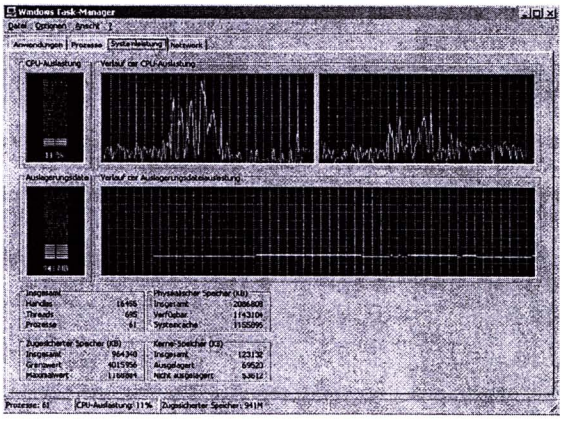
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

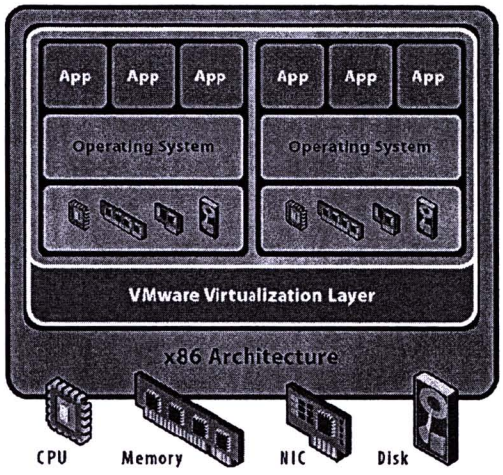
2.1.1 เทคโนโลยีการจำลองระบบเสมือนจริง (Virtualization Technology)

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

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

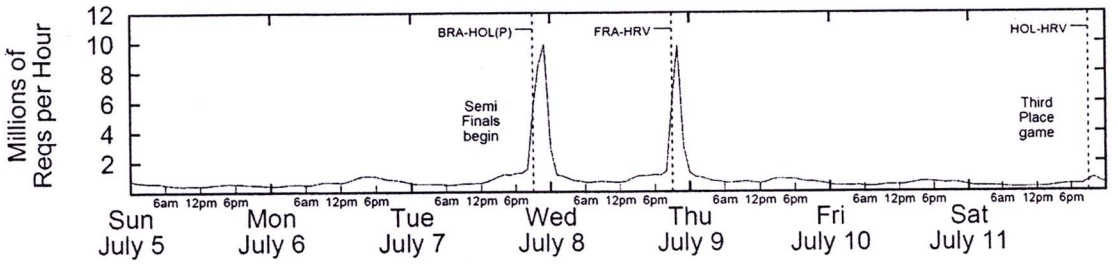
ในงานวิจัยนี้ได้ใช้ตัวอย่างเว็บแอปพลิเคชัน (Simple Web Application) และเครื่องมือที่ใช้ในการจำลองลักษณะของภาระงาน (Workload Generation) โดยอ้างอิงจากงานวิจัย [5] ซึ่งได้ใช้ตัวอย่างเว็บแอปพลิเคชันและเครื่องมือในการจำลองภาระงานดังกล่าวนี้ ทำการทดลองเพื่อประเมินประสิทธิภาพในการขยายขนาดของทรัพยากรเว็บแอปพลิเคชันเซิร์ฟเวอร์ในระบบ 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

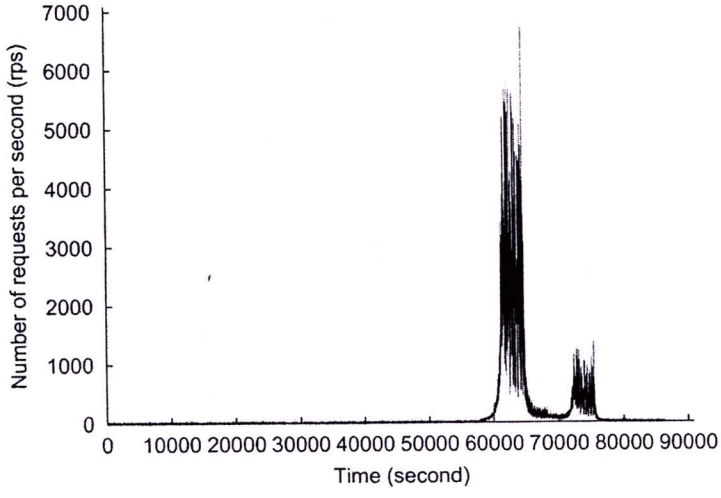
สำหรับลักษณะของภาระงานที่งานวิจัยนี้จะใช้ในการทดสอบระบบเว็บแอปพลิเคชันนั้นได้แบ่งออกเป็น 3 สภาพแวดล้อมหลัก คือ ในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะค่อยๆ เพิ่มขึ้นในปริมาณที่ไม่มาก และในสภาพแวดล้อมที่ภาระงานมีแนวโน้มอัตราการเข้าใช้งานในลักษณะที่เพิ่มขึ้นอย่างรวดเร็ว รวมถึงในสภาพแวดล้อมที่ภาระงานมีอัตราการเข้าใช้งานในลักษณะที่เพิ่มสูงขึ้นอย่างกะทันหัน 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 เท่าตัวเพียงแค่ในช่วงเวลาก่อนจะเริ่มมีการแข่งขันในแต่ละเกมส์เท่านั้น ดังแสดงในรูปที่ 2.4

