กรอบการวัดประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์เพื่อบรรเทาภัยคุกคามต่อระบบการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐาน

ธงรบ อักษร

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2565

กรอบการวัดประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์เพื่อบรรเทาภัยคุกคามต่อระบบการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐาน

ธงรบ อักษร

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2565

วิทยานิพนธ์ เรื่อง "กรอบการวัดประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์เพื่อบรรเทาภัยคุกคามต่อระบบการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐาน"

ของ ธงรบ อักษร

ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบวิทยานิพนธ์

---------- ประธานกรรมการสอบวิทยานิพนธ์

(รองศาสตราจารย์ ดร.เอกรัฐ บุญเชียง)

---------- ประธานที่ปรึกษาวิทยานิพนธ์

(ผู้ช่วยศาสตราจารย์ ดร.วินัย วงษ์ไทย)

---------- กรรมการผู้ทรงคุณวุฒิภายใน

(รองศาสตราจารย์ ดร.จักรกฤษณ์ เสน่ห์ นมะหุต)

---------- กรรมการผู้ทรงคุณวุฒิภายใน

(ผู้ช่วยศาสตราจารย์ ดร.จันทร์จิรา พยัคฆ์เพศ)

---------- กรรมการผู้ทรงคุณวุฒิภายใน

(ผู้ช่วยศาสตราจารย์ ดร.ธนะธร พ่อค้า)

---------- กรรมการผู้ทรงคุณวุฒิภายใน

(ผู้ช่วยศาสตราจารย์ ดร.เกรียงศักดิ์ เตมีย์)

อนุมัติ

(รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์)

คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง

ผู้วิจัย ธงรบ อักษร

ประธานที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.วินัย วงษ์ไทย

ประเภทสารนิพนธ์ วิทยานิพนธ์ ปร.ด. วิทยาการคอมพิวเตอร์, มหาวิทยาลัยนเรศวร, 2565

คำสำคัญ ระบบรักษาความปลอดภัย, การประมวลผลแบบกลุ่มเมฆ, ระบบบันทึกเหตุการณ์, การทดสอบประสิทธิภาพ

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อที่เพิ่มความสามารถและเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์เพื่อบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อระบบการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐาน ผู้วิจัยจึงมีแนวคิดในการดำเนินการวิจัย 1) เพื่อปรับปรุงการทำงานของระบบบันทึกเหตุการณ์ 2) เพื่อเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ 3) เพื่อวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ โดยการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์เพื่อหาปัญหา ข้อจำกัด และดำเนินการปรับปรุง ผลการวิจัยพบว่า 1) สามารถเพิ่มประสิทธิภาพของการทำงานของระบบบันทึกเหตุการณ์โดยลดข้อจำกัดที่ได้จากการศึกษาระบบบันทึกเหตุการณ์เช่น การจัดการเวลาการทำงาน การจัดการหน่วยความจำ 2) สามารถเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์เพื่อใช้ในการพิจารณาหาผู้รับผิดชอบต่อการกระทำที่มีความผิดปกติได้อย่างถูกต้องและแม่นยำ 3) สามารถบอกถึงประสิทธิภาพการทำงานของผู้ใช้บริการเมื่อมีการใช้งานระบบบันทึกเหตุการณ์ โดยผู้ใช้บริการสามารถพิจารณาตัดสินใจเลือกรูปแบบการทำงานของระบบบันทึกเหตุการณ์ได้อย่างเหมาะสมตามความต้องการ

Title	A PERFORMANCE MEASUREMENT FRAMEWORK FOR IAAS LOGGING SYSTEM WITH THREAT MITIGATION
Author	Thongrob Auxsorn
Advisor	Assistant Professor Winai Wongthai, Ph.D.
Academic Paper	Ph.D. Dissertation in Computer Science, Naresuan University, 2022
Keywords	Security, Cloud computing, Longing system, Performance measurement

ABSTRACT

This research aims to performance and threat mitigation enhancements of logging systems for Infrastructure as a Service Cloud. The researcher, therefore, has a concept for conducting research 1) To improve the operation of the logging system. 2) To increase information in the log file from the operation of the logging system. 3) To measure and test users' computer work performance when using the logging system. The researcher tested the performance of the logging system operation for problems limitations and develop. The results showed that 1) the performance of the logging system could be increased by reducing the limitations obtained from the study of the logging systems, such as time operating management and memory management. 2) Able to increase information in the log file from the operation of the logging system to be used in determining the person responsible for the abnormal actions accurately and precisely. 3) Able to show the performance of computer operating of users when using the logging system. The users can consider and decide on the type of operation of the logging system according to their needs.

ประกาศคุณูปการ

การดำเนินการจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาของท่านผู้ช่วยศาสตราจารย์ ดร. วินัย วงษ์ไทย ประธานที่ปรึกษาวิทยานิพนธ์ ที่ได้สละเวลาอันมีค่ามาเป็นที่ปรึกษาได้ทุ่มเทอบรมสั่งสอนให้ความรู้ พร้อมทั้งให้คำปรึกษาในด้านเทคนิคต่างๆ ที่เกี่ยวข้อง ตลอดจนข้อเสนอแนะปรับปรุงพัฒนาการทำวิทยานิพนธ์ฉบับนี้ จนเสร็จสมบูรณ์เป็นอย่างดียิ่ง ผู้วิจัยต้องขอขอบพระคุณท่านเป็นอย่างสูง ไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณท่านคณาจารย์กรรมการสอบป้องกันวิทยานิพนธ์ทุกท่าน ประกอบด้วย ผู้ช่วยศาสตราจารย์ ดร. วินัย วงษ์ไทย ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร. จักรกฤษณ์ เสน่ห์ นมะหุต ผู้ช่วยศาสตราจารย์ ดร.เกรียงศักดิ์ เตมีย์ ผู้ช่วยศาสตราจารย์ ดร.จันทร์จิรา พยัคฆ์เพศ ผู้ช่วยศาสตราจารย์ ดร.ธนะธร พ่อค้า กรรมการผู้ทรงคุณวุฒิภายในทั้ง 4 ท่านดังกล่าว และรองศาสตราจารย์ ดร.เอกรัฐ บุญเชียง กรรมการผู้ทรงคุณวุฒิภายนอก ที่ได้กรุณาให้คำแนะนำข้อเสนอแนะ ปรับปรุงพัฒนาด้วยความเอาใจใส่ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์และทรงคุณค่า

ขอกราบขอบพระคุณอาจารย์ ดร.วันสุรีย์ มาศกรัม ที่ได้ให้คำปรึกษาทั้งด้านการนำเสนอผลงานและด้านการทำวิทยานิพนธ์จนทำให้ผู้วิจัยสามารถดำเนินการทำวิทยานิพนธ์ฉบับนี้ได้สำเร็จและมีความสมบูรณ์ยิ่ง ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ธนะธร พ่อค้า อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วมที่คอยให้คำปรึกษาและให้ความรู้เกี่ยวกับหลักการทฤษฎีต่างๆ ทางศาสตร์วิชาที่เกี่ยวข้องกับงานวิจัย

ขอขอบคุณรุ่นพี่และเพื่อนๆ ในสาขาวิชาวิทยาการคอมพิวเตอร์และสาขาวิชาเทคโนโลยีสารสนเทศ รวมถึงทุกๆคนที่มีส่วนร่วมในการทำวิทยานิพนธ์ฉบับนี้จนประสบผลสำเร็จ

ผู้วิจัยจึงขอมอบส่วนคุณงามความดีทั้งหลายให้แด่คณาจารย์ที่ได้ประสิทธิ์ประสาทวิชาจนทำให้ผลงานวิทยานิพนธ์เป็นประโยชน์ต่อผู้ที่เกี่ยวข้อง และขอบคุณแด่ บิดา มารดา และผู้มีพระคุณทุกท่านที่ให้การสนับสนุนในทุกๆ ด้าน จนได้บรรลุผลการเรียนในระดับดุษฎีบัณฑิตศึกษาครั้งนี้ สำหรับข้อบกพร่องต่างๆ ที่อาจจะเกิดขึ้นในงานวิทยานิพนธ์นั้น ผู้วิจัยขอน้อมรับและยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษาเพื่อเป็นประโยชน์ในการพัฒนางานวิจัยต่อไปในอนาคต

ธงรบ อักษร

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1

บทนำ

การประมวลแบบกลุ่มเมฆหรือคลาวด์คอมพิวติ้ง (Cloud computing) เป็นเทคโนโลยีที่มีการใช้งานในปัจจุบันอย่างแพร่หลายและมีแนวโน้มการใช้งานจะเพิ่มมากขึ้น โดยในงานวิทยานิพนธ์เล่มนี้จะใช้คำย่อว่า คลาวด์ (cloud) ซึ่งรูปแบบการให้บริการของคลาวด์จะเป็นให้การบริการทรัพยากรทางคอมพิวเตอร์ผ่านระบบอินเทอร์เน็ตหรืออินทราเน็ต โดยการให้บริการนั้นจะให้ผู้ใช้งานเช่าทรัพยากร เช่น พื้นที่ในการจัดเก็บข้อมูล โครงสร้างพื้นฐานทางคอมพิวเตอร์ ระบบปฏิบัติการ เครือข่ายอินเทอร์เน็ตและแอปพลิเคชัน การเข้าถึงการบริการเหล่านี้จะต้องทำผ่านระบบอินเทอร์เน็ตหรืออินทราเน็ต นอกจากนี้รูปแบบการให้บริการคลาวด์ยังมีความยืดหยุ่น สามารถลด เพิ่ม หรือยกเลิกการใช้บริการของคลาวด์ได้ตลอดเวลาและคำนวณอัตราค่าใช้จ่ายตามจริง ดังนั้นจะพบว่ามีองค์กรต่าง ๆ สนใจที่จะนำคลาวด์ไปประยุกต์ใช้งานทางด้านไอที (Information Technology: IT) เป็นจำนวนมาก เนื่องจากการใช้บริการคลาวด์นั้นสามารถลดต้นทุนค่าดูแลบำรุงรักษา เนื่องจากค่าบริการได้รวมค่าใช้จ่ายตามที่ใช้งานจริง เช่น ค่าจ้างพนักงาน ค่าซ่อมแซม ค่าลิขสิทธิ์ ค่าไฟฟ้า ค่าน้ำ ค่าน้ำมันเชื้อเพลิง ค่าอัพเกรด และค่าเช่าคู่สาย ลดความเสี่ยงการเริ่มต้น หรือการทดลองโครงการ สามารถลดหรือขยายได้ตามความต้องการ ได้ใช้ทรัพยากรที่มีประสิทธิภาพ มีระบบสำรองข้อมูลที่ดี มีเครือข่ายความเร็วสูง อยู่ภายใต้การดูแลของผู้เชี่ยวชาญ

แต่อย่างไรก็ตามปัญหาทางด้านความปลอดภัยของการใช้งานคลาวด์นั้นก็เป็นปัจจัยหลักสำหรับผู้ใช้บริการเนื่องจากข้อมูลขององค์กรจะถูกจัดเก็บไว้ในทรัพยากรที่ให้บริการโดยคลาวด์ซึ่งสิ่งนี้อาจทำให้องค์กรมีความเสี่ยงสูง อีกทั้งการเก็บข้อมูลในคลาวด์จะมีแนวโน้มที่จะถูกโจมตีจากผู้ไม่ประสงค์ดีโดยมีเป้าหมายเพื่อขโมย ทำลาย หาผลประโยชน์ จากข้อมูลเหล่านั้น เนื่องจากไม่มีสิ่งใดบนโลกอินเทอร์เน็ตที่มีความปลอดภัยอย่างสมบูรณ์และคลาวด์ก็เป็นบริการที่อยู่บนอินเทอร์เน็ต จึงทำให้มีองค์กรที่ได้ทำวิจัยเกี่ยวกับปัญหาทางด้านความปลอดภัยของคลาวด์เกิดขึ้น เช่น องค์กร Cloud Security Alliance (CSA) และจากปัญหาทางด้านความปลอดภัยของคลาวด์ที่ได้จากการวิจัยขององค์กร CSA ทำให้มีนักวิจัยได้ทำการศึกษาวิจัยเกี่ยวกับวิธีการป้องกันและวิธีการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ ซึ่งการป้องกันหรือบรรเทาปัจจัยเสี่ยงดังกล่าวจะสร้างความเชื่อมั่นต่อผู้ใช้บริการ

ในงานวิทยานิพนธ์นี้มุ่งเน้นที่จะปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยระบบบันทึกเหตุการณ์เป็นระบบที่สามารถช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ ลักษณะการทำงานของระบบบันทึกเหตุการณ์จะทำให้ทราบว่า บุคคลใดเข้าถึงไฟล์อะไรบ้างหรือได้กระทำอะไรกับไฟล์เหล่านั้นบ้าง การกระทำลักษณะนี้อาจเป็นส่วนหนึ่งของภัยคุกคามที่เรียกว่าการละเมิดข้อมูล (Data breach) ซึ่งเป็นภัยคุกคามที่รายงานขององค์กร CSA ให้อยู่ในลำดับที่ 1 ของภัยคุกคามทั้งหมดซึ่งกล่าวได้ว่าเป็นภัยคุกคามที่สำคัญและเกิดขึ้นบ่อยครั้ง ดังนั้นระบบบันทึกเหตุการณ์ดังกล่าวจะเป็นส่วนหนึ่งในการช่วยให้ผู้มีส่วนเกี่ยวข้องสามารถดำเนินการสืบสวนหาตัวบุคคลที่กระทำผิดมารับผิดชอบต่อการกระทำได้ การศึกษาการทำงานเกี่ยวกับระบบบันทึกเหตุการณ์เป็นสิ่งที่สำคัญของงานวิทยานิพนธ์นี้ การปรับปรุงและเพิ่มประสิทธิภาพการทำงานจะทำให้ระบบบันทึกเหตุการณ์สามารถทำงานได้อย่างรวดเร็ว ถูกต้องแม่นยำ มากขึ้น อีกทั้งยังช่วยลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงาน ทั้งในส่วนของผู้ให้บริการและผู้ใช้บริการ โดยรายละเอียดของบทนำผู้วิจัยได้จำแนกเป็นหัวข้อต่าง ๆ ดังนี้

ความเป็นมาและความสำคัญของปัญหา

ในหัวข้อนี้จะอธิบายความหมายและภาพรวมของคลาวด์ โครงสร้างพื้นฐานและประเภทของคลาวด์ ปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ ความสำคัญของการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ วิธีการที่ใช้บรรเทาปัจจัยเสี่ยง รวมถึงการปรับปรุงและเพิ่มประสิทธิภาพของการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ โดยหัวข้อนี้มีวัตถุประสงค์เพื่อให้รู้ถึงภาพรวมของวิทยานิพนธ์โดยมีรายละเอียด ดังนี้

1. **ความหมายและภาพรวมของคลาวด์**

คลาวด์ คือ รูปแบบการให้บริการทรัพยากรทางคอมพิวเตอร์ผ่านทางเน็ตเวิร์ก เช่น Amazon Web Service, Microsoft Azure, Google cloud, Dropbox, Microsoft Office 365 เป็นต้น โดยผู้ให้บริการได้มีการจัดเตรียมทรัพยากรไว้ให้ผู้ใช้บริการเลือกใช้ตามความต้องการ ผู้ใช้บริการสามารถเข้าใช้ทรัพยากรต่าง ๆ ได้ผ่านทางระบบอินเทอร์เน็ตหรืออินทราเน็ต สามารถเพิ่ม ลด และยกเลิกการใช้ทรัพยากรคอมพิวเตอร์ได้ ขั้นตอนในการดำเนินการขอใช้บริการของผู้ใช้มีความสะดวกรวดเร็วและติดต่อผู้ให้บริการน้อยที่สุด

ด้วยเหตุผลข้างต้นที่การเลือกใช้บริการคลาวด์จึงทำให้สามารถช่วยลดต้นทุนและค่าใช้จ่ายทางด้านการลงทุนทางด้านคอมพิวเตอร์และอุปกรณ์เครือข่ายขององค์กรต่าง ๆ ลงได้ อีกทั้งการให้บริการของคลาวด์มีความยืดหยุ่นสามารถเพิ่ม ลด และยกเลิกทรัพยากรที่ใช้งานได้ตามความต้อง เสียค่าใช้จ่ายตามบริการที่ใช้งาน ทำให้แนวโน้มในการการใช้งานคลาวด์มีอัตรา

การเติบโตที่รวดเร็ว เนื่องจากความต้องการขององค์กรที่เกิดการเปลี่ยนแปลงไปตามเศรษฐกิจไม่ว่าจะทำธุรกิจประสบความสำเร็จหรือล้มเหลว การลงทุนกับการใช้บริการคลาวด์ก็สามารถทำให้องค์กรสามารถควบคุมและกำหนดค่าใช้จ่ายที่ต้องลงทุนได้

คลาวด์ประกอบด้วย 3 ส่วนหลักโดยส่วนแรก คือ ผู้ให้บริการ (provider) ที่เป็นองค์กรขนาดใหญ่มีหน้าที่จัดการเตรียมทรัพยากรที่ มีประสิทธิภาพสูง มีขนาดใหญ่ และมีจำนวนมากไว้ให้บริการ ซึ่งทรัพยากรเหล่านี้ถือว่าเป็นส่วนที่สอง ในส่วนที่สองนี้จะประกอบไปด้วยฮาร์ดแวร์และซอฟต์แวร์ที่จำเป็นสำหรับผู้ให้บริการในสร้างให้เกิดบริการคลาวด์ โดยทรัพยากรที่ให้บริการจะอยู่ในรูปแบบของคอมพิวเตอร์เสมือน (Virtual Machine) ทำหน้าที่จัดสรรทรัพยากร จัดเตรียมแพลตฟอร์มหรือซอฟต์แวร์และแอปพลิเคชันเพื่อให้บริการ ส่วนสุดท้าย ผู้ใช้บริการ (Consumer) คือลูกค้าที่ขอใช้บริการทรัพยากร และเข้าใช้ทรัพยากรผ่านทางระบบอินเทอร์เน็ตหรืออินทราเน็ต

คลาวด์สามารถแบ่งตามลักษณะการให้บริการไว้ 3 ลักษณะ คือ คลาวด์แบบให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (Iaas) เป็นการให้บริการเฉพาะโครงสร้างพื้นฐานโดยมีการจัดเตรียม หน่วยประมวลผลกลาง หน่วยความจำหลัก หน่วยความจำสำรอง และระบบปฏิบัติการ มีประโยชน์ในการประมวลผล สร้างเครื่องแม่ข่าย และเครื่องคอมพิวเตอร์เพื่อให้บริการต่าง ๆ เช่น Microsoft Azure, Dropbox, Google Drive for business, Amazon Web Services ลักษณะที่สองเรียกว่า คลาวด์แบบให้บริการด้านแพลตฟอร์ม (PaaS) การให้บริการด้านแพลตฟอร์มสำหรับผู้ใช้งานเช่น ผู้พัฒนาโปรแกรมหรือแอปพลิเคชันที่ทำงานด้าน ซอฟต์แวร์และแอปพลิเคชันโดยผู้ให้บริการคลาวด์จะจัดเตรียมสิ่งที่จำเป็นต้องใช้ในการพัฒนา เช่น Database Server, Web Application เป็นต้น ลักษณะสุดท้าย คลาวด์แบบให้บริการด้านซอฟต์แวร์ (SaaS) คือการให้บริการด้านซอฟต์แวร์และแอปพลิเคชันผ่านทางอินเทอร์เน็ต คล้ายกับการเช่าใช้ คิดค่าบริการตามลักษณะการใช้งาน เช่น Microsoft Office 365, Dropbox, Google G suit เป็นต้น

จากประเภทของคลาวด์ที่แบ่งตามลักษณะการให้บริการนั้นจะเห็นได้ว่าการให้บริการของคลาวด์แบบให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์นั้นสามารถจัดการทรัพยากรได้มากที่สุดตั้งแต่ หน่วยประมวลผลกลาง หน่วยความจำหลัก หน่วยความจำสำรอง ระบบปฏิบัติการ ต่างกับคลาวด์แบบให้บริการด้านแพลตฟอร์มที่สามารถใช้งานแพลตฟอร์มที่ติดตั้งไว้ให้บนระบบปฏิบัติการเพื่อให้สามารถพัฒนางานได้ แต่ไม่สามารถจัดการทรัพยากรอื่น ๆ ได้ และสุดท้ายคลาวด์แบบให้บริการด้านซอฟต์แวร์นั้นให้บริการการใช้งานซอฟต์แวร์และ

แอปพลิเคชันเท่านั้น จากข้อมูลข้างต้นในวิทยานิพนธ์นี้จะมุ่งเน้นการทำวิจัยกับบริการคลาวด์แบบให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์

การนำคลาวด์มาใช้งานนั้นจะมีปัญหาความเสี่ยงที่อาจก่อให้ภัยคุกคามต่อความปลอดภัยของคลาวด์จึงได้มีองค์กรที่ทำการวิจัย รวบรวม และระบุปัญหาเกี่ยวกับภัยคุกคามต่อคลาวด์เกิดขึ้นซึ่งมีชื่อว่า Cloud Security Alliance หรือ CSA (CSA, 2016a, 2016b) โดยปัญหาที่ทางองค์กร CSA ได้นำมาเผยแพร่ไว้นั้นจะถูกอธิบายเพิ่มเติมในหัวข้อถัดไป

2. ปัญหาภัยคุกคามของคลาวด์

ปัญหาภัยคุกคามเป็นปัญหาและอุปสรรคที่สำคัญต่อการใช้งานคลาวด์ ในหัวข้อนี้จึงได้มีการอธิบายและนำเสนอเพิ่มเติมเกี่ยวกับปัญหาภัยคุกคามของคลาวด์ รวมถึงปัญหาภัยคุกคามที่ผู้วิจัยสนใจที่จะบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์

องค์กร CSA ได้ทำการรวบรวม นำเสนอ และเผยแพร่ข้อมูลของปัญหาที่เกี่ยวกับภัยคุกคามต่อคลาวด์โดยได้จัดทำรายงานที่มีชื่อว่า Top Threats Cloud Computing V1.0 - Cloud Security Alliance ซึ่งเป็นรายงานฉบับแรกที่จัดทำขึ้นในปี 2010 (CSA, 2010) โดยรายงานฉบับแรกดังกล่าวทาง CSA ได้ระบุปัญหาภัยคุกคามของคลาวด์ไว้ 7 ข้อ ในปี 2013 จึงได้จัดทำรายงานที่ชื่อว่า The Notorious Nine: Cloud Computing Top Threats in 2013 (CSA, 2013) เป็นรายงานเกี่ยวกับภัยคุกคามต่อคลาวด์ 9 หัวข้อ ซึ่งมีการเพิ่มภัยคุกคามของคลาวด์จากรายงานฉบับแรกที่มีเพียง 7 หัวข้อ ในปี 2016 ซึ่งเป็นรายงานฉบับที่ 3 ชื่อว่า The Treacherous 12 Cloud Computing Top Threats in 2016 (CSA, 2013) ซึ่งในรายงานฉบับล่าสุดนี้ทางองค์กร CSA ได้ระบุเกี่ยวกับภัยคุกคามที่เกิดขึ้นต่อคลาวด์เพิ่มเป็น 12 หัวข้อ และล่าสุดในปี 2019 ใช้ชื่อรายงานว่า Top Threats to Cloud Computing: Egregious Eleven (CSA, 2013) ซึ่งได้นำเสนอภัยคุกคามไว้ 11 หัวข้อ โดยจะกล่าวถึงรายละเอียดเพิ่มเติมในบทที่ 2 เรื่อง เอกสารและงานวิจัยที่เกี่ยวข้อง ในข้อเรื่อง ปัญหาด้านความปลอดภัยของคลาวด์

จากภัยคุกคามที่เกิดขึ้นต่อคลาวด์ใน 11 หัวข้อตามรายงาน Top Threats to Cloud Computing: Egregious Eleven ในปี 2019 ขององค์กร CSA จะพบว่าหัวข้อภัยคุกคามลำดับที่ 1 คือ การละเมิดข้อมูล (Data breach) ซึ่งผู้วิจัยเห็นว่ามีความสำคัญจึงมีแนวคิดเพื่อบรรเทาปัญหาภัยคุกคามนี้ โดยจะใช้ระบบบันทึกเหตุการณ์ในการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ ซึ่งภาพรวมของระบบบันทึกเหตุการณ์จะถูกกล่าวหัวข้อถัดไปเรื่องระบบบันทึกเหตุการณ์ และรายละเอียดของระบบบันทึกเหตุการณ์จะถูกกล่าวในบทที่ 2 ในหัวข้อเรื่องระบบบันทึกเหตุการณ์ (logging system)

3. ระบบบันทึกเหตุการณ์และการนำระบบบันทึกเหตุการณ์ไปใช้เพื่อการวิจัย

ระบบบันทึกเหตุการณ์เป็นระบบที่ใช้เพื่อการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ ผู้วิจัยได้นำมาใช้ในการทดลองและทำการปรับปรุงการทำงานของระบบบันทึกเหตุการณ์ เพื่อเพิ่มประสิทธิภาพในการใช้งานทั้งผู้ให้บริการและผู้ใช้บริการ โดยในหัวข้อนี้จะอธิบายรายละเอียดเกี่ยวกับระบบบันทึกเหตุการณ์ ดังนี้

ระบบบันทึกเหตุการณ์จะเป็นระบบที่ทำหน้าที่บันทึกเหตุการณ์ที่เกิดขึ้นในคลาวด์ ซึ่งระบบบันทึกเหตุการณ์นี้ มีผู้วิจัยมากมายได้ทำการพัฒนาสร้างระบบบันทึกเหตุการณ์ ที่สามารถติดตั้งไว้ที่ฝั่งของผู้ให้บริการ หรือผู้ใช้บริการ หรือทั้งสองฝั่ง ซึ่งมีหลักการทำงานที่แตกต่างกัน ในวิทยานิพนธ์นี้ผู้วิจัยเลือกใช้ระบบบันทึกเหตุการณ์ที่ติดตั้งในส่วนของผู้ให้บริการ เนื่องจากการทำงานของระบบบันทึกเหตุการณ์ลักษณะนี้มีผลกระทบต่อการใช้งานผู้ใช้บริการน้อยมากเมื่อเปรียบเทียบกับระบบบันทึกเหตุการณ์ลักษณะอื่น ๆ การทำงานของระบบบันทึกเหตุการณ์จะใช้กระบวนการบันทึกเหตุการณ์โดยวิธี Introspection โดยวิธีการนี้ได้ถูกนำมาใช้ในงานของ Lata & Singh (2022) ซึ่งเป็นเทคนิคสำหรับการตรวจสอบการทำงานของโพรเซสที่เกิดขึ้นในเครื่องคอมพิวเตอร์เสมือน (Virtual Machine)[1] โดยการตรวจสอบข้อมูลต่าง ๆ ผ่านทางหน่วยความจำหลัก หรือเรียกว่าการ Intrusion ดังงานของ Mayuranathan, Saravanan, Muthusenthil & Samydurai (2022) โดยในระบบบันทึกเหตุการณ์นี้จะมีโพรเซสที่ทำหน้าที่ตรวจสอบการทำงานของโพรเซสที่เกิดขึ้นในเครื่องคอมพิวเตอร์เสมือนเรียกว่า ล็อกเกอร์ (logger)[2] ในการทำงานหากข้อมูลในโพรเซสที่ตรวจสอบนั้นตรงตามเงื่อนไขที่กำหนดไว้ จะทำการบันทึกข้อมูลเหตุการณ์ของการเข้าถึงไฟล์ข้อมูลที่เกิดขึ้นในคอมพิวเตอร์เสมือน เช่น การอ่าน การเปลี่ยนแปลงแก้ไข การสำเนา การคัดลอก และการลบไฟล์ข้อมูลของผู้ใช้บริการบนคลาวด์ ซึ่งการบันทึกข้อมูลเหตุการณ์จะถูกเก็บไว้ใน ล็อกไฟล์ (log files)[3] เพื่อนำไปใช้พิจารณาเพื่อค้นหาผู้รับผิดชอบเมื่อเกิดปัญหาหรือเกิดความเสียหาย

ในวิทยานิพนธ์เล่มนี้มุ่งเน้นเรื่องการปรับปรุงและเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ให้สามารถทำงานได้อย่างรวดเร็ว ถูกต้อง แม่นยำ มากขึ้น อีกทั้งยังช่วยลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงาน ทั้งในส่วนของผู้ให้บริการ และผู้ใช้บริการ โดยปรับปรุง

---

[1] คอมพิวเตอร์เสมือน หรือ Virtual Machine เป็นเครื่องคอมพิวเตอร์ที่ถูกสร้างขึ้นภายใต้คลาวด์

[2] ล็อกเกอร์ หรือ logger เป็นโพรเซสที่ถูกรันจากคำสั่งโค้ดในภาษา C เพื่อใช้สำหรับบันทึกเหตุการณ์ที่เกิดขึ้นต่อคลาวด์

[3] ล็อกไฟล์ หรือ log files เป็นแหล่งสำหรับเก็บข้อมูลจากการทำงานของล็อกเกอร์

กระบวนการทำงานของล็อกเกอร์ เพิ่มจำนวนข้อมูลในล็อกไฟล์และวัดประสิทธิภาพในการทำงาน เพื่อให้เหมาะสมและตรงตามความต้องการของผู้ใช้บริการ จะถูกกล่าวในหัวข้อถัดไป

4. การพัฒนาระบบบันทึกเหตุการณ์

การปรับปรุงและเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ให้สามารถทำงานได้อย่างรวดเร็ว ถูกต้อง แม่นยำ มากขึ้น อีกทั้งยังช่วยลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงาน ทั้งในส่วนของผู้ให้บริการ และผู้ใช้บริการนั้นผู้วิจัยจะใช้กระบวนการวิธีของวงจรการพัฒนาระบบ (SDLC : System development Life Cycle) (Molyneaux, 2014) เพื่อเป็นแนวทางในการปรับปรุงและพัฒนาระบบเพื่อเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ หัวข้อนี้จะอธิบายรายละเอียดเบื้องต้นเกี่ยวกับการศึกษาการทำงาน ทดลองประสิทธิภาพของระบบบันทึกเหตุการณ์และเครื่องมือที่ใช้ในการทดลองของวิทยานิพนธ์เล่มนี้

จากหัวข้อเรื่องระบบบันทึกเหตุการณ์ ในเรื่องความเป็นมาและความสำคัญได้กล่าวว่างานวิทยานิพนธ์เล่มนี้มุ่งเน้นเรื่องปรับปรุงและเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ ให้สามารถทำงานได้อย่างรวดเร็ว ถูกต้อง แม่นยำ มากขึ้น อีกทั้งยังช่วยลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงาน ทั้งในส่วนของผู้ให้บริการ และผู้ใช้บริการ โดยในย่อหน้านี้จะอธิบายเกี่ยวกับรูปแบบการทดสอบประสิทธิภาพเพิ่มเติม คือ จากงานของ Molyneaux ได้อธิบายว่า Key Performance Indicator หรือ KPI สำหรับการทดสอบประสิทธิภาพมี 4 หัวข้อ คือ 1) สภาพพร้อมใช้งาน (availability) 2) เวลาตอบสนอง (response time) 3) ปริมาณงานที่ทำในช่วงเวลาหนึ่ง (throughput) และ 4) ความจุ (capacity) (Molyneaux, 2014) การทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ที่ใช้ในงานวิทยานิพนธ์เล่มนี้ จะคำนึงถึง 2 ด้านด้วยกันคือ เวลาตอบสนอง หรือ response time และปริมาณงานที่ทำในช่วงเวลาหนึ่ง หรือ throughput โดยการวัดประสิทธิภาพเรื่องเวลาตอบสนองในส่วนของกระบวนการทำงานของล็อกเกอร์ และปริมาณงานที่ทำในช่วงเวลาหนึ่งกับการเก็บข้อมูลในล็อกไฟล์

ในย่อหน้าที่ผ่านมาได้อธิบายรายละเอียดเบื้องต้นเกี่ยวกับการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ และการทดลองที่คาดว่าจะดำเนินการปฏิบัติ รวมถึงแนวทางเบื้องต้นสำหรับการวิเคราะห์ข้อมูลที่เกี่ยวกับระบบบันทึกเหตุการณ์ต่อประสิทธิภาพที่เกิดขึ้น การนำข้อมูลที่ผ่านกระบวนการวิเคราะห์มาสรุปหาสาเหตุเกี่ยวกับผลกระทบด้านประสิทธิภาพที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์ เมื่อทราบสาเหตุดังกล่าวแล้ว จะทำให้ผู้วิจัยสามารถพิจารณาเกี่ยวกับประสิทธิภาพของระบบบันทึกเหตุการณ์ได้หลังมีการปรับปรุงกระบวนการทำงานของล็อกเกอร์และเพิ่มจำนวนข้อมูลของล็อกไฟล์ อีกทั้งยังสามารถวางแผนปรับปรุงประสิทธิภาพในขั้นตอน

ต่อไปได้ ดังนั้น กระบวนการทดสอบประสิทธิภาพด้วยวิธีการนำข้อมูลที่ได้ไปวิเคราะห์ หาสาเหตุและหาวิธีการปรับปรุงประสิทธิภาพ จึงสามารถอธิบายรายละเอียดได้ ดังนี้

1. ทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์เดิม

2. นำข้อมูลจากการทดสอบไปวิเคราะห์ สรุปหาสาเหตุเกี่ยวกับผลกระทบที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์

3. หาวิธีการปรับปรุงที่ได้จากการวิเคราะห์ สรุปหาสาเหตุเกี่ยวกับผลกระทบที่เกิดขึ้นต่อระบบบันทึกเหตุการณ์

4. ทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์หลังจากมีการปรับปรุง

สำหรับการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์นั้นสิ่งที่จำเป็น คือ ต้องมีข้อสรุปและผลวิเคราะห์ของกระบวนการทำงานของล็อกเกอร์ในระบบบันทึกเหตุการณ์ และข้อมูลของล็อกไฟล์ เพื่อนำข้อสรุปและผลวิเคราะห์ดังกล่าวไปพิจารณาและหาวิธีปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ ดังนั้นการวิเคราะห์กระบวนการทำงานที่ส่งผลกระทบต่อล็อกเกอร์ในระบบบันทึกเหตุการณ์จึงมีความสำคัญและจำเป็นต้องมีการทดสอบประสิทธิภาพ เช่น ทดสอบเกี่ยวกับผลกระทบของการจัดการหน่วยความจำ ความรวดเร็วในการทำงานของล็อกเกอร์ และอีกส่วนที่สำคัญคือการพิจารณาข้อมูลที่อยู่ในล็อกไฟล์นั้นมีเพียงพอต่อการนำไปใช้พิจารณาหาความผิดปกติหรือไม่ โดยข้อมูลจากการดำเนินการเหล่านี้ สามารถนำไปเป็นข้อสรุปและวิเคราะห์ผลกระทบที่มีต่อระบบบันทึกเหตุการณ์ เพื่อนำไปเป็นข้อมูลพื้นฐานสำหรับการออกแบบเพื่อปรับปรุงประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ดังกล่าวนี้ต่อไป

5. **ปัญหาวิจัยหลักของวิทยานิพนธ์** (research gaps)

เป้าหมายหลักของวิทยานิพนธ์ คือ สิ่งที่ผู้วิจัยได้ทำการศึกษาและได้ระบุเพื่อให้บรรลุวัตถุประสงค์ของงานวิจัยและสามารถออกแบบการทดลองเพื่อปรับปรุงในการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ และลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงานทั้งในส่วนของผู้ให้บริการ

รายละเอียดของปัญหาวิจัยหลักหรือ research gap ของวิทยานิพนธ์เล่มนี้สามารถอธิบายเป็นรายข้อได้ 3 หัวข้อ โดย research gap ทุกหัวข้อจะสอดคล้องกับวัตถุประสงค์ทั้งหมดของงานวิทยานิพนธ์เล่มนี้ รายละเอียดของวัตถุประสงค์จะถูกกล่าวในหัวข้อถัดไปหรือหัวข้อเรื่องวัตถุประสงค์ของการศึกษา ซึ่ง research gap จะมีรายละเอียด ดังนี้

1. จากการศึกษาวรรณกรรมที่เกี่ยวข้อง (literature reviews) ที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์และทดลองใช้งาน พบว่าระบบบันทึกเหตุการณ์ยังมีข้อจำกัดเกี่ยวกับการทำงาน

ในการจัดการทรัพยากรเช่นหน่วยความจำจะมีการเพิ่มขนาดจนไม่สามารถทำงานได้และกระบวนการทำงานของล็อกเกอร์ซึ่งเป็นโพรเซสที่สำคัญในการทำงานของระบบบันทึกเหตุการณ์เกี่ยวกับระยะเวลาในการตรวจสอบเหตุการณ์ที่ใช้เวลามาก

2. จากการศึกษาวรรณกรรมที่เกี่ยวข้อง (literature reviews) ที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์และทดลองใช้งาน พบว่าข้อมูลที่ถูกจัดเก็บไว้ในล็อกไฟล์ของระบบบันทึกเหตุการณ์นั้นยังไม่ครอบคลุมการพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูลของผู้ใช้

3. จากการศึกษาวรรณกรรมที่เกี่ยวข้อง (literature reviews) ที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์และทดลองใช้งาน ยังไม่ปรากฏการวัดและทดสอบประสิทธิภาพการทำงานในส่วนของผู้ใช้บริการเมื่อระบบบันทึกเหตุการณ์กำลังทำงานในเวลาเดียวกัน โดยเมื่อมีการใช้งานระบบบันทึกเหตุการณ์จะมีผลกระทบกับการทำงานโดยรวมของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการ ซึ่งจะทำการวัดและทดสอบทำงานโดยรวมของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการว่าลดลงมากน้อยเพียงใด อีกทั้งยังจะนำเสนอรูปแบบการใช้งานระบบบันทึกเหตุการณ์ให้ตรงตามความต้องการของผู้ใช้บริการ ในส่วนของผลกระทบการทำงานโดยรวมของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการที่เหมาะสม

6. สรุปภาพรวม

หัวข้อนี้จะกล่าวถึงภาพรวมทั้งหมดของความเป็นมาและความสำคัญของปัญหาโดยมีรายละเอียด ดังนี้

6.1 ตามหัวข้อเรื่อง ความหมายและภาพรวมของคลาวด์ ในหัวข้อนี้ได้อธิบายเกี่ยวกับนิยามและความสำคัญของคลาวด์ โดยลักษณะการให้บริการของคลาวด์มีความยืดหยุ่นสามารถเพิ่มหรือลดทรัพยากรที่ใช้งานได้ตามความต้องการของผู้ใช้บริการ ด้วยลักษณะดังกล่าวทำให้มีองค์กรต่าง ๆ ให้ความสนใจต้องการนำคลาวด์มาประยุกต์ใช้งานด้านเทคโนโลยีสารสนเทศภายในองค์กร นอกจากนี้การนำคลาวด์มาประยุกต์ใช้งานภายในองค์กรสิ่งที่ต้องคำนึงถึงคือความปลอดภัย เมื่อความปลอดภัยเป็นสิ่งที่ต้องคำนึงถึงในการนำคลาวด์มาใช้ในองค์กร จึงทำให้มีองค์กรที่ได้ทำวิจัยและระบุปัญหาเกี่ยวกับภัยคุกคามของคลาวด์ซึ่งก็คือองค์กร CSA

6.2 ตามหัวข้อเรื่อง ปัญหาภัยคุกคามของคลาวด์ โดยปัญหาเกี่ยวกับภัยคุกคามนั้นเป็นปัญหาสำคัญของคลาวด์ ดังที่กล่าวไว้ในบทสรุปในหัวข้อเรื่องความหมายและภาพรวมของคลาวด์ โดยปัญหาที่เกี่ยวกับภัยคุกคามต่อคลาวด์ที่ทาง CSA ได้วิจัยอย่างต่อเนื่องจนถึงปี 2016 ทางองค์กร CSA ได้ระบุปัญหาภัยคุกคามไว้ 12 หัวข้อและเรียงลำดับตามความรุนแรง โดย

รายงานฉบับดังกล่าวผู้วิจัยสนใจที่จะบรรเทาภัยคุกคามที่อาจก่อให้เกิดปัญหาต่อคลาวด์ ด้วยวิธีการใช้ระบบบันทึกเหตุการณ์ โดยรายละเอียดของระบบบันทึกเหตุการณ์สามารถสรุปภาพรวมของลักษณะการทำงานได้ในหัวข้อเรื่อง ระบบบันทึกเหตุการณ์

6.3 ตามหัวข้อเรื่อง ระบบบันทึกเหตุการณ์ ในหัวข้อนี้ผู้วิจัยได้อธิบายหลักการทำงานเบื้องต้นของระบบบันทึกเหตุการณ์ว่า สามารถบันทึกเหตุการณ์ที่เกิดขึ้นเกี่ยวกับการเข้าถึงไฟล์ข้อมูลบนคลาวด์ เพื่อนำเหตุการณ์ที่บันทึกไปเป็นหลักฐานเพื่อหาบุคคลที่กระทำผิดมารับผิดชอบต่อสิ่งที่เกิดขึ้น โดยในงานวิจัยนี้มุ่งเน้นที่เพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ด้วยการปรับปรุงกระบวนการทำงานของล็อกเกอร์และเพิ่มข้อมูลของล็อกไฟล์

6.4 ตามหัวข้อเรื่อง การพัฒนาระบบบันทึกเหตุการณ์ โดยการปรับปรุงและเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ให้สามารถทำงานได้อย่างรวดเร็ว ถูกต้องแม่นยำ มากขึ้น อีกทั้งยังช่วยลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงานทั้งในส่วนของผู้ให้บริการและผู้ใช้บริการ เป็นจุดประสงค์หลักของวิทยานิพนธ์เล่มนี้ ดังนั้นสิ่งสำคัญในการทดสอบประสิทธิภาพ คือ กระบวนการทำงานของล็อกเกอร์ ข้อมูลในล็อกไฟล์ และการวัดประสิทธิภาพของระบบการทำงานที่มีผลต่อผู้ให้บริการและผู้ใช้บริการ ซึ่งข้อมูลดังกล่าวจะสามารถนำไปวิเคราะห์หาสาเหตุของผลกระทบที่มีต่อระบบบันทึกเหตุการณ์ และเมื่อทราบสาเหตุดังกล่าวจะสามารถออกแบบวิธีการปรับปรุงประสิทธิภาพระบบบันทึกเหตุการณ์ได้

6.5 ตามหัวข้อเรื่อง ปัญหาวิจัยหลัก (research gap) หัวข้อนี้ได้ระบุเป้าหมายของวิทยานิพนธ์เล่มนี้ เพื่อให้สอดคล้องและบรรลุวัตถุประสงค์ของงานวิทยานิพนธ์

วัตถุประสงค์ของการศึกษา

จากหัวข้อเรื่อง ความเป็นมาและความสำคัญของปัญหา ได้กล่าวถึงความสำคัญของประสิทธิภาพของระบบและวิเคราะห์ผลกระทบที่มีต่อระบบบันทึกเหตุการณ์ ซึ่งมีความสำคัญและเป็นวัตถุประสงค์ของวิทยานิพนธ์เล่มนี้ ในหัวข้อนี้จึงได้ระบุวัตถุประสงค์ทั้งหมดของวิทยานิพนธ์เล่มนี้ ซึ่งจะสอดคล้องกับปัญหาวิจัยหลักของวิทยานิพนธ์ (research gaps) ที่กล่าวในหัวข้อปัญหาวิจัยหลักทั้งสามข้อโดยรายละเอียดเกี่ยวกับวัตถุประสงค์ของการศึกษา มีดังนี้

1. เพื่อเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ในด้านการจัดการหน่วยความจำและความรวดเร็วของกระบวนการทำงาน

2. เพื่อเพิ่มข้อมูลในล็อกไฟล์ของระบบบันทึกเหตุการณ์เกี่ยวกับการพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูลของผู้ใช้บริการ

3. การวัดและประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ของผู้ให้บริการที่มีผลต่อการใช้งานของผู้ใช้บริการ

ขอบเขตของการวิจัย

เพื่อให้บรรลุวัตถุประสงค์ของงานวิจัยในวิทยานิพนธ์เล่มนี้ ผู้วิจัยจึงได้กำหนดขอบเขตการวิจัย โดยมีรายละเอียดดังหัวข้อต่อไปนี้

1. **ขอบเขตด้านเทคโนโลยี**

1) เครื่องคอมพิวเตอร์ (personal computer) Intel I5 1 เครื่อง สำหรับการทดลองสร้างคลาวด์ โดยติดตั้งระบบปฏิบัติการ Linux Fedora 16 ขนาด 64 bit

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

3) ติดตั้งระบบบันทึกเหตุการณ์ (logging system) บนคอมพิวเตอร์ Intel I5 2.4 GH สำหรับผู้ให้บริการ[4] ซึ่งใช้จำลองสร้างเครื่องคอมพิวเตอร์เสมือน (Virtual Machine) บนคลาวด์ IaaS และทดสอบการทำงาน

2. **ขอบเขตด้านวิธีการ**

2.1 ทำการจำลองสร้างคลาวด์ บนเครื่องคอมพิวเตอร์ 1 เครื่องพร้อมติดตั้งระบบบันทึกเหตุการณ์ (logging system) เมื่อทำการทดลองและทดสอบจะเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางและขนาดของหน่วยความจำ เพื่อหาผลกระทบที่เกิดขึ้นในการทำงานงานของระบบบันทึกเหตุการณ์ ที่ส่งผลต่อประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์

2.2 ทำการจำลองพฤติกรรมผู้ไม่ประสงค์ดีหรือแฮกเกอร์ (hacker) เข้าถึงไฟล์ข้อมูลบนคลาวด์ในขณะที่ระบบบันทึกเหตุการณ์ทำงาน เมื่อทำการทดลองและทดสอบการทำงานของระบบบันทึกเหตุการณ์ จำเป็นต้องสร้างเหตุการณ์การเข้าถึงไฟล์ข้อมูลที่ระบบบันทึกเหตุการณ์ตรวจสอบเพื่อให้เกิดล็อกไฟล์หรือข้อมูลเหตุการณ์

2.3 ทำการวัดประสิทธิภาพของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการเมื่อมีการใช้งานระบบบันทึกเหตุการณ์โดยใช้โปรแกรม Sysbench หาเวลาในการประมวลผลในขณะที่เครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการยังไม่มีการใช้งานระบบบันทึกเหตุการณ์และมีการใช้งานระบบบันทึกเหตุการณ์

---

[4] ผู้ให้บริการ หรือ provider เป็นผู้ที่เตรียมบริการประมวลผลแบบกลุ่มเมฆไว้ เช่น เน็ตเวิร์ก เครื่องแม่ข่าย เครื่องคอมพิวเตอร์ แหล่งเก็บข้อมูล แอปพลิเคชันและบริการอื่นๆ ให้กับผู้ใช้บริการ

นิยามศัพท์เฉพาะ

นิยามศัพท์เฉพาะเป็นคำที่ถูกกำหนดเพื่อช่วยในการสืบค้นและสามารถเข้าถึงงานวิจัยในวิทยานิพนธ์เล่มนี้ได้สะดวกรวดเร็วยิ่งขึ้น โดยนิยามศัพท์เฉพะของวิทยานิพนธ์ คือ การรักษาความปลอดภัย การประมวลผลแบบกลุ่มเมฆ ระบบบันทึกเหตุการณ์และการทดสอบประสิทธิภาพ โดยมีรายละเอียดดังนี้

1. **การรักษาความปลอดภัย** (Security)

การรักษาความปลอดภัย (Security) หมายความว่า ขั้นตอน เทคนิค วิธีการรักษาดูแล และป้องกันอันตรายต่อข้อมูลหรือการทำงาน ซึ่งการรักษาความปลอดภัยของข้อมูลถือว่าเป็นเรื่องที่สำคัญ เนื่องจากข้อมูลถือว่าเป็นความลับของผู้ใช้บริการ ผู้ที่สามารถเข้าถึงข้อมูลได้จะต้องเป็นผู้ที่มีสิทธิ์เข้าถึงหรือเจ้าของข้อมูลเท่านั้น ดังนั้นการรักษาความปลอดภัยของข้อมูลจึงเป็นสิ่งที่สำคัญ นอกจากนี้การรักษาความปลอดภัยถือว่าเป็นกระบวนการหนึ่งที่จะช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดปัญหาภัยคุกคามต่อคลาวด์ ดังนั้นการรักษาความปลอดภัยจึงเป็นเรื่องที่สำคัญสำหรับวิทยานิพนธ์เล่มนี้

2. **การประมวลผลแบบกลุ่มเมฆ** (cloud computing or cloud)

การประมวลผลแบบกลุ่มเมฆหรือคลาวด์ หมายความว่า การบริการที่ครอบคลุมถึงการให้ใช้งานหน่วยจัดเก็บข้อมูล โครงสร้างพื้นฐานทางคอมพิวเตอร์หรืออื่น ๆ จากผู้ให้บริการโดยการเข้าถึงต้องผ่านระบบอินเทอร์เน็ตหรืออินทราเน็ต โดยรูปแบบของคลาวด์จะสามารถลดความยุ่งยากในการติดตั้ง ดูแลระบบ ทำให้สามารถช่วยประหยัดเวลา และลดต้นทุนในการสร้างระบบคอมพิวเตอร์และเครือข่าย โดยมีทั้งรูปแบบเสียค่าบริการและไม่เสียค่าบริการ ในวิทยานิพนธ์เล่มนี้ได้นำเสนอวิธีการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ ซึ่งระบบบันทึกเหตุการณ์ดังกล่าวยังเป็นวิธีการหนึ่งที่ช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามที่มีต่อคลาวด์ ดังนั้นการทราบข้อมูลพื้นฐานของคลาวด์หรือการประมวลผลแบบกลุ่มเมฆจึงมีความสำคัญ โดยข้อมูลพื้นฐานนิยามประเภทของคลาวด์จะถูกกล่าวในบทที่ 2 เอกสารที่เกี่ยวข้องหัวข้อเรื่อง ความเป็นมาและนิยามของคลาวด์

3 **โครงสร้างพื้นฐานทางคอมพิวเตอร์** (Infrastructure as a service: IaaS)

โครงสร้างพื้นฐานทางคอมพิวเตอร์(Infrastructure as a service: IaaS) หมายความว่าเป็นการให้บริการทรัพยากรคอมพิวเตอร์พื้นฐาน เช่น หน่วยประมวลผล หน่วยความจำหลัก หน่วยความจำสำรอง และการให้บริการคลาวด์ลักษณะ IaaS เป็นโครงสร้างพื้นฐานของคลาวด์อีกสองแบบคือ คลาวด์แบบการให้บริการด้านแพลตฟอร์ม(Platform as a Service: PaaS) และคลาวด์แบบการให้บริการด้านซอฟต์แวร์ (Software as a Service: SaaS) ซึ่งคลาวด์นั้นมีการใช้

งานได้หลากหลายและเหมาะสมกับงานต่าง ๆ มีทั้งความยืดหยุ่นในการใช้งานเป็นการให้บริการกับผู้ใช้บริการทางฝั่ง Consumer Side และ ผู้ให้บริการคือฝั่ง Provider Side แต่ในเล่มวิทยานิพนธ์นี้ผู้จัดทำได้ทำการเลือก IaaS ซึ่งการให้บริการคลาวด์แบบนี้เป็นเทคโนโลยีที่ถูกนำมาทดแทนการใช้งานคอมพิวเตอร์แบบดั้งเดิมและมีแนวโน้มในการเติมโตอย่างรวด

4 **ระบบบันทึกเหตุการณ์** (logging system)

ระบบบันทึกเหตุการณ์ หมายความว่า ระบบที่สามารถบันทึกข้อมูลความเป็นไปในระบบคลาวด์ เช่น การเข้าถึง ปรับปรุง เปลี่ยนแปลง แก้ไขและลบข้อมูล ข้อมูลที่ได้จากการบันทึกข้อมูลของระบบบันทึกเหตุการณ์นี้ สามารถนำไปใช้เป็นหลักฐานในการสืบหาผู้รับผิดชอบเมื่อเกิดปัญหาหรือความเสียหายต่อคลาวด์ และระบบบันทึกเหตุการณ์เป็นวิธีการหนึ่งที่สามารถช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ซึ่งถือว่าระบบบันทึกเหตุการณ์เป็นวิธีการที่สำคัญของวิทยานิพนธ์เล่มนี้ โดยรายละเอียดเพิ่มเติมเกี่ยวกับระบบบันทึกเหตุการณ์จะกล่าวในบทที่ 2 เอกสารที่เกี่ยวข้อง หัวข้อเรื่อง ระบบบันทึกเหตุการณ์

5. **ภัยคุกคาม** (Threats)

ภัยคุกคาม หมายความว่า ผู้ใช้บริการคลาวด์มีไฟล์ข้อมูลที่สำคัญ อาจจะเกิดผู้ที่ไม่ประสงค์ดีต้องการที่จะนำข้อมูลของผู้ใช้บริการคลาวด์ไปทำลายหรือนำไปใช้ประโยชน์ในทางที่ไม่ดีอาจมีผลกระทบต่อผู้ใช้บริการคลาวด์

6 **โครงสร้างข้อมูลไอโหนด** (inode structure)

โครงสร้างข้อมูลไอโหนด คือ โครงสร้างข้อมูลที่เก็บข้อมูลการกระทำของผู้ใช้ที่มีไฟล์ข้อมูลจากเดิมให้มีเพิ่มมากขึ้นจากเดิมที่มีข้อมูลเพียง ชื่อไฟล์ หมายเลขโพรเซส ชื่อโพรเซสและหมายเลขผู้ใช้ ซึ่งจะอธิบายข้อมูลเพิ่มเติมในบทที่ 2 เอกสารที่เกี่ยวข้อง

7. **การทดสอบประสิทธิภาพ** (performance measurement)

การทดสอบประสิทธิภาพเป็นปัจจัยสำคัญของวงจรพัฒนาระบบ SDLC (SDLC : System development Life Cycle) ถ้าหากไม่มีการทดสอบประสิทธิภาพในระบบ จะส่งผลให้องค์กรขาดผลกำไรตามที่ขาดหวังได้ เนื่องจากการทดสอบประสิทธิภาพทำให้ทราบข้อมูลเกี่ยวกับสมรรถนะของระบบจนสามารถวางแผนต่อผลกำไรล่วงหน้าขององค์กรได้ นอกจากนี้การทดสอบประสิทธิภาพถือว่าเป็นวัตถุประสงค์หลักของงานวิจัยเล่มนี้เพราะว่าวัตถุประสงค์ของงานวิจัยนี้ต้องมีการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์และนำไปวิเคราะห์ผลกระทบเพื่อเป็นข้อมูลสำหรับการออกแบบเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ จากนั้นจึงจะสามารถดำเนินการขั้นต่อไปได้ โดยรายละเอียดเพิ่มเติมของหัวข้อนี้จะถูกอธิบายเพิ่มเติมในบทที่ 2 เอกสารที่เกี่ยวข้อง

8. **เวลารอคอยสำหรับการทำงาน** (sleeping time)

เวลารอคอยสำหรับการทำงาน ในวิทยานิพนธ์นี้จะขอใช้คำว่า sleeping time ซึ่งขอแบ่งออกเป็นสองส่วนดังนี้ sleeping time ของระบบบันทึกเหตุการณ์เดิม และ sleeping time ของระบบบันทึกเหตุการณ์ใหม่ที่ได้มีการปรับปรุงกระบวนการทำงาน ในส่วน sleeping time ของระบบบันทึกเหตุการณ์เดิมจะเป็นเวลารอคอยของการเกิดเหตุการณ์ ที่จะต้องหน่วงเวลาให้เหมาะสมกับการตรวจสอบเหตุการณ์ของระบบบันทึกเหตุการณ์เนื่องจากระบบบันทึกเหตุการณ์เดิมมีข้อจำกัดทางระยะเวลาในการตรวจสอบเหตุการณ์ เช่นเหตุการณ์ที่เกิดขึ้นจะต้องมีระยะห่างกันไม่น้อยกว่า 60 millisecond และในส่วนของระบบบันทึกเหตุการณ์ใหม่ที่ได้รับการปรับปรุงสามารถตรวจสอบเหตุการณ์ที่เกิดขึ้นได้ทุกช่วงเวลา ดังนั้นเวลารอคอยสำหรับการทำงาน (sleeping time) จะเป็นการหน่วงเวลาการตรวจสอบเหตุการณ์ของระบบบันทึกเหตุการณ์ เพื่อทำให้เครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการสามารถทำงาน เนื่องจากตามหลักการทำงานของระบบบันทึกเหตุการณ์จะหยุดการทำงานของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการเพื่อทำการตรวจสอบเหตุการณ์ หากว่าไม่มีการกำหนด sleeping time ที่เหมาะสมจะส่งผลกระทบต่อการทำงานของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการทำให้มีประสิทธิภาพการทำงานลดลง

