

ตัวแบบแนะนำสำหรับการจัดสรรทรัพยากร บนการทำงานด้วยเทคโนโลยีเสมือน
โดยการใช้ฟิชชีล่อจิก

รัตนเดช ชมภูนุช

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
เมษายน 2558
ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม



ตัวแบบแนะนำสำหรับการจัดสรรทรัพยากร บนการทำงานด้วยเทคโนโลยีเสมือน
โดยการใช้ฟิชชีล่อจิก

รัตน์เดช ชมภูนุช

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

เมษายน 2558

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม





คณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของนายรัตนเดช ชมภูนุช
แล้วเห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบวิทยานิพนธ์

(อาจารย์ ดร. นัตถระกุล สมบัติธีระ)

ประธานกรรมการ

(กรรมการบัณฑิตศึกษาประจำคณะ)

(ผศ.ดร.นัตถระกุล เจริญผล)

กรรมการ

(อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก)

(ผศ.ดร.จิรัฏฐา ญบุญออบ)

กรรมการ

(อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม)

(อาจารย์ ดร.สมนึก พ่วงพรพิทักษ์)

กรรมการ

(กรรมการบัณฑิตศึกษาประจำคณะ)

(ผศ.ดร.วรารัตน์ สงฆ์แป้น)

กรรมการ

(ผู้ทรงคุณวุฒิ)

มหาวิทยาลัยขอนแก่นได้รับวิทยานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

(ผศ.ดร.สุจิน บุตรดีสุวรรณ)

คณบดีคณะวิทยาการสารสนเทศ

(ศ.ดร.ประดิษฐ์ เทอดทูล)

คณบดีบัณฑิตวิทยาลัย

วันที่ 29 เดือน 12.ย. พ.ศ. 2559

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ เป็นผลจากการศึกษาระดับปริญญาโทสาขาวิชาเทคโนโลยีสารสนเทศ โดยได้รับคำชี้แนะในการศึกษาและแนวทางในการดำเนินการวิจัยจากผู้ช่วยศาสตราจารย์ ดร.ฉัตรเกล้า เจริญผล ประธานกรรมการคุมวิทยานิพนธ์

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.จิรัฏฐา ภูบุญอบ และผู้ช่วยศาสตราจารย์สุชาติ คุ่มมณี ที่เป็นผู้ผลักดันให้วิทยานิพนธ์นี้เสร็จสมบูรณ์

ขอขอบพระคุณอาจารย์ ดร.ฉัตรตระกูล สมบัติธีระ ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ วรรัตน์ สงฆ์แป้น และอาจารย์ ดร.สมนึก พ่วงพรพิทักษ์ ที่เป็นผู้ให้คำชี้แนะ เพื่อปรับปรุงข้อบกพร่องของวิทยานิพนธ์ฉบับนี้

ขอบพระคุณบิดาและมารดาที่สั่งสอนและให้กำลังใจเสมอมา

รัตนเดช ชมภูณูช



ชื่อเรื่อง	ตัวแบบแนะนำสำหรับการจัดสรรทรัพยากร บนการทำงานด้วยเทคโนโลยีเสมือน โดยการใช้ฟิซซีลอจิก
ผู้วิจัย	นายรัตนเดช ชมภูนุช
ปริญญา	วิทยาศาสตรมหาบัณฑิต สาขาวิชา เทคโนโลยีสารสนเทศ
กรรมการควบคุม	ผู้ช่วยศาสตราจารย์ ดร.ฉัตรเกล้า เจริญพล ผู้ช่วยศาสตราจารย์ ดร.จิรัฏฐา ภูบุญชอบ
มหาวิทยาลัย	มหาวิทยาลัยมหาสารคาม ปีที่พิมพ์ 2558

บทคัดย่อ

ปัจจุบันคอมพิวเตอร์เสมือน (Virtual Machines) เข้ามามีบทบาทและแทนที่การให้บริการแทนเครื่องคอมพิวเตอร์แม่ข่ายแบบเดิมที่ให้บริการ ด้วยเหตุผลหลักของการที่ใช้ฮาร์ดแวร์ได้อย่างไม่เต็มประสิทธิภาพ วิทยานิพนธ์นี้ใช้เกณฑ์แบบฟิซซี ในการพัฒนาแบบจำลองการตัดสินใจ สำหรับการบริหารจัดการทรัพยากรของเครื่องเสมือน เพื่อหารูปแบบที่เหมาะสมในการจัดสรรทรัพยากรให้เครื่องต่อการทำงาน โดยใช้เกณฑ์แบบฟิซซี ซึ่งมีตัวแปรที่ใช้ในการตัดสินใจคือระยะเวลาในการทำงานอย่างต่อเนื่องของ ซีพียูและหน่วยความจำสำรอง ซึ่งในการตัดสินใจการจัดสรรทรัพยากรของเครื่องเสมือนคือโหมดการเพิ่ม Core ซีพียู และหน่วยความจำสำรอง ตามจำนวนที่เหมาะสมให้แก่เครื่องเสมือนโดยไม่กระทบต่อทรัพยากรในการทำงานหลักของระบบ ผลลัพธ์ที่ได้จากวิธีการนี้จะสามารถบริหารจัดการทรัพยากรของเครื่องเสมือน แตกต่างไปจากการบริหารจัดการทรัพยากรในรูปแบบเดิมๆ โดยใช้ดัชนี 3 ตัว คือ Low, Medium และ High ซึ่งพบว่าแบบจำลองฟิซซีมีการตัดสินใจในการบริหารจัดการทรัพยากรให้เครื่องเสมือนได้ตรงตามความต้องการกับภาระงานที่เป็นอยู่ ณ ขณะนั้น นอกจากนี้จากการวัดประสิทธิภาพด้านเวลา พบว่าแบบจำลองที่นำเสนอมีค่าเวลาในการคำนวณเท่ากับ Big $O(1)$ จึงสามารถนำไปใช้ในระบบที่มีการเปลี่ยนแปลงตลอดเวลาได้ (real time)

คำสำคัญ:ระบบคอมพิวเตอร์เสมือน, ฟิซซี, การบริหารจัดการทรัพยากรของเครื่องเสมือน, ภาระงาน



TITLE A model for allocation resource recommendation Virtualization Technology using Fuzzy logic

AUTHOR Mr. Ruttanadech Chompoonuch

DEGREE Master Degree of Science **MAJOR** Information Technology

ADVISORS Assistant Professor, Ph.D., Chatklaw Jareanpon.
Assistant Professor, Ph.D., Jiratta Phuboon-ob.

UNIVERSITY Mahasarakham University **YEAR** 2015

ABSTRACT

The Virtual Machine has come to service instead of the Physical Server Machine because of the ineffectively used hardware. In this thesis, we proposed to develop the Fuzzy Decision Model for resource management of Virtual Machines to find the allocated-resource Model for operating the machines. The criteria by Fuzzy variables are used to determine the duration of the continued work of CPU and Secondary Memory. The decision output allocates the resources of the Virtual Machine to increase the appropriate CPU core and Secondary Memory without affecting the operation of the System Resources. From this paper, the proposed method is different from the other resource managements. The Virtual Machines is classified into 3 level indexes that are Low, Medium and High. It was found that the proposed model with Fuzzy decision-making in resource management for Virtual machines that is able to use for resource management of the interactive workload. Moreover The Computation time result is equal to Big O (1), this proposed is able to apply in many changeable systems (Real-time)

Key Words: Virtual Machine, Fuzzy, Resource management of virtual machines, Workload



สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญภาพประกอบ	ช
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ความสำคัญของการวิจัย	2
1.4 ขอบเขตของการวิจัย	2
1.5 นิยามศัพท์เฉพาะ	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	6
2.1 Virtualization Technology	6
2.2 Hosted Architecture	6
2.3 Hypervisor Architecture	7
2.4 Virtualization Method	8
2.5 Xen Hypervisor	9
2.6 ตรรกศาสตร์คลุมเครือ	12
2.7 Web Stress Tool	16
2.8 งานวิจัยที่เกี่ยวข้อง	17
บทที่ 3 วิธีดำเนินการวิจัย	20
3.1 ศึกษาโปรแกรมเฝ้าระวังทรัพยากรบนเครื่องจริง บน Xen Hypervisor	21
3.2 วิเคราะห์และพัฒนาตัวแบบด้วย Fuzzy logic ในการบริหารจัดการทรัพยากรของเครื่องเสมือนบน Xen Hypervisors	23
3.3 เปรียบเทียบประสิทธิภาพการแนวทางการบริหารจัดการทรัพยากรเครื่องเสมือนบน Xen แนวทางใหม่และแนวทางเดิม	30
บทที่ 4 ผลการวิจัย	32
4.1 ผลการวิเคราะห์ประสิทธิภาพสภาวะภาระงานของระบบปฏิบัติการเซ็น	32
4.2 เสนอแนวทางในการบริหารจัดการทรัพยากรด้วย Fuzzy Logic ในการพัฒนาตัวแบบ	32
4.3 ผลการเปรียบเทียบประสิทธิภาพการแนวทางการบริหารจัดการทรัพยากรเครื่องเสมือนบน Xen แนวทางใหม่และแนวทางเดิม	41



	หน้า
บทที่ 5 สรุปผล และอภิปรายผล	44
5.1 สรุปผล และอภิปรายผล	44
5.2 ข้อเสนอแนะ	45
เอกสารอ้างอิง	46
ภาคผนวก	50
ภาคผนวก ก การติดตั้งระบบปฏิบัติการ Xen Server	51
ภาคผนวก ข การติดตั้งโปรแกรม ทดสอบระบบ Webserver Stress Tool	61
ประวัติย่อผู้วิจัย	65



สารบัญตาราง

	หน้า
ตาราง 2.1 ข้อมูลส่วนเชื่อมต่อที่สนับสนุน	18
ตาราง 3.1 การจำแนก Workload	25
ตาราง 4.1 ผลการศึกษาโปรแกรมเฝ้าระวังทรัพยากรเครื่องจริงบน Xen Hypervisor	32
ตาราง 4.2 ผลการทดสอบและวัดประสิทธิภาพด้วยโปรแกรม Web Stress Tool	34
ตาราง 4.3 ฐานความรู้ (Knowledge Base)	35
ตาราง 4.4 พจน์ภาษาของ CPU และ Memory	36
ตาราง 4.5 รูปแบบข้อมูลที่จัดเก็บในฐานความรู้	38
ตาราง 4.6 ตัวอย่างการประเมินภาระงานการใช้ทรัพยากรของเครื่องเสมือน	41
ตาราง 4.7 ผลเปรียบเทียบระหว่าง Non-Fuzzy Model, Fuzzy Model	42



สารบัญภาพประกอบ

	หน้า
ภาพประกอบ 1.1 โครงสร้างการทำงานของโมเดลรวบรวมข้อมูล (VM Sensors)	3
ภาพประกอบ 2.1 Hosted Architecture	7
ภาพประกอบ 2.2 Hypervisor Architecture	7
ภาพประกอบ 2.3 เทคนิค Full Virtualization	8
ภาพประกอบ 2.4 เทคนิค Para-Virtualization	9
ภาพประกอบ 2.5 สถาปัตยกรรมการ Xen Hypervisor	10
ภาพประกอบ 2.6 แสดงการทำงานของการทำงาน CPU Virtualization	11
ภาพประกอบ 2.7 การประยุกต์ใช้งานพีซี	13
ภาพประกอบ 2.8 รูปแบบโครงสร้างการทำงาน ของ NEX	17
ภาพประกอบ 3.1 ภาพรวมงานวิจัย	20
ภาพประกอบ 3.2 ขั้นตอนการทดสอบหน่วยความจำ (Memory)	21
ภาพประกอบ 3.3 ขั้นตอนการทดสอบหน่วยประมวลผลกลาง (CPU)	22
ภาพประกอบ 3.4 ขั้นตอนการทดสอบหน่วยความจำหลักสถานการณ์การทดลอง	24
ภาพประกอบ 3.5 ตัวแปรอินพุต	26
ภาพประกอบ 3.6 แสดงการตรวจสอบเงื่อนไขภาระงานต่ำ	27
ภาพประกอบ 3.7 แสดงการตรวจสอบเงื่อนไขภาระงานปานกลาง	28
ภาพประกอบ 3.8 แสดงการตรวจสอบเงื่อนไขภาระงานสูง	28
ภาพประกอบ 3.9 แสดงการตรวจสอบเงื่อนไขในทุกรณี ของ CPU	29
ภาพประกอบ 3.10 แสดงการตรวจสอบเงื่อนไขในทุกรณี ของ Memory	29
ภาพประกอบ 4.1 การออกแบบตัวแบบ Fuzzy Logic ในการบริหารจัดการเครื่องเสมือนใน Xen	33
ภาพประกอบ 4.2 การทดสอบประสิทธิภาพของตัวแบบโดยการจำลองงานให้เครื่องเสมือน	38
ภาพประกอบ 4.3 แสดงขั้นตอนการทำงานในการคำนวณหาค่ากรณีค่าภาระงานต่ำของ CPU และ Memory	39
ภาพประกอบ 4.4 แสดงขั้นตอนการทำงานในการคำนวณหาค่ากรณีค่าภาระงานปานกลางของ CPU และ Memory	39
ภาพประกอบ 4.5 แสดงขั้นตอนการทำงานในการคำนวณหาค่ากรณีค่าภาระงานสูงของ CPU และ Memory	40
ภาพประกอบ 4.6 การทำงานของตัวแบบฟัซซีลอจิก ด้วยการหา Big-O Notation	43
ภาพประกอบ ก.1 เริ่มติดตั้ง XenServer	52
ภาพประกอบ ก.2 เลือกรูปแบบภาษาในการติดตั้ง	53
ภาพประกอบ ก.3 ทำการตรวจสอบความถูกต้องของข้อมูลก่อนติดตั้ง	53
ภาพประกอบ ก.4 เลือกพื้นที่ติดตั้ง	54
ภาพประกอบ ก.5 ยืนยันการติดตั้งดังกล่าว	54



	หน้า
ภาพประกอบ ก.6 เลือกแหล่งข้อมูลที่นำมาติดตั้ง	55
ภาพประกอบ ก.7 ติดตั้งรายละเอียดส่วนเสริม	55
ภาพประกอบ ก.8 ตรวจสอบยืนยันแหล่งข้อมูลที่นำมาติดตั้งว่าถูกต้องหรือไม่	56
ภาพประกอบ ก.9 ป้อนชุดรหัสเพื่อใช้บริหารจัดการระบบ	56
ภาพประกอบ ก.10 เลือกขาอินเตอร์เฟซเน็ตเวิร์กเพื่อใช้บริหารจัดการผ่านเครือข่าย	57
ภาพประกอบ ก.11 กำหนดหมายเลข IP เพื่อใช้บริหารจัดการผ่านเครือข่าย	57
ภาพประกอบ ก.12 กำหนดชื่อเครื่อง และหมายเลข IP ของ DNS	58
ภาพประกอบ ก.13 เลือกประเทศที่ในการติดตั้ง	58
ภาพประกอบ ก.14 ติดตั้งระบบปฏิบัติการ	59
ภาพประกอบ ก.15 แสดงสถานะ ติดตั้งระบบปฏิบัติการ	59
ภาพประกอบ ก.16 เมื่อติดตั้งระบบปฏิบัติการเสร็จจะทำการเริ่มต้นระบบปฏิบัติการใหม่	60
ภาพประกอบ ก.17 เข้าสู่หน้าบริหารจัดการระบบปฏิบัติการ Xen	60
ภาพประกอบ ข.1 เริ่มติดตั้งโปรแกรมทดสอบ Webserver Stress Tool ที่เครื่องลูกข่าย	62
ภาพประกอบ ข.2 ยอมรับ License Agreement	62
ภาพประกอบ ข.3 พื้นที่ติดตั้งโปรแกรม	63
ภาพประกอบ ข.4 สร้าง Shortcut	63
ภาพประกอบ ข.5 การติดตั้งเสร็จสมบูรณ์	64
ภาพประกอบ ข.6 ตัวอย่างหน้าหลักโปรแกรม	64



บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

ปัจจุบันการระบบคอมพิวเตอร์เสมือน (Virtual Machines) [1] เป็นที่นิยมอย่างมากในนำเข้ามาให้บริการแทนระบบคอมพิวเตอร์แบบเดิมด้วยเหตุผลหลักของการที่ใช้ฮาร์ดแวร์ได้อย่างไม่คุ้มค่า ไม่เต็มประสิทธิภาพ จึงมีแนวคิดที่จะสร้างระบบคอมพิวเตอร์เสมือน ทำให้เกิดการใช้งานทรัพยากรได้คุ้มค่าและมีประสิทธิภาพมากยิ่งขึ้น นอกจากนั้นยังช่วยเรื่องการประหยัดพลังงาน ทั้งเพื่อลดค่าใช้จ่าย และดูแลสภาพแวดล้อมให้ยั่งยืน ยังมีส่วนในการผลักดันให้หลายองค์กรหันมาให้ความสนใจกับแนวคิดเทคโนโลยีเสมือน (Virtualization Technology) อย่างจริงจังกันมากขึ้น

ระบบคอมพิวเตอร์เสมือนมีแนวคิดจากการสร้างระบบขึ้นจากคอมพิวเตอร์จริง โดยการใช้ออฟต์แวร์เข้ามาช่วยทำหน้าที่ในการจัดสรรทรัพยากรทางกายภาพต่างๆ เรียก โปรแกรมนั้นว่า ไฮเปอร์ไวเซอร์ (Hypervisor) [2] โดยในเครื่องจริงนั้น สามารถสร้างได้หลายๆ ระบบคอมพิวเตอร์เสมือนเท่าที่ประสิทธิภาพของเครื่องจริงจะรองรับระบบเครื่องเสมือนได้ เนื่องจากการใช้เทคโนโลยีเสมือน จะมีการสร้างเครื่องเสมือนขึ้นมาซึ่งแต่ละเครื่องเสมือนจะมีความแตกต่างของระบบปฏิบัติการจึงจำเป็นต้องใช้ทรัพยากรจากเครื่องจริงมาเพื่อให้บริการเครื่องเสมือนนั้นๆ เช่น หน่วยประมวลผล (CPU), หน่วยความจำ (Memory), ส่วนนำเข้า/ส่งออก (I/O device) เป็นต้น ซึ่งภาระต่างๆ จะตกไปเป็นภาระงาน (Workloads) ของระบบปฏิบัติการเครื่องจริง ที่จำเป็นต้องจัดทรัพยากรภายในเครื่องให้กับเครื่องเสมือนต่างๆ ที่ทำงานอยู่ โดยเรียกระบบปฏิบัติการเหล่านั้นว่า Hypervisor โดยปัจจุบันระบบปฏิบัติการที่ได้รับความนิยม คือ VMware [3] และ Xen [4] ซึ่งมีข้อดีและข้อเสียแตกต่างกันออกไป โดย VMware นั้นเป็นระบบปฏิบัติการแบบปิดต้องซื้อลิขสิทธิ์เพื่อใช้บริการ มีความสะดวกในการบริหารจัดการแต่มีราคาสูงมาก ส่วน Xen เป็นระบบปฏิบัติการแบบเปิด (Open Source) มีประสิทธิภาพในการให้บริการในระดับดีแต่ต้องอาศัยความเชี่ยวชาญของผู้ดูแลระบบในการบริหารจัดการเครื่องมือช่วยบริหารจัดการต่างๆ ยังมีน้อย

ในปัจจุบันได้มีงานวิจัยต่างๆ เช่นงานวิจัยใน [5-7] เป็นการนำเสนอในรูปแบบการบริหารจัดการภาพรวมของการทำงานแบบคลาวด์ ซึ่งอาศัยหลักการบริหารจัดการแบบข้อตกลงระดับการให้บริการ หรือ เอสแอลเอ (Service Level Agreement – SLA) ซึ่ง เอสแอลเอ จะเป็นการให้บริการที่สอดคล้องกับงบประมาณหรือข้อตกลงที่ทำไว้ระหว่างผู้ให้บริการและผู้ให้บริการ ในการจัดสรรทรัพยากร โดยไม่คำนึงถึงความต้องการทรัพยากรที่แท้จริงของระบบ ณ เวลานั้นๆ

ระบบปฏิบัติการ Xen ได้รับความนิยมในการใช้เป็น ไฮเปอร์ไวเซอร์นั้น แต่ยังมีขาดเครื่องมือในการเฝ้าระวัง การบริหารจัดการทรัพยากรและวิเคราะห์ระบบผู้วิจัยจึงได้มีแนวคิดการแก้ไขปัญหของระบบภายใต้แนวคิดการเฝ้าระวังและการบริหารจัดการทรัพยากรระบบแบบ อัตโนมัติสำหรับระบบปฏิบัติการ Xen โดยที่จะสามารถทำงานได้โดยอัตโนมัติปรับให้เข้ากับการเปลี่ยนแปลงเสมอของระบบ โดยระบบปฏิบัติการเครื่องเสมือนสามารถเรียนรู้ลักษณะการทำงานในการบริหารจัดการทรัพยากรและสามารถเฝ้าระวังทรัพยากรได้อย่างมีประสิทธิภาพอย่างต่อเนื่องโดยในงานวิจัยนี้จะนำ



หลักการ ตรรกะคลุมเครือ (Fuzzy Logic) มาเป็นต้นแบบในการศึกษาใช้ทำนายถึงสภาวะภาระงานต่างๆเพื่อใช้บริหารจัดการ การใช้ทรัพยากรของเครื่องจริงต่อเครื่องเสมือนในระบบปฏิบัติการเห็นได้อย่างมีประสิทธิภาพพร้อมทั้งทำการตรวจสอบความผิดพลาดที่จะเกิดขึ้นกับอุปกรณ์ต่าง ๆ ของคอมพิวเตอร์จริงในอนาคตได้

1.2 วัตถุประสงค์ของการวิจัย

12.1 เพื่อวิเคราะห์และพัฒนาต้นแบบในการใช้ทำนายถึงสภาวะภาระงานต่างๆ ของการใช้ทรัพยากรของระบบปฏิบัติการเซิน

1.2.2 เพื่อเสนอแนวทางในการบริหารจัดการทรัพยากรแบบอัตโนมัติโดยประยุกต์เข้ากับตรรกะคลุมเครือ (Fuzzy Logic) ในระบบปฏิบัติการเสมือน

1.2.3 เพื่อทดสอบประสิทธิภาพของต้นแบบที่พัฒนา

1.3 ความสำคัญของการวิจัย

เทคโนโลยี Cloud กำลังได้รับความนิยมอย่างสูงในปัจจุบัน กลไกสำคัญที่ทำให้เทคโนโลยีดังกล่าวประสบความสำเร็จอยู่เบื้องหลังคือ เวอร์ช่วลไลเซชันแต่การบริหารจัดการ เวอร์ช่วลไลเซชันจะบริหารด้วยการทำงานแบบ ปรับแต่งด้วยผู้ดูแลระบบซึ่งประสิทธิภาพการทำงานขึ้นอยู่กับผู้ดูแลระบบซึ่งมีความชำนาญในการบริหารจัดการทรัพยากรที่แตกต่างกัน งานวิจัยนี้เล็งเห็นว่า การ จัดสรรทรัพยากรควรจะทำด้วยวิธีการที่เหมาะสมและขึ้นต่อสภาพการทำงานของระบบที่เกิดขึ้นจริงในช่วงเวลานั้นๆ แต่เนื่องจากความคลุมเครือ (เช่น ความสัมพันธ์ระหว่างหน่วยประมวลผลกลาง หน่วยความจำไม่แสดงความสัมพันธ์กันอย่างเด่นชัด) ในการบริหารทรัพยากรจะเป็นส่วนที่ยากต่อการคำนวณด้วยการประมาณค่าจากผู้ดูแลระบบ ดังนั้นเพื่อแก้ปัญหาการจัดสรรทรัพยากรด้วยความเป็นธรรมและเหมาะสมภายใต้สภาวะของความคลุมเครือ งานวิจัยนี้จึงเลือกเทคนิคแบบ ฟัชซีเข้ามาช่วยในการจัดสรรทรัพยากรในการทำงานในรูปแบบ Dynamic Allocation ที่ให้บริการให้บริการใน Private Cloud Computing

1.3.1 เพิ่มประสิทธิภาพในเครื่องเสมือน ที่ทำงานบนระบบปฏิบัติการ Xen Hypervisor

1.3.2 สามารถวางแผนเพื่อรับมือกับปัญหาที่เกิดขึ้นในและส่งผลกระทบต่อการทำงานของเครื่องเสมือนที่ทำงานบน ระบบปฏิบัติการ Xen Hypervisor

1.4 ขอบเขตของการวิจัย

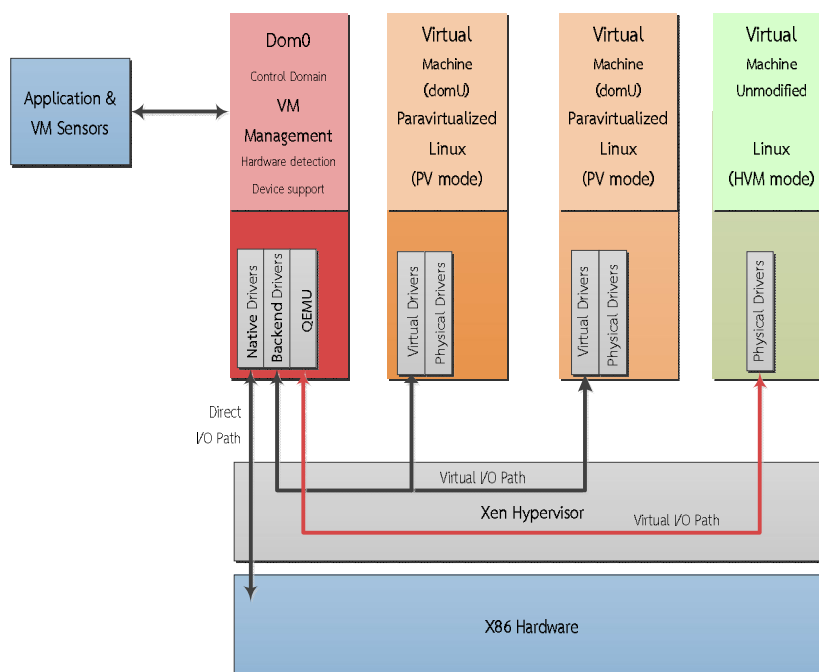
1.4.1 ศึกษาการทำงานของเครื่องเสมือนด้วย Xen Hypervisor

1.4.2 พัฒนาต้นแบบในการใช้ทำนายถึงสภาวะภาระงานต่างๆ ของการใช้ทรัพยากรของเครื่องจริงที่จะเกิดขึ้นในระบบปฏิบัติการเซินแบ่งออกเป็น 3 โมเดล



1.4.2.1 โมเดลรวบรวมข้อมูล (VM Sensors)

ทำหน้าที่ในการเก็บรวบรวมภาระงานที่เกิดจากการใช้ทรัพยากรต่างๆ ภายในเครื่อง เช่น หน่วยประมวลผล (CPU), หน่วยความจำ (Memory), ส่วนนำเข้า/ส่งออก (I/O device), หน่วยเก็บข้อมูล (DISK), การรับส่งข้อมูลทางเครือข่าย (Network) โดยการใช้ชุดคำสั่งเพื่อบ่งชี้ข้อมูลรายละเอียดที่ต้องการ จาก kernel ดังภาพประกอบ 1.1



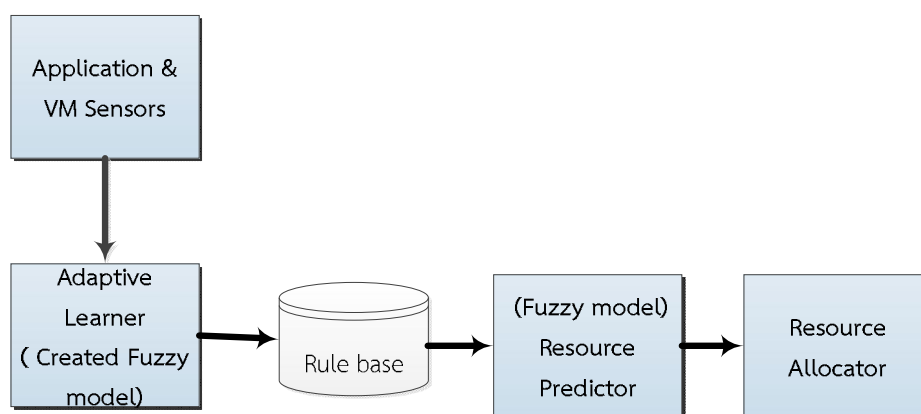
ภาพประกอบ 1.1 โครงสร้างการทำงานของโมเดลรวบรวมข้อมูล (VM Sensors)
ที่มา : [29]

1.4.2.2 โมเดลการวิเคราะห์ข้อมูล (Analysis Model)

