี้จะเป็นอุปกรณ์ที่ใช้ในการควบคุมให้ การสื่อสารต่างๆที่เกี่ยวกับ Web นั้นสามารถผ่านเข้าไปใน secured intranets ได้ ส่วนคำว่า information gateways นั้นจะหมายถึงอุปกรณ์ที่ทำหน้าที่อินเทอร์เฟสข้อมูลอื่นๆ เข้ากับเว็บ

Internet/Firewall gateways นั้นเคยถูกเรียกว่าเป็น "พรอกซี่เซิร์ฟเวอร์" เพราะเหตุที่ว่ามันทำงานในลักษณะของ "on behalf of the client" หรือทำงานเป็นตัวแทนให้กับ ไคลเอ็นต์ แต่สำหรับ information gateway จะตรงกันข้ามนั่นคือมันจะทำงานในลักษณะของ "on behalf of the server" หรือทำงานเป็นตัวแทนให้กับเซิร์ฟเวอร์ ดังนั้นบางครั้งมันจึงถูกเรียกว่า reverse proxy อย่างไรก็ตามในบทความนี้ก็จะยังคงใช้คำว่า gateway อยู่ ในปัจจุบันแล้ว information gateway ส่วนใหญ่จะถูกนำไปใช้งานให้อยู่ในรูปของโปรแกรมประยุกต์ CGI หรือเว็บเซิร์ฟเวอร์ API มากกว่าที่จะถูกนำไปใช้งานในรูปของเซิร์ฟเวอร์เดี่ยว (standalone server) ซึ่งเขียนขึ้นมาเพื่อทำงานนั้นๆ โดยเฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2. CERN Proxy Server

พรอกซี่เซิร์ฟเวอร์ที่ชื่อ CERN (CERN httpd)นั้นเป็น Internet/firewall พรอกซี่เซิร์ฟเวอร์ตัวแรก โดยที่จริงๆ แล้วนั้นมันเป็นลูกผสมระหว่าง เว็บเซิร์ฟเวอร์และพรอกซี่เซิร์ฟเวอร์และผู้ที่เขียนบทความฉบับนี้เป็นหัวหน้าในการกำหนดสถาปัตยกรรมในการพัฒนา CERN server และเป็นผู้เขียน code โดยส่วนใหญ่ของ พรอกซี่เซิร์ฟเวอร์และส่วนของการทำ caching

ตัวผู้เขียนบทความเองและ Kevin Altis ทำงานด้วยกันมามากในการเผยแพร่ พรอกซี่เซิร์ฟเวอร์ให้กับคนใช้เว็บต่างๆ หลังจากฤดูใบไม้ผลิของปี 1994 เป็นต้นมาก็ได้รับความนิยมใช้งานเพิ่มขึ้นเป็นอย่างมาก

ผู้เขียนได้ร่วมมือกับบริษัท Netscape Communication ในการเขียนซอฟท์แวร์ Netscape Proxy Server ตัวแรกขึ้น และนำออกเผยแพร่เมื่อปี 1995 หลังจากนั้นก็มีพรอกซี่เซิร์ฟเวอร์ต่างๆเกิดขึ้นมากมายจากหลายหน่วยงานและหลายบริษัทด้วยกัน ตัวอย่างเช่น W3C ต่อมาการพัฒนา CERN server นั้นได้ถูกเปลี่ยนมาเป็น W3C ดังนั้นปัจจุบันจึงเรียก W3C httpd แทน CERN httpd

2.4.3. คุณสมบัติโดยทั่วไปของ Proxy Server

คุณสมบัติโดยทั่วไปของ พรอกซี่เซิร์ฟเวอร์มีดังนี้

- Transparancy : การใช้งานพรอกซี่หรือไม่ใช้งานพรอกซี่นั้นผลลัพธ์ที่ได้จะต้องเหมือนกัน ในแง่ของสิ่งที่ตอบสนองกลับมาจะต้องเหมือนกันทุกประการไม่ว่าการเชื่อมต่อนั้นจะทำโดยตรงหรือผ่านพรอกซี่เซิร์ฟเวอร์ก็ตาม
- ไคลเอ็นต์สามารถตัดสินใจได้ว่าจะใช้พรอกซี่หรือไม่
- Destination server จะไม่ได้รับผลกระทบจากพรอกซี่เซิร์ฟเวอร์ที่นำมาคั่นกลางแต่อย่างใด และ Destination server จะไม่รู้ว่ามีการใช้ พรอกซี่เซิร์ฟเวอร์ด้วยซ้ำไป

สิ่งที่จะพูดต่อไปนี้จะเป็นรายละเอียดที่มีผลต่อคุณสมบัติของพรอกซี่เซิร์ฟเวอร์และจะยกตัวอย่างขึ้นมาอธิบาย สิ่งที่จะพูดถึงอีกอย่างก็คือมีพรอกซี่เซิร์ฟเวอร์หลายตัวด้วยกันที่ไม่จำเป็นต้องมีคุณสมบัติตรงตามที่กล่าวมาแล้วในข้างต้น สาเหตุก็อาจเนื่องมาจากต้องการให้มีความสามารถบางประการซึ่งมีประโยชน์มากกว่าคุณสมบัติที่กล่าวมาแล้วก็ได้ การทำ filtering นี้จะมีผลกระทบต่อคุณสมบัติ transparancy ค่า status code ค่า 305 Use proxy ใน HTTP จะอนุญาตให้ตัวเซิร์ฟเวอร์สามารถระบุได้ว่าไคลเอ็นต์ ควรใช้พรอกซี่และ destination server สามารถที่จะดูการร้องขอได้เพื่อที่จะทำการตัดสินใจว่าการร้องขอที่มีเข้ามานั้นผ่าน พรอกซี่เซิร์ฟเวอร์มาหรือไม่และทำการส่งข้อมูลที่เหมาะสมกลับไป

Transparancy

คำว่า transparancy นี้ในเรื่องของพรอกซี่เซิร์ฟเวอร์จะหมายถึงผู้ใช้งานไม่จำเป็นต้องรู้เลยว่ามีการใช้ พรอกซี่เซิร์ฟเวอร์อยู่ ผลลัพธ์ที่ผู้ใช้งานมองเห็นนั้นจะเป็นการทำงานร่วมกันระหว่าง origin server และพรอกซี่เซิร์ฟเวอร์ ถ้าพรอกซี่เซิร์ฟเวอร์ไม่ไปรบกวนการส่งข้อมูลแล้วจะหมายความว่าผลลัพธ์ที่ได้จากการใช้หรือไม่ใช้พรอกซี่จะไม่มีความแตกต่างกันเลยอย่างแน่นอน บ่อยครั้งที่การทำ caching นี้จะเพิ่มประสิทธิภาพขึ้นจนผู้ใช้รู้สึกได้และ การตอบสนองจะได้รับเร็วขึ้นโดยใช้พรอกซี่ในทางกลับกัน หาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พรอกซี่ ทำการควบคุมการเข้าถึงและทำการ filtering แล้ว ผู้ใช้งานจะสามารถรู้ได้ว่าพรอกซี่ ได้เข้ามามีผลกระทบต่อการทำงานแล้ว นอกจากนี้แล้วข้อความแสดงความผิดพลาด (error message) ของปัญหาที่เกิดขึ้นเมื่อติดต่อกับ remote server จะถูกสร้างขึ้นโดย พรอกซี่เซิร์ฟเวอร์และข้อความแสดงความผิดพลาดอาจบอกด้วยว่าข้อความมาจากพรอกซี่เซิร์ฟเวอร์

ในอดีตนั้นมีพรอกซี่เซิร์ฟเวอร์แบบ non-transparent ด้วยเหมือนกัน ซึ่งพรอกซี่เซิร์ฟเวอร์จะถูกเข้าถึงโดยตรงจากการใช้งาน client program และ URL ที่ร้องขอจะถูกเพิ่มเติมเข้าไปยังส่วนสุดท้ายของ URL ของพรอกซี่เซิร์ฟเวอร์จากนั้น พรอกซี่เซิร์ฟเวอร์จะทำการรับข้อมูลและเปลี่ยนแปลงการอ้างถึง URL นั้นให้ชี้กลับไปยังพรอกซี่โดยการเพิ่มเติมค่า URL เข้าไปจริงๆ หรือพูดอีกอย่างก็คือ URL ที่ผู้ใช้งานมองเห็นนั้นจะต่างกัน

พรอกซี่เซิร์ฟเวอร์ที่มีการทำงานแบบ non-transparent ตัวหนึ่งได้แก่ Lagoon ซึ่งภายหลังได้เปลี่ยนเป็นแบบ transparent แล้ว ข้อสังเกตคือคำว่า transparent ในบทนี้จะหมายความถึงความจริงที่ว่าจะไม่มีการยุ่งกับ URL อีก โดยที่ พรอกซี่เซิร์ฟเวอร์แบบ non-transparent นี้จะทำการเปลี่ยน URL ใน HTML document และ HTTP redirection นั้นให้ชี้กลับไปยังพรอกซี่เซิร์ฟเวอร์

พรอกซี่เซิร์ฟเวอร์แบบ non-transparent นั้นเป็นขั้นแรกของเทคโนโลยีของพรอกซี่โดยมันไม่ต้องการการเปลี่ยนแปลงซอฟท์แวร์ที่ไคลเอ็นต์แต่อย่างใดและสามารถทำงานได้กับไคลเอ็นต์ที่มีอยู่ในปัจจุบัน ต่อมาเมื่อไคลเอ็นต์ ได้รับการทำให้สามารถรู้จักกับพรอกซี่ได้แล้ว เป็นเหตุทำให้ non-transparent นั้นค่อยๆ หมดไปในที่สุด

Use Client-Controlled

แม้ว่าพรอกซี่ต่างๆ จะถูกมองไม่เห็นโดยผู้ใช้งานก็ตาม เราสามารถบอก client software ให้มันรู้ได้ว่ากำลังติดต่ออยู่กับ พรอกซี่เซิร์ฟเวอร์เพื่อที่จะบอกให้ซอฟท์แวร์รู้ว่าให้ต่อกับพรอกซี่แทนการติดต่อกับเว็บเซิร์ฟเวอร์โดยตรง

2.4.4. Origin Server Unaware of Proxy Server

เวบเซอร์ฟเวอร์จะไม่สามารถรู้ได้ถึงความแตกต่างระหว่างการติดต่อโดยตรงกับไคลเอ็นต์หรือติดต่อกับไคลเอ็นต์โดยผ่านพรอกซี่โดยที่โพรโตคอล HTTP จะจัดเตรียมข้อมูลเกี่ยวกับการมี intermediate พรอกซี่เซิร์ฟเวอร์ ไว้ให้ แต่บ่อยครั้งที่ข้อมูลดังกล่าวจะไม่ได้รับความสนใจจากตัวเว็บเซอร์ฟเวอร์เลย

Different type of Proxy servers

พรอกซี่เซิร์ฟเวอร์นั้นมีอยู่หลายชนิดด้วยกัน บางตัวเป็นพรอกซี่ธรรมดาที่มุ่งหมายสำหรับการทำ Web access และการทำ Caching ในขณะที่บางตัวอาจสร้างขึ้นมาเป็นพิเศษโดยเฉพาะกับโปรแกรมประยุกต์บางอย่าง รายชื่อที่จะกล่าวต่อไปนี้เป็น พรอกซี่เซิร์ฟเวอร์ในระดับ application level แบบต่างๆ

- generic firewall proxies
- proxy chaining
- departmental proxies
- personal proxies

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- specialized proxies
  - proxies between clients and other proxies
  - proxies doing format conversions
  - accelerators
- reverse proxies

Generic Firewall Proxy Server

เป็นพรอกซี่เซิร์ฟเวอร์แบบทั่วไปซึ่งสามารถจัดการเกี่ยวกับ Web traffic ซึ่งรวมทั้งโพรโตคอลแบบ HTTP, FTP และ Gopher และยังสนับสนุน Secure protocol โดยใช้ SSL ตัวอย่างเช่น SSL, HTTPS และ SNEWS ได้

> *สิ่งที่มักเข้าใจผิดกันบ่อยๆ ก็คือ Web proxy server สามารถใช้กับ FTP software ทั่วๆ ไปเพื่อเข้าไปยัง firewall ได้ ซึ่งความจริงไม่ใช่ Web client จะใช้โพรโตคอล HTTP ในการส่งการร้องขอไปให้ Web proxy server หรือแม้แต่การรับ FTP URLs แต่อย่างไรก็ตาม FTP software มักจะคุยกันด้วยโพรโตคอล FTP และไม่สามารถทำงานร่วมกับ Web proxy servers ได้ แม้ว่าจะมี FTP client ใหม่ๆ ที่เพิ่งออกมาจะสามารถทำงานกับ Web proxy server ได้ก็ตาม*

Generic Proxy server นั้นมีความสามารถมากมาย โดยมันสามารถทำการจัดการควบคุมการเข้าถึงต่างๆได้ ตลอดจนถึงการทำ filtering, logging และ การทำ caching

Firewall proxy servers ซึ่งชื่อก็บอกอยู่แล้วว่ามันจะรัน Firewall ใน DMZ โดยมันจะรับการร้องขอจากภายใน firewall และทำการ forward ออกมาและส่งออกไปยังอินเทอร์เน็ตและทำการผ่านค่าผลลัพธ์กลับมาสู่ ไคลเอ็นต์ ที่ทำการร้องขอ พรอกซี่ชนิดนี้มักจะใช้ความสามารถในการทำ caching อยู่บ่อยๆ ดังนั้นบางการร้องขอนั้นไม่จำเป็นต้อง forward ไปให้กับ origin server ทั้งหมดก็ได้แต่จะได้จากการนำข้อมูลในแคชมาใช้งาน

Traffic ทั้งหมดจากภายในและภายนอกอินเทอร์เนตจะผ่าน firewall proxy ซึ่งเป็นแค่จุดเดียวที่สามารถเข้าถึงอินเทอร์เนตได้ ข้อสังเกตคือในทางปฏิบัติแล้วอาจจะมี firewall proxy server หลายๆ ตัวทำงานขนานกันเนื่องจาก พรอกซี่เซิร์ฟเวอร์แค่ตัวเดียวอาจไม่เพียงพอต่อการให้บริการต่อการร้องขอที่มีเข้ามามากๆ ซึ่งต้องการการจัดการ Load balancing ที่ดีด้วย

Proxy Chaining

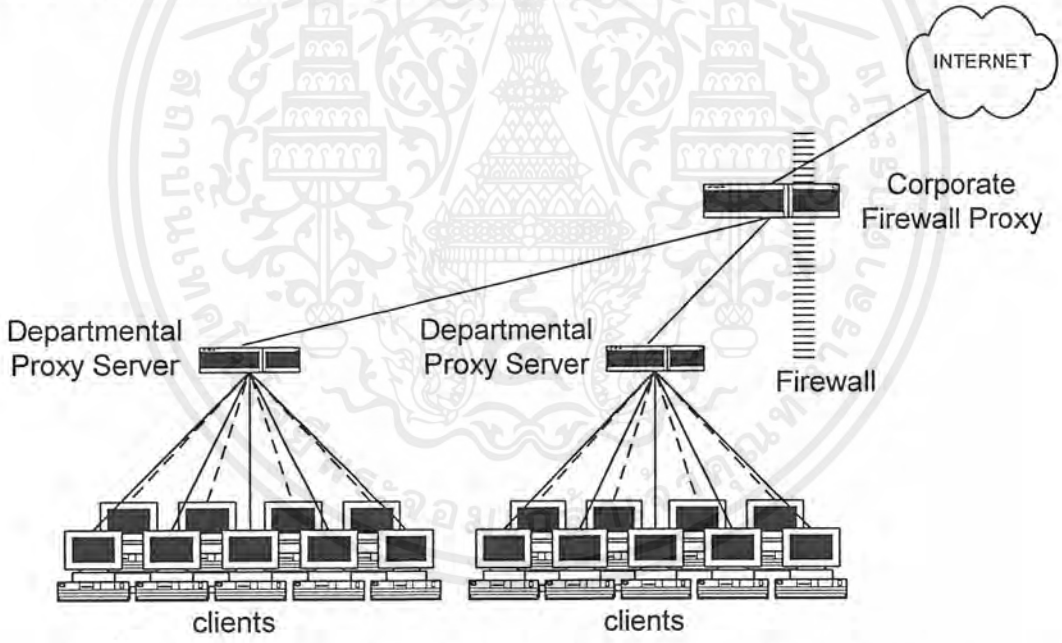
ไคลเอ็นต์ อาจจะทำการร้องขอข้อมูลโดยผ่าน departmental proxy server ซึ่งต่อแบบ daisy-chain เข้ากับ firewall proxy server การต่อแบบ daisy-chain นี้หมายถึงการ redirect ให้ departmental proxy server ทำการร้องขอผ่านไปยัง พรอกซี่เซิร์ฟเวอร์ตัวอื่น ซึ่งในกรณีนี้คือ firewall proxy นั่นเอง การทำ proxy chaining นี้จะอนุญาตให้ down stream proxy (ตัวที่อยู่ใกล้กับไคลเอ็นต์) ให้ได้รับประโยชน์จากแคชที่มีอยู่ใน upstream proxy server หาก firewall proxy ตัวหลักนั้นมีออบเจกต์สำหรับ departmental proxy อยู่แล้ว ตัว departmental proxy ตัวอื่นๆอาจจะสำเนาข้อมูลจากแคชของ firewall proxy ได้โดยไม่ต้องมีการทำการติดต่อไปยัง remote server ซึ่งอยู่ภายนอกอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ proxy chaining นี้จะช่วยลดภาระของโหลดที่ตกอยู่กับ main firewall proxy ได้ด้วยการมี departmental proxy เอาไว้ให้บริการการร้องขอบางอย่างโดยตรงจากแคชของมัน เฉพาะการร้องขอที่ departmental cache ไม่สามารถให้บริการได้เท่านั้นจึงจะถูกส่งไปยัง main firewall proxy และมีโอกาสความเป็นไปได้ที่จะพบข้อมูลในแคชก็มีมากเนื่องจากจำนวนของผู้ใช้งานที่ร้องขอเข้ามานั้นก็มักจะมีมากอยู่แล้ว

Departmental Proxy Server

เป็นพรอกซี่เซิร์ฟเวอร์ทั่วไปซึ่งคล้ายกับ firewall proxy แต่ฐานของผู้ใช้จะแคบกว่าเช่นจำกัดอยู่เฉพาะในแผนกของบริษัทหรือองค์กรขนาดใหญ่เท่านั้น proxy server software ที่ติดตั้งในระดับ departmental level นั้นอาจเป็นซอฟท์แวร์ตัวเดียวกับซอฟท์แวร์ที่ใช้ใน firewall แต่ต่างกันตรงการกำหนดค่าคอนฟิกกูเรชั่นก็ได้ ตัวอย่างเช่น ในบางแผนกอาจจะมีสิทธิในการเข้าถึงการควบคุมที่จำกัดกว่าแผนกอื่นและการควบคุมการเข้าถึงที่อยู่บน firewall proxy นั้นอาจจะเปลี่ยนแปลงไปตามแต่ละ departmental proxy ก็ได้ การต่อ Departmental proxy server เข้าไปยัง firewall proxy server นั้นอาจทำในรูปแบบของ daisy-chained ทำให้เกิดเลเยอร์ของ proxy ขึ้นมา 2 เลยอร์ดังรูปที่ 2.3 ข้อสังเกตคือ departmental proxy server นั้นเป็นเพียงตัวอย่างหนึ่งของการทำ multilevel proxy เท่านั้น



รูปที่ 2.3 Departmental proxy server daisy-chained to a corporate firewall proxy server

Personal Proxy Server

เป็น พรอกซี่เซิร์ฟเวอร์ที่ได้รับการปรับแต่งให้เหมาะกับการใช้งานจากผู้ใช้งานแค่คนเดียว ซึ่งมักจะรันอยู่บนโฮสต์เดียวกันกับ client program ข้อแตกต่างระหว่างความสามารถของ client software และ personal proxy นั้นยังคงเป็นที่สับสนกันอยู่ ดังนั้นจึงมีหลายคนที่ทำการโต้แย้งว่า personal proxy นั้นได้ถูกรวมเอาไว้ใน client software เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถที่มีให้ของ personal proxy ก็คือการทำ local caching, การทำ active cache update, การตัดสินใจว่าควรจะอัพเดทข้อมูลหรือไม่, การทำการจัดการเกี่ยวกับ hot list ต่างๆ และการทำ local search เป็นต้น

Specialized Proxy Server

เป็นพรอกซี่ที่มีหลากหลายชนิดมากโดยมันจะทำงานที่เฉพาะเจาะจงบางอย่างที่เหมาะสมกับสภาพแวดล้อมเป้าหมายของมัน ตัวอย่างที่ดีตัวหนึ่งก็คือ พรอกซี่เซิร์ฟเวอร์ที่ให้บริการแก่ client software ที่เป็นอุปกรณ์ประเภท palmtop ซึ่งพรอกซี่แบบนี้จะทำการลดคุณภาพของภาพและจำนวนของสีในภาพลงและทำการแปลงรูปให้อยู่ในฟอร์แมตที่ตัว palmtop สามารถเข้าใจได้ ซึ่งการทำดังกล่าวก็เพื่อลดความต้องการแบนด์วิดธ์ซึ่งมีจำกัดในอุปกรณ์ประเภท palmtop ลงและในขณะเดียวกันฟอร์แมตของข้อมูลดังกล่าวยังเหมาะสมสำหรับฮาร์ดแวร์และซอฟท์แวร์ของ palmtop อีกด้วย

- **ความแตกต่างระหว่าง Proxy กับ ไคลเอ็นต์**

Specialized Proxy Server แบบอื่นที่น่าสนใจก็คือแบบที่ติดตั้งอยู่หน้า พรอกซี่เซิร์ฟเวอร์จริงๆ ซึ่งพรอกซี่นี้จะทำการฟอร์เวิร์ด traffic ทั้งหมดที่มันรับเข้ามาไปให้กับพอร์ตต่างๆที่อยู่บนโฮสต์เดียวกัน (หรือต่างโฮสต์ก็ได้) ซึ่งรันตัวพรอกซี่เซิร์ฟเวอร์จริงๆ อยู่ พรอกซี่เซิร์ฟเวอร์แบบ specialized นี้มักจะใช้ในการทำงานเฉพาะเจาะจงอย่างใดอย่างหนึ่ง เช่นทำการฟิลเตอร์ Java applet หรือการทำ virus screening

- **Accelerators**

จะคล้ายกับ specialized proxy server ที่พูดถึงข้างต้นโดยจะติดตั้งอยู่หน้า actual server ซึ่ง actual server นี้อาจจะเป็น พรอกซี่เซิร์ฟเวอร์ก็ได้แต่มักจะเป็นตัว origin Web server จุดประสงค์ของ special accelerator proxy นี้ก็คือเพื่อให้สามารถทำ caching และ fast I/O ได้อย่างมีประสิทธิภาพ ซึ่งการการร้องขอนี้จะได้รับการบริการบ่อยๆ จากแคชของ accelerator proxy server ได้แทนที่จะต้อง forward ไปให้กับ origin server ซึ่งช้ากว่า

accelerator proxy servers นั้นเคยถูกใช้ช่วยแก้ปัญหาของความล่าช้าของ Web server software ได้ แต่ใน server software ตัวที่เร็วๆในปัจจุบันนั้นจะไม่ได้รับผลดีของ accelerator แบบนี้อีก และอาจจะทำให้เซอร์ฟเวอร์ช้าลงอีกด้วยอันเนื่องมาจากมีเลเยอร์เพิ่มขึ้นมาอีกชั้นหนึ่ง

Web server software บางตัวนั้นจะรวมเอาการทำงานของ accelerator proxy เอาไว้ภายในด้วย โดยการทำ caching การตอบรับครั้งก่อนที่เพิ่งส่งออกไปเอาไว้และนำมาใช้ใหม่แทนที่จะทำงานแบบเดิมกับการร้องขอตัวใหม่ที่เพิ่งเข้ามาแต่เหมือนกันกับครั้งก่อน นอกจากนี้อาจมีการเพิ่มความสามารถบางอย่างเข้าไปใน accelerator type separate server ซึ่งตั้งอยู่หน้าตัว พรอกซี่เซิร์ฟเวอร์จริงๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

Reverse Proxy

การกำหนด "reverse proxying" อ้างถึงการติดตั้งเมื่อ พรอกซี่เซิร์ฟเวอร์มีการแสดงทำงานในไคลเอ็นต์ราวกับเป็นเว็บเซิร์ฟเวอร์ธรรมดานั้นคือ ไคลเอ็นต์จะทำการติดต่อโดยพิจารณาจุดมุ่งหมายไปที่เซิร์ฟเวอร์ต้นทางและไม่มีทางรู้การร้องขอ มันจะถูกเปลี่ยนโดยเซิร์ฟเวอร์อื่นต่อไปโดยผ่านทาง พรอกซี่เซิร์ฟเวอร์อื่นอีก ในเรื่องพรอกซี่เราแนะนำทางเลือกในการกำหนดสำหรับ พรอกซี่เซิร์ฟเวอร์นั้นคือ gatways ในบทนี่เราจะใช้การกำหนด "reverse proxying" เมื่อ getways ถูกสนับสนุนโดยวิธิทางของ proxy server software ทั่วๆ ไป บางแห่งเขาจะใช้ข้อมูลของ getways ที่มีการสนับสนุนมาแต่เดิมในการติดต่อกับเซิร์ฟเวอร์โดยใช้ Third-party protocol และในกรณีนี้การกำหนด "getways" ซึ่งอาจจะเหมาะสมกว่า อย่างไรก็ตาม "reverse proxy" ยังจะต้องดูข้อกำหนดที่มีความแน่นอนเสมอสำหรับพวกเขา

