การพัฒนาแบบจำลองระบบและรูปแบบการแสดงผล เพื่อการจัดการศูนย์ข้อมูล

GRAPHICAL MODELING & INTERACTIVE VISUALIZATION FOR DATA CENTER MANAGEMENT

นพกร ใช้บุญเรือง
นนทยา วิไลเลิศสมบัติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

การพัฒนาแบบจำลองระบบและรูปแบบการแสดงผล เพื่อการจัดการศูนย์ข้อมูล

GRAPHICAL MODELING & INTERACTIVE VISUALIZATION FOR DATA CENTER MANAGEMENT

นพกร ใช้บุญเรือง
นนทยา วิไลเลิศสมบัติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ปริญญานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาแบบจำลองระบบและรูปแบบการแสดงผลเพื่อการจัดการศูนย์ข้อมูล
GRAPHICAL MODELING & INTERACTIVE VISUALIZATION FOR DATA CENTER MANAGEMENT

ผู้จัดทำ

1. นางสาวนนทยา วิไลเลิศสมบัติ รหัสนักศึกษา 54010654
2. นายนพกร ใช้บุญเรือง รหัสนักศึกษา 54010656

Orathai Sangpetch อาจารย์ที่ปรึกษา

(ดร. อรทัย สังข์เพ็ชร)

อาจารย์ที่ปรึกษาร่วม

(ดร. อักฤทธิ์ สังข์เพ็ชร)

การพัฒนาแบบจำลองระบบและรูปแบบการแสดงผล เพื่อการจัดการศูนย์ข้อมูล

นางสาวนนทยา	วิไลเลิศสมบัติ	54010654
นายนพกร	ใช้บุญเรือง	54010656
อาจารย์อรทัย	สังข์เพ็ชร	อาจารย์ที่ปรึกษา
อาจารย์อักฤทธิ์	สังข์เพ็ชร	อาจารย์ที่ปรึกษาร่วม
ปีการศึกษา	2557	

บทคัดย่อ

ในการบริหารจัดการศูนย์ข้อมูลนั้นผู้ดูแลจำเป็นที่จะต้องตรวจสอบและทำความเข้าใจข้อมูลเกี่ยวกับปริมาณการใช้งานทรัพยากร การเชื่อมต่อ การส่งข้อมูลในเครือข่าย และข้อมูลอื่นๆ เพื่อประกอบการตัดสินใจ วางแผนและเป็นข้อมูลประกอบในการแก้ปัญหาต่างๆ เมื่อจำนวนเครื่องในศูนย์ข้อมูลมีมากขึ้นเนื่องจากการนำเอาเทคโนโลยีเวอร์ชวลไลเซชั่น (Virtualization) เข้ามาใช้งานในการให้บริการการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) เมื่อรวมกับความซับซ้อนของโครงสร้างพื้นฐาน (Infrastructure) ในการให้บริการ ทำให้การบริหารจัดการศูนย์ข้อมูลทำได้ยากขึ้นและต้องอาศัยเครื่องมือหลายๆ อย่างในการเก็บข้อมูลที่มีจำนวนมากซึ่งยากแก่การตีความและตัดสินใจ

โครงงานที่จัดทำขึ้นนี้มีจุดประสงค์เพื่อที่จะสร้างระบบรวบรวมข้อมูลจากเครื่องมือเหล่านั้นและจัดทำแบบจำลองขึ้นเพื่อให้ง่ายต่อการคัดกรองและการค้นหาข้อมูลที่เกี่ยวข้องง่ายขึ้น ซึ่งจะช่วยให้ผู้ดูแลสามารถทำความเข้าใจข้อมูลเหล่านั้นได้อย่างรวดเร็วและสามารถคาดเดาแนวโน้มการใช้งานหรือปัญหาที่จะเกิดขึ้นในอนาคตได้

นอกจากนี้ระบบที่เสนอในโครงงานถูกพัฒนาให้รองรับข้อมูลจากเครื่องมือหรือเทคโนโลยีอื่นๆที่อาจจะเพิ่มเข้ามาในอนาคต ตัวอย่างเช่น ถ้ามีข้อมูลจากระบบความปลอดภัยที่สามารถแจ้งเตือนปัญหาความปลอดภัย หรือเครื่องมือที่เก็บบันทึกเหตุการณ์ต่างๆที่เกิดขึ้นที่อุปกรณ์ ระบบที่นำเสนอสามารถนำข้อมูลเหล่านี้มารวมกับข้อมูลพื้นฐานที่มีอยู่และแจ้งเตือนในรูปแบบที่บูรณาการมากขึ้น

GRAPHICAL MODELING & INTERACTIVE VISUALIZATION FOR DATA CENTER MANAGEMENT

Ms. Nontaya	Wilailertsombat	54010654
Mr. Noppakorn	Chaiboonruang	54010656
Mrs. Orathai	Sangpetch	Advisor
Mr. Akkarit	Sangpetch	Co-Advisor
Academic year	2014	

ABSTRACT

A datacenter typically contains a number of interconnected physical devices and logical components, such as servers, virtual machines (VMs), network switches, virtual switches, and so on. With the growth of the datacenter, the underlying infrastructure has become more complex and challenging to have a prompt response when a problem arises. This is because the operators may need to go through several monitoring tools in order to investigate the problem. To mitigate this challenge, we propose a graphical tool, which combines and integrates the data from different tools into one view and then interactively display only the relevant information to users. The goal of this tool is to improve the manageability of the data center by providing a more effective visualization tool beyond existing tabulated user interface. The tool deploys our data model, which connects various types of information from different tools together into a graph structure. With our tool, operators can easily locate, identify and search the data of interest quickly without apprehending a variety set of tools. As a result, operators can manage the datacenter more effectively and respond to problems more promptly.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีด้วยความช่วยเหลือจากหลายฝ่ายทั้งในทางตรงและทางอ้อม โครงงานฉบับนี้จะสำเร็จลงไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้

อาจารย์ที่ปรึกษาทั้งสองท่านคือ ดร. อักฤทธิ์ สังข์เพ็ชร และ ดร. อรทัย สังเพ็ชร เป็นผู้ที่ให้คำแนะนำ ปรึกษา และให้ความช่วยเหลือตลอดการทำโครงงาน ซึ่งทำให้การทำงานต่างๆเป็นไปได้อย่างราบรื่นและทำให้โครงงานฉบับนี้สำเร็จไปได้ด้วยดี

บริษัท อินเทอร์เน็ตประเทศไทย จำกัด (มหาชน) หรือ INET ซึ่งเป็นผู้ให้บริการ Internet Data Center (IDC) และ Cloud Services ในประเทศไทย ที่ได้ให้คำปรึกษาต่างๆเกี่ยวกับวิธีการดำเนินงาน ข้อมูล และขั้นตอนการปฏิบัติต่างๆที่ใช้งานจริง ในการบริหารจัดการและดูแลศูนย์ข้อมูล

อาจารย์และบุคลากรต่างๆในสาขาวิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำและสั่งสอนความรู้ต่างๆมาตลอด รวมถึงห้องวิจัยและพัฒนาการรักษาความปลอดภัยข้อมูล (Information Security Advisory Group: ISAG) ที่ได้เอื้อเฟื้อสถานที่ในการทำวิจัยและพัฒนาโครงงาน

ในท้ายที่สุดนี้ขอขอบคุณบิดา มารดา และครอบครัวที่ได้เลี้ยงดูและสั่งสอน พร้อมทั้งให้โอกาสในการศึกษา และให้กำลังใจเสมอมา

นนทยา วิไลเลิศสมบัติ
นพกร ใช้บุญเรือง

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูปภาพ VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินการ 2

ประโยชน์ที่คาดว่าจะได้รับ 1.5 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 เทคโนโลยีเวอร์ชวลไลเซชั่น (Virtualization Technology) 4

2.2 การประมวลผลแบบกลุ่มเมฆ)Cloud Computing) 5

2.3 การบริหารจัดการศูนย์ข้อมูล 6

2.4 ฐานข้อมูลแบบ NoSQL 8

2.5 Distributed system 11

2.6 Data visualization 12

บทที่ 3 การวิเคราะห์และออกแบบระบบ 14

ภาพรวมของระบบ 3.1 14

3.2 โครงสร้างในการพัฒนาระบบ 16

3.3 แบบจำลองข้อมูล 18

สารบัญ (ต่อ)

หน้า

3.4 ส่วนแสดงผล DataVis. 22

บทที่ 4 การทดลองและผลการทดลอง 26

4.1 การทดสอบเกี่ยวกับความถูกต้องของโปรแกรม 26

4.2 การทดสอบเกี่ยวกับการแสดงผลข้อมูล 32

บทที่ 5 บทสรุปและข้อเสนอแนะ 38

5.1 บทสรุปของโครงงาน 38

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข 38

5.3 แนวทางในการพัฒนาต่อ 39

บรรณานุกรม 40

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 3.1 แสดงรายละเอียดชนิดของวัตถุในแบบจำลอง..... 18

สารบัญรูป

รูปที่ **หน้า**

รูปที่ 2.1 ภาพเปรียบเทียบข้อแตกต่างก่อนและหลังการใช้งานเทคโนโลยีเวอร์ชั่วไลเซชั่น 4
รูปที่ 2.2 ข้อแตกต่างของการให้บริการแบบต่างบน Cloud 6
รูปที่ 2.3 แนวทางการบริหารจัดการศูนย์ข้อมูล 7
รูปที่ 2.4 การบริหารจัดการเครื่องคอมพิวเตอร์เสมือนโดย VMware vCenter 8
รูปที่ 2.5 แสดงตัวอย่าง field และ value 11
รูปที่ 3.1 ภาพแสดงการติดต่อกันระหว่างสองส่วนของระบบ 14
รูปที่ 3.2 ภาพการแสดงผลของ DataVis. 15
รูปที่ 3.3 แผนภาพการเชื่อมต่อระหว่างส่วนย่อยของระบบ 16
รูปที่ 3.4 ตัวอย่างแบบจำลองและความสัมพันธ์ 20
รูปที่ 3.5 รายละเอียดแต่ละส่วนภายในแบบจำลองข้อมูล 21
รูปที่ 3.6 หน้าแรกของการเข้าใช้งาน DataVis. 22
รูปที่ 3.7 หน้าหลักของการใช้งาน DataVis 22
รูปที่ 3.8 ตัวอย่างการเลือกแสดงความสัมพันธ์เฉพาะอุปกรณ์ 23
รูปที่ 3.9 ตัวอย่างการเน้นความสัมพันธ์และข้อมูลของอุปกรณ์ 23
รูปที่ 3.10 ตัวอย่างการแสดงปัญหาที่เกิดขึ้น 24
รูปที่ 3.11 ผลลัพธ์เมื่อผู้ใช้เลือกดูปัญหาที่เกิดขึ้น 24
รูปที่ 3.12 ผลลัพธ์เมื่อค้นหาด้วยชื่อ Host0 24
รูปที่ 4.1 ผลลัพธ์จากการทดลองดึงข้อมูล 27
รูปที่ 4.2 ผลการทดลองแปลงข้อมูลโดยใช้ modeler 27
รูปที่ 4.4 ผลลัพธ์จากการทดลองที่ 4.1.3.1 29
รูปที่ 4.5 ผลลัพธ์จากการทดลองที่ 4.1.3.2 30
รูปที่ 4.6 ผลลัพธ์จากการทดลองที่ 4.1.3.3 30
รูปที่ 4.7 ผลลัพธ์จากการทดลองที่ 4.1.3.4 31
รูปที่ 4.8 ผลลัพธ์จากการทดลองที่ 4.1.3.5 32
รูปที่ 4.9 กราฟวงกลมแสดงสาเหตุของปัญหาภายในศูนย์ข้อมูล 33
รูปที่ 4.10 กราฟวงกลมแสดง Tools ที่ใช้ในการหาสาเหตุของปัญหา 33

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 4.11 กราฟแท่งแสดงความรู้สึกในการใช้งาน 34
รูปที่ 4.12 กราฟแท่งแสดงประโยชน์ในการใช้งานฟังก์ชัน 35
รูปที่ 4.13 กราฟแท่งแสดงการเลือกเครื่องมือสำหรับแต่ละการทำงาน 36

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

การบริหารจัดการศูนย์ข้อมูลนั้นจำเป็นต้องอาศัยข้อมูลจำนวนมาก ซึ่งถูกเก็บบันทึกตลอดเวลา นอกจากนี้ในปัจจุบันศูนย์ข้อมูลส่วนใหญ่มักมีการใช้งานเทคโนโลยีเวอร์ชวลไลเซชั่น (Virtualization) ในการจำลองเครื่องคอมพิวเตอร์หลายๆเครื่องภายในเครื่องเพียงเครื่องเดียว ซึ่งเพิ่มความซับซ้อนและจำนวนข้อมูลที่ควรเก็บบันทึกมากขึ้นอีก เมื่อเกิดปัญหาหรือเหตุการณ์ต่างๆขึ้นทำให้ผู้ดูแลระบบไม่สามารถรวบรวมและตีความข้อมูลปริมาณมากเหล่านี้ได้อย่างรวดเร็ว ยิ่งใช้เวลาในการแก้ปัญหานานมากเท่าไหร่ความสูญเสียที่เกิดขึ้นและผลกระทบต่อผู้ใช้ก็จะยิ่งมากขึ้นเท่านั้น ข้อมูลที่เกิดขึ้นในระบบสามารถนำมาใช้ให้เกิดประโยชน์ในการบริหารจัดการศูนย์ข้อมูลได้อีกมาก หากถูกจัดกลุ่มหรือจัดเรียงให้อยู่ในรูปแบบที่เข้าใจได้ง่ายและง่ายต่อการค้นหา ซึ่งปัจจุบันแม้จะมีเครื่องมือที่สามารถทำงานดังกล่าวได้แต่บางส่วนก็ต้องเสียค่าบริการในราคาที่แพงหรือไม่ก็จำเป็นต้องปรับแต่งหรือพัฒนาเพิ่มเติมเพื่อให้ทำงานได้ตามจุดประสงค์ นอกจากนี้การแสดงผลข้อมูลที่มีอยู่ยังเน้นการแสดงผลเป็นส่วนๆ หรือเครื่องมือที่พยายามรวมข้อมูลจากหลายๆ เครื่องมือก็ยังไม่มีการเชื่อมต่อข้อมูลที่ดีทำให้ผู้ใช้หาข้อมูลที่เกี่ยวข้องได้ยากและไม่สามารถเห็นถึงสัญญาณ ผลกระทบหรือแนวโน้มของปัญหาได้ชัดเจน

1.2 วัตถุประสงค์ของโครงงาน

ระบบที่โครงงานนี้นำเสนอประกอบด้วยแบบจำลองข้อมูลในข้อ 1 และส่วนแสดงผลที่สร้างขึ้นในข้อ 2 โดยระบบนี้จะต้องสามารถแสดงผลข้อมูลที่ได้รับอย่างถูกต้องและสามารถแสดงการเปลี่ยนแปลงของข้อมูลใกล้เคียงหรือเท่ากับสถานการณ์ที่เกิดขึ้นจริง

1) ออกแบบแบบจำลองข้อมูลสำหรับเก็บข้อมูลเกี่ยวกับคอมพิวเตอร์เสมือน เครื่องแม่ข่าย และอุปกรณ์อื่นๆ ในศูนย์ข้อมูล ให้อยู่ในรูปแบบที่สามารถนำมาวิเคราะห์ ค้นหา หรือใช้งานต่อได้อย่างมีประสิทธิภาพ แบบจำลองนี้จะต้องสามารถแสดงผลที่เกิดขึ้นจริง (Real-time) และต่อเนื่อง นอกจากนี้ต้องรองรับข้อมูลจากโปรแกรมอื่นๆ เช่น ข้อมูลเกี่ยวกับความปลอดภัยและการโจมตีที่เกิดขึ้น ข้อมูลเกี่ยวกับความสัมพันธ์ของแต่ละองค์ประกอบในศูนย์ข้อมูล การ

แสดงข้อมูลเหล่านี้แบบบูรณาการจะช่วยทำให้ผู้ใช้เล็งเห็นถึงผลกระทบที่เกิดจากปัญหาที่พบได้มีประสิทธิภาพมากขึ้น

2) สร้างส่วนแสดงผลโดยดึงข้อมูลจากแบบจำลองที่สร้างในข้อ 1 ซึ่งส่วนแสดงผลนี้จะสามารถเลือกแสดงเฉพาะข้อมูลที่ผู้ใช้สนใจรวมถึงข้อมูลอื่นๆ ที่มีความเกี่ยวเนื่องกันซึ่งจะถูกแสดงให้อยู่ในรูปแบบที่เข้าใจได้ง่ายและง่ายต่อการสืบค้นข้อมูลต่อ

1.3 ขอบเขตของโครงงาน

1) ระบบสามารถรองรับข้อมูลจากเครื่องมือ VMware vCenter ซึ่งเป็นระบบบริหารจัดการคอมพิวเตอร์เสมือนและเครื่องแม่ข่ายที่ใช้อยู่อย่างแพร่หลาย และ Nagios ซึ่งเป็นเครื่องมือที่ใช้เพื่อบริหารจัดการเครือข่าย
2) ระบบมี API เพื่อรับรองเครื่องมืออื่นๆ นอกเหนือจากที่กล่าวมาในข้อ 1
3) ระบบสามารถสร้างแบบจำลองจากข้อมูลที่ได้รับจากเครื่องมือในข้อ 1 และสามารถแสดงผลแบบจำลองดังกล่าวในรูปแบบที่ผู้ใช้สามารถทำความเข้าใจ ค้นหาและเข้าถึงข้อมูลต่างๆได้ง่ายและมีประสิทธิภาพ
4) ระบบสามารถประมวลผลข้อมูลที่ได้รับจากข้อที่ 1 โดยให้ผลลัพธ์แบบ real-time เพื่อให้ผู้ใช้สามารถเห็นการเปลี่ยนแปลงที่เกิดขึ้นในศูนย์ข้อมูลได้ทันที
5) ระบบสามารถแสดงผลผ่าน Web browser ได้