ทำหน้าที่วิเคราะห์ข้อมูลที่ได้รับจากโมเดลรวบรวมข้อมูลเพื่อทำการวิเคราะห์ ตรวจสอบภาระงานต่างๆ ที่เกิดขึ้นภายใน Xen Hypervisor ว่าทำงานอยู่ในสถานะใดเพื่อสามารถบริหารจัดการทรัพยากรให้กับเครื่องเสมือนต่างๆ การวิเคราะห์จะใช้หลักการของตรรกศาสตร์คลุมเครือ (Fuzzy Logic) เพื่อทำการบริหารภาระงานภายในเครื่อง Xen Hypervisor โดยการบริหารทรัพยากรให้กับเครื่องเสมือนจะทำการบริหารจัดการทรัพยากร ดังนี้

1. หน่วยความจำเข้าถึงแบบสุ่ม (RAM)
2. หน่วยประมวลผลกลาง (CPU) ดังภาพประกอบ 1.2





ภาพประกอบ 1.2 โมเดลการวิเคราะห์ข้อมูล (Analysis Model)

1.4.3 สถานที่ที่ใช้ในการวิจัยคือ ห้องปฏิบัติการเครือข่ายคอมพิวเตอร์ สำนักคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม

1.5 นิยามศัพท์เฉพาะ

1.5.1 เครื่องเสมือน (VMs : Virtual Machines) หมายถึง ระบบปฏิบัติการที่ทำให้สามารถใช้ซอฟต์แวร์จำลองการทำงานของคอมพิวเตอร์เครื่องอื่นได้ มักใช้สำหรับเป็นตัวทดสอบโปรแกรมซอฟต์แวร์ต่าง ๆ เช่นวิศวกรอาจคิดสร้างคอมพิวเตอร์ขึ้นมาใหม่ แล้วใช้ซอฟต์แวร์จำลองการทำงานของเครื่องใหม่นี้ (กำหนดให้เครื่องเสมือนนี้ลองทำโปรแกรมดู) เพื่อดูว่ามีปัญหาอย่างไรหรือไม่ ถ้ามี ก็จะได้จัดการแก้ไขปัญหานั้นเสียก่อนที่จะลงมือผลิตเครื่องออกมาจริง ๆ

1.5.2 ระบบควบคุมการทำงานของเครื่องเสมือน (Hypervisor) หรือ Virtual Machine Monitor คือซอฟต์แวร์ที่เป็น Virtual Machine ประเภทหนึ่งทำงานอยู่บนลำดับชั้น (Layer) ขึ้นเดียวกับกับระบบปฏิบัติการหลัก (Host OS) ควบคุมดูแลระบบปฏิบัติการที่เป็นเสมือนแขกของระบบปฏิบัติการหลัก (Guest OS) เสมือนว่าระบบปฏิบัติการที่เป็นแขกเหล่านั้นเป็น Application หนึ่งๆและทำงานอยู่บนลำดับชั้น (Layer) เหนือกว่า

1.5.3 ซอฟต์แวร์โอเพนซอร์ซ (Open Source Software) เป็นชุดคำสั่ง หรือซอฟต์แวร์ที่เปิดเผยต้นฉบับให้ใช้งานหรือแก้ไขได้โดยไม่เสียค่าใช้จ่าย

1.5.4 การเฝ้าระวัง (Monitoring) หมายถึง การตรวจสอบ ติดตาม ความผิดปกติ ประสิทธิภาพ เพื่อให้ระบบสามารถทำงานได้อย่างต่อเนื่อง หากมีสัญญาณบ่งชี้ว่ากำลังเกิดปัญหาขึ้นจะได้เตรียมหาทางป้องกันได้

1.5.5 ตรรกะคลุมเครือ (Fuzzy Logic) พัฒนาจาก ทฤษฎีเซตวิภันย์ โดยเป็นการใช้เหตุผลแบบประมาณ ซึ่งแตกต่างจากการใช้เหตุผลแบบเด็ดขาดในลักษณะ ถูก/ผิด ใช่/ไม่ใช่ ของ ตรรกศาสตร์

1.5.6 แบบฉบับ (Classical Logic) ตรรกศาสตร์คลุมเครือนั้นสามารถถือเป็นการประยุกต์ใช้งานเซตวิภันย์ เพื่อจำลองการตัดสินใจของผู้เชี่ยวชาญ ต่อปัญหาที่ซับซ้อน



1.5.7 การประมวลผลแบบกลุ่มเมฆ (Cloud Computing) เป็นลักษณะของการทำงานของ
ผู้ใช้งานคอมพิวเตอร์ผ่านอินเทอร์เน็ต ที่ให้บริการใดบริการหนึ่งกับผู้ใช้ โดยผู้ให้บริการจะแบ่งปัน
ทรัพยากรให้กับผู้ต้องการใช้งานนั้น การประมวลผลแบบกลุ่มเมฆ เป็นลักษณะที่พัฒนาขึ้นต่อมาจาก
ความคิดและบริการของเวอร์ชวลไลเซชัน และเว็บเซอร์วิส โดยผู้ใช้งานนั้นไม่จำเป็นต้องมีความรู้ในเชิง
เทคนิคสำหรับตัวพื้นฐานการทำงานนั้น



บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้ได้ทำการวิเคราะห์ แนวคิดการแก้ไขปัญหาระบบภายใต้แนวคิดการเฝ้าระวัง และบริหารจัดการทรัพยากรระบบแบบอัตโนมัติ ด้วยหลักการ ตรรกะคลุมเครือ (Fuzzy Logic) สำหรับระบบปฏิบัติการเซ็น พร้อมการตรวจสอบความผิดพลาดที่เกิดขึ้นกับอุปกรณ์ต่าง ๆ ของคอมพิวเตอร์จริงในอนาคตได้ โดยมีทฤษฎีและงานวิจัยที่เกี่ยวข้องต่างๆ ดังนี้

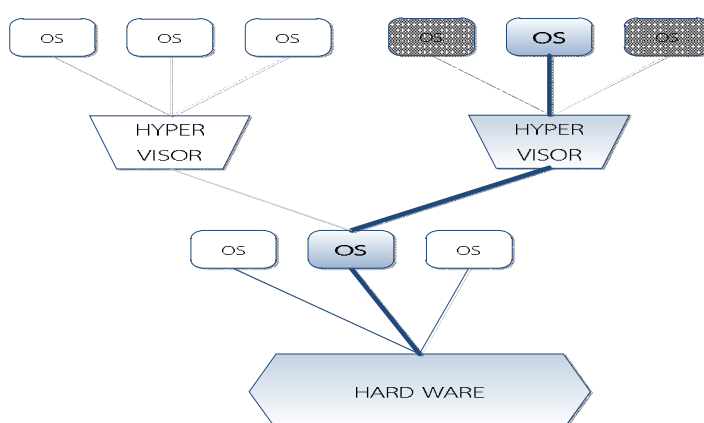
2.1 Virtualization Technology

Virtualization Technology หรือ เทคโนโลยีเสมือน [8,9] คือการจำลองทรัพยากรจริงในระบบคอมพิวเตอร์ไปเป็นระบบคอมพิวเตอร์เสมือนจริงหลายๆ ระบบทำให้สามารถใช้งานระบบปฏิบัติการและแอปพลิเคชันได้หลายๆ ระบบพร้อมกัน และมีความเป็นอิสระไม่ขึ้นอยู่กับทรัพยากรจริงเป็นอิสระต่อกันแม้จะต่าง Platform นอกจากนี้ยังหมายถึงการรวบรวมทรัพยากรด้านการประมวลผล การจัดเก็บข้อมูล และการติดต่อสื่อสารในแต่ละอุปกรณ์มารวมกันไว้ที่ศูนย์กลาง จากนั้นจึงให้ผู้ใช้สามารถนำทรัพยากรเหล่านั้นไปจัดสรรใช้ประโยชน์ได้ตามลักษณะงาน หรือความต้องการของแต่ละระบบโดยระบบเหล่านั้นถูกเรียกว่าเครื่องเสมือน (Virtual Machine) โดยสถาปัตยกรรมเทคโนโลยีเสมือนแบ่งออกเป็นสอง สถาปัตยกรรม [10] คือแบบ Hosted Architecture และแบบ Hypervisor Architecture

2.2 Hosted Architecture

Hosted Architecture [11] เป็นรูปสถาปัตยกรรมโดยมีหลักการทำงานคือ จะทำการเรียกการทำงานของระบบเสมือนอยู่บน ระบบปฏิบัติการของเครื่อง เช่นเดียวกับการเรียกใช้งานโปรแกรมประยุกต์ทั่วไป ดังภาพประกอบ 2.1 ซึ่งในสถาปัตยกรรมดังกล่าวมีข้อดีคือ สามารถรองรับฮาร์ดแวร์ได้หลากหลายกว่า

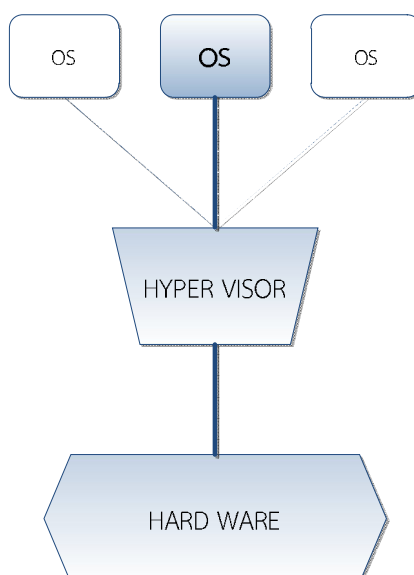




ภาพประกอบ 2.1 Hosted Architecture
ที่มา : [30]

2.3 Hypervisor Architecture

Hypervisor Architecture [12] เป็นรูปแบบสถาปัตยกรรมโดยที่มีหลักการทำงานคือ จะทำการเรียกการทำงานระบบเสมือนอยู่บน ฮาร์ดแวร์ ซึ่งเป็นระดับเดียวกับ ระบบปฏิบัติการเมื่อเปรียบเทียบกับ สถาปัตยกรรมแบบ Hosted Architecture ทำให้สามารถเข้าถึงทรัพยากรของฮาร์ดแวร์ได้โดยตรงไม่ต้องผ่านระบบปฏิบัติการอีก จึงทำให้สถาปัตยกรรมแบบ Hypervisor มีประสิทธิภาพเหนือกว่าแบบ Hosted



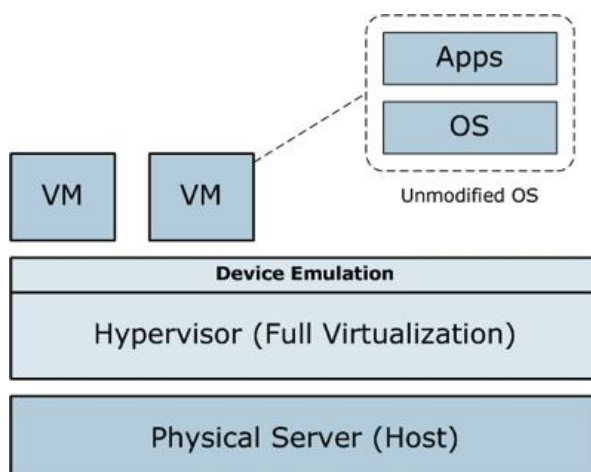
ภาพประกอบ 2.2 Hypervisor Architecture
ที่มา : [30]



2.4 Virtualization Method

Virtualization Method หรือเทคนิคการทำงานของเทคโนโลยีเสมือนเนื่องจากผู้พัฒนาซอฟต์แวร์ที่ Virtualization ต่างก็หาเทคนิคต่างๆ เพื่อพัฒนาเพิ่มเติมทำให้ Virtualization ตรงตามความต้องการและมีประสิทธิภาพมากยิ่งขึ้นซึ่งโดยทั่วไปแบ่งได้เป็น 3 เทคนิค [13] คือ

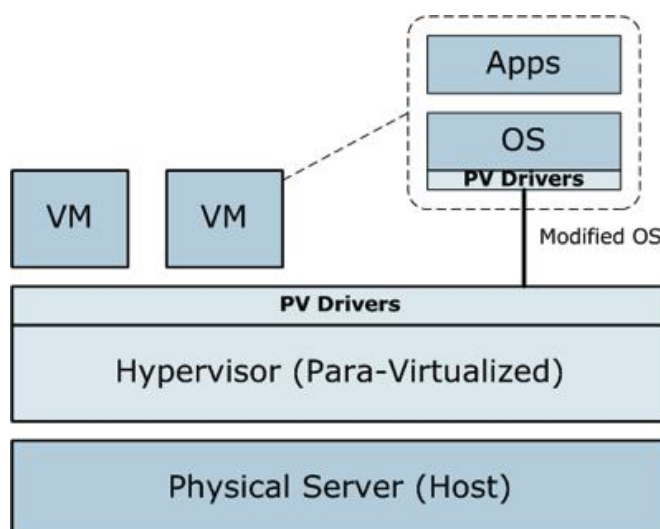
2.4.1 Full Virtualization เทคนิคนี้ถูกออกแบบเพื่อเตรียมการทำ ฮาร์ดแวร์ให้เป็นรูปแบบเสมือนทั้งหมด และสร้างระบบเสมือนที่สมบูรณ์ โดยเทคนิคนี้จะทำให้สามารถนำระบบปฏิบัติการอื่นๆ มาติดตั้งและสามารถที่จะทำงานอยู่บนเครื่องคอมพิวเตอร์เดียวกันได้ โดยเรียกระบบปฏิบัติการที่ติดตั้งเพิ่มนี้ว่า ระบบปฏิบัติการเยือน (Guest Operating System : Guest OS) โดยที่ระบบปฏิบัติการเยือนสามารถที่จะทำงานได้โดยไม่ต้องมีการเปลี่ยนแปลงแก้ไขเคอร์เนล (Kernel) กับคำสั่งที่ถูกร้องขอจาก ระบบปฏิบัติการเยือน หรือในตัวของมันเองเพราะฉะนั้นระบบปฏิบัติการเยือนหรือโปรแกรม จะไม่ทราบถึงสภาพแวดล้อมจำลองเสมือนจริงที่เกิดขึ้น จึงทำให้ระบบปฏิบัติการเยือนและโปรแกรมของทำงานอยู่บน เครื่องเสมือน ในขณะที่ในความเป็นจริง จะต้องทำงานบนสภาวะแวดล้อมเสมือนของระบบจริง (Physical system) วิธีการนี้ทำให้เกิดประโยชน์ เพราะว่ามันได้แยกการเชื่อมต่อของซอฟต์แวร์และระบบปฏิบัติการเยือน ออกจากฮาร์ดแวร์อย่างสมบูรณ์ ดังนั้นผลลัพธ์ของวิธีการแบบ Full virtualization ก็คือสามารถให้มีเส้นทางการเคลื่อนย้ายของตัวซอฟต์แวร์ และภาระงานต่างๆ (Workloads) ระหว่างระบบปฏิบัติการที่มีคุณสมบัติที่แตกต่างกัน ตัวอย่างของซอฟต์แวร์ที่ใช้เทคนิค Full virtualization เช่น Microsoft Virtual Server และ VMware ESX Server จากภาพประกอบ 2.3 จะเห็นได้ว่า เหนือ Hypervisor จะมี Device Emulation ทำหน้าที่จำลองอุปกรณ์เสมือนต่างๆให้ กับ VM ที่ทำงานอยู่บน Hypervisor



ภาพประกอบ 2.3 เทคนิค Full Virtualization
ที่มา : [31]



2.4.2 Para-virtualization เป็นเทคนิคที่มีลักษณะการทำงานคล้ายกับ เทคนิค Full Virtualization แต่สิ่งที่เทคนิคแตกต่างออกไปคือในเทคนิคนี้คือ ระบบปฏิบัติการเียนสามารถกำหนดการเข้าถึงอุปกรณ์ส่วนประสานต่างๆ ให้สามารถเข้าถึงไปยังการเชื่อมต่อจริงทางกายภาพได้โดยตรง ซึ่งการทำงานแบบนี้ต้องการอาศัยการปรับแต่งระบบปฏิบัติการเียนให้สามารถทำงานได้กับไฮเบอร์ไวเซสเพื่อให้สามารถทำงานได้ด้วยเทคนิคเสมือนนี้ได้ตัวอย่างของซอฟต์แวร์ที่ใช้เทคนิค Para-Virtualization ที่ได้รับความนิยมเช่น Xen



ภาพประกอบ 2.4 เทคนิค Para-Virtualization
ที่มา : [32]

จากภาพประกอบ 2.4 เทคนิค Para-Virtualization สำหรับระบบปฏิบัติการเียนจะมีการปรับแต่งเพื่อให้ส่วนอุปกรณ์การเชื่อมต่อ (PV Drivers) สามารถเข้าถึง Hypervisor ได้โดยตรง ต่างกับแบบ Full Virtualization ที่ต้องมีการแปลงค่าก่อนบน VM จึงจะส่งมายัง Hypervisor ได้

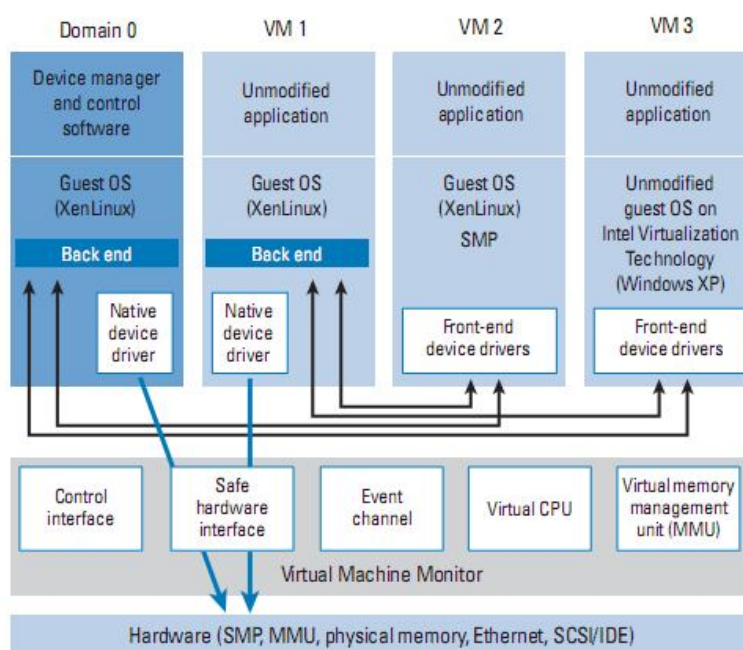
2.5 Xen Hypervisor

Xen Hypervisor[4] คือซอฟต์แวร์ ที่สามารถให้บริการให้คอมพิวเตอร์หลายๆ ระบบปฏิบัติการที่จะสามารถเรียกใช้ทรัพยากรได้ร่วมกัน โดยมีสถาปัตยกรรมการทำงานแบบ Hypervisor โดยผู้ที่คิดค้น Xen ในเวอร์ชันแรกคือ มหาวิทยาลัยเคมบริดจ์ (The University of Cambridge) ภายในห้องปฏิบัติการคอมพิวเตอร์ ของมหาวิทยาลัย โดยมี เอียน แพรตต์ เป็นผู้ดูแลภายใต้โครงการ Xen Source และ เมื่อปี ค.ศ. 2003 และต่อมาในเดือนตุลาคม ค.ศ. 2007 บริษัท Citrix ได้เข้ามาให้การสนับสนุน โดยถูกพัฒนาเป็น Free Software ภายใต้ License ของ the GNU General Public License (GPL2)



2.5.1 สถาปัตยกรรม Xen Hypervisor

Xen Hypervisor พัฒนาอยู่บนพื้นฐานเทคนิคแบบ Para-virtualization [14, 15, 16] จากภาพประกอบ 2.5 แสดงสถาปัตยกรรมของ Xen มี Hosting ประกอบไปด้วยเครื่องเสมือน 4 ตัว (Domain 0, VM1, VM2, และ VM3) โดยในสถาปัตยกรรม Xen ได้มีการรวมเอา Virtual Machine Monitor (VMM) ซึ่งเตรียมให้รูปแบบเสมือนของฮาร์ดแวร์ ทำให้สามารถเข้าถึงฮาร์ดแวร์ได้สำหรับซอฟต์แวร์เครื่องเสมือนที่แตกต่างกัน จากภาพประกอบ 2.5 ยังได้แสดงหน้าที่พิเศษของเครื่องเสมือนที่เรียกว่า Domain 0



ภาพประกอบ 2.5 สถาปัตยกรรมการ Xen Hypervisor

ที่มา : [29]

โดย Domain 0 เท่านั้นที่สามารถเข้าถึงตัวควบคุมที่เรียกว่า Control Interface ของ VMM โดยเครื่องเสมือนต่างๆที่เกิดขึ้นจากระบบจะถูกสร้าง ถูกทำลาย และ จัดการ จากส่วนนี้ ในการบริหารจัดการและควบคุมการทำงานของระบบ จะเริ่มจาก Domain 0 โดยผู้ดูแลระบบสามารถที่จะสร้าง เครื่องเสมือนกับการให้สิทธิพิเศษต่างๆ ตัวอย่างเช่น VM1 สามารถเข้าถึงฮาร์ดแวร์ได้โดยตรงผ่าน Secure Interface ที่เตรียมไว้โดย Xen ผู้ดูแลระบบสามารถสร้างเครื่องเสมือนอื่นๆ ที่สามารถเข้าถึงทรัพยากรของเครื่องที่จัดเตรียมไว้ให้ โดยการควบคุมของ Domain 0 และการจัดการอินเทอร์เฟซใน Xen

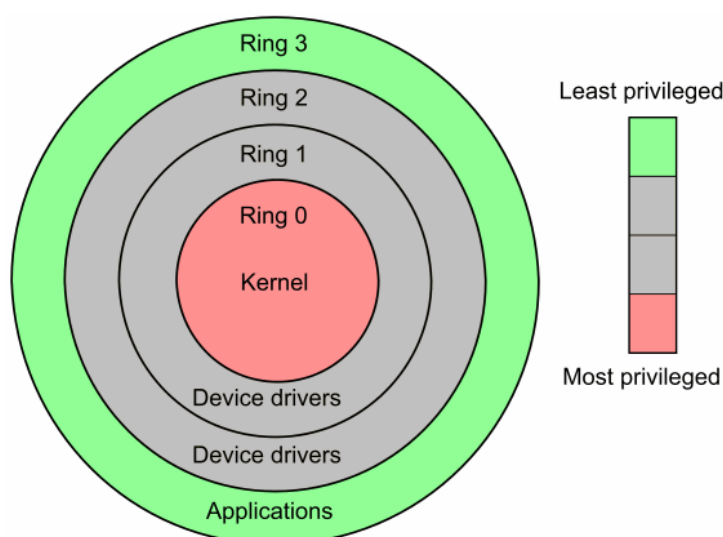
ในตัวอย่างนี้ จะเห็นว่าระบบปฏิบัติการเยื่อน ใน VM1 และใน VM2 ถูกเปลี่ยนแปลงแก้ไขให้ทำงานอยู่เหนือการทำงานของ Xen ด้วย ทำให้มีประสิทธิภาพในการทำงานสูง ใกล้เคียงกับประสิทธิภาพตามธรรมชาติของ ระบบปฏิบัติการเยื่อน วิธีการนี้สามารถถูกเรียกใช้ได้ตลอดเวลาการทำงาน การไม่มีการเปลี่ยนแปลงคำร้องขอใดๆ ของระบบปฏิบัติการเยื่อน (ในเทคนิคแบบ Full



Virtualization) ก็ได้ถูกรองรับด้วยเช่นกัน ดังนั้นเทคนิคแบบนี้จึงได้รับการพิจารณาใน Xen และ เทคโนโลยี เวกวลไลเซชันของอินเทล (Intel Virtualization Technology) นักพัฒนา Xen วางแผนที่จะรวมความสามารถในการรองรับความสามารถของเวอร์ชวลแมชีนกับ Symmetric multiprocessing (SMP), ระบบปฏิบัติการเยื่อนที่เป็นระบบปฏิบัติการแบบ 64 bits, การจัดการกับ Accelerated Graphics Port (AGP), และ Advanced Configuration and Power Interface (ACPI)

ในโครงสร้างหลักส่วนกลางที่สำคัญในการทำของข้อมูลเสมือนจริงคือการทำเสมือนจริงของตัวโปรเซสเซอร์(CPU), หน่วยความจำ(memory) และอุปกรณ์ I/O ในการออกแบบใน Xen จะต้องออกแบบให้มีความสามารถในการทำ เสมือนแบบ Para-virtualization ของโครงสร้างทั้งสามองค์ประกอบหลัก [17,18] ของฮาร์ดแวร์ประกอบด้วย

1. CPU Operations การทำงานของซีพียูเสมือน สถาปัตยกรรม x86 ของอินเทลเตรียมโหมดสิทธิพิเศษไว้ 4 ระดับ โดยโหมดเหล่านี้หรือเรียกว่า “rings” คือมีตัวเลขที่บอกเป็นตัวเลขระดับตั้งแต่ระดับ 0 ถึง 3 จากภาพประกอบที่ 2.6 กำหนดให้ 0 เป็นระดับที่มีค่าสิทธิพิเศษมากที่สุด ในระบบที่ไม่เป็นเทคนิคเสมือน ระบบปฏิบัติการ จะทำงานอยู่ที่ ring 0 และในส่วนของโปรแกรมจะทำงานที่ ring 3 ส่วนใน ring 1 และ ring 2 ปกติจะไม่ถูกนำมาใช้งาน ใน Xen รูปแบบ Para-virtualization การทำงานของซอฟต์แวร์ VMM จากภาพประกอบ 2.6 จะทำงานที่ระดับ ring 0 , ส่วนระบบปฏิบัติการเยื่อน จะทำงานอยู่ที่ ring 1 และ โปรแกรมต่างๆ จะทำงานที่ระดับ ring 3. วิธีการนี้ช่วยรับประกันได้ว่ากระบวนการการทำงานของ VMM ซึ่งใน Xen มันจะถูกเรียกว่า Hypervisor มีสิทธิพิเศษสูงที่สุด ในขณะที่ ระบบปฏิบัติการเยื่อน จะทำงานในสิทธิพิเศษที่สูงกว่าตัวโปรแกรม ดังนั้นคำสั่งและสิทธิพิเศษที่ถูกส่งออกไปโดยระบบปฏิบัติการเยื่อน จะถูกตรวจสอบและดำเนินการโดย VMM



ภาพประกอบ 2.6 แสดงการทำงานของการทำงาน CPU Virtualization
ที่มา : [33]



2. Memory Operations การทำงานของหน่วยความจำ ในสภาพแวดล้อมแบบ Non Para-virtualized ระบบปฏิบัติการ (Operating System) คาดว่าจะมีหน่วยความจำที่ต่อเนื่องกัน ในกระบวนการทำงานของ Xen ในการใช้เทคนิคแบบ Para-virtualization ระบบปฏิบัติการเยือนจะถูกเปลี่ยนแปลงแก้ไขในการเข้าถึงหน่วยความจำในแบบวิธี ที่ไม่ต่อเนื่องกัน ระบบปฏิบัติการเยือน จะรับผิดชอบในการจองและจัดการ page table อย่างไรก็ตามการเขียนข้อมูลโดยตรงจะถูกดักและตรวจสอบการทำงานด้วยโปรแกรม VMM ของ Xen ดังภาพประกอบ 2.5

3. IO operation การทำงานของไอโอ ในสภาพสิ่งแวดล้อม Fully virtualized อุปกรณ์ฮาร์ดแวร์คือสิ่งที่ถูกจำลองขึ้นมา ใน Xen รูปแบบ Para-virtualization จะทำงานอยู่บนพื้นฐานการทำงานที่เรียกว่า Producer - Consumer Pointer การทำงานเช่นนี้สามารถที่จะแยกระหว่าง Data กับ Event Notification ตัวอย่างการทำงานเช่น ข้อมูลที่เป็นข้อมูลของ I/O ถูกส่งและรับจากระบบปฏิบัติการเยือนนั้น จะถูกย้ายขึ้นไปทำงานบนสถาปัตยกรรม ring โดยใช้ในการแชร์หน่วยความจำ ผ่านทาง Buffer Descriptor Ring หรือที่เราเรียกว่า I/O Descriptor rings มีลักษณะเป็นโครงสร้างลักษณะแบบคิววงกลม (Circular Queues) การตอบสนองข้อมูลจะถูกส่งกลับไปให้ตัว I/O Descriptors Ring ทำให้หน่วยความจำถูกแชร์ ระหว่าง Domain 0 และ Guest Domain ตลอดเวลาที่มีการรับเข้าและส่งออกของข้อมูล ระบบปฏิบัติการเยือนจึงทำหน้าที่เสมือน Consumer และ Xen (ตัวโปรแกรม Hypervisor) จะทำหน้าที่เสมือน Producer การเปลี่ยนแปลงแก้ไขการร้องขอของระบบปฏิบัติการเยือน เป็นสิ่งไม่ยืดหยุ่น สำหรับ Platforms แบบ Non – Open Source เช่น ระบบปฏิบัติการ Microsoft Windows 2000 [19], ระบบปฏิบัติการ Windows Server 2003 [20], ผลลัพธ์ก็คือ ระบบปฏิบัติการเหล่านั้นจะไม่สนับสนุนสภาพแวดล้อมแบบ Para-virtualization นั้นเอง

