# การวัดและการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆ

ไกรยวิชช์ ศุภโสภาพงศ์

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2561

# การวัดและการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆ

ไกรยวิชช์ ศุภโสภาพงศ์

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2561

วิทยานิพนธ์ เรื่อง "การวัดและการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆ"
ของ ไกรยวิชช์ ศุภโสภาพงศ์
ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

**คณะกรรมการสอบวิทยานิพนธ์**

..... ประธานกรรมการสอบวิทยานิพนธ์
(ดร.พรเทพ โรจนวสุ)

..... ประธานที่ปรึกษาวิทยานิพนธ์
(ผู้ช่วยศาสตราจารย์ ดร.วินัย วงษ์ไทย)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.เกรียงศักดิ์ เตมีย์)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.ธนะธร พ่อค้า)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.จันทร์จิรา พยัคฆ์เพศ)

**อนุมัติ**

.....
(ศาสตราจารย์ ดร.ไพศาล มุณีสว่าง)

คณบดีบัณฑิตวิทยาลัย

**ชื่อเรื่อง** การวัดและการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆ

**ผู้วิจัย** ไกรยวิชช์ ศุภโสภาพงศ์

**ประธานที่ปรึกษา** ผู้ช่วยศาสตราจารย์ ดร. วินัย วงษ์ไทย

**ประเภทสารนิพนธ์** วิทยานิพนธ์ วท.ม. สาขาวิชาวิทยาการคอมพิวเตอร์, มหาวิทยาลัยนเรศวร, 2561

**คำสำคัญ** ระบบรักษาความปลอดภัย การประมวลผลแบบกลุ่มเมฆ ระบบบันทึกเหตุการณ์ การทดสอบประสิทธิภาพ

## บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อที่จะทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆหรือคลาวด์และทำการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์เพื่อให้มีประสิทธิภาพที่ดียิ่งขึ้น โดยระบบบันทึกเหตุการณ์เป็นหนึ่งในวิธีที่สามารถช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ซึ่งวิธีการปรับปรุงประสิทธิภาพดังกล่าวจำเป็นต้องอาศัยผลการวิเคราะห์จากการทดสอบประสิทธิภาพของฮาร์ดแวร์ที่ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ ทางผู้วิจัยจึงได้เลือกใช้วิธีการปรับปรุงประสิทธิภาพโดยกำหนดการทำงานของโปรเซสล็อกเกอร์ภายในหน่วยประมวลผลกลาง ซึ่งโปรเซสล็อกเกอร์เป็นโปรเซสที่สำคัญของระบบบันทึกเหตุการณ์ที่สามารถช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้และวิธีการปรับปรุงประสิทธิภาพดังกล่าวจำเป็นที่จะต้องใช้ผลวิเคราะห์จากการวัดและทดสอบประสิทธิภาพของฮาร์ดแวร์ที่ส่งผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ เมื่อได้ผลการวัดประสิทธิภาพของระบบบันทึกเหตุการณ์แล้วจึงจะสามารถดำเนินการปรับปรุงประสิทธิภาพต่อไปได้และผลลัพธ์จากการปรับปรุงประสิทธิภาพจะทำให้ระบบบันทึกเหตุการณ์สามารถตรวจจับเหตุการณ์ที่เกิดขึ้นภายในคลาวด์ได้รวดเร็วและมีความแม่นยำมากยิ่งขึ้น

**Title** MEASUREMENT AND IMPROVEMENT OF CLOUD LOGGING SYSTEMS

**Author** KRAIYAWICH SUPPASOPAPONG

**Advisor** Assistant Professor Winai Wongthai, Ph.D.

**Academic Paper** Thesis M.S. in Computer Science, Naresuan University, 2018

**Keywords** Security Cloud computing Longing system Performance measurement

## ABSTRACT

This research aimed for testing performance of logging systems on cloud computing and improvement of logging systems.The logging systems Can mitigate risk factors that may cause a threat to the cloud. So, the improvement of cloud logging systems requires results analysis from testing performance of hardware. How to improve the efficiency of the cloud logging systems that the researcher has chosen to use is configure working process logger on logging systems in Central Processing Unit (CPU). That process logger is an important of logging systems. So, Improvement of cloud logging systems derived from the result of analysis and measurement. When measuring the efficiency of the event recording system, then proceeding to improve efficiency in the next order.The result of logging system has can be monitoring are faster accurate and used in real work environments.

# ประกาศคุณูปการ

การดำเนินการจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาของ ผู้ช่วยศาสตราจารย์ ดร. วินัย วงษ์ไทย ประธานที่ปรึกษาวิทยานิพนธ์ ที่ได้สละเวลาอันมีค่ามาเป็นที่ปรึกษาได้ทุ่มเทอบรมสั่งสอนให้ความรู้ พร้อมทั้งให้คำปรึกษาในด้านเทคนิคต่างๆ ในศาสตร์ที่เกี่ยวข้องเพื่อคุณภาพของงานวิจัย ตลอดจนข้อเสนอแนะปรับปรุงพัฒนาการทำวิทยานิพนธ์ฉบับนี้ จนเสร็จสมบูรณ์เป็นอย่างดียิ่ง ผู้วิจัยต้องขอขอบพระคุณท่านเป็นอย่างสูง ไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณท่านคณาจารย์กรรมการสอบป้องกันวิทยานิพนธ์ทุกท่าน ประกอบด้วย ดร. วินัย วงษ์ไทย ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.เกรียงศักดิ์ เตมีย์ ดร.จันทร์จิรา พยัคฆ์เพศ ดร.ธนะธร พ่อค้า กรรมการผู้ทรงคุณวุฒิภายในทั้ง 3 ท่านดังกล่าว และ ดร.พรเทพ โรจนวสุ กรรมการผู้ทรงคุณวุฒิภายนอก ที่ได้กรุณาให้คำแนะนำ ข้อเสนอแนะ ปรับปรุงพัฒนาด้วยความเอาใจใส่ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์และทรงคุณค่า

ขอกราบขอบพระคุณอาจารย์ ดร.วันสุรีย์ มาศกรัม อาจารย์ที่ปรึกษาที่ได้ให้คำปรึกษาทั้งด้านการเรียนและด้านการทำวิทยานิพนธ์จนทำให้ผู้วิจัยสามารถดำเนินการทำวิทยานิพนธ์ฉบับนี้ได้สำเร็จและมีความสมบูรณ์ยิ่ง

ขอกราบขอบพระคุณ ดร.ธนะธร พ่อค้า อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วมที่คอยให้คำปรึกษาและให้ความรู้เกี่ยวกับหลักการทฤษฎีต่างๆ ทางศาสตร์วิชาที่เกี่ยวข้องกับการทำงานของระบบบันทึกเหตุการณ์ การทำงานของโค้ดในระบบบันทึกเหตุการณ์ และช่องโหว่ต่างๆ ของระบบบันทึกเหตุการณ์ ขอขอบคุณรุ่นพี่และเพื่อนๆ ในสาขาวิชาเทคโนโลยีสารสนเทศ วิทยาการคอมพิวเตอร์ รวมถึงทุกๆ คนที่คอยเป็นกำลังใจในการทำวิทยานิพนธ์ฉบับนี้จนประสบผลสำเร็จ

ผู้วิจัยจึงขอมอบส่วนคุณงามความดีทั้งหลายให้แด่คณาจารย์ที่ได้ประสิทธิ์ประสาทวิชาจนทำให้ผลงานวิทยานิพนธ์เป็นประโยชน์ต่อผู้ที่เกี่ยวข้อง และขอมอบความกตัญญูกตเวทิตาคุณแด่ บิดา มารดา และผู้มีพระคุณทุกท่านที่คอยให้ความห่วงใยและเป็นกำลังใจ ตลอดจนการสนับสนุนในทุกๆ ด้าน จนได้บรรลุผลการเรียนในระดับมหาบัณฑิตศึกษาครั้งนี้ สำหรับข้อบกพร่องต่างๆ ที่อาจจะเกิดขึ้นในงานวิทยานิพนธ์นั้น ผู้วิจัยขอน้อมรับและยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษาเพื่อเป็นประโยชน์ในการพัฒนางานวิจัยต่อไปในอนาคต

ไกรยวิชช์ ศุภโสภาพงศ์

# สารบัญ

หน้า

# สารบัญตาราง

**หน้า**

# สารบัญภาพ

หน้า

# บทที่ 1

# บทนำ

การประมวลแบบกลุ่มเมฆหรือคลาวด์คอมพิวติ้ง (Cloud computing) เป็นเทคโนโลยีที่ได้รับความนิยมในปัจจุบันและอนาคต โดยในงานวิทยานิพนธ์เล่มนี้จะใช้คำย่อว่า คลาวด์ (cloud) ซึ่งรูปแบบการให้บริการของคลาวด์จะให้การบริการทรัพยากรทางคอมพิวเตอร์ผ่านระบบอินเทอร์เน็ตหรืออินทราเน็ตและให้บริการเช่าทรัพยากรดังกล่าว เช่น พื้นที่ในการจัดเก็บข้อมูล โครงสร้างพื้นฐานทางคอมพิวเตอร์ ระบบปฏิบัติการ เครือข่ายอินเตอร์เน็ตและแอปพลิเคชัน โดยการเข้าถึงการบริการเหล่านี้จะต้องทำผ่านระบบอินเตอร์เน็ตหรืออินทราเน็ต นอกจากนี้รูปแบบการให้บริการคลาวด์ยังยืดหยุ่น สามารถลด เพิ่มหรือยกเลิกการใช้บริการของคลาวด์ได้ทันที ด้วยรูปแบบการให้บริการของคลาวด์ที่มีการให้เช่าบริการที่หลากหลายและยังยืดหยุ่นในการเข้าถึงการให้บริการ ทำให้ในปัจจุบันมีองค์กรต่าง ๆ สนใจที่จะนำคลาวด์ไปประยุกต์ใช้งานทางด้านไอที (Information Technology : IT) ภายในองค์กรต่าง ๆ

แต่อย่างไรก็ตามปัญหาทางด้านความปลอดภัยของคลาวด์ก็เป็นปัจจัยหลักสำหรับการนำคลาวด์มาพิจารณาเพื่อนำไปใช้งานจริงในองค์กรต่าง ๆ จึงทำให้มีองค์กรที่ได้ทำวิจัยเกี่ยวกับปัญหาทางด้านความปลอดภัยของคลาวด์เกิดขึ้น เช่น องค์กร Cloud Security Alliance (CSA) และจากปัญหาทางด้านความปลอดภัยของคลาวด์ที่ได้จากการวิจัยขององค์กร CSA ทำให้มีนักวิจัยได้ทำการศึกษาวิจัยเกี่ยวกับวิธีการป้องกันและวิธีการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ ซึ่งการป้องกันหรือบรรเทาปัจจัยเสี่ยงดังกล่าวจะสร้างความเชื่อมั่นต่อผู้ใช้บริการ[1]

ในงานวิทยานิพนธ์นี้มุ่งเน้นที่จะปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยระบบบันทึกเหตุการณ์เป็นระบบที่สามารถช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ ลักษณะการทำงานของระบบบันทึกเหตุการณ์จะทำให้ทราบว่า บุคคลใดเข้าถึงไฟล์อะไรบ้างหรือได้กระทำอะไรกับไฟล์เหล่านั้นบ้าง ระบบบันทึกเหตุการณ์ดังกล่าวจะเป็นส่วนหนึ่งในการช่วยให้ผู้มีส่วนเกี่ยวข้องสามารถดำเนินการสืบสวนหาตัวบุคคลที่กระทำผิดมารับผิดชอบต่อการกระทำดังกล่าวได้ ซึ่งการศึกษาการทำงานเกี่ยวกับระบบบันทึกเหตุการณ์เป็นสิ่งที่สำคัญของงาน

[1] ผู้ใช้บริการ หรือ customer สามารถเข้าใช้บริการที่ทางผู้ให้บริการได้จัดเตรียมไว้ทั้งแบบเสียค่าใช้จ่ายและไม่เสียค่าใช้จ่ายได้โดยผ่านทางเน็ตเวิร์ค

วิทยานิพนธ์นี้เพราะระบบบันทึกเหตุการณ์ยังมีปัญหาเรื่องประสิทธิภาพ ทำให้ระบบบันทึกเหตุการณ์ไม่สามารถบันทึกข้อมูลของเหตุการณ์ได้ทั้งหมด ดังนั้นในงานวิทยานิพนธ์นี้จึงมุ่งเน้นที่จะปรับปรุงประสิทธิภาพเพื่อทำให้ระบบบันทึกเหตุการณ์สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น โดยรายละเอียดของบทนำผู้วิจัยได้จำแนกเป็นหัวข้อต่าง ๆ ดังนี้

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ในหัวข้อนี้ ผู้วิจัยจะอธิบายความหมายของคลาวด์ ภาพรวมของคลาวด์ ปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ ความสำคัญของการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ วิธีการใดบ้างที่ใช้บรรเทาปัจจัยเสี่ยงเหล่านั้น รวมถึงการเพิ่มประสิทธิภาพของวิธีการดังกล่าวเพื่อให้มีประสิทธิภาพที่ดียิ่งขึ้นและเหมาะสมกับการนำไปใช้งานในสภาพแวดล้อมจริง โดยหัวข้อดังกล่าวมีวัตถุประสงค์เพื่อให้ผู้อ่านได้เห็นภาพรวมของวิทยานิพนธ์โดยมีรายละเอียด ดังนี้

### 1.1.1 ความหมายและภาพรวมของคลาวด์

หัวข้อนี้จะอธิบายเกี่ยวกับนิยามและความสำคัญของคลาวด์เพื่อให้ผู้อ่านได้เข้าใจเกี่ยวกับนิยามและความสำคัญของคลาวด์

ซึ่งผู้วิจัยได้ศึกษาและสรุปงานวิจัยของ Chiba และ Surbriyala โดยได้สรุปนิยามเกี่ยวกับคลาวด์ไว้ดังนี้ คลาวด์คอมพิวติ้งหรือคลาวด์ (Cloud computing or cloud) เป็นเทคโนโลยีที่มีการเจริญเติบโตอย่างรวดเร็วทางด้านเทคโนโลยีสารสนเทศ โดยมีอัตราการใช้งานที่เติบโตขึ้นทุกปี ซึ่งคลาวด์ได้ถูกนำไปใช้อย่างกว้างขวางในการจัดเก็บข้อมูลทรัพยากรต่าง ๆ นอกจากการให้บริการจัดเก็บข้อมูลทรัพยากรดังกล่าวแล้ว คลาวด์ยังมีการให้บริการเช่าทรัพยากรทางคอมพิวเตอร์อื่น ๆ อีกด้วย เช่น โครงสร้างพื้นฐานทางคอมพิวเตอร์ ระบบปฏิบัติการ เครือข่ายอินเตอร์เน็ตและแอปพลิเคชัน โดยรูปแบบการคิดค่าบริการการใช้งานต่อคลาวด์ส่วนใหญ่ คือ ผู้ใช้บริการเสียค่าใช้จ่ายเท่ากับจำนวนทรัพยากรที่ใช้ (Chiba, et al., 2016; Surbiryala, et al., 2017) ตัวอย่างของการให้บริการคลาวด์ เช่น Amazon Web Service (AWS) หรือ Dropbox หรือOffice 365 และอื่น ๆ (Surbiryala, et al., 2017) ด้วยเหตุผลข้างต้นที่คลาวด์มีลักษณะการให้บริการแบบเสียค่าใช้จ่ายเท่ากับจำนวนทรัพยากรที่ใช้ จึงทำให้สามารถช่วยลดค่าใช้จ่ายทางด้านไอทีขององค์กรต่าง ๆ ลงได้ ส่งผลให้หลายๆ องค์กรสนใจที่จะนำคลาวด์มาใช้งานด้านไอทีภายในองค์กร

จากลักษณะการให้บริการของคลาวด์ที่มีความยืดหยุ่นสามารถเพิ่มหรือลดทรัพยากรที่ใช้งานได้ตามความต้องการและยังเสียค่าใช้จ่ายตามบริการที่ใช้งานจริงแล้ว จึงทำให้มีอัตราการใช้บริการที่เพิ่มสูงขึ้น ในงานของ Chiba ยังกล่าวอีกว่าองค์กรต่าง ๆ ได้มีการนำคลาวด์มาประยุกต์ใช้งานด้านเทคโนโลยีสารสนเทศ เพราะเป็นทางเลือกที่ดีสำหรับองค์กรที่ต้องการลดต้นทุน แต่อย่างไรก็ตามปัญหาด้านความปลอดภัยก็เป็นปัญหาสำคัญ ต่อการนำคลาวด์มาใช้งานทางด้านไอทีให้ประสบผลสำเร็จภายในองค์กร โดยเป็นการให้เหตุผลจาก Chief Information Officer หรือ CIO ซึ่งเป็นเจ้าหน้าที่ด้านเทคโนโลยีสารสนเทศขององค์กรต่าง ๆ (Chiba, et al., 2016) นอกจากนี้ ในงานวิจัยของ Surbiraya ยังกล่าวว่าปัญหาด้านความปลอดภัยต่อคลาวด์เป็นสิ่งที่หลีกเลี่ยงไม่ได้สำหรับการนำไปพิจารณาเพื่อนำคลาวด์มาใช้งานจริง ดังนั้นการนำคลาวด์มาใช้งานจริงจึงต้องคำนึงถึงความปลอดภัย (Surbiryala, et al., 2017) และ Chiba ยังได้กล่าวอีกว่าความปลอดภัยต่อคลาวด์เป็นอุปสรรคในการนำคลาวด์มาใช้งานจริงในองค์กร (Chiba, et al., 2016)

จากปัญหาความเสี่ยงที่อาจก่อให้ภัยคุกคามต่อความปลอดภัยที่ของคลาวด์จึงได้มีองค์กรที่ทำการวิจัยและระบุปัญหาเกี่ยวกับภัยคุกคามต่อคลาวด์ นั่นคือ องค์กร Cloud Security Alliance หรือ CSA (CSA, 2016a, 2016b) โดยปัญหาที่ทางองค์กร CSA ได้ระบุไว้จะถูกอธิบายเพิ่มเติมในหัวข้อถัดไป

### 1.1.2 ปัญหาภัยคุกคามของคลาวด์

เนื่องจากปัญหาภัยคุกคามเป็นปัญหาที่สำคัญของคลาวด์ ในหัวข้อนี้จึงได้มีการอธิบายเพิ่มเติมเกี่ยวกับปัญหาภัยคุกคามของคลาวด์ รวมถึงปัญหาข้อใดที่ผู้วิจัยสนใจที่จะบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์

องค์กร CSA ได้ทำการวิจัยในเรื่องของปัญหาที่เกี่ยวกับภัยคุกคามต่อคลาวด์และได้จัดทำรายงานที่มีชื่อว่า Top Threats Cloud Computing V1.0 - Cloud Security Alliance ซึ่งเป็นรายงานฉบับแรกที่จัดทำขึ้นในปี 2010 (CSA, 2010) โดยรายงานฉบับแรกดังกล่าวทาง CSA ได้ระบุปัญหาภัยคุกคามของคลาวด์ไว้ 7 ข้อ แต่ยังไม่มีการจัดอันดับความรุนแรงเกี่ยวกับภัยคุกคามดังกล่าว ตัวอย่างภัยคุกคามของคลาวด์ที่องค์กร CSA ได้ระบุ เช่น ผู้ใช้บริการคลาวด์เพื่อนำไปใช้ในทางที่ผิด หรือ Abuse and Nefarious Use of Cloud Computing เป็นต้น ต่อมาทางองค์กร CSA ได้ทำการวิจัยอย่างต่อเนื่อง จนกระทั่งในปี 2013 จึงได้จัดทำรายงานที่ชื่อว่า The Notorious Nine: Cloud Computing Top Threats in 2013 (CSA, 2013) เป็นรายงานเกี่ยวกับภัยคุกคามต่อคลาวด์

9 หัวข้อ ซึ่งมีการเพิ่มภัยคุกคามของคลาวด์จากรายงานฉบับแรกที่มีเพียง 7 หัวข้อ นอกจากนี้ทางองค์กร CSA ยังได้จัดอันดับความรุนแรงของปัญหาภัยคุกคามที่เกิดขึ้นต่อคลาวด์ไว้ในรายงานล่าสุดเมื่อปี 2016 โดยใช้ชื่อว่า The Treacherous 12 Cloud Computing Top Threats in 2016 ซึ่งเป็นรายงานฉบับล่าสุด ในรายงานฉบับล่าสุดนี้ทางองค์กร CSA ได้ระบุเกี่ยวกับภัยคุกคามที่เกิดขึ้นต่อคลาวด์เป็น 12 หัวข้อ โดยรายละเอียดเพิ่มเติมถูกกล่าวในบทที่ 2 เรื่อง เอกสารและงานวิจัยที่เกี่ยวข้อง ในหัวข้อ 2.5 เรื่อง ปัญหาด้านความปลอดภัยของคลาวด์และในรายงานฉบับล่าสุดได้เรียงลำดับความรุนแรงของภัยคุกคามดังกล่าว โดยตัวอย่างของภัยคุกคามที่ทางองค์กร CSA ได้ระบุ เช่น การละเมิดข้อมูล หรือ Data Breaches เป็นภัยคุกคามที่ส่งผลความรุนแรงต่อคลาวด์เป็นอันดับ 1

จากภัยคุกคามที่เกิดขึ้นต่อคลาวด์ใน 12 หัวข้อตามรายงานขององค์กร CSA และวิทยานิพนธ์เล่มนี้สนใจเกี่ยวกับการบรรเทาภัยคุกคามที่อาจก่อให้เกิดปัญหาต่อคลาวด์ในหัวข้อที่ 1 คือการละเมิดข้อมูลด้วยวิธีการใช้ระบบบันทึกเหตุการณ์ในการบรรเทาปัจจัยเสี่ยงดังกล่าว ภาพรวมของระบบบันทึกเหตุการณ์จะถูกกล่าวหัวข้อถัดไป คือ หัวข้อที่ 1.1.3 เรื่อง ระบบบันทึกเหตุการณ์ และรายละเอียดของระบบบันทึกเหตุการณ์จะถูกกล่าวในบทที่ 2 ในหัวข้อที่ 2.9 เรื่อง ระบบบันทึกเหตุการณ์ (logging system)

จากปัญหาภัยคุกคามของคลาวด์ที่ทางองค์กร CSA ได้ระบุไว้ทำให้มีนักวิจัยหลายท่านได้ทำการศึกษาและเสนอวิธีการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ เช่น การนำคลาวด์ไปใช้ในทางที่ผิดหรือ Abuse and Nefarious Use of Cloud Computing เป็นต้น ซึ่งระบบบันทึกเหตุการณ์หรือ logging system เป็นหนึ่งในวิธีการที่สามารถนำไปใช้บรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ แต่ระบบดังกล่าวยังมีปัญหาเรื่องประสิทธิภาพ ดังนั้นผู้วิจัยจึงสนใจที่จะทำการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยระบบดังกล่าวได้ถูกใช้ในงานวิจัยของ Chan-in Wongthai Moorsel และ Ko (Chan-in & Wongthai, 2017; Ko, et al., 2011; Wongthai, 2014; Wongthai, et al., 2013b; Wongthai & van Moorsel, 2016) โดยในระบบบันทึกเหตุการณ์สามารถบันทึกความเป็นไปที่เกิดขึ้นต่อคลาวด์ได้ เช่น บันทึกว่าใครเคยเข้าถึงไฟล์อะไรหรือทำอะไรกับไฟล์เหล่านั้นบ้าง โดยข้อมูลที่ได้จากการบันทึกนี้สามารถนำไปใช้เป็นหลักฐานเมื่อเกิดปัญหาเกี่ยวกับความปลอดภัยต่อคลาวด์ ภาพรวมของระบบบันทึกเหตุการณ์จะถูกกล่าวในหัวข้อถัดไป

### 1.1.3 ระบบบันทึกเหตุการณ์และการนำระบบบันทึกเหตุการณ์ไปใช้เพื่อการวิจัย

ระบบบันทึกเหตุการณ์เป็นระบบที่สำคัญของวิทยานิพนธ์เล่มนี้เพราะเป็นระบบที่ผู้วิจัยได้ใช้ในการทดลองและผู้วิจัยต้องการที่จะปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์เพื่อให้เหมาะสมกับการนำไปใช้ในสภาพแวดล้อมจริง โดยในหัวข้อนี้จะอธิบายรายละเอียดเกี่ยวกับระบบบันทึกเหตุการณ์ ดังนี้

ในระบบบันทึกเหตุการณ์จะมีโปรเซสที่ใช้สำหรับบันทึกความเป็นไปที่เกิดขึ้นต่อคลาวด์เรียกว่า ล็อกเกอร์ (logger)[2] โดยวิทยานิพนธ์เล่มนี้ มุ่งเน้นเรื่องประสิทธิภาพการทำงานของ logger ในระบบบันทึกเหตุการณ์ดังกล่าวเมื่อมีการปรับขนาดของฮาร์ดแวร์ (hardware) ที่ใช้ติดตั้งระบบดังกล่าวอาจจะส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ ผู้วิจัยจึงได้ทำการทดสอบประสิทธิภาพของโปรเซส logger โดยมีการปรับขนาดของฮาร์ดแวร์ เช่น RAM (random access memory) หรือ CPU (central processing unit)[3] และนำผลการทดลองมาสรุป วิเคราะห์ อภิปรายผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ จากนั้นจะนำข้อมูลของการทดลองที่ผ่านการวิเคราะห์ อภิปรายผลแล้ว นำไปเป็นข้อมูลพื้นฐานสำหรับออกแบบการทดลองเพื่อปรับปรุงเพิ่มประสิทธิภาพให้กับโปรเซส logger[4] ในระบบบันทึกเหตุการณ์เพื่อให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพที่ดียิ่งขึ้นต่อไป โดยวิธีการทดสอบเกี่ยวกับประสิทธิภาพของโปรเซส logger เบื้องต้นจะถูกกล่าวในหัวข้อถัดไป

---

[2] logger หรือ ล็อกเกอร์ เป็นคำสั่งโค้ดในภาษา C ที่ยังไม่ถูกสั่งให้โปรเซสทำงาน

[3] CPU (central processing unit) ในงานวิจัยจะใช้คำว่า CPU core สาเหตุที่ผู้วิจัยใช้คำว่า CPU core เนื่องจากว่าผู้วิจัยได้ใช้คำว่า CPU core ตามเอกสารของ Dan Nanni (Nanni) ซึ่งเป็นเอกสารเกี่ยวกับการรันโปรแกรมหรือโปรเซสบนระบบปฏิบัติการลินุกส์หรือ linux

[4] process logger หรือ โปรเซสล็อกเกอร์ เป็นคำสั่งโค้ดในภาษา C ที่ถูกรันจาก logger เพื่อใช้สำหรับบันทึกเหตุการณ์ที่เกิดขึ้นต่อคลาวด์

### 1.1.4 การทดสอบประสิทธิภาพของโปรเซส logger ในระบบบันทึกเหตุการณ์เบื้องต้น

การทดสอบประสิทธิภาพถือว่าเป็นปัจจัยสำคัญของวงจรการพัฒนาระบบ (SDLC : System development Life Cycle) (Molyneaux, 2014) เพราะการทดสอบประสิทธิภาพของระบบทำให้ทราบเกี่ยวกับสมรรถนะของระบบและสามารถวางแผนการพัฒนาหรือสามารถปรับปรุงระบบดังกล่าวในอนาคตได้ ดังนั้นการทดสอบประสิทธิภาพของโปรเซส logger จึงเป็นวัตถุประสงค์สำคัญของวิทยานิพนธ์นี้ซึ่งจะกล่าวเพิ่มเติมในหัวข้อที่ 1.2 เรื่อง วัตถุประสงค์ของการศึกษา ส่วนหัวข้อนี้จะอธิบายรายละเอียดเบื้องต้นเกี่ยวกับการทดสอบประสิทธิภาพของโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์และเครื่องมือที่ใช้ในการทดลองของวิทยานิพนธ์เล่มนี้

จากหัวข้อ 1.1.3 เรื่อง ระบบบันทึกเหตุการณ์ และได้กล่าวว่างานวิทยานิพนธ์เล่มนี้มุ่งเน้นเรื่องการวัดและปรับปรุงประสิทธิภาพการทำงานของโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์เมื่อมีการปรับขนาดของฮาร์ดแวร์ที่ส่งผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ และจะวิเคราะห์ผลกระทบที่เกิดขึ้นกับโปรเซส logger ในระบบบันทึกเหตุการณ์ โดยในย่อหน้านี้จะอธิบายเกี่ยวกับรูปแบบการทดสอบประสิทธิภาพเพิ่มเติม คือ จากงานของ Molyneaux ได้อธิบายว่า Key Performance Indicator หรือ KPI สำหรับการทดสอบประสิทธิภาพของเว็บแอปพลิเคชันมี 4 หัวข้อ คือ 1. สภาพพร้อมใช้งาน (availability) 2. เวลาตอบสนอง (response time) 3. ปริมาณงานที่ทำในช่วงเวลาหนึ่ง (throughput) และ 4. ความจุ (capacity) (Molyneaux, 2014) โดยเครื่องมือสำหรับการทดสอบประสิทธิภาพที่ใช้ในงานวิทยานิพนธ์เล่มนี้ จะอยู่ในหัวข้อที่ 2 คือ เวลาตอบสนอง หรือ response time

สาเหตุที่ผู้วิจัยได้เลือก KPI ที่ 2 มีจุดประสงค์เพื่อเป็นตัวอย่างของ KPI ในการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ และทางผู้วิจัยมุ่งหวังว่าในอนาคตจะได้ทำการทดสอบประสิทธิภาพของ KPI ที่ 1 3 และ 4 ต่อไป ผู้วิจัยจึงได้เลือกการทดสอบเพียง KPI ที่ 2 ซึ่งอาจจะใช้เป็นแนวทางสำหรับวิธีการทดสอบ KPI แบบอื่น ๆ ต่อไปในอนาคตได้

ในย่อหน้าที่ผ่านมาได้อธิบายรายละเอียดเบื้องต้นเกี่ยวกับการทดสอบประสิทธิภาพของโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ และการทดลองที่คาดว่าจะทำ รวมถึงแนวทางเบื้องต้นสำหรับการวิเคราะห์ข้อมูลที่เกี่ยวกับผลกระทบของขนาดของฮาร์ดแวร์ที่ใช้ติดตั้งระบบบันทึกเหตุการณ์ต่อประสิทธิภาพที่เกิดขึ้น ขั้นตอนต่อไป คือ การนำข้อมูลที่ผ่านกระบวนการวิเคราะห์มาสรุปหาสาเหตุเกี่ยวกับผลกระทบด้านประสิทธิภาพที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์ เมื่อทราบสาเหตุดังกล่าวแล้ว จะทำให้ผู้วิจัยสามารถพิจารณาเกี่ยวกับประสิทธิภาพของระบบบันทึกเหตุการณ์ได้หลังมีการปรับขนาดของฮาร์ดแวร์ และสามารถวางแผนปรับปรุงประสิทธิภาพ

ในขั้นตอนต่อไปได้ ดังนั้น กระบวนการทดสอบประสิทธิภาพด้วยวิธีการนำข้อมูลที่ได้ไปวิเคราะห์หาสาเหตุและหาวิธีการปรับปรุงประสิทธิภาพ จึงสามารถอธิบายรายละเอียดได้ ดังนี้

1) ทดสอบประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์

2) นำข้อมูลจากการทดสอบไปวิเคราะห์ สรุปหาสาเหตุเกี่ยวกับผลกระทบที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์

3) หาวิธีการปรับปรุงโดยการวิเคราะห์หาสาเหตุจากผลกระทบที่เกิดขึ้นจากประสิทธิภาพของฮาร์ดแวร์เมื่อมีการเปลี่ยนแปลง

หลังจากที่ทราบสาเหตุเกี่ยวกับผลกระทบที่เกิดขึ้นดังกล่าวแล้ว ทางผู้วิจัยจึงสามารถตรวจสอบการทำงานของโปรเซส logger บนระบบปฏิบัติการและใช้วิธีการกำหนดพื้นที่การทำงานและจัดลำดับการทำงานของโปรเซส logger ให้กับ CPU core ได้ โดยเลือกใช้เครื่องมือ taskset ในระบบปฏิบัติการลินุกซ์ (linux) เหตุผลที่ผู้วิจัยได้เลือกใช้เครื่องมือดังกล่าวก็คือการใช้เครื่องมือ taskset ในระบบปฏิบัติการ linux มากำหนดพื้นที่การทำงานและจัดลำดับการทำงานของโปรเซส logger บนระบบบันทึกเหตุการณ์เพื่อปรับปรุงประสิทธิภาพ เนื่องจากว่าการทอสอบระบบบันทึกเหตุการณ์ของ Pakon และ Winai (Chan-in & Wongthai, 2017; Wongthai, et al., 2013b) ไม่ได้มีการจัดสรรพื้นที่การทำงานให้เหมาะสมกับโปรเซส logger ทำให้เมื่อโปรเซส logger ทำงานอาจจะส่งผลต่อประสิทธิภาพได้โดยตรง ซึ่งวิธีการใช้เครื่องมือ taskset ได้อธิบายเพิ่มเติมในบทที่ 3 เรื่อง วิธีการดำเนินการวิจัย ในหัวข้อที่ 3.3 การออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพ

สำหรับการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์นั้นสิ่งที่จำเป็น คือ ต้องมีข้อสรุปและผลวิเคราะห์จากผลกระทบที่เกิดขึ้นต่อประสิทธิภาพของโปรเซส logger ในระบบบันทึกเหตุการณ์ เพื่อนำข้อสรุปและผลวิเคราะห์ดังกล่าวไปพิจารณาและหาวิธีปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ ดังนั้นการวิเคราะห์ส่วนประกอบต่าง ๆ ของฮาร์ดแวร์ที่ส่งผลกระทบต่อโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์จึงมีความสำคัญและจำเป็นต้องมีการทดสอบประสิทธิภาพ เช่น ทดสอบเกี่ยวกับผลกระทบของหน่วยความจำหลัก (RAM) ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ หรือผลกระทบของหน่วยประมวลผลกลาง (CPU core) ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยข้อมูลจากการทดสอบเหล่านี้ สามารถนำไปเป็นข้อสรุปและวิเคราะห์ผลกระทบที่มีต่อโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์เพื่อนำไปเป็นข้อมูลพื้นฐานสำหรับการออกแบบเพื่อปรับปรุงประสิทธิภาพให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์ดังกล่าวนี้ต่อไป

### 1.1.5 ปัญหาวิจัยหลักของวิทยานิพนธ์ (research gaps)

เป้าหมายหลักของวิทยานิพนธ์ คือ สิ่งที่ผู้วิจัยได้ทำการศึกษาและได้ระบุเพื่อให้บรรลุวัตถุประสงค์ของงานวิจัยและสามารถออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ได้

รายละเอียดของปัญหาวิจัยหลักหรือ research gap ของวิทยานิพนธ์เล่มนี้สามารถอธิบายเป็นรายข้อได้ 3 หัวข้อ โดย research gap ทุกหัวข้อจะสอดคล้องกับวัตถุประสงค์ทั้งหมดของงานวิทยานิพนธ์เล่มนี้ รายละเอียดของวัตถุประสงค์จะถูกกล่าวในหัวข้อถัดไปหรือหัวข้อที่ 1.2 เรื่อง วัตถุประสงค์ของการศึกษาซึ่ง research gap จะมีรายละเอียด ดังนี้

1) จากการศึกษาวรรณกรรมที่เกี่ยวข้อง (literature reviews) ที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์ยังมีงานวิจัยแค่ส่วนน้อยได้ทำการทดสอบวัดประสิทธิภาพของหน่วยประมวลผลกลาง (CPU core) โดยการปรับขนาดของ CPU core และงานวิจัยดังกล่าวมีการทดลองไม่ครอบคลุมการวัดประสิทธิภาพของ CPU core แบบสมบูรณ์

2) จากการศึกษาวรรณกรรมที่เกี่ยวข้อง (literature reviews) ที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์ ยังไม่ปรากฏงานทดลองใดได้ทำการทดสอบวัดประสิทธิภาพของหน่วยความจำหลัก (RAM) โดยการปรับขนาดของ RAM

สาเหตุของการทดสอบประสิทธิภาพของข้อ 1.1.5.1 และ 1.1.5.2 เนื่องจากในงานวิจัยของ Wongthai (Wongthai & Moorsel, 2016) ได้ให้ข้อเสนอแนะว่าการเพิ่มประสิทธิภาพของฮาร์ดแวร์ เช่น RAM หรือ CPU core อาจจะส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น แต่ในงานวิทยานิพนธ์ดังกล่าวยังไม่มีโอกาสได้ทดสอบประสิทธิภาพและปรับปรุงประสิทธิภาพเนื่องด้วยเหตุผลทางด้านเวลาของผู้วิจัย

3) จากการศึกษาวรรณกรรมที่เกี่ยวข้อง (literature reviews) ยังมีงานวิจัยแค่ส่วนน้อยได้ปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และงานวิจัยดังกล่าวยังมีการทดลองไม่ครอบคลุมการวัดประสิทธิภาพของ CPU core และ RAM แบบสมบูรณ์

สาเหตุของการทดสอบประสิทธิภาพข้อ 1.1.5.3 จากงานวิจัยของ Molyneaux ได้กล่าวว่าการปรับปรุงประสิทธิภาพของระบบถือว่าเป็นปัจจัยสำคัญของวงจรพัฒนาระบบ (Molyneaux, 2014) และถือว่าเป็นทรัพย์สินที่มีค่า เพราะว่าหากไม่มีการทดสอบประสิทธิภาพของซอฟต์แวร์แล้วจะถือว่าซอฟต์แวร์นั้นเป็นทรัพยากรที่ไร้ค่า (Ebert, et al., 2005) ดังนั้นการปรับปรุง

ประสิทธิภาพของระบบบันทึกเหตุการณ์จึงเป็นเรื่องสำคัญ นอกจากนี้ความสำคัญในการทดสอบประสิทธิภาพจะถูกอธิบายเพิ่มเติมในบทที่ 2 เอกสารที่เกี่ยวข้อง หัวข้อ 2.8 การทดสอบประสิทธิภาพ

### 1.1.6 สรุปภาพรวมหัวข้อ 1.1

หัวข้อนี้จะกล่าวถึงภาพรวมทั้งหมดของความเป็นมาและความสำคัญของปัญหาโดยมีรายละเอียด ดังนี้

หัวข้อ 1.1.1 เรื่อง ความหมายและภาพรวมของคลาวด์ ในหัวข้อนี้ได้อธิบายเกี่ยวกับนิยามและความสำคัญของคลาวด์ โดยลักษณะการให้บริการของคลาวด์มีความยืดหยุ่น สามารถเพิ่มหรือลดทรัพยากรที่ใช้งานได้ตามความต้องการของผู้ใช้งาน ด้วยลักษณะดังกล่าวทำให้มีองค์กรต่างๆ ให้ความสนใจต้องการนำคลาวด์มาประยุกต์ใช้งานด้านเทคโนโลยีสารสนเทศภายในองค์กร นอกจากนี้การนำคลาวด์มาประยุกต์ใช้งานภายในองค์กรสิ่งที่ต้องคำนึงถึงคือความปลอดภัย เมื่อความปลอดภัยเป็นสิ่งที่ต้องคำนึงถึงในการนำคลาวด์มาใช้ในองค์กร จึงทำให้มีองค์กรที่ได้ทำวิจัยและระบุปัญหาเกี่วกับภัยคุกคามของคลาวด์ซึ่งก็คือองค์กร CSA

หัวข้อ 1.1.2 เรื่อง ปัญหาภัยคุกคามของคลาวด์ โดยปัญหาเกี่ยวกับภัยคุกคามนั้นเป็นปัญหาสำคัญของคลาวด์ ดังที่กล่าวไว้ในบทสรุปในหัวข้อ 1.1.1 เรื่องความหมายและภาพรวมของคลาวด์ โดยปัญหาที่เกี่ยวกับภัยคุกคามต่อคลาวด์ที่ทาง CSA ได้วิจัยอย่างต่อเนื่องจนถึงปี 2016 ทางองค์กร CSA ได้ระบุปัญหาภัยคุกคามไว้ 12 หัวข้อและเรียงลำดับตามความรุนแรงโดยเป็นรายงานฉบับล่าสุด โดยรายงานฉบับดังกล่าวผู้วิจัยสนใจที่จะบรรเทาภัยคุกคามที่อาจก่อให้เกิดปัญหาต่อคลาวด์ ด้วยวิธีการใช้ระบบบันทึกเหตุการณ์ โดยรายละเอียดของระบบบันทึกเหตุการณ์สามารถสรุปภาพรวมของลักษณะการทำงานได้ในหัวข้อ 1.1.3 เรื่อง ระบบบันทึกเหตุการณ์

หัวข้อ 1.1.3 เรื่อง ระบบบันทึกเหตุการณ์ ในหัวข้อนี้ผู้วิจัยได้อธิบายหลักการทำงานเบื้องต้นของระบบบันทึกเหตุการณ์ว่า สามารถบันทึกเหตุการณ์ที่เกิดขึ้นบนคลาวด์ เพื่อนำเหตุการณ์ที่บันทึกไปเป็นหลักฐานเพื่อหาบุคคลที่กระทำผิดมารับผิดชอบต่อสิ่งที่เกิดขึ้น โดยโปรเซสที่ใช้ในการบันทึกเหตุการณ์ เรียกว่า ล็อกเกอร์ (logger) และในงานวิจัยนี้มุ่งเน้นที่จะทดสอบประสิทธิภาพด้วยการปรับขนาดของฮาร์ดแวร์ (hardware) ที่ส่งผลกระทบต่อการทำงานของ logger และหาวิธีการปรับปรุงประสิทธิภาพดังกล่าว ซึ่งจะถูกกล่าวในย่อหน้าข้อถัดไป

หัวข้อ 1.1.4 เรื่อง การทดสอบประสิทธิภาพของโปรเซส logger ในระบบบันทึกเหตุการณ์ โดยการทดสอบประสิทธิภาพเป็นจุดประสงค์หลักของวิทยานิพนธ์เล่มนี้ดังนั้นสิ่งสำคัญในการทดสอบ

ประสิทธิภาพ คือ ข้อมูลที่ได้จากการทดสอบประสิทธิภาพของโปรเซส logger ซึ่งข้อมูลดังกล่าวจะสามารถนำไปวิเคราะห์หาสาเหตุของผลกระทบจากฮาร์ดแวร์ ที่มีต่อระบบบันทึกเหตุการณ์ และเมื่อทราบสาเหตุดังกล่าวจะสามารถออกแบบวิธีการปรับปรุงประสิทธิภาพของ logger บนระบบบันทึกเหตุการณ์ได้ และเพื่อให้บรรลุวัตถุประสงค์ของงานวิจัยนี้ ผู้วิจัยจึงได้ระบุปัญหาวิจัยหลักของวิทยานิพนธ์ (research gap)

ในหัวข้อ 1.1.5 เรื่อง ปัญหาวิจัยหลัก (research gap) หัวข้อนี้ได้ระบุเป้าหมายของวิทยานิพนธ์เล่มนี้ เพื่อให้สอดคล้องและบรรลุวัตถุประสงค์ของงานวิทยานิพนธ์

## 1.2 วัตถุประสงค์ของการศึกษา

จากหัวข้อที่ 1.1 เรื่อง ความเป็นมาและความสำคัญของปัญหา ได้กล่าวถึงความสำคัญของการทดสอบประสิทธิภาพและวิเคราะห์ผลกระทบจากฮาร์ดแวร์ที่อาจส่งผลกระทบต่อ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ซึ่งมีความสำคัญและเป็นวัตถุประสงค์ของวิทยานิพนธ์เล่มนี้ ในหัวข้อนี้จึงได้ระบุวัตถุประสงค์ทั้งหมดของวิทยานิพนธ์เล่มนี้ ซึ่งจะสอดคล้องกับปัญหาวิจัยหลักของวิทยานิพนธ์ (research gaps) ที่กล่าวในหัวข้อ 1.1.5 ทั้งสามข้อโดยรายละเอียดเกี่ยวกับวัตถุประสงค์ของการศึกษา มีดังนี้

1.2.1 เพื่อทดสอบประสิทธิภาพและวิเคราะห์ผลกระทบของหน่วยความจำหลักหรือ RAM ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์บนคลาวด์

1.2.2 เพื่อทดสอบประสิทธิภาพและวิเคราะห์ผลกระทบของหน่วยประมวลผลกลางหรือ CPU core ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์บนคลาวด์

1.2.3 เพื่อปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ จากผลการวิเคราะห์ที่เกิดขึ้นจากหน่วยความจำหลักและหน่วยประมวลผลกลางที่ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์

## 1.3 ขอบเขตของการวิจัย

เพื่อให้บรรลุวัตถุประสงค์ของงานวิจัยในวิทยานิพนธ์เล่มนี้ ผู้วิจัยจึงได้กำหนดขอบเขตการวิจัย โดยมีรายละเอียดดังหัวข้อต่อไปนี้

### 1.3.1 ขอบเขตด้านเทคโนโลยี

1) เครื่องคอมพิวเตอร์ (personal computer) Xeon 1 เครื่อง สำหรับการทดลองสร้างคลาวด์ IaaS โดยติดตั้งระบบปฏิบัติการ Linux Fedora 16 ขนาด 64 bit

2) ซอฟต์แวร์ (software) Xen 4.1-4-unstable บนระบบปฏิบัติการ Fedora 16 ขนาด 64 bit เพื่อจำลองสภาพแวดล้อมการทำงานของคลาวด์ IaaS

3) ติดตั้งระบบบันทึกเหตุการณ์ (logging system) บนคอมพิวเตอร์ Intel Xeon w 3550 3.06 GH สำหรับผู้ให้บริการ[5] ซึ่งใช้จำลองสร้างคลาวด์ IaaS และทดสอบการทำงานขณะตรวจสอบไฟล์

### 1.3.2 ขอบเขตด้านวิธีการ

1) ทำการจำลองสร้างคลาวด์ IaaS บนเครื่องคอมพิวเตอร์ 1 เครื่องพร้อมติดตั้งระบบบันทึกเหตุการณ์ (logging system)

2) ทำการจำลองพฤติกรรมแฮกเกอร์หรือ hacker เมื่อแฮกเกอร์เหล่านี้เข้าเปิดอ่านไฟล์สำคัญของลูกค้าขณะที่ระบบบันทึกเหตุการณ์ตรวจสอบไฟล์ที่อยู่ในคลาวด์ IaaS โดยจำลองแฮกเกอร์เพียงคนเดียว

## 1.4 นิยามศัพท์เฉพาะ

นิยามศัพท์เฉพาะเป็นคำที่ถูกกำหนดเพื่อช่วยในการสืบค้นและสามารถเข้าถึงงานวิจัยในวิทยานิพนธ์เล่มนี้ได้สะดวกรวดเร็วยิ่งขึ้น โดยนิยามศัพท์เฉพะของวิทยานิพนธ์ คือ การรรักษาความปลอดภัย การประมวลผลแบบกลุ่มเมฆ ระบบบันทึกเหตุการณ์และการทดสอบประสิทธิภาพ โดยมีรายละเอียดดังนี้

---

[5] ผู้ให้บริการ หรือ provider เป็นผู้ที่เตรียมบริการประมวลผลแบบกลุ่มเมฆไว้ เช่น เน็ตเวิร์ก เครื่องเซิฟเวอร์ แหล่งเก็บข้อมูล แอปพลิเคชันและบริการอื่น ๆ ซึ่งบริการนี้เรียกรวมว่า Cloud Computing หรือ การประมวลผลแบบกลุ่มเมฆ

### 1.4.1 การรักษาความปลอดภัย (Security)

การรักษาความปลอดภัย (Security) หมายความว่า ขั้นตอน เทคนิค วิธีการรักษา ดูแลและป้องกันอันตรายต่อข้อมูลหรือการทำงาน ซึ่งการรักษาความปลอดภัยของข้อมูลถือว่าเป็นเรื่องที่สำคัญ เนื่องจากข้อมูลถือว่าเป็นความลับของผู้ใช้บริการ ผู้ที่สามารถเข้าถึงข้อมูลได้จะต้องเป็นผู้ที่มีสิทธิ์เข้าถึงหรือเจ้าของข้อมูลเท่านั้น ดังนั้นการรักษาความปลอดภัยของข้อมูลจึงเป็นสิ่งที่สำคัญ นอกจากนี้การรักษาความปลอดภัยถือว่าเป็นกระบวนการหนึ่งที่จะช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ ดังนั้นการรักษาความปลอดภัยจึงเป็นเรื่องที่สำคัญสำหรับวิทยานิพนธ์เล่มนี้

### 1.4.2 การประมวลผลแบบกลุ่มเมฆ (cloud computing or cloud)

การประมวลผลแบบกลุ่มเมฆหรือคลาวด์ หมายความว่า การบริการที่ครอบคลุมถึงการให้ใช้งานหน่วยจัดเก็บข้อมูล โครงสร้างพื้นฐานทางคอมพิวเตอร์หรืออื่น ๆ จากผู้ให้บริการโดยการเข้าถึงต้องผ่านระบบอินเตอร์เน็ตหรืออินทราเน็ต โดยรูปแบบของคลาวด์จะสามารถลดความยุ่งยากในการติดตั้ง ดูแลระบบ ทำให้สามารถช่วยประหยัดเวลา และลดต้นทุนในการสร้างระบบคอมพิวเตอร์และเครือข่าย โดยมีทั้งรูปแบบเสียค่าบริการและไม่เสียค่าบริการ ในวิทยานิพนธ์เล่มนี้ได้นำเสนอวิธีการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ ซึ่งระบบบันทึกเหตุการณ์ดังกล่าวยังเป็นวิธีการหนึ่งที่ช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามที่มีต่อคลาวด์ ดังนั้นการทราบข้อมูลพื้นฐานของคลาวด์หรือการประมวลผลแบบกลุ่มเมฆจึงมีความสำคัญ โดยข้อมูลพื้นฐาน นิยาม ประเภทของคลาวด์จะถูกกล่าวในบทที่ 2 เอกสารที่เกี่ยวข้อง หัวข้อ 2.1 ความเป็นมาและนิยามของคลาวด์

### 1.4.3 ระบบบันทึกเหตุการณ์ (logging system)

ระบบบันทึกเหตุการณ์ หมายความว่า ระบบที่สามารถบันทึกข้อมูลความเป็นไปในระบบคลาวด์ เช่น การเข้าถึง ปรับปรุง เปลี่ยนแปลง แก้ไขและลบข้อมูล ข้อมูลที่ได้จากการบันทึกข้อมูลของระบบบันทึกเหตุการณ์นี้ สามารถนำไปใช้เป็นหลักฐานในการสืบหาผู้รับผิดชอบเมื่อเกิดปัญหาหรือความเสียหายต่อคลาวด์ และระบบบันทึกเหตุการณ์เป็นวิธีการหนึ่งที่สามารถช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ซึ่งถือว่าระบบบันทึกเหตุการณ์เป็นวิธีการที่สำคัญของ

วิทยานิพนธ์เล่มนี้ โดยรายละเอียดเพิ่มเติมเกี่ยวกับระบบบันทึกเหตุการณ์จะกล่าวในบทที่ 2 เอกสารที่เกี่ยวข้อง หัวข้อ 2.7 ระบบบันทึกเหตุการณ์

1.4.4 การทดสอบประสิทธิภาพ (performance measurement)

การทดสอบประสิทธิภาพเป็นปัจจัยสำคัญของวงจรพัฒนาระบบ SDLC (SDLC : System development Life Cycle) (Molyneaux, 2014) ถ้าหากไม่มีการทดสอบประสิทธิภาพในระบบ จะส่งผลให้องค์กรขาดผลกำไรตามที่ขาดหวังได้ เนื่องจากการทดสอบประสิทธิภาพทำให้ทราบข้อมูลเกี่ยวกับสมรรถนะของระบบจนสามารถวางแผนต่อผลกำไรล่วงหน้าขององค์กรได้ นอกจากนี้การทดสอบประสิทธิภาพถือว่าเป็นวัตถุประสงค์หลักของงานวิจัยเล่มนี้ เพราะว่าวัตถุประสงค์ของงานวิจัยนี้ต้องมีการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์และนำไปวิเคราะห์ผลกระทบเพื่อเป็นข้อมูลสำหรับการออกแบบเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ จากนั้นจึงจะสามารถดำเนินการขั้นต่อไปได้ โดยรายละเอียดเพิ่มเติมของหัวข้อนี้จะถูกอธิบายเพิ่มเติมในบทที่ 2 เอกสารที่เกี่ยวข้อง

จากนิยามศัพท์เฉพาะของวิทยานิพนธ์ที่ผู้วิจัยได้ระบุ ซึ่งถูกอธิบายไว้ในหัวข้อดังกล่าว โดยทุกๆ หัวข้อของนิยามศัพท์เฉพาะทางผู้วิจัยได้ศึกษางานวิจัยที่เกี่ยวข้องและอธิบายเพิ่มเติมในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง

## 1.5 สมมติฐานการวิจัย

เพื่อให้บรรลุวัตถุประสงค์ของการศึกษาที่กล่าวไว้ในหัวข้อที่ 1.2 เรื่อง วัตถุประสงค์ของการศึกษาของบทที่ 1 บทนำ ผู้วิจัยได้ทำการตั้งสมมติฐานไว้ดังนี้

1.5.1 การเพิ่มขนาดของ RAM ในส่วนของผู้ให้บริการและผู้ใช้บริการอาจจะส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น

1.5.2 การเพิ่มขนาดของ CPU core ในส่วนของผู้ให้บริการและผู้ใช้บริการอาจจะส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น

1.5.3 ประสิทธิภาพของระบบบันทึกเหตุการณ์อาจเกิดขึ้นจากระบบปฏิบัติการหรือ OS ได้จัดสรรพื้นที่ในการทำงานให้โปรเซส logger ได้ประมวลผลในการทำงาน

## 1.6 กรอบการดำเนินงานวิจัย

จากวัตถุประสงค์ของงานวิจัยในหัวข้อที่ 1.2 ทั้ง 3 หัวข้อและเพื่อให้บรรลุวัตถุประสงค์ทั้ง 3 หัวข้อนั้น ผู้วิจัยได้กำหนดกรอบแนวคิดการวิจัยเพื่อให้บรรลุวัตถุประสงค์ดังกล่าว โดยมีรายละเอียดดังนี้

วิทยานิพนธ์เล่มนี้มุ่งเน้นการวัดประสิทธิภาพโดยการปรับขนาดของฮาร์ดแวร์ ในส่วนของ RAM และ CPU core ที่คาดว่าจะส่งผลกระทบต่อประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ และนำผลการทดลองที่ได้มาสรุป วิเคราะห์ผลกระทบที่เกิดขึ้นเพื่อนำไปเป็นข้อมูลในการออกแบบวิธีการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ โดยสามารถแบ่งการทดลองและการออกแบบระบบเพื่อเพิ่มประสิทธิภาพได้ 2 ส่วนหลัก คือ 1) ส่วนของการทดลองและนำผลการทดลองที่ได้มาสรุป และวิเคราะห์ผลการทดลอง 2) ส่วนของการออกแบบวิธีการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ โดยสามารถอธิบายรายละเอียดของ 2 ส่วนหลักดังกล่าวได้ ดังหัวข้อ 1.6.1 และ 1.6.2 ดังนี้

### 1.6.1 ส่วนของการทดลองและนำผลทดลองที่ได้มาสรุป วิเคราะห์ผลการทดลอง

โดยในการทดลองได้ออกแบบการทดลองเป็น 2 ส่วนเพื่อให้สอดคล้องกับวัตถุประสงค์ที่ 1.2.1 และ 1.2.2 ตามลำดับดังนี้ คือ ทดสอบประสิทธิภาพของ RAM ที่ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ และ ทดสอบประสิทธิภาพของ CPU core ที่ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยมีรายละเอียดดังนี้

ก่อนจะอธิบายรายละเอียดของการทดลองและออกแบบระบบเพื่อเพิ่มประสิทธิภาพในแต่ละส่วน ทางผู้วิจัยขออธิบายเกี่ยวกับคำว่าผู้ให้บริการและผู้ใช้บริการก่อนเพื่อให้ผู้อ่านได้เข้าใจเกี่ยวกับผู้ให้บริการและผู้ใช้บริการเบื้องต้น ดังนี้

1) ทดสอบประสิทธิภาพของ RAM ที่ส่งผลกระทบต่อระบบบันทึกเหตุการณ์ทำการทดลองด้วยวิธีการปรับขนาดของ RAM ในส่วนของเครื่องคอมพิวเตอร์ผู้ให้บริการและของผู้ใช้บริการ จากนั้นสรุปและวิเคราะห์ผลการทดลองโดยแบ่งเป็น

1.1) ทดสอบปรับขนาด RAM ในส่วนของคอมพิวเตอร์ของผู้ให้บริการที่ติดตั้งลง logger

1.2) ทดสอบปรับขนาด RAM ในส่วนของคอมพิวเตอร์ของผู้ใช้บริการ

2) ทดสอบประสิทธิภาพของหน่วยประมวลผลกลาง (CPU core) ที่ส่งผลกระทบต่อระบบบันทึกเหตุการณ์ ทำการทดลองโดยวิธีการเดียวกันกับการทดสอบประสิทธิภาพของ RAM นั่นคือ วิธีปรับขนาดของ CPU core ในส่วนของผู้ใช้บริการและผู้ให้บริการ แล้วสรุปผลและวิเคราะห์ผลการทดลอง

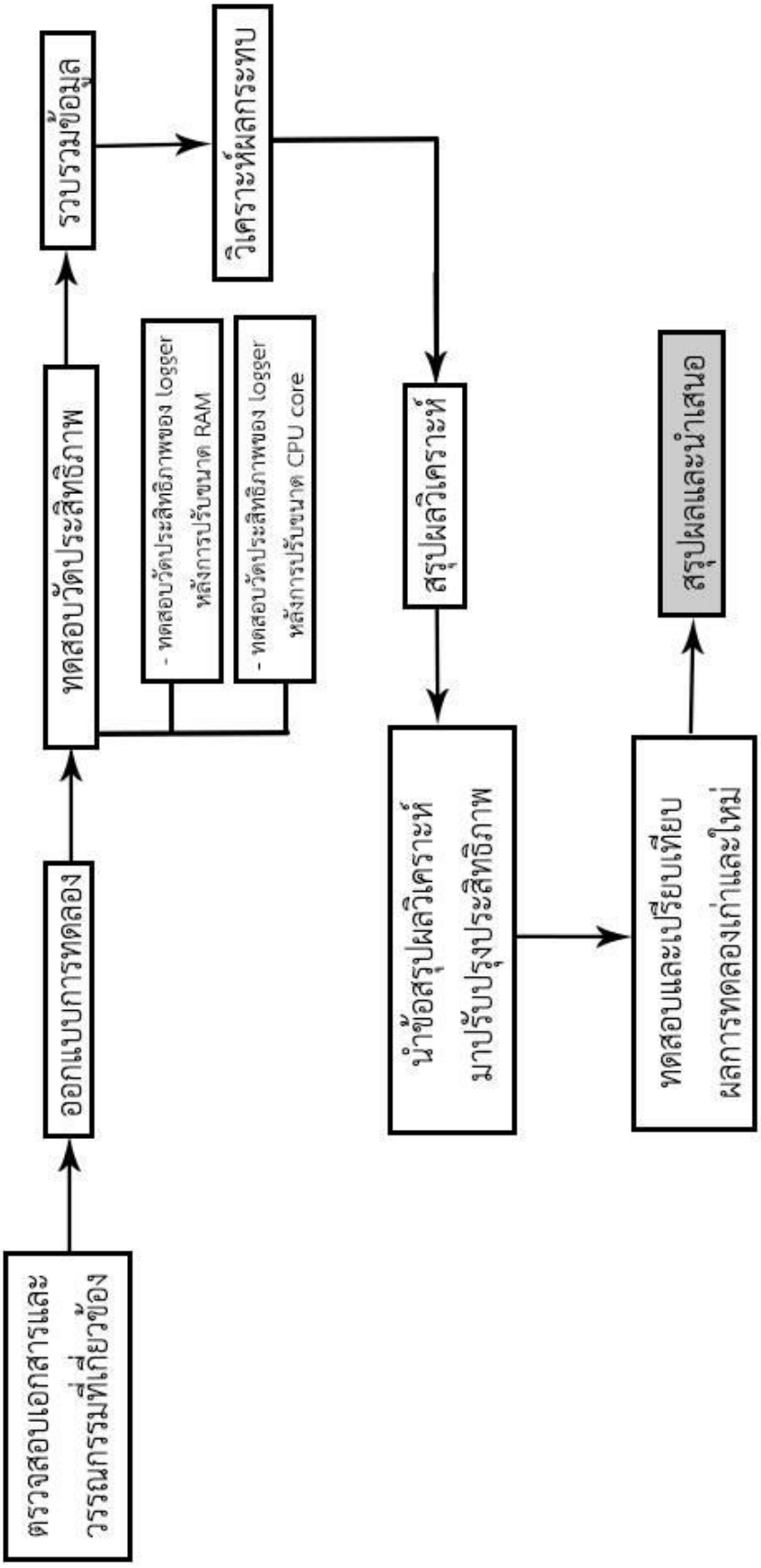
2.1) ทดสอบปรับขนาด CPU core ในส่วนของคอมพิวเตอร์ของผู้ให้บริการที่ลง logger

2.2) ทดสอบปรับขนาด CPU core ในส่วนของคอมพิวเตอร์ของผู้ใช้บริการ

**1.6.2 ส่วนของการออกแบบวิธีการเพิ่มประสิทธิภาพ**

หลังจากทำการทดลองใน 6.1.1 และได้ผลการวิเคราะห์แล้ว จะนำข้อมูลทั้งหมดที่ได้มาสรุปผล หาสาเหตุที่เกิดขึ้นจากนั้นออกแบบวิธีการปรับปรุงเพื่อเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ต่อไป

ในหัวข้อ 1.6 กรอบการดำเนินการวิจัย ผู้วิจัยได้สรุปภาพรวมเป็นภาพความสัมพันธ์ของกรอบการดำเนินการวิจัยของวิทยานิพนธ์นี้ ดังภาพที่ 1




**ภาพที่ 1 กรอบแนวคิดการวิจัย**

จากภาพที่ 1 กรอบแนวคิดการวิจัย สามารถอธิบายรายละเอียดเป็นลำดับขั้นตอนได้ ดังนี้

1) ตรวจสอบเอกสารและวรรณกรรมที่เกี่ยวข้อง โดยผู้วิจัยได้ศึกษางานวิจัยต่าง ๆ เพื่อนำไปเป็นข้อมูลสำหรับการออกแบบการทดลองเพื่อวัดและปรับปรุงประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์

2) ออกแบบการทดลอง หลังจากที่ได้ศึกษาเอกสารและวรรณกรรมที่เกี่ยวข้องแล้ว ผู้วิจัยจึงได้นำข้อมูลที่ได้ไปเป็นส่วนหนึ่งในการออกแบบการทดลองเพื่อวัดและปรับปรุงประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์

3) ทดสอบวัดประสิทธิภาพ เมื่อออกแบบการทดลองเสร็จแล้ว ผู้วิจัยจึงได้เริ่มทำการทดสอบวัดประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์ โดยสามารถแยกการทดสอบประสิทธิภาพได้เป็น 2 ส่วนหลัก คือ

3.1) ทดสอบวัดประสิทธิภาพของ logger โดยการปรับขนาด RAM ซึ่งในส่วนนี้ผู้วิจัยจะทดสอบโดยการเพิ่มและลดขนาดของ RAM ในส่วนของผู้ใช้บริการและผู้ให้บริการ เพื่อให้ทราบว่าหากมีการปรับขนาดของ RAM แล้วจะส่งผลกระทบอย่างไรต่อ logger ในระบบบันทึกเหตุการณ์

3.2) ทดสอบวัดประสิทธิภาพของ logger โดยการปรับขนาด CPU core ซึ่งในส่วนนี้ผู้วิจัยจะทดสอบโดยการเพิ่มและลดขนาดของ CPU core ในส่วนของผู้ใช้บริการและผู้ให้บริการเพื่อให้ทราบว่าหากมีการปรับขนาดของ CPU core แล้วจะส่งผลกระทบอย่างไรต่อ logger ในระบบบันทึกเหตุการณ์

4) รวบรวมข้อมูล หลังจากที่ได้ทำการทดสอบวัดประสิทธิภาพในรูปแบบต่างๆ ที่กำหนดไว้ดังข้อที่ 3. เรื่อง ทดสอบประสิทธิภาพแล้ว ผู้วิจัยจะนำข้อมูลที่ได้จากการทดสอบไปรวบรวมเป็นข้อมูลพื้นฐานเกี่ยวกับผลกระทบต่าง ๆ ที่เกิดขึ้น เพื่อนำไปวิเคราะห์เกี่ยวกับผลกระทบที่เกิดขึ้นและออกแบบการปรับปรุงประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์ต่อไป

5) วิเคราะห์ผลกระทบหลังจากรวบรวมข้อมูลของผลกระทบที่เกิดขึ้นแล้ว ผู้วิจัยนำข้อมูลเหล่านั้นไปวิเคราะห์หาสาเหตุต่าง ๆ เกี่ยวกับผลกระทบที่เกิดขึ้นเพื่อหาแนวทางในการเพิ่มประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์

6) สรุปผลการวิเคราะห์ หลังจากที่ทราบสาเหตุต่าง ๆ เกี่ยวกับผลกระทบที่เกิดขึ้นจากการวิเคราะห์แล้ว ขั้นตอนถัดไป คือ นำสาเหตุต่าง ๆ ที่เกี่ยวกับผลกระทบที่เกิดขึ้นไปสรุปผลเพื่อเป็นแนวทางในการออกแบบการทดลองสำหรับการปรับปรุงประสิทธิภาพ

7) นำข้อสรุปผลวิเคราะห์มาปรับปรุงประสิทธิภาพ หลังจากได้ข้อสรุปจากผลการวิเคราะห์และได้ออกแบบวิธีการปรับปรุงประสิทธิภาพแล้ว ขั้นตอนต่อไป คือ นำวิธีการปรับปรุงประสิทธิภาพที่ได้ออกแบบไปใช้ทดลองปรับปรุงประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์

8) ทดสอบและเปรียบเทียบผลการทดลองของระบบก่อนและหลังการปรับปรุง เมื่อปรับปรุงประสิทธิภาพของ logger บนระบบบันทึกเหตุการณ์แล้ว ผู้วิจัยจะนำผลการปรับปรุงประสิทธิภาพของระบบก่อนการปรับปรุงและหลังการปรับปรุงมาเปรียบเทียบและสรุปผลการทดลอง

9) สรุปผลและนำเสนอ หลังจากได้ผลเปรียบเทียบของระบบก่อนการปรับปรุงและหลังการปรับปรุงแล้ว ผู้วิจัยจะสรุปผลการปรับปรุงที่ได้และนำเสนอผลการเปลี่ยนแปลงที่เกิดขึ้น

โดยรวมของบทที่ 1 บทนำได้กล่าวถึงความเป็นมาของคลาวด์ ปัญหาด้านความปลอดภัยของคลาวด์ การรักษาความปลอดภัยของคลาวด์ ระบบบันทึกเหตุการณ์ซึ่งเป็นเครื่องมือที่สามารถลดปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ การทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ซึ่งเป็นวัตถุประสงค์หลักของงานวิทยานิพนธ์เล่มนี้อีกด้วย รวมถึงการตั้งสมมติฐานที่เกี่ยวข้องกับทฤษฎีของระบบปฏิบัติการ ซึ่งอาจส่งผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ รวมไปถึงวิธีการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ โดยข้อมูลข้างต้นที่กล่าวมา ผู้วิจัยได้เขียนอธิบายเพิ่มเติมในบทที่ 2 ทฤษฎีที่เกี่ยวข้อง

## 1.7 ประโยชน์ที่คาดว่าจะได้รับ

จากที่กล่าวไว้ในหัวข้อที่ 1.2 วัตถุประสงค์ของการศึกษา ได้กล่าวว่าในงานทดลองของวิทยานิพนธ์เล่มนี้จะวิเคราะห์ผลกระทบของฮาร์ดแวร์ที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์จากการวัดประสิทธิภาพและนำเสนอวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ ดังนั้นประโยชน์ของงานวิจัยนี้จึงมีความสำคัญต่อการนำไปพิจารณาเพื่อทำให้ระบบบันทึกเหตุการณ์สามารถนำไปใช้งานได้จริงต่อไปโดยมีรายละเอียด ดังนี้

1.7.1 สามารถทราบเกี่ยวกับผลกระทบของ RAM และ CPU core ที่มีต่อระบบบันทึกเหตุการณ์ จากการทดสอบประสิทธิภาพของ logger ระบบบันทึกเหตุการณ์

1.7.2 ทราบวิธีการพัฒนาประสิทธิภาพของระบบบันทึกเหตุการณ์ให้มีประสิทธิภาพที่ดียิ่งขึ้น

1.7.3 เป็นการนำเสนอแนวทางและวิธีการปรับปรุงประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ต่อไป

# บทที่ 2

# เอกสารและงานวิจัยที่เกี่ยวข้อง

ระบบบันทึกเหตุการณ์หรือ logging system เป็นวิธีการที่สำคัญที่ช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ ซึ่งเป็นวิธีการสำหรับเก็บข้อมูลหลักฐานเพื่อใช้เป็นส่วนหนึ่งในการช่วยหาตัวบุคคลที่กระทำผิดมารับผิดชอบต่อการกระทำที่เกิดขึ้น (Accountability) โดยวิธีการเก็บหลักฐานเพื่อช่วยหาตัวบุคคลกระทำผิดมารับผิดชอบต่อการกระทำที่เกิดขึ้นถือว่าเป็นหนึ่งในวิธีการตรวจสอบ (Detection) ซึ่งเป็นวิธีการที่สามารถนำไปสนับสนุนการทำงานของวิธีการป้องกันไม่ให้เกิดเหตุการณ์ขึ้น (preventive) ได้ แต่อย่างไรก็ตามการพัฒนาประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ถือว่าเป็นเรื่องสำคัญ เพราะการทำงานของระบบบันทึกเหตุการณ์ยังมีข้อบกพร่องที่ทำให้ไม่สามารถบันทึกข้อมูลเหตุการณ์ที่เกิดขึ้นได้ทั้งหมด ดังนั้นในงานวิทยานิพนธ์เล่มนี้จึงได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องต่าง ๆ เพื่อหาวิธีการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้มีประสิทธิภาพที่เพิ่มมากขึ้น โดยเอกสารและงานวิจัยที่เกี่ยวข้องที่ผู้วิจัยได้ศึกษา และสรุปเป็นหัวข้อต่าง ๆ ได้ดังนี้

