แพลตฟอร์มเพื่อการจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์

TIRAMISU: SOFTWARE DEFINED DATA PLATFORM

รินทร์ลภัส เลิศธนกุลพงษ์

ณัฐพงศ์ อมรบัญชรเวช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แพลตฟอร์มเพื่อการจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์

TIRAMISU: SOFTWARE DEFINED DATA PLATFORM

รินทร์ลภัส เลิศธนกุลพงษ์

ณัฐพงศ์ อมรบัญชรเวช

สรรหมู่.....
เลขทะเบียน 144394
วัน,เดือน,ปี 24 พ.ย. 2559

b.12818641
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แพลตฟอร์มเพื่อการจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์

TIRAMISU: SOFTWARE DEFINED DATA PLATFORM

ผู้จัดทำ

1. นายณัฐพงศ์ อมรบัญชรเวช รหัสนักศึกษา 55010371

2. นางสาวรินทร์ลภัส เลิศธนกุลพงษ์ รหัสนักศึกษา 55011049

_____ อาจารย์ที่ปรึกษา

(ดร.อรทัย สังข์เพ็ชร)

_____ อาจารย์ที่ปรึกษา

(ดร.อักฤทธิ์ สังข์เพ็ชร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพลตฟอร์มเพื่อการจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์

นายณัฐพงศ์	อมรบัญชรเวช	55010371
นางสาวรินทร์ลภัส	เลิศธนกุลพงษ์	55011049
ดร.อรทัย	สังข์เพ็ชร	อาจารย์ที่ปรึกษา
ดร.อักฤทธิ์	สังข์เพ็ชร	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2558

บทคัดย่อ

แอปพลิเคชันในปัจจุบันมีความต้องการใช้ข้อมูลที่แตกต่างกันทั้งในเชิงรูปแบบและคุณสมบัติของการจัดเก็บข้อมูล และเพื่อให้แอปพลิเคชันทำงานได้อย่างมีประสิทธิภาพมากที่สุด ซึ่งอุปกรณ์การจัดเก็บข้อมูลเป็นปัจจัยสำคัญที่ส่งผลต่อประสิทธิภาพการทำงานของแอปพลิเคชัน ซึ่งอาจจะถูกในรูปของความต้องการใช้งานอุปกรณ์การจัดเก็บข้อมูลของแอปพลิเคชัน เช่น ระยะเวลาการตอบสนอง (latency) และ จำนวนครั้งในการเรียกใช้อุปกรณ์การจัดเก็บข้อมูล (IOPS) แต่โดยปกติแล้วผู้พัฒนาแอปพลิเคชันมักจะไม่มีความเชี่ยวชาญและความรู้เชิงลึกเกี่ยวกับอุปกรณ์การจัดเก็บข้อมูลทำให้เป็นไปได้ยากที่นักพัฒนาแอปพลิเคชันจะเลือกอุปกรณ์การจัดเก็บข้อมูลที่เหมาะสม

โครงงานจึงได้พัฒนาแพลตฟอร์มเพื่อช่วยนักพัฒนาเลือกอุปกรณ์การจัดเก็บข้อมูลให้เหมาะสมกับคุณภาพของแอปพลิเคชันและงบประมาณที่ผู้ใช้กำหนด โดยจะมีอุปกรณ์จัดเก็บข้อมูลหลายชนิดทำงานร่วมกันโดยใช้เทคโนโลยีการจัดเก็บข้อมูลแบบกระจาย (Distributed storage system)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tiramisu: Software Defined Data Platform

Mr. Nattapong	Amornbunchornvej	55010371
Ms. Rinlapat	Loetthanakulpong	55011049
Dr. Orathai	Sangpetch	Advisor
Dr. Akkarit	Sangpetch	Co-Advisor

Academic Year 2015

ABSTRACT

Different applications have various requirements in terms of data usage and characteristics. In order to provide the acceptable level of application service quality, developers need to understand and appropriately select the underlying data storage technology for such requirements. As a result, developers may need to spend a significant amount of time learning unfamiliar storage technology.

This project aims to develop Tiramisu, a software-defined data platform, which can automatically translate the application requirements to the data storage requirements and calculate this metric to find appropriate storage such as storage that can give performance nearly application request. We have various type of storage work together on distributed storage system technology. Tiramisu then provisions the datastore according to the translated storage characteristics.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีด้วยความช่วยเหลือจากหลายฝ่ายทั้งในทางตรงและทางอ้อม โครงงานฉบับนี้จะสำเร็จลงไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้

ทุนอุดหนุนโครงการ “แพลตฟอร์มเพื่อการจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์” จากการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์ แห่งประเทศไทย ครั้งที่ 18 จากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

อาจารย์ที่ปรึกษาทั้งสองท่านคือ ดร.อรทัย สังข์เพ็ชร และ ดร.อักฤทธิ์ สังข์เพ็ชร เป็นผู้ที่ให้คำแนะนำ ปรึกษา และให้ความช่วยเหลือตลอดการทำโครงงาน ซึ่งทำให้การทำงานต่างๆเป็นไปได้อย่างราบรื่น

อาจารย์และบุคลากรต่างๆในสาขาวิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำและสั่งสอนความรู้ต่างๆมาตลอด รวมถึงห้องแลป SAIG (Software and Application Interest Group Comunity) และ SOUP (System Operations Usability Parallel computing) ที่ได้เอื้อเฟื้อสถานที่ในการทำวิจัยและพัฒนาโครงงาน

ในท้ายที่สุดนี้ขอขอบคุณบิดา มารดา และครอบครัวที่ได้เลี้ยงดูและสั่งสอน พร้อมทั้งให้โอกาสในการศึกษา และให้กำลังใจเสมอมา

ณัฐพงศ์ อมรบัญชรเวช

รินทร์ลภัส เลิศธนกุลพงษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ประโยชน์ที่คาดว่าจะได้รับ 2

1.4 ขอบเขตของโครงงาน 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 3

2.1 Distributed Storage System 3

2.2 การประมวลผลบนกลุ่มเมฆ (Cloud Computing) 4

2.3 เทคโนโลยีเวอร์ชวลไลเซชั่น (Virtualization Technology) 4

2.4 การจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์ (Software-Defined Storage) 5

2.5 เทคโนโลยีการติดตามพฤติกรรมของแอปพลิเคชัน 6

2.6 เทคโนโลยีการใช้งานอุปกรณ์จัดเก็บข้อมูลของระบบปฏิบัติการ Linux 6

2.7 งานที่เกี่ยวข้อง 7

บทที่ 3 การวิเคราะห์และออกแบบระบบ 8

3.1 ภาพรวมของระบบ 8

3.2 โครงสร้างในการพัฒนาระบบ 10

3.3 แผนผังลำดับการทำงาน 12

3.4 แบบจำลองในการแนะนำอุปกรณ์จัดเก็บข้อมูล 13

3.5 ส่วนแสดงผล 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 การทดลองและผลการทดลอง 25
4.1 การทดสอบเกี่ยวกับความถูกต้องของแบบจำลอง 25
4.2 ประสิทธิภาพการทำงานของแบบจำลอง 30

บทที่ 5 บทสรุปและข้อเสนอแนะ 32
5.1 บทสรุปของโครงงาน 32
5.2 แนวทางในการพัฒนาต่อในอนาคต 32

เอกสารอ้างอิง 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง หน้า

3.1 สัญลักษณ์ภายในกราฟแบบจำลอง the cube 13

3.2 สัญลักษณ์ภายในกราฟแบบจำลอง line model 17

3.3 การทำงานร่วมกันระหว่าง 2 แบบจำลอง 18

4.1 กล่องความต้องการใช้งานของผู้ใช้ที่สร้างขึ้นจากค่าความต้องการที่ผู้ใช้ป้อนเข้าสู่ระบบผ่านทางหน้าเว็บ 26

4.2 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 1 26

4.3 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 2 27

4.4 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 3.1 28

4.5 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 3.2 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป หน้า

1.1 ประสิทธิภาพโดยทั่วไปของอุปกรณ์จัดเก็บข้อมูลแต่ละชนิด 1

2.1 ชั้นการทำงานของ Ceph RBD 3

2.2 ข้อแตกต่างก่อนและหลังการใช้งานเทคโนโลยีเวอร์ชวลไลเซชั่น 5

2.3 ชั้นการทำงานของการติดต่อระหว่างแอปพลิเคชันและไฟล์ซิสเต็ม (File System) 6

3.1 ภาพรวมการทำงานของระบบ 8

3.2 แผนภาพการเชื่อมต่อระหว่างแต่ละส่วนของระบบ 10

3.3 แผนผังลำดับการทำงาน (Flowchart) ของระบบทั้งหมด 12

3.4 กราฟตัวอย่างการสร้างกล่อง requirements และ plot จุด performance storage 14

3.5 แบบจำลอง the cube กรณีที่ 1 15

3.7 แบบจำลอง the cube กรณีที่ 3 16

3.8 ตัวอย่างของเส้นจำนวนกรณีที่ 1 17

3.9 ตัวอย่างของเส้นจำนวนกรณีที่ 2 17

3.12 หน้าเข้าสู่ระบบของเว็บ Tiramisu 19

3.13 หน้าหลักของเว็บไซต์ Tiramisu 19

3.14 หน้าจอแสดงข้อความเตือนผู้ใช้หลังจากกดปุ่ม Delete 20

3.15 แบบฟอร์มสำหรับสร้างเครื่องจำลอง 21

3.16 หน้าจอหลังจากสร้าง VM สำเร็จ 21

3.17 หน้าแก้ไข Requirements 22

3.18 ระบบนำค่า requirements ที่ผู้ใช้ป้อนเข้ามาใหม่ไปทำการคำนวณโดยแบบจำลอง และแนะนำ Storage ที่เหมาะสม 23

3.19 นำค่า requirements ที่ผู้ใช้ป้อนเข้ามาใหม่ไปทำการคำนวณโดยแบบจำลอง ได้ว่าไม่ควรย้าย Storage 23

3.20 สี่เหลี่ยมสีแดงแสดงข้อความแจ้งเตือนผู้ใช้ 24

3.21 หน้าจอแสดงรายละเอียด VM เมื่อ ผู้ใช้ปิดการแจ้งเตือน 24

4.1 Requirements ของผู้ใช้ที่กำหนดขึ้นเพื่อการทดลอง 25

4.2 ผลการคำนวณจากแบบจำลอง 26

4.3 ผลการคำนวณจากแบบจำลอง 27

4.4 รายละเอียดของ VM จะเห็นว่า Current storage ตรงกับ Appropriate storage 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป **หน้า**

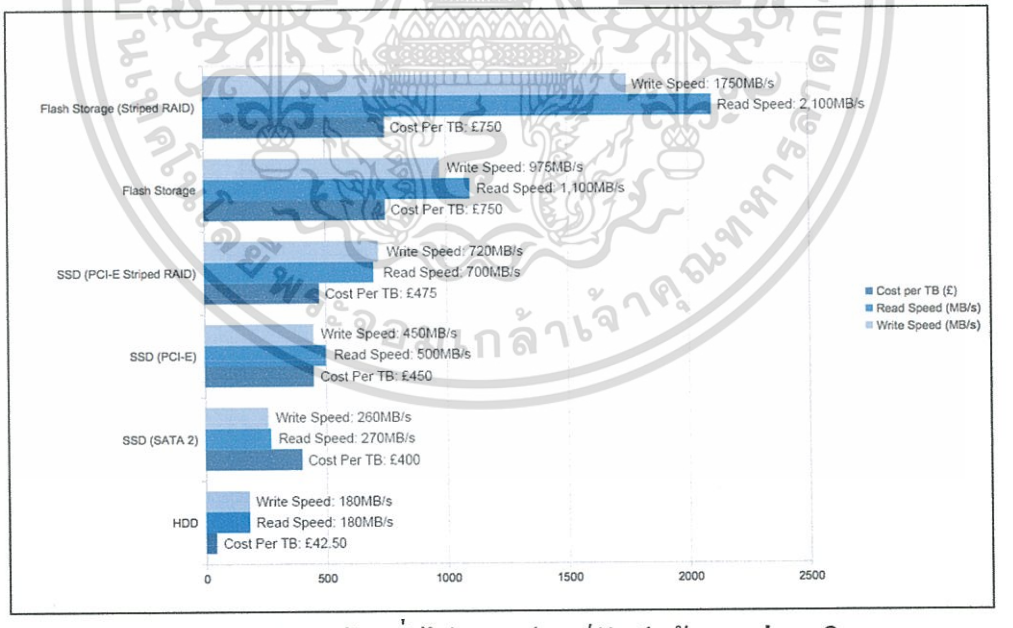
4.5 รายละเอียดของ VM จะเห็นว่า Current storage ไม่ตรงกับ Appropriate storage
และมีการแสดงการแจ้งเตือนเพื่อให้ย้ายไปยังอุปกรณ์จัดเก็บข้อมูลที่เหมาะสม 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

เนื่องจากในปัจจุบันการสร้างแอปพลิเคชันมีแนวโน้มที่จะเพิ่มความซับซ้อนมากขึ้นเรื่อย ๆ และหนึ่งในปัจจัยสำคัญในการทำงานที่ต้องมีอย่างหนึ่งก็คือส่วนจัดเก็บข้อมูล ซึ่งเป็นสิ่งใกล้ตัวที่จำเป็นและมีมากมายหลากหลายชนิดให้เลือกใช้ และส่งผลต่อประสิทธิภาพการทำงานของแอปพลิเคชัน ดังแสดงในรูป 1.1 แต่ผู้ใช้จำนวนมากไม่รู้ว่าในเวลาที่แอปพลิเคชันทำงานอยู่ต้องการอุปกรณ์จัดเก็บข้อมูลระดับไหนถึงจะเพียงพอต่อความต้องการโดยไม่เกินงบประมาณหรือเกิดความสิ้นเปลืองต่อทรัพยากรที่มีให้น้อยที่สุด ทำให้บางครั้งอุปกรณ์จัดเก็บข้อมูลทำให้แอปพลิเคชันทำงานได้ช้าลงเพราะมีอุปกรณ์จัดเก็บข้อมูลที่ไม่เหมาะทำให้เพิ่มเวลาที่ใช้ในการเรียกข้อมูล จากความซับซ้อนและพฤติกรรมของแอปพลิเคชันที่อาจจะมีการเปลี่ยนแปลงตลอดเวลาอันเป็นปัจจัยในการเลือกอุปกรณ์จัดเก็บข้อมูลนี้เอง ที่เป็นปัญหาในการพัฒนาแอปพลิเคชัน โดยเฉพาะกระบวนการพัฒนาแอปพลิเคชันที่ทำงานในระบบขนาดใหญ่ ที่ต้องวางแผนให้สามารถรองรับการขยายหรือปรับขนาดระบบได้



รูป 1.1 ประสิทธิภาพโดยทั่วไปของอุปกรณ์จัดเก็บข้อมูลแต่ละชนิด [1]