จากนิยามศัพท์เฉพาะของวิทยานิพนธ์ที่ผู้วิจัยได้ระบุ ซึ่งถูกอธิบายไว้ในหัวข้อดังกล่าว โดยทุก ๆ หัวข้อของนิยามศัพท์เฉพาะทางผู้วิจัยได้ศึกษางานวิจัยที่เกี่ยวข้องและอธิบายเพิ่มเติมในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง

สมมติฐานการวิจัย

เพื่อให้บรรลุวัตถุประสงค์ของการศึกษาที่กล่าวไว้ในหัวข้อเรื่อง วัตถุประสงค์ของการศึกษาของบทที่ 1 บทนำ ผู้วิจัยได้ทำการตั้งสมมติฐานไว้ดังนี้

1. การปรับปรุงกระบวนการทำงานและการจัดการทรัพยากรของล็อกเกอร์ส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ทำงานได้ดีขึ้น

2. การวัดประสิทธิภาพในส่วนของผู้ใช้บริการเมื่อมีการใช้งานระบบบันทึกเหตุการณ์ส่งผลให้ผู้ใช้สามารถกำหนดทรัพยากรในการทำงานได้ตรงความต้องการ สร้างความเชื่อมั่นและความพึงพอใจต่อการใช้งานของผู้ใช้บริการที่มีต่อการใช้บริการ

3. การเพิ่มข้อมูลในล็อกไฟล์ของระบบบันทึกเหตุการณ์สามารถใช้เป็นข้อมูลในการตรวจสอบและพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูลของผู้ใช้บริการได้อย่างถูกต้อง

กรอบการดำเนินงานวิจัย

จากวัตถุประสงค์ของงานวิจัยในหัวข้อเรื่อง วัตถุประสงค์ของการศึกษาทั้ง 3 หัวข้อและเพื่อให้บรรลุวัตถุประสงค์ทั้ง 3 หัวข้อนั้น ผู้วิจัยได้กำหนดกรอบแนวคิดการวิจัยเพื่อให้บรรลุวัตถุประสงค์ดังกล่าวโดยมีรายละเอียดดังนี้

วิทยานิพนธ์เล่มนี้มุ่งเน้นเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์โดยการปรับปรุงกระบวนการทำงานและการจัดการทรัพยากรของล็อกเกอร์และเพิ่มข้อมูลของล็อกไฟล์ และทดลองประสิทธิภาพการทำงานของผู้ใช้เมื่อระบบบันทึกเหตุการณ์ทำงานเพื่อให้ผู้ใช้บริการได้ทรัพยากรที่เหมาะสมในการทำงาน สามารถแบ่งการทดลองและการออกแบบระบบเพื่อเพิ่มประสิทธิภาพได้ 3 ส่วนหลัก คือ 1) ส่วนของการทดลองและนำผลการทดลองที่ได้มาสรุป และวิเคราะห์ผลการทดลอง 2) ส่วนของการออกแบบวิธีการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ 3) การทดลองวัดประสิทธิภาพการทำงานในส่วนของผู้ใช้บริการเพื่อเลือกใช้ทรัพยากรได้อย่างเหมาะสม โดยสามารถอธิบายรายละเอียดของ 3 ส่วนหลักดังกล่าวได้ ดังหัวข้อ สรุปภาพรวม ของบทที่ 1

1. **ส่วนของการทดลองและนำผลทดลองที่ได้มาสรุป วิเคราะห์ผลการทดลอง**

ในการทดลองได้ออกแบบการทดลองเป็น 3 ส่วนเพื่อให้สอดคล้องกับวัตถุประสงค์ที่ 1.2.1 1.2.2 และ 1.2.3 ตามลำดับดังนี้ คือ การทดลองใช้และวัดประสิทธิภาพระบบบันทึกเหตุการณ์เดิมเพื่อหาข้อจำกัด ศึกษาข้อมูลคุณลักษณะของไฟล์เพื่อนำมาใช้ในการพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูล และทดสอบวัดประสิทธิในส่วนของการทำงานของผู้ใช้บริการเมื่อเปิดใช้งานระบบบันทึกเหตุการณ์ โดยมีรายละเอียดดังนี้

1.1 การทดลองใช้และวัดประสิทธิภาพของระบบบันทึกเหตุการณ์เดิมซึ่งทำงานในส่วนของผู้ให้บริการ แล้วสรุปผลและวิเคราะห์ผลการทดลองใช้และวัดประสิทธิภาพ

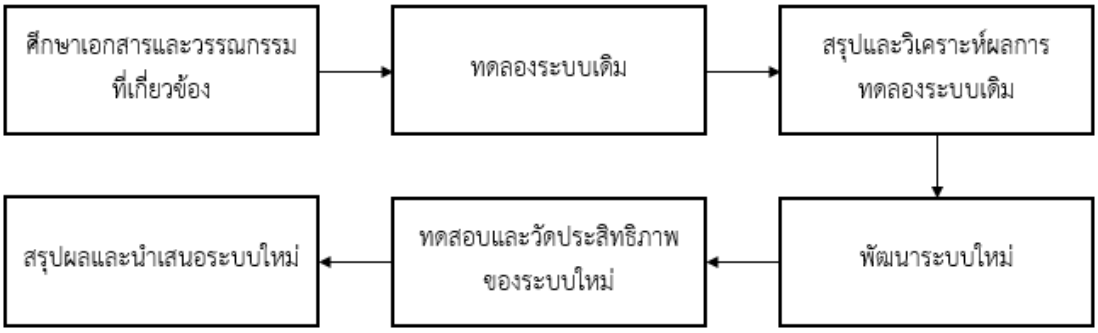
1.2 ศึกษาข้อมูลคุณลักษณะของไฟล์ข้อมูลเพื่อนำข้อมูลเหล่านั้นมาใช้ในการพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูล แล้วสรุปผลและวิเคราะห์ผลการศึกษา

1.3 ทดสอบวัดประสิทธิภาพในส่วนของการทำงานของผู้ใช้บริการเมื่อเปิดใช้งานระบบบันทึกเหตุการณ์ แล้วสรุปผลและวิเคราะห์ผลการวัดประสิทธิภาพ

2. **ส่วนของการออกแบบวิธีการเพิ่มประสิทธิภาพของระบบ**

หลังจากทำการทดลอง หัวข้อส่วนของการทดลองและนำผลทดลองที่ได้มาสรุปวิเคราะห์ผลการทดลองในหัวข้อที่ผ่านมา เมื่อได้ผลการวิเคราะห์แล้ว จะนำข้อมูลทั้งหมดที่ได้มาสรุปผล หาสาเหตุที่เกิดขึ้นจากนั้นออกแบบวิธีการปรับปรุงเพื่อเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ต่อไป

ในหัวข้อเรื่อง กรอบการดำเนินการวิจัย ผู้วิจัยได้สรุปภาพรวมเป็นภาพความสัมพันธ์ของกรอบการดำเนินการวิจัยของวิทยานิพนธ์นี้ ดังภาพ 1 (NSA, 2013; Parkin & Morgan, 2012)



ภาพ 1 กรอบแนวคิดการวิจัย

จากภาพ 1 กรอบแนวคิดการวิจัย สามารถอธิบายรายละเอียดเป็นลำดับขั้นตอนได้ ดังนี้

1. ศึกษาเอกสารและวรรณกรรมที่เกี่ยวข้อง โดยผู้วิจัยได้ศึกษางานวิจัยต่าง ๆ เพื่อนำไปเป็นข้อมูลสำหรับการออกแบบ พัฒนา และปรับปรุงระบบบันทึกเหตุการณ์

2. ทดลองระบบเดิม หลังจากที่ได้ศึกษาเอกสารและวรรณกรรมที่เกี่ยวข้องแล้วผู้วิจัยจึงได้นำระบบมาทดลองเพื่อหาข้อจำกัด

3. สรุปและวิเคราะห์ผลการทดลองระบบเดิม จากการทดลองใช้งานระบบเดิม ผู้วิจัยได้รวบรวมข้อมูลต่าง ๆ จากการทดลองใช้ นำมาสรุปและวิเคราะห์ข้อมูลเพื่อหาวิธีการและแนวทางในการพัฒนา

4). พัฒนาระบบใหม่ ผู้วิจัยได้นำข้อมูลที่ได้จากขั้นตอน สรุปและวิเคราะห์ผลการทดลองระบบเดิม มาเพื่อทำการพัฒนาระบบใหม่ที่มีประสิทธิภาพที่ดีขึ้น

5. ทดสอบและวัดประสิทธิภาพของระบบใหม่ หลังจากการพัฒนาระบบใหม่ผู้วิจัยนำระบบที่พัฒนาขึ้นใหม่มาทดลองและวัดประสิทธิภาพการทำงาน

6. สรุปผลและนำเสนอระบบใหม่ หลังจากได้ผลการทดลอง นำผลการทดลองมาเปรียบเทียบกันระหว่างระบบก่อนการปรับปรุงและหลังการปรับปรุงแล้ว ผู้วิจัยจะสรุปผลการปรับปรุงที่ได้และนำเสนอผลการเปลี่ยนแปลงที่เกิดขึ้น

โดยรวมของบทที่ 1 บทนำได้กล่าวถึงความเป็นมาของคลาวด์ ปัญหาด้านความปลอดภัยของคลาวด์ การรักษาความปลอดภัยของคลาวด์ ระบบบันทึกเหตุการณ์ซึ่งเป็นเครื่องมือที่สามารถลดปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ การปรับปรุงประสิทธิภาพของ

ระบบบันทึกเหตุการณ์ซึ่งเป็นวัตถุประสงค์หลักของงานวิทยานิพนธ์เล่มนี้อีกด้วย รวมถึงการตั้งสมมติฐานที่เกี่ยวข้อง ที่ส่งผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์ รวมไปถึงวิธีการเพิ่มประสิทธิภาพให้กับระบบบันทึกเหตุการณ์ โดยข้อมูลข้างต้นที่กล่าวมา ผู้วิจัยได้เขียนอธิบายเพิ่มเติมในบทที่ 2 ทฤษฎีที่เกี่ยวข้อง

ประโยชน์ที่คาดว่าจะได้รับ

จากที่กล่าวไว้ในหัวข้อเรื่อง วัตถุประสงค์ของการศึกษา ได้กล่าวว่าในงานทดลองของวิทยานิพนธ์เล่มนี้ จะนำเสนอวิธีการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ ดังนั้นประโยชน์ของงานวิจัยนี้จึงมีความสำคัญต่อการนำไปพิจารณาเพื่อทำให้ระบบบันทึกเหตุการณ์สามารถนำไปใช้งานได้จริงต่อไปโดยมีรายละเอียด ดังนี้

1. ระบบบันทึกเหตุการณ์ มีการทำงานที่รวดเร็ว และสามารถจัดการทรัพยากรได้อย่างมีประสิทธิภาพ
2. ระบบบันทึกเหตุการณ์ สามารถตรวจสอบความผิดปกติของการเข้าถึงไฟล์ข้อมูลเพื่อหาผู้รับผิดชอบได้อย่างถูกต้อง และแม่นยำ
3. ผู้ใช้สามารถเลือกใช้ทรัพยากรได้เหมาะสมกับตรงตามความต้องการใช้งานในขณะที่มีการใช้งานระบบบันทึกเหตุการณ์
4. สร้างความพึงพอใจและความมั่นใจในการใช้บริการคลาวด์ให้กับผู้ใช้บริการที่มีต่อผู้ให้บริการ

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

สำหรับเนื้อหาทั้งหมดในบทที่ 2 นี้ ผู้วิจัยจะขอนำเสนอเรื่องราวที่เกี่ยวข้องกับเอกสารและงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์ในเรื่องระบบบันทึกเหตุการณ์ (logging system) คลาวด์ประเภทการให้บริการโครงสร้างพื้นฐานคอมพิวเตอร์ เป็นวิธีการที่สำคัญที่ช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ ตามรายงานเรื่องภัยคุกคามในคลาวด์ขององค์กร CSA แสดงให้เห็นว่าคลาวด์นั้นมีจุดอ่อนและความเสี่ยงในการถูกโจมตีเนื่องจากการทำงานของคลาวด์นั้นจะถูกดำเนินการผ่านทางเครือข่ายคอมพิวเตอร์ ซึ่งภัยคุกคามที่เกิดขึ้นอาจเกิดจากการขาดความรู้ความเข้าใจ ความรอบคอบในการใช้งานของผู้ใช้บริการหรืออาจเป็นการโจมตีของผู้ไม่ประสงค์ดีทำให้เกิดความเสียหายต่อธุรกิจและการทำงาน ระบบบันทึกเหตุการณ์ มีกลไกที่สำคัญ ทำหน้าที่เก็บข้อมูลการกระทำต่างๆที่เกิดขึ้นต่อไฟล์ข้อมูลของผู้ใช้บริการเรียกว่า ล็อกไฟล์ (log file) และจากข้อมูลในล็อกไฟล์นี้สามารถนำไปใช้เป็นหลักฐานในการสืบสวน (Investigate) และตรวจสอบ (Detection) เพื่อหาบุคคลหรือผู้เกี่ยวข้องที่ก่อให้เกิดความเสียหายมารับผิดชอบต่อการกระทำที่เกิดขึ้น (Accountability) และวิธีลักษณะนี้สามารถนำไปสนับสนุนการทำงานของวิธีการป้องกัน (Preventive) ไม่ให้เกิดเหตุการณ์ขึ้นได้ ในวิทยานิพนธ์นี้เนื้อหาทั้งหมดผู้วิจัยได้แบ่งเนื้อหาที่เกี่ยวข้องตามหัวข้อย่อยดังนี้

1. ข้อมูลพื้นฐานของคลาวด์
2. คลาวด์ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์
3. ปัญหาภัยคุกคามของคลาวด์
4. วิธีการรักษาความปลอดภัยของการละเมิดข้อมูล
5. ล็อกไฟล์
6. ระบบบันทึกเหตุการณ์
7. libVMI
8. โครงสร้างข้อมูลไอโหนด (inode Struct)
9. วิธีการได้ผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์
10. Sysbench
11. สรุปภาพรวมบทที่ 2

ข้อมูลพื้นฐานของคลาวด์

ในหัวข้อนี้ทางผู้วิจัยจะอธิบายเกี่ยวกับ ลักษณะการทำงานคลาวด์ ประเภทของคลาวด์ รูปแบบการให้บริการของคลาวด์ และ ประโยชน์ของคลาวด์ เพื่อให้ผู้อ่านได้เข้าใจและเห็นภาพรวมทั้งหมดของคลาวด์ เพื่อเป็นข้อมูลพื้นฐานในการทำความเข้าใจเกี่ยวกับคลาวด์

การประมวลผลแบบกลุ่มเมฆ (cloud computing) ซึ่งในวิทยานิพนธ์เล่มนี้เราจะใช้คำว่า คลาวด์ เป็นบริการที่ให้ผู้ใช้บริการใช้หรือเช่าใช้ระบบคอมพิวเตอร์หรือทรัพยากรด้านคอมพิวเตอร์ของผู้ให้บริการ โดยครอบคลุมทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการประมวลผล การจัดเก็บข้อมูล และระบบออนไลน์ต่าง ๆ ผ่านอินเตอร์เน็ต ซึ่งเราสามารถเลือกกำลังการประมวลผล เลือกจำนวนทรัพยากร ได้ตามความต้องการในการใช้งานจากผู้ให้บริการระบบคลาวด์เช่น Amazon Web Services Microsoft Azure Google Cloud Platform เป็นต้น แทนการซื้อ การเป็นเจ้าของ รวมถึงการดูแลรักษาศูนย์ข้อมูลจริงและเซิร์ฟเวอร์ โดยองค์กรทุกประเภท ทุกขนาด และทุกภาคอุตสาหกรรมใช้คลาวด์ในงานที่หลากหลาย เช่น การสำรองข้อมูล การประมวลผลข้อมูล การพัฒนาและการทดสอบซอฟต์แวร์ การวิเคราะห์ข้อมูลขนาดใหญ่ และบริการซอฟท์แวร์หรือแอปพลิเคชัน จากรายงานของ Gartner (2022) คลาวด์เป็นนวัตกรรมรูปแบบใหม่ที่ผู้ใช้บริการสามารถเข้าใช้งานได้หลายช่องทางตามความต้องการ ส่งผลทำให้อุตสาหกรรมด้านไอทีเกิดการเปลี่ยนแปลงรูปแบบการทำงานอย่างรวดเร็ว ผลจากการเปลี่ยนแปลงทำให้ธุรกิจคลาวด์เจริญเติบโตขึ้นอย่างรวดเร็ว คลาวด์เข้ามามีบทบาทต่อธุรกิจโลกมากขึ้น กระบวนการทางธุรกิจเกือบทั้งหมดบนโลกทำงานอยู่บนคลาวด์ คลาวด์ คือ อนาคตของโครงสร้างพื้นฐาน และกุญแจสู่ความสำเร็จในโลกธุรกิจที่ไร้พรมแดน คาดการณ์มูลค่าใช้จ่ายบริการคลาวด์สาธารณะของผู้ใช้ปลายทางปี 2565 พุ่งสูงเกือบ 5 แสนล้านดอลลาร์ เติบโต 20% เพิ่มจากปี 2564 มูลค่า 4.1 แสนล้านดอลลาร์ และคาดว่าปี 2566 มูลค่าจะเพิ่มขึ้น 6 แสนล้านดอลลาร์

ประโยชน์ของการประมวลผลบนคลาวด์จากข้อมูลของ Amazon Web Services (2022)

1. ความคล่องตัว ระบบคลาวด์ช่วยให้เข้าถึงความหลากหลายของเทคโนโลยี ซึ่งช่วยให้สามารถสร้างสรรค์นวัตกรรมได้รวดเร็วยิ่งขึ้นและสร้างงานได้แทบทุกอย่างตามจินตนาการ สามารถหมุนเวียนทรัพยากรจากบริการโครงสร้างพื้นฐาน เช่น การประมวลผล (Processing) พื้นที่จัดเก็บ (Storage) ฐานข้อมูล (Database) อินเทอร์เน็ตในทุกสิ่ง (Internet of Thing) การเรียนรู้ของโปรแกรมด้วยตัวเอง (Machine Learning) และแหล่งข้อมูลขนาดใหญ่ (Big Data)

รวมถึงการวิเคราะห์ และอีกมากมายได้รวดเร็วอย่างที่ต้องการ สามารถปรับใช้บริการเทคโนโลยีได้ภายในไม่กี่นาที และนำแนวคิดมาทำให้ลำดับต่างๆ ของขนาดเป็นจริงได้เร็วยิ่งขึ้น โดยสิ่งนี้จะให้อิสระในการทดลอง ทดสอบแนวคิดใหม่เพื่อทำให้ประสบการณ์ของผู้ใช้บริการเกิดแนวทางใหม่แตกต่าง และเปลี่ยนโฉมธุรกิจ

2. ความยืดหยุ่น ด้วยการประมวลผลบนระบบคลาวด์นี้ไม่จำเป็นต้องจัดเตรียมทรัพยากรล่วงหน้าเพื่อดำเนินกิจกรรมทางธุรกิจในระดับสูง แต่จะจัดเตรียมจำนวนทรัพยากรเท่าที่ต้องการตามจริง สามารถขยายทรัพยากรให้มากขึ้นหรือน้อยลงตามขนาดที่เติบโตและหดตัวอย่างกะทันหันเมื่อธุรกิจต้องการการเปลี่ยนแปลง

3. ประหยัดค่าใช้จ่าย คลาวด์ทำให้ผู้ใช้บริการเปลี่ยนค่าใช้จ่ายคงที่ เช่น การลงทุนกับพื้นที่จัดเก็บข้อมูล เครื่องเซิร์ฟเวอร์ เป็นค่าใช้จ่ายผันแปร และเสียค่าใช้จ่ายด้านไอทีเท่าที่ใช้งาน นอกจากนี้ยังลดค่าใช้จ่ายในการดูแลและบำรุงรักษา

4. ปรับใช้งานได้ทั่วโลก ด้วยคลาวด์นี้สามารถขยายไปยังพื้นที่ทางภูมิศาสตร์ใหม่และปรับใช้ทั่วโลกในไม่กี่นาที ยกตัวอย่างเช่น AWS มีโครงสร้างพื้นฐานทั่วโลก จึงสามารถปรับใช้งานแอปพลิเคชันของคุณในหลายภูมิภาคทั่วโลกได้ในไม่กี่คลิก การนำแอปพลิเคชันมาใกล้ชิดกับผู้ใช้งานให้มากขึ้นช่วยลดความล่าช้าและปรับปรุงประสบการณ์ใช้งานของผู้ใช้

คลาวด์แบ่งออกเป็นสามประเภทหลักประกอบด้วย

1. ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (IaaS: Infrastructure as a Service) ประกอบด้วยการสร้างบล็อกพื้นฐานสำหรับงานไอที ซึ่งโดยทั่วไปจะมอบสิทธิ์การเข้าถึงคุณสมบัติเครือข่าย คอมพิวเตอร์เสมือนจริงหรือบนฮาร์ดแวร์เฉพาะ และพื้นที่จัดเก็บข้อมูล IaaS มอบความยืดหยุ่นและการควบคุมจัดการทรัพยากรด้านไอทีในระดับสูงสุดให้แก่ผู้ใช้บริการ โดยมีความคล้ายคลึงกับทรัพยากรไอทีที่มีอยู่ซึ่งแผนกไอทีและนักพัฒนาคุ้นเคยในปัจจุบัน

2. ประเภทการให้บริการแพลตฟอร์ม (PaaS: Platform as a Service) ช่วยลดความต้องการในการจัดการโครงสร้างพื้นฐานขององค์กร โดยทั่วไปหมายถึงฮาร์ดแวร์และระบบปฏิบัติการ และช่วยให้มุ่งเน้นไปที่การปรับใช้และการจัดการแอปพลิเคชัน สิ่งนี้ช่วยให้คุณมีประสิทธิภาพการทำงานมากขึ้นเนื่องจากไม่จำเป็นต้องกังวลเรื่องการจัดซื้อทรัพยากร การวางแผนขีดความสามารถ การบำรุงรักษาซอฟต์แวร์ การอัพเดท หรืองานที่ยากและซับซ้อนอื่นๆ ที่ไม่เกี่ยวข้องกับการทำงานของแอปพลิเคชัน

3. ประเภทสุดท้าย การให้บริการซอฟต์แวร์ (SaaS: Software as a Service) ให้บริการผลิตภัณฑ์ที่สมบูรณ์ซึ่งทำงานและได้รับการจัดการ โดยไม่จำเป็นต้องคิดถึงวิธีการดูแลรักษา การบริการหรือการจัดการโครงสร้างพื้นฐาน เพียงแค่ต้องคิดถึงวิธีที่นำซอฟต์แวร์นั้นๆไปใช้งาน

คลาวด์ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์

คลาวด์ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ หรือ Infrastructure as a Service ซึ่งในวิทยานิพนธ์เล่มนี้จะเรียกว่า คลาวด์ IaaS เป็นรูปแบบธุรกิจที่นำเสนอโครงสร้างพื้นฐานด้านไอที เช่นการประมวลผล การจัดเก็บ และทรัพยากรเครือข่าย บนพื้นฐานค่าบริการตามการใช้งานจริงผ่านทางอินเทอร์เน็ต สามารถใช้ คลาวด์ IaaS เพื่อขอและกำหนดทรัพยากรที่ต้องการในการทำงานเกี่ยวกับคอมพิวเตอร์และและระบบไอที ผู้ใช้บริการมีหน้าที่รับผิดชอบทำการปรับใช้ บำรุงรักษา และสนับสนุนแอปพลิเคชัน ซอฟท์แวร์ และระบบปฏิบัติการที่ติดตังบนคลาวด์ IaaS ส่วนการรับผิดชอบในการบำรุงรักษาโครงสร้างพื้นฐานทางกายภาพ เป็นหน้าที่ของผู้ให้บริการ จึงทำให้มีความยืดหยุ่นและควบคุมทรัพยากรด้านไอทีได้อย่างคุ้มค่า

ความสำคัญของการใช้งาน คลาวด์ IaaS โดยทั่วไป องค์กรต่างๆ จะซื้อและดูแลรักษาอุปกรณ์คอมพิวเตอร์เพื่อใช้งานประมวลผลข้อมูลภายในองค์กร การดำเนินการลักษณะนี้ต้องใช้เงินลงทุนจำนวนมาก เพื่อจัดการกับงานที่มีความซับซ้อนหรือต้องการการประมวลผลที่มีประสิทธิภาพสูง ซึ่งการลงทุนนี้จะต้องมีค่าใช้จ่ายเรื่องพลังงานไฟฟ้า การดูแลและซ่อมบำรุง อายุการใช้งานของอุปกรณ์ และในบางช่วงเวลาต้องการเพิ่มความสามารถให้เพิ่มขึ้น องค์กรต้องลงทุนเพื่อจัดซื้อเพิ่มเติมและหากเป็นการใช้งานในช่วงระยะเวลาสั้นๆ ก็จะพบว่าเป็นการลงทุนที่ไม่คุ้มค่า เช่น มหาวิทยาลัยมีการเข้าดูข้อมูลแอปพลิเคชันและการลงทำเบียนเพิ่มขึ้นในช่วงการลงทะเบียนเรียน ในการจัดการปริมาณการทำงาน ต้องมีการเพิ่มความสามารถของเซิร์ฟเวอร์เพิ่มเติมจากการใช้งานปกติ เพื่อรองรับการทำงาน ก็จำเป็นต้องลงทุนเพื่อเพิ่มความสามารถ บริษัทอีคอมเมิร์ซมีการเข้าชมแอปพลิเคชันเพิ่มขึ้นสามเท่าในช่วงเทศกาลวันหยุด ในการจัดการปริมาณการเข้าชมนี้ พวกเขาต้องซื้อเครื่องเซิร์ฟเวอร์เพิ่มเติมซึ่งไม่ได้มีการใช้งานตลอดช่วงที่เหลือของปี จากตัวอย่างที่กล่าวมา ผู้ให้บริการระบบคลาวด์ จะดูแลศูนย์ข้อมูลที่มีความปลอดภัยสูงด้วยอุปกรณ์ฮาร์ดแวร์จำนวนมาก สามารถให้บริการเข้าถึงโครงสร้างพื้นฐานของการประมวลผลบนคลาวด์ โดยจ่ายค่าบริการตามการใช้งานจริง สามารถเข้าถึงทรัพยากรแบบไม่จำกัดได้อย่างคล่องตัวและปลอดภัย เพื่อให้คุณสามารถตอบสนองความต้องการทางธุรกิจ และสามารถปรับเพิ่มหรือลดทรัพยากรได้ตลอดเวลา

ประโยชน์ของคลาวด์ IaaS สามารถจำแนกออกเป็นข้อๆจากข้อมูลของ Amazon Web Services (2022) ที่เป็นผู้ให้บริการที่ครอบคลุมและใช้กันอย่างแพร่หลายมากที่สุดในโลก โดยนำเสนอบริการอันโดดเด่นเต็มรูปแบบกว่า 200 บริการจากศูนย์ข้อมูลต่างๆ ทั่วโลก ลูกค้าหลายล้านราย ซึ่งรวมถึงบริษัทสตาร์ทอัพที่เติบโตเร็วที่สุด องค์กรที่ใหญ่ที่สุด และหน่วยงานรัฐบาลชั้นนำต่างใช้ AWS เพื่อลดต้นทุน เพิ่มความคล่องตัว และสร้างสรรค์สิ่งใหม่ๆ ได้อย่างรวดเร็ว ดังนี้

1. ความเร็ว ผู้ใช้บริการสามารถจัดเตรียมทรัพยากรจำนวนเท่าใดก็ได้ภายในไม่กี่นาที ติดตั้งระบบทดสอบการทำงาน และนำไปใช้งานได้อย่างรวดเร็ว

2. ประสิทธิภาพ ผู้ให้บริการระบบคลาวด์มีศูนย์ข้อมูลแบบกระจายตามภูมิศาสตร์ ทั่วโลก ซึ่งสามารถทำงานในตำแหน่งที่ใกล้กับผู้ใช้บริการมากขึ้น ทำให้เพิ่มประสิทธิภาพในการประมวลผล และลดเวลาแฝงของเครือข่าย

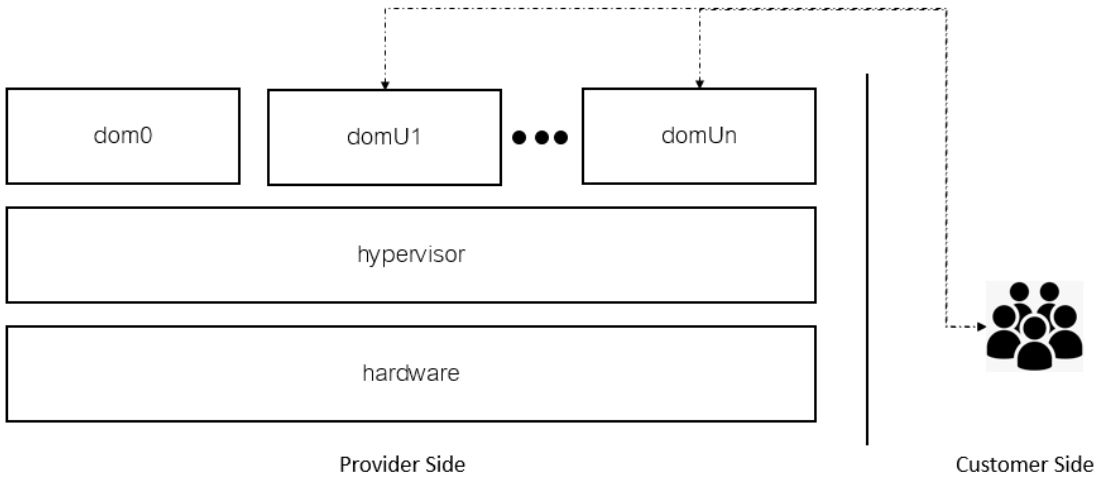
3. ความเสถียร คลาวด์ IaaS สามารถทำงานได้ตลอดเวลาเมื่อต้องการและไม่มีการหยุดการทำงานจากปัญหาและอุปสรรคในการทำงาน เช่นเครื่องคอมพิวเตอร์หยุดการทำงานเนื่องจากเกิดชำรุดเสียหายของอุปกรณ์ ไฟฟ้าดับ อินเทอร์เน็ตมีปัญหา ซึ่งผู้ให้บริการอย่าง Amazon ได้กำหนดเป้าหมาย ข้อตกลงระดับการให้บริการ Amazon EC2 คือ ความพร้อมใช้งานที่ 99.99% สำหรับ Amazon EC2

4. สำรองข้อมูลและการกู้คืน ผู้ให้บริการ IaaS ให้ผู้ใช้บริการเข้าถึงโครงสร้างพื้นฐานที่ไม่จำกัดสำหรับการสำรองข้อมูลและกระบวนการกู้คืนจากความเสียหาย ตัวอย่างเช่น ผู้ใช้สามารถทำแอปพลิเคชันซ้ำในหลายเซิร์ฟเวอร์ ดังนั้นหากตัวใดตัวหนึ่งล้มเหลว อีกตัวหนึ่งจะเข้ามาแทนที่ ในทำนองเดียวกัน สามารถซิงค์ข้อมูลสำรองได้อัตโนมัติและบ่อยครั้ง เพื่อให้เกิดส่วนซ้ำสำรองและความต่อเนื่องทางธุรกิจ

5. ราคาค่าใช้บริการ คลาวด์ IaaS คือ โมเดลการประมวลผลบนคลาวด์ที่ผู้ใช้บริการจ่ายเฉพาะทรัพยากรที่ใช้เท่านั้น การตั้งค่าดังกล่าวส่งเสริมการจัดการทรัพยากรไอทีที่มีประสิทธิภาพมากขึ้นและส่งเสริมนวัตกรรมโดยการทำให้บริการคลาวด์ราคาไม่แพงไปสำหรับธุรกิจขนาดเล็ก

คลาวด์ IaaS เป็นบริการที่เน้นการทำงานสำหรับการประมวลผล ซึ่งมีนักวิจัยหลายกลุ่มได้นำไปใช้อย่างกว้างขวางโดยองค์กร ภาครัฐ ภาคการศึกษา หรือแม้กระทั่งใช้ในการประมวลผลในการทดลองทางการแพทย์ ที่สามารถใช้เวลาในการประมวลผลอันสั้นและยังลดค่าใช้จ่ายในการทำงานของนักวิจัยได้อีกด้วย เมื่อใช้งานเสร็จก็จะทำการยกเลิกการใช้งานทันทีสรุปความหมาย

ของคลาวด์ IaaS จากงานวิจัยของWongthai & Moorsel (2016) ได้ให้คำอธิบายว่าการให้บริการใช้คอมพิวเตอร์เสมือน (Virtual Machine) แก่ผู้ใช้บริการในการประมวลผล ดังสถาปัตยกรรมภาพ 2 โดยในการทดลองของวิทยานิพนธ์เล่มนี้ใช้คลาวด์ IaaS ในการทดลอง จึงต้องมีการอธิบายเพิ่มเติมเกี่ยวกับสถาปัตยกรรมของคลาวด์ IaaS โดยกล่าวเพิ่มเติมในย่อหน้าถัดไปคือสถาปัตยกรรมของคลาวด์ IaaS



ภาพ 2 สถาปัตยกรรมของคลาวด์ IaaS

สถาปัตยกรรมของคลาวด์ IaaS ได้แบ่งสภาพแวดล้อมทำงานออกเป็น 2 ส่วน ได้แก่ ส่วนผู้ให้บริการ (Provider side) และส่วนผู้ใช้บริการ (Customer side) ดังแสดงในภาพ 2 ซึ่งในส่วนของผู้ให้บริการ ซึ่งหมายถึงองค์กรที่เตรียมบริการคอมพิวเตอร์เสมือนให้กับผู้ใช้บริการ มีองค์ประกอบที่จำเป็นจะต้องศึกษาเพื่อใช้เป็นข้อมูลในการทดลองของวิทยานิพนธ์เล่มนี้ ดังนี้

1. hardware คือเครื่องคอมพิวเตอร์ที่ประกอบไปด้วยหน่วยประมวลผลกลาง หน่วยความจำ หน่วยสำรองข้อมูลและอุปกรณ์ ที่ใช้ในการสร้าง คลาวด์ IaaS

2. hypervisor คือซอฟท์แวร์ที่สามารถทำให้เครื่องคอมพิวเตอร์หนึ่งเครื่องรันระบบปฏิบัติการได้มากกว่าหนึ่งระบบในเวลาเดียวกันซึ่งถูกกล่าวถึงในงานวิจัยของ Thyagaturu, Shantharama, Nasrallah & Reisslein (2022) ซอฟท์แวร์ hypervisor นี้มีทั้งแบบการค้า (Commercial) เช่น VMware หรือ ใช้งานฟรี (Open source) เช่น xen ซอฟท์แวร์ hypervisor จะถูกติดตั้งบน hardware ความรับผิดชอบของ hypervisor สามารถจัดการหน่วยความจำ และหน่วยประมวลผลของเครื่องคอมพิวเตอร์เสมือนได้ทั้งหมด

3. dom0 เป็นเครื่องคอมพิวเตอร์เสมือน เรียกว่า ดอมซีโร่ ย่อมาจาก Domain Zero ซึ่งเป็นคำศัพท์เฉพาะที่ใช้กับ Xen ที่ทำหน้าที่บริหารจัดการ domU และเป็นเครื่องคอมพิวเตอร์เสมือนเพียงเครื่องเดียวที่มีการเข้าถึง hardware ได้โดยตรง ซึ่ง hypervisor ไม่สามารถใช้งานได้หากไม่มี dom0 หรืออาจจะกล่าวอีกด้านหนึ่งว่า dom0 จะทำงานเมื่อ hypervisor ถูกเปิดใช้งาน ถูกกล่าวไว้ในงานวิจัยของ Tiwari et al. (2022)

4. domU เป็นเครื่องคอมพิวเตอร์เสมือนที่ถูกสร้างขึ้นและครอบครองใช้งานโดยผู้ใช้บริการ เรียกว่า ดอมยู ย่อมาจาก Unprivileged Domains โดยแต่ละตัวจะทำงานเป็นอิสระ สามารถมีได้หลายตัวภายใต้การควบคุมดูแลจัดการของ dom0 อาจกล่าวที่ได้ว่าผู้ใช้บริการที่ขอใช้บริการ คลาวด์ IaaS เป็นเจ้าของ domU เสมือนมีคอมพิวเตอร์หรือเซิร์ฟเวอร์ไว้ใช้งานส่วนตัว และสามารถเป็นเจ้าของ domU ได้มากกว่าหนึ่งตัว ถูกกล่าวไว้ในงานวิจัยของ Dordevic, Timcenko, Sakic & Davidovic (2022)

จากหัวข้อเรื่อง คลาวด์ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ มีเนื้อหาเกี่ยวกับลักษณะการให้บริการ ความสำคัญ ประโยชน์และ สถาปัตยกรรมของคลาวด์ IaaS ที่แสดงถึงข้อมูลเบื้องต้น การนำไปใช้งาน หลักการทำงานและองค์ประกอบในการทำงานของคลาวด์ IaaS ซึ่งพบว่ามีแนวโน้มในการที่จะถูกใช้องค์กร หน่วยงาน หรือบุคคล นำไปใช้งานมากขึ้นทดแทนระบบเดิม ในขณะเดียวกันเมื่อได้รับความนิยมในการใช้งานเพิ่มมากขึ้น ก็จะต้องมีผู้ที่ต้องการแสวงหาผลประโยชน์ ผู้ไม่ประสงค์ดี เข้ามาสร้างความเสียหาย ก่อกวน หรือกระทำการใดที่ส่งผลกระทบต่อการใช้งาน ทั้งหมดที่กล่ามานั้นถือว่าเป็นภัยคุกคามต่อการใช้งานของผู้ใช้บริการและการให้บริการของผู้ให้บริการ ดังนั้นเพื่อจึงต้องทำการศึกษาเพื่อเข้าใจถึงปัญหาภัยคุกคามที่มีผลต่อคลาวด์ว่ามีอะไรบ้าง มีวิธีการอย่างไร และแนวทางป้องกันหรือวิธีการบรรเทาปัญหา ซึ่งจะกล่าวในหัวข้อ ถัดไป

ปัญหาภัยคุกคามของคลาวด์

การใช้งานคลาวด์มีประโยชน์มากมาย ปริมาณงานหรือข้อมูลที่มีอยู่นั้นยากที่จะจัดเก็บและสามารถประมวลผลได้ภายในองค์กร จึงเป็นเหตุผลที่ทำให้ระบบคลาวด์มีความนิยมมากขึ้นด้วยบริการต่างๆบนคลาวด์ ทำให้แม้แต่ในธุรกิจขนาดเล็กหรือบุคคลธรรมดาทั่วไปก็สามารถเข้าถึงและใช้งานได้ แม้ว่าคลาวด์จะมีประโยชน์เป็นอย่างมาก แต่ก็ได้รับการจับตามองด้านความปลอดภัยจากทั้งผู้ประสงค์ดีและไม่ประสงค์ดี เนื่องจากการที่คลาวด์มีอัตราการเติบโตอย่างรวดเร็ว การทำงานและข้อมูลที่อยู่บนคลาวด์นั้นส่วนมากเป็นข้อมูลที่สำคัญและมีคุณค่าสูง หากเกิดสูญหายหรือถูกโจรกรรมอาจจะส่งผลกระทบต่ออย่างร้ายแรง คลาวด์นั้นมีความเสี่ยงที่จะถูก

โจมตีสูง เพราะจากข้อมูลที่ผู้ใช้บริการได้เก็บไว้กลายเป็นเป้าหมายของการโจมตีจากผู้ที่ไม่ประสงค์ดี ซึ่งผู้ใช้บริการเองก็ไม่สามารถทราบได้ว่าเบื้องหลังการทำงานของคลาวด์นั้นทำงานอย่างไร และมีช่องโหว่อะไรบ้างที่ทำให้ง่ายต่อการโจมตีข้อมูล ดังนั้นในหัวข้อนี้เราจะกล่าวถึงภัยคุกคามที่เกิดขึ้น ปัจจุบันมีองค์กรที่ชื่อว่า Cloud Security Alliance หรือ CSA ซึ่งเป็นองค์กรชั้นนำของโลกที่อุทิศตนเพื่อกำหนดมาตรฐาน การรับรอง และแนวปฏิบัติที่ดีที่สุดเพื่อช่วยให้มั่นใจถึงสภาพแวดล้อมการประมวลผลบนคลาวด์ที่ปลอดภัย ได้ทำรายงานเกี่ยวกับลำดับภัยคุกคามที่เกิดขึ้นกับคลาวด์ไว้ทุกๆ 3 ปี โดยในปี ค.ศ. 2019 ได้จัดลำดับภัยคุกคามของคลาวด์ชื่อว่า Top Threats to Cloud Computing: The Egregious Eleven (CSA, 2019) โดยมีทั้งหมดภัยคุกคามทั้งหมด 11 ประเภท ดังนี้

1. การละเมิดข้อมูล (Data Breaches) คือ การละเมิดข้อมูลเป็นเหตุการณ์ที่บุคคลที่ไม่ได้รับอนุญาตขโมยข้อมูลที่เป็นความลับ ขอบเขตของความเสียหายที่เกิดขึ้นมักจะขึ้นอยู่กับลักษณะของข้อมูลที่เปิดเผย การละเมิดข้อมูลมีแนวโน้มที่จะเกิดขึ้นสำหรับธุรกิจที่ใช้ระบบคลาวด์มากกว่าธุรกิจที่ไม่ได้ใช้งานถึงสามเท่า ข้อมูลจำนวนมหาศาลบนคลาวด์และความสะดวกในการเข้าถึงทำให้เป็นเป้าหมายที่น่าสนใจ

2. การกำหนดค่าผิดพลาดและการควบคุมการเปลี่ยนแปลงไม่ดีเพียงพอ (Misconfiguration and inadequate change control) คือ การที่ผู้ใช้กำหนดหรือปรับเปลี่ยนการใช้ทรัพยากรไม่ถูกต้องโดยที่ระบบไม่มีการป้องกันความผิดพลาด ทำให้ทรัพยากรของผู้ใช้บริการที่เป็นเจ้าของหรือคนอื่นถูกลบหรือแก้ไข ส่งผลให้เกิดความผิดพลาด สูญเสียข้อมูลและการทำงานหยุดชะงัก

3. การขาดสถาปัตยกรรมและกลยุทธ์การรักษาความปลอดภัยบนคลาวด์ (Lack of cloud security architecture and strategy) คือหากผู้ใช้บริการไม่มีการวางแผนการรักษาความปลอดภัยที่ดีและเหมาะสม จะเสี่ยงต่อการโจมตีทางไซเบอร์ที่อาจส่งผลให้เกิดความสูญเสียความเสียหาย ปัญหาทางกฎหมายและผิดกฎข้อบังคับ

4. การระบุตัวตน ข้อมูลประจำตัว การเข้าถึงและการจัดการคีย์ ไม่ดีเพียงพอ (Insufficient identity, credential, access and key management) คือการถูกโจมตี จากขาดระบบการจัดการการเข้าถึงข้อมูลประจำตัว ความล้มเหลวในการใช้การตรวจสอบสิทธิ์แบบหลายปัจจัย การใช้รหัสผ่านที่ไม่รัดกุม และการจัดการคีย์และใบรับรองที่ไม่ดี CSA กล่าวว่าผู้ประสงค์ร้ายที่ปลอมแปลงเป็นผู้ใช้ ผู้ปฏิบัติงาน หรือนักพัฒนาที่ถูกต้องตามกฎหมาย สามารถอ่าน แก้ไขและลบข้อมูลได้

5. บัญชีผู้ใช้งานถูกขโมย (Account hijacking) คือ ข้อผิดพลาดของผู้ใช้หรือการโจมตีที่เป็นอันตรายอาจทำให้บัญชีถูกขโมย เมื่อเข้าไปข้างในแล้ว แฮกเกอร์สามารถดักฟังกิจกรรม แก้ไขข้อมูล หรือจัดการธุรกรรมได้ ผู้บุกรุกอาจเปลี่ยนเส้นทางผู้ใช้บริการไปยังเนื้อหาที่ไม่เหมาะสมหรือเว็บไซต์ของคู่แข่งได้ กลวิธีที่เป็นอันตรายทั้งหมดเหล่านี้สร้างความเสียหายให้กับธุรกิจ หากผู้ใช้ที่ไม่ประสงค์ดีเข้าถึงบัญชีคลาวด์ของคุณได้ พวกเขาสามารถเปิดการโจมตีเพิ่มเติมจากภายในบริการได้ ซึ่งอาจขัดต่อบริษัทหรือผู้ใช้รายอื่น

6. ภัยคุกคามที่เกิดจากบุคคลภายใน (Insider threat) คือบุคคลภายในที่เป็นอันตราย เช่น ผู้ดูแลระบบสามารถเข้าถึงข้อมูลที่ละเอียดอ่อนได้ พวกเขาสามารถมีระดับการเข้าถึงที่เพิ่มขึ้นไปยังระบบที่สำคัญยิ่งขึ้นและในที่สุดก็ถึงข้อมูล นอกจากนี้ยังรวมถึงข้อผิดพลาดของมนุษย์ที่เกิดจากผู้ดูแลระบบ ภัยคุกคามนี้สามารถลดลงได้ด้วยการใช้นโยบายที่เหมาะสม การแบ่งแยกหน้าที่ ลดการเข้าถึงตามบทบาท และการบันทึกที่มีประสิทธิภาพ การติดตามและตรวจสอบกิจกรรมของผู้ดูแลระบบ

7. อินเทอร์เฟซและส่วนต่อประสานโปรแกรมประยุกต์ที่ไม่ปลอดภัย (Insecure interfaces and APIs) ความปลอดภัยและความพร้อมใช้งานของบริการคลาวด์ทั่วไปขึ้นอยู่กับ API และอินเทอร์เฟซที่ใช้โดยผู้ให้บริการคลาวด์ ทั้งสองส่วนนี้เป็นส่วนที่เปิดเผยมากที่สุดของระบบเพราะเมื่อมีการเข้าใช้สามารถเห็นขั้นตอนการทำงานได้ตลอด ซึ่งเสี่ยงต่อภัยคุกคามความปลอดภัยที่หลากหลาย ต้องได้รับการออกแบบและสามารถป้องกันความพยายามในการหลบเลี่ยงหรือโจมตีระบบทั้งโดยไม่ได้ตั้งใจและโดยประสงค์ร้าย

8. ระดับชั้นการควบคุมที่อ่อนแอ (Weak control plane) คือการดำเนินการในการจัดการ ควบคุม ดูแล ปัญหานั้นต่ำมาก เช่นเมื่อบริการคลาวด์ไม่ให้ความปลอดภัยที่เพียงพอที่ไปตามข้อกำหนดด้านความปลอดภัยของลูกค้า ตัวอย่างหนึ่งคือไม่มีการยืนยันตัวตน 2 ชั้นที่สมควรนำมาบังคับใช้งาน และไม่มีการแจ้งปัญหาหรือจัดการปัญหาที่เกิดขึ้นก่อนให้ผู้ใช้เข้าใช้บริการ

9. ความล้มเหลวของผู้บริหารและการจัดการ (Metastructure and applistructure failures) คือ ความล้มเหลวในการดำเนินการด้านความปลอดภัยที่เกิดขึ้นกับผู้บริหารในองค์กรที่ให้บริการระบบคลาวด์ ไม่เข้าใจถึงสำคัญในการพัฒนาระบบความปลอดภัยให้กับผู้ใช้บริการคลาวด์

10. การมีข้อจำกัดในมุมมองของการใช้งานคลาวด์ (Limited cloud usage visibility) คือการขาดความรู้เกี่ยวกับการใช้คลาวด์ทำให้ไม่สามารถจำแนกได้ว่าการใช้บริการใดของคลาวด์นั้นมีความปลอดภัยในการใช้งานภายในองค์กรปลอดภัยหรือเป็นอันตรายต่อการใช้งาน

11. การใช้บริการคลาวด์ในทางที่ผิดและในทางที่ผิด (Abuse and nefarious use of cloud services) คือการที่ผู้ใช้บริการใช้บริการคลาวด์ไปแสวงหาผลประโยชน์และสร้างความเดือนร้อนให้ผู้อื่น เนื่องจากเป็นบุคคลใดก็ได้สามารถสมัครใช้บริการคลาวด์ได้อย่างง่าย และมีค่าใช้จ่ายน้อย

โดยในวิทยานิพนธ์เล่มนี้ มีความสนใจที่จะบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ในลำดับ 1 คือ การละเมิดข้อมูล ซึ่งเป็นภัยคุกคามที่สำคัญและมีแนวโน้มเกี่ยวกับการการกระผิดประเภทนี้ที่สูงขึ้น เนื่องจากข้อมูลที่มีความสำคัญในคลาวด์มีความเสี่ยงในการถูกโจรกรรม นำไปใช้ หรือมีบุคคลที่ไม่ได้รับอนุญาตเข้าถึงข้อมูลได้ โดยสาเหตุมาจากช่องโหว่ที่เกิดจากข้อผิดพลาดของผู้ให้บริการหรือผู้ใช้บริการรวมถึงช่องโหว่จากการรักษาความปลอดภัยที่ไม่ดี จากรายงานของ CSA การละเมิดข้อมูล จะมีด้วยกัน 4 รูปแบบ คือ 1) การดูข้อมูล (viewed) 2) การเปิดเผยข้อมูล (release) 3) การขโมยข้อมูล (Stolen) 4) การใช้งานโดยไม่ได้รับอนุญาต (Used) ตัวอย่างการละเมิดข้อมูล ในเดือนเมษายนปี 2021 Facebook รายงานว่ามีการละเมิดข้อมูลซึ่งส่งผลกระทบต่อบันทึกผู้ใช้หลายร้อยล้านเรคคอร์ด แม้ว่า Facebook จะยืนยันว่าตรวจพบและแก้ไขปัญหาในทันที ซึ่งสาเหตุเกิดจาก บริษัทผู้พัฒนาแอปพลิเคชันให้กับ Facebook สองบริษัทได้โพสข้อมูลที่มีข้อมูลส่วนตัว ซึ่งทำให้ผู้ไม่ประสงค์ดี ใช้ข้อมูลส่วนตัวที่ได้จากการโพสของบริษัทผู้พัฒนาแอปพลิเคชันเข้าไปในคลาวด์และนำข้อมูลออกมาเผยแพร่ จะเห็นได้ว่ามีความสอดคล้องกับรูปแบบทั้ง 4 รูปแบบของ CSA จากเหตุการณ์นี้สามารถเปรียบเทียบได้ดังนี้ บริษัทผู้พัฒนาแอปพลิเคชันได้นำข้อมูลออกมาเผยแพร่แต่ข้อมูลที่เผยแพร่ออกไปนั้นมีข้อมูลส่วนตัวที่ผู้ไม่ประสงค์ดีนำไปใช้เพื่อเข้าถึงข้อมูลในคลาวด์ในเหตุการณ์ช่วงนี้เป็นการกระทำที่ผิดพลาดของบริษัทผู้พัฒนาแอปพลิเคชันหรือผู้ใช้บริการ ส่งผลให้ผู้ไม่ประสงค์ดี ใช้ข้อมูลที่ได้มาเข้าไปใช้งานโดยไม่ได้รับอนุญาต เมื่อเข้าถึงข้อมูลผู้ไม่ประสงค์ดี ก็เข้าดูข้อมูล และคัดลอกข้อมูลที่ต้องการออกมาซึ่งการทำแบบนี้เป็นการขโมยข้อมูล และสุดท้ายนำข้อมูลที่ได้มาไปเผยแพร่ ก่อให้เกิดความเสียหาย

จากปัญหาด้านความปลอดภัยของคลาวด์เกี่ยวกับภัยคุกคามประเภทการละเมิดข้อมูลที่องค์กร CSA ได้รายงานมีผลกระทบอย่างมากต่อการให้บริการคลาวด์ IaaS ในวิทยานิพนธ์เล่มนี้ผู้วิจัยสนใจที่จะทำการศึกษาเกี่ยวกับการบรรเทาปัญหาหรือป้องกันการละเมิดข้อมูลในคลาวด์จากการศึกษาการรักษาความปลอดภัยของการละเมิดข้อมูล เกี่ยวกับวิธีการ หลักการทำงานและแนวทางในการรักษาความปลอดภัย โดยจะอธิบายเพิ่มเติมในหัวข้อ วิธีการรักษาความปลอดภัยของการละเมิดข้อมูล ลำดับถัดไป

วิธีการรักษาความปลอดภัยของการละเมิดข้อมูล

ในปัจจุบันได้มีเครื่องมือและวิธีการหลากหลายที่ช่วยสนับสนุนในการรักษาความปลอดภัยเกี่ยวกับการละเมิดข้อมูล โดยมีนักวิจัยของได้แบ่งวิธีการรักษาความปลอดภัยไว้เป็น 2 วิธีการด้วยกัน ได้ถูกกล่าวไว้ในงานของ Lal, Prasad, Kumar & Kumar (2022) โดยมีรายละเอียดดังนี้

1. การป้องกัน (Preventive) วิธีการนี้จะเป็นกันรักษาความปลอดภัยโดยการควบคุมการเข้าถึงการทำงาน เช่นการกำหนดสิทธิเข้าถึงข้อมูล การใช้ยืนยันตัวตน และกำหนดกลไกการทำงานโดยออกเป็นนโยบาย แนวทางปฏิบัติ หรือการใช้เครื่องมือหรือวิธีการเช่นระบบไฟร์วอลล์ (Firewall) การเข้ารหัสข้อมูล (encryption) โดยวิธีการนี้จะป้องกันไม่ให้เกิดเหตุการณ์ที่ส่งผลกระทบต่อการละเมิดข้อมูล ซึ่งถูกกล่าวไว้ในงานของ Lee, de Guzman, Wang, Gupta & Rao (2022) แต่ไม่มีระบบป้องกันใดที่สามารถทำงานได้สมบูรณ์แบบดังนั้นจึงจะต้องมีการจัดการเพื่อช่วยลดโอกาสหรือบรรเทาปัญหาเช่นการเก็บหลักฐานเพื่อพิจารณาและค้นความจริง

2. การตรวจสอบ (Detective) คือตรวจสอบหาร่องรอยของการกระทำ (Audit trails) จากนั้นจึงระบุพฤติกรรมที่น่าสงสัยหรือโค้ดที่เป็นอันตรายจากข้อมูลที่รวบรวมโดยใช้วิธีการและอัลกอริธึมที่หลากหลาย และนำมาวิเคราะห์ (Analysis) โดยมนุษย์หรือคอมพิวเตอร์ซึ่งถูกกล่าวไว้ในงานของ Achar (2022) เพื่อบุคคลที่รับผิดชอบต่อสิ่งที่เกิดขึ้น (Accountability)