2.1 ความเป็นมาและนิยามของคลาวด์

2.2 ประเภทของคลาวด์

2.3 รูปแบบการให้บริการแบบ Infrastructure as a Service (IaaS) แบบสาธารณะ

2.4 สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ

2.5 ปัญหาด้านความปลอดภัยของคลาวด์

2.6 ล็อกไฟล์ (log files)

2.7 ระบบบันทึกเหตุการณ์ (logging system)

2.8 การทดสอบประสิทธิภาพ (performance)

2.9 ระบบปฏิบัติการ (Operating System)

2.10 สรุปภาพรวมบทที่ 2

จากหัวข้อทั้งหมดที่ผู้วิจัยได้ทำการศึกษาและสรุปมีจุดประสงค์เพื่อให้ผู้อ่านสามารถเข้าใจลักษณะการทำงานของระบบบันทึกเหตุการณ์และเข้าใจเกี่ยวกับวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ที่ใช้ในงานวิทยานิพนธ์เล่มนี้

โดยการศึกษางานวิจัยในครั้งนี้ ผู้วิจัยได้นำเสนอแนวความคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง ซึ่งสามารถระบุรายละเอียดตามหัวข้อ โดยมีรายละเอียดดังนี้

## 2.1 ความเป็นมาและนิยามของคลาวด์

ในหัวข้อนี้ทางผู้วิจัยจะอธิบายเกี่ยวกับ นิยามของคลาวด์ ประเภทของคลาวด์ และรูปแบบการให้บริการของคลาวด์ เพื่อให้ผู้อ่านได้เข้าใจและเห็นภาพรวมทั้งหมดของคลาวด์ เพื่อเป็นข้อมูลพื้นฐานในการทำความเข้าใจเกี่ยวกับคลาวด์ที่ผู้วิจัยได้ใช้ในการทดลอง

คำว่าคลาวด์หรือคลาวด์คอมพิวติ้ง (cloud or cloud computing) นั้นในพจนานุกรมราชบัณฑิตยสถานไม่ได้กำหนดคำแปลในภาษาไทยไว้ แต่ในงานวิจัยต่าง ๆ ได้มีการใช้คำว่า “การประมวลผลแบบกลุ่มเมฆ” เป็นคำในภาษาไทยที่ใช้เรียกแทนคำว่า cloud หรือ cloud computing หรือ คลาวด์ (ศรีสมรักษ์ อินทุจันทร์ยง, 2553) นอกจากนี้สถาบันแห่งชาติเพื่อมาตรฐานและเทคโนโลยีของสหรัฐอเมริกา (The National Institute of Standards and Technology : NIST) ซึ่งเป็นสถาบันที่ทำหน้าที่ในการกำหนดมาตรฐานและนิยามข้อกำหนดต่าง ๆ ให้กับหน่วยงานทั้งภาครัฐและเอกชนในประเทศสหรัฐอเมริกา ได้ให้นิยามคำว่าคลาวด์ไว้ว่า “คลาวด์หรือการประมวลผลแบบกลุ่มเมฆคือ รูปแบบการใช้งานทรัพยากรคอมพิวเตอร์ (เช่น เน็ตเวิร์ก เครื่องเซิร์ฟเวอร์ แหล่งเก็บข้อมูล แอปพลิเคชันและบริการอื่น ๆ) โดยที่ทรัพยากรดังกล่าวได้ถูกกำหนดหรือจัดเตรียมไว้ล่วงหน้าแล้วจากผู้ให้บริการ นอกจากนี้การใช้งานทรัพยากรดังกล่าว ผู้ใช้สามารถใช้งานผ่านเน็ตเวิร์ก อินทราเน็ต (intranet) หรืออินเตอร์เน็ต (internet) โดยการใช้ทรัพยากรดังกล่าวสามารถเข้าถึงจากที่ใดก็ได้ผ่านเน็ตเวิร์ก การเข้าถึงต้องสะดวก และทรัพยากรดังกล่าวยังต้องสอดคล้องตามความต้องการที่เปลี่ยนไปของผู้ใช้ทั้งนี้ผู้ใช้สามารถใช้ทรัพยากรคอมพิวเตอร์ดังกล่าวได้อย่างรวดเร็วและเสียเวลาน้อยที่สุดเมื่อขอใช้หรือยกเลิกการใช้ทรัพยากรทรัพยากรดังกล่าว โดยต้องมีการติดต่อกับผู้ให้บริการน้อยที่สุดด้วย”

NIST ได้จัดประเภทของคลาวด์ไว้ 2 ประเภทคือ 1) แบ่งตามลักษณะการให้บริการและ 2) แบ่งตามขอบเขตการจัดการ ซึ่งจะถูกอธิบายเพิ่มเติมในหัวข้อถัดไปตามลำดับ (Mell & Grance, 2011; National Institute of Standards and Technology, 2014)

ในหัวข้อที่ 2.1 เรื่อง ความเป็นมาและนิยามของคลาวด์ได้อธิบายเกี่ยวกับองค์กรที่ทำหน้าที่เกี่ยวกับการกำหนดมาตรฐานและนิยามของคลาวด์ซึ่งจะทำให้ผู้อ่านได้เข้าใจถึงนิยามของคลาวด์เพิ่มมากขึ้น นอกจากนี้องค์กรดังกล่าวยังได้ระบุเกี่ยวกับลักษณะสำคัญของคลาวด์ ประเภทของคลาวด์ รวมถึงรูปแบบการให้บริการของคลาวด์โดยแต่ละหัวข้อข้างต้นจะถูกอธิบายเพิ่มเติมในหัวข้อถัดไปตามลำดับ เริ่มจากหัวข้อถัดไป คือ หัวข้อ 2.2 จะอธิบายประเภทของคลาวด์

## 2.2 ประเภทของคลาวด์

จากข้อสรุปในหัวข้อ 2.1 ความเป็นมาและนิยามของคลาวด์ ได้กล่าวว่าจะมีการอธิบายเพิ่มเติมเกี่ยวกับประเภทของคลาวด์ ภายในหัวข้อนี้จะอธิบายเพิ่มเติมเกี่ยวกับประเภทของคลาวด์ดังกล่าวที่ทางสถาบันแห่งชาติเพื่อมาตรฐานและเทคโนโลยีของสหรัฐหรือ NIST ได้กำหนด โดยจะแบ่งเป็น 2 หัวข้อ คือ 2.2.1 แบ่งตามขอบเขตการจัดการและ 2.2.2 แบ่งตามลักษณะการให้บริการนอกจากนี้ในหัวข้อนี้ยังอธิบายเกี่ยวกับเหตุผลของผู้วิจัยในการเลือกประเภทของคลาวด์เพื่อมาใช้ในการทดลอง โดยมีรายละเอียดดังนี้

### 2.2.1 แบ่งตามขอบเขตการจัดการ

จากนิยามของสถาบันแห่งชาติเพื่อมาตรฐานและเทคโนโลยีของสหรัฐอเมริกา (NIST) ได้ระบุว่าประเภทของคลลาวด์ที่แบ่งตามขอบเขตการจัดการมี 4 ประเภท คือ 1. private cloud 2. public cloud 3. hybrid cloud และ 4. community cloud โดยมีรายละเอียดดังนี้

**1) การประมวลผลแบบกลุ่มเมฆภายในองค์กรหรือการให้บริการคลาวด์ภายในองค์กร (private cloud)**

หัวข้อนี้อธิบายการประมวลผลแบบกลุ่มเมฆภายในองค์กรหรือ private cloud ซคือ cloud infrastructure ที่ถูกสร้างให้ใช้ภายในองค์กรใดองค์กรหนึ่งเท่านั้นและให้บริการเฉพาะผู้ใช้ภายในองค์กรนี้เท่านั้น โดยผู้ใช้อาจมาจากหลายๆ หน่วยงานในองค์กรดังกล่าวนี้ เช่น หน่วยบัญชี หน่วยควบคุมการผลิต private cloud อาจถูกเป็นเจ้าของ ถูกจัดการและถูกดำเนินงานได้ด้วยองค์กรดังกล่าวนี้ หรือโดย องค์กรที่เป็นกลาง หรือ รวมทั้งสององค์กรก็ได้ โดย private cloud สร้างขึ้นภายในหรือภายนอกขอบเขต ขององค์กรที่เป็นเจ้าของการประมวลผลแบบกลุ่มเมฆนี้ก็ได้ (Mell & Grance, 2011)

หมายเหตุ ผู้เขียนขอให้ความหมายของคำว่าขอบเขตขององค์กรว่าพื้นที่ (รวมถึงที่สิ่งอยู่ในพื้นที่) ภายใต้การควบคุมขององค์กรใดองค์กรหนึ่งไม่ว่าจะอยู่ภายในหรือภายนอกองค์กรก็ตาม หรือสามารถอธิบาย private cloud ได้ดังนี้ datacentersภายในขององค์กรหรือบริษัททางธุรกิจ ซึ่ง datacenters ดังกล่าวจะไม่สามารถเข้าโดยผู้ใช้ภายนอกองค์กรได้ (Armbrust, et al., 2010)

**2) การประมวลผลแบบกลุ่มเมฆสำหรับผู้ใช้เฉพาะกลุ่ม (community cloud)**

การประมวลผลแบบกลุ่มเมฆสำหรับผู้ใช้เฉพาะกลุ่ม หรือ community cloud คือ cloud infrastructure ที่สร้างขึ้นสำหรับผู้ใช้เฉพาะกลุ่มเท่านั้นอาจสร้างโดยความร่วมมือกันหลายองค์กรที่มีความ ต้องการร่วมกันในด้านต่าง ๆ เช่น ด้านภารกิจขององค์กรทั้งหลายดังกล่าว

ด้านความต้องการทางความปลอดภัย ด้านนโยบาย การประมวลผลแบบกลุ่มเมฆชนิดนี้อาจถูกเป็นเจ้าของ ถูกจัดการ ถูกดำเนินงานโดยองค์กรตั้งแต่หนึ่งองค์กรขึ้นไปที่อยู่ในกลุ่มเดียวกัน หรือโดยองค์กรที่เป็นกลาง หรือโดยผสมผสานองค์กรทั้ง สองแบบดังกล่าว ซึ่ง community cloud สร้างขึ้นภายในหรือภายนอก ขอบเขตขององค์กรที่เป็นเจ้าของการ ประมวลผลแบบกลุ่มเมฆนี้ก็ได้ (Mell & Grance, 2011)

**3) การประมวลผลแบบกลุ่มเมฆสาธารณะหรือการให้บริการแบบสาธารณะ (public cloud)**

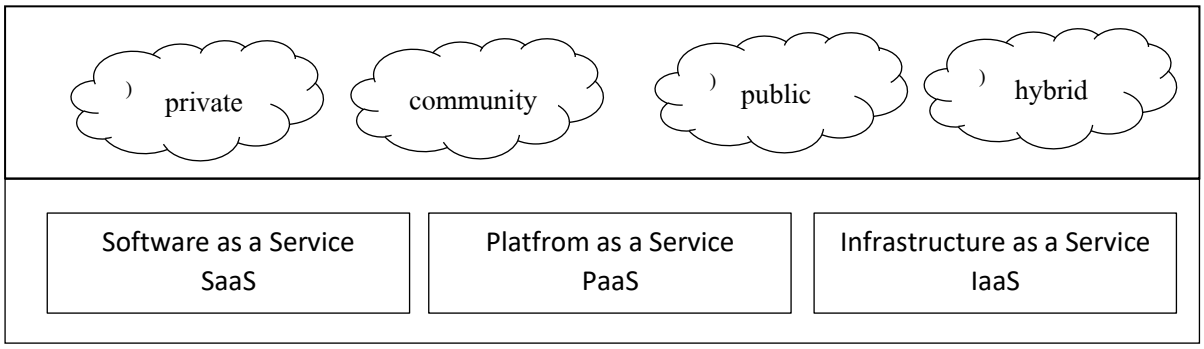
การประมวลผลแบบกลุ่มเมฆสาธารณะ หรือ public cloud คือ cloud infrastructure ที่สร้างโดยองค์กรที่มีจุดประสงค์ที่ชัดเจนว่าเพื่อให้บริการแก่ผู้ใช้ภายในนอกองค์กร public cloud ถูกเป็นเจ้าของ ถูกจัดการ และถูกดำเนินงานได้ด้วยองค์กรทางธุรกิจ องค์กรทางการศึกษาองค์กรของรัฐ หรือผสมทั้งสามองค์กร ดังกล่าวร่วมกันสร้าง public cloud ก็ได้ public cloud เกิดขึ้นใต้ขอบเขตขององค์กรที่สร้างการประมวลผล แบบกลุ่มเมฆนี้ขึ้นมาเท่านั้น (Mell & Grance, 2011) หรือ สามารถอธิบาย public cloud ได้ดังนี้ เมื่อเราเรียก datacenter hardware และ datacenter software ว่าการประมวลผลแบบกลุ่มเมฆและเมื่อไรก็ตามที่องค์กรดังกล่าวมีการให้บริการการประมวลผลแบบกลุ่มเมฆดังกล่าวในลักษณะการเก็บเงินค่าใช้บริการแบบจ่ายตามใช้งานจริง และไม่มีสัญญาผูกมัดรายเดือนหรือรายปี จะเรียกการประมวลผลแบบกลุ่มเมฆนี้ว่า public cloud (Armbrust, et al., 2010)

**4) การประมวลผลแบบกลุ่มเมฆแบบผสมผสาน (hybrid cloud)**

การประมวลผลแบบกลุ่มเมฆแบบผสมผสาน หรือ hybrid cloud คือ cloud infrastructure ที่ประกอบขึ้นด้วยการประมวลผลแบบกลุ่มเมฆที่กล่าวข้างต้น คือ (private cloud, community cloud และ public cloud) ตั้งแต่สองประเภทขึ้นไปแต่มีการผสมผสานกันของการประมวลผลแบบกลุ่มเมฆทั้งหมดด้วย เทคโนโลยีที่สร้างขึ้นเฉพาะและเป็นมาตรฐานเดียวกันสำหรับการประมวลผลแบบกลุ่มเมฆทั้งหมดที่นำผสมผสานกัน ซึ่งทำให้ทั้งข้อมูลและแอปพลิเคชันมีการย้ายไปมาระหว่างการประมวลผลแบบกลุ่มเมฆทั้งหมดได้การประมวลผลแบบกลุ่มเมฆใหม่ที่เกิดจากการประมวลผลแบบกลุ่มเมฆที่ผสมผสานกันนี้ทำให้เกิด hybrid cloud ซึ่งสร้างขึ้นเพื่อวัตถุประสงค์บางอย่างของผู้สร้าง เช่น เพื่อแบ่งภาระงานให้เหมาะสมกันระหว่าง private cloud และ public cloud (Mell & Grance, 2011) หรือสามารถอธิบาย hybrid cloud ได้ดังนี้ทั้งนี้เนื่องจาก public cloud นั้นมีประโยชน์มากทั้งด้านลดค่าใช้จ่าย สะดวกและ มีทรัพยากรไม่จำกัด ดังนั้นเมื่อ private

cloud ต้องการสร้างประสิทธิภาพให้กับตนเองให้มากขึ้น private cloud นี้สามารถไปใช้บริการจาก public cloud อีกต่อหนึ่งเมื่อ private cloud ไปใช้ public cloud ถือว่าเกิด hybrid cloud เช่น hybrid cloud เกิดขึ้นเมื่อ public cloud ถูก datacenter ใน private cloud ดึงไปใช้ทำงานหนักๆชั่วคราว เนื่องจากการทำงานดังกล่าว private cloud เองไม่สามารถทำได้ในขณะนั้นอาจเป็นเพราะด้วยมีงานอื่นที่ private cloud ดังกล่าวนี้กำลังทำอยู่จึงมีกำลังไม่พอไปทำงานใหม่ที่กำลังเข้ามาจึงต้องดึง public cloud มาใช้งานชั่วคราว (Armbrust, et al., 2010) เมื่อใช้เสร็จสามารถยกเลิกการใช้ public cloud นี้ได้ทันทีถ้าต้องการ

ในหัวข้อ 2.2.1 ที่ผ่านมาได้อธิบายเกี่ยวกับประเภทของคลาวด์ที่แบ่งตามขอบเขตการจัดการด้วยกัน 4 ประเภท คือ 1. private cloud 2. community cloud 3. public cloud และ 4. hybrid cloud โดยแสดงได้ดังภาพที่ 2 ในส่วนของภาพก้อนเมฆ จากประเภทของคลาวด์ที่แบ่งตามขอบเขตการจัดการทั้ง 4 ประเภท ในวิทยานิพนธ์เล่มนี้ผู้วิจัยสนใจที่จะบรรเทาปัญหาที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ในประเภทที่ 3 การประมวลผลกลุ่มเมฆแบบสาธารณะหรือการให้บริการแบบสาธารณะ (public cloud) สาเหตุที่เลือกการให้บริการดังกล่าวเนื่องจากว่าการให้บริการแบบสาธารณะ ทุก ๆคนมีสิทธิเข้าถึงในการใช้งาน จึงทำให้มีความหลากหลายของบุคคลที่เข้ามาใช้งาน ผู้วิจัยจึงคิดว่าด้วยสิทธิการเข้าถึงดังกล่าวอาจส่งผลให้การบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ในประเภทนี้มีความยากกว่าการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ในประเภทอื่น ๆ อีก 3 ประเภท นอกจากนี้เมื่อมีการบรรเทาปัจจัยเสี่ยงให้กับการบริการสาธารณะแล้ว จะสามารถนำวิธีการบรรเทาปัจจัยเสี่ยงดังกล่าวไปปรับใช้กับประเภทของคลาวด์ประเภทอื่น ๆ ได้ โดยรูปแบบการให้บริการคลาวด์สาธารณะจะถูกอธิบายรายละเอียดภายในหัวข้อ 2.2.2 เรื่อง แบ่งตามลักษณะการให้บริการ ในหัวข้อถัดไป




**ภาพที่ 2 ประเภทของคลาวด์และรูปแบบการให้บริการ**

จากภาพที่ 2 ได้แสดงประเภทของการให้บริการคลาวด์แบ่งตามขอบเขตการจัดการ 4 ประเภท ภายในภาพก้อนเมฆ คือ private community public และ hybrid โดยรายละเอียดเกี่ยวกับประเภทการให้บริการคลาวด์ที่แบ่งตามขอบเขตการจัดการได้ถูกกล่าวในหัวข้อที่ 2.2.1 แบ่งตามขอบเขตการจัดการในหัวข้อที่ผ่านมาแล้ว และส่วนของกรอบสี่เหลี่ยมด้านล่างจะเป็นรูปแบบการให้บริการของคลาวด์แบ่งตามลักษณะการให้บริการ 3 ประเภท คือ SaaS (Software as a Service) PaaS (Platform as a Service) และ IaaS (Infrastructure as a Service) โดยรูปแบบการให้บริการคลาวด์ที่แบ่งตามลักษณะการให้บริการ 3 ประเภทจะอธิบายเพิ่มเติมในหัวข้อ 2.2.1 แบ่งตามลักษณะการให้บริการในหัวข้อถัดไป

**2.2.2 แบ่งตามลักษณะการให้บริการ**

ในหัวข้อนี้จะอธิบายเพิ่มเติมเกี่ยวกับประเภทของการประมวลผลแบบกลุ่มเมฆหรือคลาวด์โดยแบ่งตามลักษณะการให้บริการ ซึ่งแบ่งโดยองค์กร NIST ซึ่งสามารถแบ่งออกได้ 3 ประเภท คือ SaaS (Software as a Service) PaaS (Platform as a Service) และ IaaS (Infrastructure as a Service) (Mell & Grance, 2011) ก่อนทำความเข้าใจประเภทของคลาวด์ทั้งสามประเภท ผู้อ่านควรที่จะเห็นภาพโครงสร้างพื้นฐานของคลาวด์รวมถึงต้องเข้าใจสภาพแวดล้อมพื้นฐานของคลาวด์ เพราะคลาวด์ทั้งสามประเภทจะมีความสัมพันธ์กันเนื่องจาก คลาวด์ IaaS เป็นพื้นฐานในการสร้าง คลาวด์ PaaS และคลาวด์ SaaS ตามคำกล่าวของ CSA และ W. Dawoud (CSA, 2011; Dawoud, et al., 2010) ดังนั้นถ้าเข้าใจโครงสร้างพื้นฐานของคลาวด์และสภาพแวดล้อมพื้นฐานของคลาวด์แล้ว จะทำให้ผู้อ่านเข้าใจคลาวด์ทั้งสามประเภทได้อย่างต่อเนื่องและง่ายขึ้น ดังนั้นในหัวข้อนี้จะอธิบายเกี่ยวกับคลาวด์ทั้งสามประเภท โดยจะเน้นไปที่คลาวด์ประเภท IaaS ซึ่งเป็นประเภทที่ผู้วิจัยได้เลือกใช้ในการทดลองโดยเหตุผลที่ผู้วิจัยได้เลือกใช้คลาวด์ประเภทดังกล่าวจะอธิบายเพิ่มเติมในบทสรุปของหัวข้อนี้ โดยมีรายละเอียด ดังนี้

**1) สภาพแวดล้อมพื้นฐานของคลาวด์**

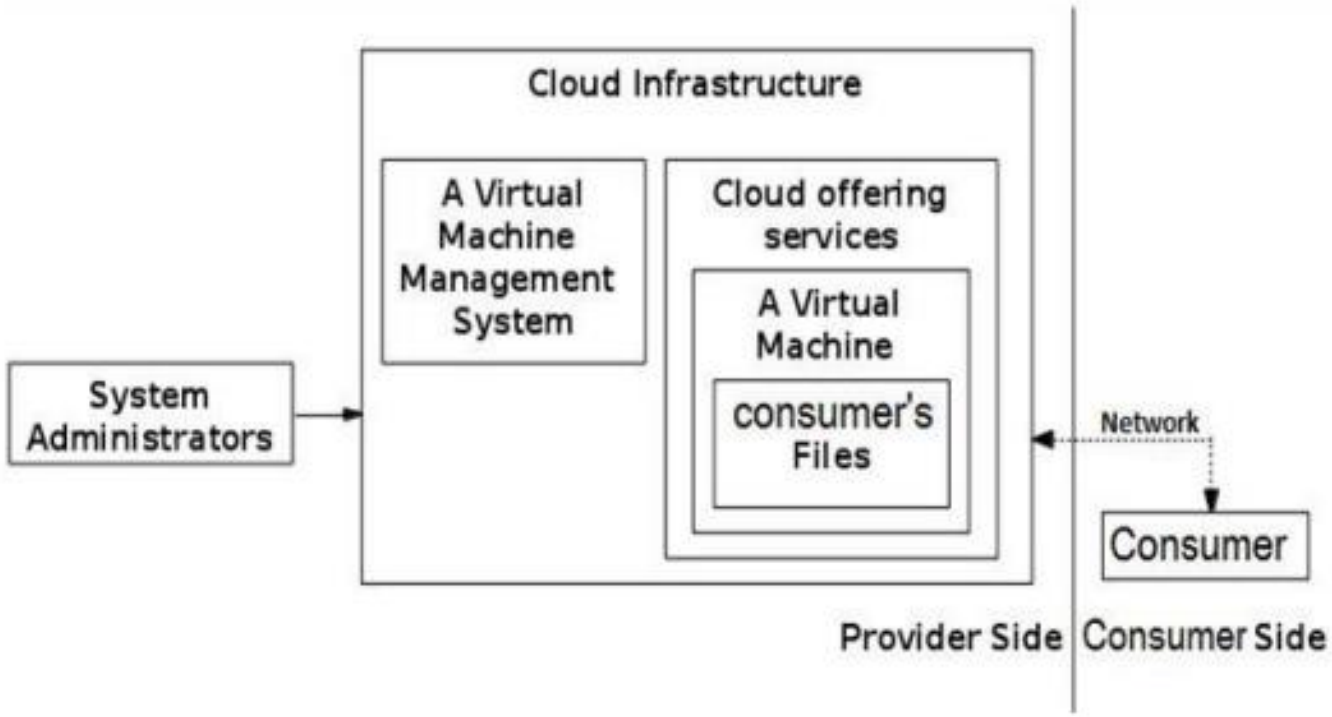
เพื่อความเข้าใจการปฏิสัมพันธ์ระหว่างบริการการประมวลผลแบบกลุ่มเมฆหรือคลาวด์กับมนุษย์รวมถึงผู้อ่านเพื่อให้เข้าใจถึงกลไกข้างในของคลาวด์เชิงลึกดังนั้นในหัวข้อนี้ผู้วิจัยจะบรรยายถึงสภาพแวดล้อมพื้นฐานของคลาวด์ (colud computing environment) ซึ่งรายละเอียดเพิ่มเติมแสดงในภาพที่ 3 สภาพแวดล้อมของคลาวด์ (Wongthai, 2014) โดยคลาวด์ประกอบไปด้วย 2 ส่วนหลักที่เกี่ยวข้องกัน คือ ส่วนผู้ให้บริการ (provider side) และส่วนผู้ใช้บริการ (consumer side) โดยมีรายละเอียดดังนี้

**1.1) ส่วนผู้ให้บริการ หรือ provider side**

ในส่วนผู้ให้บริการนี้คือเจ้าของบริการคลาวด์ทั้งหมด โดย ประกอบไปด้วย 2 ส่วนย่อย คือ Cloud infrastructure และ System Administrators โดยทั้งสองส่วน มีรายละเอียดดังนี้

**1.1.1) Cloud infrastructure**

Cloud infrastructure คือ โครงสร้างพื้นฐานของคลาวด์ โดยผู้อ่านสามารถดูได้จากด้านซ้ายมือของภาพที่ 3 ซึ่งเป็นส่วนผู้ให้บริการ ดูตรงกล่องสี่เหลี่ยมใหญ่ที่เขียนว่า Cloud Infrastructure หรือโครงสร้างพื้นฐานของคลาวด์ ซึ่งผู้ให้บริการเป็นเจ้าของ โครงสร้างพื้นฐานนี้โดยสิ้นเชิง โครงสร้างพื้นฐานของคลาวด์ ประกอบด้วย 2 ส่วนย่อยดังนี้




**ภาพที่ 3 สภาพแวดล้อมพื้นฐานของคลาวด์**

**ส่วนที่ 1 A virtual machine management system**

A virtual machine management system หรือระบบบริหารจัดการคอมพิวเตอร์เสมือนสำหรับระบบบริหารจัดการคอมพิวเตอร์เสมือนนี้คือซอฟต์แวร์ที่สร้างคอมพิวเตอร์เสมือนและควบคุม คอมพิวเตอร์เสมือนที่สร้างขึ้นนี้ด้วย หรือคือ software ที่สามารถทำให้เครื่องคอมพิวเตอร์หนึ่งเครื่อง รันระบบปฏิบัติการมากกว่าหนึ่งระบบ ในเวลาเดียวกัน ตัวอย่างของซอฟต์แวร์นี้ เช่น Xen หรือ VMware

**ส่วนที่ 2 Cloud offering services**

**1) Cloud offering services**

Cloud offering services หรือเรียกว่าคลาวด์ที่ผู้ให้บริการเตรียมไว้ให้แก่ผู้ใช้บริการ เช่น เครื่องเซิร์ฟเวอร์ ระบบปฏิบัติการ หน่วยความจำ ฮาร์ดดิสก์ ระบบเครือข่าย และอื่นๆ อย่างไรก็ตามการบริการบางอย่างมีความจำเป็นที่จะต้องเก็บไฟล์ของผู้ใช้บริการ (consumer's Files) ไว้ในคอมพิวเตอร์เสมือนหรือ VM ตัวอย่างไฟล์เหล่านี้ได้แก่ไฟล์โปรแกรมไฟล์ฐานข้อมูล

**2) System Administrators**

ต่อมาจะกล่าวในส่วนผู้ให้บริการอีกครั้งโดยภายในภาพที่ 3 องค์ประกอบต่าง ๆ ที่อยู่ในกล่อง สี่เหลี่ยมของ Cloud Infrastructure ที่ได้อธิบายไว้ก่อนหน้านั้นทั้งหมดจะถูกควบคุมดูแลโดยผู้จัดการระบบหรือเรียกว่า System Administrators ซึ่งอาจเป็นพนักงานในองค์กรของผู้ให้บริการ

**1.2) ส่วนผู้ใช้บริการหรือ consumer side**

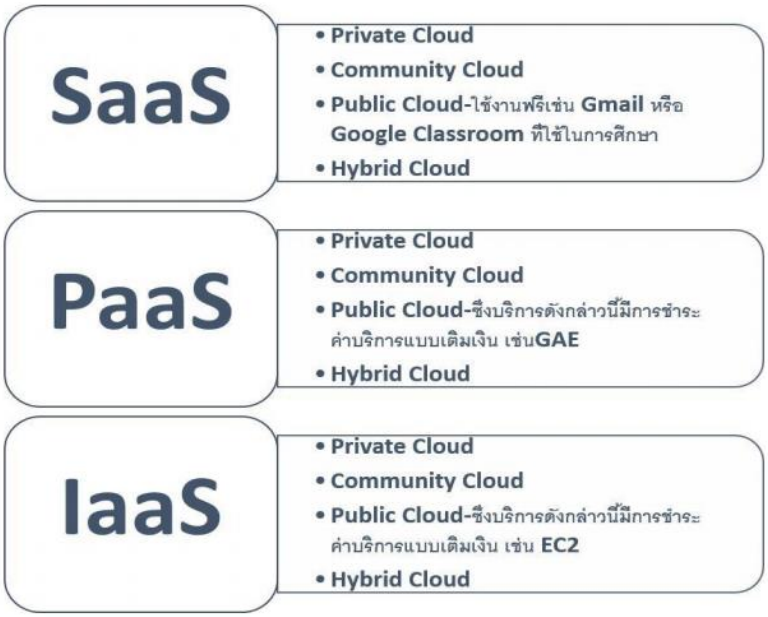
ส่วนสุดท้ายในภาพที่ 3 ด้านขวามือสุด คือส่วนผู้ใช้บริการตรงกล่องสี่เหลี่ยมที่เขียนคำว่า consumer หรือเรียกอีกอย่างว่าผู้ใช้บริการซึ่งหมายถึงผู้ที่สามารถใช้บริการการประมวลผลแบบกลุ่มเมฆหรือคลาวด์แบบต่าง ๆ ที่สร้างโดยผู้ให้บริการโดยผู้ให้บริการสามารถเข้าไปใช้งานผ่านระบบเน็ตเวิร์ก อินทราเน็ต (intranet) หรืออินเตอร์เน็ต (internet) มีทั้งแบบคิดค่าบริการและไม่คิดค่าบริการ

**2) ประเภทของคลาวด์แบ่งตามลักษณะการให้บริการ**

ในหัวข้อนี้จะอธิบายประเภทของคลาวด์แบ่งตามลักษณะการให้บริการ (cloud service delivery models) ซึ่งจากหัวข้อ 2.2.2.1 เรื่อง สภาพแวดล้อมพื้นฐานของคลาวด์ จะทำให้ผู้อ่านเข้าใจเกี่ยวกับคลาวด์ทั้งสามประเภทในหัวข้อนี้ได้ง่ายขึ้น และเนื่องจากองค์กร CSA และ W. Dawoud ได้กล่าวว่าคลาวด์ IaaS เป็นพื้นฐานในการสร้าง PaaS และ SaaS (CSA, 2011; Dawoud, et al., 2010) ในหัวข้อนี้จึงจะอธิบายเพิ่มเติมเกี่ยวกับคลาวด์ IaaS เพียงประเภทเดียว เนื่องจากคลาวด์ในประเภทดังกล่าวเป็นประเภทของคลาวด์ที่ผู้วิจัยได้เลือกใช้ในการทดลอง โดยมีรายละเอียดดังนี้

คลาวด์ประเภท IaaS นั้นเป็นพื้นฐานในการสร้างคลาวด์ประเภท PaaS และ SaaS ซึ่งผู้ให้บริการสามารถขายบริการของคลาวด์ให้กับผู้ใช้บริการ การขายบริการของคลาวด์ดังกล่าว เช่น ทรัพยากรทางคอมพิวเตอร์ พื้นที่ในการจัดเก็บข้อมูลในรูปแบบของอินทราเน็ตหรืออินเตอร์เน็ตมีทั้งคิดค่าบริการและไม่เสียค่าบริการ

ในหัวข้อที่ 2.2 เรื่อง ประเภทของคลาวด์นี้จะทำให้ผู้อื่นได้ทราบว่าคลาวด์ได้แบ่งประเภทของการให้บริการไว้ 2 ลักษณะ คือ 1. แบ่งตามขอบเขตการจัดการ โดยรายละเอียดได้กล่าวไว้ในหัวข้อที่ 2.2.1 แบ่งตามขอบเขตการจัดการและอีกลักษณะคือ 2. แบ่งตามลักษณะการให้บริการ โดยกล่าวรายละเอียดไว้ในหัวข้อ 2.2.2 แบ่งตามลักษณะการให้บริการ ซึ่งคลาวด์แบ่งตามลักษณะการให้บริการแบ่งเป็นสามรูปแบบซึ่งแบ่งโดย NIST คลาวด์สามรูปแบบดังกล่าวได้แก่ IaaS PaaS และ SaaS นอกจากนี้ประเภทของคลาวด์ที่แบ่งตามขอบเขตการจัดการหรือ cloud service deployment models ได้แบ่งเป็น 4 ประเภท คือ 1) private cloud 2) public cloud 3) hybrid cloud และ 4) community cloud ให้พิจารณาภาพที่ 4 ด้านล่าง จะเห็นว่า IaaS PaaS และ SaaS นั้นคลาวด์แต่ละประเภทนั้นสามารถให้บริการได้สี่แบบ คือ 1) private cloud 2) public cloud 3) hybrid cloud และ 4) community cloud ตัวอย่างเช่น IaaS สามารถเป็นได้ทั้ง IaaS private cloud และ IaaS public cloud และ IaaS hybrid cloud และ IaaS community cloud ตัวอย่างของ IaaS public cloud ได้แก่ EC2 (Mell & Grance, 2011)




ภาพที่ **4 ประเภททั้งหมดของการประมวลผลแบบกลุ่มเมฆ**

ภายในหัวข้อที่ 2.2 ประเภทของคลาวด์ผู้อ่านจะได้ทราบเกี่ยวกับประเภทของคลาวด์ทั้งหมด โดยจะแบ่งตามขอบเขตการจัดการ 4 ประเภทได้แก่ 1) private cloud 2) public cloud 3) hybrid cloud และ 4) community cloud และแบ่งตามลักษณะการให้บริการ 3 ประเภท ได้แก่ 1) IaaS 2) PaaS และ 3) SaaS เนื่องจากประเภทของคลาวด์นั้นมีมากมายจึงไม่สามารถทำการทดลองได้ทั้งหมด ผู้วิจัยจึงคิดว่าหากมีการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาต่อคลาวด์บนประเภทใดประเภทหนึ่งของคลาวด์แล้ว วิธีดังกล่าวจะเป็นต้นแบบในการนำวิธีการบรรเทาไปประยุกต์ใช้กับคลาวด์ประเภทอื่น ๆ ได้เช่นเดียวกัน ดังนั้นในงานวิทยานิพนธ์เล่มนี้ผู้วิจัยจึงได้เลือกที่จะบรรเทาปัญหาให้กับคลาวด์ในรูปแบบการให้บริการแบบ Infrastructure as a Serviece (IaaS) แบบสาธารณะ เหตุผลที่ได้เลือกคลาวด์ประเภทนี้ คือ คลาวด์ประเภท IaaS เป็นคลาวด์ลำดับชั้นล่างสุดสำหรับการสร้างคลาวด์ในลำดับชั้นถัดมา สังเกตได้จากภาพที่ 4 ส่วนกรอบสี่เหลี่ยมล่างสุดเขียนว่า IaaS เมื่อมีการบรรเทาปัจจัยเสี่ยงในลำดับชั้นล่างสุดแล้วก็จะทำให้การสร้างคลาวด์ในลำดับชั้นถัดมามีความปลอดภัยมากยิ่งขึ้นและอีกเหตุผลที่เลือกบรรเทาปัจจัยเสี่ยงของคลาวด์ในประเภทของคลาวด์ สาธารณะเนื่องจากว่าคลาวด์สาธารณะทุก ๆ คนมีสิทธิในการเข้าถึง ผู้วิจัยจึงคิดว่าเมื่อทุก ๆ คนมีสิทธิเข้าถึง การบรรเทาปัจจัยเสี่ยงให้กับคลาวด์ประเภทนี้จึงน่าจะยากกว่าคลาวด์ประเภทอื่น ๆ ดังนั้นในงานวิทยานิพนธ์เล่มนี้จึงได้เลือกที่จะทำการทดลองบนคลาวด์รูปแบบการให้บริการแบบ Infrastruce as a Serviece (IaaS) แบบสาธารณะ โดยรายละเอียดจะกล่าวเพิ่มเติมในหัวข้อถัดไป

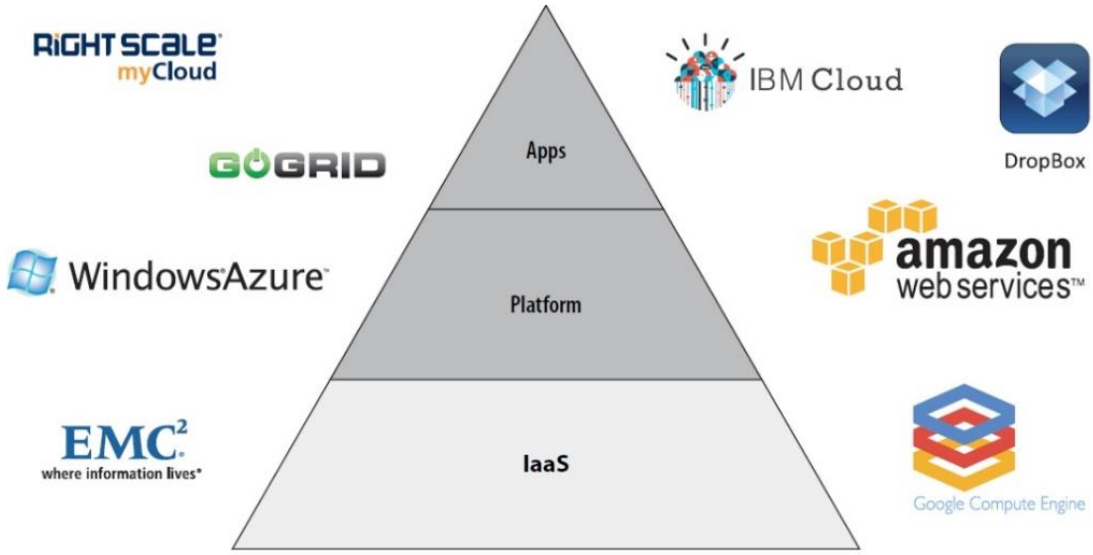
## 2.3 รูปแบบการให้บริการแบบ Infrastructure as a Serviece (IaaS) แบบสาธารณะ

จากข้อสรุปในหัวข้อ 2.2 เรื่อง ประเภทของคลาวด์ ที่ได้อธิบายเกี่ยวกับคลาวด์ประเภทต่าง ๆ รวมถึงเหตุผลของผู้วิจัยที่ได้เลือกทำการทดลองบนคลาวด์ IaaS แบบสาธารณะ ดังนั้นหัวข้อนี้จึงได้อธิบายเพิ่มเติมเกี่ยวกับรูปแบบการให้บริการของคลาวด์ IaaS แบบสาธารณะ โดยรายละเอียดเพิ่มเติมจะกล่าวในย่อหน้าถัดไป

ภายในหัวข้อนี้จะอธิบายรายละเอียดของคลาวด์ IaaS แบบสาธารณะ โดยตัวอย่างของคลาวด์ IaaS แบบสาธารณะ คือ Amazon Elastic Compute Cloud หรือ Amazon EC2 หรือเรียกสั้นๆว่า EC2 (Services, 2012) ซึ่งเป็นตัวอย่างบริการของคลาวด์ที่ผู้อ่านอาจไม่คุ้นเคยเท่ากับการให้บริการคลาวด์ประเภทบริการแหล่งเก็บข้อมูลหรือที่เรียกว่า cloud storage service เช่น Dropbox หรือ Google Drive หรือ Microsoft OneDrive เนื่องจากคลาวด์ประเภท IaaS คือการบริการทรัพยากรคอมพิวเตอร์เสมือนเน้นสำหรับการประมวลผล ซึ่งคอมพิวเตอร์เสมือนอาจประกอบด้วยพื้นที่เก็บข้อมูลและเน็ตเวิร์ก ซึ่งคลาวด์ประเภท IaaS ไม่ได้เน้นเรื่องเก็บข้อมูลแต่เน้นเรื่องให้บริการการประมวลผล

มีนักวิจัยหลายกลุ่ม เช่น Flanagan และ R. De Paris และ Amazon web Services และ The University of Manchester ได้กล่าวว่าคลาวด์ IaaS ถูกใช้อย่างกว้างขวาง โดยองค์กรต่าง ๆ เช่น ภาครัฐ ภาคการศึกษา หรือใช้ในประมวลผลในการทดลองทางการแพทย์ (Flanagan, et al., 2012; Flanagan, 2010; Manchester, 2011; R. De Paris, 2012) เช่น นักวิจัยใน (R. De Paris, 2012) ใช้ EC2 ซึ่งเป็นคอมพิวเตอร์เสมือนหลายๆ เครื่องที่เชื่อมต่อกันผ่านเน็ตเวิร์กในการประมวลผลหาสูตรยาใหม่ๆในการรักษาโรคใหม่ เนื่องจากถ้าไม่ใช้ EC2 แล้วอาจจะต้องมีการประมวลผลหาสูตรยาดังกล่าวอาจจะต้องใช้เวลาเป็นปีด้วยคอมพิวเตอร์ของนักวิจัยเองที่อยู่ในองค์กรของตนเอง แต่สามารถประมวลผลหาสูตรยาได้โดยใช้เวลาเพียงไม่เดือน ด้วย EC2 ซึ่งเป็นการลดค่าเวลาและค่าใช้จ่ายในการทำงานของนักวิจัยกลุ่มนี้อย่างมาก เมื่อใช้งานเสร็จก็ยกเลิกการใช้งานทันที

สำหรับตัวอย่างของคลาวด์ประเภท IaaS หรือ Infrastructure as a Service แบบสาธารณะ ซึ่งมักใช้โดยนักวิจัยหรือนักวิทยาศาสตร์ โดยตัวอย่างผู้ให้บริการเหล่านั้นแสดงได้ดังภาพที่ 5 ตัวอย่าง ผู้ให้บริการคลาวด์ IaaS เช่น IBM Cloud หรือ amazon web services หรือ DropBox หรือ Google Compute Engine หรือ $EMC^2$ หรือ Windows Azure หรือ RiGHTSCALe my Cloud




**ภาพที่ 5 ตัวอย่างผู้ให้บริการคลาวด์ IaaS (IaaS Architecture)**

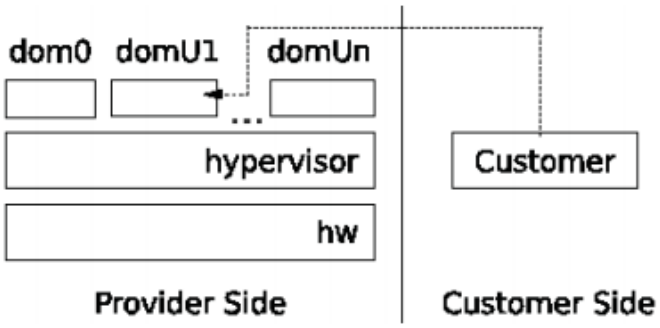
สรุปความหมายของคลาวด์ IaaS แบบสาธารณะจาก Wongthai คือ การให้บริการใช้คอมพิวเตอร์เสมือนหรือ VM แก่ผู้ใช้บริการ (Wongthai, et al., 2013a) โดยในการทดลองของวิทยานิพนธ์เล่มนี้ได้ใช้คลาวด์ IaaS แบบสาธารณะและเพื่อให้ผู้อ่านเข้าใจหลักการทำงานของคลาวด์ IaaS แบบสาธารณะมากขึ้น จึงต้องมีการอธิบายเพิ่มเติมเกี่ยวกับสถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ โดยจะกล่าวเพิ่มเติมในหัวข้อถัดไป คือหัวข้อที่ 2.4 เรื่อง สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ

## 2.4 สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ

จากหัวข้อที่ 2.2 ได้มีการอธิบายเกี่ยวกับประเภทของคลาวด์แบ่งตามลักษณะการให้บริการ 3 ประเภท คือ Software as a Service (SaaS) Platform as a Service (PaaS) และ Infrastructure as a Service (IaaS) และข้อสรุปจากหัวข้อ 2.3 ได้อธิบายว่ารูปแบบการให้บริการคลาวด์ IaaS แบบสาธารณะเป็นส่วนสำคัญของการทดลองในวิทยานิพนธ์เล่มนี้ ซึ่งในหัวข้อนี้จะมุ่งเน้นไปที่สถาปัตยกรรมของการบริการคลาวด์ IaaS แบบสารณะ เนื่องจากว่าการให้บริการคลาวด์ IaaS นี้จะเป็นส่วนที่ผู้วิจัยได้นำสถาปัตยกรรมไปใช้ออกแบบในการทดลอง จากงานของ Wongthai ได้อธิบายว่ารูปแบบการให้บริการคลาวด์ IaaS แบบสาธารณะเป็นบริการโครงสร้างพื้นฐานทางสารสนเทศ ซึ่งอยู่ในรูปแบบของคอมพิวเตอร์เสมือน บริการนี้เป็นการเตรียมเครื่องคอมพิวเตอร์

ระบบปฏิบัติการ โปรแกรมประยุกต์ ระบบอินเทอร์เน็ตเสมือน และจัดเตรียมแบนด์วิดท์ (bandwidth) ให้แก่ผู้ขอใช้บริการ เพื่อให้ผู้ใช้บริการสามารถติดตั้งซอฟต์แวร์ที่ต้องการในเครื่องคอมพิวเตอร์เสมือน (Wongthai, et al., 2013b) และยังมีงานวิจัยอีกหลายท่าน เช่น Flanagan และ R. De Paris และ Amazon web Services และ The University of Manchester ได้อธิบายเกี่ยวกับการให้บริการเช่าคอมพิวเตอร์เสมือนของคลาวด์ IaaS แบบสาธารณะว่าจะเน้นที่รูปแบบการทำงานในเรื่องของการประมวลผลข้อมูลที่ต้องการความไว เช่น การทดลองในงานด้านวิทยาศาสตร์ บริษัทที่ให้การบริการ คลาวด์ IaaS เช่น Amazon Web Service (AWS) IBM Cloud Windows Azure และอื่น ๆ (Manchester, 2011; Mell & Grance, 2011; R. De Paris, 2012; Services, 2012) โดยรายละเอียดเกี่ยวกับสถาปัตยกรรมจอคลาวด์ IaaS แบบสาธารณะจะถูกอธิบายเพิ่มเติมในหัวข้อนี้

ผู้วิจัยเห็นว่าสถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะถือว่าเป็นเรื่องสำคัญของงานวิทยานิพนธ์เล่มนี้ เพราะการศึกษาสถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะจะทำให้เห็นภาพรวมของโครงสร้าง วิธีการทำงานของคลาวด์ IaaS แบบสาธารณะ ทำให้ง่ายต่อการออกแบบการทดลอง ผู้วิจัยจึงนำสถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะมาออกแบบสำหรับการทดลอง ดังนั้นการทำความเข้าใจเกี่ยวกับสถาปัตกรรมของคลาวด์ IaaS แบบสาธารณะจึงเป็นสิ่งที่สำคัญ โดยสถาปัตยกรรมคลาวด์ของ IaaS แบบสาธารณะที่ผู้วิจัยได้ทำการศึกษาและอ้างอิงจะใช้สถาปัตยกรรมคลาวด์ IaaS ของ Wongthai (Wongthai, et al., 2013b) แสดงได้ดังภาพที่ 6 สถาปัตยกรรมของคลาวด์ IaaS (IaaS Architecture)




**ภาพที่ 6 สถาปัตยกรรมของคลาวด์ IaaS (IaaS Architecture)**

มีงานวิจัยได้นำเสนอและอธิบายเพื่อความเข้าใจเกี่ยวกับคลาวด์ IaaS แบบสาธารณะได้แสดงในภาพที่ 6 สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ (Wongthai, et al., 2013a) คำอธิบายขององค์ประกอบของคลาวด์ IaaS แบบสาธารณะบางส่วนได้ถูกอธิบายไว้แล้วในหัวข้อ 2.3 เรื่อง รูปแบบการให้บริการแบบ Infrastructure as a Serviece (IaaS) แบบสาธารณะ อย่างไรก็ตามมีบางองค์ประกอบที่ต้องอธิบายเพิ่มเติม โดยคลาวด์ IaaS แบ่งสภาพแวดล้อมการทำงานออกเป็น 2 ส่วน ได้แก่ส่วนผู้ให้บริการ (provider side) และส่วนผู้ใช้บริการ (consumer side) เช่นเดียวกับภาพที่ 3 โดยภายในหัวข้อนี้จะเน้นอธิบายรายละเอียดของผู้ให้บริการ ซึ่งหมายถึงองค์กรที่เตรียมบริการคอมพิวเตอร์เสมือนให้กับผู้ใช้บริการ ยกตัวอย่างองค์กรเช่น Amazon Google หรือ Microsoft หรือ องค์กรใด ๆ ก็ตามที่ผู้ให้บริการจะเน้นให้บริการคลาวด์ IaaS อย่างไรก็ตามเมื่อผู้ให้บริการได้สร้างคลาวด์ IaaS แบบสาธารณะแล้ว คลาวด์ IaaS แบบสาธารณะนี้สามารถใช้เป็นฐานในการ สร้าง PaaS และ SaaS ต่อไปได้

คลาวด์ IaaS แบบสาธารณะมีองค์ประกอบหลักดังตารางต่อไปนี้

| ส่วนประกอบ | คำอธิบาย |
|---|---|
| hw (ย่อมาจาก hardware) | hardware หรือฮาร์ดแวร์ คือ เครื่องคอมพิวเตอร์ที่จะใช้ในการสร้าง IaaS คลาวด์แบบสาธารณะซึ่งถูกใช้ในการทดลอง |
| hypervisor | hypervisor เช่น software หรือซอฟต์แวร์ที่สามารถทำให้เครื่องคอมพิวเตอร์หนึ่งเครื่องรันระบบปฏิบัติการ ได้มากกว่าหนึ่งระบบใน เวลาเดียวกัน (Rocha, et al., 2011) โดย software ดังกล่าวมีทั้งแบบการค้า เช่น VMware หรือแบบ open source เช่น Xen และ software เหล่านี้จะถูกติดตั้งบน hypervisor หรือ hw ซึ่งถูกติดตั้งบนเครื่องคอมพิวเตอร์ของผู้ให้บริการคลาวด์หรือ dom0 |

| | |
|---|---|
| **ส่วนประกอบ** | **คำอธิบาย** |
| domU1…..domUn (ย่อมาจาก Domain User) | หมายถึงเวอร์ชวลแมชชีนหรือคอมพิวเตอร์เสมือน ผู้ใช้บริการถือเป็นเจ้าของ domU แต่ละ domU จะทำงานอิสระไม่ขึ้นต่อกัน มีได้หลาย domU เช่น domU1 domU2 ,..., domUn |
| dom0 (ย่อมาจาก Domain Zero เป็นคำศัพท์เฉพาะที่ใช้กับ Xen) | เป็น VM ที่บริหารจัดการ domU และ hw และทำงานเมื่อ hypervisor ถูกเปิดใช้งานนั้นโดยการ (hw หรือเครื่องคอมพิวเตอร์เปิดใช้งาน) เมื่อนำมาเทียบการทำงานกับภาพที่ 3 ถือเป็นส่วนหนึ่งของ A Virtual Machine Management System และdom0 นี้สามารถ สร้าง จัดการ ลบ domU ได้ |

**ตารางที่ 1 ส่วนประกอบการทำงานในสถาปัตยกรรมคลาวด์ IaaS**

**หมายเหตุ** โครงสร้างของ IaaS ที่ได้อธิบายผู้วิจัยได้จำลองจากเครื่องคอมพิวเตอร์เครื่องเดียวที่ยังไม่ได้มีการเชื่อมต่อกับคอมพิวเตอร์เครื่องอื่น

จากหัวข้อที่ 2.4 สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะจะทำให้ผู้อ่านได้เข้าใจเกี่ยวกับหลักการทำงานของคลาวด์ IaaS แบบสาธารณะมากยิ่งขึ้นและเมื่อผู้อ่านเข้าใจหลักการทำงานเพิ่มมากขึ้นแล้วจะทำให้ผู้อ่านเข้าใจถึงวิธีการปรับปรุงประสิทธิภาพของระบบ logger ใน

ระบบบันทึกเหตุการณ์โดยจะกล่าวเพิ่มเติมในหัวข้อที่ 2.8 การทดสอบประสิทธิภาพ (performance) แต่ก่อนที่ผู้วิจัยจะอธิบายเกี่ยวกับการปรับปรุงประสิทธิภาพ ผู้วิจัยต้องการให้ผู้อ่านได้เข้าใจเกี่ยวกับปัญหาด้านความปลอดภัยของคลาวด์ก่อน โดยมีวัตถุประสงค์เพื่อให้ผู้อ่านทราบว่าปัญหาด้านความปลอดภัยของคลาวด์มีอะไรบ้างและปัญหาใดที่ผู้วิจัยสนใจจะบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ โดยปัญหาด้านความปลอดภัยจะกล่าวเพิ่มในหัวข้อที่ 2.5 ปัญหาด้านความปลอดภัยของคลาวด์ ในหัวข้อถัดไป

## 2.5 ปัญหาด้านความปลอดภัยของคลาวด์

สิ่งสำคัญนอกจากจะต้องทราบเกี่ยวกับข้อมูลพื้นฐานของคลาวด์ซึ่งถูกกล่าวไว้ในต่าง ๆ คือ หัวข้อ 2.1 เรื่อง ความเป็นมาและนิยามของคลาวด์ 2.2 ประเภทของคลาวด์ 2.3 รูปแบบการให้บริการแบบ Infrastructure as a Service (IaaS) แบบสาธารณะ และ 2.4 สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ ซึ่งหัวข้อทั้งหมดได้ถูกอธิบายไว้หัวข้อก่อนหน้านี้ โดยในหัวข้อ 2.5 นี้จะอธิบายเกี่ยวกับปัญหาด้านความปลอดภัยของคลาวด์ซึ่งถือว่าเป็นข้อมูลพื้นฐานที่จะทำให้ทราบว่าระบบบันทึกเหตุการณ์จะสามารถเป็นส่วนหนึ่งในการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ในข้อใดได้บ้าง เพราะ งานวิทยานิพนธ์นี้มุ่งเน้นการวัดและปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ให้ดีขึ้นเพื่อเป็นประโยชน์ส่วนหนึ่งในการช่วยการบรรเทาปัจจัยเสี่ยงดังกล่าว

ในปี 2016 CSA รวบรวมและได้รายงานเกี่ยวกับภัยคุกคามที่เกิดขึ้นต่อคลาวด์และได้จัดอันดับความรุนแรงไว้ในรายงานที่ชื่อว่า The Treacherous 12 Cloud Computing Top Threats in 2016 (CSA, 2016a, 2016b) ซึ่งมีรายการภัยคุกคาม คือ

1) การละเมิดข้อมูล (Data Breaches)

2) การพิสูจน์ตัวตนที่ง่ายเกินไป (Weak Identity, Credential and Access Management)

3) ช่องทางการติดต่อที่ไม่ปลอดภัย (Insecure APIs)

4) ช่องโหว่อาจเกิดจากระบบและแอปพลิเคชัน (System and Application Vulnerabilities)

5) บัญชีผู้ใช้งานถูกขโมย (Account Hijacking)

6) ภัยคุกคามที่เกิดจากบุคคลภายใน (Malicious Insiders)

7) Advanced Persistent Threats (APTs)

8) ข้อมูลสูญหายหรือรั่วไหล (Data Loss)

9) Insufficient Due Diligence

10) การนำคลาวด์ไปใช้ในทางที่ผิด (Abuse and Nefarious Use of Cloud Services)

11) การปฏิเสธการให้บริการ (Denial of Service)

12) ปัญหาเกิดจากเทคโนโลยีการแชร์ (Shared Technology Issues)

โดยในวิทยานิพนธ์เล่มนี้สนใจที่จะบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ในข้อที่ 1 Data Breaches (การละเมิดข้อมูล) ผู้วิจัยจึงอธิบายรายละเอียดภัยคุกคามดังกล่าวเพียงหัวข้อเดียวโดยมีรายละเอียด ดังนี้

**1) Data Breaches (การละเมิดข้อมูล)**

ข้อมูลที่เป็นความลับอาจจะถูกขโมยและอาจจะถูกนำไปใช้หรือบุคคลที่ไม่ได้รับอนุญาตอาจจะสามารถเข้าถึงข้อมูลได้ โดยสาเหตุมาจากช่องโหว่ที่เกิดจากข้อผิดพลาดของผู้ให้บริการหรือผู้ใช้บริการรวมถึงช่องโหว่จากการรักษาความปลอดภัยที่ไม่ดี

จากรายงานของ CSA การละเมิดข้อมูลจะมีรูปแบบการละเมิดข้อมูลด้วยกัน 4 รูปแบบ คือ 1. การเข้าไปดูไฟล์ (viewed) 2. การคัดลอกข้อมูลเพื่อวัตถุประสงค์ต่าง ๆ เช่น คัดลอกข้อมูลและนำไปเผยแพร่หรือคัดลอกข้อมูลและย้ายไปไว้ตำแหน่งอื่น ๆ (release) 3. การแก้ไขไฟล์เดิม รวมถึงการลบข้อมูลเดิมที่มีอยู่ด้วย (Stolen) 4. การนำข้อมูลที่ได้ไปใช้งาน (Used)

จากรูปแบบการละเมิดข้อมูลทั้ง 4 รูปแบบ ก่อนจะเกิดการคัดลอกข้อมูล ย้ายข้อมูล แก้ไขข้อมูลหรือแม้กระทั่งลบข้อมูล เหตุการณ์ดังกล่าวจะเกิดขึ้นได้ แฮกเกอร์ต้องเข้าไปดูไฟล์ก่อน ซึ่งการเข้าไปดูไฟล์หรือ view เป็นรูปแบบที่ 1 ของการละเมิดข้อมูลและอาจส่งผลให้เกิดการระเมิดข้อมูลในแบบอื่น ๆ ตามมาด้วยเช่นกัน ดังนั้น ในงานวิทยานิพนธ์เล่มนี้จึงเน้นวิธีการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ในรูปแบบที่ 1 การเข้าไปดูไฟล์ โดยผู้วิจัยสมมติเหตุการณ์ให้แฮกเกอร์เข้าไปอ่านไฟล์ที่สำคัญ คือไฟล์ s.txt ซึ่งเป็นไฟล์ที่ผู้วิจัยได้สร้างขึ้นและสมมติให้เป็นไฟล์ที่สำคัญ ที่แฮกเกอร์ต้องการเข้าไปอ่านไฟล์ โดยการเข้าไปอ่านไฟล์จึงถือว่าเป็นการเข้าไปดูไฟล์ซึ่งเป็นรูปแบบที่ 1 ของการละเมิดข้อมูล โดยผู้วิจัยจะเน้นเพียงรูปแบบที่ 1 การดูไฟล์เพียงอย่างเดียวเท่านั้น โดยสาเหตุที่ต้องผู้วิจัยต้องเลือกวิธีที่ 1 เพื่อเป็นแนวทางสำหรับบรรเทาปัญหาและผู้วิจัยหวังว่าจะได้ทดสอบในอนาคตต่อไป

จากปัญหาด้านความปลอดภัยของคลาวด์ทั้ง 12 ข้อที่ทาง องค์กร CSA ได้จัดทำรายงานขึ้นในปี 2016 ปัญหาดังกล่าวทั้งหมดจะเป็นผลกระทบต่อการให้บริการคลาวด์สาธารณะทั้งสิ้นและยังส่งผลกระทบต่อการให้บริการคลาวด์ IaaS แบบสาธารณะทุกข้อ โดยในข้อที่ 1 ปัญหาของการ

ละเมิดข้อมูลได้ถูกจัดให้เป็นปัญหาที่มีความรุนแรงมากที่สุด ในวิทยานิพนธ์ผู้วิจัยสนใจที่จะทำการปรับปรุงประสิทธิภาพของ logger บนระบบบันทึกเหตุการณ์ เพราะ การปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ เป็นส่วนหนึ่งในการช่วยบรรเทาปัญหาในข้อดังกล่าว โดยผู้วิจัยได้สมมติเหตุการณ์ให้มีแฮกเกอร์ได้เข้าไปครอบครองส่วนของผู้ใช้บริการจากนั้นแฮกเกอร์ได้อ่านไฟล์ของผู้ใช้บริการคลาวด์ IaaS แบบสาธารณะ ซึ่งการอ่านไฟล์เป็นการละเมิดข้อมูลรูปแบบหนึ่งของการละเมิดลิขสิทธิ์จาก 4 วิธีการที่ได้อธิบายไว้ในข้อที่ 1 Data Breaches (การละเมิดข้อมูล) โดยหลักฐานที่ได้จากล็อกไฟล์ จะเป็นหลักฐานสำคัญในการหาผู้กระทำผิดมารับผิดชอบต่อเหตุการณ์ ซึ่งรายละเอียดของล็อกไฟล์ (log files) จะอธิบายเพิ่มเติมในหัวข้อถัดไป หัวข้อ 2.8 เรื่อง ล็อกไฟล์ (log files)

### 2.5.1 วิธีการรักษาความปลอดภัยของการละเมิดข้อมูล

ในปัจจุบันได้มีเครื่องมือและวิธีการหลากหลายที่ช่วยสนับสนุนในการรักษาความปลอดภัยของคลาวด์ โดยมีนักวิจัยของได้แบ่งวิธีการรักษาความปลอดภัยไว้เป็น 2 วิธีการด้วยกัน (Ko, 2014) โดยมีรายละเอียด ดังนี้

1) วิธีการป้องกัน (preventive) เช่น ระบบไฟร์วอลล์ (firewall) การเข้ารหัสข้อมูล (encryption) โดยวิธีการนี้จะป้องกันไม่ให้เกิดเหตุการณ์ที่ส่งผลกระทบต่อความปลอดภัยต่อคลาวด์

2) การตรวจสอบ (detective) เช่น การตรวจสอบหาร่องรอยของการกระทำ (audit trails) การบันทึกเหตุการณ์ที่เกิดขึ้น (logging system) รวมถึงเครื่องมือประเภทวิเคราะห์ต่าง ๆ (analysis tools) (Ko, 2014)

ในย่อหน้าที่ผ่านมาได้อธิบายเกี่ยวกับวิธีการรักษาความปลอดภัยของคลาวด์ ซึ่งมีด้วยกัน 2 วิธี เมื่อนำวิธีการรักษาความปลอดภัยทั้ง 2 วิธีมาพิจารณา ทางผู้วิจัยเห็นว่าระบบบันทึกเหตุการณ์ (logging system) ถือว่าเป็นกระบวนรักษาความปลอดภัยในวิธีการที่สอง นั่นคือ วิธีการตรวจสอบ (detective) นอกจากนี้ในงานวิจัยของ Gartner ได้อธิบายว่าการรักษาความปลอดภัยในประเภทที่ 2 การตรวจสอบหรือ detective หากมีการการนำวิธีนี้มาใช้ประสบผลสำเร็จยังสามารถนำวิธีดังกล่าวไปสนับสนุนให้วิธีการที่ 1 หรือวิธีการป้องกันมีประสิทธิภาพมากขึ้นได้ (Gartner, 2017)

นอกจากนี้งานวิจัยของ Ko ยังกล่าวอีกว่าความปลอดภัยในคลาวด์จะส่งผลกระทบทั้ง 2 ฝ่าย ทั้งผู้ให้บริการและผู้ใช้บริการ (Ko, 2014) เนื่องจากรูปแบบการใช้งานของคลาวด์ต้องเก็บไฟล์หรือทรัพยากรต่าง ๆ ของผู้ใช้บริการ ด้วยรูปแบบดังกล่าวที่คลาวด์ได้เก็บไฟล์หรือทรัพยากรต่าง ๆ ของผู้ใช้บริการ ทำให้ผู้ใช้บริการเกิดความกังวลเกี่ยวกับความปลอดภัยว่า บุคคลใดสามารถเข้าถึงไฟล์ของตนเองได้บ้าง (Subashini & Kavitha, 2011) ระบบบันทึกเหตุการณ์จึงถือว่าเป็นวิธีการหนึ่งที่สำคัญต่อการหาผู้รับผิดชอบต่อการกระทำผิด (accountability) ผู้วิจัยเห็นว่าระบบดังกล่าวจะช่วยบรรเทาปัจจัยเสี่ยงที่ก่อให้เกิดภัยคุกคามต่อคลาวด์ในข้อที่ 1 การละเมิดข้อมูล ตามรายงานการวิจัยของ CSA ทั้ง 12 ข้อในปี 2016 โดยใช้ชื่อรายงานว่า The Treacherous 12 Cloud Computing Top Threats in 2016 (CSA, 2016a, 2016b) ในระบบบันทึกเหตุการณ์จะประกอบไปด้วยโปรเซสที่ใช้บันทึกเหตุการณ์ เรียกว่า logger (logger เป็นโปรเซสสำหรับตรวจบันทึกเหตุการณ์ต่าง ๆ ที่เกิดขึ้นภายในคลาวด์) (Wongthai & Moorsel, 2016) โดยวิธีการทำงานของ logger จะถูกอธิบายในรูปแบบของสถาปัตยกรรมภายในของระบบบันทึกเหตุการณ์ ซึ่งถูกอธิบายเพิ่มเติมไว้ในหัวข้อ 2.9 ระบบบันทึกเหตุการณ์ (logging system)

ภาพรวมทั้งหมดของหัวข้อ 2.5.1 เรื่อง วิธีการรักษาความปลอดภัย ผู้อ่านจะได้ทราบเกี่ยวกับนิยามของล็อกไฟล์ ประเภทของล็อกไฟล์ โดยในวิทยานิพนธ์เล่มนี้ผู้วิจัยมุ่งเน้นไปที่ล็อกไฟล์ประเภทที่ 2 คือ การตรวจสอบร่องรอยของการกระทำ ซึ่งวิธีการดังกล่าวทำให้ระบบบันทึกเหตุการณ์สามารถสร้างล็อกไฟล์ได้ นอกจากนี้เมื่อมีเหตุการณ์ผิดปกติเกิดขึ้นหรือเหตุการณ์ใด ๆ ก็ตามที่เกิดขึ้น ล็อกไฟล์จะจัดเก็บเหตุการณ์ต่าง ๆ ที่เกิดขึ้นเหล่านั้นเพื่อเป็นหลักฐานและหลักฐานจากล็อกไฟล์จะถูกใช้ในกระบวนการตรวจสอบร่องรอยของการกระทำซึ่งระบบบันทึกเหตุการณ์ (logging system) โดยรายละเอียดของระบบบันทึกเหตุการณ์จะถูกอธิบายในหัวข้อถัดไปหัวข้อ 2.7 เรื่อง ระบบบันทึกเหตุการณ์ (logging system) ต่อไป

### 2.5.2 ความรับผิดชอบต่อสิ่งที่เกิดขึ้น (Accountability)

จากข้อสรุปในหัวข้อ 2.5 ปัญหาด้านความปลอดภัยของคลาวด์ได้กล่าวว่าในงานวิทยานิพนธ์เล่มนี้มุ่งหวังที่จะวัดและปรับปรุงประสิทธิของระบบบันทึกเหตุการณ์เพื่อช่วยเป็นส่วนหนึ่งในการบรรเทาปัญหาในข้อที่ 1 ปัญหาของการละเมิดข้อมูลด้วยวิธีการเข้าไปอ่านไฟล์จากแฮกเกอร์ ซึ่งระบบบันทึกเหตุการณ์สามารถเป็นส่วนหนึ่งในการช่วยบรรเทาปัญหาดังกล่าวได้ด้วยวิธีการเก็บหลักฐานให้ผู้มีส่วนเกี่ยวข้องทราบว่าใครเคยเข้าถึงไฟล์อะไร เมื่อทราบว่าใครเคยเข้าถึง

ไฟล์อะไร ผู้มีส่วนเกี่ยวข้อง เช่น ผู้เป็นเจ้าของไฟล์หรือผู้ให้บริการ อาจจะดำเนินขั้นตอนต่อมาทำให้รู้ว่าใครต้องเป็นคนรับผิดชอบ ด้วยกระบวนการเก็บหลักฐานดังกล่าวจนนำไปสู่กระบวนการหาคนรับผิดชอบต่อการกระทำ เรียกว่า ความรับผิดชอบต่อสิ่งที่เกิดขึ้นหรือ accountability โดยในหัวข้อนี้จะอธิบายเพิ่มเติมเกี่ยวกับ accountability