1.4 วิธีการดำเนินการ

1) ศึกษาเครื่องมือที่ใช้ภายในศูนย์ข้อมูล คือ VMware vCenter และ Nagios และทดลองใช้งานเพื่อเก็บข้อมูลทั้งในส่วนการใช้งาน User Interface ว่ามีข้อดีข้อเสียอย่างไร เพื่อเป็นแนวทางในการพัฒนาส่วนแสดงผลและในส่วนโครงสร้างการเก็บข้อมูลเพื่อเป็นแนวทางในการออกแบบแบบจำลองและให้สามารถดึงข้อมูลส่วนที่ต้องการได้อย่างถูกต้อง
2) ออกแบบระบบทั้งหมด โดยแบ่งออกเป็นสองส่วนหลัก คือ ส่วนสร้างแบบจำลองและส่วนแสดงผล
3) ออกแบบแบบจำลองข้อมูลที่จะใช้ในการเก็บข้อมูลเพื่อแสดงผลต่อไป
4) สร้างส่วนดึงข้อมูลจากเครื่องมือในข้อ 1
5) ทดสอบดึงข้อมูลและพัฒนาปรับปรุงเพื่อให้ได้ข้อมูลตามต้องการ
6) สร้างส่วนประมวลผลข้อมูลให้อยู่ในรูปแบบแบบจำลองที่ได้ออกแบบไว้

7) ทดสอบและประเมินการทำงานของส่วนประมวลผลข้อมูลเพื่อพัฒนาให้ได้ผลลัพธ์ที่ต้องการ
8) สร้างส่วนเก็บข้อมูล
9) ออกแบบการแสดงผลข้อมูล
10) สร้างส่วนการแสดงผลที่แสดงผลผ่าน Web Browser
11) ทดสอบการทำงานภาพรวมของระบบทั้งหมด ประเมินและปรับปรุงแก้ไข
12) สรุปผลและจัดทำรูปเล่มรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

โครงงานนี้สามารถนำไปช่วยเพิ่มประสิทธิภาพในการบริหารจัดการศูนย์ข้อมูล เนื่องจากระบบที่นำเสนอจะช่วยรวบรวมข้อมูลต่างๆ เพื่อใช้ในการตรวจสอบสถานะของเครื่องคอมพิวเตอร์และการบริการต่างๆ รวมถึงการเฝ้าระวังเหตุการณ์ต่างๆ หรือผลกระทบที่อาจจะสืบเนื่องจากเหตุการณ์ก่อนหน้า ผู้ใช้สามารถค้นหาและเข้าถึงข้อมูลที่สนใจได้ง่าย รวมถึงสามารถมองเห็นความสัมพันธ์หรือข้อมูลที่เกี่ยวข้อง เช่น เครื่องคอมพิวเตอร์ และอุปกรณ์ต่างๆ ได้ง่ายขึ้น ซึ่งจะช่วยลดทั้งเวลาในการเรียนรู้การใช้งาน เวลาในการทำความเข้าใจข้อมูล

ระบบมีความสามารถที่จะรองรับการค้นหาได้หลายรูปแบบและจัดให้ใช้งานได้ง่าย โครงงานนี้จะช่วยให้ผู้ใช้สามารถค้นหาและเข้าถึงข้อมูลที่สนใจได้สะดวกรวดเร็วยิ่งขึ้น รวมถึงสามารถมองเห็นความสัมพันธ์หรือข้อมูลที่เกี่ยวข้อง เช่น เครื่องคอมพิวเตอร์ และอุปกรณ์ต่างๆอีกด้วย ซึ่งจะช่วยลดทั้งเวลาในการเรียนรู้การใช้งาน เวลาในการทำความเข้าใจข้อมูล ตัวอย่างการค้นหาโดยออกแบบให้สามารถรองรับการค้นหาดังต่อไปนี้

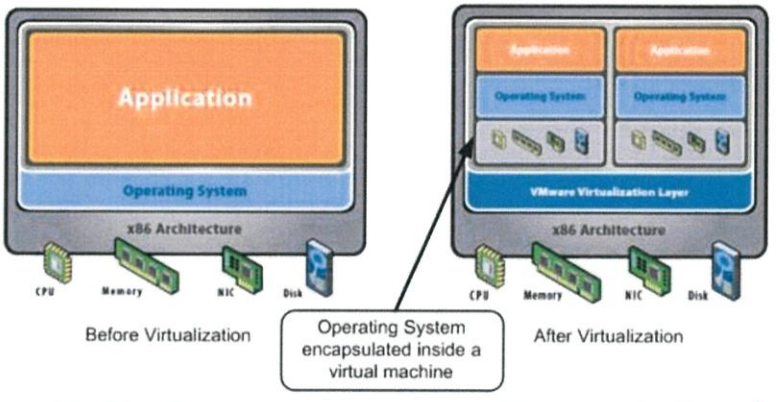
1. การค้นหาเพื่อตอบคำถามพื้นฐาน เช่น การค้นหาเครื่องเป้าหมายด้วย IP Address, Hostname, สถานะที่ต้องการ เป็นต้น หรือการระบุปริมาณการใช้งานทรัพยากรในเครื่อง การค้นหาว่าเครื่อง Virtual Machine เครื่องนั้นอยู่บน Physical Host ใด

2. การค้นหาเพื่อตอบคำถามที่ซับซ้อนยิ่งขึ้น เช่น อุปกรณ์มีการเชื่อมต่อกันอย่างไร การแจ้งเตือนอาจเกิดจากสาเหตุใดบ้าง อุปกรณ์ใดบ้างที่อาจทำให้เกิดปัญหาดังกล่าว เป็นต้น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 เทคโนโลยีเวอร์ชวลไลเซชั่น (Virtualization Technology)

เทคโนโลยีเวอร์ชวลไลเซชั่นเป็นเทคโนโลยีที่ใช้ในการสร้างระบบเสมือนบางอย่างขึ้นมา เช่น เครื่องคอมพิวเตอร์ ระบบปฏิบัติการ หน่วยเก็บข้อมูล หรือเครือข่ายในการเชื่อมต่อเป็นต้น โดยระบบเสมือนเหล่านี้ทำงานอยู่บนฮาร์ดแวร์เดียวกัน โดยใช้งานซอฟต์แวร์เวอร์ชวลไลเซชั่น (virtualization software) ในการจัดการทรัพยากรด้านฮาร์ดแวร์ [1] ดังในรูปที่ 2.1



รูปที่ 2.1 ภาพเปรียบเทียบข้อแตกต่างก่อนและหลังการใช้งานเทคโนโลยีเวอร์ชั่วไลเซชั่น

สาเหตุที่เทคโนโลยีเวอร์ชวลไลเซชั่นเข้ามามีบทบาทมากขึ้นเป็นผลมาจากหลายๆ เหตุผล เช่น

1) ทำให้ผู้ใช้งานหลายๆคนสามารถใช้งานเครื่องคอมพิวเตอร์ร่วมกันได้ซึ่งทำให้ค่าใช้จ่ายต่างๆ ทั้งในการจัดซื้อและบำรุงรักษาเครื่องลดลงอย่างมาก
2) สามารถจำลองฮาร์ดแวร์หรือระบบปฏิบัติการได้หลากหลายรูปแบบ ทำให้ซอฟต์แวร์ที่ต้องการใช้งานฮาร์ดแวร์หรือระบบปฏิบัติการเก่าๆ สามารถติดตั้งและทำงานได้
3) การบริหารจัดการต่างๆทำได้ง่ายขึ้นเนื่องจากการใช้งานของผู้ใช้ เช่น การติดตั้งแอพพลิเคชั่น (Application) หรือเครื่องไม่ถูกจำกัดและผูกกับฮาร์ดแวร์ที่ใช้งานอีกต่อไป ทำให้การขอใช้งานเครื่อง การย้ายทรัพยากร หรือการสำรองข้อมูลง่ายขึ้น

2.2 การประมวลผลแบบกลุ่มเมฆ (Cloud Computing)

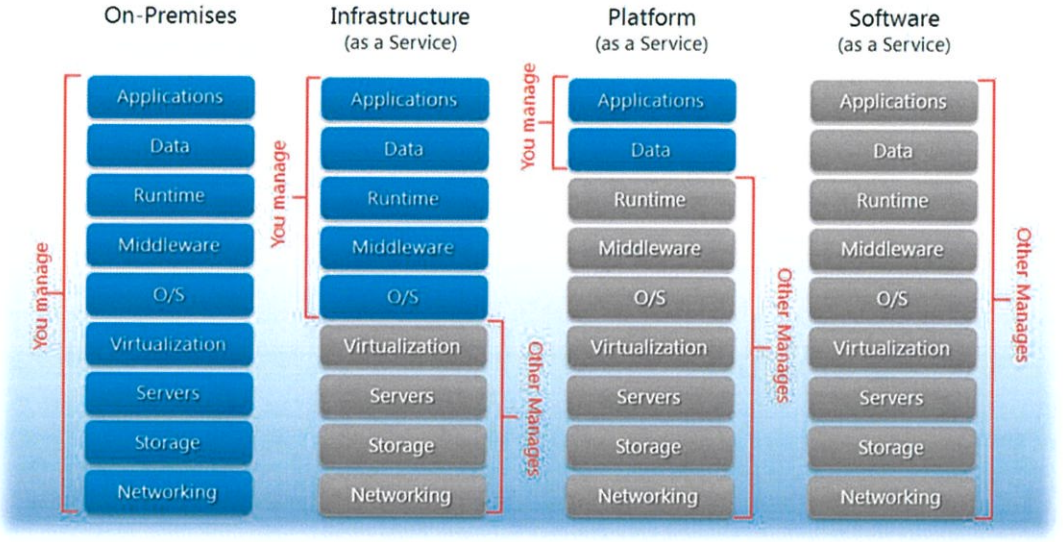
Cloud computing หรือการประมวลผลแบบกลุ่มเมฆ เป็นแนวคิดในการให้บริการคอมพิวเตอร์หรือบริการต่างๆ ให้กับผู้ใช้งานโดยผู้ให้บริการเป็นผู้รวบรวม ติดตั้ง และจัดการทรัพยากรต่างๆ เช่น เครือข่าย เครื่องแม่ข่าย หน่วยเก็บข้อมูล แอพพลิเคชั่น และนำทรัพยากรต่างๆ เหล่านี้ไปให้บริการกับผู้ใช้งาน ข้อดีของการที่ทรัพยากรต่างๆ ถูกรวมและจัดการโดยผู้ให้บริการเพียงฝ่ายเดียวนั้นคือ ผู้ใช้งานไม่ต้องลงทุนระบบขนาดใหญ่ตั้งแต่วันแรก แต่ใช้ทรัพยากรที่ผู้ให้บริการมีให้ ซึ่งเมื่อต้องการระบบที่ใหญ่ขึ้นก็สามารถเพิ่มได้ตามต้องการ โดยจะเสียค่าใช้จ่ายเพียงเท่าที่ใช้เท่านั้น (pay-as-you-go model)

ส่วนหนึ่งที่ทำให้แนวคิดในการสร้างการประมวลผลแบบกลุ่มเมฆเป็นจริงขึ้นมาได้นั้นเป็นผลมาจากการเกิดขึ้นของเทคโนโลยีเวอร์ชวลไลเซชั่นซึ่งทำให้ผู้ใช้หลายๆ คนสามารถแชร์เครื่องคอมพิวเตอร์แม่ข่ายกันได้โดยไม่กระทบด้านความปลอดภัย ทำให้ค่าใช้จ่ายในการซื้อและจัดการทรัพยากรทางด้านประมวลผลลดลง

การให้บริการต่างๆบนการประมวลผลแบบกลุ่มเมฆนั้นถูกแยกออกเป็น 3 ประเภทตามรูป 2.2 ดังนี้ [2]

1) Software as a Service หรือ SaaS คือการให้บริการแอพพลิเคชั่นแก่ผู้ใช้บริการ โดยที่แอพพลิเคชั่นนั้นทำงานอยู่บนโครงสร้างการประมวลผลแบบกลุ่มเมฆ (Cloud infrastructure) ของผู้ให้บริการ และผู้ใช้บริการสามารถเข้าถึงแอพพลิเคชั่นจากอุปกรณ์ต่างๆ เช่น เว็ปบราวเซอร์ (Web browser) หรือ ส่วนหน้าติดต่อของโปรแกรม (Program interface) ผู้ใช้บริการไม่ต้องดูและควบคุมโครงสร้างการประมวลผลแบบกลุ่มเมฆ ทั้งในส่วน network, เครื่องแม่ข่าย, ระบบปฏิบัติการ, storage เป็นต้น
2) Platform as a Service หรือ PaaS คือการที่ผู้ใช้บริการสามารถติดตั้ง (deploy) แอพพลิเคชั่นลงบนโครงสร้างการประมวลผลแบบกลุ่มเมฆ ผู้ใช้บริการสามารถสร้างหรือพัฒนาแอพพลิเคชั่นโดยใช้ภาษาโปรแกรมมิ่ง (Programming languages), libraries, บริการ และเครื่องมือต่างๆที่ผู้ให้บริการมีให้ โดยที่ผู้ใช้บริการไม่ต้องจัดการและดูแลส่วนของโครงสร้างการประมวลผมแบบกลุ่มเมฆ ทั้งในส่วนของ Network, เครื่องแม่ข่าย, ระบบปฏิบัติการ หรือ storage แต่จะต้องดูแลในส่วนของแอพพลิเคชั่นที่ติดตั้งและการตั้งค่าสภาพแวดล้อมเครื่อง host ต่างๆเพื่อให้เข้ากับการทำงานของแอพพลิเคชั่น
3) Infrastructure as a Service หรือ IaaS คือการที่ผู้ให้บริการจะต้องจัดหา process, storage, network และทรัพยากรพื้นฐานในการประมวลผลให้ โดยที่ผู้ใช้บริการสามารถติดตั้งและรัน software ต่างๆได้ซึ่งรวมถึงระบบปฏิบัติการและแอพพลิเคชั่น ผู้ใช้บริการไม่

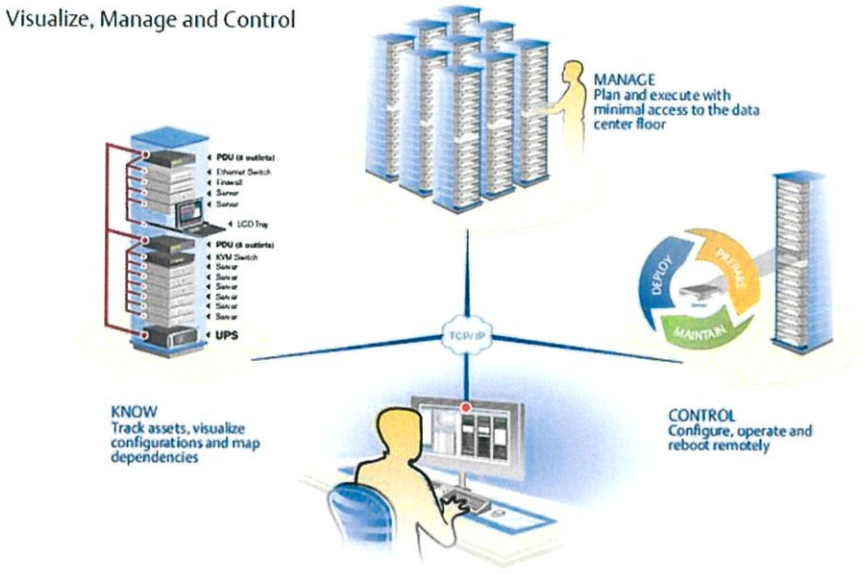
ต้องจัดการและดูแลส่วนโครงสร้างการประมวลผลแบบกลุ่มเมฆ แต่ต้องดูแลระบบปฏิบัติการ , storage และแอพพิเคชั่นที่ติดตั้งเอง และอาจจะสามารถจัดการได้อย่างจำกัดในส่วนของการเลือกองค์ประกอบของ network เช่น host firewall เป็นต้น



รูปที่ 2.2 ข้อแตกต่างของการให้บริการแบบต่างบน Cloud

2.3 การบริหารจัดการศูนย์ข้อมูล

ศูนย์ข้อมูลในที่นี้หมายถึงสถานที่ซึ่งเป็นที่ตั้งและเก็บรักษาเครื่องคอมพิวเตอร์รวมถึงอุปกรณ์ต่างๆ ที่เกี่ยวข้องอย่างอุปกรณ์เครือข่าย เช่น อุปกรณ์จัดเส้นทาง (Router) หรือ สวิตช์ ซึ่งโดยทั่วไปจะรวมถึงอุปกรณ์ต่างๆ ที่ใช้เพื่อรักษาให้เครื่องทำงานต่อไปได้ ตัวอย่างเช่น เครื่องสำรองไฟ อุปกรณ์ควบคุมความชื้นและอุณหภูมิ เป็นต้น โดยส่วนใหญ่ศูนย์ข้อมูลมักประกอบไปด้วยเครื่องแม่ข่าย ข้อมูลและอุปกรณ์ที่เกี่ยวข้องกับเครือข่ายจำนวนมาก โดยส่วนใหญ่จึงมีการเก็บข้อมูลการใช้งานและเหตุการณ์ต่างๆที่เกิดขึ้นภายในศูนย์ข้อมูลเพื่อใช้ประกอบการทำงานต่างๆ