จากข้อมูลเกี่ยวกับการป้องกันและการตรวจสอบจะเห็นได้ว่าไม่มีวิธีการป้องกันที่สมบูรณ์เนื่องจากอาจเกิดความผิดพลาดของมนุษย์หรือจุดอ่อนของระบบ ดังนั้นการตรวจสอบเพื่อหาร่องรอยของการกระทำนั้นเป็นวิธีที่สามารถเพิ่มประสิทธิภาพของการป้องกันได้ดียิ่งขึ้น แต่ในขั้นตอนของการตรวจสอบนั้นจำเป็นต้องมีหลักฐานหรือข้อมูลเพื่อนำมาใช้ในการพิจารณาค้นหา พิสูจน์ความจริงเพื่อหาบุคคลที่รับผิดชอบต่อสิ่งที่เกิดขึ้น ในระบบคอมพิวเตอร์จะมีการเก็บข้อมูลของการกระทำต่างๆเป็นไฟล์ข้อมูลเรียกว่า ล็อกไฟล์ (log files) รายละเอียดของ ล็อกไฟล์จะถูกอธิบายเพิ่มเติมในหัวข้อถัดไป

ล็อกไฟล์ (logs files)

การตรวจสอบ พิจารณา ค้นหา พิสูจน์ความจริงเพื่อหาความรับผิดชอบต่อสิ่งที่เกิดขึ้นจะเห็นว่าหลักฐานคือสิ่งสำคัญที่จะนำไปสู่กระบวนการหาตัวบุคคลกระทำผิดมารับผิดชอบต่อสิ่งที่เกิดขึ้น โดยหลักฐานดังกล่าวทำให้ทราบว่าบุคคลใดที่มีส่วนเกี่ยวข้องในการกระทำผิดบ้างและต้องรับผิดชอบต่อสิ่งที่เกิดขึ้นอย่างไร โดยล็อกไฟล์จะบันทึกเหตุการณ์หรือการกระทำต่าง ๆ ที่เกิดขึ้น การเก็บหลักฐานเกี่ยวกับเหตุการณ์หรือการกระทำที่เกิดขึ้นที่อยู่ในล็อกไฟล์จะเป็น

ประโยชน์ให้ทั้งผู้รับการบริการและผู้ให้บริการซึ่งส่งผลต่อความมั่นใจต่อการใช้บริการคลาวด์ ถูกอธิบายไว้ในงานของ Ko (2014) ล็อกไฟล์ของระบบ ถูกแบ่งออกเป็น 2 ประเภท ดังนี้

1. File centric log หรือ File log ล็อกไฟล์ประเภทนี้เป็นไฟล์ข้อมูลที่มีวัตถุประสงค์เพื่อเก็บประวัติการทำงานต่างๆที่เกิดขึ้น เช่น ล็อกไฟล์ของการเข้าถึงข้อมูล จะเก็บเหตุการณ์ที่เกิดขึ้นเมื่อมีการกระทำใดๆกับข้อมูล เช่น การอ่าน การแก้ไข การเปลี่ยนแปลง และการลบข้อมูล ได้ถูกกล่าวไว้ในงานของ (Popa, 2019)

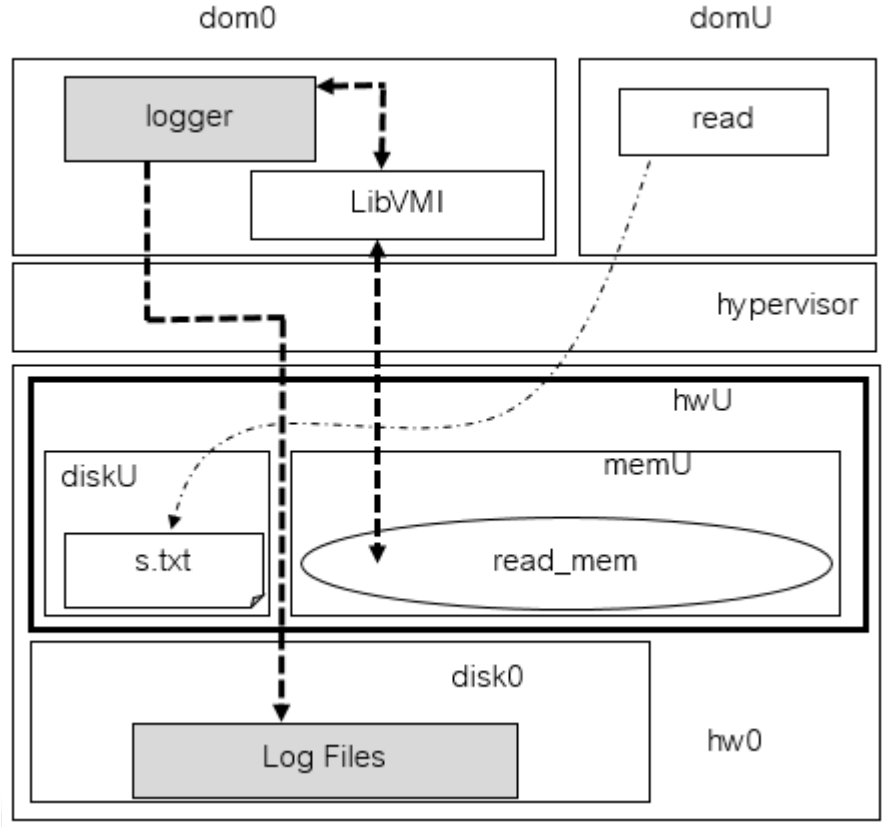
2. System centric log หรือ Syslog ในงานของ Parra et al. (2022) ได้ให้ความหมายล็อกไฟล์ประเภทนี้ว่าเป็นไฟล์ข้อมูลที่เก็บข้อมูลการทำงานของซอฟท์แวร์ ฮาร์ดแวร์ กระบวนการและส่วนประกอบของระบบ เช่น อุณหภูมิที่เกิดขึ้น แรงดันฟ้าที่ใช้ การทำงานหน่วยต่างๆ และสถานภาพของส่วนประกอบของระบบ

จากรายละเอียดเกี่ยวกับล็อกไฟล์ที่ถูกแบ่งออกเป็น 2 ประเภท โดยในวิทยานิพนธ์เล่มนี้จะมุ่งเน้นเกี่ยวกับ file-centric log หรือ file-log ซึ่ง ล็อกไฟล์ที่นำมาใช้ในการทดลองนี้ได้มาจากระบบบันทึกเหตุการณ์ ซึ่งจะอธิบายการทำงานในหัวข้อถัดไป ในวิทยานิพนธ์นี้บางกรณีนั้นจะเรียกล็อกไฟล์ว่าไฟล์บันทึกเหตุการณ์เพื่อความเข้าใจที่ง่าย

ระบบบันทึกเหตุการณ์ (Logging system)

ระบบบันทึกเหตุการณ์เป็นระบบที่ทำหน้าที่บันทึกเหตุการณ์หรือการกระทำต่าง ๆ ที่เกิดขึ้น และเก็บไว้ใน ล็อกไฟล์ เพื่อนำไปสู่หลักฐานสำหรับตรวจสอบร่องรอยของการกระทำ เช่น งานของ Adjepon-Yamoah & Ebo (2022); Melvin et al. (2022); Nguyen, Orenbach & Atamli (2022) เป็นต้น โดยในหัวข้อนี้ผู้วิจัยจะอธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ที่ใช้ในการทดลอง และการทำงานของระบบบันทึกเหตุการณ์

โดยหัวข้อนี้จะอธิบายเกี่ยวกับสถาปัตยกรรมของระบบบันทึกเหตุการณ์ ที่สามารถใช้ในคลาวด์ IaaS ซึ่งสถาปัตยกรรมดังกล่าวเป็นการอ้างอิงจากงานวิจัย (Wongthai & Moorsel, 2016) โดยทางผู้วิจัยได้มีการแก้ไขภาพของสถาปัตยกรรมดังกล่าวเพื่อให้สอดคล้องกับงานทดลองในวิทยานิพนธ์เล่มนี้ โดยผู้วิจัยจะนำเสนอในส่วนที่ถูกนำไปใช้งานจริง แสดงในภาพ 3 ภายในสถาปัตยกรรมจะแบ่งการทำงานออกเป็น 2 ส่วน คือ 1) ส่วนของผู้ให้บริการ ซึ่งจะเป็นเครื่องคอมพิวเตอร์เสมือนที่มีโปรเซสซึ่งทำหน้าที่ตรวจสอบเหตุการณ์หรือการกระทำที่เกิดขึ้นกับไฟล์ข้อมูลชื่อว่า ล็อกเกอร์ (Logger) และบันทึกเหตุการณ์ลงใน ล็อกไฟล์ 2) ส่วนของผู้ใช้บริการหรือลูกค้าซึ่งเป็นเครื่องคอมพิวเตอร์เสมือนที่ถูกตรวจสอบ



ภาพ 3 สถาปัตยกรรมของระบบบันทึกเหตุการณ์

1. **ส่วนของผู้ให้บริการ** (dom0 : Domain zero)

จากภาพ 3 dom0 เป็นเครื่องคอมพิวเตอร์เสมือนของผู้ให้บริการที่ถูกสร้างขึ้นมาเพื่อใช้ในการจัดการ domU ซึ่งมีการทำงานร่วมกับ hypervisor สามารถดูและทบทวนการทำงานได้ที่หัวข้อเรื่อง คลาวด์ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ dom0 เป็นเจ้าของทรัพยากรทั้งหมดสามารถสร้างส่วนจำลองให้กับ domU เช่น หน่วยประมวลผล หน่วยความจำหลัก หน่วยความจำสำรองเป็นต้น ในการทำงานของระบบบันทึกเหตุการณ์นั้นจะโปรแกรมระบบบันทึกเหตุการณ์ (Logging system) จะถูกติดตั้งไว้ใน dom0 โดยกระบวนการทำงานของของระบบบันทึกเหตุการณ์ มี 3 ขั้นตอน ดังนี้

1. ขั้นตอนที่ 1. ผู้ให้บริการเปิดการทำงานของระบบบันทึกเหตุการณ์ จะเกิดโปรเซสล็อกเกอร์ (logger) ที่มีความสามารถอ่านข้อมูลในหน่วยความจำเสมือนของเครื่องผู้ใช้บริการ (memU)

2. ขั้นตอนที่ 2. ล็อกเกอร์ จะอ่านข้อมูลของโปรเซส (read_mem) ที่เกิดขึ้นใน memU แล้วจะทำการตรวจสอบตามเงื่อนไขที่ถูกโปรแกรมไว้ หากเงื่อนไขไม่ถูกต้องจะตรวจสอบ

วนซ้ำไปเรื่อยๆ หรือ หากเงื่อนไขถูกต้องจะดำเนินการในขั้นตอนที่ 3. เมื่อเสร็จสิ้นขั้นตอนที่ 3. จะทำขั้นตอนที่ 2. ซ้ำไปจนกว่าจะหยุดการทำงาน

3. ขั้นตอนที่ 3. หาก ล็อกเกอร์ ตรวจสอบเงื่อนไขแล้วพบว่าตรงตามเงื่อนไขที่โปรแกรมไว้จะนำข้อมูลที่อยู่ใน memU ซึ่งประกอบไปด้วย หมายเลขของผู้ใช้งาน หมายเลขของโปรเซส ชื่อโปรเซส เวลาที่โปรเซสทำงาน และชื่อไฟล์ที่ถูกโปรเซสเรียกใช้งาน เก็บบันทึกลงไปในล็อกไฟล์

จากการทำงานของระบบบันทึกเหตุการณ์นี้เมื่อเริ่มทำงานจะมีการสร้างโปรเซสชื่อว่าล็อกเกอร์ ขึ้นมาเพื่อเข้าไปทำการอ่านข้อมูลในหน่วยความจำเสมือนของเครื่องผู้ใช้บริการ ในขั้นตอนการเข้าถึงหน่วยความจำเสมือนจะใช้ Library ภาษาซี ที่ชื่อว่า libVMI ซึ่งจะอธิบายการทำงานในหัวข้อเรื่อง libVMI สำหรับการตรวจสอบเหตุการณ์และการกระทำต่าง ๆ ที่มีต่อไฟล์ข้อมูล หากพบว่ามีเหตุการณ์หรือการกระทำใดๆ ที่ตรงตามเงื่อนไขจะบันทึกข้อมูลเหตุการณ์ลงใน ล็อกไฟล์ ตามขั้นตอนที่อธิบายไว้ข้างต้น

2. **ส่วนของผู้ใช้บริการหรือลูกค้า** (domU : Unprivileged Domains)

จากภาพ 3 domU เป็นเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการโดยมีการจำลองหน่วยความจำสำรอง (Hard disk) เรียกว่า diskU ภายใน disk0 ซึ่ง disk0 เป็นหน่วยความจำสำรองทั้งหมดที่มีอยู่จริงที่มีไว้เพื่อให้บริการกับ domU ทุกตัวในคลาวด์ IaaS และมีการจำลองหน่วยความจำหลัก (Main memory) เรียกว่า memU ภายใน mem0 ซึ่ง mem0 เป็นหน่วยความจำหลักทั้งหมดที่มีอยู่จริงที่มีไว้เพื่อให้บริการกับ domU ทุกตัวในคลาวด์ IaaS ในการสร้าง domU ในการทดลองการทำงานของระบบบันทึกเหตุการณ์นี้ ผู้วิจัยได้สร้างไฟล์ s.txt ไว้ใน diskU จากนั้นได้จำลองกระบวนการทำงานภายใน domU โดยผู้ใช้บริการการสร้างโปรแกรม read เพื่อทำการอ่านไฟล์ s.txt เมื่อรันโปรแกรมจะทำการสร้างโปรเซส read ใน memU ทำหน้าที่อ่านไฟล์ s.txt

ต่อจากนี้จะอธิบายกระบวนการทำงานของระบบบันทึกเหตุการณ์ในส่วนผู้ให้บริการและการทดลองอ่านไฟล์ข้อมูลของผู้ใช้บริการ โดยเริ่มต้นจากการจำลองการทดลองอ่านไฟล์ข้อมูลในเครื่องคอมพิวเตอร์เสมือนของผู้ใช้งาน (domU) และ ต่อด้วยการทำงานของระบบบันทึกเหตุการณ์ในเครื่องคอมพิวเตอร์เสมือนของผู้ให้บริการ (dom0)

1. การจำลองการทดลองอ่านไฟล์ข้อมูลใน (domU) เมื่อผู้ใช้งานรันโปรแกรม read ในเครื่องคอมพิวเตอร์เสมือนของผู้ใช้งานจะทำการสร้างโปรเซส read เพื่อทำการอ่านไฟล์ s.txt ใน diskU กระบวนการนี้จะมีการสร้างรายละเอียดของโปรเซส และดึงข้อมูลจากโครงสร้างไฟล์ข้อมูล

(File Structure) ที่เกี่ยวข้องกับไฟล์ s.txt ไว้บน memU ซึ่งข้อมูลที่อยู่บน memU นั้นจะประกอบไปด้วยข้อมูลต่าง ๆ เช่น ชื่อไฟล์ หมายเลขของโปรเซส ชื่อโปรเซส ผู้ใช้งานโปรเซส

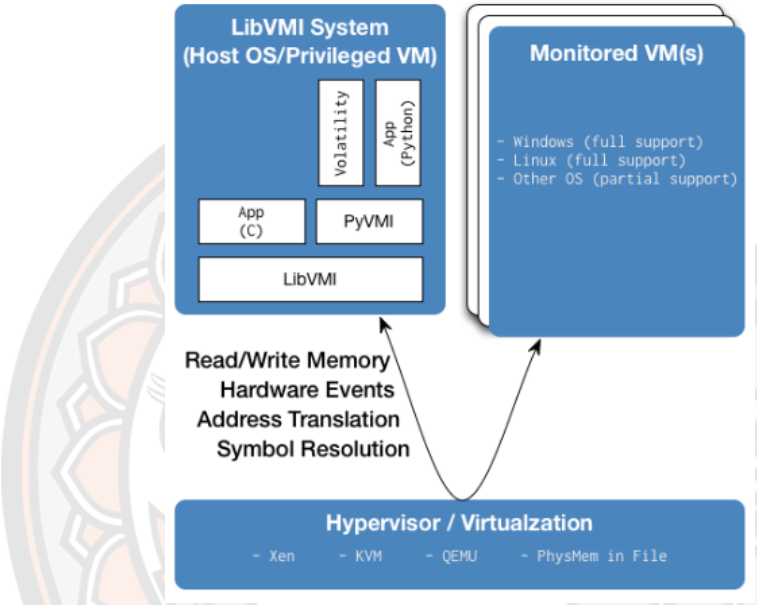
2. การทำงานของระบบบันทึกเหตุการณ์ใน (dom0) เมื่อมีการเปิดใช้งานระบบบันทึกเหตุการณ์ จะมีการสร้างโปรเซส ล็อกเกอร์ (logger) ใน dom0 ซึ่งทำหน้าที่อ่านข้อมูลของโปรเซส (read _mem) ใน memU โดยการตรวจสอบข้อมูลนี้จะพิจารณาชื่อไฟล์ในรายละเอียดของโปรเซส โดยทำงานวนซ้ำจนกว่าจะตรงตามเงื่อนไขที่กำหนด หากมีโปรเซสใด ๆ ที่มีชื่อไฟล์เป็น s.txt ล็อกเกอร์ จะทำการบันทึกข้อมูล ผู้ใช้งานโปรเซส ชื่อไฟล์ หมายเลขของโปรเซส ชื่อโปรเซส และระบุเวลาที่ตรวจสอบได้ ลงใน ล็อกไฟล์

ในหัวข้อเรื่อง ระบบบันทึกเหตุการณ์ นี้ได้อธิบายถึงองค์ประกอบ หลักการ หน้าที่ กระบวนการทำงานของระบบบันทึกเหตุการณ์ในเครื่องคอมพิวเตอร์เสมือนของผู้ให้บริการ และการสร้างสถานการณ์จำลองเหตุการณ์ ซึ่งระบบบันทึกเหตุการณ์จะถูกนำมาทดลองเพื่อทำการปรับปรุงเพื่อเพิ่มประสิทธิภาพในการทำงานของระบบ ในกระบวนการของระบบบันทึกเหตุการณ์มีส่วนที่เป็นหัวใจสำคัญในการทำงานคือการใช้เทคนิควิธีเพื่อทำการอ่านข้อมูลในหน่วยความจำหลักในเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการ เรียกว่า libVMI เป็นเป็นไลบรารี ในภาษาซี โดยจะอธิบายรายละเอียดในหัวข้อถัดไป

libVMI

LibVMI เป็นไลบรารีของภาษาซี ซึ่งใช้เทคนิควิธี introspection ในทางจิตวิทยาเรียกว่าวิธีการพินิจภายใน หมายถึงการที่จะทราบถึงจิตธาตุของบุคคลโดยวิธีการขอร้องให้รายงานการสัมผัสและความรู้สึก รวมทั้งความคิดของตนเองต่อสิ่งต่าง ๆ ดังคำอธิบายของ Spener (2022) การทำงานของ LibVMI ที่ถูกพัฒนาโดย Payne (2012) คือการเข้าถึงหน่วยความจำหลักของเครื่องคอมพิวเตอร์เสมือน ซึ่งใช้การหยุดการทำงานของเครื่องคอมพิวเตอร์เสมือนชั่วคราวเพื่อทำการถ่ายภาพ (Snapshot) โครงสร้างข้อมูลในหน่วยความจำหลักและยกเลิกการหยุด ช่วงเวลาในการหยุดนี้จะเป็นการหยุดในช่วงระยะเวลาสั้นๆ มีประโยชน์สำหรับการวิเคราะห์ และพิจารณาโครงสร้างข้อมูลในหน่วยความจำหลักที่ได้จากการถ่ายภาพในช่วงเวลาหยุดทำงานของเครื่องคอมพิวเตอร์เสมือนชั่วคราว ซึ่งมีงานวิจัยที่อธิบายความสามารถนำไปประยุกต์ใช้งานอย่างมากมาย เช่น Lata & Singh (2022); Mishra, Pilli, Varadharajan & Tupakula (2017) เป็นต้น ในการเข้าถึงหน่วยความจำ LibVMI รองรับการกระทำต่าง ๆ ในหน่วยความจำหลักของเครื่องคอมพิวเตอร์เสมือน โดยมีการแจ้งเตือนเมื่อหน่วยความจำของเครื่องคอมพิวเตอร์เสมือนมีการ executed write และ read ผู้พัฒนา LibVMI ได้ออกแบบให้ทำงานอยู่บนระบบปฏิบัติการลินุกซ์

(Linux) และ Mac OS X โดยใช้งานร่วมกับ Hypervisors ของ Xen หรือ KVM การทำงานร่วมกันของระบบปฏิบัติการและ Hypervisor ที่ทำให้ LibVMI มีประสิทธิภาพที่สุดจากการทำสอบใช้งานคือการใช้ระบบปฏิบัติการลินุกซ์ร่วมกับ Xen สำหรับการเข้าถึงหน่วยความจำของเครื่องคอมพิวเตอร์เสมือนนั้นสามารถเข้าถึงได้ทั้งระบบปฏิบัติการไมโครซอฟท์วินโดว์ (Microsoft Windows) และ ลินุกซ์ ดังภาพ 4



ภาพ 4 โครงสร้างการทำงานของ LibVMI

ที่มา: LibVMi (n.d.)

จากเนื้อหาข้างต้น LibVMI สามารถเข้าถึงหน่วยความจำหลักของเครื่องคอมพิวเตอร์เสมือน เพื่อดูโครงสร้างข้อมูลที่อยู่ในหน่วยความจำหลัก ซึ่งเป็นกระบวนการที่สำคัญในระบบบันทึกเหตุการณ์ และหัวข้อเรื่อง ระบบบันทึกเหตุการณ์ได้อธิบายข้อมูลที่ได้จากการตรวจสอบและนำไปบันทึกใน ล็อกไฟล์ จะพบว่ามีข้อมูล ผู้ใช้งานโปรเซส ชื่อไฟล์ หมายเลขของโปรเซส ชื่อโปรเซส และระบุเวลาที่ตรวจสอบได้ ซึ่งในการนำไปพิจาณา ค้นหาผู้กระทำผิดนั้น อาจจะมีข้อมูลในขั้นตอนการพิจารณาค้นหาน้อยเกินไป ดังนั้นผู้วิจัยจึงได้ศึกษาหาโครงสร้างข้อมูลที่มีรายละเอียดที่สามารถช่วยให้กระบวนการพิจารณา ค้นหาผู้รับผิดชอบนั้นมีความรวดเร็วและครอบคลุม โดยโครงสร้างข้อมูลดังกล่าวจะถูกกล่าวถึงในหัวข้อถัดไป

โครงสร้างข้อมูลไอโหนด (inode Struct)

ในหัวข้อนี้จะอธิบายถึงความสำคัญของโครงสร้างข้อมูลไอโหนด เป็นข้อมูลที่สำคัญที่ใช้ในการเพิ่มประสิทธิภาพของการทำงานของระบบบันทึกเหตุการณ์ในวิทยานิพนธ์เล่มนี้ จากการทดลองใช้ระบบบันทึกเหตุการณ์เดิม ในการทดลองนั้นมีข้อมูลที่ได้จากวิธีการ Introspection ของ LibVMI ได้แก่ ผู้ใช้งานโปรเซส ชื่อไฟล์ หมายเลขของโปรเซส ชื่อโปรเซส และระบุเวลาที่ตรวจสอบได้ ซึ่งพบว่าข้อมูลมีจำนวนน้อยมากในการนำไปเป็นหลักฐานในการพิจารณา ค้นหา ผู้รับผิดชอบต่อการกระทำผิดที่ส่งผลทำให้เกิดความเสียหาย ดังนั้นผู้วิจัยจึงนำเสนอการเพิ่มจำนวนข้อมูลในล็อกไฟล์ โดยใช้โครงสร้างข้อมูลไอโหนดที่ มีรายละเอียดต่าง ๆ ที่เกี่ยวกับกระทำต่อไฟล์ข้อมูล ดังนี้

1. i_size เป็นข้อมูลที่แสดงถึงขนาดของไฟล์ข้อมูล

2. i_mode เป็นการกำหนดสิทธิของการใช้งานไฟล์ข้อมูลเช่น –RWXRWXRWX โดย R มีความหมายว่าสามารถเข้าดูข้อมูลได้ Wมีความหมายว่าสามารถเขียนหรือเปลี่ยนแปลงข้อมูลได้ X มีความหมายว่าสามารถ execute ได้ และยังสามารถแบ่งเป็นสามกลุ่มด้วยกันคือ สามตัวแรกมีหมายถึงเจ้าของไฟล์ สามตัวถัดมาหมายถึงกลุ่มของผู้ใช้ และสามตัวสุดท้ายคือผู้ใช้ทุกคน

3. i_uid เป็นข้อมูลของผู้ที่มีสิทธิครอบครองและเป็นเจ้าของไฟล์ข้อมูล เป็นข้อมูลหมายเลขที่ไม่ซ้ำกัน โดยปกติหมายเลขชุดนี้จะถูกกำหนดโดยระบบปฏิบัติการตั้งแต่ขั้นตอนการสร้างชื่อผู้ใช้งาน (Username) ใหม่เพิ่มในระบบปฏิบัติการ โดยผู้ใช้จะใช้ ชื่อผู้ใช้งานในการทำงาน ระบบปฏิบัติการจะรู้จักผู้ใช้งานผ่านหมายเลขนี้เท่านั้น สร้างชื่อผู้ใช้งานใหม่ระบบปฏิบัติการจะมีการกำหนดตัวเลข โดยปกติผู้ใช้ทั่วไปจะมีหมายเลขเริ่มต้นที่ 1001 เป็นต้นไป

4. i_gid เป็นข้อมูลของกลุ่มของผู้ใช้ที่มีสิทธิเข้าถึงไฟล์ข้อมูล โดยปกติข้อมูลจะเป็นตัวเลข โดยหากเป็นกลุ่มของระบบจะเป็นเลข 0-1000 และหากเป็นกลุ่มที่ถูกสร้างขึ้นใหม่จะเริ่มต้นที่ 1001 เป็นต้นไป กลุ่มของผู้ใช้ (GID) ในระบบปฏิบัติการใช้แทนรายชื่อกลุ่ม (Group name) โดยทั่วไปผู้ใช้มีกลุ่มหลัก (หรือค่าเริ่มต้น) ที่กำหนดไว้และยังสามารถเป็นสมาชิกของกลุ่มเพิ่มเติมที่เรียกว่ากลุ่มรอง เมื่อผู้ใช้สร้างไฟล์หรือเรียกใช้โปรแกรมไฟล์และโปรแกรมเหล่านั้นจะเชื่อมโยงกับกลุ่มหลักหรือกลุ่มรอง ผู้ใช้ที่เป็นส่วนหนึ่งของกลุ่มสามารถเข้าถึงไฟล์ข้อมูลเหล่านี้ได้ตามสิทธิ์การใช้งาน

5. i_ino เป็นข้อมูลที่เก็บในลักษณะของกลุ่มตัวเลขหนึ่งชุด จะไม่มีวันเกิดขึ้นซ้ำได้อีก และจะไม่เปลี่ยนแปลงจนกว่าจะมีการย้ายที่อยู่หรือลบและถูกสร้างใหม่ของไฟล์ข้อมูล

6. i_nlink เป็นข้อมูลที่มีลักษณะของการสำรองไฟล์ข้อมูลไว้อีกทีหนึ่งเพื่อป้องกันการสูญหายหรือไม่ต้องการให้ผู้ใช้เข้าถึงไฟล์ได้โดยตรง โดยปกติข้อมูลจะมีค่าเริ่มต้นเท่ากับ 1 หากมีการสำรองไฟล์ข้อมูลจะโดยคำสั่ง ln ข้อมูลจะเปลี่ยน 2 หรือตามจำนวนการสำรองไฟล์ที่เกิดขึ้นบวกหนึ่ง เช่น ผู้ใช้งานต้องการลิงค์ ไฟล์ข้อมูล s.txt ชื่อว่า z.txt โดยใช้คำสั่ง ln s.txt z.txt เมื่อสร้าง z.txt สำเร็จจะเห็นได้ว่า z.txt มีข้อมูล เหมือนกับ s.txt ทุกประการ พร้อมทั้งเมื่อมีการเปลี่ยนแปลงแก้ไขข้อมูลใน s.txt ไฟล์ข้อมูล z.txt ก็ จะเปลี่ยนแปลงไปด้วย ในทางกลับกันหากมีการเปลี่ยนแปลงใน z.txt ไฟล์ข้อมูล s.txt ก็จะเปลี่ยนแปลงเช่นเดียวกัน และจำนวนตัวเลข 1 จะเปลี่ยนเป็น 2

7. i_atime เป็นข้อมูลที่แสดงเวลาของผู้ใช้งานนี้ล่าสุดโดยข้อมูลนี้จะถูกเปลี่ยนแปลงทุกครั้งที่มีการเข้าถึงไฟล์ข้อมูล

8. i_mtime เป็นข้อมูลที่แสดงถึงการเปลี่ยนแปลงครั้งล่าสุดของไฟล์โดยหากมีการเปลี่ยนแปลงข้อมูลในไฟล์ข้อมูล

9. i_ctime เป็นข้อมูลที่บันทึกเกี่ยวกับการเปลี่ยนแปลงของเจ้าของไฟล์ข้อมูล กลุ่มของผู้ใช้และคุณสมบัติของไฟล์ข้อมูล ซึ่งจะมีความแตกต่างกันของข้อมูล

ดังนั้นจากข้อมูลที่อยู่ในโครงสร้างข้อมูลไอโหนดจะทำให้ ล็อกไฟล์ นั้นมีข้อมูลมากเพียงพอต่อการตรวจสอบและค้นหาผู้ไม่ประสงค์ดีมารับผิดชอบได้

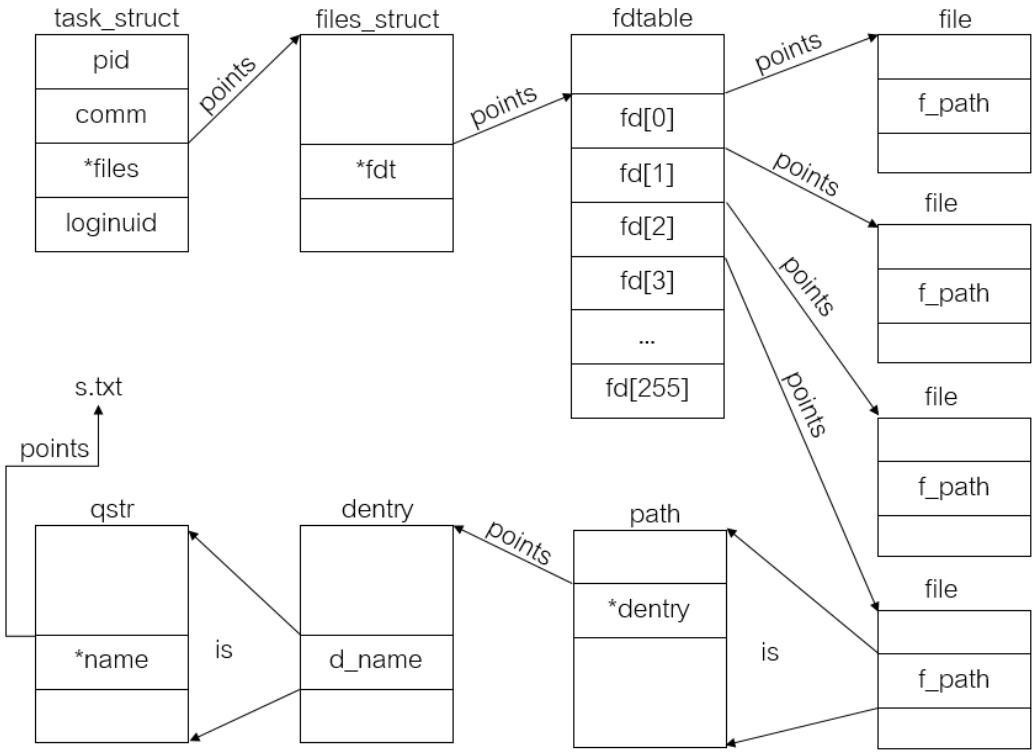
วิธีการได้ผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

การได้ผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์จะต้องเข้าใจการทำงานของระบบปฏิบัติการลินุกซ์และโครงสร้างข้อมูล สำหรับเนื้อหาในหัวข้อนี้ส่วนแรกจะอธิบายและทำความเข้าใจเกี่ยวกับ ระบบปฏิบัติการลินุกซ์ (Linux) เคอร์เนล (Kernel) โปรเซส (Process) หน่วยความจำหลัก (RAM) ซอร์สโค้ด (Source code) และ User space และส่วนที่สองจะอธิบายเกี่ยวกับโครงสร้างข้อมูล

1. การทำงานของระบบปฏิบัติการลินุกซ์เกี่ยวกับหน่วยความจำหลัก

เอกสารที่นิยามเกี่ยวกับคำศัพท์ของระบบปฏิบัติการลินุกซ์ ที่ถูกกล่าวไว้ในงานของ Homscheid (2020) กล่าวถึงองค์กรที่ไม่แสวงหาผลกำไร Bellevue Linux User Group (BELUG) เพื่อส่งเสริมความน่าสนใจและการใช้งานของระบบปฏิบัติการลินุกซ์ ในเอกสารนี้ได้อธิบายความหมายของเคอร์เนล (Kernel) ว่าโปรแกรมศูนย์กลางทำหน้าที่ควบคุมบริหารจัดการ หน่วยประมวลผล หน่วยความจำ หน่วยรับข้อมูลและหน่วยแสดงผลข้อมูล รวมไปถึงซอฟท์แวร์หรือโปรแกรมต่างๆที่ถูกติดตั้ง ภายใต้สภาพแวดล้อมของระบบปฏิบัติการลินุกซ์ โดยเริ่มต้นตั้งแต่

ระบบปฏิบัติการเปิดการทำงาน จะมีจัดสรรพื้นที่ของหน่วยความจำหลักบางส่วนให้เก็บข้อมูลที่เกี่ยวข้องกับเคอร์เนลไว้ พื้นที่ส่วนนี้จะไม่มีการถูกใช้งานจากโปรแกรมใดๆ การเรียกพื้นที่ที่ถูกจัดสรรนี้ว่า Kernel space เมื่อใดที่ผู้ใช้งานต้องการใช้งานกับอุปกรณ์ จะต้องทำการติดต่อหรือร้องขอไปยังเคอร์เนล ซึ่งกระบวนการนี้เรียกว่า System call ในระบบบันทึกเหตุการณ์จะมีการใช้งานเคอร์เนลจัดการหน่วยความจำหลักหรือ RAM (Read Access Memory) ซึ่งหน่วยความจำหลักนั้นจะมีความสามารถเก็บข้อมูลที่ต้องการประมวลผลไว้ชั่วคราว เนื่องจากข้อมูลที่ถูกเก็บไว้บนหน่วยความจำหลักนั้นสามารถอ่านและเขียนได้อย่างรวดเร็ว หน่วยความจำหลักนี้เป็นตัวกลางที่ทำหน้าที่พักข้อมูลระหว่างการประมวลผล เก็บข้อมูลบางส่วนของระบบปฏิบัติการ โปรแกรมหรือข้อมูลที่มีแนวโน้มถูกใช้งานบ่อยๆ ณ เวลาที่หน่วยความจำหลักทำงานจะประกอบด้วย ข้อมูลของเคอร์เนล ข้อมูลของภาษาเครื่องหรือรหัสโค้ด (Machine code) ของโปรเซสที่กำลังประมวลผล ข้อมูลของตัวแปรของโปรเซส และสุดท้ายคือการสำเนาไฟล์ข้อมูลที่ถูกเปิดใช้งาน ซึ่งการสำเนาไฟล์ข้อมูลที่ถูกเปิดใช้งานคือโปรแกรมที่ถูกแปลให้เป็นภาษาเครื่องโดยการคอมไฟล์จากซอร์สโค้ดให้เป็นเลขฐานสอง (Binary number) ตัวอย่างการทำงานเช่น เมื่อมีการเปิดใช้งานไฟล์ข้อมูลบนหน่วยความจำสำรอง (Storage disk) ระบบปฏิบัติการจะทำการสำเนาไฟล์ข้อมูลดังกล่าวเก็บไว้ในหน่วยความจำหลัก และเมื่อเสร็จสิ้นกระบวนการจะถูกนำไปเขียนลงในตำแหน่งเดิมของหน่วยความจำสำรองหรือเขียนลงบนอุปกรณ์อื่นๆที่กำหนดไว้ ดังนั้นจะเห็นว่าทุกครั้งที่ไฟล์ข้อมูลถูกเรียกใช้งานจะถูกสำเนาไปไว้ในหน่วยความจำหลักเสมอ จะเห็นได้ว่าก่อนมีการสำเนาไฟล์ข้อมูลจะต้องมีกระบวนการที่เกิดขึ้นเรียกว่าโปรเซส (Process) ซึ่งเป็นกระบวนการทำงานที่ถูกใช้งาน (Running) หรือ เอ็กซีคิว (Execute) โปรแกรมที่ถูกเขียนด้วยโปรแกรมภาษาคอมพิวเตอร์ เช่น ภาษา C โปรแกรมที่ถูกเขียนขึ้นนี้เรียกว่า ซอร์สโค้ด (Source code) พื้นที่หน่วยความจำหลักที่เก็บข้อมูลเมื่อโปรเซสเกิดขึ้นจะถูกเรียกว่า User space



ภาพ 5 แสดงโครงสร้างตัวแปรข้อมูลใน task_struct

2. โครงสร้างข้อมูล

จากหัวข้อเรื่อง การทำงานของระบบปฏิบัติการลินุกซ์เกี่ยวกับหน่วยความจำหลัก นั้นทำให้เข้าใจถึงหลักการทำงานของหน่วยความจำหลัก ในระบบบันทึกเหตุการณ์ในวิทยานิพนธ์เล่มนี้จะอาศัยการทำงานของ LibVMI ซึ่งเป็นไลบรารี่ของโปรแกรมภาษาซี โดยวิธี Introspection ที่มีความสามารถอ่านข้อมูลในหน่วยความจำหลักเสมือนของเครื่องคอมพิวเตอร์เสมือน ดังนั้นจึงจำเป็นต้องเข้าใจโครงสร้างข้อมูลเพื่อให้ได้ผลลัพธ์ที่ต้องการ

ในการทำงานของระบบปฏิบัติการเมื่อมีการเปิดใช้งานโปรแกรมจะเกิดโปรเซสขึ้น จะมีการสำเนาไฟล์ข้อมูลที่เกี่ยวข้องกับกระบวนการประมวลผลไปไว้ที่หน่วยความจำ โดยจะสร้างโครงสร้างข้อมูลชื่อว่า task_struct และในโครงสร้างข้อมูลนี้จะประกอบไปด้วยตัวแปรข้อมูลหลายๆชนิดอยู่ภายในเช่น แบบอักขระ แบบจำนวนเต็ม แบบพอยน์เตอร์ เป็นต้น ในเวลาที่ระบบปฏิบัติการทำงานจะมีโครงสร้างข้อมูล task_struct เกิดขึ้นมากมายตามจำนวนงานที่ระบบปฏิบัติการ กำลังทำงานอยู่ในขณะนั้น ในโครงสร้างข้อมูลนี้จะมีข้อมูลตัวแปรที่มีความจำเป็นต่อการได้มาซึ่งผลลัพธ์ของระบบบันทึกเหตุการณ์ ดังภาพ 5 ซึ่งจะประกอบไปด้วย

1. pid หรือ Process ID เป็นข้อมูลหมายเลขลำดับของโปรเซสที่ถูกสร้างขึ้นหลังจากเรียกใช้งานโปรแกรมภายในระบบปฏิบัติการ

2. comm หรือ Process name เป็นข้อมูลชื่อของโปรเซสที่เกิดถูกสร้างขึ้นหลังจากเรียกใช้งานโปรแกรมภายในระบบปฏิบัติการ

3. files เป็นข้อมูลตัวแปรแบบพอยน์เตอร์ โดยจะชี้ไปยังตำแหน่งที่ของโครงสร้างข้อมูล files_struct ซึ่งเป็นโครงสร้างข้อมูลที่เก็บข้อมูลต่างๆของไฟล์ข้อมูลที่โปรเซสเรียกใช้งาน

4. loginuid เป็นหมายเลขของผู้ใช้งานที่เรียกใช้โปรแกรม

Sysbench

เป็นโปรแกรมที่ใช้วัดประสิทธิภาพ (benchmark) ของเครื่องคอมพิวเตอร์ได้หลายอย่าง ทั้ง หน่วยประมวลผลกลาง (CPU) หน่วยความจำหลัก (Main memory) หน่วยความจำสำรอง (Storage disk) และ ฐานข้อมูล (Database) เป็นต้น ทั้งนี้ยังสามารถระบุฟังก์ชั่นสำหรับการทดสอบได้เช่น จำนวนแกนของหน่วยประมวลผลกลาง (CPU Core) จำนวนเทรด (Thread) ที่ทำการประมวลผล ขนาดที่จะทดสอบ ระยะเวลาที่ใช้ในการประมวลผล ซึ่งผลลัพธ์ที่ได้ในการทดสอบนำมาเปรียบเทียบเพื่อตัดสินใจในการเลือกซื้อ แก้ไข และปรับปรุงให้มีการทำงานได้ดีขึ้น Sysbench ในงานของ Kopytov (2017) ได้นิยามว่าเป็นโปรแกรมที่มีขนาดเล็กการติดตั้งโปรแกรมและการใช้งานจะเป็นรูปแบบการใช้คำสั่งแบบ Command line

การใช้ Sysbench สามารถใช้วัดประสิทธิภาพการทำงานของคอมพิวเตอร์เสมือนได้โดยมีขั้นตอนดังนี้

1. การติดตั้ง Sysbench ผู้ใช้งานจะต้องทำการติดตั้ง Sysbench บนเครื่องคอมพิวเตอร์เสมือนก่อน โดยใช้คำสั่งเพื่อติดตั้ง Sysbench จากที่มาตรฐาน เช่น apt-get หรือ yum เป็นต้น

2. การติดตั้งและกำหนดค่าเครื่องเสมือน การติดตั้งและกำหนดค่าเครื่องเสมือนที่จะใช้ทดสอบ เช่น ขนาด RAM, CPU core, และความเร็วของเครื่องเสมือน โดยใช้โปรแกรมจำลองเครื่องเสมือน เช่น VirtualBox หรือ VMware

3. การทดสอบ Sysbench จะทำการทดสอบการทำงานของเครื่องเสมือนโดยใช้คำสั่งที่เหมาะสม เช่น sysbench cpu สำหรับทดสอบประสิทธิภาพ CPU หรือ sysbench fileio สำหรับทดสอบประสิทธิภาพการอ่าน/เขียนไฟล์

4. การสรุปผล Sysbench จะสรุปผลการทดสอบและแสดงรายงานผลการทดสอบเพื่อให้ผู้ใช้สามารถวิเคราะห์ประสิทธิภาพของเครื่องเสมือนได้

การใช้ Sysbench เพื่อวัดประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์เสมือน สามารถช่วยให้ผู้ใช้งานได้รับข้อมูลเกี่ยวกับประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์เสมือน ทำให้สามารถนำข้อมูลที่ได้ นำมาปรับปรุงประสิทธิภาพการทำงานให้มีประสิทธิภาพได้ดีขึ้นตรงตามความต้องการของผู้ใช้งานมีผู้วิจัยใช้ Sysbench เป็นเครื่องมือที่ใช้วัดประสิทธิภาพการทำงาน เช่นงานวิจัยเรื่อง A container-based technique to improve virtual machine migration in cloud computing ของ Bhardwaj & Rama Krishna (2022) เกี่ยวกับการวัดประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์เสมือนเมื่อมีการย้ายข้อมูล งานวิจัยเรื่อง An architecture supervisor scheme toward performance differentiation and optimization in cloud systems ของ Fareghzadeh (2022)เกี่ยวกับการวัดประสิทธิภาพการทำงานและการจัดการบนคลาวด์ให้เหมาะสมกับการใช้งานของผู้ใช้บริการ เป็นต้น

สรุปภาพรวมบทที่ 2

จากเป้าหมายหลักของวิทยานิพนธ์ ผู้วิจัยได้ทำการศึกษาและได้ระบุเพื่อให้บรรลุวัตถุประสงค์ของงานวิจัยและสามารถออกแบบการทดลองเพื่อปรับปรุงในการเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ และลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงานทั้งในส่วนของผู้ให้บริการ โดยรายละเอียดของปัญหาวิจัยหลักหรือ research gap ของวิทยานิพนธ์เล่มนี้สามารถอธิบายเป็นรายข้อได้ 3 หัวข้อ โดยแต่ละหัวข้อจะสอดคล้องกับเอกสารและงานวิจัยที่เกี่ยวข้อง ดังนี้

1. จากการศึกษาวรรณกรรมที่เกี่ยวข้องที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์และทดลองใช้งาน พบว่าระบบบันทึกเหตุการณ์ยังมีข้อจำกัดเกี่ยวกับ การทำงานในการจัดการทรัพยากร โดยผู้วิจัยจะทำการศึกษาข้อมูลในหัวข้อเรื่อง ข้อมูลพื้นฐานของคลาวด์ คลาวด์ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ ปัญหาภัยคุกคามของคลาวด์ วิธีการรักษาความปลอดภัยของการละเมิดข้อมูล ระบบบันทึกเหตุการณ์ และ libVMI

2. จากการศึกษาวรรณกรรมที่เกี่ยวข้องที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์และทดลองใช้งาน พบว่าข้อมูลที่ถูกจัดเก็บไว้ใน ล็อกไฟล์ของระบบบันทึกเหตุการณ์นั้นยังไม่ครอบคลุมการพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูลของผู้ใช้ จากข้อมูลที่ผู้วิจัยได้ทำการศึกษาในหัวข้อเรื่อง ล็อกไฟล์ ระบบบันทึกเหตุการณ์ libVMI โครงสร้างข้อมูลไอโหนด (inode Struct) และ วิธีการได้ผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

3. จากการศึกษาวรรณกรรมที่เกี่ยวข้องที่เกี่ยวข้องกับระบบบันทึกเหตุการณ์และทดลองใช้งาน ยังไม่ปรากฏการวัดและทดสอบประสิทธิภาพการทำงานในส่วนของผู้ใช้บริการเมื่อระบบบันทึกเหตุการณ์กำลังทำงานในเวลาเดียวกัน โดยเมื่อมีการใช้งานระบบบันทึกเหตุการณ์จะมีผลกระทบกับการทำงานโดยรวมของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการ ซึ่งจะทำการวัดและทดสอบทำงานโดยรวมของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการว่าลดลงมากน้อยเพียงใด อีกทั้งยังจะนำเสนอรูปแบบการใช้งานระบบบันทึกเหตุการณ์ให้ตรงตามความต้องการของผู้ใช้บริการ ในส่วนของผลกระทบการทำงานโดยรวมของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการที่เหมาะสมโดย Sysbench ซึ่งเครื่องมือในการวัดประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการ

บทที่ 3

วิธีดำเนินการวิจัย

ผู้วิจัยได้ทำการศึกษาแนวคิด ทฤษฎี ตลอดจนเอกสารจากเนื้อหาบทที่ 2 ที่ได้กล่าวถึงเอกสารและงานวิจัยที่เกี่ยวข้องเพื่อเป็นพื้นฐานความรู้สำหรับการวิจัยในครั้งนี้ ดังนั้นผู้วิจัยจึงได้อธิบายเกี่ยวกับ เครื่องมือ วิธีการ และขั้นตอน โดยกำหนดหัวข้อในบทที่ 3 วิธีดำเนินงานวิจัย ซึ่งประกอบด้วย 7 หัวข้อ ดังนี้

1. เครื่องมือที่ใช้ในการวิจัย
2. กรอบวิธีการดำเนินวิจัย
3. การติดตั้งซอฟท์แวร์ที่ใช้ในการทำลอง
4. การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม
5. การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์
6. การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์
7. การวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์
8. สรุปภาพรวมบทที่ 3

เครื่องมือที่ใช้ในการวิจัย

หัวข้อนี้จะกล่าวถึงเครื่องมือที่ใช้ในงานวิจัย ซึ่งในงานวิทยานิพนธ์เล่มนี้ได้แบ่งฮาร์ดแวร์ (Hardware) และซอฟท์แวร์ (software) สำหรับการทำงานไว้อย่างละ 2 ส่วน คือ ส่วนสำหรับการทดลอง และ ส่วนสำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์โดยมีรายละเอียด ดังนี้

1. ฮาร์ดแวร์ (Hardware)

ส่วนของฮาร์ดแวร์ผู้วิจัยได้แบ่งเป็น 2 ส่วนตามที่ได้กล่าวข้างต้น คือ 1. ฮาร์ดแวร์สำหรับการทดลอง และ 2. ฮาร์ดแวร์สำหรับบันทึกผลการทดลองและเขียน วิทยานิพนธ์ โดยรายละเอียดมี ดังนี้

1.1 ฮาร์ดแวร์สำหรับการทดลอง

1.1.1 คอมพิวเตอร์ PC 1 เครื่อง (personal computer)

1.1.2 CPU Intel ® i5 Processor ขนาด 4 core

1.1.3 RAM DDR 3 ขนาด 4 GB.

1.1.41Hard Disk ขนาด 320 GB.

