



โมเดลที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำและ
ประสิทธิภาพการทำงานคอมพิวเตอร์เสมือนบนการประมวลผลแบบกลุ่มเมฆ



วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2565
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

โมเดลที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำและ
ประสิทธิภาพการทำงานคอมพิวเตอร์เสมือนบนการประมวลผลแบบกลุ่มเมฆ



วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา 2565
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

วิทยานิพนธ์ เรื่อง "โมเดลที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำและ
ประสิทธิภาพการทำงานคอมพิวเตอร์เสมือนบนการประมวลผลแบบกลุ่มเมฆ"

ของ สุรพงษ์ วิริยะ

ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการสอบวิทยานิพนธ์
(รองศาสตราจารย์ ดร.เอกรัฐ บุญเชียง)

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

..... กรรมการผู้ทรงคุณวุฒิภายใน
(รองศาสตราจารย์ ดร.จักรกฤษณ์ เสน่ห์ นมะหุต)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.ธนธรร พ้อคำ)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ดร.ณัฐพล คุ่มใหญ่โต)

..... กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.จันทร์จิรา พยัคฆ์เทศ)

อนุมัติ

.....
(รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์)

คณบดีบัณฑิตวิทยาลัย



ชื่อเรื่อง	โมเดลที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความ แม่นยำและประสิทธิภาพการทำงานคอมพิวเตอร์เสมือนบนการ ประมวลผลแบบกลุ่มเมฆ
ผู้วิจัย	สุรพงษ์ วิริยะ
ประธานที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.วินัย วงษ์ไทย
ประเภทสารนิพนธ์	วิทยานิพนธ์ ปร.ด. วิทยาการคอมพิวเตอร์, มหาวิทยาลัยนเรศวร, 2565
คำสำคัญ	ระบบบันทึกเหตุการณ์, การประมวลผลแบบกลุ่มเมฆ, ความแม่นยำ, ประสิทธิภาพ, คอมพิวเตอร์เสมือน

บทคัดย่อ

งานวิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อ 1) วางรูปแบบการทดสอบความสามารถในการรองรับการทำงานของระบบบันทึกเหตุการณ์ 2) สร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บนระบบการประมวลผลแบบกลุ่มเมฆ ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (IaaS) ที่มีประสิทธิภาพ และ 3) สร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ โดยใช้กระบวนการหาค่าที่เหมาะสม

ในการดำเนินการวิจัยนั้นผู้วิจัยได้ทำการวางรูปแบบการทดสอบปริมาณการใช้งานหน่วยประมวลผล และหน่วยความจำแรมระหว่างการทำงานของระบบบันทึกเหตุการณ์ในเครื่องคอมพิวเตอร์เสมือน และนำผลมาวิเคราะห์เพื่อนำไปวางรูปแบบในการสร้างโมเดลทางคณิตศาสตร์ โดยการจำลองการจัดสรรเครื่องคอมพิวเตอร์เสมือนให้กับระบบบันทึกเหตุการณ์ด้วยการใช้วิธีการในการหาค่าที่เหมาะสม และกำหนดข้อจำกัดจากระดับความปลอดภัยของไฟล์และระดับประสิทธิภาพการทำงานของ VM ออกมาในรูปแบบโมเดลทางคณิตศาสตร์ โดยสามารถปรับเปลี่ยนการทำงานได้ตามความต้องการของผู้ใช้บริการ และนำมาทำการตรวจสอบหาวิธีการจัดสรรระบบบันทึกเหตุการณ์จากการสร้างโมเดลทางคณิตศาสตร์ด้วยการพัฒนาโปรแกรมด้วยภาษาไพธอน ด้วยเทคนิคการค้นหาแบบสุ่มมาใช้ในการหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ และหาค่าที่เหมาะสมสำหรับระบบบันทึกเหตุการณ์ที่ไม่ได้ใช้งานมากที่สุด

โดยผลการวิจัย พบว่า 1) การวางรูปแบบในการทดสอบนั้นเริ่มตั้งแต่การทดสอบนั้นเมื่อมี

การใช้งานระบบบันทึกเหตุการณ์ที่เพิ่มขึ้นทำให้ปริมาณการใช้งานหน่วยประมวลผล และหน่วยความจำรวมเพิ่มขึ้นจากเดิมเป็นเท่าตัว 2) การสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์สามารถสร้างด้วยการใช้รูปแบบของการกำหนดข้อจำกัดและแบบไม่กำหนดข้อจำกัด โดยการใช้ฟังก์ชันการปรับ ได้เป็นโมเดลทางคณิตศาสตร์ คือ maximize $f(X) = (\sum_{j=1}^n O_j(X)) + \sum_{i=1}^m cost(U_i)$ ในการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมกับผู้ใช้บริการและผู้ให้บริการ โดยสามารถจัดสรรระบบบันทึกเหตุการณ์ได้สำหรับข้อมูลนำเข้าในหลายรูปแบบ และเลือกวิธีการที่สามารถนำเสนอได้อย่างถูกต้อง (3) การสร้างการปรับรูปแบบการทำงานด้วยโมเดลทางคณิตศาสตร์ที่ปรับเปลี่ยนผู้ใช้บริการได้ทั้งการยกเลิกการใช้บริการ การเพิ่มผู้ใช้บริการใหม่ ได้เป็นโมเดลทางคณิตศาสตร์ คือ maximize $f(X) = (\sum_{j=1}^n O_j(X)) + \sum_{i=1}^m cost(U_i) + \sum_{i=1}^m Scost(U_i)$ ซึ่งสามารถสร้างวิธีการจัดสรรระบบบันทึกเหตุการณ์จากการทำงานของโปรแกรมที่ผู้วิจัยได้พัฒนาด้วยภาษา python ได้ โดยไม่กระทบกับผู้ใช้บริการเดิม



Title	OPTIMAL LOGGING SYSTEM MODEL FOR RECALL AND VIRTUAL MACHINE PERFORMANCE ON CLOUD COMPUTING
Author	Surapong Wiriya
Advisor	Assistant Professor Dr. Winai Wongthai
Academic Paper	Ph.D. Dissertation in Computer Science - (Type 2.1), Naresuan University, 2022
Keywords	Logging system, Cloud computing, Recall, Virtual machine

ABSTRACT

The objective of this thesis is threefold: (1) Setting up a test pattern is crucial to evaluate the logging system's ability to support its operations effectively (2) develop a mathematical model for allocating logging systems on an efficient Infrastructure-as-a-Service (IaaS) cloud computing platform and (3) create a customizable working model for service providers and customer to allocate logging systems on IaaS cloud by utilizing an optimization process.

In conducting the research, the researcher established a test model to examine the utilization of the CPU and RAM during the operation of the logging system in the virtual computer. The obtained results were subsequently analyzed to form a mathematical model. This model simulates the allocation of a virtual machine to the logging system using optimization methods, while incorporating mathematical constraints based on file security level and VM performance level. The proposed work is adaptable to the needs of service customer and aims to investigate efficient allocation strategies for the logging system. Mathematical modeling was utilized to develop programs using the Python language. The random search algorithm was employed to determine an optimal allocation strategy for the logging system, aiming to find the most workload-free logging system.

The results of the research show that 1) the testing involves modeling the scenario where the utilization of the logging system increases, resulting in a doubling of both CPU and RAM from their original level. 2) a mathematical model for allocating logging systems can be developed using both constraint-based and unconstrained approaches, employing penalty functions. The objective function of the mathematical model is to maximize $f(X) = (\sum_{j=1}^n O_j(X)) + \sum_{i=1}^m \text{cost}(U_i)$ allocation efficiency, providing methods suitable for service providers and customers. It can handle input data in various formats and facilitate the selection of appropriate allocation methods. 3) the creation of a flexible work adjustment system with a mathematical model allows service customer to customize their allocations, including terminating services and adding new users. The objective function of the mathematical model is to maximize $f(X) = (\sum_{j=1}^n O_j(x)) + \sum_{i=1}^m \text{cost}(U_i) + \sum_{i=1}^m \text{Scost}(U_i)$, which maximizes allocation effectiveness, enables the allocation of logging systems without affecting existing users, executed through a Python program developed by the researcher.

ประกาศคุณูปการ

ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงในความกรุณาของ ผู้ช่วยศาสตราจารย์ ดร.วินัย วงษ์ไทย ประธานที่ปรึกษาวิทยานิพนธ์ ที่ได้อุทิศส่วสละเวลาอันมีค่ามาเป็นทั้งที่ปรึกษา พร้อมทั้งให้คำแนะนำตลอดระยะเวลาในการทำวิทยานิพนธ์ฉบับนี้ และขอกราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์อันประกอบไปด้วย รองศาสตราจารย์ ดร.เอกรัฐ บุญเสียง ประธานกรรมการสอบป้องกันวิทยานิพนธ์ รองศาสตราจารย์ ดร.จักรกฤษณ์ เสน่ห์ นมะหุต ผู้ช่วยศาสตราจารย์ ดร.ธนะธร พอค้า ผู้ช่วยศาสตราจารย์ ดร.จันทร์จิรา พยัคฆ์เทศ และ ดร.ณัฐพล คุ่มใหญ่โต กรรมการผู้ทรงคุณวุฒิภายใน ที่ได้กรุณาให้คำแนะนำตลอดจนแก้ไขข้อบกพร่องของวิทยานิพนธ์ด้วยความเอาใจใส่ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์และทรงคุณค่า

เหนือสิ่งอื่นใดขอกราบขอบพระคุณ บิดา มารดา รุ่นพี่ และเพื่อนของผู้วิจัยที่ทำให้กำลังใจและให้การสนับสนุนในทุก ๆ ด้านอย่างดีที่สุดเสมอมา

คุณค่าและคุณประโยชน์อันพึงจะมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบและอุทิศแต่ผู้มีพระคุณทุก ๆ ท่าน ผู้วิจัยหวังเป็นอย่างยิ่งว่า งานวิจัยนี้จะเป็นประโยชน์ต่อผู้ที่สนใจนำไปใช้ประโยชน์ หากมีข้อบกพร่องประการใดที่อาจจะเกิดขึ้นภายในวิทยานิพนธ์ ผู้วิจัยขอน้อมรับและยินดีรับฟังคำแนะนำซึ่งจะเป็นประโยชน์ในการพัฒนางานวิจัยต่อไป

สุรพงษ์ วิริยะ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	จ
ประกาศคุณูปการ.....	ช
สารบัญ.....	ซ
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฉ
บทที่ 1 บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
ปัญหาวิจัย.....	8
เป้าหมายของงานวิจัย.....	9
วัตถุประสงค์ของงานวิจัย.....	9
ขอบเขตของงานวิจัย.....	9
ประโยชน์ที่คาดว่าจะได้รับ.....	10
นิยามศัพท์เฉพาะ.....	10
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	12
การประมวลผลแบบกลุ่มเมฆ.....	12
ปัญหาด้านความปลอดภัยบน IaaS คลาวด์.....	18
การทำงานของระบบบันทึกเหตุการณ์บน IaaS คลาวด์.....	21
ความปลอดภัยของไฟล์และประสิทธิภาพของคอมพิวเตอร์เสมือน.....	23
กระบวนการหาค่าที่เหมาะสม (Optimization).....	30
ขั้นตอนวิธีการค้นหาแบบสุ่ม (Random Search Algorithm).....	36
การกำหนดรูปแบบการหาค่าที่เหมาะสม.....	38
เครื่องมือในการวัดประสิทธิภาพ.....	40

งานวิจัยที่เกี่ยวข้อง	41
กรอบแนวคิดงานวิจัย	54
บทที่ 3 วิธีดำเนินการวิจัย	56
วิธีการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์.....	57
วิธีการทดสอบปริมาณการใช้งาน CPU และ RAM ของระบบบันทึกเหตุการณ์บนเครื่อง คอมพิวเตอร์ของผู้ให้บริการ	59
แนวคิดวิธีการในการจัดสรรระบบบันทึกเหตุการณ์.....	62
วิธีการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์.....	66
วิธีการสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการในการจัดสรร ระบบบันทึกเหตุการณ์บน IaaS คลาวด์.....	78
บทที่ 4 ผลการวิจัย	86
ผลการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์.....	86
ผลการทดสอบปริมาณการใช้งาน CPU และ RAM ของระบบบันทึกเหตุการณ์บนเครื่อง คอมพิวเตอร์ของผู้ให้บริการ	88
ผลการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์.....	90
ผลการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์.....	92
ผลการสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการในการจัดสรร ระบบบันทึกเหตุการณ์บน IaaS คลาวด์.....	108
บทที่ 5 บทสรุป.....	135
สรุปผลการวิจัย.....	137
อภิปรายผล.....	140
ข้อเสนอแนะ.....	143
บรรณานุกรม.....	145
ภาคผนวก.....	155
ประวัติผู้วิจัย.....	166

สารบัญตาราง

	หน้า
ตาราง 1 แสดงระดับความปลอดภัยของไฟล์ของผู้ใช้บริการ	28
ตาราง 2 แสดงระดับประสิทธิภาพของ VM	29
ตาราง 3 แสดงการสรุปการจัดการทรัพยากรบนคลาวด์ด้วยการทำ optimization	34
ตาราง 4 แสดงการเปรียบเทียบข้อดี ข้อด้อยของอัลกอริธึมในการหาค่าที่เหมาะสม	37
ตาราง 5 แสดงผลสรุปงานวิจัยที่เกี่ยวข้อง	45
ตาราง 6 แสดงการเปรียบเทียบการใช้ constrained และ unconstrained สำหรับข้อมูลไม่ซับซ้อน	77
ตาราง 7 แสดงการเปรียบเทียบการใช้ constrained และ unconstrained สำหรับข้อมูลซับซ้อน ..	78
ตาราง 8 แสดงเปรียบเทียบการทำงานในกรณี best case และ worst case แบบ unconstrained optimization	108
ตาราง 9 แสดงเปรียบเทียบการทำงานในกรณี best case และ worst case แบบ unconstrained optimization สำหรับการปรับรูปแบบการทำงาน	134

สารบัญภาพ

	หน้า
ภาพ 1 แสดงสถาปัตยกรรม IaaS คลาวด์.....	2
ภาพ 2 แสดงรูปแบบการให้บริการคลาวด์.....	14
ภาพ 3 แสดงสถาปัตยกรรม IaaS คลาวด์.....	17
ภาพ 4 แสดงสภาพแวดล้อมการทำงานระหว่าง Host และ VM	21
ภาพ 5 แสดงกรอบแนวคิดการวิจัย	54
ภาพ 6 แสดงการออกแบบการทำงานของระบบบันทึกเหตุการณ์แบบ 1 คู่, 2 คู่ และหลายคู่.....	57
ภาพ 7 แสดงติดตั้งระบบ IaaS คลาวด์.....	59
ภาพ 8 แสดงคำสั่งการวัดปริมาณการใช้งาน CPU	60
ภาพ 9 แสดงคำสั่งการวัดปริมาณการใช้งาน RAM.....	60
ภาพ 10 แสดงการรันแอปพลิเคชัน read บนเครื่อง VM	60
ภาพ 11 แสดงการรันแอปพลิเคชัน read บนเครื่อง VM พร้อมด้วยการรันแอปพลิเคชัน logger ...	61
ภาพ 12 แสดงการทำงานของ Logger แบบรวมศูนย์	62
ภาพ 13 แสดงช่วงเวลาการทำงานของล็อกเกอร์.....	63
ภาพ 14 แสดงกรณีการทำงานของล็อกเกอร์ 1 ตัว ต่อ VM 3 เครื่อง	63
ภาพ 15 แสดงกรณีการทำงานของล็อกเกอร์ 1 ตัว ต่อ VM 2 เครื่อง	64
ภาพ 16 แสดงวิธีการการจัดสรรล็อกเกอร์ให้กับผู้ใช้บริการ	65
ภาพ 17 แสดงข้อมูลนำเข้าสำหรับการสร้างโมเดลทางคณิตศาสตร์	72
ภาพ 18 แสดงวิธีการจัดสรร VM ให้กับล็อกเกอร์	73
ภาพ 19 แสดงผังงานการทำงานของกรหาวิธีการที่เหมาะสมด้วยรูปแบบมีข้อจำกัด	75
ภาพ 20 แสดงผังงานการทำงานของกรหาวิธีการที่เหมาะสมด้วยรูปแบบไม่มีข้อจำกัด.....	76
ภาพ 21 แสดงข้อมูลนำเข้ากรณีที่ไม่ซับซ้อน.....	77
ภาพ 22 แสดงข้อมูลนำเข้ากรณีที่ซับซ้อน	78
ภาพ 23 แสดงผลลัพธ์การจัดสรรระบบบันทึกเหตุการณ์.....	78
ภาพ 24 แสดงเพิ่มผู้ใช้บริการรายใหม่ โดยกำหนดค่าให้เป็น 1	79
ภาพ 25 แสดงผังงานการทำงานของกรสร้างการปรับรูปแบบ	84
ภาพ 26 แสดงการทำงานของแอปพลิเคชันล็อกเกอร์	86

ภาพ 27 แสดงการทำงานของแอปพลิเคชัน read อ่านไฟล์ s.txt	87
ภาพ 28 แสดงการรันแอปพลิเคชัน read บนเครื่อง VM พร้อมด้วยการรันแอปพลิเคชัน logger ...	87
ภาพ 29 แสดงผลการใช้งานของ CPU (%) ของโฮสต์ในกรณีที่ VMs มีการทำงานแบบไม่รันแอปพลิเคชัน read กับ VMs รันแอปพลิเคชัน read	88
ภาพ 30 แสดงผลการใช้งานของ RAM (%) ของโฮสต์ในกรณีที่ VM มีการรันแอปพลิเคชัน read ..	89
ภาพ 31 แสดงผลการใช้งานของ CPU (%) ของโฮสต์ในกรณีที่โฮสต์มีการรันแอปพลิเคชัน Logger และ VMs มีการรันแอปพลิเคชัน read.....	89
ภาพ 32 แสดงผลการใช้งานของ RAM (%) ของโฮสต์ในกรณีที่โฮสต์มีการรันแอปพลิเคชัน Logger และ VMs มีการรันแอปพลิเคชัน read.....	90
ภาพ 33 แสดงการทำงานของ Logger แบบรวมศูนย์	91
ภาพ 34 แสดงวิธีการจัดสรรล็อกเกอร์ให้กับผู้ใช้บริการ และผู้ให้บริการ	92
ภาพ 35 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์และ VM เท่ากัน สำหรับกรณีที่ดีที่สุด.....	95
ภาพ 36 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์และ VM เท่ากัน สำหรับกรณีที่แย่ที่สุด.....	97
ภาพ 37 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์น้อยกว่าจำนวนของ VM สำหรับกรณีที่ดีที่สุด	99
ภาพ 38 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์น้อยกว่าจำนวนของ VM สำหรับกรณีที่แย่ที่สุด	101
ภาพ 39 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์มากกว่าจำนวนของ VM สำหรับกรณีที่ดีที่สุด.....	103
ภาพ 40 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์มากกว่าจำนวนของ VM สำหรับกรณีที่แย่ที่สุด.....	105
ภาพ 41 แสดงผลลัพธ์จากการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในรูปแบบเดิม	109
ภาพ 42 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก.....	110
ภาพ 43 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีผู้ใช้บริการยกเลิก สำหรับกรณีที่ดีที่สุด	110
ภาพ 44 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีผู้ใช้บริการยกเลิก สำหรับกรณีที่แย่ที่สุด	113

ภาพ 45 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ใช้บริการใหม่	115
ภาพ 46 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการใหม่ สำหรับกรณีที่ดีที่สุด	116
ภาพ 47 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการใหม่ สำหรับกรณีที่แย่ที่สุด	119
ภาพ 48 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการ ใหม่	121
ภาพ 49 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่ สำหรับกรณีที่ดีที่สุด	122
ภาพ 50 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่ สำหรับกรณีที่แย่ที่สุด	125
ภาพ 51 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ใช้บริการปรับระดับความปลอดภัย และระดับประสิทธิภาพ	127
ภาพ 52 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการปรับ ระดับความปลอดภัยและระดับประสิทธิภาพ กรณีที่ดีที่สุด	128
ภาพ 53 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการปรับ ระดับความปลอดภัยและระดับประสิทธิภาพ กรณีที่แย่ที่สุด	132

บทที่ 1

บทนำ

เนื้อหาในบทที่ 1 ผู้วิจัยจะขอกล่าวถึงภาพรวมของเรื่องราวทั้งหมดในวิทยานิพนธ์เรื่อง โมเดลที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำและประสิทธิภาพ การทำงานคอมพิวเตอร์เสมือนบนการประมวลผลแบบกลุ่มเมฆ เพื่อที่ผู้อ่านจะได้เข้าใจง่ายก่อนที่จะศึกษาในรายละเอียดเชิงลึกในบทต่อไป โดยในบทที่ 1 จะประกอบไปด้วยเนื้อหาและรายละเอียด โดยแบ่งแยกตามหัวข้อดังนี้

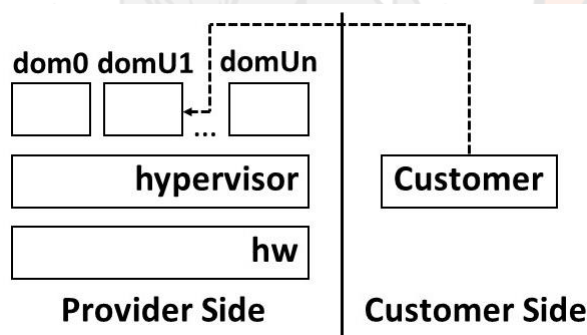
1. ความเป็นมาและความสำคัญของปัญหา
2. ปัญหาวิจัย
3. เป้าหมายของงานวิจัย
4. วัตถุประสงค์ของงานวิจัย
5. ขอบเขตของงานวิจัย
6. ประโยชน์ที่คาดว่าจะได้รับ
7. นิยามศัพท์เฉพาะ

ความเป็นมาและความสำคัญของปัญหา

การประมวลผลแบบกลุ่มเมฆหรือคลาวด์คอมพิวติ้ง (Cloud Computing) โดยงานวิทยานิพนธ์นี้ขอเรียกว่า คลาวด์ (Cloud) คลาวด์เป็นเทคโนโลยีที่มีการเปลี่ยนวิธีการสร้างและการใช้ บริการข้อมูล โดยก่อนที่จะเข้าสู่ในยุคของคลาวด์ การให้บริการคอมพิวเตอร์ เช่น เว็บไซต์ขนาดใหญ่ หรือเซิร์ฟเวอร์ไฟล้นั้น จะมีค่าใช้จ่ายด้านต้นทุนจำนวนมาก ได้แก่ พื้นที่ในการจัดเก็บข้อมูล การเชื่อมต่อเครือข่าย เซิร์ฟเวอร์ หรือแม้กระทั่งหลังจากที่ได้มีการลงทุนไปแล้วองค์กรยังต้องการบุคลากร ทางด้านไอทีและนักพัฒนามาสร้าง ติดตั้ง และบำรุงรักษาระบบอีกด้วย ถ้าเป็นในรูปแบบการทำงานของคลาวด์นั้นองค์กรสามารถสร้างทรัพยากรการประมวลผลได้ตามความต้องการ และสามารถ ประกอบเป็นแอปพลิเคชันได้อย่างรวดเร็ว (Cable, 2015) คลาวด์เป็นการใช้เทคโนโลยีการจำลอง เสมือนเพื่อให้ตัวเองสามารถจัดเก็บและประมวลผลข้อมูลขนาดใหญ่ได้ (Bhawiyuga, Kartikasari, Amron, Pratama, & Habibi, 2019) ซึ่งคลาวด์มีความสำคัญมากขึ้นสำหรับเทคโนโลยีสารสนเทศ หรือไอที (Alshamrani, 2019; Kaur & Kaur, 2017) ปัจจัยหนึ่งที่ทำให้แอปพลิเคชันอัจฉริยะนั้น สามารถพัฒนาชีวิตความเป็นอยู่ของมนุษย์ได้ นั่นก็คือระบบคลาวด์ ซึ่งสามารถจัดเก็บและ

ประมวลผลข้อมูลซึ่งเกิดมาจากกิจกรรมของมนุษย์ อย่างไรก็ตาม ความปลอดภัยของข้อมูลนี้ก็มี ความสำคัญเช่นกัน

การประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (Infrastructure as a Service : IaaS) เช่น Amazon Elastic Compute Cloud (Chaisiri, 2011) มุ่งเน้นไปที่การนำเสนอเครื่องเสมือน (Virtual Machine: VM) ให้แก่ลูกค้าที่ใช้บริการ IaaS ลูกค้าสามารถเช่าเครื่องเสมือนและสามารถเข้าถึงโดยผ่านทางอินเทอร์เน็ต ส่วนหลัก ๆ ของ VM จะประกอบไปด้วยการจัดเก็บข้อมูล และเครือข่าย ลูกค้าสามารถติดตั้งระบบปฏิบัติการที่ต้องการใช้ใน VM ได้ ประเภทของคลาวด์แบบ IaaS นั้นมีการใช้งานอย่างแพร่หลายในหลากหลายกิจกรรม เช่น งานในภาครัฐ งานด้านการศึกษา หรือการทดลองด้านการแพทย์ เป็นต้น การให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ของคลาวด์ เป็นการจัดเตรียมในส่วนของเครื่องคอมพิวเตอร์เสมือน (Virtual Machine, VMs) ไว้สำหรับผู้ใช้บริการ โดยผู้ใช้งานสามารถติดตั้งในส่วนจากระบบปฏิบัติการได้ตาม การใช้งานของตนเอง (Inokuchi & Kourai, 2020) โดยสถาปัตยกรรมของ IaaS คลาวด์ แบ่งออกเป็น 2 ส่วน ได้แก่ ฝ่ายผู้ให้บริการ (Provider Side) และฝั่งผู้ใช้บริการ (Customer Side) ดัง ภาพ 1



ภาพ 1 แสดงสถาปัตยกรรม IaaS คลาวด์

ซึ่งภายในโครงสร้างจะประกอบไปด้วยส่วนสำคัญคือ hw (ย่อมาจาก hardware) คือ เครื่องคอมพิวเตอร์ที่ใช้สำหรับสร้าง IaaS คลาวด์ hypervisor คือ ซอฟต์แวร์ที่สามารถทำให้เครื่องคอมพิวเตอร์หนึ่งเครื่องสามารถรันระบบปฏิบัติการได้มากกว่าหนึ่งระบบในเครื่องเดียวกัน ซึ่งซอฟต์แวร์ที่เป็นแบบฟรี (Open Source) ที่ใช้ในการติดตั้งเช่น Xen, dom0 (ย่อมาจาก Domain 0) อ่านว่า ดอมซีโร่ เป็นคำศัพท์เฉพาะทางที่ใช้กับ Xen หรือ Host VM คือ ระบบบริหารจัดการ domU และ hw และทำงานเมื่อ hypervisor ถูกเปิดใช้งาน โดยในงานวิทยานิพนธ์นี้จะขอเรียกว่า โฮสต์ (Host) และ domU (ย่อมาจาก Domain User) อ่านว่า ดอมยู เป็นคำศัพท์เฉพาะทางที่ใช้กับ Xen

หรือ Guest VM คือ VMs หรือคอมพิวเตอร์เสมือน ที่ผู้ใช้บริการเป็นเจ้าของ ซึ่งสามารถใช้บริการได้มากกว่า 1 domU เช่น domU1, domU2, ..., domUn โดยในงานวิทยานิพนธ์นี้จะขอเรียกว่า VM

ในงานวิทยานิพนธ์นี้มุ่งไปที่ระบบคลาวด์แบบสาธารณะเนื่องจากเป็นที่นิยมใช้ในองค์กร ตามที่ Vailshery (2022) ได้รายงานไว้ว่า ในปี 2022 ตลาดการให้บริการคลาวด์สาธารณะทั่วโลกคาดว่าจะเติบโตประมาณ 18.8 เปอร์เซ็นต์ ซึ่งมีมูลค่าประมาณ 490 พันล้านดอลลาร์สหรัฐ ดังนั้นองค์กรผู้ให้บริการคลาวด์แบบสาธารณะพยายามที่จะเพิ่มประสิทธิภาพและความปลอดภัย ในขณะที่ลดค่าใช้จ่าย และรับประกันความพร้อมใช้งานของบริการและแอปพลิเคชันที่กว้างขึ้น และการให้บริการ IaaS คลาวด์ นั้นรายได้มูลค่าโดยรวมในปี 2020 อยู่ประมาณ 5 หมื่นล้านดอลลาร์สหรัฐ และคาดว่าจะรายได้ IaaS บนคลาวด์สาธารณะจะเพิ่มขึ้นอย่างมากในอีกไม่กี่ปีข้างหน้า จะมีรายได้รวมมูลค่าสูงถึง 156 พันล้านดอลลาร์สหรัฐภายในปี 2023 โดยมีบริษัทที่ใหญ่ที่สุดในการให้บริการ IaaS cloud ได้แก่ Amazon (Web service), Microsoft (Azure) และ Google (Compute Engine) รวมถึงบริษัทเทคโนโลยีข้ามชาติสัญชาติจีนอย่าง Alibaba (Vailshery, 2022) ในปี 2022 การใช้จ่ายด้านโครงสร้างพื้นฐานไอทีบนคลาวด์คาดว่าจะสูงถึงเกือบ 9 หมื่นล้านดอลลาร์สหรัฐ โครงสร้างพื้นฐานของคลาวด์สาธารณะยังคงขยายตัวและเป็นตัวขับเคลื่อนการใช้จ่ายด้านไอทีจำนวนมาก โดยข้อดีของการที่มีการใช้คลาวด์แบบสาธารณะบนโครงสร้างแบบ IaaS คลาวด์นั้นยังเห็นแนวโน้มของการเติบโตของตลาดการให้บริการคลาวด์โดยนักวิจัยต่าง ๆ สามารถหาวิธีในการป้องกันปัญหาด้านความปลอดภัยที่เกิดขึ้นบนคลาวด์ได้ เช่น งานวิจัยของ Taubmann et al. (2019) ได้ใช้เครื่องมือที่ชื่อ TwinPorter ในการตรวจสอบการย้ายข้อมูลบน IaaS คลาวด์ด้วยการตรวจสอบผ่าน VMI monitoring

ในเรื่องของความปลอดภัยของคลาวด์ Cloud Security Alliance ได้มีการจัดทำรายงานภัยคุกคามที่เกิดขึ้นกับคลาวด์ในทุก ๆ 3 ปี โดยในปี 2019 (CSA, 2019) ได้รายงานภัยคุกคามของคลาวด์ที่ชื่อว่า Top Threats to Cloud Computing: The Egregious Eleven ซึ่งประกอบไปด้วย 1) การละเมิดข้อมูล 2) การกำหนดค่าผิดพลาดและการควบคุมการเปลี่ยนแปลงที่ไม่ดีพอ 3) การขาดสถาปัตยกรรมและกลยุทธ์การรักษาความปลอดภัยบนคลาวด์ 4) การระบุตัวตน ข้อมูลประจำตัว การเข้าถึงการจัดการคีย์ 5) บัญชีผู้ใช้ถูกขโมย 6) ภัยคุกคามที่เกิดจากบุคคลภายใน 7) อินเทอร์เน็ตเฟสและส่วนต่อประสานโปรแกรมประยุกต์ที่ไม่ปลอดภัย 8) ระดับขั้นการควบคุมที่อ่อนแอ 9) ความล้มเหลวของผู้บริการและการจัดการ 10) การมีข้อจำกัดในมุมมองของการใช้งานคลาวด์ และ 11) การใช้บริการคลาวด์ในทางที่ผิด และในปี 2022 CSA (CSA, 2022) ได้รายงานภัยคุกคามของการประมวลผลคลาวด์ในรายงาน Top Threats to Cloud Computing Pandemic Eleven โดยมีรายละเอียดการคุกคามดังนี้ คือ

1. การระบุตัวตน ข้อมูลประจำตัว การเข้าถึงการจัดการคีย์
2. อินเทอร์เน็ตและส่วนต่อประสานโปรแกรมประยุกต์ที่ไม่ปลอดภัย
3. การกำหนดค่าผิดพลาดและการควบคุมการเปลี่ยนแปลงที่ไม่ดีพอ
4. การขาดสถาปัตยกรรมและกลยุทธ์การรักษาความปลอดภัยบนคลาวด์
5. การพัฒนาซอฟต์แวร์ที่ไม่ปลอดภัย
6. ทรัพยากรของบุคคลที่สามไม่ปลอดภัย
7. ช่องโหว่ของระบบ
8. การเปิดเผยข้อมูลบนคลาวด์โดยไม่ได้ตั้งใจ
9. การกำหนดค่าผิดพลาดและการใช้ประโยชน์จากปริมาณงานแบบไร้เซิร์ฟเวอร์และคอนเทนเนอร์
10. การก่ออาชญากรรม / แฮกเกอร์ / ภัยคุกคามต่อเนื่องขั้นสูง
11. การกรองข้อมูลที่เกิดขึ้นบนคลาวด์

และในรายงานของ Cyber Security Insiders เรื่อง 2022 Cloud Security Report ได้สอบถามผู้เชี่ยวชาญด้านความปลอดภัยทางไซเบอร์ในประเด็นที่กังวลมากที่สุดได้แก่ การกำหนดค่าความปลอดภัยของคลาวด์ยังคงเป็นความเสี่ยงด้านความปลอดภัยบนคลาวด์ที่ใหญ่ที่สุดตาม 62% ตามมาด้วยอินเทอร์เน็ต/API ที่ไม่ปลอดภัย (54%) การกรองข้อมูลที่ไม่ละเอียดอ่อน (51%) และการเข้าถึงโดยไม่ได้รับอนุญาต (50%) อย่างไรก็ตาม ปัญหาความปลอดภัยของระบบคลาวด์ IaaS นั้นสำคัญมากเมื่อถูกนำไปใช้งานจริงและจำเป็นต้องได้รับการบรรเทาเมื่อเกิดปัญหา จึงทำให้มีหน่วยงานที่ได้มีการดำเนินการด้านการวิจัยที่เกี่ยวกับความปลอดภัยบนคลาวด์ขึ้น เช่น Cyber Security Insiders, องค์กร CSA ที่มีนักวิจัยที่ทำงานวิจัยเกี่ยวกับความปลอดภัยบนคลาวด์ เพื่อหาแนวทางในการป้องกันหรือบรรเทาผลกระทบที่เกิดจากการใช้งานบนระบบคลาวด์ เช่น ในงานวิจัยของ Ko & Will (2014), Shamseddine et al. (2022), Fargo et al. (2020) ที่ได้มีการใช้ Virtual Machine Introspection (VMI) ในรูปแบบเครื่องมือที่แตกต่างกันไปมาใช้ในการตรวจสอบหาความผิดปกติของการทำงานจากผู้ไม่ประสงค์ดี โดย VMI เป็นการนำข้อมูลการทำงานจากหน่วยความจำแรมของเครื่องเสมือนมาพิจารณาการถูกโจมตีหรือการเข้าถึงไฟล์ข้อมูลซึ่งเป็นเทคนิคในรูปแบบหนึ่งของ intrusion detection system (IDS) โดยได้มีการใช้ Virtual Machine Introspection (VMI) ในการเข้าถึง Memory ของเครื่อง VMs ได้ โดยใช้ไลบรารี LibVMI ในการดึงข้อมูลจาก VMs ซึ่งไลบรารี LibVMI เป็นการทำงานหลักของวิธีการของ VMI (Borisaniya & Patel, 2019) เมื่อสามารถเข้าถึง Memory ของเครื่อง VM ได้นั้นก็จะสามารถนำข้อมูลโปรเซสการทำงานของไฟล์งานต่าง ๆ ออกมาใช้ในการพิจารณาการถูกโจมตีได้ สิ่งที่สำคัญของการบังคับซึ่งถึงความสามารถของเครื่องมือในการป้องกันหรือบรรเทาผลกระทบที่เกิดจากการใช้งานบนระบบคลาวด์นั้นคือการวัดประสิทธิภาพการทำงาน ใน

งานวิจัยหลากหลายงานวิจัยที่เกี่ยวข้องกับการป้องกันหรือบรรเทาผลกระทบที่เกิดจากการใช้งานบนระบบคลาวด์ได้มีการวัดประสิทธิภาพ โดย แบ่งการวัดประสิทธิภาพการทำงานได้ดังนี้ 1) การวัดประสิทธิภาพด้านเวลาดังที่ปรากฏในงานวิจัยของ Patrascu & Patriciu (2015) เป็นการวัดระยะเวลาในการดึงข้อมูลออกมาจาก VMs เมื่อเทียบขนาดของไฟล์ ในการใช้ cloud forensic module ที่ติดตั้งไว้ในส่วนของ Management Layer Yang et al. (2017) รองรับการเข้าถึงข้อมูลที่รวดเร็วโดยการให้สิทธิ์และการบันทึกอย่างรวดเร็วในการทำงานของ Cloud Data Accountability Framework Rane et al. (2021) วัดประสิทธิภาพเวลาตอบสนอง (Respond Time) เวลาในการตรวจสอบข้อมูลนำเข้า (Proof Insertion Time) และเวลาในการพิสูจน์ยืนยัน (proof verification time) ในการปกป้องข้อมูล Log File ของผู้ใช้งานคลาวด์ด้วย BlockSLaaS Ko & Will (2014) วัดประสิทธิภาพในเรื่องของเวลาในการดึงข้อมูลโดย System Call ที่เปรียบเทียบเวลาของค่าเริ่มต้นกับเวลาเมื่อใช้งาน Progger 2) การวัดประสิทธิภาพด้านทรัพยากรดังที่ปรากฏในงานวิจัยของ Csáti, Pelle, & Toka. (2022) ในการเพิ่มประสิทธิภาพและทรัพยากรของผู้ใช้บริการด้วยการวัดปริมาณการใช้หน่วยประมวลผล (CPU Usage) และปริมาณการใช้หน่วยความจำแรม (RAM Usage) สำหรับปริมาณการไหลของข้อมูลที่แตกต่างกัน Mavridis & Karatza (2017) ได้ทำการเปรียบเทียบการใช้ทรัพยากรของ Apache Hadoop และ Apache Spark จากผู้ให้บริการ IaaS โดยทำงานด้วยแอปพลิเคชันเดียวกันในทั้งสองเฟรมเวิร์ก ด้วยอินพุต Log File ที่มีขนาด 5.5 GB และทำการบันทึกข้อมูลการใช้ CPU, หน่วยความจำหลัก, ดิสก์ และเครือข่ายจากโหนดทั้งหมด 5 โหนด 3) การวัดประสิทธิภาพด้านความถูกต้องของการตรวจจับความผิดปกติ ในงานวิจัยของ Hickman et al. (2018) ทำการทดสอบการตรวจจับบรรทัดของซอร์สโค้ดจะจับคู่กับ Log Message ในการหาความผิดปกติ พบว่ามีความเชื่อมั่นอยู่ที่ 92% และสำหรับการทดสอบการคาดการณ์ข้อมูลใหม่นั้นพบว่ามี ความเชื่อมั่นอยู่ที่ 88.2% - 95.8% ของเวลาที่ใช้ทั้งหมด

ในงานวิจัยของ Auxsorn (2020), Chan-In (2017), Jaiboo (2020), และ Wongthai (2017) ได้มีการใช้ระบบบันทึกเหตุการณ์ ซึ่งเป็นวิธีการหนึ่งที่สำคัญสำหรับช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามขึ้นต่อคลาวด์ โดยภายในระบบบันทึกเหตุการณ์ได้จัดเก็บหลักฐาน Log File ไว้เป็นหลักฐานและหลักฐานจาก Log File สามารถนำไปช่วยหาบุคคลที่กระทำผิดมารับผิดชอบได้ โดยการทำงานของระบบ Logging System ใน IaaS คลาวด์ แบ่งการทำงานออกเป็น 2 ส่วน ได้แก่ 1) ส่วนที่เป็นผู้ให้บริการ (Host) ในส่วนนี้จะได้ทำการติดตั้งล็อกเกอร์ (Logger) ซึ่งเป็นโปรเซสสำหรับใช้ในการตรวจจับบันทึกเหตุการณ์ที่เกิดขึ้นใน VM ซึ่งจะเป็นการทดลองการทำงานในการตรวจจับโปรเซสการทำงานของไฟล์ในเครื่อง VM 1 เครื่อง โดยล็อกเกอร์ 1 ล็อกเกอร์ เท่านั้น และยังไม่มีการวัดปริมาณการใช้งานของ CPU และ RAM ผู้วิจัยจึงได้มีแนวคิดในการที่จะทำการทดสอบการทำงานของล็อกเกอร์ในการตรวจจับโปรเซสการทำงานของไฟล์ที่เครื่อง VM ที่มากกว่า 1 เครื่อง เพื่อวัด

ปริมาณการใช้งานของ CPU และ RAM ในการรองรับการทำงานได้สูงสุดเท่าไร และจะได้นำมาซึ่งการจัดสรรทรัพยากรที่เหมาะสม

การจัดการทรัพยากรนั้นเป็นประเด็นหลักในการประมวลผลแบบคลาวด์ เช่นเดียวกับการประมวลผลรูปแบบอื่น ๆ เนื่องจากความพร้อมใช้งานของทรัพยากรที่จำกัด ผู้ให้บริการคลาวด์จึงเห็นเป็นเรื่องที่ยากมากที่จะจัดหาทรัพยากรตามที่คุณใช้บริการร้องขอทั้งหมด (Parikh, Patel, & Prajapati, 2017) โดยการจัดการทรัพยากรเป็นปัญหาที่สำคัญที่สุดในโครงสร้างพื้นฐานระบบคลาวด์ เนื่องจากมีวัตถุประสงค์ที่ขัดแย้งกันหลายประการไม่ว่าจะเป็นการรักษาประสิทธิภาพ การลดพลังงาน หรือการลดต้นทุน เป็นต้น ซึ่งจำเป็นต้องมีการปรับให้เหมาะสมในการจัดการทรัพยากร ในการจัดการทรัพยากรบนคลาวด์การรวม VMs และการจัดสรรทรัพยากรนั้นใช้ในการแก้ปัญหาความผันผวนของปริมาณงาน โดยการย้าย VM จากโฮสต์หนึ่งไปยังอีกโฮสต์หนึ่งเพื่อลดจำนวนโฮสต์ที่ใช้งานและเพื่อเป็นการประหยัดพลังงาน (Aldossary, 2021) ในงานวิจัยของ Ye et al. (2011) ได้นำเสนอเฟรมเวิร์กในการย้าย VM หลายเครื่องตามการจองทรัพยากรที่แตกต่างกัน เพื่อให้มีประสิทธิภาพในการย้ายข้อมูลแบบขนานและการรับรู้ภาระงาน งานวิจัยของ Zhao et al. (2015) ได้นำเสนอวิธีการจัดวาง VM ตามประสิทธิภาพของบริการ โดยนำลักษณะการใช้ทรัพยากรมาใช้ในการพิจารณาในการดูว่า VM จะไปอยู่ที่โฮสต์ใดที่จะให้เกิดความเหมาะสมเพื่อรับประกันประสิทธิภาพของ VM และรับประกันคุณภาพประสบการณ์ของลูกค้า (QoE) งานวิจัยของ Malekloo et al. (2018) ได้นำเสนอแนวทาง Multi-objective Ant Colony Optimization (MACO) ในการรวบรวมและจัดวาง VM โดยมีเป้าหมายเพื่อลดการใช้พลังงาน การสิ้นเปลืองทรัพยากรหน่วยประมวลผล และต้นทุนการสื่อสาร มีเป้าหมายเพื่อลดการละเมิดข้อตกลงการให้บริการระหว่างผู้ให้บริการและผู้ให้บริการ (Service Level Agreement: SLA) การย้าย VM และจำนวนโฮสต์ที่ใช้งานอยู่ สำหรับการใช้งานทรัพยากรของคอมพิวเตอร์นั้น เมื่อพบว่าปริมาณการใช้งานของ CPU มีระดับสูง 80% ขึ้นไป เป็นระยะเวลานาน ประสิทธิภาพของระบบหรือแอปพลิเคชันจะได้รับผลกระทบด้วย (Microsoft, 2022) ในการทำงานของ IaaS คลาวด์นั้นการจัดสรรทรัพยากรเป็นสิ่งที่จำเป็นอย่างยิ่งเพื่อไม่ให้งานของ VM เกิดปัญหาในการทำงาน

กระบวนการหาค่าที่เหมาะสม (Optimization) เป็นสิ่งสำคัญในการสร้างระบบให้มีประสิทธิภาพ (Zeng, 2023) และเป็นขั้นตอนวิธีในการตัดสินใจเลือกผลลัพธ์ที่ดีที่สุดในการแก้ปัญหาที่เหมาะสม (Optimization Problem) ในงานวิจัยทางด้านคลาวด์ ได้มีการนำเทคนิคการหาค่าที่เหมาะสมมาใช้ในการทำงานให้มีประสิทธิภาพ เช่น ในงานวิจัยของ Hosseini et al. (2021) ได้ใช้อัลกอริทึม Whale Optimization Algorithm ในการแก้ปัญหาในเรื่องของการจัดสรรทรัพยากรให้เหมาะสมกับความต้องการในการทำงานของผู้ใช้บริการบนคลาวด์ เพื่อลดระยะเวลาที่โปรแกรมทำงานจากการร้องขอของผู้ใช้บริการ Ala'a Al-Shaikh (2016) ได้นำวิธีการในการหาวิธีการที่เหมาะสมมา

ใช้ในการแก้ปัญหาเรื่องของการใช้ทรัพยากรบนระบบคลาวด์ได้ ด้วยการใช่วิธี Greedy Wei (2020) ได้ทำการแก้ปัญหาเรื่องของการโหลดที่ไม่สมดุล ความเร็วในการรวมกันซ้ำ หรือใช้ทรัพยากรของ VM ที่ต่ำ ด้วยการปรับปรุงวิธีการของ Ant Colony Optimization บนคลาวด์ เป็นต้น ซึ่งจะเห็นได้ว่าการหาค่าที่เหมาะสมบนคลาวด์นั้นจะเป็นการทำในส่วนของการจัดสรรทรัพยากร เช่น ซีพียู แรม ฮาร์ดดิสก์ เป็นต้น แต่ยังไม่มีการดำเนินการทำในส่วนของหาวิธีในการจัดสรรระบบบันทึกเหตุการณ์

จากเนื้อหาทั้งหมดที่กล่าวมาในหัวข้อความเป็นมาของปัญหาวิจัยนี้ สามารถสรุปได้ว่าระบบบันทึกเหตุการณ์นั้นมีความสำคัญในการที่จะช่วยป้องกันการเกิดปัญหาในการโจรกรรมหรือไม่ประสงค์ดีในการเข้าถึงการทำงานของไฟล์ในเครื่อง VM ของผู้ใช้งาน บน IaaS คลาวด์ได้ โดยการทำงานระบบบันทึกเหตุการณ์นั้นจะทำการตรวจจับโปรเซสการทำงานของไฟล์ในเครื่อง VM แบบ 1 ล็อกเกอร์ต่อ 1 VM ถ้าจะให้การทำงานมีประสิทธิภาพล็อกเกอร์ 1 ล็อกเกอร์ ควรที่จะตรวจจับโปรเซสการทำงานของไฟล์ในเครื่อง VM ได้หลาย ๆ เครื่อง เพื่อที่จะสามารถลดปริมาณการใช้งานของทรัพยากร CPU และ RAM ในฝั่งเครื่องผู้ให้บริการในการใช้ทรัพยากรให้น้อยที่สุด เพื่อไปเพิ่มประสิทธิภาพในการทำงานของฝั่งผู้ให้บริการ โดยมีการใช้งานกระบวนการการหาจุดที่เหมาะสมมาใช้ในการหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมกับผู้ให้บริการและผู้ใช้งานเพื่อให้เครื่องคอมพิวเตอร์สามารถทำงานได้เต็มประสิทธิภาพมากที่สุด ซึ่งจะเห็นได้ว่าการหาค่าที่เหมาะสมบนคลาวด์นั้น การจัดสรรพยากรณ์ให้กับ VM นั้นจะพิจารณาจาก ซีพียู แรม ฮาร์ดดิสก์ เป็นต้น ทางผู้วิจัยจึงได้นำระบบบันทึกเหตุการณ์ ซึ่งเป็นระบบที่ดูแลในด้านความปลอดภัยของ VM มาสร้างโมเดลทางคณิตศาสตร์ในการจัดสรรระบบบันทึกเหตุการณ์ให้กับ VM ในการหาวิธีการจัดสรรที่เหมาะสมที่สุด เพื่อให้เกิดประโยชน์ทั้งผู้ให้บริการ และผู้ใช้งาน โดยผู้ให้บริการจะได้ประโยชน์ในเรื่องของการลดปริมาณการใช้งาน CPU และ RAM ที่จะเกิดในเครื่องคอมพิวเตอร์ของผู้ให้บริการ ให้สามารถทำงานได้โดยไม่เกิดปัญหาในเรื่องของการใช้งาน CPU และ RAM ที่หนักเกินไปจนทำให้ระบบคอมพิวเตอร์เกิดความล้มเหลวได้ และนอกจากนี้ยังรองรับในการเพิ่มจำนวน VM ในการให้บริการได้มากขึ้นด้วย ทำให้เกิดรายได้เพิ่มขึ้น และในส่วนของผู้ใช้นั้นนอกจากที่จะเลือกขนาดของเครื่องคอมพิวเตอร์แล้วนั้น ผู้ใช้บริการจะยังสามารถเลือกการดูแลความปลอดภัยของไฟล์หรือประสิทธิภาพในการทำงานในเครื่อง VM ของตนเองได้ และสามารถปรับเปลี่ยนความต้องการในการใช้งานของตนเองในภายหลังได้อีกด้วย

ปัญหาวิจัย

จากทั้งหมดที่กล่าวมาในหัวข้อความเป็นมาและความสำคัญของปัญหา และเนื้อหาเอกสาร และงานวิจัยที่เกี่ยวข้องในบทที่ 2 สามารถสรุปปัญหาในการวิจัย ได้ดังนี้

1. จากงานวิจัยของ Auxsorn (2020); Chan-In (2017); Jaiboon (2020); Wongthai (2017) ได้มีการทำทดสอบการทำงานของระบบบันทึกเหตุการณ์ในเรื่องของความแม่นยำในการตรวจจับเป็นหลัก และงานวิจัยของ Patrascu & Patriciu (2015), Yang, Wang, & Huang (2017); Rane, Wagh, & Dixit (2021); Ko & Will (2014) ได้มีการวัดประสิทธิภาพในการเข้าถึงข้อมูลด้วยระยะเวลาในการทำงานเพียงด้านเดียว โดยไม่ได้ทำการวัดประสิทธิภาพด้านทรัพยากร ได้แก่ ปริมาณการใช้งานหน่วยประมวลผล และปริมาณการใช้งานหน่วยความจำแรมในเครื่องของผู้ให้บริการ ดังนั้นในวิทยานิพนธ์นี้ ผู้วิจัยจึงได้ทำการวัดประสิทธิภาพในการใช้งานปริมาณการใช้งานหน่วยประมวลผล และปริมาณการใช้งานหน่วยความจำแรม เพื่อให้ผู้ให้บริการ IaaS คลาวด์สามารถสร้าง VM ไว้รองรับการใช้งานได้คุ้มค่ากับการลงทุนมากที่สุด ในขณะที่ยังมีการทำงานของระบบบันทึกเหตุการณ์

2. จากปัญหาวิจัยหลักในข้อที่ 1 และงานวิจัยของ Ye et al. (2011), Zhao et al. (2015), Malekloo et al. (2018) มีการจัดสรรในส่วนของ VM ให้เหมาะสมกับประสิทธิภาพการทำงาน และงานวิจัยของ Ala'a Al-Shaikh (2016) ได้สร้างโมเดลทางคณิตศาสตร์ในการหาค่าได้ประโยชน์ในการใช้งานให้มากที่สุด Wei (2020) ได้สร้างโมเดลทางคณิตศาสตร์ในการหาฟังก์ชันในการใช้ระยะเวลาที่น้อยที่สุด แต่ยังไม่มีการนำระบบบันทึกเหตุการณ์บน IaaS คลาวด์มาพิจารณาในการหาความคุ้มค่าและการใช้งานให้เกิดประโยชน์สูงสุด ในวิทยานิพนธ์นี้จึงได้นำแนวทางในการจัดสรร VM มาใช้ในการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บนระบบการประมวลผลแบบกลุ่มเมฆ ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (IaaS) ที่มีประสิทธิภาพ เพื่อให้เป็นไปตามข้อตกลงการให้บริการระหว่างผู้ให้บริการและผู้ให้บริการให้มากที่สุด และมีการใช้ทรัพยากรของเครื่องคอมพิวเตอร์ให้คุ้มค่ามากที่สุดในเรื่องของปริมาณการใช้งานหน่วยประมวลผลกลาง และปริมาณการใช้งานหน่วยความจำแรม โดยโมเดลทางคณิตศาสตร์ในการหาจำนวนล็อกเกอร์ที่ไม่ได้ถูกใช้งานมากที่สุด และนำไปทดสอบการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ด้วยขั้นตอนวิธีการค้นหาแบบสุ่ม (Random Search Algorithm)

3. จากปัญหาวิจัยหลักในข้อที่ 2 ผู้วิจัยได้ทำการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ เพื่อให้รองรับการทำงานที่มีการเปลี่ยนแปลงไปในส่วนของผู้ให้บริการ ซึ่งทำให้ผู้ให้บริการสามารถบริหารจัดการในการจัดสรรระบบบันทึกเหตุการณ์ได้อย่างมีประสิทธิภาพ

เป้าหมายของงานวิจัย

สร้างรูปแบบการจัดสรรระบบบันทึกเหตุการณ์บนระบบการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (IaaS) ที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการ

วัตถุประสงค์ของงานวิจัย

1. เพื่อวางรูปแบบการทดสอบความสามารถในการรองรับการทำงานของระบบบันทึกเหตุการณ์
2. เพื่อสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บนระบบการประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (IaaS) ที่มีประสิทธิภาพ
3. เพื่อสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

ขอบเขตของงานวิจัย

1. ขอบเขตด้านเทคโนโลยีและอุปกรณ์ซอฟต์แวร์

1.1 เครื่องคอมพิวเตอร์ CPU Intel Xeon@ CPU W3520 2.67 GHz 64 bits 8 Cores, SDRAM 24 GB และ HDD 1 TB 1 เครื่อง ติดตั้งระบบปฏิบัติการ Linux Fedora 16 ขนาด 64 bits สำหรับการทดลองสร้าง IaaS คลาวด์

1.2 ติดตั้งซอฟต์แวร์ Xen 4.1.-4-unstable บนระบบปฏิบัติการ Fedora 16 ขนาด 64 bit เพื่อจำลองสภาพแวดล้อมการทำงานของ IaaS คลาวด์ ให้เครื่องคอมพิวเตอร์หนึ่งเครื่องสามารถรันระบบปฏิบัติการได้มากกว่าหนึ่งระบบปฏิบัติการ

1.3 ติดตั้งระบบบันทึกเหตุการณ์ บนเครื่องคอมพิวเตอร์ Intel Xeon สำหรับผู้ให้บริการ

1.4 ใช้โปรแกรม Anaconda ในการเขียนโปรแกรมภาษา Python เพื่อทำการเขียนโปรแกรมการทำงานของโมเดลทางคณิตศาสตร์ในการจัดสรรระบบบันทึกเหตุการณ์

2. ขอบเขตด้านวิธีการ

2.1 การจำลองสร้าง IaaS Cloud บนเครื่องคอมพิวเตอร์ 1 เครื่องพร้อมติดตั้งระบบบันทึกเหตุการณ์

2.2 ทำการวัดปริมาณการใช้งานของหน่วยประมวลผล และปริมาณการใช้งานหน่วยความจำแรม

2.3 สร้างวิธีการจัดสรรระบบบันทึกเหตุการณ์ให้แก่แต่ละ VM

2.4 ร่างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ และทำการทดสอบการทำงานของโมเดล ด้วยการเขียนโปรแกรมด้วยภาษา Python ในการทดลองการทำงานของโมเดลทางคณิตศาสตร์

2.5 สร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ และทำการทดสอบการทำงานของโมเดล ด้วยการเขียนโปรแกรมด้วยภาษา Python ในการทดลองการทำงานของโมเดลทางคณิตศาสตร์

ประโยชน์ที่คาดว่าจะได้รับ

1. เป็นเครื่องมือในการบันทึกเหตุการณ์การทำงานของงานของการเข้าใช้งานไฟล์ในการตรวจสอบหาผู้เข้ามาโจมตีกรรมข้อมูลในเครื่องเสมือนของผู้ให้บริการ
2. เป็นเครื่องมือที่ช่วยในการบันทึกเหตุการณ์ โดยที่มีการใช้ทรัพยากรคอมพิวเตอร์ของผู้ให้บริการลดลง
3. เป็นเครื่องมือที่ช่วยในการเพิ่มประสิทธิภาพในการให้บริการของระบบ IaaS คลาวด์
4. ได้วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่ผู้ให้บริการสามารถจัดสรร VM ให้ทำงานภายใต้การดูแลของล็อกเกอร์ให้เป็นไปตามความต้องการของผู้ใช้บริการ
5. ได้วิธีการทำงานที่มีความยืดหยุ่น โดยผู้ให้บริการสามารถปรับลดหรือเพิ่มระดับความปลอดภัย หรือระดับประสิทธิภาพของตนเองได้

นิยามศัพท์เฉพาะ

ระบบบันทึกเหตุการณ์ (Logging System) หมายถึง ระบบที่สามารถจัดการและจัดเก็บข้อมูลการทำงานของผู้ใช้บริการเพื่อนำข้อมูลมาใช้ในการตรวจสอบความผิดปกติของการทำงาน

ประมวลผลแบบกลุ่มเมฆ ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (Infrastructure as a Service Cloud) หมายถึง ทรัพยากรของเครื่องเสมือน (Virtual Machine) เช่น หน่วยประมวลผล หน่วยความจำ หน่วยจัดเก็บข้อมูล และเครือข่าย ที่ทางผู้ให้บริการได้จัดเตรียมไว้สำหรับให้ผู้ให้บริการเข้าใช้งาน

การจัดสรรระบบบันทึกเหตุการณ์ (Allocation for Logging System) หมายถึง การบริหารจัดการระบบบันทึกเหตุการณ์ในการตรวจจับโปรเซสการทำงานในเครื่องคอมพิวเตอร์เสมือน

ประสิทธิภาพของเครื่องคอมพิวเตอร์ของผู้ให้บริการ (Performance of Host) หมายถึง ปริมาณการใช้งาน CPU และ RAM ในเครื่องคอมพิวเตอร์ของฝั่งผู้ให้บริการ

ประสิทธิภาพของเครื่องคอมพิวเตอร์ของผู้ใช้บริการ (Performance of VM) หมายถึง ความสามารถในการทำงานของเครื่องคอมพิวเตอร์ของผู้ใช้บริการ โดยไม่ถูกหยุดการทำงานของระบบบันทึกเหตุการณ์

ความแม่นยำ (Accuracy) หมายถึง ความสามารถในการตรวจจับโปรเซสการทำงานของระบบบันทึกเหตุการณ์



บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ในเนื้อหาบทที่ 2 ผู้วิจัยขอแนะนำเสนอเนื้อหาที่เกี่ยวกับเอกสารและงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์ เรื่อง โมเดลที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำและประสิทธิภาพการทำงานของคอมพิวเตอร์เสมือนบนการประมวลผลแบบกลุ่มเมฆ เนื่องจากการให้บริการบนคลาวด์นั้นต้องคำนึงถึงในเรื่องของความปลอดภัยในการที่ถูกผู้ไม่หวังดีมาทำการโจรกรรมข้อมูล จึงได้มีระบบบันทึกเหตุการณ์ (Logging System) หรือล็อกเกอร์ (Logger) ที่เป็นเครื่องมือนำมาใช้ในการตรวจสอบกิจกรรมต่าง ๆ ที่เกิดขึ้นบนการทำงานของคลาวด์เพื่อนำไปสู่การสนับสนุนด้านความปลอดภัยของข้อมูลในการตรวจสอบหาบุคคลผู้ซึ่งไม่ประสงค์ดีหรือเรียนกว่า แอ็กเกอร์ ที่เข้าถึงข้อมูลของผู้ใช้บริการโดยไม่ได้รับอนุญาต หลักฐานจากการทำงานของล็อกเกอร์นั้นจะช่วยให้สามารถยืนยันได้ว่าบุคคลที่เข้ามาทำงานในไฟล์ข้อมูลนั้นเป็นบุคคลที่ได้รับอนุญาตหรือไม่

สำหรับในงานวิทยานิพนธ์นี้จะเน้นไปที่เรื่องราวที่เกี่ยวข้องกับการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ซึ่งผู้วิจัยได้แบ่งเนื้อหาที่เกี่ยวข้องตามหัวข้อย่อยดังนี้

1. การประมวลผลแบบกลุ่มเมฆ
2. ปัญหาด้านความปลอดภัยบน IaaS คลาวด์
3. การทำงานของระบบบันทึกเหตุการณ์บน IaaS คลาวด์
4. ความปลอดภัยของไฟล์และประสิทธิภาพของคอมพิวเตอร์เสมือน
5. กระบวนการหาค่าที่เหมาะสม (Optimization)
6. ขั้นตอนวิธีการค้นหาแบบสุ่ม
7. การกำหนดรูปแบบการหาค่าที่เหมาะสม
8. เครื่องมือในการวัดประสิทธิภาพ
9. งานวิจัยที่เกี่ยวข้อง
10. กรอบแนวคิดการวิจัย

การประมวลผลแบบกลุ่มเมฆ

การประมวลผลแบบกลุ่มเมฆ หรือคลาวด์ (Cloud หรือ Cloud Computing) เปลี่ยนวิธีการสร้างและการให้บริการข้อมูล ก่อนยุคของคลาวด์ การให้บริการคอมพิวเตอร์แบบใหม่ เช่น เว็บไซต์ขนาดใหญ่ หรือเซิร์ฟเวอร์ไฟล์ จะหมายถึงค่าใช้จ่ายด้านต้นทุนจำนวนมากในพื้นที่สำหรับการเก็บข้อมูล การเชื่อมต่อเครือข่าย และเซิร์ฟเวอร์ คลาวด์คอมพิวเตอร์ตั้งเป็นเทคโนโลยีที่ใช้อินเทอร์เน็ตเพื่อ

จัดหาทรัพยากรคอมพิวเตอร์เป็นบริการ นวัตกรรมนี้ช่วยให้สามารถแบ่งปันทรัพยากรและค่าใช้จ่ายระหว่างลูกค้าระบบคลาวด์ได้ตามต้องการ นอกจากนี้ยังให้บริการคอมพิวเตอร์ออนไลน์ที่หลากหลายแก่ลูกค้าในราคาที่เหมาะสม เพื่อจัดการ การประมวลผล และจัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ ด้วยระบบคลาวด์ ลูกค้าไม่จำเป็นต้องติดตั้งซอฟต์แวร์ประเภทใดบนเครื่องของตน トラบเท่าที่มีการเชื่อมต่ออินเทอร์เน็ต ผู้ใช้บริการสามารถเข้าถึงข้อมูลทั่วโลกจากคอมพิวเตอร์เครื่องใดก็ได้ (AL-Roomi, 2013) และเมื่อองค์กรต้องการในการที่จะให้บริการคอมพิวเตอร์ในรูปแบบใหม่นั้น จำเป็นต้องมีบุคลากรทางด้านไอทีในการที่พัฒนาและดูแลระบบดังกล่าวซึ่งทำให้ต้องใช้เวลาในการพัฒนาและเสียค่าใช้จ่ายเป็นจำนวนมาก เมื่อเปรียบเทียบกับการใช้ระบบการประมวลผลคลาวด์ที่สร้างขึ้นและประมวลผลตามความต้องการในระยะเวลาที่รวดเร็วกว่า (Scheer et al., 2016) ปัจจุบันคลาวด์ได้ถูกนำมาใช้ในองค์กรมากขึ้นเรื่อย ๆ เพราะสามารถปรับขนาดและยืดหยุ่นได้ตามความต้องการของผู้ใช้งาน

สถาบันแห่งชาติเพื่อมาตรฐานและเทคโนโลยี (The National Institute of Standards and Technology : NIST) ได้ให้ความหมายของคลาวด์ไว้ว่า (Mell & Grance, 2011) คลาวด์ คือรูปแบบการใช้งานของทรัพยากรคอมพิวเตอร์ (เช่น เครือข่าย เครื่องเซิร์ฟเวอร์ แหล่งเก็บข้อมูล แอปพลิเคชัน และการให้บริการ) ที่สามารถเข้าถึงการใช้งานผ่านทางระบบเครือข่ายได้อย่างสะดวก รวดเร็ว โดยที่ทรัพยากรดังกล่าวยังต้องสอดคล้องกับความต้องการที่เปลี่ยนแปลงไปของผู้ใช้งาน ทั้งนี้ผู้ใช้งานสามารถใช้และยกเลิกการใช้ทรัพยากรดังกล่าวได้อย่างรวดเร็วและเสียเวลาน้อยที่สุด

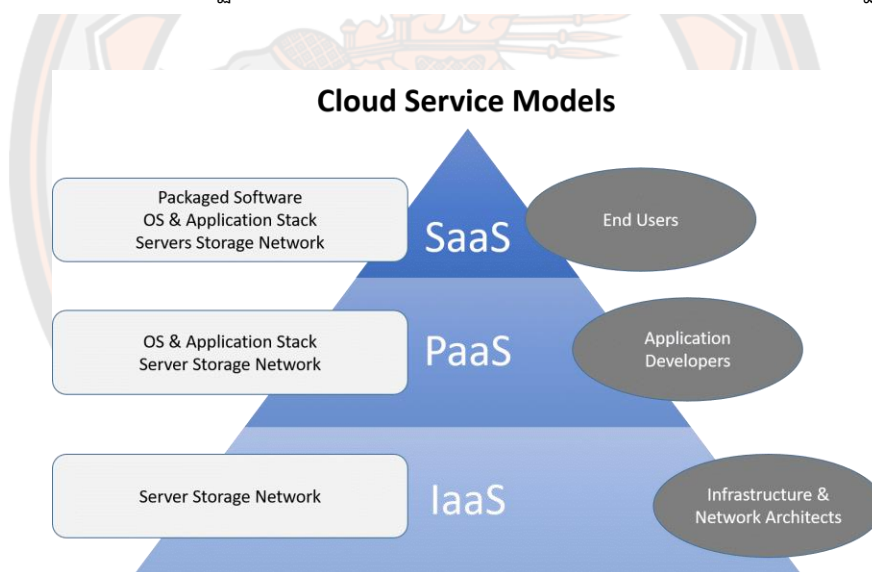
และในเอกสารการเผยแพร่ของ NIST (Mell & Grance, 2011) ยังได้ระบุรูปแบบการให้บริการของคลาวด์ โดยแบ่งเป็น 3 ประเภทได้แก่ 1) ประเภทการให้บริการซอฟต์แวร์ (Software as a Service) หรือเรียกว่า SaaS คลาวด์ เป็นการให้บริการในส่วนซอฟต์แวร์โดยที่ไม่ต้องติดตั้งซอฟต์แวร์ลงบนเครื่องคอมพิวเตอร์ แต่ซอฟต์แวร์จะถูกจัดเก็บอยู่บนคลาวด์ 2) ประเภทการให้บริการแพลตฟอร์ม (Platform as a Service) หรือเรียกว่า PaaS คลาวด์ เป็นการให้บริการแก่นักพัฒนาที่สามารถสร้างซอฟต์แวร์ได้เอง 3) ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (Infrastructure as a Service) หรือเรียกว่า IaaS คลาวด์ เป็นการให้บริการพื้นฐานทางโครงสร้างคอมพิวเตอร์ เช่น การประมวลผล การจัดเก็บข้อมูล เครือข่าย เป็นต้น และนอกจากนี้ยังแบ่งประเภทของคลาวด์ตามขอบเขตการจัดการเป็น 4 ประเภท ได้แก่ 1) คลาวด์แบบองค์กร (Private Cloud) เป็นโครงสร้างของคลาวด์ที่ถูกสร้างขึ้นเพื่อให้ใช้ภายในองค์กรใดองค์กรหนึ่งเท่านั้นและให้บริการเฉพาะผู้ใช้งานภายในองค์กรนี้เท่านั้น 2) คลาวด์สำหรับผู้ใช้งานเฉพาะกลุ่ม (Community Cloud) เป็นโครงสร้างของคลาวด์ที่สร้างขึ้นสำหรับผู้ใช้งานเฉพาะกลุ่มโดยอาจจะสร้างมาจากความร่วมมือกันหลาย ๆ องค์กรที่มีความต้องการร่วมกัน 3) คลาวด์แบบสาธารณะ (Public Cloud) เป็นโครงสร้างของคลาวด์ที่สร้างขึ้นโดยองค์กรที่มีจุดประสงค์ในการให้บริการแก่ผู้ใช้งานภายนอกองค์กร และ 4)

คลาวด์แบบผสมผสาน (Hybrid Cloud) เป็นโครงสร้างของคลาวด์ที่สร้างขึ้นด้วยการนำเอาประเภทการจัดการในรูปแบบคลาวด์แบบองค์กร คลาวด์สำหรับผู้ใช้เฉพาะกลุ่ม และคลาวด์แบบสาธารณะ มารวมกันตั้งแต่ 2 รูปแบบขึ้นไป

จากการเผยแพร่ของ NIST (Mell & Grance, 2011) ในช่วงต้น ผู้วิจัยจะได้อธิบายการแบ่งประเภทของคลาวด์ ดังนี้

1. การแบ่งประเภทของคลาวด์ตามลักษณะการให้บริการ

การแบ่งประเภทของคลาวด์ตามลักษณะการให้บริการ โดยสามารถแบ่งได้เป็น 3 ประเภท ได้แก่ 1) ประเภทการให้บริการซอฟต์แวร์ (Software as a Service) หรือเรียกว่า SaaS 2) ประเภทการให้บริการแพลตฟอร์ม (Platform as a Service) หรือเรียกว่า PaaS คลาวด์ และ 3) ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (Infrastructure as a Service) หรือเรียกว่า IaaS คลาวด์ ประเภทของคลาวด์ทั้ง 3 ประเภทนั้นจะมีความสัมพันธ์กัน เนื่องมาจาก IaaS คลาวด์นั้นเป็นโครงสร้างพื้นฐานในการสร้าง PaaS คลาวด์ และ SaaS คลาวด์ ดังปรากฏในภาพ 2



ภาพ 2 แสดงรูปแบบการให้บริการคลาวด์

ที่มา: Arron, 2022

1. ประเภทการให้บริการซอฟต์แวร์ (Software as a Service: SaaS) เป็นรูปแบบการให้บริการซอฟต์แวร์โดยผู้ให้บริการในการใช้งานผ่านระบบเครือข่ายอินเทอร์เน็ต (Mohammed & Zeebaree, 2021) โดยที่ซอฟต์แวร์หรือแอปพลิเคชันถูกติดตั้งบนเว็บคลาวด์ แทนที่จะทำการติดตั้งและจัดเก็บในเครื่องของผู้ใช้บริการ (Shakor & Surameery, 2021)

2. ประเภทการให้บริการแพลตฟอร์ม (Platform as a Service: PaaS) เป็นรูปแบบการให้บริการในกลุ่มของเครื่องมือในการพัฒนาแอปพลิเคชันได้อย่างมีประสิทธิภาพ (Shakor & Surameery, 2021) โดยไม่ต้องทำการดาวน์โหลดหรือจัดการกับอินเทอร์เน็ตเฟซของผู้ใช้ (Mohammed & Zeebaree, 2021) ผู้ใช้บริการสามารถเขียนโปรแกรมด้วยภาษาต่าง ๆ แอปพลิเคชัน และเครื่องมือที่ผู้ให้บริการระบบคลาวด์ในการเผยแพร่แอปพลิเคชันที่ผู้ให้บริการได้ทำการสร้างหรือซื้อไปยังโครงสร้างพื้นฐานของคลาวด์ (Sunyaev, 2020)

3. ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (Infrastructure as a Service: IaaS) เป็นรูปแบบการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ในลักษณะของเครื่องคอมพิวเตอร์เสมือน (Virtual Computer) เซิร์ฟเวอร์ อุปกรณ์จัดเก็บข้อมูล เป็นต้น (Sultan, 2010) เพื่อใช้ในการประมวลผลซึ่งสามารถกำหนดค่าและจัดการผ่านทางเครือข่ายอินเทอร์เน็ต (Shakor & Surameery, 2021) ซึ่งเป็นการให้บริการในการเช่าเครื่องคอมพิวเตอร์เสมือนจากผู้ให้บริการคลาวด์ (Auxsorn et al., 2020)

ผู้วิจัยจึงขออธิบายรายละเอียดในส่วนของโครงสร้างพื้นฐานของคลาวด์ และสภาพแวดล้อมของคลาวด์ตามโครงสร้างของ IaaS คลาวด์ ในหัวข้อถัดไป เนื่องจากในงานวิทยานิพนธ์นี้จะมุ่งเน้นไปที่โครงสร้างแบบ IaaS คลาวด์ เป็นหลักที่นำมาในการทดลอง

2. การแบ่งประเภทของคลาวด์รูปแบบตามขอบเขตการจัดการ

การแบ่งประเภทของคลาวด์รูปแบบตามขอบเขตการจัดการ ประกอบไปด้วย 4 ประเภทหลัก ๆ ได้แก่ Public Cloud, Private Cloud, Community Cloud และ Hybrid Cloud (Kumari, 2021) โดยอธิบายรายละเอียดของคลาวด์แต่ละประเภท ดังนี้

1. คลาวด์แบบสาธารณะ (Public Cloud) เป็นรูปแบบการให้บริการที่สร้างโดยองค์กร เพื่อให้การบริการกับผู้ใช้งานภายนอกองค์กร ในการเป็นเจ้าของ ถูกจัดการ และถูกดำเนินงาน การให้บริการประเภทนี้จะสร้างปัญหาในด้านของความปลอดภัยอย่างมาก (Alam, 2020) เนื่องจากการถูกควบคุมและจัดการโดยบุคคลภายนอก

2. คลาวด์ภายในองค์กร (Private Cloud) เป็นรูปแบบการให้บริการที่สร้างขึ้นเพื่อให้ใช้ภายในองค์กรใดองค์กรหนึ่งเท่านั้น ซึ่งเป็นการออกแบบมาสำหรับองค์กรเดียว เช่น ระบบคลาวด์เฉพาะสำหรับบริษัท (Rashid & Chaturvedi, 2019) โดยการให้บริการในรูปแบบนี้จะช่วยในเรื่องความปลอดภัยของข้อมูล ความยืดหยุ่น ความสามารถในการปรับขนาด และความน่าเชื่อถือ (Bokhari et al., 2018)

3. คลาวด์สำหรับผู้ใช้เฉพาะกลุ่ม (Community Cloud) เป็นรูปแบบการให้บริการที่สร้างขึ้นมาจากความร่วมมือของหลายองค์กรเพื่อให้ใช้งานเฉพาะกลุ่มเท่านั้น และโครงสร้างพื้นฐานระบบคลาวด์ทั้งหมดจะถูกแชร์ผ่านหลายองค์กรที่อยู่ในชุมชนเดียวกัน (Singh & Baheti, 2017)

4. คลาวด์แบบผสมผสาน (Hybrid Cloud) เป็นรูปแบบการให้บริการที่ประกอบไปด้วย Public Cloud, Private Cloud และ Community Cloud ตั้งแต่สองประเภทขึ้นไปที่มีการผสมผสานกัน โดยเทคโนโลยีที่สร้างขึ้นเฉพาะต้องเป็นมาตรฐานเดียวกันสำหรับคลาวด์ทั้งหมดที่นำมาผสมผสานกัน (Mell & Grance, 2011)

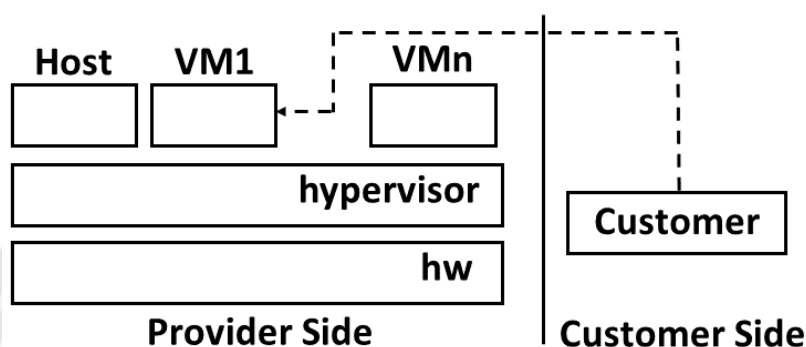
การประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์แบบสาธารณะ

การประมวลผลแบบกลุ่มเมฆประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (Infrastructure as a Service : IaaS) เช่น Amazon Elastic Compute Cloud (Chaisiri, 2011) หรือการให้บริการคลาวด์ประเภทบริการแหล่งเก็บข้อมูล (Cloud Storage Service) เช่น Google Drive, Microsoft OneDrive เป็นต้น มุ่งเน้นไปที่การนำเสนอเครื่องเสมือน (Virtual Machine: VM) ให้แก่ลูกค้าที่ใช้บริการ IaaS ลูกค้าสามารถเช่าเครื่องเสมือนและสามารถเข้าถึงโดยผ่านทางอินเทอร์เน็ต ส่วนหลัก ๆ ของ VM จะประกอบไปด้วยการจัดเก็บข้อมูล และเครือข่าย ลูกค้าสามารถติดตั้งระบบปฏิบัติการที่ต้องการใช้ใน VM ได้ ประเภทของคลาวด์แบบ IaaS นั้นมีการใช้งานอย่างแพร่หลายในหลากหลายกิจกรรม เช่น งานในภาครัฐ งานด้านการศึกษา หรือการทดลองด้านการแพทย์ เป็นต้น การให้บริการต่าง ๆ โดย IaaS คลาวด์ ทำงานในรูปแบบของ virtualization ซึ่งอนุญาตให้แบ่งปันทรัพยากรคลาวด์ระหว่างผู้ใช้บริการได้ (Borisaniya & Patel, 2019)

สถาบันแห่งชาติเพื่อมาตรฐานและเทคโนโลยี (Mell & Grance, 2011) ได้ให้ความหมายของ IaaS คลาวด์ไว้ว่า IaaS คลาวด์ คือ ความสามารถของคลาวด์ในการจัดเตรียมทรัพยากร การประมวลผล การจัดเก็บข้อมูล ระบบเครือข่าย และอย่างอื่นที่เป็นพื้นฐานในการให้ผู้ใช้บริการสามารถติดตั้งและใช้งานซอฟต์แวร์ตามที่ต้องการได้ และ Inokuchi & Kourai (2020) ได้กล่าวไว้ว่า การให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ของคลาวด์ เป็นการจัดเตรียมในส่วนเครื่องคอมพิวเตอร์เสมือน (Virtual Machine, VMs) ไว้สำหรับผู้ใช้บริการ โดยผู้ใช้งานสามารถติดตั้งในส่วนระบบปฏิบัติการได้ตามการใช้งานของตนเอง ซึ่งสามารถสรุปได้ว่า IaaS คลาวด์ เป็นการ

ให้บริการโครงสร้างทางคอมพิวเตอร์ในรูปแบบของคอมพิวเตอร์เสมือนที่ผู้ให้บริการสามารถเลือกคุณลักษณะของเครื่องคอมพิวเตอร์เสมือนและซอฟต์แวร์ตามที่ต้องการใช้งานได้

สถาปัตยกรรมของ IaaS คลาวด์ของ Wongthai (2014) ผู้วิจัยจึงได้ขอปรับปรุงสถาปัตยกรรมของ IaaS คลาวด์ให้มีชื่อเรียกที่ผู้อ่านสามารถเข้าใจได้ง่าย โดย dom0 ในงานวิทยานิพนธ์นี้จะขอเรียกว่า โฮสต์ (Host) และ domU ในงานวิทยานิพนธ์นี้จะขอเรียกว่า VM ดังปรากฏในภาพ 3



ภาพ 3 แสดงสถาปัตยกรรม IaaS คลาวด์

ที่มา: ปรับปรุงจาก Wongthai, 2014

จากภาพ 3 สามารถแบ่งโครงสร้างของสถาปัตยกรรมได้เป็น 2 ส่วน ได้แก่ ฝั่งผู้ให้บริการ (Provider Side) และฝั่งผู้ให้บริการ (Customer Side)

ฝั่งผู้ให้บริการ (Provider Side) หมายถึง องค์กรที่บริการให้เช่าคอมพิวเตอร์เสมือนแก่ผู้ให้บริการ เช่น บริษัท Amazon, Microsoft และ Google ซึ่งภายในโครงสร้างจะประกอบไปด้วยส่วนสำคัญคือ

1. hw (ย่อมาจาก Hardware) คือ เครื่องคอมพิวเตอร์ที่ใช้สำหรับสร้าง IaaS คลาวด์
2. hypervisor คือ ซอฟต์แวร์ที่สามารถทำให้เครื่องคอมพิวเตอร์หนึ่งเครื่องสามารถรันระบบปฏิบัติการได้มากกว่าหนึ่งระบบในเครื่องเดียวกัน ซึ่งซอฟต์แวร์ที่เป็นแบบฟรีที่ใช้ในการติดตั้ง เช่น Xen
3. Host คือ ระบบบริหารจัดการ VM และ hw และทำงานเมื่อ hypervisor ถูกเปิดใช้งาน
4. VM คือ คอมพิวเตอร์เสมือน ที่ผู้ให้บริการเป็นเจ้าของ ซึ่งสามารถให้บริการได้มากกว่า 1 VM เช่น VM1, VM2, ..., VMn

ปัญหาด้านความปลอดภัยบน IaaS คลาวด์

จากการนำเสนอรายละเอียดในหัวข้อความเป็นมาและความสำคัญของปัญหาแล้วนั้น ผู้วิจัยของขยายความของรายงานภัยคุกคามของการประมวลผลคลาวด์ในรายงาน Top Threats to Cloud Computing Pandemic Eleven (CSA, 2022) โดยมีรายละเอียดการคุกคามดังนี้ คือ

1. การระบุตัวตน ข้อมูลประจำตัว การเข้าถึงการจัดการคีย์ (Insufficient Identity, Credentials, Access, and Key Management)

การระบุตัวตนในการเข้าถึง คือขอบเขตใหม่ซึ่งเป็นวิธีหลักในการจำกัดการเข้าถึงเนื่องจากบนระบบคลาวด์มีบัญชีการใช้งานที่มีจำนวนมาก และกว้างขวางมากขึ้น และผู้ใช้สามารถเข้าถึงสิ่งเหล่านี้ได้จากสถานที่ต่าง ๆ การจัดการข้อมูลประจำตัวจึงเป็นอันดับแรกๆ ที่จัดให้เป็นปัญหาภัยคุกคามบนระบบคลาวด์ จากการรายงานการวิจัยของ Gartner (2020) เรื่อง “Managing Privileged Access in Cloud Infrastructure” คาดการณ์ว่าภายในปี 2566 ความล้มเหลวในการรักษาความปลอดภัย มีสาเหตุมาจากการจัดการข้อมูลส่วนตัว การเข้าถึง และสิทธิ์ที่ไม่เพียงพอเพิ่มขึ้นจาก 50% ในปี 2563 เป็น 75% ในปี 2566 ปัจจุบันมีการระบุข้อมูลตัวตนของเครื่องมากกว่าการระบุข้อมูลตัวตนของผู้ใช้งานมากถึงห้าเท่าเครื่องไม่ได้เป็นเพียงทรัพย์สินแบบคงที่ เช่น เซิร์ฟเวอร์ ฐานข้อมูล และระบบจัดเก็บข้อมูล แต่เป็นทรัพย์สินชั่วคราวจึงส่งผลให้มีปัญหาการระบุตัวตนดังกล่าวในปัญหาด้านความปลอดภัยนี้จะมีผลกระทบโดยตรงกับผู้ใช้บริการ

2. อินเทอร์เฟซและส่วนต่อประสานโปรแกรมประยุกต์ที่ไม่ปลอดภัย (Insecure Interfaces and APIs) API แพร่หลายในพื้นที่คลาวด์เพราะทุกอย่างเป็นโค้ด (เสมือน) และทุกแอปพลิเคชันจำเป็นต้องสื่อสารกับแอปพลิเคชันอื่นเพื่อให้ทำงานได้ เช่นเดียวกับแอปพลิเคชันดั้งเดิม API มีแนวโน้มที่จะกำหนดค่าผิดพลาด การเขียนโค้ดที่ไม่ดี ขาดการตรวจสอบสิทธิ์และการให้สิทธิ์ที่ไม่เหมาะสม นักพัฒนาควรใช้เครื่องมือปรับขนาดและการทำงานอัตโนมัติสำหรับการจัดการ API และบำรุงรักษาแคตตาล็อก API เพื่อรวมรายละเอียดต่าง ๆ เช่น การเผชิญหน้าภายในหรือภายนอก สิ่งที่เราใช้สำหรับ ข้อมูลที่ถูกเปิดเผย และวิธีการใช้งาน

3. การกำหนดค่าผิดพลาดและการควบคุมการเปลี่ยนแปลงที่ไม่ดีพอ (Misconfiguration and Inadequate Change Control) ผู้ให้บริการระบบคลาวด์ช่วยให้เปิดใช้ทรัพยากรในระบบคลาวด์ได้ง่าย เช่น เซิร์ฟเวอร์ ฐานข้อมูล แอปพลิเคชันคอนเทนเนอร์ ฯลฯ ข้อเสียของความง่ายในการเตรียมใช้งานนี้คือการกำหนดค่าผิดพลาดได้ง่าย การเปิดใช้งานแอซเซทได้ภายในไม่กี่นาที ทำซ้ำและกระจายไปทั่วโครงสร้างพื้นฐานคลาวด์ของผู้ให้บริการ และข้อผิดพลาดในการกำหนดค่าที่ผิดพลาดพื้นฐานจะเกิดขึ้นซ้ำ ๆ เมื่อเวลาผ่านไป เซิร์ฟเวอร์หรือฐานข้อมูลจะได้รับการกำหนดค่าใหม่ในทันทีเพื่อวัตถุประสงค์เฉพาะ และการติดตามการเปลี่ยนแปลงเหล่านี้ในโครงสร้างพื้นฐานทั้งหมดเป็นเรื่องยาก สิ่งนี้เรียกอีกอย่างว่าการเลื่อนการกำหนดค่า และเป็นปัญหาจริงในระบบคลาวด์

4. การขาดสถาปัตยกรรมและกลยุทธ์การรักษาความปลอดภัยบนคลาวด์ (Lack of Cloud Security Architecture and Strategy) ทุกองค์กรจำเป็นต้องมีกลยุทธ์การรักษาความปลอดภัยที่ไม่เหมือนกันเพื่อตอบสนองความต้องการการทำงานของแต่ละองค์กร แต่มีในบางกรณีที่ทุกองค์กรต้องพิจารณาหลักการพื้นฐานบางประการ เช่น การทำความเข้าใจในรูปแบบความรับผิดชอบร่วมกันและความรับผิดชอบของผู้ให้บริการ การปกป้องข้อมูลระหว่างการเคลื่อนย้ายข้อมูลด้วยการเข้ารหัส เป็นต้น

5. การพัฒนาซอฟต์แวร์ที่ไม่ปลอดภัย (Insecure Software Development) แอปพลิเคชันที่พัฒนาบนระบบคลาวด์ที่ทันสมัยและซับซ้อนที่สุด ต้องการให้นักพัฒนาเข้าถึงระบบคลาวด์จากแล็ปท็อปแบบเดิม นักพัฒนาอาจใช้แล็ปท็อปเพื่อลงชื่อเข้าใช้ทรัพยากรหลายรายการผ่านผู้ให้บริการข้อมูลประจำตัว ดาวน์โฮสต์โค้ดจากที่เก็บ Git สาธารณะ ใช้ส่วนขยาย Chrome สำหรับงานพัฒนาเข้าถึงเครื่องมือนักพัฒนาซอฟต์แวร์ SaaS บนคลาวด์ ใช้คอนเทนเนอร์และ Kubernetes เพื่อจัดแพ็คเกจและส่งมอบแอปพลิเคชันของตน และจัดการทุกด้านของกระบวนการสร้าง ทดสอบ ปรับใช้ CI/CD ปัญหาด้านซัพพลายเชนเป็นสิ่งที่ทีมรักษาความปลอดภัยส่วนใหญ่คำนึงถึง และจะเห็นว่ามีช่องโหว่มากมายที่อาจเล็ดลอดเข้าสู่กระบวนการพัฒนาได้

6. ทรัพยากรของบุคคลที่สามไม่ปลอดภัย (Unsecured Third-Party Resources) ผลิตภัณฑ์หรือบริการคือผลรวมของผลิตภัณฑ์และบริการอื่น ๆ ทั้งหมดที่ใช้ เช่น API ผลิตภัณฑ์ SaaS และที่เก็บรหัสโอเพ่นซอร์ส ผลิตภัณฑ์ดิจิทัลเกือบทั้งหมดมีองค์ประกอบบางอย่างจากซัพพลายเออร์บุคคลที่สาม ผู้โจมตีเพียงต้องการหาจุดอ่อนในห่วงโซ่อุปทานเพื่อใช้ประโยชน์จากผลิตภัณฑ์ทั้งหมดที่เกิดจากมัน สิ่งนี้เรียกว่าช่องโหว่ของห่วงโซ่อุปทานห่วงโซ่อุปทานของซอฟต์แวร์ประกอบด้วยส่วนประกอบ ไลบรารี เครื่องมือ และกระบวนการที่ใช้ในการพัฒนา สร้าง และเผยแพร่ซอฟต์แวร์ ทุกวันนี้ แอปพลิเคชันส่วนใหญ่ไม่ได้สร้างขึ้นจากศูนย์ แต่สร้างขึ้นจากโค้ดที่มีอยู่แล้วจากไลบรารี และนำมาปะติดปะต่อกับโค้ดดั้งเดิมบางส่วนเพื่อสร้างผลิตภัณฑ์ขั้นสุดท้ายได้เร็วกว่าการคิดค้นวงล้อใหม่ตั้งแต่เริ่มต้น อาจมีช่องโหว่แฝงตัวอยู่ในส่วนประกอบของโค้ดที่รวบรวมจากหลายแหล่ง

7. ช่องโหว่ของระบบ (System Vulnerabilities) ช่องโหว่ใหม่ถูกค้นพบอย่างต่อเนื่อง ดังนั้นควรสแกนระบบอย่างต่อเนื่องเช่นกัน แพลตฟอร์มการวิเคราะห์ความปลอดภัยบนคลาวด์ เช่น Uptycs สามารถให้ความสามารถในการค้นหาว่าช่องโหว่ที่เพิ่งประกาศนั้นแฝงตัวอยู่ในระบบคลาวด์ของผู้ให้บริการ

8. การเปิดเผยข้อมูลบนคลาวด์โดยไม่ได้ตั้งใจ (Accidental Cloud Data Disclosure) การเปิดเผยข้อมูลโดยไม่ได้ตั้งใจสามารถเกิดขึ้นได้ในระบบคลาวด์ ตัวอย่างบางส่วนของ การเปิดเผยข้อมูลโดยไม่ได้ตั้งใจ ได้แก่ พนักงานแชร์ลิงก์ไปยังเอกสารภายในในอีเมล ซึ่งอาจจะมีการแชร์ข้อมูลกับบุคคลภายนอกองค์กร และผู้ให้บริการควรให้ทางเลือกที่ปลอดภัยกว่าการปฏิบัติที่เป็นอันตราย

เหล่านี้ด้วย อย่างไรก็ตามผู้ใช้บริการควรมีเครื่องมือป้องกันการสูญหายของข้อมูลและใช้ประโยชน์จากข้อเสนอจากผู้ให้บริการคลาวด์ เช่น CloudDLP ของ Google

9. การกำหนดค่าผิดพลาดและการใช้ประโยชน์จากปริมาณงานแบบไร้เซิร์ฟเวอร์และคอนเทนเนอร์ (Misconfiguration and Exploitation of Serverless and Container Workloads)

ในระบบคลาวด์ การสร้างแอปพลิเคชันด้วยคอนเทนเนอร์และ Kubernetes เป็นเรื่องใหม่ แม้ว่าคอนเทนเนอร์อาจอยู่บนเซิร์ฟเวอร์ใดก็ตาม แต่ก็ถือว่าเป็นอุปกรณ์ที่ใช้ระบบคลาวด์ พวกเขาเป็นเครื่องเสมือนที่มีน้ำหนักเบาโดยไม่มีระบบปฏิบัติการ เฉพาะซอฟต์แวร์ที่จำเป็นสำหรับการเรียกใช้แอปพลิเคชันคอนเทนเนอร์ เช่นเดียวกับเครื่องเสมือน แต่แยกจากฮาร์ดแวร์เซิร์ฟเวอร์แบบดั้งเดิม สามารถเรียกใช้งาน และปิดเครื่องในช่วงเวลาสั้น ๆ เพื่อดำเนินการสั้น ๆ และโดยปกติแล้วจะถูกจัดเตรียมให้ทำงานสอดคล้องกันราวกับว่าเป็นแอปพลิเคชันขนาดใหญ่ ด้วยวิธีนี้แอปพลิเคชันขนาดใหญ่ไม่จำเป็นต้องเปิดและทำงานอยู่ตลอดเวลา แต่ละคอนเทนเนอร์ดำเนินการงานที่เรียกตามที่ต้องการ และจากนั้นปิดอีกครั้ง รันใหม่คอนเทนเนอร์ยอดนิยม ได้แก่ CRI-O, Docker และ LXC

10. การก่ออาชญากรรม / แฮกเกอร์ / ภัยคุกคามต่อเนื่องขั้นสูง (Organized Crime/Hackers/APT)

แฮกเกอร์ โดยเฉพาะอย่างยิ่งกลุ่มภัยคุกคามขั้นสูงที่ต่อเนื่อง เป็นเหตุผลที่เราต้องเสริมการป้องกันการไล่จับระหว่างผู้เชี่ยวชาญด้านความปลอดภัยทางไซเบอร์กับแฮกเกอร์ จะเกิดขึ้นเมื่อแฮกเกอร์นั้นมีการดำเนินกิจกรรมต่าง ๆ เพื่อประโยชน์ของแฮกเกอร์เอง ถึงแม้ว่าผู้เชี่ยวชาญด้านความปลอดภัยจะไม่สามารถควบคุมได้ว่าแฮกเกอร์นั้นกำลังทำอะไรอยู่ แต่ก็สามารถช่วยในการตรวจจับกิจกรรมของแฮกเกอร์ที่เกิดขึ้นในระบบคลาวด์ของผู้ให้บริการได้ด้วยความสามารถในการตรวจจับภัยคุกคามตามพฤติกรรม

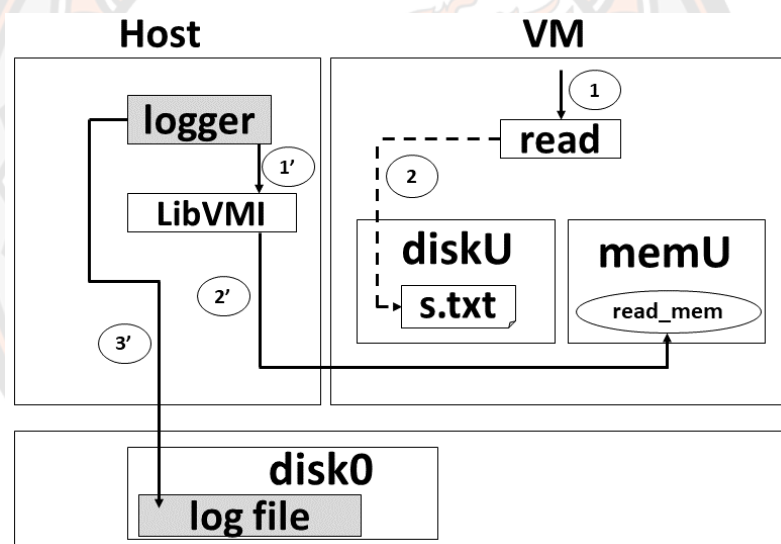
11. การกรองข้อมูลที่เก็บบนคลาวด์ (Cloud Storage Data Exfiltration) การโจมตีแบบเจาะข้อมูลเป็นหนึ่งในสิ่งที่น่ากลัวที่สุด (โดยเจ้าของข้อมูล) และก่อให้เกิดผลกำไรมากที่สุด (โดยผู้โจมตี) เครื่องมือ CDR ที่ดีสามารถช่วยผู้ใช้บริการตรวจจับการโจมตีเหล่านี้ได้แบบเรียลไทม์โดยใช้การผสมผสานระหว่าง IOC และการตรวจจับภัยคุกคามตามพฤติกรรม

ผู้วิจัยจึงได้มองเห็นถึงการแก้ปัญหาภัยคุกคามในประเด็นการเข้าถึงข้อมูล ในรายงานของ CSA ปี 2019 และ และการก่ออาชญากรรม / แฮกเกอร์ / ภัยคุกคามต่อเนื่องขั้นสูง ในรายงานของ CSA ปี 2022 ที่เน้นไปที่การจัดการการเข้าถึงโดยแฮกเกอร์ และการโจมตีแบบเจาะข้อมูลเพื่อจะป้องกันการเกิดภัยคุกคามแก่ผู้ใช้บริการได้อย่างมีประสิทธิภาพ ซึ่งมีวิธีการในการที่จะรักษาความปลอดภัยในการแฮ็กเกอร์ และโจมตีแบบเจาะข้อมูล

ดังนั้นในงานวิจัยนี้จึงใช้ระบบบันทึกเหตุการณ์ที่สามารถตรวจสอบร่องรอยของการกระทำต่าง ๆ มาเป็นข้อมูลในการสร้างโมเดลทางคณิตศาสตร์สำหรับการจัดสรรระบบบันทึกเหตุการณ์

การทำงานของระบบบันทึกเหตุการณ์บน IaaS คลาวด์

ระบบบันทึกเหตุการณ์ เป็นวิธีการหนึ่งที่สำคัญสำหรับช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามขึ้นต่อคลาวด์ โดยภายในระบบบันทึกเหตุการณ์ได้จัดเก็บหลักฐานล็อกไฟล์ (Log File) ไว้เป็นหลักฐานและหลักฐานจาก Log file สามารถนำไปช่วยหาบุคคลที่กระทำผิดมารับผิดชอบได้ โดยการทำงานของระบบ Logging System ใน IaaS คลาวด์ แบ่งการทำงานออกเป็น 2 ส่วน ได้แก่ 1) ส่วนที่เป็นผู้ให้บริการ (Host) ในส่วนนี้จะได้ทำการติดตั้ง Logger ซึ่งเป็นโปรเซสสำหรับใช้ในการตรวจจับบันทึกเหตุการณ์ที่เกิดขึ้นใน VM 2) ส่วนของผู้ใช้บริการ (VM) ที่เป็นเครื่องที่ถูกตรวจสอบ ดังปรากฏในภาพ 4



ภาพ 4 แสดงสภาพแวดล้อมการทำงานระหว่าง Host และ VM

ที่มา: Wongthai & Moorsel (2016), แก้วไขภาพจาก Wongthai

1. ส่วนผู้ให้บริการ (VM)

จากภาพ 4 ส่วนของ VM คือ กรอบสี่เหลี่ยมทางด้านขวาที่มีคำว่า VM อยู่ด้านบนกรอบสี่เหลี่ยม ในส่วนนี้ผู้ให้บริการเป็นเจ้าของ diskU (พื้นที่จัดเก็บเสมือน) ภายใน diskU ได้เก็บไฟล์ s.txt โดยสมมติให้ไฟล์นี้เป็นไฟล์ที่สำคัญของผู้ให้บริการ และผู้วิจัยได้สมมติฐานในกระบวนการทำงานภายใน VM และกิจกรรมที่เกิดขึ้นของแอปพลิเคชัน read ได้ดังนี้

กระบวนการทำงานภายใน VM ประกอบไปด้วย 3 ขั้นตอน ดังนี้

1. VM อาจถูกครอบครองโดยผู้ไม่หวังดี (จากภาพ 4 ในกล่อง VM แยกเกอร์เป็นผู้ครอบครอง)
2. แยกเกอร์ครอบครอง VM และทำการรันแอปพลิเคชัน read เพื่ออ่านไฟล์ s.txt ภายใน diskU ตามเส้นปะ หมายเลข 2
3. เมื่อแอปพลิเคชัน read ถูกรันจากแอปพลิเคชัน read จึงเปลี่ยนเป็นโปรเซส read และกระบวนการทำงานเหล่านี้เรียกว่า “กิจกรรมของแอปพลิเคชัน read” (โปรเซส คือ โปรแกรมที่ถูกรัน (Saha, 2006) หรือโปรแกรมที่กำลังทำงาน (Bryant et al., 2010))

กิจกรรมของแอปพลิเคชัน read สำหรับการทดลองมี 4 ขั้นตอน ดังนี้

1. แยกเกอร์รันแอปพลิเคชัน read เพื่อเปิดไฟล์ s.txt ภายใน VM ตามเส้นปะ หมายเลข 2
2. เมื่อแยกเกอร์เปิดไฟล์ s.txt ได้แล้ว แยกเกอร์ได้พิมพ์ข้อมูลภายใน s.txt และอ่านไฟล์
3. ปิดไฟล์
4. จบการทำงาน

ข้อมูลที่เกิดขึ้นจากกิจกรรมของแอปพลิเคชัน read จะถูกเก็บไว้ใน read_mem ที่อยู่ใน memU (หน่วยความจำหลักของ VM) ดังแสดงในภาพ 4 จะอยู่ในกล่องสี่เหลี่ยมทางด้านขวาภายในกล่องของ VM โดยการตรวจสอบและบันทึกข้อมูลที่เกี่ยวข้องกับกระบวนการเหล่านี้ เรียกว่าการบันทึกประวัติไฟล์สำคัญ

2. ส่วนผู้ให้บริการ (Host)

จากภาพ 4 ส่วนของผู้ให้บริการคือในส่วนกรอบสี่เหลี่ยมด้านขวา ที่มีคำว่า Host อยู่ด้านบน โดยในส่วนนี้จะเป็นส่วนที่ทำการติดตั้ง logger ที่ใช้สำหรับการบันทึกเหตุการณ์ โดยการทำงานของ logger นั้นมี 3 ขั้นตอน โดยมีรายละเอียดดังนี้

ขั้นที่ 1) ผู้ให้บริการรันโปรเซส logger (หมายเลข 1') โดยลือกเกอร์ทำการเรียนรู้ใช้ LibVMI เพื่อเข้าไปใน memU ของ VM โดย LibVMI เป็นไลบรารีที่เขียนในภาษาซี ที่สามารถเข้าไปอ่านข้อมูลที่ถูกจัดเก็บภายใน read_mem ซึ่งอยู่ใน memU

ขั้นที่ 2) หลังจากทีลือกเกอร์เรียกใช้ LibVMI แล้ว LibVMI จะทำหน้าที่เข้าไปอ่านข้อมูลที่ถูเก็บภายใน read_mem (หมายเลข 2')

ขั้นที่ 3) หลังจากที LibVMI เข้าไปอ่านข้อมูลใน read_mem แล้วจะส่งกลับมาที่ลือกเกอร์ แล้วลือกเกอร์จะนำข้อมูลที่ได้รับจาก read_mem มาเก็บไว้ในลือกไฟล์ซึ่งอยู่ใน disk0 (หมายเลข 3')

ความปลอดภัยของไฟล์และประสิทธิภาพของคอมพิวเตอร์เสมือน

ในการให้บริการ IaaS คลาวด์แบบสาธารณะนั้นผู้ให้บริการต้องคำนึงถึงด้านความปลอดภัยของไฟล์และประสิทธิภาพการทำงานของ VM เพื่อให้ผู้ใช้บริการเกิดความมั่นใจในการให้บริการของ IaaS คลาวด์ ในงานวิจัยของ Auxsorn (2020) ได้ทำการพิจารณาประสิทธิภาพและความแม่นยำในการตรวจจับโปรเซสการทำงานของล็อกเกอร์ ซึ่งความแม่นยำนั้นแสดงให้เห็นถึงความปลอดภัยของไฟล์ของผู้ใช้บริการ เมื่อมีความแม่นยำในการตรวจจับโปรเซสการทำงานมากเท่าไร นั้นแสดงให้เห็นว่า จะสามารถที่จะนำข้อมูลโปรเซสที่ตรวจจับได้นั้นไปวิเคราะห์ความผิดปกติในการทำงานของไฟล์ได้มากขึ้นจึงทำให้มีไฟล์ของผู้ใช้บริการมีความปลอดภัยมากขึ้น ในงานวิทยานิพนธ์นี้ ผู้วิจัยจะขอเรียกว่าความแม่นยำ ว่า “ความปลอดภัย” โดยคำว่าความปลอดภัย (Security) คือ กระบวนการปกป้องจากการใช้ทรัพย์สินทางกายภาพหรือดิจิทัลโดยไม่ได้รับอนุญาต (Pfleeger, 2009) ความปลอดภัยเป็นคุณสมบัติที่สำคัญสำหรับการให้บริการ ที่ต้องมีความปลอดภัยและความน่าเชื่อถือทั้งผู้ให้บริการและผู้ใช้บริการตามข้อตกลงระดับบริการ (Service Level Agreement: SLA) (Tutorials Point, 2022) โดยงานวิทยานิพนธ์นี้ได้กำหนดความปลอดภัย คือ ความปลอดภัยของไฟล์ของผู้ให้บริการประสิทธิภาพ (Performance) คือ ประสิทธิภาพในการทำงานของ VM ขณะที่มีการตรวจจับการทำงานโดยล็อกเกอร์ Auxsorn (2020)

ผู้วิจัยจึงได้นำผลการทดลองของ Auxsorn (2020) มาเป็นต้นแบบในการสร้างวิธีการจัดสรรระบบบันทึกเหตุการณ์ เพื่อให้มีการใช้งานที่เหมาะสมกับความต้องการของผู้ใช้บริการมากที่สุด

ผู้วิจัยจึงได้แบ่งค่าระดับของความปลอดภัยและประสิทธิภาพการทำงานทั้งหมด 5 ระดับ (ระดับที่ 1 – 5) (Sheikh, 2020) โดยมีความหมายของแต่ละระดับ ดังนี้

ระดับความปลอดภัยของไฟล์ของผู้ใช้บริการ

โดยระดับความปลอดภัย คือ ระดับของความสามารถในการตรวจจับโปรเซสการทำงาน ของระบบบันทึกเหตุการณ์ ที่ทำให้การทำงานของผู้ใช้งานมีความปลอดภัย โดยอธิบายระดับความปลอดภัย ได้ดังนี้

ระดับ 1 คือ ระดับความปลอดภัยมากที่สุด โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 250 μ s จะได้ค่าความปลอดภัยอยู่ที่ 99.86% ยกตัวอย่างเช่น ผู้ใช้บริการ #1 ทำการระดับความปลอดภัยที่ระดับ 1 ซึ่งล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1 ของตนเองได้เพียงเครื่องเดียวเท่านั้น โดยที่ล็อกเกอร์นี้จะไม่ไปตรวจจับโปรเซสการทำงานใน VMs อื่น ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger1) ต่อ 1 (VM1) โดยที่ล็อกเกอร์นี้จะไม่ใช้งานร่วมกับ VMs อื่น จึงทำให้มีความสามารถในการตรวจจับโปรเซสการทำงานที่ผิดปกติ (malicious process) มาก และค่าความถูกต้องในการตรวจจับโปรเซสการทำงานสูง จึงกำหนดให้อยู่ในระดับความปลอดภัยมากที่สุด ที่ค่าความปลอดภัย 99.86%

ระดับ 2 ระดับความปลอดภัยมาก โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานในแต่ละ VM เป็นระยะเวลา 500 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น ล็อกเกอร์จะไปทำการตรวจจับโปรเซสการทำงานใน VM2 แทน และจะกลับไปทำงานใน VM1 อีกครั้งเมื่อครบรอบการทำงาน จะได้ค่าความปลอดภัยอยู่ที่ 99.82% ซึ่งจะใช้เวลาในการหยุดการตรวจจับ ยกตัวอย่างเช่น ผู้ให้บริการ #2 ทำการเลือกความปลอดภัยระดับ 2 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1 และ VM2 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger2) ต่อ 2 (VM1, VM2) โดยล็อกเกอร์แชร์การทำงานไปยัง VM2 ไม่ใช่แค่ VM1 อย่างเดียวจึงทำให้ต้องเว้นระยะเวลาในการทำงานที่ไปทำการตรวจจับโปรเซสการทำงานทั้ง 2 VMs จึงทำให้ค่าความแม่นยำในตรวจจับลดลงซึ่งหมายถึงความปลอดภัยลดลงไปด้วยเมื่อเทียบกับความปลอดภัยระดับ 1

ระดับ 3 ระดับความปลอดภัยปานกลาง โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 750 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น จะไปทำการตรวจจับโปรเซสการทำงานใน VM2 แทน และเมื่อตรวจจับโปรเซสการทำงานใน VM2 เสร็จแล้ว ล็อกเกอร์ก็จะเลื่อนไปตรวจจับโปรเซสการทำงานใน VM3 เมื่อครบรอบการทำงานของล็อกเกอร์ทั้ง 3 VMs แล้ว ล็อกเกอร์ก็จะไปเริ่มทำงานกับ VM1 ใหม่อีกครั้ง จะได้ค่าความปลอดภัยอยู่ที่ 99.75% ยกตัวอย่างเช่น ผู้ให้บริการ #3 ทำการเลือกความปลอดภัยระดับ 3 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1, VM2 และ VM3 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger3) ต่อ 3 (VM1, VM2, VM3) โดยล็อกเกอร์แชร์การทำงานไปยัง VM3, VM2 ไม่ใช่แค่ VM1 อย่างเดียวจึงทำให้ต้องเว้นระยะเวลาในการทำงานที่ไปทำการตรวจจับโปรเซสการทำงานเพิ่มมากขึ้นเพื่อให้สามารถรองรับการทำงานได้ทั้ง 3 VMs จึงทำให้ค่าความแม่นยำในตรวจจับลดลงซึ่งหมายถึงความปลอดภัยลดลงไปด้วยเมื่อเทียบกับความปลอดภัยระดับ 1 และ 2

ระดับ 4 ระดับความปลอดภัยน้อย โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 1,000 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น จะไปทำการตรวจจับโปรเซสการทำงานใน VM2 แทน และเมื่อตรวจจับโปรเซสการทำงานใน VM2 เสร็จแล้ว ล็อกเกอร์ก็จะเลื่อนไปตรวจจับโปรเซสการทำงานใน VM3 และ VM4 ตามลำดับ เมื่อครบรอบการทำงานของล็อกเกอร์ทั้ง 4 VMs แล้ว ล็อกเกอร์ก็จะไปเริ่มทำงานกับ VM1 ใหม่อีกครั้ง จะได้ค่าความปลอดภัยอยู่ที่ 99.67% ยกตัวอย่างเช่น ผู้ให้บริการ #4 ทำการเลือกความปลอดภัยระดับ 4 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1, VM2, VM3 และ VM4 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger4) ต่อ 4 (VM1, VM2, VM3, VM4) โดยล็อกเกอร์แชร์การทำงานไปยังทั้ง 4 VMs จึงใช้ระยะเวลาในการวนรอบการทำงานของการ

ตรวจจับโปรเซสการทำงานเพิ่มมากขึ้น จึงส่งผลให้ค่าความแม่นยำในตรวจจับลดลงซึ่งหมายถึงความปลอดภัยลดลงไปด้วยเมื่อเทียบกับความปลอดภัยระดับ 3

ระดับ 5 ระดับความปลอดภัยน้อยที่สุด โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 1,250 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น จะไปทำตรวจจับโปรเซสการทำงานใน VM2 แทน และเมื่อตรวจจับโปรเซสการทำงานใน VM2 เสร็จแล้ว ล็อกเกอร์ก็จะเลื่อนไปตรวจจับโปรเซสการทำงานใน VM3, VM4 และ VM5 ตามลำดับ เมื่อครบรอบการทำงานของล็อกเกอร์ทั้ง 5 VMs แล้ว ล็อกเกอร์ก็จะไปเริ่มทำงานกับ VM1 ใหม่อีกครั้ง จะได้ค่าความปลอดภัยอยู่ที่ 86.47% ยกตัวอย่างเช่น ผู้ใช้บริการ #5 ทำการเลือกความปลอดภัยระดับ 5 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1, VM2, VM3, VM4 และ VM5 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger5) ต่อ 5 (VM1, VM2, VM3, VM4, VM5) โดยล็อกเกอร์แชร์การทำงานไปยังทั้ง 5 VMs จึงใช้ระยะเวลาในการวนรอบการทำงานของการตรวจจับโปรเซสการทำงานเพิ่มมากขึ้น จึงส่งผลให้ค่าความแม่นยำในตรวจจับลดลงซึ่งหมายถึงความปลอดภัยลดลงไปด้วยเมื่อเทียบกับความปลอดภัยระดับ 4

ระดับประสิทธิภาพการทำงานของ VM

โดยระดับประสิทธิภาพการทำงานของ VM คือ ระดับของความสามารถในการทำงานของเครื่องผู้ให้บริการให้สามารถทำงานได้อย่างต่อเนื่อง โดยที่มีการหยุดการทำงานของ VM โดยล็อกเกอร์ โดยอธิบายระดับประสิทธิภาพการทำงานของ VM (Auxsorn, 2020) ได้ดังนี้

ระดับ 1 ระดับประสิทธิภาพน้อยที่สุด โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 250 μ s เมื่อครบเวลาที่กำหนดล็อกเกอร์จะเข้าไปตรวจจับโปรเซสการทำงานใน VM1 ต่อ โดยจะมีความถี่ในการเข้าไปตรวจจับโปรเซสการทำงานมากเนื่องจากล็อกเกอร์จะทำการตรวจจับเพียง VM เดียวเท่านั้น จึงทำให้ค่าประสิทธิภาพการทำงานของ VM อยู่ที่ 50.00% ยกตัวอย่างเช่น ผู้ใช้บริการ #1 ทำการระดับประสิทธิภาพที่ระดับ 1 ซึ่งล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1 ของตนเองได้เพียงเครื่องเดียวเท่านั้น โดยที่ล็อกเกอร์นี้จะไม่ไปตรวจจับโปรเซสการทำงานใน VMs อื่น จึงทำให้ VM เกิดการหยุดการทำงาน เพื่อให้ล็อกเกอร์เข้ามาตรวจจับโปรเซสการทำงานได้ จึงทำให้ประสิทธิภาพในการทำงานของเครื่อง VM น้อยเมื่อเทียบกับระดับประสิทธิภาพในระดับอื่น

ระดับ 2 ระดับประสิทธิภาพน้อย โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 500 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น จะไปทำตรวจจับโปรเซสการทำงานใน VM2 แทน และเมื่อตรวจจับโปรเซสการทำงานใน VM2 เสร็จแล้ว ซึ่งในระหว่างที่ VM1 รอให้ล็อกเกอร์ไปทำงานที่ VM2 นั้น การทำงานของ VM1 ก็ยังคงทำงานได้อย่างต่อเนื่อง โดยที่ล็อกเกอร์จะยังไม่เข้ามาหยุดการทำงาน จนกว่าจะครบรอบและกลับมา

ทำการตรวจจับโปรเซสการทำงานใน VM1 อีกครั้ง จึงทำให้ค่าประสิทธิภาพการทำงานของ VM อยู่ที่ 66.00% ยกตัวอย่างเช่น ผู้ใช้บริการ #2 ทำการเลือกประสิทธิภาพระดับ 2 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1 และ VM2 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger2) ต่อ 2 (VM1, VM2) โดยล็อกเกอร์แชรการทำงานไปยัง VM2 ไม่ใช่แค่ VM1 อย่างเดียว ในระหว่างที่ล็อกเกอร์ไปทำงานที่ VM2 นั้น VM1 จึงมีเวลาในการทำงานของ VM ตนเองระหว่างที่รอล็อกเกอร์กลับเข้ามาทำงานใน VM1 นานขึ้น จึงทำให้ VM แต่ละตัวที่เวลาในการทำงานของโดยไม่ถูกหยุดการทำงานมากกว่าระดับประสิทธิภาพในระดับ 1

ระดับ 3 ระดับประสิทธิภาพปานกลาง โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 750 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น จะไปทำการตรวจจับโปรเซสการทำงานใน VM2 แทน และเมื่อตรวจจับโปรเซสการทำงานใน VM2 เสร็จแล้ว ล็อกเกอร์ก็จะเลื่อนไปตรวจจับโปรเซสการทำงานใน VM3 ตามลำดับ ซึ่งในระหว่างที่ VM1 รอให้ล็อกเกอร์ไปทำงานที่ VM อื่น ๆ นั้น การทำงานของ VM1 ก็ยังคงทำงานได้อย่างต่อเนื่องเป็นเวลานานขึ้น โดยที่ล็อกเกอร์จะยังไม่เข้ามาหยุดการทำงาน จนกว่าจะครบรอบและกลับมาทำการตรวจจับโปรเซสการทำงานใน VM1 อีกครั้ง จึงทำให้ค่าประสิทธิภาพการทำงานของ VM อยู่ที่ 73.00% ยกตัวอย่างเช่น ผู้ใช้บริการ #3 ทำการเลือกประสิทธิภาพระดับ 3 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1, VM2 และ VM3 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger3) ต่อ 3 (VM1, VM2, VM3) โดยล็อกเกอร์แชรการทำงานไปยัง VM3, VM2 ไม่ใช่แค่ VM1 อย่างเดียวจึงทำให้ต้องเว้นระยะเวลาในการทำงานที่ไปทำการตรวจจับโปรเซสการทำงานเพิ่มมากขึ้นเพื่อให้สามารถรองรับการทำงานได้ทั้ง 3 VMs จึงทำให้มีเวลาในการทำงานของ VM ตนเองระหว่างที่รอล็อกเกอร์กลับเข้ามาทำงานใน VM ของตนเองนานขึ้นกว่าระดับ 2

