การใช้ประโยชน์จากทรัพยากรคอมพิวเตอร์ที่ไม่ได้ใช้งานอย่างเต็มที่สำหรับการจัดสรรทรัพยากรแบบไดนามิคของโปรแกรมประยุกต์ผ่านเว็บ

นายปวิธ ฟุ้งสิริวิโรจน์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2553
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

UTILIZING IDLE COMPUTING RESOURCES FOR DYNAMIC RESOURCE ALLOCATION OF WEB-BASED APPLICATIONS

Mr. Pawit Fungsiriwirot

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

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

Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 2010

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การใช้ประโยชน์จากทรัพยากรคอมพิวเตอร์ที่ไม่ได้ใช้งานอย่างเต็มที่สำหรับการจัดสรรทรัพยากรแบบไดนามิคของโปรแกรมประยุกต์ผ่านเว็บ
โดย	นายปวิธ ฟุ้งสิริวิโรจน์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วีระ เหมืองสิน)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.ภุชงค์ อุทโยภาศ)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ปวิธ ฟุ้งสิริวิโรจน์ : การใช้ประโยชน์จากทรัพยากรคอมพิวเตอร์ที่ไม่ได้ใช้งานอย่างเต็มที่สำหรับการจัดสรรทรัพยากรแบบไดนามิคของโปรแกรมประยุกต์ผ่านเว็บ. (UTILIZING IDLE COMPUTING RESOURCES FOR DYNAMIC RESOURCE ALLOCATION OF WEB-BASED APPLICATIONS) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์, 103 หน้า.

การพัฒนาแอปพลิเคชันในปัจจุบันมีแนวโน้มอยู่ในรูปแบบของเว็บแอปพลิเคชันมากขึ้น และมีเว็บแอปพลิเคชันจำนวนไม่น้อยที่มีรูปแบบการใช้งานในปริมาณมากเป็นช่วงๆ ในลักษณะตามฤดูกาล ทำให้หน่วยงานทั่วไปมักจะอุทิศทรัพยากรเพื่อให้ระบบสามารถรองรับการให้บริการที่เพียงพอตามระดับความต้องการโดยเฉลี่ยเท่านั้น ซึ่งจากลักษณะดังกล่าวอาจส่งผลให้เว็บแอปพลิเคชันเซิร์ฟเวอร์ที่มีอยู่ในระบบไม่สามารถรองรับการใช้งานในช่วงที่มีความต้องการที่สูงได้ สำหรับงานวิจัยที่ผ่านมาได้มีการนำเสนอแนวทางในการแก้ปัญหานี้ เช่น การเสนอเทคนิคการรวมหลายเว็บแอปพลิเคชันให้ร่วมกันใช้ทรัพยากรบนเครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง หรือเทคนิคการขยายทรัพยากรสำหรับเว็บแอปพลิเคชันตามปริมาณการใช้งาน อย่างไรก็ตามค่าใช้จ่ายในส่วนของเครื่องเซิร์ฟเวอร์สำรองที่จะใช้เป็นทรัพยากรเสริมในเทคนิคเหล่านี้ก็เป็นประเด็นสำคัญสำหรับองค์กรที่มีงบประมาณที่จำกัด

วิทยานิพนธ์นี้ได้นำเสนอวิธีการจัดสรรทรัพยากรแบบไดนามิคสำหรับโปรแกรมประยุกต์ผ่านเว็บประเภทที่มีการใช้งานมากเป็นบางช่วงเวลา โดยอาศัยทรัพยากรจากเครื่องเวิร์กสเตชันที่มีอยู่แล้วในหน่วยงาน และไม่ได้ถูกใช้งานอย่างเต็มที่ตลอดเวลา มาใช้เป็นทรัพยากรเสริมให้กับระบบในช่วงที่มีผู้ใช้งานเป็นจำนวนมาก ซึ่งจะช่วยลดค่าใช้จ่ายแทนที่จะต้องใช้ในการอุทิศเครื่องเซิร์ฟเวอร์สำรองให้กับระบบ รวมถึงจะช่วยให้เกิดการใช้ประโยชน์ของทรัพยากรที่ไม่ได้ถูกใช้งานอย่างเต็มที่ให้เกิดประโยชน์สูงขึ้นด้วย นอกจากนี้ งานวิจัยนี้ได้ทำการนำเสนอขั้นตอนวิธีในการขยายขนาดของระบบเว็บแอปพลิเคชัน ให้เหมาะสมกับการใช้ทรัพยากรจากเครื่องเวิร์กสเตชัน โดยสามารถคาดการณ์แนวโน้มของลักษณะภาระงานล่วงหน้า และสามารถขยายขนาดของระบบเว็บแอปพลิเคชันได้ในลักษณะหลายเครื่องตามจำนวนภาระงานที่เพิ่มขึ้นด้วย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต ปวิธ ฟุ้งสิริวิโรจน์

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2553

5270389221 : MAJOR COMPUTER ENGINEERING

KEYWORDS : WEB APPLICATION / VIRTUALIZATION / RESOURCE UTILIZATION / SCALABILITY

PAWIT FUNGSIRIWIROT : UTILIZING IDLE COMPUTING RESOURCES FOR DYNAMIC RESOURCE ALLOCATION OF WEB-BASED APPLICATIONS.

ADVISOR : NATAWUT NUPAIROJ, Ph.D., 103 pp.

At present, new applications are tended to be developed using web-based technology. And many web-based applications exhibit seasonal behaviors such that they are usually being used very lightly most of the time and heavily used from time to time. As the results, organizations usually provide dedicated resources for these systems to support only average workloads. This can lead to inadequate resources during peak periods. In order to solve this problem, several techniques such as web application consolidation on a powerful server or dynamic scaling of web application have been proposed. However the cost of dedicated backend servers required by these techniques can be quite expensive.

This thesis proposes dynamic resource management for seasonal web-based application using remaining capacity of idle workstations to share the load during the peak workload period instead of using a large number of dedicated backend servers. This method will reduce the cost of dedicate servers and increase the utilization of idle computing resources. In addition, we propose the dynamic scaling of web application algorithm suitable for workstation environment by predicting workload in advance and adding several servers at once to handle sudden load surges.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Department : Computer Engineering Student's Signature ปวิต ฟังสิริวิโรจน์

Field of Study : Computer Engineering Advisor's Signature

Academic Year : 2010

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้มิอาจสำเร็จสมบูรณ์ได้หากปราศจากความช่วยเหลือ ข้อคิดเห็น คำแนะนำ และความอนุเคราะห์ต่างๆ จากอาจารย์ที่ปรึกษา อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ ที่คอยให้ความเมตตา รวมทั้งโอกาสและสิ่งที่ดีแก่ผู้วิจัยเสมอมา ผู้วิจัยขอขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.วีระ เหมืองสิน ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา ผู้ช่วยศาสตราจารย์ ดร.ภุชงค์ อุทโยภาศ ประธานกรรมการและคณะกรรมการสอบวิทยานิพนธ์ที่สละเวลามาให้ข้อเสนอแนะอันเป็นประโยชน์ต่อการพัฒนาวิทยานิพนธ์ฉบับนี้เป็นอย่างดี และขอขอบพระคุณคณาจารย์ในจุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ประสิทธิ์ประสาทความรู้อันมีค่ายิ่งแก่ผู้วิจัย

ท้ายสุดนี้ขอขอบพระคุณ คุณพ่อ คุณแม่ พี่ชาย พี่สาว น้องสาว และญาติทุกคนที่บ้านของผู้วิจัยที่คอยสนับสนุนการศึกษาของผู้วิจัยมาโดยตลอด และขอขอบคุณ เพื่อนๆ พี่ๆ และน้องๆ ทุกคน ที่เปรียบเสมือนแรงผลักดันและให้ความช่วยเหลือในทุกๆ ด้านจนผู้วิจัยสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ญ

สารบัญภาพ..... ฏ

บทที่

1 บทนำ..... 1

1.1 ที่มาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของการวิจัย..... 3

1.3 ขอบเขตของการวิจัย..... 4

1.4 ขั้นตอนและวิธีดำเนินการวิจัย..... 5

1.5 คุณค่าทางวิชาการ..... 5

1.6 ผลงานตีพิมพ์จากวิทยานิพนธ์..... 5

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 6

2.1 ทฤษฎีที่เกี่ยวข้อง..... 6

2.1.1 เทคโนโลยีการจำลองระบบเสมือนจริง (Virtualization Technology).... 6

2.1.2 โครงการอัลตรามังค์กี้ (Ultra Monkey Project)..... 9

2.1.3 การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression Analysis)..... 11

2.1.4 ตัวอย่างเว็บแอปพลิเคชันและเครื่องมือที่ใช้ในการจำลองลักษณะของภาระงาน (Sample Web Application and Workload Generation).... 12

2.2 งานวิจัยที่เกี่ยวข้อง..... 16

2.2.1 แนวทางในการรวมหลายเว็บแอปพลิเคชันให้ร่วมกันใช้ทรัพยากรบนเครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง..... 16

2.2.2 แนวทางการขยายทรัพยากรสำหรับเว็บแอปพลิเคชันตามปริมาณการใช้งาน..... 17

บทที่ หน้า

2.2.3 แนวทางการแก้ปัญหาโดยอาศัยทรัพยากรที่มีอยู่แล้วในระบบและไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้เป็นทรัพยากรเสริม..... 18
3 การจัดสรรทรัพยากรสำหรับโปรแกรมประยุกต์ผ่านเว็บโดยอาศัยทรัพยากรจากเครื่องเวิร์กสเตชัน..... 19
3.1 บทกล่าวนำ..... 19
3.2 ภาพรวมแนวความคิดด้านโครงสร้างสถาปัตยกรรม..... 19
3.2.1 ส่วนกระจายภาระงาน (Load Balancer)..... 20
3.2.2 แอปพลิเคชันเซิร์ฟเวอร์หลักของระบบ (Main Application Server)..... 20
3.2.3 เครื่องเวิร์กสเตชันพลูในระบบ (Workstation Pool)..... 21
3.2.4 ส่วนตรวจสอบสถานะของแต่ละแอปพลิเคชันเซิร์ฟเวอร์ (Application Server Monitor)..... 22
3.2.5 ขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชัน (Scaling Algorithm)..... 23
3.2.6 ส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลู (Workstation Pool Manager)..... 24
3.3 ขั้นตอนวิธีการปรับขนาดของเว็บแอปพลิเคชันแบบไดนามิค (Dynamic Scaling Algorithm)..... 27
3.3.1 ขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง (Dynamic Scaling Algorithm: Scale Each)..... 30
3.3.2 ขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยใช้วิธีการคาดการณ์ล่วงหน้า (Dynamic Scaling Algorithm: Scale Predict)..... 36
4 การทดสอบและผลการทดสอบงานวิจัย..... 45
4.1 ตัววัดสมรรถนะของระบบ (Performance Metrics)..... 46
4.2 การทดสอบเพื่อวัดประสิทธิภาพในการรองรับปริมาณภาระงานของแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบ..... 46
4.2.1 เครื่องมือที่ใช้ในการทดลอง..... 46
4.2.2 สภาพแวดล้อมและวิธีการทดลอง..... 48
4.2.3 ผลการทดสอบ..... 48

บทที่ หน้า

4.2.4 วิเคราะห์ผลการทดลอง..... 52
4.3 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพของวิธีการจัดสรรทรัพยากรในระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆเพิ่มขึ้นในปริมาณที่ไม่มาก..... 52
4.3.1 เครื่องมือที่ใช้ในการทดลอง..... 53
4.3.2 สภาพแวดล้อมและวิธีการทดลอง..... 56
4.3.3 ผลการทดสอบ..... 61
4.3.4 วิเคราะห์ผลการทดลอง..... 68
4.4 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพสำหรับวิธีการจัดสรรทรัพยากรของระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีแนวโน้มอัตราการเข้าใช้งานในลักษณะที่เพิ่มขึ้นอย่างรวดเร็ว..... 71
4.4.1 เครื่องมือที่ใช้ในการทดลอง..... 72
4.4.2 สภาพแวดล้อมและวิธีการทดลอง..... 75
4.4.3 ผลการทดสอบ..... 76
4.4.4 วิเคราะห์ผลการทดลอง..... 82
4.5 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพสำหรับวิธีการจัดสรรทรัพยากรของระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน..... 84
4.5.1 เครื่องมือที่ใช้ในการทดลอง..... 85
4.5.2 สภาพแวดล้อมและวิธีการทดลอง..... 88
4.5.3 ผลการทดสอบ..... 89
4.5.4 วิเคราะห์ผลการทดลอง..... 95
5 สรุปผลการวิจัยและข้อเสนอแนะ..... 97
5.1 สรุปผลการวิจัย..... 97
5.2 ข้อเสนอแนะ..... 99
รายการอ้างอิง..... 101
ประวัติผู้เขียนวิทยานิพนธ์..... 103

สารบัญตาราง

ตารางที่		หน้า
4.1	ผลการทดสอบประสิทธิภาพในการรองรับปริมาณภาระงานของแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์	48
4.2	ปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก	61
4.3	ค่าเฉลี่ยของระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณไม่มาก	62
4.4	ปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว	77
4.5	ค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว	77
4.6	ปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน	90
4.7	ค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน	90

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพที่		หน้า
1.1	แสดงแนวโน้มของงานด้านเว็บแอปพลิเคชันในปัจจุบัน [1].....	1
1.2	กราฟแสดงจำนวนครั้งในการคลิกเข้าใช้งานในหน้าเว็บลงทะเบียนเรียนของจุฬาลงกรณ์มหาวิทยาลัยในปี ค.ศ. 2009 [2].....	2
2.1	กราฟแสดงทรัพยากรในคอมพิวเตอร์ซึ่งถูกใช้ประโยชน์ได้ไม่เต็มที่.....	6
2.2	แสดงระดับชั้นเสมือนในสถาปัตยกรรม x86.....	7
2.3	แสดงตารางเปรียบเทียบเทคนิคในการทำเวอร์ชวลไลเซชันที่แตกต่างกัน.....	8
2.4	แสดงลักษณะภาระงานของระบบเว็บไซต์ www.france98.com ในช่วง 1 สัปดาห์ [18].....	14
2.5	แสดงลักษณะปริมาณภาระงานของระบบ Play-along เว็บไซต์ [19].....	14
2.6	แสดงลักษณะปริมาณภาระงานของระบบ Chilean Election Site [19].....	15
2.7	แสดงลักษณะภาระงานในช่วง 20 นาทีแรกที่ระบบ Play-along เว็บไซต์เริ่มมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว [19].....	15
2.8	แสดงลักษณะภาระงานในช่วง 15 นาทีแรกของระบบ Chilean Election Site ที่อัตราการเข้าใช้งานเริ่มมีค่าเพิ่มสูงขึ้นอย่างกะทันหัน [19].....	16
3.1	แสดงภาพรวมของระบบที่นำเสนอ.....	20
3.2	แสดงวงจรชีวิตของเครื่องเวิร์กสเตชัน.....	21
3.3	แสดงภาพรวมการทำงานในส่วนการควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลู.....	24
3.4	แสดงแผนภาพขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง.....	29
3.5	แสดงแผนภาพขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยใช้วิธีการคาดการณ์ล่วงหน้า.....	35
4.1	แสดงปริมาณงานโดยเฉลี่ยที่แต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ใช้ในการตอบสนองต่อผู้ใช้งาน.....	51
4.2	แสดงค่าเฉลี่ยระยะเวลาที่แต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ใช้ในการตอบสนองต่อผู้ใช้งาน.....	51

ภาพที่		หน้า
4.3	แสดงลักษณะของภาระงานที่มีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก	57
4.4	แสดงสภาพแวดล้อมของระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต	58
4.5	แสดงสภาพแวดล้อมของระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง	59
4.6	แสดงสภาพแวดล้อมของระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่อง	60
4.7	กราฟแสดงปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก	64
4.8	กราฟแสดงค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก	64
4.9	กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก	65
4.10	กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่สามารถให้บริการได้อย่างสมบูรณ์ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก	65
4.11	กราฟแสดงปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเวิร์กสเตชันสามารถใช้บริการได้ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก	66
4.12	กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละเครื่องในสภาพแวดล้อมที่ภาระงานมีลักษณะค่อยๆ เพิ่มขึ้น	67
4.13	แสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละหลายเครื่องโดยการคาดการณ์ล่วงหน้าในสภาพแวดล้อมที่ภาระงานค่อยๆ เพิ่มขึ้น	67

ภาพที่ หน้า

4.14 กราฟแสดงการเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีลักษณะค่อยๆ เพิ่มขึ้น..... 68
4.15 แสดงลักษณะภาระงานในช่วงเวลา 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว..... 76
4.16 กราฟแสดงปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว. 78
4.17 กราฟแสดงค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว..... 78
4.18 แสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบในช่วงเวลา 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว..... 79
4.19 กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่สามารถให้บริการได้อย่างสมบูรณ์ในช่วง 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว..... 79
4.20 กราฟแสดงปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเวิร์กสเตชันสามารถใช้บริการได้ในช่วงเวลา 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว..... 80
4.21 กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละเครื่องในช่วง 20 นาทีแรกที่ระบบเริ่มมีการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว..... 81
4.22 แสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละหลายเครื่องในช่วง 20 นาทีแรกที่ระบบเริ่มมีการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว..... 81
4.23 กราฟแสดงการเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันในช่วง 20 นาทีแรกที่ระบบเริ่มมีการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว.. 82
4.24 แสดงภาระงานในช่วงเวลา 15 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 89

ภาพที่ หน้า

4.25 กราฟแสดงปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 91
4.26 กราฟแสดงค่าเฉลี่ยของระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 91
4.27 กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 92
4.28 กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่สามารถให้บริการได้อย่างสมบูรณ์ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 92
4.29 กราฟแสดงปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเวิร์กสเตชันสามารถใช้บริการได้ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 93
4.30 กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละเครื่องในสภาพแวดล้อมที่ภาระงานมีลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 94
4.31 กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละหลายเครื่องในสภาพแวดล้อมที่ภาระงานมีลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 94
4.32 กราฟเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน..... 95

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

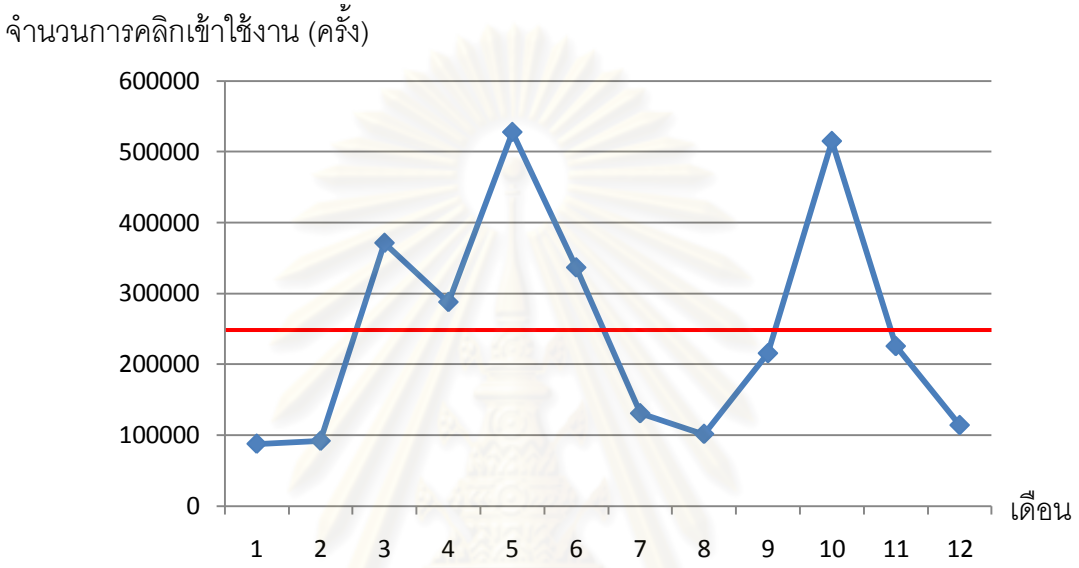
การพัฒนาแอปพลิเคชันในปัจจุบัน มีแนวโน้มที่อยู่ในรูปแบบของเว็บแอปพลิเคชันมากขึ้น สังเกตได้จากแนวโน้มของงานทางด้านเว็บแอปพลิเคชันในปี ค.ศ. 2011 ซึ่งมีค่าเพิ่มขึ้นถึง 77% เมื่อเทียบกับในปี ค.ศ. 2009 [1] แสดงดังรูปที่ 1.1

รูปที่ 1.1 แสดงแนวโน้มของงานด้านเว็บแอปพลิเคชันในปัจจุบัน [1]

เนื่องจากเว็บแอปพลิเคชันโดยทั่วไปมักมีผู้ใช้งานเป็นจำนวนมาก ทำให้ระบบงานเหล่านี้จะต้องถูกออกแบบระบบให้สามารถรองรับการใช้งานอย่างเพียงพอแก่ความต้องการ และเมื่อศึกษาลักษณะการใช้งานของเว็บแอปพลิเคชันเหล่านี้จะพบว่ามีเว็บแอปพลิเคชันจำนวนไม่น้อย ที่มีรูปแบบการใช้งานในปริมาณมากเป็นช่วงๆ ในลักษณะตามฤดูกาล (Seasonal Workload) ตัวอย่าง เช่น เว็บระบบลงทะเบียนเรียนของมหาวิทยาลัย ซึ่งจัดเป็นเว็บแอปพลิเคชันที่มีการใช้งานตามฤดูกาลประเภทหนึ่ง ลักษณะการใช้งานนั้นสามารถแบ่งออกได้เป็น 2 ช่วงคือ

ช่วงที่ 1 ช่วงฤดูกาลลงทะเบียนเรียน คือในช่วงเริ่มเปิดภาคเรียน เช่นในปลายเดือนพฤษภาคมหรือตุลาคมสำหรับภาคปกติ และในปลายเดือนมีนาคมสำหรับภาคฤดูร้อน จะมีนักศึกษาเข้ามาใช้งานในระบบเพื่อทำการลงทะเบียนเรียนพร้อมกันเป็นจำนวนมาก

ช่วงที่ 2 ช่วงปกติ คือหลังจากช่วงที่มีการลงทะเบียนเรียนแล้วเสร็จหรือผ่านพ้นกำหนดการให้ทำการลงทะเบียนเรียนแล้ว เช่นในช่วงเดือนพฤศจิกายนถึงกุมภาพันธ์ และช่วงเดือนมิถุนายนถึงกันยายน ซึ่งในช่วงนี้จะมีจำนวนผู้ใช้งานระบบในจำนวนปกติน้อยมากเมื่อเทียบกับในช่วงที่ 1



รูปที่ 1.2 กราฟแสดงจำนวนครั้งในการคลิกเข้าใช้งานในหน้าเว็บลงทะเบียนเรียนของจุฬาลงกรณ์มหาวิทยาลัยในปี ค.ศ. 2009 [2]

จากกราฟรูปที่ 1.2 โดยปกติในระหว่างภาคการศึกษาในช่วงเดือนพฤศจิกายนถึงกุมภาพันธ์ และช่วงเดือนมิถุนายนถึงกันยายน จะมีผู้ใช้งานที่คลิกเขาใช้งานในหน้าเว็บลงทะเบียนเรียนในจำนวนไม่มากนักเฉลี่ยประมาณ 162,604 ครั้ง แต่จะมีจำนวนการคลิกเข้าใช้งานในหน้าเว็บลงทะเบียนสูงสุดเฉพาะในช่วงเดือนพฤษภาคมและตุลาคมถึงประมาณ 527,417 ครั้ง ซึ่งจะต่างจากปริมาณการใช้งานโดยเฉลี่ยทั้งหมด (Average Workload) ที่มีจำนวนการคลิกเข้าใช้งานประมาณ 250,113 ครั้ง ค่อนข้างมาก จากลักษณะดังกล่าวทำให้หน่วยงานโดยทั่วไปมักจะไม่อุทิศทรัพยากรเพื่อให้ระบบสามารถรองรับการให้บริการปริมาณการใช้งานสูงสุด แต่จะมีเพียงพอแค่การทำงานตามระดับความต้องการโดยเฉลี่ยเท่านั้น ซึ่งจากลักษณะดังกล่าวอาจทำให้เว็บแอปพลิเคชันเซิร์ฟเวอร์ที่มีอยู่ในระบบไม่สามารถรองรับการใช้งานในช่วงที่มีความต้องการที่สูงได้ ส่งผลให้ประสิทธิภาพของระบบและระยะเวลาในการตอบสนองจากระบบต่อผู้ใช้งานอาจช้ากว่าปกติ

ในขณะเดียวในกันการแก้ปัญหาโดยการเตรียมทรัพยากรเผื่อไว้ให้สามารถรองรับการใช้งานให้ได้ทุกช่วงเวลานั้น พบว่าในช่วงเวลาส่วนใหญ่ทรัพยากรที่เตรียมสำรองไว้จะถูกใช้ประโยชน์น้อยมาก ซึ่งทำให้เป็นการใช้ทรัพยากรสิ้นเปลืองโดยใช่เหตุ และจากงานวิจัยที่ผ่านมาได้มีการนำเสนอแนวทางในการแก้ปัญหานี้ เช่น การเสนอเทคนิคการรวมหลายเว็บแอปพลิเคชันให้ร่วมกันใช้ทรัพยากรบนเครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง [3] หรือการนำเสนอเทคนิคการขยายทรัพยากรสำหรับเว็บแอปพลิเคชันตามปริมาณการใช้งาน [4], [5], [6] อย่างไรก็ตามค่าใช้จ่ายในส่วนของเซิร์ฟเวอร์สำรองที่จะใช้เป็นทรัพยากรเสริมในเทคนิคเหล่านี้ก็เป็นประเด็นสำคัญสำหรับองค์กรที่มีงบประมาณที่จำกัด จึงมีงานวิจัย [7] ที่พยายามแก้ปัญหาเหล่านี้โดยอาศัยทรัพยากรบางส่วนจากดีเอ็นเอสเซิร์ฟเวอร์ (DNS Server) หรือเมลเซิร์ฟเวอร์ (Mail Server) ที่มีอยู่แล้วในระบบเพื่อใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานเป็นจำนวนมากแทนที่จะต้องอุทิศเครื่องเซิร์ฟเวอร์สำรองให้แก่ระบบ อย่างไรก็ตามเนื่องจากโดยปกติปริมาณเครื่องเซิร์ฟเวอร์รวมถึงดีเอ็นเอสเซิร์ฟเวอร์หรือเมลเซิร์ฟเวอร์ในแต่ละองค์กรนั้นอาจมีจำนวนไม่มากนักเมื่อเทียบกับทรัพยากรอื่น เช่นเครื่องเวิร์กสเตชันในระบบซึ่งในปัจจุบันถือว่ามีราคาที่ค่อนข้างต่ำกว่ามาก ดังนั้นผู้วิจัยจึงมีแนวคิดว่าการนำทรัพยากรบางส่วนจากเครื่องเวิร์กสเตชันในหน่วยงาน ซึ่งไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้อาจจะเหมาะสมกว่าในการใช้เป็นทรัพยากรเสริมเพื่อให้ระบบเว็บแอปพลิเคชันสามารถรองรับปริมาณการใช้งานในช่วงที่มีผู้ใช้งานเป็นจำนวนมากได้

จากปัญหาที่กล่าวมา ผู้วิจัยมีแนวคิดที่จะนำเสนอวิธีการจัดสรรทรัพยากรแบบไดนามิคสำหรับโปรแกรมประยุกต์ผ่านเว็บประเภทที่มีการใช้งานมากเป็นบางช่วงเวลา โดยอาศัยทรัพยากรจากเครื่องเวิร์กสเตชันที่มีอยู่แล้วในหน่วยงาน และไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานเป็นจำนวนมาก ซึ่งจะเหมาะกับองค์กรที่มีงบประมาณที่จำกัดต้องการลดค่าใช้จ่ายในส่วนของเครื่องเซิร์ฟเวอร์สำรองโดยการใช้ทรัพยากรที่มีอยู่แล้วให้เกิดประโยชน์อย่างเต็มที่

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอวิธีการจัดสรรทรัพยากรสำหรับโปรแกรมประยุกต์ผ่านเว็บประเภทที่มีการใช้งานมากเป็นบางช่วงเวลา โดยอาศัยทรัพยากรจากเครื่องเวิร์กสเตชันที่มีอยู่แล้วในหน่วยงาน และไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานเป็นจำนวนมาก

1.3 ขอบเขตของการวิจัย

1) ในการทดสอบระบบที่นำเสนอจะใช้วิธีการทำให้เกิดผล (Implementation) ซึ่งประกอบด้วย

   - ส่วนของไคล์เอ็น (Client) ในงานวิจัยนี้จะจำลองภาระงาน โดยใช้เครื่องมือวัดประสิทธิภาพ Httperf เวอร์ชัน 0.9.0 ซึ่งสามารถจำลองภาระงานในลักษณะที่ไม่ซับซ้อน คือในแต่ละการเชื่อมต่อ (Connection) จะมีลักษณะภาระงานที่เหมือนกัน และเน้นการประมวลผลในส่วนของแอปพลิเคชันเซิร์ฟเวอร์เป็นหลัก

   - ส่วนของแอปพลิเคชันเซิร์ฟเวอร์หลักของระบบ ในงานวิจัยนี้ได้จำลองเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักซึ่งมีคุณลักษณะเหมือนกับเครื่องเวอร์ชวลแอปพลิเคชันเซิร์ฟเวอร์เนื่องจากข้อจำกัดทางด้านอุปกรณ์เครื่องเซิร์ฟเวอร์จริงที่จะใช้ในการทดลอง

   - ส่วนของเครื่องเวิร์กสเตชัน ในงานวิจัยนี้กำหนดให้แต่ละเครื่องเวิร์กสเตชันมีคุณลักษณะที่ไม่แตกต่างกัน (Homogeneous)

   - ส่วนควบคุมการขยายขนาดของเว็บแอปพลิเคชัน

2) งานวิจัยนี้ได้กำหนดตัววัดประสิทธิภาพของระบบเว็บแอปพลิเคชัน โดยใช้ค่าปริมาณงานโดยเฉลี่ยของระบบ (Average Throughput) หรือค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งาน (Average Response Time) ในการประเมินประสิทธิภาพ

3) สำหรับทรัพยากรในระบบโปรแกรมประยุกต์ผ่านเว็บ งานวิจัยนี้จะพิจารณาเฉพาะทรัพยากรในส่วนของเว็บแอปพลิเคชันเซิร์ฟเวอร์เท่านั้น

4) งานวิจัยนี้กำหนดให้แต่ละเวอร์ชวลแอปพลิเคชันเซิร์ฟเวอร์ที่จำลองขึ้นในระบบมีคุณลักษณะที่ไม่แตกต่างกัน (Homogeneous)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1) ศึกษาลักษณะและปัญหาของระบบเว็บแอปพลิเคชันประเภทที่มีผู้ใช้งานมากตามฤดูกาล

2) ศึกษาทฤษฎีและแนวทางการแก้ปัญหาจากงานวิจัยที่เกี่ยวข้อง

3) วิเคราะห์และออกแบบวิธีการจัดสรรทรัพยากรที่เหมาะสมกับระบบเว็บแอปพลิเคชันประเภทนี้