คำว่า "reverse" ใน "reverse proxy" จะหมายถึงการกลับหน้าที่ของพรอกซี่เซิร์ฟเวอร์ในการทำงานตามปรกติของ(forward)พรอกซี่ พรอกซี่เซิร์ฟเวอร์จะทำตัวเป็นพรอกซี่ให้กับการร้องขอของไคลเอ็นต์จะถูกสร้างขึ้นมาแทนไคลเอ็นต์โดยพรอกซี่เซิร์ฟเวอร์อย่างไรก็ตามในการทำงานของ reverse proxy ตัว reverse proxy server เองก็มีการทำงานเช่นเดียวกับพรอกซี่สำหรับเซิร์ฟเวอร์ซึ่งพรอกซี่เองก็จะให้บริการส่ง การร้องขอแทนเซิร์ฟเวอร์ระหว่างนี้จะมองหาบางสิ่งหมายความว่าในทางที่ต่างกัน 2 ทาง ความแตกต่างทีว่าจะชัดเจนขึ้นเมื่อมีการพิจารณาความสัมพันธ์ของ พรอกซี่เซิร์ฟเวอร์กับไคลเอ็นต์ของมันและเซิร์ฟเวอร์ต้นทาง

forward proxy server หรือการกำหนดจากพวกเขาจะให้การทำงานของไคลเอ็นต์ 1 ตัวหรือไคลเอ็นต์หลายตัวที่เหมือนกัน ภาพของไคลเอ็นต์ที่ปรากฏ พรอกซี่เซิร์ฟเวอร์จะเป็นตัวให้บริการตามความต้องการของไคลเอ็นต์และการร้องขอ ทั้งหมดจะถูกส่งไปยัง พรอกซี่เซิร์ฟเวอร์กำหนดให้ไคลเอ็นต์จะใช้พรอกซี่เซิร์ฟเวอร์บางตัวเกินเวลาที่กำหนดไว้และการ configuration proxy จะขึ้นกับไซต์ที่ไคลเอ็นต์ทำงานอยู่ forward proxy server มักจะทำงานในองค์กรของตัวมันเองหรือการจัดหาบริการของอินเทอร์เน็ตตัว forward proxy server จะทำการปิดไคลเอ็นต์ ได้อย่างถูกต้อง

Reverse proxy server ในส่วนช่วยเหลืออื่นๆ จะอ้างได้จากเซิร์ฟเวอร์ต้นทางเพียงหนึ่งหรือจำนวนน้อย การสุ่มเซิร์ฟเวอร์ไม่สามารถที่จะเข้าถึงโดยผ่าน reverse proxy server ได้เช่นการกำหนดสภาพ ของไฟล์เพียงอย่างเดียวโดยหาได้จากเซิร์ฟเวอร์ต้นทาง นั้นคือ reverse proxy เป็นพรอกซี่เพราะการหาได้มาจาก reverse proxy server ซึ่ง reverse proxy server เองเป็นการออกแบบ พรอกซี่เซิร์ฟเวอร์แก่เซิร์ฟเวอร์เหล่านั้น และมันจะถูกใช้โดยไคลเอ็นต์ทั้งหมดในการเข้าถึงยังลักษณะของไซต์ที่มันให้การบริการ reverse proxy server มักจะทำงานโดยองค์การบ้างองค์การนั้นคือจะทำงานบน server ต้นทางแล้วพรอกซี่คือ reverse proxy

ในบทสรุปไคลเอ็นต์จะมอง "forward proxy" เหมือนกับ พรอกซี่เซิร์ฟเวอร์และ "reverse proxy" เหมือนกับเซิร์ฟเวอร์ต้นทางตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
37055

User of Reverse Proxy Server

Reverse proxy server จะมีจุดมุงหมายที่สำคัญ 2 ประการคือ

- จำลองสิ่งที่บรรจุใน geographically dispersed areas
- จำลองสิ่งที่บรรจุสำหรับ load balancing

Replication for Content Distribution

Reverse proxy server สามารถใช้การสร้าง เซิร์ฟเวอร์ จำลองทั่วไปของ เซิร์ฟเวอร์ หลักเพียงตัวเดียวตามขอบเขตของการกระจายในทางภูมิศาสตร์ ตัวอย่างเช่นบริษัทที่มีสำนักงานย่อยอยู่ทั่วโลกกล่าวได้ว่าบริษัทจะมีเว็บเซิร์ฟเวอร์ หลักอยู่ภายในโดยลูกจ้างทั้งหมดจะใช้แทนข้อมูลภายในของบริษัท reverse proxy server จะถูกติดตั้งลงในสำนักงานย่อยลูกจ้างจะใช้งาน (reverse proxy)เซิร์ฟเวอร์ เมื่อต้องการไปยังเซิร์ฟเวอร์หลัก นั้นคือถ้าที่อยู่ของเว็บเซิร์ฟเวอร์หลักอยู่ที่

Htttp://www.mysite.com

เมื่อสำนักงานย่อยถูกจำลองสามารถที่จะเรียกได้ดังนี้

http://www-london.mysite.com/

http://www-paris.mysite.com/

http://www-tokyo.mysite.com/

ข้างในแต่ละอันจะมีการจำลองพรอกซี่เซิร์ฟเวอร์ คือการกำหนด config รับมาจาก สิ่งที่บรรจุในตัวมัน ตั้งแต่เซิร์ฟเวอร์ หลักอยู่ที่ http://www.mysite.com/

เป็นไปได้ที่เราจะ config local DNS ของแต่ละสาขาย่อยโดยจะตั้งใจให้ host-name http://www-mysite.com ให้ตรงกับรูปจำลองใน local network ในกรณีนี้ เซิร์ฟเวอร์ จริงจะมีชื่อที่ต่างกันใน local DNS และชื่อนั้นจะถูกใช้ในพรอกซี่เซิร์ฟเวอร์ configuration หรืออาจจะใช้ เลขหมายประจำเครื่อง ของเซิร์ฟเวอร์หลักโดยตรงเลย อีกทางหนึ่งก็คือ DNS srver จะใช้โดยพรอกซี่เซิร์ฟเวอร์ อาจจะมีความแตกต่างกันและจะกระจาย www.mysite .com ไปยัง เซิร์ฟเวอร์ หลักและไม่มีการจำลองใน local

ในการจำลองจะมีการทำอย่างอิสระมากเว็บในทุกวันนี้ จะไม่ถูกจำลองในความรู้สึกอีกต่อไปตัวอย่างเช่น Lotus notes เมื่อมันทำงานภายในของมันก็จะมีการแก้ไขอาจจะโอนการจำลองออกไปและระบบจะ update การเปลี่ยนแปลงกับไปยัง database หลัก แม้ว่า reverse proxy server มันจะปิด caching อย่างไรก็ตามกลุ่มข้อมูลในส่วน reverse proxy จะมีข้อจำกัดไปยังเซิร์ฟเวอร์ หนึ่งหรือจำนวนน้อยๆ ซึ่งตรงข้ามกับข้อมูลที่ไม่มีขอบเขตนั้นเอง(web content ทั้งหมด)ในกรณีของ forward proxy ยิ่งไปกว่านั้น reverse proxy จะถูก configuration โดยดึงข้อมูลทั้งหมดที่ on-command อีกครั้งที่ระบุความแตกต่างหรือเมื่อมันเปลี่ยน no-command เหมือนเป็นส่วนของกฎ caching เราจะอ้าง no-command caching ในขณะที่ no-command caching อ้างไปยัง caching แบบง่ายๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Replication for Load Balancing

Reverse proxy server จะถูกใช้สำหรับ Load Balancing ของเว็บเซิร์ฟเวอร์ที่มีจำนวน Load มากๆ การร้องขอจากไคลเอ็นต์จะกระจายไปยัง เซิร์ฟเวอร์ หลายๆ ตัว โดยใช้ DNS round robin หรือจากวิธีอื่นๆ หนึ่งใน เซิร์ฟเวอร์ คือกฎของเว็บเซิร์ฟเวอร์ มันจะกระทำราวกับเป็น เซิร์ฟเวอร์หลัก content จะถูกแก้ไขบนเซิร์ฟเวอร์ หลักเพียงตัวเดียวเท่านั้นเซิร์ฟเวอร์อื่นๆ เป็น reverse proxy server โดยจะมี config ไปยังการแก้ไข content จากเซิร์ฟเวอร์หลัก reverse proxy จะบรรจุแคชและในไม่นานนี้ content ทั้งหมดคือการบริการโดยตรงจากแคชของพรอกซี่

> มันจะไม่เลือกใช้ forward proxy server อาจจะใช้เวลาที่ไปถึง cache hit rate ชนิดของแคชที่มี hit rates ที่ดีบน forward proxy server ที่สนับสนุน HTTP/1.0 จะผ่าน 30-60% มันสมควรกับข้อมูลที่ล้นบน web client จะร้องขอเอกสารใดจากเว็บที่ครบถ้วน พรอกซี่เซิร์ฟเวอร์จะมีข้อจำกัดจะมีขอบเขตในเรื่องของพื้นที่ว่างในแคช โดยจะเก็บข้อมูลได้ขนาดเล็กเพียงอย่างเดียวตลอดทั้งเว็บ
>
> ในการควบคุม reverse proxy server มันจะมีอำนาจในการปิด cache hit rate ได้ 100% ตั้งแต่มีการใช้การร้องขอไปยังเซิร์ฟเวอร์ต้นทางเพียงตัวเดียวหรือไม่กี่ตัว เราสามารถที่จะเก็บ URL ทั้งหมด และไม่มีการรบกวนในการทำงานภายนอกของแคช การร้องขอจะเข้ามาใน reverse proxy คดจะเป็นจำนวนเล็กๆ กลุ่มเล็กๆ และพรอกซี่ จะคล้ายกับเป็นเพชรในพื้นที่นี้

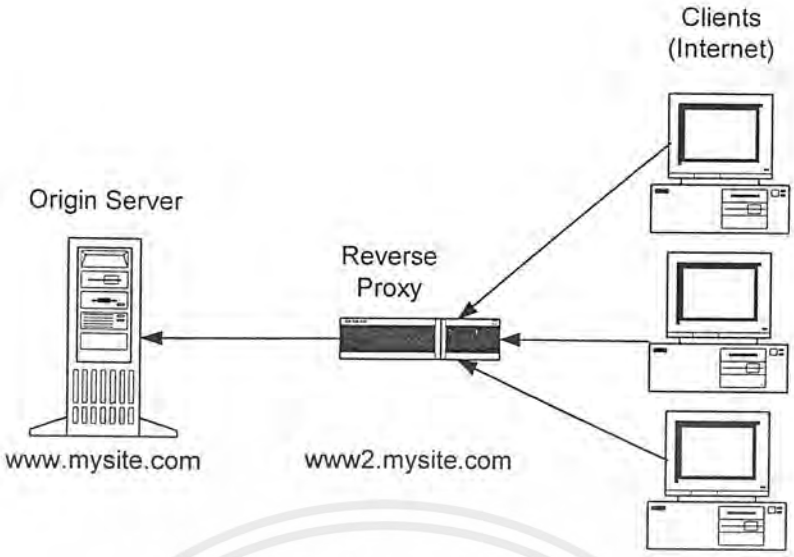
Components Of A Reverse Proxy Setup

การติดตั้ง reverse proxy ที่สมบูรณ์ จะประกอบไปด้วยส่วนประกบต่างๆ ซึ่งแต่ละส่วนจำเป็นมาก โดยส่วนหนึ่งจะมีผลกัยอีกส่วนหนึ่ง ในการเพิ่มความสะดวกในการ maintenance ของการจำลอง content จะมีส่วนประกอบหลักๆ ดังนี้

- Request URL remappings
- Request header remappings
- Response header remappings
- Content remappings
- Virtual multihosting

ส่วนย่อยต่อไปนี้จะช้อนแต่ละพื้นที่หลักและการเปลี่ยนแปลงการหาทางออกเมื่อประยุกษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ลักษณะของ Reverse Proxy

เราจะแสดง config ให้เห็นดังตัวอย่างรูป 2.4 reverse proxy server อยู่ที่

http://www2.mysite.com/

คือจะเป็นการ config ไปยังพรอกซี่แทน เซิร์ฟเวอร์ ต้นทางที่อยู่ที่

http://www.mysite.com/

reverse proxy server จะอยู่ที่ http://www2.mysite.com/ จะเป็นการประกาศที่อยู่และผู้ใช้สามารถที่จะเข้าไปข้างในมันได้โดยไม่ต้องรู้เซิร์ฟเวอร์ต้นทาง http://www.mysite.com/ เลย

เราจะใช้ Netscapt ทำนองเดียวกับการกำหนด reverse proxy ตัวอย่างการ config นั้นจะแตกต่างกันไปตามพื้นฐานของ proxy server software

Request URL Remapping

สิ่งที่ส่วนใหญ่จะทำเป็นอันดับแรกคือการให้ความสนใจกับการ config reverse proxy server คือมันจะต้องมีการทำการ map การร้องขอของ URLs ไปยัง URLs นั้นคือจุดมุ่งหมายของ เซิร์ฟเวอร์ จริงๆ

ในตัวอย่างเราต้องการการร้องขอทั้งหมดที่เข้ามาใน www2.mysite.com และจะส่งต่อไปให้เซิร์ฟเวอร์ฐานwww.mysite.com

ใน HTTP/1.0 เซิร์ฟเวอร์ จะเก็บแค่ port ของ URL ถ้าURL เริ่มเข้าถึงคือ

http://www2.mysite.com/dir/file.html

แล้วเซิร์ฟเวอร์จะรับเฉพาะพอร์ตอย่างเดียว /dir/file.html สำหรับ(forward)พรอกซี่เซิร์ฟเวอร์ มันจะมีความแตกต่างพวกเรามักจะได้รับ URL ทั้งหมด อย่างไรก็ตามคงจำกันได้ว่าในกรณีของ reverse พรอกซี่ตัวไคลเอ็นต์ จะคิดว่านั้นเป็นระเบียบของเว็บเซิร์ฟเวอร์ เพราะฉะนั้น reverse proxy server จะได้รับเพียงส่วนของพอร์ตอย่างเดียวเว็บเซิร์ฟเวอร์ ต้องการความเหมาะสม ดังนั้นในตัวอย่างจะเกิดตาม URL prefix mapping

/ => http://www.mysite.com/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์ของพรอกซี่เซิร์ฟเวอร์จะรองรับทั้ง mode ของ forward และ reverse proxy ในบางเวลา ในกรณีเหล่านั้น forward proxy จะใช้เมื่อมีการร้องขออย่างเจตนามาจาก reverse proxy server คำสั่งที่จะหลีกเลี่ยงการร้องขอที่สร้างมาเป็นพิเศษ ไม่จำเป็นทีจะต้องทำเป็นวงจาก forward ไปยัง reverse proxy (เช่นเดียวกับ เซิร์ฟเวอร์) เราจะเพิ่มการ mapping สำหรับ URL ที่มีขนาดเต็ม

http://www2/ => http://www.mysite.com/

มันจะอยู่ใน local domain เมื่อจะทำการเข้าถึงชื่อของ domain จะอยู่ซ้ายมือของเราในการ mapping ครั้งที่ 3 นี้จะเป็น

http://www2/ => http://www.mysite.com/

HTTP/1.1 จะระบุสถานะของ HTTP/1.1 เซิร์ฟเวอร์ทั้งหมดมันสามารถที่จะตอบรับการร้องขอทั้งหมดของ URL อย่างไรก็ตาม HTTP/1.1 ไคลเอ็นต์อาจจะไม่เคยใช้การร้องขอจนเต็ม URL ไปยัง (forward)พรอกซี่เซิร์ฟเวอร์ในการจัดหาการทำชื่อของ forward ที่เข้ากันได้หน้าตาของ version ของ HTTP จะเริ่มใช้ URL จนเต็มในการร้องขอทั้งหมดความเห็นในส่วนนี้กฎ 2 mapping จะจำเป็นใน HTTP/1.1 reverse proxy จะไม่มีการใช้รวมทั้งสอง mode ด้วยกัน(forward และ reverse)

Request Header Remappings

แน่นอนว่า headers ของการร้องขอ อาจจะบรรจุไปด้วยข้อมูลที่สร้างมาจากการสันนิฐานใน reverse proxy เซิร์ฟเวอร์คือ เซิร์ฟเวอร์ต้นทางหนึ่งใน header คือโฮสต์โดย header จะโอน hostname ไปใน URL นั้นคือการทำของการร้องขอ ความสำคัญของ header ก็คือการ miss ในส่วนของ URL จะเป็นการเติม “http://” ไว้ข้างหน้า

เซิร์ฟเวอร์ ต้นทางใช้ host header จะถูกกำหนดโดย DNS ในการใช้ชื่อปลอมที่จะติดต่อกับเซิร์ฟเวอร์ในทางนี้เว็บเซิร์ฟเวอร์ เดี่ยว host site ต่างๆ แต่ละตัวก็จะมีการปลอมของ DNS เป็นของตัวเอง จุดที่จะชี้ไปคือ เลขหมายประจำเครื่องและเว็บเซิร์ฟเวอร์ จะมองที่ Host header จะถูกตัดสินว่าจะถูกบรรจุไว้ในไซต์ ไหนโดการเลือกจากการให้บริการ

ในตัวอย่างของเรา reverse proxy server จะทำการเก็บตามนี้

Host: headers:

Host: www2.mysite.com

Host: www2

ถ้าเลขของ port ถูกระบุใน URLมันก็จะแสดงใน Host

Host: www2.mysite.com:80

Host: www2 :80

เห็นได้ชัดเลยว่า header ต้องการที่จะ remapped ให้เหมาะสมถูกต้องกับเซิร์ฟเวอร์ต้นทางยอมจะรับ Host header นั้นคือจุดมุ่งหมายของตัวมัน

Host: www.mysite.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Netscape Proxy Server 2.53 และการทำงานที่ต่ำกว่า Host หรือ Header จะถูก remapping โดยอัตโนมัติเหมือนกับผลของ URL remapping นั้นคือไม่แสดงออกมาโดยตรงโฮส์ตจะต้องมีการระบุกฎของ header remapping

host header จะมีประโยชน์ใกล้กับ virtual Multihosting ใน reverse proxy server

เมื่อส่วนหลักๆ ของ virtual Multihosting ในซอฟต์แวร์ของเว็บเซิร์ฟเวอร์ ไม่มีการให้สิทธิ software sever บางตัว จะใช้โฮส์ตอย่างนิ่งๆ header จะเป็นพื้นฐานในการสร้าง redirection แบบอัตโนมัติด้วยตัวเอง

redirection แบบอัตโนมัติจะทำเมื่อผู้ใช้มีการเข้าถึง URL เช่น

http://home .nestcape.com/people/ari

ที่ /people/ari บนเว็บเซิร์ฟเวอร์จะเกิดไดเรกทอรีขึ้นพร้อมกับไฟล์ index.html เซิร์ฟเวอร์จะส่ง redirection ในทุกๆ ครั้ง

http://home .nestcape.com/people/ari/index.html

http://home .nestcape.com/people/ari/

เป็นการบำรุงที่สอดคล้องกับ hypertext link พูดได้ว่าไฟล์ index.html จะมีการอ้างไปถึงไฟล์ “foo.html” ถ้า redirection ไม่มีการทำในจุดนั้นไคลเอ็นต์ “ari” เป็นไฟล์ในไดเรกทอรี “people” สาเหตุนี้ความสัมพันธ์ในการอ้างไปยัง “foo.html” จะถูก incorrectly แปลไปยัง URL

http://home .nestcape.com/people/foo.html

Response Header Remappings

การตอบกลับของ header จะบรรจุข้อมูลของเซิร์ฟเวอร์ต้นทางออกมา ตัวอย่างที่สมบูรณ์คือ Location โดย Header จะใช้ redirection Location header จะบรรจุ redirection ปลายทางอยู่ที่ URL โดยปกติปลายทางของ URL จะเป็นจุดที่ไปยังเซิร์ฟเวอร์ที่สร้าง redirection ในตัวอย่างแรกของเราเซิร์ฟเวอร์จะเป็นผลที่ตามมาจากการตอบรับ redirection

HTTP/1.0 302 Found

Server: Netscape-Enterprise/2.01

Date: Sun, 15 Jun 1997 05:34: 28 GMT

Location: http://www.mydite.com/people/ari/

Reverse proxy server ต้องการ remap Location ไฟล์การเปลี่ยนการอ้างไปยัง www.mysite.com พร้อมด้วยตำแหน่งของมัน www2.mysite.com

www.mysite.com => www2.mysite.com

ใน Netscape Proxy Server Location จะตอบรับตามกฎของ header remapping เรียกว่า “reverse Mapping”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Content Remappings

มีความยุ่งยากมากในแง่ของ reverse proxy เซิร์ฟเวอร์ คือมันจะแสดง content ที่ URL อ้างอิงอย่างไร นั้นคือ HTMLจะบรรจุ URL ที่ชัดเจนที่อ้างกลับไปยังเซิร์ฟเวอร์ต้นทางจริงๆ ในตัวอย่าง ในที่นี้ HTML จะอ้างถึง

: http://www.mydite.com/somedir/somepic.gif

: http://www.mydite.com/somedir/somepage.html

นี้เป็นการกำหนดปัญหาที่ใหญ่มากในการที่จะสามารถหาที่อยู่ในทางต่างๆ ไม่มีอันไหนไม่เป็นสาระ, troubfee, หรือเป็นไปได้บางโอกาส

- ใช้ความสัมพันธ์การอ้างถึงเพียงครั้งเดียว
- วิเคราะห์และแก้ไขการอ้าง on-the-fly
- ใช้การปลอมชื่อของ DNS

ปัญหาที่คงอยู่ของ content เพราะว่า reverse proxy เกิดขึ้นใน level ของ HTTP protocal Remapping สามารถที่จะกำจัดการทำงานสำหรับการร้องขอของ URL โดยจะรวมทั้ง header ของทั้งการร้องขอและการตอบรับ อย่างไรก็ตามใน HTML จะติดต่อไปยัง transport layer ในรูปแบบของ URL ผลสุดท้ายคือในคำสั่งที่กระทำตามฟังก์ชันของ reverse proxy บทนำระหว่าง protocal layer และ presentation layer ให้มันแยกออกจากกัน reverse proxy จะควบคุม HTML object

Eliminate Absolute References

ใน URL ที่สมบูรณ์ HTML จะต้องมีการอ้างความสำคัญ 2 ชนิดคือความสัมพันธ์ของ Link relative กับเอกสารปัจจุบัน และ Link relative กับเอกสารต้นของ เซิร์ฟเวอร์โดยแสดงตัวอย่างให้เห็น

htttp://home.netscape.com/people/ari/index.html

ในกรอ้างไฟล์ "picture.gif" ซึ่งจะอยู่ในไดเรกทอรีของ "index.html" ทั้งหมดจะเป็นไปตามการใช้งานของ URL

http://home.netscape.com/people/ari/picture	URL ที่สมบูรณ์
/people/ari/picture.thml	ความสัมพันธ์กับรูต
picture.html	ความสัมพันธ์กับ"index.html"

ในขั้นแรกของการอ้างชนิดของข้อมูลคือมันจะเกิดปัญหาขึ้น ถ้าเซิร์ฟเวอร์ต้นทางมีการอ้าง URL ที่สมบูรณ์ด้วยตัวของมัน ไคลเอ็นต์จะพยายามที่จะแก้ไขไดเรกทอรีจากเซิร์ฟเวอร์ต้นทางและไม่มีตัว reverse proxy

โดยทั่วไปการใช้การอ้าง URL ที่สมบูรณ์จะเป็นความคิดที่ไม่ดีนัก ถ้าเราต้องการย้ายเอกสารไปยังเซิร์ฟเวอร์ตัวอื่นๆ การเชื่อมต่อจะแจ้งเป็นโมฆะและไฟล์จะถูกแก้ไขและเปลี่ยนแปลงโดย hostname ของเซิร์ฟเวอร์ ใน URL เมื่อมีการสร้างการอ้างไปยัง resource อื่นๆ บน เซิร์ฟเวอร์บางตัว URLs ที่สมบูรณ์เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งจำเป็นเสมอ ความสัมพันธ์ในการอ้างอยู่เสมอและได้รับสิทธิพิเศษ อย่างไรก็ตามมันก็ยังอ้างไปยังไฟล์หรือไดเรกทอรีปัจจุบันเสมอ หรือในรากของเอกกสาร โดยปรกติมันจะไม่ทำให้เกิดการเปลี่ยนแปลง

อย่างไรก็ตามเมื่อใช้ reverse proxy ตามลำพังสำหรับเซิร์ฟเวอร์ต้นทางหลายๆ ตัวการอ้างจะไปเกี่ยวเนื่องกับจุบัน และไม่มีความเกี่ยวเนื่องกับรากของเอกสาร

Modify References On-the-fly

ในบางเวลามันจะเป็นไปไม่ได้ที่แยกแยะการอ้าง URL ที่สมบูรณ์ใน content ได้ content จะมี preauthored และ จะมีการอ้าง URL ที่สมบูรณ์หลายอัน ซึ่งเป็นไปไม่ได้ที่มันจะถูกเปลี่ยนทั้งหมด มันอาจจะเก็บไว้บนเครื่องมือที่ไม่สามารถเปลี่ยนได้เช่นในแผ่นดีสก์ ปัญหาอื่นๆที่อาจจะเกิดได้คือการอ้างโดยที่เริ่มสร้างขึ้นมาโดยซอฟต์แวร์จะไม่สามารถเปลี่ยนแปลงได้ง่าย