รูปที่ 2.3 แนวทางการบริหารจัดการศูนย์ข้อมูล

เพื่อให้การจัดการเครื่องคอมพิวเตอร์และอุปกรณ์ต่างๆเป็นไปอย่างมีประสิทธิภาพการบริหารจัดการศูนย์ข้อมูลจึงต้องใช้งานเครื่องมือต่างๆ เพื่อเข้ามาช่วยในการทำงานอย่างการจัดเตรียมทรัพยากร (Provisioning), การตั้งค่า (Configuration), การควบคุมการทำงานให้สอดคล้องกัน (Orchestration/Automation) และการสอดส่องดูแลอุปกรณ์ต่างๆ (Monitoring) ดังในรูปที่ 2.3

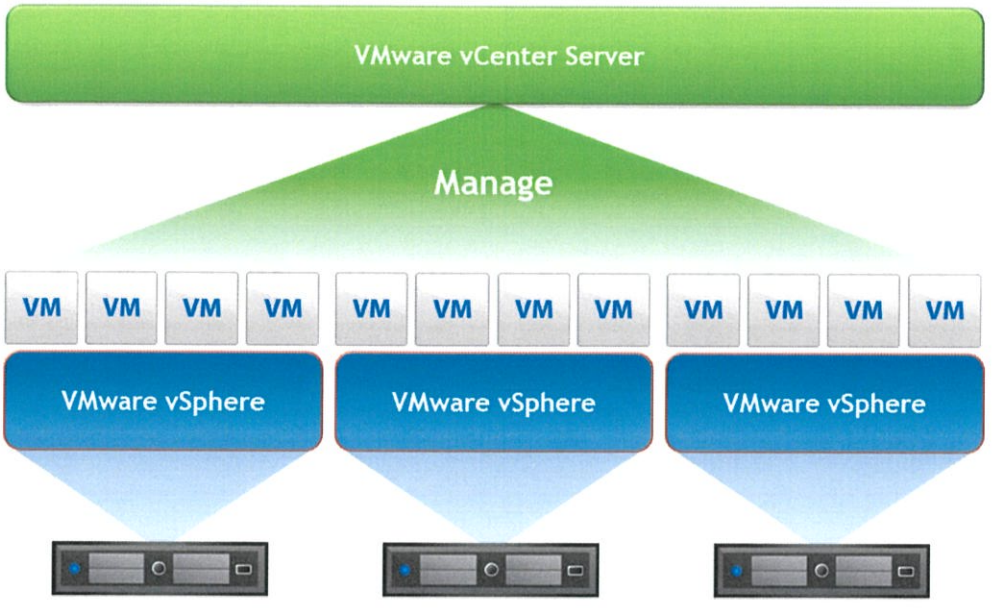
2.3.1 เครื่องมือที่ใช้ในการบริหารจัดการศูนย์ข้อมูล

เครื่องมือที่จะกล่าวถึงคือ VMware vCenter และ Nagios ซึ่งแม้ว่ารูปแบบการทำงานของระบบที่ผู้พัฒนาจะจัดทำขึ้นจะมีจุดประสงค์ใกล้เคียงกับซอฟต์แวร์ หรือ เครื่องมือทั้งสองที่จะกล่าวถึงหรือเครื่องมืออื่นๆที่ใช้เพื่อบริหารจัดการศูนย์ข้อมูล แต่ยังคงมีข้อแตกต่างหรือข้อเปรียบเทียบสำคัญหลายอย่างเนื่องจากเครื่องมือส่วนใหญ่มักพัฒนาขึ้นเพื่อใช้งานกับผลิตภัณฑ์ใดผลิตภัณฑ์หนึ่งเท่านั้น หรือในหลายๆ ผลิตภัณฑ์ที่พยายามรวมข้อมูลจากหลายๆเครื่องมือ กลับไม่สามารถดึงข้อมูลเหล่านั้นมาแสดงได้อย่างมีประสิทธิภาพหรือทำให้ผู้ใช้ไม่สามารถเห็นภาพรวมของระบบได้ดีนัก

2.3.1.1 VMware vCenter

VMware vCenter เป็นหนึ่งในเครื่องมือที่ใช้ในการบริหารจัดการศูนย์ข้อมูล โดยใช้ในงานที่เกี่ยวข้องกับการบริหารจัดการเครื่องคอมพิวเตอร์เสมือน (Virtual Machine) โดยมีรูปแบบการติดตั้งและใช้งานเป็นแบบรวมศูนย์ (Centralized Management) โดยสามารถแบ่งส่วนการติดตั้งและใช้งานออกเป็น 3 ส่วนหลักๆ [3] คือ

1) VMware vCenter Server : หมายถึงเครื่องที่ติดตั้ง VMware vCenter ซึ่งจะมีการตั้งค่าการเชื่อมต่อไปยังเครื่องอื่นที่ติดตั้ง VMware vSphere เพื่อให้สามารถควบคุมและดูแลเครื่องเหล่านั้นได้จากระยะไกล
2) VMware vSphere : หมายถึงเครื่องที่ติดตั้ง Virtual Machine Monitor (VMM) หรือ Hypervisor ซึ่งมีความสามารถในการจัดการกับเครื่องคอมพิวเตอร์เสมือนได้
3) Virtual Machine : หมายถึงเครื่องคอมพิวเตอร์เสมือนซึ่งติดตั้งและควบคุมโดย VMM



รูปที่ 2.4 การบริหารจัดการเครื่องคอมพิวเตอร์เสมือนโดย VMware vCenter

2.3.1.2 Nagios

Nagios เป็นหนึ่งในเครื่องมือที่ใช้ในการบริหารจัดการศูนย์ข้อมูล โดยใช้ในงานที่เกี่ยวข้องกับการตรวจสอบสถานะของอุปกรณ์เครือข่าย สถานะของเครื่องคอมพิวเตอร์และบริการต่างๆ โดยใช้งานโปรโตคอลต่างๆ เช่น SMTP, POP3, HTTP, NNTP, ICMP, SNMP, FTP หรือ SSH เป็นต้น [4]

2.4 ฐานข้อมูลแบบ NoSQL

NoSQL database ถูกนิยามว่าเป็น “non-relational, distributed, open-source and horizontal scalable” ซึ่ง non-relational หมายความว่าฐานข้อมูลนี้ไม่ใช่ relational database ไม่มีความสัมพันธ์ (relation) หรือ ตาราง ทำให้หลักการแตกต่างไปอย่างสิ้นเชิง Distributed คือการที่ข้อมูลถูกเก็บและจัดการด้วยเครื่องต่างกัน ซึ่งอาจมี data replication ได้ Open source หมายความว่าทุกคนสามารถดูโค้ดต้นฉบับได้ฟรี ปรับเปลี่ยนและคอมไพล์ได้ด้วยตนเอง

Horizontally scalable หมายความว่ายิ่งเราเพิ่ม database server มากขึ้นเท่าไหร่ ประสิทธิภาพก็จะเพิ่มขึ้นแทบจะเป็นการเพิ่มแบบเส้นตรง ซึ่งเราเรียกอีกอย่างว่า out scaling

NoSQL นั้นไม่มีโครงตาราง (schema free) หรือมีโครงตารางที่ไม่ถูกบังคับมากนัก ไม่มีส่วนที่ใช้ข้อมูลร่วมกัน (โหนด (node) นั้นเป็นอิสระต่อกัน) ซึ่งรูปแบบของความถูกต้อง (consistency model) ก็จะไม่ใช่ ACID (atomicity, consistency, isolation, durability) ประเภทของ NoSQL database จะแบ่งได้เป็น 4 ประเภทหลักๆ คือ

1. Key/Value databases เป็น NoSQL database ที่มีเพียง key และ value รวมอยู่ด้วยกันเป็นชุดข้อมูล ซึ่ง key ควรจะไม่ซ้ำกัน value นั้นสามารถเป็นได้หลายรูปแบบ เช่น STRING, INTEGERS, FLOATS หรือ BYTE ARRAY เป็นต้น บางฐานข้อมูลไม่ได้สนใจรูปแบบของ value โดยจะทำหน้าที่เพียงเก็บสิ่งที่ผู้ใช้ป้อนเข้ามาโดยไม่ตีความ

2. Document store จะใช้เอกสาร (document) ทั้งเอกสารในรูปแบบต่างๆเป็นชุดข้อมูล ตัวอย่างของรูปแบบเอกสาร เช่น XML (EXtensible Markup Language) หรือ JSON (JavaScript Object Notation) เป็นต้น ซึ่ง document database นี้อาจจะมองว่าเป็นกรณีพิเศษของ Key/Value databases ก็ว่าได้ แต่ฐานข้อมูลจำเป็นต้องรู้รูปแบบของเอกสารที่เก็บซึ่งหมายความว่า document database จำเป็นต้องตีความ value ที่เข้ามาซึ่งส่วนนี้ทำให้แตกต่างกับ Key/Value database

3. Column-oriented databases แนวคิดของ Column-oriented databases คือ จะไม่ใช่การเก็บแบบ ชุดข้อมูล และ attributes ของข้อมูล เก็บเป็น 1 หน่วย (unit) (เก็บเป็นแถว) เหมือนใน SQL database แต่จะเก็บหนึ่ง attribute ของชุดข้อมูลเป็น 1 หน่วย (เก็บเป็นหลัก) แทน

4. Graph databases นั้นมาจากทฤษฎีกราฟ (graph theory) ที่มี โหนดหรือจุด (vertices) และมี เส้นขอบ (edges) เชื่อมโหนด ซึ่งในฐานข้อมูลโหนดคือ entities เช่น คน เป็นต้น และ เส้นขอบ คือ ความสัมพันธ์ระหว่างแต่ละ entities ตัวอย่างเช่น “เฟรด” – “รู้จัก” – “ฮอลลี่” [5]

2.4.1 Neo4j

Neo4j เป็นฐานข้อมูลแบบ NoSQL ที่เลือกใช้โครงสร้างข้อมูลแบบ graph และพัฒนาขึ้นด้วยภาษา Java โดยเปิดให้ใช้งานและแก้ไขในลักษณะของ open-source โดยมีจุดเด่นอยู่ที่รองรับการใช้งาน transaction และรับประกันคุณสมบัติ ACID (Atomicity, Consistency, Isolation และ Durability) [6]

โครงสร้างข้อมูลของ Neo4j เป็นแบบ graph จึงทำให้การเข้าถึงข้อมูลและการค้นหาเส้นทางจากความสัมพันธ์ต่างๆ เป็นไปได้ง่ายและรวดเร็วกว่าฐานข้อมูลแบบอื่นๆ [7] โดยโครงสร้างข้อมูลของ Neo4j มีรูปแบบที่เรียกว่า Property Graph ซึ่งมีแบ่งคุณสมบัติและความสำคัญของข้อมูลออกเป็น 4 รูปแบบดังนี้ [8]

1) Node : หมายถึงข้อมูลหลักที่เราสนใจเปรียบเสมือนเป็น document ที่เก็บ property หลายๆ อันไว้ด้วยกัน โดย node สามารถมี label หรือไม่มีก็ได้
2) Relationship : เป็นส่วนที่อธิบายความสัมพันธ์หรือความเชื่อมโยงระหว่าง 2 nodes โดยจะมีทิศทางเดียวและมี label กำกับเสมอ
3) Property : ประกอบด้วยชื่อชนิดข้อมูล (key) และข้อมูลนั้น (value) โดย key นั้นเป็นข้อมูลแบบ string ขณะที่ value มีรูปแบบของข้อมูลไม่แน่นอน (ดูเพิ่มเติมได้จากเอกสารของ Neo4j [9])
4) Label : เปรียบเสมือนป้ายชื่อซึ่งบอกชนิดของข้อมูลหรือความสัมพันธ์ในกราฟ

โดย Neo4j มีภาษาที่ใช้ในการจัดการข้อมูลเป็นของตนเองคือ Cypher query โดยสามารถศึกษาเพิ่มเติมได้จากเว็บไซต์ของ Neo4j [10]

neo4j นั้นสามารถรองรับ node ได้มากที่สุดประมาณ 34 พันล้าน nodes, 34 พันล้าน relationships, 68 พันล้าน properties เนื่องจากเป็นข้อจำกัดของ address space [11] และในส่วนของประสิทธิภาพในการ query จะขึ้นอยู่กับปัจจัยต่างๆ ทั้ง test environment, ชุดข้อมูล รวมไปถึงความซับซ้อนและวิธีการ query ดังที่ได้สรุปไว้ในเอกสารอ้างอิง ซึ่งจะนำมาใช้อ้างอิงในการพัฒนาระบบต่อไป [12]

การนำ Graph Database มาใช้ในการจัดการเกี่ยวกับความสัมพันธ์ของอุปกรณ์ต่างๆภายในศูนย์ข้อมูลนั้นเคยถูกนำมาใช้ในงานงานวิจัย [13] โดยถูกนำมาใช้ในการแสดงผลเป็นแผนที่แสดงความเชื่อมโยงของเครื่องต่างๆในการจัดการเครื่องต่างๆใน vCenter ซึ่งแม้จะมีรูปแบบในการจำลองข้อมูลคล้ายกันแต่งานวิจัยดังกล่าวมีจุดประสงค์ที่จะใช้เพื่อตอบคำถามเชิงการบริหารจัดการความเสี่ยง (Risk Management) ซึ่งเป็นเพียงส่วนหนึ่งของปัญหาที่ต้องการแก้ในงานชิ้นนี้ นอกจากนี้ในงานวิจัยดังกล่าวยังเน้นวิธีการแก้ปัญหาต่างๆไปในเชิงการค้นหา มากกว่าการแสดงผลข้อมูลซึ่งเป็นส่วนสำคัญในการสื่อสารระหว่างผู้ใช้กับระบบที่มีข้อมูลจำนวนมาก

2.4.2 MongoDB

MongoDB เป็นฐานข้อมูลแบบ MongoDB เก็บข้อมูลในรูปแบบ document ซึ่งจะเป็นลักษณะคล้ายกับ JSON ที่จะมี key และ value เป็นคู่ ซึ่ง document นั้นมีความคล้ายคลึงกันกับโครงสร้างในภาษาโปรแกรมมิ่งที่เกี่ยวข้องกับ key และ value เช่น hashes, maps หรือ arrays) โดยทางการแล้ว MongoDB document นั้นจะเป็น BSON document ซึ่ง BSON คือไบนารี่ที่เป็นตัวแทนของ JSON พร้อมกับข้อมูลรูปแบบเพิ่มเติม ใน document value ของ field จะเป็นรูปแบบข้อมูล BSON อะไรก็ได้ ทั้ง document อื่นๆ, arrays และ arrays ของ document ตัวอย่างของ field และ value เป็นดังรูปที่ 2.5

รูปที่ 2.5 แสดงตัวอย่าง field และ value

MongoDB จะเก็บ document ทั้งหมดเป็น collection ซึ่ง collection คือกลุ่มของ document ที่เกี่ยวข้องกันที่มีชุดของ common indexes ร่วมกัน collection จะคล้ายกับตารางใน relational databases [14]

2.5 Distributed system

คือการที่องค์ประกอบฮาร์ดแวร์หรือซอฟต์แวร์ติดต่อและทำงานร่วมกันโดยการส่งข้อความเท่านั้น (passing messages) ซึ่งคุณลักษณะของ distributed system ที่จะได้ตามมาจะเป็นดังนี้

1. Concurrency คือ ใน network ของคอมพิวเตอร์การรัน program แบบ concurrent นั้นเป็นเรื่องปกติ คือเป็นลักษณะ ฉันทำงานของฉันที่คอมพิวเตอร์ของฉัน เธอทำงานของเธอบนคอมพิวเตอร์ของเธอ โดยจะใช้ทรัพยากรบางอย่างร่วมกัน เช่น เว็บเพจ (web page) หรือ ไฟล์ เมื่อจำเป็น

2. No global clock เมื่อโปรแกรมต้องการจะทำงานร่วมกัน โปรแกรมทั้งสองจะติดต่อกันผ่านการแลกเปลี่ยนข้อความ (exchange message) การทำงานร่วมกันอย่างใกล้เคียงกันจะขึ้นอยู่กับเวลาที่แชร์ร่วมกันจากการทำงานของโปรแกรม ซึ่งความแม่นยำยังคงถูกจำกัดด้วยการประสานเวลา (synchronize) ของคอมพิวเตอร์ภายใน network

3. Independent failures แต่ละส่วนของระบบสามารถล้มเหลวได้โดยไม่กระทบกับส่วนอื่นๆ ซึ่งส่วนอื่นจะยังคงทำงานต่อไปได้ เช่น ความผิดพลาดของ network ส่งผลให้คอมพิวเตอร์เครื่องหนึ่งเกิดการแยกออกจากกลุ่มคอมพิวเตอร์เครื่องอื่นๆที่เชื่อมต่อ network เดียวกัน แต่ไม่ได้หมายความว่าคอมพิวเตอร์เครื่องนั้นจะหยุดทำงาน ซึ่งโปรแกรมอาจจะเพียงตรวจหา network ไม่เจอ หรือ อยู่ๆความเร็วของ network ก็ลดลง ซึ่งคอมพิวเตอร์เครื่องอื่นๆก็ยังคงทำงานได้อย่างปกติเป็นต้น [15]

ข้อดีของระบบในลักษณะนี้คือลดความเสี่ยงที่อาจจะทำให้ระบบล้มเหลวเมื่อส่วนใดส่วนหนึ่งของระบบผิดพลาดและเนื่องจากแต่ละส่วนของระบบไม่ขึ้นต่อกันทำให้การเพิ่มหรือลดแต่ละส่วนใน

ระบบเพื่อให้สอดคล้องกับจำนวนงานที่ได้รับเป็นไปได้แต่ถ้าระบบมีขนาดใหญ่อาจจะยากต่อการจัดการ [16]

2.5.1 Message-Oriented Middleware

Message-Oriented Middleware หรือ MOM หมายถึงซอฟต์แวร์หรือฮาร์ดแวร์ที่มีหน้าที่ในการรับ พักข้อมูล (buffered) และส่งข้อมูล ซึ่งถูกนำมาใช้งานใน distributed system เนื่องจากการแลกเปลี่ยนข้อมูลระหว่างเครื่องจำเป็นต้องใช้การส่งผ่านข้อมูล [17]

