

ระบบฐานความรู้แบบยืดหยุ่นสำหรับการวิเคราะห์ข้อมูลบนคลาวด์คอมพิวติง

Elastic Knowledge-based System for Data Analytics on Cloud Computing Platform

เอกสิทธิ์ กิจสิพงษ์ (Ekasit Kijisipongse)* สุริยะ อุรุเอกโอร (Suriya U-ruekolan)*
และนำฝน อัสวเมชิน (Namfon Assawamekin)**

Received: January 23, 2023

Revised: March 23, 2023

Accepted: April 10, 2023

* ผู้นิพนธ์ประสานงาน: นำฝน อัสวเมชิน (Namfon Assawamekin) อีเมล: namfona@gmail.com

บทคัดย่อ

ระบบฐานความรู้เป็นระเบียบวิธีหนึ่งที่จะช่วยวิเคราะห์ข้อมูลที่มีรูปแบบของความสัมพันธ์ระหว่างข้อมูลที่ซับซ้อนด้วยการสกัดสารสนเทศและความรู้แบบปริยายออกมาจากข้อมูลข้อเท็จจริงที่ชัดเจนผ่านกระบวนการอนุมานด้วยเหตุผลอย่างใดก็ตาม เมื่อข้อมูลที่ต้องนำมาวิเคราะห์มีจำนวนมากและขนาดใหญ่ ทำให้ต้องใช้เวลาในการประมวลผลนาน งานวิจัยนี้เป็นการนำเสนอวิธีการวิเคราะห์ข้อมูลขนาดใหญ่ด้วยการใช้ระบบฐานความรู้บนคลาวด์คอมพิวติงเพื่อวิเคราะห์ข้อมูลจำนวนมากได้ในเวลาอันรวดเร็ว โดยกระจายการประมวลผลของระบบฐานความรู้ไปในทรัพยากรคอมพิวเตอร์ในระบบคลาวด์คอมพิวติงอย่างยืดหยุ่นด้วยการปรับขนาดแบบอัตโนมัติเพื่อเพิ่มหรือลดจำนวนทรัพยากรประมวลผลตามปริมาณงาน ระบบนี้ถูกทดสอบด้วยการเปรียบเทียบเวลาที่ใช้ในการวิเคราะห์ข้อมูลการชำรุดของอุปกรณ์เมื่อใช้การปรับขนาดทรัพยากรแบบอัตโนมัติและเมื่อใช้ทรัพยากรแบบคงที่ ผลการทดสอบแสดงให้เห็นว่า สามารถวิเคราะห์และประมวลผลข้อมูลอุปกรณ์ชำรุดได้อย่างรวดเร็ว โดยใช้เวลาในการประมวลผลเสร็จใกล้เคียงกับการใช้เครื่องเสมือนจำนวน 9 เครื่อง แต่เสียค่าใช้จ่ายใกล้เคียงกับการใช้เครื่องเสมือนจำนวน 2 เครื่อง ทำให้ได้ประสิทธิภาพต่อราคาที่สูงสุด

คำสำคัญ: ระบบฐานความรู้บนคลาวด์คอมพิวติง การประมวลผลแบบกระจาย

Abstract

Knowledge-based system (KBS) is a methodology to carry out an analysis on data having complex interrelationships as it can derive implicit information and knowledge from explicit facts through logical inference process. However, due to the growing number and size of data, the data analysis process spends longer time. This research presents a framework to analyze big data with KBS on cloud computing. Our work aims to shorten the time to analyze big data by distributing KBS processes to multiple cloud computing resources elastically. The system can auto-scale to increase or decrease computing resources to match the workload at each moment. The proposed system is tested by comparing the execution time when using and not using auto-scaling to analyze equipment failure dataset. The results show that the analysis of the dataset is completed faster. The execution time is close to that of 9 virtual machines while the cost is similar to that of 2 virtual machines, achieving the best price-performance ratio.

Keywords: Knowledge-based System, Cloud Computing, Distributed Computing

1. บทนำ

การวิเคราะห์ข้อมูลด้วยการใช้ระบบฐานความรู้ (Knowledge-based System) เป็นระเบียบวิธีหนึ่ง ซึ่งจะช่วยให้สามารถวิเคราะห์ข้อมูลที่มีรูปแบบของความสัมพัทธ์ระหว่างข้อมูลที่ซับซ้อน เพื่อสกัดสารสนเทศ (Information)

* ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.)

* National Electronics and Computer Technology Center, National Science and Technology Development Agency (NSTDA)

** คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยหอการค้าไทย

** School of Science and Technology, University of the Thai Chamber of Commerce

และความรู้แบบปริยาย (Implicit Knowledge) ออกมาจากข้อมูลข้อเท็จจริงที่ชัดเจน (Explicit Facts) ผ่านกระบวนการอนุมานด้วยเหตุผล (Inference) เพื่อจะนำไปสู่การตัดสินใจที่แม่นยำขึ้น สามารถเพิ่มประสิทธิภาพการทำงาน เพิ่มผลกำไรให้แก่ธุรกิจ และลดค่าใช้จ่ายขององค์กร ตัวอย่างเช่น การวิเคราะห์ข้อมูลอุปกรณ์ชำรุด จะช่วยให้หน่วยงานสามารถหาระยะเวลาเฉลี่ยระหว่างการเสียหายแต่ละครั้ง (Mean Time Between Failure) สืบค้นสาเหตุของการชำรุดที่แท้จริง และคำนวณค่าอัตราการชำรุด (Failure Rate) ได้ ช่วยให้หน่วยงานสามารถนำข้อมูลนี้มาปรับปรุงคุณลักษณะเฉพาะของอุปกรณ์ วิธีการจัดซื้อการใช้งาน และซ่อมบำรุงสำหรับอุปกรณ์แต่ละกลุ่มให้ดียิ่งขึ้น

โดยทั่วไปการวิเคราะห์ข้อมูลอุปกรณ์ชำรุดจะต้องมีการรวบรวมข้อมูลต่าง ๆ ที่เกี่ยวกับอุปกรณ์ ตั้งแต่การจัดซื้อการใช้งาน การซ่อมบำรุง ซึ่งข้อมูลเหล่านี้มีความเชื่อมโยงและสัมพันธ์กันในเชิงเวลา (Temporal Relation) แต่ข้อมูลมักจะถูกจัดเก็บแยกกันคนละฐานข้อมูลตามโครงสร้างองค์กรของหน่วยงาน ดังนั้น ข้อมูลเหล่านี้จะต้องถูกรวบรวมมาจากหลายระบบและนำมาวิเคราะห์ร่วมกัน เพื่อให้ได้ข้อมูลอุปกรณ์ชำรุดที่ถูกต้องครบถ้วน แต่จากการศึกษาข้อมูลอุปกรณ์ชำรุดของหน่วยงานหนึ่ง พบว่า ข้อมูลไม่ครบถ้วนตามที่ต้องการ และมีความคลาดเคลื่อน อันเนื่องมาจากการบันทึกข้อมูลของผู้ปฏิบัติงาน ทำให้ต้องใช้ระบบฐานความรู้ในการวิเคราะห์ข้อมูลอุปกรณ์ชำรุดนี้ เพื่อตรวจสอบความสอดคล้องระหว่างชุดข้อมูล และเติมเต็มส่วนที่หายไปเพื่อให้ได้ข้อมูลการชำรุดที่ครบถ้วนที่สุด ภายใต้องค์ความรู้ที่ได้จากผู้เชี่ยวชาญ

นอกจากนี้ข้อมูลที่น่าวิเคราะห์มีจำนวนมากและขนาดใหญ่ ทำให้ต้องใช้เวลาในการประมวลผลนาน ดังนั้น งานวิจัยนี้ได้นำเสนอวิธีการวิเคราะห์ข้อมูลขนาดใหญ่ด้วยการใช้ระบบฐานความรู้บนคลาวด์คอมพิวติง เพื่อวิเคราะห์ข้อมูลจำนวนมากได้ในเวลาอันรวดเร็ว โดยการกระจายการประมวลผลของระบบฐานความรู้ไปยังทรัพยากรคอมพิวเตอร์ในระบบคลาวด์คอมพิวติงอย่างยืดหยุ่นด้วยการปรับขนาดทรัพยากรอัตโนมัติ (Auto-scaling) เพื่อเพิ่มหรือลดจำนวนทรัพยากรประมวลผลให้สอดคล้องตามปริมาณงาน บทความวิจัยชิ้นนี้จะนำเสนอแนวคิดและประโยชน์ที่สำคัญของงานวิจัยใน 2 ด้านได้แก่

1) การออกแบบและพัฒนา QoS-based Elastic Data Processing and Analyzing (QEDPA) Framework เป็นระบบจัดงานและกระจายงาน เพื่อประมวลผลในลักษณะของ High Throughput Batch Computing บนทรัพยากรในระบบคลาวด์พิวติง สามารถเพิ่มลดจำนวนทรัพยากรได้อย่างยืดหยุ่นโดยอัตโนมัติตามปริมาณงานที่ต้องประมวลผล ซึ่งจะช่วยให้ประมวลผลงานได้เร็วขึ้น เนื่องจากสามารถใช้ทรัพยากรจากระบบคลาวด์อย่างเต็มที่ ทรัพยากรการคำนวณมีเวลาสูญเปล่า (Idle Time) ลดลง ซึ่งการเพิ่มลดทรัพยากรแบบอัตโนมัติ จะทำให้มีประสิทธิภาพต่อราคาในการใช้ระบบคลาวด์สูงขึ้น

2) การประยุกต์ใช้ QEDPA Framework ในการวิเคราะห์ข้อมูลขนาดใหญ่บนระบบฐานความรู้ (KBS) ด้วยการกระจายการประมวลผลของ KBS ไปยังทรัพยากรคอมพิวเตอร์ของคลาวด์คอมพิวติงเพื่อวิเคราะห์ข้อมูลการชำรุดของอุปกรณ์