ระดับ 4 ระดับประสิทธิภาพมาก โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 1,000 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น จะไปทำการตรวจจับโปรเซสการทำงานใน VM2 แทน และเมื่อตรวจจับโปรเซสการทำงานใน VM2 เสร็จแล้ว ล็อกเกอร์ก็จะเลื่อนไปตรวจจับโปรเซสการทำงานใน VM3 และ VM4 ตามลำดับ ซึ่งในระหว่างที่ VM1 รอให้ล็อกเกอร์ไปทำงานที่ VM อื่น ๆ นั้น การทำงานของ VM1 ก็ยังคงทำงานได้อย่างต่อเนื่องเป็นเวลานานขึ้น โดยที่ล็อกเกอร์จะยังไม่เข้ามาหยุดการทำงาน จนกว่าจะครบรอบและกลับมาทำการตรวจจับโปรเซสการทำงานใน VM1 อีกครั้ง จึงทำให้ค่าประสิทธิภาพการทำงานของ VM อยู่ที่ 80.00% ยกตัวอย่างเช่น ผู้ใช้บริการ #4 ทำการเลือกประสิทธิภาพระดับ 4 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1, VM2, VM3 และ VM4 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger4) ต่อ 4 (VM1, VM2, VM3, VM4) โดยล็อกเกอร์แชรการทำงาน

ทำงานไปยังทั้ง 4 VMs จึงใช้ระยะเวลาในการวนรอบการทำงานของการตรวจจับโปรเซสการทำงานเพิ่มมากขึ้น ทำให้ในแต่ละ VM มีเวลาในการทำงานของตัวเองมากขึ้นกว่าในระดับ 3

ระดับ 5 ระดับประสิทธิภาพมากที่สุด โดยล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 เป็นระยะเวลา 1,250 μ s ซึ่งในระหว่างที่ล็อกเกอร์หยุดการตรวจจับโปรเซสการทำงานใน VM1 นั้น จะไปทำตรวจจับโปรเซสการทำงานใน VM2 แทน และเมื่อตรวจจับโปรเซสการทำงานใน VM2 เสร็จแล้ว ล็อกเกอร์ก็จะเลื่อนไปตรวจจับโปรเซสการทำงานใน VM3, VM4 และ VM5 ตามลำดับ ซึ่งในระหว่างที่ VM1 รอให้ล็อกเกอร์ไปทำงานที่ VM อื่น ๆ นั้น การทำงานของ VM1 ก็ยังคงทำงานได้อย่างต่อเนื่องเป็นเวลานานขึ้น โดยที่ล็อกเกอร์จะยังไม่เข้ามาหยุดการทำงาน จนกว่าจะครบรอบและกลับมาทำการตรวจจับโปรเซสการทำงานใน VM1 อีกครั้ง จึงทำให้ใช้เวลานานกว่าระดับ 4 จึงทำให้ค่าประสิทธิภาพการทำงานของ VM อยู่ที่ 83.00% ยกตัวอย่างเช่น ผู้ใช้บริการ #5 ทำการเลือกความปลอดภัยระดับ 5 ล็อกเกอร์ 1 ตัวจะทำการตรวจจับโปรเซสการทำงานใน VM1, VM2, VM3, VM4 และ VM5 ด้วย ซึ่งเป็นลักษณะการตรวจจับโปรเซสการทำงานแบบ 1 (logger5) ต่อ 5 (VM1, VM2, VM3, VM4, VM5) โดยล็อกเกอร์แชร์การทำงานไปยังทั้ง 5 VMs จึงใช้ระยะเวลาในการวนรอบการทำงานของการตรวจจับโปรเซสการทำงานเพิ่มมากขึ้น ทำให้ในแต่ละ VM มีเวลาในการทำงานของตัวเองมากขึ้นกว่าในระดับ 4

ซึ่งจากการอธิบายความหมายของแต่ละระดับในข้างต้น ทางผู้วิจัยได้พิจารณาจากค่าที่ถูกกำหนดให้ logger sleeping time คือค่า sleeping time ของล็อกเกอร์ที่จะหยุดการตรวจจับโปรเซสการทำงานโดยมีค่าในหน่วยไมโครวินาที (μ s) ที่ 250 μ s., 500 μ s., 750 μ s., 1,000 μ s., 1,250 μ s. ตามลำดับ และกำหนดให้ logger process time คือ เวลาในการทำงานของล็อกเกอร์มีค่าเท่ากับ 250 μ s. ซึ่งได้ข้อมูลดังตาราง 1 และ ตาราง 2 ดังนี้

ตาราง 1 แสดงระดับความปลอดภัยของไฟล์ของผู้ใช้บริการ

ระดับความปลอดภัย	ความหมาย	Logger sleeping time	ร้อยละของความปลอดภัย
5	ระดับความปลอดภัยน้อยที่สุด	1,250 μ s.	86.47%
4	ระดับความปลอดภัยน้อย	1,000 μ s.	99.67%
3	ระดับความปลอดภัยปานกลาง	750 μ s.	99.75%
2	ระดับความปลอดภัยมาก	500 μ s.	99.82%
1	ระดับความปลอดภัยมากที่สุด	250 μ s.	99.86%

ที่มา: Auxsorn, 2020

จากตาราง 1 ระดับความปลอดภัยน้อยที่สุด จะหมายความว่าล็อกเกอร์เข้าไปทำการตรวจจับโปรเซสการทำงานในเครื่อง VM ทุก ๆ 1,250 μ s. โดยคิดเปรียบเทียบความปลอดภัยได้ร้อยละ 85.47% ซึ่งถ้าระดับความปลอดภัยน้อยนั้นหมายถึงว่าล็อกเกอร์เข้าไปทำการตรวจจับโปรเซสการทำงานในเครื่อง VM ทุก ๆ 1,000 μ s. โดยคิดเปรียบเทียบความปลอดภัยได้ร้อยละ 99.67% จนกระทั่งระดับความปลอดภัยมากที่สุดนั้นหมายถึงว่าล็อกเกอร์เข้าไปทำการตรวจจับโปรเซสการทำงานในเครื่อง VM ทุก ๆ 250 μ s. โดยคิดเปรียบเทียบความปลอดภัยได้ร้อยละ 99.86%

ซึ่งจากงานวิจัยของ Auxsorn (2020), Chan-In (2017), Jaiboo (2020), และ Wongthai (2017) ได้เน้นในการวัดความแม่นยำในการตรวจจับโปรเซสการทำงานจากเครื่อง VM ให้สามารถตรวจจับโปรเซสการทำงานให้ได้มากที่สุด จึงพบว่าเมื่อล็อกเกอร์เข้าไปตรวจจับโปรเซสการทำงานในเครื่อง VM บ่อยครั้งมากเท่าไร ค่าความแม่นยำในการตรวจจับโปรเซสการทำงานจากเครื่อง VM จะมากขึ้น เป็นเพราะว่ามีกรเข้าไปตรวจจับโปรเซสการทำงานได้เยอะ และนำไปสู่การนำข้อมูลที่ได้จากการตรวจจับโปรเซสการทำงานมาทำการวิเคราะห์หากการทำงานที่ไม่พึงประสงค์หรือถูกโจมตีจากผู้ไม่หวังดีได้

ตาราง 2 แสดงระดับประสิทธิภาพของ VM

ระดับประสิทธิภาพ	ความหมาย	Logger sleeping time	ร้อยละของประสิทธิภาพ
5	ระดับประสิทธิภาพในการทำงานมากที่สุด	1,250 μ s.	83.00%
4	ระดับประสิทธิภาพในการทำงานมาก	1,000 μ s.	80.00%
3	ระดับประสิทธิภาพในการทำงานปานกลาง	750 μ s.	73.00%
2	ระดับประสิทธิภาพในการทำงานน้อย	500 μ s.	66.00%
1	ระดับประสิทธิภาพในการทำงานน้อยที่สุด	250 μ s.	50.00%

ที่มา: Auxorn, 2020

จากตาราง 2 ระดับประสิทธิภาพมากที่สุด จะหมายความว่าสล็อตเกอร์เข้าไปทำการตรวจจับโปรเซสการทำงานในเครื่อง VM ทุก ๆ 1,250 μ s. โดยคิดเปรียบเทียบประสิทธิภาพได้ร้อยละ 83.00% ซึ่งถ้าระดับประสิทธิภาพมากนั้นหมายถึงว่าสล็อตเกอร์เข้าไปทำการตรวจจับโปรเซสการทำงานในเครื่อง VM ทุก ๆ 1,000 μ s. โดยคิดเปรียบเทียบประสิทธิภาพได้ร้อยละ 80.00% จนกระทั่งระดับประสิทธิภาพน้อยที่สุดนั้นหมายถึงว่าสล็อตเกอร์เข้าไปทำการตรวจจับโปรเซสการทำงานในเครื่อง VM ทุก ๆ 250 μ s. โดยคิดเปรียบเทียบประสิทธิภาพได้ร้อยละ 50.00%

จากงานวิจัยของ Auxorn (2020) ผู้วิจัยได้นำข้อมูลการวัดประสิทธิภาพการทำงานของ VM และระดับความถูกต้องในการตรวจจับข้อมูลจากเครื่อง VM มากำหนดเป็นระดับ 5 ระดับเพื่อนำมาใช้ในการกำหนดข้อจำกัดของการสร้างโมเดลทางคณิตศาสตร์ในการจัดสรรระบบบันทึกเหตุการณ์ ซึ่งผู้วิจัยได้นำผลการทดลองมากำหนดค่าให้เป็นช่วงระยะเวลาในการทำงานในช่วงระยะเวลาที่เท่ากันด้วยการหาค่าเฉลี่ยของข้อมูล

กระบวนการหาค่าที่เหมาะสม (Optimization)

กระบวนการหาค่าที่เหมาะสม (Optimization) เป็นสิ่งที่สำคัญในการสร้างระบบให้มีประสิทธิภาพ (Zeng, 2023) และเป็นขั้นตอนวิธีในการตัดสินใจเลือกผลลัพธ์ที่ดีที่สุดในการแก้ปัญหาที่เหมาะสม (Optimization Problem) โดยการหาค่าที่เหมาะสมสามารถกำหนดได้ว่าเป็นกระบวนการในการค้นหาเงื่อนไขที่ให้ค่ามากที่สุดหรือค่าน้อยที่สุดของฟังก์ชัน การทำงานกระบวนการหาค่าที่เหมาะสม เกิดจากการที่นักวิทยาศาสตร์ต้องการแก้โจทย์ปัญหาทางคณิตศาสตร์ เพื่อให้ได้ผลลัพธ์ที่เหมาะสมที่สุด ซึ่งถูกนำมาประยุกต์ใช้กับปัญหาทางด้านฟิสิกส์บางประเภท หรือทางด้านวิศวกรรม เช่น การออกแบบเครื่องบินหรือยานอวกาศให้มีน้ำหนักน้อยที่สุด การหาเส้นทางการบินที่สั้นที่สุด เป็นต้น (Rao, 2019) ในปัจจุบันการหาค่าที่เหมาะสมไม่ได้จำกัดในด้านทางวิทยาศาสตร์ แต่มีความสำคัญในทุก ๆ สาขาวิชาโดยเฉพาะอย่างยิ่งทางด้านเศรษฐศาสตร์ เนื่องจากนำหลักการของการหาค่าที่เหมาะสมมาช่วยในเรื่องของเงินลงทุน และผลกำไร เพื่อเป็นการลดต้นทุนการผลิต และเพิ่มผลกำไรจากการผลิต เป็นต้น ในแบบจำลองทางคณิตศาสตร์

แบบจำลองทางคณิตศาสตร์ (Mathematical Model) คือ แบบจำลองนามธรรม ที่อธิบายบางส่วนของสถานการณ์จริงในภาษาทางคณิตศาสตร์ ที่มีตัวแปรบางอย่างในการแทนค่าที่ไม่รู้จัก และแสดงออกมาในรูปของสูตรทางคณิตศาสตร์ที่สามารถนำมาทำการคำนวณค่าต่าง ๆ ที่ต้องการได้ (Bisschop, 2006)

1. หลักการพื้นฐานของการทำ optimization

ปัจจุบันได้มีการนำเทคนิคการจำลองของสถานการณ์เพื่อหาคำตอบที่เหมาะสม (Simulation-Optimization Technique) มาใช้ในการแก้ปัญหาในสถานการณ์จริงต่าง ๆ ในรูปของแบบจำลองทางคณิตศาสตร์ ที่ให้ผลลัพธ์ในเชิงปริมาณ (Quantity) โดยการหาค่าที่เหมาะสมมีจุดประสงค์เพื่อหาค่าสูงสุด (Maximum) หรือค่าต่ำสุด (Minimum) ของฟังก์ชันจุดประสงค์ (Objective Function) ที่กำหนด โดยรูปแบบจำลองทางคณิตศาสตร์ ประกอบไปด้วย ฟังก์ชันจุดประสงค์ และเซตข้อจำกัด (Constraint Set) ดังตัวอย่างสมการ (1) (Rao, 2019)

$$X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ which minimizes/maximize } f(X) \quad (1)$$

Subject to:

$$\begin{aligned} g_j(X) &\leq 0, & j &= 1, 2, \dots, m \\ l_j(X) &= 0, & j &= 1, 2, \dots, p \end{aligned}$$

โดยที่ X คือ เวกเตอร์ที่ n-dimensional เรียกว่า Design Vector

$f(X)$ คือ ฟังก์ชันจุดประสงค์

$g_j(X)$ และ $l_j(X)$ คือ ข้อจำกัดที่ระบุในรูปแบบของการเท่ากัน (สมการ) และการไม่เท่ากัน (อสมการ)

การนิยามปัญหาการหาค่าเหมาะสมจะต้องกำหนด 3 องค์ประกอบสำคัญ ดังนี้

1. ตัวแปรการตัดสินใจ (Decision Variable)

ตัวแปรการตัดสินใจ คือ ตัวแปรที่เป็นคำตอบของการแก้ปัญหาที่เหมาะสม ยกตัวอย่างเช่น ในงานวิทยานิพนธ์นี้ต้องการหาจำนวนล็อกเกอร์ที่ว่างมากที่สุด

2. ฟังก์ชันจุดประสงค์ (Objective function)

ฟังก์ชันจุดประสงค์ คือ ฟังก์ชันที่ต้องการหาค่าสูงสุดหรือต่ำสุด เป็นเกณฑ์ที่เกี่ยวข้องกับการออกแบบที่เหมาะสม (Rao, 2019) โดยกำหนดฟังก์ชันจุดประสงค์ให้อยู่ในรูปแบบของสมการทางคณิตศาสตร์ที่มีตัวแปรที่ตัดสินใจประกอบอยู่ ยกตัวอย่างเช่น ในงานวิทยานิพนธ์นี้ต้องการหาจำนวนล็อกเกอร์ที่ว่างมากที่สุด ดังนั้น จึงได้นำเอาตัวแปรที่ได้ออกแบบไว้ มาสร้างฟังก์ชันจุดประสงค์

3. ข้อจำกัดต่าง ๆ (Constraint)

ข้อจำกัด คือ ข้อจำกัดที่แสดงถึงข้อจำกัดที่เกี่ยวกับพฤติกรรมหรือประสิทธิภาพของระบบเรียกว่า ข้อจำกัดของพฤติกรรมหรือการทำงาน (Zeng, 2023) หรือข้อจำกัดของฟังก์ชันจุดประสงค์ (Rao, 2019) ในงานวิทยานิพนธ์นี้มีข้อจำกัดในเรื่องของระดับความปลอดภัยและระดับประสิทธิภาพ

ในงานวิจัยทางด้านคลาวด์ ได้มีการนำเทคนิคการหาค่าที่เหมาะสม มาใช้ในการทำงานให้มีประสิทธิภาพ เช่น ในงานวิจัยของ Hosseini et al. (2021) ได้ใช้อัลกอริธึม Whale Optimization Algorithm ในการแก้ปัญหาในเรื่องของการจัดสรรทรัพยากรให้เหมาะสมกับความต้องการในการทำงานของผู้ใช้บริการบนคลาวด์ เพื่อลดระยะเวลาที่โปรแกรมทำงานจากการร้องขอของผู้ใช้บริการ Ala'a Al-Shaikh (2016) ได้นำวิธีการในการหาวิธีการที่เหมาะสมมาใช้ในการแก้ปัญหาเรื่องของการใช้ทรัพยากรบนระบบคลาวด์ได้ ด้วยการใช่วิธี Greedy Wei (2020) ได้ทำการหาวิธีการที่เหมาะสมในการจัดตารางการทำงานบนคลาวด์เพื่อแก้ปัญหาเรื่องของการโหลดที่ไม่สมดุลความเร็วในการรวมกันซ้ำ หรือใช้ทรัพยากรของ VM ที่ต่ำ ด้วยการปรับปรุงวิธีการของ Ant Colony Optimization Megahed et al. (2019) ได้นำเสนอการออกแบบวิธีการของคลาวด์ที่ใช้โมเดลทางคณิตศาสตร์ในการออกแบบวิธีการที่เพิ่มประสิทธิภาพการทำงานของคลาวด์ที่มีผลต่อค่าใช้จ่ายที่ลดลงในมุมมองของผู้ใช้บริการ ด้วยการใช้นิยาม Greedy Algorithm และ Random Algorithm Khan & Santhosh (2022) ได้ใช้กระบวนการในการหาค่าที่เหมาะสมในการลดการใช้พลังงาน เวลา

ในการคำนวณ และค่าใช้จ่ายในการย้ายข้อมูล โดยการใช้ทรัพยากรให้เกิดประโยชน์สูงสุดในการเคลื่อนย้าย VM ด้วยการใช้เทคนิคของ Cuckoo Search Optimization ร่วมกับ Particle Swarm Optimization เป็นต้น

การจัดการทรัพยากรเพื่อเพิ่มประสิทธิภาพในการทำงานของคลาวด์ (Resource Management Optimization in Cloud Computing) หมายถึง กระบวนการในการจัดสรรการใช้ทรัพยากรในคลาวด์ได้อย่างมีประสิทธิภาพ เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดในด้านของประสิทธิภาพการทำงาน ความคุ้มค่าต่อต้นทุน และความยืดหยุ่นในการขยายขนาด การจัดสรรทรัพยากรเป็นหนึ่งในแง่มุมที่สำคัญของคลาวด์เพื่อให้การแยกประสิทธิภาพและการใช้งานฮาร์ดแวร์พื้นฐานอย่างมีประสิทธิภาพ (Jennings & Stadler, 2015) ซึ่งทั้งหมดนี้จะเกี่ยวข้องกับการจัดสรรทรัพยากร การกระจายภาระการทำงาน การตั้งตารางงาน โดยมีเป้าหมายในการเพิ่มประสิทธิภาพการใช้ทรัพยากร ลดความสูญเสียทรัพยากร และให้ความต้องการทรัพยากรที่มีระดับคุณภาพได้ถูกต้องตามความต้องการในการใช้งาน (Service Level Agreement: SLA)

เทคนิคการทำเพื่อเพิ่มประสิทธิภาพในการจัดการทรัพยากรในคอมพิวเตอร์คลาวด์สามารถสรุปได้ดังนี้

1. การจัดสรรทรัพยากร (Resource Provisioning): เป็นการจัดสรรทรัพยากรของผู้ให้บริการแก่ลูกค้า (Manvi & Shyam, 2014) การกำหนดปริมาณทรัพยากรที่เหมาะสมของเครื่อง VM ในการจัดเก็บข้อมูล และแบนด์วิดท์ของเครือข่าย โดยพิจารณาความต้องการและลักษณะของโหนดการทำงาน เพื่อให้การใช้ทรัพยากรเป็นไปอย่างมีประสิทธิภาพและหลีกเลี่ยงการใช้ทรัพยากรน้อยหรือมากเกินไป การจัดสรรทรัพยากรตามความต้องการนั้นเป็นสิ่งที่ผู้ให้บริการคำนึงถึงเนื่องจากการปรับเพิ่มลดทรัพยากร เช่น CPU, RAM เป็นต้น ระบบของผู้ให้บริการต้องสามารถรองรับการปรับขนาดได้โดยอัตโนมัติ เพื่อให้เหมาะสมกับการเปลี่ยนแปลงปริมาณงานและรับประกันประสิทธิภาพที่สม่ำเสมอด้วยต้นทุนที่ต่ำกว่า (Aldossary, 2021) เช่น งานวิจัยของ Megahed et al. (2019) ออกแบบวิธีการของคลาวด์โดยลดค่าใช้จ่ายในมุมมองของผู้ใช้บริการ Ala'a Al-Shaikh (2016) แก้ปัญหาเรื่องของการใช้ทรัพยากรบนระบบคลาวด์ด้วยกระบวนการหาค่าที่เหมาะสม Zhao et al. (2015) วิธีการจัดวาง VM ตามประสิทธิภาพของการบริการ โดยนำลักษณะการใช้ทรัพยากรมาใช้ในการพิจารณา และ Malekloo et al. (2018) ได้รวบรวมและจัดวาง VM โดยมีเป้าหมายเพื่อลดการใช้พลังงาน การสิ้นเปลืองทรัพยากรหน่วยประมวลผล และต้นทุนการสื่อสาร เป็นต้น

2. ประสิทธิภาพการใช้พลังงาน (Energy Efficiency): กระบวนการปรับปรุงการจัดการทรัพยากรเพื่อลดการใช้พลังงานและสนับสนุนการคอมพิวเตอร์เพียงพอ ประสิทธิภาพการใช้พลังงานเป็นหนึ่งในประเด็นหลักที่ต้องแก้ไขในระบบคลาวด์ (Gao et al., 2014) เช่น งานวิจัยของ Khan & Santhosh (2022) ได้ใช้กระบวนการหาค่าที่เหมาะสมในการลดการใช้พลังงาน เวลาในการ

คำนวณ และค่าใช้จ่ายในการย้ายข้อมูล โดยการใช้ทรัพยากรให้เกิดประโยชน์สูงสุดในการเคลื่อนย้าย VM Malekloo et al. (2018) ได้รวบรวมและจัดวาง VM โดยมีเป้าหมายเพื่อลดการใช้พลังงาน และ Bermejo et al. (2016) ลดการใช้พลังงานเพื่อให้ระบบมีประสิทธิภาพมากที่สุดในการจัดสรรทรัพยากร เป็นต้น

3. การจัดตารางการทำงาน (Task Scheduling): กระบวนการกำหนดงานหรืองานให้กับทรัพยากรที่พร้อมใช้งานตามปัจจัยต่างๆ เช่น ความพร้อมของทรัพยากร นายหน้าการตั้งตารางงานจะมุ่งเน้นการลดเวลาในการทำงาน การใช้ทรัพยากรอย่างมีประสิทธิภาพ และบรรลุเป้าหมายการทำงาน การจัดตารางงานในคลาวด์นั้นส่งผลกระทบต่อการใช้ทรัพยากรและต้นทุนการดำเนินงานของระบบ (Chen et al., 2020) เช่นงานวิจัยของ Chen et al. (2020) ได้ทำการปรับปรุงการทำงานของ Whale Optimization Algorithm ให้มีความสามารถในการค้นหาวิธีการที่เหมาะสมในการจัดตารางการทำงานบนระบบคลาวด์ Bacanin et al. (2019) ได้นำเสนออัลกอริทึม Grey Wolf Optimizer ในการจัดตารางการทำงานให้เหมาะสมและมีประสิทธิภาพ โดยลดระยะเวลาช่วงกว้างของเวลาการทำงานให้มีค่าน้อยที่สุด (makespan) เป็นต้น

4. การจัดสรรภาระงาน (Load Balancing): กระบวนการกระจายภาระงานอย่างเท่าเทียมกันไปยังทรัพยากรที่พร้อมใช้งาน ช่วยให้ผู้ใช้ให้บริการคลาวด์จัดการความต้องการปริมาณงานหรือความต้องการของแอปพลิเคชันโดยการกระจายทรัพยากรระหว่างคอมพิวเตอร์ เครือข่าย หรือ เซิร์ฟเวอร์ (Devaraj et al., 2020) เพื่อหลีกเลี่ยงปัญหาการเกิดข้อจำกัดของทรัพยากรและเพิ่มประสิทธิภาพของระบบโดยรวม อัลกอริทึมการทำลดพิจารณาปัจจัยต่างๆ เช่น การใช้ทรัพยากร ค่าความล่าช้าของเครือข่าย และการกระจายงานเพื่อจัดสรรค่าขอหรืองานเข้าสู่ทรัพยากรที่เหมาะสมที่สุด เช่น งานวิจัยของ Shafiq et al. (2020) ได้ทำการเพิ่มประสิทธิภาพทรัพยากรและปรับปรุง Load Balancing ในมุมมองของพารามิเตอร์งานด้วยการใช้อัลกอริทึม Load Balancing เป็นต้น

การจัดการทรัพยากรที่มีประสิทธิภาพในคอมพิวเตอร์คลาวด์ จำเป็นต้องใช้เทคนิคและอัลกอริทึมการปรับปรุงที่ต่างกัน เช่น Mathematical Programming, Heuristic Algorithm, Machine Learning และ Evolutionary Algorithms วิธีการเหล่านี้จะช่วยให้หาวิธีการจัดสรรทรัพยากร การจัดตารางการทำงานที่ดีที่สุด โดยจะแปรผกผันระหว่างประสิทธิภาพ ต้นทุน การใช้พลังงาน และปัจจัยอื่น ๆ ที่เกี่ยวข้อง

จากการศึกษางานวิจัยที่เกี่ยวข้องกับการหาค่าที่เหมาะสมในการจัดสรรทรัพยากรบนคลาวด์ ผู้วิจัยขอสรุปข้อมูลได้ดังตาราง

ตาราง 3 แสดงการสรุปการจัดการทรัพยากรบนคลาวด์ด้วยการทำ optimization

ผู้วิจัย	การแก้ปัญหาที่เหมาะสม	เทคนิค
Megahed et al. (2019)	หาค่าใช้จ่ายที่น้อยที่สุดในการจัดสรร การให้บริการคลาวด์ตามความต้องการของผู้ใช้บริการ ด้วยโมเดลทางคณิตศาสตร์	Greedy Algorithm, Random Algorithm
Hosseini et al. (2021)	จัดสรรทรัพยากรให้เหมาะสมกับความต้องการในการทำงานของผู้ใช้บริการบนคลาวด์	Whale Optimization Algorithm
Ala'a Al-Shaikh (2016)	หาค่าประโยชน์ที่มากที่สุดจากการใช้งานทรัพยากรบนคลาวด์ ด้วยโมเดลทางคณิตศาสตร์	Greedy Algorithm
Chen et al. (2020)	การหาวิธีการในการจัดตารางการทำงานของ VM โดยให้มีการใช้ทรัพยากรที่น้อยที่สุด ด้วยโมเดลทางคณิตศาสตร์	Improve Whale Optimization Algorithm
Khan & Santhosh (2022)	การลดการใช้พลังงานในการทำ VM migration	Cuckoo Search Optimization + Particle Swarm Optimization
Kumrai et al. (2016)	หาค่าผลประโยชน์ที่มากที่สุดให้กับนายหน้าของระบบคลาวด์ (cloud broker) โดยการลดเวลาตอบสนองของคำร้องขอและการใช้พลังงาน ด้วยโมเดลทางคณิตศาสตร์	Multiobjective PSO (MOPSO) compared Genetic Algorithm and Random Search Algorithm
Bacanin et al. (2019)	การจัดตารางการทำงานให้เหมาะสมและมีประสิทธิภาพ โดยลดระยะเวลาช่วงกว้างของเวลาการทำงานให้มีค่าน้อยที่สุด ด้วยโมเดลทางคณิตศาสตร์	Grey Wolf Optimizer
Wei (2020)	ได้ทำการหาวิธีการที่เหมาะสมในการจัดตารางการทำงานบนคลาวด์ โดยใช้เวลาน้อยที่สุดที่ผู้ใช้งานรอคอย	Improve Ant Colony Optimization

ตาราง 3 (ต่อ)

ผู้วิจัย	การแก้ปัญหาที่เหมาะสม	เทคนิค
Malekloo et al. (2018)	การรวบรวมและจัดวาง VM โดยมีเป้าหมายเพื่อลดการใช้พลังงาน การสิ้นเปลืองทรัพยากรหน่วยประมวลผล และต้นทุนการสื่อสาร	Multi-objective Ant Colony Optimization (MACO)
Shafiq et al. (2020)	เพิ่มประสิทธิภาพทรัพยากรและปรับปรุง load balancing ในมุมมองของพารามิเตอร์งาน และใช้เวลาในการดำเนินงาน และค่า makespan น้อยที่สุด	Load balancing Algorithm
Our	การหาจำนวนระบบบันทึกเหตุการณ์ที่ไม่ได้มีภาระงานมากที่สุด เพื่อลดปริมาณการใช้งาน CPU และ RAM ในเครื่องของผู้ให้บริการ ด้วยโมเดลทางคณิตศาสตร์	Random Search Algorithm

ในงานวิทยานิพนธ์นี้จึงใช้วิธีการหาล็อกเกอร์ที่ไม่ได้มีภาระงาน (workload Free Logger) มากที่สุด เพื่อลดปริมาณการใช้ทรัพยากรของ CPU ให้มีการใช้งานที่น้อยที่สุด และให้ระบบคอมพิวเตอร์สามารถทำงานต่อไปได้ เพื่อรองรับการขยาย VM ได้มากขึ้น และเป็นไปตามความต้องการของผู้ใช้บริการให้มากที่สุด โดยการสร้างโมเดลทางคณิตศาสตร์ด้วยการใช้เทคนิค Random search algorithm ในการแก้ปัญหาจากโมเดลทางคณิตศาสตร์ที่ได้สร้างขึ้นมา ซึ่งเป็นพื้นฐานของเทคนิคการทำงานในรูปแบบอื่น ทั้งนี้ประสิทธิภาพของอัลกอริทึมการค้นหาแบบสุ่มนั้นขึ้นอยู่กับพารามิเตอร์และปัญหาในการหาค่าที่เหมาะสม ซึ่งในงานวิทยานิพนธ์นี้มีพารามิเตอร์ไม่มากและปัญหาไม่ได้ซับซ้อนจึงเลือกใช้เทคนิคนี้เข้ามาใช้ในการแก้ปัญหา โดยผู้วิจัยไม่ได้ทำการเปรียบเทียบประสิทธิภาพ แต่ใช้เพื่อแสดงให้เห็นการทำงานที่มีความถูกต้องของโมเดลทางคณิตศาสตร์

ขั้นตอนวิธีการค้นหาแบบสุ่ม (Random Search Algorithm)

อัลกอริทึมการค้นหาแบบสุ่มคืออัลกอริทึมประเภทหนึ่งที่รวมเอาการสุ่มหรือความน่าจะเป็น ซึ่งโดยปกติจะใช้ตัวสร้างตัวเลขสุ่มหลอกในวิธีการของมัน อัลกอริทึมประเภทนี้ในเอกสารทางวิชาการอาจเรียกว่าวิธีมอนติคาร์โล (Monte Carlo Method) หรืออัลกอริทึมสุ่ม (Zabinsky, 2009) อัลกอริทึมการค้นหาแบบสุ่มที่ได้รับการพัฒนา ได้แก่ Simulated Annealing, Genetic Algorithms, Evolutionary Programming, Particle Swarm Optimization, Ant Colony Optimization, Cross-Entropy, Stochastic Approximation, Multi-Start, Clustering Algorithms, and other

Random Search Algorithm วิธีการเหล่านี้ถูกนำมาใช้อย่างกว้างขวางในการแก้ปัญหา การเพิ่มประสิทธิภาพในการหาค่าที่เหมาะสมที่แท้จริงทั้งแบบต่อเนื่องและแบบไม่ต่อเนื่อง ไม่ว่าจะเป็นงานวิจัยของ Wei (2020) ที่มีการปรับปรุงวิธีการของ Ant Colony Optimization ในการจัดเวลาการทำงานที่เหมาะสมในสภาพแวดล้อมคลาวด์ งานวิจัยของ Al-Muhammed (2018) ได้มีการนำข้อดีของ Random Search Algorithm ที่เป็นการค้นหาที่มีประสิทธิภาพมาปรับปรุง ซึ่งเรียกว่า Moving Directives Algorithm (MDA-3) ในการหาค่าที่เหมาะสมที่แท้จริง (Global Optimization) งานวิจัยของ Dogani & Khunjush (2021) ได้ทำการใช้ Genetic Algorithm ร่วมกับ Particle Swarm Optimization ในการนำเสนอองค์ประกอบการให้บริการคลาวด์ซึ่งสะท้อนถึงการเลือกเซิร์ฟเวอร์สำหรับแต่ละบริการตามคุณภาพบริการเพื่อเพิ่มความพึงพอใจสูงสุดแก่ผู้ใช้ Mostafavi & Hakami (2020) เสนอวิธีการจัดตารางเวลาตามการเรียนรู้เสริมแรง การปรับให้เข้ากับสภาพแวดล้อมและการตอบสนองต่อคำขอที่ไม่แน่นอน ด้วยการใช้เทคนิค Stochastic Approximation สำหรับในงานวิทยานิพนธ์นี้ผู้วิจัยได้ทำการเลือกวิธีการแบบ Random Search ในการหาค่าฟังก์ชันจุดประสงค์ ซึ่งในงานวิจัยของ Bergstra & Bengio (2012) ได้ทำการทดลองการกำหนด Hyper Parameter ของการทำ Optimization ด้วยการใช้ Random Search และ Grid Search พบว่า Random Search ให้ผลลัพธ์ที่ดีกว่า grid search ในการทำ Machine Learning และ Zabinsky (2009) ได้ระบุว่าวิธีการของ Random Search ได้รับการรับรองว่าเป็นวิธีการในการหาวิธีการที่ดีได้อย่างรวดเร็วด้วยความน่าจะเป็น ดังนั้นผู้วิจัยจึงเลือกขั้นตอนวิธีการค้นหาแบบสุ่มเพื่อมาใช้ในการสร้างแบบจำลองสถานการณ์การทำงานของกระบวนการหาค่าที่เหมาะสม

ตาราง 4 แสดงการเปรียบเทียบข้อดี ข้อด้อยของอัลกอริธึมในการหาค่าที่เหมาะสม

อัลกอริธึม	ข้อดี	ข้อด้อย
Genetic Algorithm (GA)	<ol style="list-style-type: none"> 1. ความยืดหยุ่นและความสามารถในการค้นหาที่กว้าง 2. สามารถหาคำตอบได้รวดเร็ว 3. ทำงานกับข้อมูลที่ไม่ต่อเนื่องได้ 	<ol style="list-style-type: none"> 1. ใช้ทรัพยากรในการคำนวณมาก 2. การตั้งค่าพารามิเตอร์ส่งผลต่อผลลัพธ์ที่ได้ 3. อาจเกิดปัญหาในการพบคำตอบที่ซ้ำซ้อน
Particle Swarm Optimization (PSO)	<ol style="list-style-type: none"> 1. สามารถหาคำตอบได้รวดเร็ว 2. การเคลื่อนที่แบบสุ่มทำให้หลีกเลี่ยงปัญหาของการ convergence ได้ 3. เป็นเทคนิคที่มีความยืดหยุ่นและสามารถใช้ปัญหาที่มีลักษณะต่าง ๆ ได้ 	<ol style="list-style-type: none"> 1. การติดขัดของคำตอบที่พบในแต่ละรอบการทำงาน 2. ความซ้ำซ้อนในการกำหนดพารามิเตอร์ 3. การติดตามคำตอบที่ไม่เหมาะสม
Ant Colony Optimization (ACO)	<ol style="list-style-type: none"> 1. สามารถค้นหาคำตอบได้อย่างรวดเร็ว 2. สามารถสร้างการค้นหาลู่ทางที่แตกต่างกันได้ 3. สามารถปรับเปลี่ยนพารามิเตอร์ได้ 	<ol style="list-style-type: none"> 1. ใช้ทรัพยากรในการคำนวณมาก 2. ความซ้ำซ้อนในการกำหนดพารามิเตอร์ 3. มีปัญหาสำหรับปัญหาที่มีความเปลี่ยนแปลงต่อเนื่อง
Greedy Algorithm	<ol style="list-style-type: none"> 1. สามารถใช้งานได้กับปัญหาใหญ่ 2. นำไปประยุกต์ใช้งานได้ง่าย 3. ทำงานได้อย่างรวดเร็ว 	<ol style="list-style-type: none"> 1. อาจให้ผลลัพธ์ที่ผิดพลาดได้ 2. ไม่สามารถใช้งานกับปัญหาที่มีข้อจำกัดหลาย ๆ ข้อจำกัด 3. ไม่สามารถให้ผลลัพธ์ที่ดีที่สุดได้
Random Search Algorithm	<ol style="list-style-type: none"> 1. สามารถค้นหาคำตอบที่ซ่อนอยู่จากพื้นที่ที่ไม่ได้ทำการค้นหาก่อนหน้า 2. สามารถกำหนดพื้นที่ในการค้นหาที่กว้างได้ตามช่วงที่กำหนด 3. นำไปปรับใช้งาน ไม่มีความซับซ้อนของพารามิเตอร์ 	<ol style="list-style-type: none"> 1. อาจใช้ทรัพยากรในการสร้างตัวอย่างของคำตอบมากขึ้น 2. ความแม่นยำในการค้นหา คำตอบ ต้องทำการทดลองหลาย ๆ ครั้ง 3. ไม่คำนึงถึงความสัมพันธ์หรือลำดับของตัวแปร

ในงานวิทยานิพนธ์นี้ได้เลือกเทคนิค Random Search Algorithm มาใช้ในการแก้ปัญหาในการหาค่าที่เหมาะสม โดยข้อดีของวิธีการนี้คือ การกำหนดพารามิเตอร์ที่ไม่มีความซ้ำซ้อน และสามารถกำหนดพื้นที่ในการค้นหาแต่ละรอบได้ แต่สำหรับประสิทธิภาพของอัลกอริธึมการค้นหาแบบสุ่มนั้นขึ้นอยู่กับพารามิเตอร์และปัญหาในการหาค่าที่เหมาะสม ซึ่งในงานวิทยานิพนธ์นี้มีพารามิเตอร์ไม่มากและปัญหาไม่ได้ซับซ้อนจึงเลือกใช้เทคนิคนี้เข้ามาใช้ในการแก้ปัญหา โดยขั้นตอนการทำงานของการค้นหาแบบสุ่ม มีขั้นตอนดังนี้

Start

1. Set $best_solution = initial_bad_value$.
2. Set $i = 0$.
3. Repeat the following steps until a stopping criterion is met:
 - a. Generate a new candidate solution x_{new} randomly.
 - b. Check constraint
 - c. Evaluate the objective function $f(x_{new})$ at x_{new} .
 - d. If $f(x_{new})$ is better than $f(best_solution)$, set $best_solution = x_{new}$.
 - e. Increment i by 1.
 - f. If the maximum number of iterations has been reached, exit the loop.
4. Return $best_solution$.

Stop

การกำหนดรูปแบบการหาค่าที่เหมาะสม

1. การโปรแกรมเชิงเส้น (Linear Programming)

การโปรแกรมเชิงเส้น เป็นเทคนิคการทำหาค่าที่เหมาะสม สำหรับปัญหาที่มี Objective Function และ Constraint เป็นฟังก์ชันเชิงเส้นโดยอยู่ในรูปของอสมการ และหาค่า Optimal สำหรับตัวแปรการตัดสินใจได้ เป็นเทคนิคที่อาศัยวิธีการทางคณิตศาสตร์ในการแก้ปัญหาการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัด และใช้งานอย่างกว้างขวางในโมเดลประเภท Constrained Optimization (Bisschop, 2006) ปัญหาบางปัญหาที่เป็นโปรแกรมไม่เชิงเส้น สามารถทำ linearization คือการแปลงให้อยู่ในรูปแบบเชิงเส้นได้ เพื่อใช้วิธีการของ linear programming ในการหาค่าตอบที่เป็น nonlinear ได้

การหาค่าตอบสำหรับการโปรแกรมเชิงเส้น สามารถแก้ปัญหาได้ 2 วิธี คือ

1. วิธีการกราฟ (Graphical Method) เหมาะกับปัญหาที่ต้องตัดสินใจที่มีตัวแปร 2 ตัวแปร
2. วิธีซิมเพล็กซ์ (Simplex Method) เหมาะกับปัญหาที่ต้องตัดสินใจที่มีตัวแปรตั้งแต่ 2 ตัวแปรขึ้นไป โดยตัวแปรทุกตัวต้องมีค่ามากกว่า หรือเท่ากับศูนย์เสมอ

ในงานวิจัยของ Ibrahim et al. (2018) สร้างโมเดล Integer Linear Programming (ILP) สำหรับการใช้พลังงานที่น้อยที่สุดของโครงสร้างศูนย์รวมข้อมูลของคลาวด์ Rankothge et al. (2017) ออกแบบโครงร่างตำแหน่ง VNF (Virtual Network Function) เพื่อกำหนดตำแหน่งเริ่มต้นของ VNF และมาตราส่วน VNF ที่เกิดจากความผันผวนของทราฟฟิก สำหรับตำแหน่งเริ่มต้นของ VNF โดยลดจำนวนเซิร์ฟเวอร์และลิงก์ที่จำเป็นให้เหลือน้อยที่สุด เพื่อลดการใช้ทรัพยากรของผู้ให้บริการระบบคลาวด์ สำหรับการปรับขนาด VNF ได้ทำการเปลี่ยนทรัพยากรแบบไดนามิกเพื่อให้สอดคล้องกับความผันผวนของการรับส่งข้อมูลในขณะที่ลดจำนวนการเปลี่ยนแปลงการกำหนดค่าเพื่อลดการหยุดชะงักของการบริการ โดยใช้โปรแกรมเชิงเส้นจำนวนเต็มและอัลกอริทึมเชิงพันธุกรรม Hu et al. (2018) ตรวจสอบปัญหาของการจัดสรรทรัพยากรบนคลาวด์ที่แตกต่างกันเพื่อเพิ่มประสิทธิภาพการปฏิบัติการบนคลาวด์ วิธีการนี้ได้รับประโยชน์อย่างมากโดยการจัดลำดับความสำคัญของงานหลักในเวลาที่กำหนด แทนที่จะเป็นงานที่ไม่ไวต่อการทำงานให้เสร็จ โดยเฉพาะอย่างยิ่งปัญหาการจัดสรรทรัพยากรถูกกำหนดให้เป็นปัญหาการโปรแกรมเลขจำนวนเต็มก่อน ต่อจากนั้น ด้วยการใช้โครงสร้าง singlemode ของพื้นที่การแก้ปัญหา ปัญหาการโปรแกรมเลขจำนวนเต็มจะถูกสร้างขึ้นใหม่ให้เป็นปัญหาการโปรแกรมเชิงเส้น ซึ่งสามารถแก้ไขได้อย่างมีประสิทธิภาพและเหมาะสมที่สุด โดยเพิ่มยูทิลิตี้กรณี Worst Case ให้สูงสุดและปรับปรุงยูทิลิตี้กรณี Worst Case Preethi et al. (2015) ใช้การโปรแกรมเชิงเส้นในการแก้ปัญหาของการจัดลำดับและการเลือกการให้บริการคลาวด์แบบไดนามิก โดยพิจารณาข้อมูลเชิงปริมาณจากความเร็วของหน่วยประมวล ค่าใช้จ่ายต่าง ๆ เป็นต้น และข้อมูลเชิงคุณภาพได้แก่ช่วงของความต้องการการให้บริการคลาวด์ที่หลากหลายตามความต้องการของผู้ใช้บริการ

2. การโปรแกรมไม่เชิงเส้น (Nonlinear Programming)

การโปรแกรมไม่เชิงเส้น เป็นวิธีการในการแก้ปัญหาที่เหมาะสม เป็นโปรแกรมเชิงคณิตศาสตร์ที่ใช้ในการหาคำตอบ โดยลักษณะของปัญหามักอยู่ในรูปแบบไม่เชิงเส้น ในกรณีที่ตัวแปรการตัดสินใจนั้นมีความสัมพันธ์ในลักษณะที่ไม่เชิงเส้น ภายใต้ข้อจำกัด สำหรับในบางกรณีข้อจำกัดอาจอยู่ในรูปของการโปรแกรมเชิงเส้น (Bazaraa et al., 2013) การหาคำตอบในกรณีที่โปรแกรมไม่เป็นเส้นตรงนั้น จะมีความยุ่งยากกว่าโปรแกรมเชิงเส้น โดยที่ไม่มีอัลกอริทึมใดโดยเฉพาะที่จะใช้ในการหาคำตอบได้ แต่ต้องใช้วิธีการปรับปรุงอัลกอริทึมเพื่อให้ได้คำตอบ การโปรแกรมแบบไม่เชิงเส้นนั้นประกอบไปด้วยฟังก์ชันจุดประสงค์เชิงพีชคณิต มีข้อจำกัดที่เป็นข้อจำกัดเกี่ยวกับพีชคณิตซึ่งประกอบไปด้วยสมการและ/หรืออสมการ (Bisschop, 2006)

ในงานวิจัยของ Lin et al. (2014) ได้จำลองการจัดตารางงานที่ประหยัดต้นทุนด้วยวิธีการเขียนโปรแกรมแบบไม่เชิงเส้นเพื่อเพิ่มประสิทธิภาพกระบวนการจัดสรรงานโดยใช้อัลกอริทึมฮิวริสติกในระบบคลาวด์ Du et al. (2021) ได้ทำการสร้างสูตรของปัญหาการหาค่าที่เหมาะสมด้วย

รูปแบบของ การโปรแกรมไม่เชิงเส้น ที่ประกอบไปด้วยข้อจำกัดแบบเชิงเส้นและแบบไม่เชิงเส้น แล้วนำมาสร้างโปรโตคอลในด้านความปลอดภัยจากภายนอกไปยังระบบคลาวด์ โดยมีการเสนอแผนการเปลี่ยนแปลงที่ปลอดภัยเพื่อปกป้องข้อมูลส่วนตัวของลูกค้า Razaque et al. (2016) ได้นำเสนออัลกอริธึมในการจัดตารางการทำงานที่มีประสิทธิภาพ โดยการพิจารณาจากแบนด์วิธของเครือข่ายสำหรับอัลกอริธึมที่ใช้ในการจัดตารางการทำงานนั้นโดยใช้โมเดลการโปรแกรมไม่เชิงเส้น ซึ่งกำหนดจำนวนงานที่ถูกต้องให้กับเครื่อง VM ตามการจัดสรร Li et al. (2019) ใช้กลยุทธ์ในการแลกเปลี่ยนที่ตระหนักถึงการใช้พลังงานที่เหมาะสมด้วยเกณฑ์การหน่วงเวลา โดยใช้ Three-Layers Fog-Cloud Cooperation ในอธิบายพลังงานและฟังก์ชันการหน่วงเวลา และใช้เทคนิคการโปรแกรมไม่เชิงเส้นมาใช้ในการแก้ปัญหาพลังงานที่เหมาะสมโดยการคำนวณหาค่าที่เหมาะสมของการทำงานแต่ละเลเยอร์

3. การโปรแกรมเลขจำนวนเต็ม (Integer Programming)

การโปรแกรมเลขจำนวนเต็ม เป็นเทคนิคทางคณิตศาสตร์ที่ใช้ในการแก้ปัญหาการวางแผนและการตัดสินใจที่ต้องมีการกำหนดตัวแปรให้เป็นจำนวนเต็มเท่านั้น ในการแก้ปัญหาแบบการโปรแกรมเลขจำนวนเต็ม จะมีการกำหนดข้อจำกัดและฟังก์ชันวัตถุประสงค์ที่ต้องทำให้เข้ากับข้อมูลและเงื่อนไขของปัญหาที่กำหนด โดยโปรแกรมเลขจำนวนเต็มนั้นเป็นโปรแกรมเชิงคณิตศาสตร์ที่มีข้อจำกัดของตัวแปร จะต้องเป็นเลขจำนวนเต็มที่ไม่มีค่าเป็นลบ

ในงานวิจัยของ Meng et al. (2010) ได้ใช้ Integer Linear Programming (ILP) แบบดั้งเดิมถูกนำมาใช้ในการจัดสรรทรัพยากรบนคลาวด์เพื่อเพิ่มประสิทธิภาพการจัดสรร VM และการจัดการเครือข่าย Chase et al. (2014) แก้ปัญหาการโปรแกรมเลขจำนวนเต็มแบบสุ่มเพื่อให้ได้การจัดเตรียมที่เหมาะสมที่สุดสำหรับทั้งเครื่องเสมือนและแบนด์วิธของเครือข่าย เมื่อความต้องการของลูกค้าไม่แน่นอน โดยผลลัพธ์ที่เป็นตัวเลขแสดงให้เห็นอย่างชัดเจนว่าวิธีการนี้ช่วยลดค่าใช้จ่ายของผู้ใช้และให้ประสิทธิภาพที่เหนือกว่าวิธีการอื่น ๆ

เครื่องมือในการวัดประสิทธิภาพ

Sysstat เป็นเครื่องมือตรวจสอบและบันทึกข้อมูลที่มีประสิทธิภาพสูงสำหรับระบบ Linux/Unix สามารถใช้สำหรับการตรวจสอบประสิทธิภาพของระบบและระบุและแก้ไขปัญหา ภายในแพ็คเกจ Sysstat มีคำสั่งต่างๆ สำหรับระบบ Linux เช่น sar, mpstat และ iostat (Raju & Govindarajulu, 2014) แพ็คเกจนี้มีเครื่องมือที่หลากหลายเพื่อดึงข้อมูลเกี่ยวกับการใช้ CPU หน่วยความจำ เครือข่าย และฮาร์ดดิสก์ได้อย่างง่ายดาย (Vasar, 2012)

sar หรือ System Activity Reported เป็นโปรแกรมรายงานกิจกรรมของระบบหรือโปรแกรมอรรถประโยชน์ระบบ sar มีตัวเลือกมากมายในการประเมินประสิทธิภาพของระบบ Linux โดยให้ข้อมูลที่ครอบคลุมเกี่ยวกับประสิทธิภาพในด้านต่าง ๆ เช่น system load, CPU usage,

memory usage, paging, swapping, disk usage, device load, network activity และอื่น ๆ (Henry, 2021) สำหรับการวัดปริมาณการใช้งาน CPU (CPU usage) คำสั่ง sar จัดเตรียมสองแอมป์ที่ทริบิวต์ที่สามารถใช้เพื่อคำนวณการใช้งาน CPU ของระบบ แอมป์ที่ทริบิวต์แรก %user แสดงถึงเปอร์เซ็นต์ของเวลาที่ CPU ใช้ในการประมวลผลของผู้ใช้ ในขณะที่แอมป์ที่ทริบิวต์ที่สอง %system แสดงถึงเปอร์เซ็นต์ของเวลาที่ CPU ใช้ในการเรียกใช้งานเคอร์เนล การวัดปริมาณการใช้งาน CPU จึงเป็นการนำทั้ง 2 แอมป์ที่ทริบิวต์มาบวกกัน (Basheer, 2016) และนอกจากนี้คำสั่ง sar ยังแสดงในส่วนของการวัดปริมาณในการใช้งานของ I/O wait, nice, steal และ idle อีกด้วย

CPU usage หมายถึงสัดส่วนของความจุของโปรเซสเซอร์ที่กำลังใช้งานอยู่ และค่านี้อาจผันผวนได้ขึ้นอยู่กับประเภทของงานที่ดำเนินการโดยโปรเซสเซอร์

Memory usage คือเปอร์เซ็นต์ของการใช้หน่วยความจำจากหน่วยความจำทั้งหมดที่จัดสรรให้กับเครื่องเสมือน ประกอบด้วยหน่วยความจำแคชและหน่วยความจำบัฟเฟอร์ในการคำนวณเมื่อทำการวัดการใช้หน่วยความจำ

ในงานวิจัยต่าง ๆ ได้มีการนำเครื่องมือ Sysstat มาใช้ในการวัดประสิทธิภาพการทำงานของคอมพิวเตอร์ที่ติดตั้งระบบ linux เช่นงานวิจัยของ Rizvandi (2012) ได้มีการใช้คำสั่งจากเครื่องมือ Sysstat ในการหาค่าการใช้งานของ CPU ในขณะที่ MapReduce application ทำงาน งานวิจัยของ Berggren & Karlsson (2022) ใช้ เครื่องมือ Sysstat ในการวัดปริมาณการใช้งาน CPU และ RAM ในทุก ๆ วินาทีระหว่างที่มีการทำการทดสอบประสิทธิภาพการทำงานของ containerization และ virtualization งานวิจัยของ Mishra et al. (2022) ใช้เครื่องมือ Sysstat ในการเก็บรวบรวมข้อมูลจากการทำงานของ linux มาใช้ในการวิเคราะห์การทำงานที่คล้ายกันด้วย Machine learning งานวิจัยของ Samara & Abandah (2021) ได้ใช้เครื่องมือ Sysstat ในการวัดการใช้ประโยชน์จาก CPU ในการวัดการทำงานของโมเดล เป็นต้น

งานวิจัยที่เกี่ยวข้อง

โดยสรุปงานวิทยานิพนธ์นี้ สามารถสรุปรายละเอียดของงานวิจัยที่เกี่ยวข้องได้ดังนี้

จากปริมาณการใช้งานของ IaaS คลาวด์แบบสาธารณะที่มีเพิ่มมากขึ้นอย่างต่อเนื่อง ดังที่กล่าวไปแล้วในหัวข้อการประมวลผลแบบกลุ่มเมฆนั้น ผู้ให้บริการคลาวด์ได้แก่ Amazon, Microsoft และ Google ต่างพยายามที่จะเพิ่มประสิทธิภาพและการรักษาความปลอดภัยให้กับผู้ใช้บริการ ความปลอดภัยถือว่าเป็นปัญหาที่สำคัญอย่างยิ่งที่ทำให้ผู้ใช้บริการเกิดความกังวลใจ เนื่องจากการโจรกรรมข้อมูลโดยผู้ที่ไม่ประสงค์ดีในการเข้าถึงไฟล์ข้อมูลของผู้ใช้บริการ ในปี 2022 Cyber Security Insiders (isc2, 2022) ได้มีการรายงานเกี่ยวกับความปลอดภัยบนคลาวด์ในรายงานที่ชื่อว่า 2022 Cloud Security Report ซึ่งได้รายงานถึง Biggest Security Threats ในประเด็น

unauthorized access (การไม่อนุญาตในการเข้าถึง) และ Cloud Security Alliance (CSA, 2022) ได้รายงานเกี่ยวกับภัยคุกคามที่เกิดขึ้นกับคลาวด์ในรายงานที่ชื่อว่า Top Threats to Cloud Computing : The Pandemic 11 ในประเด็นในเรื่องของ การจัดการการเข้าถึง และเมื่อมีการเข้าถึงข้อมูลในเครื่องของผู้ใช้บริการแล้วจะมีการตรวจสอบได้อย่างไร ในหลายงานวิจัยได้มีการใช้ระบบบันทึกเหตุการณ์บน IaaS คลาวด์ในชื่อเรียกที่แตกต่างกันไป เช่น Nitro ที่ใช้ฮาร์ดแวร์เป็นเครื่องมือในการสนับสนุนการทำงาน Virtual Machine Introspection (VMI) (Pfoh, Schneider & Eckert, 2011), Progger (Ko & Will, 2014), VMIGuard (Fargo et al., 2020) เป็นต้น ที่ใช้ในการดึงข้อมูลจาก VM มาใช้ในการตรวจสอบหาความผิดปกติของการทำงาน

VMI เป็นการนำข้อมูลการทำงานจากหน่วยความจำแรมของเครื่องเสมือนมาพิจารณาการถูกโจมตีในการเข้าถึงไฟล์ข้อมูลซึ่งเป็นเทคนิคในรูปแบบหนึ่งของ intrusion detection system (IDS) โดยมีการใช้ VMI ในการเข้าถึง Memory ของเครื่อง VMs ได้ โดยใช้ไลบรารี LibVMI ในการดึงข้อมูลจาก VMs ซึ่งไลบรารี LibVMI เป็นการทำงานหลักของวิธีการของ VMI (Borisaniya & Patel, 2019)

การทำงานของซอฟต์แวร์ในการเข้าถึงข้อมูล Logs นั้น ในงานวิจัยของ Ko & Will (2014) ที่ต้องมีการใช้งานซอฟต์แวร์ในการดึงข้อมูล 1 ตัว ต่อ Virtual Machine 1 เครื่อง งานวิจัยของ Auxson (2020), Chan-In (2017), Jaiboo (2020), และ Wongthai (2017) ได้มีการใช้ระบบบันทึกเหตุการณ์ ซึ่งเป็นวิธีการหนึ่งที่สำคัญสำหรับช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามขึ้นต่อคลาวด์ โดยภายในระบบบันทึกเหตุการณ์ได้จัดเก็บหลักฐาน Log File ไว้เป็นหลักฐานและหลักฐานจาก Log File สามารถนำไปช่วยหาบุคคลที่กระทำผิดมารับผิดชอบได้โดยมีการทำงานแบบ ล็อกเกอร์ 1 ตัวเข้าไปตรวจจับโปรเซสการทำงานใน VM 1 เครื่อง รวมไปถึงงานวิจัยของ Noor et al. (2019) ที่ดึงข้อมูลจาก Virtual Machine หลายเครื่องมาเก็บไว้ในฐานข้อมูลโดยผ่าน Monitoring Manager งานวิจัยของ Borisaniya & Patel (2019) ที่ได้ใช้งาน Nitro ที่ติดตั้งไว้บน Hypervisor ในการดึงข้อมูลของโปรเซสที่กำลังทำงานภายในของ VMs โดยใช้หลักการของการทำ Multi-Thread ในการดึงข้อมูลโปรเซสจาก VM หลายเครื่อง งานวิจัยของ Shamseddine et al. (2022) ที่ใช้วิธีการแบบ divide-and-conquer มาใช้ในการตรวจสอบการทำงานของ VM และในการวัดประสิทธิภาพโดยงานวิจัยของ Patrascu & Patriciu (2015), Yang, Wang, & Huang (2017), Rane, Wagh, & Dixit (2021), Ko & Will (2014) ได้มีการวัดประสิทธิภาพในการเข้าถึงข้อมูลด้วยระยะเวลาในการทำงานเพียงด้านเดียว โดยไม่ได้มองไปถึงทรัพยากรที่ใช้ในการทำงานว่าสามารถทำงานได้ครอบคลุมทุก VM หรือไม่ และในงานวิจัยของ Csáti, Pelle, & Toka (2022) การวัดประสิทธิภาพของการเก็บข้อมูลบริการโดย Elastic Stack ซึ่งเป็น Logging Framework ทำการทดลองการใช้งาน CPU และ

RAM ตามขนาดไฟล์ และงานวิจัยของ Noor et al. (2019) เปรียบเทียบการใช้งานของ CPU, RAM และ Network ของผู้ให้บริการคลาวด์ (Azure และ Amazon)