โครงงานนี้จึงพัฒนามาเพื่อเลือกอุปกรณ์จัดเก็บข้อมูลที่มีความสามารถเพียงพอต่อความต้องการของผู้ใช้ เพื่อให้ผู้ใช้ไม่ต้องศึกษาและเลือกอุปกรณ์ที่คุ้มที่สุดเอง ทำให้อุปกรณ์จัดเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลช่วยส่งเสริมการทำงานของแอปพลิเคชันให้สามารถทำงานได้รวดเร็วมากยิ่งขึ้น โดยการลดเวลาในการเขียนหรืออ่านข้อมูลของแอปพลิเคชัน

1.2 วัตถุประสงค์ของโครงงาน

1) สร้างระบบที่ใช้แบบจำลองที่สร้างโดยให้ผู้ใช้เลือกคุณภาพการจัดเก็บข้อมูลที่ต้องการ ได้แก่ จำนวนครั้งในการเรียกใช้อุปกรณ์การจัดเก็บข้อมูลต่อวินาที (IOPS) และ ระยะเวลาในการตอบสนอง (latency) ที่อยู่ภายในงบประมาณที่กำหนด โดยแบบจำลองจะนำค่าที่ผู้ใช้ให้ มาทำการประมวลผลและเลือกอุปกรณ์จัดเก็บข้อมูลที่ใกล้เคียงกับที่คุณภาพที่ผู้ใช้ต้องการ
2) แบบจำลองที่ถูกสร้างจะสามารถปรับเปลี่ยนอุปกรณ์จัดเก็บข้อมูลตามการเปลี่ยนแปลงพฤติกรรมของแอปพลิเคชัน โดยมีโมเดลทางคณิตศาสตร์ที่คิดค้นขึ้นเพื่อใช้ในการเลือก

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) นักพัฒนาไม่ต้องมีความรู้ในเชิงลึก ก็สามารถได้คุณภาพที่ต้องการ ในงบประมาณที่สามารถรับได้
2) มีเครื่องมือที่ช่วยในการจัดเก็บข้อมูลให้ตรงตามความต้องการของแอปพลิเคชัน
3) มีความเข้าใจเรื่องการทำงานของอุปกรณ์จัดเก็บข้อมูลมากขึ้น
4) มีความรู้เรื่องกระบวนการจัดเก็บข้อมูลของแอปพลิเคชันมากขึ้น

1.4 ขอบเขตของโครงงาน

1) เครื่องจำลอง (Virtual Machine) ที่สร้างขึ้น 1 เครื่อง จะใช้ในการทำงานของแอปพลิเคชัน 1 แอปพลิเคชันเท่านั้น
2) ระบบสามารถย้ายเครื่องจำลองที่มีแอปพลิเคชันอยู่ ไปไว้บนอุปกรณ์จัดเก็บข้อมูลชนิดใหม่ที่ระบบเลือกมาได้ เพื่อให้มีประสิทธิภาพการทำงานดีขึ้น โดยในขณะที่ทำการย้ายเครื่องจำลองจะถูกหยุดการทำงานและไม่สามารถใช้งานได้ ผู้ใช้จะเห็นสถานะของเครื่องจำลองเป็น Down ในขณะที่กำลังย้าย และจะเปลี่ยนเป็น Up ทันทีที่การย้ายเสร็จสมบูรณ์ โดยระยะเวลาที่เครื่องไม่สามารถใช้งานได้ จะขึ้นอยู่กับขนาดข้อมูลภายในเครื่องจำลอง
3) ไม่สามารถใช้ร่วมกับอุปกรณ์จัดเก็บข้อมูลประเภท Object storage ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

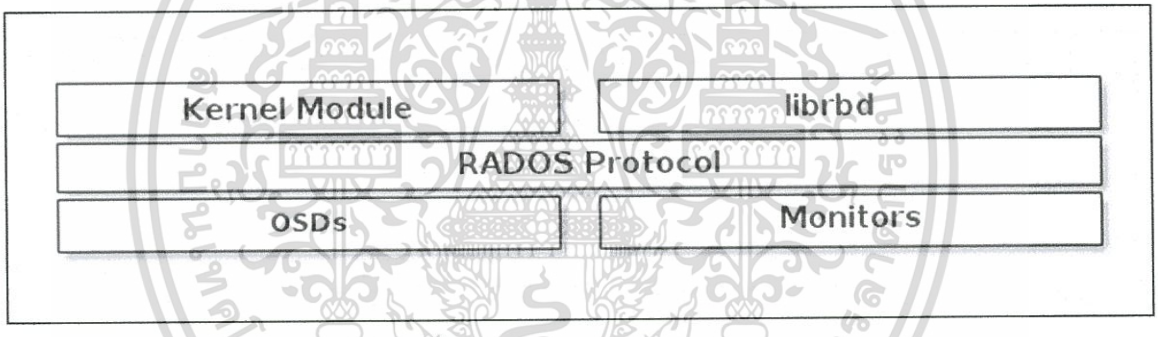
บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Distributed Storage System

คือการที่สามารถรวมอุปกรณ์จัดเก็บข้อมูลจากหลากหลายที่หลากหลายชนิดมาใช้งานเป็นอุปกรณ์ขนาดใหญ่อันเดียวได้ โดยไม่จำเป็นต้องอยู่บนเครื่องเดียวกัน ซึ่งจะมีข้อดีคือ

1) ป้องกันข้อมูลสูญหายในกรณีที่อุปกรณ์บางส่วนเกิดความเสียหาย เนื่องจากมีการทำสำเนาข้อมูลไว้ในเครื่องอื่น ๆ ภายในระบบ
2) แบ่งเบาภาระงานภายในระบบ เนื่องจากมีคอมพิวเตอร์หลายเครื่องช่วยกันทำงาน
3) อุปกรณ์จัดเก็บข้อมูลชนิดเดียวกันสามารถอยู่คนละเครื่องได้

ในโครงงานนี้จะเลือกใช้ Ceph RBD (RADOS Block Device) ในการทำ Distributed Storage System ซึ่งเหมาะกับการใช้งานกับเครื่องคอมพิวเตอร์จำลอง มีชั้นการทำงานดังนี้ [4]



รูป 2.1 ชั้นการทำงานของ Ceph RBD

Ceph RBD จะทำงานบน Ceph storage cluster ซึ่งประกอบด้วย

1) OSDs (Object Storage Daemon) มีหน้าที่จัดเก็บข้อมูลลงบนอุปกรณ์จัดเก็บข้อมูลของเครื่องนั้นๆ และให้สิทธิ์การเข้าถึงข้อมูลนี้ผ่านทางเนตเวิร์กได้
2) Monitors เป็นส่วนควบคุมทุกอย่างใน cluster และเป็นส่วนเชื่อมต่อกับเครื่องภายนอก

โดยทั้ง OSDs และ Monitor จะถูกครอบด้วย RADOS (Reliable Autonomic Distributed Object Store) Protocol และถูกเรียกใช้งานผ่านทาง librbd (ตามรูป 2.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การประมวลผลบนกลุ่มเมฆ (Cloud Computing)

Cloud computing หรือการประมวลผลแบบกลุ่มเมฆ เป็นแนวคิดในการให้บริการคอมพิวเตอร์หรือบริการต่างๆให้กับผู้ใช้งาน โดยผู้ให้บริการเป็นผู้รวบรวม ติดตั้ง และจัดการทรัพยากรต่างๆ เช่น เครือข่าย เครื่องแม่ข่าย หน่วยเก็บข้อมูล แอปพลิเคชัน และนำทรัพยากรต่างๆเหล่านี้ไปให้บริการกับผู้ใช้งาน ข้อดีของการที่ทรัพยากรต่างๆถูกรวมและจัดการ โดยผู้ให้บริการเพียงฝ่ายเดียวนั้นคือทำให้ผู้ใช้งานไม่ต้องรับผิดชอบค่าใช้จ่ายในการจัดซื้อและบำรุงรักษาเครื่องหรือบริการต่างๆเหล่านั้นเอง

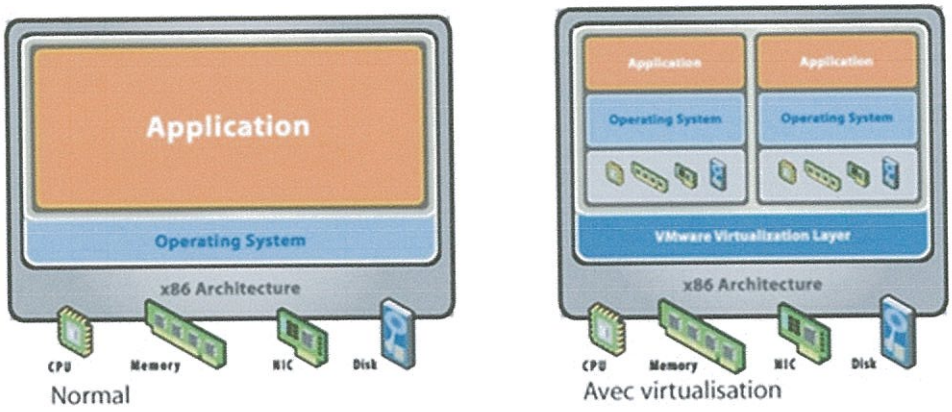
การให้บริการต่างๆบน cloud นั้นถูกแยกออกเป็น 3 ประเภท ดังนี้ [5]

1) Software as a Service หรือ SaaS เป็นการเปิดให้ผู้ใช้งานเข้าใช้งานซอฟต์แวร์หรือแอปพลิเคชันต่างๆตัวอย่างเช่น Google Doc เป็นต้น
2) Platform as a Service หรือ PaaS เป็นการเปิดให้ผู้ใช้งานเช่าและเข้าใช้งานเครื่องคอมพิวเตอร์เสมือนเพื่อพัฒนาหรือติดตั้ง Web Application โดยไม่จำเป็นต้องจัดการสภาพแวดล้อมต่างๆเอง
3) Infrastructure as a Service หรือ IaaS ที่ให้อิสระกับผู้เข้าใช้งานในการจัดสรรทรัพยากรต่างๆของเครื่องคอมพิวเตอร์เสมือน

2.3 เทคโนโลยีเวอร์ชวลไลเซชั่น (Virtualization Technology)

เทคโนโลยีเวอร์ชวลไลเซชั่นเป็นเทคโนโลยีที่ใช้ในการสร้างระบบเสมือนขึ้นมา เช่น เครื่องคอมพิวเตอร์, หน่วยประมวลผล (CPU), ระบบปฏิบัติการ (OS), อุปกรณ์จัดเก็บข้อมูล หรือเครือข่ายในการเชื่อมต่อ เป็นต้น โดยระบบเสมือนเหล่านี้ทำงานอยู่บนฮาร์ดแวร์เดียวกัน โดยใช้งานซอฟต์แวร์เวอร์ชวลไลเซชั่น (virtualization software) ในการจัดการทรัพยากรด้านฮาร์ดแวร์ เปรียบเทียบดังรูป 2.2 ด้านซ้ายเป็นเครื่องปกติ ด้านขวาที่ใช้เทคโนโลยีเวอร์ชวลไลเซชั่น จะมีการแบ่งฮาร์ดแวร์เสมือนให้กับเครื่องจำลองแต่ละเครื่องที่ใช้ฮาร์ดแวร์จริงร่วมกัน [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.2 ข้อแตกต่างก่อนและหลังการใช้งานเทคโนโลยีเวอร์ชวลไลเซชั่น

สาเหตุที่เทคโนโลยีเวอร์ชวลไลเซชั่นเข้ามามีบทบาทมากขึ้น มาจากหลายๆเหตุผล ดังนี้

1) ทำให้ผู้ใช้งานหลายคนสามารถใช้งานเครื่องคอมพิวเตอร์ร่วมกันได้ซึ่งทำให้ค่าใช้จ่ายต่างๆ ทั้งในการจัดซื้อและบำรุงรักษาเครื่องลดลงอย่างมาก
2) สามารถจำลองฮาร์ดแวร์หรือระบบปฏิบัติการได้หลากหลายรูปแบบ
3) การบริหารจัดการต่างๆทำได้ง่ายขึ้นเนื่องจากการใช้งานของผู้ใช้ เช่น การติดตั้งแอปพลิเคชัน หรือ เครื่องจำลองไม่ถูกจำกัดและผูกกับฮาร์ดแวร์ที่ใช้งานอีกต่อไป ทำให้การขอใช้งานเครื่อง การย้ายทรัพยากร หรือการสำรองข้อมูลง่ายขึ้น

2.3.1 KVM (Kernel Virtual Machine)

KVM (Kernel Virtual Machine) สามารถจำลองการทำงานของเครื่องคอมพิวเตอร์ได้หลายๆเครื่องพร้อมกัน และหลากหลายระบบปฏิบัติการ โดยแต่ละเครื่องจำลองจะมีฮาร์ดแวร์จำลองเป็นของตัวเอง เช่น CPU, RAM, network card, disk, graphics adapter และ อื่นๆ โดยสามารถเชื่อมต่อ RBD image จาก Ceph ให้เป็นอุปกรณ์จัดเก็บข้อมูลของเครื่องจำลองได้ ทำให้ง่ายต่อการควบคุมอุปกรณ์จัดเก็บข้อมูล [7]

2.3.2 libvirt

libvirt เป็นเครื่องมือที่ใช้ในการจัดการแพลตฟอร์มเวอร์ชวลไลเซชั่น โดยสามารถใช้งานร่วมกับ KVM ได้ ใช้เพื่อควบคุมเครื่องจำลองต่างๆ [8] ซึ่งโครงงานนี้จะใช้ KVM ในการสร้างเครื่องจำลอง และใช้ libvirt ในการจัดการต่างๆเกี่ยวกับเครื่องจำลอง

2.4 การจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์ (Software-Defined Storage)

เป็นเวอร์ชวลไลเซชั่นชนิดหนึ่งที่ใช้ซอฟแวร์ในการรวมทรัพยากรอุปกรณ์จัดเก็บข้อมูล (storage resource) ที่หลากหลายชนิดเข้าด้วยกัน เพื่อให้มีความสะดวกในการใช้งานมากยิ่งขึ้น[9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากผู้ดูแลระบบ (administrators) สามารถจัดการกับอุปกรณ์จัดเก็บข้อมูลได้ง่ายขึ้นผ่านทางการเขียนโปรแกรมควบคุม (programming) โดยปราศจากข้อจำกัดในทางกายภาพ (physical)

นอกจากนี้ยังทำให้การใช้งานอุปกรณ์จัดเก็บข้อมูลมีประสิทธิภาพมากขึ้น และสามารถใช้งานอุปกรณ์จัดเก็บข้อมูลร่วมกันหลายๆคนได้[10]