ข้อดีของการใช้งาน MOM คือช่วยให้ระบบทำงานแบบ asynchronous ทำให้เมื่อมีข้อมูลเข้ามาในระบบจำนวนมาก MOM จะเป็นส่วนที่ช่วยพักข้อมูลดังกล่าวเพื่อให้เครื่องที่จะนำข้อมูลเหล่านี้ไปประมวลผลสามารถทำงานได้ทัน ในอีกทางหนึ่งเมื่อเกิดข้อผิดพลาดขึ้นที่เครื่องใดเครื่องหนึ่งที่นำข้อมูลไปประมวลผล MOM ซึ่งสำรองข้อมูลเหล่านี้ไว้สามารถส่งต่อข้อมูลนี้ให้เครื่องอื่นๆนำไปประมวลผลใหม่ได้ นอกจากนี้ยังสามารถส่งต่อข้อมูลไปยังเครื่องที่กำหนด (routing) หรือเปลี่ยนรูปแบบของข้อมูล (transformation) ได้

ข้อเสียของการใช้งาน MOM คืออาจทำให้เกิดผลกระทบกับประสิทธิภาพของระบบเนื่องจากความล่าช้าในการส่งข้อมูลได้

2.5.1.1 RabbitMQ

RabbitMQ เป็น Message-Oriented Software ที่ให้ใช้งานแบบไม่เสียค่าใช้จ่าย (freeware) ซึ่งใช้งาน Advanced Message Queuing Protocol หรือ AMQP ซึ่งเป็นมาตรฐานหนึ่งที่กำหนดโปรโตคอลที่จะใช้งานใน MOM [16]

RabbitMQ ถูกเขียนขึ้นด้วยภาษา Erlang และสร้างขึ้นตามรูปแบบของ Open Telecom Platform เพื่อให้สามารถทำงานในลักษณะของกลุ่ม (cluster) และสามารถทำงานต่อได้เมื่อเกิดปัญหา (failover) นอกจากนี้ยังมี library พื้นฐานที่ใช้ในการติดต่อหลายภาษาและส่วนเสริมสำหรับการบริหารจัดการผ่านหน้าเว็บจึงทำให้ง่ายแก่การพัฒนาและใช้งาน

2.6 Data visualization

Data visualization เป็นขั้นตอนหนึ่งในการประมวลผลข้อมูลและนำเสนอต่อผู้ใช้งาน โดยใช้วิธีการแปลงข้อมูลให้อยู่ในรูปของการแสดงผล เช่น จุด, เส้น หรือกราฟแท่ง เป็นต้น เป้าหมายหลักของการทำ data visualization คือการนำเสนอข้อมูลให้อยู่ในรูปที่ชัดเจนและมีประสิทธิภาพสำหรับผู้ใช้ [18] เพื่อให้ได้ผลตามที่กล่าวมาการนำเสนอข้อมูลที่ดีนั้นจึงต้องคำนึงถึงทั้งความสวยงาม และประสิทธิภาพในการใช้งานร่วมกัน โดยนำเสนอข้อมูลในรูปแบบที่นำเสนอข้อมูลได้ชัดเจนและสามารถ

นำไปใช้งานได้ขณะเดียวกันก็ยังต้องสามารถนำเสนอส่วนสำคัญของข้อมูลได้เช่นกัน ซึ่งในปัจจุบันการวิจัยและพัฒนาเกี่ยวกับ Data visualization ยังคงมีขึ้นอย่างต่อเนื่อง

บทที่ 3
การวิเคราะห์และออกแบบระบบ

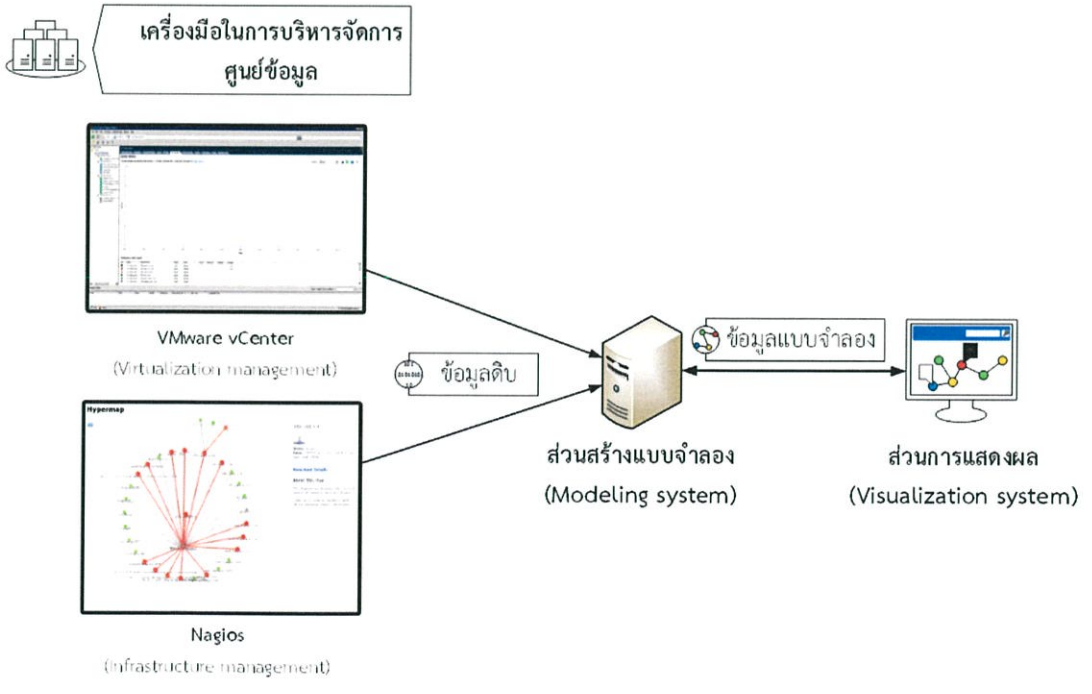
3.1 ภาพรวมของระบบ

ในการจัดการศูนย์ข้อมูลนั้น ผู้ดูแลมักจะต้องดูข้อมูลหลายๆอย่าง เช่น ร้อยละของการใช้งาน ปริมาณข้อมูลที่รับ-ส่งผ่านเครือข่าย หรือสถานะของเครื่องในศูนย์ข้อมูล เป็นต้น เพื่อใช้ในการตัดสินใจในการบริหารจัดการทรัพยากรให้มีประสิทธิภาพหรือแก้ปัญหาต่างๆ ที่เกิดขึ้น ข้อมูลเหล่านี้มักถูกเก็บรวบรวมมาจากเครื่องมือต่างๆ ซึ่งใช้ช่วยในการบริหารจัดการศูนย์ข้อมูล เช่น VMware vCenter หรือ Nagios เป็นต้น

ระบบที่นำเสนอถูกแบ่งการทำงานออกเป็นสองส่วนหลักๆดังรูปที่ 3.1 คือ

1) ส่วนสร้างแบบจำลองซึ่งทำหน้าที่ในการรวบรวมข้อมูล ประมวลผล จัดรูปแบบข้อมูลต่างๆ ที่ได้รับให้ง่ายต่อการนำไปใช้งาน

2) ส่วนแสดงผลข้อมูลต่อผู้ใช้งานซึ่งใช้เพื่อให้การเข้าถึงข้อมูลที่ระบบจัดเก็บไว้ทำได้ง่ายขึ้น



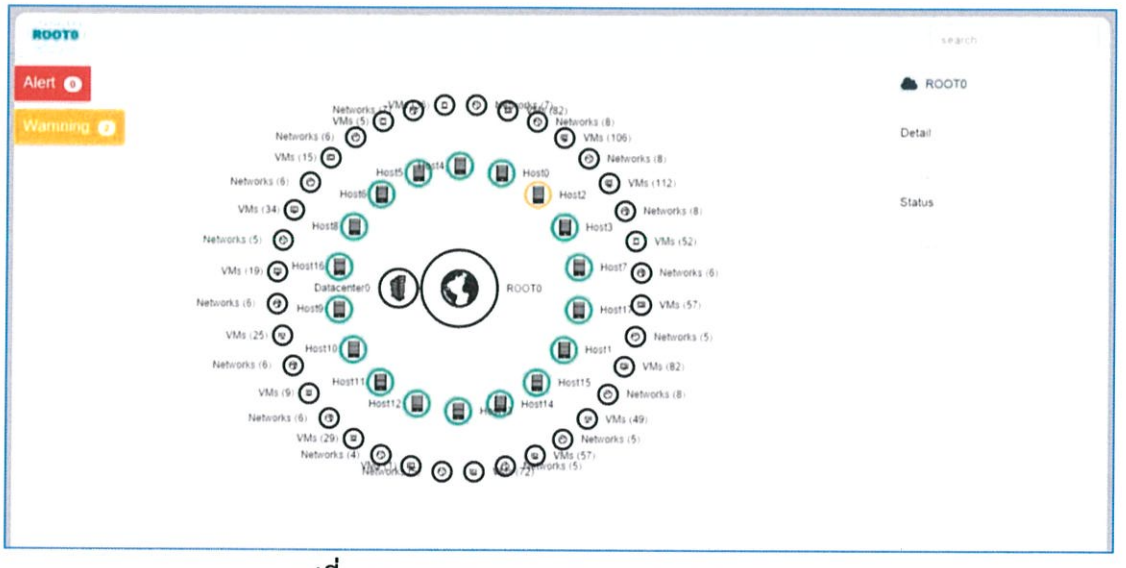
รูปที่ 3.1 ภาพแสดงการติดต่อกันระหว่างสองส่วนของระบบ

3.1.1 ส่วนสร้างแบบจำลอง

เป็นส่วนที่ทำหน้าที่รับข้อมูลจากเครื่องมือหลายๆ ชนิดเพื่อรวบรวมข้อมูลและหาความสัมพันธ์และความเชื่อมโยงของข้อมูลเหล่านั้นและนำไปสร้างแบบจำลองข้อมูล และเพื่อให้ระบบสามารถรองรับการเชื่อมต่อจากเครื่องมือหลายๆ ชนิดได้จึงออกแบบให้ผู้ใช้สามารถเขียนตัวแปลงข้อมูลของเครื่องมือนั้นๆ (Adapter) เพื่อแปลงข้อมูลดังกล่าวให้อยู่ในรูปแบบพื้นฐานที่ระบบสามารถประมวลผลได้ (ดูเพิ่มเติมในหัวข้อแบบจำลองข้อมูล) และติดต่อมายังส่วนสร้างแบบจำลองเพื่อนำไปประมวลผลและสร้างแบบจำลองข้อมูลต่อไป

3.1.2 ส่วนการแสดงผล

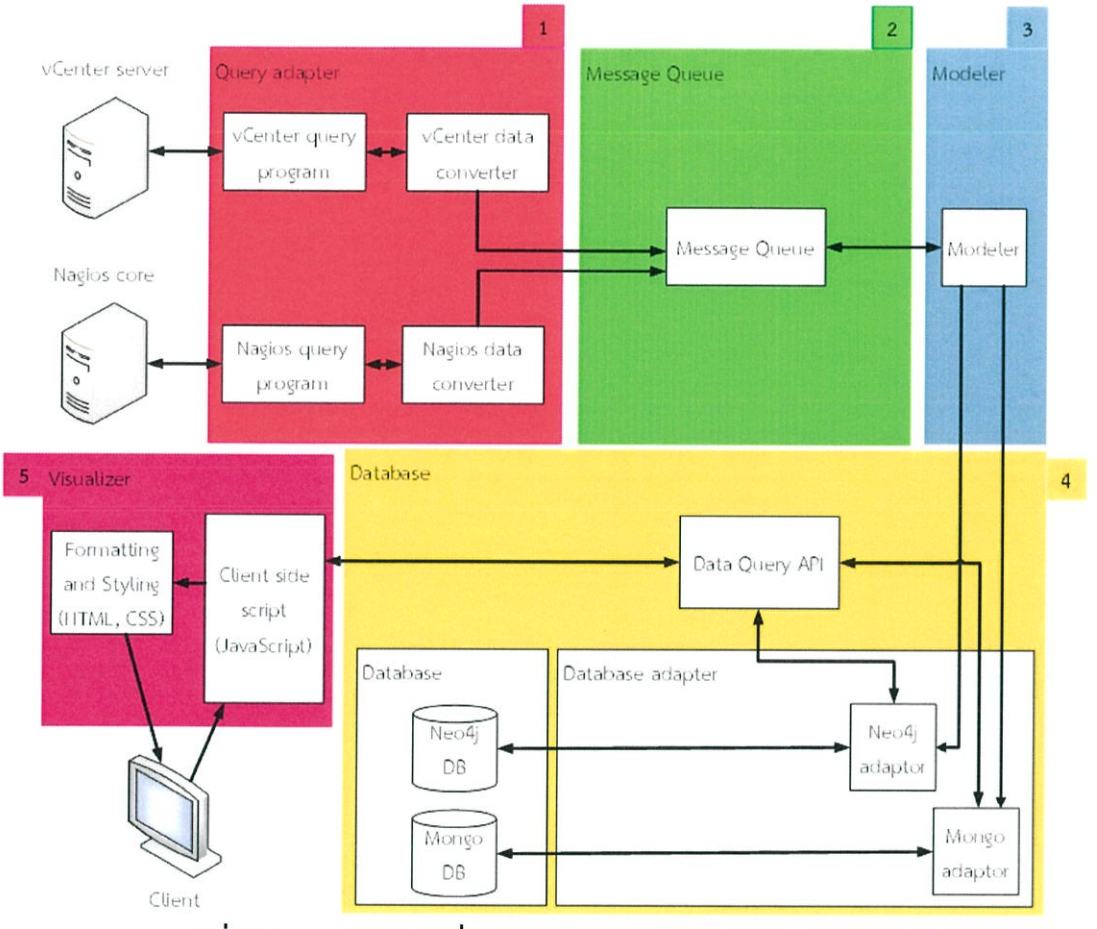
ในการพัฒนามุ่งเน้นที่จะให้ผู้ใช้สามารถใช้งานระบบและแสดงผลให้เข้าใจง่าย โดยไม่มีข้อจำกัดของเครื่องที่ผู้ใช้จะใช้แสดงผล จึงเลือกที่จะแสดงผลผ่านทางหน้าเว็บ โดยใช้งาน HTML และ CSS ในการแสดงผล และ JavaScript ซึ่งจะทำหน้าที่ในการติดต่อกับผู้ใช้ และเมื่อมีการเปลี่ยนแปลงต่างๆ เกิดขึ้นในระบบหรือผู้ใช้ต้องการดูข้อมูลในส่วนที่ไม่เคยขอข้อมูลมาก่อนจะทำหน้าติดต่อไปยัง Data Query API เพื่อร้องขอข้อมูลจากระบบ ซึ่งข้อมูลดังกล่าวจะถูกนำมาแสดงผลให้กับผู้ใช้ต่อไป วิธีนี้จะช่วยลดการส่งข้อมูลระหว่างผู้ใช้กับระบบ ทำให้การแสดงผลข้อมูลมีการตอบโต้กับผู้ใช้ได้ดีขึ้น โดยมีตัวอย่างการแสดงผลเบื้องต้นดังรูปที่ 3.2



รูปที่ 3.2 ภาพการแสดงผลของ DataVis.