การพิจารณาถึงความปลอดภัยต่อคลาวด์เป็นเรื่องที่สำคัญ ความรับผิดชอบต่อสิ่งที่เกิดขึ้น (accountability) มีระบบบันทึกเหตุการณ์ (กระบวนการทำงานของระบบบันทึกเหตุการณ์จะถูกกล่าวในหัวข้อ 2.7) เป็นส่วนประกอบสำคัญ โดยทั่วไปคำว่า accountability ได้ถูกนิยามจาก Macmillan's dictionary (macmillandictionary) ความหมาย คือ ต้องทราบว่าใครเป็นผู้กระทำผิดและสามารถหาบุคคลที่ต้องรับผิดชอบต่อสิ่งที่เกิดขึ้น แต่ความหมายในเชิงความปลอดภัยนั้น Papanikolaou และ Pearson ได้กล่าวว่า accountability หมายถึง การจัดการความรับผิดชอบโดยมี 3 องค์ประกอบด้วยกัน คือ 1) กลไกทางกฎหมาย 2) กลไกข้อบังคับ และ 3) กลไกทางเทคนิค (Papanikolaou & Pearson, 2012) เมื่อนำองค์ประกอบทั้ง 3 ข้อข้างต้นมาเปรียบเทียบ ผู้วิจัยเห็นว่าระบบบันทึกเหตุการณ์ถือว่าเป็นเครื่องมือในองค์ประกอบที่ 3 นั่นคือ กลไกทางเทคนิค เนื่องจากว่าระบบบันทึกเหตุการณ์เป็นเครื่องมือสำหรับเก็บข้อมูลให้ผู้เกี่ยวข้องหรือผู้มีสิทธิ เพื่อนำไปใช้วิเคราะห์เหตุการณ์ที่เกิดขึ้น โดยเหตุการณ์เหล่านั้นอาจจะเกี่ยวข้องกับความปลอดภัยของคลาวด์

ในหัวข้อนี้ สิ่งที่สำคัญที่นำไปสู่กระบวนการหาบุคคลที่กระทำผิดมารับผิดชอบต่อสิ่งที่เกิดขึ้น คือ ต้องมีหลักฐาน โดยหลักฐานดังกล่าวสามารถหาได้จากล็อกไฟล์ (log files) รายละเอียดของล็อกไฟล์จะถูกอธิบายเพิ่มเติมในหัวข้อ 2.6 ล็อกไฟล์ (log files)

## 2.6 ล็อกไฟล์ (logs files) และประเภทของล็อกไฟล์

จากข้อสรุปในข้อ 2.5.2 ความรับผิดชอบต่อสิ่งที่เกิดขึ้น จะเห็นว่าหลักฐานคือสิ่งสำคัญที่จะนำไปสู่กระบวนการหาตัวบุคคลกระทำผิดมารับผิดชอบต่อสิ่งที่เกิดขึ้น โดยหลักฐานดังกล่าวทำให้ทราบว่าบุคคลใดที่มีส่วนเกี่ยวข้องในการกระทำผิดบ้างและต้องรับผิดชอบต่อสิ่งที่เกิดขึ้นอย่างไร โดยล็อกไฟล์จะบันทึกเหตุการณ์ที่เกิดขึ้นและสามารถนำล็อกไฟล์นั้นไปเป็นหลักฐาน โดยในหัวข้อนี้จะอธิบายเพิ่มเติมเกี่ยวกับประเภทของล็อกไฟล์และล็อกไฟล์ประเภทใดที่ใช้ในงานวิทยานิพนธ์เล่มนี้

มีนักวิจัยได้อธิบายว่าระบบบันทึกเหตุการณ์เป็นวิธีการหนึ่งที่สำคัญสำหรับช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามขึ้นต่อคลาวด์ โดยภายในระบบบันทึกเหตุการณ์ได้จัดเก็บหลักฐานล็

อกไฟล์ (log files) ไว้เป็นหลักฐานและหลักฐานจากล็อกไฟล์สามารถนำไปช่วยหาบุคคลที่กระทำผิดมารับผิดชอบต่อสิ่งที่เกิดขึ้นได้ (Wongthai, et al., 2013a) โดยในงานวิจัยของ Ko กล่าวว่าการเก็บหลักฐานหรือเหตุการณ์ที่เกิดขึ้นที่อยู่ในล็อกไฟล์จะเป็นประโยชน์ให้ทั้งผู้รับการบริการและผู้ให้บริการทำให้เกิดความเชื่อใจต่อระบบความปลอดภัยต่อคลาวด์ (Ko, 2014) โดยล็อกไฟล์ถูกแบ่งเป็น 2 ประเภทโดยจะกล่าวในย่อหน้าถัดไป

งานวิจัยของ Ko ได้อธิบายเกี่ยวกับประเภทของล็อกไฟล์ว่าจะประเภทของล็อกไฟล์ถูกแบ่งเป็น 2 ประเภท ได้แก่ 1) file-centric log และ 2) system-centric log (Ko, et al., 2011)

**1) file-centric log หรือ file-log**

file-centric log หรือ file-log ล็อกไฟล์ประเภทแรกนี้ เป็นล็อกไฟล์ที่ถูกสร้างโดยมีวัตถุประสงค์เพื่อเก็บไฟล์ และวิธีการเก็บประวัติของไฟล์นั้นจะเก็บตั้งแต่ไฟล์ถูกสร้างขึ้นจนกระทั่งไฟล์ถูกลบหายไป

**2) system-centric log หรือ system-log**

system-centric log หรือ system-log เป็นล็อกไฟล์สำหรับเก็บข้อมูลการทำงานในระดับของฮาร์ดแวร์ (hardware layer) เช่น การเข้าใช้งานในหน่วยความจำ การใช้งานบนฮาร์ดดิสก์ อุณหภูมิ แรงดันไฟฟ้า การล็อกอินเข้าสู่ระบบตั้งแต่ระดับแอดมิน (admin) จนถึงระดับยูเซอร์ (user) ต่าง ๆ รวมถึงสภาวะการทำงานต่าง ๆ ในหน่วยประมวลผลกลาง (CPU) และอื่น ๆ (Ko, et al., 2011)

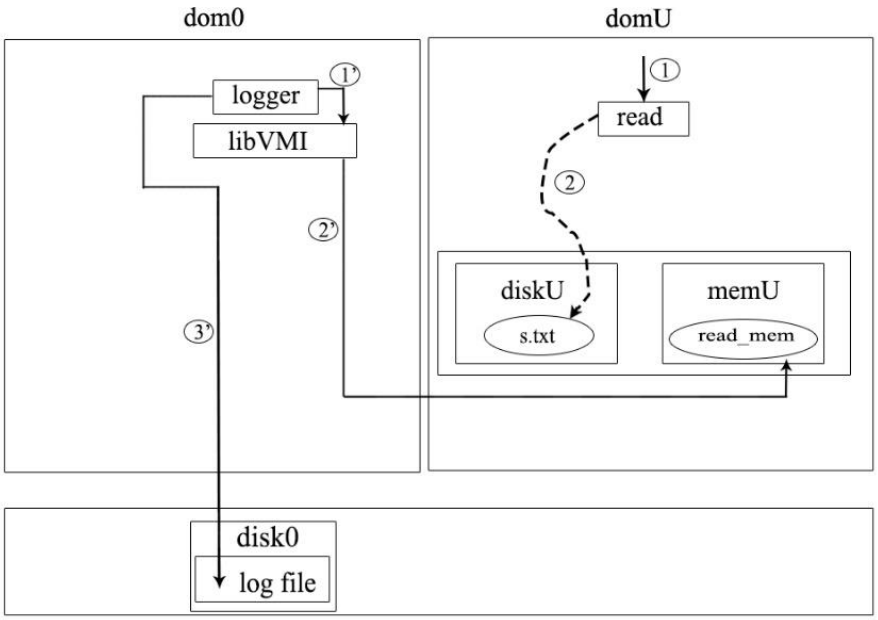
ในหัวข้อ 2.6 ล็อกไฟล์ (logs files) ผู้อ่านจะได้ทราบเกี่ยวกับประเภทของล็อกไฟล์ถูกแบ่งออกเป็น 2 ประเภท โดยในระบบบันทึกเหตุการณ์ที่ผู้วิจัยใช้ในการทดลองของวิทยานิพนธ์เล่มนี้ถือว่าเป็นล็อกไฟล์ในประเภทที่ 1 คือ file-centric log หรือ file-log เนื่องจากระบบบันทึกเหตุการณ์มีลักษณะการทำงานที่ต้องตรวจสอบเหตุการณ์ว่าใครเป็นคนเข้าถึงไฟล์อะไรและได้กระทำอะไรกับไฟล์เหล่านั้นบ้าง

## 2.7 ระบบบันทึกเหตุการณ์ (logging system)

จากข้อสรุปในหัวข้อ 2.6 เรื่อง ล็อกไฟล์ได้กล่าวว่าระบบบันทึกเหตุการณ์เป็นระบบสำหรับบันทึกเหตุการณ์เพื่อนำไปสู่หลักฐานสำหรับตรวจสอบร่องรอยของการกระทำ โดยในหัวข้อนี้ผู้วิจัยจะอธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ที่ใช้ในการทดลอง และการทำงานของระบบบันทึกหตุการณ์

โดยหัวข้อนี้จะอธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ต่อคลาวด์ที่สามารถใช้ในรูปแบบการให้บริการคลาวด์ IaaS แบบสาธารณะ ซึ่งสถาปัตยกรรมดังกล่าวเป็นการอ้างอิงจากงานวิจัยของ Wongthai และ Moorsel (Wongthai & Moorsel, 2016) โดยทางผู้วิจัยได้มีการแก้ไขภาพของสถาปัตยกรรมดังกล่าวเพื่อให้สอดคล้องกับงานทดลองในวิทยานิพนธ์เล่มนี้ โดยผู้วิจัยจะนำเสนอในส่วนที่ถูกนำไปใช้งานจริง แสดงในภาพที่ 7 ภายในสถาปัตยกรรมจะแบ่งการ

ทำงานออกเป็น 2 ส่วน คือ 1) ส่วนของผู้ใช้บริการหรือลูกค้า (domU : domain user) ซึ่งเป็นเครื่องที่ถูกตรวจสอบ 2) ส่วนของผู้ให้บริการ (dom0 : domain zero) ซึ่งจะเป็นเครื่องที่ทำการติดตั้ง logger (logger เป็นโปรเซสสำหรับตรวจจับบันทึกเหตุการณ์ที่เกิดขึ้นใน domU)




**ภาพที่ 7 สภาพแวดล้อมการทำงานระหว่าง domU and dom0**
**แก้ไขภาพจาก Wongthai (Wongthai & Moorsel, 2016).**

**หมายเหตุ** สาเหตุที่ต้องมีการแก้ไขภาพของสภาพแวดล้อมการทำงานระหว่าง dom0 และ domU เนื่องจากผู้วิจัยต้องการให้ภาพดังกล่าวสอดคล้องกับสภาพการทดลองจริงในการทดลองของวิทยานิพนธ์เล่มนี้เนื่องจากสภาพแวดล้อมการทำงานเดิมมีบางส่วนที่ผู้วิจัยไม่ได้ใช้ในงานทดลอง จึงต้องมีการแก้ไขสภาพแวดล้อมให้สอดคล้องกับการทดลองของวิทยานิพนธ์เล่มนี้

**2.7.1 ส่วนของผู้ใช้บริการหรือลูกค้า (domU) และข้อมูลในประวัติไฟล์สำคัญ**

จากภาพที่ 7 ส่วนของ domU คือ กรอบสี่เหลี่ยมขวามือที่มีคำว่า domU อยู่เหนือกรอบสี่เหลี่ยม ในส่วนนี้ลูกค้าเป็นเจ้าของ diskU (diskU เป็น virtual disk ซึ่งเป็นพื้นที่จัดเก็บข้อมูลเสมือนที่ทางผู้ใช้บริการได้เช่าพื้นที่) ภายใน diskU ได้จัดเก็บไฟล์ชื่อ s.txt โดยสมมติให้ไฟล์ดังกล่าวเป็นไฟล์ที่สำคัญของลูกค้า ภายใน diskU อาจจัดเก็บไฟล์ประเภทใดก็ได้ เช่น text file executable file หรือ database file (Wongthai & Moorsel, 2016) จากนั้นผู้วิจัยได้สมมติกระบวนการทำงานภายใน domU และกิจกรรมของแอปพลิเคชัน read สำหรับการทดลอง ดังนี้

**กระบวนการทำงานภายใน domU** มี 3 ขั้นตอน ดังนี้

1) domU อาจถูกครอบครองโดยผู้ไม่หวังดีซึ่งก็คือแฮกเกอร์ (Hacker) (จากภาพที่ 7 กล่องของ domU ตอนนี้แฮกเกอร์เป็นผู้ครอบครอง)

2) แฮกเกอร์ได้ครอบครอง domU และรันแอปพลิเคชัน read เพื่ออ่านไฟล์ s.txt ภายใน diskU (ตามเส้นประหมายเลข 2 ในภาพที่ 7) (โปรเซส คือ โปรแกรมที่ถูกรัน (A.Saha, 2006) หรือโปรแกรมที่กำลังทำงาน (Bryant & O'Hallaron, 2010))

3) เมื่อแอปพลิเคชัน read ถูกรันจากแอปพลิเคชัน read [6] จึงเปลี่ยนเป็นโปรเซส read และกระบวนการทำงานเหล่านี้จะเรียกว่ากิจกรรมของแอปพลิเคชัน read

**กิจกรรมของแอปพลิเคชัน read สำหรับการทดลอง**มี 4 ขั้นตอน ดังนี้

1) แฮกเกอร์รันแอปพลิเคชัน read เพื่อเปิดไฟล์ s.txt ภายใน domU (ตามเส้นปะหมายเลข 2 ในภาพที่ 7)

2) เมื่อแฮกเกอร์เปิดไฟล์ s.txt ได้แล้ว แฮกเกอร์ได้พิมพ์ข้อมูลภายใน s.txt และอ่านไฟล์ (ในภาพวงรีเขียนว่า s.txt ถูกเก็บในกล่องสี่เหลี่ยมเขียนว่า diskU ภาพขวามือ)

3) ปิดไฟล์

4) จบการทำงาน

ข้อมูลกิจกรรมของแอปพลิเคชัน read เหล่านี้จะถูกจัดเก็บไว้ภายใน read_mem และ read_mem จะอยู่ใน memU (memU คือส่วนของหน่วยความจำหลักหรือ RAM : Random Access Memory ของ domU) ดังแสดงในภาพที่ 7 ในส่วนของกล่องสี่เหลี่ยมเล็กขวามือเขียนว่า memU ซึ่งอยู่ภายในกล่องสี่เหลี่ยม domU โดยการตรวจสอบและบันทึกข้อมูลที่เกี่ยวข้องกับ

[6] โปรเซส คือ โปรแกรมที่ถูกรัน (A.Saha, 2006) หรือโปรแกรมที่กำลังทำงาน (Bryant & O'Hallaron, 2010)

กระบวนการเหล่านี้ เรียกว่า การบันทึกประวัติไฟล์สำคัญ ซึ่งภายใน read_mem ได้จัดเก็บข้อมูลดังแสดงในตารางที่ 2

| f_nm | p_id | p_nm | p_ownId |
|---|---|---|---|
| s.txt | 4624 | read | 1002 (alice) |

**ตารางที่ 2 ข้อมูลไฟล์สำคัญของลูกค้า**

จากตารางที่ 2 แสดงข้อมูลที่ถูกจัดเก็บใน read_mem ดังนี้

1. f_nm (ชื่อไฟล์ s.txt)
2. p_id (ไอดีของโปรเซสที่ถูกรันในที่นี้คือ 4624)
3. p_nm (ชื่อของโปรเซสที่ถูกรัน read)
4. p_ownId (ไอดีผู้รันโปรเซส สมมติคือ alice)

**ข้อมูลในประวัติไฟล์สำคัญ**

จากตารางที่ 2 ที่ผู้วิจัยได้อธิบายเกี่ยวกับข้อมูลประวัติไฟล์สำคัญที่โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์สามารถบันทึกได้ซึ่งจะมีข้อมูลด้วยกัน 4 ส่วน คือ 1. f_nm (ชื่อไฟล์ s.txt) 2. p_id (ไอดีของโปรเซสที่ถูกรัน) 3. p_nm (ชื่อของโปรเซสที่ถูกรัน) และ 4. p_ownId (ไอดีผู้รันโปรเซส) โดยข้อมูลที่บันทึกได้เหล่านี้จะสามารถนำไปเป็นหลักฐานสำหรับหาบุคคลกระทำผิดมารับผิดชอบต่อการกระทำที่เกิดขึ้น โดยผู้วิจัยได้ศึกษาตัวอย่างงานวิจัยของ Wongthai (Wongthai, 2014) ภายในงานวิจัยดังกล่าวได้ยกตัวอย่างประวัติของไฟล์สำคัญไว้และให้ข้อสังเกตุว่า มีเหตุการณ์ผิดปกติหรืออาจจะมีแฮกเกอร์เข้ามาภายในระบบ เพราะอาจเข้าดูไฟล์สำคัญดังกล่าว โดยสามารถแสดงได้ดังตารางที่ 3 ด้านล่าง ดังนี้

| f_nm (ชื่อไฟล์) | p_id (ไอดีของโปรเซสที่ถูกรัน) | p_nm (ชื่อของโปรเซสที่ถูกรัน) | p_ownID (ไอดีผู้รันโปรเซส) | Time (เวลา) |
|---|---|---|---|---|
| s.txt | 4624 | read | 1001 | t1 |
| s.txt | 4725 | read | *1000 | t2 |
| s.txt | 4726 | *rootRead | *1000 | t3 |

| | | | | |
|---|---|---|---|---|
| s.txt | 4727 | read | *1002 | t4 |
| s.txt | 46268 | read | 1001 | *t5 |

**ตารางที่ 3 บันทึกประวัติของไฟล์สำคัญ**

จากตารางที่ 3 ได้แสดงตัวอย่างข้อมูลประวัติของไฟล์สำคัญ โดยโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์สามารถบันทึกประวัติไฟล์ต่างๆ ที่อยู่ภายในเครื่องของลูกค้าหรือ domU แต่ในงานวิทยานิพนธ์เล่มนี้จะสนใจไฟล์ s.txt ซึ่งเป็นไฟล์ที่ผู้วิจัยได้สร้างและสมมติว่า ให้เป็นไฟล์ที่สำคัญของลูกค้า ดังนั้นผู้วิจัยจึงนำตัวอย่างประวัติไฟล์ s.txt มาตรวจสอบเหตุการณ์ที่ต้องสงสัย โดยภายในตารางในแถวที่มีข้อความที่มีตัวหนาและมีเครื่องหมายดอกจันท์ (*) ด้านหน้า คือเหตุการณ์ที่ต้องสงสัยว่ามีแฮกเกอร์เข้ามาอ่านไฟล์ ซึ่งในงานวิจัยนี้สมมติให้แฮกเกอร์เข้ามาอ่านไฟล์เท่านั้น โดยเหตุการณ์แรกที่อาจเป็นสาเหตุ ต้องสงสัยภายในตาราง สังเกตจากแถวที่ 4 และคอลัมภ์ที่ 3 ของตาราง ซึ่งแสดงในช่อง p_nm หรือชื่อของโปรเซสที่ถูกรันในบรรทัดที่ 4 ที่เขียนว่า *rootRead ตัวหนา ซึ่งถ้าหากลูกค้าที่เป็นเจ้าของไฟล์ตัวจริงและต้องการเข้ามาอ่านไฟล์จะต้องรันโปรเซส read เท่านั้น แต่เหตุการณ์บรรทัดที่ 4 ดังกล่าว มีผู้รันโปรเซสโดยใช้โปรเซส rootRead ดังนั้นจึงเป็นเหตุการณ์ที่สงสัยว่าอาจจะมีแฮกเกอร์เข้ามาอ่านไฟล์เพราะ โปรเซสที่ใช้ในการอ่านไฟล์ต่างจากโปรเซสปกติที่ลูกค้าใช้นั่นคือ Read

นอกจากเหตุการณ์อ่านไฟล์ที่ผู้วิจัยได้อธิบายในย่อหน้าด้านบนแล้ว ในช่องที่ 4 ลำดับถัดมาในส่วนของไอดีผู้รันโปรเซสหรือ p_ownID โดยไอดีของลูกค้าที่ถูกต้องคือ 1001 และเมื่อมีไอดีของโปรเซสอื่น ๆ เข้ามารัน เช่น *1000 และ *1002 จึงเป็นเหตุการณ์ที่สงสัยว่าแฮกเกอร์อาจจะเข้ามาอ่านไฟล์ s.txt ซึ่งเป็นไฟล์สำคัญและภายในช่องที่ 5 ลำดับสุดท้ายช่องของเวลาหรือ time ภายช่วงเวลา *t5 เป็นเหตุการณ์ที่น่าสงสัย เนื่องจากช่วงเวลาดังกล่าวเป็นช่วงเวลาที่ลูกค้าไม่ได้เข้าไปอ่านไฟล์ s.txt เช่น เวลาของ *t5 คือ เวลา 23.00 น. โดยเวลาดังกล่าวลูกค้าไม่ได้เข้าไปอ่านไฟล์ s.txt แต่ปรากฏว่าช่วงเวลาดังกล่าวกลับมีบุคคลอื่นเข้าไปอ่านไฟล์ s.txt ของลูกค้า จึงเป็นเหตุการณ์ที่น่าสงสัยว่ามีแฮกเกอร์เข้าไปอ่านไฟล์สำคัญของลูกค้า

จากเหตุการณ์ตัวอย่างที่ผู้วิจัยได้อธิบายไว้ข้างต้น หัวข้อถัดไปที่ผู้วิจัยจะอธิบาย คือ ส่วนของลักษณะการทำงานของผู้ให้บริการ (dom0) และจะอธิบายเกี่ยวกับค่าของ sleeping time และ accuracy คืออะไร โดยรายละเอียดจะอธิบายเพิ่มเติมในหัวข้อที่ 2.7.2 เรื่อง ส่วนของผู้ให้บริการ (dom0) หัวข้อถัดไป

**2.7.2 ส่วนของผู้ให้บริการ (dom0)**

ในส่วนของผู้ให้บริการหรือ dom0 แสดงในภาพที่ 7 ในกรอบสี่เหลี่ยมใหญ่ซ้ายมือด้านบนกรอบเขียนว่า dom0 สามารถอธิบายรายละเอียดได้ดังนี้

จากหัวข้อ 2.4 ได้อธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ในคลาวด์สภาพแวดล้อมและกระบวนการทำงานของระบบบันทึกเหตุการณ์ โดยในภาพที่ 7 จะแบ่งการทำงานเป็น 2 ส่วน คือ 1) ส่วนของผู้ใช้บริการหรือลูกค้าซึ่งจะถูกกล่าวในหัวข้อ 2.7.1 และ 2) ส่วนของผู้ให้บริการซึ่งจะถูกกล่าวในย่อหน้าถัดไป

ในย่อหน้านี้จะอธิบายการทำงานในส่วนของผู้ให้บริการ โดยกระบวนการทำงานในส่วนของผู้ให้บริการจะอยู่ในส่วนของ dom0 ภายใน dom0 กล่องสี่เหลี่ยมใหญ่ซ้ายมือด้านบนกล่องสี่เหลี่ยมเขียนว่า dom0 ได้ติดตั้ง logger สำหรับใช้บันทึกเหตุการณ์ โดยกระบวนการทำงานของ logger มีขั้นตอน 3 ขั้นตอนดังนี้

ขั้นที่ 1') ผู้ให้บริการรันโปรเซส logger (หมายเลข 1' ในภาพที่ 7) จากนั้นโปรเซส logger จะเรียกใช้ libVMI เพื่อเข้าไปใน memU ของ domU โดย libVMI เป็น library ที่เขียนในภาษา C ที่สามารถเข้าไปอ่านข้อมูลที่ถูกจัดเก็บภายใน read_mem ซึ่งอยู่ใน memU (ข้อมูลที่ถูกจัดเก็บใน read_mem ได้ถูกอธิบายไว้ในตารางที่ 2 โดยรายละเอียดจะอยู่ภายใต้ตารางที่ 2)

ขั้นที่ 2') หลังจาก logger เรียกใช้ libVMI แล้ว libVMI จะทำหน้าที่เข้าไปอ่านข้อมูลที่ถูกเก็บภายใน read_mem (หมายเลข 2' ในภาพที่ 7)

ขั้นที่ 3') หลังจาก logger เรียกใช้ libVMI เพื่อเข้าไปอ่านข้อมูลที่ถูกเก็บใน read_mem แล้ว logger จะนำข้อมูลใน read_mem ตามตารางที่ 2 และตารางที่ 3 มาจัดเก็บลงในล็อกไฟล์ (log files) ซึ่งอยู่ใน disk0 (หมายเลข 3' ในภาพที่ 7)

สรุปหัวข้อนี้ได้อธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ การทำงานใน domU และ dom0 โดยในหัวข้อ 2.9.1 ในส่วนท้ายของเนื้อหาก่อนย่อหน้าสุดท้าย ได้มีการอธิบายเกี่ยวกับกิจกรรมของแอปพลิเคชัน read ซึ่งกิจกรรมของแอปพลิเคชัน read ดังกล่าว จะถูกปรับปรุงเพื่อนำไปใช้ในการทดลอง ซึ่งจะอธิบายในหัวข้อ 2.8.1 เรื่อง sleeping time และ accuracy ต่อไป

## 2.8 การทดสอบประสิทธิภาพ (performance) สลีปปิ้งทาม (sleeping time) และความแม่นยำ (Accuracy)

จากข้อสรุปในหัวข้อ 2.7 ระบบบันทึกเหตุการณ์ ได้กล่าวว่าการทดสอบประสิทธิภาพเป็นเรื่องสำคัญสำหรับการพัฒนาประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ ในหัวข้อนี้จึงอธิบายเกี่ยวกับรูปแบบของการทดสอบประสิทธิภาพของเว็บแอปพลิเคชัน โดยมีรายละเอียดดังนี้

ผู้วิจัยได้สรุปงานของ Molyneaux ซึ่งได้กล่าวว่าการทดสอบประสิทธิภาพของระบบซอฟต์แวร์จะทำให้ทราบถึงประสิทธิภาพของระบบซอฟต์แวร์และสามารถวางแผนการพัฒนาหรือปรับปรุงระบบในอนาคตได้ และถ้าหากไม่มีการทดสอบประสิทธิภาพของแอปพลิเคชันจะถือว่าแอปพลิเคชันนั้นไม่น่าใช้งาน เพราะอาจเกิดปัญหาในการใช้งานตามมา และจะลดความน่าเชื่อถือในการใช้งานของลูกค้าลงเนื่องจากจะทำให้เสียเวลาและเงินโดยไม่จำเป็น นอกจากนี้แอปพลิเคชันดังกล่าวจะไม่นับเป็นสินทรัพย์ที่เชื่อถือได้ (Molyneaux, 2014) และงานวิจัยของ Meier ได้กล่าวว่าการทดสอบประสิทธิภาพเป็นกุญแจสำคัญที่สามารถลดความเสี่ยงเกี่ยวกับค่าใช้จ่ายและสร้างความเชื่อถือให้กับองค์กรหรือบริษัทได้ (Meier, et al., 2007)

รูปแบบของการทดสอบประสิทธิภาพของเว็บแอปพลิเคชันมี 4 หัวข้อ โดยมีรายละเอียด (Molyneaux, 2014) ดังนี้

1) สภาพพร้อมใช้งาน (availability)

2) เวลาตอบสนอง (response time) เป็นช่วงเวลาที่นับตั้งแต่ส่งคำสั่งเข้าเครื่องคอมพิวเตอร์จนได้รับคำสั่งตอบกลับมา

3) ปริมาณงานที่ทำในช่วงเวลาหนึ่ง (throughput)

4) ความจุ (capacity)

จากเครื่องมือที่ใช้ในการทดสอบประสิทธิภาพของแอปพลิเคชันทั้ง 4 ประเภท ผู้วิจัยเห็นว่าการทดสอบประสิทธิภาพในการทดลองเป็นการทดสอบในแบบเวลาตอบสนองหรือ response time ซึ่งเป็นการตอบสนองในช่วงเวลาขณะที่โปรเซส logger กำลังเข้าไปบันทึกเหตุการณ์ในช่วงเวลาหนึ่งและแสดงผลข้อมูลที่สามารถตรวจจับได้กลับมา โดยในงานวิทยานิพนธ์เล่มนี้มุ่งเน้นไปที่วิธีการทดสอบประสิทธิภาพแบบ response time เพื่อเป็นต้นแบบให้กับการทดสอบประสิทธิภาพในประเภทอื่น ๆ ต่อไป ดังนั้นในงานวิทยานิพนธ์เล่มนี้จึงได้ทดสอบเพียงวิธีการ response time เพียงวิธีเดียวเท่านั้น

ในหัวข้อของการทดสอบประสิทธิภาพที่ผู้วิจัยได้อธิบายเกี่ยวกับเครื่องมือที่ใช้ในการทดสอบประสิทธิภาพแล้ว ภายในหัวข้อนี้จะอธิบายเกี่ยวกับคำศัพท์ 2 คำที่ใช้สำหรับการทดลองในบทที่ 3 วิธีการดำเนินการวิจัยเพิ่มเติม ซึ่งจะทำให้ผู้อ่านเข้าใจวิธีการทดสอบประสิทธิภาพได้ง่ายยิ่งขึ้น ซึ่งคำศัพท์ทั้ง 2 คำ จะถูกระบุเป็นหัวข้อย่อยของหัวข้อนี้ คือ 2.8.1 sleeping time และ accuracy โดยรายละเอียดของ sleeping time และ accuracy สามารถอธิบายเพิ่มเติมดังนี้

ความหมายของ sleeping time และ accuracy ได้ถูกนิยามไว้ในเอกสารวิชาการของ Wongthai Chan-in และ ไกรยวิชช์ ศุภโสภาพงศ์ (Chan-in & Wongthai, 2017; ไกรยวิชช์ ศุภโสภาพงศ์ & Wongthai, 2017a) ซึ่งผู้วิจัยได้สรุปไว้ ดังนี้

**2.8.1 สลีปปิ้งทาม (sleeping time)**

จากข้อสรุปในหัวข้อ 2.7.1 ได้กล่าวว่าจะมีการปรับปรุงแอปพลิเคชัน read เพื่อใช้ในการทดลองในวิทยานิพนธ์นี้ หัวข้อนี้จึงได้อธิบายเพิ่มเติมเกี่ยวกับการปรับปรุงแอปพลิเคชัน read ด้วยการเพิ่มค่า sleeping time ก่อนนำแอปพลิเคชัน read ไปใช้ในการทดลองดังนี้

ค่า sleeping time คือ ค่าเวลาที่ถูกกำหนดเพิ่มในกระบวนการทำงานของกิจกรรมในแอปพลิเคชัน read (กิจกรรมในแอปพลิเคชัน read ถูกกล่าวไว้ในหัวข้อ 2.7.1 เรื่อง ส่วนของผู้ใช้บริการหรือลูกค้า (domU) โดยในกระบวนการปรับปรุงแอปพลิเคชัน read จะเพิ่มค่า sleeping time ก่อนการปิดการอ่านไฟล์ ซึ่งมีขั้นตอนการทำงาน ดังนี้

1) เปิดไฟล์

2) อ่านไฟล์และพิมพ์ข้อมูลใน s.txt

3) ในส่วนของก่อนการปิดการอ่านไฟล์ ผู้วิจัยได้กำหนดเวลา x ms (millisecond) เช่น x = 60 ms เพื่อไม่ให้โปรเซสปิดการทำงานทันทีแต่จะให้ปิดหลัง 60 ms (ดังนั้น 60 ms เรียกว่า sleeping time)

4) ปิดการอ่านไฟล์

5) จบการทำงาน

วิธีการทั้งหมดนี้เป็นการรันแอปพลิเคชัน read เพียง 1 ครั้ง ในการทดลองในบทที่ 3 วิธีการดำเนินวิจัย ผู้วิจัยได้รันแอปพลิเคชัน read 1000 ครั้ง จำนวน 10 รอบ

ในการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ค่าที่ใช้ทดสอบประสิทธิภาพสำหรับการทดลองนอกจากจะใช้ค่า sleeping time แล้ว อีกค่าที่ใช้เป็นเกณฑ์ทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์คือค่าความแม่นยำหรือ accuracy ซึ่งจะถูกกล่าวในหัวข้อต่อไป

**2.8.2 ความแม่นยำ (accuracy)**

รายละเอียดของค่าความแม่นยำหรือ accuracy ผู้วิจัยเคยนิยามไว้ในบทความทางวิชาการ ชื่อว่าผลกระทบของหน่วยความจำหลักต่อประสิทธิภาพของระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆและได้สรุปงานของ Wongthai (Wongthai, et al., 2013b; ไกรยวิชช์ ศุภโสภาพงศ์ & Wongthai, 2017a) ซึ่งสามารถสรุปได้ ดังนี้

ผู้วิจัยได้ทดสอบรัน logger ให้ตรวจจับไฟล์ชื่อ s.txt เมื่อมีการรันโปรเซส read จำนวน 100 ครั้ง หาก logger สามารถตรวจจับไฟล์ s.txt ได้ครบ 100 ครั้ง นั้นคือ logger มีค่าความแม่นยำหรือ accuracy 100% ค่าความแม่นยำของ logger สามารถมีได้ตั้งแต่ 0%-100% ถ้าหาก logger ไม่สามารถตรวจจับไฟล์ s.txt ได้เลย นั้นคือค่าความแม่นยำของ logger เป็น 0% ในการทดลอง ถ้าหาก logger ไม่สามารถตรวจจับไฟล์ s.txt ได้จะเรียกว่า “miss” และเมื่อ logger สามารถตรวจจับไฟล์ s.txt ได้จะเรียกว่า “hit” คำว่า miss รวมถึงการเกิดจาก logger ทำงานมีปัญหาอีกด้วย ดังนั้นถ้ารัน logger ไปจับไฟล์ชื่อ s.txt เมื่อมีการรันโปรเซส read จำนวน 100 ครั้ง และ logger ตรวจจับไฟล์ s.txt ได้ 80 ครั้ง (hit = 80) นั้นหมายความว่า logger มีค่าความแม่นยำที่ 80% (accuracy = 80%)

สรุปภาพรวมของหัวข้อนี้ผู้อ่านจะได้รู้จักวิธีการปรับปรุงประสิทธิภาพและวิธีการปรับปรุงประสิทธิภาพข้อใดที่ถูกใช้ในการทดลอง นอกจากนี้ในหัวข้อนี้ยังได้ระบุเกี่ยวกับคำศัพท์ 2 คำคือ sleeping time และ accuracy โดยคำศัพท์ทั้ง 2 คำดังกล่าวถือว่าเป็นเกณฑ์สำหรับการทดสอบประสิทธิภาพในงานวิทยานิพนธ์เล่มนี้ เมื่อผู้อ่านเข้าใจคำศัพท์ 2 คำนี้จะทำให้ผู้อ่านสามารถเข้าใจงานทดลองในบทที่ 3 วิธีการดำเนินวิจัยได้ง่ายยิ่งขึ้น นอกจากนี้ผู้วิจัยเห็นว่าถ้าหากรู้วิธีการทำงานของระบบปฏิบัติการ CPU core และ RAM จะทำให้สามารถทดสอบประสิทธิภาพได้อย่างตรงประเด็นและสามารถนำไปปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ตามวัตถุประสงค์ของการวิจัย โดยการทำงานของระบบปฏิบัติการ CPU core และ RAM ผู้วิจัยได้สรุปเอกสารที่เกี่ยวข้องและได้อธิบายเพิ่มเติมในหัวข้อที่ 2.9 ระบบปฏิบัติการหัวข้อถัดไป

## 2.9 ระบบปฏิบัติการ (Operating System)

การศึกษาข้อมูลพื้นฐานของฮาร์ดแวร์ที่จะนำมาทดลองถือว่าเป็นเรื่องสำคัญ ในหัวข้อ 2.9 ระบบปฏิบัติการนี้จึงจะอธิบายเกี่ยวกับข้อมูลพื้นฐานของฮาร์ดแวร์ที่ใช้ในการทดลองของวิทยานิพนธ์เล่มนี้

ความหมาย และนิยามของระบบปฏิบัติการทางผู้วิจัยได้สรุปงานของ Carrick Galvinc อัจจิมา เลี้ยงอยู่ Gene Cooperman และ N.P.Jouppi. (Cache, 15 Mar. 1999; Carrick, et al., 2010; Cooperman, 2003; N.P.Jouppi., 1990. ) โดยระบบปฏิบัติการ (operating system หรือ OS) เป็นซอต์ฟแวร์ตัวหนึ่งในระบบคอมพิวเตอร์โดยทั่วไปองค์ประกอบของคอมพิวเตอร์จะประกอบไปด้วย 3 ส่วน คือ ฮาร์ดแวร์ (hardware) ซอร์ฟแวร์ (software) และบุคคล (people ware) โดยในหัวข้อนี้จะมุ่งเน้นไปที่การอธิบายส่วนของฮาร์ดแวร์ที่ถูกใช้ในการทดลอง คือ หน่วยประมวลผลกลาง (CPU) และหน่วยความจำหลัก (RAM) โดยมีรายละเอียด ดังนี้

### 2.9.1 หน่วยประมวลผลกลาง (CPU : Central Processing Unit)

ในส่วนของหัวข้อ 2.9.1 จะอธิบายเกี่ยวกับหน่วยประมวลผลกลางหรือ Central Processing Unit หรือ CPU โดยในงานวิทยานิพนธ์เล่มนี้จะใช้คำว่า CPU core ซึ่งเหตุผลที่ผู้วิจัยใช้คำว่า CPU core เพื่อให้ตรงกับเอกสารการรันโปรเซสบน CPU core ของ Dan Nanni (Nanni) โดยหน่วยประมวลผลกลางเป็นส่วนของฮาร์ดแวร์ที่ผู้วิจัยใช้ในการทดลอง จึงต้องมีการอธิบายเพิ่มเติมเพื่อให้ผู้อ่านได้เข้าใจเกี่ยวกับส่วนของฮาร์ดแวร์ที่ผู้วิจัยใช้ในการทดลอง

หน่วยประมวลผลกลาง (Central Processing Unit : CPU) ทำหน้าที่ประมวลคำสั่งและควบคุมการทำงานทั้งหมดของระบบคอมพิวเตอร์ โดยจะรับข้อมูลจากหน่วยรับข้อมูลแล้วนำไปประมวลผลตามรูปแบบของข้อมูล CPU core ถือว่าเป็นหัวใจสำคัญของคอมพิวเตอร์ เนื่องจากคอมพิวเตอร์จะทำงานได้ถูกต้องและรวดเร็วเพียงใดนั้นขึ้นอยู่กับ CPU core ทั้งสิ้น ในคอมพิวเตอร์ขนาดใหญ่ เช่น Mainframe Computer หรือ Super Computer จะเรียก CPU core ว่า Multiprocessor สำหรับคอมพิวเตอร์ขนาดเล็ก เช่น Pc (personal computer) จะมี CPU core เพียงหนึ่งตัว (Single Chip) เรียกว่า Microprocessor โดยคำว่าหน่วยประมวลผลกลางถูกบัญญัติไว้ในราชบัณฑิตยสภา (ราชบัณฑิตยสภา, 2546, 2549; ศัพท์บัญญัติราชบัณฑิตยสถาน)

โปรเซสเซอร์ (processor) หรือเรียกอีกอย่างหนึ่งว่า หน่วยประมวลผลเป็นวงจรอิเล็กทรอนิกส์ที่ทำหน้าที่ประมวลผลข้อมูลและชุดคำสั่งต่าง ๆ ของโปรแกรม เพื่อขับเคลื่อนการทำงานของระบบคอมพิวเตอร์ โดยทั่วไปเมื่อกล่าวถึง processor อาจหมายถึง หน่วยประมวลผลกลางหรือ CPU core ภายในระบบคอมพิวเตอร์จะมี processor หลายลักษณะ ดังนี้

**1) หน่วยประมวลผลกลาง (Central Processing Unit : CPU)**

Central Processing Unit (CPU) คือ processor ที่ทำหน้าที่เป็นหน่วยประมวลผลกลางในการดำเนินการตามคำสั่งต่าง ๆ ของโปรแกรม ลักษณะของ CPU core จะเหมือน Chip ตัวเล็ก ๆ ภายใน CPU core จะประกอบไปด้วยวงจรอิเล็กทรอนิกส์จำนวนมาก ซึ่งไม่สามารถมองเห็นด้วยตาเปล่า กระบวนการทำงานของ CPU core จะเริ่มจากการคำนวณตัวเลขของชุดคำสั่งที่ได้รับจากผู้ใช้ จากนั้น CPU core จะอ่านชุดคำสั่งดังกล่าวแล้วแปลความหมายและทำการคำนวณเมื่อได้ผลลัพธ์ แล้วจะส่งไปยังจอภาพ

CPU core ประกอบด้วยหน่วยย่อย 2 หน่วยดังนี้ 1. หน่วยควบคุม (Control Unit : CU) และ 2. หน่วยคำนวณและตรรกะ (Arithmetic and Logical Unit : ALU)

1.1) หน่วยควบคุม (Control Unit : CU) ทำหน้าที่ควบคุมการทำงานของหน่วยอื่นๆ ทั้งหมด และคอยจัดการเวลาการประมวลผลคำสั่งที่รับเข้ามาเป็นจังหวะตามสัญญาณนาฬิกา

1.2) หน่วยคำนวณและตรรกะ (Arithmetic and Logical Unit : ALU) ทำหน้าที่ประมวลผลคำสั่งด้วยวิธีการทางคณิตศาสตร์ เช่น บวก (+) ลบ (-) คูณ (x) หาร (/) และเปรียบเทียบค่าของข้อมูล เช่น มากกว่า (>) หรือน้อยกว่า (<) เป็นต้น ข้อมูลเกี่ยวกับ CPU core ผู้วิจัยได้สรุปจากหนังสือและงานวิจัยของ Carrick Galvin และ อัจจิมา(Carrick, et al., 2010; Galvin, et al., 2006; อัจจิมา เลี้ยงอยู่ สุธี พงศาสกุลชัย และ พีรพร หมุนสนิท, 2553)

นอกจากการทำงานของ CPU core ที่ผู้วิจัยได้อธิบายเพิ่มเติมในหัวข้อที่ 2.9.1 ข้อที่ 1) หน่วยประมวลผลกลางแล้ว การทำงานภายใน CPU core จะต้องมีการทำงานร่วมกับ แคช หรือ cache ซึ่งแคชถือว่าเป็นฮาร์ดแวร์ที่สำคัญของการทดลองในงานวิทยาวิทยานิพนธ์เล่มนี้ เพราะต้องมีการทำงานร่วมกับ CPU core ดังนั้นจึงต้องมีการศึกษาสถาปัตยกรรมของ CPU core และ cache เพื่อให้สามารถปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ได้อย่างมีประสิทธิภาพ โดยผู้วิจัยจะอธิบายข้อมูลเกี่ยวกับ cache ในย่อหน้าถัดไป

**2) แคช หรือ Cache**

ในหัวข้อนี้จะอธิบายเกี่ยวกับความหมายของ cache การทำงานของ cache ตัวอย่างการทำงานของ cache รวมถึงสถาปัตยกรรมการทำงานร่วมกันระหว่าง CPU core และ cache โดยมีรายละเอียด ดังนี้

Cache คือ หน่วยความจำที่มีความเร็วสูงซึ่งสามารถเก็บข้อมูล หรือคำสั่งที่ถูกเรียกใช้หรือเรียกใช้บ่อย ๆ โดยข้อมูลและคำสั่งที่เก็บอยู่ใน cache ทำงานโดยใช้ SRAM[7] (Static RAM) (Skorobogatov, June 2002) ซึ่งจะถูกดึงไปใช้งานได้เร็วกว่าการดึงข้อมูลจากหน่วยความจำหลัก (Main memory) ซึ่งใช้ DRAM[8] (Dynamic RAM) (S. Mittal, 2012) หลายเท่าตัว

**2.1) ขั้นตอนการทำงานของแคช (cache)**

ลักษณะขั้นตอนการทำงานของ cache มีลักษณะขั้นตอนการทำงาน ดังนี้

ขณะที่โปรแกรมกำลังโปรเซสโดยผ่าน CPU core โปรแกรมนั้นจะทำการเรียกข้อมูลที่ CPU core จำเป็นต้องใช้ภายใน RAM cache ซึ่งเป็นส่วนหนึ่งของวงจรหลักในเครื่องคอมพิวเตอร์ เมื่อได้รับสัญญานการเรียกข้อมูลในขณะที่คำสั่งการเรียกข้อมูลกำลังเดินทางไปยัง RAM และ cache โปรเซสจะทำการค้นหาข้อมูลจาก RAM และส่งต่อข้อมูลไปยัง CPU core ในการค้นหาข้อมูลครั้งแรก ซึ่งอาจจะใช้เวลานาน โดยที่ตัว CPU core จะไม่สามารถทำงานอย่างอื่นได้ในเวลานั้น ในขั้นตอนการค้นหาข้อมูลนี้ cache จะทำการบันทึกข้อมูลที่ค้นพบไว้ใน high-speed memory chips ที่มีเฉพาะภายใน cache และเมื่อ cache ตรวจสอบพบว่า CPU core ได้ทำงานเสร็จสิ้นและกำลังว่างอยู่ cache จะทำการค้นหาข้อมูล ซึ่งอยู่ใกล้เคียงกับตำแหน่งของข้อมูลที่ทางโปรแกรมได้เรียกใช้ก่อนหน้านี้จาก memory address และจัดเก็บข้อมูลไว้ใน high-speed memory chips ครั้งต่อไป เมื่อทางโปรแกรมถามหาข้อมูลจาก CPU core cache จะตรวจสอบดูว่าข้อมูลที่โปรแกรมต้องการมีอยู่ใน high-speed memory chips แล้วหรือไม่ ถ้ามีอยู่แล้ว cache จะส่งข้อมูลไปให้ CPU core โดยไม่จำเป็นต้องผ่านหน่วยความจำหลักซึ่งมีการทำงานที่ช้ากว่าทำให้ CPU core สามารถลดเวลาในการทำงานได้มากขึ้น (Handy, January 27, 1998)

---

[7] SRAM (static RAM) เป็นหน่วยความจำชั่วคราวที่รักษาบิตข้อมูลในหน่วยความจำ ต้องมีกระแสไฟเลี้ยงตลอดเวลา

[8] DRAM (Dynamic RAM) เป็นหน่วยความจำประเภท RAM เช่นเดียวกับ SRAM การใช้งานต้องมีไฟเลี้ยงตลอดเวลาและยังต้องการ การ Refresh ข้อมูลเป็นระยะ ๆ เสมือนการเตือนความทรงจำซึ่งต่างจาก SRAM ที่ไม่ต้องมีการ Refresh ตลอดเวลา

ในหัวข้อที่ 2.1) ขั้นตอนการทำงานของ cache ได้อธิบายว่า cache จะเก็บข้อมูลคำสั่งต่าง ๆ หรือคำสั่งที่ถูกเรียกใช้บ่อย ๆ ไว้บน high-speed memory chips และเมื่อ CPU core ต้องการประมวลผลคำสั่งเหล่านั้นอีกก็สามารถดึงข้อมูลของคำสั่งเหล่านั้นจาก cache มาใช้งานได้ทันทีโดยที่ไม่ต้องผ่านหน่วยความจำหลัก ซึ่งจะช่วยลดเวลาในการประมวลผลของคำสั่ง และเพื่อให้ผู้อ่านได้เข้าใจเกี่ยวกับการทำงานของ cache มากยิ่งขึ้น ผู้วิจัยจะยกตัวอย่างการทำงานของ cache ไว้ในหัวข้อถัดไป

**2.2) ตัวอย่างการทำงานของ cache**

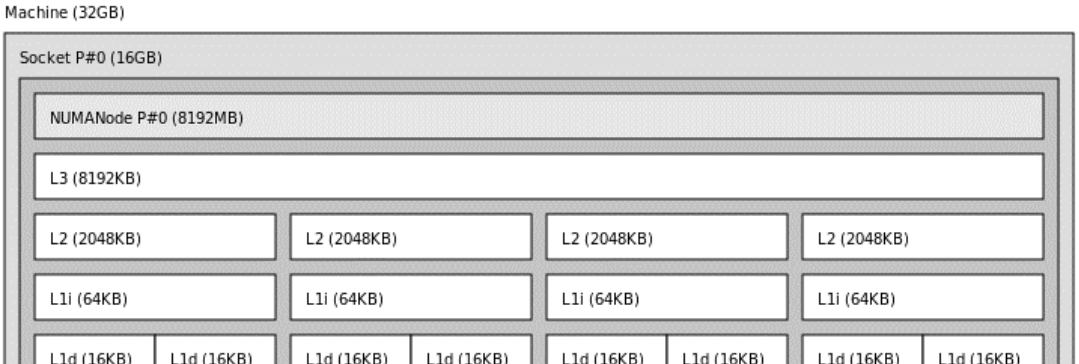
ในหัวข้อนี้จะยกตัวอย่างการทำงานของ cache เพื่อเป็นข้อมูลพื้นฐานสำหรับการอธิบายการทำงานของ CPU core และ cache ในรูปแบบของสถาปัตยกรรม

เมื่อ CPU core ทำการอ่านข้อมูล CPU core จะตรวจสอบว่าข้อมูลที่ต้องการเรียกใช้นั้นมีอยู่ใน cache หรือไม่ ถ้าหากมีข้อมูล ข้อมูลเหล่านั้นก็จะถูกถ่ายทอดไปยังหน่วยประมวลผลอย่างรวดเร็ว เพราะไม่ต้องเรียกข้อมูลจากหน่วยความจำหลัก (DRAM หรือ RAM) ซึ่งจะใช้เวลานาน แต่ถ้าตรวจสอบแล้วไม่มีข้อมูลอยู่ใน cache หน่วยประมวลผลกลางจะดึงข้อมูลจาก (DRAM หรือ RAM) จากนั้นจะทำสำเนาและเก็บข้อมูลไว้ใน cache ด้วย ถ้ามีการเรียกใช้ข้อมูลนี้อีก ข้อมูลจะถูกดึงจาก cache ทันที โดยชนิดของ cache แบ่งเป็น 2 ชนิดดังนี้

2.2.1) Level 1 (L1 cache) คือ cache ระดับ 1 จะเป็นส่วนที่สำคัญที่สุด และตำแหน่งก็จะอยู่ใกล้ๆ กับตัว CPU ที่สุด ทำให้ CPU สามารถเข้าถึงได้รวดเร็วมาก ซึ่งโดยปกติแล้วขนาดของ L1 จะมีขนาดเล็ก เช่น สำหรับ CPU Intel Pentium II หรือ Intel Celeron จะมี L1 Cache ขนาดเพียง 32 KB และบน AMD K6-2 จะมีขนาด 64 KB มีหน้าที่ในการเก็บข้อมูลของคำสั่งที่ถูกเรียกใช้จาก CPU core และบันทึกลง cache memory

2.2.2) L2 (L2 cache) คือ cache ระดับ 2 มีหน้าที่ในการเก็บข้อมูลของคำสั่งที่ถูกเรียกใช้จาก CPU core เช่นเดียวกับ L1 แต่การบันทึกข้อมูลของคำสั่งลง L2 นั้นจะกระทำหลังจากที่ พื้นที่ในการบันทึกของ L1 มีขนาดพื้นที่เต็ม

จากตัวอย่างการทำงานของ cache ที่ได้อธิบายในหัวข้อ 2.2) ตัวอย่างการทำงานของ cache จะสามารถแสดงภาพสถาปัตยกรรมได้ดังภาพที่ 8 ดังนี้



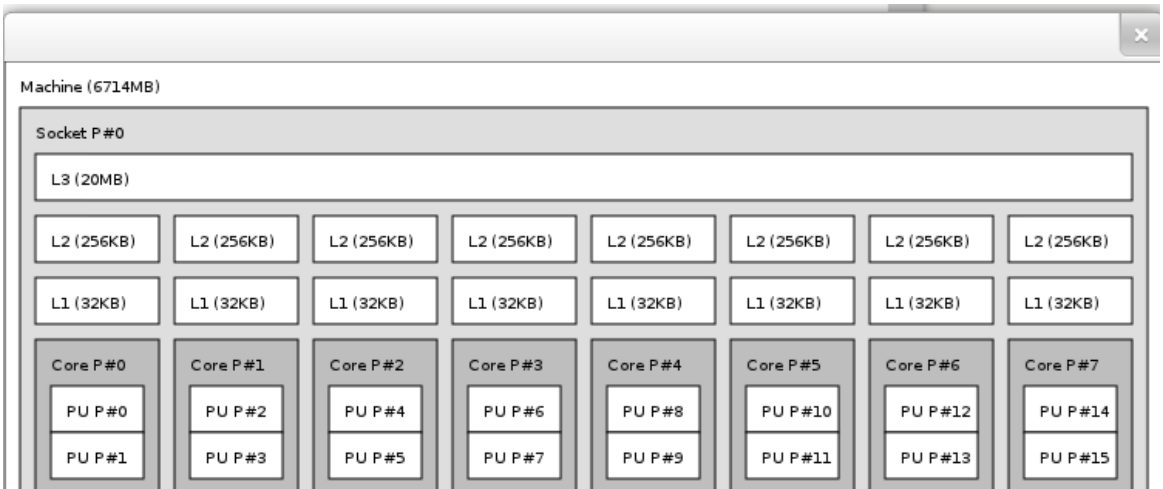
**ภาพที่ 8 ตัวอย่างสถาปัตยกรรมของ CPU core ขนาด 8 core**

จากภาพที่ 8 จะสังเกตุได้ว่าขนาดของ CPU core มีขนาด 8 core โดยเริ่มจาก Core P#0 ซึ่งหมายถึง CPU core ขนาด 1 core Core P#1 ซึ่งหมายถึง CPU core ขนาด 2 core ไปจนถึง Core P#7 ซึ่งหมายถึง CPU core ขนาด 8 core ภายในกรอบสีดำด้านล่างสุดของภาพ และแต่ละ CPU core จะมี cache L1 L2 และ L3 ดังภาพกรอบสี่เหลี่ยมสีขาว โดย cache L1 และ L2 ของแต่ละ CPU core จะทำหน้าที่เก็บข้อมูลของคำสั่งที่ถูกเรียกใช้ของแต่ละ CPU core ดังที่เคยอธิบายไว้ข้างต้น และ L3 จะเป็นพื้นที่ของ cache สำหรับแลกเปลี่ยนข้อมูลของ cache L1 และ L2 ของแต่ละ CPU core ไปยัง CPU core ตัวอื่น ๆ โดยที่ไม่ต้องประมวลผลผ่านหน่วยความจำหลัก

จากภาพตัวอย่างที่ 8 ผู้วิจัยได้อธิบายลักษณะการทำงานร่วมกันของ CPU core และ cache ในหัวข้อถัดไปผู้วิจัยจะอธิบายการทำงานของ CPU core และ cache ในสภาพแวดล้อมที่ใช้ในการทดลองบนสถาปัตยกรรมของเครื่อง Xeon ที่ใช้ในการทดลองของงานวิทยานิพนธ์เล่มนี้

### 2.3) สถาปัตยกรรมของ CPU core Xeon ที่ใช้ในการทดลอง

ในหัวข้อนี้จะอธิบายการทำงานของ CPU core และ cache บน CPU core Xeon ที่ใช้ในการทดลอง เพื่อให้เห็นผลกระทบที่เกิดขึ้นเมื่อมีการรันโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ และแนวทางการปรับปรุงประสิทธิภาพโดยการกำหนด CPU core ในการทำงานให้กับโปรเซส logger ซึ่งสามารถแสดงได้ดังภาพที่ 9



**ภาพที่ 9 ตัวอย่างสถาปัตยกรรมของ CPU core Xeon บนระบบปฏิบัติการ linux ที่ใช้ในการทดลอง**

จากภาพที่ 9 จะแสดงสถาปัตยกรรมของ CPU core Xeon บนระบบปฏิบัติการ linux ที่ใช้ในการทดลองโดยลักษณะการทำงานจะเหมือนกับการทำงานของ CPU core และ cache ทั่วไปที่ได้อธิบายไว้ในหัวข้อที่ 2) cache หรือ แคช ในหัวข้อที่ 2.9 ระบบปฏิบัติการ โดยการทำงานของ CPU core Xeon บนระบบปฏิบัติการ linux ผู้วิจัยจะเริ่มอธิบายจากกล่องสีเทาด้านล่างสุด 8 กล่อง ซึ่งภายในแต่ละกล่องสี่เหลี่ยมประกอบไปด้วยกล่องสีขาว 2 กล่องที่เขียนว่า PU P#0 PU P#1 เป็นต้น ซึ่งกล่องสีเทาที่ผู้วิจัยได้กล่าวถึง คือ กล่องของขนาด CPU core ที่มี โดยจะสังเกตได้ว่ามีกล่องสีเทา 8 กล่อง เริ่มจาก Core P#0 จนถึง Core P#7 นั้นคือ มี CPU core ขนาด 8 core และภายในแต่ละ CPU core จะมี thread 2 thread สำหรับช่วยในการทำงานของ CPU core ซึ่งในระบบปฏิบัติการ linux จะมองว่า 2 thread เป็น 1 core ดังนั้น หากต้องการกำหนดพื้นที่สำหรับการทำงานให้กับโปรเซส logger เพียง CPU core เดียว จึงต้องกำหนดเป็น 2 thread เพื่อให้การทำงานเป็น 1 core ส่วนต่อมากล่องสีขาวด้านบนของแต่ละ CPU core จะเขียนว่า L1 ซึ่ง L1 ทำหน้าที่เก็บข้อมูลคำสั่งต่าง ๆ หรือคำสั่งที่มีการเรียกใช้บ่อยในครั้งแรกไว้ เพื่อให้การประมวลผลครั้งต่อไปทำงานได้ไวยิ่งขึ้นเพราะสามารถเรียกข้อมูลที่อยู่ใน L1 ได้ทันทีโดยไม่ผ่านหน่วยความจำหลัก และลำดับชั้นต่อมาของแต่ละ CPU core ด้านบนของ L1 ที่เขียนว่า L2 โดยหน้าที่ของ L2 จะทำงานเหมือน L1 เพียงแต่ว่าจะมีขนาดที่เพิ่มขึ้นสามารถเก็บข้อมูลคำสั่งได้มากขึ้น เมื่อพื้นที่ใน L1 เต็มข้อมูลของคำสั่งต่าง ๆ หรือคำสั่งที่ถูกเรียกใช้บ่อย ๆ ก็จะถูกบันทึกลง L2 และส่วนสุดท้ายด้านบนสุดกล่องสีขาวยาวที่สุด ที่เขียนว่า L3 หน้าที่ของ L3 คือ การแลกเปลี่ยนข้อมูลที่ถูกบันทึกไว้ในแต่ละ L1 และ L2 ของแต่ละ CPU core ยกตัวอย่าง เช่น โปรเซส logger ถูกประมวลผลบน CPU core ตัวที่ 1 คือ CPU P#0 และข้อมูลของการโปรเซส logger ก็ถูกเก็บไว้ใน L1 และ L2 ของ CPU P#0 ต่อมาเมื่อโปรเซส logger ประมวลผลครั้งแรกเสร็จแล้วและจะประมวลผลครั้งที่ 2 ต่อไป แต่ระบบปฏิบัติการมองเห็นว่า CPU P#1 ว่างอยู่จึงให้โปรเซส logger ย้ายไปประมวลผลที่ CPU P#1

แต่ข้อมูลของโปรเซส logger ก็ยังถูกบันทึกไว้ที่ L1 และ L2 ของ CPU P#0 ดังนั้นระบบปฏิบัติการจึงต้องย้ายข้อมูลของโปรเซส logger จาก CPU P#0 ไปยัง CPU P#1 โดยผ่าน L3 มีการสลับข้อมูลระหว่าง CPU core

โดยขณะที่ระบบปฏิบัติการได้สลับการทำงานของโปรเซส logger ไปมาและต้องย้ายข้อมูลจาก L1 และ L2 ของแต่ละ CPU core สลับไปมาเช่นเดียวกัน ทำให้เสียเวลาในการสลับข้อมูลจึงส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง ซึ่งเป็นสมมติฐานที่ทางผู้วิจัยได้ตั้งไว้ในบทที่ 1 บทนำ แต่การวิเคราะห์ผลกระทบดังกล่าวผู้วิจัยได้ทำการทดลองและได้สรุปผลไว้ในบทที่ 3 วิธีการดำเนินวิจัย นอกจากส่วนของ CPU core ที่ใช้ในในการวิจัยที่ได้อธิบายรายละเอียดไว้ในหัวข้อที่ 2.9.1 หน่วยประมวลผลกลางแล้ว ยังมีฮาร์ดแวร์อีกส่วนที่ใช้ในการทดลอง คือ หน่วยความจำหลัก ซึ่งจะอธิบายเพิ่มเติมในหัวข้อถัดไป

### 2.9.2 หน่วยความจำ (Memory Unit)

ในส่วนของหัวข้อนี้จะอธิบายเกี่ยวกับหน่วยความจำหรือ Memory Unit ซึ่งเป็นส่วนของฮาร์ดแวร์ที่ผู้วิจัยใช้ในการทดลอง จึงต้องมีการอธิบายเพิ่มเติมในหัวข้อนี้ เพื่อให้ผู้อ่านได้เข้าใจเกี่ยวกับส่วนของฮาร์ดแวร์ที่ผู้วิจัยใช้ในการทดลอง

หน่วยความจำหลัก (main memory) เรียกอีกอย่างหนึ่งว่า “หน่วยความจำหลัก (Primary Memory/Primary Storage)” เป็นหน่วยความจำที่เก็บข้อมูลหรือคำสั่งที่รับเข้ามา เพื่อรอให้ CPU เข้าถึงข้อมูลหรือคำสั่งนั้น จากนั้นจะทำการคัดลอกมาประมวลผล หากมีการคำนวณจะถูกส่งไปยัง ALU (arithmetic and logical unit) แล้วส่งผลลัพธ์กลับมาพักไว้ที่หน่วยความจำอีกครั้ง เพื่อรอคำสั่งแสดงผลลัพธ์ต่อไป ลักษณะดังกล่าว เรียกว่า 1 รอบปฏิบัติการ หน่วยความจำหลักแบ่งออกเป็น 2 ประเภท ได้แก่ ROM และ RAM

#### 1) หน่วยความจำอ่านอย่างเดียวหรือ ROM (Read Only Memory)

ROM (Read Only Memory) เป็นหน่วยความจำที่บันทึกข้อมูลแบบถาวร (nonvolatile memory) ข้อมูลภายใน ROM จะยังคงถูกเก็บอยู่ได้โดยไม่ต้องมีไฟฟ้าเลี้ยง ROM ถูกใช้ในการบันทึกชุดคำสั่ง ROM bootstrap เพื่อสั่งให้ CPU coreทำงานเมื่อเปิดหรือรีสตาร์ทเครื่อง (restart) และชุดคำสั่ง ROM BIOS (Read Only Memory Basic I O Input Output System)

เพื่อใช้ในการส่งผ่านข้อมูลระหว่าง CPU กับแป้นพิมพ์ (keyboard) จอภาพ (monitor) และฮาร์ดแวร์อื่นๆ (hardware)

**2) หน่วยความจำหลักของคอมพิวเตอร์หรือ RAM (Random Access Memory)**

RAM (Random Access Memory) เป็นหน่วยความจำที่บันทึกข้อมูลแบบชั่วคราว (volatile memory) ดังนั้นถ้าไม่มีกระแสไฟฟ้าหรือเมื่อปิดเครื่อง ข้อมูลที่อยู่ใน RAM ก็จะหายไป RAM ใช้เก็บข้อมูลหรือชุดคำสั่งจากโปรแกรมในระหว่างที่เครื่องคอมพิวเตอร์กำลังทำงานอยู่และสามารถแบ่ง RAM ได้หลายประเภท ในงานวิจัยนี้ขอกล่าวถึงเฉพาะ SDRAM (Synchronous Dynamic RAM และ DDR SDRAM (Double data rate synchronous dynamic random-access memory)

SDRAM (Synchronous DRAM) แต่เดิม DRAM จะต้อง Refresh ตัวเองตลอดเวลาเพื่อให้ข้อมูลยังคงอยู่ทำให้เกิดการหน่วงเวลาขึ้น ต่อมาได้มีการพัฒนาให้เป็น Synchronous DRAM ซึ่งใช้สัญญาณนาฬิกาเป็นตัวกำหนดจังหวะการทำงาน จึงไม่เกิดการ หน่วงเวลา ทำให้ทำงานเร็วขึ้น ที่เรียกว่า DDR RAM (Double Data Rate SDRAM) ข้อเสียของ SDRAM คือ ใช้สัญญาณนาฬิกาเพียง 1 ใน 2 ของ 1 จังหวะสัญญาณเท่านั้น แต่ DDR SDRAM จะใช้ 1 จังหวะเต็มของสัญญาณ จึงทำให้ทำงานได้เร็วมากขึ้นเท่าตัวซึ่งเป็นการสรุปจาก Carrick Galvin และอัจจิมา เลี้ยงอยู่ (Carrick, et al., 2010; Galvin, et al., 2006; อัจจิมา เลี้ยงอยู่ สุธี พงศาสกุลชัย และ พีรพร หมุนสนิท, 2553)

ภาพรวมของหัวข้อ 2.9 ระบบปฏิบัติการ ได้อธิบายเกี่ยวกับข้อมูลพื้นฐานของฮาร์ดแวร์ที่สำคัญที่ถูกใช้ในการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ คือ 2.9.1 หน่วยประมวลผลกลาง (CPU : Central Processing Unit) และ 2.9.2 หน่วยความจำ (Memory Unit/ RAM : Random Access Memory) ซึ่งเป็นส่วนสำคัญในการพิจารณาเกี่ยวกับผลกระทบที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์ ก่อนจะนำผลกระทบดังกล่าวไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ให้ดียิ่งขึ้น

## 2.10 สรุปภาพรวมบทที่ 2

ในหัวข้อจะกล่าวถึงภาพรวมทั้งหมดของบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง โดยแบ่งรายละเอียดตามหัวข้อดังนี้

หัวข้อที่ 2.1 ความเป็นมาและนิยามของคลาวด์

หัวข้อที่ 2.2 ประเภทของคลาวด์

หัวข้อที่ 2.3 รูปแบบการให้บริการแบบ Infrastructure as a Service (IaaS) แบบสาธารณะ

หัวข้อที่ 2.4 สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ

หัวข้อที่ 2.5 ปัญหาด้านความปลอดภัยของคลาวด์

หัวข้อที่ 2.6 ล็อกไฟล์ (logs files)

หัวข้อที่ 2.7 ระบบบันทึกเหตุการณ์ (logging system)

หัวข้อที่ 2.8 การทดสอบประสิทธิภาพ (performance)

หัวข้อที่ 2.9 ระบบปฏิบัติการ

โดยรายละเอียดของข้อสรุปแต่ละหัวข้อมีรายละเอียดดังนี้

ข้อสรุปจากหัวข้อที่ 2.1 เรื่อง ความเป็นมาและนิยามของคลาวด์ได้อธิบายเกี่ยวกับองค์กรที่ทำหน้าที่เกี่ยวกับการกำหนดมาตรฐานและนิยามของคลาวด์ นั้นคือ องกรค์ NIST ซึ่งจะทำให้ผู้อ่านได้เข้าใจถึงนิยามของคลาวด์เพิ่มมากขึ้น นอกจากนี้องค์กรดังกล่าวยังได้ระบุเกี่ยวกับลักษณะสำคัญของคลาวด์ ประเภทของคลาวด์ รวมถึงรูปแบบการให้บริการของคลาวด์

ข้อสรุปภายในหัวข้อที่ 2.2 ประเภทของคลาวด์ จะทำให้ผู้อ่านได้ทราบเกี่ยวกับประเภทของคลาวด์ทั้งหมด โดยจะแบ่งตามขอบเขตการจัดการ 4 ประเภทได้แก่ 1) private cloud 2) public cloud 3) hybrid cloud และ 4) community cloud และแบ่งตามลักษณะการให้บริการ 3 ประเภท ได้แก่ 1) IaaS 2) PaaS และ 3) SaaS เนื่องจากประเภทของคลาวด์นั้นมีมากมายจึงไม่สามารถทำการทดลองได้ทั้งหมด ผู้วิจัยจึงคิดว่าหากมีการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาต่อคลาวด์บนประเภทใดประเภทหนึ่งของคลาวด์แล้ว วิธีดังกล่าวจะเป็นต้นแบบในการนำวิธีการบรรเทาไปประยุกต์ใช้กับคลาวด์ประเภทอื่น ๆ ได้เช่นเดียวกัน ดังนั้นในงานวิทยานิพนธ์เล่มนี้ผู้วิจัยจึงได้เลือกที่จะบรรเทาปัญหาให้กับคลาวด์ในรูปแบบการให้บริการแบบ Infrastructure as a Service (IaaS) แบบสาธารณะ เหตุผลที่ได้เลือกคลาวด์ประเภทนี้ คือ คลาวด์ประเภท IaaS เป็นคลาวด์ลำดับชั้นล่างสุดสำหรับการสร้างคลาวด์ในลำดับชั้นถัดมา สังเกตได้จากภาพที่ 4 ส่วนกรอบสี่เหลี่ยมล่างสุด

เขียนว่า IaaS เมื่อมีการบรรเทาปัจจัยเสี่ยงในลำดับชั้นล่างสุดแล้วก็จะทำให้การสร้างคลาวด์ในลำดับชั้นถัดมามีความปลอดภัยมากยิ่งขึ้นและอีกเหตุผลที่เลือกบรรเทาปัจจัยเสี่ยงของคลาวด์ในประเภทของคลาวด์สาธารณะเนื่องจากว่าคลาวด์สาธารณะทุกๆ คนมีสิทธิในการเข้าถึง ผู้วิจัยจึงคิดว่าเมื่อทุก ๆ คนมีสิทธิเข้าถึง การบรรเทาปัจจัยเสี่ยงให้กับคลาวด์ประเภทนี้จึงน่าจะยากกว่าคลาวด์ประเภทอื่น ๆ ดังนั้นในงานวิทยานิพนธ์เล่มนี้จึงได้เลือกที่จะทำการทดลองบนคลาวด์รูปแบบการให้บริการแบบ Infrastructure as a Service (IaaS) แบบสาธารณะ