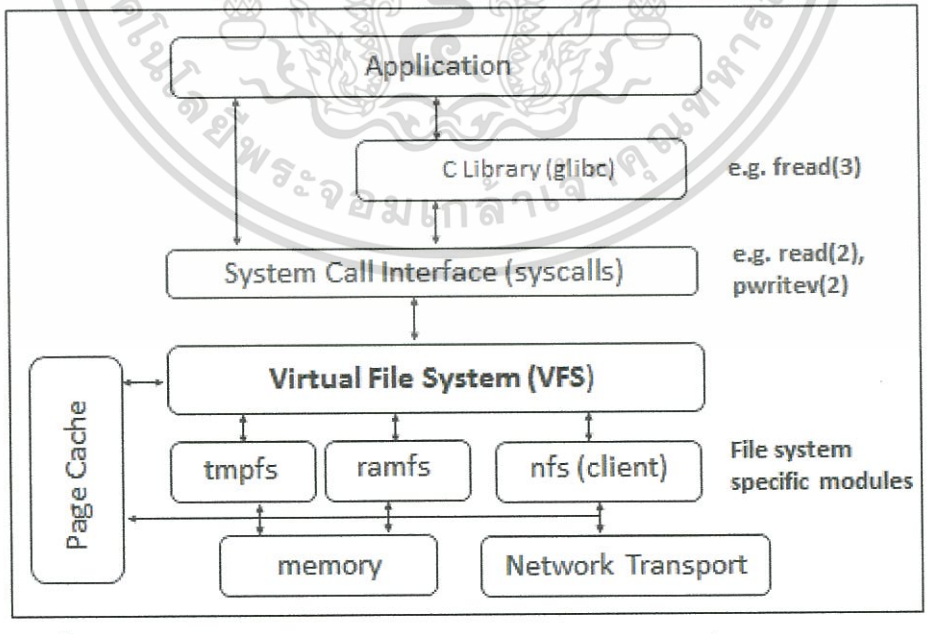
2.5 เทคโนโลยีการติดตามพฤติกรรมของแอปพลิเคชัน

ในการติดตามพฤติกรรมของแอปพลิเคชันต่างๆนั้น จะใช้การเขียนโค้ดเพื่อตรวจสอบพฤติกรรมของแอปพลิเคชัน โดยใช้ Systemtap เป็นเครื่องมือในการเขียนโค้ด ซึ่ง Systemtap เป็นภาษาสคริปต์(scripting language) และเป็นเครื่องมืออย่างหนึ่งที่นิยมใช้ในการวัดประสิทธิภาพของระบบต่างๆบนระบบปฏิบัติการ Linux kernel-based ที่มีความยืดหยุ่นในการใช้งาน [3]

จุดเด่นของ Systemtap มีดังนี้

1) ความยืดหยุ่น : สามารถสร้าง script ของตัวเองเพื่อวัดสิ่งที่ต้องการได้ โดยมี function ที่หลากหลาย
2) ความสะดวกในการใช้งาน : ใช้งานง่าย และ ผู้ใช้สามารถเขียน script เพื่อวัดค่าที่ต้องการได้จาก kernel-space และสร้างเป็น module file ที่นำไปใช้บนเครื่องใดๆก็ได้ที่มี kernel ตรงกันโดยไม่ต้องใช้เครื่องมือมากมาย, การคอมไพล์ซ้ำๆ หรือ การรีบูท kernel บ่อยๆ [11]

2.6 เทคโนโลยีการใช้งานอุปกรณ์จัดเก็บข้อมูลของระบบปฏิบัติการ Linux



รูป 2.3 ชั้นการทำงานของการติดต่อระหว่างแอปพลิเคชันและไฟล์ซิสเต็ม (File System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจาก glibc ถูกเรียกใช้งานแล้ว จะทำการเรียกใช้ System Call Interface ซึ่งบางแอปพลิเคชันจะข้ามชั้นนี้ไปเพื่อความรวดเร็วในการอ่านเขียนข้อมูลจากอุปกรณ์จัดเก็บข้อมูล

จากนั้น System Call จะถูกส่งไปยัง VFS (Virtual File System) โดย VFS มีไว้เพื่อครอบไฟล์ซิสเต็มที่หลากหลายชนิด เพื่อให้ทำงานร่วมกันได้ ซึ่ง VFS เป็นชั้นที่ใกล้กับอุปกรณ์จัดเก็บข้อมูลมากที่สุด ที่สามารถแทรกแซงเพื่อตรวจสอบพฤติกรรมการทำงานได้ จึงมีส่วนตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์จัดเก็บข้อมูล ทำงานอยู่ที่ VFS

2.7 งานที่เกี่ยวข้อง

2.7.1 Architecting a Software-Defined Storage Platform for Cloud Storage Service

สำหรับงานชิ้นนี้พูดถึงระบบ Software-Defined Storage นั้นจะต้องประกอบไปด้วย 3 ส่วนหลัก คือ การแยกกันระหว่างส่วนควบคุมและส่วนของข้อมูลที่ชัดเจน, การที่อุปกรณ์จัดเก็บข้อมูลสามารถจัดการตัวเองได้ และ RESTful API สำหรับการนำไปใช้งานในเชิงธุรกิจ โดยในงานชิ้นนี้มีส่วนประกอบหลักคือ Federator ที่เป็นส่วนที่แอปพลิเคชันจะสามารถติดต่อกับอุปกรณ์จัดเก็บข้อมูลได้ผ่าน RESTful API และ Adaptive Quick-Response Storage (AQR) ที่ใช้ในการกำหนดทรัพยากรด้วย Neural Network สองประเภท คือ Constant Neural Network (CNN) และ Adaptive Neural Network (ANN) โดยจะใช้ CNN เมื่อเกิดการละเมิด SLA เพื่อเพิ่มทรัพยากรเข้าไปทันทีที่ตรวจพบ แล้วจึงใช้ ANN เพื่อค่อย ๆ ปรับลดทรัพยากรที่ใช้ลงมาอยู่ในจุดที่เหมาะสม โดยที่ยังไม่ละเมิด SLA [15]

ส่วนที่แตกต่างกันก็คือ ในโครงงานชิ้นนี้จะใช้วิธีการสร้าง Virtual Machine ขึ้นมาเพื่อให้แอปพลิเคชันที่มีอยู่แล้วในปัจจุบันสามารถทำงานได้โดยไม่ต้องพัฒนาขึ้นใหม่ ในขณะที่ยังสามารถใช้ข้อได้เปรียบของความยืดหยุ่นที่ได้จากระบบ Software-Defined Storage ที่สามารถเลือก Storage ที่เหมาะสมกับ SLA ได้โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

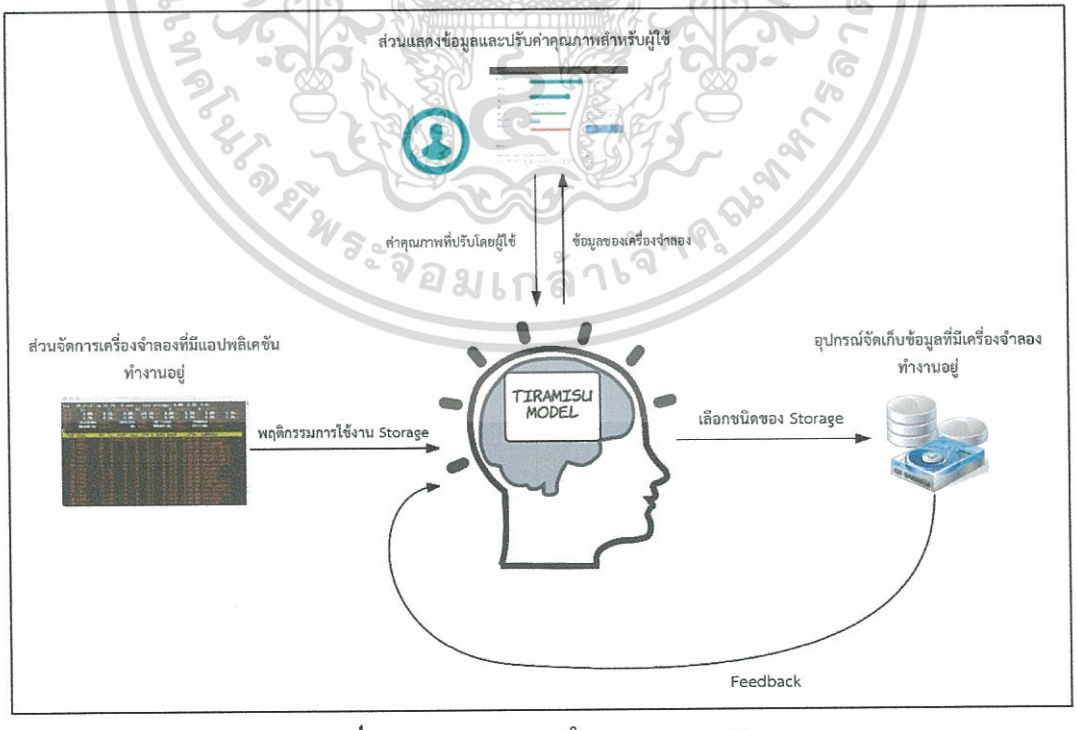
บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 ภาพรวมของระบบ

ในการเลือกชนิดของอุปกรณ์จัดเก็บข้อมูลที่มีความใกล้เคียงกับความต้องการของแอปพลิเคชัน ต้องดูถึงค่าต่างๆที่แอปพลิเคชันเรียกใช้จากอุปกรณ์จัดเก็บข้อมูล เช่น จำนวน I/O ที่แอปพลิเคชันเรียกใช้อุปกรณ์จัดเก็บข้อมูลต่อวินาที (IOPS) และ ความหน่วงของการอ่าน/เขียนข้อมูลแต่ละครั้ง (latency) นอกจากนี้ยังต้องคอยตรวจสอบประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลแต่ละชนิดในแต่ละช่วงเวลาว่าสามารถตอบสนองความต้องการของแอปพลิเคชันได้เพียงพอหรือไม่ โดยต้องมีการตรวจสอบผลการทำงานที่ได้กลับมาจากทั้งแอปพลิเคชันและอุปกรณ์จัดเก็บข้อมูลเพื่อนำค่าไปคำนวณหาอุปกรณ์ที่เหมาะสม

ระบบที่นำเสนอถูกแบ่งการทำงานออกเป็นสามส่วนหลักๆ คือ

1) ส่วนเลือกอุปกรณ์จัดเก็บข้อมูลที่สามารถให้ประสิทธิภาพได้ใกล้เคียงกับที่แอปพลิเคชันต้องการ
2) ส่วนจัดการเครื่องจำลองที่มีแอปพลิเคชันทำงานอยู่
3) ส่วนติดต่อกับผู้ใช้ผ่านทางหน้าเว็บ



รูป 3.1 ภาพรวมการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ส่วนเลือกชนิดอุปกรณ์จัดเก็บข้อมูล

เป็นส่วนที่ทำหน้าที่รับค่าพฤติกรรมการเรียกใช้งานอุปกรณ์จัดเก็บข้อมูลของเครื่องคอมพิวเตอร์จำลองและรับค่าประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูล ได้แก่ จำนวนครั้ง ที่แอปพลิเคชันเรียกใช้งานอุปกรณ์จัดเก็บข้อมูลต่อวินาที (IOPS) และ ความหน่วงในการอ่าน/เขียนข้อมูล (Latency)

เพื่อนำค่าที่ได้จากทั้งเครื่องคอมพิวเตอร์จำลอง และ อุปกรณ์จัดเก็บข้อมูล ไปคำนวณเพื่อหาอุปกรณ์จัดเก็บข้อมูลที่สามารถให้ประสิทธิภาพได้ใกล้เคียงกับที่ความต้องการของผู้ใช้มากที่สุด โดยใช้แบบจำลองที่สร้างขึ้น

3.1.2 ส่วนจัดการเครื่องจำลองที่มีแอปพลิเคชันทำงานอยู่

ส่วนนี้ทำหน้านี้ในการย้ายเครื่องจำลองที่มีแอปพลิเคชันทำงานอยู่ ไปไว้ที่อุปกรณ์จัดเก็บข้อมูลที่เลือกได้จากส่วนแรก โดยจะหยุดการทำงานของเครื่องจำลอง จากนั้นจึงทำสำเนา (copy) อิมเมจที่เป็นส่วนจัดเก็บข้อมูลของเครื่องจำลอง ไปไว้ในอุปกรณ์จัดเก็บข้อมูลชนิดใหม่และเริ่มการทำงานของเครื่องจำลองจากอิมเมจใหม่ และลบอิมเมจเดิมออกจากอุปกรณ์จัดเก็บข้อมูลชนิดเดิม

โดยส่วนนี้จะทำหน้าที่ส่งข้อมูลพฤติกรรมการใช้งานอุปกรณ์จัดเก็บข้อมูลของแอปพลิเคชันที่วัดผ่านทางเครื่องจำลองไปให้ส่วนแรกด้วย

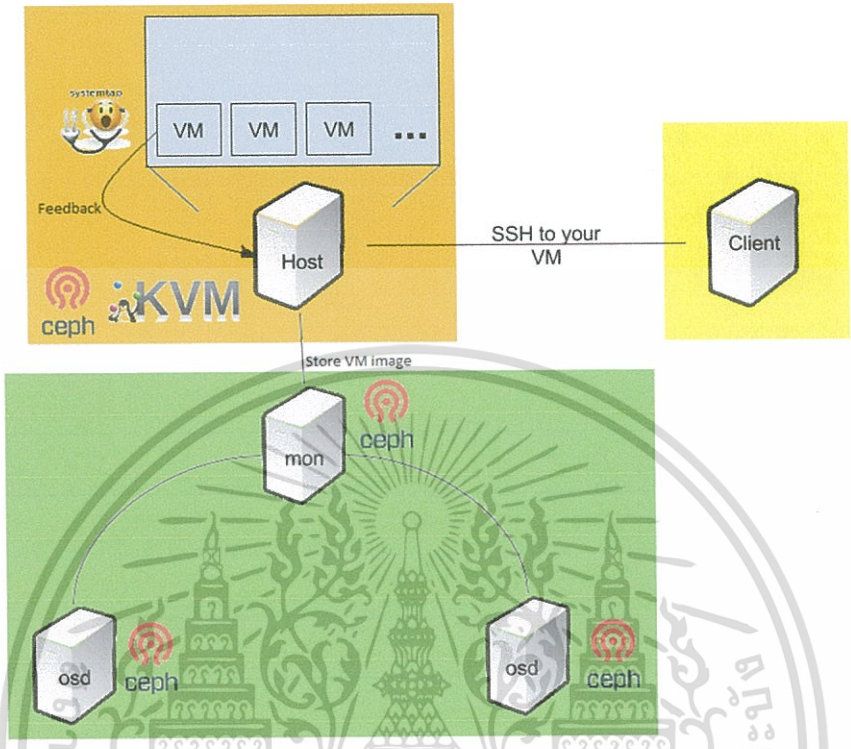
3.1.3 ส่วนติดต่อกับผู้ใช้งาน

ในการพัฒนามุ่งเน้นที่จะให้ผู้ใช้งานได้รับประสิทธิภาพในการทำงานของแอปพลิเคชันได้มาก ในงบประมาณที่จำกัด โดยที่ไม่ต้องใช้ความรู้ด้านอุปกรณ์จัดเก็บข้อมูลมากนัก แต่ยังสามารถปรับแต่งความสามารถของอุปกรณ์จัดเก็บข้อมูลได้เอง จึงมีส่วนปรับค่าประสิทธิภาพต่างๆ ได้แก่ จำนวนครั้งที่แอปพลิเคชันเรียกใช้อุปกรณ์จัดเก็บข้อมูลต่อวินาที (IOPS) และ ความหน่วงของการอ่าน/เขียนข้อมูลแต่ละครั้ง (latency) โดยหลังจากปรับค่าแล้ว จะนำค่าใหม่ไปคำนวณเพื่อหาอุปกรณ์จัดเก็บข้อมูลที่สามารถให้ค่าประสิทธิภาพได้ใกล้เคียงกับที่ผู้ใช้ต้องการ โดยจะคำนวณเป็นราคาเพื่อแสดงให้ผู้ใช้ทราบเพื่อใช้ในการตัดสินใจด้วย นอกจากนี้ ยังมีการแสดงข้อมูลต่างๆ ของเครื่องจำลอง และการแจ้งเตือนหากเครื่องจำลองอยู่บนอุปกรณ์จัดเก็บข้อมูลที่ไม่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โครงสร้างในการพัฒนาระบบ