ความเป็นไปได้อีกอย่างหนึ่งก็คือมันจะทำการกรอง content ที่ใช้อ้างอิงบนพรอกซี่เซิร์ฟเวอร์ และการแก้ไขพวกมันโดยใช้ On-the-fly ซึ่งจุดมุ่งหมายของมันจะกลับไปยัง reverse proxy server ในทางเลือกนี้มันจะข้อเสียหายเพราะ

- filtering จะเป็นสาเหตุของ overhead บน (reverse) พรอกซี่ เซิร์ฟเวอร์
- filtering จะเปลี่ยน Content-Length ของไฟล์

overhead จะก่อให้เกิดการกระจายของเอกสารสำหรับการอ้างและการแก้ไข ในกรณีที่เอกสารเป็นแบบ non-dynamic ตัว filter จะทำการเก็บมันไว้ก่อน หลังจากนั้น filter จะปฏิบัติเมื่อมีการแก้ไขเอกสารเพียงอย่างเดียว ต่อไป ไฟล์ HTML เพียงอย่างเดียวที่ต้องการ filter ไฟล์รูปภาพสามารถที่จะส่ง filter จังหวะได้ การสร้างไฟล์ HTML ที่มีการเปลี่ยนแปลงอยู่เสมอมักจะต้องการ filter อยู่ตลอดเวลา

ในเวลาที่มาก Content-Length คือ header จะถูกส่งโดยเซิร์ฟเวอร์ต้นทาง มันจะส่อให้เห็นถึงขนาดของ object และมันมักจะเอามาจาก ข้อมูลของ filesystem ตัว HTTP1.0 จะไม่ต้องการ Content-Length ก็คือ header นอกจากจะยืนกรานว่าจะใช้ในการติดต่อ อย่างไรก็ตาม HTTP/1.1 จะร้องขออยู่เสมอ Content-Length header ถ้ามัน missing ก้อนของการโอนข้อมูลจะถูกถอดรหัสออกมาใช้นอกจาก hostname ของ reverse proxy เท่ากับความยาวของ hostname ในเซิร์ฟเวอร์ต้นทาง การเปลี่ยน URL จะเกี่ยวข้องกับขนาดของไฟล์ด้วย

Use of DNS Aliases

เป็นหนทางที่แก้ไขปัญหาได้รวดเร็ว ใน บทนำของการปลอม DNS สำหรับ reverse proxy โดยที่ hostname ของเว็บเซิร์ฟเวอร์หลักจะกระจายไปยังเลขหมายประจำเครื่องของ reverse proxy server ในการติดตั้ง network เมื่อ reverse proxy server ถูกเลือกมาใช้แทน การกำหนด DNS จะต้องการการเปลี่ยนแปลงที่ hostname ของเว็บเซิร์ฟเวอร์หลักที่เข้ามาคือจะลบล้างโดยจุดมุงหมายใหม่ของ reverse proxy server ที่เข้ามา

ในบทบาทนี้ reverse proxy จะไม่ถูกเข้าถึงโดยผ่านทาง URL

http://www2.mysite.com/...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

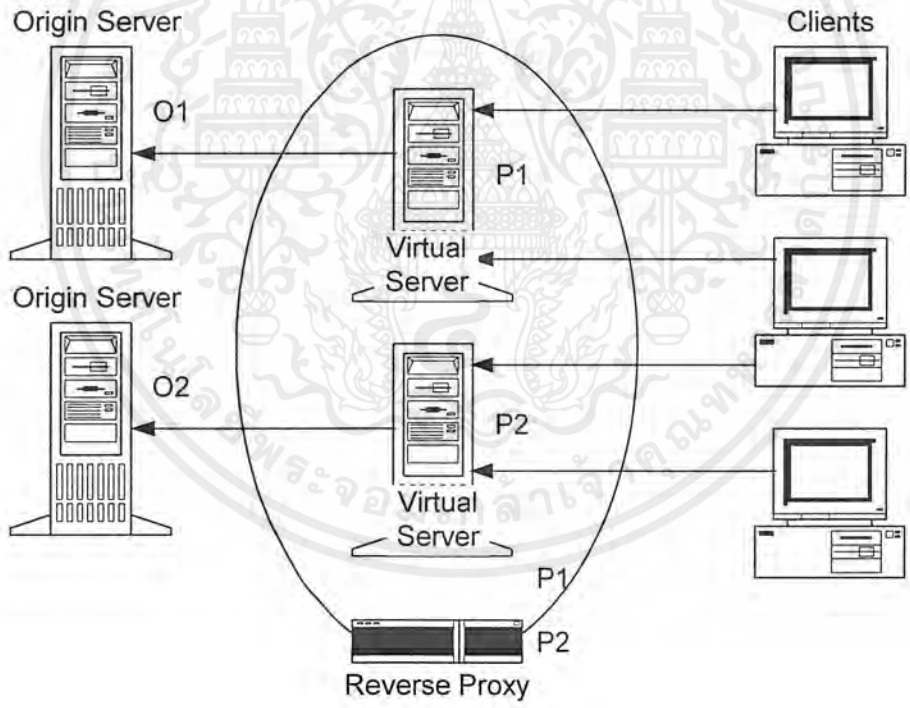
สิ่งที่เข้ามา ในส่วน URL พร้อมกับ hostname ของเว็บเซิร์ฟเวอร์ หลัก

http://www.mysite.com/...

ในชุดของคำสั่งจะมีการย้ายตำแหน่งจาก URL level ไปยัง name service level การแก้ไขปัญหานี้จะง่ายสำหรับผู้ใช้(โดยไม่ต้องรู้เกี่ยวกับ reverse proxy หรือชื่อของมัน) อย่างไรก็ตาม header ก็จะถูกกำหนดและดูแลจาก DNS level

Vitual Multihosting and Reverse Proxy Server

Reverse proxy server จะทำตัวเช่นเดียวกับเว็บเซิร์ฟเวอร์และจะแบ่ง features ต่างๆ ของเว็บเซิร์ฟเวอร์แต่ก่อนเราจะเห็น virtual multihosting คือจะถูกใช้โดย Host header จะถูกพิจารณาจากการจำลอง DNS ที่แสดงอยู่ใน URL และที่ฐานของ hostname ความแตกต่างของ content จะถูกส่งกลับไปยังไคลเอ็นต์บนเว็บเซิร์ฟเวอร์ content จะเหมือนกับตัวอื่นๆ ที่ถูกส่งมาจาก ความแตกต่างของ sub directory บน reverse proxy server URLจะถูก map ไปยังเซิร์ฟเวอร์ต้นทางที่ต่างกัน ในคำว่า Host header เข้าไปข้างใน account เมื่อมีการทำ URL remapping



รูปที่ 2.5 แสดง virtual multihosting โดย Reverse proxy server โดยที่ proxy server

มี DNS แทนด้วย P1และ P2

ซึ่งสามารถแสดงเป็นตัวอย่างได้ดังนี้ พิจารณาการติดตั้งในรูป 2.5 พรอกซี่เซิร์ฟเวอร์ จะถูกติดตั้งให้เป็น reverse proxy สำหรับ web 2 web ที่ต่างกัน DNS จำลองสองตัวจะทำขึ้นสำหรับโฮสต์ของ reverse proxy

P1.mysite.com

P2.mysite.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะสอดคล้องกับเซิร์ฟเวอร์ต้นทางตามลำดับดังนี้

O1.mysite.com

O2.mysite.com

ตามที่ URL ทำการ mapping ได้มีดังนี้

P1.mysite.com => O1.mysite.com

P1 => O1.mysite.com

P2.mysite.com => O2.mysite.com

P2 => O2.mysite.com

และจากการทำ mapping ย้อนกลับ(อ้างอิง: header remapping)ก็จะเพิ่มเป็น

O1.mysite.com => P1.mysite.com

O2.mysite.com => P2.mysite.com

โดยปกติการทำงานของซอฟต์แวร์ของ reverse proxy server จะต้องรองรับ virtual multihosting และ URL remapping

2.4.5. ทำไมพรอกซี่เซิร์ฟเวอร์ถึงไม่รวมอยู่ในตัวเดียวกับเว็บเซิร์ฟเวอร์

คำถามนี้มักจะเป็นคำถามที่ผู้ดูแลเซิร์ฟเวอร์ถามกันบ่อย นั่นคือ "ทำไมพรอกซี่จึงเป็นซอฟต์แวร์ที่แยกออกมาต่างหาก ทำไมไม่รวมกับเว็บเซิร์ฟเวอร์เข้าด้วยกันเสียเลย ?"

ถ้าจะตอบแบบในเชิงที่ไม่ใช่ในทางเทคนิคนักก็คือ มันใช้งานกับผู้ใช้งานที่แตกต่างกัน พรอกซี่เซิร์ฟเวอร์และ เว็บเซิร์ฟเวอร์นั้นมีฐานของผู้ใช้งานที่แตกต่างกัน โดย Origin server นั้นมีจุดมุ่งหมายสำหรับให้ใช้งานได้ทั้งอินเทอร์เน็ตหรือทั้งบริษัท ในขณะที่พรอกซี่เซิร์ฟเวอร์นั้นจะให้บริการเฉพาะผู้ที่เป็นสมาชิกของบริษัทหรือเฉพาะในแต่ละแผนกเท่านั้น ด้วยเหตุผลพื้นฐานตรงที่มันเจาะกลุ่มผู้ใช้ต่างกัน ดังนั้นการที่จะรวมมันไว้ด้วยกันนั้นจึงเป็นการที่ไม่สมควร

หากจะกล่าวถึงเหตุผลในทางเทคนิคแล้วก็ดูเหมือนว่าจะสามารถทำการสร้างเซิร์ฟเวอร์ให้สามารถทำงานเป็นทั้ง origin Web server และพรอกซี่เซิร์ฟเวอร์ในเวลาเดียวกันได้ แต่อย่างไรก็ตาม ยังมีเหตุผลในทางปฏิบัติอีกหลายตัวที่สมเหตุสมผลที่จะแยกทั้งสองตัวออกจากกัน

Enhanced Security

ในมุมมองด้านความปลอดภัยแล้ว จะเป็นการดีถ้าสามารถแยก origin server และ พรอกซี่เซิร์ฟเวอร์ออกจากกันได้ อันที่จริงแล้ว Origin server นั้นตั้งใจให้สามารถเข้าถึงได้จากผู้ใช้งานใดๆ ก็ได้ที่สามารถเข้าถึงอินเทอร์เนตได้ซึ่งรวมทั้งจากภายนอก firewall หรือบน DMZ อีกด้วย ซึ่งบริเวณดังกล่าวนั้นเสี่ยงในด้านความปลอดภัยพอสมควร

Origin Web Server นั้นไม่จำเป็นต้องมีการเชื่อมต่อกับเครือข่ายภายใน แม้ว่าเราสามารถที่จะกำหนดให้ firewall สามารถที่จะบล็อกการเชื่อมต่อที่มีมาจากเว็บเซิร์ฟเวอร์ได้ก็ตาม การกระทำดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการป้องกันเครือข่ายภายในไว้หากเครื่องเว็บเซิร์ฟเวอร์นั้นถูกเจาะระบบได้หรือแม้กระทั่งผู้ที่บุกรุกนั้นจะสามารถทำการควบคุมเครื่องเว็บเซิร์ฟเวอร์ได้แล้วก็ตาม แต่เขาก็ไม่สามารถที่จะทำการเชื่อมต่อกับโฮสต์ที่อยู่ภายใน firewall ได้ ยิ่งกว่านั้นเว็บเซิร์ฟเวอร์จะไม่สามารถทำการส่งการเชื่อมต่อขาออกไปภายนอกได้เนื่องจากสามารถถูกบล๊อกจากการเซ็ทอัพค่าคอนฟิกูเรชันในตัวเราเตอร์ได้ การทำดังกล่าวจะช่วยให้ผู้บุกรุกไม่สามารถทำการเชื่อมต่อจากโฮสต์ของ origin server หรือจากไซต์ที่เขาปลอมเป็นผู้ใช้งานได้

แต่สำหรับพรอกซี่เซิร์ฟเวอร์กลับต่างไปจากนั้น เพราะมันไม่จำเป็นที่จะต้องสามารถรับการเชื่อมต่อภายนอกที่มีมาจากโฮสต์ที่อยู่ในอินเทอร์เน็ต แต่มันจะรับการเชื่อมต่อเฉพาะจากโฮสต์ภายใน firewall เท่านั้น นั่นหมายถึงเราสามารถที่จะเซ็ทอัพพรอกซี่เซิร์ฟเวอร์ให้มีความปลอดภัยภายใน DMZ ได้มากกว่า origin web server ทำให้สามารถป้องกันผู้บุกรุกได้ดีกว่า ตัว firewall router นั้นสามารถที่จะเซ็ทให้บล๊อกการเชื่อมต่อทั้งหมดที่พยายามจะทำกับตัวโฮสต์ของพรอกซี่เซิร์ฟเวอร์จากอินเทอร์เน็ตภายนอกได้ และผู้บุกรุกไม่มีโอกาสแม้กระทั่ง เข้าถึงโฮสต์ของ พรอกซี่เซิร์ฟเวอร์ ได้เลย

Ease of Administration

การแยกการทำงานของ Origin Web Server และ พรอกซี่เซิร์ฟเวอร์ออกจากกันนั้นทำให้การดูแลระบบง่ายขึ้น เนื่องจากการอินเทอร์เฟสในการดูแลการใช้งานของ origin server และ พรอกซี่เซิร์ฟเวอร์นั้นถูกแยกออกจากกันทำให้ลดความผิดพลาดจากการคอนฟิกูเลชันได้ ตัวอย่างเช่น access control นั้นอาจจะถูกกำหนดไม่ถูกต้องหากเราไปทำฟังก์ชันดังกล่าวกับ origin server แทนที่จะทำกับพรอกซี่เซิร์ฟเวอร์

Modularization of Development

จากมุมมองของผู้พัฒนาซอฟต์แวร์แล้ว การแยกสองฟังก์ชันออกมาจะทำให้การพัฒนาโปรแกรมสามารถทำได้ง่ายขึ้น ทั้งเว็บเซิร์ฟเวอร์และพรอกซี่เซิร์ฟเวอร์แม้บางการทำงานอาจดูเหมือนจะแบ่งฟังก์ชันการทำงานกันได้แต่ก็มีบางส่วนที่แตกต่างกันทำให้ยากแก่การรวมมาไว้เข้าด้วยกัน การแยกกันพัฒนาจะทำให้เสถียรภาพของโปรแกรมตลอดจน ความรวดเร็วในการทดสอบนั้นดีกว่าเนื่องจากขนาดของซอฟต์แวร์นั้นเล็กกว่า

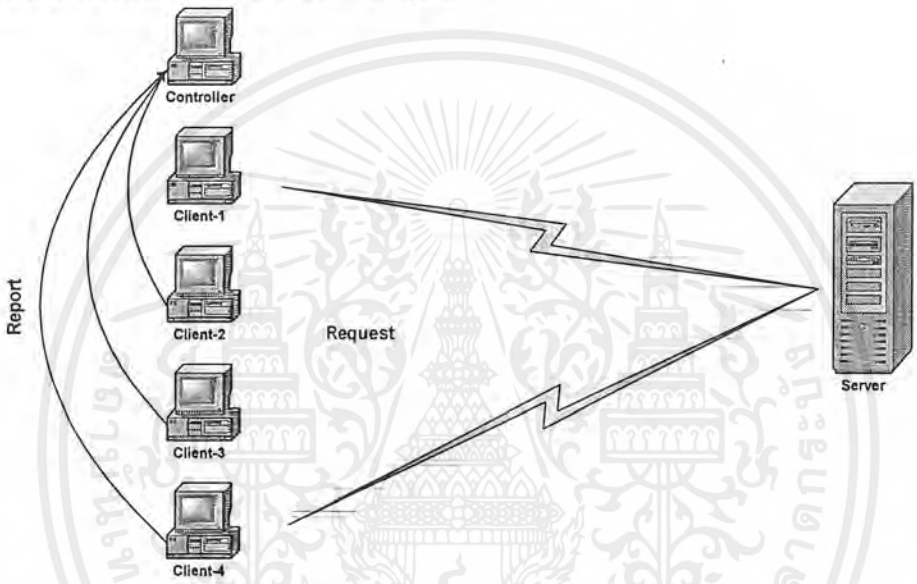
Marketing

ในมุมมองทางการตลาดของผู้ผลิตซอฟต์แวร์อินเทอร์เน็ตแล้ว เป็นธรรมดาอยู่แล้วว่าการแยกผลิตภัณฑ์ออกจากกันจะทำให้สามารถได้กำไรที่ดีกว่า เนื่องจากสามารถที่จะทำตลาดของทั้งสองตัวแยกจากกันได้ และในมุมมองของผู้ใช้แล้วก็เป็นการดีที่จะสามารถซื้อเฉพาะซอฟต์แวร์ที่ต้องการจริงๆ และไม่ต้องเสียเงินกับส่วนที่ตัวเองไม่ได้ใช้งาน(ในกรณีรวมเป็นโปรแกรมประยุกต์ตัวเดียวเดียว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5. Test Bench Program

โปรแกรม WebBench จัดเป็นโปรแกรมประเภท TestBench ซึ่งทำหน้าที่หลักในการทดสอบประสิทธิภาพการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ ลักษณะการทำงานคือ มีเครื่องไคลเอนต์ เป็นตัวที่ทำงานเสมือนเป็นโปรแกรมบราวเซอร์ แต่จะไม่เหมือนไปเสียทีด้วย กล่าวคือ โปรแกรม WebBench จะไม่แสดงผลของการสัญญาณร้องขอที่ส่งไปยังเซิร์ฟเวอร์ปลายทาง ซึ่งถ้าเป็นโปรแกรมบราวเซอร์ธรรมดาจะแสดงข้อมูลที่รับมาได้ออกทางจอภาพเป็นต้น เมื่อโปรแกรม WebBench ได้รับข้อมูลกลับมาจากเว็บเซิร์ฟเวอร์ก็จะส่งสัญญาณร้องขอไปใหม่ จึงเสมือนเป็นการสร้างโหลดเพื่อทดสอบการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ปลายทางได้ ลักษณะดังกล่าวแสดงดังรูป 2.6



รูปที่ 2.6 แสดงการทำงานของ Test Bench Program

จากรูป 2.6 สังเกตการควบคุมการทำงานของเครื่องไคลเอนต์จะถูกควบคุมโดยเครื่องคอนโทรลเลอร์ ดังรูป เราสามารถที่จะเพิ่มจำนวนของเครื่องไคลเอนต์เป็นจำนวนมากๆได้

2.5.1 ประโยชน์ของโปรแกรม Webbench

กล่าวอย่างง่ายๆ คือ เมื่อทำการติดตั้งโปรแกรม WebBench และนำชุดไฟล์ที่ใช้สำหรับทำการทดสอบเข้าไปเก็บไว้ในเครื่องเซิร์ฟเวอร์ที่เราจะทำการทดสอบ ชุดไฟล์ที่ใช้สำหรับการทดสอบดังกล่าวจะประกอบไปด้วยไฟล์เอกสารที่อยู่ในรูปแบบของเอกสาร HTML, ไฟล์รูปภาพที่อยู่ในรูปแบบของไฟล์ GIF และ ไฟล์โปรแกรมที่สามารถรันได้จำพวก CGI เป็นต้น

ตัวไคลเอนต์ของโปรแกรม WebBench จะใช้โพรโตคอล HTTP ในการติดต่อกับเครื่องเซิร์ฟเวอร์ ลักษณะในการส่งสัญญาณร้องขอไปยังเครื่องเซิร์ฟเวอร์นั้นก็จะส่งสัญญาณร้องขอที่จะดึงดู ข้อมูลในชุดไฟล์สำหรับการทดสอบที่ได้กล่าวมาแล้วในเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อตัวไคลเอนต์ได้รับข้อมูลที่ตอบกลับจากเครื่องเซิร์ฟเวอร์เราได้จะได้ข้อมูลต่างซึ่งจะบ่งบอกถึงประสิทธิภาพการทำงานของเครื่องเซิร์ฟเวอร์นั้น เช่น จำนวนความสำเร็จของการส่งสัญญาณร้องขอ, จำนวนของความล้มเหลวของการส่งสัญญาณร้องขอเมื่อจบการทดสอบ ตัวโปรแกรมควบคุมกลางของ WebBench จะรวบรวมข้อมูลจากตัวไคลเอนต์ทั้งหลาย แล้วนำมาสรุปเป็นผลของการทดสอบครั้งนั้นๆ ผลที่ได้จะออกมาสองแบบคือ อัตราการส่งข้อมูลร้องขอต่อนาที และ อัตราของงานที่ทำเสร็จต่อนาที ซึ่งโปรแกรม WebBench จะแสดงผลที่ได้ออกมาในลักษณะที่เป็นตาราง

2.5.2 จุดหลักๆ ของโปรแกรม Webbench

จุดหลักๆของโปรแกรม WebBench สรุปได้เป็นข้อๆดังนี้

- โปรแกรม WebBench เป็นโปรแกรมที่ใช้สำหรับวัดประสิทธิภาพการทำงานของเว็บเซิร์ฟเวอร์ในส่วนของซอฟต์แวร์และฮาร์ดแวร์
- ในส่วนของตัวไคลเอนต์จะใช้โพรโตคอล HTTP เป็นโพรโตคอลหลักที่ใช้ในการติดต่อกับเซิร์ฟเวอร์
- ในการทำงานของโปรแกรม WebBench โปรแกรมดังกล่าวจำเป็นต้องมีส่วนประกอบสองส่วนคือ ส่วนของโปรแกรมควบคุมส่วนกลาง (Controller) และตัวไคลเอนต์ (Client) ซึ่งตัวไคลเอนต์อาจจะมีเครื่องเดียวหรือหลายเครื่องก็ได้
- โปรแกรม WebBench เป็นโปรแกรมของค่าย ZD จึงใช้มาตราฐานการวัดต่างๆของค่าย ZD และในส่วนของหน้าจอแสดงผล ก็จะคล้ายกับโปรแกรม TestBench ตัวอื่นๆของค่าย ZD
- โปรแกรม WebBench มีชุดทดสอบสำหรับ E-Commerce ซึ่งผู้ใช้งานสามารถที่จะทดสอบในส่วนที่ใช้งาน SSL ได้
- โปรแกรม WebBench สนับสนุนการทำสอบความสามารถทางการใช้งาน SSL ของเซิร์ฟเวอร์
- ในการทดสอบแต่ละครั้ง ไม่ควรมีการติดต่อของผู้ใช้งานปกติกับเซิร์ฟเวอร์ เพราะจะทำให้ผลของการทดสอบที่ได้ผิดพลาดไป
- ส่วนของโปรแกรมควบคุมกลาง (Controller) จะต้องรันบนเครื่องที่ลงระบบปฏิบัติการ Windows NT 3.51 หรือเป็น 4.0 ส่วนของตัวไคลเอนต์นั้น สามารถรันบนเครื่องที่ลงระบบปฏิบัติการ Windows 95/98 หรือแม้ว่าจะเป็น Windows NT
- ในการแสดงผลของการทดสอบครั้งนั้นๆ โปรแกรม WebBench จะแสดงผลในรูปของเอกสารของ Microsoft Excel ซึ่งแสดงในรูปของตาราง
- ส่วนของชุดไฟล์ที่ใช้ในการทดสอบ ผู้ใช้สามารถที่จะปรับแต่งได้อย่างง่ายดาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

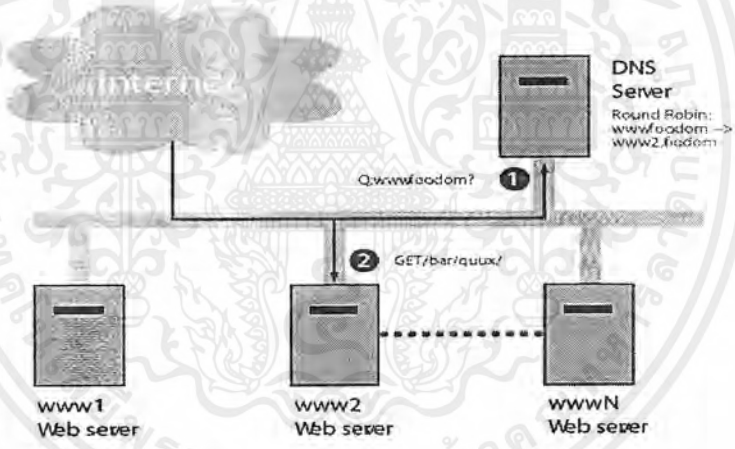
2.6 Load Balancing Web Server Technic

ปัจจุบันการใช้งานอินเทอร์เน็ตได้รับความนิยมอย่างกว้างขวาง ซึ่งบริการหนึ่งที่นิยมใช้งานกันมากคือ บริการเกี่ยวกับเอกสารอิเล็กทรอนิกส์ (HTML) ซึ่งบริการให้โดยโปรแกรมเว็บเซิร์ฟเวอร์ แน่นอนว่าเมื่อเว็บไซต์ใดที่ได้รับความนิยมสูง จำนวนของผู้เข้าเยี่ยมชมก็จะมากตามไปด้วย ส่งผลถึงภาระงานที่เพิ่มมากขึ้นของเซิร์ฟเวอร์ ตัวอย่างของเว็บไซต์ที่มีภาระงานมากๆ เช่น ไซต์ของบริษัทไมโครซอฟต์ เป็นต้น ทำให้ผู้เข้าเยี่ยมชมเว็บไซต์ดังกล่าวรู้สึกว่าเว็บไซต์ดังกล่าวทำงานช้า และอาจทำให้ความนิยมในการเข้าเยี่ยมชมเว็บไซต์ดังกล่าวน้อยลง กว่าที่ควรจะเป็น