4) พัฒนาระบบดังกล่าวตามวิธีที่ได้เลือกและศึกษา

5) ทำการทดลองและวัดผลการทดลอง

6) วิเคราะห์ผลการทดลอง

7) สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 คุณค่าทางวิชาการ

สามารถนำวิธีการจัดสรรทรัพยากรที่นำเสนอนี้ไปใช้แก้ปัญหาให้กับระบบโปรแกรมประยุกต์ผ่านเว็บ ซึ่งมีทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์ในระบบไม่เพียงพอที่จะรองรับปริมาณการใช้งานในช่วงที่มีผู้ใช้งานพร้อมกันเป็นจำนวนมากได้ โดยไม่ต้องเสียค่าใช้จ่ายในส่วนของเครื่องเซิร์ฟเวอร์สำรองเพิ่มเติม

1.6 ผลงานตีพิมพ์จากวิทยานิพนธ์

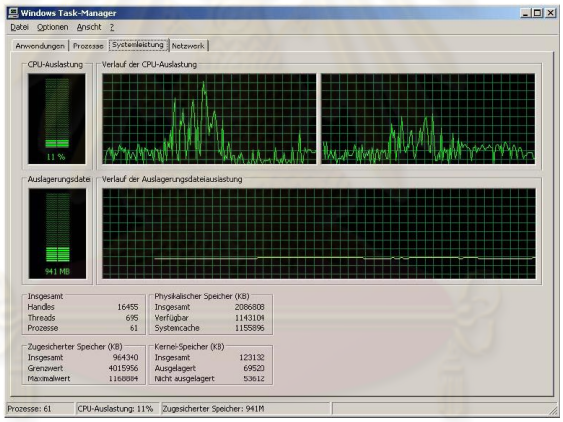
ส่วนหนึ่งของงานวิทยานิพนธ์ได้รับการตีพิมพ์เป็นบทความวิชาการในหัวข้อเรื่อง ”Utilizing Idle Computing Resources for Dynamic Resource Allocation of Web-Based Applications” โดย Pawit Fungsiriwirot and Natawut Nupairoj ในงานประชุมวิชาการ “15th International Annual Symposium on Computational Science and Engineering (ANSCSE15)” ซึ่งจัดขึ้น ณ มหาวิทยาลัยกรุงเทพ ประเทศไทย ระหว่างวันที่ 30 มีนาคม - 2 เมษายน 2554

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

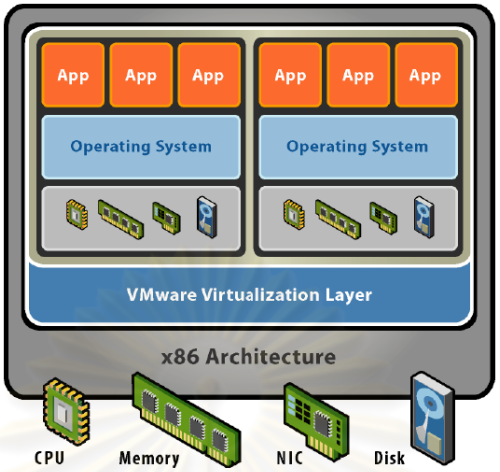
2.1.1 เทคโนโลยีการจำลองระบบเสมือนจริง (Virtualization Technology)

เทคโนโลยีการจำลองระบบเสมือนจริง [8], [9] คือเทคโนโลยีสำหรับสร้างระบบจำลองเสมือนจริงเพื่อให้ระบบปฏิบัติการหลายๆ ระบบสามารถทำงานได้พร้อมกันบนฮาร์ดแวร์หรือเครื่องคอมพิวเตอร์เดียวกันทำให้เกิดการใช้ประโยชน์ทรัพยากรคอมพิวเตอร์ได้อย่างมีประสิทธิภาพสูงสุด เนื่องมาจากเทคโนโลยีทางด้านฮาร์ดแวร์ในปัจจุบันนั้นมีความก้าวหน้าไปมาก ไม่ว่าจะเป็นส่วนของประสิทธิภาพและความเร็วในการประมวลผลซึ่งอาจทำให้ทรัพยากรไม่ได้ถูกใช้ประโยชน์อย่างเต็มที่แสดงดังรูปที่ 2.1 จึงเป็นที่มาของเทคโนโลยีทางด้านเวอร์ชวลไลเซชันในปัจจุบันนี้



รูปที่ 2.1 กราฟแสดงทรัพยากรในคอมพิวเตอร์ซึ่งถูกใช้ประโยชน์ได้ไม่เต็มที่

สำหรับหลักการในการจำลองระบบเสมือนจริงในสถาปัตยกรรมโพรเซสเซอร์ x86 นั้นจะมีการสร้างระดับชั้นเสมือน (Virtualization Layer) ขึ้นมาอยู่ระหว่างส่วนฮาร์ดแวร์และส่วนของระบบปฏิบัติการแสดงดังรูปที่ 2.2 โดยที่ระดับชั้นเสมือนจะมีหน้าที่หลักในการควบคุมการทำงานให้แต่ละระบบปฏิบัติการสามารถใช้งานบนเครื่องเสมือนที่สร้างขึ้นและสามารถใช้ทรัพยากรร่วมกันได้ เช่นสามารถใช้ซีพียู (CPU), หน่วยเก็บข้อมูล (Storage), หน่วยความจำหลัก (Main Memory) และอุปกรณ์อินพุต/เอาต์พุต (I/O devices) ร่วมกันได้ สำหรับแนวทางในการจำลองระบบเสมือนจริงในปัจจุบันมีอยู่ 3 แนวทางหลักคือ



รูปที่ 2.2 แสดงระดับชั้นเสมือนในสถาปัตยกรรม x86

2.1.1.1 Full Virtualization

ในการจำลองระบบเสมือนจริงในเทคนิคนี้จะไม่มีการแก้ไขข้อมูลแกนกลางในส่วนของระบบปฏิบัติการเยือน (Guest Operation System Kernel) แต่จะใช้ไฮเปอร์ไวเซอร์ (Hypervisor) ทำการแปลทุกๆ คำสั่งของระบบปฏิบัติการเพื่อใช้ติดต่อกับในส่วนของฮาร์ดแวร์เพื่อที่จะจำลองระบบเสมือนจริงแทน และเนื่องจากระบบปฏิบัติการเยือนที่จำลองขึ้นเสมือนกับได้แยกการเชื่อมต่อออกจากเครื่องหลักโดยสิ้นเชิง ระบบเสมือนที่สร้างขึ้นจึงมีลักษณะที่สมบูรณ์สามารถนำระบบปฏิบัติการใดๆ มาติดตั้งก็ได้ ตัวอย่างของโปรแกรมการจำลองระบบเสมือนจริงที่ใช้เทคนิคแบบ Full Virtualization เช่น Microsoft Virtual Server [10], VMware [11] และ VirtualBox [12] เป็นต้น

2.1.1.2 OS Assisted Virtualization (Paravirtualization)

ในการจำลองระบบเสมือนจริงในเทคนิคนี้จะมีการดัดแปลงข้อมูลแกนกลางในส่วนของระบบปฏิบัติการเยือนเพื่อให้สามารถจำลองระบบบนสภาวะเครื่องเสมือนได้ อย่างไรก็ตามในบางระบบปฏิบัติการที่ไม่สามารถแก้ไขข้อมูลในส่วนของแกนกลางได้ก็จะไม่สามารถจำลองเครื่องเสมือนได้โดยวิธีการนี้ เช่น Windows 2000/XP เป็นต้น และเนื่องจากวิธีการนี้ระบบปฏิบัติการเยือนสามารถรับรู้ได้ว่ากำลังทำงานอยู่บนซอฟต์แวร์เสมือนจึงทำให้ดูเหมือนกับยังไม่ได้แยกการเชื่อมต่อออกจากเครื่องหลักโดยสิ้นเชิง สำหรับประสิทธิภาพของระบบที่จำลองขึ้นมานั้นเนื่องจากมีการแก้ไขข้อมูล

แกนกลางในส่วนของระบบปฏิบัติการเยือนให้มีความเหมาะสมกับสภาวะเครื่องเสมือนที่จะใช้งานจึงทำให้ประสิทธิภาพของระบบที่จำลองขึ้นมามีค่าใกล้เคียงกับประสิทธิภาพตามธรรมชาติของระบบปฏิบัติการนั้นๆ และมีประสิทธิภาพสูงกว่าการใช้วิธี Full Virtualization ตัวอย่างของโปรแกรมการจำลองระบบเสมือนจริงที่ใช้เทคนิคแบบ Paravirtualization เช่น Xen [13] เป็นต้น

2.1.1.3 Hardware Assisted Virtualization

ในส่วนของผู้ผลิตฮาร์ดแวร์ก็ได้มีการพัฒนาฮาร์ดแวร์รุ่นใหม่ที่มีคุณสมบัติเพื่อสนับสนุนเทคโนโลยีเวอร์ชวลไลเซชันด้วย เช่น Intel Virtualization Technology (VT-x) และ AMD-V เป็นต้น โดยโปรเซสเซอร์เหล่านี้ถูกออกแบบมาให้สามารถรองรับการทำเวอร์ชวลไลเซชันได้เป็นอย่างดี อย่างไรก็ตามสำหรับวิธีการนี้เพิ่งเริ่มมีการพัฒนาไม่นานนัก จึงมักพบเฉพาะในระบบคอมพิวเตอร์รุ่นใหม่เท่านั้นที่สามารถใช้วิธีการนี้ได้ และเนื่องจากเทคนิคนี้ไม่มีการแก้ไขข้อมูลแกนกลางในส่วนของระบบปฏิบัติการเยือนให้มีความสอดคล้องกับสภาวะเครื่องเสมือนจริงที่ได้จำลองขึ้นมาเช่นเดียวกับวิธี 2.1.1.2 รวมถึงยังอยู่ในช่วงการเริ่มพัฒนา จึงทำให้ประสิทธิภาพโดยรวมของระบบเสมือนจริงที่จำลองนั้นยังไม่สูงมากเมื่อเทียบกับในวิธี 2.1.1.1 และ 2.1.1.2 ซึ่งสามารถสรุปความแตกต่างของ 3 เทคนิคนี้ได้ ตามรูปที่ 2.3

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic

รูปที่ 2.3 แสดงตารางเปรียบเทียบเทคนิคในการทำเวอร์ชวลไลเซชันที่แตกต่างกัน

สำหรับในงานวิจัยนี้ได้ใช้เทคโนโลยีของเวอร์ชวลบ็อกซ์ไฮเปอร์ไวเซอร์ (Virtualbox Hypervisor) [12] ซึ่งเป็นเทคโนโลยีการจำลองระบบเสมือนจริงที่ใช้เทคนิคแบบ Full Virtualization เพื่อใช้ในการจำลองทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานเป็นจำนวนมาก เนื่องจากเทคโนโลยีของเวอร์ชวลบ็อกซ์ไฮเปอร์ไวเซอร์นั้นมีลักษณะเป็นโอเพ็นซอร์ส (Open Source) และมีความสามารถในการรองรับได้หลายระบบปฏิบัติการ เช่น ระบบปฏิบัติการวินโดส์ (Windows), ระบบปฏิบัติการลินุกซ์ (Linux) หรือระบบปฏิบัติการสำหรับเครื่องคอมพิวเตอร์แมคอินทอช (Macintosh) เป็นต้น

2.1.2 **โครงการอัลตรามังค์กี้** (Ultra Monkey Project)

โครงการอัลตรามังค์กี้ [14] เป็นโครงการโอเพ็นซอร์ส (Open Source Project) ที่ถูกออกแบบมาเพื่อสร้างตัวกระจายภาระงานที่สามารถให้บริการบนเครือข่ายเพื่อให้มีสภาพพร้อมใช้งานที่สูง (High Availability) ยกตัวอย่างเช่น ในกรณีที่เกิดข้อผิดพลาดกับเครื่องเว็บเซิร์ฟเวอร์เครื่องใดเครื่องหนึ่งในกลุ่มของเว็บเซิร์ฟเวอร์ในระบบ ระบบก็ยังจะสามารถให้บริการแก่ผู้ใช้งานได้ โดยในการใช้งานนั้นผู้ใช้งานจะมีความรู้สึกเหมือนกับว่ากำลังติดต่อกับระบบที่มีเครื่องเซิร์ฟเวอร์ที่ให้บริการอยู่เพียงเครื่องเดียวเท่านั้น องค์ประกอบหลักในโครงการอัลตรามังค์กี้ประกอบไปด้วย 3 ส่วนหลักคือ

1) **ลินุกซ์เวอร์ชวลเซิร์ฟเวอร์** (Linux Virtual Server)

เป็นเทคโนโลยีที่ใช้ในการกระจายภาระงาน (Workload) ให้กับแต่ละเครื่องเซิร์ฟเวอร์ในระบบ เพื่อให้ระบบสามารถรองรับปริมาณการใช้งานที่มีจำนวนมากได้ (Highly Scalable) และเนื่องจากลินุกซ์เวอร์ชวลเซิร์ฟเวอร์มีคุณสมบัติที่สามารถให้ทำการเปลี่ยนแปลงค่าในไฟล์โครงแบบ (Configuration File) ได้โดยไม่จำเป็นต้องหยุดการให้บริการที่กำลังดำเนินอยู่ในขณะนั้น คุณสมบัตินี้จึงเป็นข้อดีสำหรับงานวิจัยนี้ที่จะช่วยให้สามารถเพิ่มหรือลดปริมาณเครื่องเซิร์ฟเวอร์ในระบบเว็บแอปพลิเคชันได้โดยที่ระบบไม่จำเป็นต้องหยุดการให้บริการ

2) **ลินุกซ์เอชเอเฟรมเวิร์ก** (Linux-HA framework)

ลินุกซ์เอชเอเฟรมเวิร์ก จะช่วยในการเฝ้าสังเกต (Monitor) เครื่องที่ทำหน้าที่เป็นตัวกระจายภาระงานในระบบว่าในขณะนั้นมีสถานะอย่างไร ถ้าในกรณีที่ตัวกระจายภาระงานหลัก (Active Load Balancer) ของระบบมีปัญหาเกิดขึ้นก็จะสามารถเรียกใช้ตัวกระจายภาระงานสำรอง (Standby Load Balancer) มาให้บริการแทนได้ทันที

3) **ลินุกซ์ไดเร็คเตอร์** (Ldirectord)

ลินุกซ์ไดเร็คเตอร์ จะทำหน้าที่ในการเฝ้าสังเกตเครื่องเซิร์ฟเวอร์ทั้งหมดในระบบว่าในขณะนั้นมีสถานะปกติพร้อมให้บริการอยู่หรือไม่ ถ้าในกรณีที่มีปัญหาเกิดขึ้นกับเครื่องเซิร์ฟเวอร์เครื่องใดหรือยังไม่พร้อมที่จะให้บริการ ก็จะทำการแจ้งข้อมูลนี้กลับไปยังตัวกระจายภาระงานทราบ เพื่อให้สามารถกระจายภาระงานให้กับเฉพาะเครื่องเซิร์ฟเวอร์ที่มีสถานะพร้อมที่จะให้บริการเท่านั้น

จากลักษณะและองค์ประกอบหลักที่ได้กล่าวมานี้ ในเบื้องต้นงานวิจัยนี้ได้ใช้เทคโนโลยีของโครงการอัลตรามังค์กี้ดังกล่าวในการควบคุมการกระจายภาระงานของระบบเว็บแอปพลิเคชันให้สามารถกระจายภาระงานได้อย่างมีประสิทธิภาพและมีสภาพพร้อมใช้งานที่สูงสำหรับในการให้บริการบนเครือข่าย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.1.3 **การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย** (Simple Linear Regression Analysis)

การวิเคราะห์การถดถอยเชิงเส้นอย่างง่ายเป็นวิธีการทางสถิติอย่างหนึ่ง [15], [16] ที่นิยมใช้ในการคาดการณ์แนวโน้มของตัวแปรที่ต้องการทราบค่าในอนาคตด้วยสมการทางคณิตศาสตร์ โดยรูปแบบของความสัมพันธ์ระหว่างตัวแปรตาม (Y) และตัวแปรอิสระ (X) สามารถแทนได้ด้วยสมการทางคณิตศาสตร์ที่มีลักษณะเป็นเชิงเส้น (Linear Model) ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$Y = \beta X + \varepsilon$$

โดยที่ Y คือตัวแปรตาม

X คือตัวแปรอิสระ

β คือค่าสัมประสิทธิ์ของสมการ

ε คือค่าคงที่

โดยแนวคิดของการวิเคราะห์การถดถอยเชิงเส้นอย่างง่ายจะนำข้อมูลจากตัวแปรที่ทำการศึกษามาวิเคราะห์หาความสัมพันธ์ที่สามารถบอกแนวโน้มของความสัมพันธ์โดยใช้แผนภาพเส้นตรงแทนได้ และจะทำการหาเส้นตรงที่ดีที่สุดเพื่อเป็นตัวแทนของรูปแบบความสัมพันธ์ของตัวแปรที่ศึกษา เส้นตรงที่ดีที่สุดจะมีเพียงเส้นเดียว และจากเส้นตรงดังกล่าวใช้กระบวนการทางสถิติเพื่อหาค่าคงที่และสัมประสิทธิ์ของสมการ สร้างเป็นแบบจำลองในรูปสมการทางคณิตศาสตร์ซึ่งเรียกว่า สมการถดถอยเชิงเส้นหรือสมการพยากรณ์ ที่สามารถใช้ในการพยากรณ์ค่าของตัวแปรตาม (Y) ในอนาคตได้

ขั้นตอนวิธีแบบการถดถอยเชิงเส้นถือเป็นเครื่องมือทางสถิติที่มีการประยุกต์ใช้ในการประมวลผลข้อมูลในงานวิจัยค่อนข้างมาก ในเบื้องต้นงานวิจัยนี้ได้ใช้ขั้นตอนวิธีแบบการถดถอยเชิงเส้นอย่างง่ายนี้ในการคาดการณ์แนวโน้มของจำนวนแอ็กทีพคอนเน็กชัน (Active Connection) ที่ระบบเว็บแอปพลิเคชันจะต้องให้บริการในอนาคตเพื่อให้สามารถเตรียมทรัพยากรการคำนวณไว้ก่อนล่วงหน้าได้

2.1.4 **ตัวอย่างเว็บแอปพลิเคชันและเครื่องมือที่ใช้ในการจำลองลักษณะของภาระงาน** (Sample Web Application and Workload Generation)

ในหัวข้อนี้จะกล่าวถึงรายละเอียดของตัวอย่างเว็บแอปพลิเคชันและเครื่องมือที่ใช้ในการจำลองภาระงานสำหรับใช้ในงานวิจัยนี้ รวมถึงลักษณะของภาระงานที่จะใช้ในการทดสอบระบบเว็บแอปพลิเคชัน พร้อมกับยกตัวอย่างของระบบเว็บแอปพลิเคชันจริงที่มีลักษณะภาระงานที่สอดคล้องกับในแต่ละสภาพแวดล้อม

2.1.4.1 **ตัวอย่างเว็บแอปพลิเคชันและเครื่องมือที่ใช้ในการจำลองภาระงาน**

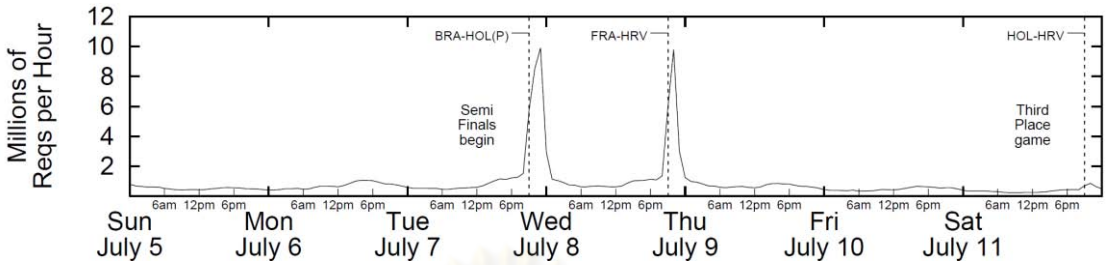
ในงานวิจัยนี้ได้ใช้ตัวอย่างเว็บแอปพลิเคชัน (Simple Web Application) และเครื่องมือที่ใช้ในการจำลองลักษณะของภาระงาน (Workload Generation) โดยอ้างอิงจากงานวิจัย [5] ซึ่งได้ใช้ตัวอย่างเว็บแอปพลิเคชันและเครื่องมือในการจำลองภาระงานดังกล่าวนี้ ทำการทดลองเพื่อประเมินประสิทธิภาพในการขยายขนาดของทรัพยากรเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบ สำหรับตัวอย่างเว็บแอปพลิเคชันดังกล่าวจะเน้นการประมวลผลในส่วนซีพียูของเครื่องแอปพลิเคชันเซิร์ฟเวอร์เป็นหลัก การทำงานหลักคือจะรับอินพุตจากผู้ใช้งานเพื่อใช้ในการคำนวณค่าเมทริกซ์ (Matrix) ตามจำนวนรอบที่ผู้ใช้งานได้กำหนดไว้ โดยลักษณะของเว็บแอปพลิเคชันดังกล่าวจะประกอบด้วยหนึ่งจาวาเซิร์ฟเลท (Java Servlet) ซึ่งใช้เป็นตัวแทนในการจำลองพฤติกรรมการใช้งานจริงของไดนามิคเว็บแอปพลิเคชัน คือจะสลับการใช้งานระหว่างทรัพยากรในส่วนประมวลผลซีพียูของแอปพลิเคชันเซิร์ฟเวอร์ และการใช้งานในส่วนที่ไม่ใช่ทรัพยากรในการคำนวณ (Non-Compute Resources) เช่น การใช้งานในอุปกรณ์อินพุต/เอาต์พุต (Local I/O) และการเชื่อมต่อเครือข่าย (Network Connections) เป็นต้น และในงานวิจัยดังกล่าว [5] ได้ใช้เครื่องมือ Httperf [17] ในการจำลองภาระงานให้กับระบบเว็บแอปพลิเคชันในลักษณะที่ไม่ซับซ้อน คือในแต่ละการเชื่อมต่อ (Connection) จะมีลักษณะภาระงานที่เหมือนกันและจะเน้นการประมวลผลในส่วนของแอปพลิเคชันเซิร์ฟเวอร์เป็นหลัก โดยในการทดลองได้สร้างภาระงานที่มีจำนวนผู้ใช้งานในระบบเว็บแอปพลิเคชันในอัตรายูสเซอร์เซสชันต่อวินาที (User Sessions per Second) ในปริมาณที่แตกต่างกัน เพื่อแสดงถึงค่าความหนาแน่นของการเข้าใช้งานในระบบเว็บแอปพลิเคชันที่มีปริมาณที่แตกต่างกันด้วย โดยลักษณะภาระงานของในแต่ละยูสเซอร์เซสชันนั้นจะมีลักษณะที่เหมือนกัน คือจะ

ประกอบด้วย 10 การร้องขอ (Requests) และช่วงพัก (Pause) จำนวน 4 ช่วง เพื่อใช้ในการจำลองเวลาที่ผู้ใช้งานใช้ในการคิดตัดสินใจ (User Think Time)

2.1.4.2 **ลักษณะของภาระงานที่ใช้ในการทดสอบระบบเว็บแอปพลิเคชัน**

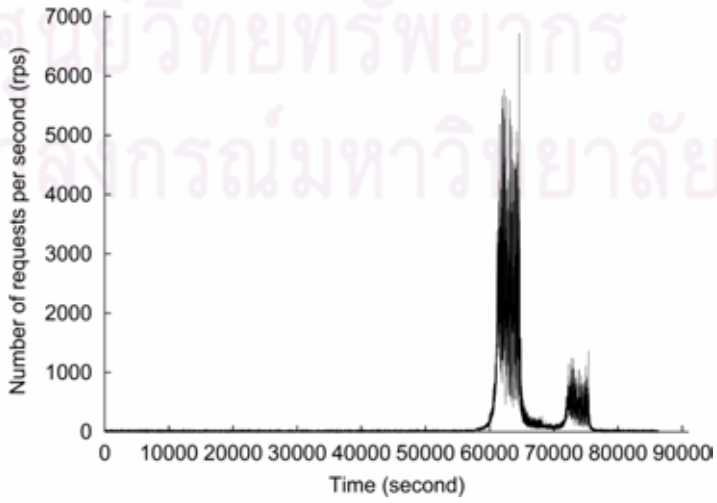
สำหรับลักษณะของภาระงานที่งานวิจัยนี้จะใช้ในการทดสอบระบบเว็บแอปพลิเคชันนั้นได้แบ่งออกเป็น 3 สภาพแวดล้อมหลัก คือ ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก และในสภาพแวดล้อมที่ภาระงานมีแนวโน้มอัตราการเข้าใช้งานในลักษณะที่เพิ่มขึ้นอย่างรวดเร็ว รวมถึงในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน โดยในการทดลองจะจำลองสภาพแวดล้อมภาระงานของระบบเว็บแอปพลิเคชันที่มีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มากโดยอ้างอิงลักษณะของภาระงานที่จะใช้ในการทดสอบจากงานวิจัย [5] ที่ได้จำลองลักษณะภาระงานที่มีอัตราการเข้าใช้งานในระบบเว็บแอปพลิเคชันในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มากเพื่อทดสอบการขยายขนาดของระบบเว็บแอปพลิเคชัน

สำหรับสภาพแวดล้อมที่ภาระงานของระบบเว็บแอปพลิเคชันมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็วนั้น ผู้วิจัยได้อ้างอิงลักษณะของภาระงานดังกล่าวจากงานวิจัย [18], [19] ซึ่งสภาพแวดล้อมดังกล่าวมักเกิดขึ้นบ่อยกับระบบเว็บแอปพลิเคชันที่มีผู้ใช้งานให้ความสนใจและนิยมเข้าใช้บริการพร้อมๆ กันเป็นจำนวนมากในช่วงเวลาเดียวกัน ยกตัวอย่างเช่น ในงานวิจัยของ Arlitt และคณะ [18] ที่ได้ทำการศึกษาลักษณะภาระงานของระบบเว็บไซต์ www.france98.com ซึ่งเป็นระบบเว็บแอปพลิเคชันที่ให้ข้อมูลรายละเอียดรวมถึงการร่วมกิจกรรมต่างๆ เกี่ยวกับการแข่งขันฟุตบอลโลก อย่างเช่น การตรวจผลการแข่งขันของแต่ละเกมส์การแข่งขันแบบทันที (Real Time) การให้ข้อมูลประวัติหรือสถิติของผู้เล่นและทีมที่สนใจ รวมถึงกิจกรรมต่างๆ เพื่อชิงรางวัลทายผลการแข่งขัน เป็นต้น ซึ่งพบว่าลักษณะของภาระงานจะมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็วประมาณ 5 ถึง 10 เท่าตัวเพียงแค่ในช่วงเวลาก่อนจะเริ่มมีการแข่งขั้นในแต่ละเกมส์เท่านั้น ดังแสดงในรูปที่ 2.4

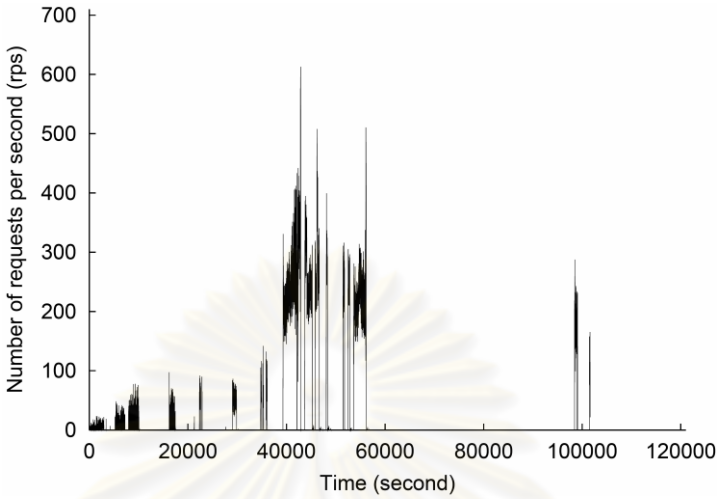


รูปที่ 2.4 แสดงลักษณะภาระงานของระบบเว็บไซต์ www.france98.com ในช่วง 1 สัปดาห์ [18]

นอกจากนี้ในงานวิจัยของ Jung และคณะ [19] ยังได้ทำการศึกษาลักษณะของภาระงานที่มีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็วของระบบ Play-along เว็บไซต์ ซึ่งเป็นระบบเว็บแอปพลิเคชันสำหรับใช้แสดงรายการโทรทัศน์แบบเชื่อมตรง (On-line) สำหรับรายการโทรทัศน์ที่กำลังเป็นที่นิยมมากในขณะนั้น และระบบ Chilean Election Site ซึ่งเป็นระบบเว็บแอปพลิเคชันสำหรับปรับ (Update) ข้อมูลผลการเลือกตั้งประธานาธิบดีในชิลีแบบทันที (Real Time) โดยจากรูปที่ 2.5 และ 2.6 จะสังเกตเห็นได้ว่าปริมาณภาระงานของระบบดังกล่าวจะมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว โดยเฉพาะระบบ Play-along เว็บไซต์จะมีอัตราการเข้าใช้งานที่เพิ่มสูงเฉพาะในช่วงเวลาที่เริ่มมีการแสดงรายการโทรทัศน์เท่านั้น หลังจากช่วงเวลาดังกล่าวปริมาณภาระงานก็จะลดลงเข้าสู่สภาวะปกติซึ่งมีปริมาณที่ไม่มากนัก

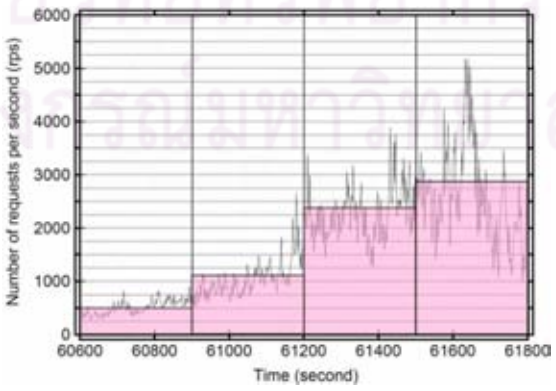


รูปที่ 2.5 แสดงลักษณะปริมาณภาระงานของระบบ Play-along เว็บไซต์ [19]

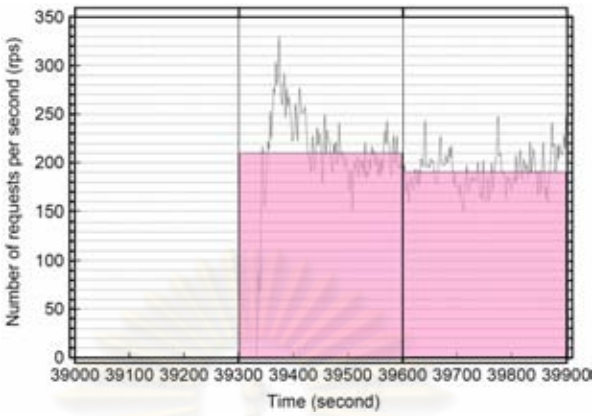


รูปที่ 2.6 แสดงลักษณะปริมาณภาระงานของระบบ Chilean Election Site [19]