ระบบที่นำเสนอมีโครงสร้างดังต่อไปนี้



รูป 3.2 แผนภาพการเชื่อมต่อระหว่างแต่ละส่วนของระบบ

3.2.1 ส่วนจัดเก็บข้อมูลแบบกระจาย (Distributed Storage System) [ส่วนสีเขียว]

ในส่วนนี้จะใช้ Ceph ในการทำแหล่งเก็บข้อมูลแบบกระจาย โดยเลือกใช้ Ceph Block Device เพื่อใช้ในการเชื่อมต่อกับเครื่องจำลอง (Virtual Machine) ซึ่งประกอบด้วย Monitor node และ OSD node โดยในแต่ละ OSDs มีการใช้งานอุปกรณ์จัดเก็บข้อมูลที่ต่างชนิดกัน และมี Monitor node เป็นตัวควบคุมการทำงานของระบบเก็บข้อมูลแบบกระจาย

โดยจะใช้การแบ่ง pool ในการควบคุมให้อุปกรณ์จัดเก็บข้อมูลแต่ละชนิดแยกออกจากกัน โดยให้ใน pool ประกอบไปด้วยอุปกรณ์จัดเก็บข้อมูลชนิดเดียวกันเท่านั้น เมื่อแอปพลิเคชันต้องย้ายไปใช้อุปกรณ์จัดเก็บข้อมูลชนิดอื่น ก็จะดำเนินการโดยการย้ายไปทำงานที่ pool อื่นที่มีอุปกรณ์จัดเก็บข้อมูลชนิดนั้นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ส่วนเครื่องจำลอง [ส่วนสีส้ม]

เป็นส่วนที่ประกอบไปด้วย KVM และ libvirt เพื่อสร้างและจัดการเครื่องจำลองให้กับผู้ใช้ โดยจะใช้สคริปต์ของ SystemTap เพื่อคอยตรวจสอบพฤติกรรมการเรียกใช้อุปกรณ์จัดเก็บข้อมูลของเครื่องจำลองที่มีแอปพลิเคชันทำงานอยู่ภายใน จากนั้นนำค่ามาทำการคำนวณเพื่อหาอุปกรณ์จัดเก็บข้อมูลที่ให้ประสิทธิภาพได้ใกล้เคียงกับที่ความต้องการของผู้ใช้ หากผลลัพธ์ที่ได้ไม่ตรงกับชนิดอุปกรณ์จัดเก็บข้อมูลที่อิมเมจของเครื่องจำลองอยู่ในปัจจุบันจะมีการแจ้งเตือนให้ผู้ใช้ทราบผ่านทางหน้าเว็บ เพื่อให้ผู้ใช้สามารถตัดสินใจว่าจะย้ายเครื่องคอมพิวเตอร์จำลองไปอยู่บนอุปกรณ์จัดเก็บข้อมูลที่แนะนำหรือไม่

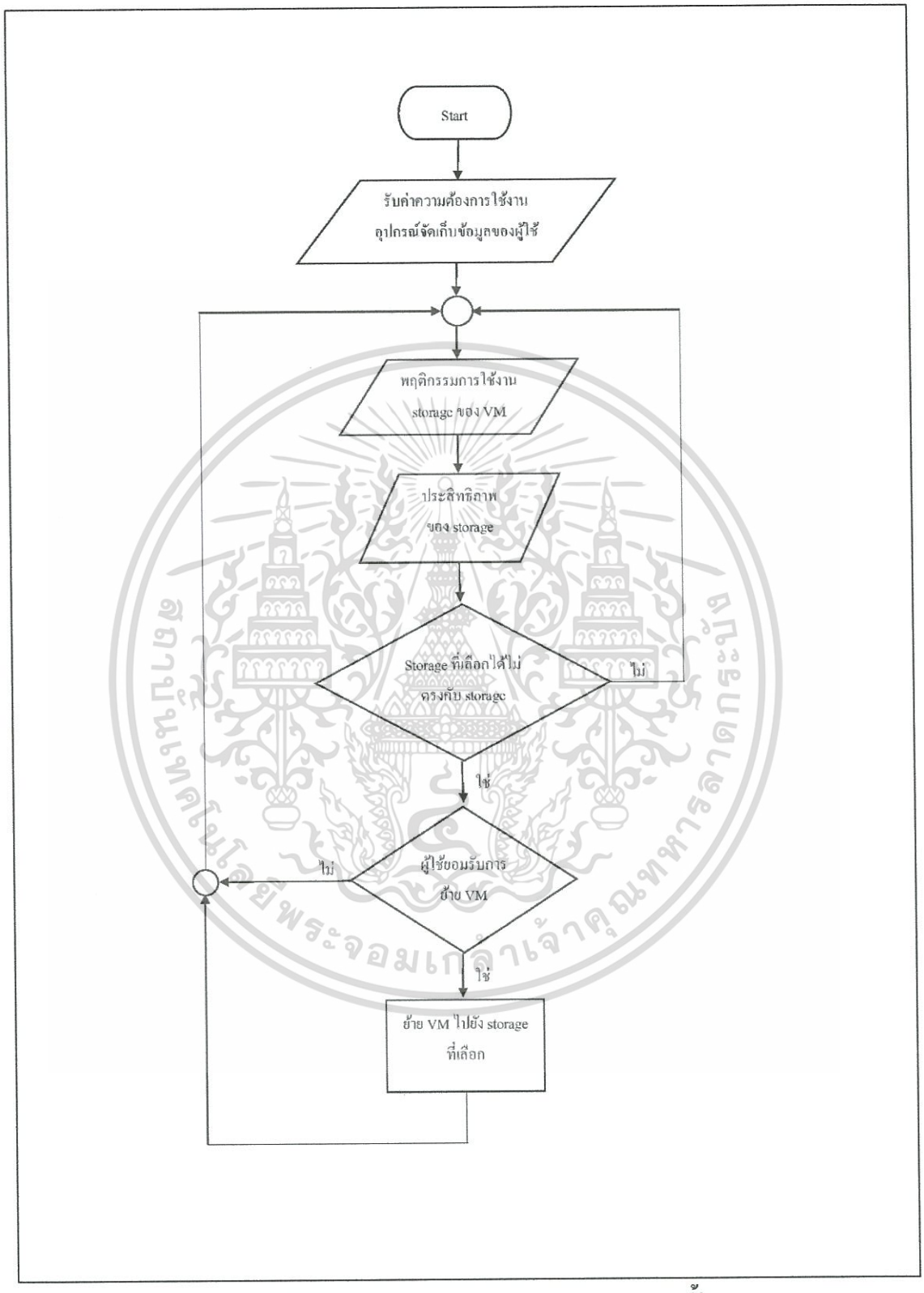
3.3.3 ส่วนติดต่อกับผู้ใช้ [ส่วนสีเหลือง]

ส่วนนี้แสดงผ่านทางหน้าเว็บ โดยนอกจากจะมีหน้าที่แสดงสถานะของเครื่องและการแจ้งเตือนตามที่อธิบายไปในส่วนที่แล้ว ยังมีการแสดงข้อมูลอื่นด้วย เช่น IP Address และสถานะการใช้งานอุปกรณ์จัดเก็บข้อมูล

นอกจากการแสดงผลแล้ว ผู้ใช้สามารถปรับแต่งค่าความหน่วงในการอ่าน/เขียนข้อมูล (Latency) และ จำนวนครั้งในการเรียกใช้อุปกรณ์จัดเก็บข้อมูลต่อวินาที (IOPS) ได้ด้วยตัวเอง ผ่านทางตัวเลื่อน (Slide bar) โดยจะมีการแสดงค่าของแอปพลิเคชันต้องการในขณะนี้ และค่าที่อุปกรณ์จัดเก็บข้อมูลปัจจุบันให้ได้ หากมีคำนวณได้ว่าแอปพลิเคชันควรย้ายไปใช้ชนิดอุปกรณ์จัดเก็บข้อมูลชนิดอื่น จะแจ้งให้ผู้ใช้ทราบพร้อมทั้งราคาใหม่ที่ผู้ใช้จะต้องเสียค่าบริการ เพื่อให้ผู้ใช้ตัดสินใจว่าจะย้ายหรือไม่ หากผู้ใช้แจ้งว่าต้องการย้ายจึงจะทำการย้ายอิมเมจของเครื่องจำลองไปไว้ยังอุปกรณ์จัดเก็บข้อมูลชนิดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนผังลำดับการทำงาน



รูป 3.3 แผนผังลำดับการทำงาน (Flowchart) ของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.3 แสดงลำดับการทำงานของระบบโดยรวม เริ่มด้วยการรับค่าความต้องการใช้งานอุปกรณ์จัดเก็บข้อมูลจากผู้ใช้(requirements) จากนั้นรับค่าพฤติกรรมการใช้งานอุปกรณ์จัดเก็บข้อมูลจากเครื่องคอมพิวเตอร์จำลองและค่าประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูล แล้วนำค่าทั้งหมดที่ได้ไปทำการคำนวณตามแบบจำลองที่มีเพื่อหาอุปกรณ์จัดเก็บข้อมูลที่มีประสิทธิภาพใกล้เคียงกับความต้องการของผู้ใช้งานมากที่สุดโดยไม่เกินความจำเป็น ถ้าหากอุปกรณ์จัดเก็บข้อมูลที่คำนวณออกมาได้ ไม่ใช่อุปกรณ์จัดเก็บข้อมูลที่แอปพลิเคชันทำงานอยู่ตอนนั้น จะแสดงข้อความแจ้งเตือนผู้ใช้ผ่านทางหน้าเว็บ เพื่อให้ผู้ใช้ตัดสินใจว่าจะย้ายแอปพลิเคชันไปทำงานบนอุปกรณ์จัดเก็บข้อมูลที่เหมาะสมกว่าหรือไม่ ซึ่งหากผู้ใช้ตกลงที่จะย้ายแอปพลิเคชัน ระบบก็จะเริ่มการย้ายแอปพลิเคชันไปไว้ยังอุปกรณ์จัดเก็บข้อมูลที่เหมาะสม

3.4 แบบจำลองในการแนะนำอุปกรณ์จัดเก็บข้อมูล

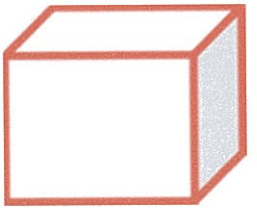
3.4.1 The cube

เป็นแบบจำลองที่ใช้ในการเลือกอุปกรณ์จัดเก็บข้อมูลที่สามารถให้ค่าประสิทธิภาพสูงสุดได้ใกล้เคียงกับความต้องการของผู้ใช้

โครงงานนี้ ได้มีการสร้างแบบจำลอง (Model) ที่ใช้ในการหาอุปกรณ์จัดเก็บข้อมูลที่ให้ประสิทธิภาพได้ใกล้เคียงกับที่ผู้ใช้ต้องการ โดยการสร้างกราฟ 3 มิติขึ้นมา ให้แกน x, y, z แทนค่า Throughput, Latency และ Cost ตามลำดับ จากนั้นจะรับค่าช่วงความต้องการใช้งาน(requirements) ของผู้ใช้มาตีกรอบในกราฟเพื่อสร้างเป็นกล่องของความต้องการใช้งานของผู้ใช้ (กล่องสีแดง)

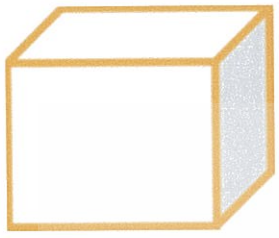
ในแต่ละช่วงเวลาจะมีการตรวจสอบประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูล แล้วแทนด้วยจุดลงในกราฟ

ตาราง 3.1 สัญลักษณ์ภายในกราฟแบบจำลอง the cube

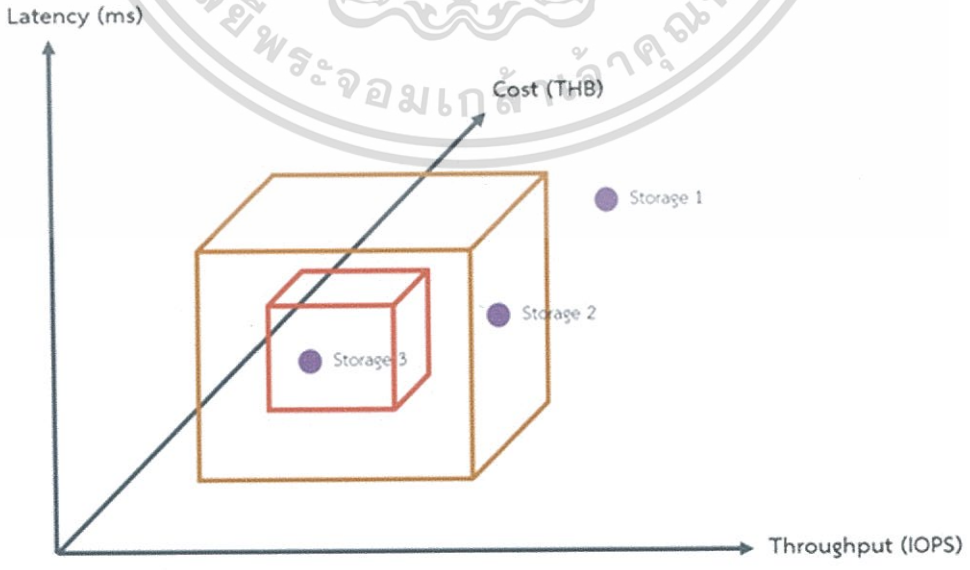
สัญลักษณ์	ชื่อ	ความหมาย
	กล่องความต้องการของผู้ใช้งาน	กล่องที่สร้างจากช่วงความต้องการประสิทธิภาพจากผู้ใช้ ที่ผู้ใช้ป้อนเข้าสู่ระบบผ่านทางหน้าเว็บ กล่องนี้จะถูกคงรูปและขนาดไว้เช่นนี้ตลอด เพื่อใช้ในการเปรียบเทียบว่าประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลที่ใช้อยู่ปัจจุบันตรงตามความต้องการของผู้ใช้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.1 สัญลักษณ์ภายในกราฟแบบจำลอง the cube (ต่อ)

