สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

กรณีศึกษาการเปรียบเทียบประสิทธิภาพการประมวลผลแบบกลุ่มเมฆของวีเอ็มแวร์ โอเพ่นสแต็ค และไฮเปอร์วี

CASE STUDY PERFORMANCE COMPARISON OF VMWARE, OPENSTACK AND HYPER-V CLOUD COMPUTING

T146502

โดย

พิชญา พิทักษ์วงศ์

PICHAYA PITAKWONG

อาจารย์ที่ปรึกษา

รศ.ดร.โชติพัชร์ ภรณวลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 146502
วัน.เดือน.ปี 23 พ.ค. 2560

b 18842400
i.....

รายงานนี้เป็นส่วนหนึ่งของรายวิชาการศึกษาอิสระ 2

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CASE STUDY PERFORMANCE COMPARISON OF VMWARE, OPENSTACK AND HYPER-V CLOUD COMPUTING

PICHAYA PITAKWONG

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS OF THE COURSE

INDEPENDENT STUDY 2

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2016

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองการศึกษาอิสระ 2 (Independent Study 2)

เรื่อง

กรณีศึกษาการเปรียบเทียบประสิทธิภาพการประมวลผลแบบกลุ่มเมฆของวีเอ็มแวร์ โอเพ่นสแต็คและไฮเปอร์วี

CASE STUDY PERFORMANCE COMPARISON OF VMWARE, OPENSTACK AND HYPER-V CLOUD COMPUTING

นางสาวพิชญา พิทักษ์วงศ์

รหัสประจำตัว 57606064

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด

รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ

การศึกษาวิชาการศึกษาอิสระ 2 หลักสูตรวิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศ)

ภาคเรียนที่ 2 ปีการศึกษา 2558

.....อาจารย์ที่ปรึกษา

(รศ.ดร. โชติพัชร์ ภรณวลัย)

.....กรรมการสอบ

(รศ.อาริต ธรรมโน)

.....กรรมการสอบ

(ผศ.ดร.ปานวิทย์ ธุวะนุติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	กรณีศึกษาการเปรียบเทียบประสิทธิภาพการประมวลผลแบบกลุ่มเมฆของ วีเอ็มแวร์ โอเพ่นสแต็ค และไฮเปอร์วี
นักศึกษา	นางสาวพิชญา พิทักษ์วงศ์
รหัสนักศึกษา	57606064
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	เทคโนโลยีเครือข่ายและระบบ
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	รศ.ดร. โชติพัชร์ ภรณวลัย

บทคัดย่อ

ปัจจุบันเทคโนโลยี Cloud Computing เริ่มได้รับความนิยมเป็นอย่างมากเนื่องจากเป็นเทคโนโลยีที่เข้ามาช่วยบริหารจัดทรัพยากรของแต่ละองค์กร ช่วยลดต้นทุน รวมถึงการบริหารจัดการ และผู้ให้บริการปัจจุบันมีการแข่งขันกันสูงพยายามพัฒนาโครงสร้างพื้นฐานรวมถึงบริการให้ตอบสนองกับความต้องการของแต่ละองค์กร ซึ่งแต่ละผู้ให้บริการมีรูปแบบการให้บริการที่หลากหลายตั้งแต่การให้บริการโครงสร้างพื้นฐานไปถึงระดับแอพพลิเคชั่น เกิดทางเลือกมากขึ้น โดยหลายองค์กรสามารถนำระบบขึ้นมาทดสอบหรือเลือกใช้บริการกับผู้ให้บริการรายนั้นๆ ทั้งนี้แต่ละระบบงานต้องคำนึงถึงรูปแบบบริการที่เหมาะสมกับระบบงานขององค์กร ช่วยลดต้นทุนให้ต่ำลง สามารถเข้าถึงได้ง่าย และสามารถบริหารจัดการทรัพยากรได้อย่างมีประสิทธิภาพ การศึกษาค้นคว้านี้จะทำการศึกษารวมถึงทำการทดสอบระบบงานบนรูปแบบ Hypervisor ที่แตกต่างกันซึ่งผลลัพธ์ที่ได้จะสามารถนำไปประกอบความรู้ความเข้าใจของผู้ใช้บริการเพื่อให้สามารถเลือกรูปแบบบริการของระบบการประมวลผลแบบกลุ่มเมฆที่เหมาะสมกับระบบงานขององค์กรได้ ทำให้ค่าใช้จ่ายที่เกิดขึ้นเหมาะสมและเกิดประโยชน์สูงสุดต่อองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Case study performance comparison of VMware, OpenStack
and	Hyper-V cloud computing
Student	Miss. Pichaya Pitakwong
Student ID.	57606064
Degree	Master of Science
Program	Information Technology
Major	Network and System Technology
Academic Year	2015
Advisor	Assoc. Prof. Dr. Chotipat Pornavalai

ABSTRACT

Cloud Computing technology began to gain popularity due to being the technology to help managing the resources of each organization to reduce costs and management. Currently, service providers are highly competitive. Each organization is trying to improve the infrastructure and services. Each provider has a form of providing a wide range of services, from infrastructure to applications to result in more choices. Several organizations can test the system or use the service provider. However, each application must consider a form of service appropriate to the organization work system to reduce costs, easily access, and efficiently manage resources. The study was conducted to test the work system with different hypervisors. The results would be used to enhanace user's understanding and decision to choose the form of cloud computing system that is appropriate for the organization. The costs incurred would be reasonable and the organization would gain the highest benefit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานการศึกษาค้นคว้าอิสระฉบับนี้ได้รับความอนุเคราะห์จากบุคคลหลายท่านซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด

ขอขอบพระคุณรศ.ดร.โชติพัชร์ ภรณวลัย ซึ่งเป็นอาจารย์ที่ปรึกษาได้กรุณาสละเวลาให้ความรู้ แนวคิด คำแนะนำ รวมทั้งช่วยแก้ไขข้อผิดพลาดต่างๆ เพื่อให้รายงานการศึกษาค้นคว้าฉบับนี้มีความถูกต้องและสมบูรณ์ที่สุด

ขอขอบพระคุณพนักงานบริษัทอินเตอร์เน็ตประเทศไทย จำกัด มหาชน ที่คอยให้คำปรึกษาเป็นอย่างดี

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อโอฬาร พิทักษ์วงศ์ และคุณแม่วรรณดี พิทักษ์วงศ์ ผู้เป็นเบื้องหลังของความสำเร็จในการศึกษารายงานฉบับนี้ ที่คอยช่วยเหลือสนับสนุนให้กำลังใจที่ดีเสมอมา

พิชญา พิทักษ์วงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

ABSTRACT II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูป VI

สารบัญตาราง IX

บทที่ 1 ii

บทนำ ii

1.1 ที่มาและความสำคัญของปัญหา ii

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตการศึกษา 2

1.4 ขั้นตอนในการศึกษา 2

1.5 ผลที่คาดว่าจะได้รับ 3

บทที่ 2 4

ทฤษฎีเกี่ยวกับการประมวลผลแบบกลุ่มเมฆและเครือข่ายเชื่อมโยงการกระจายทรัพยากร 4

2.1 เครือข่ายเชื่อมโยงการกระจายทรัพยากร 4

2.2 เทคโนโลยีเครือข่ายการประมวลผลแบบกลุ่มเมฆ 4

2.3 รูปแบบโครงสร้างของ Cloud Computing และ Grid Computing 5

2.4 ระบบทรัพยากรเสมือน (Virtualization) 7

2.5 โครงสร้างการออกแบบ Hypervisor 8

2.6 รูปแบบการให้บริการ Cloud Computing 10

2.7 โครงสร้างพื้นฐานของ VMware 17

2.8 โครงสร้างพื้นฐานของ OpenStack 20

2.9 โครงสร้างพื้นฐานของ Hyper-V 23

บทที่ 3 26

ศึกษาระบบปัจจุบันและจัดเตรียมทรัพยากรสำหรับทดสอบระบบ 26

3.1 ระบบงานในปัจจุบัน 26

3.2 สร้างสภาพแวดล้อมสำหรับทดสอบระบบ 28

3.3 ฮาร์ดแวร์และรายละเอียดเครื่องเซิฟเวอร์เสมือนสำหรับทดสอบ 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.4 เครื่องมือสำหรับทดสอบประสิทธิภาพ 34

บทที่ 4 42

การทดลองและวิเคราะห์ผลการทดลอง 42

4.1 การทดสอบประสิทธิภาพของการประมวลผลของ CPU 42

4.2 การทดสอบประสิทธิภาพของหน่วยความจำ 46

4.3 การทดสอบประสิทธิภาพของดิสก์ 50

4.4 การทดสอบประสิทธิภาพของ Network Throughput ภายในเครือข่าย 56

4.5 เปรียบเทียบการลงทุนและค่าใช้จ่ายลิขสิทธิ์ License 63

บทที่ 5 67

สรุปผลการทดลอง 67

บรรณานุกรม 70

ประวัติผู้เขียน 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

2.1 โครงสร้างลำดับชั้นการทำงานของ Grid Computing [1] 6
2.2 โครงสร้างลำดับชั้นการทำงานของการประมวลผลแบบกลุ่มเมฆ [1] 7
2.3 ตัวอย่างโครงสร้างของ Hypervisor [2] 9
2.4 โครงสร้างของรูปแบบ Native หรือ Bare-Metal [2] 9
2.5 โครงสร้างของรูปแบบ Hosted [2] 10
2.6 รูปแบบ Cloud Service แบบ On-Premises [7] 11
2.7 รูปแบบ Cloud Service แบบ IAAS [7] 12
2.8 รูปแบบ Cloud Service แบบ PAAS [7] 13
2.9 รูปแบบ Cloud Service แบบ SAAS [7] 14
2.10 รูปแบบ Private Cloud 15
2.11 รูปแบบ Public Cloud 16
2.12 รูปแบบ Hybrid Cloud 17
2.13 โครงสร้างของ VMkernel และ Service Console [10] 18
2.14 โครงสร้างองค์ประกอบพื้นฐานของ VMware [12] 19
2.15 การทำงานของวีเซ็นเตอร์เซิฟเวอร์ [4] 20
2.16 สถาปัตยกรรมของโอเพ็นสแต็ค [4] 21
2.17 รูปแบบโครงสร้างเน็ตเวิร์กของโอเพ่นสแต็ค 23
2.18 โครงสร้างสถาปัตยกรรมของ Hyper-V [17] 24
2.19 รูปแบบการเชื่อมต่อเน็ตเวิร์กของไฮเปอร์วี [17] 25
3.1 ภาพรวมลักษณะการเชื่อมต่อเพื่อเข้าใช้งานระบบบน VMware 27
3.2 โครงสร้างการเข้าถึงทรัพยากรบน VMware 28
3.3 IP Configure VMware 29
3.4 VMware Vsphere Client 29
3.5 โครงสร้างการเข้าถึงทรัพยากรบน Hyper-V 30
3.6 Hyper-V Management 31
3.7 การติดตั้ง DevStack แบบ All-In-One 32
3.8 Specification Physical Server 35
3.9 Parameter Configuration บน IOmeter 38
3.10 ตั้งค่า parameter บน IOmeter 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

3.11 Configuration Client-Server บน IPerf3 40
4.1 กราฟเปรียบเทียบการประมวลผลแบบ 1 VCPU 44
4.2 กราฟเปรียบเทียบการประมวลผลแบบ 2 VCPU 45
4.3 กราฟเปรียบเทียบการประมวลผลแบบ 4 VCPU 46
4.4 ผลการทดสอบการอ่านเขียนหน่วยความจำของ Memory ขนาด 2 GB 48
4.5 ผลการทดสอบการอ่านเขียนหน่วยความจำของ Memory ขนาด 4 GB 49
4.6 ผลการทดสอบการอ่านเขียนหน่วยความจำของ Memory ขนาด 8 GB 50
4.7 Read - Write Operation 50
4.8 ผลลัพธ์การทดสอบ I/O ขนาดของ Disk 50 GB 51
4.9 ผลลัพธ์ Latency ของ Disk ขนาด 50 GB (ms) 52
4.10 ผลลัพธ์การทดสอบ I/O ขนาดของ Disk 100 GB 53
4.11 ผลลัพธ์ Latency ของ Disk ขนาด 100 GB (ms) 53
4.12 ผลลัพธ์การทดสอบ I/O ขนาดของ Disk 150 GB 54
4.13 ผลลัพธ์ Latency ของ Disk ขนาด 150 GB (ms) 55
4.14 Iperf3 บนเครื่อง Server เพื่อทดสอบ TCP 56
4.15 Iperf3 บนเครื่อง Client เพื่อทดสอบ TCP 57
4.16 Iperf3 บนเครื่อง Server เพื่อทดสอบ UDP 57
4.17 ผลลัพธ์จำลองเครื่อง VMS-1 เป็น Client และ VMS-2, VMS-3 เป็นเครื่อง Server 59
4.18 ผลลัพธ์การส่งข้อมูลแบบ TCP บน OpenStack Hypervisor 59
4.19 ผลลัพธ์การส่งข้อมูลแบบ TCP บน Hyper-V Hypervisor 60
4.20 ผลลัพธ์การส่งข้อมูลแบบ TCP บน VMware Hypervisor 60
4.21 ผลลัพธ์ส่งข้อมูลแบบ UDP บน VMware จาก VMS-1 ไปยัง VMS-2 61
4.22 ผลลัพธ์ส่งข้อมูลแบบ UDP บน VMware จาก VMS-1 ไปยัง VMS-3 61
4.23 ผลลัพธ์ส่งข้อมูลแบบ UDP บน Hyper-V จาก VMS-1 ไปยัง VMS-2 62
4.24 ผลลัพธ์ส่งข้อมูลแบบ UDP บน Hyper-V จาก VMS-1 ไปยัง VMS-3 62
4.25 ผลลัพธ์ส่งข้อมูลแบบ UDP บน OpenStack จาก VMS-1 ไปยัง VMS-2 62
4.26 ผลลัพธ์ส่งข้อมูลแบบ UDP บน OpenStack จาก VMS-1 ไปยัง VMS-3 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

4.27 cost compare : VMware On-Premise VS Cloud Service Provider 64

4.28 cost compare : MS Cloud On-Premise VS Cloud Service Provider 65

4.29 cost compare : OpenStack On-Premise VS Cloud Service Provider 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 ตารางแสดงรายละเอียดทรัพยากรเสมือนที่ใช้สำหรับทดสอบ 33
3.2 CPU Parameter 36
3.3 Memory Parameter 37
3.4 Disk Parameter 39
3.5 Iperf3 Parameter 41
4.1 ผลการทดสอบประสิทธิภาพของการประมวลผลของ CPU 42
4.2 ผลทดสอบประสิทธิภาพของหน่วยความจำ 47
4.3 ผลการทดสอบการส่งข้อมูลแบบ TCP 58
4.4 ผลการทดสอบการส่งข้อมูลแบบ UDP 60
5.1 ค่าใช้จ่ายลิขสิทธิ์หากเป็น Private Cloud 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ระบบการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) หมายถึง ระบบที่มีการรวมทรัยพากรด้านเทคโนโลยีสารสนเทศเข้าไว้ด้วยกันได้แก่ Compute, Storage และ Network และดึงทรัพยากรมาเหล่านั้นมาใช้ในการประมวลผลได้เพื่อให้สามารถใช้งานได้เต็มประสิทธิภาพ สามารถรองรับการประมวลผลข้อมูลจำนวนมากได้ด้วยการใช้ต้นทุนที่ต่ำกว่าการลงทุนเอง นอกจากนี้ยังสามารถเข้าถึงได้ง่ายสามารถบริหารจัดการทรัพยากรได้ง่าย โดยอาจจะอยู่ในพื้นที่เดียวกันหรือต่างพื้นที่กันโดยการเชื่อมโยงผ่านเครือข่ายอินเตอร์เน็ต ซึ่งจากแนวคิดการรวมเครื่องคอมพิวเตอร์หลายเครื่องเข้าด้วยกันทำให้มีประสิทธิภาพเทียบเท่ากับเครื่องแม่ข่ายขนาดใหญ่ สามารถประมวลผลข้อมูลได้ในปริมาณมาก รวมถึงการทำงานกับข้อมูลที่มีความซับซ้อน จึงถือได้ว่าโครงสร้างการประมวลผลแบบกลุ่มเมฆจึงเป็นโครงสร้างที่นำทรัพยากรไปใช้งานได้อย่างมีประโยชน์สูงสุด

ปัจจุบันผู้ให้บริการได้ออกแบบระบบการประมวลผลแบบกลุ่มเมฆโดยแบ่งเป็น 2 รูปแบบคือแบบ Private Cloud คือการทำระบบการประมวลผลแบบกลุ่มเมฆเพื่อใช้งานภายในองค์กร และอีกรูปแบบหนึ่งคือ Public Cloud ที่เปิดให้บริการแก่บุคคลทั่วไป องค์กร และหน่วยงานราชการ โดยลักษณะของ Public Cloud จะเป็นการให้บริการแบบ Infrastructure As A Service เป็นหลัก นอกจากนี้ยังเพิ่มทางเลือกในการให้บริการแก่ผู้ใช้บริการ โดยคำนึงถึงผู้ใช้บริการจะสามารถเลือกรูปแบบบริการที่เหมาะสมกับระบบงานขององค์กรโดยที่ใช้ต้นทุนต่ำ เข้าถึงได้ง่าย และสามารถบริหารจัดการทรัพยากรได้อย่างมีประสิทธิภาพ บนโครงสร้างพื้นฐานของการประมวลผลแบบกลุ่มเมฆ ดังนั้นระบบที่มีความสามารถแบบเดียวกัน รูปแบบการให้บริการหรือชื่อที่แตกต่างกันหรือความรู้ความเข้าใจของผู้ใช้บริการที่ไม่เพียงพอ อาจส่งผลให้ผู้ใช้บริการเกิดความสับสนกับการเลือกรูปแบบบริการของระบบการประมวลผลแบบกลุ่มเมฆที่ไม่เหมาะสมระบบงานขององค์กรได้ทำให้กลายเป็นการเพิ่มภาระค่าใช้จ่ายและการสิ้นเปลืองทรัพยากรโดยไม่มีประโยชน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาโครงสร้างพื้นฐานของการประมวลผลแบบกลุ่มเมฆ

1.2.2 เพื่อทดสอบประสิทธิภาพของระบบงานลูกค้าตามความต้องการของผู้ใช้งาน โดยอยู่บนโครงสร้างพื้นฐานของการประมวลผลแบบกลุ่มเมฆ

1.2.3 เพื่อเพิ่มองค์ความรู้ให้ผู้ใช้บริการสามารถเลือกใช้บริการได้ตรงตามความเหมาะสมของระบบงานและเกิดประโยชน์สูงสุดต่อองค์กร

1.3 ขอบเขตการศึกษา

1.3.1 ศึกษาโครงสร้างพื้นฐานของระบบการประมวลผลแบบกลุ่มเมฆ 3 แบบ คือ วีเอ็มแวร์ (VMware), โอเพ่นสแต็ค (OpenStack) และไฮเปอร์วี (Hyper-V)

1.3.2 ศึกษาผลการทดสอบประสิทธิภาพของระบบในการเก็บข้อมูลโดยทำวัด ความเร็วในการอ่านและเขียนข้อมูลดิสก์

1.3.3 ศึกษาผลการทดสอบประสิทธิภาพของหน่วยประมวลผลและหน่วยความจำ

1.3.4 ศึกษาผลการทดสอบประสิทธิภาพของระบบเครือข่าย โดยทำการวัดประสิทธิภาพในการส่ง/รับข้อมูลโดยหา Throughput การเชื่อมต่อไปยัง Server ภายในระบบเครือข่าย

1.4 ขั้นตอนในการศึกษา

1.4.1 ศึกษางานวิจัยที่เกี่ยวข้องกับการประมวลผลแบบกลุ่มเมฆ

1.4.2 ออกแบบโครงสร้างพื้นฐานในส่วนของการประมวลผลบนไฮเปอร์ไวเซอร์ VMware, OpenStack และ Hyper-V สำหรับจำลองระบบทดสอบเพื่อทดสอบและเก็บผลการทดลอง

1.4.3 สร้างระบบโครงสร้างพื้นฐานของ VMware, OpenStack และ Hyper-V เพื่อใช้จำลองระบบสำหรับทำการทดสอบ โดยทำการดัดแปลงองค์ประกอบโครงสร้างของระบบและ Configuration ให้ใกล้เคียงกัน

1.4.4 ทำการเปรียบเทียบการคำนวณค่าใช้จ่ายในส่วนของ License และการ Maintenance ระบบลูกค้าบนโครงสร้างพื้นฐาน VMware, OpenStack และ Hyper-V

1.4.5 วิเคราะห์ผลการทดสอบระบบและสรุปผลที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 เข้าใจทฤษฎีและเพิ่มองค์ความรู้ที่เกี่ยวกับการทำงานของระบบประมวลผลแบบกลุ่มเมฆ

1.5.2 เพิ่มทางเลือกให้กับผู้ใช้บริการตัดสินใจนำระบบงานมาติดตั้งใช้งานได้เต็มประสิทธิภาพ

1.5.3 เพื่อใช้เป็นข้อมูลช่วยเปรียบเทียบในการลดต้นทุนในส่วนของ License และการบริหารจัดการทรัพยากรได้คุ้มค่า

1.5.4 เพิ่มความคล่องตัวในการใช้งาน ช่วยลดค่าใช้จ่ายในการลงทุนศูนย์ข้อมูล (Data Center) และการบำรุงรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเกี่ยวกับการประมวลผลแบบกลุ่มเมฆและเครือข่ายเชื่อมโยงการกระจายทรัพยากร

การประมวลผลแบบกลุ่มเมฆและเครือข่ายเชื่อมโยงการกระจายทรัพยากรเป็นรูปแบบที่มีการศึกษาและพัฒนาโครงสร้างพื้นฐานเพื่อรองรับการใช้งานทรัพยากรที่ก่อให้เกิดความคุ้มค่าและมีความยืดหยุ่นในการใช้งานโดยมีรายละเอียดดังนี้

2.1 เครือข่ายเชื่อมโยงการกระจายทรัพยากร

ในปี 1995 ได้มีความร่วมมือกันระหว่างมหาวิทยาลัยและนักวิจัยในการพัฒนาระบบ Grid Computing โดยการนำคอมพิวเตอร์หลายเครื่องมารวมกันเพื่อให้ได้มาซึ่งระบบคอมพิวเตอร์ที่มีขนาดใหญ่หรือเทียบเท่าความสามารถของซุปเปอร์คอมพิวเตอร์ ที่สามารถประมวลผลด้วยความเร็วที่สูงและมีประสิทธิภาพ มีความยืดหยุ่น ซึ่งเป็นการนำทรัพยากรของในองค์กรหรือต่างองค์กร ไม่ว่าจะเป็นหน่วยประมวลผล หน่วยความจำ คลังข้อมูล และแบนด์วิธเครือข่ายมาเชื่อมโยงกันกลายเป็นระบบประมวลผลที่มีขนาดใหญ่ ที่สามารถแชร์ทรัพยากรเพื่อใช้ในการประมวลผลลักษณะงานที่เป็นแบบรูปแบบเดียวกัน หรือต่างกัน โดยอาศัยโครงสร้างพื้นฐาน (Infrastructure) ซึ่งอาจจะมีรูปแบบ รุ่น ที่แตกต่างกัน แต่รองรับการทำงานของ Application หรือข้อมูลที่แตกต่างกันหรือเหมือนกันได้ซึ่งการทำงานของ Grid Computing โดยทั่วไปจะอาศัยตัวกลางมาคอยช่วยสนับสนุนระบบการทำงาน ซึ่งตัวกลางจะคอยทำหน้าที่ควบคุมประสานงานระหว่าง Application และ Operating System โดยเรียกหน่วยทำงานนั้นว่า Middleware ซึ่งเมื่อทรัพยากรถูกนำมารวมกัน จนมีขนาดใหญ่การเชื่อมโยงผ่านระบบเครือข่ายที่มีอยู่ในการทำงานของระบบระบบเมื่อ Software มีความต้องการที่จะใช้งานทรัพยากรจะถูกนำมาจัดสรรแบ่งการใช้งานให้ผู้ใช้งาน สามารถเข้าถึงระบบประมวลนี้ได้ตามความต้องการของผู้ใช้ระบบ Grid Computing มีลักษณะทำให้สามารถใช้งานทรัพยากรขององค์กรที่อยู่ไกลออกไปได้ โดยจะทำงานในลักษณะการกระจายโดยมีข้อกำหนดการควบคุมภายใต้ VO (V-irtual Organization)

2.2 เทคโนโลยีเครือข่ายการประมวลผลแบบกลุ่มเมฆ

การประมวลผลแบบกลุ่มเมฆมีความคล้ายคลึงกันกันมากกับเครือข่ายการเชื่อมโยงแบบการกระจาย ถึงแม้จะเป็นการนำทรัพยากรหลายเครื่องมารวมกันเป็นทรัพยากรเสมือนที่มีขนาดใหญ่ มีความเร็วในการประมวลผลที่สูง ยืดหยุ่น และรองรับการขยายตัวของระบบ แต่การประมวลผลแบบกลุ่มเมฆจะตอบโจทย์ตามความต้องการของผู้ใช้และมีความยืดหยุ่นในด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรัพยากรเสมือนในการใช้งานที่มากกว่า เป็นลักษณะการแชร์ทรัพยากรร่วมกันโดยผู้ใช้เพียงแค่แจ้งความต้องการที่จะใช้งานกับผู้ให้บริการโดยไม่ต้องสนใจโครงสร้างการบริหารจัดการหรือมีองค์ความรู้ในส่วนของโครงสร้างพื้นฐานมากนักเพราะเป็นส่วนของผู้ให้บริการที่จะบริหารจัดการและคอยดูแล เปรียบเสมือนผู้ใช้งานมีหน้าที่หลักคือการเข้ามาใช้งานอย่างเดียว นอกจากนี้ยังสามารถเพิ่มหรือลดจำนวนทรัพยากรได้ รูปแบบจะเป็นลักษณะที่เรียกว่า Pay-Per-Use เป็นส่วนใหญ่ คือค่าใช้จ่ายก็จะจ่ายตามการใช้งานจริง ระบบการประมวลผลแบบกลุ่มเมฆจะเน้นการใช้งานผ่านระบบเครือข่ายอินเตอร์เน็ต และจะเป็นการเชื่อมโยงระยะไกล ซึ่งเมื่อเทียบกับเครือข่ายการเชื่อมโยงแบบกระจาย เมื่อมีขนาดที่ใหญ่ขึ้นต้นทุนจะค่อยข้างสูง แต่เมื่อเทียบกับการประมวลผลแบบกลุ่มเมฆจะมีต้นทุนที่ต่ำลงกว่าเดิม

2.3 รูปแบบโครงสร้างของ Cloud Computing และ Grid Computing

จากที่ได้กล่าวมาแล้วข้างต้นจะเห็นได้ว่ารูปแบบของ Grid Computing ส่วนใหญ่จะมุ่งเน้นไปที่ตัวระบบมากกว่าในขณะที่ Cloud Computing จะมุ่งเน้นไปที่ผู้ใช้งานเป็นหลัก ดังนั้นจะจำแนกโครงสร้างของทั้งสองแบบออกเป็นกรณีดังนี้

2.3.1 รูปแบบทางธุรกิจ

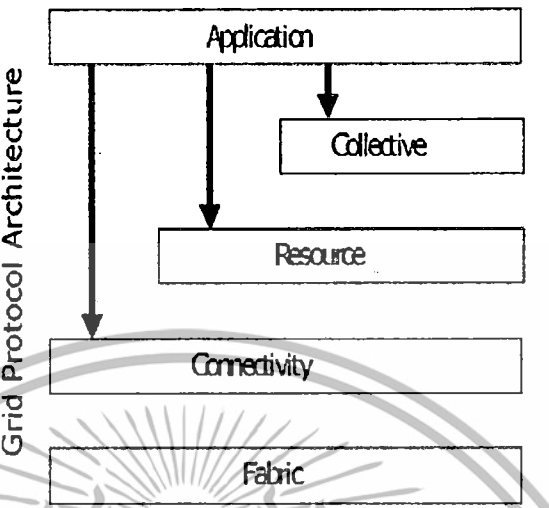
การสร้างระบบที่มีความสามารถให้ประมวลผลได้รวดเร็วนั้น ระบบต้องใช้ทรัพยากรที่มีประสิทธิภาพทำให้มีการลงทุนที่สูงเมื่อลองเปรียบเทียบเครือข่ายการเชื่อมโยงแบบกระจายมักถูกนำมาพัฒนาและใช้งานในหน่วยงานหรือสถาบันทางการศึกษาในด้านงานค้นคว้าและวิจัยซึ่งมีการนำระบบ Grid Computing เข้ามาใช้งานอย่างแพร่หลาย เพื่อพยายามเชื่อมโยงเครือข่ายของแต่ละหน่วยงานหรือสถาบันเข้าด้วยกันโดยใช้กฎและควบคุมภายใต้ VO (Virtual Organization) โดยพยายามรวมทรัพยากรให้เกิดเป็นระบบที่มีขนาดใหญ่สามารถประมวลผลได้เร็ว และให้แต่ละองค์กรสามารถใช้งานทรัพยากรเพื่อจะทำงานร่วมกันได้ แตกต่างจากระบบการประมวลผลแบบกลุ่มเมฆที่มุ่งเน้นตามการใช้งานของผู้ใช้งานเป็นหลักซึ่งศัพท์ที่ผู้ให้บริการหลายรายเรียกเหมือนกันคือ Pay-Per-Use เป็นลักษณะการใช้บริการแบบเช่าใช้และจ่ายตามที่มีการใช้งานจริง ค่าใช้จ่ายที่เกิดขึ้นจะถูกคำนวณออกมาโดยคิดอัตราการใช้งานเป็นลักษณะชั่วโมง คิดตามอัตราการโอนถ่ายข้อมูล หรือคิดตามขนาดของพื้นที่ที่มีการใช้งานต่อเดือน

2.3.2 โครงสร้างด้านสถาปัตยกรรม

ลำดับชั้นการทำงานซึ่งเป็นแนวความคิดพื้นฐานของเทคโนโลยีการประมวลผลแบบกลุ่มเมฆโครงสร้างด้านสถาปัตยกรรมของ Grid Computing ตามรูปที่ 2.1 และ Cloud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Computing ตามรูปที่ 2.2 สามารถจำแนกการทำงานเป็นลำดับชั้น ได้ 5 ชั้นการทำงานซึ่งประกอบด้วย [1]



รูปที่ 2.1 โครงสร้างลำดับชั้นการทำงานของ Grid Computing [1]