สำหรับในงานวิจัยนี้จะทำการทดสอบระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว โดยอ้างอิงลักษณะภาระงานในช่วง 20 นาทีแรกที่กำลังเริ่มมีค่าเพิ่มสูงขึ้นอย่างรวดเร็วของระบบ Play-along เว็บไซต์ คือตั้งแต่วินาทีที่ 60,600 ถึง 61,800 ซึ่งมีแนวโน้มเพิ่มสูงเฉลี่ยประมาณ 6 เท่าตัวของปริมาณภาระงานเริ่มต้น และอ้างอิงลักษณะของภาระงานในช่วง 15 นาทีแรกที่มีลักษณะเพิ่มสูงขึ้นอย่างกะทันหันของระบบ Chilean Election Site คือตั้งแต่วินาทีที่ 39,000 ถึง 39,900 ซึ่งสามารถแสดงได้ดังรูปที่ 2.7 และ 2.8 ตามลำดับ



รูปที่ 2.7 แสดงลักษณะภาระงานในช่วง 20 นาทีแรกที่ระบบ Play-along เว็บไซต์เริ่มมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว [19]



รูปที่ 2.8 แสดงลักษณะภาระงานในช่วง 15 นาทีแรกของระบบ Chilean Election Site ที่อัตราการเข้าใช้งานเริ่มมีค่าเพิ่มสูงขึ้นอย่างกะทันหัน [19]

2.2 งานวิจัยที่เกี่ยวข้อง

จากการศึกษางานวิจัยที่เกี่ยวข้องกับการจัดสรรทรัพยากรสำหรับระบบงานผ่านเว็บประเภทที่มีการใช้งานมากเป็นบางช่วงเวลานั้น ผู้วิจัยสามารถแบ่งแนวทางออกได้เป็น 3 แนวทางหลัก คือ

2.2.1 แนวทางในการรวมหลายเว็บแอปพลิเคชันให้ร่วมกันใช้ทรัพยากรบนเครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง

Padala และคณะ [3] ได้นำเสนอ Adaptive Controller ซึ่งทำหน้าที่ควบคุมเปอร์เซ็นต์การใช้งานซีพียูของแต่ละเว็บแอปพลิเคชันที่แชร์ใช้ทรัพยากรบนเครื่องเซิร์ฟเวอร์ร่วมกันให้เป็นไปตามค่าที่ต้องการ ทำให้สามารถกำหนดเปอร์เซ็นต์การใช้งานของซีพียูส่วนใหญ่ให้กับเว็บแอปพลิเคชันที่มีภาระงานจำนวนมาก (Peak Workload) ในช่วงเวลานั้นได้ อย่างไรก็ตามความสามารถในการขยายขนาดของระบบก็เป็นเป็นประเด็นสำคัญที่จะต้องคำนึงถึงด้วย เช่นในอนาคตกรณีที่มีผู้ใช้งานเพิ่มขึ้นเป็นจำนวนมาก ทรัพยากรบนเครื่องเซิร์ฟเวอร์ทั้งหมดอาจไม่เพียงพอที่จะรองรับการใช้งานในปริมาณมากเหล่านั้นได้ จำเป็นต้องทำการยกระดับ (Upgrade) หรือเปลี่ยนขนาดเครื่องใหม่เพื่อให้สามารถรองรับปริมาณผู้ใช้งานที่มีมากขึ้น ซึ่งอาจจะกลายเป็น

ปัญหาตามมาได้ จึงมีการเสนอแนวทางในการแก้ปัญหาเหล่านี้โดยใช้เทคนิคของการขยายทรัพยากรสำหรับเว็บแอปพลิเคชันตามปริมาณการใช้งานขึ้น

2.2.2 แนวทางการขยายทรัพยากรสำหรับเว็บแอปพลิเคชันตามปริมาณการใช้งาน

Chieu และคณะ [4] ได้นำเสนอขั้นตอนวิธีในการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละเครื่องตามการใช้งาน โดยใช้เทคโนโลยีเวอร์ชวลไลเซชัน [8] เข้าช่วย และคำนวณค่าน้ำหนักในการกระจายภาระงานให้กับแต่ละเว็บแอปพลิเคชันอย่างเหมาะสม นอกจากนี้ Iqbal และคณะ [5] ยังได้นำเสนอขั้นตอนวิธีในการขยายเว็บแอปพลิเคชันในลักษณะทีละเครื่องตามการใช้งานเช่นกัน โดยใช้ค่าเฉลี่ยของระยะเวลาที่ระบบตอบสนองต่อผู้ใช้งาน (Average Response Time) เป็นตัวชี้วัดในการสเกล วิธีการคือในกรณีที่แอปพลิเคชันเซิร์ฟเวอร์เครื่องใดเครื่องหนึ่งให้ระยะเวลาในการตอบสนองต่อผู้ใช้งานเกินกว่าค่าที่กำหนด ระบบจะทำการเพิ่มแอปพลิเคชันเซิร์ฟเวอร์อีกเครื่องเข้าในระบบเพื่อให้สามารถรองรับปริมาณการใช้งานที่เพิ่มขึ้นและช่วยรักษาค่าเฉลี่ยของระยะเวลาที่ระบบจะตอบสนองต่อผู้ใช้งานให้อยู่ในช่วงที่กำหนดไว้ โดยงานวิจัยนี้มีลักษณะคล้ายกับงานวิจัยของ Araki [6] ที่ได้เสนอ Autonomic WWW Server Management ซึ่งใช้ควบคุมปริมาณเว็บเซิร์ฟเวอร์ในระบบตามการใช้งาน โดยใช้ค่าเฉลี่ยของระยะเวลาที่ระบบใช้ตอบสนองต่อผู้ใช้งานเป็นตัวกำหนดในการเพิ่มหรือลดจำนวนเว็บเซิร์ฟเวอร์และทำหน้าที่ควบคุมการกระจายภาระงานของผู้ใช้งานให้ไปประมวลผลบนเครื่องเซิร์ฟเวอร์ที่มีภาระงานที่น้อยและอยู่ใกล้กับผู้ใช้งาน สำหรับจุดเด่นของงานวิจัยนี้คือ ทรัพยากรที่ใช้นั้นสามารถแชร์ใช้ร่วมกันระหว่างองค์กรกันได้ โดยอาศัยเทคโนโลยีของกริด (Grid Technology) เข้ามาช่วย จึงทำให้ทรัพยากรสามารถถูกใช้ประโยชน์ได้มากขึ้นกว่าการใช้เฉพาะภายในองค์กรเดียวเท่านั้น

จะเห็นได้ว่าโดยภาพรวมของแนวทางนี้สามารถแก้ปัญหาข้างต้นได้ ด้วยวิธีการเพิ่มเครื่องเซิร์ฟเวอร์สำรองให้กับระบบเว็บแอปพลิเคชันเพื่อให้สามารถรองรับปริมาณการใช้งานในช่วงเวลาที่มีผู้ใช้งานเป็นจำนวนมาก อย่างไรก็ตามค่าใช้จ่ายในส่วนของเครื่องเซิร์ฟเวอร์สำรองที่จะใช้เป็นทรัพยากรเสริมในเทคนิคเหล่านี้ก็เป็นประเด็นสำคัญสำหรับองค์กรที่มีงบประมาณที่จำกัด จึงมีการเสนอแนวทางในการแก้ปัญหาเหล่านี้โดยอาศัยทรัพยากรที่มีอยู่แล้วในระบบมาใช้เป็นทรัพยากรเสริมแทนเพื่อพยายามลดค่าใช้จ่ายในส่วนของเซิร์ฟเวอร์สำรองนี้ลง

2.2.3 แนวทางการแก้ปัญหาโดยอาศัยทรัพยากรที่มีอยู่แล้วในระบบและไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้เป็นทรัพยากรเสริม

Tsang-Long และ Jian-Bo [7] ได้เสนอวิธีการแก้ปัญหาเหล่านี้โดยพยายามใช้ทรัพยากรบางส่วนจากดีเอ็นเอสเซิร์ฟเวอร์และเมลเซิร์ฟเวอร์ที่มีอยู่แล้วในระบบ มาใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานเป็นจำนวนมาก แทนที่จะต้องอุทิศเครื่องเซิร์ฟเวอร์สำรองให้แก่ระบบ ซึ่งวิธีการดังกล่าวจะช่วยลดค่าใช้จ่ายในส่วนของเครื่องเซิร์ฟเวอร์สำรองเหล่านี้ลงได้ และจากผลการทดลองได้แสดงให้เห็นว่าการใช้ทรัพยากรบางส่วนจากดีเอ็นเอสเซิร์ฟเวอร์หรือเมลเซิร์ฟเวอร์นั้นสามารถให้ประสิทธิภาพได้ดีเทียบเท่ากับการอุทิศเครื่องเซิร์ฟเวอร์สำรองให้กับระบบ

จะเห็นได้ว่าข้อดีของแนวทางนี้คือสามารถแก้ปัญหาข้างต้น ในกรณีที่ระบบเว็บแอปพลิเคชันมีทรัพยากรไม่เพียงพอสำหรับรองรับการใช้งานในปริมาณมาก โดยการใช้ประโยชน์จากทรัพยากรที่มีอยู่แล้วในระบบซึ่งจะช่วยลดค่าใช้จ่ายในส่วนของการใช้เครื่องเซิร์ฟเวอร์สำรองลงได้ อย่างไรก็ตามโดยปกติปริมาณเครื่องเซิร์ฟเวอร์รวมถึงดีเอ็นเอสเซิร์ฟเวอร์หรือเมลเซิร์ฟเวอร์ในองค์กรส่วนใหญ่ มักมีจำนวนไม่มากนักเมื่อเทียบกับทรัพยากรอย่างอื่น เช่นเครื่องเวิร์กสเตชันในระบบ ซึ่งในปัจจุบันถือว่ามีราคาที่ค่อนข้างต่ำกว่ามาก ดังนั้นผู้วิจัยจึงมีแนวคิดว่าการนำทรัพยากรบางส่วนจากเครื่องเวิร์กสเตชันที่มีอยู่แล้วในหน่วยงาน และไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้อาจจะเหมาะสมกว่าในการใช้เป็นทรัพยากรเสริม เพื่อให้ระบบเว็บแอปพลิเคชันสามารถรองรับปริมาณการใช้งานในช่วงที่มีผู้ใช้งานพร้อมกันเป็นจำนวนมากได้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

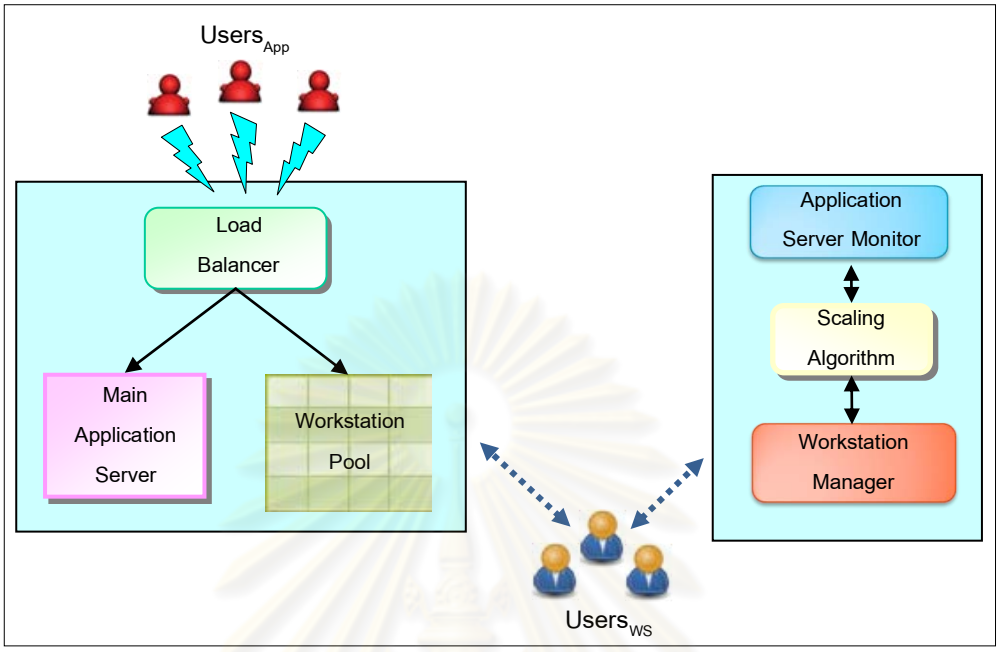
การจัดสรรทรัพยากรสำหรับโปรแกรมประยุกต์ผ่านเว็บโดยอาศัยทรัพยากรจากเครื่องเวิร์กสเตชัน

3.1 บทกล่าวนำ

งานวิจัยนี้มีแนวคิดในการนำทรัพยากรบางส่วนจากเครื่องเวิร์กสเตชันที่มีอยู่แล้วในหน่วยงาน และไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานเป็นจำนวนมาก เพื่อให้ระบบสามารถรองรับปริมาณการใช้งานจำนวนมากเหล่านี้ให้ได้ ซึ่งจะทำให้ช่วยลดค่าใช้จ่ายในส่วนของเครื่องเซิร์ฟเวอร์สำรองลง รวมถึงจะช่วยทำให้เกิดการใช้ประโยชน์ของทรัพยากรที่ไม่ได้ถูกใช้งานอย่างเต็มที่ให้เกิดประโยชน์สูงขึ้นด้วย และงานวิจัยนี้ยังได้ทำการพัฒนาขั้นตอนวิธีในการขยายขนาดของเว็บแอปพลิเคชัน โดยปรับปรุงขั้นตอนวิธีการของ [4] จากเดิมซึ่งมีลักษณะการขยายขนาดได้ครั้งละเครื่อง ให้สามารถขยายขนาดของเว็บแอปพลิเคชันให้ได้ครั้งละหลายเครื่องตามปริมาณภาระงานที่เพิ่มขึ้นด้วย โดยเนื้อหาภายในบทนี้จะแบ่งออกเป็น 3 ส่วนคือ ในส่วนแรกจะเป็นบทกล่าวนำ ส่วนที่สองจะอธิบายเกี่ยวกับภาพรวมแนวความคิดด้านโครงสร้างสถาปัตยกรรมของระบบ และส่วนสุดท้ายจะเป็นการอธิบายถึงขั้นตอนวิธีการปรับขนาดของเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบในลักษณะทีละเครื่องและขั้นตอนวิธีการปรับขนาดของเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบในลักษณะทีละหลายเครื่องโดยใช้วิธีการคาดการณ์แนวโน้มของลักษณะภาระงานก่อนล่วงหน้า ตามลำดับ

3.2 ภาพรวมแนวความคิดด้านโครงสร้างสถาปัตยกรรม

ในหัวข้อนี้จะอธิบายสถาปัตยกรรมของระบบซึ่งประกอบไปด้วย 6 ส่วนหลักดังนี้ คือ 1) ส่วนกระจายภาระงาน 2) แอปพลิเคชันเซิร์ฟเวอร์หลักของระบบ 3) เครื่องเวิร์กสเตชันพลูในระบบ 4) ส่วนตรวจสอบสถานะของแต่ละแอปพลิเคชันเซิร์ฟเวอร์ 5) ขั้นตอนวิธีการปรับขนาดของเว็บแอปพลิเคชัน และ 6) ส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลู ซึ่งภาพรวมของระบบที่นำเสนอได้แสดงดังรูปที่ 3.1



รูปที่ 3.1 แสดงภาพรวมของระบบที่นำเสนอ

3.2.1 **ส่วนกระจายภาระงาน** (Load Balancer)

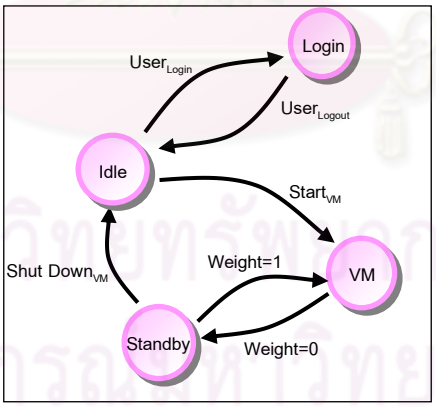
เป็นส่วนที่ทำหน้าที่ควบคุมการกระจายภาระงานจากผู้ใช้งานทั้งหมดโดยจะกระจายภาระงานในปริมาณที่เหมาะสมให้กับแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ที่กำลังเปิดให้บริการอยู่ในขณะนั้น สำหรับในกรณีที่มีปัญหาเกิดขึ้นกับเครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องใดหรือยังไม่พร้อมที่จะให้บริการ ส่วนกระจายภาระงานนี้ก็จะทำการปรับปรุงข้อมูลสถานะดังกล่าวแบบทันที (Real Time) เพื่อให้สามารถกระจายภาระงานให้กับเฉพาะเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะพร้อมที่จะให้บริการได้เท่านั้น

3.2.2 **แอปพลิเคชันเซิร์ฟเวอร์หลักของระบบ** (Main Application Server)

คือเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักที่ได้อุทิศให้กับระบบเว็บแอปพลิเคชัน ซึ่งจะสามารถรองรับปริมาณของผู้ใช้งานได้เฉพาะในช่วงปกติที่มีการใช้งานไม่มากเท่านั้น อย่างไรก็ตามสำหรับในช่วงที่มีผู้ใช้งานในระบบเว็บแอปพลิเคชันพร้อมกันเป็นจำนวนมาก (Peak Load) ทรัพยากรเครื่องเซิร์ฟเวอร์หลักนี้อาจไม่เพียงพอที่จะสามารถรองรับการใช้บริการที่มีจำนวนมากได้

3.2.3 **เครื่องเวิร์กสเตชันพลูในระบบ** (Workstation Pool)

คือทรัพยากรเครื่องเวิร์กสเตชันทั้งหมดในระบบ ที่ไม่ได้ถูกใช้ประโยชน์อย่างเต็มที่ ซึ่งในงานวิจัยนี้ได้กำหนดให้ทรัพยากรเครื่องเวิร์กสเตชันในระบบนั้น มีเพียงพอสำหรับการให้บริการแก่ผู้ใช้งานเครื่องเวิร์กสเตชัน และมีเพียงพอต่อการใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันด้วย โดยที่ทรัพยากรเครื่องเวิร์กสเตชันเหล่านี้สามารถนำมาใช้จำลองเป็นเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือน เพื่อนำมาใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานพร้อมกันเป็นจำนวนมากได้ สำหรับในงานวิจัยนี้ได้ใช้เทคโนโลยีของเวอร์ชวลบ็อกซ์ไฮเปอร์ไวเซอร์ (Virtualbox Hypervisor) เวอร์ชั่น 4.0 เพื่อใช้จำลองเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนให้กับระบบเว็บแอปพลิเคชัน เนื่องจากเทคโนโลยีของเวอร์ชวลบ็อกซ์ไฮเปอร์ไวเซอร์นั้นมีลักษณะเป็นโอเพ็นซอร์ส และมีความสามารถในการรองรับได้หลายระบบปฏิบัติการ เช่น ระบบปฏิบัติการวินโดส์ (Windows), ระบบปฏิบัติการลินุกซ์ (Linux) หรือ ระบบปฏิบัติการสำหรับเครื่องคอมพิวเตอร์แมคอินทอช (Macintosh) เป็นต้น และจากลักษณะการใช้ประโยชน์จากทรัพยากรของเครื่องเวิร์กสเตชันดังที่ได้กล่าวมานั้น สามารถสรุปวงจรชีวิต (Life Cycle) ของแต่ละเครื่องเวิร์กสเตชันในระบบได้ตามรูปที่ 3.2 โดยที่แต่ละสถานะมีรายละเอียดดังนี้คือ



รูปที่ 3.2 แสดงวงจรชีวิตของเครื่องเวิร์กสเตชัน

สถานะ Login หมายถึง เครื่องเวิร์กสเตชันที่กำลังมีผู้ใช้งานลงบันทึกเข้า (Login) ใช้งานอยู่

Idle หมายถึง เครื่องเวิร์กสเตชันในเวิร์กสเตชันพลูที่ว่าง ไม่ได้ถูกใช้งาน

VM หมายถึง เครื่องเวิร์กสเตชันที่กำลังใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในขณะนั้น

Standby หมายถึง เครื่องเวิร์กสเตชันที่กำลังใช้เป็นทรัพยากรเสริมให้กับเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนซึ่งกำลังจะถูกปิดการทำงาน (Shut Down) ลงในไม่ช้า โดยจะรอให้บริการกับการเชื่อต่อที่ยังคงค้างอยู่ให้เสร็จสมบูรณ์ก่อน

3.2.4 ส่วนตรวจสอบสถานะของแต่ละแอปพลิเคชันเซิร์ฟเวอร์ (Application Server Monitor)

ในส่วนนี้จะทำหน้าที่ตรวจสอบและเก็บข้อมูลตัวชี้วัดการปรับขนาด (Scaling Indicator) ของระบบเว็บแอปพลิเคชันและข้อมูลสถานะของแต่ละแอปพลิเคชันเซิร์ฟเวอร์ เพื่อใช้เป็นข้อมูลสำหรับขั้นตอนวิธีในการปรับขนาดของระบบเว็บแอปพลิเคชันต่อไป

สำหรับตัวอย่างของตัวชี้วัดในการปรับขนาดของระบบเว็บแอปพลิเคชันที่นิยมใช้กัน ได้แก่ ปริมาณแอ็กทีพคอนเน็กชัน (Active Concurrent Connection) หรือค่าเฉลี่ยของระยะเวลาที่ระบบใช้ตอบสนองต่อผู้ใช้งาน (Average Response Time) เป็นต้น สำหรับในการใช้งานจริงนั้นก็ควรเลือกตัวชี้วัดให้เหมาะกับลักษณะของแต่ละระบบเว็บแอปพลิเคชันด้วย อย่างเช่น ในกรณีที่ระบบเว็บแอปพลิเคชันมีลักษณะเป็นแบบกระแสข้อมูล (Streaming) หรือระบบเว็บแอปพลิเคชันที่ต้องการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ให้ได้ครั้งละหลายเครื่อง การใช้ปริมาณแอ็กทีพคอนเน็กชันเป็นตัวชี้วัดในการสเกลนั้นอาจจะสะดวกกว่าการใช้ค่าเฉลี่ยของระยะเวลาที่ระบบใช้ตอบสนองต่อผู้ใช้งานเป็นตัวชี้วัดในการสเกล และเนื่องจากในงานวิจัยนี้ต้องการเพิ่มความสามารถให้ระบบเว็บแอปพลิเคชันสามารถขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ให้ได้ครั้งละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า ดังนั้นในเบื้องต้นจะทำการเลือกพิจารณาข้อมูลจำนวนแอ็กทีพคอนเน็กชันที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์นั้นได้ให้บริการ เพื่อใช้เป็นข้อมูลสำหรับขั้นตอนวิธีในการปรับขนาดของระบบเว็บแอปพลิเคชันต่อไป

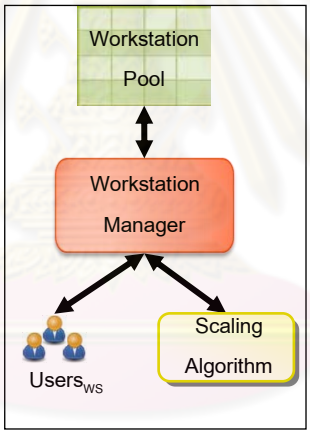
อย่างไรก็ตามในการใช้ปริมาณแอ็กทีพคอนเน็กชันเป็นตัวชี้วัดในการสเกลนั้น พบว่าอาจเกิดกรณีที่ลักษณะเนื้อหา (Content) ของแต่ละระบบเว็บแอปพลิเคชันนั้นอาจมีความหลากหลาย ซึ่งจะส่งผลให้แต่ละการเชื่อมต่ออาจมีปริมาณภาระงานที่แตกต่างกันได้ ดังนั้นในการใช้งานจริงควรมีการพิจารณาค่าน้ำหนัก (Weight) ของแต่ละประเภทของลักษณะภาระงานให้มีความสอดคล้องกับปริมาณค่าแอ็กทีพคอนเน็กชันหรือค่ามาตรฐาน (Baseline) เดียวกันก่อน เพื่อให้สามารถประเมินลักษณะปริมาณภาระงานโดยรวมของระบบเว็บแอปพลิเคชันดังกล่าวได้อย่างถูกต้องมากขึ้น

3.2.5 ขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชัน (Scaling Algorithm)

ในส่วนของขั้นตอนวิธีการที่ใช้ในการปรับขนาดของระบบเว็บแอปพลิเคชันที่นำเสนอในงานวิจัยนี้มี 2 ลักษณะคือ ขั้นตอนวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง ซึ่งโดยหลักการแล้วได้ใช้แนวคิดและวิธีการหลักจากงานวิจัย [4] คือในแต่ละครั้งที่ตรวจพบว่าควรที่จะขยายขนาดของระบบเว็บแอปพลิเคชัน เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชัน (Mng) จะทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือน (Virtual Application Server) ให้แก่ระบบเพียงครั้งละหนึ่งเครื่องเท่านั้น อย่างไรก็ตามสำหรับในช่วงที่มีผู้ใช้งานเพิ่มขึ้นในปริมาณมาก มีความจำเป็นต้องใช้ทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนเพื่อรองรับปริมาณภาระงานในจำนวนหลายเครื่อง การเพิ่มปริมาณทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนในลักษณะทีละเครื่องนี้อาจจะไม่เหมาะกับปริมาณภาระงานของระบบที่มีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว ดังนั้นผู้วิจัยจึงทำการพัฒนาขั้นตอนวิธีในการขยายขนาดของระบบเว็บแอปพลิเคชันดังกล่าว ให้สามารถขยายขนาดของระบบเว็บแอปพลิเคชันให้ได้ครั้งละหลายเครื่อง พร้อมกับเพิ่มความสามารถให้สามารถคาดการณ์แนวโน้มของปริมาณภาระงานในอนาคต ทำให้ระบบสามารถเตรียมทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนไว้ก่อนล่วงหน้าได้ จากลักษณะดังกล่าวช่วยทำให้ระบบสามารถรองรับปริมาณผู้ใช้งานที่มีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็วได้เป็นอย่างดี สำหรับขั้นตอนในการปรับขนาดของระบบเว็บแอปพลิเคชันทั้ง 2 ลักษณะนั้นได้อธิบายขั้นตอนวิธีการโดยละเอียดไว้ในหัวข้อที่ 3.3

3.2.6 **ส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลู** (Workstation Pool Manager)

เนื่องจากในงานวิจัยนี้มีแนวคิดในการใช้ทรัพยากรบางส่วนจากเครื่องเวิร์กสเตชันที่มีอยู่แล้วในหน่วยงาน และไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้เป็นทรัพยากรเสริมให้กับระบบ ดังนั้นจึงมีความจำเป็นที่จะต้องมีระบบควบคุมการใช้งานของเครื่องเวิร์กสเตชันเหล่านี้ เพื่อให้สามารถถูกใช้ประโยชน์ได้อย่างเต็มที่และไม่รบกวนการใช้งานกัน โดยองค์ประกอบในส่วนนี้จะทำหน้าที่ตรวจสอบและเก็บสถานะของแต่ละเครื่องเวิร์กสเตชันในขณะนั้นว่าอยู่ในสถานะใดบ้าง หลังจากนั้นจะใช้ข้อมูลเหล่านี้ในการค้นหาเครื่องเวิร์กสเตชันที่เหมาะสมให้กับผู้มาขอใช้งานเครื่องเวิร์กสเตชัน หรือค้นหาทรัพยากรเครื่องเวิร์กสเตชันที่ไม่ได้ถูกใช้งานอย่างเต็มที่ เพื่อนำมาใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานพร้อมกันเป็นจำนวนมากได้โดยไม่ทำให้เกิดการแย่งการใช้งานทรัพยากรระหว่างกัน



รูปที่ 3.3 แสดงภาพรวมการทำงานในส่วนการควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลู

สำหรับขั้นตอนวิธีในการควบคุมทรัพยากรในเครื่องเวิร์กสเตชันพลูนั้น เพื่อให้เกิดความเข้าใจได้ง่ายจะขออธิบายการทำงานออกเป็น 2 ส่วนหลักดังที่ได้แสดงในรูปที่ 3.3 คือ ขั้นตอนการทำงานที่ใช้ติดต่อกับผู้มาขอใช้งานเครื่องเวิร์กสเตชัน และขั้นตอนการทำงานที่ใช้ติดต่อกับส่วนของขั้นตอนวิธีในการปรับขนาดของระบบเว็บแอปพลิเคชันในหัวข้อที่ 3.2.6.1 และ 3.2.6.2 ตามลำดับ โดยในแต่ละขั้นตอนวิธีนั้นจะมีลักษณะการทำงานที่สอดคล้องกับวงจรชีวิตของแต่ละเครื่องเวิร์กสเตชันดังที่ได้แสดงไว้แล้วในรูปที่ 3.2

3.2.6.1 ขั้นตอนการทำงานที่ใช้ติดต่อกับผู้มาขอใช้งานเครื่องเวิร์กสเตชัน

1) ในกรณีที่มีผู้ต้องการใช้งานเครื่องเวิร์กสเตชัน ผู้ใช้งานนั้นจะทำการขอหมายเลขเครื่องเวิร์กสเตชันจากในส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลูก่อน

2) ส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลูจะทำการค้นหาเครื่องเวิร์กสเตชันที่มีสถานะ Idle คือไม่ได้ถูกใช้งานในขณะนั้น และส่งหมายเลขเครื่องดังกล่าวให้แก่ผู้ที่ต้องการใช้งาน

3) หลังจากนั้นส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลู จะทำการเปลี่ยนสถานะของเครื่องเวิร์กสเตชันเครื่องนั้นจากเดิมมีสถานะ Idle ให้มีสถานะเป็น Login ซึ่งหมายความว่ามีผู้ใช้งานได้ทำการลงบันทึกเข้าใช้งานอยู่ในขณะนั้น

4) หลังจากที่ผู้ใช้งานเครื่องเวิร์กสเตชันดังกล่าวได้ใช้งานเสร็จเรียบร้อยแล้ว เมื่อผู้ใช้งานได้ทำการลงบันทึกออกจากระบบ (Logout) เครื่องเวิร์กสเตชันเครื่องนั้นก็จะส่งข้อมูลกลับไปยังส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลูว่าผู้ใช้งานได้คืนทรัพยากรให้กับระบบเรียบร้อยแล้ว

5) ส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลูก็จะทำการเปลี่ยนสถานะของเครื่องเวิร์กสเตชันเครื่องนั้นจากสถานะ Login ให้มีสถานะเป็น Idle คือไม่ได้ถูกใช้งานและพร้อมที่จะนำทรัพยากรนี้ไปใช้งานอื่นได้อีกในครั้งต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.2.6.2 ขั้นตอนการทำงานที่ใช้ติดต่อกับในส่วนควบคุมการปรับขนาดของระบบเว็บแอปพลิเคชัน

1) สำหรับในช่วงที่มีผู้ใช้งานในระบบเว็บแอปพลิเคชันพร้อมกันเป็นจำนวนมาก เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะตรวจพบได้ว่า ภาระงานในขณะนั้นมีปริมาณมากขึ้นส่งผลให้ทรัพยากรเว็บแอปพลิเคชันหลักในระบบมีไม่เพียงพอที่จะใช้รองรับปริมาณภาระงานดังกล่าว จำเป็นต้องเพิ่มจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมให้แก่ระบบ ก็จะมาร้องขอทรัพยากรเครื่องเวิร์กสเตชันที่ไม่ได้ถูกใช้งานอย่างเต็มที่ เพื่อนำไปใช้จำลองเป็นทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือน เพื่อใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานพร้อมกันเป็นจำนวนมาก