ข้อสรุปในหัวข้อ 2.3 ความหมายของคลาวด์ IaaS แบบสาธารณะจาก Wongthai (Wongthai, et al., 2013a) คือ การให้บริการใช้คอมพิวเตอร์เสมือนหรือ VM แก่ผู้ใช้บริการ โดยในการทดลองของวิทยานิพนธ์เล่มนี้ได้ใช้คลาวด์ IaaS แบบสาธารณะและเพื่อให้ผู้อ่านเข้าใจหลักการทำงานของคลาวด์ IaaS แบบสาธารณะมากขึ้น จึงต้องมีการอธิบายเพิ่มเติมเกี่ยวกับสถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ โดยจะกล่าวเพิ่มเติมในหัวข้อที่ 2.4 เรื่อง สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ

ข้อสรุปในหัวข้อ 2.4 เรื่อง สถาปัตยกรรมของคลาวด์ IaaS แบบสาธารณะ ผู้อ่านจะเข้าใจหลักการทำงานของคลาวด์ IaaS แบบสาธารณะเพิ่มมากขึ้น และจะเป็นข้อมูลพื้นฐานสำหรับให้ผู้อ่านเข้าใจถึงวิธีการปรับปรุงประสิทธิภาพของระบบ logger ในระบบบันทึกเหตุการณ์ของวิทยานิพนธ์เล่มนี้ ซึ่งผู้วิจัยได้อธิบายเพิ่มเติมไว้ในหัวข้อที่ 2.8 การทดสอบประสิทธิภาพ (performance) นอกจากนี้ผู้วิจัยต้องการให้ผู้อ่านได้เข้าใจเกี่ยวกับปัญหาด้านความปลอดภัยของคลาวด์ก่อน โดยมีวัตถุประสงค์เพื่อให้ผู้อ่านทราบว่าปัญหาด้านความปลอดภัยของคลาวด์มีอะไรบ้างและปัญหาใดที่ผู้วิจัยสนใจที่จะบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ โดยปัญหาด้านความปลอดภัยจะกล่าวเพิ่มในหัวข้อที่ 2.5 ปัญหาด้านความปลอดภัยของคลาวด์

ข้อสรุปในหัวข้อ 2.5 การพิจารณาถึงความปลอดภัยต่อคลาวด์เป็นเรื่องที่สำคัญ ความรับผิดชอบต่อสิ่งที่เกิดขึ้น (accountability) มีระบบบันทึกเหตุการณ์ (กระบวนการทำงานของระบบบันทึกเหตุการณ์จะถูกกล่าวในหัวข้อ 2.7) เป็นส่วนประกอบสำคัญ โดยทั่วไปคำว่า accountability ได้ถูกนิยามจาก Macmillan’s dictionary (macmillandictionary) ความหมาย คือ ต้องทราบว่าใครเป็นผู้กระทำผิดและสามารถหาบุคคลที่ต้องรับผิดชอบต่อสิ่งที่เกิดขึ้น แต่ความหมายในเชิงความปลอดภัยนั้น Papanikolaou และ Pearson ได้กล่าวว่า accountability หมายถึง การจัดการความรับผิดชอบ โดยมี 3 องค์ประกอบด้วยกัน คือ 1) กลไกทางกฎหมาย 2) กลไกข้อบังคับ และ 3) กลไกทางเทคนิค (Papanikolaou & Pearson, 2012) เมื่อนำองค์ประกอบทั้ง 3 ข้อข้างต้นมาเปรียบเทียบ

ผู้วิจัยเห็นว่าระบบบันทึกเหตุการณ์ถือว่าเป็นเครื่องมือในองค์ประกอบที่ 3 นั่นคือ กลไกทางเทคนิค เนื่องจากว่าระบบบันทึกเหตุการณ์เป็นเครื่องมือสำหรับเก็บข้อมูลให้ผู้เกี่ยวข้องหรือผู้มีสิทธิ เพื่อใช้วิเคราะห์เหตุการณ์ที่เกิดขึ้น โดยเหตุการณ์เหล่านั้น อาจจะเกี่ยวข้องกับความปลอดภัยของคลาวด์

ในหัวข้อ 2.6 ล็อกไฟล์ (logs files) ผู้อ่านจะได้ทราบเกี่ยวกับประเภทของล็อกไฟล์มีประเภทใดบ้าง และจากประเภทของล็อกไฟล์ที่ถูกแบ่งออกเป็น 2 ประเภท ในระบบบันทึกเหตุการณ์ที่ผู้วิจัยใช้ในการทดลองของวิทยานิพนธ์เล่มนี้ถือว่าเป็นล็อกไฟล์ในประเภทที่ 1 คือ file-centric log หรือ file-log เนื่องจากระบบบันทึกเหตุการณ์มีลักษณะการทำงานที่ต้องตรวจสอบเหตุการณ์ว่าใครเป็นคนเข้าถึงไฟล์อะไรและได้กระทำอะไรกับไฟล์เหล่านั้นบ้าง

ข้อสรุปในหัวข้อ 2.7 ระบบบันทึกเหตุการณ์ (loggings systems) ได้อธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ การทำงานใน domU และ dom0 และในหัวข้อ 2.9.1 ในส่วนท้ายของเนื้อหาก่อนย่อหน้าสุดท้าย ได้มีการอธิบายเกี่ยวกับกิจกรรมของแอปพลิเคชัน read ซึ่งกิจกรรมของแอปพลิเคชัน read ดังกล่าว จะถูกปรับปรุงเพื่อนำไปใช้ในการทดลอง

สรุปภาพรวมในหัวข้อ 2.8 การทดสอบประสิทธิภาพ (performance) ผู้อ่านจะได้รู้จักวิธีการปรับปรุงประสิทธิภาพและวิธีการปรับปรุงประสิทธิภาพข้อใดที่ถูกใช้ในการทดลอง นอกจากนี้ในหัวข้อนี้ยังได้ระบุเกี่ยวกับคำศัพท์ 2 คำคือ sleeping time และ accuracy โดยคำศัพท์ทั้ง 2 คำดังกล่าวถือว่าเป็นเกณฑ์สำหรับการทดสอบประสิทธิภาพในงานวิทยานิพนธ์เล่มนี้ เมื่อผู้อ่านเข้าใจคำศัพท์ 2 คำนี้จะทำให้ผู้อ่านสามารถเข้าใจงานทดลองในบทที่ 3 วิธีการดำเนินวิจัยได้ง่ายยิ่งขึ้น

ภาพรวมของหัวข้อ 2.9 ระบบปฏิบัติการ ได้อธิบายเกี่ยวกับข้อมูลพื้นฐานของฮาร์ดแวร์ที่สำคัญที่ถูกใช้ในการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ คือ 2.9.1 หน่วยประมวลผลกลาง (CPU : Central Processing Unit) และ 2.9.2 หน่วยความจำ (Memory Unit/ RAM : Random Access Memory) ซึ่งเป็นส่วนสำคัญในการพิจารณาเกี่ยวกับผลกระทบที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์ ก่อนจะนำผลกระทบดังกล่าวไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ให้ดียิ่งขึ้น

เมื่อผู้อ่านได้ทราบเกี่ยวกับข้อมูลพื้นฐานต่าง ๆ ที่ผู้วิจัยได้สรุปไว้ในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้องแล้ว ข้อมูลพื้นฐานในแต่ละหัวข้อเหล่านี้จะเป็นองค์ความรู้ให้ผู้อ่านได้เข้าใจเกี่ยวกับวิธีการทดลองของวิทยานิพนธ์เล่มนี้และวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยรายละเอียดของการทดลองจะถูกกล่าวในบทที่ 3 วิธีการดำเนินการวิจัย

# บทที่ 3

# วิธีดำเนินการวิจัย

วิทยานิพนธ์เล่มนี้ได้ทำการทดสอบวัดประสิทธิภาพของระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆหรือคลาวด์และนำผลการทดลองที่ได้ไปวิเคราะห์เพื่อหาสาเหตุเกี่ยวกับผลกระทบของฮาร์ดแวร์ (RAM : Random Access Memory และ CPU : Central Processing Unit) ที่ส่งผลกระทบต่อระบบบันทึกเหตุการณ์บนคลาวด์ และนำผลวิเคราะห์ที่ได้ไปออกแบบวิธีการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์เพื่อให้ระบบบันทึกเหตุการณ์บนคลาวด์มีประสิทธิภาพที่ดีขึ้นตามสมมติฐานที่ตั้งไว้ ซึ่งผู้วิจัยได้ทำการศึกษาแนวคิด ทฤษฎี ตลอดจนเอกสารและงานวิจัยที่เกี่ยวข้องจากบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง เพื่อเป็นพื้นฐานความรู้สำหรับการวิจัยในครั้งนี้ โดยผู้วิจัยได้กำหนดวิธีการดำเนินงานวิจัยประกอบด้วย 4 หัวข้อ ดังนี้

3.1 เครื่องมือที่ใช้ในการวิจัย

3.2 กรอบวิธีการดำเนินวิจัย

3.3 วิธีออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพ

3.4 วิธีเปรียบเทียบประสิทธิภาพของระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและหลังปรับปรุงประสิทธิภาพ

3.5 วิธีการทำงานของเครื่องมือ taskset และสาเหตุที่ผู้วิจัยเลือกใช้เครื่องมือ taskset ประสิทธิภาพและหลังปรับปรุงประสิทธิภาพ

3.6 วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

3.7 สรุปภาพรวมบทที่ 3

## 3.1 เครื่องมือที่ใช้ในการวิจัย

หัวข้อนี้จะกล่าวถึงเครื่องมือที่ใช้ในงานวิจัย ซึ่งในงานวิทยานิพนธ์เล่มนี้ได้แบ่งฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (software) สำหรับการทำงานไว้อย่างละ 2 ส่วน คือ 1. ส่วนสำหรับการทดลอง และ 2. ส่วนสำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์โดยมีรายละเอียด ดังนี้

### 3.1.1 ฮาร์ดแวร์ (Hardware)

ส่วนของฮาร์ดแวร์ผู้วิจัยได้แบ่งเป็น 2 ส่วนตามที่ได้กล่าวข้างต้น คือ 1. ฮาร์ดแวร์สำหรับการทดลอง และ 2. ฮาร์ดแวร์สำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์ โดยรายละเอียดมี ดังนี้

1) ฮาร์ดแวร์สำหรับการทดลอง
    1.1) คอมพิวเตอร์ PC 1 เครื่อง (personal computer)
    1.2) CPU Intel® Xeon® Processor ขนาด 8 core
    1.3) RAM DDR 3 ขนาด 8 GB.
    1.4) Hard Disk ขนาด 320 GB.
2) ฮาร์ดแวร์สำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์
    2.1) คอมพิวเตอร์ โน๊ตบุ๊ค 1 เครื่อง (Notebook computers)
    2.2) CPU Intel® Core(TM) i7-3610QM @ 2.30 GHz 2.30 GHz
    2.3) RAM DDR 3 ขนาด 8 GB.
    2.4) Hard Disk ขนาด 700 GB.
    2.5) Graphics Card nVIDIA GEFORCE® GT 630M 2 GB.

### 3.1.2 ซอฟต์แวร์ (software)

ส่วนของซอฟต์แวร์ผู้วิจัยได้แบ่งเป็น 2 ส่วนตามที่ได้กล่าวข้างต้น คือ 1. ซอฟต์แวร์สำหรับการทดลอง และ 2. ซอฟต์แวร์สำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์ โดยรายละเอียดมี ดังนี้

1) ซอฟต์แวร์สำหรับการทดลอง
    1.1) ระบบปฏิบัติการ linux fedora 16 (64 bit)
    1.2) terminal commands linux
    1.3) screenshot
    1.4) Xen เวอร์ชัน 4.2.5 (version 4.2.5)
2) ซอฟต์แวร์สำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์
    2.1) windows 10 Professional (64-bit)
    2.2) Microsoft word 2016
    2.3) Microsoft excel 2016
    2.4) Microsoft PowerPoint 2016
    2.5) Adobe Photoshop

เมื่อผู้อ่านได้ทราบเกี่ยวกับฮาร์ดแวร์และซอฟต์แวร์ต่าง ๆ ที่เกี่ยวข้องในการทดลองของวิทยานิพนธ์เล่มนี้จากหัวข้อ 3.1 เรื่อง เครื่องมือที่ใช้ในการวิจัยนี้แล้ว ในหัวข้อถัดไปผู้วิจัยจะอธิบายเกี่ยวกับกรอบวิธีการดำเนินงานวิจัย

## 3.2 กรอบวิธีการดำเนินงานวิจัย

ในหัวข้อนี้จะนำเสนอกรอบวิธีการดำเนินการวิจัย หลักการ รวมไปถึงการออกแบบการทดลอง โดยผู้วิจัยได้แบ่งการทดลองออกเป็น 2 ส่วน คือ 3.2.1 วิธีการศึกษาและเตรียมข้อมูลสำหรับการทดลอง และ 3.2.2 วิธีการออกแบบการทดลองและวิเคราะห์ผลกระทบ โดยในแต่ละหัวข้อจะมีรายละเอียด ดังนี้

### 3.2.1 การศึกษาและเตรียมข้อมูลสำหรับการทดลอง

หัวข้อการศึกษาและเตรียมข้อมูลสำหรับการทดลองนี้ จะอธิบายเกี่ยวกับวิธีการศึกษาที่ผู้วิจัยได้ทำการศึกษา รวบรวมจนกระทั่งออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ได้ โดยมีรายละเอียด ดังนี้

ในส่วนของการศึกษาผู้วิจัยได้เริ่มทำการศึกษาเกี่ยวกับความเป็นมาของคลาวด์ ประเภทของคลาวด์ ไปจนถึงภัยคุกคามที่เกิดขึ้นกับคลาวด์เมื่อทราบว่าภัยคุกคามที่เกิดขึ้นกับคลาวด์ส่งผลกระทบต่อความเชื่อมั่นในการนำคลาวด์มาใช้งานจริงของลูกค้าที่เป็นส่วนบุคคลและองค์กร ผู้วิจัยจึงได้ศึกษาเกี่ยวกับวิธีการบรรเทาปัญหาที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ นั้นคือ ระบบบันทึกเหตุการณ์ ซึ่งระบบบันทึกเหตุการณ์เป็นหนึ่งในวิธีการที่สามารถบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้และยังมีงานวิจัยที่หลากหลายได้นำเสนอเกี่ยวกับระบบบันทึกเหตุการณ์ไว้ เช่น Ko, Ryan (Ko, et al., 2011) Wongthai และ Rocha (Wongthai, et al., 2013a; Wongthai, et al., 2013b) หรือ Pakorn (Chan-in & Wongthai, 2017) แต่อย่างไรก็ตามการพิจารณาถึงประสิทธิภาพของระบบบันทึกเหตุการณ์ก็เป็นเรื่องสำคัญ เพราะระบบบันทึกเหตุการณ์ คือ ระบบที่ต้องบันทึกเหตุการณ์เกี่ยวกับบุคคลที่เข้ามาภายในระบบ มากระทำอะไรต่อไฟล์บ้าง เช่น ลบ แก้ไข หรืออ่านไฟล์ โดยจะต้องสามารถบันทึกเหตุการณ์เหล่านั้นได้ทั้งหมดหรือคิดเป็น 100% แต่ระบบบันทึกเหตุการณ์ในปัจจุบันยังมีช่องว่างที่ไม่สามารถบันทึกเหตุการณ์เหล่านั้นได้ทั้งหมด ผู้วิจัยจึงสนใจที่จะปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยต้องการเพิ่มโอกาสให้กับระบบบันทึกเหตุการณ์สามารถบันทึกเหตุการณ์เหล่านั้นได้เพิ่มขึ้น ซึ่งเอกสารที่ผู้วิจัยได้ใช้ศึกษาและสรุป ได้ถูกกล่าวไว้ในบทที่ 2 เอกสารงานวิจัยที่เกี่ยวข้องแล้ว เมื่อผู้วิจัยเห็นปัญหาและแนวทางการปรับปรุงประสิทธิภาพ ผู้วิจัยจึงได้ออกแบบการปรับปรุงปรับประสิทธิภาพในการทดลองให้กับระบบบันทึกเหตุการณ์ โดยรายละเอียดจะอธิบายเพิ่มเติมในหัวข้อถัดๆ ไป

ในหัวข้อ 3.2.1 เรื่อง วิธีการศึกษาและเตรียมข้อมูลสำหรับการทดลองนี้ จากเอกสารที่ได้ศึกษาและค้นคว้าเพื่อเตรียมออกแบบการทดลองในการปรับปรุงประสิทธิภาพ ได้ถูกกล่าวไว้ในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง ซึ่งรายเอียดของการออกแบบการทดลองและผลของการทดลองที่ผู้วิจัยได้ออกแบบจะกล่าวเพิ่มเติมในหัวข้อถัดไป

### 3.2.2 วิธีการออกแบบการทดลองและวิเคราะห์ผลกระทบ

ในหัวข้อนี้จะอธิบายเกี่ยวกับวิธีการออกแบบการทดลอง ซึ่งผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องต่าง ๆ จากบทที่ 2 และหัวข้อที่ 2.1 การศึกษาและเตรียมข้อมูลสำหรับการทดลอง และนำข้อมูลดังกล่าวมาออกแบบการทดลอง จากนั้นนำผลของการทดลองที่ได้ไปวิเคราะห์หาสาเหตุของผลกระทบสำหรับออกแบบการทดลองใหม่เพื่อปรับปรุงประสิทธิภาพของ ระบบบันทึกเหตุการณ์บนคลาวด์ให้ดียิ่งขึ้น โดยวิธีการออกแบบการทดลองผู้วิจัยได้แบ่งส่วนของการทดลองไว้ 2 ส่วนหลัก คือ 1. ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM และ 2. ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core ซึ่งสามารถแสดงรายละเอียดได้ภายในหัวข้อย่อยถัดไป

จากหัวข้อที่ 2.9.3 sleeping time และ accuracy ในบทที่ 2 ได้มีการอธิบายเกี่ยวกับค่า sleeping time เบื้องต้นแล้ว ในหัวข้อนี้จึงจะกล่าวถึงการหาค่า sleeping time วิธีการนำค่า sleeping time ไปใช้ในงานทดลอง รวมถึงการกำหนดขนาดของ RAM และ CPU core ของการทดลองแต่ละส่วน โดยในการทดลองจะใช้สถาปัตยกรรมคลาวด์ IaaS ของ Wongthai ที่ผู้วิจัยได้ทำการปรับปรุงสถาปัตยกรรมดังกล่าว เพื่อให้สอดคล้องกับการทดลองของงานวิทยานิพนธ์เล่มนี้ดังภาพที่ 6 ของบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง หัวข้อ 2.9 ระบบบันทึกเหตุการณ์สาเหตุที่ต้องมีการแก้ไขเกี่ยวกับสถาปัตยกรรมดังกล่าว เพื่อให้สอดคล้องกับการทดลองของวิทยานิพนธ์เล่มนี้ โดยผู้วิจัยได้นำสถาปัตยกรรมของคลาวด์ IaaS ที่แก้ไขมาออกแบบการทดลอง จุดมุ่งหมายของการทดลอง คือ ทดสอบประสิทธิภาพของ logger เมื่อทำการปรับขนาดของ RAM และ CPU core ใน dom0 และ domU โดยในการทดลองได้กำหนดค่า sleeping time ไว้ที่ 65 ms ซึ่งเป็นค่าที่ได้มาจากการศึกษางานวิจัยที่เกี่ยวข้องของ Wongthai (Wongthai & Moorsel, 2016) ซึ่งเป็นค่า sleeping time ที่ดีที่สุดที่ใช้ในการทดลองดังกล่าว คือ 65 ms. จากนั้นผู้วิจัยจึงนำค่า sleeping time จากการทดลองดังกล่าว ไปใช้ในการทดลอง

#### 1) ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM

ในส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM (Random Access Memory) นี้ ผู้วิจัยได้แบ่งวิธีการออกแบบการทดลองไว้อีก 2 ส่วนย่อย คือ 1. ส่วนของการปรับขนาด RAM ในฝั่งของผู้ใช้บริการ และ 2. ส่วนของการปรับขนาด RAM ในฝั่งของผู้ให้บริการโดยรายละเอียดเพิ่มเติมจะอธิบายในแต่ละหัวข้อ ดังนี้

**1.1) การปรับขนาด RAM ในฝั่งของผู้ใช้บริการ**

ส่วนที่ 1 การปรับขนาด RAM ในฝั่งของผู้ใช้บริการหรือ customer (domU) (แสดงผลการทดลองใน หัวข้อ 4.1.1 การปรับขนาด RAM ในฝั่งของผู้ใช้บริการของบทที่ 3 วิธีการดำเนินการวิจัย) โดยผู้วิจัยได้ทำการตั้งค่า RAM ของ domU ให้มีขนาด 1GB 2GB และ 3GB ตามลำดับและ RAM ของ dom0 มีขนาดคงที่[9] ขนาด 8GB โดยที่กำหนดให้ขนาด CPU core ของ dom0 มีขนาด 8 core และ CPU core ของ domU มีขนาดระหว่าง 1 ถึง 8 core (d08c-u1c d08c-u2c d08c-u3c d08c-u4c d08c-u5c d08c-u6c d08c-u7c และ d08c-u8c บน RAM ขนาด 1 GB ของ domU) หลังจากทำการทดลองเสร็จ บันทึกผลการทดลองและทำการปรับขนาด RAM ของ domU เป็น 2GB และ 3GB ตามลำดับและทดลองโดยใช้กระบวนการเดิม

**1.2) การปรับขนาด RAM ในฝั่งของผู้ให้บริการ**

หลังจากทำการทดลองส่วนที่ 1 การปรับขนาด RAM ใน domU แล้วจะเริ่มทำการทดลองในส่วนที่ 2 (แสดงผลการทดลองในหัวข้อ 4.1.2 การปรับขนาด RAM ในฝั่งของผู้ให้บริการของบทที่ 3 วิธีการดำเนินการวิจัย) การปรับ RAM ในฝั่งของผู้ให้บริการหรือ provider (dom0) โดยผู้วิจัยได้กำหนดขนาด RAM ของ dom0 ไว้ที่ 2GB 4GB 6GB และ 8GB ตามลำดับ และกำหนด RAM ของ domU ไว้คงที่ 1GB หลังจากมีการเปลี่ยนแปลง RAM และทำการทดลองแล้วจะนำข้อมูลจากผลการทดลองที่ได้ไปวิเคราะห์เพื่อหาสาเหตุเกี่ยวกับผลกระทบที่เกิดขึ้น

**2) ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core**

ในหัวข้อนี้จะอธิบายเกี่ยวกับวิธีการทดลองในส่วนของผลกระที่เกิดจากการปรับขนาดของ CPU core (central processing unit) โดยผู้วิจัยได้แบ่งวิธีการออกแบบการทดลองไว้อีก 2 ส่วนย่อย คือ 1. การปรับขนาดของ CPU core ในฝั่งของผู้ใช้บริการ และ 2. การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการ โดยรายละเอียดทั้งหมดจะอธิบายในหัวข้อถัดไป

[9] ขนาดคงที่ คำว่าขนาดคงที่ในการทดลองของงานวิทยานิพนธ์เล่มนี้ หมายถึง ขนาดของ RAM หรือ CPU core ที่ผู้วิจัยได้ทำการทดลองโดยไม่เปลี่ยนแปลงขนาด

**2.1) การปรับขนาดของ CPU core ในฝั่งของผู้ใช้บริการ**

หัวข้อนี้จะอธิบายเกี่ยวกับการปรับขนาดของ CPU core ในฝั่งของผู้ใช้บริการหรือ customer (domU) โดยผู้วิจัยเริ่มการทดลองจากการตั้งค่าขนาด CPU core ของ dom0 ให้มีขนาดระหว่าง 1 core ถึง 8 core โดยที่ dom0 เป็นขนาดคงที่ (ขนาดคงที่ หมายถึง การกำหนดขนาดของ CPU core โดยที่ไม่เปลี่ยนแปลงขนาด) และปรับขนาด CPU core ของ domU ให้มีขนาดระหว่าง 1 core ถึง 8 core โดยที่ domU จะเป็นขนาดที่เปลี่ยนแปลง (ขนาดที่เปลี่ยนแปลง หมายถึง การกำหนดขนาดของ CPU core เปลี่ยนแปลงขนาดตั้งแต่ 1 core ถึง 8 core โดยแสดงผลเปรียบเทียบบนขนาดคงที่) เช่น กำหนดให้ขนาด CPU core ของ dom0 (ขนาดคงที่) มีขนาด 8 core และ ขนาด CPU core ของ domU (ขนาดที่เปลี่ยนแปลง) มีขนาดระหว่าง 1 core ถึง 8 core (d08c-u1c d08c-u2c d08c-u3c d08c-u4c d08c-u5c d08c-u6c d08c-u7c และ d08c-u8c) เมื่อกำหนดขนาดของ CPU core แล้ว ผู้วิจัยจะกำหนดค่า sleeping time เพื่อใช้ในการทดลอง (วิธีการหาค่า sleeping time อธิบายไว้ในบทที่ 3 หัวข้อที่ 2.2 ส่วนของ ย่อหน้าที่สองในส่วนสุดท้าย) เมื่อทำการทดลองบน CPU core dom0 ขนาด 8 core เสร็จแล้ว จะปรับค่า CPU core ของ dom0 ลงเป็น 7 core 6 core 5 core 4 core 3 core 2 core และ 1 core ตามลำดับ จากนั้นใช้กระบวนการทดลองเดิม เพื่อตรวจสอบประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ในสภาพแวดล้อมที่หลากหลาย

**2.2) การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการ**

หัวข้อนี้จะอธิบายเกี่ยวกับการปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการหรือ provider (dom0) โดยผู้วิจัยจะตั้งค่าการทดลองโดยกำหนดขนาดของ dom0 และขนาดของ domU ดังนี้ ตั้งค่าขนาดของ dom0 ให้มีขนาดระหว่าง 1 core ถึง 8 core และตั้งค่าขนาดของ domU ให้มีขนาดระหว่าง 1 core ถึง 8 core เช่นเดียวกัน ในการทดลองได้กำหนดค่า domU เป็นขนาดคงที่ dom0 เป็นขนาดที่ถูกเปลี่ยนแปลง เช่น กำหนดให้ขนาดของ domU มีขนาด 1 core และกำหนดขนาด ของ dom0 ให้มีขนาดระหว่าง 1 core ถึง 8 core (u1c-d01c u1c-d02c u1c-d03c u1c-d04c u1c-d05c u1c-d06c u1c-d07c และ u1c-d08c) หลังจากกำหนดขนาดของ CPU core แล้ว ก็จะกำหนดช่วงเวลาของ sleeping time และเมื่อทำการทดลองปรับขนาดของ dom0 ขนาดระหว่าง 1 core ถึง 8 core บน domU ขนาด 1 core เสร็จแล้ว

ก็จะทำการเพิ่มขนาด CPU core ของ dom0 ให้มีขนาด 2 core 3 core 4 core 5 core 6 core 7 core และ 8 core ตามลำดับ และใช้กระบวนการทดลองเดิม เพื่อให้ได้สภาพแวดล้อมที่หลากหลาย

ในหัวข้อ 3.2 เรื่องกรอบวิธีการดำเนินงานวิจัยได้อธิบายเกี่ยวกับวิธีการออกแบบการทดลองที่ใช้ในงานวิทยานิพนธ์เล่มนี้ โดยแบ่งเป็นหัวข้อหลัก 2 หัวข้อ โดยหัวข้อหลักแต่ละหัวข้อจะถูกแบ่งเป็นข้อย่อยอีก 2 ข้อ ทำให้สามารถออกแบบการทดลองได้ 4 รูปแบบ ซึ่งผลของการทดลองที่ได้ก็จะมี 4 ผลการทดลอง นอกจากนี้สาเหตุของผลกระทบที่เกิดขึ้นจะถูกกล่าวเพิ่มเติมในหัวข้อที่ 4.1 เรื่อง การออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพในบทที่ 4

## 3.3 เปรียบเทียบประสิทธิภาพของระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและหลังปรับปรุงประสิทธิภาพ

ในหัวข้อนี้จะกล่าวถึงการนำผลการวิเคราะห์ที่ได้ทั้งหมดไปวิเคราะห์และออกแบบวิธีการปรับปรุงประสิทธิภาพ โดยวิธีการปรับปรุงประสิทธิภาพในระบบบันทึกเหตุการณ์ที่ผู้วิจัยได้ใช้ คือ เครื่องมือในระบบปฏิบัติการลินุกซ์ (linux operating system) ซึ่งเป็นระบบปฏิบัติการที่ทางผู้วิจัยได้เลือกใช้ในการทดลอง เพราะเป็นระบบปฏิบัติการ ฟรีซอฟต์แวร์ (free software) ทำให้สะดวกต่อการนำโปรเซส logger ในระบบบันทึกเหตุการณ์ไปทดลองใช้งาน เพื่อเป็นแนวทางสำหรับการนำระบบบันทึกเหตุการณ์ดังกล่าวไปใช้งานในสภาพแวดล้อมจริง โดยเครื่องมือดังกล่าวที่ใช้ในการทดลอง เรียกว่า taskset (A.Saha, 2006) (หลักการทำงานและสาเหตุที่ผู้วิจัยเลือกใช้เครื่องมือ taskset จะกล่าวในหัวข้อ 3.4 เรื่อง วิธีการทำงานของเครื่องมือ taskset และสาเหตุที่ผู้วิจัยเลือกใช้เครื่องมือ taskset)

เมื่อผู้วิจัยเลือกใช้เครื่องมือดังกล่าวแล้ว จะนำเครื่องมือ taskset ดังกล่าวไปใช้ทดลองเพื่อปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยวิธีการทดลองจะใช้วิธีการเดิมทั้งหมดในหัวข้อ 2.2 การออกแบบการทดลองและวิเคราะห์ผลกระทบ หลังจากทำการทดลองใหม่โดยใช้เครื่องมือ taskset แล้ว ผู้วิจัยจะนำผลการทดลองที่ได้ไปสร้างกราฟเพื่อเปรียบเทียบผลการปรับปรุงประสิทธิภาพดังกล่าว โดยรายละเอียดวิธีการทดลองใหม่และผลเปรียบเทียบระหว่างการทำงานของระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและหลังการปรับปรุงประสิทธิภาพ จะกล่าวเพิ่มเติมในบทที่ 4 ผลการวิจัย

ในวิทยานิพนธ์เล่มนี้ ผู้วิจัยได้เลือกการทดลองที่ต้องการปรับปรุงประสิทธิภาพจากการวิเคราะห์ผลกระทบที่เกิดขึ้น โดยในส่วนของการทดลองที่ได้เลือกมีจำนวน 2 การทดลอง (ผลของการทดลองจะปรากฏในบทที่ 4 เรื่อง ผลการวิจัย) เหตุผลที่ผู้วิจัยได้เลือกจำนวนผลการทดลองเพียง 2 ผลการทดลองมาปรับปรุงประสิทธิภาพ เนื่องจากผลกระทบที่เกิดขึ้นจากการทดลองมีเพียง 2 การทดลองเท่านั้นที่ต้องมีการปรับปรุงประสิทธิภาพเพื่อให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดียิ่งขึ้น ส่วนการทดลองอีก 2 การทดลองนั้น ทางผู้วิจัยได้สรุปไว้ว่าการปรับขนาดของ RAM ใน dom0 และการปรับขนาดของ CPU core ใน domU ทั้ง 2 การทดลองนี้ไม่ส่งผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์หากมีการเพิ่มขนาดของฮาร์ดแวร์ จากผลสรุปทำให้ไม่จำเป็นที่ต้องปรับปรุงประสิทธิภาพในส่วนของการทดลองนี้ ส่วนการทดลองการปรับขนาดของ RAM ใน domU และการปรับขนาดของ CPU core ใน dom0 ทั้ง 2 ส่วนนี้ส่งผลกระทบต่อระบบบันทึกเหตุการณ์หากมีการเพิ่มประสิทธิภาพของฮาร์ดแวร์ จึงต้องมีการปรับปรุงประสิทธิภาพให้ดียิ่งขึ้น ซึ่งผลของการปรับปรุงประสิทธิภาพจะถูกกล่าวในบทที่ 4 ผลการวิจัย นอกจากนี้หลักการทำงานและสาเหตุที่ผู้วิจัยได้เลือกใช้เครื่องมือ taskset ในการปรับปรุงประสิทธิภาพจะกล่าวในหัวข้อถัดไป

## 3.4 วิธีการทำงานของเครื่องมือ taskset

ในหัวข้อนี้จะอธิบายเกี่ยวกับหลักการทำงานของ taskset และเหตุผลที่ผู้วิจัยเลือกใช้วิธีการดังกล่าวในการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์

เครื่องมือ taskset คือ เครื่องมือในระบบปฏิบัติการ linux ใช้สำหรับกำหนดพื้นที่ให้โปรเซสทำงาน (A.Saha, 2006) โดยสาเหตุที่วิจัยได้เลือกใช้เครื่องมือดังกล่าว เพราะว่า จากผลการวิเคราะห์ที่ผู้วิจัยเคยกล่าวไปในหัวข้อ 3.2.2 การออกแบบการทดลองและวิเคราะห์ผลกระทบ โดยเหตุผลของผลกระทบหลักที่ทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง จากการจัดสรรพื้นที่ของระบบปฏิบัติการที่มีการจัดสรรให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ได้ทำงาน โดยระบบปฏิบัติการได้มีการจองพื้นที่สำหรับให้โปรเซส logger และโปรเซสอื่น ๆ ได้ทำงานโดยมีการสลับการทำงานระหว่าง CPU core แต่ละตัว หาก CPU core ตัวใดตัวหนึ่งว่างระบบปฏิบัติการก็จะจองพื้นที่ CPU core ตัวดังกล่าวให้โปรเซสต่าง ๆ เข้ามาประมวลผลรวมถึงโปรเซส logger เช่นเดียวกัน ทำให้โปรเซส logger มีโอกาสสลับการทำงานของ CPU core ตัวหนึ่งไปยัง CPU core ตัวอื่น ๆ จึงส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง ดังนั้นเมื่อมีการใช้เครื่องมือ taskset มากำหนดขอบเขตของ CPU core ให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์

ประมวลผลเพียง CPU core เดียว จึงส่งผลให้การทำงานของโปรเซส logger ในระบบบันทึกเหตุการณ์ไม่มีการสลับการทำงานไปมาระหว่าง CPU core แต่ละตัวจึงส่งผลให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพ ที่ดีขึ้น โดยรายละเอียดของการทดลองและผลเปรียบเทียบจะอธิบายในบทที่ 4 ผลการวิจัย

ในหัวข้อนี้ได้อธิบายเกี่ยวกับการทำงานของเครื่องมือ taskset ซึ่งเป็นเครื่องมือในระบบปฏิบัติการ linux โดยผู้วิจัยได้เลือกนำมาใช้เพื่อปรับปรุงประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์ นอกจากวิธีการใช้เครื่องมือ taskset ในระบบ linux แล้ว วิธีการอื่น ๆ ที่อาจจะสามารถใช้ปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ได้ ผู้วิจัยได้ศึกษาเอกสารเพิ่มเติมและวิเคราะห์แล้วว่าวิธีการอื่น ๆ อาจจะไม่เหมาะสมที่จะนำมาใช้ปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ซึ่งจะกล่าวในหัวข้อถัดไป

## 3.5 วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

จากบทสรุปในหัวข้อที่ 3.4 เรื่อง วิธีการทำงานของเครื่องมือ taskset และสาเหตุที่ผู้วิจัยเลือกใช้เครื่องมือ taskset ได้กล่าวว่า ยังมีวิธีการอื่น ๆ ที่อาจจะใช้เพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ แต่อาจจะยังไม่เหมาะสมที่จะนำมาใช้ในการทดลองนี้ เช่น 1) วิธีการพัฒนาโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ด้วยการเขียนโปรแกรมภาษาใหม่ 2) วิธีการปรับปรุงประสิทธิภาพของโค้ดดิ้ง (coding) โดยรายละเอียดและเหตุผลดังกล่าวจะอธิบายรายละเอียดในย่อหน้าถัดไป

**1) วิธีการพัฒนาโปรเซส logger ในระบบบันทึกเหตุการณ์ด้วยการเขียนโปรแกรมภาษาใหม่**

วิธีการพัฒนาโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ด้วยการเขียนโปรแกรมภาษาใหม่ วิธีการนี้ไม่เหมาะสม เพราะว่า โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ได้ใช้ภาษาซี ( C language) ในการพัฒนาซึ่งเป็นภาษาระดับต่ำที่สุด มีการทำงานไวที่สุดที่เหมาะสมกับระบบบันทึกเหตุการณ์อยู่แล้ว ดังนั้นการเขียนภาษาใหม่จึงไม่ได้ทำให้ประสิทธิภาพ logger ในระบบบันทึกเหตุกรณ์บนคลาวด์ ทำงานได้รวดเร็วขึ้น จึงไม่ส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น

**2) วิธีการปรับปรุงประสิทธิภาพของโค้ดดิ้ง (coding)**

วิธีการปรับปรุงประสิทธิภาพของโค้ดดิ้ง วิธีการนี้น่าจะเป็นวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ได้ แต่ในงานวิทยานิพนธ์เล่มนี้ยังไม่มีการทดลองด้วยวิธีการดังกล่าว เนื่องจากจุดประสงค์ของงานวิจัยนี้ ต้องการนำเสนอแนวทางการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ด้วยวิธี taskset และวิเคราะห์เกี่ยวกับวิธีการอื่น ๆ ว่าสามารถใช้ได้หรือไม่เพื่อเป็นข้อเสนอแนะสำหรับงานทดลองในอนาคต เนื่องด้วยการทดลองจำกัดด้วยเวลาสำหรับการทดลองของผู้วิจัย

ในหัวข้อที่ 3.5 เรื่อง วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ ได้นำเสนอวิธีการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ด้วยวิธีการต่าง ๆ แต่ในงานวิทยานิพนธ์จะใช้เพียงวิธี taskset เพียงวิธีเดียว ส่วนวิธีการอื่น ๆ ที่ผู้วิจัยได้อธิบายเป็นเพียงข้อเสนอแนะสำหรับงานวิจัยในอนาคต เนื่องด้วยการจำกัดทางด้านเวลาในการทดลองของผู้วิจัยจึงสามารถทำการทดลองได้เพียงวิธีเดียว และการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ อาจจะต้องใช้วิธีการอื่น ๆ ร่วมด้วยก็ได้จึงจะทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดียิ่งขึ้นซึ่งอาจจะเป็นงานวิจัยในอนาคตต่อไป นอกจากนี้ผลการทดลองการปรับปรุงประสิทธิภาพด้วยวิธี taskset จะถูกนำเสนอในบทที่ 4 ผลการวิจัยและมีการเปรียบเทียบระหว่างผลการทดลองเก่าที่ยังไม่มีการปรับปรุงประสิทธิภาพและผลการทดลองใหม่ที่มีการปรับปรุงประสิทธิภาพแล้ว

## 3.6 สรุปภาพรวมบทที่ 3

ในหัวข้อจะกล่าวถึงภาพรวมทั้งหมดของบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง โดยแบ่งรายละเอียดตามหัวข้อดังนี้

หัวข้อที่ 1 เครื่องมือที่ใช้ในการวิจัย

หัวข้อที่ 2 กรอบวิธีการดำเนินการวิจัย

หัวข้อที่ 3 วิธีเปรียบเทียบประสิทธิภาพของระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและหลังการปรับปรุงประสิทธิภาพ

หัวข้อที่ 4 วิธีการทำงานของเครื่องมือ taskset และสาเหตุที่ผู้วิจัยเลือกใช้เครื่องมือ taskset

หัวข้อที่ 5 วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

รายละเอียดของข้อสรุปในแต่ละหัวข้อมี ดังนี้

ภายในหัวข้อ 3.1 เรื่อง เครื่องมือที่ใช้ในการวิจัยนี้ ผู้อ่านจะได้ทราบเกี่ยวกับฮาร์ดแวร์และซอฟต์แวร์ต่าง ๆ ที่เกี่ยวข้องในการทดลองของวิทยานิพนธ์นี้โดยฮาร์ดแวร์และซอฟต์แวร์แต่ละหัวข้อได้แบ่งหัวข้อย่อยไว้ 2 หัวข้อด้วยกันคือ 1. ฮาร์ดแวร์และซอฟต์แวร์สำหรับการทดลอง 2. ฮาร์ดแวร์และซอฟต์แวร์สำหรับการบันทึกผลการทดลองและเขียนวิทยานิพนธ์

ในหัวข้อ 3.2 เรื่องกรอบวิธีการดำเนินงานวิจัยได้อธิบายเกี่ยวกับวิธีการออกแบบการทดลองที่ใช้ในงานวิทยานิพนธ์เล่มนี้ โดยแบ่งเป็นหัวข้อหลัก 2 หัวข้อ โดยหัวข้อหลักแต่ละหัวข้อจะถูกแบ่งเป็นข้อย่อยอีก 2 ข้อ ทำให้สามารถออกแบบการทดลองได้ 4 รูปแบบ ซึ่งผลของการทดลองที่ได้ก็จะมี 4 ผลการทดลอง นอกจากนี้สาเหตุของผลกระทบที่เกิดขึ้นจะถูกกล่าวเพิ่มเติมในหัวข้อที่ 4.1 เรื่อง การออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพในบทที่ 4

หัวข้อ 3.3 เรื่อง วิธีเปรียบเทียบประสิทธิภาพของระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและหลังปรับปรุงประสิทธิภาพ ได้กล่าวว่าผู้วิจัยได้เลือกการทดลองที่ต้องการปรับปรุงประสิทธิภาพจากการวิเคราะห์ผลกระทบที่เกิดขึ้น โดยในส่วนของการทดลองที่ได้เลือกมีจำนวน 2 การทดลอง (ผลของการทดลองจะปรากฏในบทที่ 4 เรื่อง ผลการวิจัย) เหตุผลที่ผู้วิจัยได้เลือกจำนวนผลการทดลองเพียง 2 ผลการทดลองมาปรับปรุงประสิทธิภาพ เนื่องจากผลกระทบที่เกิดขึ้นจากการทดลองมีเพียง 2 การทดลองเท่านั้นที่ต้องมีการปรับปรุงประสิทธิภาพเพื่อให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดียิ่งขึ้น ส่วนการทดลองอีก 2 การทดลองนั้น ทางผู้วิจัยได้สรุปไว้ว่าการปรับขนาดของ RAM ใน dom0 และการปรับขนาดของ CPU core ใน domU ทั้ง 2 การทดลองนี้ไม่ส่งผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์หากมีการเพิ่มขนาดของฮาร์ดแวร์ จากผลสรุปทำให้ไม่จำเป็นที่ต้องปรับปรุงประสิทธิภาพในส่วนของการทดลองนี้ ส่วนการทดลองการปรับขนาดของ RAM ใน domU และการปรับขนาดของ CPU core ใน dom0 ทั้ง 2 ส่วนนี้ส่งผลกระทบต่อระบบบันทึกเหตุการณ์หากมีการเพิ่มประสิทธิภาพของฮาร์ดแวร์ จึงต้องมีการปรับปรุงประสิทธิภาพให้ดียิ่งขึ้น ซึ่งผลของการปรับปรุงประสิทธิภาพจะถูกกล่าวในบทที่ 4 ผลการวิจัย นอกจากนี้หลักการทำงานและสาเหตุที่ผู้วิจัยได้เลือกใช้เครื่องมือ taskset ในการปรับปรุงประสิทธิภาพจะกล่าวในหัวข้อถัดไป

ในหัวข้อที่ 3.5 เรื่อง วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ ได้นำเสนอวิธีการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ด้วยวิธีการต่าง ๆ แต่ในงานวิทยานิพนธ์จะใช้เพียงวิธี taskset เพียงวิธีเดียว ส่วนวิธีการอื่น ๆ ที่ผู้วิจัยได้อธิบายเป็นเพียงข้อเสนอแนะสำหรับงานวิจัยในอนาคต เนื่องด้วยการจำกัดทางด้านเวลาในการทดลองของผู้วิจัยจึงสามารถทำการทดลองได้เพียงวิธีเดียว และการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ อาจจะต้องใช้วิธีการอื่น ๆ ร่วมด้วยก็ได้จึงจะทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดียิ่งขึ้นซึ่งอาจจะเป็นงานวิจัยในอนาคตต่อไป นอกจากนี้ผลการทดลองการปรับปรุงประสิทธิภาพด้วยวิธี taskset จะถูกนำเสนอในบทที่ 4 ผลการวิจัยและมีการเปรียบเทียบระหว่างผลการทดลองเก่าที่ยังไม่มีการปรับปรุงประสิทธิภาพและผลการทดลองใหม่ที่มีการปรับปรุงประสิทธิภาพแล้ว

# บทที่ 4

# ผลการวิจัย

ในบทนี้ผู้วิจัยจะนำเสนอเกี่ยวกับผลการวิเคราะห์ที่ได้จากการทดลอง และได้ใช้วิธีการออกแบบการทดลองที่เคยกล่าวไว้ในบทที่ 3 วิธีการดำเนินวิจัยนำมาช่วยในการออกแบบการทดลอง นอกจากนี้การวัดประสิทธิภาพของระบบบันทึกเหตุการณ์บนคลาวด์ได้มาจากการปรับขนาดของฮาร์ดแวร์ในส่วนของ RAM และ CPU core ซึ่งรายละเอียดของการทดลองจะกล่าวเพิ่มเติมในบทที่ 4 ผลการวิจัย นอกจากนี้ผลจากการวิเคราะห์ที่ได้จากการทดลองจะตรงกับทฤษฎีที่ผู้วิจัยได้นำเสนอไว้ในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้องและสอดคล้องกับสมมติฐานที่ผู้วิจัยตั้งไว้ในบทที่ 1 บทนำ อีกด้วย โดยผลการวิเคราะห์ที่ได้จะนำไปเป็นข้อมูลพื้นฐานสำหรับปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์บนคลาวด์ จากนั้นนำผลการทดลองที่ผ่านการปรับปรุงประสิทธิภาพไปเปรียบเทียบระหว่างการทดลองเดิมและการทดลองใหม่ที่ผ่านกระบวนการปรับปรุงประสิทธิภาพโดยผู้วิจัยจะใช้เครื่องมือ taskset ในการปรับปรุงประสิทธิภาพซึ่งเป็นเครื่องมือในระบบปฏิบัติการ linux โดยรายละเอียดกล่าวไว้ในบทที่ 3 วิธีการดำเนินวิจัย หัวข้อที่ 3.4 วิธีการทำงานของเครื่องมือ taskset และสาเหตุที่ผู้วิจัยเลือกใช้เครื่องมือ taskset จากนั้นผู้วิจัยได้นำผลการทดลองใหม่ที่ผ่านการปรับปรุงประสิทธิภาพแล้วนำไปเปรียบเทียบเพื่อหาประสิทธิภาพที่ดีขึ้น โดยผู้วิจัยจะนำเสนอผลการเปรียบเทียบในรูปแบบของกราฟเส้นเปรียบเทียบและสรุปผลการเปรียบเทียบออกมาเป็นเปอร์เซ็นของประสิทธิภาพที่เพิ่มขึ้นหลังจากที่ระบบบันทึกเหตุการณ์ถูกปรับปรุงประสิทธิภาพให้ดียิ่งขึ้นและในบทที่ 4 ผลการวิจัยของวิทยานิพนธ์เล่มนี้ได้แบ่งหัวข้อไว้ 4 หัวข้อหลักดังต่อไปนี้

4.1 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM

4.2 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core

4.3 ผลส่วนที่ 1 วิเคราะห์ผลกระทบที่เกิดขึ้นกับระบบบันทึกเหตุการณ์โดยใช้หลักของระบบปฏิบัติการมาช่วยในการวิเคราะห์

4.4 ผลส่วนที่ 2 นำผลวิเคราะห์ที่ได้ไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และเปรียบเทียบผลการปรับปรุงประสิทธิภาพ

4.5 สรุปภาพรวมบทที่ 4

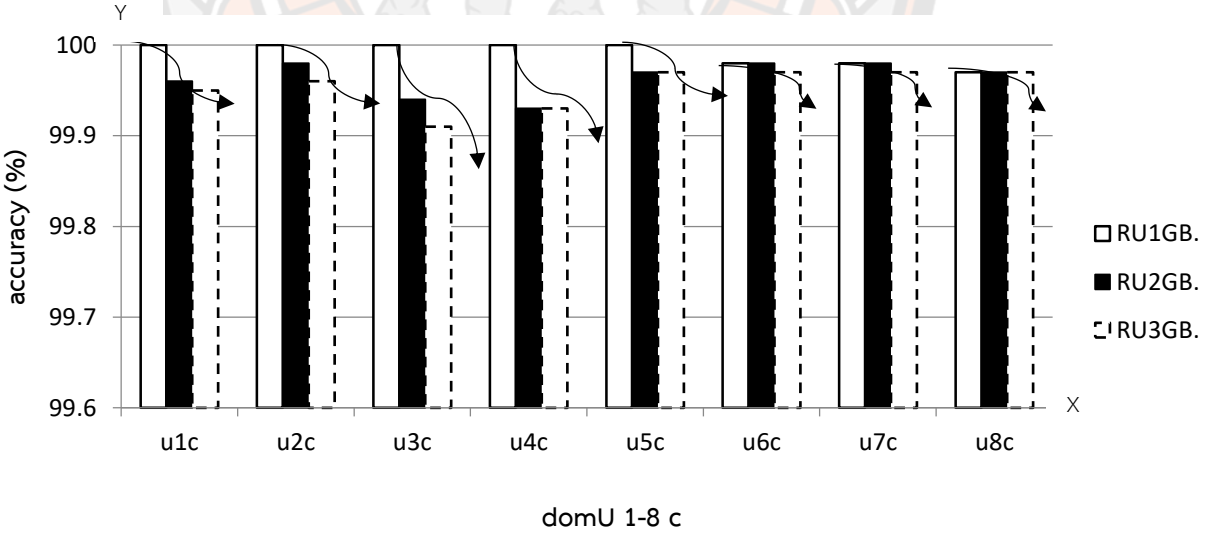
รายละเอียดเพิ่มเติมของการทดลองในแต่ละส่วนมีรายละเอียดดังนี้

## 4.1 ผลการวิจัยเกี่ยวกับผลกระทบที่เกิดจากการปรับขนาดของ RAM

ภายในหัวข้อนี้จะนำเสนอเกี่ยวกับผลการทดลองและผลวิเคราะห์ของผลกระทบเกี่ยวกับการปรับขนาดของ RAM ที่ส่งผลต่อประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ โดยวิธีการทดลองได้ถูกอธิบายไว้ในหัวข้อที่ 3.3.1 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM ในหัวข้อก่อนหน้า และผลสรุปของการทดลองได้ถูกแบ่งเป็น 2 หัวข้อย่อย คือ 4.1.1 การปรับขนาด RAM ในฝั่งของผู้ใช้บริการ และ 4.2.2 การปรับขนาด RAM ในฝั่งของผู้ให้บริการ

### 4.1.1 การปรับขนาดของ RAM ในฝั่งของผู้ใช้บริการ

จากหัวข้อ 3.1.1 เรื่อง ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAMในบทที่ 3 เรื่อง วิธีการดำเนินการวิจัย ได้อธิบายเกี่ยวกับวิธีการทดลองในส่วนของการปรับขนาด RAM แล้ว ในหัวข้อนี้จะอธิบายเกี่ยวกับผลการทดลองและผลกระทบที่เกิดขึ้นจากการปรับขนาด RAM ในฝั่งของผู้ใช้บริการหรือ customer (domU) ผลการทดลองคือ เมื่อเพิ่มขนาด RAM ให้กับ domU จะทำให้ความแม่นยำของ logger ลดลง แสดงในภาพที่ 10 ภายในภาพแสดงการเพิ่มขนาด RAM ของ domU แต่ละ CPU core ใน domU (u1c -u8c) แล้วความแม่นยำของ logger ในระบบบันทึกเหตุการณ์ลดลง



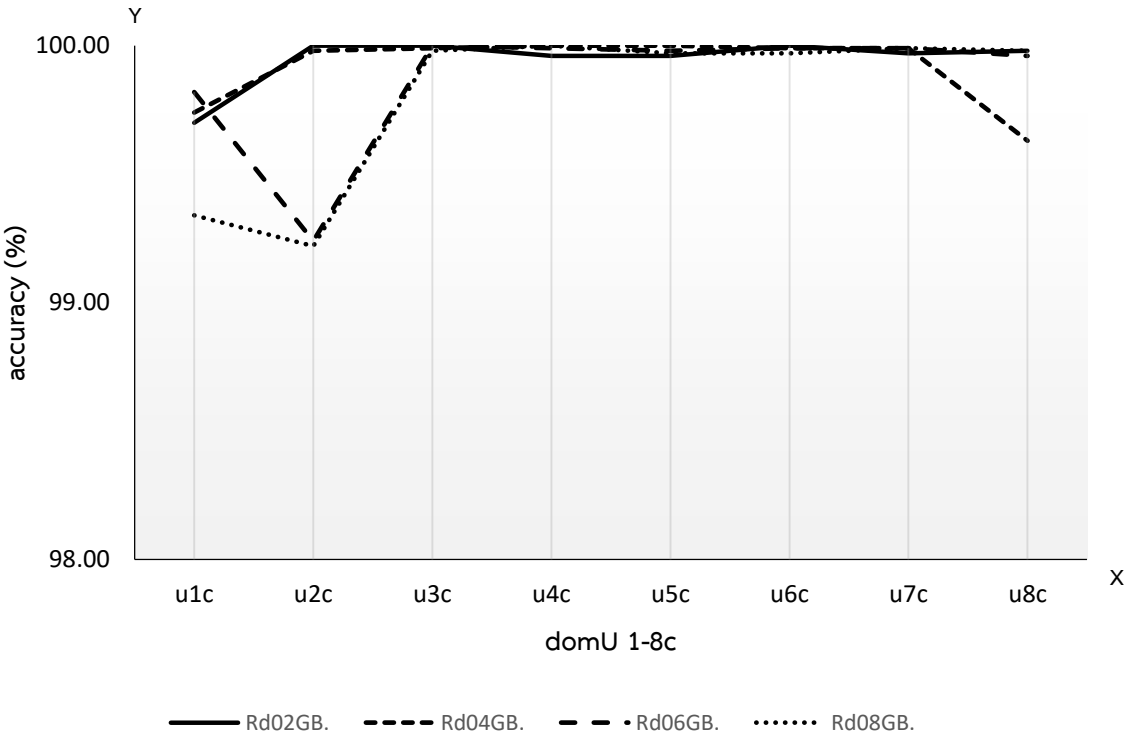

**ภาพที่ 10 ความแม่นยำของล็อกเกอร์ (logger) บน domU 1 -8 core (u1c - u8c) กับหน่วยประมวลผลกลาง (RAM) บน domU 1 -3 GB. (RU1GB., Ru2GB. and Ru3GB.)**

จากภาพที่ 10 ใช้กราฟแท่งแสดงผลการทดลองบนแกน X โดยกำหนดเป็นจำนวน CPU core ของ domU ขนาดตั้งแต่ 1 core ถึง 8 core (u1c - u8c) บนแกน Y แสดงค่าความแม่นยำของ logger ที่มีค่าตั้งแต่ 99.6% – 100% โดยกราฟแท่งเส้นทึบซ้ายสุดของแต่ละ CPU core domU (u1c u2c u3c u4c u5c u6c u7c และ u8c) แทนด้วย RAM ของ domU ขนาด 1GB (RU1GB) กราฟแท่งสีดำถัดมาแทนด้วย RAM ของ domU ขนาด 2GB (RU2GB) กราฟแท่งเส้นปะขวาสุดของแต่ละ domU แทนด้วย RAM ของ domU ขนาด 3GB (RU3GB) จะเห็นได้ว่าไม่ว่าจำนวน CPU core ของ domU มีจำนวนเท่าไรถ้าหากเพิ่มขนาดของ RAM ให้กับ domU ความแม่นยำของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์จะลดลง ส่งผลให้ประสิทธิภาพของระบบบันทึกหตุการณ์ลดลงตามไปด้วย ทั้งนี้เกิดจากเมื่อ memU (RAM) ที่อยู่ใน domU มีขนาดใหญ่ขึ้นและโปรเซส logger ในระบบบันทึกเหตุการณ์ได้เข้าไปบันทึกข้อมูลของโปรเซสที่จัดเก็บภายใน read_mem ซึ่งอยู่ใน memU ที่มีขนาดใหญ่ขึ้นทำให้ logger ต้องค้นหาโปรเซส read ในขนาดของพื้นที่ที่กว้างมากขึ้น จึงส่งผลให้โปรเซส logger ทำงานมากขึ้น นอกจากนี้ด้วยวิธีการทำงานของระบบบันทึกเหตุการณ์ก่อนการบันทึกข้อมูลของโปรเซส read ต้องมีจังหวะที่โปรเซส read หยุดการทำงานเพื่อให้โปรเซส logger เข้าไปบันทึกข้อมูลและเมื่อโปรเซส logger เข้าไปบันทึกข้อมูลโดยที่โปรเซส logger ได้เรียกใช้ libVMI เพื่อให้ libVMI เข้าไปดูข้อมูลทั้งหมดที่อยู่ภายใน RAM จากนั้นโปรเซส logger ได้อ่านไฟล์ทั้งหมด ดังนั้นเมื่อขนาดพื้นที่ของ RAM เพิ่มมากขึ้น จะทำให้โปรเซส logger ต้องทำงานมากขึ้น ความแม่นยำของ logger ดังกล่าวจึงลดลง ดังภาพที่ 10 เช่น ขณะที่ CPU core ของ domU 1 core (u1c) จะเห็นว่าเมื่อ logger เข้าไปจับโปรเซส read ใน memU ขนาด 1GB ความแม่นยำที่ได้ คือ 100% แต่เมื่อเพิ่มขนาด RAM ให้กับ memU เป็น 2GB ค่าความแม่นยำที่ได้ลดลงมาที่ 99.96% และเมื่อเพิ่มขนาด RAM เป็น 3GB ความแม่นยำจะลดลงเหลือ 99.95% ตามลำดับจึงสังเกตได้ว่าถ้าหากเพิ่มขนาดของ RAM ให้กับ domU ความแม่นยำจะลดลง

### 4.1.2 การปรับขนาดของ RAM ในฝั่งของผู้ให้บริการ

การปรับขนาดของ RAM ในฝั่งของผู้ให้บริการหรือ provider (dom0) (การทดลองส่วนที่ 2) เมื่อเพิ่ม RAM ให้กับ dom0 ความแม่นยำของ logger มีค่าความแม่นยำไม่เปลี่ยนแปลงดังแสดงในภาพที่ 10 ซึ่งแสดงผลการทดลองหลังจากเพิ่ม RAM ใน dom0 แล้วความแม่นยำไม่มีการเปลี่ยนแปลง ทั้งนี้ผู้วิจัยได้กำหนดช่วงของเปอร์เซ็นต์ที่ 99-100 ว่า logger ในระบบบันทึกเหตุการณ์มีความแม่นยำไม่ต่างกัน เพราะค่าความความต่าง 1% ที่เกิดขึ้นอาจเกิดจากระบบปฏิบัติการทำงานผิดปกติเพียงชั่วครู่ สามารถสังเกตได้ว่าเมื่อ logger มีความแม่นยำ 100%

และความแม่นยำตกมาที่ 99% จากนั้นเมื่อทำงานต่อความแม่นยำที่ตกลงมา 99% ก็กลับขึ้นไปทำงาน 100% เช่นเดิม ทางผู้วิจัยจึงเห็นว่าการทำงานของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ ถ้าความแม่นยำอยู่ในช่วงเปอร์เซ็นต์ที่ 99-100 จะถือว่า logger มีประสิทธิภาพไม่ต่างกัน




**ภาพที่ 11 ความแม่นยำของล็อกเกอร์ (logger) บน domU 1-8 core (u1c-u8c) กับหน่วยประมวลผลกลาง (RAM) in dom0 2 GB., 4 GB., 6 GB. and 8 GB. (Rd02GB., Rd04GB., Rd06GB. and Rd08GB.)**

จากกราฟผู้วิจัยจะมุ่งเน้นไปที่ส่วนของ 99.00%-100% ซึ่งในส่วนนี้เป็นส่วนของเปอร์เซ็นต์ความแม่นยำของโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ไม่เปลี่ยนแปลง ซึ่งจะใช้กราฟเส้นแสดงผลการทดลองโดยกราฟเส้นสีดำทึบแทน RAM ของ dom0 ขนาด 2GB (Rd02GB) เส้นปะถี่แทน RAM ของ dom0 ขนาด 4GB (Rd04GB) เส้นปะห่าง แทน RAM ของ dom0 ที่มีขนาด 6GB (Rd06GB) และเส้นจุดไข่ปลาแทน RAM ของ dom0 ขนาด 8GB โดยกราฟเส้นทั้ง 3 เส้น ดังกล่าวจะมีค่าที่ใกล้เคียงกัน คือ 99 -100% สาเหตุที่ทำให้ logger ทำงานได้ประสิทธิภาพความแม่นยำใกล้เคียงกัน เนื่องจากกระบวนการทำงานของระบบปฏิบัติการจะมีการจัดสรรพื้นที่ใน RAM เพื่อให้โปรเซสทำงานในขนาดที่เหมาะสมอยู่แล้ว (Carrick, et al., 2010; Deitel, et al., 2003;

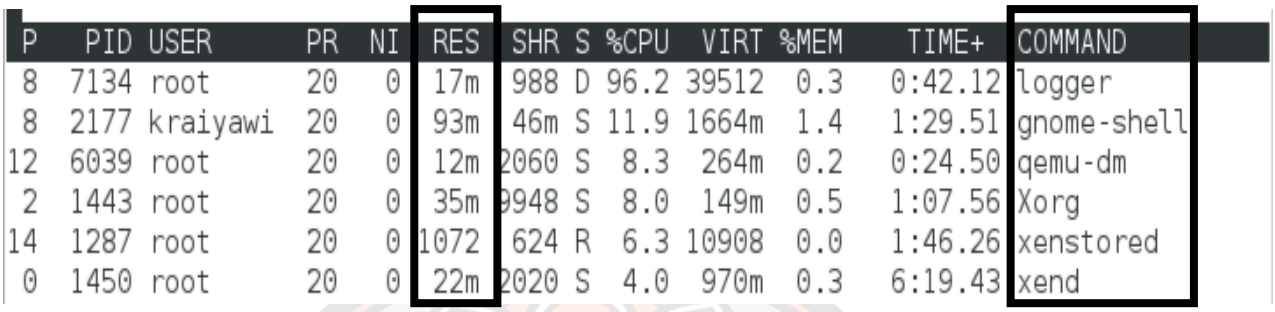
Galvin, et al., 2006; อัจจิมา เลี้ยงอยู่ สุธี พงศาสกุลชัย และ พีรพร หมุนสนิท, 2553) ดังนั้นเมื่อ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ใช้พื้นที่ไม่มากในการประมวลผลและเมื่อมีการเพิ่มขนาดของ RAM เข้าไปแต่โปรเซส logger ก็ยังคงใช้พื้นที่ที่ระบบปฏิบัติการจัดสรรให้ประมวลผลคงเดิม จึงไม่ส่งผลต่อการทำงานของโปรเซส logger จึงทำให้ค่าความแม่นยำคงเดิม ซึ่งสามารถสังเกตได้จากพื้นที่ของ logger ขณะที่ โปรเซสบน RAM ได้ใน ตารางที่ 4

| **RAM dom0** | **logger โปรเซสบน RAM** |
|---|---|
| 2 GB. | 27.25 m |
| 4 GB. | 27m |
| 6 GB. | 27m |
| 8 GB. | 27.25m |

**ตารางที่ 4 ล็อกเกอร์ตรวจจับไฟล์สำคัญของลูกค้าขณะทำงานบนขนาดของพื้นที่หน่วยประมวลผลกลาง (RAM) จำนวน 1000 ครั้ง**

จากตารางที่ 4 แสดงขนาดของพื้นที่ขณะที่โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์กำลังโปรเซสบน RAM โดยวิธีการหาพื้นที่ขณะโปรเซสกำลังทำางานอยู่ หาได้จากคำสั่ง top ซึ่งอยู่ในระบบปฏิบัติการ linux โดยคำสั่ง top จะแสดงค่าสถานะต่าง ๆ (Shotts., 2012) และผู้วิจัยจะเน้นไปที่ผลลัพธ์จาก RES โดยค่า RES คือ การใช้หน่วยความจำในส่วนเฉพาะภายใน RAM ซึ่งก็คือ พื้นที่ที่ RAM กำลังถูกโปรเซส (Deitel, et al., 2003; Galvin, et al., 2006) และในการทดลองได้รันโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ ให้จับโปรเซส read จำนวน 1000 รอบ ทำการทดสอบซ้ำไปมาจำนวน 10 ครั้ง และหาค่าเฉลี่ยในสภาวะ RAM ที่ต่างกัน คือ RAM ของ dom0 ขนาด 2GB 4GB 6GB และ 8GB จะสังเกตุได้ว่าไม่ว่า RAM จะมีขนาดเท่าไร โปรเซส logger ในระบบบันทึกเหตุการณ์จะใช้พื้นที่ในการโปรเซสที่ใกล้เคียงกัน จึงสรุปได้ว่า หากมีการเพิ่ม RAM ให้กับ dom0 จะไม่ส่งผลต่อประสิทธิภาพความแม่นยำของ logger ในระบบบันทึก

เหตุการณ์บนคลาวด์ โดยสถานะการทำงานของคำสั่ง top ที่ผู้วิจัยได้ใช้ในการทดลอง แสดงได้ดังภาพที่ 12 แสดงพื้นที่ที่โปรเซส logger กำลังโปรเซสบนระบบปฏิบัติการ



| P | PID | USER | PR | NI | RES | SHR | S | %CPU | VIRT | %MEM | TIME+ | COMMAND |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 7134 | root | 20 | 0 | 17m | 988 | D | 96.2 | 39512 | 0.3 | 0:42.12 | logger |
| 8 | 2177 | kraiyawi | 20 | 0 | 93m | 46m | S | 11.9 | 1664m | 1.4 | 1:29.51 | gnome-shell |
| 12 | 6039 | root | 20 | 0 | 12m | 2060 | S | 8.3 | 264m | 0.2 | 0:24.50 | qemu-dm |
| 2 | 1443 | root | 20 | 0 | 35m | 9948 | S | 8.0 | 149m | 0.5 | 1:07.56 | Xorg |
| 14 | 1287 | root | 20 | 0 | 1072 | 624 | R | 6.3 | 10908 | 0.0 | 1:46.26 | xenstored |
| 0 | 1450 | root | 20 | 0 | 22m | 2020 | S | 4.0 | 970m | 0.3 | 6:19.43 | xend |

**ภาพที่ 12 ขนาดพื้นที่ที่โปรเซส logger กำลังดำเนินการ (โปรเซส) บนระบบปฏิบัติการ**

จากภาพที่ 12 จะแสดงสถานะการทำงานของโปรเซสผ่านคำสั่ง top โดย ผู้วิจัยจะอธิบายเฉพาะในส่วนของกรอบสีดำ 2 กรอบ กรอบสีดำแรก คือ RES เป็นขนาดพื้นที่ที่ RAM กำลังถูกโปรเซสโดยข้อมูลขนาดของพื้นที่ที่โปรเซส logger ใช้ในการโปรเซสแสดงดังตารางที่ 4 ด้านบน ส่วนกรอบสีดำหลังสุด ช่อง COMMAND หมายถึงโปรเซสที่กำลังรันอยู่บนระบบปฏิบัติการ โดยจะสังเกตได้ว่าภายในเครื่องโปรเซส logger จะไม่ได้ถูกรันเพียงโปรเซสเดียวแต่จะมีโปรเซสอื่น ๆ ทำงานร่วมด้วย แต่ในงานวิทยานิพนธ์จะสนใจเพียงการทำงานของโปรเซส logger ในช่องของ RES เท่านั้น

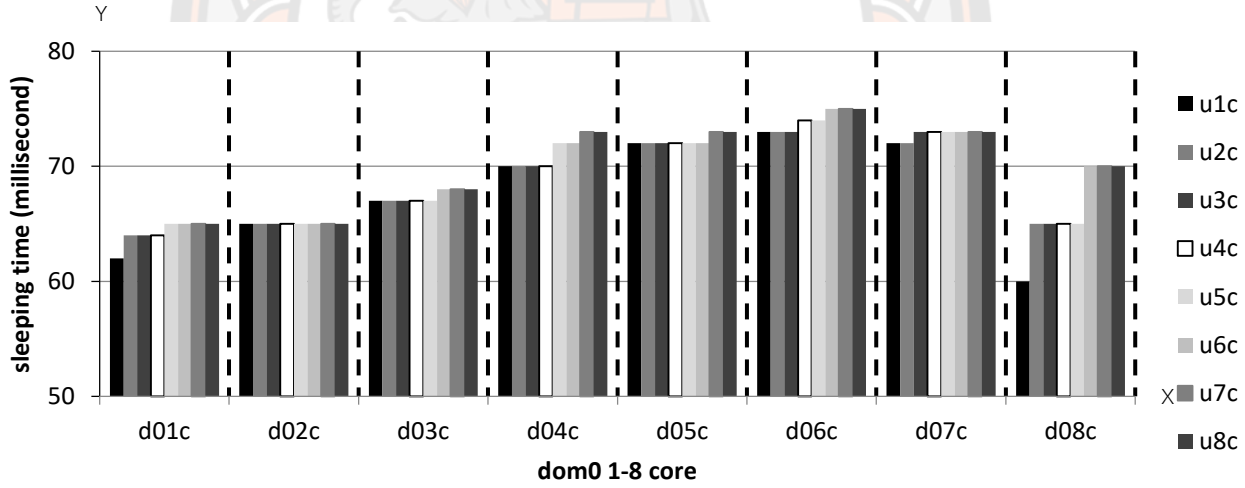
ในหัวข้อ 4.1 เรื่อง ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM หัวข้อนี้จะอธิบายเกี่ยวกับการทดลองที่ผู้วิจัยได้ทำการปรับขนาดของ RAM ทั้งในส่วนของผู้ให้บริการและผู้ใช้บริการ จากนั้นบันทึกผลการทดลองและสรุปเกี่ยวกับผลกระทบที่เกิดขึ้น ในส่วนของการทดลองแรกเมื่อมีการปรับขนาด RAM ในฝั่งผู้ใช้บริการแล้วผลกระทบที่ตามมาคือประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง โดยสาเหตุที่เกิดขึ้นได้ถูกอธิบายไว้ในหัวข้อที่ 4.1.1 การปรับขนาด RAM ในฝั่งของผู้ใช้บริการ และอีกผลการทดลองที่ทางผู้วิจัยได้ทดลองปรับขนาด RAM ในฝั่งผู้ให้บริการผลปรากฏว่าเมื่อมีการปรับขนาดของ RAM ในส่วนนี้ไม่ส่งผลกระทบใด ๆ ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์และเมื่อผู้วิจัยได้ทดลองในส่วนของการปรับขนาด RAM สำเร็จแล้วการทดลองต่อไปคือการปรับขนาดของ CPU core และสรุปผลกระทบที่เกิดขึ้น ซึ่งผลกระทบที่เกิดจากการปรับขนาดของ CPU core จะกล่าวในหัวข้อถัดไป

