การพัฒนาส่วนติดต่อผู้ใช้งานสำหรับซอฟต์แวร์จำลองการทำงานคลาวด์ซิม

DEVELOPMENT OF USER INTERFACE FOR CLOUDSIM SIMULATION SOFTWARE

ศิขา ตติยานนท์

วุฒิชัย บัณฑิตพิพัฒนกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การพัฒนาส่วนติดต่อผู้ใช้งานสำหรับซอฟต์แวร์จำลองการทำงานคลาวด์ซิม

DEVELOPMENT OF USER INTERFACE FOR CLOUDSIM SIMULATION SOFTWARE

คีตา อติอานนท์
วรุตม์ ยันตรัตนะวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาส่วนติดต่อผู้ใช้งานสำหรับซอฟต์แวร์จำลองการทำงานคลาวด์ซิม
DEVELOPMENT OF USER INTERFACE FOR CLOUDSIM SIMULATION SOFTWARE

ผู้จัดทำ

1. นางสาวคีตา	อติอานนท์	รหัสนักศึกษา	53010170
2. นายวรุตม์	ยันตรัตนะวรรณ	รหัสนักศึกษา	53011438

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาส่วนติดต่อผู้ใช้งานสำหรับซอฟต์แวร์จำลองการทำงานคลาวด์ซิม

นางสาว คีตา อติอานนท์ 53010170
นาย วรุตม์ ยันตรัตนะวรรณ 53011438
ผศ.ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2556

บทคัดย่อ

คลาวด์ซิม (CloudSim) เป็นหนึ่งในซอฟต์แวร์จำลองการทำงานของการประมวลผลแบบกลุ่มเมฆที่มีนักวิจัยด้านการประมวลผลแบบกลุ่มเมฆใช้เป็นเครื่องมือในการประเมินประสิทธิภาพของระบบ คลาวด์ซิมเป็นซอฟต์แวร์ที่ทำงานด้วยภาษาจาวา ในการใช้งานคลาวด์ซิมผู้ใช้งานต้องมีความรู้ภาษาจาวาและต้องกำหนดค่าพารามิเตอร์ต่างๆ สำหรับการจำลองการทำงานเพียงหนึ่งสถานการณ์เท่านั้น ถ้าผู้ใช้งานต้องการผลการจำลองการทำงานหลายสถานการณ์ ผู้ใช้งานต้องทำการกำหนดค่าพารามิเตอร์ใหม่สำหรับสถานการณ์นั้นๆ ซึ่งอาจก่อให้เกิดความยุ่งยากในการใช้งาน การต้องใช้เวลาเพิ่มขึ้นกว่าจะได้ผลการจำลองการทำงานของทุกสถานการณ์และข้อผิดพลาดระหว่างการกำหนดพารามิเตอร์ เราจึงเกิดแรงบันดาลใจให้เราทำการการพัฒนาส่วนติดต่อผู้ใช้งานสำหรับซอฟต์แวร์จำลองการทำงานคลาวด์ซิม เพื่อให้ผู้ใช้งานสามารถใช้งานได้สะดวกขึ้น ผู้ใช้งานไม่จำเป็นต้องรู้ภาษาจาวา สามารถกำหนดค่าพารามิเตอร์ต่างๆ สำหรับหลายสถานการณ์ได้ผ่านซอฟต์แวร์ที่เราพัฒนาขึ้น หลังจากนั้นซอฟต์แวร์ที่เราพัฒนาขึ้นจะทำการจำลองการทำงานและแสดงผลการจำลองการทำงานในรูปแบบของกราฟแสดงประสิทธิภาพของระบบที่ผู้ใช้งานต้องการ เราเรียกซอฟต์แวร์ที่เราพัฒนาขึ้นว่า ส่วนติดต่อผู้ใช้งานสำหรับซอฟต์แวร์จำลองการทำงานคลาวด์ซิมหรือยูไอฟอร์คลาวด์ซิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVELOPMENT OF USER INTERFACE FOR CLOUDSIM SIMULATION SOFTWARE

Ms. Keeta	Atiarnon	53010170
Mr. Warut	Yantarattanawan	53011438
Asst. Prof. Dr. Sakchai	Thipchaksurat	Advisor

Academic Year 2013

ABSTRACT

CloudSim is a simulation software for cloud computing. Several researchers have used CloudSim as a tool for evaluation the system performances. CloudSim is developed by using Java language on the window platform. For using CloudSim, the knowledge of Java programming is needed for the user. Before running CloudSim, user has to set some parameters which are used for simulating only one situation limited by simulation time. In other to evaluate the system performance, we have to simulate the several situations. Therefore, the user has to repeat the same process again starting from setting parameter which may increase the time for obtaining the simulation results of all situation and error during setting parameter. We consider that the usage of CloudSim may be more easily if the user can do everything explained above through the user interface and this is the motivation of our project. In this project, we develop the user interface for CloudSim simulation software called UI4CloudSim. The user can set the parameters of all situations and select the performance metrics through the user interface. Then, UI4CloudSim will run CloudSim automatically. The simulation results of selected performance metrics will be shown in the form of graph.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาจากอาจารย์ศักดิ์ชัย ทิพย์จักษุรัตน์ อาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำเสนอแนะ แนวคิด ตลอดจนแก้ไขข้อบกพร่องต่างๆมาโดยตลอดจนโครงงานเล่มนี้เสร็จสมบูรณ์ ผู้ศึกษาขอกราบขอบพระคุณเป็นอย่างสูง

ขอบคุณพี่ๆ ที่ช่วยให้คำแนะนำและช่วยสละเวลามาให้คำปรึกษาในการทำโครงงาน

ขอบคุณเพื่อนๆ ที่ช่วยให้คำแนะนำดีๆ ในด้านความคิดสร้างสรรค์ และคอยเป็นกำลังใจในการทำงานโดยเสมอมา

ขอกราบขอบพระคุณคุณพ่อ คุณแม่และผู้ปกครองที่ให้คำปรึกษาในเรื่องต่างๆ รวมทั้งเป็นกำลังใจที่ดีเสมอมา

สุดท้ายขอขอบพระคุณทุกๆ ท่านที่มีส่วนช่วยให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดีที่มิได้กล่าวถึงมา ณ ที่นี้ด้วย

นางสาวคีตา อติอานนท์

นายวรุตม์ ยันตรัตนะวรรณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VIII
สารบัญรูป IX

บทที่ 1 บทนำ 1
1.1 ความสำคัญและที่มาของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 2
1.4 วิธีการดำเนินการ 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 ส่วนประกอบของปริญญานิพนธ์ 4

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 5
2.1 ทฤษฎีการประมวลผลแบบกลุ่มเมฆเบื้องต้น 5
2.1.1 ประวัติความเป็นมาของการประมวลผลแบบกลุ่มเมฆ 5
2.1.2 ความหมายของการประมวลผลแบบกลุ่มเมฆ 6
2.1.3 หลักการของการประมวลผลแบบกลุ่มเมฆ 7
2.1.4 โมเดลของการประมวลผลแบบกลุ่มเมฆ 8
2.1.5 รูปแบบการใช้งานของการประมวลผลแบบกลุ่มเมฆ 10
2.1.6 สถาปัตยกรรมของการประมวลผลแบบกลุ่มเมฆ 11
2.1.7 ลักษณะเด่นของการประมวลผลแบบกลุ่มเมฆ 12
2.1.8 การใช้งานการประมวลผลแบบกลุ่มเมฆในปัจจุบัน 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2 หลักการทำงานพื้นฐานของคลาวด์ซิม 14
2.2.1 ประวัติความเป็นมาของคลาวด์ซิม 14
2.2.2 ความหมายของคลาวด์ซิม 15
2.2.3 สถาปัตยกรรมของคลาวด์ซิม 15
2.2.4 หลักการทำงานของคลาวด์ซิม 18
2.2.5 องค์ประกอบของคลาวด์ซิม 22
2.2.6 โมเดลของคลาวด์ซิม 24
2.2.7 ข้อดีของคลาวด์ซิม 27
2.2.8 ข้อเสียของคลาวด์ซิม 28
2.3 ใช้โปรแกรม Netbeans IDE 7.3.1 28
2.3.1 ข้อดีของเน็ตบีนส์ 29
2.4 โปรแกรมภาษาจาวา 29
2.4.1 ประวัติความเป็นมาของจาวา 29
2.4.2 องค์ประกอบของจาวา 30
2.4.3 จาวาแพลตฟอร์ม 31
2.4.4 ข้อดีของภาษาจาวา 32
2.4.5 ข้อเสียของภาษาจาวา 32
2.5 การจำลองสถานการณ์ 33
2.5.1 หลักการทำงานของการจำลองสถานการณ์ 33
2.5.2 ประเภทของสถานการณ์จำลอง 34
บทที่ 3 การออกแบบและพัฒนาซอฟต์แวร์ 35
3.1 แนวคิดในการพัฒนา 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2 เครื่องมือที่ใช้ในการพัฒนา 39
3.2.1 ระบบปฏิบัติการ 39
3.2.2 ภาษาโปรแกรมที่ใช้ 39
3.2.3 เครื่องมือพัฒนาซอฟต์แวร์ (Integrated Development Environment) 39
3.2.4 JRE (Java Runtime Environment) 40
3.2.5 JDK (Java Development Kit) 40
3.3 รายละเอียดของซอฟต์แวร์เชิงเทคนิค (Software Specification) 40
3.3.1 การสร้างส่วนติดต่อกับผู้ใช้แบบกราฟิก 40
3.3.2 การแสดงโครงสร้างของคลาวด์ (Topology) 41
3.3.3 การแสดงกราฟวัดประสิทธิภาพ 42
3.4 โครงสร้างของซอฟต์แวร์ 46
3.5 แผนภาพ UML (Unified Modeling Language Diagram) 47
3.5.1 Use Case Diagram 47
3.5.2 Class Diagram 48
3.6 พารามิเตอร์ที่ใช้ในการทดลอง 49

บทที่ 4 การทดลองและผลการทดลอง 51
4.1 การทดลองตั้งค่าพารามิเตอร์และองค์ประกอบภายในคลาวด์ 51
4.2 การแสดงกราฟวัดประสิทธิภาพ 57
4.3 รายงานสรุปผลการจำลอง 63
4.4 คู่มือการใช้งาน 67

บทที่ 5 บทสรุปและข้อเสนอแนะ 72
5.1 บทสรุปของโครงงาน 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข 72
5.3 แนวทางในการพัฒนาต่อ 73

บรรนาณุกรม 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 1.1 วิธีดำเนินการในภาคเรียนที่ 1 2
ตารางที่ 1.2 วิธีดำเนินการในภาคเรียนที่ 2 3
ตารางที่ 2.1 แสดงจำนวนดาต้าเซ็นเตอร์ของแต่ละบริษัท 13
ตารางที่ 3.1 แสดงผลการจำลองที่ใช้ Resource Allocation Policy แบบ Time Shared 43
ตารางที่ 3.2 แสดงผลการจำลองที่ใช้ Resource Allocation Policy แบบ Space Shared 43
ตารางที่ 3.1 กำหนดพารามิเตอร์สำหรับโบรกเกอร์ 49
ตารางที่ 3.2 กำหนดพารามิเตอร์สำหรับดาต้าเซ็นเตอร์ 49
ตารางที่ 3.3 กำหนดพารามิเตอร์สำหรับโฮสต์ 50
ตารางที่ 3.4 กำหนดพารามิเตอร์สำหรับเครื่องเสมือน 50
ตารางที่ 3.5 กำหนดพารามิเตอร์สำหรับงาน 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

รูปที่ 2.1 ภาพรวมของการประมวลผลแบบกลุ่มเมฆ 7
รูปที่ 2.2 โมเดลของการประมวลผลแบบกลุ่มเมฆ 10
รูปที่ 2.3 ประเภทของการประมวลผลแบบกลุ่มเมฆ 11
รูปที่ 2.4 สถาปัตยกรรมการประมวลผลแบบกลุ่มเมฆ 12
รูปที่ 2.5 เลเยอร์ของสถาปัตยกรรมการประมวลผลแบบกลุ่มเมฆ 16
รูปที่ 2.6 CloudSim Flow diagram 19
รูปที่ 2.7 ผลลัพธ์ของการจำลอง 21
รูปที่ 2.8 CloudSim class design diagram 22
รูปที่ 2.9 ผลของการดำเนินงานที่ใช้นโยบายแตกต่างกัน 26
รูปที่ 2.10 NetBeans IDE startup 29
รูปที่ 2.11องค์ประกอบภาษาจาวา 30
รูปที่ 2.12 ลำดับขั้นตอนการทำงานของจาวา 31
รูปที่ 3.1 ขั้นตอนการทำงานของคลาวด์ซิมและส่วนที่เกี่ยวข้อง 36
รูปที่ 3.2 ขั้นตอนการทำงานของยูไอฟอร์คลาวด์ซิมและส่วนที่เกี่ยวข้อง 38
รูปที่ 3.3 แสดงส่วนติดต่อผู้ใช้แบบกราฟิกของจาวา 41
รูปที่ 3.4 แสดงโทพอโลยีของคลาวด์ 41
รูปที่ 3.5 แสดงการจัดการแบบ First Come First Served 42
รูปที่ 3.6 แสดงการจัดการแบบ Round Robin 42
รูปที่ 3.7 กราฟแสดง Resource Allocation Policy แบบ Space Shared 44
รูปที่ 3.8 กราฟแสดง Resource Allocation Policy แบบ Time Shared 44
รูปที่ 3.9 โครงสร้างและการทำงานของยูไอฟอร์คลาวด์ซิม (UI4CloudSim) 46
รูปที่ 3.10 แผนภาพ Class Diagram ของยูไอฟอร์คลาวด์ซิม (UI4CloudSim) 49
รูปที่ 4.1 หน้าต่างหลักของซอฟต์แวร์ 51
รูปที่ 4.2 หน้าต่างสำหรับผู้ใช้ทำการตั้งค่าตัวแปร 52
รูปที่ 4.3 แสดงการตั้งค่าโบรกเกอร์ 52
รูปที่ 4.4 แสดงทอพอโลยีประกอบด้วย 2 ดาต้าเซ็นเตอร์ และ 3 ผู้ใช้งาน 53
รูปที่ 4.5 แสดงการตั้งค่าดาต้าเซ็นเตอร์ 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 4.6 แสดงทอพอโลยีเพิ่มเติมในส่วนของโฮสต์ 54
รูปที่ 4.7 แสดงการตั้งค่าโฮสต์ 55
รูปที่ 4.8 แสดงทอพอโลยีเพิ่มเติมในส่วนของเครื่องเสมือน 56
รูปที่ 4.9 แสดงการตั้งค่าเครื่องเสมือน 56
รูปที่ 4.10 แสดงการตั้งค่างานและแกนของกราฟ 57
รูปที่ 4.11 แสดงการประมวลผลการจำลอง 58
รูปที่ 4.12 แสดงการเลือกกราฟวัดประสิทธิภาพ 58
รูปที่ 4.13 แสดงกราฟ Makespan Time – Scheduling Algorithm 59
รูปที่ 4.14 แสดงกราฟ Makespan Time – Resource Allocation (Task: Space-Shared) 60
รูปที่ 4.15 แสดงกราฟ Makespan Time – Resource Allocation (Task: Time-Shared) 60
รูปที่ 4.16 แสดงกราฟ Load Balancing (Task: Space-Shared) 61
รูปที่ 4.17 แสดงกราฟ Load Balancing (Task: Time-Shared) 61
รูปที่ 4.18 แสดงกราฟ Load Balancing-VM ของ 1000 งาน 62
รูปที่ 4.19 แสดงกราฟ Execution Time (Task: Space-Shared) 62
รูปที่ 4.20 แสดงกราฟ Execution Time (Task: Time-Shared) 63
รูปที่ 4.21 แสดง Topology ของคลาวด์ 64
รูปที่ 4.22 แสดงตารางพารามิเตอร์ 64
รูปที่ 4.23 แสดงกราฟวัดประสิทธิภาพ 65
รูปที่ 4.24 แสดงหน้าต่างบันทึกรายงานสรุปผล 65
รูปที่ 4.25 แสดงรายงานสรุปผลเป็นไฟล์ PDF 66
รูปที่ 4.26 แสดงขั้นตอนที่ 1 67
รูปที่ 4.27 แสดงขั้นตอนที่ 2 67
รูปที่ 4.28 แสดงขั้นตอนที่ 3 และ 4 68
รูปที่ 4.29 แสดงขั้นตอนที่ 5 และ 6 68
รูปที่ 4.30 แสดงขั้นตอนที่ 7 และ 8 69
รูปที่ 4.31 แสดงขั้นตอนที่ 9 และ 10 69
รูปที่ 4.32 แสดงขั้นตอนที่ 11 และ 12 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 4.33 แสดงขั้นตอนที่ 13 และ 14 70
รูปที่ 4.34 แสดงขั้นตอนที่ 15 และ 16 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

คลาวด์ซิม (CloudSim) เป็นซอฟต์แวร์จำลองการทำงานของการประมวลผลแบบกลุ่มเมฆที่มีประสิทธิภาพ สามารถจำลองการทำงานขององค์ประกอบต่างๆในคลาวด์ซิม ไม่ว่าจะเป็นระบบงาน (Cloudlet) , ดาต้าเซ็นเตอร์ (Datacenter), โบรกเกอร์ (Broker), เครื่องเสมือน (Virtual Machine), และ โฮสต์ (Host) ตลอดจนการรายงานสรุปผลการจำลอง นอกจากนี้คลาวด์ซิมยังสามารถดาวน์โหลดมาใช้งานได้โดยไม่เสียค่าใช้จ่าย แต่เนื่องจากคลาวด์ซิมเป็นซอฟต์แวร์ที่ค่อนข้างมีความซับซ้อนในการใช้งานและถูกพัฒนาขึ้นจากภาษาจาวา ทำให้การใช้งานคลาวด์ซิมต้องเขียนด้วยภาษาจาวา ซึ่งผู้ใช้ต้องมีความรู้เกี่ยวกับภาษาจาวาด้วย ดังนั้นเพื่อให้ผู้ใช้สามารถศึกษาการจำลองการทำงานของคลาวด์ได้ง่ายและสะดวกยิ่งขึ้น โดยที่ผู้ใช้ไม่จำเป็นต้องมีพื้นฐานมาก่อน ซึ่งผู้ใช้สามารถเข้าใจและสั่งงานจำลองการทำงานได้โดยใช้ส่วนติดต่อผู้ใช้ที่เราพัฒนาขึ้นและสามารถดูผลการทำงานของการจำลองระบบคลาวด์ในรูปแบบของกราฟและรายงานสรุปผลการจำลองได้

เนื่องจากคลาวด์ซิมเป็นซอฟต์แวร์จำลองการทำงานที่ใช้ในด้านการประมวลแบบกลุ่มเมฆเป็นเครื่องมือสำหรับนักวิจัยที่ใช้ในการพัฒนา จึงเกิดแนวคิดในการพัฒนาซอฟต์แวร์คลาวด์ซิมสำหรับงานที่ไม่ใช่การวิจัย แต่เหมาะสำหรับนักศึกษาหรือบุคคลที่สนใจศึกษาคลาวด์ซิม เช่น นำไปประกอบการเรียนการสอนเรื่องการประมวลผลแบบกลุ่มเมฆ หรือประกอบการตัดสินใจในการลงทุนเชิงธุรกิจด้านคลาวด์

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อให้ผู้ใช้งานสามารถใช้คลาวด์ซิมได้ง่ายและสะดวก โดยจะใช้งานผ่านส่วนติดต่อผู้ใช้แบบกราฟิก (Graphic User Interface) ซึ่งเป็นซอฟต์แวร์ที่ทางผู้จัดทำได้พัฒนาขึ้น
2) เพื่อให้ผู้ใช้งานสามารถดูผลประสิทธิภาพของคลาวด์ในรูปแบบของกราฟต่างๆได้ ทำให้สามารถวิเคราะห์การทำงานของคลาวด์ได้ง่ายขึ้น
3) เป็นการศึกษาการทำงานของคลาวด์ซิมเพื่อนำมาใช้ในงานวิจัย เนื่องจากการทำงานวิจัยด้านคลาวด์จำเป็นต้องมีการใช้เครื่องมือจำลองการทำงานของคลาวด์ ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่จะต้องเข้าใจการทำงานของซอฟต์แวร์ ช่วยให้ผู้ที่ทำงานวิจัยทำสามารถลดระยะเวลาในส่วนนี้ได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ใช้ประกอบการตัดสินใจการลงทุนด้านคลาวด์ในเชิงธุรกิจ โดยไม่ต้องเสียค่าใช้จ่ายก่อนการลงทุนและไม่สิ้นเปลืองเวลาในการศึกษาโครงสร้างคลาวด์
5) เพื่อให้ผู้ใช้ไม่ต้องกังวลเกี่ยวกับการทำงานของซอฟต์แวร์คลาวด์ซิม เช่น การตั้งค่าพารามิเตอร์ หรือ การทำงานของฟังก์ชันต่างๆ มีผลการทดลองที่เชื่อถือได้

1.3 ขอบเขตของโครงงาน

1) ศึกษาการทำงานของระบบการประมวลผลแบบกลุ่มเมฆ รวมถึงนโยบายการเข้าใช้พื้นที่เพื่อจัดสรรทรัพยากร
2) สามารถจำลองการทำงานของคลาวด์ในกรณีศึกษาด้านนโยบายการเข้าใช้พื้นที่เพื่อจัดสรรทรัพยากรได้
3) สามารถแสดงประสิทธิภาพการทำงานของคลาวด์ในรูปแบบของกราฟ Makespan Time (Scheduling Policy), Makespan Time (Resource Allocation Policy), Load Balancing และ Execution Time ได้
4) แสดงผลการจำลองเป็นรายงาน ประกอบไปด้วยค่าตัวแปรที่ใช้ในการจำลองระบบคลาวด์ทอพอโลยีของคลาวด์ และประสิทธิภาพของการจำลองในรูปแบบของกราฟ
5) กำหนดให้ดาต้าเซ็นเตอร์, โฮสต์ และเครื่องเสมือน มีคุณสมบัติด้านฮาร์ดแวร์และการตั้งค่าที่เหมือนกัน
6) ซอฟต์แวร์ถูกพัฒนาโดยโปรแกรม NetBeans IDE 7.3.1 และสามารถทำงานได้บนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 7 (Microsoft Windows 7)

1.4 วิธีการดำเนินการ

เพื่อให้โครงงานนี้ดำเนินไปอย่างราบรื่น กระชับ และเสร็จสิ้นตามกำหนด จึงได้วางแผนการทำงานของแต่ละภาคเรียนดังตารางที่ 1.1 และ 1.2

ตารางที่ 1.1 วิธีดำเนินการในภาคเรียนที่ 1

ลำดับ	การดำเนินงาน
1	ศึกษาทฤษฎีของการประมวลผลแบบกลุ่มเมฆ
2	ศึกษาทฤษฎีของคลาวด์ซิม
	- หลักการทำงานของคลาวด์ซิม
	- การใช้งานไลบารี่, ฟังก์ชัน และตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	การดำเนินงาน
3	ศึกษางานวิจัยที่เกี่ยวข้องกับคลาวด์ซิม
4	ศึกษาการเขียนโปรแกรมภาษาจาวา
5	วิเคราะห์ฟังก์ชันการทำงานภายในของคลาวด์ซิม เพื่อนำมาประยุกต์ใช้กับซอฟต์แวร์ที่จะพัฒนา
6	ออกแบบซอฟต์แวร์ยูไอฟอร์คลาวด์ซิม (UI4CloudSim)

ตารางที่ 1.2 วิธีดำเนินการในภาคเรียนที่ 2

ลำดับ	การดำเนินงาน
1	สร้างส่วนติดต่อผู้ใช้แบบกราฟิก
2	สร้างส่วนคำนวณกราฟทั้ง 4 แบบ
3	สร้าง Topology ที่ได้จากรับค่าที่ผู้ใช้กำหนด
4	สร้างฟังก์ชันการทำงานภายในซอฟต์แวร์
	- การรับค่าจากผู้ใช้ใส่ในตัวแปร
	- การแสดงผลเป็นกราฟและรายงาน
	- การจัดสรรงานให้อุปกรณ์ต่างๆ
5	ตรวจสอบการทำงานของซอฟต์แวร์ที่พัฒนาและแก้ไขข้อผิดพลาด
6	ทำรูปเล่มรายงานและเตรียมเอกสารการนำเสนอผลงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) มีความรู้ความเข้าใจเกี่ยวกับการประมวลผลแบบกลุ่มเมฆ รวมถึงโครงสร้างภาพรวมของระบบคลาวด์ ทำให้เข้าใจเทคโนโลยีที่พัฒนาขึ้นมารองรับคลาวด์มากยิ่งขึ้น
2) มีความรู้ความเข้าใจเกี่ยวกับซอฟต์แวร์คลาวด์ซิมมากยิ่งขึ้น สามารถนำไปประยุกต์ใช้และออกแบบงานที่เกี่ยวข้องกับระบบการประมวลผลแบบกลุ่มเมฆได้
3) สามารถเขียนโปรแกรมภาษาจาวาได้และใช้ไลบารี่ต่างๆได้ดียิ่งขึ้น รวมถึงนำไปประยุกต์ใช้ในการเขียนโปรแกรมภาษาอื่นๆได้
4) เป็นแนวทางในการศึกษาต่อและค้นคว้าวิจัยในเรื่องของการประมวลผลแบบกลุ่มเมฆได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) สามารถนำซอฟต์แวร์ยูไอฟอร์คลาวด์ซิมไปพัฒนาต่อยอดได้ เพื่อให้ได้ประสิทธิภาพที่ดียิ่งขึ้น เช่น พัฒนาให้ครอบคลุมถึงด้านเครือข่าย (Network), SAN หรือ ด้านพลังงาน (Power) เป็นต้น

1.6 ส่วนประกอบของปริญญานิพนธ์

เนื้อหาของปริญญานิพนธ์ฉบับนี้ประกอบด้วย 5 บท คือ บทนำ ทฤษฎีที่เกี่ยวข้อง การออกแบบและพัฒนาซอฟต์แวร์ การทดลองและผลการทดลอง และบทสรุป โดยสามารถจำแนกรายละเอียดได้ดังนี้

บทที่ 1 บทนำ กล่าวถึง ความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ และ ประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึง ทฤษฎีการประมวลผลแบบกลุ่มเมฆเบื้องต้น หลักการทำงานพื้นฐานของคลาวด์ซิม โปรแกรม Netbeans IDE 7.3.1 โปรแกรมภาษาจาวา และการจำลองสถานการณ์

บทที่ 3 การออกแบบและพัฒนาซอฟต์แวร์ กล่าวถึง เครื่องมือที่ใช้ในการพัฒนาภาพรวมของโปรแกรม และกระบวนการทำงานของโปรแกรม

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึง การทดลองการตั้งค่าพารามิเตอร์และองค์ประกอบภายในคลาวด์, แสดงกราฟวัดประสิทธิภาพ รายงานสรุปผลการจำลองและคู่มือการใช้งาน

บทที่ 5 บทสรุป กล่าวถึง ผลที่ได้จากการทำโครงงาน ปัญหาอุปสรรคและแนวทางการแก้ไข และแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับการทำงาน ได้แก่ ทฤษฎีการประมวลผลแบบกลุ่มเมฆเบื้องต้น หลักการทำงานพื้นฐานของคลาวด์ซิม โปรแกรม Netbeans IDE 7.3.1 โปรแกรมภาษาจาวา และการจำลองสถานการณ์

2.1 ทฤษฎีการประมวลผลแบบกลุ่มเมฆเบื้องต้น

2.1.1 ประวัติความเป็นมาของการประมวลผลแบบกลุ่มเมฆ

คำว่า คลาวด์ (Cloud) นั้นมาจากการสื่อสารแบบโทรศัพท์ในยุคปี 90 ดาต้าเซอร์กิจได้เกิดขึ้น (บนเครือข่ายข้อมูลอินเทอร์เน็ต) เป็นการยากที่จะเดินสายเชื่อมต่อระหว่างปลายทางแต่ละจุด ดังนั้นผู้ให้บริการเริ่มมีบริการเครือข่ายส่วนตัวเสมือน (VPN) โดยผู้ให้บริการสามารถรับประกันแบนด์วิธเทียบเท่าที่การใช้วงจรแบบฟิกส์ในราคาที่ต่ำกว่า จากการที่อาศัยสวิตช์ทราฟฟิกนี้เองจึงเป็นการยากที่จะบอกล่วงหน้าได้ว่าข้อมูลเดินทางผ่านเส้นทางใด คำว่า "Telecom cloud" จึงถูกใช้เพื่ออธิบายเครือข่ายประเภทนี้ และการประมวลผลแบบกลุ่มเมฆก็มีหลักการค่อนข้างคล้ายกัน โดยอาศัยพื้นฐานเครื่องเสมือน (Virtual Machine) ซึ่งเกิดขึ้นหรือลดจำนวนเพื่อให้ตรงกับความต้องการของผู้ใช้

แนวคิดของการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) เมื่อ 1960 John McCarthy ได้เสนอความคิดเห็นว่าการคำนวณจะถูกจัดการให้สามารถใช้ได้อย่างสาธารณะ โดยลักษณะการแบ่งปันบริการกัน แต่ส่วนคำว่าคลาวด์นำเข้ามาใช้ในเชิงพาณิชย์ ในช่วงต้นปี 1990 คือ Asynchronous Transfer Mode หรือเครือข่ายแบบ ATM ต่อมา General Magic ออกผลิตภัณฑ์ ในปี 1995 โดยร่วมกับพันธมิตรหลายองค์กร เช่น ATT&T ก่อนที่ผู้บริโภคจะใช้บริการอินเตอร์เน็ต ในศตวรรษที่ 21 การประมวลผลแบบกลุ่มเมฆเริ่มปรากฏอย่างแพร่หลาย แต่ส่วนมากจะมุ่งไปในลักษณะการให้บริการซอฟต์แวร์ผ่านเว็บแอพพลิเคชั่น (Software as a Service: SaaS)

ในปี 1999 Salesforce.com ได้ก่อตั้งขึ้นโดย Marc Benioff และ Parker Harris ได้ใช้เทคโนโลยีหลายอย่างที่พัฒนาโดยบริษัท เช่น กูเกิ้ล (Google) และ ยาฮู! (Yahoo!) เพื่อประยุกต์ในเชิงธุรกิจ นอกจากนี้ยังให้แนวคิดของ "On demand" และ SaaS กับธุรกิจและลูกค้าที่ประสบความสำเร็จ กุญแจสำหรับการให้บริการซอฟต์แวร์ผ่านเว็บแอพพลิเคชั่น คือ ลูกค้าสามารถปรับแต่งได้โดยให้การสนับสนุนทางเทคนิคเท่าที่จำเป็น ซึ่งผู้ใช้พอใจกับความยืดหยุ่นและความเร็วที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไปดัดแปลง เผยแพร่ หรือนำไปใช้

ในช่วงต้นปี 2000 ไมโครซอฟต์ (Microsoft) ขยายแนวคิดของการให้บริการซอฟต์แวร์ผ่านเว็บแอพพลิเคชั่นจากการพัฒนาเว็บเซอร์วิส ต่อมาไอบีเอ็ม (IBM) เพิ่มแนวคิดเหล่านี้ในปี 2001 กับ Autonomic Computing Manifesto ซึ่งอธิบายถึงเทคนิคอัตโนมัติขั้นสูง เช่น self-monitoring , self-healing , self-configuring และ self-optimizing เพื่อเพิ่มประสิทธิภาพในการจัดการระบบไอทีที่ซับซ้อนโดยมี การเก็บข้อมูล (Storage) , เซิร์ฟเวอร์ (Server) , แอพพลิเคชัน (Application) และเครือข่าย (Network) , ระบบความปลอดภัย (Security) และองค์ประกอบอื่นๆที่แตกต่างกันนั้น สามารถจำลองข้ามเอ็นเตอร์ไพรส์ (Enterprise) กันได้

อเมซอน (Amazon) เริ่มมีบทบาทสำคัญในการพัฒนาการประมวลผลแบบกลุ่มเมฆโดยพัฒนาดาต้าเซ็นเตอร์ (Data Center) และสถาปัตยกรรมคลาวด์ใหม่ เพื่อปรับปรุงประสิทธิภาพ อเมซอนได้เปิดให้เข้าถึงระบบของคลาวด์ผ่านทางอเมซอน เว็บเซอร์วิส (Amazon Web Services) ในปี 2005