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การเพิ่มลดทรัพยากรคอมพิวเตอร์แบบยืดหยุ่นบนระบบคลาวด์คอมพิวติง

งานวิจัย และแพลตฟอร์มที่เกี่ยวข้องกับการปรับขนาดทรัพยากรคอมพิวเตอร์อัตโนมัติบนระบบคลาวด์คอมพิวติงจากการทบทวนวรรณกรรมพบว่า มีงานวิจัยของ Chen และคณะ [1] และ Bao และคณะ [2] นำเสนองานวิจัยการปรับจำนวนเครื่องคอมพิวเตอร์เสมือน (Virtual Machine) บนแพลตฟอร์ม OpenStack แบบคลาวด์ส่วนตัว (Private Cloud) โดยใช้โมเดลการทำนายภาระงานที่จะเกิดขึ้นมาเป็นตัวตัดสินใจในการเพิ่มลดจำนวนเครื่องเสมือน ซึ่งสามารถรองรับภาระงานที่มีการเพิ่มขึ้นและลดลงอย่างรวดเร็วบนหน่วยประมวลผล และหน่วยความจำได้อย่างมีประสิทธิภาพ งานของ Souza และ Netto [3] เสนอการปรับขนาดทรัพยากรอัตโนมัติบนสถานะแวดล้อมการทำงานแบบคลาวด์คอมพิวติง โดยอาศัยการตัดสินใจจากข้อมูลการทำงานของแอปพลิเคชันกับภาระงานที่เกิดขึ้นบนหน่วยประมวลผล โดยวิธีการที่นำเสนอสามารถทำให้การประมวลผลงานเสร็จได้ตามข้อตกลงในการให้บริการ (SLA) ที่กำหนดไว้ รวมถึงมีประสิทธิภาพต่อราคาที่ดีในการใช้หน่วยประมวลผลบนระบบคลาวด์คอมพิวติง งานวิจัยของ Wang และคณะ [4] ได้เสนอวิธีการปรับปรุงประสิทธิภาพของการปรับขนาดทรัพยากรในแพลตฟอร์ม Kubernetes

¹ หน่วยงานภายใต้การกำกับของรัฐที่มีขนาดใหญ่แห่งหนึ่ง ที่มีจำนวนอุปกรณ์ทั้งขนาดเล็กและใหญ่ทั้งหมดหลายแสนตัว

คลัสเตอร์ให้มีการทำงานที่เร็วขึ้นโดยอาศัยเทคนิคการโหลดอิมเมจไฟล์ไว้ล่วงหน้า (Pre-load Image) บนโหนดสำรองที่มีการแชร์ให้ใช้งานบน Kubernetes คลัสเตอร์โดยโหนดสำรองจะถูกนำมาใช้เมื่อค่าภาระงานของหน่วยประมวลผลและหน่วยความจำของโหนดที่มีการใช้งานอยู่บน Kubernetes คลัสเตอร์เกินค่าที่กำหนด นอกจากนั้น ในส่วนของแพลตฟอร์ม Kubernetes [5] เองได้มีการนำเสนอการปรับขนาดทรัพยากรอัตโนมัติสำหรับคอนเทนเนอร์ (Container) ที่อยู่ภายในโหนดเดียวกันบน Kubernetes คลัสเตอร์โดยอาศัยการตัดสินใจจากภาระงานที่เกิดขึ้นกับหน่วยประมวลผล และหน่วยความจำ รวมถึงการขยายคอนเทนเนอร์ไปยังโหนดอื่น ๆ ที่เพิ่มขึ้นมาภายใน Kubernetes คลัสเตอร์ อย่างไรก็ตาม งานวิจัยและแพลตฟอร์มที่กล่าวมาไม่ได้รองรับการเพิ่มลดทรัพยากรคอมพิวเตอร์บนระบบคลาวด์คอมพิวติงในลักษณะที่ทำงานร่วมกับระบบจัดการงานแบบ High Throughput Batch Computing แบบอัตโนมัติ ตามที่ได้นำเสนอไว้ใน QEDPA Framework นี้

สำหรับบริการคลาวด์อย่าง Amazon ParallelCluster [6] รองรับการเพิ่มและลดทรัพยากรคอมพิวเตอร์แบบอัตโนมัติตามปริมาณงานที่รออยู่ในคิวของระบบจัดการงานเช่นเดียวกับของ QEDPA Framework แต่การใช้งาน Amazon Parallel Cluster จะถูกจำกัดอยู่บนสถานะแวดล้อมการทำงานของ Amazon EC2 ซึ่งเป็นระบบคลาวด์แบบสาธารณะเท่านั้น ทั้งนี้ QEDPA Framework สามารถใช้งานได้ทั้งระบบคลาวด์แบบส่วนตัวและแบบสาธารณะอื่น ๆ ที่มี API สำหรับบริหารจัดการเครื่องคอมพิวเตอร์เสมือนบนระบบคลาวด์

แพลตฟอร์มสำหรับการวิเคราะห์ข้อมูลขนาดใหญ่อย่าง Hadoop/ Spark [7] เป็นแพลตฟอร์มที่สามารถกระจายงานไปประมวลผลได้ทั้งระบบคลัสเตอร์คอมพิวเตอร์ และระบบคลาวด์คอมพิวติง รองรับการทำงานเขียนโปรแกรมวิเคราะห์ข้อมูลในรูปแบบของ MapReduce ที่มีการกระจายการประมวลผลไปยังคอมพิวเตอร์หลาย ๆ เครื่อง แต่ไม่ได้รองรับงานในลักษณะ High Throughput Batch Computing ที่มีการใช้งานอยู่ทั่วไป ซึ่งแต่ละงานเป็นการประมวลผลข้อมูลโดยใช้โปรแกรมที่หลากหลายตั้งแต่ Python, R หรืออื่น ๆ และโดยมากไม่ได้เขียนด้วยรูปแบบ MapReduce

2.2 ระบบฐานความรู้

ระบบฐานความรู้ (KBS) คือโปรแกรมคอมพิวเตอร์ที่เลียนแบบกระบวนการทำงานของมนุษย์ในการแก้ไขปัญหาด้วยการใช้

ความรู้ [8] โดยหนึ่งในรูปแบบของ KBS ที่ใช้กันมาตั้งแต่ยุคแรก ๆ จนถึงปัจจุบัน คือ ระบบผู้เชี่ยวชาญ (Expert System) [9] ซึ่งเป็นการนำความรู้ที่ได้มาจากผู้เชี่ยวชาญมาวิเคราะห์และตัดสินใจเชิงตรรกะ KBS ประกอบด้วย ส่วนสำคัญ 3 ส่วน ได้แก่ 1) ฐานข้อมูลข้อเท็จจริง (Fact) เป็นข้อมูลของวัตถุหรือเหตุการณ์ที่เกิดขึ้นอย่างชัดเจน (Explicit) 2) ฐานความรู้ ได้แก่ ระเบียบปฏิบัติ (Regulation) แนวทางปฏิบัติ (Practice) และประสบการณ์ (Experience) ของผู้เชี่ยวชาญ 3) กลไกการอนุมาน (Inference Engine) เป็นกระบวนการใช้เหตุผลซึ่งใช้ได้ทั้งแบบ Forward Chaining และ Backward Chaining และ 4) ข้อสรุป ซึ่งเป็นผลลัพธ์จากการทำการอนุมานจัดเป็นข้อมูลที่ซ่อนอยู่ (Implicit) โดยค้นหาได้จากการใช้ความรู้ของผู้เชี่ยวชาญในฐานความรู้มาวิเคราะห์เชิงตรรกะร่วมกับข้อมูลข้อเท็จจริงที่เกิดขึ้นที่บันทึกไว้ กระบวนการของ KBS แตกต่างไปจากการใช้วิธีอื่น เช่น การเรียนรู้แบบมีผู้สอน (Supervised Machine Learning) ซึ่งต้องอาศัยข้อมูลตัวอย่างเป็นจำนวนมาก เพื่อให้คอมพิวเตอร์เรียนรู้กฎเกณฑ์การวิเคราะห์จากข้อมูลตัวอย่างเหล่านี้ด้วยตนเอง

KBS ถูกนำมาใช้ในงานต่าง ๆ ได้แก่ การแนะนำอุปกรณ์เพื่อใช้ในการก่อสร้าง [10] การค้นหาคุณสมบัติใหม่ของสสารโดยจากอนุมานจากความรู้เดิม [11] และการวินิจฉัยสาเหตุการชำรุดจากลักษณะการชำรุดของหม้อแปลงไฟฟ้า [12] เป็นต้น