1.2 ฮาร์ดแวร์สำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์

1.2.1 คอมพิวเตอร์ โน้ตบุ๊ก 1 เครื่อง (Notebook computers)

1.2.2 CPU Intel® Core(TM) i7-3610QM @ 2.30 GHz 2.30 GHz

1.2.3 RAM DDR 3 ขนาด 16 GB.

1.2.4 Hard Disk ขนาด 1 TB.

1.2.5 Graphics Card NVidia GEFORCE® GT 730M 4 GB

2. ซอฟท์แวร์ (software)

ส่วนของซอฟท์แวร์ผู้วิจัยได้แบ่งเป็น 2 ส่วนตามที่ได้กล่าวข้างต้น คือ ซอฟท์แวร์สำหรับการทดลอง และ ซอฟท์แวร์สำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์ โดยรายละเอียดมี ดังนี้

2.1 ซอฟท์แวร์สำหรับการทดลอง

2.1.1 ระบบปฏิบัติการ Linux fedora 16 (64 bit)

2.1.2 terminal commands Linux

2.1.3 screenshot

2.1.4 Xen เวอร์ชัน 4.2.5 (version 4.2.5)

2.2 ซอฟท์แวร์สำหรับบันทึกผลการทดลองและเขียนวิทยานิพนธ์

2.2.1 Windows 10 Professional (64-bit)

2.2.2 Microsoft word 2016

2.2.3 Microsoft excel 2016

2.2.4 Microsoft PowerPoint 2016

2.2.5 Adobe Photoshop

เมื่อผู้อ่านได้ทราบเกี่ยวกับฮาร์ดแวร์และซอฟท์แวร์ต่าง ๆ ที่เกี่ยวข้องในการทดลองของวิทยานิพนธ์เล่มนี้จากหัวข้อเรื่อง เครื่องมือที่ใช้ในการวิจัย แล้ว ในหัวข้อถัดไปผู้วิจัยจะอธิบายเกี่ยวกับกรอบวิธีการดำเนินงานวิจัย

กรอบวิธีการดำเนินงานวิจัย

ในหัวข้อนี้จะนำเสนอกรอบวิธีการดำเนินการวิจัย หลักการ รวมไปถึงการออกแบบการทดลอง โดยผู้วิจัยได้แบ่งการทดลองออกเป็น 2 ส่วน คือ วิธีการศึกษาและเตรียมข้อมูลสำหรับการทดลอง และวิธีการออกแบบการทดลอง โดยในแต่ละหัวข้อจะมีรายละเอียด ดังนี้

1. วิธีการศึกษาและเตรียมข้อมูลสำหรับการทดลอง

หัวข้อการศึกษาและเตรียมข้อมูลสำหรับการทดลองนี้จะอธิบายเกี่ยวกับวิธีการศึกษาที่ผู้วิจัยได้ทำการศึกษา รวบรวมจนกระทั่งออกแบบการทดลองเพื่อปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ได้ โดยมีรายละเอียด ดังนี้

ในส่วนของการศึกษาผู้วิจัยได้เริ่มทำการศึกษาเกี่ยวกับความเป็นมาของคลาวด์ ประเภทของคลาวด์ ไปจนถึงภัยคุกคามที่เกิดขึ้นกับคลาวด์ เมื่อทราบว่าภัยคุกคามที่เกิดขึ้นกับคลาวด์ส่งผลกระทบต่อความเชื่อมั่นในการนำคลาวด์มาใช้งานจริงของผู้ใช้บริการ ที่เป็นส่วนบุคคลและองค์กร ผู้วิจัยจึงได้ศึกษาเกี่ยวกับวิธีการบรรเทาปัญหาที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ นั้นคือระบบบันทึกเหตุการณ์ ซึ่งระบบบันทึกเหตุการณ์เป็นหนึ่งในวิธีการที่สามารถบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ งานวิจัยที่ได้นำเสนอเกี่ยวกับระบบบันทึกเหตุการณ์ไว้มากมาย ระบบบันทึกเหตุการณ์ คือระบบที่บันทึกเหตุการณ์เกี่ยวกับเหตุการณ์และกระทำต่าง ๆ ต่อไฟล์ข้อมูล เช่น ลบ แก้ไข เปลี่ยนแปลงหรืออ่านไฟล์ข้อมูล โดยระบบบันทึกเหตุการณ์ต้องสามารถทำงานได้อย่างรวดเร็วถูกต้องแม่นยำรวมไปถึงข้อมูลที่บันทึกได้ จะต้องมีความเพียงพอต่อการนำไปใช้หลักฐานในการพิจาณาเพื่อหาความรับผิดชอบต่อการกระทำทีเกิดขึ้น ซึ่งในการศึกษาข้อมูลเบื้องต้นผู้วิจัยพบว่าการทำงานของระบบบันทึกเหตุการณ์นั้นยังมีข้อจำกัดอยู่หลายด้านเช่น

1.1 ในด้านการทำงานของระบบบันทึกเหตุการณ์ เกี่ยวกับ ความรวดเร็ว จากการศึกษาการทำงานของระบบบันทึกเหตุการณ์เดิมนั้น จะต้องมีการกำหนดระยะหน่วงหรือ Sleeping time อยู่ในช่วง 70-100 ms ในการทำงานเพื่อตรวจสอบข้อมูลในหน่วยความจำหลักซึ่งอาจส่งผลต่อความความถูกต้องแม่นยำในการตรวจจับเหตุการณ์ที่เกิดขึ้นต่อการกระทำกับไฟล์ข้อมูล อีกด้านหนึ่งที่พบและเป็นปัญหาที่สำคัญคือการจัดหน่วยความจำในการทำงานที่ยังไม่ดีพอ เมื่อมีการทำงานของระบบบันทึกเหตุการณ์จะมีปริมาณการใช้งานของระบบเพิ่มขึ้น หากว่ามีการเปิดใช้งานเป็นระยะเวลานานจะทำให้หน่วยความจำหลักภายในเครื่องเต็มส่งผลเกิดปัญหาในการทำงานของเครื่องคอมพิวเตอร์ที่รันระบบบันทึกเหตุการณ์ เช่น ทำงานช้าลง ค้าง หรือหยุดการทำงาน

1.2 ในด้านผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ ข้อมูลยังไม่เพียงพอ ซึ่งจากการศึกษาพบว่ามีข้อมูลที่ได้จากการบันทึกเหตุการณ์ ได้แก่ ผู้ใช้งานโปรเซส ชื่อไฟล์ หมายเลขของโปรเซส ชื่อโปรเซส และระบุเวลาที่ตรวจสอบ ซึ่งหากว่ามีข้อมูลที่เกี่ยวข้องในการกระทำเกี่ยวกับไฟล์ที่มากขึ้นจะส่งผลให้การพิจารณาหรือตรวจสอบอย่างมีประสิทธิภาพได้ดีขึ้น

1.3 ในสุดท้ายคือประสิทธิภาพการทำงานของผู้ใช้บริการ เมื่อมีการเปิดใช้งานระบบบันทึกเหตุการณ์ ซึ่งยังไม่มีงานวิจัยใดวัดประสิทธิภาพในด้านการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

เมื่อผู้วิจัยเห็นปัญหาและแนวทางการปรับปรุงประสิทธิภาพผู้วิจัยจึงได้ออกแบบการทดลองเพื่อปรับปรุงและเพิ่มประสิทธิภาพระบบบันทึกเหตุการณ์ โดยรายละเอียดจะอธิบายเพิ่มเติมในหัวข้อถัดๆ ไป ในหัวข้อเรื่อง วิธีการศึกษาและเตรียมข้อมูลสำหรับการทดลองนี้ จากเอกสารที่ได้ศึกษาและค้นคว้าเพื่อเตรียมออกแบบการทดลองในการปรับปรุงประสิทธิภาพ ได้ถูกกล่าวไว้ในบทที่ 2 เรื่อง เอกสารและงานวิจัยที่เกี่ยวข้อง ซึ่งรายเอียดของการออกแบบการทดลองและผลของการทดลองที่ผู้วิจัยได้ออกแบบจะกล่าวเพิ่มเติมในหัวข้อถัดไป

2. วิธีการออกแบบการทดลอง

ในหัวข้อนี้จะอธิบายเกี่ยวกับวิธีการออกแบบการทดลอง ซึ่งได้ศึกษาเอกสารและโปรแกรมจากงานวิจัยที่เกี่ยวข้องต่างๆ จากบทที่ 2 และหัวข้อวิธีการศึกษาและเตรียมข้อมูลสำหรับการทดลอง นำข้อมูลที่ศึกษามาออกแบบการทดลองและนำผลจากการทดลองที่ได้ไปวิเคราะห์หาสาเหตุ เพื่อทำการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์โดยวิธีการออกแบบการทดลองได้แบ่งส่วนของการทดลองไว้ 4 ด้านคือ 1) ส่วนของการทดลองการทำงานของระบบบันทึกเหตุการณ์เดิมเพื่อหาข้อจำกัดในการทำงานของระบบ 2) ส่วนการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ 3) ส่วนของการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ และสุดท้ายคือ 4) ส่วนการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

1. ส่วนของการทดลองการทำงานของระบบบันทึกเหตุการณ์เดิมเพื่อหาข้อจำกัดในการทำงานของระบบ ในส่วนของการทดลองการทำงานของระบบบันทึกเหตุการณ์นั้น ได้มีการติดตั้งระบบบันทึกเหตุการณ์ นำระบบบันทึกเหตุการณ์มาทดลองทำงานอยู่บนทรัพยากรที่มีการเปลี่ยนแปลงขนาดของหน่วยความจำหลัก (RAM) และจำนวนแกนของหน่วยประมวลผลกลาง (CPU Core) ในเครื่องของผู้ให้บริการ (dom0) และผู้ใช้บริการ (domU) การปรับเปลี่ยนค่าหน่วงเวลาของระบบบันทึกเหตุการณ์ การวัดความแม่นยำในการตรวจสอบ โดยมีจุดมุ่งหมายเพื่อที่จะหาข้อจำกัดในการทำงานของระบบบันทึกเหตุการณ์

2. ส่วนการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ ส่วนนี้จะเป็นการนำข้อมูลต่างๆที่ได้จากการทดลองใช้งานระบบบันทึกเหตุการณ์และการศึกษาโครงสร้างข้อมูลของระบบปฏิบัติการมาเพื่อทำการแก้ไข ดัดแปลง เพิ่มเติม เพื่อทำให้การทำงานของระบบบันทึกเหตุการณ์มีการทำงานทำงานที่รวดเร็ว ถูกต้องแม่นยำ และมีผลลัพธ์ที่มากขึ้น

3. ส่วนของการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ ในส่วนนี้จากการทำงานของระบบบันทึกเหตุการณ์นั้นใช้หลักการทำงานของ LibVMI ที่มีความสามารถเข้าถึงข้อมูล (read_mem) ในหน่วยความจำหลักเสมือน (memU) ของเครื่องคอมพิวเตอร์เสมือน (domU) ดังภาพ 3 ซึ่งผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ได้จากโครงสร้างข้อมูลของระบบปฏิบัติการ ในวิทยานิพนธ์เล่มนี้ใช้ระบบปฏิบัติการลินุกซ์ (Linux) ในการทดลอง ดังนั้นจึงต้องศึกษาโครงสร้างข้อมูลที่อยู่ในหน่วยความจำหลักเพื่อเพิ่มข้อมูลบันทึกเหตุการณ์หรือผลลัพธ์ของระบบบันทึกเหตุการณ์

4. ส่วนการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ ในส่วนนี้จะทำการวัดและทดสอบประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการเมื่อมีการใช้งานระบบบันทึกเหตุการณ์ เนื่องจากการทำงานของระบบบันทึกเหตุการณ์มีผลกระทบต่อการทำงานของผู้ใช้บริการในเครื่องคอมพิวเตอร์เสมือน เพราะว่าการทำงานของ LibVMI ที่เป็นเครื่องมือสำคัญของระบบบันทึกเหตุการณ์ที่เข้าไปตรวจสอบหน่วยความจำหลักของเครื่องคอมพิวเตอร์เสมือนนั้นจะส่งผลทำให้การทำงานหรือการประมวลผลของผู้ใช้บริการนั้นใช้เวลามากขึ้น

การติดตั้งซอฟท์แวร์ที่ใช้ในการทำลอง

ในการทดลองในวิทยานิพนธ์เล่มนี้ ระบบบันทึกเหตุการณ์เป็นหัวใจสำคัญที่ใช้ในการทดลองและทดสอบการทำงาน ดังนั้นจึงต้องมีการติดตั้งระบบปฏิบัติการ โปรแกรมบันทึกเหตุการณ์ และโปรแกรมต่างๆ ที่ใช้ในการทดลองและทดสอบ เพื่อนำข้อมูลที่ได้มาใช้ในการศึกษาวิเคราะห์ และนำไปสู่การพัฒนาปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ ในหัวข้อนี้จะแสดงถึงขั้นตอนการติดตั้งระบบบันทึกเหตุการณ์

1. ขั้นตอนที่ 1 ติดตั้งฟีโดร่า (Fedora) ซึ่งเป็นระบบปฏิบัติการลินุกซ์ตระกูลหนึ่งที่นิยมใช้ในการทดลองและทดสอบเนื่องจากเป็นฟรีแวร์ เพื่อใช้เป็นระบบปฏิบัติการพื้นฐานสำหรับสร้างคลาวด์ IaaS และเป็นการทำงานในส่วนของ dom0

2. ขั้นตอนที่ 2 ติดตั้งซอฟแวร์ Xen ซึ่งเป็น hypervisor ทำหน้าที่สร้างและจัดการเครื่องคอมพิวเตอร์เสมือนที่ให้บริการแก่ผู้ใช้บริการ

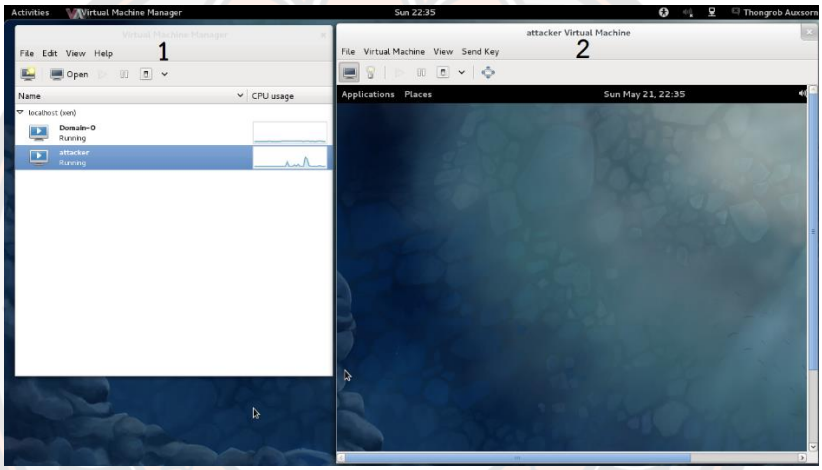
3. ขั้นตอนที่ 3 ติดตั้ง LibVMI ซึ่งเป็นไลบรารีของภาษาซีบน dom0

4. ขั้นตอนที่ 4 ติดตั้งโปรแกรมบันทึกเหตุการณ์ (logger) ซึ่งเป็นซอร์สโค้ด (logger.c)ของภาษาซีที่คอมไพเลอร์แล้วตั้งชื่อว่า logger บน dom0

5. ขั้นตอนที่ 5 สร้าง domU พร้อมติดตั้งฟีโดร่า (Fedora) เพื่อเป็นระบบปฏิบัติการของ domU และใช้ในการทดลองการเข้าถึงไฟล์ในลักษณะต่างๆ

6. ขั้นตอนที่ 6 สร้างไฟล์ข้อมูลชื่อว่า s.txt ใน domU

7. ขั้นตอนที่ 7 ติดตั้งโปรแกรมจำลองการเข้าถึงไฟล์ข้อมูล (read) ซึ่งเป็นซอร์สโค้ด (read.c) ของภาษาซีที่คอมไพเลอร์แล้วตั้งชื่อว่า read ใน domU เพื่อจำลองการเข้าถึงไฟล์ข้อมูล



ภาพ 6 หน้าต่างการทำงานหลังติดตั้งซอท์ฟแวร์ในขั้นตอนต่าง ๆทั้งหมดสำเร็จ

จากการทำขั้นตอนการติดตั้งทั้ง 6 ขั้นตอน ในขั้นตอนที่ 1 ดำเนินการติดตั้งฟีโดร่า (Fedora) ซึ่งเป็นระบบปฏิบัติการลินุกซ์ บนเครื่องคอมพิวเตอร์ที่เป็นเครื่องที่ทำหน้าที่ให้บริการคลาวด์โดยเครื่องนี้จะมีทรัพยากรไว้ให้บริการมากมายเพื่อจัดสรรให้กับเครื่องคอมพิวเตอร์เสมือนที่ถูกสร้างขึ้นซึ่งจะเรียกว่า dom0 ดังภาพ 6 หลังจากดำเนินการขั้นตอนที่ 1 เสร็จสิ้น จะติดตั้ง Xen ซึ่งเป็น hypervisor ที่มีความสามารถจำลองให้ระบบปฏิบัติการหลายระบบสามารถทำงานพร้อมกันบนเครื่องคอมพิวเตอร์ได้ โดยหน้าที่ของ Xen จะจัดการควบคุมระบบปฏิบัติการแต่ละระบบไม่ให้เกิดการทำงานที่ทับซ้อนกัน Hypervisor ได้ถูกเรียกอีกอย่างว่า Virtual Machine Management (VMM) ทำหน้าที่สร้างและจัดการเครื่องคอมพิวเตอร์เสมือนที่ให้บริการแก่ผู้ใช้บริการ โดยจะทำงานอยู่บน dom0 ดังภาพ 6(1) ในขั้นตอนที่ 3 ทำการติดตั้ง LibVMI ซึ่งเป็นไลบรารีของภาษาซี บน dom0 ที่เป็นสิ่งสำคัญในการทำงานของโปรแกรมบันทึกเหตุการณ์ซึ่ง

ความสามารถของ LibVMI ถูกกล่าวไว้แล้วในหัวข้อเรื่อง LibVMI เมื่อติดตั้ง LibVMI เรียบร้อย สามารถดำเนินการในขั้นตอนที่ 3 เพื่อเขียนซอร์สโค้ดของภาษาซี และทำการคอมไพเลอร์ให้เป็นโปรแกรมบันทึกเหตุการณ์ (logger) บน dom0 ซึ่งจะทำหน้าที่ตรวจสอบข้อมูลในหน่วยความจำหลักของเครื่องคอมพิวเตอร์เสมือนที่ถูกสร้างขึ้นในขั้นตอนต่อไป ในขั้นตอนที่ 5 เมื่อติดตั้งขั้นตอนที่ 1-4 เรียบร้อยแล้วทำการสร้างเครื่องคอมพิวเตอร์เสมือนใน Xen บน dom0 ทำการติดตั้งติดตั้งฟีโดร่า (Fedora) ซึ่งเป็นระบบปฏิบัติการลินุกซ์ เครื่องคอมพิวเตอร์เสมือนที่สร้างขึ้นนี้จะเรียกว่า domU ดังภาพ 6(2) ขั้นตอนที่ 6 ทำการสร้างไฟล์ข้อมูลชื่อว่า s.txt ไว้ใน domU เพื่อใช้เป็นไฟล์ที่ถูกกระทำเหตุการณ์ต่างๆเช่น อ่าน เขียน ลบ และเปลี่ยนแปลงข้อมูลประจำตัวของไฟล์ สุดท้ายในขั้นตอนที่ 7 เขียนซอร์สโค้ด (read.c) ของภาษาซีที่คอมไพเลอร์แล้วตั้งชื่อว่า read ใน domU เพื่อโปรแกรมจำลองการเข้าถึงไฟล์ข้อมูล

การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม

เมื่อดำเนินการในหัวข้อเรื่อง การติดตั้งซอฟท์แวร์ที่ใช้ในการทำลอง เสร็จเรียบร้อยแล้ว ในหัวข้อนี้จะแสดงถึงขั้นตอนการวิธีใช้งานระบบบันทึกเหตุการณ์ และวิธีการวัดประสิทธิภาพของการทำงานของระบบบันทึกเหตุการณ์

1. **วิธีใช้งานระบบบันทึกเหตุการณ์**

ในการทดลอง ผู้วิจัยได้ได้ศึกษาวิธีใช้งานของระบบบันทึกเหตุการณ์ ซึ่งการดำเนินการใช้งาน ผู้วิจัยทำการเรียกใช้งาน โปรแกรมบันทึกเหตุการณ์ (logger) ซึ่งต่อไปนี้จะเรียกว่า ล็อกเกอร์ โดยใช้คำสั่ง ./logger ดังภาพ 7(1) ภายใต้สิทธิ์ของยูสเซอร์ที่ชื่อว่า root หรือเจ้าของระบบ ซึ่งยูสเซอร์นี้มีสิทธิที่สามารถจัดการกับไฟล์ โฟลเดอร์ และโปรแกรมต่างๆบนระบบปฏิบัติการได้ ดังนั้น root จึงทำหน้าที่เป็นยูสเซอร์ที่บริหารจัดการทำงาน dom0 ได้อย่างเต็มรูปแบบ ซึ่งเมื่อเรียกใช้งาน ล็อกเกอร์จะสร้างโปรเซสเพื่อทำหน้าที่ตรวจจับโปรเซส read ซึ่งเกิดจากโปรแกรมจำลองการเข้าถึงไฟล์ข้อมูล ใน domU เมื่อตรวจจับพบจะแสดงข้อมูลที่ได้ ดังภาพ 7(2)

```
① [root@localhost godz]# ./logger
② s.txt, PId: 3759, PName: read, Uid: 1001
```

ภาพ 7 แสดงขั้นตอนการใช้งาน ล็อกเกอร์ และการแสดงผลลัพธ์

ในการทดลองจะใช้โปรแกรมจำลองการเข้าถึงไฟล์ข้อมูลเพื่อเป็นการจำลองการอ่านไฟล์ซึ่งจะทำงานใน domU ซึ่งเรียกใช้งานด้วยคำสั่ง ./read ดังภาพ 8(1) เมื่อเรียกใช้คำสั่งแล้วจะสร้างโปรเซส read ซึ่งโปรเซสนี้จะไปอ่านไฟล์ข้อมูล s.txt และจะแสดงข้อมูลที่อยู่ภายในดังภาพ 8(2)

```
① [godz@localhost Download]# ./read
② Thongrob auxsorn
```

ภาพ 8 แสดงขั้นตอนการใช้งานโปรแกรมจำลองการเข้าถึงไฟล์ข้อมูล

2. วิธีการทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์

สำหรับการทดลองในหัวข้อนี้ผู้วิจัยมีความจำเป็นต้องออกแบบสภาพแวดล้อมในการทดลองเพื่อนำมาใช้ในการทดสอบค่าความถูกต้องของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง (CPU Core) และ ขนาดหน่วยความจำหลัก (RAM) ทั้งในส่วนของผู้ให้บริการและผู้ใช้บริการ เพื่อหาผลกระทบในการทำงานของระบบบันทึกเหตุการณ์

2.1 การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลได้รับอิทธิพลมาจากข้อเสนอแนะของงานวิจัย (Wongthai & Van Moorsel, 2016) ว่าการเพิ่มจำนว นแกนของหน่วยประมวลกลางใน dom0 หรือผู้ให้บริการ นั้น สามารถทำให้การประมวลผลข้อมูลได้เร็วขึ้น ทำให้ค่าความถูกต้องในการตรวจจับของระบบบันทึกเหตุการณ์เพิ่มขึ้นส่งผลให้ประสิทธิภาพในการทำงานเพิ่มขึ้นด้วย และในการเปลี่ยนแปลงลักษณะเดียวกันนี้ ในส่วนของ domU หรือผู้ใช้บริการจะมีผลต่อการทำงานของระบบบันทึกเหตุการณ์บน dom0 หรือไม่อย่างไรโดยในการทดลองผู้วิจัยได้กำหนดแนวทางไว้ดังนี้

2.2 จำนวนแกนของหน่วยประมวลผลกลางในส่วนของ dom0 หรือผู้ให้บริการ ที่ถูกติดตั้งระบบบันทึกเหตุการณ์ การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลกลาง โดยเรียกใช้งานตามคำสั่ง taskset บนระบบปฏิบัติการลินุกซ์ซึ่งเป็นวิธีการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง (Nanni, 2014) และให้ ล็อกเกอร์ ตรวจจับและบันทึกข้อมูลเหตุการณ์ที่เกี่ยวข้องกับไฟล์ข้อมูล s.txt ใน domU โดยตั้งค่าขนาดของ dom0 ให้มีจำนวนแกนของหน่วยประมวลผลกลาง 1 ถึง 8 แกน และตั้งค่าขนาดของ domU ให้มีจำนวนแกนของหน่วยประมวลผลกลาง 1 ถึง 8 แกน เช่นเดียวกัน ในการทดลองได้กำหนดค่า domU ให้มีจำนวนแกนของหน่วยประมวลผล

กลางคงที่ ส่วน dom0 นั้นมีการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง เช่น กำหนดให้ขนาดของ domU มีจำนวนแกนของหน่วยประมวลผลกลาง 1 แกน และกำหนดจำนวนแกนของหน่วยประมวลผลกลางของ dom0 ให้มีจำนวนแกนระหว่าง 1 ถึง 8 แกน หลังจากกำหนดจำนวนแกนของหน่วยประมวลผลกลางแล้ว กำหนดช่วงเวลาของ sleeping time และเริ่มทำงานของ ล็อกเกอร์ เมื่อทำการทดลองปรับจำนวนแกนของหน่วยประมวลผลกลางของ dom0 จนครบ 8 แกน ในขณะที่ domU มีจำนวนแกนของหน่วยประมวลผลกลาง 1 แกน เสร็จแล้ว ก็จะทำการเพิ่มจำนวนแกนของหน่วยประมวลผลกลางของ domU ให้มีจำนวนแกนเพิ่มขึ้นจนถึง 8 แกน ตามลำดับโดยใช้กระบวนการทดลองเดิมจนครบ

2.3 จำนวนแกนของหน่วยประมวลผลกลางในส่วนของ domU หรือผู้ใช้บริการ ที่ถูกติดตั้งโปรแกรมจำลองการเข้าถึงไฟล์ข้อมูล การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลกลางในลักษณะเดียวกันกับ dom0 โดยในการทดลองได้กำหนดค่า dom0 ให้มีจำนวนแกนของหน่วยประมวลผลกลางคงที่ ส่วน domU นั้นมีการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง เช่น กำหนดให้ขนาดของ dom0 มีจำนวนแกนของหน่วยประมวลผลกลาง 1 แกน และกำหนดจำนวนแกนของหน่วยประมวลผลกลางของ domU ให้มีจำนวนแกนระหว่าง 1 ถึง 8 แกน และทำการจำลองการเปิดอ่านไฟล์ข้อมูล s.txt จำนวน 10,000 ครั้งทำซ้ำกันจำนวน 10 รอบ เมื่อทำการทดลองปรับจำนวนแกนของหน่วยประมวลผลกลางของ domU จนครบ 8 แกน ในขณะที่ dom0 มีจำนวนแกนของหน่วยประมวลผลกลาง 1 แกน เสร็จแล้ว ทำการเพิ่มจำนวนแกนของหน่วยประมวลผลกลางของ dom0 ให้มีจำนวนแกนเพิ่มขึ้นจนถึง 8 แกน ตามลำดับโดยใช้กระบวนการทดลองเดิมจนครบ

2.4 การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงขนาดของหน่วยความจำหลัก จากความศึกษาที่ผ่านมาพบว่าขนาดของหน่วยความจำหลักมีผลต่อการทำงานของระบบคอมพิวเตอร์ ดังนั้นผู้วิจัยต้องการทราบว่าหากมีการเปลี่ยนแปลงขนาดของหน่วยความจำหลักจะส่งผลกระทบหรือไม่ อย่างไรกับการทำงานของระบบบันทึกเหตุการณ์ ในการทดลองนี้ทำการเปลี่ยนแปลงขนาดของหน่วยความจำหลักในส่วนของผู้บริการที่มีขนาดของหน่วยความจำหลักทั้งหมด 8GB และผู้ใช้บริการที่เป็นหน่วยความจำเสมือนมีขนาดสูงสุดอยู่ที่ 3 GB ซึ่งเป็นข้อจำกัดในการทดลอง

2.5 ขนาดหน่วยความจำหลักในส่วนของ dom0 หรือผู้ให้บริการ จากจำนวนหน่วยความจำหลักทั้งหมดของ dom0 ที่มีขนาด 8 GB ในการทดลองผู้วิจัยจะทำการเปลี่ยนแปลงขนาดหน่วยความจะหลักของ dom0 ให้มีขนาดเท่ากับ 4GB 6GB และ 8GB ตามลำดับ และ

กำหนดขนาดหน่วยความจำหลักเสมือนของ domU เท่ากับ 1GB เมื่อทำการทดลองปรับขนาดหน่วยความจำหลักของ dom0 จนครบ 3 ค่าแล้ว ในขณะที่ domU มีขนาดหน่วยความจำหลักเสมือนเท่ากับ 1GB เสร็จแล้ว ทำการเพิ่มขนาดหน่วยความจำหลักเสมือนของ domU เป็น 2GB และ 3GB ตามลำดับ โดยใช้กระบวนการทดลองเดิมจนครบ

2.6 ขนาดหน่วยความจำหลักในส่วนของ domU หรือผู้ใช้บริการ จากจำนวนหน่วยความจำหลักเสมือนทั้งหมดของ domU ที่มีขนาดมากที่สุดที่ 3GB ในการทดลองผู้วิจัยจะทำการเปลี่ยนแปลงขนาดหน่วยความจะหลักของ domU ให้มีขนาดเท่ากับ 1GB 2GB และ 3GB ตามลำดับ และกำหนดขนาดหน่วยความจำหลักของ dom0 เท่ากับ 4GB เมื่อทำการทดลองปรับขนาดหน่วยความจำหลักของ domU จนครบ 3 ค่าแล้ว ในขณะที่ dom0 มีขนาดหน่วยความจำหลักเท่ากับ 4GB เสร็จแล้ว ทำการเพิ่มขนาดหน่วยความจำหลักเสมือนของ dom0 เป็น 6GB และ 8GB ตามลำดับ โดยใช้กระบวนการทดลองเดิมจนครบ

การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

เมื่อดำเนินการในหัวข้อ 3.4 เรื่อง การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม ซึ่งประกอบไปด้วย วิธีใช้งานระบบบันทึกเหตุการณ์ และ วิธีการทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ เพื่อหาข้อจำกัดของการทำงานของระบบบันทึกเหตุการณ์ เสร็จเรียบร้อยแล้ว ในหัวข้อนี้จะแสดง 2 หัวข้อย่อย โดยในหัวข้อแรกจะแสดงถึงขั้นตอนการวิธีการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยการศึกษาโครงสร้างของการทำงานของระบบบันทึกเหตุการณ์ ในส่วนของ ขั้นตอนการทำงานของ LibVMI การเขียนคำสั่งโค้ดในภาษา C เพื่อลดข้อจำกัดเกี่ยวกับการตรวจจับข้อมูลของล็อกเกอร์ และการจัดการหน่วยความจำ หัวข้อที่สองจะเป็นการเปรียบเทียบการทำงานและประสิทธิภาพของระบบบันทึกเหตุการณ์ก่อนการปรับปรุงและหลังปรับปรุง

การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

ในการทดลองระบบบันทึกเหตุการณ์เดิมนั้นมีข้อมูลที่ได้จากวิธีการ Introspection ของ LibVMI ได้แก่ ผู้ใช้งานโปรเซส ชื่อไฟล์ หมายเลขของโปรเซส ชื่อโปรเซส และระบุเวลาที่ตรวจสอบได้ ซึ่งพบว่าข้อมูลมีจำนวนน้อยมากในการนำไปเป็นหลักฐานในการพิจารณา ค้นหา ผู้รับผิดชอบต่อการกระทำผิดที่ส่งผลทำให้เกิดความเสียหาย ดังนั้นผู้วิจัยจึงนำเสนอการเพิ่มจำนวนข้อมูลในล็อกไฟล์ โดยใช้ข้อมูลภายในโครงสร้างข้อมูลไอโหนด ที่ได้กล่าวในหัวข้อเรื่อง โครงสร้างข้อมูลไอโหนด ในการศึกษาหัวข้อนี้มีจุดประสงค์เพื่อเพิ่มข้อมูลใน ล็อกไฟล์ ที่จะส่งผลให้การตรวจสอบ

พิจารณา ค้นหา ผู้รับผิดชอบต่อการกระทำผิดที่ส่งผลทำให้เกิดความเสียหาย ได้อย่างมีประสิทธิภาพมากขึ้น วิธีการเพิ่มข้อมูลดังกล่าวนี้ผู้วิจัยต้องทำการศึกษาโครงสร้างข้อมูลสถาปัตยกรรมของระบบปฏิบัติการเพื่อหาที่อยู่ (Address) และการอ้างอิง (Pointer) ของข้อมูลที่ต้องการนำมาใช้งาน เนื่องจากการศึกษาการทำงานของ LibVMI นั้นจะพบว่า LibVMI มีความสามารถดึงข้อมูลภายในโครงสร้างข้อมูลไอโหนด ที่เป็นโครงสร้างข้อมูลซึ่งมีการเก็บข้อมูลเกี่ยวกับข้อมูลที่จำเป็นต่อการเพิ่มจำนวนข้อมูลในล็อกไฟล์ การที่จะนำผลลัพธ์หรือข้อมูลที่อยู่ในโครงสร้างข้อมูลไอโหนดออกมาได้นั้น ต้องรู้ถึงโครงสร้างข้อมูลของระบบปฏิบัติการ การอ้างอิงข้อมูล ที่อยู่ของข้อมูล รวมไปถึงชนิดและขนาดของข้อมูลที่ถูกจัดเก็บไว้ในโรงสร้างข้อมูลไอโหนด

การวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

ในส่วนการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ ในส่วนนี้จะทำการวัดและทดสอบประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้บริการเมื่อมีการใช้งานระบบบันทึกเหตุการณ์ เนื่องจากการทำงานของระบบบันทึกเหตุการณ์มีผลกระทบต่อการทำงานของผู้ใช้บริการในเครื่องคอมพิวเตอร์เสมือน เพราะว่าการทำงานของ LibVMI ที่เป็นเครื่องมือสำคัญของระบบบันทึกเหตุการณ์ที่เข้าไปตรวจสอบหน่วยความจำหลักของเครื่องคอมพิวเตอร์เสมือนนั้นจะส่งผลทำให้การทำงานหรือการประมวลผลของผู้ใช้บริการนั้นใช้เวลามากขึ้น ในการทดลองนี้ผู้วิจัยจะแบ่งออกเป็นสองส่วนคือ ส่วนที่หนึ่งผู้วิจัยจะทำการวัดประสิทธิภาพการทำงานของระบบปฏิบัติการที่ยังไม่มีการติดตั้งระบบบันทึกเหตุการณ์เปรียบเทียบกับระบบปฏิบัติการที่ติดตั้งระบบบันทึกเหตุการณ์โดยใช้โปรแกรม sysbench ส่วนที่สอง ผู้วิจัยจะทำการทดลองปรับช่วงระยะเวลาการทำงานของระบบบันทึกเหตุการณ์ ซึ่งการปรับช่วงเวลานี้ จะมีผลต่อความแม่นยำของการตรวจสอบของระบบบันทึกเหตุการณ์กล่าวคือหากต้องการความแม่นยำสูงแต่ประสิทธิภาพการทำงานจะถูกลดทอนลง ในการทดลองนี้มีจุดมุ่งหมายเพื่อเสนอเป็นแนวทางให้ผู้ใช้บริการ สามารถเลือกประสิทธิภาพการทำงานระหว่างระบบบันทึกเหตุการณ์ที่มีความแม่นยำสูงแต่ประสิทธิภาพในการประมวลของเครื่องคอมพิวเตอร์จะลดลง และในทางกลับกันถ้าเครื่องคอมพิวเตอร์มีประสิทธิภาพในการประมวลผลที่สูง ความแม่นยำของระบบบันทึกเหตุการณ์ก็จะลดลงเช่นเดียวกัน ดังนั้นผู้วิจัยจะทำการวัดประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์และประสิทธิภาพการทำงานของคอมพิวเตอร์ โดยจะเสนอเป็นข้อมูลให้ผู้ใช้บริการเป็นผู้เลือกว่าต้องการใช้บริการว่าต้องการใช้งานระบบบันทึกเหตุการณ์ในลักษณะใดที่เหมาะสมกับการใช้งานของผู้ใช้บริการเอง

สรุปภาพรวมบทที่ 3

ในหัวข้อนี้จะกล่าวถึงภาพรวมทั้งหมดของบทที่ 3 วิธีการดำเนินการวิจัยโดยแบ่งรายละเอียดตามหัวข้อดังนี้

หัวข้อเรื่อง เครื่องมือที่ใช้ในการวิจัย

หัวข้อเรื่อง กรอบวิธีการดำเนินวิจัย

หัวข้อเรื่อง การติดตั้งซอฟท์แวร์ที่ใช้ในการทำลอง

หัวข้อเรื่อง การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม

หัวข้อเรื่อง การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

หัวข้อที่เรื่อง การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

หัวข้อเรื่อง การวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

รายละเอียดของข้อสรุปในแต่ละหัวข้อมี ดังนี้

ในหัวข้อเรื่อง เครื่องมือที่ใช้ในการวิจัยนี้ ผู้อ่านจะได้ทราบเกี่ยวกับฮาร์ดแวร์และซอฟท์แวร์ต่างๆ ที่เกี่ยวข้องในการทดลองของวิทยานิพนธ์นี้โดยฮาร์ดแวร์และซอฟท์แวร์แต่ละหัวข้อได้แบ่งหัวข้อย่อยไว้ 2 หัวข้อด้วยกันคือ 1. ฮาร์ดแวร์และซอฟท์แวร์สำหรับการทดลอง 2. ฮาร์ดแวร์ และซอฟท์แวร์สำหรับการบันทึกผลการทดลองและเขียนวิทยานิพนธ์ ในหัวข้อเรื่องกรอบวิธีการดำเนินงานวิจัยได้อธิบายเกี่ยวกับวิธีการออกแบบการทดลองที่ใช้ในงานวิทยานิพนธ์เล่มนี้ โดยแบ่งเป็น 2 ส่วน คือ วิธีการศึกษาและเตรียมข้อมูลสำหรับการทดลอง และ วิธีการออกแบบการทดลอง ซึ่งในส่วนของวิธีการออกแบบการทดลองนั้นได้แบ่งออกเป็น 4 ส่วนคือ หัวข้อเรื่อง การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม หัวข้อเรื่อง การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ หัวข้อเรื่อง การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ และหัวข้อเรื่อง การวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ ซึ่งผลจากการทดลองในหัวข้อเรื่อง การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ และการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ ทั้ง 4 หัวข้อนี้จะถูกกล่าวไว้ในบทที่ 4. ผลการวิจัย ต่อไป

บทที่ 4

ผลการวิจัย

จากเนื้อหาทั้งหมดของบทที่ 3 เรื่อง วิธีการดำเนินการวิจัย ที่ประกอบด้วยวิธีต่างๆที่ใช้สำหรับกำหนดแนวทางการทดลองและทดสอบ ซึ่งประกอบไปด้วย การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ และ การวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ ซึ่งทั้งหมดที่กล่าวมาข้างต้นมีเป้าหมายเพื่อเพิ่มประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ จากการปฏิบัติทดลองและทดสอบจะได้ผลลัพธ์จากการดำเนินการต่างๆ ผู้วิจัยได้แบ่งออกเป็นหัวข้อดังนี้

1. ผลจากการทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม
2. ผลจากการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์
3. ผลจากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์
4. ผลจากการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์
5. สรุปภาพรวมบทที่ 4

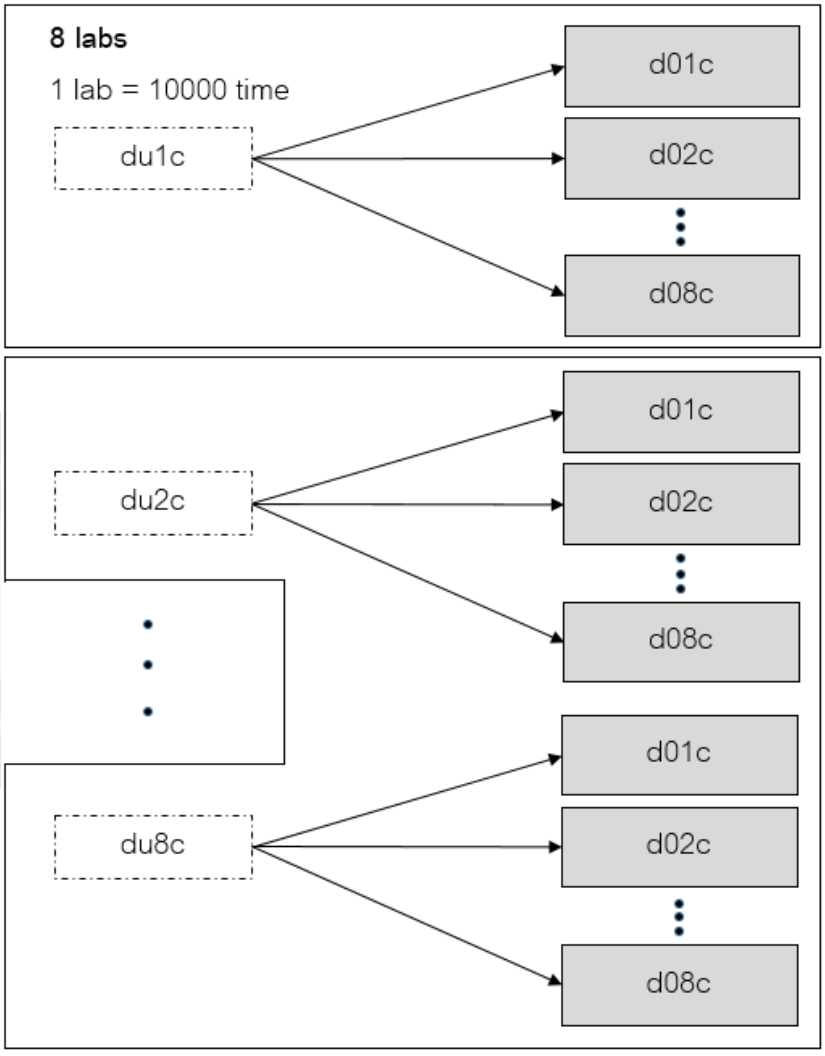
ผลจากการทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม

ในหัวข้อนี้จะนำเสนอเกี่ยวกับผลการทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม มุ่งเน้นการทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ โดยจะแบ่งออกเป็น 3 ส่วน คือการทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงขนาดของหน่วยความจำหลัก และ สรุปปัญหาและข้อจำกัดที่ได้จากการทดลอง

1. การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง

ในการทดสอบนี้จะนำเสนอเกี่ยวกับการทดลองและผลกระทบเกี่ยวกับการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง ที่ส่งผลต่อประสิทธิภาพการทำงานของระบบ

บันทึกเหตุการณ์ โดยแบ่งการทดสอบออกเป็นสองด้านคือ การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการและการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

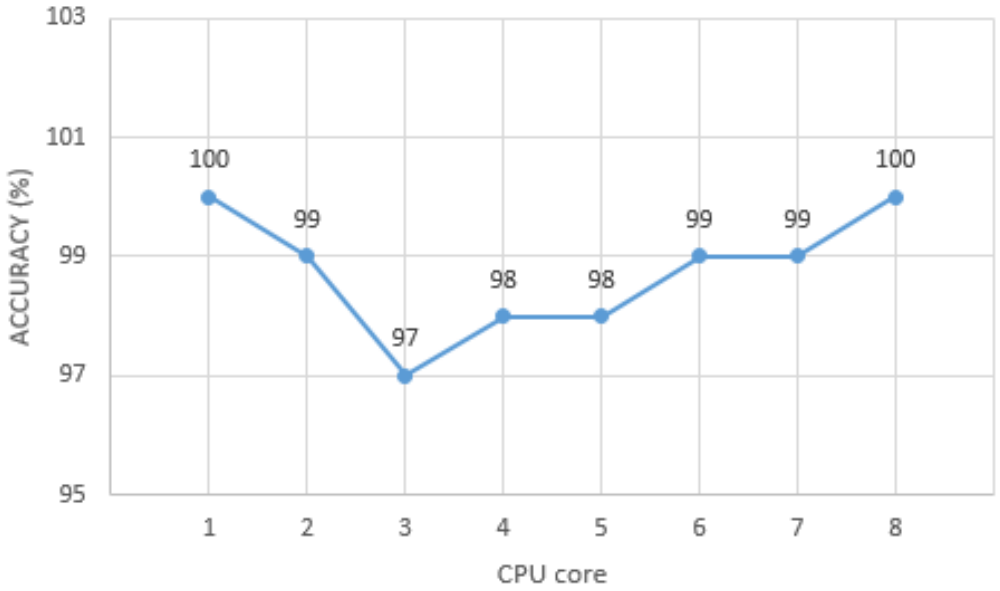


ภาพ 9 การทดลองเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

1. การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการซึ่งในวิทยานิพนธ์เล่มนี้ผู้วิจัยจะเรียก เครื่องคอมพิวเตอร์ของผู้ให้บริการ ว่า dom0

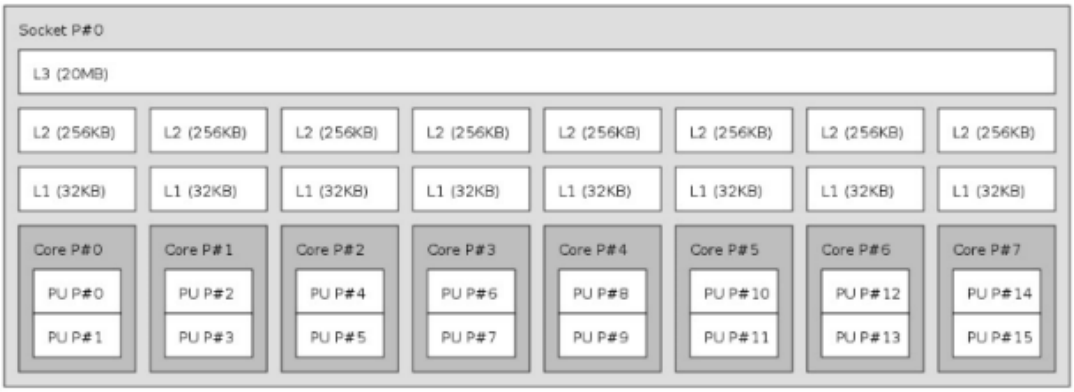
และเครื่องคอมพิวเตอร์ของผู้ใช้บริการ ว่า domU ซึ่งทั้งสองมีจำนวนแกนของหน่วยประมวลผลกลางอยู่ทั้งหมด 8 แกน ในการทดลองผู้วิจัยจะตั้งค่าการทดลอง โดยกำหนดจำนวนแกนของหน่วยประมวลผลกลางของ domU และจำนวนแกนของหน่วยประมวลผลกลางของ domU ดังนี้ เมื่อเริ่มการทดลองจะกำหนดให้ domU เป็นขนาดคงที่ และกำหนดให้ dom0 มีการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางตั้งแต่ 1 – 8 แกน เช่น domU ใช้จำนวนแกนของหน่วยประมลผลกลางเท่ากับ 1 แกน และในส่วนของ dom0 จะเริ่มใช้จำนวนแกนของหน่วยประมลผลกลางเท่ากับ 1 แกน เริ่มทดสอบการทำงานของระบบบันทึกเหตุการณ์ 10 รอบ โดยแต่ละรอบระบบบันทึกเหตุการณ์จะทำการตรวจสอบจำนวน 1000 ครั้ง เมื่อครบแล้วจะเพิ่มจำนวนแกนหน่วยประมวลผลกลางไปจนถึง 8 แกน (du1c-d01c du1c-d02c du1c-d03c du1c-d04c du1c-d05c du1c-d06c du1c-d07c แ ล ะ du1c-d08c) ดังภาพ 9



ภาพ 10 ผลการทดลองเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

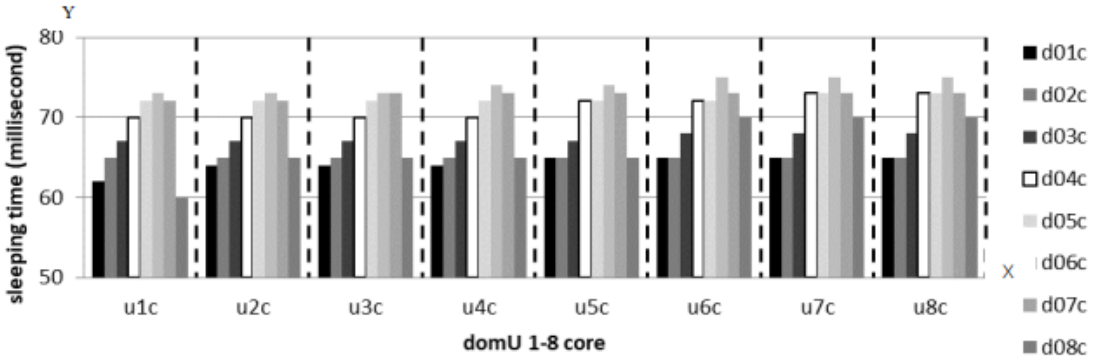
จากภาพ 10 แสดงผลการทดลองเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการ โดยแกน Y จะแสดงความแม่นยำของระบบบันทึกเหตุการณ์ แกน X จะแสดงถึงจำนวนแกนของหน่วยประมวลผลกลาง จากการทดลองพบว่าเมื่อมีการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางนั้น ความแม่นยำของระบบบันทึกเหตุการณ์จะมีผลลัพธ์ที่ใกล้เคียงกัน แต่จะพบว่าในการทำงานของระบบบันทึกเหตุการณ์ที่ใช้

จำนวนแกนของหน่วยประมวลผลกลางที่ 1 และ 8 แกน จะมีผลลัพธ์ที่ได้ 100% ในขณะที่มีการใช้จำนวนแกนของหน่วยประมวลผลกลาง 2 – 7 แกน จะมีความแม่นยำเป็น 99% 97% 98% 98% 99% และ 99% ตามลำดับ ซึ่งความแม่นยำที่แตกต่างกันนี้เกิดจากการทำงานของระบบปฏิบัติการที่มีการจัดสรรพื้นที่สำหรับโพรเซสต่างๆ ได้มีพื้นที่สำหรับการประมวลผลในการทำงานอย่างลงตัวโดย Avudaiyappan & Abdallah (2014) ได้อธิบายไว้ว่า หน่วยประมวลผลกลางแต่ละตัวจะมีแคช (Cache) ซึ่งเป็นหน่วยความจำชนิดหนึ่ง ที่มีความเร็วในการเข้าถึงและโอนถ่ายข้อมูลที่สูง โดยมีหน้าที่ในการเก็บและพักข้อมูลที่ใช้งานบ่อยๆ การเรียกใช้ข้อมูลที่อยู่ในแคชจะมีความรวดเร็วกว่าการเรียกใช้ข้อมูลที่อยู่ในหน่วยความจำหลักจึงส่งผลให้การประมวลผลของหน่วยประมวลผลกลางมีความเร็วมากขึ้น ในหน่วยประมวลผลกลางแต่ละแกนจะมี L1 และ L2 เป็นแคชอยู่ภายในและมี L3 เป็นแคชร่วมสำหรับการทำงานของแต่ละแกนของหน่วยประมวลผลกลางดังภาพ 11 ซึ่งจะแสดงจำนวนแกนของการประมวลผลกลางและแคช L1 L2 และ L3 และอีกส่วนหนึ่งที่แสดงในภาพ 11 คือ Process Unit หรือ PU ตั้งแต่ P#0-15 ซึ่งเป็นจำนวนของเทรด (thread) PU เป็นหน่วยการทำงานย่อยที่อยู่ภายในแกนของการประมวลผลกลางโดยปกติหากมี 1 เทรดจะเรียกว่า single thread หรือ Heavy Weight Process จะพบมากในหน่วยประมวลผลกลางรุ่นเก่า ซึ่งปัจจุบันจะมีการเพิ่มจำนวนเทรดมากกว่า 1 เทรด จะเรียกว่า Multithread หรือ Light Weight Process การที่มีจำนวนเทรดที่มากขึ้นเปรียบเสมือนมีผู้ทำงานมากขึ้นในเวลาเดียวกันส่งผลให้การประมวลผลได้รวดเร็วขึ้นซึ่งถูกนิยามไว้โดย Avudaiyappan (2017) จากข้อมูลนี้ทำให้ผู้วิจัยมีความเห็นว่าเมื่อมีการกำหนดแกนของหน่วยประมวลผลกลาง ให้เป็น 1 แกน โพรเซสล็อกเกอร์ จะทำงานภายในแกนของหน่วยประมวลผลกลางเพียงแกนเดียว ดังนั้นจึงมีการเรียกใช้ข้อมูลที่อยู่ในแคช L1 และ L2 เท่านั้นทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์มีความแม่นยำ เนื่องจากการเรียกใช้ข้อมูลใช้ได้ทันทีจากแคชโดยไม่ต้องสลับการทำงานไปยังแคชของแกนของหน่วยประมวลผลกลางตัวอื่น ในขณะที่หากมีการเพิ่มจำนวนแกนของหน่วยประมวลผลกลาง การจัดสรรของหน่วยประมวลผลกลางของระบบปฏิบัติการจะมีการสลับการประมวลผลไปที่แกนของหน่วยประมวลผลกลางตัวอื่นทำให้ต้องเสียเวลาในการย้ายข้อมูลไปยังแคชของแกนของหน่วยประมวลผลกลางนั้นๆ แต่ทว่าในการทำงานจริงนั้นหากมีโพรเซสจำนวนมากก็อาจจะมีการสลับข้อมูลในการประมวลผลส่งผลทำให้ค่าความแม่นยำลดลง



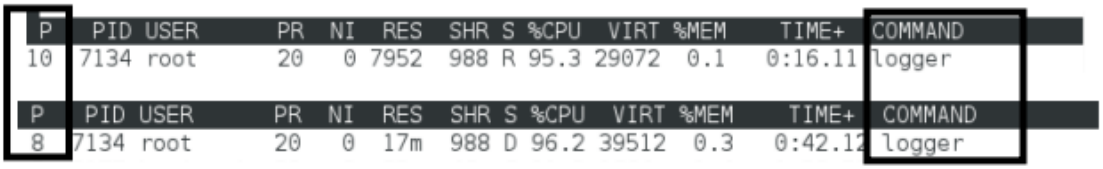
ภาพ 11 สถาปัตยกรรมของหน่วยประมวลผลกลางที่ใช้ในการทดลอง

ในการทดลองการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการนี้ มีส่วนหนึ่งที่สำคัญมากที่ส่งผลต่อประสิทธิภาพของระบบบันทึกเหตุการณ์เกี่ยวกับความแม่นยำในการตรวจสอบคือ sleeping time ซึ่ง sleeping time นี้ คือ ช่วงเวลาที่ระบบบันทึกเหตุการณ์จะทำการหยุดการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้บริการเพื่อตรวจสอบข้อมูลในหน่วยความจำหลักของเครื่องคอมพิวเตอร์ของผู้ใช้ ซึ่งจากผลการทดลองจากภาพ 12 โดยแกน Y ระยะเวลาของ sleeping time มีหน่วยเป็น millisecond แกน X จะแสดงถึงจำนวนแกนของหน่วยประมวลผลกลาง จะพบว่าเมื่อมีการเปลี่ยนแปลงแกนของหน่วยประมวลผลกลางนั้น ระบบบันทึกเหตุการณ์จะมีการใช้เวลาของ sleeping time เพียง 60 millisecond เมื่อแกนของหน่วยประมวลผลกลาง 1 และ 8 แกน โดยเวลาที่น้อยที่สุดอยู่ที่ d08c-du1c และ 62 millisecond เมื่อ d01c-du1c และใช้เวลามากที่สุดที่ 75 millisecond เมื่อ d06c-du6c d06c-du7c และ d06c-du8c ภาพรวมจากกราฟในภาพ 12 จะมีลักษณะคล้ายๆกันคือถ้า dom0 มีแกนของประมวลผลกลาง เท่ากับ 1 และ 8 แกนจะมีการใช้เวลา sleeping time น้อยกว่าจำนวนแกนของประมวลผลส่วนกลางเท่ากับ 2 – 7 แกน ในการทดลองนี้ผู้วิจัยจึงมีแนวคิดว่าระยะเวลา sleeping time ที่ทำให้ระบบบันทึกเหตุการณ์มีประสิทธิภาพเกี่ยวกับความแม่นยำในการตรวจสอบที่ถูกต้องใกล้เคียงหรือเท่ากับ 100% ที่มีค่าเวลาที่น้อยนั้น เกิดจากการประมวลผลที่รวดเร็วของการจัดการของระบบปฏิบัติการที่เกี่ยวกับขั้นตอนการประมวลผลของหน่วยประมวลผลกลาง

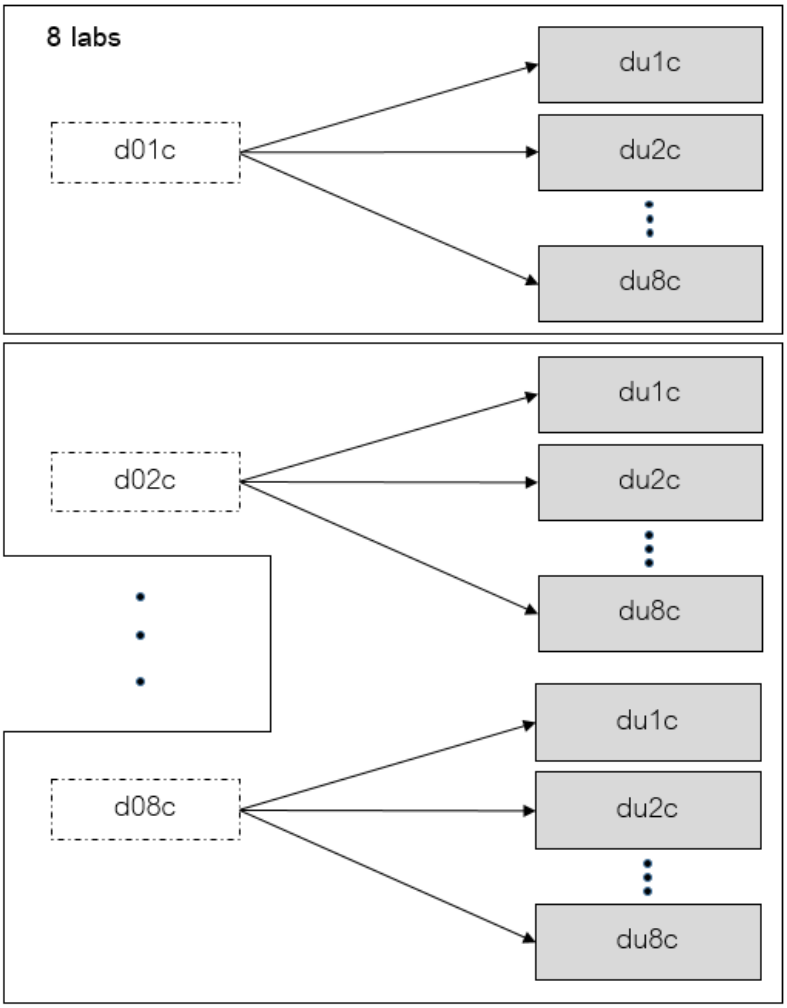


ภาพ 12 sleeping time ของ dom0 จำนวน 1-8 แกน บน domU จำนวน 1-8 แกน

เพื่อให้เข้าใจในเรื่องเกี่ยวกับการทำงานของระบบปฏิบัติการที่มีการสลับการทำงานของแกนของหน่วยประมวลผลกลางจนส่งผลให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลงนั้น สามารถพิสูจน์ข้อมูลที่สันนิฐานนี้ได้จากคำสั่ง top ของระบบปฏิบัติการ (BISWAS, 2017) เพื่อเช็คสถานะโพรเซสของระบบบันทึกเหตุการณ์ที่เกิดขึ้น ดังภาพ 13 ซึ่งจะแสดงให้เห็นว่าโพรเซสของระบบบันทึกเหตุการณ์นั้นมีการสลับการทำงานไปมาของแกนของหน่วยประมวลผลกลาง โดย P หมายถึงหมายเลขของเทรด และ COMMAND หมายถึงชื่อโพรเซสที่เกิดขึ้น โพรเซสที่เกิดขึ้นจากระบบบันทึกเหตุการณ์มีชื่อว่า logger โดยจากภาพที่ 13 จะเห็นว่าการทำงานของระบบบันทึกเหตุการณ์ครั้งแรกจะมีการเรียกใช้เทรดหมายเลข 10 ซึ่งอยู่ในแกนของประมวลผลกลาง แกนที่ 5 มีชื่อโพรเซส ว่า logger และเมื่อมีการทำงานอีกครั้งของระบบบันทึกเหตุการณ์จะมีการสลับจากเทรดหมายเลข 10 ซึ่งอยู่ในแกนของประมวลผลกลาง แกนที่ 5 ไปยังเทรดหมายเลข 8 ซึ่งอยู่ในแกนของประมวลผลกลาง แกนที่ 4 ในชื่อโพรเซส ว่า logger ซึ่งหมายถึงว่าเป็นโพรเซสที่เกิดจากการทำงานของระบบบันทึกเหตุการณ์ และเหตุการณ์ลักษณะนี้จะเกิดขึ้นตลอดเวลาจึงส่งผลทำให้ประสิทธิภาพของระบบบันทึกเหตุการณ์ลดลง เนื่องจากสลับการทำงานของแกนของหน่วยประมวลผลกลางส่งผลมีช่วงเวลา sleeping time ที่เพิ่มมากขึ้น



