การพัฒนาฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่ง ในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์

นายอภิชาติ วรรธะมานี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

DEVELOPMENT OF VIRTUAL FUNCTION FOR INTERNET OF THINGS IN BUILDING ENERGY MANAGEMENT SYSTEM USING CLOUD TECHNOLOGY

MR. APICHART WANTAMANEE

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2016
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์
โดย	นายอภิชาติ วรรธะมานี
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล)

.....กรรมการ
(อาจารย์ ดร.ภาณุวัฒน์ จันทร์ภักดี)

.....กรรมการภายนอกมหาวิทยาลัย
(ดร.พนิตา พงษ์ไพบูลย์)

อภิชาติ วรรธะมานี : การพัฒนาฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์
(DEVELOPMENT OF VIRTUAL FUNCTION FOR INTERNET OF THINGS IN BUILDING ENERGY MANAGEMENT SYSTEM USING CLOUD TECHNOLOGY)
อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ. ดร. เชาวน์ดิศ อัศวกุล, 119 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนา และการสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์ ฟังก์ชันเสมือนนี้ถูกสร้างบนแพลตฟอร์มโอเพนสแต็กซึ่งเป็นแพลตฟอร์มคลาวด์แบบโอเพนซอร์สร่วมกับโปรแกรมโอเพนบาตงซึ่งเป็นโปรแกรมประสานฟังก์ชันโครงข่ายเสมือน เพื่อให้ผู้ดูแลระบบสร้างฟังก์ชันเสมือนสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารได้โดยง่าย และจัดสรรทรัพยากรคอมพิวเตอร์ให้เหมาะสมกับความต้องการของฟังก์ชันเสมือน วิทยานิพนธ์นี้สร้างและทดสอบฟังก์ชันเสมือนจำนวน 3 ฟังก์ชันเพื่อเป็นตัวอย่าง ลำดับที่หนึ่ง ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M สามารถเริ่มต้นการทำงานในระบบทดสอบได้ภายในระยะเวลาเฉลี่ย 262 วินาที โอเพนบาตงสามารถปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติให้สอดคล้องกับการเปลี่ยนแปลงปริมาณข้อมูลที่ถูกส่งเข้าสู่ระบบ จากระบบทดสอบที่สร้างขึ้น เครื่องเสมือนเอ็นเอสซีแอลสามารถถูกสร้างเพิ่มขึ้นภายในระยะเวลาเฉลี่ย 119 วินาที และถูกทำลายภายในระยะเวลา 5 วินาที ผลการทดสอบพบว่า เมื่อใช้เครื่องเสมือนเอ็นเอสซีแอล 2 เครื่อง แพลตฟอร์มโอเพนเอ็มทีซีสามารถรองรับข้อมูลต่อนาทีเพิ่มขึ้น 29.2 เปอร์เซ็นต์ และเมื่อใช้เครื่องเสมือนเอ็นเอสซีแอล 3 เครื่องสามารถรองรับข้อมูลต่อนาทีเพิ่มขึ้น 32.9 เปอร์เซ็นต์เมื่อเทียบกับการใช้เครื่องเสมือนเอ็นเอสซีแอล 1 เครื่อง เครื่องเสมือนมองโกดีบีสามารถเริ่มต้นการทำงานแบบอัตโนมัติเพื่อเพิ่มพื้นที่เก็บข้อมูลของแพลตฟอร์มโอเพนเอ็มทีซีภายในระยะเวลา 98 วินาที ลำดับที่สอง ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite สามารถเริ่มต้นทำงานภายในระยะเวลา 210 วินาที ฟังก์ชันเสมือนนี้สามารถวิเคราะห์ข้อมูลอุณหภูมิในห้องคลัสเตอร์คอมพิวเตอร์จากระบบ CU-BEMS เพื่อควบคุมการทำงานของเครื่องปรับอากาศที่เชื่อมต่อกับโนด ECHONET Lite ส่งผลให้สามารถควบคุมอุณหภูมิในห้องคลัสเตอร์คอมพิวเตอร์นี้ได้ ลำดับสุดท้าย ฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 สามารถเริ่มต้นการทำงานได้ภายในระยะเวลา 450 วินาที ฟังก์ชันเสมือนนี้แสดงอัตราส่วนค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศภายในอาคารผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบ เพื่อให้ผู้ใช้รับรู้ค่าความสูญเปล่าจากการใช้ไฟฟ้า นอกจากนี้ฟังก์ชันเสมือนนี้สามารถวิเคราะห์ข้อมูลการใช้งานโปรแกรมสร้างมโนภาพเชิงโต้ตอบ เพื่อแก้ไขปัญหาที่ผู้ใช้สามารถเข้าถึงได้เฉพาะแคนวาสปริยายเท่านั้น เนื่องจากโปรแกรมไม่ตอบสนองต่อการโบกมือของผู้ใช้งาน ยกตัวอย่างเช่น ฟังก์ชันเสมือนนี้จะดำเนินการเริ่มโปรแกรมใหม่เมื่อผู้ใช้งานมากกว่า 5 คนไม่สามารถเข้าใช้แคนวาสอื่นที่ไม่ใช่แคนวาสปริยายได้ภายในระยะเวลา 20 นาที โดยสรุปแล้ว ผลลัพธ์จากวิทยานิพนธ์นี้คาดหวังจะสามารถใช้เป็นแนวทางสำหรับการพัฒนาฟังก์ชันเสมือนต่างๆบนแพลตฟอร์มคลาวด์ในอนาคตได้ต่อไป

ภาควิชา วิศวกรรมไฟฟ้า **ลายมือชื่อนิสิต** .....
สาขาวิชา วิศวกรรมไฟฟ้า **ลายมือชื่อ อ.ที่ปรึกษาหลัก** .....
ปีการศึกษา 2559

5770572121 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS: VIRTUAL FUNCTION/ CLOUD/ IEEE1888/ ETSI M2M/ ECHONET LITE.
APICHART WANTAMANEE : DEVELOPMENT OF VIRTUAL FUNCTION FOR INTERNET OF THINGS IN BUILDING ENERGY MANAGEMENT SYSTEM USING CLOUD TECHNOLOGY.
ADVISOR: ASSOC. PROF. CHAODIT ASWAKUL, Ph.D., 119 pp.

This thesis introduces the development and implementation of a virtual function for internet of things in building energy management system using cloud technology. This virtual function has been implemented on an open-source cloud platform OpenStack using Open Baton, a network function virtualization orchestrator software. Thus, system administrators can implement the virtual function for building energy management system easily and allocate a suitable computer resource for the virtual function requirement. This thesis implements and tests 3 virtual functions as an example. Firstly, a virtual function of real-time interworking system between IEEE1888 and ETSI M2M can be started in the testbed within 262 seconds. Open Baton can adjust automatically the number of NSCL instances in response to the changing amount of input data. With the constructed testbed, an NSCL instance can be started within 119 seconds and terminated within 5 seconds. Based on the tested result, the OpenMTC platform can increase the data recording rate by 29.2 percentages with 2 NSCL instances, and 32.9 percentages with 3 NSCL instances, when compared with the case using only one NSCL instance. A MongoDB instance can be started automatically for extending the storage of OpenMTC platform within 98 seconds. Secondly, a virtual function of real-time interworking system between IEEE1888 and ECHONET Lite can be started within 210 seconds. This virtual function can analyze a temperature data inside a computer cluster room from CU-BEMS for controlling the air conditioner connecting with an ECHONET Lite node. As a result, the temperature inside this room can then be controlled. Finally, a virtual function of data analytic program for the IEEE1888-based building energy management system can be started within 450 seconds. This virtual function displays a wasted energy ratio of electrical energy used by air conditioning system on an interactive visualization program. Users thus get information about a wasted electricity consumption. In addition, this virtual function can analyze the interactive visualization program usage data for solving the problem that users can access only a default canvas. Since the program cannot interact with a user's hand waving. For instance, this virtual function would reset the interactive visualization program when more than 5 users cannot access another non-default canvas within 20 minutes. In conclusion, the result of this thesis is expected to demonstrate a proven framework for developing arbitrary virtual functions on a cloud platform in the future.

Department	: Electrical Engineering	**Student's Signature**	.....
Field of Study	: Electrical Engineering	**Advisor's Signature**	.....
Academic Year	: 2016		

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ รองศาสตราจารย์ ดร. เชาวน์ดิศ อัศวกุล ที่ถ่ายทอดความรู้ทั้งทางด้านวิจัยและด้านวิชาการ รวมถึงช่วยตรวจทานวิทยานิพนธ์ฉบับนี้อย่างดีเสมอมา ผู้วิจัยจึงขอกราบขอบพระคุณมา ณ ที่นี้ รวมถึงกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ชัยเชษฐ์ สายวิจิตร ประธานกรรมการสอบวิทยานิพนธ์ อาจารย์ ดร. ภาณุวัฒน์ จันทร์ภักดี และ ดร. พนิตา พงษ์ไพบูลย์ กรรมการสอบวิทยานิพนธ์ ที่สละเวลาตรวจสอบและให้คำแนะนำสำหรับวิทยานิพนธ์ฉบับนี้เพื่อความสมบูรณ์ยิ่งขึ้น

งานวิจัยนี้ได้รับการสนับสนุนจากหน่วยปฏิบัติการวิจัยโครงข่ายไร้สาย และอินเทอร์เน็ตอนาคต (Wireless Network and Future Internet Research Unit) กองทุนรัชดาภิเษกสมโภช จุฬาลงกรณ์มหาวิทยาลัย

ขอขอบคุณกลุ่มวิจัยโครงข่ายไร้สายและอินเทอร์เน็ตอนาคตภายใต้การดูแลของรองศาสตราจารย์ ดร. เชาวน์ดิศ อัศวกุล และผู้ช่วยศาสตราจารย์ ดร. ชัยเชษฐ์ สายวิจิตร ที่จัดกิจกรรมเพื่อส่งเสริมการเรียน การทำงาน และทักษะเพื่อเป็นผู้วิจัยที่มีประสิทธิภาพที่ดียิ่งขึ้น

ขอขอบคุณทุนอินทัชโดยบริษัท ไทยคม จำกัด (มหาชน) สำหรับการสนับสนุนทุนการศึกษาตลอดหลักสูตร

ขอขอบคุณเพื่อน พี่ น้องนักวิจัย เจ้าหน้าที่ บุคลากร และคณาจารย์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้คำแนะนำ ความช่วยเหลือในเรื่องต่าง ๆ และการสนับสนุนที่ดีเสมอมา

สุดท้ายนี้ขอขอบคุณครอบครัวของผู้วิจัย ซึ่งได้ให้การสนับสนุนและเป็นกำลังใจให้แก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สารบัญ

หน้า

บทคัดย่อภาษาไทย . . . . . ง
บทคัดย่อภาษาอังกฤษ . . . . . จ
กิตติกรรมประกาศ . . . . . ฉ
สารบัญ . . . . . ช
สารบัญตาราง . . . . . ญ
สารบัญรูป . . . . . ฎ
บทที่
1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของงานวิทยานิพนธ์ 3
1.3 ขอบเขตวิทยานิพนธ์ 4
1.4 ขั้นตอนและวิธีการดำเนินงาน 4
1.5 ประโยชน์ที่คาดว่าจะได้รับ 4
1.6 ประมวลวิทยานิพนธ์ 5
2 ทฤษฎีพื้นฐาน 6
2.1 โปรแกรมโอเพนสแต็ก 6
2.2 โปรแกรมโอเพนบาตง 7
2.2.1 องค์ประกอบที่สำคัญของมาตรฐานเอ็นเอฟวีมาโน 8
2.2.2 การทำงานของโปรแกรมโอเพนบาตงเบื้องต้น 10
2.3 การทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M สำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคาร 12
2.4 การพัฒนาการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite สำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคาร 13
2.5 การพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 13
3 โครงสร้างและการทำงานของระบบ 15
3.1 โครงสร้างของระบบที่นำเสนอ 15
3.2 ส่วนประกอบของระบบ 16
3.3 การทำงานของระบบ 18
3.3.1 การสร้างสภาพแวดล้อมให้กับโปรแกรมโอเพนสแต็ก 19
3.3.2 การลงทะเบียนโนดโอเพนสแต็กในโปรแกรมโอเพนบาตง 23
3.3.3 การทำงานร่วมกันระหว่างโปรแกรมโอเพนบาตงและโอเพนสแต็ก 24
4 การสร้างและการทดสอบฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์ 26
4.1 ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 26

บทที่ หน้า

4.1.1 การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 26
4.1.2 การประเมินฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 29
4.1.3 การประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอล 31
4.1.4 การประเมินความสามารถการปรับจำนวนทรัพยากรแบบอัตโนมัติของแพลตฟอร์มโอเพนเอ็มทีซีโดยใช้โปรแกรมโอเพนบาตง 32
4.1.5 การประเมินสมรรถนะการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอล 34
4.2 ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 34
4.2.1 การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 34
4.2.2 การประเมินการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 37
4.3 ฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 38
4.3.1 การสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 38
4.3.2 การแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศภายในอาคารผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบ 40
4.3.3 การตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ 40
5 ผลการสร้างและการทดสอบฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์ 42
5.1 ผลการทดสอบฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 42
5.1.1 ผลการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 42
5.1.2 ผลการประเมินฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 43
5.1.3 ผลการประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอล 45
5.1.4 ผลการประเมินความสามารถการปรับจำนวนทรัพยากรแบบอัตโนมัติของแพลตฟอร์มโอเพนเอ็มทีซีโดยใช้โปรแกรมโอเพนบาตง 46
5.1.5 ผลการประเมินสมรรถนะการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีหลังจากเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอล 53
5.2 ผลการทดสอบฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 57
5.2.1 ผลการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 57

บทที่ หน้า

5.2.2 ผลการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 57
5.3 ผลการทดสอบฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 59
5.3.1 ผลการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 59
5.3.2 ผลการแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศภายในอาคารผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบ 59
5.3.3 ผลการประเมินการตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ . . 61
6 บทสรุปและข้อเสนอแนะ 64
6.1 บทสรุป 64
6.2 ข้อเสนอแนะ 65
รายการอ้างอิง 66
ภาคผนวก 69
ประวัติผู้เขียนวิทยานิพนธ์ 119

สารบัญตาราง

หน้า

ตารางที่ 3.1 รายละเอียดของส่วนประกอบหลักที่ใช้สร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคาร 16
ตารางที่ 3.2 รายละเอียดของอุปกรณ์ที่ใช้สร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 18
ตารางที่ 4.1 ขั้นตอนการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 28
ตารางที่ 4.2 ขั้นตอนการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 36
ตารางที่ 4.3 ขั้นตอนการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 39

สารบัญรูป

หน้า

รูปที่ 2.1 องค์ประกอบที่สำคัญของโปรแกรมโอเพนสแต็กที่เกี่ยวข้องกับวิทยานิพนธ์นี้ 6
รูปที่ 2.2 สถาปัตยกรรมเอ็นเอฟวีมาโน [24] 8
รูปที่ 2.3 โครงสร้างการทำงานของโปรแกรมโอเพนบาตงตามมาตรฐานเอ็นเอฟวีมาโน [22] . . . 11
รูปที่ 2.4 ขั้นตอนการสร้างฟังก์ชันเสมือนตามมาตรฐาน ETSI MANO B.3 [24] 12
รูปที่ 2.5 แผนภาพการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 12
รูปที่ 2.6 แผนภาพการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 13
รูปที่ 2.7 แผนภาพการทำงานของโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 14
รูปที่ 3.1 โครงสร้างการเชื่อมต่อเพื่อสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารและหน่วยเก็บข้อมูลของระบบ CU-BEMS . . 15
รูปที่ 3.2 หน่วยเก็บข้อมูลของระบบ CU-BEMS 17
รูปที่ 3.3 เครื่องคอมพิวเตอร์สำหรับติดตั้งโปรแกรมโอเพนสแต็ก 17
รูปที่ 3.4 เครื่องคอมพิวเตอร์สำหรับติดตั้งโปรแกรมโอเพนบาตง 17
รูปที่ 3.5 อุปกรณ์จัดเส้นทางตราสินค้า Edimax BR-6208AC 17
รูปที่ 3.6 เครื่องปรับอากาศ MITSUBISHI 18
รูปที่ 3.7 โนด ECHONET Lite 18
รูปที่ 3.8 อุปกรณ์จัดเส้นทางตราสินค้า Tenda W309R+ 18
รูปที่ 3.9 ผลการบรรจุไฟล์ระบบฏิบัติการ Ubuntu cloud 16.04 LTS ที่ถูกปรับแต่งเข้าสู่โปรแกรมโอเพนสแต็ก 20
รูปที่ 3.10 ผลการสร้างโครงข่ายส่วนบุคคลเสมือนเพื่อใช้ติดต่อระหว่างเครื่องเสมือน 20
รูปที่ 3.11 ผลการสร้างกุญแจเพื่อให้ผู้ใช้สามารถเข้าถึงเครื่องเสมือน 21
รูปที่ 3.12 ผลการสร้างข้อกำหนดความปลอดภัยของกลุ่มเครื่องเสมือนให้กับเครื่องเสมือน . . . 21
รูปที่ 3.13 ผลการสร้างบริการ Load-Balancing-as-a-Service ในโปรแกรมโอเพนสแต็ก 22
รูปที่ 3.14 ผลการสร้างโครงข่ายส่วนบุคคลเสมือนเพื่อใช้ติดต่อกับโครงข่าย ECHONET Lite . . 22
รูปที่ 3.15 ผลการสร้างโครงข่ายส่วนบุคคลเสมือนเพื่อใช้ติดต่อกับโครงข่าย CU-BEMS 23
รูปที่ 3.16 ข้อมูลของโนดโอเพนสแต็กที่ใช้ลงทะเบียนในรูปแบบ JSON 23
รูปที่ 3.17 ผังเวลาวิธีการสร้างฟังก์ชันเสมือนขึ้นบนแพลตฟอร์มโอเพนสแต็กด้วยโปรแกรมโอเพนบาตง [22] 24
รูปที่ 4.1 โครงสร้างการเชื่อมต่อของการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 27
รูปที่ 4.2 ผังเวลาการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 29
รูปที่ 4.3 แผนผังแสดงตำแหน่งตัวรับรู้ที่ใช้เพื่อทดสอบฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 30

หน้า

รูปที่ 4.4 ผังเวลาการถ่ายโอนข้อมูลของตัวรับรู้จากระบบ CU-BEMS ไปยังแพลตฟอร์มโอเพนเอ็มทีซีแบบทันที 30
รูปที่ 4.5 ผังเวลาการถ่ายโอนข้อมูลตัวรับรู้เลียนแบบไปยังแพลตฟอร์มโอเพนเอ็มทีซีเพื่อทดสอบสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอล 32
รูปที่ 4.6 ข้อกำหนดการปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลโดยโปรแกรมโอเพนบาตง . . . 33
รูปที่ 4.7 ข้อกำหนดการปรับเพิ่มจำนวนเครื่องเสมือนมองโกดีบีโดยโปรแกรมโอเพนบาตง . . . 33
รูปที่ 4.8 โครงสร้างการเชื่อมต่อของการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 35
รูปที่ 4.9 ไฟล์ข้อมูลรายละเอียดบริการโครงข่ายฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 35
รูปที่ 4.10 การกำหนดการไหลของข้อมูลของบริดจ์ brvlan เพื่อเชื่อมต่อโครงข่ายเสมือน private_echonet และโครงข่าย ECHONET Lite 36
รูปที่ 4.11 ผังการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite 37
รูปที่ 4.12 โครงสร้างการเชื่อมต่อของการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 38
รูปที่ 4.13 การกำหนดการไหลของข้อมูลของบริดจ์ brvlan เพื่อเชื่อมต่อโครงข่ายเสมือน private_cubems1 และโครงข่ายของระบบ CU-BEMS 39
รูปที่ 4.14 ขั้นตอนการตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ 41
รูปที่ 5.1 ไฟล์บันทึกขั้นตอนการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 42
รูปที่ 5.2 ไฟล์บันทึกขั้นตอนการยุติการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M 43
รูปที่ 5.3 ข้อมูลตัวรับรู้การเคลื่อนไหวที่ถูกบันทึกในหน่วยเก็บข้อมูล CU-BEMS 44
รูปที่ 5.4 ผลการส่งข้อมูล subscription เพื่อติดตามการเปลี่ยนแปลงข้อมูลของเอกสารที่ฐานข้อมูลมองโกดีบี 44
รูปที่ 5.5 ข้อมูลจากตัวรับรู้การเคลื่อนไหวของคนในระบบ CU-BEMS ที่เครื่องเสมือนเอ็นไอพี 44
รูปที่ 5.6 ข้อมูลตัวรับรู้การเคลื่อนไหวขของคนที่ถูกส่งไปยังเครื่องเสมือนเอ็นเอสซีแอลเพื่อบันทึกลงฐานข้อมูลมองโกดีบีของแพลตฟอร์มโอเพนเอ็มทีซี 45
รูปที่ 5.7 ข้อมูลตัวรับรู้การเคลื่อนไหวของคนที่ถูกบันทึกในฐานข้อมูลมองโกดีบีที่เครื่องเสมือนมองโกดีบี 45
รูปที่ 5.8 ข้อมูลตัวรับรู้ที่ถูกส่งกลับไปยังเครื่องเสมือนเอ็นไอพีด้วยโพรโทคอล NOTIFY 45
รูปที่ 5.9 ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเมื่อมีการเปลี่ยนแปลงความถี่การส่งข้อมูลเลียนแบบ 46
รูปที่ 5.10 ค่าเฉลี่ยการบันทึกข้อมูลเลียนแบบต่อนาทีของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อมีการเปลี่ยนแปลงความถี่การส่งข้อมูลเลียนแบบ 46
รูปที่ 5.11 ผังเวลาแสดงการเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติซึ่งควบคุมโดยโปรแกรมโอเพนบาตง 47
รูปที่ 5.12 ระดับหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลและเครื่องเสมือนมองโกดีบี 48