3. วิธีดำเนินการวิจัย

วิธีดำเนินการวิจัยเป็นไปตามขั้นตอนที่แสดงในภาพที่ 1 ประกอบด้วย

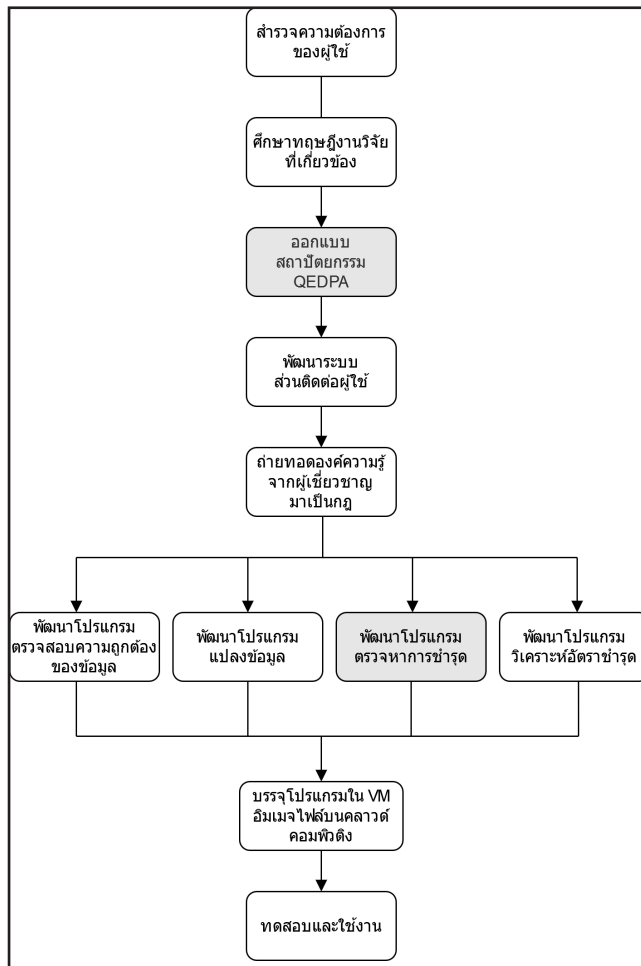
- (1) สำรวจความต้องการของผู้ใช้ระบุปัญหาที่ต้องการแก้ไขในที่นี้ คือ ความต้องการในการวิเคราะห์ข้อมูลขนาดใหญ่ในเวลาอันรวดเร็ว
- (2) ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง เพื่อหาเครื่องมือที่จะนำมาใช้ในการแก้ไขปัญหา
- (3) ออกแบบสถาปัตยกรรม QEDPA Framework ที่จะรองรับการประมวลผลแบบยืดหยุ่นบนคลาวด์คอมพิวติง
- (4) พัฒนาระบบ และส่วนติดต่อผู้ใช้
- (5) ถ่ายทอดองค์ความรู้จากผู้เชี่ยวชาญมาเป็นกฎด้วยการสัมภาษณ์ การศึกษาระเบียบ สังเกตวิธีปฏิบัติงาน และการสัมภาษณ์ระดมสมอง เพื่อให้ผู้เชี่ยวชาญทบทวนความถูกต้องของกฎที่ได้มา

(6) พัฒนาโปรแกรมสำหรับการประมวลผลข้อมูล ได้แก่ การตรวจสอบความถูกต้องของข้อมูล (Data Validation) การแปลงรูปแบบข้อมูล (Data Transformation) การตรวจหาการชำรุด (Failure Detection) และการวิเคราะห์อัตราชำรุด (Failure Rate Analysis) โดยโปรแกรมการตรวจหาการชำรุดเป็นส่วนที่มีการใช้ KBS ในการวิเคราะห์ข้อมูล

(7) นำโปรแกรมที่พัฒนามาบรรจุในอิมเมจไฟล์แล้วส่งอิมเมจไฟล์ไปเก็บบนคลาวด์คอมพิวติง

(8) ทดสอบและทดลองใช้งาน

สำหรับบทความนี้จะอธิบายถึงส่วนที่สำคัญของงานวิจัยนี้ คือ การออกแบบสถาปัตยกรรมของ QEDPA Framework และการพัฒนาโปรแกรมตรวจหาการชำรุดที่ใช้ KBS ซึ่งจะได้อธิบายในหัวข้อที่ 3.1 และ 3.2 ตามลำดับ



ภาพที่ 1 ขั้นตอนการดำเนินการวิจัย

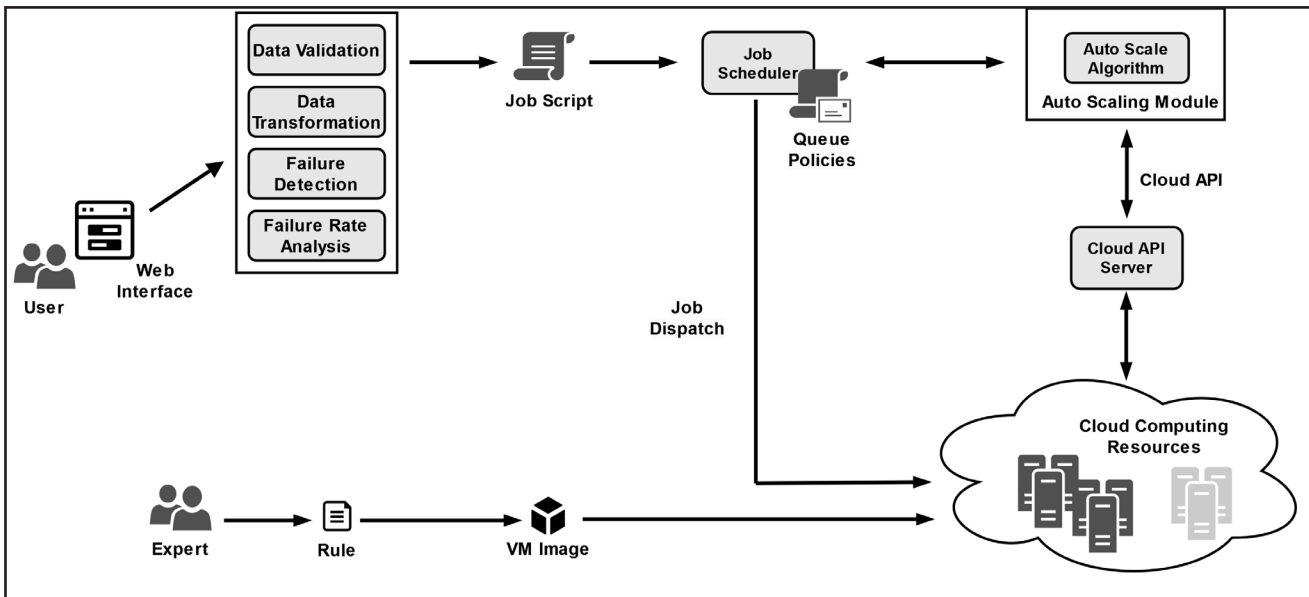
3.1 การออกแบบสถาปัตยกรรมของ QoS-based Elastic Data Processing and Analyzing (QEDPA) Framework

QEDPA เป็น Framework สำหรับการจัดและกระจายงานไปประมวลผลในคลาวด์คอมพิวติงในลักษณะของ High Throughput Batch Computing รองรับปริมาณงานจำนวนมากที่มีปริมาณเปลี่ยนแปลงได้ตลอดเวลา ด้วยการเพิ่มลดเครื่องเสมือนแบบยืดหยุ่นบนคลาวด์คอมพิวติง QEDPA Framework ประกอบด้วยส่วนต่าง ๆ ตามสถาปัตยกรรมในภาพที่ 2

1) คลาวด์คอมพิวติงแบบให้บริการโครงสร้างพื้นฐาน (IaaS) เพื่อให้บริการทรัพยากรคอมพิวเตอร์แบบเสมือน (VM) ที่จะใช้สำหรับประมวลผลข้อมูลภายใต้ QEDPA Framework โดยเครื่องคอมพิวเตอร์เสมือนจะถูกสร้างขึ้นตามปริมาณงานภายใต้การควบคุมของ Auto-scaling โมดูล

2) ส่วนเชื่อมต่อผู้ใช้แบบเว็บ (Web Interface) สำหรับให้ผู้ใช้ส่งงาน โดยผู้ใช้จะต้องระบุชนิดของงาน และพารามิเตอร์ที่ใช้ในประมวลผล พร้อมความต้องการทรัพยากร (ถ้ามี) เช่น จำนวนหน่วยประมวลผลหรือขนาดหน่วยความจำ เป็นต้น สำหรับในงานวิจัยนี้เป็นการประยุกต์นำไปใช้วิเคราะห์ข้อมูลอุปกรณ์ชำรุด จึงประกอบด้วยงานที่ใช้ในขั้นตอนต่าง ๆ ของการวิเคราะห์ ได้แก่ งานตรวจสอบความถูกต้องของข้อมูล งานแปลงรูปแบบข้อมูล งานตรวจหาการชำรุด และงานวิเคราะห์อัตราชำรุด ซึ่งแต่ละงานจะถูกสร้างและส่งเข้าไปในรอในคิวงานในรูปแบบของสคริปต์งาน (Job Script)

3) สคริปต์งานเป็นไฟล์ที่กำหนดชนิดงาน (โปรแกรมและพารามิเตอร์) ที่ต้องใช้ในการประมวลผลใน QEDPA Framework งานที่กำหนดในสคริปต์งานจะต้องทำงานในลักษณะประมวลผลแบบกลุ่ม (Batch Processing) โดยหนึ่งระบบจะรองรับงานหลายชนิด และผู้ใช้งานได้หลายคน และแต่ละคนสามารถส่งงานได้หลายงาน ในสคริปต์งานนี้ ประกอบด้วยความต้องการของทรัพยากรที่ใช้ในการประมวลผล และโปรแกรมสำหรับประมวลผล พร้อมพารามิเตอร์ตามที่ระบุผ่านส่วนเชื่อมต่อผู้ใช้แบบเว็บ ซึ่งงานแต่ละงานจะมีรายละเอียดของสคริปต์งานต่างกัน ดังแสดงในภาพที่ 3 บรรทัดที่ 1-4 เป็นการกำหนดสภาพแวดล้อม (Environment) ของการรันงาน ได้แก่ ชื่อคิวงาน VIR ชื่องาน และชื่อ Log ไฟล์สำหรับ stdout และ stderr บรรทัดที่ 6 เป็นการระบุคำสั่งสำหรับรันโปรแกรมวิเคราะห์ข้อมูลอุปกรณ์ชำรุด โดยใช้สคริปต์ภาษา R พร้อมพารามิเตอร์ที่รับจากผู้ใช้งานผ่านทางส่วนเชื่อมต่อผู้ใช้แบบเว็บ เช่น ชนิดของอุปกรณ์ที่จะวิเคราะห์ หรือเอาต์พุตใดเรกทอรี เป็นต้น