ภาพ 13 สถานะโพรเซสจากคำสั่ง top ของระบบปฏิบัติการ

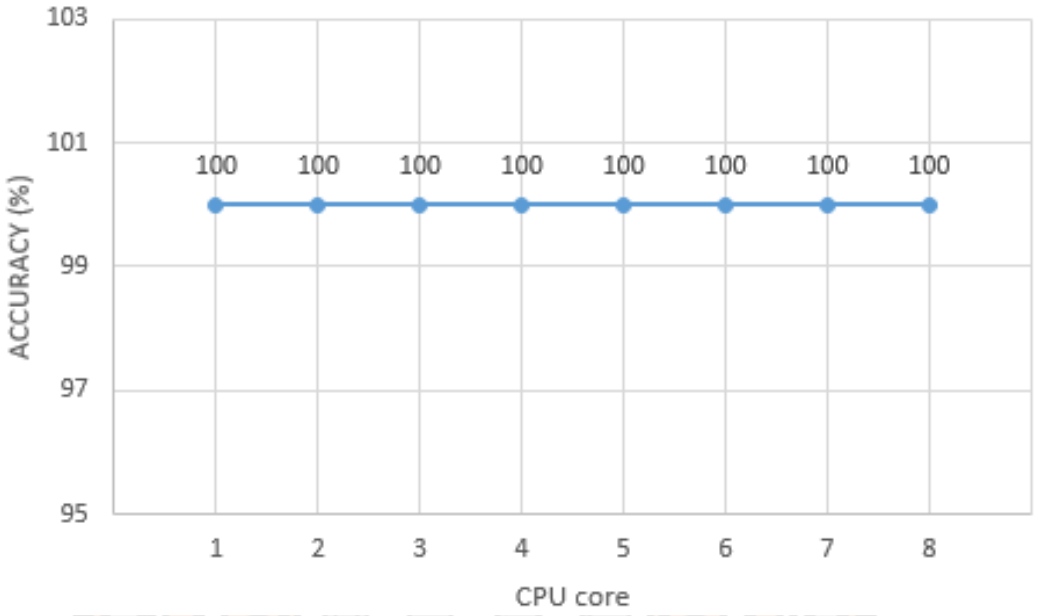


ภาพ 14 การทดลองเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

2. การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

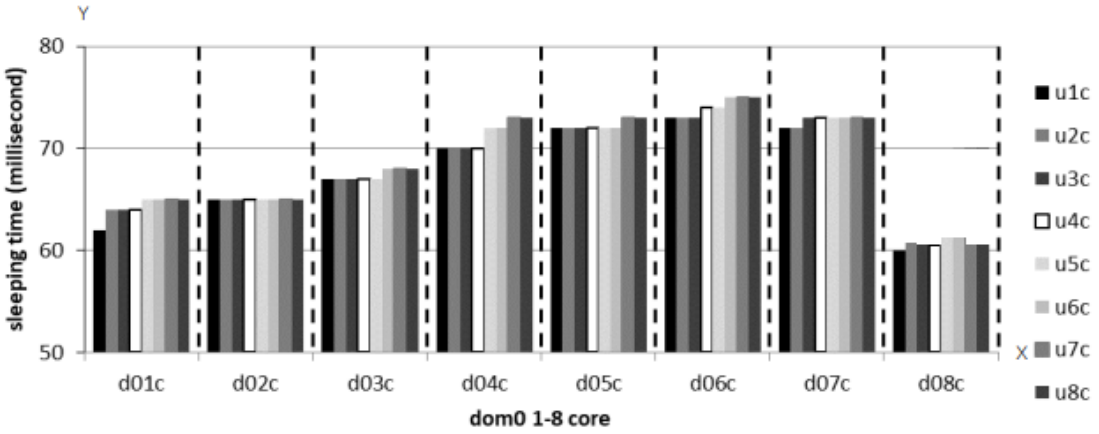
การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ ซึ่งทั้ง dom0 และ domU มีจำนวนแกนของหน่วยประมวลผลกลางอยู่ทั้งหมด 8 แกน ในการทดลองผู้วิจัยจะตั้งค่าการทดลอง โดยกำหนดจำนวนแกนของหน่วยประมวลผลกลางของ domU และจำนวนแกนของหน่วยประมวลผลกลางของ domU ดังนี้ เมื่อเริ่มการทดลองจะกำหนดให้ dom0 เป็นขนาดคงที่ และกำหนดให้ domU มีการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางตั้งแต่ 1 – 8 แกน เช่น dom0 ใช้จำนวนแกนของหน่วยประมลผลกลางเท่ากับ 1 แกน และในส่วนของ domU จะเริ่มใช้จำนวนแกนของหน่วยประมลผลกลางเท่ากับ 1 แกน เริ่ม

ทดสอบการทำงานของระบบบันทึกเหตุการณ์ 10 รอบ โดยแต่ละรอบระบบบันทึกเหตุการณ์จะทำการตรวจสอบจำนวน 1000 ครั้ง เมื่อครบแล้วจะเพิ่มจำนวนแกนหน่วยประมวลผลกลางไปจนถึง 8 แกน (d01c-du1c d01c-du2c d01c-du3c d01c-d04c d01c-du5c d01c-du6c d01c-du7c และ d01c-du8c) ดังภาพ 14



ภาพ 15 ผลการทดลองเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

จากภาพ 15 แสดงผลการทดลองเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการ โดยแกน Y จะแสดงความแม่นยำของระบบบันทึกเหตุการณ์ แกน X จะแสดงถึงจำนวนแกนของหน่วยประมวลผลกลาง จากการทดลองพบว่าเมื่อมีการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางนั้น ไม่มีผลต่อความแม่นยำของระบบบันทึกเหตุการณ์ โดยผู้วิจัยตั้งสมมติฐานไว้ว่าระบบบันทึกเหตุการณ์ทำงานอยู่บน dom0 ไม่ใช่บน domU และในการทำงานของระบบบันทึกเหตุการณ์จะเข้าถึงข้อมูลของหน่วยความจำหลักของเครื่องคอมพิวเตอร์ของผู้ใช้เท่านั้น ดังนั้นการเปลี่ยนแปลงแกนของหน่วยประมวลผลกลางของเครื่องคอมพิวเตอร์ของผู้ใช้บริการจึงไม่มีผลต่อการทำงานของระบบบันทึกเหตุการณ์

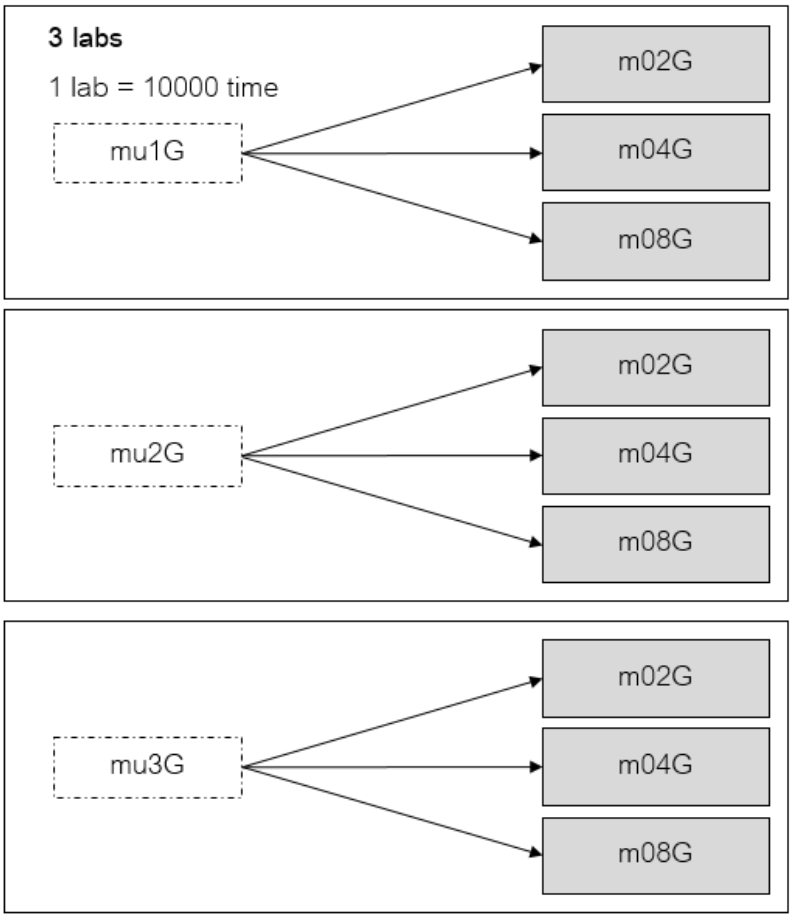


ภาพ 16 sleeping time ของ domU จำนวน 1-8 แกน บน dom0 จำนวน 1-8 แกน

ในการทดลองการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ใช้บริการก็จะพบว่าค่าเวลาของ sleeping time ก็มีความแตกต่างกันเพียงเล็กน้อยดังแสดงในภาพ 16

2. การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงขนาดของหน่วยความจำหลัก

ในการทดสอบนี้จะนำเสนอเกี่ยวกับการทดลองและผลกระทบเกี่ยวกับการเปลี่ยนแปลงขนาดของหน่วยความจำหลัก ที่ส่งผลต่อประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ โดยแบ่งการทดสอบออกเป็นสองด้านคือ การเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการและการเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

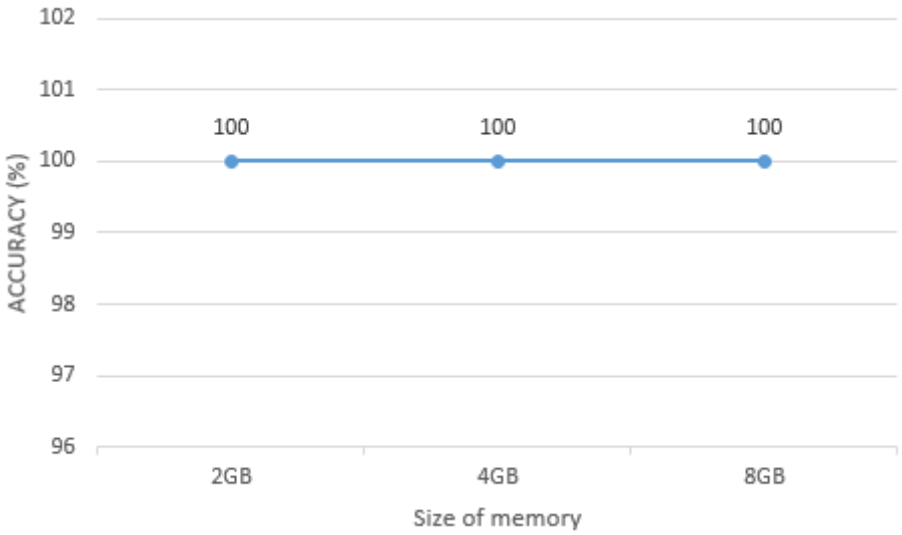


ภาพ 17 การทดลองเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

1. การเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

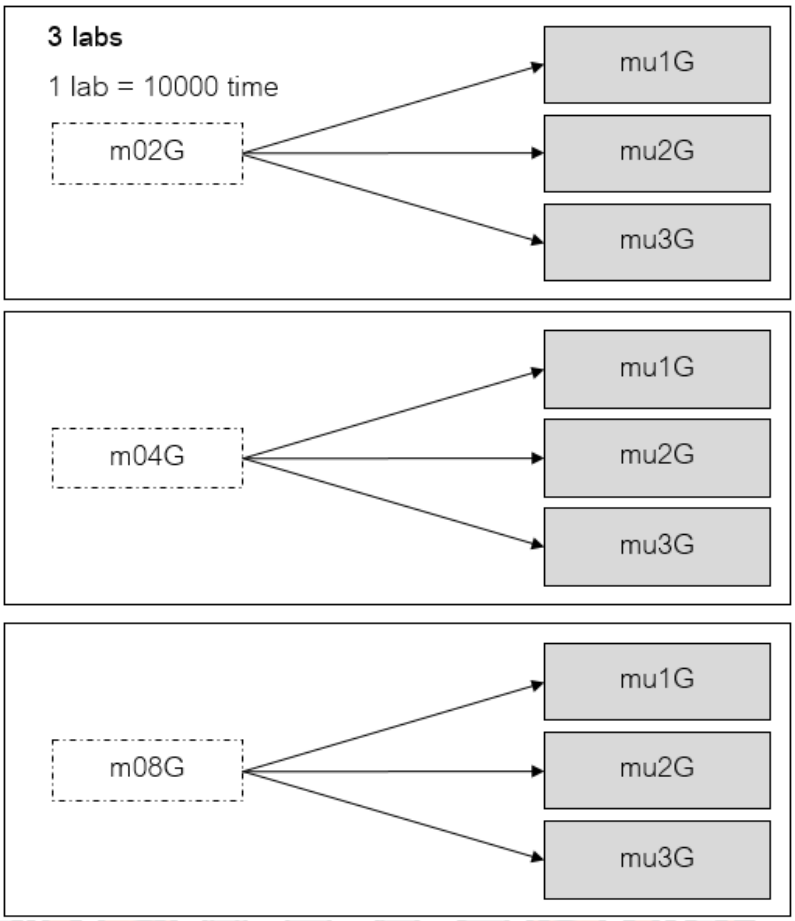
การเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการ ซึ่งการทดลองในวิทยานิพนธ์เล่มนี้ เครื่องคอมพิวเตอร์ของผู้ให้บริการที่ติดตั้งระบบบันทึกเหตุการณ์มีหน่วยความจำหลักทั้งหมด 8 GB ในการทดลองผู้วิจัยจะทำการทดลองด้วยการกำหนดขนาดของหน่วยความจำหลักไว้ 2GB 4GB และ 8GB เพื่อหาผลของการทำงานของระบบบันทึกเหตุการณ์ในหน่วยความจำขนาดที่ต่างกัน เริ่มการทดลองโดยกำหนดขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการ (dom0) และขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ (domU) ดังนี้ เมื่อเริ่มการทดลองจะกำหนดให้หน่วยความจำหลักของ domU เป็นขนาดคงที่ และกำหนดให้หน่วยความจำหลักของ dom0 มีการเปลี่ยนแปลงขนาดของหน่วยความจำหลักตั้งแต่ 2 GB 4GB และ8 GB ตามลำดับ เช่น domU ใช้

ขนาดของหน่วยความจำหลักเท่ากับ 1 GB และในส่วนของ dom0 จะเริ่มใช้ขนาดของหน่วยความจำหลักเท่ากับ 2 GB เริ่มทดสอบการทำงานของระบบบันทึกเหตุการณ์ 10 รอบ โดยแต่ละรอบระบบบันทึกเหตุการณ์จะทำการตรวจสอบจำนวน 1000 ครั้ง เมื่อครบแล้วจะเพิ่มขนาดของหน่วยความจำหลัก 4 GB และ 8 GB ตามลำดับ (mu1G-m02G mu1G-m04G และ mu1G-m08G) ดังภาพ 17



ภาพ 18 ผลการทดลองเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

จากภาพ 18 ผลการทดลองเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการ โดยแกน Y จะแสดงความแม่นยำของระบบบันทึกเหตุการณ์ แกน X จะแสดงถึงขนาดของหน่วยความจำหลักในเครื่องผู้ให้บริการ 2 GB 4 GB และ 8 GB ตามลำดับ จากการทดลองพบว่าเมื่อมีการเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องของผู้ให้บริการนั้น ไม่มีผลต่อความแม่นยำของระบบบันทึกเหตุการณ์ โดยจะเห็นว่าการเพิ่มขนาดหน่วยความจำของเครื่องคอมพิวเตอร์ของผู้ให้บริการจาก 2 GB 4 GB และ 8 GB ค่าความแม่นยำทั้งหมดจะมีค่า 100% จากการสมมติฐานระบบบันทึกเหตุการณ์นั้นมีโพรเซสที่ขนาดเล็ก ดังนั้นไม่ว่าจะมีการเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการอย่างไรจึงไม่ส่งผลต่อประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ และอาจจะเป็นไปได้ว่ายิ่งมีขนาดของหน่วยความจำหลักที่สูงขึ้นจะทำให้ระบบบันทึกเหตุการณ์ทำงานได้ดีขึ้น

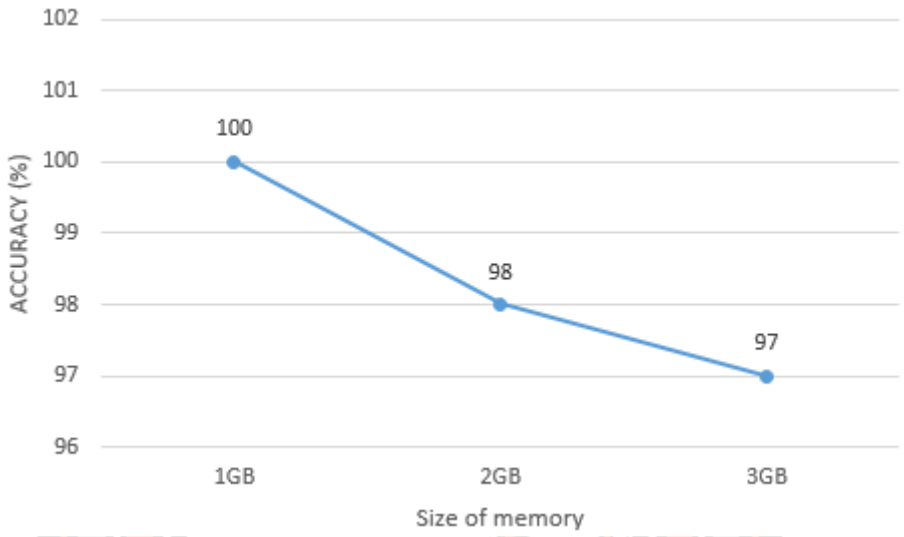


ภาพ 19 การทดลองเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

2. การเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

การเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ ซึ่งการทดลองในวิทยานิพนธ์เล่มนี้ ระบบบันทึกเหตุการณ์จะสามารถตรวจสอบข้อมูลในหน่วยความจำหลักของเครื่องคอมพิวเตอร์ของผู้ใช้งานได้อยู่ 3 ขนาดคือ 1 GB 2 GB และ3 GB ในการทดลองผู้วิจัยจะตั้งค่าการทดลอง โดยกำหนดขนาดของหน่วยความจำหลักของ domU และขนาดของหน่วยความจำหลักของ dom0 ดังนี้ เมื่อเริ่มการทดลองจะกำหนดให้ dom0 เป็นขนาดคงที่ และกำหนดให้ domU มีการเปลี่ยนแปลงขนาดของหน่วยความจำหลักตั้งแต่ 1 – 3 GB เช่น dom0 ใช้ขนาดของหน่วยความจำหลักเท่ากับ 2 GB และในส่วนของ dom0 จะเริ่มใช้ขนาดของหน่วยความจำหลักเท่ากับ 1 GB เริ่มทดสอบการทำงานของระบบบันทึกเหตุการณ์ 10 รอบ โดยแต่ละรอบระบบบันทึกเหตุการณ์จะทำการตรวจสอบจำนวน 1000 ครั้ง เมื่อครบแล้วจะเพิ่มขนาด

ของหน่วยความจำหลักไปจนถึง 3 GB (m02G-mu1G m02G-mu2G และ m02G-mu3G) ดังภาพ 19



ภาพ 20 ผลการทดลองเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

จากภาพ 20 ผลการทดลองเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการโดยแกน Y จะแสดงความแม่นยำของระบบบันทึกเหตุการณ์ แกน X จะแสดงถึงขนาดของหน่วยความจำหลัก จากการทดลองพบว่าเมื่อมีการเปลี่ยนแปลงขนาดของหน่วยความจำหลักนั้น มีผลต่อความแม่นยำของระบบบันทึกเหตุการณ์ โดยจะเห็นว่าการเพิ่มขนาดหน่วยความจำของเครื่องคอมพิวเตอร์ของผู้ใช้บริการจาก 1 GB ไปจนถึง 3 GB ค่าความแม่นยำจะลดลงตามลำดับ 100% 98% และ 97% ผู้วิจัยตั้งสมมติฐานไว้ว่าการเพิ่มขนาดหน่วยความจำของเครื่องคอมพิวเตอร์ของผู้ใช้บริการจะเป็นการขยายพื้นที่ที่ระบบบันทึกเหตุการณ์นั้นต้องตรวจสอบ ดังนั้นจึงทำให้มีการใช้ระยะเวลาในการตรวจสอบมากขึ้นและเป็นไปได้ว่าหากมีการเพิ่มขนาดหน่วยความจำของเครื่องคอมพิวเตอร์ของผู้ใช้บริการที่มีขนาดใหญ่ขึ้นความแม่นยำก็จะลดลงไป

3. **สรุปข้อจำกัดและปัญหาที่ได้จากการทดลอง**

จากการทดลองและผลการทดลองในหัวข้อเรื่อง การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง และ การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงขนาดของ

หน่วยความจำหลักนั้น ผู้วิจัยได้พบข้อจำกัดและปัญหาของระบบบันทึกเหตุการณ์จากการทดลองดังนี้

1. การทำงานของระบบบันทึกเหตุการณ์นั้นจะต้องทำการตลอดเวลาเพื่อทำการตรวจสอบบันทึกเหตุการณ์ที่เกิดขึ้นต่อการกระทำต่างๆของผู้ใช้งาน ในระบบบันทึกเหตุการณ์แบบดั้งเดิมนี้จะมีการออกแบบการทำงานด้วยการเรียกใช้งานระบบบันทึกเหตุการณ์ 1 ครั้ง จะทำการตรวจสอบจนกระทั่งพบข้อมูลที่ต้องการ เมื่อพบแล้วจะจบการทำงาน ในขณะที่ระบบบันทึกเหตุการณ์จบการทำงานจะมีโปรแกรมคำสั่งไปกระตุ้นเพื่อเปิดการใช้งานระบบเหตุการณ์ 1 ครั้ง ทำลักษณะนี้วนไปเรื่อยๆ ซึ่งการทำลักษณะนี้มีผลต่อการตรวจสอบ ที่ทำให้ความแม่นยำลดลง เนื่องจากว่าหากในเวลาที่มีการเริ่มการทำงานระบบบันทึกเหตุการณ์ใหม่ ระบบปฏิบัติการต้องมีการจัดเตรียมทรัพยากรต่างๆเพื่อทำให้พร้อมต่อการทำงาน ซึ่งในช่วงเวลานี้เองหากมีการกระทำใดๆ การกระทำนั้นๆจะไม่ได้ถูกตรวจสอบและบันทึกไว้

2. ช่วงเวลาในการหยุดหรือ sleeping time ที่ใช้สำหรับการตรวจสอบข้อมูลในหน่วยความจำหลักของเครื่องคอมพิวเตอร์ของผู้ใช้งาน ซึ่งผู้วิจัยมีแนวคิดว่าถ้าหากว่าเวลาของ sleeping time นี้ มีค่าน้อยลง หรือ สามารถปรับเปลี่ยนได้จะสามารถทำให้ระบบบันทึกเหตุการณ์นั้นมีประสิทธิภาพมากขึ้น

3. การจัดการทรัพยากรหน่วยความจำหลักที่ดีไม่เพียงพอ ในการทดลองการทำงานของระบบบันทึกเหตุการณ์ หากผู้วิจัยทดลองใช้งานเป็นระยะเวลานานจะพบว่าหน่วยความจำหลักของคอมพิวเตอร์ของผู้ให้บริการจะถูกใช้งานในปริมาณมากขึ้นเรื่อยๆ จนทำให้หน่วยความจำหลักไม่เพียงพอต่อการทำงาน เครื่องคอมพิวเตอร์ของผู้ใช้บริการมีความเร็วในการทำงานภาพรวมลดลง และค้างหรือหยุดทำงานในที่สุด ทั้งที่ในความเป็นจริงระบบบันทึกเหตุการณ์เป็นโปรแกรมที่มีการใช้ทรัพยากรที่น้อยมาก และแทบจะไม่มีส่งผลต่อการทำงานภาพรวมของเครื่องคอมพิวเตอร์ของผู้ให้บริการเลย

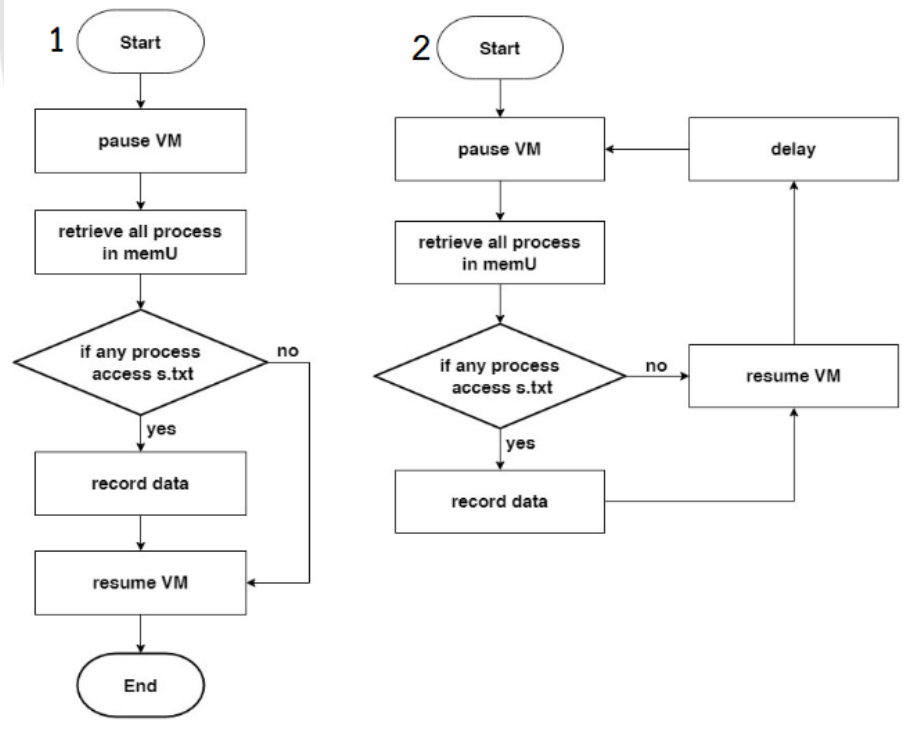
4. ข้อมูลเหตุการณ์ หรือ log file ที่ได้รับจากการทำงานของระบบบันทึกเหตุการณ์นั้น ผู้วิจัยมองว่ายังน้อยเกินไปมากหากต้องการนำไปใช้ในการตรวจสอบหาผู้รับผิดชอบต่อการกระทำ เมื่อต้องการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคาม ตามรายงานของ CSA เรื่องการละเมิดข้อมูล (Data breaches) จะมีด้วยกัน 4 รูปแบบ คือ การดูข้อมูล (Viewed) การเปิดเผยข้อมูล (Release) การขโมยข้อมูล (Stolen) และ การใช้งานโดยไม่ได้รับอนุญาต (Used)

ผลจากการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

จากหัวข้อเรื่อง สรุปข้อจำกัดและปัญหาที่ได้จากการทดลอง ผู้วิจัยได้ศึกษาวิธีการแนวทาง แก้ไขและปรับปรุงข้อจำกัดและปัญหาที่เกิดขึ้นจากการทดลองใช้งานระบบบันทึกเหตุการณ์ ซึ่งในหัวข้อนี้จะดำเนินแก้ไขปรับปรุงในส่วนของขั้นตอนการทำงานของระบบบันทึกเหตุการณ์ การจัดการหน่วยความจำในการทำงานของระบบบันทึกเหตุการณ์ และ การจัดการเวลา sleeping time ที่เป็นระยะเวลาที่มีผลต่อประสิทธิภาพการทำงานของของระบบบันทึกเหตุการณ์ โดยผู้วิจัยได้ทำการออกแบบขั้นตอนและกระบวนทำงานของของระบบบันทึกเหตุการณ์ใหม่

1. **การออกแบบขั้นตอนและกระบวนการทำงานของระบบบันทึกเหตุการณ์**

ผู้วิจัยได้ออกแบบขั้นตอนและกระบวนการทำงานที่ได้ศึกษาจากขั้นตอนกระบวนการของระบบบันทึกเหตุการณ์แบบเดิม โดยการปรับปรุงขั้นตอนและกระบวนการทำงานที่ลดเวลาการทำงานของหน่วยประมวลผลกลางที่ไม่มีประโยชน์ (Overhead) เมื่อมีการใช้งานระบบบันทึกเหตุการณ์ อีกทั้งยังได้เพิ่มส่วนการจัดการเวลาของ sleeping time ดังภาพ 21

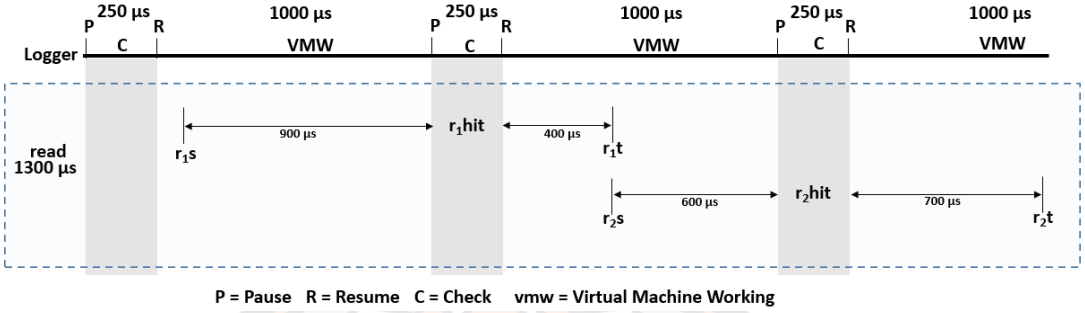


ภาพ 21 กระบวนการทำงานของระบบบันทึกเหตุการณ์แบบเดิมและใหม่

จากภาพ 21(1) แสดงกระบวนการทำงานของระบบบันทึกเหตุการณ์แบบเดิมที่ยังไม่ได้ปรับปรุงกระบวนการทำงานจะพบว่าในกระบวนการทำงานของระบบบันทึกเหตุการณ์แบบเดิมนั้นจะต้องมีเปิดโปรแกรมทุกๆครั้งเมื่อตรวจสอบพบเหตุการณ์ทำให้เกิดเวลาการทำงานของหน่วยประมวลผลกลางที่ไม่มีประโยชน์ (Overhead) ซึ่งเวลาดังกล่าวเป็นเวลาในการรอคอยการใช้ทรัพยากรต่างๆของระบบปฏิบัติการ เช่น การจัดสรรพื้นที่ในหน่วยความจำหลัก การอ่านข้อมูลในหน่วยความจำหลัก การโหลดข้อมูลเพื่อประมวลผล หากว่าสามารถทำให้เวลาการทำงานของหน่วยประมวลผลกลางที่ไม่มีประโยชน์นี้ลดน้อยลงได้จะส่งผลให้กระบวนการทำงานของระบบบันทึกเหตุการณ์จะมีความรวดเร็วมากขึ้น โดยในวิทยานิพนธ์นี้จะทำให้ระบบบันทึกเหตุการณ์นั้นทำงานเป็น Background process เพื่อลดเวลาการทำงานของหน่วยประมวลผลกลางที่ไม่มีประโยชน์

การออกแบบกระบวนการการทำงานของระบบบันทึกเหตุการณ์ใหม่ที่ผู้วิจัยได้ทำการปรับปรุงให้สามารถมีกระบวนการทำงานที่รวดเร็ว และมีสามารถกำหนดเวลาของ sleeping time ดังภาพ 21(2) ซึ่งจะส่งผลทำให้ประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์เกี่ยวกับการตรวจสอบข้อมูลในหน่วยความจำหลักมีความแม่นยำมากขึ้น โดยกระบวนการทำงานจากภาพ 21(2) เมื่อเริ่มการทำงานระบบบันทึกเหตุการณ์ (start) จะเกิดโพรเซสที่ชื่อว่า ล็อกเกอร์ ซึ่งโพรเซสนี้จะเป็นโพรเซสที่ทำงานตลอดเวลาเพื่อใช้ในการตรวจสอบข้อมูล วงจรการทำงานนี้จะเริ่มจากการหยุดการทำงานช่วงสั้นๆเป็นระยะเวลา 250 microsecond (pause) ในช่วงเวลานี้จะมีการใช้เทคนิควิธีเพื่อทำการอ่านข้อมูลในหน่วยความจำหลักของเครื่องคอมพิวเตอร์ของผู้ใช้บริการโดย ไลบรารี libVMI (Process list) เมื่อได้ข้อมูลแล้วจะทำการตรวจสอบข้อมูลโดยการเปรียบเทียบ (Check process()) กับข้อมูลที่กำหนดไว้หากข้อมูลตรงกันจะมีการบันทึกข้อมูลเก็บไว้ (record data) หรือหากว่าไม่ตรงตามเงื่อนไขที่กำหนดไว้จะเริ่มการทำงานใหม่ (resume VM) หลังจากนั้นจะมีการหยุดรอเวลาตามกำหนดที่เวลาที่ได้กำหนดไว้ (delay) ซึ่งเป็นค่า sleeping time เช่น 1000 microsecond ก่อนที่จะเริ่มการทำงานรอบใหม่ ซึ่งภาพ 22 คือตัวอย่างการทำงานของระบบบันทึกเหตุการณ์ ดังนั้นกระบวนการทำงานของระบบบันทึกเหตุการณ์ใหม่นี้ผู้วิจัยได้มีการสร้างกระบวนการที่สามารถทำงานได้อย่างต่อเนื่องโดยไม่มีการสิ้นสุดการทำงาน กล่าวคือสามารถทำงานได้ตลอดเวลาจนกว่าผู้ให้บริการต้องการหยุดการทำงานภายในการเปิดการทำงานระบบบันทึกเหตุการณ์ครั้งเดียว ซึ่งมีความแตกต่างจากระบบบันทึกเหตุการณ์เดิมที่

ทุกๆ ครั้งที่ตรวจสอบสำเร็จจะหยุดการทำงานและเรียกใช้งานระบบบันทึกเหตุการณ์ใหม่ตลอดเวลา



ภาพ 22 ตัวอย่างการทำงานของระบบบันทึกเหตุการณ์

การทำงานของระบบบันทึกเหตุการณ์ใหม่นั้นจะเห็นได้ว่ามีกระบวนการทำงานที่รวดเร็วขึ้นเมื่อเปรียบเทียบกับระบบบันทึกเหตุการณ์เดิมอีกทั้งยังมีความสามารถกำหนดระยะเวลา sleeping time ซึ่งค่า sleeping time นี้ หากมีค่าน้อยเกินไปจะส่งผลทำให้ประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้บริการลดลง ในทางกลับกันหากมีค่ามากจะส่งผลทำให้ความแม่นยำของระบบบันทึกเหตุการณ์ลดลง ส่วนในเรื่องของการจัดการหน่วยความจำนั้นระบบบันทึกเหตุการณ์ใหม่ไม่ได้มีการเรียกใช้งานใหม่ตลอดเวลาแต่เป็นการทำงานแบบต่อเนื่องจึงส่งผลให้การใช้หน่วยความจำหลักนั้นไม่มีการเติบโตเพราะว่ามีการเรียกใช้งานหน่วยความจำหลักครั้งเดียวเมื่อเริ่มทำงานระบบบันทึกเหตุการณ์

2. **สรุปผลจากการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์**

จากหัวข้อเรื่อง การออกแบบขั้นตอนและกระบวนการทำงานของระบบบันทึกเหตุการณ์ ผู้วิจัยได้การออกแบบขั้นตอนและกระบวนการทำงานของระบบบันทึกเหตุการณ์ใหม่เพื่อแก้ไขและปรับปรุงข้อจำกัดและปัญหาที่เกิดขึ้นในหัวข้อเรื่อง สรุปข้อจำกัดและปัญหาที่ได้จากการทดลอง เรื่องของขั้นตอนการทำงานของระบบบันทึกเหตุการณ์ การจัดการหน่วยความจำในการทำงานของระบบบันทึกเหตุการณ์ และ การจัดการเวลา sleeping time ที่เป็นระยะเวลาที่มีผลต่อประสิทธิภาพการทำงานของของระบบบันทึกเหตุการณ์ แต่ทว่าในเรื่องของข้อมูลเหตุการณ์หรือ log file ที่ได้รับจากการทำงานของระบบบันทึกเหตุการณ์นั้นยังไม่เพียงพอหรือยังน้อยเกินไปหากต้องการนำไปใช้ในการตรวจสอบหาผู้รับผิดชอบต่อการกระทำ เมื่อต้องการบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคาม ตามรายงานของ CSA เรื่องการละเมิดข้อมูล (Data breaches) จะมี

ด้วยกัน 4 รูปแบบ คือ การดูข้อมูล (Viewed) การเปิดเผยข้อมูล (Release) การขโมยข้อมูล (Stolen) และ การใช้งานโดยไม่ได้รับอนุญาต (Used) โดยในการปรับปรุงนี้จะถูกกล่าวถึงในหัวข้อ 4.3 จากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ ซึ่งเป็นหัวข้อถัดไป

ผลจากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

จากหัวข้อเรื่อง สรุปข้อจำกัดและปัญหาที่ได้จากการทดลอง ผู้วิจัยได้ศึกษาวิธีการ แนวทาง แก้ไขและปรับปรุงข้อจำกัดและปัญหาที่เกิดขึ้นจากการทดลองใช้งานระบบบันทึกเหตุการณ์ ในเรื่องของข้อมูลเหตุการณ์ หรือ log file ที่ได้รับจากการทำงานของระบบบันทึกเหตุการณ์นั้นยังไม่เพียงพอหรือยังน้อยเกินไป ซึ่งผู้วิจัยได้ศึกษาสถาปัตยกรรม การทำงานของระบบปฏิบัติการ โครงสร้างข้อมูลของระบบปฏิบัติการลินุกซ์เพื่อต้องการหาข้อมูลที่ต้องการที่จะนำมาใช้เป็นข้อมูลในการพิสูจน์ ค้นหา ตรวจสอบ เพื่อหาเหตุการณ์ที่ผิดปกติและผู้รับผิดชอบต่อการกระทำในเหตุการณ์ที่เกิดขึ้น

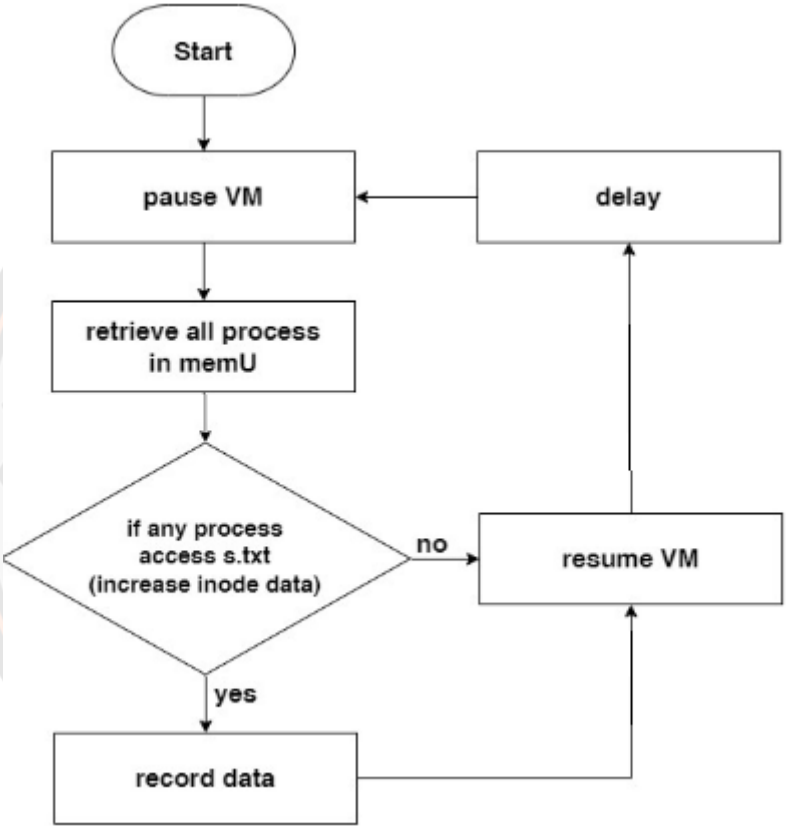
```
struct inode {
        umode_t                 i_mode;
        unsigned short          i_opflags;
        kuid_t                  i_uid;
        kgid_t                  i_gid;
        unsigned int            i_flags;

#ifdef CONFIG_FS_POSIX_ACL
        struct posix_acl        *i_acl;
        struct posix_acl        *i_default_acl;
#endif

        const struct inode_operations   *i_op;
        struct super_block      *i_sb;
        struct address_space    *i_mapping;

#ifdef CONFIG_SECURITY
        void                    *i_security;
#endif

        /* Stat data, not accessed from path walking */
        unsigned long           i_ino;
        /*
         * Filesystems may only read i_nlink directly.  They shall use the
         * following functions for modification:
         *
         *    (set|clear|inc|drop)_nlink
         *    inode_(inc|dec)_link_count
         */
        union {
                const unsigned int i_nlink;
                unsigned int __i_nlink;
        };
        dev_t                   i_rdev;
        loff_t                  i_size;
        struct timespec         i_atime;
        struct timespec         i_mtime;
        struct timespec         i_ctime;
        spinlock_t              i_lock; /* i_blocks, i_bytes, maybe i_size */
        unsigned short          i_bytes;
        unsigned int            i_blkbits;
        blkcnt_t                i_blocks;
```

ภาพ 23 โครงสร้างข้อมูล inode บนระบบปฏิบัติการลินุกซ์

1. **การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์**

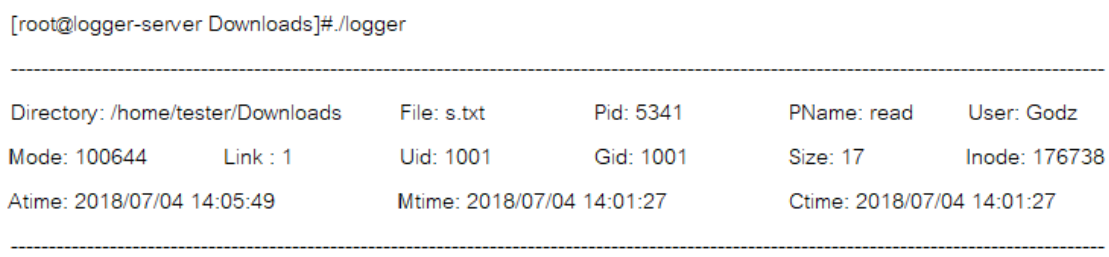
จากการศึกษาผู้วิจัยพบว่าข้อมูลที่สามารถนำมาใช้เพื่อเพื่อหาเหตุการณ์ที่ผิดปกติและผู้รับผิดชอบต่อการกระทำในเหตุการณ์ที่เกิดขึ้นได้นั้นอยู่ในโครงสร้างข้อมูลชื่อว่าไอโหนด (inode structure) ดังภาพ 23 โดยข้อมูลที่นำมาใช้ประกอบไปด้วย i_size i_mode _uid i_gid

i_ino i_nlink i_atime i_mtime และ i_ctime ซึ่งจากเดิมนั้นมีข้อมูลในข้อมูลเหตุการณ์ คือ loginuid pid files path และ Process name หากมีการเพิ่มข้อมูลทั้ง 9 ข้อมูลของโครงสร้างข้อมูลไอโหนดจะทำให้ครอบคลุมในการพิจารณา เรื่องการละเมิดข้อมูล (Data breaches) คือ การดูข้อมูล (Viewed) การเปิดเผยข้อมูล (Release) การขโมยข้อมูล Stolen) การใช้งานโดยไม่ได้รับอนุญาต (Used) และ รวมไปถึงการลบข้อมูล (Remove)



ภาพ 24 กระบวนการทำงานของระบบบันทึกเหตุการณ์ที่เพิ่มข้อมูลจากโครงสร้างข้อมูลไอโหนด

จากภาพ 24 แสดงถึงกระบวนการการทำงานของระบบบันทึกเหตุการณ์ใหม่ที่เพิ่มข้อมูลของโครงสร้างข้อมูลไอโหนด ผู้วิจัยได้ทำการปรับปรุงโดยการเพิ่มกระบวนการทำงานที่อยู่ในส่วนของ (Check process()) ซึ่งจะทำให้มีข้อมูลของระบบบันทึกเหตุการณ์เพิ่มขึ้นจากเดิมรวมทั้งหมดเป็น 14 ข้อมูลที่ประกอบไปด้วย Directory File Pid Pname User Mode Link Uid Gid Size Inode Atime Mtime และ Ctime ดังภาพ 25 แสดงผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์



ภาพ 25 ข้อมูลเหตุการณ์ที่มีการเพิ่มข้อมูลโครงสร้างข้อมูลไอโหนด

2. ผลจากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

ผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ที่มีการปรับปรุงในส่วนของการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนดนั้นจะเห็นได้ว่ามีข้อมูลทั้งหมด 14 ข้อมูลซึ่งแต่ละข้อมูลสามารถนำไปใช้ในการพิจารณาเหตุการณ์ที่เกิดขึ้นจากการกระทำของผู้ใช้งานคอมพิวเตอร์ของผู้ใช้บริการดังนี้ จากภาพ 25 ข้อมูล Directory จะแสดงข้อมูลพาธ (path) หรือเส้นทางที่บ่งบอกพื้นที่ที่ใช้เก็บข้อมูลที่ต้องการตรวจสอบในการทดลองนี้คือไฟล์ข้อมูล s.txt ที่ถูกแสดงในส่วนข้อมูล File สมมติเหตุการณ์ว่าไฟล์ข้อมูล s.txt เป็นไฟล์สำคัญที่มีเพียงไฟล์เดียวในเครื่องคอมพิวเตอร์ของผู้ใช้บริการหากมีเส้นทางของไฟล์นี้เกิดการเปลี่ยนแปลงก็แสดงว่าอาจเกิดเหตุการณ์ที่ผิดปกติขึ้น ข้อมูล Pid ข้อมูลนี้คือหมายเลขโพรเซสของโพรเซสที่ถูกสร้างขึ้นมาเพื่อใช้ในการประมวลผลโดยข้อมูลนี้จะไม่มีวันซ้ำกันและมีจำนวนมากขึ้นเรื่อย ๆ จนกว่าจะมีการรีบูทระบบปฏิบัติการใหม่ ซึ่งข้อมูล Pid นี้สามารถนำไปใช้ในการตรวจสอบความผิดปกติของโพรเซสที่เกิดขึ้นเมื่อต้องการตรวจสอบจะต้องนำข้อมูลนี้ไปค้นหาข้อมูลใน System log ของระบบปฏิบัติการ ข้อมูล Pname เป็นข้อมูลที่แสดงชื่อของโพรเซสที่เรียกใช้ เข้าถึง หรือเปลี่ยนแปลงไฟล์ข้อมูล s.txt โดยทั่วไปชื่อของโพรเซสนี้จะเป็นชื่อเดียวกันกับ ซอฟท์แวร์ หรือ โปรแกรม ดังนั้นในส่วนของข้อมูล Pname สามารถนำมาใช้ตรวจสอบความผิดปกติของการใช้งานของผู้ใช้งานคอมพิวเตอร์ของผู้ใช้บริการเช่น โดยปกติผู้ใช้งานจะใช้ ซอฟท์แวร์ หรือ โปรแกรม ชื่อว่า gedit ในการเข้าถึงข้อมูล แต่ข้อมูลจากไฟล์เหตุการณ์ที่ได้บันทึกไว้กับเป็นโปรแกรมตัวอื่น ข้อมูล User แสดงถึงชื่อผู้ใช้งานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการที่กระทำเหตุการณ์ที่เกี่ยวข้องกับไฟล์ข้อมูลสำคัญ ซึ่งข้อมูลนี้จะเป็นข้อมูลที่สำคัญมากในการหาผู้รับผิดชอบต่อการกระทำเหตุการณ์ที่เกิดขึ้น

ข้อมูล Mode เป็นการกำหนดสิทธิของการใช้งานไฟล์ข้อมูลเช่น –RWXRWXRWX โดย R มีความหมายว่าสามารถเข้าดูข้อมูลได้ Wมีความหมายว่าสามารถเขียนหรือเปลี่ยนแปลงข้อมูลได้ X มีความหมายว่าสามารถ execute ได้ และยังสามารถแบ่งเป็นสามกลุ่มด้วยกันคือ

สามตัวแรกมีหมายถึงเจ้าของไฟล์ สามตัวถัดมาหมายถึงกลุ่มของผู้ใช้ และสามตัวสุดท้ายคือผู้ใช้ทุกคน โดยข้อมูลในส่วนนี้หามีการเปลี่ยนแปลงจะส่งผลทำให้ผู้ใช้งานกลุ่มอื่นๆมีสิทธิในการเข้าถึง ซึ่งทำให้เกิดความไม่ปลอดภัยต่อไฟล์ข้อมูลที่สำคัญ ข้อมูล Link เป็นข้อมูลที่มีลักษณะของการสำรองไฟล์ข้อมูลไว้อีกทีหนึ่งเพื่อป้องกันการสูญหายหรือไม่ต้องการให้ผู้ใช้เข้าถึงไฟล์ได้โดยตรง โดยปกติข้อมูลจะมีค่าเริ่มต้นเท่ากับ 1 หากมีการสำรองไฟล์ข้อมูลจะโดยคำสั่ง ln ข้อมูลจะเปลี่ยนเป็น 2 หรือตามจำนวนการสำรองไฟล์ที่เกิดขึ้นบวกหนึ่ง เช่น ผู้ใช้งานต้องการลิงค์ไฟล์ข้อมูล s.txt ชื่อว่า z.txt โดยใช้คำสั่ง ln s.txt z.txt เมื่อสร้าง z.txt สำเร็จจะเห็นได้ว่า z.txt มีข้อมูล เหมือนกับ s.txt ทุกประการ พร้อมทั้งเมื่อมีการเปลี่ยนแปลงแก้ไขข้อมูลใน s.txt ไฟล์ข้อมูล z.txt ก็ จะเปลี่ยนแปลงไปด้วย ในทางกลับกันหากมีการเปลี่ยนแปลงใน z.txt ไฟล์ข้อมูล s.txt ก็จะเปลี่ยนแปลงเช่นเดียวกัน และจำนวนตัวเลข 1 จะเปลี่ยนเป็น 2 ซึ่งจะเห็นได้ว่าการสร้างลิงค์ไฟล์อาจมีความต้องการกระทำในสิ่งที่ไม่พึงประสงค์ ข้อมูล Uid เป็นข้อมูลของผู้ที่มีสิทธิครอบครองและเป็นเจ้าของไฟล์ข้อมูล เป็นข้อมูลหมายเลขที่ไม่ซ้ำกัน โดยปกติหมายเลขชุดนี้จะถูกกำหนดโดยระบบปฏิบัติการตั้งแต่ขั้นตอนการสร้างชื่อผู้ใช้งาน (Username) ใหม่เพิ่มในระบบปฏิบัติการ โดยผู้ใช้จะใช้ ชื่อผู้ใช้งานในการทำงาน ระบบปฏิบัติการจะรู้จักผู้ใช้งานผ่านหมายเลขนี้เท่านั้น สร้างชื่อผู้ใช้งานใหม่ระบบปฏิบัติการจะมีการกำหนดตัวเลข โดยปกติผู้ใช้ทั่วไปจะมีหมายเลขเริ่มต้นที่ 1001 เป็นต้นไป หากมีการเปลี่ยนแปลงก็อาจเป็นไปได้ว่าเกิดความผิดปกติเช่นมีการสร้างไฟล์ขึ้นมาใหม่ที่มีชื่อเดียวกัน โดยอาจกเกิดจากการลบไฟล์ข้อมูลแล้วสร้างใหม่ การคัดลอกไฟล์ข้อมูลมาแทนที่เป็นต้น ข้อมูล Gid เป็นข้อมูลของกลุ่มของผู้ใช้ที่มีสิทธิเข้าถึงไฟล์ข้อมูล โดยปกติข้อมูลจะเป็นตัวเลข โดยหากเป็นกลุ่มของระบบจะเป็นเลข 0-1000 และหากเป็นกลุ่มที่ถูกสร้างขึ้นใหม่จะเริ่มต้นที่ 1001 เป็นต้นไป ข้อมูล Gid นี้ก็มีลักษณะสอดคล้องกันกับข้อมูล Uid ในการใช้ในการพิจารณาตรวจสอบหาผู้รับผิดชอบต่อการกระทำในเหตุการณ์ที่เกิดขึ้น ข้อมูล Size เป็นข้อมูลที่แสดงถึงขนาดของไฟล์ข้อมูล สามารถใช้ในการตรวจสอบข้อมูลได้ถูกเปลี่ยนแปลงถูกเพิ่มหรือลบทิ้ง แต่ในบางกรณีการเกิดเหตุการณ์เปลี่ยนแปลงนี้อาจจะทำให้มีขนาดเท่าเดิมจึงต้องใช้ข้อมูลอื่นๆเข้าร่วมในการพิจารณา ข้อมูล Inode เป็นข้อมูลที่เก็บในลักษณะของกลุ่มตัวเลขหนึ่งชุด จะไม่มีวันเกิดขึ้นซ้ำได้อีกและจะไม่เปลี่ยนแปลงจนกว่าจะมีการย้ายที่อยู่หรือลบและถูกสร้างใหม่ของไฟล์ข้อมูล ดังนั้นข้อมูลนี้จึงสามารถบ่งบอกถึงไฟล์ข้อมูลว่าเป็นต้นฉบับหรือไม่หากกลุ่มตัวเลขเปลี่ยนไปก็แสดงว่าไฟล์ต้นฉบับถูกลบทิ้งและมีการสร้างไฟล์ใหม่ขึ้นมาทดแทน

ข้อมูล Atime เป็นข้อมูลที่แสดงเวลาของผู้ใช้งานนี้ล่าสุดโดยข้อมูลนี้จะถูกเปลี่ยนแปลงทุกครั้งที่มีการเข้าถึงไฟล์ข้อมูล ข้อมูล Mtime เป็นข้อมูลที่แสดงถึงการเปลี่ยนแปลง

ครั้งล่าสุดของไฟล์โดยหากมีการเปลี่ยนแปลงข้อมูลในไฟล์ข้อมูล ข้อมูล Ctime เป็นข้อมูลที่บันทึกเกี่ยวกับการเปลี่ยนแปลงของเจ้าของไฟล์ข้อมูล กลุ่มของผู้ใช้และคุณสมบัติของไฟล์ข้อมูล ซึ่งจะมีความแตกต่างกันของข้อมูล จากข้อมูลทั้ง 3 ข้อมูลนี้หากเวลาที่บันทึกไว้มีความเปลี่ยนแปลงก็สามารถนำไปใช้ในการพิจารณาหาความผิดปกติได้และทั้ง 3 ข้อมูลนี้สามารถนำไปใช้พิจารณาร่วมกับข้อมูลอื่นๆเพื่อให้ได้ข้อมูลที่ถูกต้องมากขึ้นด้วย

3. **การเปรียบเทียบระหว่างการทำงานของระบบบันทึกเหตุการณ์ที่ไม่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนดกับการทำงานของระบบบันทึกเหตุการณ์ที่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนด**

ในการทดลองเพื่อการเปรียบเทียบระหว่างการทำงานของระบบบันทึกเหตุการณ์ที่ไม่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนดกับการทำงานของระบบบันทึกเหตุการณ์ที่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนด ผู้วิจัยจะตั้งค่าการทดลอง โดยกำหนดให้ระบบบันทึกเหตุการณ์ทั้งสองแบบ ตรวจสอบการเข้าถึงไฟล์ข้อมูล s.txt ตั้งค่า sleeping time 250 microsecond และทำการทดสอบการทำงานจำนวน 10 รอบ โดยแต่ละรอบระบบบันทึกเหตุการณ์จะทำการตรวจสอบจำนวน 1000 ครั้ง และหาค่าเฉลี่ยของระยะเวลาในการทำงานของระบบบันทึกเหตุการณ์

ตาราง 1 แสดงผลการทดลองการทำงานระบบบันทึกเหตุการณ์ที่ไม่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนดและระบบบันทึกเหตุการณ์ที่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนด

	NoInode			Inode		
Round	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**
Round 1.	327	1451	433	324	865	432
Round 2.	327	2614	426	287	884	419
Round 3.	346	1343	424	311	819	416
Round 4.	310	1297	425	320	1724	421
Round 5.	315	1261	407	327	1970	427
Round 6.	274	1181	425	327	819	411
Round 7.	336	3300	419	289	1393	409
Round 8.	317	1127	429	308	782	426
Round 9.	312	1680	431	292	930	415
Round 10.	283	2320	405	357	1678	408
Average	315	1757	**422**	314	1186	**418**