2) ส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลูจะทำการเลือกเครื่องเวิร์กสเตชันที่มีสถานะ Idle คือมีสถานะว่างและไม่ได้ถูกใช้งานอยู่ในขณะนั้น และส่งหมายเลขที่อยู่ไอพีของเครื่องเวิร์กสเตชันดังกล่าวให้กับส่วนที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชัน เพื่อนำไปใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันต่อไป

3) หลังจากนั้นส่วนควบคุมการใช้งานของเครื่องเวิร์กสเตชันพลู ก็จะทำการเปลี่ยนสถานะของเครื่องเวิร์กสเตชันเครื่องนั้นจากเดิมมีสถานะเป็น Idle ให้มีสถานะเป็น VM ซึ่งหมายความว่ากำลังถูกใช้งานเป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันอยู่ในขณะนั้น

4) เมื่อผู้ใช้งานในระบบเว็บแอปพลิเคชันมีปริมาณลดลง ส่วนที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันก็จะตรวจพบได้ว่าภาระงานในขณะนั้นมีปริมาณน้อยลงด้วย และควรที่จะลดขนาดของทรัพยากรเว็บแอปพลิเคชันในระบบลงเพื่อให้สอดคล้องกับปริมาณการใช้งานในขณะนั้น ดังนั้นส่วนที่

ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันก็จะปิดการทำงาน (Shut Down) ของเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนลงเพื่อคืนทรัพยากรให้กับระบบ

5) หลังจากนั้นสถานะของเครื่องเวิร์กสเตชันดังกล่าวก็จะถูกเปลี่ยนจากเดิมที่มีสถานะเป็น VM ให้มีสถานะเป็น Standby และ Idle ตามลำดับและพร้อมที่จะนำทรัพยากรนี้ไปใช้งานอื่นได้อีกในครั้งต่อไป

ซึ่งโดยภาพรวมแล้วองค์ประกอบในส่วนนี้จะทำหน้าที่ควบคุมการใช้งานของทรัพยากรเครื่องเวิร์กสเตชันให้สามารถถูกใช้ประโยชน์ได้มากขึ้น ทั้งให้ใช้งานเป็นเครื่องเวิร์กสเตชันทั่วไปและใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีการใช้งานเป็นจำนวนมากโดยไม่ให้มีการแย่งกันใช้งาน

3.3 ขั้นตอนวิธีการปรับขนาดของเว็บแอปพลิเคชันแบบไดนามิค (Dynamic Scaling Algorithm)

ในหัวข้อนี้จะกล่าวถึงรายละเอียดขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชัน ซึ่งโดยภาพรวมแล้วก่อนที่จะทำการปรับขนาดของระบบเว็บแอปพลิเคชันจะต้องทำการกำหนดค่าขอบเขตบนสุด (Upper-threshold) และค่าขอบเขตล่างสุด (Lower-threshold) ที่จะใช้เป็นเกณฑ์ในการปรับขนาดของระบบเว็บแอปพลิเคชันก่อน โดยที่

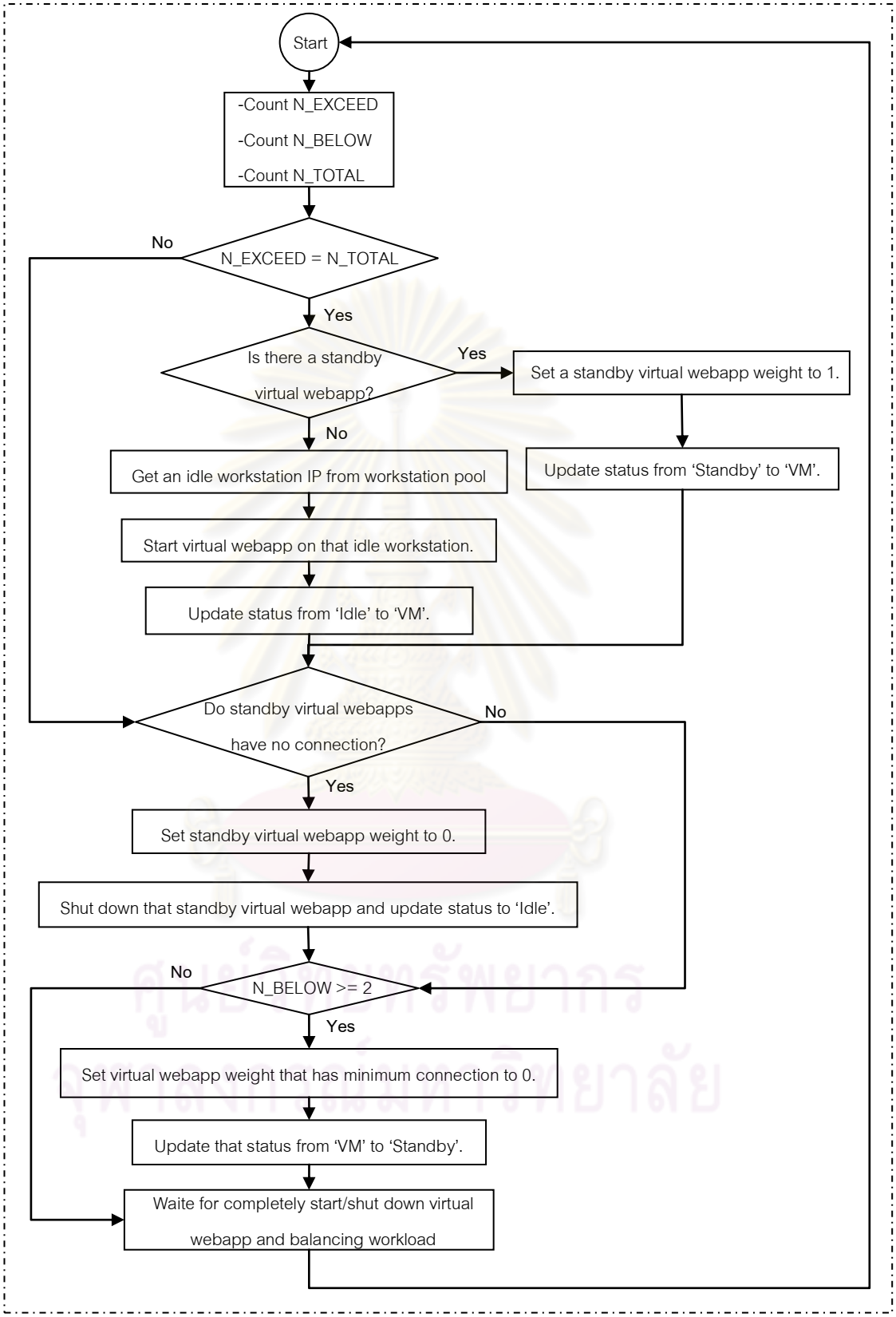
ค่าขอบเขตบนสุด (Upper-threshold) คือ ค่าเปอร์เซ็นต์มากที่สุดของอัตราส่วนระหว่างจำนวนแอ็กทีพคอนเน็กชันที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์กำลังให้บริการอยู่ในขณะนั้นกับจำนวนแอ็กทีพคอนเน็กชันที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับได้ (ตัวอย่างเช่น กำหนดให้ค่าขอบเขตบนสุด มีค่าเท่ากับ 90% หมายความว่ากำหนดให้แต่ละแอปพลิเคชันเซิร์ฟเวอร์ในระบบ ควรมีปริมาณแอ็กทีพคอนเน็กชันไม่เกิน 90% ของปริมาณแอ็กทีพคอนเน็กชันที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับได้)

ค่าขอบเขตล่างสุด (Lower-threshold) คือ ค่าเปอร์เซ็นต์น้อยที่สุดของอัตราส่วนระหว่างจำนวนแอ็กทีพคอนเน็กชันที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์กำลังให้บริการอยู่ในขณะนั้นกับจำนวนแอ็กทีพคอนเน็กชันที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับได้ (ตัวอย่างเช่น กำหนดให้ค่าขอบเขตล่างสุด มีค่าเท่ากับ 45% หมายความว่ากำหนดให้แต่ละแอป

พลิเคชันเซิร์ฟเวอร์ในระบบ ควรมีปริมาณแอ็กทีพคอนเน็กชัน มากกว่า 45% ของปริมาณแอ็กทีพคอนเน็กชันที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับได้)

โดยหลักการแล้วเมื่อกำหนดค่าขอบเขตบนสุด และค่าขอบเขตล่างสุดที่จะใช้เป็นเกณฑ์ในการปรับขนาดของระบบเว็บแอปพลิเคชัน เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบจำนวนแอ็กทีพคอนเน็กชันของแต่ละแอปพลิเคชันเซิร์ฟเวอร์ เปรียบเทียบกับค่าขอบเขตบนสุดและค่าขอบเขตล่างสุดที่ได้กำหนดไว้ เพื่อใช้ในการตัดสินใจในการขยายขนาดหรือลดขนาดของระบบเว็บแอปพลิเคชันตามปริมาณการใช้งานของผู้ใช้งานในขณะนั้น สำหรับขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่องและขั้นตอนวิธีการปรับขนาดของเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยใช้วิธีการคาดการณ์ล่วงหน้า สามารถเขียนเป็นแผนภาพแสดงขั้นตอนการทำงานได้ดังรูปที่ 3.4 และ 3.5 ตามลำดับ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.4 แสดงแผนภาพขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง

3.3.1 ขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง (Dynamic Scaling Algorithm: Scale Each)

1) เริ่มต้นเครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบค่าต่างๆ ดังนี้

   a. N_EXCEED คือ จำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในระบบที่มีจำนวน แอ็กทีพคอนเน็กชันเกินกว่าค่าขอบเขตบนสุด (Upper-threshold) ที่ได้กำหนดไว้ในขณะนั้น

   b. N_BELOW คือจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในระบบที่มีจำนวนแอ็กทีพคอนเน็กชันต่ำกว่าค่าขอบเขตล่างสุด (Lower-threshold) ที่ได้กำหนดไว้ในขณะนั้น

   c. N_TOTAL คือ จำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ทั้งหมดในระบบในขณะนั้น

2) เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบว่าจำนวน N_EXCEED ในขณะนั้นมีค่าเท่ากับจำนวน N_TOTAL หรือไม่

   a. ในกรณีที่จำนวน N_EXCEED มีค่าไม่เท่ากับจำนวน N_TOTAL แสดงว่าระบบเว็บแอปพลิเคชันในปัจจุบันยังมีทรัพยากรเพียงพอที่จะใช้รองรับปริมาณภาระงานในขณะนั้น ดังนั้นเครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบจะยังไม่ทำการขยายขนาดของระบบเว็บแอปพลิเคชัน

   b. ในกรณีที่จำนวน N_EXCEED มีค่าเท่ากับจำนวน N_TOTAL หมายความว่าทรัพยากรทั้งหมดในระบบเว็บแอปพลิเคชันเริ่มจะไม่สามารถรองรับปริมาณภาระงานในขณะนั้นได้ ดังนั้นเครื่องที่ควบคุมการขยายขนาดของระบบจะเริ่มทำการขยายขนาดของเว็บแอปพลิเคชัน โดยมีขั้นตอนดังนี้คือ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบจะทำการค้นหาว่าในขณะนั้นมีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby อยู่ในระบบบ้างหรือไม่
    - ในกรณีที่มี
        - เครื่องที่ควบคุมการขยายขนาดของระบบ จะทำการส่งข้อมูลไปยังเครื่องที่ทำหน้าที่กระจายภาระงานให้กับระบบเว็บแอปพลิเคชันให้ทำการเปลี่ยนแปลงค่าน้ำหนักของเครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องนั้น จากเดิมที่มีค่าน้ำหนักเท่ากับ 0 ให้มีค่าเท่ากับ 1 เพื่อให้สามารถใช้รองรับปริมาณภาระงานใหม่ที่จะเข้ามาในระบบได้ และทำการเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชัน ที่ได้ใช้จำลองเป็นเครื่องแอปพลิเคชันเซิร์ฟเวอร์ดังกล่าวจากเดิมที่มีสถานะเป็น Standby ให้มีสถานะเป็น VM
    - ในกรณีที่ไม่มี
        - เครื่องที่ควบคุมการขยายขนาดของระบบจะทำการค้นหาหมายเลขที่อยู่ไอพีสของเครื่องเวิร์กสเตชันที่มีสถานะเป็น Idle ในเวิร์กสเตชันพลูจำนวนหนึ่งเครื่อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- หลังจากนั้นเครื่องที่ควบคุมการขยายขนาดของระบบ จะทำการจำลองเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนบนเครื่องเวิร์กสเตชันดังกล่าว เพื่อใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชัน
- เครื่องที่ควบคุมการขยายขนาดของระบบ จะทำการเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชันที่ได้ใช้จำลองเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนดังกล่าว จากเดิมที่มีสถานะเป็น Idle ให้มีสถานะเป็น VM เพื่อแสดงว่ากำลังให้บริการแก่ผู้ใช้งานในระบบเว็บแอปพลิเคชัน

3) เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบว่าในขณะนั้นระบบมีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby และไม่มีการเชื่อมต่อกับผู้ใช้งานอยู่บ้างหรือไม่

   a. ในกรณีที่ไม่มีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะดังกล่าว แสดงว่าในขณะนั้นระบบอาจไม่มีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะ Standby หรือเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะ Standby ในระบบกำลังให้บริการแก่ผู้ใช้งานอยู่ ซึ่งไม่ควรที่จะปิดการทำงานลง (Shut Down)

   b. ในกรณีที่ระบบมีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะดังกล่าว เครื่องที่ควบคุมการขยายขนาดของระบบจะปิดการทำงานของเครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องนั้นลงโดยมีขั้นตอนดังนี้คือ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- เครื่องที่ควบคุมการขยายขนาดของระบบจะส่งข้อมูลไปยังเครื่องที่ทำหน้าที่กระจายภาระงานให้กับระบบเว็บแอปพลิเคชัน ให้ทำการเปลี่ยนแปลงค่าน้ำหนักของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ดังกล่าวให้มีค่าเท่ากับศูนย์

- หลังจากนั้นเครื่องที่ควบคุมการขยายขนาดของระบบจะสั่งให้เครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องดังกล่าวปิดการทำงานลง และเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชันที่ได้ใช้จำลองเครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องนั้น จากเดิมที่มีสถานะ Standby ให้มีสถานะเป็น Idle แทน

4) เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบระบบว่า ในขณะนั้นมีจำนวน N_BELOW มากกว่าหรือเท่ากับสองเครื่องหรือไม่

   a. ในกรณีที่ค่าของ N_BELOW มีจำนวนไม่ถึงสองเครื่อง เครื่องที่ควบคุมการขยายขนาดของระบบจะยังไม่ทำการลดขนาดของระบบเว็บแอปพลิเคชันลง โดยถือว่าระบบเว็บแอปพลิเคชันในขณะนั้นยังมีการใช้งานในปริมาณที่มากอยู่ เนื่องจากยังมีการใช้บริการของในแต่ละเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในปริมาณที่มากกว่าค่าขอบเขตล่างสุดที่ได้กำหนดไว้

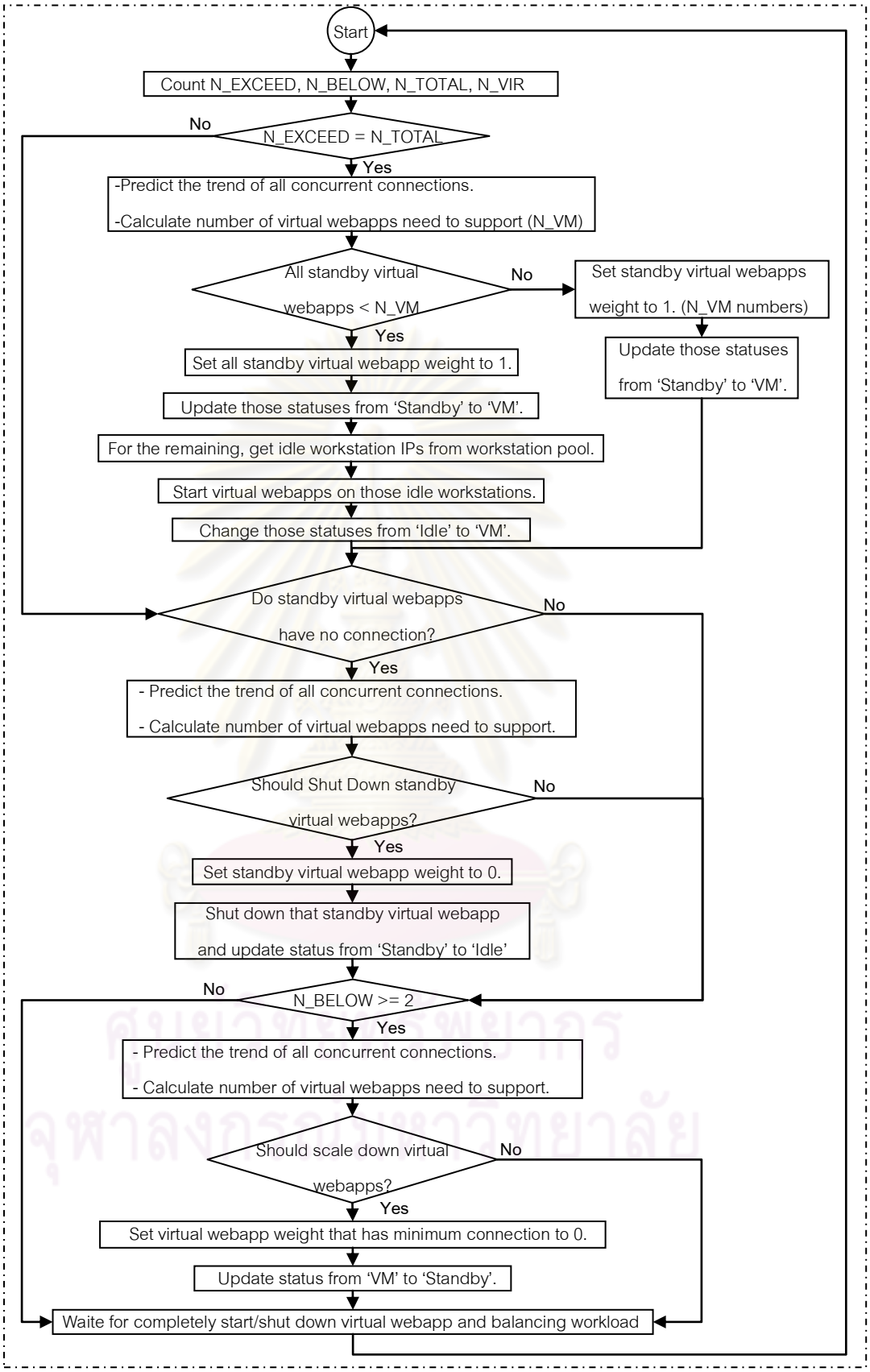
   b. ในกรณีที่ค่าของ N_BELOW มีจำนวนมากกว่าหรือเท่ากับ 2 เครื่อง เครื่องที่ควบคุมการขยายขนาดของระบบจะเริ่มทำการลดขนาดของระบบเว็บแอปพลิเคชันลง โดยมีขั้นตอนดังต่อไปนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- เครื่องที่ควบคุมการขยายขนาดของระบบจะส่งข้อมูลไปยังเครื่องที่ทำหน้าที่กระจายภาระงานให้กับระบบเว็บแอปพลิเคชัน ให้ทำการเปลี่ยนแปลงค่าน้ำหนักของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่ในขณะนั้นมีปริมาณการเชื่อมต่อกับผู้ใช้งานน้อยที่สุด ให้มีค่าน้ำหนักเท่ากับศูนย์ เพื่อที่จะได้ไม่ต้องส่งภาระงานใหม่ให้แก่เครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องดังกล่าว
- เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบจะทำการเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชันที่ได้ใช้จำลองเป็นเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนเครื่องดังกล่าว จากเดิมที่มีสถานะเป็น VM ให้มีสถานะเป็น Standby เพื่อแสดงว่ากำลังจะคืนทรัพยากรให้กับระบบ ทั้งนี้ยังไม่ทำการปิดการทำงานลง เนื่องจากได้เผื่อเวลาให้เครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องดังกล่าวสามารถให้บริการแก่การเชื่อมต่อที่อาจจะยังคงค้างการให้บริการอยู่

5) ระบบหยุดรอเวลาให้เครื่องแอปพลิเคชันเซิร์ฟเวอร์สามารถเปิดการให้บริการ (Start) หรือปิดการทำงานลง (Shut Down) ได้อย่างสมบูรณ์ รวมถึงรอเวลาให้ภาระงานได้กระจายให้กับแต่ละแอปพลิเคชันเซิร์ฟเวอร์ในระบบให้อยู่ในสภาวะสมดุลก่อนจะเริ่มทำงานในขั้นตอนที่ 1 ใหม่อีกครั้ง (เพื่อป้องกันการปรับขนาดของระบบเว็บแอปพลิเคชันในครั้งต่อไปที่อาจจะเกิดความผิดพลาดได้)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.5 แสดงแผนภาพขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะที่ละหลายเครื่องโดยใช้วิธีการคาดการณ์ล่วงหน้า

3.3.2 ขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยใช้วิธีการคาดการณ์ล่วงหน้า (Dynamic Scaling Algorithm: Scale Predict)

1) เริ่มต้นเครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบค่าต่างๆ ดังนี้

   a. N_EXCEED คือ จำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในระบบที่มีจำนวน แอ็กทีพคอนเน็กชันเกินกว่าค่าขอบเขตบนสุด (Upper-threshold) ที่ได้กำหนดไว้ในขณะนั้น

   b. N_BELOW คือจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในระบบที่มีจำนวน แอ็กทีพคอนเน็กชันต่ำกว่าค่าขอบเขตล่างสุด (Lower-threshold) ที่ได้กำหนดไว้ในขณะนั้น

   c. N_TOTAL คือ จำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ทั้งหมดในระบบในขณะนั้น

   d. N_VIR คือ จำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนที่ใช้เป็นทรัพยากรเสริมให้กับระบบและกำลังเปิดให้บริการอยู่ในขณะนั้น

2) เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบว่าจำนวน N_EXCEED ในขณะนั้นมีค่าเท่ากับจำนวน N_TOTAL หรือไม่

   a. ในกรณีที่จำนวน N_EXCEED มีค่าไม่เท่ากับจำนวน N_TOTAL แสดงว่าระบบเว็บแอปพลิเคชันในปัจจุบันยังมีทรัพยากรเพียงพอที่จะใช้รองรับปริมาณภาระงานในขณะนั้น ดังนั้นเครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบจะยังไม่ทำการขยายขนาดของระบบเว็บแอปพลิเคชัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

b. ในกรณีที่จำนวน N_EXCEED มีค่าเท่ากับจำนวน N_TOTAL หมายความว่าทรัพยากรทั้งหมดในระบบเว็บแอปพลิเคชันเริ่มจะไม่สามารถรองรับปริมาณภาระงานในขณะนั้นได้ ดังนั้นเครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบจะเริ่มทำการขยายขนาดของระบบเว็บแอปพลิเคชัน โดยมีขั้นตอนดังนี้คือ

    - เครื่องที่ควบคุมการขยายขนาดของระบบจะทำการคาดการณ์แนวโน้มของปริมาณแอ็กทีพคอนเน็กชันที่ระบบจะต้องให้บริการในอนาคต โดยใช้วิธีการวิเคราะห์การถดถอยเชิงเส้นอย่างง่ายในการวิเคราะห์
    - หลังจากนั้นระบบจะคำนวณจำนวนทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือน ว่าจะต้องเพิ่มจำนวนอีกกี่เครื่องเพื่อใช้รองรับปริมาณแอ็กทีพคอนเน็กชันจากที่ได้คาดการณ์ไว้แล้ว ในที่นี้ใช้เป็นค่า N_VM เครื่อง
    - เครื่องที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการเปรียบเทียบว่า ในปัจจุบันระบบมีปริมาณเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby น้อยกว่าปริมาณ N_VM ที่ได้จากการคำนวณหรือไม่
        - ในกรณีที่ปริมาณของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby มีปริมาณมากกว่าหรือเท่ากับค่า N_VM

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- เครื่องที่ควบคุมการขยายขนาดของระบบ จะทำการส่งข้อมูลไปยังเครื่องที่ทำหน้าที่กระจายภาระงานให้กับระบบเว็บแอปพลิเคชัน ให้ทำการเปลี่ยนแปลงค่าน้ำหนักของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในระบบที่มีสถานะเป็น Standby ดังกล่าวจำนวน N_VM เครื่อง จากเดิมที่มีค่าน้ำหนักเท่ากับ 0 ให้มีค่าเท่ากับ 1 เพื่อให้สามารถรองรับปริมาณภาระงานใหม่ที่จะเข้ามาในระบบได้

- เครื่องที่ควบคุมการขยายขนาดของระบบ จะทำการเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชันที่ได้ใช้จำลองเป็นเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนดังกล่าว จากเดิมที่มีสถานะเป็น Standby ให้มีสถานะเป็น VM

- ในกรณีที่ปริมาณของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby มีปริมาณน้อยกว่าค่า N_VM

  - เครื่องที่ควบคุมการขยายขนาดของระบบจะทำการส่งข้อมูลไปยังเครื่องที่ทำหน้าที่กระจายภาระงานให้กับระบบเว็บแอปพลิเคชัน ให้ทำการเปลี่ยนแปลงค่าน้ำหนักของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby ทั้งหมดในระบบจากเดิมที่มีค่าน้ำหนักเท่ากับ 0 ให้มีค่าเท่ากับ 1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

เพื่อให้สามารถใช้รองรับปริมาณภาระงานใหม่ที่จะเข้ามาในระบบได้

- และทำการเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชันที่ได้ใช้จำลองเป็นเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนเหล่านั้น จากเดิมที่มีสถานะเป็น Standby ให้มีสถานะเป็น VM
- สำหรับทรัพยากรในส่วนที่เหลือ เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบ จะทำการค้นหาหมายเลขที่อยู่ไอพีของเครื่องเวิร์กสเตชันที่มีสถานะเป็น Idle ในเวิร์กสเตชันพลู จำนวน n เครื่อง (โดยที่ n = N_VM – จำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ทั้งหมดที่มีสถานะเป็น Standby)
- หลังจากนั้นเครื่องที่ควบคุมการขยายขนาดของระบบ จะทำการจำลองเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนบนเครื่องเวิร์กสเตชันเหล่านั้นพร้อมกัน เพื่อใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชัน
- เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบ จะทำการเปลี่ยนสถานะของเครื่องเวิร์กสเตชันที่ได้ใช้จำลองเครื่องแอปพลิเคชันเสมือนเหล่านั้น จากเดิมที่มีสถานะเป็น Idle ให้มีสถานะเป็น VM เพื่อแสดงว่ากำลัง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ให้บริการแก่ผู้ใช้งานในระบบเว็บแอปพลิเคชัน

3) เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบจะทำการตรวจสอบว่า ในขณะนั้นระบบมีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby และไม่มีการเชื่อมต่อกับผู้ใช้งานอยู่บ้างหรือไม่

   a. ในกรณีที่ไม่มีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะดังกล่าว แสดงว่าในขณะนั้นระบบอาจไม่มีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะ Standby หรือเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะ Standby ในระบบกำลังให้บริการแก่ผู้ใช้งานอยู่ ซึ่งไม่ควรที่จะปิดการทำงานลง (Shut Down)

   b. ในกรณีที่ระบบมีเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะดังกล่าว เครื่องที่ควบคุมการขยายขนาดของระบบจะปิดการทำงานของเครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องนั้นลงโดยมีขั้นตอนดังนี้คือ

      - เครื่องที่ควบคุมการขยายขนาดของระบบจะทำการคาดการณ์แนวโน้มของปริมาณแอ็กทีพคอนเน็กชันที่ระบบเว็บแอปพลิเคชันจะต้องให้บริการในอนาคตโดยใช้วิธีการวิเคราะห์การถดถอยเชิงเส้นอย่างง่ายในการวิเคราะห์ หลังจากนั้นระบบจะทำการคำนวณจำนวนทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์ ที่จะใช้สำหรับรองรับปริมาณแอ็กทีพคอนเน็กชันจากที่ได้คาดการณ์ไว้

      - จากข้อมูลดังกล่าว เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชัน จะสามารถตัดสินใจได้ว่าควรที่จะปิดการทำงานในส่วนของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby เหล่านั้นหรือไม่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- ในกรณีที่ปริมาณภาระงานของระบบมีแนวโน้มเพิ่มสูงขึ้นและอาจจะต้องใช้ทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์ซึ่งในขณะนี้มีสถานะเป็น Standby เพื่อใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันอีกในอนาคต เครื่องที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันก็จะยังไม่ปิดการทำงานของเครื่องแอปพลิเคชันเซิร์ฟเวอร์เหล่านั้นลง

- ในกรณีที่ปริมาณภาระงานของระบบมีแนวโน้มลดลงหรือคงที่ ซึ่งระบบไม่จำเป็นต้องใช้ทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เหล่านี้มาใช้เป็นทรัพยากรเสริมอีกในอนาคต เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบก็จะปิดการทำงานของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่มีสถานะเป็น Standby เหล่านั้นลง โดยมีขั้นตอนดังต่อไปนี้ คือ

    - เครื่องที่ควบคุมการขยายขนาดของระบบ จะทำการส่งข้อมูลไปยังเครื่องที่ทำหน้าที่กระจายภาระงานให้กับระบบเว็บแอปพลิเคชันให้ทำการเปลี่ยนแปลงค่าน้ำหนักของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ดังกล่าวให้มีค่าเท่ากับศูนย์

    - หลังจากนั้นเครื่องที่ควบคุมการขยายขนาดของระบบจะสั่งให้เครื่องแอปพลิเคชันเซิร์ฟเวอร์ดังกล่าวปิดการทำงานลง และเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชันที่ใช้จำลอง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

เครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องนั้น จากเดิมที่มีสถานะ Standby ให้มีสถานะเป็น Idle แทน

4) เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันจะทำการตรวจสอบระบบว่า ในขณะนั้นมีจำนวน N_BELOW มากกว่าหรือเท่ากับสองเครื่องหรือไม่

   a. ในกรณีที่ค่าของ N_BELOW มีจำนวนไม่ถึงสองเครื่อง เครื่องที่ควบคุมการขยายขนาดของระบบจะยังไม่ทำการลดขนาดของระบบเว็บแอปพลิเคชันลง โดยถือว่าระบบเว็บแอปพลิเคชันในขณะนั้นยังมีการใช้งานในปริมาณที่มากอยู่ เนื่องจากยังมีการใช้บริการของในแต่ละแอปพลิเคชันเซิร์ฟเวอร์ในปริมาณที่มากกว่าค่าขอบเขตล่างสุดที่ได้กำหนดไว้

   b. ในกรณีที่ค่าของ N_BELOW มีจำนวนมากกว่าหรือเท่ากับ 2 เครื่อง เครื่องที่ควบคุมการขยายขนาดของระบบจะเริ่มทำการลดขนาดของระบบเว็บแอปพลิเคชันลง โดยมีขั้นตอนดังต่อไปนี้

      - เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบ จะทำการคาดการณ์แนวโน้มของปริมาณแอ็กทีพคอนเน็กชัน ที่ระบบจะต้องให้บริการในอนาคต โดยใช้วิธีการวิเคราะห์การถดถอยเชิงเส้นอย่างง่ายในการวิเคราะห์ หลังจากนั้นระบบจะทำการคำนวณจำนวนทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์ ที่จะใช้สำหรับรองรับปริมาณแอ็กทีพคอนเน็กชันจากที่ได้คาดการณ์ไว้