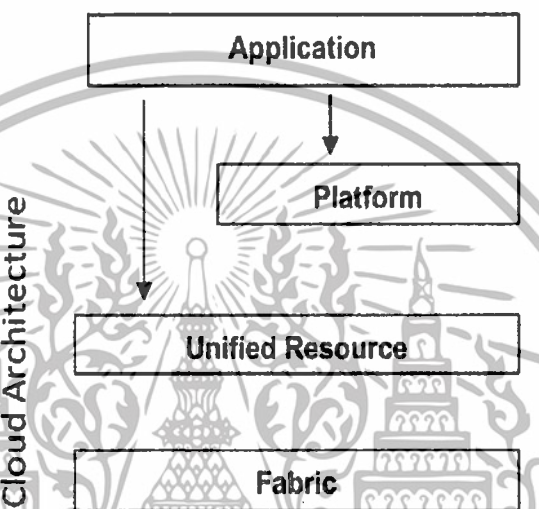
- Fabric Layer ชั้นนี้เป็นส่วนหนึ่งของการบริหารจัดการการเข้าถึงทรัพยากร หรือการใช้งานทรัพยากร เช่น กลุ่มของคอมพิวเตอร์ พื้นที่สำหรับจัดเก็บข้อมูล รวมถึงการเชื่อมโยงทรัพยากรระหว่างเครือข่าย
- Connectivity Layer ชั้นนี้ประกอบด้วยส่วนของโพรโตคอลที่ใช้ในการติดต่อสื่อสารมีหน้าที่หลักในการกำหนดการติดต่อสื่อสาร และกำหนดกฎเกณฑ์ที่ง่ายต่อการเข้าถึง รวมถึงการมีโพรโตคอลที่ช่วยกำหนดความมั่งคงปลอดภัย ซึ่งจะมีการกำหนดสิทธิ์ของผู้ใช้งาน และการเข้าถึงทรัพยากรของระบบ
- Resource Layer ชั้นนี้ทำหน้าที่ในการจัดการทรัพยากรระบบที่จะใช้ร่วมกัน
- Collective Layer ชั้นนี้จะทำหน้าที่คอยตรวจสอบการเข้าถึงทรัพยากรที่หลากหลายรูปแบบ ซึ่งจะประกอบด้วยบริการสำหรับการกู้คืนทรัพยากร การจัดแบ่งทรัพยากร การกำหนด Task ของการดำเนินงานรวมถึงการทำลักษณะของการ Replication ข้อมูล
- Application Layer ชั้นนี้จะประกอบด้วย Application ที่ใช้ในการทำงานร่วมกับ VO (Virtual Organization) ของสภาพแวดล้อมของ Grid Computing

ส่วนของโครงสร้างลำดับชั้นการทำงานของ Cloud Computing ตามรูปที่ 2.2 สามารถแบ่งได้ดังนี้

- Fabric Layer ชั้นการทำงานที่ทำหน้าที่เกี่ยวข้องกับส่วนทางกายภาพ ซึ่งประกอบด้วยทรัพยากรระบบพื้นฐาน เช่น เครือข่าย พื้นที่การจัดเก็บข้อมูล รวมถึงการประมวลผลข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Resource Layer ชั้นนี้จะทำหน้าที่ในการจัดการทรัพยากรที่จะใช้ร่วมกันซึ่งทรัพยากรที่มีการใช้งานบนโครงสร้าง Cloud Computing หลักจะเป็นลักษณะของทรัพยากรเสมือน
- Platform Layer ชั้นนี้จะมีตัวกลางที่ทำหน้าที่บริหารจัดการระบบโดยทำหน้าที่ให้บริการกับเครื่องทรัพยากรเสมือนให้สามารถทำงานกับระบบได้
- Application Layer ชั้นนี้จะเกี่ยวข้องกับโปรแกรมต่างๆ ที่จะใช้งานบนระบบ Cloud Computing



รูปที่ 2.2 โครงสร้างลำดับชั้นการทำงานของการประมวลผลแบบกลุ่มเมฆ [1]

2.4 ระบบทรัพยากรเสมือน (Virtualization)

ปัจจุบัน Hardware มีการพัฒนาอย่างรวดเร็วและมีประสิทธิภาพสูง ดังนั้นเพื่อให้เกิดความสอดคล้องกับเทคโนโลยีนี้โดยการสร้างสภาพแวดล้อมที่มีทรัพยากรเสมือนหลายเครื่องให้สามารถใช้งาน Software และ Application ในจำนวนที่มากขึ้นโดยสามารถทำงานบนทรัพยากรหลักแค่หนึ่งเครื่องคอมพิวเตอร์ ด้วยหลักการนี้ระบบ Cloud Computing ใช้หลักการของ Virtualization ซึ่งช่วยให้ผู้ใช้งานไม่ต้องรอคิวการใช้งานทรัพยากร เมื่อไหร่ที่ผู้ใช้งานต้องการทรัพยากรหรือต้องการเข้าใช้งานทรัพยากรที่พร้อมกัน ระบบจะมีทรัพยากรที่เพียงพอให้กับผู้ใช้งานนั้นๆ โดยปกติการทำงานของระบบทรัพยากรเสมือนหนึ่งๆ จะมี Software ตัวกลางที่ทำหน้าที่ในการบริหารจัดการทรัพยากรเสมือน คอยจัดสรรทรัพยากรของระบบ และเป็นตัวกลางประสานงานระหว่างทรัพยากรเสมือนกับฮาร์ดแวร์ ดังนั้นรูปแบบการบริหารจัดการทรัพยากรเสมือนสามารถแบ่งออกเป็น 3 แบบดังนี้ [3]

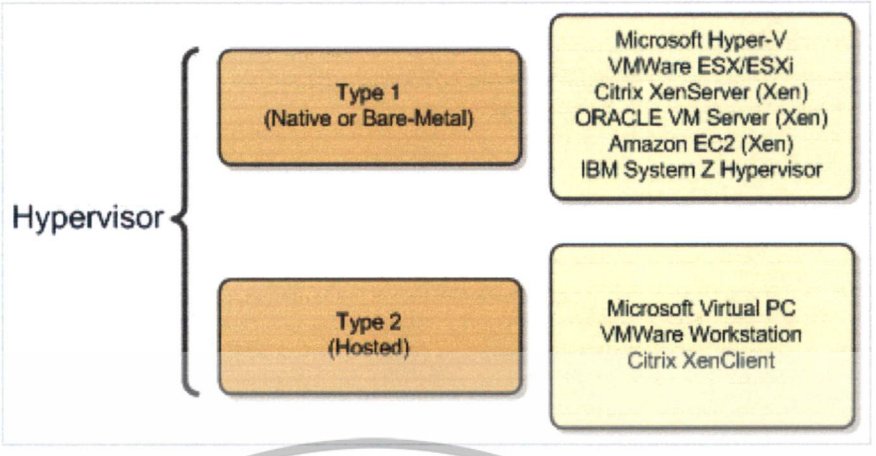
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Server Virtualization เป็นส่วนของการสร้างเครื่องเซิฟเวอร์จากทรัพยากรเสมือนที่เรียกว่า Virtual Machine เพื่อให้สามารถประมวลผลหรือคำนวณค่าได้สูงสุด
- Storage Virtualization การสร้างส่วนของการจัดเก็บข้อมูลโดยทำการแบ่งพื้นที่ใช้งานย่อยให้กับเครื่องเซิฟเวอร์เสมือน โดยหลักการคือนำ Physical Storage หลายๆ ส่วนมาทำการ Consolidate หลังจากนั้นทำเป็น Virtualization การใช้งานจะใช้งานเป็นลักษณะการใช้งาน Logical Storage Pool
- Network Virtualization ในระบบทรัพยากรเสมือนจะมีการแบ่งแยกในส่วนของ Local Area Network ออกจากกันทำให้สามารถแบ่งแยกระบบและใช้งานได้เต็มประสิทธิภาพ

2.5 โครงสร้างการออกแบบ Hypervisor

เทคโนโลยี Virtualization เป็นเทคโนโลยีที่ช่วยบริหารจัดการทรัพยากรทั้งในส่วนของการประมวล พื้นที่การจัดเก็บข้อมูล และเครือข่ายเน็ตเวิร์ก และทรัพยากรระบบอื่นๆ ให้มีประสิทธิภาพทั้งนี้จะมี Hypervisor เป็นตัวบริหารจัดการหรือที่เรียกอีกอย่างหนึ่งคือ Virtual Machine Manager (VMM) ที่เป็นโปรแกรมคอยบริหารจัดการให้ระบบปฏิบัติที่มีความแตกต่างกันหลายระบบปฏิบัติการสามารถทำงานร่วมกันได้บนอุปกรณ์ Hardware ตัวเดียวกัน ซึ่งเครื่องทรัพยากรเสมือนจะได้รับทรัพยากรในส่วนของหน่วยประมวลผล หน่วยความจำ พื้นที่จัดเก็บข้อมูลที่เป็นของตัวเอง Hypervisor จะบริหารกิจกรรมที่เกิดขึ้นเพื่อให้เครื่องคอมพิวเตอร์เสมือนสามารถใช้ทรัพยากรได้โดยที่ไม่ Disrupt มี Interface สำหรับผู้ดูแลระบบในระดับสูงสุดและคอยตรวจสอบสถานะการทำงานของเครื่องเซิฟเวอร์เสมือนอยู่ตลอดเวลา จากรูปที่ 2.3 เป็นตัวอย่างของโครงสร้างของ Hypervisor ที่มีการติดตั้งใช้งานในปัจจุบัน ซึ่งรูปแบบการแบ่งโครงสร้างของ Hypervisor มีอยู่ 2 รูปแบบคือโครงสร้างแบบ Native or Bare –Metal และโครงสร้างแบบ Hosted ซึ่งแต่ละแบบมีรายละเอียดตามโครงสร้างของ Hypervisor ดังนี้

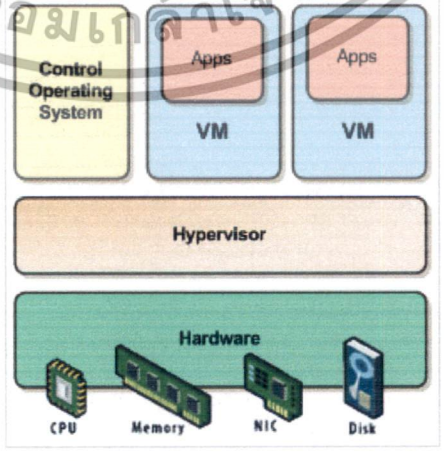
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ตัวอย่างโครงสร้างของ Hypervisor [2]

2.5.1. การออกแบบ Hypervisor แบบ Native หรือ Bare-Metal

จากรูปที่ 2.4 รูปแบบโครงสร้างของ Native หรือ Bare–Metal ซึ่งเป็นลักษณะของการออกแบบ Hypervisor โดยจะทำการติดตั้งลงบน Physical Hardware โดยตรงซึ่งจะทำหน้าที่เสมือนเป็นระบบปฏิบัติการซึ่งในการทำงานสามารถสร้างเครื่องเซิฟเวอร์เสมือนที่มี Guest Operation System (OS) แยกของแต่ละ VM (Virtual Machine) และติดตั้ง Apps (Application) ใช้งานแบบเดียวกันหรือแตกต่างกันได้หลายเครื่องให้ทำงานอยู่บน Hardware เดียวกันได้ โดยมี Control Operating System ทำหน้าที่คอยควบคุมการทำงานของระบบ ตัวอย่างการใช้งาน Hypervisor รูปแบบ Native ตามรูปที่ 2.3 ได้อธิบายไว้ก่อนหน้านี้ ได้แก่ Microsoft Hyper-V, VMware ESX/ESXI, Citrix XenServer (Xen), Amazon EC2 (Xen), IBM Sytem Z Hypervisor

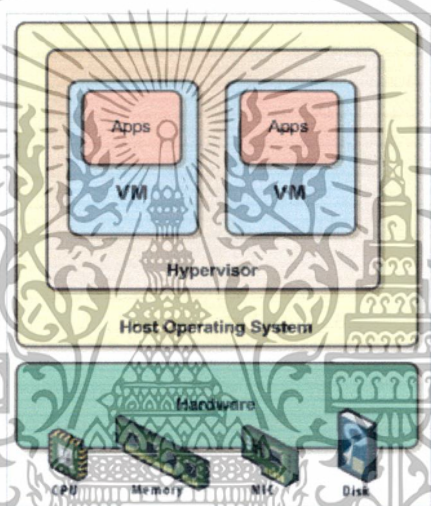


รูปที่ 2.4 โครงสร้างของรูปแบบ Native หรือ Bare-Metal [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2. การออกแบบ Hypervisor แบบ Hosted

รูปแบบโครงสร้างที่มีสถาปัตยกรรมของ Hypervisor แบบ Hosted ดังรูปที่ 2.5 การติดตั้งใช้งานที่จำเป็นต้องทำการติดตั้ง Host Operating System ใดๆ ลงไปบน Physical Hardware ก่อน โดย Hypervisor เป็นเพียง Application ที่มารันบนระบบปฏิบัติการนี้เท่านั้น หลังจากนั้นจึงจะสามารถสร้าง VM (Virtual Machine) แต่ละเครื่องและติดตั้ง Apps (Application) ใช้งานโดยอาจจะทำการสร้างเครื่องเซิฟเวอร์เสมือนที่มีรูปแบบเหมือนหรือแตกต่างกันได้ ตัวอย่างการใช้งาน Hypervisor ตามรูปที่ 2.3 ที่ได้อธิบายไว้ก่อนหน้านี้ได้แก่ Microsoft Virtual PC, VMware Workstation, Citrix XenClient

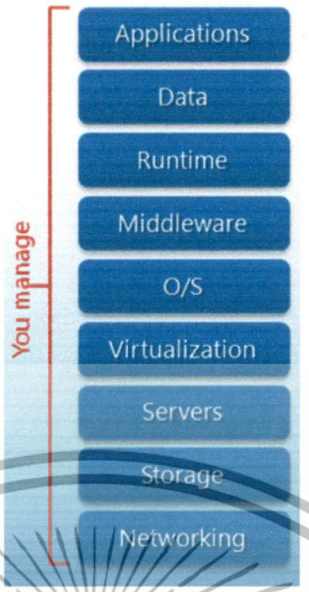


รูปที่ 2.5 โครงสร้างของรูปแบบ Hosted [2]

2.6 รูปแบบการให้บริการ Cloud Computing

รูปแบบระบบการจัดการเทคโนโลยีแบบ On-Premises ที่แสดงไว้ดังรูปที่ 2.6 แต่ละองค์กรต้องบริหารจัดการในส่วนของทรัพยากรเองทั้งหมดซึ่งประกอบไปด้วย Networking, Storage, Server Virtualization ไปจนถึงการ Application เอง ซึ่งรูปแบบนี้เรียกว่าอีกอย่างหนึ่งว่า On-Premises

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 รูปแบบ Cloud Service แบบ On-Premises [7]

ปัจจุบันผู้ให้บริการ Cloud Service หลายๆ รายมีรูปแบบการให้บริการที่หลากหลายซึ่งโครงสร้างการให้บริการระบบประมวลผลแบบกลุ่มเมฆโดยทั่วไปสามารถจำแนกรูปแบบการให้บริการออกเป็น 3 รูปแบบหลัก [3] ซึ่งรูปแบบการให้บริการมีตั้งแต่ทรัพยากรที่เป็นโครงสร้างพื้นฐานซึ่งประกอบด้วยส่วนของ Networking, Storage, Server, Virtualization ไปจนถึงระดับ Application ดังนี้

2.6.1 แบ่งตามโครงสร้างของการให้บริการ

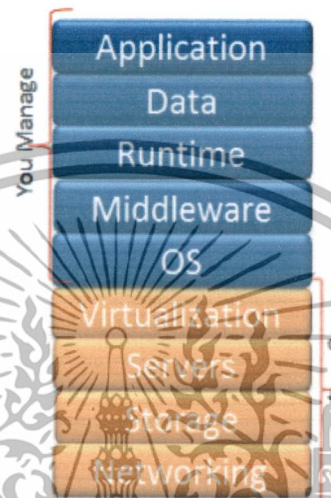
โครงสร้างของการให้บริการ Cloud ในปัจจุบันสามารถแบ่งได้เป็น 3 รูปแบบคือ IAAS (Infrastructure-As-A-Service), PAAS (Platform-As-A-Service), SAAS (Software-As-A-Service) แต่ละโครงสร้างมีรายละเอียดของการให้บริการดังนี้

2.6.1.1 IAAS (Infrastructure-As-A-Service)

โครงสร้างรูปแบบการให้บริการในลักษณะ IASS นั้นแสดงโครงสร้างไว้ตามรูปที่ 2.7 ผู้ให้บริการจะจัดเตรียมโครงสร้างพื้นฐานประกอบด้วย Networking, Storage, Server, Virtualization ให้ผู้ใช้งานไว้เรียบร้อยแล้วและจะบริหารจัดการทรัพยากรส่วนนี้ให้กับผู้ใช้บริการ ผู้ใช้บริการเพียงแค่แจ้งรายละเอียดในส่วนของทรัพยากรให้กับผู้ใช้บริการจัดเตรียมให้ ส่วนที่ทางผู้ใช้บริการต้องดำเนินการติดตั้งเองคือตั้งแต่ Operating System ขึ้นไปจนถึง Application ซึ่งโดยปกติทางผู้ใช้บริการต้องกำหนดขนาดของหน่วยประมวลผล ขนาดของหน่วยความจำ ขนาดของพื้นที่สำหรับการจัดเก็บข้อมูล และเครือข่ายเน็ตเวิร์กที่ต้องการใช้งาน หลังจากได้รับ Resource แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปการเข้าบริหารจัดการทรัพยากรเหล่านี้จะใช้ผ่านทางระบบอินเตอร์เน็ตเป็นหลัก ผู้ใช้บริการสามารถร้องขอเพื่อใช้งานทรัพยากรเสมือนได้ภายในไม่กี่นาที [9] ตัวอย่างผู้ให้บริการด้วยโครงสร้าง IASS เช่น INET, Amazon, Rackspace จากขอบเขตของ IAAS จะเห็นได้ว่า ผู้ใช้งานจะต้องดูแลและบริหารจัดการตั้งแต่ Operating System ไปจนถึง Application

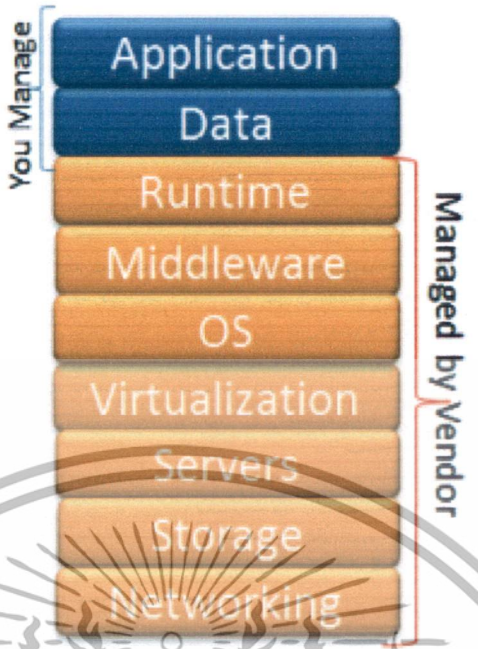


รูปที่ 2.7 รูปแบบ Cloud Service แบบ IAAS [7]

2.6.1.2. PAAS (Platform-As-A-Service)

โครงสร้างรูปแบบการให้บริการตามรูปที่ 2.8 เป็นโครงสร้างของรูปแบบการให้บริการที่ผู้ให้บริการจัดเตรียมโครงสร้างพื้นฐานประกอบด้วย Networking, Storage, Server, Virtualization, Operating System, Middleware และ Runtime ไว้ให้ผู้ใช้งานเรียบร้อยแล้ว การบริหารจัดการเป็นหน้าที่ของผู้ให้บริการที่จะดูแลในส่วนนี้ ผู้ใช้งานเพียงแค่นำ Data และ Application ที่ต้องการใช้งานมาติดตั้ง โดยทั่วไปการเข้าบริหารจัดการทรัพยากรเหล่านี้จะผ่านทางระบบอินเตอร์เน็ตเป็นหลัก เช่นเดียวกับรูปแบบการใช้งาน IAAS ตัวอย่างผู้ให้บริการด้วยโครงสร้าง PAAS เช่น Microsoft (Azure), Amazon E2, Google App Engine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

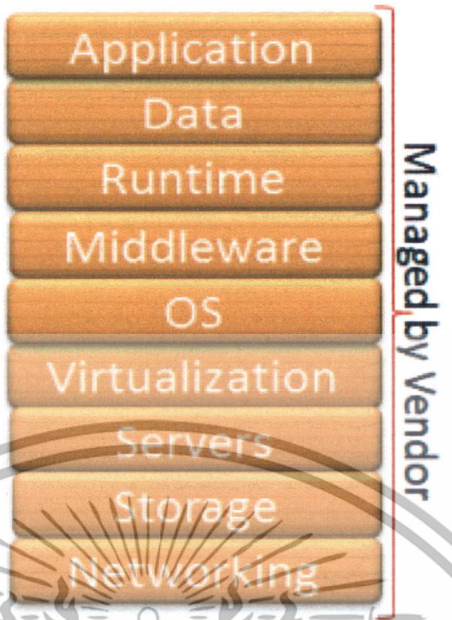


รูปที่ 2.8 รูปแบบ Cloud Service แบบ PAAS [7]

2.6.1.3. SAAS (Software-As-A-Service)

โครงสร้างรูปแบบการให้บริการตามรูปที่ 2.9 เป็นโครงสร้างรูปแบบการให้บริการที่ผู้ให้บริการจัดเตรียมโครงสร้างพื้นฐานประกอบด้วย Networking, Storage, Server, Virtualization, Operationg System, Middleware, Runtime, Data, Aplication การบริหารจัดการทั้งหมดจะเป็นส่วนของผู้ให้บริการดูแลทั้งหมด ผู้ใช้บริการเมื่อต้องการเข้าถึง Application บนเครื่องเซิฟเวอร์เสมือนผ่านทางเครือข่ายอินเตอร์เน็ตหรืออาจจะผ่านเว็บเบราว์เซอร์ที่ผู้ให้บริการจัดหาไว้ให้ ตัวอย่างผู้ให้บริการด้วยโครงสร้าง SAAS เช่น Netflix, Cisco WebEx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 รูปแบบ Cloud Service แบบ SAAS [7]

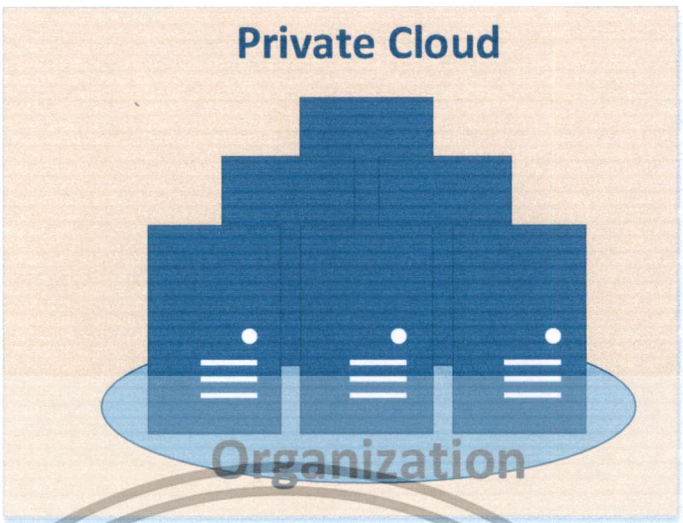
2.6.2 แบ่งตามรูปแบบการใช้งานทรัพยากร

จากหัวข้อ 2.6.1 เป็นรูปแบบการแบ่ง Cloud ตามโครงสร้างของผู้ให้บริการ นอกจากนี้ระบบ Cloud Computing ในปัจจุบันยังสามารถแบ่งแยกตามการติดตั้งใช้งานทรัพยากร [8] ซึ่งสามารถจำแนกรูปแบบ Cloud ออกเป็น 3 รูปแบบดังนี้

2.6.2.1 รูปแบบ Cloud แบบส่วนตัว (Private Cloud Computing)

รูปแบบ Private Cloud ตามรูปที่ 2.10 เป็นระบบ Cloud ที่เรียกอีกอย่างหนึ่งว่า Cloud แบบ On-Premises นั่นคือเป็นการสร้างระบบ Cloud ที่มีการใช้งานภายใน Organization เท่านั้น โดยเป็นการใช้งานเทคโนโลยีเวอร์ชวลไลเซชั่นและทรัพยากรที่อยู่ทั้งหมดอยู่ภายในศูนย์ข้อมูลขององค์กร มีการบริหารจัดการและดูแลกันเองภายในองค์กร และไม่มีการเชื่อมต่อเครือข่ายเพื่อใช้งานทรัพยากรจากภายนอก ดังนั้นจะมีการควบคุมในส่วนของ Security ที่ค่อนข้างสูง มีความเป็น Private รวมถึงการดูแลส่วนของ Maintenance ระบบเองด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

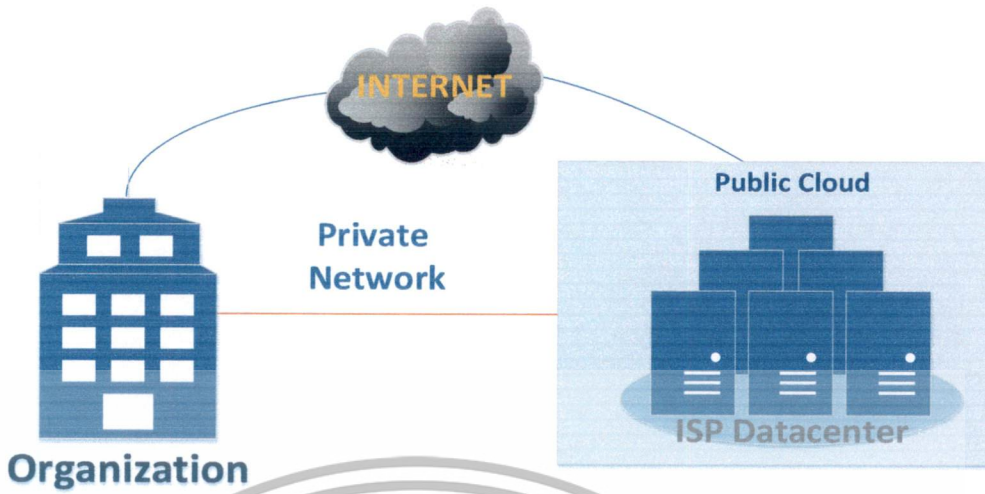


รูปที่ 2.10 รูปแบบ Private Cloud

2.6.2.2 รูปแบบ Cloud แบบสาธารณะ (Public Cloud Computing)

รูปแบบ Public Cloud Computing เป็นรูปแบบที่มีการใช้ทรัพยากรภายนอกองค์กรทั้งหมด จากรูปที่ 2.11 Server ทั้งหมดจะถูกย้ายไปอยู่บน Datacenter ของผู้ให้บริการ (ISP Datacenter) ซึ่งโครงสร้างพื้นฐานที่ผู้ให้บริการมีเช่น Compute, Network และ Storage จะมีการหลายองค์มาแชร์กันใช้ทรัพยากรนั้น ในมุมของผู้ใช้งานจะเข้าใช้บริการผ่านช่องทางอินเตอร์เน็ตเป็นหลักหรืออาจจะมีการเชื่อมต่อเน็ตเวิร์กจากองค์กรเพื่อไปใช้งานทรัพยากรที่อยู่บน ISP Datacenter ด้วยโครงข่ายที่เป็นเครือข่ายเน็ตเวิร์กส่วนตัว เช่น ผ่านบริการ Link MPLS ของ ISP ดังนั้นการควบคุมในส่วนของ Security ต่างๆ รวมถึงการบำรุงรักษา (Maintenance) ระบบเป็นความรับผิดชอบในส่วนของผู้ให้บริการ ผู้ใช้บริการสนใจแต่เพียงบริการที่ได้รับเท่านั้น และค่าใช้จ่ายที่เกิดเป็นลักษณะการเช่าใช้โดยคิดเป็นรายชั่วโมงหรือคำนวณตามการใช้งานจริง (Pay-Per-Use)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



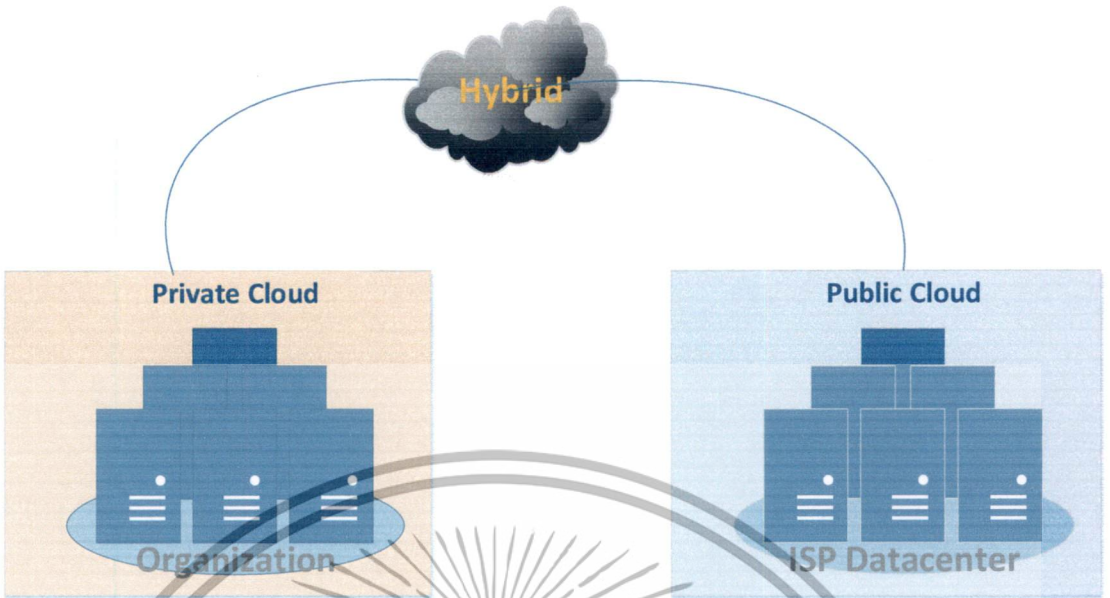
รูปที่ 2.11 รูปแบบ Public Cloud

2.6.2.3 รูปแบบ Cloud แบบผสม (Hybrid Cloud Computing)