ในปี 2007 ของกูเกิ้ล, ไอบีเอ็ม และมหาวิทยาลัยจำนวนหนึ่งได้เริ่มใช้การประมวลผลแบบกลุ่มเมฆกันอย่างกว้างขวาง ในเดือนสิงหาคม 2008 Gartner Research พบว่า องค์กรต่างๆเริ่มวางแผนจากเดิมที่บริษัทนั้นเป็นเจ้าของฮาร์ดแวร์และซอฟต์แวร์เพื่อให้บริการ ได้มีแผนที่จะย้ายไปยังการประมวลผลแบบกลุ่มเมฆซึ่งกำลังเติบโต [1]

2.1.2 ความหมายของการประมวลผลแบบกลุ่มเมฆ

เนื่องจากมีความหลากหลายในเรื่องวิธีและแนวทางการพัฒนาการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) ดังนั้นจึงเป็นสาเหตุให้ผู้คนนิยามคำว่าการประมวลผลแบบกลุ่มเมฆแตกต่างกันไปตามแต่เทคโนโลยีที่ใช้พัฒนาหรือมุมมองของแต่ละบุคคล

บริษัท Gartner กล่าวว่า การประมวลผลแบบกลุ่มเมฆ คือแนวทางการประมวลผลที่พลังของโครงสร้างทางไอทีขนาดใหญ่ที่ขยายตัวได้ถูกนำเสนอยังลูกค้าภายนอกจำนวนมหาศาลในรูปแบบของบริการ

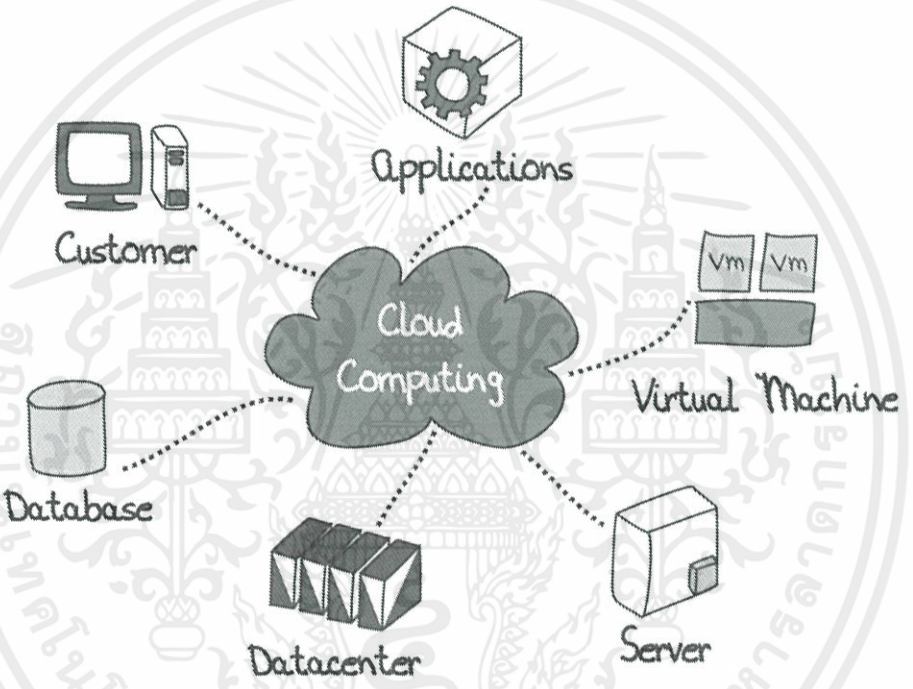
ฟอเรสเตอร์กรุ๊ป กล่าวว่า การประมวลผลแบบกลุ่มเมฆ คือกลุ่มของโครงสร้างพื้นฐานที่ถูกบริหารจัดการและขยายตัวได้อย่างมาก ซึ่งมีขีดความสามารถในการรองรับโปรแกรมประยุกต์ต่างๆของผู้ใช้และเก็บค่าบริการตามการใช้งาน

Lizhe Wang และ Gregor von Laszewski กล่าวว่า การประมวลผลแบบกลุ่มเมฆเป็นเซตของเครือข่ายที่ให้บริการโดยมีคุณลักษณะในการปรับขนาดทรัพยากร และขีดความสามรถ รวมถึงการรับประกันคุณภาพของบริการ (Quality of Service) การทำให้มีลักษณะเป็นส่วนตัวโดยมีค่าใช้จ่ายในการคำนวณตามต้องการ และสามารถเข้าถึงได้โดยง่ายจากหลายทาง

เอกสารนี้เป็นลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น การอ้างอิงหรือสำเนาส่วนหนึ่งส่วนใดต้องได้รับอนุญาตเป็นลายลักษณ์อักษรจากเจ้าของลิขสิทธิ์ก่อนทุกครั้งที่มีการนำไปใช้

วิกิพีเดีย นิยามว่า การประมวลผลแบบกลุ่มเมฆหมายถึงทรัพยากรสำหรับการประมวลผลที่จัดเตรียมและจัดการโดยบุคคลหรือองค์กรที่สาม โดยทรัพยากรเหล่านี้ถูกจัดเตรียมไว้ที่ Data center จากนั้นผู้ใช้สามารถเข้าไปใช้งานทรัพยากรเหล่านี้โดยการซื้อหรือเช่าได้ตามต้องการ [2]

จากคำนิยามข้างต้นสามารถสรุปได้ว่า การประมวลผลแบบกลุ่มเมฆ คือการประมวลผลที่อิงกับความต้องการของผู้ใช้ โดยระบบต้องจัดสรรทรัพยากรและบริการให้ตรงกับคำร้องขอ รวมถึงเสนอบริการที่เหมาะสมกับผู้ใช้ได้ตลอดเวลา



รูปที่ 2.1 ภาพรวมของการประมวลผลแบบกลุ่มเมฆ

2.1.3 หลักการของการประมวลผลแบบกลุ่มเมฆ

หลักการของการประมวลผลแบบกลุ่มเมฆ (Cloud Computing Concept) คือการย้ายการประมวลผลบนคอมพิวเตอร์ส่วนบุคคล (Personal Computer) หรือเซิร์ฟเวอร์ (Server) มาเป็นการประมวลผลบนการทำงานของกลุ่มเซิร์ฟเวอร์ที่เชื่อมโยงกันผ่านทางระบบเครือข่าย โดยกลุ่มเซิร์ฟเวอร์ เหล่านั้นได้รับการควบคุมและบริหารจัดการอย่างเป็นรูปแบบ ที่สำคัญผู้ให้บริการสามารถกำหนดระดับประสิทธิภาพ (QoS) ให้กับลูกค้าที่จะใช้บริการได้ตามเงื่อนไขสัญญา การใช้บริการที่เป็นข้อตกลงระหว่างกัน (SLA)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหลักการข้างต้น การประมวลผลแบบกลุ่มเมฆจึงจัดอยู่ในรูปแบบการกระจายการบริหารจัดการทรัพยากร (Distributed Resource Management) ซึ่งการทำงานลักษณะนี้ ทำให้การประมวลผลแบบกลุ่มเมฆสามารถจัดสรรทรัพยากรเครือข่ายทั้งหมดที่ต้องใช้ประมวลผลให้กับบริการที่มีระดับ SLA แตกต่างกันได้อย่างมีประสิทธิภาพ โดยเน้นการบริการเพื่อตอบรับความต้องการของผู้ใช้บริการที่สามารถยืดหยุ่นได้เป็นหลัก

หลักการทำงานของการประมวลผลแบบกลุ่มเมฆแบ่งออกเป็น 4 ส่วน คือ

1) ผู้ใช้บริการ (User) / ผู้ให้บริการ (Broker) ผู้ใช้บริการ หรือเว็บเซอร์วิสต่างๆ เช่น เฟสบุ๊ค (Facebook) หรือ กูเกิ้ล (Google) ซึ่งมีความต้องการขอใช้บริการประมวลผลเพื่อดำเนินกิจกรรมใดจากการให้บริการของการประมวลผลแบบกลุ่มเมฆและผู้ให้บริการ คือผู้ให้บริการธุรกิจด้านการประมวลผลแบบกลุ่มเมฆที่ดำเนินธุรกิจโดยซื้อทรัพยากรจากหลายแหล่งเพื่อนำมาเสนอขายต่อให้กับผู้ใช้บริการอีกทอดหนึ่ง ตามอัตราหรือเงื่อนไขการบริการ (QoS/SLA) ที่ตกลงกันไว้
2) เงื่อนไขบริการในการจองทรัพยากร (SLA Resources Allocator) เป็นส่วนบริหารจัดการที่สำคัญที่สุดทำหน้าที่เสมือนเป็นสะพานเชื่อมต่อระหว่างผู้ใช้บริการหรือผู้ให้บริการกับทรัพยากรภายในระบบ
3) เครื่องเสมือน (Virtual Machine : VMs) คือกลไกจำลองทรัพยากรของเซิร์ฟเวอร์ ทั้งหน่วยประมวลผล หน่วยความจำ อุปกรณ์บันทึกข้อมูล ให้กลายเป็นเซิร์ฟเวอร์เสมือน (Virtual Server) หลายเครื่อง โดยแต่ละเซิร์ฟเวอร์เสมือนสามารถติดตั้งระบบปฏิบัติการ (Operating System) ที่แตกต่างกันหรือเหมือนกันก็ได้ เพื่อให้ใช้ระบบได้เต็มประสิทธิภาพและประโยชน์สูงสุด โดยใช้ซอฟต์แวร์ทำหน้าที่เป็นสภาพแวดล้อมเสมือน (Virtualization Environment) ในการจัดทำเครื่องเสมือน
4) เครื่องจริง (Physical Machine) ทำหน้าที่ให้บริการที่แท้จริงหรือทำหน้าที่เป็นฐานข้อมูล (Database) ประมวลผล (Process) บริการด้านแอพพลิเคชั่น (Application) และระบบอื่นที่เกี่ยวข้อง โดยอาจอยู่ในสถานที่เดียวกันหรือต่างกัน รวมไปถึงคุณสมบัติของอุปกรณ์อาจแตกต่างหรือเหมือนกันก็ได้

2.1.4 โมเดลของการประมวลผลแบบกลุ่มเมฆ

โมเดลของการประมวลผลกลุ่มเมฆประกอบด้วย 3 ส่วน คือ บริการซอฟต์แวร์ บริการด้านแพลตฟอร์ม และบริการด้านโครงสร้างพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) บริการซอฟต์แวร์ (Software as a Service) Software as a Service (SaaS) หรือเรียกว่า On Demand Software คือ รูปแบบการให้บริการซอฟต์แวร์ผ่านทางอินเทอร์เน็ต คล้ายกับ

การเช่าใช้ เพียงแค่ผู้ซื้อจ่ายค่าซอฟต์แวร์ตามลักษณะการใช้งานที่ต้องการ (Pay-as-you-go) เช่น ตามจำนวนผู้ใช้และตามระยะเวลาที่ต้องการใช้ ก็สามารถเข้าใช้งานซอฟต์แวร์นั้นๆ ได้ผ่านทางเว็บบราวเซอร์ โดยไม่ต้องติดตั้งโปรแกรมลงเครื่องเหมือนการซื้อซอฟต์แวร์แบบลิขสิทธิ์ (License) ผู้ให้บริการซอฟต์แวร์ประเภทการให้บริการซอฟต์แวร์ผ่านเว็บแอพพลิเคชั่น ทำหน้าที่เหมือนเป็นโฮสต์แอพพลิเคชัน (Host Application) โดยเปิดสิทธิ์ให้ลูกค้า (End User) จากทั่วโลกเข้ามาแบ่งปันการใช้งานซอฟต์แวร์และฮาร์ดแวร์ร่วมกันผ่านทางเว็บบราวเซอร์ โดยใช้ชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) เพื่อระบุความเป็นเจ้าของซอฟต์แวร์นั้นๆ เพื่อเข้าสู่ระบบ

2) บริการด้านแพลตฟอร์ม (Platform as a Service) Platform as a Service (PaaS) เป็นการให้บริการแพลตฟอร์มที่รองรับการทำงานของแอปพลิเคชั่น โดยผู้ใช้บริการสามารถปรับใช้และจัดการได้เอง การให้บริการแพลตฟอร์มนั้นประกอบด้วยระบบปฏิบัติการ ระบบฐานข้อมูล และระบบมิดเดิ้ลแวร์ ตัวอย่างเช่น Window Server , Linux , Oracle Database เป็นต้น โดยผู้ให้บริการในรูปแบบนี้เป็นการให้บริการในลักษณะของเครื่องมือที่สนับสนุนการสร้างแอพพลิเคชันแบบครบวงจร (Software Development Lifecycle) ในรูปแบบของ Integrated solution เครื่องมือเหล่านี้ครอบคลุมตั้งแต่การออกแบบและพัฒนา (Design and Development) การทดสอบ (Testing) และการติดตั้ง (Deployment) แอพพลิเคชัน
3) บริการด้านโครงสร้างพื้นฐาน (Infrastructure as a Service) Infrastructure as a Service (IaaS) คือบริการด้านโครงสร้างพื้นฐาน ซึ่งเป็นการให้บริการฮาร์ดแวร์สำหรับเซิร์ฟเวอร์ หน่วยเก็บข้อมูล ระบบเครือข่ายและระบบรักษาความปลอดภัย ในรูปแบบการจำลอง (Virtualization) ซึ่งทำให้เราสามารถจัดสรรทรัพยากรได้แบบไดนามิก (Dynamic) เช่น การเพิ่มหรือลดขนาดของหน่วยประมวลผล (CPU) ฮาร์ดดิสก์ หรือหน่วยความจำหลัก (RAM) ของเครื่องเซิร์ฟเวอร์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 โมเดลของการประมวลผลแบบกลุ่มเมฆ

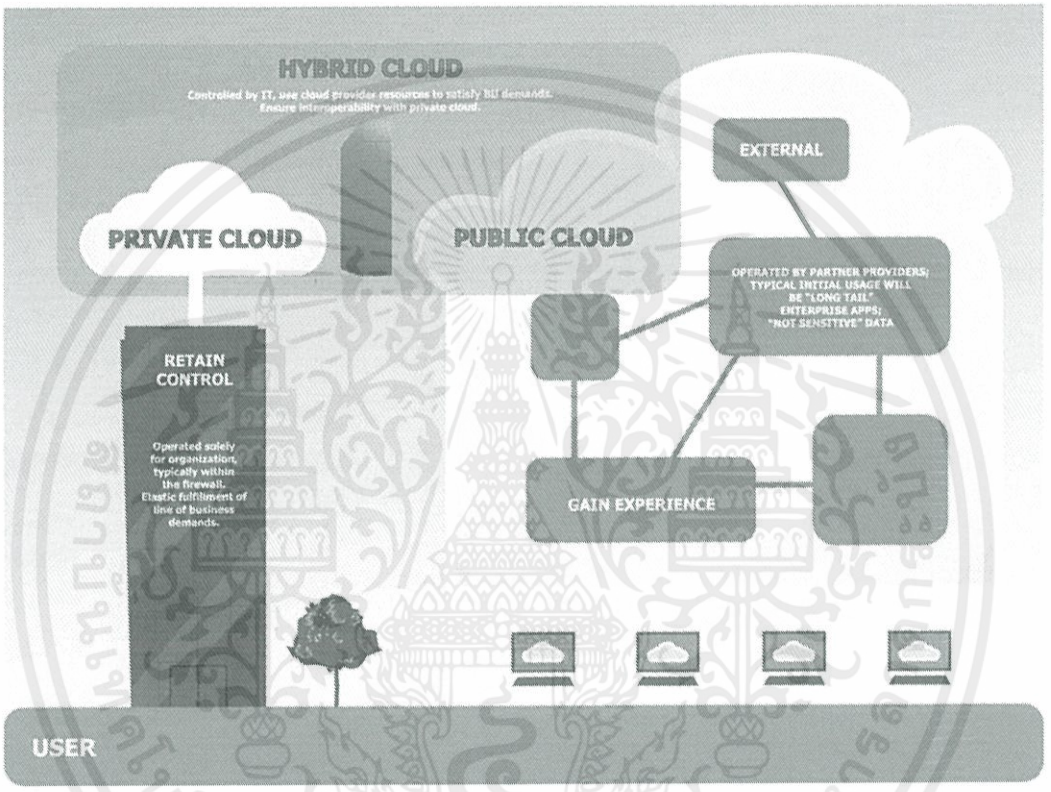
2.1.5 รูปแบบการใช้งานของการประมวลผลแบบกลุ่มเมฆ

รูปแบบการใช้งานของการประมวลผลแบบกลุ่มเมฆมี 3 รูปแบบคือ คลาวด์สาธารณะ คลาวด์ส่วนตัว และคลาวด์ลูกผสม

1) คลาวด์สาธารณะ (Public Cloud) หรือ คลาวด์ภายนอก (External Cloud) หมายถึงการประมวลผลแบบกลุ่มเมฆที่มีทรัพยากรเป็นสาธารณะ สร้างขึ้นเพื่อให้ทุกคนสามารถใช้งานได้ สามารถใช้บริการผ่านทางอินเตอร์เน็ต เว็บแอพพลิเคชัน หรือเว็บเซอร์วิส เป็นการใช้ทรัพยากรและค่าใช้จ่ายร่วมกัน (Resource and Cost Sharing) ทำให้ค่าใช้จ่ายและต้นทุนขององค์กรต่ำลง รวมถึงความสามารถในการขยายระบบ (On-Demand Scalability) องค์กรสามารถเลือกขนาดของระบบที่เหมาะสมตรงกับความต้องการในปัจจุบันและในอนาคต ซึ่งจ่ายค่าบริการตามปริมาณการใช้งานที่เกิดขึ้นจริง (Pay-as-you-go)
2) คลาวด์ส่วนตัว (Private Cloud) หรือ คลาวด์ภายใน (Internal Cloud) คือระบบที่ทำงานอยู่บนคลาวด์และบริหารจัดการโดยบริษัทเพื่อการใช้งานภายในองค์กรเท่านั้น ผู้ให้บริการและผู้ใช้สามารถควบคุมและปรับปรุงระบบความปลอดภัยได้ด้วยตนเอง ระบบคอมพิวเตอร์ที่ให้บริการคลาวด์ส่วนตัวอาจต้องติดตั้งระบบไฟล์วอลล์เพื่อป้องกันการบุกรุกจากภายนอกองค์กร เนื่องจากการใช้งานคลาวด์ส่วนตัวทำให้สามารถจัดสรรทรัพยากรภายใต้การควบคุมขององค์กรได้ ทำให้เกิดค่าใช้จ่ายที่สูงกว่าการใช้งานคลาวด์สาธารณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) คลาวด์ลูกผสม (Hybrid Cloud) คือระบบที่รวมระหว่างคลาวด์ส่วนตัวและคลาวด์สาธารณะ ผู้ให้บริการที่ให้บริการภายใต้คลาวด์ลูกผสมจะทำงานแบบอิสระ โดยระบบสามารถทำงานร่วมกันในระดับข้อมูลและระบบซอฟต์แวร์ประยุกต์ได้ ทั้งระบบที่มีการเชื่อมโยงโดยใช้เทคนิคที่เป็นมาตรฐานและเทคโนโลยีเฉพาะ



รูปที่ 2.3 ประเภทของการประมวลผลแบบกลุ่มเมฆ

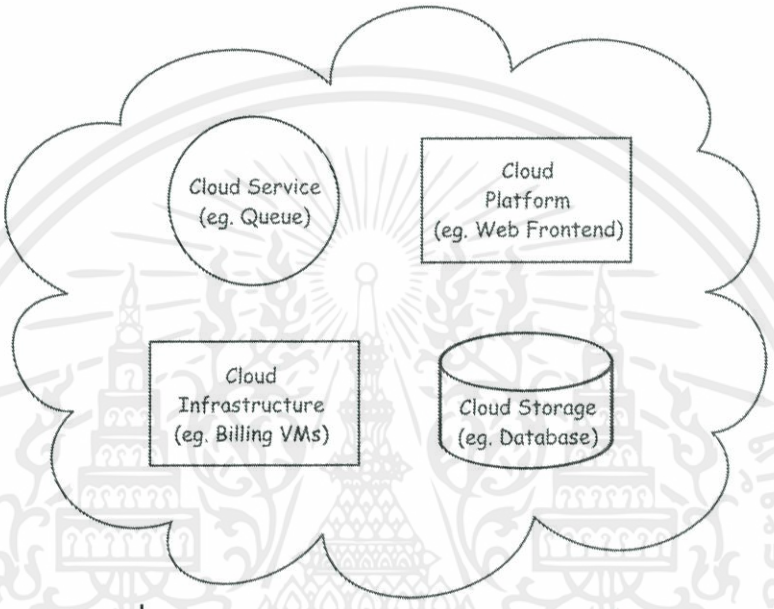
2.1.6 สถาปัตยกรรมของการประมวลผลแบบกลุ่มเมฆ

สถาปัตยกรรมเทคโนโลยีเริ่มมีการยอมรับการประมวลผลแบบกลุ่มเมฆเข้ามาในระบบ เพราะการประมวลผลแบบกลุ่มเมฆเป็นประเด็นหลักที่ทำให้ผู้บริหารหรือกลุ่มผู้ดูแลเทคโนโลยีขององค์กร เกิดการเปลี่ยนแผนการดำเนินงานและปรับพื้นฐานเทคโนโลยีเพื่อลดต้นทุนการผลิต โดยใช้เทคโนโลยีคลาวด์ (Cloud) ในการติดตั้งระบบงานใหม่หรือใช้บริการด้านการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) หรือเปลี่ยนรูปแบบธุรกิจ ปรับเปลี่ยนการลงทุนโดยการเป็นผู้ให้บริการด้านการประมวลผลแบบกลุ่มเมฆ

สถาปัตยกรรมของการประมวลผลแบบกลุ่มเมฆ คือระบบซอฟต์แวร์ที่ถูกรวมเป็นการประมวลผลแบบกลุ่มเมฆ ประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์ที่ออกแบบด้วยสถาปัตยกรรมแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาวด์ โดยทั่วไปจะทำการรวมส่วนประกอบของการประมวลผลแบบกลุ่มเมฆ เพื่อให้สามารถสื่อสารกันได้ผ่านทางเว็บเซอร์วิส สถาปัตยกรรมของการประมวลผลแบบกลุ่มเมฆอาจรวมไปถึงลูกค้า (Client) หมายถึงเว็บเบราว์เซอร์ (Web Browser) หรือซอฟต์แวร์แอพพลิเคชัน (Software Application) ที่ใช้ในการเข้าถึงแอพพลิเคชันของระบบคลาวด์



รูปที่ 2.4 สถาปัตยกรรมการประมวลผลแบบกลุ่มเมฆ

2.1.7 ลักษณะเด่นของการประมวลผลแบบกลุ่มเมฆ

1) ความคล่องตัว (Agility) มีความรวดเร็วในการใช้งาน
2) ค่าใช้จ่าย (Cost) สามารถลดค่าใช้จ่ายด้านการลงทุนในระบบไอที โดยผู้ให้บริการจะเป็นผู้ลงทุนในโครงสร้างพื้นฐานของระบบไอที และผู้ใช้ไม่จำเป็นต้องลงทุนติดตั้งหรือซื้อลิขสิทธิ์ (License) ของซอฟต์แวร์ ซึ่งจะมีค่าใช้จ่ายเพิ่มขึ้นหากองค์กรธุรกิจจะเพิ่มจำนวนพนักงานหรือต้องการปรับปรุงซอฟต์แวร์
3) อุปกรณ์และสถานที่ตั้งไม่ขึ้นต่อกัน (Device and Location Independence) ผู้ใช้สามารถเข้าถึงระบบจากสถานที่ใดก็ตาม และสามารถใช้อุปกรณ์ได้หลากหลายรูปแบบ เช่น คอมพิวเตอร์ หรือ โทรศัพท์เคลื่อนที่ เป็นต้น
4) แบ่งปันทรัพยากร (MultiTenancy) เพิ่มความสามารถในการแบ่งปันทรัพยากร เป็นการรวมศูนย์ของโครงสร้างพื้นฐานเพื่อลดต้นทุนทั้งค่าเช่าสถานที่ และค่าใช้จ่ายอื่น ๆ เพิ่มความสามารถรองรับช่วงเวลาทำงานหนัก (Peak-Load Capacity) รวมทั้งช่วยปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถและประสิทธิภาพ (Utilization and Efficiency) ของทรัพยากรเทคโนโลยีสารสนเทศ

5) ความน่าเชื่อถือ (Reliability) การมีมาตรการป้องกันระบบล่ม เพื่อให้ระบบพร้อมให้บริการตลอดเวลา มีความน่าเชื่อถือ เนื่องจากมีเครื่องแม่ข่ายและระบบสำรองข้อมูลที่มีประสิทธิภาพสูง
6) ความยืดหยุ่น (Scalability) มีความยืดหยุ่นในการเพิ่มหรือลดระบบตามความต้องการของผู้ใช้บริการ
7) ความปลอดภัย (Security) สำหรับข้อมูลและทรัพยากรของระบบ อย่างไรก็ตามยังมีความกังวลเกี่ยวกับการสูญเสียความสามารถในการกำกับดูแลการเข้าถึงและความปลอดภัยของข้อมูลที่สำคัญ
8) ความมั่นคง (Sustainability) สามารถกำกับดูแลและมีความเสถียร แต่อาจได้รับผลกระทบจากการสื่อสารบนอินเตอร์เน็ตที่ไม่มีประสิทธิภาพ หรือช่วงเวลาที่มีการใช้งานพร้อมกันจำนวนมาก

2.1.8 การใช้งานการประมวลผลแบบกลุ่มเมฆในปัจจุบัน

ในปัจจุบันมีการใช้งานการประมวลผลแบบกลุ่มเมฆกันอย่างแพร่หลายในกลุ่มธุรกิจ ทั้งการใช้งานเพื่อธุรกิจของตนเองและการเป็นผู้ให้บริการด้าน Cloud computing แก่ผู้อื่น จึงมีความหลากหลายของโครงสร้างคลาวด์เป็นอย่างมาก [3][4][5][6][7][8] แสดงดังตารางที่ 2.1

ตารางที่ 2.1 แสดงจำนวนดาต้าเซ็นเตอร์ของแต่ละบริษัท

รายชื่อบริษัท	จำนวนดาต้าเซ็นเตอร์
Microsoft	38
IBM	22
Google	12
Amazon	8
eBay	2
Twitter	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยส่วนมากในการจัดซื้อดาต้าเซ็นเตอร์, โฮสต์ หรือเครื่องเสมือน บริษัทหรือองค์กรจะจัดซื้อเป็นชุดๆ เพื่อให้ง่ายต่อการคำนวณคุณสมบัติของวัสดุอุปกรณ์และสะดวกในการตั้งค่าที่ต้องอาศัย

ความเหมือนกันของอุปกรณ์ เช่น การตั้งค่าเครือข่าย หรือ การจัดทำโหลดบาลานซ์ (Load Balance) เป็นต้น

2.2หลักการทำงานพื้นฐานของคลาวด์ซิม

2.2.1 ประวัติความเป็นมาของคลาวด์ซิม

การประมวลผลแบบกลุ่มเมฆกลายเป็นเทคโนโลยีที่ได้รับความนิยมเป็นอย่างมาก ทั้งด้านการส่งข้อมูลที่น่าเชื่อถือ การรักษาความปลอดภัย ความคงทนต่อความเสียหาย ความมีเสถียรภาพ และความยืดหยุ่นในการให้บริการ ซึ่งถูกนำเสนอในรูปแบบการให้บริการของซอฟต์แวร์ โครงสร้างพื้นฐาน และแพลตฟอร์ม (SaaS , IaaS , PaaS) โดยการให้บริการเหล่านี้อาจถูกนำเสนอแบบคลาวด์ส่วนตัว (Private Clouds) หรือในเชิงพาณิชย์เป็นรูปแบบคลาวด์สาธารณะ (Public Clouds) หรือรวมกันเป็นคลาวด์ลูกผสม (Hybrid Clouds)

เนื่องด้วยความต้องการเทคโนโลยีที่มีประสิทธิภาพเพิ่มมากขึ้น การประหยัดเวลา การทำซ้ำ และวิธีการควบคุมสำหรับการประเมินผลของอัลกอริทึม แอพพลิเคชัน และนโยบาย ก่อนจะนำมาพัฒนาเป็นคลาวด์จริง เพราะการใช้ประโยชน์จากการสร้างการจำลองที่สามารถจำกัดการทดสอบขนาดและทำการทดลองซ้ำได้ จึงเป็นทางเลือกสำหรับการทดสอบและพัฒนาวิธีการทดสอบของเทคโนโลยีคลาวด์แบบใหม่

ทางเลือกที่เหมาะสมคือการใช้ประโยชน์จากโปรแกรมจำลองสถานการณ์ ซึ่งเปิดโอกาสให้สามารถทดลองประมวลผลข้อสันนิษฐานก่อนที่จะพัฒนาในส่วนของซอฟต์แวร์จริง ในสภาพแวดล้อมที่สามารถทดลองซ้ำได้ โดยเฉพาะในกรณีของการประมวลผลแบบกลุ่มเมฆ พื้นฐานของการจำลองมีความใกล้เคียงกับข้อเสนอที่มีผลประโยชน์ที่สำคัญเหมือนที่อนุญาตให้ผู้ใช้บริการคลาวด์สามารถทดสอบบริการของพวกเขาในสภาพแวดล้อมซ้ำๆและควบคุมสภาพแวดล้อมได้โดยไม่เสียค่าใช้จ่ายเพื่อการปรับปรุงปัญหาคอขวดก่อนนำไปใช้ในคลาวด์จริง ในด้านของผู้ให้บริการ การจำลองสภาพแวดล้อมยอมให้ประมวลผลทรัพยากรได้หลายชนิดในการจำลองแต่ละครั้งภายใต้ความผันแปรของภาระงาน (Load) และราคา เช่น ช่วยให้ผู้บริการสามารถคำนวณราคาที่เหมาะสมในการเข้าใช้ทรัพยากรที่ให้ความสำคัญกับการเพิ่มกำไร ในส่วนที่ขาดหายไปของการจำลองแพลตฟอร์มผู้ใช้บริการและผู้ให้บริการคลาวด์ควรเชื่อในทฤษฎีและการประมวลผลที่เชื่อถือไม่ได้หรือทดลองเพื่อให้เข้าใจถึงบริการที่ไร้ประสิทธิภาพ

วัตถุประสงค์หลักของคลาวด์ซิม คือจัดเตรียมส่วนทั่วไปของโครงสร้างการจำลอง (Simulation Framework) และส่วนที่ขยายเพิ่มเติมที่สามารถทำโมเดล การจำลอง และการทดลองการรวมกันของโครงสร้างพื้นฐานและการให้บริการแอพพลิเคชันของการประมวลผลแบบกลุ่มเมฆโดย

เอกสารนี้เป็น... ไม่ว่ากรณี...