หน้า

รูปที่ 5.13 หมายเลขไอพีของเครื่องเสมือนเอ็นเอสซีแอลที่ถูกเพิ่มเข้าสู่บริการ LBaaS 48
รูปที่ 5.14 ผังเวลาแสดงการลดจำนวนของเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติซึ่งควบคุม โดยโปรแกรมโอเพนบาตง 49
รูปที่ 5.15 ระดับหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลเมื่อทำงานร่วมกับบริการ LBaaS ของโปรแกรมโอเพนสแต็ก 49
รูปที่ 5.16 ระดับหน่วยประมวลผลกลางของเครื่องเสมือนมองโกดีบีเมื่อเครื่องเสมือนเอ็นเอสซีแอล ลดจำนวนลงแบบอัตโนมัติ 50
รูปที่ 5.17 การจัดเก็บข้อมูลในฐานข้อมูลมองโกดีบีเมื่อมีเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่อง 50
รูปที่ 5.18 การจัดเก็บข้อมูลในฐานข้อมูลมองโกดีบีเมื่อมีเครื่องเสมือนมองโกดีบีจำนวน 2 เครื่อง 51
รูปที่ 5.19 ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่ง 51
รูปที่ 5.20 ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่สอง 51
รูปที่ 5.21 การถ่ายโอนข้อมูลจากเครื่องเสมือนมองโกดีบีเครื่องแรกไปสู่เครื่องเสมือนมองโกดีบี เครื่องที่สอง 52
รูปที่ 5.22 การรับถ่ายโอนข้อมูลจากเครื่องเสมือนมองโกดีบีเครื่องแรกไปสู่เครื่องเสมือนมองโก ดีบีเครื่องที่สอง 52
รูปที่ 5.23 การจัดเก็บข้อมูลในฐานข้อมูลมองโกดีบีเมื่อมีเครื่องเสมือนมองโกดีบีจำนวน 2 เครื่อง หลังการทดสอบส่งข้อมูลครั้งที่สอง 53
รูปที่ 5.24 ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่ง หลังการทดสอบส่ง ข้อมูลครั้งที่สอง 53
รูปที่ 5.25 ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่สอง หลังการทดสอบ ส่งข้อมูลครั้งที่สอง 53
รูปที่ 5.26 ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 1 เครื่องและเครื่องเสมือนมองโกดีบีเมื่อเปลี่ยนแปลงความถี่การส่งข้อมูล 54
รูปที่ 5.27 ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 2 เครื่องและเครื่องเสมือนมองโกดีบีเมื่อเปลี่ยนแปลงความถี่การส่งข้อมูล 55
รูปที่ 5.28 ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 3 เครื่องและเครื่องเสมือนมองโกดีบีเมื่อเปลี่ยนแปลงความถี่การส่งข้อมูล 55
รูปที่ 5.29 ค่าเฉลี่ยของการบันทึกข้อมูลเลียนแบบของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อเปลี่ยน แปลงจำนวนเครื่องเสมือนเอ็นเอสซีแอลและเปลี่ยนแปลงแปลงความถี่การส่งข้อมูล . 56
รูปที่ 5.30 ประสิทธิภาพการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อใช้เครื่องเสมือนเอ็นเอสซี แอลทำงานร่วมกับบริการ LBaaS เทียบกับเครื่องเสมือนเอ็นเอสซีแอลแบบดั้งเดิม . . 56
รูปที่ 5.31 การเชื่อมต่อโครงข่ายของเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์บนแพลตฟอร์ม โอเพนสแต็ก 57
รูปที่ 5.32 อุณหภูมิของห้องคลัสเตอร์ระหว่างวันที่ไม่เปิดใช้งานและวันที่เปิดใช้งานฟังก์ชันเสมือน 58
รูปที่ 5.33 อุณหภูมิของห้องคลัสเตอร์และสถานะของเครื่องปรับอากาศเมื่อเปิดใช้งานฟังก์ชัน เสมือน 58
รูปที่ 5.34 การเชื่อมต่อโครงข่ายของเครื่องเสมือนสำหรับการวิเคราะห์ข้อมูลบนแพลตฟอร์ม โอเพนสแต็ก 59

หน้า

รูปที่ 5.35 ค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศบริเวณศูนย์เชี่ยวชาญเฉพาะด้านเทคโนโลยีโทรคมนาคม 60
รูปที่ 5.36 ค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศบริเวณห้องพักอาจารย์และห้องเรียนชั้น 13 60
รูปที่ 5.37 การนำเสนอค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศผ่านทางโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟท์ชั้น 13 61
รูปที่ 5.38 จำนวนการโบกมือของผู้ใช้งานผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟต์ชั้น 13 61
รูปที่ 5.39 จำนวนผู้ใช้งานที่กระตุ้นให้โปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟต์ชั้น 13 เข้าสู่แคนวาสปริยาย 62
รูปที่ 5.40 สถานะการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟต์ชั้น 13 . . . 62
รูปที่ 5.41 การส่งข้อความแจ้งเตือนผ่านทางโปรแกรมประยุกต์ไลน์ 63
รูปที่ 5.42 การส่งข้อความแจ้งเตือนผ่านทางอีเมล์ 63

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีเสมือน (virtualization technology) [1] เข้ามามีบทบาทสำคัญอย่างมากในอุตสาหกรรมคอมพิวเตอร์ จุดมุ่งหมายของเทคโนโลยีเสมือนคือการแบ่งปันการใช้งานทรัพยากรคอมพิวเตอร์ในรูปของทรัพยากรเสมือนและใช้งานทรัพยากรเสมือนเหล่านี้แยกจากกัน ด้วยเหตุนี้ทำให้นักพัฒนาสามารถสร้างบริการใหม่ขึ้นมาได้อย่างรวดเร็วด้วยการใช้งานทรัพยากรเสมือน ดังนั้นไม่เพียงแต่บริษัทขนาดใหญ่ที่จะสร้างบริการรูปแบบใหม่ให้เกิดขึ้นมา แต่กลุ่มนักพัฒนาขนาดเล็กและรวมถึงนักวิจัยในมหาวิทยาลัยก็สามารถสร้างบริการใหม่ ๆ ให้เกิดขึ้นมาด้วยการใช้เทคโนโลยีเสมือนเป็นโครงสร้างพื้นฐานสำหรับการพัฒนาบริการขึ้นมา และด้วยความสามารถของเทคโนโลยีเสมือนที่มีความยืดหยุ่น ทำให้ผู้ดูแลระบบสามารถปรับขนาดของทรัพยากรเสมือนให้เหมาะสมกับปริมาณการใช้งาน และสำหรับบริการที่ไม่ประสบความสำเร็จ ผู้ดูแลระบบสามารถปรับลดขนาดของทรัพยากรเสมือนลงมาเพื่อลดภาระค่าใช้จ่าย หรือแบ่งปันทรัพยากรเสมือนที่ไม่ได้ใช้งานไปใช้กับโครงการอื่นได้ รวมทั้งผู้ดูแลระบบสามารถจัดการทรัพยากรเสมือนเพื่อให้เกิดการกระจายการใช้งานทรัพยากรเสมือนซึ่งทำให้ผู้ดูแลระบบสามารถจัดสรรการใช้งานทรัพยากรคอมพิวเตอร์ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น นอกจากนี้เทคโนโลยีเสมือนทำให้การย้ายเครื่องเสมือนไปยังทรัพยากรคอมพิวเตอร์อื่นสามารถทำได้สะดวกมากขึ้น ทำให้สามารถลดระยะเวลาที่ไม่สามารถให้บริการได้อันเนื่องจากปัญหาที่เกิดจากการขัดข้องของทรัพยากรคอมพิวเตอร์จากเหตุสุดวิสัยหรือภัยพิบัติ โดยที่เทคโนโลยีเสมือนสามารถแบ่งออกเป็น 2 ประเภทหลัก ได้แก่ เทคโนโลยีฮาร์ดแวร์เสมือน เช่น เทคโนโลยีเครื่องเสมือน (virtual machine) [2] โครงข่ายส่วนบุคคลเสมือน (virtual private network, VPN) [3] เป็นต้น และเทคโนโลยีฟังก์ชันเสมือนซึ่งเป็นการสร้างโปรแกรมประยุกต์ให้สามารถทำงานได้บนเทคโนโลยีเครื่องเสมือน เช่น เทคโนโลยีฟังก์ชันโครงข่ายเสมือน (network function virtualization, NFV) [4] ซึ่งเป็นการนำฟังก์ชันการทำงานของระบบโครงข่าย เช่น พร็อกซีเกตเวย์ ไฟร์วอลล์ หรือระบบบริการชื่อโดเมนติดตั้งบนแพลตฟอร์มเสมือน ทำให้ฟังก์ชันของระบบโครงข่ายเหล่านี้ไม่จำเป็นต้องติดตั้งบนอุปกรณ์คอมพิวเตอร์เฉพาะ ส่งผลให้เกิดความยืดหยุ่นสำหรับการใช้งานเพิ่มมากขึ้น ขณะที่ระบบโครงสร้างพื้นฐานแบบแบ่งปันทรัพยากรผ่านโครงข่าย หรือเทคโนโลยีคลาวด์ (cloud computing technology) ก็มีพื้นฐานมาจากเทคโนโลยีเสมือนเช่นกัน

สถาบันมาตรฐานและเทคโนโลยีแห่งชาติ (National Institute of Standards and Technology, NIST) [5] นิยามเทคโนโลยีคลาวด์เป็นรูปแบบการเข้าถึงบริการแบบแบ่งปันทรัพยากรคอมพิวเตอร์ เช่น พื้นที่จัดเก็บข้อมูล โครงข่าย เครื่องบริการ รวมทั้งโปรแกรมประยุกต์จากทุกสถานที่ตามความต้องการของผู้ใช้งาน ซึ่งเทคโนโลยีนี้มีลักษณะสำคัญ 5 ประการคือ ผู้ใช้งานสามารถเลือกใช้บริการต่าง ๆ ได้ด้วยตัวเอง ผู้ใช้งานสามารถเข้าใช้บริการได้จากโครงข่ายต่าง ๆ ผ่านอุปกรณ์ที่แตกต่างกันได้ เป็นบริการที่สามารถรวบรวมทรัพยากรคอมพิวเตอร์จากหลายแห่งไว้ด้วยกัน บริการมีความยืดหยุ่นสูงเพื่อให้ผู้ใช้บริการเลือกใช้ทรัพยากรเสมือนได้เหมาะสมตามความต้องการ และเป็นบริการที่สามารถตรวจสอบการใช้งานได้ โดยเทคโนโลยีคลาวด์ได้แบ่งประเภทของบริการออกเป็น 3 ประเภทหลักคือ การให้บริการซอฟแวร์ (software as a service, SaaS) การให้บริการแพลตฟอร์ม (platform as a service, PaaS) และการบริการโครงสร้างพื้นฐาน (infrastructure as a service, IaaS)

ในขณะที่อุปกรณ์อัจฉริยะได้ถูกผลิตขึ้นมาเป็นจำนวนมากเพื่อนำไปใช้ในอุตสาหกรรมต่าง ๆ ซึ่งอุปกรณ์อัจฉริยะเหล่านี้สามารถเชื่อมต่อกันได้เพื่อเก็บสะสมข้อมูล ประมวลผลข้อมูล และนำเสนอข้อมูลในรูปแบบต่าง ๆ เทคโนโลยีนี้ถูกเรียกว่าอินเทอร์เน็ตของสรรพสิ่ง (internet of things, IoT) [6] โดยคาดว่าจะมีอุปกรณ์อัจฉริยะประมาณ 25,000 ล้านชิ้นเชื่อมต่อกับอินเทอร์เน็ตภายในปี ค.ศ. 2015 และจะเพิ่มเป็น 50,000 ล้านชิ้นภายในปี ค.ศ. 2020 [7] ซึ่งอุปกรณ์อัจฉริยะจำนวนหนึ่งจะถูกนำมาใช้ในระบบจัดการพลังงานไฟฟ้าภายในอาคาร ทำให้นักวิจัยสามารถนำข้อมูลเหล่านี้มาวิเคราะห์ และพัฒนาให้เกิดการใช้พลังงานไฟฟ้าอย่างมีประสิทธิภาพ

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยออกแบบและสร้างระบบ CU-BEMS [8] ซึ่งเป็นระบบจัดการพลังงานไฟฟ้าภายในอาคารภายใต้มาตรฐาน IEEE1888 [9] ที่ถูกกำหนดโดยสถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers, IEEE) ให้เป็นมาตรฐานเปิดสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคาร โดยกำหนดรูปแบบข้อมูลเพื่อใช้สำหรับโต้ตอบระหว่างระบบกับอุปกรณ์อัจฉริยะ เพื่อให้เกิดเป็นวิธีการเก็บข้อมูลการใช้พลังงานไฟฟ้า โดยโครงการ CU-BEMS มีวัตถุประสงค์เพื่อสังเกตการใช้พลังงานไฟฟ้า และสภาพแวดล้อมภายในอาคารซึ่งประกอบไปด้วยมาตรวัดอัจฉริยะ [10] และตัวรับรู้สิ่งแวดล้อมต่าง ๆ ได้แก่ ตัวรับรู้อุณหภูมิ ตัวรับรู้แสง ตัวรับรู้ความชื้นสัมพัทธ์ และตัวรับรู้การเคลื่อนไหวของมนุษย์ รวมถึงการนำข้อมูลนำเสนอผ่านเว็บไซต์ และจอแสดงผลภายในอาคาร [11][12]

ขณะที่สถาบันมาตรฐานโทรคมนาคมยุโรป (European Telecommunication Standards Institute, ETSI) ได้กำหนดมาตรฐาน ETSI Machinet-to-Machine communications (ETSI M2M) [13] ซึ่งเป็นมาตรฐานการสื่อสารระหว่างอุปกรณ์ซึ่งไม่จำเป็นต้องพึ่งพามนุษย์ โดยมีวัตถุประสงค์ที่จะสร้างการสื่อสาร และการตัดสินใจแบบอัตโนมัติ สถาบัน Fraunhofer Institute for Open Communication Systems (Fraunhofer FOKUS) และมหาวิทยาลัยแห่งเบอร์ลิน (Technical University of Berlin) ร่วมกันพัฒนาแพลตฟอร์มโอเพนเอ็มทีซี (OpenMTC) [14] ซึ่งเป็นมิดเดิลแวร์สำหรับการสื่อสารของอุปกรณ์ และมีรูปแบบสอดคล้องกับมาตรฐานการสื่อสารระหว่างอุปกรณ์ เช่น ETSI M2M oneM2M งานวิจัย [15] นำแพลตฟอร์มโอเพนเอ็มทีซีประยุกต์ใช้ร่วมกับเทคโนโลยีคลาวด์เพื่อนำไปใช้ในโครงการการทดสอบเมืองอัจฉริยะที่มีการสื่อสารระหว่างอุปกรณ์ โดยใช้โปรแกรมโอเพนสแต็ก (OpenStack) [16] ซึ่งเป็นโปรแกรมประยุกต์สำหรับการสร้างระบบโครงสร้างพื้นฐานแบบแบ่งปันทรัพยากรผ่านโครงข่าย โดยมีเป้าหมายเพื่อแลกเปลี่ยนข้อมูลสภาพแวดล้อม เช่น คุณภาพอากาศ อุณหภูมิ และการคมนาคม นอกจากนี้แพลตฟอร์มโอเพนเอ็มทีซีนี้สามารถทำงานร่วมกับมาตรฐานการสื่อสารระหว่างอุปกรณ์อื่น ๆ ได้โดยใช้อินเตอร์เวิร์กกิงพร็อกซี (interworking proxy) เพื่อเปลี่ยนแปลงรูปแบบข้อมูลจากมาตรฐานอื่นให้เป็นรูปแบบข้อมูลตามมาตรฐาน ETSI M2M และเนื่องจากจุฬาลงกรณ์มหาวิทยาลัยเข้าร่วมเป็นส่วนหนึ่งของโครงการมหาวิทยาลัยสำหรับอินเทอร์เน็ตในอนาคต (university for future internet, UNIFI) [17] และด้วยความร่วมมือจากสถาบัน Fraunhofer FOKUS และมหาวิทยาลัยแห่งเบอร์ลิน ทำให้เกิดการพัฒนาการทำงานร่วมกันระหว่างโครงการ CU-BEMS ที่ใช้มาตรฐาน IEEE1888 และแพลตฟอร์มโอเพนเอ็มทีซีภายใต้มาตรฐาน ETSI M2M ขึ้น งานวิจัย [18] นำเสนอการพัฒนาการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M โดยการพัฒนาอินเตอร์เวิร์กกิงพร็อกซีสำหรับมาตรฐาน IEEE1888 บนแพลตฟอร์มโอเพนเอ็มทีซี และงานวิจัย [19] นำเสนอการพัฒนาการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M เพื่อให้เกิดการตอบสนองของข้อมูลในทันทีที่มีการเปลี่ยนแปลง นอกจากนี้ยังมีงานวิจัยที่พัฒนาทำให้ระบบ CU-BEMS สามารถทำงานร่วมกับมาตรฐานการสื่อสารระหว่างอุปกรณ์อื่น เช่น งานวิจัย [20] นำเสนอการทำงานร่วมกันแบบทันที

ระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite และงานวิจัย [21] นำเสนอการพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 ซึ่งงานวิจัยเหล่านี้ [19][20][21] มีขั้นตอนการติดตั้งระบบเพื่อนำไปใช้งานที่ซับซ้อน และจำเป็นต้องพิจารณาในเรื่องการติดตั้งโครงข่าย และความปลอดภัยของระบบ รวมถึงยังไม่ได้พิจารณาในเรื่องการขยายระบบเพื่อรองรับปริมาณอุปกรณ์อัจฉริยะที่จะมีปริมาณเพิ่มมากขึ้นในอนาคต และการเข้าถึงข้อมูลเหล่านี้มีแนวโน้มสูงขึ้น ดังนั้นเพื่อให้ผู้ดูแลระบบสามารถนำงานวิจัยเหล่านี้มาใช้ได้อย่างสะดวก จึงควรพัฒนาระบบจัดการพลังงานไฟฟ้าภายในอาคารให้สามารถติดตั้งได้อย่างสะดวกรวดเร็ว ลดความผิดพลาดที่เกิดจากผู้ติดตั้งระบบ และรองรับการเพิ่มจำนวนขึ้นของอุปกรณ์อัจฉริยะได้อย่างอัตโนมัติ

วิทยานิพนธ์ฉบับนี้จึงขอนำเสนอการพัฒนา และการสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารด้วยเทคโนโลยีคลาวด์ โดยใช้โอเพนบาตง [22] ซึ่งเป็นโปรแกรมประสานฟังก์ชันโครงข่ายเสมือนสร้างฟังก์ชันเสมือนขึ้นบนแพลตฟอร์มโอเพนสแต็ก วิทยานิพนธ์ฉบับนี้ได้นำงานวิจัย [19] มาสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M นอกจากนี้นำงานวิจัย [20] มาพัฒนาเพื่อควบคุมการทำงานของเครื่องปรับอากาศที่ใช้มาตรฐาน ECHONET Lite ด้วยข้อมูลสภาพแวดล้อมจากระบบ CU-BEMS รวมถึงนำงานวิจัย [21] มาประยุกต์เพื่อนำเสนอค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศในอาคารบริเวณชั้น 13 อาคารเจริญวิศวกรรมด้วยโปรแกรมสร้างมโนภาพเชิงโต้ตอบ รวมถึงนำข้อมูลการใช้งานโปรแกรมสร้างมโนภาพเชิงโต้ตอบจากระบบ CU-BEMS มาวิเคราะห์เพื่อแก้ไขปัญหาที่ผู้ใช้งานโปรแกรมสร้างมโนภาพเชิงโต้ตอบไม่สามารถใช้งานแคนวาสอื่นได้นอกจากแคนวาสปริยาย และแจ้งเตือนผู้ดูแลระบบเมื่อเกิดปัญหา งานวิจัย [19][20][21] ถูกสร้างเป็นฟังก์ชันเสมือนบนแพลตฟอร์มโอเพนสแต็ก โดยผู้ใช้สามารถกำหนดค่าพารามิเตอร์ต่าง ๆ ในรูปแบบ JSON (JavaScript object notation) และสามารถสร้างฟังก์ชันเสมือนผ่านทางเว็บเบราว์-เซอร์ นอกจากนี้เพื่อให้ระบบจัดการพลังงานไฟฟ้าภายในอาคารใช้ทรัพยากรคอมพิวเตอร์ได้เหมาะสมกับปริมาณของข้อมูลที่เข้าสู่ระบบ วิทยานิพนธ์ฉบับนี้จึงได้ออกแบบ และสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารที่มีความสามารถขยาย และปรับลดทรัพยากรของระบบแบบอัตโนมัติให้สอดคล้องกับปริมาณงานของระบบ และวัดประสิทธิภาพของระบบที่สร้างขึ้น โดยนำแพลตฟอร์มโอเพนเอ็มทีซีมาใช้สำหรับการทดสอบ

1.2 วัตถุประสงค์ของงานวิทยานิพนธ์

1. ประเมินความเป็นไปได้ของการสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารบนแพลตฟอร์มโอเพนสแต็กด้วยโปรแกรมโอเพนบาตง โดยนำการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M การทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite และการพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 และนำผลมาแสดงที่โปรแกรมสร้างมโนภาพเชิงโต้ตอบ

2. ออกแบบ และสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารที่มีความสามารถขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติเพื่อให้สอดคล้องกับปริมาณงานของระบบ และเพิ่มพื้นที่สำหรับจัดเก็บข้อมูลแบบอัตโนมัติ