โครงสร้างรูปแบบ Hybrid Cloud Computing เป็นโครงสร้างที่มีการใช้งานทรัพยากรร่วมกันทั้งฝั่งผู้ให้บริการและทรัพยากรฝั่งผู้ใช้บริการ ดังแสดงตามรูปที่ 2.12 ผู้ใช้บริการจะมีการใช้งานทรัพยากรทั้งภายใน Organization และทรัพยากรที่ ISP Datacenter ได้ทั้งสองฝั่งทั้งนี้เพื่อรองรับการทำงานของระบบงาน โดยปกติหากเป็นลักษณะการใช้งานแบบนี้ทรัพยากรหลักจะมีการใช้งานภายใน Organization เมื่อทรัพยากรใน Organization ไม่เพียงพอผู้ใช้งานสามารถดึงทรัพยากรจากผู้ให้บริการมาใช้งานได้ นอกจากนี้การใช้งานแบบ Hybrid ยังช่วยให้องค์กรสามารถกระจาย Workload ในการทำงานของระบบโดยสามารถกระจายโหลดระบบงานให้มีการใช้งานระบบพร้อมกันได้ทั้งสองฝั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



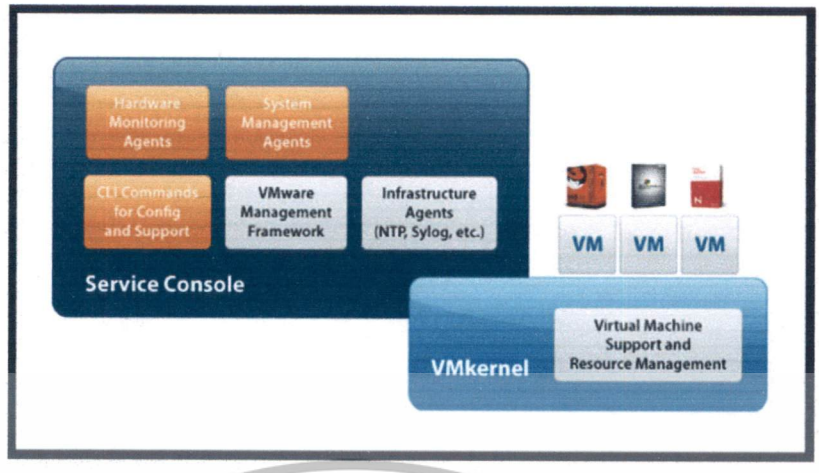
รูปที่ 2.12 รูปแบบ Hybrid Cloud

2.7 โครงสร้างพื้นฐานของ VMware

รูปแบบโครงสร้างพื้นฐานของ VMware เป็นเทคโนโลยีในส่วนของการทำ Server, Storage, Network และ Virtualization ที่ช่วยลดต้นทุนและช่วยบริหารจัดการ Datacenter ให้กับผู้ใช้บริการได้ค่อนข้างมาก เทคโนโลยีของ VMware จะใช้หลักการออกแบบสถาปัตยกรรมของ Hypervisor แบบ Bare-Metal หรือ Native ซึ่งมีการติดตั้ง Hypervisor ที่มีชื่อเรียกว่า ESX Hypervisor โดยสามารถติดตั้งลงบน Hardware ได้โดยตรงไม่จำเป็นต้องลงระบบปฏิบัติการใดๆ ก่อน ดังที่แสดงไว้ตามรูปที่ 2.13 โครงสร้างของ VMkernal และส่วนของ Service Console นั้นในการบริหารจัดการของ ESX จะมีส่วนของ VMkernel ที่ทำหน้าที่คอยบริหารจัดการทรัพยากรทางด้านฮาร์ดแวร์รวมถึงบริหารจัดการ Virtual Machine (VM) [11]

146502

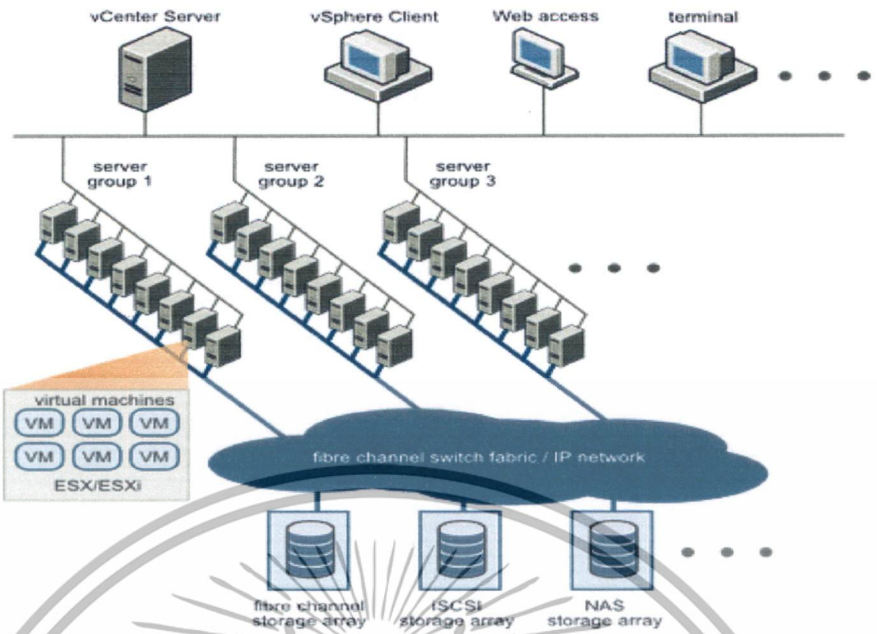
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 โครงสร้างของ VMkernel และ Service Console [10]

นอกจากนี้ส่วนของ Service Console ยังทำหน้าที่เป็นตัวกลางในการบริหารจัดการซึ่งประกอบด้วยส่วนของ Hardware Monitoring Agents ทำหน้าเฝ้าระวังการทำงานหรือความผิดปกติที่เกิดขึ้นจาก Hardware ส่วนของ System Management Agent ในการบริหารทรัพยากรเสมือน ส่วนของ CLI Command for Configuration and Support เป็นส่วนของการปรับแก้พารามิเตอร์ต่างๆบนระบบผ่าน Command รองรับส่วนของ Management Framework ต่างๆ ที่จะติดตั้งหรือเชื่อมต่อกับระบบ ส่วนสุดท้ายคือ Infrastrcuture Agents ได้แก่ NTP, Syslog บนระบบ ในการเข้าไปจัดการทรัพยากรหรือปรับตั้งค่าการใช้งานต่างๆ บน VMware Hypervisor นั้น สามารถเข้าผ่าน Service Console ซึ่งจะเป็นตัวกลางที่จะช่วยบริหารจัดการระหว่าง Physical Hardware และ VMkernel ของ Hypervisor นอกจากนี้ Service Console ยังทำหน้าที่คอยตรวจสอบความพร้อมใช้ของฮาร์ดแวร์รวมไปถึงสถานะของระบบตามนอกจากนี้ โครงสร้าง VMware ตามรูปที่ 2.14 นั้น Service Console หรือเรียกอีกอย่างหนึ่งว่า vCenter Server สามารถใช้บริหารจัดการหรือแบ่งกลุ่มทรัพยากรได้หลายกลุ่ม นอกจากนี้ยังสามารถเข้าบริหารจัดการทรัพยากรได้ทาง Web Access, vSphere Client หรือ Terminal ได้ ซึ่งรูปแบบของพื้นที่ในการจัดเก็บข้อมูลบน VMware นั้นรองรับทั้งแบบ Fibre Channel, ISCSI และ NAS Storage

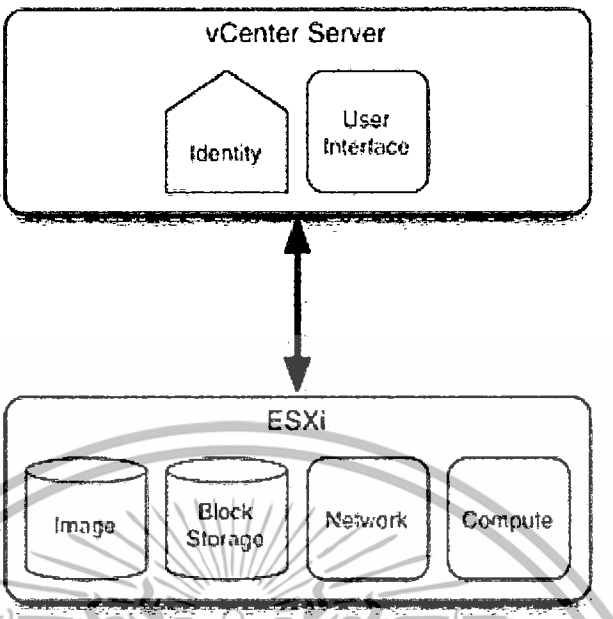
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 โครงสร้างองค์ประกอบพื้นฐานของ VMware [12]

vCenter Server ตามรูปที่ 2.15 ถูกออกแบบมาให้เป็น Single Point ในการบริหารจัดการ Cloud Datacenter เช่น การควบคุมการเข้าถึง ประสิทธิภาพระบบ การเฝ้าระวังการทำงาน และการปรับตั้งค่าการใช้งาน หากต้องการสร้างทรัพยากรเสมือนจะคอยสั่งการไปยัง ESX/ESXi ซึ่งเป็น VMkernal ให้ทำการสร้าง Image , Block Storage, Network และกลุ่มของ Compute ESX/ESXI จะควบคุมและกำหนดทรัพยากร เช่น หน่วยประมวลผล หน่วยจัดเก็บข้อมูลและกำหนด Network Interface Card ให้กับแต่ละ VM โดยช่องทางหลักในการเข้าถึง vCenter จากที่ได้กล่าวมาแล้ว นอกจาก vCenter แล้วอาจจะผ่านทาง Web Service หรือผ่านทาง Vsphere Client ซึ่ง Vsphere Client สามารถใช้ในลักษณะของการ Remote ไปจัดการ ESXi Server ได้โดยตรง โดยปกติการเข้าถึงระบบได้นั้นจำเป็นต้องมีการตรวจสอบสิทธิ์และยืนยันตัวตนของผู้เข้าใช้งาน (Identity) ก่อนเสมอซึ่งส่วนของการ Authentication อาจจะประกอบด้วย Server AD และ Server SSO ที่ทำหน้าที่ในการกำหนด Role และกำหนดสิทธิ์ในการเข้าถึงของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 การทำงานของวีเซ็นเตอร์เซิฟเวอร์ [4]

2.8 โครงสร้างพื้นฐานของ OpenStack

รูปแบบโครงสร้างพื้นฐานของ OpenStack เป็น OpenSource ที่ถูกพัฒนาขึ้นมาจาก 2 องค์กรคือ Rackspace และ NASA ภายใต้ลิขสิทธิ์ Apache 2.0 โดยเปิดให้มีการใช้งานแบบไม่มีค่าลิขสิทธิ์ ถูกออกแบบมาให้ง่ายในการบริหารจัดการกลุ่มของ Server, Storage และ Network ภายในศูนย์ข้อมูล ซึ่งแสดงรายละเอียดของโครงสร้างไว้ตามรูปที่ 2.16 OpenStack จะมีส่วนของ Service และโมดูลหลายส่วน โดยจะมีส่วนของ KVM ทำหน้าที่ในการบริหารจัดการทรัพยากรของระบบให้กับเครื่องเซิฟเวอร์เสมือน โดยสามารถแบ่งแยกองค์ประกอบตามหน้าที่การทำงานได้ดังนี้

2.8.1 องค์ประกอบพื้นฐานของ OpenStack

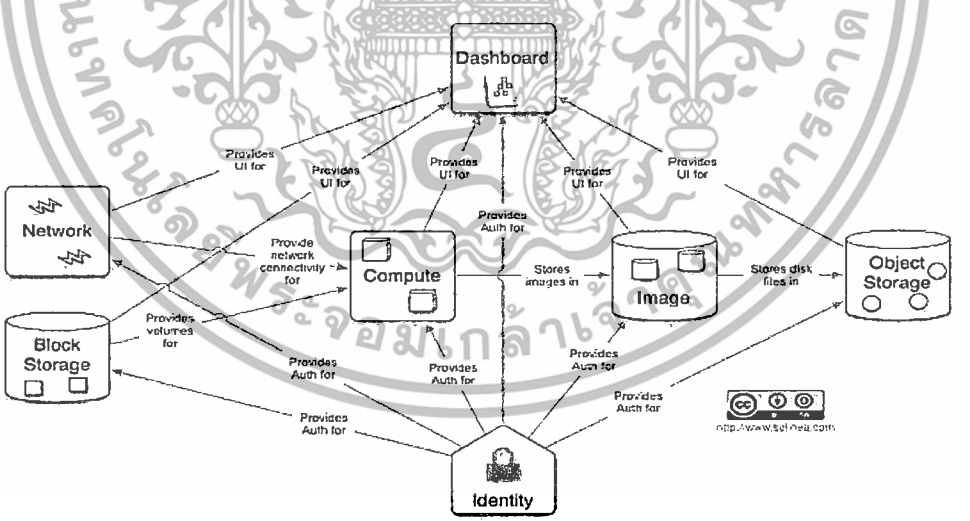
องค์ประกอบพื้นฐานของ OpenStack ประกอบด้วย Service และโมดูลดังนี้

- Horizon เป็นส่วนของ Dashboard ช่วยให้ผู้ใช้งานง่ายต่อการเข้าถึงทรัพยากรได้แก่ กลุ่มของ Compute, Storage และ Network ที่จำเป็นสำหรับการสร้าง Virtual Machine ได้อย่างเต็มประสิทธิภาพ ซึ่งผู้ใช้งานเองสามารถที่จะเข้าใช้งานหรือสร้าง Virtual Machine ได้จากส่วนนี้
- Nova เป็นส่วนของการบริหารจัดการ Compute Service สำหรับดำเนินการสร้างและบริหารจัดการ Virtual Machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Neuton เป็นส่วนที่ใช้ในการบริหารจัดการ Virtual Network Interface ให้กับ Nova นอกจากนี้ยังอนุญาตให้ผู้ใช้งานสามารถสร้างหรือกำหนดเน็ตเวิร์กภายในได้เอง
- Glance เป็นส่วนของ Storage Image ทำหน้าที่เก็บ Image ของ Virtual Disk ใน Object Storage ที่เรียกว่า Swift สำหรับการใช้งาน Sever Image ของระบบ
- Swift เป็นส่วนของ Object Storage โดยมีความสามารถในการจัดเก็บข้อมูลแยกตามประเภทโครงสร้างการใช้งาน
- Keystone คือส่วนของการจัดการ Identity หน้าที่หลักของส่วนนี้คือการกำหนดสิทธิ์ในการเข้าใช้งานของผู้ใช้งานในแต่ละโมดูล เพื่อเพิ่มความปลอดภัยในการเข้าถึงระบบของ OpenStack
- Cinder คือส่วนที่ทำหน้าที่จัดการ Block Storage จัดเก็บข้อมูลของแต่ละ VM
- Ceilometer คือส่วนที่ทำหน้าที่คอยตรวจสอบระบบและเฝ้าระวังการใช้งานของทรัพยากรของระบบ ได้แก่ หน่วยประมวลผล หน่วยความจำ และเน็ตเวิร์ก
- Heat เป็น Orchestration Tool ในการสร้าง Template ทำให้ง่ายต่อการสร้างทรัพยากรใช้งาน

โดยแต่ละองค์ประกอบภายในของโครงสร้างการใช้งาน OpenStack นั้นสามารถอธิบายความสัมพันธ์ในการทำงานร่วมกันได้ดังรูปที่ 2.16



รูปที่ 2.16 สถาปัตยกรรมของโอเพ็นสแต็ค [4]

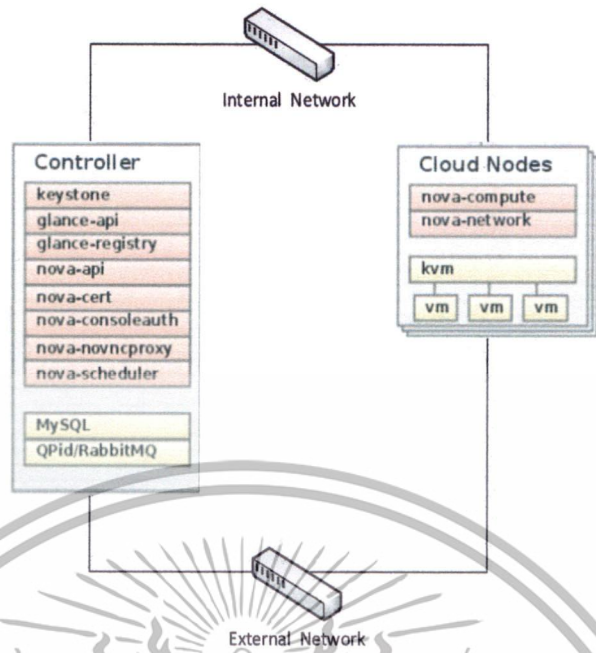
จากองค์ประกอบที่ได้กล่าวมาข้างต้น โครงสร้างสถาปัตยกรรมของ OpenStack ตามรูปที่ 2.16 มีหลักการทำงานพื้นฐานดังนี้ เมื่อผู้ใช้งานต้องการสร้างเครื่องเซิฟเวอร์เสมือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นใหม่ ผู้ใช้งานจะสั่งงานผ่าน Dashboard ซึ่งเป็นส่วนของ User Interface ที่เรียกว่า Horizon หรือจะใช้คำสั่งผ่านทาง Command Line จากนั้นคำสั่งจะถูกส่งไปยัง Keystone เพื่อตรวจสอบสิทธิ์หรือยืนยันตัวตน (Identity) ก่อนการเข้าใช้งานทรัพยากรต่างๆ ได้แก่ ส่วนของ Network, Block Storage, Compute, Image และ Object Storage เมื่อผู้ใช้งาน Request ทรัพยากร Nova-API จะทำหน้าที่เชื่อมต่อไปยัง Compute เพื่อขอจัดสรรทรัพยากรเสมือนในการสร้าง Instance เชื่อมต่อไปยังส่วนของการจัดสรรพื้นที่จัดเก็บข้อมูล และติดต่อไปยัง Neutron เพื่อสร้าง Virtual Network Interface ให้กับทรัพยากรเสมือนหลังจากระบบได้ทำการสร้างทรัพยากรเสมือนขึ้นมาเรียบร้อยแล้ว ระบบจะจัดสรร IP-Address ให้กับเครื่องเซิฟเวอร์ ดังนั้นผู้ใช้งานจะได้รับ IP Address สำหรับเข้าใช้งานทรัพยากร

จากรูปที่ 2.17 โครงสร้างการทำงานของ OpenStack องค์ประกอบหลักมี 2 ส่วนคือ ส่วนของ Controller Node และส่วนของ Cloud Node โดยหน้าที่หลักของ Controller Node เป็นส่วนที่คอยบริหารจัดการทุกอย่างของระบบ การจัดการทรัพยากร การจัดการฐานข้อมูล การจัดการด้านการแก้ไขหรือปรับปรุงค่า Configuration ต่างๆ รวมถึง API ในขณะที่ Cloud Node จะเป็นส่วนที่เกี่ยวข้องกับการจัดการทรัพยากรเสมือน รวมถึงการทำ Virtual Network Interface ให้กับทรัพยากรเสมือน โดยลักษณะการเชื่อมต่อระหว่าง Controller และ Cloud Nodes สามารถเชื่อมต่อได้ทั้งรูปแบบที่เป็น Internal Network และ External Network โดยส่วนใหญ่หากเป็น System ภายในจะมีการใช้งานในลักษณะที่เป็น Internal Netork เท่านั้น และส่วนที่ต้องการให้ผู้ใช้งานภายนอกเข้ามาใช้งาน เช่น ส่วนทรัพยากรเสมือนที่มีการสร้างขึ้นใช้งานจะมีการใช้งาน External Network เป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 รูปแบบโครงสร้างเน็ตเวิร์กของโอเพ่นสแต็ค

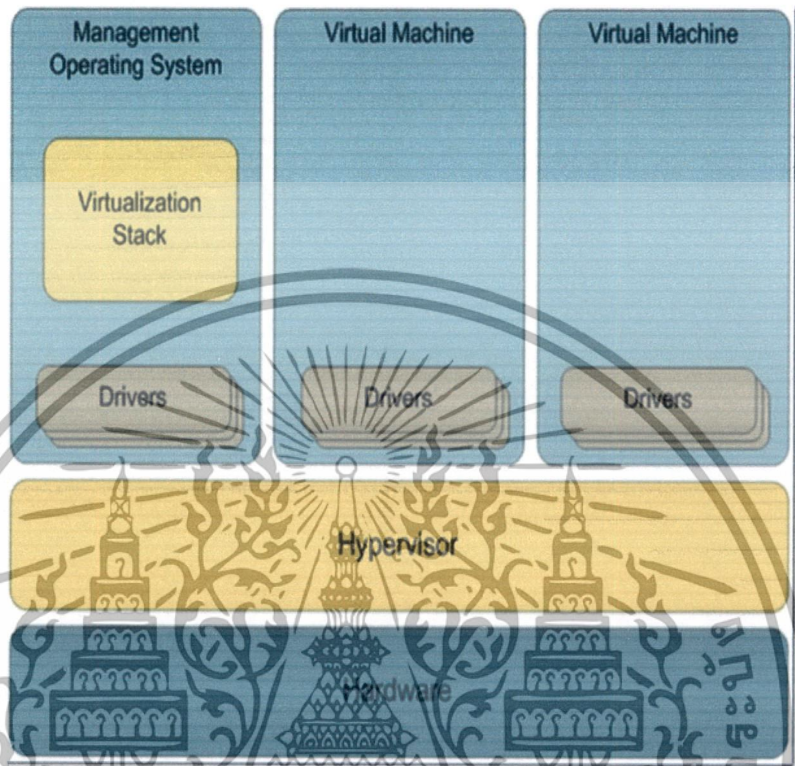
2.9 โครงสร้างพื้นฐานของ Hyper-V

รูปแบบโครงสร้างของ Hyper-V คือ Role หนึ่งบน Window Server ซึ่ง Microsoft ได้พัฒนาให้สามารถทำหน้าที่เป็น Hypervisor ในการบริหารทรัพยากรเสมือนได้โดยได้เพิ่มเข้าไปใน Window Version 2008 และปรับปรุงเรื่อยมาจนปัจจุบัน Window Version 2012 รวมถึง Window 8 และ 8.1 โครงสร้างการออกแบบ Hypervisor จะเป็นลักษณะ Native หรือ Bare - Metal โดยทั่วไปแล้วการใช้งานโดยพื้นฐาน Physical Hardware ต้องรองรับการทำ Server Virtualization ความต้องการพื้นฐานเป็น Intel VT, AMD Virtualization เป็นเทคโนโลยีเวอร์ชวลไลเซชั่นที่รันอยู่บนสถาปัตยกรรม x86

โครงสร้างสถาปัตยกรรมของ Hyper-V ตามรูปที่ 2.18 เป็น Layer ที่คั่นกลางระหว่าง Hardware และ Operating System ซึ่ง Hypervisor เองจะยินยอมให้มีหลายๆ ระบบปฏิบัติการสามารถทำงานอยู่บน Physical เดียวกันที่เวลาเดียวกันได้ ทำให้สามารถสร้างทรัพยากรหลายๆ Virtual Machine ให้ทำงานบน Physical Hardware เดียวกัน โดยมี Management Operating System ซึ่งภายในจะมี Virtualization Stack ถูกสร้างขึ้นมาเสมือนเป็น Virtual Machine ที่ทำหน้าที่เป็นตัวกลางบริหารจัดการและเป็นตัวกลางการประสานงานระหว่าง I/O Device ของ Driver ต่างๆ เปรียบการทำงานเป็นลักษณะ Root และ Child Partition นอกเหนือจากรูปองค์ประกอบที่ไม่ได้แสดงไว้ในรูปที่ 2.18 ในการสร้างทรัพยากรเสมือนก็จะเรียกผ่าน API ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

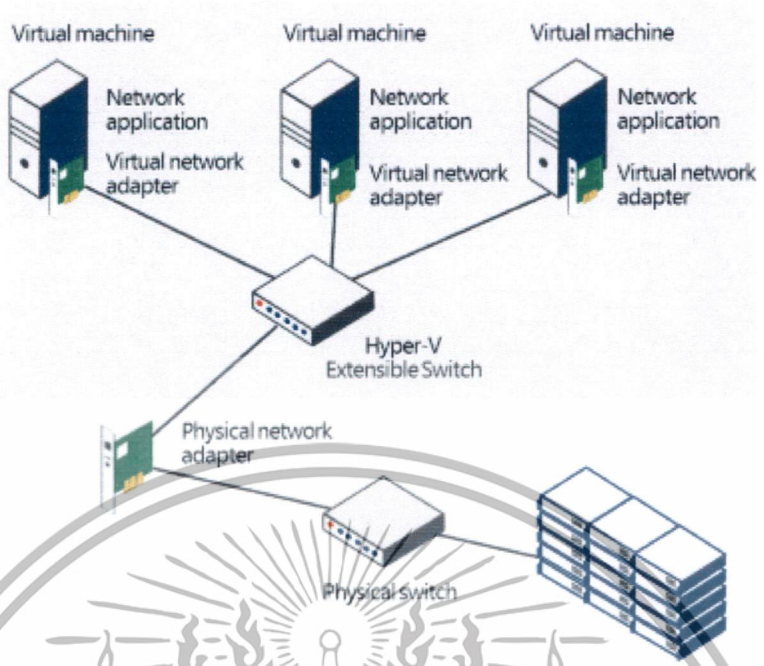
ชื่อว่า Hypercalls นอกจากนี้องค์ประกอบของ Hypervisor เองก็ถูกออกแบบมาให้มีขนาดของ Configuration ที่มีขนาดเล็กเพื่อลดความซ้ำซ้อน



รูปที่ 2.18 โครงสร้างสถาปัตยกรรมของ Hyper-V [17]

การเข้าบริหารจัดการทรัพยากรของ Hyper-V โดยปกติสามารถเข้าจัดการได้ 2 ช่องทางคือ ผ่านทางการใช้งานคำสั่ง Command Line (Power Shell) หรือผ่านทางการใช้งาน Hyper-V Management Console โดยลักษณะการใช้งานอย่างง่ายโดยทั่วไปจะมีตัวกลางที่เข้าบริหารจัดการช่วยอำนวยความสะดวกให้ผู้ใช้งานได้ง่ายขึ้นเรียกว่า SCVMM (System Center Virtual Machine Manager)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 รูปแบบการเชื่อมต่อเน็ตเวิร์กของไฮเปอร์วี [17]

ในส่วนของการเชื่อมต่อเน็ตเวิร์กเพื่อเข้าใช้งานทรัพยากรเสมือนของ Hyper-V ตามรูปที่ 2.19 Hyper-V Extensible Switch เป็นลักษณะสวิตซ์เน็ตเวิร์กเสมือนที่ทำงานใน Layer 2 [15] โดยจะแบ่งส่วนของเน็ตเวิร์กออกเป็น 3 ส่วนคือส่วนของ External Network คือส่วนที่ยินยอมให้ Traffic ของ Virtual Machine มีการติดต่อสื่อสารกับภายนอกได้โดย Traffic จะวิ่งผ่าน Physical Network Adapter ที่เชื่อมต่อกับ Physical Switch เพื่อติดต่อกับภายนอกหรือเป็นส่วนที่ให้ผู้ใช้งานสามารถเข้าถึงทรัพยากรได้ผ่านทางอินเตอร์เน็ต ส่วนของ Internal Network เป็นการเชื่อมต่อระหว่าง Virtual Machine และ Host เท่านั้น และส่วนสุดท้ายคือส่วนของ Private Network คือส่วนเน็ตเวิร์กที่ให้แต่ละ Virtual Machine สามารถติดต่อกันภายในได้ โดยไม่มีการติดต่อกับภายนอก ในการใช้งานเมื่อมีการสร้าง Virtual Machine ขึ้นมาใหม่จะต้องมีการสร้างส่วนของ Virtual Network Adapter อย่างน้อย 1 Interface เพื่อให้ Virtual Machine สามารถรับส่งข้อมูลบนเครือข่ายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

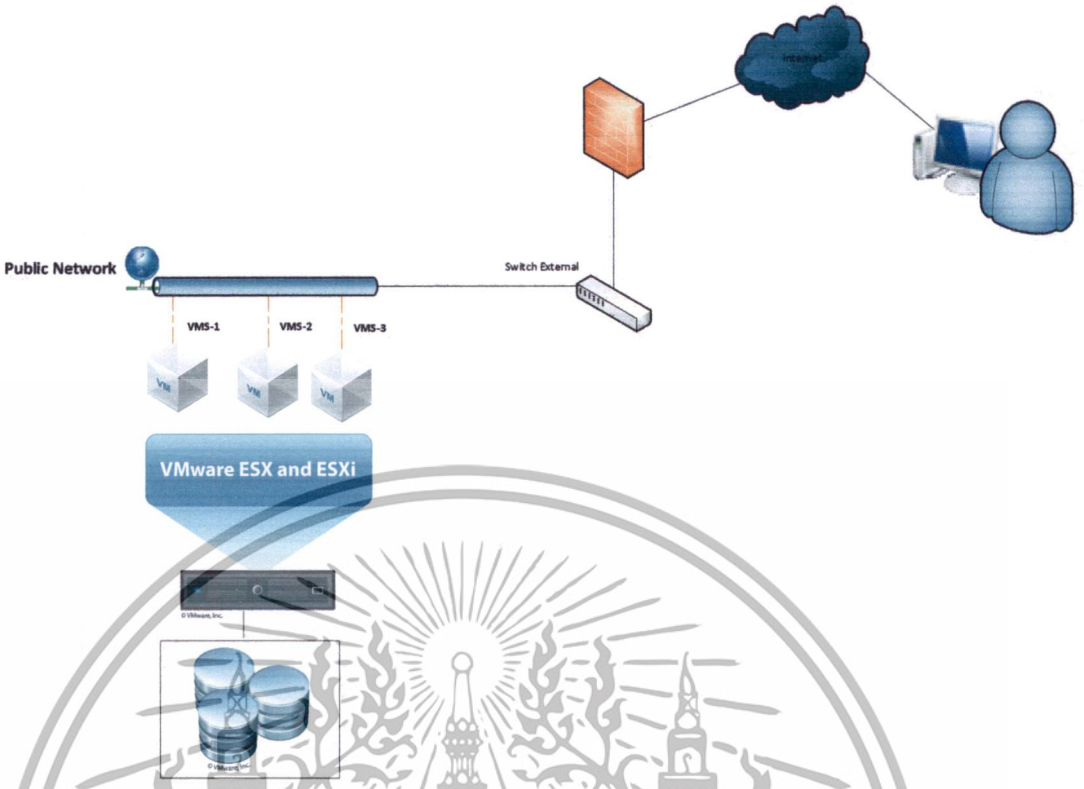
ศึกษาระบบปัจจุบันและจัดเตรียมทรัพยากรสำหรับทดสอบระบบ