3.2 โครงสร้างในการพัฒนาระบบ

ระบบที่นำเสนอมีโครงสร้างดังต่อไปนี้



รูปที่ 3.3 แผนภาพการเชื่อมต่อระหว่างส่วนย่อยของระบบ

3.2.1 ส่วนสร้างแบบจำลอง

3.2.1.1 Query adapter (รูปที่ 3.3 ส่วนที่ 1)

Query adapter ทำหน้าที่เชื่อมต่อและรวบรวมข้อมูลต่างๆ จากเครื่องมือที่ใช้ภายในศูนย์ข้อมูลโดยเป็นโปรแกรมที่ทำหน้าที่ติดต่อกับ API หรือฐานข้อมูลของเครื่องมือต่างๆ โดยโครงงานนี้จะออกแบบให้รองรับการดึงข้อมูลจาก vCenter และ Nagios โดยแยกการทำงานของ Query adapter ออกเป็นสองส่วนย่อยๆ คือ query program ซึ่งทำหน้าที่ติดต่อดึงข้อมูลจากเครื่องมือ และ data converter ซึ่งทำหน้าที่แปลงข้อมูลให้อยู่ในรูปแบบพื้นฐานที่ระบบสามารถประมวลผลได้ ข้อมูลเหล่านี้จะถูกส่งไปยัง Message Queue ต่อไป

ในส่วนนี้ออกแบบให้ผู้ใช้งานสามารถเขียน Query adapter เพิ่มเติมได้ในอนาคตเพื่อให้รองรับการเชื่อมต่อกับเครื่องมือใหม่ๆ กับระบบ

3.2.1.2 Message Queue (รูปที่ 3.3 ส่วนที่ 2)

ทำหน้าที่เป็นที่พักข้อมูลชั่วคราวก่อนที่จะถูกนำไปประมวลผลต่อที่ modeler โดย Message queue จะกระจายงานให้กับ modeler แต่ละตัว โดยในโครงงานนี้ได้เลือกใช้งาน RabbitMQ เนื่องจากรองรับการใช้งาน AMQP (Advanced Message Queuing Protocol)นอกจากนี้ยังมีเอกสารเกี่ยวกับการทดสอบการใช้งาน [19] message queue ผลการทดลองแสดงให้เห็นว่า RabbitMQ สามารถทำงานกับข้อมูลขนาดเล็กถึงปานกลางได้ดีกว่า message queue สำเร็จรูปตัวอื่นๆ เช่น HornetQ 2.3.0 หรือ QPID 0.20 ถึง 3 เท่า

3.2.1.3 Modeler (รูปที่ 3.3 ส่วนที่ 3)

Modeler เป็นส่วนที่ทำหน้าที่หลักในระบบโดยประมวลผลสร้างความสัมพันธ์ต่างๆจากข้อมูลที่มีอยู่ในฐานข้อมูลร่วมกับข้อมูลที่ได้รับจากการดึงข้อมูลจาก Message Queue เพื่อสร้างและปรับปรุงแบบจำลองของระบบ (รายละเอียดของแบบจำลองข้อมูลสามารถดูเพิ่มเติมในหัวข้อที่ 3.3 แบบจำลองข้อมูล)

3.2.1.4 Database service (รูปที่ 3.3 ส่วนที่ 4)

ทำหน้าที่บันทึกแบบจำลองข้อมูลจาก modeler และติดต่อกับส่วน Data Query API ที่ทำหน้าที่ในการแปลงข้อมูล HTTP จาก Visualizer ให้เป็น Query ที่ใช้ติดต่อกับ Database

ในโครงงานนี้ได้เลือกใช้ Database 2 ตัวดังนี้

1. Neo4j เป็นส่วนที่เก็บข้อมูลในรูปของ Graph Database ซึ่งใช้จำลองโครงสร้างและข้อมูลพื้นฐาน (node) และความสัมพันธ์ระหว่างข้อมูล (edge)
2. MongoDB เป็นส่วนที่เก็บข้อมูลสถานะและการตั้งค่าต่างๆ ของแบบจำลอง เช่น configuration, performance เป็นต้น โดยใช้งาน Document Database

3.2.2 ส่วนการแสดงผล

3.2.2.1 Visualizer (รูปที่ 3.3 ส่วนที่ 5)

ทำหน้าที่แปลงข้อมูลในรูปแบบ JSON ให้เป็นภาพและรับคำสั่งการทำงานต่างๆ ของผู้ใช้เพื่อติดต่อขอข้อมูลจาก Data Query API

3.3 แบบจำลองข้อมูล

แบบจำลองข้อมูลจะทำหน้าที่ในการรวมข้อมูลหลากหลายรูปแบบจากหลายๆ เครื่องมือให้มาอยู่ในรูปเดียวกัน ทางผู้พัฒนาจึงได้ออกแบบแบบจำลองข้อมูลที่ยืดหยุ่น สามารถรองรับข้อมูลจากหลายเครื่องมือพร้อมๆ กันได้และสามารถเชื่อมต่อกับเครื่องมือจัดการศูนย์ข้อมูลอื่นๆ ในอนาคตได้ นอกจากนี้เพื่อให้การค้นหาความสัมพันธ์ระหว่างข้อมูล และการค้นหาข้อมูลเป็นไปได้ง่ายและใช้เวลาน้อยลงจึงออกแบบแบบจำลองข้อมูลโดยใช้องค์ประกอบพื้นฐานและลักษณะเฉพาะของวัตถุเหล่านั้น ในการจัดหมวดวัตถุในศูนย์ข้อมูล ผู้พัฒนาได้จำแนกวัตถุออกเป็น 5 ประเภทหลักตามตารางที่ 3.1 และวัตถุแต่ละประเภทมีสัญลักษณ์ตามที่กำหนดในข้อ 3.3.2

3.3.1 ชนิดของวัตถุในแบบจำลอง

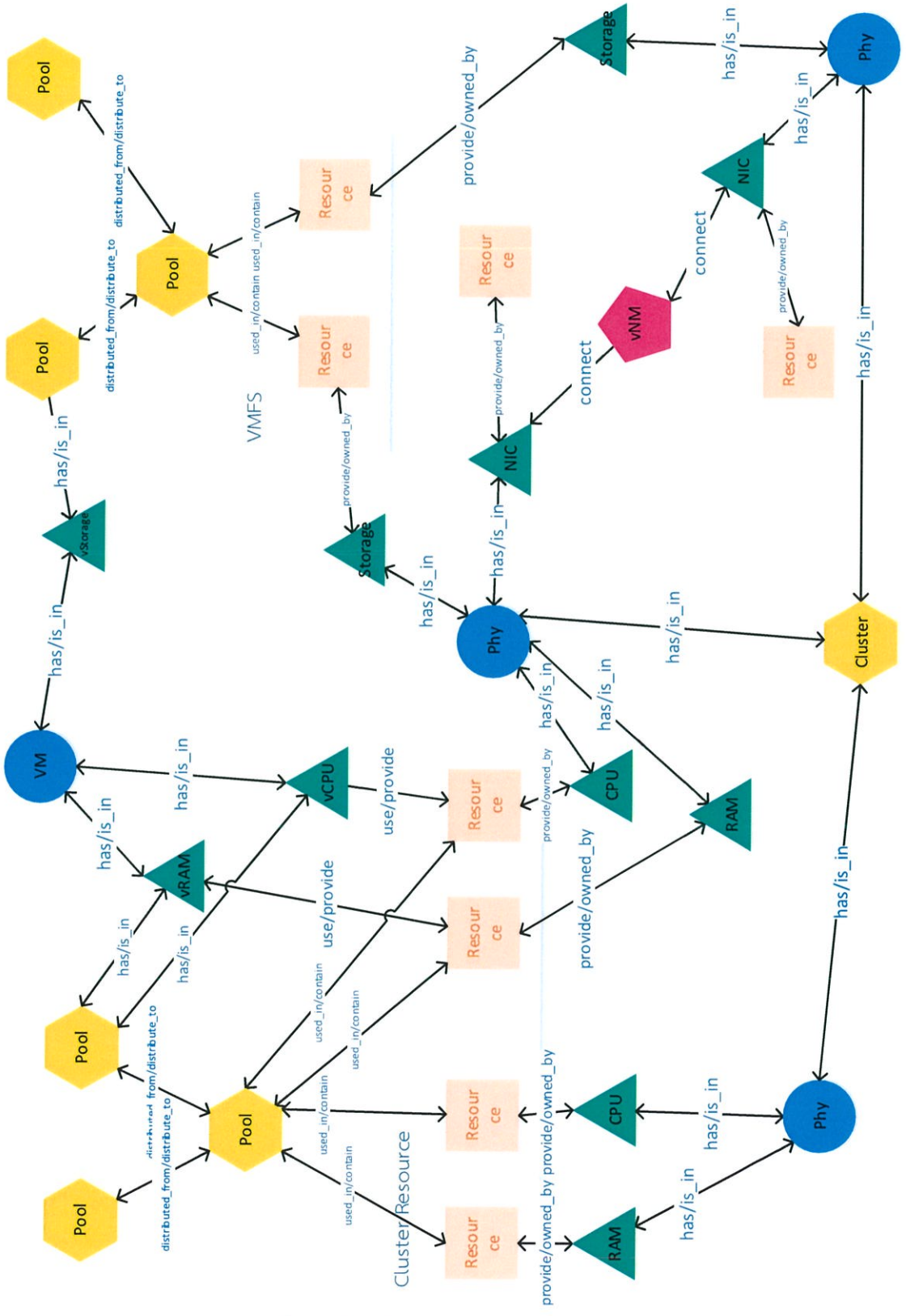
ตารางที่ 3.1 แสดงรายละเอียดชนิดของวัตถุในแบบจำลอง

ชื่อชนิด	ลักษณะเฉพาะ	ตัวอย่างโครงสร้างจริงในศูนย์ข้อมูล
Compute unit	เป็นรูปแบบข้อมูลซึ่งจำลองการเป็นหน่วยประมวลผลในศูนย์ข้อมูล	เครื่องคอมพิวเตอร์อย่างเครื่องแม่ข่าย, เครื่องเสมือน (Virtual machine)
Device	เป็นรูปแบบข้อมูลซึ่งจำลองอุปกรณ์ต่างๆ ในศูนย์ข้อมูล	Storage Server, Storage, Network Interface Card (NIC), CPU, RAM หรืออุปกรณ์ซึ่งให้บริการเครือข่ายหรือการเชื่อมต่อระหว่าง Compute Unit เช่น Switch, Router, Firewall เป็นต้น
Resource	เป็นรูปแบบข้อมูลซึ่งจำลองทรัพยากรของอุปกรณ์ต่างๆ	พื้นที่ในการเก็บข้อมูลของ storage, ความสามารถในการประมวลผลของCPU, ความจุของ RAM, Bandwidth ของ NIC เป็นต้น
Connection	เป็นรูปแบบข้อมูลซึ่งจำลองการเชื่อมต่อระหว่างวัตถุใดๆที่เป็นวัตถุชนิดเดียวกัน	Connection ระหว่าง Device ของ Compute Unit ใดๆ อาจหมายถึงความเป็นไปได้ที่ Compute Unit เหล่านั้นจะติดต่อกันผ่านทางการใช้งานอินเตอร์เน็ต (Internet)

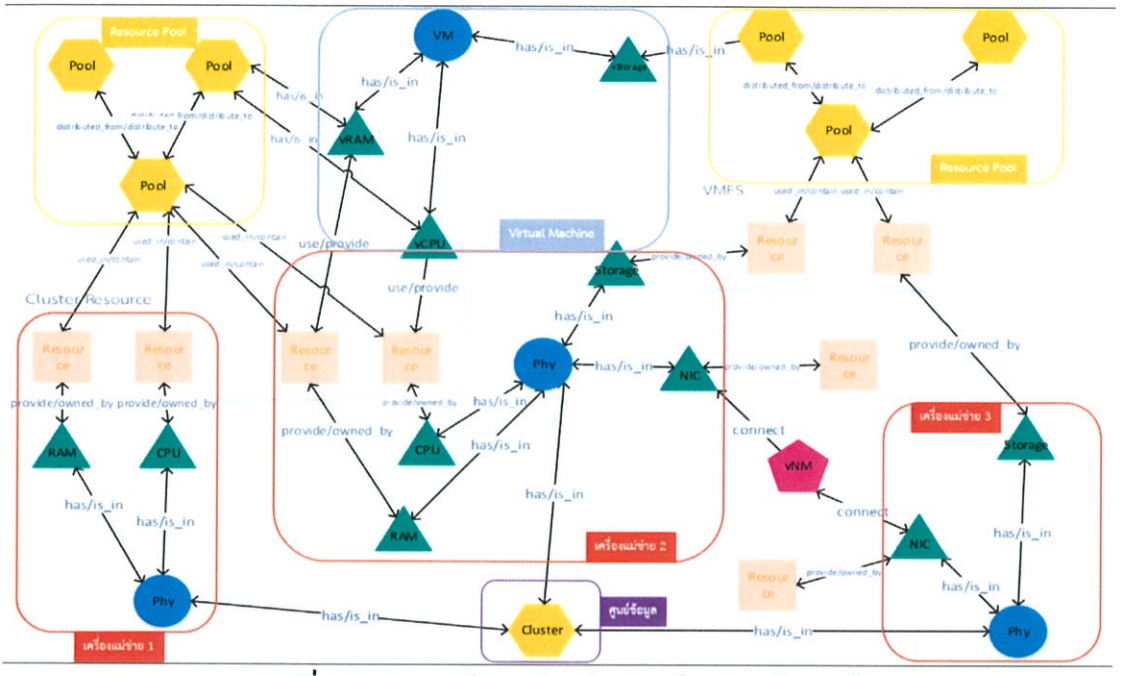
Aggregation /Distribution	เป็นรูปแบบข้อมูลซึ่งจำลองการรวม การจัดกลุ่ม หรือการแยกออกของวัตถุที่เป็นวัตถุชนิดเดียวกัน	การทำ Cluster ระหว่างเครื่องคอมพิวเตอร์, การรวมทรัพยากรเพื่อทำ Resource Pool เป็นต้น

3.3.2 สัญลักษณ์ของแต่ละประเภทของวัตถุ

Compute Unit	Device	Resource	Connection	Aggregation/ Distribution



รูปที่ 3.4 ตัวอย่างแบบจำลองและความสัมพันธ์

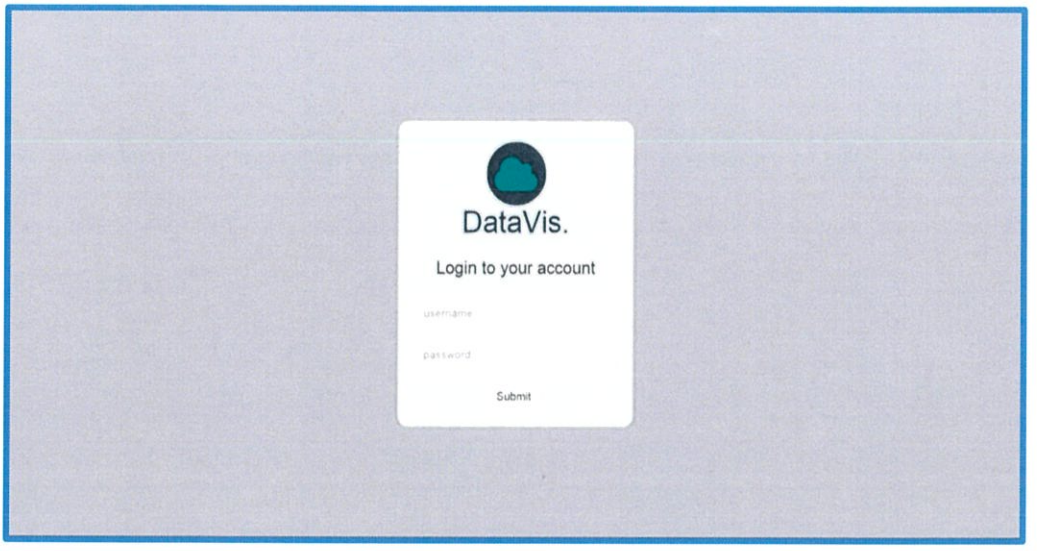


รูปที่ 3.5 รายละเอียดแต่ละส่วนภายในแบบจำลองข้อมูล

จากรูปที่ 3.5 เป็นตัวอย่างของแบบจำลองข้อมูลของศูนย์ข้อมูลซึ่งมีเครื่องแม่ข่าย 3 ตัว (กรอบสีแดง) ต่อกับเชื่อมต่อกันเป็นศูนย์ข้อมูล (กรอบสีม่วง) โดยเครื่องแม่ข่าย 1 และ เครื่องแม่ข่าย 2 ได้นำทรัพยากร RAM และ CPU มารวมกันทำเป็น resource pool และแยกออกเป็น resource pool ย่อย 2 resource pool (กรอบสีเหลืองด้านซ้าย) ส่วน resource pool ในด้านขวา (กรอบสีเหลืองด้านขวา) เป็นการรวมทรัพยากรของ storage เป็น resource pool และแยกออกเป็น resource pool ย่อย 2 resource pool และเครื่อง virtual machine (กรอบสีฟ้า) ได้ใช้ทรัพยากรจาก resource pool 2 resource pool ประกอบกัน

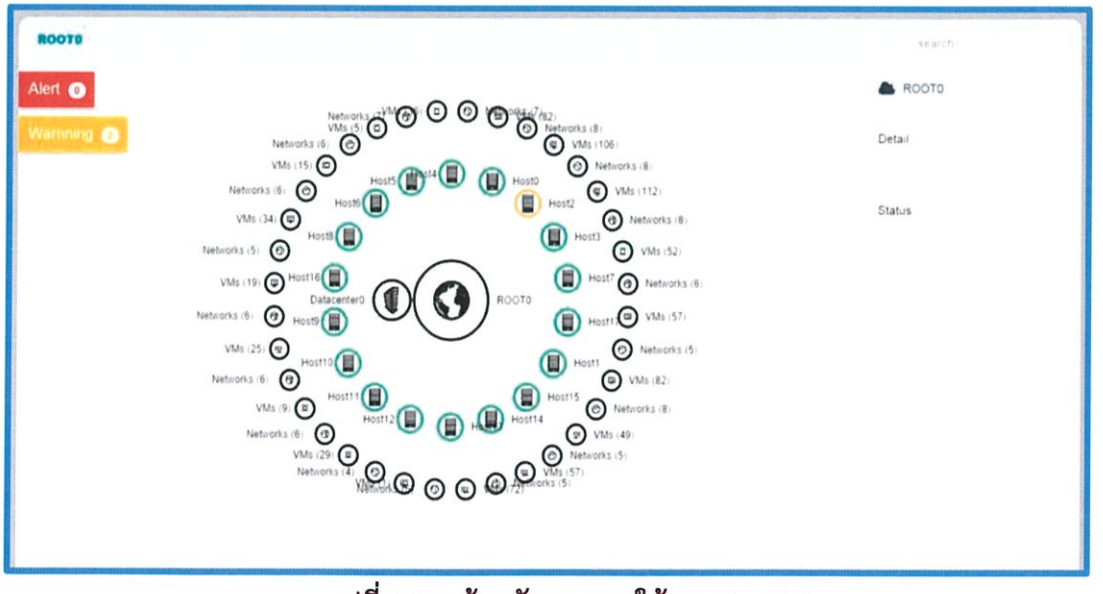
3.4 ส่วนแสดงผล DataVis.

ผู้ใช้สามารถเข้าใช้งาน DataVis. ได้โดยผ่านหน้าเว็บไซต์ โดยจะได้ผลลัพธ์ดังรูปที่ 3.6



รูปที่ 3.6 หน้าแรกของการเข้าใช้งาน DataVis.