1.3 ขอบเขตวิทยานิพนธ์

1. ออกแบบ และสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์ ได้แก่ การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite และฟังก์ชันเสมือนการพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 และนำผลมาแสดงที่โปรแกรมสร้างมโนภาพเชิงโต้ตอบ
2. ออกแบบ และสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารที่มีความสามารถขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติ โดยใช้โปรแกรมโอเพนบาตงร่วมกับโปรแกรมโอเพนสแต็ก และใช้หน่วยประมวลผลกลางเฉลี่ยและพื้นที่จัดเก็บข้อมูลเป็นตัวชี้วัด
3. ทดสอบ และวัดประสิทธิภาพของระบบจัดการพลังงานไฟฟ้าภายในอาคารที่มีความสามารถขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติ

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษามาตรฐาน IEEE1888 มาตรฐาน ETSI M2M มาตรฐาน ECHONET Lite โปรแกรมโอเพนสแต็กและโปรแกรมโอเพนบาตง และศึกษาการทำงานร่วมกันของทั้งสองโปรแกรม
2. สร้างฟังก์ชันเสมือนขึ้นบนแพลตฟอร์มโอเพนสแต็กให้สามารถทำงานตามหน้าที่ที่กำหนดโดยใช้โปรแกรมโอเพนบาตงควบคุมขั้นตอนการติดตั้ง และปรับแต่งพารามิเตอร์เพื่อให้สามารถทำงานร่วมกับระบบ CU-BEMS แบบทันทีได้
3. ศึกษาโปรแกรมแซบบิกซ์ (Zabbix) [23] เพื่อใช้ตรวจสอบเครื่องเสมือนและนำไปใช้ร่วมกับโปรแกรมโอเพนบาตงเพื่อเป็นตัวชี้วัดในการขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติ
4. ออกแบบ และสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารที่มีความสามารถขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติได้ และควบคุมขั้นตอนการติดตั้ง รวมถึงขั้นตอนการขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติด้วยโปรแกรมโอเพนบาตง
5. ทดสอบ วัดประสิทธิภาพ และสรุปผลการทำงานจากระบบที่ได้พัฒนาขึ้น
6. เผยแพร่ผลการวิจัยลงในบทความ และจัดทำเล่มวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารบนแพลตฟอร์มโอเพนสแต็ก ได้แก่ การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite และฟังก์ชัน

เสมือนการพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 และนำผลมาแสดงผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยโปรแกรมโอเพนบาตงได้

2. สามารถสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารที่มีความสามารถขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติ

1.6 ประมวลวิทยานิพนธ์

บทที่ 1 บทนำ: อธิบายถึงความเป็นมา ความสำคัญ และประโยชน์ของการพัฒนาฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์ วัตถุประสงค์ของงานวิทยานิพนธ์ ขอบเขตวิทยานิพนธ์ ขั้นตอน และวิธีการดำเนินงาน

บทที่ 2 ทฤษฎีพื้นฐาน: อธิบายองค์ประกอบและการทำงานของโปรแกรมโอเพนสแต็ก และโปรแกรมโอเพนบาตง รวมถึงการทำงานร่วมกันของทั้งสองโปรแกรม และการทำงานประสานข้อมูลในระบบจัดการพลังงานไฟฟ้าภายในอาคารที่ถูกนำมาสร้างเป็นฟังก์ชันเสมือน

บทที่ 3 โครงสร้างและการทำงานของระบบ: อธิบายโครงสร้าง ส่วนประกอบ และวิธีการสร้างฟังก์ชันเสมือนทำงานด้วยโปรแกรมโอเพนบาตงบนแพลตฟอร์มโอเพนสแต็ก

บทที่ 4 การสร้างและการทดสอบฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์: กล่าวถึงวิธีการสร้าง และการทดสอบฟังก์ชันเสมือนที่สร้างขึ้น รวมถึงวิธีการขยาย และการปรับลดทรัพยากรของระบบแบบอัตโนมัติด้วยโปรแกรมโอเพนบาตง

บทที่ 5 ผลการสร้างและการทดสอบฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์: กล่าวถึงผลการสร้างและผลการทดสอบฟังก์ชันเสมือนที่สร้างขึ้น ประเมินประสิทธิภาพของการขยาย และการปรับลดทรัพยากรระบบแบบอัตโนมัติ

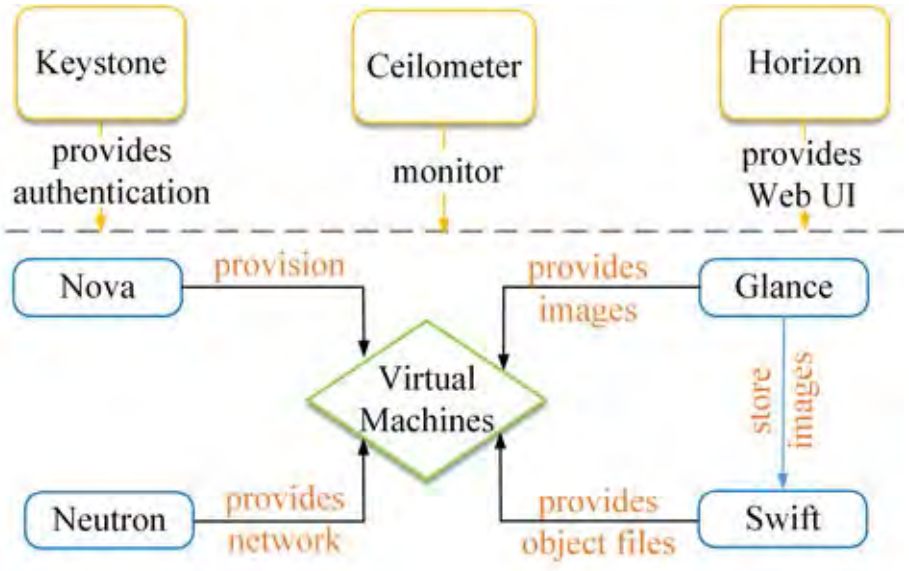
บทที่ 6 บทสรุปและข้อเสนอแนะ: สรุปผลการสร้างฟังก์ชันเสมือนที่สร้างขึ้นในวิทยานิพนธ์ฉบับนี้ และแนวทางการพัฒนางานวิจัยต่อไป

บทที่ 2
ทฤษฎีพื้นฐาน

2.1 โปรแกรมโอเพนสแต็ก

โปรแกรมโอเพนสแต็ก [16] เป็นซอฟแวร์สำหรับการสร้างระบบโครงสร้างพื้นฐานแบบแบ่งปันทรัพยากรผ่านโครงข่ายทั้งแบบส่วนบุคคลและแบบสาธารณะ โปรแกรมโอเพนสแต็กนี้ควบคุมแหล่งรวมทรัพยากรสำหรับการประมวลผล การจัดเก็บข้อมูล และการเชื่อมต่อโครงข่าย ผู้ใช้งานสามารถควบคุมการทำงานโปรแกรมโอเพนสแต็กผ่านทางเว็บเบราว์เซอร์ โอเพนสแต็ก เอพีไอและการป้อนคำสั่ง และเนื่องจากโปรแกรมโอเพนสแต็กเป็นโปรแกรมแบบโอเพนซอร์ส ส่งผลให้นักพัฒนาสามารถพัฒนาฟังก์ชันมากมายเพิ่มเข้ามาในโปรแกรมโอเพนสแต็ก เช่น ระบบฐานข้อมูล ระบบการเฝ้าสังเกตทรัพยากร ระบบบริการชื่อโดเมน เป็นต้น โครงการโอเพนสแต็กมีเป้าหมายทำให้เกิดการนำไปใช้งานได้ง่าย สามารถปรับขนาดได้ตามความเหมาะสมและมีฟังก์ชันที่พร้อมใช้งานตามความต้องการ

โปรแกรมโอเพนสแต็กเป็นซอฟแวร์ที่ให้บริการเทคโนโลยีคลาวด์ที่มีหลากหลายรูปแบบบริการให้ผู้ใช้งานสามารถเลือกใช้ตามความต้องการ ทำให้โปรแกรมโอเพนสแต็กแยกองค์ประกอบของซอฟแวร์ตามประเภทของการให้บริการเพื่อให้ผู้พัฒนาสามารถเลือกพัฒนาในด้านที่สนใจ และผู้ที่นำไปใช้งานได้เลือกเฉพาะบริการที่สนใจจะนำไปใช้ บริการที่ถูกสร้างขึ้นในวิทยานิพนธ์ฉบับนี้แสดงในรูปที่ 2.1 ได้แก่



รูปที่ 2.1: องค์ประกอบที่สำคัญของโปรแกรมโอเพนสแต็กที่เกี่ยวข้องกับวิทยานิพนธ์นี้

1. Keystone เป็นการให้บริการยืนยันตัวตนของโปรแกรมโอเพนสแต็ก ได้แก่ การอนุญาตให้ใช้บริการต่าง ๆ ของโปรแกรมโอเพนสแต็ก การจัดการบัญชีผู้ใช้งาน และสิทธิ์การใช้งานของผู้ใช้แต่ละราย นอกจากนี้ Keystone ทำหน้าที่จัดเตรียมรายการการเข้าถึงสำหรับแต่ละบริการของโปรแกรมโอเพนสแต็ก โดยแต่ละครั้งของการใช้บริการจะต้องยืนยันตัวตนด้วยโทเค็นใน

รูปแบบของตัวอักษรและตัวเลขผสมกัน และมีระยะเวลาการใช้งานที่จำกัด โดย Keystone ใช้ฐานข้อมูล MySQL สำหรับการเก็บข้อมูลของบริการ

2. Nova เป็นการให้บริการเครื่องเสมือนบนโปรแกรมโอเพนสแต็ก โดย Nova มีหน้าที่จัดการเครื่องเสมือนตั้งแต่การสร้าง การจัดกำหนดการ และการเลิกใช้งานตามความต้องการของผู้ใช้ ผู้ใช้งานสามารถสร้างเครื่องเสมือนขึ้นมาได้ตามจำนวนที่ถูกจัดสรรไว้ Nova สามารถทำงานร่วมกับเทคโนโลยีเครื่องเสมือนต่าง ๆ ได้แก่ KVM, VMware, Xen และ Hyper-V ทำให้โปรแกรมโอเพนสแต็กรองรับการใช้งานเครื่องเสมือนที่สร้างจากมาตรฐานอื่นได้ ซึ่ง Nova ถือเป็นส่วนสำคัญในการให้บริการโครงสร้างพื้นฐานแบบแบ่งปันทรัพยากรผ่านโครงข่าย

3. Glance เป็นการให้บริการไฟล์ระบบปฏิบัติการสำหรับติดตั้งลงบนเครื่องเสมือนบนโปรแกรมโอเพนสแต็ก โดยผู้ใช้งานสามารถอัพโหลดไฟล์ระบบปฏิบัติการและเรียกใช้ รวมทั้งสามารถบันทึกเครื่องเสมือนที่ถูกสร้างขึ้น และนำไปใช้งานบนโปรแกรมโอเพนสแต็กที่ถูกติดตั้งบนเครื่องคอมพิวเตอร์อื่นได้

4. Neutron เป็นการให้บริการโครงข่ายเสมือนสำหรับเครื่องเสมือนบนโปรแกรมโอเพนสแต็ก และสามารถเชื่อมต่อกับโครงข่ายจริงและอุปกรณ์โครงข่ายที่ใช้งานอยู่ โดยผู้ใช้งานสามารถสร้างโครงข่ายเสมือนเพื่อเชื่อมต่อเครื่องเสมือนภายในโครงการเดียวกัน รวมทั้งสามารถเชื่อมต่อกับโครงการอื่นบนโปรแกรมโอเพนสแต็กได้ นอกจากนี้ยังมีบริการอื่นที่เกี่ยวข้องกับการให้บริการโครงข่าย ได้แก่ การกำหนดกฎความปลอดภัยให้กับเครื่องเสมือน โดยผู้ใช้งานสามารถกำหนดหลักเกณฑ์ความปลอดภัย เช่น ระบุช่องทางที่อนุญาตให้สามารถติดต่อกับเครื่องเสมือนได้ รวมทั้งมีบริการเสริมอื่นได้แก่ โครงข่ายส่วนบุคคลเสมือน ไฟร์วอล์ เป็นต้น

5. Swift เป็นการให้บริการจัดเก็บไฟล์ข้อมูล ซึ่งผู้ใช้งานสามารถจับเก็บไฟล์ข้อมูลและเรียกใช้งานไฟล์เหล่านี้ได้ด้วยโปรโตคอล RESTful ซึ่งการจัดเก็บไฟล์ของบริการ Swift มีการทำสำเนาไฟล์เพื่อป้องกันข้อมูลสูญหาย

6. Ceilometer เป็นการให้บริการติดตาม และตรวจวัดการใช้งานทรัพยากรเสมือน เช่น การใช้งานหน่วยประมวลผลกลาง การใช้งานพื้นที่เก็บข้อมูลบนโปรแกรมโอเพนสแต็กเพื่อบันทึกการใช้งานทรัพยากร เก็บสถิติ และคำนวณค่าใช้จ่าย รวมถึงแจ้งเตือนเมื่อมีการใช้งานเกินกำหนด

7. Horizon เป็นการให้บริการการใช้งานโปรแกรมโอเพนสแต็กผ่านทางเว็บเบราเซอร์ตามความต้องการของผู้ใช้งาน โดยผู้ใช้งานสามารถสร้างทรัพยากรเสมือนบนบริการต่าง ๆ ได้ด้วยตัวเองตามสิทธิที่ได้รับอนุญาตจากผู้ดูแลระบบ

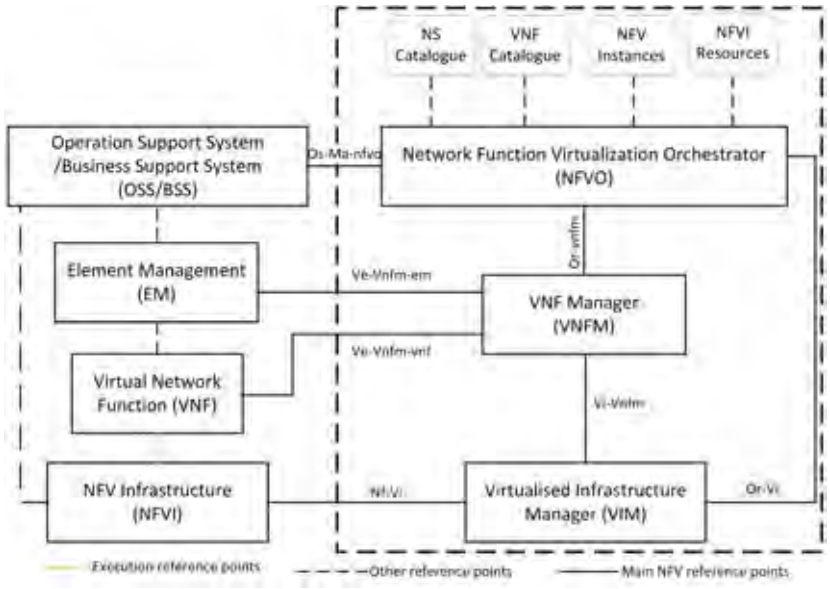
2.2 โปรแกรมโอเพนบาตง

โปรแกรมโอเพนบาตง [22] เป็นโปรแกรมประยุกต์ประสานการทำงานของฟังก์ชันโครงข่ายเสมือนตามมาตรฐาน European Telecommunications Standards Institute Network Function Virtualization Management and Orchestration (ETSI NFV MANO) [24] และเป็นส่วนหนึ่งของโครงการ OpenSDNCore [25] ซึ่งพัฒนาโดยสถาบัน Fraunhofer FOKUS และมหาวิทยาลัยเทคนิคแห่งเบอร์ลิน มีจุดมุ่งหมายเพื่อสร้างฟังก์ชันโครงข่ายเสมือนตามข้อกำหนดของ ETSI NFV ซึ่งเกิดจากความร่วมมือของผู้ให้บริการโทรศัพท์เคลื่อนที่ที่ต้องการพัฒนาโครงสร้างพื้นฐานโครงข่ายเสมือน

ด้วยการสร้างฟังก์ชันโครงข่ายบนเทคโนโลยีคลาวด์ โปรแกรมโอเพนบาตงจัดเตรียมวิธีดำเนินการเพื่อสนับสนุนการทำงานของตัวจัดการโครงสร้างพื้นฐานเสมือน รวมทั้งสามารถทำงานร่วมกับโปรแกรมประยุกต์ที่สร้างจากนักพัฒนารายอื่นได้ กลไกสำหรับการสื่อสารกับตัวจัดการฟังก์ชันโครงข่ายเสมือนคือการใช้กลไกการประกาศและการลงทะเบียน (publish/subscribe messaging, Pub/Sub) และการใช้การเชื่อมต่อแบบ RESTful

2.2.1 องค์ประกอบที่สำคัญของมาตรฐานเอ็นเอฟวีมาโน

รูปที่ 2.2 แสดงสถาปัตยกรรมของการจัดการและการประสานฟังก์ชันโครงข่ายเสมือน (NFV MANO) [24] ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้



รูปที่ 2.2: สถาปัตยกรรมเอ็นเอฟวีมาโน [24]

1. ตัวจัดการโครงสร้างพื้นฐานเสมือน (virtualised infrastructure manager, VIM) ทำหน้าที่จัดการโครงสร้างพื้นฐานของฟังก์ชันโครงข่ายเสมือน (network function virtualization infrastructure, NFVI) ได้แก่ ทรัพยากรคอมพิวเตอร์ เช่น เครื่องบริการ หน่วยเก็บข้อมูล รวมถึงทรัพยากรเสมือน และตัวจัดการเครื่องเสมือน (hypervisor) ที่อยู่ในระบบฟังก์ชันโครงข่ายเสมือน ซึ่งมีหน้าที่ดังนี้

   (a) การจัดการวัฏจักรการทำงานของทรัพยากรเสมือน ได้แก่ การสร้าง การตรวจสอบ และการทำลายเครื่องเสมือนที่อยู่ในโครงสร้างพื้นฐานของฟังก์ชันโครงข่ายเสมือน

   (b) การควบคุมรายการเครื่องเสมือนที่เกี่ยวข้องกับฮาร์ดแวร์

   (c) การควบคุมประสิทธิภาพการทำงาน และตรวจสอบข้อผิดพลาดของฮาร์ดแวร์ ซอฟแวร์ และทรัพยากรเสมือน

   (d) การจัดการเอพีไอทำให้สามารถเชื่อมต่อทรัพยากรคอมพิวเตอร์และทรัพยากรเสมือนกับระบบการจัดการอื่น ๆ ได้

2. ตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน (virtual network function manager, VNFM) ทำหน้าที่จัดการเครื่องเสมือนซึ่งทำหน้าที่ให้บริการฟังก์ชันโครงข่ายเสมือน โดยมีหน้าที่การทำงานดังนี้

   (a) การจัดการวัฏจักรการทำงานของเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน ได้แก่ การสร้าง การติดตั้ง และการทำลายเครื่องเสมือน

   (b) การทำหน้าที่จัดการข้อผิดพลาด การปรับแต่ง การจัดทำรายการ ประสิทธิภาพ และความปลอดภัยของเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน

   (c) การทำหน้าที่ปรับขนาดเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน ซึ่งส่งผลให้เกิดการปรับขนาดการใช้งานหน่วยประมวลผล

3. ตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือน (network function virtualization orchestrator, NFVO) แบ่งหน้าที่ออกเป็นสองส่วนหลักคือ

   (a) การประสานทรัพยากร: ตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือนอนุญาตให้ใช้งาน เรียกคืน และติดต่อกับทรัพยากรของโครงสร้างพื้นฐานของฟังก์ชันโครงข่ายเสมือน ซึ่งจะเชื่อมต่อกับตัวจัดการโครงสร้างพื้นฐานเสมือนโดยใช้เอพีไอ

   (b) การประสานบริการ: ตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือนสามารถสร้างบริการระหว่างเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน โดยประสานงานกับตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน เช่น การสร้างบริการระหว่างสถานีฐานของผู้ให้บริการรายหนึ่งที่สร้างจากเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนสามารถเชื่อมต่อกับระบบส่วนกลางของโครงข่ายอีกรายหนึ่งที่สร้างจากเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน และตัวประสานการทำงานฟังก์ชันโครงข่ายสามารถสร้างทอพอโลยีของการบริการโครงข่ายได้

4. คลังข้อมูล (repositories) ทำหน้าที่เก็บข้อมูลต่าง ๆ ของการให้บริการฟังก์ชันโครงข่ายเสมือนตามสถาปัตยกรรมเอ็นเอฟวีมาโนซึ่งสามารถแบ่งการเก็บข้อมูลออกเป็น 4 ประเภท ได้แก่

   (a) รายการวีเอ็นเอฟ ทำหน้าที่เป็นแหล่งเก็บข้อมูลสำหรับเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน (VNF descriptor, VNFD) ที่สามารถใช้งานได้ ข้อมูลเครื่องเสมือนเหล่านี้เป็นข้อมูลรูปแบบการติดตั้งซึ่งอธิบายการติดตั้งและวิธีการทำงาน ซึ่งถูกเรียกใช้โดยตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน และเครื่องเสมือนเหล่านี้จะถูกใช้งานโดยตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือน

   (b) รายการบริการโครงข่าย ทำหน้าที่รวบรวมการบริการโครงข่ายที่สร้างขึ้น วิธีการติดตั้งการให้บริการโครงข่าย และรูปแบบการเชื่อมโยงโครงข่ายเสมือน

   (c) รายการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน ทำหน้าที่เก็บข้อมูลทั้งหมดที่เกี่ยวกับการบริการโครงข่าย และข้อมูลที่เกี่ยวข้องกับเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน

   (d) รายการทรัพยากรโครงสร้างพื้นฐานของฟังก์ชันโครงข่ายเสมือน ทำหน้าที่เก็บข้อมูลทรัพยากรของโครงสร้างพื้นฐานที่ให้บริการฟังก์ชันโครงข่ายเสมือน