ในการศึกษาค้นคว้าในครั้งนี้ได้ศึกษาระบบงานของผู้ใช้บริการรายหนึ่งที่กำลังจะย้ายระบบจากเดิมที่เป็น On-Premise มาขึ้นบนระบบ Cloud ของผู้ให้บริการ ดังนั้นจึงมีการศึกษาทั้งในส่วนของระบบงานเดิมที่มีการใช้งาน รวมถึงการจัดเตรียมโครงสร้างพื้นฐานสำหรับรองรับการย้ายและทดสอบประสิทธิภาพเมื่อมาอยู่บน Cloud ของผู้ให้บริการซึ่งมีรายละเอียดดังนี้

3.1 ระบบงานในปัจจุบัน

ระบบงานเดิมที่จะนำมาทดสอบบนทรัพยากรของผู้ให้บริการนั้น เป็นระบบที่เริ่มทดสอบโดยยังไม่มีการใช้งานของ User ภายในองค์กรเนื่องจากเป็นช่วง Migration ระบบ และปรับ Tuning Parameter ต่างๆ ซึ่งเดิมระบบมีการ Run อยู่บน Physical Hardware ด้วยข้อจำกัดของระบบงานเดิม ในเรื่องของการขยายทรัพยากรเพื่อให้รองรับระบบงานไม่สามารถทำได้ รวมถึงการบริหารจัดการค่อนข้างยุ่งยาก ค่าใช้จ่ายในการบำรุงรักษาสูง ทางผู้ใช้บริการเล็งเห็นถึงข้อดีของเทคโนโลยี Cloud Computing ที่ผู้ให้บริการให้บริการอยู่ในปัจจุบัน โดยทำการ Migrate System ขึ้นมาเพื่อทำการทดสอบระบบงานบนสถาปัตยกรรมที่เป็น โครงสร้าง x86 ซึ่งระบบที่จะนำมาทดสอบในช่วงแรก จะนำระบบงานมาติดตั้งบน Hypervisor ของ VMware และ Hyper-V ซึ่งจำเป็นต้องใช้ Software VMware Converter สำหรับทำการ Convert ในลักษณะ Online ซึ่งจะไม่กระทบกับการใช้งานระบบเดิมด้วยการทำ P2V (Physical To Virtual Machine) ลักษณะของการทำ P2V จะทำการ Clone Server และสร้าง Image ใหม่ที่ Host ปลายทางหรือสามารถ Clone และสร้าง Image ที่ Local แล้วทำการ Import ขึ้นระบบ Cloud ของผู้ให้บริการได้เช่นเดียวกัน ส่วนในการ Convert Virtual Machine สำหรับไปรันบน OpenStack จะต้องใช้ Qemu-Img Convert ซึ่งเป็น Command ที่ใช้ในการ Convert .VMDK ของ VMware เป็น Qcow2 Image File จากรูปที่ 3.1 รูปแบบการใช้งานเมื่อ Resource มาอยู่ที่ผู้ให้บริการแล้ว ผู้ใช้บริการเองจะสามารถเข้าถึงทรัพยากรได้ผ่านทาง Internet เป็นหลัก แต่การใช้งานอาจจะไม่เสถียรขึ้นอยู่กับ Bandwidth Internet ของแต่ละองค์กร ที่มีการใช้งานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ภาพรวมลักษณะการเชื่อมต่อเพื่อเข้าใช้งานระบบบน VMware

จากโครงสร้างดังรูปที่ 3.1 การเชื่อมต่อเพื่อเข้าใช้งานระบบบน VMware นั้น หลังจาก Import Image ขึ้นมาอยู่บนระบบ Cloud ได้ทำการแบ่ง Virtual Machine ที่จะทำการทดสอบออกเป็น 3 VM ประกอบด้วย VMS-1, VMS-2, VMS-3 ซึ่งรายละเอียดขนาดของ Virtual Machine อ้างอิงตามตารางที่ 3.1 ตามที่ได้กล่าวไปในบทที่ 2 หัวข้อ 2.7 VMware จะมี VMware ESX Hypervisor ทำหน้าที่เป็นตัวกลางคอยบริหารจัดการให้สามารถสร้าง VM ใช้งานได้บน Physical Hardware เดียว นอกจากนี้ได้ทำการสร้าง Volume ชนิด NFS ใช้สำหรับจัดเก็บ Image และ Data ของทั้ง 3 VM ในการเชื่อมต่อเข้ากับ Public Network บน VMware จะมีการสร้าง Virtual Switch ซึ่ง Virtual Switch จะถูก Map เข้ากับ Switch External เพื่อให้ผู้ใช้งานสามารถเข้าใช้งาน Virtual Machine ได้ผ่านทางอินเตอร์เน็ต ซึ่ง Traffic ทั้งหมดจะวิ่งผ่าน Firewall ที่เป็น Infrastructure กลางของผู้ให้บริการซึ่งจะอนุญาตให้เฉพาะ Source IP ขององค์กรผู้ใช้บริการเท่านั้นที่สามารถเข้าใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 สร้างสภาพแวดล้อมสำหรับทดสอบระบบ

โครงสร้างพื้นฐานของ Hypervisor ที่จะใช้ในการทดสอบมีองค์ประกอบของแต่ละแบบค่อนข้างแตกต่างกัน ดังนั้นผู้ศึกษาจำเป็นต้องปรับปรุงโครงสร้างให้มีรูปแบบที่ใกล้เคียงกันก่อนทำการทดสอบระบบ โดยมีการจัดเตรียมระบบสำหรับทดสอบดังนี้

3.2.1 VMware

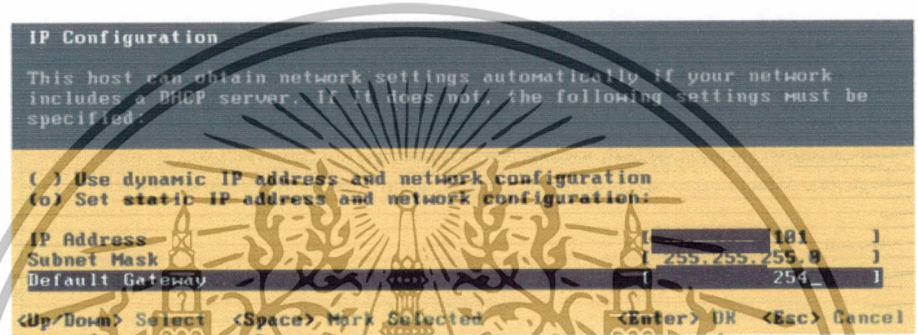
จากภาพรวมของระบบและลักษณะการใช้งานทรัพยากรเสมือนบนโครงสร้างของระบบ VMware สำหรับการทดสอบในครั้งนี้จะทำการติดตั้ง VMM ซึ่งเป็นตัวกลางที่ใช้บริหารจัดการอยู่บน Node (ESX Server) ที่อยู่บน Physical Hardware เดียวกัน ดังแสดงไว้ในรูปที่ 3.2 จากนั้นทำการสร้างทรัพยากรเสมือนซึ่งประกอบด้วย VMS-1, VMS-2, VMS-3 อ้างอิงขนาดของ VM ตามตารางที่ 3.1 โดยทำการอัพโหลดและติดตั้ง Image File .OVA เพื่อให้รันอยู่บน ESX Server ซึ่งการเข้าบริหารจัดการจะใช้ตัว System Center หรือที่เรียกว่า VMware vSphere Client สำหรับ Remote เข้าไปบริหารจัดการทรัพยากรเสมือน

รูปที่ 3.2 โครงสร้างการเข้าถึงทรัพยากรบน VMware

จากเดิมที่มีการ Convert Image มาใช้งานบนระบบ Cloud เรียบร้อยแล้วนั้น ในการทดสอบได้ทำการติดตั้ง Hypervisor ลงบน Physical Server เดียว นอกจากนี้ยังทำการแยกชุด Environment ในการทดสอบออกจากระบบที่มีการใช้งานในลักษณะแชร์ทรัพยากรร่วมกับองค์กรอื่น และจะทำการย้ายไปอยู่ใน Environment Production เมื่อได้ผลการ

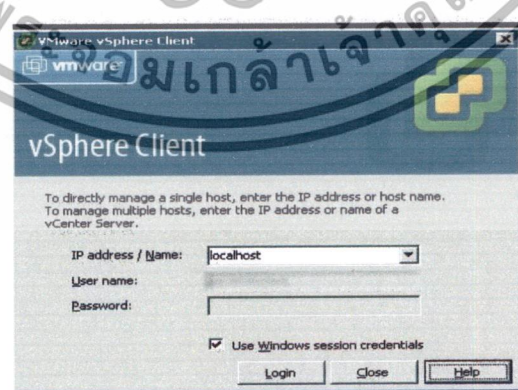
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบในส่วนของประสิทธิภาพเรียบร้อยแล้ว ซึ่ง Specification ของ Physical Hardware ที่ใช้ในการทดสอบครั้งนี้ได้แสดงรายละเอียดไว้ในบทที่ 3 หัวข้อ 3.3 โดยในส่วนของ VMware Hypervisor จะทำการติดตั้ง ESX5.1 ซึ่งทำหน้าที่เสมือนเป็นระบบปฏิบัติการลงบน Physical Server ซึ่งส่วนหนึ่งของขั้นตอนการติดตั้ง ในรูปที่ 3.3 จำเป็นต้องมีการกำหนดรายละเอียด IP ของ Server โดยมีการกำหนด IP Address, Subnet Mask, Default Gateway ที่จะใช้งาน เมื่อติดตั้งเสร็จแล้วจะสามารถเข้าใช้งานได้ผ่านทาง IP Address ที่ได้กำหนดไว้



รูปที่ 3.3 IP Configure VMware

หลังจากนั้นทำการติดตั้ง vCenter Server, Component System และติดตั้ง Vsphere Client ซึ่งใช้ในการรีโมทเข้าไปบริหารจัดการทรัพยากรและทำการสร้าง Virtual Machine ได้โดยในการใช้งานดังรูปที่ 3.4 ทำการระบุ Localhost หรือ IP Address ของ ESX Server และ Authen User/password ที่ได้กำหนดไว้

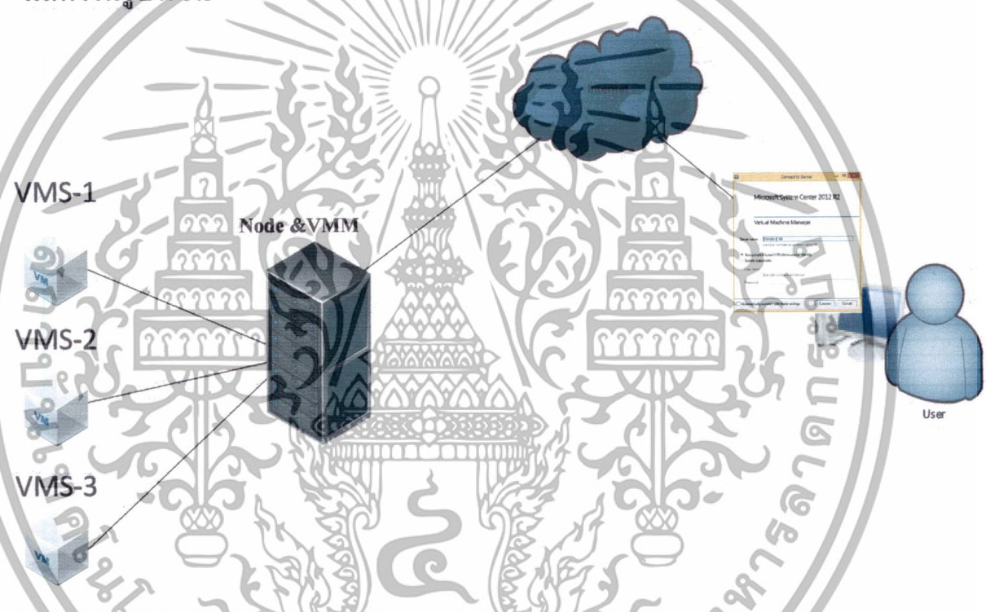


รูปที่ 3.4 VMware Vsphere Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Hyper-V

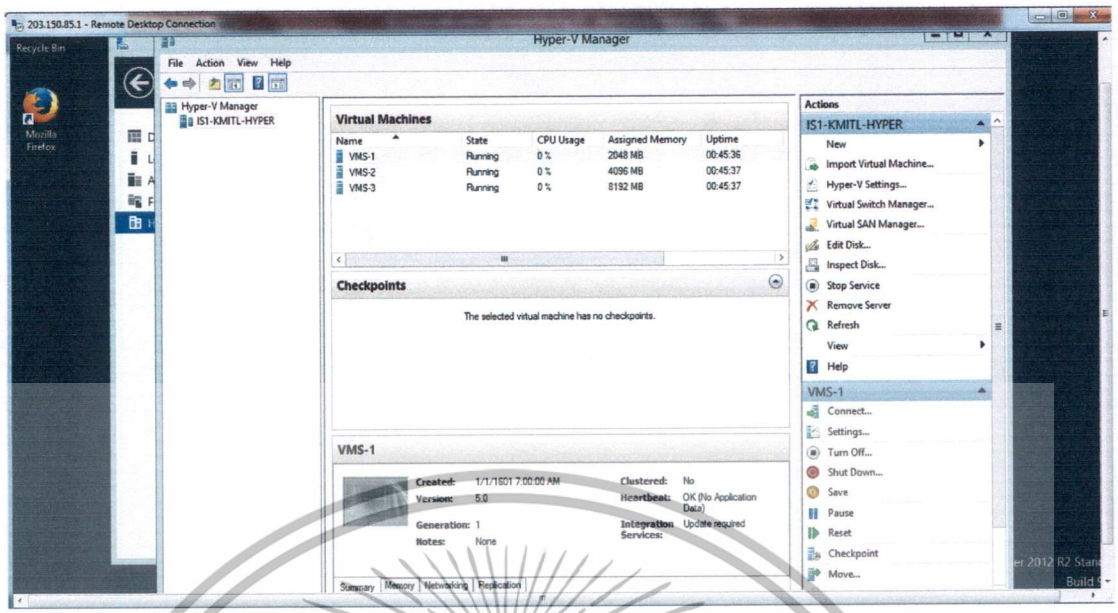
จากภาพรวมของระบบและลักษณะการใช้งานของระบบบน Hyper-V การติดตั้ง Hyper-V Hypervisor เพียงแค่เพิ่ม Role บน Window Server 2012 ก็จะสามารถใช้งานได้ จากรูปที่ 3.5 การเข้าบริหารจัดการทรัพยากรเสมือนซึ่งประกอบด้วย VMS-1, VMS-2, VMS-3 อ้างอิงขนาดของ VM ตามตารางที่ 3.1 บน Hyper-V จะมี VMM ซึ่งเป็นตัวกลางในการสร้างทรัพยากรเสมือนที่อยู่บน Node (ESX Server) ซึ่งในที่นี้ได้ทำการ Convert Image โดย Software VMware Converter เพื่อทำการ Convert Image จาก .VMDK ที่ใช้งานบน VMware แปลงเป็น .VHD เพื่อใช้งานบน Hyper-V จากนั้นก็ทำการ Import Image ขึ้นบนระบบ Cloud ในการใช้งานผู้ใช้งานสามารถเข้าบริหารจัดการได้ทาง System Center ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 โครงสร้างการเข้าถึงทรัพยากรบน Hyper-V

ในการเข้าใช้งานหรือบริหารจัดการทรัพยากรเสมือนจะใช้ Hyper-V manager ดังรูปที่ 3.6 เมื่อเข้าใช้งานสามารถบริหารจัดการ VMS-1, VMS-2, VMS-3 รวมถึงสามารถสร้างส่วนของ Network Interface, Storage และปรับเปลี่ยน Parameter Configuration อื่นๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Hyper-V Management

3.2.3 OpenStack

จากภาพรวมของระบบและลักษณะการใช้งานโครงสร้างของระบบ OpenStack จะมีรูปแบบที่ยังไม่ใกล้เคียงกับ VMware และ Hyper-V จำเป็นต้องทำการปรับเปลี่ยนระบบให้มีรูปแบบที่ใกล้เคียงกัน ซึ่งการเข้าบริหารจัดการทรัพยากรเสมือนจะเข้าผ่านตัวกลางคือ Nova ในการจัดการ Compute, Storage และ Network การจัดเตรียมระบบสำหรับทดสอบจะดำเนินการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.7 การติดตั้ง DevStack แบบ All-In-One

จากรูปที่ 3.7 เป็น Flow ขั้นตอนการติดตั้ง DevStack แบบ All-In-One ซึ่งจะติดตั้งลงบน Physical Hardware เดียว เริ่มต้นทำการติดตั้งระบบปฏิบัติการ Ubuntu Version 14.04 ลงบน Physical Hardware จากนั้นทำการติดตั้ง DevStack โดย Download Devstack ด้วยคำสั่ง “git clone https://git.openstack.org/openstack-dev/devstack” ทำการแก้ไข Path File : local.conf เพื่อกำหนด Permission, Credential และ Network IP ที่ใช้สำหรับ Virtual Guest จากนั้นทำการพิมพ์คำสั่งเพื่อ Run DevStack ด้วย Command “./stack.sh” เมื่อติดตั้งเสร็จสามารถเข้าใช้งาน OpenStack ผ่านทางหน้า Dashboard เพื่อสร้าง Virtual Machine ที่ต้องการทดสอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ฮาร์ดแวร์และรายละเอียดเครื่องเซิฟเวอร์เสมือนสำหรับทดสอบ

การศึกษาและทดสอบประสิทธิภาพในครั้งนี้ได้จัดเตรียม Physical Hardware สำหรับทดสอบระบบงานโดยมีรายละเอียดดังนี้ Physical Specification: CiscoB200M3 2 Processor Intel® Xeon® E5-2600 and E5-2600 v2, 256 GB Ram [16], NetApp series 8040, Raid-5 storage (NAS) หลังจากจัดเตรียม Physical Hardware และติดตั้ง Hypervisor เรียบร้อยแล้ว ได้จัดเตรียม Virtual Machine สำหรับทดสอบตามตารางที่ 3.1 ซึ่งประกอบด้วย VMS-1, VMS-2, VMS-3 บน VMware, Hyper-V, OpenStack Hypervisor โดย Virtual Machine ที่ใช้ในการทดสอบทั้ง 3 แบบมีการติดตั้งระบบปฏิบัติการเป็น Window Version 2008R2

ตารางที่ 3.1 ตารางแสดงรายละเอียดทรัพยากรเสมือนที่ใช้สำหรับทดสอบ

VM name	Spec	Operating System	CPU (VCPU)	Memory (GB)	Disk (GB)	Hypervisor
VMS-1	1vcpu,2 GB ram, 50 GB disk	Window2008R2	1	2	50	VMware
VMS-2	2vcpu,4 GB ram, 100 GB disk	Window2008R2	2	4	100	VMware
VMS-3	4vcpu,8 GB ram, 150 GB disk	Window2008R2	4	8	150	VMware
VMS-1	1vcpu,2 GB ram, 50 GB disk	Window2008R2	1	2	50	Hyper-V
VMS-2	2vcpu,4 GB ram, 100 GB disk	Window2008R2	2	4	100	Hyper-V
VMS-3	4vcpu,8 GB ram, 150 GB disk	Window2008R2	4	8	150	Hyper-V
VMS-1	1vcpu,2 GB ram, 50 GB disk	Window2008R2	1	2	50	OpenStack
VMS-2	2vcpu,4 GB ram, 100 GB disk	Window2008R2	2	4	100	OpenStack
VMS-3	4vcpu,8 GB ram, 150 GB disk	Window2008R2	4	8	150	OpenStack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 เครื่องมือสำหรับทดสอบประสิทธิภาพ

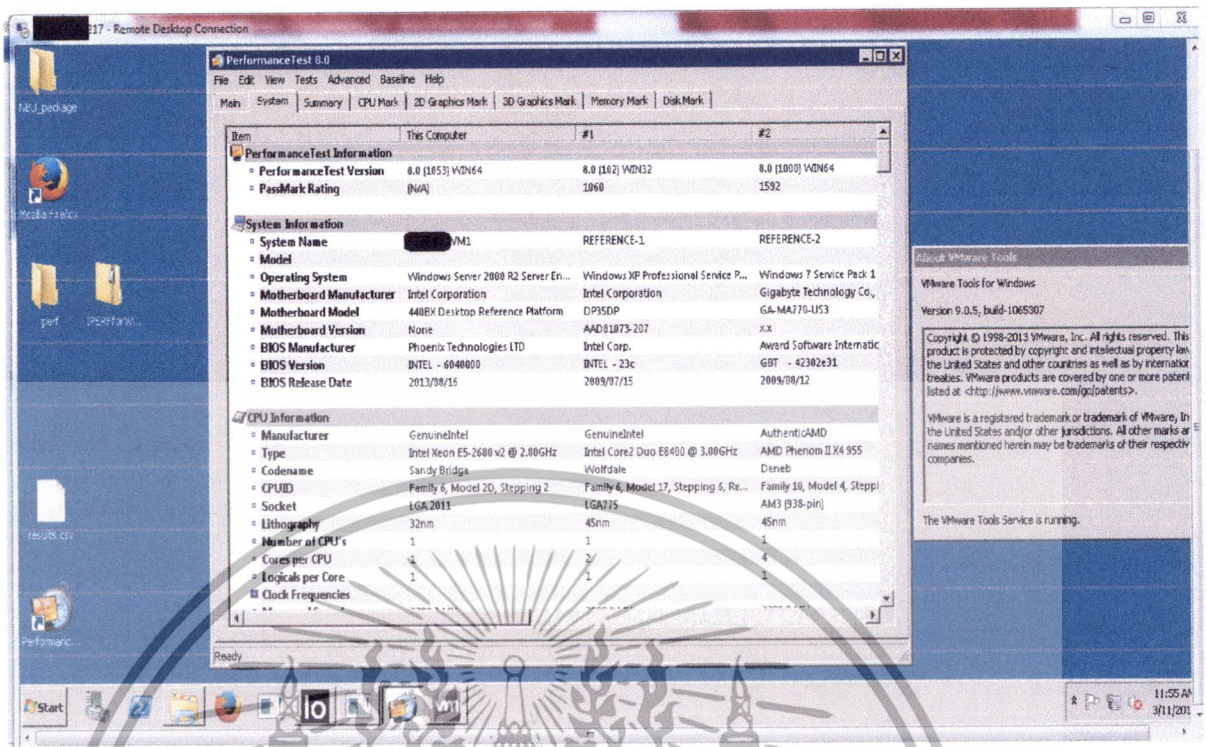
ระบบงานของผู้ใช้บริการเมื่อมีการนำระบบมาอยู่บน Virtualization แล้วจำเป็นอย่างยิ่งที่จะต้องทำการทดสอบประสิทธิภาพของระบบซึ่งในการศึกษาครั้งนี้จะทำการวัดประสิทธิภาพส่วนของการประมวลผลและส่วนของหน่วยความจำด้วย Passmark Tool และทำการทดสอบการอ่านเขียนข้อมูลด้วย IOmeter Tool สำหรับทดสอบประสิทธิภาพในการอ่านเขียนข้อมูลของทรัพยากรเสมือนซึ่งปกติการอ่านเขียนข้อมูลจะมีรูปแบบทั้งการอ่านเขียนแบบต่อเนื่องและการอ่านเขียนแบบสุ่ม และสุดท้ายคือการทดสอบการรับส่งข้อมูลโดยใช้ IPerf3 Tool เป็น Software เสรีโดยการทำงานจะจำลองการเชื่อมต่อในลักษณะ Client-Server ซึ่งหลักการทำงานจะทำการ Simulate Package เพื่อส่งผ่านเครือข่ายไปยังเครื่องปลายทางที่มีการเชื่อมโยงกันภายในเครือข่ายทั้งในรูปแบบ TCP และ UDP เพื่อตรวจสอบฝั่งผู้รับจะสามารถรับข้อมูลได้ในปริมาณเท่าไหร่ รวมถึงจำนวนปริมาณข้อมูลสูญหายระหว่างการรับส่ง โดยมีรายละเอียดของ Software ที่ใช้ในการทดสอบมีดังนี้

3.4.1 Passmark Tool

ในการทำงานของระบบงานในช่วงก่อนการติดตั้งใช้งานจริงจำเป็นต้องใช้เครื่องมือเข้ามาช่วยในการวัดผลประสิทธิภาพความพร้อมในการทำงานและการปรับ Tuning เพื่อให้รองรับกับระบบงาน ซึ่ง Passmark Tool เป็นเครื่องมือหนึ่งที่เข้ามาช่วยทดสอบประสิทธิภาพโดยทำการวัดผลเพื่อนำไปเปรียบเทียบและปรับปรุงระบบ ดังนั้นในการการกำหนด Parameter ของ Passmark Tool เพื่อทำการทดสอบในงานวิจัยครั้งนี้ ได้มีการกำหนดทรัพยากรที่จะทำการทดสอบบนระบบปฏิบัติการ Window Version 2008R2 อ้างอิงข้อมูลทรัพยากรที่ใช้ทดสอบตามตารางที่ 3.1 ซึ่งเป็นระบบงานเดิมของลูกค้า โดยจะทำการวัดประสิทธิภาพทางด้านการประมวลผลของ CPU ความสามารถของหน่วยความจำโดยกำหนดขอบเขตของการทดสอบ Parameter ดังนี้

ในส่วนของการทดสอบประสิทธิภาพ CPU, Memory ดำเนินการติดตั้งเครื่องมือที่ใช้ทดสอบลงใน Virtual Machine ตามรูปที่ 3.8 โดยมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Specification Physical Server

หลังจากทำการติดตั้ง Passmask Tool แล้วตามรูปที่ 3.8 จะแสดงรายละเอียดของข้อมูลพื้นฐานของ Virtual Machine ที่จะทำการทดสอบ ซึ่งประกอบด้วยรายละเอียดของ Processor, Memory, Disk, Driver ต่างๆ เป็นต้น ซึ่ง Virtual Machine ที่ได้ทดสอบมีการสร้างบน Physical Hardware ที่มี CPU Processor Architecture (x86) ซึ่งเป็นโครงสร้างสถาปัตยกรรมที่มีการออกแบบให้สามารถทำงานได้หลายๆ คำสั่งที่เกี่ยวข้องกับการประมวล ผู้ศึกษาจึงได้กำหนด Parameter ที่ใช้งานการทดสอบทั้งบน VMware, Hyper-V, OpenStack Hypervisor ตามตารางที่ 3.2

ตามตารางที่ 3.2 มีการกำหนด Parameter ที่จะทำการทดสอบทั้งหมด 5 Parameter ซึ่งประกอบด้วย การประมวลผลคำสั่งที่เกี่ยวข้องกับการคำนวณ (Integer Math), การประมวลผลคำสั่งที่เกี่ยวข้องกับเลขทศนิยม (Floating Point Math), การประมวลผลคำสั่งที่เกี่ยวข้องกับเลขเฉพาะ (Prime Number) , การประมวลผลที่เกี่ยวข้องกับการเรียงลำดับข้อมูล (Sorting) และสุดท้ายคือการประมวลผลคำสั่งแบบ Single Thread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 CPU Parameter

Parameter	Description
Integer Math (millions of operations per Seconds)	ความสามารถในการคำนวณ (+, -, *, และ /)
Floating Point Math (millions of operations per seconds)	ความสามารถในการคำนวนเกี่ยวกับทศนิยม (.01, .0001, .00001)
Prime Number (millions of prime per seconds)	ความสามารถในการคำนวนเกี่ยวกับเลขเฉพาะ (3, 5, 7, 11, ..)
Sorting (millions string per second)	ความสามารถในการเรียงลำดับข้อมูล
Single Thread (million of operation per second)	ความสามารถในการประมวลผลแบบ Single Thread

ส่วนประสิทธิภาพของการอ่านเขียนหน่วยจำเป็นส่วนหนึ่งที่สำคัญสำหรับการทำงานของระบบ เมื่อ CPU มีการประมวลผล จะมีการดึงข้อมูลและชุดคำสั่งจากหน่วยความจำไปใช้งาน ดังนั้นในส่วนของหน่วยความจำจึงมีการเพิ่ม Cash Memory เข้ามาเก็บชุดคำสั่งที่ใช้งานบ่อยๆ CPU เมื่อต้องการประมวลผลก็จะมาดึงจากส่วนนี้ ทำให้การประมวลได้เร็วขึ้น เพราะหน่วยประมวลไม่ต้องเข้าถึงหน่วยความจำหลัก ดังนั้นจึงกำหนดให้มีการทดสอบ Parameter เพื่อเทียบประสิทธิภาพการทำงานของทั้ง 3 Hypervisor ดังตารางที่ 3.3 โดยกำหนดให้มีการทดสอบทั้งหมด 4 Parameter ประกอบด้วยความสามารถในการอ่านข้อมูลจากหน่วยความจำแคช, ความสามารถในการอ่านข้อมูลโดยไม่ใช้หน่วยความจำแคช, ความสามารถในการเขียนข้อมูลหน่วยความจำ และสุดท้ายช่วงเวลาหน่วงในการเข้าถึงหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