การจัดการทรัพยากรนั้นเป็นประเด็นหลักในการประมวลผลแบบคลาวด์ เช่นเดียวกับการประมวลผลรูปแบบอื่น ๆ เนื่องจากความพร้อมใช้งานของทรัพยากรที่จำกัด ผู้ให้บริการคลาวด์จึงเห็นเป็นเรื่องที่ยากมากที่จะจัดหาทรัพยากรตามที่คุณใช้บริการร้องขอทั้งหมด (Parikh, Patel, & Prajapati, 2017) โดยการจัดการทรัพยากรเป็นปัญหาที่สำคัญที่สุดในโครงสร้างพื้นฐานระบบคลาวด์ เนื่องจากมีวัตถุประสงค์ที่ขัดแย้งกันหลายประการไม่ว่าจะเป็นการรักษาประสิทธิภาพ การลดพลังงาน หรือการลดต้นทุน เป็นต้น ซึ่งจำเป็นต้องมีการปรับให้เหมาะสมในการจัดการทรัพยากร ในการจัดการทรัพยากรบนคลาวด์การรวม VMs และการจัดสรรทรัพยากรนั้นใช้ในการแก้ปัญหาความผันผวนของปริมาณงาน โดยการย้าย VM จากโฮสต์หนึ่งไปยังอีกโฮสต์หนึ่งเพื่อลดจำนวนโฮสต์ที่ใช้งานและเพื่อเป็นการประหยัดพลังงาน (Aldossary, 2021) ในงานวิจัยของ Ye et al. (2011) ได้นำเสนอเฟรมเวิร์กในการย้าย VM หลายตัวตามการจองทรัพยากรที่แตกต่างกัน เพื่อให้มีประสิทธิภาพในการย้ายข้อมูลแบบขนานและการรับรู้ภาระงาน งานวิจัยของ Zhao et al. (2015) ได้นำเสนอวิธีการจัดวาง VM ตามประสิทธิภาพของการบริการ โดยนำลักษณะการใช้ทรัพยากรมาใช้ในการพิจารณาในการดูว่า VM จะไปอยู่ที่โฮสต์ใดที่จะให้เกิดความเหมาะสมเพื่อรับประกันประสิทธิภาพของ VM และรับประกันคุณภาพประสบการณ์ของลูกค้า (QoE) งานวิจัยของ Malekloo et al. (2018) ได้นำเสนอแนวทาง Multi-objective Ant Colony Optimization (MACO) ในการรวบรวมและจัดวาง VM โดยมีเป้าหมายเพื่อลดการใช้พลังงาน การสิ้นเปลืองทรัพยากรหน่วยประมวลผล และต้นทุนการสื่อสาร มีเป้าหมายเพื่อลดการละเมิดข้อตกลงการให้บริการระหว่างผู้ให้บริการและผู้ให้บริการ (Service Level Agreement: SLA) การย้าย VM และจำนวนโฮสต์ที่ใช้งานอยู่ สำหรับการใช้งานทรัพยากรของคอมพิวเตอร์นั้น เมื่อพบว่าปริมาณการใช้งานของ CPU มีระดับสูง 80% ขึ้นไป เป็นระยะเวลาานาน ประสิทธิภาพของระบบหรือแอปพลิเคชันจะได้รับผลกระทบด้วย (Microsoft, 2022) ในการทำงาน of IaaS คลาวด์นั้นการจัดสรรทรัพยากรเป็นสิ่งจำเป็นอย่างยิ่งเพื่อไม่ให้งานของ VM เกิดปัญหาในการทำงาน

กระบวนการหาค่าที่เหมาะสม เป็นสิ่งสำคัญในการสร้างระบบให้มีประสิทธิภาพ (Zeng, 2023) และเป็นขั้นตอนวิธีในการตัดสินใจเลือกผลลัพธ์ที่ดีที่สุดในการแก้ปัญหาที่เหมาะสมในงานวิจัยทางด้านคลาวด์ ได้มีการนำเทคนิคการทำ optimization มาใช้ในการทำงานให้มีประสิทธิภาพ เช่นในงานวิจัยของ chaisiri (2011) ที่ทำการหาวิธีการที่เหมาะสมในการจัดการด้านต้นทุนในการจัดหาทรัพยากร Ala'a Al-Shaikh (2016) ได้นำวิธีการในการหาวิธีการที่เหมาะสมมาใช้ในการแก้ปัญหาเรื่องของการใช้ทรัพยากรบนระบบคลาวด์ได้ Wei (2020) ได้ทำการแก้ปัญหาเรื่องของการโหลดที่ไม่สมดุล ความเร็วในการรวมกันซ้ำ หรือใช้ทรัพยากรของ VM ที่ต่ำ ด้วยการใช้กลยุทธ์การหาค่าที่

เหมาะสม เป็นต้น ซึ่งจะเห็นได้ว่าการทำการหาค่าที่เหมาะสมบนคลาวด์นั้นจะเป็นการทำในส่วนของ
การจัดสรรทรัพยากร เช่น ซีพียู แรม ฮาร์ดดิสก์ เป็นต้น แต่ยังไม่มีการดำเนินการทำในส่วนของ
หาวิธีในการจัดสรรระบบบันทึกเหตุการณ์



ตาราง 5 แสดงผลสรุปงานวิจัยที่เกี่ยวข้อง

งานวิจัยโดย	System-Centric Log/ Files Centric Log	การวัดประสิทธิภาพ ทรัพยากร	รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
Patrascu & Patriciu (2015)	System-Centric Log	CPU x RAM x	เปรียบเทียบระยะเวลาในการดึงข้อมูลมาเก็บไว้ใน Storage Module โดยทดลองในการกำหนดขนาดของ RAM ออกเป็น 2 กรณีคือ RAM ที่มีขนาดไม่เกิน 100 MB และ RAM ที่มีขนาดมากกว่า 100 MB	ทำการวัดเพียงระยะเวลาในการถ่ายโอนข้อมูลระหว่างส่วนตรวจจับความผิดปกติไปยังหน่วยจัดเก็บข้อมูล โดยแบ่งการทดสอบตามปริมาณ RAM ที่แบ่งออกเป็น 100 MB และมากกว่า 100 MB

ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/ Files Centric Log	การวัดประสิทธิภาพ ทรัพยากร	รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
		CPU RAM		
Csáti, Pelle, & Toka (2022)	System-Centric Log	✓ ✓	ผลการทดลองการใช้งาน CPU และ RAM ตามขนาดไฟล์ของผู้ใช้งาน พบว่า การทำงานด้วย Filebeat ที่ Logs ขนาด 0.5-5 kB, Elasticsearch ใช้ CPU เฉลี่ย 91% RAM 1.8 GiB และสามารถจัดการการไหลได้โดยเครื่องคอมพิวเตอร์ไม่มีปัญหา และการทำงานโดยส่งข้อมูลผ่าน HTTP ที่ Logs ขนาด 10-64 kB, Elasticsearch ใช้ CPU 100% RAM 1.7 GiB	สนใจเพียงปริมาณการเก็บ ข้อมูล log file
Yang, Wang, & Huang (2017)	System-Centric Log	✗	นำเสนอวิธีการทำงานของ Cloud Data Accountability Framework ในการรับประกันการบันทึกข้อมูลที่เชื่อถือได้ โดยการใช้ประโยชน์จากไฟล์ Java JAR ในการป้องกันการเข้าถึง ที่มีการทำงานแบบ	มีการจำกัดจำนวนการดำเนินการเข้ารหัสในกระบวนการบันทึกเพื่อลดขนาดการรับรองและลดเวลาในการสร้างการรับรอง

ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/ Files Centric Log	การวัดประสิทธิภาพ ทรัพยากร	รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
Hickman et al. (2018)	System-Centric Log	CPU RAM	Client-Server การวัดประสิทธิภาพของ Server-Client JARs ได้ทำการประเมินประสิทธิภาพในสี่ด้าน: การจัดเก็บ การอนุญาต การบันทึก และการถอดรหัส แสดงให้เห็นว่า Cloud Data Accountability Framework มีประสิทธิภาพในการใช้พื้นที่จัดเก็บและรองรับการเข้าถึงข้อมูลที่รวดเร็ว โดยการให้สิทธิ์และการบันทึกอย่างรวดเร็ว	เป็นการสาธิตเครื่องมือต้นแบบในการสร้างระบบเชี่ยวชาญ
			ในงานวิจัยนี้จึงได้ทดลองและปรับปรุงระบบ Logan ของ Los Alamos National Laboratory (LANL) ให้สามารถระบุได้ว่าซอฟต์แวร์ตัวไหนที่สร้างข้อความ syslog นั้น โดยระบบ Logan ใหม่ จะสามารถโต้ตอบกับผู้ใช้และระบบพร้อมกันโดยง่าย	

ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/ Files Centric Log	การวัดประสิทธิภาพ ทรัพยากร	รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
		CPU RAM	ผลลัพธ์ได้ เช่น ข้อมูลที่ตรวจพบเป็นข้อมูลเกี่ยวกับเคอร์เนล ข้อมูลเกี่ยวกับเครือข่าย เป็นต้น และเป็นเป้าหมายสุดท้ายคือ การที่รู้ถึงกระบวนการในการสร้างข้อความใน syslog ที่ผิดปกติ โดยเชื่อมโยงข้อความกับรหัสที่สร้างข้อความนั้น เมื่อปรับปรุงเสร็จแล้วผู้วิจัยได้มีการทดลองและวัดประสิทธิภาพการทำงาน พบว่า การทดสอบการตรวจจับบรรทัดของซอร์สโค้ดจะจับคู่กับ log message มีความเชื่อมั่นอยู่ที่ 92% และสำหรับการทดสอบการคาดการณ์ข้อมูลใหม่ นั้นพบว่ามีความเชื่อมั่นอยู่ที่ 88.2% - 95.8% ของเวลาที่ใช้ทั้งหมด	

ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/ Files Centric Log	การวัดประสิทธิภาพ ทรัพยากร	รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
Rane, Wagh, & Dixit (2021)		CPU RAM ✗ ✗	เป็นงานวิจัยที่มุ่งเน้นในการปกป้องข้อมูล บันทึกกิจกรรมของผู้ใช้ระบบคลาวด์ในการ พัฒนาการให้บริการการบันทึกข้อมูลที่ ผสมผสานระหว่างบล็อกเชน (BlockSLaaS) กับ IPFS ผลการวัดประสิทธิภาพเวลา ตอบสนอง เวลาในการตรวจสอบข้อมูล นำเข้า และเวลาในการพิสูจน์ยืนยันตัวตน ของ BlockSLaaS คือ 38.3, 29.7 และ 26.3 มิลลิวินาทีตามลำดับ ซึ่งมี ประสิทธิภาพดีกว่าวิธีการที่มีอยู่	ต้องเข้ารหัสในระบบคลาวด์ เพื่อยืนยันตัวตน ทำให้ เสียเวลา
Ko & Will (2014)	File-Centric Log	✗	เก็บข้อมูล log file บน Virtual disk หรือ physical host โหลดข้อมูลโดยใช้ system call ติดตั้ง Progger ในทุก VM สามารถ เรียงลำดับเหตุการณ์ในการเข้าถึงไฟล์ได้ วัต เป็นการทำงานแบบ 1	ไม่มีไฟล์บันทึกที่จัดเก็บไว้ ในดีสก์หลัก หรือมีการ ส่งผ่านไปยังโฮสต์จริง และ

ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/		การวัดประสิทธิภาพ		รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
	Files Centric Log	ทรัพยากร	CPU	RAM		
Fargo et al. (2020)	File-Centric Log	x	x		ประสิทธิภาพในเรื่องเวลาในการดึงข้อมูลโดยเวลาที่ใช้ขึ้นอยู่กับประสิทธิภาพของ VM ด้วย	ไม่มีการวัดประสิทธิภาพการทำงานของตัวระบบเพียงแต่ระบุว่าอนุญาตให้โปรเซสส์นั้นทำงานหรือไม่ทำงาน ถ้าไม่อนุญาตจะทำการปิด VM โดยอัตโนมัติ

งานวิจัยนี้ใช้คลาวด์ส่วนตัว (private cloud) ที่ใช้ไฮเปอร์เวเซอร์ Xen และสร้าง VM หลายตัว ใช้ LibVMI ในการตรวจสอบข้อมูลจาก VMs หลาย ๆ ตัว มีการสร้าง allowlisting ที่จะสามารถระบุชื่อของแอปพลิเคชันและเข้าถึงที่อยู่ในหน่วยความจำของ VMs ได้ หากมี VM ใดที่ได้รับการบุกรุกและดำเนินการระบุว่าเป็นอันตรายเราจะสามารถระบุได้ว่าเป็นการทำงานของ malicious โดยดูจากกระบวนการทำงานของ malicious จะมีความแตกต่างจากการ

ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/		การวัดประสิทธิภาพ		รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
	System-Centric Log	Files Centric Log	ทรัพยากร	ทรัพยากร		
			CPU	RAM		
Noor et al. (2019)	System-Centric Log		✓	✓	ทำงานปกติและการใช้งานทรัพยากร เช่น Bot-Attacks, Cryptocurrency Mining, Ransomware, DDoS	ยังไม่มีการตรวจสอบอุปกรณ์ทางกายภาพและการโยกย้ายคอนเทนเนอร์
					เปรียบเทียบการใช้งานของ CPU, RAM และ Network ของผู้ให้บริการคลาวด์ (Azure และ Amazon) ที่มีผู้ให้บริการหลายเจ้า และเครื่องเสมือนหลายเครื่อง	แอปพลิเคชันเพื่อพัฒนาการยุทธ์การปรับใช้และการประสานที่มีประสิทธิภาพสำหรับแอปพลิเคชันไชนเซอร์และกายภาพ
Borisaniya & Patel (2019)	File-Centric Log		✗	✗	ที่ได้ใช้งาน Nitro ในการดึงข้อมูลของโปรเซสที่กำลังทำงานภายในของ VM โดยใช้หลักการของการทำ Multi-thread ใน	ไม่มีการวัดประสิทธิภาพการทำงานของระบบตรวจสอบการตั้งข้อมูลใน

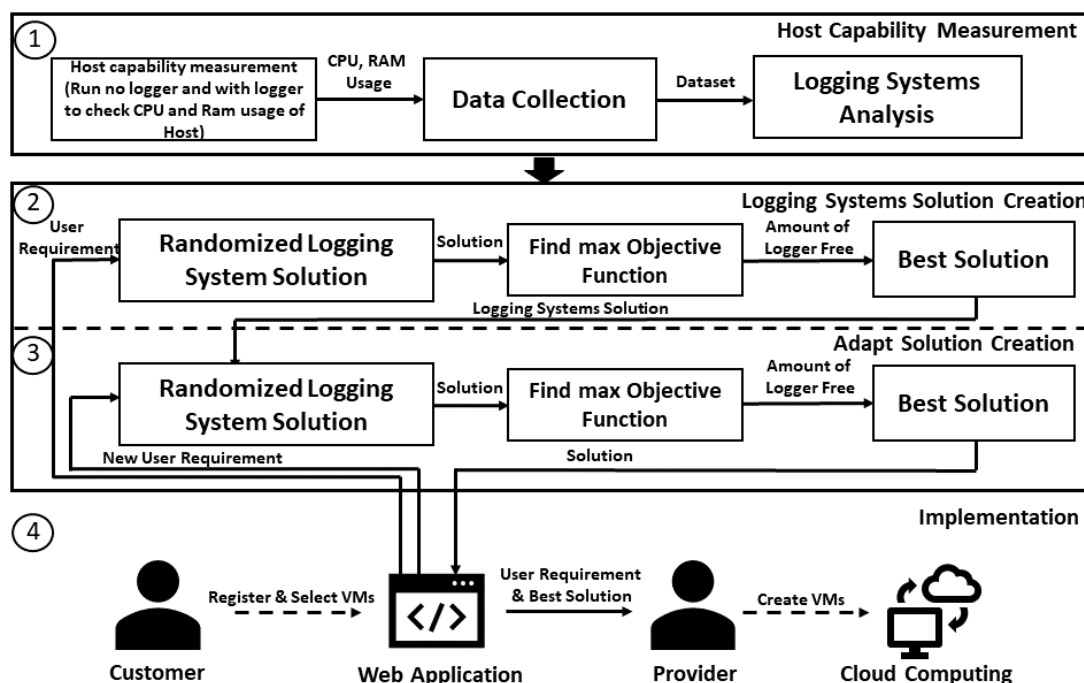
ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/ Files Centric Log	การวัดประสิทธิภาพ ทรัพยากร	รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
		CPU RAM		
Auxsorn et al. (2020)	File-Centric Log	✗ ✗	การดึงข้อมูลไปตรวจสอบจาก VMs หลายเครื่อง มาเก็บไว้ในตัว Analyzer เพื่อทำการวิเคราะห์ว่าเป็น Benign หรือ Malicious วัดค่าความถูกต้องของการจำแนกประเภท วัดค่าเฉลี่ยของระยะเวลาในการตอบสนอง	ด้านประสิทธิภาพของผู้ให้บริการ และผู้ใช้บริการ
			ทำการวัดประสิทธิภาพของระบบบันทึก เหตุการณ์บนฝั่งของผู้ให้บริการในการตรวจจับการเข้าถึงไฟล์ใน VM โดยพบว่ามี การกำหนดในส่วนของ Sleeping Time ให้กับ Logger มากจะทำให้ความแม่นยำในการตรวจจับข้อมูลไฟล์นั้นลดลง แต่พบว่า ประสิทธิภาพการทำงานของการทำงานฝั่ง ผู้ใช้บริการนั้นมีเพิ่มมากขึ้น	ทำงานแบบตรวจจับ Logger 1 ตัวต่อ 1 VM มีการวัดประสิทธิภาพการทำงานในส่วนของผู้ใช้งาน แต่ไม่มีการวัดประสิทธิภาพในส่วนของผู้ใช้บริการ

ตาราง 5 (ต่อ)

งานวิจัยโดย	System-Centric Log/ Files Centric Log	การวัดประสิทธิภาพ ทรัพยากร	รายละเอียดวิธีการ/เทคนิค	ข้อจำกัดงานวิจัย
Jaiboon et al. (2020)	File-Centric Log	CPU x RAM x	ทำการติดตั้งระบบบันทึกเหตุการณ์บน Openstack ในการทดสอบการใช้งานบนสภาพแวดล้อมจริง ระบบสามารถตรวจจับการเข้าถึงไฟล์ที่เป็นประโยชน์ของผู้ไม่ประสงค์ได้	ในงานวิจัยนี้ไม่ได้มีการวัดประสิทธิภาพการทำงานของระบบบันทึกเหตุการณ์เพียงแต่เป็นการนำระบบไปใช้ในสภาพแวดล้อมการทำงานคลาวด์บน Open stack
This Dissertation	File-Centric Log	✓	จัดการทรัพยากรของเครื่องคอมพิวเตอร์ในการใช้งาน CPU ให้มีการใช้งานลดลง เพื่อให้สามารถใช้งาน VM ได้หลายตัว ใน การตรวจจับ File-Centric Log โดยวัดประสิทธิภาพการใช้งานของ CPU ในฝั่งผู้ให้บริการ และนำไปสู่การสร้างโมเดลทางคณิตศาสตร์ในการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมที่สุด	

กรอบแนวคิดงานวิจัย



ภาพ 5 แสดงกรอบแนวคิดการวิจัย

จากภาพ 5 สามารถอธิบายขั้นตอนการดำเนินงานวิทยานิพนธ์ได้ดังนี้

หมายเลข 1 เป็นขั้นตอนในการวางรูปแบบการทดสอบความสามารถในการรองรับการทำงานของระบบบันทึกเหตุการณ์ โดยทำการวัดปริมาณการใช้งานของ CPU และ RAM ด้วยคำสั่ง Sysstat ในขณะที่ปล่อยให้ VM ทำงานโดยไม่มีการใช้งานระบบบันทึกเหตุการณ์ โดยเพิ่มจำนวน VM ขึ้นไปเรื่อย ๆ และทำการวัดในกรณีที่มีการทำงานของระบบบันทึกเหตุการณ์ที่ใช้ในการตรวจจับโปรเซสการทำงานในเครื่อง VM โดยการวัดการทำงานเป็นคู่จาก 1 คู่ เพิ่มเป็น 2 คู่ ไปจนถึง n คู่ แล้วนำข้อมูลที่ได้อัวิเคราะห์เพื่อนำไปใช้ในการสร้างโมเดลทางคณิตศาสตร์ต่อไป

หมายเลข 2 คือขั้นตอนในการสร้างโมเดลทางคณิตศาสตร์ในการจัดสรรระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำและประสิทธิภาพการทำงานคอมพิวเตอร์เสมือนบน IaaS คลาวด์ด้วยการกำหนดข้อจำกัด จากระดับความปลอดภัยและระดับประสิทธิภาพด้วยการหาค่าล๊อเกอร์ที่ไม่ได้มีภาระงานมากที่สุด และได้วิธีการจัดสรรระบบบันทึกเหตุการณ์ที่นำไปใช้ในการบริหารจัดการของผู้ให้บริการได้ด้วยการใช้เทคนิค Random Search ในการแสดงให้เห็นว่าโมเดลทางคณิตศาสตร์ที่ได้สร้างขึ้นมานั้นสามารถนำไปใช้งานได้จริง

หมายเลข 3 เป็นขั้นตอนของการปรับโมเดลทางคณิตศาสตร์ที่เมื่อผู้ใช้บริการมีการเปลี่ยนแปลงความต้องการในส่วนจากระดับความปลอดภัยหรือระดับประสิทธิภาพ แล้วนำมาพิจารณาหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ใหม่โดยที่ยังคงให้ผู้ใช้บริการเดิมที่ผลกระทบน้อยที่สุด และยังคงได้ค่าล็อกเกอร์ที่ไม่ได้มีภาระงานมากที่สุด

หมายเลข 4 เป็นส่วนที่ผู้ให้บริการสามารถนำเอาโมเดลทางคณิตศาสตร์จากระบบบันทึกเหตุการณ์ไปประยุกต์ใช้เป็น Web Application เพื่อให้บริการกับผู้ใช้งานต่อไปได้



บทที่ 3

วิธีดำเนินการวิจัย

จากการศึกษาเนื้อหาในบทที่ 2 ในเอกสารและงานวิจัยที่เกี่ยวข้องนั้น ผู้วิจัยได้ทำการสรุปเป้าหมายของวิทยานิพนธ์นี้ คือการสร้างโมเดลทางคณิตศาสตร์เพื่อจัดสรรทรัพยากรแก่ผู้ใช้บริการ และสร้างตัวปรับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ที่เหมาะสมกับผู้ใช้บริการและผู้ให้บริการ โดยเนื้อหาในบทที่ 3 นี้จะถูกนำมาใช้เพื่อเป็นคำตอบการนำไปสู่การแก้ปัญหาหลักในงานวิทยานิพนธ์ โดยงานวิจัยส่วนใหญ่จะเน้นไปที่การจัดสรรทรัพยากรที่เหมาะสมของอุปกรณ์ในการสร้างเครื่องคอมพิวเตอร์เสมือน แต่ไม่ได้ทำในเรื่องของการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมกับผู้ใช้บริการและผู้ให้บริการ

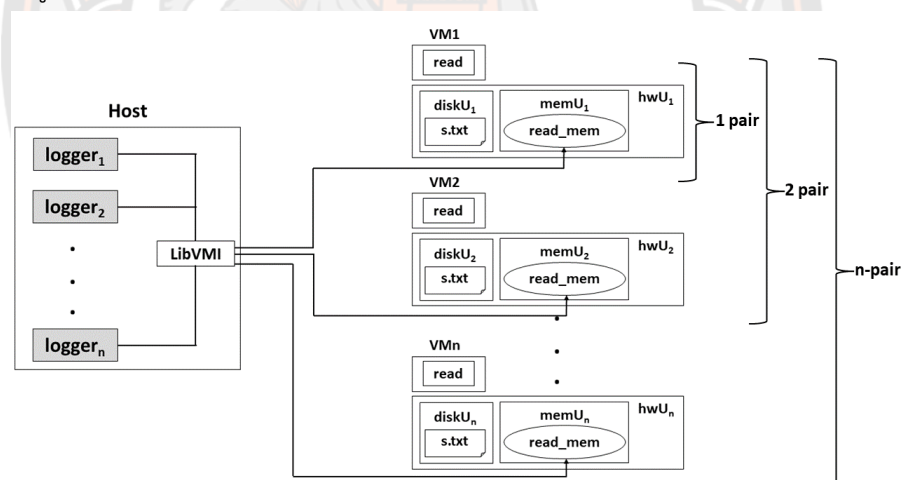
สำหรับการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมตามชื่อเรื่องของวิทยานิพนธ์ ได้แก่โมเดลที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำและประสิทธิภาพการทำงานของคอมพิวเตอร์เสมือนบนการประมวลผลแบบกลุ่มเมฆ ที่นำเสนอในบทที่ 3 ซึ่งจะช่วยให้ผู้อ่านได้เข้าใจถึงวิธีการสร้างรูปแบบทางคณิตศาสตร์ของระบบบันทึกเหตุการณ์ เพื่อนำไปใช้ในการแก้ไขปัญหาของการให้บริการของระบบบันทึกเหตุการณ์บน IaaS คลาวด์ได้ โดยผู้วิจัยได้ดำเนินการทดลองต่าง โดยแบ่งเป็นหัวข้อย่อยของการดำเนินงานดังต่อไปนี้

1. วิธีการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์
2. วิธีการทดสอบปริมาณการใช้งาน CPU และ RAM ของระบบบันทึกเหตุการณ์บนเครื่องคอมพิวเตอร์ของผู้ให้บริการ
3. แนวคิดวิธีการในการจัดสรรระบบบันทึกเหตุการณ์
4. วิธีการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์
5. วิธีการสร้างการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ใช้บริการและผู้ให้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

วิธีการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์

ในหลายงานวิจัยได้มีการใช้ระบบบันทึกเหตุการณ์บน IaaS คลาวด์ในชื่อเรียกที่แตกต่างกันไป เช่น Nitro ที่ใช้ฮาร์ดแวร์เป็นเครื่องมือในการสนับสนุนการทำงานของ Virtual Machine Introspection (VMI) (Pfoh, Schneider & Eckert, 2011), Progger (Ko & Will, 2014), VMIGuard (Fargo et al., 2020) เป็นต้น ที่ใช้ในการดึงข้อมูลจาก VM มาใช้ในการตรวจสอบหาความผิดปกติของการทำงาน การทำงานของซอฟต์แวร์ในการเข้าถึงข้อมูล Logs นั้น ในงานวิจัยของ Ko & Will (2014) ที่ต้องมีการใช้งานซอฟต์แวร์ในการดึงข้อมูล 1 ซอฟต์แวร์ ต่อ Virtual Machine 1 เครื่อง รวมไปถึงงานวิจัยของ Auxorn et al. (2020) ที่ได้มีการทดสอบประสิทธิภาพของระบบบันทึกเหตุการณ์แบบบล็อกเกอร์ 1 ซอฟต์แวร์ต่อ VM 1 เครื่อง และงานวิจัยของ Jaiboon et al. (2020) ที่ได้ทดสอบการทำงานของระบบบันทึกเหตุการณ์แบบบล็อกเกอร์ 1 ซอฟต์แวร์ต่อ VM 1 เครื่อง บนสภาพแวดล้อมของ Openstack

ดังนั้นจะเห็นว่างานวิจัยที่ผ่านมา ยังไม่มีการทำการทดลองการทำงานของระบบบันทึกเหตุการณ์ที่ทำงานมากกว่า 1 คู่ ผู้วิจัยจึงได้ทำการทดลองการทำงานของระบบบันทึกเหตุการณ์ที่มากกว่า 1 คู่ โดยได้ออกแบบการทำงานดังภาพ 6 ซึ่งอ้างอิงการทำงานมาจาก Auxorn (2020)



ภาพ 6 แสดงการออกแบบการทำงานของระบบบันทึกเหตุการณ์แบบ 1 คู่, 2 คู่ และหลายคู่

จากภาพ 6 ในกล่องด้านซ้ายมือส่วนของโฮสต์ (Host) ประกอบไปด้วยบล็อกเกอร์ตัวที่ 1 ถึงตัวที่ n และ LibVMI ส่วนในกล่องทางด้านขวามือจะประกอบไปด้วยส่วนประกอบของ VM ตัวที่ 1 ถึงตัวที่ n โดยส่วนประกอบภายใน VM ได้อธิบายไว้ในบทที่ 2 หัวข้อการทำงานของระบบบันทึกเหตุการณ์บน IaaS คลาวด์ การทำการทดลองนั้นจะทำการทดลองแบบ 1 คู่, 2 คู่ และ n คู่ตามลำดับ โดย 1 คู่คือบล็อกเกอร์ตัวที่ 1 ทำการตรวจจับโปรเซสการทำงานโดยดึงข้อมูลจากหน่วยความจำของ

VM1 2 คู่คือ ล็อกเกอร์ตัวที่ 1 ทำการตรวจจับโปรเซสการทำงานโดยดึงข้อมูลจากหน่วยความจำของ VM1 และล็อกเกอร์ ตัวที่ 2 ทำการตรวจจับโปรเซสการทำงานโดยดึงข้อมูลจากหน่วยความจำของ VM2 และ n คู่คือล็อกเกอร์ตัวที่ 1 ทำการตรวจจับโปรเซสการทำงานโดยดึงข้อมูลจากหน่วยความจำของ VM1 และ logger ตัวที่ 2 ทำการตรวจจับโปรเซสการทำงานโดยดึงข้อมูลจากหน่วยความจำของ VM2 จนถึงล็อกเกอร์ตัวที่ n ทำการตรวจจับโปรเซสการทำงานโดยดึงข้อมูลจากหน่วยความจำของ VMn

ผู้วิจัยดำเนินการสร้างการทดลองโดยแบ่งออกเป็น 2 หัวข้อ ดังนี้

1. การเตรียมอุปกรณ์สำหรับการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์
2. การสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์

1. การเตรียมอุปกรณ์สำหรับการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์

อุปกรณ์ที่ใช้ในการดำเนินการวิจัย โดยใช้เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ใช้ในการสร้าง IaaS คลาวด์ โดยแบ่งส่วนอุปกรณ์ออกเป็น ส่วนของโฮสต์ และส่วนของ VMs ดังรายละเอียดต่อไปนี้ ส่วนของเครื่อง Host ประกอบไปด้วยฮาร์ดแวร์และซอฟต์แวร์ดังนี้

1. เครื่องคอมพิวเตอร์ส่วนบุคคลใช้หน่วยประมวลผลกลาง CPU Intel Xeon@ CPU W3520 ความเร็ว 2.67 GHz จำนวน 8 Cores แบบ 64 บิต หน่วยความจำหลัก 24 GB และ ฮาร์ดดิสก์พื้นที่ 1 TB

2. ระบบปฏิบัติการฟิโดรา (Fedora) เวอร์ชัน 16 แบบ 64 บิต
3. ซอฟต์แวร์ไฮเปอร์ไวเซอร์ (Hypervisor) เช่น Xen เวอร์ชัน 4.1
4. ไบบารี libVMI
5. แอปพลิเคชัน logger

ส่วนของเครื่อง VM ประกอบไปด้วยฮาร์ดแวร์และซอฟต์แวร์ดังนี้

1. หน่วยประมวลผลกลางเสมือนที่สร้างจากซอฟต์แวร์ไฮเปอร์ไวเซอร์ จำนวน 1 core หน่วยความจำเสมือน 1 GB และฮาร์ดดิสก์ที่มีพื้นที่ 100 GB

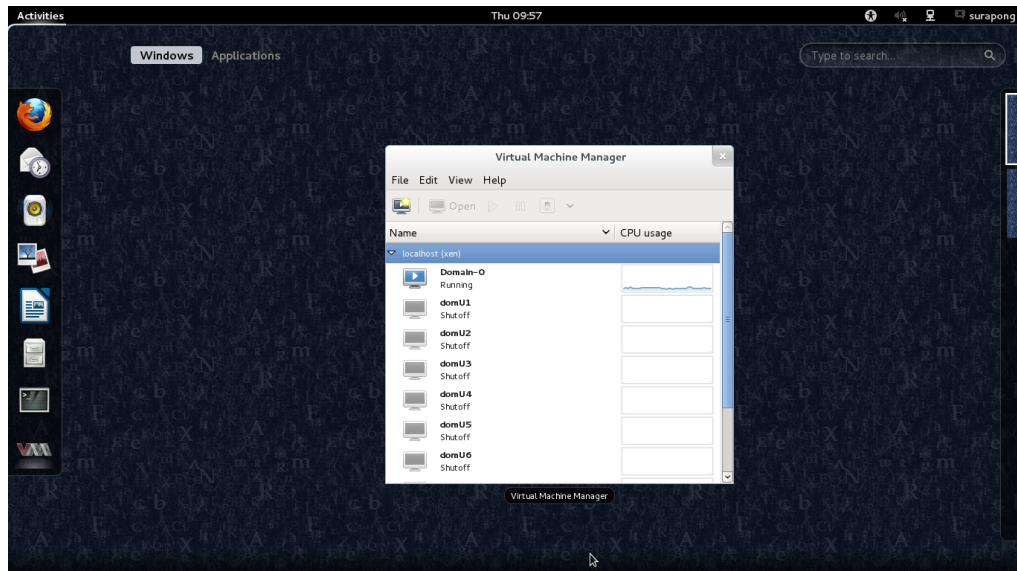
2. ระบบปฏิบัติการฟิโดรา (Fedora) เวอร์ชัน 16 แบบ 64 บิต
3. แอปพลิเคชัน read

2. การสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์

สำหรับการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ดังปรากฏในภาพ 7 ผู้วิจัยได้มีขั้นตอนการดำเนินงานในการสร้างระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ดังนี้

1. ติดตั้งระบบปฏิบัติการฟิโดรา ที่เป็นการทำงานในส่วนของผู้ให้บริการ (host)
2. ติดตั้งซอฟต์แวร์ไฮเปอร์ไวเซอร์ สำหรับการสร้าง IaaS คลาวด์
3. ติดตั้งไลบารี libVMI บนเครื่องผู้ให้บริการ

4. สร้าง IaaS คลาวด์ ซึ่งเป็นการทำงานในส่วนของ VM
5. ติดตั้งแอปพลิเคชันระบบบันทึกเหตุการณ์ลงในเครื่องผู้ให้บริการ
6. ติดตั้งแอปพลิเคชัน read ลงในเครื่องของผู้ใช้บริการ



ภาพ 7 แสดงติดตั้งระบบ IaaS คลาวด์

วิธีการทดสอบปริมาณการใช้งาน CPU และ RAM ของระบบบันทึกเหตุการณ์บนเครื่องคอมพิวเตอร์ของผู้ให้บริการ

ในการออกแบบการทดสอบนั้นผู้วิจัยได้ทำการออกแบบสภาวะแวดล้อมการทดลองในภาพ 6 โดยทำการทดสอบประมาณการใช้งานของ CPU และ RAM ในฝั่งของผู้ให้บริการ เพื่อนำข้อมูลที่ได้มาใช้ในการวิเคราะห์หาแนวทางในการบริหารจัดการระบบบันทึกเหตุการณ์ โดยผู้วิจัยได้แบ่งการทดสอบดังนี้

1 ทดสอบปริมาณการใช้งาน CPU และ RAM ของฝั่งโฮสต์ ด้วยการเปิดใช้งานเครื่องคอมพิวเตอร์ โดยที่ไม่เปิด VM หลังจากนั้นทำการเปิด VM 1 VM เปิด VM 2 VMs ไปจนถึง VM n VMs พร้อมกัน ตามลำดับ และทำการวัดในเรื่องของปริมาณการใช้งาน CPU และ RAM ด้วยคำสั่ง #SAR ในระบบปฏิบัติการ LINUX ดังภาพ 8 และ 9 โดยวัดปริมาณทุก ๆ 5 วินาที จำนวน 30 รอบ

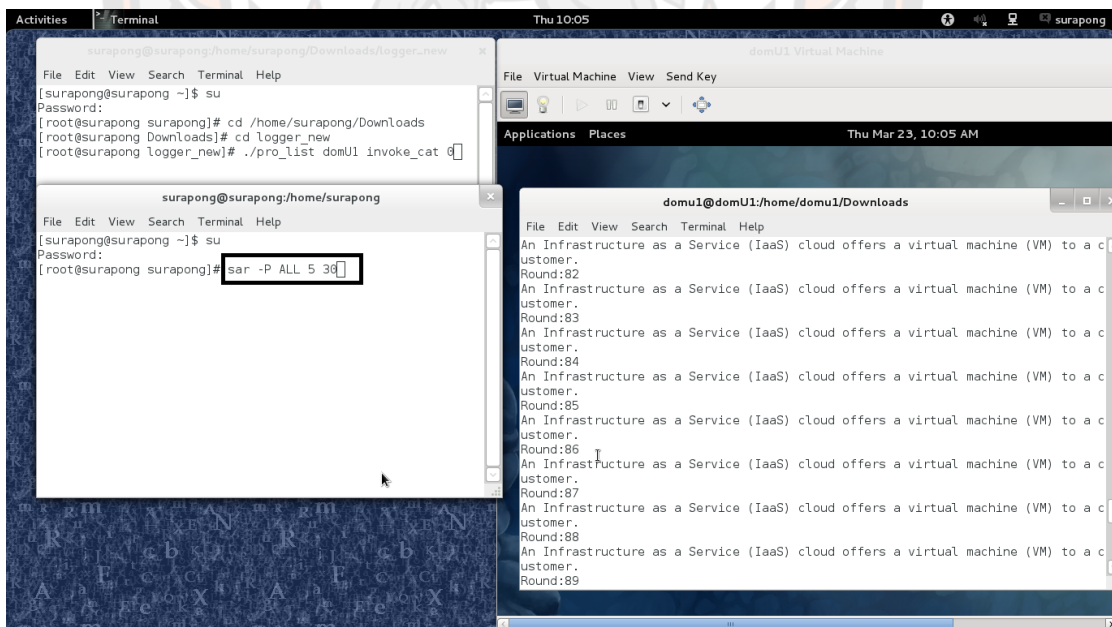
```
[root@surapong surapong]# sar -P ALL 5 30
```

ภาพ 8 แสดงคำสั่งการวัดปริมาณการใช้งาน CPU

```
[root@surapong surapong]# sar -r 5 30
```

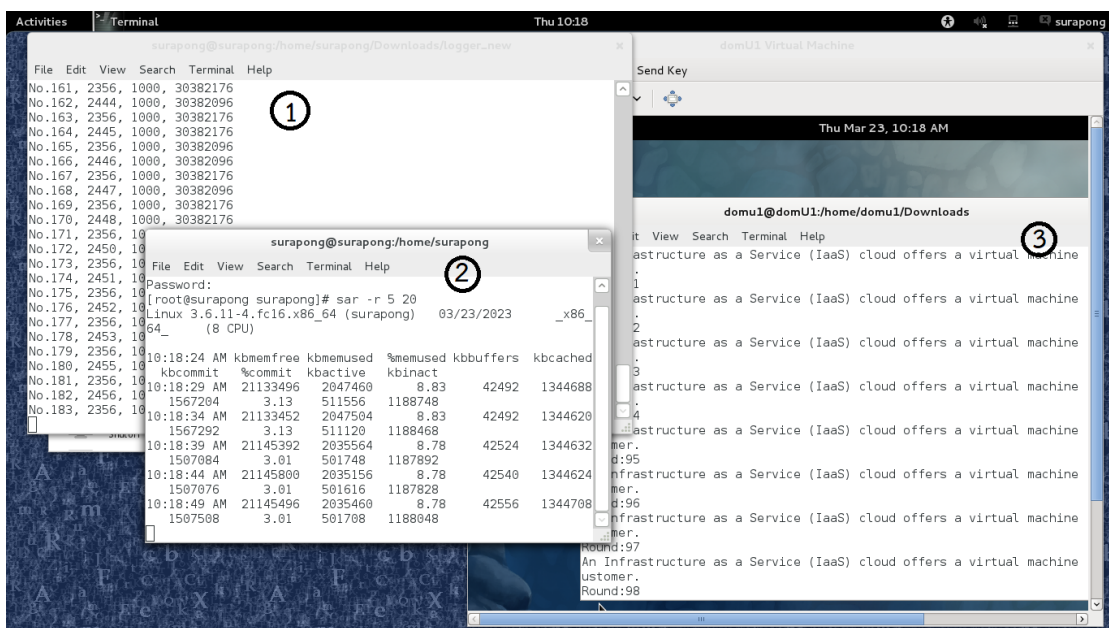
ภาพ 9 แสดงคำสั่งการวัดปริมาณการใช้งาน RAM

2. ทดสอบปริมาณการใช้งาน CPU และ RAM ของฝั่งโฮสต์ ด้วยการให้ VM มีการทำงานเกิดขึ้นด้วยการรันแอปพลิเคชัน read จำนวน 10,000 รอบ โดยวัดปริมาณการใช้งานของ CPU ในขณะที่ VM รันโปรเซส ทำงาน 1 VM, 2 VMs, 3 VMs, 4 VMs และ 5 VMs พร้อมกันตามลำดับ และทำการวัดในเรื่องของปริมาณการใช้งาน CPU และ RAM ด้วยคำสั่ง #SAR ในระบบปฏิบัติการ LINUX ดังภาพ 8 และ 9 โดยวัดปริมาณทุก ๆ 5 วินาที จำนวน 30 รอบ ดังภาพ 10



ภาพ 10 แสดงการรันแอปพลิเคชัน read บนเครื่อง VM

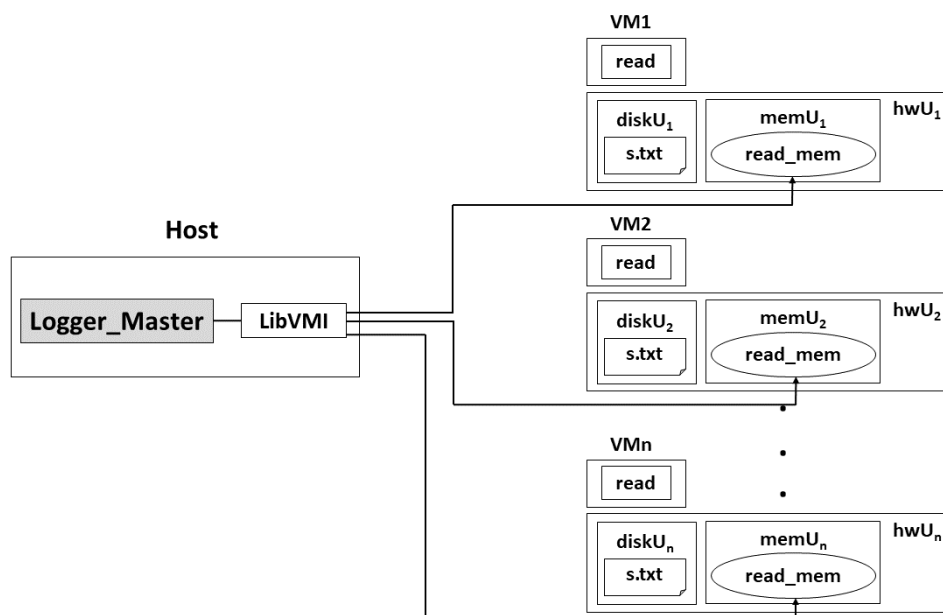
3. ทดสอบปริมาณการใช้งาน CPU และ RAM ของฝั่งโฮสต์ ด้วยการติดตั้งล็อกเกอร์ ในการทำงานตามภาพ 8 โดยให้ล็อกเกอร์ จับคู่ในการทำงานกับ VM โดยในฝั่งของโฮสต์นั้นจะทำการรันแอปพลิเคชันล็อกเกอร์ และฝั่งของ VMs จะทำการรันแอปพลิเคชัน read โดยเริ่มวัดปริมาณการใช้งาน CPU และ RAM แบบ 1 คู่, 2 คู่ และ 3 คู่ตามลำดับ และทำการวัดในเรื่องของปริมาณการใช้งาน CPU และ RAM ด้วยคำสั่ง #SAR ในระบบปฏิบัติการ LINUX ดังภาพ 8 และ 9 โดยวัดปริมาณ ทุก ๆ 5 วินาที จำนวน 30 รอบ



ภาพ 11 แสดงการรันแอปพลิเคชัน read บนเครื่อง VM พร้อมด้วยการรันแอปพลิเคชัน logger

จากภาพ 11 เมื่อทำการรันแอปพลิเคชัน read บนเครื่อง VM พร้อมด้วยการรันแอปพลิเคชัน logger บนเครื่องโฮสต์ โดยหมายเลข 1 เป็นผลลัพธ์ที่ได้จากการดึงข้อมูลใน memory ของเครื่อง VM ออกมาซึ่งประกอบไปด้วยส่วนของการนับ (counter) รหัสโปรเซส (pid) รหัสผู้ใช้ (uid) และชื่อโปรเซส (procname) หมายเลข 2 เป็นผลลัพธ์ที่ได้จากการรันคำสั่งในการวัดปริมาณการใช้งาน RAM โดยดูค่าจาก %memused ที่เป็นปริมาณการใช้งาน RAM ที่ผู้ใช้งานให้ใช้ในหน่วยของเปอร์เซ็นต์ และหมายเลข 3 แสดงหน้าจอการทำงานของเครื่อง VM ในการอ่านไฟล์ s.txt

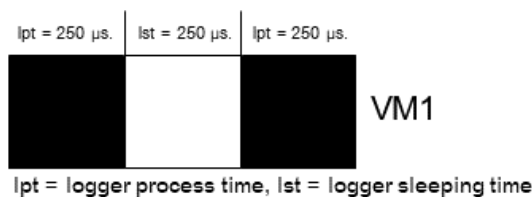
แนวคิดวิธีการในการจัดสรรระบบบันทึกเหตุการณ์



ภาพ 12 แสดงการทำงานของ Logger แบบรวมศูนย์

จากภาพ 12 ในกล่องด้านซ้ายมือส่วนของ Host ประกอบไปด้วย Logger_Master 1 ล็อกเกอร์และ LibVMI ส่วนในกล่องทางด้านขวามือจะประกอบไปด้วยส่วนประกอบของ VM ตัวที่ 1 ถึงตัวที่ n

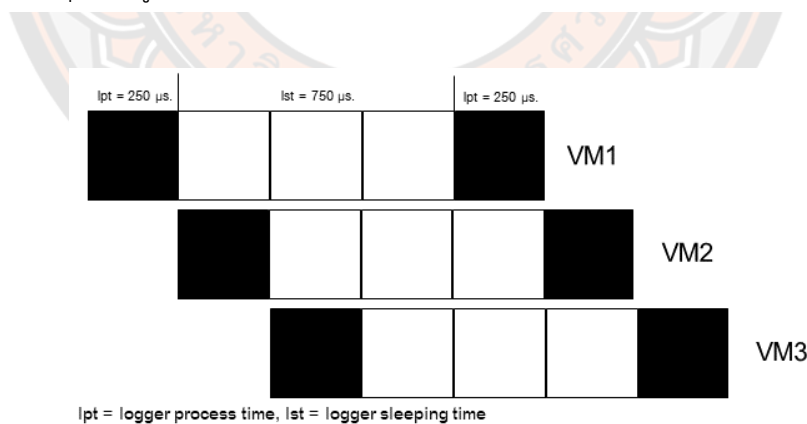
โดยปกติการทำงานของล็อกเกอร์ในการตรวจจับโปรเซสการทำงานใน VM นั้นจะเข้าไปหยุดการทำงานของ VM เพื่อที่จะเข้าไปดึงข้อมูลจากหน่วยความจำของ VM ออกมา ถ้ายังมีการหยุดการทำงานของ VM มากเท่าไรแสดงว่ามีการดึงข้อมูลจากหน่วยความจำมาตรวจสอบมากเท่านั้น ทำให้ความถูกต้องในการตรวจจับนั้นสูงขึ้น แต่จะทำให้ VM นั้นหยุดการทำงานบ่อยส่งผลให้ประสิทธิภาพการทำงานของ VM ลดลง จากงานวิจัยของ Auxorn et al. (2020) ได้แสดงช่วงเวลาการทำงานของล็อกเกอร์ดังนี้



ภาพ 13 แสดงช่วงเวลาการทำงานของล็อกเกอร์

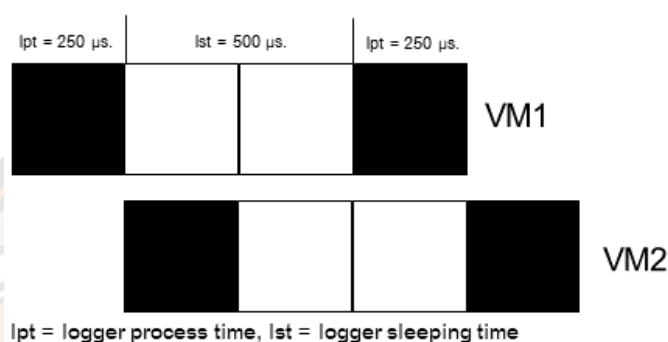
จากภาพ 13 ช่วงเวลาการทำงานของล็อกเกอร์ กล่องสีดำคือช่วงที่ล็อกเกอร์ เข้าไปทำงาน และหยุดการทำงานของ VM เรียกว่า logger process time หรือ lpt ส่วนกล่องสีขาวคือช่วงที่ล็อกเกอร์ปล่อยให้ VM ทำงานตามปกติ เรียกว่า logger sleeping time หรือ lst

ในการทำงานของล็อกเกอร์ นั้นโดยที่ล็อกเกอร์ 1 ล็อกเกอร์ จะเข้าไปจับ VM 1 VM ซึ่งเกิดปัญหาว่าเครื่องคอมพิวเตอร์ที่จำลองการทำงานนั้นสามารถที่จะรองรับล็อกเกอร์ 5 ตัว กับ VM 5 VMs เท่านั้น ทั้งนี้การบริหารจัดการ VMI จะสามารถช่วยให้รองรับความต้องการของผู้ใช้บริการ VM ได้ โดยที่ความต้องการของผู้ใช้งานนั้นย่อมมีความแตกต่างกันไป ขึ้นอยู่กับว่าผู้ใช้บริการต้องการประสิทธิภาพการทำงาน หรือต้องการความปลอดภัยในการตรวจจับโปรเซสการทำงานโดยล็อกเกอร์ โดยงานวิทยานิพนธ์นี้ไม่ได้ลงรายละเอียดไปที่การปรับปรุงล็อกเกอร์ แต่เน้นไปที่การสร้างวิธีการในการบริหารจัดการของ VMI (Logger) เพื่อให้ตอบสนองกับความต้องการของผู้ใช้งาน โดยที่จะได้แสดงการแบ่งกลุ่มของผู้ใช้บริการในการจัดสรรทรัพยากรของ VM ได้ดังตัวอย่างในภาพ 16 – 17



ภาพ 14 แสดงกรณีการทำงานของล็อกเกอร์ 1 ตัว ต่อ VM 3 เครื่อง

จากภาพ 14 การทำงานของล็อกเกอร์ กล่องสีดำคือช่วงที่ล็อกเกอร์ เข้าไปทำงาน และหยุดการทำงานของ VM ส่วนกล่องสีขาวคือช่วงที่ล็อกเกอร์ปล่อยให้ VM ทำงานตามปกติ โดยที่ล็อกเกอร์ปล่อยให้ VM1 ทำงานในช่วงเวลากล่องสีขาวเป็นระยะเวลา 750 μ s. และล็อกเกอร์จะมาทำงานที่ VM2 แทน เมื่อล็อกเกอร์ทำงานที่ VM2 เสร็จ ล็อกเกอร์จะปล่อยให้ VM2 ทำงาน และจะมาทำงานที่ VM3 แทน จะวนการทำงานแบบนี้ไปเรื่อย ๆ จนกลับไปทำงานที่ VM1 อีกครั้ง สำหรับในกรณีนี้จะหมายถึงว่าล็อกเกอร์ 1 ล็อกเกอร์ จะบริหารจัดการ VM ได้ 3 VMs



ภาพ 15 แสดงกรณีการทำงานของล็อกเกอร์ 1 ตัว ต่อ VM 2 เครื่อง

จากภาพ 15 ลักษณะการทำงานของล็อกเกอร์ จะเหมือนกับภาพ 16 ซึ่งจะเห็นได้ว่าลักษณะการทำงานแบบนี้จะหมายถึงล็อกเกอร์ 1 ล็อกเกอร์ จะบริหารจัดการ VM ได้ 2 VMs

เมื่อเปรียบเทียบกันระหว่างการทำงานของภาพ 14 และ 15 จะแสดงให้เห็นว่าการทำงานในภาพ 14 จะทำให้เกิดประสิทธิภาพในการทำงานของ VM มากกว่าภาพ 15 และภาพ 15 จะให้ค่าความถูกต้องในการตรวจจับโปรเซสการทำงานมากกว่า โดยการอ้างอิงจากงานวิจัยของ Auxsorn et al. (2020) ที่ได้ทำการทดสอบประสิทธิภาพการทำงานของการทำงานเครื่อง VM ของผู้ให้บริการ โดยพบว่าเมื่อมีการเข้าไปหยุดการทำงานของเครื่อง VM ของผู้ให้บริการบ่อยมากเท่าไร ยิ่งทำให้ประสิทธิภาพการทำงานของเครื่อง VM ของผู้ให้บริการลดลง เนื่องจากมีการเข้าไปหยุดการทำงานของเครื่อง VM ของผู้ให้บริการบ่อย แต่ความถูกต้องในการตรวจจับโปรเซสการทำงานจะมีมากขึ้นซึ่งจะแปรผกผันกันระหว่างความถูกต้องกับประสิทธิภาพการทำงานของเครื่อง VM ของผู้ให้บริการ โดยได้อธิบายไว้ในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้องในหัวข้อความปลอดภัยและประสิทธิภาพการทำงานของคอมพิวเตอร์เสมือน

เครื่องคอมพิวเตอร์ที่ใช้ในการทดลองนั้น ประกอบไปด้วย CPU Intel Xeon@ CPU W3520 2.67 GHz 64 bits 8 Cores, SDRAM 24 GB และ HDD 1 TB โดยรองรับการทำงานของ logger ได้ 5 ตัว และ VM 5 ตัว ผู้วิจัยจึงได้วางแนวทางในการจัดสรรทรัพยากร ได้ 5 รูปแบบดังนี้

Logger #1	Logger #2	Logger #3	Logger #4	Logger #5	
1	1	1	1	1	Solution 1
1	1	1	2		Solution 2
1	1	3			Solution 3
1	4				Solution 4
5					Solution 5

ภาพ 16 แสดงวิธีการการจัดสรรล็อกเกอร์ให้กับผู้ใช้บริการ

จากภาพ 16 กล่องสีขาวแสดงถึงล็อกเกอร์ และหมายเลขที่อยู่ในกล่องสีขาว หมายถึงจำนวนของ VM ที่ทำงานกับล็อกเกอร์นั้น โดยจำลองถึงผู้ใช้บริการจำนวน 5 คน ในกรณีการจัดสรรการทำงานในรูปแบบที่ 1 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความปลอดภัยในการตรวจจับโปรเซสโดยล็อกเกอร์ ทั้งหมด 5 คน จะมีการทำงานของล็อกเกอร์ แบบ 1 ล็อกเกอร์ ต่อ 1 VM การจัดสรรการทำงานในรูปแบบที่ 2 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความปลอดภัยในการตรวจจับโปรเซสโดยล็อกเกอร์ จำนวน 3 คน และต้องการประสิทธิภาพในการทำงานของ VM จำนวน 2 คน การจัดสรรการทำงานในรูปแบบที่ 3 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความปลอดภัยในการตรวจจับโปรเซสโดยล็อกเกอร์ จำนวน 2 คน และต้องการประสิทธิภาพในการทำงานของ VM จำนวน 3 คน การจัดสรรการทำงานในรูปแบบที่ 4 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความปลอดภัยในการตรวจจับโปรเซสโดยล็อกเกอร์ จำนวน 1 คน และต้องการประสิทธิภาพในการทำงานของ VM จำนวน 4 คน และการจัดสรรการทำงานในรูปแบบที่ 5 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการประสิทธิภาพในการทำงานของ VM จำนวน 5 คน

วิธีการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

ในการสร้างโมเดลทางคณิตศาสตร์ได้ใช้รูปแบบของการหาค่าที่เหมาะสม (optimization) มาใช้ในการสร้างโมเดลทางคณิตศาสตร์เพื่อให้ได้วิธีการจัดสรรระบบบันทึกเหตุการณ์ โดยให้เหลือจำนวนตัวล็อกเกอร์ที่ไม่ถูกใช้งานมากที่สุด ซึ่งสืบเนื่องมาจากการทดลองในข้อการทดสอบปริมาณการใช้งาน CPU และ RAM ของระบบบันทึกเหตุการณ์บนเครื่องคอมพิวเตอร์ของผู้ให้บริการ โดยเมื่อมีการใช้ล็อกเกอร์ในการทำงานน้อยตัวจะทำให้สามารถลดปริมาณการใช้งาน CPU ของเครื่องคอมพิวเตอร์ได้ ทั้งนี้ได้ทำการสร้างโมเดลทางคณิตศาสตร์ ตามขั้นตอนของการทำ optimization ดังนี้

1. ตัวแปรการตัดสินใจ (Decision Variable)
2. ฟังก์ชันจุดประสงค์ (Objective Function)
3. ข้อจำกัด (Constraint) ต่าง ๆ ได้แก่ Performance และ security และแบ่งการดำเนินงานออกเป็น 4 ขั้นตอนดังนี้

1. การสร้างโมเดลทางคณิตศาสตร์แบบมีข้อจำกัด (Constrained Optimization)
2. การสร้างโมเดลทางคณิตศาสตร์แบบไม่มีข้อจำกัด (Unconstrained Optimization)
3. การสร้างผังงานทำงานของตัวปรับรูปแบบการจัดสรรทรัพยากร
4. การเปรียบเทียบการสร้างวิธีการระหว่างแบบ Constrained กับแบบ Unconstrained

1. การสร้างโมเดลทางคณิตศาสตร์แบบมีข้อจำกัด (Constrained Optimization)

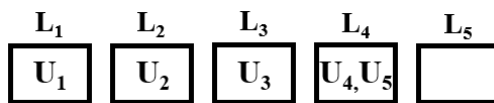
โดยในกระบวนการสร้างโมเดลทางคณิตศาสตร์ผู้วิจัยเริ่มต้นจากการคิดวิธีการทำงาน โดยนำวิธีการจัดสรรล็อกเกอร์จากภาพ 16 วิธีการการจัดสรรล็อกเกอร์ให้กับผู้ใช้บริการ และผู้ให้บริการ (โดยมีข้อจำกัดว่า VM 1 เครื่อง สามารถถูกบริหารจัดการโดยล็อกเกอร์ได้เพียงหนึ่งล็อกเกอร์เท่านั้น) มาทำการหาโมเดลคณิตศาสตร์ (Mathematic Model) สำหรับการจัดสรรล็อกเกอร์ ดังนี้

กำหนดให้ logger แทนด้วยสัญลักษณ์ $L_1, L_2, L_3, \dots, L_m$

VM แทนด้วยสัญลักษณ์ $U_1, U_2, U_3, \dots, U_n$

สมมติมีล็อกเกอร์ 5 ตัว VM 5 ตัว ผู้วิจัยแทนล็อกเกอร์ตัวที่ 1 ด้วย L_1 ล็อกเกอร์ตัวที่ 2 ด้วย L_2 ล็อกเกอร์ตัวที่ 3 ด้วย L_3 ล็อกเกอร์ตัวที่ 4 ด้วย L_4 และล็อกเกอร์ตัวที่ 5 ด้วย L_5 ตามลำดับ และแทน VM ตัวที่ 1 ด้วย U_1 VM ตัวที่ 2 ด้วย U_2 VM ตัวที่ 3 ด้วย U_3 VM ตัวที่ 4 ด้วย U_4 และ VM ตัวที่ 5 ด้วย U_5 ตามลำดับ