5. การจัดการองค์ประกอบย่อย (element management, EM) ทำหน้าที่ประสานการทำงานกับตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนเพื่อการติดตั้ง การปรับแต่งพารา-มิเตอร์ การจัดทำรายการ การจัดการข้อผิดพลาดและความปลอดภัยของเครื่องเสมือน ซึ่งจะทำหน้าที่เหมือนกับตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน แต่การจัดการองค์ประกอบย่อยเชื่อมต่อกับเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนโดยตรง ด้วยการติดตั้งตัวจัดการองค์ประกอบย่อยในเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือน

6. ระบบสนับสนุนการทำงาน/ระบบสนับสนุนการดำเนินธุรกิจ (operations support systems/business support systems, OSS/BSS) เป็นการรวมระบบสนับสนุนการทำงานและโปรแกรมประยุกต์ที่ผู้ให้บริการใช้ในการดำเนินธุรกิจ โดยที่ฟังก์ชันโครงข่ายเสมือนต้องสามารถทำงานร่วมกับระบบนี้ได้ ทำให้ผู้ให้บริการสามารถให้บริการและควบคุมบริการที่เกิดจากฟังก์ชันโครงข่ายเสมือนได้

2.2.2 การทำงานของโปรแกรมโอเพนบาตงเบื้องต้น

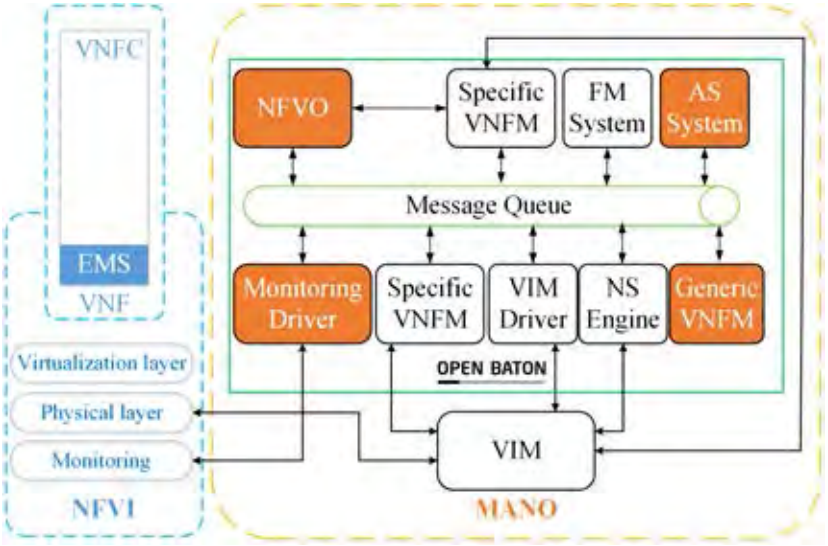
โปรแกรมโอเพนบาตงได้จัดเตรียมวิธีการสร้างเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนโดยอาศัยตัวจัดการเครื่องเสมือนซึ่งมี 3 วิธีดังนี้ [22]

1. ตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนแบบพื้นฐาน (generic VNFM) โปรแกรมโอเพนบาตงมีตัวจัดการเครื่องเสมือนนี้มาให้ใช้งานโดยปริยาย ซึ่งตัวจัดการนี้สามารถสร้าง และจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนแบบเครื่องเดียวหรือแบบหลายเครื่องเสมือนได้

2. การสร้างตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนโดยใช้ชุดพัฒนาซอฟแวร์ (software development kit, SDK) โปรแกรมโอเพนบาตงได้จัดเตรียมชุดพัฒนาซอฟแวร์ภาษาจาวาซึ่งถูกเรียกว่า vmfm-sdk เพื่อช่วยให้ผู้ใช้สามารถสร้างตัวจัดการเครื่องเสมือนขึ้นมาได้ โดยการสื่อสารระหว่างตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือนและตัวจัดการ ผู้ใช้งานสามารถเลือกใช้ได้ระหว่างโพรโทคอล java message service (JMS) หรือ โพรโทคอล representational state transfer (REST)

3. การใช้ตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนที่พัฒนาขึ้นเอง โดยวิธีการนี้อนุญาตให้ผู้ใช้งานรวมตัวจัดการที่ผู้ใช้งานสร้างขึ้นและตัวจัดการเครื่องเสมือนเข้าด้วยกัน และใช้โพรโทคอล REST สำหรับการสื่อสาร

รูปที่ 2.3 นำเสนอโครงสร้างการทำงานของโปรแกรมโอเพนบาตงตามมาตรฐานเอ็นเอฟวีมาโน ซึ่งวิทยานิพนธ์ฉบับนี้ติดตั้งโปรแกรมโอเพนสแต็กเป็นโครงสร้างพื้นฐานสำหรับสร้างฟังก์ชันเสมือนและทำงานร่วมกับโปรแกรมโอเพนบาตง โดยใช้ตัวจัดการเครื่องเสมือนที่ให้บริการฟังก์ชันโครงข่ายเสมือนแบบพื้นฐานสำหรับการสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคาร รวมถึงการนำฟังก์ชันการปรับจำนวนเครื่องเสมือนแบบอัตโนมัติตามเงื่อนไขที่กำหนด (Autoscaling) ของโปรแกรมโอเพนบาตงมาใช้ ซึ่งฟังก์ชันนี้ทำงานร่วมกับแซบบิกซ์ซึ่งเป็นโปรแกรมตรวจวัดการใช้ทรัพยากรของเครื่องเสมือน โดยนำฟังก์ชันการปรับจำนวนเครื่องเสมือนแบบอัตโนมัตินี้มาประยุกต์ใช้กับระบบจัดการพลังงานไฟฟ้าภายในอาคาร ทำให้ฟังก์ชันเสมือนที่ถูกสร้างขึ้นมีจำนวนเครื่องเสมือนสอดคล้องกับปริมาณงานที่เกิดขึ้นในระบบ ส่งผลให้ระบบจัดการพลังงานไฟฟ้าภายในอาคารใช้ทรัพยากรคอมพิวเตอร์ได้อย่างมีประสิทธิภาพเพิ่มมากขึ้น



รูปที่ 2.3: โครงสร้างการทำงานของโปรแกรมโอเพนบาตงตามมาตรฐานเอ็นเอฟวีมาโน [22]

ตัวจัดการเครื่องเสมือนแบบพื้นฐานนี้จะดำเนินการสร้างฟังก์ชันเสมือนตามมาตรฐาน ETSI MANO B.3 [24] ซึ่งจัดสรรทรัพยากรเสมือนโดยตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือนดังรูปที่ 2.4 ซึ่งโปรแกรมโอเพนบาตงมีวัฏจักรการทำงานเพื่อจัดการกับเครื่องเสมือนดังนี้

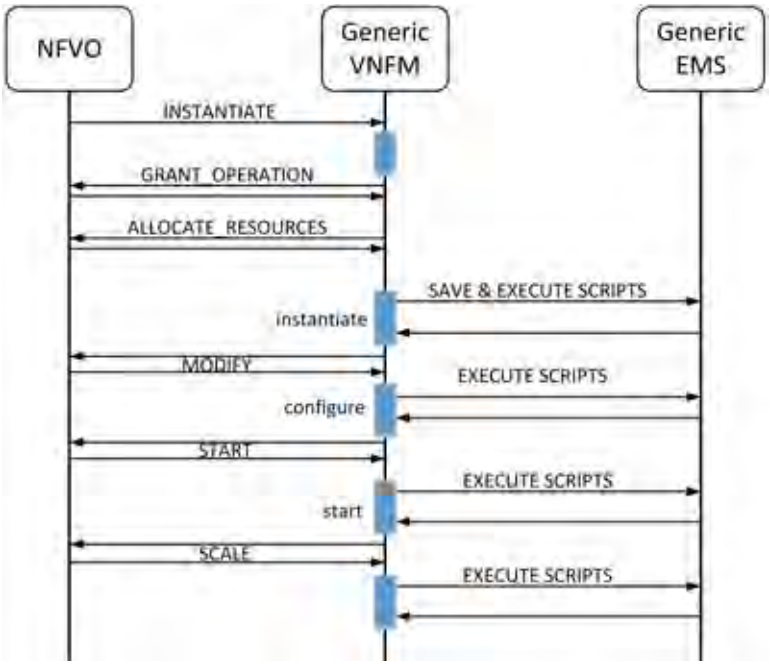
1. เหตุการณ์ INSTANTIATE เป็นขั้นตอนเริ่มต้นของการสร้างฟังก์ชันเสมือน โดยเหตุการณ์นี้เกิดขึ้นหลังจากการติดตั้งระบบปฏิบัติการลงบนเครื่องเสมือนสำเร็จ ซึ่งเป็นเหตุการณ์เพื่อการปรับแต่งพารามิเตอร์ภายในของเครื่องเสมือน และติดตั้งซอฟแวร์ที่จำเป็นต่อการสร้างฟังก์ชันเสมือน

2. เหตุการณ์ CONFIGURE เป็นขั้นตอนที่เครื่องเสมือนสามารถแลกเปลี่ยนพารามิเตอร์ต่าง ๆ เช่น หมายเลขไอพี ชื่อของเครื่องเสมือน ระหว่างเครื่องเสมือนที่อยู่ภายใต้ฟังก์ชันเสมือนเดียวกัน โดยหลักเกณฑ์สำหรับการตั้งชื่อของไฟล์ชุดคำสั่งที่อยู่ในเหตุการณ์นี้จะขึ้นต้นด้วยประเภทของเครื่องเสมือนที่เป็นเจ้าของพารามิเตอร์ และตามด้วยเครื่องหมายเส้นใต้อักขระ (underscore) และตามด้วยชื่อของคำสั่ง ส่งผลให้เครื่องเสมือนต้นทางสามารถส่งพารามิเตอร์ที่กำหนดไว้ไปยังเครื่องเสมือนปลายทางได้ โดยเครื่องเสมือนต้นทาง เครื่องเสมือนปลายทาง และพารามิเตอร์ที่ต้องการส่งจะถูกระบุอยู่ในไฟล์ข้อมูลรายละเอียดบริการโครงข่ายของฟังก์ชันเสมือน

3. เหตุการณ์ START เป็นขั้นตอนการทำงานหลังจากเหตุการณ์ CONFIGURE ใช้เพื่อเริ่มการทำงานของฟังก์ชันเสมือน

4. เหตุการณ์ TERMINATE ใช้ในกรณีที่ต้องการประมวลผลชุดคำสั่งก่อนที่จะยุติการทำงานของฟังก์ชันเสมือนลง

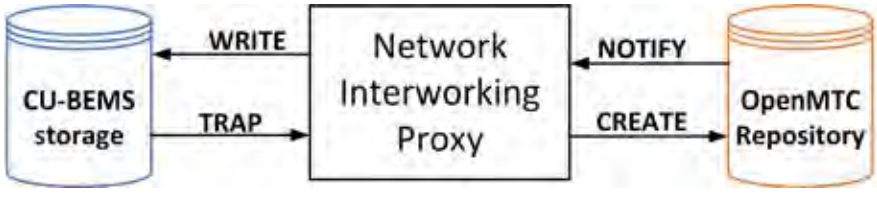
5. เหตุการณ์ SCALE ใช้ในกรณีที่ต้องการประมวลผลชุดคำสั่งเมื่อมีการเพิ่มหรือลดจำนวนเครื่องเสมือนในฟังก์ชันเสมือน



รูปที่ 2.4: ขั้นตอนการสร้างฟังก์ชันเสมือนตามมาตรฐาน ETSI MANO B.3 [24]

2.3 การทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M สำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคาร

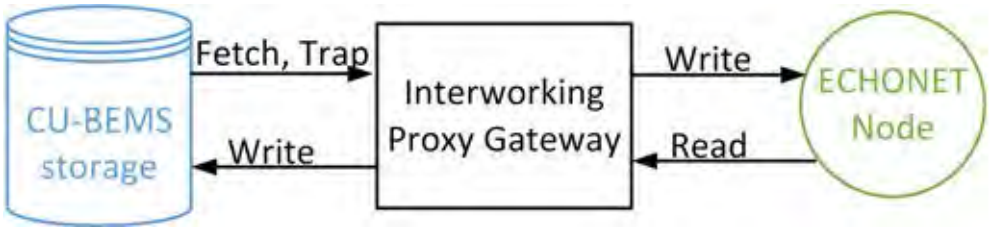
งานวิจัย [19] นำเสนอการพัฒนาเน็ทเวิร์คอินเตอร์เวิร์กกิงพร็อกซี (network interworking proxy, NIP) เพื่อใช้ทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ทำให้ระบบ CU-BEMS สามารถประสานข้อมูลกับแพลตฟอร์มโอเพนเอ็มทีซีซึ่งเป็นอีกหนึ่งมาตรฐานการเชื่อมต่อของอุปกรณ์อัจฉริยะ โดยงานวิจัยนี้ใช้โพรโทคอล WRITE สำหรับบันทึกข้อมูลลงฐานข้อมูล PostgreSQL ของระบบ CU-BEMS และโพรโทคอล TRAP ใช้สำหรับการร้องขอการจดทะเบียนเหตุการณ์ในการแจ้งเตือนข้อมูลจากฐานข้อมูลของระบบ CU-BEMS และทำการเปลี่ยนโครงสร้างข้อมูลในรูปแบบ XML ตามมาตรฐาน IEEE1888 ให้อยู่ในรูปแบบ JSON ตามมาตรฐาน ETSI M2M ในขณะที่โพรโทคอล CREATE ถูกใช้สำหรับบันทึกข้อมูลลงฐานข้อมูล MongoDB และโพรโทคอล NOTIFY ถูกใช้สำหรับการเรียกใช้ข้อมูลตามมาตรฐาน ETSI M2M ดังรูปที่ 2.5



รูปที่ 2.5: แผนภาพการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

2.4 การพัฒนาการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite สำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคาร

งานวิจัย [20] นำเสนอการพัฒนาอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์เพื่อใช้ทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite สำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคาร เพื่อเพิ่มความยืดหยุ่นให้ระบบ CU-BEMS สามารถทำงานร่วมกับอุปกรณ์อัจฉริยะที่ใช้มาตรฐานการสื่อสารที่แตกต่างออกไป โดย ECHONET Lite เป็นมาตรฐานการสื่อสารที่พัฒนาขึ้นสำหรับระบบจัดการพลังงานไฟฟ้าภายในบ้านซึ่งถูกออกแบบให้ผู้ใช้สามารถเชื่อมต่ออุปกรณ์อัจฉริยะได้สะดวกและรวดเร็ว งานวิจัยนี้ใช้โพรโทคอล TRAP เพื่อให้หน่วยเก็บข้อมูลของระบบ CU-BEMS ส่งข้อมูลออกมาทันทีที่มีการเปลี่ยนแปลง หรือโพรโทคอล FETCH เพื่อให้อินเตอร์เวิร์กกิงพร็อกซีสามารถร้องขอข้อมูลได้ตามต้องการ และใช้โพรโทคอล WRITE เพื่อบันทึกค่าเข้าสู่หน่วยเก็บข้อมูลของระบบ CU-BEMS โดยอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ทำหน้าที่รับค่าข้อมูลจากหน่วยเก็บข้อมูล CU-BEMS ประมวลผล และส่งคำสั่งไปยังอุปกรณ์อัจฉริยะที่ใช้มาตรฐาน ECHONET Lite รวมทั้งการร้องขอค่าข้อมูลจากอุปกรณ์เหล่านี้และส่งค่าไปบันทึกยังหน่วยเก็บข้อมูล CU-BEMS รูปที่ 2.6 แสดงการทำงานร่วมกันระหว่างหน่วยเก็บข้อมูลของระบบ CU-BEMS ภายใต้มาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite



รูปที่ 2.6: แผนภาพการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

2.5 การพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

งานวิจัย [21] นำเสนอการพัฒนา และการสร้างโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 ด้วยการวิเคราะห์พลังงานสูญเปล่าของเครื่องปรับอากาศภายในอาคาร โดยงานวิจัยนี้นำข้อมูลจากระบบ CU-BEMS ได้แก่ ข้อมูลการใช้พลังงานไฟฟ้า ข้อมูลอุณหภูมิ และข้อมูลการตรวจจับความเคลื่อนไหวของคนในแต่ละพื้นที่ มาประมวลผลเพื่อหาความสัมพันธ์ของข้อมูล โดยประยุกต์ใช้วิธีการทางสถิติเพื่อหาค่าพลังงานสูญเปล่าของเครื่องปรับอากาศภายในอาคาร และนำมาแสดงผลด้วยโปรแกรมสร้างมโนภาพเชิงโต้ตอบ [12] เพื่อให้ผู้ใช้รับรู้ถึงพฤติกรรมการใช้พลังงานไฟฟ้าในแต่ละพื้นที่เพิ่มมากขึ้น วิธีการดำเนินการของงานวิจัย [21] แสดงในรูปที่ 2.7

รูปที่ 2.7: แผนภาพการทำงานของโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

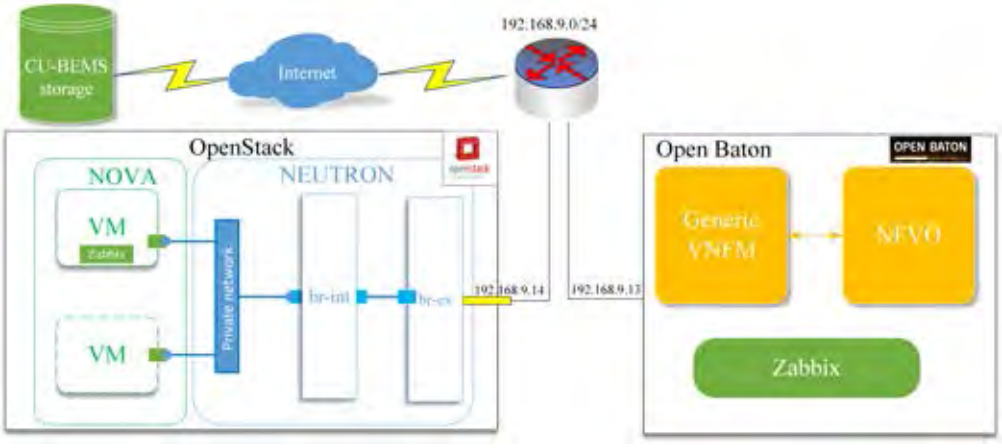
บทที่ 3

โครงสร้างและการทำงานของระบบ

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนาฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารบนโปรแกรมโอเพนสแต็ก โดยประยุกต์ใช้โปรแกรมโอเพนบาตงควบคุมขั้นตอนการสร้าง และการปรับแต่งพารามิเตอร์ ซึ่งวิทยานิพนธ์ฉบับนี้นำการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M [19] การทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite [20] และการพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 [21] มาใช้ในการประเมินความเป็นไปได้ของการสร้างฟังก์ชันเสมือนขึ้น นอกจากนี้งานวิจัยฉบับนี้นำเสนอการออกแบบ และสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารให้มีความสามารถในการขยาย และปรับลดทรัพยากรระบบแบบอัตโนมัติ โดยใช้หน่วยประมวลผลกลางเฉลี่ยและพื้นที่ของหน่วยเก็บข้อมูลของเครื่องเสมือนเป็นตัวชี้วัด

3.1 โครงสร้างของระบบที่นำเสนอ

รูปที่ 3.1 แสดงโครงสร้างของระบบการสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคาร โดยใช้โปรแกรมโอเพนบาตง [22] สร้างฟังก์ชันเสมือนที่ประกอบด้วยเครื่องเสมือนซึ่งทำงานตามหน้าที่ที่กำหนดขึ้นบนโปรแกรมโอเพนสแต็ก [16] เพื่อทำงานร่วมกับระบบ CU-BEMS และนำโปรแกรมแซบบิกซ์ [23] ซึ่งเป็นโปรแกรมวัดสมรรถนะของทรัพยากรคอมพิวเตอร์มาใช้วัดสมรรถนะของเครื่องเสมือนที่สร้างขึ้น รวมทั้งใช้เป็นตัวชี้วัดสำหรับการปรับขนาดของแพลตฟอร์มโอเพนเอ็มทีซีให้เหมาะสมกับปริมาณงานที่เกิดขึ้นในระบบ



รูปที่ 3.1: โครงสร้างการเชื่อมต่อเพื่อสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารและหน่วยเก็บข้อมูลของระบบ CU-BEMS

3.2 ส่วนประกอบของระบบ

วิทยานิพนธ์ฉบับนี้ประกอบด้วยองค์ประกอบสองส่วนเพื่อใช้สำหรับการสร้างฟังก์ชันเสมือนขึ้น ได้แก่ โปรแกรมโอเพนบาตง และโปรแกรมโอเพนสแต็ก โดยโปรแกรมโอเพนบาตงถูกใช้เพื่อควบคุมการสร้างฟังก์ชันเสมือนซึ่งประกอบด้วยเครื่องเสมือน โครงข่ายเสมือน และบริการอื่น ๆ ที่จำเป็นขึ้นบนโปรแกรมโอเพนสแต็ก โปรแกรมโอเพนบาตงถูกติดตั้งลงบนระบบปฏิบัติการ Ubuntu Server 16.04 LTS ขณะที่โปรแกรมโอเพนสแต็กซึ่งทำหน้าที่เป็นแพลตฟอร์มโครงสร้างพื้นฐานแบบแบ่งปันทรัพยากรผ่านโครงข่ายถูกติดตั้งลงบนระบบปฏิบัติการ Ubuntu Server 16.04 LTS เช่นกัน รวมถึงได้นำอุปกรณ์จัดเส้นทาง (router) มาใช้จัดเส้นทางเชื่อมต่อระหว่างระบบ CU-BEMS ที่ใช้หมายเลขไอพีสาธารณะและเครื่องเสมือนภายในโปรแกรมโอเพนสแต็กซึ่งใช้หมายเลขไอพีส่วนบุคคล