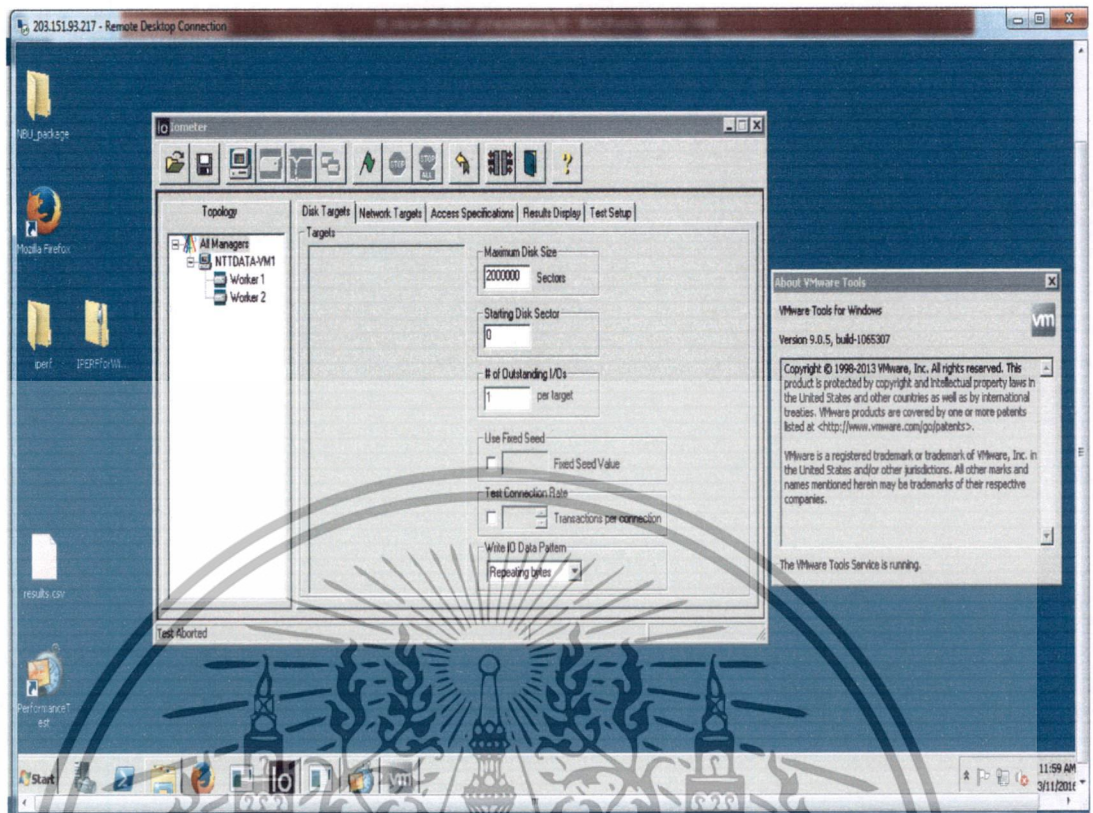
ตารางที่ 3.3 Memory Parameter

Parameter	Description
Read Cached (Mbytes Transferred per second)	ความสามารถในการอ่านข้อมูลจากหน่วยความจำแคช
Read UnCached (Mbytes Transferred per second)	ความสามารถในการอ่านข้อมูลโดยไม่ใช้หน่วยความ จำแคช
Write (Mbytes Transferred per second)	ความสามารถในการเขียนข้อมูลหน่วยความจำ
Latency (Nanoseconds (lower is better))	ช่วงเวลาหน่วงในการเข้าถึงหน่วยความจำ (ค่าน้อยยิ่งดี)

3.4.2 IOmeter Tool

การทดสอบประสิทธิภาพการอ่านเขียนข้อมูลดิสก์ ตามการทดลองมีการจัดเตรียมทรัพยากรใช้งาน Hard disk แบบ SAS ซึ่ง Server ที่มีการติดตั้ง Virtualization ของทั้ง 3 Hypervisor มีการเชื่อมต่อแบบ NAS โดยทำการติดตั้ง IOmeter Tool ซึ่งเป็น .exe file ลงบน Virtaul Machine เมื่อเปิดโปรแกรมใช้งาน ดังรูปที่ 3.9 ด้านซ้ายมือแสดงรายละเอียด Topology ประกอบด้วยชื่อ VM และจำนวน Workload ที่ต้องการทดสอบ และด้านขวามือจะแสดงรายละเอียดในการกำหนด Parameter สำหรับทดสอบ ซึ่งประกอบด้วยรายละเอียดของ Sector, การกำหนดการอ่านเขียนแบบสุ่มหรือลำดับ ซึ่งรูปที่ 3.9 เป็นตัวอย่างของการกำหนดค่า Parameter บน VMware โดย Parameter ที่ได้ทำการทดสอบอ้างอิงตามตารางที่ 3.4

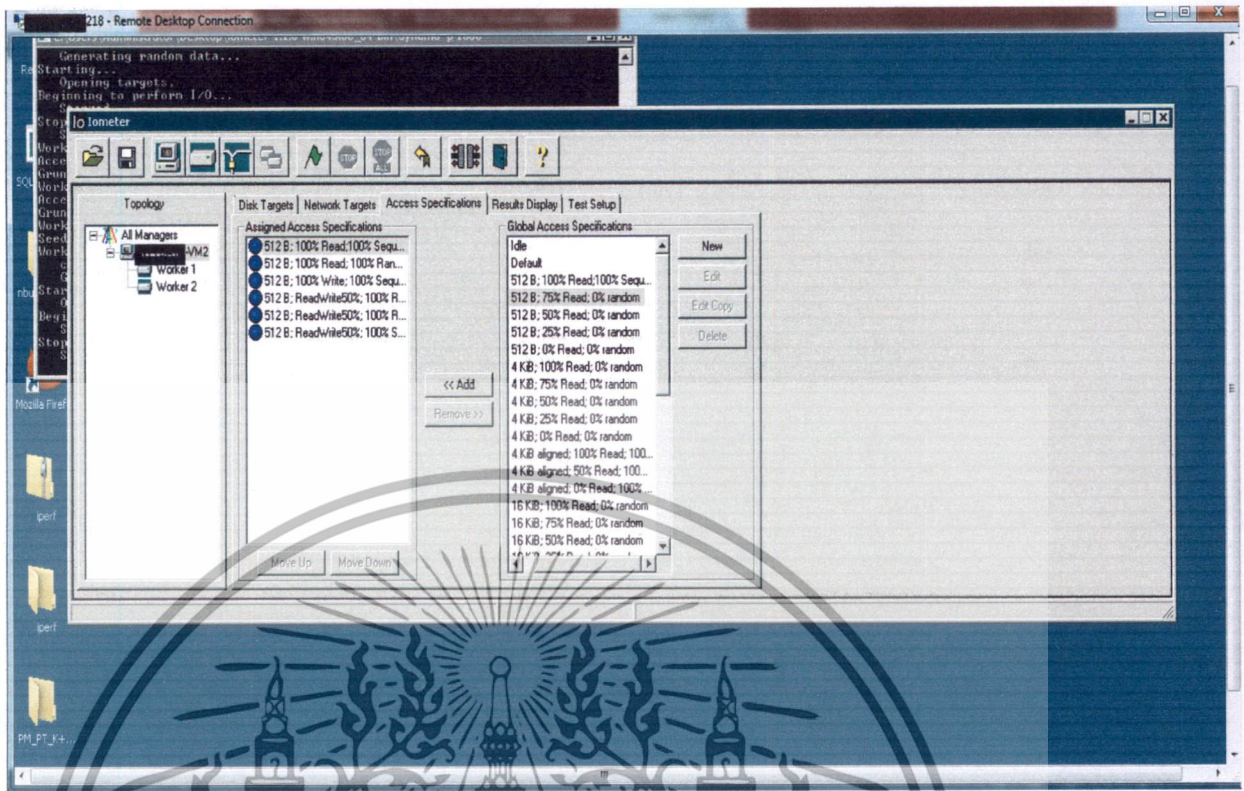
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 Parameter Configuration บน IOmeter

ในการศึกษาครั้งนี้ได้มีการกำหนด Parameter ในการทดสอบประสิทธิภาพของ Virtual Disk ที่มีการใช้งานตามรูปที่ 3.10 โดยได้ทำการกำหนดจำนวน Worker (Thread) ในการทำงานบน Disk ที่ต้องการทดสอบจำนวน 2 Workload และกำหนด Parameter ที่จะทดสอบดังตารางที่ 3.4 การทดสอบจะทำการทดสอบทั้งการอ่านเขียนแบบสุ่มและแบบลำดับกับขนาดของข้อมูล Logical Block Size ขนาด 512 KB หรือเทียบเท่ากับ 8 Block ของ 1 Physical Block Size (4 KByte)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ตั้งค่า parameter บน IOmeter

จากข้อมูลการกำหนดค่า Parameter เพื่อใช้สำหรับทดสอบประสิทธิภาพในการอ่านเขียนทั้งแบบสุ่มและแบบลำดับที่ได้กล่าวไว้ข้างต้นสามารถอธิบายรายละเอียดได้ตามตารางที่ 3.4

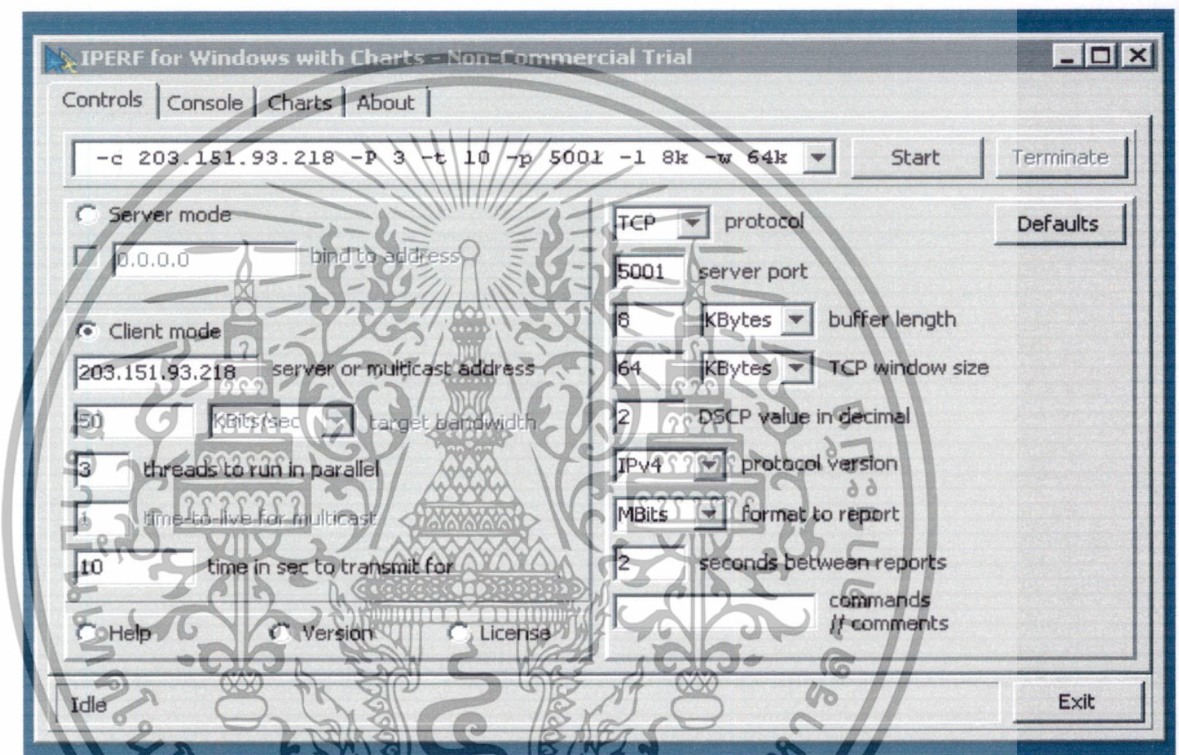
ตารางที่ 3.4 Disk Parameter

Parameter	Description
512 KB 100% Read, 100 %Sequential	อ่านข้อมูลขนาด 512 KB แบบลำดับ
512 KB 100% Read, 100% Random	อ่านข้อมูลขนาด 512 KB แบบสุ่ม
512 KB 100% Read, 100 %Sequential	เขียนข้อมูลขนาด 512 KB แบบลำดับ
512 KB 100% Write, 100% Random	เขียนข้อมูลขนาด 512 KB แบบสุ่ม
512 KB Read&Write, 50% Sequential	อ่านข้อมูลขนาด 512 KB 50% และเขียนข้อมูลขนาด 512 KB 50% แบบลำดับ
512 KB Read&Write, 50% Random	อ่านข้อมูลขนาด 512 KB 50% และเขียนข้อมูลขนาด 512 KB 50% แบบสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 IPerf3 Tool

ในการทดสอบความสามารถในการรับส่งข้อมูลภายในเครือข่ายของทั้ง 3 Hypervisor การทดลองได้ใช้ IPerf3 Tool หลักการทำงาน Software จะทำการ Simulate Package จากนั้นจะทำการส่งข้อมูลไปยังฝั่งรับทั้งแบบ TCP และ UDP ในลักษณะของการทำงานที่เรียกว่า Client - Server ดังรูปที่ 3.11



รูปที่ 3.11 Configuration Client-Server บน IPerf3

โดยในการกำหนดค่า Parameter ที่ใช้ในการทดสอบจะต้องระบุ IP ของ Server, Client รวมถึงรายละเอียดของ Port และขนาดข้อมูลที่ต้องการส่ง เช่น Buffer Length, TCP Window Size, Protocol Version ซึ่งได้อธิบายรายละเอียดของค่า Parameter ไว้ตามตารางที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 Iperf3 Parameter

Command line	Description
p, --port n	กำหนด Server Port และ Client Port ในการเชื่อมต่อโดยมี Default ที่ใช้งานคือ Port 5201
-f, --format [kmKM]	รูปแบบการแสดงผลจำนวน Bandwidth 'k' = Kbits/sec 'K' = KBytes/sec 'm' = Mbits/sec 'M' = MBytes/sec
-i, --interval n	การกำหนดช่วงเวลาในหน่วยวินาทีของช่วงระหว่าง Periodic Bandwidth , Jitter, Loss
-w size	การกำหนดขนาดของ TCP Window Size โดย Default ขนาด 64 Kbytes
-z	อนุญาตให้ใช้ Congestion Control Algorithm สำหรับการส่งแบบ TCP
-u	กำหนดการส่ง Package แบบ UDP
-p	กำหนดการส่ง Package แบบ TCP
-b, --bandwidth	กำหนดขนาดของแบนด์วิธที่ใช้ในการส่ง n bits/sec (Default 1 Mbit/sec for UDP, Unlimited for TCP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและวิเคราะห์ผลการทดลอง

จากบทที่ 3 ที่ได้กล่าวถึงรายละเอียดของทรัพยากรที่จะใช้ในการทดสอบ รวมถึงเครื่องมือที่จะใช้เป็นตัววัดประสิทธิภาพของหน่วยประมวลผล หน่วยความจำ การอ่านและเขียนข้อมูลดิสก์ รวมถึงการรับส่งข้อมูลภายในเครือข่าย ในบทที่ 4 นี้จะกล่าวถึงผลการทดสอบที่ได้จากการทดสอบประสิทธิภาพบน 3 Hypervisor ได้แก่ VMware, Hyper-V, OpenStack ดังนี้

4.1 การทดสอบประสิทธิภาพของการประมวลผลของ CPU

หลังจากทำการ Provision Virtual Machine และปรับขนาดของ VCPU ตามตารางที่ 4.1 โดยประกอบด้วย VMS-1 (1vcpu), VMS-2 (2vcpu) และ VMS-3 (4vcpu) จากนั้นกำหนดค่า Parameter ที่จะทำการทดสอบทั้งหมด 5 ค่าซึ่งได้อธิบายไว้ในตารางที่ 3.2 เพื่อทดสอบประสิทธิภาพของการประมวลผลของ CPU บน VMware , OpenStack, Hyper-V Hypervisor ซึ่งได้ผลการทดลองดังค่าที่แสดงไว้ในตารางที่ 4.1

ตารางที่ 4.1 ผลการทดสอบประสิทธิภาพของการประมวลผลของ CPU

Spec	Hypervisor	Integer math (millions of operations per seconds)	Floating point math (millions of operations per seconds)	Prime Number (millions of prime per seconds)	Sorting(million of string per second)	Single Threaded(Million of operation per second)
1vcpu	VMWare	1785	1631	9.4	1226	1618
2vcpu	VMWare	4054	2927	18.7	2445	1629
4vcpu	VMWare	8012	5701	39.4	4846	1607
1vcpu	Hyper-V	1092	636	4.75	584	836
2vcpu	Hyper-V	1896	1292	9.6	829	645
4vcpu	Hyper-V	4120	1900	15.6	2049	844
1vcpu	OpenStack	1364	1257	10.2	1194	1650
2vcpu	OpenStack	3586	3285	20.9	2299	1081
4vcpu	OpenStack	7266	6442	39.8	4653	1282

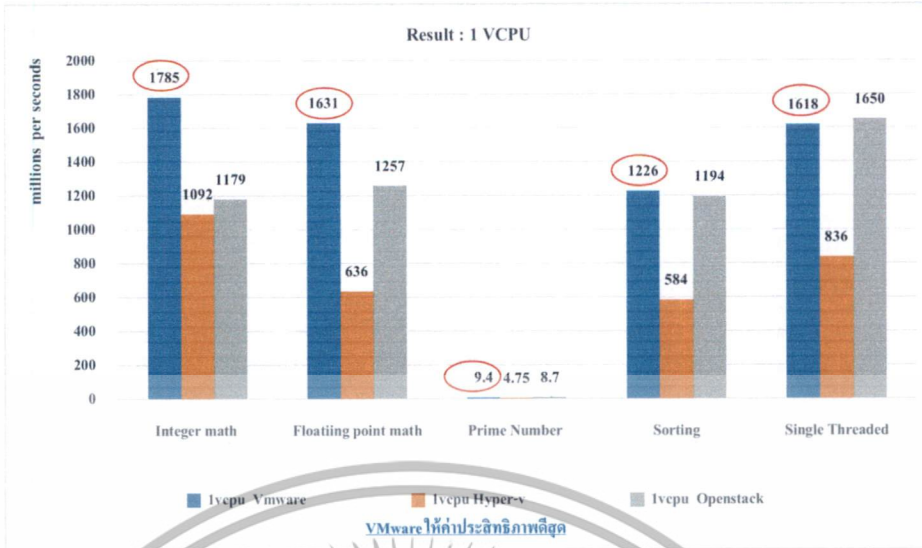
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถในการประมวลผลของ VCPU เทคโนโลยีของ Virtualization ถือเป็นอีกส่วนที่สำคัญไม่น้อยกว่า Memory, Disk และ Network ในมุมมองของ Infrastructure ของผู้ให้บริการ เมื่อผู้ใช้งานต้องการนำ Application, Software หรือระบบขึ้นมาติดตั้งใช้งาน ประสิทธิภาพการประมวลผลเป็นปัจจัยสำคัญ Physical Hardware เมื่อได้ทำการติดตั้ง Hypervisor แล้วทรัพยากรจะถูกแบ่งออกเป็น 2 Layer คือ Layer ของ Physical Hardware และ Logical Hardware ในอัตราส่วน 1:2 และมีการ Enable Hyper Thread เพื่อเพิ่มประสิทธิภาพให้ CPU สามารถประมวลผลหลายคำสั่งได้มากขึ้น ผลการทดลองที่ได้ตามตารางที่ 4.1 สามารถเปรียบเทียบประสิทธิภาพของแต่ละ Hypervisor ตามขนาดของ VCPU ได้ดังนี้

4.1.1 ความสามารถในการประมวลผลแบบ 1 VCPU

จากผลการทดสอบการประมวลผลของ 1 VCPU บน VMware, Hyper-V, OpenStack Hypervisor ดังแสดงในรูปที่ 4.1 ผลลัพธ์ที่ได้ทำการทดสอบการประมวลผลที่เกี่ยวข้องกับส่วนของ Integer Math, Floating Point Math, Prime Number, Sorting และ Single Threaded การประมวลผลบน VMware ให้ค่าออกมาสูงสุดตามที่ได้ทำสัญลักษณ์วงกลมสีแดงไว้ในรูปที่ 4.1 นั่นหมายความว่าขนาดของ VMS-1 ที่ 1 VCPU บน VMware Hypervisor สามารถประมวลผลได้เร็วกว่า OpenStack และ Hyper-V นอกจากนี้จากกราฟในรูปที่ 4.1 ผลการทดสอบทุกค่า Parameter ของ Hyper-V ให้ผลลัพธ์ที่ออกมาต่ำสุดเมื่อเทียบกับ VMware และ OpenStack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



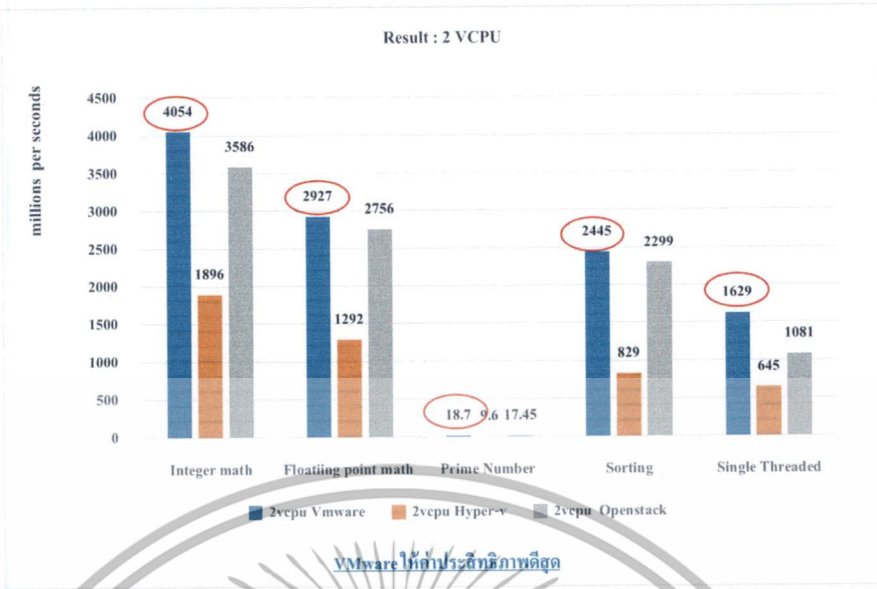
รูปที่ 4.1 กราฟเปรียบเทียบการประมวลผลแบบ 1 VCPU

จากผลลัพธ์ในรูปที่ 4.1 ผลลัพธ์ในส่วนของการประมวลที่เกี่ยวข้องกับ Integer บน VMware Hypersor ให้ค่าสูงสุดที่ 1785 millions per seconds, การประมวลผลที่เกี่ยวข้องกับ Floating Point Math ให้ค่าสูงสุดที่ 1631 millions per seconds, การประมวลผลที่เกี่ยวข้องกับ Prime Number ให้ค่าสูงสุดที่ 9.4 millions per seconds, การประมวลผลที่เกี่ยวข้องกับการเรียงลำดับข้อมูลให้ค่าสูงสุดที่ 1226 millions per seconds และการประมวลผลแบบ Single Threaded ให้ค่าสูงสุดที่ 1618 millions per seconds จากผลที่ได้ VMware มีความสามารถในการประมวลที่เร็วกว่า OpenStack และ Hyper-V ซึ่งเมื่อสังเกตุผลลัพธ์ที่ได้จากรูปที่ 4.1 ประสิทธิภาพในการประมวลของ Hyper-V Hypervisor ให้ค่าที่ต่ำสุด

4.1.2 ความสามารถในการประมวลผลแบบ 2 VCPU

จากผลการทดสอบการประมวลผลของ 2 VCPU บน VMware, Hyper-V, OpenStack Hypervisor ดังแสดงในรูปที่ 4.2 ผลลัพธ์ที่ได้ทำการทดสอบการประมวลผลผลที่เกี่ยวข้อง Integer Math, Floating Point Math, Prime Number, Sorting และ Single Threaded การประมวลผลของ VMS-2 ที่มีขนาด 2 VCPU บน VMware Hypervisor ยังคงให้ค่าผลลัพธ์ออกมาสูงสุดตามที่ได้ทำสัญลักษณ์วงกลมสีแดงไว้ในรูปที่ 4.2 นั่นหมายความว่าขนาดของ Virtual Machine ที่ 2 VCPU บน VMware Hypervisor สามารถประมวลผลได้เร็วกว่า OpenStack และ Hyper-V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 กราฟเปรียบเทียบการประมวลผลแบบ 2 VCPU

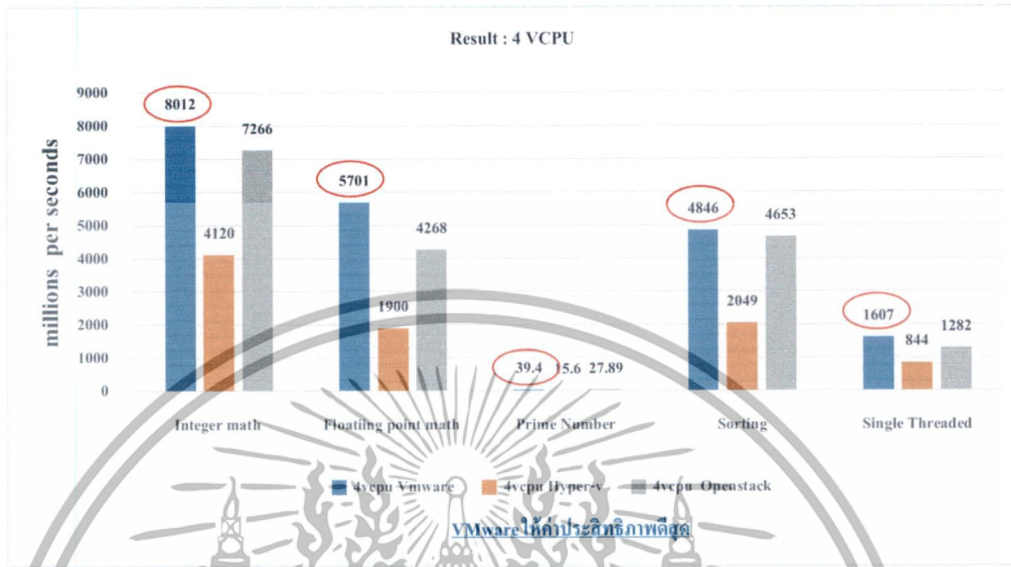
จากผลลัพธ์การทดสอบการประมวลผลของ 2 VCPU บน VMware, OpenStack และ Hyper-V ในรูปที่ 4.2 ผลลัพธ์ในส่วนของการประมวลที่เกี่ยวข้องกับ Integer บน VMware Hypervisor ให้ค่าสูงสุดที่ 4054 millions per seconds, การประมวลผลที่เกี่ยวข้องกับ Floating Point Math ให้ค่าสูงสุดที่ 2927 millions per seconds, การประมวลผลที่เกี่ยวข้องกับ Prime Number ให้ค่าสูงสุดที่ 18.7 millions per seconds, การประมวลผลที่เกี่ยวข้องกับการเรียงลำดับข้อมูลให้ค่าสูงสุดที่ 2445 millions per seconds และการประมวลผลแบบ Single Threaded ให้ค่าสูงสุดที่ 1629 millions per seconds เมื่อเทียบกับผลการทดลองของ VM ขนาด 1 VCPU ในหัวข้อ 4.1.1 ที่แสดงให้เห็นว่าการปรับขนาดของ VCPU ส่งผลให้การประมวลผลเกี่ยวกับชุดคำสั่งได้มากขึ้น นอกจากนี้ Hyper-V Hypervisor ยังคงให้ค่าออกมาต่ำสุดเมื่อเทียบกับ VMware และ OpenStack

4.1.3 ความสามารถในการประมวลผลแบบ 4 VCPU

จากผลการทดสอบประสิทธิภาพการประมวลผลของ VMS-3 ที่มีขนาด 4 VCPU บน VMware, Hyper-V, OpenStack Hypervisor ดังแสดงในรูปที่ 4.3 ผลลัพธ์ส่วนของการประมวลผลที่เกี่ยวข้อง Integer Math, Floating Point Math, Prime Number, Sorting และ Single Threaded บน VMware Hypervisor ยังคงให้ค่าผลลัพธ์ออกมาสูงสุด นั่นคือการประมวลบน VMware Hypervisor สามารถประมวลผลชุดคำสั่งได้รวดเร็วกว่า OpenStack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ Hyper-V ค่าผลลัพธ์ของ VMware ที่ออกมาสูงสุดแสดงไว้ตามสัญลักษณ์วงกลมสีแดง ในรูปที่ 4.3



รูปที่ 4.3 กราฟเปรียบเทียบการประมวลผลแบบ 4 VCPU

จากหัวข้อ 4.1.1-4.1.2 ที่ทำการทดสอบกับขนาด 1 VCPU และ 2 VCPU ยิ่งมีการปรับค่า VCPU เพิ่มขึ้น ความสามารถในการประมวลผลชุดคำสั่งก็ยิ่งทำได้มากขึ้น ในขณะเดียวกัน Hyper-V Hypervisor ถึงแม้จะมีการปรับขนาด VCPU จาก 1 VCPU, 2 VCP และ 4 VCPU แต่ยังคงให้ค่าประสิทธิภาพออกมาได้ต่ำสุดเมื่อเทียบกับ VMware และ OpenStack

4.2 การทดสอบประสิทธิภาพของหน่วยความจำ

ในส่วนของการทดสอบประสิทธิภาพของหน่วยความจำ การใช้งาน Virtual Memory ได้กำหนด Parameter ในการทดสอบไว้ตามตารางที่ 3.3 โดยใช้ Passmark Tool ในการทดสอบบน VMware, OpenStack, Hyper-V Hypervisor ซึ่งได้ผลลัพธ์ประสิทธิภาพดังตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลทดสอบประสิทธิภาพของหน่วยความจำ

Spec	Hypervisor	Read Cached(Mbytes Transferred per second)	Read Uncashed(Mbytes Transferred per second)	Write (Mbytes Transferred per second)
2 GB ram	VMware	17412	10260	7888
4 GB ram	VMware	21994	10226	7833
8 GB ram	VMware	21800	10652	8279
2 GB ram	Hyper-V	8400	3186	4423
4 GB ram	Hyper-V	7695	3097	4425
8 GB ram	Hyper-V	8344	2790	4444
2 GB ram	OpenStack	21760	6970	7193
4 GB ram	OpenStack	14599	4672	4693
8 GB ram	OpenStack	17027	5401	6130