ใช้คลาวด์ซิม นักวิจัยและนักพัฒนาสามารถให้ความสำคัญกับรายละเอียดปัญหาของการออกแบบระบบที่พวกเขาต้องการตรวจสอบ โดยไม่ต้องกังวลเกี่ยวกับรายละเอียดในระดับล่างของพื้นฐานคลาวด์ทั้งส่วนของโครงสร้างพื้นฐานและการให้บริการ [9]

2.2.2 ความหมายของคลาวด์ซิม

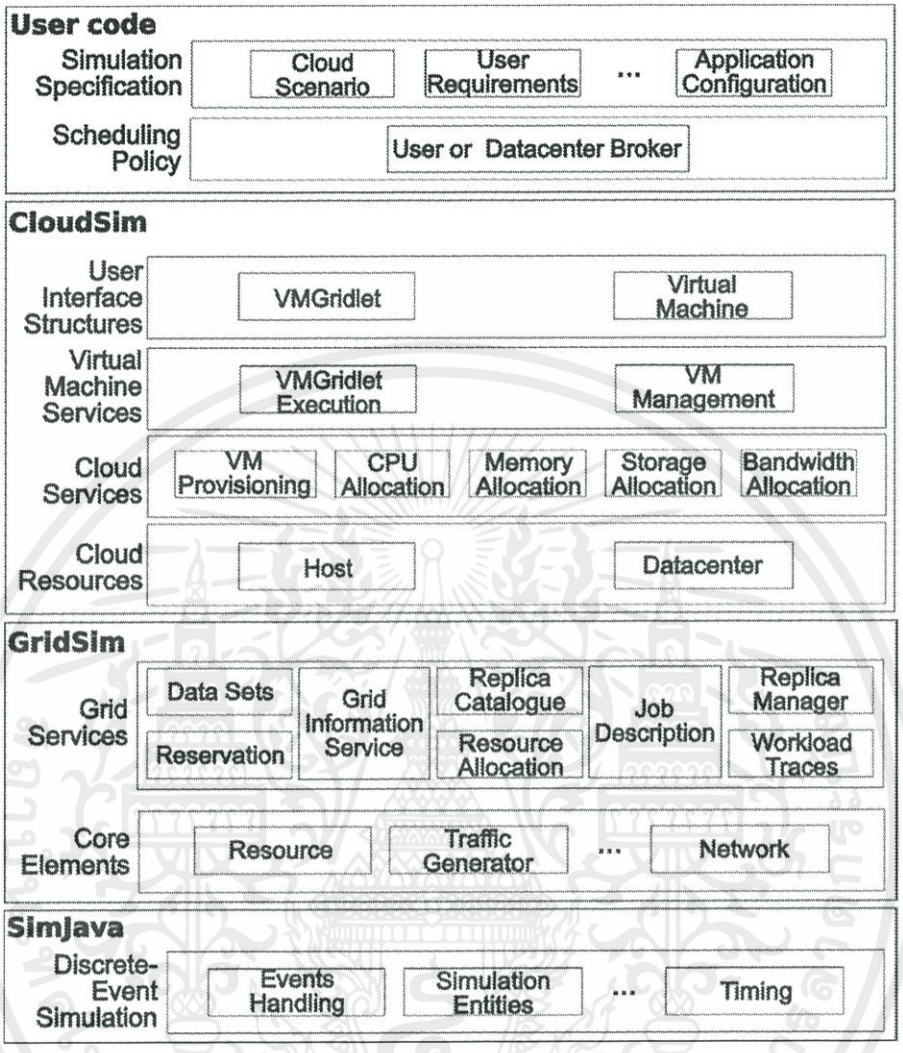
คลาวด์ซิม คือเครื่องมือที่สามารถจำลองการสร้างโมเดลและจำลองการทำงานของการประมวลผลแบบกลุ่มเมฆ ซึ่งสามารถจัดเตรียมสภาวะแวดล้อมและทรัพยากรให้กับแอพพลิเคชันได้ โปรแกรมคลาวด์ซิมมีการสนับสนุนการจำลองโมเดลที่เป็นส่วนประกอบของระบบคลาวด์ ทั้งโมเดลแบบระบบและโมเดลแบบพฤติกรรม เช่น การจำลองดาต้าเซ็นเตอร์ (Datacenters) , เครื่องเสมือน (Virtual Machines) รวมถึงนโยบายการจัดสรรทรัพยากร คลาวด์ซิมมีเทคนิคการจัดสรรทรัพยากรให้แอพพลิเคชันทั่วๆไปทำให้ใช้งานได้ง่ายและสะดวกสบายยิ่งขึ้น ในปัจจุบันการทำงานของคลาวด์ซิมสนับสนุนโมเดลและการจำลองสภาวะแวดล้อมของการประมวลผลระบบกลุ่มเมฆทั้งแบบคลาวด์เดี่ยว (Single Clouds) และสหพันธ์ของคลาวด์ (Inter-Networked Clouds) ในอนาคตคลาวด์ซิมจะมีการทำส่วนติดต่อที่สามารถปรับแต่งค่าได้ เพื่อพัฒนานโยบายและเทคนิคการจัดสรรทรัพยากรสำหรับการจองเครื่องเสมือนภายใต้สถานการณ์จำลองของสหพันธ์ของการประมวลผลแบบกลุ่มเมฆ (Inter-Networked Cloud Computing)

นักวิจัยหลายคนจากองค์กรต่างๆในประเทศสหรัฐอเมริกาใช้คลาวด์ซิมในการทดสอบการจัดสรรทรัพยากรของคลาวด์ และทดสอบการจัดการด้านพลังงานของดาต้าเซนเตอร์ให้มีประสิทธิภาพสูงสุด

2.2.3 สถาปัตยกรรมของคลาวด์ซิม

คลาวด์ซิมเป็นซอฟต์แวร์ที่ถูกพัฒนาอยู่ในระดับ (Level) ระหว่างกริดซิมและยูเซอร์โค้ด (User Code) [10] ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 เลเยอร์ของสถาปัตยกรรมการประมวลผลแบบกลุ่มเมฆ

จากรูปที่ 2.5 ได้แสดงถึงเลเยอร์ต่างๆของการพัฒนาโครงสร้างซอฟต์แวร์คลาวด์ซิม (CloudSim Software Framework) และส่วนประกอบต่างๆในเชิงสถาปัตยกรรม ในชั้นเลเยอร์สุดท้ายคือ ซิมจาวา (SimJava) เป็นตัวประมวลผล (Engine) ที่ช่วยแยกแยะเหตุการณ์ (Event) ของการจำลองที่มีส่วนช่วยในเรื่องความต้องการของฟังก์ชันหลักสำหรับโครงสร้างการจำลองในระดับสูง เช่น การเข้าคิว (Queue) และการประมวลผลของเหตุการณ์ การสร้างองค์ประกอบของระบบ การติดต่อสื่อสารระหว่างองค์ประกอบแต่ละองค์ประกอบและการจัดการลำดับในการจำลอง ในเลเยอร์ถัดไปคือ กริดซิม (GridSim) จะสนับสนุนองค์ประกอบของโปรแกรมในลำดับชั้นสูงขึ้นไป สำหรับการทำโมเดลของโครงสร้างพื้นฐานหลายๆกริด รวมทั้งเครือข่ายและประวัติการเชื่อมต่อการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานและองค์ประกอบพื้นฐานของกริด (Grid) เช่น ทรัพยากร ชุดของข้อมูล ปริมาณงานที่เหมาะสม การติดตาม และข้อมูลด้านการให้บริการ

เลเยอร์คลาวด์ซิมจะอยู่ถัดจากเลเยอร์กริดซิม ซึ่งได้จัดเตรียมเกี่ยวกับการสนับสนุนการทำโมเดลและการจำลองของสภาพแวดล้อมดาต้าเซ็นเตอร์ที่มีพื้นฐานการทำงานแบบคลาวด์เสมือน เช่น การจัดการส่วนติดต่อ (Interface) สำหรับเครื่องเสมือน หน่วยความจำ หน่วยเก็บข้อมูล และแบนด์วิธ (Bandwidth) เลเยอร์คลาวด์ซิมจะจัดการสร้างกรณีตัวอย่างและการดำเนินการของเอนติตี้ (Entity) หลัก ได้แก่ เครื่องเสมือน โฮสต์ ดาต้าเซ็นเตอร์ และแอพพลิเคชัน ระหว่างที่อยู่ในช่วงเวลาของการจำลอง โดยเลเยอร์นี้สามารถทำการสร้างกรณีตัวอย่างพร้อมกันและจัดการโครงสร้างพื้นฐานของคลาวด์ขนาดใหญ่ที่ประกอบไปด้วยองค์ประกอบของระบบจำนวนมากได้ เช่น การจัดสรรเครื่องเสมือนให้กับโฮสต์ขึ้นอยู่กับความต้องการของผู้ใช้ การจัดการดำเนินงานของแอพพลิเคชัน และการเฝ้าสังเกตแบบไดนามิก (Dynamic Monitoring) จะถูกดูแลในเลเยอร์คลาวด์ซิม ผู้ให้บริการคลาวด์ที่ต้องการทราบประสิทธิภาพของนโยบายที่แตกต่างกันในการจองโฮสต์ จะต้องพัฒนาอัลกอริทึมของตนในเลเยอร์นี้ โดยการเขียนโปรแกรมขยายในส่วนการทำงานหลักของการกำหนดเครื่องเสมือน ทำให้เห็นว่าเครื่องเสมือนต่างๆจะถูกจองโดยโฮสต์ที่แข่งขันกันได้อย่างไรในคลาวด์ โฮสต์สามารถแบ่งปันเครื่องเสมือนพร้อมๆกันได้หลายเครื่องได้อย่างไร ซึ่งเครื่องเสมือนเหล่านั้นจัดการแอพพลิเคชันโดยขึ้นอยู่กับเงื่อนไขความต้องการของผู้ใช้

ส่วนบนสุดของเลเยอร์ คือ ยูเซอร์โค้ด (User Code) สามารถตั้งค่าเกี่ยวกับฟังก์ชันการทำงานของโฮสต์ (จำนวนเครื่อง รายละเอียดของโฮสต์ และอื่นๆ) แอพพลิเคชัน (จำนวนของงานและความต้องการของแอพพลิเคชัน) เครื่องเสมือน จำนวนของผู้ใช้งาน ประเภทแอพพลิเคชันของผู้ใช้ และนโยบายการจัดการของผู้ให้บริการ โดยนักพัฒนาแอพพลิเคชันของคลาวด์สามารถสร้างการรวมกันระหว่างการกระจายความต้องการของผู้ใช้ การตั้งค่าแอพพลิเคชัน และแผนการทำงานของคลาวด์ในเลเยอร์นี้ รวมถึงสามารถแสดงการทดสอบที่อิงกับการตั้งค่าโดยผู้ใช้ได้

การประมวลผลแบบกลุ่มเมฆได้พัฒนาขอบเขตของการค้นคว้าวิจัย ให้มีเครื่องมือและวิธีการที่สามารถแก้ปัญหาโครงสร้างพื้นฐานและแอพพลิเคชันที่มีระดับซับซ้อนได้อย่างมีประสิทธิภาพ ดังนั้นในอนาคตจะมีความพยายามในการทำวิจัยทั้งในทางทฤษฎีและอุตสาหกรรมด้านอัลกอริทึม นโยบาย เบนช์มาร์คกิ้งแอพพลิเคชัน ที่อาศัยสภาพแวดล้อมของการดำเนินงาน นักวิจัยจะสามารถทำการทดลองที่สร้างขึ้นจากรายละเอียดของเหตุการณ์ และทำการตั้งค่า โดยใช้ส่วนขยายฟังก์ชันการทำงานพื้นฐานที่มีในคลาวด์ซิม

สาเหตุที่สร้างคลาวด์ซิมและพัฒนาออกมาเพื่อนำไลบรารีการจำลองที่มีอยู่มาใช้ใหม่ เพราะการประยุกต์ใช้งาน การทดสอบ ไลบารีที่สมบูรณ์ สามารถดูแลความต้องการในระดับล่างของระบบได้ ตัวอย่างเช่น สามารถหลีกเลี่ยงการพัฒนาซ้ำของการจัดการเหตุการณ์และการส่งผ่านข้อความใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

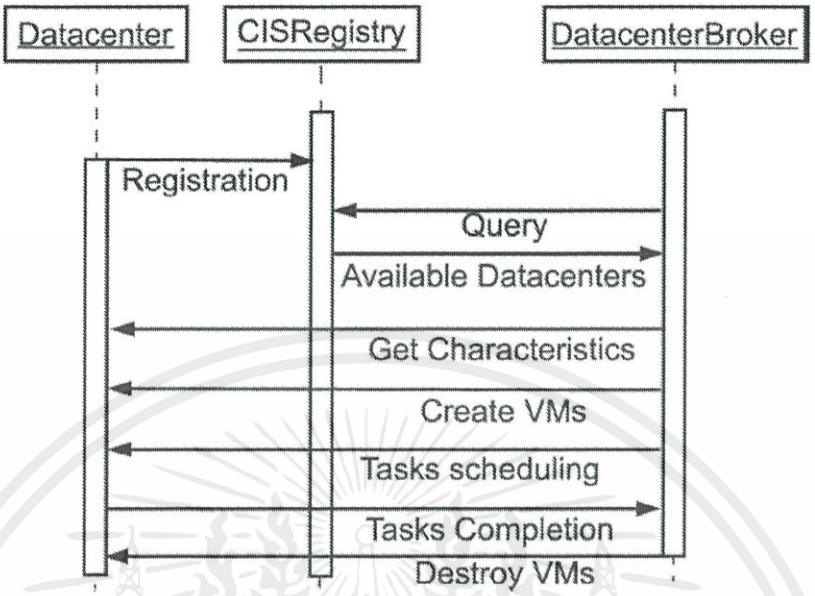
องค์ประกอบนี้ ทำให้ประหยัดเวลาในการทำงานด้านวิศวกรรมซอฟต์แวร์ (Software Engineering) และการทำการทดสอบ ขณะเดียวกันการใช้โครงสร้างกริดซิมช่วยให้สามารถนำการดำเนินงานกลับมาใช้ใหม่ได้ ซึ่งจะเป็นการดำเนินงานของเครือข่าย ข้อมูลการให้บริการ ไฟล์ ผู้ใช้ และทรัพยากร เมื่อนำซิมจาวา (SimJava) และกริดจาวา (GridJava) มาใช้ประโยชน์ในการจัดการทรัพยากรกริด (Grid) นักวิจัยหลายคนตรวจพบและทำการปรับปรุงข้อบกพร่องหลายจุดของการจำลองโดยใช้การตรวจสอบโครงสร้างซ้ำ ทำให้สามารถมุ่งเน้นไปที่ความสำคัญในด้านของระบบที่เกี่ยวข้องกับการประมวลผลกลุ่มเมฆ

2.2.4 หลักการทำงานของคลาวด์ซิม

คลาวด์ซิมสนับสนุนโมเดลและการจำลองของโครงสร้างต่างๆ เช่น ดาต้าเซ็นเตอร์ในการประมวลผลกลุ่มเมฆที่มีขนาดใหญ่ โฮสต์เซิร์ฟเวอร์เสมือนที่สามารถปรับแต่งนโยบายสำหรับการจัดสรรทรัพยากรของโฮสต์ไปยังเครื่องเสมือน เครือข่ายทอพอโลยี (Network Topology) ของดาต้าเซ็นเตอร์และการส่งข้อความผ่านในแอพพลิเคชัน การรวมกันของคลาวด์ (Federated Clouds) การเพิ่มองค์ประกอบของการจำลองแบบไดนามิคสามารถหยุดการจำลองและย้อนกลับไปขณะทำการจำลองได้ การสร้างนโยบายขึ้นมาใช้งานเองโดยผู้ใช้ สำหรับการจองโฮสต์ให้เครื่องเสมือนและนโยบายสำหรับการจองทรัพยากรของโฮสต์เพื่อให้เครื่องเสมือนนำไปใช้งาน

การติดต่อสื่อสารระหว่างเอนติตี้ (Entity) ในรูปที่ 2.6 แสดงให้เห็นถึงการไหลของการสื่อสารระหว่างเอนติตี้ในคลาวด์ซิม [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 CloudSim Flow diagram

ในการเริ่มต้นของการจำลอง แต่ละดาต้าเซ็นเตอร์เอนติตี้จะต้องลงทะเบียนตนเองกับ Cloud Information Service (CIS) Registry โดย CIS จะจัดเตรียมการบริการฐานข้อมูลในระดับ match-making สำหรับแมปการร้องขอของผู้ใช้ไปยังผู้ให้บริการคลาวด์ที่เหมาะสม โบรกเกอร์จะทำหน้าที่เป็นตัวแทนของผู้ใช้ เพื่อปรึกษา CIS service เกี่ยวกับลำดับของคลาวด์ผู้ซึ่งเสนอการให้บริการโครงสร้างพื้นฐานที่ตรงกับความต้องการแอพพลิเคชันของผู้ใช้ ในกรณีที่การจับคู่เกิดขึ้น โบรกเกอร์จะปรับใช้แอพพลิเคชันกับคลาวด์ที่ถูกแนะนำมาจาก CIS การไหลของการสื่อสารที่อธิบายมาทั้งหมดนั้นเกี่ยวข้องกับพื้นฐานการไหลในการทดลองการจำลอง สามารถกำหนดที่นโยบายที่เปลี่ยนแปลงการไหลของข้อมูลได้

การทำงานโดยรวมของคลาวด์ซิม ผู้ใช้ทำการจำลองการทำงานระบบคลาวด์ (Cloud Systems) โดยเรียกใช้งานซอฟต์แวร์ผ่านโปรแกรมที่รองรับการเขียนด้วยภาษาจาวา (Java Language) ผู้ใช้กำหนดชุดคำสั่ง (Library) ในการจำลองสภาพแวดล้อมและการทำงานพื้นฐานของระบบคลาวด์ จากนั้นตั้งค่าตัวแปรต่างๆ ทั้งในส่วนของฮาร์ดแวร์ นโยบายที่ใช้ควบคุมการทำงาน จำนวนงาน และองค์ประกอบพื้นฐานอื่นๆที่มีผลต่อการจำลองการทำงาน จากนั้นเรียกใช้คำสั่งดำเนินการจำลอง [10]

โปรแกรมที่ 2.1 การเรียกใช้ชุดคำสั่งของคลาวด์ซิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
```

```
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.DatacenterCharacteristics;
import org.cloudbus.cloudsim.Host;
import org.cloudbus.cloudsim.Log;
import org.cloudbus.cloudsim.Pe;
import org.cloudbus.cloudsim.Storage;
import org.cloudbus.cloudsim.UtilizationModel;
import org.cloudbus.cloudsim.UtilizationModelFull;
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerTimeShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
```

โปรแกรมที่ 2.2 การตั้งค่าองค์ประกอบของเครื่องเสมือน

```
// VM description
int vmid = 0;
int mips = 1000;
long size = 10000;
int ram = 512;
long bw = 1000;
int pesNumber = 1;
String vmm = "Xen";
```

โปรแกรมที่ 2.3 คำสั่งจำลองการทำงาน

```
//Starts the simulation
CloudSim.startSimulation();
CloudSim.stopSimulation();
```

โปรแกรมที่ 2.4 แสดงผลการจำลอง

```
//Print results when simulation is over

List<Cloudlet> newList = broker.getCloudletReceivedList();
printCloudletList(newList);
Log.printLine("CloudSimExample1 finished!");

private static void printCloudletList(List<Cloudlet> list) {
    int size = list.size();
    Cloudlet cloudlet;
    String indent = "    ";
    Log.printLine();
    Log.printLine("========== OUTPUT ==========");
    Log.printLine("Cloudlet ID" + indent + "STATUS" + indent +
   "Data center ID" + indent + "VM ID" + indent + "Time" +
    indent + "Start Time" + indent + "Finish Time");
    DecimalFormat dft = new DecimalFormat("###.##");
    for (int i = 0; i < size; i++) {
          cloudlet = list.get(i);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Log.print(indent + cloudlet.getCloudletId()
            + indent + indent);

            if(cloudlet.getCloudletStatus() == Cloudlet.SUCCESS){
              Log.print("SUCCESS");

              Log.printLine(indent+indent+cloudlet.getResourceId()
              +indent + indent + indent + cloudlet.getVmId()
              +indent + indent
              +dft.format(cloudlet.getActualCPUTime()) + indent
              +indent + dft.format(cloudlet.getExecStartTime())
              +indent + indent
              +dft.format(cloudlet.getFinishTime()));
            }
        }
    }
```

เมื่อสั่งดำเนินการจำลองจะได้ผลลัพธ์เป็นรายงานแสดงผลเป็นข้อความดังรูปที่ 2.7 โดยรูปแบบของผลลัพธ์จะขึ้นอยู่กับคำสั่งที่เรียกให้โปรแกรมแสดงผล

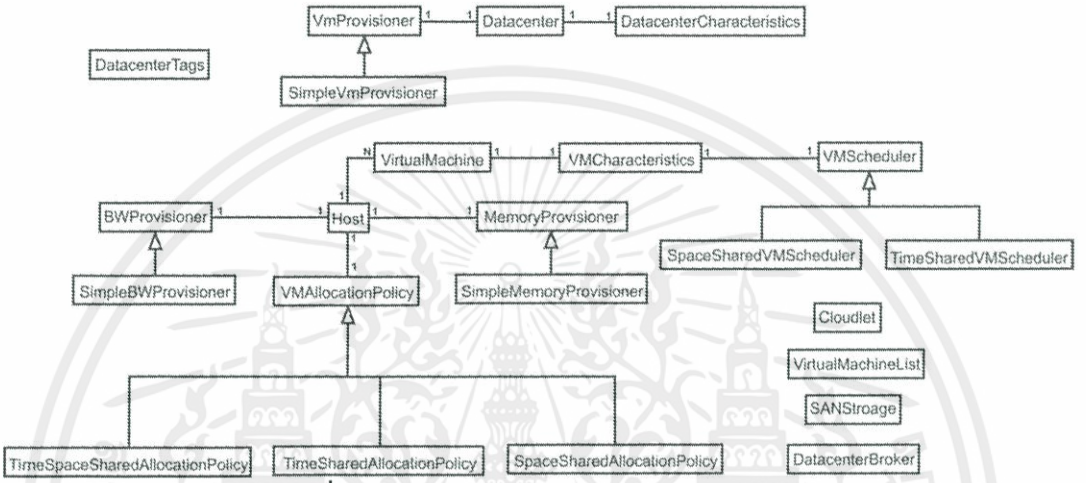
```
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
     0         SUCCESS         2             0        400         0.1         400.1
CloudSimExample1 finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

รูปที่ 2.7 ผลลัพธ์ของการจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 องค์ประกอบของคลาวด์ซิม

องค์ประกอบของคลาวด์ซิมจะอธิบายถึงรายละเอียดที่เกี่ยวข้องกับคลาส (Class) พื้นฐานของคลาวด์ซิม ซึ่งเป็นตัวสร้างบล็อกต่างๆของการจำลอง [11] ดังรูปที่ 2.8



รูปที่ 2.8 CloudSim class design diagram

1) Datacenter คลาสนี้จะจำลองการให้บริการระดับโครงสร้างพื้นฐานหลัก เช่น ฮาร์ดแวร์และซอฟต์แวร์ ถูกเสนอโดยผู้ให้บริการทรัพยากรในการประมวลผลกลุ่มเมฆ จะคลอบคลุมชุดของโฮสต์ เช่น blade servers ที่สามารถเหมือนหรือต่างกันได้ในส่วนของการตั้งค่าทรัพยากรต่างๆ ทั้งหน่วยความจำ หน่วยประมวลผล และหน่วยเก็บข้อมูล นอกจากนี้ทุกๆ องค์ประกอบของดาต้าเซ็นเตอร์ เช่น องค์ประกอบในการจัดหาทรัพยากรพื้นฐานที่ดำเนินการนโยบายสำหรับการจัดสรรแบนด์วิธ หน่วยความจำ และอุปกรณ์เก็บข้อมูล
2) DatacenterBroker คลาสนี้จะจำลองการทำงานของโบรกเกอร์ (Broker) ซึ่งมีหน้าที่เป็นสื่อกลางระหว่างผู้ใช้และผู้ให้บริการที่ขึ้นอยู่กับความต้องการของผู้ใช้และการจัดการให้บริการงานผ่านคลาวด์ โบรกเกอร์ทำหน้าที่ในนามของผู้ใช้ เพื่อระบุว่าผู้ให้บริการคลาวด์ใดที่เหมาะสมที่สุดผ่านทาง Cloud Information Service (CIS) และการเจรจากับ CIS สำหรับการจัดสรรทรัพยากรที่ตรงตามความต้องการของผู้ใช้ นักวิจัยและนักพัฒนาระบบจะต้องพัฒนาและขยายคลาสนี้เพื่อใช้ควบคุมการทดลองกับการพัฒนาแอพพลิเคชันในการจัดการนโยบาย
3) SANStorage คลาสนี้จะกล่าวถึง Storage Area Network โดยทั่วไปจะเปิดให้บริการ Cloud-based data centers สำหรับจัดเก็บข้อมูลขนาดใหญ่ SANStorage จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเตอร์เฟสที่ง่าย ซึ่งสามารถใช้ในการจำลองการเก็บข้อมูลและการกู้ข้อมูล ณ เวลาใดๆที่แสดงถึงความพร้อมของแบนด์วิธเครือข่าย การเข้าถึงไฟล์ใน SAN ขณะดำเนินงานจะเกิดดีเลย์เพิ่มขึ้นสำหรับงานที่กำลังประมวลผล เนื่องจากมีระยะเวลาขณะถ่ายโอนข้อมูลไฟล์ที่จำเป็นผ่านเครือข่ายภายในองค์กรในดาต้าเซ็นเตอร์

4) VirtualMachine คลาสนี้เป็นตัวแทนของเครื่องเสมือน ซึ่งจะจัดการในช่วงการทำงานทั้งหมดของเครื่องเสมือน คือความรับผิดชอบของโฮสต์ ตามที่ได้กล่าวไว้โฮสต์สามารถจำลองเครื่องเสมือนหลายๆเครื่องและจัดสรรตามนโยบายการแชร์หน่วยประมวลผลที่กำหนดไว้ล่วงหน้า (Space-shared , Time-shared) ทุกๆองค์ประกอบของเครื่องเสมือนมีการเข้าถึงองค์ประกอบที่เก็บลักษณะที่เกี่ยวข้อง เช่น หน่วยความจำ หน่วยเก็บข้อมูล หน่วยประมวลผล และนโยบายการจัดการทรัพยากรภายในเครื่องเสมือนด้วย ซึ่งจะขยายออกไปจากองค์ประกอบที่เรียกว่า VMScheduling
5) Cloudlet คลาสนี้จะจำลองส่วนของการให้บริการแอพพลิเคชัน Cloud-based เช่น การจัดส่งข้อมูล การติดต่อสื่อสารในเครือข่าย เวิร์กโฟลว์ของธุรกิจ (Business Workflow) ซึ่งจะใช้งานทั่วไปในดาต้าเซ็นเตอร์ คลาวด์ซิมแสดงให้เห็นถึงความซับซ้อนของแอพพลิเคชันในด้านการคำนวณของคลาวด์ซิม ทุกๆองค์ประกอบของแอพพลิเคชันมีระยะเวลาของคำสั่งก่อนจะดำเนินงานและจำนวนของข้อมูลที่จะส่งโดยถูกรับรองว่าสามารถทำงานได้สำเร็จ
6) BWProvisioner เป็น abstract class ที่จำลองนโยบายการจัดการของแบนด์วิธไปยังเครื่องเสมือน ซึ่งนำไปใช้ในองค์ประกอบของโฮสต์ ฟังก์ชั่นขององค์ประกอบนี้จะดำเนินการจัดสรรแบนด์วิธของเครือข่ายเพื่อกำหนดค่าเครื่องเสมือนที่นำไปใช้ในดาต้าเซ็นเตอร์
7) MemoryProvisioner เป็น abstract class ที่แสดงให้เห็นถึงนโยบายสำหรับการจัดสรรหน่วยประมวลผลให้เครื่องเสมือนที่ทำงานอยู่ โดยจะจำลองนโยบาย สำหรับจัดสรรพื้นที่ของหน่วยความจำของเครื่องเสมือนที่ทำงานอยู่พร้อมๆกัน การดำเนินงานและการใช้งานเครื่องเสมือนบนโฮสต์จะใช้ได้ก็ต่อเมื่อ MemoryProvisioner กำหนดว่าโฮสต์มีหน่วยความจำว่างอยู่ ซึ่งต้องมีการร้องขอสำหรับการใช้งานเครื่องเสมือนใหม่
8) VMProvisioner เป็น abstract class ที่แสดงให้เห็นถึงการจัดเตรียมนโยบายที่ใช้ในการเฝ้าติดตามเครื่องเสมือน เพื่อจัดสรรเครื่องเสมือนไปยังโฮสต์ ฟังก์ชันหลักของ VMProvisioner คือการเลือกโฮสต์ที่สามารถทำงานได้ในดาต้าเซ็นเตอร์ ซึ่งตรงกับความต้องการหน่วยความจำ หน่วยเก็บข้อมูล ความพร้อมสำหรับการใช้งานเครื่องเสมือน ค่าเริ่มต้นของ VMProvisioner จะกำหนดให้กับแพคเกจของคลาวด์ซิม ซึ่งจัดสรรเครื่องเสมือนไปยังโฮสต์แรกที่ตรงตามความต้องการดังกล่าว โฮสต์ได้รับการพิจารณาสำหรับการแมป (Mapping) ตามลำดับ อย่างไรก็ตามนโยบายที่มีความซับซ้อนมากขึ้น สามารถดำเนินงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ง่ายภายใต้องค์ประกอบนี้เพื่อให้สามารถจัดสรรได้ดีที่สุด ตัวอย่างเช่น การเลือกของโฮสต์บนพื้นฐานของความต้องการที่สามารถรองรับได้ เช่น เวลาที่ใช้ หรือ งบประมาณ

9) VMMAllocationPolicy เป็น abstract class ซึ่งโฮสต์จะจำลองนโยบาย (Space-shared , Time-shared) ที่จำเป็นสำหรับการจัดสรรความสามารถในการประมวลผลให้เครื่องเสมือน ฟังก์ชั่นการทำงานของคลาสนี้สามารถยกเลิกได้ง่ายเพื่อรองรับการประยุกต์ใช้นโยบายการแบ่งปันหน่วยประมวลผลที่เฉพาะเจาะจง

2.2.6 โมเดลของคลาวด์ซิม

การให้บริการอุปกรณ์โครงสร้างพื้นฐานสำคัญที่เกี่ยวข้องกับคลาวด์ได้จำลองอยู่ในตัวจำลอง โดยส่วนประกอบดาต้าเซ็นเตอร์สำหรับการจัดการการร้องขอบริการ การร้องขอเหล่านี้เป็นส่วนประกอบแอพพลิเคชันของแซนบอกซ์ (Sandboxed) ในเครื่องเสมือน ซึ่งจำเป็นต้องได้รับการจองในโฮสต์โดยการประมวลผลของเครื่องเสมือน ซึ่งหมายถึงชุดของการดำเนินการที่เกี่ยวข้องกับเครื่องเสมือน เช่น การจัดเตรียมโฮสต์ให้เครื่องเสมือน การสร้าง การทำลาย และการอพยพเครื่องเสมือน

ดาต้าเซ็นเตอร์ประกอบด้วยชุดของโฮสต์ ซึ่งเป็นความรับผิดชอบสำหรับการจัดการเครื่องเสมือน โฮสต์เป็นองค์ประกอบที่แสดงให้เห็นถึงโหนด (Node) ของการประมวลผลทางกายภาพในระบบคลาวด์คือการประมวลผลก่อนการตั้งค่า จะถูกแสดงในรูปของล้านคำสั่งต่อวินาที (MIPS) ต่อ 1 หน่วยประมวลผล (CPU Core) หน่วยความจำ หน่วยเก็บข้อมูลและนโยบายการจัดสรรหน่วยประมวลผลให้กับเครื่องเสมือน ส่วนประกอบของโฮสต์ในการติดต่อกับผู้ใช้ที่สนับสนุนการสร้างโมเดลและการจำลองทั้งแบบซิงเกิ้ลคอร์ (Single-core) และ มัลติคอร์ (Multi-core)

การจัดสรรแอพพลิเคชันที่ถูกส่งไปยังโฮสต์ในคลาวด์เบส (Cloud-based) ดาต้าเซ็นเตอร์ จะแสดงจำนวนของวิธีที่กำหนดเองได้สำหรับนักวิจัย ซึ่งช่วยในการดำเนินการของนโยบายใหม่ในการจัดการเครื่องเสมือน โดยขึ้นอยู่กับเป้าหมาย นโยบายพื้นฐานในการจัดการเป็นนโยบายที่ตรงไปตรงมา (Straightforward) ที่จัดสรรเครื่องเสมือนไปยังโฮสต์บนพื้นฐานของ First-Come-First-Serve (FCFS) ตัวแปรของระบบ เช่น ความต้องการของจำนวนหน่วยประมวลผล หน่วยความจำ และหน่วยเก็บข้อมูล เป็นการร้องขอโดยผู้ใช้บริการคลาวด์