2.6 ตรรกศาสตร์คลุมเครือ

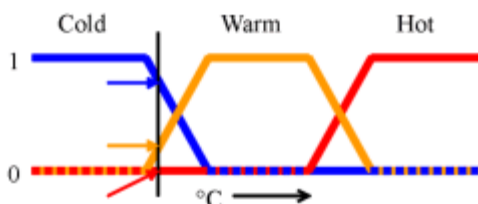
ตรรกศาสตร์คลุมเครือ หรือ ฟัซซีลอจิก (Fuzzy Logic) [21,22,23] พัฒนาจาก ทฤษฎีเซตวิชันนัย โดยเป็นการใช้เหตุผลแบบประมาณ ซึ่งแตกต่างจากการใช้เหตุผลแบบเด็ดขาดในลักษณะ ถูก/ผิด ใช่/ไม่ใช่ ของ ตรรกศาสตร์แบบฉบับ (Classical Logic) ตรรกศาสตร์คลุมเครือนั้นสามารถถือเป็นการประยุกต์ใช้งานเซตวิชันนัย เพื่อจำลองการตัดสินใจของผู้เชี่ยวชาญ ต่อปัญหาที่ซับซ้อนค่าระดับความจริง ในตรรกศาสตร์คลุมเครือนั้นมักจะสับสนกับ ค่าความน่าจะเป็น ซึ่งมีแนวความคิดที่แตกต่างกัน ค่าระดับความจริงคลุมเครือนั้นใช้ในการระบุ ค่าความเป็นสมาชิก ของเซต แต่ค่าความน่าจะเป็นนั้นระบุความเป็นไปได้ของสภาพการณ์แต่ละรูปแบบที่อาจจะเกิดขึ้น ตัวอย่างเช่น สมมติ นาย ก กำลังเดินเข้าบ้าน สถานะของนาย ก ตามตรรกศาสตร์แบบฉบับ คือ "อยู่ในบ้าน" หรือ "อยู่นอกบ้าน" หากเขากำลังยืนอยู่ระหว่างช่องประตูละ เราอาจพิจารณาว่าเขา "อยู่ในบ้านบางส่วน" ระดับของสถานะกึ่งนี้ จะระบุด้วยค่าความเป็นสมาชิกของเซตวิชันนัย สมมติเขาเพิ่งจะก้าวปลายนิ้วเท้าผ่านข้ามธรณีประตูเข้าบ้าน เราอาจกล่าวว่า นาย ก นั้น 0.99 "อยู่นอกบ้าน" ซึ่งต่างจากความน่าจะเป็นของเหตุการณ์สุ่ม (เช่น ความน่าจะเป็นระบุผลลัพธ์ของการโยนเหรียญ แต่ผลลัพธ์จะออก หัว หรือ ก้อย) หากพิจารณาความน่าจะเป็นที่นาย ก "อยู่นอกบ้าน" และ "อยู่ในบ้าน" จะออกผลลัพธ์เป็น นาย ก อยู่นอกบ้าน หรือ ในบ้าน ไม่ได้จำลองสถานะกึ่ง คือ กำลังยืนอยู่ที่ประตู เซตวิชันนัยนั้นมีหลักการพื้นฐานจากเซตที่มีขอบเขตคลุมเครือไม่ชัดเจน ไม่ได้มีพื้นฐานจากการสุ่ม



ตรรกศาสตร์คลุมเครือนั้น สามารถระบุค่าความเป็นสมาชิกของเซต (Set Membership Values) ด้วยค่าระหว่าง 0 และ 1 ทำให้เกิดระดับถึงในลักษณะของ สีเทา นอกจาก ขาว และ ดำ ซึ่งมีประโยชน์ในการจำลองระดับซึ่งสามารถระบุด้วยคำพูด "เล็กน้อย" "ค่อนข้าง" "มาก" โดยใช้ค่าความเป็นสมาชิกของเซตบางส่วน ตรรกศาสตร์คลุมเครือนี้มีความสัมพันธ์กับ เซตวิภังค์ (en : fuzzy set) และ ทฤษฎีความเป็นไปได้ (en : possibility theory) ซึ่งคิดค้นขึ้นในปี ค.ศ. 1965 โดยศาสตราจารย์ ลอตฟิ ซาเดห์ แห่งมหาวิทยาลัยแห่งรัฐแคลิฟอร์เนีย เบิร์กลีย์ ตรรกศาสตร์คลุมเครือ ถึงแม้ว่าจะได้รับการยอมรับค่อนข้างกว้างขวาง แต่ก็ยังถูกโต้แย้งจากบางกลุ่ม เช่น จากวิศวกรระบบควบคุม ในเรื่องของการอธิบายพฤติกรรมต่างๆ และ จากนักสถิติบางกลุ่ม ซึ่งถือนั่นว่าทฤษฎีความน่าจะเป็น เป็นวิธีทางคณิตศาสตร์ที่เคร่งครัดเพียงวิธีเดียว ในการจำลองความไม่แน่นอน (en : uncertainty) นอกจากนั้นแล้ว ก็ยังมีการวิเคราะห์วิจารณ์ว่า เซตวิภังค์ นั้นไม่ได้เป็นซูปเปอร์เซตของ ทฤษฎีเซตสามัญ เนื่องจากฟังก์ชันภาวะสมาชิก นั้นกำหนดในรูปของ เซตแบบดั้งเดิม

2.6.1 การประยุกต์ใช้งาน

ตรรกศาสตร์คลุมเครือนั้น สามารถใช้ในการควบคุม อุปกรณ์เครื่องใช้ในครัวเรือน[24] เช่น เครื่องซักผ้า (โดยการวัดปริมาณผ้า และ ความเข้มข้นของน้ำยาซักผ้า และปรับวงจรการซักให้เหมาะสม) และ ตู้เย็นวิธีการใช้งานง่าย ๆ นั้นอาจใช้ในการจำลองช่วงย่อยๆ ของตัวแปรที่มีค่าต่อเนื่อง เช่น การวัดอุณหภูมิในระบบห้ามล้อแบบกันล้อตาย อาจมีฟังก์ชันความเป็นสมาชิกของเซตหลายฟังก์ชัน สำหรับอุณหภูมิซึ่งแบ่งเป็นหลายช่วง เพื่อควบคุมการห้ามล้อให้เหมาะสม โดยแต่ละฟังก์ชันจะทำการส่งค่าอุณหภูมิหนึ่งๆ ไปเป็นค่าความจริงในช่วง 0 ถึง 1 ซึ่งค่าความจริงเหล่านี้จะถูกนำไปใช้ในการควบคุมการห้ามล้อ



ภาพประกอบ 2.7 การประยุกต์ใช้งานฟัซซี
ที่มา : [34]

จากภาพประกอบ 2.7 cold (เย็น) warm (อุ่น) and hot (ร้อน) เป็นฟังก์ชันในการส่งค่าระดับอุณหภูมิ ที่แต่ละจุดของอุณหภูมิจะมีค่าความจริง 3 ค่า ซึ่งเป็นค่าของแต่ละฟังก์ชัน ซึ่งค่าความจริงทั้งสามนี้ สามารถใช้ในการตีความค่าอุณหภูมิใดๆ ว่า "ค่อนข้างเย็น" "อุ่นนิดๆ" "ไม่ร้อน" ตัวอย่างของการใช้งานที่ซับซ้อนขึ้น ก็คือ การใช้งานตรรกศาสตร์คลุมเครือใน การแก้ความผิดพลาด (error correction) ประสิทธิภาพสูง เพื่อเพิ่มประสิทธิภาพของการรับข้อมูลผ่านช่องสัญญาณแบนด์วิดท์จำกัด ที่ข้อมูลถูกรบกวนด้วยสัญญาณรบกวน โดยการเข้ารหัสเทอร์โบ (en : turbo code) ส่วนต้นของภาคถอดรหัส จะวัดค่าความควรจะเป็น ของค่าบิตในสายบิตที่ส่งมาจากภาคส่ง (ว่าควรจะเป็น 0 หรือ 1) ค่าความควรจะเป็นอาจแบ่งเป็น 256 ระดับ ระหว่างระดับสูงสุดหมายถึง "ค่าควรจะเป็น 1 แน่แน่นอน" และต่ำสุด



หมายถึง "ค่าควรจะเป็น 0 แน่แน่นอน" ตัวถอตรหัสสองตัวอาจวิเคราะห์ข้อมูลที่ได้รับมาพร้อมกัน ได้เป็นค่าความควรจะเป็นที่ต่างกัน ตัวถอตรหัสแต่ละตัวสามารถใช้ค่าความควรจะเป็นที่ได้จากตัวถอตรหัสอื่นเข้าช่วยในการตีความ จนได้ข้อสรุปค่าที่ควรจะเป็นมากที่สุด

2.6.2 การประยุกต์ใช้ตรรกศาสตร์คลุมเครือ

การประยุกต์ใช้งานตรรกศาสตร์คลุมเครือโดยทั่วไป [25, 26] จะใช้ในการจำลองความรู้ หรือประสบการณ์ของผู้เชี่ยวชาญ โดยการใช้เหตุผล หรือ การตัดสินใจต่อสภาวะการณ์ต่างๆ ของมนุษย์นั้น สามารถเขียนอยู่ในรูปเชิงภาษาศาสตร์ของ ระบบกฎเกณฑ์ (Rule-Based System) คือ เงื่อนไข IF/THEN หรือ อยู่ในรูปอื่นที่เท่าเทียมกัน เช่น เมทริกซ์เปลี่ยนหมู่ฟัซซี (Fuzzy Associative Matrices)

กฎเกณฑ์:

IF(ถ้า)<เงื่อนไข>THEN(แล้ว)<ผลที่ตามมา>

การใช้เหตุผล การตัดสินใจ หรือ การตอบสนองต่อเหตุการณ์ต่างๆ ของมนุษย์นั้น โดยปกติจะมีลักษณะที่คลุมเครือ เช่นการประเมินสภาวะการณ์ หรือ การระบุการตอบสนอง โดยไม่ได้ระบุเป็นค่าที่แน่นอนชัดเจน ดังนั้นจึงถูกจำลองไว้ในกฎเกณฑ์ด้วย เซตวิภังค์

ตัวอย่าง เช่น ระบบควบคุมอุณหภูมิโดยใช้พัดลม อาจมีกฎเกณฑ์ดังนี้:

IF อุณหภูมิ เย็นมาก THEN หยุดพัดลม
IF อุณหภูมิ เย็น THEN ปรับพัดลมให้ช้าลง
IF อุณหภูมิ ปานกลาง THEN รักษาระดับความเร็ว
IF อุณหภูมิ ร้อน THEN ปรับพัดลมให้เร็วขึ้น

สังเกตว่าไม่มีการใช้ ELSE (ไม่เช่นนั้น) ทุกเงื่อนไขจะต้องถูกนำมาพิจารณา เนื่องจากสภาพกำลังในตรรกศาสตร์คลุมเครือ เช่น อุณหภูมิอาจเป็นสมาชิกของ ทั้งเซต "เย็น" และ "ปานกลาง" โดยอาจมีค่าระดับความเป็นสมาชิกของแต่ละเซตต่างกัน ซึ่งชุดของกฎเกณฑ์ดังกล่าวนี้ เรียกว่า กฎเกณฑ์ฟัซซี (fuzzy rule) ฐานกฎเกณฑ์ฟัซซี (fuzzy rule base) หรือ ฐานความรู้ (knowledge base)การดำเนินการทางตรรกะ เช่น AND (และ) OR (หรือ) NOT (ไม่) นั้นก็มีใช้สำหรับตรรกศาสตร์คลุมเครือ สำหรับการสร้างเงื่อนไขของเหตุการณ์ที่ซับซ้อน เช่นเดียวกับตรรกศาสตร์แบบฉบับ โดยปกติแล้วจะนิยาม AND=minimum OR=maximum และ NOT=complementซึ่งการนิยามตามแบบนี้เรียกว่า ตัวดำเนินการซาเดห์ (Zadeh operator) เนื่องจากเป็นไปตามลักษณะนิยามที่ ซาเดห์ ใช้ในบทความดั้งเดิมของเขา ดังนั้นหากเรามีตัวแปรฟัซซี x และ y:



$$\begin{aligned}\text{NOT } x &= (1 - \text{truth}(x)) \\ x \text{ AND } y &= \text{minimum}(\text{truth}(x), \text{truth}(y)) \\ x \text{ OR } y &= \text{maximum}(\text{truth}(x), \text{truth}(y))\end{aligned}$$

นอกจากนี้แล้วยังมีตัวดำเนินการอื่นที่สื่อความหมายในเชิงภาษาพูด เรียกในภาษาอังกฤษว่า Hedges หรือ Linguistic Hedges (หมายถึง คำที่ทำให้คลุมเครือ) เช่น "มาก" "บ้าง" ซึ่งใช้ในการแปลงความหมายของเซตโดยใช้วิธีการทางคณิตศาสตร์ในทางปฏิบัติ การนำกฎเกณฑ์ความชำนาญนี้มาใช้งาน มักจะเกี่ยวข้องกับค่าที่แน่นอน เช่นค่าที่ได้จากการวัด และค่าที่ส่งออกไปใช้งานก็มักจะเป็นค่าที่แน่นอนเช่นเดียวกัน ส่วนที่ทำการแปลงค่ารับเข้าที่เป็นค่าแน่นอนนี้ ไปสู่เซตภัณัย ด้วยฟังก์ชันภาวะสมาชิก เรียกว่า ตัวทำให้คลุมเครือ (Fuzzifier) และส่วนที่ทำการส่งค่าผลลัพธ์จากเงื่อนไขไปสู่ค่าแน่นอนเพื่อส่งออกไปใช้งานจริง เรียก ตัวกำจัดความคลุมเครือ (Defuzzifier) นอกจากนั้นแล้ว ในการสร้างกฎเกณฑ์ฟัซซี ที่มีตัวแปรค่ารับเข้าเป็นจำนวนมาก ให้ครอบคลุมทุกเงื่อนไขนั้นเป็นไปได้ยาก ในทางปฏิบัติ บางครั้งจึงต้องมีการผสมเงื่อนไขหลายเงื่อนไขในกฎเกณฑ์ฟัซซีเข้าด้วยกัน โดยมีส่วนตีความผลลัพธ์รวมจากการผสมเงื่อนไข เรียก เครื่องอนุมานฟัซซี (Fuzzy Inference Engine) ซึ่งมีหลายชนิดด้วยกัน

2.6.3 ฟังก์ชันความเป็นสมาชิก

ฟังก์ชันความเป็นสมาชิก (Membership Function) เป็นฟังก์ชันที่มีการกำหนดระดับความเป็นสมาชิกของตัวแปรที่ต้องใช้งาน โดยการกำหนดระดับความเป็นสมาชิกที่มีความไม่ชัดเจน ไม่แน่นอน และคลุมเครือ ดังนั้นจึงเป็นส่วนสำคัญต่อการดำเนินการของฟัซซี เพราะรูปร่างของฟังก์ชันความเป็นสมาชิกมีความสำคัญต่อกระบวนการคิดและแก้ไขปัญหา โดยฟังก์ชันความเป็นสมาชิกมีความสำคัญต่อกระบวนการคิดและแก้ไขปัญหาโดยใช้ฟังก์ชันความเป็นสมาชิกมีความสำคัญต่อกระบวนการคิดและแก้ไขปัญหา โดยฟังก์ชันความเป็นสมาชิกที่งานทั่วไปมีหลายชนิดแต่ในที่นี้จะกล่าวถึงเพียง 4 ชนิด [27] ดังนี้

1) ฟังก์ชันสามเหลี่ยม (Triangular Membership Function) เป็นฟังก์ชันความเป็นสมาชิกที่มีพารามิเตอร์ทั้งหมด 3 ค่า คือ $\{a, b, c\}$ ดังสมการที่ 1

$$\text{triangular}(x; a, b, c) = f(x) = \begin{cases} 0, & x < a \\ (x - a)/(b - a), & a \leq x < b \\ (c - x)/(c - b), & b \leq x \leq c \\ 0, & x > c \end{cases} \quad (1)$$

2) ฟังก์ชันสี่เหลี่ยมคางหมู (Trapezoidal Membership Function) เป็นฟังก์ชันความเป็นสมาชิกที่มีพารามิเตอร์ทั้งหมด 4 ค่าคือ $\{a, b, c, d\}$ ดังสมการที่ 2



$$\text{trapezoidal}(x: a, b, c, d) = f(x) = \begin{cases} 0, & x < a \\ (x-a)/(b-a), & a \leq x < b \\ 1, & b \leq x < c \\ (d-x)/(d-c), & c \leq x < d \\ 0, & x \geq d \end{cases} \quad (2)$$

3) ฟังก์ชันเกาส์เซียน (Gaussian Membership Function) เป็นฟังก์ชันความเป็นสมาชิกที่มีพารามิเตอร์ทั้งหมด 2 ค่า คือ $\{m, \sigma\}$ ซึ่ง m หมายถึงค่าเฉลี่ย และ σ หมายถึงค่าเบี่ยงเบนมาตรฐาน ดังสมการที่ (3)

$$\text{gaussian}(x: m: \sigma) = \exp\left(-\frac{(x-m)^2}{\sigma^2}\right) \quad (3)$$

4) ฟังก์ชันระฆังคว่ำ (Bell-shaped Membership Function) เป็นฟังก์ชันความเป็นสมาชิกที่มีพารามิเตอร์ทั้ง 3 ค่า คือ $\{a, b, c\}$ ดังสมการ ที่ (4)

$$\text{bell-shaped}(x: a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \quad (4)$$

การเลือกฟังก์ชันของความเป็นสมาชิกขึ้นอยู่กับความเหมาะสมของข้อมูลที่ได้รับเข้ามา โดยสามารถที่ทับซ้อนกันเพื่อให้การดำเนินงานราบเรียบ ซึ่งมีความเป็นสมาชิกหลายค่าได้และฟังก์ชันความเป็นสมาชิกเปลี่ยนแปลงแก้ไขให้เหมาะสมกับงานที่กำลังปฏิบัติหรือตามความต้องการ

2.7 Web Stress Tool

Web Stress Tool 7 [35] คือชุดโปรแกรมทดสอบประสิทธิภาพการให้บริการของ เว็บเซิร์ฟเวอร์ (Web Server) ที่จำลองการทำงานของเครื่องลูกข่าย (Client) โดยทำการเรียกข้อมูลมายังเครื่อง เว็บเซิร์ฟเวอร์ โดยจะทำการจำลองการร้องขอ ทางโปรโตคอล HTTP ที่สร้างขึ้นมามากมายหรือหลายพันผู้ใช้พร้อมกัน ซึ่งออกแบบให้เหมือนกับการทำงานในสภาพแวดล้อมจริง ในการวัดประสิทธิภาพและ ตรวจสอบปัญหาการทำงานของเครื่องที่ให้บริการเป็น เว็บเซิร์ฟเวอร์ เพื่อให้ทราบถึงขีดความสามารถในการให้บริการ ภายใต้สภาพแวดล้อมการใช้งานต่างๆ จากผู้ใช้บริการ โดยที่นิยมใช้ในการทดสอบสามารถแบ่งออกเป็น 2 รูปแบบดังนี้

2.7.1 แบบ CLICK การทดสอบรูปแบบดังกล่าวจะสิ้นสุดกระบวนการเมื่อผู้ใช้แต่ละคนได้เริ่มทำการคลิกงานตามจำนวนที่ได้กำหนดให้ นิยมใช้ในการทดสอบเว็บเซิร์ฟเวอร์ ที่มีหน้าเว็บไซต์ให้บริการตามลำดับการเข้าถึงข้อมูล ตัวอย่างเช่น กำหนดให้ User เท่ากับ 50 โดยมีการคลิกหน้าเว็บไซต์ User ละ 3 ครั้ง ให้ระยะเวลาในการรอข้อมูลระหว่าง User และ เว็บเซิร์ฟเวอร์ เท่ากับ 30 วินาที การทดสอบจะดำเนินการโดย ระบบจะจำลองการทำงานในการ คลิกหน้าเว็บไซต์เพื่อเรียกข้อมูลจาก เว็บเซิร์ฟเวอร์ เมื่อครบจำนวน 50 แล้ว จะแสดงผลลัพธ์ของระยะเวลาในการให้บริการ และ จำนวนการร้องขอที่มีการคลิกของที่เปิดหน้าเว็บไซต์ได้สำเร็จ



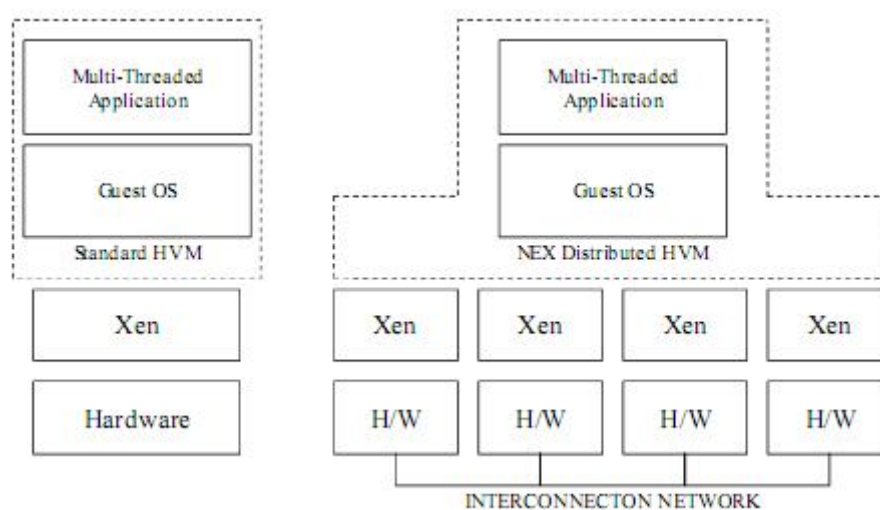
2.7.2 แบบ TIME การทดสอบรูปแบบดังกล่าวจะเป็นการทดสอบโดยการระบุเวลาในการทดสอบประสิทธิภาพของเว็บเซิร์ฟเวอร์ เพื่อให้ทราบถึง ระยะเวลาที่ต้องการใช้ในการทดสอบเว็บเซิร์ฟเวอร์ ในความสามารถให้บริการได้ในจำนวนคลิกเท่าไรต่อจำนวน User ที่ระบุ ตัวอย่างเช่น กำหนดให้ User เท่ากับ 50 โดยให้เวลาในการทำงานเท่ากับ 1 นาที เพื่อให้ระบบจำลองการทำงานของ แต่ละ User ได้จำนวนคลิกสูงสุดที่ประสิทธิภาพของเว็บเซิร์ฟเวอร์ให้บริการได้ ใน 1 นาที

2.8 งานวิจัยที่เกี่ยวข้อง

การตรวจสอบและการศึกษาถึงประสิทธิภาพในการทำงานในของ Xen Hypervisor ที่ถูกนำเสนอในปัจจุบัน

2.8.1 Virtual Machine Monitor Level Single System Support in Xen (NEX)

NEX [5] เป็นระบบการทำงานแบบกระจาย ซึ่งประกอบด้วย hypervisors หลายเครื่อง จากภาพประกอบ 2.8 แสดงให้เห็นถึงรูปแบบโครงสร้างการทำงาน ของ NEX โดย การใช้ hypervisors ที่ทำงานในแต่ละโหนด โดย NEX จะทำงานครอบคลุมในเครื่องเสมือนที่ อยู่กระจายโดยทำงานอยู่ตามระบบที่มีโครงสร้างข้างในที่แตกต่างกันในแต่ละเครื่องเสมือน เพื่อทำการตรวจสอบข้อมูลจากเครื่องเสมือนต่างๆ



ภาพประกอบ 2.8 รูปแบบโครงสร้างการทำงาน ของ NEX
ที่มา : [5]

NEX สามารถทำงานได้เป็นอย่างดี ใน เครื่องเสมือนต่างๆ ซึ่งสนับสนุนการทำงานบนระบบปฏิบัติการ Xen โดยตาราง 2.1 จะแสดงข้อมูลส่วนเชื่อมต่อที่สนับสนุน



ตาราง 2.1 ข้อมูลส่วนเชื่อมต่อที่สนับสนุน

Machine subsystem	Virtualization Method
Memory	Hardware assisted paging and distributed shared memory protocol
Processor	Virtual processor scheduling : Emulation on chip devices over nodes; Extended guest firmware
Instruction Set	Hardware assisted full virtualization
Peripheral devices	DSM assisted device emulation

2.8.2 Automatic virtual resource management for hosting service platforms

Automatic virtual resource management for hosting service platforms

งานวิจัยของ Hein Nguyen และคณะ [6] เป็นการแก้ปัญหาการบริหารจัดการสำหรับผู้ให้บริการโฮสต์ โดยใช้สถาปัตยกรรมแบบสองระดับ (two-level) โดยการแยกการใช้งานโปรแกรมเฉพาะ ออกจาก การใช้งานทั่วไปด้วยการแบ่งออกเป็น 2 ชั้น โดยจะนำโฮสต์เข้าการทดสอบ ทั้ง 2 รูปแบบ เพื่อหาประสิทธิภาพการทำงานโดยมีจุดประสงค์เพื่อให้ การทำงานของโฮสต์ ทั้งแบบ application และ แบบ objective ซึ่งจะส่งผลต่อการทำงานของเครื่องเสมือนที่ทำงานอยู่บนเครื่องจริง และจะนำไปสู่ขั้นตอนการปรับแต่งประสิทธิภาพด้วยตนเอง โดยจะครอบคลุมถึงความพึงพอใจในข้อจำกัดโดยการใช้หลักการ SLA

2.8.3 QRSM

Quadratic Response Surface Model (QRSM) [7] ถูกออกแบบมาเพื่อใช้

ในการศึกษาพฤติกรรมในการทำงานของภาระงานต่างๆ โดยการใช้การทดสอบเพื่อแสดงให้เห็นถึงความต้องการในรูปแบบการทำงานเฉพาะด้าน โดยการใช้รูปแบบการใช้งานระยะยาว และในการทำงานที่เน้นประสิทธิภาพ จะใช้รูปแบบการทำงาน ระยะสั้น โดยการทำงานของเครื่องเสมือนจะครอบคลุมกลไก ซึ่งจะเป็นการเปรียบเทียบในส่วนของ คลัสเตอร์ต่างๆ และการใช้เทคนิครูปแบบ k-mean

2.8.4 Resource Allocation Techniques in Cloud computing-Research

Challenges for Application

Resource Allocation Techniques in Cloud computing-Research

Challenges for Application [28] โดยงานวิจัยนี้ จะมุ่งเน้นการนำเสนอรูปแบบการทำงานของ คลาวด์ที่ให้บริการผ่านเครือข่ายอินเทอร์เน็ต โดยจะครอบคลุม การจัดสรรทรัพยากร ตารางลำดับการทำงาน และการร้องขอทรัพยากรที่มีความแตกต่างออกไปจากรูปแบบทั่วไป โดยในงานวิจัยจะมุ่งเน้น ศึกษาถึงการประมาณการต่างๆ ของการจัดสรรทรัพยากรโดยใช้กลยุทธ์ สำหรับประยุกต์ใช้ใน ระบบ คลาวด์นี้ ซึ่งสามารถแบ่งรูปแบบการจัดสรรทรัพยากรได้ 9 รูปแบบดังนี้



1) แบบ Optimized Resource scheduling algorithm เป็นรูปแบบที่นำมาประยุกต์ใช้สำหรับการทำงาน ที่มีรูปแบบภาระงานที่มีความแน่นอน โดยการปรับแต่งดังกล่าวจะมุ่งเน้นที่ส่วนของ อุปกรณ์ หรือทรัพยากรของเครื่องโดยตรง

2) แบบ Resource allocation strategy based on market (RAS-M) เป็นรูปแบบการจัดสรรทรัพยากร การให้บริการ โดยมุ่งเน้นจาก ผลตอบแทนที่ได้รับ โดยการจัดสรรทรัพยากรจะมีค่าความสำคัญสูงหากมีรายได้หรือผลกำไรเกิดขึ้น ณ ช่วงเวลานั้นๆ เพื่อให้ทรัพยากรของเครื่องมีเพียงพอต่อการให้บริการ