ภาพที่ 2 สถาปัตยกรรมของระบบ QoS-based Elastic Data Processing and Analyzing (QEDPA)

```

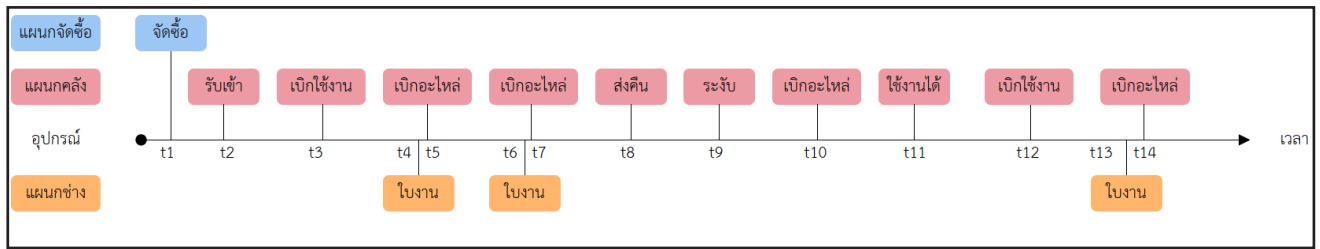
1: #SBATCH --partition=VIR
2: #SBATCH --job-name=failure_detection
3: #SBATCH --output=%j.out
4: #SBATCH --er=%j.err
5:
6: Rscript ~/apps/failure_detection.r $PARAM_1 $PARAM_2
    
```

ภาพที่ 3 ตัวอย่างสคริปต์งานสำหรับงานวิเคราะห์ข้อมูลอุปกรณ์ชำรุด

```

BEGIN:
  Initialize pending_jobs=[], available_instances=[],
    minimum_instance, threshold_instance_idle_time,
    threshold_job_waiting_time
  WHILE true:
    let pending_jobs = Array of pending jobs
    let available_instances = Array of available instances
    IF pending_jobs is not empty &&
      There exists job in pending_jobs whose waiting time >= threshold_job_waiting_time
    THEN
      let pending_too_long_jobs = [for job in pending_jobs if waiting time >=
        threshold_job_waiting_time]
      let number_of_new_instances = length(pending_too_long_jobs)
      create_new_instances(number_of_new_instances)
    ELSE IF length(available_instances) > minimum_instance
      && there exists instance in available_instances whose idle time >=
        threshold_instance_idle_time
      THEN
        let idle_too_long_instances = [for instance in available_instances if idle time >=
          threshold_instance_idle_time]
        destroy_instances(idle_too_long_instances)
      ENDIF
    ENDIF
  ENDWHILE
END
    
```

ภาพที่ 4 อัลกอริทึม Auto Scale สำหรับการเพิ่มลดทรัพยากรบนระบบคลาวด์แบบอัตโนมัติ



ภาพที่ 5 ตัวอย่างเส้นเวลาของเหตุการณ์ที่เกิดขึ้นกับอุปกรณ์

4) ระบบจัดการงาน (Job Scheduler) สำหรับการประมวลผลแบบกลุ่ม ทำหน้าที่บริหารจัดการงานประมวลผลข้อมูลในขั้นตอนต่าง ๆ ตั้งแต่การตรวจสอบความถูกต้อง การแปลงรูปแบบ การตรวจหาอุปกรณ์ชำรุด และการวิเคราะห์อุปกรณ์ชำรุด ซึ่งแต่ละงานอาจใช้ระยะเวลาในประมวลผลข้อมูลตั้งแต่หลักนาทีจนถึงระดับหลายชั่วโมง ขึ้นอยู่กับขนาดของข้อมูล และความซับซ้อนซับซ้อนของกฎหรือวิธีการที่นำมาใช้ในการวิเคราะห์ข้อมูล ทางผู้วิจัยจึงได้นำระบบจัดการงาน SLURM [13] ที่มีการใช้งานอย่างแพร่หลายบนระบบคอมพิวเตอร์คลัสเตอร์เข้ามาช่วยในการจัดการงานดังกล่าวในกลุ่มของเครื่องเสมือน (VM Cluster) บนคลาวด์คอมพิวติง เพื่อช่วยในการจัดลำดับการกระจายงาน และตรวจสอบสถานะการประมวลผลของงานที่ถูกกระจายไปยังเครื่องเสมือนอย่างถูกต้อง ครบถ้วน และมีประสิทธิภาพ

5) Auto Scaling โมดูล ทำหน้าที่เพิ่มลดทรัพยากรการคำนวณด้วยการสร้าง และลบเครื่องเสมือนจากคลาวด์คอมพิวติงแบบอัตโนมัติให้สอดคล้องและเพียงพอต่อปริมาณงานที่มี สำหรับการตัดสินใจเพิ่มลดจำนวนทรัพยากรของ Auto Scaling โมดูลนั้น ใช้อัลกอริทึม Auto Scale ดังแสดงในภาพที่ 4 ซึ่งใช้เวลาที่งานรอคอยในคิว และเวลาสูญเสียของเครื่องเสมือนเป็นเงื่อนไขสำคัญในการตัดสินใจเพิ่ม และลดจำนวนทรัพยากรตามลำดับ โดย Auto Scaling ทำหน้าที่ติดต่อกับเครื่องแม่ข่ายคลาวด์ API (Cloud API Server) เช่น OpenStack [14] ผ่านคลาวด์ API

6) อิมเมจไฟล์สำหรับให้คลาวด์คอมพิวติงใช้ในการสร้างเครื่องเสมือน อิมเมจไฟล์ ประกอบด้วย ระบบปฏิบัติการ ไลบรารี และโปรแกรมต่าง ๆ ที่จำเป็นต่อการวิเคราะห์ข้อมูล ซึ่งจะต้องทำการจัดเตรียมไว้ล่วงหน้าก่อนการใช้งาน QEDPA Framework

7) ผู้ใช้ คือ บุคคลากรที่ทำหน้าที่ในการรวบรวมข้อมูล และออกรายงานเชิงสถิติของอุปกรณ์ชำรุด ส่วนผู้เชี่ยวชาญ

ได้แก่ วิศวกร ช่าง และเจ้าหน้าที่คลังพัสดุ ที่ให้องค์ความรู้เกี่ยวกับวิธีการและกระบวนการปฏิบัติงานกับอุปกรณ์

3.2 การพัฒนาโปรแกรมตรวจหาการชำรุด: ระบบ KBS สำหรับการวิเคราะห์ตรวจหาอุปกรณ์ชำรุด

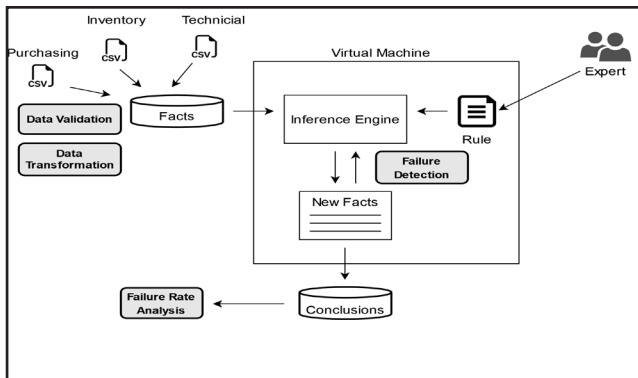
ในหัวข้อนี้จะอธิบายถึงการประยุกต์ใช้ QEDPA Framework เพื่อตรวจหาอุปกรณ์ชำรุดจากข้อมูลขนาดใหญ่ของหน่วยงาน ซึ่งเป็นข้อมูลที่เกี่ยวข้องกับอุปกรณ์ ตั้งแต่การจัดซื้อ การใช้งาน การซ่อมบำรุง ซึ่งมีความเชื่อมโยงและสัมพันธ์กับในเชิงเวลา แต่ถูกจัดเก็บแยกกัน คนละฐานข้อมูลตามโครงสร้างองค์กรของหน่วยงาน ข้อมูลที่สำคัญในการวิเคราะห์ ได้แก่

- ข้อมูลการจัดซื้อ (แผนการจัดซื้อ) ประกอบด้วย คุณลักษณะเฉพาะ หมายเลขอุปกรณ์ ยี่ห้อ รุ่น
- ข้อมูลการรับเข้าคลังและเบิกเพื่อใช้งาน (แผนคลังพัสดุ) ประกอบด้วย วันที่รับเข้า วันที่คืน สถานะ ผู้ใช้งานอุปกรณ์
- ข้อมูลการเบิกจ่ายอะไหล่เพื่อการซ่อมบำรุง (แผนคลังพัสดุ) ประกอบด้วย วันที่เบิก จำนวน ผู้เบิกอะไหล่
- ข้อมูลการออกไปงานแก่ช่างที่ทำหน้าที่ซ่อมบำรุง (แผนช่าง) ประกอบด้วย วันที่ออก วันที่ปฏิบัติงาน วันที่เสร็จช่าง รายละเอียดการซ่อม หมายเลขอุปกรณ์ที่เสีย

ภาพที่ 5 แสดงเส้นเวลา (Timeline) ของเหตุการณ์ที่เกิดขึ้นกับอุปกรณ์ตัวอย่างหนึ่ง ตั้งแต่การจัดซื้อ การใช้งาน และการซ่อมบำรุง ซึ่งจะบันทึกในฐานข้อมูลของแต่ละแผนก โดยข้อมูลของอุปกรณ์จะถูกบันทึกเข้าสู่ฐานข้อมูลของแผนกจัดซื้อเมื่อเวลา t1 หลังจากที่มีการจัดซื้อสำเร็จ อุปกรณ์จะถูกส่งไปเก็บรักษาในแผนกคลังซึ่งจะมีการบันทึกข้อมูลเข้าสู่ฐานข้อมูลของแผนกคลังในเวลา t2 เมื่ออุปกรณ์ถูกเบิกออกไปใช้งานจะมีการบันทึกข้อมูลในฐานข้อมูลของแผนกคลังในเวลา t3 ต่อมาหลังจากที่อุปกรณ์มีการชำรุด จะมีการออกไปงานแจ้งช่างออกไปทำการซ่อมแซม ซึ่งจะมีการบันทึกข้อมูลในฐานข้อมูลของแผนกช่างตามด้วยการเบิกอะไหล่เพื่อนำไปซ่อมแซมในเวลา t4 และ t5 ตามลำดับ