สัญลักษณ์	ชื่อ	ความหมาย
	กล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบัน	กล่องที่ครอบคลุมจุดของอุปกรณ์จัดเก็บข้อมูลปัจจุบันที่เครื่องคอมพิวเตอร์จำลองทำงานอยู่ โดยกล่องนี้จะสามารถขยายขนาดหรือลดขนาดได้ เพื่อหาอุปกรณ์จัดเก็บข้อมูลที่สามารถให้ประสิทธิภาพได้ใกล้เคียงความต้องการของผู้ใช้งานมากที่สุด โดยกล่องนี้จะมีขนาดได้มากกว่าหรือเท่ากับกล่องความต้องการของผู้ใช้งานเท่านั้น (เมื่อเครื่องคอมพิวเตอร์จำลองถูกสร้างขึ้นครั้งแรก กล่องนี้จะมีขนาดเท่ากับกล่องความต้องการของผู้ใช้งาน)
	จุดประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูล	จุดที่สร้างจากการวัดประสิทธิภาพของอุปกรณ์จุดเก็บข้อมูล โดยจุดนี้จะสามารถเปลี่ยนตำแหน่งได้ตามประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูล

หลักการทำงานของแบบจำลอง the cube



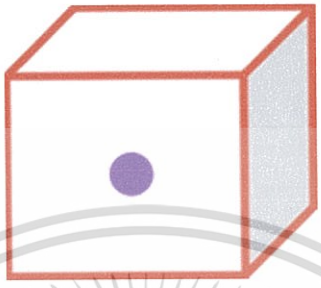
รูป 3.4 กราฟตัวอย่างการสร้างกล่อง requirements และ plot จุด performance storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองจะพิจารณาโดยแบ่งเป็นกรณี ดังนี้

3.4.1.1 กรณีที่ 1

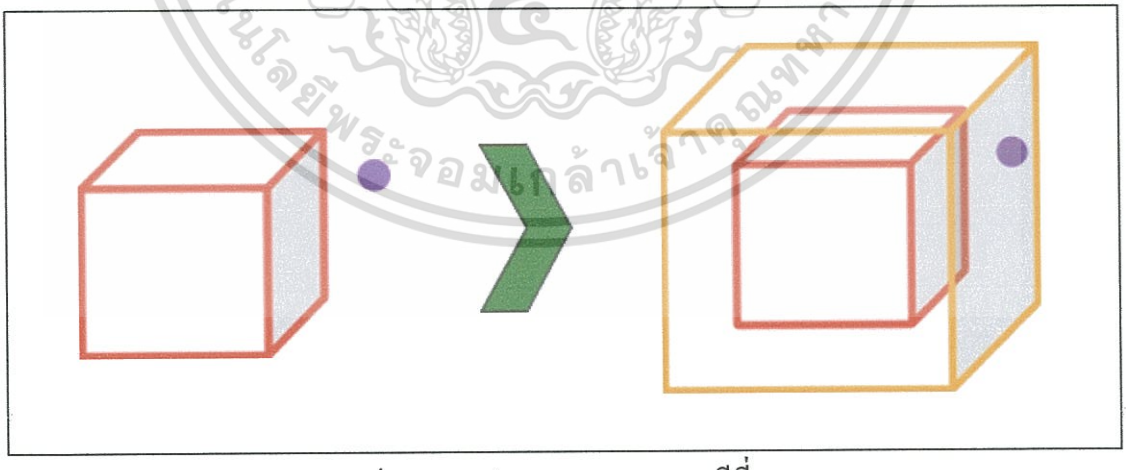
จุดของอุปกรณ์จัดเก็บข้อมูลที่เครื่องคอมพิวเตอร์จำลองทำงานอยู่ในปัจจุบัน อยู่ในกล่องของความต้องการใช้งานของผู้ใช้ หมายถึงอุปกรณ์จัดเก็บข้อมูลที่ใช้อยู่ในปัจจุบันตรงตามความต้องการของผู้ใช้แล้ว แบบจำลองจะแนะนำอุปกรณ์จัดเก็บข้อมูลปัจจุบันเช่นเดิม



รูป 3.5 แบบจำลอง the cube กรณีที่ 1

3.4.1.2 กรณีที่ 2

จุดของอุปกรณ์จัดเก็บข้อมูลที่เครื่องคอมพิวเตอร์จำลองทำงานอยู่ในปัจจุบัน อยู่นอกกล่องของความต้องการใช้งานของผู้ใช้และกล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบัน แบบจำลองจะทำการขยายกล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบันจนกว่าจะเจอจุดของอุปกรณ์จัดเก็บข้อมูลจุดแรกที่เข้ามาอยู่ในกล่อง และแบบจำลองจะแนะนำจุดของอุปกรณ์จัดเก็บข้อมูลนั้น

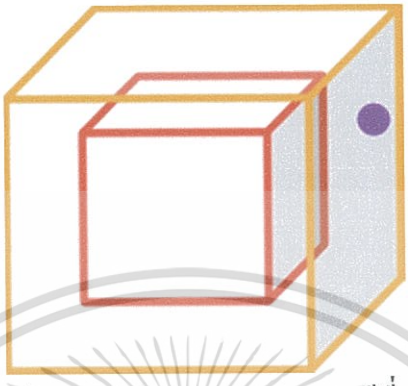


รูป 3.6 แบบจำลอง the cube กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.3 กรณีที่ 3

จุดของอุปกรณ์จัดเก็บข้อมูลที่เครื่องคอมพิวเตอร์จำลองทำงานอยู่ในปัจจุบัน อยู่นอกกล่องของความต้องการใช้งานของผู้ใช้ แต่อยู่ในกล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบัน



รูป 3.7 แบบจำลอง the cube กรณีที่ 3

1) เมื่อลดขนาดกล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบันลงมาแล้วไม่มีจุดอื่นอยู่ในกล่องนั้นเลย แบบจำลองจะแนะนำอุปกรณ์จัดเก็บข้อมูลปัจจุบัน และขยายขนาดกล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบันกลับคืนไปเหมือนเดิม
2) ลดขนาดกล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบันจนเล็กที่สุดที่ยังมีจุดอยู่ข้างในกล่อง แบบจำลองจะแนะนำอุปกรณ์จัดเก็บข้อมูลที่ให้ค่าที่มีความสำคัญต่อแอปพลิเคชันนั้นดีที่สุด

3.4.2 Line model

เป็นแบบจำลองที่ใช้ในการตรวจสอบการใช้งานของ VM ในปัจจุบันว่าอยู่ในช่วงของอุปกรณ์จัดเก็บข้อมูลชนิดใด เพื่อป้องกันการตั้งค่าความต้องการประสิทธิภาพอุปกรณ์จัดเก็บข้อมูลของผู้ใช้งานที่สูงเกินความจำเป็น

แบบจำลองนี้จะสร้างเส้นจำนวนขึ้นมา โดยให้แต่ละขีดบนเส้นจำนวนเป็นประสิทธิภาพสูงสุดที่อุปกรณ์จัดเก็บข้อมูลแต่ละชนิดสามารถให้ได้ จากนั้นในแต่ละช่วงเวลาจะมีการตรวจสอบการใช้งานอุปกรณ์จัดเก็บข้อมูลของ VM แล้วแทนด้วยจุดลงในกราฟ เพื่อดูว่าอยู่ในช่วงของอุปกรณ์จัดเก็บข้อมูลชนิดใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตาราง 3.2 สัญลักษณ์ภายในกราฟแบบจำลอง line model

สัญลักษณ์	ชื่อ	ความหมาย
	ค่าสูงสุดของอุปกรณ์จัดเก็บข้อมูล	ขีดแสดงค่าสูงสุดที่อุปกรณ์จัดเก็บข้อมูลแต่ละชนิดสามารถให้ได้ หาจากการรัน workload บนอุปกรณ์จัดเก็บข้อมูลแต่ละชนิดแล้ววัดค่า โดยขีดนี้จะสามารถเปลี่ยนตำแหน่งได้ตามประสิทธิภาพสูงสุดของอุปกรณ์จัดเก็บข้อมูลแต่ละชนิด
	จุดการใช้งานอุปกรณ์จัดเก็บข้อมูลของ VM	จุดที่สร้างจากการวัดการใช้งานอุปกรณ์จัดเก็บข้อมูลของ VM โดยจุดนี้จะสามารถเปลี่ยนตำแหน่งได้ตามการใช้งานอุปกรณ์จัดเก็บข้อมูลปัจจุบันของ VM

<u>หลักการทำงานของแบบจำลอง line model</u>

แบบจำลองจะพิจารณาโดยแบ่งเป็นกรณี ดังนี้

3.4.2.1 กรณีที่ 1

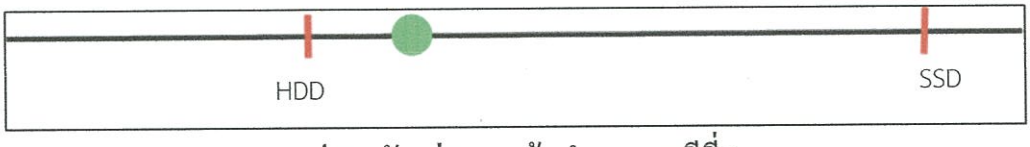
การใช้งานอุปกรณ์จัดเก็บข้อมูลของ VM มี throughput อยู่ในช่วง HDD

รูป 3.8 ตัวอย่างของเส้นจำนวนกรณีที่ 1

ในกรณีนี้แบบจำลองจะให้คำตอบเป็น HDD

3.4.2.2 กรณีที่ 2

การใช้งานอุปกรณ์จัดเก็บข้อมูลของ VM มี throughput อยู่ในช่วง SSD



รูป 3.9 ตัวอย่างของเส้นจำนวนกรณีที่ 2

ในกรณีนี้แบบจำลองจะให้คำตอบเป็น SSD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
144394

3.4.3 การทำงานร่วมกันระหว่าง 2 แบบจำลอง

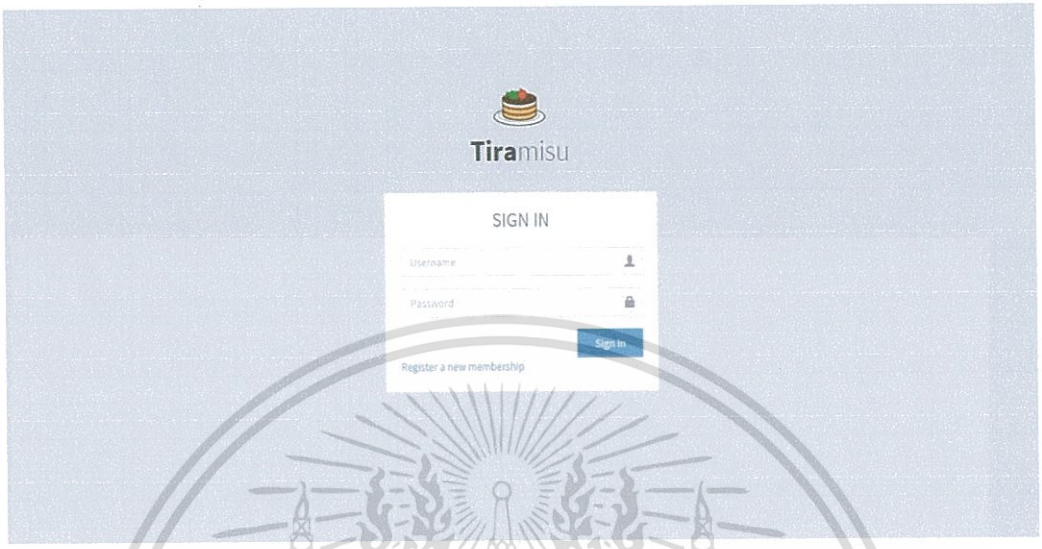
ตาราง 3.3 การทำงานร่วมกันระหว่าง 2 แบบจำลอง

The cube	Line model	คำตอบ	คำอธิบาย
HDD	HDD	HDD	ผู้ใช้งานตั้ง requirements ได้เหมาะสมกับการใช้งานอุปกรณ์จัดเก็บข้อมูลของ VM
HDD	SSD	SSD	VM มีการใช้งานอุปกรณ์จัดเก็บข้อมูลที่สูงกว่า requirements ที่ผู้ใช้ตั้งไว้
SSD	HDD	HDD	ผู้ใช้งานตั้ง requirements สูงเกินความต้องการของ VM
SSD	SSD	SSD	ผู้ใช้งานตั้ง requirements ได้เหมาะสมกับการใช้งานอุปกรณ์จัดเก็บข้อมูลของ VM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

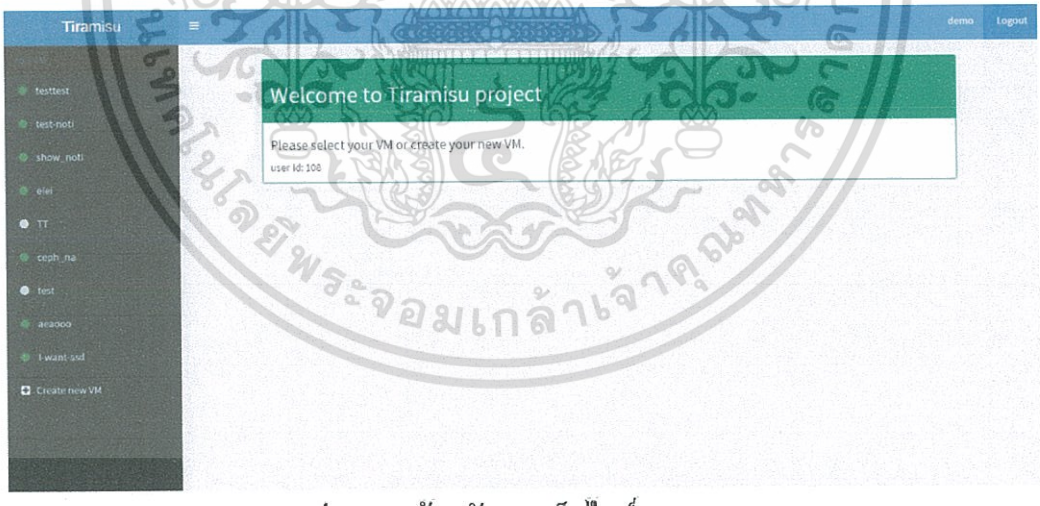
3.5 ส่วนแสดงผล

ผู้ใช้สามารถเข้าใช้งานโครงงาน Tiramisu ได้โดยผ่านหน้าเว็บไซต์ โดยจะได้ผลลัพธ์ดังรูป 3.12