จากภาพ 16 ผู้วิจัยขอยกตัวอย่างการทำงานของวิธีการที่ 2 (Solution 2) จะได้



สามารถแสดงกระบวนการในการหาโมเดลทางคณิตศาสตร์ได้ตามขั้นตอนดังนี้

ขั้นตอนที่ 1 แทนค่าการทำงานของลือกเกอร์ ในกรณีที่ VM ถูกกำหนดให้อยู่ในลือกเกอร์ ลือกเกอร์นั้นจะถูกกำหนดค่าให้เป็น '1' กรณีที่ VM ไม่ได้ถูกกำหนดให้อยู่ในลือกเกอร์ ลือกเกอร์นั้นจะถูกกำหนดค่าให้เป็น '0'

จะได้

$$\begin{aligned}
 U_1L_1 &= 1, & U_1L_2 &= 0, & U_1L_3 &= 0, & U_1L_4 &= 0, & U_1L_5 &= 0 \\
 U_2L_1 &= 0, & U_2L_2 &= 1, & U_2L_3 &= 0, & U_2L_4 &= 0, & U_2L_5 &= 0 \\
 U_3L_1 &= 0, & U_3L_2 &= 0, & U_3L_3 &= 1, & U_3L_4 &= 0, & U_3L_5 &= 0 \\
 U_4L_1 &= 0, & U_4L_2 &= 0, & U_4L_3 &= 0, & U_4L_4 &= 1, & U_4L_5 &= 0 \\
 U_5L_1 &= 0, & U_5L_2 &= 0, & U_5L_3 &= 0, & U_5L_4 &= 1, & U_5L_5 &= 0
 \end{aligned}$$

ดังนั้นจะได้

$$\begin{aligned}
 U_1 &= [1 \ 0 \ 0 \ 0 \ 0] \\
 U_2 &= [0 \ 1 \ 0 \ 0 \ 0] \\
 U_3 &= [0 \ 0 \ 1 \ 0 \ 0] \\
 U_4 &= [0 \ 0 \ 0 \ 1 \ 0] \\
 U_5 &= [0 \ 0 \ 0 \ 1 \ 0]
 \end{aligned}$$

$$L_1 = \{U_1\}^T, L_2 = \{U_2\}^T, L_3 = \{U_3\}^T, L_4 = \{U_4\}^T, L_5 = \{U_5\}^T$$

และทำการแปลงข้อมูลให้อยู่ในรูปของเมทริกซ์ขนาด $m \times n$ (5×5)

โดยที่ m คือ จำนวนของ VM

n คือ จำนวนของลือกเกอร์

โดยให้ X เป็นตัวแปรการตัดสินใจ ในขณะที่ $X \in I^{m \times n}$

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$$

หรือ

$$X = [x_{ij}]_{m \times n}$$

โดยที่

$$x_{ij} \in \{0,1\} \text{ สำหรับแต่ละ } i \in \{1,2, \dots, m\}, j \in \{1,2, \dots, n\}$$

$$x_{ij} = \begin{cases} 1, & \text{the } i^{\text{th}} \text{ VM is assigned to the } j^{\text{th}} \text{ logger} \\ 0, & \text{otherwise} \end{cases}$$

สำหรับแต่ละ i และ j

$$U_i = [x_{i1} \quad \dots \quad x_{in}]$$

$$L_j = [x_{1j} \quad \dots \quad x_{mj}]^T$$

จะได้

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5}$$

$$\therefore X = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix}_{5 \times 5}$$

สำหรับการกำหนดให้ VM 1 เครื่องสามารถรองรับการทำงานของล็อกเกอร์ได้เพียง 1 ตัวเท่านั้น เนื่องมาจากการทำงานของ LibVMI นั้นเมื่อเข้าไปดึงข้อมูลจากหน่วยความจำของเครื่อง VM นั้น จะทำการล็อกหน่วยความจำ ไม่ให้ล็อกเกอร์ตัวอื่นสามารถเข้ามาดึงข้อมูลได้ จึงเป็นเหตุผลที่ผู้วิจัยได้กำหนดให้ VM 1 เครื่อง สามารถอยู่ภายใต้การทำงานของล็อกเกอร์ได้เพียง 1 ตัวเท่านั้น

ขั้นตอนที่ 2 พิจารณาการใช้ล็อกเกอร์ให้น้อยที่สุด โดยเหลือล็อกเกอร์ที่ไม่ได้มีภาระงานมากที่สุด ซึ่งเป็นสิ่งที่ผู้ให้บริการต้องการ โดยจะนับจำนวน VM ที่อยู่ในล็อกเกอร์แต่ละตัว

$$\begin{aligned} \text{จำนวน VM ที่อยู่ในล็อกเกอร์ตัวที่ 1} &= x_{11} + x_{21} + x_{31} + x_{41} + x_{51} \\ &= 1 + 0 + 0 + 0 + 0 = 1 \end{aligned}$$

$$\begin{aligned} \text{จำนวน VM ที่อยู่ในล็อกเกอร์ตัวที่ 2} &= x_{12} + x_{22} + x_{32} + x_{42} + x_{52} \\ &= 0 + 1 + 0 + 0 + 0 = 1 \end{aligned}$$

$$\begin{aligned} \text{จำนวน VM ที่อยู่ในล็อกเกอร์ตัวที่ 3} &= x_{13} + x_{23} + x_{33} + x_{43} + x_{53} \\ &= 0 + 0 + 1 + 0 + 0 = 1 \end{aligned}$$

$$\text{จำนวน VM ที่อยู่ในล็อกเกอร์ตัวที่ 4} = x_{14} + x_{24} + x_{34} + x_{44} + x_{54}$$

$$= 0 + 0 + 0 + 1 + 1 = 2$$

จำนวน VM ที่อยู่ในล็อกเกอร์ตัวที่ 5 = $x_{15} + x_{25} + x_{35} + x_{45} + x_{55}$

$$= 0 + 0 + 0 + 0 + 0 = 0$$

ดังนั้นจึงได้สมการในการหาจำนวน VM ที่อยู่ในล็อกเกอร์ ดังนี้

$$N(L_j) = \sum_{i=1}^m x_{ij} \quad (2)$$

โดยที่ $N(L_j)$ คือ จำนวน VM ในล็อกเกอร์แต่ละตัว

x_{ij} คือ ค่าที่อยู่ในเมทริกซ์ X ตำแหน่งที่ i, j

m คือ จำนวนของ VM

ขั้นตอนที่ 3 เมื่อได้จำนวนของ VM ในล็อกเกอร์แต่ละตัวแล้ว จึงได้ทำการกำหนดค่า '0' ในกรณีที่ VM ถูกกำหนดให้อยู่ในล็อกเกอร์ หมายถึงค่า $N(L_j)$ มีค่ามากกว่า 0 และกำหนดค่า '1' ในกรณีที่ VM ไม่ได้ถูกกำหนดให้อยู่ในล็อกเกอร์ หมายถึงค่า $N(L_j)$ มีค่าเท่ากับ 0 เพื่อจะนับจำนวนของล็อกเกอร์ที่ไม่ถูกใช้งาน โดยอธิบายในรูปแบบของซูดโค้ด ได้ดังนี้

```

Begin
    Variables:  $N(L_j), O_j$ 
    Input  $N(L_j)$ 
    If  $N(L_j) = 0$  Then
         $O_j = 1$ 
    Else
         $O_j = 0$ 
End

```

จากขั้นตอนที่ 2 เมื่อนำค่า $N(L_j)$ มากำหนดค่าตามการทำงานของซูดโค้ด จะพบว่า O_j มีค่าเป็น 0

เนื่องจาก $O_1 + O_2 + O_3 + O_4 + O_5 = 0 + 0 + 0 + 0 + 1 = 1$
จะได้

$$O(N(L_j)) = \begin{cases} 1, & \text{if } N(L_j) = 0 \\ 0, & \text{otherwise} \end{cases}$$

ดังนั้น จะได้ฟังก์ชันจุดประสงค์ ในรูปแบบโมเดลทางคณิตศาสตร์ ดังนี้

$$\text{maximize } f(X) = \sum_{j=1}^n O_j(X) \quad (3)$$

subject to

$x_{ij} \in \{0,1\}$ สำหรับแต่ละ $i \in \{1,2, \dots, m\}, j \in \{1,2, \dots, n\}$

$$\sum_{j=1}^n x_{ij}, \quad \text{สำหรับแต่ละ } j \in \{1,2, \dots, n\}$$

โดยที่

X คือ เมทริกซ์ขนาด $m \times n$

m คือ จำนวนของ VM

n คือ จำนวนของสล็อตเกอร์

และ

$O_j(X) \in \{0,1\}$ คือ จำนวนสล็อตเกอร์ที่ไม่ได้ใช้งาน

ขั้นตอนที่ 4 ทำการกำหนดข้อจำกัด โดยผู้วิจัยได้พิจารณารูปแบบความต้องการของผู้ใช้บริการในการกำหนดระดับความปลอดภัยของไฟล์และระดับประสิทธิภาพในการทำงานของเครื่อง VM โดยได้อธิบายไว้ในบทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้องในหัวข้อความปลอดภัยของไฟล์และประสิทธิภาพการทำงานของคอมพิวเตอร์เสมือน โดยอ้างอิงจากงานวิจัยของ Auxsorn et al. (2020) ผู้วิจัยจึงได้แบ่งค่าระดับของความปลอดภัยของไฟล์และประสิทธิภาพการทำงานทั้งหมด 1 – 5 ระดับ

ระดับความปลอดภัย

ระดับ 5 หมายถึง ระดับความปลอดภัยน้อยที่สุด

(ระดับความปลอดภัยอยู่ที่ 86.47%)

ระดับ 4 หมายถึง ระดับความปลอดภัยน้อย

(ระดับความปลอดภัยอยู่ที่ 99.67%)

ระดับ 3 หมายถึง ระดับความปลอดภัยปานกลาง

(ระดับความปลอดภัยอยู่ที่ 99.75%)

ระดับ 2 หมายถึง ระดับความปลอดภัยมาก

(ระดับความปลอดภัยอยู่ที่ 99.82%)

ระดับ 1 หมายถึง ระดับความปลอดภัยมากที่สุด

(ระดับความปลอดภัยอยู่ที่ 99.86%)

ระดับประสิทธิภาพในการทำงาน

ระดับ 5 หมายถึง ระดับประสิทธิภาพในการทำงานมากที่สุด

(ระดับประสิทธิภาพอยู่ที่ 83.00%)

ระดับ 4 หมายถึง ระดับประสิทธิภาพในการทำงานมาก

(ระดับประสิทธิภาพอยู่ที่ 80.00%)

ระดับ 3 หมายถึง ระดับประสิทธิภาพในการทำงานปานกลาง

(ระดับประสิทธิภาพอยู่ที่ 73.00%)

ระดับ 2 หมายถึง ระดับประสิทธิภาพในการทำงานน้อย

(ระดับประสิทธิภาพอยู่ที่ 66.00%)

ระดับ 1 หมายถึง ระดับประสิทธิภาพในการทำงานน้อยที่สุด

(ระดับประสิทธิภาพอยู่ที่ 50.00%)

จากระดับความต้องการทั้งระดับความปลอดภัยของไฟล์ และประสิทธิภาพในการทำงานของเครื่อง VM ผู้วิจัยจึงได้ทำการออกแบบข้อจำกัดในการทำงานของการจัดสรรระบบบันทึกเหตุการณ์ โดยจำนวน VM ในลือกเกอร์น้อยกว่า ระดับความปลอดภัย และจำนวน VM ในลือกเกอร์มากกว่า ระดับประสิทธิภาพ ซึ่งระดับความปลอดภัยแปรผกผันกับระดับประสิทธิภาพ และระดับประสิทธิภาพแปรผันกับจำนวนลือกเกอร์ จึงได้ข้อจำกัด ดังนี้

$$\begin{aligned} N(L_j) &\leq S_i, \quad U_i \in L_j \\ N(L_j) &\geq P_i, \quad U_i \in L_j \\ \therefore P_i &\leq N(L_j) \leq S_i, \quad U_i \in L_j \end{aligned} \quad (4)$$

S_i คือ ค่าระดับความปลอดภัยของไฟล์ของผู้ใช้บริการ

P_i คือ ค่าระดับประสิทธิภาพการทำงานของเครื่อง VMs

โดยที่ U_i เป็นสมาชิกของ L_j

จากการกำหนดข้อจำกัดความต้องการทั้งระดับความปลอดภัย และประสิทธิภาพ จึงได้ฟังก์ชันจุดประสงค์ ในรูปแบบโมเดลทางคณิตศาสตร์ ดังนี้

$$\text{maximize } f(X) = \sum_{j=1}^n O_j(X) \quad (5)$$

subject to

$$x_{ij} \in \{0,1\} \text{ สำหรับแต่ละ } i \in \{1,2, \dots, m\}, j \in \{1,2, \dots, n\}$$

$$\sum_{j=1}^n x_{ij}, \quad \text{สำหรับแต่ละ } j \in \{1,2, \dots, n\}$$

$$P_i \leq N_j \leq S_i, \quad U_i \in L_j$$

โดยที่

X คือ เมทริกซ์ขนาด $m \times n$

m คือ จำนวนของ VM

n คือ จำนวนของล็อกเกอร์

S_i คือ ค่าระดับความปลอดภัยของไฟล์ของผู้ใช้บริการ

P_i คือ ค่าระดับประสิทธิภาพการทำงานของเครื่อง VMs

โดยที่ U_i เป็นสมาชิกของ L_j

และ

$O_j(X) \in \{0,1\}$ คือ จำนวนล็อกเกอร์ที่ไม่ได้ใช้งาน

จากกระบวนการในข้างต้น ผู้วิจัยสามารถกำหนดตัวแปรการตัดสินใจที่เหมาะสมสำหรับ

ปัญหาดังสมการ (6)

2. การสร้างโมเดลทางคณิตศาสตร์แบบไม่มีข้อจำกัด (Unconstrained Optimization)

สำหรับในกรณีที่ไม่สามารถหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่เป็นไปตามข้อจำกัดได้ ทางผู้วิจัยได้ทำการกำหนดรูปแบบในการใช้ฟังก์ชันการปรับ ซึ่งเป็นการเปลี่ยนโมเดลที่มีข้อจำกัดเป็นโมเดลที่ไม่มีข้อจำกัด (Unconstraint) เพื่อนำมาใช้ในแก้ไขปัญหาดังกล่าว ยกตัวอย่างเช่น

Number of Loggers: 5

Number of VMs: 5

Security Level of VMs: [1, 2, 4, 4, 3]

Performance Level of VMs: [1, 2, 3, 3, 3]

ภาพ 17 แสดงข้อมูลนำเข้าสำหรับการสร้างโมเดลทางคณิตศาสตร์

ในกรณีที่ระดับความปลอดภัยเป็น 1 และระดับประสิทธิภาพเป็น 1 สามารถจัดสรร logger 1 ตัว ให้กับ VM1 ได้ดังภาพ 16 ในรูปแบบของ Logger#1 และในกรณีที่ระดับความปลอดภัยเป็น 2, 4, 4, 3 และระดับประสิทธิภาพเป็น 2, 3, 3, 3 สามารถจัดวิธีการในการจัดสรร logger ได้ 3 รูปแบบของ Logger#2 และ Logger#3 ดังภาพ 18

Logger #1	Logger #2	Logger #3	Logger #4	Logger #5	
1	1	3			Solution 1
1	2	2			Solution 2
1	4				Solution 3

ภาพ 18 แสดงวิธีการจัดสรร VM ให้กับล็อกเกอร์

จากภาพ 18 จะเห็นได้ว่าการจัดสรรนั้นไม่เป็นไปตามข้อจำกัด ที่กำหนดไว้ในสมการที่ (4) จึงต้องทำการคำนวณหาวิธีการจัดสรรล็อกเกอร์แบบไม่เป็นไปตามข้อจำกัด โดยกำหนดฟังก์ชันการปรับ ในกรณีที่ไม่เป็นไปตามข้อจำกัดตามชุดโค้ด

Begin

Variables: $N(L_j), S_i, P_i, cost, g_s, g_p$

Input $N(L_j), S_i, P_i$

If $S_i < N(L_j)$ Then

$cost += g_s * (S_i - N(L_j))$

If $P_i > N(L_j)$ Then

$cost += g_p * (N(L_j) - P_i)$

End

$cost$ คือ ค่าลงโทษในกรณีที่เกิดข้อจำกัด (เป็นค่าติดลบ)

g_s คือ ค่าถ่วงน้ำหนักของการปรับในกรณีผิดข้อจำกัดของความปลอดภัย โดยกำหนดให้มีค่าน้ำหนักเท่ากับ 2 เนื่องจากมองในเรื่องความปลอดภัยหลัก

g_p คือ ค่าถ่วงน้ำหนักของการปรับในกรณีผิดข้อจำกัดของประสิทธิภาพการทำงาน โดยกำหนดให้มีค่าน้ำหนักเท่ากับ 1

และทำการหาค่าสูงสุดของ Objective Function ดังนี้ สมการที่ (6)

$$\text{maximize } f(X) = \left(\sum_{j=1}^n O_j(X) \right) + \sum_{i=1}^m cost(U_i) \quad (6)$$

subject to

X คือ เมทริกซ์ขนาด $m \times n$

$x_{ij} \in \{0,1\}$ สำหรับแต่ละ $i \in \{1,2, \dots, m\}, j \in \{1,2, \dots, n$

$$\sum_{j=1}^n x_{ij}, \quad \text{สำหรับแต่ละ } j \in \{1,2, \dots, n\}$$

สำหรับแต่ละ $i \in \{1,2, \dots, m\}$ สามารถกำหนด cost function ได้ตามสมการ

$$\text{cost}(U_i) = \begin{cases} g_p(N_k - P_i), & \text{if } N_k < P_i \\ 0, & \text{if } P_i \leq N_k \leq S_i \\ g_s(S_i - N_k), & \text{if } N_k > S_i \end{cases}$$

โดยที่

$$U_i = [x_{i1}, x_{i2}, \dots, x_{ik}]$$

และ

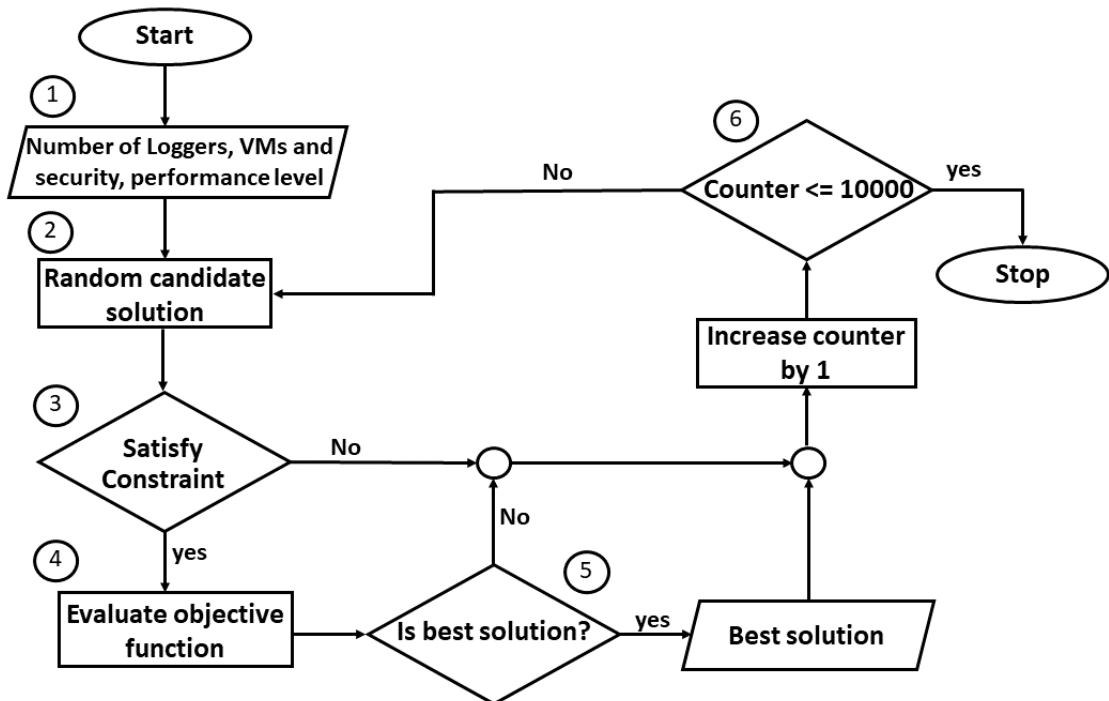
$$x_{ik} = 1 \text{ สำหรับบาง } k \in \{1,2, \dots, n\}$$

$$\text{ซึ่ง } N(L_j) = N_k$$

3. การสร้างฟังก์ชันของการทำงานโมเดลทางคณิตศาสตร์

การสร้างฟังก์ชันการทำงานของโมเดลทางคณิตศาสตร์ที่ได้สร้างขึ้น ผู้วิจัยได้กำหนดใน ส่วนของข้อมูลนำเข้าที่ประกอบไปด้วยส่วนของผู้ให้บริการ ได้แก่ จำนวนลือกเกอร์ และจำนวน VM ที่สามารถใช้งานได้ และในส่วนของผู้ใช้บริการ ได้แก่ ระดับความปลอดภัย และระดับประสิทธิภาพ การทำงาน และการสร้างสถานการณ์การวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ผู้วิจัยได้ใช้ วิธีการ Random Search Algorithm ในการหาวิธีการที่เป็นไปได้ โดยมีการกำหนดข้อจำกัดแฝง สำหรับ VM 1 ตัว จะไม่สามารถไปอยู่กับลือกเกอร์ได้หลายตัว แล้วนำไปเขียนโปรแกรมการทำงาน ด้วยภาษา Python

แผนผังการทำงานในรูปแบบมีข้อจำกัด



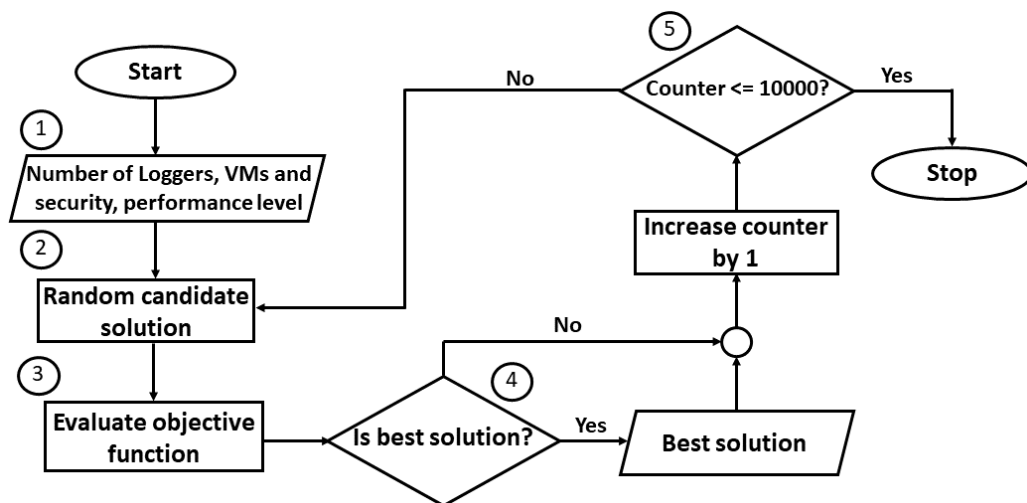
ภาพ 19 แสดงผังงานการทำงานของวิธีการที่เหมาะสมด้วยรูปแบบมีข้อจำกัด

จากภาพ 19 ผังงานการทำงานของวิธีการที่เหมาะสมด้วยรูปแบบมีข้อจำกัด โดยมีวิธีการทำงานดังนี้

1. เริ่มต้นด้วยการรับข้อมูลนำเข้าประกอบไปด้วย จำนวนล็อกเกอร์ จำนวน VM ระดับความปลอดภัย และระดับประสิทธิภาพการทำงาน
2. ทำการสุ่มสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ สำหรับพารามิเตอร์ของฟังก์ชันจุดประสงค์
3. ตรวจสอบข้อจำกัดของ constraint หรือไม่ ถ้าใช่ ทำการคำนวณหาค่าฟังก์ชันจุดประสงค์ ถ้าไม่ใช่ ทำการเพิ่มค่าจำนวนรอบการทำงาน และเช็คเงื่อนไขรอบการทำงาน
4. เมื่อผ่านการตรวจสอบข้อจำกัดของ constraint แล้ว ทำการคำนวณหาค่าฟังก์ชันจุดประสงค์สำหรับพารามิเตอร์ที่สุ่มได้
5. ทำการตรวจสอบค่าฟังก์ชันจุดประสงค์ที่ดีที่สุดในแต่ละรอบว่าเป็นค่าที่ดีที่สุดหรือไม่ ถ้าใช่ให้ทำการส่งค่าพารามิเตอร์และค่าฟังก์ชันจุดประสงค์นั้นกลับไป และเพิ่มจำนวนรอบการทำงานไปที่ละ 1 และทำซ้ำในขั้นตอนที่ 2 และ 3

6. ตรวจสอบว่าเงื่อนไขจำนวนรอบการทำงานครบตามจำนวนรอบหรือไม่ โดยกำหนดให้จำนวนรอบเท่ากับ 10,000 รอบ ถ้าทำงานครบจำนวนรอบให้สิ้นสุดการทำงาน และส่งค่าพารามิเตอร์ที่ได้กลับ แต่ถ้าทำงานไม่ครบตามจำนวนรอบให้ทำซ้ำขั้นตอน 2 – 6

แผนผังการทำงานในรูปแบบไม่มีข้อจำกัด



ภาพ 20 แสดงผังงานการทำงานของวิธีการที่เหมาะสมด้วยรูปแบบไม่มีข้อจำกัด

จากภาพ 20 ผังงานการทำงานของวิธีการที่เหมาะสมด้วยรูปแบบไม่มีข้อจำกัด โดยมีวิธีการทำงานดังนี้

1. เริ่มต้นด้วยการรับข้อมูลนำเข้าประกอบไปด้วย จำนวนล็อกเกอร์ จำนวน VM ระดับความปลอดภัย และระดับประสิทธิภาพการทำงาน
2. ทำการสุ่มสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ สำหรับพารามิเตอร์ของฟังก์ชันจุดประสงค์
3. ทำการคำนวณหาค่าฟังก์ชันจุดประสงค์สำหรับพารามิเตอร์ที่สุ่มได้ โดยมีการคำนวณของฟังก์ชันการปรับ
4. ทำการตรวจสอบค่าฟังก์ชันจุดประสงค์ที่ดีที่สุดในแต่ละรอบว่าเป็นค่าที่ดีที่สุดหรือไม่ ถ้าใช่ให้ทำการส่งค่าพารามิเตอร์และค่าฟังก์ชันจุดประสงค์นั้นกลับไป และเพิ่มจำนวนรอบการทำงานไปที่ละ 1 และทำซ้ำในขั้นตอนที่ 2 และ 3

5. ตรวจสอบว่าเงื่อนไขจำนวนรอบการทำงานครบตามจำนวนรอบหรือไม่ โดยกำหนดให้จำนวนรอบเท่ากับ 10,000 รอบ ถ้าทำงานครบจำนวนรอบให้สิ้นสุดการทำงาน และส่งค่าพารามิเตอร์ที่ได้กลับ แต่ถ้าทำงานไม่ครบตามจำนวนรอบให้ทำซ้ำขั้นตอน 2 – 5

4. การเปรียบเทียบระหว่างการทำงานแบบ constrained และแบบ unconstrained

โดยกำหนดรอบของการสุ่มวิธีการโดยการใช้ Random Search Method ด้วยการกำหนดจำนวนรอบของการสุ่มครั้งที่ 1 จำนวน 100 รอบ ครั้งที่ 2 จำนวน 1,000 รอบ ครั้งที่ 3 จำนวน 10,000 รอบ ครั้งที่ 4 จำนวน 100,000 รอบ และครั้งที่ 5 จำนวน 1,000,000 รอบ ตามลำดับ เพื่อหาวิธีการที่เหมาะสม โดยในแต่ละครั้งจะทำการทดลองจำนวน 10 รอบ โดยในการทดลองนั้นได้แบ่งข้อมูลที่ใช้สำหรับการทดลอง เป็น 2 กรณี ได้แก่ 1) กรณีที่ข้อจำกัดของค่าระดับความปลอดภัย และค่าระดับประสิทธิภาพที่ไม่ซับซ้อน 2) กรณีที่ข้อจำกัดของค่าระดับความปลอดภัย และค่าระดับประสิทธิภาพที่มีความซับซ้อน

1. กรณีที่ข้อจำกัดของค่าระดับความปลอดภัย และค่าระดับประสิทธิภาพที่ไม่ซับซ้อน โดยมีข้อมูลนำเข้าดังภาพ 21

Number of Loggers: 5
 Number of VMs: 5
 Security Level of VMs: [1, 1, 1, 2, 2]
 Performance Level of VMs: [1, 1, 1, 2, 2]

ภาพ 21 แสดงข้อมูลนำเข้ากรณีที่ไม่ซับซ้อน

ตาราง 6 แสดงการเปรียบเทียบการใช้ constrained และ unconstrained สำหรับข้อมูลไม่ซับซ้อน

จำนวนรอบของการสุ่ม	constrained	unconstrained
100	100%	100%
1,000	100%	100%
10,000	100%	100%
100,000	100%	100%
1,000,000	100%	100%
เฉลี่ย	100%	100%

2. กรณีที่ข้อจำกัดของค่าระดับความปลอดภัย และค่าระดับประสิทธิภาพที่มีความซับซ้อนโดยมีข้อมูลนำเข้าดังภาพ 22

Number of Loggers: 5
 Number of VMs: 5
 Security Level of VMs: [1, 2, 4, 4, 3]
 Performance Level of VMs: [1, 2, 3, 3, 3]

ภาพ 22 แสดงข้อมูลนำเข้ากรณีที่ซับซ้อน

ตาราง 7 แสดงการเปรียบเทียบการใช้ constrained และ unconstrained สำหรับข้อมูลซับซ้อน

จำนวนรอบของการสุ่ม	constrained	unconstrained
100	0%	100%
1,000	0%	100%
10,000	0%	100%
100,000	0%	100%
1,000,000	0%	100%
เฉลี่ย	0%	100%

วิธีการสร้างการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

ในกรณีที่มีผู้ให้บริการเดิมยกเลิกการใช้บริการ และมีผู้ให้บริการใหม่เข้ามาใช้บริการ ยกตัวอย่างจากผลลัพธ์ของการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่ใช้แบบ unconstrained โดยมีผลลัพธ์การจัดสรรระบบบันทึกเหตุการณ์ดังภาพ 23

	L ₁	L ₂	L ₃	L ₄	L ₅
U ₁	[1	0	0	0	0]
U ₂	[0	1	0	0	0]
U ₃	[0	1	0	0	0]
U ₄	[0	0	1	0	0]
U ₅	[0	0	1	0	0]

ภาพ 23 แสดงผลลัพธ์การจัดสรรระบบบันทึกเหตุการณ์

จากภาพ 23 กำหนดให้คอลัมน์ (L_1 ถึง L_5) หมายถึง ล็อกเกอร์ที่พร้อมใช้งานตัวที่ 1 – 5 โดยสามารถกำหนดให้อยู่ในรูปของ L_x โดยที่ x คือ 1, 2, ..., 5 และแถว (U_1 ถึง U_5) หมายถึง ผู้ใช้บริการจำนวน 5 คนที่ได้ทำการเช่า VM กำหนดให้อยู่ในรูปของ U_y โดยที่ y คือ 1, 2, ..., 5 ซึ่งหมายถึง VM1 ถึง VM5 ตามลำดับ

สมมติในกรณีผู้ใช้งาน VM4 (U_4) ต้องการยกเลิกการใช้บริการ ทางผู้ให้บริการทำงานยกเลิกข้อมูลออก และผู้ให้บริการรายใหม่ (U_{new} หรือ U_{new}) จะถูกกำหนดค่าให้เป็น '1' ดังภาพ 24

	L_1	L_2	L_3	L_4	L_5
U_1	[1	0	0	0	0]
U_2	[0	1	0	0	0]
U_3	[0	1	0	0	0]
U_5	[0	0	1	0	0]
U_{new}	[1	1	1	1	1]

ภาพ 24 แสดงเพิ่มผู้ให้บริการรายใหม่ โดยกำหนดค่าให้เป็น 1

ดังนั้นจะได้

$$X' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{5 \times 5}$$

ผู้วิจัยดำเนินการทดลองโดยแบ่งออกเป็น 2 กรณี ได้แก่ กรณีที่ไม่เปลี่ยนล็อกเกอร์ หมายถึง การที่ VM ยังอยู่ในล็อกเกอร์ตัวเดิม และกรณีที่เปลี่ยนล็อกเกอร์ หมายถึง การที่ VM เปลี่ยนจาก ล็อกเกอร์หนึ่งไปยังอีกล็อกเกอร์หนึ่ง เช่น VM1 เดิมอยู่ในล็อกเกอร์ L_1 แต่เมื่อมีการหาวิธีการใหม่เพื่อมารองรับกับผู้ให้บริการใหม่ แล้ว VM1 ถูกย้ายไปอยู่ในล็อกเกอร์ L_2 ในกรณีนี้ถือว่าเป็นการเปลี่ยนล็อกเกอร์

1. กรณีที่ 1 ไม่เปลี่ยนล็อกเกอร์

สมมติให้การจัดสรรระบบบันทึกเหตุการณ์สำหรับ U_{new} อยู่ในล็อกเกอร์ L_4 และ VMs ไม่มีการเปลี่ยนล็อกเกอร์ จะได้

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5}$$

ดังนั้นผู้วิจัยทำการคูณระหว่าง X' (ค่าของ X เดิม) กับ X โดยใช้วิธีการคูณแบบสมาชิก (elementwise) ภายในเมทริกซ์ เนื่องจากถ้าไม่มีการเปลี่ยนล็อกเกอร์ค่าที่ได้จากผลคูณจะมีค่าเท่ากับ 1 แต่ถ้ามีการเปลี่ยนล็อกเกอร์ไปจากเดิมผลคูณจะมีค่าเท่ากับ 0

$$X' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{5 \times 5} \times X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5}$$

เมื่อได้ผลคูณระหว่าง X' กับ X แล้ว ทำการหาค่าผลรวมของแต่ละ VM จากสมการที่ (7) เพื่อตรวจสอบว่ามีการเปลี่ยนล็อกเกอร์หรือไม่

$$M_i = \sum_{j=1}^n x'_{ij} \times x_{ij} \quad (7)$$

โดยที่

M_i คือ จำนวนของผลคูณของวิธีการจัดสรรรูปแบบเก่ากับวิธีการจัดสรรรูปแบบใหม่ของแต่ละ VM

x'_{ij} คือ ค่าที่อยู่ในเมทริกซ์ของการจัดสรรรูปแบบเก่า ในตำแหน่งที่ i, j

x_{ij} คือ ค่าที่อยู่ในเมทริกซ์ของการจัดสรรรูปแบบใหม่ในตำแหน่งที่ i, j จะได้

$$M = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

จะเห็นว่าในกรณีนี้ค่าผลรวมมีค่าเท่ากับ 1 ทุกตัว นั้นหมายความว่าไม่มีการเปลี่ยนล็อกเกอร์เกิดขึ้น เมื่อไม่มีการเปลี่ยนล็อกเกอร์เกิดขึ้น ผู้วิจัยจึงได้ทำการนำ 1 ไปลบค่าผลรวมของแต่ละ VM ออก เพราะค่าที่ได้จะเป็น 0 ซึ่งหมายความว่าจะไม่มีการโดนลงโทษด้วยฟังก์ชันการปรับ

สำหรับในกรณีการลงโทษในส่วนของการเปลี่ยนบล็อกเกอร์นั้น ผู้วิจัยได้กำหนดด้วยตัวแปร g_c โดยมีค่าน้ำหนักเท่ากับ 0.5 เนื่องจากเป็นกรณีที่ไม่ร้ายแรงสำหรับการจัดสรรระบบบันทึกเหตุการณ์ จึงได้สมการในการคิดค่าสำหรับการเปลี่ยนบล็อกเกอร์ ดังสมการที่ (8)

$$Scost(U_i) = \sum_{i=1}^m (1 - M_i) * g_c \quad (8)$$

$Scost(U_i)$ คือ ค่าลงโทษในกรณีที่มีการเปลี่ยนบล็อกเกอร์สำหรับ U_i

g_c คือ ค่าถ่วงน้ำหนักในกรณีที่มีการเปลี่ยนบล็อกเกอร์

เพราะฉะนั้น

$$Scost = (((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5))$$

$$Scost = (0 + 0 + 0 + 0 + 0) = 0$$

2. กรณีที่ 2 เปลี่ยนบล็อกเกอร์

สมมติให้การจัดสรรระบบบันทึกเหตุการณ์สำหรับ VM_{new} อยู่ในบล็อกเกอร์ L4 และ VM1 มีการเปลี่ยนบล็อกเกอร์ จะได้

$$X = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5}$$

ดังนั้นผู้วิจัยทำการคูณระหว่าง X' (ค่าของ X เดิม) กับ X โดยใช้วิธีการคูณแบบสมาชิก (Elementwise) ภายในเมทริกซ์ เนื่องจากถ้าไม่มีการเปลี่ยนบล็อกเกอร์ค่าที่ได้จากผลคูณจะมีค่าเท่ากับ 1 แต่ถ้ามีการเปลี่ยนบล็อกเกอร์ไปจากเดิมผลคูณจะมีค่าเท่ากับ 0

$$X' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{5 \times 5} \times X = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{5 \times 5}$$

เมื่อได้ผลคูณระหว่าง X' กับ X แล้ว ทำการหาค่าผลรวมของแต่ละ VM จากสมการที่ (7) เพื่อตรวจสอบว่ามีการเปลี่ยนบล็อกเกอร์หรือไม่

$$M = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

จะเห็นว่าในกรณีนี้ค่าผลรวมของ VM1 มีค่าเท่ากับ 0 นั้นหมายความว่ามีการเปลี่ยนสล็อตเกอร์เกิดขึ้น ส่วน VMs ตัวอื่น จะไม่มีการเปลี่ยนสล็อตเกอร์เนื่องจากมีค่าผลรวมเป็น 1 ผู้วิจัยจึงได้ทำการนำ 1 ไปลบค่าผลรวมของแต่ละ VM ออก จะเห็นได้ว่า เมื่อนำ 1 ไปลบผลรวมของ VM1 ซึ่งมีค่าเท่ากับ 0 นั้นหมายความว่าในกรณีนี้จะได้รับบทลงโทษ โดยการลดค่าไป 0.5 ดังนั้นสามารถคำนวณหาค่า *Scost* ได้ดังนี้

$$\begin{aligned} Scost &= ((1 - 0) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ &\quad + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ Scost &= (0.5 + 0 + 0 + 0 + 0) = 0.5 \end{aligned}$$

จากสมการ (6) ในการหาฟังก์ชันจุดประสงค์ของการทดลองการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบนที่เหตุการณ์บนบน IaaS คลาวด์ จะได้ฟังก์ชันจุดประสงค์ของการปรับการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการในการจัดสรรระบบบนที่เหตุการณ์บน IaaS คลาวด์ ดังสมการ (9)

$$\text{maximize } f(X) = \left(\sum_{j=1}^n O_j(X)\right) + \sum_{i=1}^m \text{cost}(U_i) + \sum_{i=1}^m \text{Scost}(U_i) \quad (9)$$

subject to

X คือ เมทริกซ์ขนาด $m \times n$

$$x_{ij} \in \{0,1\} \text{ สำหรับแต่ละ } i \in \{1,2, \dots, m\}, j \in \{1,2, \dots, n\}$$

$$\sum_{j=1}^n x_{ij}, \quad \text{สำหรับแต่ละ } j \in \{1,2, \dots, n\}$$

สำหรับแต่ละ $i \in \{1,2, \dots, m\}$ สามารถกำหนด cost function ได้ตามสมการ

$$\text{cost}(U_i) = \begin{cases} g_p(N_k - P_i), & \text{if } N_k < P_i \\ 0, & \text{if } P_i \leq N_k \leq S_i \\ g_s(S_i - N_k), & \text{if } N_k > S_i \end{cases}$$

โดยที่

$$U_i = [x_{i1}, x_{i2}, \dots, x_{ik}]$$

และ

$$x_{ik} = 1 \text{ สำหรับบาง } k \in \{1, 2, \dots, n\}$$

$$\text{ซึ่ง } N(L_j) = N_k$$

และสำหรับแต่ละ $i \in \{1, 2, \dots, m\}$ สามารถกำหนด Scost function ได้ตามสมการ

$$\text{Scost}(U_i) = \sum_{i=1}^m (1 - M_i) * g_c$$

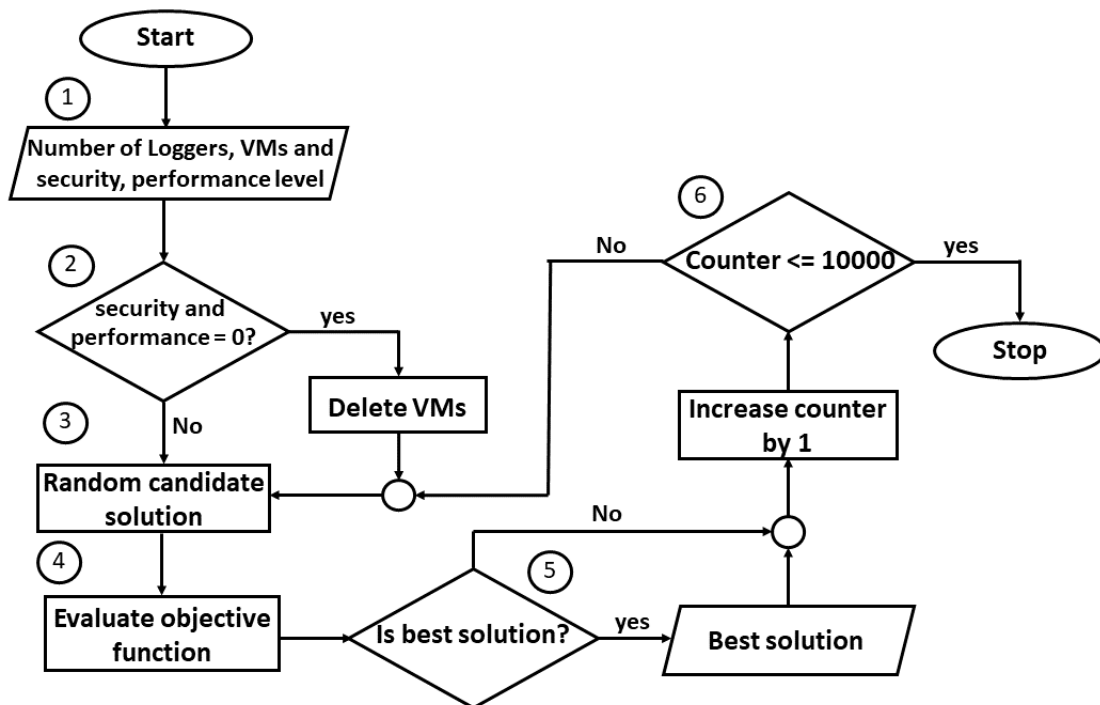
โดยที่

$$U_i = [x_{i1}, x_{i2}, \dots, x_{ik}] \text{ และ } M_i \in \{0,1\}$$

3. การสร้างผังงานของการทำงานโมเดลทางคณิตศาสตร์การสร้างการปรับรูปแบบ

การสร้างผังงานการทำงานของโมเดลทางคณิตศาสตร์ที่ได้สร้างขึ้น ผู้วิจัยได้กำหนดใน ส่วนของข้อมูลนำเข้าที่ประกอบไปด้วยส่วนของผู้ให้บริการ ได้แก่ จำนวนลือกเกอร์ และจำนวน VM ที่สามารถใช้งานได้ และในส่วนของผู้ใช้บริการ ได้แก่ ระดับความปลอดภัย และระดับประสิทธิภาพการทำงาน ที่มีการปรับเปลี่ยนผู้ใช้บริการจากการจัดสรรในรูปแบบเดิม และการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ผู้วิจัยได้ใช้วิธีการ Random Search ในการสร้างวิธีการที่เป็นไปได้ โดยมีการสุ่มค่าความน่าจะเป็นในการเปลี่ยนตำแหน่งของลือกเกอร์เข้ามา โดยกำหนดค่าความน่าจะเป็นโดยค่ามากกว่าร้อยละ 20 จะใช้วิธีการจัดสรรในรูปแบบเดิม แต่ถ้าน้อยกว่าร้อยละ 20 จะทำการ สลับลือกเกอร์ แล้วนำไปเขียนโปรแกรมการทำงานด้วยภาษา Python

แผนผังการทำงานการสร้างการปรับการปรับรูปแบบ



ภาพ 25 แสดงผังการทำงานของการสร้างการปรับรูปแบบ

จากภาพ 25 ผังงานการทำงานของการหาวิธีการที่เหมาะสมด้วยการสร้างการปรับการปรับรูปแบบ โดยมีวิธีการทำงานดังนี้

1. เริ่มต้นด้วยการรับข้อมูลนำเข้าประกอบไปด้วย จำนวนล็อกเกอร์ จำนวน VM ระดับความปลอดภัย และระดับประสิทธิภาพการทำงาน ที่มีการปรับเปลี่ยนผู้ใช้บริการ
2. ทำการตรวจสอบว่ามีการยกเลิกการใช้บริการหรือไม่ โดยถ้ามีจะทำการลบ VM นั้น
3. ทำการสุ่มสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ สำหรับพารามิเตอร์ของฟังก์ชันจุดประสงค์ โดยกำหนดค่าความน่าจะเป็นสำหรับการเปลี่ยนล็อกเกอร์
4. ทำการคำนวณหาค่าฟังก์ชันจุดประสงค์ โดยมีการคำนวณของฟังก์ชันการปรับของการผิดข้อจำกัดระดับความปลอดภัยและระดับประสิทธิภาพ และการเปลี่ยนล็อกเกอร์โดยทำการคูณแบบสมาชิกภายในเมทริกซ์
5. ทำการตรวจสอบค่าฟังก์ชันจุดประสงค์ที่ดีที่สุดในแต่ละรอบว่าเป็นค่าที่ดีที่สุดหรือไม่ ถ้าใช่ให้ทำการส่งค่าพารามิเตอร์และค่าฟังก์ชันจุดประสงค์นั้นกลับไป และเพิ่มจำนวนรอบการทำงานไปที่ละ 1 และทำซ้ำในขั้นตอนที่ 3 และ 4

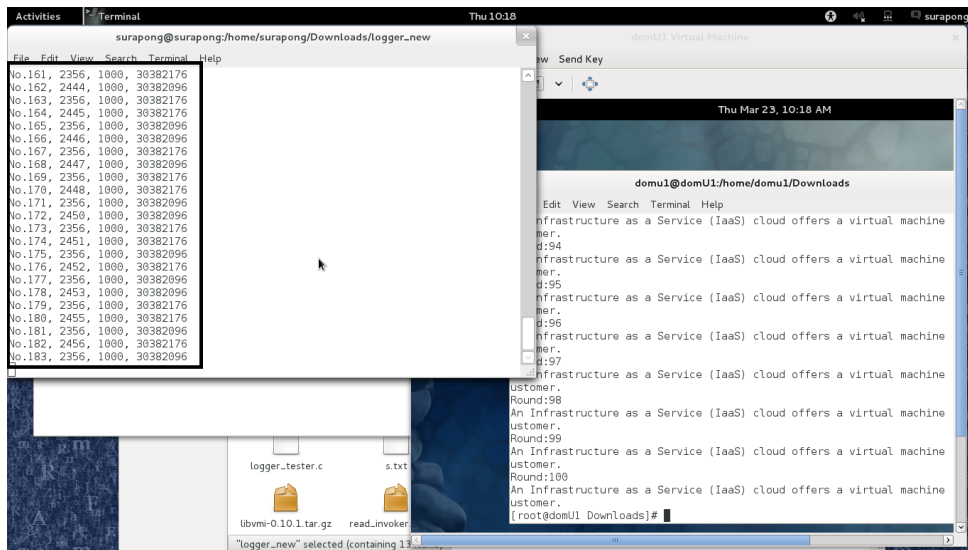
6. ตรวจสอบว่าเงื่อนไขจำนวนรอบการทำงานครบตามจำนวนรอบหรือไม่ โดยกำหนดให้จำนวนรอบเท่ากับ 10,000 รอบ ถ้าทำงานครบจำนวนรอบให้สิ้นสุดการทำงาน และส่งค่าพารามิเตอร์ที่ได้กลับ แต่ถ้าทำงานไม่ครบตามจำนวนรอบให้ทำซ้ำขั้นตอน 2 – 5



1
2
3

[root@domu1 domu1]#
./invoke_cat
1000
100

ภาพ 27 แสดงการทำงานของแอปพลิเคชัน read อ่านไฟล์ s.txt

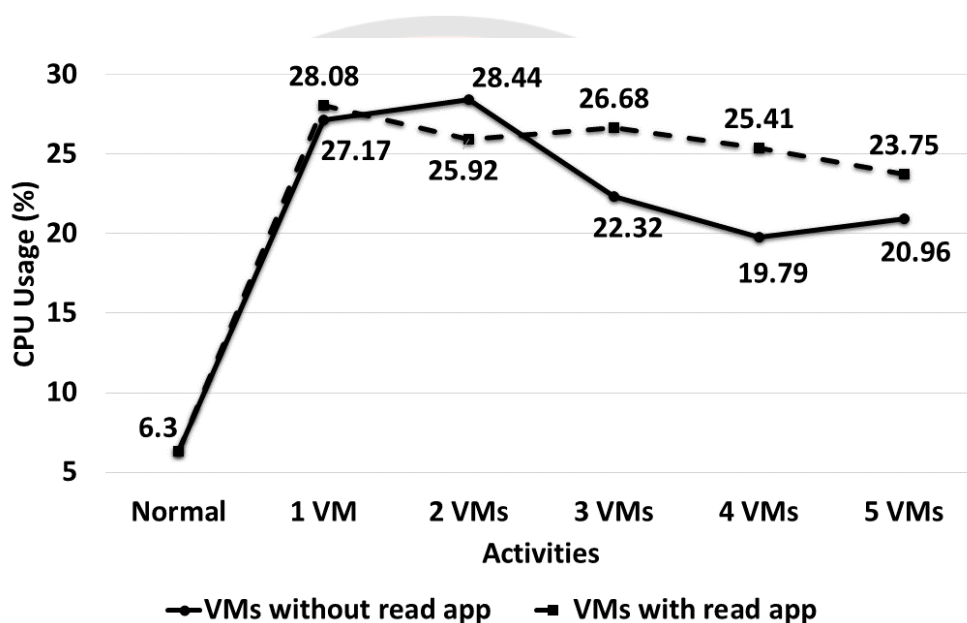


ภาพ 28 แสดงการรันแอปพลิเคชัน read บนเครื่อง VM พร้อมด้วยการรันแอปพลิเคชัน logger

จากภาพ 26 แสดงการทำงานของแอปพลิเคชันล็อกเกอร์ ในขณะที่กำลังทำงานในฝั่ง Host ดังที่ได้อธิบายในบทที่ 2 หัวข้อการทำงานของระบบบันทึกเหตุการณ์บน IaaS คลาวด์ โดยเรียกคำสั่งให้รันแอปพลิเคชันล็อกเกอร์ ชื่อว่า pro_list ในกรอบสี่เหลี่ยมหมายเลข 1 ในกรอบสี่เหลี่ยมหมายเลข 2 domU1 คือ ชื่อของ VM ที่ต้องการเข้าไปตรวจสอบโปรเซสการทำงาน กรอบสี่เหลี่ยมหมายเลข 3 invoke_cat คือ โปรเซสที่ต้องการตรวจสอบที่อยู่ใน VM ชื่อว่า domU1 และกรอบสี่เหลี่ยมหมายเลข 4 คือการหนดค่า sleeping time โดยกำหนดให้เป็น 0 μ s. (sleeping time คือ ค่าที่กำหนดให้กับล็อกเกอร์ในการกำหนดระยะเวลาที่จะเริ่มเข้าไปทำการตรวจสอบโปรเซสการทำงานใน VM รอบใหม่) แอปพลิเคชันล็อกเกอร์จะทำการตรวจสอบ invoke_cat_mem จนกว่าจะพบสำหรับโปรเซส invoke_cat จะปรากฏใน invoke_cat_mem ได้นั้น เครื่อง VM จะต้องทำการรันโปรเซสที่ชื่อว่า invoke_cat ก่อน (ซึ่งในโปรเซส invoke_cat จะเป็นการทำงานของแอปพลิเคชัน read ดังที่ได้กล่าวไว้ในบทที่ 2 หัวข้อการทำงานของระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ดังปรากฏในภาพ 27 กรอบสี่เหลี่ยมหมายเลข 1 คือการที่เครื่อง VM สั่งให้รันโปรเซสที่ชื่อว่า invoke_cat ซึ่งภายในการทำงานของโปรเซสนี้จะเป็นการอ่าน (read) ไฟล์ที่ชื่อว่า s.txt กรอบสี่เหลี่ยมหมายเลข 2 คือ พารามิเตอร์สำหรับกำหนดระยะเวลาในการทำงานของโปรเซส invoke_cat

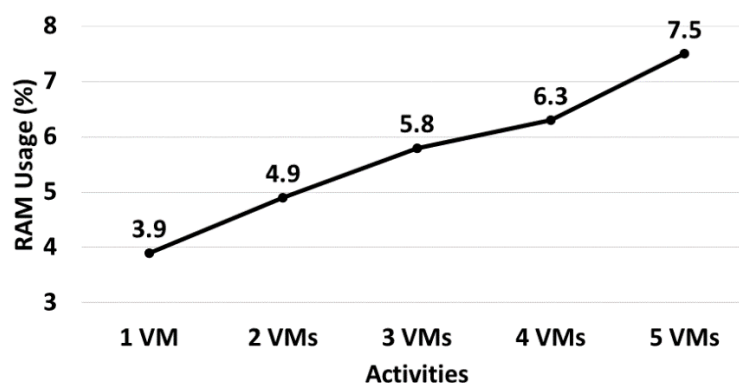
ในแต่ละรอบ และกรอบสี่เหลี่ยมหมายเลข 3 คือ จำนวนรอบการทำงานของการอ่านไฟล์ s.txt และในภาพ 28 แสดงผลลัพธ์ที่ได้จากการดึงค่าจาก memU มาแสดงในส่วนของ Host ในกรอบสี่เหลี่ยมประกอบไปด้วย หมายเลขในการดึงข้อมูล (counter), หมายเลขโปรเซส (pid), หมายเลขผู้ใช้ (uid) และชื่อโปรเซส (procname)

ผลการทดสอบปริมาณการใช้งาน CPU และ RAM ของระบบบันทึกเหตุการณ์บนเครื่องคอมพิวเตอร์ของผู้ให้บริการ



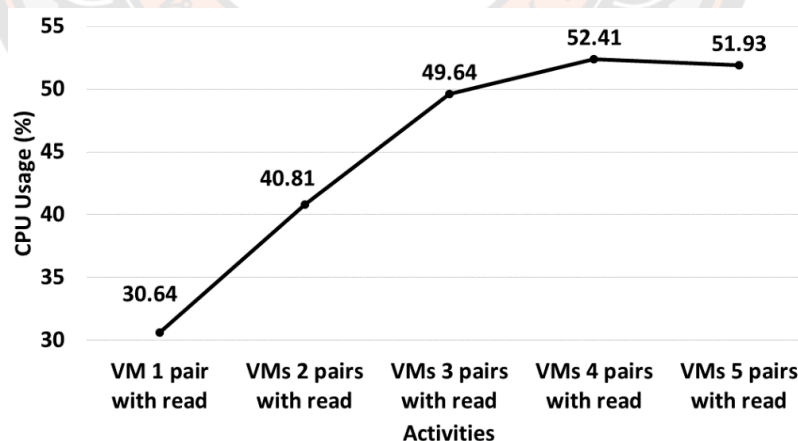
ภาพ 29 แสดงผลการใช้งานของ CPU (%) ของโฮสต์ในกรณีที่ VMs มีการทำงานแบบไม่รันแอปพลิเคชัน read กับ VMs รันแอปพลิเคชัน read

จากภาพ 29 เส้นในแนวแกน X คือ จำนวนการรัน VM และเส้นในแนวแกน Y คือ ปริมาณการใช้งาน CPU ฝั่ง Host มีหน่วยเป็นเปอร์เซ็นต์ กราฟเส้นตรงแสดงค่าปริมาณการใช้งาน CPU ในกรณีที่ไม่มีรันแอปพลิเคชัน read และกราฟเส้นปะแสดงค่าปริมาณการใช้งาน CPU ในกรณีที่รันแอปพลิเคชัน read จากกราฟจะเห็นได้ว่าเมื่อมีการรันแอปพลิเคชัน read กับปริมาณ VM ที่เพิ่มขึ้นจาก 1 ตัว เป็น 2 ตัวพร้อมกัน จนถึง 5 ตัวพร้อมกัน ตามลำดับ ปริมาณการใช้งาน CPU จะมีปริมาณการใช้งานที่ใกล้เคียงกันระหว่างที่ VM กรณีที่ไม่รันแอปพลิเคชัน read ซึ่งในการรันแอปพลิเคชัน read ที่จำลองการทำงานของเครื่อง VM นั้น จะมีปริมาณการใช้งาน CPU เพิ่มขึ้นโดยเฉลี่ย 2.23% เท่านั้น แสดงให้เห็นว่าในสภาวะการทำงานของเครื่อง VM ที่มีการรันโปรแกรมการทำงานทำด้วยแอปพลิเคชัน read ที่จำลองขึ้นมา นั้น จะมีปริมาณการใช้งาน CPU ที่ไม่สูงมาก