สำหรับองค์ประกอบแต่ละตัวของโฮสต์ การจัดสรรของหน่วยประมวลผล จะจัดทำบนพื้นฐานของการจองโฮสต์ นโยบายจะคำนึงถึงจำนวนของหน่วยประมวลผลที่ใช้สำหรับแต่ละเครื่องเสมือนและความสามารถของหน่วยประมวลผลเท่าใดที่จะกำหนดให้กับเครื่องเสมือน ดังนั้นจึงสามารถกำหนดจำนวนหน่วยประมวลผลให้กับเครื่องเสมือนที่เราต้องการ (Space-shared Policy) หรือกระจายแบบไดนามิกในกลุ่มของเครื่องเสมือนหลัก (Time-shared Policy) และการกำหนดคอร์

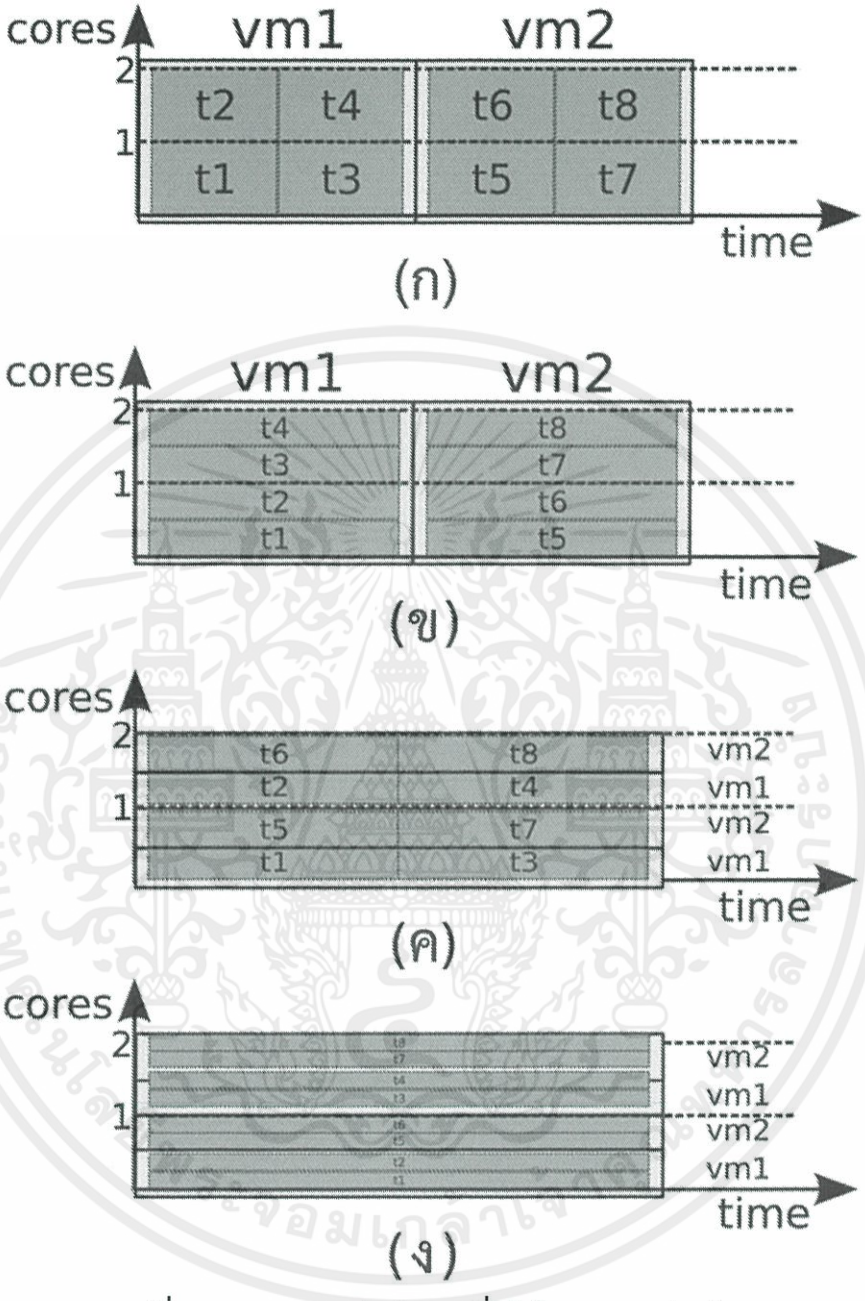
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Cores) ให้กับเครื่องเสมือนตามความต้องการหรือกำหนดนโยบายอื่นๆ ในแต่ละองค์ประกอบของโฮสต์ เช่น การจัดการงานของเครื่องเสมือนที่ใช้ space-shared policy หรือ time-shared policy สำหรับจัดสรรคอร์ให้กับเครื่องเสมือน นักพัฒนาระบบคลาวด์และนักวิจัยสามารถขยายในส่วนขององค์ประกอบการจัดการงานของเครื่องเสมือน สำหรับการทดลองที่มีนโยบายแบบกำหนดเอง

การจองเครื่องเสมือนนั้น หากต้องการให้มีการจำลองนโยบายที่แตกต่างกันภายใต้ระดับความต้องการประสิทธิภาพที่แตกต่างกัน คลาวด์ซิมจึงสนับสนุนการจัดงานของเครื่องเสมือน 2 ระดับ คือระดับโฮสต์และระดับเครื่องเสมือน ในระดับโฮสต์สามารถระบุความสามารถในการประมวลผลโดยรวมของแต่ละคอร์เป็นเท่าใด และที่ระดับเครื่องเสมือนจะกำหนดระดับความสามารถในการประมวลผลที่มีอยู่ให้แต่ละงานซึ่งอยู่ภายใต้การดำเนินงานในแต่ละเครื่อง

ที่แต่ละระดับคลาวด์ซิมจะเลือกนโยบายการจองทรัพยากรทั้งแบบ time-shared และ space-shared เพื่อแสดงให้เห็นถึงความแตกต่างระหว่างนโยบายและผลกระทบต่อประสิทธิภาพการทำงานของแอพพลิเคชัน [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ผลของการดำเนินงานที่ใช้นโยบายแตกต่างกัน

(ก) space-shared ทั้งในส่วนของเครื่องเสมือนและงาน

(ข) space-shared สำหรับจัดสรรเครื่องเสมือน time-shared สำหรับงาน

(ค) time-shared สำหรับจัดสรรเครื่องเสมือน space-shared สำหรับงาน

(ง) time-shared ทั้งในส่วนของเครื่องเสมือนและงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป (ก) ใช้นโยบาย space-shared ทั้งในส่วนของเครื่องเสมือนและงาน เครื่องเสมือนแต่ละตัวต้องการหน่วยประมวลผล 2 คอร์ จะมีเพียงเครื่องเสมือน 1 ที่สามารถทำงานในเวลานั้นๆ โดยเครื่องเสมือน 2 จะได้ทำงานก็ต่อเมื่อเครื่องเสมือน 1 ทำงานเสร็จแล้วเท่านั้น เช่นเดียวกับงานที่ต้องการเพียง 1 คอร์ ทำให้สามารถทำงานในเวลาเดียวกันได้ 2 งาน คือ t1 และ t2 ซึ่งอีก 2 งานจะต้องรอตามลำดับจนกระทั่งการทำงานเสร็จทุกงาน

รูป (ข) ใช้นโยบาย space-shared สำหรับจัดสรรเครื่องเสมือน และใช้นโยบาย time-shared สำหรับงานในแต่ละงาน ดังนั้นในช่วงการทำงานของเครื่องเสมือน งานทั้งหมดที่กำหนดให้จะเรียงลำดับกันตามเวลา และสลับไปใช้เครื่องเสมือน 2 เมื่อทำงานในเครื่องเสมือน 1 เสร็จ ผลกระทบที่สำคัญต่อเวลาของการทำงานแต่ละงานคือลำดับคิว (Queue)

รูป (ค) ใช้นโยบาย time-shared scheduling สำหรับจัดสรรเครื่องเสมือน และใช้นโยบาย space-shared สำหรับงานในแต่ละเครื่องเสมือน เครื่องเสมือนแต่ละตัวได้รับส่วน (Slice) ของเวลาแต่ละหน่วยประมวลผล และเมื่อส่วนดังกล่าวได้ถูกกระจายไปยังงาน โดยนโยบายแบบ space-shared เมื่อหน่วยประมวลผลถูกแชร์ (Shared) ความสามารถในการประมวลผลที่มีอยู่จะน้อยกว่ารูป (ก) และ (ข) โดยที่แต่ละงานจะถูกกำหนดให้เป็น space-shared จึงมีเพียง 1 งานที่สามารถจัดสรรให้แต่ละคอร์ได้ ขณะที่งานอื่นๆจะถูกเข้าลำดับไว้เพื่อรอการประมวลผลต่อไป

รูป (ง) ใช้นโยบาย Time-shared ทั้งในส่วนของเครื่องเสมือนและงาน ดังนั้นความสามารถในการประมวลผลจะถูกแบ่งกันไปโดยเครื่องเสมือน และแต่ละเครื่องเสมือนจะถูกแบ่งให้ทำงานพร้อมๆกัน ในกรณีนี้จะทำให้ไม่มีงานที่ต้องรอการทำงานทั้งในเครื่องเสมือนและงาน

2.2.7 ข้อดีของคลาวด์ซิม

1) คลาวด์ซิมสามารถสาธิตกรณีศึกษาที่เกี่ยวกับการจัดสรรทรัพยากรที่เป็นไดนามิค (Dynamic) ของแอพพลิเคชันที่ให้บริการได้
2) สามารถปรับปรุงความต้องการในการใช้แอพพลิเคชันภายใต้ความเปลี่ยนแปลงของทรัพยากรและรูปแบบของการเรียกใช้บริการได้
3) สามารถให้ความสำคัญกับรายละเอียดปัญหาของการออกแบบระบบโดยไม่ต้องกังวลเกี่ยวกับรายละเอียดในระดับล่างของคลาวด์ ทั้งส่วนของโครงสร้างพื้นฐานและการให้บริการ
4) สามารถจำลองการทำงานโดยไม่เสียค่าใช้จ่ายจริง ผู้ใช้งานสามารถศึกษาการทำงานของระบบคลาวด์ในเชิงลึกได้
5) สามารถทดลองประมวลผลข้อสันนิษฐานก่อนที่จะพัฒนาในส่วนของซอฟต์แวร์จริงในสภาวะแวดล้อมที่สามารถทดลองซ้ำได้ เพื่อปรับปรุงประสิทธิภาพของปัญหาคอขวดก่อนนำไปใช้ในคลาวด์จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ลบข้อความนี้ออก และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

6) ทำให้ผู้บริหารสามารถมีข้อมูล ช่วยให้ผู้ให้บริการสามารถคำนวณและสรุปราคาในการเข้าใช้ทรัพยากรที่ซึ่งให้ความสำคัญกับการเพิ่มผลกำไรได้มากขึ้น

2.2.8 ข้อเสียของคลาวด์ซิม

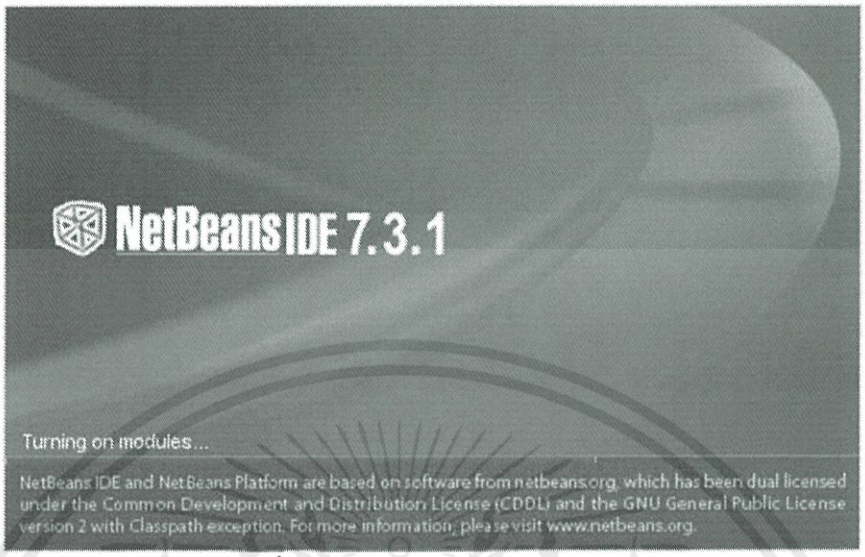
1) มีส่วนที่ขาดหายไปของการจำลองแพลตฟอร์ม (Simulation Platforms)
2) ผู้ใช้บริการและผู้ให้บริการคลาวด์ไม่สามารถเชื่อถือการประมวลผลได้ทั้งหมด หากยังไม่เข้าใจถึงการจัดการทรัพยากรอย่างแท้จริง เนื่องจากการกำหนดค่าให้กับทรัพยากรนั้นจะมีผลต่อลำดับการทำงานและประสิทธิภาพของการจำลอง
3) เนื่องจากการทำงานของคลาวด์ซิมใช้ภาษาจาวา ทำให้ผู้ใช้งานจำเป็นต้องเป็นผู้ที่รู้การเขียนโปรแกรมภาษาจาวาเป็นอย่างดี จึงเป็นการยากต่อการใช้งาน จำกัดผู้ใช้งานอยู่ในกลุ่มของนักพัฒนาเท่านั้น

2.3 การใช้โปรแกรม Netbeans IDE 7.3.1

เน็ตบีนส์ (NetBeans) เป็นเครื่องมือสำหรับโปรแกรมเมอร์ที่จะใช้พัฒนา Application ด้วยภาษา Java เกิดขึ้นมาจากกลุ่มโปรแกรมเมอร์ "Rock Solid Software" ได้พัฒนาซอฟต์แวร์ขึ้นมาตัวหนึ่งที่จะใช้ในการพัฒนาแอพพลิเคชันด้วยภาษาจาวา โดยตั้งชื่อว่าเน็ตบีนส์ และเป็นโปรแกรมประเภท OpenSource Software ทำให้ผู้ใช้งานไม่ต้องเสียค่าใช้จ่ายในการซื้อโปรแกรม และยังเปิดเผย Source Code ให้ผู้สนใจและนักพัฒนานำไปดัดแปลง แก้ไข ตามกฎของ OpenSource โดยมี Sun Micro System เป็นผู้สนับสนุนโครงการ โปรแกรมนี้นอกจากจะเป็น IDE สำหรับภาษา Java แล้ว โปรแกรมนี้สามารถใช้ภาษา C/C++ , PHP , JavaScript , Groovy และ Ruby ได้อีกด้วย

ปัจจุบัน NetBeans IDE ได้รับความนิยมมากยิ่งขึ้น และได้รับการพัฒนาให้มีความสามารถสูงยิ่งขึ้น นอกจากจะใช้ในการพัฒนาแอพพลิเคชันด้วยภาษาจาวาแล้ว ยังสามารถพัฒนาอื่นๆได้อีกหลากหลาย โดยติดตั้งโปรแกรมเสริม (Add-on) ได้จากเว็บไซต์ http://www.netbeans.org

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 NetBeans IDE startup

2.3.1 ข้อดีของเน็ตบีนส์

1) โปรแกรม NetBeans นั้นทำงานแยกส่วนต่างๆ ออกจากกันเป็น Module จึงทำให้สามารถนำ Module ต่างๆที่มีผู้ที่ได้พัฒนาต่อเติมมาติดตั้งเพิ่มเติมในภายหลังได้ และผู้ใช้งานไม่ต้องเสียเวลาในการหา Plugin มาลงเพิ่มเติม
2) ใช้งานได้กับระบบปฏิบัติการ Windows , Linux , Mac OS X และ Solaris ทำให้ผู้ใช้งานสามารถประยุกต์ให้เข้ากับงานต่างๆได้ง่าย
3) ส่วนติดต่อประสานผู้ใช้แบ่งได้เป็นสัดส่วน เข้าใจง่าย ทำให้โปรแกรมเมอร์สามารถสร้างสรรค์งานได้ตรงกับความต้องการได้มากกว่าและงานมีคุณภาพมากยิ่งขึ้น

2.4 โปรแกรมภาษาจาวา

จาวา คือภาษาสำหรับการเขียนโปรแกรมภาษาหนึ่งที่สนับสนุนโปรแกรมเชิงวัตถุ โดยโปรแกรมต่างๆถูกสร้างภายในคลาส เรียกว่า เมธอด (Method) โดยปกติจะเรียกแต่ละคลาสว่าอ็อบเจกต์ โปรแกรมที่สมบูรณ์จะเกิดจากหลายอ็อบเจกต์ หรือหลายคลาสมารวมกัน โดยแต่ละคลาสจะมีเมธอดแตกต่างกันไป

2.4.1 ประวัติความเป็นมาของจาวา

เอกสารนี้เป็นเอกสาร[illegible]ไม่ว่ากรณีใด[illegible]ทางการค้า

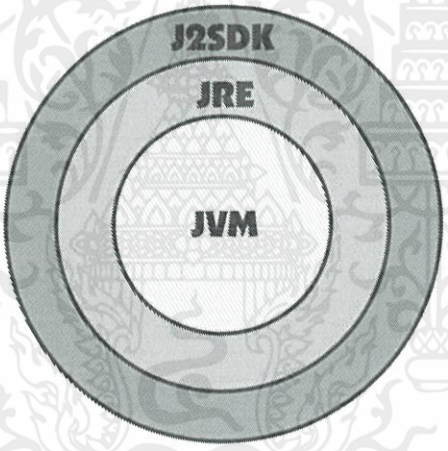
ในปี ค.ศ. 1991 ได้มีการคิดค้นโครงการขึ้นมาใหม่ชื่อว่า “Green Project” ซึ่งผู้คิดค้นคือทีมวิศวกรของบริษัทซันไมโครซิสเต็ม (Sun Microsystems) นำทีมโดย James Gosling เพื่อทำการ

วิจัยและสร้างอุปกรณ์อิเล็กทรอนิกส์ที่สามารถควบคุมการทำงานได้ด้วยรีโมทคอนโทรล ซึ่งมีหน้าที่ในการสร้างภาษาใหม่สำหรับติดตั้งให้กับอุปกรณ์ เพื่อควบคุมให้อุปกรณ์สามารถทำงานได้อย่างมีประสิทธิภาพ โดยภาษาที่สร้างขึ้นต้องเป็นภาษาที่มีขนาดเล็ก เพื่อที่อุปกรณ์เหล่านี้จะได้ไม่ต้องใช้หน่วยประมวลผลที่สูงมากนักสำหรับการประมวลผล

ต่อมาภาษานี้ได้รับการพัฒนาจนสำเร็จ และตั้งชื่อภาษานี้ว่า โอ๊ค (Oak) หลังจากได้ทำการตรวจสอบพบว่ามีผู้จดทะเบียนชื่อ โอ๊ค เป็นภาษาโปรแกรม (Programming language) แล้วจึงได้เปลี่ยนชื่อจาก โอ๊ค มาเป็นภาษาจาวา (Java) แทน

2.4.2 องค์ประกอบของจาวา

ภาษาจาวามีส่วนประกอบหลัก 3 ส่วน คือ Java Virtual Machine (JVM) , Java Runtime Environment (JRE) และ Java 2 Software Developer Kit (J2SDK) ดังรูป 2.11

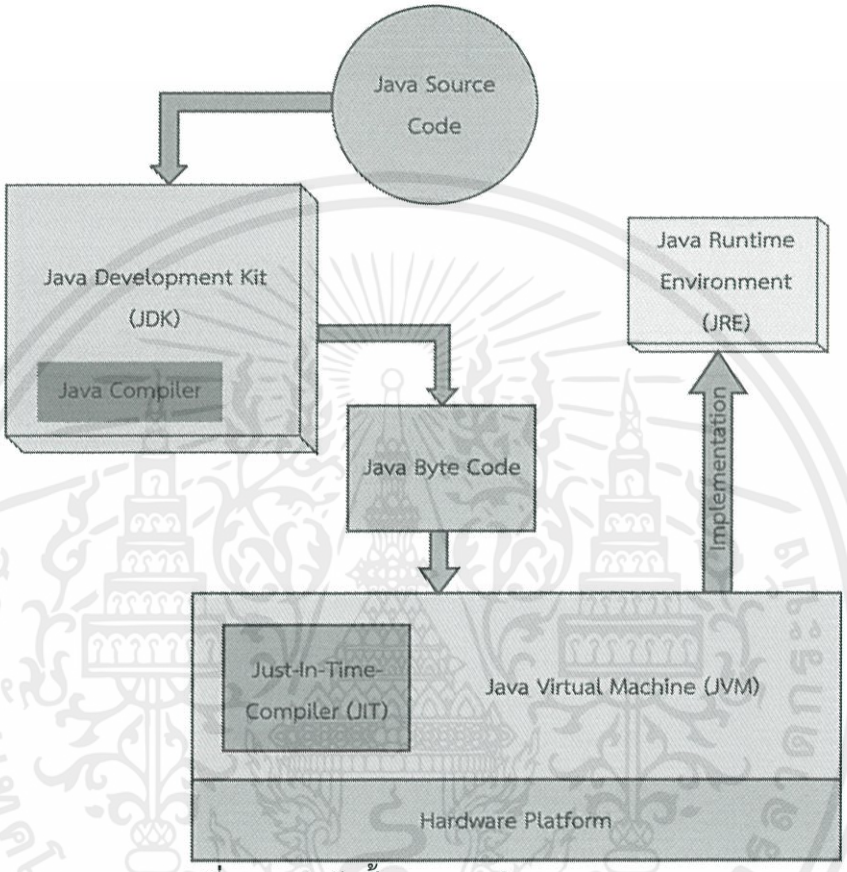


รูปที่ 2.11องค์ประกอบภาษาจาวา

1) Java Virtual Machine (JVM) คือ Software Program ซึ่งเป็นส่วนประกอบหนึ่งอยู่ใน JRE ซึ่งมีหน้าที่จำลองเครื่องเสมือน (Virtual machine) ขึ้นบนเครื่องคอมพิวเตอร์จริงๆ คอมพิวเตอร์เสมือนนี้จะทำหน้าที่ในการแปลไบต์โค้ด (Byte code) ไปเป็นภาษาเครื่องที่เหมาะสมกับแพลตฟอร์มจริงๆของแต่ละเครื่อง หลักการนี้ทำให้จาวาเป็นภาษาที่สามารถทำงานได้ทุกระบบปฏิบัติการดังแนวคิดเริ่มต้น “Write Once Run Anywhere”
2) Java Runtime Environment (JRE) เป็นเทคโนโลยีจาวาที่ใช้ในการรันโปรแกรมภาษาจาวา ที่จะรวบรวมคลาสและอินเตอร์เฟสต่างๆที่จำเป็นต่อการใช้งานของโปรแกรมภาษาจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Java 2 Software Developer Kit (J2SDK) เป็นชุดพัฒนาโปรแกรมประกอบด้วย คอมไพเลอร์ (Compiler) และดีบักเกอร์ (Debugger) [11]



รูปที่ 2.12 ลำดับขั้นตอนการทำงานของจาวา

2.4.3 จาวาแพลตฟอร์ม

จาวาแพลตฟอร์มแบ่งออกเป็น 3 แบบ แต่ละแพลตฟอร์มจะมีจุดประสงค์การใช้งานที่แตกต่างกันออกไปดังนี้

1) Java Platform : Standard Edition (Java SE) แพลตฟอร์มที่ใช้ในการพัฒนาโปรแกรมภาษาจาวากับเครื่องคอมพิวเตอร์ทั่วไปประกอบด้วย Java Application ใช้ในการพัฒนาโปรแกรมประยุกต์ทั่วไป และ Java Applet ที่ใช้ในการพัฒนาโปรแกรมเพื่อประมวลผลบนเว็บบราวเซอร์

2) Java Platform : Enterprise Edition (Java EE) แพลตฟอร์มที่มุ่งเน้นในการพัฒนาโปรแกรมเครือข่ายสำหรับองค์กร โดยใช้โปรแกรม Application Server หรือ Web Server สำหรับพัฒนาโปรแกรมที่มีขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Java Platform : Micro Edition (Java ME) แพลตฟอร์มที่ใช้พัฒนาโปรแกรมเพื่อใช้งานกับอุปกรณ์อิเล็กทรอนิกส์ที่มีทรัพยากรจำกัด เช่น โทรศัพท์มือถือ , Pocket PC หรือพีดีเอ (PDA : Personal Digital Assistant) สำหรับพัฒนาโปรแกรมที่มีข้อจำกัดด้านฮาร์ดแวร์

2.4.4 ข้อดีของภาษาจาวา

1) ความง่าย (Simple) ภาษาจาวาเป็นภาษาที่ง่ายต่อการศึกษาและพัฒนาโปรแกรม เนื่องจากภาษาจาวาสามารถพัฒนาได้โดยการตัดข้อด้อยของภาษาซีพลัสพลัส (C++) ออกไป เช่น เรื่องของการใช้พอยต์เตอร์ (Pointer) เป็นต้น
2) การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming , OOP) ภาษาจาวาเป็นภาษาคอมพิวเตอร์เชิงวัตถุที่สมบูรณ์ โดยให้ความสำคัญกับข้อมูล (Data) พฤติกรรม (Behavior) ของวัตถุ และความสัมพันธ์กันระหว่างวัตถุ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน
3) การป้องกันข้อผิดพลาด (Robust) ภาษาจาวาออกแบบมาเพื่อเป็นโปรแกรมที่น่าเชื่อถือ โดยมีขั้นตอนในการตรวจสอบข้อผิดพลาดที่อาจเกิดขึ้นในขั้นตอน Compile Time และ Runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้แก้จุดบกพร่อง (Debug) โปรแกรมได้ง่าย
4) เคลื่อนย้ายง่าย (Portable) ข้อกำหนดของภาษาจาวาไม่ขึ้นอยู่กับระบบปฏิบัติการใดๆ จึงสามารถประมวลผลได้กับระบบคอมพิวเตอร์ทุกประเภท
5) ประหยัด (Economize) เนื่องจากมี IDE , Application Server และ Library ต่างๆซึ่งใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้สามารถลดค่าใช้จ่ายสำหรับการซื้อเครื่องมือ (Tool) และซอฟต์แวร์ต่างๆ

2.4.5 ข้อเสียของภาษาจาวา

1) ภาษาจาวาทำงานได้ช้ากว่าภาษาเครื่อง (Native code) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น เช่น C หรือ C++ เป็นต้น เนื่องจากโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลางก่อน เมื่อโปรแกรมทำงานคำสั่งของภาษากลาง โปรแกรมจะถูกเปลี่ยนเป็นภาษาเครื่องและทำงานทีล่ะคำสั่ง (หรือกลุ่มของคำสั่ง) ทำให้ทำงานช้ากว่าภาษาเครื่อง
2) เครื่องมือที่ใช้พัฒนาโปรแกรมภาษาจาวามีความสามารถที่จำกัด ทำให้ผู้เขียนโปรแกรมต้องสร้างเครื่องมือใช้เอง ซึ่งเป็นการเสียเวลาในการพัฒนา

เอกสารนี้เป็นเอกสาร[illegible]ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การจำลองสถานการณ์

การจำลองสถานการณ์ (Simulation) เป็นการรวบรวมวิธีการต่างๆที่ใช้จำลองสถานการณ์จริงหรือพฤติกรรมของระบบต่างๆ มาไว้บนคอมพิวเตอร์โดยการใช้โปรแกรมคอมพิวเตอร์ (Software) เข้ามาช่วย เพื่อที่จะศึกษาการไหลของกิจกรรมในรูปแบบต่างๆ โดยมีการเก็บข้อมูล และทำการวิเคราะห์หารูปแบบที่ถูกต้องจากโปรแกรมคอมพิวเตอร์เพื่อปรับปรุงในอนาคต

ในปัจจุบันนี้การจำลองสถานการณ์เป็นที่นิยมอย่างมาก เนื่องจากระบบโปรแกรมคอมพิวเตอร์ได้มีการพัฒนาอย่างต่อเนื่อง จึงทำให้การจำลองสถานการณ์สามารถนำมาไปประยุกต์ใช้ได้กับหลากหลายอุตสาหกรรม อาทิเช่น อุตสาหกรรมในโรงงาน , การขนส่ง , การกระจายสินค้าหรือแม้กระทั่งการให้การบริการทางธุรกิจต่างๆ เช่น ธนาคาร โรงพยาบาล เป็นต้น [12]

2.5.1 หลักการทำงานของการจำลองสถานการณ์

การจำลองสถานการณ์คอมพิวเตอร์ (Computer Simulation) คือโปรแกรมคอมพิวเตอร์ที่จำลองลักษณะการทำงานของโปรแกรมที่สนใจตลอดเวลาหรือเฉพาะช่วงเวลาที่ให้ความสนใจ ตัวแปรโปรแกรม (Program Variables) หรือ ตัวแปรสถานะ (State Variables) เป็นตัวแปรซึ่งใช้เป็นสื่อกลางในการแสดงถึงสถานะปัจจุบันของระบบที่จำลอง โดยโปรแกรมจำลอง (Simulation Program) จะปรับเปลี่ยนค่าของตัวแปรสถานะ เพื่อที่จะพัฒนาแบบจำลองไปตามช่วงเวลาที่เปลี่ยนแปลงไป

รูปแบบของเวลาที่มีการใช้งานในระบบการจำลองสถานการณ์ มี 3 รูปแบบ ดังนี้

1) เวลาที่ระบบต้นแบบใช้ในการทำงาน (Physical Time) คือเวลาที่ระบบต้นแบบหรือระบบจริงใช้ในการทำงานดังกล่าวที่จะนำมาทำเป็นแบบจำลองจริง
2) เวลาที่เลื่อนไปในแบบจำลอง (Simulation Time) คือ เวลาที่แบบจำลองใช้ในการทำงานเพื่อจำลองระบบ ซึ่งเวลาดังกล่าวจะมีความสัมพันธ์กับเวลาที่ระบบต้นแบบใช้ในการทำงานจริง (Physical Time)
3) เวลาที่เลื่อนไปในความเป็นจริง (Wallclock Time) คือเวลาที่เลื่อนไปในความเป็นจริง (เหมือนกับเวลาที่แสดงโดยนาฬิกา) ในระหว่างที่แบบจำลองมีกำลังทำงาน

ลักษณะการทำงานของระบบการจำลองสถานการณ์ เมื่อแบ่งตามเวลาในการทำงาน จะมี 3 รูปแบบ ดังนี้

1) As-fast-as-possible Execution คือ การทำงานของแบบจำลองจะทำงานไปด้วยความเร็วสูงสุดเท่าที่จะเป็นไปได้โดยไม่สัมพันธ์กับเวลาที่เลื่อนไปในโลกของความเป็นจริง หรือการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานจะไม่ขึ้นกับเวลาที่ใช้ในการทำงาน (แต่ยังคงสัมพันธ์กับเวลาที่ระบบต้นแบบใช้ในการทำงาน)

2) Real Time Execution คือการทำงานของแบบจำลองจะมีความสัมพันธ์กับเวลาจริงที่ใช้ในการทำงานของแบบจำลอง

3) Scalable Real-Time Execution คือการทำงานจะมีความสัมพันธ์กับเวลาจริงที่ใช้ในการทำงานในลักษณะที่เป็นอัตราส่วนต่อกัน ในกรณีที่อัตราส่วนมีค่ามากกว่าหนึ่ง แบบจำลองก็จะทำงานได้เร็วกว่าระบบจริง , ในกรณีที่อัตราส่วนมีค่าน้อยกว่าหนึ่ง (อัตราส่วนจะไม่ต่ำกว่า 0) แบบจำลองจะทำงานได้ช้ากว่าระบบจริง และในกรณีที่อัตราส่วนเท่ากับหนึ่งระบบจะทำงานแบบ Real-Time Execution

2.5.2 ประเภทของสถานการณ์จำลอง

สถานการณ์จำลองแบ่งเป็น 3 ประเภท คือ สแตติก ต่อเนื่อง และชี้เฉพาะ ดังนี้

1) สแตติก (Static) และ ไดนามิก (Dynamics) สแตติก คือการเกิดของเหตุการณ์ในระบบการทำงาน ที่คงที่กับเวลาเสมอ ส่วนไดนามิก คือการเปลี่ยนแปลงของเวลาจะมีความสำคัญและมีผลกระทบต่อเหตุการณ์ต่างๆหรือตัวแปรที่กำลังสนใจ