จากตาราง 1 ผลการทดลองการทำงานระบบบันทึกเหตุการณ์ที่ไม่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนดและระบบบันทึกเหตุการณ์ที่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนด ผู้วิจัยพบว่า จากการทดลองการหาระยะเวลาในการทำงานของระบบบันทึกเหตุการณ์ทั้งสองแบบ นั้นมีการใช้เวลาในการทำงานที่ใกล้เคียงกัน การเพิ่มข้อมูลโครงสร้างข้อมูลไอโหนดของระบบบันทึกเหตุการณ์ ไม่มีความแตกต่างจากการทำงานของ ระบบบันทึกเหตุการณ์ที่ไม่มีการเพิ่มข้อมูลโครงสร้างข้อมูลไอโหนด โดยเวลาการทำงานเฉลี่ยจากการทดลองนั้น ระบบบันทึกเหตุการณ์ที่ไม่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนด เท่ากับ 418 Microsecond และ ระบบบันทึกเหตุการณ์ที่มีการเพิ่มข้อมูลในโครงสร้างข้อมูลไอโหนด เท่ากับ 422 Microsecond จากการสังเกตจะพบว่าระยะเวลาในการทำงานของระบบบันทึกเหตุการณ์ ที่ค่าสูงนั้นเกิดจากในช่วงเวลาทำงานของโปรแกรมนั้นมีการทำงานอื่นๆ ของระบบปฏิบัติการจึงส่งผลให้มีระยะเวลาการทำงานที่เพิ่มมากขึ้นแต่ก็เป็นระยะเวลาในช่วงสั้นเท่านั้น

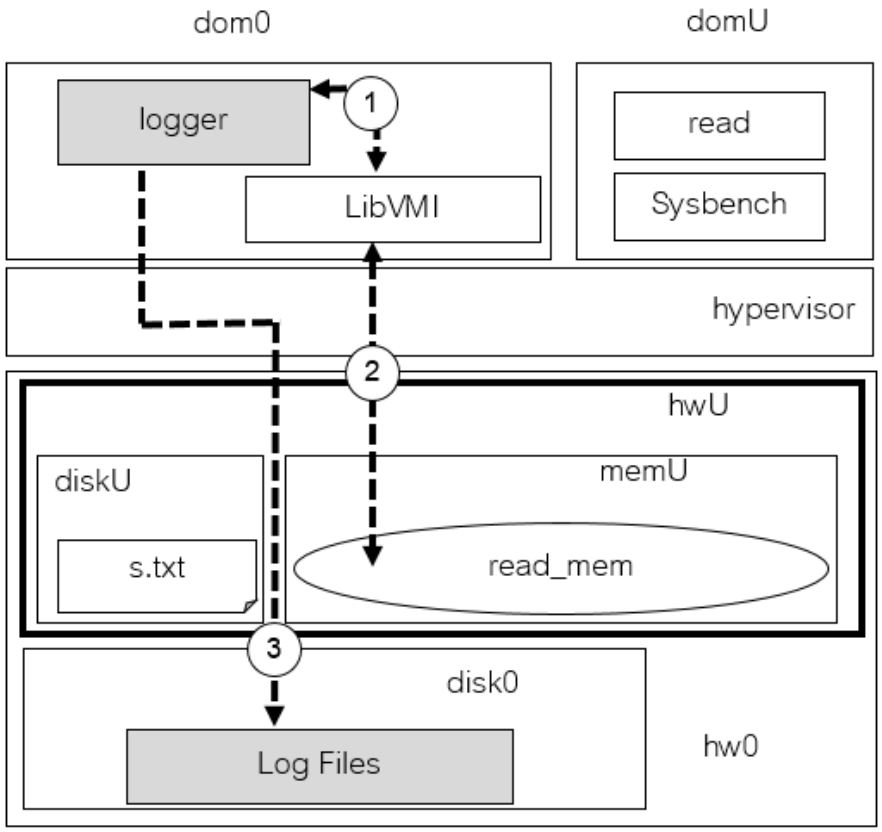
4. สรุปผลจากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

จากหัวข้อเรื่อง การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์นั้น ทำให้มีข้อมูลของระบบบันทึกเหตุการณ์เพิ่มขึ้นจากเดิมรวมทั้งหมดเป็น 14 ข้อมูลที่ประกอบไปด้วย Directory File Pid Pname User Mode Link Uid Gid Size Inode Atime Mtime และ Ctime การเพิ่มข้อมูลเหล่านี้ส่งผลให้สามารถนำมาใช้ในการพิจารณาตรวจสอบหาผู้รับผิดชอบต่อการกระทำของผู้ใช้งาน เรื่องการละเมิดข้อมูล (Data breaches) คือ การดูข้อมูล (Viewed) การเปิดเผยข้อมูล (Release) การขโมยข้อมูล Stolen) การใช้งานโดยไม่ได้รับอนุญาต (Used) และ รวมไปถึงการลบข้อมูล (Remove) โดยไม่ส่งผลต่อระยะเวลาการทำงานของระบบบันทึกเหตุการณ์

ผลจากการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

ในหัวข้อนี้จะนำเสนอเกี่ยวกับการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ เนื่องการศึกษาของผู้วิจัยยังไม่ปรากฏการวัดและทดสอบประสิทธิภาพการทำงานในส่วนของผู้ใช้บริการเมื่อระบบบันทึกเหตุการณ์กำลังทำงานในเวลาเดียวกัน ในการวัดประสิทธิภาพของเครื่องคอมพิวเตอร์ของผู้ใช้บริการด้วย SysBench รายละเอียดของ SysBench ได้ถูกอธิบายไว้ในหัวข้อเรื่อง Sysbench ในบทที่ 2 ที่ติดตั้งไว้บน

เครื่องคอมพิวเตอร์ของผู้ใช้บริการดังภาพ 26 ซึ่งจะทำหน้าที่วัดการประมวลผล โดยผู้วิจัยได้ทำการวัดการประมวลผลเมื่อเปิดการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้งานโดยไม่มีการเปิดใช้งานโปรแกรมใดๆที่นอกเหนือจากโปรแกรมที่จำเป็นของระบบปฏิบัติการ ซึ่งการวัดค่านี้จะได้ผลลัพธ์ เท่ากับ 10.0546 microsecond จากนั้นผู้วิจัยได้วัดค่าอีกครั้งพร้อมทั้งมีการเปิดการทำงานระบบบันทึกเหตุการณ์ ซึ่งผลลัพธ์ที่ได้เท่ากับ 23.5905 microsecond ซึ่งพบว่าความสามารถในการประมวลผลของเครื่องคอมพิวเตอร์เพิ่มขึ้นถึง 42% จึงแสดงให้เห็นว่าเมื่อมีการใช้งานระบบบันทึกเหตุการณ์มีผลต่อการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้บริการ เนื่องจากการทำงานของล็อกเกอร์ที่จะมีการหยุดการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้บริการเพื่อเข้าถึงหน่วยความจำหลักในการตรวจสอบข้อมูลเหตุการณ์



ภาพ 26 สถาปัตยกรรมของระบบบันทึกเหตุการณ์ที่ใช้ทดลองเมื่อติดตั้ง SysBench

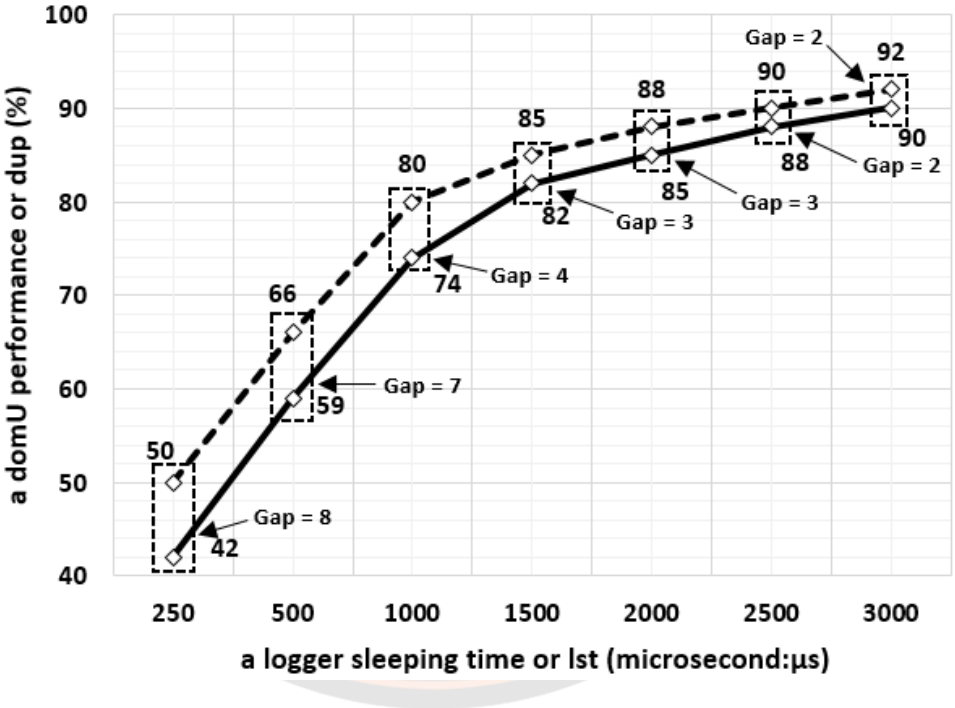
ในการออกแบบการทดลองการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์นี้ ต้องรู้ถึงการทำงานของระบบบันทึกเหตุการณ์ จากภาพ 26 จะแสดงถึงสถาปัตยกรรมของระบบบันทึกเหตุการณ์ที่อยู่บนการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐาน (Infrastructure as a service cloud) ประกอบด้วย hypervisor dom0 domU hw0 hwU disk0 diskU และ memU ในการทำงานของระบบบันทึกเหตุการณ์จะมีขั้นตอนที่จำเป็นต่อการทดลอง อยู่ 3 ขั้นตอน คือจากภาพ 26 ในวงกลมหมายเลข 1 -3 ขั้นตอนที่ 1 ล็อกเกอร์เรียกใช้งาน LibVMI ขั้นตอนที่ 2 เมื่อเรียกใช้งาน LibVMI แล้วจะทำการตรวจสอบข้อมูล (read_mem) ในหน่วยความจำหลักของเครื่องคอมพิวเตอร์ของผู้ใช้บริการ (memU) หากตรงตามเงื่อนไขจะได้ข้อมูลทั้งหมด 15 ข้อมูล ประกอบไปด้วย loginuid pid Process name path files i_size i_mode i_uid i_gid i_ino i_nlink i_atime i_mtime และ i_ctime ขั้นตอนที่ 3 จะส่งข้อมูลที่ได้รับไปบันทึกลงในไฟล์ข้อมูลเหตุการณ์ (Log Files) ในหน่วยความจำสำรองของเครื่องคอมพิวเตอร์ของผู้ให้บริการ โดยในระบบระบบบันทึกเหตุการณ์ที่ปรับปรุงนี้ เมื่อดำเนินการเสร็จสิ้นในขั้นตอนที่ 3 ล็อกเกอร์ จะหยุดรอเพื่อเริ่มทำงานใหม่ตามเวลา sleeping time ที่กำหนดไว้ จากขั้นตอนการทำงานเหล่านี้ ผู้วิจัยได้นำวิธีทางคำนวณทางคณิตศาสตร์หาผลลัพธ์ เพื่อนำไปเปรียบเทียบกับการทดลอง โดยสามารถคำนวณจำนวนรอบของการทำงานของล็อกเกอร์ได้จากสมการที่ (1) โดย n คือจำนวนรอบที่ระบบบันทึกเหตุการณ์ทำงานเพื่อตรวจสอบข้อมูล sbt คือค่าผลลัพธ์ที่ได้จากการวัดจาก SysBench lst คือ เวลาที่กำหนดไว้ของ sleeping time ซึ่งในการทดลองได้กำหนด sleeping time คือ 250 500 1000 1500 2000 2500 และ 3000 microsecond และ ในส่วนของสมการที่ (2) แสดงการคำนวณหา dup หรือร้อยละ (%)ของประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ โดย lpt คือ เวลาในการทำงานของล็อกเกอร์ มีค่าเท่ากับ 250 microsecond ซึ่งค่านี้ได้จากการทดลองวัดเวลาเฉลี่ยการทำงานของล็อกเกอร์จากการใช้งานจริง ϵ คือค่าเวลาที่เกิดขึ้นโดยไม่ทราบสาเหตุที่แน่ชัด ค่านี้ผู้วิจัยได้กำหนดไว้ที่ 330 microsecond

$$n = \frac{sbt}{lst} \tag{1}$$

$$dup = \left(\frac{\sum_{i=1}^{n} lst_i}{\sum_{i=1}^{n} lpt_i + \sum_{i=1}^{n} lst_i + \sum_{i=1}^{n} \epsilon_i}\right) * 100 \tag{2}$$

จากภาพ 27 แสดงผลเปรียบเทียบ dup จากการคำนวณ กับ dup ที่ได้จากการทดลอง โดยแกน Y จะแสดง dup หรือร้อยละของประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของ

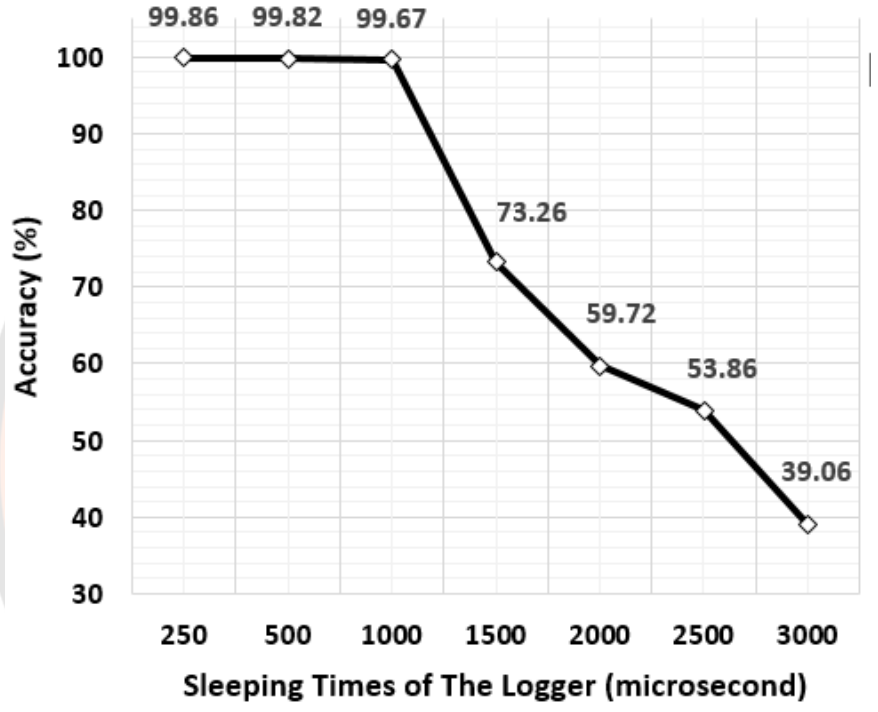
ผู้ใช้บริการ แกน X จะแสดงถึงค่าเวลาของ sleeping time ในช่วงเวลา 250 - 3000 microsecond โดยเพิ่มครั้งล่ะ 500 microsecond ตามลำดับ การทดลองเปลี่ยนแปลงค่า sleeping time ของระบบบันทึกเหตุการณ์ เมื่อค่า sleeping time เท่ากับ 250 microsecond ค่า dup จากการคำนวณ จะมีค่าเท่ากับ 50% ในส่วนของ dup ที่ได้จากการทดลอง มีค่าเท่ากับ 42% เมื่อเพิ่มค่า sleeping time เท่ากับ 500 microsecond ค่า dup จากการคำนวณ จะมีค่าเท่ากับ 66% ในส่วนของ dup ที่ได้จากการทดลอง มีค่าเท่ากับ 59% และในช่วง 1000 – 3000 microsecond ค่า dup จากการคำนวณ จะมีค่าเท่ากับ 80 - 92% ในส่วนของ dup ที่ได้จากการทดลอง มีค่าเท่ากับ 74 - 90%



ภาพ 27 เปรียบเทียบ dup จากการคำนวณ กับ dup ที่ได้จากการทดลอง

จากผลของ dup ที่ได้จากการคำนวณกับ dup ที่ได้จากการทดลองจะเห็นได้ว่าเมื่อมีการเปลี่ยนแปลงค่า sleeping time ที่สูงขึ้น ค่าความแตกต่างกันระหว่างของ dup จากการคำนวณกับ dup ที่ได้จากการทดลอง จะมีค่าใกล้เคียงกัน เนื่องจาก ϵ หรือค่าเวลาที่เกิดขึ้นโดยไม่ทราบสาเหตุที่แน่ชัด ที่เกิดขึ้นจากการทำงานของระบบบันทึกเหตุการณ์ในแต่ละรอบนั้น เกิดขึ้นน้อยกว่าน้อยกว่าเนื่องจากจำนวนรอบในการทำงานนั้นน้อยกว่านั้นเอง แต่ทว่าค่า ϵ นี้เป็นค่าที่มี

ไม่แน่นอนอาจจะมีค่ามากหรือมีค่าน้อย ในการทดลองค่าของ ϵ นี้จะมีค่ามากสุดในช่วงเปิดใช้งานระบบบันทึกเหตุการณ์ครั้งแรก ในขณะที่ระบบบันทึกเหตุการณ์ทำงานไปสักระยะค่านี้จะมีค่าน้อยลงซึ่งอาจจะส่งผลทำให้ค่า dup ในการใช้งานจริงมีประสิทธิภาพมากขึ้น ขณะเดียวกันในการทำงานของระบบบันทึกเหตุการณ์ที่มีค่า sleeping time ที่มากขึ้น dup จะมีประสิทธิภาพมากขึ้นกว่าระบบบันทึกเหตุการณ์ที่ใช้ sleeping time ที่มีค่าน้อยกว่า

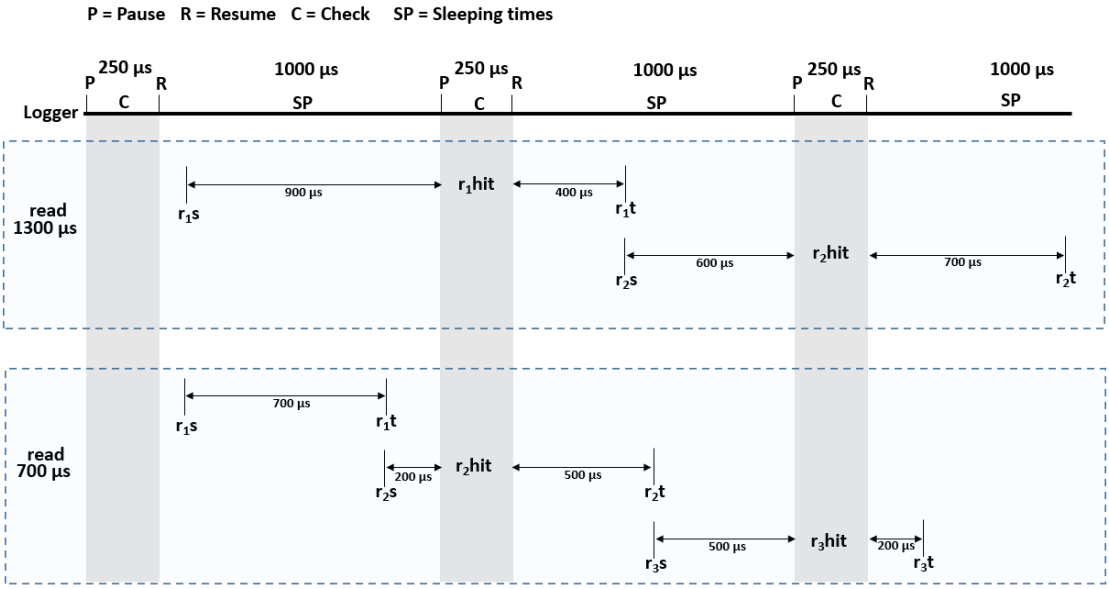


ภาพ 28 ค่าความแม่นยำของระบบบันทึกเหตุการณ์ในแต่ละช่วงเวลาของ sleeping time

เมื่อมีการทดลองเพื่อหาประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการแล้วจะต้องทดลองวัดความแม่นยำของระบบบันทึกเหตุการณ์เมื่อมีการเปลี่ยนแปลง sleeping time ด้วยเพื่อหาความสอดคล้องกันระหว่างประสิทธิภาพของระบบบันทึกเหตุการณ์และการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ ซึ่งผู้วิจัยได้ทำการทดลองเพื่อทดสอบการทำงานของระบบบันทึกเหตุการณ์จำนวน 10 รอบ โดยแต่ละรอบระบบบันทึกเหตุการณ์จะทำการตรวจสอบจำนวน 1000 ครั้ง ในแต่ล่ะช่วงเวลาของ sleeping time เริ่มจาก 250 500 1000 1500 2000 2500 และ 3000 microsecond ตามลำดับ

จากภาพ 28 แสดงค่าความแม่นยำของระบบบันทึกเหตุการณ์ในแต่ละช่วงเวลาของ sleeping time โดยแกน Y จะแสดงค่าความแม่นยำ แกน X จะแสดงถึงค่าเวลาของ sleeping time ในช่วงเวลา 250 - 3000 microsecond โดยเพิ่มครั้งล่ะ 500 microsecond ตามลำดับ การทดลองเปลี่ยนแปลงค่า sleeping time ของระบบบันทึกเหตุการณ์ เริ่มการทำลองโดยการกำหนดเวลาของ sleeping time เท่ากับ 250 microsecond ความแม่นยำได้นั้นจะเท่ากับ 99.86% เมื่อค่า sleeping time เท่ากับ 500 microsecond ความแม่นยำได้นั้นจะเท่ากับ 99.82% และ เมื่อค่า sleeping time เท่ากับ 1000 microsecond ความแม่นยำได้นั้นจะเท่ากับ 99.67% ซึ่งความแม่นยำที่เกิดขึ้นใน 250 500 และ 1000 microsecond จะมีค่าความแม่นยำที่ใกล้เคียง 100% แต่เมื่อค่า sleeping time เท่ากับ 1500 2000 2500 และ3000 microsecond ความแม่นยำได้นั้นจะเท่ากับ 73.26% 59.72% 53.86% และ 39.06% จะพบว่าความแม่นยำของระบบบันทึกเหตุการณ์จะลดลงอย่างชัดเจน

การลดลงของความแม่นยำในการทำงานของระบบบันทึกเหตุการณ์ เมื่อมีการปรับเปลี่ยนค่าเวลาของ sleeping time ที่แตกต่างกันนั้น เกิดจากระยะเวลาการทำงานของโพรเซสของผู้ใช้งานมีระยะเวลาน้อยกว่าค่า sleeping time หมายความว่าการเริ่มต้นและสิ้นสุดการทำงานของโพรเซสนั้นเกิดขึ้นในช่วงเวลา sleeping time ทำให้ระบบบันทึกเหตุการณ์ไม่สามารถตรวจสอบโพรเซสนี้ได้ดังภาพ 29 ซึ่งแสดงกลไกการทำงานของการตรวจสอบ เมื่อเริ่มตรวจสอบหรือจุด P มีความหมายว่าระบบบันทึกเหตุการณ์จะสั่งให้เครื่องคอมพิวเตอร์ของผู้ใช้บริการหยุดทำงาน ซึ่งในเวลานี้จะทำการตรวจสอบข้อมูลเป็นเวลา 250 ดังจุด C เมื่อหยุดจนครบเวลาแล้วจะสั่งให้คอมพิวเตอร์ของผู้ใช้บริการทำงานตามปกติในจุด R ซึ่งเวลาที่ให้ทำงานนั้นจะเป็นเวลาที่ถูกกำหนดไว้ให้กับ sleeping time มีค่าเท่ากับ 1000 microsecond ในจุด SP เมื่อครบกำหนดเวลาก็เริ่มทำงานที่จุด C ทำงานวนไปเรื่อยๆ จากกลไกลการทำงานของระบบบันทึกเหตุการณ์นี้ ดังนั้นจะเห็นได้ว่าระบบบันทึกเหตุการณ์ได้มีการกำหนดเวลาของ sleeping time มีค่าเท่ากับ 1000 microsecond โพรเซส read ที่มีระยะเวลาการเริ่มต้นและสิ้นสุดการทำงานอยู่ที่ 1300 microsecond จะถูกตรวจสอบพบทุกครั้งที่โพรเซสเกิดขึ้น ในขณะเดียวกัน โพรเซส read ที่มีระยะเวลาการเริ่มต้นและสิ้นสุดการทำงานอยู่ที่ 700 microsecond นั้นมีโอกาสที่ระบบบันทึกเหตุการณ์ไม่สามารถตรวจสอบพบโพรเซสนี้ได้ ถ้าหากว่าโพรเซสนี้เกิดขึ้นในช่วงเวลา sleeping time ของระบบบันทึกเหตุการณ์



ภาพ 29 กลไกลการตรวจสอบของระบบบันทึกเหตุการณ์

สรุปภาพรวมบทที่ 4

ในหัวข้อนี้จะกล่าวถึงภาพรวมทั้งหมดของบทที่ 4 ผลการวิจัย รายละเอียดแต่ละหัวข้อดังต่อไปนี้

หัวข้อเรื่อง ผลจากการทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม

หัวข้อเรื่อง ผลจากการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

หัวข้อเรื่อง ผลจากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

หัวข้อเรื่อง ผลจากการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

รายละเอียดของข้อสรุปในแต่ละหัวข้อ

ในหัวข้อเรื่อง ผลจากการทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม ผู้อ่านจะได้ทราบการทดสอบเพื่อหาประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์เดิมซึ่งประกอบไปด้วยการทดลองทั้งหมด 2 ส่วนคือ การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลาง และ การทดสอบประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์โดยการเปลี่ยนแปลงขนาดของหน่วยความจำหลัก และจากการ

ทดผู้วิจัยได้พบข้อจำกัดและปัญหาในการทำงานของระบบบันทึกเหตุการณ์ คือ ขั้นตอนการทำงานของระบบบันทึกเหตุการณ์ การจัดการหน่วยความจำในการทำงานของระบบบันทึกเหตุการณ์ การจัดการเวลา sleeping time ที่เป็นระยะเวลาที่มีผลต่อประสิทธิภาพการทำงานของของระบบบันทึกเหตุการณ์ และ ข้อมูลเหตุการณ์ที่มีน้อยเกินไป ดังนั้นผู้วิจัยได้ทำการออกแบบขั้นตอนและกระบวนทำงานของของระบบบันทึกเหตุการณ์ใหม่ ในหัวข้อเรื่อง ผลจากการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ สามารถแก้ไขข้อจำกัดและปัญหา ในเรื่องของขั้นตอนการทำงานของระบบบันทึกเหตุการณ์ การจัดการหน่วยความจำในการทำงานของระบบบันทึกเหตุการณ์ และ การจัดการเวลา sleeping time ที่เป็นระยะเวลาที่มีผลต่อประสิทธิภาพการทำงานของของระบบบันทึกเหตุการณ์ ซึ่งยังมีอีกหนึ่งปัญหาที่ได้รับการแก้ไขในหัวข้อเรื่อง ผลจากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ โดยจะเห็นว่า การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์นั้น ทำให้มีข้อมูลของระบบบันทึกเหตุการณ์เพิ่มขึ้นจากเดิมรวมทั้งหมดเป็น 14 ข้อมูลที่ประกอบไปด้วย Directory File Pid Pname User Mode Link Uid Gid Size Inode Atime Mtime และ Ctime การเพิ่มข้อมูลเหล่านี้ส่งผลให้สามารถนำมาใช้ในการพิจารณาตรวจสอบหาผู้รับผิดชอบต่อการกระทำของผู้ใช้งาน เรื่องการละเมิดข้อมูล (Data breaches) คือ การดูข้อมูล (Viewed) การเปิดเผยข้อมูล (Release) การขโมยข้อมูล Stolen) การใช้งานโดยไม่ได้รับอนุญาต (Used) และ รวมไปถึงการลบข้อมูล (Remove) โดยไม่ส่งผลต่อระยะเวลาการทำงานของระบบบันทึกเหตุการณ์ และสุดท้ายหัวข้อเรื่อง ผลจากการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ เป็นการวัดและทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์ที่คำนึงถึงการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการโดยออกแบบวิธีการวัด การทดลอง ซึ่งผลที่ได้นั้นจะได้ผลลัพธ์ที่มีความสัมพันธ์กัน ซึ่งจะกล่าวในบทที่ 5 สรุปและอภิปราย ต่อไป

บทที่ 5

บทสรุป

ระบบบันทึกเหตุการณ์จะเป็นระบบที่ทำหน้าที่บันทึกเหตุการณ์ที่เกิดขึ้นในคลาวด์ ซึ่งระบบบันทึกเหตุการณ์นี้ มีผู้วิจัยมากมายได้ทำการพัฒนาสร้างระบบบันทึกเหตุการณ์ ที่สามารถติดตั้งไว้ที่ฝั่งของผู้ให้บริการ หรือผู้ใช้บริการ หรือทั้งสองฝั่ง ซึ่งมีหลักการทำงานที่แตกต่างกัน ในวิทยานิพนธ์นี้ผู้วิจัยเลือกศึกษาระบบบันทึกเหตุการณ์ที่ติดตั้งในส่วนของผู้ให้บริการ เนื่องจากการทำงานของระบบบันทึกเหตุการณ์ลักษณะนี้มีผลกระทบต่อการใช้งานผู้ใช้บริการน้อยมากเมื่อเปรียบเทียบกับระบบบันทึกเหตุการณ์ลักษณะอื่น ๆ การทำงานของระบบบันทึกเหตุการณ์จะใช้กระบวนการบันทึกเหตุการณ์โดยวิธี Introspection ซึ่งเป็นเทคนิคสำหรับการตรวจสอบการทำงานของโพรเซสที่เกิดขึ้นในเครื่องคอมพิวเตอร์เสมือน (Virtual Machine)โดยการตรวจสอบข้อมูลต่างๆ ผ่านทางหน่วยความจำหลัก โดยในระบบบันทึกเหตุการณ์นี้จะมีโพรเซสที่ทำหน้าที่ตรวจสอบการทำงานของโพรเซสที่เกิดขึ้นในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ เรียกว่า ล็อกเกอร์ (logger) ในการทำงานหากข้อมูลในโพรเซสที่ตรวจสอบนั้นตรงตามเงื่อนไขที่กำหนดไว้ จะทำการบันทึกเหตุการณ์ที่เกิดขึ้นในคอมพิวเตอร์ในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ เช่น การอ่าน การเปลี่ยนแปลง แก้ไข การสำเนา การคัดลอก และการลบไฟล์ข้อมูลของผู้ใช้บริการบนคลาวด์ ซึ่งการบันทึกข้อมูลเหตุการณ์จะถูกเก็บไว้ใน ล็อกไฟล์ (log files)หรือไฟล์บันทึกเหตุการณ์ เพื่อนำไปใช้พิจารณาเพื่อค้นหาผู้รับผิดชอบเมื่อเกิดปัญหาหรือเกิดความเสียหาย

วิทยานิพนธ์นี้มุ่งเน้นที่จะปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ โดยระบบบันทึกเหตุการณ์เป็นระบบที่สามารถช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อคลาวด์ได้ ลักษณะการทำงานของระบบบันทึกเหตุการณ์จะทำให้ทราบว่า บุคคลใดเข้าถึงไฟล์อะไรบ้าง หรือได้กระทำอะไรกับไฟล์เหล่านั้นบ้าง การกระทำลักษณะนี้อาจเป็นส่วนหนึ่งของภัยคุกคามที่เรียกว่าการละเมิดข้อมูล (Data breach) ซึ่งเป็นภัยคุกคามที่รายงานขององค์กร CSA ให้อยู่ในลำดับที่ 1 ของภัยคุกคามทั้งหมดซึ่งกล่าวได้ว่าเป็นภัยคุกคามที่สำคัญและเกิดขึ้นบ่อยครั้ง ดังนั้นระบบบันทึกเหตุการณ์ดังกล่าวจะเป็นส่วนหนึ่งในการช่วยให้ผู้มีส่วนเกี่ยวข้องสามารถดำเนินการสืบสวนหาตัวบุคคลที่กระทำผิดมารับผิดชอบต่อการกระทำได้ การศึกษาการทำงานเกี่ยวกับระบบบันทึกเหตุการณ์เป็นสิ่งที่สำคัญของงานวิทยานิพนธ์นี้ การปรับปรุงและเพิ่มประสิทธิภาพการทำงานจะทำให้ระบบบันทึกเหตุการณ์สามารถทำงานได้อย่างรวดเร็ว ถูกต้อง แม่นยำ มากขึ้น อีกทั้งยังช่วยลดข้อจำกัดต่าง ๆ ที่เกิดจากการทำงาน ทั้งในส่วนของผู้ให้บริการ และผู้ใช้บริการ ซึ่ง

ผู้วิจัยได้กำหนดวัตถุประสงค์การศึกษาไว้ทั้งหมด 3 หัวข้อ ซึ่งจะสอดคล้องกับปัญหาวิจัยหลักของวิทยานิพนธ์ (research gaps) ที่กล่าวในหัวข้อเรื่อง ปัญหาวิจัยหลักของวิทยานิพนธ์ โดยรายละเอียดเกี่ยวกับวัตถุประสงค์ของงานวิจัย มีดังนี้

1. เพื่อเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ในด้านการจัดการทรัพยากรและความรวดเร็วของกระบวนการทำงาน

2. เพื่อเพิ่มข้อมูลในล็อกไฟล์ของระบบบันทึกเหตุการณ์เกี่ยวกับการพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูลของผู้ใช้บริการ

3. การวัดและประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ของผู้ให้บริการที่มีผลต่อการใช้งานของผู้ใช้บริการ

สรุปผลการวิจัย

จากวัตถุประสงค์ของงานวิจัยทั้ง 3 ข้อ คือ 1. เพื่อเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ในด้านการจัดการทรัพยากรและความรวดเร็วของกระบวนการทำงาน 2. เพื่อเพิ่มข้อมูลในล็อกไฟล์ของระบบบันทึกเหตุการณ์เกี่ยวกับการพิจารณาหาความผิดปกติของการเข้าถึงไฟล์ข้อมูลของผู้ใช้บริการ 3. การวัดและประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์ของผู้ให้บริการที่มีผลต่อการใช้งานของผู้ใช้บริการ ซึ่งเป็นวัตถุประสงค์ที่ถูกกล่าวไว้ที่หัวข้อเรื่องวัตถุประสงค์ของการศึกษา และการทดลองในวิทยานิพนธ์นี้มุ่งเน้นการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์ คือ การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์ การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ การวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งาน โดยรายละเอียดจะอยู่ในบทที่ 3 วิธีการดำเนินงานวิจัย

1. การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม

ในการทดลองในหัวข้อเรื่อง การทดลองการทำงานของระบบบันทึกเหตุการณ์เดิม แบ่งการออกทดลองออกเป็น 4 การทดลองประกอบไปด้วย

1.1 การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

1.2 การเปลี่ยนแปลงจำนวนแกนของหน่วยประมวลผลกลางในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

1.3 การเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ให้บริการ

1.4 การเปลี่ยนแปลงขนาดของหน่วยความจำหลักในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ

ผลจากการทดลองที่นำเสนอในบทความวิชาการชื่อเรื่อง “The Accuracy Measurement of Logging Systems on Different Hardware Environments in Infrastructure as a Service Cloud” (Auxsorn, Wongthai, Porka, & Jaiboon, 2020) ผู้วิจัยได้พบข้อจำกัดและปัญหาของระบบบันทึกเหตุการณ์ในเรื่องเกี่ยวกับ การจัดการกระบวนการทำงานของระบบบันทึกเหตุการณ์ที่ไม่มีประสิทธิภาพ การจัดการทรัพยากรหน่วยความจำหลักที่ไม่ดีพอ จำนวนข้อมูลในไฟล์บันทึกเหตุการณ์ที่น้อยไม่เพียงพอต่อการตรวจสอบหาผู้รับผิดชอบต่อการกระทำ อีกทั้งยังพบว่า sleeping time นั้น มีผลต่อการทำงานของระบบบันทึกเหตุการณ์ ซึ่งทั้งหมดนี้ถูกล่าวไว้ที่หัวข้อ เรื่อง สรุปข้อจำกัดและปัญหาที่ได้จากการทดลอง

2. การปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

การทดลองนี้ ผู้วิจัยได้ศึกษาวิธีการ แนวทาง แก้ไขและปรับปรุงข้อจำกัดและปัญหาที่เกิดขึ้นจากการทดลองใช้งานระบบบันทึกเหตุการณ์ และดำเนินแก้ไขปรับปรุงในส่วนของขั้นตอนการทำงานของระบบบันทึกเหตุการณ์ การจัดการหน่วยความจำในการทำงานของระบบบันทึกเหตุการณ์ และ การจัดการเวลา sleeping time ที่เป็นระยะเวลาที่มีผลต่อประสิทธิภาพการทำงานของของระบบบันทึกเหตุการณ์ โดยการออกแบบขั้นตอนและกระบวนการทำงานของระบบบันทึกเหตุการณ์ใหม่ที่มีการทำงานที่ตัดการทำงานที่ซ้ำซ้อนและไม่จำเป็นออกไปพร้อมทั้งสามารถจัดการเวลาของ sleeping time ได้ ซึ่งวิธีการทดลอง ผลการทดลอง การวัดประสิทธิภาพต่างๆจะถูกล่าวไว้ในหัวข้อเรื่อง ผลจากการปรับปรุงการทำงานและเพิ่มประสิทธิภาพของระบบบันทึกเหตุการณ์

3. การเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

จากปัญหาเรื่องข้อมูลของไฟล์บันทึกเหตุการณ์ที่ยังมีข้อมูลที่นำมาใช้พิจารณาตรวจสอบเพื่อหาผู้รับผิดชอบต่อการกระทำไม่เพียงพอผู้วิจัยจึงมีแนวคิดนำข้อมูลในโครงสร้างข้อมูลไอโหนดจำนวน 9 ข้อมูลประกอบไปด้วย ด้วย i_size i_mode i_uid i_gid i_ino i_nlink i_atime i_mtime และ i_ctime ซึ่งจากการเพิ่มข้อมูลเหล่านี้ทำให้มีความสามารถพิจารณาตรวจสอบ เรื่องการละเมิดข้อมูล (Data breaches) เช่น การดูข้อมูล (Viewed) การเปิดเผยข้อมูล (Release) การขโมยข้อมูล Stolen) การใช้งานโดยไม่ได้รับอนุญาต (Used) และ รวมไปถึงการลบข้อมูล (Remove) ได้ และที่สำคัญการเพิ่มข้อมูลเหล่านี้ไม่มีผลต่อเวลาการทำงานของระบบบันทึกเหตุการณ์ ซึ่งเนื้อหาทั้งหมดนี้ถูกกล่าวไว้ที่หัวข้อเรื่อง ผลจากการเพิ่มข้อมูลที่เป็นผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์

4. จากการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

การทดลองในหัวข้อนี้จะการวัดและทดสอบประสิทธิภาพการทำงานในส่วนของผู้ใช้บริการเมื่อระบบบันทึกเหตุการณ์กำลังทำงานในเวลาเดียวกัน นำเสนอในบทความวิชาการชื่อเรื่อง “Performance Considerations of a Logging System Simultaneously with a Customer Virtual Machine in Infrastructure as a Service Cloud” (Auxsorn, Wongthai, Phoka, & Jaiboon, 2020) ผู้วิจัยได้ใช้วิธีการคำนวณทางคณิตศาสตร์เพื่อหาประสิทธิภาพการทำงานโดยนำข้อมูลพื้นฐานที่ใช้ในการคำนวณมาจากเวลาของกระบวนการทำงานของระบบบันทึกเหตุการณ์เปรียบเทียบกับการทดลองใช้งานจริง ซึ่งผลปรากฏว่ามีค่าใกล้เคียงกันในทุกช่วงเวลาที่มีการเปลี่ยนแปลง sleeping time โดยเมื่อค่าเวลาของ sleeping time มากขึ้น ประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการจะมากขึ้น แต่ความแม่นยำของระบบบันทึกเหตุการณ์จะลดลง ในทางตรงกันข้ามหากค่าเวลาของ sleeping time น้อยลง ประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการจะน้อยลง แต่ความแม่นยำของระบบบันทึกเหตุการณ์จะมากขึ้นได้ ซึ่งวิธีการทดลอง ผลการทดลอง การวัดประสิทธิภาพต่างๆจะถูกล่าวไว้ในหัวข้อเรื่อง ผลจากการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์

อภิปรายผล

ระบบบันทึกเหตุการณ์ที่ได้รับการปรับปรุงนี้บรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามต่อการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐานได้อย่างมีประสิทธิภาพมากขึ้น ซึ่งจากผลการทดลองแสดงให้เห็นว่าระบบบันทึกเหตุการณ์มีกระบวนการทำงานที่เร็วมากขึ้นส่งผลทำให้มีความแม่นยำในการตรวจจับข้อมูล ใช้ทรัพยากรต่างๆ อย่างมีประสิทธิภาพ และที่เป็นหัวใจสำคัญของระบบบันทึกเหตุการณ์คือการค้นหาตรวจสอบเพื่อหาผู้รับผิดชอบต่อการกระทำ ในระบบบันทึกเหตุการณ์ใหม่นี้ได้มีการเพิ่มข้อมูลที่ใช้สำหรับการพิจารณาทั้งหมด 9 ตัว ส่งผลให้มีตรวจสอบเหตุการณ์ที่ปกติที่เกิดจากการกระทำได้ดีมากขึ้น ครอบคลุมการบรรเทาปัญหาภัยคุกคามเรื่องการละเมิดข้อมูล (Data breaches) เช่น การดูข้อมูล (Viewed) การเปิดเผยข้อมูล (Release) การขโมยข้อมูล Stolen) การใช้งานโดยไม่ได้รับอนุญาต (Used) และ รวมไปถึงการลบข้อมูล (Remove) ได้ ซึ่งผู้วิจัยขอยกตัวอย่างการตรวจสอบเหตุการณ์ผิดปกติจากข้อมูลของไฟล์ข้อมูลบันทึกเหตุการณ์ที่ได้จากระบบบันทึกเหตุการณ์ดังนี้

1. **ความผิดปกติจากการดูข้อมูล**

จากตาราง 2 แสดงการตรวจสอบเหตุการณ์เกี่ยวกับการดูข้อมูลที่น่าสงสัยว่าผิดปกติ จะเห็นได้ว่าโดยปกติแล้ว Alice ซึ่งเป็นผู้ใช้งานนั้นจะเรียกดูข้อมูลด้วยใช้ read แต่ข้อมูลที่แสดงในตาราง 2 แถวที่ 4 คอลัมน์ที่ 3 ซึ่งเป็นข้อมูลแสดงถึงชื่อโพสเซส ในการเรียกดูไฟล์ข้อมูลนั้นแตกต่างจากปกติคือเรียกดูข้อมูลด้วย cat หรือ ข้อมูลที่แสดงในตาราง 2 แถวที่ 5 คอลัมน์ที่ 8 ซึ่งเป็นข้อมูลแสดงเวลาในการเข้าถึงไฟล์ข้อมูล s.txt จะเห็นว่าเวลาที่มีการเรียกดูไฟล์ข้อมูลนั้นเกิดในช่วงเวลา 01:17:38 ซึ่งไม่ใช่เวลาทำงานของ Alice เป็นต้น ในการเข้าถึงข้อมูลที่ผิดปกตินี้อาจจะมีผลการละเมิดข้อมูลในเรื่องการเปิดเผยข้อมูล และการขโมยข้อมูลอีกด้วย

ตาราง 2 แสดงการตรวจสอบเหตุการณ์เมื่อมีการอ่านข้อมูล

user	File	Pname	Pid	Path	Atime
Alice	s.txt	read	5438	/home/tester/Downloads	2018/07/04 09:49:51
Bob	s.txt	read	6259	/home/tester/Downloads	2018/07/04 13:05:49
Alice	s.txt	cat	7261	/home/tester/Downloads	2018/07/04 15:22:12
Alice	s.txt	read	7758	/home/tester/Downloads	2018/07/04 01:17:38

2. **ความผิดปกติจากการตรวจสอบไฟล์ข้อมูลบันทึกเหตุการณ์ในการกระทำของผู้ใช้งาน**

จากตาราง 3 แสดงเหตุการณ์ที่เป็นการกระทำของผู้ใช้งานที่อาจกส่งผลให้เกิดความเสียหาย ในตาราง 3 แถวที่ 4 แสดงถึงการแก้ไขข้อมูลในไฟล์ข้อมูล s.txt โดยจะตรวจสอบได้จาก คอลัมน์ที่ 5 แถวที่ 4 ซึ่งจะแสดงขนาดของไฟล์ข้อมูล และ คอลัมน์ที่ 9 แถวที่ 4 ซึ่งแสดงข้อมูลเวลาที่มีการแก้ไข ดังนั้นหากมีการค้นหาตรวจสอบผู้รับผิดชอบที่ทำการแก้ไขข้อมูลก็จะชี้ไปที่ Alice

ในตาราง 3 แถวที่ 5 แสดงถึงมีการสร้างไฟล์ข้อมูลสำรองของไฟล์ข้อมูล s.txt ซึ่งสามารถอ่านรายละเอียดเกี่ยวกับการสำรองไฟล์ข้อมูล ที่หัวข้อเรื่อง โครงสร้างข้อมูลไอโหนด ข้อ 6 เรื่อง i_nlink โดยจะตรวจสอบได้จาก คอลัมน์ที่ 4 แถวที่ 5 ซึ่งจะแสดงจำนวนของไฟล์ข้อมูลสำรอง และ คอลัมน์ที่ 10 แถวที่ 5 ซึ่งแสดงข้อมูลเวลาที่มีการสร้างไฟล์ข้อมูลสำรอง ดังนั้นหากมีการค้นหาตรวจสอบผู้รับผิดชอบที่ได้สร้างไฟล์ข้อมูลสำรองก็จะชี้ไปที่ Bob

ในตาราง 3 แถวที่ 6 แสดงถึงมีเปลี่ยนแปลงสิทธิการใช้งานของไฟล์ข้อมูล s.txt ซึ่งสามารถอ่านรายละเอียดเกี่ยวกับสิทธิการใช้งานของไฟล์ข้อมูลที่หัวข้อเรื่อง โครงสร้างข้อมูลไอ

โหนด ข้อ 2 เรื่อง i_mode โดยจะตรวจสอบได้จาก คอลัมน์ที่ 6 แถวที่ 6 ซึ่งจะแสดงสิทธิการใช้งานไฟล์ข้อมูล และ คอลัมน์ที่ 10 แถวที่ 5 ซึ่งแสดงข้อมูลเวลาที่มีการเปลี่ยนแปลงสิทธิการใช้งานไฟล์ข้อมูล ดังนั้นหากมีการค้นหาตรวจสอบผู้รับผิดชอบที่ได้สร้างไฟล์ข้อมูลสำรองก็จะชี้ไปที่ Alice

ในตาราง 3 แถวที่ 7 แสดงถึงมีการเปลี่ยนความเป็นเจ้าของไฟล์ข้อมูล โดยจะตรวจสอบได้จาก คอลัมน์ที่ 7 แถวที่ 7 ซึ่งจะแสดงหมายเลขประจำตัวของผู้เป็นเจ้าของไฟล์ข้อมูล และ คอลัมน์ที่ 10 แถวที่ 7 ซึ่งแสดงข้อมูลเวลาที่มีการเปลี่ยนแปลงความเป็นเจ้าของไฟล์ข้อมูล ดังนั้นหากมีการค้นหาตรวจสอบผู้รับผิดชอบที่ได้ทำการเปลี่ยนแปลงความเป็นเจ้าของไฟล์ข้อมูลก็จะชี้ไปที่ Alice

ในตาราง 3 แถวที่ 8 แสดงถึงมีการเปลี่ยนกลุ่มของไฟล์ข้อมูล โดยจะตรวจสอบได้จาก คอลัมน์ที่ 8 แถวที่ 8 ซึ่งจะแสดงหมายเลขประจำกลุ่มของไฟล์ข้อมูล และ คอลัมน์ที่ 10 แถวที่ 8 ซึ่งแสดงข้อมูลเวลาที่มีการเปลี่ยนแปลงกลุ่มของไฟล์ข้อมูล ดังนั้นหากมีการค้นหาตรวจสอบผู้รับผิดชอบที่ได้ทำการเปลี่ยนแปลงกลุ่มของไฟล์ข้อมูลก็จะชี้ไปที่ Bob

ตาราง 3 แสดงข้อมูลความผิดปกติจากไฟล์ข้อมูลบันทึกเหตุการณ์

user	File	Pname	Link	Size	Mode	Uid	Gid	Mtime	Ctime
Alice	s.txt	read	1	17	rwxrwxr--	1001	1001	08:51:48	08:51:48
Bob	s.txt	cat	1	17	rwxrwxr--	1001	1001	08:51:48	08:51:48
Alice	s.txt	gedit	1	21	rwxrwxr--	1001	1001	09:49:51	08:51:48
Bob	s.txt	ln	2	21	rwxrwxr--	1001	1001	09:49:51	10:23:14
Alice	s.txt	read	2	21	rwxrwxrwx	1001	1001	09:49:51	11:04:17
Alice	s.txt	chown	2	21	rwxrwxrwx	1002	1001	09:49:51	14:02:23
Bob	s.txt	chgrp	2	21	rwxrwxrwx	1002	1002	09:49:51	14:58:43
Alice	s.txt	read	2	21	rwxrwxrwx	1002	1001	09:49:51	09:49:51

3. การตรวจสอบการลบไฟล์ข้อมูลและสร้างใหม่ไฟล์ข้อมูลใหม่ทดแทน

จากตาราง 4 จะแสดงเหตุการณ์เมื่อมีการลบไฟล์ข้อมูลซึ่งระบบบันทึกเหตุการณ์จะบันทึกเหตุการณ์เกี่ยวกับการลบไฟล์ดังตาราง 4 คอลัมน์ที่ 3 แถวที่ 3 ซึ่งจะแสดงโปรเซสที่ใช้ลบข้อมูล และ คอลัมน์ที่ 6 แถวที่ 3 อีกทั้งยังสามารถบอกถึงไฟล์ที่ถูกสร้างขึ้นใหม่ได้แม้จะมีชื่อเหมือนเดิม เมื่อ Bob คำสั่งสร้างไฟล์ข้อมูล s.txt ขึ้นมาใหม่ โดยจงใจใช้ชื่อไฟล์ข้อมูลเดิมที่ทำการลบไป ซึ่งจะเห็นได้ว่าไฟล์ข้อมูลใหม่ที่ได้ทำการสร้างขึ้นมานั้นจะมีค่าข้อมูลของ Inode ที่ไม่

เหมือนเดิม ดังตาราง 4 คอลัมน์ที่ 4 แถวที่ 4 ซึ่งคุณสมบัติเกี่ยวกับ Inode ถูกกล่าวไว้ที่หัวข้อเรื่อง โครงสร้างข้อมูลไอโหนด ข้อ 5 เรื่อง i_no

ตาราง 4 แสดงข้อมูลความผิดปกติของการลบและสร้างไฟล์ข้อมูลใหม่ทดแทน

user	File	Pname	Inode	Size	Atime	Mtime	Ctime
Alice	s.txt	read	176738	21	2018/07/04 09:49:51	2018/07/04 08:48:57	2018/07/04 08:48:57
Bob	s.txt	rm	176738	21	2018/07/04 13:05:49	2018/07/04 08:48:57	2018/07/04 08:48:57
Bob	s.txt	touch	176963	0	2018/07/04 15:22:12	2018/07/04 15:22:12	2018/07/04 15:22:12
Alice	s.txt	read	176963	0	2018/08/04 10:17:38	2018/07/04 15:22:12	2018/07/04 15:22:12

ในส่วนของการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์นั้น จากการทดลองนั้นจะเห็นได้ว่าแต่ละช่วงเวลาของ sleeping time มีผลต่อประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการ โดยถ้าค่าที่กำหนดไว้ของ sleeping time มีค่าน้อยประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการก็จะน้อยลง และเมื่อมีการเพิ่มค่าที่กำหนดไว้ของ sleeping time ประสิทธิภาพการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการก็จะเพิ่มมากขึ้น นอกจากนี้ ค่า sleeping time มีผลต่อความแม่นยำของระบบบันทึกเหตุการณ์ยิ่งมีค่าน้อยค่าความแม่นยำของระบบบันทึกเหตุการณ์จะแม่นยำมากขึ้นตามผลการทดลองในหัวข้อเรื่อง ผลจากการวัดและทดสอบประสิทธิภาพการทำงานของผู้ใช้บริการต่อการใช้งานระบบบันทึกเหตุการณ์ ดังนั้นจะเห็นได้ว่าประสิทธิภาพของระบบบันทึกเหตุการณ์ และการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้บริการนั้นมีผลตรงกันข้ามกัน ผู้วิจัยจึงเสนอแนวทางเพื่อให้ผู้ใช้บริการเลือกระหว่างประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้บริการและความแม่นยำของระบบบันทึกเหตุการณ์ดังนี้

ตาราง 5 แสดงรูปแบบการใช้งานระบบบันทึกเหตุการณ์นำเสนอให้กับผู้ใช้บริการ

ประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้บริการ	ประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์
~95%	~55%
~85%	~75%
~75%	~100%

จากข้อเสนอที่ผู้วิจัยนำเสนอให้เห็นดังตาราง 5 นั้น หากผู้ใช้บริการต้องการให้การทำงานทั้งสองด้านมีประสิทธิภาพให้ดีใกล้เคียงกันนั้นสามารถทำได้โดยการเพิ่มความสามารถของการประมวลผลให้สูงยิ่งขึ้น หรือจากข้อสังเกตในการทดลองโพสเซสที่เกิดขึ้นมีระยะเวลาน้อยมาก แต่ในการนำไปใช้งานจริงการเข้าถึงไฟล์ข้อมูลของผู้ใช้บริการอาจมีระยะเวลาที่มากกว่าที่ใช้ในการทดลองจึงเป็นไปได้ว่าสามารถเลือกใช้ประสิทธิภาพในการทำงานของเครื่องคอมพิวเตอร์ได้เต็มประสิทธิภาพ เช่น ในการทดลองประสิทธิภาพของเครื่องคอมพิวเตอร์ของผู้ใช้บริการจะมีประสิทธิภาพดีที่สุดที่ ระบบบันทึกเหตุการณ์ทำงานในช่วงเวลา sleeping time มากกว่า 2500 microsecond ดังนั้นหากเวลาการทำงานของโพรเซสที่น้อยที่สุดของผู้ใช้บริการ ก็สามารถนำมาใช้เป็นค่าที่กำหนดไว้ให้กับ sleeping time จะส่งผลให้ผู้ใช้บริการได้รับประโยชน์สูงสุดในการใช้งานทั้งระบบบันทึกเหตุการณ์และการทำงานของผู้ใช้บริการเอง

ข้อเสนอแนะ

จากการทดลองการปรับปรุงประสิทธิภาพของระบบบันทึกเหตุการณ์จากวิทยานิพนธ์นี้ ผู้วิจัยจึงมีข้อเสนอแนะในการพัฒนาระบบบันทึกเหตุการณ์ให้มีประสิทธิภาพในการทำงานและการนำไปใช้งานได้ดียิ่งขึ้นในอนาคต

1. **วิธีการการตรวจสอบไฟล์ข้อมูลบันทึกเหตุการณ์**

การเก็บผลลัพธ์จากการทำงานของระบบบันทึกเหตุการณ์ ยังไม่มีวิธีการตรวจสอบเพื่อหาเหตุการณ์ที่ผิดปกติ และเนื่องจากว่าข้อมูลเหล่านี้มีจำนวนมากการตรวจสอบโดยมนุษย์อาจจะไม่มีประสิทธิภาพเพียงพอ หากมีวิธีการตรวจสอบที่รวดเร็วและมีประสิทธิภาพที่ใช้ค้นหาการกระทำที่ผิดปกติ จะส่งผลให้การทำงานของระบบบันทึกเหตุการณ์ทำงานได้สมบูรณ์มากยิ่งขึ้น ในด้านวิธีการตรวจสอบเพื่อหาผู้กระทำผิดจากข้อมูลที่ได้ไฟล์ข้อมูลบันทึกเหตุการณ์ อาจใช้กระบวนการของ Machine Learning หรือ ปัญญาประดิษฐ์

2. วิธีการเพิ่มข้อมูลทางด้านเครือข่ายคอมพิวเตอร์ให้กับไฟล์ข้อมูลบันทึกเหตุการณ์

ข้อมูลที่ใช้ในการตรวจสอบผู้รับผิดชอบต่อการกระทำนั้นหากมีข้อมูลที่เกี่ยวกับเครือข่ายคอมพิวเตอร์ เช่นหมายเลขไอพี (IP Address) ชื่อเครื่อง ผู้ให้บริการอินเทอร์เน็ต รหัสประจำตัวของอุปกรณ์ (MAC Address) และเส้นทางการเดินทางของข้อมูล จะทำให้ความสามารถในการค้นหาและตรวจสอบหาผู้กระทำผิดมีประสิทธิภาพมากขึ้น

3. การนำไปประยุกต์ใช้กับโอเพนสแต็ก

โอเพนสแต็ก (OpernStack) เป็นชุดระบบซอฟต์แวร์ Open Source ใช้สร้างระบบการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐาน หรือ Infrastructure as a Service Cloud (IaaS) มีผู้นิยมใช้งาน มีการเติบโต และมีบทบาทในตลาดขององค์กรที่ให้บริการคลาวด์อย่างมากในปัจจุบัน หากมีการนำระบบบันทึกเหตุการณ์ไปประยุกต์ใช้งานบน OpernStack จะทำให้มีความปลอดภัยสูงมากขึ้น

บรรณานุกรม

Achar, S. (2022). Cloud Computing Forensics. *International Journal of Computer Engineering and Technology, 13*(3), 1-10.