ในกรณีที่อุปกรณ์เสียหายมากจำเป็นต้องส่งกลับมาซ่อมที่คลัง ก็จะมีการบันทึกข้อมูลในฐานข้อมูลของแผนกคลังเพื่อส่งคืน และระบบการใช้งานในเวลา t8 และ t9 ตามลำดับ เพื่อรอการซ่อมแซมและเบิกใช้งานต่อไป

ทั้งนี้ข้อมูลของวันที่และสาเหตุที่อุปกรณ์ชำรุดไม่ได้มีการบันทึกอย่างชัดเจน เนื่องจากหน่วยงานไม่ได้มีกระบวนการบันทึกข้อมูลนี้ อย่างเคร่งครัดและเหมาะสม จึงทำให้ข้อมูลอุปกรณ์ชำรุด ไม่ครบถ้วนสมบูรณ์ตามที่ต้องการ และยังมีปัญหาของความคลาดเคลื่อน อันเนื่องมาจากการบันทึกของผู้ปฏิบัติงานอีกด้วย เช่น ในรายละเอียดการซ่อมที่อยู่ในงาน อาจคลาดเคลื่อนกับข้อเท็จจริงที่เกิดขึ้น ส่งผลให้การวิเคราะห์ปัญหาผิดพลาด จึงทำให้ต้องใช้ KBS ในการวิเคราะห์ข้อมูลอุปกรณ์ชำรุดนี้ เพื่อตรวจสอบความสอดคล้องระหว่างชุดข้อมูล และเติมเต็มส่วนที่หายไปให้ข้อมูลการชำรุดที่ครบถ้วนที่สุด



ภาพที่ 6 การทำงานของโปรแกรมตรวจหาการชำรุด

โปรแกรมตรวจหาการชำรุดเป็นระบบ KBS เพื่อวิเคราะห์ข้อมูลอุปกรณ์ว่าชำรุดหรือไม่ ชำรุดเมื่อใด และชำรุดด้วยสาเหตุใด ด้วยการนำข้อมูลที่เกี่ยวข้องมาวิเคราะห์ร่วมกันภายใต้ความรู้จากผู้เชี่ยวชาญ ดังแสดงในภาพที่ 6 โดยที่

1. แหล่งข้อมูลภายนอก ประกอบด้วย ฐานข้อมูลจากแผนกจัดซื้อ แผนกคลัง และแผนกช่าง ที่บันทึกโดยเจ้าหน้าที่ในแต่ละแผนก โดยผู้ใช้งานจะต้องทำการนำออก (Export) ข้อมูลออกมาเป็นไฟล์ จากนั้นจะต้องนำไฟล์ข้อมูลเหล่านี้ส่งเข้ามาใน KBS ผ่านทางส่วนเชื่อมต่อผู้ใช้แบบเว็บของระบบ
2. ไฟล์ข้อมูลที่ส่งเข้ามาจะถูกตรวจสอบความถูกต้อง และถูกแปลงให้อยู่ในรูปแบบที่พร้อมต่อการนำเข้า (Import) สู่ฐานข้อมูลข้อเท็จจริง ซึ่งเป็นระบบฐานข้อมูลที่มีโครงสร้างดังแสดงในตารางที่ 1

3. กฎในการวิเคราะห์ข้อมูล (Rule) ที่ได้มาจากความรู้ของผู้เชี่ยวชาญ ซึ่งจะต้องครอบคลุมกับเงื่อนไขการชำรุดในรูปแบบต่าง ๆ ที่เป็นไปได้ให้มากที่สุด ในงานวิจัยนี้ นอกจากการนำกฎที่ได้ไปทบทวนกับผู้เชี่ยวชาญแล้ว ยังมีการนำไปสุ่มทดสอบกับข้อมูลบางส่วนที่เพื่อยืนยันความถูกต้องของกฎ

4. กลไกการอนุมาน (Inference Engine) เป็นกระบวนการประมวลผลเชิงตรรกะระบบกฎ และข้อเท็จจริงที่มีอยู่ เพื่อยืนยันให้เกิดความรู้ หรือข้อเท็จจริงใหม่ที่เกี่ยวข้องกับการชำรุดของอุปกรณ์ ซึ่งผู้วิจัยได้เลือกใช้ซอฟต์แวร์ Drools [15] เป็นเครื่องมือในการอนุมานแบบ Forward Chaining เนื่องจาก Drools เป็นซอฟต์แวร์แบบโอเพนซอร์สที่เหมาะสมกับการใช้งานอันหลากหลาย และยังมีความเร็วในการประมวลผลที่ดี [16] โดยทั้งซอฟต์แวร์ Drools และกฎจะถูกบรรจุอยู่ในอิมเมจไฟล์ที่ใช้สำหรับสร้างเครื่องเสมือน

5. ข้อสรุป (Conclusions) คือ ผลลัพธ์สุดท้ายของการอนุมาน คือ ข้อสรุปที่ระบุว่าอุปกรณ์นั้นชำรุดหรือไม่ ชำรุดเมื่อใด และชำรุดด้วยสาเหตุใด ซึ่งจะถูกบันทึกกลับลงในระบบฐานข้อมูลที่มีโครงสร้างดังแสดงในตารางที่ 2

6. ผู้ใช้สามารถวิเคราะห์ค่าอัตราชำรุดจากข้อสรุป โดยการส่งงานผ่านทางส่วนเชื่อมต่อผู้ใช้แบบเว็บของระบบ กฎที่ใช้ในการวิเคราะห์หาอุปกรณ์ชำรุดสามารถแสดงอยู่ในรูปแบบของ

IF <condition> THEN <consequence>

ซึ่งได้มาจากการแปลงความรู้จากผู้เชี่ยวชาญ และจากกระบวนการทำงานของหน่วยงานมาเป็นกฎ โดยกฎที่ใช้จะต้องคำนึงถึงมิติของเวลาเพื่อตรวจสอบความสัมพันธ์ของลำดับเหตุการณ์ก่อนหลังให้ถูกต้องด้วย ตัวอย่างของกฎที่สำคัญ ได้แก่

1) กฎการระบุอุปกรณ์ชำรุด หากมีการออกไปงานให้ช่างออกไปซ่อมบำรุงที่จุดเกิดเหตุ ในกรณีนี้ เมื่ออุปกรณ์ถูกใช้งานในพื้นที่และเกิดการชำรุด อุปกรณ์จะได้รับการซ่อมแซมที่จุดเกิดเหตุ กฎการระบุอุปกรณ์ชำรุดที่สอดคล้องกับกรณีนี้ได้แสดงเป็นตัวอย่างในภาพที่ 7 ในบรรทัดที่ 2-3 เป็นการค้นหารูปแบบในฐานข้อมูลข้อเท็จจริงว่า มีโรงงานให้ช่างออกไปซ่อมอุปกรณ์ และอุปกรณ์นั้น มีสถานะถูกเบิกออกไปใช้งาน บรรทัดที่ 4 เป็นการกำจัดการเบิกไปใช้งานครั้งล่าสุดก่อนการซ่อมเท่านั้น บรรทัดที่ 6 จะสร้างข้อสรุปเบื้องต้นว่าพบอุปกรณ์ชำรุดเมื่อวันที่ใด

วันที่เริ่มใช้งานล่าสุด และวันที่ซ่อมเสร็จ โดยสาเหตุจะยังเป็น ไม่ทราบอยู่ชั่วคราว โดยจะมีกฎอื่นทำการอนุมานหาสาเหตุ จากข้อมูลการเบิกอะไหล่มาปรับให้ในภายหลัง

2) กฎการระบุอุปกรณ์ชำรุด หากมีอุปกรณ์ถูกส่งคืน กลับมาในคลังเพื่อการซ่อมแซมใหญ่ ภาพที่ 8 เป็นตัวอย่างกฎ ที่ใช้ระบุอุปกรณ์ชำรุดในกรณีนี้ หากพบว่าอุปกรณ์ถูกเคลื่อนย้าย กลับเข้ามาเก็บในคลัง และถูกระงับการใช้งาน บรรทัดที่ 2-6 เป็นเงื่อนไขของการค้นหารูปแบบ การเบิกจ่าย -> ส่งคืน -> ระงับการใช้งาน ที่เกิดขึ้นในฐานข้อมูลข้อเท็จจริงสำหรับ อุปกรณ์ใด ๆ บรรทัดที่ 7 เป็นการป้องกันการนับการชำรุดซ้ำ หากมีการออกไปงานซ้ำซ้อนกัน (เช่น กรณีช่างออกไปแล้ว พบว่าไม่สามารถซ่อมที่พื้นที่เกิดเหตุได้) บรรทัดที่ 8 เป็นการหาว่าอุปกรณ์ซ่อมเสร็จเมื่อใด โดยอาศัยการปรับ สถานะจาก ระงับการใช้งาน เป็นอนุญาตให้ใช้ได้

3) กฎหาสาเหตุของอุปกรณ์ชำรุด กฎนี้ ประกอบด้วย กลุ่มของกฎย่อย ๆ เพื่อประเมินสาเหตุการชำรุดว่าน่าจะ เกิดจากอะไร โดยคาดการณ์จากรายการเบิกอะไหล่

ที่เกิดภายในช่วงเวลาเริ่มต้น และสิ้นสุดการทำงานในใบเบิก อะไหล่ที่เกิดขึ้นระหว่างเวลาที่ชำรุด ดังแสดงเป็นตัวอย่าง ในภาพที่ 9 ซึ่งในบรรทัดที่ 4-7 กำหนดว่า หากอะไหล่เป็น ประเภ็นชนิดหนึ่ง ประเมินว่าสาเหตุ คือ น้ำมันรั่ว ซึ่งจะต้องใช้ความรู้จากผู้เชี่ยวชาญมาเขียนเป็นกฎ ให้สอดคล้องกับสาเหตุที่เป็นไปได้ต่าง ๆ