## 4.2 ผลการวิจัยเกี่ยวกับผลกระทบที่เกิดจากการปรับขนาดของ CPU core

ในหัวข้อนี้จะนำเสนอเกี่ยวกับผลการทดลองและผลการวิเคราะห์ของผลกระทบเกี่ยวกับการปรับขนาดของ CPU core ที่ส่งผลต่อประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ โดยส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core ในหัวข้อนี้สามารถสรุปผลของการทดลองได้เป็น 2 หัวข้อ คือ 4.2.1 การปรับขนาดของ CPU core ในฝั่งของผู้ใช้บริการ และ 4.2.2 การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการ

### 4.2.1 การปรับขนาดของ CPU core ในฝั่งของผู้ใช้บริการ

ในส่วนที่ 1 การปรับขนาดของ CPU core ใน domU ผลการทดลองปรากฏว่าเมื่อมีการเพิ่มขนาดของ CPU core ในฝั่งของผู้ใช้บริการหรือ customer (domU) ค่า sleeping time มีค่าใกล้เคียงกันส่งผลให้ประสิทธิภาพของโปรเซส logger คงเดิม ช่วงเวลาของ sleeping time ถ้าหากอยู่ในช่วงเวลาที่แตกต่างไม่เกิน 10 ms ผู้วิจัยนิยามให้โปรเซส logger มีประสิทธิภาพคงเดิม แสดงในภาพที่ 13



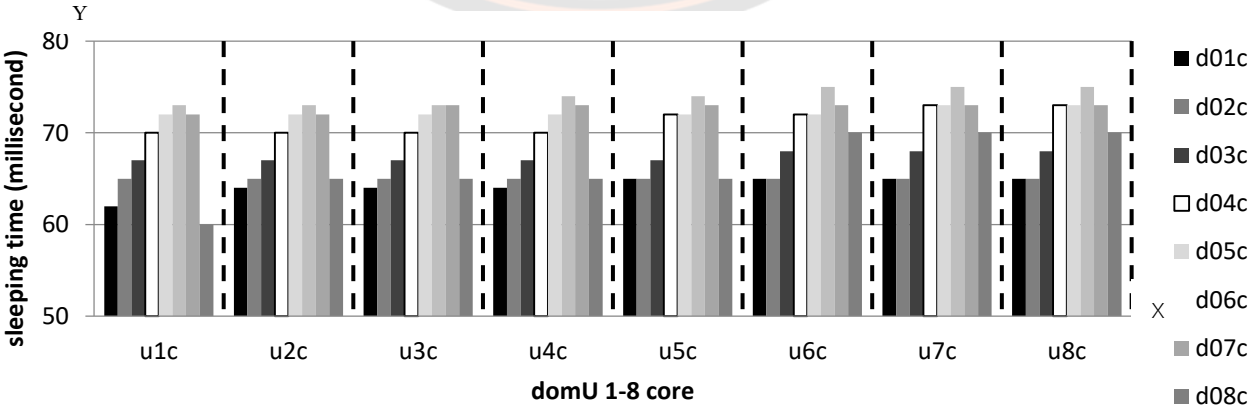

**ภาพที่ 13 สลีปปิ้งทาม (sleeping time) ของ domU's ขนาด 1 - 8 core บน dom0's ขนาด 1-8 core**

จากภาพที่ 13 จะใช้กราฟแท่งแสดงผลการทดลองโดยจะกำหนดแกน X แทนด้วยขนาด CPU core ของ dom0 ขนาดระหว่าง 1 core ถึง 8 core บนแกน Y แสดงค่าของ sleeping time และกราฟแท่งแทนขนาด CPU core ใน domU ขนาดระหว่าง 1 core ถึง 8 core โดยจะกำหนดกราฟแท่งสีดำซ้ายสุดของแต่ละ dom0 แทนด้วย domU ขนาด 1 core (u1c) กราฟแท่งลำดับถัด

มาแทนด้วย domU ขนาด 2 core (u2c) ลำดับถัดมาแทนด้วย domU ขนาด 3 core (u3c) domU ขนาด 4 core (u4c) domU ขนาด 5 core (u5c) domU ขนาด 6 core(u6c) domU ขนาด 7 core (u7c) ตามลำดับจนถึงกราฟแท่งสีเทาในลำดับสุดท้ายของแต่ละ dom0 แทนด้วย domU ขนาด 8 core (u8c) โดยกราฟแท่งดังกล่าว จะแสดงในลักษณะเปรียบเทียบทีละชุด โดยขนาด CPU core ของ dom0 เป็นค่าคงที่ และกราฟแท่งบนแกน X จะแสดงขนาด CPU core ของ domU เป็นค่าที่เปลี่ยนแปลง เช่น ในชุด CPU core ของ dom0 ขนาด 1 core (d01c) และ CPU core ของ domU ขนาดระหว่าง 1 core ถึง 8 core (u1c-u8c) นำมาเปรียบเทียบกัน (d01c-u1c d01c-u2c d01c-u3c d01c-u4c d01c-u5c d01c-u6c d01c-u7c และ d01c-u8c) จะเห็นว่าค่าเวลาใน sleeping time ของ d01c จะมีความต่างของช่วงเวลาอยู่ที่ 62-65 ms ค่า sleeping time ใกล้เคียงกันจึงทำให้ประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์คงเดิม สาเหตุที่ในการทดลองค่าของ sleeping time ใกล้เคียงกันเกิดจากการทำงานของโปรเซส logger จะถูกประมวลผลบน dom0 ซึ่งในงานทดลองได้ติดตั้ง logger ไว้ใน dom0 โดยสามารถสังเกตได้จากสถาปัตยกรรมของคลาวด์ IaaS ที่ใช้ในการทดลองในภาพที่ 13 ของบทที่ 2 ซึ่งเมื่อเพิ่มขนาด CPU core ใน domU หรือเครื่องผู้ใช้บริการ จึงไม่ส่งผลต่อการทำงานของโปรเซส logger เพราะเป็นการประมวลผลการทำงานจากคนละส่วน

### 4.2.2 การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการ (ส่วนของ CPU core ถูกกำหนดให้มีขนาด 1 core)

การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการในส่วนที่ 2 การปรับขนาด CPU core ใน dom0 นี้ผลการทดลองปรากฏว่าการเพิ่มขนาดของ CPU core ให้กับ dom0 เกิดกราฟคล้ายระฆังคว่ำดังภาพที่ 14 จุดที่ดีที่สุดเมื่อ CPU core dom0 มีขนาด 1 core และ 8 core




**ภาพที่ 14 สลีปปิ้งทาม (sleeping time) ของ dom0's ขนาด 1-8 core บน domU's ขนาด 1-8 core**

จากภาพที่ 14 ใช้กราฟแท่งแสดงผลการทดลองโดยกำหนดแกน X แทนด้วยขนาดของ CPU core ใน domU ขนาดระหว่าง 1 core ถึง 8 core บนแกน Y แสดงค่า sleeping time และกราฟแท่งแสดง CPU core ของ dom0 ขนาดระหว่าง 1 core ถึง 8 core โดยลักษณะของกราฟจะแบ่งเป็นชุด ชุดละ 8 แท่ง ดังนี้ กราฟแท่งสีดำฝั่งซ้ายสุดแต่ละ CPU core ของ domU แทนด้วย dom0 ขนาด 1 core (d01c) ลำดับถัดมาแทนด้วย dom0 ขนาด 2 core (d02c) 3 core (d03c) 4core (d04c) 5 core (d05c) 6 core (d06c) 7 core (d07c) ตามลำดับและกราฟแท่งสีเทาฝั่งขวาสุดของแต่ละ CPU core domU แทนด้วย CPU core ขนาด 8 core (d08c) โดยกราฟนี้จะแสดงการเปรียบเทียบเมื่อเพิ่ม CPU core ใน dom0 แล้วเกิดลักษณะที่คล้ายระฆังคว่ำ วิธีการเปรียบเทียบสามารถดูได้จากกราฟทีละชุดโดยให้ domU มีขนาดคงที่ dom0 มีขนาดที่เปลี่ยนแปลงระหว่าง 1 core ถึง 8 core เช่น เมื่อกำหนดขนาด domU คงที่ให้มีขนาด 1 core และ dom0 มีขนาดที่เปลี่ยนแปลง ให้มีขนาดระหว่าง 1 core ถึง 8 core (d01c-u1c d02c-u1c d03c-u1c d04c-u1c d05c-u1c d06c-u1c d07c-u1c และ d08c-u1c จะสังเกตได้ว่าเมื่อ dom0 มีขนาด 1 core ค่าของ sleeping time มีช่วงเวลาที่ใกล้เคียง 0 ms เมื่อเพิ่มขนาด CPU core เป็น 2 core จนถึง 7 core ค่า sleeping time จะค่อยๆ เพิ่มขึ้นและลดลงอีกครั้งเมื่อ CPU core มีขนาด 8 core ทำให้ได้กราฟที่มีลักษณะคล้ายระฆังคว่ำ ที่ค่า sleeping time ดีที่สุดเมื่อ CPU core ของ dom0 มีขนาด 1 core และ 8 core โดยสาเหตุที่ทำให้เกิดกราฟลักษณะคล้ายระฆังคว่ำในส่วนหัวของกราฟและส่วนท้ายของกราฟเป็นส่วนที่ประสิทธิของ logger ในระบบบันทึกเหตุการณ์มีประสิทธิภาพดีที่สุด สาเหตุดังกล่าวสามารถอธิบายได้ 2 ส่วน คือ 1) ส่วนของ CPU core ถูกกำหนดให้มีขนาด 1 core และ 2) ส่วนของ CPU core ถูกกำหนดให้มีขนาด 8 core โดยได้ข้อสรุปดังนี้

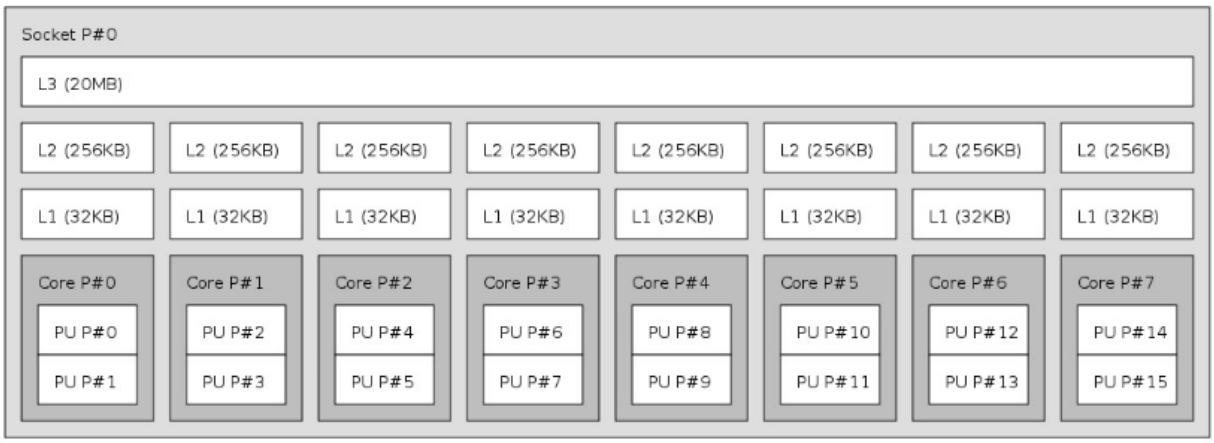
**1) ส่วนของ CPU core ถูกกำหนดให้มีขนาด 1 core**

ส่วนของ CPU core ถูกกำหนดให้มีขนาด 1 core มีค่า sleeping time ต่ำ หมายความว่าประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์มีค่าดีที่สุดซึ่งเกิดจากการทำงานของระบบปฏิบัติการที่มีการจัดสรรพื้นที่สำหรับโปรเซสต่าง ๆ ได้มีพื้นที่สำหรับการประมวลผลในการทำงานอย่างลงตัว โดย Avudaiyappan & Abdallah ได้อธิบายไว้ว่า ใน CPU core แต่ละตัวจะมีแคช[10] L1 และ L2 สำหรับการทำงานเฉพาะภายใน CPU core ของตัวเองและมีการแชร์แคช L3 เป็นแคชร่วมสำหรับการทำงานของทุก ๆ CPU core (L1, L2 และ L3 คือแคช) ดังภาพที่ 15

[10] แคช (cache) คือ หน่วยความจำชนิดหนึ่ง ซึ่งจะมีความเร็วในการเข้าถึงและการถ่ายโอนข้อมูลที่สูง โดยจะมีหน้าที่ในการเก็บ หรือพักข้อมูลที่มีการใช้งานบ่อย ๆ เพื่อเวลาที่ CPU ต้องการใช้ข้อมูลนั้น ๆ จะได้ค้นหาได้เร็ว โดยที่ไม่จำเป็นที่จะต้องไปค้นหาจากข้อมูลทั้งหมด (Cache, 15 Mar. 1999)

ด้านล่าง (กรอบสีเทาแต่ละอันเขียนว่า Core P#0 จนถึง Core P#7 ซึ่งเป็นจำนวน CPU core ทั้งหมด 8 core และภายในแต่ละ CPU core จะแบ่งเป็น 2 เทรด หรือ thread เช่น ในกล่องสีเทาล่างสุดที่เขียนว่า Core P#0 ภายในจะมีกล่องสี่เหลี่ยมสีขาว 2 กล่อง คือ PU #0 และ PU P#1 ซึ่งหมายถึงเทรดที่อยู่ภายใน CPU core นั้นๆ) (Avudaiyappan & Abdallah, 2014, 2017) จากที่ Avudaiyappan & Abdallah กล่าวไว้ข้างต้น ทำให้ผู้วิจัยเห็นว่าเมื่อมีการกำหนด CPU core เป็น 1 core โปรเซส logger ก็จะทำงานภายใน CPU core เดียวและมีการเรียกใช้แคชเพียง L1 และ L2 เท่านั้น ทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดี เพราะ CPU core จะสามารถโหลดข้อมูลจากแคชได้ทันทีโดยที่ไม่ต้องมีการสลับการทำงานไปยังแคชของ CPU core ตัวอื่นๆ แต่ถ้าหากมีการกำหนด CPU core เพิ่มจาก 1 core เป็น 2 core ไปจนถึง 8 core การจัดสรร CPU core โดยระบบปฏิบัติการจะเป็นตัวกำหนดให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนการประมวลผลแบบกลุ่มเมฆไปทำงานบน CPU core ใด CPU core หนึ่ง เมื่อทำงานเสร็จ 1 รอบ (การทำงานเสร็จ 1 รอบ คือ เมื่อรันโปรเซส logger 1 ครั้งและโปรเซส logger สามารถค้นหาโปรเซส read เจอและแสดงข้อมูลภายในของโปรเซส read ได้) และการทำงานรอบถัดๆ[11] ไปอาจจะมีการสลับไปใช้ CPU core อื่นๆ ทำให้ต้องการเสียเวลาในย้ายข้อมูลใน L1 และ L2 ของ CPU Core หนึ่งไป L1 และ L2 ของอีก CPU core ตัวอื่นๆ เพื่อประมวลอย่างต่อเนื่อง ซึ่งการย้ายข้อมูลจากแคชทำได้โดยทำผ่าน L3 ดังนั้นอาจทำให้ประสิทธิภาพของ logger ลดลง เมื่อมีการสลับการทำงานจาก CPU core หนึ่งไปยังอีก CPU core หนึ่ง ยกตัวอย่างเช่น เมื่อโปรเซส logger ทำงานบน CPU core 0 (ดูภาพที่ 15 บริเวณกล่องที่เขียนว่า Core P#0) เมื่อ logger ดังกล่าวทำงานเสร็จ 1 รอบ ขั้นตอนถัดไประบบปฏิบัติการจะเป็นตัวกำหนดให้โปรเซส logger ทำงานที่ CPU core เดิมหรืออาจจะมีการสลับไปทำงานยัง CPU core อื่นๆ หากภายใน CPU core เดิมที่โปรเซส logger เคยประมวลผลและมีเหตุการณ์ที่ระบบปฏิบัติการได้ทำการจองพื้นที่ให้กับโปรเซสอื่นที่ต้องการประมวลผลแทรกเข้ามาทำงานใน CPU core เดิมที่โปรเซส logger เคยใช้ประมวลผลทำให้โปรเซส logger ต้องย้ายจาก CPU core เดิมไปประมวลผลบน CPU core ใหม่ส่งผลให้ประสิทธิภาพของโปรเซส logger ลดลง เพราะโปรเซส logger ได้มีการประมวลผลบน CPU core ตัวใหม่ ทำให้ระบบปฏิบัติการต้องย้ายข้อมูลของโปรเซส logger จาก CPU core ตัวเดิมที่โปรเซส logger เคยประมวลผลย้ายข้อมูลของโปรเซส logger ไปยัง CPU core ใหม่เพื่อประมวลผลต่อจึงส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง

[11] ในการทำงานของ logger ไม่ได้รัน 1 รอบแล้วโปรเซสถูกสลับการทำงานระหว่าง CPU core ความจริง คือ มันอาจถูกแบ่งเป็น 2 โปรเซสแล้วทำงานคนละ CPU core กัน แต่ในการอธิบายในงานทดลองนี้ ขอสมมุตว่า เมื่อโปรเซส logger ทำงาน 1 รอบ โปรเซส logger จะรันบน CPU core เดียวกัน โดยที่โปรเซส logger ไม่ได้ถูกแยกให้ทำงานคนละ CPU core




**ภาพที่ 15 สถาปัตยกรรมของ CPU core Xeon ที่ถูกใช้ในการทดลอง**

**2) ส่วนของ CPU core ถูกกำหนดให้มีขนาด 8 core**

ส่วนของ CPU core ถูกกำหนดให้มีขนาด 8 core ส่งผลให้ประสิทธิภาพของ logger ในระบบบันทึกเหตุการณ์ดี เหตุผลอาจเนื่องมาจากเมื่อภายในเครื่องคอมพิวเตอร์มี CPU core จำนวน 8 core ถูกใช้เต็มจำนวน 8 core ระบบปฏิบัติการอาจจะมีการจัดสรรพื้นที่ของ CPU core ให้กับโปรเซสทั้งหมดในระบบได้เหมาะสมทำให้ขณะที่ระบบปฏิบัติการเรียกใช้จำนวน CPU core จำนวนมากขึ้นเมื่อเทียบกับกรณีที่ผู้วิจัยได้กำหนดให้ใช้ CPU core เพียงแค่ 1 core ดังนั้นทำให้การทำงานของ logger ในระบบบันทึกเหตุการณ์ ภายใต้การจัดการของระบบปฏิบัติการมีโอกาสสลับการทำงานของ logger ในระบบบันทึกเหตุการณ์จาก CPU core เดิมลดน้อยลงเมื่อมีการรันโปรเซส logger จำนวน 1000 รอบ โดยเมื่อ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ถูกรันครั้งแรกไปเป็นครั้งที่ 2 ครั้งที่ 3 จนถึงครั้งที่ 1000 รอบ มีโอกาสที่การทำงานของโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์จะถูกสลับการทำงานของโปรเซสบน CPU core เดิมกับ CPU core อื่น ๆ โอกาสสลับการทำงานจึงลดน้อยลงเนื่องจากว่าขนาดของ CPU core 8 core มีจำนวน CPU core ที่มากขึ้นทำให้โปรเซสอื่น ๆ กระจายการทำงานไปทุก ๆ CPU core ซึ่งเมื่อมีการรันโปรเซส logger เดิมซ้ำไปมา อาจไม่จำเป็นต้องย้ายโปรเซส logger ไปมาระหว่าง CPU core ซึ่งตรงกับทฤษฎีของระบบปฏิบัติการที่ อัจจิมา สุธีและพีรพรได้กล่าวไว้ (อัจจิมา เลี้ยงอยู่ สุธี พงศาสกุลชัย และ พีรพร หมุนสนิท, 2553) ดังนั้นในการทดลองในหัวข้อ 4.2.2.2 ส่วนของ CPU core ถูกกำหนดให้มีขนาด 8 core เหตุผลที่ทำให้โปรเซส logger ในระบบบันทึกเหตุการณ์มีประสิทธิภาพดีขึ้นเกิดจากโปรเซส logger มีโอกาสที่จะใช้ CPU core เดิมในการประมวลผลเพิ่มมากขึ้นทำให้ประสิทธิภาพของโปรเซส logger ในระบบบนทึกเหตุการณ์ดีที่สุดเพราะ sleeping time ต่ำสุด ดังแสดงไว้ในภาพที่ 11 และได้อธิบายไว้ในส่วนของย่อหน้าแรกของหัวข้อ 4.2.2 การปรับขนาดของ CPU core

ในฝั่งผู้ให้บริการผลการทดลองทำให้เกิดกราฟคล้ายระฆังคว่ำ ซึ่งเป็นจุดที่ดีที่สุดเมื่อ CPU core ของผู้ให้บริการหรือ dom0 มีขนาด 1 core และ 8 core

เพื่อให้ผู้อ่านเข้าใจมากขึ้นเกี่ยวกับการทำงานของระบบปฏิบัติการที่มีการสลับการทำงานของ CPU core จนส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง ผู้วิจัยจึงได้ใช้คำสั่ง top บนระบบปฏิบัติ linux เพื่อเช็คสถานะการทำงานของโปรเซสบน CPU core โดยแสดงดังภาพที่ 16 แสดงสถานการณ์ทำงานบนโปรเซสบน CPU core ของโปรเซส logger และแสดงให้เห็นว่าโปรเซส logger มีการสลับการทำงานระหว่าง CPU core

```
  P   PID USER      PR  NI    RES    SHR S %CPU  VIRT %MEM     TIME+ COMMAND
 10  7134 root      20   0   7952    988 R 95.3 29072  0.1   0:16.11 logger

  P   PID USER      PR  NI    RES    SHR S %CPU  VIRT %MEM     TIME+ COMMAND
  8  7134 root      20   0     17m   988 D 96.2 39512  0.3   0:42.12 logger
```

**ภาพที่ 16 สถานะการทำงานของโปรเซสบน CPU core จากคำสั่ง top**

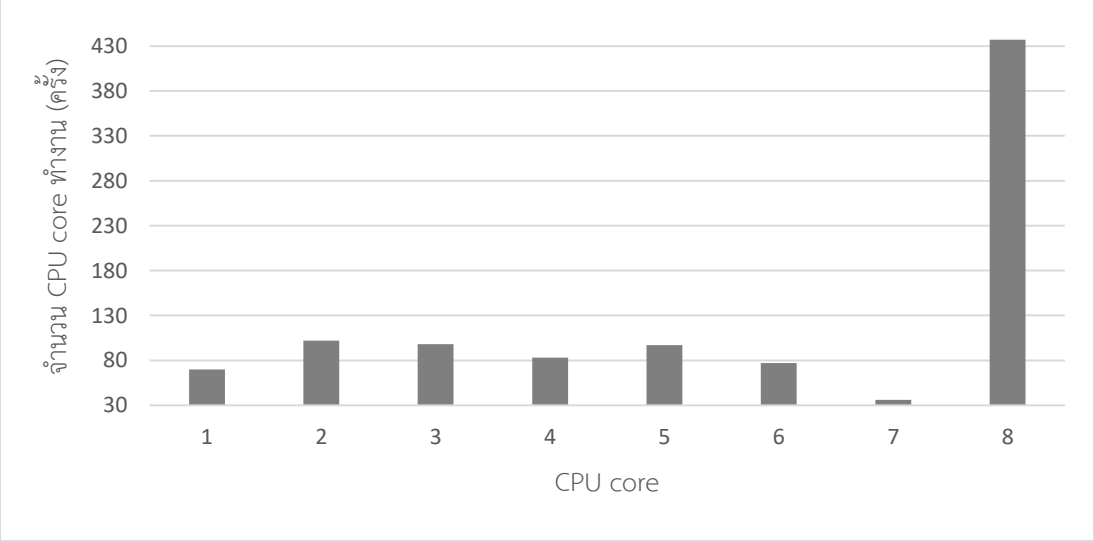
จากภาพที่ 16 จะแสดงสถานะทำงานของโปรเซสบน CPU core จากคำสั่ง top (Biswas, September, 2017) โดยผู้วิจัยจะอธิบายเฉพาะกรอบสีแดง 2 ส่วน โดยกรอบสีดำส่วนแรกสถานะของโปรเซสจะขึ้นด้วยตัว P ซึ่งหมายถึง CPU core ที่ถูกโปรเซสกำลังทำงานบน CPU core ลำดับนั้น ๆ และกรอบสีดำด้านหลังสุดช่อง COMMAND หมายถึงสถานะของโปรเซส logger ที่กำลังทำงานอยู่ โดยจะสังเกตได้ว่าเมื่อรันโปรเซส logger ในครั้งแรก โปรเซส logger จะทำงานบน CPU core ที่ 10 (P 10) (สาเหตุที่ขึ้น CPU core 10 แต่ CPU core ในเครื่องมีจำนวนสูงสุด 8 core เนื่องจาก ระบบปฏิบัติการ linux จะมองเห็น thread ใน CPU core 2 thread เป็น 1 core ดังที่ผู้วิจัยเคยอธิบายไว้ในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง ในหัวข้อที่ 2.9.1 หน่วยประมวลผลกลาง ในข้อที่ 2.9.2 cache หรือ แคช ดังนั้นเมื่อโปรเซส logger รันบน CPU 10 หมายความว่าโปรเซส logger กำลังทำงานบน CPU core ที่ 5 เมื่อโปรเซส logger ประมวลผลต่อเนื่องจาก CPU 10 ที่โปรเซส logger ได้ทำงาน โปรเซส logger ก็ถูกเปลี่ยนไปที่ CPU 8 (P 8) ในการโปรเซสครั้งต่อไป ซึ่งช่วงเวลาที่มีการสับเปลี่ยนของการทำงานของโปรเซส ทำให้ประสิทธิภาพของ logger ลดลงจึงส่งผลให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพลดลงตามไปด้วย

นอกจากนี้ผู้วิจัยยังได้ทำการตรวจสอบการทำงานของโปรเซส logger บนระบบปฏิบัติการ Fedora 16 เพื่อให้แน่ใจว่าเมื่อโปรเซส logger ได้ทำงานบน CPU core แล้วจะมีโอกาสที่โปรเซส logger จะทำงานบน CPU core ตัวเดิมซ้ำซึ่งผู้วิจัยได้ใช้วิธีการตรวจสอบคือ เพิ่มคำสั่ง Top สำหรับ

ดูโปรเซส logger ทำงานบน CPU core ตัวใดในส่วนของภาษา C โดยรันโปรเซส logger จับไฟล์ read 1000 ครั้งเช่นเดิมและให้แสดงข้อมูล P ที่อยู่ในคำสั่ง Top ออกทางหน้าจอดังภาพที่ 17


**ภาพที่ 17 สถานการณ์ทำงานของโปรเซส logger บน CPU core จากคำสั่ง top**

จากภาพที่ 17 จะเห็นว่าเมื่อโปรเซส logger รันการทำงานระบบปฏิบัติการจะทำการจองพื้นให้กับโปรเซส logger ได้ทำงานและมีโอกาสที่โปรเซส logger จะทำงานบน CPU core ตัวเดิมซ้ำดังตัวอย่างภาพที่ 17 ที่ logger ทำงานบน CPU core 8 ซ้ำมากกว่า CPU core อื่น ซึ่งจะแสดงผลการทดลองทั้งหมดดังกราฟ




**ภาพที่ 18 แสดงโปรเซส logger ประมวลผลบน CPU core**

จากภาพที่ 18 จะเห็นได้ว่าเมื่อโปรเซส logger ได้ทำงานทางระบบปฏิบัติการได้จองพื้นที่สำหรับการทำงานให้และ CPU core ที่ 8 core ก็มีการประมวลผลของโปรเซส logger มากที่สุดทำให้การทำงานของโปรเซส logger ทำงานซ้ำที่ core ที่ 8 ส่งผลให้ระบบบันทึกเหตุการณ์ในส่วนของ 8 core มีประสิทธิภาพดี

ในหัวข้อที่ 4.2 เรื่อง ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core ในหัวข้อนี้จะอธิบายเกี่ยวกับผลการทดลองที่ผู้วิจัยได้ทำการปรับขนากของ CPU core ทั้งในส่วนของผู้ให้บริการและผู้ใช้บริการจากนั้นบันทึกผลการทดลอง สรุปผลการทดลองและอธิบายเกี่ยวกับผลกระทบที่เกิดขึ้นโดยในส่วนของการทดลองแรกเมื่อมีการปรับขนาดของ CPU core ในฝั่งผู้ใช้บริการแล้วไม่ส่งผลกระทบใดๆ ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์และส่วนของการทดลองที่ 2 การปรับขนาดของ CPU core ในฝั่งผู้ให้บริการผลกระทบที่เกิดขึ้นคือส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ใน logger มีประสิทธิภาพลดลงโดยสาเหตุที่เกิดขึ้นได้ถูกอธิบายไว้ในหัวข้อที่ 4.2.2 การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการ

หลังจากได้ทำการทดลองเกี่ยวกับผลกระทบที่เกิดขึ้นทั้งหมดแล้ว ผู้วิจัยจะนำผลทดลองดังกล่าวไปวิเคราะห์หาสาเหตุเพื่อนำไปเป็นต้นแบบสำหรับออกแบบเพื่อปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ ซึ่งข้อมูลจากหัวข้อดังกล่าวผู้วิจัยได้นำไปใช้เป็นแนวทางในการปรับปรุงประสิทธิภาพโดยรายละเอียด วิธีการปรับปรุงประสิทธิภาพจะถูกนำเสนอในหัวข้อ 4.4 นำผลวิเคราะห์ที่ได้ไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และเปรียบเทียบผลการปรับปรุงประสิทธิภาพและหัวข้อถัดไปจะอธิบายเกี่ยวกับการวิเคราะห์ผลกระทบที่เกิดขึ้นกับระบบบันทึกเหตุการณ์โดยใช้หลักการทำงานของระบบปฏิบัติการมาช่วยในการวิเคราะห์

## 4.3 ผลการวิจัยเกี่ยวกับการวิเคราะห์ผลกระทบที่เกิดขึ้นกับระบบบันทึกเหตุการณ์โดยใช้หลักการทำงานของระบบปฏิบัติการมาช่วยในการวิเคราะห์

ภายในหัวข้อนี้จะอธิบายเกี่ยวกับผลกระทบที่เกิดขึ้นกับระบบบันทึกเหตุการณ์โดยสาเหตุของผลกระทบที่เกิดขึ้นได้ถูกกล่าวไว้ในหัวข้อที่ 4.1 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM และ 4.2 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core โดยสามารถแบ่งผลกระทบดังกล่าวได้ เป็น 2 ส่วน คือ 4.3.1 ผลวิเคราะห์ที่เกิดจากการปรับขนาดของ RAM 4.3.2 ผลวิเคราะห์ที่เกิดจากการปรับขนาดของ CPU core รายละเอียดมีดังนี้

### 4.3.1 ผลวิเคราะห์ที่เกิดจากการปรับขนาดของ RAM

ในหัวข้อส่วนที่ 1 จะอธิบายเกี่ยวกับผลการวิเคราะห์การปรับขนาดของ RAM และ CPU core ซึ่งรายละเอียดทั้งหมดถูกกล่าวไว้ในหัวข้อที่ 2.2 การออกแบบการทดลองและวิเคราะห์ผลการทดลองของบทที่ 3 วิธีการดำเนินวิจัย ซึ่งผลการวิเคราะห์ดังกล่าว ผู้วิจัยได้สรุปและนำมาเขียนในหัวข้อนี้เพื่อให้ผู้อ่านได้ทราบเกี่ยวกับผลกระทบที่เกิดขึ้น ตลอดจนการออกแบบวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และสรุปผลการทดลอง นอกจากนี้หัวข้อนี้ยังได้อธิบายเกี่ยวกับผลการทดลองใดที่ต้องมีการปรับปรุงประสิทธิภาพและผลการทดลองใดที่ไม่จำเป็นต้องมีการปรับปรุงประสิทธิภาพเนื่องด้วยสาเหตุใด โดยมีรายละเอียดดังนี้

#### 1) ผลวิเคราะห์จากการปรับขนาดของ RAM

ผลการวิเคราะห์จากการปรับขนาดของ RAM เป็นการสรุปเหตุผลจากหัวข้อ 4.1 เรื่อง ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM ในบทที่ 4 ผลการวิจัย เพื่อให้ผู้อ่านเข้าใจเกี่ยวกับผลกระทบดังกล่าว โดยผลการวิเคราะห์จากการปรับขนาดของ RAM สามารถแบ่งได้เป็น 2 ส่วน ดังนี้

#### 1.1) การปรับขนาดของ RAM ในส่วนของผู้ใช้บริการ

รายละเอียดเกี่ยวกับการปรับขนาดของ RAM ในส่วนของผู้ใช้บริการ มีรายละเอียดดังนี้

ส่วนของการปรับขนาดของ RAM ในส่วนของผู้ใช้บริการ ผลการทดลองที่ได้ คือ เมื่อผู้ใช้บริการมีการเพิ่มขนาดของ RAM จะทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงโดยรายละเอียดเกี่ยวกับผลกระทบดังกล่าว ผู้วิจัยได้อธิบายในหัวข้อที่ 4.1.1 การปรับขนาด RAM ในฝั่งของผู้ใช้บริการ ภายในส่วนนี้ผู้วิจัยได้ทำการปรับปรุงประสิทธิภาพโดยใช้เครื่องมือ taskset บนระบบปฏิบัติการ linux ซึ่งใช้เพื่อปรับปรุงประสิทธิภาพ ผลการทดลองที่ได้จะกล่าวในหัวข้อที่ 4.4.2 ผลของการปรับปรุงประสิทธิภาพของบทที่ 4 ผลการวิจัยต่อไป

**1.2) การปรับขนาดของ RAM ในส่วนของผู้ให้บริการ**

ในหัวข้อการปรับขนาดของ RAM ในส่วนของผู้ให้บริการผู้วิจัยได้อธิบายรายละเอียดดังนี้

ส่วนของการปรับขนาดของ RAM ในส่วนของผู้ให้บริการหากมีการปรับขนาดของ RAM ในส่วนของผู้ใช้บริการ ไม่ได้ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ ซึ่งผู้วิจัยได้อธิบายเกี่ยวกับสาเหตุดังกล่าวในบทที่ 4 ผลการวิจัย หัวข้อที่ 4.1.2 การปรับขนาด RAM ในฝั่งของผู้ให้บริการแล้ว ทำให้ไม่ต้องมีการปรับปรุงประสิทธิภาพในส่วนนี้

จากการทดลองปรับขนาดของ RAM ในส่วนของผู้ให้บริการและผู้ใช้บริการ ผลสรุปที่ได้ทำให้ทราบว่าการปรับขนาด RAM ในส่วนของผู้ให้บริการไม่จำเป็นต้องมีการปรับปรุงประสิทธิภาพ เนื่องจากว่าเมื่อมีการปรับขนาดของ RAM ในส่วนของผู้ให้บริการจะไม่ส่งผลกระทบใด ๆ ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ และการปรับขนาดของ RAM ในส่วนของผู้ใช้บริการจำเป็นต้องมีการปรับปรุงประสิทธิภาพเนื่องจากหากมีการเพิ่มขนาดของ RAM ในส่วนของผู้ให้บริการจะส่งผลให้ความแม่นยำของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์จะมีความแม่นยำลดลงเรื่อย ๆ ทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงจึงต้องมีการปรับปรุงประสิทธิภาพในส่วนนี้และผลการวิเคราะห์การปรับขนาดของ CPU core จะถูกกล่าวในหัวข้อถัดไป

### 4.3.2 ผลวิเคราะห์จากการปรับขนาดของ CPU core

ในหัวข้อส่วนที่ 2 จะอธิบายเกี่ยวกับผลการวิเคราะห์การปรับขนาดของ CPU core ที่ได้จากการทดลองซึ่งรายละเอียดของผลการทดลองได้ถูกกล่าวในหัวข้อที่ 4.2 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core โดยผู้วิจัยได้นำผลการทดลองดังกล่าวมาวิเคราะห์สาเหตุที่เกิดขึ้นและนำมาเขียนในหัวข้อนี้เพื่อให้ผู้อ่านได้ทราบเกี่ยวกับผลกระทบที่เกิดขึ้น ตลอดจนการออกแบบวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยมีรายละเอียดดังนี้

**2) ผลวิเคราะห์จากการปรับขนาดของ CPU core**

ผลการวิเคราะห์จากการปรับขนาดของ CPU core เป็นการสรุปเหตุผลจากหัวข้อ 4.2 เรื่อง ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core ในบทที่ 4 ผลการวิจัย เพื่อให้ผู้อ่านเข้าใจเกี่ยวกับผลกระทบดังกล่าวโดยผลการวิเคราะห์จากการปรับขนาดของ CPU core สามารถแบ่งได้เป็น 2 ส่วนดังนี้

**2.1) การปรับขนาดของ CPU core ในส่วนของผู้ใช้บริการ**

รายละเอียดเกี่ยวกับผลกระทบที่เกิดขึ้นในส่วนของการปรับขนาดของ CPU core ในส่วนของผู้ใช้บริการ ผู้วิจัยได้กล่าวรายละเอียดในย่อหน้าถัดไป

ส่วนของการปรับขนาดของ CPU core ในส่วนของผู้ใช้บริการหากมีการปรับขนาดของ CPU core ในส่วนของผู้ใช้บริการแล้วผลกระทบในส่วนนี้ไม่ได้ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ ทำให้ผู้ใช้บริการสามารถเลือกใช้ CPU core ได้ตามต้องการโดยไม่ต้องกังวลเกี่ยวกับผลกระทบที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์ ซึ่งสาเหตุของผลกระทบที่ทำให้การปรับขนาดของ CPU core ไม่ส่งผลต่อระบบบันทึกเหตุการณ์ ผู้วิจัยได้อธิบายในบทที่ 4 ผลการวิจัย หัวข้อที่ 4.2.1 การปรับขนาดของ CPU core ในฝั่งของผู้ใช้บริการ ทำให้ไม่ต้องมีการปรับปรุงประสิทธิภาพในส่วนนี้

**2.2) การปรับขนาดของ CPU core ในส่วนของผู้ให้บริการ**

ในหัวข้อการปรับขนาดของ CPU core ในส่วนของผู้ให้บริการจะถูกอธิบายรายละเอียดทั้งหมดในย่อหน้าถัดไป

ส่วนของการปรับขนาดของ CPU core ในส่วนของผู้ให้บริการ ผลกระทบที่เกิดขึ้น คือ เมื่อเพิ่มขนาดของ CPU core ในฝั่งของผู้ให้บริการแล้ว ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง โดยสาเหตุของผลกระทบเกิดจากการเพิ่มจำนวนของ CPU core ทำให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ประมวลผลโดยสลับการทำงานกลับไปกลับมาทุก ๆ CPU core ส่งผลให้ขณะที่โปรเซส logger ประมวลผลเสร็จใน CPU core หนึ่งและสับเปลี่ยนไปยัง CPU core อื่น ๆ ในช่วงของการเปลี่ยน CPU core สำหรับการโปรเซส จึงส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง โดยรายละเอียดเกี่ยวกับผลกระทบทั้งหมดของการปรับขนาด CPU core ในส่วนของผู้ให้บริการถูกกล่าวในหัวข้อที่ 4.2.2 การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการ ของบทที่ 4 ผลการวิจัย

จากผลการวิเคราะห์เกี่ยวกับผลกระทบที่เกิดขึ้นกับการปรับขนาดของ CPU core ทั้ง 2 ประเด็น ทำให้ทราบว่าการปรับขนาดของ CPU core ในส่วนของผู้ใช้บริการไม่ได้ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ทำให้ไม่ต้องมีการปรับปรุงประสิทธิภาพในส่วนของผู้ใช้บริการ นอกจากนี้ในส่วนของการปรับขนาด CPU core ในส่วนของผู้ให้บริการผลกระทบที่เกิดขึ้นทำให้ทราบว่าการเพิ่มขนาดของ CPU core จะส่งผลกระทบต่อการทำงานของโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์และส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์

จากการทดลองและผลการวิเคราะห์ที่ได้ ในบทที่ 3 วิธีการดำเนินการวิจัย ส่วนที่ผู้วิจัยจะทำการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์เพื่อให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดียิ่งขึ้นตามวัตถุประสงค์ของการวิจัย ผู้วิจัยจึงได้ออกแบบการปรับปรุงประสิทธิภาพให้กับการทดลอง 2 ส่วน คือ 1. การปรับขนาดของ RAM ในส่วนของผู้ใช้บริการ และ 2. การปรับขนาดของ CPU core ในส่วนของผู้ให้บริการ โดยวิธีการทดลองและผลการปรับปรุงประสิทธิภาพจะกล่าวใน ส่วนที่ 2 นำผลวิเคราะห์ที่ได้ไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และเปรียบเทียบผลการปรับปรุงประสิทธิภาพ หัวข้อถัดไป

## 4.4 ผลการวิจัยเกี่ยวกับการนำผลการวิเคราะห์ที่ได้ไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และเปรียบเทียบผลการปรับปรุงประสิทธิภาพ

ในส่วนที่ 2 นี้ผู้วิจัยได้แบ่งการปรับปรุงประสิทธิภาพและนำผลการทดลองที่ได้ไปเปรียบเทียบกับผลการทดลองเดิมแบ่งเป็น 3 หัวข้อ คือ 4.4.1 วิธีการปรับปรุงประสิทธิภาพ 4.4.2 ผลของการปรับปรุงประสิทธิภาพ และ 4.4.3 เปรียบเทียบผลการปรับปรุงประสิทธิภาพโดยมีรายละเอียดดังนี้

### 4.4.1 วิธีการปรับปรุงประสิทธิภาพ

ในหัวข้อนี้ผู้วิจัยจะอธิบายเกี่ยวกับวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ด้วยเครื่องมือ taskset ใน linux ซึ่งรายละเอียดของเครื่องมือ taskset มีการอธิบายไว้ในหัวข้อ 3.4 วิธีการทำงานของเครื่องมือ taskset และสาเหตุที่ผู้วิจัยเลือกใช้เครื่องมือ taskset และข้อกำหนดเงื่อนไขต่าง ๆ ที่ใช้ในการทดลองโดยมีรายละเอียดดังนี้

#### 1) เครื่องมือ taskset ใน linux

ในหัวข้อนี้จะอธิบายเกี่ยวกับเหตุผลของการใช้เครื่องมือ taskset ใน linux ลักษณะการทำงานของเครื่องมือดังกล่าว โดยรายละเอียดมีดังนี้

สาเหตุที่ใช้เครื่องมือ taskset ใน linux เพื่อปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ เพราะว่าจากการวิเคราะห์ของผลกระทบที่เกิดขึ้นซึ่งถูกอธิบายในบทที่ 3 ในหัวข้อที่ 3.3 วิธีออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพ ทำให้ทราบว่าผลกระทบที่เกิดขึ้นส่วนใหญ่เกิดจากการทำงานของโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ถูกระบบปฏิบัติการกำหนดพื้นที่ให้โปรเซส logger ทำงาน โดยลักษณะการทำงานของโปรเซส logger จะสลับกระบวนการโปรเซสไปมาแต่ละ CPU core ส่งผลให้ยิ่งมีการเพิ่มขนาดของ CPU core ยิ่งทำให้โปรเซส logger มีโอกาสสลับการทำงานมากขึ้น จนส่งผลกระทบให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง ดังนั้นเมื่อ

วิเคราะห์เกี่ยวกับผลกระทบดังกล่าวแล้วทำให้ทราบว่าต้องมีการกำหนด CPU core สำหรับการทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์โดยเป็นการกำหนดจัดสรรพื้นที่ให้โปรเซส logger ได้ทำงานจากผู้ทดลองกำหนดเอง โดยเครื่องมือที่ใช้กำหนดพื้นที่สำหรับการทำงานให้กับ logger เรียกว่า taskset ซึ่งเป็นเครื่องมือที่ใช้ในระบบปฏิบัติการ linux โดยเครื่องมือ taskset ดังกล่าวเป็นเครื่องมือสำหรับกำหนดพื้นที่การทำงานให้กับโปรเซส เช่น กำหนดขนาดพื้นที่ของ CPU core หรือการกำหนดขนาดพื้นที่ของ RAM โดยคำสั่งที่ผู้วิจัยใช้ในการทดลองแสดงดังภาพที่ 19 (A.Saha, 2006)

```
[root@localhost logger_tester_new]# taskset -c 6-7 ./logger4
```

**ภาพที่ 19 การใช้งานของคำสั่ง taskset โดยกำหนดให้โปรเซส logger ทำงานที่ CPU core ที่ 4**

จากภาพที่ 19 การใช้งานของคำสั่ง taskset สาเหตุที่ต้องมีการกำหนด CPU core 6-7 ให้กับโปรเซส logger ประมวลผล (โปรเซส logger ผู้วิจัยได้ใช้ชื่อโปรเซส logger ในการทดลองว่า logger4) เพราะว่า จากสถาปัตยกรรมของ CPU core ที่ใช้ในการทดลองภาพที่ 9 บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง หัวข้อที่ 2.9.1 หน่วยประมวลผลกลาง ข้อที่ 2.9.2 Cache หรือ แคช จะสังเกตได้ว่า ระบบปฏิบัติการ linux จะมองเห็นการทำงาน 2 thread เป็น 1 CPU core ดังนั้นการกำหนดให้โปรเซส logger ประมวลผลบน CPU core เดียว จำเป็นต้องกำหนด 2 thread เพื่อให้เป็น 1 CPU core และสาเหตุที่ผู้วิจัยได้เลือกการประมวลผลให้โปรเซส logger ทำงานบน CPU core ที่ 4 เพื่อให้ทราบว่าการกำหนด CPU core ใดก็ตามจะไม่ส่งผลกระทบใดๆ ต่อประสิทธิภาพที่เกิดขึ้น หากมีการเปลี่ยน CPU core ในการทำงาน เช่น จาก CPU core ที่ 4 ไป CPU core อื่น ๆ ประสิทธิภาพก็ยังคงเดิม

หลังจากผู้วิจัยเลือกเครื่องมือสำหรับการปรับปรุงประสิทธิภาพในการทดลองได้แล้ว ผู้วิจัยจะกำหนดเงื่อนไขต่าง ๆ ในการทดลองโดยจะกล่าวใน หัวข้อ 4.4.1.2 เงื่อนไขในการทดลอง หัวข้อถัดไป

**2) เงื่อนไขในการทดลอง**

ในหัวข้อเงื่อนไขในการทดลองนี้จะอธิบายเกี่ยวกับเงื่อนไขของการทดลองที่ผู้วิจัยได้ใช้ในการทดลอง การกำหนดค่า sleeping time เงื่อนไขการกำหนดฮาร์ดแวร์ต่าง ๆ ในการทดลอง โดยมีรายละเอียดดังนี้

การทดลองหลังการปรับปรุงประสิทธิภาพได้ใช้เครื่องมือ taskset ในระบบปฏิบัติการ linux ผู้วิจัยจะใช้วิธีการทดลองเหมือนเดิมกับในบทที่ 3 วิธีการดำเนินวิจัย หัวข้อที่ 3.3 วิธีออกแบบการทดลองเพื่อปรับปรุงประสิทธิภาพ เหตุผลเพื่อให้ลักษณะการทำงานของการทดลองเหมือนเดิมและสามารถเปรียบเทียบประสิทธิภาพที่เกิดขึ้นได้โดยสิ่งแรกที่ผู้วิจัยได้กำหนด คือ ค่าของ sleeping time (ค่าของ sleeping time ถูกกล่าวในหัวข้อ 2.8.1 sleeping time และ accuracy ของบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง) โดยค่า sleeping time ที่ผู้วิจัยได้กำหนดในการทดลองอยู่ที่ช่วงเวลา 65 ms. โดยนำค่า sleeping time ดังกล่าวมาเป็นเกณฑ์เพื่อเปรียบเทียบประสิทธิภาพในการทดลอง ซึ่งค่า sleeping time ดังกล่าวผู้วิจัยได้ศึกษางานของ Wongthai (Wongthai & Moorsel, 2016) โดยในงานวิจัยดังกล่าวได้มีการทดลองวัดประสิทธิภาพความแม่นยำของ logger ในระบบบันทึกเหตุการณ์ ซึ่งประสิทธิภาพที่ logger สามารถตรวจสอบได้ 100 % คือ ช่วงเวลาของ sleeping time ที่ 65 ms ดังนั้นผู้วิจัยจึงมุ่งหวังว่าการปรับปรุงประสิทธิภาพในงานวิทยานิพนธ์เล่มนี้จะต้องมีค่าของ sleeping time ที่ลดลงกว่า 65 ms เนื่องจากยิ่งค่า sleeping time ลดลงจะยิ่งทำให้โปรเซส logger สามารถตรวจสอบโปรเซสได้ดียิ่งขึ้น ส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้นตามลำดับ โดยผู้วิจัยได้กำหนดฮาร์ดแวร์สำหรับการทดลอง ดังนี้

1) ฮาร์ดแวร์ที่ใช้ในการทดลอง

ด้านฮาร์ดแวร์ที่ผู้วิจัยได้กำหนดในการทดลอง คือ

- CPU Intel® Xeon® Processor ขนาด 8 core
- RAM DDR 3 ขนาด 8 GB.
- Hard Disk ขนาด 700 GB.

จากข้อมูลฮาร์ดแวร์ทั้งหมดที่ผู้วิจัยได้ระบุไว้ข้างต้น ผู้วิจัยได้ทำการทดลองบนฮาร์ดแวร์ดังกล่าวโดยใช้ประสิทธิภาพของฮาร์ดแวร์สูงสุดเท่านั้น ไม่ได้มีการปรับลดขนาดลงเหมือนการทดลองเดิมก่อนการปรับปรุงประสิทธิภาพ เนื่องจากการทดลองเดิมผู้วิจัยต้องการเห็นผลกระทบในสภาพแวดล้อมที่หลากหลายของการทดลองและเมื่อทราบผลกระทบแล้ว การทดลองหลังมีการปรับปรุงประสิทธิภาพเพื่อเปรียบเทียบในบทที่ 4 ผลการวิจัยจึงได้ใช้ฮาร์ดแวร์เพียงขนาดเดียวเพื่อเปรียบเทียบ เพราะการเปรียบเทียบประสิทธิภาพเพียงขนาดเดียวก็เพียงพอต่อการสรุปผลหลังการปรับปรุงประสิทธิภาพได้ หลังจากได้กำหนดฮาร์ดแวร์สำหรับการทดลองแล้ว ผู้วิจัยจะเริ่มทำการ

ทดลองและเก็บผลการปรับปรุงประสิทธิภาพโดยจะกล่าวในหัวข้อที่ 4.4.2 ผลของการปรับปรุงประสิทธิภาพหัวข้อถัดไป

### 4.4.2 ผลของการปรับปรุงประสิทธิภาพ

ในหัวข้อผลของการปรับปรุงประสิทธิภาพ จะอธิบายเกี่ยวกับวิธีการทดลอง วิธีการปรับปรุงประสิทธิภาพและผลการปรับปรุงประสิทธิภาพโดยรายละเอียดจะอธิบายในย่อหน้าถัดไป

ก่อนจะอธิบายเกี่ยวกับผลของการปรับปรุงประสิทธิภาพ ผู้วิจัยขออธิบายเกี่ยวกับวิธีการทดลองและวิธีการปรับปรุงประสิทธิภาพเพื่อให้ผู้อ่านได้เข้าใจเกี่ยวกับผลของการปรับปรุงประสิทธิภาพก่อน โดยในการทดลองของวิทยานิพนธ์เล่มนี้ผู้วิจัยได้กำหนดเงื่อนไขเกี่ยวกับการทดลองไว้ในหัวข้อ 4.4.1.2 เงื่อนไขในการทดลองของหัวข้อที่ผ่านมา และนอกจากเงื่อนไขทั้งหมด ที่ผู้วิจัยได้กำหนดไว้โดยผู้วิจัยยังได้สร้างไฟล์ขนาดต่าง ๆ เพื่อใช้เปรียบเทียบในการทดลองระหว่างระบบบันทึกเหตุการณ์ที่ยังไม่มีการปรับปรุงประสิทธิภาพและระบบบันทึกเหตุการณ์หลังมีการปรับปรุงประสิทธิภาพและไฟล์ที่ผู้วิจัยได้สร้างไฟล์ขึ้นโดยตั้งชื่อว่า t3.txt ภายในแต่ละไฟล์มีขนาดที่แตกต่างกัน ดังนี้ 300,000 byte 200,000 byte 100,000 byte 50,000 byte 25,000 byte และ 20,000 byte สาเหตุของการสร้างไฟล์หลายๆ ขนาด เพื่อต้องการทดสอบว่า โปรเซส logger ในระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและหลังการปรับปรุงประสิทธิภาพจะสามารถตรวจสอบจับไฟล์ขนาดได้เล็กที่สุดที่ขนาดเท่าไร เพราะยิ่งไฟล์ที่มีขนาดเล็กจะส่งผลให้ โปรเซส logger ในระบบบันทึกเหตุการณ์ในคลาวด์ไม่สามารถตรวจจับไฟล์ได้ทันเพราะการโปรเซสเพื่ออ่านไฟล์ที่มีขนาดเล็กจะสามารถโปรเซสได้ไวส่งผลให้โปรเซส logger อาจจะเก็บข้อมูลที่อ่านไม่ทัน ซึ่งในการทดลองผู้วิจัยได้ทดสอบด้วยการใช้คำสั่ง cat ในระบบปฏิบัติการ linux โดยคำสั่ง cat เป็นคำสั่งสำหรับใช้อ่านไฟล์ ผู้วิจัยจึงได้ใช้คำสั่ง cat ดังกล่าวเพื่อเข้าไปอ่านไฟล์ที่ต้องการนั่นคือ ไฟล์ t3.txt โดยสามารถพิมพ์คำสั่งได้ดังนี้ cat t3.txt ดังภาพที่ 20

```
[root@localhost test]# cat t3.txt
```

**ภาพที่ 20 วิธีการอ่านไฟล์ t3.txt ด้วยคำสั่ง cat**

เมื่อรันคำสั่ง cat เพื่ออ่านไฟล์ t3.txt แล้ว โปรเซส logger สามารถตรวจจับข้อมูลที่คำสั่ง cat อ่านได้จะแสดงได้ดังภาพที่ 21

```
[root@localhost logger_tester]# ./logger4
t3.txt,PId: 3100,PName:cat,OId:1000
```

**ภาพที่ 21 ข้อมูลที่โปรเซส logger สามารถบันทึกได้จากการโปรเซสของคำสั่ง cat**

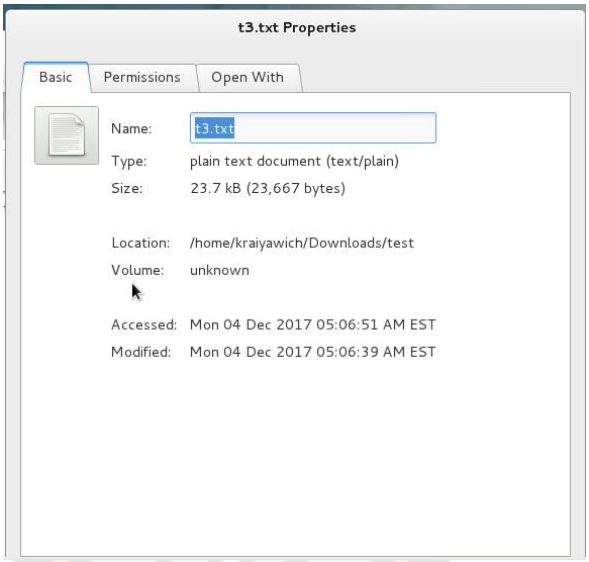
ข้อมูลของคำสั่ง cat ที่โปรเซส logger สามารถบันทึกได้ จะอธิบายดังตาราง ดังนี้

| **โปรเซสที่คำสั่ง cat อ่านได้** | **ข้อมูลภายในโปรเซส** |
|---|---|
| t3.txt (ชื่อไฟล์ของผู้ใช้บริการ) | t3.txt |
| PId (ไอดีของโปรเซส) | 3100 |
| PName (ชื่อของโปรเซสที่รัน) | Cat |
| OId (ไอดีของผู้ให้บริการ) | 1000 |

**ตารางที่ 5 ข้อมูลที่โปรเซส logger สามารถบันทึกได้ขณะที่คำสั่ง cat กำลังอ่านไฟล์ t3.txt**

ในการทดลองผู้วิจัยจะทดลองให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ได้ตรวจสอบจับคำสั่ง cat ขณะที่กำลังโปรเซสอ่านไฟล์ t3.txt จำนวน 10 ครั้ง และต้องสามารถเก็บข้อมูลที่อยู่ให้ t3.txt ได้ครบทั้ง 10 ครั้ง โดยความแตกต่างของขนาดไฟล์ขนาดต่าง ๆ จะเป็นสิ่งที่ใช้วัดประสิทธิภาพ เพราะ โปรเซส logger ในระบบบันทึกเหตุการณ์ที่ผ่านการปรับปรุงประสิทธิภาพแล้ว จะต้องสามารถตรวจสอบและจับไฟล์ที่มีขนาดเล็กลงได้ เนื่องจากยิ่งไฟล์มีขนาดเล็กลงจะยิ่งทำให้การโปรเซสเพื่ออ่านไฟล์ทำได้ไวขึ้นซึ่งเป็นช่องโหว่ของระบบบันทึกเหตุการณ์แบบ monitoring software ตามรายงานของ Wang, Gary (Wang, et al., 2015)

ทำให้ความแม่นยำลดลงหากซอฟต์แวร์มีการโปรเซสที่ไวขึ้นเพราะระบบบันทึกเหตุการณ์ไม่สามารถจัดเก็บข้อมูลดังกล่าวได้ทันโดยตัวอย่างขนาดของไฟล์ t3.txt ที่ใช้ในการทดลองสามารถแสดงในภาพที่ 22




**ภาพที่ 22 ตัวอย่างขนาดไฟล์ของ t3.txt ที่ใช้ในการทดลอง**

หลังจากผู้วิจัยได้สร้างไฟล์ขนาดต่าง ๆ ตามเงื่อนไขที่กล่าวในย่อหน้าข้างต้นแล้ว ผู้วิจัยจึงได้เริ่มการทดลอง โดยเริ่มจากระบบบันทึกเหตุการณ์ที่ยังไม่มีการปรับปรุงประสิทธิภาพ ซึ่งไฟล์ขนาดเล็กที่สุดที่ระบบบันทึกเหตุการณ์แบบยังไม่มีการปรับปรุงประสิทธิภาพสามารถจับไฟล์ได้ขนาดประมาณ 26,952 bytes และเมื่อใช้คำสั่ง taskset เพื่อปรับปรุงประสิทธิภาพในระบบบันทึกเหตุการณ์ ผลปรากฏว่าไฟล์ที่ระบบบันทึกเหตุการณ์สามารถตรวจจับได้มีขนาดประมาณ 23,667 bytes โดยมีขนาดเล็กลงถึง 3,285 bytes หรือคิดเป็น 32.85% นอกจากนี้ค่า sleeping time จากงานวิจัยของ Wongthai (Wongthai & Moorsel, 2016) ได้ทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์มีค่า sleeping time เวลาดีที่สุด คือ 65 ms. และผลของการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์มีค่า sleeping time เวลาที่ดีที่สุด คือ 60 ms. ซึ่งช่วงเวลาของ sleeping time ที่ต่างกัน 5 ms. ทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ที่ผ่านการปรับปรุงประสิทธิภาพสามารถจับไฟล์ได้เล็กลงถึง 32.85%

จากผลการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้นจากค่า sleeping time เวลา 65 ms. สามารถลดลงได้ถึง 60 ms. ยิ่งระบบบันทึกเหตุการณ์มีค่า sleeping time ใกล้ 0 จะทำให้ประสิทธิภาพยิ่งดีขึ้น โดยค่า sleeping time 65 ms. ระบบบันทึกเหตุการณ์จะสามารถจับไฟล์ขนาดเล็กที่สุดได้ ขนาดที่ 26,952 bytes และเมื่อปรับปรุงประสิทธิภาพแล้วช่วงเวลาของ sleeping time ที่ได้คือ 60 ms. ซึ่งช่วงเวลาของ sleeping time 60 ms. ระบบบันทึกเหตุการณ์จะสามารถจับไฟล์ได้ขนาดเล็กที่สุดขนาด 23,667 byte ซึ่งมีขนาดเล็กลงถึง 32.85% โดยสามารถอธิบายเป็นกราฟเปรียบเทียบในหัวข้อถัดไป

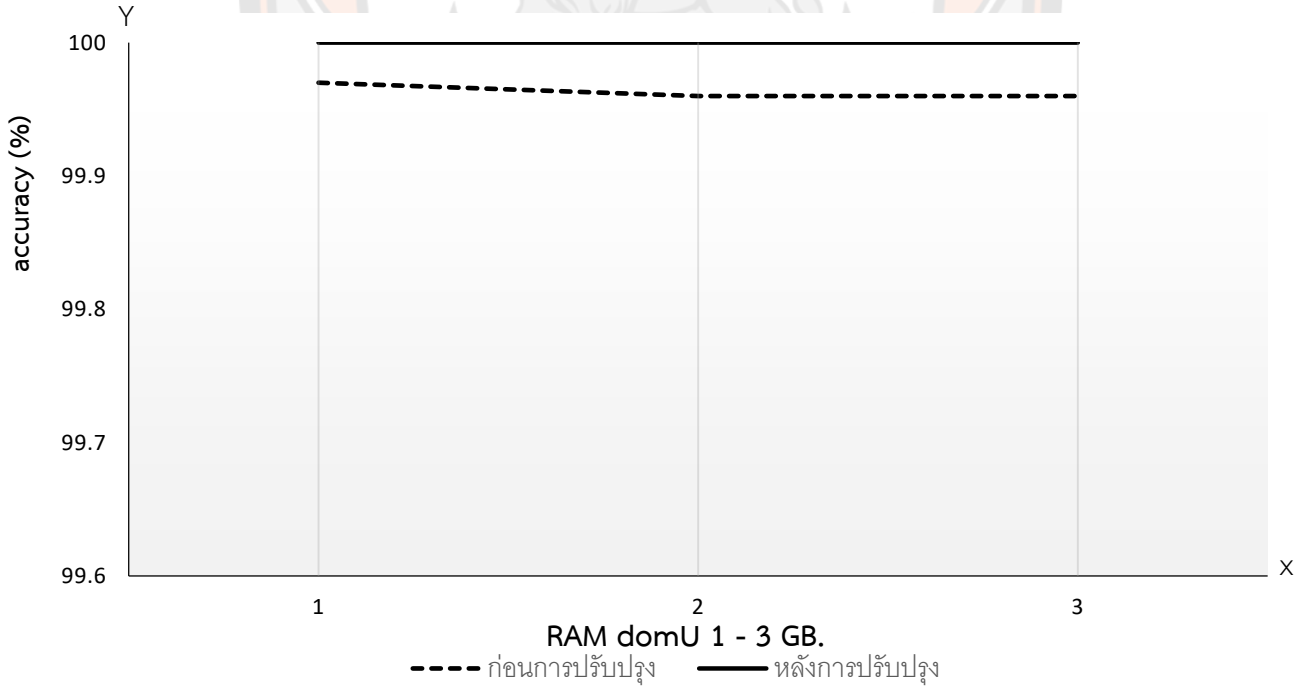
### 4.4.3 เปรียบเทียบผลการปรับปรุงประสิทธิภาพ

จากข้อสรุปในหัวข้อที่ 4.4.2 ผลของการปรับปรุงประสิทธิภาพได้อธิบายว่าผลการทดลองของการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์จะถูกสร้างเป็นกราฟเปรียบเทียบโดยผู้วิจัยจะกล่าวในหัวข้อนี้และจะอธิบายเกี่ยวกับกราฟในย่อหน้าถัดไป

จากผลการทดลองที่ได้ ระบบบันทึกเหตุการณ์ที่ผ่านการปรับปรุงประสิทธิภาพสามารถตรวจจับไฟล์ที่มีขนาดเล็กลงได้ถึง 32.85% และสามารถลดระยะเวลาของ sleeping time ซึ่งผลการปรับปรุงดังกล่าวจะสามารถสร้างกราฟเปรียบเทียบได้ 2 ส่วน คือ ส่วนที่ 1 ส่วนของการปรับปรุงประสิทธิภาพของ RAM ฝั่งผู้ใช้บริการซึ่งจะถูกกล่าวในหัวข้อ 4.4.3.1 ส่วนของการปรับปรุงประสิทธิภาพของ RAM และส่วนที่ 2 คือ ส่วนของการปรับปรุงประสิทธิภาพของ CPU core ในส่วนของผู้ให้บริการโดยจะถูกกล่าวในหัวข้อ 4.4.3.2 ส่วนของการปรับปรุงประสิทธิภาพส่วนของ CPU core ซึ่งจะมีรายละเอียด ดังนี้

**1) ส่วนของการปรับปรุงประสิทธิภาพของ RAM ในส่วนของผู้ใช้บริการ**

ส่วนของการปรับปรุงประสิทธิภาพของ RAM ในส่วนของผู้ใช้บริการสามารถอธิบายรายละเอียดดังกราฟในภาพที่ 23



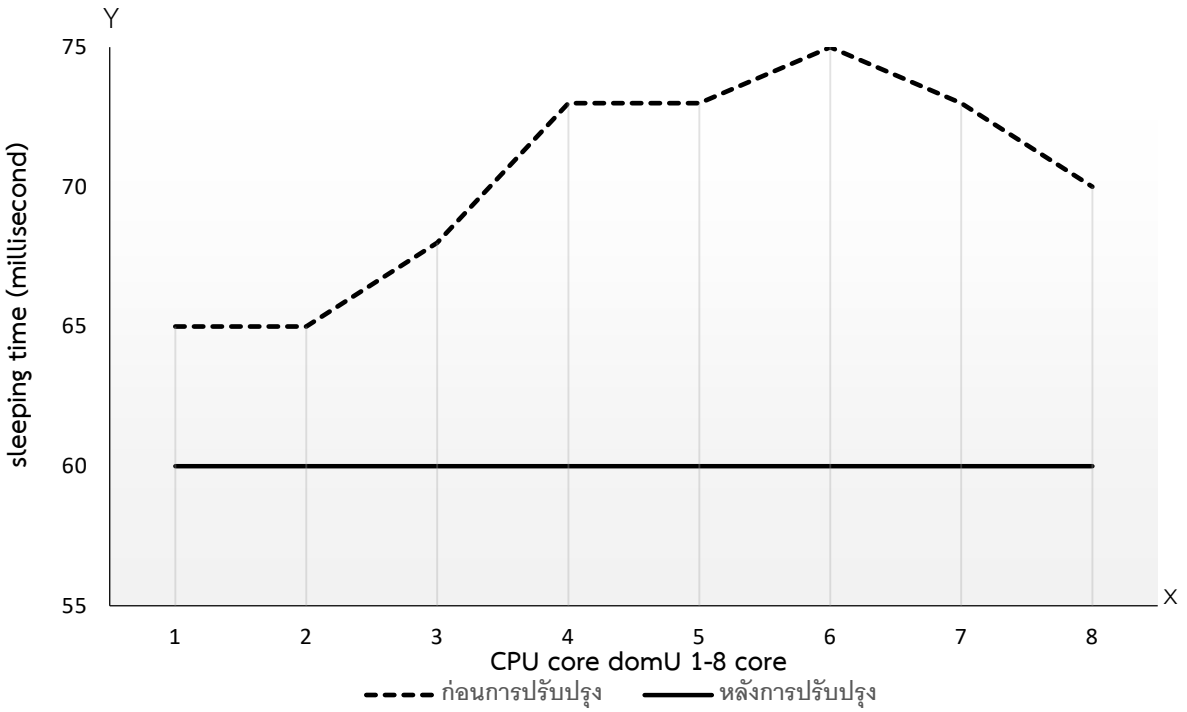

**ภาพที่ 23 การเปรียบเทียบระหว่างระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและระบบบันทึกเหตุการณ์หลังการปรับปรุงประสิทธิภาพในส่วนของ RAM**