3) แบบ Scheduling with multiple SLA parameters คือรูปแบบการให้บริการที่เป็นข้อตกลง โดยการทำข้อตกลงจะทำให้หลายรูปแบบในการให้บริการเพื่อให้สอดคล้องต่อการให้บริการของระบบ เช่น การให้บริการทางกายภาพ มีการขยายหรือปรับขนาดทรัพยากรของระบบตามความเหมาะสมของงานนั้น การขยายหรือเพิ่มจำนวนการทำงานด้านโปรแกรม และการปรับเปลี่ยนการทำงานของระบบปฏิบัติการที่ให้บริการให้เหมาะสมกับการให้บริการมากที่สุด

4) แบบ Rule Based Resource Allocation Model (RBRAM) เป็นรูปแบบที่กำหนดขึ้นจาก กฎข้อบังคับของ ระบบ โดยระบบตกอยู่ในสถานการณ์ให้บริการใดๆ ตามที่กำหนดไว้แล้วสอดคล้องกับเงื่อนไขที่ได้ กำหนดขึ้นไว้ก่อนหน้านี้ จะมีเรื่องการใช้กฎหรือข้อบังคับ นั้นๆมาทำการจัดสรรทรัพยากรเครื่อง

5) แบบ Resource allocation using Scalable computing คือการทำงานในรูปแบบการจัดสรรทรัพยากร จากความต้องการของทรัพยากรโดยสามารถปรับเปลี่ยนได้ จากการเรียนรู้รูปแบบการใช้งานโดยการกำหนดค่าขึ้นสำหรับให้บริการ

6) แบบ Fair Resource Allocation for congestion Control เมื่อมีทรัพยากรต่างๆในเครื่องให้บริการ มีความต้องการ การใช้งานพร้อมกัน การจัดสรรทรัพยากรไปให้เครื่องต่างๆ อาจไม่เท่าเทียมกัน จึงทำให้เกิดความไม่เท่าเทียมในการให้บริการ การใช้รูปแบบนี้จึงเป็นการให้บริการทรัพยากรของเครื่องที่เรียกใช้งาน มีความเป็นธรรมเท่ากัน

7) แบบ Service-request prediction model เป็นรูปแบบการทำงานที่อาศัยข้อมูลความต้องการของระบบ ณ เวลา นั้นๆ เพื่อคาดการณ์ความต้องการที่คาดหวังจะเกิดขึ้นเพื่อเตรียมพร้อมทรัพยากรให้ระบบ โดยส่วนมากทรัพยากรจะถูกเตรียมไว้ก่อนอยู่แล้ว เพื่อสามารถให้บริการขึ้นมาใช้ได้ทันที

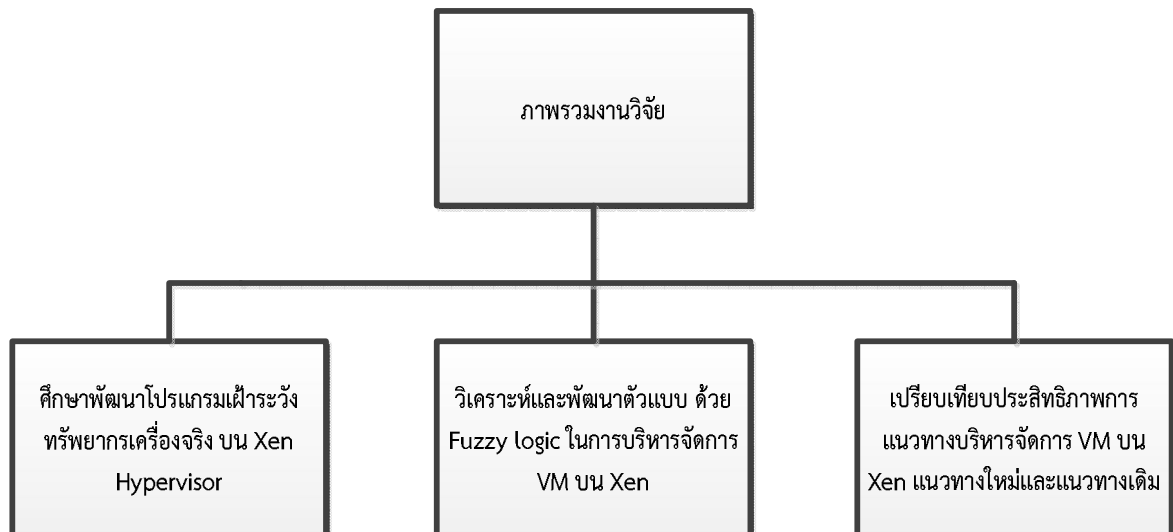
8) แบบ Balanced scheduling through genetic algorithm เป็นร่วมแบบการทำงานที่เน้นถึงความสมดุลในการจัดการตารางงานด้วยอัลกอริทึมแบบพันธุกรรม ที่จะจับคู่ภาระงานในเครื่องเสมือนเมื่อมีการเรียกใช้ทรัพยากร โดยการจัดคู่จะทำการหาเครื่องเสมือนที่มีภาระงานสูงและภาระงานต่ำทำให้ระบบลดภาระงานของภาพรวมที่เกิดขึ้น



บทที่ 3

วิธีดำเนินการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำผลการการวิจัยที่ได้ไปใช้ในการปรับปรุงประสิทธิภาพและเฝ้าระวังการให้บริการของ ระบบ Xen Hypervisor ซึ่งผู้วิจัยได้กำหนดวิธีดำเนินการวิเคราะห์ปรับปรุงประสิทธิภาพในการใช้ความสามารถเรียนรู้ลักษณะการทำงานในการบริหารจัดการทรัพยากร และเฝ้าระวังทรัพยากรได้อย่างมีประสิทธิภาพอย่างต่อเนื่องโดยในงานวิจัยนี้จะนำหลักการ ตรรกะคลุมเครือ (Fuzzy logic) มาเป็นตัวแบบในการศึกษาเพื่อใช้ทำนายถึงสภาวะภาระงานต่างๆในการบริหารจัดการการใช้ทรัพยากรของเครื่องจริงต่อเครื่องเสมือนในระบบ ให้ได้อย่างมีประสิทธิภาพพร้อมทั้งทำการตรวจสอบความผิดพลาดที่จะเกิดขึ้นกับอุปกรณ์ต่าง ๆ ของคอมพิวเตอร์จริงในอนาคตได้



ภาพประกอบ 3.1 ภาพรวมงานวิจัย

จากภาพประกอบ 3.1 เป็นภาพรวมงานวิจัยโดยแบ่ง การวิเคราะห์และประเมินประสิทธิภาพด้านการทำงานที่อาจเกิดความเสี่ยงใน Xen Hypervisor การวิเคราะห์และพัฒนาตัวแบบด้วย Fuzzy logic ในการบริหารจัดการ เครื่องเสมือนให้ทำงานได้อย่างมีประสิทธิภาพบน Xen Hypervisor เปรียบเทียบประสิทธิภาพและแนวทางบริหารจัดการเครื่องเสมือนบน Xen Hypervisor ระหว่างแนวทางใหม่และแนวทางเดิม

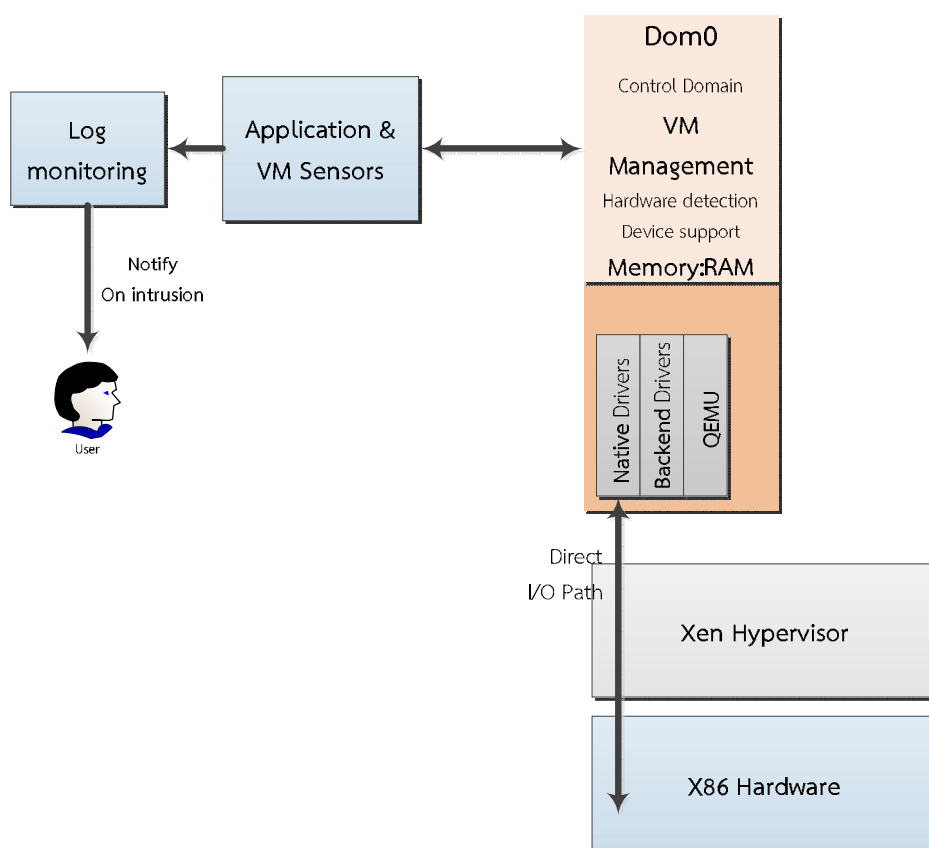


3.1 ศึกษาโปรแกรมเฝ้าระวังทรัพยากรบนเครื่องจริง บน Xen Hypervisor

เทคโนโลยีเสมือน ได้ถูกนำมาใช้งานเพิ่มมากขึ้นในองค์กรต่างๆ โดยเฉพาะการนำ Xen Hypervisor ซึ่งเป็นซอฟต์แวร์แบบเปิด แต่เนื่องจากยังมีข้อจำกัดในการจัดทรัพยากร เช่นการกำหนดทรัพยากรให้เครื่องเสมือนภายในใช้ทรัพยากร จากเครื่องจริงสูงเกินไปจน เครื่องจริงมีทรัพยากรไม่เพียงพอในการบริหารจัดการระบบตัวของตัวมันเอง ซึ่งจะนำไปสู่ความล้มเหลวของระบบต่อการให้บริการ และส่งผลไปยังเครื่องเสมือนต่างๆที่ทำงานบนระบบเครื่องจริงที่ทำงานอยู่ ดังนั้นในงานวิจัยจึงเห็นว่าการเฝ้าระวังทรัพยากรของเครื่องจริงใน Xen Hypervisor มีความสำคัญเพื่อให้สามารถวางแผนรับมือกับความเสี่ยงที่อาจจะเกิดขึ้นตามมา โดยมีการออกแบบการทดลองเพื่อประเมินประสิทธิภาพด้านการทำงานที่อาจเกิดความเสียหายใน Xen Hypervisor ดังนี้

3.1.1 เฝ้าระวังหน่วยความจำ

การทดสอบเป็นการบันทึกการใช้งานของหน่วยความจำสำรอง โดยพิจารณารายละเอียดของหน่วยความจำสำรองเช่น ขนาดหน่วยความจำสำรองทั้งหมด พื้นที่ที่ถูกใช้และพื้นที่ว่างของหน่วยความจำเพื่อการให้บริการได้ต่อเนื่องหากหน่วยความจำสำรองไม่เพียงพอต่อการให้บริการหรือมาการใช้งานที่ผิดปกติ ความสามารถในการให้บริการก็จะสูญเสียตามไปด้วย ดังนั้นการรายงานให้ผู้ดูแลระบบทราบก่อนที่ทรัพยากรของเครื่องจริงจะไม่สามารถให้บริการได้จะทำให้สามารถแก้ไขหรือปรับปรุงให้ระบบสามารถทำงานต่อไปได้



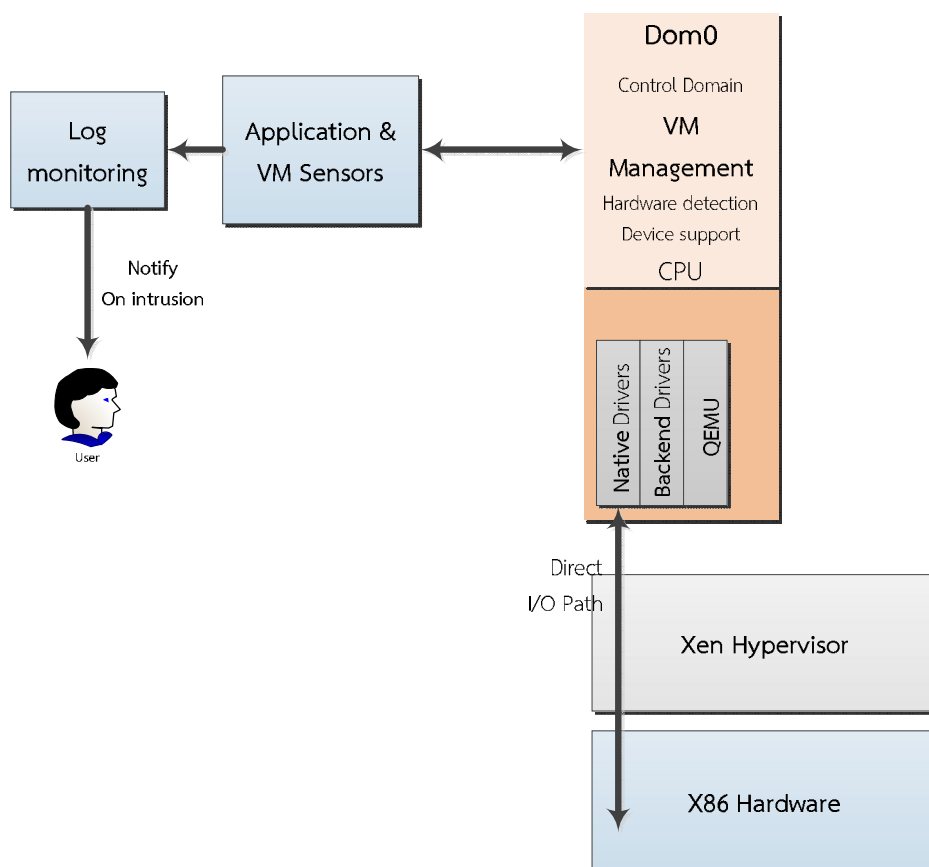
ภาพประกอบ 3.2 ขั้นตอนการทดสอบหน่วยความจำ (Memory)



จากภาพประกอบ 3.2 Application & VM Sensors จะทำการตรวจสอบเฟิร์มแวร์การทำงานของหน่วยความจำสำรอง โดยส่งคำร้องมายัง Dom0 ใน Xen ซึ่ง Dom0 ที่ทำหน้าที่ตรวจสอบการทำงานทางกายภาพทั้งหมดของระบบ และรายงานคำร้องนั้น ไปยัง ผู้ดูแลระบบ โดยมีเงื่อนไขบ่งบอกว่าเหตุการณ์ไม่ปกติในการใช้งานคือเท่ากับหรือมากกว่า 95 เปอร์เซ็นต์ ของพื้นที่ทั้งหมดจะส่งสารถึงผู้ดูแลระบบ

3.1.2 เฟิร์มแวร์การทำงานของหน่วยประมวลผลกลาง

การทดสอบหน่วยประมวลผลกลาง CPU โดยพิจารณารายละเอียดของเปอร์เซ็นต์ การใช้งาน เปอร์เซ็นต์ทั้งหมดที่ว่าง เพื่อการให้บริการได้ต่อเนื่องหากมีการใช้งานหน่วยประมวลผลที่สูงอย่างต่อเนื่องอาจส่งผลต่อการให้บริการหรืออาจเป็นการทำงานที่ผิดปกติ ความสามารถในการให้บริการก็จะสูญเสียตามไปด้วย ดังนั้นการรายงานให้ผู้ดูแลระบบทราบก่อนที่จะทรัพยากรก่อนของเครื่องจริงจะไม่สามารถให้บริการได้จะทำให้แก้ไขหรือปรับปรุงให้ระบบสามารถทำงานต่อไปได้



ภาพประกอบ 3.3 ขั้นตอนการทดสอบหน่วยประมวลผลกลาง (CPU)



จากภาพประกอบ 3.3 Application & VM Sensors จะทำการตรวจสอบฝ่ายะวังการทำงาน ของหน่วยความจำสำรอง โดยส่งคำร้องมายัง Dom0 ใน Xen ซึ่ง Dom0 ที่ทำหน้าที่ตรวจสอบการทำงานทางกายภาพทั้งหมดของระบบ และรายงานคำร้องนั้น ไปยัง ผู้ดูแลระบบ เงื่อนไขบ่งบอกว่า เหตุการณ์ไม่ปกติในการใช้งานคือเท่ากับหรือมากกว่า 95 เปอร์เซ็นต์ เป็นเวลาต่อเนื่อง 5 นาทีจะส่งสารถึงผู้ดูแลระบบ

3.2 วิเคราะห์และพัฒนาตัวแบบด้วย Fuzzy logic ในการบริหารจัดการทรัพยากรของเครื่องเสมือนบน Xen Hypervisors

การใช้งานเครื่องเสมือนในระบบของ Xen Hypervisors นั้น จะต้องมีการกำหนดเพื่อ ทรัพยากรเริ่มต้นของระบบขึ้นมาเพื่อให้เครื่องเสมือนนั้นๆ สามารถใช้ทรัพยากรในการเริ่มต้นการทำงาน ของระบบได้ แต่ในบางช่วงเวลาการทำงานการใช้งานทรัพยากรต่างๆในเครื่องเสมือนที่มีการจองไว้ใน ระบบกับไม่มีการใช้งานใดๆ เลย เช่นเครื่องเสมือนที่ให้บริการระบบทะเบียนของมหาวิทยาลัย ที่สภาพ การให้บริการปกติ จะมีการเข้าใช้งานจากผู้ให้บริการเพียงวันละ 200-300 ครั้งต่อวัน ในขณะที่ช่วง ลงทะเบียนเรียนในแต่ละภาคการศึกษา ซึ่ง 1 ปี จะมีเพียงช่วง 3 ช่วง ช่วงละประมาณ 15-20 วัน โดย การให้บริการช่วงนั้นจะมีผู้เข้าใช้บริการประมาณ 10,000-15,000 ครั้งต่อวัน ซึ่งถือว่าเป็นสัดส่วนที่สูง มาก ในการสร้างเครื่องเสมือนจึงจำเป็นต้องจองทรัพยากรให้เครื่องเสมือนนั้นจำนวนสูง เพียงเพื่อใ้ งานให้ตอบสนองการให้บริการของผู้ใช้งานเพียงร้อยละ 30 เท่านั้น ทำให้เสียโอกาสในการใช้ทรัพยากร ในการสร้างหรือจัดสรรให้เครื่องเสมือนอื่นๆ จากการศึกษาดังกล่าวจึงมีแนวคิดเพื่อออกแบบวิธีการ แก้ไขปัญหาดังกล่าว โดยจะแบ่งเป็นสองส่วนได้แก่ การวิเคราะห์ประสิทธิภาพการทำงานเครื่องเสมือนที่ ทำงานบน Xen Hypervisors และการพัฒนาตัวแบบโดยใช้ Fuzzy logic เข้ามาช่วยในการบริหารจัดการ ทรัพยากรในเครื่อง โดยมีการออกแบบการทดลองเพื่อประเมินประสิทธิภาพที่เกิดขึ้นบน Xen Hypervisor

3.2.1 เครื่องมือที่ใช้

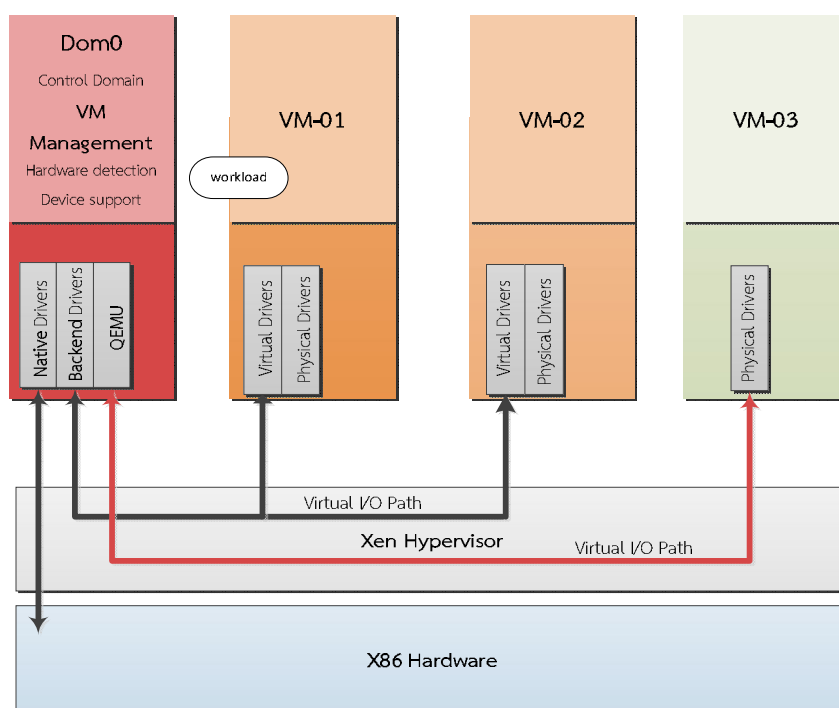
การทดสอบประสิทธิภาพเครื่องคอมพิวเตอร์ใช้เทคนิคที่เรียกว่าการสร้างงานจำลอง เครื่องมือที่เรียกว่า Webserver Stress Tools 7 สำหรับการทดสอบประสิทธิภาพโดยการทดสอบจะ ทดสอบกับเครื่องเสมือนใน Xen Server 6.2

การทดลองในเครื่องเสมือนที่ใช้ซอฟต์แวร์ Xen Server 6.2 ติดตั้งบนเครื่อง Intel Core 2 QuadCPU 2.80 GHz, 4096MB of MEMORY, 100/1000 Mbps Ethernet interface card แบ่งออกเป็นจำนวน 3 เครื่อง

ซอฟต์แวร์สำหรับติดตั้งในเครื่องเสมือนเพื่อใช้ให้บริการเว็บ เซิร์ฟเวอร์ประกอบไปด้วย Apache 2.2.8, PHP 5.2.6, MySQL 5.0.51b และ phpMyAdmin-2.10.3 โดยใช้ Joomla 2.5 เป็น ระบบให้บริการเว็บไซต์

3.2.2 สถานการณ์ทดลอง





ภาพประกอบ 3.4 ขั้นตอนการทดสอบหน่วยความจำหลักสถานการณ์การทดลอง

จากภาพประกอบ 3.4 เครื่องเสมือนทั้ง 3 เครื่อง (VM-01, VM-02, VM-03) ทำการติดตั้งระบบปฏิบัติการ Windows Server 2003 โดยจัดสรรทรัพยากรจากเครื่องจริงให้แก่เครื่องดังนี้ CPUs=1, Memory = 640 MB, DISK = 24GB รูปแบบโครงสร้างการทดสอบดังภาพประกอบ 3.4 โดยการทดสอบดังนี้

ทดสอบด้วยโปรแกรม Webserver Stress Tools กับเครื่องเสมือนที่ 1 (VM-01) ใช้รูปแบบการทดสอบแบบ CLICKS ซึ่งรูปแบบการทำงานประเภทนี้ จะทดสอบตามจำนวนที่ระบุโดยจะสร้างผู้ใช้งานจำลองขึ้นมา ซึ่งการทดสอบจะสิ้นสุดลงเมื่อผู้ใช้งานจำลองได้สร้างภาระงานตามจำนวนคลิกที่ระบุไว้ครบ โดยในการทดสอบแต่ละครั้งจะทำการเพิ่มจำนวน ผู้ใช้งานขึ้นที่รอบการทดสอบจนพบจุดที่เครื่องเสมือนไม่สามารถให้บริการเว็บเซิร์ฟเวอร์ ได้ตามจำนวนผู้ใช้งานจำลองที่ร้องขอมา

3.2.3 วิเคราะห์ผลการทดลอง

การทดลองโปรแกรม Webserver Stress Tools โดยการสร้างภาระงานไปยังเครื่องเสมือน VM-01 จนเครื่องเสมือนดังกล่าวปฏิเสธการให้บริการ (DoS : Denial of Service) โดยประเมินประสิทธิภาพของเครื่องเสมือน ด้วยตัวชี้วัด ดังนี้

1) การใช้งานหน่วยประมวลผลกลาง (CPU Utilization) หลังจากเครื่องเสมือน VM-01 ได้รับภาระงานแล้วจะทำการประมวลผลภาระงานนั้นๆ โดยการเรียกข้อมูลเว็บเพจขึ้นมาแสดง มีการใช้งานหน่วยประมวลผลกลาง ก็เปอร์เซ็นต์ ณ ขณะเวลานั้น



2) การใช้งานหน่วยความจำ (Memory Usage) หลังจากเครื่องเสมือน VM-01 ได้รับภาระงานแล้วจะทำการประมวลผลภาระงานนั้นๆ โดยการเรียกข้อมูลเว็บเพจขึ้นมาแสดง มีการใช้งานหน่วยความจำ ก็เปอร์เซ็นต์ ณ ขณะเวลานั้น

3) การใช้งานของผู้ใช้อื่น ในเครื่องเสมือน VM-02 และ VM-03 ในระหว่างที่มีการทดลอง ด้วยโปรแกรม Webserver Stress Tools กับเครื่อง VM-01 จะทำการเข้าใช้งานจากผู้ใช้อื่นเพื่อตรวจสอบการถูกปฏิเสธการให้บริการหรือไม่

ตาราง 3.1 การจำแนก Workload

Content	Workload	CPU	Memory usage
VM01	50 user	100 %	90%
VM02	2 user	0.5-1%	50 %
VM03	2 user	0.5-1%	50 %

จากตาราง 3.1 การทดสอบโดยการใช้โปรแกรม Webserver Stress Tools ทำการสร้างภาระงานขึ้นในเครื่องเสมือน VM-01 โดยทำการทดสอบ ส่งภาระงานเข้าไปในระบบจำนวน 50 user พบว่า ผลการทดสอบจึงมีโอกาสมากกว่า 100 % (CPU=100%) หน่วยความจำสำรองถูกใช้ไปทั้งหมด 90 % ในขณะที่เครื่อง เสมือน VM-02 และ VM-03 ยังเหลือทรัพยากรอีกมาก โดยยังไม่มีการใช้งานใดๆ

3.2.4 ออกแบบแนวทางการแก้ปัญหาการใช้จัดการทรัพยากรของเครื่องเสมือนด้วย Fuzzy Logic

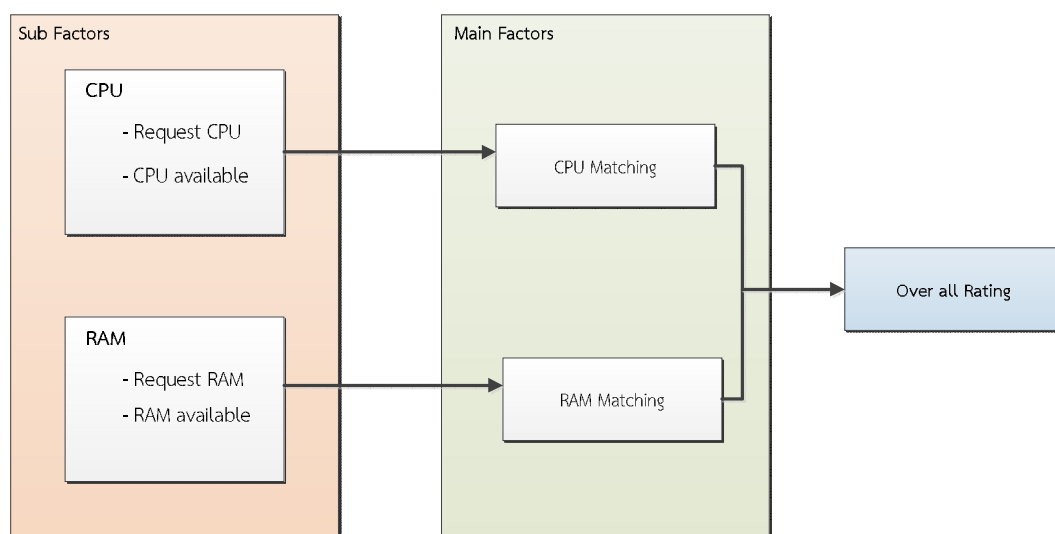
การออกแบบแนวทางการพัฒนาบริหารจัดการทรัพยากรของเครื่องเสมือนด้วย Fuzzy Logic Rule-Based System มีรายละเอียดดังนี้