เพื่อให้โปรแกรมตรวจหาการชำรุดสามารถประมวลผล ข้อมูลจำนวนมากได้บน QEDFA Framework เมื่อผู้ใช้ ส่งประมวลผลผ่านทางส่วนเชื่อมต่อผู้ใช้แบบเว็บ ระบบจะสร้างสคริปต์งานหลายอัน โดยแต่ละสคริปต์ งานจะทำงานกับข้อมูลคนละชุดกัน (ด้วยการแบ่งชุดข้อมูล ตามเขตพื้นที่รับผิดชอบในหน่วยงาน) ซึ่งจะทำให้เกิด กลุ่มงานตรวจหาการชำรุดที่จะสามารถส่งกระจายไปประมวลผล ยังทรัพยากรที่อยู่ในคลาวด์คอมพิวเตอร์ภายใต้ QEDPA Framework ได้ซึ่งงานตรวจหาการชำรุดแต่ละงานจะดึงชุดข้อมูล ของตัวเองจากฐานข้อมูลข้อเท็จจริงเข้าสู่หน่วยความจำ แล้วทำการอนุมาน เพื่อสร้างผลสรุปของอุปกรณ์ชำรุดต่อไป

ตารางที่ 1 โครงสร้างข้อมูลในฐานข้อเท็จจริง

ตารางและฟิลด์ที่สำคัญ	คำอธิบาย
equipment(id, type, brand, model)	ระเบียบข้อมูลอุปกรณ์
inv(date, equip_id, status, user_id)	เป็นระเบียบข้อมูลในคลังพัสดุเมื่อเกิดการเบิกจ่ายอุปกรณ์ (disburse) การรับคืน (return) จากผู้ใช้งาน รวมถึงการระงับการใช้งาน (suspend) อุปกรณ์ในคลังพัสดุหากชำรุด และอนุญาตให้ใช้ได้ (avail)
spare_part_requisition(id, date, equip_id, spare_part_item, amount, technician_id)	ใบเบิกอะไหล่ ประกอบด้วยฟิลด์หมายเลขอุปกรณ์ ชนิดของอะไหล่ จำนวนที่เบิก และชื่อผู้เบิก
work_order(id, equip_id, issue_date, complete_date, technician_id, job_description)	ใบงานที่ออกเพื่อให้ช่างเดินทางไปซ่อมบำรุงที่จุดเกิดเหตุ ประกอบด้วยฟิลด์ วันที่ออกไปงาน ชื่อผู้ปฏิบัติงาน รายละเอียดที่ปฏิบัติ และวันที่สำเร็จ

ตารางที่ 2 โครงสร้างข้อมูลของข้อสรุป

ตารางและฟิลด์ที่สำคัญ	คำอธิบาย
fault(id, disbursed_date, fault_date, fixed_date, cause, user_id)	ประกอบด้วยหมายเลขอุปกรณ์ วันที่เบิกไปใช้ วันที่ชำรุด วันที่ซ่อมเสร็จ สาเหตุการชำรุด ชื่อผู้ใช้งาน



```

1: IF
2:   There exists $w: work_order()
3:   There exists $d: inv(equip_id == $w.equip_id, status == 'disburse',
                        date < $w.issue_date)
4:   Not inv(status == 'disburse', $d.date < date < $w.issue_date)
5: THEN
6:   New fault($d.equip_id, $d.date, $w.issue_date, $w.complete_date, 'unknown',
$d.user_id)

```

ภาพที่ 7 ตัวอย่างกฎที่ใช้ระบุอุปกรณ์ชำรุดหากถูกซ่อมแซมที่จุดเกิดเหตุ

```

1: IF
2:   There exists $r: inv(status == 'return')
3:   There exists $s: inv(equip_id == $d.equip_id, status == 'suspend')
4:   Not inv($r.date < date < $s.date) // suspend immediately follows return
5:   There exists $d: inv(status == 'disburse', equip_id == $r.equip_id)
6:   Not inv($d.date < date < $r.date) // return immediately follows disburse
7:   Not work_order(equip_id == $d.equip_id, $r.date - issue_date <= 3 days) // prevent
duplication
8:   There exists $a: inv(equip_id == $d.equip_id, status == 'avail', $s.date < date )
9:   Not inv($s.date < date < $a.date) // avail immediately follows suspend
10: THEN
11:   New fault($d.equip_id, $d.date, $r.date, $a.date, 'unknown', $d.user_id)

```

ภาพที่ 8 ตัวอย่างกฎที่ใช้ระบุอุปกรณ์ชำรุดจากการส่งคืนคลังในสภาพที่ห้ามนำไปใช้งาน

```

1: IF
2:   There exists $f: fault(cause == 'unknown')
3:   There exists $s: spare_part_requisition(equip_id == $f.id, $f.fault_date <
date < $f.fixed_date)
4:   There exists $g: gasket($s.spare_part_item == name)
5:   There exists $e: equipment(id == $f.id, type == 'tractor')
6: THEN
7:   $f.cause = 'oil leak'

```

ภาพที่ 9 ตัวอย่างกฎที่ใช้ประเมินสาเหตุที่อุปกรณ์ชำรุด

4. ผลการดำเนินงาน

4.1 ผลการทดสอบประสิทธิภาพการประมวลผล KBS บน QEDPA Framework

ผู้วิจัยได้ทำการทดสอบโดยการส่งงานตรวจหาการชำรุดเพื่อวิเคราะห์ข้อมูลอุปกรณ์ชำรุด โดยปรับใช้พารามิเตอร์ต่าง ๆ 4 ชุดเพื่อให้งานแต่ละชุดมีเวลาประมวลผลโดยเฉลี่ยที่ 1, 3, 6 และ 10 นาที ตามลำดับ โดยมีงานทั้งหมดรวมกัน 126 งาน งานจะถูกสุ่มเพื่อส่งเข้าระบบด้วยช่วงห่างระหว่างงานเฉลี่ย (Interarrival Time) 30 วินาทีตามการแจกแจงแบบเอ็กซ์โพเนนเชียล (Exponential Distribution) จากนั้น ทำการเปรียบเทียบเวลาสะสมเฉลี่ยที่งานรอคอยในคิวและเวลาที่งานประมวลผลเสร็จเฉลี่ยระหว่างการใช้ QEDPA Framework ที่มีการเพิ่มลดจำนวนเครื่องเสมือนโดยอัตโนมัติ และการกำหนดจำนวน

เครื่องเสมือนคงที่จำนวน 2, 5 และ 9 เครื่อง ในแต่ละกรณี จะมีการทดสอบ 5 ครั้งเพื่อหาค่าเฉลี่ย ทั้งนี้ เครื่องเสมือนทั้งหมดถูกกำหนดให้มีหน่วยประมวลผล 2 แกน และหน่วยความจำ 4 GB สำหรับชุดข้อมูลที่ใช้ในการทดสอบมีลักษณะดังแสดงในตารางที่ 3

ตารางที่ 3 ลักษณะของชุดข้อมูลที่ใช้ทดสอบ

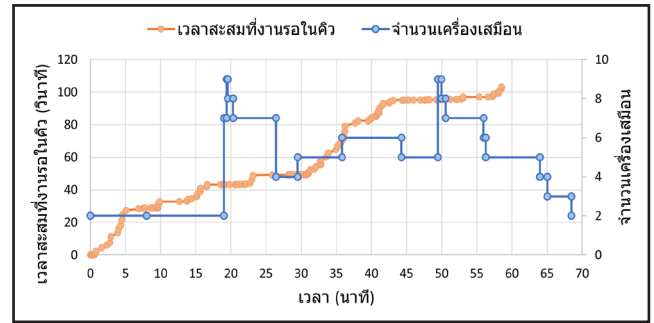
ตารางข้อมูล	จำนวนฟิลด์	จำนวน ระเบียน	ขนาดไฟล์
equipment	10	350,000	54 MB
inv	25	654,000	158 MB
spare_part_requisition	27	2,400,000	600 MB
work_order	21	501,000	145 MB

ภาพที่ 10 แสดงให้เห็นถึงการทำงานของ QEDPA Framework ที่สามารถเพิ่มลดจำนวนเครื่องเสมือนได้อย่างยืดหยุ่นตาม อัลกอริทึม Auto Scale ที่ปรับเปลี่ยนตามภาระงานที่รออยู่ในคิวของระบบจัดการงาน โดยจำนวนเครื่องเสมือนจะเพิ่มขึ้นเมื่องานที่รออยู่ในคิวมีระยะเวลาที่รอคอยถึงระดับที่กำหนดไว้ จากภาพ เมื่อเวลาผ่านไปประมาณ 20 นาที จำนวนเครื่องเสมือนจะเพิ่มขึ้นจาก 2 เครื่องเป็น 7, 8 และ 9 เครื่องตามลำดับ เนื่องจากมีงานที่มีระยะเวลาการรอคอยเกินกว่าระดับที่กำหนด จำนวนมากขึ้นเรื่อย ๆ จากนั้นด้วยจำนวนเครื่องเสมือน 9 เครื่อง ทำให้สามารถรันงานได้จนกระทั่งจำนวนงานที่รอคอยในคิวเกินระดับที่กำหนดลดลงน้อยกว่าจำนวนเครื่องเสมือน QEDPA Framework จึงเริ่มลดจำนวนเครื่องเสมือนลงในช่วงเวลาที่ 20 - 30 นาที จากนั้นในช่วงเวลาที่ 30 - 50 นาที จะมีการเพิ่มลดเครื่องเสมือนอยู่ระหว่าง 5 - 8 เครื่อง หลังจากนั้นที่ 50 จำนวนงานที่รออยู่ในคิวเริ่มลดลงเรื่อย ๆ เพราะงานส่วนใหญ่เสร็จสิ้นแล้ว และเครื่องเสมือนไม่ได้กำลังประมวลผลงานใด ๆ QEDPA Framework จะทำการลดจำนวนเครื่องเสมือนลงจาก 9 เครื่อง เหลือ 2 เครื่อง ทั้งนี้ QEDPA Framework สามารถรันงานจนแล้วเสร็จทั้งหมด 126 งาน โดยใช้เวลาทั้งสิ้นประมาณ 68 นาที สำหรับเวลาสะสมที่งานรออยู่ในคิวคือผลรวมของเวลาที่งานเข้ามาจนเริ่มต้นประมวลผล ซึ่งมีค่าเพิ่มขึ้นเรื่อย ๆ ตั้งแต่งานแรกจนถึงงานสุดท้าย