รูป 3.12 หน้าเข้าสู่ระบบของเว็บ Tiramisu

เมื่อการลงชื่อเข้าใช้สำเร็จจะพบหน้าหลักของ Tiramisu ดังรูป 3.13



รูป 3.13 หน้าหลักของเว็บไซต์ Tiramisu

ด้านซ้ายจะมีชื่อ VM พร้อมวงกลมแสดงสถานะ ถ้าเป็นสีเขียว แปลว่าเครื่องเปิดอยู่ แต่ถ้าเป็นสีเทา แปลว่าเครื่องปิดอยู่ สามารถกดย่อแถบได้โดยการกดที่สามขีดด้านบน เมื่อย่อแถบแล้ว จะแสดงแค่วงกลมแสดงสถานะเท่านั้น สามารถดูได้คร่าวๆว่ามีเครื่องที่เปิดหรือปิดอยู่กี่เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกดที่ชื่อ VM ที่แสดงอยู่ด้านซ้าย จะเข้าสู่หน้าจอแสดงรายละเอียดเบื้องต้นของ VM ดังตัวอย่างรูป 3.20 ในหน้านี้จะมีปุ่ม 3 ปุ่มอยู่ด้านล่าง ได้แก่

1)

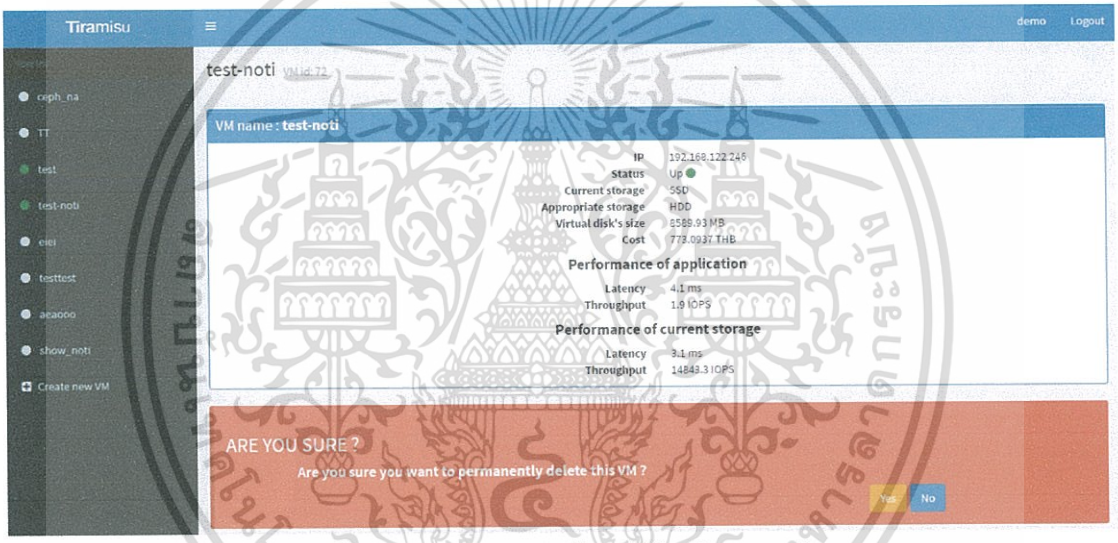
เป็นปุ่ม เปิด/ปิด VM ถ้าเครื่องเปิดอยู่จะเป็นคำว่า Shutdown กดเพื่อปิดเครื่อง, แต่ถ้าเครื่องปิดอยู่จะเป็นคำว่า Start กดเพื่อเปิดเครื่อง โดยหลังจากปิดเครื่องแล้วหน้านี้จะแสดงสถานะเป็น Down และวงกลมแสดงสถานะทางด้านซ้ายจะเป็นสีเทา

2)

สามารถกดปุ่มนี้เพื่อเข้าหน้าสำหรับเปลี่ยนแปลงค่า Requirements ต่างๆ

3)

กดเพื่อลบ VM เครื่องนี้ออกจากระบบ ซึ่งหลังจากกดแล้วจะแสดงหน้าจอดังรูป 3.14 เพื่อให้ผู้ใช้ตัดสินใจอีกครั้ง เนื่องจากหลังจากลบแล้วไม่สามารถนำกลับมาได้อีก โดยจะเริ่มลบเมื่อผู้ใช้กด Yes

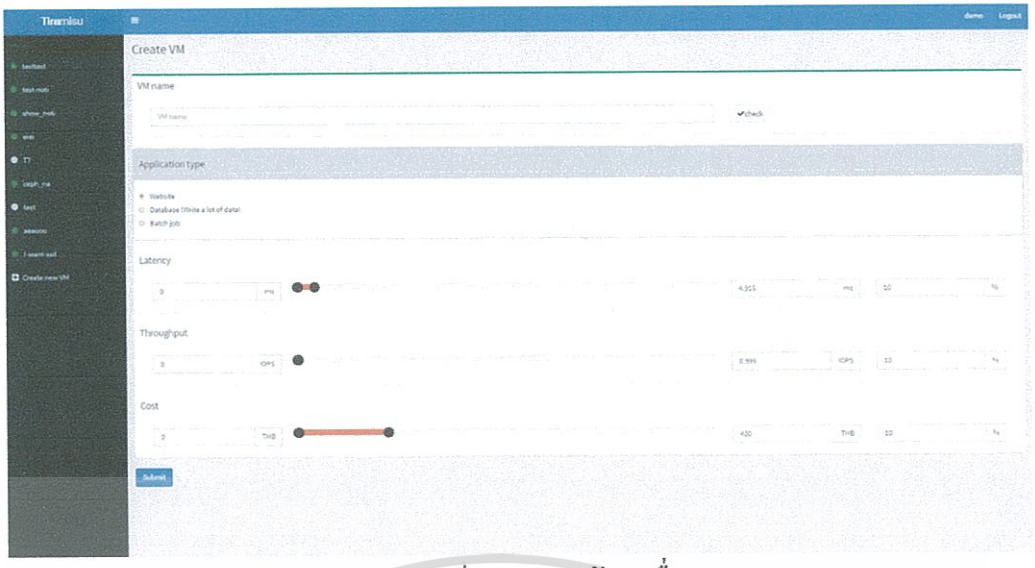


รูป 3.14 หน้าจอแสดงข้อความเตือนผู้ใช้หลังจากกดปุ่ม Delete

3.5.1 การสร้างเครื่องคอมพิวเตอร์จำลอง

หากผู้ใช้งานต้องการสร้าง VM ขึ้นมาใหม่ สามารถกด Create new VM ที่อยู่ล่างสุดของแถบด้านซ้าย (หรือเป็นเครื่องหมายบวก หากแถบโดนย่ออยู่)

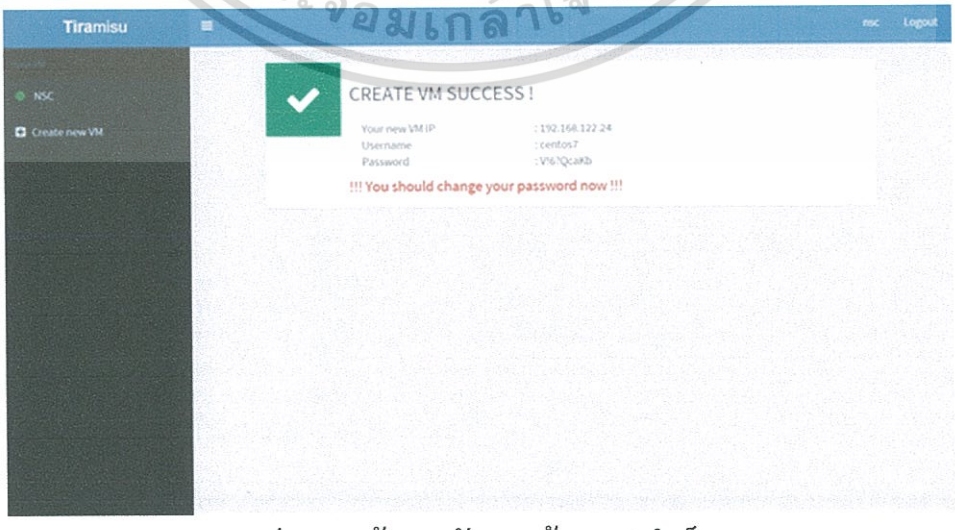
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.15 แบบฟอร์มสำหรับสร้างเครื่องจำลอง

รูป 3.15 เป็นหน้าจอหลังจากกดปุ่ม Create new VM ที่อยู่ทางด้านซ้าย ผู้ใช้จะต้องเลือกชนิดของแอปพลิเคชันที่จะนำมารันใน VM นี้ โดยเมื่อเลือกชนิดแล้ว ระบบจะใส่ค่าที่เหมาะสมให้โดยดูจากประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลในปัจจุบัน โดยที่ค่าที่ใส่ให้เป็นค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูล ซึ่งผู้ใช้สามารถเปลี่ยนแปลงค่าได้เองตามต้องการได้ เครื่องคอมพิวเตอร์จำลองที่จะสร้างขึ้นใหม่**ต้องห้ามใช้ชื่อซ้ำ**กับเครื่องคอมพิวเตอร์จำลองที่สร้างโดย account เดียวกัน โดยก่อนหลังจากใส่ชื่อของ VM เครื่องใหม่ลงไปในช่องแล้ว จะต้องกดปุ่ม check ที่อยู่หลังช่อง VM name ถ้าเป็นสีเขียว แปลว่าชื่อยังไม่ถูกใช้ สามารถใช้ชื่อนี้ได้ แต่ถ้าปุ่มเช็คเป็นสีแดง แปลว่าชื่อถูกใช้ไปแล้ว ต้องใช้ชื่ออื่น

หลังจากกรอกข้อมูลครบทุกช่องแล้วกดปุ่ม Submit ที่อยู่ล่างสุดของฟอร์ม ซึ่งหลังจากกด Submit ระบบจะเริ่มสร้าง VM โดยใช้เวลาประมาณ 2-3 นาที **ห้ามผู้ใช้งานปิดหน้าเว็บนี้เด็ดขาด**

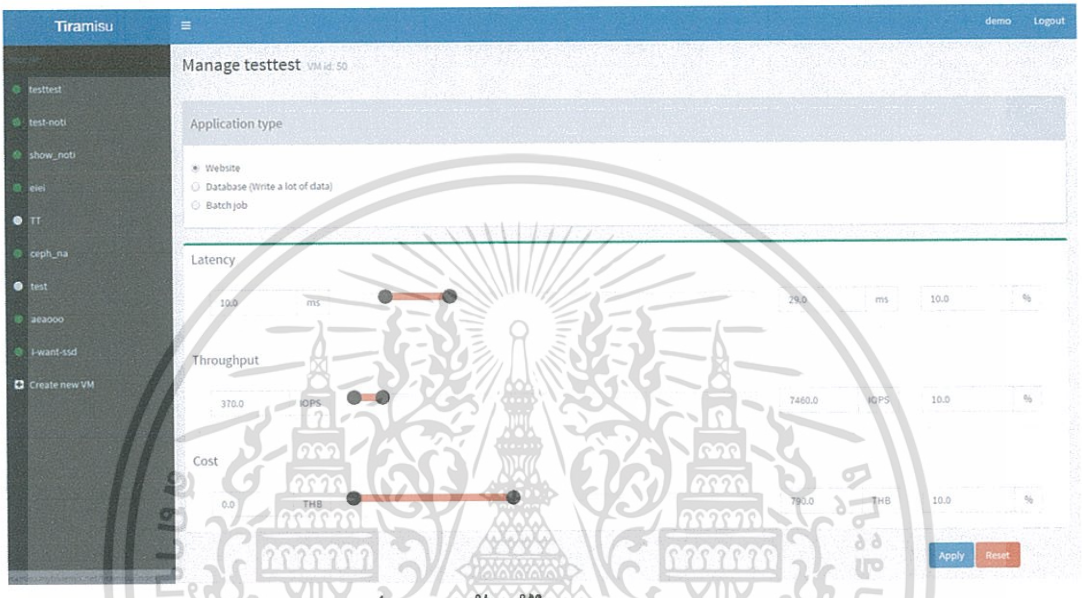


รูป 3.16 หน้าจอหลังจากสร้าง VM สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากระบบทำการสร้าง VM ใหม่สำเร็จแล้ว จะ Redirect ไปสู่หน้าจอดังรูป 3.16 ซึ่งจะแสดง IP ของเครื่อง และ Username Password ที่ใช้สำหรับเข้าสู่ระบบในครั้งแรก ซึ่งหลังจากผู้ใช้ SSH เข้าไปใน VM ครั้งแรกแล้ว ผู้ใช้ควรจะเปลี่ยน Password ทั้งของ User ตัวเอง และ Root เพื่อความปลอดภัย

3.5.2 การเปลี่ยนค่าความต้องการของผู้ใช้งาน (Requirements)



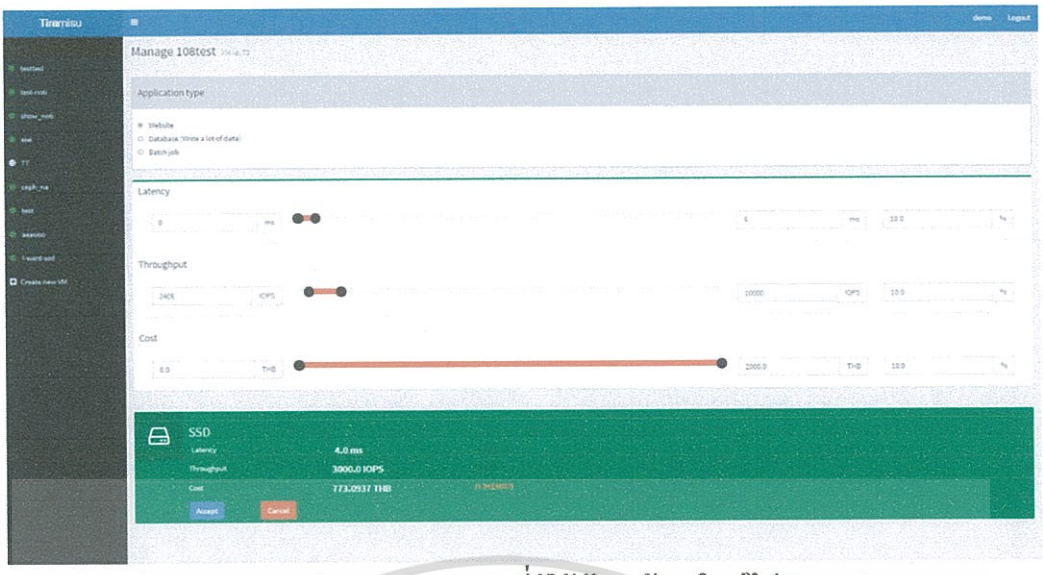
รูป 3.17 หน้าแก้ไข Requirements

จากหน้ารายละเอียดของ VM สามารถกดปุ่ม Manage เพื่อเข้าสู่หน้าปรับเปลี่ยน requirements ได้ โดยหลังจากเปลี่ยนค่า Requirements ที่ต้องการแล้ว กดปุ่ม Apply ที่อยู่ด้านล่าง ระบบจะนำค่าใหม่ไปคำนวณเพื่อหา Storage ที่เหมาะสม หาก Storage ที่เหมาะสม ไม่ตรงกับ Storage ที่ VM อยู่ตอนนี้ จะมีแถบแสดงชื่อ Storage ที่เหมาะสม, ค่าประมาณของ Latency และ IOPS ของ VM ที่อยู่ใน Storage พร้อมทั้งบอกราคาที่ต้องจ่ายหากเราย้ายไป Storage นั้น