      - จากข้อมูลดังกล่าว เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชัน จะสามารถตัดสินใจได้ว่าควรที่จะลดปริมาณเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในระบบลงหรือไม่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- ในกรณีที่ปริมาณภาระงานของระบบมีแนวโน้มเพิ่มสูงขึ้นและอาจจะต้องใช้ทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์ดังกล่าวเป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันอีกในอนาคต เครื่องที่ควบคุมการขยายขนาดของระบบเว็บแอปพลิเคชันก็จะยังไม่ทำการลดขนาดของระบบเว็บแอปพลิเคชันลง

- ในกรณีที่ปริมาณภาระงานของระบบมีแนวโน้มลดลงหรือคงที่ ซึ่งระบบไม่จำเป็นต้องใช้ทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เหล่านี้มาใช้เป็นทรัพยากรเสริมอีกในอนาคต เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบก็จะทำการลดขนาดของระบบเว็บแอปพลิเคชันลง โดยมีขั้นตอนดังต่อไปนี้ คือ

    - เครื่องที่ควบคุมการขยายขนาดของระบบจะทำการส่งข้อมูลไปยังเครื่องที่ทำหน้าที่กระจายภาระงานให้กับระบบเว็บแอปพลิเคชัน ให้ทำการเปลี่ยนแปลงค่าน้ำหนักของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่ในขณะนั้นมีปริมาณการเชื่อมต่อกับผู้ใช้งานน้อยที่สุด ให้มีค่าน้ำหนักเท่ากับศูนย์ เพื่อที่จะได้ไม่ต้องส่งภาระงานใหม่ให้แก่เครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องดังกล่าว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- เครื่องที่ทำหน้าที่ควบคุมการขยายขนาดของระบบจะทำการเปลี่ยนแปลงสถานะของเครื่องเวิร์กสเตชันที่ได้ใช้จำลองเป็นเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนเครื่องดังกล่าว จากเดิมที่มีสถานะเป็น VM ให้มีสถานะเป็น Standby เพื่อแสดงว่ากำลังจะคืนทรัพยากรให้กลับระบบ ทั้งนี้ยังไม่ทำการปิดการทำงานลง เนื่องจากได้เผื่อเวลาให้เครื่องแอปพลิเคชันเซิร์ฟเวอร์เครื่องดังกล่าวสามารถให้บริการแก่การเชื่อมต่อที่อาจจะยังคงค้างการให้บริการอยู่

5) ระบบหยุดรอเวลาให้เครื่องแอปพลิเคชันเซิร์ฟเวอร์สามารถเปิดการให้บริการ (Start) หรือปิดการทำงานลง (Shut Down) ได้อย่างสมบูรณ์ รวมถึงรอเวลาให้ภาระงานได้กระจายให้กับแต่ละแอปพลิเคชันเซิร์ฟเวอร์ในระบบให้อยู่ในสภาวะสมดุลก่อนจะเริ่มทำงานในขั้นตอนที่ 1 ใหม่อีกครั้ง (เพื่อป้องกันการปรับขนาดของระบบเว็บแอปพลิเคชันในครั้งต่อไปที่อาจจะเกิดความผิดพลาดได้)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การทดสอบและผลการทดสอบงานวิจัย

ในบทนี้จะทำการทดสอบและวิเคราะห์สมรรถนะของระบบเว็บแอปพลิเคชันที่เกิดจากการนำทฤษฎีที่ได้นำเสนอในงานวิจัยนี้มาทำให้เกิดผล (Implementation) โดยจะขอเริ่มจากการกำหนดตัววัดสมรรถนะของระบบเว็บแอปพลิเคชัน (Web Application Performance Metrics) ที่จะใช้ในงานวิจัยนี้ก่อน หลังจากนั้นผู้วิจัยได้แบ่งประเภทของการทดสอบประสิทธิภาพของระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่แตกต่างกัน ได้แก่ การทดสอบเพื่อวัดประสิทธิภาพในการรองรับปริมาณภาระงานของแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบ การทดสอบเพื่อเปรียบเทียบประสิทธิภาพของวิธีการจัดสรรทรัพยากรในระบบเว็บแอปพลิเคชันในสภาพแวดล้อมของลักษณะภาระงานที่แตกต่างกัน คือ ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก และในสภาพแวดล้อมที่ภาระงานมีแนวโน้มอัตราการเข้าใช้งานในลักษณะที่เพิ่มขึ้นอย่างรวดเร็ว รวมถึงในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน โดยอ้างอิงลักษณะของภาระงานดังกล่าวจากงานวิจัย [5] และ [19] ตามที่ได้กล่าวไว้แล้วในหัวข้อที่ 2.1.4.2

สำหรับวิธีการจัดสรรทรัพยากรที่จะใช้ในการทดสอบเพื่อเปรียบเทียบประสิทธิภาพในแต่ละสภาพแวดล้อมนั้นจะใช้วิธีการจัดสรรทรัพยากรแบบสถิต (No Scale) วิธีการจัดสรรทรัพยากรโดยใช้วิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง (Scale Each) และวิธีการจัดสรรทรัพยากรโดยใช้วิธีในการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า (Scale Predict) ตามลำดับ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.1 ตัววัดสมรรถนะของระบบ (Performance Metrics)

สำหรับในงานวิจัยนี้ได้ทำการวัดสมรรถนะของระบบเว็บแอปพลิเคชันใน 2 ด้าน ด้วยกันคือ

1) ปริมาณงานโดยเฉลี่ยของระบบ (Average Throughput)

2) ค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งาน (Average Response Time)

สำหรับปริมาณงานโดยเฉลี่ยของระบบที่สามารถประมวลผลได้นั้นจะทำการวัดจากอัตราการตอบกลับ (Reply Rate) โดยเฉลี่ยที่ระบบได้ตอบกลับให้กับผู้ใช้งานในหนึ่งวินาที (replies/s) ซึ่งค่าปริมาณงานโดยเฉลี่ยของระบบที่มีค่ามากย่อมแสดงว่าระบบดังกล่าวมีประสิทธิภาพในการประมวลผลที่ดี ส่วนค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานคือ ค่าเฉลี่ยของระยะเวลาที่ผู้ใช้งานใช้ในการรอการตอบสนองจากระบบในหน่วยมิลลิวินาที (ms) ซึ่งจะมีความสัมพันธ์โดยตรงกับความพอใจในการใช้งานของผู้ใช้งานด้วย โดยระบบที่มีค่าเฉลี่ยของระยะเวลาที่ใช้ในการตอบสนองของระบบต่อผู้ใช้งานในค่าที่น้อย ย่อมแสดงว่าระบบดังกล่าวให้ประสิทธิภาพที่ดีกว่า

4.2 การทดสอบเพื่อวัดประสิทธิภาพในการรองรับปริมาณภาระงานของแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบ

ในการทดลองนี้มีจุดประสงค์เพื่อต้องการวัดประสิทธิภาพของแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบ ว่าในช่วงใดที่เว็บแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับปริมาณภาระงานได้ ซึ่งจะทำให้ทราบถึงปริมาณแอ็กทีพคอนเน็กชันมากที่สุด (Maximum Active Concurrent Connection) ที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับได้ด้วย รวมถึงข้อมูลดังกล่าวจะเป็นประโยชน์ต่อการใช้เป็นข้อมูลอ้างอิงเพื่อใช้ในการขยายขนาดของระบบเว็บแอปพลิเคชันได้ต่อไป

4.2.1 เครื่องมือที่ใช้ในการทดลอง

1) **เครื่องที่ใช้จำลองภาระงานให้แก่ระบบ** (Client Emulator)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)

- ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
- หน่วยความจำหลัก 1 กิกะไบต์
- เครือข่ายแลน (LAN bandwidth 100 MB/s)

○ **ส่วนชุดคำสั่ง (Software)**

- ระบบปฏิบัติการ CentOS 5.5
- Httperf 0.9.0

2) **เครื่องแอปพลิเคชันเซิร์ฟเวอร์ของระบบ (Main Application Server)**

○ **ส่วนอุปกรณ์ (Hardware) และเครือข่าย (Network)**

- 1 ซีพียูเสมือน (1 Virtual CPU)
- หน่วยความจำเสมือน 450 เมกะไบต์
- ต่อประสานแบบบริดจ์ (Bridge network interface)
- เครือข่ายแลน (LAN bandwidth 100 MB/s)

○ **ส่วนชุดคำสั่ง (Software)**

- ระบบปฏิบัติการ CentOS5.5
- Apache 2.2.3
- Tomcat 5.5.23
- Java 1.6.0_17

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.2.2 สภาพแวดล้อมและวิธีการทดลอง

ระบบเว็บแอปพลิเคชันในการทดลองนี้จะประกอบด้วยเครื่องเว็บแอปพลิเคชันเซิร์ฟเวอร์จำนวนหนึ่งเครื่องติดต่อกับเครื่องที่ใช้จำลองภาระงาน เริ่มต้นเครื่องที่ทำหน้าที่จำลองภาระงานจะทำการจำลองภาระงานเข้าสู่ระบบเว็บแอปพลิเคชันดังกล่าว โดยปริมาณภาระงานที่ใช้ในการทดลองจะค่อยๆ เพิ่มขึ้นจากปริมาณน้อยไปหามาก คือจากระดับ 0.2, 0.4, 0.6, 0.8 และเพิ่มค่าขึ้นเรื่อยๆ จนถึง 10 ยูสเซอร์เซสชันต่อวินาที เพื่อพยายามหาว่าระดับภาระงานในช่วงใดที่แอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถให้บริการได้ ซึ่งจะทำให้สามารถทราบถึงปริมาณค่าแอ็กทีพคอนเน็กชันมากที่สุดที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับได้ด้วย

4.2.3 ผลการทดสอบ

ตารางที่ 4.1 ผลการทดสอบประสิทธิภาพในการรองรับปริมาณภาระงานของแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์

Rate (user sessions/s)	Average Throughput (replies/s)	Average Reply Time (ms)	Maximum Active Concurrent Connection
0.2	1.9	82.2	5
0.4	3.8	82.5	9
0.6	5.7	82.4	13
0.8	7.6	82	17
1	9.5	81.4	21
1.2	11.4	82.9	25
1.4	13.4	82	29
1.6	15.3	82.5	33
1.8	17.2	82.6	37
2	19.1	82.6	41
2.2	21	82.6	45
2.4	22.9	83.4	49
2.6	24.8	83.3	53
2.8	26.7	83.1	57

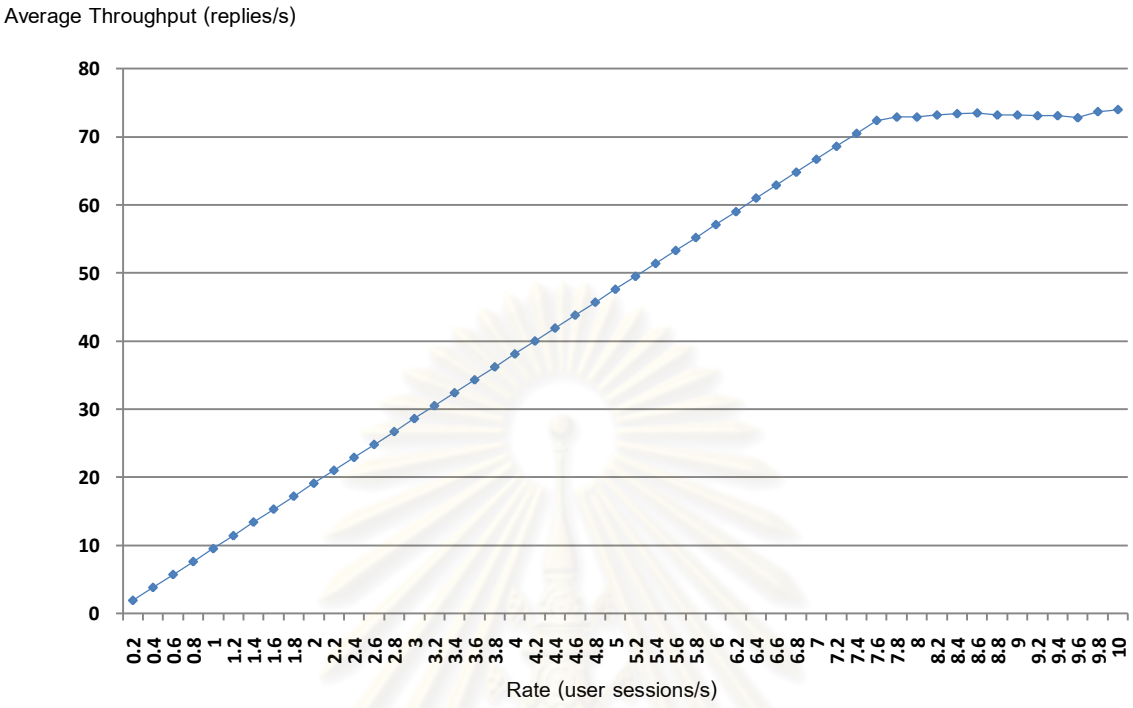
Rate (user sessions/s)	Average Throughput (replies/s)	Average Reply Time (ms)	Maximum Active Concurrent Connection
3	28.6	83.6	62
3.2	30.5	83.2	66
3.4	32.4	83.8	70
3.6	34.3	85	75
3.8	36.2	85.2	80
4	38.1	84.8	83
4.2	40	84.5	88
4.4	41.9	86	94
4.6	43.8	87.6	100
4.8	45.7	87.2	103
5	47.6	87.9	109
5.2	49.5	88.2	115
5.4	51.4	91.1	120
5.6	53.3	90.7	124
5.8	55.2	91.4	128
6	57.1	96.9	136
6.2	59	97.3	142
6.4	61	98.1	144
6.6	62.9	105.8	154
6.8	64.8	110.3	157
7	66.7	114.6	164
7.2	68.6	130.9	172
7.4	70.5	173.5	175
7.6	72.4	235.4	186
7.8	72.9	559	220
8	72.9	781.6	290

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

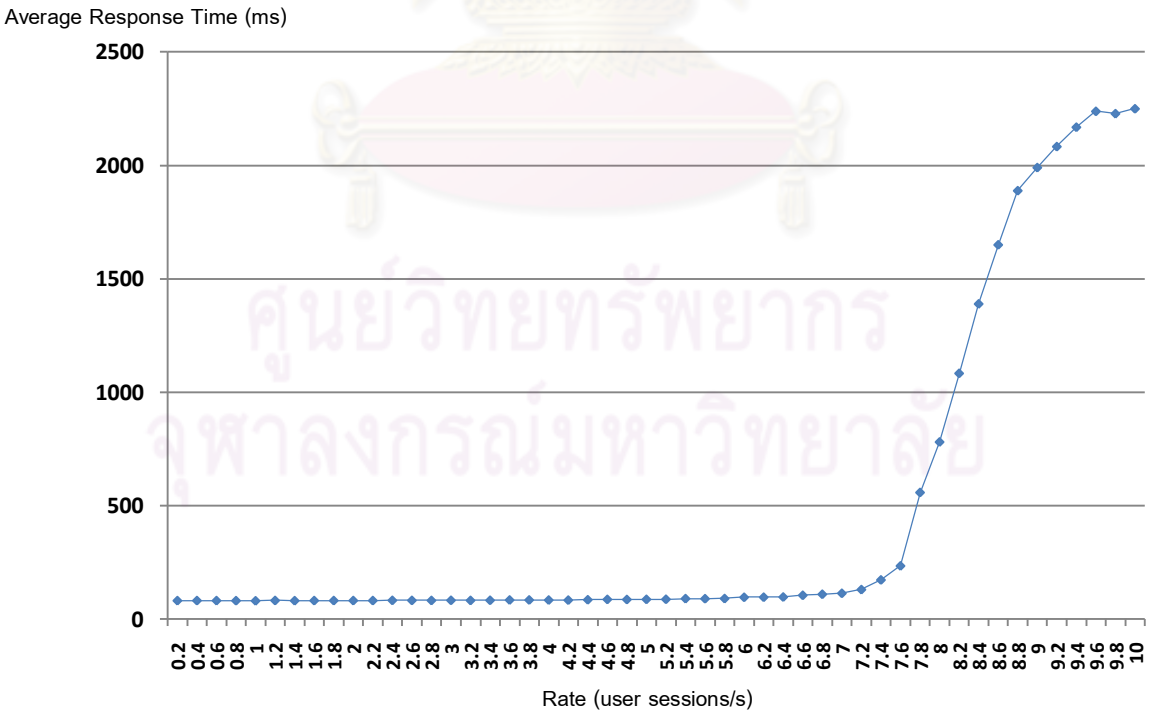
Rate (user sessions/s)	Average Throughput (replies/s)	Average Reply Time (ms)	Maximum Active Concurrent Connection
8.2	73.2	1084.2	357
8.4	73.4	1390.3	417
8.6	73.5	1650.7	465
8.8	73.2	1889.3	507
9	73.2	1991.3	562
9.2	73.1	2083.7	612
9.4	73.1	2169.2	662
9.6	72.8	2239.2	711
9.8	73.7	2228.9	737
10	74	2250.5	785

เนื่องจากค่าปริมาณงานโดยเฉลี่ยของระบบและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานจากการทดลองถูกกำหนดให้เป็นตัววัดสมรรถนะของระบบ ซึ่งแสดงถึงประสิทธิภาพโดยรวมของระบบเว็บแอปพลิเคชันในการรองรับปริมาณภาระงานดังกล่าว ดังนั้น เมื่อนำค่าปริมาณงานโดยเฉลี่ยของระบบ และค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานจากตารางที่ 4.1 มาแสดงเป็นกราฟสามารถแสดงได้ดังรูปที่ 4.1 และ 4.2

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 แสดงปริมาณงานโดยเฉลี่ยที่แต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ใช้ในการตอบสนองต่อผู้ใช้งาน



รูปที่ 4.2 แสดงค่าเฉลี่ยระยะเวลาที่แต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ใช้ในการตอบสนองต่อผู้ใช้งาน

4.2.4 วิเคราะห์ผลการทดลอง

จากผลการทดลอง ในรูปที่ 4.1 และ 4.2 จะเห็นได้ว่าในช่วงที่ภาระงานที่มีอัตราการเข้าใช้งานในระบบเว็บแอปพลิเคชันเท่ากับ 7.8 ยูสเซอร์เซสชันต่อวินาที เว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบเริ่มไม่สามารถรองรับปริมาณภาระงานที่มีปริมาณมากเหล่านั้นได้ สังเกตได้จากในรูปที่ 4.2 ค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งาน ที่มีค่าเพิ่มสูงขึ้นอย่างรวดเร็วตั้งแต่ในช่วงที่ภาระงานที่มีอัตราการเข้าใช้งานเท่ากับ 7.8 ยูสเซอร์เซสชันต่อวินาที ซึ่งมีค่าต่างจากค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานในช่วงก่อนหน้านี้อย่างชัดเจน และสอดคล้องกับค่าของปริมาณงานโดยเฉลี่ยของระบบในรูปที่ 4.1 ที่เริ่มจะคงที่ตั้งแต่ในในช่วงที่ภาระงานที่มีอัตราการเข้าใช้งานเท่ากับ 7.8 ยูสเซอร์เซสชันต่อวินาที เป็นต้นไป

สำหรับอัตราการเข้าใช้งานในระดับดังกล่าว สามารถวัดปริมาณแอ็กทีพคอนเน็กชันมากที่สุดได้ประมาณ 220 การเชื่อมต่อที่พร้อมกัน (Active Concurrent Connections) และเนื่องจากในงานวิจัยนี้ได้กำหนดให้ทุกๆแอปพลิเคชันเซิร์ฟเวอร์มีลักษณะที่ไม่แตกต่างกัน (Homogeneous) เพราะฉะนั้นจากการทดลองนี้ทำให้สามารถทราบได้ว่าแต่ละเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบจะเริ่มไม่สามารถรองรับภาระงานที่มีปริมาณแอ็กทีพคอนเน็กชันมากที่สุดประมาณ 220 การเชื่อมต่อที่พร้อมกันด้วย

4.3 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพของวิธีการจัดสรรทรัพยากรในระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อย ๆ เพิ่มขึ้นในปริมาณที่ไม่มาก

ในการทดลองนี้มีจุดประสงค์เพื่อต้องการเปรียบเทียบประสิทธิภาพของระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรในลักษณะที่แตกต่างกันคือ ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต (No Scale) ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละเครื่อง (Scale Each) และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า (Scale Predict) ตามลำดับ บนสภาพแวดล้อมที่ภาระงานของระบบเว็บแอปพลิเคชันมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก โดยจะใช้ลักษณะภาระงานที่อ้างอิงจากงานวิจัย [5] ตามที่ได้อธิบายไว้ในหัวข้อที่ 2.1.4.2

4.3.1 เครื่องมือที่ใช้ในการทดลอง

1) **เครื่องที่ใช้จำลองภาระงานให้แก่ระบบ** (Client Emulator)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Httperf 0.9.0

2) **เครื่องที่ใช้ในการกระจายภาระงาน** (Load Balancer)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - PHP 5.1.6
  - Mysql-Client 5.0.77

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- UltraMonkey 3
  - IP Virtual Server version 1.2.1
  - heartbeat-2.1.3
  - Linux Director v1.186-ha-2.1.3

3) **เครื่องที่ใช้ในการจัดการระบบเว็บแอปพลิเคชัน** (Manager)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Apache 2.2.3
  - PHP 5.3.5
  - Mysql-Server 5.0.77

4) **เครื่องเวิร์กสเตชันในเวิร์กสเตชันพลู** (Workstation Pool)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - VirtualBox 4.0.0 hyperisor

5) **เครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักของระบบ** (Main Application Server)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - 1 ซีพียูเสมือน (1 Virtual CPU)
  - หน่วยความจำเสมือน 450 เมกะไบต์
  - ต่อประสานแบบบริดจ์ (Bridge network interface)
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Apache 2.2.3
  - Tomcat 5.5.23
  - Java 1.6.0_17

6) **เครื่องเวอร์ชวลแอปพลิเคชันเซิร์ฟเวอร์ที่ใช้จำลองขึ้นในระบบ** (Virtual Application Server)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - 1 ซีพียูเสมือน (1 Virtual CPU)
  - หน่วยความจำเสมือน 450 เมกะไบต์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- ต่อประสานแบบบริดจ์ (Bridge network interface)
- เครือข่ายแลน (LAN bandwidth 100 MB/s)

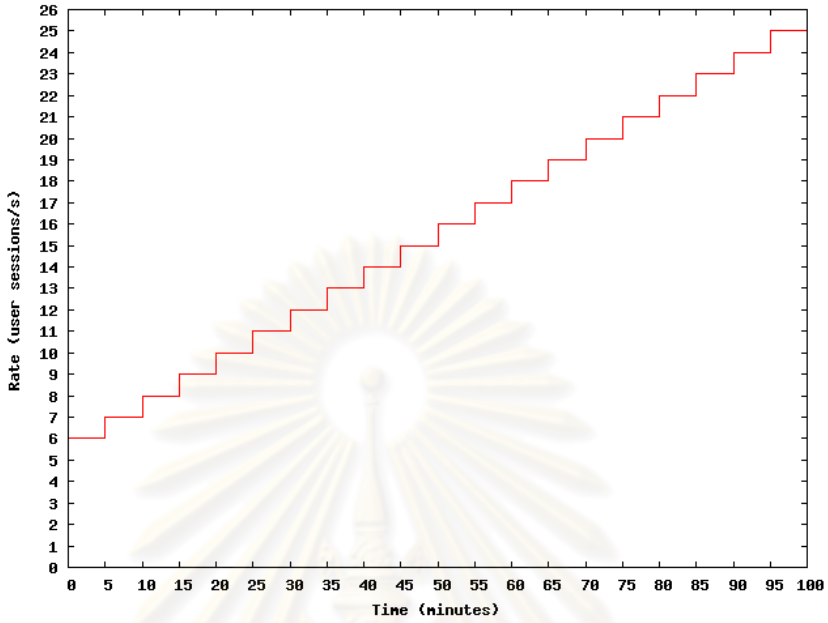
○ **ส่วนชุดคำสั่ง** (Software)

- ระบบปฏิบัติการ CentOS 5.5
- Apache 2.2.3
- Tomcat 5.5.23
- Java 1.6.0_17

4.3.2 สภาพแวดล้อมและวิธีการทดลอง

ในการทดลองนี้เครื่องที่ทำหน้าที่จำลองภาระงานจะทำการสร้างภาระงานให้แก่ระบบในลักษณะแสดงดังรูปที่ 4.3 คือ เริ่มต้นในสภาวะปกติกำหนดให้ระบบเว็บแอปพลิเคชันมีอัตราการการเข้าใช้งานเท่ากับ 6 ยูสเซอร์เซสชันต่อวินาที เพื่อใช้เป็นตัวแทนของปริมาณภาระงานในช่วงปกติที่ระบบเว็บแอปพลิเคชันสามารถรองรับได้โดยไม่จำเป็นต้องทำการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ และหลังจากนั้นในทุกๆ 5 นาที ปริมาณอัตราการเข้าใช้งานของระบบก็จะค่อยๆ มีค่าเพิ่มขึ้นเรื่อยๆ จนกระทั้งอัตราการเข้าใช้งานของระบบเว็บแอปพลิเคชันมีค่าเท่ากับ 25 ยูสเซอร์เซสชันต่อวินาที (เนื่องจากปริมาณทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่ใช้ในการทดลองมีจำกัด)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

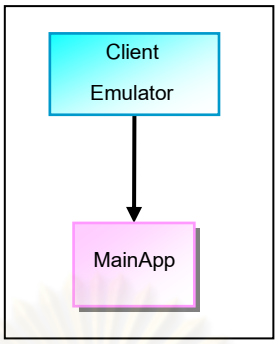


รูปที่ 4.3 แสดงลักษณะของภาระงานที่มีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก

สำหรับสภาพแวดล้อมของวิธีการจัดสรรทรัพยากรในระบบเว็บแอปพลิเคชันที่จะใช้ในการเปรียบเทียบประสิทธิภาพนั้นมีรายละเอียดดังนี้

4.3.2.1 ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต (Static Allocation)

ในสภาพแวดล้อมนี้จะแสดงถึงระบบเว็บแอปพลิเคชันที่ไม่ได้มีการปรับขนาดของแอปพลิเคชันเซิร์ฟเวอร์ คือจะเป็นลักษณะการอุทิศทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์หลักให้กับระบบเว็บแอปพลิเคชันตลอดเวลา โดยสภาพแวดล้อมสำหรับการทดลองสามารถแสดงการเชื่อมต่อได้ดังในรูปที่ 4.4

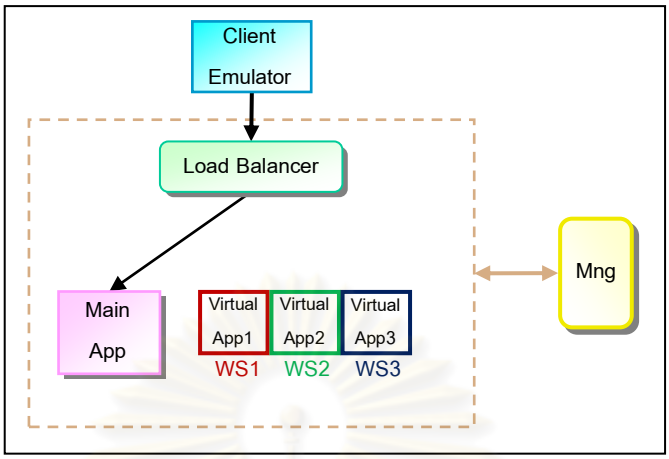


รูปที่ 4.4 แสดงสภาพแวดล้อมของระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต

จากรูปที่ 4.4 ได้ใช้เครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือน (Main App) จำนวน 1 เครื่องเป็นตัวแทนของทรัพยากรเว็บแอปพลิเคชันเซิร์ฟเวอร์หลักที่ได้อุทิศให้กับระบบเว็บแอปพลิเคชันเนื่องจากข้อจำกัดของอุปกรณ์เครื่องเซิร์ฟเวอร์จริงที่จะใช้ในการทดลอง และใช้เครื่องที่ทำหน้าที่จำลองภาระงานเพื่อสร้างภาระงานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆเพิ่มขึ้นในปริมาณที่ไม่มาก ตามที่ได้กล่าวมาแล้วในหัวข้อที่ 4.3.2 เข้าสู่ในระบบเว็บแอปพลิเคชันดังกล่าว

4.3.2.2 **ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละเครื่อง** (Scale Each)

ในสภาพแวดล้อมที่ใช้ในการทดลองนี้จะแสดงถึงระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง โดยสภาพแวดล้อมสำหรับการทดลองสามารถแสดงการเชื่อมต่อได้ดังในรูปที่ 4.5

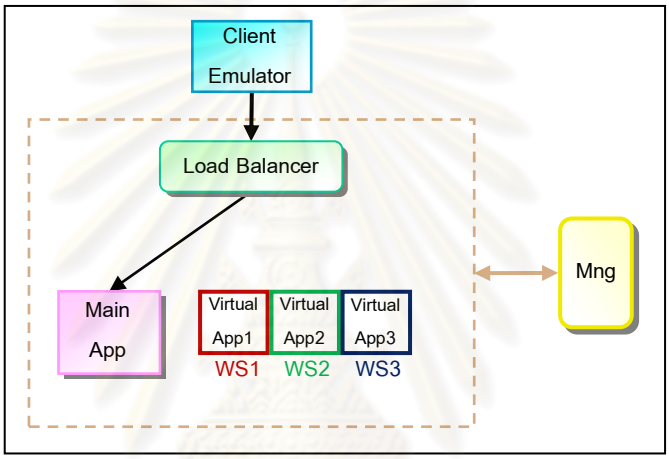


รูปที่ 4.5 แสดงสภาพแวดล้อมของระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง

จากรูปที่ 4.5 เครื่อง VirtualApp1, VirtualApp2 และ VirtualApp3 หมายถึง เครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนที่จะถูกจำลองขึ้นบนทรัพยากรเครื่องเวิร์กสเตชันที่มีอยู่แล้วในระบบและไม่ได้ถูกใช้งานอย่างเต็มที่ โดยในเบื้องต้นจะจำลองให้มีคุณลักษณะที่เหมือนกับเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลัก (MainApp) เริ่มแรกในระบบเว็บแอปพลิเคชันนี้จะมีทรัพยากรเฉพาะเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักเท่านั้น ในกรณีที่ภาระงานมีปริมาณมากขึ้นทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักอาจไม่เพียงพอที่จะใช้รองรับปริมาณภาระงานเหล่านั้น ส่วนของเครื่องที่ใช้ในการจัดการระบบเว็บแอปพลิเคชัน (Mng) ก็จะทำการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ โดยการจำลองทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนขึ้น (VirtualApp1, VirtualApp2, VirtualApp3) ในลักษณะทีละเครื่องตามปริมาณภาระงานในขณะนั้นเพื่อช่วยแบ่งเบาปริมาณภาระงานให้กับระบบเว็บแอปพลิเคชันดังกล่าว และในสภาพแวดล้อมนี้ได้กำหนดให้ค่าขอบเขตบนสุดและค่าขอบเขตล่างสุดที่จะใช้เป็นเกณฑ์ในการปรับขนาดของระบบเว็บแอปพลิเคชันตามที่ได้กล่าวไว้แล้วในหัวข้อที่ 3.3 ให้มีค่าเท่ากับ 90% และ 45% ตามลำดับ