2) ต่อเนื่อง (Continuous) และ ไม่ต่อเนื่อง (Discrete) ต่อเนื่อง คือสภาวการณ์ของระบบที่สามารถเปลี่ยนแปลงได้อย่างต่อเนื่องตลอดเวลา ส่วนไม่ต่อเนื่อง คือสภาวการณ์ของระบบที่สามารถเปลี่ยนแปลงได้ ณ จุดหนึ่งจุดใดของเวลา โดยมีความน่าจะเป็น (Probability) เข้ามาเกี่ยวข้อง

3) ชี้เฉพาะ (Deterministic) และ การสุ่ม (Stochastic) ชี้เฉพาะ คือเหตุการณ์ที่เกิดขึ้นทั้งหมดจะเกิดขึ้นภายใต้กฎเกณฑ์ที่แน่นอนและได้มีการกำหนดเวลาที่แน่นอน ส่วนการสุ่ม คือเวลาจะมีผลกระทบมาจากความน่าจะเป็นหรือความแปรปรวนจากการมาของเวลาที่ไม่คงที่ [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนาซอฟต์แวร์

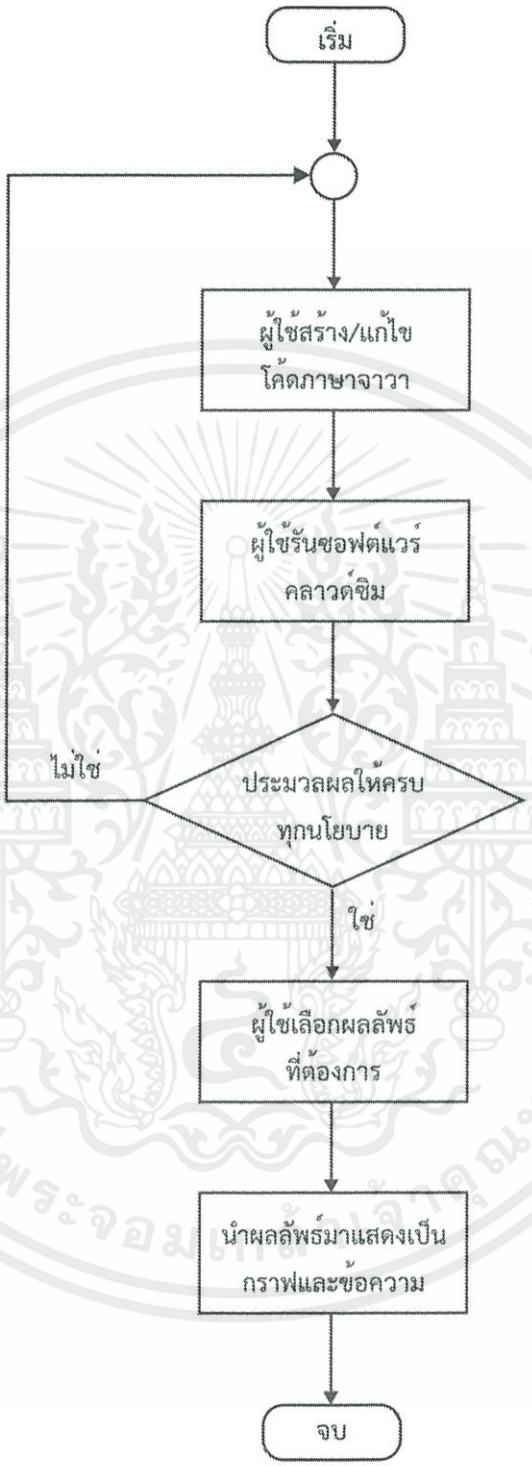
ในการพัฒนาซอฟต์แวร์จำเป็นต้องมีการวางแผนและออกแบบซอฟต์แวร์ไว้ล่วงหน้า ซึ่งในบทนี้จะอธิบายถึงแนวความคิดในการพัฒนาซอฟต์แวร์ เครื่องมือและภาษาที่ใช้ในการพัฒนา รายละเอียดของซอฟต์แวร์เชิงเทคนิค (Software Specification) โครงสร้างหลักของซอฟต์แวร์ แผนภาพ UML ที่เป็นภาพรวมของซอฟต์แวร์ที่พัฒนา และพารามิเตอร์ที่ใช้ในการทดลอง

3.1 แนวคิดในการพัฒนา

คลาวด์ซิม (CloudSim) เป็นซอฟต์แวร์จำลองการทำงานของการประมวลผลแบบกลุ่มเมฆที่มีประสิทธิภาพ เนื่องจากมีความสามารถและประสิทธิภาพในการจำลองการทำงานขององค์ประกอบต่างๆในคลาวด์ซิม และมีการพัฒนาความสามารถของซอฟต์แวร์อย่างต่อเนื่อง ทำให้เป็นประโยชน์อย่างยิ่งในการศึกษาพฤติกรรมต่างๆของระบบการประมวลผลแบบกลุ่มเมฆ (Cloud computing)

ในการใช้งานซอฟต์แวร์คลาวด์ซิมสำหรับจำลองการทำงานระบบคลาวด์ (Cloud Systems) เริ่มจากผู้ใช้เรียกใช้งานซอฟต์แวร์ผ่านโปรแกรมที่รองรับการเขียนด้วยภาษาจาวา (Java Language) ในโครงงานนี้จะใช้โปรแกรม Netbeans เป็นเครื่องมือในการสร้าง กำหนดชุดคำสั่งเรียกว่าไลบารี (Library) ในการจำลองสภาพแวดล้อมและการทำงานพื้นฐานของระบบคลาวด์ จากนั้นผู้ใช้ต้องตั้งค่าตัวแปรต่างๆ ทั้งในส่วนของฮาร์ดแวร์ นโยบายที่ใช้ควบคุมการทำงาน จำนวนงาน และองค์ประกอบพื้นฐานอื่นๆที่มีผลต่อการจำลองการทำงาน และเรียกใช้คำสั่งดำเนินการจำลอง เมื่อสั่งดำเนินการจำลองจะได้ผลลัพธ์เป็นรายงานแบบข้อความ (Text) และรูปภาพที่แสดงถึงภาพรวมของระบบคลาวด์ จากการจำลองผู้ใช้จะได้ผลการจำลองที่ต้องการเพื่อนำมาเขียนกราฟ โดยในโปรแกรมจะทำการจำลองทั้งหมด 8 รอบ เพื่อให้ครอบคลุมกับนโยบายที่ใช้ในการจัดการงานและทรัพยากรได้ทั้งหมด แต่ซอฟต์แวร์จะไม่หาค่าเฉลี่ยของการจำลอง เนื่องจากในสภาพแวดล้อมและตัวแปรต่างๆมีการกำหนดค่าที่แน่นอน ไม่ได้มาจากการสุ่มค่า (random) เมื่อได้ผลการทดลองแล้วจึงนำมาสร้างกราฟแสดงประสิทธิภาพได้ โดยขั้นตอนการทำงานของคลาวด์ซิมและส่วนที่เกี่ยวข้องจะแสดงในรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



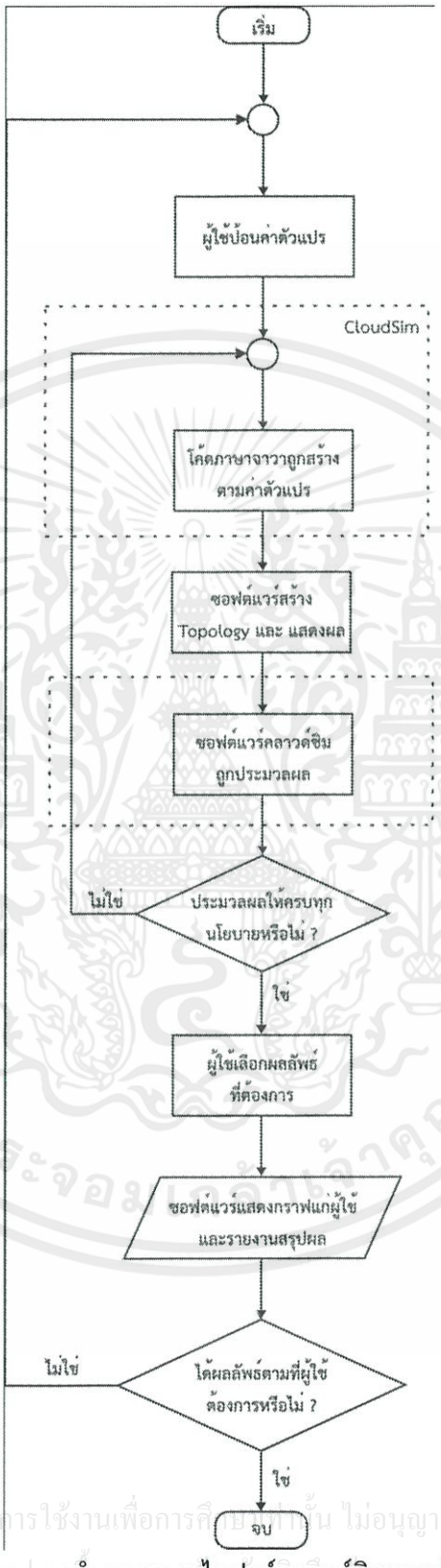
รูปที่ 3.1 ขั้นตอนการทำงานของคลาวด์ซิมและส่วนที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความยุ่งยากและซับซ้อนของการใช้งานคลาวด์ซิม เราจึงได้พัฒนาซอฟต์แวร์ที่ช่วยให้จำลองการทำงานระบบคลาวด์ได้ง่ายขึ้น ซึ่งจะคลอบคลุมการทำงานของคลาวด์ซิมไว้ให้ผู้ใช้เห็นแค่

ส่วนของซอฟต์แวร์ที่ถูกพัฒนาขึ้น เช่น การกำหนดค่าตัวแปรที่เป็นองค์ประกอบพื้นฐาน, รูปแบบของกราฟที่ต้องการวัดประสิทธิภาพ หรือ การเลือกเก็บค่าตัวแปรที่ตั้งค่าไว้ เป็นต้น จากนั้นซอฟต์แวร์ยูไอฟอร์คลาวด์ซิม (UI4CloudSim) จะทำการจำลองการทำงานตามที่ผู้ใช้กำหนด แสดงโครงสร้างของระบบคลาวด์และแสดงกราฟในรูปแบบต่างๆ โดยผู้ใช้ไม่จำเป็นต้องรู้ถึงการทำงานของคลาวด์ซิม หลังจากเลือก “Run Simulation” ผู้ใช้จะได้กราฟที่ต้องการวัดประสิทธิภาพและรายงานสรุปผลการจำลอง กราฟทั้ง 4 แบบที่ผู้ใช้สามารถเลือกแสดงผลการจำลองได้ คือ Makespan Time (Scheduling Policy), Makespan Time (Resource Allocation Policy), Load Balancing และ Execution Time ซึ่งขั้นตอนการทำงานของยูไอฟอร์คลาวด์ซิม (UI4CloudSim) สามารถแสดงได้ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 ขั้นตอนการทำงานของยูไอฟอร์คลาวด์ซิมและส่วนที่เกี่ยวข้อง

3.2 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาซอฟต์แวร์ได้มีการกำหนดเครื่องมือและภาษาที่ใช้ในการพัฒนาเพื่อให้มีความเหมาะสมกับกับระบบและความต้องการใช้งาน ดังนี้

3.2.1 ระบบปฏิบัติการ

ซอฟต์แวร์คลาวด์ซิมสามารถติดตั้งเพื่อใช้งานได้บนระบบปฏิบัติการ Window, Mac, Linux และ Solaris ซึ่งผู้จัดทำเลือกพัฒนาในระบบปฏิบัติการวินโดวส์ 7 เนื่องจากสามารถติดตั้งซอฟต์แวร์คลาวด์ซิมได้ง่าย สมบูรณ์และสามารถประยุกต์ใช้ได้หลายระบบปฏิบัติการ อีกทั้งเป็นระบบปฏิบัติการที่ค่อนข้างมีข้อผิดพลาดน้อยในการใช้งาน

3.2.2 ภาษาโปรแกรมที่ใช้

ใช้ภาษาจาวา (Java Language) ในการพัฒนา ซึ่งเป็นภาษาเชิงวัตถุ (Object Oriented Programming) เหมาะกับระบบที่มีความซับซ้อน สามารถใช้ได้หลายแพลตฟอร์มโดยไม่จำเป็นต้องแก้ไขหรือคอมไพล์ใหม่ มีขั้นตอนในการตรวจสอบข้อผิดพลาดที่อาจเกิดขึ้นในขั้นตอน Compile Time และ Runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และเนื่องจากภาษาจาวาเป็นภาษาที่ได้รับความนิยมสูงจึงมีผู้สร้างไลบรารี (Library) , IDE และ Application Server เป็นจำนวนมากโดยไม่เสียค่าใช้จ่ายในการใช้งาน ทำให้ช่วยลดระยะเวลาและค่าใช้จ่ายในการพัฒนาโปรแกรม รวมถึงภาษาจาวามีไวยากรณ์ของภาษาที่ถูกออกแบบมาเป็นอย่างดีอีกด้วย

3.2.3 เครื่องมือพัฒนาซอฟต์แวร์ (Integrated Development Environment)

เครื่องมือพัฒนาซอฟต์แวร์ (IDE) จะช่วยพัฒนาซอฟต์แวร์โดยมีสิ่งอำนวยความสะดวกต่างๆ โดยจะรวมคำสั่ง เมนู และส่วนติดต่อผู้ใช้งานแบบกราฟิก (Graphical User Interface) ต่างๆมาสร้างเป็นซอฟต์แวร์ที่รวมคอมไพเลอร์ไปในตัว หรือเรียกว่าเป็นซอฟต์แวร์ที่สร้างสภาพแวดล้อม (Environment) ให้เหมาะแก่การพัฒนาซอฟต์แวร์ เพื่อเสริมให้เกิดความถูกต้องแม่นยำ ทำให้สามารถพัฒนาได้สะดวกและรวดเร็วมากยิ่งขึ้น ซึ่งเครื่องมือพัฒนาซอฟต์แวร์ที่ผู้จัดทำเลือกใช้คือ Netbeans IDE 7.3.1

NetBeans IDE เป็นเครื่องมือพัฒนาซอฟต์แวร์ที่มีประสิทธิภาพสูง ทำงานแยกส่วนต่างๆ ออกจากกันเป็น Module จึงทำให้สามารถนำ Module ต่างๆที่มีผู้ที่ได้พัฒนาต่อเติมมาติดตั้งเพิ่มเติมในภายหลังได้ และผู้ใช้งานไม่ต้องเสียเวลาในการหา Plugin มาลงเพิ่มเติม สามารถใช้งานง่าย ที่สำคัญคือเป็นเครื่องมือที่นำมาใช้งานได้โดยไม่เสียค่าใช้จ่าย นอกจากนี้ Netbeans IDE ยังสามารถ

เอกสารนี้เป็น... ไม่ว่ากรณีใด... ไม่ได้รับอนุญาต

พัฒนาซอฟต์แวร์ด้วยภาษาอื่นๆได้อีกหลากหลาย ทำได้โดยติดตั้งโปรแกรมเสริม (Add-on) จากเว็บไซต์ หรือผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของ NetBeans IDE เช่น ภาษา C, C++ Ruby, UML และ SOA เป็นต้น

3.2.4 JRE (Java Runtime Environment)

เป็นเทคโนโลยีจาวาที่ใช้ในการรันซอฟต์แวร์ภาษาจาวาที่จะรวบรวมคลาสและอินเตอร์เฟสต่างๆที่มีความจำเป็นต่อการใช้งานของซอฟต์แวร์ภาษาจาวา เพื่อให้ผู้ใช้สามารถใช้งานซอฟต์แวร์ภาษาจาวาบนระบบปฏิบัติการนั้นๆได้ โดยในโครงงานนี้ใช้ JRE 7 ซึ่งสามารถดาวน์โหลดได้โดยไม่เสียค่าใช้จ่ายจากเว็บไซต์ของจาวา

3.2.5 JDK (Java Development Kit)

ชุดของเครื่องมือ (Tools) ที่ใช้ในการพัฒนาโปรแกรม JAVA ของบริษัท ซันไมโครซิสเต็มส์ ซึ่งผู้ที่ต้องการจะพัฒนาโปรแกรมโดยใช้ภาษาจาวา เช่น Java Compiler , Java Debugger , Java Doc และ Java Interpreter หรือ Java VM จะต้องติดตั้ง JDK นี้ จึงจะสามารถประมวลผลและดำเนินการทำงานภาษาจาวาได้ เวอร์ชันปัจจุบันของ JDK คือเวอร์ชั่น 7 ประกอบไปด้วยโปรแกรมต่างๆ เช่น โปรแกรมคอมไพเลอร์ (javac.exe) , โปรแกรมอินเตอร์พรีตเตอร์ (java.exe) , โปรแกรมดีบักเกอร์ เป็นต้น โดยในโครงงานนี้ผู้จัดทำใช้ jdk-7u25 สำหรับระบบปฏิบัติการ Window x64

3.3 รายละเอียดของซอฟต์แวร์เชิงเทคนิค (Software Specification)

ซอฟต์แวร์เชิงเทคนิคประกอบด้วยการสร้างส่วนติดต่อกับผู้ใช้แบบกราฟิก การแสดงโครงสร้างของคลาวด์ (Topology) และการแสดงกราฟวัดประสิทธิภาพ ดังนี้

3.3.1 การสร้างส่วนติดต่อกับผู้ใช้แบบกราฟิก

ซอฟต์แวร์ที่พัฒนาใช้แพ็คเกต (Package) และ คลาส (Class) ของ java.awt. และ javax.swing ในการส่วนติดต่อกับผู้ใช้ โดยสร้างหน้าจออินเตอร์เฟส (Interface) เป็นแบบกราฟิก (Graphic) เพื่อให้รองรับการใช้งานบนระบบปฏิบัติการวินโดว์จึงเลือกใช้ Window Form ที่ประกอบไปด้วย Label, Textbox, Button รวมถึงการสร้าง Event Handler จากการกดเม้าส์

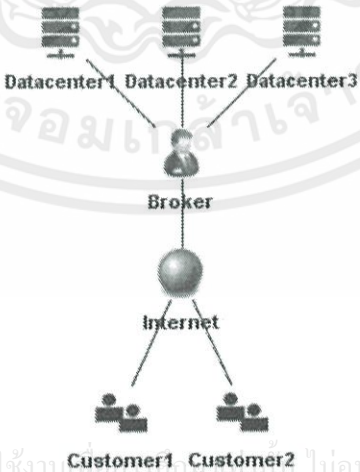
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดงส่วนติดต่อผู้ใช้แบบกราฟิกของจาวา

เพื่อให้ผู้ใช้สามารถใช้งานได้ง่าย จึงสร้างหน้าต่างให้เป็นรูปแบบของวินโดว์ มีการใช้ Textbox และ Button ผู้ใช้สามารถใช้งานตามคำสั่งต่างๆบนหน้าอินเตอร์เฟสที่เป็นภาษาอังกฤษ

3.3.2 การแสดงโครงสร้างของคลาวด์ (Topology)

ซอฟต์แวร์ที่พัฒนาจะแสดงภาพรวมของระบบคลาวด์ออกมาในรูปแบบของ Topology ซึ่งประกอบด้วย โบรกเกอร์, ผู้ใช้งาน, ดาต้าเซ็นเตอร์, โฮสต์ และเครื่องเสมือน ผู้ใช้จะเข้าใช้งานผ่านอินเตอร์เน็ต และทำการเชื่อมต่อกับคลาวด์ผ่านโบรกเกอร์ ในการแสดง Topology จะใช้ package ของ mxgraph เป็น package ที่เหมาะสำหรับการสร้างกราฟที่มีการเชื่อมต่อกัน



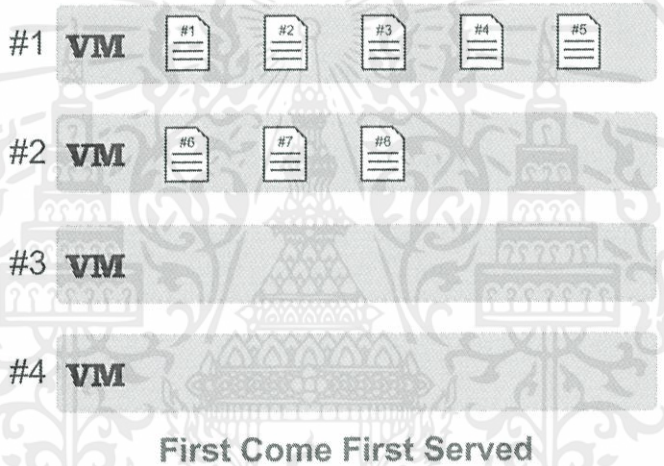
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 แสดงโทพอโลยีของคลาวด์

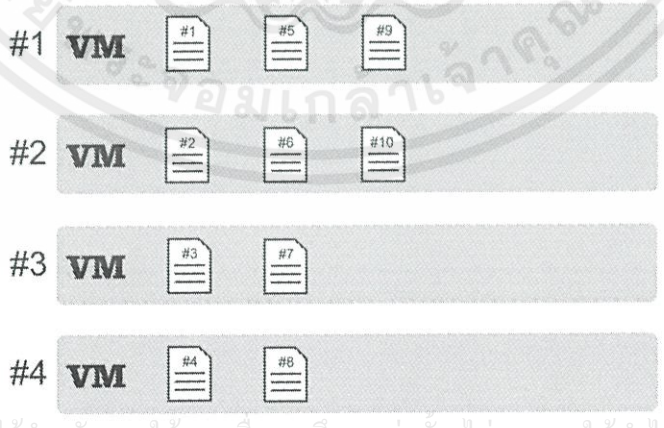
3.3.3 การแสดงกราฟวัดประสิทธิภาพ

ซอฟต์แวร์สามารถสร้างกราฟโดยใช้ไลบารี JFreeChart ซึ่งเป็นไลบารีที่ใช้ในการแสดงกราฟในภาษาจาวา เพื่อแสดงประสิทธิภาพตามที่ผู้ใช้ต้องการได้ ซึ่งแต่ละกราฟจะใช้หลักการคำนวณที่แตกต่างกันดังนี้

1) กราฟ Makespan Time (Scheduling Policy): กราฟแสดงเวลาที่ใช้ในการทำงานทั้งหมด ตั้งแต่เริ่มต้นจนจบการทำงานเทียบกับจำนวนงาน โดยเปรียบเทียบระหว่าง Scheduling Policy 2 แบบ คือ First Come First Served และ Round Robin ซึ่งทั้ง 2 แบบมีการจัดการงานดังรูปที่ 3.5 และ 3.6 [13]



รูปที่ 3.5 แสดงการจัดการแบบ First Come First Served



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 แสดงการจัดการแบบ Round Robin

หมายเหตุ: ประสิทธิภาพของอัลกอริทึม Round Robin ขึ้นอยู่กับขนาดของเวลาควอนตัมที่กำหนดถ้าเวลาควอนตัมมีขนาดใหญ่มาก พบกว่าการทำงานของอัลกอริทึม Round Robin จะเหมือนกับการทำงานของอัลกอริทึมแบบ First Come First Serverd

2) กราฟ Makespan Time (Resource Allocation Policy): กราฟแสดงเวลาที่ใช้ในการทำงานทั้งหมด ตั้งแต่เริ่มต้นจนจบการทำงาน เทียบกับจำนวนงาน โดยเปรียบเทียบระหว่าง Resource Allocation Policy 2 แบบ คือ Time Shared และ Space Shared[14]

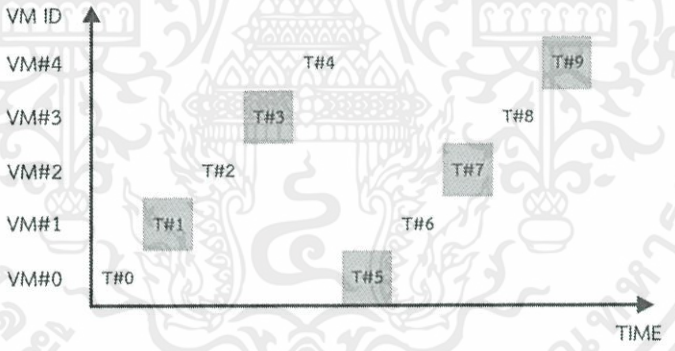
ตารางที่ 3.1 แสดงผลการจำลองที่ใช้ Resource Allocation Policy แบบ Time Shared

VM ID	Cloudlet ID
0	0
0	5
1	1
1	6
2	2
2	7
3	3
3	8
4	4
4	9

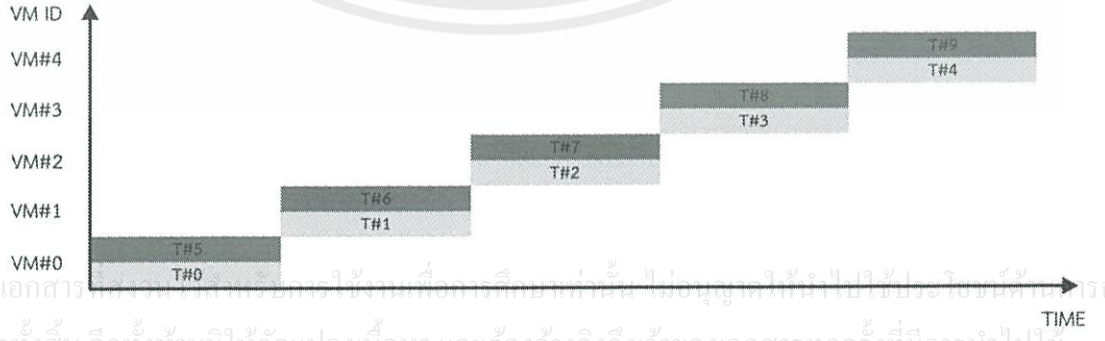
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 แสดงผลการจำลองที่ใช้ Resource Allocation Policy แบบ Space Shared

VM ID	Cloudlet ID
0	0
1	1
2	2
3	3
4	4
0	5
1	6
2	7
3	8
4	9



รูปที่ 3.7 กราฟแสดง Resource Allocation Policy แบบ Space Shared



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 กราฟแสดง Resource Allocation Policy แบบ Time Shared

ในการคำนวณกราฟ Makespan Time สามารถคำนวณได้จากสมการ 3.1

$$T_{sum} = T_{sum} + T_i \quad (3.1)$$

โดย T_{sum} คือ เวลารวมทั้งหมด

T_i คือ เวลาที่งาน i ใช้ไป

3) กราฟ Load Balancing: กราฟแสดงความสมดุลของการกระจายงานในแต่ละช่วง โดยผู้ใช้สามารถเลือกกราฟให้แสดงจำนวนงานของแต่ละเครื่องเสมือนได้ สามารถคำนวณได้จากสมการ 3.2 และ 3.3 [15]

$$T_i = \frac{total_tasklength_j}{pe_num_j \times pe_mips_j} \quad (3.2)$$

$$DI = \frac{T_{max} - T_{min}}{T_{avg}} \quad (3.3)$$

โดย T_i คือ เวลาของงานที่ i

DI คือ ค่าความไม่สมดุล (Degree of Imbalance)

(โดย DI ที่มีค่ามาก หมายถึง การกระจายงานยังไม่สมดุล)

กราฟ Load Balancing VMs: เป็นกราฟย่อยที่แสดงจำนวนงานในแต่ละเครื่องเสมือน คำนวณจากสมการ 3.4

$$vm_j = \sum cloudlet_i \quad (3.4)$$

โดย $Cloudlet_i$ คือ งานที่ i

sum_vm_j คือ ผลรวมของ $Cloudlet_i$ ใน vm_j

4) กราฟ Execution Time: แสดงเวลาที่ใช้ในแต่ละงาน โดยแสดงลำดับของงานเทียบกับเวลา คำนวณจากสมการ 3.5 และ 3.6

$$avg_TimeStart_i = \frac{\sum TimeStart_i}{i} \quad (3.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$avg_TimeFinish_i = \frac{\sum TimeFinish_i}{i} \quad (3.6)$$

โดย $TimeStart_i$ คือ เวลาที่เริ่มต้นของงาน i

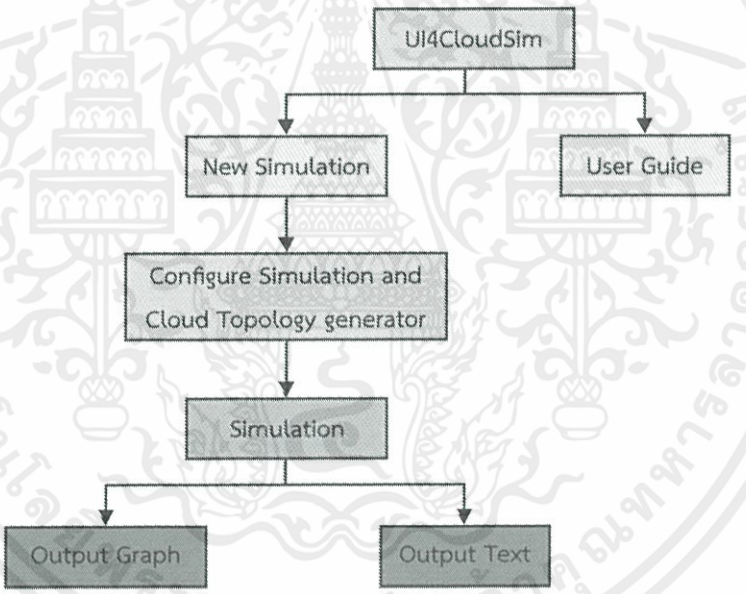
$TimeFinish_i$ คือ เวลาที่สิ้นสุดของงาน i

$avg_TimeStart_i$ คือ ค่าเฉลี่ยของเวลาเริ่มต้น

$avg_TimeFinish_i$ คือ ค่าเฉลี่ยของเวลาสิ้นสุด

3.4 โครงสร้างของซอฟต์แวร์

โครงสร้างหลักของซอฟต์แวร์และกระบวนการทำงานภายในส่วนต่างๆโดยภาพรวมสามารถแสดงให้เห็นดังรูปที่ 3.9



รูปที่ 3.9 โครงสร้างและการทำงานของยูไอฟอร์คลาวด์ซิม (UI4CloudSim)

จากรูปสามารถอธิบายได้ว่าซอฟต์แวร์ยูไอฟอร์คลาวด์ซิม (UI4CloudSim) ซึ่งเป็นส่วนติดต่อผู้ใช้จะมีส่วนประกอบ 5 ส่วน คือ สร้างการจำลอง (New Simulation) คู่มือการใช้งาน (User Guide) การตั้งค่าการจำลอง (ตัวแปรและองค์ประกอบของระบบคลาวด์) และสร้างคลาวด์ทอพอโลยี (Configure Simulation and Cloud Topology Generator) การจำลอง (Simulation) และเอาท์พุทเป็นกราฟ (Output Graph) และข้อความ (Output Text) แต่ละส่วนจะมีรายละเอียดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) สร้างการจำลอง (New Simulation) จะเข้าสู่หน้าเพื่อให้ผู้ใช้กำหนดค่าตัวแปร
2) คู่มือการใช้งาน (User Guide) แสดงวิธีการใช้งานซอฟต์แวร์ UI4CloudSim
3) การตั้งค่าการจำลองและสร้างคลาวด์ทอพอโลยี (Configure Simulation and Cloud Topology Generator) เมื่อผู้ใช้กำหนดค่าในส่วนของฮาร์ดแวร์และพารามิเตอร์ต่างๆ ซอฟต์แวร์จะสร้างทอพอโลยีเพื่อแสดงโครงสร้างของคลาวด์ตามที่ผู้ใช้กำหนด
4) การจำลอง (Simulation) ทำการประมวลผลการจำลอง โดยจะทำการประมวลผลทั้งหมด 8 ครั้ง เพื่อให้ครอบคลุม Scheduling Policy (First Come First Served และ Round Robin) และ Resource Allocation Policy ทั้ง 4 แบบ คือ Space-shared ทั้งในส่วนของ Vm และ Cloudlet, Space-Shared ในส่วนของ Vm และ Time-Shared ในส่วนของ Cloudlet, Time-Shared ในส่วนของ Vm และ Space-Shared ในส่วนของ Cloudlet และ Time-shared ทั้งในส่วนของ Vm และ Cloudlet
5) เอาท์พุทเป็นกราฟและข้อความ (Output Graph and Text) แสดงสรุปผลการจำลองในรูปแบบของกราฟและรายงานซึ่งประกอบด้วยค่าตัวแปรที่ผู้ใช้กำหนด, ทอพอโลยี และกราฟทั้ง 4 แบบ