1) แนวคิดการออกแบบ

โดยปกติการเริ่มต้นการสร้างเครื่องเสมือนในระบบปฏิบัติการต้องมีการจองทรัพยากรจากระบบปฏิบัติการ Xen Server 6.2 ก่อน จึงจะสามารถเริ่มต้นการทำงานของระบบได้ โดยในงานวิจัยนี้จะนำเสนอการปรับปรุงประสิทธิภาพในการบริหารจัดการทรัพยากรโดยการ Fuzzy logic โดยจะแบ่งการทำงานออกเป็น 3 ส่วนคือ ส่วนตัวแปรอินพุต ส่วนการนิยามพจน์ภาษา และส่วนการประมวลผล Fuzzy ของระบบ

1.1) ตัวแปรอินพุตในการใช้ปรับปรุงประสิทธิภาพทรัพยากรของเครื่องเสมือนใน Xen Server 6.2 มีตัวแปรที่สำคัญคือ หน่วยประมวลผล (CPU) และหน่วยความจำ (Memory) ดังภาพประกอบ 3.5





ภาพประกอบ 3.5 ตัวแปรอินพุต

1.2) การนิยามพจน์ภาษาเป็นการแปลงอินพุตที่เป็นค่าจริงเปลี่ยนเป็นอินพุตแบบ Fuzzy โดยสามารถสร้างฟังก์ชันความเป็นสมาชิกจากการทำงานให้อยู่ในรูปของพจน์ภาษา

การนิยามพจน์ภาษาของหน่วยประมวลผลกลาง (CPU) และ พจน์ภาษาของหน่วยความจำ (Memory) ได้ใช้นิยามเดียวกับการอ้างอิงจากโปรแกรม Reliability and Performance Monitor Microsoft Corporation [36] ซึ่งเป็นโปรแกรมมาตรฐานที่ถูกติดตั้งมาพร้อมเพื่อใช้ตรวจสอบการทำงานของระบบตั้งแต่ Windows Server 2008 - 2012 โดยได้กำหนดภาระงานต่ำ (Low) หมายถึง ภาระงานของ CPU ต่ำกว่า 50 % ภาระงานปานกลาง (Medium) หมายถึง ภาระงานของ CPU ตั้งแต่ 25% ถึง 75% และ ภาระงานสูง (High) หมายถึงภาระงานของ CPU ตั้งแต่ 50 % ขึ้นไป

1.3) การประมวลผล Fuzzy ของระบบ ในการหาค่าภาระงานของ CPU และ Memory ได้ทำการกำหนดให้ค่าความเป็นสมาชิกขึ้นอยู่กับค่าที่ได้รับภาระงานจากระบบ (x) โดยในงานวิจัยนี้เลือกใช้ ฟังก์ชันของความเป็นสมาชิกแบบ สามเหลี่ยม (Triangular Membership Function) เนื่องจากตรงกับค่าตัวแปร 3 ตัวที่ได้กำหนดไว้ คือค่า Low, Medium, High และเป็นฟังก์ชันที่นิยมใช้เกี่ยวกับการควบคุมที่ต้องการความรวดเร็วทันเวลา เนื่องจากการคำนวณน้อย ซึ่งมีพารามิเตอร์ ดังสมการที่ 1

$$triangular(x; a, b, c) = f(x) = \begin{cases} 0, & x < a \\ (x - a)/(b - a), & a \leq x < b \\ (c - x)/(c - b), & b \leq x \leq c \\ 0, & x > c \end{cases} \quad (1)$$



โดยกำหนดให้

ให้	x	แทน	ตัวแปรอินพุต
	Low	แทน	ฟังก์ชันความเป็นสมาชิกของเซตภาระงานระดับต่ำ
	Medium	แทน	ฟังก์ชันความเป็นสมาชิกของเซตภาระงานระดับปานกลาง
	High	แทน	ฟังก์ชันความเป็นสมาชิกของเซตภาระงานระดับสูง

ในการกำหนดฟังก์ชันความเป็นสมาชิกของเซต ทั้ง 3 ซึ่งมีรายละเอียดดังนี้

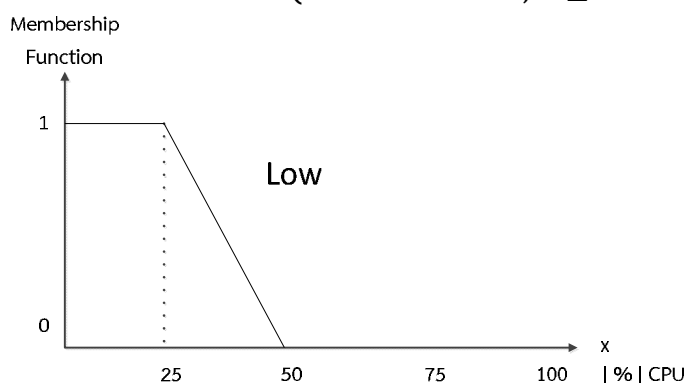
สมการที่ใช้ตรวจสอบเงื่อนไข ภาระงานระดับต่ำ (Low)

ถ้า $x < 25$ แล้ว 1 (True)

ถ้า $x \geq 25$ และ $x < 50$ แล้ว $(50 - x) / 25$

ถ้า $x \geq 50$ แล้ว 0 (False)

$$\text{ฟังก์ชัน } Low(x: 25, 50, 75) \begin{cases} 1 & , x < 25 \\ (50 - x) / 25 & , 25 \leq x < 50 \\ 0 & , x \geq 50 \end{cases}$$



ภาพประกอบ 3.6 แสดงการตรวจสอบเงื่อนไข ภาระงานต่ำ

สมการที่ใช้ตรวจสอบเงื่อนไข ภาระงานระดับปานกลาง (Medium)

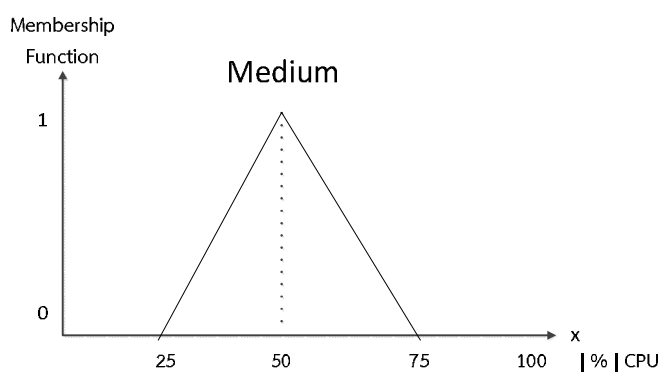
ถ้า $x < 25$ หรือ $x > 75$ แล้ว 0 (FALSE)

ถ้า $x \geq 25$ และ $x < 50$ แล้ว $(x - 25) / 25$

ถ้า $x \geq 50$ และ $x \leq 75$ แล้ว $(75 - x) / 50$

$$\text{ฟังก์ชัน } Medium(x: 25, 50, 75) \begin{cases} 0 & , x < 25, x > 75 \\ (x - 25) / 25 & , 25 \leq x < 50 \\ (75 - x) / 25 & , 50 \leq x \leq 75 \end{cases}$$





ภาพประกอบ 3.7 แสดงการตรวจสอบเงื่อนไข ภาระงานปานกลาง

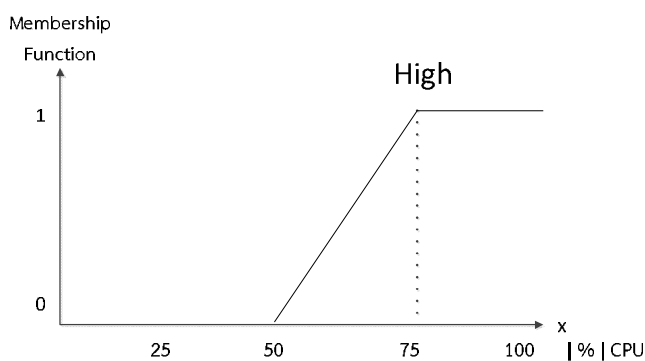
สมการที่ใช้ตรวจสอบเงื่อนไข ภาระงานระดับสูง (High)

ถ้า $x < 50$ แล้ว 0 (FALSE)

ถ้า $x \geq 50$ และ $x < 75$ แล้ว $(x - 50) / 25$

ถ้า $x \geq 75$ แล้ว 1 (True)

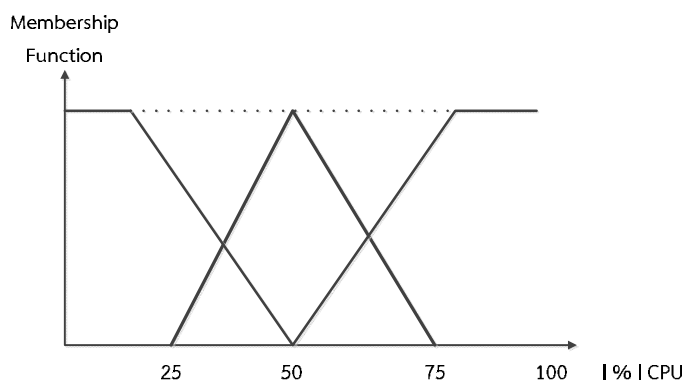
$$\text{ฟังก์ชัน High } (x: 25, 50, 75) \begin{cases} 0 & , x < 50 \\ (x - 50) / 25 & , 50 \leq x < 75 \\ 1 & , x \geq 75 \end{cases}$$



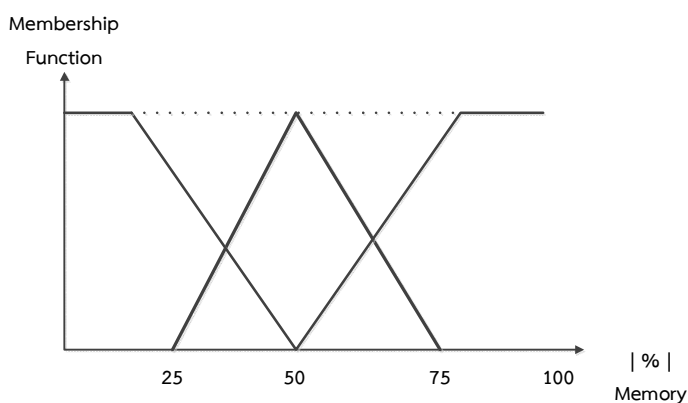
ภาพประกอบ 3.8 แสดงการตรวจสอบเงื่อนไข ภาระงานสูง

เมื่อนำฟังก์ชันความเป็นสมาชิก 3 เซต มารวมกันภายใต้แกนเดียวกันจะได้ ภาพประกอบ 3.9





ภาพประกอบ 3.9 แสดงการตรวจสอบเงื่อนไขในทฤษฎี ของ CPU



ภาพประกอบ 3.10 แสดงการตรวจสอบเงื่อนไขในทฤษฎี ของ Memory

2) ระยะเวลาในการส่งคำสั่งเพื่อร้องขอการจัดสรรทรัพยากร

เมื่อได้ผลลัพธ์จาก ฟัชชีแล้ว โดยจากเงื่อนไขได้แบ่งภาระงานออกเป็น 3 ระดับคือ

Low ภาระงานระดับต่ำ

Medium ภาระงานระดับปานกลาง

High ภาระงานระดับสูง

ถ้า ภาระงาน Low จะทำการตรวจสอบว่า เครื่องเสมือนมีการเปลี่ยนแปลงทรัพยากรก่อนหน้าหรือไม่ถ้ามี จะทำการคืนค่าทรัพยากรที่ได้รับการจัดสรรให้เป็นค่าเริ่มต้น

ถ้า ภาระงาน Medium ไม่มีการตรวจสอบ ทำงาน

ถ้า ภาระงานสูง High จะทำการส่งคำขอเพื่อขอรับการจัดสรรทรัพยากรจากรบบโดย CPU จะจัดสรรเป็น Core ที่ ละ 1 ส่วน Memory จะจัดสรรทีละ 64 MB

3) การจัดสรรทรัพยากร

เมื่อได้รับคำร้องขอจากผลการวิเคราะห์ด้วยฟัชชีและมีเวลาการทำงานที่เป็นไปตามเงื่อนไขโมดูล Resource Predictor จะทำการส่งคำขอมายัง โมดูล Resource Allocator เพื่อทำการตรวจสอบทรัพยากรเครื่องเสมือนต่างๆและระบบเครื่องจริงว่ามีทรัพยากรเพียงพอ ต่อคำร้องขอ



หรือไม่ ถ้าหากมีเพียงพอ จะทำการจัดสรรทรัพยากรตามคำร้องขอ โดยการกำหนดค่าทรัพยากรที่ต้องการไปยัง Kernel ของ Dom0 Control Domain แต่ถ้าหากไม่มีทรัพยากรเพียงพอจะทำยกเลิกคำขอ และให้เครื่องเสมือนนั้นส่งคำขอมายังรอบถัดไป

3.3 เปรียบเทียบประสิทธิภาพการแนวทางการบริหารจัดการทรัพยากรเครื่องเสมือนบน Xen แนวทางใหม่และแนวทางเดิม

การวิเคราะห์การทดสอบประสิทธิภาพและประสิทธิผลของโปรแกรมต้นแบบดังนี้

3.3.1 การทดสอบประสิทธิภาพและประสิทธิผลของการบริหารจัดการทรัพยากรเครื่องเสมือน

เกณฑ์การทดสอบประสิทธิภาพและประสิทธิผลของการบริหารจัดการทรัพยากรเครื่องเสมือนมี 3 วิธีด้วยกัน ได้แก่ Analytical Modeling, Simulation และ Measurement

1) Analytical Modeling เป็นการใช้โมเดลทางคณิตศาสตร์เพื่อออกแบบในการแก้ปัญหา นั้น ๆ แต่วิธีการนี้ใช้ไม่ได้ผลกับระบบที่มีความซับซ้อนมาก ซึ่งจะใช้วิเคราะห์ได้ผลในปัญหาเล็ก ๆ

2) Simulations เป็นการใช้โปรแกรมจำลองเพื่อวิเคราะห์ปัญหา ซึ่งจะได้ผลกับระบบที่ซับซ้อนดีกว่าการวิเคราะห์แบบ Analytical Modeling แต่วิธีการนี้ยังมีข้อเสียเพราะเป็นการจำลอง ซึ่งวัดจากความน่าจะเป็น ดังนั้นจึงไม่เห็นถึงสภาพจริง

3) Measurement เป็นวิธีการวัดค่าจริงโดยแบ่งเป็น 2 วิธีย่อยได้แก่

3.1) Real System เป็นการวัดค่าจริงจากระบบจริง แต่ค่าที่ได้จากวิธีนี้อาจจะไม่สามารถนำมาใช้งานได้ เพราะเป็นการวัดจากระบบจริง ซึ่งไม่ถูกควบคุมค่าพารามิเตอร์ต่าง ๆ ที่มีผลต่อค่าที่วัดได้ อีกทั้งยังมีความเสี่ยงที่จะส่งผลต่อระบบจริง แต่วิธีการนี้มีข้อดีคือได้เห็นถึงสภาพการใช้งานจริง ซึ่งส่วนใหญ่จะใช้เป็นขั้นตอนสุดท้ายหลังจากที่ผ่านการวิเคราะห์ด้วยวิธีอื่น ๆ มาแล้ว

3.2) Test-beds เป็นการสร้างสภาวะภาระงานจำลองเพื่อทดสอบที่มีลักษณะคล้ายกับภาระงานจริงแต่มีขนาดเล็กกว่า สามารถควบคุมพารามิเตอร์ต่าง ๆ ได้ สามารถวัดประสิทธิภาพและประสิทธิผลได้ดีกว่า

ในงานวิจัยนี้ได้เลือกวิธีทดสอบประสิทธิภาพด้วยวิธี Test-beds ทดสอบ 30 ครั้ง เพราะการวัดประสิทธิภาพของภาระงานจะเห็นผลได้ดีกว่า

3.3.2 ตัวชี้วัดประสิทธิภาพและประสิทธิผลของการใช้ฟิซซีโลจิสต์กับภาระงาน CPU

การทดสอบผลสัมฤทธิ์ในการบริหารจัดการทรัพยากรตามการออกแบบการทดสอบผลสัมฤทธิ์ในการบริหารจัดการทรัพยากรตามการออกแบบโดยการ ทดลองสร้างภาระงานระหว่างเครื่องเสมือนที่ไม่มีการใช้ ฟิซซีโลจิสต์กับเครื่องเสมือนที่มีการใช้ฟิซซีโลจิสต์ ในงานวิจัยนี้ในการทดสอบการสร้างภาระงานให้กับ CPU โดยการใช้โปรแกรมจำลองภาระงานของ CPU

3.3.3 ตัวชี้วัดประสิทธิภาพและประสิทธิผลของการใช้ ฟิซซีโลจิสต์กับภาระงาน Memory

การทดสอบผลสัมฤทธิ์ในการบริหารจัดการทรัพยากรตามการออกแบบการทดสอบผลสัมฤทธิ์ในการบริหารจัดการทรัพยากรตามการออกแบบโดยการ ทดลองสร้างภาระงานระหว่าง



เครื่องเสมือนที่ไม่มีการใช้ ฟัชชีโลจิสต์กับเครื่องเสมือนที่มีการใช้ฟัชชีโลจิสต์ ในงานวิจัยนี้ในการทดสอบการสร้างภาระงานให้กับ Memory โดยการใช้โปรแกรมจำลองภาระงานของ Memory

3.3.4 การวัดประสิทธิภาพ การทำงานของตัวแบบ ฟัชชีลอจิก ด้วย Big-O Notation

การทดสอบประสิทธิภาพการทำงานของตัวแบบฟัชชีลอจิกตามการออกแบบโดยการวัดประสิทธิภาพของ อัลกอริทึม โดยใช้ Big-O Notation โดยผลการทดสอบจะต้องทำงานไม่ด้อยไปกว่า Big O ของอัลกอริทึม ที่ออกแบบไว้



บทที่ 4

ผลการวิจัย

ผลการดำเนินงานแบ่งออกเป็น 3 ส่วนคือ ผลการวิเคราะห์ประสิทธิภาพสภาวะภาระงานของระบบปฏิบัติการเซ็น เสนอแนวทาง Fuzzy Logic พัฒนาตัวแบบในระบบปฏิบัติการเสมือน และทดสอบประสิทธิภาพและประสิทธิผลของตัวแบบที่พัฒนา

4.1 ผลการวิเคราะห์ประสิทธิภาพสภาวะภาระงานของระบบปฏิบัติการเซ็น

ผลการศึกษาโปรแกรมเฝ้าระวังทรัพยากรเครื่องจริงบน Xen Hypervisor ในบทที่ 3 สามารถแบ่งการเฝ้าระวัง ออกเป็น 2 อย่างคือ เฝ้าระวังหน่วยความจำสำรอง และ เฝ้าระวังการทำงานของหน่วยประมวลผลกลาง แสดงผลการทดสอบ ดังตาราง 4.1

ตาราง 4.1 ผลการศึกษาโปรแกรมเฝ้าระวังทรัพยากรเครื่องจริงบน Xen Hypervisor

รูปแบบการทดสอบ	ไม่มีการเฝ้าระวัง	มีการเฝ้าระวัง
เฝ้าระวังหน่วยความจำสำรอง เฝ้าระวังการทำงานของหน่วย ประมวลผลกลาง	ไม่สามารถทราบถึงขีดจำกัดใน การให้บริการได้ซึ่งทำให้เกิด	สามารถทราบได้ถึงการแข่งขัน และสถานการณ์ภาระงาน ปัจจุบันที่เครื่องทำอยู่โดยใช้ ข้อมูลดังกล่าวเพื่อจัดหา ทรัพยากรให้เพียงพอต่องาน

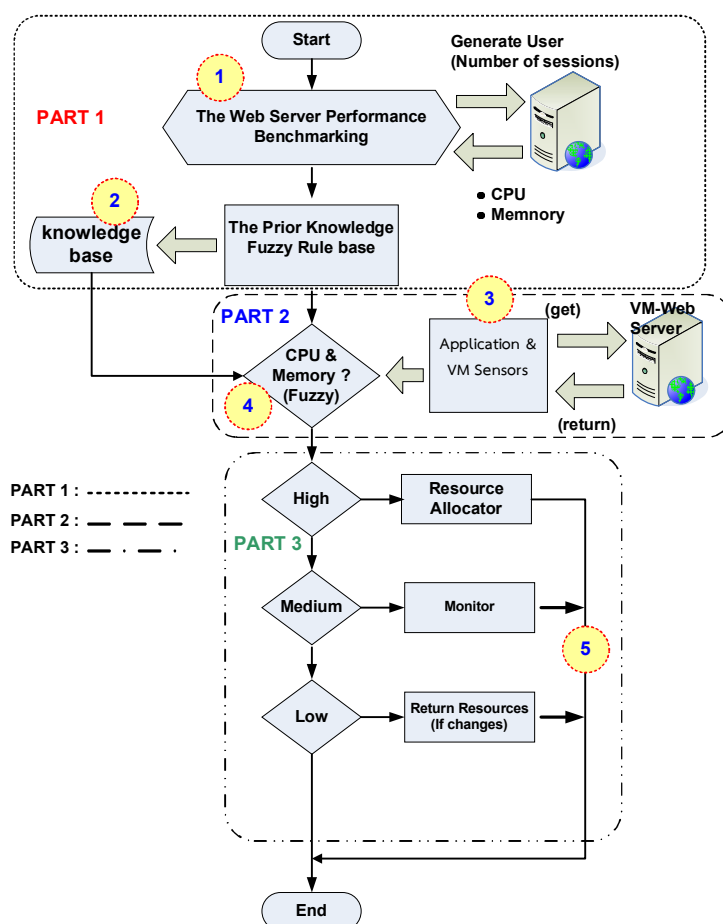
จากตาราง 4.1 สรุปผลที่ได้จากการทดสอบโดยให้มีการเฝ้าระวังทรัพยากรเครื่องจริงบน Xen Hypervisor การเฝ้าระวังทรัพยากรของเครื่องจริงใน Xen Hypervisor มีความสำคัญเพื่อให้สามารถวางแผนรับมือกับความเสี่ยงที่อาจจะเกิดขึ้นตามมาและสามารถทำให้ผู้ดูแลระบบติดตามการทำงานได้อย่างใกล้ชิดมากยิ่งขึ้นเมื่อมีการแจ้งเตือนจากระบบ

4.2 เสนอแนวทางในการบริหารจัดการทรัพยากรด้วย Fuzzy Logic ในการพัฒนาตัวแบบ

การออกแบบและพัฒนาแนวพัฒนาตัวแบบด้วย Fuzzy Logic ในการบริหารจัดการเครื่องเสมือนใน Xen Hypervisor

4.2.1 การออกแบบตัวแบบ Fuzzy Logic ในการบริหารจัดการเครื่องเสมือนใน Xen Hypervisor





ภาพประกอบ 4.1 การออกแบบตัวแบบ Fuzzy Logic ในการบริหารจัดการเครื่องเสมือนใน Xen

จากภาพประกอบ 4.1 เป็นตัวแบบ Fuzzy Logic ในการบริหารจัดการเครื่องเสมือนใน Xen เพื่อใช้ทำนายถึงสภาวะภาระงานต่างๆ ในการบริหารจัดการ การใช้ทรัพยากรของเครื่องจริงต่อเครื่องเสมือนในระบบ เช่น เซิร์ฟเวอร์ ให้ได้อย่างมีประสิทธิภาพ โดยแบ่งออกเป็น 3 ส่วน 5 ขั้นตอน

ส่วนที่ 1 : ทำหน้าที่เก็บข้อมูลการทำงานของเครื่องเสมือนในการทำงานในสภาวะงานและทรัพยากรที่ได้รับจัดสรรในช่วงต่างๆ ที่มีผลกระทบโดยตรงต่อประสิทธิภาพโดยรวมของเครื่อง ประกอบด้วย ซีพียู หน่วยความจำหลัก และระยะเวลาในการทำงาน โดยให้เรียกหน้า Index ของ Joomla เพื่อใช้เป็นฐานกฎเกณฑ์ฟัซซี (Fuzzy Rule Base) หรือ ฐานความรู้ (Knowledge Base) สำหรับป้อนเข้าสู่กระบวนการวิเคราะห์ ในงานวิจัยนี้ได้ใช้ ฟัซซี ในการวิเคราะห์หาสถานการณ์ของเครื่องเสมือนที่ให้บริการ ผู้วิจัยจึงต้องออกแบบและเก็บรวบรวมข้อมูลโดยอาศัยเครื่องมือที่เรียกว่า Web Stress Tool ในการวัดประสิทธิภาพของระบบในรูปแบบการทำงานในทรัพยากรลักษณะต่างๆ ซึ่งสามารถแสดงขั้นตอนได้ดังนี้

ขั้นตอนที่ 1: ทำการติดตั้งโปรแกรม Web Stress Tool ไว้บนเครื่องที่จะทำการเก็บข้อมูล โดยทดสอบไปยังเครื่องเสมือนที่ให้บริการเป็นเว็บเซิร์ฟเวอร์ เพื่อวัดประสิทธิภาพ จากนั้นทำการสร้างภาระงานให้กับเครื่องให้บริการ (User) ที่ได้รับจัดสรรทรัพยากร ในรูปแบบต่างๆ ที่ได้ออกแบบไว้ ส่วนปริมาณภาระงานในการทดสอบนั้น แต่ละรอบการทดสอบจะทำการทดสอบจนกระทั่งพบ

ข้อผิดพลาดในการให้บริการในเครื่องเสมือน โดยซีพียูมีหน่วยเป็นเปอร์เซ็นต์ (%) หน่วยความจำที่ใช้
งาน มีหน่วยเป็นเปอร์เซ็นต์ (%) และ ระยะเวลาการทำงานได้โดยไม่พบข้อผิดพลาดมีหน่วยเป็นวินาที
(Second) ดังตาราง 4.2

ตาราง 4.2 ผลการทดสอบและวัดประสิทธิภาพด้วยโปรแกรม Web Stress Tool

User	Memory (MB)	CPU 1 Core			CPU 2 Core			CPU 3 Core			CPU 4 Core		
		CPU (%)	Mem (%)	Time (S)	CPU (%)	Mem (%)	Time (S)	CPU (%)	Mem (%)	Time (S)	CPU (%)	Mem (%)	Time (S)
1	640	50±1.2	58±0.9	6±0.8	5±1.1	59±0.8	47±1.5	45±1.1	58±1.7	3±1.4	14±0.9	63±1.3	64±1.7
2		52±1.4	60±1.5	6±1.3	5±0.8	62±1.2	51±1.4	48±0.9	61±0.6	3±1.6	14±1.7	65±1.4	67±1.5
3		62±0.9	60±1.7	8±1.4	5±1.4	63±1.7	63±0.7	51±1.4	62±0.9	4±0.8	18±1.4	66±0.8	67±1.4
4		65±1.7	61±1.4	9±0.7	5±0.9	63±0.5	64±1.2	55±1.5	64±1.4	4±0.9	22±1.1	67±1.1	67±0.9
5	640	77±0.6	60±0.8	11±0.4	5±0.9	66±1.6	65±0.9	57±1.2	66±1.7	4±0.3	5±0.6	57±0.5	67±0.7
10		91±0.9	68±0.8	10±0.7	10±0.7	66±1.7	67±0.8	57±0.9	70±1.3	7±0.4	10±0.7	58±0.8	69±0.9
15		91±0.8	64±1.6	20±0.7	15±0.7	74±1.2	69±1.1	68±1.4	72±1.7	10±0.5	15±0.7	65±1.0	71±0.9
20		95±1.2	70±1.1	30±1.0	20±0.7	79±1.4	72±1.5	72±1.2	75±1.0	10±0.7	20±1.2	69±0.7	76±0.8
25		95±1.4	69±1.8	31±0.9	25±0.9	81±1.2	79±1.7	77±1.1	82±0.6	10±0.7	25±1.0	72±0.7	83±0.7
30		94±0.9	72±1.7	42±0.7	30±1.5	83±1.1	83±0.6	66±1.2	87±2.1	21±0.7	30±1.0	79±1.0	74±1.5
35		96±0.5	68±1.9	50±1.2	35±1.4	86±1.2	85±0.7	79±0.9	82±0.6	10±0.5	35±1.2	81±1.5	79±0.7
40		95±0.6	77±0.4	60±0.7	40±1.6	88±1.3	81±1.3	83±1.5	79±1.3	21±0.9	40±1.3	82±1.0	79±0.7
45		96±0.3	73±0.5	62±1.4	45±1.5	89±1.3	77±0.7	84±1.2	76±1.0	20±0.5	45±1.2	77±1.2	77±1.2
50	704	99±0.5	68±0.3	70±0.9	50±1.2	90±1.4	78±1.0	78±1.8	78±1.3	30±0.5	50±1.3	73±0.9	79±1.5
55					85±1.2	77±1.3	40±0.7	73±1.9	87±0.7	40±0.6	55±1.0	64±0.8	71±0.8
60					60±1.2	93±1.0	79±1.7	67±0.7	81±0.8	40±0.6	68±0.9	74±0.8	21±1.0
65					65±1.0	89±0.7	72±0.8	76±0.7	80±0.5	42±0.8	83±0.8	76±1.3	30±0.5
70	768				95±0.7	83±1.5	50±1.9	83±0.7	82±0.5	40±0.4	74±0.3	76±0.8	30±0.5
75					96±1.2	76±0.9	60±1.3	88±0.7	79±0.5	40±0.6	83±0.8	76±1.1	30±0.5
80	832				96±1.0	84±1.6	60±1.4	88±0.8	79±1.1	40±0.4	88±1.0	78±1.0	30±0.5
85								88±1.0	79±2.0	50±1.9	85±0.3	78±1.1	40±0.4
90								88±1.5	79±1.4	50±1.8	88±1.5	79±0.8	40±0.6
95								91±1.1	81±1.1	50±1.0	85±1.2	77±1.5	40±0.4
100								91±1.2	81±1.2	50±0.6	88±0.9	78±1.0	40±0.4
110								92±1.1	82±1.1	60±0.8	87±0.9	80±1.5	50±0.9
120								95±0.9	84±0.4	60±1.3	89±1.2	84±1.5	50±0.9
130								94±0.8	83±1.0	70±0.7	89±0.7	70±1.2	60±0.8
140								95±0.8	84±0.4	70±0.8	91±0.3	84±0.9	60±1.3
150								94±0.5	83±0.5	70±0.9	91±0.8	83±0.7	70±0.8
160	896							95±0.4	85±0.4	70±0.8	90±1.3	83±1.0	70±0.9
170											92±0.9	83±0.7	80±1.5
180											90±1.5	80±1.5	80±1.5
190											93±0.3	85±1.9	90±1.7
195	960										94±1.4	84±1.3	90±1.7
200	1024										92±1.4	78±0.7	90±0.7