โดยวิทยานิพนธ์ฉบับนี้พัฒนาชุดคำสั่งเพื่อสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้ shell script และส่วนต่อประสานโปรแกรมโอเพนสแต็ก (OpenStack API) เพื่อให้การสร้างฟังก์ชันเสมือนบนเครื่องเสมือน และการทำงานของฟังก์ชันเสมือนสามารถประสานกันได้อย่างทุกต้อง รวมถึงชุดคำสั่งนี้สามารถทำงานร่วมกับโปรแกรมโอเพนสแต็กเพื่อสร้างฟังก์ชันเสมือนที่สามารถปรับจำนวนเครื่องเสมือนตามเงื่อนไขที่กำหนดไว้ ทำให้ฟังก์ชันเสมือนที่สร้างขึ้นใช้ทรัพยากรคอมพิวเตอร์ที่มีอยู่ได้เหมาะสมตามสถานการณ์ โดยที่ส่วนประกอบหลักสำหรับการสร้างฟังก์ชันเสมือนในวิทยานิพนธ์ฉบับนี้แสดงในตารางที่ 3.1

ตารางที่ 3.1: รายละเอียดของส่วนประกอบหลักที่ใช้สร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคาร

ส่วนประกอบหลัก	รายละเอียด
หน่วยเก็บข้อมูลของระบบ CU-BEMS รูปที่ 3.2	ตราสินค้า IBM คุณลักษณะ Intel Xeon dual core 2.40 GHz x 2 processor, 8 GB RAM ระบบปฏิบัติการ Ubuntu 12.04 LTS, 64-bit โปรแกรม Apache และโปรแกรม PostgreSQL
โปรแกรมโอเพนสแต็ก รูปที่ 3.3	ตราสินค้า HP คุณลักษณะ Intel i5-4590 3.30 GHz x 4 processor, 32 GB RAM ระบบปฏิบัติการ Ubuntu 16.04 LTS, 64-bit โปรแกรม OpenStack รุ่น Newton
โปรแกรมโอเพนบาตง รูปที่ 3.4	คุณลักษณะ Intel Celeron G1620, 2.70 GHz processor, 8 GB RAM ระบบปฏิบัติการ Ubuntu 16.04 LTS, 64-bit โปรแกรม Open Baton รุ่น 3.1.0 โปรแกรม Zabbix รุ่น 3.2
อุปกรณ์จัดเส้นทาง รูปที่ 3.5	ตราสินค้า Edimax BR-6208AC ความเร็วแบบใช้สาย 10/100 เมกะบิตต่อวินาที

รูปที่ 3.2: หน่วยเก็บข้อมูลของระบบ CU-BEMS

รูปที่ 3.3: เครื่องคอมพิวเตอร์สำหรับติดตั้งโปรแกรมโอเพนสแต็ก

รูปที่ 3.4: เครื่องคอมพิวเตอร์สำหรับติดตั้งโปรแกรมโอเพนบาตง

รูปที่ 3.5: อุปกรณ์จัดเส้นทางตราสินค้า Edimax BR-6208AC

นอกจากนี้สำหรับการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite วิทยานิพนธ์ฉบับนี้นำโนดของ ECHONET Lite เครื่องปรับอากาศ และอุปกรณ์จัดเส้นทางสำหรับโครงข่ายส่วนบุคคลเสมือนตามมาตรฐาน ECHONET Lite มาใช้เพื่อทดสอบการสร้าง และการทำงานของฟังก์ชันเสมือนนี้ ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2: รายละเอียดของอุปกรณ์ที่ใช้สร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

ส่วนประกอบ	รายละเอียด
เครื่องปรับอากาศ รูปที่ 3.6	ตราสินค้า Mitsubishi รุ่น MSZ-FK09VA
โนด ECHONET Lite รูปที่ 3.7	ตราสินค้า Mitsubishi รุ่น MAC-894IF
อุปกรณ์จัดเส้นทางสำหรับโครงข่าย ECHONET Lite รูปที่ 3.8	ตราสินค้า Tenda W309R+ ความเร็วแบบใช้สาย 10/100 เมกะบิตต่อวินาที

รูปที่ 3.6: เครื่องปรับอากาศ MITSUBISHI

รูปที่ 3.7: โนด ECHONET Lite

รูปที่ 3.8: อุปกรณ์จัดเส้นทางตราสินค้า Tenda W309R+

3.3 การทำงานของระบบ

งานวิจัย [19] พัฒนาเน็ทเวิร์คอินเตอร์เวิร์กกิงพร็อกซีเพื่อการทำงานร่วมกับแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M บนเครื่องคอมพิวเตอร์ ขณะที่งานวิจัย [20] ได้พัฒนาอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์เพื่อการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite บนอุปกรณ์ Raspberry Pi B+ และงานวิจัย [21] ได้

พัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 ซึ่งงานวิจัยเหล่านี้มีข้อจำกัด โดยผู้ติดตั้งต้องมีความชำนาญในด้านการติดตั้งระบบ รวมไปถึงการเชื่อมต่อกับโครงข่าย บางครั้งอาจมีความผิดพลาดที่เกิดขึ้นโดยผู้ติดตั้ง และเนื่องจากต้องอาศัยผู้เชี่ยวชาญดำเนินการติดตั้งระบบเอง จึงทำให้งานวิจัยเดิมนั้นขาดความสามารถการปรับจำนวนทรัพยากรระบบแบบอัตโนมัติเพื่อรองรับการเปลี่ยนแปลงของปริมาณข้อมูลที่เข้ามา วิทยานิพนธ์นี้จึงพัฒนาฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยนำงานวิจัย [19][20][21] มาสร้าง และควบคุมกระบวนการสร้างด้วยโปรแกรมโอเพนบาตง ซึ่งควบคุมขั้นตอนการสร้างด้วยไฟล์ข้อมูลรายละเอียดบริการโครงข่ายของฟังก์ชันเสมือนในรูปแบบ JSON และนำชุดคำสั่ง Shell script และส่วนต่อประสานโปรแกรมประยุกต์โอเพนสแต็กมาใช้ โดยฟังก์ชันเสมือนในวิทยานิพนธ์นี้ประกอบไปด้วย

1. ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ดังภาคผนวก ข.
2. ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite ดังภาคผนวก ค.
3. ฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 ดังภาคผนวก ง.

นอกจากนี้วิทยานิพนธ์ฉบับนี้พัฒนาชุดคำสั่งเพื่อสร้างทรัพยากรเสมือนบนโปรแกรมโอเพนสแต็ก รวมถึงชุดคำสั่งสำหรับการปรับจำนวนเครื่องเสมือนของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M เพื่อให้สอดคล้องกับภาระการทำงานของระบบ โดยประกอบด้วย

1. ชุดคำสั่งสร้างทรัพยากรเสมือนบนโปรมแกรมโอเพนสแต็ก ได้แก่ การสร้างโครงข่ายเสมือน การสร้างอุปกรณ์จัดเส้นทางเสมือน และข้อกำหนดด้านความปลอดภัย ดังภาคผนวก ก.
2. ชุดคำสั่งสำหรับการเพิ่ม และลดจำนวนเครื่องเสมือนแบบอัตโนมัติตามเงื่อนไขที่กำหนดเพื่อให้สอดคล้องกับปริมาณข้อมูลที่ส่งเข้ามาในระบบ ดังภาคผนวก ข9. และ ข11.
3. ชุดคำสั่งสำหรับการเพิ่มจำนวนเครื่องเสมือนแบบอัตโนมัติตามเงื่อนไขที่กำหนดไว้เพื่อรองรับจำนวนข้อมูลในฐานข้อมูลที่เพิ่มมากขึ้น ดังภาคผนวก ข14.

3.3.1 การสร้างสภาพแวดล้อมให้กับโปรแกรมโอเพนสแต็ก

วิทยานิพนธ์ฉบับนี้ติดตั้งโปรแกรมโอเพนสแต็กรุ่น Newton ลงบนระบบปฏิบัติการ Ubuntu server 16.04 LTS หลังจากนั้นจึงสร้างสภาพแวดล้อมให้กับเครื่องเสมือนที่จะทำหน้าที่ให้บริการฟังก์ชันเสมือน เพื่อให้ผู้ใช้งานอื่นสามารถนำฟังก์ชันเสมือนนี้ไปใช้ได้กับโปรแกรมโอเพนสแต็กที่สร้างขึ้นมาได้ วิทยานิพนธ์นี้จึงพัฒนาชุดคำสั่งสำหรับโอเพนสแต็กเพื่อให้การสร้างสภาพแวดล้อมของเครื่องเสมือนสอดคล้องกับชุดคำสั่งที่จะใช้สร้างฟังก์ชันเสมือนขึ้นด้วยโปรแกรมโอเพนบาตง โดยสภาพแวดล้อมที่จะถูกสร้างขึ้นดังภาคผนวก ก. มีลำดับขั้นตอนดังนี้

1. บรรจุไฟล์ระบบปฏิบัติการ Ubuntu cloud 16.04 LTS ที่ถูกปรับแต่งเพื่อให้ผู้ใช้งานสามารถติดตั้งฟังก์ชันเสมือนได้อย่างรวดเร็วเข้าสู่บริการ Glance ดังแสดงในรูปที่ 3.9

```
chart@openstack12:~$ openstack image show openmtcv8
+------------------+------------------------------------------------------+
| Field            | Value                                                |
+------------------+------------------------------------------------------+
| checksum         | 90a4a52952cbd5884863ae467a026e7a                     |
| container_format | bare                                                 |
| created_at       | 2017-02-18T10:24:04Z                                 |
| disk_format      | qcow2                                                |
| file             | /v2/images/181fba2b-ed70-4c31-b689-6f9894255a35/file |
| id               | 181fba2b-ed70-4c31-b689-6f9894255a35                 |
| min_disk         | 0                                                    |
| min_ram          | 0                                                    |
| name             | openmtcv8                                            |
| owner            | 2865dc4496c748a192919695372101bf                     |
| protected        | False                                                |
| schema           | /v2/schemas/image                                    |
| size             | 4226416640                                           |
| status           | active                                               |
| tags             |                                                      |
| updated_at       | 2017-02-18T10:25:33Z                                 |
| virtual_size     | None                                                 |
| visibility       | public                                               |
+------------------+------------------------------------------------------+
```

รูปที่ 3.9: ผลการบรรจุไฟล์ระบบปฏิบัติการ Ubuntu cloud 16.04 LTS ที่ถูกปรับแต่งเข้าสู่โปรแกรมโอเพนสแต็ก

2. สร้างโครงข่ายส่วนบุคคลเสมือนแบบกำหนดหมายเลขไอพีให้กับเครื่องเสมือนโดยอัตโนมัติ (dynamic host configuration protocol, DHCP) เพื่อใช้ติดต่อระหว่างเครื่องเสมือนที่ถูกสร้างขึ้น และสร้างอุปกรณ์จัดเส้นทางเสมือนเพื่อเชื่อมต่อโครงข่ายระหว่างเครื่องเสมือนที่สร้างขึ้นกับระบบ CU-BEMS รวมถึงโครงข่ายอินเทอร์เน็ต โดยใช้บริการ Neutron สำหรับการจัดการทางด้านโครงข่ายดังรูปที่ 3.10

```
chart@openstack12:~$ openstack subnet show private1_subnet
+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| allocation_pools  | 10.0.12.2-10.0.12.254                |
| cidr              | 10.0.12.0/24                         |
| created_at        | 2017-02-18T10:38:36Z                 |
| description       |                                      |
| dns_nameservers   | 161.200.80.1                         |
| enable_dhcp       | True                                 |
| gateway_ip        | 10.0.12.1                            |
| host_routes       |                                      |
| id                | 9a518dca-9a03-4ee6-9bbd-bc36216d9d94 |
| ip_version        | 4                                    |
| ipv6_address_mode | None                                 |
| ipv6_ra_mode      | None                                 |
| name              | private1_subnet                      |
| network_id        | 2d88cf5f-1c41-48b8-84c4-38f00ad9d9b2 |
| project_id        | 2865dc4496c748a192919695372101bf     |
| project_id        | 2865dc4496c748a192919695372101bf     |
| revision_number   | 2                                    |
| service_types     | []                                   |
| subnetpool_id     | None                                 |
| updated_at        | 2017-02-18T10:38:36Z                 |
+-------------------+--------------------------------------+
```

รูปที่ 3.10: ผลการสร้างโครงข่ายส่วนบุคคลเสมือนเพื่อใช้ติดต่อระหว่างเครื่องเสมือน

3. สร้างกุญแจ (key pair) เพื่อให้ผู้ใช้เข้าถึงเครื่องเสมือนผ่านโพรโทคอล secure shell (ssh) ซึ่งเมื่อสร้างสำเร็จ กุญแจส่วนบุคคล (private key) จะถูกเก็บไว้ในเครื่องคอมพิวเตอร์ของผู้ที่สร้างกุญแจนี้ขึ้นมาดังรูปที่ 3.11

```
chart@openstack12:~$ openstack keypair show openstack_key12
+-------------+-------------------------------------------------+
| Field       | Value                                           |
+-------------+-------------------------------------------------+
| created_at  | 2017-02-18T10:25:43.000000                      |
| deleted     | False                                           |
| deleted_at  | None                                            |
| fingerprint | c0:9a:99:0c:54:d1:bb:80:11:f9:41:5f:4e:8b:3c:34 |
| id          | 2                                               |
| name        | openstack_key12                                 |
| updated_at  | None                                            |
| user_id     | 264de891d01b4c11a9d77cee454f0fd9                |
+-------------+-------------------------------------------------+
```

รูปที่ 3.11: ผลการสร้างกุญแจเพื่อให้ผู้ใช้สามารถเข้าถึงเครื่องเสมือน

4. กำหนดข้อกำหนดความปลอดภัยของกลุ่มเครื่องเสมือน (security group rules) ให้กับเครื่องเสมือนที่จะใช้ติดตั้งฟังก์ชันเสมือน โดยข้อกำหนดความปลอดภัยที่ถูกกำหนดขึ้นจะอนุญาตให้ข้อมูลสามารถเข้าหรือออกได้เฉพาะช่องเข้า/ออกที่กำหนดไว้เท่านั้น ข้อกำหนดความปลอดภัยแสดงในรูปที่ 3.12

```
chart@openstack12:~$ openstack security group rule list 5b4e7167-9640-43fb-87ae-0fbd664e9e79
+--------------------------------------+-------------+-----------+-------------+-----------------------+
| ID                                   | IP Protocol | IP Range  | Port Range  | Remote Security Group |
+--------------------------------------+-------------+-----------+-------------+-----------------------+
| 077e8a7d-3b29-4cac-b5be-ced07c2c33a5 | tcp         | 0.0.0.0/0 | 1000:1000   | None                  |
| 233aa70e-c535-4906-a2ae-0007e01da546 | tcp         | 0.0.0.0/0 | 10050:10050 | None                  |
| 5c9540c0-9c28-4147-8e88-1b0ead3bbdec | tcp         | 0.0.0.0/0 | 80:80       | None                  |
| 6a07c3b4-f1d6-444a-98c3-f494029868e7 | tcp         | 0.0.0.0/0 | 15000:15000 | None                  |
| 6fb55287-b117-425a-a117-80bfdeed6205 | tcp         | 0.0.0.0/0 | 8081:8081   | None                  |
| 96c72a83-2a9f-484b-bdf1-fe4cf5d5ac97 | icmp        | 0.0.0.0/0 |             | None                  |
| de29b24d-6e38-4bd4-9e17-9d680cfe51ec | tcp         | 0.0.0.0/0 | 22:22       | None                  |
+--------------------------------------+-------------+-----------+-------------+-----------------------+
```

รูปที่ 3.12: ผลการสร้างข้อกำหนดความปลอดภัยของกลุ่มเครื่องเสมือนให้กับเครื่องเสมือน

5. สำหรับการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M วิทยานิพนธ์นี้นำบริการ Load-Balancing-as-a-Service (LBaaS) มาใช้ของโปรแกรมโอเพนสแต็กเพื่อกระจายข้อมูลไปยังเครื่องเสมือนเอ็นเอสซีแอลที่มีจำนวนตั้งแต่ 1 เครื่องขึ้นไป โดยกำหนดการทำงานแบบวนรอบ (round robin) กำหนดพอร์ตใช้งานหมายเลข 15000 และโพรโทคอลทีซีพี (TCP) เพื่อทดสอบการเพิ่มประสิทธิภาพการทำงานของแพลตฟอร์มโอเพนเอ็มทีซี ผลการสร้างบริการ LBaaS แสดงในรูปที่ 3.13

6. สำหรับการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite เนื่องจากเครื่องเสมือนจำเป็นต้องติดต่อกับโครงข่าย ECHONET Lite ซึ่งเป็นข่ายงานบริเวณเฉพาะที่ (local area network) ดังนั้นจึงสร้างโครงข่ายส่วนบุคคลเสมือนชื่อ private_echonet เพื่อเชื่อมต่อกับส่วนต่อประสานโครงข่ายจุดที่สองของเครื่องเสมือน โดยโครงข่ายส่วนบุคคลเสมือนที่สร้างขึ้นจะจัดสรรหมายเลขไอพีให้อยู่ในช่วงของหมายเลขไอพีของโครงข่าย ECHONET Lite ดังแสดงในรูปที่ 3.14

```
chart@openstack12:~$ neutron lbaas-loadbalancer-show loadbalancer1
+---------------------+--------------------------------------------------+
| Field               | Value                                            |
+---------------------+--------------------------------------------------+
| admin_state_up      | True                                             |
| description         |                                                  |
| id                  | 68cda28d-ccce-475a-955e-483818ef6653             |
| listeners           | {"id": "f433971a-6203-4fba-9c4d-44da1f8f97eb"}   |
| name                | loadbalancer1                                    |
| operating_status    | ONLINE                                           |
| pools               | {"id": "ad9e5b2b-3262-4ba3-a06a-4246f2551cb6"}   |
| provider            | octavia                                          |
| provisioning_status | ACTIVE                                           |
| tenant_id           | 2865dc4496c748a192919695372101bf                 |
| vip_address         | 10.0.12.6                                        |
| vip_port_id         | 634f592c-0bf7-4b63-9cb5-9906e2ac2a91             |
| vip_subnet_id       | 9a518dca-9a03-4ee6-9bbd-bc36216d9d94             |
+---------------------+--------------------------------------------------+
chart@openstack12:~$ neutron lbaas-pool-show pool1
+---------------------+--------------------------------------------------+
| Field               | Value                                            |
+---------------------+--------------------------------------------------+
| admin_state_up      | True                                             |
| description         |                                                  |
| healthmonitor_id    |                                                  |
| id                  | ad9e5b2b-3262-4ba3-a06a-4246f2551cb6             |
| lb algorithm        | ROUND ROBIN                                      |
+---------------------+--------------------------------------------------+
chart@openstack12:~$ neutron lbaas-listener-show listener1
+---------------------------+--------------------------------------------------+
| Field                     | Value                                            |
+---------------------------+--------------------------------------------------+
| admin_state_up            | True                                             |
| connection_limit          | -1                                               |
| default_pool_id           | ad9e5b2b-3262-4ba3-a06a-4246f2551cb6             |
| default_tls_container_ref |                                                  |
| description               |                                                  |
| id                        | f433971a-6203-4fba-9c4d-44da1f8f97eb             |
| loadbalancers             | {"id": "68cda28d-ccce-475a-955e-483818ef6653"}   |
| name                      | listener1                                        |
| protocol                  | TCP                                              |
| protocol_port             | 15000                                            |
| sni_container_refs        |                                                  |
| tenant_id                 | 2865dc4496c748a192919695372101bf                 |
+---------------------------+--------------------------------------------------+
```

รูปที่ 3.13: ผลการสร้างบริการ Load-Balancing-as-a-Service ในโปรแกรมโอเพนสแต็ก

```
chart@openstack14:~$ neutron subnet-show private_echonet
+-------------------+------------------------------------------------------+
| Field             | Value                                                |
+-------------------+------------------------------------------------------+
| allocation_pools  | {"start": "192.168.0.150", "end": "192.168.0.200"}   |
| cidr              | 192.168.0.0/24                                       |
| created_at        | 2017-03-09T15:38:48Z                                 |
| description       |                                                      |
| dns_nameservers   | 161.200.80.1                                         |
| enable_dhcp       | False                                                |
| gateway_ip        |                                                      |
| host_routes       |                                                      |
| id                | 4f9afbd9-923b-490c-ae10-361a40ebe61d                 |
| ip_version        | 4                                                    |
| ipv6_address_mode |                                                      |
| ipv6_ra_mode      |                                                      |
| name              | private_echonet                                      |
| network_id        | 042b67ff-3969-467c-9301-4858a86ce9f6                 |
| project_id        | 4b9d95f06fbb4a2e9c6625cf73934113                     |
| revision_number   | 3                                                    |
| service_types     |                                                      |
| subnetpool_id     |                                                      |
| tenant_id         | 4b9d95f06fbb4a2e9c6625cf73934113                     |
| updated_at        | 2017-03-09T15:40:24Z                                 |
+-------------------+------------------------------------------------------+
```

รูปที่ 3.14: ผลการสร้างโครงข่ายส่วนบุคคลเสมือนเพื่อใช้ติดต่อกับโครงข่าย ECHONET Lite

7. สำหรับการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 เนื่องจากเครื่องเสมือนจำเป็นต้องร้องขอข้อมูล