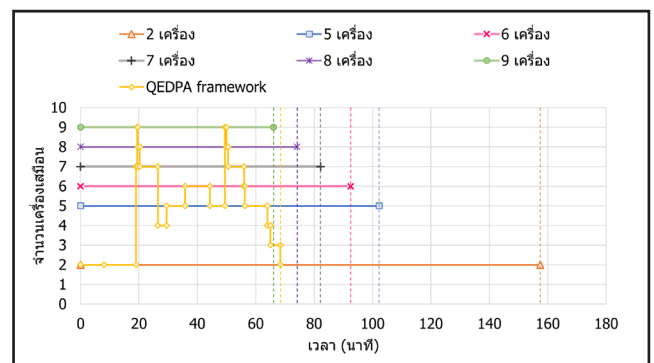
ภาพที่ 11 แสดงให้เห็นถึงจำนวนเครื่องเสมือนที่มีการเพิ่มและลดตาม Auto Scale อัลกอริทึมของ QEDPA Framework ซึ่งใช้เวลาประมวลผลงานเสร็จที่เวลาประมาณ 68 นาที โดยใกล้เคียงกับการใช้จำนวนเครื่องเสมือนคงที่ 9 เครื่องที่ใช้เวลาประมาณ 66 นาที ในขณะที่เครื่องเสมือนคงที่จำนวน 2, 5, 6, 7, และ 8 เครื่องใช้เวลาประมาณ 158, 102, 92, 82, และ 74 นาที ตามลำดับ

ภาพที่ 12 แสดงให้เห็นว่า QEDPA Framework ใช้เวลาประมวลผลเฉลี่ยใกล้เคียงกับการใช้เครื่องเสมือนจำนวน 9 เครื่อง และใช้เวลาในการประมวลผลน้อยกว่าประมาณ 2 และ 1.2 เท่า เมื่อเทียบกับการใช้เครื่องเสมือนจำนวน 2 และ 5 เครื่อง ตามลำดับ

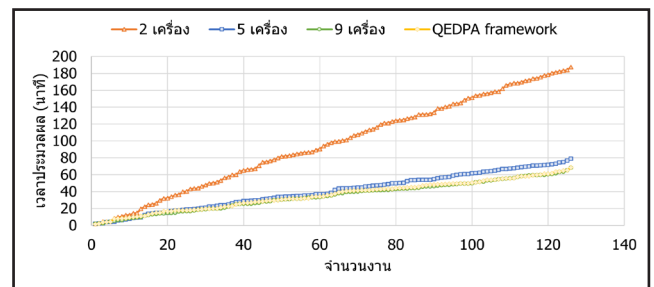
ภาพที่ 13 แสดงให้เห็นว่า QEDPA Framework มีเวลาสะสมเฉลี่ยของงานที่รอในคิวของระบบจัดการงานใกล้เคียงกับการใช้เครื่องเสมือนจำนวน 9 เครื่อง และมีเวลาสะสมที่งานรอในคิวน้อยกว่าประมาณ 75 และ 7 เท่า เมื่อเทียบกับการใช้จำนวนเครื่องเสมือนที่ 2 และ 5 เครื่อง ตามลำดับ



ภาพที่ 10 ความสัมพันธ์ระหว่างการเพิ่มลดจำนวนทรัพยากรคอมพิวเตอร์เสมือนบนคลาวด์คอมพิวเตอร์กับจำนวนเวลาสะสมที่งานรออยู่ในคิวโดยใช้ QEDPA Framework

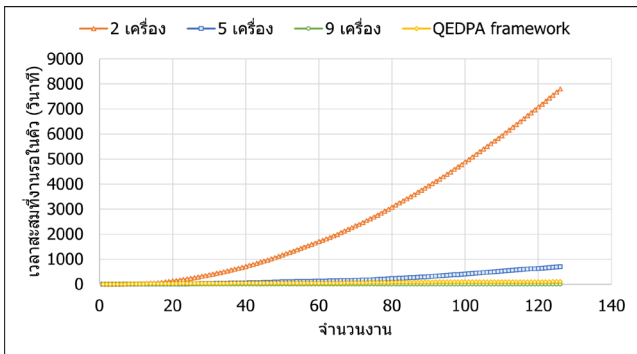


ภาพที่ 11 การเปรียบเทียบจำนวนเครื่องเสมือนกับเวลาที่ใช้ประมวลผลระหว่าง QEDPA Framework กับแบบใช้จำนวนเครื่องเสมือนคงที่



ภาพที่ 12 การเปรียบเทียบเวลาการประมวลผลเฉลี่ยระหว่างการใช้ QEDPA Framework กับการใช้เครื่องเสมือนที่มีจำนวนคงที่

นอกจากนี้ ผู้วิจัยยังได้คำนวณค่าใช้จ่ายที่จะเกิดขึ้นในการทดสอบนี้ด้วยการเทียบเคียงกับค่าใช้บริการคลาวด์คอมพิวเตอร์สาธารณะของ Amazon AWS พร้อมทั้งเปรียบเทียบประสิทธิภาพต่อราคาระหว่างการใช้จำนวนเครื่องเสมือนแบบคงที่กับ QEDPA Framework โดยค่าประสิทธิภาพต่อราคาสามารถคำนวณได้ตามสมการ



ภาพที่ 13 การเปรียบเทียบเวลาสมมูลที่งานรออยู่ในคิวระหว่างการใช้ QEDPA Framework กับการใช้เครื่องเสมือนที่มีจำนวนคงที่

$$\text{ประสิทธิภาพต่อราคา} = \frac{\text{performance}}{\text{ค่าบริการ}}$$

และ

$$\text{performance} = \text{GFLOPS}_{\text{base}} \times \left(\frac{t_{\text{base}}}{t} \right)$$

โดยที่ performance คือ ประสิทธิภาพเมื่อใช้เครื่องเสมือนจำนวนคงที่และยืดหยุ่นภายใต้ QEDPA Framework ตามลำดับค่า $\text{GFLOPS}_{\text{base}}$ คือประสิทธิภาพของหน่วยประมวลผลซึ่งได้จากการใช้โปรแกรมวัดประสิทธิภาพ LINPACK Benchmark [17] ในการคำนวณแก็ระบบสมการเชิงเส้นบนเครื่องเสมือนอ้างอิงจำนวน 2 เครื่อง มีหน่วยเป็น 10^9 floating point operations per second (Gflops) และ t_{base} คือเวลาประมวลผลเสร็จของเครื่องอ้างอิง² (Baseline Machine) ส่วนค่า t คือเวลาที่ประมวลผลเสร็จเมื่อใช้จำนวนเครื่องเสมือนคงที่และเมื่อใช้ QEDPA Framework ทั้งนี้ที่ต้องกำหนดเครื่องอ้างอิงขึ้นมานั้น เนื่องจากเวลาที่งานประมวลผลเสร็จในแต่ละกรณีไม่เท่ากัน จึงต้องใช้เครื่องอ้างอิงเป็นตัวเปรียบเทียบหาประสิทธิภาพ

ตารางที่ 4 แสดงค่าเวลาที่ประมวลผลเสร็จ (t) ประสิทธิภาพ ค่าบริการ AWS และประสิทธิภาพต่อราคาในกรณีต่าง ๆ ซึ่งเห็นได้ว่า QEDPA Framework มีประสิทธิภาพต่อราคาที่สูงกว่าการใช้จำนวนเครื่องเสมือนแบบคงที่และสามารถจัดการในเรื่องของประสิทธิภาพต่อราคาในการใช้บริการคลาวด์คอมพิวติงได้อย่างสมดุลซึ่งใช้เวลาในการรันเสร็จใกล้เคียงกับการใช้เครื่องเสมือนจำนวนสูงสุด 9 เครื่อง และเสียค่าใช้จ่ายใกล้เคียงกับการใช้เครื่องเสมือนจำนวน 2 เครื่อง

4.2 การทดสอบใช้งาน

ผู้ใช้งานสามารถใช้งานระบบผ่านทางส่วนเชื่อมต่อผู้ใช้แบบเว็บ โดยจะต้องอัปโหลดไฟล์เข้าสู่ระบบเพื่อตรวจสอบความถูกต้องของข้อมูล ดังแสดงในภาพที่ 14 ในหน้าจอก็จะมีปุ่มที่ให้ผู้ใช้อัปโหลดไฟล์ จากนั้นเมื่อผู้ใช้งานกดปุ่มเริ่มการตรวจสอบความถูกต้อง ระบบจะส่งงานตรวจสอบความถูกต้องเข้าไปในระบบจัดการงานเพื่อรอการประมวลผลไฟล์ที่ผ่านการตรวจสอบว่าถูกต้องแล้วเท่านั้นที่จะให้ผู้ใช้งานสามารถทำการนำเข้าสู่ฐานข้อมูลต่อไปได้ โดยผู้ใช้งานจะต้องกดปุ่มเริ่มการนำเข้า

ภาพที่ 14 หน้าจอสำหรับการตรวจสอบและนำเข้าไฟล์

หลังจากที่งานตรวจสอบความถูกต้องของไฟล์ถูกประมวลผลเสร็จสิ้นแล้ว จะมีรายงานผลการตรวจสอบไฟล์ภาพที่ 15 แสดงผลการตรวจสอบความถูกต้องของไฟล์ก่อนการนำเข้า ซึ่งจะแจ้งให้ผู้ใช้งานทราบว่าข้อมูลแต่ละรายการมีความผิดพลาดหรือไม่ เช่น ไฟล์ที่จำเป็นไม่มีค่า หรือรูปแบบไม่ตรงตามที่กำหนด เป็นต้น หากมีความผิดพลาดผู้ใช้งานจะต้องทำการแก้ไขไฟล์ต้นฉบับและส่งเข้ามาดำเนินการตรวจสอบอีกครั้งหนึ่ง

ภาพที่ 15 หน้าจอแสดงผลการตรวจสอบไฟล์

² ค่า Gflops ที่วัดได้จากโปรแกรม LINPACK และเวลาประมวลผลเสร็จสำหรับเครื่องอ้างอิงมีค่าประมาณ 122 Gflops และ 157 นาที ตามลำดับ

ตารางที่ 4 ผลการเปรียบเทียบการใช้เครื่องเสมือนที่มีจำนวนคงที่กับแบบใช้ QEDPA Framework ในมิติต่าง ๆ

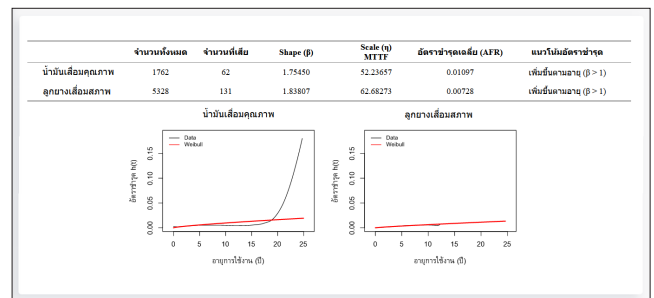
จำนวนเครื่องเสมือน	เวลารวมสะสมที่งานรอในคิว (นาทื)	เวลาที่ประมวลผลเสร็จ (t) (นาทื)	ประสิทธิภาพ (Gflops)	ค่าบริการ AWS (US\$)	ประสิทธิภาพต่อราคา
2 (Baseline Machine)	129.89	157.38	121.99	0.43	283.69
5	11.54	102.21	187.83	0.71	264.55
6	6.24	92.52	207.50	0.77	269.49
7	2.57	82.15	233.70	0.72	324.58
8	0.13	74.23	258.63	0.83	311.61
9	0.02	66.08	290.53	0.83	350.04
QEDPA Framework	1.71	68.45	280.47	0.47	596.76

เมื่อผู้ใช้กดปุ่มเริ่มการนำเข้าไฟล์ ระบบจะดำเนินการนำเข้าไฟล์เข้าสู่ฐานข้อมูลข้อเท็จจริง ซึ่งในกระบวนการนี้ประกอบด้วยการแปลงและการตรวจหาอุปกรณ์ซ้ำชุด โดยระบบจะสร้างและส่งงานให้ระบบจัดการงาน เพื่อดำเนินการประมวลผลหลังจากเสร็จสิ้น ผลลัพธ์จะถูกบันทึกไว้ในฐานข้อมูลซึ่งผู้ใช้งานสามารถส่งงานวิเคราะห์ข้อมูลการซ้ำชุด เช่น การหาค่าอัตราซ้ำชุดของอุปกรณ์ตามเงื่อนไขต่าง ๆ ที่ผู้ใช้เลือกตั้งแสดงในภาพที่ 16

ภาพที่ 16 หน้าจอสำหรับการวิเคราะห์อัตราซ้ำชุด

เมื่อผู้ใช้กดปุ่มวิเคราะห์ ระบบจะส่งงานวิเคราะห์หาอัตราซ้ำชุดจากตามตัวเลือกที่ผู้ใช้ได้กำหนดไว้และรอจนกระทั่งงานถูกประมวลผลจนเสร็จสิ้น จากนั้น ผู้ใช้จึงจะเข้าดูผลการวิเคราะห์ผ่านทางหน้าจอ ดังแสดงในภาพที่ 17

เช่นเดียวกับระบบ KBS อื่นๆ ที่ความถูกต้องของผลลัพธ์ของการตรวจหาอุปกรณ์ซ้ำชุดจะขึ้นอยู่กับความถูกต้องและความสมบูรณ์ของกฎ ที่บางครั้งผู้เชี่ยวชาญก็ไม่สามารถ



ภาพที่ 17 หน้าจอแสดงผลการวิเคราะห์อัตราซ้ำชุด

ถ่ายทอดออกมาได้ครบถ้วน ดังนั้น จึงต้องดำเนินการด้วยกระบวนการที่มีประสิทธิภาพตามแนวทางต่าง ๆ ที่มีการนำเสนอไว้ใน [8], [18] อย่างไรก็ตาม กฎย่อมมีการปรับเปลี่ยนได้ตามกระบวนการทำงานที่เปลี่ยนไป หรือการรับรู้องค์ความรู้เพิ่มเติมจากผู้เชี่ยวชาญ ทำให้ผลลัพธ์มีความแตกต่างไปจากเดิม หรืออาจส่งผลต่อเวลาในการประมวลผลข้อมูลที่ช้าลงหรือเร็วขึ้นได้ซึ่ง QEDPA Framework ยังคงสามารถทำการเพิ่มลดทรัพยากรได้ตามเดิม เพราะรูปแบบของการประมวลผลและการกระจายงานไปบนทรัพยากรคลาวด์แบบยืดหยุ่นยังไม่เปลี่ยนแปลง

5. สรุป

การวิเคราะห์ข้อมูลด้วยการใช้ระบบฐานความรู้ (Knowledge-based System) เพื่อหารูปแบบของความสัมพันธ์

ที่ซับซ้อนในข้อมูลขนาดใหญ่จำเป็นต้องใช้เวลาในการประมวลผลนาน ดังนั้นงานวิจัยนี้นำเสนอวิธีการและ Framework สำหรับแก้ปัญหา ด้วยการประมวลผลระบบฐานความรู้บนคลาวด์คอมพิวติง ด้วยการกระจายการประมวลผลระบบฐานความรู้ไปในทรัพยากรคอมพิวเตอร์ในระบบคลาวด์คอมพิวติงอย่างยืดหยุ่น ด้วยการเพิ่มหรือลดจำนวนทรัพยากรประมวลผลตามปริมาณงานแบบอัตโนมัติ ทำให้การวิเคราะห์ข้อมูลจำนวนมากเสร็จได้ในเวลาอันรวดเร็ว และใช้ทรัพยากรคำนวณอย่างคุ้มค่า มีประสิทธิภาพต่อราคาที่สูงขึ้น Framework นี้ถูกนำไปใช้กับการวิเคราะห์ข้อมูลอุปกรณ์ชำรุด เพื่อช่วยสืบค้นข้อมูล อุปกรณ์ชำรุดให้ถูกต้องครบถ้วน ทำให้การคำนวณหาค่าอัตราการชำรุดแม่นยำมากขึ้น ในอนาคตสามารถนำวิธีการ และ Framework นี้ไปประยุกต์ใช้กับการวิเคราะห์ชุดข้อมูลอื่นที่มีการประมวลผลในลักษณะของ High Throughput Batch Computing เช่นเดียวกับที่ได้นำเสนอในงานวิจัยนี้

6. เอกสารอ้างอิง

- [1] C. Chen, S. Chen, F. Yin, and W. Wang. "Efficient Hybridizing Auto-scaling for OpenStack Platforms." *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, Chengdu, pp. 1079-1085, 2015.
- [2] J. Bao, Z. Lu, J. Wu, S. Zhang, and Y. Zhong. "Implementing a Novel Load-aware Auto Scale Scheme for Private Cloud Resource Management Platform." *2014 IEEE Network Operations and Management Symposium (NOMS)*, Krakow, pp. 1-4, 2014.
- [3] A. A. D. P. Souza, and M. A. S. Netto. "Using Application Data for SLA-Aware Auto-scaling in Cloud Environments." *2015 IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Atlanta, GA, pp. 252-255, 2015.
- [4] M. Wang, D. Zhang, and B. Wu. "A Cluster Autos Based on Multiple Node Types in Kubernetes." *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, pp. 575-579, 2020.
- [5] Kubernetes Cluster Autoscaler, *Google Kubernetes Engine's Cluster Autoscaler*. Available Online at <https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-autoscaler>, accessed on 1 September 2022.
- [6] Amazon Web Services, *AWS ParallelCluster Auto Scaling*. Available Online at <https://docs.aws.amazon.com/parallelcluster/latest/ug/autoscaling.html>, accessed on 1 September 2022.
- [7] A. Luckow, I. Paraskevagos, G. Chantzialexiou, and S. Jha. "Hadoop on HPC: Integrating Hadoop and Pilot-Based Dynamic Resource Management." *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2016.
- [8] S. L. Kendal, and M. Creen. *An Introduction to Knowledge Engineering*. Springer, 2007.
- [9] S. J. Russell, and P. Norvig. "Artificial Intelligence A Modern Approach", Fourth Edition, pp. 356, Pearson, 2022.
- [10] K. Jahr, and A. Borrmann. "Semi-automated Site Equipment Selection and Configuration through Formal Knowledge Representation and Inference." *Advanced Engineering Informatics*, Vol. 38, pp. 488-500, 2018.
- [11] M. Ju, and Q. Qian. "Materials Knowledge Reasoning with Production based System." *Computational Materials Science*, Vol. 173, February 2020.
- [12] M. Li, W. Lu, D. Xiang, and Z. Wen. "Design and Realization of Transformer Fault Diagnostic Expert System Based on Drools." *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2015.
- [13] *Slurm Workload Manager*. Available Online at <https://slurm.schedmd.com/>. accessed on 1 September 2022.
- [14] *OpenStack: Open Source Cloud Computing Infrastructure*. Available Online at <https://www.openstack.org>. accessed on 1 September 2022.
- [15] *Drools*. Available Online at <https://www.drools.org/>. accessed on 1 September 2022.



- [16] A. Fobel, and N. Subramanian. "Comparison of the Performance of Drools and Jena Rule-based Systems for Event Processing on the Semantic Web." *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2016.
- [17] *LINPACK Benchmark*. Available Online at <https://netlib.org/benchmark/>. accessed on 1 September 2022.
- [18] N. R. Milton. *Knowledge Acquisition in Practice A Step-by-step Guide*. Springer, 2007.