จากตาราง 4.2 ผลการทดสอบและวัดประสิทธิภาพด้วยโปรแกรม Web Stress tool โดยการจำลองสภาวะงานที่จะทำการทดสอบไปยังเครื่องเสมือน ซึ่งกำหนดงานเป็นจำนวน User ที่ทำการทดสอบในทรัพยากรที่มี Memory ขนาด 640, 704, 768, 832, 896, 960, 1024 MB และ Core CPU จำนวน 1,2,3,4 ตามลำดับ การทดสอบ 30 ครั้ง



ผลการทดสอบการทำงานของภาระงานที่มีต่อการใช้งานทรัพยากรของเครื่องและระยะในการใช้งาน โดยเฉลี่ย ($\bar{x}$) ระดับความเชื่อมั่น 99 % ซึ่งสามารถยกตัวอย่างการคำนวณหาค่าความเชื่อมั่น ในการทดสอบ ที่ User = 5 , Memory (MB) = 640 , CPU = 1 core โดยได้ผลการทดสอบ ดังนี้ $[x_1, x_2, \dots, x_{30}] = [78, 77, 76, 80, 75, 78, 77, 76, 78, 77, 76, 78, 77, 76, 76, 80, 75, 78, 74, 75, 78, 78, 76, 78, 77, 76, 80, 75, 79, 75]$

จำหน่วยครั้งในการทดสอบ (n) = 30 มีค่าเฉลี่ยในการทดสอบ ($\bar{x}$) = 76.96 ส่วนเบี่ยงเบนมาตรฐาน (σ) = 1.60781 และ ระดับความเชื่อมั่น 99% (99% Confidence interval) = 2.58 โดยการประมาณค่าเฉลี่ยของประชากรกลุ่มเดียวแบบช่วง ที่ระดับความเชื่อมั่น 99% โดยใช้สมการที่ 1 หาค่าประมาณค่าเฉลี่ยกลุ่มประชากรที่ระดับความเชื่อมั่น 99%

$$\bar{x} - z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} < \mu < \bar{x} + z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \quad (1)$$

แทนค่าลงในสมการที่ 1 $76.96 - 2.58 \frac{1.60781}{\sqrt{30}} < \mu < 76.96 + 2.58 \frac{1.60781}{\sqrt{30}}$ ที่ระดับความเชื่อมั่นที่ 99% ของการทดสอบการทำงานของ CPU ที่ 5 User ในทรัพยากรที่มีขนาด Memory (MB) = 640 , CPU = 1 core การทำงานเฉลี่ยของ CPU มีค่าอยู่ระหว่าง 76.2106 – 77.7228 เปอร์เซนต์

ขั้นตอนที่ 2 : จัดเก็บข้อมูลที่ได้ให้อยู่ในรูปแบบที่สามารถประมวลผลได้ โดยข้อมูลที่ได้อาจเป็นข้อมูลเพื่อใช้เป็นฐานกฎเกณฑ์ฟัซซี หรือ ฐานความรู้ (Knowledge Base) ในการดำเนินการทางฟัซซี โดยให้ ฐานความรู้ (Knowledge Base) ซึ่งคือข้อมูลที่นำค่า มาจากตาราง 4.1 ในจุดที่เปลี่ยนแปลงค่า ซิฟิยู และหน่วยความจำแล้วทำให้สามารถรองรับจำนวนงานได้เพิ่มขึ้น มีค่าดังในตาราง 4.3

ตาราง 4.3 ฐานความรู้ (Knowledge Base)

Memory (MB)	CPU 1 Core			CPU 2 Core			CPU 3 Core			CPU 4 Core		
	CPU %	Mem %	Time	CPU %	Mem %	Time	CPU %	Mem %	Time	CPU %	Mem %	Time
640	96 ± 0.3	73 ± 0.5	62 ± 1.4	85 ± 1.2	77 ± 1.3	40 ± 0.7	73 ± 1.9	87 ± 0.7	40 ± 0.6	50 ± 1.3	73 ± 0.9	79 ± 1.5
704	99 ± 0.5	68 ± 0.3	70 ± 0.9	65 ± 1.0	89 ± 0.7	72 ± 0.8	76 ± 0.7	80 ± 0.5	42 ± 0.8	55 ± 1.0	64 ± 0.8	71 ± 0.8
768				96 ± 1.2	76 ± 0.9	60 ± 1.3	95 ± 0.8	84 ± 0.4	70 ± 0.8	68 ± 0.9	74 ± 0.8	21 ± 1.0
832				96 ± 1.0	84 ± 1.6	60 ± 1.4	94 ± 0.5	83 ± 0.5	70 ± 0.9	90 ± 1.5	80 ± 1.5	80 ± 1.5
896							95 ± 0.4	85 ± 0.4	70 ± 0.8	93 ± 0.3	85 ± 1.9	90 ± 1.7
960										94 ± 1.4	84 ± 1.3	90 ± 1.7
1024										92 ± 1.4	78 ± 0.7	90 ± 0.7



จากตาราง 4.3 ฐานความรู้ สามารถอธิบายได้ว่าจุดเปลี่ยนแปลงในของค่าซีพียู และ หน่วยความจำที่สามารถรองรับ จำนวนการให้บริการของจำหน่วย User เช่น ที่การทำงาน CPU 1 Core สามารถทำงานได้สูงสุด กับ Memory ขนาด 704MB โดย CPU Usage $99 \pm 0.5\%$, Memory Usage 68.03% ทำงานต่อเนื่องเป็นเวลา 70 ± 0.9 วินาที หากต้องการรองรับจำนวนงาน (User) ที่สูงขึ้นต้องเพิ่มจำนวน Core CPU อีก หรือ ในการทำงาน CPU 2 Core ทำงานใน Memory ขนาด 704 MB เมื่อ CPU Usage $65 \pm 1.0\%$ Memory Usage 89.07% โดยทำงานต่อเนื่องเป็นเวลา 72 ± 0.8 วินาที หากมีการเพิ่มจำนวน Memory ให้กับเครื่องเสมือนอีกละ 64 MB เป็น 768 MB จะสามารถรองรับจำนวน User ได้ ซึ่งพบว่าการทำงาน CPU 2 Core สามารถรองรับการให้บริการได้สูงสุดกับ ขนาด Memory 832 MB เป็นต้น

ส่วนที่ 2: ทำหน้าที่ตรวจสอบข้อมูลจากเครื่องเสมือนที่ทำงานในปัจจุบัน

ขั้นตอนที่ 3 ข้อมูลที่ตรวจสอบจะถูกส่งมาเมื่อมีการทำงานของทรัพยากรเครื่องอย่างต่อเนื่องตามระยะเวลาการทำงานของเครื่องเสมือนในทรัพยากรนั้นๆ โดยสามารถทำได้ ด้วยการนำเข้าสู่การประมวลผลเพื่อการแปลงอินพุตที่เป็นค่าจริงเปลี่ยนเป็นอินพุตแบบ ฟัซซี ซึ่งสามารถสร้างฟังก์ชันความเป็นสมาชิกจากการทำงานให้อยู่ในรูปของพจน์ภาษา ได้ดังตาราง 4.4

ตาราง 4.4 พจน์ภาษาของ CPU และ Memory

ภาระงาน	CPU	Memory
น้อย (Low)	ภาระงานต่ำกว่า 50%	
กลาง(Medium)	ภาระงานตั้งแต่ 25% ถึง 75%	
สูง (High)	สูงกว่า 50%	

ขั้นตอนที่ 4 สมการที่ใช้การประมวลผล Fuzzy ของระบบ ในการหาค่าภาระงานของ CPU และ Memory ได้ทำการกำหนดให้ค่าความเป็นสมาชิกขึ้นอยู่กับค่าที่ได้รับภาระงานจากระบบ (x) โดยใช้ฟังก์ชันความเป็นสมาชิกแบบสามเหลี่ยมซึ่งมีพารามิเตอร์ ดังสมการที่ 1

$$triangular(x: a, b, c) = f(x) = \begin{cases} 0, & x < a \\ (x - a)/(b - a), & a \leq x < b \\ (c - x)/(c - b), & b \leq x \leq c \\ 0, & x > c \end{cases} \quad (1)$$

ในการกำหนดฟังก์ชันความเป็นสมาชิกของเซต ทั้ง 3 โดยกำหนดให้

x	แทน	ตัวแปรอินพุต
Low	แทน	ฟังก์ชันความเป็นสมาชิกของเซตภาระงานระดับต่ำ
Medium	แทน	ฟังก์ชันความเป็นสมาชิกของเซตภาระงานระดับปานกลาง
High	แทน	ฟังก์ชันความเป็นสมาชิกของเซตภาระงานระดับสูง



สมการที่ใช้ตรวจสอบเงื่อนไข ภาระงานระดับต่ำ (Low)

$$Low(x: 25, 50, 75) \begin{cases} 1 & , x < 25 \\ (50 - x)/25 & , 25 \leq x < 50 \\ 0 & , x \geq 50 \end{cases} \quad (2)$$

สมการที่ใช้ตรวจสอบเงื่อนไข ภาระงานระดับปานกลาง (Medium)

$$Medium(x: 25, 50, 75) \begin{cases} 0 & , x < 25, x > 75 \\ (x - 50)/25 & , 25 \leq x < 50 \\ (75 - x)/25 & , 50 \leq x \leq 75 \end{cases} \quad (3)$$

สมการที่ใช้ตรวจสอบเงื่อนไข ภาระงานระดับสูง (High)

$$High(x: 25, 50, 75) \begin{cases} 0 & , x < 50 \\ (x - 50)/25 & , 50 \leq x < 75 \\ 1 & , x \geq 75 \end{cases} \quad (4)$$

ส่วนที่ 3: (ขั้นตอนที่ 5) ทำหน้าที่รับผลที่ได้รับการตีความจากส่วนที่ 2 มาดำเนินการจัดสรรทรัพยากร ปัจจุบันเครื่องเสมือนต้องการรับการจัดสรรทรัพยากรจากส่วนกลางเพิ่มขึ้นหรือไม่ เทียบจากความสามารถในการเพิ่มขึ้นของทรัพยากรต่อการให้บริการใน ตาราง 4.2 ฐานความรู้ โดยแบ่งออกตามภาระงานดังนี้

High จะทำการส่งคำขอเพื่อขอรับการจัดสรรทรัพยากรจากระบบโดย CPU จะจัดสรรเป็น Core ที่ ละ 1 Core RAM จะจัดสรรที่ละ 64 MB โดย ทรัพยากรในระบบปฏิบัติการเซิร์ฟเวอร์จะต้องมีเพียงพอจึงจะสามารถทำได้

Medium ไม่มีการดำเนินการอะไร

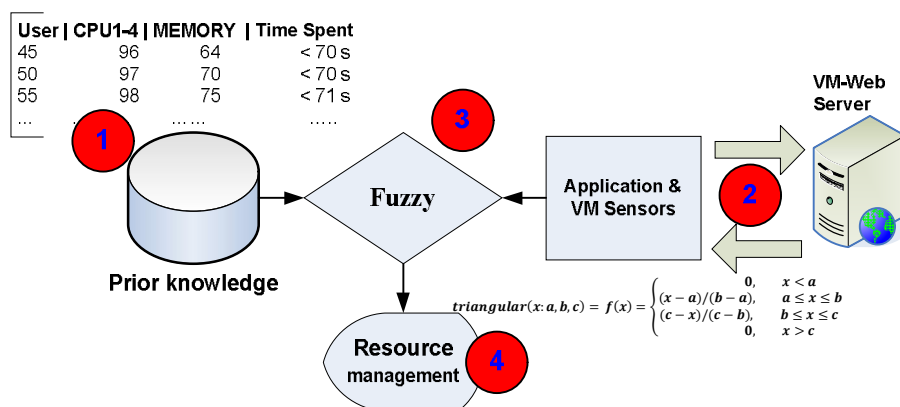
Low จะทำการตรวจสอบว่า เครื่องเสมือนมีการเปลี่ยนแปลงทรัพยากรก่อนหน้านี้หรือไม่ถ้ามี จะทำการคืนค่าทรัพยากรที่ได้รับการจัดสรรให้เป็นค่าก่อนมีการเปลี่ยนแปลง

4.2.2 ผลการทดสอบตัวแบบ Fuzzy logic ในการบริหารจัดการเครื่องเสมือนใน Xen Hypervisor

การทดสอบประสิทธิภาพของตัวแบบโดยการจำลองงานให้เครื่องเสมือน ดังภาพประกอบ

4.2





ภาพประกอบ 4.2 การทดสอบประสิทธิภาพของตัวแบบโดยการจำลองงานให้เครื่องเสมือน

จากภาพประกอบ 4.2 สามารถอธิบายการทดลองในแต่ละส่วนได้ดังนี้

1. ข้อมูลที่ได้จากการทดสอบประสิทธิภาพของเครื่องให้บริการโดยใช้ โปรแกรม Web Stress Tool รูปแบบข้อมูลที่ได้ ดังตาราง 4.5

ตาราง 4.5 รูปแบบข้อมูลที่จัดเก็บในฐานความรู้

Memory	CPU 1- 4		
	CPU %	Mem%	Time (s)

2. ตรวจสอบการทำงานของทรัพยากรในเครื่องเสมือน
3. ใช้พีชชีในการตรวจสอบสถานะงานที่เป็นอยู่
4. จัดสรรทรัพยากรให้เครื่องเสมือนตามเงื่อนไข โดยมีเงื่อนไข 3 ระดับ

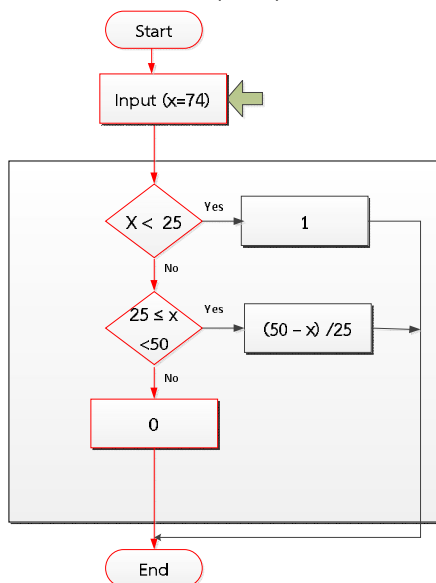
High จัดสรรทรัพยากรทีละ 1 หน่วย ทรัพยากร โดยเทียบจากค่า Prior knowledge
 CPU เพิ่มทีละ 1 Core, Memory เพิ่มครั้งละ 64 MB
 Medium คงทรัพยากรที่ได้รับการจัดสรรก่อนหน้านี้ไว้เท่าเดิม
 Low คืนค่าทรัพยากรที่ได้รับการจัดสรรก่อนหน้านี้ให้กับระบบหลัก

ตัวอย่างการคำนวณ

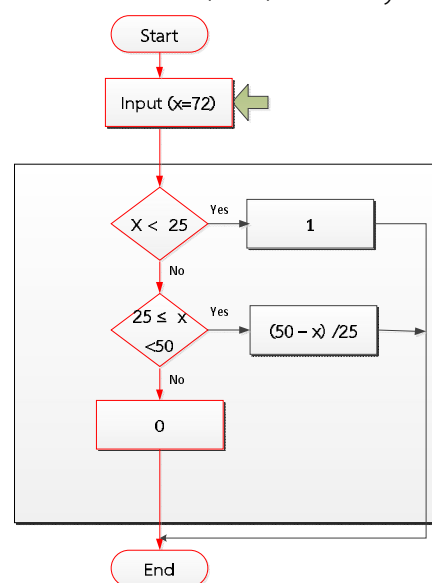
เครื่องเสมือนทรัพยากรที่ได้รับจัดสรร ซีพียู 3 Core หน่วยความจำสำรอง 768 MB ระบบ
 ตรวจพบการทำงานของทรัพยากรภายในเครื่องเสมือนอย่างต่อเนื่องเป็น 70 วินาที โดยมีค่าซีพียูเท่ากับ
 74% หน่วยความจำเท่ากับ 72% จงหาว่าเครื่องเสมือนดังกล่าวควรได้รับทรัพยากรอย่างไรให้เหมาะสม
 กับงาน (โดยใช้ข้อมูลจากตาราง 2) จากข้อมูล จะสามารถคำนวณจากพีชชีได้ว่า CPU (x) = 74 ,
 Memory (x) = 72



กรณี ภาระงานต่ำ (Low) CPU



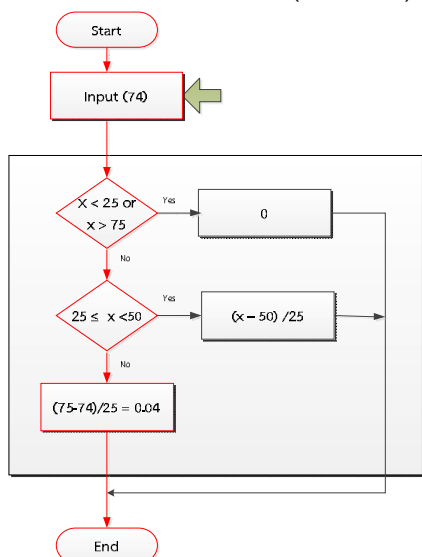
กรณี ภาระงานต่ำ (Low) Memory



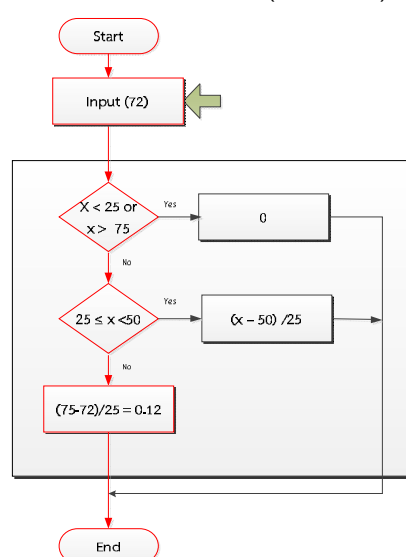
ภาพประกอบ 4.3 แสดงขั้นตอนการทำงานในการคำนวณหาค่ากรณีค่าภาระงานต่ำ ของ CPU และ Memory

จากภาพประกอบ 4.3 เมื่อทำการตรวจสอบค่า x ที่ได้กับสมการตรวจสอบเงื่อนไข กรณี ภาระงานต่ำ (Low) แล้วปรากฏว่า ภาระงานของ CPU และ Memory นี้ไม่ได้อยู่ในระดับต่ำ

กรณี ภาระงานปานกลาง (Medium) CPU



กรณี ภาระงานปานกลาง (Medium) Memory

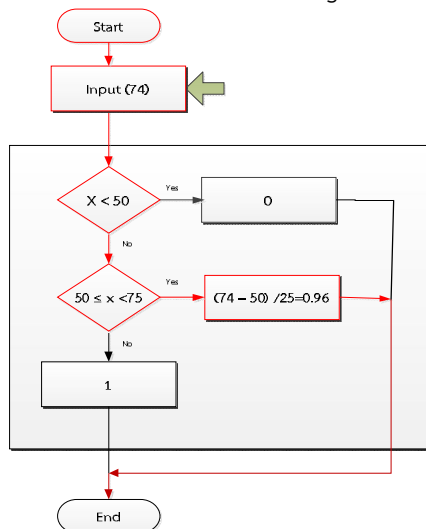


ภาพประกอบ 4.4 แสดงขั้นตอนการทำงานในการคำนวณหาค่ากรณีค่าภาระงานปานกลาง ของ CPU และ Memory

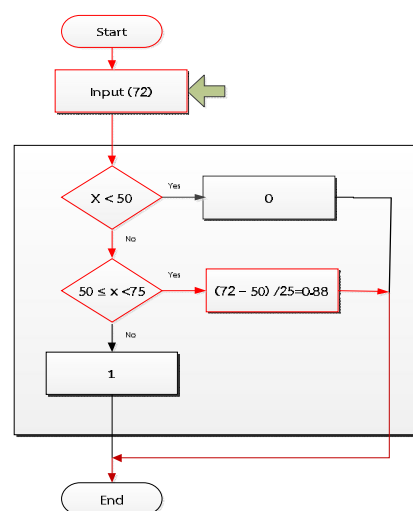


จากภาพประกอบ 4.4 เมื่อทำการตรวจสอบค่า x ที่ได้กับสมการ ตรวจสอบเงื่อนไขภาระงานปานกลาง ปรากฏว่า ภาระงานของ CPU มีค่าเท่ากับ 0.04 และ Memory มีค่าเท่ากับ 0.12 นั้นหมายความว่า ทั้ง CPU และ Memory ค่าอยู่ในภาระงานปานกลาง

กรณี ภาระงานปานกลาง (High) CPU



กรณี ภาระงานปานกลาง (High) Memory



ภาพประกอบ 4.5 แสดงขั้นตอนการทำงานในการคำนวณหาค่ากรณีค่าภาระงานสูงของ CPU และ Memory

จากภาพประกอบ 4.5 เมื่อทำการตรวจสอบค่า x ที่ได้กับสมการ ตรวจสอบเงื่อนไขภาระงานสูง แล้วปรากฏว่า ภาระงานของ CPU มีค่าเท่ากับ 0.96 และ Memory มีค่าเท่ากับ 0.88 นั้นหมายความว่า ทั้ง CPU และ Memory ค่าความรู้มีภาระงานระดับสูง

ดังนั้นจึงสรุปค่าผลของภาระงาน CPU = 74% และ Memory=72% เมื่อนำค่า x เข้าสู่เซตของพีชคณิตแล้ว สามารถสรุปผลของภาระงานนี้ได้ว่า อยู่ในภาระงานสูง และเมื่อเทียบกับตาราง 4.2 การทำงานของเครื่องเสมือนในสภาพ CPU 3 Core สามารถเพิ่ม หน่วยความจำสำรองได้ อีก 1 หน่วย คือ 64 MB จะสามารถรับงานได้เพิ่มขึ้น



ตาราง 4.6 ตัวอย่างการประเมินภาระงานการใช้ทรัพยากรของเครื่องเสมือน

ภาระงานของการใช้ งาน CPU หรือ Memory (x)	ภาระงานระดับต่ำ	ภาระงานระดับปาน กลาง	ภาระงานระดับสูง
86	0.00	0.00	1.00
74	0.00	0.04	<u>0.96</u>
53	0.00	<u>0.88</u>	0.12
32	<u>0.72</u>	0.28	0.00

จากตาราง 4.6 การประเมินภาระงาน ด้วยฟัซซีลอจิก ทำให้สามารถประเมินค่าได้ดีกว่าค่าสถิติพื้นฐานทั่วไป จากตาราง 4.5 จะเห็นได้ว่าในแถวตอนที่ 3 แถวตั้ง ที่ 4 เป็นผลลัพธ์ที่ได้จากการใช้ฟัซซีลอจิกในการะงานของ CPU (x) เท่ากับ 74 ซึ่งค่านี้ต่ำกว่าเกณฑ์ของภาระงานระดับสูงที่กำหนดไว้ (75) แต่เมื่อเข้าสู่เซตของฟัซซีลอจิกแล้ว ปรากฏว่าภาระงานของ ค่า CPU ดังกล่าวมีภาระงานระดับสูง ที่เป็นเช่นนี้เนื่องจากค่าภาระงานของ CPU ดังกล่าวมีแนวโน้มเข้าใกล้ระดับภาระงานสูงถึง 0.96 นั่นเอง

4.3 ผลการเปรียบเทียบประสิทธิภาพการแนวทางการบริหารจัดการทรัพยากรเครื่องเสมือนบน Xen แนวทางใหม่และแนวทางเดิม

4.3.1 งานวิจัยนี้ได้ออกแบบตัวแบบ Fuzzy Logic ในการบริหารจัดการเครื่องเสมือนใน Xen Hypervisor มาเป็นตัวอย่างในการศึกษาเพื่อใช้ทำนายถึงสภาวะภาระงานต่างๆ ในการบริหารจัดการการใช้ทรัพยากรของเครื่องจริงต่อเครื่องเสมือนในระบบ เช่น เซิร์ฟเวอร์ ให้ได้อย่างมีประสิทธิภาพโดยใช้ตัวชี้วัดดังนี้

- 1) ตัวชี้วัดประสิทธิภาพและประสิทธิผลของการใช้ ฟัซซีลอจิกกับภาระงาน CPU
- 2) ตัวชี้วัดประสิทธิภาพและประสิทธิผลของการใช้ ฟัซซีลอจิกกับภาระงาน Memory



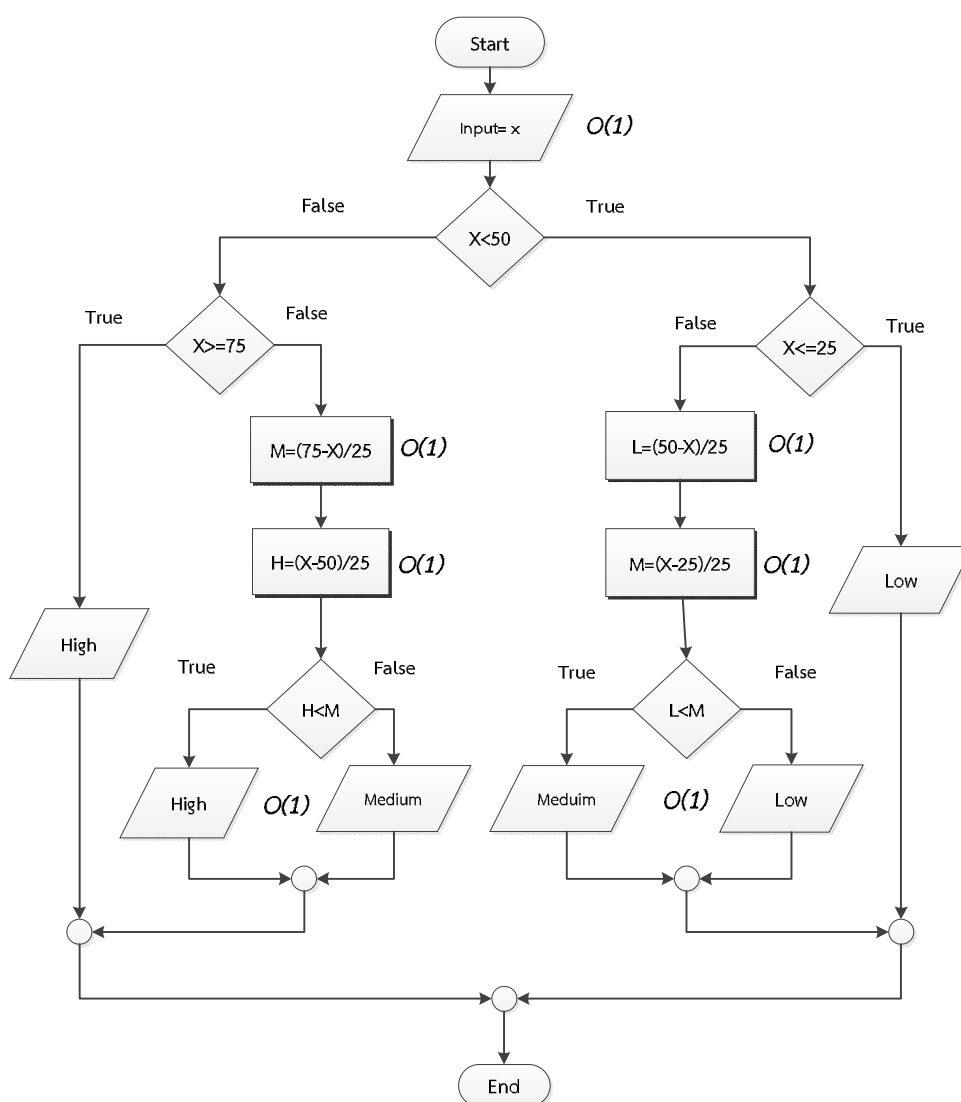
ตาราง 4.7 ผลเปรียบเทียบระหว่าง Non-Fuzzy Model, Fuzzy Model