3.5 แผนภาพ UML (Unified Modeling Language Diagram)

3.5.1 Use Case Diagram

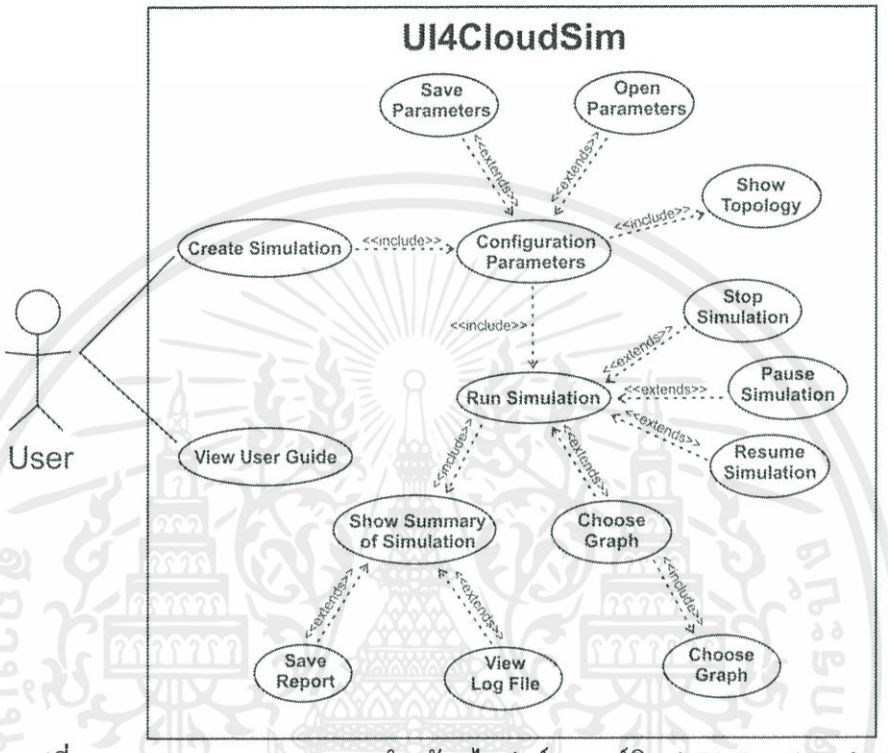
ในระบบ UI4CloudSim มีผู้ใช้ระบบ (User) โดยแยกเป็นระบบย่อยอย่างชัดเจน คือ

1) สร้างการจำลอง (Create Simulation)
2) แสดงคู่มือการใช้งาน (View User Guide)
3) ตั้งค่าพารามิเตอร์ (Configuration Paramenters)
4) ทำการจำลอง (Run Simulation)
5) เลือกกราฟ (Choose Graph)
6) แสดงผลสรุปการจำลอง (Show Summary of Simulation)

เมื่อมีการสร้างการจำลอง (Create Simulation) จะเป็นการจำลองระบบคลาวด์ขึ้นมา โดยจะมีการเรียกใช้งาน (Include) ระบบย่อย ได้แก่ บันทึกพารามิเตอร์ (Save Parameter), เปิดไฟล์พารามิเตอร์ (Open Parameter) และ แสดง Topology (Show Topology) สามารถเรียกการตั้งค่าพารามิเตอร์ (Configuration Paramenters) จากนั้นจะเรียกทำการจำลอง (Run Simulation) โดยสามารถเรียกการทำงานเพิ่มเติม (extends) ได้แก่ หยุดการจำลอง (Stop Simulation), พักการจำลอง (Pause Simulation) และการดำเนินการจำลองต่อ (Resume Simulation)

เอกสารนี้เป็น... ไม่ว่ากรณีใด... ที่มีการนำไปใช้

หลังจากทำการจำลอง จะสามารถเรียกแสดงผลสรุปการจำลอง (Show Summary of Simulation) สามารถเลือก บันทึกรายงาน (Save Report) และดู Log file (View Log file)



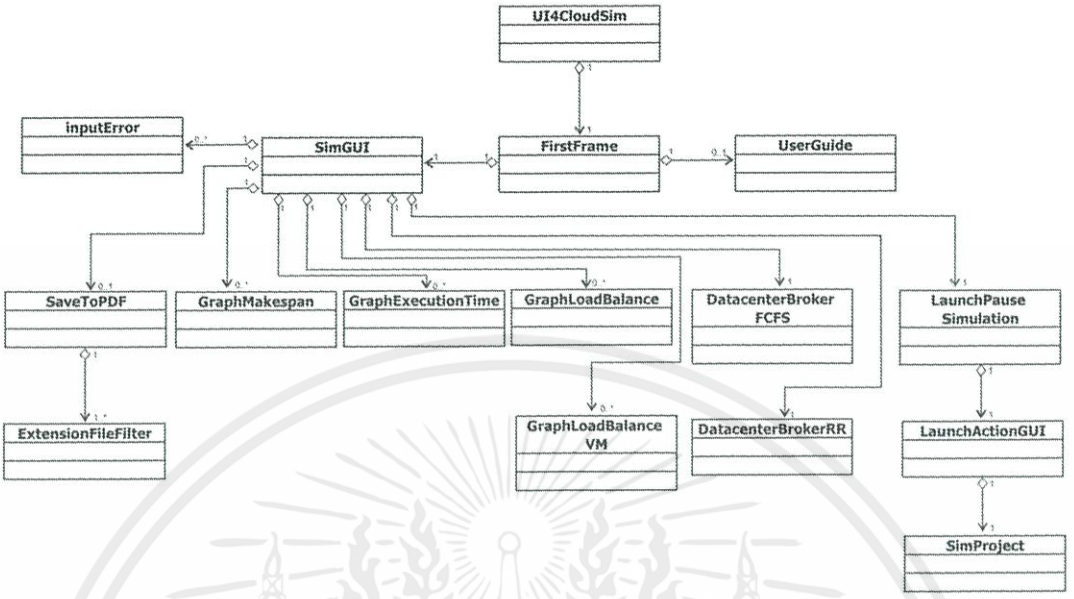
รูปที่ 3.10 แผนภาพ Use case สำหรับยูไอฟอร์คลาวด์ซิม (UI4CloudSim)

3.5.2 Class Diagram

ประกอบไปด้วยคลาสและออบเจกต์ต่างๆดังรูปที่ 3.6 โดยมีรายละเอียดดังนี้

1) Class FirstFrame สำหรับเลือกเข้าสู่ User Guide และ New Simulation
2) Class User Guide แสดงคู่มือการใช้งานโปรแกรม
3) Class SimGUI การสร้างการจำลองระบบคลาวด์ผ่านส่วนติดต่อผู้ใช้แบบกราฟิก
4) Class InputError ตรวจสอบความผิดพลาดของพารามิเตอร์ที่ผู้ใช้ใส่ในโปรแกรม
5) Class SaveToPDF แสดงรายงานเป็นไฟล์ PDF
6) Class GraphMakespan แสดงกราฟ Makespan Time
7) Class GraphExecutionTime แสดงกราฟ Execution Time
8) Class GraphLoadBalance แสดงกราฟ Load Balancing
9) Class GrpahLoadBalanceVM แสดงกราฟ Load Balancing ของ VM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แผนภาพ Class Diagram ของยูไอฟอร์คลาวด์ซิม (UI4CloudSim)

3.6 พารามิเตอร์ที่ใช้ในการทดลอง

ค่าพารามิเตอร์ที่ใช้ในการทดลองจะกำหนดจากการใช้งานคลาวด์ในปัจจุบันและกำหนดจากงานวิจัยคลาวด์ซิมที่ใช้เป็นเอกสารอ้างอิงของโครงงาน ดังนี้

ตารางที่ 3.1 กำหนดพารามิเตอร์สำหรับโบรกเกอร์

พารามิเตอร์	ค่าที่กำหนด
Number of Datacenter	1
Scheduling Policy	Round Robin
Number of Customer	2

ตารางที่ 3.2 กำหนดพารามิเตอร์สำหรับดาต้าเซ็นเตอร์

พารามิเตอร์	ค่าที่กำหนด
Number of Host	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 กำหนดพารามิเตอร์สำหรับโฮสต์

พารามิเตอร์	ค่าที่กำหนด
Number of VM	7
CPU (core)	2 Core
MIPS	1000 MB
Memory	2048 MB
Storage	1000000 MB
Bandwidth	10000 MB

ตารางที่ 3.4 กำหนดพารามิเตอร์สำหรับเครื่องเสมือน

พารามิเตอร์	ค่าที่กำหนด
Image size	10000 MB
MIPS	1000 MB
Memory	512 MB
Bandwidth	1000 MB
CPU (core)	1 Core

ตารางที่ 3.5 กำหนดพารามิเตอร์สำหรับงาน

พารามิเตอร์	ค่าที่กำหนด
Cloudlet (Task)	1300 Tasks
Length	10000 MIPS
PEs Number	1 Core

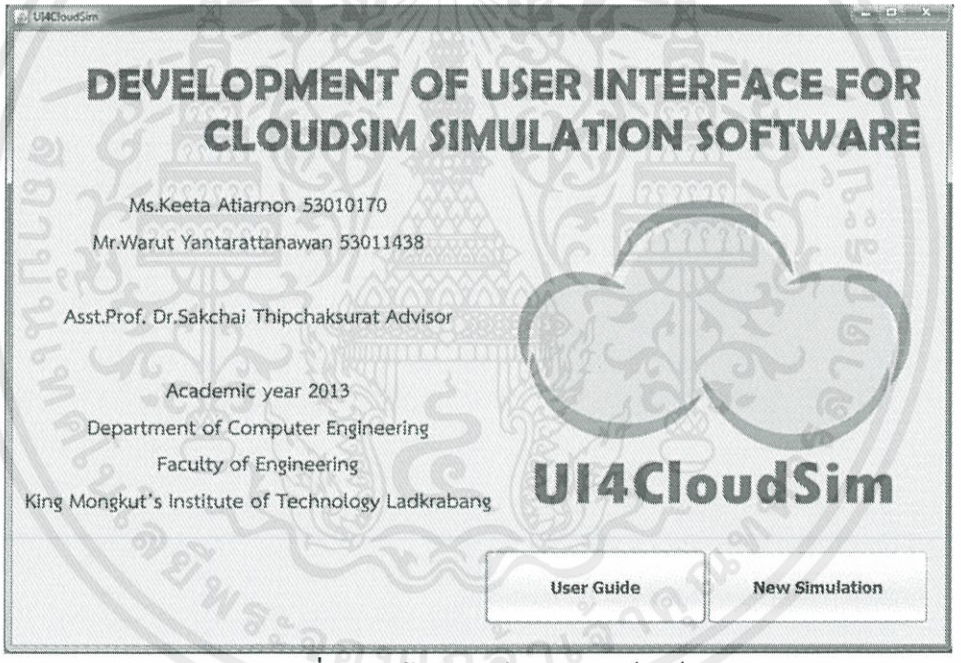
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองการตั้งค่าพารามิเตอร์และองค์ประกอบภายในคลาวด์, แสดงกราฟวัดประสิทธิภาพ รายงานสรุปผลการจำลองและคู่มือการใช้งาน

4.1 การทดลองตั้งค่าพารามิเตอร์และองค์ประกอบภายในคลาวด์

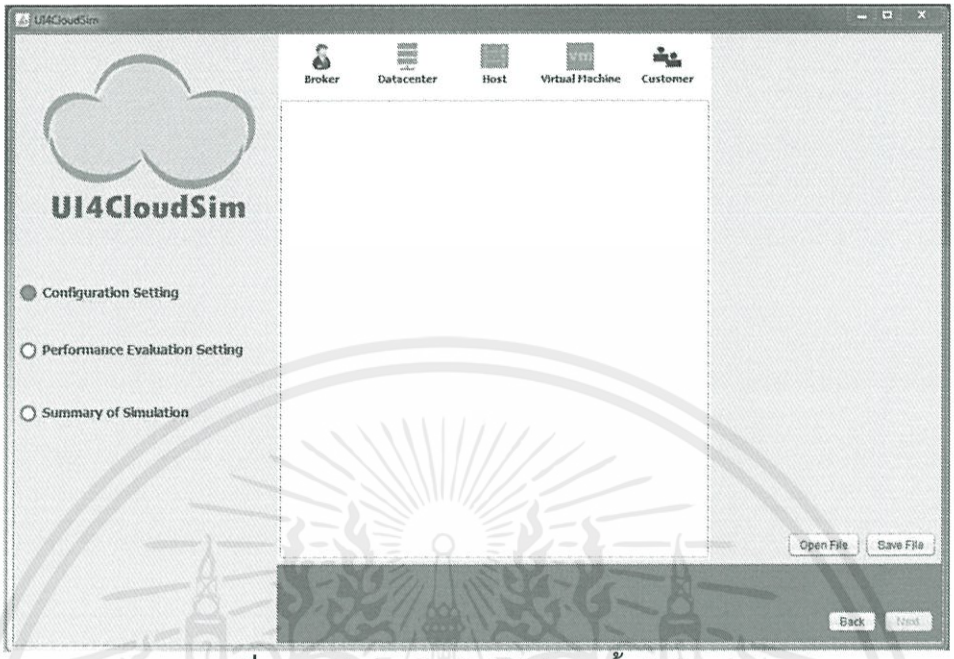
เมื่อเข้าใช้งานซอฟต์แวร์ยูไอฟอร์คลาวด์ซิมจะแสดงหน้าต่างดังรูปที่ 4.1



รูปที่ 4.1 หน้าต่างหลักของซอฟต์แวร์

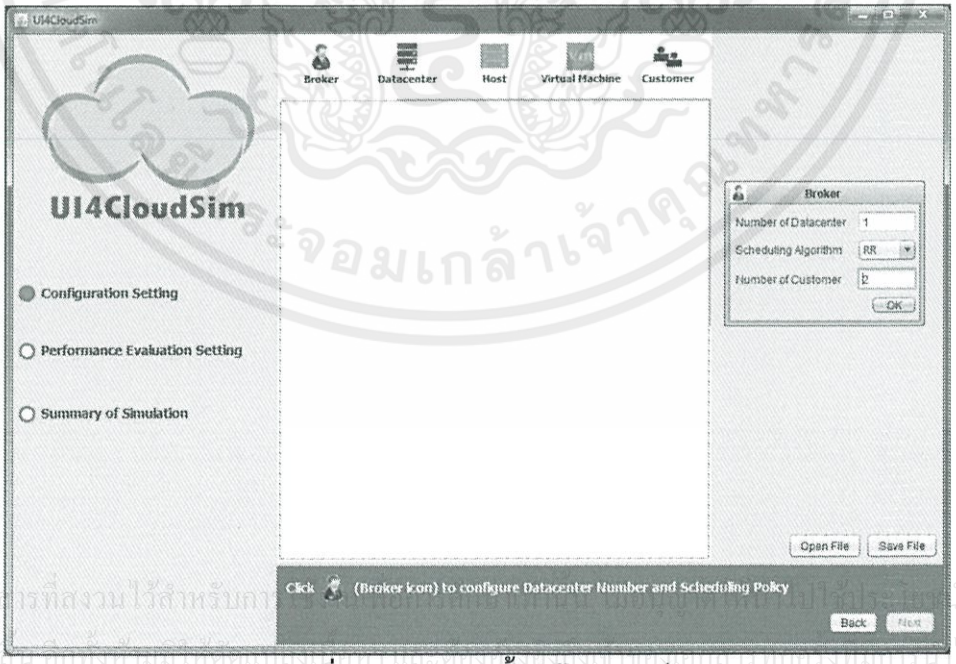
จากนั้นผู้ใช้สามารถเลือก User Guide เพื่อแสดงวิธีการใช้งานซอฟต์แวร์ UI4CloudSim หรือ เลือก New Simulation เพื่อจำลองการทำงานของระบบคลาวด์ ซึ่งจะเข้าสู่ขั้นตอนที่ 1 คือ “Configuration Setting” ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 หน้าต่างสำหรับผู้ใช้ทำการตั้งค่าตัวแปร

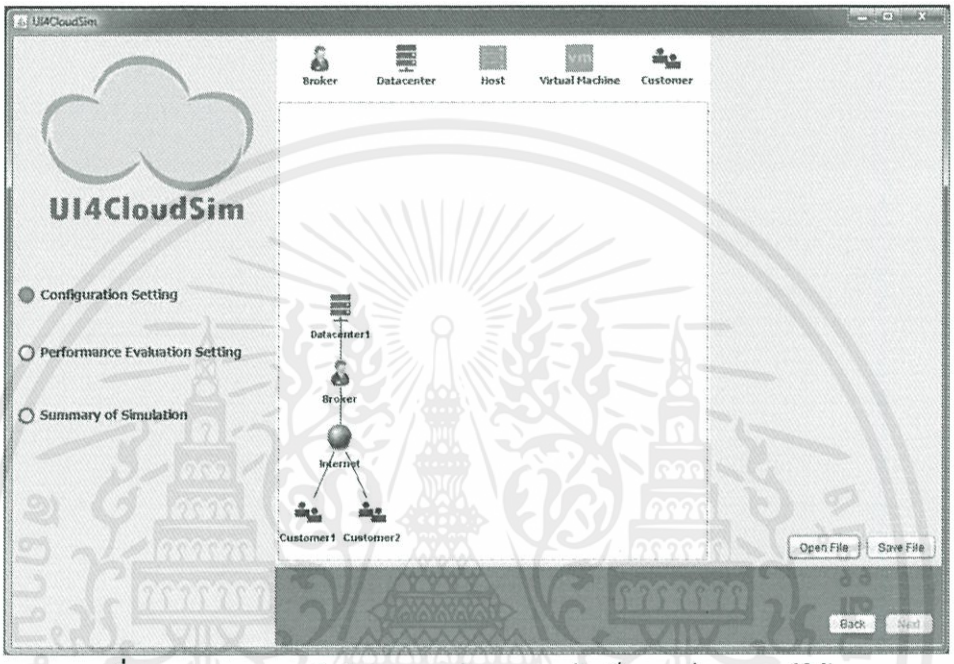
ผู้ใช้คลิกสัญลักษณ์ (Icon) “Broker” จะปรากฏหน้าต่างทางด้านขวามือ เพื่อให้ผู้ใช้กำหนดจำนวนดาต้าเซ็นเตอร์ (Number of Datacenter), นโยบายการเข้าใช้พื้นที่เพื่อจัดสรรทรัพยากร (Resource Allocation Policy) และจำนวนผู้ใช้งาน (Number of Customer) ดังแสดงในรูปที่ 4.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 แสดงการตั้งค่าโบรกเกอร์

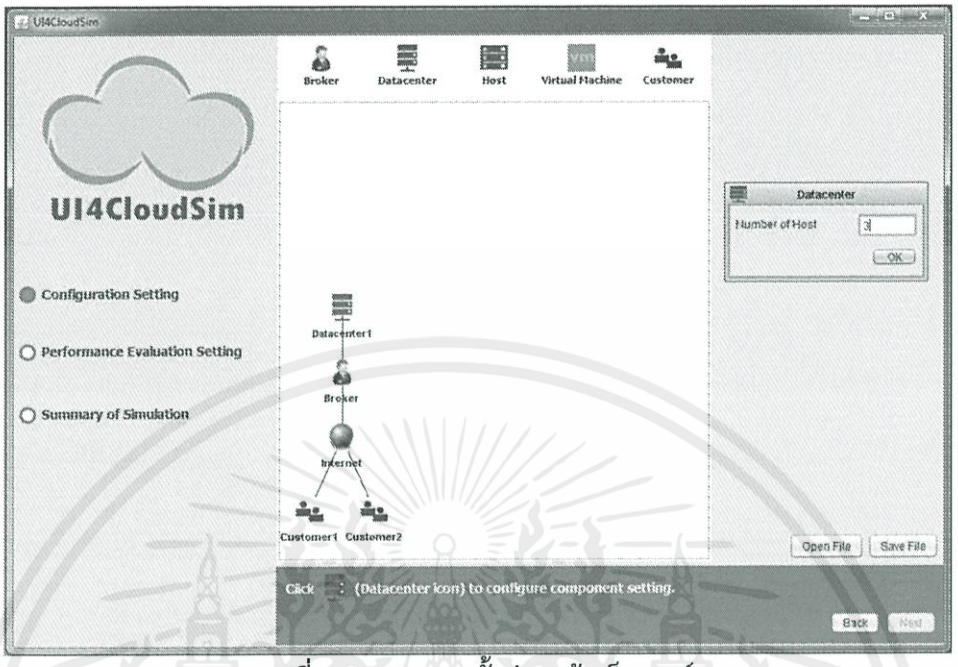
เมื่อผู้ใช้กดปุ่ม “OK” ซอฟต์แวร์จะทำการจำลองทอพอโลยีตามที่ผู้ใช้กำหนดค่า ประกอบด้วย โบรกเกอร์ (Broker) ที่ทำหน้าที่เป็นสื่อกลางระหว่างผู้ใช้ (Customer) และผู้ให้บริการ ซึ่งในที่นี้คือดาต้าเซ็นเตอร์ (Datacenter) ดังแสดงในรูปที่ 4.4



รูปที่ 4.4 แสดงทอพอโลยีประกอบด้วย 2 ดาต้าเซ็นเตอร์ และ 3 ผู้ใช้งาน

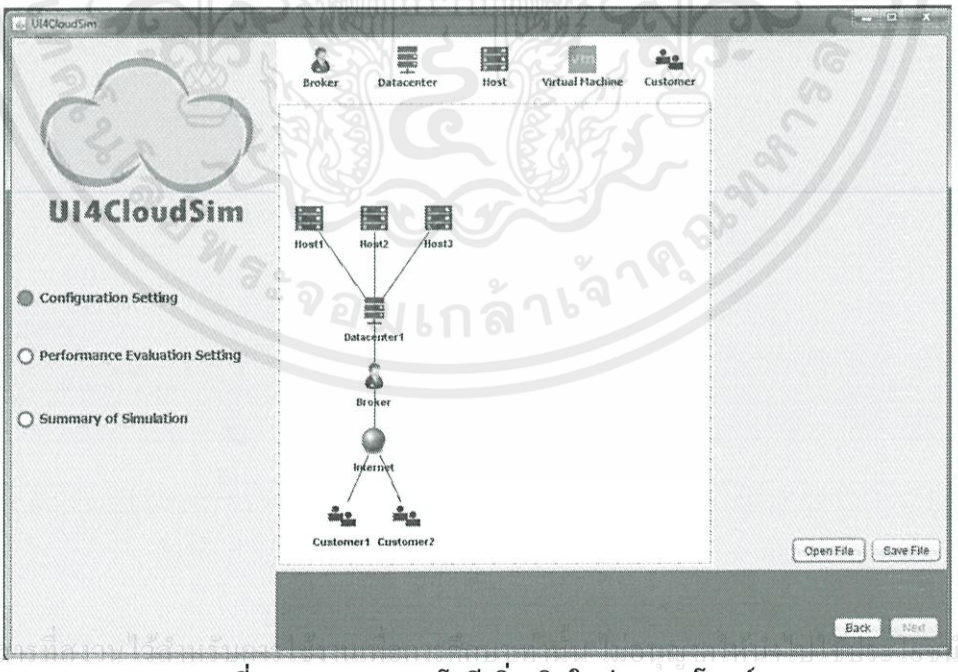
ผู้ใช้คลิกสัญลักษณ์ “Datacenter” เพื่อกำหนดจำนวนโฮสต์ (Number of Host) ที่ต้องการ ดังแสดงในรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงการตั้งค่าดาต้าเซ็นเตอร์

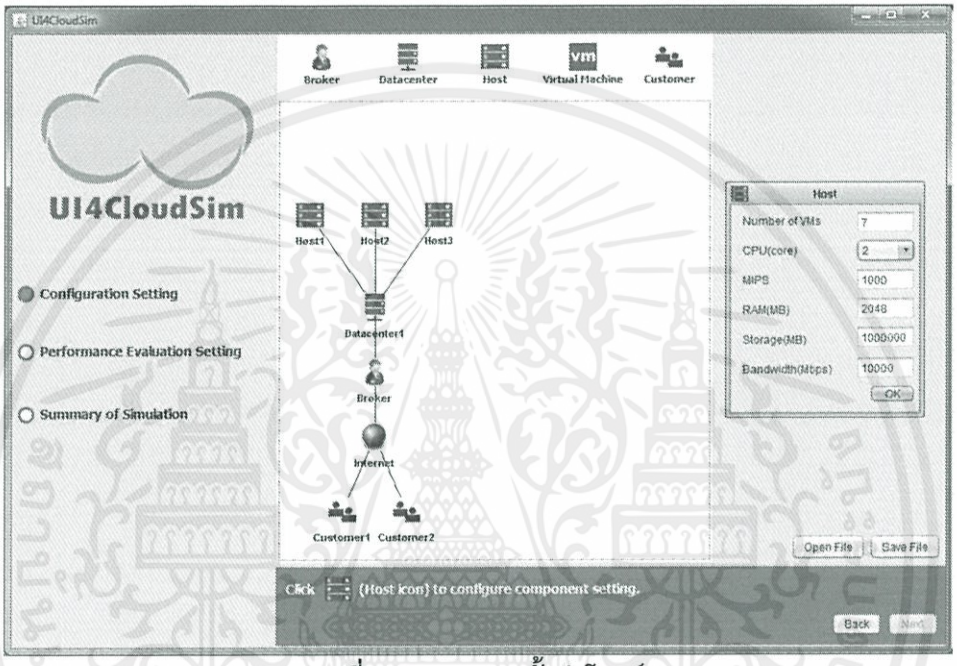
เมื่อผู้ใช้กดปุ่ม “OK” ซอฟต์แวร์จะทำการเพิ่มโครงสร้างของทอพอโลยีในส่วนของโฮสต์ (Host) โดยทำการกระจายโฮสต์ให้ในแต่ละดาต้าเซ็นเตอร์อย่างเท่าๆกัน ดังแสดงในรูปที่ 4.6



รูปที่ 4.6 แสดงทอพอโลยีเพิ่มเติมในส่วนของโฮสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

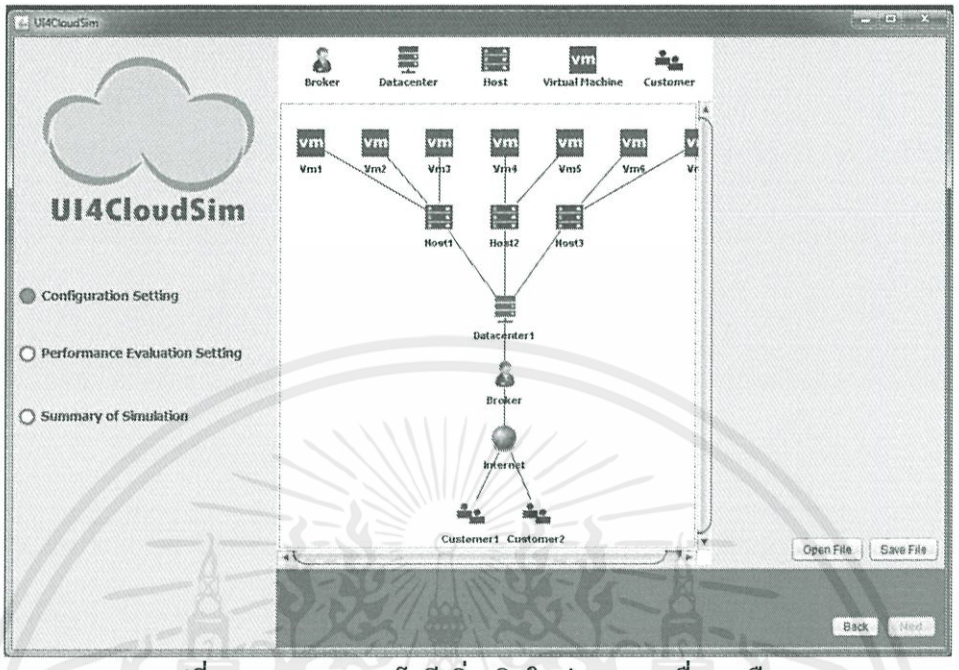
ผู้ใช้คลิกสัญลักษณ์ “Host” เพื่อกำหนดค่าตัวแปรต่างๆของโฮสต์ ได้แก่ จำนวนเครื่องเสมือน (Number of VMs), CPU cores (Processor Elements), Mips (Million instructions per second), Memory (หน่วยความจำของแต่ละโฮสต์), Storage (ที่จัดเก็บข้อมูล) และ Bandwidth (แบนด์วิธ) ดังแสดงในรูปที่ 4.7



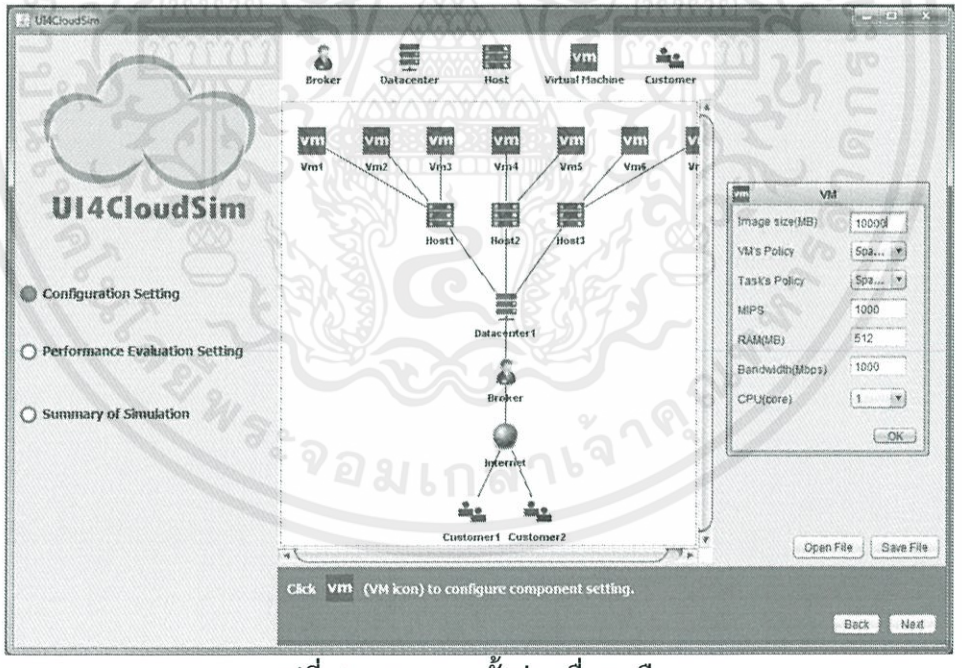
รูปที่ 4.7 แสดงการตั้งค่าโฮสต์

เมื่อผู้ใช้กดปุ่ม “OK” ซอฟต์แวร์จะทำการเพิ่มโครงสร้างของทอพอโลยีในส่วนของเครื่องเสมือน (VMs) โดยทำการกระจายเครื่องเสมือนให้ในแต่ละโฮสต์อย่างเท่าๆกัน ดังแสดงในรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงทอพอโลยีเพิ่มเติมในส่วนของเครื่องเสมือน



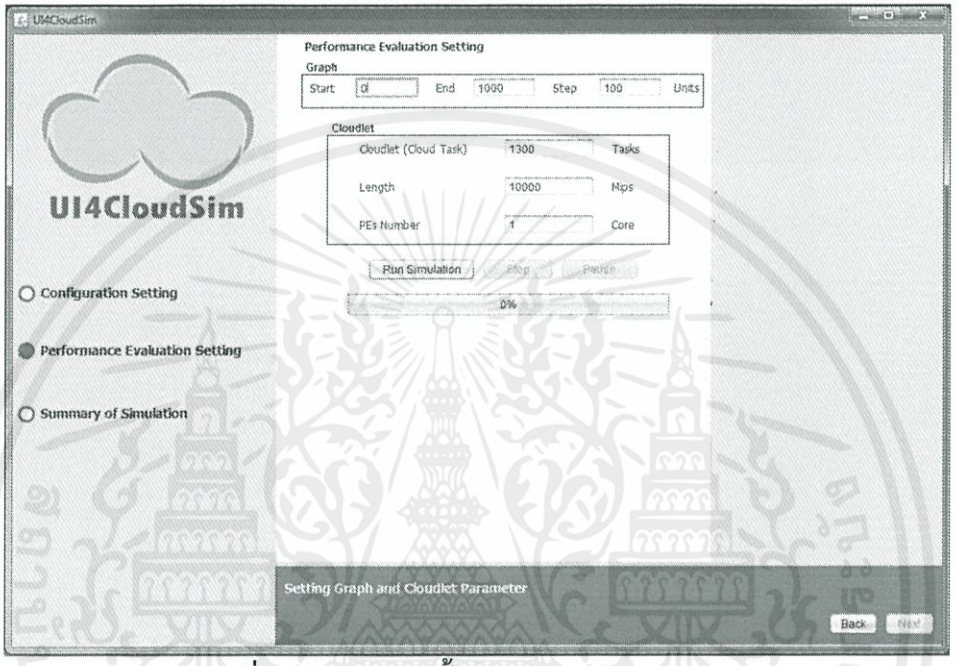
รูปที่ 4.9 แสดงการตั้งค่าเครื่องเสมือน

เมื่อผู้ใช้ทำการตั้งค่าตัวแปรต่างๆที่เป็นองค์ประกอบของคลาวด์เรียบร้อย กดปุ่ม “Next” จะเข้าสู่ขั้นตอนที่ 2 คือ “Performance Evaluation Setting”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การแสดงกราฟวัดประสิทธิภาพ

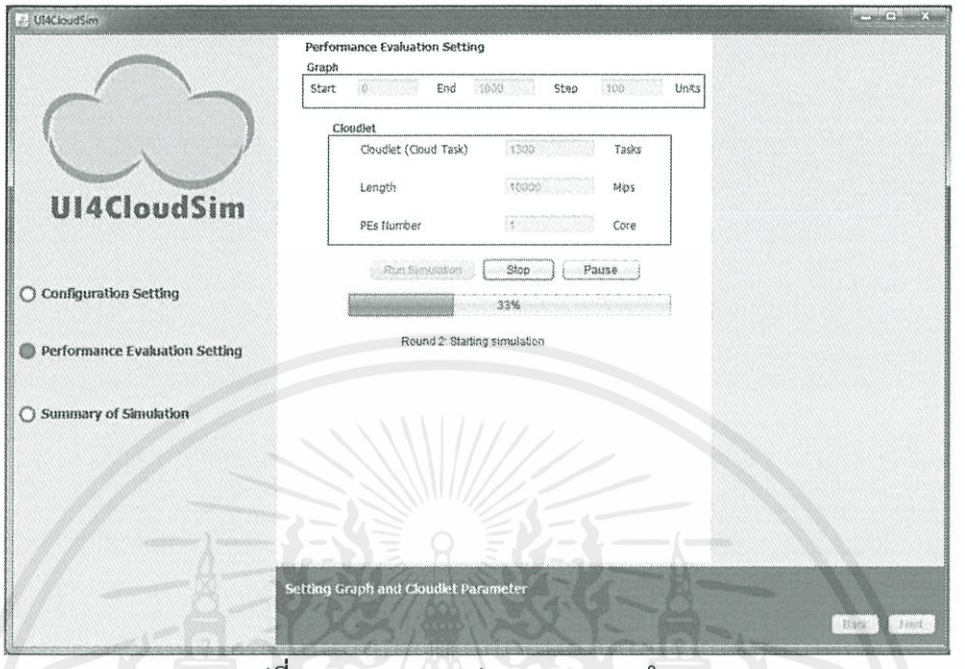
"Performance Evaluation Setting" เป็นขั้นตอนในกำหนดค่างาน (Cloudlet) และแกนของกราฟ ดังรูปที่ 4.10



รูปที่ 4.10 แสดงการตั้งค่างานและแกนของกราฟ

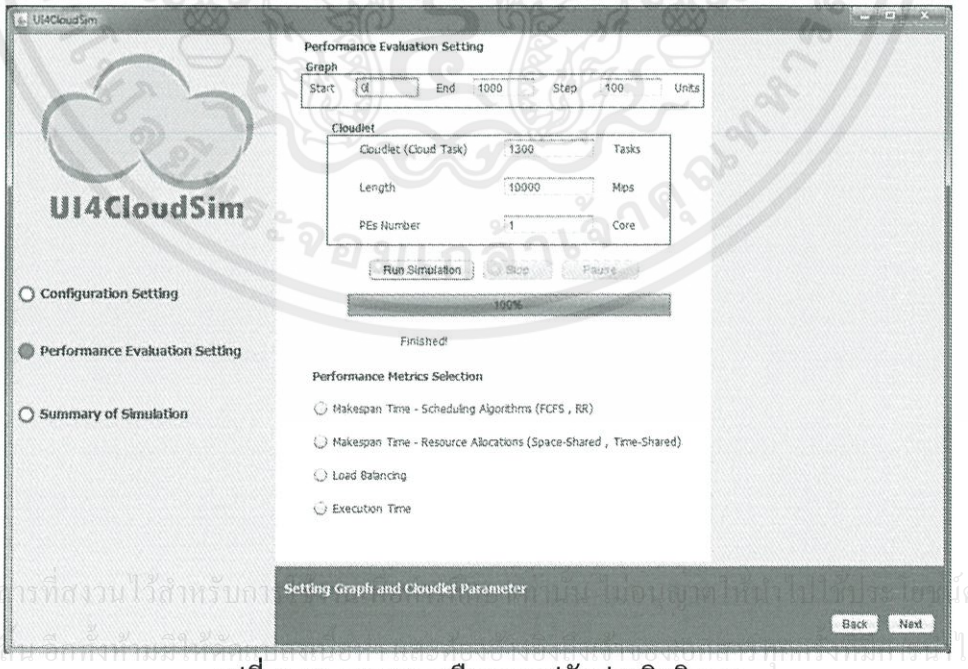
ผู้ใช้เลือกค่าในแกน Y โดยสามารถกำหนดค่าเริ่มต้น (Start) ค่าสิ้นสุด (End) และ ความถี่ (Step) จากนั้นกำหนดจำนวนงานและลักษณะของงาน เมื่อตั้งค่าเสร็จแล้วผู้ใช้กดปุ่ม "Run Simulation" เพื่อทำการจำลอง ดังรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงการประมวลผลการจำลอง

จะเห็นว่าซอฟต์แวร์จะประมวลผลทั้งหมด 8 รอบ เพื่อให้ได้ค่าตัวแปรที่ครอบคลุมทั้ง 6 นโยบาย หลังจากซอฟต์แวร์ทำการจำลองเสร็จแล้ว จะแสดง Performance Metrics Selection ให้ผู้ใช้เลือกกราฟวัดประสิทธิภาพตามที่ต้องการ ดังรูปที่ 4.12



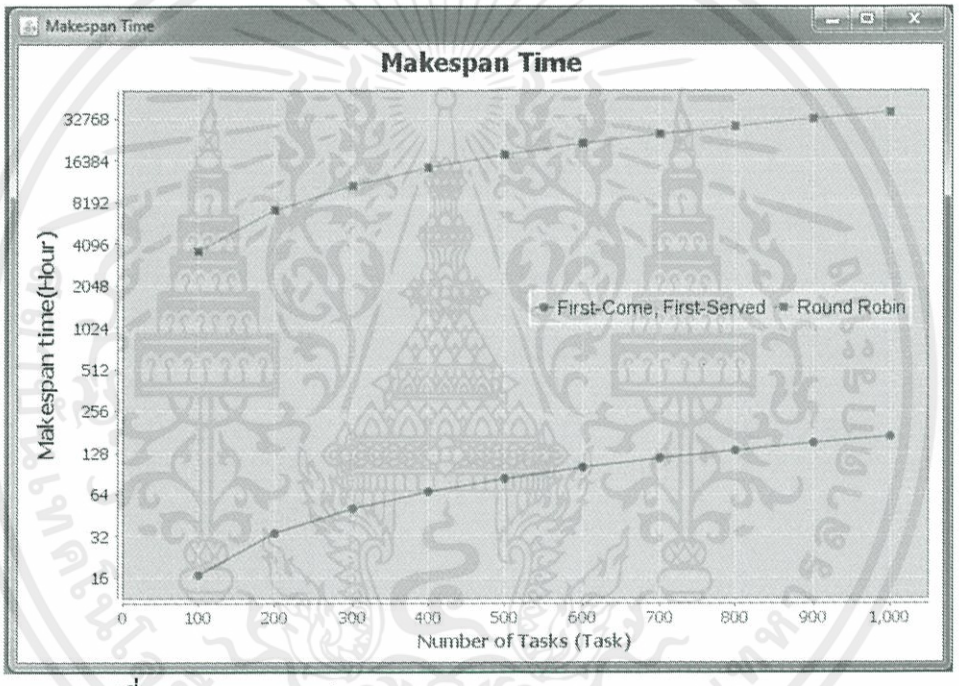
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.12 แสดงการเลือกกราฟวัดประสิทธิภาพ

หลังซอฟต์แวร์จำลองการทำงาน ผู้ใช้สามารถเลือกกราฟวัดประสิทธิภาพได้ 4 แบบ คือ

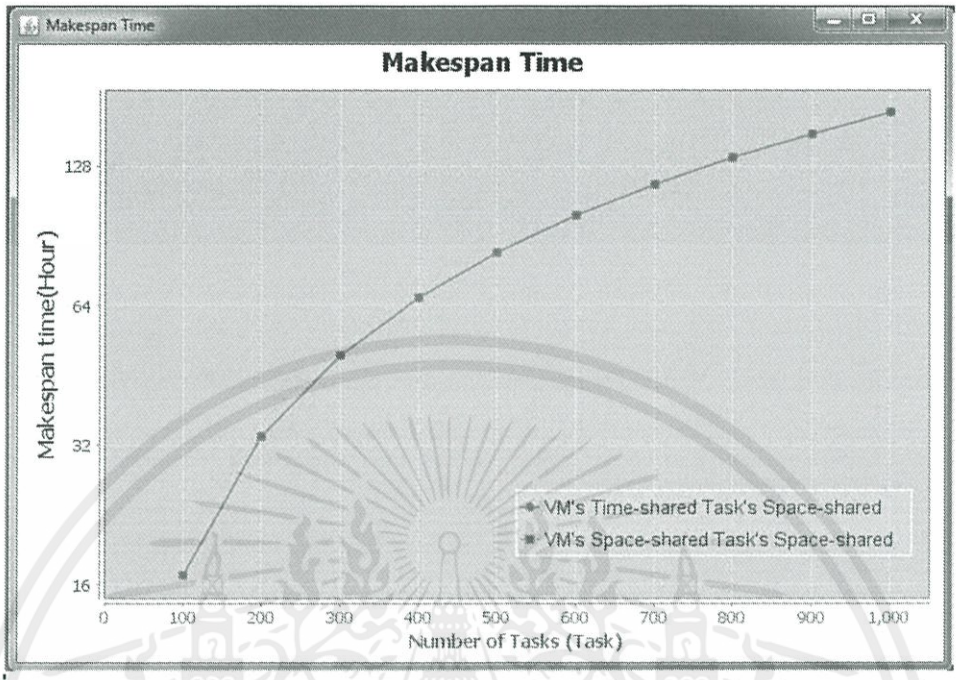
1) Makespan Time – Scheduling Algorithm (FCFS, RR)
2) Makespan Time – Resource Allocation (Space-Shared, Time-Shared)
3) Load Balancing
4) Execution Time

โดยแสดงกราฟดังรูปที่ 4.13 – 4.20

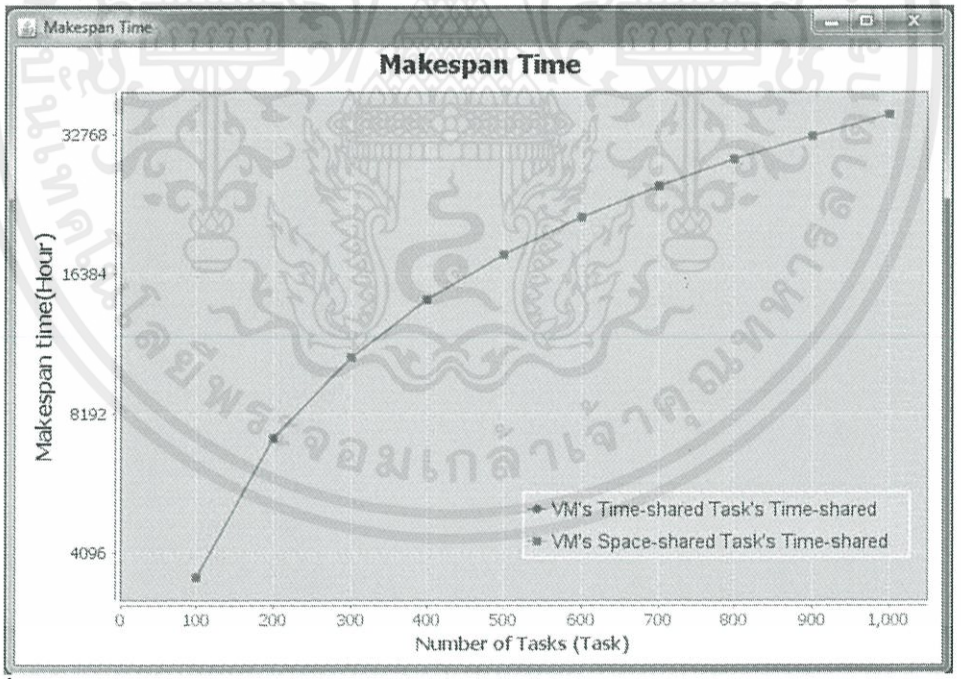


รูปที่ 4.13 แสดงกราฟ Makespan Time – Scheduling Algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

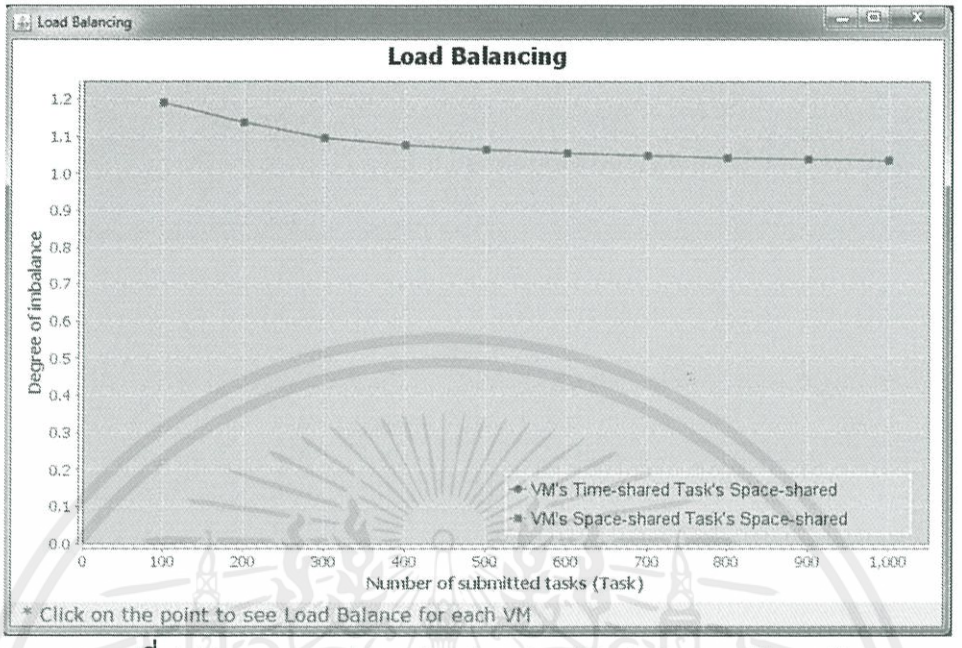


รูปที่ 4.14 แสดงกราฟ Makespan Time – Resource Allocation (Task: Space-Shared)

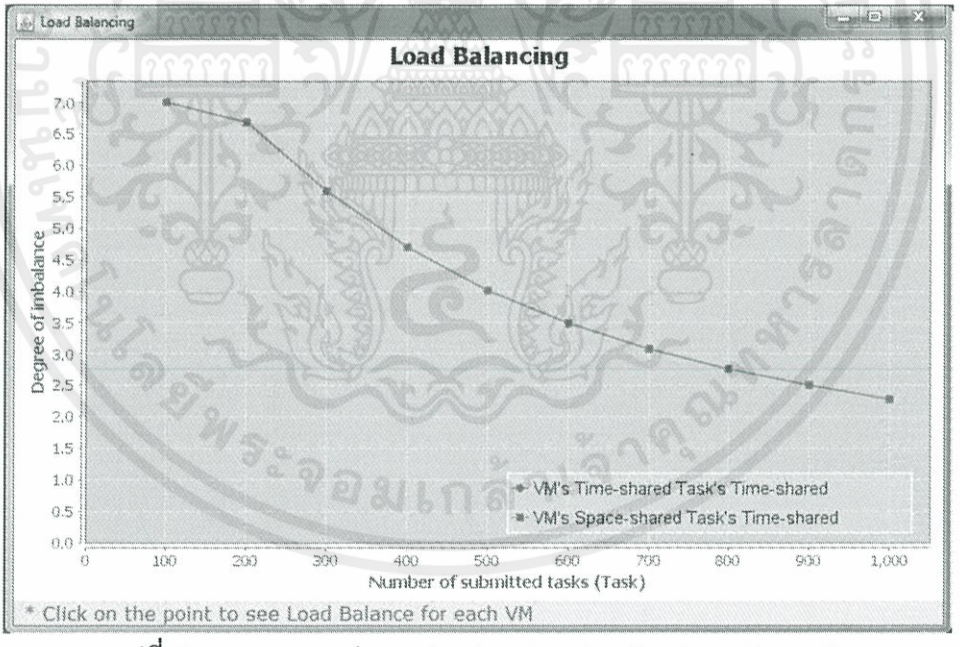


รูปที่ 4.15 แสดงกราฟ Makespan Time – Resource Allocation (Task: Time-Shared)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

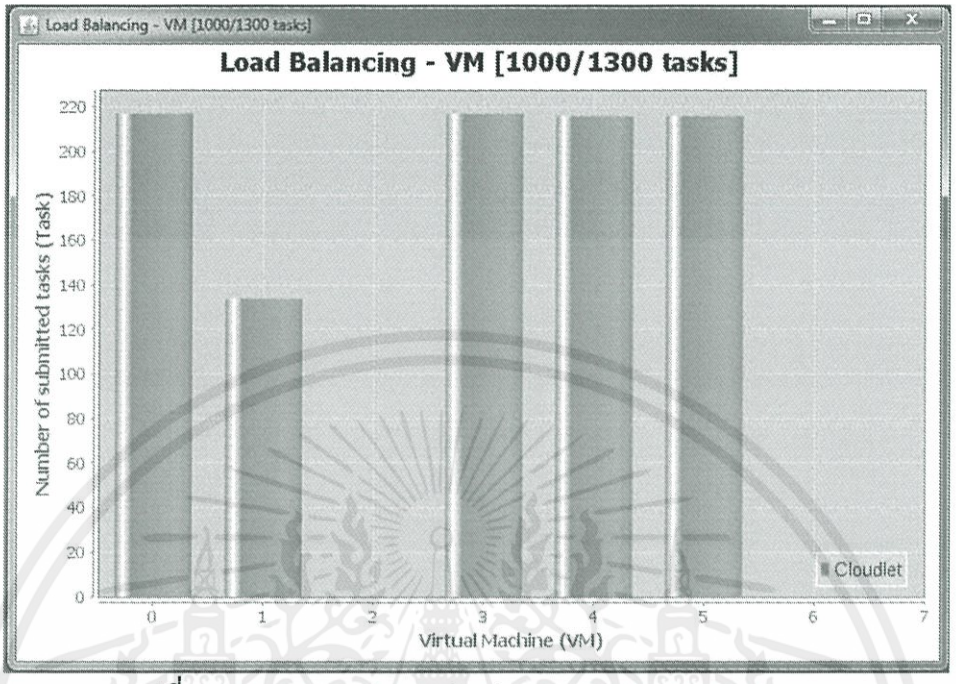


รูปที่ 4.16 แสดงกราฟ Load Balancing (Task: Space-Shared)

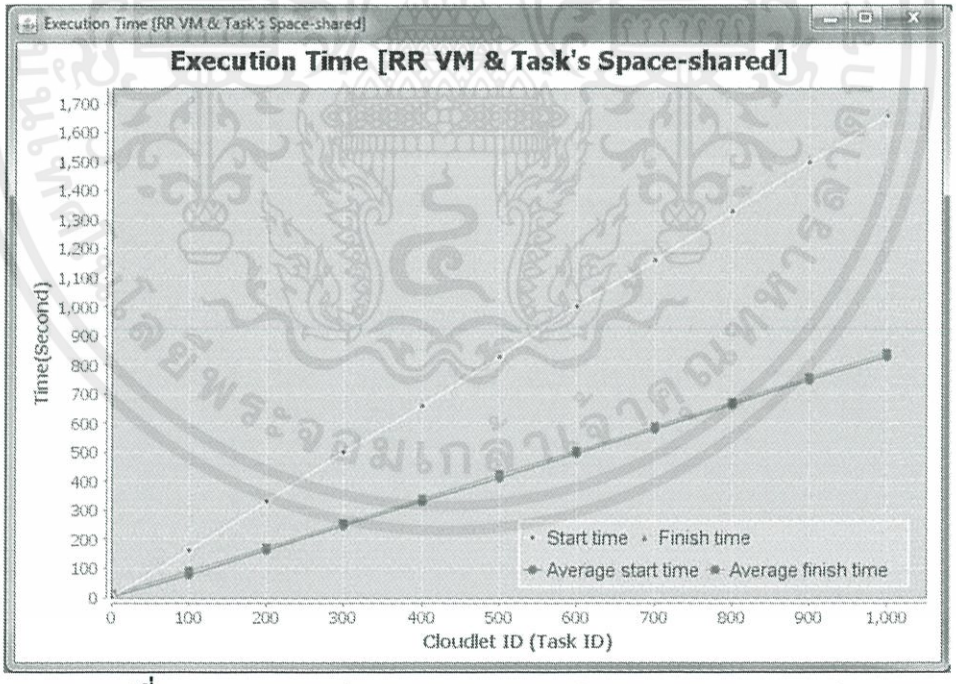


รูปที่ 4.17 แสดงกราฟ Load Balancing (Task: Time-Shared)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

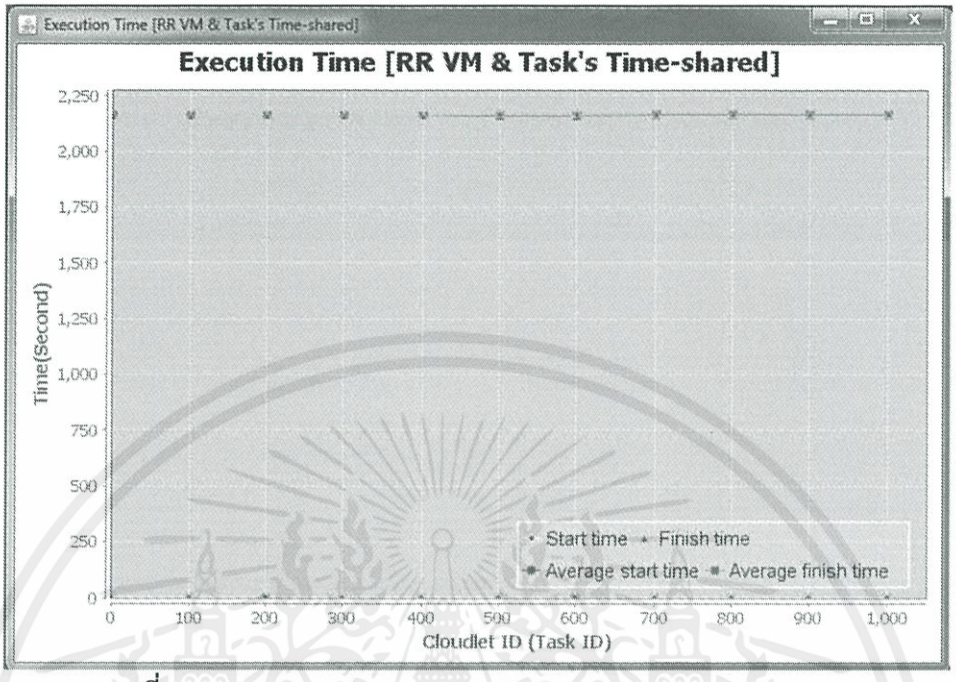


รูปที่ 4.18 แสดงกราฟ Load Balancing-VM ของ 1000 งาน



รูปที่ 4.19 แสดงกราฟ Execution Time (Task: Space-Shared)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



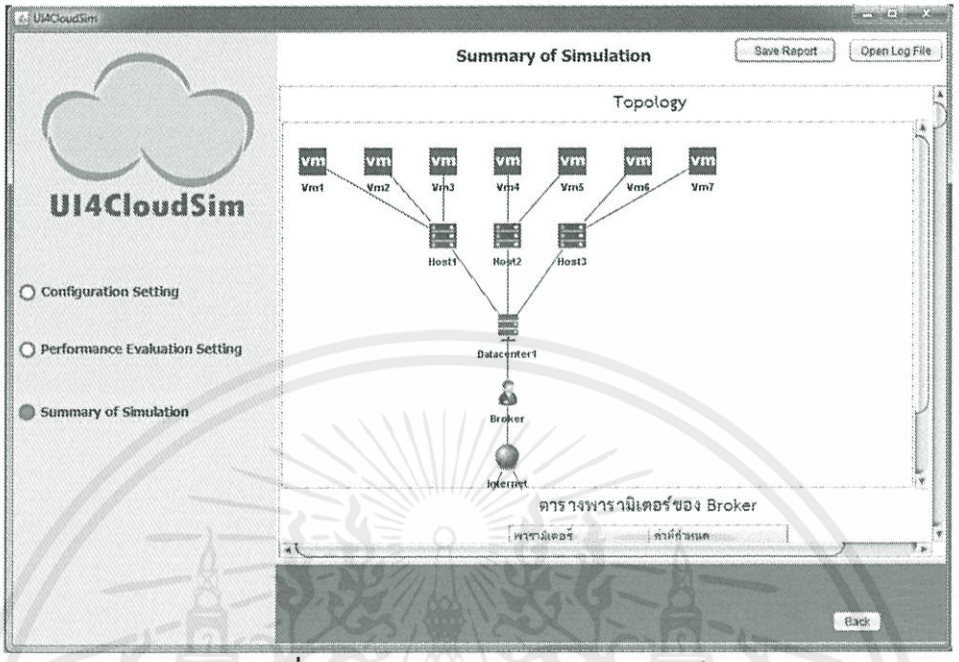
รูปที่ 4.20 แสดงกราฟ Execution Time (Task: Time-Shared)

หลังจากผู้ใช้เลือกแสดงกราฟตามที่ต้องการ จะเข้าสู่ขั้นตอนการสรุปผลการจำลอง โดยผู้ใช้กดปุ่ม Next เพื่อเข้าสู่ขั้นตอนที่ 3 “Summary of Simulation”

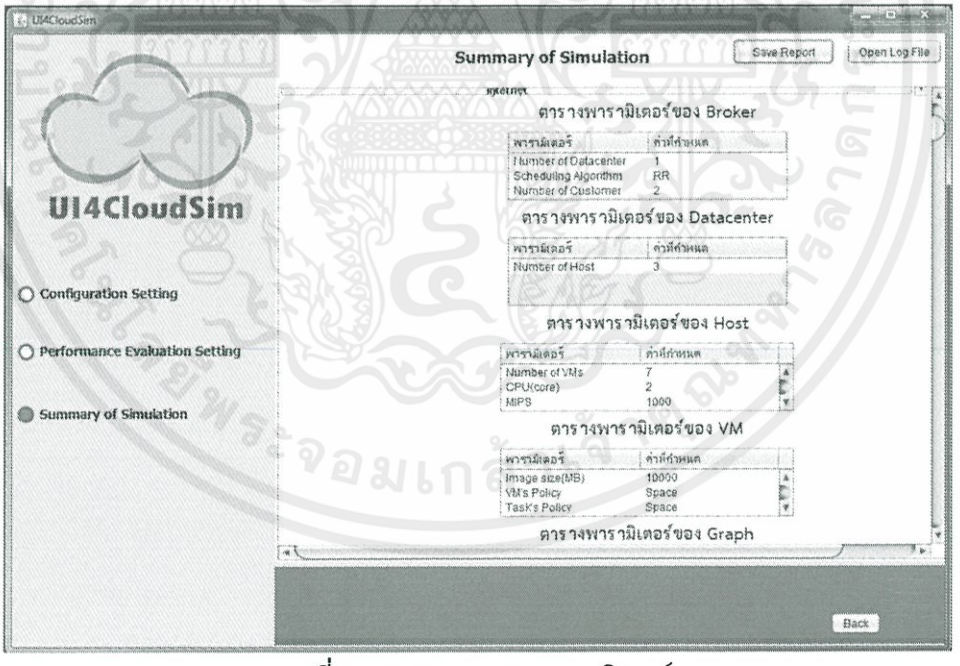
4.3 รายงานสรุปผลการจำลอง

ผู้ใช้จะได้รายงานสรุปผลการจำลองที่ประกอบด้วย Topology ของคลาวด์ ตารางพารามิเตอร์ และกราฟวัดประสิทธิภาพ ดังรูปที่ 4.21 – 4.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

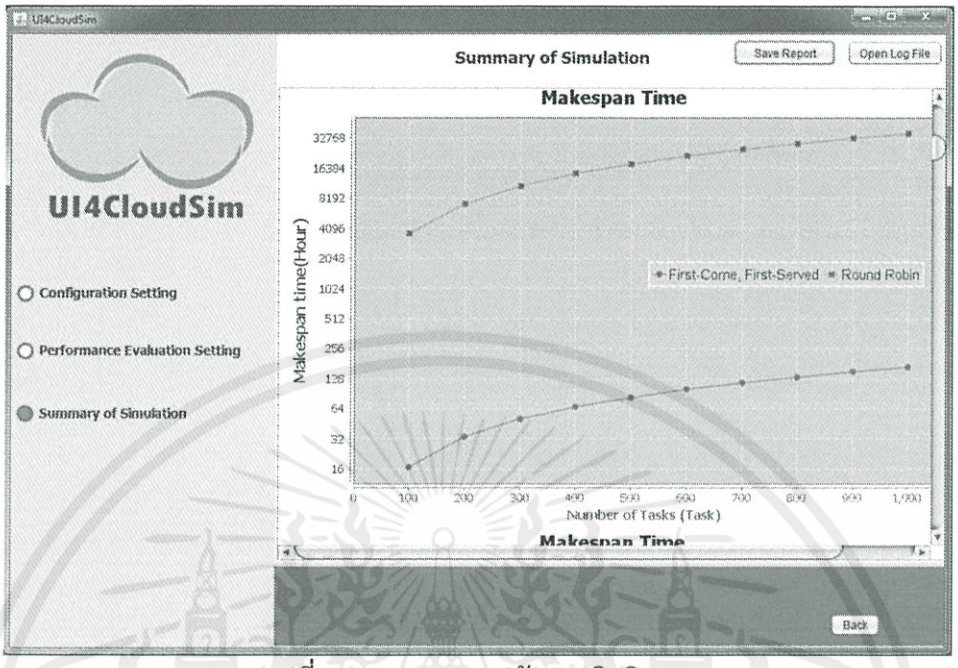


รูปที่ 4.21 แสดง Topology ของคลาวด์



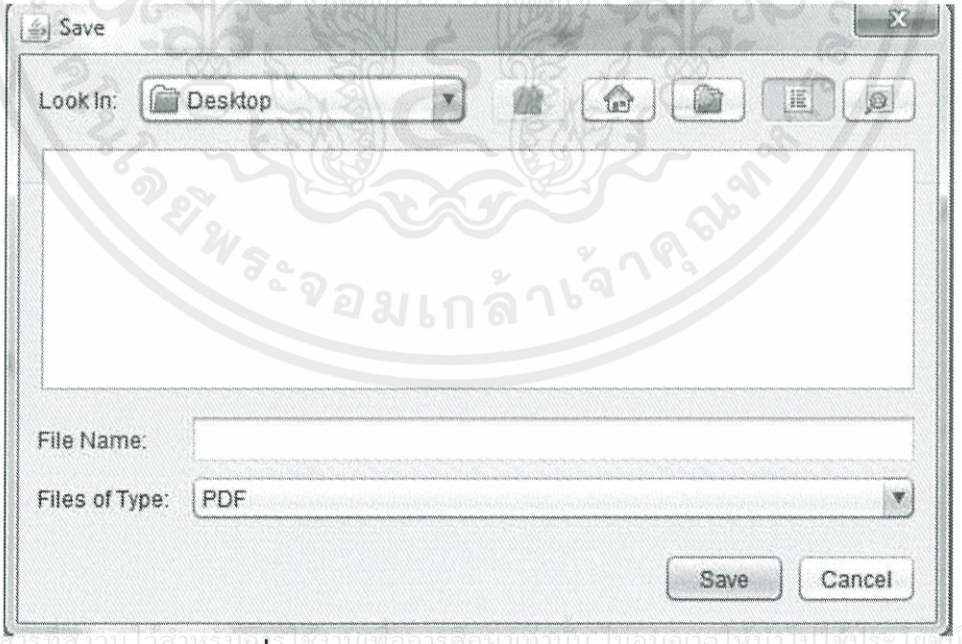
รูปที่ 4.22 แสดงตารางพารามิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 แสดงกราฟวัดประสิทธิภาพ

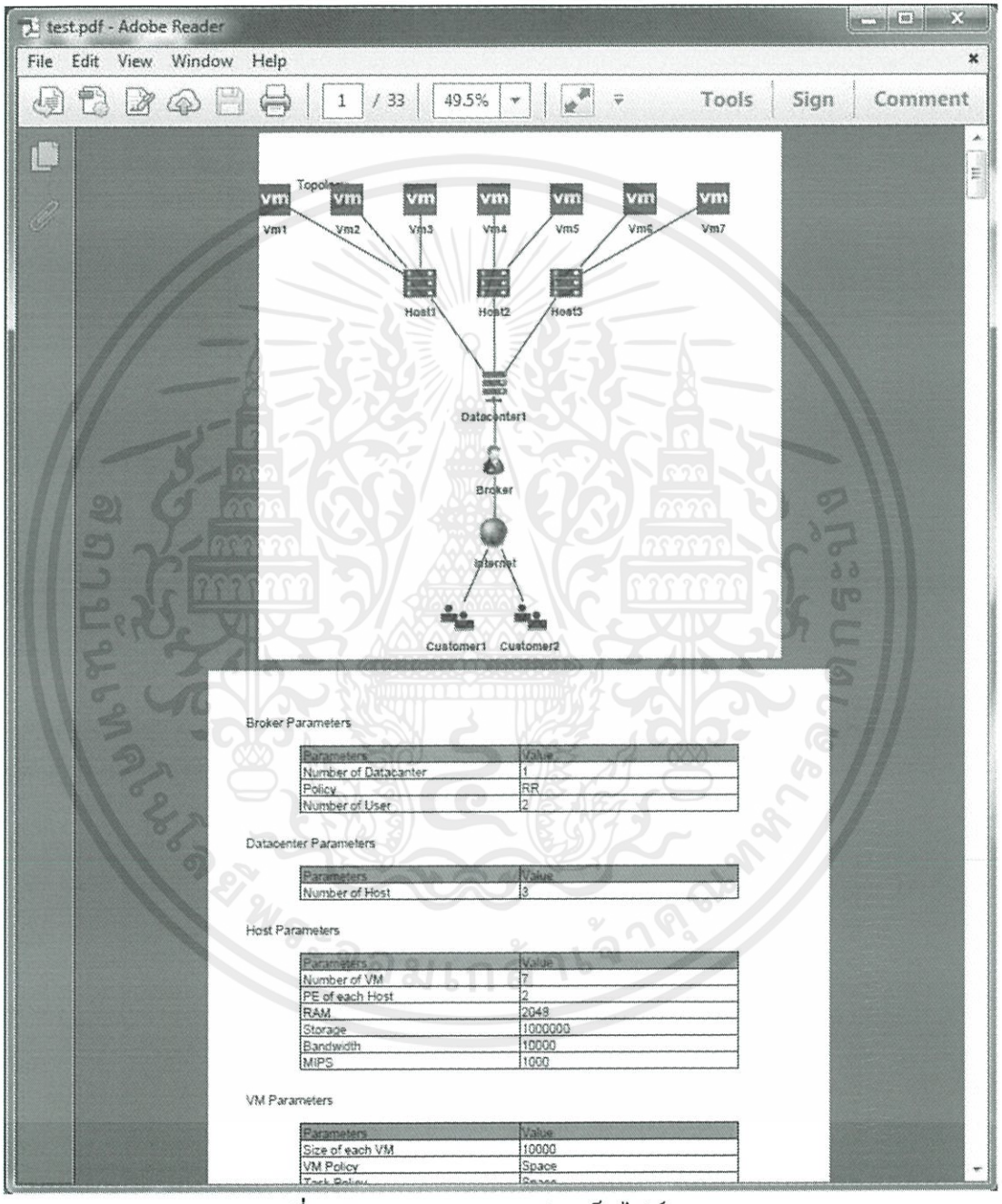
ผู้ใช้สามารถเลือกบันทึกรายงานสรุปผลได้ โดยกดปุ่ม “Save Report” ที่มุมขวาด้านบน จะแสดงหน้าต่างดังรูปที่ 4.24



รูปที่ 4.24 แสดงหน้าต่างบันทึกรายงานสรุปผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเลือก Location และ ตั้งชื่อไฟล์ แล้วกด “Save” จะได้ไฟล์รายงานเป็นไฟล์ PDF ดังรูปที่ 4.25

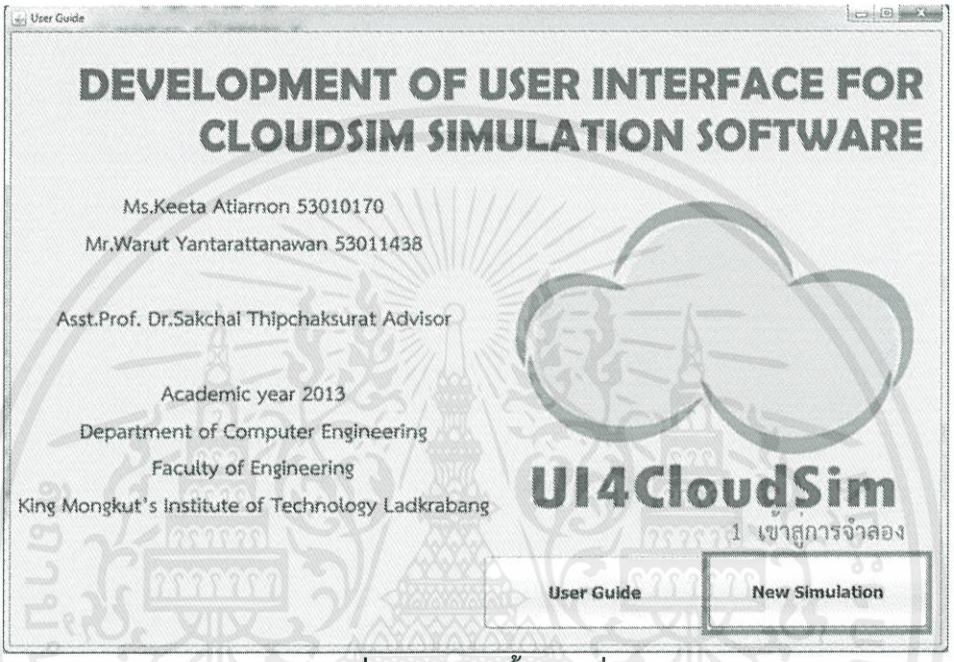


รูปที่ 4.25 แสดงรายงานสรุปผลเป็นไฟล์ PDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 คู่มือการใช้งาน

เพื่อให้ผู้ใช้สามารถเรียนรู้การใช้งานที่ถูกต้องของซอฟต์แวร์ โดยกดปุ่ม “User Guide” จะแสดงหน้าต่างดังรูปที่ 4.26 – 4.34

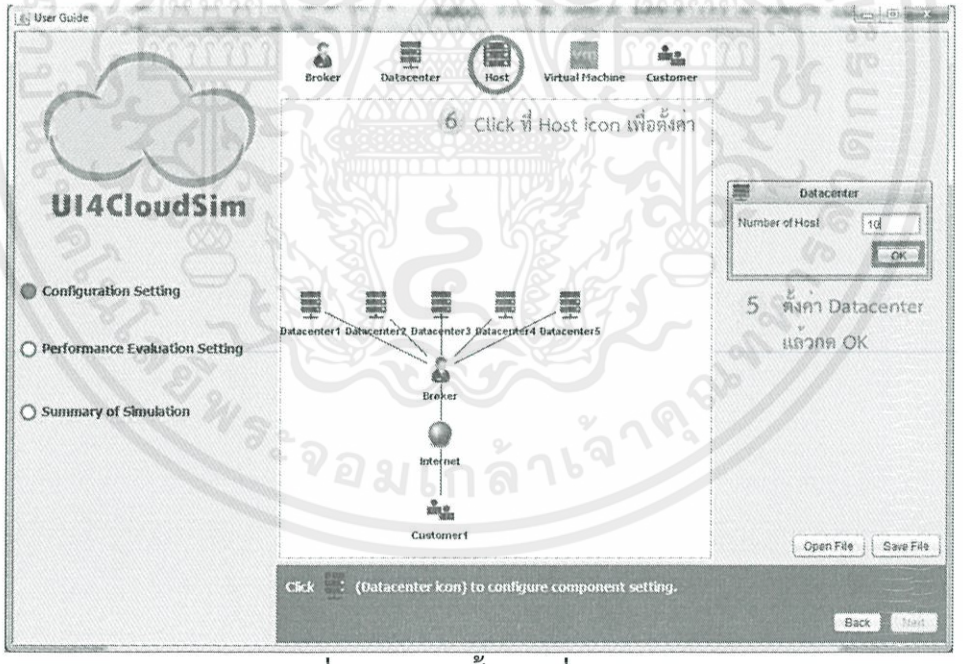


รูปที่ 4.26 แสดงขั้นตอนที่ 1

รูปที่ 4.27 แสดงขั้นตอนที่ 2

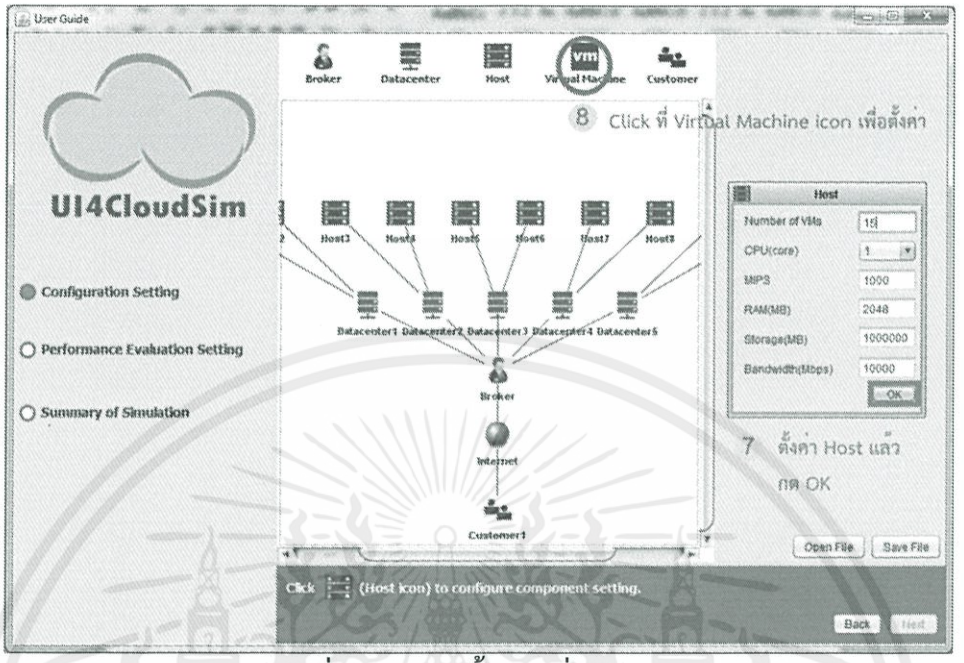
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.28 แสดงขั้นตอนที่ 3 และ 4

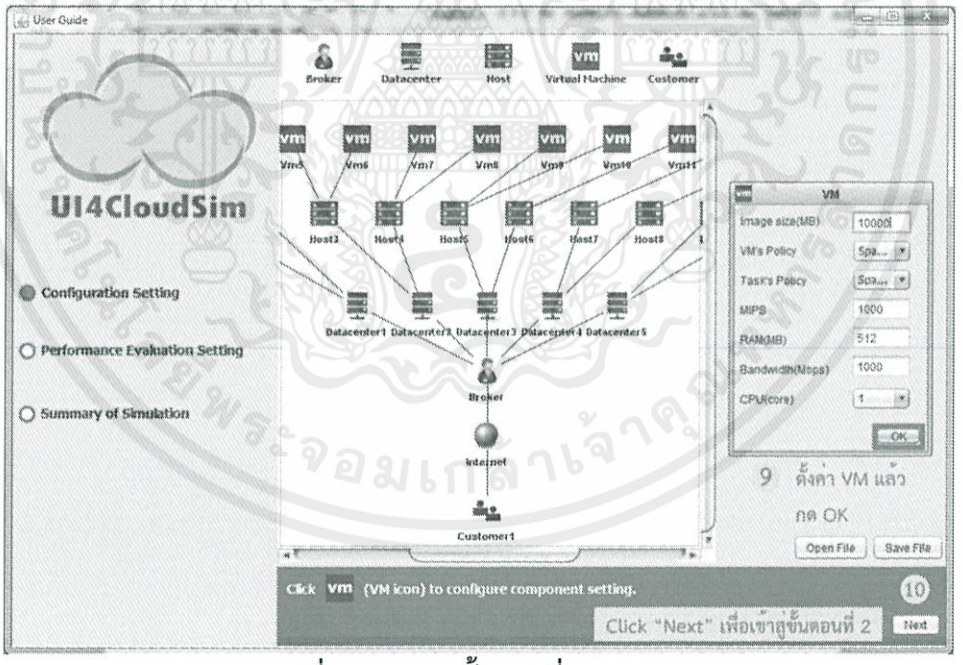


รูปที่ 4.29 แสดงขั้นตอนที่ 5 และ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

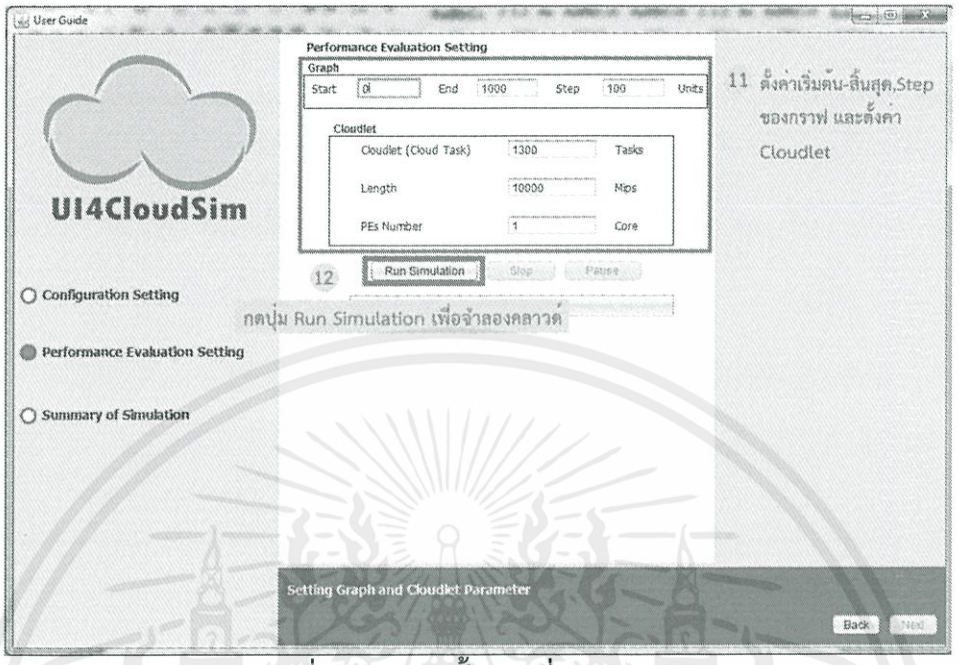


รูปที่ 4.30 แสดงขั้นตอนที่ 7 และ 8

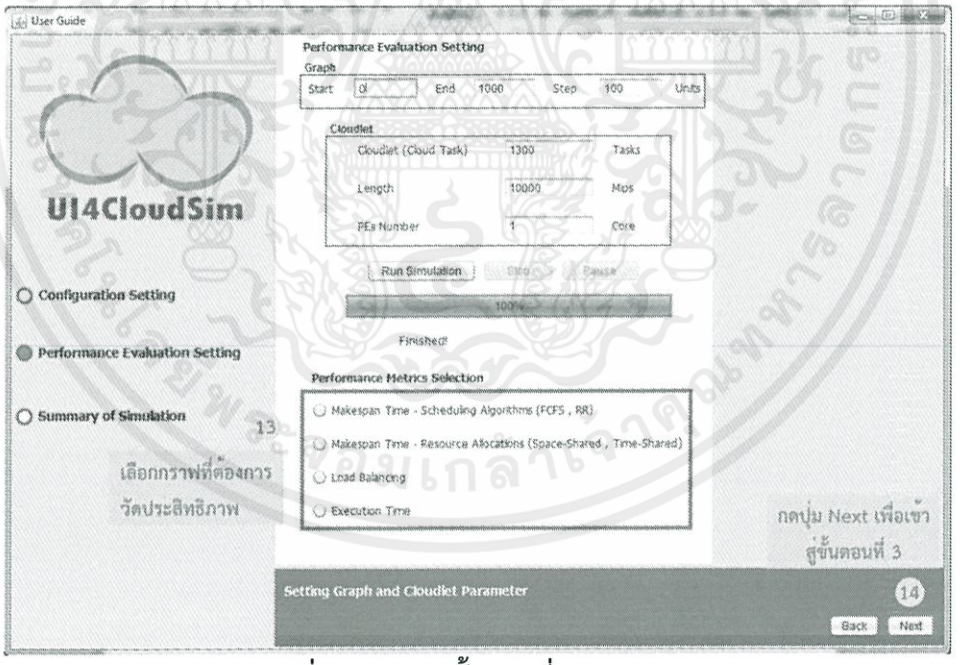


รูปที่ 4.31 แสดงขั้นตอนที่ 9 และ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

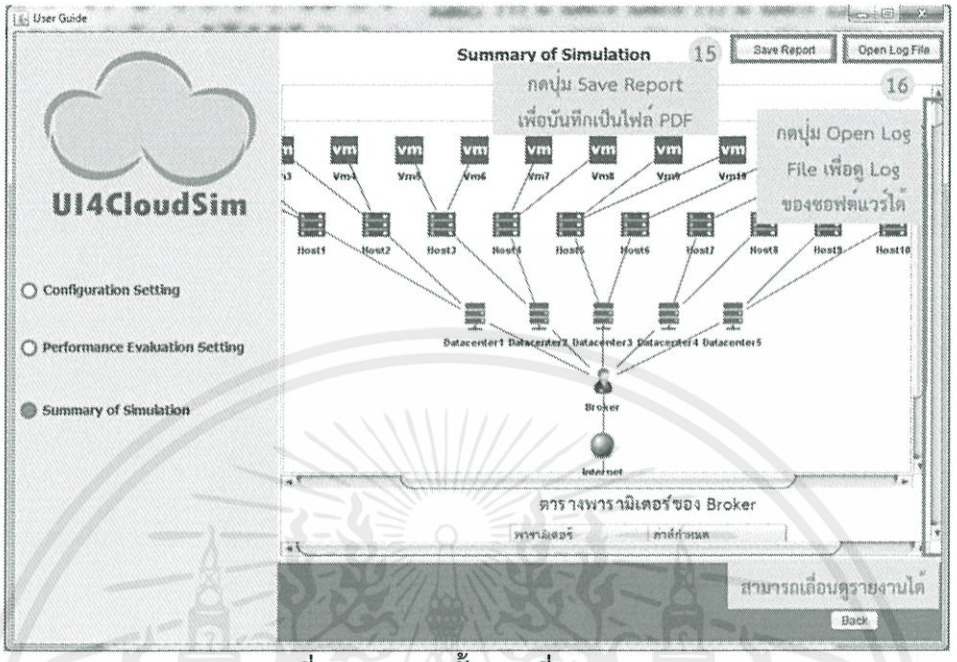


รูปที่ 4.32 แสดงขั้นตอนที่ 11 และ 12



รูปที่ 4.33 แสดงขั้นตอนที่ 13 และ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 แสดงขั้นตอนที่ 15 และ 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงบทสรุปของโครงงานในส่วนของการศึกษาและประยุกต์ใช้ทฤษฎีต่างๆที่เกี่ยวข้องกับการทดลอง ข้อมูลและผลการทดลอง ปัญหาอุปสรรคและแนวทางแก้ไข รวมถึงแนวทางในการพัฒนาต่อ

5.1 บทสรุปของโครงงาน

ยูไอฟอร์คลาวด์ซิม (UI4CloudSim) เป็นซอฟต์แวร์ที่พัฒนาขึ้นเพื่อจำลองการทำงานของคลาวด์ โดยซอฟต์แวร์ที่นำมาพัฒนาคือคลาวด์ซิม (CloudSim) ซึ่งเป็นซอฟต์แวร์ที่ทำงานด้วยภาษาจาวา คลาวด์ซิมจะจำลองการทำงานโดยผู้ใช้ต้องกำหนดค่าตัวแปรและสภาพแวดล้อมต่างๆสำหรับการจำลองการทำงาน ผู้ใช้จะทำการประมวลผลแค่ 1 ครั้ง โดยการกดปุ่ม “Run Simulation” แต่ภายใต้ซอฟต์แวร์นั้นจะทำการประมวลผลทั้งหมด 8 ครั้ง เพื่อให้ครอบคลุม Scheduling Policy และ Resource Allocation Policy ทุกนโยบาย ทำให้ผู้ใช้สามารถใช้งานได้สะดวกและประหยัดเวลา จากนั้นซอฟต์แวร์จะนำผลการจำลองที่ได้ไปแสดงกราฟวัดประสิทธิภาพ โดยผู้ใช้สามารถเลือกได้ว่าต้องการให้แสดงกราฟวัดประสิทธิภาพแบบใดบ้าง และจัดทำรายงานสรุปผลการจำลอง

ดังนั้นซอฟต์แวร์ที่พัฒนาขึ้นสามารถแก้ไขปัญหาการใช้งานที่ยุ่งยากของคลาวด์ซิมได้ และสามารถแสดงกราฟวัดประสิทธิภาพของคลาวด์ให้ผู้ใช้งานนำไปใช้ประกอบการตัดสินใจได้ เน้นการประหยัดเวลาในการศึกษาวิธีการใช้งานและไม่ต้องเสียค่าใช้จ่ายในการจำลองคลาวด์ โดยซอฟต์แวร์ที่พัฒนาขึ้นเหมาะสำหรับการใช้งานในองค์กรขนาดเล็กถึงปานกลาง เนื่องจากจะเห็นประสิทธิภาพได้อย่างชัดเจน ซอฟต์แวร์ยังสามารถพัฒนาให้เหมาะสำหรับองค์กรใหญ่ขึ้นได้ โดยการเพิ่มฟังก์ชันการทำงานให้ประมวลผลได้ในระดับสูง

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

1) เนื่องจากซอฟต์แวร์คลาวด์ซิมเป็นซอฟต์แวร์ที่ใช้ในการทดลองซึ่งไม่มีคำอธิบายการใช้งานและผลลัพธ์จากการทดลอง ทำให้ผู้จัดทำจำเป็นต้องหาแหล่งศึกษาข้อมูลและทดลองการทำงานเองจนเข้าใจการทำงานของซอฟต์แวร์คลาวด์ซิมอย่างถูกต้อง โดยอาศัยแหล่งอ้างอิงจากวิทยานิพนธ์ของมหาวิทยาลัยต่างๆที่มีการใช้ซอฟต์แวร์คลาวด์ซิมเป็นซอฟต์แวร์พื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การพัฒนาซอฟต์แวร์ลักษณะนี้สามารถออกแบบและทำได้ค่อนข้างยาก เนื่องจากการวัดหาประสิทธิภาพของการจำลองมีตัวแปรหลายอย่าง จึงเป็นการยากในการสรุปว่าผลการจำลองแบบใดมีประสิทธิภาพ ดังนั้นผู้จัดทำจึงทำการเปรียบเทียบกราฟให้ครอบคลุมทุกนโยบายการจัดสรรพื้นที่เข้าใช้งานและการจัดลำดับของงาน เพื่อให้ผู้ใช้งานสามารถตัดสินใจได้ว่าจะเลือกการทำงานแบบใด
3) เนื่องจากตัวอย่างในการพัฒนาซอฟต์แวร์ในลักษณะแสดงผลการจำลองมีน้อย ทำให้ผู้จัดทำต้องศึกษาข้อมูลและหาแนวทางการออกแบบด้วยตนเอง เพื่อให้ผลลัพธ์ตรงตามความต้องการของผู้ใช้ให้มากที่สุด โดยศึกษาจากฟังก์ชันการทำงานของภาษาจาวาแทน จากนั้นนำมาประยุกต์กับซอฟต์แวร์เพื่อแสดงผลให้ผู้ใช้เข้าใจได้ง่ายและสามารถนำไปใช้ประโยชน์ได้
4) ผู้จัดทำใช้เวลาในการศึกษาโครงสร้างของคลาวด์ซิมนาน เนื่องจากเป็นมีตัวแปรที่ใช้ในการกำหนดค่ามากและแต่ละตัวแปรมีผลกระทบถึงกัน ทำให้ผู้จัดทำต้องศึกษาทุกๆตัวแปร รวมถึงตัวแปรที่ไม่มีความสำคัญกับส่วนที่ต้องการศึกษา จึงเกิดความล่าช้าในการพัฒนาซอฟต์แวร์ ซึ่งผู้จัดทำได้แก้ปัญหาโดยแบ่งส่วนการทำงานของคลาวด์ซิมออกเป็นระบบๆ และเลือกเฉพาะระบบที่ต้องการพัฒนามาศึกษาเท่านั้น ทำให้สามารถลดระยะเวลาของการศึกษาโครงสร้างไปได้เป็นอย่างมาก
5) เนื่องจากข้อจำกัดของการใช้งานซอฟต์แวร์คลาวด์ซิม ทำให้ซอฟต์แวร์ทำงานด้วยภาษาจาวาเท่านั้น แต่ผู้จัดทำเลือกพัฒนาส่วนติดต่อผู้ใช้ขึ้นมา ผู้ใช้งานจึงไม่ต้องแก้ไขโค้ดหรือฟังก์ชันของซอฟต์แวร์ ทำให้สามารถใช้งานได้สะดวกยิ่งขึ้น และเพิ่มการทำงานในส่วนของผลลัพธ์ของการจำลอง เพื่อให้ผู้ใช้สามารถนำผลการจำลองมาวิเคราะห์ได้

5.3 แนวทางในการพัฒนาต่อ

1) พัฒนาซอฟต์แวร์ให้สามารถวัดประสิทธิภาพให้ได้หลายรูปแบบ เพื่อนำไปวิเคราะห์เชิงธุรกิจในการเลือกใช้งานจริงได้ เช่น ในด้านเครือข่าย จำลองการทำงานของแบนด์วิธว่าแบบใดจึงจะเหมาะสมกับงานของผู้ใช้ได้มากที่สุด, ในด้าน Latency (ความหน่วงเวลา) หรือ การหาเส้นทางในการเชื่อมต่อกับดาต้าเซ็นเตอร์ที่เหมาะสมที่สุด เป็นต้น
2) แสดงผลการจำลองคลาวด์ในรูปแบบที่เข้าใจได้ง่าย ผู้ใช้สามารถนำผลที่ได้ไปใช้ในการทำงานจริงได้ทันที โดยผู้ใช้ไม่จำเป็นต้นเข้าใจการทำงานคลาวด์อย่างลึกซึ้ง
3) พัฒนาอัลกอริทึมที่ใช้ในการประมวลผลให้ดีขึ้น เพื่อให้เกิดความรวดเร็วในการจำลองคลาวด์และเพิ่มความหลากหลายของฟังก์ชันการทำงาน เช่น สามารถแสดงที่ตั้งของดาต้าเซ็นเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ เพราะในบางครั้งดาต้าเซ็นเตอร์แต่ละจุดอาจไม่อยู่ใกล้กัน ทำให้ต้องมีการเลือกเส้นทางที่เหมาะสมที่จะส่งงานไปประมวลผล ณ ดาต้าเซ็นเตอร์นั้น

4) พัฒนาซอฟต์แวร์ให้สามารถรองรับขนาดใหญ่ได้ โดยใช้เวลาในการประมวลผลน้อย
5) เพิ่มขอบเขตการทำงานให้สามารถจัดการลำดับของงานตามความต้องการของผู้ใช้ได้
6) ทำให้ซอฟต์แวร์สามารถแสดงสถานะของงานขณะกำลังประมวลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Navaroj (นามแฝง). "**ประวัติ Cloud Computing**" [Online]. Available: http://navarojch.blogspot.com/2009/08/cloud-computing_3891.html. 2552.

[2] วิกิพีเดีย สารานุกรมเสรี. "**การประมวลผลแบบกลุ่มเมฆ**" [Online]. Available : http://th.wikipedia.org/wiki/การประมวลผลแบบกลุ่มเมฆ. 2556.

[3] Mary Jo Foley. "**Where in the world are Microsoft's datacenters?**" [Online]. Available: http://www.zdnet.com/blog/microsoft/where-in-the-world-are-microsofts-datacenters/5700

[4] Niharika Trivedi. "**Ibm invests 1.2 billion to boost global cloud footprint**" [Online]. Available : https://www.ibm.com/developerworks/community/blogs/

[5] Google Team. "**Data center locations**" [Online]. Available: http://www.google.com/about/datacenters/inside/locations/

[6] Amazon Team. "**AWS Edge Locations**" [Online]. Available: http://aws.amazon.com/about-aws/globalinfrastructure/regional-product-services/

[7] Dean Nelson. "**eBay Breaks New Ground in Data Center Efficiency**" [Online]. Available: https://green.ebay.com/greenteam/blog/eBay-Breaks-New-Ground-in-Data-Center-Efficiency-/8361

[8] Jean Paul Cozzatti. "**A Twitter data center**" [Online]. Available: https://blog.twitter.com/2010/room-grow-twitter-data-center

[9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", **Software: Practice and Experience (SPE),** Volume 41, Number 1, 2011, Pages: 23-50.

[10] Rodrigo N. Calheiros, Rajiv Ranjan, César A. F. De Rose and Rajkumar Buyya. "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services". Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, 2009.

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆทั้งนี้... ...ไปใช้ประโยชน์ด้านการค้า ...ไปใช้

[11] Sathish (นามแฝง). "**Differentiate JVM JRE JDK JIT**" [Online]. Available: http://www.maanavan.com/differentiate-jvm-jre-jdk-jit. 2012.

[12] Kelton ,D.W., Sadowski , R.P. and Sturrock D.T. "**Simulation with Arena**"-3rd ed., International Edition, McGraw-Hill, The McGraw-Hill Company, Inc. 2003.

[13] Gaurav Raj and Sonika Setia, Asst. Prof., Lovely Professional University, Phagwara, "Effective Cost Mechanism for Cloudlet Retransmission and Prioritized VM Scheduling Mechanism over Broker Virtual Machine Communication Framework" **International Journal on Cloud Computing: Services and Architecture (IJCCSA)**, Vol.2, No.3, June 2012.

[14] Sumit Khurana and Khyati Marwah, "Performance evaluation of Virtual Machine(VM) Scheduling Policies in Cloud Computing (Space Shared & Time Shared)" **IEEE – 31661, 4th ICCCNT - 2013 July 4 - 6, 2013**, Tiruchengode, India.

[15] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang, "Cloud Task scheduling based on Load Balancing Ant Colony Optimization" **2011 Sixth Annual ChinaGrid Conference. ChainaGrid 2011.17.**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้