4.3.2.3 ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานล่วงหน้า (Scale Predict)

ในสภาพแวดล้อมที่ใช้ในการทดลองนี้จะแสดงถึงระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ ในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของภาระงานที่จะเกิดขึ้นในอนาคต ซึ่งสามารถแสดงการเชื่อมต่อได้ดังในรูปที่ 4.6



รูปที่ 4.6 แสดงสภาพแวดล้อมของระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่อง

จากรูปที่ 4.6 สภาพแวดล้อมในส่วนของเครื่อง Load Balancer, MainApp, VirtualApp1, VirtualApp2 และ VirtualApp3 รวมถึงค่าขอบเขตบนสุดและค่าขอบเขตล่างสุดที่จะใช้เป็นเกณฑ์ในการปรับขนาดของระบบเว็บแอปพลิเคชันนั้นจะมีลักษณะเหมือนกับในสภาพแวดล้อมของระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องตามที่ได้กล่าวไว้ในหัวข้อที่ 4.3.2.2 ต่างเพียงแค่เครื่องที่ใช้ในการจัดการระบบเว็บแอปพลิเคชัน (Mng) จะใช้ขั้นตอนวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานที่จะเกิดขึ้นในอีก 5 นาทีข้างหน้าโดยใช้การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย แทนการใช้ขั้นตอนวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง

4.3.3 ผลการทดสอบ

สำหรับผลการทดลองค่าปริมาณงานโดยเฉลี่ยของระบบและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานในแต่ละระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรที่แตกต่างกันสามารถแสดงในตารางที่ 4.2 และ 4.3 ตามลำดับ

ตารางที่ 4.2 ปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก

Time(minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
0-5	6	57.7	57.7	57.7
5-10	7	67.3	67.3	67.3
10-15	8	73.8	75.2	74.9
15-20	9	86.6	81.1	75.6
20-25	10	96.2	94.7	78.4
25-30	11	105.8	105.8	77.4
30-35	12	115.4	115.4	79.3
35-40	13	125.0	125.0	76.7
40-45	14	134.5	134.5	75.3
45-50	15	144.0	143.9	78.6
50-55	16	153.0	151.9	75.2
55-60	17	163.5	163.5	74.6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

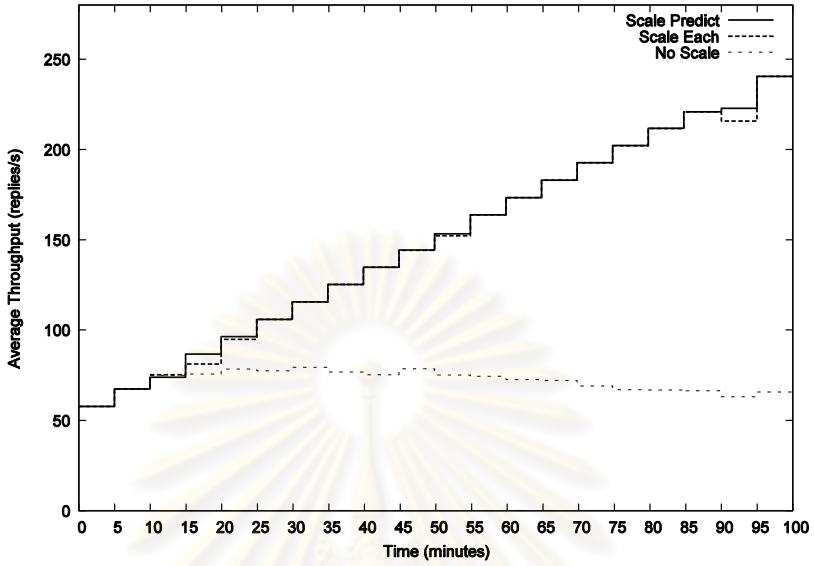
Time(minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
60-65	18	173.1	173.1	72.8
65-70	19	182.7	182.7	72.3
70-75	20	192.3	192.3	69.2
75-80	21	201.8	201.7	67.1
80-85	22	211.3	211.2	66.9
85-90	23	220.8	220.8	66.4
90-95	24	222.7	215.7	63.1
95-100	25	240.4	240.4	65.8

ตารางที่ 4.3 ค่าเฉลี่ยของระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณไม่มาก

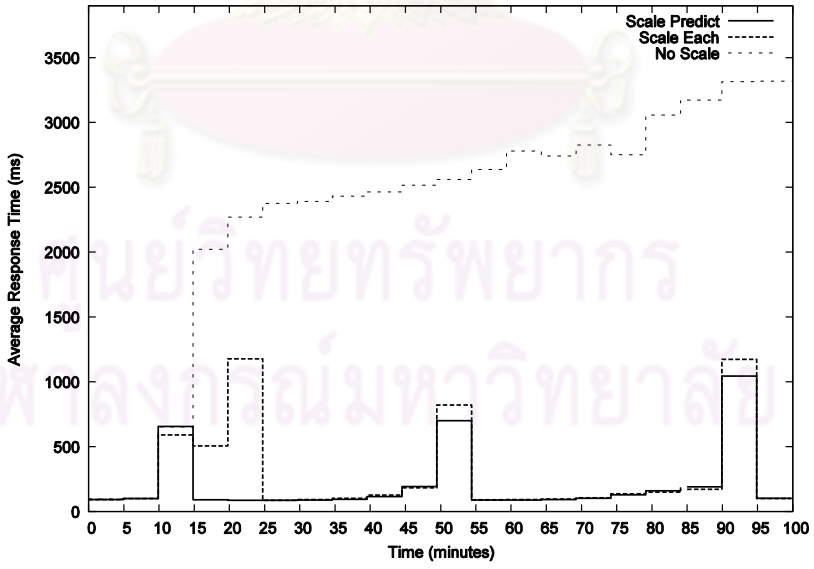
Time(minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
0-5	6	91.0	93.0	94.9
5-10	7	98.4	99.2	98.6
10-15	8	655.5	590.9	651.8
15-20	9	89.9	506.1	2020.6
20-25	10	86.5	1177.2	2269.9
25-30	11	87.2	85.8	2376.3

Time(minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
30-35	12	89.7	90.9	2391.8
35-40	13	95.5	99.2	2432.1
40-45	14	115.9	124.6	2464.7
45-50	15	192.4	181.0	2516.7
50-55	16	701.0	819.0	2560.7
55-60	17	90.0	89.0	2637.8
60-65	18	90.4	91.5	2780.8
65-70	19	94.7	96.1	2742.4
70-75	20	103.9	105.1	2827.9
75-80	21	130.9	135.7	2752.3
80-85	22	162.0	150.2	3057
85-90	23	189.4	171.5	3170.4
90-95	24	1044.3	1174.0	3313.8
95-100	25	102.2	100.1	3317.9

เมื่อนำค่าปริมาณงานโดยเฉลี่ยของระบบและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งาน จากตารางที่ 4.2 และ4.3 มาวาดกราฟสามารถแสดงได้ดังรูปที่ 4.7 และ 4.8

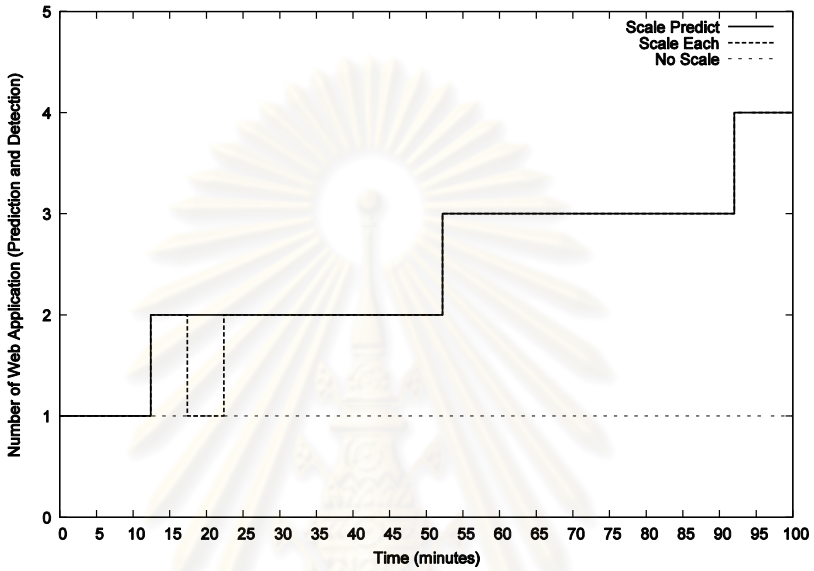


รูปที่ 4.7 กราฟแสดงปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก

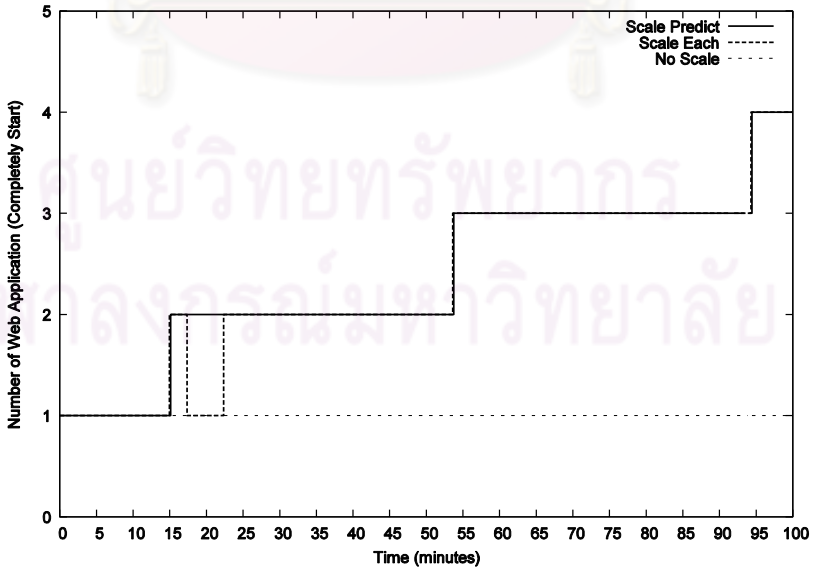


รูปที่ 4.8 กราฟแสดงค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก

สำหรับกราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบของระบบเว็บแอปพลิเคชัน (Prediction and Detection Graph) รวมถึงกราฟแสดงจำนวนเครื่องเว็บแอปพลิเคชันที่สามารถเปิดให้บริการได้อย่างสมบูรณ์ สามารถแสดงได้ดังรูปที่ 4.9 และ 4.10 ตามลำดับ

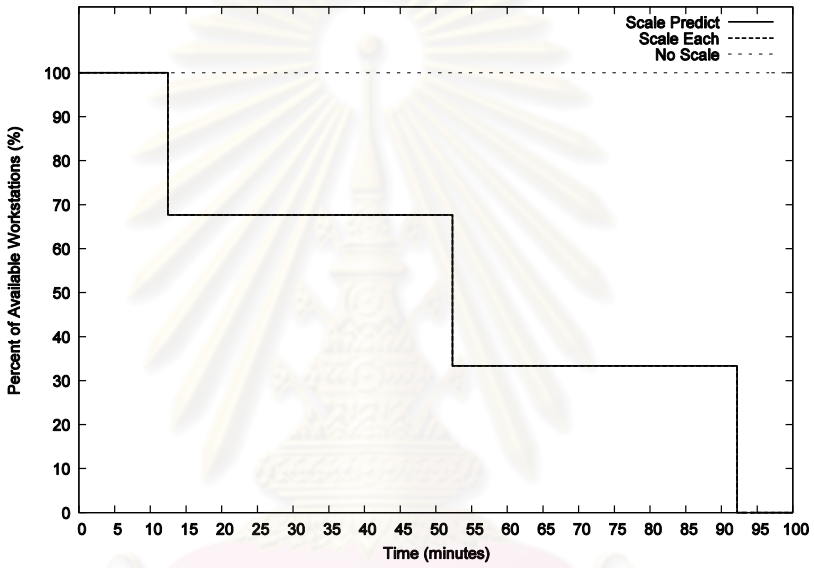


รูปที่ 4.9 กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก



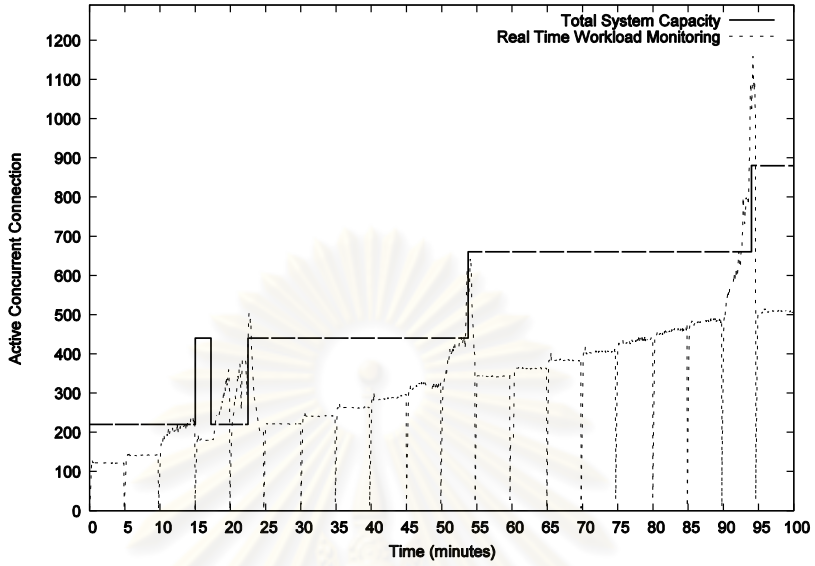
รูปที่ 4.10 กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่สามารถให้บริการได้อย่างสมบูรณ์ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก

สำหรับกราฟแสดงปริมาณเครื่องเวิร์กสเตชันในระบบที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้ และกราฟแสดงการเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชัน (Resources Utilization Graph) ระหว่างวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง และวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์ลักษณะภาระงานก่อนล่วงหน้า สามารถแสดงได้ดังรูปที่ 4.11 ถึง 4.14

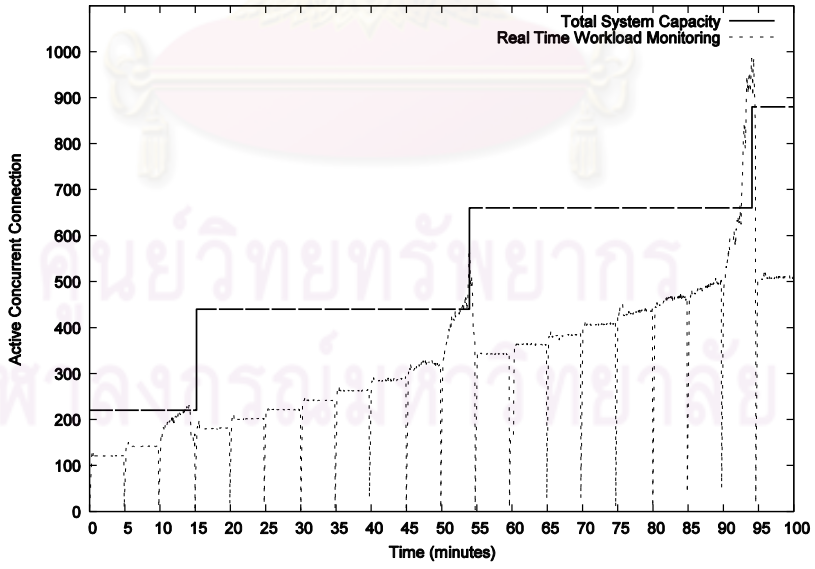


รูปที่ 4.11 กราฟแสดงปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเวิร์กสเตชันสามารถใช้บริการได้ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก

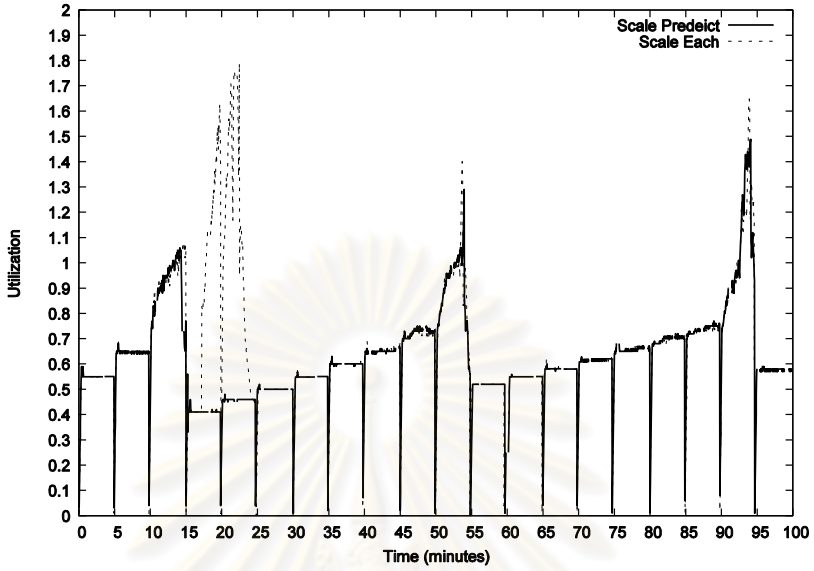
สำหรับรูปที่ 4.11 ค่าในแนวแกนวาย (Y) ใช้แสดงปริมาณเปอร์เซ็นต์ของเครื่องเวิร์กสเตชันในหน่วยงานที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้ และเนื่องจากในการทดลองได้ใช้เครื่องเวิร์กสเตชันทั้งหมด 3 เครื่อง ในการทดสอบระบบ ดังนั้นค่า 100 เปอร์เซ็นต์ จึงหมายถึงช่วงที่ระบบมีปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้เท่ากับ 3 เครื่องด้วย



รูปที่ 4.12 กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละเครื่องในสภาพแวดล้อมที่ภาระงานมีลักษณะค่อยๆ เพิ่มขึ้น



รูปที่ 4.13 แสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละหลายเครื่องโดยการคาดการณ์ล่วงหน้าในสภาพแวดล้อมที่ภาระงานค่อยๆ เพิ่มขึ้น



รูปที่ 4.14 กราฟแสดงการเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีลักษณะค่อยๆ เพิ่มขึ้น

4.3.4 วิเคราะห์ผลการทดลอง

จากผลการทดลอง รูปที่ 4.7 และ 4.8 สำหรับระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต (No Scale) โดยการอุทิศทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักจำนวนหนึ่งเครื่องให้แก่ระบบนั้น จะสามารถรองรับปริมาณภาระงานในช่วงปกติคือในช่วง 10 นาทีแรก ที่ภาระงานมีอัตราการเข้าใช้งานเท่ากับ 6 และ 7 ยูสเซอร์เซสชันต่อวินาทีได้เท่านั้น อย่างไรก็ตามสำหรับในช่วงที่อัตราการเข้าใช้งานของผู้ใช้งานในระบบมีค่าเพิ่มสูงขึ้นตั้งแต่ 8 ยูสเซอร์เซสชันต่อวินาทีเป็นต้นไป จากผลการทดลอง รูปที่ 4.8 ค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานจะมีค่าเพิ่มสูงขึ้นอย่างรวดเร็วซึ่งมีค่าต่างจากค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานในช่วงก่อนหน้านี้อย่างชัดเจน สอดคล้องกับค่าของปริมาณงานโดยเฉลี่ยในรูปที่ 4.7 ที่เริ่มจะมีค่าคงที่ ทำให้สามารถสรุปได้ว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิตจะสามารถรองรับปริมาณภาระงานได้จำกัด เฉพาะในช่วงปกติที่ระบบมีปริมาณภาระงานในปริมาณที่ไม่มากเท่านั้น อย่างไรก็ตามจากผลการทดลองเมื่อปริมาณภาระงานของระบบมีปริมาณเพิ่มมากขึ้นเกินกว่าทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักของระบบที่จะสามารถรองรับได้ ย่อมส่งผลให้ประสิทธิภาพของระบบดังกล่าวมีค่าลดลงด้วย ในทางตรงกันข้ามสำหรับระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ใน

ลักษณะทีละเครื่อง (Scale Each) และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องนั้น (Scale Predict) ได้ใช้ทรัพยากรของเครื่องเวิร์กสเตชันที่ไม่ได้ถูกใช้งานอย่างเต็มที่มาจำลองเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือน เพื่อใช้เป็นทรัพยากรเสริมสำหรับช่วยรองรับปริมาณภาระงานของระบบเว็บแอปพลิเคชัน ในช่วงที่มีปริมาณมากดังกล่าว จึงทำให้ค่าของปริมาณงานโดยเฉลี่ยและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานมีค่าที่ดีกว่าในระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต

จากผลการทดลองในรูปที่ 4.8 และ 4.9 พบว่า ระบบเว็บแอปพลิเคชันเริ่มจะไม่สามารถรองรับปริมาณภาระงานได้ในช่วงนาทีที่ 10-15, 50-55 และช่วงนาทีที่ 90-95 ซึ่งเป็นช่วงที่ภาระงานมีอัตราการเข้าใช้งานเท่ากับ 8, 16 และ 24 ยูสเซอร์เซสชันต่อวินาทีตามลำดับ สำหรับในช่วงดังกล่าวระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง ได้ทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเข้าสู่ระบบอีกจำนวน 1 เครื่องในช่วงที่มีอัตราการเข้าใช้งานเท่ากับ 8, 16 และ 24 ยูสเซอร์เซสชันต่อวินาที ตามลำดับ สำหรับในส่วนของระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องนั้น เนื่องมาจากในสภาพแวดล้อมที่ปริมาณภาระงานของระบบมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก ส่งผลให้ค่าแนวโน้มของปริมาณภาระงานที่ได้จากการคาดการณ์ล่วงหน้าของวิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องมีอัตราการเพิ่มขึ้นในปริมาณน้อยตามไปด้วย เมื่อค่าแนวโน้มของปริมาณภาระงานที่ได้จากการคาดการณ์ล่วงหน้ามีปริมาณที่น้อย ส่งผลทำให้ระบบต้องการปริมาณทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนที่จะใช้รองรับปริมาณภาระงานดังกล่าวในปริมาณที่น้อยตาม ซึ่งจากผลการทดลองพบว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องได้ทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเข้าสู่ในระบบอีกจำนวน 1 เครื่องในช่วงที่ระบบมีอัตราการเข้าใช้งานเท่ากับ 8, 16 และ 24 ยูสเซอร์เซสชันต่อวินาทีตามลำดับ ในลักษณะเช่นเดียวกันกับระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง จากลักษณะดังกล่าวจึงส่งผลให้ประสิทธิภาพของระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องได้ให้ประสิทธิภาพที่ใกล้เคียงกัน

และจากผลการทดลองในรูปที่ 4.8 และ 4.9 สำหรับในช่วงนาทีที่ 10 ถึง 15 ซึ่งภาระงานมีอัตราการเข้าใช้งานเท่ากับ 8 ยูสเซอร์เซสชันต่อวินาที ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องได้ตรวจพบว่าทรัพยากรในระบบดังกล่าวเริ่มจะไม่สามารถรองรับปริมาณภาระงานได้แล้ว จึงทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมให้แก่ระบบอีกหนึ่งเครื่อง และในช่วงเวลาต่อมาในนาทีที่ 15 ถึง 20 เมื่อผู้ใช้งานมีอัตราการเข้าใช้งานเพิ่มขึ้นเป็น 9 ยูสเซอร์เซสชันต่อวินาที ปริมาณภาระงานที่เข้าสู่ระบบก็ได้ถูกกระจายแบ่งให้กับแต่ละแอปพลิเคชันเซิร์ฟเวอร์ ซึ่งจากการทดลองพบว่าในขณะนั้นแต่ละแอปพลิเคชันเซิร์ฟเวอร์มีปริมาณแอ็กทีพคอนเน็กชันโดยเฉลี่ยประมาณ 42% ของปริมาณแอ็กทีพคอนเน็กชันมากที่สุด ที่แต่ละแอปพลิเคชันเซิร์ฟเวอร์เริ่มจะไม่สามารถรองรับได้ ซึ่งถือว่ามีปริมาณที่ไม่มาก และเพื่อให้เห็นความแตกต่างของขั้นตอนวิธีการลดขนาดของระบบเว็บแอปพลิเคชัน ในการทดลองนี้ได้กำหนดให้ค่าขอบเขตล่างสุดสำหรับการปรับขนาดของระบบเว็บแอปพลิเคชัน (Lower-threshold) มีค่าเท่ากับ 45% ทำให้ในช่วงที่ภาระงานมีอัตราการเข้าใช้งานเท่ากับ 9 ยูสเซอร์เซสชันต่อวินาที ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง ได้ทำการลดขนาดของแอปพลิเคชันซิร์ฟเวอร์ลง เนื่องจากในขณะนั้นปริมาณแอ็กทีพคอนเน็กชันของแต่ละแอปพลิเคชันเซิร์ฟเวอร์มีปริมาณน้อยกว่าค่าขอบเขตล่างสุดสำหรับการปรับขนาดของระบบเว็บแอปพลิเคชัน และในเวลาต่อมาในช่วงนาทีที่ 20 ถึง 25 เมื่อภาระงานมีอัตราการเข้าใช้งานเพิ่มสูงขึ้นเท่ากับ 10 ยูสเซอร์เซสชันต่อวินาที ระบบที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง ก็ตรวจพบได้ว่าจำเป็นต้องเพิ่มทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์เสริมดังกล่าวให้แก่ระบบอีก แต่ในทางตรงกันข้ามสำหรับระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มปริมาณภาระงานล่วงหน้านั้น ในช่วงนาทีที่ 15 ถึง 20 ภาระงานมีอัตราการเข้าใช้งานเพิ่มสูงขึ้นเท่ากับ 9 ยูสเซอร์เซสชันต่อวินาที ระบบจะยังไม่ทำการลดขนาดของแอปพลิเคชันเซิร์ฟเวอร์ลงทันที เนื่องจากได้คาดการณ์แนวโน้มของปริมาณภาระงานล่วงหน้าพบว่ามีแนวโน้มที่เพิ่มขึ้นและอาจต้องใช้ทรัพยากรดังกล่าวมาเป็นทรัพยากรเสริมอีก เพราะฉะนั้นจึงทำให้ในช่วงเวลาดังกล่าวที่อัตราการใช้งานมีค่าเท่ากับ 9 และ 10 ยูสเซอร์เซสชันต่อวินาที ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการ์แนวโน้มภาระงานล่วงหน้าสามารถให้ประสิทธิภาพที่ดีกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง ซึ่งไม่ได้มีการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า

สำหรับผลการทดลองในรูปที่ 4.11 และ 4.14 พบว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องและระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มปริมาณภาระงานก่อนล่วงหน้านั้น ได้ให้ค่าปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานสามารถใช้บริการได้ในแต่ละช่วงเวลา รวมถึงลักษณะการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันในปริมาณที่ใกล้เคียงกัน เนื่องจากปริมาณเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมที่ได้เพิ่มเข้าสู่ในระบบในแต่ละช่วงเวลานั้นมีปริมาณที่เท่ากัน และจากผลการทดลองในรูปที่ 4.14 พบว่าในช่วงนาทีที่ 15 ถึง 25 ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องซึ่งไม่ได้คาดการณ์ลักษณะของภาระงานล่วงหน้า ได้เริ่มทำการลดขนาดของระบบเว็บแอปพลิเคชันลงโดยการเปลี่ยนสถานะของแอปพลิเคชันเซิร์ฟเวอร์เสริมให้มีสถานะเป็น Standby ตามที่ได้กล่าวมาแล้ว จึงส่งผลให้ทรัพยากรในระบบเว็บแอปพลิเคชันที่สามารถให้บริการได้มีปริมาณลดลง จึงทำให้ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องให้ค่าการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่สูงกว่าวิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ที่มีการคาดการณ์แนวโน้มของภาระงานก่อนล่วงหน้า

4.4 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพสำหรับวิธีการจัดสรรทรัพยากรของระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีแนวโน้มอัตราการเข้าใช้งานในลักษณะที่เพิ่มขึ้นอย่างรวดเร็ว

ในการทดลองนี้มีจุดประสงค์เพื่อต้องการเปรียบเทียบประสิทธิภาพของระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรในลักษณะที่แตกต่างกันคือ ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต (No Scale) ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละเครื่อง (Scale Each) และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า (Scale Predict) ตามลำดับ บนสภาพแวดล้อมที่ภาระงานของระบบเว็บแอปพลิเคชันมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว โดยอ้างอิงลักษณะภาระงานในสภาพแวดล้อมดังกล่าวจากงานวิจัย [19] ตามที่ได้กล่าวไว้แล้วในหัวข้อที่ 2.1.4.2

4.4.1 เครื่องมือที่ใช้ในการทดลอง

1) **เครื่องที่ใช้จำลองภาระงานให้แก่ระบบ** (Client Emulator)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Httperf 0.9.0

2) **เครื่องที่ใช้ในการกระจายภาระงาน** (Load Balancer)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - PHP 5.1.6
  - Mysql-Client 5.0.77

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- UltraMonkey 3
  - IP Virtual Server version 1.2.1
  - heartbeat-2.1.3
  - Linux Director v1.186-ha-2.1.3

3) **เครื่องที่ใช้ในการจัดการระบบเว็บแอปพลิเคชัน** (Manager)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Apache 2.2.3
  - PHP 5.3.5
  - Mysql-Server 5.0.77

4) **เครื่องเวิร์กสเตชันในเวิร์กสเตชันพลู** (Workstation Pool)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - VirtualBox 4.0.0 hyperisor

5) **เครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักของระบบ** (Main Application Server)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - 1 ซีพียูเสมือน (1 Virtual CPU)
  - หน่วยความจำเสมือน 450 เมกะไบต์
  - ต่อประสานแบบบริดจ์ (Bridge network interface)
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Apache 2.2.3
  - Tomcat 5.5.23
  - Java 1.6.0_17

6) **เครื่องเวอร์ชวลแอปพลิเคชันเซิร์ฟเวอร์ที่ใช้จำลองขึ้นในระบบ** (Virtual Application Server)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - 1 ซีพียูเสมือน (1 Virtual CPU)
  - หน่วยความจำเสมือน 450 เมกะไบต์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- ต่อประสานแบบบริดจ์ (Bridge network interface)
- เครือข่ายแลน (LAN bandwidth 100 MB/s)

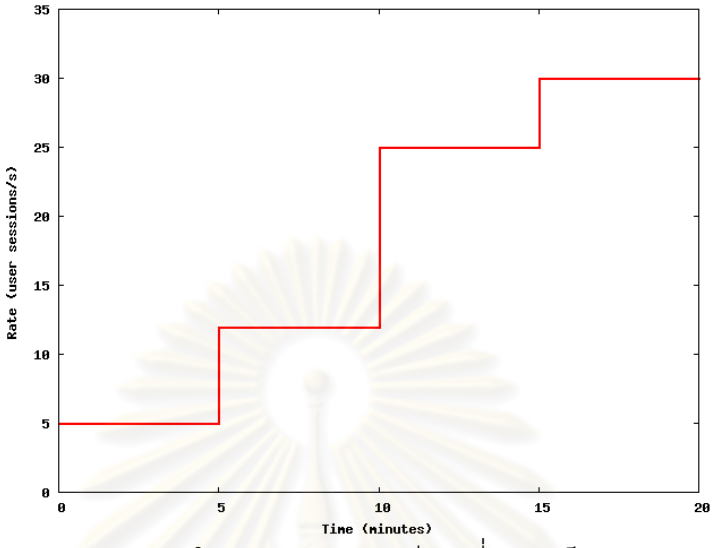
○ **ส่วนชุดคำสั่ง** (Software)