VM-Resources User	Non-Fuzzy Model			Fuzzy Model		
	State core CPU up	State Memory up	System fail	State core CPU up	State Memory up	System fail
45	No (1CORE)	No (640MB)	No	No (1CORE)	No (640MB)	No
50	No	No	Yes	No	Yes (704MB)	No
100	No	No	Yes	Yes (3CORE)	Yes (832MB)	No
150	No	No	Yes	No	No	No
200	No	No	Yes	Yes (4CORE)	Yes (1024MB)	No

จากตาราง 4.7 พบว่าการจำลองการทำงานด้วยตัวแบบ Fuzzy Model ให้ปรับเปลี่ยนทรัพยากรได้ตามจำนวนภาระงานของ User ที่ส่งเข้ามา ส่งผลให้บริการของระบบไม่ผิดพลาดหรือล้มเหลวต่อการให้บริการ ตรงกันข้ามกับ การให้บริการตามปกติของเครื่องเสมือน ที่กำหนดทรัพยากรคงที่ ไม่สามารถปรับเปลี่ยนทรัพยากรให้สามารถรองรับจำนวนงานที่เพิ่มขึ้นได้

4.3.2 ผลการวัดประสิทธิภาพ การทำงานของตัวแบบฟัซซีลอจิก ด้วยการหา Big-O Notation พิจารณาจากอัลกอริทึมดังภาพประกอบ 4.6





ภาพประกอบ 4.6 การทำงานของตัวแบบฟัซซีลอจิก ด้วยการหา Big-O Notation

จากภาพประกอบ 4.6 เป็นการวัดค่า Big-O โดยพิจารณาค่าสิ่งต่างๆ ใน อัลกอริทึม โดยไม่สนใจคำสั่งพื้นฐานที่ไม่ส่งผลต่อความเร็วในการทำงาน เช่น การกำหนดค่าตัวแปร (เนื่องจากค่า Big-O ของคำสั่งเหล่านี้เป็นค่าคงที่) พบว่าค่า Big-O ของตัวแบบฟัซซีลอจิก มีค่าเท่ากับ $O(1)$ Constant Time Algorithm



บทที่ 5

สรุปผล และอภิปรายผล

งานวิจัยนี้ได้นำหลักการฟัซซีมาเป็นตัวแบบในการศึกษาเพื่อใช้ทำนายถึงสภาวะภาระงานต่างๆในการบริหารจัดการ การใช้ทรัพยากรของเครื่อง การหาค่าความเหมาะสมที่สุดและการจำลองระบบเพื่อหาปัจจัยที่มีผลต่อประสิทธิภาพในการทำงานของเครื่องเสมือน เช่นจำนวนภาระงาน ทรัพยากรที่ได้รับจัดสรรเพียงพอต่อการให้บริการหรือไม่ซึ่งล้วนเป็นข้อมูลที่กำหนดให้แน่นอนได้ยาก มีความคลาดเคลื่อน โดยใช้เกณฑ์เงื่อนไขแบบฟัซซีแทนฟังก์ชันที่มีความซับซ้อนไม่เชิงเส้น สร้างจากฐานความรู้โดยกำหนดตัวแปรในรูปแบบภาษา เช่น มาก ปานกลาง น้อยโดยสามารถสรุปผลการศึกษาอภิปรายผล รวมทั้งข้อเสนอแนะแนวทางในการวิจัยต่อ โดยเรียงลำดับดังนี้

5.1 สรุปผล และอภิปรายผล

งานวิจัยนี้ใช้เกณฑ์แบบฟัซซีในการพัฒนาแบบจำลองการตัดสินใจสำหรับการบริหารจัดการทรัพยากรของเครื่องเสมือนเพื่อหารูปแบบที่เหมาะสมในการจัดสรรทรัพยากรให้เครื่องต่อการทำงาน โดยเกณฑ์แบบฟัซซีมีตัวแปรที่ใช้ในการตัดสินใจคือระยะเวลาในการทำงานอย่างต่อเนื่องของ ซีพียูและหน่วยความจำสำรอง ซึ่งในการตัดสินใจการจัดสรรทรัพยากรของเครื่องเสมือน คือโหมดการเพิ่ม Core ซีพียู และหน่วยความจำสำรอง ตามจำนวนที่เหมาะสมให้แก่เครื่องเสมือนโดยไม่กระทบต่อทรัพยากรในการทำงานหลักของระบบ ผลลัพธ์ที่ได้จากวิธีการนี้จะสามารถบริหารจัดการทรัพยากรของเครื่องเสมือนแตกต่างไปจากการบริหารจัดการทรัพยากรในรูปแบบเดิมๆ คือ ใช้ดัชนี 3 ตัวคือ Low, Medium และ High ซึ่งพบว่าแบบจำลองฟัซซีมีการตัดสินใจในการบริหารจัดการทรัพยากรให้เครื่องเสมือนได้ตรงตามความต้องการกับภาระงานที่เป็นอยู่ ณ ขณะนั้น ซึ่งแบ่งรายละเอียดของแต่ละประเด็นนำเสนอต่อไปนี้

5.1.1 ประสิทธิภาพการเฝ้าระวังทรัพยากรของเครื่องจริง

เนื่องจากการเฝ้าระวังทรัพยากรของเครื่องจริงคือการทราบถึงสถานการณ์ให้บริการของเครื่องคอมพิวเตอร์เพื่อให้มีทรัพยากรเพียงพอต่อการให้บริการ ไม่นำไปสู่เหตุการณ์การหยุดให้บริการด้วย ที่ไม่มีทรัพยากรคงเหลือในการให้บริการ ซึ่งจากการศึกษาและทดสอบการทำงานพบว่าสามารถเฝ้าระวังและทราบถึงการเปลี่ยนแปลงของทรัพยากรของเครื่องจริงได้อย่างมีประสิทธิภาพ

5.1.2 ประสิทธิภาพของตัวต้นแบบ

เนื่องจากวัตถุประสงค์ของตัวต้นแบบที่นำเสนอคือเสนอแนวทางในการบริหารจัดการทรัพยากรโดยประยุกต์เข้ากับฟัซซีลอจิก (Fuzzy Logic) ในเครื่องให้บริการการเสมือนเพื่อค้นหาว่าเครื่องเสมือนควรได้รับการจัดสรรทรัพยากรอย่างไรให้เพียงพอต่อการให้บริการเนื่องด้วยปัจจัยต่างๆ ที่เกี่ยวข้องกันจะมีผลต่อการตอบสนองที่ต่างกัน ปัจจัยที่นำมาวิเคราะห์ร่วมกันได้แก่ ซีพียู หน่วยความจำสำรอง จำนวนผู้ใช้งาน และเวลาในการทำงาน สำหรับสร้างฐานความรู้ (Knowledge Base) ที่ต้องจัดเตรียมให้กับฟัซซี โดยใช้เครื่องมือทดสอบประสิทธิภาพของระบบที่เรียกว่า Web Stress Tool 7 จากผลการทดลองปรากฏว่า ถ้าข้อมูลของ ฐานความรู้ มีความละเอียดมากขึ้นจะทำให้ผลของการ



วิเคราะห์มีความละเอียดตามไปด้วย (ค่าที่คำนวณได้จะให้คำตอบที่มีความเป็นไปได้หลายค่า) และถ้าข้อมูลที่เก็บมีความหยาบมาก จะทำให้การจัดสรรทรัพยากรไม่เหมาะสมกับภาระงานที่เป็นอยู่ และหากเปรียบเทียบกับการทำงานในรูปแบบเดิม หรือใช้โปรแกรมการตรวจสอบการทำงาน เช่น โปรแกรม Reliability and Performance Monitor ของ Microsoft Corporation จะพบว่าโปรแกรกดังกล่าวสามารถทำหน้าที่เพียงแค่งานการทำงานของทรัพยากรของเครื่องเท่านั้น ไม่สามารถปรับแต่ง หรือสร้างรูปแบบทรัพยากรที่เหมาะสมในการทำงานได้ ณ ขณะนั้นเพื่อให้เครื่องเสมือน มีประสิทธิภาพสามารถรองรับภาระงานที่เป็นปัจจุบันได้

5.1.3 ประสิทธิภาพของอัลกอริทึม

สำหรับการทำงานของตัวต้นแบบโดยการวัดประสิทธิภาพในการใช้อัลกอริทึมวิเคราะห์หาค่าระดับภาระงานนั้น โดยแทนค่าด้วยฟังก์ชันที่มีขนาดของข้อมูลป้อนเข้าเป็นตัวแปร ซึ่งเวลาที่ใช้ในการทำงานของคอมพิวเตอร์แต่ละเครื่องอาจใช้เวลาในการทำงานแก้ไขปัญหาคือแตกต่างกันไป ดังนั้นเพื่อหลีกเลี่ยงความแตกต่างในจุดนี้จึงใช้ Big-O notation เป็นตัววัดความซับซ้อน ปัญหาที่มีความซับซ้อนด้านเวลาเป็น $O(n)^2$ บนเครื่องอื่นๆ ด้วยเช่นกัน จะเห็นได้ว่าการใช้ Big-O notation ในการวิเคราะห์จะช่วยในการหลีกเลี่ยงการกล่าวถึงรายละเอียดที่มีความแตกต่างระหว่างเครื่องคอมพิวเตอร์เสมือน และกล่าวถึงความซับซ้อนด้านเวลาว่าเป็น “เวลาที่ใช้ในการแก้ปัญหา” หรือ “เวลาที่ใช้ในการทำงาน” อัลกอริทึม ของตัวแบบนี้จึงเหมาะกับการทำงานของระบบที่มีการเปลี่ยนแปลงได้ตลอดเวลา (Real Time)

5.2 ข้อเสนอแนะ

จากการทำงานวิจัยนี้และศึกษางานวิจัยที่เกี่ยวข้องของผู้วิจัยข้อเสนอแนะประเด็นในการนำงานวิจัยไปพัฒนาในอนาคตมีดังนี้

5.2.1 งานวิจัยนี้สร้างต้นแบบในการเพิ่มประสิทธิภาพ การบริหารจัดการทรัพยากรโดยประยุกต์เข้ากับฟัซซี (Fuzzy Logic) ในเครื่องให้บริการการเสมือนที่ให้บริการแบบเว็บเซิร์ฟเวอร์เท่านั้น เพราะฉะนั้นบางส่วนของงานจึงมีข้อจำกัดในการเอาไปทดสอบสำหรับการพัฒนา หรือนำไปใช้ในระบบจริง ดังนั้นหากต้องการนำไปใช้งานกับระบบจริงควรมีการพัฒนาปรับปรุงประสิทธิภาพและโปรแกรม โดยคำนึงถึงเรื่องการพัฒนาโปรแกรมให้ผู้ใช้สามารถใช้งานได้ง่าย

5.2.2 ทำการศึกษาการสร้างฐานความรู้ (Knowledge Base) ที่ต้องจัดเตรียมให้กับฟัซซี โดยใช้โปรแกรม Web Stress Tool ออกแบบให้สร้างฐานความรู้สำหรับการให้บริการของเครื่องเซิร์ฟเวอร์ ที่มีการให้บริการที่หลากหลาย เช่น ไฟล์เซิร์ฟเวอร์ (File Server) วิดีโอเซิร์ฟเวอร์หรือ (Streaming Server) เมล์เซิร์ฟเวอร์ (Mail Server) ซึ่งเป็นส่วนที่สำคัญมากในการในการตัดสินใจของกับฟัซซี (Fuzzy Logic) เพื่อหาความสัมพันธ์ จะทำให้ การบริหารจัดการทรัพยากรของระบบมีประสิทธิภาพมากขึ้น



เอกสารอ้างอิง



เอกสารอ้างอิง

- [1] Smith JE, Nair R. The architecture of virtual machines. *IEEE Computer Society* 2005; 38(5): 32 – 38: May.
- [2] Hypervisor. [online]. 2010 [cited 8 August 2012]; Available from: <http://en.wikipedia.org/wiki/Hypervisor>.
- [3] VMware. [online]. 2011 [cited 8 August 2012]; Available from: <http://www.vmware.com>.
- [4] Barham P, Dragovic B, Fraser K, Hand S, Harris T. Xen and the art of virtualization. *Proceedings of the nineteenth ACM symposium on Operating systems principles 2003*; 4: 164-177: December.
- [5] Quiroz A, Kim H, Parashar M, Gnanasambandam N, Sharma N. Towards Autonomic Workload Provision for Enterprise Grids and Clouds. *IEEE/ACM International Conference on Grid Computing 2009*; 2: 50-57.
- [6] Van HN, Tran FD, Menaud JM. *Autonomic virtual resource management for service hosting platforms*. ICSE International Conference on Software Engineering; Vancouver, BC. Software Engineering Challenges of Cloud Computing 2009. CLOUD '09. ICSE Workshop on; 1-8.
- [7] Yazir YO, Matthews C, Farahbod R, Neville S, Guitouni A, Ganti S, et al. *Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis*. Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing; 5-10 July 2010; Miami, FL. IEEE; 91-98.
- [8] VMware. *Virtualization: architectural considerations and other evaluation criteria*. [online]. 2010 [cited 8 August 2011]; Available from: http://www.vmware.com/pdf/virtualization_considerations.pdf.
- [9] IBM. *Virtualization in education*. [online]. 2011 [cited 30 September 2012]; Available from: <http://www-07.ibm.com/solutions/in/education/download/Virtualizationin20Education.pdf>.
- [10] Goldberg, PR. *Architectural Principles for Virtual Computer Systems*. Harvard University. 22-26.
- [11] Lim B-H. A Hosted Virtual Machine Architecture. [online]. 2011 [cited 19 August 2012]; Available from: http://static.usenix.org/event/usenix01/sugerman/sugerman_html/node3.html.
- [12] Sugerman J, Venkitachalam G, Lim B-H. *Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor*. Proceedings of the 2001 USENIX Annual Technical Conference; June 25–30; Boston, Massachusetts, USA. 1-14.



- [13] Rose R. *Survey of System Virtualization Techniques*. [online]. 2011 [cited 14 March 2004]; Available from: <http://ir.library.oregonstate.edu/xmlui/bitstream/handle/1957/9907/rose-virtualization.pdf>.
- [14] Xen Windows GplPv. [online]. 2011 [cited 10 September 2012]; Available from: <http://wiki.xensource.com/xenwiki/XenWindowsGplPv>.
- [15] IBM. VM History and Heritage. [online]. 2011 [cited 10 September 2012]; Available from: <http://www.vm.ibm.com/history>.
- [16] VMware. *VMware Introduces Support for Cross-Platform Paravirtualization*. [online]. 2011 [cited 10 September 2012]; Available from: <http://www.vmware.com/company/news/releases/050907PV.html>.
- [17] LWN.net. Kernel development. [online]. 2011 [cited 10 September 2012]; Available from: <http://lwn.net/Articles/412164>.
- [18] Neiger G, Santoni M, Leung F, Rodgers D, Uhlig R. Intel Virtualization Technology: Software-only virtualization with the IA-32 and Itanium architectures. *Intel Technology Journal* 2006; 10(3): 167-178.
- [19] Microsoft Windows 2000. [online]. 2011 [cited 10 September 2014]; Available from: <http://lwn.net/Articles/412164>. <http://www.microsoft.com>.
- [20] Microsoft. Microsoft Windows 2003. [online]. 2011 [cited 10 September 2014]; Available from: <http://www.microsoft.com>.
- [21] Altrock CV. *Fuzzy logic and neurofuzzy applications explained*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA 1995.
- [22] Biacino L, Gerla G. Fuzzy logic, continuity and effectiveness. *Archive for Mathematical Logic* 2002; 41(7): 643-667.
- [23] Pedrycz W, Gomide F. *Fuzzy Systems Engineering: Toward Human Centric Computing* Hardcover; 2007.
- [24] Wiedermann J. Characterizing the super-Turing computing power and efficiency of classical fuzzy Turing machines. *Theoretical Computer Science – Super-recursive algorithms and hypercomputation* 2004; 317(1-3): 61-69.
- [25] Novák V. On fuzzy type theory. *Fuzzy Sets and Systems* 2004; 2: 235-237.
- [26] Pelt V, Miles. Fuzzy Logic Applied to Daily Life. *Seattle* 2008; 4: 134-146.
- [27] พพยุ่ง มีสัจ. ระบบฟัซซีและโครงข่ายประสาทเทียม. กรุงเทพฯ: ศูนย์ผลิตตำราเรียนมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ; 2555.
- [28] RamMohan N.R, Baburaj E. Resource Allocation Techniques in Cloud computing-Research Challenges for Applications. Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on; IEEE; 556-560.



- [29] Korzeniewski R, Plik:XEN-schema. [online]. 2011 [cited 18 March 2014]; Available from: <http://pl.wikipedia.org/wiki/Plik:XEN-schema.png>.
- [30] Type-1 and type-2 hypervisors. [online]. 2011 [cited 10 February 2014]; Available from: <http://upload.wikimedia.org/wikipedia/commons/e/e1/Hyperviseur.png>
- [31] Full Virtualization of the Underlying Platform. [online]. 2011 [cited 10 February 2014]; Available from: <http://itmanagement.earthweb.com/img/2010/05/virtualization-figure3.jpg>
- [32] virtualization-figure3.[online]. 2011 [cited 10 February 2014]; Available from: <http://www.datamation.com/img/2010/05/virtualization-figure3.jpg>.
- [33] Butler J, Sparks S. Symantec. Windows rootkits of 2005, part one.[online]. 2012 [cited 2 November 2012]; Available from: <http://www.symantec.com/connect/articles/windows-rootkits-2005-part-one>.
- [34] Fuzzy logic temperature. [online]. 2011 [cited 10 February 2014]; Available from: http://en.wikipedia.org/wiki/File:Fuzzy_logic_temperature_en.svg.
- [35] Webserver Stress Tool.[online]. 2012 [cited 20 February 2015]; Available from: <http://download-cdn.paessler.com/download/webstressmanual.pdf>.



ภาคผนวก



ภาคผนวก ก
การติดตั้งระบบปฏิบัติการ Xen Server



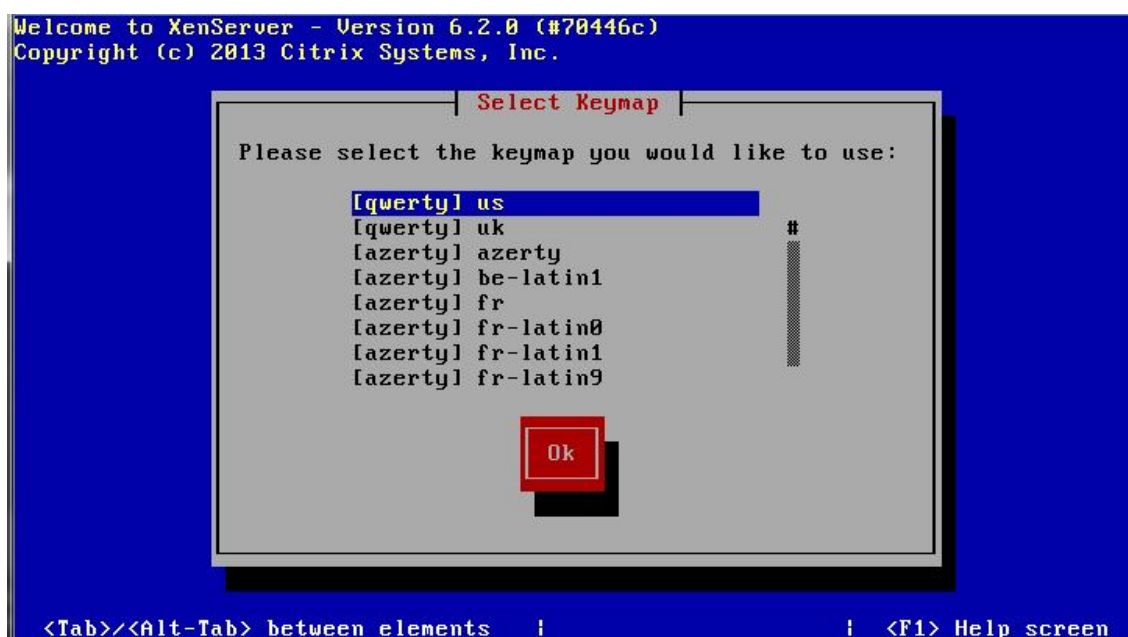
การติดตั้งโปรแกรม

1. ติดตั้ง XenServer ลงบนเครื่องจริงที่ได้จัดเตรียมไว้ โดยใส่แผ่น CD เข้าไปยังเครื่องดังกล่าว แล้วจะปรากฏภาพการติดตั้ง