ภาพ 30 แสดงผลการใช้งานของ RAM (%) ของโฮสต์ในกรณีที่ VM มีการรันแอปพลิเคชัน read

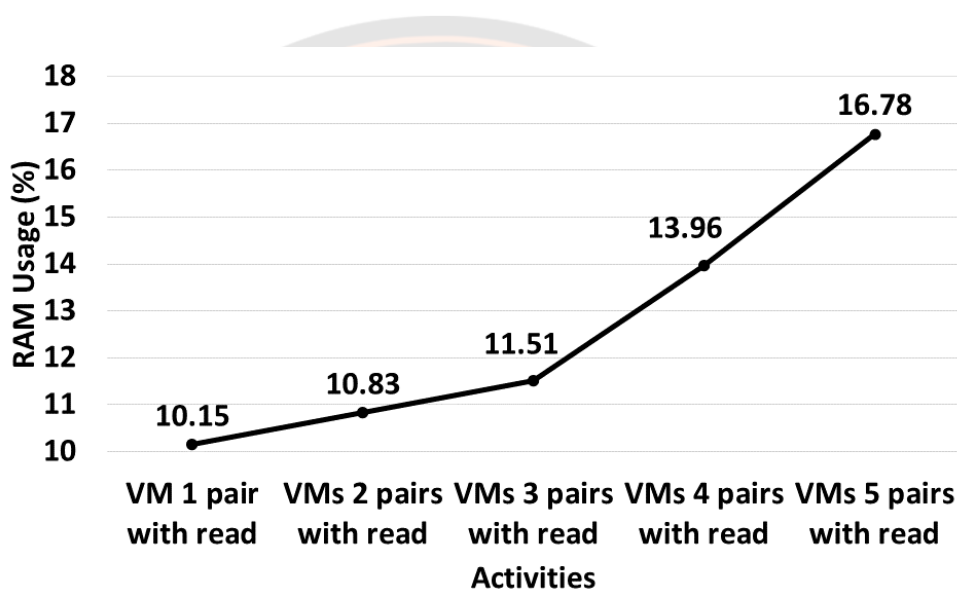
จากภาพ 30 เส้นในแนวแกน X คือ จำนวน VM และเส้นในแนวแกน Y คือ ปริมาณการใช้งาน RAM ฝั่ง Host มีหน่วยเป็นเปอร์เซ็นต์ จากกราฟเส้นตรงเป็นปริมาณการใช้งาน RAM ซึ่งจะเห็นว่ามีการทำงานของแอปพลิเคชัน read ที่ VM 1 VM ปริมาณการใช้งาน RAM อยู่ที่ 3.90% เมื่อมีการทำงานของแอปพลิเคชัน read ที่ VM 2 VMs ปริมาณการใช้งาน RAM อยู่ที่ 4.90% เมื่อมีการทำงานของแอปพลิเคชัน read ที่ VM 3 VMs ปริมาณการใช้งาน RAM อยู่ที่ 5.80% เมื่อมีการทำงานของแอปพลิเคชัน read ที่ VM 4 VMs ปริมาณการใช้งาน RAM อยู่ที่ 6.30% และเมื่อมีการทำงานของแอปพลิเคชัน read ที่ VM 5 VMs ปริมาณการใช้งาน RAM อยู่ที่ 7.50%



ภาพ 31 แสดงผลการใช้งานของ CPU (%) ของโฮสต์ในกรณีที่โฮสต์มีการรันแอปพลิเคชัน Logger และ VMs มีการรันแอปพลิเคชัน read

จากภาพ 31 เส้นในแนวแกน X คือ จำนวนคู่ในการทำงานของ logger กับ VM กับ และเส้นในแนวแกน Y คือ ปริมาณการใช้งาน CPU ฝั่ง Host มีหน่วยเป็นเปอร์เซ็นต์ จากกราฟเส้นตรงเป็น

ปริมาณการใช้งาน CPU ซึ่งจะเห็นว่ามีการทำงานของ Logger กับ VM จำนวน 1 คู่ ปริมาณการใช้งาน CPU อยู่ที่ 30.64% เมื่อมีการทำงานของ Logger กับ VM จำนวน 2 คู่ ปริมาณการใช้งาน CPU อยู่ที่ 40.81% เมื่อมีการทำงานของ Logger กับ VM จำนวน 3 คู่ ปริมาณการใช้งาน CPU อยู่ที่ 49.64% เมื่อเปิดการใช้งานพร้อมกันจำนวน 5 คู่ จะทำให้เครื่องของ VM และ Host มีการทำงานที่ช้าลง ซึ่งแสดงให้เห็นว่าล็อกเกอร์ที่ใช้ในการตรวจจับโปรเซสการทำงานในเครื่อง VM นั้นมีการใช้ปริมาณ CPU ที่มาก จึงทำให้เครื่องคอมพิวเตอร์สามารถใช้งาน VM ในสถานะที่ทำงานได้เพียง 5 เครื่องเท่านั้นเมื่อเทียบกับสเปกของเครื่องคอมพิวเตอร์ที่ใช้ในการทดลอง



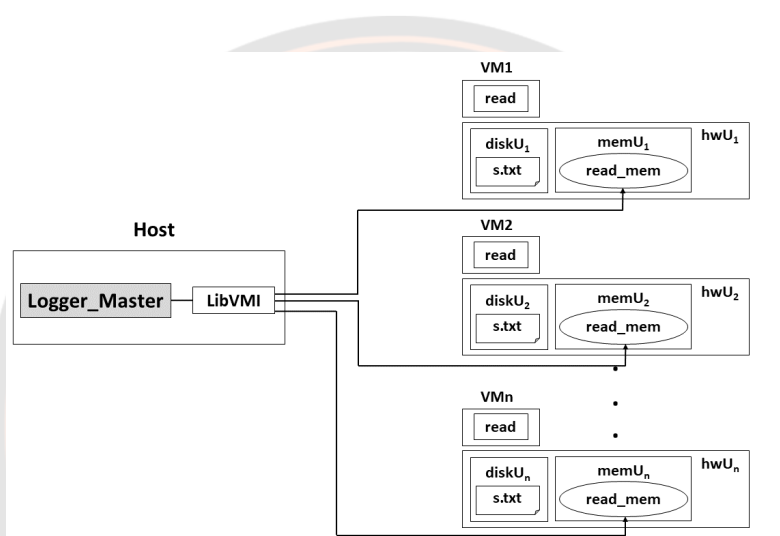
ภาพ 32 แสดงผลการใช้งานของ RAM (%) ของโฮสต์ในกรณีที่โฮสต์มีการรันแอปพลิเคชัน Logger และ VMs มีการรันแอปพลิเคชัน read

จากภาพ 32 เส้นในแนวแกน X คือ จำนวนคู่ในการทำงานของ logger กับ VM และเส้นในแนวแกน Y คือ ปริมาณการใช้งาน RAM ฝั่งโฮสต์มีหน่วยเป็นเปอร์เซ็นต์ จากกราฟเส้นตรงเป็นปริมาณการใช้งาน RAM ซึ่งจะเห็นว่ามีการทำงานของ Logger กับ VM จำนวน 1 คู่ ปริมาณการใช้งาน RAM อยู่ที่ 10.15% เมื่อมีการทำงานของ Logger กับ VM จำนวน 2 คู่ ปริมาณการใช้งาน RAM อยู่ที่ 10.83% เมื่อมีการทำงานของ Logger กับ VM จำนวน 3 คู่ ปริมาณการใช้งาน RAM อยู่ที่ 11.51% เมื่อเปิดการใช้งานพร้อมกันจำนวน 5 คู่ จะทำให้เครื่องของ VM และ Host มีการทำงานที่ช้าลง

ผลการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์

เมื่อผู้วิจัยได้ทำการวัดปริมาณการใช้งาน CPU และ RAM ในหัวข้อผลการทดสอบปริมาณการใช้งาน CPU และ RAM ของระบบบันทึกเหตุการณ์บนเครื่องคอมพิวเตอร์ของผู้ให้บริการ แล้วนั้น

พบว่า เมื่อมีการใช้งานแอปพลิเคชันล็อกเกอร์เพิ่มขึ้น ปริมาณการใช้งาน CPU และ RAM ก็จะมีมากขึ้นด้วย จากการทดลองพบว่าปริมาณการใช้งาน CPU ส่งผลต่อการทำงานในการตรวจจับโปรเซสการทำงานโดยล็อกเกอร์ช้าลง จนทำให้เครื่องคอมพิวเตอร์ที่ใช้ในการทดลองมีปัญหาไม่สามารถใช้งานได้ ผู้วิจัยจึงได้สร้างรูปแบบการจัดการระบบบันทึกเหตุการณ์แบบรวมศูนย์ที่ล็อกเกอร์ 1 ตัวสามารถตรวจจับโปรเซสการทำงานใน VM ได้มากกว่า 1 เครื่อง ดังภาพ 33 และนำไปสู่การสร้างรูปแบบการจัดการระบบบันทึกเหตุการณ์ที่มีความหลากหลายเหมาะสมกับผู้ให้บริการและผู้ใช้บริการ ดังภาพ 33



ภาพ 33 แสดงการทำงานของ Logger แบบรวมศูนย์

จากภาพ 33 ในกล่องด้านซ้ายมือส่วนของ Host ประกอบไปด้วย Logger_Master 1 ล็อกเกอร์และ LibVMI ส่วนในกล่องทางด้านขวามือจะประกอบไปด้วยส่วนประกอบของ VM1 ตัวที่ 1 ถึงตัวที่ n การทำการทดลองนั้นจะทำการทดลองแบบเพิ่มจำนวน VM ไปทีละ 1 VM จนถึง n VMs ตามลำดับ

Logger #1	Logger #2	Logger #3	Logger #4	Logger #5	
1	1	1	1	1	Solution 1
1	1	1	2		Solution 2
1	1	3			Solution 3
1	4				Solution 4
5					Solution 5

ภาพ 34 แสดงวิธีการจัดสรรล็อกเกอร์ให้กับผู้ใช้บริการ และผู้ให้บริการ

จากภาพ 34 กล่องสีขาวยกแสดงแสดงถึงล็อกเกอร์ และหมายเลขที่อยู่ในกล่องสีขาวยก หมายถึง จำนวนของ VM ที่ทำงานกับล็อกเกอร์นั้น โดยจำลองถึงผู้ใช้บริการจำนวน 5 คน ในกรณีการจัดสรรการทำงานในรูปแบบที่ 1 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความถูกต้องในการตรวจจับโปเรสโดยล็อกเกอร์ ทั้งหมด 5 คน จะมีการทำงานของล็อกเกอร์ แบบ 1 เครื่อง ต่อ 1 VM การจัดสรรการทำงานในรูปแบบที่ 2 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความถูกต้องในการตรวจจับโปเรสโดยล็อกเกอร์ จำนวน 3 คน และต้องการประสิทธิภาพในการทำงานของ VM จำนวน 2 คน การจัดสรรการทำงานในรูปแบบที่ 3 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความถูกต้องในการตรวจจับโปเรสโดยล็อกเกอร์ จำนวน 2 คน และต้องการประสิทธิภาพในการทำงานของ VM จำนวน 3 คน การจัดสรรการทำงานในรูปแบบที่ 4 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการความถูกต้องในการตรวจจับโปเรสโดยล็อกเกอร์ จำนวน 1 คน และต้องการประสิทธิภาพในการทำงานของ VM จำนวน 4 คน และการจัดสรรการทำงานในรูปแบบที่ 5 หมายถึง การทำงานในลักษณะที่ผู้ใช้บริการต้องการประสิทธิภาพในการทำงานของ VM จำนวน 5 คน

ผลการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

เมื่อได้รูปแบบแนวทางการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ในภาพ 34 แล้วนั้นผู้วิจัยได้ทำการสร้างโมเดลทางคณิตศาสตร์ในการจัดสรรระบบบันทึกเหตุการณ์เพื่อให้มีการใช้งานล็อกเกอร์ให้น้อยที่สุด เพื่อลดปริมาณการใช้งาน CPU ของเครื่องผู้ให้บริการ โดยได้ยึดความต้องการของผู้ใช้บริการในการมองในเรื่องของความปลอดภัยและประสิทธิภาพในการทำงาน ผู้วิจัยจึงขอเสนอผลลัพธ์ที่ได้จากการสร้างโมเดลทางคณิตศาสตร์ในการจัดสรรระบบบันทึกเหตุการณ์ ใน 2

รูปแบบได้แก่ 1) โมเดลทางคณิตศาสตร์แบบมีข้อจำกัด (Constrained Optimization) และ 2) โมเดลทางคณิตศาสตร์แบบไม่มีข้อจำกัด (Unconstrained Optimization) โดยการใช้ฟังก์ชันการปรับ (Penalty Function) โดยสามารถอธิบายรายละเอียดได้ดังนี้

1. โมเดลทางคณิตศาสตร์แบบมีข้อจำกัด (Constraint Optimization)

1.1 ตัวแปรการตัดสินใจ (Decision Variable)

กำหนดให้ X แทนวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ซึ่งอยู่ในรูปของเมทริกซ์ขนาด $M \times N$ (5×5) โดยกำหนดให้คอลัมน์แทนล็อกเกอร์ตัวที่ 1 – 5 และให้แถวแทน VM ตัวที่ 1 – 5 ตามลำดับ และหมายเลข '1' แทนการแสดงว่า VM1 อยู่ภายใต้การทำงานของล็อกเกอร์ 1 หมายเลข '0' แทนการแสดงว่าไม่มีการทำงานจากล็อกเกอร์ ซึ่ง VM 1 ตัว สามารถอยู่ภายใต้การทำงานของล็อกเกอร์ได้เพียงตัวเดียวเท่านั้น (เรียกว่าข้อจำกัดแบบแฝง (Constrained Parameters))

$$\therefore X = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix}_{5 \times 5}$$

X เป็นเวกเตอร์ที่แทนวิธีการจัดสรรระบบบันทึกเหตุการณ์

โดยวัตถุประสงค์ของการทำสร้างวิธีการจัดสรรระบบบันทึกเหตุการณ์นั้นจะต้องให้เหลือจำนวนล็อกเกอร์ที่ไม่ได้มีภาระงานมากที่สุด เพื่อจะทำให้ CPU ของเครื่องคอมพิวเตอร์ฝั่งผู้ให้บริการใช้งานลดลง จึงต้องทำการนับจำนวน VM ในล็อกเกอร์แต่ละตัวว่ามีจำนวนกี่เครื่อง และใช้ล็อกเกอร์ไปทั้งหมดเท่าไร จึงได้ทำการออกแบบตัวแปรการตัดสินใจ X ดังสมการ (2)

1.2 ฟังก์ชันจุดประสงค์ (Objective Function)

จะได้ฟังก์ชันจุดประสงค์ที่เป็นค่ามากที่สุดของผลรวมของจำนวนล็อกเกอร์ที่ไม่ได้ใช้งาน ในรูปแบบโมเดลทางคณิตศาสตร์ ได้ดังสมการ (3)

ฟังก์ชันจุดประสงค์ที่ได้นั้นเป็นค่าสูงสุด (Maximize) เนื่องจากวัตถุประสงค์ต้องการให้เหลือจำนวนล็อกเกอร์ที่ไม่ได้มีภาระงานมากที่สุด เพื่อลดปริมาณการใช้งาน CPU

1.3 ระบุข้อจำกัดต่าง ๆ (Constraint) จากระดับความต้องการทั้งระดับความปลอดภัยและประสิทธิภาพในการทำงานของเครื่อง VM จึงได้ข้อจำกัดในการทำงานของการจัดสรรระบบบันทึกเหตุการณ์ โดยจำนวน VM ในล็อกเกอร์น้อยกว่า ระดับความปลอดภัย และจำนวน VM ใน

ลือกเกอร์มากกว่า ระดับประสิทธิภาพ ซึ่งระดับความปลอดภัยแปรผกผันกับระดับประสิทธิภาพ และระดับประสิทธิภาพแปรผกผันกับจำนวนลือกเกอร์ จึงได้ข้อจำกัด ได้ตั้งสมการ (4)

จากการกำหนดข้อจำกัดความต้องการทั้งระดับความปลอดภัย และประสิทธิภาพ จึงได้ฟังก์ชันจุดประสงค์ ในรูปแบบโมเดลทางคณิตศาสตร์ ได้ตั้งสมการ (5)

โมเดลทางคณิตศาสตร์แบบไม่มีข้อจำกัด (Unconstrained Optimization)

ในกรณีโมเดลทางคณิตศาสตร์ในรูปแบบมีข้อจำกัดนั้น ในบางกรณีจะไม่สามารถหาวิธีการในการจัดสรรระบบบนที่เหตุการณ์ได้ ทางผู้วิจัยจึงได้ทำการออกแบบโมเดลทางคณิตศาสตร์แบบไม่มีข้อจำกัดโดยการใช้ฟังก์ชันการปรับ (Penalty) มาเป็นบทลงโทษสำหรับกรณีที่ไม่เป็นไปตามข้อจำกัด ดังนี้

$$cost(U_i) = \begin{cases} g_p(N_k - P_i), & \text{if } N_k < P_i \\ 0, & \text{if } P_i \leq N_k \leq S_i \\ g_s(S_i - N_k), & \text{if } N_k > S_i \end{cases}$$

ทั้งนี้เพื่อให้ผู้ใช้บริการจัดวิธีการในการจัดสรรระบบบนที่เหตุการณ์ได้ โดยได้อธิบายรายละเอียดในบทที่ 3 หัวข้อการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบนที่เหตุการณ์บน IaaS คลาวด์ จึงได้โมเดลทางคณิตศาสตร์ ได้ตั้งสมการ (6)

เมื่อผู้วิจัยได้ทำการออกแบบฟังก์ชันจุดประสงค์แล้ว ผู้วิจัยได้นำเอาหลักการในการทำงานไปเขียนโปรแกรมด้วยภาษา Python โดยใช้วิธีการในการหาค่าที่เหมาะสม ด้วยเทคนิค Random Search Algorithm และทำการทดสอบการทำงานของโปรแกรม โดยแบ่งออกเป็น

1. การทดสอบข้อมูลนำเข้าของจำนวนลือกเกอร์ และ VM เท่ากัน
2. การทดสอบข้อมูลนำเข้าของจำนวนลือกเกอร์น้อยกว่าจำนวน VM
3. การทดสอบข้อมูลนำเข้าของจำนวนลือกเกอร์มากกว่าจำนวน VM

จำนวนล๊อคเกอร์ 5 คอลัมน์ (L1 ถึง L5) หมายถึง ล๊อคเกอร์ที่พร้อมใช้งานตัวที่ 1 – 5 โดยสามารถกำหนดให้อยู่ในรูปของ L_x โดยที่ x คือ 1, 2, ..., 5 ในบรรทัดที่ 7 – 11 แสดงจำนวน VM 5 แถว (U1 ถึง U5) หมายถึง ผู้ให้บริการจำนวน 5 คนที่ได้ทำการเช่า VM กำหนดให้อยู่ในรูปของ U_y โดยที่ y คือ 1, 2, ..., 5 ส่วนค่าที่แสดงในเมทริกซ์ของวิธีการจัดสรรระบบบันทึกเหตุการณ์นั้นจะมีค่าเป็น 0 และ 1 ซึ่ง 0 หมายถึง ล๊อคเกอร์ตัวที่ L_x ไม่มีการตรวจจับโปรเซสการทำงานใน U_y และ 1 หมายถึง ล๊อคเกอร์ตัวที่ L_x มีการตรวจจับโปรเซสการทำงานใน U_y

บรรทัดที่ 12 แสดงจำนวนล๊อคเกอร์ 5 คอลัมน์ (L1 ถึง L5) บรรทัดที่ 13 เป็นการสรุปว่า ล๊อคเกอร์ที่มีการใช้งานนั้นได้แก่ L2, L3 และ L5 โดย L2 ทำการตรวจจับโปรเซสการทำงานใน VM 3 VMs L3 และ L5 ทำการตรวจจับโปรเซสการทำงานใน VM อย่างละ 1 VM ในขณะที่ L1 และ L4 จะเป็นล๊อคเกอร์ที่ไม่ได้ใช้งาน จะสรุปได้ในบรรทัดที่ 14 ว่าวิธีการนี้จะมีล๊อคเกอร์ที่ไม่ได้มีภาระงาน จำนวน 2 ล๊อคเกอร์ สำหรับในบรรทัดที่ 15 แสดงค่าที่ถูกลงโทษด้วยฟังก์ชันการปรับ ดังสมการ (4) เนื่องจากไม่เข้าเงื่อนไขของข้อจำกัด เพราะ VM2 ไม่เป็นไปตามข้อจำกัดจึงถูกทำงานในฟังก์ชันการปรับ ซึ่งจะได้อธิบายการคำนวณในหัวข้อการตรวจสอบการผิดข้อจำกัด และในบรรทัดที่ 16 จะเป็นการหาผลรวมของค่า optimal ดังสมการ (6) ซึ่งมีค่าเป็น 1.0 นั้นหมายความว่าผู้ให้บริการต้องพึงระวังในการจัดสรรล๊อคเกอร์ เนื่องจากไม่เป็นไปตามความต้องการของผู้ใช้งานทั้งหมด เมื่อมีการปรับเปลี่ยนผู้ให้บริการในรอบถัดไปผู้ให้บริการควรที่จะปรับเปลี่ยนให้เป็นไปตามความต้องการของผู้ใช้งานให้ได้มากที่สุด เพื่อให้เกิดความพึงพอใจของทั้งผู้ให้บริการและผู้ใช้งาน

ตรวจสอบการผิดข้อจำกัด

ผู้วิจัยได้ทำการตรวจสอบการผิดข้อจำกัดในล๊อคเกอร์ที่ใช้งาน โดยมีรายละเอียดดังนี้

1. Logger2 (L2) มีสมาชิกของ VM 3 VMs ได้แก่ VM3, VM4 และ VM5 เมื่อนำจำนวน VM ของ Logger2 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM3

$3 \leq 3 \leq 4$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

1.2 VM4

$3 \leq 3 \leq 4$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

1.3 VM5

$3 \leq 3 \leq 3$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

จาก Logger2 จะมี VM ที่เป็นไปตามข้อจำกัดจำนวน 3 VMs โดยมีการลงโทษจากฟังก์ชันการปรับ

2. Logger3 (L3) มีสมาชิกของ VM 1 ตัว ได้แก่ VM1 เมื่อนำไปจำนวน VM ของ Logger3 ไปตรวจสอบในฟังก์ชันการปรับดังสมการ (4) จะได้

2.1 VM1

$1 \leq 1 \leq 1$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

จาก Logger3 จะมี VM ที่เป็นไปตามข้อจำกัดจำนวน 1 VM โดยมีการการลงโทษจากฟังก์ชันการปรับ

3. Logger5 (L5) มีสมาชิกของ VM 1 VM ได้แก่ VM2 เมื่อนำไปจำนวน VM ของ Logger2 ไปตรวจสอบในฟังก์ชันการปรับดังสมการ (4) จะได้

3.1 VM2

$2 \not\leq 1 \leq 2$ มีค่าการปรับ = -1 เนื่องจากผิดข้อจำกัดในกรณีที่ $P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_p * (N(L_j) - P_i)$$

จะได้

$$cost = 0 + 1 * (2 - 1) = -1$$

จาก Logger5 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 1 VM โดยมีการการลงโทษจากฟังก์ชันการปรับ -1

ดังนั้นจะสรุปได้ว่ากรณีการทดสอบในภาพ 35 จะมีค่าลงโทษจากฟังก์ชันการปรับทั้งหมดเท่ากับ -1 โดย VM2 ไม่เป็นไปตามข้อจำกัดซึ่งมีค่าลงโทษ -1

1.2 กรณีที่แย่ที่สุด (Worst Case)

1. The number of Loggers: 5
2. The number of VMs: 5
3. The security Levels of VMs: [1, 2, 4, 4, 3]
4. The performance Levels of VMs: [1, 2, 3, 3, 3]
5. Worst solution==>
6. L1 L2 L3 L4 L5
7. U1 [1, 0, 0, 0, 0]
8. U2 [1, 0, 0, 0, 0]
9. U3 [1, 0, 0, 0, 0]
10. U4 [1, 0, 0, 0, 0]
11. U5 [1, 0, 0, 0, 0]
12. N(L1) N(L2) N(L3) N(L4) N(L5)
13. The number of VMs in Loggers: ['5' '0' '0' '0' '0']
14. The number of workload-free Logger: 4.0
15. The penalty value: -22.0
16. The optimal value: -18.0

ภาพ 36 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์และ VM เท่ากัน สำหรับกรณีที่แย่ที่สุด

ตรวจสอบการผิดข้อจำกัด

ผู้วิจัยได้ทำการตรวจสอบการผิดข้อจำกัดในล็อกเกอร์ที่ใช้งาน โดยมีรายละเอียดดังนี้

1. Logger1 (L1) มีสมาชิกของ VM 5 VMs ได้แก่ VM1 ถึง VM5 เมื่อนำจำนวน VM ของ Logger1 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM1

$1 \leq 5 \leq 1$ มีค่าการปรับ = -8 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = g_s * (S_i - N(L_j))$$

จะได้

$$cost = 0 + 2 * (1 - 5) = -8$$

1.2 VM2

$2 \leq 5 \leq 2$ มีค่าการปรับ = -6 เนื่องจากผิดข้อจำกัดในกรณีที่ $N_j \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -8 + 2 * (2 - 5) = -14$$

1.3 VM3

$3 \leq 5 \leq 4$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $N_j \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -14 + 2 * (4 - 5) = -16$$

1.4 VM4

$3 \leq 5 \leq 4$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -16 + 2 * (4 - 5) = -18$$

1.5 VM5

$3 \leq 5 \leq 3$ มีค่าการปรับ = -4 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -16 + 2 * (3 - 5) = -24$$

จาก Logger1 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 5 VM โดยมีการการลงโทษจากฟังก์ชันการปรับ -24

ดังนั้นจะสรุปได้ว่ากรณีการทดสอบในภาพ 36 จะมีค่าลงโทษจากฟังก์ชันการปรับทั้งหมดเท่ากับ -24 โดย VM ที่ผิดข้อจำกัดมากที่สุดได้แก่ VM1

ตรวจสอบการผิดข้อกำหนด

1. Logger2 (L2) มีสมาชิกของ VM 5 VMs ได้แก่ VM1, VM5, VM6, VM7 และ VM8 เมื่อนำจำนวน VM ของ Logger2 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM1

$$5 \leq 5 \leq 5 \text{ มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อกำหนด}$$

1.2 VM5

$$5 \leq 5 \leq 5 \text{ มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อกำหนด}$$

1.3 VM6

$$5 \leq 5 \leq 5 \text{ มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อกำหนด}$$

1.4 VM7

$$5 \leq 5 \leq 5 \text{ มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อกำหนด}$$

1.5 VM8

$$5 \leq 5 \leq 5 \text{ มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อกำหนด}$$

จาก Logger2 พบว่า VM1, VM5, VM6, VM7 และ VM8 เป็นไปตามข้อกำหนดทั้งหมดโดยที่ความต้องการระดับความปลอดภัยที่ระดับ 5 และระดับประสิทธิภาพที่ระดับ 5 เช่นกัน จึงทำให้ไม่มีค่าฟังก์ชันการปรับ

2. Logger3 (L3) มีสมาชิกของ VM 3 VMs ได้แก่ VM2, VM3 และ VM4 เมื่อนำจำนวน VM ของ Logger3 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

2.1 VM2

$2 \leq 3 \not\leq 2$ มีค่าการปรับ = -2 เนื่องจากผิดข้อกำหนดในกรณีที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$cost += g_s * (S_i - N(L_j))$$

จะได้

$$cost = 0 + 2 * (2 - 3) = -2$$

2.2 VM3

$$3 \leq 3 \leq 4 \text{ มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อกำหนด}$$

2.3 VM4

$$3 \leq 3 \leq 4 \text{ มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อกำหนด}$$

$5 \leq 8 \not\leq 5$ มีค่าการปรับ = -6 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_s * (S_i - N(L_j))$$

จะได้

$$cost = 0 + 2 * (5 - 8) = -6$$

1.2 VM2

$2 \leq 8 \not\leq 2$ มีค่าการปรับ = -12 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -6 + 2 * (2 - 8) = -18$$

1.3 VM3

$3 \leq 8 \not\leq 4$ มีค่าการปรับ = -8 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -18 + 2 * (4 - 8) = -26$$

1.4 VM4

$3 \leq 8 \not\leq 4$ มีค่าการปรับ = -8 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -26 + 2 * (4 - 8) = -34$$

1.5 VM5

$5 \leq 8 \not\leq 5$ มีค่าการปรับ = -6 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -34 + 2 * (5 - 8) = -40$$

1.6 VM6

$5 \leq 8 \not\leq 5$ มีค่าการปรับ = -6 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -40 + 2 * (5 - 8) = -46$$

1.7 VM7

$5 \leq 8 \leq 5$ มีค่าการปรับ = -6 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$cost = -46 + 2 * (5 - 8) = -52$$

1.8 VM8

$5 \leq 8 \leq 5$ มีค่าการปรับ = -6 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$cost = -52 + 2 * (5 - 8) = -58$$

จาก Logger1 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 8 VM โดยมีการลงโทษจากฟังก์ชันการปรับ -58

ดังนั้นจะสรุปได้ว่ากรณีการทดสอบในภาพ 38 จะมีค่าลงโทษจากฟังก์ชันการปรับทั้งหมดเท่ากับ -58 โดย VM ที่ผิดข้อจำกัดมากที่สุดได้แก่ VM2

3. การทดสอบข้อมูลนำเข้าของจำนวนล็อกเกอร์มากกว่าจำนวน VM (ข้อมูลชุดที่

3)

3.1 กรณีที่ดีที่สุด (Best Case)

1. The number of Loggers: 10
2. The number of VMs: 8
3. The security Levels of VMs: [5, 5, 5, 5, 5, 5, 2, 2]
4. The performance Levels of VMs: [3, 3, 4, 4, 3, 3, 2, 2]
5. Best Model ==>
6. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
7. U1 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
8. U2 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
9. U3 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
10. U4 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
11. U5 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U6 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U7 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
14. U8 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
15. N(L1) N(L2) N(L3) N(L4) N(L5) N(L6) N(L7) N(L8) N(L9) N(L10)
16. The numbers of VMs in Loggers: ['0' '0' '0' '3' '0' '0' '0' '3' '0' '2']
17. The number of workload-free Logger: 7.0
18. The penalty value: -2.0
19. The optimal value: 5.0

ภาพ 39 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์มากกว่าจำนวนของ VM สำหรับกรณีที่ดีที่สุด

จากภาพ 39 ภาพผลลัพธ์นี้แบ่งออกเป็น 3 ส่วน ได้แก่ ส่วนข้อมูลนำเข้า(บรรทัดที่ 1 – 4) ส่วนวิธีการจัดสรรระบบบันทึกเหตุการณ์ที่ดีที่สุด (บรรทัดที่ 6 – 14) และส่วนสรุป (บรรทัดที่ 15 – 19) (รายละเอียดได้อธิบายไว้ในข้อที่ 1) การทดสอบข้อมูลนำเข้าของจำนวนล็อกเกอร์ และ VM เท่ากัน) ในผลลัพธ์จากภาพ 39 แสดงให้เห็นว่าในกรณีที่จำนวนล็อกเกอร์มากกว่าจำนวน VM การจัดสรรระบบบันทึกเหตุการณ์สามารถสร้างวิธีการในการจัดสรรได้ ซึ่งพบว่า ล็อกเกอร์ที่มีการใช้งานนั้น ได้แก่ L4, L8 และ L10 โดย L4 ทำการตรวจจับโปรเซสการทำงานใน VM 3 VMs ได้แก่ VM4, VM5 และ VM6 ล็อกเกอร์ L8 ทำการตรวจจับโปรเซสการทำงานใน VM 3 VMs ได้แก่ VM1, VM2 และ VM3 ล็อกเกอร์ L10 ทำการตรวจจับโปรเซสการทำงานใน VM 2 VMs ได้แก่ VM7 และ VM8 ในขณะที่ L1, L2, L3, L5, L6, L7 และ L9 จะเป็นล็อกเกอร์ที่ไม่ได้ใช้งาน ดังแสดงในบรรทัดที่ 16 และสามารถแสดงจำนวนล็อกเกอร์ที่ไม่ได้ใช้งานก่อนการคำนวณหาฟังก์ชันการปรับได้ โดยมีค่าเท่ากับ 7 ดังแสดงในบรรทัดที่ 17 สำหรับในกรณีตัวอย่างนี้จะมีค่าลงโทษจากฟังก์ชันการปรับเท่ากับ -2.0 ซึ่งจะแสดงให้เห็นว่า VM ไหนที่มีค่าลงโทษมากที่สุด ดังนี้

ตรวจสอบการผิดข้อจำกัด

1. Logger4 (L4) มีสมาชิกของ VM 3 VMs ได้แก่ VM4, VM5 และ VM6 เมื่อนำจำนวน VM ของ Logger1 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM4

$4 \not\leq 3 \leq 5$ มีค่าการปรับ = -1 เนื่องจากผิดข้อจำกัดในกรณีที่ $P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = g_p * (N(L_j) - P_i)$$

จะได้

$$cost = 0 + 1 * (2 - 3) = -1$$

1.2 VM5

$3 \leq 3 \leq 5$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

1.4 VM6

$3 \leq 3 \leq 5$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

จาก Logger4 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 1 VM โดยมีการการลงโทษจากฟังก์ชันการปรับ -1

2. Logger8 (L8) มีสมาชิกของ VM 3 VMs ได้แก่ VM1, VM2 และ VM3 เมื่อนำจำนวน VM ของ Logger8 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

2.1 VM1

$3 \leq 3 \leq 5$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

2.2 VM2

$3 \leq 3 \leq 5$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

2.3 VM3

$4 \not\leq 3 \leq 5$ มีค่าการปรับ = -1 เนื่องจากผิดข้อจำกัดในกรณีที่ $P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = 0 + 1 * (2 - 3) = -1$$

จาก Logger8 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 1 VM โดยมีการการลงโทษจากฟังก์ชันการปรับ -1

3. Logger10 (L10) มีสมาชิกของ VM 2 VMs ได้แก่ VM7 และ VM8 เมื่อนำจำนวน VM ของ Logger10 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

3.1 VM7

$2 \leq 2 \leq 2$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

3.2 VM8

$2 \leq 2 \leq 2$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

จาก Logger10 จะมี VM ที่เป็นไปตามข้อจำกัดจำนวน 2 VMs โดยมีการการลงโทษจากฟังก์ชันการปรับ

ดังนั้นจะสรุปได้ว่ากรณีการทดสอบในภาพ 39 จะมีค่าลงโทษจากฟังก์ชันการปรับทั้งหมดเท่ากับ -2 โดย VM3 และ VM4 มีค่าลงโทษมากที่สุดเท่ากับ -1 ส่วน VM1, VM2 และ VM5 ถึง VM 8 เป็นไปตามข้อจำกัดจึงไม่มีค่าลงโทษ

3.2 กรณีที่แย่ที่สุด (Worst Case)

1. The number of Loggers: 10
2. The number of VMs: 8
3. The security Levels of VMs: [5, 5, 5, 5, 5, 5, 2, 2]
4. The performance Levels of VMs: [3, 3, 4, 4, 3, 3, 2, 2]
5. Worst Model ==>
6. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
7. U1 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
8. U2 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
9. U3 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
10. U4 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
11. U5 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
12. U6 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
13. U7 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
14. U8 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
15. N(L1) N(L2) N(L3) N(L4) N(L5) N(L6) N(L7) N(L8) N(L9) N(L10)
16. The numbers of VMs in Loggers: ['0' '1' '0' '0' '0' '6' '0' '0' '0' '1']
17. The number of workload-free Logger: 7.0
18. The penalty value: -29.0
19. The optimal value: -22.0

ภาพ 40 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีจำนวนล็อกเกอร์มากกว่าจำนวนของ VM สำหรับกรณีที่แย่ที่สุด

ตรวจสอบการผิดข้อจำกัด

ผู้วิจัยได้ทำการตรวจสอบการผิดข้อจำกัดในล็อกเกอร์ที่ใช้งาน โดยมีรายละเอียดดังนี้

1. Logger2 (L2) มีสมาชิกของ VM 1 VM ได้แก่ VM5 เมื่อนำจำนวน VM ของ Logger2 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM5

$3 \leq 1 \leq 5$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $P_i \leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_p * (N(L_j) - P_i)$$

จะได้

$$cost = 0 + 1 * (1 - 3) = -2$$

จาก Logger2 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 1 VM โดยมีการการลงโทษจากฟังก์ชันการปรับ -2

2. Logger6 (L6) มีสมาชิกของ VM 6 VMs ได้แก่ VM1, VM2, VM4, VM6, VM7 และ VM8 เมื่อนำจำนวน VM ของ Logger6 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

2.1 VM1

$3 \leq 6 \leq 5$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -2 + 2 * (5 - 6) = -4$$

2.2 VM2

$3 \leq 6 \leq 5$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -4 + 2 * (5 - 6) = -6$$

2.3 VM4

$4 \leq 6 \leq 5$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -6 + 2 * (5 - 6) = -8$$

2.4 VM6

$3 \leq 6 \leq 5$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $N_j \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost = -8 + 2 * (5 - 6) = -10$$

2.5 VM7

$2 \leq 6 \not\leq 2$ มีค่าการปรับ = -8 เนื่องจากผิดข้อจำกัดในกรณี
ที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$cost = -10 + 2 * (2 - 6) = -18$$

2.6 VM8

$2 \leq 6 \not\leq 2$ มีค่าการปรับ = -8 เนื่องจากผิดข้อจำกัดในกรณี
ที่ $N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับ
ได้ ดังนี้

$$cost = -18 + 2 * (2 - 6) = -26$$

จาก Logger6 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 6 VMs โดยมี
การการลงโทษจากฟังก์ชันการปรับ -24

3. Logger10 (L10) มีสมาชิกของ VM 1 VM ได้แก่ VM3 เมื่อนำจำนวน VM ของ
Logger10 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

3.1 VM3

$4 \not\leq 1 \leq 5$ มีค่าการปรับ = -3 เนื่องจากผิดข้อจำกัดในกรณี
ที่ $P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$cost = -26 + 1 * (1 - 4) = -29$$

จาก Logger10 จะมี VM ที่ไม่เป็นไปตามข้อจำกัดจำนวน 1 VM โดยมีการ
การลงโทษจากฟังก์ชันการปรับ -3

ดังนั้นจะสรุปได้ว่ากรณีการทดสอบในภาพ 40 จะมีค่าลงโทษจากฟังก์ชัน
การปรับทั้งหมดเท่ากับ -29 โดย VM ที่ผิดข้อจำกัดมากที่สุดได้แก่ VM7 และ VM8

จากการทดลองทั้งหมดที่แสดงให้เห็นนี้ การสร้างโมเดลทางคณิตศาสตร์การ
จัดสรรระบบบนทึกเหตุการณ์บน IaaS คลาวด์ที่ผู้วิจัยได้คำนึงถึงในเรื่องของความแม่นยำในการ
ตรวจจับซึ่งนำไปสู่ความปลอดภัยและในเรื่องของประสิทธิภาพการทำงานของ VM ทำให้เห็นว่าโมเดล
ทางคณิตศาสตร์การจัดสรรระบบบนทึกเหตุการณ์บน IaaS คลาวด์ที่ผู้วิจัยได้สร้างขึ้นนั้นสามารถ
รองรับการทำงานของคุณลักษณะของเครื่องคอมพิวเตอร์ที่นำมาใช้ในการเป็นเครื่องโฮสต์ได้หลาย
รูปแบบ ไม่ว่าจะมีความสามารถในการรองรับได้มากน้อยเท่าไร โมเดลทางคณิตศาสตร์การจัดสรร
ระบบบนทึกเหตุการณ์บน IaaS คลาวด์ที่ผู้วิจัยได้สร้างขึ้นนี้สามารถรองรับการทำงานได้ครบทุก
รูปแบบ

ผู้วิจัยจึงสามารถสรุปตารางเปรียบเทียบการทำงานของโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ในกรณีที่ดีที่สุดในกรณีที่แย่ที่สุด ได้ดังตาราง

ตาราง 8 แสดงเปรียบเทียบการทำงานในกรณี best case และ worst case แบบ unconstrained optimization

ชุดข้อมูล	Best case			Worst case		
	Workload-free Logger	Penalty value	Optimal value Total	Workload-free Logger	Penalty value	Optimal value Total
ชุดที่ 1	2.0	-1.0	1.0	4.0	-22.0	-18.0
ชุดที่ 2	3.0	-2.0	1.0	4.0	-58.0	-54.0
ชุดที่ 3	7.0	-2.0	5.0	7.0	-29.0	-22.0

ผลการสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

จากการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ผู้วิจัยได้นำมาปรับปรุงให้เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในกรณีที่ผู้ใช้บริการเดิมยกเลิกการใช้งาน และมีผู้ใช้บริการรายใหม่เข้ามาใช้บริการ ซึ่งในการกรณีที่ผู้ใช้บริการยกเลิกการใช้งานนั้น จะได้ทำการลบข้อมูลออกจากวิธีการจัดสรรระบบบันทึกเหตุการณ์ โดยได้อธิบายรายละเอียดไว้ในบทที่ 3 หัวข้อการสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ในหัวข้อนี้ผู้วิจัยจึงขอสรุปผลที่ได้จากการสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ดังนี้

1. การตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ สามารถตรวจสอบได้จาก สมการ (7)

2. การคำนวณค่าฟังก์ชันการปรับสำหรับในกรณีที่มีการเปลี่ยนล็อกเกอร์ สำหรับในกรณีการลงโทษในส่วนของ การเปลี่ยนล็อกเกอร์นั้น ผู้วิจัยได้กำหนดด้วยตัวแปร g_c โดยมีค่าน้ำหนักเท่ากับ 0.5 เนื่องจากเป็นกรณีที่ไม่มีร้ายแรงสำหรับการจัดสรรระบบบันทึกเหตุการณ์ จึงได้สมการในการคิดค่าสำหรับการเปลี่ยนล็อกเกอร์ ดังสมการ (8)

3. ฟังก์ชันจุดประสงค์

จากสมการ (8) ในการหาฟังก์ชันจุดประสงค์ของการทดลองการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บนบน IaaS คลาวด์ จะได้ฟังก์ชันจุดประสงค์ของการปรับการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ดังสมการ (9)

เมื่อผู้วิจัยได้ทำการออกแบบฟังก์ชันจุดประสงค์แล้ว ผู้วิจัยได้นำเอาหลักการในการทำงานไปเขียนโปรแกรมด้วยภาษา Python โดยใช้วิธีการในการหาค่าที่เหมาะสม ด้วยเทคนิค Random search algorithm และทำการทดสอบการทำงานของโปรแกรม โดยแบ่งออกเป็น

1. การทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก
2. การทดสอบในกรณีที่มีผู้ใช้บริการใหม่
3. การทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่
4. การทดสอบในกรณีที่มีผู้ใช้บริการมีการปรับระดับความปลอดภัย และระดับประสิทธิภาพ

โดยผู้วิจัยใช้ผลลัพธ์การจัดสรรระบบบันทึกเหตุการณ์จากภาพ 41 ในการจำลองการจัดสรรระบบบันทึกเหตุการณ์ในรูปแบบเดิมก่อนมีการเปลี่ยนแปลงผู้ใช้บริการ

1. The number of Loggers: 10
2. The number of VMs: 6
3. The security Levels of VMs: [1, 1, 5, 5, 5, 5]
4. The performance Levels of VMs: [1, 1, 4, 4, 3, 3]
5. Best Solution==>
6. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
7. U1 [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
8. U2 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
9. U3 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
10. U4 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
11. U5 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U6 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. N(L1) N(L2) N(L3) N(L4) N(L5) N(L6) N(L7) N(L8) N(L9) N(L10)
14. The numbers of VMs in Loggers: ['1' '0' '0' '4' '0' '0' '0' '0' '1' '0']
15. The number of workload-free Logger: 7.0
16. The penalty value: 0.0
17. The optimal value: 7.0

ภาพ 41 แสดงผลลัพธ์จากการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในรูปแบบเดิม

1. การทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก (ข้อมูลชุดที่ 4)

1. 10
2. 1 0 5 5 5 5
3. 1 0 4 4 3 3

ภาพ 42 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก

จากภาพ 42 คือข้อมูลนำเข้าที่อยู่ในไฟล์ input_change1.txt เป็นการยกตัวอย่างในกรณีที่ผู้ใช้บริการคนที่ 2 (VM2) ยกเลิกการใช้บริการ บรรทัดที่ 1 คือจำนวนล็อกเกอร์ที่พร้อมใช้งาน บรรทัดที่ 2 คือ ระดับความปลอดภัย และบรรทัดที่ 3 คือ ระดับประสิทธิภาพ โดยค่าที่เป็น 0 นั้น หมายถึง กรณีที่ผู้ใช้บริการคนที่ 2 (VM2) ยกเลิกการใช้บริการ ผู้วิจัยจึงทำการกำหนดค่าให้เป็น 0 เพื่อจะได้นำไปลบออกจากรูปแบบการจัดสรรระบบบันทึกเหตุการณ์ ซึ่งเมื่อนำเข้าข้อมูลตามภาพ 42 โปรแกรมจะทำการคำนวณรูปแบบการจัดสรรระบบบันทึกเหตุการณ์ได้ดังภาพ 43

1.1 กรณีที่ดีที่สุด (Best Case)

1. The security Levels of VMs new: [1, 5, 5, 5, 5]
2. The performance Levels of VMs new: [1, 4, 4, 3, 3]
3. The number of workload-free Logger: 8.0
4. The penalty value: 0.0
5. The penalty swap: 0.0
6. The optimal value: 8.0
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
11. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
14. New Solution ==>
15. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
16. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
17. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
18. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
19. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
20. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

ภาพ 43 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีผู้ใช้บริการยกเลิก สำหรับกรณีที่ดีที่สุด

จากภาพ 43 เป็นการแสดงผลลัพธ์ของวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีการยกเลิกผู้ใช้งาน ในภาพผลลัพธ์นี้แบ่งออกเป็น 4 ส่วน ได้แก่ 1) ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) 2) ส่วนสรุป (บรรทัดที่ 3 – 6) 3) ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 13) และ 4) ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 15 – 20) ผู้วิจัยจะได้อธิบายในแต่ละส่วนดังนี้

1. ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) แสดงระดับความปลอดภัยและระดับประสิทธิภาพที่ผู้ใช้บริการต้องการโดยได้ทำการตัดข้อมูลของผู้ใช้บริการที่ยกเลิกออกไป

2. ส่วนสรุป (บรรทัดที่ 3 – 6) เป็นข้อมูลสรุปการจัดสรรระบบบันทึกเหตุการณ์ ในบรรทัดที่ 3 แสดงจำนวนล็อกเกอร์ที่ว่างจำนวน 8 ล็อกเกอร์ บรรทัดที่ 4 แสดงค่าที่ถูกลงทะเบียนในการผิดข้อกำหนดของความปลอดภัยและประสิทธิภาพ ตามสมการ (4) บรรทัดที่ 5 แสดงค่าที่ถูกลงทะเบียนในการกรณีที่มีการเปลี่ยนล็อกเกอร์ของ VM ตามสมการ (8) ซึ่งในกรณีนี้มีค่าเป็น 0 เนื่องจากไม่มีการผิดข้อกำหนดทั้งในเรื่องของระดับความปลอดภัยและประสิทธิภาพ และการเปลี่ยนล็อกเกอร์ บรรทัดที่ 6 แสดงผลลัพธ์จากสมการ (9) ในการคำนวณหาค่าจำนวนล็อกเกอร์ที่ว่างมากที่สุดหลังจากหักค่าในส่วนของฟังก์ชันการปรับแล้ว นั่นก็คือค่า optimal

3. ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 13) แสดงวิธีการจัดสรรระบบบันทึกเหตุการณ์โดยการลบข้อมูลของ VM2 ออกเนื่องจากยกเลิกการใช้บริการจึงเหลือแสดงรายละเอียด 5 VMs โดย L4 จะตรวจจับโปรเซสการทำงานใน VM3, VM4, VM5 และ VM6 ส่วน L9 จะตรวจจับโปรเซสการทำงานใน VM1 และล็อกเกอร์ที่ไม่ได้ใช้งานได้แก่ L1, L2, L3, L5, L6, L7, L8 และ L10

4. ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 15 – 20) วิธีการจัดสรรระบบบันทึกเหตุการณ์ใหม่ที่ได้อยู่ในรูปแบบเดิม เนื่องจากไม่มีการเปลี่ยนล็อกเกอร์ โดยจะแสดงให้เห็นในการตรวจสอบการผิดข้อกำหนด

ตรวจสอบการผิดข้อกำหนด

กรณีผิดข้อกำหนดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger4 (L4) มีสมาชิกของ VM 4 VMs ได้แก่ VM3, VM4, VM5 และ VM6 เมื่อนำจำนวน VM ของ Logger4 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM3

$$4 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อกำหนด}$$

1.2 VM4

$$4 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อกำหนด}$$

1.3 VM5

$$3 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อกำหนด}$$

1.4 VM6

$$3 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

2. Logger9 (L9) มีสมาชิกของ VM 1 VM ได้แก่ VM1 เมื่อนำจำนวน VM ของ Logger9 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (12) จะได้

2.1 VM1

$$1 \leq 1 \leq 1 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผลคูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบใหม่ การคูณแบบสมาชิก มีค่าเป็น 1 ทั้งหมด ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อกเกอร์ ซึ่งได้อธิบายรายละเอียดไว้ในบทที่ 3 การสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า $Scost$ ดังสมการ (8) ได้ดังนี้

$$Scost = (((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5))$$

$$Scost = (0 + 0 + 0 + 0 + 0) = 0$$

ดังนั้นจะสรุปได้ว่าการทดสอบในกรณีที่มีผู้ให้บริการยกเลิกในภาพ 43 ไม่พบการผิดข้อจำกัดในเรื่องของระดับความปลอดภัยและประสิทธิภาพ และการเปลี่ยนล็อกเกอร์

1.2 สำหรับกรณีที่แย่ที่สุด (Worst Case)

1. The security Levels of VMs new: [1, 5, 5, 5, 5]
2. The performance Levels of VMs new: [1, 4, 4, 3, 3]
3. The number of workload-free Logger: 5.0
4. The penalty value: -10.0
5. The penalty swap: -2.5
6. The optimal value: -7.5
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
11. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
14. New Solution ==>
15. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
16. U1[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
17. U3[0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
18. U4[0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
19. U5[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
20. U6[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

ภาพ 44 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีผู้ใช้บริการยกเลิก สำหรับกรณีที่แย่มากที่สุด

ตรวจสอบการผิดข้อจำกัด

กรณีผิดข้อจำกัดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger5 (L5) มีสมาชิกของ VM 1 VM ได้แก่ VM6 เมื่อนำจำนวน VM ของ Logger5 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM6

$3 \not\leq 1 \leq 5$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่

$P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$cost += g_p * (N(L_j) - P_i)$$

จะได้

$$cost = 0 + 1 * (1 - 3) = -2$$

2. Logger6 (L6) มีสมาชิกของ VM 1 VM ได้แก่ VM1 เมื่อนำจำนวน VM ของ Logger6 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (12) จะได้

2.1 VM1

$$1 \leq 1 \leq 1 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อจำกัด}$$

3. Logger7 (L7) มีสมาชิกของ VM 1 VM ได้แก่ VM4 เมื่อนำจำนวน VM ของ Logger7 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

3.1 VM4

$$4 \not\leq 1 \leq 5 \quad \text{มีค่าการปรับ} = -3 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$\text{cost} = -2 + 1 * (1 - 4) = -5$$

4. Logger8 (L8) มีสมาชิกของ VM 1 VM ได้แก่ VM3 เมื่อนำจำนวน VM ของ Logger8 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

4.1 VM3

$$4 \not\leq 1 \leq 5 \quad \text{มีค่าการปรับ} = -3 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$\text{cost} = -5 + 1 * (1 - 4) = -8$$

5. Logger9 (L9) มีสมาชิกของ VM 1 VM ได้แก่ VM5 เมื่อนำจำนวน VM ของ Logger9 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

5.1 VM5

$$3 \not\leq 1 \leq 5 \quad \text{มีค่าการปรับ} = -2 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$\text{cost} = -8 + 1 * (1 - 3) = -10$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผลคูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบใหม่ การคูณแบบสมาชิก มีค่าเป็น 1 ทั้งหมด ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อกเกอร์ ซึ่งได้อธิบายรายละเอียดไว้ในบทที่ 3 การสร้างการปรับปรุงแบบการทำงานที่

เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า $Scost$ ดังสมการ (8) ได้ดังนี้

$$Scost = (((1 - 0) \times 0.5) + ((1 - 0) \times 0.5) + ((1 - 0) \times 0.5) + ((1 - 0) \times 0.5) + ((1 - 0) \times 0.5))$$

$$Scost = (0.5 + 0.5 + 0.5 + 0.5 + 0.5) = 2.5$$

ดังนั้นจะสรุปได้ว่าการทดสอบในกรณีที่มีผู้ให้บริการยกเลิกในกรณีที่แย่ที่สุดพบการผิดข้อจำกัดในเรื่องของระดับประสิทธิภาพ -10 และการเปลี่ยนล็อกเกอร์ -2.5

2 การทดสอบในกรณีที่มีผู้ให้บริการใหม่ (ข้อมูลชุดที่ 5)

1.	10
2.	1 1 5 5 5 5 4 3
3.	1 1 4 4 3 3 2 2

ภาพ 45 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ให้บริการใหม่

จากภาพ 45 คือข้อมูลนำเข้าที่อยู่ในไฟล์ `input_change2.txt` เป็นการยกตัวอย่างในกรณีที่ผู้ให้บริการคนที่ 7 และคนที่ 8 (VM7 และ VM8) เข้ามาใช้บริการใหม่ บรรทัดที่ 1 คือจำนวนล็อกเกอร์ที่พร้อมใช้งาน บรรทัดที่ 2 คือ ระดับความปลอดภัย และบรรทัดที่ 3 คือ ระดับประสิทธิภาพ เมื่อนำเข้าข้อมูลตามภาพ 45 โปรแกรมจะทำการคำนวณรูปแบบการจัดสรรระบบบันทึกเหตุการณ์ได้ดังภาพ 46

2.1 กรณีที่ดีที่สุด (Best Case)

1. The security Levels of VMs new: [1, 1, 5, 5, 5, 5, 4, 3]
2. The performance Levels of VMs new: [1, 1, 4, 4, 3, 3, 2, 2]
3. The number of workload-free Logger: 6.0
4. The penalty value: 0.0
5. The penalty swap: 0.0
6. The optimal value: 6.0
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
11. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
14. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
15. U7[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
16. U8[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
17. New Solution ==>
18. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
19. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
20. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
21. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
22. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
23. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
24. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
25. U7[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
26. U8[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

ภาพ 46 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการใหม่ สำหรับกรณีที่ดีที่สุด

จากภาพ 46 เป็นการแสดงผลลัพธ์ของวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีการยกเลิกผู้ใช้งาน ในภาพผลลัพธ์นี้แบ่งออกเป็น 4 ส่วน ได้แก่ 1) ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) 2) ส่วนสรุป (บรรทัดที่ 3 – 6) 3) ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 16) และ 4) ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 18 – 26) ผู้วิจัยจะได้อธิบายในแต่ละส่วนดังนี้

1. ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) แสดงระดับความปลอดภัยและระดับประสิทธิภาพที่ผู้ใช้บริการต้องการโดยมีผู้ใช้บริการใหม่เพิ่มเข้ามาจำนวน 2 คน ต่อท้าย จึงทำให้จำนวน VMs มีทั้งหมด 8 VMs

2. ส่วนสรุป (บรรทัดที่ 3 – 6) เป็นข้อมูลสรุปการจัดสรรระบบบันทึกเหตุการณ์ ในบรรทัดที่ 3 แสดงจำนวนล็อกเกอร์ที่ว่างจำนวน 6 ล็อกเกอร์ บรรทัดที่ 4 แสดงค่าที่ถูกลงทะเบียนในการผิดข้อจำกัดของความปลอดภัยและประสิทธิภาพ ตามสมการ (12) บรรทัดที่ 5 แสดงค่าที่ถูกลงทะเบียน

ในกรณีที่มีการเปลี่ยนล็อกเกอร์ของ VM ตามสมการ (16) ซึ่งในกรณีนี้มีค่าเป็น 0.0 เนื่องจากไม่มีการผิดข้อกำหนดทั้งในเรื่องของระดับความปลอดภัยและประสิทธิภาพ และการเปลี่ยนล็อกเกอร์ บรรทัดที่ 6 แสดงผลลัพธ์จากสมการ (17) ในการคำนวณหาค่าจำนวนล็อกเกอร์ที่ว่างมากที่สุดหลังจากหักค่าในส่วนของฟังก์ชันการปรับแล้ว

3. ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 16) แสดงวิธีการจัดสรรระบบบันทึกเหตุการณ์โดยการเพิ่มผู้ใช้งานใหม่ได้แก่ VM7 และ VM8 โดยกำหนดค่าให้เป็น 1 ทั้งหมด เพราะในกรณีที่มีการเพิ่มผู้ใช้ใหม่เข้ามาจะไม่นำมาพิจารณาในการคำนวณฟังก์ชันการปรับสำหรับการเปลี่ยนล็อกเกอร์ โดยจะพบว่า VM1 ถึง VM6 ไม่มีการเปลี่ยนล็อกเกอร์

4. ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 18 – 26) วิธีการจัดสรรระบบบันทึกเหตุการณ์ใหม่ที่ได้อำนาจ VM1 ถึง VM6 จะอยู่ในรูปแบบเดิมโดย L1 จะตรวจจับโปรเซสการทำงานใน VM2 L4 จะตรวจจับโปรเซสการทำงานใน VM3, VM4, VM5 และ VM6 ส่วน L9 จะตรวจจับโปรเซสการทำงานใน VM1 ส่วน VM7 และ VM8 จะถูกจัดสรรให้อยู่กับ L10 เพราะฉะนั้นล็อกเกอร์ที่ไม่ได้ใช้งานมีทั้งหมด 6 ล็อกเกอร์ ได้แก่ L2, L3, L5, L6, L7 และ L8

ตรวจสอบการผิดข้อกำหนด

กรณีผิดข้อกำหนดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger1 (L1) มีสมาชิกของ VM 1 VM ได้แก่ VM2 เมื่อนำจำนวน VM ของ Logger1 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM2

$$1 \leq 1 \leq 1 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อกำหนด}$$

2. Logger4 (L4) มีสมาชิกของ VM 4 VMs ได้แก่ VM3, VM4, VM5 และ VM6 เมื่อนำจำนวน VM ของ Logger4 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

2.1 VM3

$$4 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อกำหนด}$$

2.2 VM4

$$4 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อกำหนด}$$

2.3 VM5

$$3 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อกำหนด}$$

2.4 VM6

$$3 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

3. Logger9 (L9) มีสมาชิกของ VM 1 VM ได้แก่ VM1 เมื่อนำจำนวน VM ของ Logger9 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

3.1 VM1

$$1 \leq 1 \leq 1 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

4. Logger10 (L10) มีสมาชิกของ VM 2 VMs ได้แก่ VM7 และ VM8 เมื่อนำจำนวน VM ของ Logger10 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

4.1 VM7

$$2 \leq 2 \leq 4 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

4.2 VM8

$$2 \leq 2 \leq 3 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผลคูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบใหม่ด้วยการคูณแบบสมาชิก มีค่าเป็น 1 ทั้งหมด ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อกเกอร์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า $Scost$ ดังสมการ (8) ได้ดังนี้

$$\begin{aligned} Scost &= ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ &\quad + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ &\quad + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ Scost &= (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 0 \end{aligned}$$

ดังนั้นจะสรุปได้ว่าการทดสอบในกรณีที่มีผู้ใช้บริการใหม่เพิ่มเข้ามาในภาพ 46 ไม่พบการผิดข้อจำกัดในเรื่องของระดับความปลอดภัยและประสิทธิภาพ และการเปลี่ยนล็อกเกอร์