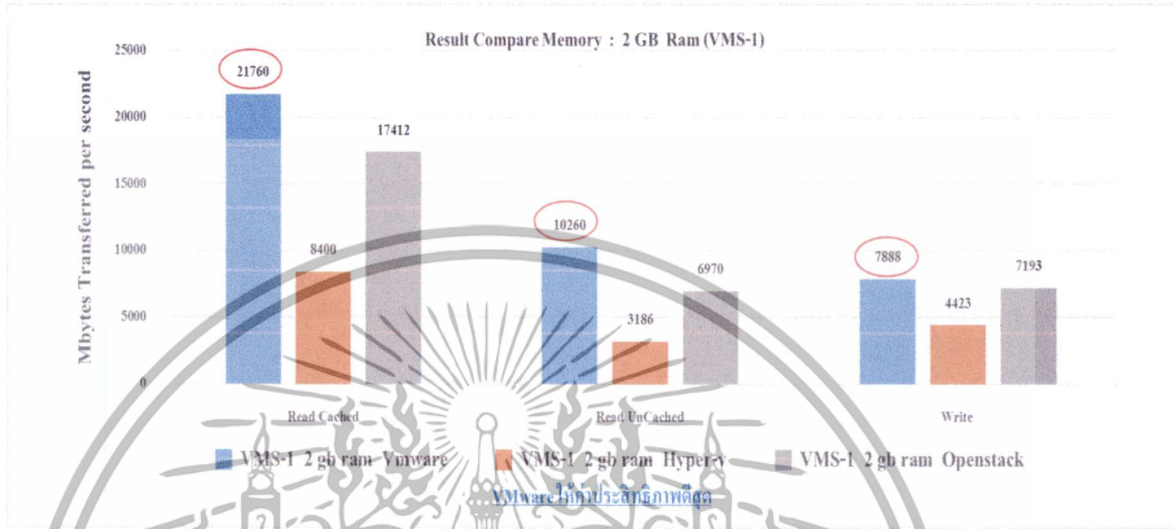
จากตารางผลลัพธ์ที่ 4.2 การบริหารจัดการหน่วยความจำสำหรับ 3 Hypervisor จากการทดสอบได้ทำการวัดค่าการอ่านข้อมูลจากหน่วยความจำแคชแทนการอ่านข้อมูลตรงจากหน่วยความจำหลัก, การอ่านข้อมูลโดยไม่ใช้งานหน่วยความจำ แคช, ความสามารถในการเขียนข้อมูลลงดิสก์ รวมถึงระยะเวลาหน่วงในการอ่านเขียนหน่วยความจำ โดยแยกตาม Sizing ของ Memory ได้ดังนี้

4.2.1 ความสามารถในการอ่านเขียนหน่วยความจำของ Memory ขนาด 2 GB

ผลการทดสอบวัดประสิทธิภาพในการอ่านเขียนหน่วยความจำของ VMS-1 ที่มีขนาด Memory เท่ากับ 2 GB บน VMware, OpenStack, Hyper-V Hypervisor ดังรูปที่ 4.4 การอ่านหน่วยความจำแบบใช้หน่วยความจำแคช (Read Cached) แทนการอ่านข้อมูลตรงจากหน่วยความจำหลัก VMware มีการถ่ายโอนสูงสุดที่ 21,760 Mbytes transfer per seconds ซึ่งสูงกว่า OpenStack และ Hyper-V และการอ่านข้อมูลโดยไม่ใช้งานหน่วยความจำแคช (Read Uncached) ผลการทดสอบที่ได้แสดงให้เห็นว่า VMWare Hypervisor ให้ค่าผลลัพธ์ดีที่สุดโดยการอ่านข้อมูลได้ที่ 10,260 Mbytes transfer per seconds รองลงมาคือ OpenStack Hypervisor และ Hyper-V Hypervisor ยังคงให้ค่าที่ต่ำสุด ส่วนการเขียนข้อมูลหน่วยความจำ (Write) VMware ยังคงให้ผลลัพธ์ที่ดีสุดที่ 7,888 Mbytes transfer per

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

seconds ใกล้เคียงกับ OpenStack ซึ่งเขียนข้อมูลหน่วยความจำได้เท่ากับ 7,193 Mbytes transfer per seconds ในขณะที่ Hyper-V ยังคงให้ผลลัพธ์ที่ต่ำสุดในทุกค่าที่ทำการทดสอบทั้งการอ่าน (Read Cached, Read UnCached) และเขียนหน่วยความจำ (Write)

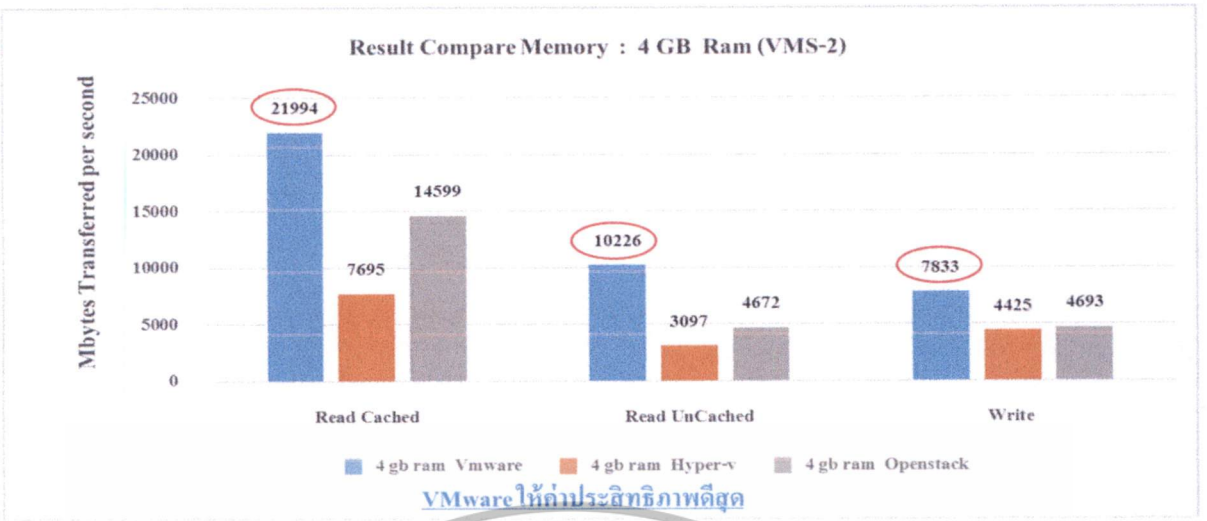


รูปที่ 4.4 ผลการทดสอบการอ่านเขียนหน่วยความจำของ Memory ขนาด 2 GB

4.2.2 ความสามารถในการอ่านเขียนหน่วยความจำของ Memory ขนาด 4 GB

ผลการทดสอบวัดประสิทธิภาพในการอ่านเขียนหน่วยความจำของ VMS-2 ที่มีขนาด Memory เท่ากับ 4 GB บน VMware, OpenStack, Hyper-V Hypervisor ดังรูปที่ 4.5 การอ่านหน่วยความจำแบบใช้หน่วยความจำแคช (Read Cached) แทนการอ่านข้อมูลตรงจากหน่วยความจำหลัก VMware มีการถ่ายโอนสูงสุดที่ 21,994 Mbytes transfer per seconds เช่นเดียวกับการอ่านข้อมูลโดยไม่ใช้งานหน่วยความจำแคช (Read Uncached) ผลการทดสอบที่ได้ VMware Hypervisor ให้ค่าผลลัพธ์ดีที่สุดโดยอ่านข้อมูลได้ที่ 10,226 Mbytes transfer per seconds รองลงมาคือ OpenStack และ Hyper-V ส่วนการเขียนข้อมูลหน่วยความจำ (Write) ทางด้าน VMWare ยังคงให้ผลลัพธ์ที่ดีสุดที่ 7,833 Mbytes transfer per seconds ในขณะที่ Hyper-V Hypervisor ยังคงให้ผลลัพธ์ที่ต่ำสุดในทุกค่าที่ทำการทดสอบทั้งการอ่าน (Read Cached, Read UnCached) และเขียนหน่วยความจำ (Write)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

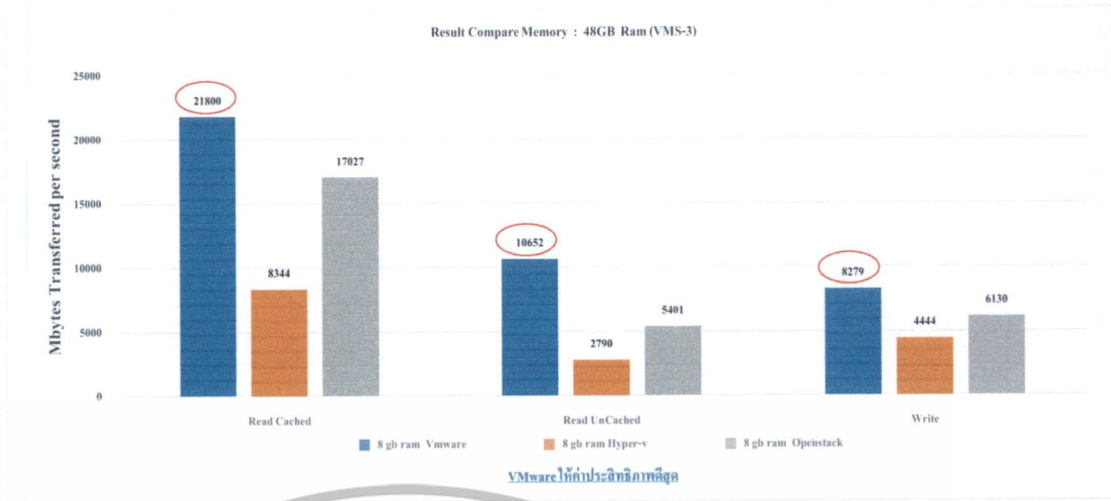


รูปที่ 4.5 ผลการทดสอบการอ่านเขียนหน่วยความจำของ Memory ขนาด 4 GB

4.2.3 ความสามารถในการอ่านเขียนหน่วยความจำของ Memory ขนาด 8 GB

ผลการทดสอบวัดประสิทธิภาพในการอ่านเขียนหน่วยความจำของ VMS-3 ที่มีขนาด Memory เท่ากับ 8 GB บน VMware, OpenStack, Hyper-V Hypervisor ดังรูปที่ 4.6 การอ่านหน่วยความจำแบบใช้หน่วยความจำแคช (Read Cached) แทนการอ่านข้อมูลตรงจากหน่วยความจำหลักทางด้าน VMware ยังคงมีการถ่ายโอนสูงสุดที่ 21,800 Mbytes transfer per seconds เช่นเดียวกับการอ่านข้อมูลโดยไม่ใช้งานหน่วยความจำแคช (Read Uncached) ผลการทดสอบที่ได้ VMware Hypervisor ให้ค่าผลลัพธ์ดีที่สุดโดยอ่านข้อมูลได้ที่ 10,652 Mbytes transfer per seconds ส่วนการเขียนข้อมูลหน่วยความจำ (Write) ทางด้าน VMWare ยังคงให้ผลลัพธ์ที่ดีสุดที่ 8,279 Mbytes transfer per seconds ในขณะที่ Hyper-V Hypervisor ยังคงให้ผลลัพธ์ที่ต่ำสุดในทุกค่าที่ทำการทดสอบทั้งการอ่าน (Read Cached, Read UnCached) และเขียนหน่วยความจำ (Write) จากผลการทดลองในหัวข้อ 4.2.1–4.2.3 เมื่อทำการทดสอบกับขนาด Memory ที่มีการปรับเพิ่มขึ้นจากเดิม 2 GB เป็น 4 GB และ 8 GB ผลลัพธ์ประสิทธิภาพในการอ่านเขียนหน่วยความจำ VMware Hypervisor ยังคงให้ผลลัพธ์ออกมาสูงสุดเมื่อเทียบกับ OpenStack และ Hyper-V

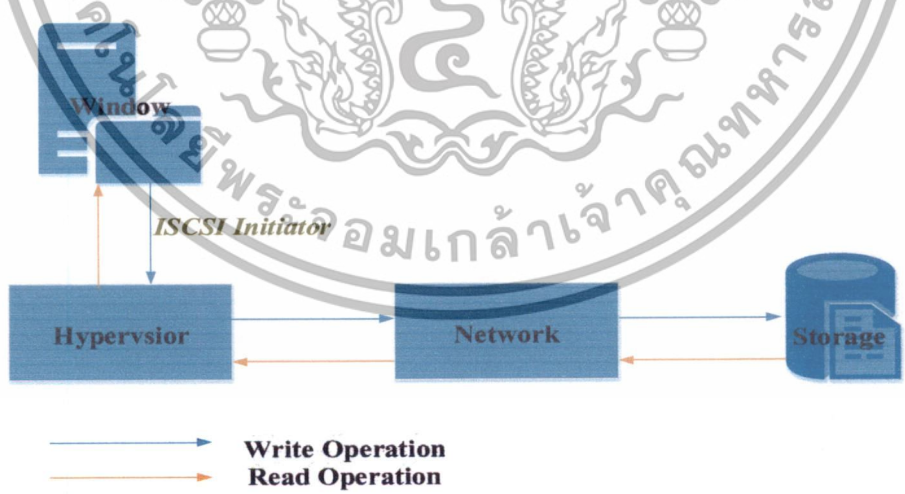
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ผลการทดสอบการอ่านเขียนหน่วยความจำของ Memory ขนาด 8 GB

4.3 การทดสอบประสิทธิภาพของดิสก์

การทดสอบประสิทธิภาพของดิสก์ทั้ง 3 Hypervisor มีการสร้างสภาพแวดล้อมสำหรับการทดสอบหลังจากมีการติดตั้ง Hypervsior และมีการสร้าง Virtual Machine ที่ติดตั้งระบบปฏิบัติการเป็น Window Server ในการใช้งาน Storage มีการ Map Disk ผ่าน ISCSI Initiater ซึ่งในการทำงานจะมี Operation ที่เกี่ยวข้องอยู่ 2 ส่วนคือส่วนของ Write Operation และส่วนของ Read Operation ซึ่งการเชื่อมต่อไปยัง Storage จะมีการเชื่อมต่อผ่าน Network IP โดยมีรายละเอียดการเชื่อมต่อดังแสดงในรูปที่ 4.7



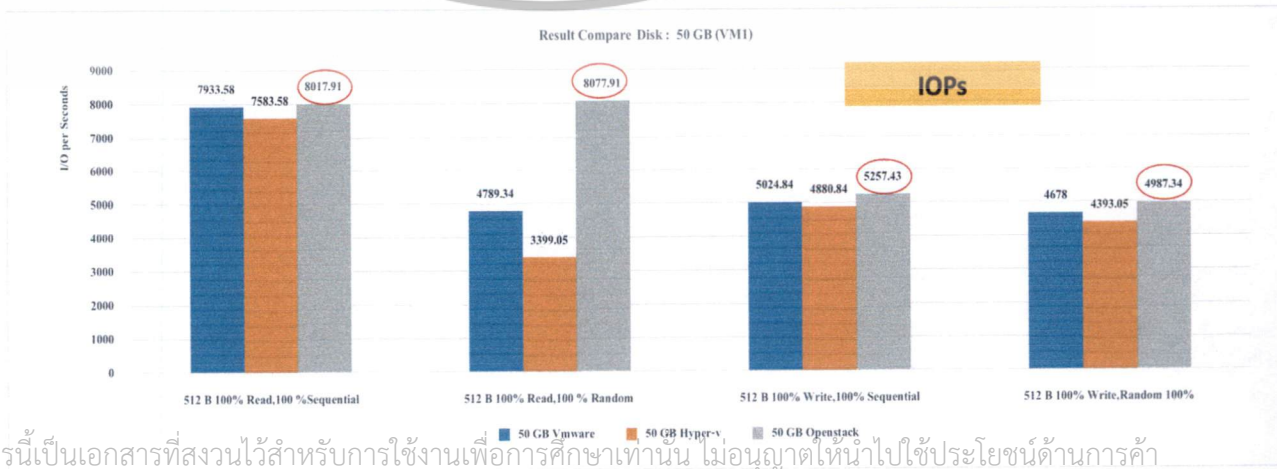
รูปที่ 4.7 Read - Write Operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นกำหนด Parameter ที่ทำการทดสอบสำหรับการอ่านเขียนข้อมูลลงดิสก์ทั้งแบบสุ่มและแบบลำดับตามตารางที่ 3.4 การทดสอบบน VMware, OpenStack และ Hyper-V Hypervisor ครั้งนี้ใช้ กำหนดให้มีการใช้ Workload จำนวน 2 Worker ในการทดสอบจากนั้นกำหนดค่า Block Size การอ่านเขียนข้อมูลขนาดเท่ากับ 512 Byte ของ Logical block ทั้งแบบสุ่มและลำดับ หรือเทียบเท่ากับ 8 Block (4 Kbyte) ของ Physical Block โดยสภาพแวดล้อมที่ใช้งานและติดตั้งทดสอบเป็นการทดสอบบนทรัพยากรแบบ Share Disk ซึ่งการทดสอบมีการเชื่อมต่อระหว่าง Host กับ Target ผ่าน ISCSI Initiator จากการทดสอบ I/O Workload และ Latency ได้ผลดังนี้

4.3.1 ทดสอบการอ่านเขียนข้อมูลขนาด 50 GB

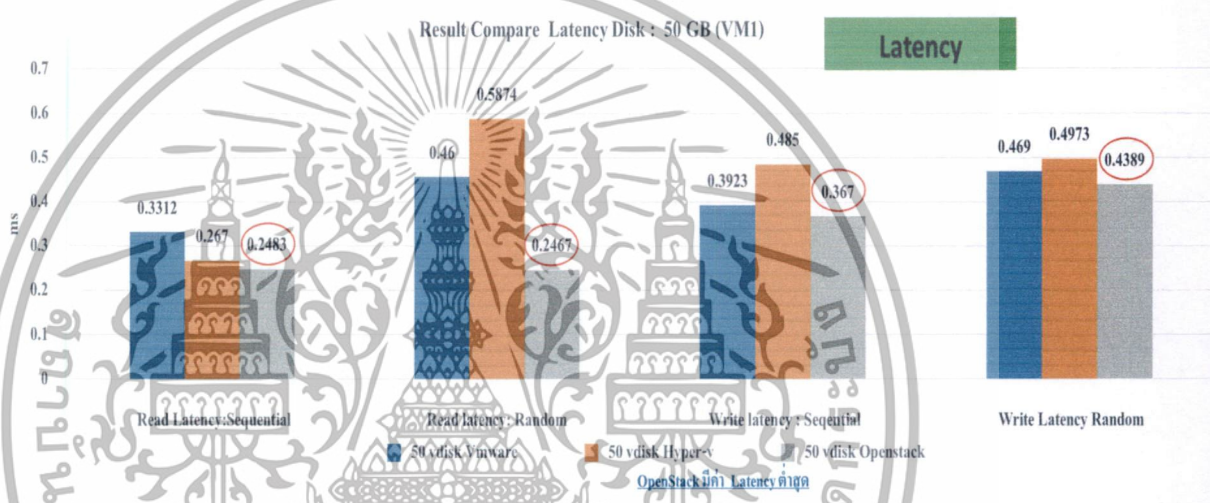
จากรูปที่ 4.8 ผลลัพธ์การทดสอบ I/O per seconds บน VMware, OpenStack, Hyper-V Hypervisor ทำให้ได้ผลลัพธ์ Hypervisor ที่มีค่า I/O สูงสุดในส่วนของการอ่านเขียนข้อมูลขนาด 512 Byte ทั้งแบบ Sequential และ Random ซึ่งมีการกำหนด Parameter ที่ทดสอบอ้างอิงตามตารางที่ 3.4 ผลลัพธ์ที่ได้จากการทดสอบ OpenStack Hypervisor ให้ค่า I/O per seconds สูงสุดทุกค่าโดยการอ่านข้อมูลแบบ Sequential ให้ค่าสูงสุดที่ 8,017.91 I/O per seconds, การอ่านข้อมูลแบบ Randomให้ค่าสูงสุดที่ 8,077.91 I/O per seconds, การเขียนข้อมูลแบบลำดับให้ค่าสูงสุดที่ 5,257.43 I/O per seconds และการเขียนข้อมูลแบบสุ่มให้ค่าสูงสุดที่ 4,987.34 I/O per seconds รองลงมาคือ VMware Hypervisor โดยการอ่านข้อมูลแบบ Sequential ได้เท่ากับ 7,933.58 I/O per seconds, การอ่านข้อมูลแบบ Random ได้เท่ากับ 4,789.34 I/O per seconds, การเขียนข้อมูลแบบลำดับ ได้เท่ากับ 5,024.84 I/O per seconds และการเขียนข้อมูลแบบสุ่มได้เท่ากับ 4,678 I/O per seconds และสุดท้ายคือ Hyper-V ที่ให้ค่า I/O per seconds ที่ต่ำสุดเมื่อเทียบกับ OpenStack และ VMware ค่าที่ได้แสดงดังรูปที่ 4.8 จากการทดสอบด้วยจำนวน Workload ที่เท่ากัน OpenStack มีความสามารถในการอ่านเขียนข้อมูลหรือทำงานระดับ Block File ได้ดีกว่า VMware และ Hyper-V



รูปที่ 4.8 ผลลัพธ์การทดสอบ I/O ขนาดของ Disk 50 GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ในส่วนของผลลัพธ์ของ Latency ที่ได้จากการทดสอบดังแสดงไว้ในรูปที่ 4.9 บน OpenStack Hypervisor การอ่านเขียนข้อมูลขนาด 512 Byte ทั้งแบบ Sequential และ Random ให้ค่าผลลัพธ์ที่ต่ำสุด (ปกติค่า Latency ยิ่งน้อยยิ่งดี) เมื่อเทียบกับ VMware และ Hyper-V โดยการอ่านแบบ Sequential ค่า Latency ของ OpenStack ต่ำสุดที่ 0.2483 ms, การอ่านแบบ Random ค่า Latency ของ OpenStack ต่ำสุดที่ 0.2467 ms, การเขียนแบบ Sequential ค่า Latency ต่ำสุดที่ 0.367 ms และสุดท้ายคือการเขียนแบบ Random ค่า Latency ของ OpenStack ต่ำสุดที่ 0.4389 ms การที่มีค่า Latency ที่ต่ำจะส่งผลดีกับประสิทธิภาพการทำงานของ Database และ Application

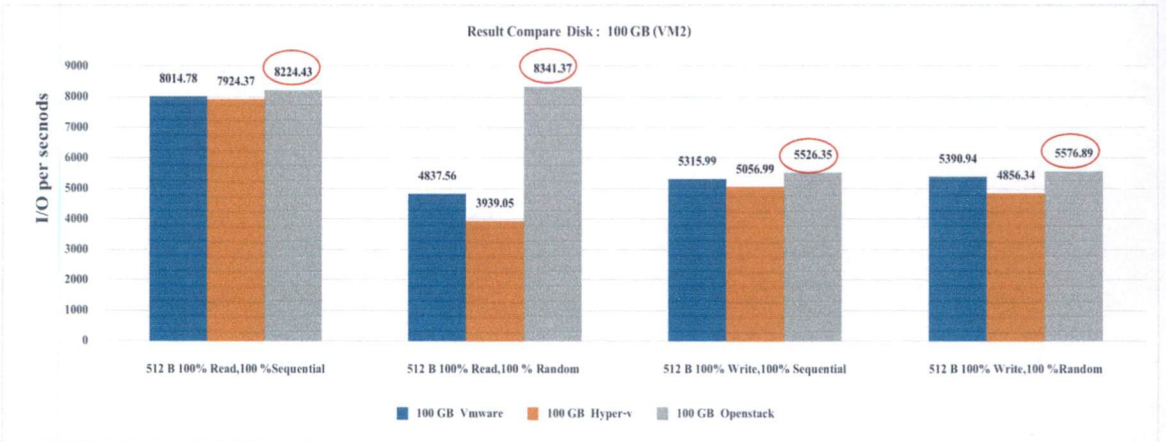


รูปที่ 4.9 ผลลัพธ์ Latency ของ Disk ขนาด 50 GB (ms)

4.3.2 ทดสอบการอ่านเขียนข้อมูลขนาด 100 GB

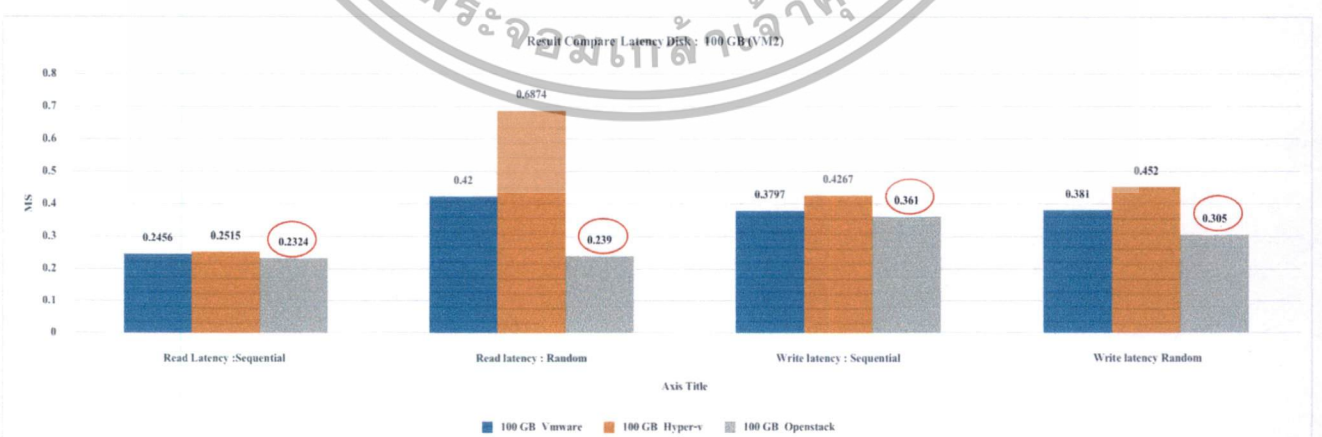
ผลการทดสอบ I/O per seconds จากรูปที่ 4.10 เมื่อทำการทดสอบ Disk ขนาด 100 GB เพื่อหาค่า I/O per second สูงสุดบน VMware, OpenStack, Hyper-V Hypervisor ในการอ่านและเขียนข้อมูลขนาด 512 Byte ทั้งแบบ Sequential และ Random ผลการทดสอบบน OpenStack Hypervisor ยังคงให้ค่า I/O สูงสุดเมื่อเทียบกับ VMware และ Hyper-V โดยให้ค่า I/O ในการอ่านข้อมูลแบบ Sequential เท่ากับ 8,224.43 I/O per second, การอ่านข้อมูลแบบ Random ให้ค่า I/O เท่ากับ 8,341.37 I/O per second ซึ่งเมื่อเทียบกับ VMware ที่ให้ค่า I/O เท่ากับ 4837.56 I/O per second แต่ต่างกันเกือบถึงสองเท่า ในขณะที่การเขียนข้อมูบแบบ Sequential ทางด้าน OpenStack ให้ค่าสูงสุดเท่ากับ 5,526.35 I/O per second และสุดท้ายการเขียนข้อมูลแบบ Random บน OpenStack ให้ค่าสูงสุดเท่ากับ 5,576.89 I/O per seconds ทั้งนี้จากผลลัพธ์ที่ได้ทำการทดสอบ Hyper-V ให้ค่า I/O per seconds ที่ต่ำสุดทั้งการอ่านเขียนแบบ Sequential และ Random แสดงไว้ดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ผลลัพธ์การทดสอบ I/O ขนาดของ Disk 100 GB

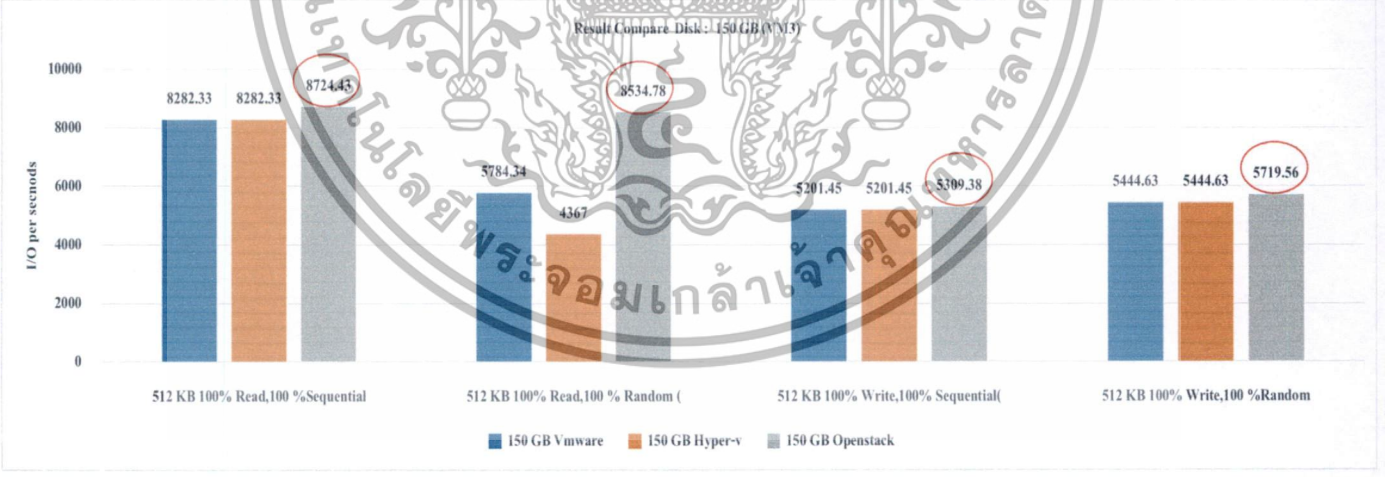
นอกจากนี้ในส่วนของ Latency ของการทดสอบ Disk ขนาด 100 GB ผลลัพธ์ที่ได้ทดสอบกับการอ่านเขียนข้อมูลขนาด 512 Byte ทั้งแบบ Sequential และ Random ของ OpenStack Hypervisor ยังคงให้ผลลัพธ์ที่ต่ำที่สุด (Latency ยิ่งน้อยยิ่งดี) ดังแสดงในรูปที่ 4.11 เมื่อเทียบกับ VMware และ Hyper-V โดยการอ่านข้อมูลแบบ Sequential ทางด้าน OpenStack มีค่า Latency ที่ต่ำสุดเท่ากับ 0.2323 ms, การอ่านข้อมูลแบบ Random ทางด้าน OpenStack มีค่า Latency ที่ต่ำสุดเท่ากับ 0.239 ms เมื่อเทียบกับ I/O ของทางด้าน Hyper-V กลับให้ค่า Latency ที่สูงถึง 0.6874 ms ซึ่งค่าสูงยิ่งไม่ดีต่อการทำงานของ Database และ Application ส่วนการเขียนข้อมูลแบบ Sequential ทางด้าน OpenStack มีค่า Latency ต่ำสุดเท่ากับ 0.361 ms และการเขียนข้อมูลแบบ Random ทางด้าน OpenStack ยังคงให้ค่า Latency ที่ต่ำสุดเท่ากับ 0.305 ms จากผลการทดสอบที่ได้ Hyper-V Hypervisor มีค่า Latency ที่สูงกว่า OpenStack และ VMware ค่าที่ได้แสดงไว้ในรูปที่ 4.11



รูปที่ 4.11 ผลลัพธ์ Latency ของ Disk ขนาด 100 GB (ms)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