ภาพประกอบ ก.1 เริ่มติดตั้ง XenServer



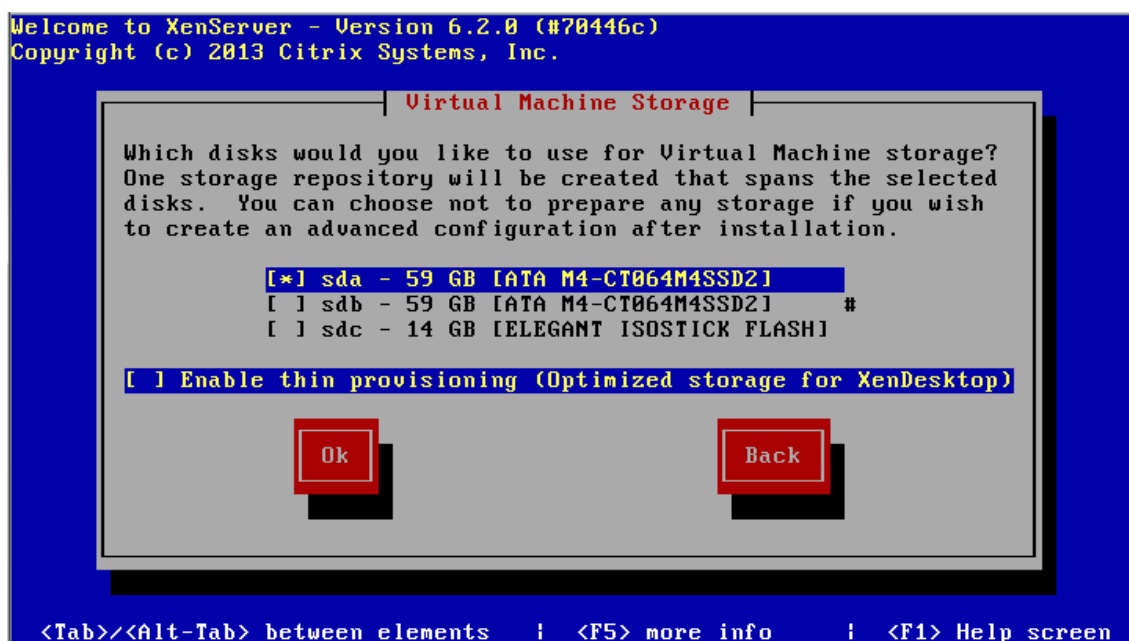


ภาพประกอบ ก.2 เลือกรูปแบบภาษาในการติดตั้ง

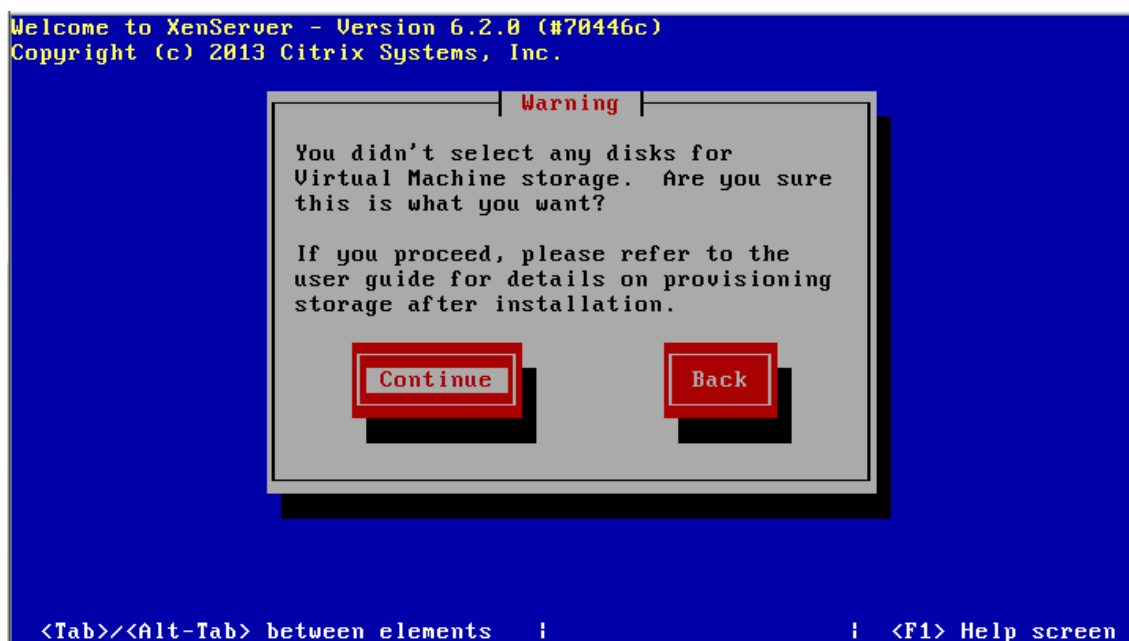


ภาพประกอบ ก.3 ทำการตรวจสอบความถูกต้องของข้อมูลก่อนติดตั้ง



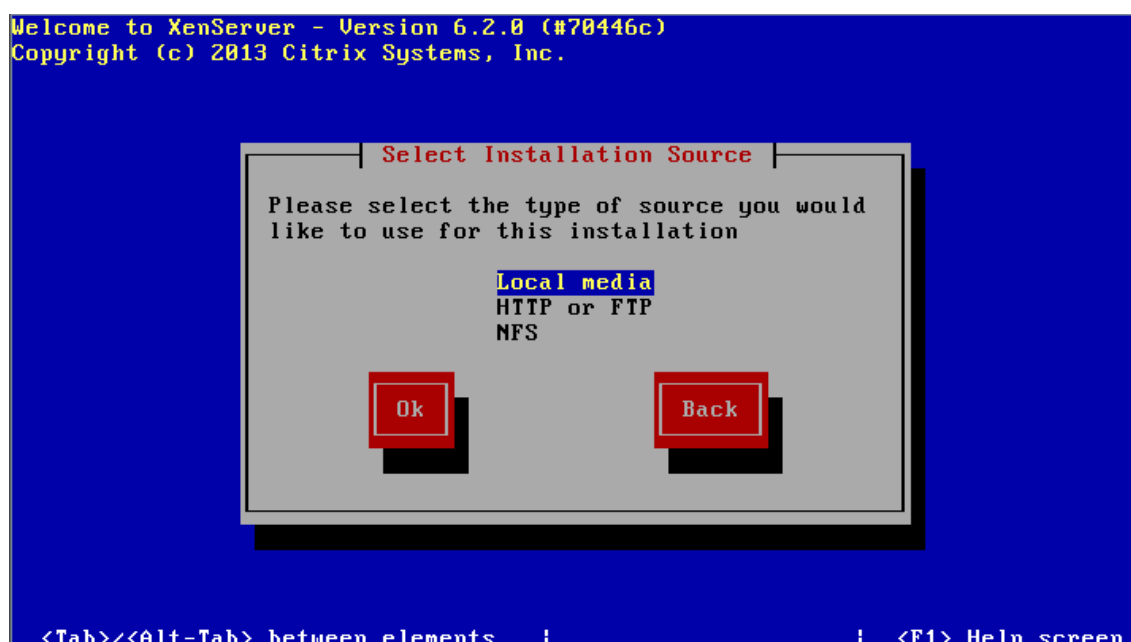


ภาพประกอบ ก.4 เลือกพื้นที่ติดตั้ง

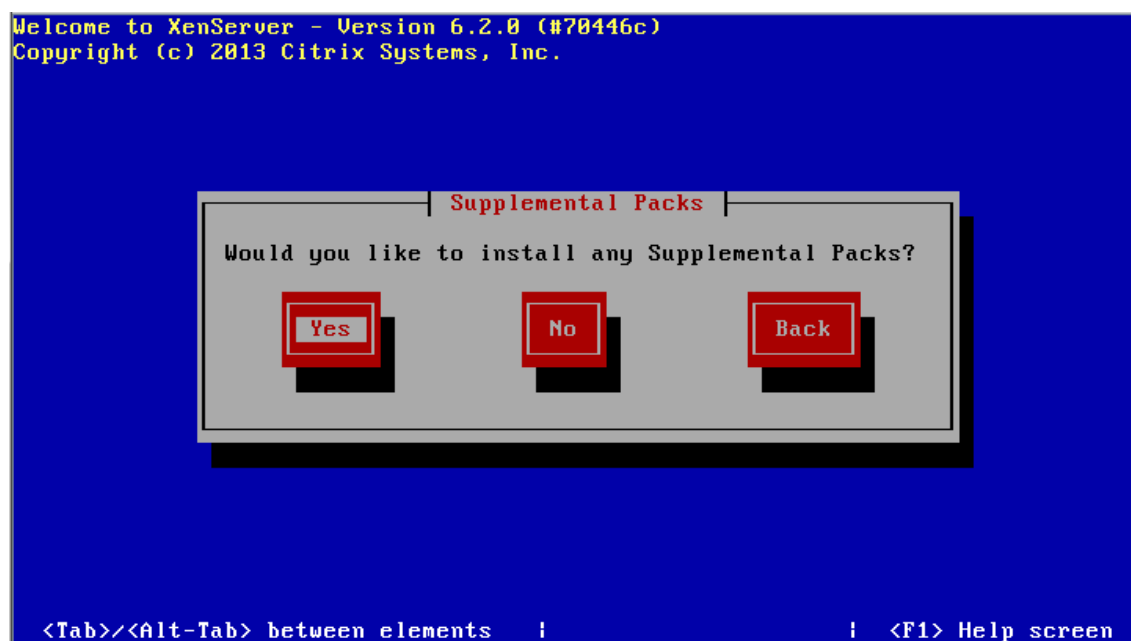


ภาพประกอบ ก.5 ยืนยันการติดตั้งดังกล่าว



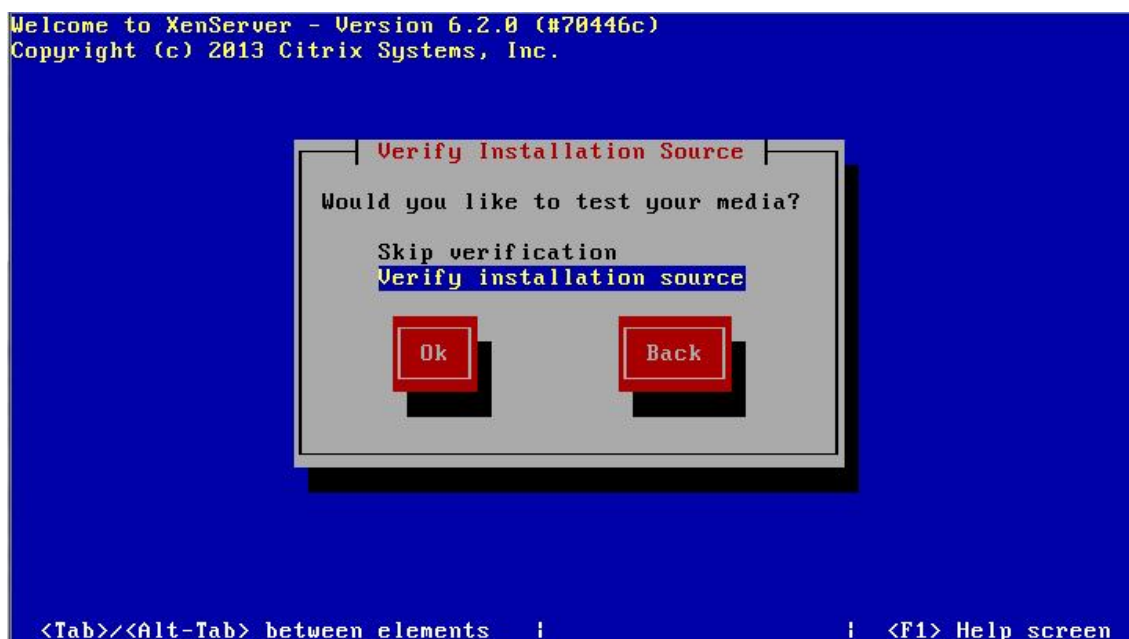


ภาพประกอบ ก.6 เลือกแหล่งข้อมูลที่จะนำมาติดตั้ง

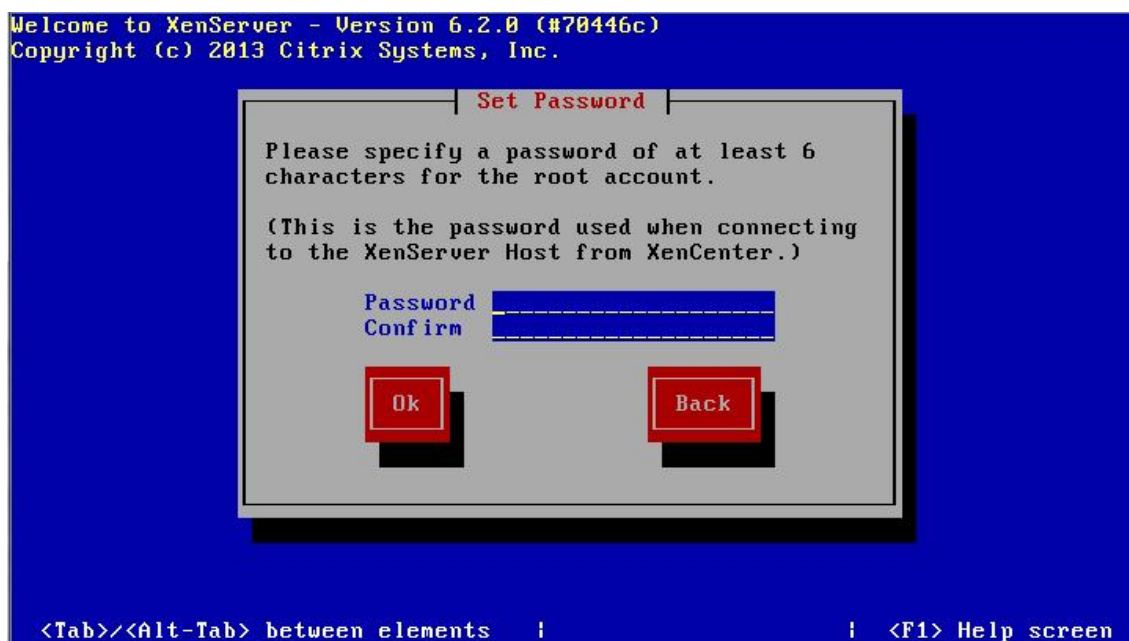


ภาพประกอบ ก.7 ติดตั้งรายละเอียดส่วนเสริม



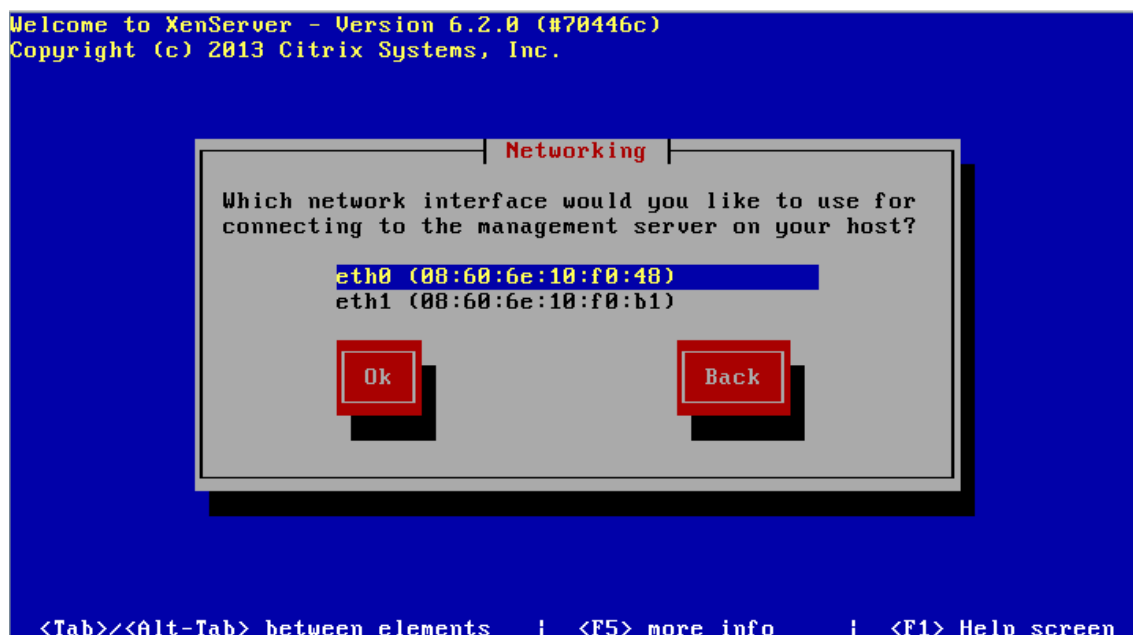


ภาพประกอบ ก.8 ตรวจสอบยืนยันแหล่งข้อมูลที่นำมาติดตั้งว่าถูกต้องหรือไม่

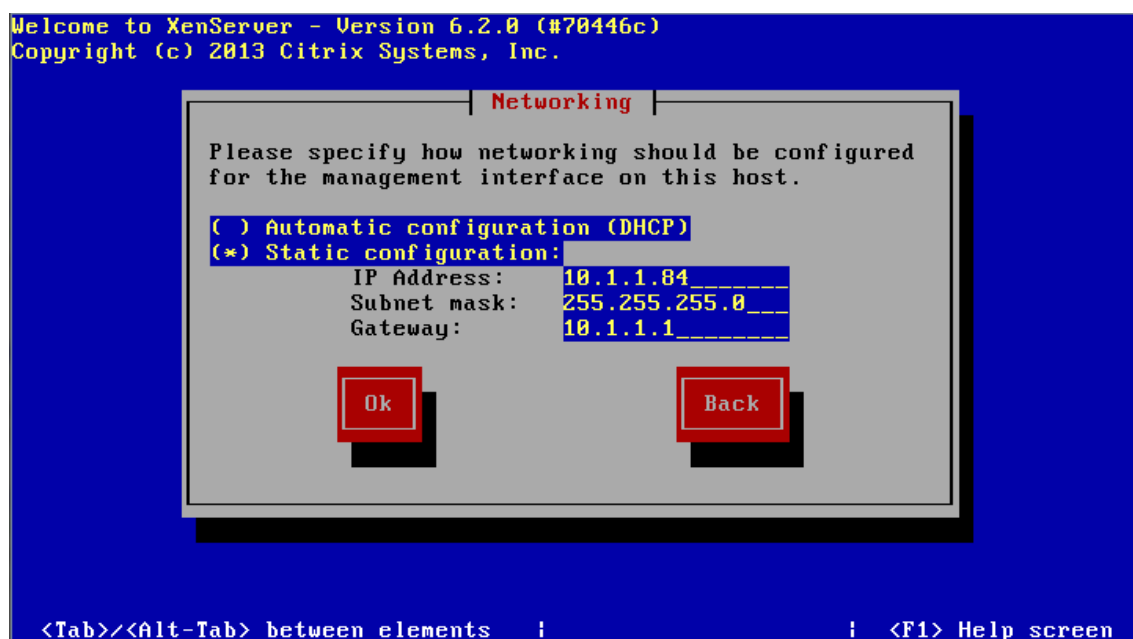


ภาพประกอบ ก.9 ป้อนชุดรหัสเพื่อใช้บริหารจัดการระบบ



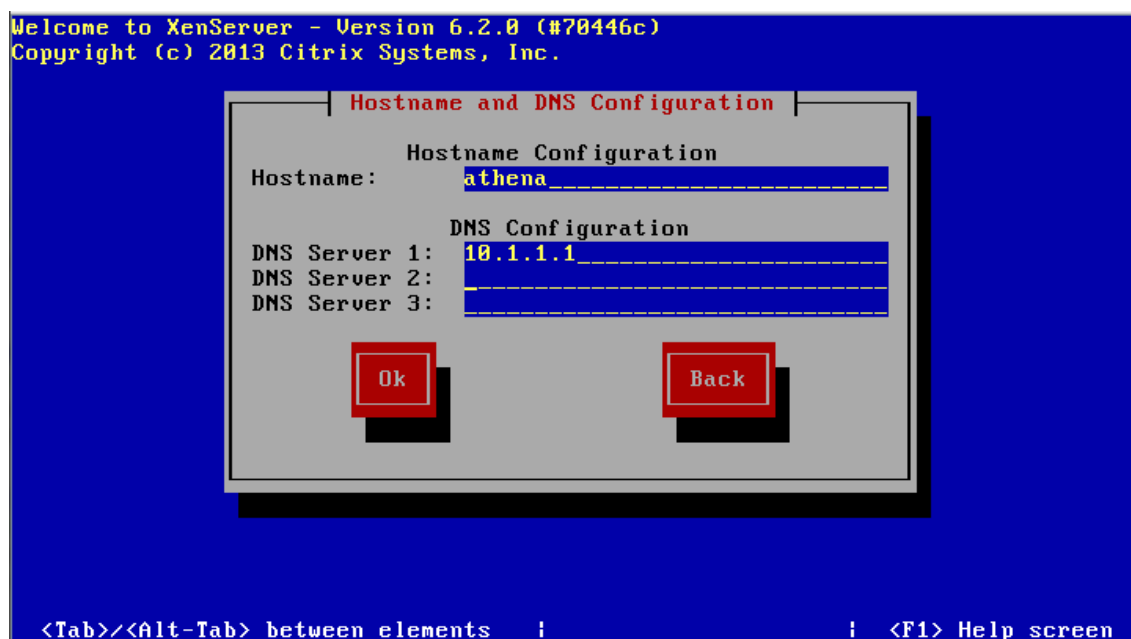


ภาพประกอบ ก.10 เลือกใช้อินเตอร์เฟซเน็ตเวิร์คเพื่อใช้บริหารจัดการผ่านเครือข่าย



ภาพประกอบ ก.11 กำหนดหมายเลข IP เพื่อใช้บริหารจัดการผ่านเครือข่าย



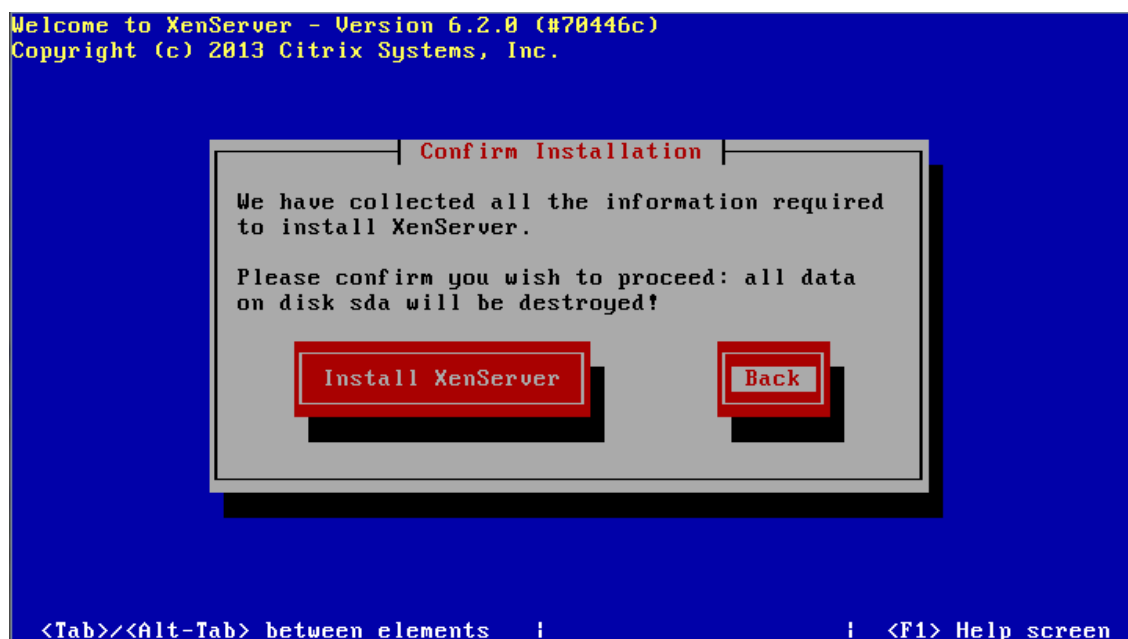


ภาพประกอบ ก.12 กำหนดชื่อเครื่อง และหมายเลข IP ของ DNS

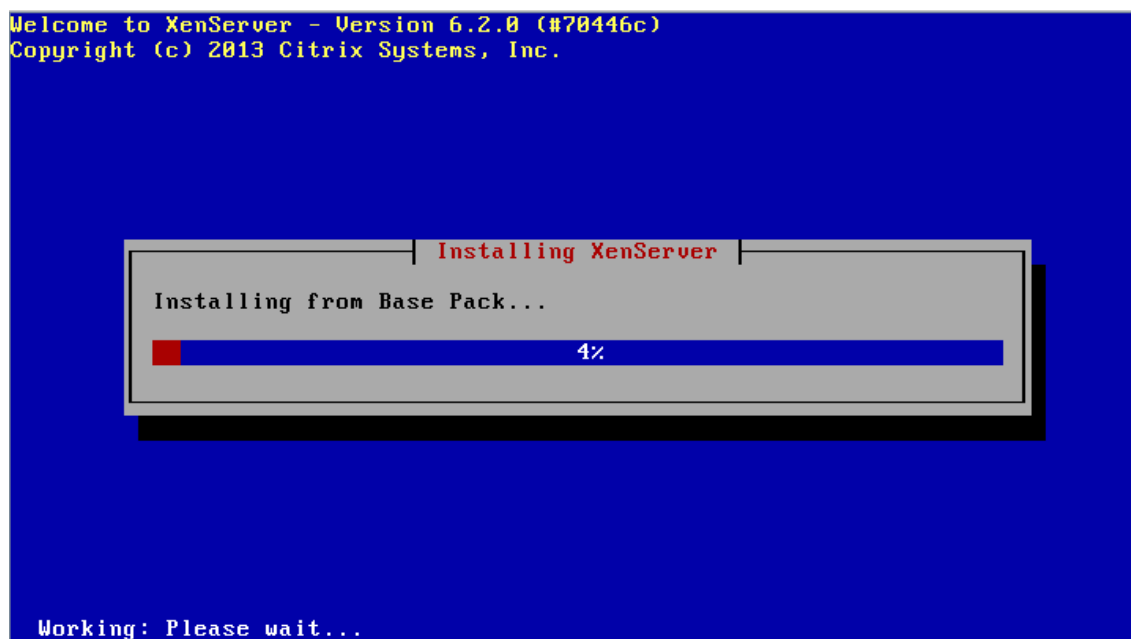


ภาพประกอบ ก.13 เลือกประเทศที่ในการติดตั้ง





ภาพประกอบ ก.14 ติดตั้งระบบปฏิบัติการ

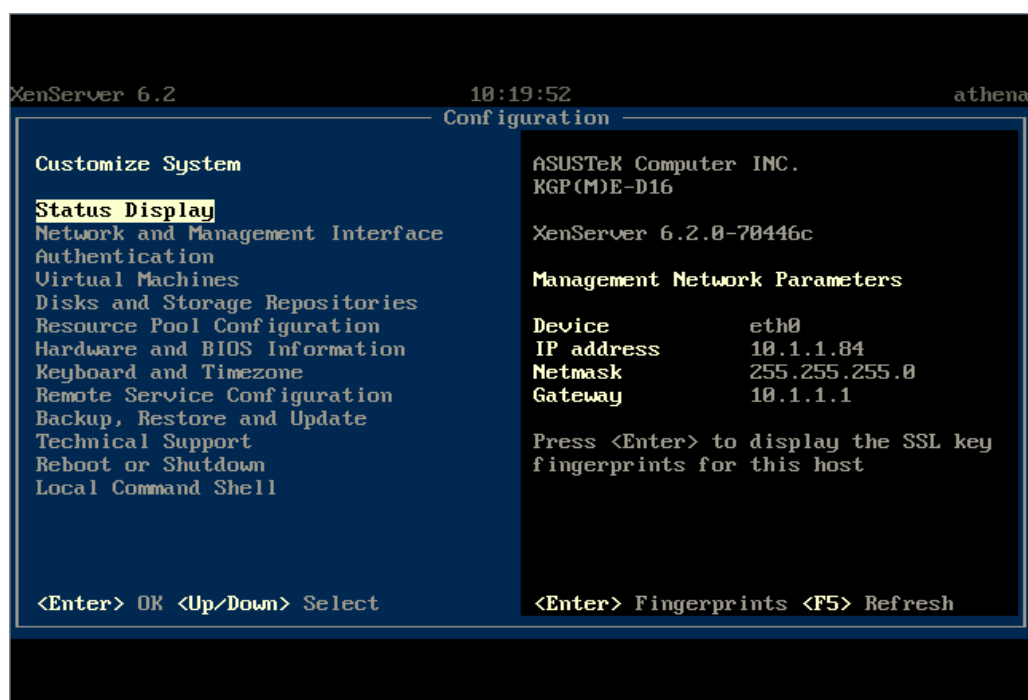


ภาพประกอบ ก.15 แสดงสถานะ ติดตั้งระบบปฏิบัติการ





ภาพประกอบ ก.16 เมื่อติดตั้งระบบปฏิบัติการเสร็จจะทำการเริ่มต้นระบบปฏิบัติการใหม่



ภาพประกอบ ก.17 เข้าสู่หน้าบริหารจัดการระบบปฏิบัติการ Xen



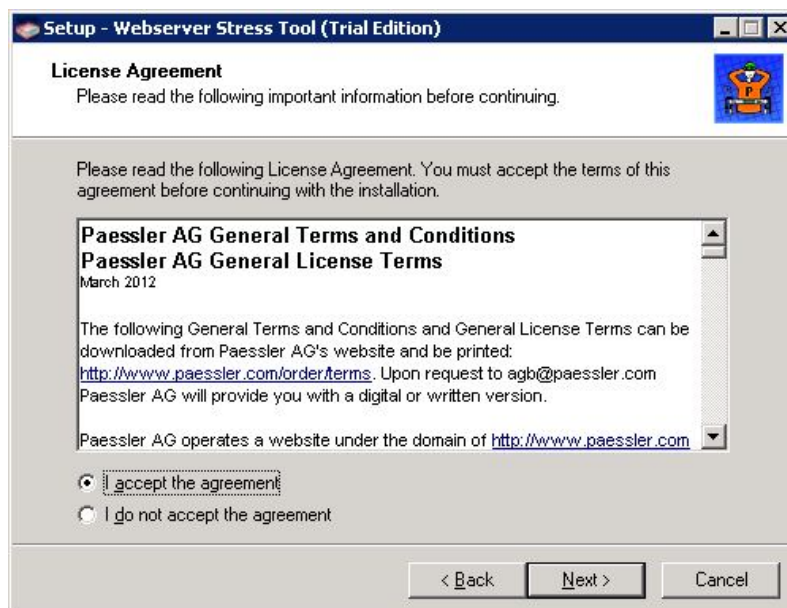
ภาคผนวก ข
การติดตั้งโปรแกรม ทดสอบระบบ Webserver Stress Tool



1. ติดตั้งโปรแกรมป้องกัน Webserver Stress Tool



ภาพประกอบ ข.1 เริ่มติดตั้งโปรแกรมทดสอบ Webserver Stress Tool ที่เครื่องลูกข่าย

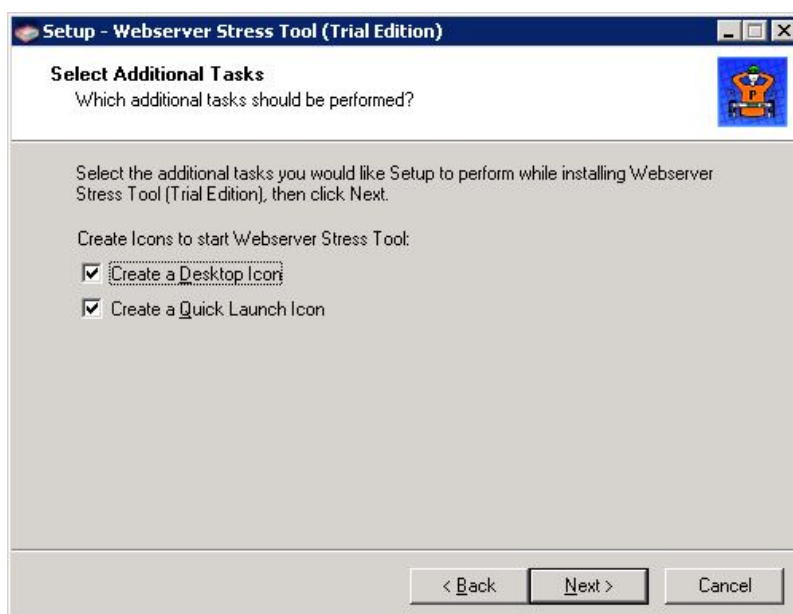


ภาพประกอบ ข.2 ยอมรับ License Agreement





ภาพประกอบ ข.3 พื้นที่ติดตั้งโปรแกรม

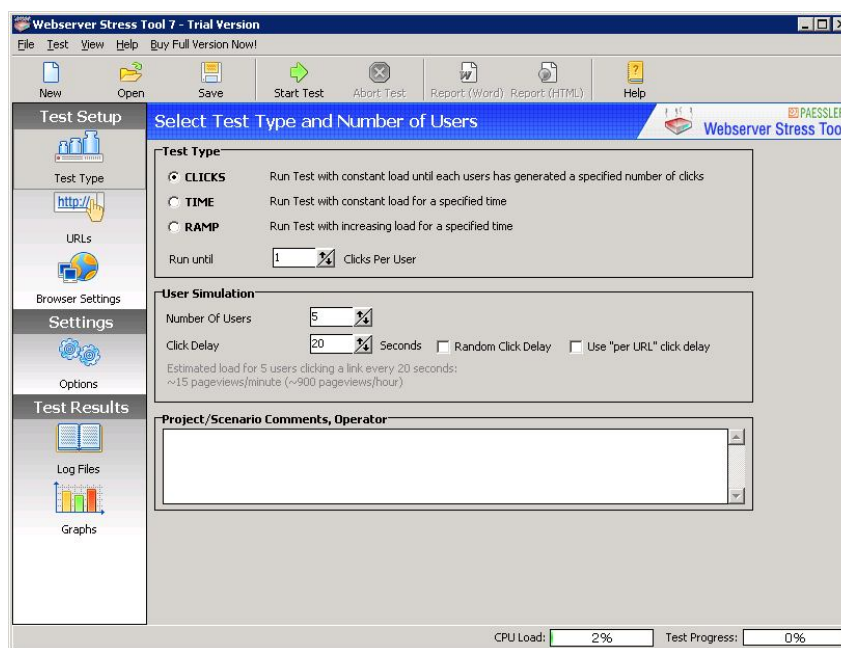


ภาพประกอบ ข.4 สร้าง Shortcut





ภาพประกอบ ข.5 การติดตั้งเสร็จสมบูรณ์



ภาพประกอบ ข.6 ตัวอย่างหน้าต่างหลักโปรแกรม



ประวัติย่อผู้วิจัย



ประวัติย่อผู้วิจัย

ชื่อ-นามสกุล	นายรัตนเดช ชมภูนุช
วัน เดือน ปีเกิด	วันที่ 21 มิถุนายน พ.ศ. 2529
จังหวัด และประเทศที่เกิด	จังหวัดอุดรธานี ประเทศไทย
ประวัติการศึกษา	พ.ศ. 2548 มัธยมศึกษาตอนปลาย โรงเรียนอุดรพัฒนาการ จังหวัดอุดรธานี พ.ศ. 2551 ปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม พ.ศ. 2558 ปริญญาวิทยาศาสตรบัณฑิต (วท.ม.) สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยมหาสารคาม

ที่อยู่ที่สามารถติดต่อได้

งานระบบเครือข่ายคอมพิวเตอร์ สำนักคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม
ชั้น 3, 4 อาคารวิทยบริการ B ตำบลขามเรียง อำเภอกันทรวิชัย จังหวัดมหาสารคาม 44150

ผลงานวิจัย

รัตนเดช ชมภูนุช ศรัณยู บุตรโคตร และสุชาติ คุ่มมณี. (2551). “ระบบเฝ้าระวังแบบ
ปัจจัยสัมพันธ์” ใน: การประชุมวิชาการ Joint Conference on Computer Science and
Software Engineering (JCSSE-2008), หน้า 150-155.