จากหน่วยเก็บข้อมูลระบบ CU-BEMS เป็นปริมาณมาก จึงจำเป็นต้องเชื่อมต่อกับโครงข่ายของระบบ CU-BEMS ผ่านทางข่ายงานบริเวณเฉพาะที่ ดังนั้นจึงสร้างโครงข่ายส่วนบุคคลเสมือนชื่อ private_cubems1 เพื่อเชื่อมต่อกับส่วนต่อประสานโครงข่ายจุดที่สองของเครื่องเสมือน โดยกำหนดการจัดสรรหมายเลขไอพีให้อยู่ในช่วงของหมายเลขไอพีของโครงข่าย CU-BEMS ตามที่ผู้ดูแลระบบ CU-BEMS กำหนดให้ ดังรูปที่ 3.15

```
 (neutron) subnet-show private_cubems1
+-------------------+-----------------------------------------------------+
| Field             | Value                                               |
+-------------------+-----------------------------------------------------+
| allocation_pools  | {"start": "192.168.90.132", "end": "192.168.90.134"} |
| cidr              | 192.168.90.0/24                                     |
| created_at        | 2017-03-08T13:00:14Z                                |
| description       |                                                     |
| dns_nameservers   |                                                     |
| enable_dhcp       | False                                               |
| gateway_ip        |                                                     |
| host_routes       |                                                     |
| id                | 340dfd2e-1f9f-47ad-a4aa-f1f442f61495                |
| ip_version        | 4                                                   |
| ipv6_address_mode |                                                     |
| ipv6_ra_mode      |                                                     |
| name              | private_cubems1                                     |
| network_id        | b74dc91c-393b-4cec-a9ad-717b65c601a8                |
| project_id        | 4b9d95f06fbb4a2e9c6625cf73934113                    |
| revision_number   | 2                                                   |
| service_types     |                                                     |
| subnetpool_id     |                                                     |
| tenant_id         | 4b9d95f06fbb4a2e9c6625cf73934113                    |
| updated_at        | 2017-03-08T13:00:14Z                                |
+-------------------+-----------------------------------------------------+
```

รูปที่ 3.15: ผลการสร้างโครงข่ายส่วนบุคคลเสมือนเพื่อใช้ติดต่อกับโครงข่าย CU-BEMS

3.3.2 การลงทะเบียนโนดโอเพนสแต็กในโปรแกรมโอเพนบาตง

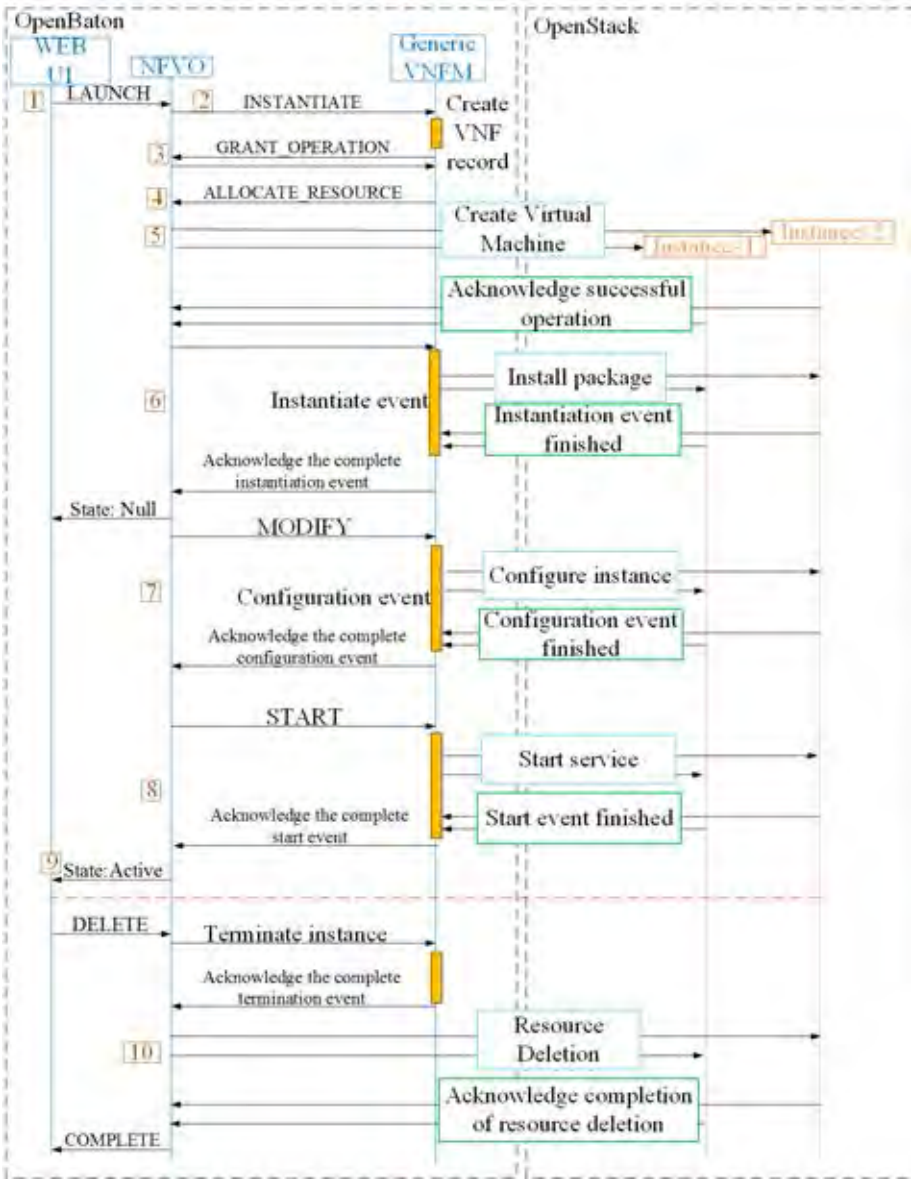
เพื่อให้โปรแกรมโอเพนบาตงสามารถสร้างและควบคุมทรัพยากรเสมือนที่มีอยู่ในโนดโอเพนสแต็กที่สร้างขึ้นได้ จึงจำเป็นต้องลงทะเบียนโนดโอเพนสแต็กบนโปรแกรมโอเพนบาตงด้วยการระบุข้อมูลของโนดโอเพนสแต็กในรูปแบบ JSON ซึ่งประกอบไปด้วย ชื่อของโนด หมายเลขไอพีของโนด กลุ่มที่ต้องการใช้งาน กุญแจที่สร้างขึ้น และชื่อของกฎความปลอดภัยที่สร้างขึ้นตามหัวข้อที่ 3.3.1 และการระบุข้อมูลโนดโอเพนสแต็กในรูปแบบ JSON เป็นไปตามรูปที่ 3.16

```
{
  "name":"vim12",
  "authUrl":"http://192.168.9.12:5000/v2.0",
  "tenant":"admin",
  "username":"admin",
  "password":"chula2016",
  "keyPair":"openstack_key12",
  "securityGroups": [
    "openstack_sec12"
  ],
  "type":"openstack",
  "location":{
        "name":"TH1",
        "latitude":"13.738622",
        "longitude":"100.530947"
  }
}
```

รูปที่ 3.16: ข้อมูลของโนดโอเพนสแต็กที่ใช้ลงทะเบียนในรูปแบบ JSON

3.3.3 การทำงานร่วมกันระหว่างโปรแกรมโอเพนบาตงและโอเพนสแต็ก

วิทยานิพนธ์นี้เลือกใช้ตัวจัดการเครื่องเสมือนแบบพื้นฐานสำหรับการสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคาร โดยขั้นตอนการสร้างฟังก์ชันเสมือนเป็นไปตามมาตรฐาน ETSI MANO B.3 ซึ่งทำการจัดสรรทรัพยากรเสมือนโดยตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือน โดยมีขั้นตอนการทำงานตามรูปที่ 3.17 ดังนี้



รูปที่ 3.17: ผังเวลาวิธีการสร้างฟังก์ชันเสมือนขึ้นบนแพลตฟอร์มโอเพนสแต็กด้วยโปรแกรมโอเพนบาตง [22]

1. ขั้นตอนที่ 1: ผู้ใช้งานสร้างไฟล์ข้อมูลรายละเอียดบริการโครงข่ายของฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคาร ในรูปแบบ JSON และ

เรียกใช้งานผ่านเว็บเบราว์เซอร์ หลังจากนั้นเว็บเบราว์เซอร์จะร้องขอข้อมูลไปที่ตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือน

2. ขั้นตอนที่ 2: ตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือนตรวจสอบความถูกต้องของข้อมูลเสร็จแล้วจึงส่งข้อความ “INSTANTIATE” ไปให้ตัวจัดการเครื่องเสมือนแบบพื้นฐาน เพื่อร้องขอรายการบันทึกฟังก์ชันโครงข่ายเสมือน (virtual network function record, VNFR)

3. ขั้นตอนที่ 3: ตัวจัดการเครื่องเสมือนแบบพื้นฐานตรวจสอบจำนวนทรัพยากรเสมือนของโปรแกรมโอเพนสแต็กที่ถูกลงทะเบียนไว้ ถ้ามีทรัพยากรเสมือนเพียงพอต่อการสร้างฟังก์ชันเสมือน ตัวจัดการเครื่องเสมือนแบบพื้นฐานตอบกลับด้วยข้อความ "GRANT_OPERATION" ไปยังตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือน

4. ขั้นตอนที่ 4: ตัวจัดการเครื่องเสมือนแบบพื้นฐานส่งการร้องขอเพื่อสร้างทรัพยากรเสมือนด้วยข้อความ "ALLOCATE_RESOURCE" ไปที่ตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือน

5. ขั้นตอนที่ 5: ตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือนดำเนินการสร้างทรัพยากรเสมือนขึ้นบนโปรแกรมโอเพนสแต็ก ได้แก่ การสร้างเครื่องเสมือน การติดตั้งระบบปฏิบัติการ รวมทั้งติดตั้งระบบจัดการส่วนย่อย (element management system, EMS) ซึ่งทำหน้าที่ประมวลผลคำสั่งตามที่ถูกกำหนดไว้ในไฟล์ข้อมูลรายละเอียดบริการโครงข่ายของฟังก์ชันเสมือน และถูกเชื่อมต่อเข้ากับโครงข่ายส่วนบุคคลที่ถูกสร้างขึ้นบนโปรแกรมโอเพนสแต็ก

6. ขั้นตอนที่ 6: เครื่องเสมือนที่ติดตั้งระบบปฏิบัติการสำเร็จแล้ว จะถูกติดตั้งซอฟแวร์ที่จำเป็นสำหรับการสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารในขั้นตอน instantiate event

7. ขั้นตอนที่ 7: เครื่องเสมือนที่ถูกสร้างขึ้นมาจะแลกเปลี่ยนข้อมูลระหว่างเครื่องเสมือนที่อยู่ในฟังก์ชันเสมือนเดียวกัน เช่น หมายเลขไอพีและชื่อของเครื่องเสมือน (hostname) ในขั้นตอน configure event

8. ขั้นตอนที่ 8: เครื่องเสมือนแต่ละเครื่องเริ่มให้บริการตามหน้าที่ที่กำหนดไว้ เพื่อให้สามารถทำหน้าที่เป็นฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคาร

9. ขั้นตอนที่ 9: เมื่อฟังก์ชันเสมือนถูกติดตั้งและเริ่มใช้งานสำเร็จ สถานะของบริการบนหน้าเว็บเบราว์เซอร์จะเปลี่ยนเป็น “ACTIVE”

10. ถ้าผู้ใช้งานต้องการยุติการทำงานของฟังก์ชันเสมือน สามารถเลือกคำสั่ง “DELETE” ได้จากหน้าเว็บเบราว์เซอร์ จากนั้นตัวประสานการทำงานฟังก์ชันโครงข่ายเสมือนจะทำการทำลายเครื่องเสมือนที่อยู่ในฟังก์ชันเสมือนนี้บนโปรแกรมโอเพนสแต็กทิ้ง

บทที่ 4

การสร้างและการทดสอบฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์

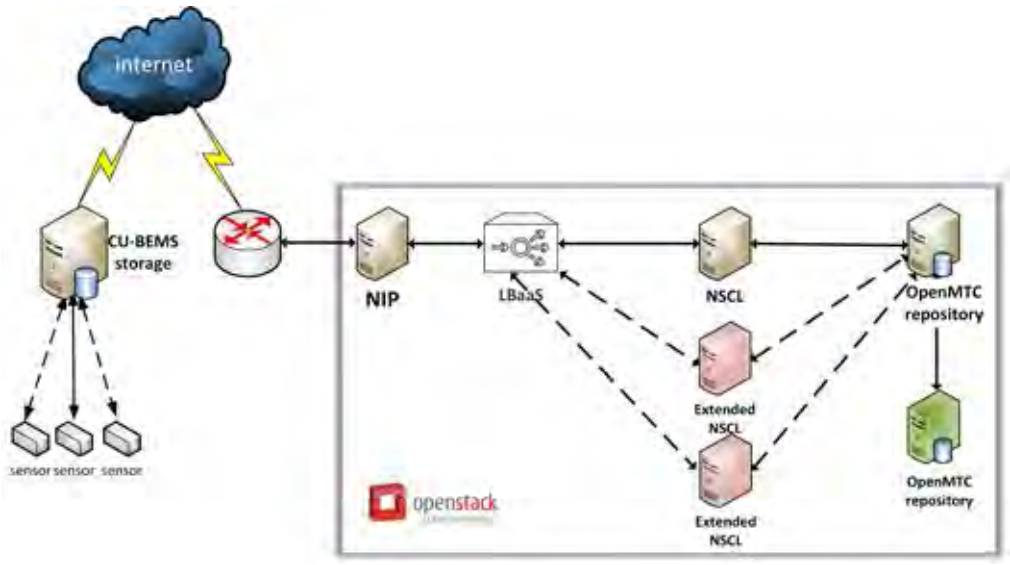
จากบทที่ 3 ได้อธิบายถึงโครงสร้างการทำงานร่วมกันระหว่างโปรแกรมโอเพนสแต็กและโปรแกรมโอเพนบาตงเพื่อสร้างฟังก์ชันเสมือนขึ้น สำหรับวิธีการสร้าง และวิธีการทดสอบฟังก์ชันเสมือนตามขอบเขตวิทยานิพนธ์นั้นจะถูกนำเสนอในบทนี้ ซึ่งได้แก่ ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M รวมถึงการปรับขนาดของแพลตฟอร์มโอเพนเอ็มทีซีแบบอัตโนมัติ ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite และฟังก์ชันเสมือนการวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 รวมถึงแสดงผลผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบ ฟังก์ชันเสมือนในวิทยานิพนธ์ฉบับนี้ถูกสร้างขึ้นด้วยโปรแกรมโอเพนบาตง ซึ่งถูกกำหนดวิธีการสร้าง และการทำงานด้วยไฟล์ข้อมูลรายละเอียดบริการโครงข่าย (network service descriptor, NSD) ในรูปแบบ JSON ซึ่งพารามิเตอร์ต่าง ๆ เป็นไปตามข้อกำหนดของเอ็นเอฟวีมาโน และหลังจากระบบปฏิบัติการถูกติดตั้งเสร็จสิ้น ระบบจัดการส่วนย่อยจะถูกติดตั้งและทำหน้าที่ดาวน์โหลดชุดคำสั่งรวมถึงดำเนินการในแต่ละชุดคำสั่งตามลำดับของเหตุการณ์ที่ระบุไว้

4.1 ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

4.1.1 การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M โดยใช้โปรแกรมโอเพนบาตงเป็นดังรูปที่ 4.1 ซึ่งประกอบไปด้วยเครื่องเสมือนจำนวน 3 เครื่อง ได้แก่ เครื่องเสมือนเอ็นไอพี เครื่องเสมือนเอ็นเอสซีแอล และเครื่องเสมือนมองโกดีบีอย่างละ 1 เครื่อง ซึ่งเครื่องเสมือนแต่ละเครื่องมีหน้าที่ดังนี้

1. เครื่องเสมือนเอ็นไอพี ทำหน้าที่เปลี่ยนแปลงโครงสร้างข้อมูลในรูปแบบ XML ตามมาตรฐาน IEEE1888 ให้เป็นข้อมูลในรูปแบบ JSON ตามมาตรฐาน ETSI M2M รวมถึงเปลี่ยนแปลงโครงสร้างข้อมูลจากรูปแบบ JSON ตามมาตรฐาน ETSI M2M ให้เป็นข้อมูลในรูปแบบ XML ตามมาตรฐาน IEEE1888
2. เครื่องเสมือนเอ็นเอสซีแอล ทำหน้าที่รับข้อมูลในรูปแบบ JSON ตามมาตรฐาน ETSI M2M ประมวลผลและส่งต่อไปจัดเก็บที่ฐานข้อมูลมองโกดีบีที่เครื่องเสมือนมองโกดีบี
3. เครื่องเสมือนมองโกดีบี ทำหน้าที่เป็นหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M



รูปที่ 4.1: โครงสร้างการเชื่อมต่อของการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

วิทยานิพนธ์ฉบับนี้ได้แยกส่วนการประมวลผลข้อมูล (เครื่องเสมือนเอ็นเอสซีแอล) และส่วนการจัดเก็บข้อมูล (เครื่องเสมือนมองโกดีบี) ของแพลตฟอร์มโอเพนเอ็มทีซีออกจากกัน เนื่องจากการทดสอบเบื้องต้นพบว่าเมื่อป้อนข้อมูลเข้าสู่แพลตฟอร์มโอเพนเอ็มทีซีจำนวนมาก การทำงานของทั้งส่วนการประมวลผลข้อมูล และส่วนการจัดเก็บข้อมูลใช้หน่วยประมวลผลกลางสูง และเพื่อทำให้การปรับจำนวนทรัพยากรของส่วนการประมวลผลข้อมูล และส่วนการจัดเก็บข้อมูลเป็นอิสระต่อกัน โดยการสร้างฟังก์ชันเสมือนมีขั้นตอนการเรียกใช้ชุดคำสั่งของเครื่องเสมือนในแต่ละเหตุการณ์ตามตารางที่ 4.1 และขั้นตอนการสร้างฟังก์ชันเสมือนเป็นไปตามรูปที่ 4.2 โดยไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M รวมถึงเงื่อนไขสำหรับการปรับจำนวนทรัพยากรของแพลตฟอร์มโอเพนเอ็มทีซีอยู่ในภาคผนวก ข1.

ตารางที่ 4.1: ขั้นตอนการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

เหตุการณ์	เครื่องเสมือน เอ็นไอพี	เครื่องเสมือน เอ็นเอสซีแอล	เครื่องเสมือน มองโกดีบี
INSTANTIATE	install_nip.sh	install_nscl.sh	install_db.sh
CONFIGURE	server_configgnip.sh db_configgnip.sh	client_configgnscl.sh db_config.sh	server_config.sh client_configdb.sh
START	start_nip.sh	start_nscl.sh	start_db.sh
TERMINATE	-	terminate_lbaas.sh	-
SCALE_IN	-	scale_in_lbaas.sh	-

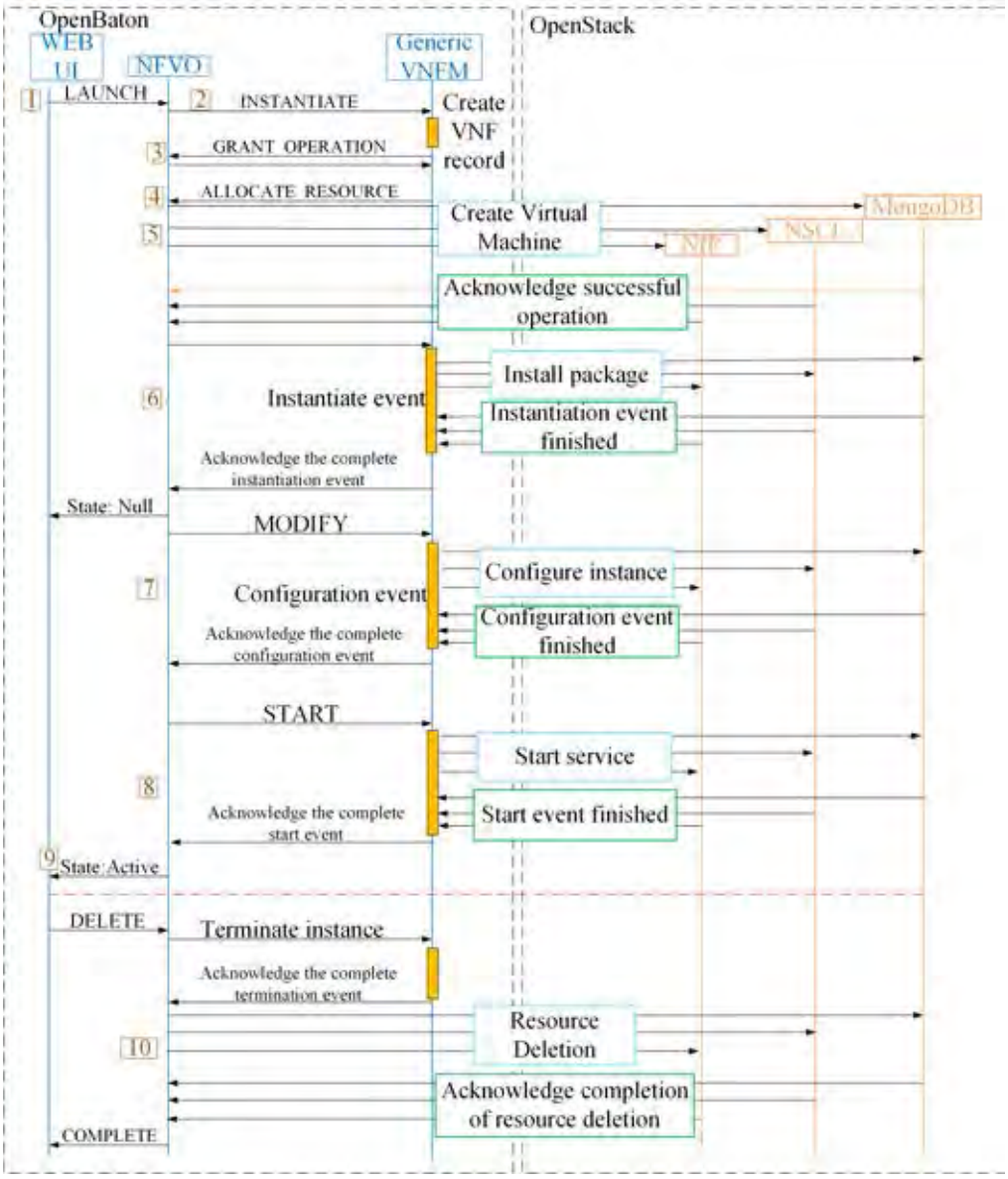
- เหตุการณ์ INSTANTIATE
  เมื่อเครื่องเสมือนทั้งสามเครื่องถูกสร้างและติดตั้งระบบปฏิบัติการ Ubuntu cloud 16.04 LTS ที่มีการติดตั้งโปรแกรมที่จำเป็นสำหรับการสร้างฟังก์ชันเสมือนนี้ รวมถึงเชื่อมต่อกับโครงข่ายที่ถูกกำหนดไว้สำเร็จแล้ว จะเข้าสู่เหตุการณ์ INSTANTIATE ซึ่งเครื่องเสมือนทั้งสามจะถูกปรับแต่งค่าพารามิเตอร์เช่น หมายเลขไอพี ชื่อของเครื่องเสมือนภายในเครื่องของตัวเอง รวมทั้งการประสานเวลาจาก ntp.chula.ac.th เพื่อให้เครื่องเสมือนทั้งสามเครื่องมีระบบเวลาเท่ากันและเท่ากับเวลาของระบบ CU-BEMS