- ระบบปฏิบัติการ CentOS 5.5
- Apache 2.2.3
- Tomcat 5.5.23
- Java 1.6.0_17

4.4.2 สภาพแวดล้อมและวิธีการทดลอง

ในการทดลองนี้เครื่องที่ทำหน้าที่จำลองภาระงานให้แก่ระบบเว็บแอปพลิเคชันจะทำการสร้างภาระงานให้แก่ระบบในลักษณะแสดงดังรูปที่ 4.15 คืออัตราการเข้าใช้งานเฉลี่ยของระบบเว็บแอปพลิเคชันจะเพิ่มสูงขึ้นอย่างรวดเร็วจาก 5 ยูสเซอร์เซสชันต่อวินาที ไปเป็น 12, 25 และ 30 ยูสเซอร์เซสชันต่อวินาที ตามลำดับในทุกๆ 5 นาที ซึ่งจะเห็นได้ว่าเพียงแค่ 20 นาทีแรกเท่านั้น ปริมาณภาระงานโดยเฉลี่ยของระบบดังกล่าวมีแนวโน้มเพิ่มสูงขึ้นเฉลี่ยถึง 6 เท่าตัวของปริมาณภาระงานเริ่มต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.15 แสดงลักษณะภาระงานในช่วงเวลา 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว

สำหรับสภาพแวดล้อมของวิธีการจัดสรรทรัพยากรในระบบเว็บแอปพลิเคชันที่จะใช้ในการเปรียบเทียบประสิทธิภาพได้แก่ ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า ซึ่งรายละเอียดของแต่ละสภาพแวดล้อมได้กล่าวไว้แล้วในหัวข้อที่ 4.3.2.1, 4.3.2.2 และ 4.3.2.3 ตามลำดับ

4.4.3 ผลการทดสอบ

สำหรับผลการทดสอบค่าปริมาณงานโดยเฉลี่ยของระบบและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานของแต่ละระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรที่แตกต่างกันได้แสดงในตารางที่ 4.4 และ 4.5 ตามลำดับ

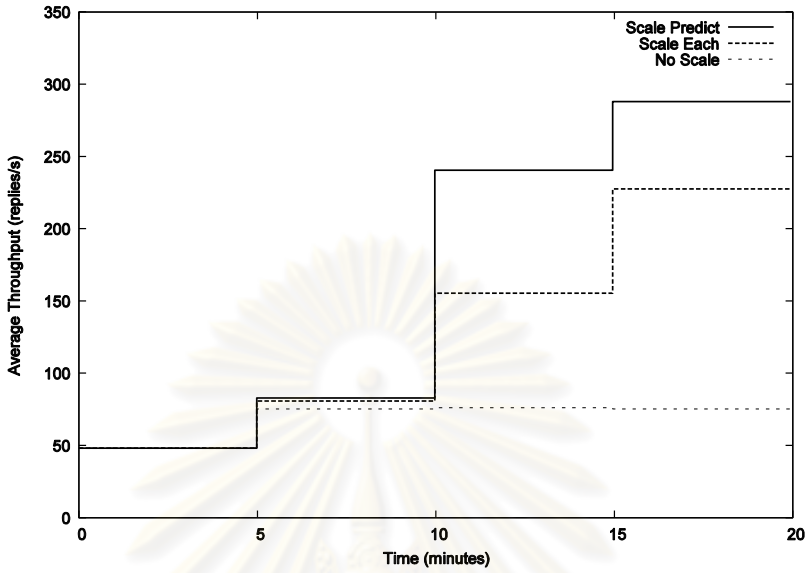
ตารางที่ 4.4 ปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว

Time (minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
0-5	5	48.1	48.1	48.1
5-10	12	82.8	80.8	75.2
10-15	25	240.4	155.4	76.0
15-20	30	287.9	227.5	75.2

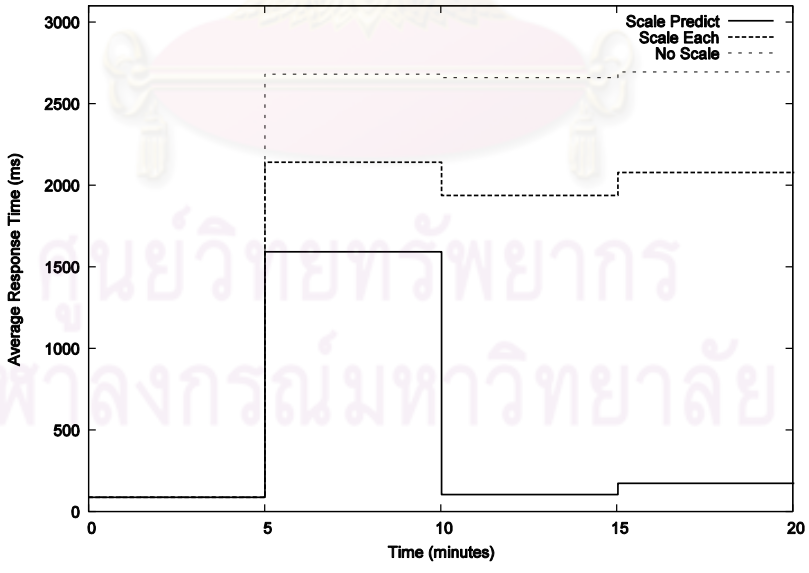
ตารางที่ 4.5 ค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว

Time (minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
0-5	5	86.9	87.0	87.5
5-10	12	1592.0	2140.8	2680.0
10-15	25	104.7	1936.5	2658.6
15-20	30	172.9	2077.1	2694.5

เมื่อนำค่าปริมาณงานโดยเฉลี่ยของระบบและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งาน จากตารางที่ 4.4 และ 4.5 มาวาดกราฟสามารถแสดงได้ดังรูปที่ 4.16 และ 4.17

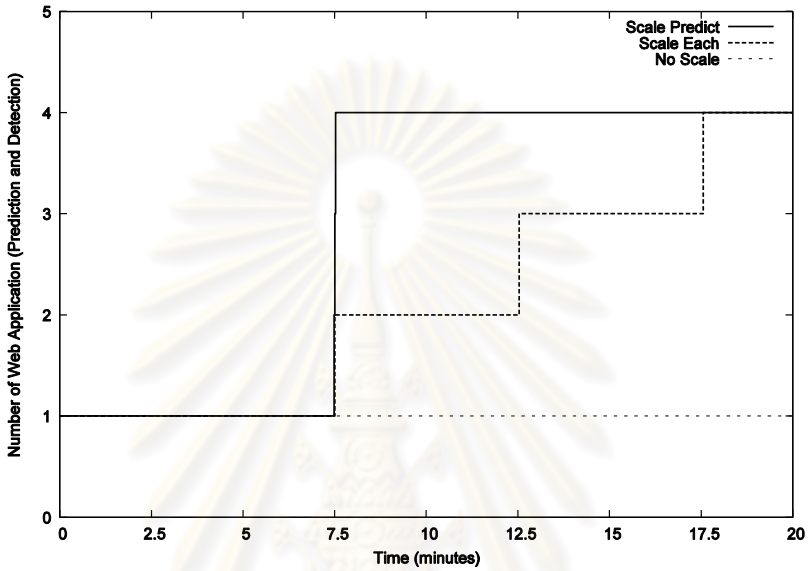


รูปที่ 4.16 กราฟแสดงปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว

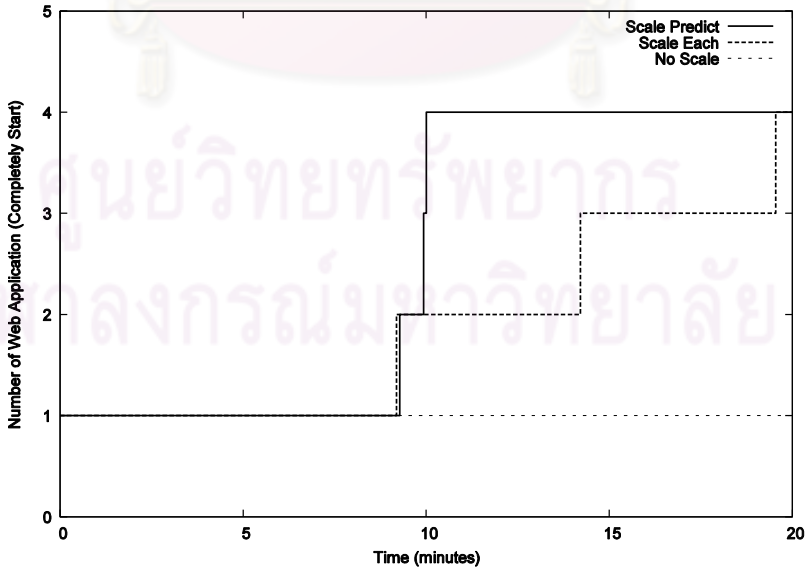


รูปที่ 4.17 กราฟแสดงค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในช่วงเวลา 20 นาทีแรกที่ระบบเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว

สำหรับกราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบของระบบเว็บแอปพลิเคชัน (Prediction and Detection Graph) รวมถึงกราฟแสดงจำนวนเครื่องเว็บแอปพลิเคชันที่สามารถเปิดให้บริการได้อย่างสมบูรณ์ สามารถแสดงได้ดังรูปที่ 4.18 และ 4.19 ตามลำดับ

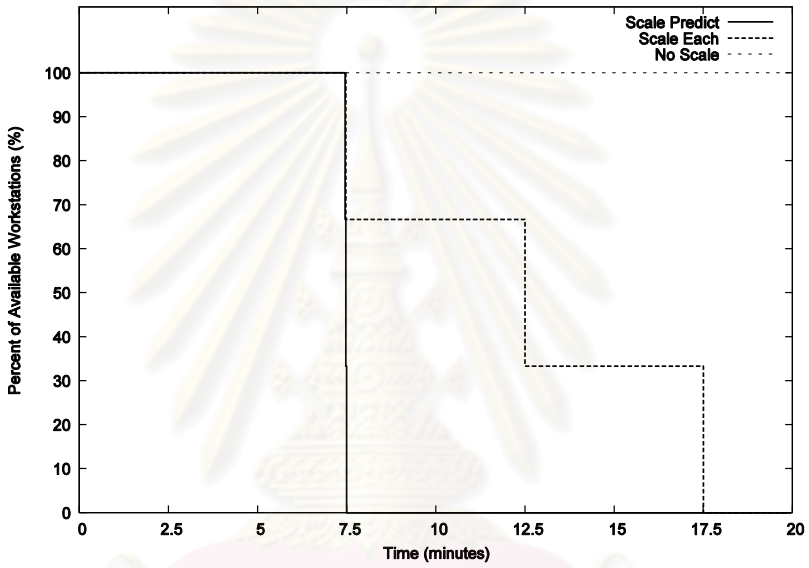


รูปที่ 4.18 แสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบในช่วงเวลา 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว



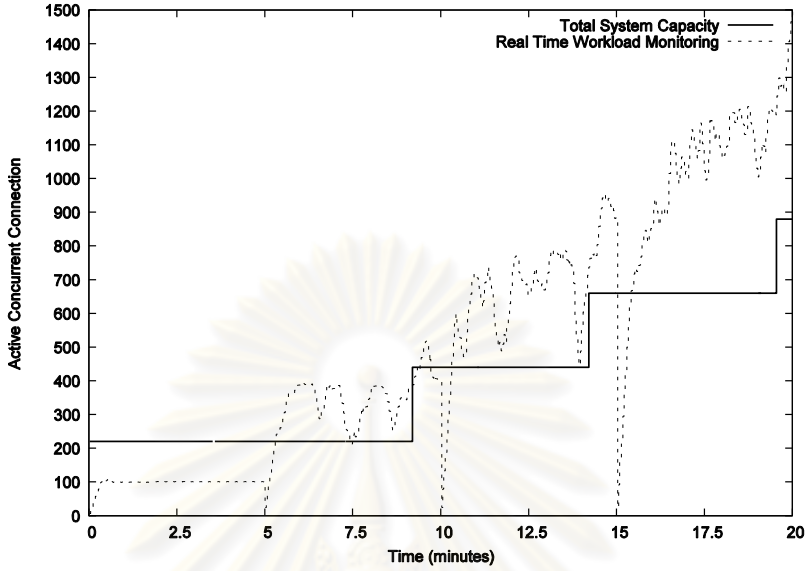
รูปที่ 4.19 กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่สามารถให้บริการได้อย่างสมบูรณ์ในช่วง 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว

สำหรับกราฟแสดงปริมาณเครื่องเวิร์กสเตชันในระบบที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้ และกราฟแสดงการเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชัน (Resources Utilization Graph) ระหว่างวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง และวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์ลักษณะภาระงานก่อนล่วงหน้า สามารถแสดงได้ดังรูปที่ 4.20 ถึง 4.23

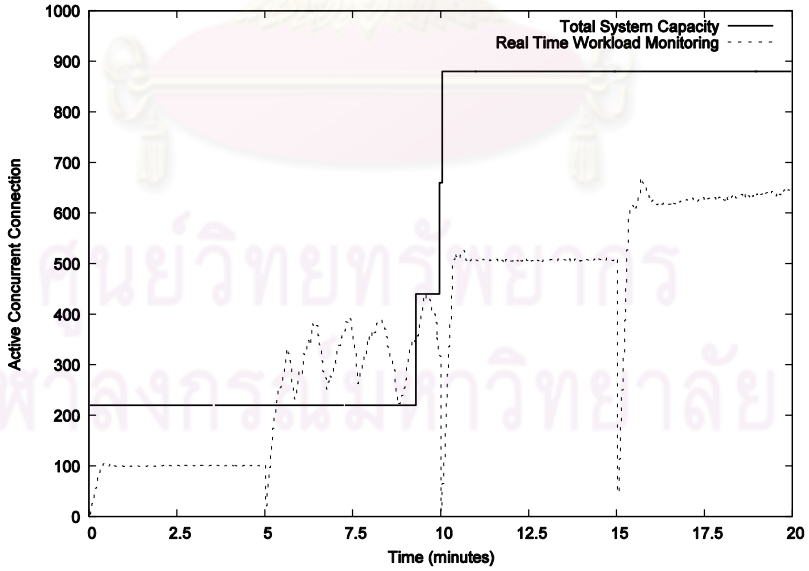


รูปที่ 4.20 กราฟแสดงปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเวิร์กสเตชันสามารถใช้บริการได้ในช่วงเวลา 20 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว

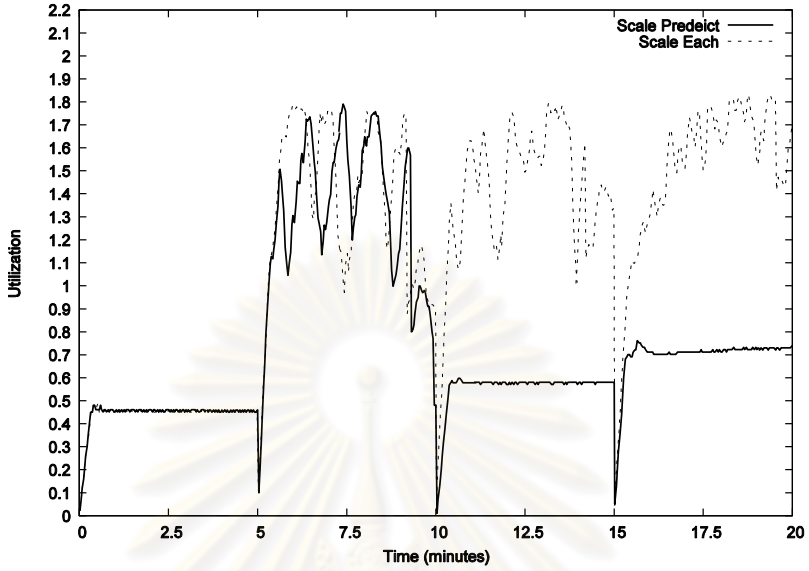
สำหรับรูปที่ 4.20 ค่าในแนวแกนวาย (Y) ใช้แสดงปริมาณเปอร์เซ็นต์ของเครื่องเวิร์กสเตชันในหน่วยงานที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้ และเนื่องจากในการทดลองได้ใช้เครื่องเวิร์กสเตชันทั้งหมด 3 เครื่อง ในการทดสอบระบบ ดังนั้นค่า 100 เปอร์เซ็นต์ จึงหมายถึงช่วงที่ระบบมีปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้เท่ากับ 3 เครื่องด้วย



รูปที่ 4.21 กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละเครื่องในช่วง 20 นาทีแรกที่ระบบเริ่มมีการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว



รูปที่ 4.22 แสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละหลายเครื่องในช่วง 20 นาทีแรกที่ระบบเริ่มมีการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว



รูปที่ 4.23 กราฟแสดงการเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันในช่วง 20 นาทีแรกที่ระบบเริ่มมีการเข้าใช้งานเพิ่มสูงขึ้นอย่างรวดเร็ว

4.4.4 วิเคราะห์ผลการทดลอง

จากผลการทดลอง สำหรับระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต โดยการอุทิศทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักจำนวนหนึ่งเครื่องให้แก่ระบบนั้น จะสามารถรองรับปริมาณภาระงานดังกล่าวในช่วง 5 นาทีแรกที่ภาระงานมีอัตราการเข้าใช้งานเท่ากับ 5 ยูสเซอร์เซสชันต่อวินาทีได้เท่านั้น อย่างไรก็ตามสำหรับในช่วงที่อัตราการเข้าใช้งานของผู้ใช้งานในระบบมีค่าเพิ่มสูงขึ้นเป็น 12, 25 และ 30 ยูสเซอร์เซสชันต่อวินาที จากผลการทดลองค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานจะมีค่าเพิ่มสูงขึ้นอย่างรวดเร็วซึ่งมีค่าต่างจากค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานในช่วงก่อนหน้านี้อย่างชัดเจน สอดคล้องกับค่าของปริมาณงานโดยเฉลี่ยในรูปที่ 4.16 ที่เริ่มจะมีค่าคงที่ ทำให้สามารถสรุปได้ว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิตจะสามารถรองรับปริมาณภาระงานได้จำกัด เฉพาะในช่วง 5 นาทีแรกที่ระบบมีปริมาณภาระงานในปริมาณที่ไม่มากเท่านั้น ในทางตรงกันข้ามสำหรับระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องและทีละหลายเครื่องนั้น ได้ใช้ทรัพยากร

ของเครื่องเวิร์กสเตชันที่ไม่ได้ถูกใช้งานอย่างเต็มที่มาจำลองเป็นเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือน เพื่อใช้เป็นทรัพยากรเสริมสำหรับช่วยรองรับปริมาณภาระงานของระบบในช่วงที่มีปริมาณมากดังกล่าว จึงทำให้ค่าของปริมาณงานโดยเฉลี่ยและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานมีค่าที่ดีกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต

จากผลการทดลองในรูปที่ 4.17 และ 4.18 พบว่าในช่วงนาทีที่ 5 ถึง 10 เป็นช่วงที่ปริมาณภาระงานของระบบมีอัตราการเข้าใช้งานเพิ่มสูงขึ้นเป็น 12 ยูสเซอร์เซสชันต่อวินาที ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องได้ตรวจพบว่า ในขณะนั้นระบบมีปริมาณภาระงานที่เพิ่มสูงขึ้นจำเป็นต้องเพิ่มทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์เสมือนเพื่อช่วยรองรับปริมาณภาระงานเหล่านั้น โดยได้ทำการเพิ่มทรัพยากรเครื่องแอปพลิเคชันเสมือนอีกจำนวนหนึ่งเครื่องให้แก่ระบบเว็บแอปพลิเคชันดังกล่าว ในขณะเดียวกันระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องนั้น จะใช้ข้อมูลจำนวนแอ็กทีพคอนเน็กชันของระบบในขณะนั้นและข้อมูลจากในอดีตมาคำนวณแนวโน้มของปริมาณภาระงานในอีก 5 นาทีข้างหน้า และจากการคาดการณ์แนวโน้มของปริมาณภาระงานดังกล่าว ระบบได้ทำการเพิ่มทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนเพิ่มอีก 3 เครื่องเพื่อใช้รองรับปริมาณภาระงานเผื่อไว้ล่วงหน้า ในเวลาต่อมาเมื่อทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนที่ได้เพิ่มเข้าสู่ระบบสามารถเปิดให้บริการได้อย่างสมบูรณ์รวมถึงปริมาณภาระงานได้กระจายให้กับทุกๆ แอปพลิเคชันเซิร์ฟเวอร์ในระบบอย่างสมดุลแล้ว ปริมาณภาระงานของระบบในขณะนั้นก็มีปริมาณเพิ่มขึ้นเป็น 25 ยูสเซอร์เซสชันต่อวินาที และเมื่อระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องตรวจพบว่าปริมาณภาระงานของระบบในขณะนั้นมีปริมาณที่เพิ่มสูงขึ้น ทรัพยากรในระบบเว็บแอปพลิเคชันที่มีอยู่ในขณะนั้นยังคงไม่เพียงพอที่จะใช้รองรับปริมาณภาระงานดังกล่าวก็จะทำการเพิ่มทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนขึ้นมาใหม่อีก 1 เครื่อง ในทางตรงกันข้ามระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องนั้นได้ทำการเพิ่มทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์ให้กับระบบเผื่อไว้ก่อนล่วงหน้าแล้ว ดังนั้นจึงทำให้ระบบสามารถรองรับปริมาณภาระงานดังกล่าวได้ ส่งผลให้ค่าของปริมาณงานโดยเฉลี่ยและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานมีค่าที่ดีกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง โดยเฉพาะในช่วงที่ปริมาณภาระงานของระบบมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว

อย่างไรก็ตาม จากผลการทดลองในรูปที่ 4.20 พบว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง จะค่อยๆใช้ประโยชน์จากทรัพยากรเครื่องเวิร์กสเตชันในระบบส่งผลให้ปริมาณเครื่องเวิร์กสเตชันที่สามารถให้บริการแก่ผู้ใช้งานได้นั้นมีปริมาณที่มากกว่าเมื่อเทียบกับระบบเว็บแอปพลิเคชันที่ใช้วิธีการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเผื่อไว้ครั้งละหลายเครื่องก่อนล่วงหน้า และจากผลการทดลองในรูปที่ 4.23 พบว่าในช่วง 10 นาทีแรกลักษณะการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันของวิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง และวิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องนั้นให้ค่าที่ใกล้เคียงกัน เนื่องจากในช่วงเวลาดังกล่าวทั้ง 2 ระบบต่างก็มีปริมาณทรัพยากรที่ใกล้เคียงกัน อย่างไรก็ตามสำหรับในช่วงนาทีที่ 10 ถึง 20 พบว่าวิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องจะให้ค่าการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่สูงกว่า เนื่องจากลักษณะที่ค่อยๆเพิ่มของทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมในลักษณะทีละเครื่อง จึงส่งผลให้ทรัพยากรโดยรวมของระบบมีปริมาณที่จำกัด เมื่อเทียบกับการเพิ่มปริมาณเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเผื่อไว้ครั้งละหลายเครื่องก่อนล่วงหน้า จึงทำให้ค่าการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องจึงมีค่าที่สูงกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการณ์ลักษณะของภาระงานก่อนล่วงหน้า

4.5 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพสำหรับวิธีการจัดสรรทรัพยากรของระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน

ในการทดลองนี้มีจุดประสงค์เพื่อต้องการเปรียบเทียบประสิทธิภาพของระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรในลักษณะที่แตกต่างกันคือ ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต (No Scale) ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละเครื่อง (Scale Each) และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า (Scale Predict) ตามลำดับ บนสภาพแวดล้อมที่ภาระงานของระบบเว็บแอปพลิเคชันมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน โดยอ้างอิงลักษณะภาระงานในสภาพแวดล้อมดังกล่าวจากงานวิจัย [19] ตามที่ได้กล่าวไว้แล้วในหัวข้อที่ 2.1.4.2

4.5.1 เครื่องมือที่ใช้ในการทดลอง

1) **เครื่องที่ใช้จำลองภาระงานให้แก่ระบบ** (Client Emulator)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Httperf 0.9.0

2) **เครื่องที่ใช้ในการกระจายภาระงาน** (Load Balancer)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - PHP 5.1.6
  - Mysql-Client 5.0.77

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- UltraMonkey 3
  - IP Virtual Server version 1.2.1
  - heartbeat-2.1.3
  - Linux Director v1.186-ha-2.1.3

3) **เครื่องที่ใช้ในการจัดการระบบเว็บแอปพลิเคชัน** (Manager)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Apache 2.2.3
  - PHP 5.3.5
  - Mysql-Server 5.0.77

4) **เครื่องเวิร์กสเตชันในเวิร์กสเตชันพลู** (Workstation Pool)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - ซีพียู อินเทล (Intel Pentium 4, 3.60GHz)
  - หน่วยความจำหลัก 1 กิกะไบต์
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - VirtualBox 4.0.0 hyperisor

5) **เครื่องแอปพลิเคชันเซิร์ฟเวอร์หลักของระบบ** (Main Application Server)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - 1 ซีพียูเสมือน (1 Virtual CPU)
  - หน่วยความจำเสมือน 450 เมกะไบต์
  - ต่อประสานแบบบริดจ์ (Bridge network interface)
  - เครือข่ายแลน (LAN bandwidth 100 MB/s)
- **ส่วนชุดคำสั่ง** (Software)
  - ระบบปฏิบัติการ CentOS 5.5
  - Apache 2.2.3
  - Tomcat 5.5.23
  - Java 1.6.0_17

6) **เครื่องเวอร์ชวลแอปพลิเคชันเซิร์ฟเวอร์ที่ใช้จำลองขึ้นในระบบ** (Virtual Application Server)

- **ส่วนอุปกรณ์** (Hardware) **และเครือข่าย** (Network)
  - 1 ซีพียูเสมือน (1 Virtual CPU)
  - หน่วยความจำเสมือน 450 เมกะไบต์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- ต่อประสานแบบบริดจ์ (Bridge network interface)
- เครือข่ายแลน (LAN bandwidth 100 MB/s)

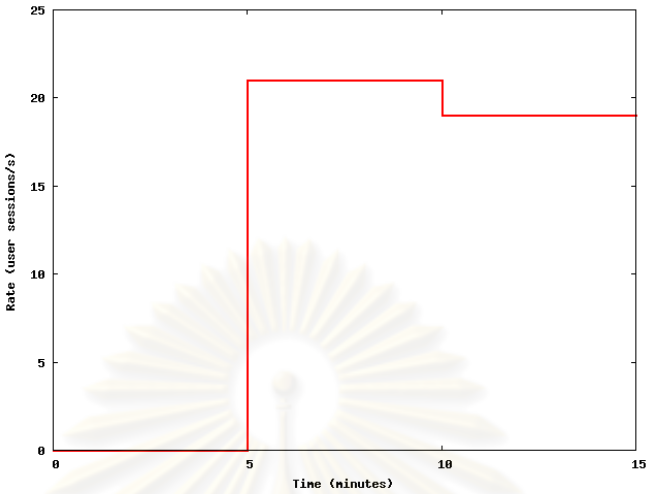
○ **ส่วนชุดคำสั่ง** (Software)

- ระบบปฏิบัติการ CentOS 5.5
- Apache 2.2.3
- Tomcat 5.5.23
- Java 1.6.0_17

4.5.2 **สภาพแวดล้อมและวิธีการทดลอง**

ในการทดลองนี้เครื่องที่ทำหน้าที่จำลองภาระงานให้แก่ระบบเว็บแอปพลิเคชันจะทำการสร้างภาระงานให้แก่ระบบในลักษณะแสดงดังรูปที่ 4.24 คือในช่วง 5 นาทีแรกจะยังไม่มีผู้ใช้งานในระบบเว็บแอปพลิเคชัน แต่หลังจากช่วง 5 นาทีแรกผ่านไป อัตราการเข้าใช้งานเฉลี่ยของระบบเว็บแอปพลิเคชันกลับเพิ่มสูงขึ้นอย่างกะทันหันเป็น 21 และ 19 ยูสเซอร์เซสชันต่อวินาทีตามลำดับ จากลักษณะดังกล่าวสามารถใช้เป็นตัวแทนของสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน เพื่อใช้สำหรับทดสอบประสิทธิภาพของระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรในลักษณะที่แตกต่างกันได้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.24 แสดงภาระงานในช่วงเวลา 15 นาทีแรกที่ระบบเว็บแอปพลิเคชันเริ่มมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

สำหรับสภาพแวดล้อมของวิธีการจัดสรรทรัพยากรในระบบเว็บแอปพลิเคชันที่จะใช้ในการเปรียบเทียบประสิทธิภาพได้แก่ ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า ซึ่งรายละเอียดของแต่ละสภาพแวดล้อมได้กล่าวไว้แล้วในหัวข้อที่ 4.3.2.1, 4.3.2.2 และ 4.3.2.3 ตามลำดับ

4.5.3 ผลการทดสอบ

สำหรับผลการทดสอบค่าปริมาณงานโดยเฉลี่ยของระบบและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานของแต่ละระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรที่แตกต่างกันได้แสดงในตารางที่ 4.6 และ 4.7 ตามลำดับ

ตารางที่ 4.6 ปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

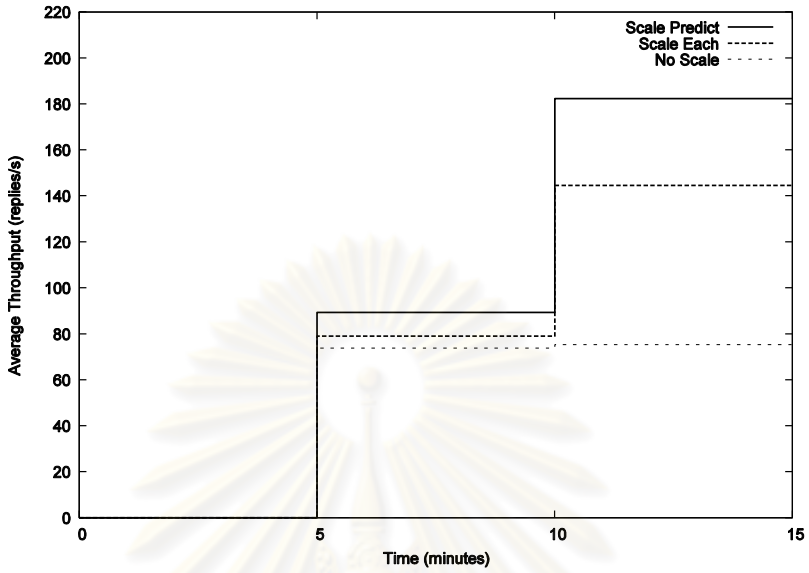
Time (minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
0-5	0	0.0	0.0	0.0
5-10	21	89.3	79.0	73.8
10-15	19	182.3	144.5	75.3