2.2 สำหรับกรณีที่แย่ที่สุด (Worst Case)

1. The security Levels of VMs new: [1, 1, 5, 5, 5, 5, 4, 3]
2. The performance Levels of VMs new: [1, 1, 4, 4, 3, 3, 2, 2]
3. The number of workload-free Logger: 8.0
4. The penalty value: -49.0
5. The penalty swap: -1.0
6. The optimal value: -42.0
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
11. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
14. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
15. U7[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
16. U8[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
17. New Solution ==>
18. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
19. U1[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
20. U2[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
21. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
22. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
23. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
24. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
25. U7[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
26. U8[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

ภาพ 47 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการใหม่ สำหรับกรณีที่แย่ที่สุด

ตรวจสอบการผิดข้อจำกัด

กรณีผิดข้อจำกัดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger4 (L4) มีสมาชิกของ VM 7 VMs ได้แก่ VM1 ถึง VM6 และ VM8 เมื่อนำจำนวน VM ของ Logger4 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM1

$1 \leq 7 \leq 1$ มีค่าการปรับ = -12 เนื่องจากผิดข้อจำกัดในกรณีที่

$N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$cost = 0 + 2 * (1 - 7) = -12$$

1.2 VM2

$$1 \leq 7 \not\leq 1 \quad \text{มีค่าการปรับ} = -12 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -12 + 2 * (1 - 7) = -24$$

1.3 VM3

$$4 \leq 7 \not\leq 5 \quad \text{มีค่าการปรับ} = -4 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -24 + 2 * (5 - 7) = -28$$

1.4 VM4

$$4 \leq 7 \not\leq 5 \quad \text{มีค่าการปรับ} = -4 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -28 + 2 * (5 - 7) = -32$$

1.5 VM5

$$3 \leq 7 \not\leq 5 \quad \text{มีค่าการปรับ} = -4 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -32 + 2 * (5 - 7) = -36$$

1.6 VM6

$$3 \leq 7 \not\leq 5 \quad \text{มีค่าการปรับ} = -4 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึง
คำนวณฟังก์ชันการปรับได้ ดังนี้

$$\text{cost} = -36 + 2 * (5 - 7) = -40$$

1.7 VM8

$$1 \leq 7 \not\leq 3 \quad \text{มีค่าการปรับ} = -8 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึง
คำนวณฟังก์ชันการปรับได้ ดังนี้

$$\text{cost} = -40 + 2 * (3 - 7) = -48$$

2. Logger6 (L6) มีสมาชิกของ VM 1 VM ได้แก่ VM7 เมื่อนำจำนวน VM ของ Logger6 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

2.1 VM7

$2 \not\leq 1 \leq 3$ มีค่าการปรับ = -1 เนื่องจากผิดข้อจำกัดในกรณีที่

$P_i \not\leq N(L_j)$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$\text{cost} = -48 + 1 * (1 - 2) = -49$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผลคูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบใหม่ การคูณแบบสมาชิก มีค่าเป็น 1 ทั้งหมด ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อกเกอร์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า Scost ดังสมการ (8) ได้ดังนี้

$$\begin{aligned} \text{Scost} &= (((1 - 0) \times 0.5) + ((1 - 0) \times 0.5) + ((1 - 1) \times 0.5) \\ &\quad + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ &\quad + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5)) \\ \text{Scost} &= (0.5 + 0.5 + 0 + 0 + 0 + 0 + 0 + 0) = 1.0 \end{aligned}$$

ดังนั้นจะสรุปได้ว่าการทดสอบในกรณีที่ผู้ใช้บริการยกเลิกในกรณีที่แย่งที่สุดพบการผิดข้อจำกัดในเรื่องของระดับความปลอดภัยและประสิทธิภาพ -49 และการเปลี่ยนล็อกเกอร์ -1.0

3. การทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่ (ข้อมูลชุดที่ 6)

$$\begin{array}{l} 1. \quad 10 \\ 2. \quad 1 \ 1 \ \boxed{0} \ 5 \ 5 \ 5 \ \boxed{4 \ 3} \\ 3. \quad 1 \ 1 \ \boxed{0} \ 4 \ 3 \ 3 \ \boxed{2 \ 3} \end{array}$$

ภาพ 48 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่

จากภาพ 48 คือข้อมูลนำเข้าที่อยู่ในไฟล์ input_change3.txt เป็นการยกตัวอย่างในกรณีที่ผู้ใช้บริการคนที่ 3 (VM3) ยกเลิกการใช้บริการ บรรทัดที่ 1 คือจำนวนล็อกเกอร์ที่พร้อมใช้งาน

บรรทัดที่ 2 คือ ระดับความปลอดภัย และบรรทัดที่ 3 คือ ระดับประสิทธิภาพ โดยค่าที่เป็น 0 นั้นหมายถึง กรณีที่ผู้ใช้บริการคนที่ 3 (VM3) ยกเลิกการใช้บริการ ผู้วิจัยจึงทำการกำหนดค่าให้เป็น 0 เพื่อจะได้นำไปลบออกจากรูปแบบการจัดสรรระบบบันทึกเหตุการณ์ และมีผู้ใช้บริการใหม่เพิ่มเข้ามาจำนวน 2 คนต่อท้ายในบรรทัดที่ 2 และ 3 ซึ่งเมื่อนำเข้าข้อมูลตามภาพ 48 โปรแกรมจะทำการคำนวณรูปแบบการจัดสรรระบบบันทึกเหตุการณ์ได้ดังภาพ 49

3.1 กรณีที่ดีที่สุด (Best Case)

1. The security Levels of VMs new: [1, 1, 5, 5, 5, 4, 3]
2. The performance Levels of VMs new: [1, 1, 4, 3, 3, 2, 3]
3. The number of workload-free Logger: 6.0
4. The penalty value: -2.0
5. The penalty swap: 0.0
6. The optimal value: 4.0
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
11. U4[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U5[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U6[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
14. U7[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
15. U8[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
16. New Solution ==>
17. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
18. U1[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
19. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
20. U4[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
21. U5[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
22. U6[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
23. U7[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
24. U8[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

ภาพ 49 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่ สำหรับกรณีที่ที่ดีที่สุด

จากภาพ 49 เป็นการแสดงผลลัพธ์ของวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีการยกเลิกผู้ใช้งาน และมีผู้ใช้บริการใหม่ ในภาพผลลัพธ์นี้แบ่งออกเป็น 4 ส่วน ได้แก่ 1) ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) 2) ส่วนสรุป (บรรทัดที่ 3 – 6) 3) ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 15) และ 4) ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 17 – 24) ผู้วิจัยจะได้อธิบายในแต่ละส่วนดังนี้

1. ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) แสดงระดับความปลอดภัยและระดับประสิทธิภาพที่ผู้ใช้บริการต้องการโดยได้ทำการตัดข้อมูลของผู้ใช้บริการที่ยกเลิกออกไป และเพิ่มผู้ใช้บริการใหม่เข้ามาจึงทำให้จำนวน VMs มีทั้งหมด 7 VMs

2. ส่วนสรุป (บรรทัดที่ 3 – 6) เป็นข้อมูลสรุปการจัดสรรระบบบันทึกเหตุการณ์ ในบรรทัดที่ 3 แสดงจำนวนล็อกเกอร์ที่ว่างจำนวน 6 ล็อกเกอร์ บรรทัดที่ 4 แสดงค่าที่ถูกลงทะเบียนในการผิดข้อจำกัดของความปลอดภัยและประสิทธิภาพ ตามสมการ (4) มีค่าเท่ากับ -2.0 บรรทัดที่ 5 แสดงค่าที่ถูกลงทะเบียนในกรณีที่มีการเปลี่ยนล็อกเกอร์ของ VM ตามสมการ (6) ซึ่งในกรณีนี้มีค่าเป็น 0.0 เนื่องจากไม่มีการผิดข้อจำกัดในเรื่องการเปลี่ยนล็อกเกอร์ บรรทัดที่ 6 แสดงผลลัพธ์จากสมการ (17) ในการคำนวณหาค่าจำนวนล็อกเกอร์ที่ว่างมากที่สุดหลังจากหักค่าในส่วนของฟังก์ชันการปรับแล้ว มีค่าเป็น 4.0

3. ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 15) แสดงวิธีการจัดสรรระบบบันทึกเหตุการณ์โดยมีการลบผู้ใช้บริการเดิม (VM3) ออกและเพิ่มผู้ใช้งานใหม่ได้แก่ VM7 และ VM8 โดยกำหนดค่าให้เป็น 1 ทั้งหมด เพราะในกรณีที่มีการเพิ่มผู้ใช้ใหม่เข้ามาจะไม่นำมาพิจารณาในการคำนวณฟังก์ชันการปรับสำหรับการเปลี่ยนล็อกเกอร์ โดยจะพบว่า VM1, VM2, VM4, VM5 และ VM6 ไม่มีการเปลี่ยนล็อกเกอร์

4. ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 17 – 26) วิธีการจัดสรรระบบบันทึกเหตุการณ์ใหม่ที่ได้สำหรับ VM1, VM2, VM4, VM5 และ VM6 จะอยู่ในรูปแบบเดิมโดย L1 จะตรวจจับโปรเซสการทำงานใน VM2 L4 จะตรวจจับโปรเซสการทำงานใน VM4, VM5, VM6 และเพิ่ม VM7 เข้ามาอยู่ใน L4 ด้วย ส่วน L9 จะตรวจจับโปรเซสการทำงานใน VM1 ส่วน VM8 จะถูกจัดสรรให้อยู่กับ L10 เพราะฉะนั้นล็อกเกอร์ที่ไม่ได้ใช้งานมีทั้งหมด 6 ล็อกเกอร์ ได้แก่ L2, L3, L5, L6, L7 และ L8

ตรวจสอบการผิดข้อจำกัด

กรณีผิดข้อจำกัดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger1 (L1) มีสมาชิกของ VM 1 VM ได้แก่ VM2 เมื่อนำจำนวน VM ของ Logger1 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM2

$$1 \leq 1 \leq 1 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อจำกัด}$$

2. Logger4 (L4) มีสมาชิกของ VM 4 VMs ได้แก่ VM4, VM5, VM6 และ VM7 เมื่อนำจำนวน VM ของ Logger4 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

2.1 VM4

$$4 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อจำกัด}$$

2.2 VM5

$$4 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \quad \text{เนื่องจากเป็นไปตามข้อจำกัด}$$

2.3 VM6

$$3 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

2.4 VM7

$$2 \leq 4 \leq 4 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

3. Logger9 (L9) มีสมาชิกของ VM 1 VM ได้แก่ VM1 เมื่อนำจำนวน VM ของ Logger9 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

3.1 VM1

$$1 \leq 1 \leq 1 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

4. Logger10 (L10) มีสมาชิกของ VM 1 VM ได้แก่ VM8 เมื่อนำจำนวน VM ของ Logger10 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

4.1 VM8

$$3 \not\leq 1 \leq 3 \quad \text{มีค่าการปรับ} = -2 \text{ เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$P_i \not\leq N_j$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_p * (N_j - P_i)$$

จะได้

$$cost = 0 + 1 * (1 - 3) = -2$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผลคูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบใหม่ด้วยการคูณแบบสมาชิก มีค่าเป็น 1 ทั้งหมด ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อกเกอร์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า $Scost$ ดังสมการ (8) ได้ดังนี้

$$\begin{aligned} Scost = & ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ & + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ & + ((1 - 1) \times 0.5) \end{aligned}$$

$$Scost = (0 + 0 + 0 + 0 + 0 + 0 + 0) = 0$$

ดังนั้นจะสรุปได้ว่าการทดสอบในกรณีที่มีการยกเลิกผู้ใช้งาน และมีผู้ใช้บริการใหม่เพิ่มเข้ามาในภาพ 49 จะมีค่าลงโทษจากฟังก์ชันการปรับทั้งหมดเท่ากับ -2 เนื่องจาก VM8 ผิดข้อจำกัดในเรื่องของระดับประสิทธิภาพ และในส่วนของ การเปลี่ยนล็อกเกอร์ไม่มีค่าคงโทษเนื่องจากไม่มีการเปลี่ยนล็อกเกอร์ของ VMs

3.2 สำหรับกรณีที่แย่ที่สุด (Worst Case)

1. The security Levels of VMs new: [1, 1, 5, 5, 5, 4, 3]
2. The performance Levels of VMs new: [1, 1, 4, 3, 3, 2, 3]
3. The number of workload-free Logger: 9.0
4. The penalty value: -50.0
5. The penalty swap: -1.0
6. The optimal value: -42.0
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
11. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
12. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
13. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
14. U7[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
15. U8[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
16. New Solution ==>
17. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
18. U1[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
19. U2[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
20. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
21. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
22. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
23. U7[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
24. U8[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]

ภาพ 50 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ให้บริการยกเลิก และมีผู้ให้บริการใหม่ สำหรับกรณีที่แย่ที่สุด

ตรวจสอบการผิดข้อจำกัด

กรณีผิดข้อจำกัดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger4 (L4) มีสมาชิกของ VM 7 VMs ได้แก่ VM1, VM2 และ VM4 ถึง VM8 เมื่อนำจำนวน VM ของ Logger4 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM1

$1 \leq 7 \leq 1$ มีค่าการปรับ = -12 เนื่องจากผิดข้อจำกัดในกรณีที่

$N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$cost = 0 + 2 * (1 - 7) = -12$$

1.2 VM2

$$1 \leq 7 \not\leq 1 \quad \text{มีค่าการปรับ} = -12 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -12 + 2 * (1 - 7) = -24$$

1.3 VM4

$$4 \leq 7 \not\leq 5 \quad \text{มีค่าการปรับ} = -4 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -24 + 2 * (5 - 7) = -28$$

1.4 VM5

$$3 \leq 7 \not\leq 5 \quad \text{มีค่าการปรับ} = -4 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -28 + 2 * (5 - 7) = -32$$

1.5 VM6

$$3 \leq 7 \not\leq 5 \quad \text{มีค่าการปรับ} = -4 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -32 + 2 * (5 - 7) = -36$$

1.6 VM7

$$2 \leq 7 \not\leq 4 \quad \text{มีค่าการปรับ} = -6 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -36 + 2 * (4 - 7) = -42$$

1.7 VM8

$$3 \leq 7 \not\leq 3 \quad \text{มีค่าการปรับ} = -8 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -42 + 2 * (3 - 7) = -50$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผลคูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบใหม่ การคูณแบบสมาชิก มีค่าเป็น 1 ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อกเกอร์ มีค่าเป็น 0 ซึ่งหมายความว่า VMs มีการเปลี่ยนล็อกเกอร์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า $Scost$ ดังสมการ (8) ได้ดังนี้

$$\begin{aligned}
 Scost &= ((1 - 0) \times 0.5) + ((1 - 0) \times 0.5) + ((1 - 1) \times 0.5) \\
 &\quad + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\
 &\quad + ((1 - 1) \times 0.5) \\
 Scost &= (0.5 + 0.5 + 0 + 0 + 0 + 0 + 0) = 1.0
 \end{aligned}$$

ดังนั้นจะสรุปได้ว่าการทดสอบในกรณีที่มีผู้ใช้บริการยกเลิกในกรณีที่แย่งที่สุดพบการผิดข้อจำกัดในเรื่องของระดับความปลอดภัยและประสิทธิภาพ -50 และการเปลี่ยนล็อกเกอร์ -1.0

4. การทดสอบในกรณีที่มีผู้ใช้บริการมีการปรับระดับความปลอดภัย และระดับประสิทธิภาพ (ข้อมูลชุดที่ 7)

1.	10				
2.	1	3	5	5	5
3.	1	3	4	4	3

ภาพ 51 แสดงตัวอย่างข้อมูลนำเข้าสำหรับการทดสอบในกรณีที่มีผู้ใช้บริการปรับระดับความปลอดภัยและระดับประสิทธิภาพ

จากภาพ 51 คือข้อมูลนำเข้าที่อยู่ในไฟล์ `input_change4.txt` เป็นการยกตัวอย่างในกรณีที่ผู้ใช้บริการคนที่ 2 (VM2) มีการปรับระดับความปลอดภัย และระดับประสิทธิภาพ จากเดิมกำหนดระดับความปลอดภัย และระดับประสิทธิภาพที่ระดับ 1 แต่เมื่อเวลาเปลี่ยนไปผู้ใช้บริการมีการปรับระดับระดับความปลอดภัย ในบรรทัดที่ 2 และระดับประสิทธิภาพ ในบรรทัดที่ 3 เป็นระดับ 3 ซึ่งเมื่อนำเข้าข้อมูลตามภาพ 51 โปรแกรมจะทำการคำนวณรูปแบบการจัดสรรระบบบันทึกเหตุการณ์ ได้ดังภาพ 52

4.1 กรณีที่ดีที่สุด (Best Case)

1. The security Levels of VMs new: [1, 3, 5, 5, 5, 5]
2. The performance Levels of VMs new: [1, 3, 4, 4, 3, 3]
3. The number of workload-free Logger: 7.0
4. The penalty value: -2.0
5. The penalty swap: 0.0
6. The optimal value: 5.0
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
11. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
14. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
15. New Solution ==>
16. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
17. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
18. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
17. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
18. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
19. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
20. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

ภาพ 52 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการ
ปรับระดับความปลอดภัยและระดับประสิทธิภาพ กรณีที่ดีที่สุด

จากภาพ 52 เป็นการแสดงผลของวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ ในกรณีที่มีปรับระดับความปลอดภัยและระดับประสิทธิภาพ ในภาพผลลัพธ์นี้แบ่งออกเป็น 4 ส่วน ได้แก่ 1) ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) 2) ส่วนสรุป (บรรทัดที่ 3 – 6) 3) ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 14) และ 4) ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 15 – 20) ผู้วิจัยจะได้อธิบายในแต่ละส่วนดังนี้

1. ส่วนข้อมูลนำเข้า (บรรทัดที่ 1 – 2) แสดงระดับความปลอดภัยและระดับประสิทธิภาพที่ผู้ใช้บริการต้องการโดยมีผู้ใช้บริการมีการปรับระดับระดับความปลอดภัย และระดับประสิทธิภาพ เป็นระดับ 3 สำหรับ VM2

2. ส่วนสรุป (บรรทัดที่ 3 – 6) เป็นข้อมูลสรุปการจัดสรรระบบบันทึกเหตุการณ์ ในบรรทัดที่ 3 แสดงจำนวนล็อกเกอร์ที่วางจำนวน 7 ล็อกเกอร์ บรรทัดที่ 4 แสดงค่าที่ถูกลงทะเบียนในการผิดข้อจำกัดของความปลอดภัยและประสิทธิภาพ ตามสมการ (4) บรรทัดที่ 5 แสดงค่าที่ถูกลงทะเบียนในกรณีที่มีการเปลี่ยนล็อกเกอร์ของ VM ตามสมการ (7) ซึ่งในกรณีนี้มีค่าเป็น -2.0 เนื่องจากมีการผิด

ข้อจำกัดทั้งในเรื่องของระดับความปลอดภัยและประสิทธิภาพ บรรทัดที่ 6 แสดงผลลัพธ์จากสมการ (8) ในการคำนวณหาค่าจำนวนล็อกเกอร์ที่ว่างมากที่สุดหลังจากหักค่าในส่วนของฟังก์ชันการปรับแล้ว มีค่าเท่ากับ 5.0

3. ส่วนวิธีการจัดสรรรูปแบบเก่า (บรรทัดที่ 8 – 14) แสดงวิธีการจัดสรรระบบบันทึกเหตุการณ์เดิม

4. ส่วนวิธีการจัดสรรรูปแบบใหม่ (บรรทัดที่ 15 – 20) วิธีการจัดสรรระบบบันทึกเหตุการณ์ใหม่ที่ได้สำหรับ VM1 ถึง VM6 จะอยู่ในรูปแบบเดิมโดย L1 จะตรวจจับโปรเซสการทำงานใน VM2 L4 จะตรวจจับโปรเซสการทำงานใน VM3, VM4, VM5 และ VM6 ส่วน L9 จะตรวจจับโปรเซสการทำงานใน VM1 เพราะฉะนั้นรูปแบบการจัดสรรระบบบันทึกเหตุการณ์นั้นจะไม่มีเปลี่ยนแปลง แต่จะมีการถูกลงโทษด้วยฟังก์ชันการปรับเนื่องจากผิดข้อจำกัดที่กำหนด

ตรวจสอบการผิดข้อจำกัด

กรณีผิดข้อจำกัดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger1 (L1) มีสมาชิกของ VM 1 VM ได้แก่ VM2 เมื่อนำจำนวน VM ของ Logger1 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM2

$3 \not\leq 1 \leq 3$ มีค่าการปรับ = -2 เนื่องจากผิดข้อจำกัดในกรณีที่ $P_i \not\leq N_j$ จะพบว่า จำนวน VM น้อยกว่าระดับประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_p * (N_j - P_i)$$

จะได้

$$cost = 0 + 1 * (1 - 3) = -2$$

2. Logger4 (L4) มีสมาชิกของ VM 4 VMs ได้แก่ VM3, VM4, VM5 และ VM6 เมื่อนำจำนวน VM ของ Logger4 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

2.1 M4

$4 \leq 4 \leq 5$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

2.2 VM5

$4 \leq 4 \leq 5$ มีค่าการปรับ = 0 เนื่องจากเป็นไปตามข้อจำกัด

2.3 VM6

$$3 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

2.4 VM7

$$3 \leq 4 \leq 5 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

3. Logger9 (L9) มีสมาชิกของ VM 1 VM ได้แก่ VM1 เมื่อนำจำนวน VM ของ Logger9 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

3.1 VM1

$$1 \leq 1 \leq 1 \quad \text{มีค่าการปรับ} = 0 \text{ เนื่องจากเป็นไปตามข้อจำกัด}$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผลคูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบใหม่ด้วยการคูณแบบสมาชิก มีค่าเป็น 1 ทั้งหมด ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อกเกอร์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า $Scost$ ดังสมการ (8) ได้ดังนี้

$$\begin{aligned} Scost &= (((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) \\ &\quad + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5)) \\ Scost &= (0 + 0 + 0 + 0 + 0 + 0) = 0 \end{aligned}$$

ดังนั้นจะสรุปได้ว่าการทดสอบเหตุการณ์ ในกรณีที่มีปรับระดับความปลอดภัยและระดับประสิทธิภาพในภาพ 52 จะมีค่าลงโทษจากฟังก์ชันการปรับทั้งหมดเท่ากับ -2 เนื่องจาก VM2 ผิดข้อจำกัดในเรื่องของระดับประสิทธิภาพ เนื่องจากมีการปรับเปลี่ยนระดับความปลอดภัยและระดับประสิทธิภาพที่ทำให้ต้องมีการถูกลงโทษด้วยการปรับน้อยที่สุด ซึ่งถ้าจัดให้ VM2 ไปอยู่ที่ล็อกเกอร์เดียวกับ VM1 นั้น จะทำให้เกิดการผิดข้อจำกัด เป็น -3 เนื่องจากมีการผิดข้อจำกัดดังนี้

1. VM1

$$1 \leq 2 \not\leq 1 \quad \text{มีค่าการปรับ} = -2 \text{ เนื่องจากผิดข้อจำกัดในกรณีนี้}$$

$N_j \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_s * (S_i - N_j)$$

จะได้

$$cost = 0 + 2 * (1 - 2) = -2$$

2. VM2

$$3 \not\leq 2 \leq 3 \quad \text{มีค่าการปรับ} = -1 \text{ เนื่องจากผิดข้อจำกัดในกรณีนี้}$$

$P_i \not\leq N_j$ จะพบว่า จำนวน VM น้อยกว่าระดับความประสิทธิภาพ จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_p * (N_j - P_i)$$

จะได้ผลรวมของ

$$cost = -2 + 1 * (2 - 3) = -3$$

ฟังก์ชันการปรับสำหรับการจัดรวม VM1 กับ VM2 นี้ทำให้เกิดค่าฟังก์ชันการปรับ

เท่ากับ -3

และถ้า นำ VM2 ไปอยู่ในล๊อคเกอร์ 4 ซึ่งจะมี VM ที่อยู่ภายใต้การทำงานของล๊อคเกอร์ 4 จำนวน 5 VMs จะมีค่าฟังก์ชันการปรับ ดังนี้

3. VM2

$3 \leq 5 \not\leq 3$ มีค่าการปรับ = -4 เนื่องจากผิดข้อจำกัดในกรณีที่

$N_j \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ ดังนี้

$$cost += g_s * (S_i - N_j)$$

จะได้

$$cost = 0 + 2 * (3 - 5) = -4$$

และในส่วนของการเปลี่ยนล๊อคเกอร์ไม่มีค่าคงโทษเนื่องจากการไม่มีการเปลี่ยนล๊อคเกอร์

ของ VMs

4.2 สำหรับกรณีที่แย่ที่สุด (Worst Case)

1. The security Levels of VMs new: [1, 3, 5, 5, 5, 5]
2. The performance Levels of VMs new: [1, 3, 4, 4, 3, 3]
3. The number of workload-free Logger: 9.0
4. The penalty value: -24.0
5. The penalty swap: -1.0
6. The optimal value: -16.0
7. Old Solution ==>
8. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
9. U1[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
10. U2[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
11. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
12. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
13. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
14. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
15. New Solution ==>
16. L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
17. U1[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
18. U2[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
17. U3[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
18. U4[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
19. U5[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
20. U6[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

ภาพ 53 แสดงตัวอย่างผลลัพธ์วิธีการในการจัดสรรระบบบันทึกเหตุการณ์ในกรณีที่มีผู้ใช้บริการปรับระดับความปลอดภัยและระดับประสิทธิภาพ กรณีที่แย่ที่สุด

ตรวจสอบการผิดข้อจำกัด

กรณีผิดข้อจำกัดระดับความปลอดภัยและประสิทธิภาพหรือไม่

1. Logger4 (L4) มีสมาชิกของ VM 6 VMs ได้แก่ VM1 ถึง VM6 เมื่อนำจำนวน VM ของ Logger4 ไปตรวจสอบในฟังก์ชันการปรับ ดังสมการ (4) จะได้

1.1 VM1

$$1 \leq 6 \leq 1 \quad \text{มีค่าการปรับ} = -10 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้ดังนี้

$$\text{cost} = 0 + 2 * (1 - 6) = -10$$

1.2 VM2

$$3 \leq 6 \leq 3 \quad \text{มีค่าการปรับ} = -6 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีที่}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -10 + 2 * (3 - 6) = -16$$

1.3 VM3

$$4 \leq 6 \not\leq 5 \quad \text{มีค่าการปรับ} = -2 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีนี้}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -16 + 2 * (5 - 6) = -18$$

1.4 VM4

$$4 \leq 6 \not\leq 5 \quad \text{มีค่าการปรับ} = -2 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีนี้}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -18 + 2 * (5 - 6) = -20$$

1.5 VM5

$$3 \leq 6 \not\leq 5 \quad \text{มีค่าการปรับ} = -2 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีนี้}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -20 + 2 * (5 - 6) = -22$$

1.5 VM6

$$3 \leq 6 \not\leq 5 \quad \text{มีค่าการปรับ} = -2 \quad \text{เนื่องจากผิดข้อจำกัดในกรณีนี้}$$

$N(L_j) \not\leq S_i$ จะพบว่า จำนวน VM มากกว่าระดับความปลอดภัย จึงคำนวณฟังก์ชันการปรับได้
ดังนี้

$$\text{cost} = -22 + 2 * (5 - 6) = -24$$

กรณีที่มีการเปลี่ยนล็อกเกอร์หรือไม่

โดยทำการตรวจสอบว่า VM มีการเปลี่ยนล็อกเกอร์หรือไม่ ในสมการ (7) จะพบว่าผล
คูณของเมทริกซ์ในรูปแบบการจัดสรรระบบบันทึกเหตุการณ์แบบเก่ากับรูปแบบการจัดสรรระบบ
บันทึกเหตุการณ์แบบใหม่ การคูณแบบสมาชิก มีค่าเป็น 1 ซึ่งหมายความว่า VMs ไม่มีการเปลี่ยนล็อก
เกอร์ แต่ถ้ามีค่าเป็น 0 ซึ่งหมายความว่า VMs มีการเปลี่ยนล็อกเกอร์ ซึ่งนำค่าที่ได้มาคำนวณหาค่า
Scost ดังสมการ (8) ได้ดังนี้

$$\begin{aligned} \text{Scost} = & (((1 - 0) \times 0.5) + ((1 - 0) \times 0.5) + ((1 - 1) \times 0.5) \\ & + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5) + ((1 - 1) \times 0.5)) \end{aligned}$$

$$Scost = (0.5 + 0.5 + 0 + 0 + 0 + 0) = 1.0$$

ดังนั้นจะสรุปได้ว่าการทดสอบในกรณีที่มีผู้ใช้บริการยกเลิกในกรณีที่แย่ที่สุดพบการผิดข้อจำกัดในเรื่องของระดับความปลอดภัยและประสิทธิภาพ -24 และการเปลี่ยนสล็อตเกอร์ -1.0

ผู้วิจัยจึงสามารถสรุปตารางเปรียบเทียบการทำงานของโมเดลทางคณิตศาสตร์การ จัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ในกรณีที่ดีที่สุดและในกรณีที่แย่ที่สุด ได้ดังตาราง

ตาราง 9 แสดงเปรียบเทียบการทำงานในกรณี best case และ worst case แบบ unconstrained optimization สำหรับการปรับรูปแบบการทำงาน

ชุดข้อมูล	Best Case				Worst Case			
	Workload-free Logger	Penalty value	Penalty swap	Optimal value Total	Workload-free Logger	Penalty value	Penalty swap	Optimal value Total
ชุดที่ 4	8.0	0.0	0.0	8.0	5.0	-10.0	-2.5	-7.5
ชุดที่ 5	6.0	0.0	0.0	6.0	8.0	-49.0	-1.0	-42.0
ชุดที่ 6	6.0	-2.0	0.0	4.0	9.0	-50.0	-1.0	-42.0
ชุดที่ 7	7.0	-2.0	0.0	5.0	9.0	-24.0	-1.0	-16.0

สามารถดูผลการทดลองการหาวิธีการจัดสรรระบบบันทึกเหตุการณ์ได้ที่ <http://elearning.cpu.ac.th/eventmodel/>

บทที่ 5

บทสรุป

การประมวลผลแบบกลุ่มเมฆหรือคลาวด์คอมพิวติ้ง (Cloud Computing) หรือ คลาวด์ (Cloud) คลาวด์เป็นเทคโนโลยีที่มีการเปลี่ยนวิธีการสร้างและการใช้บริการข้อมูล รูปแบบการทำงานของคลาวด์นั้นองค์กรสามารถสร้างทรัพยากรการประมวลผลได้ตามความต้องการ และสามารถประกอบเป็นแอปพลิเคชันได้อย่างรวดเร็ว (Cable, 2015) คลาวด์เป็นการใช้เทคโนโลยีการจำลองเสมือนเพื่อให้ตัวเองสามารถจัดเก็บและประมวลผลข้อมูลขนาดใหญ่ได้ (Bhawiyuga et al., 2019) ซึ่งคลาวด์มีความสำคัญมากขึ้นสำหรับเทคโนโลยีสารสนเทศ หรือไอที (Alshamrani, 2019; Kaur & Kaur, 2017) ปัจจัยหนึ่งที่ทำให้แอปพลิเคชันอัจฉริยะนั้นสามารถพัฒนาชีวิตความเป็นอยู่ของมนุษย์ได้ นั่นก็คือระบบคลาวด์ ซึ่งสามารถจัดเก็บและประมวลผลข้อมูลซึ่งเกิดมาจากกิจกรรมของมนุษย์ อย่างไรก็ตาม ความปลอดภัยของข้อมูลนี้ก็มีความสำคัญเช่นกัน โดยเฉพาะการให้บริการแบบ IaaS คลาวด์ เช่น Amazon Elastic Compute Cloud (Chaisiri, 2011) มุ่งเน้นไปที่การนำเสนอเครื่องเสมือน (Virtual Machine: VM) ให้แก่ลูกค้าที่ใช้บริการ IaaS ลูกค้าสามารถเช่าเครื่องเสมือนและสามารถเข้าถึงโดยผ่านทางอินเทอร์เน็ต ซึ่งต้องคำนึงถึงในเรื่องของความปลอดภัยของคลาวด์โดย Cloud Security Alliance (CSA, 2022) ได้รายงานภัยคุกคามของการประมวลผลคลาวด์ในรายงาน Top Threats to Cloud Computing Pandemic Eleven และในรายงานของ Cyber Security Insiders เรื่อง 2022 Cloud Security Report ที่ได้รายงานเรื่องภัยที่เกิดจากการให้บริการคลาวด์ อย่างไรก็ตาม ปัญหาความปลอดภัยของระบบคลาวด์ IaaS นั้นสำคัญมากเมื่อถูกนำไปใช้งานจริงและจำเป็นต้องได้รับการบรรเทาเมื่อเกิดปัญหา จึงทำให้มีหน่วยงานที่ได้มีการดำเนินการด้านการวิจัยที่เกี่ยวกับการความปลอดภัยบนคลาวด์ขึ้น เช่น Cyber Security Insiders, องค์กร Cloud Security Alliance (CSA) ที่มีนักวิจัยที่ทำงานวิจัยเกี่ยวกับความปลอดภัยบนคลาวด์ เพื่อหาแนวทางในการป้องกันหรือบรรเทาผลกระทบที่เกิดจากการใช้งานบนระบบคลาวด์ เช่น ในงานวิจัยของ Ko & Will (2014), Shamseddine et al. (2022), Fargo et al. (2020) ที่ได้มีการใช้ Virtual Machine Introspection (VMI) ในรูปแบบเครื่องมือที่แตกต่างกันไปมาใช้ในการตรวจสอบหาความผิดปกติของการทำงานจากผู้ไม่ประสงค์ดี

ในงานวิจัยของ Auxsorn (2020), Chan-In (2017), Jaiboo (2020), และ Wongthai (2017) ได้มีการใช้ระบบบันทึกเหตุการณ์ ซึ่งเป็นวิธีการหนึ่งที่สำคัญสำหรับสำหรับช่วยบรรเทาปัจจัยเสี่ยงที่อาจก่อให้เกิดภัยคุกคามขึ้นต่อคลาวด์ โดยภายในระบบบันทึกเหตุการณ์ได้จัดเก็บหลักฐาน log file ไว้เป็นหลักฐานและหลักฐานจาก log file สามารถนำไปช่วยหาบุคคลที่กระทำผิดมารับผิดชอบได้ โดยการทำงานของระบบ Logging System ใน IaaS คลาวด์ แบ่งการทำงานออกเป็น 2 ส่วน ได้แก่ 1) ส่วนที่เป็นผู้ให้บริการ (Host) ในส่วนนี้จะได้ทำการติดตั้งล็อกเกอร์ (Logger) ซึ่งเป็นโปรแกรมสำหรับใช้ในการตรวจจับบันทึกเหตุการณ์ที่เกิดขึ้นใน VM ซึ่งจะเป็นการทดลองการทำงานในการตรวจจับโปรแกรมการทำงานของไฟล์ในเครื่อง VM 1 เครื่อง โดยล็อกเกอร์ 1 ล็อกเกอร์ ประกอบกับการจัดสรรทรัพยากรทางด้านคลาวด์ที่มุ่งเน้นไปจัดสรรในเรื่องของ CPU, RAM, Harddisk เป็นต้น ดังในงานวิจัยของ Ye et al. (2011) ได้นำเสนอเฟรมเวิร์กในการย้าย VM หลายตัวตามการจองทรัพยากรที่แตกต่างกัน เพื่อให้มีประสิทธิภาพในการย้ายข้อมูลแบบขนานและการรับรู้ภาระงาน งานวิจัยของ Zhao et al. (2015) ได้นำเสนอวิธีการจัดวาง VM ตามประสิทธิภาพของการบริการ โดยนำลักษณะการใช้ทรัพยากรมาใช้ในการพิจารณาในการดูว่า VM จะไปอยู่ที่โฮสต์ใดที่จะเกิดความเหมาะสมเพื่อรับประกันประสิทธิภาพของ VM และรับประกันคุณภาพประสบการณ์ของลูกค้า (QoE) งานวิจัยของ Malekloo et al. (2018) ได้นำเสนอแนวทาง Multi-objective Ant Colony Optimization (MACO) ในการรวบรวมและจัดวาง VM โดยมีเป้าหมายเพื่อลดการใช้พลังงาน การสิ้นเปลืองทรัพยากรหน่วยประมวลผล และต้นทุนการสื่อสาร มีเป้าหมายเพื่อลดการละเมิดข้อตกลงการให้บริการระหว่างผู้ให้บริการและผู้ให้บริการ (Service Level Agreement: SLA) การย้าย VM และจำนวนโฮสต์ที่ใช้งานอยู่ เป็นต้นผู้วิจัยจึงได้นำเทคนิคกระบวนการหาจุดที่เหมาะสม (Optimization) มาใช้ในการจัดสรรระบบบันทึกเหตุการณ์ให้กับ VMs ต่าง ๆ ภายใต้ความต้องการของผู้ให้บริการและผู้ให้บริการ โดยให้เครื่องคอมพิวเตอร์สามารถทำงานได้เต็มประสิทธิภาพด้วยกระบวนการหาจุดที่เหมาะสมมาใช้ในการหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการเพื่อให้เครื่องคอมพิวเตอร์สามารถทำงานได้เต็มประสิทธิภาพมากที่สุด โดยงานวิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อ

1. เพื่อวางรูปแบบการทดสอบความสามารถในการรองรับการทำงานของระบบบันทึกเหตุการณ์
2. เพื่อสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บนระบบการประมวลผลแบบกลุ่มเมฆ ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (IaaS) ที่มีประสิทธิภาพ
3. เพื่อสร้างการปรับปรุงแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

สรุปผลการวิจัย

จากวัตถุประสงค์การวิจัย และปัญหาหลักของการวิจัยในบทที่ 1 นั้น ผู้วิจัยขอสรุปผลการวิจัยและอภิปรายผลการวิจัย โดยแบ่งเป็นข้อตามวัตถุประสงค์ ดังนี้

1. เพื่อวางรูปแบบการทดสอบความสามารถในการรองรับการทำงานของระบบบันทึกเหตุการณ์

ผู้วิจัยได้ดำเนินการสร้างระบบ IaaS คลาวด์ที่ติดตั้งระบบบันทึกเหตุการณ์ หรือล็อกเกอร์ และจำลองการทำงานของในการตรวจจับโปรเซสการทำงานของไฟล์ในเครื่อง VM และทำการวัดปริมาณการใช้งาน CPU และ RAM ของฝั่ง Host ที่เป็นเครื่องของผู้ให้บริการโดยให้มีการทำงานทั้งฝั่งของ VM และล็อกเกอร์ในฝั่งของ Host โดยให้มีการรันการทำงานเป็นคู่ตั้งแต่ 1 คู่ ไปจนถึง 5 คู่ ทั้งนี้ที่ได้จำนวน 5 คู่เนื่องจากมีข้อจำกัดในเรื่องของสเปกของเครื่องคอมพิวเตอร์ที่ใช้ในการทดลอง จึงพบว่า ปริมาณการใช้งาน CPU ซึ่งจะเห็นว่ามีการทำงานของ Logger กับ VM จำนวน 1 คู่ ปริมาณการใช้งาน CPU อยู่ที่ 30.64% เมื่อมีการทำงานของ Logger กับ VM จำนวน 2 คู่ ปริมาณการใช้งาน CPU อยู่ที่ 40.81% เมื่อมีการทำงานของ Logger กับ VM จำนวน 3 คู่ ปริมาณการใช้งาน CPU อยู่ที่ 49.64% เมื่อเปิดการใช้งานพร้อมกันจำนวน 5 คู่ จะทำให้เครื่องของ VM และ Host มีการทำงานที่ช้าลง และปริมาณการใช้งาน RAM ซึ่งจะเห็นว่ามีการทำงานของ Logger กับ VM จำนวน 1 คู่ ปริมาณการใช้งาน RAM อยู่ที่ 10.15% เมื่อมีการทำงานของ Logger กับ VM จำนวน 2 คู่ ปริมาณการใช้งาน RAM อยู่ที่ 10.83% เมื่อมีการทำงานของ Logger กับ VM จำนวน 3 คู่ ปริมาณการใช้งาน RAM อยู่ที่ 11.51% เมื่อเปิดการใช้งานพร้อมกันจำนวน 5 คู่ จะทำให้เครื่องของ VM และ Host มีการทำงานที่ช้าลง จึงสรุปได้ว่าการใช้จำนวนล็อกเกอร์จำนวนมากจะทำให้ปริมาณการใช้งาน CPU เพิ่มขึ้นด้วย ซึ่งตรงกับที่ Microsoft (2022) ได้กล่าวไว้ว่า สำหรับการใช้งานทรัพยากรของคอมพิวเตอร์นั้น เมื่อพบว่าปริมาณการใช้งานของ CPU มีระดับสูง 80% ขึ้นไป เป็นระยะเวลาานประสิทธิภาพของระบบหรือแอปพลิเคชันจะได้รับผลกระทบด้วย และ Csáti et al. (2022) ที่ได้ทดลองปริมาณการใช้งาน CPU และ RAM ตามขนาดไฟล์ของฝั่งผู้ใช้งาน พบว่า การทำงานด้วย Filebeat ที่ logs ขนาด 0.5-5 kB, Elasticsearch ใช้ CPU เฉลี่ย 91% RAM 1.8 GiB ซึ่งจากการทดลองเมื่อผู้วิจัยทำการปล่อยให้ CPU ทำงานหนักอย่างต่อเนื่องถึงแม้ว่าปริมาณการใช้งาน CPU จะไม่ถึง 80% ก็ตาม เครื่องคอมพิวเตอร์นั้นไม่สามารถทำต่อไปได้

2. เพื่อสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บนระบบการประมวลผลแบบกลุ่มเมฆ ประเภทการให้บริการโครงสร้างพื้นฐานทางคอมพิวเตอร์ (IaaS) ที่มีประสิทธิภาพ

ผู้วิจัยจึงนำข้อมูลจากการทดลองในวัตถุประสงค์ข้อที่ 1 มาทำการสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ เพื่อจะลดปริมาณการใช้งานของล็อกเกอร์ให้น้อยที่สุด และสามารถจัดการการทำงานใน VM ได้มากที่สุด และจากงานวิจัยของ Noor et al. (2019) ที่ดึงข้อมูลจาก Virtual Machine หลายเครื่องมาเก็บไว้ในฐานข้อมูลโดยผ่าน Monitoring Manager งานวิจัยของ Borisaniya & Patel (2019) ที่ได้ใช้งาน Nitro ที่ติดตั้งไว้บน hypervisor ในการดึงข้อมูลของโปรเซสที่กำลังทำงานภายในของ VMs โดยใช้หลักการของการทำ multi-thread ในการดึงข้อมูลโปรเซสจาก VM หลายเครื่อง งานวิจัยของ Shamseddine et al. (2022) ที่ใช้วิธีการแบบ Divide-and-Conquer มาใช้ในการตรวจสอบการทำงานของ VM และในการวัดประสิทธิภาพ ได้มีการใช้งานระบบบันทึกเหตุการณ์ที่แตกต่างกันไปในการดึงข้อมูลโปรเซสการทำงานใน VM จากหลายเครื่อง แต่ไม่ได้คำนึงถึงความต้องการของผู้ใช้บริการ และผู้ให้บริการ จึงนำไปสู่การดำเนินงานวิจัยในวัตถุประสงค์ข้อที่ 1 ในการสร้างโมเดลทางคณิตศาสตร์สำหรับการจัดสรรระบบบันทึกเหตุการณ์ โดยนำข้อมูลการทดลองของ Auxsom (2020) มาใช้ในการออกแบบการสร้างโมเดลทางคณิตศาสตร์ด้วยโมเดลทางคณิตศาสตร์ที่มีการหาค่ามากที่สุดของล็อกเกอร์ที่ไม่ได้ถูกใช้งาน ด้วยกระบวนการในการหาค่าที่เหมาะสมด้วยเทคนิค Random Search Algorithm จึงได้ผลลัพธ์ดังสมการ ดังนี้

โมเดลทางคณิตศาสตร์แบบมีข้อจำกัด (Constrained Optimization)

$$\text{maximize } f(X) = \sum_{i=1}^n O_i(X)$$

โมเดลทางคณิตศาสตร์แบบไม่มีข้อจำกัด (Unconstrained Optimization)

$$\text{maximize } f(X) = \left(\sum_{j=1}^n O_j(X) \right) + \sum_{i=1}^m \text{cost}(U_i)$$

โดยในงานวิทยานิพนธ์นี้ได้นำโมเดลทางคณิตศาสตร์แบบไม่มีข้อจำกัด โดยการใช้ฟังก์ชันการปรับไปใช้งาน เนื่องจากสามารถรองรับความต้องการที่ซับซ้อนให้สามารถสร้างวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ได้ และจากผลการทดลองในบทที่ 4 ผลการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ นั้น โมเดลทางคณิตศาสตร์ที่ได้ทำการพัฒนาด้วยภาษา Python ยังสามารถที่รองรับการคำนวณหาวิธีการที่เหมาะสมในลักษณะของสภาพแวดล้อมของเครื่องคอมพิวเตอร์ที่เปลี่ยนแปลงไป โดยดูจากจำนวนล็อกเกอร์และจำนวน VM ที่สามารถรองรับได้ โดยการทดสอบดังนี้

- 2.1 การทดสอบข้อมูลนำเข้าของจำนวนล็อกเกอร์ และ VM เท่ากัน
- 2.2 การทดสอบข้อมูลนำเข้าของจำนวนล็อกเกอร์น้อยกว่าจำนวน VM
- 2.3 การทดสอบข้อมูลนำเข้าของจำนวนล็อกเกอร์มากกว่าจำนวน VM

พบว่า การทดสอบทั้ง 3 รูปแบบ โมเดลทางคณิตศาสตร์ที่ผู้วิจัยได้สร้างขึ้นมานั้นสามารถคำนวณหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่ใช้ล็อกเกอร์น้อยตัวที่สุด เพื่อที่จะส่งผลให้ปริมาณการใช้งาน CPU และ RAM ของเครื่องคอมพิวเตอร์ฝั่งผู้ให้บริการสามารถใช้งานได้คุ้มค่าและเกิดประโยชน์มากที่สุด สิ่งที่เราควรระวังสำหรับผู้ให้บริการ IaaS คลาวด์ ในกรณีนี้ให้ผู้ใช้บริการเลือกระดับความปลอดภัยและระดับประสิทธิภาพเองนั้น คือเมื่อได้วิธีการจัดสรรระบบบันทึกเหตุการณ์ด้วยการใช้ฟังก์ชันการปรับแก้แล้วนั้น วิธีการจัดสรรระบบบันทึกเหตุการณ์อาจจะไม่ตรงกับความต้องการของผู้ใช้งานทั้งหมด ดังนั้นจึงพึงระวังในเรื่องของค่า Penalty Value ที่ได้จากการสร้างวิธีการจัดสรรระบบบันทึกเหตุการณ์ เพื่อนำไปสู่การจัดสรรใหม่ในกรณีที่มีการปรับเปลี่ยนผู้ใช้บริการ

3. เพื่อสร้างการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์

จากการสร้างโมเดลทางคณิตศาสตร์ในวัตถุประสงค์ข้อที่ 2 แล้วนั้น ผู้วิจัยได้ดำเนินการพัฒนาต่อในกรณีที่มีการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ใช้บริการ จึงได้โมเดลทางคณิตศาสตร์ ดังนี้

$$\text{maximize } f(X) = \left(\sum_{j=1}^n O_j(X) \right) + \sum_{i=1}^m \text{cost}(U_i) + \sum_{i=1}^m \text{Scost}(U_i)$$

เมื่อผู้วิจัยได้ทำการออกแบบฟังก์ชันจุดประสงค์แล้ว ผู้วิจัยได้นำเอาหลักการในการทำงานไปเขียนโปรแกรมด้วยภาษา Python โดยใช้วิธีการหาค่าที่เหมาะสมด้วยเทคนิค Random Search และทำการทดสอบการทำงานของโปรแกรม โดยแบ่งออกเป็น

- 3.1 การทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก
- 3.2 การทดสอบในกรณีที่มีผู้ใช้บริการใหม่
- 3.3 การทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่
- 3.4 การทดสอบในกรณีที่มีผู้ใช้บริการมีการปรับระดับความปลอดภัย และระดับประสิทธิภาพ

พบว่า การทดสอบทั้ง 4 รูปแบบ โมเดลทางคณิตศาสตร์ที่ผู้วิจัยได้สร้างขึ้นมานั้นสามารถคำนวณหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่ใช้ล็อกเกอร์น้อยตัวที่สุดได้ ตามผู้ใช้บริการที่

เปลี่ยนแปลงไป โดยไม่เกิดผลกระทบต่อผู้ใช้บริการเนื่องจากไม่มีการย้ายสล็อตเกอร์ และเกิดประโยชน์กับผู้ให้บริการอีกด้วย

อภิปรายผล

ในงานวิทยานิพนธ์นี้เป็นการสร้างโมเดลทางคณิตศาสตร์ที่เหมาะสมของระบบบันทึกเหตุการณ์สำหรับการจัดการความแม่นยำในการตรวจจับของระบบบันทึกเหตุการณ์และประสิทธิภาพการทำงานของคอมพิวเตอร์เสมือนของผู้ใช้บริการ ในการประมวลผลแบบกลุ่มเมฆ โดยจุดมุ่งหมายของแบบจำลองทางคณิตศาสตร์ที่พัฒนาขึ้นในงานวิจัยนี้คือการลดการใช้ตัวบันทึกเหตุการณ์ให้เหลือน้อยที่สุด เพื่อที่จะส่งผลให้ปริมาณการใช้งาน CPU และ RAM ของเครื่องคอมพิวเตอร์ฝั่งผู้ให้บริการสามารถใช้งานได้คุ้มค่าและเกิดประโยชน์มากที่สุด อย่างไรก็ตาม ผู้ให้บริการ IaaS คลาวด์ ต้องคำนึงถึงการแลกเปลี่ยนระหว่างความปลอดภัยและประสิทธิภาพ เมื่อให้อิสระแก่ลูกค้าในการเลือกระดับความปลอดภัยและประสิทธิภาพ การจัดสรรระบบบันทึกเหตุการณ์อาจไม่เหมาะกับลูกค้าทุกราย และอาจต้องปรับค่าปรับเพื่อให้แน่ใจว่ามีการจัดสรรที่เหมาะสมที่สุดสำหรับทุกราย นอกจากนี้หากความต้องการของลูกค้าเปลี่ยนแปลง อาจจำเป็นต้องมีการจัดสรรใหม่ ซึ่งจะนำไปตามวัตถุประสงค์ข้อที่ 3 ของงานวิทยานิพนธ์

จากการทดลองในวางรูปแบบการทดสอบความสามารถในการรองรับการทำงานของระบบบันทึกเหตุการณ์ ในวัตถุประสงค์ข้อที่ 1 ในจัดวางรูปแบบการทดสอบนั้นโดยวางรูปแบบการทดสอบระหว่างระบบบันทึกเหตุการณ์กับ VM ในลักษณะการทำงานเป็นคู่ ตั้งแต่ 1 คู่ ไปจนถึง 5 คู่ ตามที่เครื่องคอมพิวเตอร์สามารถรองรับการทำงานได้ และทำการวัดปริมาณการใช้งานของ CPU และ RAM ในเครื่องคอมพิวเตอร์ของผู้ใช้งานนั้น โดยที่มีการทำของระบบบันทึกเหตุการณ์ได้ทำงานอยู่ ซึ่งจะเห็นได้ว่าปริมาณการใช้งานของ CPU และ RAM เพิ่มขึ้น เมื่อมีการใช้งานระบบบันทึกเหตุการณ์หลายตัว และเมื่อมีการทำงานที่ใช้ระยะเวลาเพิ่มขึ้นส่งผลให้เครื่องคอมพิวเตอร์ Host ไม่สามารถทำงานต่อไปได้ จากการทดลองพบว่าเป็นที่ปริมาณการใช้งาน CPU ที่ส่งผลทำให้เครื่องคอมพิวเตอร์ Host ไม่สามารถทำงานต่อไปได้ ทั้งนี้จะเห็นว่าคุณลักษณะของเครื่องคอมพิวเตอร์ที่ใช้ในการทดลองสามารถรองรับการทำงานได้สูงสุดเพียง 5 เครื่องเท่านั้น ซึ่งเป็นข้อจำกัดในการทดลอง และพบได้ว่าระบบบันทึกเหตุการณ์ที่ใช้ในการทดลองนั้นมีการใช้งาน CPU ในปริมาณที่เพิ่มมากขึ้นอย่างเห็นได้ชัด เมื่อมีการใช้งานระบบบันทึกเหตุการณ์ให้ทำงานกับ VM หลาย ๆ ตัว ในส่วนของปริมาณการใช้งาน RAM นั้นก็มีปริมาณการใช้งานที่เพิ่มมากขึ้นเช่นกัน แต่เนื่องจากเป็นเพราะในเครื่องคอมพิวเตอร์ที่ใช้ในการทดลองมีปริมาณ RAM ที่มากจึงทำให้เห็นปริมาณการใช้งานที่ไม่สูงมากนัก แต่เมื่อเปรียบเทียบกับปริมาณการใช้งาน CPU นั้นจะเห็นว่าปริมาณการใช้งานเพิ่มขึ้นเป็น 2 เท่าเมื่อเปรียบเทียบกับก่อนการใช้งานระบบบันทึกเหตุการณ์ และหลังการใช้งานระบบบันทึกเหตุการณ์ และยัง

สอดคล้องกับการใช้งานของระบบ ถ้ายิ่ง VM ต่อ Host มากเท่าไร ยิ่งทำให้ประสิทธิภาพของ VM และ Host ลดลงด้วย โดยสามารถแก้ไขปัญหานี้ได้ด้วยการเพิ่มคุณลักษณะของฮาร์ดแวร์เพิ่มเข้าไป เพื่อให้เครื่องคอมพิวเตอร์ Host สามารถรองรับการทำงาน

จากการทดลองการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS ในวัตถุประสงค์ข้อที่ 1 ผู้วิจัยได้คำนึงถึงในเรื่องของความแม่นยำในการตรวจจับซึ่งนำไปสู่ความปลอดภัยและในเรื่องของประสิทธิภาพการทำงานของ VM โดยการใช้วิธีการการค้นหาแบบสุ่มในการค้นหาวิธีการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมบน IaaS คลาวด์ นั้น ผู้วิจัยได้ทำการทดสอบด้วยข้อมูลนำเข้าที่มีความแตกต่างกันได้แก่ ข้อมูลนำเข้าของจำนวนบล็อกเกอร์ และ VM เท่ากัน ข้อมูลนำเข้าของจำนวนบล็อกเกอร์น้อยกว่าจำนวนของ VM และข้อมูลนำเข้าของจำนวนบล็อกเกอร์มากกว่าจำนวนของ VM ซึ่งแสดงให้เห็นว่าโมเดลทางคณิตศาสตร์ที่ผู้วิจัยได้สร้างขึ้นมานั้นสามารถรองรับการทำงานของคุณลักษณะของเครื่องคอมพิวเตอร์ที่นำมาใช้ในการเป็นเครื่องโฮสต์ได้หลายรูปแบบ ไม่ว่าจะมีความสามารถในการรองรับได้มากน้อยเท่าไร โมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ที่ผู้วิจัยได้สร้างขึ้นนี้สามารถรองรับการทำงานได้ครบทุกรูปแบบ ซึ่งสอดคล้องกับการหากระบวนการหาค่าที่เหมาะสม ในการหาค่ามากที่สุดของการทำงาน โดยแก้ปัญหาในกรณีที่ใช้ logger 1 ตัว กับ VM 1 ตัวได้ โดยขึ้นอยู่กับความต้องการของผู้ใช้งาน

คุณประโยชน์ที่ได้จากการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์ โดยผู้ให้บริการสามารถที่จะทำไปสร้างเป็นแพ็คเกจข้อเสนอขาย IaaS คลาวด์ที่เสนอขายพร้อมด้วยระบบการตรวจจับโปรเซสการทำงานใน VM ให้กับผู้ใช้บริการได้ รวมถึงผู้ใช้บริการยังสามารถที่จะกำหนดระดับความต้องการของตนเองในด้านความปลอดภัย และประสิทธิภาพในการทำงานของเครื่อง VM ที่ตนเองได้เช่าว่าต้องการใช้งานในระดับใดได้บ้าง ซึ่งต่างจากการให้บริการที่มีอยู่ในปัจจุบันของผู้ให้บริการคลาวด์เจ้าต่าง ๆ หรือแม้สำหรับผู้ที่ต้องการที่จะเป็นผู้ประกอบการรายใหม่ของการให้บริการคลาวด์ โดยอาจจะเริ่มที่การให้เช่าเครื่องคอมพิวเตอร์เสมือนขนาดเล็ก พร้อมด้วยระบบบันทึกเหตุการณ์เพื่อการตรวจจับโปรเซสการทำงาน แล้วนำโมเดลทางคณิตศาสตร์นี้ไปใช้งานก็สามารถทำการคำนวณหาวิธีการจัดสรรระบบบันทึกเหตุการณ์ที่เหมาะสมกับเครื่องคอมพิวเตอร์ที่ใช้งานได้

จากการทดลองการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS ในวัตถุประสงค์ข้อที่ 2 ผู้วิจัยได้คำนึงถึงความเปลี่ยนแปลงของการใช้บริการของผู้ใช้บริการที่อาจมีการเปลี่ยนแปลงตามความต้องการที่จะเกิดขึ้นในอนาคต จึงได้มีการออกแบบโมเดลทางคณิตศาสตร์ที่มีการเพิ่มในส่วนของฟังก์ชันการปรับในส่วนของการที่จะมีการสลับเปลี่ยนบล็อกเกอร์ เพื่อหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ใหม่ ผู้วิจัยได้ทำการทดสอบด้วยกรณีที่มีความแตกต่างกันได้แก่ การทดสอบในกรณีที่มีผู้ใช้บริการยกเลิก การทดสอบในกรณีที่มีผู้ใช้บริการใหม่ การทดสอบในกรณีที่มี

มีผู้ใช้บริการยกเลิก และมีผู้ใช้บริการใหม่ และการทดสอบในกรณีที่ผู้ใช้บริการมีการปรับระดับความปลอดภัย และระดับประสิทธิภาพ จากตัวอย่างในการทดลองนั้นจะไม่มีมีการปรับเปลี่ยนล็อกเกอร์จึงทำให้ค่าโดนลงโทษของการเปลี่ยนล็อกเกอร์นั้นเป็น 0 เป็นเพราะข้อมูลที่ใช้ในการทดลองนั้นสามารถหาวิธีการในการจัดสรรระบบบันทึกเหตุการณ์ที่ไม่กระทบกับการทำงานเดิมจึงทำให้ไม่มีมีการเปลี่ยนล็อกเกอร์ ซึ่งนับได้ว่าเป็นผลดีกับผู้ใช้บริการที่จะไม่เกิดปัญหาในการปรับตั้งค่าการทำงาน แต่สามารถที่จะรองรับความต้องการของผู้ใช้งานได้

สำหรับการนำทรัพยากรในด้านอื่น ๆ มาประกอบในการสร้างโมเดลทางคณิตศาสตร์นั้น ไม่ว่าจะเป็น CPU, RAM หรือ Network โดยสามารถนำมาใช้ในการพิจารณาร่วมกับโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์ในการพิจารณาในส่วนของการวางเครื่อง VM ให้กับ Physical Machine (PM) ของผู้ให้บริการได้ โดยการใช้วิธีการทำที่เป็น Multi-Objective Optimization ที่เป็นวิธีการในการหาค่าที่เหมาะสมของปัญหาที่มากกว่าหนึ่งปัญหา (Gunantara, 2018) แบบ Nonlinear Optimization Problems ในการเพิ่มส่วนของการสร้างวิธีการในการจัดสรร VM ให้ไปอยู่ใน PM ใดที่เหมาะสมและทำการพิจารณาในส่วนที่สองได้แก่การพิจารณา VM ให้ไปอยู่ภายใต้การดูแลของล็อกเกอร์ตัวใด

พิจารณาจากการใช้พลังงานที่น้อยที่สุดสำหรับการจัดสรร VM ไปยัง PM โดยขึ้นอยู่กับการใช้งาน CPU จะได้สมการ โดยอ้างอิงจากงานวิจัยของ Askarizade et al. (2019)

$$u^{cpu} = \frac{h_u^{cpu}}{h_c^{cpu}}$$

$$P(u^{cpu}) = k_j \cdot e_j^{max} + (1 - k_j) \cdot e_j^{max} \cdot u^{cpu}$$

$$\text{minimize } f_1(X) = \frac{1}{N} \sum_1^N P(u^{cpu})$$

โดยที่ $h_j \in PM_{dataset}$, $1 \leq j \leq N$

u^{cpu} คือ อัตราการใช้งาน CPU

h_u^{cpu} คือ ความต้องการของ PM ในกรณีที่สามารถรองรับการทำงานรวมได้มากที่สุด

h_c^{cpu} คือ ปริมาณหน่วยประมวลผลของ PM (busy time และ idle time)

$P(u^{cpu})$ คือ พลังงานการใช้งาน CPU ในส่วนของ PM

k_j คือ สัดส่วนของการใช้พลังงานของเครื่อง PM ในระหว่างที่เซิร์ฟเวอร์ว่างงาน

e_j^{max} คือ ค่าสูงสุดของการใช้พลังงานในหน่วยประมวลผลของ PM ระหว่างที่มีการใช้งาน

แบบเต็มประสิทธิภาพของเครื่องเซิร์ฟเวอร์

เมื่อได้วิธีการในการจัดสรร VM ไปยัง PM แล้ว นำเซตของวิธีการที่ได้มาเป็นข้อมูลนำเข้าในส่วนของจำนวน VM โดยที่กำหนดข้อจำกัดเป็นระดับความปลอดภัยของไฟล์และระดับประสิทธิภาพ

ในการทำงานของ VM เพื่อทำการหาจำนวนล็อกเกอร์ที่ไม่ได้ถูกใช้งานมากที่สุด และวิธีการจัดสรร VM ให้กับล็อกเกอร์ ตามสมการ

$$\text{maximize } f_2(X) = \left(\sum_{j=1}^n O_j(X) \right) + \sum_{i=1}^m \text{cost}(U_i) + \sum_{i=1}^m \text{Scost}(U_i)$$

จะได้ $f(X) = [f_1(X), f_2(X)]$ เป็นต้น โดยสามารถนำไปต่อยอดในการพัฒนา งานวิจัยต่อไปได้โดยใช้เทคนิคในการหาวิธีการในการแก้ปัญหาที่มีประสิทธิภาพสูงขึ้นไป

คุณประโยชน์ที่ได้จากการสร้างโมเดลทางคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์ที่มีการปรับรูปแบบการทำงานที่เหมาะสมกับผู้ให้บริการและผู้ให้บริการในการจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ โดยผู้ให้บริการสามารถปรับเปลี่ยนวิธีการจัดสรรได้ในกรณีที่ผู้ใช้บริการเข้ามาใช้งานใหม่ หรือมีผู้ใช้บริการเดิมยกเลิกการใช้บริการ การทำงานของระบบบันทึกเหตุการณ์ที่ทำงานกับ VM เดิมอยู่แล้วจะไม่มีผลกระทบกับการทำงาน บริษัทหรือองค์กรจะสามารถยืดหยุ่นกับระบบการจัดสรรระบบบันทึกเหตุการณ์นี้ได้ โดยเมื่อนำไปใช้งานจริงนั้นผู้ให้บริการ IaaS คลาวด์สามารถทำการติดตั้งระบบบันทึกเหตุการณ์ ซึ่งเป็นการตรวจสอบการเข้าถึงไฟล์ โดยนำมาวิเคราะห์และแจ้งเตือนไปยังเจ้าของไฟล์ได้ ซึ่งแตกต่างจากผู้ให้บริการเดิมเช่น Amazon นั้นมีระบบการจำกัดการเข้าถึง ควบคุมด้วยการเข้ารหัสซึ่งเป็นรูปแบบของวิธีการป้องกัน (Preventive) แต่ในระบบนี้สามารถเพิ่มเติมในส่วนของการตรวจสอบ (Detective) เสริมเข้าไปในการให้บริการของ IaaS คลาวด์ได้ สำหรับการรองรับการทำงานของระบบบันทึกเหตุการณ์นั้นขึ้นอยู่กับคุณสมบัติของเครื่องคอมพิวเตอร์ที่เป็น Physical Machine ในการให้บริการ ซึ่งสามารถรองรับการขยายตัวเพิ่มได้ด้วยการปรับเปลี่ยนเครื่องคอมพิวเตอร์ให้สูงขึ้น

ข้อเสนอแนะ

จากการสร้างโมเดลคณิตศาสตร์การจัดสรรระบบบันทึกเหตุการณ์บน IaaS คลาวด์ และการทดสอบการทำงานในรูปแบบที่เปลี่ยนแปลงไป ผู้วิจัยมีข้อเสนอแนะ ดังนี้

1. การสร้างโมเดลคณิตศาสตร์ที่ผู้วิจัยได้สร้างขึ้นนั้นจะมีตัวแปรในการสร้างเพียงระดับความปลอดภัย ระดับประสิทธิภาพ จำนวนล็อกเกอร์ และจำนวน VM เท่านั้น ซึ่งสามารถที่จะนำเอาข้อมูลทางด้านฮาร์ดแวร์มาเป็นตัวแปรเพิ่มเติมในการสร้างโมเดลคณิตศาสตร์ เพื่อให้รองรับกับสภาพแวดล้อมของเครื่องคอมพิวเตอร์ที่ใช้งานจริงได้
2. การนำไปสู่การใช้งานเชิงพาณิชย์นั้น ควรจะต้องพิจารณาในเรื่องของสภาพแวดล้อมของฮาร์ดแวร์ที่ใช้งานจริงมาประกอบการพัฒนาได้

3. การสร้างโมเดลคณิตศาสตร์ อาจจะมีการกำหนดระดับความต้องการของผู้ใช้บริการในด้านอื่น ๆ ที่ผู้ใช้บริการต้องการ เช่น ในเรื่องของความเร็วในการถ่ายโอนข้อมูล การประมวลผล เป็นต้น

4. การเปรียบเทียบในการใช้วิธีการอื่น ๆ ในการวัดประสิทธิภาพการทำงานของโมเดลทางคณิตศาสตร์ที่นอกเหนือจาก Random Search อาทิ Greedy Algorithm, Particle Swarm Optimization, Ant Colony Optimization หรือเทคนิคอื่น ๆ

5. สำหรับการทดลองนี้เป็นการทำงานแค่เพียง Physical Machine เครื่องเดียว ถ้าในกรณีที่มีหลาย Physical Machine แล้ว VMs ที่สร้างนั้นไม่ระบุว่าจะไปอยู่ที่ Physical Machine ไหน สามารถขยายงานจากวิธีการนี้ในการจัดสรรให้ VMs ไปอยู่บน Physical Machine ที่เหมาะสมได้



บรรณานุกรม

- Al-Shaikh, A., Khattab, H., Sharieh, A., & Sleit, A. (2016). Resource utilization in cloud computing as an optimization problem. *International Journal of Advanced Computer Science and Applications*, 7(6).
doi:<https://doi.org/10.14569/IJACSA.2016.070643>
- Alam, T. (2020). Cloud Computing and its role in the Information Technology. *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, 1(2), 108-115.
- Aldossary, M. (2021). A Review of Dynamic Resource Management in Cloud Computing Environments. *Comput. Syst. Sci. Eng.*, 36(3), 461-476.
- Al-Muhammed, M. J., & Zitar, R. A. (2018). Probability-directed random search algorithm for unconstrained optimization problem. *Applied Soft Computing*, 71, 165-182.
- Al-Roomi, M., Al-Ebrahim, S., Buqrais, S., & Ahmad, I. (2013). Cloud computing pricing models: a survey. *International Journal of Grid and Distributed Computing*, 6(5), 93-106.
- Alshamrani, S. (2019). An efficient algorithm for monitoring virtual machines in clouds. *Bulletin of Electrical Engineering and Informatics*, 8(2), 699-705.
- Arron, F. (2022). *7 Different Types of Cloud Computing Structures*.
<https://www.uniprint.net/en/7-types-cloud-computing-structures/>
- Askarizade Haghighi, M., Maeen, M., & Haghparsat, M. (2019). An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing IaaS platforms: Energy efficient dynamic cloud resource management. *Wireless Personal Communications*, 104, 1367-1391.
- 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* (pp. 285-296). Singapore: Springer.
- 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.
- Bacanin, N., Bezdán, T., Tuba, E., Strumberger, I., Tuba, M., & Zivkovic, M. (2019, November). Task scheduling in cloud computing environment by grey wolf optimizer. In *2019 27th telecommunications forum (TELFOR)* (pp. 1-4). Belgrade, Serbia: IEEE.
- Basheer Riskhan, R. M. (2016). Virtual Machine Performance Approaches in the Online Education System. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1).
- Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear programming: theory and algorithms*. John Wiley & Sons.
- Berggren, J., & Karlsson, J. (2022). Differences in performance between containerization & virtualization: With a focus on HTTP requests.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 281-305.
- Bermejo, B., Guerrero, C., Lera, I., & Juiz, C. (2016). Cloud resource management to improve energy efficiency based on local nodes optimizations. *Procedia Computer Science*, 83, 878-885.
- Bhawiyuga, A., Kartikasari, D. P., Amron, K., Pratama, O. B., & Habibi, M. W. (2019). Architectural design of iot-cloud computing integration platform. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 17(3), 1399-1408.
- Bisschop, J. (2006). *AIMMS optimization modeling*. Lulu. com.
- Bokhari, M. U., Makki, Q., & Tamandani, Y. K. (2018). A survey on cloud computing. In *Big Data Analytics: Proceedings of CSI 2015* (pp. 149-164). Singapore: Springer
- Borisaniya, B., & Patel, D. (2019). Towards virtual machine introspection based security framework for cloud. *Sādhanā*, 44(2), 1-15.
- Bryant, R. E., David Richard, O. H., & David Richard, O. H. (2010). *Computer systems: a programmer's perspective* (Vol. 2). Upper Saddle River: Prentice Hall.
- Cable, P. T., Schear, N. A., Cunningham, R. K., Gadepally, V. N., Moyer, T. M., & Chaisiri, S., Kaewpuang, R., Lee, B. S., & Niyato, D. (2011, July). Cost minimization for

- provisioning virtual servers in amazon elastic compute cloud. In *2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems* (pp. 85-95). Singapore: IEEE.
- Chaisiri, S., Lee, B. S., & Niyato, D. (2011). Optimization of resource provisioning cost in cloud computing. *IEEE transactions on services Computing*, *5*(2), 164-177.
- Chan-In, P., & Wongthai, W. (2017). Performance improvement considerations of cloud logging systems. *ICIC Express Letters*, *11*(1), 37-43.
- Chase, J., Kaewpuang, R., Yonggang, W., & Niyato, D. (2014, June). Joint virtual machine and bandwidth allocation in software defined network (SDN) and cloud computing environments. In *2014 IEEE international conference on communications (ICC)* (pp. 2969-2974). Sydney, Australia: IEEE.
- Chen, X., Cheng, L., Liu, C., Liu, Q., Liu, J., Mao, Y., & Murphy, J. (2020). A WOA-based optimization approach for task scheduling in cloud computing systems. *IEEE Systems journal*, *14*(3), 3117-3128.
- Cloud Security Alliance. (2019). Top Threats to Cloud Computing: Egregious Eleven in 2019. The Cloud Security Alliance, Tech.REP.
- Cloud Security Alliance. (2022). Top Threats to Cloud Computing: The Pandemic 11. Retrieved November 1, 2022, from <https://cloudsecurityalliance.org/>
- Csáti, G., Pelle, I., & Toka, L. (2022, April). Optimizing Performance and Resource Consumption of Cloud-Native Logging Application Stacks. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium* (pp. 1-4). Budapest, Hungary: IEEE.
- Cyber Security Insiders. (2022). *2022 Cloud Security Report*. www.isc2.org
- Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. *Journal of Parallel and Distributed Computing*, *142*, 36-45.
- Dogani, J., & Khunjush, F. (2021, October). Cloud service composition using genetic algorithm and particle swarm optimization. In *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)* (pp. 98-104). Mashhad, Iran: IEEE.

- Du, W., Li, A., Li, Q., & Zhou, P. (2021). Privacy-Preserving and Secure Cloud Computing: A Case of Large-Scale Nonlinear Programming. *IEEE Transactions on Cloud Computing*, 11(1), 484-498.
- Fargo, F., Franza, O., Tunc, C., & Hariri, S. (2020, December). VM Introspection-based Allowlisting for IaaS. In *2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)* (pp. 1-4). Milan, Italy : IEEE.
- Gao, Y., Guan, H., Qi, Z., Song, T., Huan, F., & Liu, L. (2014). Service level agreement based energy-efficient resource management in cloud data centers. *Computers & Electrical Engineering*, 40(5), 1621-1633.
- Gartner. (2020). *Managing Privileged Access in Cloud Infrastructure*.
<https://www.gartner.com/en/documents/3986121>
- Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1), 1502242.
- Henry-Stocker, S. (2021). *Checking linux system performance with sar*.
<https://www.networkworld.com/article/3622018/checking-linux-system-performance-with-sar.html>
- Hickman, M., Fulp, D., Baseman, E., Blanchard, S., Greenberg, H., Jones, W., & DeBardleben, N. (2018, October). Enhancing HPC system log analysis by identifying message origin in source code. In *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 100-105). IEEE.
- Hosseini, S. H., Vahidi, J., Kamel Tabbakh, S. R., & Shojaei, A. A. (2021). Resource allocation optimization in cloud computing using the whale optimization algorithm. *International Journal of Nonlinear Analysis and Applications*, 12(Special), 343-360.
- Hu, S., Bai, W., Chen, K., Tian, C., Zhang, Y., & Wu, H. (2018). Providing bandwidth guarantees, work conservation and low latency simultaneously in the cloud. *IEEE Transactions on Cloud Computing*, 9(2), 763-776.
- Ibrahim, H., Aburukba, R. O., & El-Fakih, K. (2018). An integer linear programming model and adaptive genetic algorithm approach to minimize energy consumption of

- cloud computing data centers. *Computers & Electrical Engineering*, 67, 551-565.
- Inokuchi, K., & Kourai, K. (2020). Secure VM management with strong user binding in semi-trusted clouds. *Journal of Cloud Computing*, 9(1), 1-22.
- Jaiboon, W., Wongthai, W., Phoka, T., & Auxsorn, T. (2020). A logging system in openstack environment to mitigate risks associated with threats in infrastructure as a service cloud. *Icic express letter*, 14(4). 387-397.
- Jennings, B., & Stadler, R. (2015). Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, 23, 567-619.
- Kaur, R., & Kaur, G. (2017). Proactive scheduling in cloud computing. *Bulletin of Electrical Engineering and Informatics*, 6(2), 174-180.
- Khan, M. S. A., & Santhosh, R. (2022). Hybrid optimization algorithm for vm migration in cloud computing. *Computers and Electrical Engineering*, 102, 108152.
- Ko, R. K., & Will, M. A. (2014, June). Progger: An efficient, tamper-evident kernel-space logger for cloud data provenance tracking. In *2014 IEEE 7th International Conference on Cloud Computing* (pp. 881-889). Washington, DC: IEEE.
- Ko, R. K., Jagadpramana, P., & Lee, B. S. (2011, November). 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* (pp. 765-771). Changsha, China: IEEE.
- Kumari, P. (2021, June). A Review: Different Challenges in Energy-Efficient Cloud Security. In *IOP Conference Series: Earth and Environmental Science* (Vol. 785, No. 1, p. 012002). IOP Publishing.
- Kumrai, T., Ota, K., Dong, M., Kishigami, J., & Sung, D. K. (2016). Multiobjective optimization in cloud brokering systems for connected Internet of Things. *IEEE Internet of Things Journal*, 4(2), 404-413.
- Li, G., Yan, J., Chen, L., Wu, J., Lin, Q., & Zhang, Y. (2019). Energy consumption optimization with a delay threshold in cloud-fog cooperation computing. *IEEE access*, 7, 159688-159697.
- Lin, W., Liang, C., Wang, J. Z., & Buyya, R. (2014). Bandwidth-aware divisible task

- scheduling for cloud computing. *Software: Practice and Experience*, 44(2), 163-174.
- Lionel Sujay Vailshery. (2022). Annual spending on cloud IT infrastructure worldwide from 2013 to 2026. <https://www.statista.com/statistics/503686/worldwide-cloud-it-infrastructure-market-spending/>
- Lionel Sujay Vailshery. (2022). *Public IT cloud services global market growth 2011-2023*. <https://www.statista.com/topics/2739/cloud-infrastructure-as-a-service/#topicOverview>
- Malekloo, M. H., Kara, N., & El Barachi, M. (2018). An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments. *Sustainable Computing: Informatics and Systems*, 17, 9-24.
- Manvi, S. S., & Shyam, G. K. (2014). Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. *Journal of network and computer applications*, 41, 424-440.
- Mavridis, I., & Karatza, H. (2017). Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark. *Journal of Systems and Software*, 125, 133-151.
- Megahed, A., Nazeem, A., Yin, P., Tata, S., Nezhad, H. R. M., & Nakamura, T. (2019). Optimizing cloud solutioning design. *Future Generation Computer Systems*, 91, 86-95.
- Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
- Meng, X., Pappas, V., & Zhang, L. (2010, March). Improving the scalability of data center networks with traffic-aware virtual machine placement. In *2010 Proceedings IEEE INFOCOM* (pp. 1-9). Chengdu, Chin: IEEE.
- Microsoft. (2022). *High CPU usage troubleshooting guidance*. Retrieved December 10, 2022, from <https://learn.microsoft.com/en-us/troubleshoot/windows-server/performance/troubleshoot-high-cpu-usage-guidance>
- Mishra, D., Singh, V., & Ledalla, A. (2022). Workload Similarity Analysis using Machine Learning Techniques. *arXiv preprint arXiv:2210.13757*.
- Mohammed, C. M., & Zeebaree, S. R. (2021). Sufficient comparison among cloud computing services: IaaS, PaaS, and SaaS: A review. *International Journal of*

Science and Business, 5(2), 17-30.

Mostafavi, S., & Hakami, V. (2020). A stochastic approximation approach for foresighted task scheduling in cloud computing. *Wireless Personal Communications*, 114, 901-925.

Noor, A., Mitra, K., Solaiman, E., Souza, A., Jha, D. N., Demirbaga, U., ... & Ranjan, R. (2019). Cyber-physical application monitoring across multiple clouds. *Computers & Electrical Engineering*, 77, 314-324.

Parikh, S. M., Patel, N. M., & Prajapati, H. B. (2017). Resource management in cloud computing: classification and taxonomy. *arXiv preprint arXiv:1703.00374*.

Patrascu, A., & Patriciu, V. V. (2015). Logging for cloud computing forensic systems. *International Journal of Computers Communications & Control*, 10(2), 222-229.

Payne, B. D., Martim, D. D. A., & Lee, W. (2007, December). Secure and flexible monitoring of virtual machines. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)* (pp. 385-397). Miami Beach, Florida: IEEE.

Pfleeger, C. P. (2009). *Security in computing*. Pearson Education India.

Pfoh, J., Schneider, C., & Eckert, C. (2011, November). Nitro: Hardware-based system call tracing for virtual machines. In *international workshop on security* (pp. 96-112). Berlin: Heidelberg.Springer.

Preethi, J., Sujaudeen, N., Venugopal, P., & Minalinee, T. T. (2015). Cloud service ranking and selection using linear programming. *International Journal of Computer Applications*, 124(3), 39-43..

Raju, P. B. S., & Govindarajulu, P. (2014). Pi-tool to improve performance of application in multi-core architecture. *International Journal of Computer Science and Security (IJCSS)*, 8(4), 84-96.

Rane, S., Wagh, S., & Dixit, A. (2021). Blockchain driven secure and efficient logging for cloud forensics. *International Journal of Computing and Digital System*, 1339-1455.

Rankothge, W., Le, F., Russo, A., & Lobo, J. (2017). Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms. *IEEE*

Transactions on Network and Service Management, 14(2), 343-356.

- Rao, S. S. (2019). *Engineering optimization: theory and practice*. John Wiley & Sons.
- Rashid, A., & Chaturvedi, A. (2019). Cloud computing characteristics and services: a brief review. *International Journal of Computer Sciences and Engineering*, 7(2), 421-426.
- Razaque, A., Vennapusa, N. R., Soni, N., & Janapati, G. S. (2016, April). Task scheduling in cloud computing. In *2016 IEEE long island systems, applications and technology conference (LISAT)* (pp. 1-5). Farmingdale, NY: IEEE.
- Rizvandi, N. B., Lee, Y. C., & Zomaya, A. Y. (2012). On Modeling CPU Utilization of MapReduce Applications. Sydney: CoRR.
- Saha, A. (2006). *Learning about linux process The Linux Gazette*. Retrieved December 10, 2022, from <http://linuxgazette.net/133/saha.html>
- Samara, A. K., & Abandah, G. A. (2021, November). Investigating Fast BiLSTM Neural Networks for Arabic Language Applications. In *2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)* (pp. 250-255). Amman, Jordan: IEEE.
- Shafiq, D. A., Jhanjhi, N. Z., Abdullah, A., & Alzain, M. A. (2021). A load balancing algorithm for the data centres to optimize cloud computing applications. *IEEE Access*, 9, 41731-41744.
- Shakor, M. Y., & Surameery, N. M. S. (2021). Cloud computing technologies adoption in higher education institutes during COVID-19 pandemic: Case study. *Passer Journal of Basic and Applied Sciences*, 3(2), 187-193.
- Shamseddine, M., Al-Dulaimy, A., Itani, W., Nolte, T., & Papadopoulos, A. V. (2022, May). Nodeguard: A Virtualized Introspection Security Approach for the Modern Cloud Data Center. In *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)* (pp. 790-797). Melbourne, Australia: IEEE.
- Sheikh, A. (2020). *Scheduling Security Model for a Cloud Environment* (Doctoral dissertation). Durham: Durham University.
- Singh, U., & Baheti, P. K. (2017). Role and service of cloud computing for higher education system. *International Research Journal of Engineering and*

Technology, 9, 10.

- Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1), 1-11.
- Sultan, N. (2010). Cloud computing for education: A new dawn?. *International Journal of Information Management*, 30(2), 109-116.
- Sundareswaran, S., Squicciarini, A., & Lin, D. (2012). Ensuring distributed accountability for data sharing in the cloud. *IEEE transactions on dependable and secure computing*, 9(4), 556-568.
- Sunyaev, A. (2020). *Internet computing: Principles of Distributed systems and emerging internet-based technologies*. Switzerland: Springer Nature.
- Tutorials Point, (2022). *Cloud Computing – Quick Guide*. Retrieved December 10, 2022, from https://www.tutorialspoint.com/cloud_computing/cloud_computing_quick_guide.htm
- Vasar, M., Srirama, S. N., & Dumas, M. (2012). Framework for monitoring and testing web application scalability on the cloud. In *Proceedings of the WICSA/ECSA 2012 Companion Volume* (pp. 53-60). Helsinki, Finland: IEEE.
- Wei, X. (2020). Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing. *Journal of Ambient Intelligence and Humanized Computing*, 1-12. doi: 10.1007/s12652-020-02614-7
- Wongthai, W. (2014). *Systematic support for accountability in the cloud* (Doctoral dissertation). Newcastle: Newcastle University.
- Wongthai, W., & van Moorsel, A. (2017). Logging system architectures for infrastructure as a service cloud. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(2-4), 35-40.
- Yang, Z., Wang, W., & Huang, Y. (2017, May). Ensuring reliable logging for data accountability in untrusted cloud storage. In *2017 IEEE International Conference on Communications (ICC)* (pp. 1-6). Paris, France: IEEE.
- Ye, K., Jiang, X., Huang, D., Chen, J., & Wang, B. (2011, July). Live migration of multiple virtual machines with resource reservation in cloud computing environments. In *2011 IEEE 4th International Conference on Cloud Computing* (pp. 267-274).

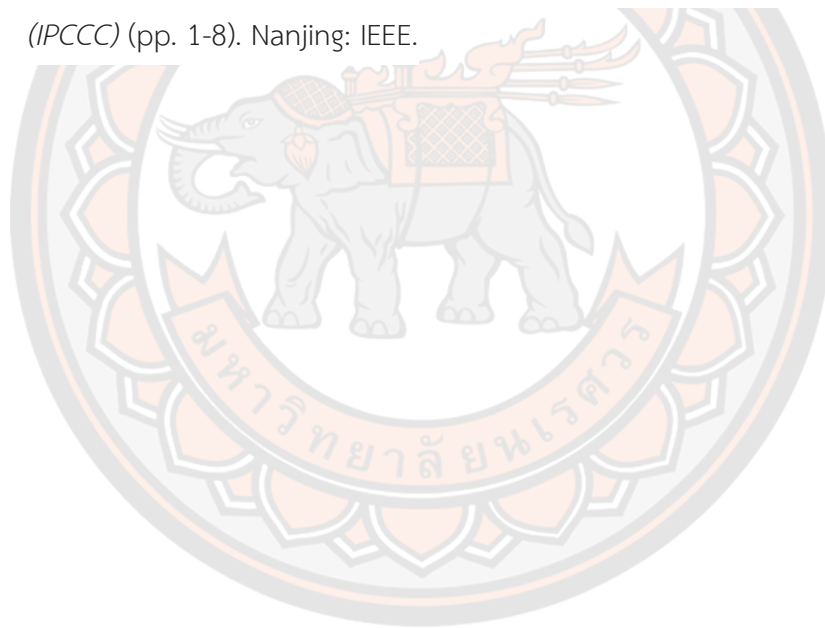
Taipei, Taiwan: EEE.

Yerukhimovich, A. B. (2015). *Secure and resilient cloud computing for the department of defense*. MIT Lincoln Laboratory Lexington United States.

Zabinsky, Z. B. (2009). Random search algorithms. *Department of Industrial and Systems Engineering, University of Washington, USA*.

Zeng, Z., Lu, D., Hu, Y., Augenbroe, G., & Chen, J. (2023). A comprehensive optimization framework for the design of high-performance building systems. *Journal of Building Engineering*, 65, 105709.

Zhao, H., Zheng, Q., Zhang, W., Chen, Y., & Huang, Y. (2015, December). Virtual machine placement based on the VM performance models in cloud. In *2015 IEEE 34th international performance computing and communications conference (IPCCC)* (pp. 1-8). Nanjing: IEEE.





ภาคผนวก

ภาคผนวก ก แบบตอบรับการตีพิมพ์เผยแพร่ผลงานวิจัย



ICIC Express Letters, Part B: Applications
An International Journal of Research and Surveys

May 19, 2023

ACCEPTANCE LETTER (Paper ID: ICICIC2023-SS02-02)

Dr. Winai Wongthai
Naresuan University
Thailand
E-mail: winaiw@nu.ac.th

Dear Dr. Winai Wongthai,

We are happy to inform you that after peer review, your following paper,

Reference No.: ICICIC2023-SS02-02

Title: Enhancing Cloud Security and Performance through VM-Logger Assignment Optimization
Author(s): Surapong Wiriya, Winai Wongthai, Thanathorn Phoka, Nattapon Kunyaito, Kittiphop Mahawan and Pornpimon Borivan

has been formally accepted for publication in ICIC Express Letters, Part B: Applications – An International Journal of Research and Surveys (ISSN 2185-2766).

We are planning tentatively to publish your paper in Volume 15, 2024.

Thank you for submitting your work to ICIC-ELB.



ICIC-ELB Editorial Office
Tokai University, Kumamoto Campus
9-1-1, Toroku, Kumamoto 862-8652, Japan
Tel: +81-96-386-2666
E-mail: office@icicelb.org
URL: <http://www.icicelb.org>

ENHANCING CLOUD SECURITY AND PERFORMANCE THROUGH VM-LOGGER ASSIGNMENT OPTIMIZATION

SURAPONG WIRIYA¹ WINAI WONGTHAI^{1,2,*} THANATHORN PHOKA^{1,2}
NATTAPON KUMYAITO¹ KITTIPHOP MAHAWAN³ AND PORNPIMON BORIWAN⁴

¹Department of Computer Science and Information Technology,

²Research center for Academic Excellence in Nonlinear Analysis and Optimization,
Faculty of Science, Naresuan University, Phitsanulok 65000, Thailand,

³Department of Computer Education, Faculty of Education,

Phranakhon Si Ayutthaya Rajabhat University, Phranakhon Si Ayutthaya 13000, Thailand, and

⁴Department of Mathematics, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

ABSTRACT. *In this work, we address the challenging "VM-logger assignment problem" in the context of large-scale computer systems that require the efficient assignment of loggers to monitor virtual machines (VMs). To tackle this problem, we propose a mathematical model formulated as a constrained optimization problem with the primary goal of maximizing the number of workload-free loggers. The constraints involve each VM's security and performance level requirements, as well as the workload of the associated logger. Our model assigns each VM to a single logger, while a logger can oversee any number of VMs, as long as the constraints are met.*

Initially, we implement a random approach to solve the optimization problem. However, this approach proves to be inadequate in finding an assignment that satisfies the constraints when they are too restrictive. To overcome this challenge, we propose relaxing these restrictions by transforming the constraints of the original optimization problem into penalties on the objective function of a new unconstrained optimization problem.

The solutions derived from the unconstrained optimization problem may not satisfy the original constraints. However, in our VM-logger assignment problem, these solutions have minimal impact on the overall VM-logger system or customer satisfaction. This research article presents a comprehensive overview of the problem, the proposed mathematical model, the relaxation of constraints, and the implications of our approach on the VM-logger assignment system and customer satisfaction.

Keywords: Optimization, Virtual Machine, Logging System, Cloud Computing

1. **Introduction.** Cloud computing has emerged as a transformative technology in the field of information technology (IT), leveraging virtualization technology for efficient data storage and processing [1, 2]. As a type of Infrastructure as a Service (IaaS), cloud computing provides virtual machines (VMs) to customers using services such as Amazon Elastic Compute Cloud [3]. With applications in various sectors, including government, education, and medical trials, the global public cloud service market is projected to grow by approximately 18.8% in 2022, valued at roughly \$490 billion [4]. Consequently, organizations aim to enhance efficiency, security, and service availability while reducing costs.

One of the critical concerns in IaaS cloud computing is security [5]. The Cloud Security Alliance (CSA) has published numerous reports on IaaS cloud security threats, emphasizing the need for robust security measures [6]. Logging systems play a crucial role in mitigating security concerns by monitoring incidents occurring within a customer's IaaS VM, such as unauthorized access or changes to a customer's file in a disk of the VM [7]. As the logging system becomes more active, CPU and RAM usage estimates increase, necessitating efficient resource allocation. To prevent compromising system or application performance, it is advisable to maintain CPU traffic demand at 80% or higher [8].

Optimization techniques have been employed in cloud research to enhance efficiency. Numerous studies have explored various aspects of cloud optimization, such as maximize resource utilization, minimize energy consumption, communication cost and security [9], optimal solutions for cloud resource usage [10], virtual machine allocation to the task using an optimization method [11], the use of optimization strategies to address issues such as unbalanced load or low VM resource utilization [12], and to achieve optimal resource allocation and minimize the total runtime of requested services in cloud computing [13]. It can

be observed that optimization in the cloud primarily focuses on resource allocation, such as CPU, RAM, hard disk, and tasks. However, one area that remains unexplored is the VM-logger assignment problem.

In this work, we aim to bridge this gap by introducing a method for addressing the VM-logger assignment problem to benefit both service providers and cloud service users. By examining the appropriate assignment of VMs to loggers, we can further optimize cloud computing resources, enhance security, and improve overall system performance.

The VM-logger assignment problem involves efficiently allocating loggers to VMs while considering the constraints and requirements of each VM and logger. Our proposed method seeks to maximize the number of workload-free loggers while ensuring the constraints involving each VM's security and performance level requirements, as well as the workload of the associated logger, are met. Addressing this problem effectively will lead to better resource utilization, reduced costs, and improved customer satisfaction.

To tackle the VM-logger assignment problem, we propose a mathematical model formulated as a constrained optimization problem. Our model assigns each VM to a single logger, while a logger can oversee any number of VMs, as long as the constraints are met. The solutions derived from this optimization problem may not satisfy the original constraints, leading to suboptimal results. To overcome this challenge, we propose relaxing these restrictions by transforming the constraints of the original optimization problem into penalties on the objective function of a new unconstrained optimization problem.

By addressing the VM-logger assignment problem, we contribute to the growing body of research on cloud optimization and security. Our findings have implications for cloud service providers, who can use the proposed method to optimize their resource allocation, resulting in enhanced security, improved system performance, and reduced costs. Furthermore, our research will benefit cloud service users, who can expect better performance and security from their VMs in an optimized environment.

Research gaps and contributions: The main contribution of this paper lies in addressing the VM-logger assignment problem. A shortcoming in earlier works (such as [7, 14–17]) is that their logging systems were limited to detecting processes within a single virtual machine, leading to reduced efficiency. And the logging system increase CPU and RAM usage when using multiple logging systems for VMs. To enhance efficiency, a logging system ought to be capable of monitoring processes across multiple virtual machines, which would, in turn, decrease CPU and RAM usage for the provider, allowing the system to operate at full capacity. To tackle the VM-logger assignment problem and achieve optimized resource allocation for both the provider and the customer, an appropriate process for determining the optimal assignment is utilized, ensuring that the computer can work at maximum efficiency. We need to find a solution to maximize the number of workload-free loggers while ensuring the constraints involving each VM's security and performance level requirements. We propose a mathematical model for VM-Logger assignment optimization, along with an example.

2. Background.

2.1. Infrastructure as a Service cloud architecture. Figure 1 illustrates the architectures of both IaaS cloud and logging systems. These architectures are adapted from our previous work [18, 19]. In Figure 1, the white boxes represent the primary elements of the IaaS architecture, namely the hypervisor, dom0, hw0, domU, hwU, disk0, diskU, and memU. Components ending with '0' indicate physical ownership and management by the IaaS provider, while those ending with 'U' denote virtual ownership and management by the cloud customer.

The hypervisor, depicted by box number 2 in Figure 1, is software that enables a physical computer to accommodate multiple VMs. The top-left box, dom0 or domain 0, serves as the manager for all VMs created by customers. During system booting, the hypervisor launches dom0, which is also a VM and has exclusive access and control over hw0 and all customer-created VMs, or domUs. The bottom box in Figure 1, hw0, represents all the physical hardware managed by dom0.

The top-right white box corresponds to domU, or the user domain, which is a customer-created VM that runs on the hypervisor. DomU is an IaaS cloud product provided by the IaaS provider to the customer. HwU, physically located in hw0, represents the virtual hardware of domU. Although it is physically owned and managed by dom0 or the provider, it is virtually owned and managed by the domU owner or the IaaS customer. Disk0 represents the physical disk of dom0, while diskU is the virtual disk of domU. Finally, memU is the virtual main memory of domU.

2.2. Logging system. A logging system operates by utilizing a logger to monitor a domU. In this context, we will refer to a domU as a VM and a dom0 as a host VM. One logger can monitor multiple VMs.

A logging system can log incidents occurring in a customer's IaaS VM, such as who has access to or what transpires with a customer file in a VM's disk [7, 18]. A logging system can consist of a logging

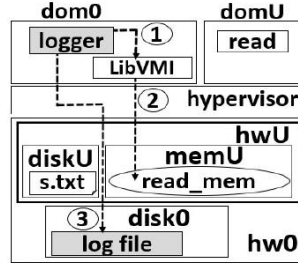


FIGURE 1. Infrastructure as a Service cloud architecture and logging system architecture, adapted from [18,19]

process and a log file [20]. In this paper, we will refer to the logging process as a logger. The system architecture of the logging system is derived from our previous work [18,19] and is also depicted in Figure 1. The box inside the domU in Figure 1 represents the read process. For the purposes of the experiment, we assume that this process could potentially be controlled by an attacker. As a result, the attacker could maliciously read a sensitive file, such as s.txt, belonging to an IaaS customer in diskU, as illustrated by the document shape within the diskU. The read mem in memU represents a reserve memory space for the read process provided by the OS hosting this process. The white box in the dom0 is LibVMI, the new name for XenAccess [21]. It is a C library installed in the dom0 that can access read mem in memU to detect the malicious activity of the read process, which is reading s.txt.

From Figure 1, the three working steps of the logger are represented by the circles numbered 1 to 3. In Step 1, the logger in the dom0 calls LibVMI to access memU to obtain the logging data from read mem (Step 2). This data includes i) the file name of s.txt or the string "s.txt" and ii) the process ID of the read process. Then, LibVMI accesses memU to obtain this data in read mem. Subsequently, it returns the obtained data to the logger. Finally, the logger processes the data and writes (in Step 3) the data into the log file.

2.3. Optimization in cloud computing. The optimization process is a critical aspect of developing a system that operates with maximum efficiency [22]. It involves the utilization of an algorithm to identify the optimal solution to an optimization problem. The development of an optimization model necessitates the establishment of a quantifiable and measurable criterion to assess the effectiveness of a decision. This criterion can aim to either maximize a favorable outcome or minimize costs associated with the decision [23].

To create an appropriate optimization model, it is imperative to establish the three fundamental components that constitute an optimization problem: the objective function, the problem constraints, and the decision variables. The definitions of each of these components are expounded in [24]. In the realm of cloud computing research, optimization techniques have been applied to enhance efficiency, such as resource provisioning cost optimization in [3], optimal resource utilization in cloud computing in [10], and optimization strategies to address unbalanced load, slow convergence speed, and low utilization of VM resources in [12]. The optimization process involves the formulation of a specific optimization problem by incorporating the design variable(s), objective function, and constraints.

2.4. Random search algorithm. A random search algorithm is an optimization method that incorporates randomness or probability, usually through a pseudo-random number generator. This type of algorithm is also known as a Monte Carlo method or stochastic algorithm in academic literature [25]. Various random search algorithms have been developed, including simulated annealing, genetic algorithms, evolutionary programming, particle swarm optimization, ant colony optimization, cross-entropy, stochastic approximation, multi-start, clustering algorithms, and other techniques, which are widely used to solve both continuous and discrete global optimization problems [12,26-30]

The random search algorithm operates in the following steps:

step 0: Initialize algorithm parameters Θ_0 initial points $X_0 \subset S$ and iteration index $k = 0$.

step 1: Generate a collection of candidate points $V_{k+1} \subset S$ according to a specific generator and associated sampling distribution.

step 2: Update X_{k+1} based on the candidate points V_{k+1} , previous iterates and algorithmic parameters. Also update algorithm parameters Θ_{k+1} .

step 3: If a stopping criterion is met, stop. Otherwise increment k and return to Step 1.

TABLE 1. Security Level

Security Level	Description	Security of files in VMs
5	Lowest security	86.47%
4	Low security	99.67%
3	Medium security	99.75%
2	High security	99.82%
1	Highest security	99.86%

TABLE 2. Performance Level

Performance Level	Description	Performance of VMs
5	Highest performance	83.00%
4	High performance	80.00%
3	Medium performance	73.00%
2	Low performance	66.00%
1	Lowest performance	50.00%

2.5. Security and performance. An accuracy refers to the precision of a logging system in recording information from volatile memory in a target monitored VM [31]. The accuracy of process detection by the logger affects the security of files on a customer's virtual machine. Hence, the researcher used this accuracy to define a security level.

Based on [31], we can define five security levels, from Level 1 to Level 5.

Level 1 is the highest level. A customer who chooses Level 1 will have one logger to monitor their VM1. This logger will not monitor any other VMs. This 1-to-1 monitoring method allows the logger to have high resource to capture malicious processes in VM1, resulting in high accuracy and security.

Level 2 is the second-highest level. A customer who chooses Level 2 will have one logger to monitor their VM1, but this logger will also monitor VM2. This 1-to-2 monitoring method requires the logger to share its resource between VM1 and VM2. Consequently, logger2 may not provide the same high accuracy and security as logger1.

Level 3 is the middle level. A customer who chooses Level 3 will have one logger to monitor their VM1, but this logger will also monitor VM2 and VM3. This 1-to-3 monitoring method requires the logger to share its resource between VM1, VM2, and VM3. Thus, logger3 may not provide the same level of accuracy and security as logger1.

Level 4 is the lower level. A customer who chooses Level 4 will have one logger to monitor their VM1, but this logger will also monitor VM2, VM3, and VM4. This 1-to-4 monitoring method requires the logger to share its resource between VM1, VM2, VM3, and VM4. Consequently, logger4 may not provide the same level of accuracy and security as logger1.

Level 5 is the lowest level. A customer who chooses Level 5 will have one logger to monitor their VM1, but this logger will also monitor VM2, VM3, VM4, and VM5. This 1-to-5 monitoring method requires the logger to share its resource between VM1, VM2, VM3, VM4, and VM5. As a result, logger5 may not provide the same level of accuracy and security as logger1.

Table 1 and Table 2, which are summarized from [30], present the tradeoffs between security levels and performance levels that will be considered in this work.

3. Design and Implementation.

3.1. Solutions for allocation of loggers and VMs Design. Suppose a demonstration host that supports 5 VMs. A solution for allocation can be designed. Figure 2 presents the solutions for the allocation of loggers and VMs. The columns represent Logger1 to Logger5, and the rows represent the allocated VMs to each logger. The white box represents a logger, and the number inside the white box represents the number of VMs working with the logger. To simulate the service for users, five scenarios are considered for allocation:

Scenario 1: All five users require security in detecting the logging process, and thus, each person will be working with one logger per VM.

Scenario 2: Three users require security in detecting the logging process and efficient performance of two VMs.

Logger #1	Logger #2	Logger #3	Logger #4	Logger #5	
1	1	1	1	1	Solution 1
1	1	1	2		Solution 2
1	1	3			Solution 3
1	4				Solution 4
5					Solution 5

FIGURE 2. Examples of solutions for assignment of loggers and VMs

L ₁	L ₂	L ₃	L ₄	L ₅
U ₁	U ₂	U ₃	U ₄	U ₅

FIGURE 3. The solutions 1 for allocation of loggers and VMs

Scenario 3: Two users require security in detecting the logging process and efficient performance of three VMs.

Scenario 4: One user requires security in detecting the logging process and efficient performance of four VMs.

Scenario 5: All five users require efficient performance of five VMs.

3.2. Optimization model. This section commences with the development of an optimization model to address the VM-logger assignment problem. The objective of this model is to determine the optimal assignment of loggers to virtual machines while maximizing the number of loggers with no workload. The model takes into account the security and performance requirements of the users to ensure that the assigned loggers meet the desired specifications.

Let X be a decision variable while $X \in I^{m \times n}$ that is,

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix},$$

where $x_{ij} \in \{0, 1\}$, for each $i \in \{1, 2, \dots, m\}$ and $j \in \{1, 2, \dots, n\}$. Notice that the notation x_{ij} indicates the assignment of the i^{th} VM is whether assigned to the j^{th} logger. That is,

$$x_{ij} = \begin{cases} 1, & \text{the } i^{th} \text{ VM is assigned to the } j^{th} \text{ logger} \\ 0, & \text{otherwise} \end{cases},$$

and for each i and j ,

$$U_i = [x_{i1} \quad \cdots \quad x_{in}],$$

$$L_j = [x_{1j} \quad \cdots \quad x_{mj}]^T,$$

and $\sum_{j=1}^n x_{ij}$ for each i must be equal to 1. This means that a VM will be assigned to only one logger. For each j , we define the workload of L_j by

$$N(L_j) = \sum_{i=1}^m x_{ij}$$

Therefore, the optimization problem for the problem of interest will be represented in the following form.

$$\max f(X),$$

subject to $X \in I^{m \times n}$, $p_i \leq h(U_i) = N(L_j) \leq s_i$ for each $i \in \{1, 2, \dots, m\}$, where $h : I^{1 \times n} \rightarrow I$ provides the workload $N(L_j)$ when $x_{ij} = 1$. The objective function $f : I^{m \times n} \rightarrow I$ is referred to as the number of workload-free loggers which will be maximized:

$$f(X) = \sum_{j=1}^n O(N(L_j)),$$

$$\text{where } O(N(L_j)) = \begin{cases} 1, & \text{if } N(L_j) = 0 \\ 0, & \text{otherwise} \end{cases},$$

p_i is the required performance level of the i^{th} VM, and s_i is the required security level of the i^{th} VM.

For instance, assuming that there are five loggers and five VMs, the researcher replaces logger1 with L_1 , logger2 with L_2 , logger3 with L_3 , logger4 with L_4 and logger5 with L_5 , respectively. Similarly, the 1th VM is replaced with U_1 , the 2nd VM with U_2 , the 3rd VM with U_3 , the 4th VM with U_4 and the

5th VM with U_5 . To demonstrate how Solution 1 from Figure 2 can allocate logger 1:1, the researcher provides a representation in Figure 3

The proposed mathematical model can be illustrated as follows.

Step 1: Let

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{5 \times 5}$$

Therefore:

$$\begin{aligned} U_1 &= [1 & 0 & 0 & 0 & 0] \\ U_2 &= [0 & 1 & 0 & 0 & 0] \\ U_3 &= [0 & 0 & 1 & 0 & 0] \\ U_4 &= [0 & 0 & 0 & 1 & 0] \\ U_5 &= [0 & 0 & 0 & 0 & 1] \end{aligned}$$

$$\begin{aligned} L_1 &= [1 & 0 & 0 & 0 & 0]^T \\ L_2 &= [0 & 1 & 0 & 0 & 0]^T \\ L_3 &= [0 & 0 & 1 & 0 & 0]^T \\ L_4 &= [0 & 0 & 0 & 1 & 0]^T \\ L_5 &= [0 & 0 & 0 & 0 & 1]^T \end{aligned}$$

Step 2: Count the number of VMs assigned to each logger.

$$\begin{aligned} N(L_1) &= x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 + 0 + 0 + 0 + 0 = 1 \\ N(L_2) &= x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 0 + 1 + 0 + 0 + 0 = 1 \\ N(L_3) &= x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 0 + 0 + 1 + 0 + 0 = 1 \\ N(L_4) &= x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 0 + 0 + 0 + 1 + 0 = 1 \\ N(L_5) &= x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 0 + 0 + 0 + 0 + 1 = 1 \end{aligned}$$

Step 3: Once the number of VMs assigned to each logger has been determined, set the value to '0' if the logger is in use (i.e., the $N(L_j)$ value is greater than 0), and set the value to '1' if the logger is inactive (i.e., the $N(L_j)$ value is equal to 0). This process is carried out to determine the number of loggers that are not assigned any workload, i.e., the number of workload-free loggers.

Based on Step 2, it can be observed that the summation of workload-free loggers is

$$O(N(L_1)) + O(N(L_2)) + O(N(L_3)) + O(N(L_4)) + O(N(L_5)) = 0 + 0 + 0 + 0 + 0 = 0$$

Step 4: The constraints for creating the optimization model were established based on the security and performance requirements of the virtual machines which is referred to Section 2.5. The allocation conditions for the logging system were designed by taking into account the needs of the service customer. As per the approach in [31], the security and performance levels were classified into five levels, numbered 1 to 5. The researcher aimed to maximize the number of virtual machines assigned to each logger while ensuring higher security levels. Additionally, a higher number of virtual machines assigned to the logger should correspond to a lower security level and a higher performance level.

If it is not possible to find an allocation solution that meets the constraints, the researcher has developed a penalty function model. This model converts a constrained optimization problem into an unconstrained one that can be used to find a solution.

We now consider an unconstrained optimization problem of the original constrained problem. The following notations will be used to define the problem:

$$cost(U_i) = \begin{cases} g_p \times (p_i - h(U_i)), & \text{if } p_i > h(U_i) \\ g_s \times (h(U_i) - s_i), & \text{if } s_i < h(U_i) \\ 0, & \text{otherwise} \end{cases}$$

denoting the negative penalty value of U_i . The weight of the penalty for a security limitation adjustment is denoted as g_s . As the main purpose is to ensure higher security levels, a weight of 2 is assigned to g_s . On the other hand, the weight of the penalty for a performance limitation adjustment is denoted as g_p and a weight of 1 is assigned to it.

Therefore, the unconstrained optimization problem is represented as follows:

$$\max f(X) + \sum_{i=1}^m cost(U_i),$$

subject to $X \in I^{m \times n}$

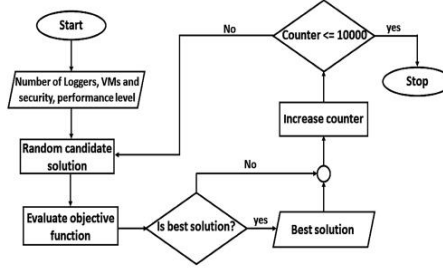


FIGURE 4. Flowchart of optimal logging system model for VMs on cloud computing

3.3. Flowchart of optimal logging system model. A flowchart demonstrating an optimal logging system model for virtual machines (VMs) on cloud computing is shown in Figure 4. The following steps will be explained in detail to describe the process involved in this logging system model.

1. Get input data are Number of Loggers, Number of VMs, Security Level, and Performance Level.
2. Generate random candidate solution for the parameters of the purpose function.
3. Calculate the value of the objective function for the randomized parameters.
4. Check whether the obtained objective function value is the best value. If so, store the value and its associated solution. Increase the number of duty cycles.
5. Check whether the duty cycle conditions have met the number of cycles. For this operation we set the number of cycles to 10000. If correct, terminate the operation and return the parameter value. But if the work does not complete the number of cycles, repeat steps 2-5.

```

1 The number of Loggers: 5
2 The number of VMs: 5
3 The security levels of VMs: [1, 2, 4, 4, 3]
4 The performance Levels of VMs: [1, 2, 3, 3, 3]
5   L1 L2 L3 L4 L5
6 U1 [0, 0, 0, 1, 0]
7 U2 [1, 0, 0, 0, 0]
8 U3 [0, 1, 0, 0, 0]
9 U4 [1, 0, 0, 0, 0]
10 U5 [0, 1, 0, 0, 0]
11
12 The number of VMs in Loggers: ['2' '2' '0' '1' '0']
13 The number of workload-free Logger: 2.0
14 The penalty value: -3.0
15 The optimal value: -1.0
  
```

FIGURE 5. The best solution for assigning VM/VMs to each logger

4. Numerical Example. In Figure 5 indicates the number of Loggers $n = 5$ and the number of VMs $m = 5$. Furthermore, the performance levels of VMs $p_1 = 1, p_2 = 2, p_3 = 3, p_4 = 3,$ and $p_5 = 3$ and the security levels of VMs $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 4,$ and $s_5 = 3$. Based on the input data values, the unconstrained optimization problem proposed results in an optimal solution.

$$X^* = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}_{5 \times 5}$$

Based on the optimal solution of the VM-logger assignment model, the $2^{nd}, 4^{th}$ VMs is assigned to the 1^{st} logger while the $3^{rd}, 5^{th}$ VMs is assigned to the 2^{nd} logger and the 1^{st} VM is assigned to the 4^{th} logger. It follows that there are two remaining workload-free of loggers $f(X^*) = 2$. We note that each i^{th} VM has the corresponding values $s_i \leq h(U_i) = N(L_j) \leq p_i$ as follows:

$$\begin{aligned}
 p_1 &= 1 \leq h(U_1) = N(L_4) = 1 \leq s_1 = 1 \\
 p_2 &= 2 \leq h(U_2) = N(L_1) = 2 \leq s_2 = 2 \\
 p_3 &= 3 \leq h(U_3) = N(L_2) = 2 \leq s_3 = 4
 \end{aligned}$$

$$\begin{aligned} p_4 = 3 &\not\leq h(U_4) = N(L_1) = 2 \leq s_4 = 4 \\ p_5 = 3 &\not\leq h(U_5) = N(L_2) = 2 \leq s_5 = 3. \end{aligned}$$

This means that the costs of U_1 and U_2 equal to 0 whereas the costs of U_3, U_4 and U_5 equal to -1 summed up to $\sum_{i=1}^5 \text{cost}(U_i) = -3$. This results the optimal value to be $f(X^*) + \sum_{i=1}^5 \text{cost}(U_i) = -1$.

The aim of the mathematical model developed in this research is to minimize the usage of loggers, thereby optimizing the CPU and RAM usage of the service provider's computer. However, cloud IaaS providers must be mindful of the trade-offs between security and performance when granting customers the freedom to choose their logging levels. The allocation of the logging system may not suit all customers, and the penalty value may need to be adjusted to ensure optimal allocation for all. Additionally, if a customer's requirements change, a new allocation may be necessary.

5. Conclusion. To develop a mathematical model for optimizing the assignment of VM-loggers, we leveraged experimental data from [31] and employed random search techniques to obtain the most effective model for the unused loggers. In this paper, we utilized an adjustment function model that can accommodate complex requirements and support the allocation of the logging system. This model is also capable of handling calculations for identifying suitable methods that align with changes in the computer environment. The mathematical model developed by the researcher facilitated the determination of an optimal allocation method that utilizes the minimum number of loggers, resulting in cost-effective and efficient utilization of the service provider's CPU and RAM resources.

In the future work, it is possible to incorporate additional hardware-related variables into the model to better accommodate real-world computing environments. For this experiment, we are only working with a single physical machine. Furthermore, if there are multiple physical machines and the VMs created do not specify their physical machine, we can extend this method to allocate the VMs to the appropriate physical machine.

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)*, vol. 17, no. 3, pp. 1399–1408, 2019.
- [2] R. Kaur and G. Kaur, "Proactive scheduling in cloud computing," *Bulletin of Electrical Engineering and Informatics*, vol. 6, no. 2, pp. 174–180, 2017.
- [3] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE transactions on services Computing*, vol. 5, no. 2, pp. 164–177, 2011.
- [4] L. S. Vailshery. (2022) Public it cloud services global market growth 2011- 2023. [Online]. Available: <https://www.statista.com/topics/2739/cloud-infrastructure-as-a-service/topicOverview>
- [5] W. Wongthai and A. van Moorsel, "Quality analysis of logging system components in the cloud," in *Information Science and Applications (ICISA) 2016*. Springer, 2016, pp. 651–662.
- [6] CSA, "The treacherous 12 top threats to cloud computing + industry insights," The Cloud Security Alliance (CSA), Tech. Rep., 2017.
- [7] R. Ko, P. Jagadpramana, and B. S. Lee, "Flogger: A file-centric logger for monitoring file access and transfers within cloud computing environments," in *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, nov. 2011, pp. 765–771.
- [8] Microsoft. (2022) High cpu usage troubleshooting guidance. [Online]. Available: <https://learn.microsoft.com/en-us/troubleshoot/windows-server/performance/troubleshoot-high-cpu-usage-guidance>
- [9] D. Saxena, I. Gupta, J. Kumar, A. K. Singh, and X. Wen, "A secure and multiobjective virtual machine placement framework for cloud data center," *IEEE Systems Journal*, vol. 16, no. 2, pp. 3163–3174, 2021.
- [10] H. K. Ala'a Al-Shaikh, A. Sharieh, and A. Sleit, "Resource utilization in cloud computing as an optimization problem," *Resource*, vol. 7, no. 6, 2016.
- [11] P. S. Rawat, P. Dimri, and G. P. Saroha, "Virtual machine allocation to the task using an optimization method in cloud computing environment," *International Journal of Information Technology*, vol. 12, pp. 485–493, 2020.
- [12] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2020.
- [13] S. H. Hosseini, J. Vahidi, S. R. Kamel Tabbakh, and A. A. Shojaei, "Resource allocation optimization in cloud computing using the whale optimization algorithm," *International Journal of Nonlinear Analysis and Applications*, vol. 12, no. Special Issue, pp. 343–360, 2021.
- [14] 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.

- [15] D.-S. Park, H.-C. Chao, Y.-S. Jeong, and J. J. J. H. Park, Eds., *Framework of Service Accountability and Policy Representation for Trustworthy Cloud Services*. Singapore: Springer Singapore, 2015.
- [16] R. K. L. Ko and M. A. Will, "Progger: An efficient, tamper-evident kernel-space logger for cloud data provenance tracking," in *2014 IEEE 7th International Conference on Cloud Computing*, 2014.
- [17] C. H. Suen, R. K. Ko, Y. S. Tan, P. Jagadpramana, and B. S. Lee, "S2logger: End-to-end data tracking mechanism for cloud data provenance," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2013, pp. 594-602.
- [18] W. Wongthai, "Systematic support for accountability in the cloud," Ph.D. dissertation, Newcastle University, 2014.
- [19] P. Chan-In and W. Wongthai, "Performance improvement considerations of cloud logging systems," *ICIC Express Letters*, 2017.
- [20] W. Wongthai and A. van Moorsel, "Logging system architectures for infrastructure as a service cloud," *Journal of Telecommunication, Electronic and Computer Engineering*, 2017.
- [21] L. Projects. (2019) Functions. [Online]. Available: <http://libvmi.com/api/#Functions>
- [22] Z. Zeng, D. Lu, Y. Hu, G. Augenbroe, and J. Chen, "A comprehensive optimization framework for the design of high-performance building systems," *Journal of Building Engineering*, vol. 65, p. 105709, 2023.
- [23] G. C. Calafiore and L. El Ghaoui, *Optimization models*. Cambridge university press, 2014.
- [24] S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.
- [25] Z. B. Zabinsky *et al.*, "Random search algorithms," *Department of Industrial and Systems Engineering, University of Washington, USA*, 2009.
- [26] M. J. Al-Muhammed and R. A. Zitar, "Probability-directed random search algorithm for unconstrained optimization problem," *Applied Soft Computing*, vol. 71, pp. 165-182, 2018.
- [27] J. Dogani and F. Khunjush, "Cloud service composition using genetic algorithm and particle swarm optimization," in *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*. IEEE, 2021, pp. 98-104.
- [28] S. Mostafavi and V. Hakami, "A stochastic approximation approach for foresighted task scheduling in cloud computing," *Wireless Personal Communications*, vol. 114, pp. 901-925, 2020.
- [29] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1-33, 2015.
- [30] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian, and J. Yang, "Sampling network guided cross-entropy method for unsupervised point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6128-6137.
- [31] T. Auxsorn, W. Wongthai, T. Phoka, and W. Jaiboon, "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*. Springer, 2020, pp. 285-296.