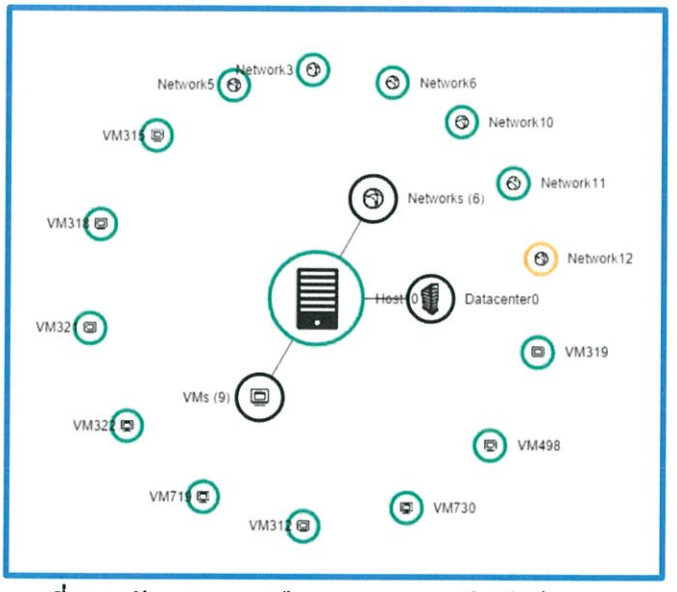
ซึ่งเมื่อการลงชื่อเข้าใช้สำเร็จจะพบหน้าหลักของ DataVis. ดังรูปที่ 3.7



รูปที่ 3.7 หน้าหลักของการใช้งาน DataVis.

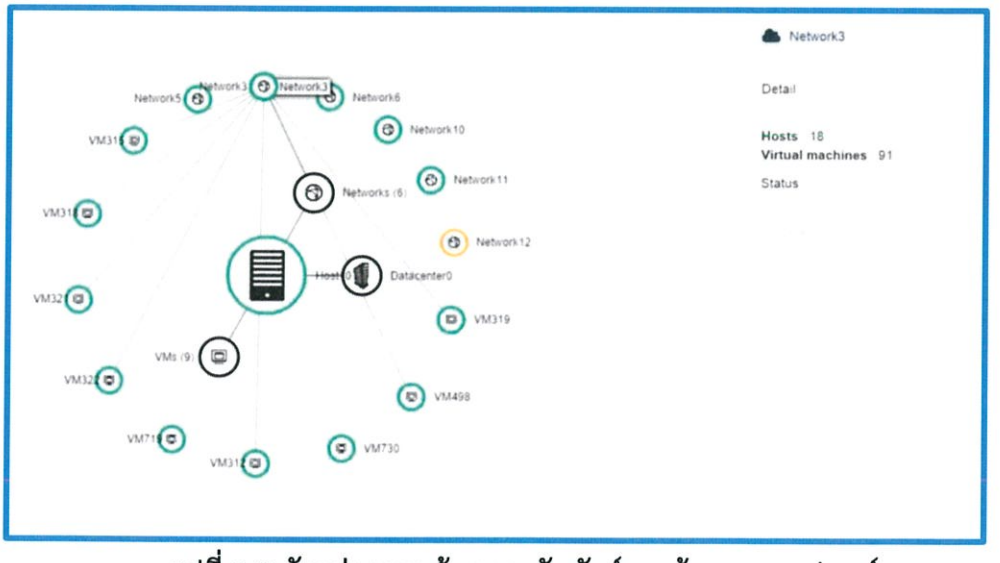
DataVis. ประกอบด้วย 5 ส่วนดังนี้

1. ผู้ใช้สามารถคลิกที่อุปกรณ์ที่สนใจเพื่อแสดงเฉพาะความสัมพันธ์ที่เกี่ยวข้องกับอุปกรณ์นั้นๆ ดังรูปที่ 3.8



รูปที่ 3.8 ตัวอย่างการเลือกแสดงความสัมพันธ์เฉพาะอุปกรณ์

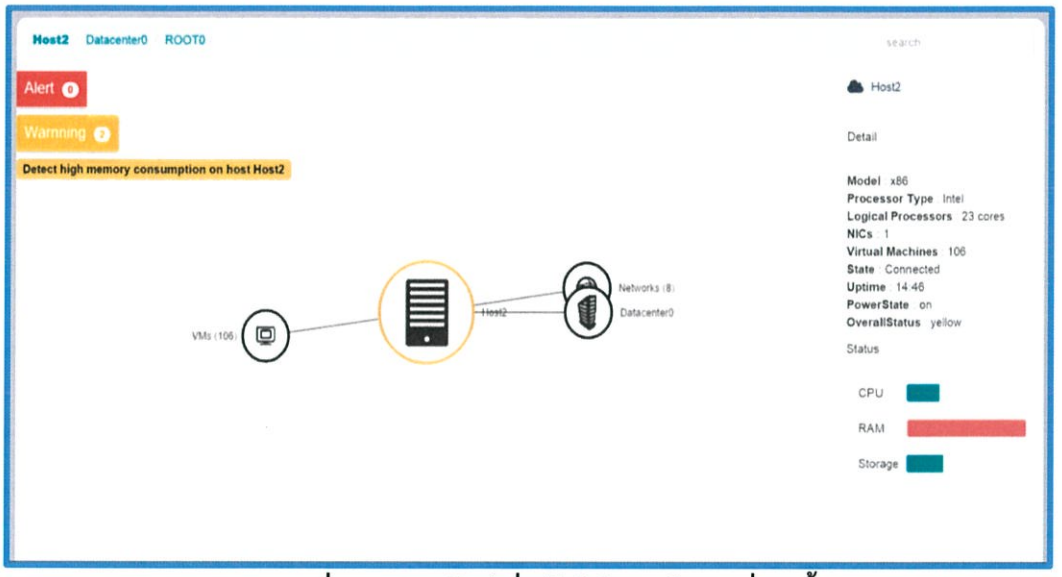
2. ผู้ใช้สามารถวางตัวชี้ลงบนอุปกรณ์ที่สนใจเพื่อแสดงความสัมพันธ์ของอุปกรณ์นั้นๆ กับอุปกรณ์รอบข้างและดูสถานะของอุปกรณ์ทางด้านขวามือไปพร้อมๆ กันได้ดังรูปที่ 3.9



รูปที่ 3.9 ตัวอย่างการเน้นความสัมพันธ์และข้อมูลของอุปกรณ์

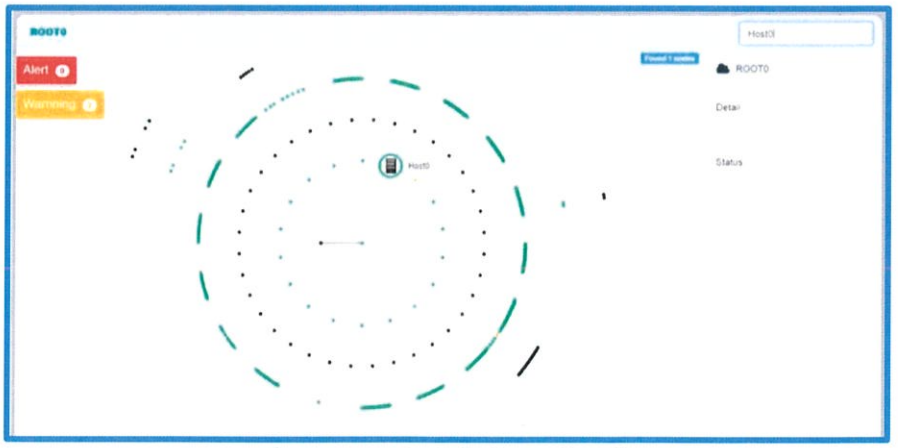
3. ผู้ใช้สามารถดูปัญหาต่างๆที่เกิดขึ้นได้โดยคลิกที่แถบ Alert และ Warning ดังรูปที่ 3.10 และรูปที่ 3.11

รูปที่ 3.10 ตัวอย่างการแสดงปัญหาที่เกิดขึ้น



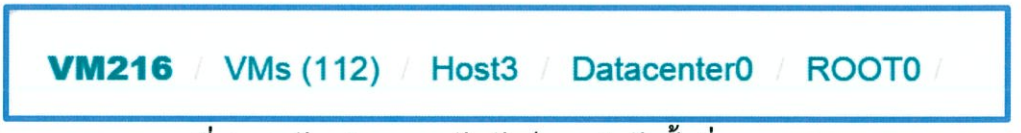
รูปที่ 3.11 ผลลัพธ์เมื่อผู้ใช้เลือกดูปัญหาที่เกิดขึ้น

4. ผู้ใช้สามารถค้นหาอุปกรณ์ที่ต้องการได้จากช่องค้นหาด้านบนขวาดังรูปที่ 3.12



รูปที่ 3.12 ผลลัพธ์เมื่อค้นหาด้วยชื่อ Host0

5. ผู้ใช้สามารถดูความสัมพันธ์แบบลำดับชั้นของอุปกรณ์ได้จากแถบแสดงความสัมพันธ์ด้านบนซ้าย จากรูปที่ 3.13 แปลว่า เครื่องหรือตำแหน่งปัจจุบันที่อยู่คือ เครื่อง VM216 ซึ่งอยู่บนเครื่อง Host3 ที่มี VM ทั้งหมด 112 เครื่องและเครื่อง Host3 นั้นอยู่ที่ Datacenter0



รูปที่ 3.13 ตัวอย่างความสัมพันธ์แบบลำดับชั้นที่ VM216

บทที่ 4
การทดลองและผลการทดลอง

ผู้พัฒนาได้แบ่งการทดสอบโปรแกรมออกเป็นสองส่วน ส่วนแรกจะทดสอบเกี่ยวกับความถูกต้องของโปรแกรมและความเร็วของการเรียกใช้ข้อมูล ส่วนที่สองจะทดสอบประสิทธิภาพของการแสดงผลข้อมูล ข้อมูลที่ใช้ในการทดลองมีขนาด 10.4 MB ประกอบด้วย

- ศูนย์ข้อมูล 1 ศูนย์
- เครื่องแม่ข่าย 5 เครื่อง
- เครื่อง virtual machine 13 เครื่อง
- การแบ่ง resource pool 3 ส่วน
- Datastore 5 ตัว
- Network เสมือน 6 เครือข่าย

4.1 การทดสอบเกี่ยวกับความถูกต้องของโปรแกรม

4.1.1 การทดสอบเกี่ยวกับความถูกต้องของการดึงข้อมูล

วิธีการทดสอบ: ทดลองดึงข้อมูลจากสภาพแวดล้อมจำลองภายใน vCenter ตรวจสอบความถูกต้องของข้อมูลที่ได้โดยเทียบกับข้อมูลจริงที่ปรากฏบน vCenter

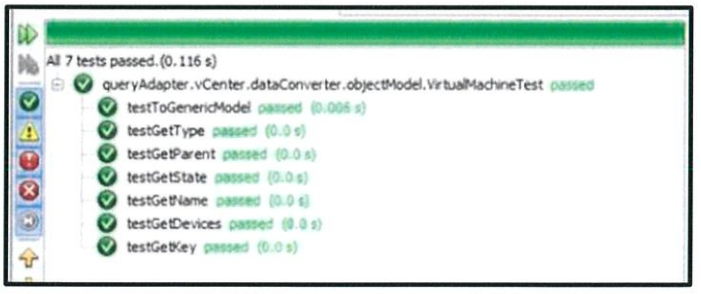
ผลลัพธ์: จากการทดลองดึงข้อมูลทุกชนิดที่สามารถเข้าถึงได้ของ vCenter ผ่าน Query Adapter พบว่าตรงกับข้อมูลใน vCenter ทั้งหมดดังรูปที่ 4.1

รูปที่ 4.1 ผลลัพธ์จากการทดลองดึงข้อมูล

4.1.2 การทดลองความถูกต้องในการแปลงข้อมูลโดยใช้ modeler

วิธีการทดสอบ: ใช้ชุดข้อมูลที่ได้จากสภาพแวดล้อมจำลองจากข้อแรกมาดำเนินการผ่าน modeler ทั้งนี้เนื่องจากข้อมูลมีจำนวนมากจึงมีการแบ่งการทดสอบออกเป็นหน่วยย่อยๆ (unit test)

ผลลัพธ์: จากการทดลองแปลงข้อมูลโดยใช้ modeler นั้นพบว่าสามารถทำงานได้อย่างถูกต้อง โดยมีระยะเวลาในการทำงานต่อข้อมูล 1 ชุด (เช่น การแปลงข้อมูลเครื่อง virtual machine 1 เครื่องให้อยู่ในรูปแบบตามแบบจำลองข้อมูล) ดังรูปที่ 4.2

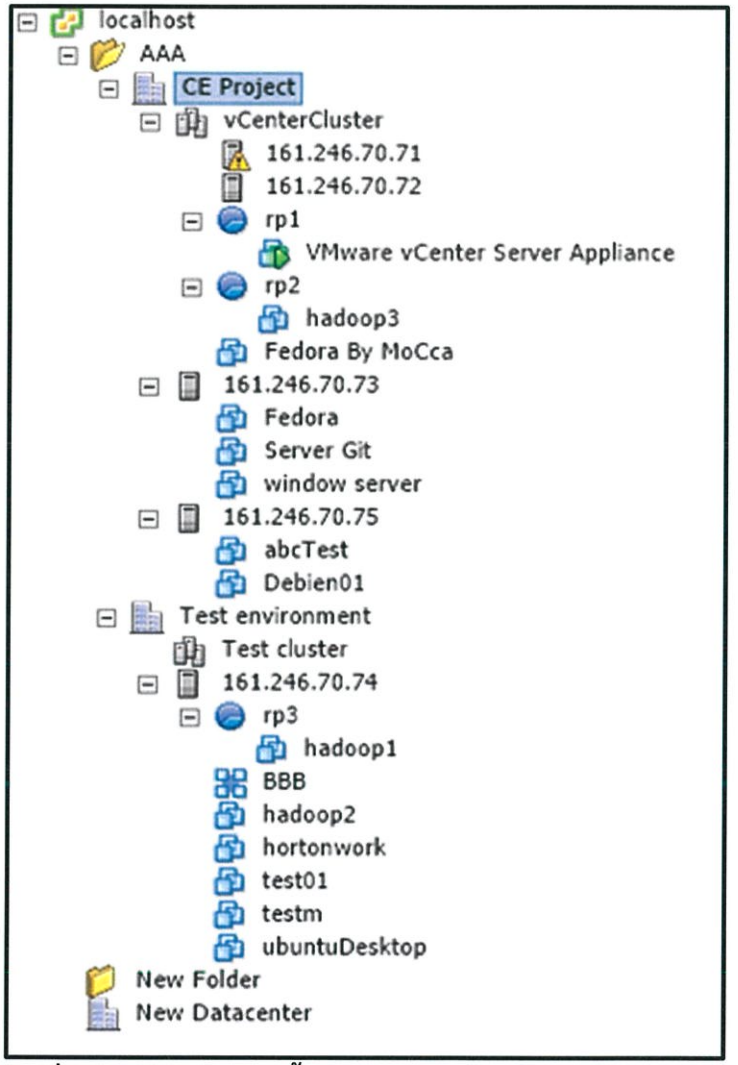


รูปที่ 4.2 ผลการทดลองแปลงข้อมูลโดยใช้ modeler

4.1.3 ทดสอบการตอบคำถามบนแบบจำลองข้อมูล

ทดสอบโดยตั้งคำถามเพื่อทดสอบความถูกต้องของข้อมูลและเพื่อให้แน่ใจได้ว่าแบบจำลองข้อมูลที่ออกแบบสามารถใช้ตอบคำถามในกรณีทดสอบได้ทั้งหมด 5 กรณีดังที่จะกล่าวต่อไป โดยแปลงคำถามในกรณีทดสอบดังกล่าวให้อยู่ในรูป cypher query เพื่อที่จะทดสอบกับข้อมูลแบบจำลองซึ่งเก็บอยู่ในฐานข้อมูล Neo4j โดยได้ จัดเตรียมสภาพแวดล้อมในการทดลองโดยใช้งานเครื่องคอมพิวเตอร์ทั้งหมด 5 เครื่องโดยแต่ละเครื่องติดตั้ง hypervisor คือ VMware ESXi 5.1 และ

ติดตั้ง VMware vSphere เป็นเครื่อง Virtual Machine ตามวิธีการที่ VMware แนะนำ [20] ซึ่งมีรายละเอียดสภาพแวดล้อมดังรูปที่ 4.3



รูปที่ 4.3 ผลจากการติดตั้งสภาพแวดล้อมจำลองบน vCenter

4.1.3.1 **ค้นหา Virtual Machine ทั้งหมดพร้อมทั้งแสดงเครื่อง Physical Host ที่ Virtual Machine นั้นๆอยู่**

โปรแกรมที่ 4.1 คำสั่ง cypher query ค้นหา Virtual Machine ทั้งหมดพร้อมทั้งแสดงเครื่อง Physical Host ที่ Virtual Machine นั้นๆอยู่

```
    MATCH (vm:ComputeUnit {type:"VirtualMachine"})
MATCH (vm)-->(:Device)-->(:Resource)<--(:Device)<--(host:ComputeUnit
```

```
{ type:"HostSystem"} )
RETURN vm.name AS VM, COLLECT(DISTINCT host.name) AS Host
```

ผลลัพธ์ : ถูกต้อง (รูปที่ 4.4)

VM	Host
hortonwork	161.246.70.74
window server	161.246.70.73
ubuntuDesktop	161.246.70.74
hadoop3	161.246.70.72
test01	161.246.70.74
testm	161.246.70.74
abcTest	161.246.70.75
hadoop2	161.246.70.74
Server Git	161.246.70.73
Fedora By MoCca	161.246.70.72
hadoop1	161.246.70.74
VMware vCenter Server Appliance	161.246.70.71
Debien01	161.246.70.75
Fedora	161.246.70.73

รูปที่ 4.4 ผลลัพธ์จากการทดลองที่ 4.1.3.1

4.1.3.2 ค้นหา Virtual Machine ที่กำหนดพร้อมทั้งแสดงเครื่อง Physical Host ที่ Virtual Machine นั้นอยู่