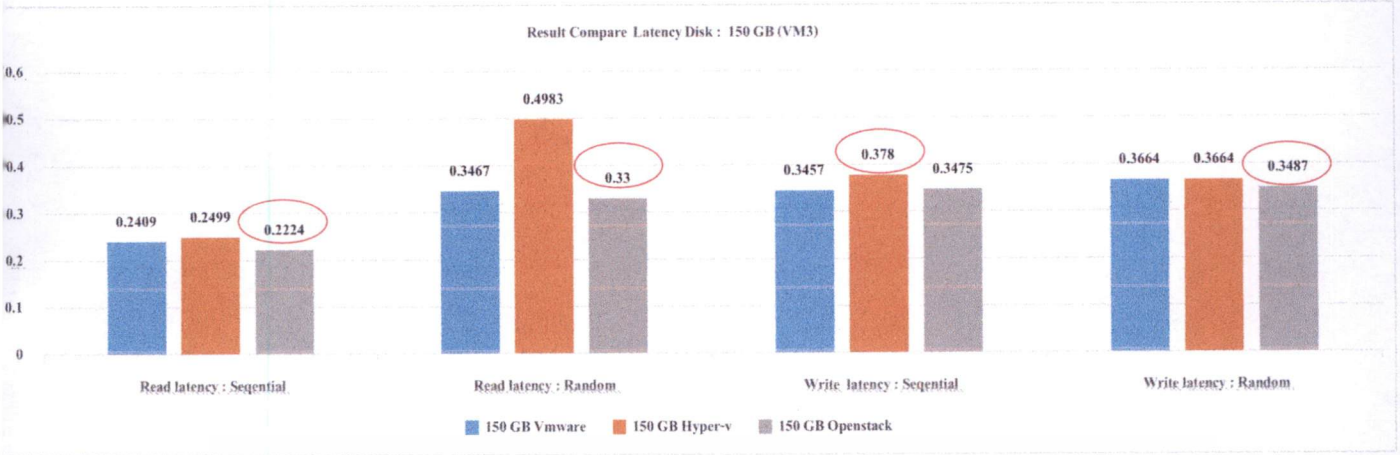
เมื่อปรับขนาดของ Disk เป็น 150 GB และทดสอบวัดประสิทธิภาพของ I/O ในการอ่านและเขียนข้อมูลบน VMware, OpenStack และ Hyper-V ผลลัพธ์การทดสอบ I/O ของทั้ง 3 Hypervisor ดังแสดงในรูปที่ 4.12 OpenStack ให้ค่า I/O ในส่วนของการอ่านเขียนข้อมูลขนาด 512 Byte แบบ Sequential และ Random ได้สูงสุดทุกค่าที่ได้ทำการทดสอบ โดยการอ่านข้อมูบแบบ Sequential ทางด้าน OpenStack Hypervisor ยังคงให้ค่า I/O สูงสุดเท่ากับ 8,724.43 I/O per second, การอ่านข้อมูลแบบ Random ทางด้าน OpenStack Hypervisor ยังให้ค่า I/O สูงสุดเท่ากับ 8,534.78 I/O per second, การเขียนข้อมูลแบบ Sequential ทางด้าน OpenStack Hypervisor ยังให้ค่า I/O สูงสุดเท่ากับ 5,309.38 I/O per second และการเขียนข้อมูลแบบ Random ทางด้าน OpenStack Hypervisor ยังให้ค่า I/O สูงสุดเท่ากับ 5,719.56 I/O per second จากผลการทดสอบสังเกตได้ว่า Hyper-V ยังคงให้ค่า I/O ต่ำสุดเมื่อเทียบกับอีก 2 Hypervisor นอกจากนี้ ขนาดของ Disk ที่เพิ่มขึ้นส่งผลให้ค่าการอ่านและเขียนคำสั่ง (I/O per second) ได้ค่าที่สูงขึ้นด้วย เมื่อกลับไปดูผลการทดลองในหัวข้อ 4.3.1-4.3.2 เป็นการทดลองกับขนาดของ Disk 50 GB และ 100 GB ในการใช้งานบน Virtualization พื้นที่ที่ใช้ในการจัดเก็บข้อมูลที่เพิ่มขึ้นทำให้สามารถใช้งาน I/O ได้เพิ่มขึ้น ยกตัวอย่างเปรียบเทียบจากข้อมูลการอ่านแบบ Sequential ของ Disk ขนาด 50 GB, 100 GB และ 150 GB ตามลำดับ ได้ผลดังนี้ 8,017.91 I/O per second, 8,224.43 I/O per second และ 8,724.43 I/O per second



รูปที่ 4.12 ผลลัพธ์การทดสอบ I/O ขนาดของ Disk 150 GB

นอกจากนี้ในส่วนของ Latency ของการทดสอบ Disk ขนาด 150 GB ผลลัพธ์ที่ได้ทดสอบกับการอ่านเขียนข้อมูลขนาด 512 Byte ทั้งแบบ Sequential และ Random ของ OpenStack Hypervisor ยังคงให้ผลลัพธ์ที่ต่ำที่สุด (Latency ยิ่งน้อยยิ่งดี) ดังแสดงในรูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ผลลัพธ์ Latency ของ Disk ขนาด 150 GB (ms)

เมื่อเทียบผล Latency ระหว่าง OpenStack กับ VMware และ Hyper-V โดยการอ่านข้อมูลแบบ Sequential ทางด้าน OpenStack มีค่า Latency ที่ต่ำสุดเท่ากับ 0.2224 ms รองลงมาคือ VMware และ Hyper-V ซึ่งให้ค่า Latency เท่ากับ 0.2409 ซึ่งมีค่าเท่ากัน ในส่วนของการอ่านข้อมูลแบบ Random ทางด้าน OpenStack มีค่า Latency ที่ต่ำสุดเท่ากับ 0.33 ms เมื่อเทียบกับ I/O ของทางด้าน Hyper-V กลับให้ค่า Latency ที่สูงถึง 0.4983 ms และ VMware ให้ค่า Latency เท่ากับ 0.3467 ms จากค่า Latency ของ Hyper-V ในส่วนของการอ่านข้อมูลแบบ Random ที่ได้ค่าสูงยิ่งไม่ดีต่อการทำงานของ Database และ Application ส่วนการเขียนข้อมูลแบบ Sequential ทางด้าน OpenStack มีค่า Latency ต่ำสุดเท่ากับ 0.3475 ms รองลงมาคือ VMware มีค่า Latency เท่ากับ 0.3457 และ Hyper-V มีค่า Latency เท่ากับ 0.378 ms ส่วนสุดท้ายคือการเขียนข้อมูลแบบ Random ทางด้าน OpenStack ยังคงให้ค่า Latency ที่ต่ำสุดเท่ากับ 0.3487 ms รองลงมาคือ VMware และ Hyper-V ซึ่งมีค่า Latency เท่ากับ 0.3664 ซึ่งเป็นตัวเลขที่เท่ากัน จากการทดลองในหัวข้อ 4.3.1-4.3.3 ในมุมของการเลือกใช้งานเพื่อรองรับการทำงานของ Application และ Database ต้องการการอ่านและเขียนชุดคำสั่งที่รวดเร็ว รวมถึงค่า Latency ที่ต่ำ โดย OpenStack มีความสามารถทั้ง 2 ด้านในการจัดการกับขนาด file ขนาด 512 KB ได้ดีกว่าทั้ง VMware และ Hyper-V

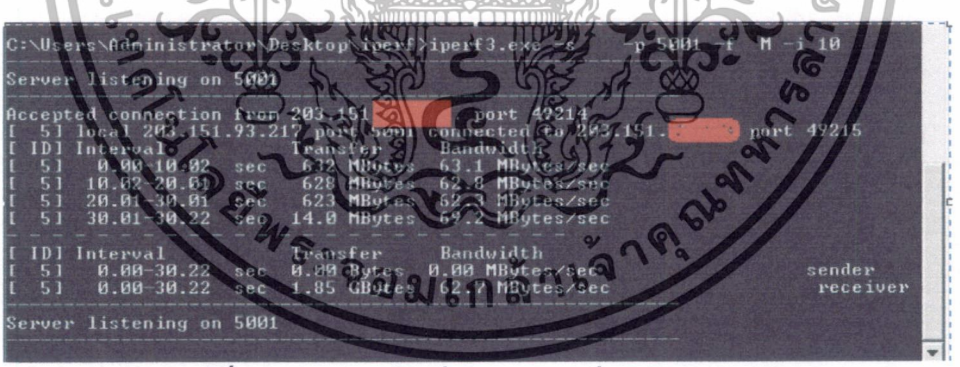
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดสอบประสิทธิภาพของ Network Throughput ภายในเครือข่าย

หลังจากทำการ Provisioning Virtual Machine และทำการสร้าง Virtual Network Interface Card ให้กับแต่ละ VM บน VMware, OpenStack และ Hyperv-V Hypervisor โดยทั้ง 3 Hypervisor สามารถสร้าง Virtual Machine ที่รองรับ Maximum Speed ได้ถึง 10 Gbps ในส่วนของการทดสอบจะทำการทดสอบโดยการส่งข้อมูลกันภายในเครือข่าย (Private Network) โดยทำการจำลองการทำงานในลักษณะของ Client-Server ซึ่ง Software จะทำการ Simulate Package และส่งออกทั้งแบบ TCP และ UDP โดยมีการกำหนด Parameter ในฝั่งของ Client – Server ในทุก Virtual Machine ที่ทำการทดสอบบนทั้ง VMware, OpenStack และ Hyper-V ดังนี้

4.4.1 การทดสอบประสิทธิภาพการส่งข้อมูลภายในเครือข่ายด้วย Iperf3 Tool บนเครื่อง Server และ Client เพื่อจำลองการส่งข้อมูลด้วยการเชื่อมต่อแบบ TCP

จากรูป 4.14 เป็นการกำหนด Parameter ที่จะทดสอบบนฝั่งของเครื่อง Server โดยมีรายละเอียดดังนี้ คำสั่ง iperf.exe -s -p 5001 -f M -i 10 คือคำสั่งที่จะทำการจำลองเครื่อง Server IP Address 203.151.93.216 ให้ส่งข้อมูลไปยัง Client IP Address 203.151.93.217 โดยใช้พอร์ต 5001 สำหรับการเชื่อมต่อเพื่อส่งข้อมูลแบบ TCP และให้แสดงผลในรูปแบบ Megabyte ทุกๆ 10 วินาที จากรูปที่ 4.14 ฝั่ง (Sender) สามารถถ่ายโอนข้อมูลให้กับฝั่งรับ (Receiver) ขนาด 1.85 GBytes โดยใช้ Bandwidth 62.7 Mbytes/sec ในการส่งข้อมูลออก

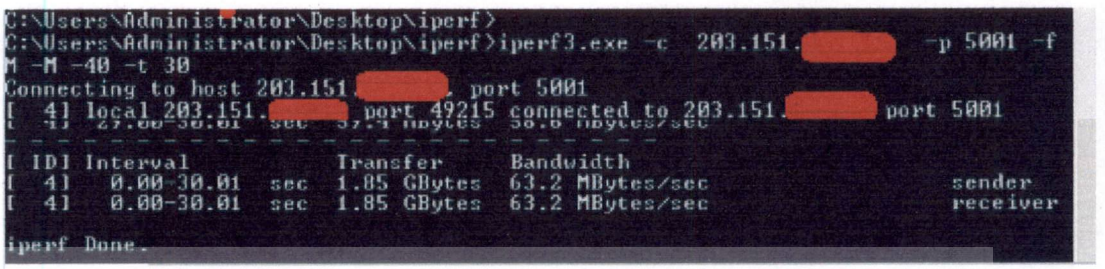


รูปที่ 4.14 Iperf3 บนเครื่อง Server เพื่อทดสอบ TCP

ในส่วนของการจำลองเครื่อง Client IP Address 203.151.93.217 ดังรูปที่ 4.15 ด้วยคำสั่ง iperf.exe -c IP-Server -p 5001 -f M -M -40 -t 30 t คือการจำลองให้ Client IP Address 203.151.93.217 มีการเชื่อมต่อกับ Server ด้วย Port 5001 โดยเป็นการส่งข้อมูลแบบ TCP หลังจากนั้นทำการแสดงผลในรูปแบบ Megabyte โดยกำหนดให้ขนาดของ TCP header มีค่าเท่ากับ 40 Byte และแสดงผลทุกๆ 30 วินาที จากผลลัพธ์ในรูปที่ 4.15 ข้อมูลที่ฝั่งส่ง (Sender) ส่งมาขนาด 1.85 Gbytes ด้วยขนาดของ Bandwidth 63.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

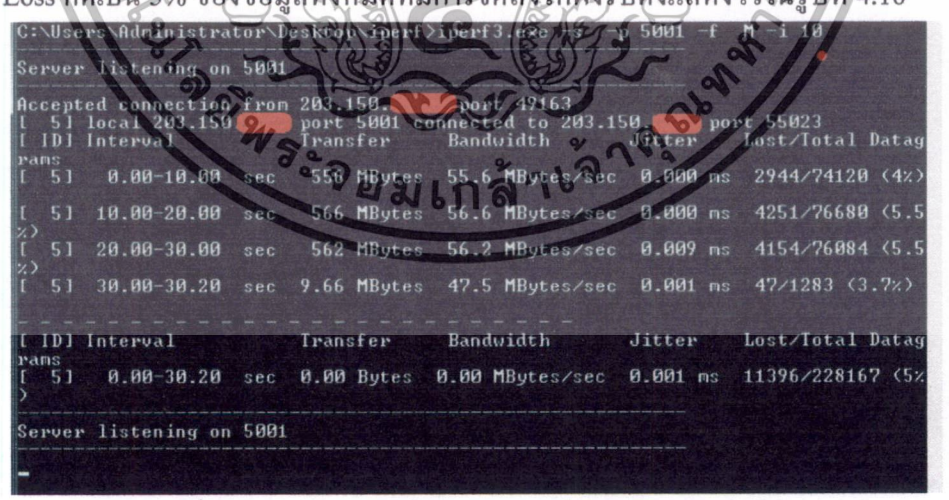
Mbytes/sec ฝั่งรับสามารถรับข้อมูลได้เท่ากับที่ฝั่งต้นทางส่งมาให้โดยไม่เกิดการสูญหายของข้อมูล



รูปที่ 4.15 Iperf3 บนเครื่อง Client เพื่อทดสอบ TCP

4.4.2 การทดสอบประสิทธิภาพการส่งข้อมูลภายในเครือข่ายด้วย Iperf3 Tool บนเครื่อง Server และ Client เพื่อจำลองการส่งข้อมูลด้วยการเชื่อมต่อแบบ UDP

จากรูป 4.16 เป็นการกำหนด Parameter ที่จะทดสอบบนฝั่งของเครื่อง Server โดยมีรายละเอียดดังนี้ คำสั่ง iperf.exe -s -p 5001 –f M -i 10 คือคำสั่งที่จะทำการจำลองเครื่อง Server ให้ส่งข้อมูลไปยัง Client โดยใช้พอร์ต 5001 สำหรับการเชื่อมต่อเพื่อส่งข้อมูลแบบ UDP และให้แสดงผลในรูปแบบ Megabyte ทุกๆ 10 วินาที จากรูปที่ 4.16 ฝั่งส่ง (Sender) ข้อมูลที่มีการจัดส่งแบบ UDP เกิด Jitter (ระยะเวลา Delay ในการส่ง Datagram) เกิดขึ้น 0.001 ms ทำให้เกิด Datagram ที่ Out-of-order (Loss) เท่ากับ 11,396 Datagram โดย Datagram ที่เกิดการ Loss คิดเป็น 5% ของข้อมูลทั้งหมดที่มีการจัดส่งให้ฝั่งรับดังแสดงไว้ในรูปที่ 4.16



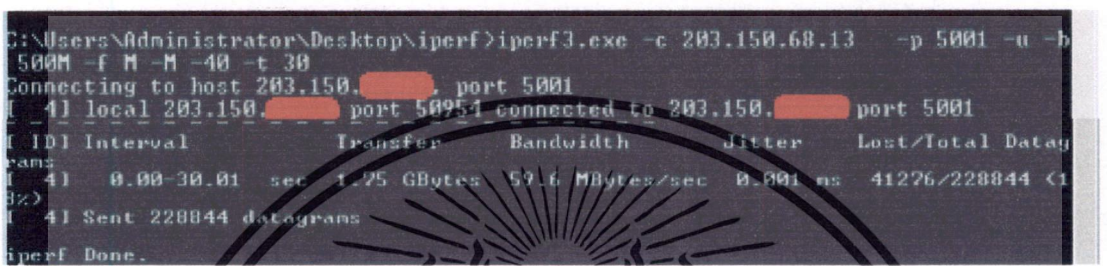
รูปที่ 4.16 Iperf3 บนเครื่อง Server เพื่อทดสอบ UDP

ในส่วนของการจำลองเครื่อง Client เพื่อให้สามารถรับข้อมูลแบบ UDP ได้ด้วยคำสั่ง iperf.exe -c IP-Server -p 5001 -u -b 500M -f M -t 30 ซึ่งทำการจำลองเครื่อง Client โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนด IP server ที่ต้องการเชื่อมต่อ กำหนดพอร์ต 5001 สำหรับการเชื่อมต่อเพื่อส่งข้อมูลแบบ UDP ด้วย Target Bandwidth 500 Mbps โดยทำการแสดงผลในรูปแบบ Megabyte ทุกๆ 30 วินาที จากผลลัพธ์การส่งข้อมูลระหว่างฝั่งส่งไปยังฝั่งรับ สามารถส่งข้อมูลขนาด 1.75 GBytes ด้วย Bandwidth 57.6 Mbytes/sec เกิด Datagram Loss ระหว่างการส่งข้อมูลเท่ากับ 41,276 Datagram โดยมี Jitter (ระยะเวลา Delay ในการส่ง Datagram) เกิดขึ้นเท่ากับ 0.001 ms



รูปที่ 4.16 Iperf3 บนเครื่อง Client เพื่อทดสอบ UDP

หลังจากติดตั้งซอฟแวร์เพื่อทดสอบบน OpenStack, VMware, Hyper-V Hypervisor จากนั้นทำการ Simulate Package ด้วย Iperf3 Tool เพื่อส่ง Package แบบ TCP และ UDP โดยได้ผลลัพธ์ของการทดสอบดังตารางที่ 4.3

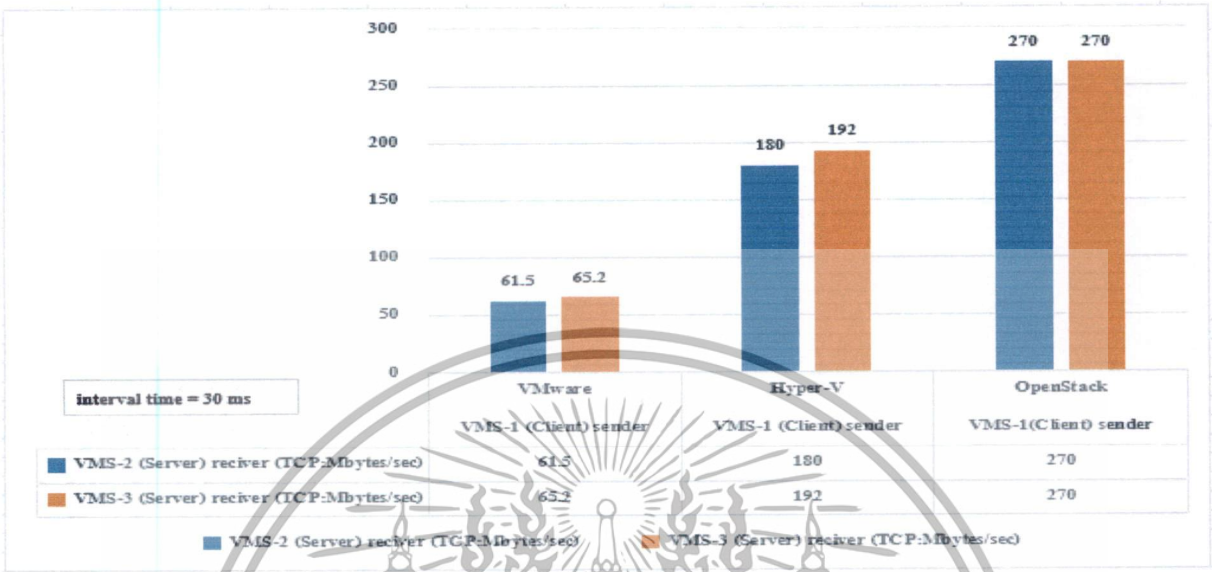
ตารางที่ 4. 3 ผลการทดสอบการส่งข้อมูลแบบ TCP

VM	Hypervisor	การส่งข้อมูลระหว่าง Client-Server ภายในเครือข่าย (Private network)		
		VMS-1 (Server) ฝั่งรับ (TCP:Mbytes/sec)	VMS-2 (Server) ฝั่งรับ (TCP:Mbytes/sec)	VMS-3 (Server) ฝั่งรับ (TCP:Mbytes/sec)
VMS-1 (Client) ฝั่งส่ง	VMware	-	61.5	65.2
VMS-1 (Client) ฝั่งส่ง	Hyper-V	-	179	192
VMS-1 (Client) ฝั่งส่ง	OpenStack	-	270	270

ตารางที่ 4.3 เป็นผลลัพธ์ที่ได้จากการจำลองการส่งข้อมูลโดยกำหนดให้เครื่อง VMS-1 เป็น Client หรือฝั่งผู้ส่ง และ VMS-2, VMS-3 เป็นเครื่อง Server หรือฝั่งผู้รับ โดยทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำลองแบบเดียวกันทั้ง VMware, Hyper-V, OpenStack ซึ่งผลการทดสอบการส่งข้อมูลจากตารางที่ 4.3 สามารถอธิบายรายละเอียดเพิ่มเติมได้ดังนี้



รูปที่ 4.17 ผลลัพธ์จำลองเครื่อง VMS-1 เป็น Client และ VMS-2, VMS-3 เป็นเครื่อง Server

จากตารางที่ 4.3 นำค่าข้อมูลที่ได้มาแสดงผลในรูปแบบของกราฟตามรูปที่ 4.17 จากการจำลองเครื่อง VMS-1 เป็น Client และ VMS-2, VMS-3 เป็นเครื่อง Server โดยกำหนดให้มีการส่ง Package แบบ TCP และกำหนดให้ Interval Time ในการจัดส่งข้อมูลมีค่าเท่ากับ 30 ms ผลการทดสอบที่ได้ Virtual Machine ที่รันอยู่บน OpenStack Hypervisor สามารถ Transfer ข้อมูลไปยัง VMS-2, VMS-3 สูงสุดได้ที่ 7.90 GBytes หรือเท่ากับ 270 MBytes/sec ซึ่งสูงสุดเมื่อเทียบกับ VMware และ Hyper-V Hypervisor โดยค่าที่ฝั่งส่ง (Sender) สามารถส่งไปยังฝั่งรับ (Receiver) ได้แสดงดังรูปที่ 4.18 ซึ่งเป็นช่วง Interval Time ตั้งแต่ 0.1 Sec – 30 Sec มีการ Transfer ข้อมูลได้ 7.98 GBytes ด้วย Bandwidth ขนาด 270 Mbytes/sec

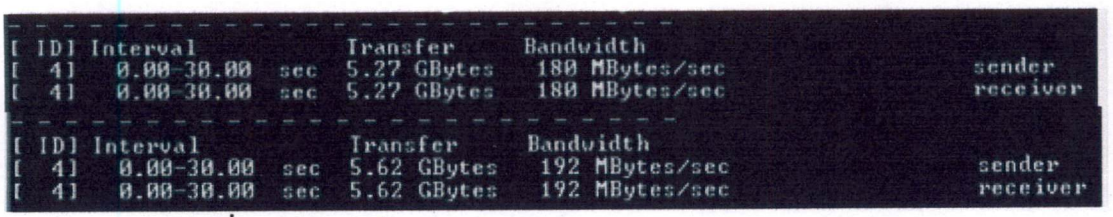
```
[ID] Interval           Transfer     Bandwidth
 4]   0.00-30.01  sec  7.90 GBytes   270 MBytes/sec                  sender
 4]   0.00-30.01  sec  7.90 GBytes   270 MBytes/sec                  receiver
```

รูปที่ 4.18 ผลลัพธ์การส่งข้อมูลแบบ TCP บน OpenStack Hypervisor

อันดับสองรองลงมาคือ Hyper-V Hypervisor จากรูปที่ 4.19 เป็นผลลัพธ์ของการส่งข้อมูลแบบ TCP โดยที่ VMS-1 (Client) สามารถ Transfer ไปยัง VMS-2 สูงสุดได้ที่ 5.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GBytes หรือเท่ากับ 180 MBytes/sec สามารถ Transfer ไปยัง VMS-3 (Server) สูงสุดได้ที่ 5.62 GBytes หรือเท่ากับ 192 MBytes/sec ดังรูปที่ 4.19



รูปที่ 4.19 ผลลัพธ์การส่งข้อมูลแบบ TCP บน Hyper-V Hypervisor

และสุดท้ายคือส่วนของ VMware Hypervisor ที่ Transfer ข้อมูลได้น้อยสุดเมื่อเทียบกับ OpenStack และ Hyper-V โดยที่ VMS-1 (Client) สามารถถ่ายโอนข้อมูลไปยัง VMS-2 สูงสุดได้ที่ 1.80 GBytes หรือเท่ากับ 61.5 MBytes/sec และ VMS-3 (Server) สูงสุดได้ที่ 7.90 GBytes หรือเท่ากับ 270 MBytes/sec ดังรูปที่ 4.20

รูปที่ 4.20 ผลลัพธ์การส่งข้อมูลแบบ TCP บน VMware Hypervisor

ตารางที่ 4. 4 ผลการทดสอบการส่งข้อมูลแบบ UDP

VM	Hypervisor	การส่งข้อมูลระหว่าง Client-Server ภายในเครือข่าย(Private network)		
		VMS-1 (Server) (UDP::Mbytes/sec)	VMS-2 (Server) (UDP:Mbytes/sec)	VMS-3 (Server) (UDP:Mbytes/sec)
VMS-1	VMware	-	59.5	58.2
VMS-1	Hyper-V	-	59.6	59.4
VMS-1	OpenStack	-	59.5	59.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.4 เป็นผลการทดสอบการส่งข้อมูลในแบบ UDP เนื่องจากการส่งจากฝั่งส่ง (VMS-1) ไปยังฝั่งรับ (VMS-2, VMS-3) บน VMware, Hyper-V, OpenStack เป็นการส่งแบบไม่มีการรับประกันการสูญหายของข้อมูล ดังนั้นจึงเกิด Package Loss และ Jitter เกิดขึ้น โดยผลลัพธ์ในการส่งข้อมูลที่ได้มีค่าใกล้เคียงกัน สามารถอธิบายรายละเอียดเพิ่มเติมดังนี้

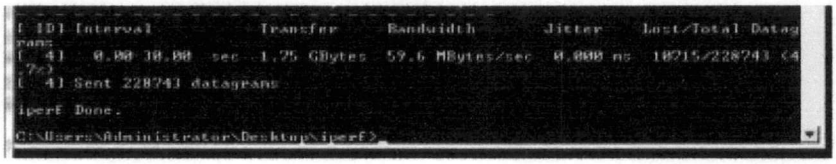
รูปที่ 4.21 ผลลัพธ์ส่งข้อมูลแบบ UDP บน VMware จาก VMS-1 ไปยัง VMS-2

รูปที่ 4.22 ผลลัพธ์ส่งข้อมูลแบบ UDP บน VMware จาก VMS-1 ไปยัง VMS-3

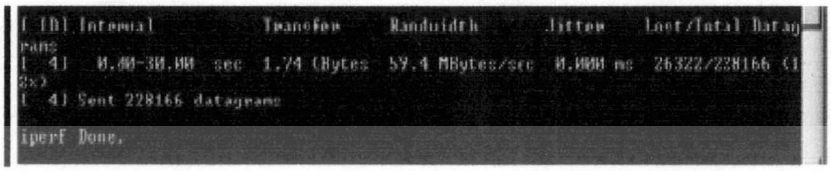
เมื่อทำการทดสอบส่งข้อมูลแบบ UDP บน VMware Hypervisor ผลการทดสอบทำให้ผลลัพธ์การส่งรับข้อมูลที่มีการจำลองเครื่อง VMS-1 ฝั่ง Client และ VMS-2, VMS-3 เป็นเครื่อง Server ในช่วงระยะเวลา 30 วินาที ดังนี้

- จากรูปที่ 4.21 VMS-1 (Client) ฝั่งส่งสามารถส่งข้อมูลไปยัง VMS-2 ได้ที่ 1.74 GBytes หรือเท่ากับ 59.5 Mbytes/sec ซึ่งมีการส่ง datagram ทั้งหมดเท่ากับ 228,389 datagram เกิด Datagram ที่ Out-f-order (Loss) เท่ากับ 191937 Datragram คิดเป็น 84% ของทั้งหมดที่สูญหายระหว่างการส่ง
- จากรูปที่ 4.22 VMS-1 (Client) ฝั่งส่งสามารถส่งข้อมูลไปยัง VMS-3 ได้ที่ 1.47 GBytes หรือเท่ากับ 58.2 Mbytes/sec จากที่มีการส่ง datagram เท่ากับ 192,814 datagram เกิด Datagram ที่ Out-of-order (Loss) เท่ากับ 120,888 Datagram คิดเป็น 63% ของทั้งหมดที่สูญหายระหว่างการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



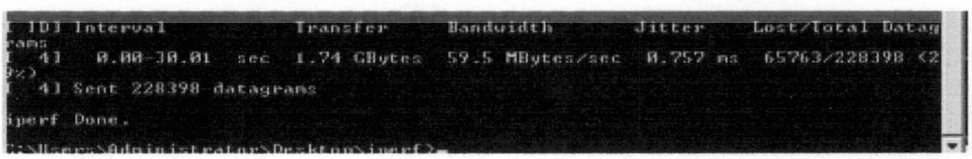
รูปที่ 4.23 ผลลัพธ์ส่งข้อมูลแบบ UDP บน Hyper-V จาก VMS-1 ไปยัง VMS-2



รูปที่ 4.24 ผลลัพธ์ส่งข้อมูลแบบ UDP บน Hyper-V จาก VMS-1 ไปยัง VMS-3

เมื่อทำการจำลองเครื่อง VMS-1 เป็น Client และ VMS-2, VMS-3 เป็นเครื่อง Server ทำการทดสอบส่งข้อมูลแบบ UDP บน ในช่วงระยะเวลา 30 วินาที ซึ่งอยู่บน Hyper-V Hypervisor ผลการทดสอบทำให้ผลลัพธ์การส่งรับข้อมูลดังนี้