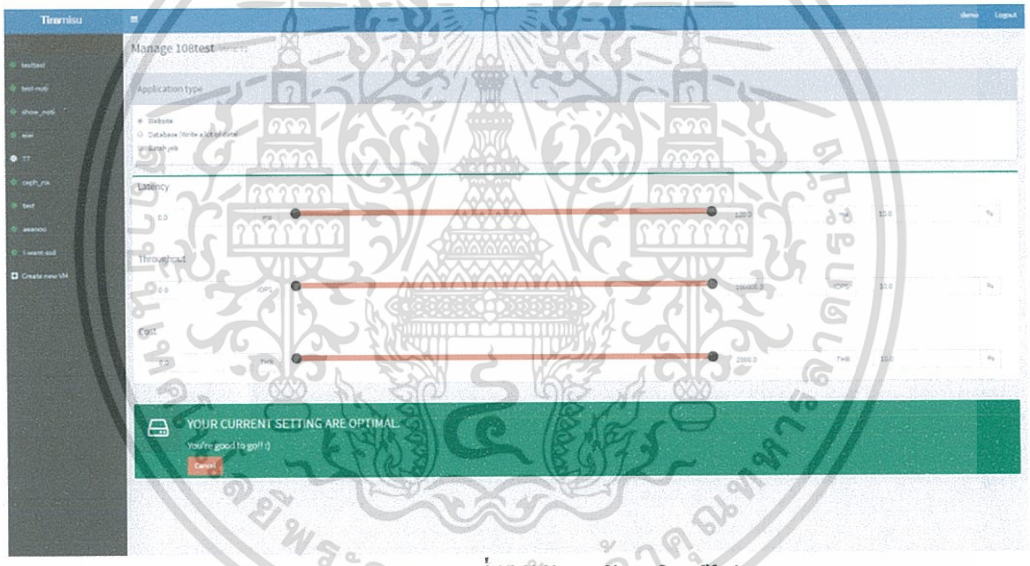
หากกดปุ่ม Accept ระบบจะทำการย้าย VM ไปยัง Storage นั้น ซึ่งเราสามารถกด Cancel ได้หากไม่ต้องการย้าย (**ระหว่างการย้าย ห้ามผู้ใช้ปิดหน้าเว็บหรือกดปุ่มใดๆเด็ดขาด** เวลาที่ใช้ในการย้ายขึ้นอยู่กับขนาด virtual storage ของ VM)

นอกจากนี้ หากกรอกค่าใหม่ลงไปในหน้า Manage แล้ว แต่ต้องการคืนค่า Requirements เดิม สามารถกดปุ่ม Reset ได้ ระบบจะล้างการเปลี่ยนแปลงออกและแสดง Requirements ล่าสุดของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.18 ระบบนำค่า requirements ที่ผู้ใช้ป้อนเข้ามาใหม่ไปทำการคำนวณ โดยแบบจำลองและแนะนำ Storage ที่เหมาะสม

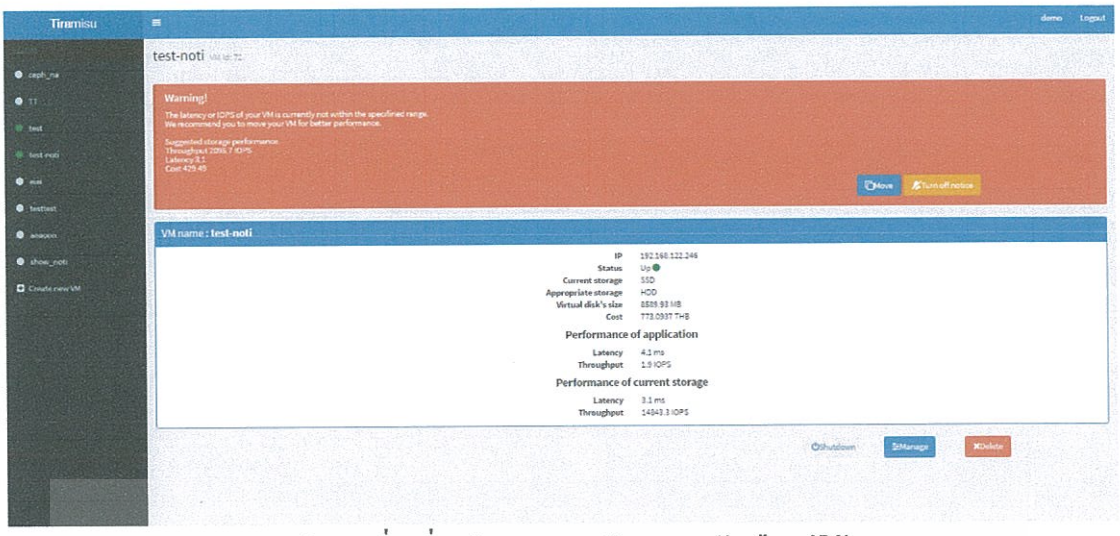


รูป 3.19 นำค่า requirements ที่ผู้ใช้ป้อนเข้ามาใหม่ไปทำการคำนวณ โดยแบบจำลองได้ว่าไม่ควรย้าย Storage

3.5.3 ระบบตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์จัดเก็บข้อมูล (Monitoring)

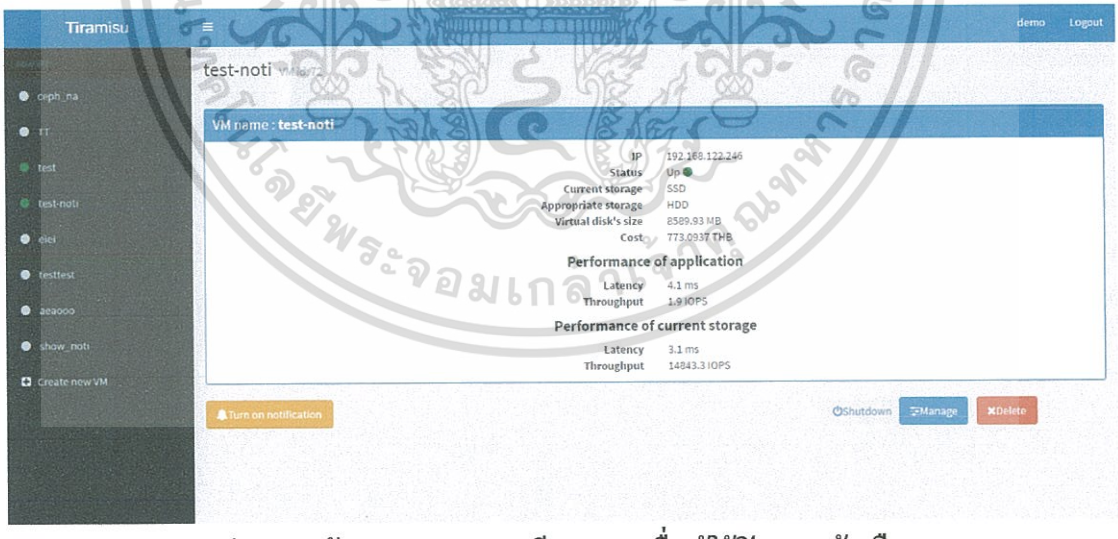
ใน Project Tiramisu จะมีส่วน Monitoring ที่คอยตรวจสอบประสิทธิภาพของ VM ที่อยู่ในอุปกรณ์จัดเก็บข้อมูลต่างๆ และตรวจสอบประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลทุกตัวในระบบอยู่ตลอดเวลา หลังจากตรวจสอบแล้วจะส่งค่าที่ตรวจสอบได้ไปประมวลผลที่แบบจำลองเพื่อหาอุปกรณ์จัดเก็บข้อมูลที่เหมาะสมกับแอปพลิเคชันที่อยู่ใน VM ซึ่งหากแบบจำลองคำนวณได้ว่า VM ไหนควรจะย้ายอุปกรณ์จัดเก็บข้อมูลจะแสดงข้อความแจ้งให้ผู้ใช้ทราบในหน้าแสดงรายละเอียดของ VM นั้นๆ ดังรูป 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.20 สี่เหลี่ยมสีแดงแสดงข้อความแจ้งเตือนผู้ใช้

ในกล่องแจ้งเตือนจะมีปุ่มให้ผู้ใช้ตัดสินใจ หากต้องการจะย้ายตามคำแนะนำให้กดปุ่ม Move ระบบจะทำการ Shutdown VM และทำการย้ายไปยังอุปกรณ์จัดเก็บข้อมูลที่เหมาะสม **(ระหว่างการย้าย ห้ามผู้ใช้ปิดหน้าเว็บหรือกดปุ่มใดๆเด็ดขาด** เวลาที่ใช้ในการย้ายขึ้นอยู่กับขนาด virtual storage ของ VM) แต่หากผู้ใช้ไม่ต้องการย้าย สามารถกด Turn off notice เพื่อปิดการแจ้งเตือนได้ ซึ่งหลังจากผู้ใช้ปิดการแจ้งเตือนไปแล้ว จะมีปุ่ม Turn on notification อยู่มุมล่างซ้ายเพื่อให้ผู้ใช้สามารถเปิดการแจ้งเตือนได้อีกครั้ง



รูป 3.21 หน้าจอแสดงรายละเอียด VM เมื่อ ผู้ใช้ปิดการแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดสอบเกี่ยวกับความถูกต้องของแบบจำลอง

การทดลองนี้จะทำการจำลองค่าต่างๆเพื่อจำลองสถานการณ์แบบต่างๆ และในการทดลองนี้เราได้สมมติค่าเริ่มต้นเป็นดังต่อไปนี้ เพื่อความสะดวกในการคำนวณเพื่อตรวจสอบความแม่นยำของแบบจำลอง

กำหนดให้

1) ราคาต่อ MB ของ HDD = 0.050 THB/MB และ SSD = 0.090 THB/MB
2) ขนาดของอุปกรณ์จัดเก็บข้อมูลจำลองของ VM ที่ใช้ในการทดสอบคือ 1000 MB
3) Current storage ของ VM ที่ใช้ทดสอบ คือ HDD
4) ค่า Requirements ของ User เป็นดังรูป 4.1 (เป็นค่าที่กำหนดขึ้นมาเพื่อความง่ายต่อการคำนวณ)

รูป 4.1 Requirements ของผู้ใช้ที่กำหนดขึ้นเพื่อการทดลอง

จาก Requirements ข้างต้น จะคำนวณได้กล่องของ Requirements (ในที่นี้จะเรียก cube) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.1 กล่องความต้องการใช้งานของผู้ใช้ที่สร้างขึ้นจากค่าความต้องการที่ผู้ใช้ป้อนเข้าสู่ระบบผ่านทางหน้าเว็บ

	Min	Max	% ความคลาดเคลื่อน
Latency (sec)	3.8	6	5
IOPS	400	550	10
Cost (THB)	45	90	10

แบ่งทดสอบตาม case ดังนี้

4.1.1 กรณีที่ 1 ความต้องการของผู้ใช้ตรงตามประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบัน

กำหนดค่า Performance ของแต่ละ Storage และของ Application เป็นดังนี้

ตาราง 4.2 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 1

	Application	HDD	SSD
Latency (sec)	5	5	2
IOPS	401	401	600
Cost (THB)	50	50	90

หมายเหตุ : หลักที่ชื่อ Application หมายถึงค่าประสิทธิภาพปัจจุบันของแอปพลิเคชัน

ผลลัพธ์

แบบจำลองแนะนำอุปกรณ์จัดเก็บข้อมูลชนิดเดิม ซึ่งก็คือ HDD ทำให้ไม่จำเป็นต้องย้ายไปยังอุปกรณ์จัดเก็บข้อมูลชนิดอื่น

รูป 4.2 ผลการคำนวณจากแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 กรณีที่ 2 ไม่มีอุปกรณ์จัดเก็บข้อมูลใดตรงตามความต้องการของผู้ใช้งาน

กำหนดค่า Performance ของแต่ละ Storage และของ Application เป็นดังนี้

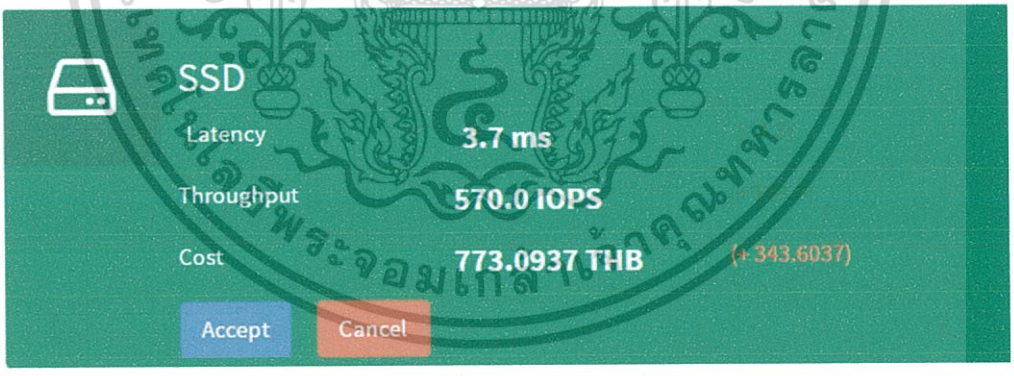
ตาราง 4.3 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 2

	Application	HDD	SSD
Latency (sec)	8	8	3.7
IOPS	401	401	570
Cost (THB)	50	50	90

หมายเหตุ : หลักที่ชื่อ Application หมายถึงค่าประสิทธิภาพปัจจุบันของแอปพลิเคชัน

ผลลัพธ์

เนื่องจากไม่มีอุปกรณ์จัดเก็บข้อมูลชนิดไหนตรงตามความต้องการของผู้ใช้ แบบจำลองจึงทำการขยายกล่องจนกว่าจะเจอจุดแรกที่เข้ามาอยู่ในกล่อง และเลือกจุดนั้น ในที่นี้ SSD เป็นจุดแรกที่แบบจำลองขยายแล้วเจอ ซึ่งหมายความว่าเป็นจุดที่อยู่ใกล้ความต้องการของผู้ใช้งานมากที่สุด