โปรแกรมที่ 4.2 คำสั่ง cypher query ค้นหา Virtual Machine ที่กำหนดพร้อมทั้งแสดงเครื่อง Physical Host ที่ Virtual Machine นั้นอยู่

```
    MATCH (vm:ComputeUnit { type:"VirtualMachine", name: "Fedora"} )
MATCH (vm)-->(:Device)-->(:Resource)<--(:Device)<--(host:ComputeUnit
{ type:"HostSystem"} )
RETURN vm.name AS VM, COLLECT(DISTINCT host.name) AS Host
```

ผลลัพธ์ : ถูกต้อง (รูปที่ 4.5)

VM	Host
Fedora	161.246.70.73

รูปที่ 4.5 ผลลัพธ์จากการทดลองที่ 4.1.3.2

4.1.3.3 ค้นหาว่าหาก Physical Host ที่กำหนดเกิดความเสียหายขึ้นจะส่งผลกระทบต่อ Virtual Machine เครื่องใดบ้าง

โปรแกรมที่ 4.3 คำสั่ง cypher query ค้นหาว่าหาก Physical Host ที่กำหนดเกิดความเสียหายขึ้นจะส่งผลกระทบต่อ Virtual Machine เครื่องใดบ้าง

```
    MATCH (host:ComputeUnit { name: "161.246.70.74"} )
MATCH (vm:ComputeUnit { type:"VirtualMachine"} )-->(:Device)--
>(:Resource)<--(:Device)<--(host)
RETURN host.name AS Host, COLLECT(DISTINCT vm.name) AS VMs
```

ผลลัพธ์ : ถูกต้อง (รูปที่ 4.6)

Host	VMs
161.246.70.74	ubuntuDesktop, hadoop1, hortonwork, hadoop2, test01

รูปที่ 4.6 ผลลัพธ์จากการทดลองที่ 4.1.3.3

4.1.3.4 แสดง Physical Host และ Virtual Machine ทั้งหมดพร้อมสถานการณ์การทำงาน

โปรแกรมที่ 4.4 คำสั่ง cypher query แสดง Physical Host และ Virtual Machine ทั้งหมดพร้อมสถานการณ์ทำงาน

```
    MATCH  (n:ComputeUnit)
RETURN n.type AS Type, n.name AS Name , n.state AS Status
```

ผลลัพธ์ : ถูกต้อง (รูปที่ 4.7)

Type	Name	Status
HostSystem	161.246.70.71	poweredOn
HostSystem	161.246.70.72	poweredOn
HostSystem	161.246.70.73	poweredOn
HostSystem	161.246.70.75	poweredOn
HostSystem	161.246.70.74	poweredOn
VirtualMachine	hortonwork	poweredOff
VirtualMachine	test01	poweredOn
VirtualMachine	hadoop1	poweredOff
VirtualMachine	VMware vCenter Server Appliance	poweredOn
VirtualMachine	window server	poweredOff
VirtualMachine	Fedora	poweredOff
VirtualMachine	Server Git	poweredOff
VirtualMachine	testm	poweredOff
VirtualMachine	ubuntuDesktop	poweredOff
VirtualMachine	hadoop2	poweredOff
VirtualMachine	abcTest	poweredOff
VirtualMachine	hadoop3	poweredOff
VirtualMachine	Debien01	poweredOff
VirtualMachine	Fedora By MoCca	poweredOff

รูปที่ 4.7 ผลลัพธ์จากการทดลองที่ 4.1.3.4

4.1.3.5 แสดงเครือข่ายเสมือนทั้งหมดพร้อมทั้งเครื่องที่เชื่อมต่อ

โปรแกรมที่ 4.5 คำสั่ง cypher query แสดงเครือข่ายเสมือนทั้งหมดพร้อมทั้งเครื่องที่เชื่อมต่อ

```
    MATCH (a:ComputeUnit {name:"VMware vCenter Server Appliance"})--
(d1:Device)--(n:Network)--(d2:Device)--(b:ComputeUnit)
RETURN a.name AS Name,collect(distinct n.name) AS
Networks,collect(distinct b.name) AS Connection
```

ผลลัพธ์ : ถูกต้อง (รูปที่ 4.8)

Name	Networks	Connection
VMware vCenter Server Appliance	VM Network	Server Git, window server, VMware vCenter Server Appliance, abcTest, hadoop3, Debien01, Fedora By MoCca, M(New), Fedora

รูปที่ 4.8 ผลลัพธ์จากการทดลองที่ 4.1.3.5

4.2 การทดสอบเกี่ยวกับการแสดงผลข้อมูล

ผู้พัฒนาได้จัดทำแบบสอบถามเพื่อประเมินผลในส่วนของการแสดงผลว่าตรงตามความต้องการในการใช้งานจริงหรือไม่ เพื่อนำผลไปปรับปรุงพัฒนาต่อไป โดยผู้ทดสอบมีผู้ที่ทำงานทางด้านการบริหารจัดการศูนย์ข้อมูลจริง และผู้ที่มีความรู้พื้นฐานทางศูนย์ข้อมูลมาทดลองใช้โปรแกรมที่พัฒนาขึ้นชื่อ DataVis. โดยแบบฟอร์มแบบสอบมีคำถามดังต่อไปนี้

ส่วนที่ 1 : ข้อมูลส่วนบุคคล

- เพศ
- ตำแหน่งงาน

ส่วนที่ 2 : รายละเอียดของปัญหา

- ปัญหาที่เจอบ่อยมักมีสาเหตุมาจากส่วนใด
- เมื่อเจอปัญหาจะใช้เครื่องมือใดบ้างในการช่วยหาสาเหตุของปัญหา
- รายละเอียดข้อมูลที่จำเป็นต้องใช้บ่อยหรือสำคัญ

ส่วนที่ 3 : การใช้งานโปรแกรม DataVis.

- Task 1 : ระบุจำนวน Virtual Machine และ Physical Host ภายใน Network
- Task 2 : สำรวจปัญหาที่เกิดขึ้นภายใน Datacenter

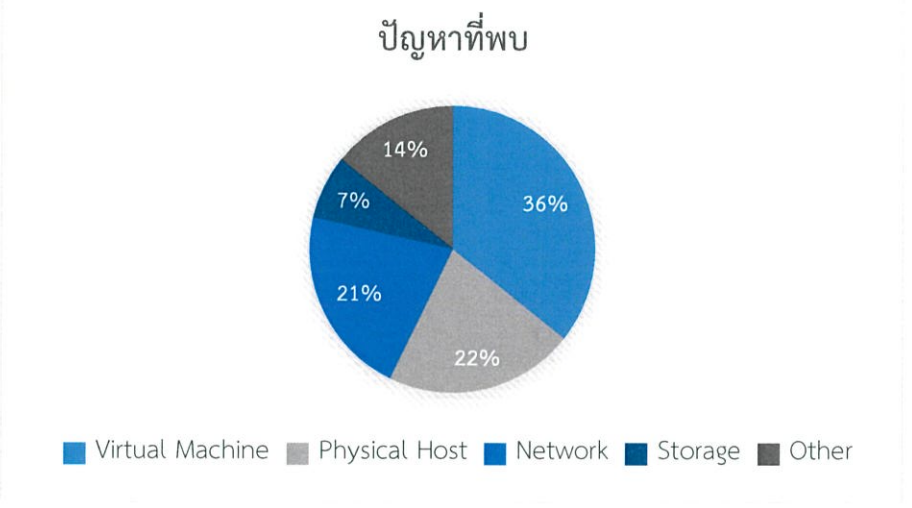
ส่วนที่ 4 : ความพึงพอใจในการใช้งานส่วนแสดงผล

- ความรู้สึกในการใช้งาน
- ประโยชน์การใช้งานในแต่ละฟังก์ชัน
- การเลือกเครื่องมือสำหรับแต่ละการทำงาน

โดยได้มีผู้ทดสอบทั้งหมดจำนวน 7 คน แบ่งเป็นผู้ที่เคยใช้งานเครื่องมือ vCenter จำนวน 6 คนและผู้ที่ทำงานในการบริหารจัดการศูนย์ข้อมูล 1 คน ซึ่งได้ผลลัพธ์ดังนี้

1) ปัญหาที่เจอบ่อยมักมีสาเหตุมาจากส่วนใด (สามารถเลือกได้มากกว่า 1 คำตอบ)

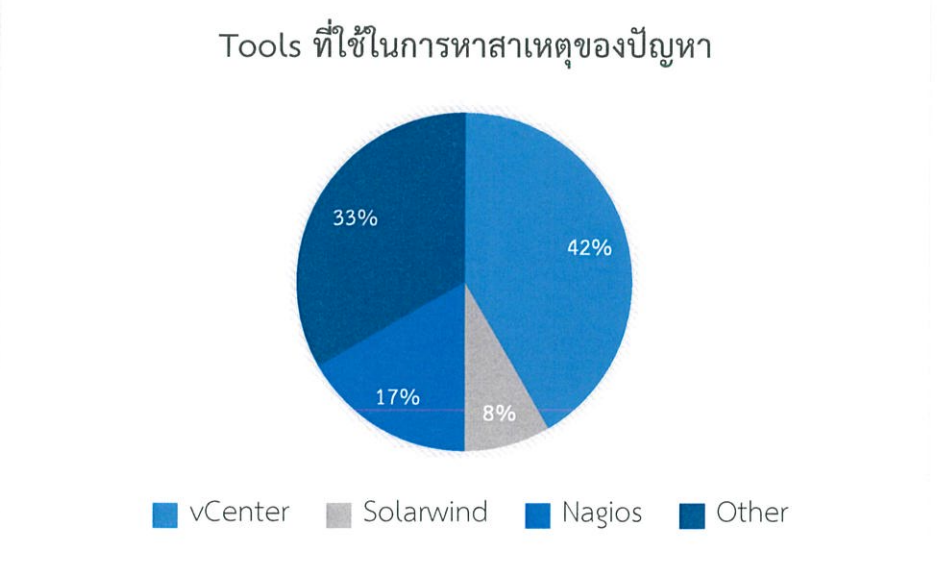
จากรูปที่ 4.9 จะเห็นได้ว่าผู้ทดสอบพบเจอปัญหาจาก Virtual Machine มากที่สุดคือ 36% รองลงมาคือ Physical Host 22% และ Network 21% และอื่นๆรวมกันอีก 21%



รูปที่ 4.9 กราฟวงกลมแสดงสาเหตุของปัญหาภายในศูนย์ข้อมูล

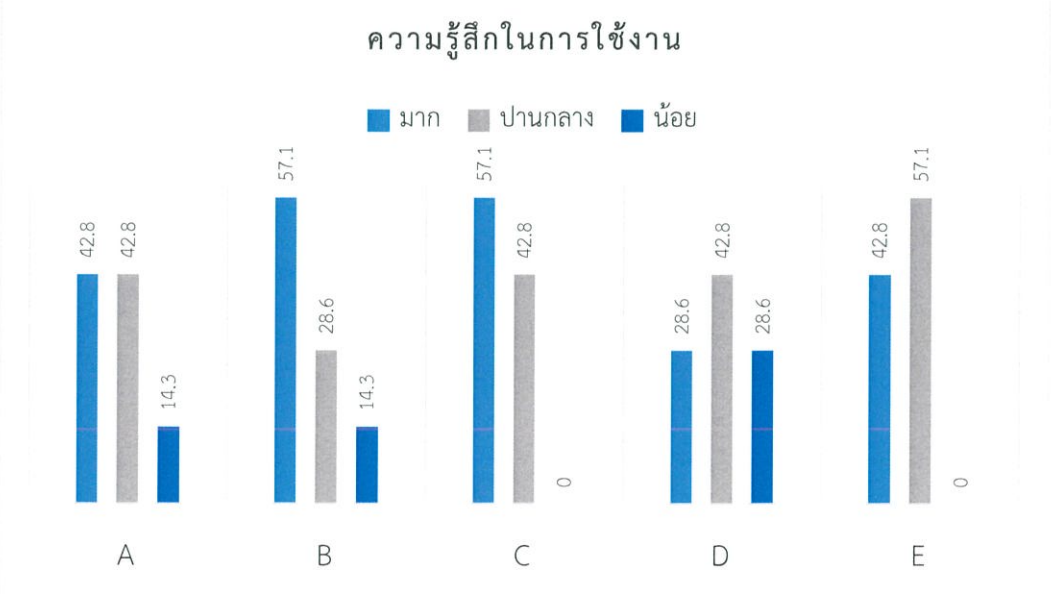
2) เมื่อเจอปัญหาจะใช้เครื่องมือใดบ้างในการช่วยหาสาเหตุของปัญหา

จากรูปที่ 4.10 จะเห็นได้ว่าผู้ทดสอบใช้เครื่องมือ vCenter ในการช่วยหาสาเหตุของปัญหามากที่สุดถึง 42% รองลงมาเป็นเครื่องมืออื่นๆ 33%



รูปที่ 4.10 กราฟวงกลมแสดง Tools ที่ใช้ในการหาสาเหตุของปัญหา

3) รายละเอียดข้อมูลที่จำเป็นต้องใช้บ่อยหรือสำคัญ
    a. 85.7 % เห็นว่าข้อมูลที่ต้องใช้บ่อยหรือสำคัญมากที่สุดคือ Status การทำงานของเครื่อง เช่น Virtual Machine หรือ Physical Host, การใช้งานทรัพยากรของอุปกรณ์ (Resource Usage) และ แจ้งเตือน (Alarm)
4) การใช้งาน DataVis. : Task 1 => ระบุจำนวน Virtual Machine และ Physical Host ภายใน Network
    a. สามารถหาคำตอบและตอบได้ถูกต้อง 42.9 %
5) การใช้งาน DataVis. : Task 2 => สำรวจปัญหาที่เกิดขึ้นภายใน Datacenter
    a. สามารถสำรวจปัญหาที่เกิดขึ้นได้ครบถ้วน 71.4 %
6) ความรู้สึกในการใช้งาน ตามที่แสดงในรูปที่ 4.11
    a. ให้รายละเอียดได้ครบถ้วน : มาก 42.8 % ปานกลาง 42.8 % น้อย 14.3%
    b. ง่ายต่อความเข้าใจ : มาก 57.1 % ปานกลาง 28.6 % น้อย 14.3%
    c. แสดงข้อมูลที่เพียงพอต่อการใช้งาน : มาก 57.1 % ปานกลาง 42.8 % น้อย 0 %
    d. ความสัมพันธ์ระหว่างแต่ละอุปกรณ์มากจนสับสน : มาก 28.6% ปานกลาง 42.8% น้อย 28.6%
    e. ประโยชน์การใช้งานในภาพรวม : มาก 42.8% ปานกลาง 57.1% น้อย 0%



รูปที่ 4.11 กราฟแท่งแสดงความรู้สึกในการใช้งาน

7) ประโยชน์การใช้งานในแต่ละฟังก์ชันตามที่แสดงในรูปที่ 4.12

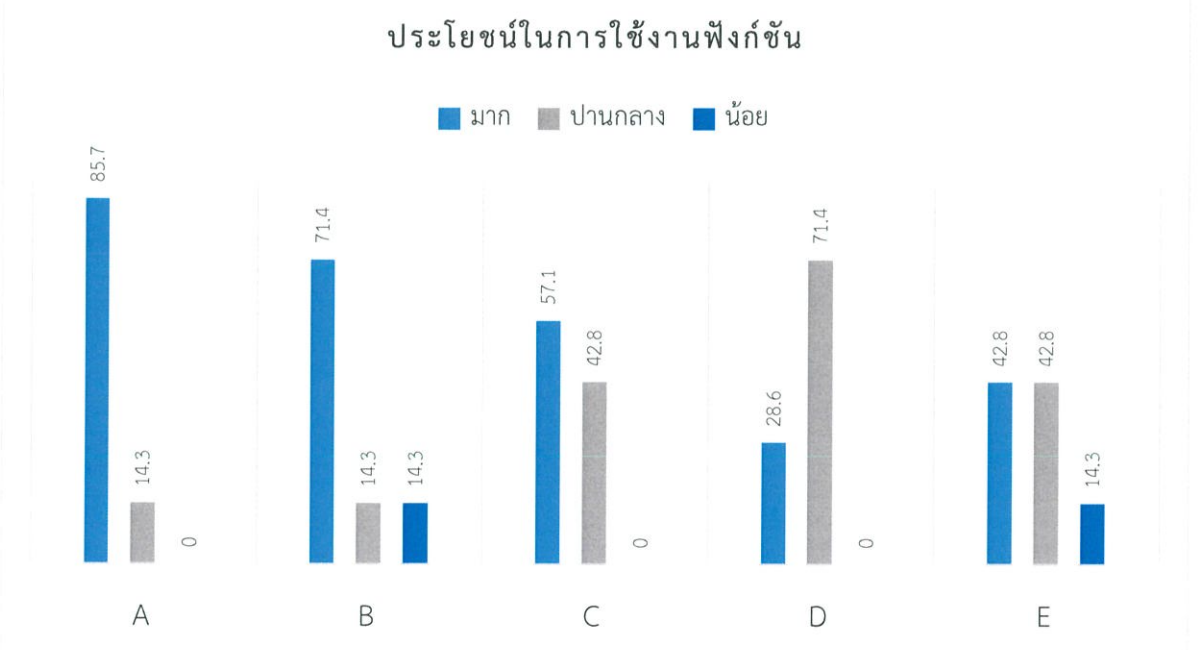
a. การแสดงผลเป็นกราฟความสัมพันธ์ระหว่างอุปกรณ์ : มาก 85.7% ปานกลาง 14.3% น้อย 0%

b. การแสดง path การเข้าถึงอุปกรณ์ : มาก 71.4% ปานกลาง 14.3% น้อย 14.3%

c. การแสดงการแจ้งเตือนปัญหา : มาก 57.1% ปานกลาง 42.8% น้อย 0%

d. Data panel ที่แสดงรายละเอียดของอุปกรณ์ : มาก 28.6% ปานกลาง 71.4% น้อย 0%

e. การค้นหาอุปกรณ์ที่ต้องการจากชื่อเครื่องในช่อง search : มาก 42.8% ปานกลาง 42.8% น้อย 14.3%