- เหตุการณ์ CONFIGURE
  เครื่องเสมือนทั้งสามแลกเปลี่ยนค่าพารามิเตอร์ของแต่ละเครื่อง ได้แก่ หมายเลขไอพี และชื่อของเครื่องเสมือน เพื่อให้เครื่องเสมือนทั้งสามรู้จักกัน และรู้ถึงเครื่องปลายทางที่จะต้องส่งข้อมูลไปให้ สำหรับเครื่องเสมือนมองโกดีบีจะถูกปรับแต่งระบบการทำงานให้รองรับการขยายฐานข้อมูล (sharding) [26] ทันทีเมื่อมีเครื่องเสมือนมองโกดีบีเพิ่มขึ้นเพื่อให้สามารถเก็บข้อมูลได้มากขึ้น ขณะที่เครื่องเสมือนเอ็นเอสซีแอลจะถูกปรับแต่งพารามิเตอร์ เพื่อให้สามารถรองรับการปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติตามระดับการทำงานของหน่วยประมวลผลกลาง และเครื่องเสมือนเอ็นเอสซีแอลจะลงทะเบียนหมายเลขไอพีของเครื่องในบริการ LBaaS ของโปรแกรมโอเพนสแต็ก ซึ่งถูกนำมาใช้เพื่อการกระจายข้อมูลไปยังเครื่องเสมือนเอ็นเอสซีแอลแต่ละเครื่อง

- เหตุการณ์ START
  หลังจากเหตุการณ์ CONFIGURE เสร็จสิ้น เครื่องเสมือนทั้งสามเครื่องจะเริ่มให้บริการตามหน้าที่ของแต่ละเครื่อง โดยที่เครื่องเสมือนเอ็นไอพีทำหน้าที่ประสานข้อมูลแบบทันทีระหว่างหน่วยเก็บข้อมูลของระบบ CU-BEMS และแพลตฟอร์มโอเพนเอ็มทีซีสำหรับ point ID ที่กำหนดไว้ ขณะที่เครื่องเสมือนเอ็นเอสซีแอลทำหน้าที่รับข้อมูลจากเครื่องเสมือนเอ็นไอพีและส่งไปจัดเก็บที่เครื่องเสมือนมองโกดีบี และเครื่องเสมือนมองโกดีบีทำหน้าที่เป็นคลังข้อมูลตามมาตรฐาน ETSI M2M

- เหตุการณ์ TERMINATE
  เป็นเหตุการณ์ที่จะเกิดขึ้นในกรณีที่ผู้ใช้งานต้องการยุติการทำงานของฟังก์ชันเสมือนนี้ เครื่องเสมือนทั้งสามเครื่องจะถูกทำลายทิ้งโดยโปรแกรมโอเพนบาตง โดยก่อนที่เครื่องเสมือนเอ็นเอ็สซีแอลจะถูกทำลายทิ้ง เครื่องเสมือนนี้จะลบหมายเลขไอพีที่ลงทะเบียนไว้ในบริการ LBaaS ออก

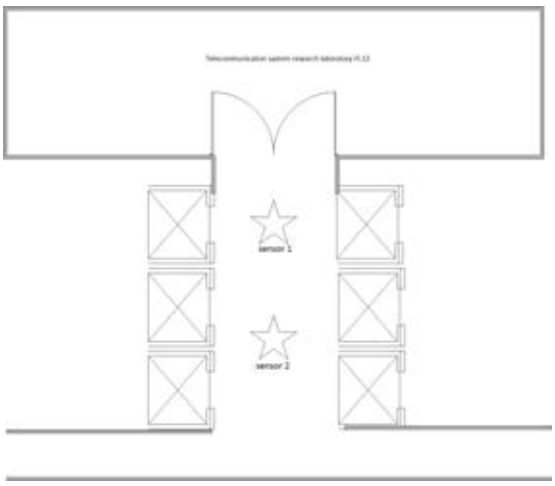
- เหตุการณ์ SCALE_IN
  เป็นเหตุการณ์ที่เกิดขึ้นเฉพาะกับเครื่องเสมือนเอ็นเอสซีแอล ในกรณีที่มีจำนวนเครื่องเสมือนมากกว่าหนึ่งเครื่องทำงานอยู่ เมื่อระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานมีค่าสูงขึ้นเกินกว่าระดับที่กำหนด เครื่องเสมือนจะถูกลดจำนวนลง โดยก่อนที่เครื่องเสมือนจะถูกทำลายเครื่องเสมือนนี้จะลบหมายเลขไอพีของตัวเองออกจากบริการ LBaaS ของโปรแกรมโอเพนสแต็ก



รูปที่ 4.2: ผังเวลาการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

4.1.2 การประเมินฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

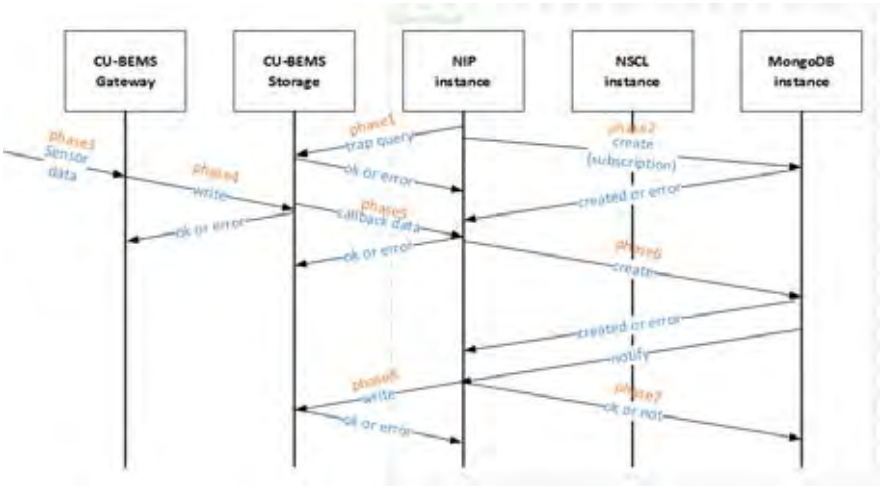
การทดสอบการฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M โดยใช้ระบบ CU-BEMS ที่มีอยู่ และแพลตฟอร์มโอเพนเอ็มทีซีที่ถูกสร้างขึ้นบนแพลตฟอร์มโอเพนสแต็ก ซึ่งการทดสอบนี้ใช้ข้อมูลจากตัวรับรู้การเคลื่อนไหวของคนซึ่งถูกติดตั้งที่บริเวณหน้าลิฟท์ชั้น 13 อาคารเจริญวิศวกรรม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย จำนวน 2 ตำแหน่ง โดยที่ชื่อของข้อมูลตัวรับรู้การเคลื่อนไหวของคนถูกกำหนดด้วย point ID ในรูปแบบยูอาร์แอลที่มีอยู่ในระบบ CU-BEMS ดังนี้



รูปที่ 4.3: แผนผังแสดงตำแหน่งตัวรับรู้ที่ใช้เพื่อทดสอบฟังก์ชันเสมือนการทำงานร่วมกันแบบทันที ระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

จุดที่ 1 : http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/elevetorfront/z1/sensor1/monitor/pir
จุดที่ 2 : http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/elevetorfront/z1/sensor2/monitor/pir
ขณะที่ point ID ที่ถูกใช้เพื่อประสานข้อมูลจากแพลตฟอร์มโอเพนเอ็มทีซีตามมาตรฐาน ETSI M2M มาอยู่ในรูปแบบตามมาตรฐาน IEEE1888 ถูกกำหนด point ID ดังนี้
จุดที่ 1 : http://xhat1.ee.eng.chula.ac.th/eng4/fl13/corridor/elevetorfront/z1/sensor1/monitor/pir
จุดที่ 2 : http://xhat1.ee.eng.chula.ac.th/eng4/fl13/corridor/elevetorfront/z1/sensor2/monitor/pir
ซึ่งตำแหน่งของตัวรับรู้แสดงในรูปที่ 4.3

สำหรับการประเมินการทำงานของฟังก์ชันเสมือนนี้ จะพิจารณาด้วยระยะเวลาการสร้างและยุติการทำงานของฟังก์ชันเสมือนนี้ และพิจารณาการทำงานของฟังก์ชันเสมือนตามวัตถุประสงค์ของการสร้างฟังก์ชันเสมือนนี้ขึ้นดังรูปที่ 4.4 และมีขั้นตอนการทำงานดังนี้

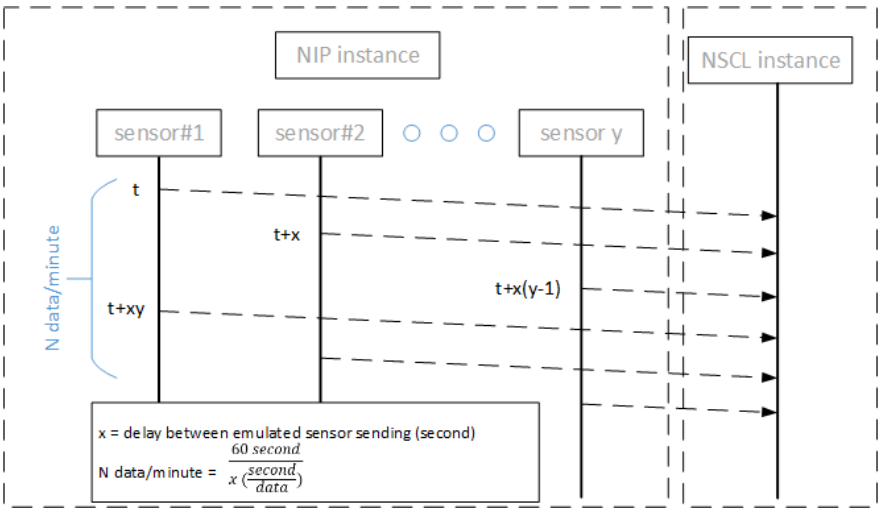


รูปที่ 4.4: ผังเวลาการถ่ายโอนข้อมูลของตัวรับรู้จากระบบ CU-BEMS ไปยังแพลตฟอร์มโอเพนเอ็มทีซีแบบทันที

1. ขั้นตอนที่ 1 (phase 1): เครื่องเสมือนเอ็นไอพีร้องขอการส่งข้อมูลแบบแทรปที่มีเงื่อนไขการแจ้งเตือนข้อมูลของตัวรับรู้ไปที่หน่วยเก็บข้อมูลของระบบ CU-BEMS โดยกำหนดยูอาร์ไอของการส่งข้อมูลกลับเป็นยูอาร์ไอของอุปกรณ์จัดเส้นทางซึ่งใช้หมายเลขไอพีสาธารณะ โดยอุปกรณ์จัดเส้นทางตั้งค่าการส่งต่อข้อมูล (port forwarding) เพื่อให้ข้อมูลสามารถส่งต่อไปยังเครื่องเสมือนได้อย่างถูกต้อง
2. ขั้นตอนที่ 2 (phase 2): เครื่องเสมือนเอ็นไอพีส่งข้อมูล subscription ไปยังเครื่องเสมือนเอ็นเอสซีแอลเพื่อติดตามการเปลี่ยนแปลงข้อมูลตัวรับรู้ในแพลตฟอร์มโอเพนเอ็มทีซี
3. ขั้นตอนที่ 3 (phase 3): เมื่อตัวรับรู้ตรวจพบการเปลี่ยนแปลงหรือครบรอบการส่งข้อมูล ตัวรับรู้จะส่งข้อมูลมาที่เกตเวย์ของระบบ CU-BEMS
4. ขั้นตอนที่ 4 (phase 4): เกตเวย์ของระบบ CU-BEMS จะส่งต่อข้อมูลของตัวรับรู้ไปยังหน่วยเก็บข้อมูลของระบบ CU-BEMS
5. ขั้นตอนที่ 5 (phase 5): เมื่อข้อมูลตัวรับรู้เปลี่ยนแปลง หน่วยเก็บข้อมูลของระบบ CU-BEMS จะส่งข้อมูลมายังเครื่องเสมือนเอ็นไอพี จากนั้นเครื่องเสมือนเอ็นไอพีจะปรับรูปแบบของข้อมูลนั้นในรูปแบบ XML ตามมาตรฐาน IEEE1888 ให้อยู่ในรูปแบบ JSON ตามมาตรฐาน ETSI M2M
6. ขั้นตอนที่ 6 (phase 6): เครื่องเสมือนเอ็นไอพีส่งข้อมูลไปบันทึกที่ฐานข้อมูลมองโกดีบีผ่านทางเครื่องเสมือนเอ็นเอสซีแอลตามมาตรฐาน ETSI M2M
7. ขั้นตอนที่ 7 (phase 7): เครื่องเสมือนเอ็นเอสซีแอลส่งข้อข้อมูลกลับไปยังเครื่องเสมือนเอ็นไอพีด้วยโพรโทคอล NOTIFY เมื่อข้อมูลที่ติดตามในขั้นตอนที่ 2 มีการเปลี่ยนแปลง
8. ขั้นตอนที่ 8 (phase 8): เครื่องเสมือนเอ็นไอพีรับข้อมูลมาจากขั้นตอนที่ 7 และเปลี่ยนแปลงรูปแบบข้อมูลจากรูปแบบ JSON ตามมาตรฐาน ETSI M2M ให้อยู่ในรูปแบบ XML ตามมาตรฐาน IEEE1888 และส่งข้อมูลนี้ไปบันทึกที่หน่วยเก็บข้อมูลของระบบ CU-BEMS

4.1.3 การประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอล

การประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอลด้วยการส่งข้อมูลจำนวนมากเพื่อประเมินสมรรถนะเครื่องเสมือนเอ็นเอสซีแอลที่ถูกสร้างขึ้น วิทยานิพนธ์ฉบับนี้เลือกใช้ระดับของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานเป็นพารามิเตอร์เพื่อประเมินสมรรถนะของเครื่องเสมือนเอ็นเอสซีแอล หัวข้อนี้เครื่องเสมือนเอ็นเอสซีแอลประกอบด้วยส่วนประมวลผลและส่วนเก็บข้อมูลดังเช่นงานวิจัย [19] เพื่อพิจารณาเรื่องคอขวดของระบบ การทดสอบนี้ดำเนินการโดยใช้เครื่องเสมือนเอ็นไอพีส่งข้อมูลเลียนแบบไปยังเครื่องเสมือนเอ็นเอสซีแอล โดยเปลี่ยนแปลงความถี่การส่งข้อมูลเลียนแบบในแต่ละรอบของการทดสอบดังรูปที่ 4.5 และคำนวณหาค่าเฉลี่ยของจำนวนข้อมูลต่อนาทีที่เครื่องเสมือนเอ็นเอสซีแอลสามารถบันทึกข้อมูลลงฐานข้อมูลมองโกดีบีของแพลตฟอร์มโอเพนเอ็มทีซีตามมาตรฐาน ETSI M2M ได้



รูปที่ 4.5: ผังเวลาการถ่ายโอนข้อมูลตัวรับรู้เลียนแบบไปยังแพลตฟอร์มโอเพนเอ็มทีซีเพื่อทดสอบสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอล

4.1.4 การประเมินความสามารถการปรับจำนวนทรัพยากรแบบอัตโนมัติของแพลตฟอร์มโอเพนเอ็มทีซีโดยใช้โปรแกรมโอเพนบาตง

หลังจากประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอล ในหัวข้อนี้จะประเมินความเป็นไปได้สำหรับการปรับจำนวนทรัพยากรแบบอัตโนมัติของแพลตฟอร์มโอเพนเอ็มทีซีด้วยโปรแกรมโอเพนบาตง เพื่อให้แพลตฟอร์มโอเพนเอ็มทีซีสามารถรองรับจำนวนข้อมูลที่เพิ่มมากขึ้น โดยจะแบ่งการประเมินออกเป็นสองส่วน ได้แก่ การประเมินความสามารถการปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติ และการประเมินความสามารถการปรับเพิ่มจำนวนเครื่องเสมือนมองโกดีบีแบบอัตโนมัติ ดังนี้

1. การประเมินความสามารถการปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติ สำหรับส่วนของการประมวลผลข้อมูลของแพลตฟอร์มโอเพนเอ็มทีซี โดยพิจารณาระดับการทำงานของหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลเป็นตัวชี้วัด เพื่อให้สอดคล้องกับปริมาณข้อมูลที่ถูกส่งเข้ามาในแพลตฟอร์มโอเพนเอ็มทีซี การกำหนดหลักเกณฑ์สำหรับตรวจวัดระดับการทำงานของหน่วยประมวลผลกลางเพื่อใช้สำหรับการปรับจำนวนของเครื่องเสมือนเอ็นเอสซีแอลเป็นไปตามรูปที่ 4.6 จำนวนเครื่องเสมือนเอ็นเอสซีแอลจะเพิ่มขึ้นครั้งละ 1 เครื่อง เมื่อระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานเฉลี่ยต่ำกว่าระดับ 70 เปอร์เซ็นต์เป็นระยะเวลา 1 นาที โดยสามารถเพิ่มเครื่องเสมือนเอ็นเอสซีแอลได้สูงสุด 3 เครื่อง ซึ่งเครื่องเสมือนเอ็นเอสซีแอลจะทำงานร่วมกับบริการ LBaaS ของโปรแกรมโอเพนสแต็กเพื่อกระจายการทำงานไปยังเครื่องเสมือนเอ็นเอสซีแอลต่าง ๆ โดยที่ยังใช้ฐานข้อมูลมองโกดีบีร่วมกัน ขณะที่การปรับลดจำนวนของเครื่องเสมือนเอ็นเอสซีแอลจะกระทำครั้งละ 1 เครื่อง เมื่อระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานเฉลี่ยสูงกว่า 85 เปอร์เซ็นต์เป็นระยะเวลา 3 นาที โดยสามารถลดเครื่องเสมือนเอ็นเอสซีแอลจนเหลือจำนวน 1 เครื่อง โดยที่การลดเครื่องเสมือนจะเป็นการทำลายเครื่องเสมือนแบบสุ่มตามมาตรฐานเอ็นเอฟวีมาโน ซึ่งพิจารณาว่าเครื่องเสมือนเอ็นเอสซีแอลที่สร้างขึ้นมาเป็นเครื่องเสมือนที่ทำหน้าที่เหมือนกัน

รูปที่ 4.6: ข้อกำหนดการปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลโดยโปรแกรมโอเพนบาตง

2. การประเมินความสามารถการปรับเพิ่มจำนวนเครื่องเสมือนมองโกดีบีแบบอัตโนมัติ เพื่อเพิ่มพื้นที่จัดเก็บข้อมูลของแพลตฟอร์มโอเพนเอ็มทีซี เนื่องจากการให้บริการระบบจัดการพลังงานไฟฟ้าภายในอาคารเป็นเวลานาน ฐานข้อมูลของระบบจะเก็บข้อมูลจำนวนมาก ส่งผลให้ผู้ดูแลระบบต้องเพิ่มพื้นที่จัดเก็บข้อมูลเพื่อรองรับจำนวนข้อมูลที่จะเพิ่มขึ้นในอนาคต ดังนั้นการเพิ่มพื้นที่จัดเก็บข้อมูลให้กับฐานข้อมูลแบบอัตโนมัติสำหรับแพลตฟอร์มโอเพนเอ็มทีซี จะพิจารณาจากพารามิเตอร์พื้นที่จัดเก็บข้อมูลที่เหลืออยู่ของเครื่องเสมือนมองโกดีบี โดยเมื่อพื้นที่จัดเก็บข้อมูลเหลือต่ำกว่า 30 เปอร์เซ็นต์ของพื้นที่จัดเก็บข้อมูลทั้งหมดเป็นระยะเวลา 90 วินาที ตามข้อกำหนดในรูปที่ 4.7 โปรแกรมโอเพนบาตงจะสร้างเครื่องเสมือนมองโกดีบีเพิ่มขึ้นแบบอัตโนมัติเพื่อเพิ่มพื้นที่จัดเก็บข้อมูลของแพลตฟอร์มดอเพนเอ็มทีซี วิทยานิพนธ์ฉบับนี้กำหนดให้สามารถเพิ่มเครื่องเสมือนมองโกดีบีได้สูงสุด 2 เครื่อง และนำระบบการขยายฐานข้อมูล (sharding) สำหรับฐานข้อมูลมองโกดีบีมาประยุกต์ใช้กับแพลตฟอร์มโอเพนเอ็มทีซีเพื่อให้เกิดการประสานข้อมูลเมื่อมีเครื่องเสมือนมองโกดีบีมากกว่า 1 เครื่องขึ้นไป