Adjepon-Yamoah, & Ebo, D. (2022). Cloud Accountability Method: Towards Accountable Cloud Service-Level Agreements. In *Sixth International Congress on Information and Communication Technology: ICICT 2021* (p. 439-458). Singapore: Springer Singapore.

Amazon Web Services. (2022). What is cloud computing? Retrieved October 31, 2022, from https://aws.amazon.com/th/what-is-cloud-computing/

Auxsorn, T., Wongthai, W., Phoka, T., & Jaiboon, W. (2020). Performance Considerations of a Logging System Simultaneously with a Customer Virtual Machine in Infrastructure as a Service Cloud. In *Information Science and Applications: ICISA 2019* (p. 285-296). Singapore: Springer Singapore.

Auxsorn, T., Wongthai, W., Porka, T., & Jaiboon, W. (2020). The accuracy measurement of logging systems on different hardware environments in infrastructure as a service cloud. ICIC Express Letters, Part B: Applications. *An International Journal of Research and Surveys, 11*(5), 427-437.

Avudaiyappan, K., & Abdallah, M. (2014). *Systems and methods for accessing a unified translation lookaside buffer. U.S. Patent No. 8,930,674*. Washington, DC: U.S. Patent and Trademark Office.

Avudaiyappan, K., and Abdallah, M. (2017). Systems and methods for supporting a plurality of load and store accesses of a cache. U.S. Patent No. 9,720,839. Washington, DC: U.S. Patent and Trademark Office.

Bhardwaj, A., & Rama Krishna, C. (2022). A container-based technique to improve virtual machine migration in cloud computing. *IETE Journal of Research, 68*(1), 401-416.

BISWAS, S. A. (2017). Guide to The Linux Top Command. Retrieved May 1, 2020, from www.booleanworld.com/guide-linux-top-command

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

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.

CSA. (2019). *Top Threats to Cloud Computing: Egregious Eleven in 2019.* The cloud Security Alliance,(CSA), Tech.REP.

Dordevic, B., Timcenko, V., Sakic, D., & Davidovic, N. (2022). File system performance for type-1 hypervisors on the Xen and VMware ESXi. In *2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH)* (p. 1-6). Jahorina: IEEE.

Fareghzadeh, N. (2022). An architecture supervisor scheme toward performance differentiation and optimization in cloud systems. *The Journal of Supercomputing, 78*(1), 1532-1563.

Gartner. (2022). Gartner Top Strategic Technology Trends for 2022. Retrieved September 9, 2022, from https://www.gartner.com/en/newsroom/press-releases/2021-10-18-gartner-identifies-the-top-strategic-technology-trends-for-2022

Homscheid, D. (2020). *The Linux Kernel Project. Firm-Sponsored Developers in Open Source Software Projects: A Social Capital Perspective.* Koblenz, Germany: Springer Gabler.

Ko, R. K. (2014). Data accountability in cloud system Security, Privacy and Trust in Cloud Systems In S. Nepal & M. Pathan (Eds.). In *Security, Privacy and Trust in Cloud Systems* (pp. 211-238). Berlin, Heidelberg: Springer.

Kopytov, A. (2017). Sy*sbench.* Retrieved March 2, 2019, from https://github.com/akopytov/sysbench

Lal, A., Prasad, A., Kumar, A., & Kumar, S. (2022). IData Exfiltration: Preventive and Detective Countermeasures. In *International Conference on Innovative Computing & Communication (ICICC)* Delhi, India: https://ssrn.com/abstract=4031852

Lata, S., & Singh, D. (2022). Intrusion detection system in cloud environment: Literature

survey & future research directions. *International Journal of Information Management Data Insights, 2*(2), 100134. doi: https://doi.org/10.1016/j.jjimei.2022.100134

Lee, J., de Guzman, M. C., Wang, J., Gupta, M., & Rao, H. R. (2022). Investigating perceptions about risk of data breaches in financial institutions: A routine activity-approach. *Computers & Security, 121*, 102832. doi: https://doi.org/10.1016/j.cose.2022.102832

LibVMi. (n.d.). Access a VM's State. From Outside of the VM. Retrieved May 1, 2020, from https://libvmi.com/

Mayuranathan, M., Saravanan, S. K., Muthusenthil, B., & Samydurai, A. (2022). An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique. *Advances in Engineering Software, 173*, 103236. doi: https://doi.org/10.1016/j.advengsoft.2022.103236

Melvin, A. A. R., Kathrine, G. J. W., Ilango, S. S., Vimal, S., Rho, S., Xiong, N. N., & Nam, Y. (2022). Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud. *Transactions on Emerging Telecommunications Technologies 33*(4), e4287.

Mishra, P., Pilli, E. S., Varadharajan, V., & Tupakula, U. (2017). Intrusion detection techniques in cloud environment: A survey. *Journal of Network and Computer Applications, 77*, 18-47.

Molyneaux, I. (2014). *The Art of Application Performance Testing From Strategy to Tools.* Sebastopol, CA: O'Reilly Media.

Nanni, D. D. (2014). *How to run program or process on specific CPU cores on Linux*. N.p.: National Institute of Standards and Technology.

Nguyen, T., Orenbach, M., & Atamli, A. (2022). Live system call trace reconstruction on Linux. *Forensic Science International: Digital Investigation, 42*, 301398.

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

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

Parra, G. D. L. T., Selvera, L., Khoury, J., Irizarry, H., Bou-Harb, E., & Rad, P. (2022). Interpretable federated transformer log learning for cloud threat forensics. In *Network and Distributed Systems Security (NDSS) Symposium* N.d.: NDSS. https://www.ndss-symposium.org/wp-content/uploads/2022-102-paper.pdf

Payne, B. D. (2012). *Simplifying virtual machine introspection using LibVMI (No. SAND2012-7818)*. Livermore, CA: Sandia National Laboratories.

Popa, N. M., and Oprescu, A. (2019). A data-centric approach to distributed tracing. In *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (p. 209-216). Sydney, NSW, Australia: IEEE.

Spener, M. (2022). Naive Introspection in the Philosophy of Perception. *Review of Philosophy and Psychology, 13*(1), 29-45.

Thyagaturu, A. S., Shantharama, P., Nasrallah, A., & Reisslein, M. (2022). Operating Systems and Hypervisors for Network Functions: A Survey of Enabling Technologies and Research Studies. *IEEE Access, 10*, 79825-79873. doi: 10.1109/ACCESS.2022.3194913

Tiwari, S. K., Neogi, S. G., Mishra, A., Singh, S., Khan, H., Kispotta, S., & Purohit, C. (2022). Trusted Infrastructure Design for Secure Virtualization in Cloud Computing: A Review. *Journal of University of Shanghai for Science and Technology, 24*(9), 1-13.

Wongthai, W., & Moorsel, A. v. (2016). Performance measurement of logging systems in infrastructure as a service cloud. *ICIC Express Letters, 10*(2), 347-354.

Wongthai, W., & Van Moorsel, A. (2016). Quality analysis of logging system components in the cloud. *Lecture Notes in Electrical Engineering, 376*, 651-662.

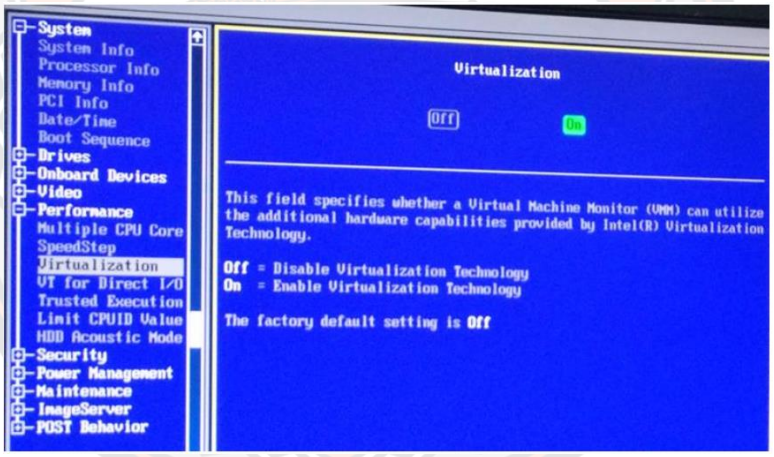
ภาคผนวก

ภาคผนวก ก ขั้นตอนการสร้างการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐาน (Infrastructure as a Service Cloud : IaaS)

1. **วิธีการสร้าง**

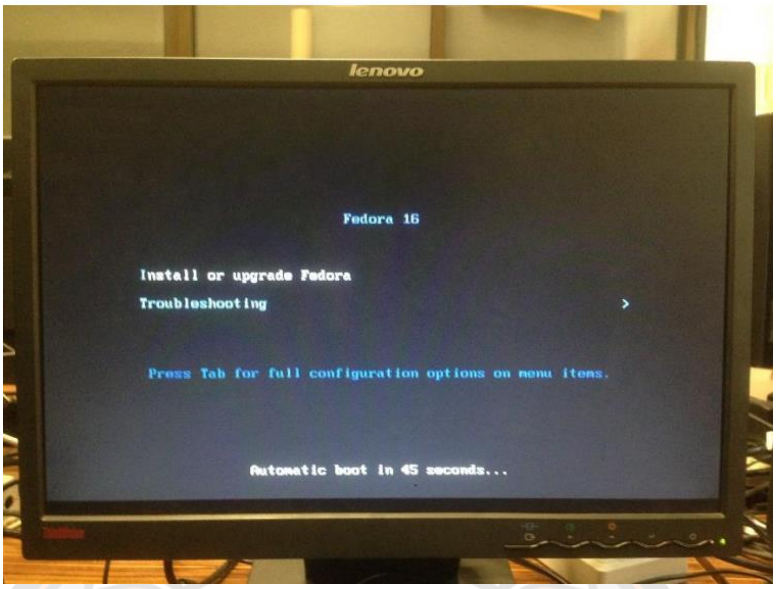
ขั้นตอนที่ 1 ติดตั้งระบบปฏิบัติการฟีโดร่า (Fedora) **เพื่อถูกใช้เป็นฐานสำหรับสร้างIaaS คลาวด์**

1.1 ให้ดำเนินการเปิดเครื่องคอมพิวเตอร์ที่ถูกใช้สำหรับสร้างคลาวด์จากนั้นระหว่างที่เครื่องคอมพิวเตอร์กำลังดำเนินการเริ่มบูทให้ กดปุ่ม F2 Del หรือ F12 บนคีย์บอร์ดตามเงื่อนไขในค่าไบออส (Bios) ของแต่ละยี่ห้อที่เป็นผู้สร้างเมนบอร์ด (Mainboard) เพื่อที่จะเข้าไปตั้งค่าไบออสเพื่อเปิดการใช้ฟังก์ชัน Virtualization ดังภาพ 30 ให้เลือกปุ่ม On แล้วกดคีย์ Enter แล้วให้ดำเนินการออกบันทึกค่าในไบออสและให้ออกจากระบบการทำงานของการตั้งค่าไบออสของเครื่องคอมพิวเตอร์



ภาพ 30 แสดงหน้าจอไบออสสำหรับเปิดใช้งานฟังก์ชัน Virtualization

1.2 ดำเนินการใส่แผ่นระบบปฏิบัติการฟีโดร่า (Fedora) เวอร์ชัน 16 แบบ 64 Bit ใส่ไว้ในช่องของตัวอ่าน DVD ในเครื่องคอมพิวเตอร์ ซึ่งจะถูกอ่านข้อมูลแล้วแสดงผลลัพธ์สำหรับให้ทำการติดตั้งระบบปฏิบัติการฟีโดร่าดังภาพ 82 ให้ ทำการเลือกข้อความว่า Install or upgrade Fedora แล้วกดคีย์ Enter



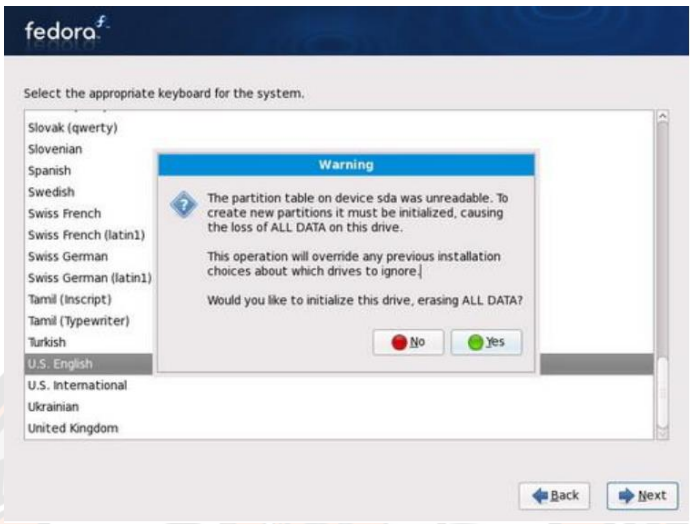
ภาพ 31 แสดงหน้าต่างการติดตั้งระบบปฏิบัติการฟีโดร่า

จากภาพ 31 แสดงหน้าต่างการติดตั้งระบบปฏิบัติการฟีโดร่า จากนั้นระบบจะทำการตรวจสอบแผ่นที่ใช้ในการติดตั้ง ว่าสามารถอ่านข้อมูลที่อยู่ในแผ่นได้หรือไม่ แต่ในที่นี้สามารถข้ามการตรวจสอบแผ่นได้โดยคลิกที่ปุ่ม "Skip" ดังภาพ 32

ภาพ 32 แสดงหน้าต่างการตรวจสอบข้อมูลในแผ่นระบบปฏิบัติการฟีโดร่า

1.3 จากนั้นแสดงหน้าต่างการติดตั้งระบบปฏิบัติการฟีโดร่าให้เลือกปุ่ม Next จากนั้นระบบจะให้เลือกภาษาที่ใช้ในการติดตั้งให้เลือกเป็นภาษา "English (English)" กดปุ่ม Next และเลือกคีย์บอร์ดเป็น "U.S. English" แล้วกดปุ่ม Next ดังภาพ 33 ซึ่งจะแสดงผลลัพธ์ที่เป็น

การแจ้ง234 เตือน (Warning) ว่าฮาร์ดดิสก์ที่ถูกติดตั้งระบบจะถูกจัดพาร์ท ชันใหม่โดยให้ทำการเลือก Yes แล้วจะเข้าสู่การเลือกประเทศให้เลือก "Asia/Bangkok” ดังภาพ 34 แล้วกดปุ่ม Next

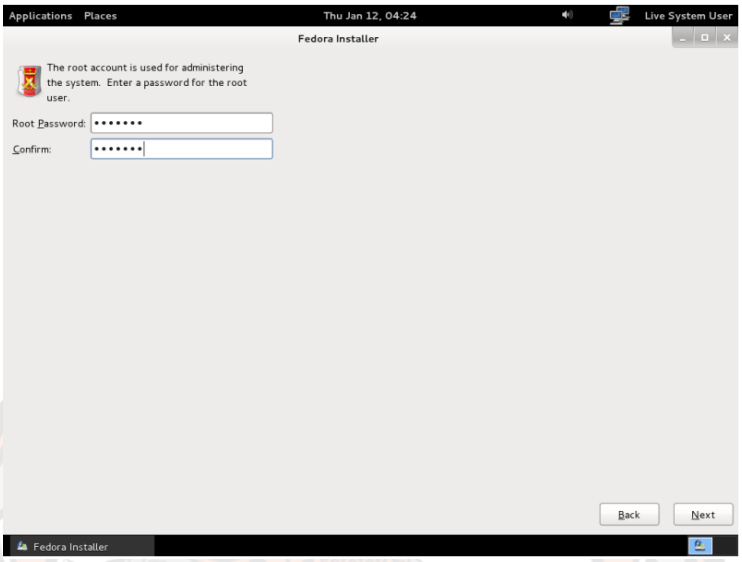


ภาพ 33 แสดงการแจ้งเตือนให้สร้างจัดพาร์ติชันสำหรับระบบปฏิบัติการฟีโดร่าใหม่

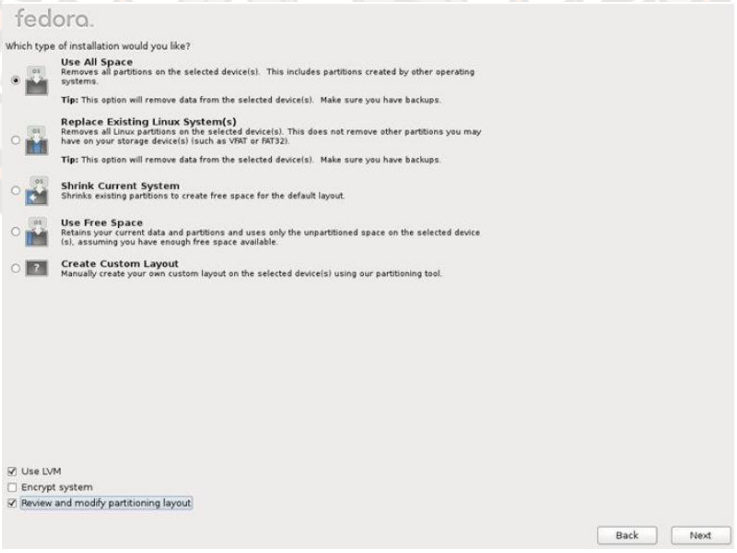
ภาพ 34 แสดงให้เลือกประเทศสำหรับติดตั้งระบบปฏิบัติการฟีโดร่าใหม่

1.4 ทำการตั้งรหัสผ่านให้กับผู้ใช้ Root ซึ่งเป็นผู้ดูแลระบบปฏิบัติการฟีโดร่าดังภาพ 35 จากนั้นจะแสดงหน้าต่างให้การแบ่งพาร์ติชันของระบบให้เลือกเป็น “Use All Space” และคลิก

ปุ่ม “Use LVM” และ “Review and modify partitioning layout” ดังภาพ 36 เพื่อจัดการพื้นที่ของหน่วยความจำสำรอง (Hard disk) สำหรับไว้สร้างคอมพิวเตอร์เสมือน



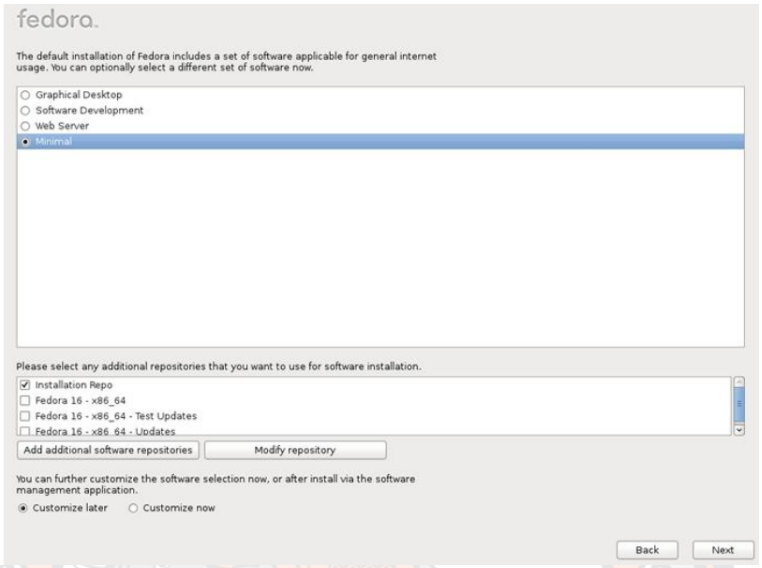
ภาพ 35 แสดงการตั้งรหัสผ่านของผู้ดูแลระบบปฏิบัติการฟีโดร่า



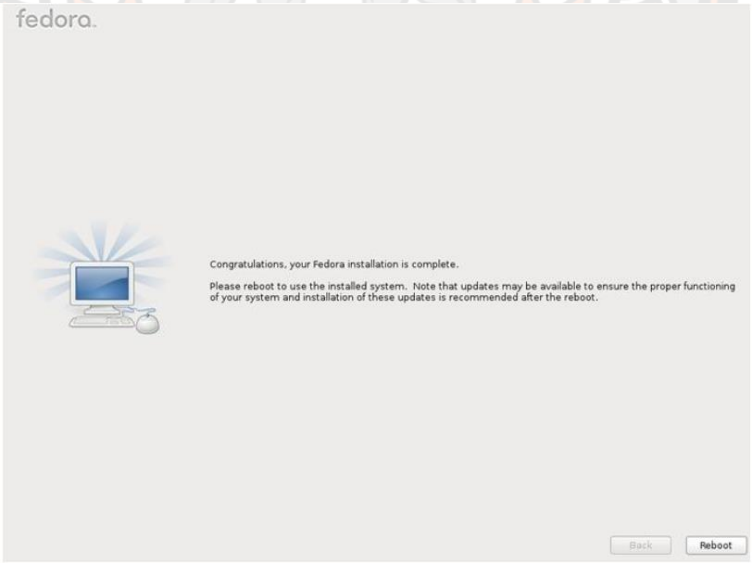
ภาพ 36 แสดงการสร้างพาร์ติชันสำรับระบบปฏิบัติการฟีโดร่า

1.5 จากภาพ 36 เมื่อกดปุ่ม Next ระบบจะทำการล้างข้อมูลและจะถูกกำหนดค่าตามที่กำหนดขึ้นมาใหม่ จากนั้นจะเข้าสู่การติดตั้งซอฟต์แวร์ต่าง ๆ ที่จำเป็นสำหรับการสร้าง

คลาวด์ดังภาพ 37 โดยให้เลือกเป็น “Graphical Desktop” แล้วกดปุ่ม Next เพื่อเข้าสู่การติดตั้งระบบปฏิบัติจนเสร็จสมบูรณ์ดังภาพ 38 แล้วดำเนินการกดปุ่ม Reboot

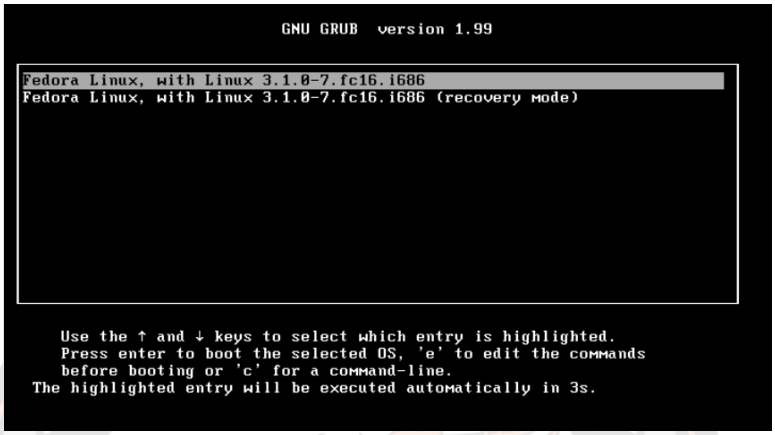


ภาพ 37 แสดงการเลือกซอฟต์แวร์ต่าง ๆ ที่จำเป็นสำหรับติดตั้งระบบปฏิบัติการ

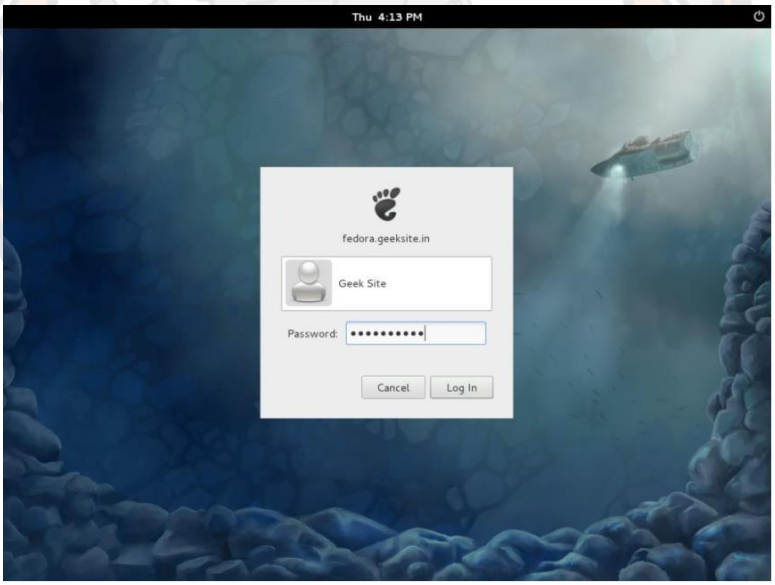


ภาพ 38 แสดงการติดตั้งระบบปฏิบัติการการฟีโดร่าเสร็จสมบูรณ์

1.6 ภาพ 39 และภาพ 40 แสดงหน้าต่างของระบบปฏิบัติการฟีโดร่าหลังจาก Reboot เครื่องคอมพิวเตอร์และการสร้างบัญชีผู้ใช้งานที่เป็นผู้ดูแลระบบ (Administrator) ที่สามารถเป็นผู้มีสิทธิ์ในการใช้งาน su เป็นสิทธิ์ของ root ได้ดังภาพ 41



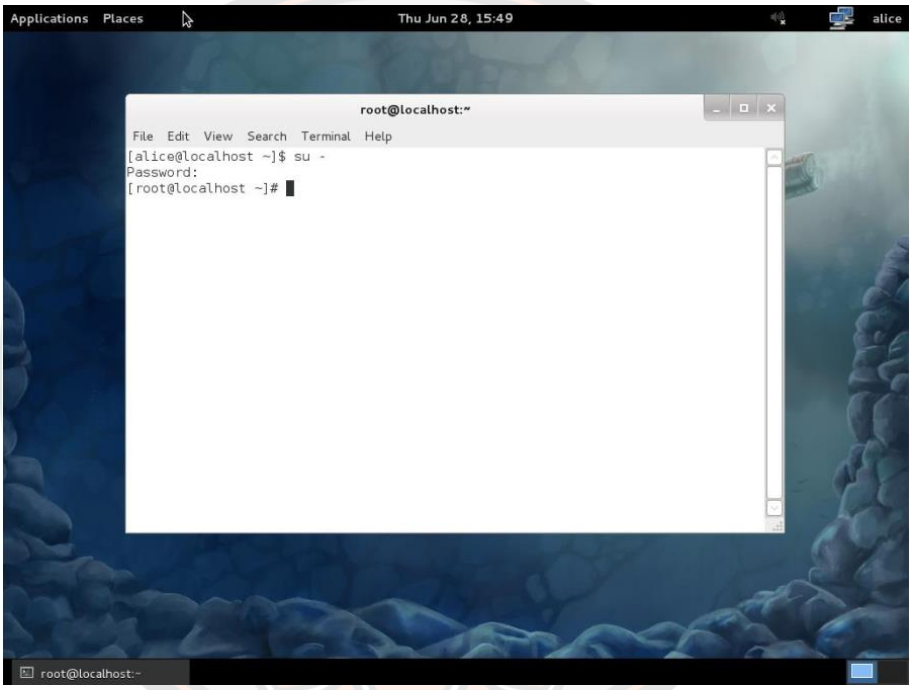
ภาพ 39 แสดงหน้าต่างเมนูของสู่ระบบปฏิบัติการฟีโดร่า



ภาพ 40 แสดงหน้าต่างสาหรับบัญชีผู้ใช้งานที่เป็นผู้ดูแลระบบ

ขั้นตอนที่ 2 ติดตั้งซอฟต์แวร์ hypervisor เพื่อถูกใช้สำหรับสร้าง IaaS คลาวด์

2.1 เมื่อทำการเปิดระบบเครื่องคอมพิวเตอร์ที่ถูกใช้สำหรับเป็นพื้นฐานของการสร้าง IaaS คลาวด์แล้วจะเข้าสู่หน้าจอการติดตั้งซอฟต์แวร์ต่าง ๆ เพื่อดำเนินการติดตั้ง ซอฟต์แวร์ hypervisor ซึ่งเป็นซอฟต์แวร์ที่ทำให้เครื่องคอมพิวเตอร์หนึ่งเครื่องสามารถรันระบบปฏิบัติการได้มากกว่าหนึ่งระบบในเครื่องเดียวกัน ดังคำสั่งการติดตั้งที่ถูกเรียกใช้งานดังในภาพ 41 และคำสั่งเรียกใช้งาน ตามคู่มือในเว็บไซต์ https://wiki.xen.org/wiki/Fedora_Host_Installation ดังสรุปต่อจากนี้



ภาพ 41 แสดงหน้าต่าง Terminal สำหรับพิมพ์คำสั่งต่าง ๆ

[root@logger-server ~] # yum -y update

[root@logger-server ~] # yum -y install @xfce

[root@logger-server ~] # yum -y groupinstall "Development Tools"

[root@logger-server ~] # reboot

[root@logger-server ~] # yum -y install xen

[root@logger-server ~] # yum install debootstrap perl-Text-Template perl-Config-IniFiles perl-FileSlurp perl-File-Which perl-Data-Dumper

[root@logger-server ~] # yum install libvirt-daemon-driver-xen python-virtinst libvirt-daemon-confignetwork libvirt-daemon-driver-network virt-manager virt-viewer tigervnc libvirt virtmanager

[root@logger-server ~] # lvcreate -nf16 -L50G /dev/VolGroup

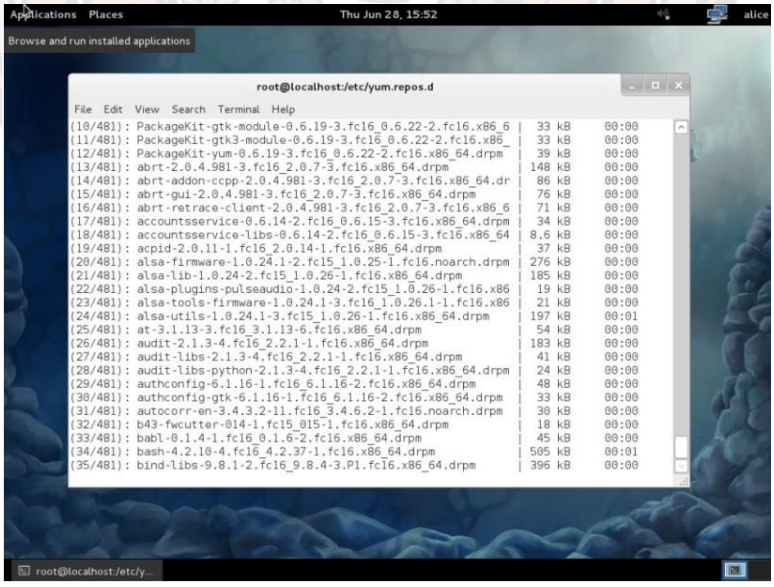
[root@logger-server ~] # yum -y install xen xen-hypervisor xen-libs xen-runtime virt-manager dejavu* xorg-x11-xauth

[root@logger-server ~] # yum -y install libvirt-daemon-driver-network libvirt-daemon-driver-storage libvirt-daemon-xen install libvirt-daemon-config-network libvirt-daemon-config-nwfilter

[root@logger-server ~] # chkconfig libvirtd on

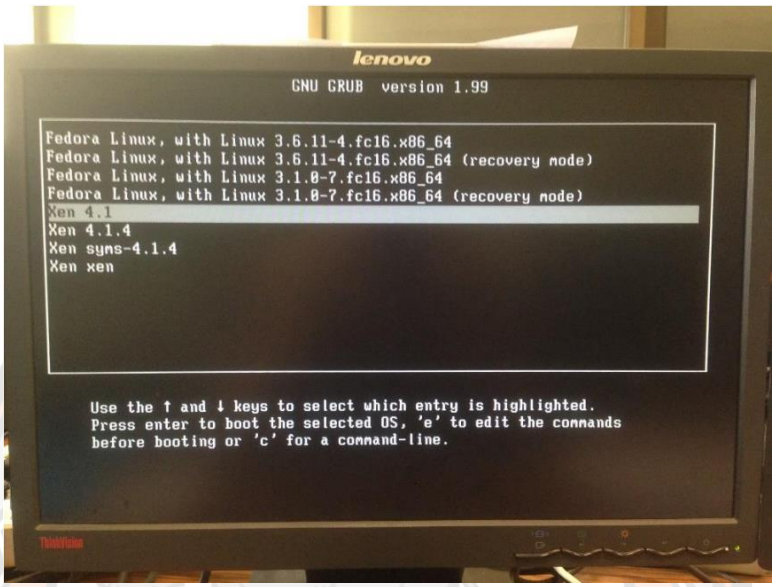
[root@logger-server ~] # service libvirtd start

2.2 จากภาพ 41 แสดงหน้าต่าง Terminal สำหรับพิมพ์คำสั่งต่าง ๆ โดยให้พิมพ์ su แล้วกด Enter จากนั้นให้ใส่รหัสผ่านของ root จากนั้นผู้ซึ่งเป็นผู้ดูแลระบบจะดำเนินการติดตั้งซอฟต์แวร์ hypervisor ในที่นี้จะใช้ซอฟต์แวร์ xen เป็นตัวบริหารจัดการคอมพิวเตอร์เสมือนหรือ domU ดังการติดตั้งด้วยคำสั่งต่าง ๆ ข้างต้นและตัวอย่างของผลลัพธ์ที่ได้ในภาพ 42 ต่อจากนี้

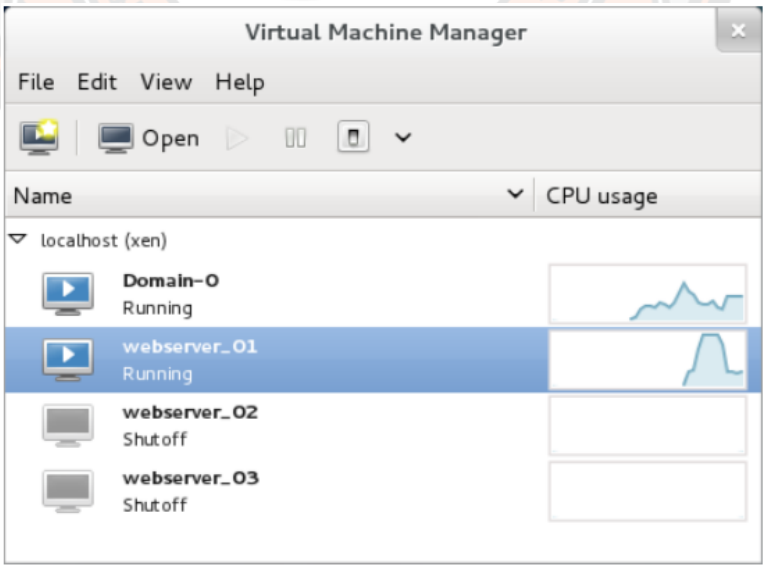


ภาพ 42 แสดงตัวอย่างของผลลัพธ์ที่ได้จากการใช้คำสั่งต่าง ๆ สำหรับติดตั้งซอฟต์แวร์ xen

2.3 จากภาพ 42 แสดงตัวอย่างของผลลัพธ์ที่ได้จากการใช้คำสั่งต่าง ๆ ภาพ 43 แสดงการติดตั้งซอฟต์แวร์ xen สำเร็จสมบูรณ์ และในภาพ 44 ที่แสดงหน้าต่างของระบบบริหารจัดการคอมพิวเตอร์เสมือน



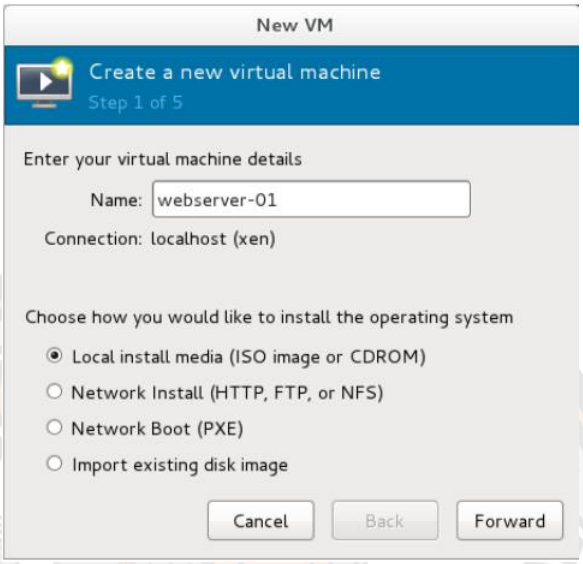
ภาพ 43 แสดงหน้าต่างการติดตั้งซอฟต์แวร์ xen สำเร็จสมบูรณ์



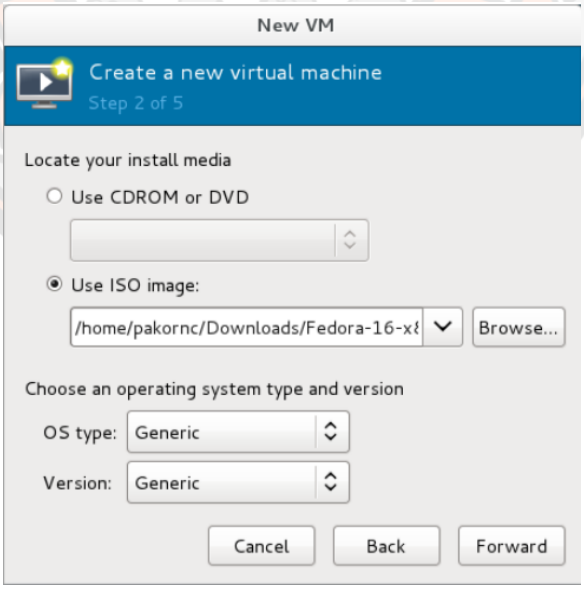
ภาพ 44 แสดงหน้าต่างของระบบบริหารจัดการคอมพิวเตอร์เสมือน (Virtual Machine Manager)

ขั้นตอนที่ 3 ทำการสร้าง domU

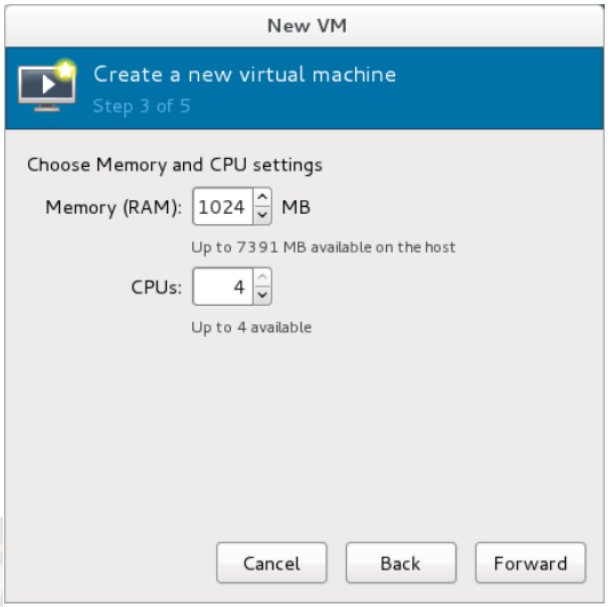
3.1 การสร้างคอมพิวเตอร์เสมือนหรือ domU ดังภาพ 45 ภาพ 46 ภาพ 47 และ ภาพ 48 ต่อจากนี้



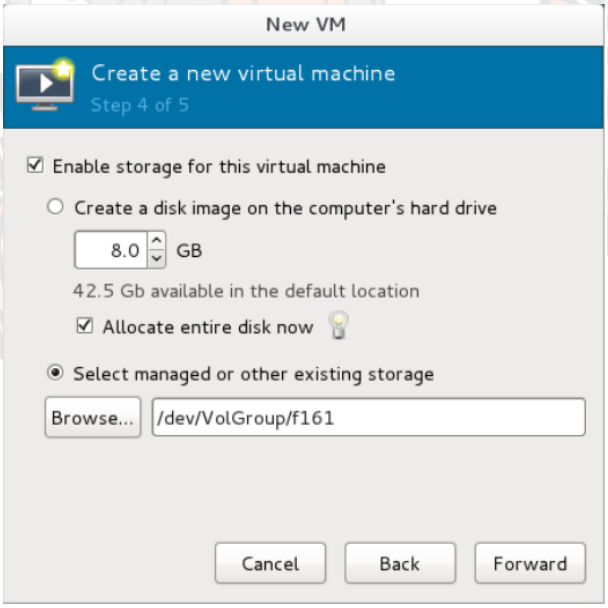
ภาพ 45 แสดงหน้าต่างการสร้างคอมพิวเตอร์เสมือน (domU)



ภาพ 46 แสดงหน้าต่างการใส่ไฟล์นามสกุล ISO **เพื่อติดตั้งระบบปฏิบัติการฟีโดร่าบน** domU

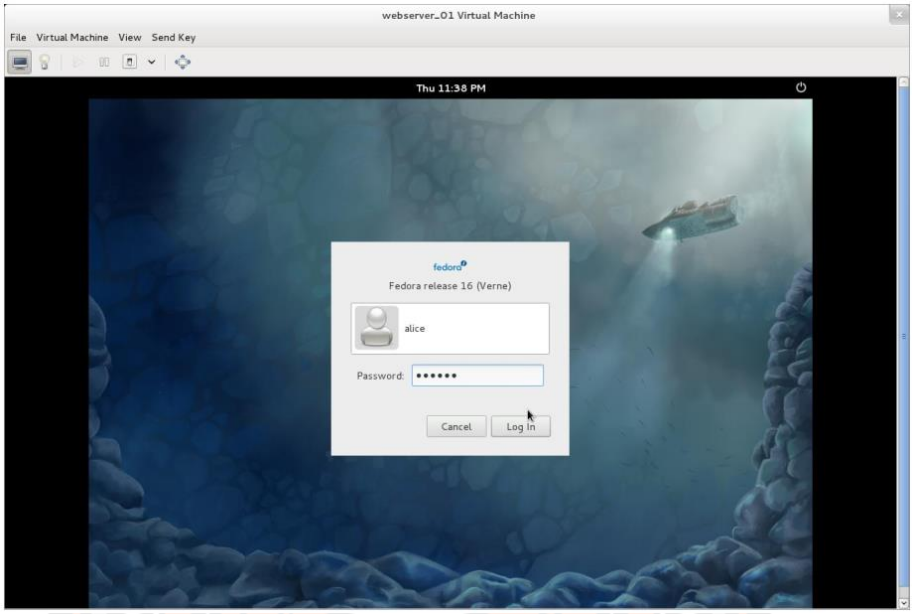


ภาพ 47 แสดงหน้าต่างการเลือกหน่วยความจำหลักและแกนประมวลผล (core) **สำหรับ** domU



ภาพ 48 แสดงหน้าต่างการกำหนดพื้นที่และแหล่งที่สำหรับเก็บ domU

3.2 จากภาพ 45 ภาพ 46 ภาพ 47 และ ภาพ 48 ดังกล่าวเป็นการกำหนดค่าเริ่มต้นของการตั้งค่าของคอมพิวเตอร์เสมือนหรือ domU จำนวน 1 เครื่องที่ชื่อว่า webserver_01 ดังที่แสดงไว้ในภาพ 44 จากภาพ 48 แสดงหน้าต่างการกำหนดพื้นที่ว่างของฮาร์ดดิกส์สำหรับ domU เมื่อกดปุ่ม Forward แล้ว domU จะถูกติดตั้งระบบปฏิบัติการจนเสร็จดังภาพ 49 โดยกำหนดให้บัญชีผู้ใช้ที่เป็นผู้ดูแลระบบมีชื่อว่า alice และดำเนินการติดตั้งซอฟต์แวร์ต่าง ๆ ที่เกี่ยวข้องกับเว็บเซิร์ฟเวอร์ดังคำสั่งการติดตั้งดังต่อจากนี้



ภาพ 49 แสดงการทำงานของระบบปฏิบัติภายใน domU

2. การทำงานของระบบบันทึกเหตูการณ์

การทำงานของระบบบันทึกเหตูการณ์ สามารถดูวีดีโอสาธิตการทำงานพร้อมคำอธิบายได้ที่ http://202.29.80.23/~godz/ Logger/ logger.mp4

ภาคผนวก ข ผลงานตีพิมพ์เผยแพร่ งานที่ 1.

ICIC Express Letters
Part B: Applications
Volume **11**, Number **5**, May **2020**

ICIC International ©2020 ISSN 2185-2766
pp. 427–437

THE ACCURACY MEASUREMENT OF LOGGING SYSTEMS ON DIFFERENT HARDWARE ENVIRONMENTS IN INFRASTRUCTURE AS A SERVICE CLOUD

Thongrob Auxsorn[1], Winai Wongthai[1,2,*], Thanathorn Porka[1] and Wichep Jaiboon[3]

[1]Department of Computer Science and Information Technology
[2]Research Center for Academic Excellence in Nonlinear Analysis and Optimization
Faculty of Science
Naresuan University
99 Moo 9, Tambon Tha Pho, Muang, Phitsanulok 65000, Thailand
*Corresponding author: winaiw@nu.ac.th

[3]Department of Computer Science
Faculty of Business and Public Administration
Nation University
444 Vajiravudh Damnoen Road, Lampang 52000, Thailand

Received November 2019; accepted February 2020

Abstract. *Infrastructure as a Service (IaaS) cloud provides virtual computing resources such as a Virtual Machine (VM) to a customer. Many companies desire to deploy this cloud; however, data security is an issue. The issue affects the reliability of a customer and provider, such as when hackers were able to access a data or critical file of a customer. Thus, researchers introduce a logging system to mitigate the risks associated with this issue. A logging system is in a host system and can capture some incidents that appeared in a customer VM. Then, the system stores the captured data in a log file to be used as evidence to mitigate the risks. This paper focuses on examining the accuracy of our previous logging system. This is done by varying the numbers of CPU cores and the size of the main memory of both the host system and targeted VM of a customer. The contributions of this paper are the following. Firstly, we fully introduced the method of varying the sizes of main memory and the numbers of CPU cores of both a host system of the logging system and targeted VM. This method can facilitate the other types of the accuracy or performance measurement of a logging system. Secondly, we found and illustrated the inappropriate hardware configurations of the host system, the targeted customer VM, and the association of both. For example, to increase the CPU cores of the host system of the logging system will not away increase the accuracy of the logging system. Thirdly, we also found and illustrated the appropriate configurations. All these contributions can help in decreasing the cost, time, effort, and energy of a logging system development in an IaaS ecosystem. To the best of our knowledge, there are no these three contributions in the literature.*

Keywords: Cloud security, Logging system, CPU cores, Main memory, Accuracy

1. **Introduction.** The cloud exploits virtualization technology to enable itself to store and process huge data [1], and is increasingly important for an IT ecosystem [2, 3]. An Infrastructure as a Service (IaaS) cloud is renting Virtual Machine (VM) resources by a cloud customer from a cloud provider. This cloud is gaining popularity in the IT ecosystem. It has been used for many types of applications. The expense of the cloud is calculated from how much the resources that were used. The cloud is a reasonable choice for organizations to apply in the IT departments. However, sensitive data that are processed outside the enterprise can cause risks to the data [4]. The cloud causes

DOI: 10.24507/icicelb.11.05.427

many security issues, which are a significant obstacle for its marketing situation [5]. To officially indicate the issues, Cloud Security Alliance (CSA) identifies the threats of this cloud [6, 7]. A single security method may not address the issues. Thus, traditional and new technologies and strategies should be combined to deal with the issues.

Many researchers are working on mitigating the risks associated with the threats. However, this paper focuses on examining the accuracy of our previous logging system. The system records some appropriate incidents in the cloud VM. Thus, whenever there is an issue, a log file can be retrieved from the logging system for analysis and investigation to mitigate the risks. The accuracy measurement is crucial for performance measurement of the logging system [8, 9, 10]. This paper focuses on examining the accuracy of the logging system. This is done by the method of varying the size of main memory and the numbers of CPU cores of both a host system of the logging system and targeted VM. The results were obtained, and the discussions and conclusion were made. **Research Gaps:** There is no the accuracy measurement of the logging system by the method of varying the CPU core numbers and main memory of both a host system of the system and targeted VM in the literature. Thus, we aim to perform the accuracy measurement of our previous logging system on different hardware environments of the host system in the IaaS cloud.

Summary of contributions: We introduced the method of varying the sizes of main memory and the numbers of CPU cores of both a host system of the logging system and targeted VM. This method can facilitate the other types of the accuracy or performance measurement of a logging system. The example is to measure the decreed performance of targeted customer VM, while running in the host system, not only to measure the logging system in the host system. Secondly, we found and illustrated the inappropriate hardware configurations of the host system, the targeted customer VM, and the association of both. For example, to increase the CPU cores of the host system of the logging system will not away increase the accuracy of the logging system. Thirdly, we also found and illustrated the appropriate configurations. For example, when the logging system works with core numbers such as 1 or 8, this may cause the best accuracy. The last two constitutions can be useful to ensure that the designers who design a logging system like our system can avoid the inappropriate hardware configurations, and can achieve the right configurations. This can help in reducing the cost, time, effort, and energy of a logging system development in an IaaS ecosystem. To the best of our knowledge, there are no these three contributions in the literature.

2. Background.

2.1. The IaaS and logging system architectures.

Figure 1 illustrates both an IaaS cloud architecture and our existing logging system architecture. Both architectures are adapted from our previous work [10] for the experimental purposes of this paper. Firstly, this section describes components of the IaaS cloud architecture using terminologies of Xen [11]. Then, Sections 2.2 and 2.3 will describe the existing logging system architecture. In Figure 1, the components of the IaaS architecture are shown as white boxes in the figure. This includes the hypervisor, dom0, domU, hw0, hwU, disk0, diskU, and memU. The box with number 2 in the figure is a hypervisor software that can create more than one VM in one physical machine. The topmost left box in the figure is the domain 0 (dom0) that is a manager of all the created VMs. The dom0 itself is also a VM and is launched by the hypervisor at the system booting duration. Any component with '0' at the end of its name indicates that it is physically managed and owned by a cloud provider. Similarly, 'U' indicates that the component is virtually managed and owned by a cloud customer. The dom0 exclusively accesses hw0 and manages all the created VMs or domUs. A user domain (domU) is a user VM that runs on top of the hypervisor, see the topmost right box of the figure. A domU is an IaaS cloud product that the provider

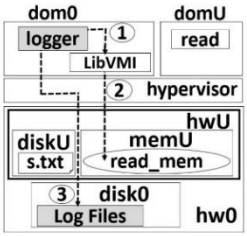


FIGURE 1. The IaaS and logging system architectures, adapted from [10]

offers to an IaaS customer. HwU is physically located in hw0 (which is owned by dom0 or by the provider and virtually owned by domU owner or by the customer), and it is domU's virtual hardware. Disk0 is a physical disk of the hw0, and diskU is a virtual disk of a domU. MemU is the virtual main memory of domU.

2.2. **The logging system types and the logger.** There are two types of logging systems. The first is an interception approach which is to intercept system calls of a domU. A system call provides the interface between a user application and kernel spaces [12]. For example, when our read process needs to read s.txt file, the read process needs to use three system calls: open, read, and close. Thus, the interception approach needs to intercept these three calls to capture desired logging data. The example of this approach is flogger [13]. It is a tool that can intercept file operations of a file that is in a customer VM. This tool can briefly check every file or folder accesses. It also can keep monitoring real-time generated system logs to save such monitored data as log files. The second approach is a memory introspection approach. The technical detail of this approach to obtain the logging data combines two factors: using memory introspection tools (such as LibVMI [14]) and the comprehension of the knowledge of Linux kernel data structures for virtual memory. In this paper, this approach is basically to access and capture logging data in the main memory of a customer targeted VM from the host VM. The example of this approach is a logging system that was introduced by [15]. They describe that this system is in the host VM of a provider. The system can also collect and store logging data about customer VM's process activities such as what process accesses which file. The logging system consists of a logging process and log file [15]. The logging process is responsible for recording necessary logging data. In our experiment in Section 3, we call this process as the 'logger'. The log file is used for storing the logging data obtained from the logger. In this paper, we focus on this logging system from [15] to monitor and capture the logging data of activities of what process accesses a critical file in a customer VM's disk.

2.3. **How the system works.** Figure 1 illustrates the system architecture of the logging system from [10]. See the box inside domU in Figure 1, it is the read process or for short 'read'. In the experiment, we assume that this process can be controlled by an attacker and maliciously read the critical file or s.txt or the document shape inside diskU. The read process in this paper refers to a process of a malicious user which he or she uses to access the critical file of the customer in diskU. When the read application is executed in domU, the operating system or OS of domU creates the read process. Then, the OS reserves some area of main memory (memU) of a domU. We call this area as read_mem, see the oval shape in memU in hwU. The read_mem stores the data of the read process such as process identification (PID), process name, user ID definition (UID), and the file name that this process is reading such as s.txt in diskU in Figure 1. In dom0, the main

components of the logging system architecture are logger or the shaded box in dom0, LibVMI or the white box in dom0, and log files or the shaded box in disk0. LibVMI is a memory introspection tool to read read_mem from memU. Figure 1 illustrates the introspection approach that uses LibVMI to obtain logging data from memU. Usually, the LibVMI performs from a dom0 to obtain logging data of the s.txt or the document shape in diskU in Figure 1. We can deploy LibVMI in the dom0 to read the memory space or read_mem in memU in Figure 1. This memory space holds all the information that we need to record, which is a log file to obtain the contents of the history of critical file s.txt using introspection listed below. The steps are the circles with numbers in Figure 1. Step1, the logger in dom0 calls LibVMI to access memU to get the information in read_mem (step2) such as a file name of s.txt. Then, the obtained information is returned to the logger. The logger manages the obtained information and then writes (in step3) the information into the log file in disk0.

2.4. **The logger and accuracy.**

2.4.1. *The logger hardware environment configurations.* The read process can have activities with s.txt or the document shape in diskU in Figure 1. The details of activities and of how the logger (the white box in the user level of the dom0) works were illustrated in Figure 1. The main objective of the experiment in this paper is to measure the accuracy of the logger when the logger is recording logging data from some of the activities with s.txt file. To see the trends of the accuracy of the logger, the experiment or the measurement will be performed on different hardware configuration environments of the logger. In Figure 1 and on the different hardware configuration environments of the logger, the experiment will be performed by measuring the accuracy values of the logger after each hardware environment configuration by the method of varying the numbers of CPU cores and sizes of the main memory of both dom0 and domU. The measurement of the accuracy of the logger is also performed i) when the read process is reading s.txt in diskU, and ii) when the logger is capturing the desired logging data from the read_mem.

2.4.2. *Hit and miss.* Briefly, the accuracy of the logger or the shaded box in the dom0 is measured by a number of times that the logger captures the right file name or a string "s.txt" in read_mem. This is called 1 'hit'; otherwise it is 1 'miss'. For example, if the read process reads s.txt files 100 times, and the logger can capture the string "s.txt" in read_mem for 100 times or 100 hits, this means that the accuracy of the logger is 100% [16].