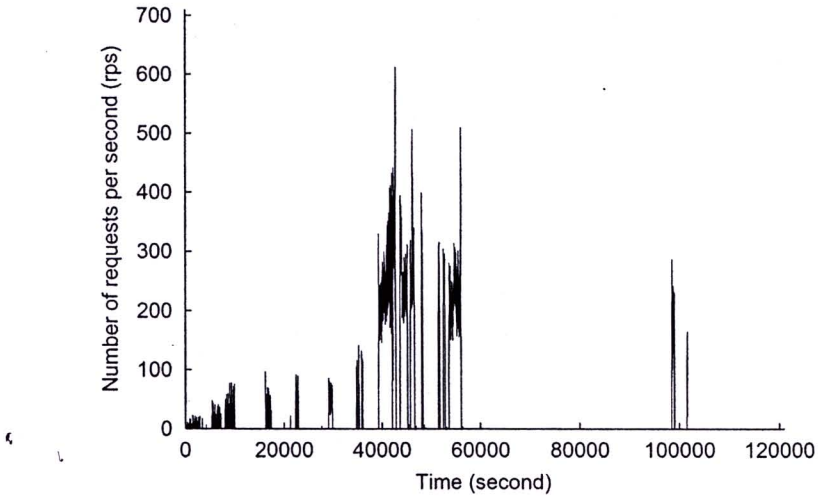


รูปที่ 2.4 แสดงลักษณะภาระงานของระบบเว็บไซต์ www.france98.com ในช่วง 1 สัปดาห์ [18]

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

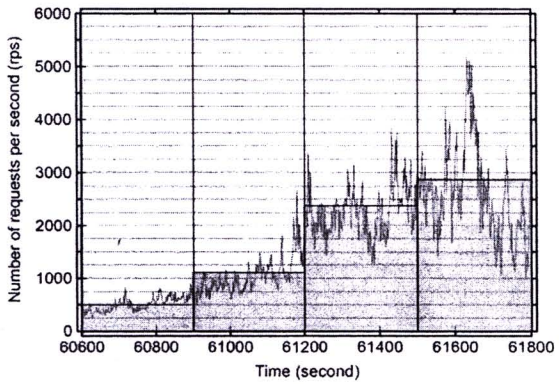


รูปที่ 2.5 แสดงลักษณะปริมาณภาระงานของระบบ Play-along เว็บไซต์ [19]

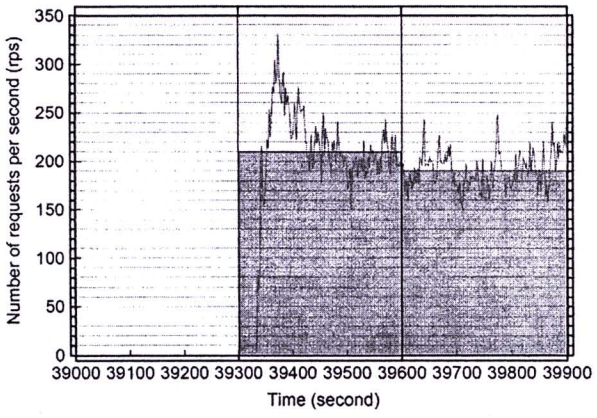


รูปที่ 2.6 แสดงลักษณะปริมาณภาระงานของระบบ Chilean Election Site [19]

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 โดยใช้ค่าเฉลี่ยของระยะเวลาที่ระบบใช้ตอบสนองต่อผู้ใช้งานเป็นตัวกำหนดในการเพิ่มหรือลดจำนวนเว็บเซิร์ฟเวอร์ และทำหน้าที่ควบคุมการกระจายภาระงานของผู้ใช้งานให้ไปประมวลผลบนเครื่องเซิร์ฟเวอร์ที่มีภาระงานที่น้อยและอยู่ใกล้กับผู้ใช้งาน 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