ตารางที่ 4.7 ค่าเฉลี่ยระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

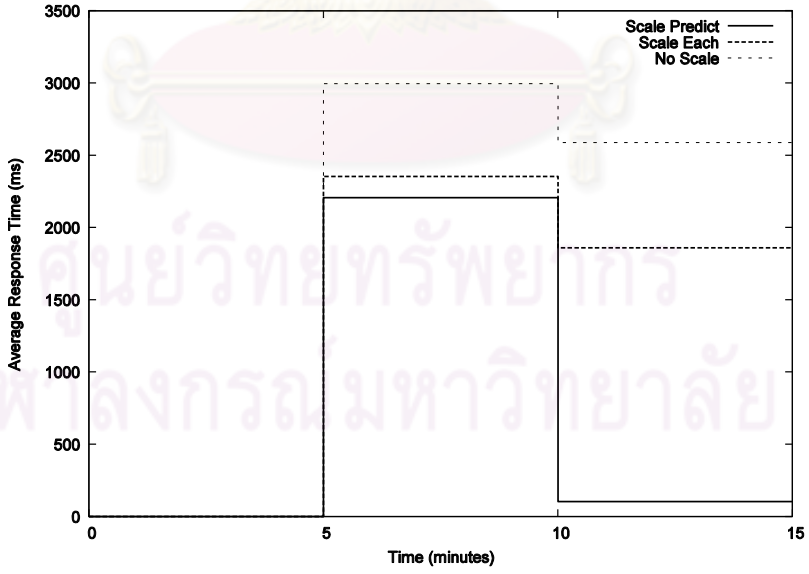
Time (minutes)	Rate (user sessions/s)	Scale Predict	Scale Each	No Scale
0-5	0	0.0	0.0	0.0
5-10	21	2207.3	2353.0	2996.0
10-15	19	104.2	1859.6	2589.5

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

เมื่อนำค่าปริมาณงานโดยเฉลี่ยของระบบและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งาน จากตารางที่ 4.6 และ 4.7 มาวาดกราฟสามารถแสดงได้ดังรูปที่ 4.25 และ 4.26

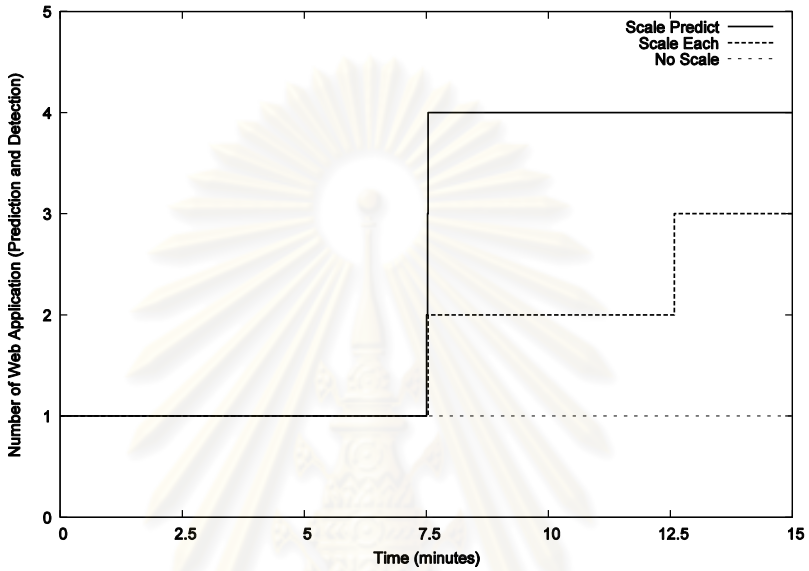


รูปที่ 4.25 กราฟแสดงปริมาณงานโดยเฉลี่ยของแต่ละระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

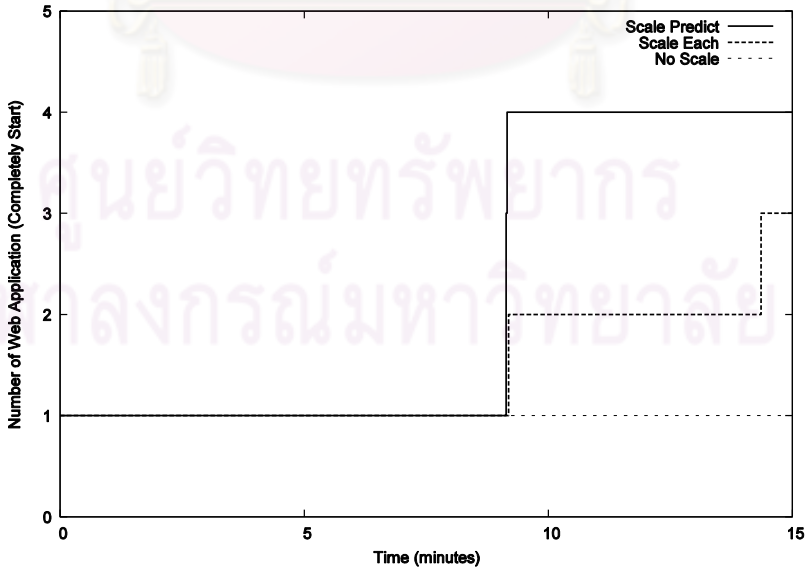


รูปที่ 4.26 กราฟแสดงค่าเฉลี่ยของระยะเวลาที่แต่ละระบบเว็บแอปพลิเคชันใช้ตอบสนองต่อผู้ใช้งานในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

สำหรับกราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบของระบบเว็บแอปพลิเคชัน (Prediction and Detection Graph) รวมถึงกราฟแสดงจำนวนเครื่องเว็บแอปพลิเคชันที่สามารถเปิดให้บริการได้อย่างสมบูรณ์ สามารถแสดงได้ดังรูปที่ 4.27 และ 4.28 ตามลำดับ

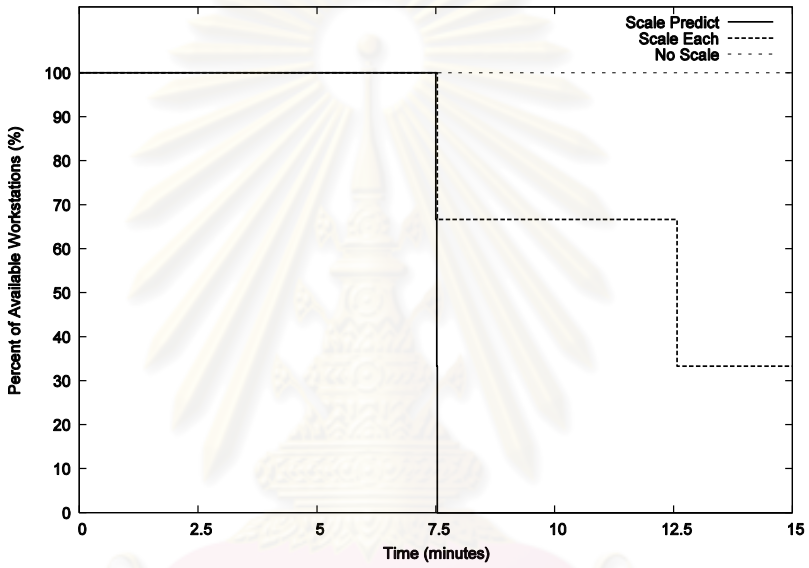


รูปที่ 4.27 กราฟแสดงจำนวนเครื่องแอปพลิเคชันเซิร์ฟเวอร์จากการทำนายและการตรวจพบในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน



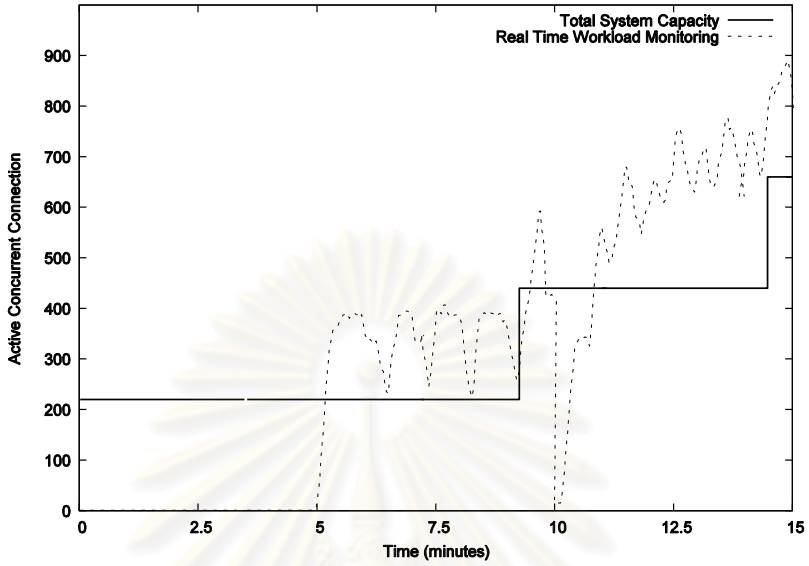
รูปที่ 4.28 กราฟแสดงจำนวนเครื่องแอพลิเคชันเซิร์ฟเวอร์ที่สามารถให้บริการได้อย่างสมบูรณ์ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

สำหรับกราฟแสดงปริมาณเครื่องเวิร์กสเตชันในระบบที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้ และกราฟแสดงการเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชัน (Resources Utilization Graph) ระหว่างวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละเครื่อง และวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะทีละหลายเครื่องโดยการคาดการณ์ลักษณะภาระงานก่อนล่วงหน้า สามารถแสดงได้ดังรูปที่ 4.29 ถึง 4.32

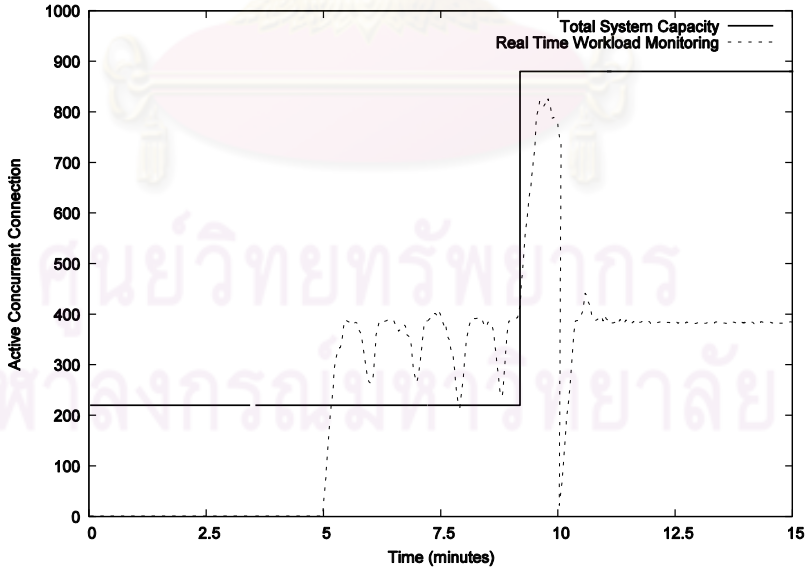


รูปที่ 4.29 กราฟแสดงปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเวิร์กสเตชันสามารถใช้บริการได้ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

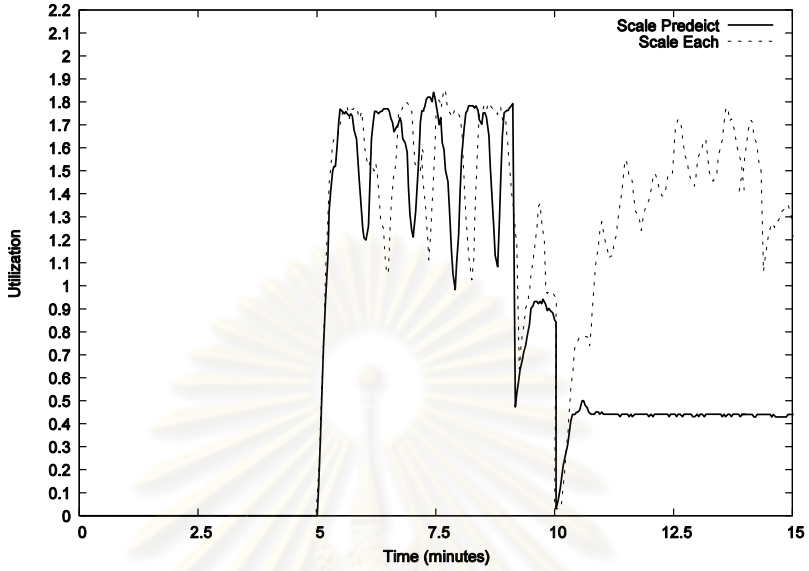
สำหรับรูปที่ 4.29 ค่าในแนวแกนวาย (Y) ใช้แสดงปริมาณเปอร์เซ็นต์ของเครื่องเวิร์กสเตชันในหน่วยงานที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้ และเนื่องจากในการทดลองได้ใช้เครื่องเวิร์กสเตชันทั้งหมด 3 เครื่อง ในการทดสอบระบบ ดังนั้นค่า 100 เปอร์เซ็นต์ จึงหมายถึงช่วงที่ระบบมีปริมาณเครื่องเวิร์กสเตชันที่ผู้ใช้งานเครื่องเวิร์กสเตชันสามารถใช้บริการได้เท่ากับ 3 เครื่องด้วย



รูปที่ 4.30 กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละเครื่องในสภาพแวดล้อมที่ภาระงานมีลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน



รูปที่ 4.31 กราฟแสดงการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดในลักษณะทีละหลายเครื่องในสภาพแวดล้อมที่ภาระงานมีลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน



รูปที่ 4.32 กราฟเปรียบเทียบการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันในสภาพแวดล้อมที่ภาระงานมีลักษณะเพิ่มสูงขึ้นอย่างกะทันหัน

4.5.4 วิเคราะห์ผลการทดลอง

จากผลการทดลองในรูปที่ 4.25 และ 4.26 พบว่าช่วงนาทีที่ 5 ถึง 10 ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิต ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง และระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่อง ต่างก็ไม่สามารถรองรับปริมาณภาระงานในช่วงดังกล่าวได้เนื่องมาจากลักษณะภาระงานของระบบมีลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน ซึ่งในช่วงเวลาดังกล่าวระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องก็ได้ทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมให้แก่ระบบอีก 1 เครื่อง ในขณะเดียวกันระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องจะทำการคาดการณ์ปริมาณแนวโน้มของปริมาณภาระงานในอีก 5 นาทีข้างหน้า และเนื่องมาจากในการทดลองนี้ลักษณะของภาระงานมีลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน ส่งผลให้ค่าแนวโน้มของปริมาณภาระงานที่ได้จากการคาดการณ์ล่วงหน้ามีอัตราการเพิ่มขึ้นในปริมาณที่มากตามไปด้วย ซึ่งจากผลการทดลองระบบได้คาดการณ์ว่าควรจะต้องการ

เพิ่มทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมอีกจำนวน 3 เครื่อง ระบบจึงได้ทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเข้าสู่ระบบในจำนวนเท่ากับ 3 เครื่อง และในเวลาต่อมาเมื่อทรัพยากรเครื่องแอปพลิเคชันเสริมที่ได้เพิ่มเข้าสู่ระบบสามารถเปิดให้บริการได้อย่างสมบูรณ์แล้ว ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องก็จะตรวจพบว่า ทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์เสริมที่ได้เพิ่มเข้าสู่ในระบบในช่วงก่อนหน้านี้ยังคงไม่เพียงพอในการใช้รองรับปริมาณภาระงานดังกล่าวได้ จึงทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเข้าสู่ในระบบเพิ่มอีก 1 เครื่อง ในทางตรงกันข้ามระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องนั้น ได้ทำการเพิ่มทรัพยากรเครื่องแอปพลิเคชันให้กับระบบเผื่อไว้ก่อนล่วงหน้าแล้ว ดังนั้นจึงทำให้ระบบสามารถรองรับปริมาณภาระงานดังกล่าวได้โดยไม่จำเป็นต้องเพิ่มทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์เสริมให้กับระบบในช่วงเวลาดังกล่าวอีก จากลักษณะดังกล่าวนี้จึงส่งผลให้ค่าของปริมาณงานโดยเฉลี่ยและค่าเฉลี่ยของระยะเวลาที่ระบบใช้ในการตอบสนองต่อผู้ใช้งานมีค่าที่ดีกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง

อย่างไรก็ตาม จากผลการทดลองในรูปที่ 4.29 พบว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องจะค่อยๆใช้ประโยชน์จากทรัพยากรเครื่องเวิร์กสเตชันในระบบส่งผลให้ปริมาณเครื่องเวิร์กสเตชันที่สามารถให้บริการแก่ผู้ใช้งานได้นั้นมีปริมาณที่มากกว่าเมื่อเทียบกับระบบเว็บแอปพลิเคชันที่ใช้วิธีการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเผื่อไว้ครั้งละหลายเครื่องก่อนล่วงหน้า และจากผลการทดลองในรูปที่ 4.32 พบว่าในช่วง 10 นาทีแรกลักษณะการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันของวิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง และวิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องนั้นให้ค่าที่ใกล้เคียงกัน เนื่องจากในช่วงเวลาดังกล่าวทั้ง 2 ระบบต่างก็มีปริมาณทรัพยากรที่ใกล้เคียงกัน อย่างไรก็ตามสำหรับในช่วงนาทีที่ 10 ถึง 15 พบว่าวิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องจะให้ค่าการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่สูงกว่า เนื่องจากได้เพิ่มทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมให้กับระบบอีก 1 เครื่อง เมื่อเทียบกับระบบเว็บแอปพลิเคชันที่เพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสริมเผื่อไว้ก่อนล่วงหน้าอีก 3 เครื่องจึงทำให้ค่าการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องจึงมีค่าที่สูงกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการณ์ลักษณะของภาระงานก่อนล่วงหน้า

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอวิธีการจัดสรรทรัพยากรสำหรับโปรแกรมประยุกต์ผ่านเว็บประเภทที่มีการใช้งานมากเป็นบางช่วงเวลา โดยอาศัยทรัพยากรจากเครื่องเวิร์กสเตชันที่มีอยู่แล้วในระบบ และไม่ได้ถูกใช้งานอย่างเต็มที่มาใช้เป็นทรัพยากรเสริมให้กับระบบในช่วงที่มีผู้ใช้งานเป็นจำนวนมาก ซึ่งจะทำให้ช่วยลดค่าใช้จ่ายที่จะต้องใช้ในการอุทิศเครื่องเซิร์ฟเวอร์สำรองให้กับระบบ รวมถึงช่วยทำให้เกิดการใช้ประโยชน์ของทรัพยากรที่ไม่ได้ถูกใช้งานอย่างเต็มที่ให้เกิดประโยชน์สูงขึ้นด้วย และได้ทำการพัฒนาขั้นตอนวิธีการขยายขนาดของระบบเว็บแอปพลิเคชัน จากการขยายขนาดของเว็บแอปพลิเคชันในลักษณะทีละเครื่อง [4] ให้สามารถคาดการณ์แนวโน้มของลักษณะภาระงานก่อนล่วงหน้า และสามารถขยายขนาดของเว็บแอปพลิเคชันได้ในลักษณะครั้งละหลายเครื่องตามปริมาณภาระงานที่เพิ่มขึ้น

จากผลการทดลองทั้งหมดสามารถสรุปได้ว่า ระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรแบบสถิตสามารถให้ประสิทธิภาพได้ดี เฉพาะในช่วงที่ปริมาณภาระงานของระบบอยู่ในขอบเขตที่ทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์หลักของระบบสามารถรองรับได้เท่านั้น ในกรณีที่ปริมาณภาระงานมีปริมาณมากขึ้นเกินกว่าทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์หลักของระบบจะสามารถรองรับได้ จำเป็นต้องทำการขยายขนาดของระบบเว็บแอปพลิเคชันเพื่อรองรับปริมาณภาระงานในช่วงที่มีปริมาณมากเหล่านั้น และสำหรับวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในงานวิจัยนี้ได้ทำการเปรียบเทียบให้เห็นว่า ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง กับระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่อง โดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้าได้ให้ประสิทธิภาพที่ใกล้เคียงกัน เนื่องมาจากในสภาพแวดล้อมที่ปริมาณภาระงานของระบบมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก ส่งผลให้ระบบมีความต้องการทรัพยากรแอปพลิเคชันเซิร์ฟเวอร์เสริมเพื่อใช้รองรับปริมาณภาระงานในลักษณะดังกล่าว ในลักษณะค่อยๆ เพิ่มในลักษณะทีละเครื่องเช่นเดียวกัน ดังนั้นการใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องก็มีความสามารถเพียงพอที่จะใช้จัดสรรทรัพยากรให้ได้ตามความต้องการของระบบเว็บแอปพลิเคชัน และเนื่องมาจากในสภาพแวดล้อมที่ปริมาณภาระงานของระบบมีอัตราการเข้าใช้งานในลักษณะค่อยๆ

เพิ่มขึ้นในปริมาณที่ไม่มาก ส่งผลให้ค่าแนวโน้มของปริมาณภาระงานที่ได้จากการคาดการณ์ล่วงหน้าของวิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องมีอัตราการเพิ่มขึ้นในปริมาณน้อยตามไปด้วย จึงทำให้ระบบดังกล่าวทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนในลักษณะทีละเครื่องเช่นเดียวกันกับระบบเว็บแอปพลิเคชันที่ใช้วิธีการจัดสรรทรัพยากรโดยการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง

อย่างไรก็ตาม สำหรับในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในระบบเว็บแอปพลิเคชันในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว พบว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่องโดยการคาดการณ์แนวโน้มของปริมาณภาระงานก่อนล่วงหน้า จะสามารถให้ประสิทธิภาพที่ดีกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง เนื่องจากในสภาพแวดล้อมดังกล่าว ระบบเว็บแอปพลิเคชันจำเป็นต้องใช้ทรัพยากรเสริมจำนวนหลายเครื่องเพื่อรองรับปริมาณภาระงานที่เพิ่มขึ้นอย่างรวดเร็ว สำหรับการทำงานของระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่องนั้น จะต้องเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนจำนวนหลายครั้ง และในแต่ละครั้งจะต้องรอช่วงเวลาให้แต่ละเครื่องแอปพลิเคชันเซิร์ฟเวอร์ที่ได้เพิ่มเข้าสู่ระบบสามารถพร้อมเปิดให้บริการได้อย่างสมบูรณ์ก่อน รวมถึงรอเวลาให้ปริมาณภาระงานได้กระจายให้กับแต่ละแอปพลิเคชันเซิร์ฟเวอร์ให้อยู่ในสภาวะสมดุลก่อนที่จะทำการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในครั้งต่อไป เพื่อป้องกันการปรับขนาดของระบบเว็บแอปพลิเคชันในครั้งต่อไปที่อาจจะเกิดความผิดพลาดได้ แต่ในขณะเดียวกัน ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่อง จะทำการคาดการณ์แนวโน้มของปริมาณภาระงานล่วงหน้าก่อนที่จะทำการขยายขนาดของระบบเว็บแอปพลิเคชัน และเนื่องจากในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะเพิ่มสูงขึ้นอย่างรวดเร็ว ส่งผลให้ปริมาณแนวโน้มของภาระงานที่ได้จากการคาดการณ์ล่วงหน้าที่ได้มีค่าสูงขึ้นด้วย จึงทำให้ระบบจำเป็นต้องเพิ่มทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนในปริมาณหลายเครื่องตามมา และสำหรับขั้นตอนในการเพิ่มทรัพยากรเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนนั้น ระบบจะทำการเพิ่มเครื่องแอปพลิเคชันเซิร์ฟเวอร์เสมือนพร้อมกันแทนที่จะเพิ่มในลักษณะทีละเครื่อง ซึ่งจากการทำงานในลักษณะดังกล่าวนี้ ส่งผลให้ระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละหลายเครื่อง โดยการคาดการณ์แนวโน้มของปริมาณภาระงานล่วงหน้า สามารถให้ประสิทธิภาพที่ดีกว่าระบบเว็บแอปพลิเคชันที่ใช้วิธีการขยายขนาดของแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง โดยเฉพาะในสภาพแวดล้อมที่ภาระ

งานมีอัตราการเข้าใช้งานในระบบเว็บแอปพลิเคชันในลักษณะที่เพิ่มสูงขึ้นอย่างรวดเร็ว อย่างไรก็ตามวิธีการขยายขนาดของระบบเว็บแอปพลิเคชันในลักษณะครั้งละหลายเครื่องโดยการเตรียมทรัพยากรเผื่อไว้ก่อนล่วงหน้านั้นก็จะส่งผลให้ปริมาณเครื่องเวิร์กสเตชันที่จะสามารถให้บริการแก่ผู้ใช้งานในระบบมีปริมาณลดลงอย่างรวดเร็ว รวมถึงปริมาณการใช้ประโยชน์ของทรัพยากรในระบบเว็บแอปพลิเคชันอาจมีค่าที่ลดลงด้วยเมื่อเทียบกับวิธีการขยายขนาดของเครื่องแอปพลิเคชันเซิร์ฟเวอร์ในลักษณะทีละเครื่อง

5.2 **ข้อเสนอแนะ**

ในการนำทรัพยากรคอมพิวเตอร์ที่ไม่ได้ใช้งานอย่างเต็มที่เพื่อนำมาใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันในช่วงที่มีผู้ใช้งานพร้อมกันเป็นจำนวนมากตามที่ได้นำเสนอมานั้น สำหรับในการใช้งานจริงจำเป็นต้องพิจารณาในประเด็นต่อไปนี้ด้วย

- ประเด็นในส่วนของการเสียพื้นที่หน่วยเก็บ (Storage) สำหรับใช้เก็บข้อมูลของระบบเว็บแอปพลิเคชันในแต่ละเครื่องเวิร์กสเตชัน

- ในการใช้งานจริงเครือข่ายของทรัพยากรแอปพลิชันเซิร์ฟเวอร์หลักของระบบ กับเครือข่ายของทรัพยากรคอมพิวเตอร์ที่ไม่ได้ใช้งานอย่างเต็มที่อาจจะอยู่คนละเครือข่ายกันได้ ส่งผลให้ประสิทธิภาพโดยรวมของระบบอาจมีค่าน้อยลงเนื่องจากการมีเวลาแฝง (Latency) ที่มากขึ้น

- ปริมาณของเครื่องเวิร์กสเตชันที่ผู้ใช้งานเวิร์กสเตชันจะสามารถใช้บริการได้นั้นจะมีปริมาณน้อยลง ซึ่งอาจส่งผลให้หน่วยงานที่มีปริมาณของเครื่องเวิร์กสเตชันในปริมาณที่น้อย อาจมีทรัพยากรไม่เพียงพอในการให้บริการแก่ผู้ใช้งานได้

- เนื่องจากในทางปฏิบัติลักษณะภาระงานของระบบเว็บแอปพลิเคชันอาจไม่ได้มีลักษณะเป็นแบบเชิงเส้นตลอดเวลา ดังนั้นในการใช้งานจริงควรพิจารณาลักษณะภาระงานของระบบเว็บแอปพลิเคชันก่อนว่ามีลักษณะเป็นแบบใด เพื่อให้สามารถเลือกใช้วิธีการวิเคราะห์ลักษณะของภาระงานได้เหมาะสมกับในแต่ละระบบเว็บแอปพลิเคชัน ตัวอย่างเช่น การวิเคราะห์ในลักษณะการถดถอยเชิงเส้น (Linear Regression) หรือการ

วิเคราะห์ในลักษณะที่ไม่ใช่การถดถอยเชิงเส้น (Non Linear Regression) เป็นต้น

สำหรับแนวทางการพัฒนาในอนาคตสำหรับในงานวิจัยนี้มีหลายแนวทางที่สามารถทำได้ เช่น

- การพัฒนาในส่วนของขั้นตอนวิธีการปรับขนาดของระบบเว็บแอปพลิเคชันให้สามารถครอบคลุมถึงทรัพยากรในส่วนฐานข้อมูลของระบบเว็บแอปพลิคชัน
- การปรับปรุงขั้นตอนวิธีการจัดสรรทรัพยากรให้สามารถใช้ประโยชน์ของทรัพยากรที่มีอยู่ให้มีประสิทธิภาพสูงสุด โดยเฉพาะอย่างยิ่งในกรณีที่ทรัพยากรเครื่องเวิร์กสเตชันที่จะใช้เป็นทรัพยากรเสริมให้กับระบบเว็บแอปพลิเคชันนั้นมีปริมาณไม่เพียงพอ หรือมีผู้ใช้งานเครื่องเวิร์กสเตชันเป็นจำนวนมาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Simply Hired, Inc.. Web Application Trends [Online]. Available from: http://www.simplyhired.com/a/jobtrends/trend/q-web+application [2011, May 9].

[2] สำนักงานการทะเบียนและประมวลผล จุฬาลงกรณ์มหาวิทยาลัย. ข้อมูลแสดงจำนวนครั้งในการคลิกเข้าใช้งานในหน้าเว็บลงทะเบียนเรียนของจุฬาลงกรณ์มหาวิทยาลัยในปี ค.ศ. 2009. สำนักงานการทะเบียนและประมวลผล จุฬาลงกรณ์มหาวิทยาลัย : สำนักงานการทะเบียนและประมวลผล จุฬาลงกรณ์มหาวิทยาลัย, 2009.

[3] Padala, P., et al.. Adaptive control of virtualized resources in utility computing environments. SIGOPS Oper. Syst. Rev. 2007 : 289-302.

[4] Chieu, T.C., Mohindra, A., Karve, A.A., and Segal, A.. Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment. IEEE International Conference on e-Business Engineering, 2009.

[5] Iqbal, W., Dailey, M., and Carrera, D.. SLA-Driven Adaptive Resource Management for Web Applications on a Heterogeneous Compute Cloud. Proceedings of the 1st International Conference on Cloud Computing, 2009.

[6] Araki, T.. Autonomic WWW server management with distributed resources. Proceedings of the 2nd workshop on Middleware for grid computing, 2004.

[7] Tsang-Long, P., and Jian-Bo, C.. The Scalability of Heterogeneous Dispatcher-Based Web Server Load Balancing Architecture. Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, 2006.

[8] VMware, Inc.. Understanding Full Virtualization, Paravirtualization, and Hardware Assist. VMware, Inc. : VMware, Inc., 2007.

[9] Principato, M.. Virtualization technology and Process Control System upgrades. Cement Industry Technical Conference, 2010.

[10] Microsoft Corporation. Microsoft Virtual Server [Online]. Available from: http://www.microsoft.com/windowsserversystem/virtualserver [2011, April 7].

[11] VMware, Inc.. VMware [Online]. Available from: http://www.vmware.com [2011, April 7].

[12] Oracle Corporation. VirtualBox Hypervisor [Online]. Available from: http://www.virtualbox.org [2011, April 7].

[13] XenSource, Inc.. Xen Hypervisor [Online]. Available from: http://www.xen.org [2011, April 7].

[14] Horman, S.. Ultra monkey [Online]. Available from: http://www.ultramonkey.org [2011, April 7].

[15] Weisberg, S.. Applied linear regression. New York : John Wiley & Sons, Inc., 1980.

[16] Draper, N.R., and Smith, H.. Applied Regression Analysis. New York : Wiley, 1998.

[17] Hewlett-Packard Development Company. Httperf [Online]. Available from: http://www.hpl.hp.com/research/linux/httperf [2011, April 7].

[18] Arlitt, M., and Jin, T.. A workload characterization study of the 1998 World Cup Web site. Network, IEEE 2000 : 30-37.

[19] Jung, J., Krishnamurthy, B., and Rabinovich, M.. Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites. Proceedings of the 11th international conference on World Wide Web, 2002.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายปวิธ ฟุ้งสิริวิโรจน์ เกิดเมื่อวันที่ 2 กรกฎาคม พ.ศ. 2529 ที่อำเภอเบตง จังหวัดยะลา สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2550 (เกียรตินิยมอันดับ 1) และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2552

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย