

ระบบตรวจสอบโครงสร้างพื้นฐานคลาวด์

MONITORING SYSTEM FOR CLOUD INFRASTRUCTURE



ปริญญานิพนธ์นี้เป็นต้นฉบับเป็นส่วนหนึ่งของรายงานการศึกษาค้นคว้าวิจัยที่ผู้สมัครเป็นวิทยานิพนธ์วิชาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
ศูนย์วิศวกรรมศาสตรบัณฑิต
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ศูนย์วิศวกรรมศาสตรบัณฑิต

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปริญญานิพนธ์นี้เป็นต้นฉบับเป็นส่วนหนึ่งของรายงานการศึกษาค้นคว้าวิจัยที่ผู้สมัครเป็นวิทยานิพนธ์วิชาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์
ศูนย์วิศวกรรมศาสตรบัณฑิต
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ระบบดูแลโครงสร้างคลาวด์
Monitoring System for Cloud Infrastructure



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบดูแลโครงสร้างคลาวด์

Monitoring System for Cloud Infrastructure

ผู้จัดทำ

- | | | | |
|-----------------|-----------------|--------------|----------|
| 1. นายธนทัต | ใจการุณย์ | รหัสนักศึกษา | 53010623 |
| 2. นายธนาศาสตร์ | อุ๋ทรัพย์ศาสตร์ | รหัสนักศึกษา | 53010666 |



Orathai Saksri อาจารย์ที่ปรึกษา
(ดร.อรัทัย สังข์เพชร)

Ar. A. อาจารย์ที่ปรึกษาร่วม
(ดร.อัครฤทธิ์ สังข์เพชร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบดูแลโครงสร้างคลาวด์

นาย ธนทัต	ใจการุณย์	53010623
นาย ธนศาสตร์	อุทัยพาศาสตร์	53010666
ดร.อรทัย	สังข์เพชร	อาจารย์ที่ปรึกษา
ดร.อภฤทธิ์	สังข์เพชร	อาจารย์ที่ปรึกษาร่วม
ปีการศึกษา 2556		

บทคัดย่อ

ระบบดูแลโครงสร้างคลาวด์ เป็นการดึงข้อมูลการใช้ทรัพยากรของเครื่องคอมพิวเตอร์เสมือน และของเครื่องเซิร์ฟเวอร์ เนื่องจากในการจัดสรรทรัพยากรของเครื่องเซิร์ฟเวอร์ให้แก่เครื่องคอมพิวเตอร์เสมือนแต่ละตัวนั้น ผู้ให้บริการจำเป็นต้องทราบถึงพฤติกรรมและการใช้งานจริงของเครื่องคอมพิวเตอร์เสมือนแต่ละตัวเพื่อให้สามารถใช้ประโยชน์ของทรัพยากรได้สูงสุด ดังนั้นโครงการเรื่องระบบดูแลโครงสร้างคลาวด์ จึงจัดทำขึ้นเพื่อตอบสนองความต้องการของผู้ให้บริการคลาวด์ในเรื่องของการแสดงข้อมูลการใช้งานทรัพยากร ของเครื่องคอมพิวเตอร์เสมือนและเครื่องเซิร์ฟเวอร์เพื่อใช้ในการจัดสรรทรัพยากรได้อย่างมีประสิทธิภาพ อีกทั้งเพื่อให้ผู้ใช้บริการคลาวด์ทราบถึงทรัพยากรที่เครื่องคอมพิวเตอร์เสมือนของตนใช้งานอยู่ เพื่อช่วยในการวางแผนจัดการการจัดสรรทรัพยากรของเครื่องคอมพิวเตอร์เสมือนได้เหมาะสมกับการใช้งานจริง ระบบดูแลโครงสร้างคลาวด์ที่จะจัดทำขึ้นนี้ แบ่งการทำงานออกเป็นสามส่วน ได้แก่ ส่วนของการดึงข้อมูล ส่วนนี้จะทำการส่งข้อมูลของเครื่องคอมพิวเตอร์เสมือนและเครื่องเซิร์ฟเวอร์ แต่ละตัวเพื่อให้ทราบว่ารายละเอียดการใช้ทรัพยากรเช่น อัตราส่วนร้อยละของการทำงานซีพียู ข้อมูลการใช้งานทรัพยากรหน่วยความจำ ข้อมูลการใช้งานทรัพยากรดิสก์ ข้อมูลการใช้งานทรัพยากรเน็ตเวิร์ค ระยะเวลาของการใช้งานเครื่องเซิร์ฟเวอร์ จำนวนซีพียู เวลาในขณะที่ยิงข้อมูล ส่วนที่สองคือการจัดเก็บข้อมูล จะทำการรับข้อมูลมาเก็บไว้ในฐานข้อมูล ส่วนสุดท้ายคือการแสดงข้อมูล จะนำข้อมูลที่จัดเก็บไว้มาแสดงในรูปแบบของเว็บแอปพลิเคชันเพื่อให้ง่ายต่อการเข้าใจและใช้งาน

Monitoring System for Cloud Infrastructure

Mr. Thanatat	Jaikaroon	53010623
Mr. Thanasart	Usappayasart	53010666
Dr. Orathai	Sangpetch	Advisor
Dr. Akkarit	Sangpetch	Co-advisor

Academic Year 2013

ABSTRACT

Cloud infrastructure monitoring system provides the resource usage information of virtual machines and physical servers. In order to efficiently manage and allocate physical resources to each virtual machine, a cloud service provider needs to know actual resource usage of every single virtual machine. The objective of this project is to show the resource usage information of individual virtual machines and physical servers in real-time. The information does not only help cloud providers manage physical resources efficiently but also allow cloud customers to visualize the real-time resource usage of their virtual machines. The cloud infrastructure monitoring system consists of three parts. The first part is the information retrieval module which gets the resource usage details of each physical server in real-time, such as percentage of CPU utilization, memory usage, disk usage, network information, uptime of a physical server, and the number of CPUs. The second part is the database which stores the data retrieved by the information retrieval module. And the last one is the information visualization which shows the retrieved data in the graphical format delivered by web application.

||

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดีด้วยคำแนะนำและคำปรึกษาเกี่ยวกับการดำเนินงาน การศึกษาและวิจัย โดยได้รับความช่วยเหลือจากบุคลากรดังต่อไปนี้

ขอขอบพระคุณ ดร.อรทัย สังข์เพชร ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการ และ ดร.อภิญญา สังข์เพชร ซึ่งเป็นอาจารย์ที่ปรึกษาร่วม ที่ได้ให้ความรู้ คำแนะนำต่าง ๆ อีกทั้งความช่วยเหลือและแนวทางในการแก้ปัญหาที่พบระหว่างการทำโครงการชิ้นนี้ จนทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณห้องวิจัย ISAG สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ได้สนับสนุนเครื่องมือ ตลอดจน ข้อมูล และหนังสือต่าง ๆ ตลอดจนเพื่อน ๆ พี่ ๆ น้อง ๆ ที่อยู่ร่วมกันมา คอยให้กำลังใจและคำแนะนำ เสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ผู้ซึ่งเป็นทั้งกำลังใจ ตลอดจนการให้การสนับสนุนในทุก ๆ เรื่อง

นาย ธนทัต

ใจการุณย์

นาย ธนศาสตร์

อุทัยพาศาสตร์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป	VI
สารบัญตาราง.....	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินการ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 ส่วนประกอบของปริญญาานิพนธ์.....	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 คลาวด์คอมพิวติ้ง (Cloud Computing).....	5
2.1.1 เวอร์ชวลไลเซชัน (Virtualization)	5
2.1.2 ลำดับชั้นของการให้บริการคลาวด์คอมพิวติ้ง (Cloud Computing Layers).....	6
2.1.2.1 ซอฟต์แวร์ แอส อะ เซอร์วิส	6
2.1.2.2 แพลตฟอร์ม แอส อะ เซอร์วิส	7
2.1.2.3 อินฟราสตรัคเจอร์ แอส อะ เซอร์วิส	7
2.1.3 ประเภทของคลาวด์คอมพิวติ้ง (Cloud Computing Types).....	7
2.1.3.1 คลาวด์แบบสาธารณะ.....	8
2.1.3.2 คลาวด์แบบเฉพาะองค์กร	8
2.1.3.3 คลาวด์แบบผสม	9
2.1.3.4 คลาวด์แบบใช้ระหว่างองค์กร	9

2.1.4 คลาวด์เซิร์ฟเวอร์ (Cloud server).....	9
2.2 ดาต้าเบส แบบ โนเอสคิวแอล (NoSQL Database).....	9
2.3 ภาษาไพทอน (Python programming language).....	10
2.4 ภาษาวินโดวส์ พาวเวอร์เชลล์ (Windows Powershell).....	10
2.5 ภาษาแอชทีเอ็มแอล (HTML)	11
2.6 ภาษาจาวาสคริปต์ (JavaScript).....	11
2.7 ผลิตภัณฑ์ที่เกี่ยวข้อง (Related Works).....	11
บทที่ 3 การออกแบบและพัฒนา.....	12
3.1 ภาพรวมและองค์ประกอบของระบบ	12
3.1.1 การดึงข้อมูลจากเครื่องเซิร์ฟเวอร์แต่ละตัว	15
3.1.1.1 ข้อมูลที่มีการเปลี่ยนแปลงน้อย.....	15
3.1.1.2 ข้อมูลที่เป็นเรียลไทม์.....	15
3.2 การเก็บข้อมูล.....	17
3.3 การแสดงข้อมูล.....	19
บทที่ 4 การทดลอง	22
4.1 การทดลองการส่งข้อมูลเพื่อตรวจสอบจำนวนสตอเรจที่ใช้งาน	22
บทที่ 5 สรุปผลการทดลอง.....	25
บรรณานุกรม.....	26

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 ภาพแสดงการเปรียบเทียบระหว่างระบบปกติกับระบบที่ใช้งาน Virtualization [2]	5
รูปที่ 2.2 ภาพแสดงลำดับขั้นตอนของการให้บริการผู้ใช้งานในคลาวด์คอมพิวติ้ง [3]	6
รูปที่ 2.3 ภาพแสดงลำดับขั้นตอนของการให้บริการผู้ใช้งานในคลาวด์คอมพิวติ้ง [3]	8
รูปที่ 3.1 โครงสร้างของระบบดูแลโครงสร้างคลาวด์.....	13
รูปที่ 3.2 แสดงหน้าเพจหลักของเว็บแอปพลิเคชัน	19
รูปที่ 3.3 แสดงหน้าเพจของเครื่องเซิร์ฟเวอร์บนเว็บแอปพลิเคชัน	20
รูปที่ 3.4 แสดงหน้าเพจของเครื่องคอมพิวเตอร์เสมือนบนเว็บแอปพลิเคชัน	21
รูปที่ 4.1 ภาพแสดงข้อมูลการใช้งานซีพียู.....	22
รูปที่ 4.2 ภาพแสดงข้อมูลจำนวนซีพียู.....	22
รูปที่ 4.3 ภาพแสดงข้อมูลระยะเวลาในการเปิดเครื่อง	22
รูปที่ 4.4 ภาพแสดงข้อมูลการใช้งานหน่วยความจำ.....	23
รูปที่ 4.5 ภาพแสดงข้อมูลการใช้งานเน็ตเวิร์ค	23
รูปที่ 4.6 ภาพแสดงข้อมูลจำนวนซีพียู.....	23
รูปที่ 4.7 ภาพแสดงข้อมูลเวลาขณะที่ทำการดึงข้อมูล	23
รูปที่ 4.8 ภาพแสดงข้อมูลปริมาณสตอเรจที่ใช้งาน	24

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3.1 แสดงการใช้คำสั่งในการดึงข้อมูลทรัพยากรจากไฮเปอร์ไวเซอร์ KVM และ VMware ESX.....	16
ตารางที่ 3.2 แสดงการเก็บข้อมูลในคอลเลคชันแบบที่หนึ่ง.....	17
ตารางที่ 3.3 แสดงการเก็บข้อมูลในคอลเลคชันแบบที่สอง.....	17
ตารางที่ 3.4 แสดงการเก็บข้อมูลในคอลเลคชันแบบที่สาม (เครื่องคอมพิวเตอร์เสมือน).....	18
ตารางที่ 3.5 แสดงการเก็บข้อมูลในคอลเลคชันแบบที่สาม (เครื่องเซิร์ฟเวอร์).....	18



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

ในปัจจุบันการให้บริการเครื่องเซิร์ฟเวอร์ (server) มีอย่างแพร่หลายเนื่องจากการทำงานโดยใช้ทรัพยากรของเซิร์ฟเวอร์ที่มีอยู่ให้เกิดประโยชน์สูงสุด โดยมีไฮเปอร์ไวเซอร์(hypervisor) เป็นตัวจัดการแบ่งปันทรัพยากรต่างๆของเซิร์ฟเวอร์ให้แก่หลายๆเครื่องคอมพิวเตอร์เสมือน (virtual machine) และในการแบ่งปันหรือการจัดสรรทรัพยากรนั้นผู้ให้จำเป็นต้องรู้พฤติกรรมในการทำงานของเครื่องคอมพิวเตอร์เสมือนนั้นๆ เช่น ผู้ให้บริการเครื่องเซิร์ฟเวอร์มีทรัพยากรเป็น ซีพียู 2คอร์และมีหน่วยความจำ 2 จิกะไบต์จะเห็นได้ว่าผู้ให้บริการสามารถให้บริการเครื่องคอมพิวเตอร์เสมือน 2 ตัวที่มีความต้องการทรัพยากร ซีพียู 1คอร์และหน่วยความจำ 1 จิกะไบต์ ได้และในกรณีที่ต้องการเพิ่มเครื่องคอมพิวเตอร์เสมือนที่มีความต้องการทรัพยากรเหมือนเดิมนั้นไม่สามารถทำได้เพราะไม่สามารถทราบได้ว่าเครื่องคอมพิวเตอร์เสมือนทั้ง2ตัวมีพฤติกรรมการใช้ทรัพยากรอย่างไร แต่หากผู้ให้บริการเซิร์ฟเวอร์ทราบถึงพฤติกรรมการใช้งานของเครื่องคอมพิวเตอร์เสมือนทั้งสองตัวว่า เครื่องคอมพิวเตอร์เสมือนตัวที่หนึ่งมีการใช้งานเฉพาะช่วงเวลา 8.00-18.00 น. และเครื่องคอมพิวเตอร์เสมือนตัวที่สองมีการใช้งานเฉพาะช่วงเวลา 19.00-4.00 น. จะสามารถทำให้เครื่องเซิร์ฟเวอร์นั้นให้บริการเครื่องคอมพิวเตอร์เสมือนตัวที่สามที่ต้องการซีพียู 1คอร์และหน่วยความจำ 1 จิกะไบต์ ได้ เพราะถึงแม้ว่าเครื่องคอมพิวเตอร์เสมือนตัวที่หนึ่งและสองใช้งานทรัพยากรเต็มประสิทธิภาพแต่เวลาที่ใช้นั้นเป็นคนละช่วงเวลากัน จึงทำส่งผลให้เครื่องเซิร์ฟเวอร์มีทรัพยากรเพียงพอที่จะสามารถให้บริการซีพียู 1คอร์และหน่วยความจำ 1 จิกะไบต์ในทุกช่วงเวลา

จะเห็นได้ว่าในการจัดสรรทรัพยากรให้ผู้ให้บริการของผู้ให้บริการคลาวด์คอมพิวเตอร์ตั้ง นั้นจำเป็นต้องทราบรายละเอียดของระบบและการใช้ทรัพยากรในระบบนั้นๆ เพื่อให้เกิดประโยชน์สูงสุดในการใช้งานเครื่องเซิร์ฟเวอร์ จึงได้จัดทำ ระบบดูแลโครงสร้างคลาวด์ (Monitoring System for Cloud Infrastructure) โดย ระบบดูแลโครงสร้างคลาวด์ จะสามารถแสดงรายละเอียดการใช้งานทรัพยากรของระบบแบบเรียลไทม์ (Real time) และง่ายต่อการใช้งาน

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อออกแบบและพัฒนาระบบที่สามารถแสดงรายละเอียด การใช้ทรัพยากรของทั้งเครื่องคอมพิวเตอร์เสมือนและเครื่องเซิร์ฟเวอร์แบบเรียลไทม์
- 2) เพื่อให้ผู้ใช้งานโครงการที่เป็นผู้ให้บริการคลาวด์ ทราบรายละเอียดการใช้ของทรัพยากรของเวอร์ชวลแมชชีนและเครื่องเซิร์ฟเวอร์ นำข้อมูลที่ได้ใช้ในการบริหารจัดการทรัพยากรคลาวด์ได้อย่างมีประสิทธิภาพ
- 3) เพื่อศึกษาและพัฒนาการนำเสนอข้อมูล ในรูปแบบเว็บแอปพลิเคชัน (web application) ซึ่งง่ายต่อการเข้าใจและใช้งาน

1.3 ขอบเขตของโครงการ

พัฒนาระบบที่สามารถแสดงรายละเอียดการใช้ทรัพยากรของระบบแบบเรียลไทม์ที่ถูกต้อง และง่ายต่อการใช้งานซึ่งระบบแบ่งออกเป็น 3 ส่วนดังนี้

ส่วนของการดึงข้อมูล ส่วนนี้จะทำการดึงข้อมูลของเครื่องเซิร์ฟเวอร์ โดยเครื่องเซิร์ฟเวอร์จะใช้ไฮเปอร์ไวเซอร์ VMware ESX หรือ KVM เพื่อให้ทราบว่ารายละเอียดการใช้งานทรัพยากรเป็นอย่างไร โดยส่วนนี้จะทำการดึงข้อมูลที่ต้องการของเครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือนทุกตัวและส่งข้อมูลเหล่านั้นไปที่ระบบจัดเก็บข้อมูล โดยข้อมูลที่ทำการดึงมานั้นจะมีดังนี้

- 1) ข้อมูลและปริมาณการใช้งานซีพียู
- 2) ข้อมูลและปริมาณการใช้งานหน่วยความจำ
- 3) ข้อมูลและปริมาณการใช้งานของดิสก์
- 4) ข้อมูลและปริมาณการใช้งานของเน็ตเวิร์ค
- 5) ระยะเวลาของการใช้งานเครื่องเซิร์ฟเวอร์
- 6) เวลาในขณะที่ดึงข้อมูล

ส่วนของการจัดเก็บข้อมูล จะทำการรับข้อมูลมาเก็บไว้ในฐานข้อมูล เนื่องจากมีการรับและจัดเก็บข้อมูลแบบเรียลไทม์จึงทำให้มีปริมาณข้อมูลที่มาก ในการจัดเก็บนั้นจึงจำเป็นต้องใช้ฐานข้อมูลที่สามารถรองรับการขยายตัวของข้อมูลได้

ส่วนของการแสดงข้อมูล จะนำข้อมูลที่จัดเก็บไว้มาแสดงในรูปแบบของเว็บแอปพลิเคชัน เพื่อให้ง่ายต่อการเข้าใจและใช้งานของผู้ดูแลระบบ

1.4 วิธีการดำเนินการ

- 1) ศึกษารายละเอียดของระบบคลาวด์คอมพิวติ้ง รวมทั้งศึกษาข้อมูลที่มีความจำเป็นต่อการจัดสรรทรัพยากร
- 2) ศึกษาเกี่ยวกับภาษาที่ใช้สำหรับการเขียนโปรแกรมดึงข้อมูลต่างๆของเครื่องคอมพิวเตอร์เสมือนและเครื่องเซิร์ฟเวอร์ ภาษาที่ใช้คือวินโดวส์ พาวเวอร์เชลล์ (Windows PowerShell) สำหรับการดึงข้อมูลจากไฮเปอร์ไวเซอร์ VMware ESX และภาษาไพธอน (Python) ในการเขียนไพธอนสคริปต์ (Python Script) สำหรับการดึงข้อมูลจากไฮเปอร์ไวเซอร์ KVM
- 3) ศึกษาเกี่ยวกับการส่งข้อมูลทรัพยากรระหว่างเซิร์ฟเวอร์ที่ให้บริการคลาวด์และดาต้าเบสเซิร์ฟเวอร์
- 4) ออกแบบและพัฒนารูปแบบการดึงและการส่งข้อมูลทรัพยากรระหว่างเซิร์ฟเวอร์ที่ให้บริการคลาวด์และดาต้าเบสเซิร์ฟเวอร์
- 5) วิเคราะห์และออกแบบฐานข้อมูลเพื่อเก็บข้อมูลการใช้ทรัพยากรของเซิร์ฟเวอร์และคอมพิวเตอร์เสมือน
- 6) ศึกษาเกี่ยวกับภาษาที่ใช้เขียนเพื่อแสดงผลบนเว็บแอปพลิเคชันรวมถึงการใช้ข้อมูลจากฐานข้อมูล
- 7) ออกแบบและพัฒนาเว็บแอปพลิเคชันที่ใช้ในการแสดงผลข้อมูล
- 8) วิเคราะห์ผลการทำงานของระบบที่พัฒนา และแก้ไขข้อผิดพลาด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ผู้ใช้งานโครงการที่เป็นผู้ให้บริการคลาวด์สามารถทราบถึง การใช้งานทรัพยากรของเครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือนได้ และสามารถนำข้อมูลที่ได้ไปใช้ในการจัดสรรการใช้งานทรัพยากรของเครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือนได้
- 2) ผู้ใช้งานโครงการที่เป็นผู้ใช้บริการคลาวด์สามารถทราบถึง การใช้งานทรัพยากรของเครื่องคอมพิวเตอร์เสมือนได้และนำข้อมูลที่ได้ไปใช้ในการบริหารจัดการ การใช้งานทรัพยากรของเครื่องคอมพิวเตอร์เสมือนที่ตนใช้งานได้

1.6 ส่วนประกอบของปฏิญญานิพนธ์

ปฏิญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาโดยทั่วไปออกเป็น 5 บทด้วยกัน

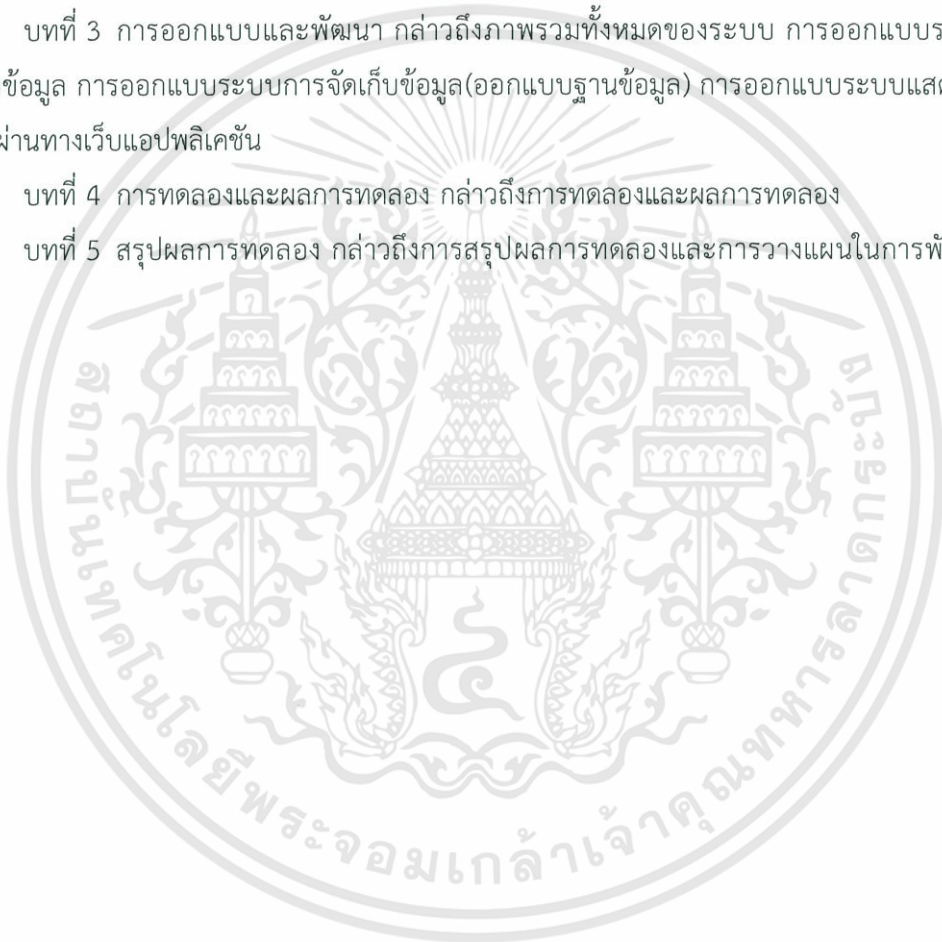
บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงทฤษฎีทั้งหมดที่นำมาศึกษาและพัฒนาในโครงการนี้

บทที่ 3 การออกแบบและพัฒนา กล่าวถึงภาพรวมทั้งหมดของระบบ การออกแบบระบบ การส่งข้อมูล การออกแบบระบบการจัดเก็บข้อมูล(ออกแบบฐานข้อมูล) การออกแบบระบบแสดงผลข้อมูลผ่านทางเว็บแอปพลิเคชัน

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการทดลองและผลการทดลอง

บทที่ 5 สรุปผลการทดลอง กล่าวถึงการสรุปผลการทดลองและการวางแผนในการพัฒนา



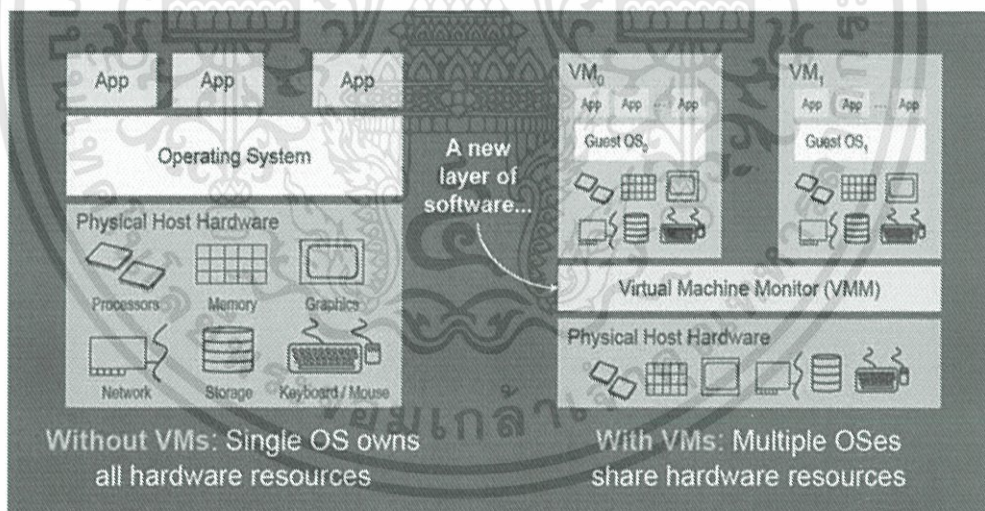
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 คลาวด์คอมพิวติ้ง (Cloud Computing)

2.1.1 เวอร์ชวลไลเซชัน (Virtualization)

เวอร์ชวลไลเซชันเป็นเทคโนโลยีที่สร้างระบบเสมือนเพื่อให้ระบบปฏิบัติการหลาย ๆ ระบบสามารถทำงานอยู่บนฮาร์ดแวร์ (hardware) ตัวเดียวกันได้ โดยอาศัยการทำงานของซอฟต์แวร์เวอร์ชวลไลเซชัน (virtualization software) เป็นตัวจัดการเรื่องของการใช้ทรัพยากรด้านฮาร์ดแวร์ต่าง ๆ เช่น ระบบปฏิบัติการ หน่วยความจำ เน็ตเวิร์ค เป็นต้น ให้กับคอมพิวเตอร์เสมือน (virtual machine) ทำให้สามารถแบ่งฮาร์ดแวร์ดังกล่าวให้เป็นคอมพิวเตอร์เสมือนหลาย ๆ เครื่องที่ทำงานอยู่บนฮาร์ดแวร์ตัวเดียวกันนั่นเอง [1]



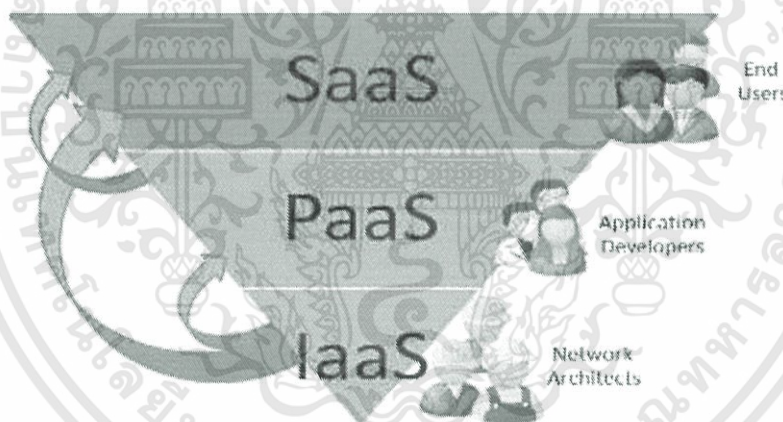
รูปที่ 2.1 ภาพแสดงการเปรียบเทียบระหว่างระบบปกติกับระบบที่ใช้งาน Virtualization [2]

มอนิเตอร์เครื่องคอมพิวเตอร์เสมือน (Virtual Machine Monitor) ซึ่งก็คือ ไฮเปอร์ไวเซอร์ (Hypervisor) จะทำหน้าที่จัดการการใช้งานพร้อม ๆ กันของคอมพิวเตอร์เสมือน ซึ่งเป็นโปรแกรมที่ทำงานอยู่ใน Layer ที่ถัดจากฟิสิคัล เลเยอร์ (Physical Layer) จากมุมมองของผู้ใช้งานเทคโนโลยีเวอร์ชวลไลเซชันนั้นจะเห็นว่าคอมพิวเตอร์เครื่องนั้นถูกใช้งานอยู่คนเดียวไม่ยุ่งเกี่ยวกับผู้ใช้งานคนอื่น แต่ในความเป็นจริงผู้ใช้งานทุกคนล้วนใช้งานอยู่บนเครื่องเดียวกันนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ลำดับชั้นของการให้บริการคลาวด์คอมพิวติ้ง (Cloud Computing Layers)

การบริการของคลาวด์คอมพิวติ้งนั้นได้แบ่งย่อยออกเป็น 3 ระดับ โดยมี ซอฟต์แวร์ แอส อะ เซอร์วิส (Software as a service) ซึ่งเป็นการที่ผู้ให้บริการคลาวด์นั้น เปิดให้บริการใช้งานซอฟต์แวร์ด้วยระบบคลาวด์คอมพิวติ้ง และ แพลตฟอร์ม แอส อะ เซอร์วิส (Platform as a service) ซึ่งเป็นการที่ผู้ให้บริการคลาวด์นั้น เปิดให้บริการเช่าเครื่องคอมพิวเตอร์เสมือนผ่านระบบคลาวด์คอมพิวติ้ง และ อินฟราสตรัคเจอร์ แอส อะ เซอร์วิส (Infrastructure as a service) ซึ่งเกี่ยวข้องกับผู้ให้บริการคลาวด์คอมพิวติ้งที่จะต้องจัดสรรทรัพยากรเครื่องเซิร์ฟเวอร์ และเครื่องคอมพิวเตอร์เสมือนที่เปิดให้บริการในระบบคลาวด์คอมพิวติ้ง โดยโครงการนี้จะทำการช่วยเหลือผู้ให้บริการในด้านการมอนิเตอร์การใช้งานทรัพยากรของเครื่องเซิร์ฟเวอร์ และเครื่องคอมพิวเตอร์เสมือน ที่ใช้ให้บริการในชั้นอินฟราสตรัคเจอร์ แอส อะ เซอร์วิส เพื่อตรวจสอบสถานะการใช้งานทรัพยากรของอุปกรณ์ในเครือข่าย เช่น การทำงานที่หนัก/เบา ของหน่วยความจำ, หน่วยประมวลผล เป็นต้น [3]



รูปที่ 2.2 ภาพแสดงลำดับชั้นของการให้บริการผู้ใช้งานในคลาวด์คอมพิวติ้ง [3]

2.1.2.1 ซอฟต์แวร์ แอส อะ เซอร์วิส

ผู้ให้บริการคลาวด์ จะใช้งานแอปพลิเคชัน (Application) ต่าง ๆ ของผู้ให้บริการคลาวด์ ซึ่งผู้ให้บริการคลาวด์จะเป็นผู้ติดตั้งและจัดการโปรแกรมเพื่อให้ผู้ใช้บริการเข้าถึงโปรแกรมผ่านคลาวด์ไคลเอนต์ (cloud client) โดยที่ผู้ใช้บริการไม่ต้องดูแล จัดการ หรือติดตั้งแพลตฟอร์มที่แอปพลิเคชันนั้น ๆ ต้องการใช้งานเลย เช่น Web Application เป็นต้น [4]

2.1.2.2 แพลตฟอร์ม แอส อะ เซอร์วิส

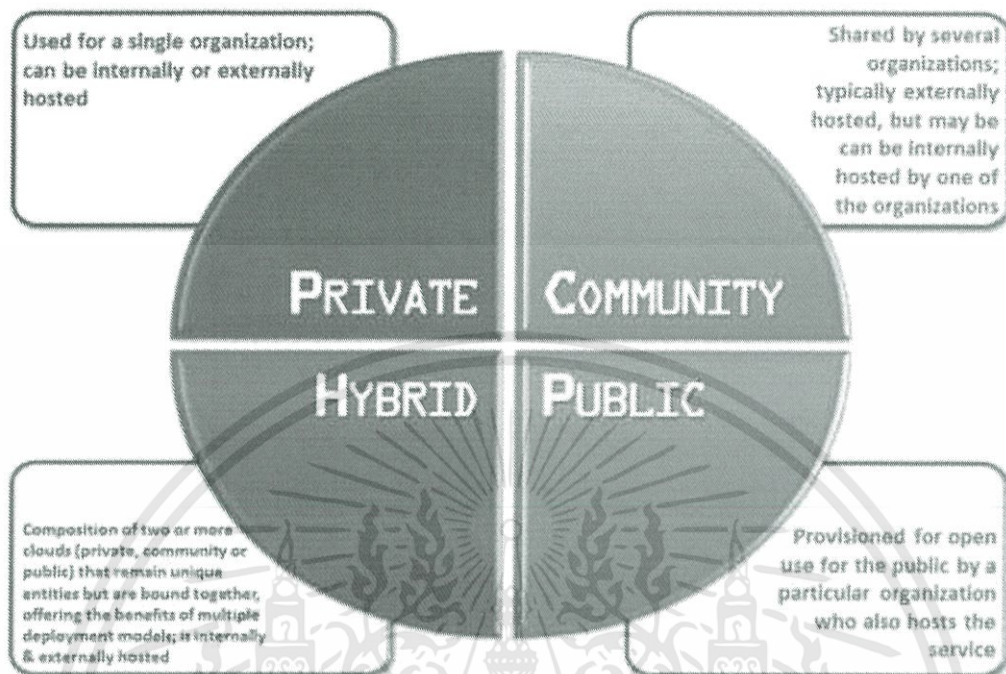
ผู้ให้บริการคลาวด์จะจัดหาแพลตฟอร์มทางคอมพิวเตอร์ที่รองรับระบบปฏิบัติการ และสิ่งจำเป็นต่าง ๆ ตามที่ผู้ใช้บริการคลาวด์ต้องการในการปฏิบัติงานต่าง ๆ เช่น การเขียนโปรแกรม, ฐานข้อมูล และเว็บเซิร์ฟเวอร์ (web server) เป็นต้น ซึ่งผู้ใช้บริการคลาวด์จะเช่าเครื่องคอมพิวเตอร์เสมือนเพื่อการพัฒนาหรือใช้งานคอมพิวเตอร์เสมือนนั้นบนคลาวด์แพลตฟอร์ม (cloud platform) โดยไม่ต้องคำนึงถึงการจัดการฮาร์ดแวร์ รวมถึงมีบริการเสริมอื่น ๆ (เช่น การรีโมทไปยังคอมพิวเตอร์ (Remote connection), การยืนยันตัวตน (Authentication) เป็นต้น) ให้ผู้ใช้บริการคลาวด์ใช้งานโปรแกรมประยุกต์เหล่านั้นได้ [4]

2.1.2.3 อินฟราสตรักเจอร์ แอส อะ เซอร์วิส

เป็นขั้นพื้นฐานของรูปแบบการให้บริการคลาวด์ โดยผู้ให้บริการคลาวด์จะจัดหาทรัพยากร เช่น คอมพิวเตอร์, ระบบปฏิบัติการ, ที่เก็บข้อมูล, อุปกรณ์เครือข่าย และอื่น ๆ เป็นเครื่องจริง ๆ หรือเครื่องคอมพิวเตอร์เสมือนที่ทำงานอยู่บนเครื่องเซิร์ฟเวอร์ ซึ่งถูกจัดการโดยไฮเปอร์ไวเซอร์ ตัวอย่างของไฮเปอร์ไวเซอร์ที่เป็นที่รู้จักอย่างแพร่หลาย คือ อีเอสเอ็กซ์ (ESX), เซ็น (Xen) และเควีเอ็ม (KVM) การบริหารจัดการทรัพยากรส่วนกลางของไฮเปอร์ไวเซอร์ในระบบที่สนับสนุนการปฏิบัติงานแบบคลาวด์ ทำให้สามารถรองรับจำนวนของเครื่องคอมพิวเตอร์เสมือนได้เพิ่มขึ้น [4]

2.1.3 ประเภทของคลาวด์คอมพิวติ้ง (Cloud Computing Types)

คลาวด์คอมพิวติ้ง เกิดจากการใช้ประโยชน์ของทรัพยากรทางคอมพิวเตอร์โดยใช้เทคโนโลยีเวอร์ชวลไลเซชัน และรูปแบบการใช้งานต่าง ๆ ขึ้นอยู่กับความหลากหลายของตำแหน่งที่ตั้งเครื่องเซิร์ฟเวอร์จริง ๆ และความต้องการขององค์กร (เช่น ความปลอดภัย เป็นต้น) โดยสามารถแบ่งตามความต้องการ ได้เป็น 4 ประเภท ได้แก่ คลาวด์แบบสาธารณะ (public cloud), คลาวด์แบบเฉพาะองค์กร (private cloud), คลาวด์แบบผสม (hybrid cloud) และคลาวด์แบบใช้ระหว่างองค์กร (community cloud) [3]



รูปที่ 2.3 ภาพแสดงลำดับชั้นของการให้บริการผู้ใช้งานในคลาวด์คอมพิวติ้ง [3]

2.1.3.1 คลาวด์แบบสาธารณะ

คลาวด์แบบสาธารณะจะอนุญาตให้ผู้ให้บริการเข้าถึงบริการต่าง ๆ ของผู้ให้บริการเครื่องเซิร์ฟเวอร์ผ่านอินเทอร์เน็ตโดยใช้เว็บแอปพลิเคชันเป็นตัวกลางในการขอทรัพยากรที่ต้องการบริการต่าง ๆ ซึ่งจะคิดค่าบริการตามที่ผู้ใช้งานต้องการ ข้อมูลที่ใช้กับคลาวด์แบบนี้อาจจะไม่ปลอดภัยเพราะต้องไปฝากกับผู้ให้บริการ ตัวอย่างผู้ให้บริการคลาวด์แบบสาธารณะ เช่น ไมโครซอฟต์ (Microsoft) และกูเกิล (Google) ซึ่งเจ้าของจะเป็นผู้ดูแลโครงสร้างพื้นฐาน และเปิดให้ใช้งานแบบสาธารณะนั่นเอง

2.1.3.2 คลาวด์แบบเฉพาะองค์กร

คลาวด์แบบเฉพาะองค์กรจะติดตั้งคลาวด์ไว้ภายในองค์กร สำหรับใช้งานภายในองค์กรเท่านั้น (ตั้งไว้ในไฟล์-วอลล์ ขององค์กร) และถูกบริหารจัดการโดยภายในองค์กรนั้น ๆ หรืออาจเป็นองค์กรอื่นที่เกี่ยวข้อง คลาวด์ในรูปแบบนี้จะคำนึงถึงด้านความปลอดภัย เป็นหลัก ที่แตกต่างจากคลาวด์แบบสาธารณะ โดยที่องค์กรนั้น ๆ เป็นคนบริหารจัดการ เปลี่ยนแปลง กำหนดการตั้งค่าทุกอย่าง และบำรุงรักษาคลาวด์นั้นด้วยตนเอง รวมถึงข้อมูลมีความปลอดภัยมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.3 คลาวด์แบบผสม

คลาวด์แบบผสม คือการใช้งานแนวคิดของคลาวด์แบบสาธารณะและคลาวด์แบบเฉพาะองค์กร ซึ่งส่งผลให้การจัดการและความรับผิดชอบต่าง ๆ จะแบ่งกันระหว่างผู้ให้บริการคลาวด์แบบสาธารณะ และ องค์กรนั้น ๆ การใช้คลาวด์แบบผสมจะทำให้สามารถจัดการใช้งานบริการต่าง ๆ ได้ตรงตามจุดประสงค์และความต้องการ ในแนวทางที่เหมาะสม

2.1.3.4 คลาวด์แบบใช้ระหว่างองค์กร

คลาวด์แบบใช้ระหว่างองค์กรนั้นจะแชร์ (share) โครงสร้างพื้นฐานระหว่างหลาย ๆ องค์กร จากกลุ่มองค์กรที่ระบุ การแชร์นั้นจะอยู่บนพื้นฐานธุรกิจที่เกี่ยวข้อง (เช่น เป้าหมาย, นโยบาย, ข้อตกลง และความปลอดภัยที่ต้องการ) ซึ่งส่งผลให้การจัดการดูแลจะเป็นภายในหรือองค์กรอื่นที่รับผิดชอบวางระบบคลาวด์ให้ ค่าใช้จ่ายจะกระจายบนผู้ใช้งานที่น้อยกว่าคลาวด์แบบสาธารณะ ทำให้ประหยัดค่าใช้จ่ายได้บางส่วน

2.1.4 คลาวด์เซิร์ฟเวอร์ (Cloud server)

คลาวด์เซิร์ฟเวอร์ คือ เซิร์ฟเวอร์ที่ใช้เทคโนโลยีเวอร์ชวลไลเซชันในการเปิดให้บริการเซิร์ฟเวอร์ต่าง ๆ อาทิเช่น เว็บเซิร์ฟเวอร์ เอฟทีพีเซิร์ฟเวอร์ (ftp server) เมลเซิร์ฟเวอร์ (mail server) เป็นต้น โดยจะสร้างระบบเสมือนให้ระบบหลาย ๆ ระบบนั้นทำงานอยู่บนฮาร์ดแวร์ชุดเดียวกัน เพื่อก่อให้เกิดการใช้ทรัพยากรให้เกิดประโยชน์ และ ประสิทธิภาพที่สูงที่สุด ซึ่งในการทำงานของเครื่องเซิร์ฟเวอร์นั้นจะสร้างเครื่องคอมพิวเตอร์เสมือนขึ้นมาหลาย ๆ ตัว โดยที่แต่ละตัวนั้นจะแบ่งการทำงานของหน่วยประมวลผล, หน่วยความจำ, สตอเรจ ออกจากกันโดยเด็ดขาดทำให้เครื่องคอมพิวเตอร์เสมือนแต่ละตัวนั้น ไม่ทำงานรบกวนกันเอง

2.2 ดาต้าเบส แบบ โนเอสคิวแอล (NoSQL Database)

ดาต้าเบสแบบโนเอสคิวแอลนั้นเกิดขึ้นในปี พ.ศ.2541 โดย Carlo Strozzi ได้ให้คำนิยามไว้ว่าเป็นระบบดาต้าเบส ที่ทำงานโดยมีความสัมพันธ์กันแต่ไม่ได้ใช้ภาษาเอสคิวแอลในการทำงาน ซึ่งเป้าหมายหลักของโนเอสคิวแอล นั้นคือการลดความยุ่งยาก ซับซ้อนในระบบดาต้าเบส เพื่อให้สามารถทำงานได้รวดเร็วขึ้น เหมาะกับการเก็บข้อมูลแบบเรียลไทม์ (real time) ซึ่งโนเอสคิวแอลนั้นจะรูปแบบในการเก็บข้อมูลที่ต่างจากแบบรีเลชันนอล (relational) โดยในโครงการนี้ได้เลือกใช้งาน MongoDB ซึ่งเป็น NoSQL แบบ Document-Oriented ซึ่งสามารถรองรับการทำงานหนัก และรับ

ข้อมูลแบบเรียลไทม์ปริมาณมาก ๆ และสามารถขยายขนาดได้เร็วกว่าดาต้าเบสแบบรีเรชันแนล ในการจัดการ [8]

2.3 ภาษาไพทอน (Python programming language)

ภาษาไพทอนถูกพัฒนาโดย กีโด ฟาน รอสซัม (Guido van Rossum) ในปีพ.ศ. 2533 ซึ่งไพทอนเป็นภาษาสคริปต์ ทำให้ใช้เวลาในการเขียนและคอมไพล์ไม่มาก ทำให้เหมาะกับงานด้านการดูแลระบบ (System administration) ไวยากรณ์ของไพทอนได้กำจัดการใช้สัญลักษณ์ที่ใช้ในการแบ่งบล็อกของโปรแกรม และใช้การย่อหน้าแทน และสามารถเรียกใช้ภาษาโปรแกรมอื่นๆได้หลายภาษา ทำให้เหมาะที่จะใช้เขียนเพื่อประสานงานโปรแกรมที่เขียนโดยภาษาที่ต่างกันได้อีกทั้งยังมี library รองรับการทำงานหลายรูปแบบอีกด้วย [5]

โมดูล (Module) ที่นำมาใช้งานในโครงงานนี้ในส่วนของเควีเอ็มคือ โมดูล psutil ซึ่งเป็นโมดูลที่ใช้ภาษาไพทอนในการเรียกข้อมูลอินเทอร์เน็ตต่างๆ เกี่ยวกับโปรเซสที่กำลังทำงานอยู่ รวมถึงข้อมูลเกี่ยวกับการทำงานของระบบ โดยผ่านไฟล์ซิสเต็ม (Files System) อาทิเช่น ซีพียู, หน่วยความจำ, ดิสก์, เน็ตเวิร์ค ฯลฯ

ส่วนอีกโมดูลที่นำมาใช้งานในโครงงานนี้คือ โมดูล pymongo ซึ่งเป็นโมดูลที่ทำให้โปรแกรมที่ใช้ภาษาไพทอนนั้น สามารถเชื่อมต่อกับดาต้าเบสมองโกล หรือก็คือ มองโกล เอพีไอ (Mongo API) นั่นเอง ซึ่งโมดูลนี้จะใช้ในการเก็บข้อมูลจาก KVM มาเก็บไว้ในดาต้าเบส

2.4 ภาษาวินโดวส์ พาวเวอร์เชลล์ (Windows Powershell)

เป็นภาษาที่ทางบริษัทไมโครซอฟท์ (Microsoft) พัฒนาขึ้นมาเพื่อการใช้งานติดต่อบริบทปฏิบัติการวินโดวส์ในรูปแบบคอมมานด์ไลน์ (command line) โดยภาษาวินโดวส์ พาวเวอร์เชลล์จะมีคอมมานด์เล็ต (command-let) เป็นตัวรวบรวมคำสั่งในออบเจ็คต่าง ๆ มาทำงานด้วยการเรียกฟังก์ชันคอมมานด์เล็ตเพียงคำสั่งเดียว เช่น Get-Date, Connect-VICenter เป็นต้น โดยคอมมานด์เล็ตแต่ละคำสั่งนั้น จะทำหน้าที่คล้ายกับคลาส (Class) ในภาษาซีคือการเรียกใช้งานนั้น ผู้ใช้งานภาษาวินโดวส์ พาวเวอร์เชลล์นั้นเพียงแค่เรียกคำสั่งของคอมมานด์เล็ต ก็จะสามารถใช้งานได้ทันที ซึ่งตัวภาษาวินโดวส์ พาวเวอร์เชลล์นั้นจะทำงานอยู่บนพื้นฐานของ .NET Platform [10]

2.5 ภาษาแอชทีเอ็มแอล (HTML)

HTML เป็นชื่อย่อมาจาก hypertext Markup Language ซึ่งพัฒนามาจากภาษา SGML ซึ่งภาษา HTML นี้เป็นภาษาที่ใช้ในการพัฒนาเว็บเพจที่ใช้กันอย่างแพร่หลาย โดยการเรียกใช้ผ่านบราวเซอร์ต่างๆ เช่น Google chrome, Internet Explorer เป็นต้น ซึ่งข้อดีของ HTML คือสามารถเรียกดูได้จากทุกระบบปฏิบัติการ ในส่วนของการพัฒนาหรือการสร้าง นั้น สามารถใช้โปรแกรม Text Editor หรือ โปรแกรมช่วยอื่นๆก็ได้

2.6 ภาษาจาวาสคริปต์ (JavaScript)

Java script เป็นภาษาที่ใช้ในการเขียนโปรแกรมการคำนวณ การประมวลผล การรับส่งข้อมูล รวมทั้งการแสดงผล บนเว็บเพจ ซึ่ง Java script นั้นสามารถเขียนร่วมกับภาษาอื่นที่ใช้สร้างเว็บเพจได้ เช่น HTML PHP เป็นต้น

2.7 ผลิตภัณฑ์ที่เกี่ยวข้อง (Related Works)

ซอฟต์แวร์ที่มีความคล้ายคลึงกับโครงงานนี้ก็คือซอฟต์แวร์ที่สามารถให้ข้อมูลเกี่ยวกับการใช้งานทรัพยากรในเครื่องใด ๆ ได้ รวมถึงข้อมูลการใช้งานทรัพยากรของเครื่องคอมพิวเตอร์เสมือนได้ ตัวอย่างโปรแกรมที่นำมาเปรียบเทียบก็คือ โปรแกรมไฮเปอร์ริค เอชคิว (Hyperic HQ) ในส่วนของลินุกซ์มอนิเตอร์ริง (Linux Monitoring) และ ESX มอนิเตอร์ริง เท่านั้น เนื่องจากโครงงานนี้สนับสนุนการใช้งานไฮเปอร์ไวเซอร์ KVM และ ไฮเปอร์ไวเซอร์ ESX เท่านั้น

จากการเปรียบเทียบระหว่าง โครงงานระบบดูแลโครงสร้างคลาวด์กับโปรแกรมไฮเปอร์ริค เอชคิว พบว่า โครงงานระบบดูแลโครงสร้างคลาวด์มีการใช้งานทรัพยากรหน่วยความจำของเซิร์ฟเวอร์น้อยกว่าโปรแกรมไฮเปอร์ริค เอชคิว

บทที่ 3

การออกแบบและพัฒนา

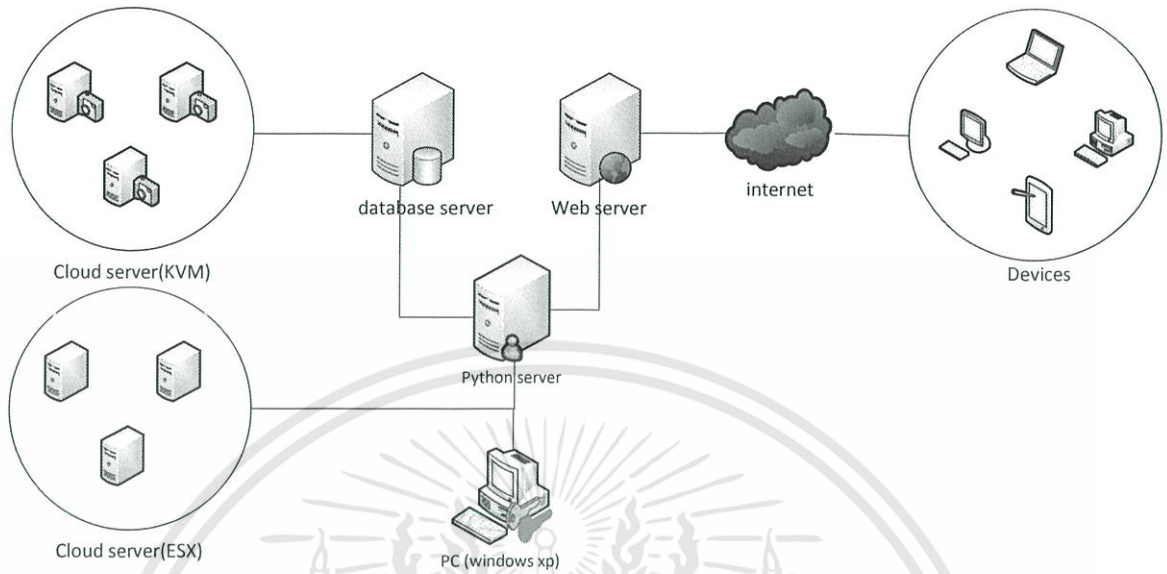
การวิเคราะห์และออกแบบระบบดูแลโครงสร้างคลาวด์นั้นจะนำทฤษฎีที่ศึกษาค้นคว้าจากบทที่ 2 มาใช้ในการออกแบบและพัฒนาระบบ โดยในบทนี้จะกล่าวถึงภาพรวมทั้งหมดของระบบและอธิบายรายละเอียดของส่วนประกอบในระบบ พร้อมทั้งอธิบายถึงเทคโนโลยีที่และภาษาที่ใช้ในการพัฒนาระบบ

โครงการนี้ได้เลือกใช้โปรแกรม VMware vSphere PowerCLI ในการดึงข้อมูลเกี่ยวกับการทำงานและการใช้งานทรัพยากรต่างๆของทั้งเครื่องคอมพิวเตอร์เสมือนและเครื่องเซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ VMware ESX และเลือกใช้โมดูล psutil ของภาษาไพธอนในการดึงข้อมูลเกี่ยวกับการทำงานและการใช้งานทรัพยากรต่างๆของทั้งเครื่องคอมพิวเตอร์เสมือนและเครื่องเซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ KVM ในการเขียนโปรแกรมเพื่อให้สามารถพัฒนาโปรแกรมให้มีขีดความสามารถสูง และสามารถใช้งานได้หลากหลาย และเลือกใช้ไพธอนเซิร์ฟเวอร์เพื่อเป็นตัวเชื่อมระหว่างการดึงข้อมูลโดยใช้โปรแกรม VMware vSphere PowerCLI มาเก็บในดาต้าเบสและการดึงข้อมูลของเว็บแอปพลิเคชันจากดาต้าเบส โดยจะเลือกใช้มอดูลโฟลคตีบี เป็นดาต้าเบส และในส่วนของเว็บแอปพลิเคชันมีการใช้เอแจ็กซ์(Ajax) ในการดึงข้อมูลแบบเรียลไทม์ ภาษาจาวาสคริปต์ ในการรับข้อมูลที่ได้จากดาต้าเบสเพื่อประมวลผลข้อมูลการใช้งานทรัพยากรของทั้งเครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือน และเลือกใช้กูเกิลชาร์ต (Google Chart) ในการแสดงข้อมูลที่ดึงมาในรูปแบบของกราฟ

3.1 ภาพรวมและองค์ประกอบของระบบ

ระบบดูแลโครงสร้างคลาวด์จะสามารถแสดงข้อมูลของการใช้ทรัพยากรและพฤติกรรมการทำงานของเครื่องเซิร์ฟเวอร์แต่ละตัว โดยข้อมูลที่ระบบเก็บไว้ในดาต้าเบสจะมีการเก็บข้อมูลแบบเรียลไทม์และสามารถตรวจสอบข้อมูลย้อนหลังได้ เพื่อให้ผู้ดูแลระบบสามารถนำข้อมูลที่ได้มาใช้ในการบริหารจัดการคลาวด์

ระบบดูแลโครงสร้างคลาวด์จะทำงานบนเซิร์ฟเวอร์ 3 ตัว โดยจะแบ่งเป็นเซิร์ฟเวอร์ที่ให้บริการเว็บแอปพลิเคชันและเซิร์ฟเวอร์ที่ทำหน้าที่จัดเก็บข้อมูลของระบบ เมื่อผู้ใช้บริการมีการติดต่อใช้งานผ่านหน้าเว็บ เซิร์ฟเวอร์ที่ให้บริการเว็บแอปพลิเคชันจะทำการดึงข้อมูลจากเซิร์ฟเวอร์ที่จัดเก็บข้อมูลตามรูปที่ 3.1



รูปที่ 3.1 โครงสร้างของระบบดูแลโครงสร้างคลาวด์



Database server เป็นเซิร์ฟเวอร์ที่ใช้ในการจัดเก็บของมูลทรัพยากรต่างๆตั้งมาจากเครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือน



Web server เป็นเซิร์ฟเวอร์ที่ใช้แสดงข้อมูลและความสามารถในการให้บริการของ



Monitoring agent เป็นซอฟต์แวร์ที่ใช้ในการดึงข้อมูลเครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือน

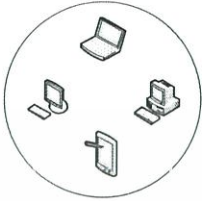


Virtual machine หรือเครื่องคอมพิวเตอร์เสมือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

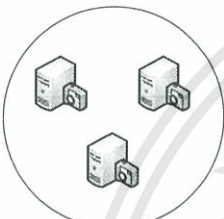


Internet เป็นตัวกลางในการเชื่อมต่อระหว่างผู้ใช้งานและระบบดูแลโครงสร้างคลาวด์



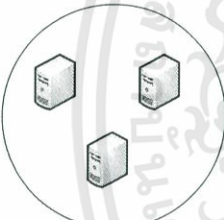
Device เป็นอุปกรณ์ที่ใช้ในการเข้าดูการใช้ทรัพยากรของระบบดูแลโครงสร้าง

คลาวด์



Cloud server(KVM)

คลาวด์เซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ KVM



Cloud server(ESX)

คลาวด์เซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ ESX



PC (windows xp)

เครื่องคอมพิวเตอร์ ที่ใช้ระบบปฏิบัติการ วินโดวส์ เอ็กซ์พี



สคริปต์ที่ใช้รันเพื่อดึงข้อมูลจากไฮเปอร์ไวเซอร์ VMware ESX

ระบบดูแลโครงสร้างคลาวด์จะแบ่งออกเป็น 3 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การดึงข้อมูลจากเครื่องเซิร์ฟเวอร์แต่ละตัว

ในการดึงข้อมูลทรัพยากรและความสามารถในการให้บริการของเครื่องเซิร์ฟเวอร์แต่ละตัวนั้น จะแบ่งเป็นสองแบบคือ แบบที่หนึ่งคือการดึงข้อมูลจากเครื่องเซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ VMware ESX ซึ่งการดึงข้อมูลใช้โปรแกรม VMware vSphere PowerCLI ซึ่งจะรันบน วินโดวส์พาวเวอร์เชลล์ (Windows PowerShell) โดยที่สคริปต์ภาษาวินโดวส์พาวเวอร์เชลล์นั้นจะทำงานอยู่บนระบบปฏิบัติการวินโดวส์อีกเครื่องหนึ่ง แบบที่สองจะเป็นการจะเลือกดึงข้อมูลจากเครื่องเซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ KVM ซึ่งจะใช้ไพธอนสคริปต์ที่เรียกใช้โมดูล psutil ในการดึงข้อมูล ไพธอนสคริปต์นี้จะรันอยู่บนเครื่องเซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ KVM โดยจะดึงเฉพาะข้อมูลที่สำคัญและมีผลต่อการบริหารจัดการระบบคลาวด์และข้อมูลที่ทำกรดึงมานั้นจะเป็นข้อมูลของทั้งเครื่องคอมพิวเตอร์เสมือนและข้อมูลของเครื่องเซิร์ฟเวอร์ โดยจะแบ่งข้อมูลเป็นดังนี้

3.1.1.1 ข้อมูลที่มีการเปลี่ยนแปลงน้อย ประกอบด้วย

- จำนวนของคอร์ของหน่วยประมวลผล
- ความจุทั้งหมดของหน่วยความจำ
- ขนาดของดิสก์

3.1.1.2 ข้อมูลที่เป็นเรียลไทม์ ประกอบด้วย

- ข้อมูลของหน่วยความจำ
 - ปริมาณการใช้งานหน่วยความจำในขณะนั้น
 - ปริมาณหน่วยความจำที่ใช้งานอยู่
 - ปริมาณหน่วยความจำที่ไม่ได้ใช้งาน
- ข้อมูลของเน็ตเวิร์ค
 - ข้อมูลของเน็ตเวิร์คจะแสดงข้อมูลการส่งของข้อมูล
 - ข้อมูลของเน็ตเวิร์คจะแสดงข้อมูลการรับของข้อมูล
 - ปริมาณข้อมูลที่เกิดการขัดข้องในการส่ง
- ข้อมูลปริมาณการใช้งานของซีพียู
- ระยะเวลาของการใช้งานเครื่องเซิร์ฟเวอร์
- เวลาในขณะที่ทำการดึงข้อมูลจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงการใช้คำสั่งในการดึงข้อมูลทรัพยากรจากไฮเปอร์ไวเซอร์ KVM และ VMware ESX

ข้อมูลทรัพยากร	Hypervisor		
	KVM	VMware ESX	
		เครื่องเซิร์ฟเวอร์	เครื่องคอมพิวเตอร์เสมือน
ข้อมูลของหน่วยความจำ - ปริมาณการใช้งานหน่วยความจำในขณะนั้น (%) - ปริมาณของหน่วยความจำทั้งหมด	psutil.phymem_usage()	Get-VMHost Where {\$_.PowerState -eq "PoweredOn"} Select Name, MemoryGB	Get-VM Where {\$_.PowerState -eq "PoweredOn"} Select Name, MemoryGB
ข้อมูลของซีพียู - ปริมาณการใช้งานซีพียูในขณะนั้น (%) - จำนวนของคอร์ของซีพียู	psutil.cpu_percent()	\$Entities = Get-VMHost Where {\$_.PowerState -eq "PoweredOn"} \$metrics = "cpu.usage.average" \$report = Get-Stat -Entity \$Entities -Stat \$metrics -Realtime	\$Entities = Get-VM Where {\$_.PowerState -eq "PoweredOn"} \$metrics = "cpu.usage.average" \$report = Get-Stat -Entity \$Entities -Stat \$metrics -Realtime
ข้อมูลของดิสก์ - ปริมาณการใช้งานดิสก์ในขณะนั้น (%) - ปริมาณความจุดิสก์ทั้งหมด - ปริมาณความจุคงเหลือ	psutil.disk_io_counters()	\$Entities = Get-VMHost Where {\$_.PowerState -eq "PoweredOn"} \$metrics = "disk.usage.average" \$report = Get-Stat -Entity \$Entities -Stat \$metrics -Realtime	\$Entities = Get-VM Where {\$_.PowerState -eq "PoweredOn"} \$metrics = "disk.usage.average" \$report = Get-Stat -Entity \$Entities -Stat \$metrics -Realtime
ข้อมูลของเน็ตเวิร์ค - ปริมาณการส่งข้อมูล (kbps) - ปริมาณการรับข้อมูล (kbps)	psutil.net_io_counters()		
ข้อมูลระยะเวลาในการเปิดใช้งานเครื่อง	startTime = time.time() def getUptime(): return time.time() - startTime	(Get-Date) - (New-Timespan - Seconds \$_.Summary.QuickStats.UptimeSeconds)	(Get-Date) - (New-Timespan - Seconds \$_.Summary.QuickStats.UptimeSeconds)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเก็บข้อมูล

ในการจัดเก็บข้อมูลจากไฮเปอร์ไวเซอร์ KVM จะมี agent อยู่ที่ตัว เซิร์ฟเวอร์และคอมพิวเตอร์เสมือน ซึ่ง agent จะทำหน้าที่ในการส่งข้อมูลไปยังมอดูลโคตีบีโดยตรง และในการจัดเก็บข้อมูลจากไฮเปอร์ไวเซอร์ VMware ESX นั้นเนื่องจากการส่งข้อมูลจากพาวเวอร์ซีแอลไอและจัดเก็บข้อมูลใน ฐานข้อมูลมอดูลโคตีบีที่มีระบบปฏิบัติการแตกต่างกัน จึงจำเป็นต้องมีไพธอนเซิร์ฟเวอร์ที่ใช้เฟล็ก (Flask) ซึ่งเป็นไมโครเฟรมเวิร์คของภาษาไพธอน (Microframework for Python) เพื่อช่วยในการติดต่อกับฐานข้อมูลผ่าน RESTful Web Service ที่ไพธอนเซิร์ฟเวอร์นั้นจะเป็นคนจัดการเมธอด (Methods) ต่าง ๆ เช่น GET, POST เป็นต้น [6]

ในการเก็บข้อมูลจะมีคอลเลคชันที่ใช้ในการเก็บ 3 แบบ

1. คอลเลคชันที่ทำการเก็บชื่อของเครื่องเซิร์ฟเวอร์ทุกตัว ตามตารางที่ 3.2
2. คอลเลคชันที่ทำการเก็บข้อมูลการใช้ทรัพยากรของเครื่องคอมพิวเตอร์เสมือนทุกตัว ตามตารางที่ 3.3
3. คอลเลคชันที่ทำการเก็บข้อมูลการใช้ทรัพยากรของเครื่องเซิร์ฟเวอร์ทุกตัวตามตารางที่ 3.4

ตารางที่ 3.2 แสดงการเก็บข้อมูลในคอลเลคชันแบบที่หนึ่ง

Field	Description	Data Type
Name server	ชื่อเครื่องเซิร์ฟเวอร์ทั้งหมดที่มีอยู่ในระบบ	String

คอลเลคชันแบบที่หนึ่งจะทำการเก็บชื่อของเครื่องเซิร์ฟเวอร์ทุกตัวที่มีอยู่ในระบบซึ่งกำลังเปิดให้บริการอยู่

ตารางที่ 3.3 แสดงการเก็บข้อมูลในคอลเลคชันแบบที่สอง

Field	Description	Data Type
Name server	ชื่อเครื่องเซิร์ฟเวอร์	String
Name VM	ชื่อเครื่องคอมพิวเตอร์เสมือนทั้งหมดที่อยู่ในระบบ	

คอลเลคชันแบบที่สองจะทำการเก็บชื่อของเครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือนภายในเครื่องเซิร์ฟเวอร์นั้น ๆ โดยจะเก็บเป็นหนึ่งเครื่องเซิร์ฟเวอร์ต่อหนึ่งคอลเลคชัน

ตารางที่ 3.4 แสดงการเก็บข้อมูลในคอลเลกชันแบบที่สาม (เครื่องคอมพิวเตอร์เสมือน)

Field	Description	Data Type
Number CPU	จำนวนโปรเซสเซอร์ที่มี ของ CPU นั้น ๆ	Integer
%CPU	เปอร์เซ็นต์การใช้งานหน่วยประมวลผล	Integer
Boot time	ระยะเวลาที่เครื่องได้เปิดใช้งาน	Integer
Network	จำนวนไบต์และแพ็คเกจที่รับ-ส่ง	Integer
Memory	ข้อมูลหน่วยความจำ และการใช้งานทรัพยากรหน่วยความจำ	Integer
Disk	ข้อมูลดิสก์ และการใช้งานทรัพยากรดิสก์	Integer
Time	เวลา ณ ขณะที่ดึงข้อมูล	Date

คอลเลกชันแบบที่สาม (เครื่องคอมพิวเตอร์เสมือน) จะทำการเก็บข้อมูลเก็บข้อมูลการใช้ทรัพยากรของเครื่องคอมพิวเตอร์เสมือนโดยจะเก็บเป็นหนึ่งเครื่องคอมพิวเตอร์เสมือนต่อหนึ่งคอลเลกชัน

ตารางที่ 3.5 แสดงการเก็บข้อมูลในคอลเลกชันแบบที่สาม (เครื่องเซิร์ฟเวอร์)

Field	Description	Data Type
Number CPU	จำนวนโปรเซสเซอร์ที่มี ของ CPU นั้น ๆ	Integer
%CPU	เปอร์เซ็นต์การใช้งานหน่วยประมวลผล	Integer
Boot time	ระยะเวลาที่เครื่องได้เปิดใช้งาน	Integer
Network	จำนวนไบต์และแพ็คเกจที่รับ-ส่ง	Integer
Memory	ข้อมูลหน่วยความจำ และการใช้งานทรัพยากรหน่วยความจำ	Integer
Disk	ข้อมูลดิสก์ และการใช้งานทรัพยากรดิสก์	Integer
Time	เวลา ณ ขณะที่ดึงข้อมูล	Date

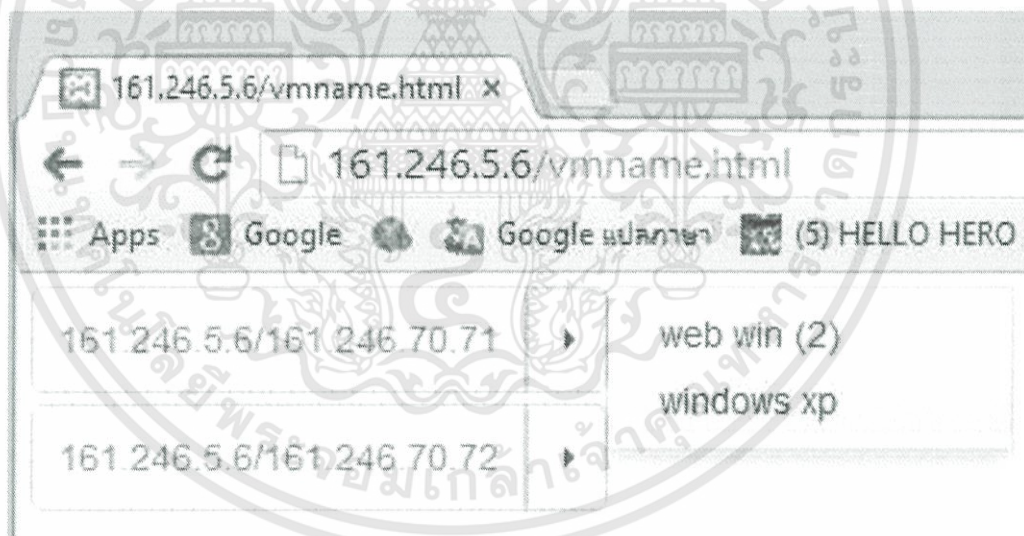
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอลเลกชันแบบที่สาม (เครื่องเซิร์ฟเวอร์) จะทำการเก็บข้อมูลการใช้ทรัพยากรของเครื่องเซิร์ฟเวอร์โดยจะเก็บเป็นหนึ่งเครื่องเซิร์ฟเวอร์ต่อหนึ่งคอลเลกชัน

3.3 การแสดงข้อมูล

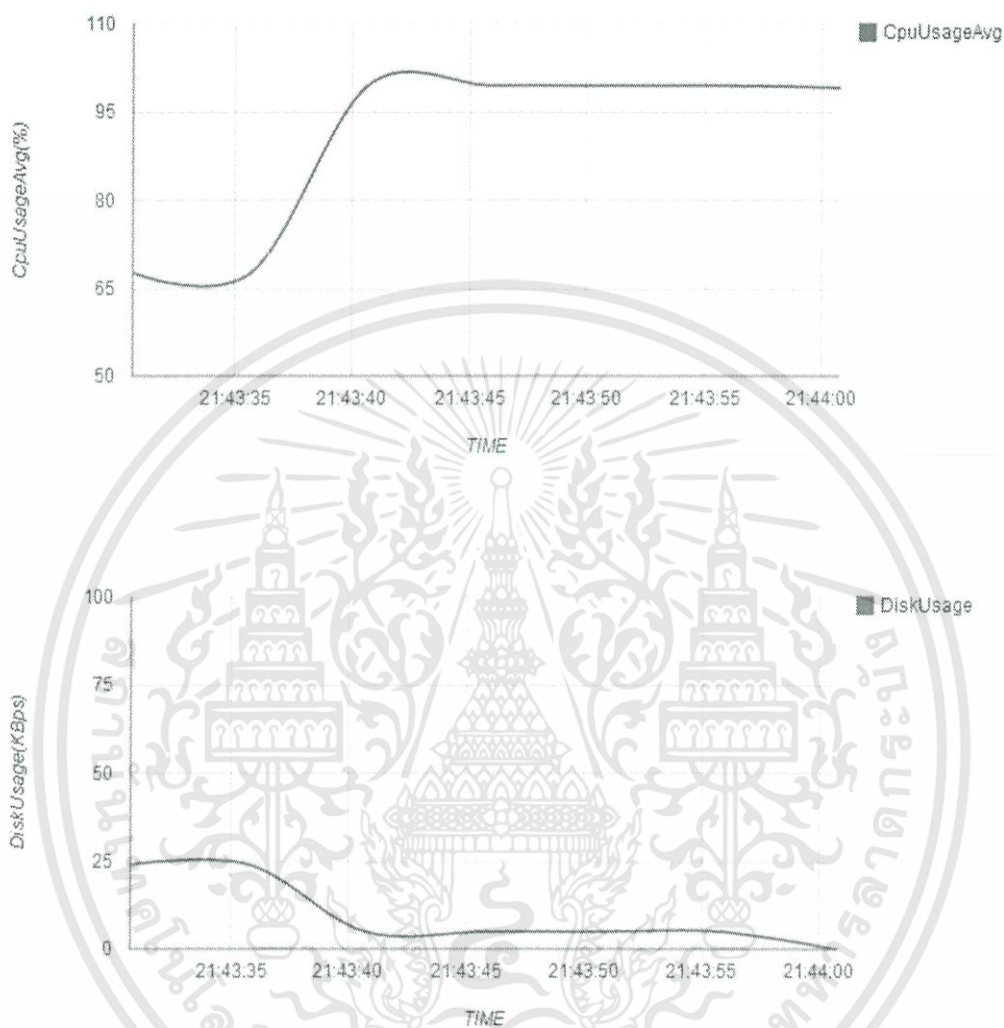
ในการแสดงข้อมูลนั้นจะแสดงผ่านทางเว็บแอปพลิเคชัน ซึ่งจะมีการออกแบบให้ง่ายต่อการเข้าใจและใช้งานเพื่อผู้ใช้บริการที่ไม่มีประสบการณ์ในการดูแลระบบ โดยโครงงานนี้จะเลือกใช้ Ajax ในการรับข้อมูลจากดาต้าเบสเซิร์ฟเวอร์เพื่อให้ได้ข้อมูลแบบเรียลไทม์ โดย Ajax จะทำการส่งคำขอไปยัง Python server ว่าต้องการข้อมูลอะไร และ Python server จะส่งข้อมูลนั้นๆกลับมา โครงงานนี้เลือกใช้ Java script ในส่วนของการประมวลผลของ Web application เพื่อให้ง่ายต่อการพัฒนา และเลือกใช้ Google chart API ในการแสดงข้อมูลในรูปแบบของกราฟ

โดย ในหน้าแรกจะเป็นหน้าที่แสดง ชื่อ เครื่องเซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือนทั้งหมด ตามรูปที่ 3.2



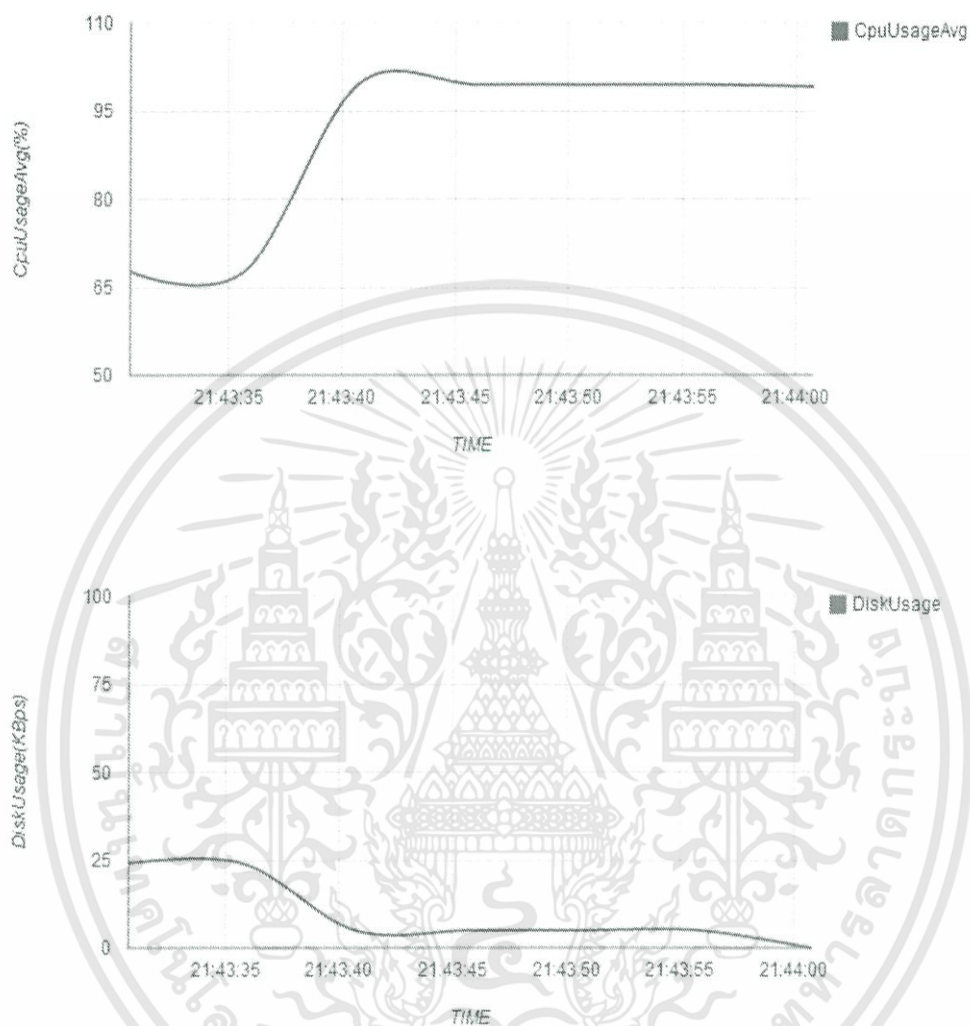
รูปที่ 3.2 แสดงหน้าเพจหลักของเว็บแอปพลิเคชัน

หน้าต่อมาก็คือ หน้า que แสดงข้อมูลการใช้ทรัพยากรของเครื่องเซิร์ฟเวอร์ ทั้งที่เปลี่ยนแปลงน้อยและกราฟแสดงข้อมูลที่เปลี่ยนแปลงแบบเรียลไทม์ ซึ่งในส่วนของกราฟจะสามารถแสดงข้อมูลย้อนหลังได้อีกด้วย ตามรูปที่ 3.3



รูปที่ 3.3 แสดงหน้าเพจของเครื่องเซิร์ฟเวอร์บนเว็บแอปพลิเคชัน

หน้าต่อมาก็คือ หน้า que แสดงข้อมูลความสามารถในการให้บริการเครื่องคอมพิวเตอร์เสมือน ทั้งที่เปลี่ยนแปลงน้อย และกราฟแสดงข้อมูลที่เปลี่ยนแปลงแบบเรียลไทม์ ซึ่งในส่วนของกราฟจะสามารถแสดงข้อมูลย้อนหลังได้อีกด้วย ตามรูปที่ 3.4



รูปที่ 3.4 แสดงหน้าเพจของเครื่องคอมพิวเตอร์เสมือนบนเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

4.1 การทดลองการส่งข้อมูลเพื่อตรวจสอบจำนวนสตอเรจที่ใช้งาน

ในการส่งข้อมูลระหว่างคลาวด์เซิร์ฟเวอร์และดาต้าเบส จึงมีการทดลองส่งข้อมูลในส่วนของข้อมูลการใช้ทรัพยากรของคลาวด์เซิร์ฟเวอร์และเครื่องคอมพิวเตอร์เสมือน เพื่อตรวจสอบดังนี้

```
#1 Update | Delete | New Field | Duplicate | Refresh | Text | Expand  
NumberInt(219),  
NumberInt(22813696),  
NumberInt(1765376),  
NumberInt(8744),  
NumberInt(92)  
"%CPU": 0  
}
```

รูปที่ 4.1 ภาพแสดงข้อมูลการใช้งานซีพียู

```
#1 Update | Delete | New Field | Duplicate | Refresh | Text | Expand  
NumberInt(0)  
"Number cpu": NumberInt(1),  
"Memory Info": [  
NumberLong(1059291136),  
NumberLong(41721856),  
NumberLong(1017569280),  
1.7  
]
```

รูปที่ 4.2 ภาพแสดงข้อมูลจำนวนซีพียู

```
#1 Update | Delete | New Field | Duplicate | Refresh | Text | Expand  
{  
"id": ObjectId("522752c66e955203b1915a18"),  
"Boot time": 1378307158,  
"network": [  
NumberInt(53227),  
NumberInt(2092441),  
NumberInt(304),  
NumberInt(17935),  
]
```

รูปที่ 4.3 ภาพแสดงข้อมูลระยะเวลาในการเปิดเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#1 Update | Delete | New Field | Duplicate | Refresh | Text | Expand

```

},
"Number cpu": NumberInt(1),
"Memory": [
  NumberLong(1059291136),
  NumberLong(41721856),
  NumberLong(1017569280),
  1.7
],
"Time": "2013-09-04 11:33:26"

```

รูปที่ 4.4 ภาพแสดงข้อมูลการใช้งานหน่วยความจำ

#1 Update | Delete | New Field | Duplicate | Refresh | Text | Expand

```

"Network": [
  NumberInt(53227),
  NumberInt(2092441),
  NumberInt(304),
  NumberInt(17935),
  NumberInt(0),
  NumberInt(0),
  NumberInt(0)
],

```

รูปที่ 4.5 ภาพแสดงข้อมูลการใช้งานเน็ตเวิร์ค

#1 Update | Delete | New Field | Duplicate | Refresh | Text | Expand

```

  NumberInt(0)
],
"Number cpu": NumberInt(1),
"Memory Info": [
  NumberLong(1059291136),
  NumberLong(41721856),
  NumberLong(1017569280),
  1.7
],

```

รูปที่ 4.6 ภาพแสดงข้อมูลจำนวนซีพียู

#1 Update | Delete | New Field | Duplicate | Refresh | Text | Expand

```

  NumberLong(1017569280),
  1.7
},
"Time": "2013-09-04 11:33:26",
"DISK": [
  NumberInt(1302),
  NumberInt(219),
  NumberInt(22813696),
  NumberInt(1725270)
],

```

รูปที่ 4.7 ภาพแสดงข้อมูลเวลาที่ทำการดึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองพบว่าดาต้าเบสที่มี แอปเครื่องคอมพิวเตอร์เสมือน ซึ่งส่งข้อมูลหนึ่งโรว์ นั้น จะมีการใช้งาน หน่วยความจำดังรูป

Objects	26
Data Size	2k
Storage Size	1.37m

รูปที่ 4.8 ภาพแสดงข้อมูลปริมาณสตอเรจที่ใช้งาน

จะเห็นได้ว่าหากมีปริมาณสตอเรจที่ใช้ต่อหนึ่งโรว์คือ 0.17125 เมกะไบต์ หากมีสตอเรจ จำนวน 500 จิกกะไบต์ จะสามารถรองรับข้อมูลทุก ๆ 1 วินาทีได้เป็นเวลา 33 วัน จึงจะทำให้ดาต้า เบสเซิร์ฟเวอร์ หนึ่งเครื่องที่มีขนาดของสตอเรจ 500 จิกกะไบต์นั้นเต็ม และไม่สามารถเก็บข้อมูลได้อีก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

จากการศึกษาและพัฒนาระบบดูแลโครงสร้างคลาวด์ ทำให้ได้รับความรู้เกี่ยวกับ การทำงานของคลาวด์คอมพิวติ้ง ดาต้าเบสแบบโนเอสคิวแอล สามารถเข้าใจการและใช้งานภาษาไพธอน ในการดึงและจัดเก็บข้อมูลจากมอดจโกตีปี การใช้งานเอแจ็กซีในการดึงข้อมูลแบบเรียลไทม์ การใช้งานภาษาจาวาสคริปต์ในการประมวลผลข้อมูลสำหรับเว็บแอปพลิเคชัน และการใช้งานกูเกิลชาร์ตเอพีไอในการแสดงข้อมูลในรูปแบบของกราฟ และพบว่าเครื่องคอมพิวเตอร์ที่ใช้สำหรับการติดตั้งไพธอนเซิร์ฟเวอร์ สำหรับการรับ-ส่งข้อมูลจากไฮเปอร์ไวเซอร์ VMware ESX นั้นจะต้องเป็นคอมพิวเตอร์ที่มีความเร็วของหน่วยประมวลผลที่สูง เพื่อที่จะทำให้การดึงข้อมูลเพื่อนำไปเก็บไว้ในดาต้าเบส และการดึงข้อมูลออกจากดาต้าเบสเพื่อมาแสดงผลนั้น มีความเร็วตามไปด้วย ซึ่งโครงการจะสามารถช่วยเหลือผู้ดูแลระบบที่ทำงานในระดับอินฟราสตรัคเจอร์ แอส อะ เซอร์วิส ของการให้บริการคลาวด์ในเรื่องการดูแลและจัดสรรการใช้ทรัพยากรภายในเครื่องเซิร์ฟเวอร์ รวมถึงเครื่องคอมพิวเตอร์เสมือนภายในเครื่องเซิร์ฟเวอร์เช่นกัน ว่ามีความผิดปกติหรือไม่ มีการใช้ทรัพยากรที่เกินขีดจำกัด (Overload) ซึ่งจะก่อให้เกิดผลกระทบต่อเครื่องคอมพิวเตอร์เสมือนภายในเครื่องเซิร์ฟเวอร์ตัวเดียวกันหรือไม่ และเนื่องจากโครงการนี้ออกแบบมาให้สามารถดึงข้อมูลการใช้ทรัพยากรของเครื่องเซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ VMware ESX และไฮเปอร์ไวเซอร์ KVM เท่านั้น ซึ่งแนวทางในการพัฒนาโครงการนี้ต่อไปก็คือการพัฒนาโปรแกรมที่ใช้ดึงข้อมูลการใช้ทรัพยากรมาจากเครื่องเซิร์ฟเวอร์ที่ใช้ไฮเปอร์ไวเซอร์ยี่ห้ออื่น ๆ เช่น Hyper-V , Citrix เป็น เพื่อความหลากหลายให้ผู้ดูแลระบบได้นำโครงการนี้ไปใช้ และการพัฒนาในส่วนการแสดงผล เนื่องจากโครงการนี้ได้พัฒนาให้ส่วนการแสดงผลนั้นเป็นเว็บแอปพลิเคชัน จึงมีข้อจำกัดที่ว่าผู้ใช้งานจะต้องมีอุปกรณ์หรือเครื่องมือที่สามารถเชื่อมต่ออินเทอร์เน็ตและใช้งานเว็บแอปพลิเคชันได้

บรรณานุกรม

- [1] Virtualization & Cloud Computing – About Cisco – Cisco Systems [Online]
Available : <http://www.cisco.com/web/TH/about/articles/virtualisation.html>
- [2] SLES Simplifies Server Virtualization | SUSE ConversationsSUSE [Online]
Available : <https://www.suse.com/communities/conversations/sles-simplifies-server-virtualization>
- [3] Yottag"IT"s: Cloud Computing in Detail [Online]
Available : <http://yotabits.blogspot.com/2011/06/cloud-computing-in-detail.html>
- [4] Buyya, James Broberg, Andrzej Goscinski. Cloud Computing : Principles and Paradigms. John Wiley & Sons, Inc. 2011.
- [5] Python Language Reference Manual – History of the software [Online]
Available : <http://www.network-theory.co.uk/docs/pylang/Historyofthesoftware.html>
- [6] RESTful Web Service | Design, Develop, Done! [Online]
Available : <http://designdevdone.wordpress.com/2012/02/19/restful-web-service/>
- [7] Types of Cloud Computing: Private, Public and Hybrid Clouds [Online]
Available : <http://blog.appcore.com/blog/bid/167543/Types-of-Cloud-Computing-Private-Public-and-Hybrid-Clouds>
- [8] NoSQL – Wikipedia, the free encyclopedia [Online]
Available : <http://en.wikipedia.org/wiki/NoSQL>
- [9] Linux Monitoring & Management [Online]
Available : <http://www.hyperic.com/products/linux-monitoring>
- [10] แนะนำ Windows Powershell ตอนที่ 1 [Online]
Available : <http://www.narisa.com/forums/index.php?app=blog&blogid=4&showentry=762>