รูปที่ 4.12 กราฟแท่งแสดงประโยชน์ในการใช้งานฟังก์ชัน

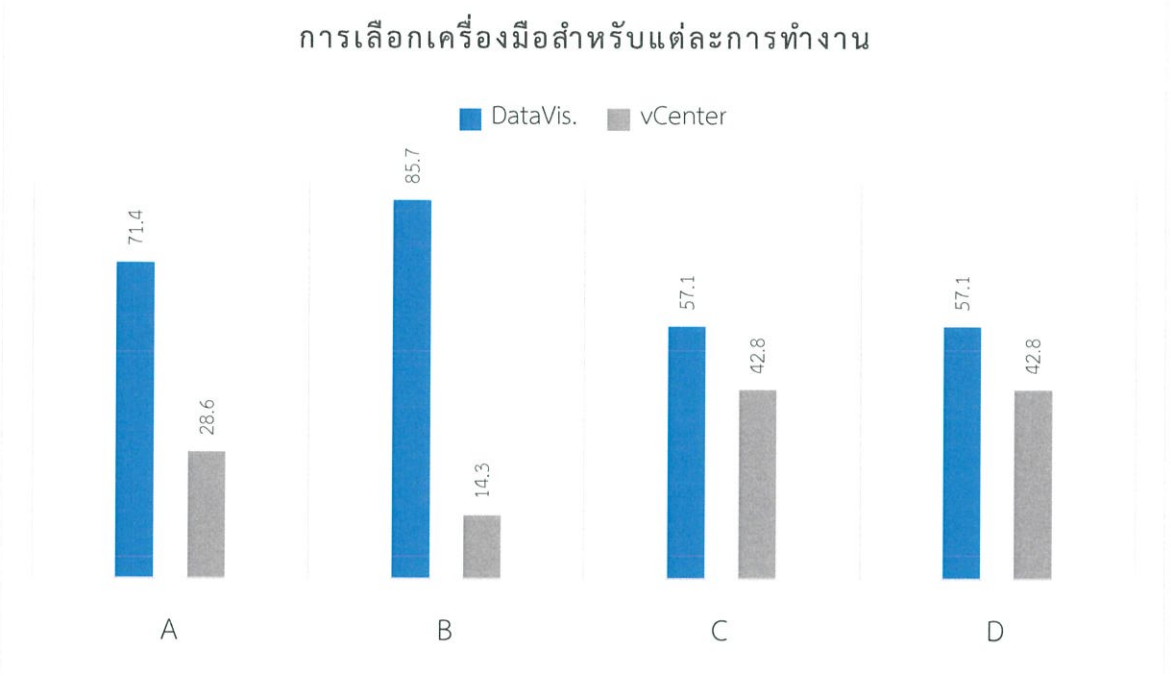
8) การเลือกเครื่องมือสำหรับแต่ละการทำงาน ตามที่แสดงในรูปที่ 4.13

a. การเข้าถึงข้อมูลที่ต้องการได้อย่างรวดเร็ว : DataVis. 71.4% vCenter 28.6%

b. การแสดงความสัมพันธ์หรือการเชื่อมต่อระหว่างแต่ละอุปกรณ์ : DataVis. 85.7% vCenter 14.3%

c. การช่วยวิเคราะห์ ตรวจสอบ ตรวจจับปัญหาได้สะดวก : DataVis. 57.1% vCenter 42.8%

d. ภาพรวมทั้งหมด : DataVis. 57.1% vCenter 42.8%



รูปที่ 4.13 กราฟแท่งแสดงการเลือกเครื่องมือสำหรับแต่ละการทำงาน

4.2.1 ผลการวิเคราะห์แบบสอบถามและข้อสรุป

จากผลสำรวจพบว่า ปัญหาที่พบเจอในศูนย์ข้อมูลนั้นมาจาก Virtual Machine และ Physical Host เป็นส่วนใหญ่ ซึ่งก็จะใช้เครื่องมือ vCenter ช่วยในการหาสาเหตุของปัญหาต่างๆ โดยข้อมูลที่มีความสำคัญในการใช้งานส่วนใหญ่ คือ

1) Status การทำงานของเครื่อง Virtual Machine หรือ Physical Host

2) Status ของ Network

3) การใช้งานทรัพยากรของอุปกรณ์

4) แจ้งเตือน (Alarm)

5) ข้อมูลที่เกี่ยวข้องกับ Virtual Machine เครื่องนั้นๆ เช่น Network หรือ Datastore

ในการใช้งานส่วนแสดงผล (DataVis.) นั้นได้ทดลองให้ผู้ใช้ตอบคำถามที่กำหนดมาโดยให้หาคำตอบด้วยตนเอง จากการใช้งานครั้งแรกพบว่า 60% สามารถตอบคำถามได้และสำรวจปัญหาที่เกิดขึ้นได้อย่างถูกต้อง โดยที่ผู้ใช้ไม่เคยใช้โปรแกรม DataVis และไม่ได้มีการศึกษาการใช้โปรแกรมนี้มาก่อน ซึ่งทำให้เห็นว่าส่วนแสดงผลนั้นง่ายต่อความเข้าใจ ไม่จำเป็นต้องผ่านการฝึกฝนการใช้เครื่องมือมาก็สามารถใช้งานได้ และสามารถแสดงความสัมพันธ์ระหว่างแต่ละอุปกรณ์ออกมาได้ดี อย่างไรก็ดีโปรแกรมยังขาดข้อมูลรายละเอียดเชิงลึกของอุปกรณ์ ซึ่งในอนาคตเราสามารถเพิ่มข้อมูลที่ผู้ใช้ต้องการได้โดยไม่ต้องเปลี่ยนแปลงส่วนอื่นๆภายในระบบได้

สรุปโดยภาพรวมนั้น DataVis. มีจุดเด่นในการแสดงความสัมพันธ์และการเชื่อมต่อของอุปกรณ์ต่างๆภายในศูนย์ข้อมูลได้ดี ทำให้เห็นภาพรวมของอุปกรณ์ที่เชื่อมต่อกันอยู่มากขึ้น และส่วนแจ้งเตือนช่วยให้สามารถจำกัดขอบเขตของปัญหาได้อย่างสะดวกและรวดเร็วมากขึ้น และยังสามารถช่วยให้เห็นถึงผลกระทบของปัญหาซึ่งอาจเกิดขึ้นกับอุปกรณ์อื่นที่เชื่อมต่อกันอยู่ ในส่วนของข้อมูลรายละเอียดของอุปกรณ์จำเป็นต้องมีข้อมูลมากกว่านี้เพื่อให้สามารถวิเคราะห์ปัญหาได้ดียิ่งขึ้น

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุปของโครงงาน

ในโครงงานนี้ได้สร้างระบบที่จะสามารถรวบรวมข้อมูลจากเครื่องมือต่างๆ ที่ใช้ในการบริหารจัดการศูนย์ข้อมูลเพื่อให้สามารถเข้าถึงข้อมูลต่างๆ เหล่านั้นได้จากที่เดียว ซึ่งจะช่วยให้สามารถวิเคราะห์ปัญหาหรือแนวโน้มในอนาคตได้ง่ายขึ้น นอกจากนี้โครงงานนี้ยังมุ่งเน้นให้สามารถแสดงผลข้อมูลเหล่านี้ออกมาให้เข้าใจง่ายโดยไม่จำเป็นต้องมีความรู้หรือความเชี่ยวชาญในเครื่องมือที่ใช้ และสามารถเข้าถึงข้อมูลต่างๆ ที่ต้องการได้อย่างสะดวกและรวดเร็ว ทางผู้พัฒนาจึงได้นำแนวคิดเกี่ยวกับการแสดงผลแบบกราฟเข้ามาใช้ในการสร้างแบบจำลองของข้อมูลขนาดใหญ่ที่มีความสัมพันธ์ซับซ้อน แบบจำลองที่สร้างขึ้นมานี้ได้มีการออกแบบให้รองรับข้อมูลที่มาจากเครื่องมือหลายๆ ประเภทได้ หน้าที่หลักของแบบจำลองคือรวบรวมและเชื่อมข้อมูลทั้งหมดเข้าด้วยกันเพื่อทำให้เห็นเป็นภาพรวมหนึ่งเดียว นอกจากนี้ผู้พัฒนาได้ออกแบบส่วนแสดงผลข้อมูลต่อผู้ใช้งาน โดยจะแสดงเฉพาะข้อมูลที่เกี่ยวข้องกับสิ่งที่ผู้ใช้สนใจเพื่อลดความสับสนและเวลาในการหาข้อมูลที่เกี่ยวข้อง แทนที่ผู้ใช้จะต้องเข้าไปค้นหาข้อมูลต่างๆ ที่กระจัดกระจายในแต่ละเครื่องมือ โดยเรียกเครื่องมือนี้ว่า DataVis. จากแบบสอบถามจะเห็นได้ว่า DataVis. มีจุดเด่นในเรื่องของการแสดงความสัมพันธ์ระหว่างอุปกรณ์ต่างๆ ซึ่งช่วยจำกัดขอบเขตในการค้นหาข้อมูลเพื่อหาสาเหตุของปัญหา ซึ่งในส่วนของรายละเอียดข้อมูลอื่นๆ ที่ยังขาดไปก็ยังสามารถเพิ่มเติมได้ในอนาคตซึ่งจะช่วยให้สามารถวิเคราะห์ส่วนต่างๆ ได้อย่างมีประสิทธิภาพมากขึ้น นอกจากนี้ระบบยังถูกออกแบบมาให้รองรับเครื่องมืออื่นๆ โดยจะมี API ที่ใช้ในการติดต่อกับระบบให้ ทำให้สามารถเพิ่มเครื่องมือเข้ามาในระบบได้ โดยในปัจจุบันสามารถรองรับข้อมูลจาก VMware vCenter ได้

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

5.2.1 ส่วนแบบจำลองข้อมูล

แบบจำลองข้อมูลที่ดีควรจะสามารถขยายและรองรับข้อมูลชนิดใหม่ๆ จากเครื่องมืออื่นๆ ได้ ทำให้ต้องใช้เวลาศึกษาข้อมูลต่างๆ จากเครื่องมือหลายๆ ประเภทเพื่อช่วยในการออกแบบหน่วยพื้นฐาน (primitive types) จริงๆ ที่ใช้กันแพร่หลายในเครื่องมือต่างๆ นอกจากนี้แบบจำลองที่ดี

จะต้องทำให้การค้นหาข้อมูลเป็นไปได้อย่างรวดเร็วเพื่อช่วยในการหาข้อมูลที่เกี่ยวข้องกับจุดที่สนใจให้ได้อย่างมีประสิทธิภาพ

5.2.2 ส่วนการแสดงผล

การแสดงผลให้ตรงกับความต้องการใช้งานของผู้ใช้งานจริงเป็นเรื่องที่จำเป็นต้องมีการเก็บรวบรวมข้อมูลเพื่อให้สามารถแสดงผลได้อย่างตรงจุดประสงค์หลักของโครงงานนี้ซึ่งก็คือให้สามารถใช้งานได้สะดวกและง่ายต่อความเข้าใจ ดังนั้นทางผู้พัฒนาจำเป็นต้องติดต่อกับผู้ทำงานจริงในด้านนี้และขอคำแนะนำเพื่อนำไปพัฒนาปรับปรุงต่อไปโดยทางผู้พัฒนาได้จัดทำแบบสอบถามเพื่อประเมินส่วนแสดงผล ทั้งนี้เพื่อให้ทราบว่าควรแสดงผลอย่างไร หรือควรแสดงข้อมูลในส่วนใดบ้างให้สอดคล้องกับความต้องการใช้งานและเพียงพอต่อการวิเคราะห์ปัญหาหรือเหตุการณ์โดยไม่มากเกินไปจนทำให้เกิดความสับสนหรือไม่สามารถกลั่นกรองข้อมูลเพื่อหาสาเหตุของปัญหาได้ ซึ่งปัจจัยหนึ่งที่จะทำให้ส่วนการแสดงผลออกมาได้ตรงตามความต้องการใช้งานของผู้ใช้คือการร่วมมือกันระหว่างผู้พัฒนาและผู้ใช้งาน ซึ่งการติดต่อกับผู้ใช้งานจริงจำเป็นต้องใช้เวลาและความร่วมมือ ผู้พัฒนาจึงต้องเข้าถึงผู้ใช้งานให้ได้เพื่อให้สามารถประเมินผล เก็บผลและสรุปวิเคราะห์ในการปรับปรุงส่วนแสดงผลต่อไป ปัญหาอีกปัญหาหนึ่งคือ ปัญหาในส่วนของความเคยชินกับเครื่องมือเดิมที่เคยใช้อยู่ทำให้อาจจะมีความรู้สึกกับเครื่องมือใหม่ๆว่าใช้งานได้ยาก ในจุดนี้จึงจำเป็นต้องสร้างจุดเด่นและความง่ายในการใช้งานเพื่อให้ผู้ใช้หันมาใช้งาน DataVis.

5.3 แนวทางในการพัฒนาต่อ

โครงงานถูกออกแบบมาให้สามารถเพิ่มเครื่องมืออื่นๆเข้ามาในระบบได้โดยจะมี API ที่ใช้ในการติดต่อกับระบบ ทำให้สามารถรองรับเครื่องมือต่างๆในอนาคตได้ และจะพัฒนาในส่วนของการแสดงผลบน native application ต่อไป ซึ่งเป็นไปได้ว่าส่วนการแสดงผลอาจต้องมีการแก้ไขเพื่อให้สามารถรองรับการแสดงผลที่แยกย่อยตามชนิดข้อมูลมากขึ้น สำหรับส่วนแสดงผล จากแบบสอบถามส่วนแสดงผลยังคงให้รายละเอียดต่างๆได้ไม่เพียงพอสำหรับการวิเคราะห์สถานการณ์ที่เกิดขึ้น ดังนั้นจะพัฒนาส่วนแสดงผลให้แสดงข้อมูลได้อย่างครบถ้วนครอบคลุมมากขึ้น และพัฒนาในส่วนการเข้าถึงอุปกรณ์ต่างๆให้เข้าถึงได้ง่ายยิ่งขึ้น เช่น เรียงลำดับ Virtual Machine หรือ Physical Host จากชื่อหรือ IP Address เป็นต้น

บรรณานุกรม

[1] "Intel® Virtualization Technology," [Online]. Available: http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html. [Accessed 12 5 2015].

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

[3] "VMware vSphere Basics," [Online]. Available: https://www.google.co.th/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0CDAQFjAD&url=https%3A%2F%2Fpubs.vmware.com%2Fvsphere-50%2Ftopic%2Fcom.vmware.ICbase%2FPDF%2Fvsphere-esxi-vcenter-server-50-basics-guide.pdf&ei=P-RRVd3gLtKXuASUp4G4BQ&usg. [Accessed 12 5 2015].

[4] [Online]. Available: http://www.nagios.org/.

[5] S. Weber, "NoSQL Databases," University of Applied Sciences HTW Chur, Switzerland, http://wiki.hsr.ch/Datenbanken/files/Weber_NoSQL_Paper.pdf.

[6] "Neo4j Highlights," [Online]. Available: http://neo4j.com/docs/stable/introduction-highlights.html. [Accessed 15 ธันวาคม 2557].

[7] J. W. a. E. E. Ian Robinson, "Models and Goals," in *Graph Database*, California, O'Reilly, 2013, p. 25.

[8] "The Neo4j Graph Database," [Online]. Available: http://neo4j.com/docs/stable/graphdb-neo4j.html. [Accessed 15 ธันวาคม 2557].

[9] "Properties," [Online]. Available: http://neo4j.com/docs/stable/graphdb-neo4j-properties.html. [Accessed 15 ธันวาคม 2557].

[10] "Cypher Query Language," [Online]. Available: http://neo4j.com/docs/stable/cypher-query-lang.html. [Accessed 15 ธันวาคม 2557].

[11] "The Neo4j Manual v2.1.6 - Capacity," [Online]. Available: http://neo4j.com/docs/stable/capabilities-capacity.html. [Accessed 15 ธันวาคม 2557].

[12] H. Huang และ Z. Dong, "Research on architecture and query performance based on distributed graph database Neo4j," *Consumer Electronics, Communications and Networks (CECNet), 2013 3rd International Conference on,* pp. 533 - 536, 2013.

[13] V. Soundararajan และ L. Spracklen, "Simplifying Virtualization Management with Graph Databases," *VMWARE TECHNICAL JOURNAL,* เล่มที่ 2, pp. 46-53, 2556.

[14] [Online]. Available: http://docs.mongodb.org/manual/core/crud-introduction/.

[15] J. D. K. B. George Coulouris, DISTRIBUTED SYSTEMS Concepts and Design Fifth Edition, Addison-wesley, 2012.

[16] " Launch of RabbitMQ Open Source Enterprise Messaging," 2007.

[17] "Message oriented middleware," [Online]. Available: http://docs.oracle.com/cd/E19340-01/820-6424/aeraq/index.html.

[18] V. Friedman, "Data Visualization and Infographics," 14 มกราคม 2551. [Online]. Available: http://www.smashingmagazine.com/2008/01/14/monday-inspiration-data-visualization-and-infographics/. [Accessed 15 ธันวาคม 2557].

[19] M. Salvan, "Muriel's Tech Blog," 10 เมษายน 2556. [Online]. Available: http://blog.x-aeon.com/2013/04/10/a-quick-message-queue-benchmark-activemq-rabbitmq-hornetq-qpid-apollo/.

[20] "Downloading and deploying the vCenter Server Appliance 5.x (2007619)," 2 สิงหาคม 2012. [Online]. Available: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2007619. [Accessed 15 ธันวาคม 2557].