เทคนิคในการเพิ่มประสิทธิภาพการทำงานให้กับเว็บไซต์นั้นมีด้วยกันหลายวิธีตัวอย่าง ในที่นี้จะกล่าวถึงเฉพาะ การปรับแต่งที่เกี่ยวกับตัวโปรแกรมเป็น โดยวิธีการต่างๆนั้นสามารถแบ่งออกได้เป็น 2 แบบ คือ

2.6.1 การใช้หลักการของโดเมนเนมเข้าช่วย (The DNS Approach)

ในการปรับแต่งแบบนี้ใช้การปรับแต่งโดเมนเนมเซิร์ฟเวอร์เข้าช่วย โดยลักษณะของการเชื่อมต่อเป็นดังรูปที่ 2.7



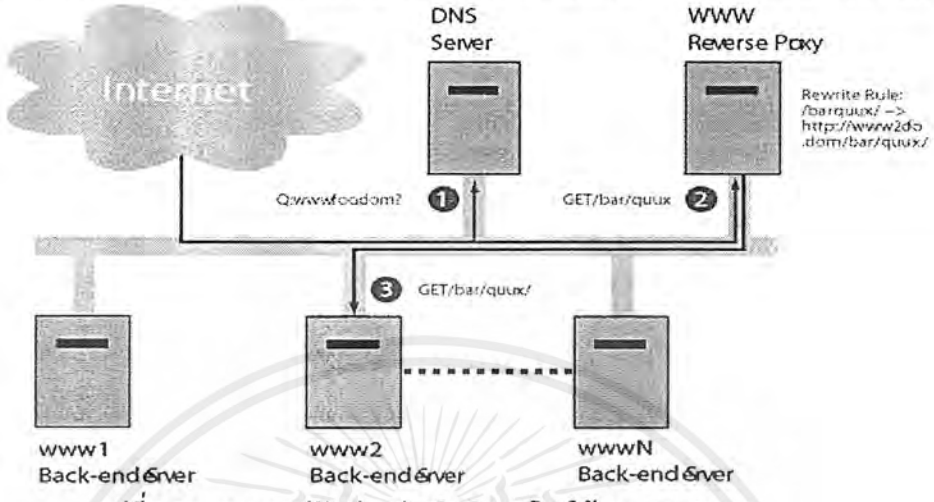
รูปที่ 2.7 แสดงการเชื่อมต่อโดยใช้โดเมนเนมเซิร์ฟเวอร์เข้าช่วย

การปรับแต่งในลักษณะนี้เป็นที่นิยมกันมากในปัจจุบัน เพราะเป็นการปรับแต่งที่ทำได้ง่ายไม่ยุ่งยากในการปรับแต่งใดๆ หลักการทำงานมีอยู่ว่า เมื่อมีสัญญาณร้องของเข้ามายังโดเมนเนมเซิร์ฟเวอร์ ซึ่งที่โดเมนเนมเซิร์ฟเวอร์นี้เองมีการเก็บรายการชื่อของเซิร์ฟเวอร์ต่างๆ ดังรูป โดเมนเนมเซิร์ฟเวอร์จะเก็บรายการชื่อของเซิร์ฟเวอร์ www1, www2 และ www3 ตามลำดับ และในการตอบกลับไอพีให้กับเครื่องที่ส่งสัญญาณร้องขอเข้ามานั้น โดเมนเนมเซิร์ฟเวอร์จะทำการสุ่มค่าไอพีของเซิร์ฟเวอร์กลับให้ไป ดังในรูปเมื่อมีสัญญาณร้องขอเข้ามาโดเมนเนมทำการสุ่มค่าแล้วได้ค่าไอพีของเซิร์ฟเวอร์ตัวที่ 2 จึงส่งค่าไอพีของเซิร์ฟเวอร์ตัวที่ 2 กลับไปให้กับทางผู้ร้องขอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 การใช้หลักการของ Reverse Proxy (The Reverse Proxy Approach)

ในการปรับแต่งลักษณะนี้ใช้ความสามารถพิเศษของพรอกซี่เซิร์ฟเวอร์เข้าช่วย เป็นการกำหนดให้พรอกซี่เซิร์ฟเวอร์ทำงานในโหมดที่เรียกว่า Reverse Proxy มีลักษณะของการจัดรูปแบบดังรูปที่ 2.8



รูปที่ 2.8 แสดงการปรับปรุงประสิทธิภาพโดยใช้ Reverse Proxy

จะสังเกตได้ว่าในแบบนี้หน้าหลักของโดเมนเนมเซิร์ฟเวอร์จะเป็นแค่โดเมนเนมธรรมดา แต่ตัวที่จะทำการ Redirect หรือตัวเลือกว่าจะเอาข้อมูลจากเซิร์ฟเวอร์ตัวไหน เป็นหน้าที่พรอกซี่ซึ่งถ้าข้อมูลที่ต้องการมีอยู่ในพรอกซี่อยู่แล้วก็ไม่จำเป็นที่จะต้องไปเอาที่เซิร์ฟเวอร์ข้อดีของ Reverse Proxy คือ เวลาที่ใช้ในการดึงข้อมูลออก ถ้าเทียบกับโปรแกรมเว็บเซิร์ฟเวอร์แล้วพรอกซี่เซิร์ฟเวอร์มีความเร็วกว่า

แต่การปรับปรุงประสิทธิภาพในแบบนี้ไม่เคยมีคนทำกันนัก เหตุผลเพราะเป็นเรื่องที่ยุ่งยากในการทำกำหนดการทำงานของพรอกซี่เซิร์ฟเวอร์ให้อยู่ในโหมด Reverse Proxy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

3.1. การติดตั้งโดเมนเนมเซิร์ฟเวอร์

Bind (Berkeley Internet Named Domain) เป็นซอฟต์แวร์ที่ทำหน้าที่เป็น เซิร์ฟเวอร์ DNS (Domain Name Service Server) ซึ่งเป็นสิ่งที่จำเป็นมากสำหรับเครือข่ายอินเทอร์เน็ต การอ้างถึงที่อยู่บนเครือข่ายอินเทอร์เน็ตนั้นจะต้องใช้หมายเลขประจำเครื่อง แต่การที่เราสามารถใช้ชื่อแทน เช่น www.ce.kmitl.ac.th นั้น ก็ทำได้เนื่องจากเรามี เซิร์ฟเวอร์ DNSซึ่งทำหน้าที่แปลงระหว่างชื่อกับเลยหมายเลขประจำเครื่อง นั่นเอง

3.1.1. ความเป็นมา

ซอฟต์แวร์ที่ทำหน้าที่ เซิร์ฟเวอร์ DNSนั้น เกิดขึ้นมาพร้อมๆกับ เครื่องข่ายอินเทอร์เน็ตในยุคแรกเลยทีเดียว และซอฟต์แวร์ BIND ก็จัดว่าเป็นตัวที่ใช้กันแพร่หลายมากที่สุด BIND พัฒนาขึ้นโดยา มหาวิทยาลัย Berkeley ในรัฐ California ประเทศสหรัฐอเมริกา โดยได้รับทุนสนับสนุนจากรัฐบาลสหรัฐ จนถึงเวอร์ชัน 4.8.3 ต่อมาได้รับการสนับสนุนจาก บริษัท Digital Equipment Corparation (ปัจจุบันคือ บริษัท Compaq Computer Corperation) ในเวอร์ชัน 4.9 และ 4.9.1 หลังจากนั้นได้รับการสนับสนุนจาก บริษัท Vixie Enterprises สำหรับเวอร์ชัน 4.9.2 และนับตั้งแต่นั้นเป็นต้นมาจนถึงปัจจุบัน ได้รับการสนับสนุนจาก Internet Software Consortium BIND สามารถทำหน้าที่ได้เป็นทั้ง DNS Primary, Secondary และ Cache Name Server

3.1.2. วิธีการติดตั้ง

โปรดอ่านบทที่ 2 เรื่องโดเมนเนมเซิร์ฟเวอร์ก่อน เพื่อให้เกิดความเข้าใจการทำงานของระบบ DNS และ เข้าใจความแตกต่างของ Primary, Secondary และ Cache Name Server ระบบ Unix ในปัจจุบันส่วนใหญ่จะมีการติดตั้ง เซิร์ฟเวอร์ DNSมาให้อยู่แล้ว แต่อาจจะยังไม่ได้ถูกสั่งให้ทำงาน สามารถตรวจสอบว่ามันทำงานอยู่หรือไม่โดยให้สั่งดังนี้

```
# ps ax | grep named
  765  ?  S   0:00  named
```

ถ้าเป็นอย่างนี้แสดงว่าระบบมีการติดตั้ง เซิร์ฟเวอร์ DNSและทำงานได้เรียบร้อยแล้ว (โดยมากจะถูกติดตั้งเป็น DNS Cache Server) หากไม่พบบรรทัดที่มีคำว่า Named ดังกล่าว ให้ท่านตรวจสอบว่าในระบบของท่านมีไฟล์ named สำหรับ Linux ส่วนมากจะถูกติดตั้งอยู่ที่ /usr/sbin/named หรือไม่ หากยังไม่มีท่านจะต้องนำมาลงเอง โดยมีวิธีการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1. ดาวน์โหลดโปรแกรมที่ ftp://ftp.isc.org/isc/bind/src / โดยแนะนำให้ใช้เวอร์ชันที่ขึ้นต้นด้วยเลข 8 (BIND จะมีอยู่ 2 เวอร์ชันใหญ่ๆ คือ 4 และ 8 เครื่องสมัยเก่าส่วนมากจะใช้เวอร์ชัน 4 กันอยู่ วิธีสังเกตแบบง่ายๆ เวอร์ชัน 4 จะใช้ Configuration file ชื่อ named.boot ส่วนเวอร์ชัน 8 จะใช้ named.conf)

3.1.2.2. ทำการ unpack ดังนี้ (สมมุติว่าใช้เวอร์ชัน 8.1.2)

```
# tar xzvf bind-8.12-src.tar.gz
```

3.1.2.3. ตัวโปรแกรมจะถูกแตกออกมา โปรดอ่านไฟล์ README และ/หรือ INSTALL ถ้ามี

3.1.2.4. ทำการ compile โปรแกรม และทำการติดตั้งดังนี้

```
# cd src
# make
# make install
```

3.1.2.5. ถึงขั้นตอนนี้ ท่านควรจะมีไฟล์ named อยู่ในไดเรกทอรี /usr/sbin เรียบร้อยแล้ว ขั้นตอนต่อไปคือจะต้องสร้างไฟล์ /etc/namded.conf โดยท่านอาจดูจากตัวอย่างซึ่งจะมีให้ที่ /src/named/named.conf หรือดังที่แสดงเป็นตัวอย่างต่อไปนี้

3.1.2.6. กรณีทำ Cache Server อย่างเดียว สร้างไฟล์ /etc/named.conf ดังนี้

```
/*
 * A simple BIND 8 configuration
 * For Linux-SIS Version 3.0
 * ott@nectec.or.th, Oct 27, 1998
 */

options {
        directory "/var/named";
};

logging {
        category lame-servers { null; };
        category cname { null; };
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
zone "." in {
        type hint;
        file "root.cache";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "master/127.0.0";
};

zone "yearm.com." in {
        type master;
        file "master/yearm.com";
};
zone "1.168.192.in-addr.arpa" in {
        type master;
        file "master/192.168.1";
};
```

- สร้างไฟล์ /var/named/root.cache โดยใช้คำสั่ง dig ดังนี้

```
# dig @rs.internet.nic .          ns  > /var/named/root.cache
```

- หากใช้คำสั่ง dia ไม่ได้ให้สร้าง /var/named/root.cache เอง โดยมีตัวอย่างดังนี้

```
; <<>> DiG 8.2 <<>>
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;; ANSWER SECTION:
.               3d8h6m16s IN NS  F.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  B.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  J.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  K.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  L.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  M.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  I.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  E.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  D.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  A.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  H.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  C.ROOT-SERVERS.NET.
.               3d8h6m16s IN NS  G.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
F.ROOT-SERVERS.NET.    6d6h14m56s IN A  192.5.5.241
B.ROOT-SERVERS.NET.    6d6h45m20s IN A  128.9.0.107
J.ROOT-SERVERS.NET.    4d8h6m16s IN A  198.41.0.10
K.ROOT-SERVERS.NET.    4d8h6m16s IN A  193.0.14.129
L.ROOT-SERVERS.NET.    4d8h6m16s IN A  198.32.64.12
M.ROOT-SERVERS.NET.    4d8h6m16s IN A  202.12.27.33
I.ROOT-SERVERS.NET.    6d10h58m3s IN A  192.36.148.17
E.ROOT-SERVERS.NET.    6d6h45m20s IN A  192.203.230.10
D.ROOT-SERVERS.NET.    6d6h14m56s IN A  128.8.10.90
A.ROOT-SERVERS.NET.    6d6h45m20s IN A  198.41.0.4
H.ROOT-SERVERS.NET.    6d6h45m20s IN A  128.63.2.53
C.ROOT-SERVERS.NET.    6d6h14m56s IN A  192.33.4.12
G.ROOT-SERVERS.NET.    6d6h45m20s IN A  192.112.36.4

;; Total query time: 16 msec
;; FROM: isag20 to SERVER: default -- 161.246.4.3
;; WHEN: Thu Dec 23 16:04:57 1999
;; MSG SIZE  sent: 17  rcvd: 436
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใ

- สร้างไดเรกทอรี /var/named/master และสร้างไฟล์ /var/named/master/5.246.161 ดังนี้

```
@   IN   SOA   isag20.ce.kmitl.ac.th. root.ce.kmitl.ac.th.(
          991223162
          10800
          3600
          432000
          38400)
    IN   NS    isag20.ce.kmitl.ac.th.

20  IN   PTR   isag20.ce.kmitl.ac.th.

4   IN   PTR   isag04.ce.kmitl.ac.th.
6   IN   PTR   isag06.ce.kmitl.ac.th.
7   IN   PTR   isag07.ce.kmitl.ac.th.
21  IN   PTR   isag21.ce.kmitl.ac.th.
22  IN   PTR   isag22.ce.kmitl.ac.th.
25  IN   PTR   isag25.ce.kmitl.ac.th.
26  IN   PTR   isag26.ce.kmitl.ac.th.
```

3.1.2.7. การทำเป็น Primary Name Server มี 2 กรณี

- กรณีที่เป็น Forword (แปลงชื่อเป็นหมายเลขประจำเครื่อง) ตัวอย่างเช่นต้องการเป็น Primary Name Server สำหรับโดเมน ce.kmitl.ac.th จะต้องเพิ่มเติมข้อความใน /etc/named.conf ต่อท้ายจากในข้อที่ 6 ดังนี้

```
zone "ce.kmitl.ac.th"{
    type master;
    file "/var/named/master/ce.kmitl.ac.th";
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- และให้สร้างไฟล์ /var/named/master/ce.kmitl.ac.th โดยมีเนื้อหาดังตัวอย่างนี้

```
@IN    SOA    isag20.ce.kmitl.ac.th. root.isag20.ce.kmitl.ac.th.(
            991223162
            604800
            3600
            604800
            86400)
            IN NS  161.246.10.21.
            IN NS  161.246.10.22.

localhost       IN  A     127.0.0.1

ftp          IN A         161.246.4.8
www          IN A         161.246.4.2

isag04       IN    A     161.246.5.4
isag06       IN    A     161.246.5.6
isag07       IN    A     161.246.5.7
isag21       IN    A     161.246.5.21
isag22       IN    A     161.246.5.22
isag25       IN    A     161.246.5.25
isag26       IN    A     161.246.5.26
```

- กรณีที่เป็น Revers (แปลงกลับหมายเลขประจำเครื่องเป็นชื่อ) เช่น จะทำ Reverse สำหรับหมายเลขประจำเครื่อง 161.246.5.0/24 จะต้องเพิ่มเติมข้อความใน /etc/named.conf ต่อท้ายจากในข้อ 6 ดังนี้

```
zone "5.246.161.in-addr.arpa" {
    type master;
    file "/var/named/master/5.246.161";
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- และให้สร้างไฟล์ /var/named/master/161.246.5 ตามตัวอย่างดังนี้

```
@    IN    SOA    isag20.ce.kmitl.ac.th. root.ce.kmitl.ac.th.(
            991223162
            10800
            3600
            432000
            38400)
     IN    NS     isag20.ce.kmitl.ac.th.

20   IN    PTR    isag20.ce.kmitl.ac.th.

4    IN    PTR    isag04.ce.kmitl.ac.th.
6    IN    PTR    isag06.ce.kmitl.ac.th.
7    IN    PTR    isag07.ce.kmitl.ac.th.
21   IN    PTR    isag21.ce.kmitl.ac.th.
22   IN    PTR    isag22.ce.kmitl.ac.th.
25   IN    PTR    isag25.ce.kmitl.ac.th.
26   IN    PTR    isag26.ce.kmitl.ac.th.
```

3.1.2.8. เมื่อทุกอย่างเรียบร้อย การติดตั้งให้ named ทำงานโดยอัตโนมัติ ทุกครั้งที่เปิดเครื่อง สำหรับ Slackware นั้น ให้ไปเอาเครื่องหมายคอมเมน ในส่วนของ named ออกดังนี้

```
# # Start the NAMED/BIND name server:
 if [ -f ${NET}/named ]; then
  echo -n " named"
  ${NET}/named -u daemon -g daemon
 fi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3. วิธีทดสอบการใช้งาน

3.1.3.1. ทดสอบด้วยคำสั่ง host หรือ nslookup โดยจะได้ผลลัพธ์ลักษณะดังนี้

```
isag20:~# host isag20.ce.kmitl.ac.th
isag20.ce.kmitl.ac.th has address 161.246.5.20
```

```
isag20:~# host 161.246.5.20 localhost
Using domain server:
Name: localhost.ce.kmitl.ac.th
Address: 127.0.0.1
Aliases:

20.5.246.161.IN-ADDR.ARPA domain name pointer isag20.ce.kmitl.ac.th
```

```
isag20:~# host -t any ce.kmitl.ac.th localhost
Using domain server:
Name: localhost.ce.kmitl.ac.th
Address: 127.0.0.1
Aliases:

ce.kmitl.ac.th name server 161.246.10.21
ce.kmitl.ac.th name server 161.246.10.22
```

3.1.4. การแก้ไขปัญหา

หากผลการทดสอบดูแล้วไม่เป็นไปตามที่ควรจะเป็น

3.1.4.1. ตรวจสอบว่าเซิร์ฟเวอร์ทำงานอยู่หรือเปล่า โดยใช้คำสั่ง ps –ax

3.1.4.2. กว่า 90 % ของความผิดพลาดในการติดตั้ง เซิร์ฟเวอร์ DNSมาจากการสะกดผิด ในค่ากำหนดต่างๆ ซึ่งความผิดพลาดเพียงนิดเดียว ทำให้ไม่สามารถใช้งานได้

3.1.4.3. ใช้คำสั่ง tail /var/log/messages เพื่อดูการทำงานของ named ช่วยในการหาสาเหตุของปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2. การติดตั้ง Apache Web Server

ในส่วนนี้เราจะกล่าวถึงการติดตั้งและการกำหนดค่าเริ่มต้นการใช้งานสำหรับ Apache Server ตัวโปรแกรม Apache Server สามารถหาได้จากทางเว็บไซต์ของ Apache เองคือ www.apache.org ซึ่งเราสามารถที่จะเลือกชุดของการติดตั้งได้ 2 แบบ คือ

3.2.1. แบบที่ผ่านการคอมไพล์มาแล้ว

3.2.2. แบบที่ยังไม่ผ่านการคอมไฟล์มา

ซึ่งในแบบที่ 2 เราต้องมาทำการคอมไฟล์เองที่เครื่องของเรา ทางทีมงานผู้พัฒนา Apache Server ได้ออกโปรแกรม Apache มาให้สามารถทำงานได้หลายแพลตฟอร์ม ดังนี้

1. AUX 3.1
2. BSDI 2.0
3. FreeBSD 2.1
4. HP-UX 9.07
5. IRIX 5.3
6. Linux
7. NetBSD 1.1
8. NeXTSTEP
9. SolarisX86 2.5
10. Solaris 2.4
11. Solaris 2.5
12. SunOS 4.1.3
13. UnixWare 1.1.2

สำหรับการติดตั้ง Apache Server นั้นค่าของ Default Path ของการติดตั้งจะอยู่ที่ /usr/local/etc/httpd ซึ่งในการติดตั้งจะต้องใช้สิทธิ์ของ Superuser ถึงจะสามารถทำการติดตั้ง ถ้าผู้ที่ต้องการติดตั้งไม่มีสิทธิ์ของ Superuser ก็ต้องติดต่อไปยังผู้ดูแลระบบ

3.2.1. การติดตั้ง Apache Server แบบที่ตัวโปรแกรมผ่านการคอมไพล์มาแล้ว

จากที่ทางผ่ายทีมงานผู้พัฒนา Apache Server ได้เตรียมไฟล์สำหรับการติดตั้งบนเครื่องหลายแพลตฟอร์มดังที่ได้กล่าวมาแล้ว ไฟล์ที่ได้ดาวน์โหลดมาจะอยู่ในรูปแบบของไบนารีไฟล์ ซึ่งชื่อของไฟล์จะมีลักษณะที่แตกต่างกันไปตามแพลตฟอร์ม โดยมีลักษณะดังนี้ httpd-architecture เช่นถ้าเป็นไฟล์ที่ใช้สำหรับการติดตั้งในแพลตฟอร์มลีนุกซ์ จะมีชื่อไฟล์เป็น httpd-linux

ในการติดตั้งเราก็คัดลอกไฟล์ที่เราดาวน์โหลดมาไปไว้ที่ไดเรกทอรี /uer/local/etc /httpd/httpd ในบางลักษณะไบนารีไฟล์บางตัวมีส่วนที่ไม่ได้ใช้งานขึ้นอยู่กับแพลตพอร์ม เราควรสั่ง Stripped เพื่อตัวส่วนที่ไม่จำเป็นต้องใช้งานออก เพื่อประหยัดพื้นที่ดิสก์ของเราเอง อีกส่วนหนึ่งคือเรื่องของความปลอดภัย คือ ไฟล์ httpd ควรจะมีจัดการเรื่องเกี่ยวกับสิทธิ์การเข้าถึงไฟล์ดังกล่าว ตามปกติแล้วยูสเซอร์ธรรมดาไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควรมีสิทธิ์แก้ไขไฟล์นี้ได้ ผู้ที่มีสิทธิ์ในการเรียกใช้หรือแก้ไขควรจะเป็นผู้ดูแลระบบเท่านั้น เราควรเป็นส่วนของ Owner ของไฟล์ให้เป็น root และ Group ของไฟล์ให้เป็น Wheel โดยทำตามขั้นตอนข้างล่างนี้

```
1. cd /usr/local/etc/httpd
2. cp /src/httpd-linux httpd
3. strip httpd
4. chown root.whell httpd
5. chmod 755 httpd
```

3.2.2. การติดตั้ง Apache Server แบบที่ตัวโปรแกรมที่ยังไม่ผ่านการคอมไพล์มาแล้ว

ในการคอมไพล์โปรแกรม Apache Server นั้นเราต้องแก้ไขรายละเอียดของไฟล์คอนฟิกุเรชันให้เหมาะสมกับระบบปฏิบัติการที่เราใช้งานอยู่ ซึ่งไฟล์ดังกล่าวเก็บไว้ที่ /usr/local/etc/httpd/src ในการแก้ไขไฟล์ดังกล่าวเราควรที่จะทำการสำรองข้อมูลดังกล่าวเอาไว้ก่อน เพื่อเมื่อเกิดความผิดพลาดเราจะได้แก้ไขได้ โดยแรกเริ่มทางทีมงานผู้สร้างได้ให้ตัวอย่างการเขียนไฟล์คอนฟิกุเรชันมาด้วย เพียงเราทำตามขั้นตอนดังนี้

```
cd /usr/local/etc/httpd/src
cp Configuration.tmpl Configuration
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3. การติดตั้ง Squid Proxy Server

3.3.1 วิธีการติดตั้ง

3.3.1.1. สามารถดาวน์โหลดตัวโปรแกรมได้จาก http://squid.nlanr.net

3.3.1.2. เมื่อได้ไฟล์มาแล้ว ทำการขยายไฟล์ดังกล่าวด้วยคำสั่ง

```
# tar –xzvf squid-version.tar.gz
```

3.3.1.3. เข้าไปยังไดเรกทอรีที่การขยายไฟล์ข้างต้นได้สร้างขึ้นมาใหม่ จากนั้นใช้คำสั่งดังต่อไปนี้

```
# cd squid-version
# ./configure -- prefixe=/usr/local/etc/squid --enable-icmp
```

การใช้งานใน Option –prefix ดังกล่าวเป็นการบอกไดเรกทอรีที่ Squid จะทำการติดตั้ง ส่วน –enable-icmp จะเป็นการบอกให้ Squid สนับสนุนการใช้ ICMP ด้วย และยังมี Option อื่นๆอีกมาก ซึ่งสามารถดูรายละเอียดเพิ่มเติมได้โดยใช้คำสั่ง ./configure – help

3.3.1.4. ทำการติดตั้งโปรแกรมโดยใช้คำสั่งดังต่อไปนี้

```
# make all
# make install
# make install-pinger
```

3.3.1.5. หลังจากทำขั้นตอนข้างต้นทั้งหมด จะได้โปแกรม Squid ติดตั้งลงในไดเรกทรอรี /usr/local/etc/squid จะมีไดเรกทอรีย่อยคือ

- bin สำหรับเก็บตัวโปรแกรม มีไฟล์ เช่น squid, pinger
- etc สำหรับเก็บ Configuration File เช่น squid.conf
- log สำหรับเก็บ Log File แสดงรายละเอียดการทำงาน
- cache สำหรับเก็บ Cache เราอาจ Link ไดเรกทอรีนี้ไปยังที่อื่นได้ ควรเผื่อขนาดของไดเรกทอรีนี้ไว้อย่างต่ำ 800 Mb สำหรับหน่วยงานขนาดเล็ก

3.3.2. การเชื่อมต่อ พรอกซี่เซิร์ฟเวอร์กับ Cache Server เครื่องอื่น

ตัวแปรcache_peer (คล้าย cache_peer ในเวอร์ชัน 1.x) ใช้ในการกำหนดความสัมพันธ์ ในการเชื่อต่อกับเครื่อง Cache Server เครื่องอื่น ซึ่งสามารถมีได้มากกว่าหนึ่งเครื่อง มีรูปแบบดังนี้

cache_peer hostname type http_port icp_port options

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- hostname คือชื่อหรือ IP ของเครื่อง Cache server ที่เราจะเชื่อมต่อด้วยถ้าเป็นชื่อจะต้องมีทั้ง forward และ reverse name หมายความว่า ถ้ารู้ IP สามารถระบุเป็นชื่อได้และถ้ารู้ชื่อก็สามารถระบุ IP ได้เช่นกัน
- type มีค่าได้เป็น parent, sibling, multicast
- http_port คือ port ที่ทำการส่งข้อมูล HTTP ของเครื่อง Proxy เซิร์ฟเวอร์
- icp_port คือ port ที่ใช้แลกเปลี่ยนข้อมูลระหว่างกันของเครื่อง Proxy เซิร์ฟเวอร์
- options จะระบุหรือไม่ก็ได้

3.3.3. การกำหนด Local Domains

เป็นการกำหนดโดเมนหรือ IP ที่ไม่ต้องให้ Cache Server ไปถามข้อมูลจากเครื่องที่เป็น parent หรือ sibling ส่วนใหญ่ชื่อที่กำหนดจะเป็นโดเมนภายในองค์กรหรือชื่อโดเมนที่ Cache Server สามารถไปหาได้เร็วกว่าที่จะไปถามจากเครื่อง parent หรือ sibling ใดๆ เช่นภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังนี้(โดเมน kmitl.ac.th) ก็ไม่ควรจะให้วิ่งไปถาม parent หรือ sibling ที่ไหน ก็ควรกำหนด kmitl.ac.th ให้อยู่เป็น Local Domain ดังตัวอย่างนี้

```
acl local_domain dstdomain kmitl.ac.th
always_direct allow local_domain
```

3.3.4. การกำหนดขนาดและไดเรกทอรีที่จะใช้เก็บข้อมูล

ใช้คำว่า cache_dir แล้วตามด้วยไดเรกทอรี, ขนาด (MB), จำนวนไดเรกทอรีย่อยภายใน ตัวอย่างเช่น

```
cache_dir  /cache/disk1 500 16 256
cache_dir  /cache/disk2 500 16 256
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

4.1. การทดลองโมเดลที่ 1

4.1.1. จุดประสงค์การทดลอง

4.1.1.1. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรมเซิร์ฟเวอร์ DNS ได้

4.1.1.2. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ได้

4.1.1.3. สามารถใช้งานโปรแกรมทดสอบประสิทธิภาพการทำงานของเครื่องซิร์ฟเวอร์ได้

4.1.2. อุปกรณ์การทดลอง

4.1.2.1. เครื่องคอมพิวเตอร์ทำหน้าที่เซิร์ฟเวอร์ DNS	1	เครื่อง
4.1.2.2. เครื่องคอมพิวเตอร์ทำหน้าที่เว็บเซิร์ฟเวอร์	1	เครื่อง
4.1.2.3. เครื่องคอมพิวเตอร์ทำหน้าที่วัดประสิทธิภาพ	10	เครื่อง
4.1.2.4. ชุดโปรแกรม WebBench	1	ชุด

4.1.3. กล่าวนำ

เว็บเซิร์ฟเวอร์ เป็น เซิร์ฟเวอร์โปรแกรมประเภทหนึ่งที่ทำงานอยู่เบื้องหลังของบริการ World Wide Web โปรแกรมดังกล่าวจะรอรับสัญญาณร้องขอจากเครื่องลูกข่าย เช่น โปรแกรมบราวเซอร์ ซึ่งได้แก่ โปรแกรมเน็ตสเคป หรือไม่ก็เป็นโปรแกรม Internet Explorer เมื่อโปรแกรมเว็บเซิร์ฟเวอร์ ได้รับสัญญาณร้องขอเรียบร้อย โปรแกรมจะจัดส่งข้อมูลบางประเภทไปยังเครื่องลูกข่ายที่ได้ส่งสัญญาณร้องขอเข้ามา ข้อมูลดังกล่าวอาจเป็นข้อมูลประเภทเท็กซ์ หรือว่าจะเป็นข้อมูลประเภทไฟล์กราฟิก เมื่อโปรแกรมเว็บบราวเซอร์ได้รับข้อมูลแล้วเรียบเรียงข้อมูลที่ได้รับมาจากนั้นนำเสนอให้กับผู้ใช้งานต่อไป ในเชิงคอนเซ็ปต์แล้วโปรแกรมเว็บเซิร์ฟเวอร์เป็นโปรแกรมที่มีลักษณะการทำงานที่ง่ายไม่ซับซ้อนอะไรคือ รอรับสัญญาณร้องขอ แล้วส่งกลับข้อมูลที่มีการร้องขอเข้าเข้ามาไปยังเครื่องลูกข่ายอย่างถูกต้อง

โปรแกรมเว็บเซิร์ฟเวอร์จะติดต่อกับเครื่องลูกข่ายต่างๆหรือติดต่อกับโปรแกรมเว็บบราวเซอร์ด้วยโพรโตคอล Hypertext Transfer Protocol (HTTP) ซึ่งโพรโตคอล ดังกล่าวเป็นโพรโตคอลที่มีลักษณะโครงสร้างที่ง่าย และเป็นมาตรฐานในตัวเอง จึงตัดปัญหาความที่เข้ากัน(การทำงาน)ได้ของแต่ละบริษัทผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์

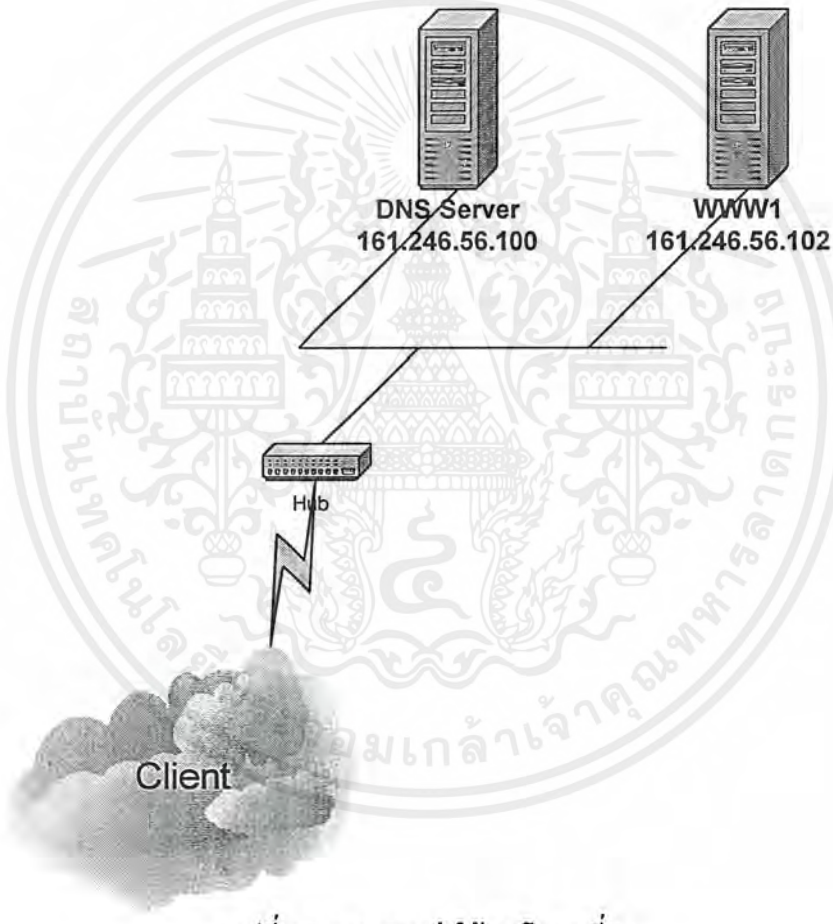
เอกสารส่วนมากที่ใช้ในการติดต่อสื่อสารกันส่วนมากจะอยู่ในรูปแบบของภาษา HTML (Hypertext Markup Language) อันเป็นส่วนย่อยของภาษาหนึ่ง ซึ่งมีชื่อเรียกว่า SGML (Standard General Markup Language) ซึ่งใช้งานส่วนอย่างแพร่หลายในองค์กรต่างๆ ของสหรัฐ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HTML เป็นเอกสารรูปแบบหลักที่ใช้งานในเว็บ HTML มีลักษณะเด่นหลายประการคือ ง่ายในการจัดเรียงและเหมาะสมสำหรับการสร้างรูปแบบของเอกสารในลักษณะเป็นเท็กซ์ โปรแกรมบราวเซอร์จะแปลงเอกสารในรูปแบบ HTML ก่อนแล้วจึงเรียบเรียงอีกทีหนึ่ง ลักษณะเด่นอีกประการหนึ่งของ HTML คือ อนุญาตให้มีการทำลิงก์เชื่อมต่อกันระหว่างเอกสารต่างๆได้

HTML อนุญาตให้ผู้ใช้งานสามารถอ้างอิงถึงเอกสารอื่นที่เก็บไว้ในเครื่องคอมพิวเตอร์อื่นได้และ HTML อนุญาตให้เอกสารหรือข้อมูลมีลักษณะดูคล้ายมีมิติได้ คือ เราไม่จำเป็นต้องอ่านแบบต่อเนื่องกันก็ได้ เราสามารถที่จะอ่านแบบกระโดดไปกระโดดมาก็ได้

ในการทดสอบใดๆ จะต้องมีการเปรียบเทียบกับสิ่งๆ หนึ่งที่เป็นสิ่งที่คงที่ ในที่นี้เราจะใช้โมเดลที่ 1 นี้เป็นโมเดลมาตรฐานในการที่จะทดสอบต่างๆ ต่อไป ในโมเดลนี้มีลักษณ์การต่อเครื่องคอมพิวเตอร์ดังรูปที่ 4.1



รูปที่ 4.1 แสดงการต่อใช้งานโมเดลที่ 1

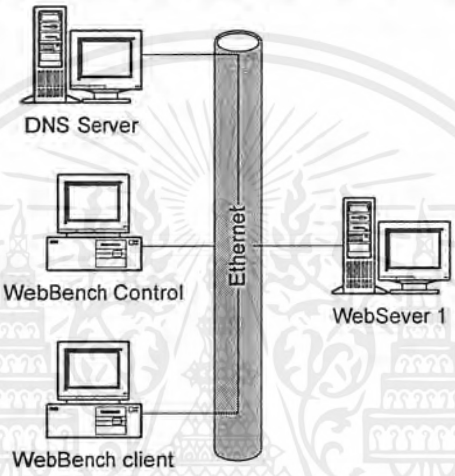
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4. ขั้นตอนการทดลอง

4.2.5.1. ติดตั้งระบบปฏิบัติการให้แก่เครื่องคอมพิวเตอร์ทั้งหมด โดยมีรายละเอียดดังนี้

- เครื่อง WebServer 1 ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง DNS Server ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง WebBench Control ใช้ระบบปฏิบัติการ Windows NT 4.0
- เครื่อง WebBench Client ใช้ระบบปฏิบัติการ Windows 95

4.2.5.2. ต่อเครื่องคอมพิวเตอร์ที่ได้จัดเตรียมไว้ให้เป็นไปตามเครือข่าย ดังรูป



รูปที่ 4.2 แสดงโมเดลมาตรฐาน

4.2.5.3. แก้ไขฐานข้อมูลในเครื่อง เซิร์ฟเวอร์ DNSในไฟล์ /var/named/master/ce.kmitl.ac.th ให้มีข้อมูลเพิ่มเติม ดังนี้

```
model1          A          161.246.56.101
```

4.2.5.4. สั่งรันโปรแกรม WebBench Controller ที่เครื่อง WebBench Control

4.2.5.5. สั่งรันโปรแกรม WebBench Client ที่เครื่อง WebBench Client

4.2.5.6. เมื่อทั้งเครื่อง Control และ Client สามารถติดต่อกันได้ จากนั้นสั่งเริ่มทดสอบจากเครื่อง WebBench Control ในการทดสอบนั้นจะใช้เครื่อง Client จำนวน 40 เครื่อง การทดสอบเริ่มจาก 1 Client แล้วเพิ่มทีล่ะ 4 เครื่องไปจนถึง 40 เครื่อง

4.2.5.7. เมื่อได้ผลการทดสอบให้บันทึกผลการวัดประสิทธิภาพไว้ โดยการอ่านค่าเวลาเฉลี่ยที่ใช้ในการทำงานนั้นๆ จากล็อกไฟล์ของโปรแกรมเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองโมเดลที่ 2

4.2.2. จุดประสงค์การทดลอง

4.2.2.1. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรมเซิร์ฟเวอร์ DNS ได้

4.2.2.2. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ได้

4.2.2.3. สามารถใช้งานโปรแกรมทดสอบประสิทธิภาพการทำงานของเครื่องเซิร์ฟเวอร์ได้

4.2.2.4. สามารถประยุกต์นำเอาเซิร์ฟเวอร์ DNS เข้ามาช่วยในการทำบาลานซ์โหลดให้แก่เซิร์ฟเวอร์

4.2.3. อุปกรณ์การทดลอง

4.2.3.1.	เครื่องคอมพิวเตอร์ทำหน้าที่เซิร์ฟเวอร์ DNS	1	เครื่อง
4.2.3.2.	เครื่องคอมพิวเตอร์ทำหน้าที่เว็บเซิร์ฟเวอร์	2	เครื่อง
4.2.3.3.	เครื่องคอมพิวเตอร์ทำหน้าที่วัดประสิทธิภาพ	10	เครื่อง
4.2.3.4.	ชุดโปรแกรม WebBench	1	ชุด

4.2.4. กล่าวนำ

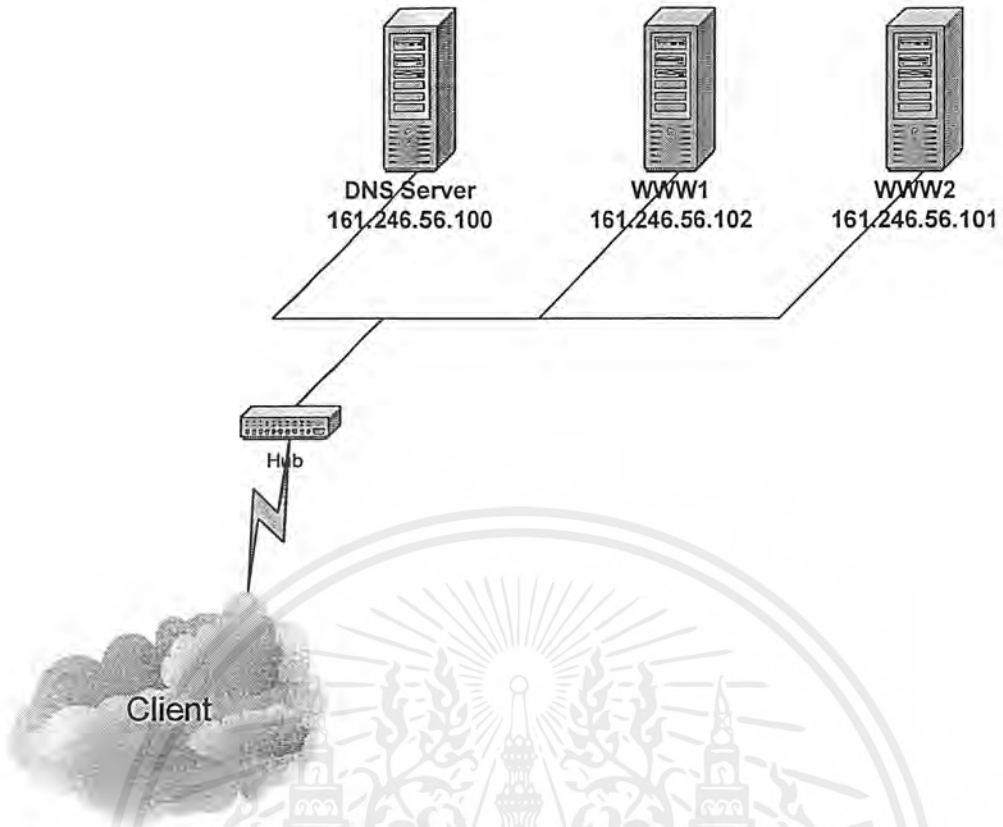
จากความต้องการที่จะเพิ่มประสิทธิภาพการทำงานให้กับเว็บไซต์ ให้สามารถที่จะรองรับจำนวนของผู้ร้องขอเข้ามาให้ได้มากที่สุด ซึ่งปรกติตามความเข้าใจของคนทั่วไปคือ หนึ่งเว็บไซต์ก็จะมีเครื่องเซิร์ฟเวอร์ทำงานรองรับเว็บไซต์นั้นอยู่เพียงหนึ่งเครื่อง แต่เมื่อเว็บไซต์เป็นที่นิยมของคนทั่วไปแล้ว จะมีสัญญาณร้องขอเข้ามาอย่างมากมาย เกินที่เครื่องเซิร์ฟเวอร์เครื่องเดียวจะรับได้ ซึ่งจำเป็นต้องนำเทคนิคการทำบาลานซ์โหลด มาช่วยให้เว็บไซต์ หรือไม่เช่นนั้นก็ต้อง เปลี่ยนเครื่องเซิร์ฟเวอร์ให้เป็นเครื่องที่มีประสิทธิภาพที่สูงกว่าเครื่องเดิม

จากที่ได้กล่าวมาข้างต้นนั้น เทคนิคหนึ่งที่ง่ายที่สุด และเป็นที่นิยมใช้งานกันมากที่สุดคือ การประยุกต์นำเอาเครื่องเซิร์ฟเวอร์ DNS มาช่วยในการทำบาลานซ์โหลด โดยหลักการมีอยู่คือ นำเอาเครื่องเซิร์ฟเวอร์อีกเครื่องหนึ่งหรือมากกว่ามาต่อร่วมด้วย เพื่อทำงานช่วยเหลือเซิร์ฟเวอร์ตัวแรก ดังแสดงในรูป

จากความต้องการที่จะเพิ่มประสิทธิภาพการทำงานให้กับเว็บไซต์ ให้สามารถที่จะรองรับจำนวนของผู้ร้องขอเข้ามาให้ได้มากที่สุด ซึ่งปรกติตามความเข้าใจของคนทั่วไปคือ หนึ่งเว็บไซต์ก็จะมีเครื่องเซิร์ฟเวอร์ทำงานรองรับเว็บไซต์นั้นอยู่เพียงหนึ่งเครื่อง แต่เมื่อเว็บไซต์เป็นที่นิยมของคนทั่วไปแล้ว จะมีสัญญาณร้องขอเข้ามาอย่างมากมาย เกินที่เครื่องเซิร์ฟเวอร์เครื่องเดียวจะรับได้ ซึ่งจำเป็นต้องนำเทคนิคการทำบาลานซ์โหลด มาช่วยให้เว็บไซต์ หรือไม่เช่นนั้นก็ต้อง เปลี่ยนเครื่องเซิร์ฟเวอร์ให้เป็นเครื่องที่มีประสิทธิภาพที่สูงกว่าเครื่องเดิม

จากที่ได้กล่าวมาข้างต้นนั้น เทคนิคหนึ่งที่ง่ายที่สุด และเป็นที่นิยมใช้งานกันมากที่สุดคือ การประยุกต์นำเอาเครื่องเซิร์ฟเวอร์ DNS มาช่วยในการทำบาลานซ์โหลด โดยหลักการมีอยู่คือ นำเอาเครื่องเซิร์ฟเวอร์อีกเครื่องหนึ่งหรือมากกว่ามาต่อร่วมด้วย เพื่อทำงานช่วยเหลือเซิร์ฟเวอร์ตัวแรก ดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงการต่อระบบในโมเดลที่ 2

จากรูปเมื่อมีสัญญาณร้องของจากภายนอกเข้ามาที่ เซิร์ฟเวอร์ DNS ต้องกลับค่า IP ของเว็บเซิร์ฟเวอร์แต่ละตัวกลับไป โดยการที่จะตอบกลับ IP ของเว็บเซิร์ฟเวอร์ตัวไหนนั้น จะใช้หลักการแบบ "Round Robin" ซึ่งการเพิ่มประสิทธิภาพให้แก่ไซต์อินเตอร์เน็ตแบบนี้ จำเป็นที่จะต้องแก้ไขข้อมูลใน DNS Server

จากรูปเมื่อมีสัญญาณร้องของจากภายนอกเข้ามาที่ เซิร์ฟเวอร์ DNS ต้องกลับค่า IP ของเว็บเซิร์ฟเวอร์แต่ละตัวกลับไป โดยการที่จะตอบกลับ IP ของเว็บเซิร์ฟเวอร์ตัวไหนนั้น จะใช้หลักการแบบ "Round Robin" ซึ่งการเพิ่มประสิทธิภาพให้แก่ไซต์อินเตอร์เน็ตแบบนี้ จำเป็นที่จะต้องแก้ไขข้อมูลใน DNS Server

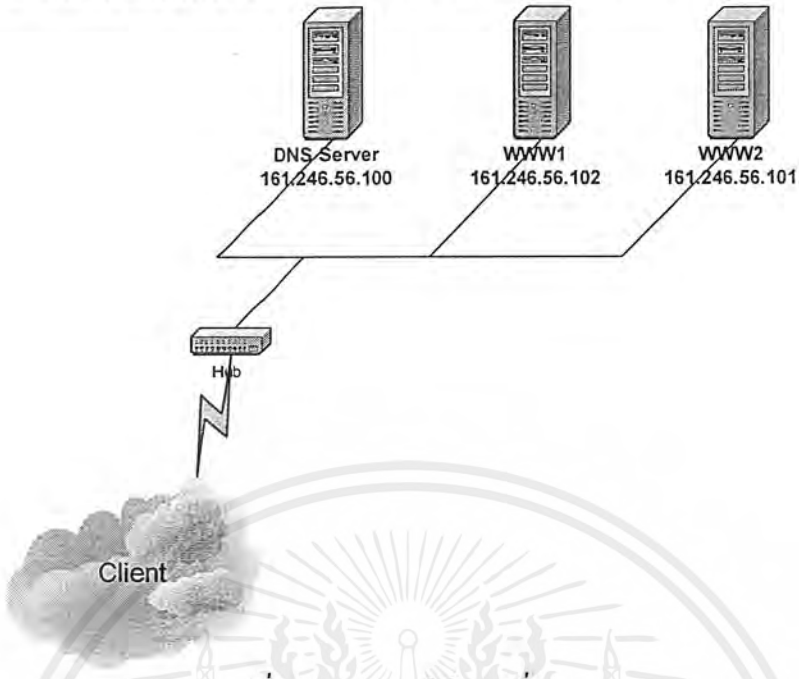
4.2.5. ขั้นตอนการทดลอง

4.2.5.1. ติดตั้งระบบปฏิบัติการให้แก่เครื่องคอมพิวเตอร์ทั้งหมด โดยมีรายละเอียดดังนี้

- เครื่อง WebServer 1 ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง WebServer 2 ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง DNS Server ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง WebBench Control ใช้ระบบปฏิบัติการ Windows NT 4.0
- เครื่อง WebBench Client ใช้ระบบปฏิบัติการ Windows 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5.2. ต่อเครื่องคอมพิวเตอร์ที่ได้จัดเตรียมไว้ให้เป็นไปตามเครือข่าย ดังรูป



รูปที่ 4.4 แสดงการต่อโมเดลที่ 2

4.2.5.3. แก้ไขฐานข้อมูลในเครื่อง เซิร์ฟเวอร์ DNSให้มีข้อมูลเพิ่มเติม ดังนี้

model2	A	161.246.56.101
	A	161.246.56.102

4.2.5.4. สั่งรันโปรแกรม WebBench Controller ที่เครื่อง WebBench Control

4.2.5.5. สั่งรันโปรแกรม WebBench Client ที่เครื่อง WebBench Client ในโมเดลนี้ใช้เครื่อง Client ในการทดสอบทั้งหมด 9 เครื่อง

4.2.5.6. เริ่มทดสอบ เมื่อได้ผลการทดสอบให้บันทึกผลการวัดประสิทธิภาพไว้ โดยการอ่านค่าเวลาเฉลี่ยที่ใช้ในการทำงานนั้นๆ จากล็อกไฟล์ของโปรแกรมเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3. การทดลองโมเดลที่ 3

4.3.1. จุดประสงค์การทดลอง

4.3.1.1. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรมเซิร์ฟเวอร์ DNSได้

4.3.1.2. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ได้

4.3.1.3. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรม พร็อกซี เซิร์ฟเวอร์ ได้

4.3.1.4. สามารถใช้งานโปรแกรมทดสอบประสิทธิภาพการทำงานของเครื่องเซิร์ฟเวอร์ได้

4.3.1.5. สามารถประยุกต์นำเอา เซิร์ฟเวอร์ DNS เข้ามาช่วยในการทำบาลานซ์โหลดให้แก่เซิร์ฟเวอร์

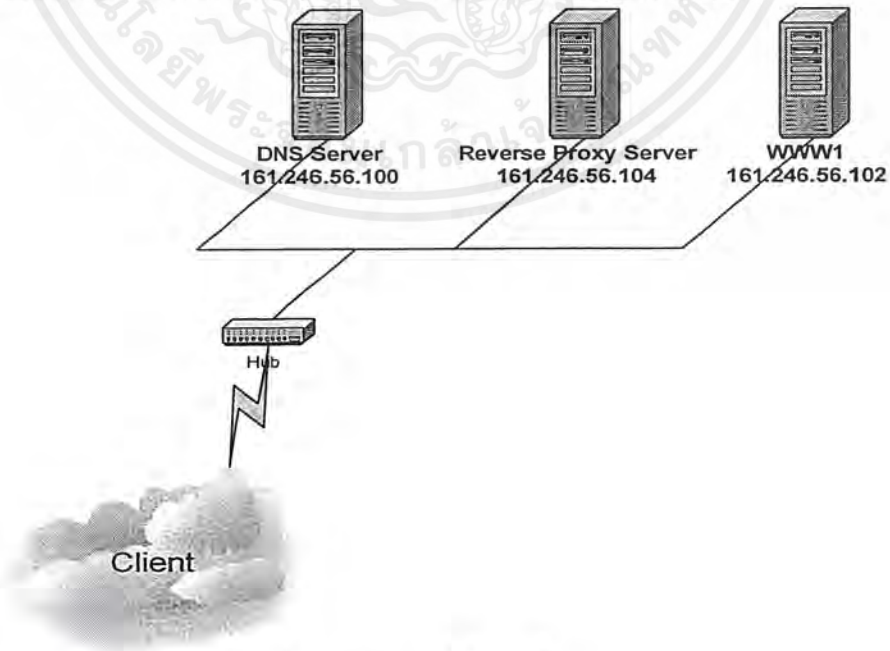
4.3.1.6. สามารถกำหนดค่าให้โปรแกรม พร็อกซี เซิร์ฟเวอร์ ทำงานในโหมด รีเวิร์ส พร็อกซี ได้

4.3.2. อุปกรณ์การทดลอง

4.3.2.1. เครื่องคอมพิวเตอร์ทำหน้าที่ เซิร์ฟเวอร์ DNS	1	เครื่อง
4.3.2.2. เครื่องคอมพิวเตอร์ทำหน้าที่ เว็บเซิร์ฟเวอร์	1	เครื่อง
4.3.2.3. เครื่องคอมพิวเตอร์ทำหน้าที่ รีเวิร์ส พร็อกซี เซิร์ฟเวอร์	1	เครื่อง
4.3.2.4. เครื่องคอมพิวเตอร์ทำหน้าที่วัดประสิทธิภาพ	10	เครื่อง
4.3.2.5. ชุดโปรแกรม WebBench	1	ชุด

4.3.3. กล่าวนำ

ในโมเดลนี้ใช้ความสามารถพิเศษของ พร็อกซี เซิร์ฟเวอร์ เข้าช่วย เป็นการกำหนดให้ พร็อกซี เซิร์ฟเวอร์ ทำงานในโหมดที่เรียกว่า รีเวิร์ส พร็อกซี มีลักษณะของการจัดรูปแบบดังรูป



รูปที่ 4.5 แสดงการต่อโมเดลที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

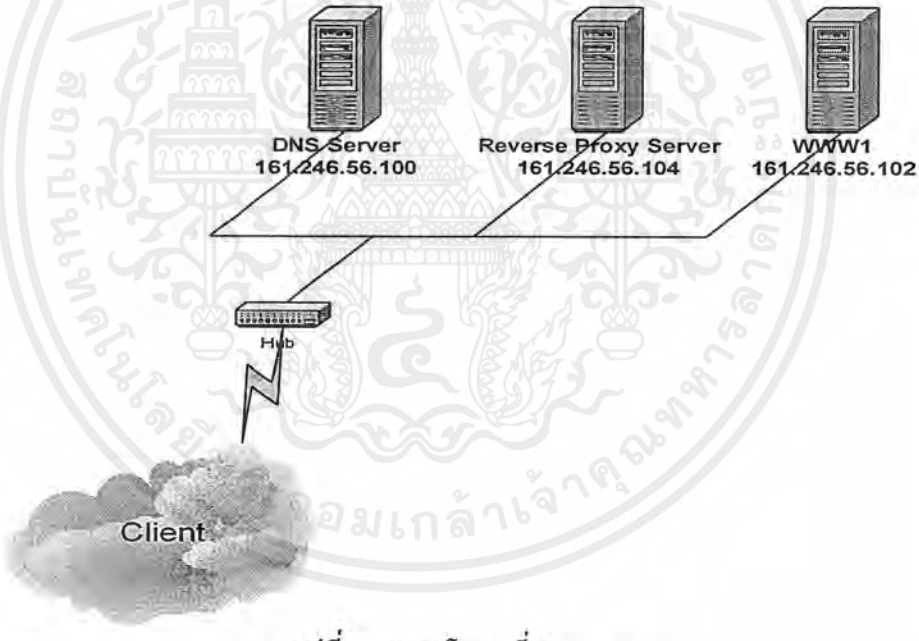
จะสังเกตได้ว่าในแบบนี้หน้าหลักของโดเมนเนมเซิร์ฟเวอร์จะเป็นแค่โดเมนเนมธรรมดา แต่ตัวที่จะ Redirect หรือตัวเลือกว่าจะเอาข้อมูลจากเซิร์ฟเวอร์ตัวไหน เป็นหน้าที่ พร็อกซี ซึ่งถ้าข้อมูลที่ต้องการมีอยู่ใน พร็อกซี อยู่แล้วก็ไม่จำเป็นที่จะต้องไปเอาที่เซิร์ฟเวอร์ ข้อดีของ รีเวิร์ส พร็อกซี คือ เวลาที่ใช้ในการดึงข้อมูลออก ถ้าเทียบกับโปรแกรมเว็บเซิร์ฟเวอร์แล้ว พร็อกซี เซิร์ฟเวอร์ มีความเร็วกว่า

4.3.4. ขั้นตอนการทดลอง

4.3.4.1. ติดตั้งระบบปฏิบัติการให้แก่เครื่องคอมพิวเตอร์ทั้งหมด โดยมีรายละเอียดดังนี้

- เครื่อง WebServer 1 ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง รีเวิร์ส พร็อกซี ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง DNS Server ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง WebBench Control ใช้ระบบปฏิบัติการ Windows NT 4.0
- เครื่อง WebBench Client ใช้ระบบปฏิบัติการ Windows 95

4.3.4.2. ต่อเครื่องคอมพิวเตอร์ที่ได้จัดเตรียมไว้ให้เป็นไปตามเครือข่าย ดังรูป



รูปที่ 4.6 แสดงโมเดลที่ 3

4.3.4.3. แก้ไขไฟล์ /var/named/master/ce.kmitl.ac.th ของเครื่อง เซิร์ฟเวอร์ DNSให้มีรายละเอียดดังนี้

```
model4          A       161.246.56.104
```

4.3.4.4. แก้ไขไฟล์ /usr/local/squid/etc/squid.conf ของเครื่อง รีเวิร์ส พร็อกซี ให้สามารถทำงานในโหมดของ รีเวิร์ส พร็อกซี ได้ โดยมีรายละเอียดการแก้ไขดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แก้ไขในส่วนของ http_port

```
# NETWORK OPTIONS
# ------------------------------------------------------------------------
# TAG: http_port
#       The port number where Squid will listen for HTTP client
#       requests.  Default is 3128, for httpd-accel mode use port 80.
#       May be overridden with -a on the command line.
#       You may specify multiple ports here, but they MUST all be on
#       a single line.
http_port 80
```

- แก้ไขในส่วนของ httpd-accelerator options

```
# HTTPD-ACCELERATOR OPTIONS
# ------------------------------------------------------------------------
# TAG: httpd_accel_host
# TAG: httpd_accel_port
#       If you want to run Squid as an httpd accelerator, define the
#       host name and port number where the real HTTP server is.
#
#       If you want virtual host support then specify the hostname
#       as "virtual".
#
#       NOTE: enabling httpd_accel_host disables พร็อกซี-caching and
#       ICP.  If you want these features enabled also, then set
#       the 'httpd_accel_with_พร็อกซี' option.
#
httpd_accel_host server1.ce.kmitl.ac.th
httpd_accel_port 80
# TAG: httpd_accel_with_พร็อกซี  on|off
#       If you want to use Squid as both a local httpd accelerator
#       and as a พร็อกซี, change this to 'on'.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#
httpd_accel_with_พร็อกซี on
# TAG: httpd_accel_uses_host_header    on|off
#     HTTP/1.1 requests include a Host: header which is basically the
#     hostname from the URL. Squid can be an accelerator for
#     different HTTP servers by looking at this header. However,
#     Squid does NOT check the value of the Host header, so it opens
#     a big security hole. We recommend that this option remain
#     disabled unless you are sure of what you are doing.
#
#     However, you will need to enable this option if you run Squid
#     as a transparent พร็อกซี. Otherwise, virtual servers which
#     require the Host: header will not be properly cached.
#     httpd_accel_uses_host_header on
```

4.3.4.5. สั่งรันโปรแกรม WebBench Controller ที่เครื่อง WebBench Control

4.3.4.6. สั่งรันโปรแกรม WebBench Client ที่เครื่อง WebBench Client ในโมเดลนี้ใช้เครื่อง Client ในการทดสอบทั้งหมด 9 เครื่อง

4.2.5.7. เริ่มทดสอบ เมื่อได้ผลการทดสอบให้บันทึกผลการวัดประสิทธิภาพไว้ โดยการอ่านค่าเวลาเฉลี่ยที่ใช้ในการทำงานนั้นๆ จากล็อกไฟล์ของโปรแกรม พร็อกซี เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4. การทดลองโมเดลที่ 4

4.4.1. จุดประสงค์การทดลอง

4.4.1.1. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรม เซิร์ฟเวอร์ DNSได้

4.4.1.2. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ได้

4.4.1.3. สามารถเข้าใจถึงขั้นตอนการทำงานของโปรแกรม พร็อกซี เซิร์ฟเวอร์ ได้

4.4.1.4. สามารถใช้งานโปรแกรมทดสอบประสิทธิภาพการทำงานของเครื่องเซิร์ฟเวอร์ได้

4.4.1.5. สามารถประยุกต์นำเอา เซิร์ฟเวอร์ DNSเข้ามาช่วยในการทำบาลานซ์โหลดให้แก่เซิร์ฟเวอร์

4.4.1.6. สามารถกำหนดค่าให้โปรแกรม พร็อกซี เซิร์ฟเวอร์ ทำงานในโหมด รีเวิร์ส พร็อกซี ได้

4.4.2. อุปกรณ์การทดลอง

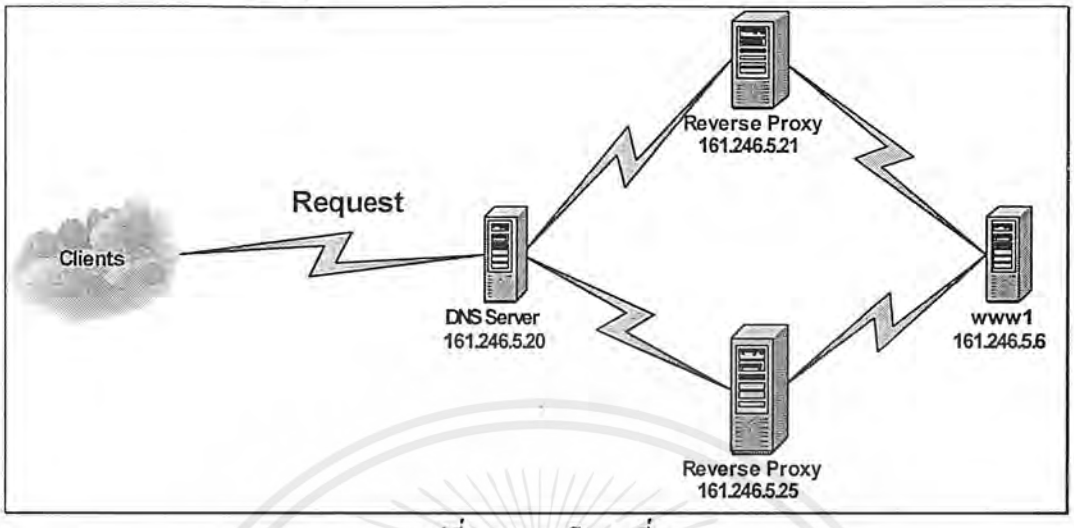
4.4.2.1. เครื่องคอมพิวเตอร์ทำหน้าที่ เซิร์ฟเวอร์ DNS	1	เครื่อง
4.4.2.2. เครื่องคอมพิวเตอร์ทำหน้าที่ เว็บเซิร์ฟเวอร์	1	เครื่อง
4.4.2.3. เครื่องคอมพิวเตอร์ทำหน้าที่ รีเวิร์ส พร็อกซี เซิร์ฟเวอร์	2	เครื่อง
4.4.2.4. เครื่องคอมพิวเตอร์ทำหน้าที่วัดประสิทธิภาพ	10	เครื่อง
4.4.2.5. ชุดโปรแกรม WebBench	1	ชุด

4.4.3. กล่าวนำ

จากที่ได้ศึกษาถึงความสามารถพิเศษของโปรแกรมพรอกซี่เซิร์ฟเวอร์ ที่สามารถจะทำงานในโหมด รีเวิร์ส พร็อกซี ได้ ซึ่งตามหลักการแล้ว ความเร็วในการดึกข้อมูลจากพรอกซี่ เปรียบเทียบกับการดึงข้อมูลโดยตรงจากเว็บเซิร์ฟเวอร์ การดึงข้อมูลจากพรอกซี่ จะให้ประสิทธิภาพ หรือความเร็วที่สูงกว่า เนื่องจากลักษณะการเก็บข้อมูลที่แตกต่างกันของตัวพรอกซี่เซิร์ฟเวอร์เองกับเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อดีดังกล่าวจึงได้ประยุกต์ออกแบบโมเดลใหม่ขึ้นมาโดยนำทั้งโมเดลที่ 2 และ โมเดลที่ 3 มาประสมกันได้โมเดลออกมาดังรูป



รูปที่ 4.7 แสดงโมเดลที่ 4

จากรูปที่ 4.7 เมื่อให้ระบบทำงานสักระยะหนึ่งเมื่อ รีเวิร์ส พร็อกซี มีข้อมูลมากพอ ก็จะเปรียบเสมือนว่าเรามีเว็บเซิร์ฟเวอร์ 2 ตัว

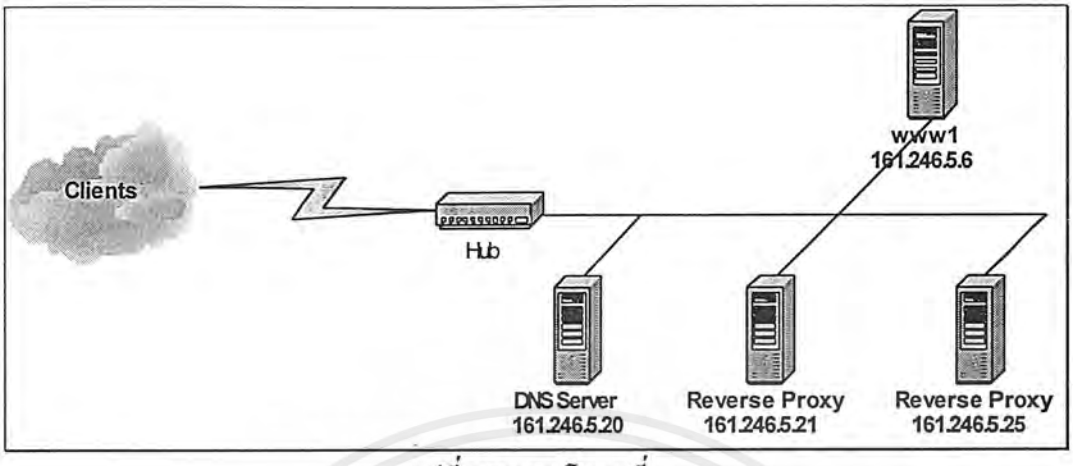
4.4.4. ขั้นตอนการทดลอง

4.4.4.1. ติดตั้งระบบปฏิบัติการให้แก่เครื่องคอมพิวเตอร์ทั้งหมด โดยมีรายละเอียดดังนี้

- เครื่อง WebServer 1 ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง รีเวิร์ส พร็อกซี1 ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง รีเวิร์ส พร็อกซี2 ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง DNS Server ใช้ระบบปฏิบัติการ Linux (Slackware 7.0)
- เครื่อง WebBench Control ใช้ระบบปฏิบัติการ Windows NT 4.0
- เครื่อง WebBench Client ใช้ระบบปฏิบัติการ Windows 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.4.2. ต่อเครื่องคอมพิวเตอร์ที่ได้จัดเตรียมไว้ให้เป็นไปตามเครือข่าย ดังรูป



รูปที่ 4.9 แสดงโมเดลที่ 4

4.4.4.2. แก้ไขไฟล์ /var/named/master/ce.kmitl.ac.th ให้มีรายละเอียดเพิ่มเติมดังนี้

```
model4          A       161.246.56.101
                A       161.246.56.104
```

4.4.4.3. usr/local/squid/etc/squid.conf ของเครื่อง รีเวิร์ส พร็อกซี (161.246.56.101) ให้ทำงานในโหมดของ รีเวิร์ส พร็อกซี ได้ โดยมีรายละเอียดการแก้ไขดังนี้

- แก้ไขในส่วนของ http_port

```
# NETWORK OPTIONS
# ------------------------------------------------------------------------
# TAG: http_port
#       The port number where Squid will listen for HTTP client
#       requests.  Default is 3128, for httpd-accel mode use port 80.
#       May be overridden with -a on the command line.
#
#       You may specify multiple ports here, but they MUST all be on
#       a single line.
http_port 80
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แก้ไขในส่วนของ httpd-accelerator options

```
# HTTPD-ACCELERATOR OPTIONS
# -----------------------------------------------------------------------------
# TAG: httpd_accel_host
# TAG: httpd_accel_port
#     If you want to run Squid as an httpd accelerator, define the
#     host name and port number where the real HTTP server is.
#
#     If you want virtual host support then specify the hostname
#     as "virtual".
#
#     NOTE: enabling httpd_accel_host disables พร็อกซี-caching and
#     ICP.  If you want these features enabled also, then set
#     the 'httpd_accel_with_พร็อกซี' option.
#
httpd_accel_host  server2.ce.kmitl.ac.th
httpd_accel_port 80
# TAG: httpd_accel_with_พร็อกซี  on|off
#     If you want to use Squid as both a local httpd accelerator
#     and as a พร็อกซี, change this to 'on'.
#
httpd_accel_with_พร็อกซี on
# TAG: httpd_accel_uses_host_header    on|off
#     HTTP/1.1 requests include a Host: header which is basically the
#     hostname from the URL.  Squid can be an accelerator for
#     different HTTP servers by looking at this header.  However,
#     Squid does NOT check the value of the Host header, so it opens
#     a big security hole.  We recommend that this option remain
#     disabled unless you are sure of what you are doing.
#     However, you will need to enable this option if you run Squid
#     as a transparent พร็อกซี.  Otherwise, virtual servers which
#     require the Host: header will not be properly cached.
httpd_accel_uses_host_header off
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

4.4.4.4. แก้ไขไฟล์ usr/local/squid/etc/squid.conf ของเครื่อง รีเวิร์ส พร็อกซี (161.246.56.104) ให้ทำงานในโหมดของ รีเวิร์ส พร็อกซี ได้ โดยมีรายละเอียดการแก้ไขดังนี้

- แก้ไขในส่วนของ http_port

```
# NETWORK OPTIONS
# ----------------------------------------------------------------------
# TAG: http_port
#     The port number where Squid will listen for HTTP client
#     requests.  Default is 3128, for httpd-accel mode use port 80.
#     May be overridden with -a on the command line.
#
#     You may specify multiple ports here, but they MUST all be on
#     a single line.
http_port 80
```

- แก้ไขในส่วนของ httpd-accelerator options

```
# HTTPD-ACCELERATOR OPTIONS
# ----------------------------------------------------------------------
# TAG: httpd_accel_host
# TAG: httpd_accel_port
#     If you want to run Squid as an httpd accelerator, define the
#     host name and port number where the real HTTP server is.
#
#     If you want virtual host support then specify the hostname
#     as "virtual".
#
#     NOTE: enabling httpd_accel_host disables พร็อกซี-caching and
#     ICP.  If you want these features enabled also, then set
#     the 'httpd_accel_with_พร็อกซี' option.
#
httpd_accel_host            server4.ce.kmitl.ac.th
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
httpd_accel_port 80
# TAG: httpd_accel_with_พร็อกซี  on|off
#     If you want to use Squid as both a local httpd accelerator
#     and as a พร็อกซี, change this to 'on'.
#
httpd_accel_with_พร็อกซี on
# TAG: httpd_accel_uses_host_header    on|off
#     HTTP/1.1 requests include a Host: header which is basically the
#     hostname from the URL.  Squid can be an accelerator for
#     different HTTP servers by looking at this header.  However,
#     Squid does NOT check the value of the Host header, so it opens
#     a big security hole.  We recommend that this option remain
#     disabled unless you are sure of what you are doing.
#
#     However, you will need to enable this option if you run Squid
#     as a transparent พร็อกซี.  Otherwise, virtual servers which
#     require the Host: header will not be properly cached.
httpd_accel_uses_host_header off
```

4.4.4.5. สั่งรันโปรแกรม WebBench Controller ที่เครื่อง WebBench Control

4.4.4.6. สั่งรันโปรแกรม WebBench Client ที่เครื่อง WebBench Client

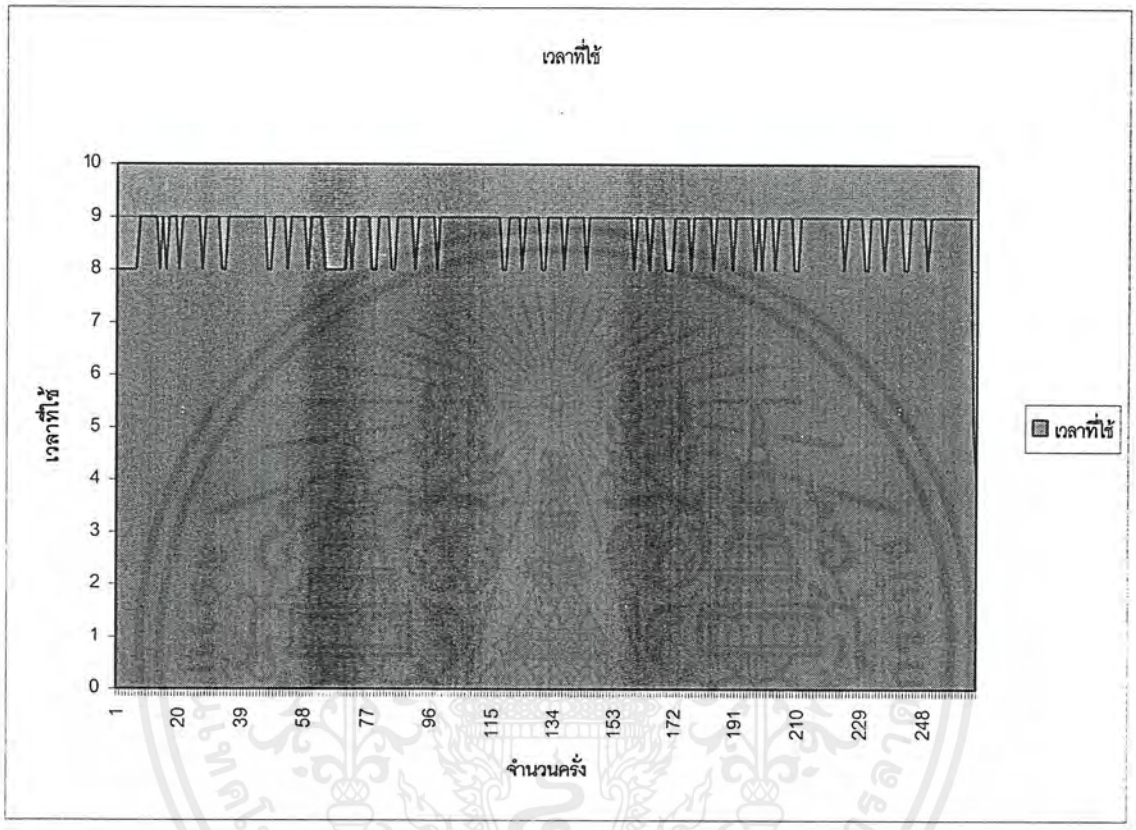
4.4.4.7. เริ่มทดสอบ เมื่อได้ผลการทดสอบให้บันทึกผลการวัดประสิทธิภาพไว้ โดยการอ่านค่าเวลาเฉลี่ยที่ใช้ในการทำงานนั้นๆ จากล็อกไฟล์ของโปรแกรม พร็อกซี เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

5.1. ผลการทดลองจากโมเดลที่ 1

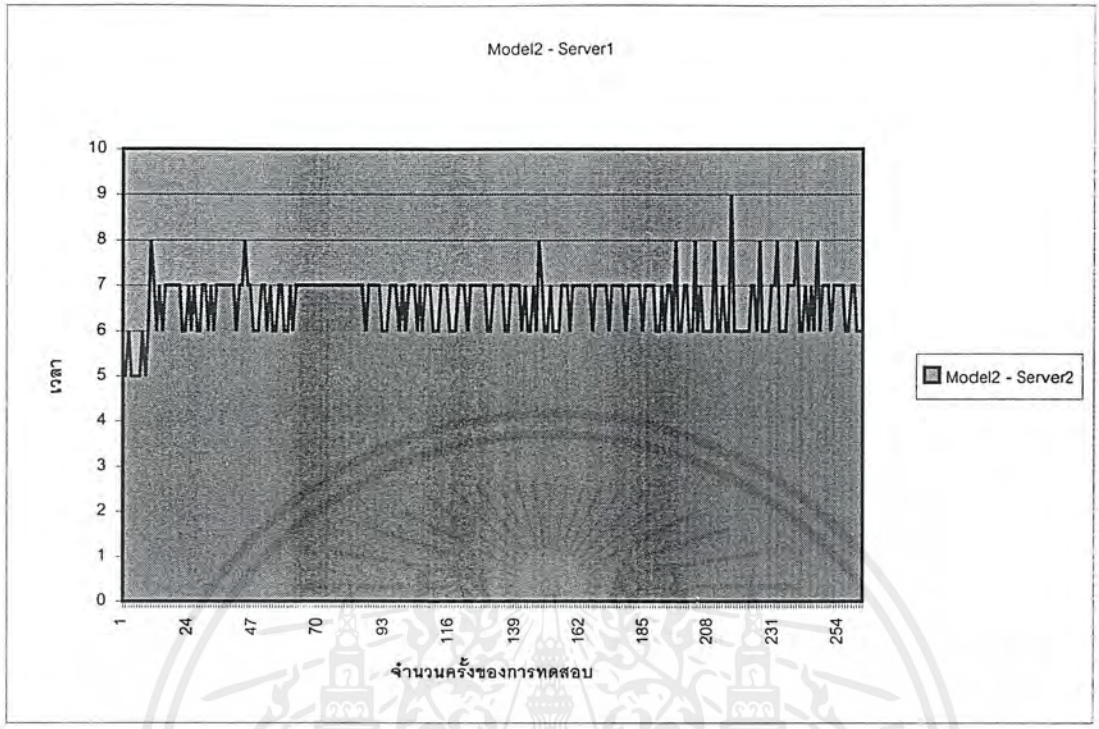


รูปที่ 5.1 แสดงผลการทดลองของโมเดลที่ 1

เวลาเฉลี่ยที่ใช้ = 8.71 วินาที

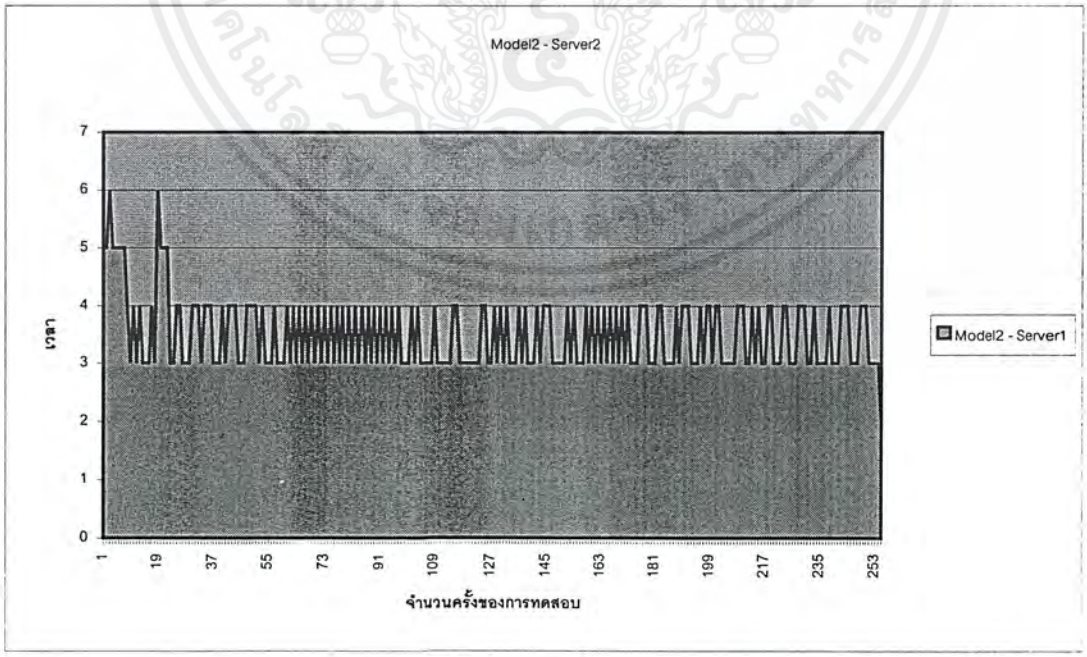
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2. ผลการทดลองจากโมเดลที่ 2



รูปที่ 5.2 แสดงผลการทดลองของโมเดลที่ 2จากเซิร์ฟเวอร์ตัวที่ 1

เวลาเฉลี่ยที่ใช้ = 6.6 วินาที



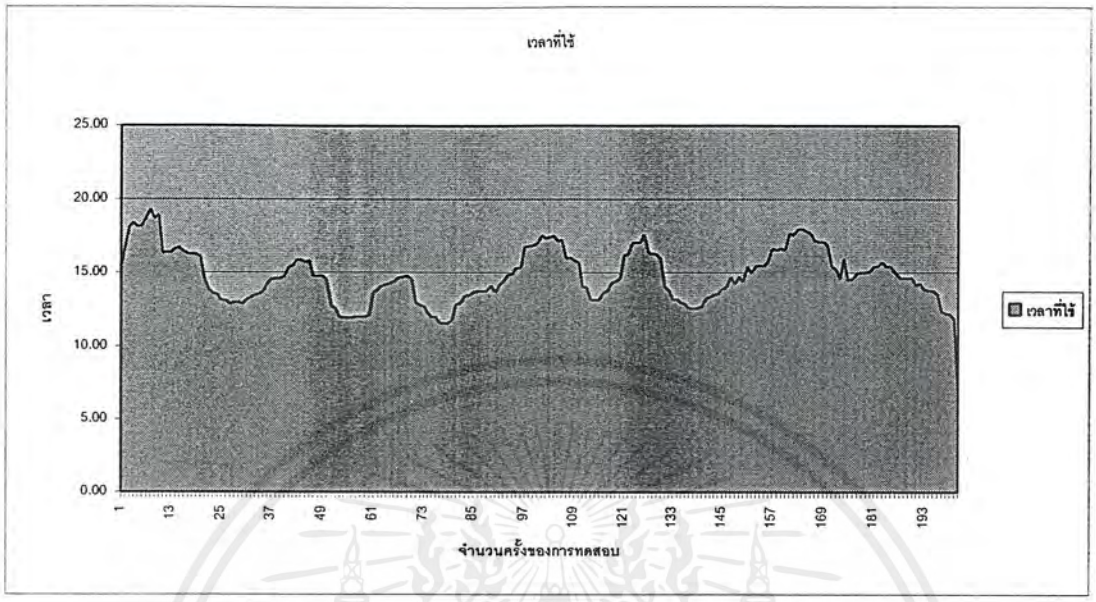
รูปที่ 5.3 แสดงผลการทดลองของโมเดลที่2 จากเซิร์ฟเวอร์ตัวที่ 2

เวลาเฉลี่ยที่ใช้ = 3.49 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3. ผลการทดลองจากโมเดลที่ 3

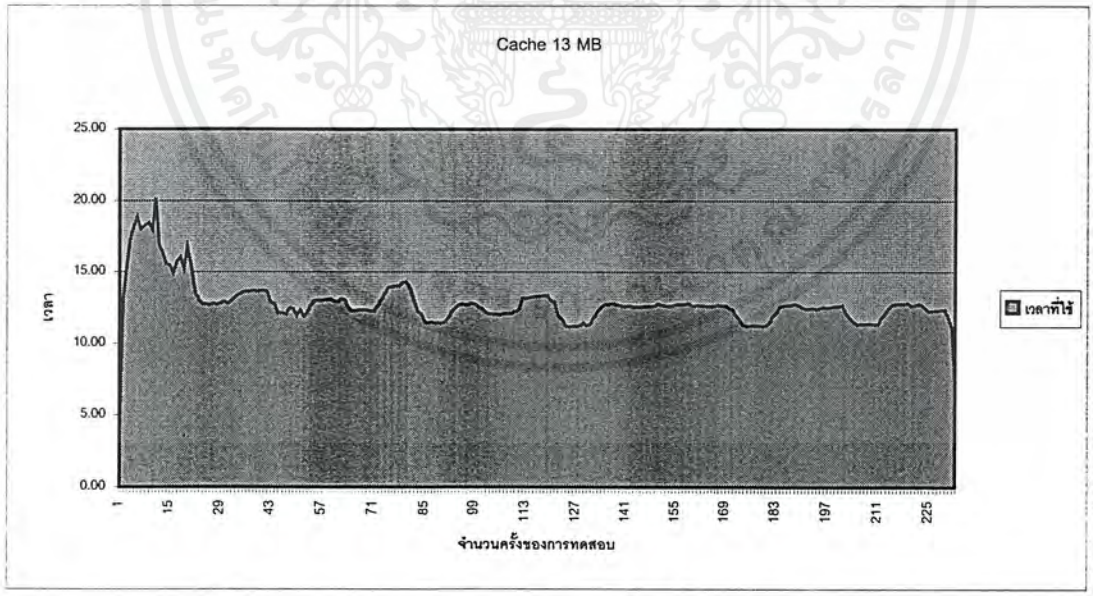
Cache 10 MB



รูปที่ 5.4 แสดงผลการทดลองของโมเดลที่ 3 ใช้ Cache ขนาด 10 MB

เวลาเฉลี่ยที่ใช้ = 14.76 วินาที

Cache 13 MB

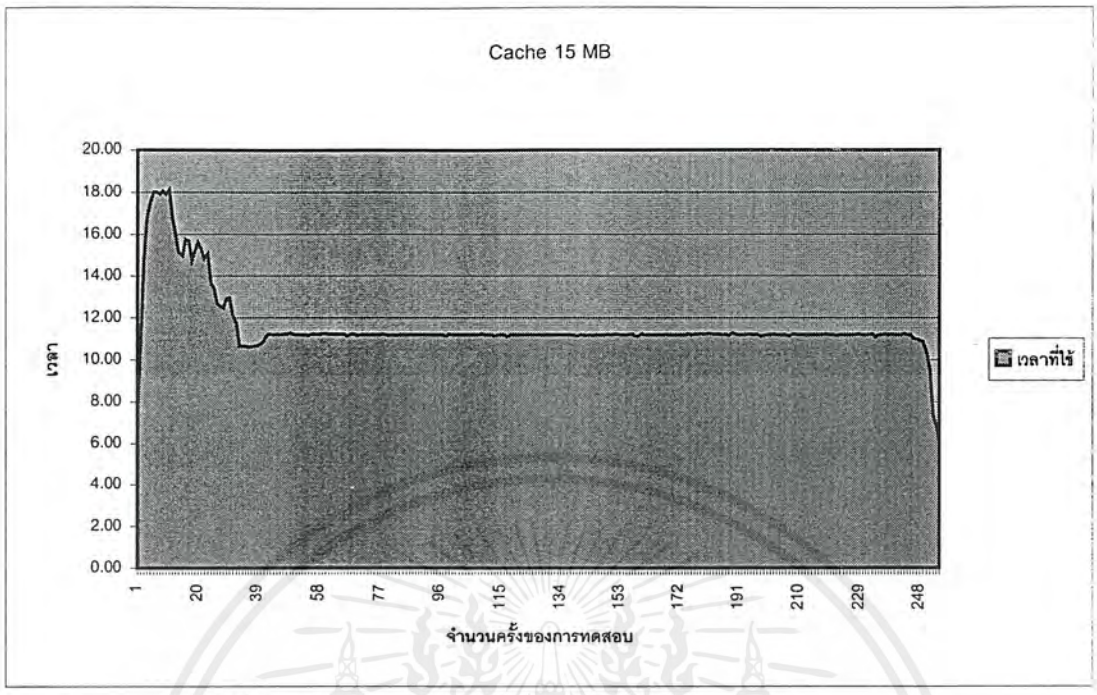


รูปที่ 5.5 แสดงผลการทดลองของโมเดลที่ 3 ใช้ Cache ขนาด 13 MB

เวลาเฉลี่ยที่ใช้ = 12.78 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

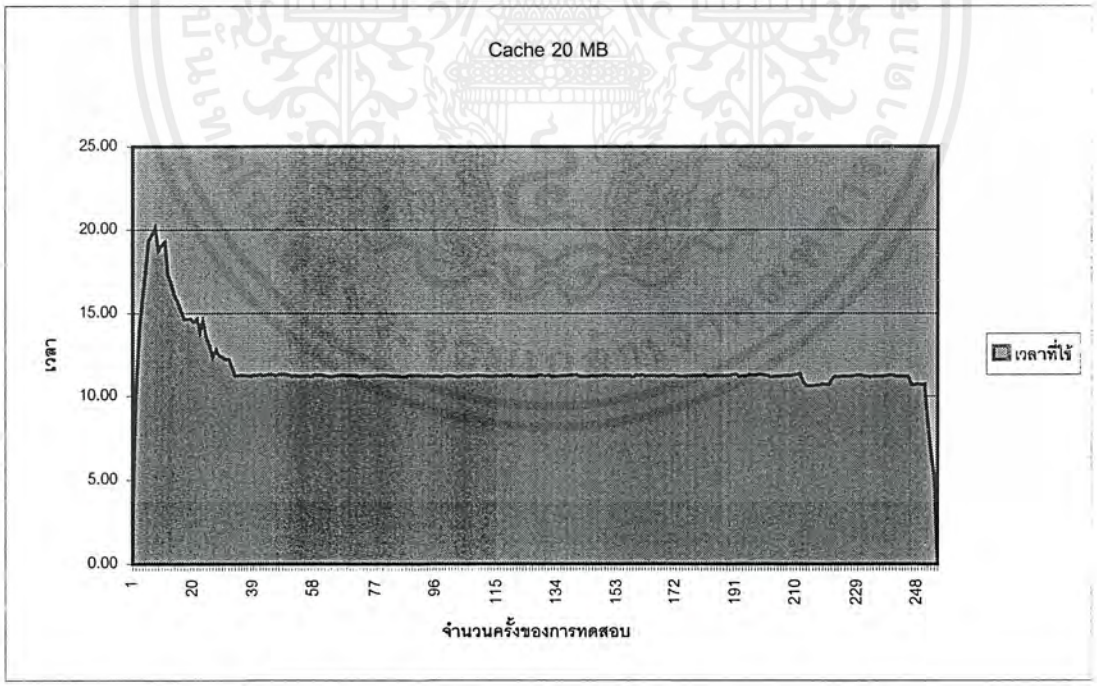
Cache 15 MB



รูปที่ 5.6 แสดงผลการทดลองของโมเดลที่ 3 ใช้ Cache ขนาด 15 MB

เวลาเฉลี่ยที่ใช้ = 11.58 วินาที

Cache 20 MB



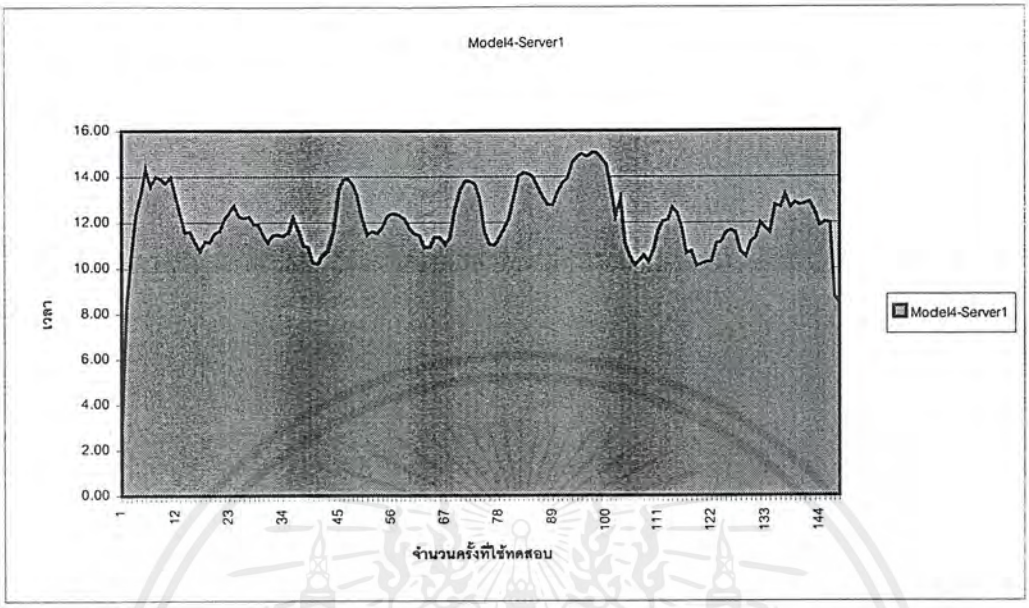
รูปที่ 5.7 แสดงผลการทดลองของโมเดลที่ 3 ใช้ Cache ขนาด 20 MB

เวลาเฉลี่ยที่ใช้ = 11.59 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4. ผลการทดลองจากโมเดลที่ 4

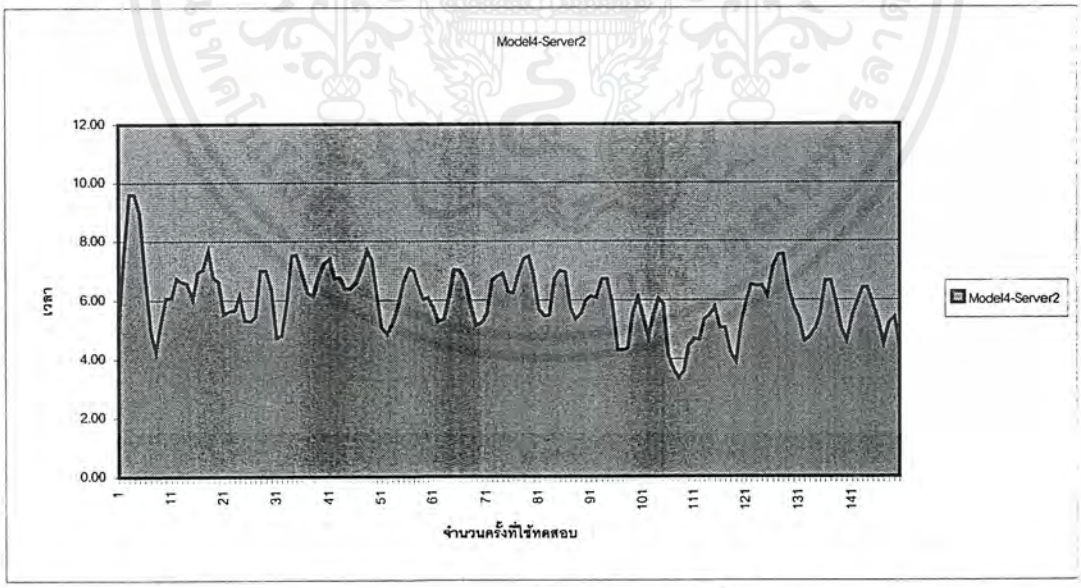
Cache 5 MB (พร็อกซี เซิร์ฟเวอร์1)



รูปที่ 5.8 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 1 ใช้ Cache ขนาด 5 MB

เวลาเฉลี่ยที่ใช้ = 12.05 วินาที

Cache 5 MB (พร็อกซี เซิร์ฟเวอร์2)

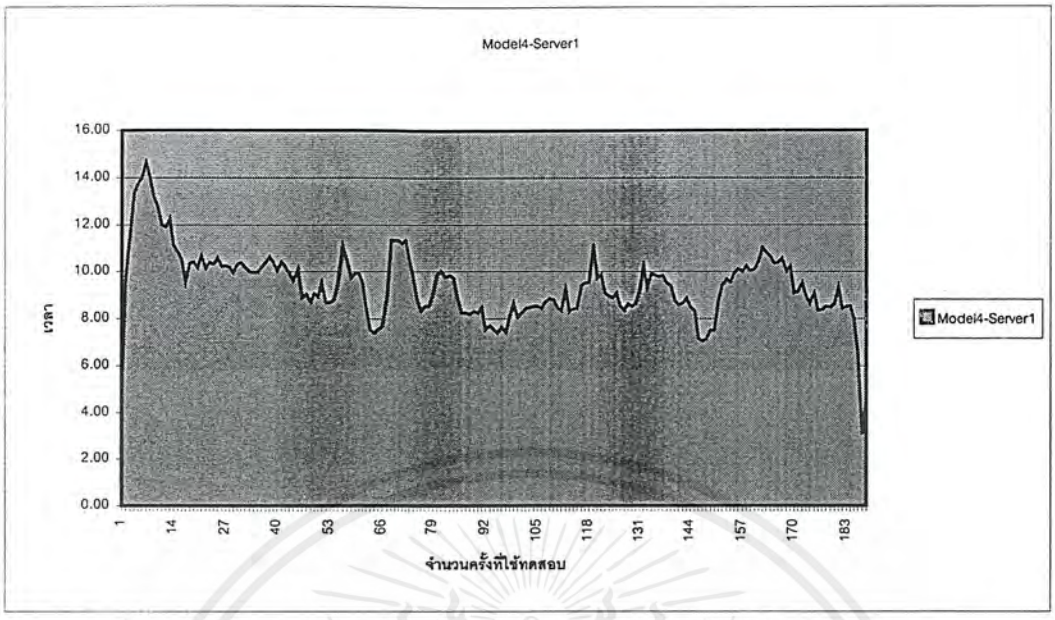


รูปที่ 5.9 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 2 ใช้ Cache ขนาด 5 MB

เวลาเฉลี่ยที่ใช้ = 6.0 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

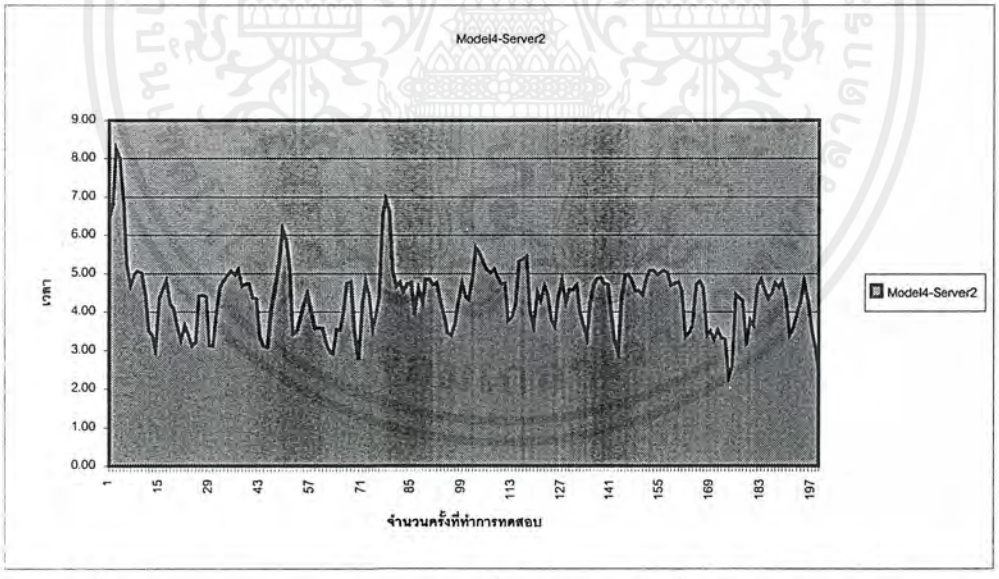
Cache 10 MB (พร็อกซี เซิร์ฟเวอร์1)



รูปที่ 5.10 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 1 ใช้ Cache ขนาด 10 MB

เวลาเฉลี่ยที่ใช้ = 9.42 วินาที

Cache 10 MB (พร็อกซี เซิร์ฟเวอร์2)

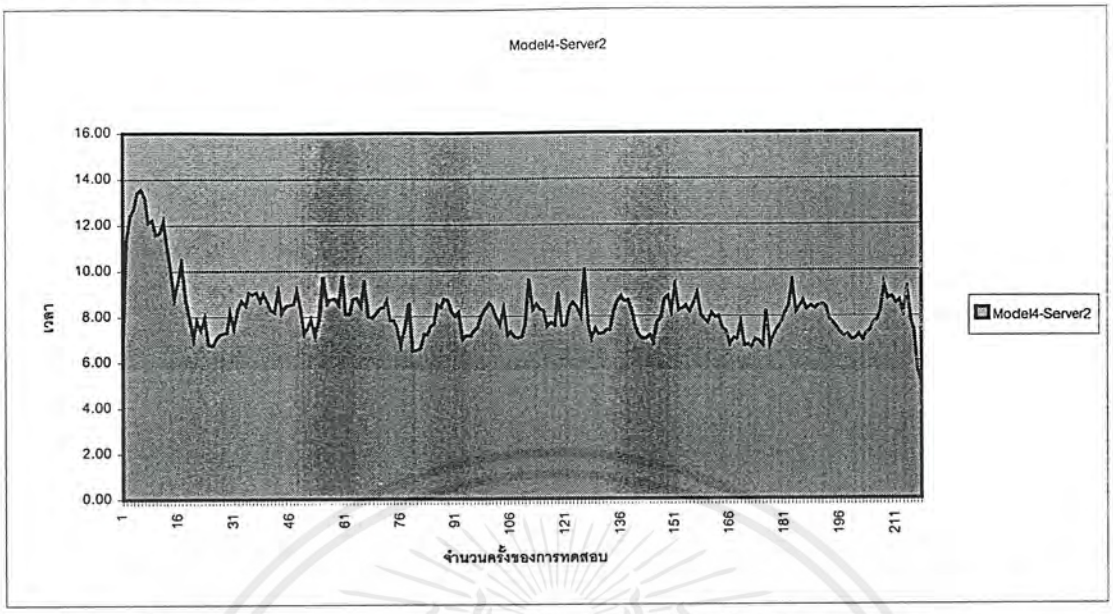


รูปที่ 5.11 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 2 ใช้ Cache ขนาด 10 MB

เวลาเฉลี่ยที่ใช้ = 4.38 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

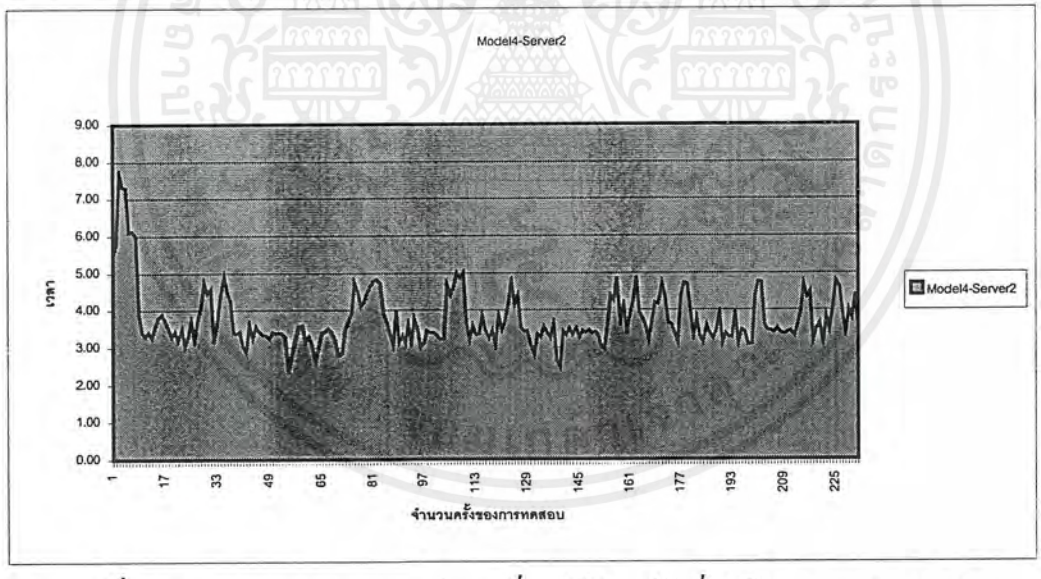
Cache 13 MB (พร็อกซี เซิร์ฟเวอร์1)



รูปที่ 5.12 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 1 ใช้ Cache ขนาด 13 MB

เวลาเฉลี่ยที่ใช้ = 8.19 วินาที

Cache 13 MB (พร็อกซี เซิร์ฟเวอร์2)

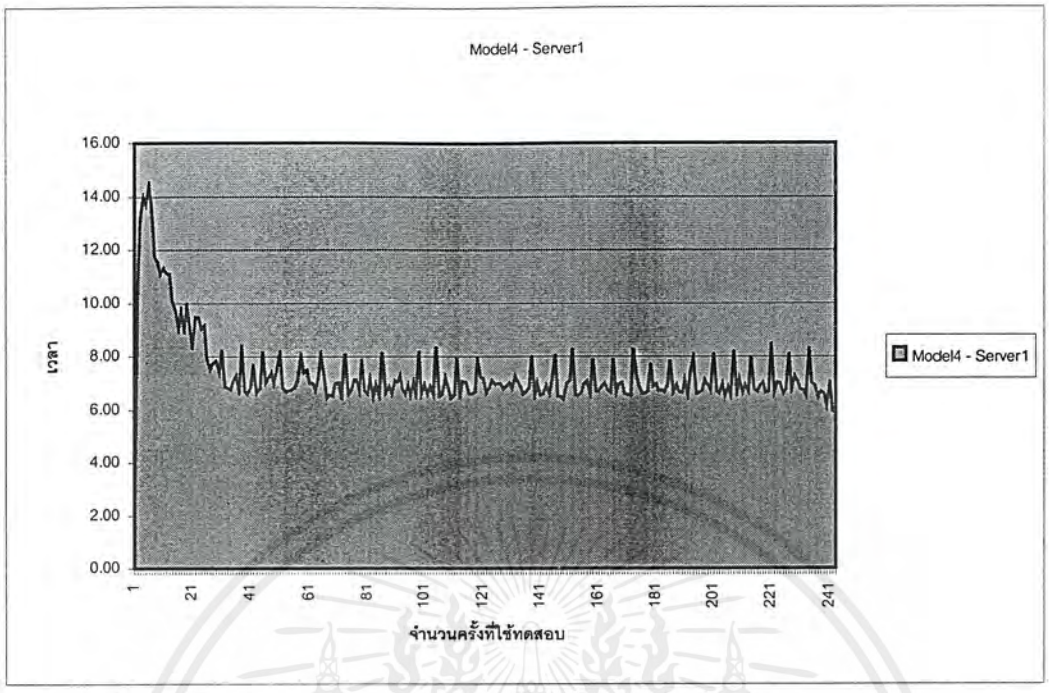


รูปที่ 5.13 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 2 ใช้ Cache ขนาด 13 MB

เวลาเฉลี่ยที่ใช้ = 3.75 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

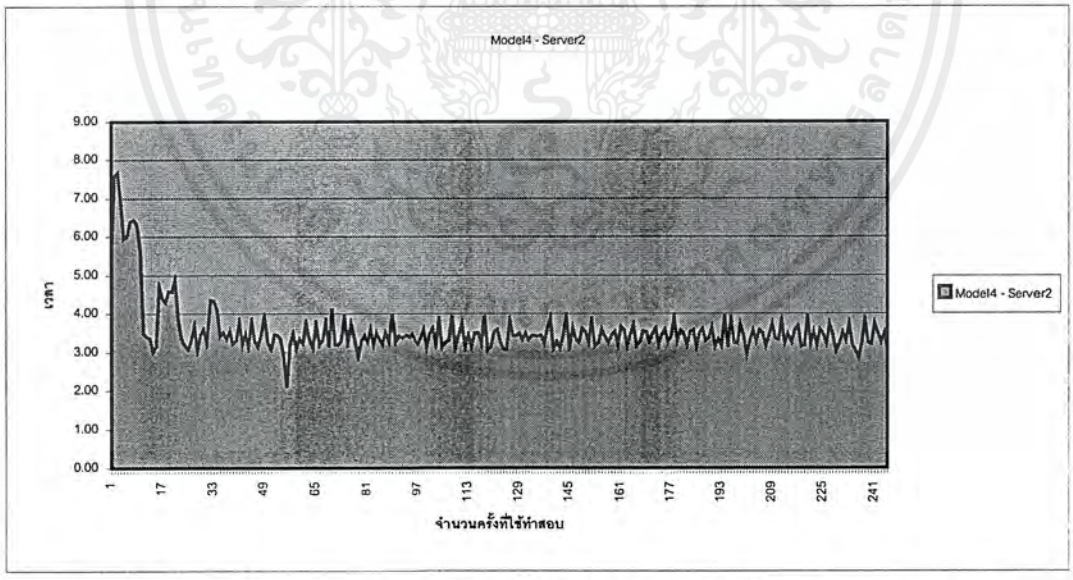
Cache 15 MB (พร็อกซี เซิร์ฟเวอร์1)



รูปที่ 5.14 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 1 ใช้ Cache ขนาด 15 MB

เวลาเฉลี่ยที่ใช้ = 7.36 วินาที

Cache 15 MB (พร็อกซี เซิร์ฟเวอร์2)

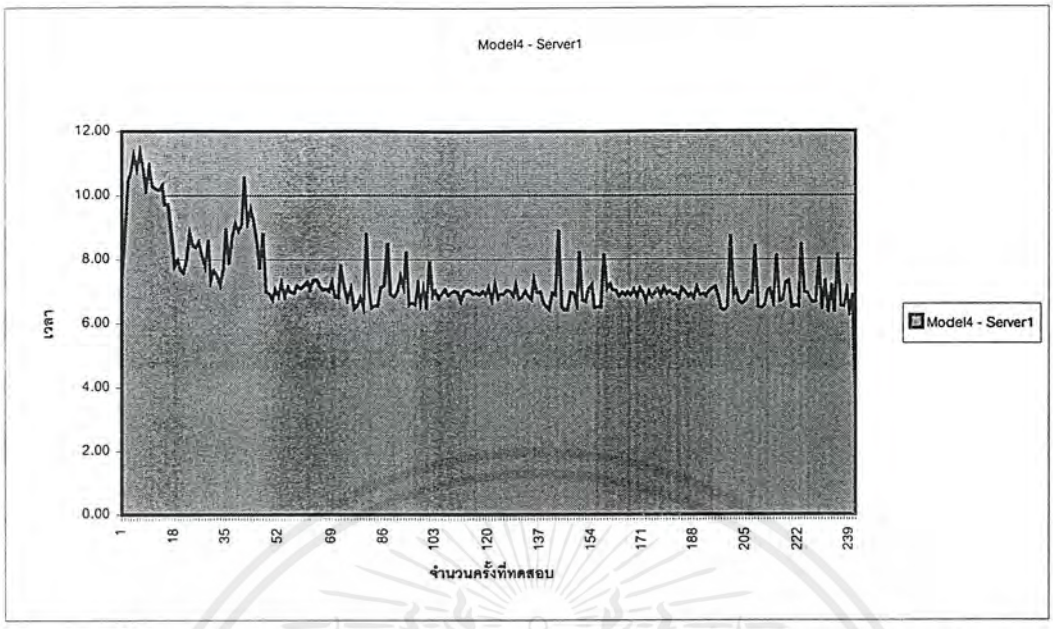


รูปที่ 5.15 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 2 ใช้ Cache ขนาด 15 MB

เวลาเฉลี่ยที่ใช้ = 3.56 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

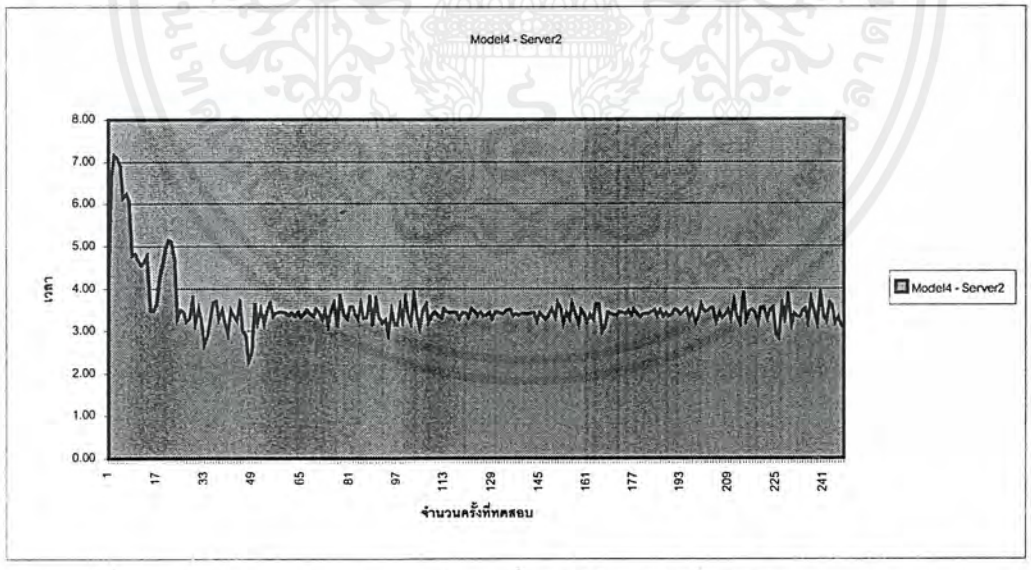
Cache 20 MB (พร็อกซี เซิร์ฟเวอร์1)



รูปที่ 5.16 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 1 ใช้ Cache ขนาด 20 MB

เวลาเฉลี่ยที่ใช้ = 7.37 วินาที

Cache 20 MB (พร็อกซี เซิร์ฟเวอร์2)



รูปที่ 5.17 แสดงผลการทดลองของโมเดลที่ 4 เซิร์ฟเวอร์ตัวที่ 2 ใช้ Cache ขนาด 20 MB

เวลาเฉลี่ยที่ใช้ = 3.54 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

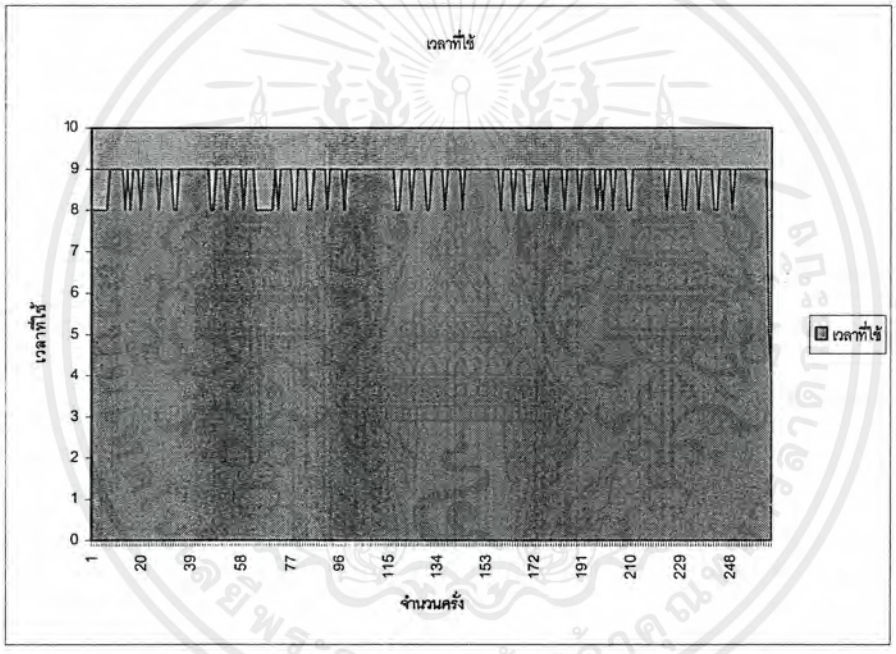
บทวิจารณ์และสรุป

6.1.สรุปผลการทดลองโมเดลที่ 1

จากผลการทดลองโมเดลที่ 1 จะได้เป็นเวลาในการตอบสนองเฉลี่ยของเว็บเซิร์ฟเวอร์ทั่วไป โดยค่าเฉลี่ยนี้จะนำไปใช้ในการเปรียบเทียบกับโมเดลหลังๆ ต่อไป

6.2.สรุปผลการทดลองโมเดลที่ 2

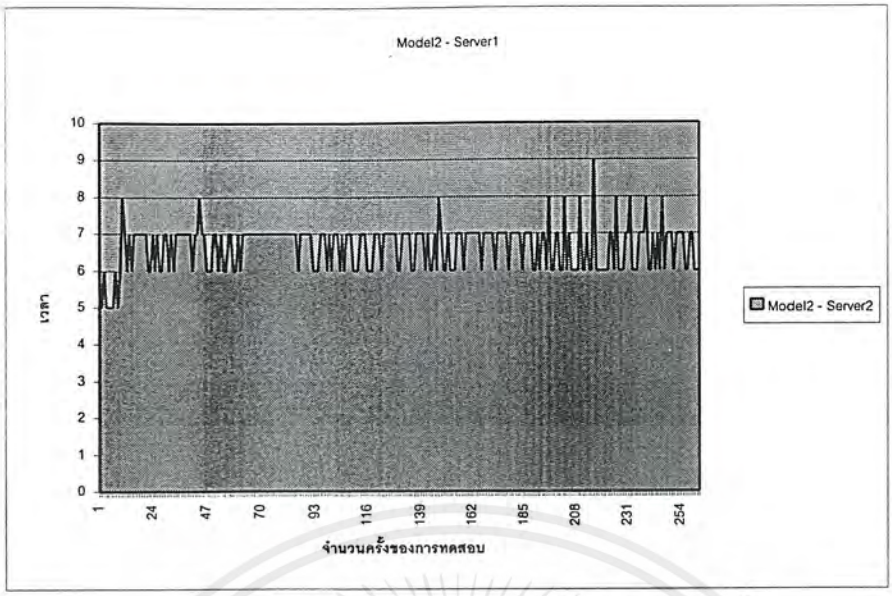
จากผลการทดลองโมเดลที่ 2 จะเห็นได้ว่าการนำเซิร์ฟเวอร์มาต่อทำงานช่วยกัน ส่งผลทำให้เวลาเฉลี่ยในการตอบสนองงานนั้นๆ มีช่วงเวลาที่ลดลง สังเกตได้จากรูปดังนี้



รูปที่ 6.1 แสดงผลการทดลองโมเดลที่ 1

จากรูปที่ 6.1 เมื่อมีการทำบาลานซ์โหลดโดยใช้ความสามารถของ เซิร์ฟเวอร์ DNSเข้าช่วย ในเครื่องเดียวกันส่งผลให้เวลาเฉลี่ยในการทำงานหนึ่งๆ ลดลงดังรูปที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

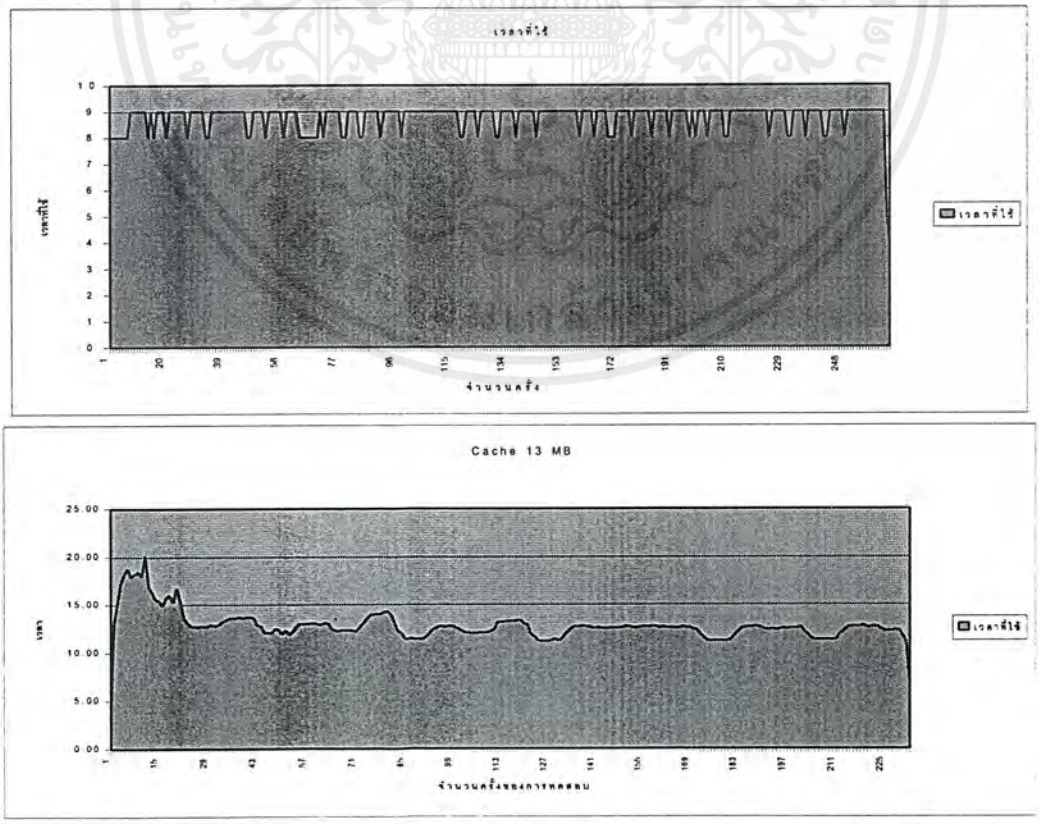


รูปที่ 6.2 แสดงผลการทดลองจากโมเดลที่ 2 เซิร์ฟเวอร์ตัวที่ 1

จากผลการทดลองทั้งรูปที่ 6.1 และรูปที่ 6.2 แสดงให้เห็นว่าในสภาพแวดล้อมและเงื่อนไขที่เรากำหนด โมเดลที่2 สามารถให้เวลาเฉลี่ยในการตอบสนองงานลดลงถึง 24 %

6.3.สรุปผลการทดลองโมเดลที่ 3

จากผลการทดลองโมเดลที่ 3 จากสภาพแวดล้อมและเงื่อนไขที่เรากำหนดขึ้นนั้น ปรากฏว่าการใช้งาน รีเวิร์ส พร็อกซี เข้ามาช่วยไม่ได้ทำให้ประสิทธิภาพการทำงานโดยรวมดีขึ้น สังเกตดังรูปที่ 6.3

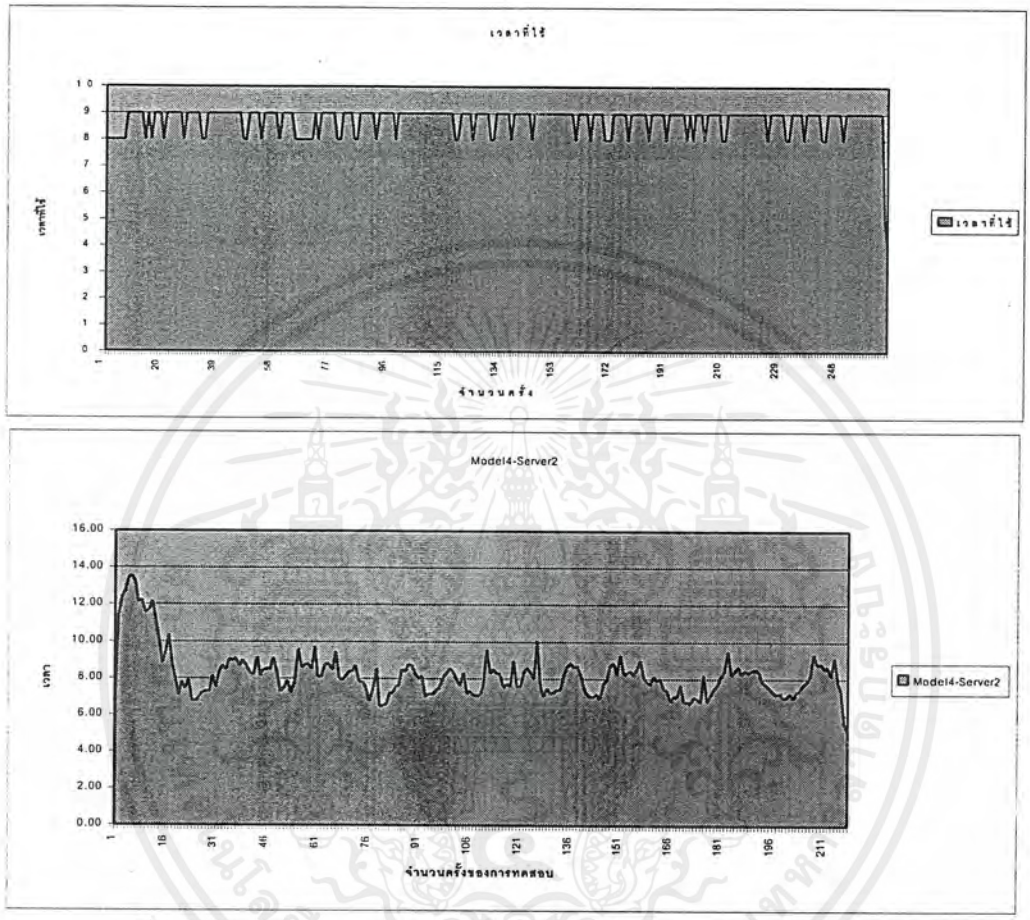


รูปที่ 6.3 แสดงผลการทดลองโมเดลที่ 1 เปรียบเทียบกับผลการทดลองโมเดลที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4. สรุปผลการทดลองโมเดลที่ 4

จากผลการทดลองโมเดลที่ 4 เมื่อเราได้นำ เซิร์ฟเวอร์ DNSเข้ามาทำงานร่วมกับการใช้งาน รีเวิร์ส พร็อกซี ในโมเดลที่ 3 ผลจากการทดลองปรากฏว่า จากสภาพแวดล้อมและเงื่อนไขที่เราจำกัดสำหรับการทดลองครั้งนี้ ผลที่ได้คือ รีเวิร์ส พร็อกซี และการใช้ เซิร์ฟเวอร์ DNSเข้ามาช่วยไม่สามารถช่วยเพิ่มประสิทธิภาพการทำงานโดยรวมของเว็บไซต์ แสดงดังรูปที่ 6.4



รูปที่ 6.4 แสดงผลการทดลองโมเดลที่ 1 เปรียบเทียบกับผลการทดลองโมเดลที่ 4

แต่ในบางลักษณะของการใช้งานโมเดลนี้อาจให้ผลที่ดี สำหรับการใช้ให้เหมาะสมกับลักษณะงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสืออ้างอิง

[1] Johnson David : "Microsoft certified systems engineer พร็อกซี เซิร์ฟเวอร์ 2", Certification Insider Press. c1998

[2] Luotonen Ari : "Web พร็อกซี เซิร์ฟเวอร์s" ,Prentice Hall PTR.c1998

[3] Nicolai Langfeldt : "DNS HOWTO", O'Reilly.1998

[4] Jon Orwant : "PERL 5",Waite Group Press.1997

[5] ภัทระ เกียรติเสวี : "สร้างอินเทอร์เน็ตเซิร์ฟเซอร์ ด้วย Linux", บริษัท ซีเอ็ดยูเคชั่น จำกัด.2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้