จากภาพที่ 23 ผู้วิจัยได้ใช้กราฟเส้นเพื่อแสดงข้อมูลเปรียบเทียบเกี่ยวกับประสิทธิภาพระหว่างระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและระบบบันทึกเหตุการณ์หลังการปรับปรุงประสิทธิภาพ โดยแกน X จะแสดงจำนวนของ RAM ในฝั่งของผู้ใช้บริการ แกน Y แสดงประสิทธิภาพความแม่นยำที่ระบบบันทึกเหตุการณ์สามารถตรวจจับได้ โดยผลการทดลองของระบบบันทึกเหตุการณ์เดิม ถ้าหากมีการเพิ่มจำนวน RAM ในฝั่งของผู้ให้บริการความแม่นยำจะลดลง โดยสาเหตุเกิดจากพื้นที่ของ RAM ที่มีขนาดเพิ่มมากขึ้นทำให้ระบบบันทึกเหตุการณ์ต้องใช้เวลาในการค้นหาโปรเซสมากขึ้นและด้วยวิธีการทำงานแบบ monitoring software ต้องมีการหยุดการทำงานของโปรเซสเพื่อเก็บข้อมูลทำให้เมื่อโปรเซส logger เข้าไปขณะโปรเซสหยุดการทำงานบนพื้นที่ที่เพิ่มมากขึ้นของ RAM ซึ่งจะสร้างภาระในการค้นหาของโปรเซส logger มากขึ้น จึงส่งผลให้ประสิทธิภาพลดลงเพราะต้องค้นหาโปรเซสในพื้นที่ที่เพิ่มมากขึ้น โดยรายละเอียดเกี่ยวกับสาเหตุดังกล่าวผู้วิจัยได้กล่าวไว้ในบทที่ 4 ผลการวิจัยหัวข้อที่ 4.4.1 การปรับขนาด RAM ในฝั่งของผู้ใช้บริการ เมื่อผู้วิจัยได้ใช้วิธีกำหนดพื้นที่สำหรับ CPU core โดยใช้เครื่องมือ taskset แล้ว ส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น สามารถตรวจจับไฟล์ได้ขนาดเล็กลงและสามารถตรวจจับโปรเซสได้เร็วขึ้นทำให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์สามารถที่จะตรวจจับไฟล์ได้ดียิ่งขึ้น และเมื่อเพิ่มขนาดของ RAM ในส่วนของผู้ให้บริการ ก็ยังสามารถตรวจจับโปรเซสได้ 100% เหมือนเดิม ซึ่งถ้าเป็นระบบบันทึกเหตุการณ์แบบเดิมที่ไม่ได้กำหนดพื้นที่ให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ประมวลผลในการทำงาน ความแม่นยำในการตรวจสอบจับโปรเซสของ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ ความแม่นยำจะตกลงทุกครั้งที่ได้ทำการเพิ่มขนาดของ RAM ในฝั่งของผู้ใช้บริการ โดยสามารถสังเกตการณ์เปลี่ยนแปลงหลังการปรับปรุงประสิทธิภาพได้จากภาพที่ 21

จากการใช้วิธีกำหนดพื้นที่สำหรับการทำงานให้โปรเซส logger ในระบบบันทึเหตุการณ์บนคลาวด์ด้วยเครื่องมือ taskset ใน linux แล้ว ทำให้สามารถเพิ่มประสิทธิภาพความแม่นยำของโปรเซส logger ดังกล่าวได้จากเดิมเมื่อเพิ่มขนาดของ RAM ในฝั่งผู้ใช้บริการแล้วความแม่นยำจะลดลง แต่เมื่อมีการกำหนดพื้นที่ดังกล่าวให้โปรเซส logger ได้ทำงาน ส่งผลให้ค่าความแม่นยำคงที่ที่ค่า 100% ไม่ว่าจะมีการเพิ่มขนาดของ RAM ฝั่งผู้ใช้บริการไปขนาดเป็นจำนวนเท่าไรนอกจากนี้วิธีการกำหนดพื้นที่สำหรับโปรเซส logger ได้ทำงาน ยังสามารถนำไปใช้ปรับปรุงในส่วนของการเพิ่มขนาดของ CPU core ในส่วนของผู้ให้บริการได้อีกด้วย ซึ่งการเพิ่มขนาดของ CPU core ในส่วนของผู้ให้บริการถ้ามีการเพิ่มขนาดดังกล่าวแล้วประสิทธิภาพจะไม่คงที่มีทั้งเพิ่มขึ้นและลดลงคล้ายระฆังคว่ำ โดยผู้วิจัยได้อธิบายไว้แล้วในบทที่ 4 ในหัวข้อที่ 4.2.2 การปรับขนาด CPU core ในฝั่งของผู้ให้บริการ ซึ่งผลการปรับปรุงประสิทธิภาพของ CPU core จะกล่าวในหัวข้อถัดไป

**2) ส่วนของการปรับปรุงประสิทธิภาพของ CPU core ในส่วนของผู้ให้บริการ**

ส่วนของการปรับปรุงประสิทธิภาพของ CPU core ในส่วนของผู้ให้บริการ มีรายละเอียดดังภาพที่ 24




**ภาพที่ 24 การเปรียบเทียบระหว่างระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพและระบบบันทึกเหตุการณ์หลังการปรับปรุงประสิทธิภาพในส่วนของ CPU core**

จากภาพที่ 24 ได้แสดงการเปรียบเทียบประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยในภาพจะแสดงการเปรียบเทียบระหว่างระบบบันทึกเหตุการณ์การก่อนการปรับปรุงประสิทธิภาพและระบบบันทึกเหตุการณ์หลังการปรับปรุงประสิทธิภาพ โดยกราฟเส้นปะสีดำแทนค่าของประสิทธิภาพก่อนการปรับปรุงประสิทธิภาพและเส้นทึบสีดำแทนประสิทธิภาพของระบบบันทึกเหตุการณ์ หลังการปรับปรุงประสิทธิภาพ โดยกราฟเส้นทั้ง 2 จะถูกเปรียบเทียบขณะที่ CPU core ของผู้ให้บริการมีขนาดคงที่ขนาด 8 core และ CPU core ของผู้ใช้บริการมีขนาดที่เปลี่ยนแปลงตั้งแต่ 1 core – 8 core บนแกน x ส่วนแกน y จะแสดงค่าของ sleeping time โดยผลที่ได้คือ ค่าประสิทธิภาพของระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพจะมีลักษณะคล้ายระฆังคว่ำและสาเหตุดังกล่าวถูกอธิบายในบทที่ 4 ผลการวิจัย หัวข้อที่ 4.2.2 การปรับขนาดของ

CPU core ในฝั่งผู้ให้บริการ โดยค่า sleeping time ที่ดีที่สุด คือ ผู้ให้บริการต้องกำหนดขนาดของ CPU core ให้มีขนาด 1 core และ 8 core เมื่อมีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ด้วยการกำหนดพื้นที่การทำงานให้กับโปรเซส logger ด้วยเครื่องมือ taskset ใน linux แล้ว จะสังเกตุจากกราฟได้ว่าการทำงานของระบบบันทึกเหตุการณ์ดีขึ้น ความแม่นยำมีความคงที่มากขึ้นทำให้ไม่เกิดระฆังคว่ำเมื่อมีการเพิ่มขนาดของ CPU core ในฝั่งของผู้ให้บริการ และนอกจากความคงที่ของประสิทธิภาพที่ดีขึ้นแล้ว ค่าของช่วงเวลา sleeping time ลดลงจาก 65 ms. ลงมาที่ 60 ms. ซึ่งในช่วงเวลา 5 ms. ที่ระบบบันทึกเหตุการณ์ที่ผ่านการปรับปรุงทำได้ ทำให้โปรเซส logger ในระบบบันทึกเหตุการณ์สามารถจับไฟล์ที่มีขนาดเล็กลงได้ถึง 32.85% (รายละเอียดถูกกล่าวในหัวข้อที่ 4.4.2 ผลของการปรับปรุงประสิทธิภาพในบทที่ 4 ผลการวิจัย) โดยวิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ได้ทำงานด้วยเครื่องมือ taskset ส่งผลให้ประสิทธิภาพดียิ่งขึ้น สามารถสังเกตได้จากภาพที่ 22

จากผลการปรับปรุงประสิทธิภาพที่อธิบายในหัวข้อนี้ ด้วยกราฟเส้นเปรียบเทียบจะทำให้เห็นความชัดเจนของระบบบันทึกเหตุการณ์ที่ผ่านการปรับปรุงประสิทธิภาพแล้วว่า ระบบบันทึกเหตุการณ์สามารถจับไฟล์ที่มีขนาดเล็กลงได้ถึง 23,667 bytes หรือ 32.85% โดยวิทยานิพนธ์เล่มนี้จะเน้นวิธีการกำหนดพื้นที่สำหรับโปรเซสในการประมวลผล โดยใช้เครื่องมือ taskset ซึ่งวิธีการปรับปรุงประสิทธิภาพแบบอื่น ๆ ทางผู้วิจัยได้เขียนเป็นข้อเสนอแนะวิธีการต่าง ๆ ในบทถัดไป บทที่ 5 อภิปรายผล สรุปผลและข้อเสนอแนะ

## 4.5 สรุปภาพรวมบทที่ 4

ในบทที่ 4 ผลการวิจัยได้อธิบายเกี่ยวกับการสรุปผลการทดลอง การวิเคราะห์ผลการทดลองและการปรับปรุงประสิทธิภาพโดยสามารถแบ่งรายละเอียดเป็นหัวข้อได้ดังนี้

หัวข้อที่ 4.1 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM

หัวข้อที่ 4.2 ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core

หัวข้อที่ 4.3 ผลส่วนที่ 1 วิเคราะห์ผลกระทบที่เกิดขึ้นกับระบบบันทึกเหตุการณ์โดยใช้หลักการทำงานของระบบปฏิบัติการมาช่วยในการวิเคราะห์

หัวข้อที่ 4.4 ผลส่วนที่ 2 นำผลวิเคราะห์ที่ได้ไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และเปรียบเทียบผลการปรับปรุงประสิทธิภาพ

จาก 4 หัวข้อข้างต้นที่ผู้วิจัยได้อธิบายรายละเอียดไว้ในบทที่ 4 วิธีการดำเนินวิจัยที่ผ่านมาแล้วสามารถสรุปแต่ละหัวข้อได้ดังนี้

ในหัวข้อ 4.1 เรื่อง ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ RAM ในหัวข้อนี้จะอธิบายเกี่ยวกับการทดลองที่ผู้วิจัยได้ทำการปรับขนาดของ RAM ทั้งในส่วนของผู้ให้บริการและผู้ใช้บริการจากนั้นบันทึกผลการทดลองและสรุปเกี่ยวกับผลกระทบที่เกิดขึ้น ในส่วนของการทดลองแรกเมื่อมีการปรับขนาด RAM ในฝั่งผู้ใช้บริการแล้วผลกระทบที่ตามมาคือประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงโดยสาเหตุที่เกิดขึ้นได้ถูกอธิบายไว้ในหัวข้อที่ 4.1.1 การปรับขนาด RAM ในฝั่งของผู้ใช้บริการ และอีกผลการทดลองคือผู้วิจัยได้ทดลองปรับขนาด RAM ในฝั่งผู้ให้บริการผลปรากฏว่าเมื่อมีการปรับขนาดของ RAM ในส่วนนี้ไม่ส่งผลกระทบใด ๆ ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์และเมื่อผู้วิจัยได้ทดลองในส่วนของการปรับขนาด RAM สำเร็จแล้วการทดลองต่อไปคือการปรับขนาดของ CPU core และสรุปผลกระทบที่เกิดขึ้น

ในหัวข้อที่ 4.2 เรื่อง ส่วนของผลกระทบที่เกิดจากการปรับขนาดของ CPU core ในหัวข้อนี้จะอธิบายเกี่ยวกับผลการทดลองที่ผู้วิจัยได้ทำการปรับขนากของ CPU core ทั้งในส่วนของผู้ให้บริการและผู้ใช้บริการจากนั้นบันทึกผลการทดลอง สรุปผลการทดลองและอธิบายเกี่ยวกับผลกระทบที่เกิดขึ้นในส่วนของการทดลองแรกเมื่อมีการปรับขนาดของ CPU core ในฝั่งผู้ใช้บริการแล้วไม่ส่งผลกระทบใดๆ ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์และส่วนของการทดลองที่ 2 การปรับขนาดของ CPU core ในฝั่งผู้ให้บริการผลกระทบที่เกิดขึ้นคือส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ใน logger มีประสิทธิภาพลดลงโดยสาเหตุที่เกิดขึ้นได้ถูกอธิบายไว้ในหัวข้อที่ 4.2.2 การปรับขนาดของ CPU core ในฝั่งของผู้ให้บริการ

ในหัวข้อที่ 4.3 เรื่อง ผลส่วนที่ 1 วิเคราะห์ผลกระทบที่เกิดขึ้นกับระบบบันทึกเหตุการณ์โดยใช้หลักการทำงานของระบบปฏิบัติการมาช่วยในการวิเคราะห์ โดยในหัวข้อนี้ได้นำผลการทดลองทั้งหมดที่ผู้วิจัยได้ทดลองทั้งการปรับขนาดของ RAM และ CPU core ในส่วนของผู้ให้บริการและผู้ใช้บริการรวมทั้งหมด 4 ผลการทดลองมาวิเคราะห์เกี่ยวกับผลกระทบที่เกิดขึ้นเมื่อมีการปรับขนาดของฮาร์ดแวร์ดังกล่าวเพื่อนำไปสู่การออกแบบการปรับปรุงประสิทธิภาพต่อไป

หัวข้อที่ 4.4 เรื่อง ผลส่วนที่ 2 นำผลวิเคราะห์ที่ได้ไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และเปรียบเทียบผลการปรัยปรุงประสิทธิภาพ ในหัวข้อนี้ได้นำเอาผลการวิเคราะห์ในหัวข้อที่ 4.3 มาเป็นแนวทางสำหรับการออกแบบเพื่อปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์โดยที่ผู้วิจัยได้ใช้เครื่องมือ taskset ในระบบปฏิบัติการ linux มาช่วยในการปรับปรุงประสิทธิภาพให้ดียิ่งขึ้นโดยรายละเอียดของการปรับปรุงประสิทธิภาพและเครื่องมือที่ใช้ผู้อ่านสามารถกลับไปอ่านในหัวข้อที่ 4.4 ผลส่วนที่ 2 นำผลวิเคราะห์ที่ได้ไปปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์และเปรียบเทียบผลการปรับปรุงประสิทธิภาพและเมื่อปรับปรุงประสิทธิภาพแล้วผู้วิจัยได้สรุปผลอภิปรายผล และให้ข้อเสนอแนะเกี่ยวกับการทดลองโดยรายละเอียดจะถูกกล่าวเพิ่มเติมในบทที่ 5

# บทที่ 5

## สรุปผล อภิปรายผล และข้อเสนอแนะ

ในยุคปัจจุบันคลาวด์คอมพิวติ้ง คลาวด์หรือการประมวลผลแบบกลุ่มเมฆได้มีการถูกใช้อย่างแพร่หลายไม่ว่าจะเป็นทางด้านของส่วนบุคคล หรือองค์กรต่าง ๆ แต่ความกังวลเกี่ยวกับเรื่องความปลอดภัยบนคลาวด์ก็เป็นประเด็นที่สำคัญต่อการนำคลาวด์ไปใช้งานจริง โดยเฉพาะการนำคลาวด์ไปใช้ภายในองค์กรต่าง ๆ ดังนั้นการวิจัยเกี่ยวกับภัยคุกคามหรือความเสี่ยงต่าง ๆ ที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์นั้นเป็นเรื่องที่สำคัญ จึงทำให้มีองค์กรได้ทำวิจัยเกี่ยวกับภัยคุกคามของคลาวด์นั่นคือ องค์กร CSA และสิ่งที่ต้องคำนึงถึงต่อมาหลังจากมีการวิจัยเกี่ยวกับภัยคุกคามของคลาวด์ คือ การแก้ปัญหาหรือการหาวิธีป้องกันเกี่ยวกับปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ซึ่งหากมีการป้องกันหรือบรรเทาปัจจัยเสี่ยงดังกล่าวแล้ว จะสร้างความเชื่อมั่นให้กับผู้ที่ใช้บริการคลาวด์ โดยวิธีการบรรเทาปัจจัยเสี่ยงที่อาจเกิดขึ้นต่อคลาวด์มีหลากหลายงานวิจัยที่ได้นำเสนอไว้ ซึ่งวิธีการใช้ระบบบันทึกเหตุการณ์ถือว่าเป็นหนึ่งในวิธีการดังกล่าวที่สามารถนำไปใช้บรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ และวิธีดังกล่าวเป็นวิธีที่ผู้วิจัยสนใจที่จะศึกษาและปรับปรุงประสิทธิภาพเพื่อให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพที่ดียิ่งขึ้นจนสามารถนำไปใช้งานจริง โดยระบบบันทึกเหตุการณ์สามารถบันทึกได้ว่าใครเคยเข้าถึงหรือทำอะไรกับไฟล์ไว้บ้างและนำข้อมูลการบันทึกที่ได้ไปเป็นหลักฐาน เพื่อสืบหาบุคคลกระทำผิดมารับผิดชอบ โดยในระบบบันทึกเหตุการณ์มีโปรเซสสำหรับใช้บันทึกเหตุการณ์ เรียกว่า logger หรือล็อกเกอร์โดยโปรเซส logger ดังกล่าวถือว่าเป็นปัจจัยหลักที่สำคัญต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ หากมีการปรับปรุงในส่วนของโปรเซส logger ก็จะสามารถเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ ดังนั้นในงานวิทยานิพนธ์เล่มนี้จึงได้สนใจที่จะปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์

เพื่อบรรลุวัตถุประสงค์ในการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ผู้วิจัยจึงได้ตั้งวัตถุประสงค์ของงานวิจัยและสมมติฐานเพื่อนำไปสู่การปรับปรุงประสิทธิภาพได้ โดยรายละเอียดจะกล่าวในหัวข้อถัดไป

## 5.1 สรุปผลการวิจัย

จากบทนำของบทที่ 5 อภิปรายผล สรุปผล และข้อเสนอแนะ ย่อหน้าที่สองด้านบนได้กล่าวว่าในส่วนของบทที่ 5 นี้ ผู้วิจัยจะอธิบายเกี่ยวกับวัตถุประสงค์การวิจัย สมติฐานการวิจัย โดยจะถูกอธิบายในหัวข้อสรุปผลการวิจัยนี้ ในย่อหน้าถัดไป

งานวิจัยในครั้งนี้มีจุดประสงค์ 3 ข้อ คือ 1. ต้องการวิเคราะห์ผลกระทบการปรับขนาดของ RAM ต่อระบบบันทึกเหตุการณ์ 2. ต้องการวิเคราะห์ผลกระทบการปรับขนาดของ CPU core ต่อระบบบันทึกเหตุการณ์ และ 3. เพื่อปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์หลังการวิเคราะห์ผลกระทบที่เกิดขึ้นจากการปรับขนาดของ RAM และ CPU core โดยจุดประสงค์ทั้ง 3 ข้อเป็นวัตถุประสงค์ที่ถูกกล่าวไว้ในบทที่ 1 บทนำ และวิธีที่ใช้ในการปรับปรุงประสิทธิภาพได้จากการวิเคราะห์ผลกระทบที่เกิดขึ้นกับระบบบันทึกเหตุการณ์ โดยวิธีที่ใช้ในการทดลองของวิทยานิพนธ์เล่มนี้ คือ การกำหนดพื้นที่สำหรับการทำงานของ CPU core ให้กับ logger ในระบบบันทึกเหตุการณ์บนคลาวด์ ด้วยเครื่องมือ taskset บนระบบปฏิบัติ linux สาเหตุที่ผู้วิจัยเลือกใช้วิธีดังกล่าวเพราะเป็นวิธีแรกที่ผู้วิจัยค้นพบหลังสรุปผลวิเคราะห์เกี่ยวกับผลกระทบ ที่เกิดขึ้นกับระบบบันทึกเหตุการณ์หลังมีการปรับขนาดของ RAM และ CPU core และยังเป็นวิธีที่ตรงกับสมมติฐานที่ผู้วิจัยได้กล่าวไว้ในบทที่ 1 บทนำ หัวข้อของสมมติฐานการวิจัย นอกจากนี้ผู้วิจัยต้องการให้วิธีการดังกล่าวเป็นต้นแบบสำหรับการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์จึงได้เลือกใช้เครื่องมือ taskset เพื่อกำหนดพื้นที่การทำงานให้กับโปรเซส logger โดยรายละเอียดจะอยู่ในบทที่ 3 กรอบวิธีการดำเนินการวิจัย หัวข้อที่ 3.5 วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ ซึ่งผลการทดลองหลังมีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ใหม่ ทำให้ทราบว่าหากมีการใช้เครื่องมือ taskset ให้กับโปรเซส logger แล้วจะส่งผลให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพดีขึ้น

เมื่อเพิ่มขนาด CPU core ในฝั่งของผู้ให้บริการผลปรากฏว่าประสิทธิภาพของระบบบันทึกเหตุการณ์ไม่ได้ดีขึ้นเสมอไปเพราะประสิทธิภาพที่ดีที่สุดของระบบบันทึกเหตุการณ์เมื่อโปรเซส logger ต้องประมวลผลบน CPU core ขนาด 1 core หรือ 8 core เท่านั้น โดยสาเหตุดังกล่าวถูกกล่าวไว้ในบทที่ 4 ผลการวิจัย ในหัวข้อที่ 4.2.1 ผลวิเคราะห์จากการปรับขนาดของ CPU core หัวข้อที่ 2) การปรับขนาดของ CPU core ในส่วนของผู้ใช้บริการ และเมื่อมีการปรับปรุงประสิทธิภาพด้วยเครื่องมือ taskset แล้ว การประมวลผลของ logger มีความคงที่มากขึ้น ไม่ว่าจำนวน CPU core จะมีขนาดเท่าใดและทำให้โปรเซส logger สามารถจับโปรเซสขณะอ่านไฟล์ที่มีขนาดเล็กลงได้จากระบบบันทึกเหตุการณ์เดิมที่สามารถจับโปรเซสขณะอ่านไฟล์ได้ขนาด 26,952 byte และระบบบันทึกเหตุการณ์ที่ปรับปรุงประสิทธิภาพแล้วสามารถจับโปรเซสขณะอ่านไฟล์ได้ขนาด 23,667 byte หรือคิดเป็นเปอร์เซ็นได้ 32.85% ซึ่งหากไฟล์มีขนาดที่เล็กลงจะทำให้โปรเซสที่

อ่านไฟล์ทำงานได้ไว้ขึ้นส่งผลให้ระบบบันทึกเหตุการณ์ไม่สามารถเก็บข้อมูลของโปรเซสที่เข้ามาอ่านไฟล์ได้ทัน ดังนั้นการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ให้โปรเซส logger สามารถจับโปรเซสที่กำลังอ่านไฟล์ที่มีขนาดเล็กลงได้

นอกจากวิธีการนำเครื่องมือ taskset บนระบบปฏิบัติการ linux ไปใช้ปรับปรุงในส่วนของการเพิ่มขนาด CPU core ของผู้ให้บริการแล้ว เครื่องมือดังกล่าวยังสามารถนำไปใช้ปรับปรุงประสิทธิภาพในส่วนของการเพิ่มขนาดของ RAM ในฝั่งของผู้ใช้บริการได้ โดยระบบบันทึกเหตุการณ์เดิมที่ยังไม่มีการปรับปรุงประสิทธิภาพเมื่อมีการเพิ่มขนาดของ RAM ในส่วนของผู้ใช้บริการ ความแม่นยำของระบบบันทึกเหตุการณ์จะลดลงเรื่อย ๆ หากผู้ใช้บริการมีการเพิ่มขนาดของ RAM แต่เมื่อมีการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์แล้วทำให้ระบบบันทึกเหตุการณ์ดังกล่าวที่ความแม่นยำเพิ่มขึ้นเป็น 100% ไม่ว่าผู้ใช้บริการจะปรับขนาดของ RAM ให้มีขนาดจำนวนเท่าใด จากทุกครั้งที่เมื่อเพิ่ม RAM ของผู้ใช้บริการความแม่นยำจะลดลง จาก 100% ลงไปที่ 99.97 99.96 และ 99.96 ตามลำดับ โดยความแม่นยำจะลดลงเมื่อมีการเพิ่มขนาดของ RAM ซึ่งระบบบันทึกเหตุการณ์ที่มีประสิทธิภาพควรจะต้องจับโปรเซสให้ได้ 100% ทุก ๆ ครั้งไม่ว่าขนาด RAM ของผู้ใช้บริการจะมีขนาดเท่าใด ดังนั้นต้องมีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ขึ้น

จากสมมติฐานการวิจัยได้ที่ผู้วิจัยได้ตั้งไว้ 3 ข้อจากบทที่ 1 บทนำสามารถสรุปได้ 3 ข้อดังนี้

1) สมมติฐานที่หนึ่งที่ผู้วิจัยบรรยายไว้บทที่ 1 บทนำได้อธิบายว่าหากมีการเพิ่มขนาดของ RAM ให้กับระบบบันทึกเหตุการณ์ในส่วนของผู้ใช้และผู้ให้บริการแล้วจะสามารถเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ซึ่งจากผลการทดลองปรากฏว่าเมื่อเพิ่มขนาดของ RAM ในส่วนของผู้ใช้บริการแล้วประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงเพราะการเพิ่ม RAM ในส่วนนี้เป็นการเพิ่มภาระในการทำงานของระบบบันทึกเหตุการณ์และเมื่อทดลองเพิ่ม RAM ในส่วนผู้ให้บริการผลปรากฏว่าในส่วนนี้ประสิทธิภาพของระบบบันทึกเหตุการณ์ยังคงเดิม

2) จากสมมติฐานที่ตั้งไว้ในบทที่ 1 บทนำและอธิบายว่าหากมีการเพิ่มขนาดของ CPU core ในส่วนของผู้ใช้และผู้ให้บริการแล้วประสิทธิภาพของระบบบันทึกเหตุการณ์จะดีขึ้น ผลปรากฏว่าหากมีการเพิ่มขนาด CPU core ให้กับผู้ใช้บริการประสิทธิภาพของระบบบันทึกเหตุการณ์จะมีประสิทธิภาพคงเดิมแต่เมื่อเพิ่มขนาดของ CPU ในส่วนของผู้ให้บริการประสิทธิภาพของระบบบันทึกเหตุการณ์กลับลดลงสาเหตุมาจากการจัดสรรพื้นที่ในการทำงานของระบบปฏิบัติการ

3) จากสมมติฐานที่ผู้วิจัยได้ตั้งไว้ว่าหากประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงสาเหตุน่าจะเกิดขึ้นจากการจัดสรรพื้นที่ในการทำงานของระบบปฏิบัติการซึ่งในการทดลองเป็นไปตามสมมติฐานโดยเมื่อระบบปฏิบัติการได้จัดสรรพื้นที่ในการทำงานให้กับโปรเซสล็อกเกอร์แล้ว

จะส่งผลต่อประสิทธิภาพในการทำงานของระบบบันทึกเหตุการณ์ลดลงจึงต้องมีการกำหนดพื้นที่ในการทำงานของโปรเซสล็อกเกอร์เพื่อให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น

จากผลสรุปการปรับปรุงประสิทธิภาพของการทดลองในวิทยานิพนธ์เล่มนี้ ผลสรุปทั้ง 2 การทดลอง ผู้วิจัยได้กล่าวไว้ในหัวข้อที่ 1. สรุปผลการวิจัยและผู้วิจัยจะนำผลการทดลองดังกล่าวไปอภิปรายผลการทดลองและอธิบายว่าตรงกับสมติฐานใดบ้างที่ผู้วิจัยเคยกล่าวในบทที่ 1 บทนำ โดยรายละเอียดจะกล่าวในหัวข้อที่ 5.2 อภิปรายผลหัวข้อถัดไป

## 5.2 อภิปรายผล

จากบทสรุปในหัวข้อที่ 1. สรุปผลการวิจัยได้กล่าวว่าจะมีการอภิปรายผลเกี่ยวกับการทดลองและอธิบายว่าสอดคล้องกับทฤษฎีใด ตามสมมติฐานในบทที่ 1 บทนำ โดยรายละเอียดของการอภิปรายผล จะถูกอธิบายในหัวข้อย่อหน้าถัดไป

จากการทดลองที่ผู้วิจัยได้ทำการทดลองจะสามารถอภิปรายผลเกี่ยวกับผลการทดลองได้ 2 ประเด็นการทดลอง คือ 1. การปรับขนาดของ RAM ในฝั่งผู้ใช้บริการและ 2. การปรับขนาดของ CPU core ในฝั่งผู้ให้บริการโดยจะมีรายละเอียดดังนี้

### 5.2.1 การปรับขนาดของ RAM ในฝั่งผู้ใช้บริการ

จากผลสรุปของการทดลองเมื่อมีการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์แล้ว หากมีการเพิ่ม RAM ในฝั่งผู้ใช้บริการ จะทำให้ประสิทธิภาพความแม่นยำลดลง โดยสมมติในการทดลองได้กำหนด RAM ของผู้ใช้บริการให้มีขนาด 1 GB. 2 GB. และ 3 GB. ตามลำดับ ความแม่นยำของระบบบันทึกเหตุการณ์ที่ได้ คือ 99.97% 99.96% และ 99.96% ตามลำดับจะสังเกตได้ว่าหากมีการเพิ่มขนาดของ RAM ในฝั่งของผู้ให้บริการค่าความแม่นยำจะลดลง แต่เมื่อมีการปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์แล้ว ค่าความแม่นยำของระบบบันทึกเหตุการณ์ที่ได้ คือ RAM ขนาด 1 GB. = 100% ขนาด 2 GB. = 100% และ 3 GB. = 100% ซึ่งการปรับปรุงประสิทธิภาพทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้นความแม่นยำไม่ลดลง โดยสาเหตุที่ประสิทธิภาพดีขึ้นมาจากการกำหนดพื้นที่ให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ได้ทำงาน ทำให้ระบบบันทึกเหตุการณ์มีความไวขึ้นจึงสามารถตรวจจับโปรเซสเมื่อมีการอ่านไฟล์ได้ทัน ทำให้เมื่อมีการเพิ่มขนาดพื้นที่ของ RAM จึงไม่ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ ซึ่งวิธีการกำหนดพื้นที่สำหรับการทำงานให้โปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ จะตรงกับสมมติฐานที่ผู้วิจัยได้ตั้งไว้ในบทที่ 1 บทนำที่กล่าวว่าหากมีการเพิ่มขนาดของฮาร์ดแวร์หรือ RAM จะส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์และวิธีการ

แก้ไขคือต้องมีการจัดสรรพื้นที่สำหรับโปรเซส logger ได้ทำงานและเมื่อผู้วิจัยได้กำหนดพื้นที่สำหรับโปรเซส logger ได้ทำงานและผลการทดลองที่ได้ คือ ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้นตรงตามสมมติฐาน

จากการอภิปรายผลในส่วนของการปรับขนาดของ RAM ในฝั่งของผู้ ให้บริการเป็นไปตามสมมติฐานที่ตั้งไว้ทำให้ประสิทธิภาพดีขึ้นเพราะการจัดสรรพื้นที่การทำงานของระบบปฏิบัติการที่ได้มีการจัดสรรพื้นที่ในลักษณะที่ระบบปฏิบัติการเป็นตัวกำหนดให้โปรเซส logger ทำงานซึ่งอาจส่งผลกระทบต่อประสิทธิภาพของโปรเซส logger ทำให้ประสิทธิภาพความแม่นยำของระบบบันทึกเหตุการณ์ลดลง ซึ่งรายละเอียดของการจัดสรรพื้นที่ในระบบปฏิบัติการจะถูกกล่าวในหัวข้อการปรับขนาดของ CPU core ฝั่งของผู้ให้บริการถัดไป

**5.2.2 การปรับขนาดของ CPU core ฝั่งของผู้ให้บริการ**

จากผลสรุปของการทดลองเมื่อมีการปรับขนาดของ CPU core ฝั่งของผู้ให้บริการผลปรากฏว่าการเพิ่มขนาดของ CPU core ไม่ได้ส่งผลให้ระบบบันทึกเหตุการณ์ดีขึ้นเสมอไปเนื่องจากเมื่อมีการเพิ่มขนาดของ CPU core จากขนาด 1 core ไป CPU core ขนาด 2 core ประสิทธิภาพของ logger กลับลดลงเมื่อเพิ่มขนาดของ CPU core ขนาด 3 core จนถึงขนาด 7 core ก็ยิ่งทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงเรื่อย ๆ แต่เมื่อขนาดของ CPU core มีขนาด 8 core กลับทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น จึงได้ข้อสรุปว่าการเพิ่มขนาดของ CPU core ไม่ได้ทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้นเสมอไป ซึ่งจะตรงกับสมมติฐานของผู้วิจัยที่อธิบายไว้ในบทที่ 1 บทนำ ว่าหากมีการเพิ่มขนาดของฮาร์ดแวร์หรือ CPU core อาจส่งผลกระทบบางอย่างต่อระบบบันทึกเหตุการณ์และอาจไม่ได้ทำให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพที่ดีขึ้น ผลเนื่องจากระบบปฏิบัติการได้กำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ ทำให้เมื่อโปรเซส logger ทำการประมวลผลต่อเนื่องแล้วระบบปฏิบัติการได้สลับพื้นที่การทำงานระหว่าง CPU core ของโปรเซส logger ไปมา ทำให้เมื่อโปรเซส logger ย้ายการทำงานจาก CPU core หนึ่งไปอีก CPU core หนึ่งส่งผลต่อประสิทธิภาพที่ลดลง เนื่องจากหลักการจัดสรรพื้นที่ของระบบปฏิบัติจะเป็นตัวกำหนดพื้นที่สำหรับการทำงานของโปรเซสทุกตัว จึงส่งผลให้เมื่อระบบปฏิบัติการกำหนดพื้นที่การทำงานให้กับโปรเซส logger แล้วมีโอกาสที่โปรเซส logger จะสลับการทำงานไปมาระหว่าง CPU core ทำให้ประสิทธิภาพลดลง

เมื่อทราบสาเหตุว่าประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงจากการกำหนดพื้นที่การทำงานของระบบปฏิบัติการ ผู้วิจัยจึงได้ทำการกำหนดพื้นที่ให้โปรเซส logger ได้ทำงานโดยไม่ต้องมีการสลับพื้นที่การทำงานของ CPU core ไปมาเหมือนที่ระบบปฏิบัติการได้จัดสรรพื้นที่ให้ ซึ่งผลของการปรับปรุงประสิทธิภาพเมื่อมีการกำหนดพื้นที่สำหรับโปรเซส logger ได้ทำงาน ส่งผลให้

ระบบบันทึกเหตุการณ์ใหม่ โปรเซส logger จะสามารถจับโปรเซสที่อ่านไฟล์ที่มีขนาดเล็กลงจากขนาด 26,952 byte ลดลงเหลือขนาด 23,667 byte หรือคิดเป็นเปอร์เซ็นได้ 32.85% เนื่องจากหากไฟล์มีขนาดที่เล็กลงและเมื่อรันโปรเซสเพื่อเข้าไปอ่านข้อมูลของไฟล์ดังกล่าว จะสามารถอ่านข้อมูลได้ไวขึ้นจนส่งผลให้โปรเซส logger ในระบบบันทึกเหตุการณ์ไม่สามารถเก็บข้อมูลจากการอ่านไฟล์เพื่อนำไปเป็นหลักฐานในการหาตัวบุคคลกระทำผิดมารับผิดชอบได้ทันดังนั้นเมื่อมีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์จนโปรเซส logger จะสามารถตรวจจับโปรเซสที่กำลังอ่านไฟล์ขนาดเล็กได้ทันและเมื่อประสิทธิภาพของโปรเซส logger ดีขึ้นจึงส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้นตามลำดับ

จากการอภิปรายผลจะเห็นได้ว่าในการทดลองของวิทยานิพนธ์เล่มนี้ได้ใช้วิธีการวิเคราะห์เกี่ยวกับผลกระทบที่เกิดขึ้น จนสามารถหาวิธีการที่จะปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ นั้นคือ วิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ ซึ่งวิธีดังกล่าวเป็นเพียงหนึ่งในวิธีการปรับปรุงประสิทธิภาพ โดยวิธีการปรับปรุงประสิทธิภาพวิธีอื่น ๆ ผู้วิจัยจะระบุในหัวข้อถัดไป 5.3 ข้อเสนแนะพร้อมทั้งอธิบายเหตุผลทำไมถึงได้เลือกใช้วิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึเหตุการณ์บนคลาวด์

## 5.3 ข้อเสนอแนะ

ในการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์นั้นอาจจะสามารถใช้วิธีการอื่นๆ ได้เช่นเดียวกันซึ่งตัวอย่างของวิธีการปรับปรุงประสิทธิภาพดังกล่าวจะเขียนอธิบายไว้ในหัวข้อที่ 5.3.1 การปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ และนอกจากนี้หากต้องการนำระบบบันทึกเหตุการณ์ไปใช้งานกับระบบปฏิบัติการอื่น ๆ นอกเหนือจาก linux เช่น Ubuntu Window ก็สามารถทำได้เช่นเดียวกันซึ่งจะอธิบายเพิ่มเติมในหัวข้อ 5.3.2 การนำระบบบันทึกเหตุการณ์ไปใช้บนระบบปฏิบัติการอื่นโดยทั้ง 2 หัวข้อดังกล่าวมีรายระเอียด ดังนี้

### 5.3.1 การปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์

ในหัวข้อ 5.3.1 ข้อเสนอแนะผู้วิจัยจะอธิบายเกี่ยวกับวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ที่ผู้วิจัยใช้ในงานวิทยานิพนธ์เล่มนี้ คือ การกำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ โดยผู้วิจัยจะให้ข้อเสนอแนะเกี่ยวกับการนำวิธีการดังกล่าวไปใช้และการนำวิธีการดังกล่าวไปใช้ร่วมกับการปรับปรุงประสิทธิภาพโดยวิธีอื่น ๆ ที่ผู้วิจัยได้เคยกล่าวไว้ในบทที่ 3 วิธีดำเนินการวิจัย หัวข้อที่ 3.5 วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์โดยรายละเอียดทั้งหมดจะอธิบายในย่อหน้าถัดไป

จากข้อสรุปในบทที่ 3 วิธีดำเนินการวิจัย หัวข้อที่ 3.5 วิธีการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ ทำให้ทราบว่านอกจากวิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์แล้ว ยังมีวิธีอื่น ๆ ที่อาจจะช่วยเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ได้เช่นเดียวกัน นั้นคือ วิธีการปรับปรุงประสิทธิภาพของโค้ดดิ้งใหม่ ซึ่งการเขียนโค้ดดิ้งของโปรเซส logger ใหม่โดยลดขั้นตอนการทำงานของโปรเซส logger ลงอาจจะทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น แต่จุดประสงค์ของวิทยานิพนธ์เล่มนี้ต้องการสร้างต้นแบบการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ จึงได้นำเสนอเพียงวิธีเดียว เพราะการปรับปรุงโค้ดดิ้งจะต้องใช้เวลาในการทดลองที่มากขึ้นผู้วิจัยจึงได้ใช้วิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger เพียงวิธีเดียวเพื่อสร้างเป็นต้นแบบสำหรับการเพิ่มประสิทธิภาพในแบบอื่น ๆ ต่อไป นอกจากนี้ถ้าหากมีการทดลองปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ด้วยวิธีการอื่น ๆ และวิธีการอื่น ๆ เหล่านั้นอาจจะใช้งานร่วมกับวิธีการกำหนดพื้นที่การทำงานของวิทยานิพนธ์เล่มนี้ได้ทำให้ได้ระบบบันทึกเหตุการณ์ที่มีประสิทธิภาพดีขึ้น พร้อมนำไปใช้ในสภาพแวดล้อมจริง

วิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger เพื่อเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ที่ผู้วิจัยได้ทดลองบนระบบปฏิบัติการ linux fedora 16 แล้วนั้น วิธีการดังกล่าวอาจจะสามารถนำไปใช้กับระบบบันทึกเหตุการณ์ที่รันบนระบบปฏิบัติการอื่น ๆ ได้ โดยใช้หลักการกำหนดพื้นที่ในการทำงานของโปรเซส logger เช่นเดียวกัน เช่น ระบบปฏิบัติการ window ระบบปฏิบัติการ linux ubuntu หรือระบบปฏิบัติการอื่น ๆ

วิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์สามารถช่วยเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ และวิธีการดังกล่าวยังอาจจะนำไปประยุกต์ใช้กับระบบปฏิบัติการอื่น ๆ นอกเหนือจากระบบปฏิบัติการ linux ที่ใช้ในการทดลองและนอกเหนือจากวิธีการดังกล่าวแล้ว ยังมีวิธีการเขียนโค้ดดิ้งใหม่ให้กับโปรเซส logger ที่น่าจะช่วยเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ แต่ในวิทยานิพนธ์เล่มนี้จะใช้เพียงวิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger เพียงวิธีเดียวเพื่อเป็นต้นแบบในการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ต่อไป

### 5.3.2 การนำระบบบันทึกเหตุการณ์ไปใช้บนระบบปฏิบัติการอื่น

ในหัวข้อนี้จะอธิบายเพิ่มเติมเกี่ยวกับการนำระบบบันทึกเหตุการณ์ไปใช้งานร่วมกับระบบปฏิบัติการอื่น ๆ เนื่องจากขอบเขตในการทดลองของวิทยานิพนธ์เล่มนี้ได้ใช้ระบบปฏิบัติการ linux fedora 16 ในการจำลองสร้างคลาวด์ IaaS แบบสาธารณะดังนั้นหากต้องการนำระบบบันทึกเหตุการณ์ดังกล่าวไปใช้งานร่วมกับระบบปฏิบัติการอื่น ๆ จำเป็นต้องทราบเกี่ยวกับโครงสร้างพื้นฐานของคลาวด์ที่ทำให้คลาวด์ฝั่งผู้ให้บริการสามารถเข้าไปบันทึกเหตุการณ์ที่เกิดขึ้นในฝั่งผู้ใช้บริการได้โดยสามารถอธิบายรายละเอียดได้ดังนี้

หากต้องการนำระบบบันทึกเหตุการณ์ไปใช้ตรวจจับเหตุการณ์ที่เกิดขึ้นบนคลาวด์ในระบบปฏิบัติการอื่น ๆ นอกจากระบบปฏิบัติการ linux ผู้ให้บริการจำเป็นที่จะต้องทราบเกี่ยวกับข้อมูลใน system.map ซึ่งเป็นไฟล์ที่จะทำให้ล็อกเกอร์สามารถเข้าไปบันทึกเหตุการณ์ที่เกิดขึ้นภายในระบบปฏิบัติการอื่น ๆ ได้โดยรายละเอียดของไฟล์ system.map ดังตารางที่ 6

| | |
|---|---|
| [domain name] { | |
| [ 1953.578814] | ostype = "Linux"; |
| [ 1953.578817] | sysmap = "[insert path here]"; |
| [ 1953.578820] | linux_name = 0x460; |
| [ 1953.578824] | linux_tasks = 0x240; |
| [ 1953.578826] | linux_mm = 0x278; |
| [ 1953.578829] | linux_pid = 0x2b4; |
| [ 1953.578832] | linux_pgd = 0x48; |
| [ 1953.578835] } | |

**ตารางที่ 6 ข้อมูลภายในไฟล์ system.map**

จากข้อมูลภายในไฟล์ system.map ซึ่งเป็นไฟล์ที่ผู้ให้บริการต้องคัดลอกไปไว้ในล็อกเกอร์เพื่อให้ล็อกเกอร์นั้นทราบตำแหน่งที่อยู่ของระบบปฏิบัติการที่ต้องการเข้าไปบันทึกข้อมูล จึงจะทำให้ระบบบันทึกเหตุการณ์สามารถมองเห็นเครื่องของผู้ใช้บริการได้จนกระทั่งสามารถเข้าไปบันทึกข้อมูลที่อยู่ในเครื่องผู้ใช้บริการ

จากวิธีการนำระบบบันทึกเหตุการณ์ไปใช้กับระบบปฏิบัติการอื่น ๆ วิธีดังกล่าวนี้เป็นเพียงวิธีที่ผู้วิจัยได้แนะนำจากประสบณ์การทดลองและตั้งสมมติฐานเมื่อต้องการนำระบบบันทึกเหตุการณ์ไปใช้งานกับระบบปฏิบัติการอื่น ๆ ซึ่งวิธีการดังกล่าวอาจจะใช้ได้จริงแต่ผู้ให้บริการต้องศึกษาโครงสร้างของสถาปัตยกรรมในระบบปฏิบัติการอื่น ๆ เช่น Window linux Ubuntu เป็นต้น

## 5.4 สรุปภาพรวมบทที่ 5

ในบทที่ 5 สรุปผล อภิปรายผล และข้อเสนอแนะสามารถสรุปรายละเอียดโดยแบ่งรายละเอียดเป็นหัวข้อได้ดังนี้

หัวข้อที่ 5.1 สรุปผลการวิจัย

หัวข้อที่ 5.2 อภิปรายผล

หัวข้อที่ 5.3 ข้อเสนอแนะ

ในหัวข้อ 5.1 จากผลสรุปการปรับปรุงประสิทธิภาพของการทดลองในวิทยานิพนธ์เล่มนี้สามารถสรุปผลการทดลองหลังการปรับปรุงประสิทธิภาพได้ 2 การทดลอง คือ 1. หากเพิ่ม RAM ในส่วนของผู้ให้บริการแล้วจะส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงจึงต้องมีการปรับปรุงประสิทธิภาพโดยกำหนดพื้นที่การทำงานให้กับโปรเซสล็อกเกอร์ซึ่งผลการทดลองที่ได้ทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์นั้นดียิ่งขึ้น และอีกผลการทดลองที่ได้ทำการปรับปรุงประสิทธิภาพนั้นคือการทดลองที่ 2. การเพิ่มขนาดของ CPU core ให้กับส่วนผู้ให้บริการโดยการทดลองเดิมนั้นเมื่อเพิ่มขนาดของ CPU core ให้กับผู้ให้บริการแล้วประสิทธิภาพที่ดีที่สุดของระบบบันทึกเหตุการณ์จะเกิดขึ้นเมื่อขนาดของ CPU core มีจำนวน 1 core และ 8 core ดังนั้นระบบบันทึกเหตุการณ์ก่อนการปรับปรุงประสิทธิภาพจึงมีประสิทธิภาพที่ลดลงและการทำงานไม่ต่อเนื่องกัน แต่เมื่อมีการกกำหนดพื้นที่การทำงานให้กับโปรเซสล็อกเกอร์ ประสิทธิภาพของระบบบันทึกเหตุการณ์จึงดีขึ้นการทำงานของโปรเซสล็อกเกอร์มีความต่อเนื่องกันมากขึ้นและสามารถบันทึกเหตุการณ์ได้ไวยิ่งขึ้น

ในหัวข้อ 5.2 อภิปรายผลจะเห็นได้ว่าในการทดลองของวิทยานิพนธ์เล่มนี้ได้ใช้วิธีการวิเคราะห์เกี่ยวกับผลกระทบที่เกิดขึ้นจนสามารถหาวิธีการที่จะปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ได้ นั้นคือวิธีการกำหนดพื้นที่การทำงานให้กับโปรเซส logger ในระบบบันทึกเหตุการณ์บนคลาวด์ ซึ่งวิธีดังกล่าวเป็นเพียงหนึ่งในวิธีการปรับปรุงประสิทธิภาพเท่านั้น โดยวิธีการปรับปรุงประสิทธิภาพวิธีอื่น ๆ ก็อาจจะใช้ได้เช่นเดียวกัน เช่น การเขียนโปรแกรมด้วยภาษาใหม่

การเขียนโปรแกรมแบบพาราเรียงโปรแกรมมิ่งหรืออาจจะต้องใช้หลายๆ วิธีมาทำงานร่วมกันจึงจะทำให้สามารถปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ได้

ในหัวข้อ 5.3 ข้อเสนอแนะได้อธิบายเกี่ยวกับการปรับปรุงประสิทธิภาพด้วยวิธีอื่น ๆ เช่น การเขียนโปรแกรมด้วยภาษาใหม่ การเขียนโปรแกรมด้วยพาราเรียลโปรแกรมมิ่งหรือจะนำวิธีการหลายๆ วิธีการไปประยุกต์ใช้ร่วมกันเพื่อให้เกิดระบบบันทึกเหตุการณ์ที่มีประสิทธิภาพดียิ่งขึ้น นอกจากนี้ในหัวข้อนี้ยังได้อธิบายเกี่ยวกับวิธีการนำระบบบันทึกเหตุการณ์ไปใช้ตรวจจับบนระบบปฏิบัติการประเภทอื่น ๆ เช่น Windows Linux Ubuntu โดยวิธีที่ใช้คือต้องศึกษาสถาปัตยกรรมของระบบปฏิบัติและสร้างไฟล์ System.map เพื่อเป็นไฟล์ที่ทำให้ระบบบันทึกเหตุการณ์มองเห็นเหตุการณ์ที่เกิดขึ้นภายในระบบปฏิบัติการอื่น ๆ

# บรรณานุกรม

A.Saha. (2006). **Learning about linux process The Linux Gazette**. [online]. Available : http://linuxgazette.net/133/saha.html

Armbrust M., A. Fox R. G., Joseph A. D., et al. (2010). "A view of Cloud Computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672.

Avudaiyappan K., & Abdallah M. (2014). **SYSTEMS AND METHODS FOR FLUSHING A CACHE WITH MODIFIED DATA**: US Patent 20,140,032,844.

Avudaiyappan K., & Abdallah M. (2017). **Systems and methods for flushing a cache with modified data**: Google Patents.

Bryant R. E., & O' Hallaron D. R. (2010). ***Computer Systems A Programmer' s Perspective*** (2nd Ed.). USA: Addison-Weslwy Publishing Company.

Cache. (15 Mar. 1999). [online]. Available : http://irrigation.rid.go.th/rid15/ppn/Knowledge/Cache/Cache.htm.

Carrick A. G., Levine D., & Elmsi R. (2010). ***Operating Systems***. McGraw Hill: McGraw Hill.

Chan-in P., & Wongthai W. (2017). **performance improvement considerations of cloud**. *ICIC international*.

Chiba Z., Abghour N., Moussaid K., et al. (2016, 22-24 Sept. 2016). ***A survey of intrusion detection systems for cloud computing environment***. Paper presented at the 2016 International Conference on Engineering & MIS (ICEMIS).

Cooperman G. ( 2003). Cache Basics. สืบค้นเมื่อ 24 ตุลาคม 2560, จาก https://course.ccs.neu.edu/com3200/parent/NOTES/cache-basics.html.

CSA. (2010). **Top Threats to Cloud Computing**, V1.0 Tech.Rep.

CSA. (2011). " Security guidance for critical areas of Cloud Computing v3.0," The Cloud Security Alliance

(CSA), Tech. Rep., 2011.

CSA. (2013). **The Notorious Nine: Cloud Computing Top Threats in 2013**, The cloud Security Alliance,(CSA),Tech.Rep.

CSA. (2016a). **The Treacherous 12 - Cloud Security Alliance. Cloud Security Alliance** (pp. 1-35).

CSA. (2016b). **The treacherous 12 cloud computing top threats in 2016**. The cloud Security Alliance,(CSA), Tech.REP.

Dawoud W., Takouna I., & Meinel C. (2010). " Infrastructure as a service security: Challenges and solutions," in Inter-national Conference on Informatics and Systems, 2010, pp. 1–8.

Deitel H. M., Deitel P. J., & Choffnes D. R. (2003). *Operating Systems (3rd Edition) 3rd Edition*. Prentice-Hall, Inc.: Upper Saddle River, NJ, USA.

Ebert C., Dumke R., Bundschuh M., et al. (2005). *Best Practices in Software Measurement: How to use metrics to improve project and process performance*: Springer Science & Business Media.

Flanagan K., Nakjang S., Hallinan J., et al. (2012). "Microbase2.0: A generic framework for computationally intensive bioinformatics

workflows in the cloud," J. Integrative Bioinformatics, 2012.

Flanagan K. S. (2010). " A grid and cloud-based framework for high throughput bioinformatics,"

Ph.D. Thesis, School of Computing Science Newcastl University, 2010.

Galvin P. B., Gagne G., & Silberschartz A. (2006). ***Operating System Principles*** (s. Education Ed.): John Wiley & Sons (Asia) Pte Ltd.

Gartner. (2017). **Gartner Says Detection and Response is Top Security Priority for Organizations in 2017**, สืบค้นเมื่อวันที่ 13 ธันวาคม 2560, จาก https://www.gartner.com/newsroom/id/3638017.

Handy J. (January 27, 1998). *the Cache Memory Book, The, Second Edition (The Morgan Kaufmann Series in Computer Architecture and Design)*. Morgan Kaufmann.

Ko R. K. (2014). **Data accountability in cloud system Security**, Privacy and Trust in Cloud Systems *Springer*, 211-238.

Ko R. K., Jagadpramana P., & Lee B. S. (2011). ***Flogger: A file-centric logger for monitoring file access and transfers within cloud computing environments***. Paper presented at the Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on.

macmillandictionary. สืบค้นเมื่อ 10 ตุลาคม 2560, จาก https://www.macmillandictionary.com/.

Manchester T. U. o. (2011). “Genetics ’cloud’ to create new opportunities for researchers and clinicians.” Online article published at The University of Manchester, 2011. [Online]. Available: http://www.manchester.ac.uk/aboutus/news/display/?id=7371.

Meier J., Farre C., Bansode P., et al. (2007). *Performance testing guidance for web applications: patterns & practices*: Microsoft press.

Mell P., & Grance T. (2011). *“The NIST definition of Cloud Computing,” Computer Security Division,*

*Information Technology Laboratory, National Institute of Standards and Technology, U.S.*

*Department of Commerce, Tech. Rep. NIST Special Publication 800-145, 2011. [Online]. Available:*

*http://csrc.nist.gov/publications/nistpubs/ 800-145/SP800-145.pdf*. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Molyneaux I. (2014). ***The Art of Application Performance Testing From Strategy to Tools.*** *United Startes of America: O'Reilly Media, Inc.,1005 Gravenstein Highway Nort, Sebastopol, CA 95472.*

N.P.Jouppi. (1990. ). Proceedings., doi:10.1109/ISCA.1990.134547 "Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers." - 17th Annual International Symposium on Computer Architecture, .

Nanni D. D. *How to run program or process on specific CPU cores on Linux*. Retrieved from National Institute of Standards and Technology. (2014). ***NIST Cloud Computing Technology Roadmap Volumes I and II,*** *Special Publication 500-293*. Retrieved from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-293.pdf

NSA. (2013). "Cloud security considerations," The National Security Agency (NSA), Tech. Rep., [Online]. Available: The National Security Agency /Central Security Service (NSA).

Papanikolaou N., & Pearson S. (2012). **Accountability in Cloud Computing An Introduction to the Issues, Approaches, and Tools.** สืบค้นเมื่อ 10 ตุลาคม 2560, จาก http://www.bic-trust.eu/files/2012/04/WG1_4_NP.pdf.

Parkin S. E., & Morgan G. (2012). Toward reusable sla monitoring capabilities. *Software: Practice and Experience, 42*(3), 261-280.

R. De Paris. (2012). "FReMI-a middleware to handle molecular docking simulations of fullyflexible

receptor model in hpc environment," Master's thesis, 2012.

Rocha F., Abreu S., & Correia M. (2011). "The final frontier: Confidentiality and privacy in the cloud," Computer, vol. 44, 2011.

S. Mittal. (2012). " A Survey of Architectural Techniques For DRAM Power Management", IJHPSA, 4(2), 110-119, 2012.

Services A. w. (2012). Inc. (2012) Amazon elastic compute cloud (Amazon EC2).

Shotts. W. E. J. (2012). The Linux Command Line: A Complete Introduction. No Starch Press.

Skorobogatov S. (June 2002). "Low temperature data remanence in static RAM". University of Cambridge, Computer Laboratory. Retrieved 2008-02-27.

Subashini S., & Kavitha V. (2011). **A survey on security issues in service delivery models of cloud computing**. *Journal of network and computer applications, 34*(1), 1-11.

Surbiryala J., Li C., & Rong C. (2017, 28-30 April 2017). ***A framework for improving security in cloud computing***. Paper presented at the 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA).

Wang G., Estrada Z. J., Pham C. M., et al. (2015). *Hypervisor Introspection: A Technique for Evading Passive Virtual Machine Monitoring.* Paper presented at the WOOT.

Wongthai W. (2014). *Systematic Support for Accountability in the Cloud.* (Degree of Doctor of Philosophy), Newcastle University.

Wongthai W., & Moorsel A. v. (2016). **Performance measurement of logging systems in infrastructure as a service cloud**. *ICIC international*.

Wongthai W., Rocha F., & Moorsel A. v. (2013a, 16-19 Dec. 2013). *"Logging solutions to mitigate risks associated with threats in infras-tructure as a service cloud," in 2013 International Conference on Cloud Computing and Big Data, Dec 2013, pp. 163–170.* Paper presented at the 2013 International Conference on Cloud Computing and Big Data.

Wongthai W., Rocha F. L., & Moorsel A. v. (2013b, 25-28 March 2013). ***A Generic Logging Template for Infrastructure as a Service Cloud***. Paper presented at the 2013 27th International Conference on Advanced Information Networking and Applications Workshops.

Wongthai W., & van Moorsel A. (2016). **Quality analysis of logging system components in the cloud** *Information Science and Applications (ICISA) 2016* (pp. 651-662): Springer.

ไกรยวิชช์ ศุภโสภาพงศ์, & Wongthai W. (2017a). การวิเคราะห์ผลกระทบของหน่วยความจำหลักต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ในระบบการประมวลผลแบบกลุ่มเมฆ. *CAS National and International Conference 2017 (CASNIC 2017)*, 1286-1295.

ไกรยวิชช์ ศุภโสภาพงศ์, & Wongthai W. (2017b). การวิเคราะห์ผลกระทบของหน่วยประมวลผลกลางต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ในระบบการประมวลผลแบบกลุ่มเมฆ

An Analysis of Impacts of CPU Cores against Logging System Performance in the Cloud. *7th Phayao research conference,* การประชุมวิชาการระดับชาติพะเยา.

ราชบัณฑิตยสภา. (2546). ศัพท์คอมพิวเตอร์และเทคโนโลยีสารสนเทศ ฉบับราชบัณฑิตยสถาน. กรุงเทพฯ: กรุงเทพฯ : ราชบัณฑิตยสถาน.

ราชบัณฑิตยสภา. (2549). ศัพท์ต่างประเทศที่ใช้คำไทยแทนได้**,** ราชบัณฑิตยสถาน**,** พิมพ์ครั้งที่ ๒ (๒๕๔๙). กรุงเทพฯ: กรุงเทพฯ : ราชบัณฑิตยสถาน.

ศรีสมรักษ์ อินทุจันทร์ยง. (2553). การประมวลผลในกลุ่มเมฆ (**Cloud Computing)**. วารสารบริหารธุรกิจ, 14-21.

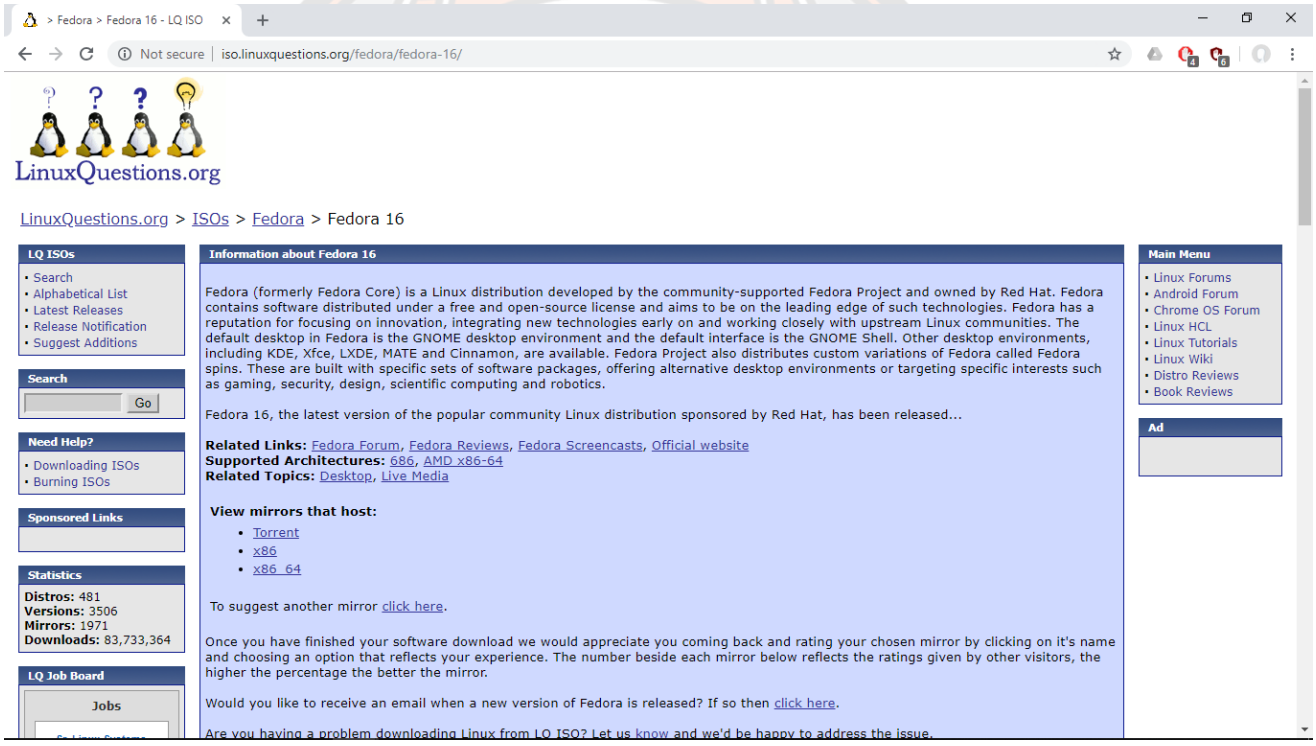
ศัพท์บัญญัติราชบัณฑิตยสถาน. สืบค้นเมื่อ 24 ตุลาคม 2560, จาก http://rirs3.royin.go.th/coinages/webcoinage.php.

อัจจิมา เลี้ยงอยู่ สุธี พงศาสกุลชัย และ พีรพร หมุนสนิท. (2553). ระบบปฏิบัติการ (***Operating Systems)***. กรุงเทพฯ: สำนักพิมพ์ เคทีพี คอมพ์ แอนด์ คอลซัลท์.