รูป 4.3 ผลการคำนวณจากแบบจำลอง

4.1.3 กรณีที่ 3 การหาอุปกรณ์จัดเก็บข้อมูลชนิดอื่นที่อาจจะให้ประสิทธิภาพได้ใกล้เคียงกับความต้องการของผู้ใช้งานมากกว่าอุปกรณ์จัดเก็บข้อมูลปัจจุบัน

4.1.3.1 กรณีที่ 3.1

เมื่อลดขนาดกล่องลงมาแล้วไม่มีจุดอื่นอยู่ในกล่องนั้นเลย แบบจำลองจะเลือกไม่แนะนำให้ย้ายอุปกรณ์จัดเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดค่า Performance ของแต่ละ Storage และของ Application เป็นดังนี้

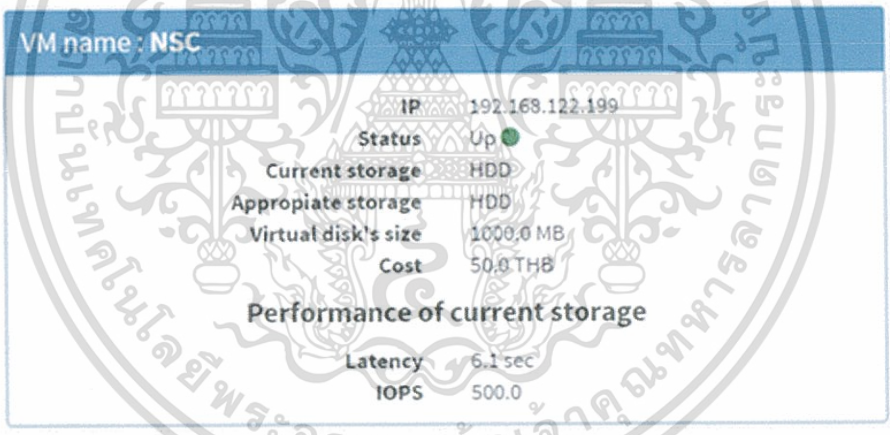
ตาราง 4.4 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 3.1

	Application	HDD	SSD
Latency (sec)	6.1	6.1	4
IOPS	500	500	600
Cost (THB)	50	50	90

หมายเหตุ : หลักที่ชื่อ Application หมายถึงค่าประสิทธิภาพปัจจุบันของแอปพลิเคชัน

ผลลัพธ์

เนื่องจากยังไม่มีอุปกรณ์จัดเก็บข้อมูลชนิดอื่นที่สามารถให้ประสิทธิภาพได้ใกล้เคียงเท่าชนิดปัจจุบัน จึงไม่มีการแนะนำให้เปลี่ยนชนิดอุปกรณ์จัดเก็บข้อมูล



รูป 4.4 รายละเอียดของ VM จะเห็นว่า Current storage ตรงกับ Appropriate storage

4.1.3.2 กรณีที่ 3.2

ลดขนาดจนเล็กสุดที่ยังมีจุดอยู่ข้างใน แบบจำลองจะแนะนำอุปกรณ์จัดเก็บข้อมูลที่ให้ค่าที่มีความสำคัญต่อแอปพลิเคชันนั้นดีที่สุด

กำหนดค่า Performance ของแต่ละ Storage และของ Application เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

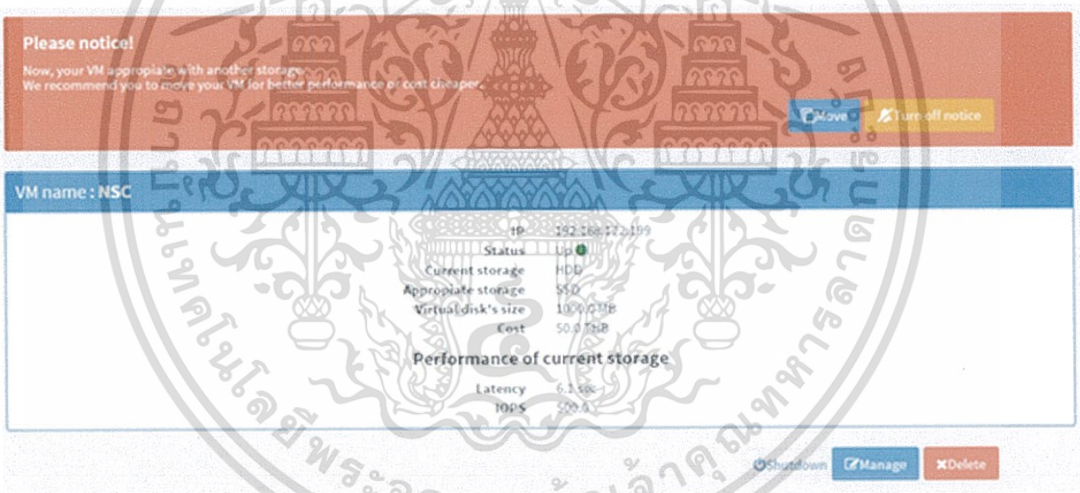
ตาราง 4.5 ค่าประสิทธิภาพปัจจุบันของอุปกรณ์จัดเก็บข้อมูลที่กำหนดขึ้นเพื่อทดสอบกรณีที่ 3.2

	Application	HDD	SSD
Latency (sec)	6.1	6.1	3.9
IOPS	500	500	500
Cost (THB)	50	50	90

หมายเหตุ : หลักที่ชื่อ Application หมายถึงค่าประสิทธิภาพปัจจุบันของแอปพลิเคชัน

ผลลัพธ์

จะเห็นว่า SSD สามารถให้ค่าที่ใกล้เคียงกับความต้องการของผู้ใช้มากกว่า HDD ซึ่งเป็นอุปกรณ์จัดเก็บข้อมูลที่ VM อยู่ในปัจจุบัน จึงมีการแจ้งเตือนผ่านทางหน้าเว็บเพื่อให้ผู้ใช้ทราบและสามารถตัดสินใจย้ายได้



รูป 4.5 รายละเอียดของ VM จะเห็นว่า Current storage ไม่ตรงกับ Appropriate storage และมีการแสดงการแจ้งเตือนเพื่อให้ย้ายไปยังอุปกรณ์จัดเก็บข้อมูลที่เหมาะสม

สรุป : จากการผลการทดสอบทั้งหมดที่ผ่านมา สรุปได้ว่า Model ให้คำตอบที่ถูกต้องในทุกกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ประสิทธิภาพการทำงานของแบบจำลอง

ในการทดลองนี้ได้มีการวิเคราะห์ประสิทธิภาพการทำงานของแบบจำลองที่สร้างขึ้น ซึ่งวิเคราะห์ได้ว่าแบบจำลองมีความซับซ้อนในกรณีที่เลวร้ายที่สุด(worse case) เป็น O(NM) โดยที่

- N หมายถึง จำนวนของอุปกรณ์จัดเก็บข้อมูลในระบบ
- M หมายถึง ช่วงของค่าที่เป็นไปได้สำหรับแต่ละค่าประสิทธิภาพที่สนใจ

จากกรณีทั้งหมดที่อาจเกิดขึ้นได้ จะเห็นว่ากรณีที่เลวร้ายที่สุด คือ แบบจำลองจะทำซ้ำตามจำนวนชนิดอุปกรณ์จัดเก็บข้อมูล(N) ที่มีในระบบเพื่อหาว่ามีชนิดใดตรงตามช่วงความต้องการของผู้ใช้งาน แต่ไม่มีอุปกรณ์จัดเก็บข้อมูลชนิดใดอยู่ในช่วงความต้องการของผู้ใช้งานเลย แบบจำลองจึงทำการขยายขนาดจนกว่าจะเจออุปกรณ์จัดเก็บข้อมูลชนิดแรก โดยอุปกรณ์จัดเก็บข้อมูลชนิดแรกที่เจออาจจะอยู่ที่ปลายสุดของช่วงที่เป็นไปได้สำหรับแต่ละค่าประสิทธิภาพ ซึ่งต้องทำซ้ำทั้งหมด M ครั้ง

จากการคำนวณข้างต้นสามารถประมาณค่าความซับซ้อนได้เป็น O(NM) ซึ่งหมายความว่าแบบจำลองนี้จะใช้เวลามากที่สุดในการคำนวณเท่ากับ NM หน่วยเวลา ซึ่งจากการทดลองแบบจำลองนี้ใช้เวลา 0.07 วินาที ในการแนะนำอุปกรณ์จัดเก็บข้อมูลที่ใกล้เคียงกับค่าความต้องการของผู้ใช้ ในขณะที่มีอุปกรณ์จัดเก็บข้อมูลในระบบ 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุปของโครงงาน

โครงงานนี้ได้ช่วยให้นักพัฒนาซอฟต์แวร์สามารถเลือกอุปกรณ์จัดเก็บข้อมูลที่เหมาะสมที่สุดจากความต้องการประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลจากผู้ใช้งาน โดยไม่ต้องมีความรู้ในเชิงลึกเกี่ยวกับส่วนจัดเก็บข้อมูลภายในเวลาสั้น ๆ ซึ่งรวดเร็วกว่ากระบวนการแบบเดิมที่ต้องพิจารณาแต่ละประเภทอย่างละเอียดและคัดเลือกอุปกรณ์จัดเก็บจัดเก็บข้อมูลที่เหมาะสมขึ้นมาใช้งานด้วยตนเอง ซึ่งนอกจากจะทำให้การพิจารณาการเลือกใช้งานอุปกรณ์จัดเก็บง่ายขึ้นแล้ว ยังช่วยลดระยะเวลาในการพัฒนาซอฟต์แวร์อีกด้วย

5.2 แนวทางในการพัฒนาต่อในอนาคต

5.2.1 ส่วนของแบบจำลองที่ใช้ในการแนะนำอุปกรณ์จัดเก็บข้อมูล

แบบจำลองที่ใช้ในปัจจุบันมีค่าความซับซ้อน(Complexity)ในกรณีที่เลวร้ายที่สุด(worse case) เป็น O(NM) แต่สามารถแก้ไขเพื่อลดความซับซ้อนลงเป็น O(N logN) เพื่อให้ใช้เวลาในการคำนวณน้อยลงได้โดยการลดกล่องในกราฟลงโดยไม่ใช้กล่องประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลปัจจุบัน เหลือแค่กล่องความต้องการของผู้ใช้งาน ถ้าหาครั้งแรกไม่มีจุดประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลอยู่ในกล่องความต้องการของผู้ใช้งาน แบบจำลองจะหาจุดที่ใกล้กล่องที่สุดโดยการหาระยะห่างระหว่างจุดประสิทธิภาพของอุปกรณ์จัดเก็บข้อมูลกับกล่อง โดยใช้วิธีหาขนาด vector ของแต่ละจุดกับกล่อง จะได้ความซับซ้อนเป็น O(N) เนื่องจากต้องมีการทำซ้ำตามจำนวนอุปกรณ์จัดเก็บข้อมูลในระบบ

จากนั้นนำขนาด vector ของแต่ละอุปกรณ์จัดเก็บข้อมูลมาเรียงลำดับโดยใช้ Sorting algorithm ที่มีความซับซ้อนน้อยที่สุด(O(N logN)) และแบบจำลองจะแนะนำจุดที่มีขนาดน้อยสุดคือใกล้กล่องสุด หมายถึงอุปกรณ์จัดเก็บข้อมูลนั้นสามารถให้ค่าประสิทธิภาพได้ใกล้เคียงกับความต้องการของผู้ใช้งานมากที่สุด

หากพัฒนาแบบจำลองตามวิธีการข้างต้นจะได้ความซับซ้อนในกรณีที่เลวร้ายที่สุด(worse case) เป็น O(N logN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ส่วนของการตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์จัดเก็บข้อมูล

ส่วนของการตรวจสอบประสิทธิภาพการทำงานของอุปกรณ์จัดเก็บข้อมูลที่ใช้ในโครงงานชิ้นนี้นั้นทำงานใน user space โดยการเรียกใช้งาน Systemtap ซึ่งวัดได้ตรงในระดับหนึ่ง แต่หากต้องการความแม่นยำที่สูงขึ้นอาจทำได้โดยพัฒนาขึ้นเป็น kernel module เพื่อลดส่วนเกิน (overhead) ในการวัด และลดผลที่อาจเกิดขึ้นต่อประสิทธิภาพของแอปพลิเคชันลงอีกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] "HDD, SSD or Flash Storage? Which storage option is the best for the Mac Pro?," [Online]. Available: http://create.pro/blog/storage-option-best-mac-pro-hdd-ssd-flash-storage/. [Accessed 31 Nov 2015]

[2] "Direct I/O in InnoDB (O_DIRECT)," [Online]. Available: http://www.psce.com/blog/kb/direct-io-in-innodb-o_direct/. [Accessed 25 Aug 2015]

[3] "SystemTap," [Online]. Available: https://en.wikipedia.org/wiki/SystemTap. [Accessed 7 Nov 2015]

[4] "CEPH BLOCK DEVICE," [Online]. Available: http://docs.ceph.com/docs/master/rbd/rbd/. [Accessed 8 Oct 2015]

[5] T. G. Peter Mell, "The NIST Definition of Cloud," NIST Special Publication 800-145, 2011.

[6] "Intel® Virtualization Technology (Intel® VT)," [Online]. Available: http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html [Accessed 11 Nov 2015]

[7] "KVM," [Online]. Available: http://www.linux-kvm.org/. [Accessed 7 Nov 2015]

[8] "libvirt," [Online]. Available: https://en.wikipedia.org/wiki/Libvirt. [Accessed 8 Nov 2015]

[9] Chao-Tung Yang, Wei-Hsiang Lien, Yu-Chuan Shen, Fang-Yi Leu, "Implementation of a Software-Defined Storage Service with Heterogeneous Storage Technologies", *2015 IEEE International Conference on Services Computing*, 2015, pp. 102-107.

[10] "software-defined storage definition", [Online]. Available: http://searchsdn.techtarget.com/definition/software-defined-storage. [Accessed 15 Nov 2015]

[11] Don Domingo และ William Cohen, "*SystemTap 2.7 SystemTap Beginners Guide,*" https://sourceware.org/systemtap/SystemTap_Beginners_Guide/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[12] "MySQL 5.0 Reference Manual", [Online]. Available: https://dev.mysql.com/doc/refman/5.0/en/linux-x86.html. [Accessed 6 Oct 2015]

[13] "Mongo DB Manual 3.0", [Online]. Available: https://docs.mongodb.org/manual/administration/production-notes/. [Accessed 6 Oct 2015]

[14] J. Spillner and A. Schill , "Orchestration of Distributed Storage Targets through Storage Flows", *2013 IEEE International Conference on Cloud Computing Technology and Science*, 2013, pp. 349-354.

[15] M. Huang, C. Huang and W.E. Chen, "Architecting a Software-Defined Storage Platform for Cloud Storage Service", *2015 IEEE International Conference on Services Computing*, 2015, pp. 375-386.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้