2.4.3. *The accuracy of the logger and sleeping time.* Section 2.4.2 described that the read process reads s.txt files 100 times and the logger can capture the string "s.txt" in read_mem 100 times or 100 hits, this means that the accuracy of the logger is 100%. And from our previous work [16], we measured our previous logger with a four CPU cores machine. Then, we found that the accuracy of this logger was 100% when the sleeping time is 65 ms. A sleeping time is an idle time after the read process completes opening and reading tasks of s.txt, and before it is terminated. Thus, when we state that 'the logger has 100% accuracy at 65 ms', this means that "in order to enable our previous logger to capture the right file name value as "s.txt" for every single time or we can say 100% at 65 ms, the read process needs to be in memU at least 65 ms after it finishes reading tasks but before it is terminated". Therefore, in this experiment, from the core(s) from 1 to 8, each core will be configured with 65 ms as the same as our previous work [16]. However, in the experiment in this paper, we will vary the CPU core numbers from 1 core to 8 cores. Then, we will measure the accuracy of the logger when the logger is running in a machine with 1 to 8 cores. Then, we will discuss all eight accuracy values of each logger with 1 to 8 CPU cores.

3. The Implementation.

3.1. **The hardware and software experimental environment.** To vary CPU core numbers for the experiment in this paper here, in dom0, we divide the environment into two parts. The first one is the hardware part, and we set up the experiment environment based on a Lenovo ThinkStation S20 computer machine. This machine comprises an Intel Xeon 3.06 GHz that is CPU 64-bit eight cores, SDRAM 8 GB of main memory, and 320 GB of secondary memory. The second one is the software part, we installed a Fedora 16 64-bit as the operating system or OS of the machine in the experiment, and also installed Xen 4.1.4 hypervisor on top of the OS. This hypervisor is used to simulate an IaaS cloud on this machine or Figure 1. Then we installed LibVMI 0.10.1 library on the OS that can be called by the logger to access to memU of domU from dom0. In domU, we also installed a Fedora 16 64-bit as the OS and set up the read application on the OS.

3.2. **The method of varying the numbers of CPU cores of both domU and dom0.** In this experiment, a domU is fixed to 1 GB main memory (memU).

3.2.1. *Alignment of varying of CPU cores.* This section studies the trend of the accuracy values of the logger in different numbers of CPU cores configuration environment. The environment is done by varying the numbers of CPU cores of dom0 and domU. In every configuration, the logger will still perform the same routines as discussed in Figure 1 in Section 2.3. Mainly from the figure, the logger needs to capture the necessary data from memU. The data are in read_mem and included: PID of the read process, the name of the read, UID or the ID of the read process's owner, and the file name that the read process is reading, as described in Section 2.3. In the alignment of varying of CPU cores, see the top dash-lined box in Figure 2(a), it is d01c which represents a dom0 that deploys 1 CPU core. Then, the second dash-lined box from the top of Figure 2(a) is d02c which represents a dom0 that deploys CPU 2 cores, and so on, until d08c or the first dash-lined box from the bottom of Figure 2(a). Thus, this box represents a dom0 that deploys 8 CPU cores. Similarly, see the shaded box on the top of Figure 2(a), it is du1c or domU that deploys 1 CPU core. Then, the second shaded box from the top of Figure 2(a) is du2c which represents a domU that deploys 2 CPU cores and so on, until du8c which represents a domU that deploys 8 CPU cores, see the third shaded box from the top of Figure 2(a).

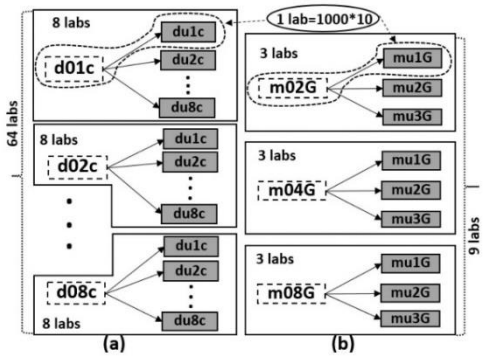


FIGURE 2. The method of varying the numbers of CPU cores and the sizes of main memory of both domU and dom0

3.2.2. *Overall CPU cores varying labs.* The full definition of a lab will be explained in Section 3.2.3. This section here explains the overall varying labs. The first eight labs, see in Figure 2(a), the freeform dotted shape that rounds both the top dash-lined box with labeled 'd01c' and the top shaded box with labeled 'du1c', this is one lab of our experiment or d01c-du1c lab. This lab is also represented by the first arrow from the top of Figure 2(a). Thus, the top white box of Figure 2(a) is d01c which is with 8 labs or d01c-du1c, d01c-du2c, ..., and d01c-du8c. The second 8 labs, see the second dash-lined box from the top of Figure 2(a), it is other 8 labs. The first lab is d02c-du1c or the fourth arrow from the top of Figure 2(a). Other seven labs are d02c-du2c, d02c-du3c, ..., and d02c-du8c. Then, the third to eighth 8 labs are aligned as the same pattern as the first two labs. Thus, there are 64 labs.

3.2.3. *A lab for varying cores.* We needed to modify the logger and the read process in Figure 1. Thus, the routines of the logger and the read here may be slightly changed from Figure 1. See in Figure 2(a) at the freeform dotted shape that rounds both the top dash-lined box with labeled 'd01c' and the top shaded box with labeled 'du1c', this is a lab. It composes of 3 steps or s1 to s3. S1, we run the logger in dom0. S2, we run the read process 1000 rounds in domU. Each round is independent but is ordered from 1^{st} to 1000^{th}. Thus, when the first read process is run and finished, then the second read process is run, and finished and so on, until the 1000^{th} read process is run and finished. In the same time, the logger that is run only one time before the first round of the read process running or s1. The logger will capture the "s.txt" string (this is 1 hit) from each round of the read process starting from the 1^{st} to 1000^{th} rounds. Then, the logger will store the string into the log file. This is also one round that the read process is done, and then the logger will wait for the next read process round running. When the logger gets the "s.txt" string of each round, each hit will be accumulated by 1 per hit. S3, the accumulated number of hits of these 1000 rounds is written to text files. Thus, a lab means we performed s1 to s3, and get the first accumulated number of hits of these first 1000 rounds. Then, we perform s1 to s3 for the other nine times; thus, we will get other nine accumulated numbers of hits. Finally, we can calculate the accuracy of the logger from these ten accumulated numbers of hits as a percentage. This is a lab and also the first lab or lab 1, which is d01c-du1c or the freeform dotted shape in Figure 2(a).

3.2.4. *First 8 for d01c and 64 labs for all d01c to d08c.* See the top box of Figure 2(a), it is 8 labs. The first lab or lab 1 is the freeform dotted shape of d01c-du1c as just discussed in Section 3.2.3 above. Then, the seven labs are outside the freeform dotted shape but still in the top box of Figure 2(a). They are lab 2 or d01c-du2c or the second arrow from the top of Figure 2(a), lab 3 or d01c-du3c,..., and lab 8 or d01c-du8c or the third arrow from the top of Figure 2(a). This is 8 labs for d01c or the top white box in Figure 2(a). What is 64 labs, the first 8 labs for d01c or the top white box in Figure 2(a) is just discussed above. Then, there will be other 8 labs for d02c (the first freeform shape from the top of Figure 2(a)) and other 8 labs for d03c and so on, until the last set of 8 labs or d08c or the first freeform box from the bottom of Figure 2(a). So, the total is 64 labs or a whole of Figure 2(a).

3.3. **Varying the sizes of main memory of both dom0 and domU.**

3.3.1. *Alignment of varying of main memory.* This section studies the trend of the accuracy of the logger when varying the sizes of the main memory of dom0 and domU. The logger needs to capture the necessary logging data from domU. The data are in read_mem and included: PID of the read process, the name of the read process, UID, and the file name that the read process is reading, as described in Section 2.3. For dom0, the top dash-lined box in Figure 2(b) is m02G which represents a dom0 that deploys 2 GB of main memory (mem0). Then, the second dash-lined box from the top of Figure 2(b) is

m04G which represents a dom0 that deploys 4 GB of mem0. Lastly, the last dash-lined box from the top of Figure 2(b) is m08G which represents a dom0 that deploys 8 GB of mem0. For domU, similarly, the shaded box on the top of Figure 2(b) is mu1G which represents a domU that deploys 1 GB of memU. Mu2G represents a domU that deploys 2 GB of memU. Finally, the third shaded box from the top of Figure 2(b) is mu3G which represents a domU that deploys 3 GB of memU. We will explain the first 3 labs or the first white box on the top of Figure 2(b), as follows. In Figure 2(b), see the freeform dotted shape that rounds both the top dash-lined box with labeled 'm02G' and the top shaded box with labeled 'mu1G', this is one lab of our experiment or m02G-mu1G lab or the first arrow on the top of Figure 2(b). Thus, see the top white box of Figure 2(b), m02G is with 3 labs or m02G-mu1G, m02G-mu2G, and m02G-mu3G. We will explain the second 3 labs or the second white box on the top of Figure 2(b), as follows. See the second dash-lined box from the top of Figure 2(b), it is the other three labs. The first lab is m04G-mu1G or the fourth arrow from the top of Figure 2(b). Thus, see the second white box from the top of Figure 2(b), m04G is with three labs or m04G-mu1G, m04G-mu2G, and m04G-mu3G. We will explain the third or last three labs or the first white box from the bottom of Figure 2(b), as follows. See the third dash-lined box from the top of Figure 2(b), it is other and last three labs. The first lab is m08G-mu1G or the seventh arrow from the top of Figure 2(b). Thus, see the first white box from the bottom of Figure 2(b), m08G is with three labs or m08G-mu1G, m08G-mu2G, and m08G-mu3G. Thus, there are 9 labs.

3.3.2. *A lab of main memory experiment and the first 3 for m02G and 9 labs for m02G to m08G.* See in Figure 2(b) at the freeform dotted shape that rounds both the top dash-lined box with labeled 'm02G' and the top shaded box with labeled 'mu1G', this is a lab of varying main memory. This lab composes of 3 steps or sm1 to sm3. These three steps are as the same s1 to s3, respectively. Thus, a lab here means we performed sm1 to sm3, and get the first accumulated number of hits of these first 1000 rounds. Then, we perform sm1 to sm3 for the other nine times. Thus, we will get other nine accumulated numbers of hits. Finally, we can calculate the accuracy of the logger from these ten accumulated numbers of hits as a percentage. This is a lab and also the first lab or lab 1 which is m02G-mu1G or the freeform dotted shape in Figure 2(b). For the first 3 and 9 labs for m02G, see the top white box of Figure 2(b), it is 3 labs. The first lab or lab 1 is in the freeform dotted shape of m02G-mu1G as just discussed above. Then, the two labs are outside the freeform shape but still in the top white box of Figure 2(b). They are lab 2 or m02G-mu2G and lab 3 or m02G-mu3G. These 3 labs are for m02G. What are 9 labs, the first 3 labs for m02G or the top white box of Figure 2(b) is just discussed above. Then, there will be 3 labs for m04G (the second white box from the top of Figure 2(b)); finally, other 3 labs for m08G or the first white box from the bottom of Figure 2(b). So, the total is 9 labs or a whole of Figure 2(b).

4. **Results and Discussions.**

4.1. **The results of varying the number of CPU cores of dom0.** The results here are from the experiment of all the three dash-lined boxes on the left of Figure 2(a). Figure 3(a) shows the results or the accuracy of the logger when varying the number of CPU cores of dom0 from 1 to 8 cores for the experiment. The results show that the accuracy of the logger is 100% when dom0 deploys 1 core or d01c, and 8 cores or d08c. When dom0 deploys 2 to 7 of CPU cores or the horizontal rectangle, this decreases the accuracy of the logger. This can be seen in Figure 3(a) that the accuracies of the logger of each one from d02c to d07c are 99%, 97%, 98%, 98%, 99%, and 99%, respectively. There is no any accuracy of the logger with d02c to d07c as 100%, compared to the accuracy of d01c and d08c.

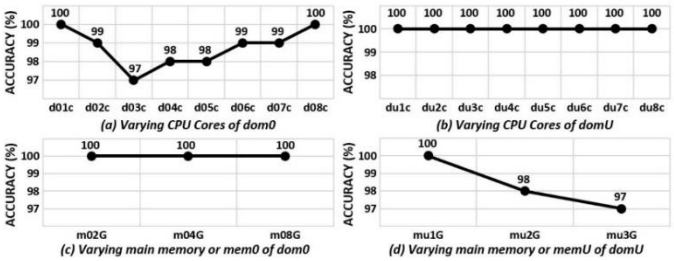


FIGURE 3. The results from varying the numbers of CPU cores and the sizes of main memory of both domU and dom0

4.2. **Discussion of varying the number of CPU cores of dom0.**

4.2.1. *General discussion of results.* For d01c, the accuracy of the logger is 100%. This is because the operating system of dom0 has only one CPU core to serve for all processes, including the logger process. Thus, the logger is never swapped from the current processing CPU core to use a new core. This may cause the accuracy of the logger to be 100%. More details of the swapping task are discussed in the next paragraph. For d02c to d07c, Section 4.1 showed that when varying the number of CPU cores of dom0 from 2 to 7 cores, this decreases the accuracy of the logger. The decreasing situation may be caused by CPU scheduling mechanisms which make the computer system have three properties: fairness, powerfulness, and rapidness, as argued by [17]. Because of these properties, whenever the CPU cores are idle, the operating system must select one of the processes from the ready queue to be executed, even the selected process is not the first one in the queue. If the logger is the selected process to be executed, and if dom0 has more than one core, then the logger process may be swapped from the current core to the new one. When swapping, from example, from the current core 1 to the new core 2, the operating system must move the processing data of the logger process from mem0 to caches of the new core 2 [18, 19]. A cache is a very high-speed memory and is a buffer between the mem0 and the CPU [20]. Thus, this swapping operation of the operating system mentioned above may consume the processing time of the logger process. This swapping task makes all the accuracy of the logger in d02c to d07c labs be lower than 100%. For d08c, the accuracy of the logger is 100%. This may be because the operating system of dom0 has enough CPU cores to serve for all processes, including the logger. Thus, the logger may have its own core to be exclusively processed. Thus, the operating system may not need to swap the logger from the current core to another core. This is no extra consuming processing time for the logger, and it can have efficient operations. Thus, this may cause the accuracy of the logger to be 100% at d08c.

4.2.2. *Trends of the accuracy from CPU cores varying.* There are two trends from the results of the varying CPU cores from the experiment. The first trend of the accuracy of the logger is for from d01c, to d02c, and finally to d03c. The trend is decreased from 100% to 99%, and finally, to 97%, respectively. This decreasing trend may be caused by the following reason. This reason is that d01c has no swapping tasks (discussed above) because it has only one core, compared to d02c, which may allow the logger to be swapped. This consumes the logger processing time. Then, this may cause the logger not be able to capture "s.txt" in time in one read process running round. Thus, to increase the cores of dom0 may decrease the accuracy of the logger in some appropriate core numbers such as 2 to 3 cores, as just discussed above. The second trend of the accuracy of the logger is

for from d03c to d08c and is increased from 97% to 100%, respectively. This increasing situation may be caused by one of the reasons. The reason is that when the logger running in dom0 that has many or enough cores such as 3 to 8, then, the operating system may not allow the logger to be swapped from one core to another. Less swapped tasks may decrease the logger's processing time. Thus, the logger has sufficient time to capture the "s.txt" in time of all the read process running rounds. Thus, to increase the cores of dom0 may increase the accuracy of the logger in some appropriate core numbers such as 3 and to 8 cores, as just discussed above. We also believe that to increase the cores for more than 8, the accuracy of the logger will still be 100%.

4.3. **Results and discussion of varying the number of CPU cores of domU.** These results and discussions are based on the experiment in Section 3.2, which already assumed that a domU is fixed to 1 GB memU. The conclusion of the results is that when we vary CPU cores of domU from 1 to 8, the accuracy of the logger has no effect by this varying task. Figure 3(b) shows all the accuracy values of the logger when increasing the number of CPU cores of domU from 1 to 8, as discussed in the experiment in Section 3.2. See, du1c to du8c in Figure 3(b), the results from the figure show that all the accuracy of the logger is all the same as 100%, when domU has the number of CPU cores for all number from 1 or du1c to 8 du8c. Thus, varying the numbers of CPU cores of domU causes no effect on the accuracy of the logger. The two appropriate reasons could be the following. Firstly, the logger performs in dom0, not domU, and the logger needs only to access memU in domU. Thus, varying by increasing domU core numbers should not affect the accuracy of the logger. This can be seen in Figure 3(b) that all the accuracy values between du1c to du8c are the same values: 100%. Secondly, the read process is small, then the process can work with both CPU 1 core or more than 1 cores of domU with no difference in processing time, as also agreed by [21]. From Figure 3(b), when the read process is running on CPU 1 core of domU, the accuracy now is 100%. And [21] states that a small process that works with 1 or many CPU cores yields no difference in the processing time of this process. Thus, from our experiment, when the logger works on 1, 2,..., or 8 CPU cores in domU, all the accuracies of the logger are all the same as 100% which is not different. This can be seen in Figure 3(b) that all the accuracy values between du1c to du8c are the same values: 100%. To sum up, when we vary CPU cores of domU from 1 to 8, the accuracy of the logger has no effect by this varying task.

4.4. **The results and discussions of varying the size of main memory of dom0.** Both core numbers of dom0 and domU are fixed to be 8. The conclusion is that when we vary mem0 of dom0 from 2, to 4, finally to 8, this does not affect the accuracy of the logger. Figure 3(c) illustrates all the accuracy of the logger when varying by increasing the sizes of mem0 of dom0 from 2, 4, to 8. The results from Figure 3(c) are that all the accuracy values of the logger are all the same as 100%, when dom0 has mem0 as 2, 4, and 8 GB. One of the reasons is that the logger process is small, and the logger also needs a small area of mem0 for processing. When increasing the mem0 to be larger, the logger still uses the same small area. Thus, this does not affect the logger accuracy. To sum up, when we vary mem0 of dom0 from 2, 4, to 8, this does not affect the accuracy of the logger.

4.5. **The results and discussions of varying the size of main memory of domU.** Both core numbers of dom0 and domU are fixed to be 8. The conclusion is that varying the memU of domU decreases the accuracy of the logger. See Figure 3(d), the results from the figure show that the accuracy of the logger is decreased from 100% to 98%, and finally to 97%, when the memU is increased from 1 GB to 2 GB, and finally to 3 GB, respectively. This may be because increasing the size of memU of domU will enlarge the area which is searched by the logger for the logging data. Thus, the logger will consume

more time to search the desired logging data in this enlarged area, compared to smaller memU area. We also believe that enlarging the size of memU to be more than 3 GB, this may still also decrease the accuracy of the logger.

5. **Conclusions.** This paper illustrated the accuracy measurement of a logging system or logger in different hardware configurations environments in the Infrastructure as a Service (IaaS) cloud. For the CPU cores measurement, i) when varying the numbers of CPU cores of the host machine (dom0) of the logger, there are two perspectives of the results. Firstly, there are three general results: at 1 core, from 2 to 7 cores, and at 8 cores. At 1 core, the accuracy of the logger is 100%. For 2 to 7 cores, it decreases the accuracy of the logger from 100%. Then, at 8 cores, the accuracy of the logger is still 100%. Secondly, there are two trends in the accuracy of the logger. The first trend is for from 1, to 2, and finally to 3 cores, and the accuracy is decreased from 100%. The second trend is for from 3 to 8 cores, and the accuracy is increased from 97% to 100%, respectively. Thus, to increase the cores for more than 8, the accuracy of the logger should still be 100%. ii) When varying the numbers of CPU cores of a customer virtual machine (VM) or domU, the results are that when we vary CPU cores of domU from 1 to 8, all the accuracy values are 100%. Thus, the accuracy of the logger has no effect by this varying task. For the main memory measurement, a) for dom0, the results are that when we vary main memory or mem0 of dom0 from 2, to 4, finally to 8, this does not affect the accuracy of the logger; b) for domU, the results are that varying the main memory memU of domU decreases the accuracy of the logger when the memU is increased from 1 GB to 2 GB, and finally to 3 GB. We also believe that enlarging the size of memU to be more than 3 GB, this may still also decrease the accuracy of the logger, compared to less than or equal to 3 GB. The future work is to apply parallel programming to enhancing the accuracy of the logger.

REFERENCES

[1] A. Bhawiyuga, D. P. Kartikasari, K. Amron, O. B. Pratama and M. W. Habibi, Architectural design of IOT-cloud computing integration platform, *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 2019.

[2] S. Alshamrani, An efficient algorithm for monitoring virtual machines in clouds, *Bulletin of Electrical Engineering and Informatics*, 2019.

[3] R. Kaur and G. Kaur, Proactive scheduling in cloud computing, *Bulletin of Electrical Engineering and Informatics*, 2017.

[4] J. Brodkin, Gartner: Seven cloud-computing security risks, *Infoworld*, 2008.

[5] W. Liu, Research on cloud computing security problem and strategy, *International Conference on Consumer Electronics, Communications and Networks*, 2012.

[6] S. Subashini and V. Kavitha, A survey on security issues in service delivery models of cloud computing, *Journal of Network and Computer Applications*, 2011.

[7] T. T. W. Group et al., The treacherous 12: Cloud computing top threats in 2016, *Cloud Security Alliance*, 2016.

[8] I. Molyneaux, *The Art of Application Performance Testing: From Strategy to Tools*, O'Reilly Media, Inc., 2014.

[9] P. Chan-In and W. Wongthai, Performance improvement considerations of cloud logging systems, *ICIC Express Letters*, vol.11, no.1, pp.37-43, 2017.

[10] W. Wongthai, *Systematic Support for Accountability in the Cloud*, Ph.D. Thesis, Newcastle University, 2014.

[11] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield, Xen and the art of virtualization, *SIGOPS Oper. Syst. Rev.*, 2003.

[12] R. Love, *Linux Kernel Development*, 3rd Edition, Addison-Wesley Professional, 2010.

[13] R. K. Ko, P. Jagadpramana and B. S. Lee, Flogger: A file-centric logger for monitoring file access and transfers within cloud computing environments, *International Conference on Trust, Security and Privacy in Computing and Communications*, 2011.

[14] B. Payne, *About the VMI Tools Project, Google Project Hosting*, 2013.

[15] W. Wongthai, F. L. Rocha and A. van Moorsel, A generic logging template for infrastructure as a service cloud, *International Conference on Advanced Information Networking and Applications Workshops*, 2013.
[16] W. Wongthai and A. van Moorsel, Performance measurement of logging systems in infrastructure as a service cloud, *ICIC Express Letters*, vol.10, no.2, pp.347-354, 2016.
[17] N. Ishkov, *A Complete Guide to Linux Process Scheduling*, Master Thesis, Tampere University, 2015.
[18] P. Gepner and M. F. Kowalik, Multi-core processors: New way to achieve high system performance, *International Symposium on Parallel Computing in Electrical Engineering*, 2006.
[19] E. Cota-Robles, *Priority Based Simultaneous Multi-Threading*, United States Patent, 2003.
[20] J. Handy, *The Cache Memory Book*, Morgan Kaufmann, 1998.
[21] M. Rouse, *Definition: Multi-Core Processor*, TechTarget, vol.6, 2013.

ภาคผนวก ค ผลงานตีพิมพ์เผยแพร่ งานที่ 2.

Performance Considerations of a Logging System Simultaneously with a Customer Virtual Machine in Infrastructure as a Service Cloud

Thongrob Auxsorn, Winai Wongthai, Thanathorn Phoka and Wichep Jaiboon

Abstract An Infrastructure as a Service or IaaS cloud can offer a virtual machine or VM to the cloud customers to rent with pay per use basis. Although this cloud is widely used in many areas of applications, its security issues are obstacles for its adoptions. A logging system can mitigate the risks associated with these issues. This system can monitor and log malicious processes in IaaS customer's VMs. However, the performance of this system is important to be measured. Moreover, the performance of this VM is also needed to be measured. Thus, this paper focuses on simultaneously measuring and evaluating the performance of both the logging system and the VM. To the best of our knowledge, there is no this kind of measuring methods in the literature. The main result of this paper is that we found the optimal point when the logging system and the VM can have suitable performance levels when both of them are working. This optimal point can be used by the IaaS providers to offer marketing packages for mitigating the risks associated with the issues of a customer VM with the consideration of his/her VM performance simultaneously. To mitigate risks associated with the security issues of an IaaS, this paper can be a guild-line to enable logging systems to be truly worked in the IaaS real-word production.

Keywords Cloud security · Logging system · Performance · Virtual machine

T. Auxsorn · W. Wongthai (✉) · T. Phoka
Department of Computer Science and Information Technology, Naresuan University, Thailand
e-mail: winaiw@nu.ac.th

T. Auxsorn
e-mail: thongroba59@email.nu.ac.th

T. Phoka
e-mail: thanathornp@nu.ac.th

W. Wongthai
Faculty of Science, Research Center for Academic Excellence in Nonlinear Analysis and Optimization, Naresuan University, Phitsanulok 65000, Thailand

W. Jaiboon
Department of Computer Science, Faculty of Business and Public Administration Nation University, Lampang 52000, Thailand
e-mail: wichep_jai@nation.ac.th

K. J. Kim and H.-Y. Kim (eds.), *Information Science and Applications*,
Lecture Notes in Electrical Engineering 621,
https://doi.org/10.1007/978-981-15-1465-4_30

1 Introduction

An Infrastructure as a Service or IaaS cloud can offer a virtual machine or VM to the cloud customers to rent with pay as you go basis [1]. This cloud is widely adopted in many areas of applications [2]. However, this cloud causes security issues, which are obstacles for its marketing growth. Cloud Security Alliance or CSA officially identifies the issues as threats of this cloud in one [3] of its reports. A logging system can help to mitigate the risks associated with the CSA threats [4]. This system can monitor and log malicious processes in customer's VMs as done by [5], and agreed by [2]. The performance of the logging system is significant and should not be neglect [4, 6]. More details of the performance are in Sect. 2.3. Moreover, while a customer VM is being monitored by the logging system, [7] state that the performance of this VM is also important, and needed to be measured. This is because while a logging system is operating, it will freeze the monitored VM to capture the data in the main memory of VM. Thus, this paper focuses on measuring the performance of both the logging system and the VM simultaneously. To the best of our knowledge, there is no this measuring method in the literature.

Summary of contributions: firstly, we introduce a framework in mathematics equation forms for measuring and evaluating the VM performance. These equations are useful and can be used for other logging systems when measuring and evaluating VMs. Secondly, from the first contribution, we measure and evaluate the VM performance simultaneously when a VM is being monitored by a logging system. Previous work such as [4, 8] focuses on solely measuring and evaluating the logging system performance not simultaneously with the VM performance. VM performance is important for customer perspectives. This is because the customer can be informed that what conditions of his/her VM are, while this VM's risks associated with the CSA threats are mitigated by the logging systems [7, 9]. Thirdly, also from the first contribution, we deeply investigate and illustrate the mechanisms of detecting and capturing methods of the logging systems for better measuring and evaluating the performance of both the logging systems and also the VM simultaneously. This is useful because we can deeply analyze how and why the logging system performance is decreased or increased. This enables us to predict the performance of this system and to enhance the system to be more efficient and effective in the future. This makes the logging system to be ready to perform in the IaaS real-word production. Lastly, we investigated and found the optimal point when the logging system and the VM can have acceptable performance levels when both of them are operating their tasks. This optimal point can be used by the IaaS providers to form marketing packages of mitigating the risks associated with the CSA threats for a customer VM with the consideration of his/her VM performance simultaneously. To the best of our knowledge, there are no these contributions above in the literature.

2 Background

2.1 *Infrastructure as a Service Cloud Architecture and Logging System Architecture*

Both an IaaS cloud architecture and logging system architecture are in Fig. 1. For the experimental purposes of this paper, both architectures are adapted from our previous work [8]. In Fig. 1, the main components of the IaaS architecture are the hypervisor, dom0, domU, hw0, hwU, disk0, diskU, and memU. The hypervisor software can run more than one virtual machine or VM in one physical machine, see the box with number 2 in Fig. 1. The domain 0 or dom0 is a manager of all the created VMs, see at the topmost left box in the figure. At the system booting period, the dom0 is launched by the hypervisor, and it is also a VM. The dom0 exclusively accesses hw0 and manages all the created domUs. See the topmost right box of Fig. 1, a user domain or domU is a user VM that is run on top of the hypervisor. A domU is an IaaS cloud product that the provider offers to an IaaS customer. HwU is physically located in hw0 and is domU's virtual hardware. It is owned by the dom0 or by the provider and virtually owned by a domU owner or by an IaaS customer.

Disk0 is a physical disk of the dom0, whereas diskU is a virtual disk of a domU. Finally, memU is domU's virtual main memory. The system architecture of the logging system from [8] is also in Fig. 1. The read process or for short 'read' is in domU, see the box inside domU in Fig. 1. This paper assumes that this process can be controlled by an attacker. Then he/she can maliciously read the critical file or s.txt or the document shape inside diskU. From Fig. 1, the steps of the logger are the circles with numbers 1–3. Step1, the logger in dom0 calls LibVMI to access memU to get the logging data from read_mem (step2) such as a file name of s.txt or the string "s.txt" or the process ID or PID of the read process. Then, LibVMI accesses memU to obtain such data in read_mem, and returns the obtained data back to the logger. Then, the logger manages the data then writes (in step3) the data into log files.

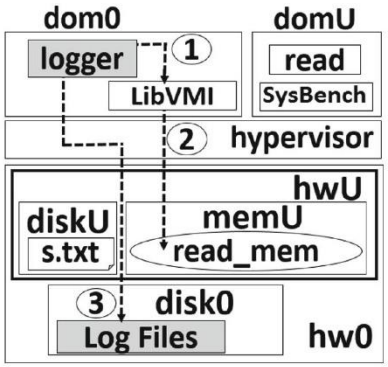


Fig. 1 IaaS cloud architecture and logging system architecture [10]

2.2 The Types of Logging Systems

Ko et al. [5, 11] divided logging systems into two main types, including a file-centric and system-centric logging system. The first type focuses on tracing a customer file's life cycle in a domU [11, 12]. Whereas, the second one focuses on producing the cloud provider's hardware layer log files, such as memory usages, voltage, temperature [13]. However, research in the literature mainly focuses on a system-centric logging system, rather than file-centric one [12]. For example, to produce mainly system-centric logs, [13, 14] proposed recommended monitoring principles that should be operated by a cloud provider. However, the cloud provider usually discloses the system-centric log files to the cloud customers [15]. Thus, this paper here focuses on a file-centric logging system rather than a system-centric one.

2.3 Performance and Accuracy of Logging Systems

The National Security Agency (NSA) [16] states that cloud architectures are more difficult to understand and non-concrete than a traditional client-server model. This can make the performance testing of logging systems in IaaS cloud to be challenged. However, to increase reliability and trust in the security of a cloud user's data and processing in an IaaS, performance measurement of logging systems is an essential factor [8, 17–19]. In brief from our previous work [8, 18], one of the four application system (including a logging system) key performance indicators (KPIs) is response time. It is the amount of time it takes for the application to respond to a user request [6]. This paper here focuses only on the response time as the example KPI of performance considerations to encourage participators to concern all the KPIs. This paper focuses on the accuracy of logging systems that are located in a dom0. The accuracy means the accuracy of the logging system in logging the logged information from volatile memory in a target monitored domU. Chan-In and Wongthai [8], Wongthai and Van Moorsel [18] consider the accuracy as one of the factors of the response time KPI. Thus, to measure the performance of a logging system, Sect. 3.1 thoroughly discusses the measurement method of the accuracy of the system.

2.4 SysBench Tool

Performance of a domU is obtained from SysBench tool. For a system running a program with intense load, this tool can be used to evaluate OS essential parameters [20, 21]. This tool is also used to measure a VM (or a domU in our paper). The tool is installed in a domU, see the box under the read box in Fig. 1. In this paper, we used SysBench to measure a domU processing, for example, after we boot a domU which have some processes in its system, then we run SysBench and the domU processing

time of Sysbench value is 10.0546 μs. Then, we used the logger to access memU of the domU for monitoring and logging, the SysBench value of this domU is increased to be 23.5905 μs, which is 42%. This means that when the logger is monitoring and logging this domU, this domU performance is dropped around 58%. Thus, the performance of this domU before being monitoring and logging is better than after it is being to do so.

3 The Implementation and Experiment

3.1 The Hardware and Software and the Logger Process Steps in the Experiment

The first one is the hardware part; we use an Intel Pentium Core 2 Quad Processor Q9400, 2.66 GHz. This processor is 64-bit CPU 4 cores, SDRAM 8 GB of main memory, and 320 GB of secondary memory. The second one is the software part; we installed a Fedora 16 64-bit as the operating system or OS of the machine in the experiment and also installed Xen 4.1.4 hypervisor on top of the OS. This hypervisor is used to simulate an IaaS cloud on this machine or Fig. 1. Then we installed LibVMI 0.10.1 library on the OS that can be called by the logger to access to memU of domU from dom0. In domU, we also installed a Fedora 16 64-bit as the OS and set up the read application on the OS. The steps of the logger are the circles with numbers 1–3 in Fig. 1, as discussed in Sect. 2.1. The experiment in this paper based on the architecture in this figure. The logger processes in the experiment are (1) the initial step of the logger (2) the logger pauses the domU (3) the logger searches the "s.txt" string in read_mem in mem_U (4) the logger resumes the domU, and (5) the logger is repeated from step (2). We call the processes of the logger from step 2 to 4 as a round of the logger. For example, when the logger repeats step 2–4 from 2 times, this is two rounds of the logger. In this experiment, we will add a sleeping time such as 1000 microseconds (μs) between the logger process steps 4 and 5. We call this sleeping time as a logger sleeping time. This sleep time allows the monitored domU to be resumed to operate its tasks.

3.2 The Performance Measurement of a domU

From our previous work [4, 8], we focused on measuring the performance of the logger, not the performance of a domU. This section explains the measurement of the performance of a domU while is being monitored by the logger. We used the logger from our previous logging system architecture or Fig. 1 to perform the measurement of a domU. The performance of a domU is in Eq. (2). From Sect. 3.1, we call the processes of the logger from step 2 to 4 as a round of the logger. Thus, from Eq. (1),

n is a number of rounds of the logger, and it can be calculated from *sbt* divided by *lst*. The *sbt* is a Sysbench value of a domU, as discussed in Sect. 2.4. *lst* is a logger sleeping time, as discussed in Sect. 3.1. In this experiment, this time is varied from 250, 500, 1000, 1500, 2000, 2500, to 3000 μs. These are seven labs or 250–3000. Before we perform the next lab of the experiment after the previous lab, we cleared the space in mem0 that is occupied by the logger of the previous lab to be empty.

$$n = \frac{sbt}{lst} \tag{1}$$

$$dup = \left(\frac{\sum_{i=1}^{n} lst_i}{\sum_{i=1}^{n} lpt_i + \sum_{i=1}^{n} lst_i + \sum_{i=1}^{n} \epsilon_i} \right) * 100 \tag{2}$$

From Eq. (2), *dup* is domU performance in percentage unit, 0–100%. The *lpt* is a logger processing time or step 1–5, discussed in Sect. 3.1 above. All the values of *lpt* were obtained from SysBench tool [20]. We already measured and obtained the average of all the 10,000 values of *lpt* for 10 times. This gives us 10 average values between 233 and 274 μs. However, we used the average value of 250 μs to be suitable for the experiment in this paper. Then, ϵ is an overhead value of the logger operations in μs. This value is generated from the logger process in steps 1–4. The examples of this value can be 30–20,000 μs. The value will be increased when the number of rounds of the repeating step (step 5 above) were increased. Note that, from the experiment, we found that the minimum value of ϵ is 330 μs. Thus, from (1), when n is higher then $\sum_{i=1}^{n} \epsilon_i$ is also higher. For example, when n is n is 2 then $\sum_{i=1}^{n} \epsilon_i = \epsilon_1 + \epsilon_2$, and n is 10 then $\sum_{i=1}^{n} \epsilon_i = \epsilon_1 + \epsilon_2 + \epsilon_3 + \cdots + \epsilon_{10}$.

3.3 The Measurement of the Logger Accuracy

Hit and miss for the logger: the steps of the logger are the circles with numbers 1–3 in Fig. 1, as discussed in Sect. 2.1. The measurement of the logger accuracy (for short la) is performed (i) when the read process is reading s.txt in diskU, and (ii) when the logger is capturing the desired logging data from the read_mem. The accuracy of the logger is measured by a number of times that the logger captures the right file name or a string "s.txt" in read_mem. This is called 1 'hit'. Otherwise, it is 1 'miss'. For example, if the read process reads s.txt files 100 times, and the logger can capture the string "s.txt" in read_mem for 100 times or 100 hits, this means that the accuracy of the longer is 100% [8].

Steps of the experiment to calculate the logger accuracy: we modified the logger discussed (in Sects. 2.1 and 3.1) to be suitable for the experiment in this paper. The experiment is S1–S3 below. S1, we run the modified logger in dom0 one time, as described above. S2, we run the read process 10,000th rounds in domU. Each round is independent but are ordered from 1st to 10,000th. Thus, when the first read process is run and finished, then the second read process is run and finished and so on, until

the 10,000th read process is run and finished. In the same time, the logger that was run before the first round of the read process running or S1. The logger will capture the "s.txt" string from each round of the read process starting from the 1st to 10,000th rounds. Then, the logger will store the string into the log file in disk0 and print result from the logger capture on screen. When the logger gets the "s.txt" string of each round, each hit will be accumulated by 1 per hit. Lastly, S3, the accumulated number of hits of these 10,000th rounds is written to text files. Thus, an experiment means we performed S1–S3, and get the first accumulated number of hits of these first 10,000th rounds. Then, we perform S1–S3 for the other nine times; thus, we will get other nine accumulated numbers of hits. Finally, we can calculate the accuracy of the logger from these ten accumulated numbers of hits as a percentage.

4 Result and Discussion

4.1 A domU Performance or dup

Figure 2 is the result of the experiment form Sect. 3.2. The result is a domU performance or *dup* in %, see y-axis of in the figure. Or, how much the performance of a domU is decreased while the logger is monitoring this domU. X-axis is a logger

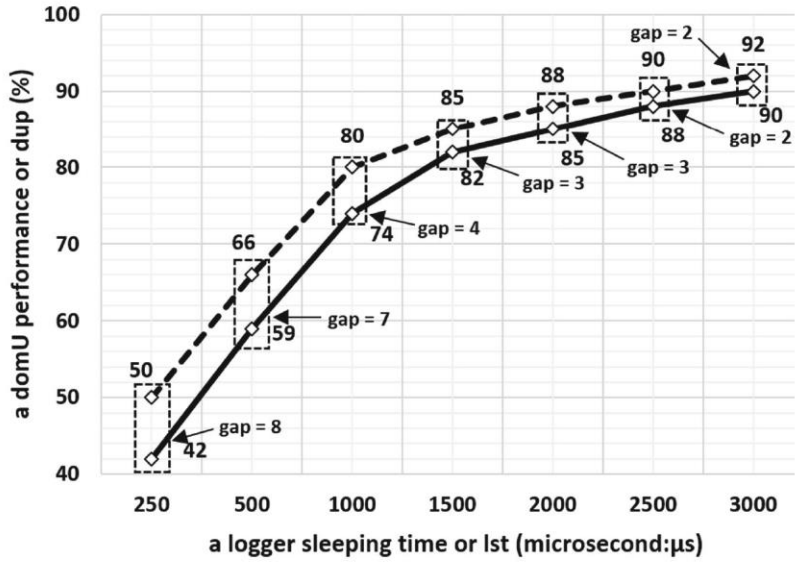


Fig. 2 Performance of a domU

sleeping time or *lst* in μs as: 250, 500, 1000, 1500, 2000, 2500, and 3000. The conclusion of this result in this section is that increasing the *lst* values (from 250 to 3000) also increases *dup* values. However, increasing the *lst* values decreases the logger accuracy or *la*. Thus and later, we found that at the *lst* value of 1000 μs the logger has the suitable or acceptable accuracy value while the monitored domU has the suitable or acceptable performance value as 74%. We can say that the combination of both numbers is an optimal pair of values (*lst*1000 μs ≫ *dup*74%) found in our experiment. This pair of values enables the *la* to be 99.67%. Again, we can call these three values as an optimal value of both the *la* value and the *dup* value or *lst*1000 μs ≫ *dup*74% ≫ la99.67%.

The ideal *dup* values: from Eq. (2), *dup* values will be calculated seven times as dup_1 to dup_7 based on seven *lst* values and with the flowing conditions: (i) *lst* values will be seven values as from 250 to 3000 μs, (ii) when $\sum_{i=1}^{n} \epsilon_i = 0$, (iii) each value of lst_1 to lst_n in $\sum_{i=1}^{n} lpt_i$ is the same as a constant value of 250 μs. All the seven *dup* values are in the dash line. For the first calculation, when *lst* value is 250 μs, each value of lst_1 to lst_n in $\sum_{i=1}^{n} lst_i$ is the same as a constant value of 250 μs. Then, dup_1 is 50%, see the dashed line when *lst* is 250 on X-axis. For the second calculation, when *lst* value is 500 μs, each value of lst_1 to lst_n in $\sum_{i=1}^{n} lst_i$ is the same as a constant value of 500 μs. Then, dup_2 is 66%, see the dashed line when *lst* is 500 on X-axis. Then, we calculated dup_3 to dup_7, and the values are 80–92%, respectively. All these seven *dup* values are in the dashed line. The conclusion of this ideal trend or the dashed line in this section is that increasing the *lst* values (from 250 to 3000) also increases *dup* values (from 50 to 92%). And all the *lst* values in $\frac{\sum_{i=1}^{n} lst_i}{\sum_{i=1}^{n} lpt_i + \sum_{i=1}^{n} lst_i}$ in Eq. (2) are the main variables that affect all the *dup* values.

The actual *dup* values: from Eq. (2), *dup* values will be obtained from the experiment for seven times as dup_1 to dup_7 based on seven *lst* values and with the flowing conditions: (i) *lst* values will be seven values as from 250 to 3000 μs, (ii) when $\sum_{i=1}^{n} \epsilon_i \neq 0$, (iii) each value of lst_1 to lst_n in $\sum_{i=1}^{n} lpt_i$ is the same as a constant value of 250 μs. All the seven *dup* values are in the thick line. Each ϵ of ϵ_1 to ϵ_n in $\sum_{i=1}^{n} \epsilon_i$ is an overhead value in μs, as discussed in Sect. 3.2. This value is generated from the logger process. For the first experiment, when *lst* value is 250 μs, each value of lst_1 to lst_n in $\sum_{i=1}^{n} lst_i$ is the same as a constant value of 250 μs. Then, dup_1 is 42%, see the thick line when *lst* is 250 on X-axis. For the second experiment, when *lst* value is 500 μs, each value of lst_1 to lst_n in $\sum_{i=1}^{n} lst_i$ is the same as a constant value of 500 μs. Then, dup_2 is 59%, see the thick line when *lst* is 500 on X-axis. Then, we obtained dup_3 to dup_7 from the experiment, and the values are 74%–90%, respectively. All these seven *dup* values are in the thick line. The conclusion of this actual trend or the thick line in this section is that increasing the *lst* values (from 250 to 3000) also increases *dup* values (from 42 to 90%). And all the *lst* values in $\frac{\sum_{i=1}^{n} lst_i}{\sum_{i=1}^{n} lpt_i + \sum_{i=1}^{n} lst_i + \sum_{i=1}^{n} \epsilon_i}$ in Eq. (2) are the main variables that affect all the *dup* values.

The comparison of the ideal *dup*, and actual *dup* values: see the gap values between the thick and dash lines, these values are 8, 7, 4, 3, 3, 2, and 2. The thick line or the actual *dup* values are always lower than the dashed line or the ideal *dup* values.

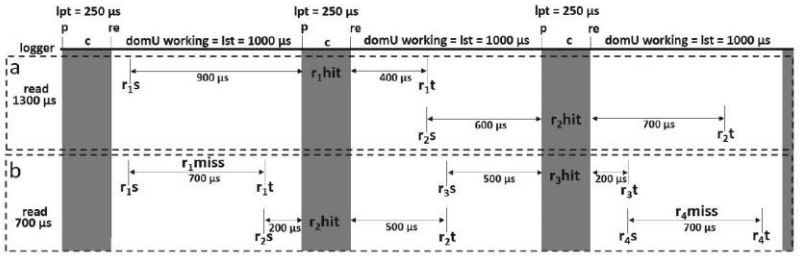


Fig. 3 Mechanisms of detecting and capturing methods of the logger

It is because all the ϵ values (ϵ_1 to ϵ_n) are generated from the logger process in the actual experiment. Thus, this makes $\sum_{i=1}^{n} lpt_i + \sum_{i=1}^{n} lst_i + \sum_{i=1}^{n} \epsilon_i$ expression to be higher. While in the idle *dup* values that are from the calculation, $\sum_{i=1}^{n} \epsilon_i$ is 0. When *lst* value is higher (from 250 to 3000 µs), then the thick line or the actual *dup* values will be closed to the dashed line or the ideal line. Or, the gab values will be closed to 0. This means that when the value of *lst* is higher, then the *dup* value is also higher. However, the higher of the *lst* values will decrease the logger accuracy. Thus, we examined and found in our experiment that an optimal pair of values is when *lst* is 1000 µs, so this makes the *dup* value will be 74%, as also discussed above.

4.2 The Logger Accuracy

The mechanisms of detecting and capturing methods of the logger to form the logger accuracy: is shown in Fig. 3. $r_m s$ is an abbreviation of $read_m$ process is started, and $r_m t$ is an abbreviation of readm process is terminated. We call the space between the $r_m s$ and $r_m t$ as a line, for example, $r_1 s$ to $r_1 t$ is one line or $r_1 s - r_1 t$. A 'hit' means that when $r_m s - r_m t$ line crosses any one shaded box in Fig. 3, for example, see in box a in Fig. 3, when $r_1 s - r_1 t$ line crosses the second shaded box, this is one hit. This is because with the $r_1 s - r_1 t$ line, this read1 process was started and terminated while the logger was not sleeping (crossing the second shaded box). Thus, the logger was able to capture this read process, a hit or r_1 hit in box a. A 'miss' is when $r_1 s - r_1 t$ line is in between the two connected shaded boxes, for example, see in box b in Fig. 3, when $r_1 s - r_1 t$ line is in between the first two connected shaded boxes, this is one miss. This is because, with the $r_1 s - r_1 t$ line, this $read_1$ process was started and terminated while the logger was sleeping. Thus, the logger did not capture this read process, a miss or $r_1$miss in box b. From the experiment, the read process working time is about 1300 µs by average, see the text 'read 1300 µs' at the beginning of box a in Fig. 3. See the tops of all the shaded boxes in Fig. 3, *lpt* is a period of time (such as 250 µs) when the logging process is working. This means that the logger can pause (p) the monitored domU, then find (f) the read process in memU, and resume (re) this domU.

See the space between the first and the second shaded boxes; this space is a period of time or *lst* (such as 1000 μs).This is the time that the logger is sleeping or no finding tasks for the read process in memU, while the monitored domU is working for *lst* or 1000 μs. When the read process working time is about 1300 μs, see the text 'read 1300 μs' in the beginning of box a in the figure, the box shows that when the read process working time is about 1300 or 'read 1300 μs', this makes no miss. This is because domU working time, *lst* value, or 1000 is less than or equal 1300 μs' ('read 1300 μs'). This makes each of all $r_m s - r_m t$ lines (for m = 1–2) crosses one shaded box, so no any miss. When the read process working time is about 700 μs, see the text 'read 700 μs' in the beginning of box b, the box shows that when the read process working time is about 700 or 'read 700 μs', this makes two misses. This is because of that domU working time, *lst* value, or 1000 is greater than 700 μs' ('read 700 μs'). This makes two lines (see $r_1 s - r_1 t$ and $r_4 s - r_4 t$ in box b) of all $r_m s - r_m t$ lines (for m = 1–4) were in between two connected shaded boxes, so two misses.

The *lst* increased, the logger accuracy decreased: from Fig. 4, there are three points of the conclusion of this result. Firstly, when lst values are increased or X-axis, logger accuracy or *la* values will be decreased. Secondly, we found that at *lst* value of about 1000 μs is the optimal point between *lst* and the *la* values. Lastly, when *lst* values are at 250, 500, and 1000 μs, the *la* value of each *lst* can be 100%. Firstly, when *lst* values are increased, the *la* values will be decreased. This can be seen when at *lst* as 250, *la* is 99.86, at *lst* as 500, *la* is 99.82, …, and lastly, at *lst* as 3000, *la* is 39.06. This is because when the *lst* values are higher, then the logger is slept with a higher gap of time. Thus, the logger may miss capturing when the read

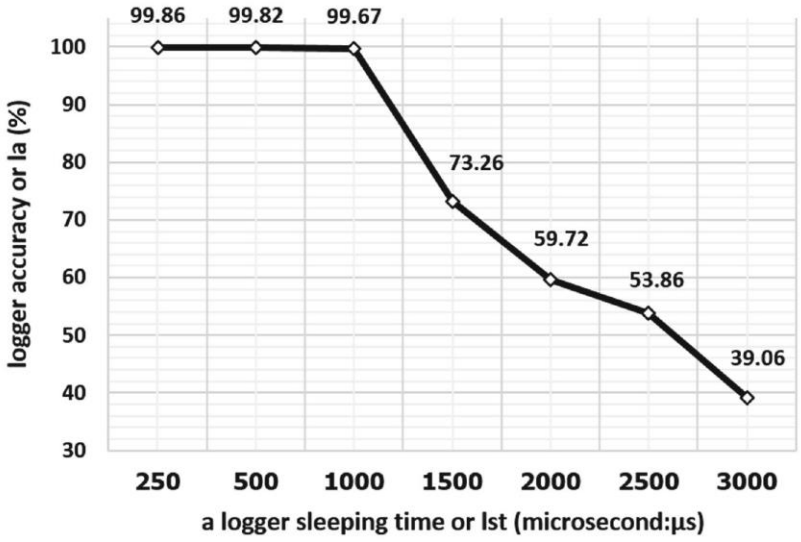


Fig. 4 The logger accuracy values at the read working time about 1300 μs by average

process stays in memU so many times. This makes many misses, and then *la* values are low.

The optimal point: see Fig. 2, we already discussed in Sect. 4.2 that the optimal pair of values found in our experiment is when $lst = 1000\ \mu s$, and $dup = 74\%$. This is the same here in Fig. 4, the optimal pair of values of both the logger accuracy value and *lst* is when $lst = 1000\ \mu s$, and $la = 99.67\%$. Again, we can call these three values as an optimal point of both the logger accuracy value and the *dup* value or $lst 1000\ \mu s \gg dup 74\% \gg la 99.67\%$. This optimal point can be used for a cloud customer to choose her own optimal point to suit her security, *la*, and *dup* requirements. From example, if she chooses $lst 1000\ \mu s \gg dup 74\% \gg la 99.67\%$, this means that her logger is good as *la* as 99.67, and this reduces her domU performance for about only 26%. However, if she chooses the optimal point of $lst 3000\ \mu s \gg dup 90\% \gg la 39.06\%$, this means that her logger will slow down her domU only 10% which is a very good condition for her domU. However, the *la* is only 39.06%, which means that her logger can capture all the hacker attacks less than half of all the attacks. This may not be acceptable for her?

5 Conclusion

Although an Infrastructure as a Service or IaaS cloud is widely used, its security is still the problem. A logging system can help to mitigate the security issue. However, researchers introduce logging systems without investigating and evaluating both the logging system's and an IaaS customer virtual machine's performance simultaneously. Thus, this paper firstly introduced a framework in mathematics equation forms for measuring and evaluating an IaaS customer virtual machine or VM performance. Secondly, based on the equations, we measured and evaluated the performance of a VM simultaneously when this VM is being monitored by a logging system. Thirdly, we deeply investigated and illustrated the mechanisms of detecting and capturing methods of the logging systems for better measuring and evaluating the performance of both the logging systems and also of the VM simultaneously. Lastly and importantly, we discuss the optimal point when a logging system and a VM can have the appropriate performance levels, when both of them are functioning. This optimal point can be used by the IaaS providers to generate marketing security options. The options can help in mitigating the risks associated with the security issue of a customer VM with the consideration of his/her VM performance simultaneously. To mitigate risks associated with the security issue of an IaaS, this paper can be a guild-line to enable logging systems to be truly performed in the IaaS real-word production. The future work is applying the guild-line to enhance the logging systems to be ready to perform in the IaaS real-word production such as in the OpenStack system [22].

References

1. Runathong W, Wongthai W, Panithansuwan S (2017) A system for classroom environment monitoring using the internet of things and cloud computing. Lecture notes in electrical engineering, vol 424. Springer, Singapore, pp 732–742
2. Haque S, Atkison T (2018) A forensic enabled data provenance model for public cloud. J Dig Forensics Secur Law 13(3)
3. CSA (2016) The treacherous 12: cloud computing top threats in 2016. Top threats working group, Cloud Security Alliance
4. Wongthai W (2014) Systematic support for accountability in the cloud: Ph.D. dissertation. Newcastle University
5. Ko RKL, Jagadpramana P, Lee BS (2011) Flogger: a file-centric logger for monitoring file access and transfers within cloud computing environments. In: 2011 IEEE 10th international conference on trust, security and privacy in computing and communications. IEEE, pp 765–771
6. Molyneaux I (2014) The art of application performance testing: from strategy to tools. O'Reilly Media, Inc
7. Payne BD, de Martim DPA, Lee W (2007) Secure and flexible monitoring of virtual machines. In: Twenty-third annual computer security applications conference (ACSAC 2007). IEEE, , pp 385–397
8. Chan-In P, Wongthai W (2017) Performance improvement considerations of cloud logging systems. ICIC Exp Lett 1:37–43
9. Tickoo O, Iyer R, Illikkal R, Newell D (2010) Modeling virtual machine performance: challenges and approaches. ACM Sigmetrics Perform Eval Rev 37(3):55–60
10. Wongthai W, van Moorsel A (2017) Logging system architectures for infrastructure as a service cloud. J Telecommun Electron Comput Eng (JTEC) 9(2):35–40
11. Ko RKL, Kirchberg M, Lee BS (2011) From system-centric to data-centric logging-accountability, trust and security in cloud computing. In: 2011 defense science research conference and expo (DSR). IEEE, pp 1–4
12. Ko RKL, Jagadpramana P, Mowbray M, Pearson S, Kirchberg M, Liang W, Lee BS (2011) TrustCloud: a framework for accountability and trust in cloud computing. In: 2011 IEEE world congress on services. IEEE, pp 584–588
13. Spring J (2011) Monitoring cloud computing by layer, part 1. IEEE Secur Priv 9(2):66–68
14. Spring J (2011) Monitoring cloud computing by layer, part 2. IEEE Secur Priv 9(3):52–55
15. Aceto G, Botta A, De Donato W, Pescapè A (2012) Cloud monitoring: definitions, issues and future directions. In: 2012 IEEE 1st international conference on cloud networking (CLOUDNET). IEEE, pp 63–67
16. NSA (2011) Cloud security considerations. The National Security Agency, Tech. Rep
17. Wongthai W, Rocha LF, van, Moorsel A (2013) A generic logging template for infrastructure as a service cloud. In: 2013 27th international conference on advanced information networking and applications workshops. IEEE, pp 1153–1160
18. Wongthai W, Van Moorsel A (2016) Performance measurement of logging systems in infrastructure as a service cloud. ICIC Exp Lett 2:347–354
19. Wongthai W, Van Moorsel A (2016) Quality analysis of logging system components in the cloud. Lecture notes in electrical engineering, vol 376. Springer, Singapore, pp 651–662
20. Kopytov A (2012) SysBench manual. MySQLAB
21. Scheuner J, Cito J, Leitner P, Gall H (2015) Cloud workbench: benchmarking iaas providers based on infrastructure-as-code. In: Proceedings of the 24th international conference on world wide web. ACM, pp 239–242
22. Sefraoui O, Aissaoui M, Eleuldj M (2012) OpenStack: toward an open-source solution for cloud computing. Int J Comput Appl 55(3):38–42