# ภาคผนวก

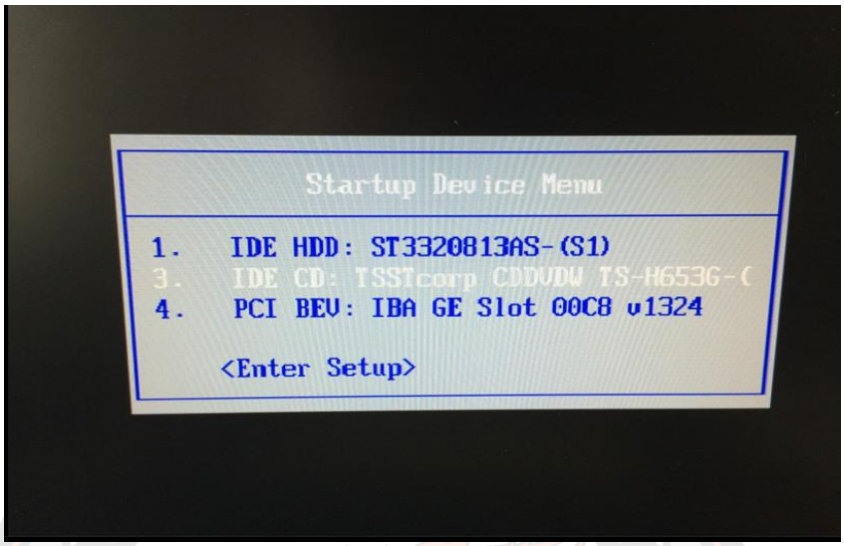
## ภาคผนวก ก การดาวน์โหลด และติดตั้งโปรแกรม Fedora 16

โปรแกรม Fedora (ฟีโดรา) คือ ระบบปฏิบัติการที่แตกแขนงมาจาก Linux และแตกย่อยจาก Redhat เวอร์ชั่น 9 โดย Fedora เป็นระบบปฏิบัติการ Open source ที่ทุกคนสามารถพัฒนาและแก้ไขได้ สามารถดาวน์โหลดได้ที่เว็บไซต์ http://iso.linuxquestions.org/fedora/fedora-16/ แสดงดังภาพที่ 25

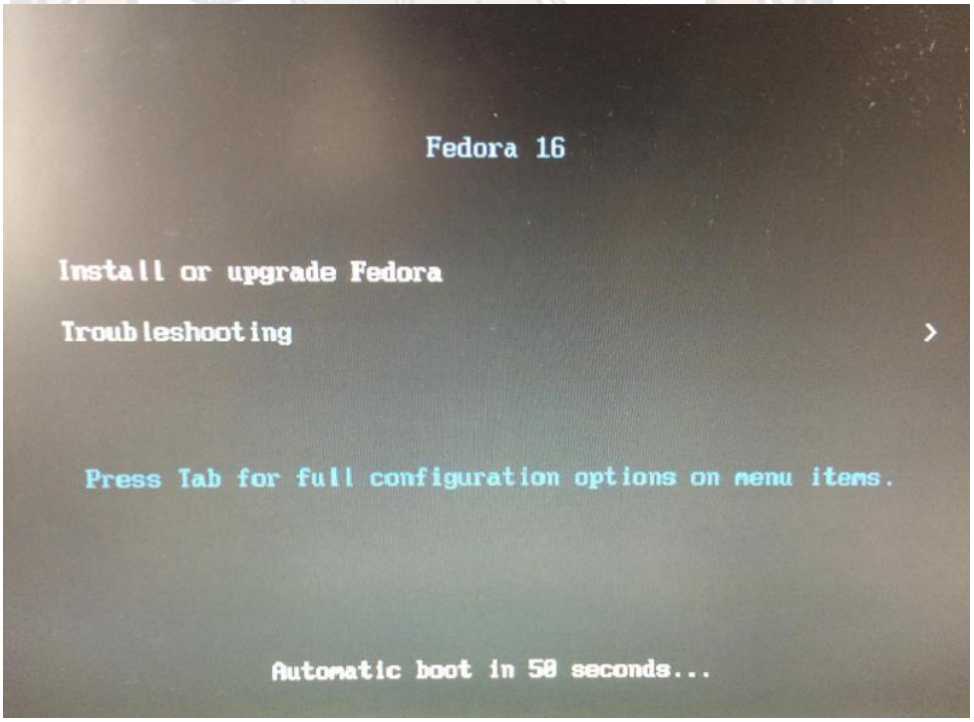



**ภาพที่ 25 แสดงเว็บไซต์สำหรับดาวน์โหลดโปรแกรม Fedora 16**

หลังจากดาวน์โหลดโปรแกรมระบบปฏิบัติการ Fedora 16 แล้ว ให้ restart คอมพิวเตอร์และตั้งค่าบูทสำหรับลงระบบปฏิบัติการ Linux Fedora 16

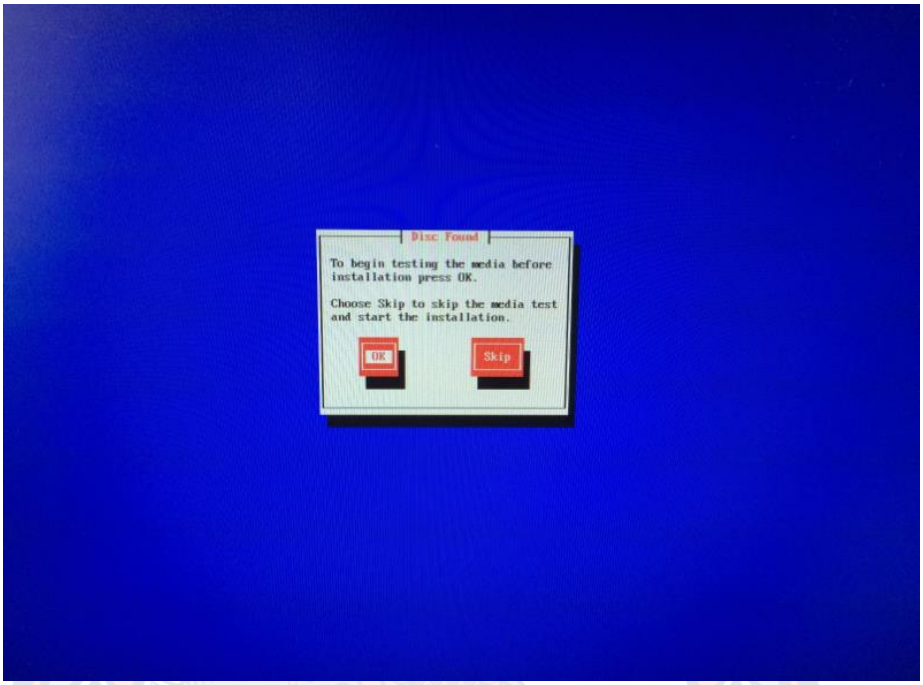



**ภาพที่ 26 แสดงการตั้งค่าบูทสำหรับลงปฏิบัติการ Linux Fedora 16**



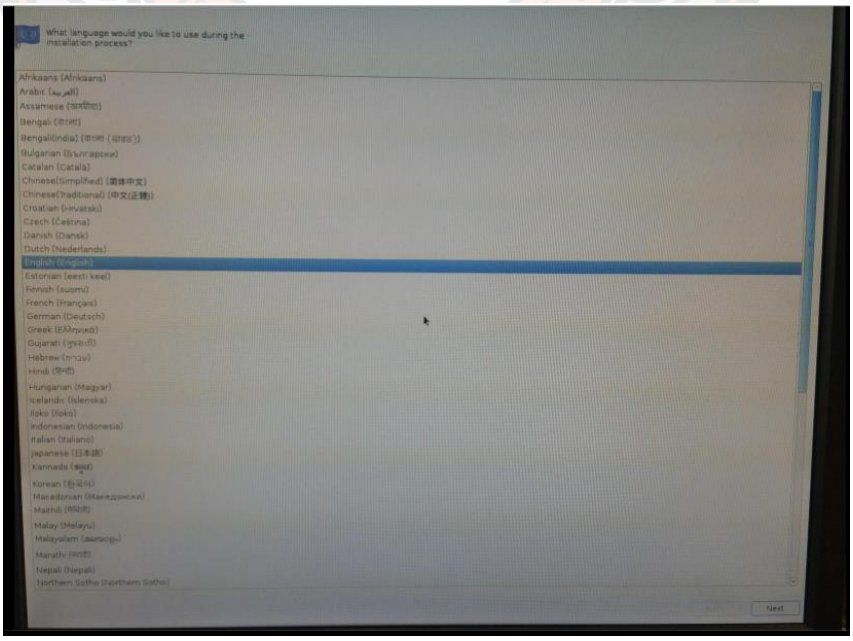

**ภาพที่ 27 แสดงขั้นตอนการติดตั้ง Install Fedora 16**

หลังจากเลือกคำสั่ง Install or upgrade Fedora แล้วจะเข้าสู่การติดตั้งในลำดับถัดไป

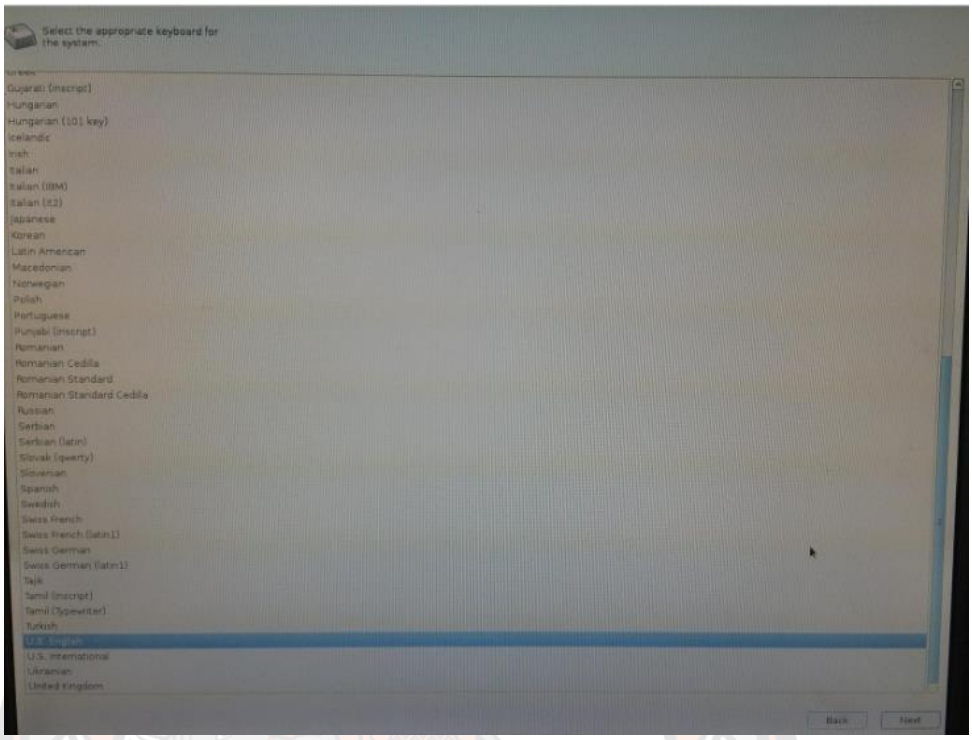



**ภาพที่ 28 แสดงขั้นตอนถัดไปของการติดตั้งระบบปฏิบัติการ Fedora โดยคลิกปุ่ม OK**

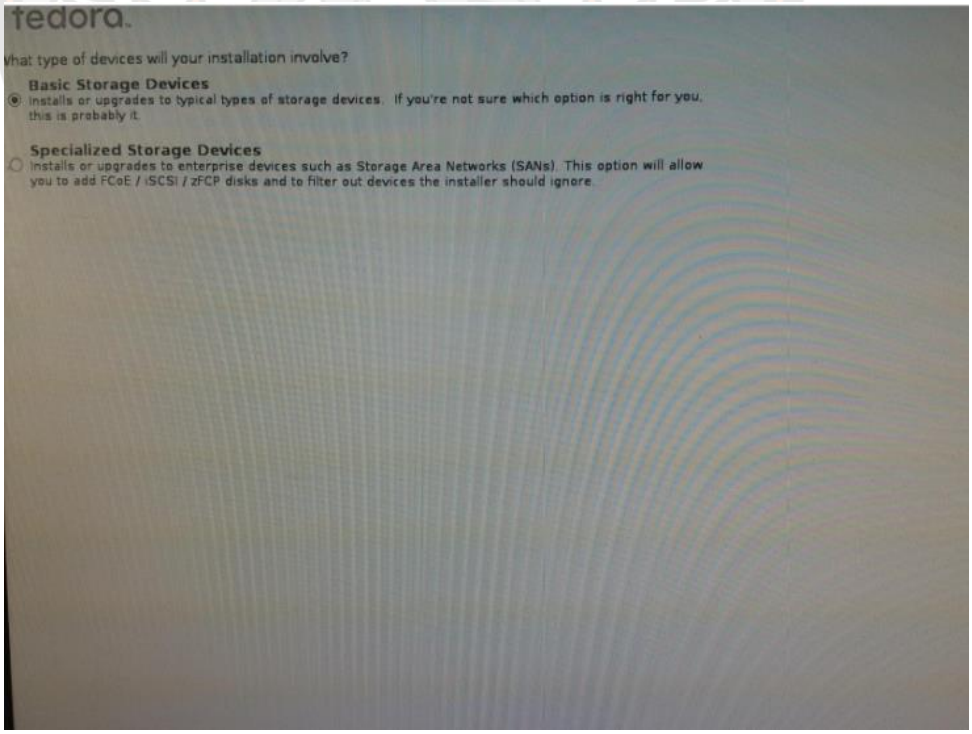
หลังจากคลิกปุ่ม OK เพื่อดำเนินการติดตั้งโปรแกรมจะแสดงหน้าต่างการตั้งค่าภาษา



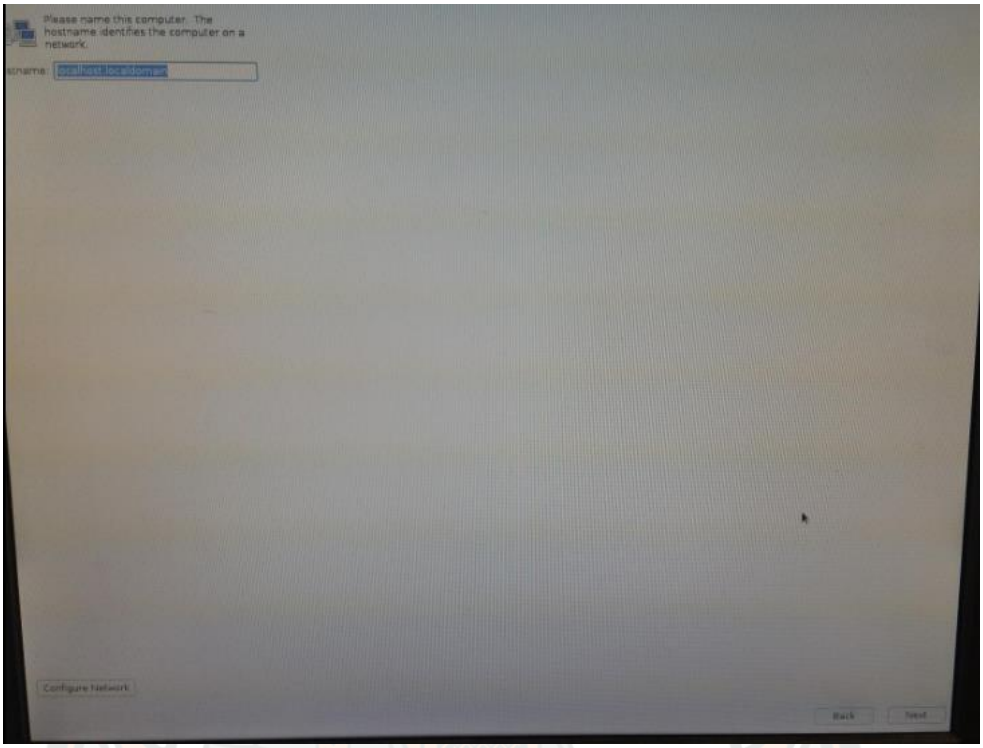

**ภาพที่ 29 แสดงการตั้งค่าภาษาสำหรับลงระบบปฏิบัติการ Linux**



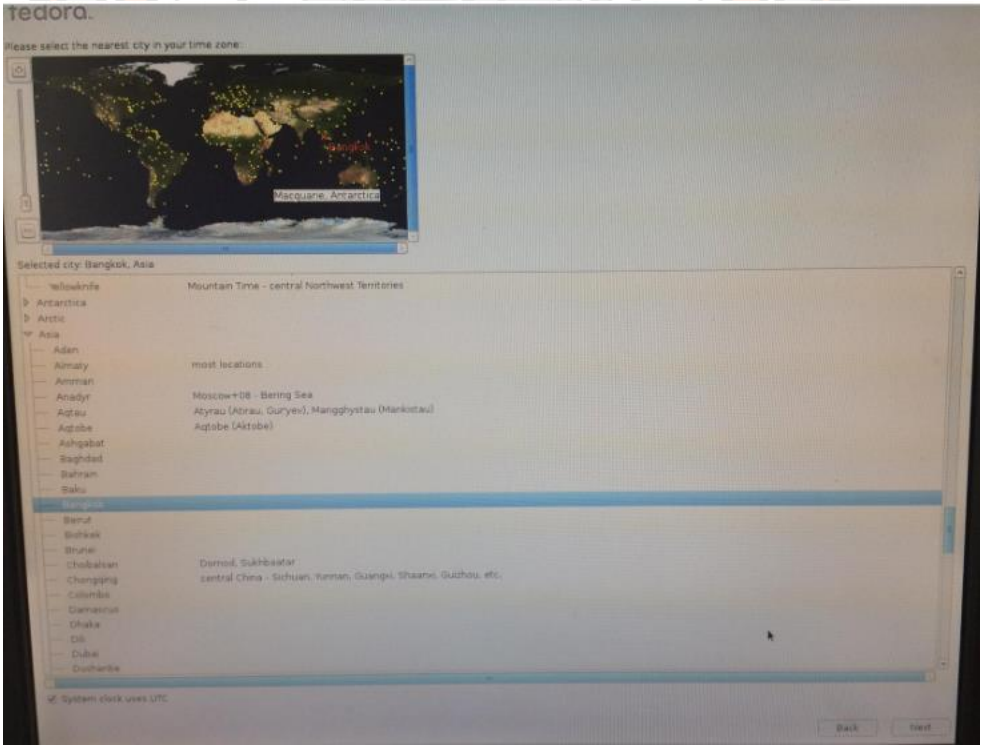

**ภาพที่ 30 ตั้งค่าภาษาสำหรับใช้ระบบปฏิบัติการ Linux**



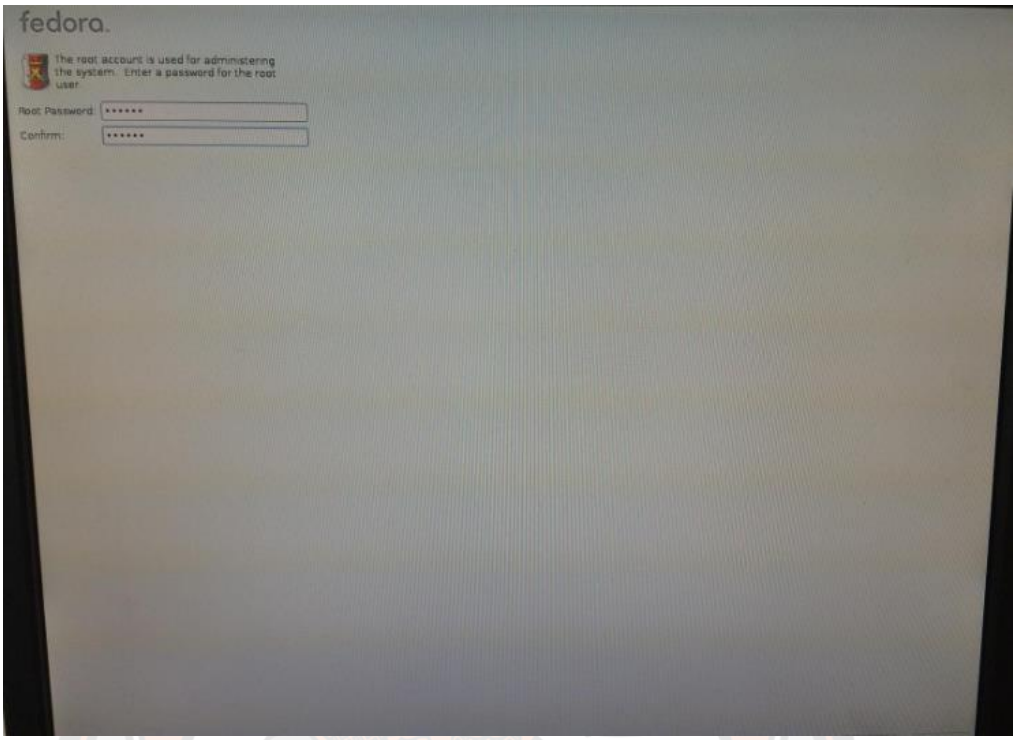

**ภาพที่ 31 เลือก Basic Storage Devices แล้วคลิด Next**



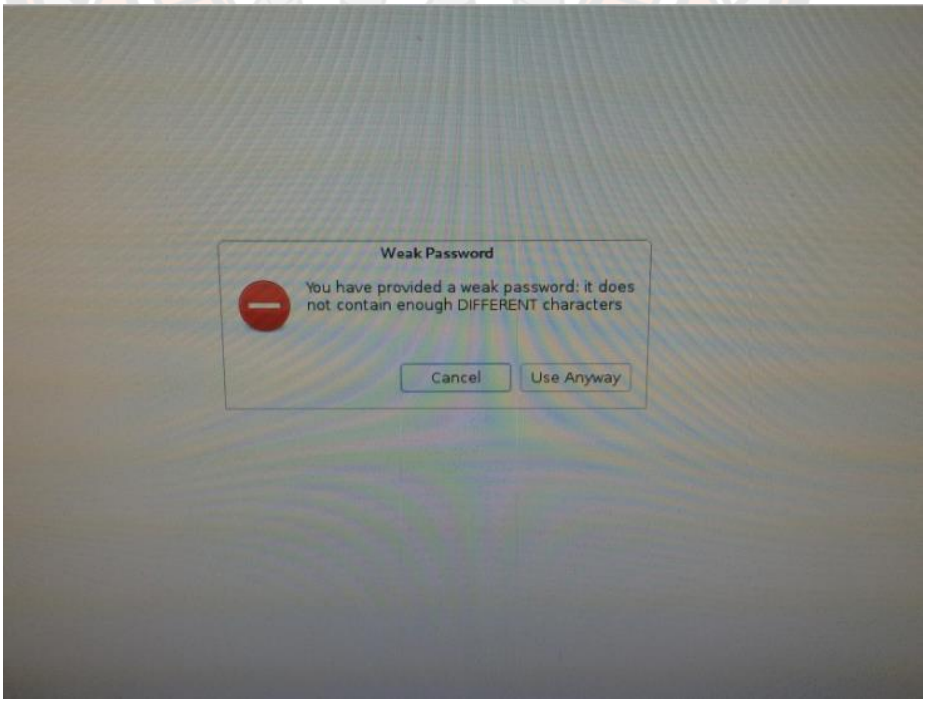

**ภาพที่ 32 ตั้งชื่อสำหรับคอมพิวเตอร์แล้วคลิก Next**



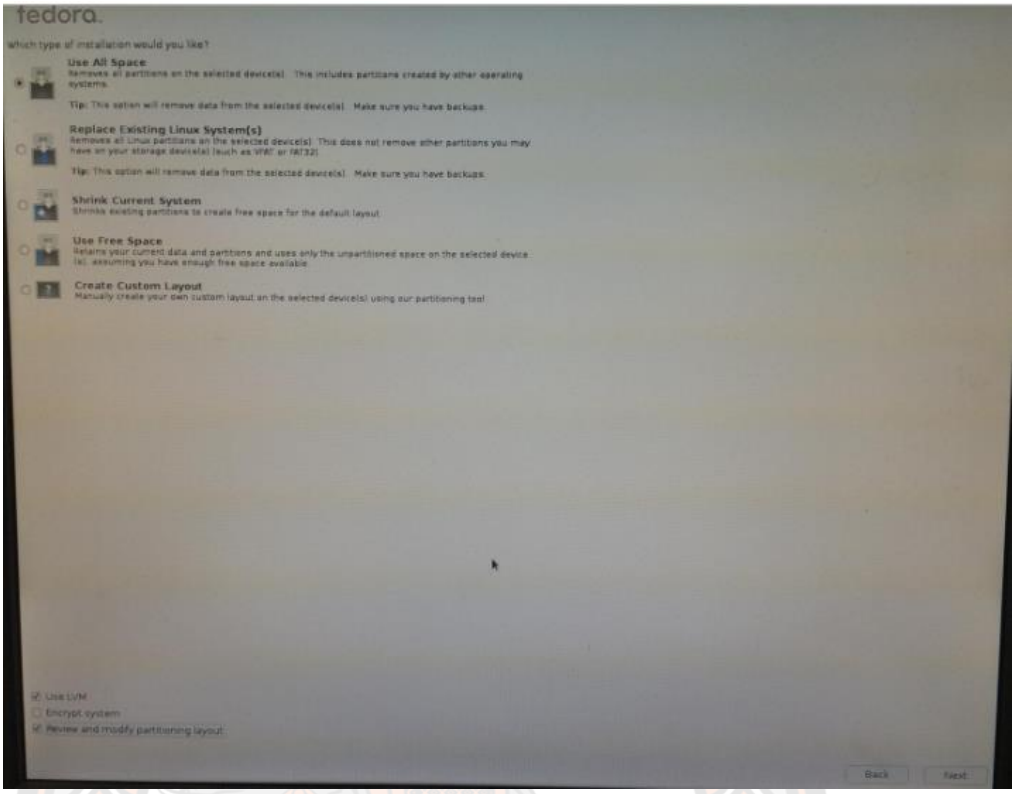

**ภาพที่ 33 เลือกภูมิภาคสำหรับผู้ใช้งาน**



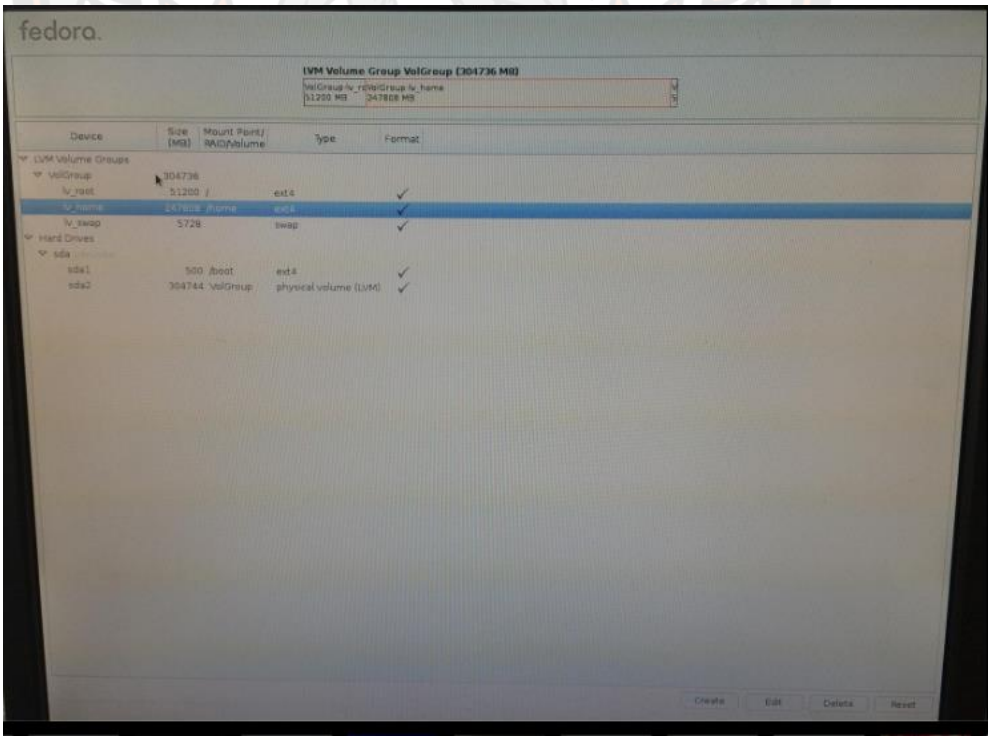

**ภาพที่ 34 ตั้งค่า Root Password สำหรับผู้ใช้**




**ภาพที่ 35 ยืนยัน Password คลิก Use Anyway เพื่อดำเนินขั้นถัดไป**

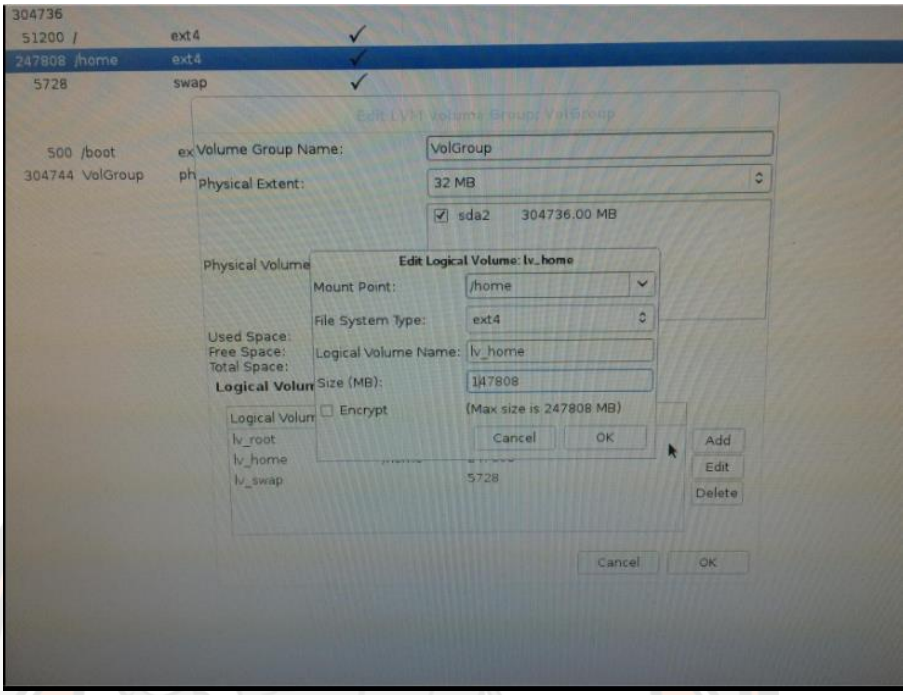



**ภาพที่ 36 คลิกเลือก Use All Space แล้วคลิก Next**

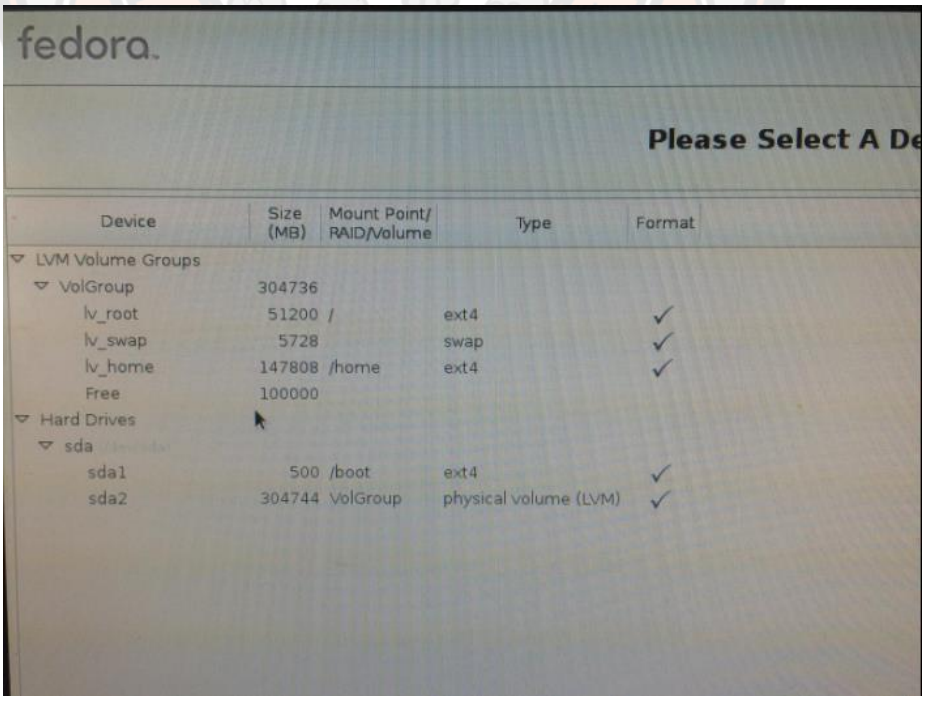



**ภาพที่ 37 จัดสรรพื้นที่สำหรับลงระบบปฏิบัติการ Fedora Linux**

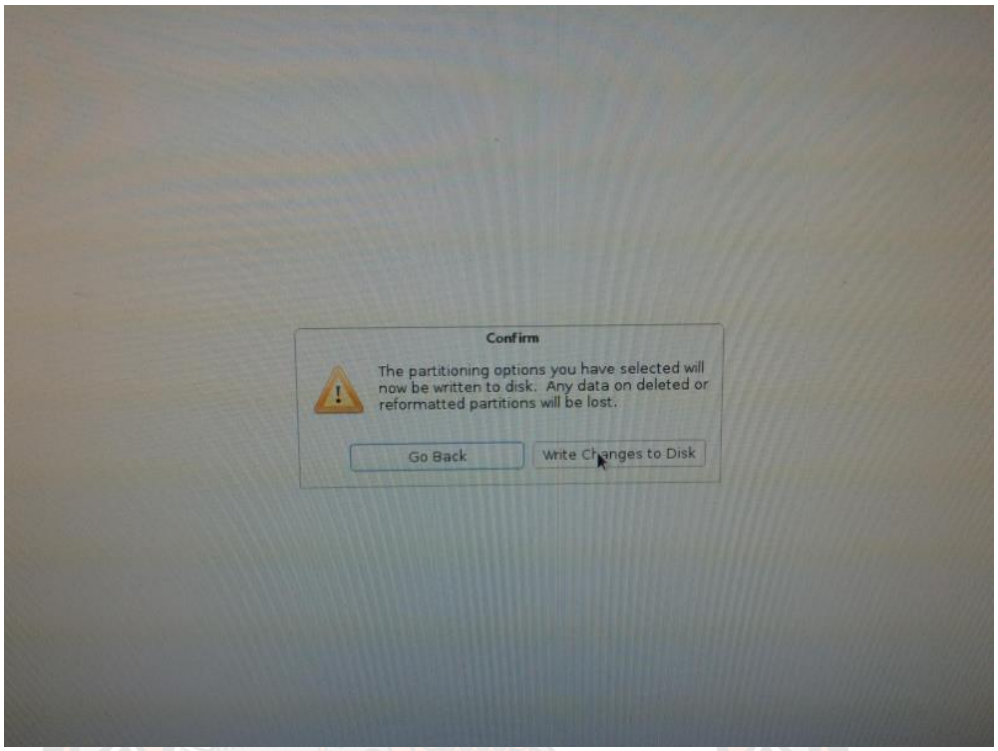
ในการจัดสรรพื้นที่สำหรับลงระบบปฏิบัติการ Fedora Linux ผู้ใช้ต้องคำนวณพื้นที่สำหรับลงโปรแกรมจำลองระบบปฏิบัติการหรือ VM สำหรับสร้างคลาวด์อย่างน้อย 100 GB.



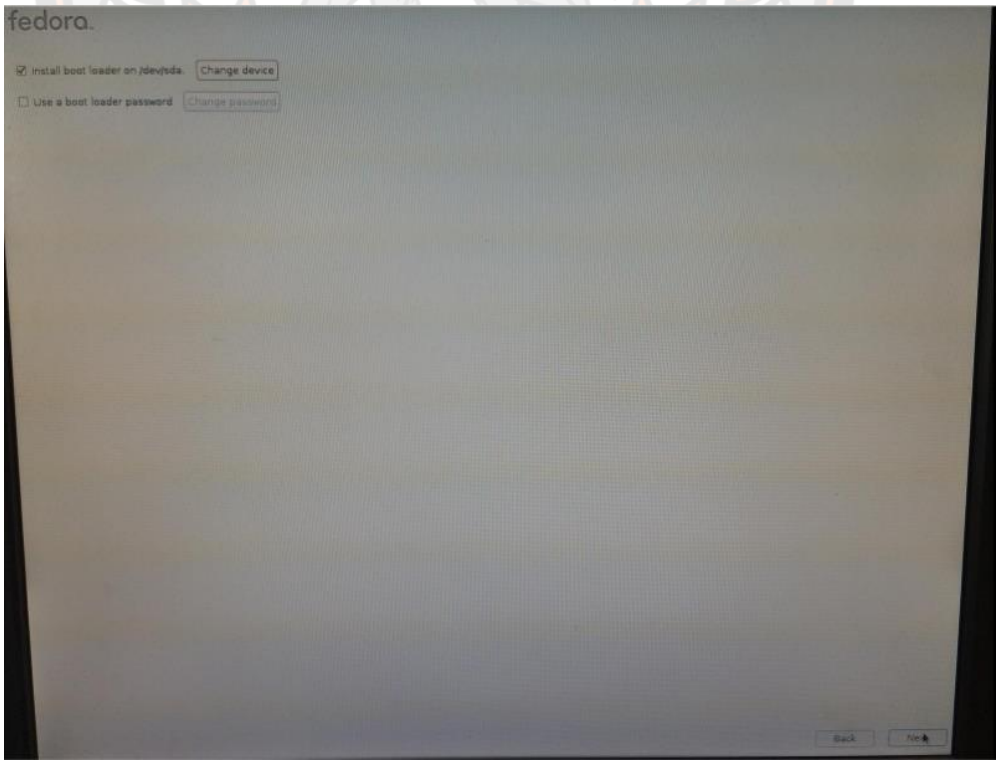

**ภาพที่ 38 แบ่งพื้นที่สำหรับ VM เพื่อใช้ในการสร้างคลาวด์**



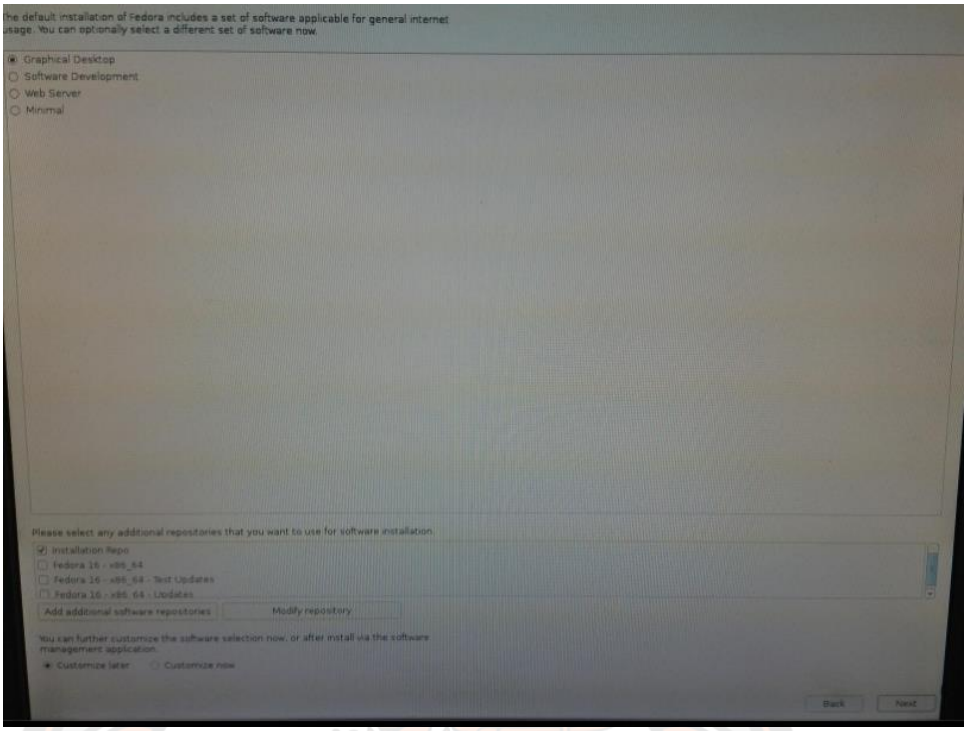

**ภาพที่ 39 แสดงพื้นที่ทั้งหมดสำหรับการลงระบบปฏิบัติ Fedora Linux**




**ภาพที่ 40 ยืนยันการติดตั้งระบบปฏิบัติการ Fedora Linux**

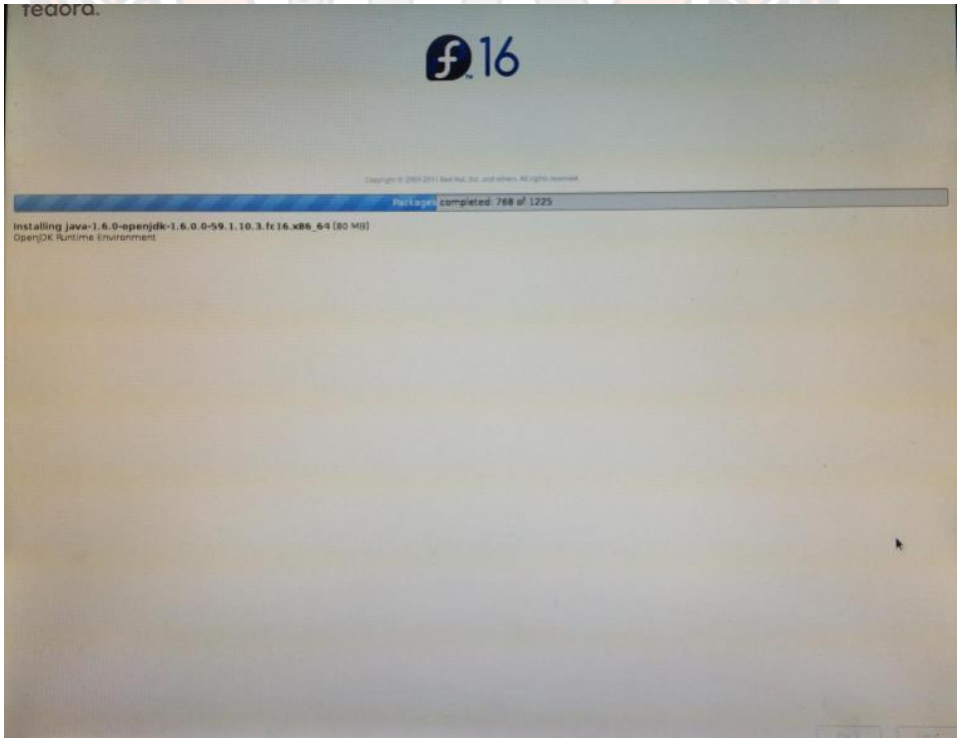



**ภาพที่ 41 คลิก Next เพื่อดำเนินการขั้นตอนถัดไป**

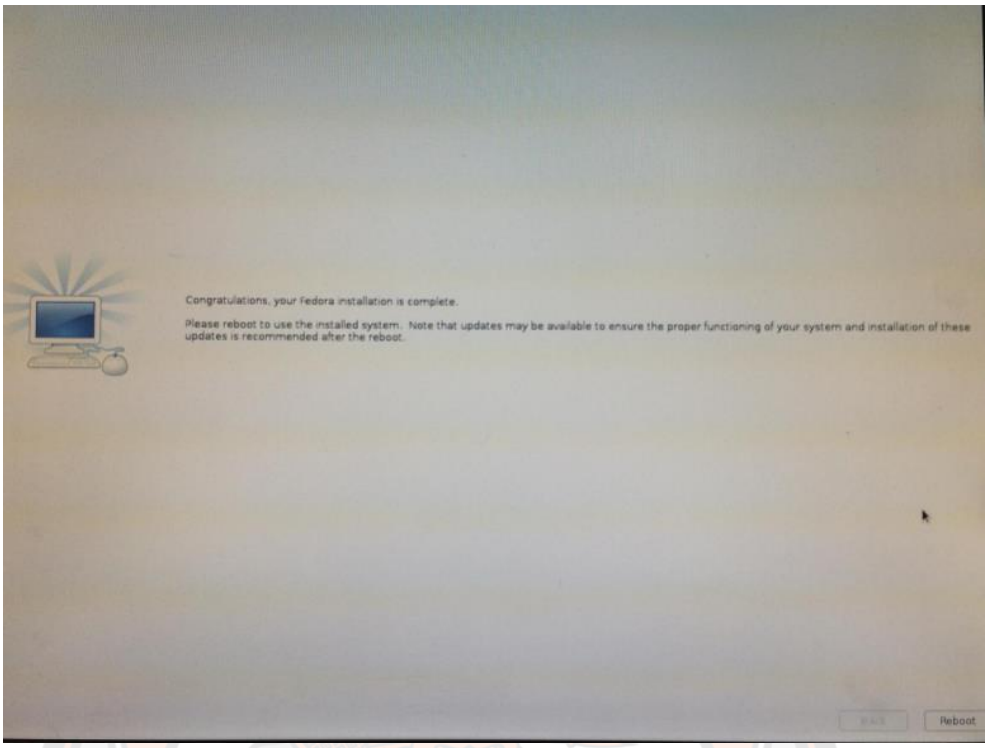



**ภาพที่ 42 เลือก Graphical Desktop และคลิกปุ่ม Next**

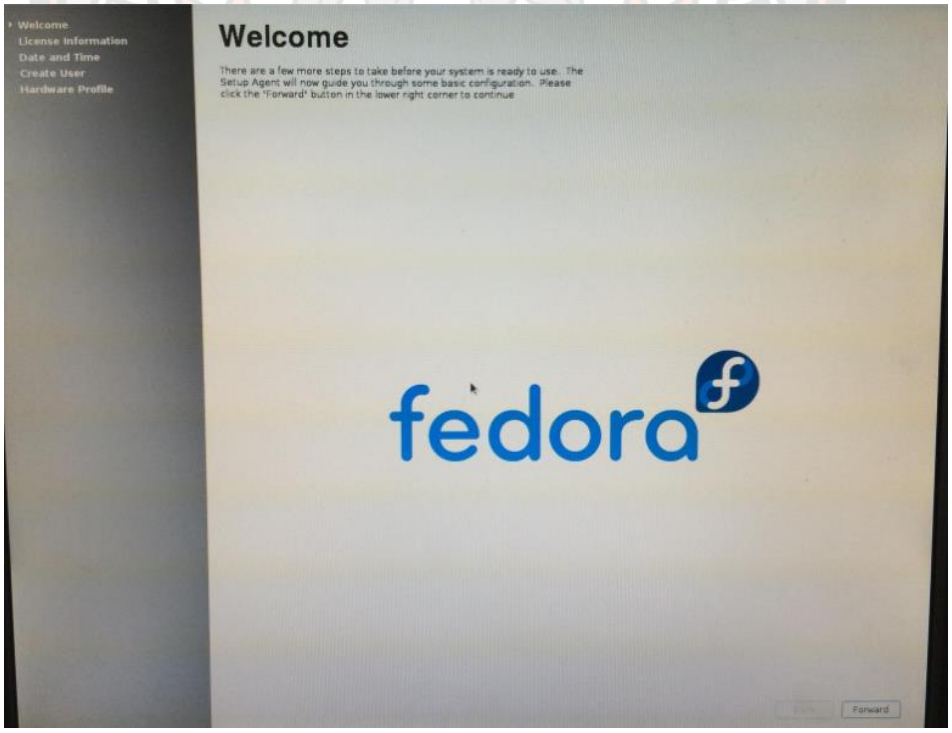
สาเหตุที่เลือก Graphical Desktop เพื่อให้ผู้ใช้งานสามารถใช้งานในโหมดของ Graphical ได้สะดวก



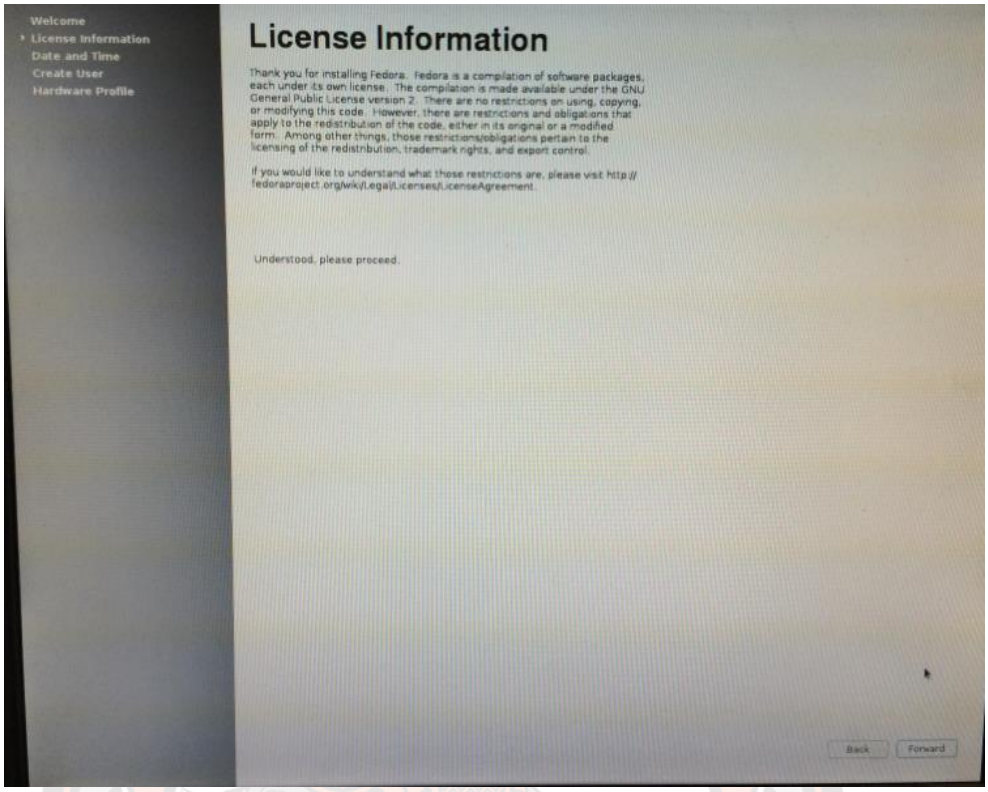

**ภาพที่ 43 โปรแกรมทำการติดตั้ง**



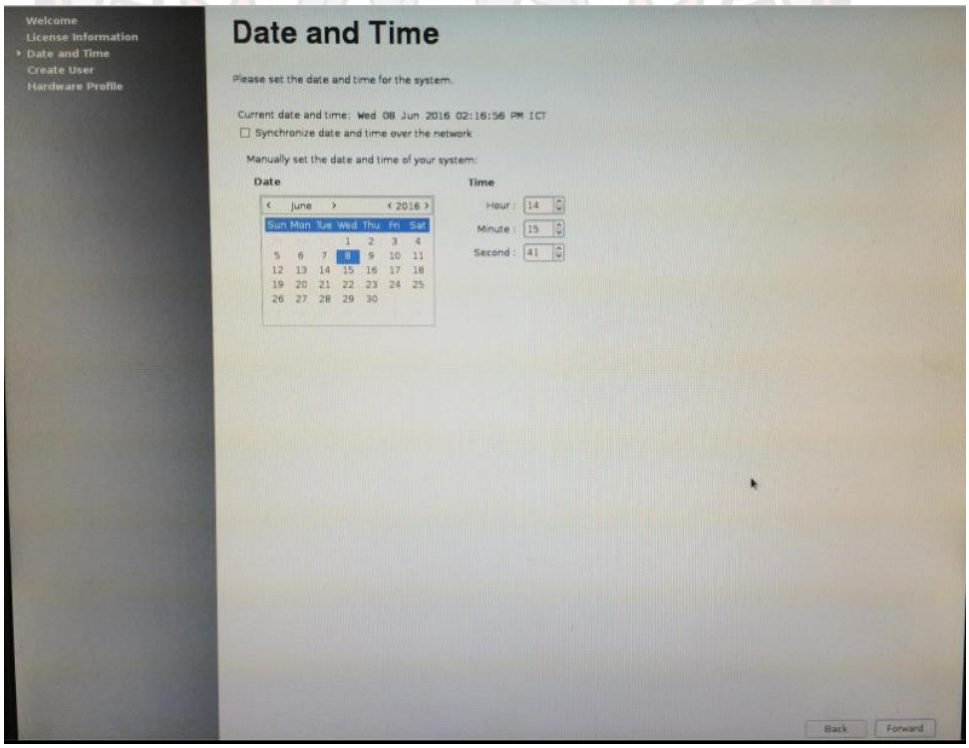

**ภาพที่ 44 หลังจากโปรแกรมติดตั้งเสร็จให้ reboot ระบบปฏิบัติการเพื่อเข้าใช้งานระบบปฏิบัติการ Fedora Linux**



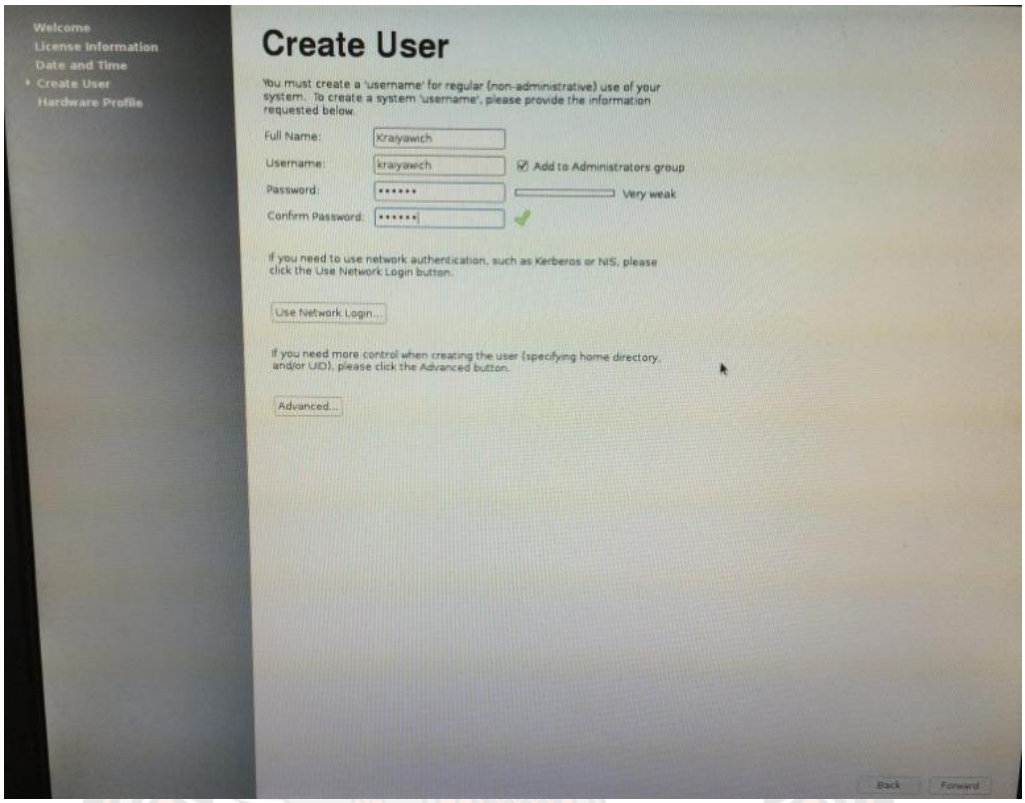

**ภาพที่ 45 เข้าสู่ระบบปฏิบัติการ Fedora Linux**



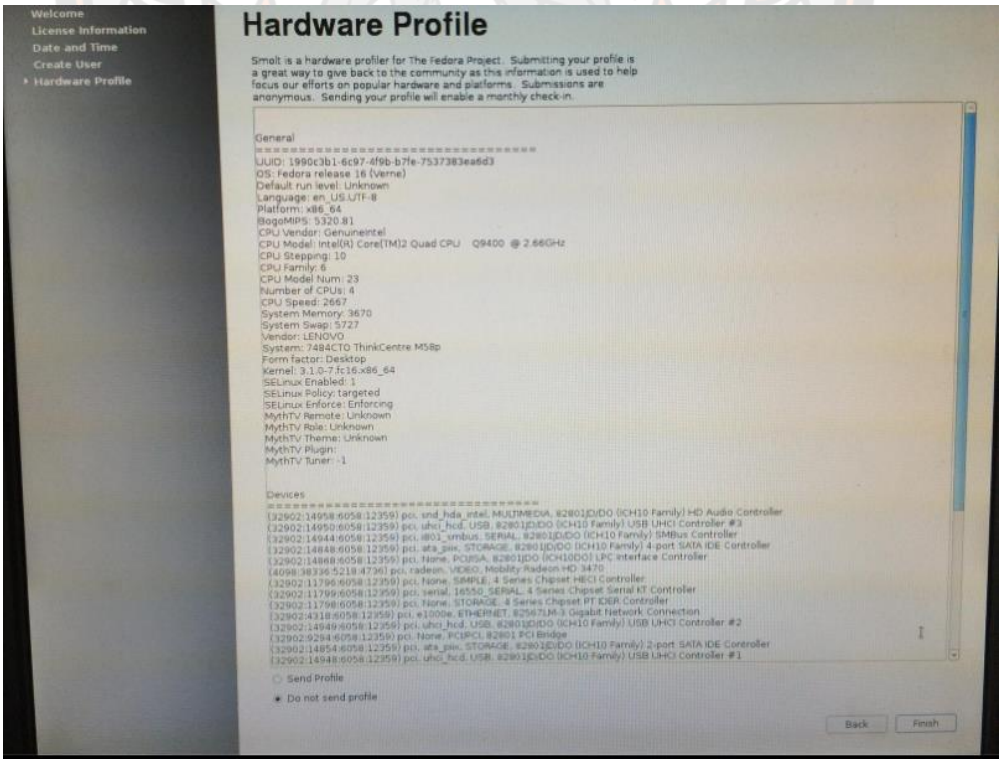

ภาพที่ 46 ยืนยัน License แล้วคลิก Forward



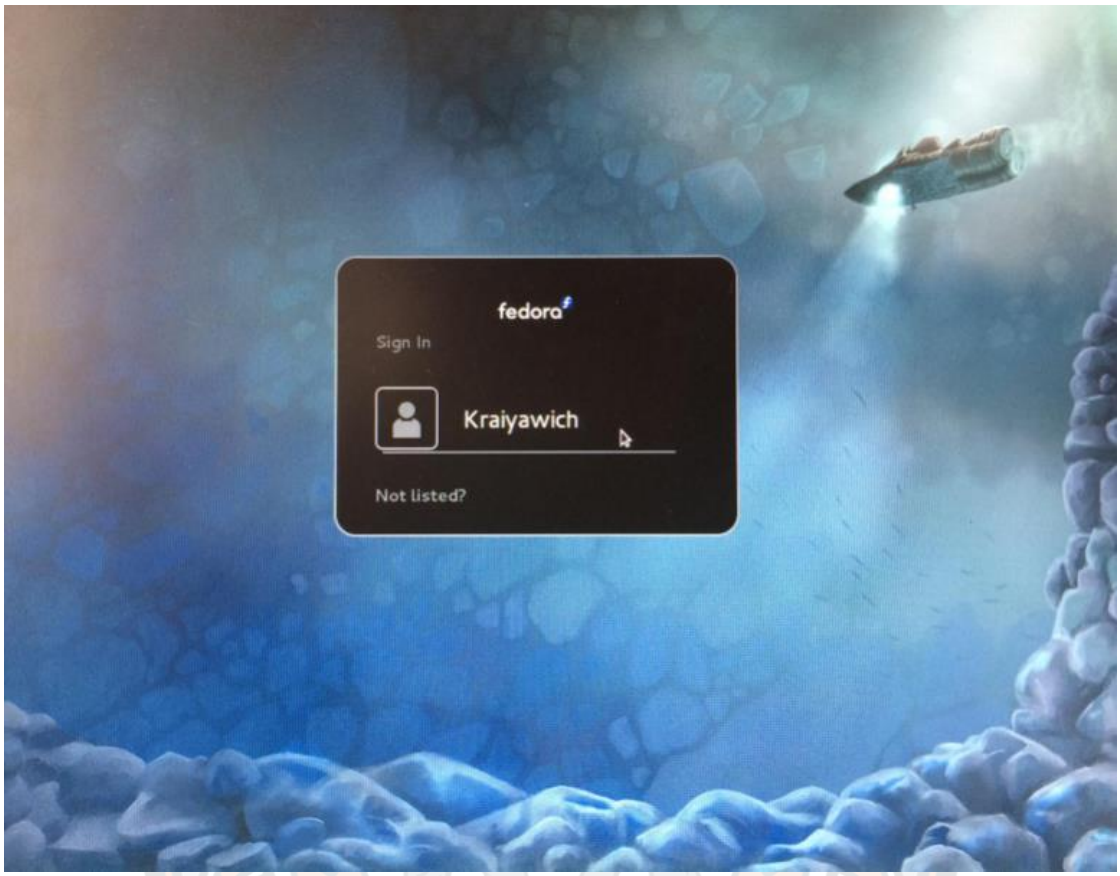

ภาพที่ 47 ตั้งค่าวันและเวลาเพื่อเข้าสู่ระบบปฏิบัติการ Fedora Linux




**ภาพที่ 48 สร้าง User สำหรับผู้ใช้งาน**




**ภาพที่ 49ตรวจสอบสถานะฮาร์ดแวร์และคลิก Finish**




**ภาพที่ 50 ระบบปฏิบัติการ Fedora Linux พร้อใช้งาน**

**ภาคผนวก ข ผลงานตีพิมพ์และเผยแพร่**



# An Analysis of Impacts of Main Memory against Performance of Logging System in the Cloud

Kraiyawich Suppasopapong[1] and Winai Wongthai[2]

[1]Department of computer Science and Information Technology, Faculty of Science, Naresuan University, Phitsanulok, Thailand, 65000

[2]Center of Excellence in Nonlinear Analysis and Optimization, Faculty of Science, Naresuan University, Phitsanulok, Thailand, 65000 winaiw@nu.ac.th

Correspondence Author : Kraiyawich Suppasopapong, 091-8411616

Email : kraiyawichs58@email.nu.ac.th

## บทคัดย่อ

**การประมวลผลแบบกลุ่มเมฆหรือคลาวด์เป็นการให้บริการทรัพยากรทางคอมพิวเตอร์ออนไลน์ เช่น พื้นที่เก็บข้อมูล ระบบปฏิบัติการ แอปพลิเคชัน โครงสร้างพื้นฐานทางคอมพิวเตอร์ และการเข้าถึงบริการดังกล่าวนี้ส่วนมากจะต้องทำผ่านอินเตอร์เน็ต ในปัจจุบันระบบคลาวด์มีการใช้อย่างแพร่หลาย และปัญหาด้านความปลอดภัยก็ยังเป็นปัญหาสำคัญของคลาวด์ในปัจจุบัน เช่น ผู้ไม่หวังดีอาจเข้าถึงไฟล์ข้อมูลของผู้ใช้คลาวด์ได้ ทั้งนี้ระบบบันทึกเหตุการณ์ หรือ logging system เป็นแนวทางหนึ่งที่ช่วยลดปัจจัยเสี่ยงที่อาจเป็นสาเหตุให้เกิดประเด็นด้านความปลอดภัยของคลาวด์ได้ ระบบบันทึกเหตุการณ์ดังกล่าวสามารถบันทึกข้อมูลความเป็นไปในระบบคลาวด์ เช่น บันทึกว่าใครเคยเข้าถึงไฟล์อะไร ข้อมูลที่บันทึกได้นี้สามารถนำไปเป็นหลักฐานเมื่อเกิดปัญหาเกี่ยวกับความปลอดภัยของคลาวด์ แต่อย่างไรก็ตามงานวิจัยก่อนหน้านี้ไม่ได้มีการทดสอบประสิทธิภาพอย่างเต็มรูปแบบเกี่ยวกับส่วนประกอบที่สำคัญของระบบบันทึกเหตุการณ์ดังกล่าว ในบทความนี้จึงวิเคราะห์คุณภาพของส่วนประกอบที่สำคัญของระบบบันทึกเหตุการณ์ ด้วยวิธีการทดสอบประสิทธิภาพของหน่วยความจำหลัก จากนั้นผู้วิจัยได้วิเคราะห์ สรุป และอภิปรายผลกระทบของหน่วยความจำหลักที่มีต่อระบบบันทึกเหตุการณ์ดังกล่าว โดยข้อสรุปของงานวิจัยนี้สามารถเป็นข้อมูลพื้นฐานส่วนหนึ่ง ซึ่งจะทำให้ทราบถึงคุณภาพและสมรรถนะของส่วนประกอบต่างๆ ของระบบบันทึกเหตุการณ์ เพราะข้อสรุปดังกล่าวนี้สามารถนำไปใช้เป็น**

**พื้นฐานส่วนหนึ่งสำหรับการวางแผน ออกแบบ และพัฒนาระบบบันทึกเหตุการณ์ได้อย่างเหมาะสมและมีประสิทธิภาพ ซึ่งจะช่วยให้ระบบดังกล่าวสามารถทำงานได้จริงในระบบคลาวด์ เพื่อเป็นส่วนหนึ่งของวิธีการในการลดปัจจัยเสี่ยงที่อาจเป็นสาเหตุให้เกิดประเด็นด้านความปลอดภัยของคลาวด์**

**คำสำคัญ : ความปลอดภัย การประมวลผลแบบกลุ่มเมฆ ระบบบันทึกเหตุการณ์ การวัดประสิทธิภาพ หน่วยความจำหลัก**

Abstract

Cloud Computing or cloud is a service model that allows consumers access to and use computing recourses provided by cloud providers. Mostly, this needs to be done via the Internet. These resources cloud be storages, virtual machines, computing infrastructures, etc. Although cloud is widely adopted in many application areas, its security is still a challenge. For example, malicious users may have unauthorized accesses to a cloud user's file. A logging system is one of approaches to mitigate risks associated with cloud security. This system can, for example, produce evidence of who was an accessor of a cloud user's file. This evidence can be used when facing cloud security. However, previous works did not provide full performance measurement of the system. This paper discusses quality analysis of a logging system through performance measurement of the system. We mainly for focused on measurement of a main memory of the system. Then to investigate that how the memory affect the performance of the system, we analyzed the results and provided discussions. This paper can be seen as a basis to provide quality analysis of a logging system. This basis can be used as a guide line to design and develop the system effectively and efficiently. Then this system can be truly used to mitigate risks associated with the cloud security.

**Keywords : Security, Cloud, Logging system, Performance Measurement, Main memory**

## 1. บทนำ

คลาวด์เป็นเทคโนโลยีที่มีการเจริญเติบโตอย่างรวดเร็วทางด้านไอทีและถูกนำไปใช้อย่างกว้างขวางในการจัดเก็บข้อมูลและคลาวด์ยังให้บริการทรัพยากรทางคอมพิวเตอร์อื่นๆ ซึ่งรูปแบบการใช้งานของคลาวด์ คือ เสียค่าใช้จ่ายเท่ากับจำนวนทรัพยากรที่ใช้ ดังนั้น คลาวด์จึงเป็นตัวเลือกที่ดีสำหรับองค์กรในการนำคลาวด์มาประยุกต์ใช้งานด้านไอที แต่อย่างไรก็ตามปัญหาด้านความปลอดภัยของคลาวด์ก็เป็นอุปสรรคสำคัญในการนำคลาวด์มาใช้ให้ประสบผลสำเร็จในองค์กรโดยปัญหาด้านความปลอดภัยดังกล่าวนี้เป็นการให้เหตุผลจากเจ้าหน้าที่ CIOs (Chief Information Officers) ซึ่งเป็นเจ้าหน้าที่ด้านไอทีขององค์กรต่างๆ [4] และด้วยเหตุผลข้างต้นที่มีเจ้าหน้าที่ดังกล่าวกังวลเกี่ยวกับด้านความปลอดภัยของคลาวด์ ทั้งนี้จึงมีองค์กรได้ทำการวิจัยและระบุปัญหาภัยคุกคามของคลาวด์ไว้ในรายงานต่างๆ เช่น องค์กร CSA (Cloud Security Alliance) [1]

จากปัญหาภัยคุกคามของคลาวด์ที่องค์กร CSA ได้ระบุไว้ทำให้มีนักวิจัยหลายท่านได้ทำการวิจัยเกี่ยวกับแนวทางการบรรเทาปัญหาดังกล่าว ในบทความนี้ผู้วิจัยได้ศึกษาและสนใจเกี่ยวกับระบบการบรรเทาภัยคุกคามด้วยระบบบันทึกเหตุการณ์ (logging systems) ซึ่งเป็นระบบที่ใช้ในงาน [5][10][11] โดยในระบบดังกล่าวสามารถบันทึกข้อมูลความเป็นไปในระบบคลาวด์ เช่น บันทึกว่าใครเคยเข้าถึงไฟล์อะไร ข้อมูลที่บันทึกนี้สามารถนำไปเป็นหลักฐานเมื่อเกิดปัญหาเกี่ยวกับความปลอดภัยของคลาวด์ โดยในระบบบันทึกเหตุการณ์จะมีโปรเซสที่ใช้บันทึกเหตุการณ์ เรียกว่า logger ในบทความนี้จะมุ่งเน้นไปที่การทำงานของ logger และทำการทดสอบ RAM ซึ่งเป็นหน่วยความจำหลัก (รายละเอียดของ RAM จะถูกอธิบายในย่อหน้ารองสุดท้ายของบทนำ) ซึ่ง RAM เป็นส่วนประกอบที่สำคัญในระบบนี้ ผู้วิจัยจึงสนใจว่าหากมีการเพิ่มจำนวน RAM จะส่งผลกระทบต่อประสิทธิภาพของ logger หรือไม่ โดยแบ่งการทดสอบเป็น 2 ส่วน คือ 1. เครื่องที่ทำการติดตั้ง logger โดยในบทความเรียกเครื่องที่ติดตั้ง logger ว่า dom0 (Domain Zero) 2. เครื่องที่ถูกตรวจสอบ ในบทความจะเรียกเครื่องที่ถูกตรวจสอบว่า domU (Domain User) ในงานเดิม [5][10] ได้มีการทดสอบประสิทธิภาพเกี่ยวกับ CPU core (รายละเอียดของ CPU core จะถูกอธิบายในย่อหน้าที่สี่ของบทนำ) ที่ส่งผลต่อ

ประสิทธิภาพการทำงานของ logger แต่ยังไม่มีผลการทดลองใดที่ทำการทดสอบว่า RAM จะส่งผลต่อประสิทธิภาพการทำงานของ logger หรือไม่

การทดสอบประสิทธิภาพของซอฟต์แวร์และฮาร์ดแวร์ถือว่าเป็นเรื่องสำคัญ [2] เพราะการทดสอบประสิทธิภาพเป็นเรื่องพื้นฐานที่สำคัญ จะทำให้ทราบถึงความสามารถของระบบคอมพิวเตอร์ทั้งในด้านซอฟต์แวร์และฮาร์ดแวร์ [13] ดังนั้นในบทความนี้จึงมุ่งเน้นไปที่การทดสอบประสิทธิภาพของ logger เมื่อมีการปรับขนาด RAM ใน dom0 และ domU โดยผู้วิจัยออกแบบการทดลองให้ระบบ logger ทำงานบน RAM ในสภาวะที่แตกต่างกัน และนำผลการทดลองที่ได้มาวิเคราะห์และสรุป หากเพิ่ม RAM ให้กับ dom0 และ domU จะส่งผลต่อประสิทธิภาพของ logger อย่างไร

หน่วยประมวลผล คือ หน่วยที่แปลคำสั่งแล้วทำการประมวลคำสั่งเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ ได้แก่ CPU และ Coprocessor ในบทความนี้ขออธิบาย CPU เพียงอย่างเดียวเพราะว่า CPU เป็นส่วนที่ใช้ทดลองในงานวิจัยนี้ CPU (Central Processing Unit) หรือ CPU core ซึ่งในงานวิจัยใช้คำว่า CPU core โดย CPU core มีความสามารถในการประมวลผลและรองรับการใช้งานซอฟต์แวร์ประยุกต์ต่างๆ [6][12] เป็นส่วนสำคัญในการประมวลผลของเครื่อง

หน่วยความจำหลัก (Main memory) เป็นหน่วยความจำที่เก็บข้อมูลหรือคำสั่งที่รับเข้ามาเพื่อรอให้ CPU ประมวลผล หน่วยความจำหลักจะแบ่งเป็น 2 ประเภท ได้แก่ ROM และ RAM ในบทความนี้สนใจการทำงานของ RAM จึงขออธิบายความหมายของ RAM เพียงอย่างเดียว RAM ย่อมาจาก Random Access Memory เป็นหน่วยความจำชั่วคราว เมื่อไม่มีกระแสไฟฟ้า ข้อมูลที่อยู่ใน RAM ก็จะหายไป ซึ่ง RAM จะเก็บข้อมูลในระหว่างที่คอมพิวเตอร์กำลังทำงาน [3][9]

บทความนี้จะทำการทดสอบประสิทธิภาพของ logger ในสภาวะ RAM ขนาดต่างๆ ผู้วิจัยจึงทำการออกแบบการทดลองโดยกำหนด RAM ให้กับ dom0 domU และทดสอบประสิทธิภาพของ logger โดยการทดสอบของ [5][10] ก่อนหน้าจะเป็นการทดสอบประสิทธิภาพของ CPU core ที่ส่งผลต่อประสิทธิภาพของ logger แต่ยังไม่มีการทดสอบประสิทธิภาพของ RAM ที่ส่งผลต่อประสิทธิภาพของ logger ผู้วิจัยจึงเล็งเห็นว่า RAM ก็เป็นส่วนสำคัญที่จะส่งผลต่อการทำงานของ logger จึงได้ทำการทดลองนี้ขึ้นมา

## 2. วัตถุประสงค์

ขั้นแรก ผู้วิจัยออกแบบการทดลองและทดลองซ้ำไปมา วัตถุประสงค์ในการทดลองเพื่อทดสอบประสิทธิภาพของ logger หลังการเพิ่ม RAM และนำผลการทดลองของ dom0 และ domU มาวิเคราะห์ผลกระทบของ RAM ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ ในการทดลองผู้วิจัยตั้งค่า RAM ให้ dom0 และ domU สำหรับการทดสอบประสิทธิภาพของ logger ในสภาพแวดล้อมที่แตกต่างกัน ดังนั้น เมื่อทราบเกี่ยวกับประสิทธิภาพของ logger ในสภาพแวดล้อม RAM ที่แตกต่างกันแล้ว จะทำให้สามารถวิเคราะห์ผลกระทบของ RAM ต่อประสิทธิภาพของระบบบันทึกเหตุการณ์และทำให้ผู้ออกแบบระบบ logger สามารถออกแบบระบบได้ง่ายขึ้นเพราะจะสามารถเลือกจัดสรรทรัพยากรได้อย่างเหมาะสม จนสามารถลดต้นทุนขององค์กรและประหยัดพลังงานได้

## 3. แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 3.1 นำเสนอสถาปัตยกรรมของระบบบันทึกเหตุการณ์

ในหัวข้อนี้จะบรรยายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์บนคลาวด์และกระบวนการทำงานของ read แอปพลิเคชัน โดยในภาพที่ 1 จะแสดงสถาปัตยกรรมของระบบบันทึกเหตุการณ์ กระบวนการทำงานของ read แอปพลิเคชัน ในภาพจะมีไฟล์สำคัญชื่อ s.txt ซึ่งถูกจัดเก็บใน diskU และ diskU จะอยู่ใน domU (domU และ diskU เป็น virtual disk) diskU คือ พื้นที่จัดเก็บไฟล์สำคัญของลูกค้า (ลูกค้าคือผู้ที่มาเช่าบริการคลาวด์และเป็นเจ้าของ domU) การจัดเก็บใน diskU จะเป็นไฟล์ประเภทอะไรก็ได้ เช่น text executable หรือ database files ในภาพที่ 1 จะแบ่งการทำงานเป็น 2 ส่วน คือ domU และ dom0 ส่วนที่ 1 ส่วนของ domU ลูกค้าเป็นเจ้าของและจัดเก็บไฟล์ s.txt ส่วนนี้มีการทำงาน คือ เมื่อลูกค้ารันแอปพลิเคชัน read เพื่อเข้าไปอ่านไฟล์ s.txt การรันแอปพลิเคชัน read จะมาจากลูกค้า หรือว่า domU ถูกครอบครองโดยแฮกเกอร์ (ตามเส้นประหมายเลข 2) (read เป็นแอปพลิเคชัน เมื่อ read ถูกรันแอปพลิเคชัน จะถูกเปลี่ยนเป็นโปรเซส read ) ในส่วนของ read_mem ในวงรีที่ถูกจัดเก็บใน memU (memU คือ หน่วยความจำหลักหรือ RAM) ในพื้นที่หน่วยความจำนี้จะเก็บข้อมูลการทำงานทั้งหมดตั้งแต่รันแอปพลิเคชัน read เพื่อใช้งาน จนกระทั่งเข้าไปอ่านไฟล์ s.txt ซึ่งกระบวนการเหล่านี้จะเรียกว่า การบันทึกประวัติของไฟล์สำคัญ ซึ่งจะถูกจัดเก็บไว้ใน read_mem โดยจะจัดเก็บ 1. ชื่อไฟล์ (f_nm) 2. ไอดีของโปรเซสที่ถูกรัน (p_id) 3. ชื่อของโปรเซสที่ถูกรัน (p_nm) 4. ชื่อผู้รันโปรเซส (p_ownID) แสดงในตารางที่ 1

| f_nm | p_id | p_nm | p_ownId |
|---|---|---|---|
| s.txt | 4624 | read | 1002(alice) |