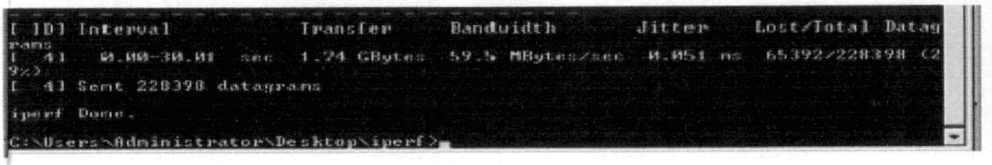
- จากรูปที่ 4.23 VMS-1 (Client) ฝั่งส่งสามารถส่งข้อมูลไปยัง VMS-2 ได้ที่ 1.75 GBytes หรือเท่ากับ 59.6 Mbytes/sec ซึ่งมีการส่ง Datagram ทั้งหมดเท่ากับ 228,743 Datagram เกิด Datagram ที่ Out-of-order (Loss) เท่ากับ 18,715 Datagram คิดเป็น 4.7% ของทั้งหมดที่สูญหายระหว่างการส่ง
- จากรูปที่ 4.24 VMS-1 (Client) ฝั่งส่งสามารถส่งข้อมูลไปยัง VMS-3 ได้ที่ 1.74 GBytes หรือเท่ากับ 59.4 Mbytes/sec ซึ่งมีการส่ง Datagram ทั้งหมดเท่ากับ 228,166 Datagram เกิด Datagram ที่ Out-of-order (Loss) เท่ากับ 26,322 Datagram คิดเป็น 12% ของทั้งหมดที่สูญหายระหว่างการส่ง



รูปที่ 4.25 ผลลัพธ์ส่งข้อมูลแบบ UDP บน OpenStack จาก VMS-1 ไปยัง VMS-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 ผลลัพธ์ส่งข้อมูลแบบ UDP บน OpenStack VMS-1 ไปยัง VMS-3

เมื่อทำการจำลองเครื่อง VMS-1 เป็น Client และ VMS-2, VMS-3 เป็นเครื่อง Server ทำการทดสอบส่งข้อมูลแบบ UDP บน ในช่วงระยะเวลา 30 วินาที ซึ่งอยู่บน OpenStack Hypervisor ผลการทดสอบทำให้ผลลัพธ์การส่งรับข้อมูลดังนี้

- จากรูปที่ 4.25 และ 4.25 VMS-1 (Client) ฝั่งส่งสามารถส่งข้อมูลไปยัง VMS-2 และ VMS-3 (Server) ให้ค่าผลลัพธ์ออกมาเท่ากันคือสามารถส่งข้อมูลได้ที่ 1.74 GBytes หรือเท่ากับ 59.5 Mbytes/sec ซึ่งมีการส่ง datagram ทั้งหมดเท่ากับ 228,389 Datagram เกิด Datagram ที่ Out-of-order เท่ากับ 65,763 คิดเป็น 29% ของทั้งหมดที่สูญหายระหว่างการส่ง

4.5 เปรียบเทียบการลงทุนและค่าใช้จ่ายลิขสิทธิ์ License

ในการจัดหาอุปกรณ์ใหม่ที่รองรับการทำงานกับทรัพยากรเวอร์ชวลไลเซชั่นในระบบงานจริง จำเป็นต้องอัพเกรด Hardware และ เทคโนโลยีที่ใช้ในการบริหารจัดการ Server Farm จากเดิมที่อยู่ในรูปแบบ On-Premise เป็น Physical ทั้งหมด การจะลงทุนเพื่อเปลี่ยน Hardware และจัดหาจัดซื้อ License ของ Hypervisor ทั้ง VMware, OpenStack และ Hyper-V เกิดค่าใช้จ่ายที่ค่อนข้างสูง ถึงแม้จะมีการทดสอบประสิทธิภาพและสามารถตัดสินใจเลือกเทคโนโลยีที่เหมาะสมกับระบบงานแล้ว ค่าใช้จ่ายที่เกิดขึ้นถือเป็นส่วนที่สำคัญในการตัดสินใจ ดังนั้นการเปรียบเทียบการลงทุนด้านการจัดหาอุปกรณ์ Hardware และ Software ใหม่เพื่อให้รองรับกับการย้ายระบบงานจากเดิมที่ผู้ใช้บริการบริหารจัดการกันเองภายในองค์กรมาขึ้นบน Cloud ของ Service Provider ที่บริษัทอินเตอร์เน็ตประเทศไทย จำกัด มหาชน จึงมีการเปรียบเทียบการลงทุนเพื่อให้เห็นความแตกต่างที่ระยะเวลาการลงทุน 3 ปี เพื่อชี้แนะให้เห็นถึงความคุ้มค่าของการลดทุนและเปรียบเทียบเพื่อให้เห็นความแตกต่างระบบ Cloud สามารถช่วยลดต้นทุนของผู้ใช้บริการลงได้ประมาณกี่เปอร์เซ็นต์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.1 เปรียบเทียบการลงทุนใช้งาน VMware ระหว่างแบบ On premise และ Cloud Service Provider

ในการด้านการลงุทน Hardware และ License ที่จะใช้งานกับระบบหากเป็น On-Premise การลงทุน VMware ที่ระยะเวลา 3 ปี ตามรายละเอียดในรูปที่ 4.24 ข้อมูลด้าน On-Premise ผู้ใช้บริการจำเป็นต้องจัดหาอุปกรณ์ Hardware , License และเจ้าหน้าที่ที่มีความรู้เฉพาะด้านเพื่อดูแลระบบทั้งด้าน Network, Storage และ VMware รวมถึงการค่าใช้จ่ายที่ต้องขยาย Bandwidth เพื่อให้รองรับกับการใช้งานระบบงาน (3 Server) ด้วยต้นทุน On-Premis ที่ต้องจ่ายทั้งหมด 719,224 บาทต่อเดือน ซึ่งเมื่อเทียบกับการเช่าใช้บริการ INET VMware Cloud เป็นรายเดือน ต้นทุนที่ต้องจ่ายทั้งหมด 55,090 บาทต่อเดือน จากการเปรียบเทียบ Cloud Service แบบ VMware ในลักษณะเช่าใช้ช่วยให้องค์กรลดค่าใช้จ่ายที่ต้องลงทุนประมาณ 92.34%

Onpremis Investment 3 Year					
Hardware HA (on Data Center)	Y1	Y2	Y3	Total	Unit
Power Edge M630 blade Server + HA (3 Unit) price from dell.com	420,000	84,000	84,000	588,000	THB
Backup Server (1 Unit) price from dell.com	140,000	28,000	28,000	196,000	THB
External Stroage (10 TB)	250,000	50,000	50,000	350,000	THB
Switch +HA 2 Unit	600,000	120,000	120,000	840,000	THB
License Window(2 Prc) 4 Unit	140,000	28,000	28,000	196,000	THB
Software backup (Exec 15)	140,000	28,000	28,000	196,000	THB
Firewall fortigate 100D	180,000	36,000	36,000	252,000	THB
Switch layer 3 cisco 48 port (2 unit)	80,000	16,000	16,000	112,000	THB
VMware (VMware vSphere Enterprise Plus)	215760	43,152	43,152	302,064	THB
Total H/W	2,165,760.00	433,152.00	433,152.00	3,032,064.00	THB
Period				36.00	Month
Cost/Period				84,224.00	THB/Month
Operational Expense					
- Manpower 3Man(Datacenter)				90,000.00	THB/Month
- Electric City on IDC 16A				5,000.00	THB/Month
Infrastructure Expense					
- Bandwidth Domestics 40Gbps				200,000.00	THB/Month
- Bandwidth inter 200Mbps				340,000.00	THB/Month
Optional					
Grand Total				719,224.00	THB/Month
INET Enterprise Cloud Saving Cost(21042016)	92.340%				

INET VMware Cloud Service (3 Year Contract)			
Year 1			
Service			
public Cloud 3 VM : 8 vcpu,16 GB,1.5 TBSAS per unit			
INET Cloud Service		84,000.00	THB/Month
Included			
- Bandwith Domestics 40GB(share)			
- Bandwith International 200 Mbps (share)			
Include : license window +antivirus			
support 24x7			
Total Cost 1 Year		1,008,000.00	THB/Year
Year 2			
Cloud Service Discount 10%		75,600.00	THB/Month
Total Year 2		907,200.00	THB/Year
Year 3			
Cloud Service Discount 10 %		68,040.00	THB/Month
Total Year 3		816,480.00	THB/Year
Total For 3 Year		1,983,240.00	THB/ 3Year
Grand Total		55,090.00	THB/Month

รูปที่ 4.27 cost compare : VMware On-Premise VS Cloud Service Provider

4.5.2 เปรียบเทียบการลงทุนใช้งาน MS-Cloud ระหว่างแบบ On premise และ Cloud Service Provider

ในการด้านการลงุทน Hardware และ License ที่จะใช้งานกับระบบหากเป็น On-Premise การลงทุน Microsoft Cloud (MS-Cloud) ที่ระยะเวลา 3 ปี ตามรายละเอียดในรูปที่ 4.25 ข้อมูลด้าน On-Premise ผู้ใช้บริการจำเป็นต้องจัดหาอุปกรณ์ Hardware, License และเจ้าหน้าที่ที่มีความรู้เฉพาะด้านเพื่อดูแลระบบทั้งด้าน Network, Storage และ Micrsoft Cloud รวมถึงการค่าใช้จ่ายที่ต้องขยาย Bandwidth เพื่อให้รองรับกับการใช้งานระบบงาน (3 Server) ด้วยต้นทุน On-Premis ที่ต้องจ่ายทั้งหมด 716,444.22 บาทต่อเดือน ซึ่งเมื่อเทียบกับการเช่าใช้บริการ INET VMware Cloud เป็นรายเดือน ต้นทุนที่ต้องจ่ายทั้งหมด 41,317.50 บาทต่อเดือน จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบ Cloud Service แบบ MS-Cloud ในลักษณะเช่าใช้ช่วยให้องค์กรลดค่าใช้จ่ายที่ต้องลงทุนประมาณ 94.233%

Onpremis Investment 3 Year					
Hardware HA (on Data Center)	Y1	Y2	Y3	Total	Unit
Power Edge M630 blade Server + HA (3 Unit) price from dell.com	420,000	84,000	84,000	588,000	THB
Backup Server (1 Unit) price from dell.com	140,000	28,000	28,000	196,000	THB
External Stroage (10 TB)	250,000	50,000	50,000	350,000	THB
Switch +HA 2 Unit	600,000	120,000	120,000	840,000	THB
License Window(2 Prc) 4 Unit	140,000	28,000	28,000	196,000	THB
Software backup (Exec 15)	140,000	28,000	28,000	196,000	THB
Firewall fortigate 100D	180,000	36,000	36,000	252,000	THB
Switch layer 3 cisco 48 port (2 unit)	80,000	16,000	16,000	112,000	THB
Microsoft (System Center 2012 R2 Datacenter Edition)	144280	28,856	28,856	201,992	THB
Total H/W	2,094,280.00	418,856.00	418,856.00	2,931,992.00	THB
Period				36.00	Month
Cost/Period				81,444.22	THB/Month
Operational Expense					
- Manpower 3Man(Datacenter)				90,000.00	THB/Month
- Electric City on IDC 16A				5,000.00	THB/Month
Infrastructure Expense					
- Bandwidth Domestics 40Gbps				200,000.00	THB/Month
- Bandwidth inter 200Mbps				340,000.00	THB/Month
Optional					
Grand Total				716,277.88	THB/Month
INET Enterprise Cloud Saving Cost(21042016)	94.233%				

INET MS Cloud Service (3 Year Contract)			
Year 1			
Service			
public Cloud 3 VM : 8 vcpu,16 GB,1.5 TBSAS per unit			
INET Cloud Service		63,000.00	THB/Month
Included			
- Bandwidth Domestics 40GB(share)			
- Bandwith International 200 Mbps (share)			
include : license window +antivirus			
support 24x7			
Total Cost 1 Year		756,000.00	THB/Year
Year 2			
Cloud Service Discount 10%		56,700.00	THB/Month
Total Year 2		680,400.00	THB/Year
Year 3			
Cloud Service Discount 10 %		51,030.00	THB/Month
Total Year 3		612,360.00	THB/Year
Total For 3 Year		1,487,430.00	THB/ 3Year
Grand Total		41,317.50	THB/Month

รูปที่ 4.28 cost compare : MS Cloud On-Premise VS Cloud Service Provider

4.5.3 เปรียบเทียบการลงทุนใช้งาน OpenStack ระหว่างแบบ On premise และ Cloud Service Provider

ในการด้านการลงทุน Hardware และ License ที่จะใช้งานกับระบบหากเป็น On-Premise การลงทุน OpenStack ที่ระยะเวลา 3 ปี ตามรายละเอียดในรูปที่ 4.26 ข้อมูลด้าน On-Premise ผู้ใช้บริการจำเป็นต้องจัดหาอุปกรณ์ Hardware, License และเจ้าหน้าที่ที่มีความรู้เฉพาะด้านเพื่อดูแลระบบทั้งด้าน Network, Storage และ OpenStack รวมถึงการค่าใช้จ่ายที่ต้องขยาย Bandwidth เพื่อให้รองรับกับการใช้งานระบบงาน (3 Server) ด้วยต้นทุน On-Premis ที่ต้องจ่ายทั้งหมด 716,277.88 บาทต่อเดือน ซึ่งเมื่อเทียบกับการเช่าใช้บริการ INET OpenStack Cloud เป็นรายเดือน ต้นทุนที่ต้องจ่ายทั้งหมด 51,155 บาทต่อเดือน จากการเปรียบเทียบ Cloud Service แบบ OpenStack ที่เป็นลักษณะเช่าใช้ช่วยให้องค์กรลดค่าใช้จ่ายที่ต้องลงทุนประมาณ 92.85%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Onpremis Investment 3 Year					
Hardware HA (on Data Center)	Y1	Y2	Y3	Total	Unit
Power Edge M630 blade Server+HA (3 Unit) price from dell.com	420,000	84,000	84,000	588,000	THB
Backup Server (1 Unit) price from dell.com	140,000	28,000	28,000	196,000	THB
External Stroage (10 TB)	250,000	50,000	50,000	350,000	THB
Switch +HA 2 Unit	600,000	120,000	120,000	840,000	THB
License Window(2 Prc) 4 Unit	140,000	28,000	28,000	196,000	THB
Software backup (Exec 15)	140,000	28,000	28,000	196,000	THB
Firewall fortigate 100D	180,000	36,000	36,000	252,000	THB
Switch layer 3 cisco 48 port (2 unit)	80,000	16,000	16,000	112,000	THB
OpenStack License + subscription	140000	28,000	28,000	196,000	THB
Total H/W	2,090,000.00	418,000.00	418,000.00	2,926,000.00	THB
Period				36.00	Month
Cost/Period				81,277.78	THB/Month
Operational Expense					
- Manpower 3Man(Datacenter)				90,000.00	THB/Month
- Electric City on IDC 16A				5,000.00	THB/Month
Infrastructure Expense					
- Bandwidth Domestics 40Gbps				200,000.00	THB/Month
- Bandwidth inter200Mbps				340,000.00	THB/Month
Optional					
Grand Total				716,277.78	THB/Month
INET Enterprise Cloud Saving Cost(21042016)	92.858%				

INET OpenStack Cloud Service (3 Year Contract)			
Year 1			
Service			
public Cloud 3 VM : 8 vcpu,16 GB,1.5 TBSAS per unit			
INET Cloud Service		78,000.00	THB/Month
Included			
- Bandwidth Domestics 40GB(share)			
- Bandwith International 200 Mbps (share)			
include : license window +antivirus			
support 24x7			
Total Cost 1 Year		936,000.00	THB/Year
Year 2			
Cloud Service Discount 10%		70,200.00	THB/Month
Total Year 2		842,400.00	THB/Year
Year 3			
Cloud Service Discount 10 %		63,180.00	THB/Month
Total Year 3		758,160.00	THB/Year
Total For 3 Year		1,841,580.00	THB/ 3Year
Grand Total		51,155.00	THB/Month

รูปที่ 4.29 cost compare : OpenStack On-Premise VS Cloud Service Provider

จากรูปที่ 4.24-4.26 เมื่อทำการเปรียบเทียบต้นทุนทางด้าน Infrastructure ที่ต้องลงทุนเพื่อเตรียมระบบงานใหม่แบบ On-Premise เทียบกับ Package ของ Cloud ของบริษัทอินเตอร์เน็ตประเทศไทย จำกัด มหาชน รูปแบบบริการของ MS-Cloud สามารถช่วยลดต้นทุนของผู้ใช้บริการได้มากที่สุด ซึ่งช่วยประหยัดต้นทุนประมาณ 94.233% รองลงมาคือ OpenStack ช่วยประหยัดต้นทุนประมาณ 92.858% และสุดท้ายคือ VMware อยู่ที่ประมาณ 92.340% ถึงแม้ค่าใช้จ่ายในส่วนของ MS-Cloud จะช่วยลดต้นทุนองค์กรได้มากสุดแต่ผู้ใช้บริการเลือกที่จะนำระบบไปติดตั้งใช้งานบน VMware เนื่องจากผลการทดสอบที่ได้ VMware ให้ประสิทธิในด้านการประมวลผล CPU และการเข้าถึงหน่วยความจำได้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

โครงสร้างพื้นฐานของเทคโนโลยี Virtualization แบบ VMware, OpenStack และ Hyper-V โดยพื้นฐานมีรูปแบบโครงสร้างที่แตกต่างกัน แต่ผู้ใช้งานหรือองค์กรที่กำลังมองหาหรือต้องการปรับปรุงระบบงานเดิมที่ใช้งานเป็น On-Premise ขึ้นมาเป็น Cloud Computing สามารถเลือกใช้งานให้เกิดประสิทธิภาพได้อย่างเต็มที่โดยการนำระบบงานมารับบน Hypervisor ทำการทดสอบแล้ววัดประสิทธิภาพทางด้านโครงสร้างพื้นฐาน Infrastructure ร่วมกันกับผู้ให้บริการ เพื่อให้ระบบงานสามารถนำขึ้นมาอยู่บน Cloud Computing แล้วเกิดการใช้งานที่เต็มประสิทธิภาพมากที่สุด ถึงแม้ชื่อของเทคโนโลยีจะมีชื่อเรียกที่แตกต่างกัน แต่โครงสร้างการบริหารจัดการยังคงมีองค์ประกอบอยู่ 2 ส่วนหลักตามที่ได้กล่าวไว้ในบทที่ 3 คือส่วนของ System Center หรือ Controller Node ที่ใช้ในสำหรับการบริหารจัดการและส่วนที่สองคือ Compute Node ประกอบด้วย Compute, Virtual Network, Virtual Storage หากเป็นเทคโนโลยี VMware จะเรียก Center Management ว่า VCenter และ Compute Node ว่า ESX Server หากเป็นเทคโนโลยี Hyper-V เรียก Center Management ว่า System Center และ Compute Node ว่า Hyper-V หากเป็นเทคโนโลยี OpenStack เรียก Center Management ว่า Controller Node และเรียก Compute Node ว่า Compute Node ทั้ง 3 Hypervisor เป็นโครงสร้างที่สามารถนำมาปรับใช้กับทุกองค์กรที่ต้องการใช้งานเป็น Cloud ทั้งนี้ต้องตอบสนองทางด้านการพัฒนาเทคโนโลยีและช่วยลดต้นทุนขององค์กรได้มากที่สุด

จากผลการทดลองที่ได้ทำการศึกษาและสร้างสภาพแวดล้อมทดสอบตามขั้นตอนในบทที่ 4 เพื่อเปรียบเทียบประสิทธิภาพของ VMware, OpenStack และ Hyper-V Hypervisor จากผลการทดสอบโดยการใช้ Tool เพื่อวัดประสิทธิภาพนั้น ในด้านการประมวลผลของ Processor เมื่อเปรียบเทียบกันระหว่าง VMware, OpenStack และ Hyper-V ทุก Parameter ที่ทำการทดสอบไม่ว่าจะเป็นการประมวลผลผลที่เกี่ยวข้องกับคณิตศาสตร์ เลขทศนิยม จำนวนเฉพาะ การประมวลผลชุดคำสั่ง รวบถึงการเรียงลำดับตัวอักษร การประมวลผลบน VMware สามารถประมวลผลได้ดีกว่า Open Stack ในขณะที่ Hyper-V ให้ค่าที่ต่ำที่สุดและในด้านการทดสอบการอ่านเขียนหน่วยความจำทั้งการอ่านและเขียนหน่วยความจำแบบใช้หน่วยความจำ Cashed และไม่ใช้หน่วยความจำ Cashed ถึงแม้จะมีการปรับขนาดของ Memory เพิ่มขึ้นและทำการวัดค่าผลลัพธ์ที่ออกมา VMware ก็ยังคงให้ผลลัพธ์ออกมาได้สูงที่สุดเมื่อเทียบกับ OpenStack และ Hyper-V และในด้านการทดสอบการอ่านเขียนข้อมูลดิสก์โครงสร้างของ OpenStack มีการแยกการบริหารจัดการในส่วนของ Object, Block, File System ที่ดีกว่าโครงสร้างของ VMware และ Hyper-V ผลลัพธ์ที่ได้จากการทดสอบประสิทธิภาพในการอ่านเขียนข้อมูลทั้งแบบ Sequential และ Random กลับให้ค่าผลลัพธ์ในส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ I/O per Second ที่สูงกว่า VMware ในขณะที่ Hyper-V กลับให้ค่าผลลัพธ์ที่ต่ำกว่ามากเมื่อเทียบกับอีก 2 Hypervisor และถึงแม้จะปรับขนาดของ Disk ที่ใช้ในการทดสอบแล้วก็ตาม นอกจากนี้ในด้านการทดสอบการส่งข้อมูลภายในเครือข่าย (Private Network) รูปแบบการส่งข้อมูลแบบ TCP ผลลัพธ์ที่ได้จากการทดสอบ OpenStack สามารถ Transfer ข้อมูลได้สูงสุดเมื่อเทียบกับ VMware และ Hyper-V และการส่งข้อมูลแบบ UDP กลับให้ค่าผลลัพธ์ที่ไม่แตกต่างกันมากนัก ทั้งนี้ผลการทดสอบที่เกิดขึ้นเป็นเพียงการทดสอบบนโครงสร้าง IAAS ของ Service ที่ผู้ให้บริการจัดเตรียมขึ้นเท่านั้น ซึ่งหากมีการนำระบบไปทดสอบบนโครงสร้าง IASS ของผู้ให้บริการรายอื่น อาจได้ผลการทดสอบที่แตกต่างกันขึ้นอยู่กับ Hardware Specification และ เทคโนโลยี Virtualization ที่ผู้ให้บริการรายนั้นให้บริการ

ปัจจุบันระบบงานที่นำมาเป็นกรณีศึกษาในครั้งนี้ทางผู้ใช้บริการได้มีการตัดสินขึ้นระบบใช้งานจริงบน VMware เนื่องจากระบบงานต้องการประสิทธิภาพของทรัพยากรในส่วนของการทำงาน CPU และ Memory ที่มีประสิทธิภาพที่ดีที่สุดเมื่อเทียบกับ OpenStack และ Hyper-V ในส่วนของ Disk ระบบงานไม่ต้องการ IOPs ที่สูงดังนั้นในอนาคตหากระบบต้องการประสิทธิภาพที่เพิ่มขึ้น สามารถปรับเปลี่ยนชนิดของ Disk จากเดิมที่เป็น SAS เปลี่ยนเป็นการใช้งาน SSD แทน นอกจากโครงสร้างพื้นฐาน IAAS ที่ดีซึ่งมีผลต่อประสิทธิภาพการทำงานของระบบแล้ว การที่จะตัดสินใจย้ายระบบงานขึ้นมาเป็น Cloud Computing อีกหนึ่งปัจจัยที่มีผลต่อการตัดสินใจคือการลงทุนช่วยลดต้นทุนมากน้อยเพียงใด หากลองเปรียบเทียบค่าใช้จ่ายลิขสิทธิ์ที่เกิดขึ้นของทั้ง 3 Hypervisor การลงทุน On-Premise (Private Cloud) ทั้ง VMware และ Hyper-V มีค่าใช้จ่ายในส่วนของลิขสิทธิ์ License โดย VMware มีค่าใช้จ่ายรวม Support 1 ปี ประมาณ 215,760 บาทต่อปี ส่วน Hyper-V มีค่าใช้จ่ายรวม Support 1 ปี ประมาณ 144,280 บาทต่อปี แต่ในส่วนของ OpenStack กลับเป็น Open Source ที่ใช้งานได้ฟรีไม่มีค่าใช้จ่ายยกเว้นหากผู้ใช้งานต้องการ Support หรือ Manage Service ซึ่งจะมีค่าใช้จ่ายประมาณ $4000 (1$= 35 บาท) หรือ 140,000บาทต่อปี ซึ่งสามารถเปรียบเทียบ License Price ได้ตามรางที่ 5.1

ตารางที่ 5. 1 ค่าใช้จ่ายลิขสิทธิ์หากเป็น Private Cloud

Virtualization License	**ค่าใช้จ่ายลิขสิทธิ์ + MA 1 ปี**
VMware (VMware vSphere Enterprise Plus)	215,760 บาท
OpenStack	ฟรีไม่มีค่าใช้จ่าย + Subscription (140,000 บาท)
Microsoft (System Center 2012 R2 Datacenter Edition)	144,280 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังมีต้นทุนในด้าน Infrastructure ของระบบงานหากที่ต้องลงทุน On-Premise เพื่อให้รองรับระบบงานใหม่ตามที่ได้อธิบายรายละเอียดการเปรียบเทียบไว้ในหัวข้อ 4.5.1-4.5.3 แต่หากเป็นรูปแบบการย้ายระบบมาเป็น Public Cloud จะกลายเป็นลักษณะการเช่าใช้งานรายเดือนแทนซึ่งเท่ากับว่าผู้ใช้บริการไม่ต้องลงทุนค่าใช้จ่ายทั้ง Infrastructure และค่าใช้จ่าย License ของ Virtualization รวมถึงการ Operation ในครั้งเดียวเพราะบางครั้งการลงทุนด้วยระยะเวลา 3 หรือ 5 ปี บางระบบงานอาจจะต้องมีการเปลี่ยน Hardware ก่อนสิ้นสุดระยะการใช้งานเพื่อรองรับการขยายของระบบงานทำให้ไม่เกิดความคุ้มทุนในการใช้งาน ในขณะที่การเช่าใช้บริการสามารถปรับทรัพยากรที่ค่อนข้างยืดหยุ่น นอกจากนี้ผู้ให้บริการยังมีหน้าที่ในการอัพเดตเวอร์ชั่นของเทคโนโลยี Virtualization โดยผู้ใช้งานไม่จำเป็นต้องเสียค่าใช้จ่ายเพิ่มเติม ดังนั้นผลลัพธ์ที่ได้จากการศึกษาและทดลองในครั้งนี้จึงเป็นประโยชน์สำหรับผู้ใช้บริการนำไปเป็นข้อมูลสนับสนุนการตัดสินใจเลือกโครงสร้างพื้นฐาน IAAS ที่เหมาะสมและตัดสินใจย้ายระบบงานมายัง Cloud Provider เป็น Public Cloud ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Amazon. **Elastic Compute Cloud User Guide.** [Online]. Available: http://aws.amazon.com/.

Cisco. **Cisco UCS B200 M3 Blade Server** [Online]. Available: http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-b200-m3-blade-server/index.html#.

Businessinsider Clearest Explanation Yet of SaaS / PaaS /IaaS.http://www.businessinsider.com/10-most-important-in-cloud-computing-2013-4?op=1#ixzz2RV1Gjiw2.

Microsoft. **Competitive Advantages of Window Server 2012 R2 Hyper-V Over VMware vSphere 5.5.** [Online]. Available: http://Microsoft.com/.

Dimpi Rani, Rajiv Kumar Ranjan. June 2014. **A Comparative Study of SaaS, PaaS and IaaS in Cloud Computing.** International Journal of Advanced Research in Computer Science and Software Engineering .Volume 4, Issue: 6 456-461.

Foster, Y. Zhau, R. Ioan, and S. Lu. 2008. **Cloud computing and grid computing 360-degree compared.** Grid Computing Environment Workshop, GCE'08. IEEE.

Gillam L and O'Loughlin J., Teaching Clouds: Lessons Taught and Lessons Learnt, **In Cloud Computing for Teaching and Learning: Strategies for Design and Implementation,** \IGI Global.

Padala, P. 2013.. **Resource Management in VMware Powered Cloud: Concepts and Techniques,** Parallel & Distributed Processing (IPDPS)", 2013 IEEE.

Foster, C. Kesselman, and S. Tuecke. **The Anatomy of the Grid: Enabling Scalable Virtual Organizations.** International Journal of High Performance Computing Applications, 15(3), 2001.

Jianfeng Yang, Zhibin Chen. 2010. **Cloud Computing Research and Security Issues.** Vol 978-1-4244-5392-4/10, IEEE.

Microsoft. **.Introduce Microsoft Hyper-V.**[Online]. Available: https://mva.microsoft.com/en-US/training-courses/-microsoft-hyper-v-8626?l=NyFJvMF1_6204984382

Ron Oglesby, Scott Herold. 2005. **VMware Technical Design Guide,** VMware vSphere 4 – ESX and venter Server[Online]. Available:https://pubs.vmware.com/vsphere-4sxvcenter/index.jsp#com.vmware.vsphere.intro.doc_41/c_physical_topology_of_vi_datacenter.hml

Shalmali Suhas Sahasrabudhe, Shilpa S. Sonawani. 2014. **Comparing Open stack and VMwa-**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

re, International Conference on Advances in Electronics, Computers and Comm uni-cation (ICAECC), Vol 978-1-4799-5496-4/14, IEEE.

VMware. **ESXi era: Using ESXi to replace ESX – Architecture Comparison Part 2** [Online]. Available: https://geeksilver.wordpress.com/2010/06/07/esxi-era-using-esxi-to-replace-esx-%E2%80%93-architecture-comparison-part-2/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวพิชญา พิทักษ์วงศ์
วันเดือนปีเกิด	27 ธันวาคม 2531
สถานที่เกิด	นครศรีธรรมราช
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตร์บัณฑิต (คอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยสงขลานครินทร์
ปีที่สำเร็จการศึกษา	2554
การทำงาน	วิศวกรอาวุโส ส่วนงาน Cloud Solution

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้