ตารางที่ 1 The content of the history of critical file

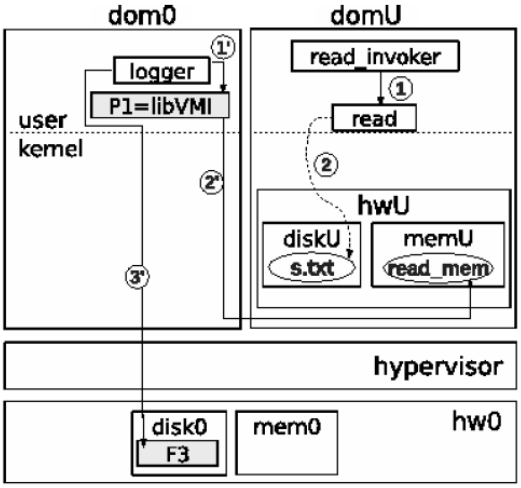
ในย่อหน้านี้ได้กล่าวถึงสถาปัตยกรรมของระบบบันทึกเหตุการณ์บนคลาวด์ กระบวนการทำงานของ read แอปพลิเคชัน และได้กล่าวถึงการทำงานของระบบบันทึกเหตุการณ์ว่าแบ่งการทำงานเป็น 2 ส่วน โดยการทำงานในส่วนที่ 1 ส่วนของ domU จะถูกอธิบายไว้ที่ย่อหน้าที่ผ่านมา และย่อหน้าถัดไปจะอธิบายการทำงานในส่วนที่ 2 ส่วนของ dom0

จากย่อหน้าที่แล้วได้อธิบายการทำงานของระบบบันทึกเหตุการณ์จะถูกแบ่งเป็น 2 ส่วน คือ domU และ dom0 ในย่อหน้านี้จะขออธิบายการทำงานในส่วนที่ 2 คือ การทำงานใน dom0 ซึ่งเป็นส่วนของผู้ให้บริการเช่าคลาวด์ แสดงในภาพที่ 1 โดยในฝั่งของ dom0 จะทำการติดตั้ง logger (หมายเลข 1’ แสดงในภาพที่ 1) สำหรับบันทึกเหตุการณ์ โดยกระบวนการทำงานของ logger คือ รัน logger จากนั้น logger เรียกใช้ libVMI เพื่อเข้าไปใน memU (หมายเลข 2’) ซึ่ง libVMI คือ library ใน ภาษา C ที่ถูกเขียนเพื่อให้สามารถเข้าไปอ่านข้อมูลที่จัดเก็บใน read_mem ที่อยู่ใน memU (ข้อมูลที่ถูกจัดเก็บใน read_mem จะแสดงในตารางที่ 1 และถูกอธิบายในส่วนของบรรทัดสุดท้ายของย่อหน้าที่แล้ว) เมื่อ logger เข้าไปบันทึกข้อมูลใน read_mem ที่อยู่ใน memU แล้ว logger จะนำข้อมูลที่ถูกจัดเก็บใน read_mem (ข้อมูลที่จัดเก็บใน read_mem ดูได้จากตารางที่ 1) บันทึกลงใน F3 (disk0)(หมายเลข 3’)

ภาพรวมทั้งหมดของหัวข้อนี้จะอธิบายการทำงานในส่วนของ domU และ dom0 โดยจะมีการกล่าวถึงการทำงานของแอปพลิเคชัน read ซึ่งในงานวิจัยจะเรียกการทำงานของแอปพลิเคชัน read ว่า กิจกรรมในแอปพลิชัน read ซึ่งหัวข้อถัดไปจะอธิบายเกี่ยวกับกิจกรรมแอปพลิเคชัน read

### 3.2 **กิจกรรมในแอปพลิชัน read**

ในหัวข้อนี้จะอธิบายเกี่ยวกับกิจกรรมในแอปพลิชัน read โดยจะมีกระบวนการทำงาน คือ 1. เปิดไฟล์ s.txt 2. อ่านไฟล์และพิมพ์ข้อมูลภายใน s.txt 3. ปิดการทำงาน 4. จบการทำงาน โดย




ภาพที่ 1 An Experimental Environment domU and dom0

**3.3 สลีปปิ้งทามและแอคคูเรซี่ (sleeping time and accuracy)**

ในหัวข้อนี้จะอธิบายความหมายของ sleeping time และ accuracy ซึ่งเป็นค่าที่ถูกใช้ในการทดลองโดย sleeping time คือ ค่าเวลาค่าหนึ่งที่กำหนดเพิ่มในกระบวนการทำงานของกิจกรรมในแอปพลิเคชัน read (โดยกิจกรรมในแอปพลิเคชัน read ถูกกล่าวไว้ในหัวข้อ 3.2) ผู้วิจัยจึงทำการแก้ไขกระบวนการในกิจกรรมของแอปพลิเคชัน read ในส่วนของก่อนการปิดโปรแกรมด้วยภาษา C โดยเขียน function usleep เพิ่มเข้าไป โดยมีการทำงาน ดังนี้ 1. เปิดไฟล์ s.txt 2. อ่านไฟล์และพิมพ์ข้อมูลใน s.txt 3. ในส่วนของก่อนการปิดการทำงาน จะกำหนดเวลา x ms (millisecond) เช่น x = 60 เพื่อไม่ให้โปรเซสปิดการทำงานทันทีแต่จะให้ปิดการทำงานหลัง 60 ms (การกำหนดเวลาเพิ่มในขั้นตอนนี้เรียกว่า sleeping time) 4. ปิดการทำงาน 5. จบการทำงาน วิธีการทั้งหมดนี้เป็นการ รันแอปพลิเคชัน read เพียง 1 ครั้ง ในการทดลองจะทำการรันแอปพลิเคชัน read 1000 ครั้ง

คำว่า accuracy ในการทดลองผู้วิจัยได้ทดสอบรัน logger ไปจับโปรเซส read ขณะที่กำลังอ่านไฟล์ชื่อ s.txt จำนวน 1000 ครั้ง หาก logger สามารถตรวจพบไฟล์ s.txt ได้ครบ 1000 ครั้ง

หมายความว่า logger มีค่าความแม่นยำ 100% (accuracy = 100%) ค่าความแม่นยำของ logger สามารถมีได้ตั้งแต่ 0-100% ถ้าหาก logger ไม่สามารถตรวจพบไฟล์ s.txt ได้เลย นั้นคือค่าความแม่นยำของ logger เป็น 0%

ในการทดลอง ถ้าหาก logger ไม่สามารถตรวจพบไฟล์ s.txt ได้จะเรียกว่า “miss” และถ้าหาก logger สามารถตรวจพบได้จะเรียก “hit” โดยคำว่า “miss” อาจเกิดจากโปรแกรมเออเร่อทำให้ logger ตรวจจับไฟล์ “miss” เช่น เมื่อรัน logger ไปจับโปรเซส read ขณะที่กำลังอ่านไฟล์ s.txt จำนวน 100 ครั้ง และ logger ตรวจเจอไฟล์ s.txt ได้ 80 ครั้ง (“hit 80”) นั้นหมายความว่า logger มีค่าความแม่นยำที่ 80% (accuracy = 80%)

หัวข้อนี้จะอธิบายเกี่ยวกับค่าของ sleeping time ซึ่งเป็นค่าที่ถูกนำไปใช้ในงานทดลองเพื่อหา accuracy ในการทดลองและค่า accuracy เป็นค่าที่ใช้วัดประสิทธิภาพของ logger โดยประสิทธิภาพของ logger ที่ดี จะมีค่า accuracy ใกล้เคียง 100%

### 3.4 สภาพแวดล้อมการทดลอง

หัวข้อนี้จะกล่าวถึงสภาพแวดล้อมของฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการทดลอง โดยในการทดลองผู้วิจัยได้ทดสอบบนเครื่องคอมพิวเตอร์ PC CPU Xeon และ RAM kington DDR3 SDRAM การออกแบบการทดลองนั้นได้กำหนดค่า CPU core ของ dom0 และ domU ให้มีขนาด 8 core จากนั้นกำหนด RAM ฝั่ง dom0 ให้มีขนาด 2GB 4GB 6GB และ 8GB ตามลำดับ ในส่วนของ RAM ฝั่ง domU กำหนดขนาดเป็น 1GB 2GB และ 3GB ตามลำดับ จากนั้นแบ่งการทดลองเป็น 2 ฝั่ง คือ 1. ทดลองฝั่ง domU และ 2. ทดลองฝั่ง dom0 ในส่วนของซอฟต์แวร์ผู้วิจัยได้ทำการลง logger ในระบบปฏิบัติการ linux บน fedora16 โดยในหัวข้อนี้จะอธิบายเกี่ยวกับสภาพแวดล้อมของฮาร์ดแวร์และซอฟต์แวร์ที่ผู้วิจัยใช้ในการทดลอง

## 4. วิธีดำเนินการ

หัวข้อนี้จะกล่าวถึงการหาค่า sleeping time วิธีการนำค่า sleeping time ไปใช้ในงานทดลอง และการกำหนดขนาดของ RAM ในการทดลองแต่ละส่วน ในการทดลองนี้จะใช้สถาปัตยกรรมของคลาวด์ในภาพที่ 1 [5][10][11] มาออกแบบการทดลอง จุดมุ่งหมายของการ

ทดลองคือ ทดสอบประสิทธิภาพของ logger เมื่อทำการปรับขนาด RAM ใน dom0 และ domU โดยในการทดลองจะกำหนดค่า sleeping time ไว้ที่ 66.25 ms ซึ่งเป็นค่าที่ได้มาจากการรัน logger บน dom0 ที่มีขนาด 8 core และ domU มีขนาดระหว่าง 1 ถึง 8 core (d08c-u1c d08c-u2c d08c-u3c d08c-u4c d08c-u5c d08c-u6c d08c-u7c และ d08c-u8c) จนครบแล้วจึงนำเอาค่า sleeping time ทั้งหมดมาหาค่าเฉลี่ย ซึ่งเท่ากับ 66.25 ms หลังจากได้ค่าเฉลี่ยของ sleeping time แล้ว จะนำค่า sleeping time ไปใช้ในการทดลอง โดยจะแบ่งการทดลองเป็น 2 ส่วน คือ

**4.1 การปรับขนาด RAM ใน domU**

ส่วนที่ 1 (แสดงผลการทดลองในหัวข้อ 5.1) คือ การทดลองปรับ RAM ของ domU และส่วนที่ 2 จะเป็นการปรับ RAM ใน dom0 การทดลองปรับ RAM ใน domU ได้ทำการเซตค่า RAM ของ domU ให้มีขนาด 1GB 2GB และ 3GB RAM ของ dom0 มีขนาดคงที่ 8GB และขนาด CPU core ของ dom0 มีขนาด 8 core และ CPU core ของ domU มีขนาดระหว่าง 1 ถึง 8 core (d08c-u1c d08c-u2c d08c-u3c d08c-u4c d08c-u5c d08c-u6c d08c-u7c และ d08c-u8c บน RAM 1 GB ของ domU) หลังจากทำการทดลองเสร็จ บันทึกผลการทดลองและทำการปรับขนาด RAM ของ domU เป็น 2GB และ 3GB ตามลำดับและทดลองโดยใช้กระบวนการเดิม

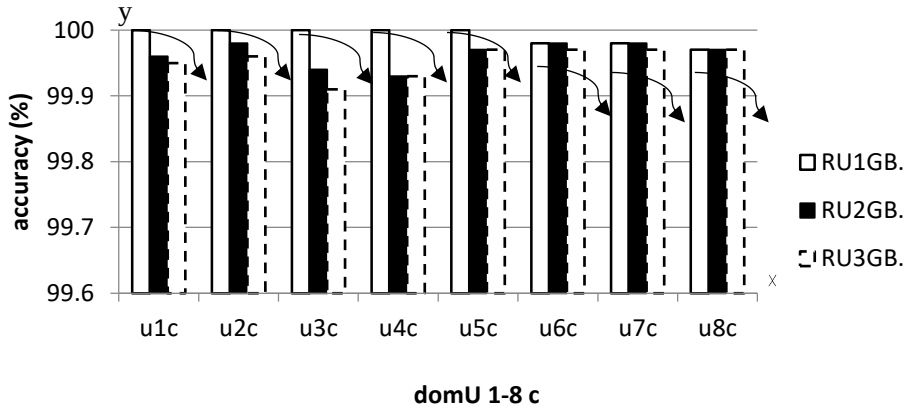
**4.2 การปรับขนาด RAM ใน dom0**

หลังจากทำการทดลองส่วนที่ 1 การปรับ RAM ใน domU แล้ว จะเริ่มทำการทดลองในส่วนที่ 2 (แสดงผลการทดลองในหัวข้อ 5.2) การปรับ RAM ใน dom0 โดยกำหนดขนาด RAM ของ dom0 ไว้ที่ 2GB 4GB 6GB และ 8GB ตามลำดับ กำหนด RAM ของ domU ไว้คงที่ 1GB หลังจากการตั้งค่าเสร็จจะทำการทดลองโดยการรัน logger บน dom0 ที่มีขนาด RAM 2GB 4GB 6GB และ 8GB ตามลำดับ บันทึกผลการทดลองและสร้างกราฟจากผลที่ได้เพื่อดูประสิทธิภาพของ logger หลังจากมีการเปลี่ยนแปลง RAM

## 5. ผลการศึกษาและอภิปรายผล

**5.1 ผลการทดลองการปรับขนาด RAM ใน domU**

จากหัวข้อ 4.1 การปรับขนาด RAM ใน domU ผลการทดลอง คือ เมื่อเพิ่ม RAM ให้กับ domU ความแม่นยำของ logger ลดลง แสดงในภาพที่ 2 ซึ่งจะแสดงการเพิ่มขนาด RAM ของ domU แต่ละ CPU core ใน domU (u1c-u8c) แล้วความแม่นยำของ logger ลดลง



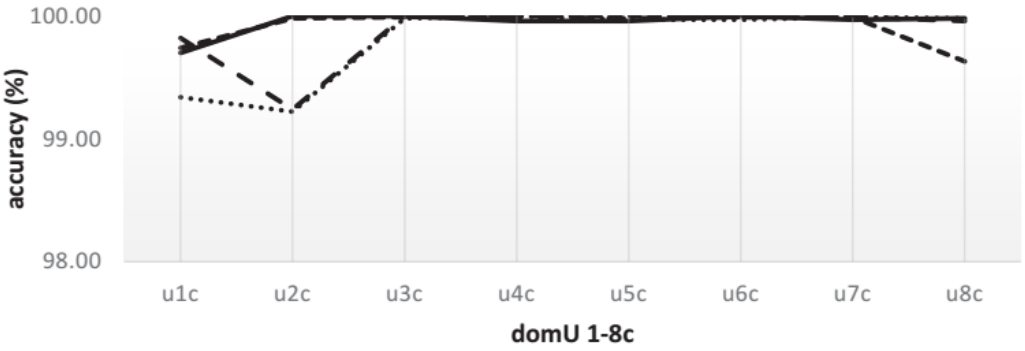

ภาพที่ 2 The accuracy of loggger in domU 1-8 core (u1c-u8c) with RAM in domU 1-3 GB. (RU1GB., Ru2GB. and Ru3GB.)

จากภาพที่ 2 ใช้กราฟแท่งแสดงผลการทดลองบนแกน x โดยกำหนดเป็นจำนวน CPU core ของ domU ขนาดตั้งแต่ 1 ถึง 8 core (u1c-u8c) บนแกน y แสดงค่าความแม่นยำของ logger ที่มีค่าตั้งแต่ 99.6% – 100% โดยกราฟแท่งเส้นทึบซ้ายสุดของแต่ละ CPU core domU (u1c u2c u3c u4c u5c u6c u7c และ u8c) แทนด้วย RAM ของ domU ขนาด 1GB (RU1GB) กราฟแท่งสีดำถัดมาแทนด้วย RAM ของ domU ขนาด 2GB (RU2GB) กราฟแท่งเส้นประขวาสุดของแต่ละ domU แทนด้วย RAM ของ domU ขนาด 3GB (RU3GB) จะเห็นได้ว่าไม่ว่าจำนวน CPU core ของ domU มีจำนวนเท่าไร ถ้าหากเพิ่ม RAM ใน domU ความแม่นยำของ logger ก็จะลดลง ทั้งนี้เกิดจากเมื่อ memU (RAM) ที่อยู่ใน domU มีขนาดเพิ่มขึ้น และเมื่อ logger เข้าไปบันทึกข้อมูลของโปรเซส read (read_mem) ซึ่งอยู่ใน memU ที่มีขนาดเพิ่มขึ้นทำให้ logger ต้องค้นหาโปรเซส read ในขนาดที่กว้างมากขึ้น จึงทำให้ logger ทำงานมากขึ้น ความแม่นยำของ logger ดังกล่าวจึงลดลง ดังภาพที่ 2 เช่น ขณะที่ CPU core ของ domU 1 core (u1c) จะเห็นว่าเมื่อ logger เข้าไปจับโปรเซส read ใน memU ขนาด 1GB ความแม่นยำที่ได้คือ 100% แต่เมื่อเพิ่มขนาด RAM ให้กับ memU เป็น 2GB ค่าความแม่นยำที่ได้ลดลงมาที่ 99.96% และเมื่อเพิ่มขนาด RAM เป็น 3GB ความแม่นยำจะลดลงเหลือ 99.95% ตามลำดับ จะสังเกตได้ว่าถ้าเพิ่ม RAM ให้กับ domU ความแม่นยำจะลดลง

5.2 **ผลการทดลองการปรับขนาด RAM ใน dom0**

จากหัวข้อ 4.2 การปรับขนาด RAM ใน dom0 (การทดลองส่วนที่ 2) เมื่อเพิ่มRAM ให้กับ

หลังจากเพิ่ม RAM ใน dom0 แล้วความแม่นยำไม่มีการเปลี่ยนแปลง ทั้งนี้ผู้วิจัยกำหนดในช่วงเปอร์เซ็นต์ที่ 99-100 ว่า logger มีความแม่นยำไม่ต่างกัน เพราะค่าความความต่าง 1% ที่เกิดขึ้นเกิดจากระบบทำงานผิดปกติเพียงชั่วครู่ จะสังเกตได้ว่าเมื่อ logger มีความแม่นยำ 100% และตกมาที่ 99% จากนั้นเมื่อทำงานต่อ ความแม่นยำที่ตกลงมา 99% ก็กลับขึ้นไปทำงาน 100% เช่นเดิม ทางผู้วิจัยจึงเห็นว่าการทำงานของ logger ถ้าในช่วงเปอร์เซ็นต์ที่ 99-100 จะถือว่า logger มีความแม่นยำไม่ต่างกัน




ภาพที่ 3 The accuracy of loggger in domU 1-8 core (u1c-u8c) with RAM in dom0 2 GB., 4 GB., 6 GB. and 8 GB. (Rd02GB., Rd04GB., Rd06GB. and Rd08GB.)

จากภาพที่ 3 ผู้วิจัยจะมุ่งเน้นไปที่ส่วนของ 99.00%-100% ซึ่งในส่วนนี้เป็นส่วนของเปอร์เซ็นต์ความแม่นยำของ logger ไม่เปลี่ยนแปลง ซึ่งจะใช้กราฟเส้นแสดงผลการทดลองโดยกราฟเส้นสีดำทึบแทน RAM ของ dom0 ขนาด 2GB (Rd02GB) เส้นประถี่แทน RAM ของ dom0 ขนาด 4GB (Rd04GB) เส้นประห่างแทน RAM ของ dom0 ที่มีขนาด 6GB (Rd06GB) และเส้นจุดไข่ปลาแทน RAM ของ dom0 ขนาด 8GB (Rd08GB) จากภาพที่ 3 จะเห็นว่าไม่ว่าขนาด RAM ของ dom0 มีจำนวนเท่าไร ค่าความแม่นยำของ logger ก็จะมีค่าที่ใกล้เคียงกัน คือ 99-100% สาเหตุที่ทำให้ logger ทำงานได้ประสิทธิภาพความแม่นยำใกล้เคียงกัน เนื่องจากกระบวนการทำงานของระบบปฏิบัติการจะมีการกำหนดพื้นที่ใน RAM ให้โปรเซสทำงานในขนาดที่เหมาะสมอยู่แล้ว[3] ดังนั้นเมื่อ logger ใช้พื้นที่ไม่มากในการประมวลผลและเมื่อมีการเพิ่มขนาดของ RAM เข้าไปแต่ logger ก็

ยังคงใช้พื้นที่ในการประมวลผลคงเดิม จึงไม่ส่งผลต่อการทำงานของ logger จึงทำให้ค่าความแม่นยำคงเดิม ซึ่งสามารถสังเกตได้จากพื้นที่ของ logger ขณะที่ โปรเซสบน RAM ได้ใน ตารางที่ 2

| RAM | พื้นที่ขณะที่ logger กำลังโปรเซสบน RAM |
|---|---|
| 2GB. | 27.25 m |
| 4GB. | 27m |
| 6GB. | 27m |
| 8GB. | 27.25m |

ตารางที่ 2 Show memory of logger process while running time at capture read process 1000 times

จาก ตารางที่ 2 แสดงพื้นที่ขณะที่ logger กำลังโปรเซสบน RAM โดยวิธีการหาพื้นที่ขณะโปรเซสกำลังทำงานอยู่ หาได้จากคำสั่ง top ซึ่งอยู่ในระบบปฏิบัติการ linux โดยคำสั่ง top จะแสดงค่าสถานะต่างๆ [8] และผู้วิจัยจะเน้นไปที่ผลลัพธ์จาก RES ซึ่ง RES คือ การใช้หน่วยความจำในส่วนเฉพาะภายใน RAM ซึ่งก็คือพื้นที่ที่ RAM กำลังถูกโปรเซสและในการทดลองจะทำการทดลอง โดยการรัน logger ให้จับโปรเซส read จำนวน 1000 รอบ และทำการทดสอบซ้ำไปมาจำนวน 10 ครั้ง และหาค่าเฉลี่ย ในสภาวะ RAM ที่ต่างกัน คือ RAM ของ dom0 ขนาด 2GB 4GB 6GB และ 8GB จะสังเกตได้ว่าไม่ว่า RAM จะมีขนาดเท่าไร logger จะใช้พื้นที่ในการโปรเซสที่ใกล้เคียงกัน จึงสรุปได้ว่าการเพิ่ม RAM ให้กับ dom0 จะไม่ส่งผลต่อประสิทธิภาพความแม่นยำของ logger

**6. สรุปและข้อเสนอแนะ**

ในบทความนี้จะนำเสนอเกี่ยวกับการวิเคราะห์ผลกระทบของหน่วยความจำหลักต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยอาศัยการวิเคราะห์ผลกระทบของ RAM ที่มีต่อประสิทธิภาพของ logger ซึ่งเมื่อทำการวิเคราะห์ผลกระทบที่เกิดขึ้นแล้ว จะทำให้สามารถวิเคราะห์การทำงานและคุณภาพของระบบบันทึกเหตุการณ์ต่อไปได้ โดยการวิเคราะห์ผลกระทบของหน่วยความจำหลัก ยังสามารถนำไปใช้ประโยชน์เป็นกรณีศึกษา เพื่อเป็นแนวทางในการออกแบบและ

นำไปใช้ในระบบบันทึกเหตุการณ์จริงได้ โดยจะทำให้สามารถจัดสรรทรัพยากรอย่างเหมาะสมและคุ้มค่าที่สุด ซึ่งจะทำให้ผู้ออกแบบระบบบันทึกเหตุการณ์สามารถออกแบบระบบได้อย่างเหมาะสมและมีประสิทธิภาพ เหมาะกับการนำไปใช้งานในระบบคลาวด์จริง

## 7. บรรณานุกรม

1. CSA. (2016). The Treacherous 12 - Cloud Security Alliance. Cloud Security Alliance. 1-35.
2. Ian Molyneaux. (2014). The Art of Application Performance Testing From Strategy to Tools. *United Startes of America: O'Reilly Media, Inc.,1 0 0 5 Gravenstein Highway Nort, Sebastopol, CA 95472*.
3. II.อัจจิมา เลี้ยงอยู่ I.สุธี พงศาสกุลชัย. (2553). *ระบบปฏิบัติการ (Operating Systems).* กรุงเทพฯ: สำนักพิมพ์ เคทีพี คอมพ์ แอนด์ คอลซัลท์.
4. Khalid Moussaid, Amina El omri, Mohamed Rida Zouhair Chiba* Noureddine Abghour. (2016). A Survey of Intrusion Detection Systems for Cloud Computing Environment. *IEEE*.
5. Pakorn Chan-in and Winai Wongthai. (2016). PERFORMANCE IMPROVEMENT CONSIDERATIONS OF CLOUD. *ICIC international*.
6. Po-Kai Chen and David J. Liou Global Unichip Corp., Kevin Ho. Tsung-Yi Chou. (2012). High Speed DDR2/3 PHY and Dual CPU Core Design for 28nm SoC. *IEEE*, 1-5.
7. San Murugesan and Lrena Bojanova. (ม.ป.ป.). Cloud Computing. *United Kingdom:Willey-IEEE Press.*
8. William E. Jr. Shotts. (2012). *The Linux Command Line: A Complete Introduction.* No Starch Press.
9. William Stallings. (2005). *Operating Systems Internals and Design Principle (Fifth Edition).* Perentice Hall.
10. Winai Wongthai and Aad van Moorsel. (2015). PERFORMANCE MEASUREMENT OF LOGGING SYSTEMS IN INFRASTRUCTURE AS A SERVICE CLOUD. *ICIC International*.
11. Winai Wongthai and Francisco Liberral Rocha and Van Moorsel. (2013). A Generic Logging Template for Infrastucture as a Service Cloud. IEEE Computer Society. *IEEE Computer Society*.
12. X. J. LI Dong. (2016). Study of Performance Testing of Information System Based on Domestic CPU and OS. *IEEE*, 112-116.

7th PHAYAO RESEARCH CONFERENCE

# An Analysis of Impacts of Main Memory against Performance of Logging System in the Cloud

นายไกรยวิชช์ ศุภโสภาพงศ์[1] และ วินัย วงษ์ไทย[2]

Kraiyawich Suppasopapong[1] and Winai Wongthai[2]

[1]Department of computer Science and Information Technology, Faculty of Science, Naresuan University, Phitsanulok, Thailand, 65000 kraiyawichs58@email.nu.ac.th

[2]Center of Excellence in Nonlinear Analysis and Optimization, Faculty of Science, Naresuan University, Phitsanulok, Thailand, 65000 HYPERLINK "mailto:winaiw@nu.ac.th" \h winaiw@nu. ac. th

**บทคัดย่อ**

การประมวลผลแบบกลุ่มเมฆหรือคลาวด์มีอัตราใช้งานที่เติบโตขึ้นทุกปี มีหลากหลายบริษัทให้การบริการ โดยจะให้บริการทรัพยากรทางคอมพิวเตอร์ออนไลน์ ปัญหาสำคัญของคลาวด์ในยุคปัจจุบัน คือ ปัญหาด้านความปลอดภัย เช่น เมื่อข้อมูลของผู้ใช้งานถูกเข้าถึงโดยผู้ไม่หวังดี ทั้งนี้ระบบบันทึกเหตุการณ์หรือ logging system เป็นหนึ่งในแนวทางที่สามารถช่วยลดความเสี่ยงของสาเหตุต่างๆ ที่อาจก่อให้เกิดปัญหาด้านความปลอดภัยบนคลาวด์ ในระบบบันทึกเหตุการณ์สามารถบันทึกข้อมูลเหตุการณ์ที่เกิดขึ้นบนคลาวด์ เช่น ผู้ไม่หวังดีเข้าถึงไฟล์โดยไม่ได้รับอนุญาต และข้อมูลที่ถูกบันทึกสามารถนำไปเป็นหลักฐานเมื่อเกิดปัญหาบนคลาวด์ แต่อย่างไรก็ตาม มีงานวิจัยที่ได้ให้ข้อเสนอแนะว่า การเพิ่มประสิทธิภาพฮาร์ดแวร์จะส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ดีขึ้น ในบทความนี้จึงนำเอาข้อเสนอแนะในงานวิจัยดังกล่าวมาปรับปรุงและออกแบบการทดลองด้วยวิธีการเพิ่มจำนวนของหน่วยประมวลผลกลาง แล้ววิเคราะห์ว่ามีผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์อย่างไร จากนั้นสรุปและอภิปรายผลเกี่ยวกับผลกระทบของหน่วยประมวลผลกลางที่มีต่อระบบบันทึกเหตุการณ์ ข้อสรุปจากงานวิจัยนี้สามารถนำไปเป็นข้อมูลพื้นฐานสำหรับการออกแบบและพัฒนาระบบบันทึกเหตุการณ์ได้ นอกจากนี้การทดลองด้วยวิธีการเพิ่มจำนวนของหน่วยประมวลผลกลางและการวิเคราะห์ผลกระทบต่อระบบบันทึกเหตุการณ์ยังไม่มีการกล่าวในวรรณกรรมใดมาก่อน ผู้วิจัยจึงได้ออกแบบและทำการทดลองนี้ขึ้นมา

**คำสำคัญ**: ความปลอดภัย การประมวลผลแบบกลุ่มเมฆ ระบบบันทึกเหตุการณ์ การวัดประสิทธิภาพ หน่วยประมวลผลกลาง

## Abstract

Cloud Computing or Cloud has usage rate is growing every year. Has many companies of service and resource service access by online as storages, virtual machines, operating system, application, infrastructure, etc. While cloud is galore use in many application areas, it security is importance, A sample, malevolent users accesses to a cloud user's file by unauthorized. A logging system is one of method to alleviate risks participatory of cloud security. This system can, a sample, create evidence for who was an accessor of user's file in cloud. This evidence can be used when has event on cloud. However, previous works has test performance of logging system and provide suggestion, if have increase performance of hardware impact to logging system. This paper has adjust experiment previous paper and discussed quality analysis of a logging system measurement of a central processing unit of the system. Therefor to consider that how the central processing unit impact the performance of the system. This paper for developer of logging system by can be use as a guide line to design and develop the logging. In addition method test by increase central processing unit yet don't any literature mentioned before.


**Keywords: Security, Cloud, Logging system, Performance Measurement, Central Processing Unit**

## 1. บทนำ

คลาวด์คอมพิวติ้งหรือคลาวด์มีอัตราการใช้งานที่เติบโตขึ้นทุกปี โดยมีหลากหลายบริการจากบริษัทที่ให้การบริการคลาวด์ เช่น Amazon Web Services (AWS) Dropbox Offiece 365 และอื่นๆ แนวโน้มด้านการใช้งานบริการคลาวด์มีอัตราการใช้งานเพิ่มมากขึ้น โดยการให้บริการคลาวด์มีรูปแบบการใช้งาน คือ เสียค่าใช้จ่ายเท่ากับจำนวนทรัพยากรที่ใช้ ด้วยรูปแบบข้างต้นคลาวด์จึงเป็นทางเลือกที่ดีสำหรับการนำคลาวด์มาประยุกต์ใช้งานด้านไอทีสำหรับองค์กร (Chiba, et al., 2016; Surbiryala, et al., 2017) แต่อย่างไรก็ตามปัญหาด้านความปลอดภัยก็เป็นสิ่งที่หลีกเลี่ยงไม่ได้สำหรับการนำคลาวด์มาใช้งานจริง ดังนั้นการนำคลาวด์มาใช้งานจริงจึงต้องมีการคำนึงถึงความปลอดภัย (Surbiryala, et al., 2017) นอกจากนี้เจ้าหน้าที่ CIO (Chief Information Officer) ซึ่งเป็นเจ้าหน้าที่ด้านไอทีขององค์กรต่างๆ กังวลว่าความปลอดภัยจะเป็นอุปสรรคต่อการนำคลาวด์มาใช้งานจริงภายในองค์กร (Chiba, et al., 2016) และด้วย 2 เหตุผลข้างต้นที่บอกว่าความปลอดภัยบนคลาวด์เป็นสิ่งที่หลีกเลี่ยงไม่ได้ (Surbiryala, et al., 2017) และจะเป็นอุปสรรคในการนำคลาวด์มาใช้งานจริงในองค์กร (Chiba, et al., 2016) จึงได้มีองค์กร CSA (Cloud Security Alliance) ที่ทำการวิจัยและระบุปัญหาเกี่ยวกับภัยคุกคามของคลาวด์ไว้ในรายงานต่างๆ เช่น รายงานความเสี่ยงภัยคุกคามของ

คลาวด์ปี 2016 หรือที่เรียกว่า The Treacherous 12 Cloud Computing Top Threats in 2016 (CSA, 2016a, 2016b)

จากรายงานของ CSA ดังกล่าว ซึ่งได้ระบุเกี่ยวกับปัญหาภัยคุกคามของคลาวด์ (CSA, 2016a, 2016b) ทำให้มีนักวิจัยหลายท่านได้ทำการวิจัยเพื่อแก้ไขและบรรเทาปัญหาดังกล่าว ในบทความนี้ผู้วิจัยได้สนใจวิธีการบรรเทาปัญหาด้วยระบบบันทึกเหตุการณ์ (logging systems) ซึ่งเป็นระบบที่ใช้ในงานวิจัยของ Chan-in Wongthai และ Moorsel (Chan-in & Wongthai, 2017; Wongthai & Moorsel, 2016; Wongthai, et al., 2013c) โดยระบบบันทึกเหตุการณ์ดังกล่าวเป็นระบบที่สามารถบันทึกข้อมูลความเป็นไปในระบบคลาวด์ เช่น บันทึกว่าใครเคยเข้าถึงไฟล์อะไร ข้อมูลที่บันทึกได้นี้จะนำไปเป็นหลักฐานเมื่อเกิดปัญหาเกี่ยวกับความปลอดภัยของคลาวด์ ภายในระบบบันทึกเหตุการณ์จะมีโปรเซสที่ใช้บันทึกเหตุการณ์ เรียกว่า logger ในบทความนี้จะเน้นไปที่กระบวนการทำงานของ logger และเน้นตรวจสอบประสิทธิภาพของ logger หลังมีการปรับเพิ่มขนาด CPU core[12]

ในงานวิจัยก่อนหน้า (Wongthai & Moorsel, 2016) ได้มีการทดสอบประสิทธิภาพของ logger และให้ข้อเสนอแนะในงานวิจัยว่าหากมีการเพิ่มขนาดของ CPU core ให้กับ logger จะทำให้ประสิทธิภาพของ logger มีประสิทธิภาพดีขึ้นเพราะจะส่งผลให้ระบบบันทึกเหตุการณ์ดีขึ้นตามลำดับ บทความนี้จึงได้นำข้อเสนอแนะดังกล่าว มาออกแบบการทดลอง ด้วยวิธีการปรับขนาดของ CPU core ว่าจะส่งผลอย่างไรต่อ logger โดยได้แบ่งการทดลองเป็น 2 ส่วน คือ 1. เครื่องที่ถูกตรวจสอบ ในบทความนี้เรียกเครื่องที่ถูกตรวจสอบว่า domU (domain User) 2. เครื่องที่ทำการติดตั้ง logger ในบทความนี้เรียกเครื่องที่ทำการติดตั้ง logger ว่า dom0 (Domain Zero)

การพัฒนาซอฟต์แวร์และการทดสอบประสิทธิภาพถือว่าเป็นเรื่องสำคัญ (Molyneaux, 2014) เช่นเดียวกันกับการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ (Wongthai & Moorsel, 2016; Wongthai & van Moorsel, 2016) ดังนั้นในบทความนี้จึงนำเสนอผลการทดสอบประสิทธิภาพของ logger ด้วยวิธีการปรับขนาด CPU core และวิเคราะห์ผลกระทบต่อระบบบันทึกเหตุการณ์ ผลการวิเคราะห์ที่ได้จะเป็นข้อมูลสำคัญสำหรับผู้ออกแบบระบบบันทึกเหตุการณ์เพราะจะทำให้สามารถเลือกใช้ทรัพยากรได้อย่างเหมาะสมและลดต้นทุนได้

หน่วยประมวลผล (Processing Unit) คือ หน่วยที่แปลคำสั่งแล้วทำการประมวลผลคำสั่งเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ ได้แก่ หน่วยประมวลผลกลาง (Central Processing Unit : CPU) และหน่วยประมวลผลร่วม (Coprocessor/Chipsets) ในบทความนี้จะอธิบายเกี่ยวกับ CPU เพียงอย่างเดียวเนื่องจากส่วนของ CPU จะเป็นส่วนที่ใช้ในงานทดลอง ในงานวิจัยจะใช้คำว่า CPU core โดย CPU core มีความสามารถใน

[12] CPU (Central Processing Unit) หน่วยประมวลผลกลาง (รายละเอียดอยู่ในย่อหน้าที่ 5 ของบทนำ)

การประมวลผลและรองรับการใช้งานซอฟต์แวร์ประยุกต์ต่างๆ (Carrick, et al., 2010; Galvin, et al., 2006; เลี้ยงอยู่ & พงศาสกุลชัย, 2553) และเป็นส่วนสำคัญในการประมวลผลของเครื่อง

บทความนี้นำเสนอผลการทดสอบประสิทธิภาพของ logger และนำผลการทดสอบมาวิเคราะห์ผลกระทบที่มีต่อระบบบันทึกเหตุการณ์ จากนั้นสรุป อภิปรายผลเพื่อเป็นข้อมูลพื้นฐานสำหรับการวางแผนและออกแบบระบบบันทึกเหตุการณ์

## 2. วัตถุประสงค์ของการวิจัย

2.1 ทดสอบประสิทธิภาพของ logger หลังการปรับเพิ่มขนาดของ CPU core ขนาดต่างๆ

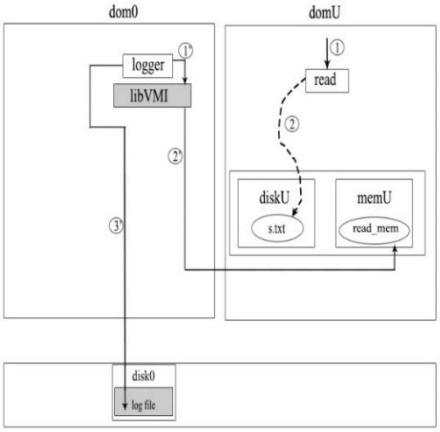
2.2 นำผลทดสอบที่ได้ มาวิเคราะห์ สรุปและอภิปรายผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์

## 3. วัสดุอุปกรณ์และวิธีการศึกษา

หัวข้อนี้จะอธิบายเกี่ยวกับคอมพิวเตอร์ ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการทดลอง รวมถึงอธิบายเกี่ยวกับสถาปัตยกรรมการทำงานของระบบบันทึกเหตุการณ์และการกำหนดขนาดของหน่วยประมวลผลกลางในการทดลอง โดยในการทดลองได้ทดลองบนเครื่องคอมพิวเตอร์ CPU core Xeon ขนาด 2GB. 4GB. 6GB. 8GB. RAM DDR3 SDRAM 8GB. บนระบบปฏิบัติการ Linux Fedora 16

### 3.1 สถาปัตยกรรมของระบบบันทึกเหตุการณ์

ในหัวข้อนี้จะอธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์บนคลาวด์ โดยสถาปัตยกรรมของคลาวด์ที่ใช้ อ้างอิงจากงานของ Wongthai และ Moorsel แสดงในภาพที่ 1 โดยมีการแก้ไขภาพจากงานวิจัยเดิม เพื่อให้สอดคล้องกับการทดลอง (Wongthai & Moorsel, 2016) โดยจะแบ่งการทำงานออกเป็น 2 ส่วน คือ 1. ส่วนของผู้ใช้บริการหรือลูกค้า (domU) 2. ส่วนของผู้ให้บริการ (dom0)




**ภาพที่ 1** An Experimental Environment domU and dom0

**ที่มา :** แก้ไขภาพจาก Wongthai (Wongthai & Moorsel, 2016). Performance measurement of logging systems in infrastructure as a service cloud. ICIC International. 2016, 349.

### 3.1.1 ส่วนของผู้ใช้บริการหรือลูกค้า (domU)

ในหัวข้อนี้จะกล่าวถึง คือ หัวข้อที่ 1 ส่วนของผู้ใช้บริการหรือลูกค้า ในภาพที่ 1 คือ ส่วนของ domU ในส่วนนี้ ลูกค้าเป็นเจ้าของ diskU (diskU เป็น virtual disk) ทุกสิ่งที่ถูกจัดเก็บใน diskU จะเป็น virtual disk และใน diskU จัดเก็บไฟล์ชื่อ s.txt ซึ่งเป็นไฟล์สำคัญของลูกค้า ภายใน diskU จะจัดเก็บไฟล์อะไรก็ได้ เช่น text executable หรือ database file กระบวนการทำงานภายใน domU มีขั้นตอน คือ 1. ลูกค้าซึ่งเป็นเจ้าของ domU และ domU อาจถูกครอบครองโดยผู้ไม่หวังดี 2. รันแอปพลิเคชัน read เพื่ออ่านไฟล์ s.txt ภายใน diskU (เส้นประหมายเลข 2) (โปรเซสคือ โปรแกรมที่ถูกรัน (A.Saha, 2006) หรือโปรแกรมที่กำลังทำงาน (Bryant & O'Hallaron, 2010) 3. เมื่อ read ถูกรัน จากแอปพลิเคชันจะเปลี่ยนเป็นโปรเซส read โดยกระบวนการทำงานเหล่านี้จะเรียกว่ากิจกรรมในแอปพลิเคชัน read ซึ่งกิจกรรมในแอปพลิเคชัน read มีกระบวนการ ดังนี้ 1. เปิดไฟล์ s.txt 2. อ่านไฟล์และพิมพ์ข้อมูลภายใน s.txt 3. ปิดการทำงาน 4. จบการทำงาน โดยกิจกรรมในแอปพลิเคชัน read เหล่านี้จะถูกจัดเก็บไว้ภายใน read_mem และ read_mem จะอยู่ใน memU ดังแสดงในรูปที่ 1 ซึ่งการตรวจสอบและบันทึกข้อมูลที่เกี่ยวข้องกับกระบวนการเหล่านี้ เรียกว่าการบันทึกประวัติไฟล์สำคัญ โดยภายใน read_mem ได้จัดเก็บข้อมูลดังแสดงในตารางที่ 1

| f_nm | p_id | p_nm | p_ownId |
|---|---|---|---|
| s.txt | 4624 | read | 1002 (alice) |

**ตารางที่ 1** The content of the history of critical file

**ที่มา** : (Wongthai, et al., 2013a). Logging solutions to mitigate risks associated with threats in infrastructure as a service cloud, IEEE, 2014, 166.

จากตารางที่ 1 แสดงข้อมูลที่ถูกจัดเก็บใน read_mem ดังนี้ 1. f_nm (ชื่อไฟล์ s.txt) 2. p_id (ไอดีของโปรเซสที่ถูกรัน 4624) 3. p_nm (ชื่อของโปรเซสที่ถูกรัน read) 4. p_ownId (ชื่อผู้รันโปรเซส alice)

### 3.1.2 ส่วนของผู้ให้บริการ (dom0)

จากหัวข้อ 3.1 ได้อธิบายถึงสถาปัตยกรรมของระบบบันทึกเหตุการณ์บนคลาวด์ สภาพแวดล้อมและกระบวนการทำงานของระบบบันทึกเหตุการณ์ โดยในภาพที่ 1 จะแบ่งการทำงานเป็น 2 ส่วน คือ 1. ส่วนของผู้ใช้บริการหรือลูกค้าซึ่งจะถูกกล่าวในหัวข้อ 3.1.1 และ 2. ส่วนของผู้ให้บริการซึ่งจะถูกกล่าวในย่อหน้าถัดไป

ในย่อหน้านี้จะอธิบายการทำงานในส่วนที่ 2. ส่วนของผู้ให้บริการ โดยกระบวนการทำงานในส่วนของผู้ให้บริการ จะอยู่ในส่วนของ dom0 ภายใน dom0 ได้ติดตั้ง logger สำหรับใช้บันทึกเหตุการณ์ โดยกระบวนการทำงานของ logger มีขั้นตอน ดังนี้ ขั้นที่ 1. รัน logger (หมายเลข 1' ในภาพที่ 1) จากนั้น logger

จะเรียกใช้ libVMI เพื่อเข้าไปใน memU ฝั่ง domU (หมายเลข 2’ ในภาพที่ 1 ซึ่งถือเป็นขั้นที่ 2) โดย libVMI เป็น library ในภาษา C ที่ถูกเขียนเพื่อให้สามารถเข้าไปอ่านข้อมูลที่ถูกจัดเก็บภายใน read_mem ใน memU ได้ (ข้อมูลที่ถูกจัดเก็บใน read_mem ได้ถูกอธิบายไว้ในตารางที่ 1) ต่อมา ขั้นที่ 3. หลังจาก logger เรียกใช้ libVMI เพื่อเข้าไปอ่านข้อมูลที่ถูกจัดเก็บใน read_mem แล้ว logger จะนำข้อมูลใน read_mem มาจัดเก็บลงในล็อกไฟล์ (log file) disk0 (หมายเลข 3’ ในภาพที่ 1)

ภาพรวมในหัวข้อนี้ได้อธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ การทำงานใน domU และ dom0 โดยในหัวข้อ 3.1.1 ได้มีการอธิบายเกี่ยวกับกิจกรรมของแอปพลิเคชัน read ซึ่งกิจกรรมของแอปพลิเคชัน readดังกล่าว จะถูกปรับปรุงเพื่อนำไปใช้ในการทดลอง ซึ่งจะอธิบายในหัวข้อ 3.2 ถัดไป

**3.2 sleeping time and accuracy**

**3.2.1 sleeping time**

จากข้อสรุปในหัวข้อ 3.1 ได้กล่าวว่าจะมีการปรับปรุงแอปพลิเคชัน read เพื่อใช้ในการทดลอง ในหัวข้อนี้จึงได้อธิบายเกี่ยวกับการปรับปรุงแอปพลิเคชัน read ด้วยการเพิ่มค่า sleeping time ก่อนนำแอปพลิเคชัน read ไปใช้ในการทดลอง ซึ่ง sleeping time คือ ค่าเวลาค่าหนึ่งที่ถูกกำหนดเพิ่มในกระบวนการทำงานของกิจกรรมในแอปพลิเคชัน read (กิจกรรมในแอปพลิเคชัน read ถูกกล่าวไว้ในหัวข้อ 3.1 ย่อหน้าที่ 2 ส่วนสุดท้ายของย่อหน้า) โดยในกระบวนการปรับปรุงแอปพลิเคชัน read จะเพิ่มค่า sleeping time ก่อนการปิดโปรแกรม ซึ่งมีขั้นตอนการทำงาน ดังนี้ 1. เปิดไฟล์ 2. อ่านไฟล์และพิมพ์ข้อมูลใน s.txt 3. ในส่วนของก่อนการปิดการทำงาน จะกำหนดเวลา x ms (millisecond) เช่น x = 60 ms เพื่อไม่ให้โปรเซสปิดการทำงานทันทีแต่จะให้ปิดการทำงานหลัง 60 ms (ดังนั้น 60 ms เรียกว่า sleeping time) 4. ปิดการทำงาน 5. จบการทำงาน วิธีการทั้งหมดนี้เป็นการรันแอปพลิเคชัน read เพียง 1 ครั้ง ในการทดลองผู้วิจัยได้รันแอปพลิเคชัน read 1000 ครั้ง จำนวน 10 รอบ

**3.2.2 accuracy**

ผู้วิจัยได้ทดสอบรัน logger ให้ตรวจจับไฟล์ชื่อ s.txt เมื่อมีการรันโปรเซส read จำนวน 100 ครั้ง หาก logger สามารถตรวจจับไฟล์ s.txt ได้ครบ 100 ครั้ง นั้นคือ logger มีค่าความแม่นยำหรือ accuracy 100% ค่าความแม่นยำของ logger สามารถมีได้ตั้งแต่ 0%–100% ถ้าหาก logger ไม่สามารถตรวจจับไฟล์ s.txt ได้เลย นั้นคือ ค่าความแม่นยำของ logger เป็น 0% ในการทดลอง ถ้าหาก logger ไม่สามารถตรวจจับไฟล์ s.txt ได้จะเรียกว่า “miss” และเมื่อ logger สามารถตรวจจับได้จะเรียก “hit” คำว่า miss รวมถึงการเกิดจาก logger ทำงานมีปัญหาอีกด้วย ดังนั้น ถ้ารัน logger ไปจับไฟล์ชื่อ s.txt เมื่อมีการรันโปรเซส read จำนวน 100 ครั้ง และ logger ตรวจจับไฟล์ s.txt ได้ 80 ครั้ง (hit = 80) นั้นหมายความว่า logger มีค่าความแม่นยำที่ 80% (accuracy = 80%)

หัวข้อ 3.2 นี้ได้อธิบายเกี่ยวกับค่าของ sleeping time และ accuracy ซึ่งเป็นค่าที่จะถูกนำไปใช้ในการทดลอง โดยค่า accuracy เป็นค่าที่ใช้วัดประสิทธิภาพของ logger ประสิทธิภาพของ logger ที่ดี ค่า accuracy จะใกล้เคียง 100% ในการทดลองนี้ผู้วิจัยจะใช้ค่า accuracy 100% เท่านั้น ดังนั้น คำว่า accuracy ในงานทดลองจึงมีค่า 100% ที่ค่า sleeping time = x ms (ค่า sleeping time ที่ดีต้องมีเวลาใกล้เคียง 0 ms.)

**3.3 วิธีดำเนินการ**

ในหัวข้อนี้จะอธิบายวิธีการดำเนินการตั้งแต่กระบวนการหาค่า sleeping time การนำค่า sleeping time ไปใช้ในการทดลอง และการกำหนดขนาดของ CPU core ในแต่ละส่วนของการทดลอง โดยในการทดลองนี้จะใช้ผลการทดลองของ Wongthai (Wongthai & Moorsel, 2016) ก่อนหน้ามาเป็นกรณีศึกษาและออกแบบการทดลอง โดยจุดมุ่งหมายในการทดลองนี้ คือ การทดสอบประสิทธิภาพของ logger เมื่อมีการกำหนดขนาดของ RAM ให้กับ domU และ dom0 โดยค่า sleeping time ที่กำหนดในการทดลองได้มาจากการอ้างอิงงานวิจัยของ Wongthai (Wongthai & Moorsel, 2016) ซึ่งได้ทำการทดลองวัดประสิทธิภาพของ dom0 โดยใช้ CPU core ของ dom0 ขนาด 4 core และผลการทดลองค่า sleeping time ที่ดีที่สุดอยู่ที่ 65 ms ผู้วิจัยจึงนำเอาค่า sleeping time 65 ms มาเป็นพื้นฐานในการออกแบบการทดลอง โดยการปรับค่า sleeping time ในการทดลองของผู้วิจัยเริ่มจากเซตค่า sleeping time ไว้ที่ 65 ms (จุดประสงค์ในการทดลองต้องการค่า accuracy = 100% เท่านั้น และค่า sleeping time ต้องใกล้เคียง 0 ms ที่สุด จึงถือว่า logger มีประสิทธิภาพดี) ถ้าหาก logger มีค่า accuracy 100% จะทำการปรับเวลาใน sleeping time ลง โดยจะปรับเวลาลงทีละ 1 ms แต่ถ้าหากค่า accuracy ไม่ใช่ 100% ก็จะปรับเวลาใน sleeping time ขึ้นทีละ 1 ms เพื่อให้เวลาใน sleeping time นานขึ้น จนกระทั่ง logger มีค่า accuracy 100% ในการทดลองแบ่งเป็น 2 ส่วน ดังนี้

**3.3.1 *ส่วนที่ 1* การปรับขนาดของ CPU core ใน domU**

หัวข้อนี้จะอธิบายเกี่ยวกับการปรับขนาดของ CPU core ใน domU โดยผู้วิจัยเริ่มการทดลองจากการตั้งค่าขนาด CPU core ของ dom0 ให้มีขนาดระหว่าง 1 core ถึง 8 core โดยที่ dom0 เป็นขนาดคงที่ (ขนาดคงที่ หมายถึง การกำหนดขนาดของ CPU core โดยที่ไม่เปลี่ยนแปลงขนาด) และปรับขนาด CPU core ของ domU ให้มีขนาดระหว่าง 1 core ถึง 8 core โดยที่ domU จะเป็นขนาดที่เปลี่ยนแปลง (ขนาดที่เปลี่ยนแปลง หมายถึง การกำหนดขนาดของ CPU core เป็นขนาดที่เปลี่ยนแปลงตั้งแต่ 1 core ถึง 8 core โดยแสดงผลเปรียบเทียบบนขนาดคงที่) เช่น กำหนดให้ขนาด CPU core ของ dom0 (ขนาดคงที่) มีขนาด 8 core และ ขนาด CPU core ของ domU (ขนาดที่เปลี่ยนแปลง) มีขนาดระหว่าง 1 core ถึง 8 core (d08c–u1c d08c–u2c d08c–u3c d08c–u4c d08c–u5c d08c–u6c d08c–u7c และ d08c–u8c) เมื่อกำหนดขนาดของ CPU core แล้ว ผู้วิจัยจะกำหนดค่า sleeping time เพื่อใช้ในการทดลอง (วิธีการหาค่า sleeping time อธิบาย

ไว้ในหัวข้อ 3.3 ในส่วนสุดท้าย) เมื่อทำการทดลองบน CPU core dom0 ขนาด 8 core เสร็จ จะปรับค่า CPU core ของ dom0 ลงเป็น 7 core 6 core 5 core 4 core 3 core 2 core และ 1 core ตามลำดับ จากนั้นใช้กระบวนการทดลองเดิม เพื่อตรวจสอบประสิทธิภาพของ logger ในสภาพแวดล้อมที่หลากหลาย

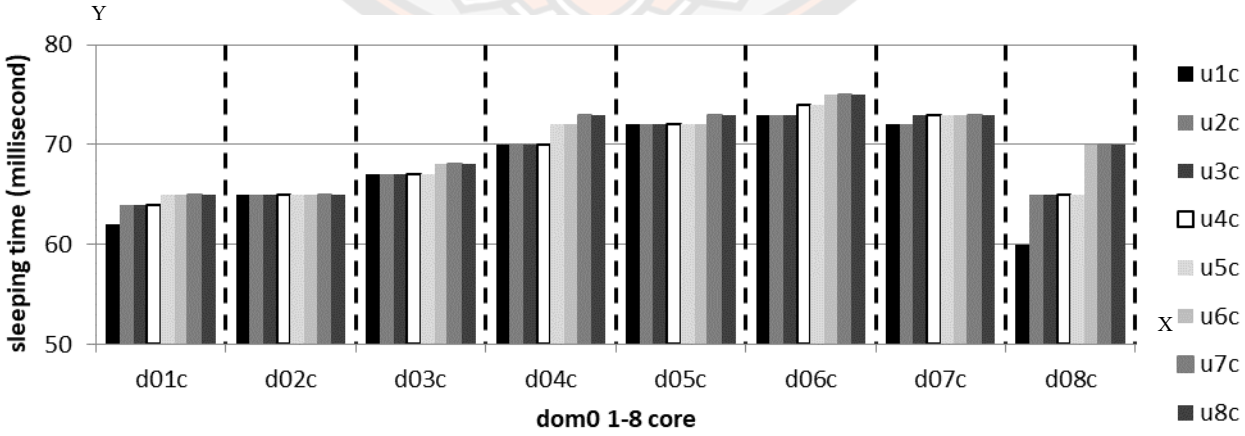
### 3.3.2 ส่วนที่ 2 การปรับขนาด CPU core ใน dom0

หัวข้อนี้จะอธิบายเกี่ยวกับการปรับขนาด CPU core ใน dom0 โดยผู้วิจัยจะตั้งค่าการทดลองโดยกำหนดขนาดของ dom0 และ domU ดังนี้ ตั้งค่าขนาดของ dom0 ให้มีขนาดระหว่าง 1 core ถึง 8 core และตั้งค่าขนาดของ domU มีขนาดระหว่าง 1 core ถึง 8 core เช่นเดียวกัน ในการทดลองได้กำหนดค่า domU เป็นขนาดคงที่ dom0 เป็นขนาดที่ถูกเปลี่ยนแปลง เช่น กำหนดขนาดของ domU เป็น 1 core และกำหนดขนาด ของ dom0 ให้มีขนาดระหว่าง 1 core ถึง 8 core (u1c–d01c u1c–d02c u1c–d03c u1c–d04c u1c–d05c u1c–d06c u1c–d07c และ u1c–d08c) หลังจากกำหนดขนาดของ CPU core แล้ว ก็จะกำหนดค่า sleeping time เมื่อทำการทดลองปรับขนาดของ dom0 ขนาดระหว่าง 1 core ถึง 8 core บน domU ขนาด 1 core เสร็จแล้ว ก็จะทำการ เพิ่มขนาด CPU core ของ dom0 ให้มีขนาด 2 core 3 core 4 core 5 core 6 core 7 core และ 8 core ตามลำดับ และใช้กระบวนการทดลองเดิม เพื่อให้ได้สภาพแวดล้อมที่หลากหลาย

## 4. ผลการศึกษา

### 4.1 การปรับขนาดของ CPU core ใน domU

จากหัวข้อ 3.3.1 ในส่วนที่ 1 การปรับขนาดของ CPU core ใน domU ผลการทดลอง คือ เมื่อมีการเพิ่มขนาดของ CPU core ใน domU แล้วค่า sleeping time มีค่าใกล้เคียงกันส่งผลให้ประสิทธิภาพของ logger คงเดิม ช่วงเวลาของ sleeping time ถ้าหากอยู่ในช่วงเวลาที่แตกต่างไม่เกิน 10 ms ผู้วิจัยนิยามให้ logger มีประสิทธิภาพคงเดิม

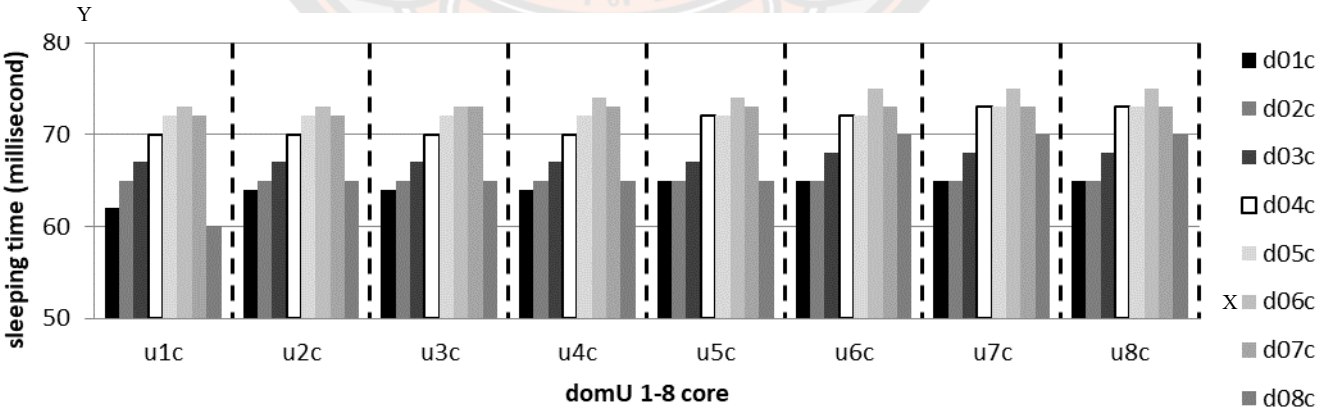



ภาพที่ 2 Show sleeping time number of domU's cores from 1–8 on number of dom0's cores from 1–8

ของ dom0 ขนาดระหวาง 1 core ถึง 8 core บนแกน Y แสดงคาของ sleeping time และกราฟแทงแทนขนาด CPU core ใน domU ขนาดระหว่าง 1 core ถึง 8 core โดยจะกำหนดกราฟแท่งสีดำซ้ายสุดของแต่ละ dom0 แทนด้วย domU ขนาด 1 core (u1c) กราฟแท่งลำดับถัดมาแทนด้วย domU ขนาด 2 core (u2c) ลำดับถัดมาแทนด้วย domU ขนาด 3 core (u3c) domU ขนาด 4 core (u4c) domU ขนาด 5 core (u5c) domU ขนาด 6 core(u6c) domU ขนาด 7 core (u7c) ตามลำดับจนถึงกราฟแท่งสีเทาในลำดับสุดท้ายของแต่ละ dom0 แทนด้วย domU ขนาด 8 core (u8c) กราฟจะแสดงในลักษณะเปรียบเทียบทีละชุด โดยขนาด CPU core ของ dom0 เป็นค่าคงที่ และกราฟแท่งบนแกน x จะแสดงขนาด CPU core ของ domU เป็นค่าที่เปลี่ยนแปลง เช่น ในชุด CPU core ของ dom0 ขนาด 1 core (d01c) และ CPU core ของ domU ขนาดระหว่าง 1 core ถึง 8 core (u1c–u8c) นำมาเปรียบเทียบกัน (d01c–u1c d01c–u2c d01c–u3c d01c–u4c d01c–u5c d01c–u6c d01c–u7c และ d01c–u8c) จะเห็นว่าค่าเวลาใน sleeping time ของ d01c จะมีความต่างของช่วงเวลาอยู่ที่ 62–65 ms ค่า sleeping time ใกล้เคียงกันจึงทำให้ประสิทธิภาพของ logger คงเดิม

**4.2 การปรับขนาดของ CPU core ใน dom0**

จากหัวข้อ 3.3.2 ในส่วนที่ 2 การปรับขนาด CPU core ใน dom0 ผลการทดลองที่ได้ คือ เมื่อมีการเพิ่มขนาดของ CPU core ให้กับ dom0 แล้วเกิดกราฟลักษณะคล้ายระฆังคว่ำดังภาพที่ 3 จุดที่ดีที่สุดอยู่ที่ CPU core dom0 มีขนาด 1 core และ 8 core




ภาพที่ 3 Show sleeping time number of dom0's cores from 1–8 on number of domU's cores from 1–8

จากภาพที่ 3 จะใช้กราฟแท่งแสดงผลการทดลองโดยกำหนดแกน X แทนด้วยขนาดของ CPU core ใน domU ขนาดระหว่าง 1 core ถึง 8 core บนแกน Y แสดงค่า sleeping time และกราฟแท่งแสดง CPU core ของ dom0 ขนาดระหว่าง 1 core ถึง 8 core โดยลักษณะของกราฟจะแบ่งเป็นชุด ชุดละ 8 แท่ง ดังนี้ กราฟแท่งสีดำฝั่งซ้ายสุดแต่ละ CPU core ของ domU แทนด้วย dom0 ขนาด 1 core (d01c) ลำดับถัดมาแทนด้วย dom0 ขนาด 2 core (d02c) 3 core (d03c) 4core (d04c) 5 core (d05c) 6 core (d06c) 7 core (d07c) ตามลำดับและกราฟแท่งสีเทาฝั่งขวาสุดของแต่ละ CPU core domU แทนด้วย CPU core ขนาด 8 core (d08c) โดยกราฟนี้จะแสดงการเปรียบเทียบเมื่อเพิ่ม CPU core ใน dom0 แล้วจะมีลักษณะที่คล้ายระฆังคว่ำ วิธีการเปรียบเทียบสามารถดูได้จากกราฟทีละชุดโดยให้ domU มีขนาดคงที่ dom0 มีขนาดที่เปลี่ยนแปลงระหว่าง 1 core ถึง 8 core เช่น เมื่อกำหนดขนาด domU คงที่ให้มีขนาด 1 core และ dom0 มีขนาดที่เปลี่ยนแปลง ให้มีขนาดระหว่าง 1 core ถึง 8 core (d01c–u1c d02c–u1c d03c–u1c d04c–u1c d05c–u1c d06c–u1c d07c–u1c และ d08c–u1c จะสังเกตได้ว่าเมื่อ dom0 มีขนาด 1 core ค่าของ sleeping time มีเวลาใกล้เคียง 0 ms เมื่อเพิ่มขนาด CPU core เป็น 2 core จนถึง 7 core ค่า sleeping time จะค่อยๆ เพิ่มและลดลงอีกครั้งเมื่อ CPU core มีขนาด 8 core ทำให้ได้กราฟที่มีลักษณะคล้ายระฆังคว่ำ ที่ค่า sleeping time ดีที่สุดเมื่อ CPU core ของ dom0 มีขนาด 1 core และ 8 core (สาเหตุที่ทำให้ CPU core ของ dom0 ขนาด 1 core และ 8 core ดีที่สุดจนทำให้เกิดกราฟลักษณะคล้ายระฆังคว่ำ ทางผู้วิจัยยังไม่ทราบสาเหตุแน่ชัด ซึ่งการสรุปสาเหตุดังกล่าวจะอยู่นอกเหนืองานวิจัยนี้ เพราะการสรุปสาเหตุนี้จำเป็นต้องใช้ความรู้พื้นฐานในการศึกษาเกี่ยวกับการทำงานของโปรเซสใน CPU core ดังนั้นในบทความนี้จึงยังไม่สามารถสรุปเหตุผลที่แน่ชัดได้)

## 5. วิจารณ์และสรุปผล

ผลการทดลองเมื่อนำมาวิเคราะห์ผลกระทบของ CPU core ต่อระบบบันทึกเหตุการณ์ แล้วจะสามารถแยกผลการวิเคราะห์ออกเป็น 2 ส่วน ได้ดังนี้

- ในส่วนที่ 1 การปรับขนาดของ CPU core ใน domU (ส่วนของผู้ใช้) หลังการทดลองปรับขนาด CPU core ขนาดต่างๆ แล้วผลปรากฏว่า logger มีประสิทธิภาพคงเดิมส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ไม่เปลี่ยนแปลง
- ในส่วนที่ 2 การปรับขนาดของ CPU core ใน dom0 (ส่วนของผู้ให้บริการ) ในงานเดิมของ Wongthai (Wongthai & Moorsel, 2016) ได้ให้ข้อเสนอแนะในงานวิจัยว่าหากมีการเพิ่ม CPU core ประสิทธิภาพของ logger จะดีขึ้น ส่งผลให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพดีขึ้น แต่ในการทดลองเมื่อทดสอบเพิ่ม CPU core ในส่วนของผู้ให้บริการผลการทดลองที่ได้ ประสิทธิภาพที่ดีที่สุดของ CPU core มีขนาดที่ 1 core และ 8

core ทำให้ได้ข้อสรุปว่าการเพิ่มขนาดของ CPU core ให้ในส่วนของผู้ให้บริการประสิทธิภาพของ logger ไม่ได้ดีขึ้นเสมอไป

วัตถุประสงค์หลักของงานวิจัยนี้คือการวิเคราะห์ผลกระทบของ CPU core ที่มีผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ เมื่อทำการทดลองทำให้ทราบว่าการปรับขนาด CPU core ในส่วนของผู้ใช้จะไม่ส่งผลกระทบต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ และเมื่อปรับขนาด CPU core ในส่วนของผู้ให้บริการจะมีผลกระทบต่อประสิทธิภาพในระบบบันทึกเหตุการณ์โดยประสิทธิภาพของระบบบันทึกเหตุการณ์ดีสุดในการทดลองเมื่อ CPU core ของ dom0 มีขนาด 1 core และ 8 core

## 6. กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนทุนอุดหนุนวิจัย จากคณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ มหาวิทยาลัยนเรศวร

## 7. เอกสารอ้างอิง

A. Saha. ( 2006) . Learning about linux process The Linux Gazette. [ online] . Available : http://linuxgazette.net/133/saha.html

Bryant R. E., & O'Hallaron D. R. (2010). *Computer Systems A Programmer's Perspective* (2nd Ed.). USA: Addison–Weslwy Publishing Company.

Carrick A. G., Levine D., & Elmsi R. (2010). *Operating Systems.* McGraw Hill: McGraw Hill.

Chan–in P., & Wongthai W. (2017). PERFORMANCE IMPROVEMENT CONSIDERATIONS OF CLOUD. *ICIC international.*

Chiba Z., Abghour N., Moussaid K., et al. (2016, 22–24 Sept. 2016). *A survey of intrusion detection systems for cloud computing environment.* Paper presented at the 2016 International Conference on Engineering & MIS (ICEMIS).

CSA. (2016a). The Treacherous 12 – Cloud Security Alliance. Cloud Security Alliance (pp. 1–35).

CSA. ( 2016b) . The treacherous 12 cloud computing top threats in 2016. The cloud Security Alliance,(CSA), Tech.REP.

Galvin P. B., Gagne G., & Silberschartz A. (2006). *Operating System Principles* (s. Education Ed.): John Wiley & Sons (Asia) Pte Ltd.

Molyneaux I. (2014). *The Art of Application Performance Testing From Strategy to Tools. United Startes of America: O'Reilly Media, Inc.,1005 Gravenstein Highway Nort, Sebastopol, CA 95472.*

Surbiryala J., Li C., & Rong C. (2017, 28-30 April 2017). *A framework for improving security in cloud computing.* Paper presented at the 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA).

Wongthai W., & Moorsel A. v. (2016). PERFORMANCE MEASUREMENT OF LOGGING SYSTEMS IN INFRASTRUCTURE AS A SERVICE CLOUD. *ICIC international*.

Wongthai W., Rocha F., & Moorsel A. v. (2013a, 16-19 Dec. 2013). *Logging Solutions to Mitigate Risks Associated with Threats in Infrastructure as a Service Cloud.* Paper presented at the 2013 International Conference on Cloud Computing and Big Data.

Wongthai W., Rocha F. L., & Moorsel A. v. (2013b, 25-28 March 2013). *A Generic Logging Template for Infrastructure as a Service Cloud.* Paper presented at the 2013 27th International Conference on Advanced Information Networking and Applications Workshops.

Wongthai W., & van Moorsel A. (2016). Quality analysis of logging system components in the cloud *Information Science and Applications (ICISA) 2016* (pp. 651-662): Springer.