รูปที่ 4.7: ข้อกำหนดการปรับเพิ่มจำนวนเครื่องเสมือนมองโกดีบีโดยโปรแกรมโอเพนบาตง

4.1.5 การประเมินสมรรถนะการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอล

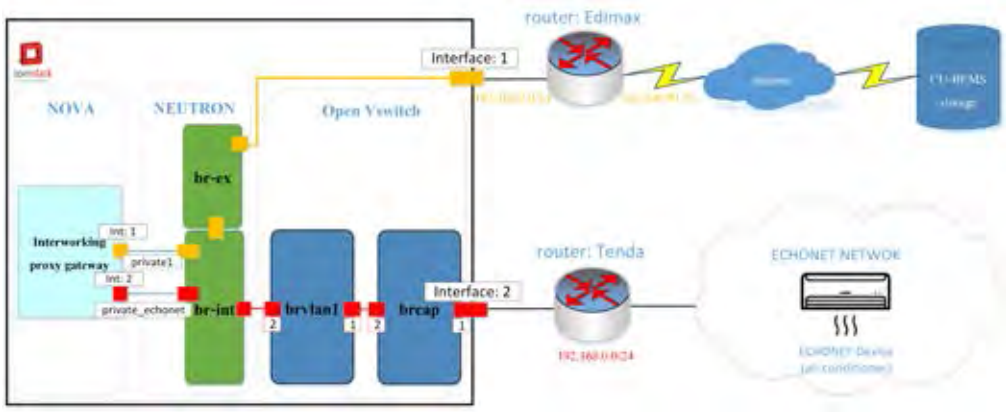
หลังจากแยกส่วนของการประมวลผลข้อมูล และส่วนของการจัดเก็บข้อมูลของแพลตฟอร์มโอเพนเอ็มทีซีออกจากกัน รวมถึงการปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติ และการทำงานร่วมกับบริการ LBaaS ของโปรแกรมโอเพนสแต็กเพื่อให้เกิดการกระจายข้อมูลที่เข้ามาไปยังเครื่องเสมือนเอ็นเอสซีแอลแต่ละเครื่องดังแสดงในรูปที่ 4.1 ในหัวข้อนี้จะประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอลเหล่านี้ ด้วยวิธีการหาค่าเฉลี่ยจำนวนข้อมูลต่อนาทีที่สามารถบันทึกลงฐานข้อมูลมองโกดีบีของแพลตฟอร์มโอเพนเอ็มทีซีได้ และวัดระดับการทำงานของหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลแต่ละเครื่อง รวมถึงเปรียบเทียบสมรรถนะการทำงานของแฟลตฟอร์มโอเพนเอ็มทีซีเมื่อใช้เครื่องเสมือนเอ็นเอสซีแอลมากกว่า 1 เครื่องเทียบกับแพลตฟอร์มโอเพนเอ็มทีซีที่ใช้เครื่องเสมือนจำนวน 1 เครื่องดังหัวข้อที่ 4.1.3 โดยแบ่งการวัดสมรรถนะการทำงานเป็นดังนี้

1. เครื่องเสมือนเอ็นเอสซีแอลจำนวน 1 เครื่องและเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่อง
2. เครื่องเสมือนเอ็นเอสซีแอลจำนวน 2 เครื่องและเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่อง
3. เครื่องเสมือนเอ็นเอสซีแอลจำนวน 3 เครื่องและเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่อง

4.2 ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

4.2.1 การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

การสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite บนแพลตฟอร์มโอเพนสแต็กมีโครงสร้างการเชื่อมต่อดังรูปที่ 4.8 ซึ่งประกอบด้วยเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์จำนวน 1 เครื่อง โดยเครื่องเสมือนนี้สามารถเชื่อมต่อกับหน่วยเก็บข้อมูลของ CU-BEMS และโครงข่าย ECHONET Lite ได้ วิทยานิพนธ์ฉบับนี้นำงานวิจัย [20] มาประยุกต์ โดยสร้างเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ซึ่งทำหน้าที่ร้องขอข้อมูล และเขียนข้อมูลบันทึกลงหน่วยเก็บข้อมูลของระบบ CU-BEMS รวมทั้งทำหน้าที่ส่งคำสั่ง และร้องขอข้อมูลจากเครื่องปรับอากาศที่เชื่อมต่อกับโนด ECHONET Lite ส่งผลทำให้เครื่องเสมือนนี้สามารถควบคุมการเปิด และปิดเครื่องปรับอากาศเครื่องนี้ได้เมื่อค่าอุณหภูมิเฉลี่ยที่ตรวจวัดได้จากตัวรับรู้สภาพแวดล้อมของระบบ CU-BEMS ซึ่งติดตั้งอยู่ภายในห้องคลัสเตอร์ที่ติดตั้งเครื่องปรับอากาศนี้อยู่ในระดับที่สูงกว่า หรือต่ำกว่าค่าอุณหภูมิที่กำหนดไว้ ผู้ดูแลระบบสามารถกำหนดค่าอุณหภูมิเพื่อใช้ควบคุมการเปิด และปิดเครื่องปรับอากาศ รวมถึงสามารถตั้งค่าอุณหภูมิของเครื่องปรับอากาศเมื่อเปิดเครื่องปรับอากาศได้ผ่านทางไฟล์ข้อมูลรายละเอียดบริการโครงข่ายในขั้นตอนการสร้างฟังก์ชันนี้ขึ้นด้วยโปรแกรมโอเพนบาตงดังรูปที่ 4.9 นอกจากนี้เครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์จะบันทึกสถานะการทำงานของเครื่องปรับอากาศเครื่องนี้เข้าสู่หน่วยเก็บข้อมูลของระบบ CU-BEMS ด้วย โดยไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite อยู่ในภาคผนวก ค.



รูปที่ 4.8: โครงสร้างการเชื่อมต่อของการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

```
"vendor":"NRG-team",
"version":"0.1",
"name":"echonet-gw",
"type":"server",
"endpoint":"generic",
"configurations":{
   "name":"config_name",
   "configurationParameters":[
        {
            "confKey":"TempUpper",
            "value":"23"
        },
        {
            "confKey":"TempLower",
            "value":"22"
        },
        {
            "confKey":"TempSet",
            "value":"21"
        }
   ]
```

รูปที่ 4.9: ไฟล์ข้อมูลรายละเอียดบริการโครงข่ายฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

เนื่องจากโครงข่าย ECHONET Lite เป็นข่ายงานบริเวณเฉพาะที่ ดังนั้นเพื่อเชื่อมต่อเครื่องเสมือนกับโครงข่าย ECHONET Lite วิทยานิพนธ์นี้จึงนำโปรแกรมโอเพนวีสวิตช์ (Open vSwitch) ซึ่งทำหน้าที่เป็นสวิตช์เสมือนภายในเครื่องคอมพิวเตอร์มาใช้ เพื่อเชื่อมต่อระหว่างโครงข่ายเสมือน private_echonet ซึ่งเป็นโครงข่ายในบริการ Neutron ของโปรแกรมโอเพนสแต็กกับส่วนต่อประสานโครงข่ายจุดที่สองของเครื่องคอมพิวเตอร์ และเชื่อมต่อเครื่องคอมพิวเตอร์เข้ากับอุปกรณ์จัดเส้นทางตราสินค้า Tenda ซึ่งเป็นอุปกรณ์จุดเส้นทางโครงข่าย ECHONET Lite โดยโปรแกรมโอเพนวีสวิตช์สร้างบริดจ์ขึ้นจำนวน 2 บริดจ์ คือ บริดจ์ brvlan และบริดจ์ brcap โดยบริดจ์ brvlan ทำหน้าที่เชื่อมต่อกับบริดจ์ br-int ซึ่งเป็นบริดจ์ของบริการ Neutron และกำหนดการไหลของข้อมูล เนื่องจากโครงข่ายเสมือน private_echonet ของฟังก์ชันเสมือนนี้มีการกำหนดหมายเลข VLAN ดังนั้นข้อมูลจากโครงข่ายเสมือน private_echonet ที่ไหลออกผ่าน brvlan จะถูกถอดป้ายระบุหมายเลข VLAN เพื่อให้สามารถใช้งานร่วมกับโครงข่าย ECHONET Lite ได้ ขณะที่ข้อมูลที่ไหลเข้า brvlan จะถูกติดป้ายระบุหมายเลข VLAN เพื่อให้สามารถใช้งานร่วมกับโครงข่ายเสมือน private_echonet ที่เชื่อมต่อ

```
ovs-vsctl add-br brvlan1
ovs-vsctl add-port brvlan1 vlan1_brcap
ovs-vsctl add-port brvlan1 vlan1_br-int
ovs-vsctl set Interface vlan1_brcap options:peer=brcap_vlan1
ovs-vsctl set Interface vlan1_br-int options:peer=br-int_vlan1
ovs-ofctl add-flow brvlan1 in_port=1,actions=mod_vlan_vid:6,output:2
ovs-ofctl add-flow brvlan1 in_port=2,dl_vlan=6,actions=strip_vlan,output:1
ovs-ofctl add-flow brvlan1 in_port=2,actions=drop
```

รูปที่ 4.10: การกำหนดการไหลของข้อมูลของบริดจ์ brvlan เพื่อเชื่อมต่อโครงข่ายเสมือน private_echonet และโครงข่าย ECHONET Lite

กับเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์บนแพลตฟอร์มโอเพนสแต็ก โดยข้อกำหนดการไหลของข้อมูลแสดงในรูปที่ 4.10 ขณะที่ brcap ทำหน้าที่เชื่อมกับส่วนต่อประสานโครงข่ายจุดที่สองของเครื่องคอมพิวเตอร์เพื่อเชื่อมต่อกับโครงข่าย ECHONET Lite

สำหรับการสร้างฟังก์ชันเสมือนนี้ขึ้นด้วยโปรแกรมโอเพนบาตงมีขั้นตอนการสร้าง และการทำงานด้วยชุดคำสั่งตามตารางที่ 4.2 ซึ่งมีรายละเอียดดังนี้

- เหตุการณ์ INSTANTIATE
  เมื่อเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ถูกสร้างขึ้นและติดตั้งระบบปฏิบัติการ Ubuntu cloud 16.04 LTS สำเร็จ จึงติดตั้งโปรแกรมที่จำเป็นสำหรับการทำงานของฟังก์ชันเสมือน ซึ่งใช้แพลตฟอร์ม Node.js และกำหนดเส้นทางให้กับเครื่องเสมือนเพื่อเชื่อมต่อไปยังโครงข่าย ECHONET Lite ผ่านทางส่วนต่อประสานโครงข่ายจุดที่สองของเครื่องเสมือน

- เหตุการณ์ CONFIGURE
  กำหนดการทำงานของฟังก์ชันเสมือนขึ้น โดยฟังก์ชันเสมือนที่สร้างขึ้นนี้ถูกกำหนดให้ตรวจสอบอุณหภูมิภายในห้องคลัสเตอร์ทุก 10 นาที

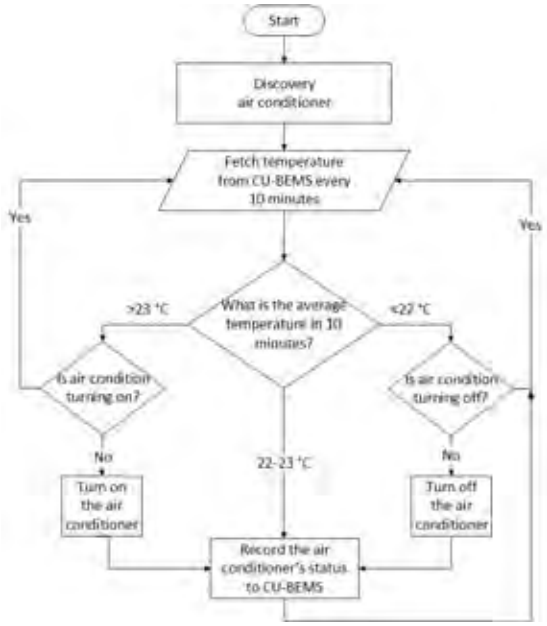
- เหตุการณ์ START
  เครื่องเสมือนจะเริ่มค้นหาหมายเลขไอพีของเครื่องปรับอากาศที่อยู่ในโครงข่าย ECHONET Lite หลังจากนั้นจะทำหน้าที่ตรวจสอบอุณหภูมิจากตัวรับรู้อุณหภูมิของระบบ CU-BEMS เพื่อกำหนดการทำงานของเครื่องปรับอากาศ

ตารางที่ 4.2: ขั้นตอนการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

เหตุการณ์	เครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์
INSTANTIATE	install_echonet.sh
CONFIGURE	server_configure.sh
START	start_echonet.sh

4.2.2 การประเมินการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

การทดสอบการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite กระทำโดยทดสอบกับเครื่องปรับอากาศที่ถูกติดตั้งในห้องคลัสเตอร์ศูนย์เชี่ยวชาญเฉพาะด้านเทคโนโลยีโทรคมนาคม ชั้น 13 อาคารเจริญวิศวกรรม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และภายในห้องนี้มีการติดตั้งตัวรับรู้สภาพแวดล้อมแบบไร้สายซึ่งมีตัวรับรู้อุณหภูมิบรรจุไว้ภายใน และส่งข้อมูลไปบันทึกที่หน่วยเก็บข้อมูลของระบบ CU-BEMS ทุก 1 นาทีซึ่งใช้ point ID คือ
http://bems.ee.eng.chula.ac.th/eng4/fl13/north/room_server/z1/sensor1/monitor/temperature
โดยการทำงานของฟังก์ชันเสมือนมีขั้นตอนดังรูปที่ 4.11 และมีรายละเอียดดังนี้



รูปที่ 4.11: ผังการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

1. เครื่องเสมือนทำการค้นหาเครื่องปรับอากาศที่อยู่ในโครงข่าย ECHONET Lite และบันทึกหมายเลขไอพีของเครื่องปรับอากาศลงไฟล์ข้อมูล
2. เครื่องเสมือนร้องขอข้อมูลอุณหภูมิภายใน 10 นาทีล่าสุดของตัวรับรู้ไร้สายที่ติดตั้งภายในห้องคลัสเตอร์จากหน่วยเก็บข้อมูล CU-BEMS และหาค่าเฉลี่ยเพื่อลดข้อผิดพลาดการอ่านค่าของตัวรับรู้ สำหรับการทดสอบนี้กำหนดให้เปิดเครื่องปรับอากาศเมื่ออุณหภูมิสูงกว่า 23 องศาเซลเซียส และปิดเครื่องปรับอากาศเมื่ออุณหภูมิต่ำกว่า 22 องศาเซลเซียส
3. เครื่องเสมือนส่งค่าสถานะล่าสุดของเครื่องปรับอากาศและค่าเฉลี่ยอุณหภูมิย้อนหลัง 10 นาทีไปบันทึกที่หน่วยเก็บข้อมูลของระบบ CU-BEMS โดยใช้ point ID
http://xhat2.test.chula.ac.th/echonet/eng4/fl13/north/server_room/z1/monitor/airstatus
http://xhat2.test.chula.ac.th/echonet/eng4/fl13/north/server_room/z1/monitor/servertemp

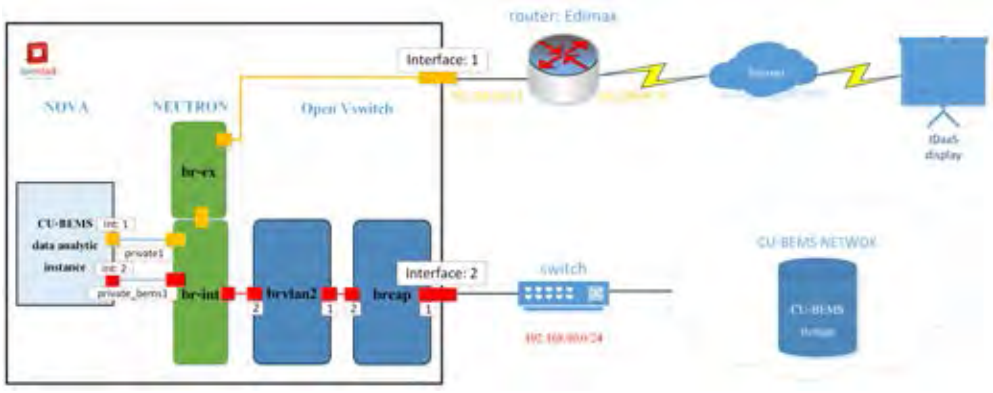
ตามลำดับ โดยเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์จะทำงานตามขั้นตอนที่ 2-3 ทุก 10 นาที

4.3 ฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

4.3.1 การสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

การสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 มีโครงสร้างการเชื่อมต่อดังรูปที่ 4.12 ซึ่งประกอบด้วยเครื่องเสมือนสำหรับการวิเคราะห์ข้อมูลจำนวน 1 เครื่องบนแพลตฟอร์มโอเพนสแต็ก เครื่องเสมือนนี้ถูกออกแบบให้มีส่วนต่อประสานโครงข่ายจำนวน 2 จุด จุดที่หนึ่งเชื่อมต่อกับโครงข่ายเสมือน private1 เพื่อการเข้าถึง และเชื่อมต่อกับอินเทอร์เน็ต ขณะที่จุดที่สองเชื่อมต่อกับโครงข่ายเสมือน private_bems1 ซึ่งจะเชื่อมต่อไปยังโครงข่ายของระบบ CU-BEMS ซึ่งเป็นข่ายงานบริเวณเฉพาะที่ ดังนั้นการเชื่อมต่อส่วนต่อประสานโครงข่ายจุดที่สองของเครื่องเสมือนจึงเชื่อมต่อซึ่งต่อประสานกับบริดจ์ brvlan2 และบริดจ์ brcap ซึ่งบริดจ์ brvlan2 ทำหน้าที่กำหนดการไหลของข้อมูล โดยข้อมูลที่ไหลออกจากโครงข่ายเสมือน private_bems1 จะถูกถอดป้ายระบุหมายเลข VLAN เพื่อให้สามารถใช้งานร่วมกับโครงข่ายของระบบ CU-BEMS ได้ ขณะที่ข้อมูลที่ไหลเข้าโครงข่ายเสมือน private_bems1 จะถูกติดป้ายระบุหมายเลข VLAN เพื่อให้สามารถติดต่อกับเครื่องเสมือนได้ ข้อกำหนดการไหลของข้อมูลแสดงในรูปที่ 4.13 วัตถุประสงค์ของการสร้างฟังก์ชันเสมือนนี้ ได้แก่

1. วิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 และแสดงผลด้วยโปรแกรมสร้างมโนภาพเชิงโต้ตอบ
2. ตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟต์ชั้น 13 อาคารเจริญวิศวกรรม



รูปที่ 4.12: โครงสร้างการเชื่อมต่อของการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

```
ovs-vsctl add-br brvlan2
ovs-vsctl add-port brvlan2 vlan2_brcap
ovs-vsctl add-port brvlan2 vlan2_br-int
ovs-vsctl set Interface vlan2_brcap options:peer=brcap_vlan2
ovs-vsctl set Interface vlan2_br-int options:peer=br-int_vlan2
ovs-ofctl add-flow brvlan2 in_port=1,actions=mod_vlan_vid:5,output:2
ovs-ofctl add-flow brvlan2 in_port=2,dl_vlan=5,actions=strip_vlan,output:1
ovs-ofctl add-flow brvlan2 in_port=2,actions=drop
```

รูปที่ 4.13: การกำหนดการไหลของข้อมูลของบริดจ์ brvlan เพื่อเชื่อมต่อโครงข่ายเสมือน private_cubems1 และโครงข่ายของระบบ CU-BEMS

ฟังก์ชันเสมือนนี้ถูกสร้างขึ้นด้วยโปรแกรมโอเพนบาตง โดยไฟล์ข้อมูลรายละเอียดบริการโครงข่าย ฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐาน มาตรฐาน IEEE1888 อยู่ในภาคผนวก ง1. และมีขั้นตอนการสร้าง และการทำงานตามตารางที่ 4.3 ซึ่งมีรายละเอียดดังนี้

- เหตุการณ์ INSTANTIATE
  เมื่อเครื่องเสมือนวิเคราะห์ข้อมูลถูกสร้างขึ้นและติดตั้งระบบปฏิบัติการ Ubuntu cloud 16.04 LTS สำเร็จ จากนั้นจึงติดตั้งโปรแกรมที่จำเป็นสำหรับการทำงานของฟังก์ชันเสมือน โดยใช้โปรแกรมไพธอนรุ่น 2.7 สำหรับการประมวลผลข้อมูล และภาษา PHP สำหรับการทำงานของฟังก์ชันเสมือน และกำหนดเส้นทางให้กับเครื่องเสมือนเพื่อเชื่อมต่อกับโครงข่ายของระบบ CU-BEMS ผ่านทางส่วนต่อประสานโครงข่ายจุดที่สอง

- เหตุการณ์ CONFIGURE
  กำหนดการทำงานของฟังก์ชันเสมือนขึ้น โดยการวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 และแสดงผลด้วยโปรแกรมสร้างมโนภาพเชิงโต้ตอบจะกระทำวันละ 1 ครั้ง ขณะที่การตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบจะกระทำทุก 20 นาที

- เหตุการณ์ START
  เครื่องเสมือนร้องขอข้อมูลจากหน่วยเก็บข้อมูลของระบบ CU-BEMS และทำงานตามเวลาที่กำหนดไว้

ตารางที่ 4.3: ขั้นตอนการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

เหตุการณ์	เครื่องเสมือนวิเคราะห์ข้อมูล
INSTANTIATE	install_idaas.sh
CONFIGURE	server_configure.sh
START	start_idaas.sh

4.3.2 การแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศภายในอาคารผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบ

วิทยานิพนธ์ฉบับนี้นำงานวิจัย [21] ซึ่งเป็นโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบการจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 มาสร้างเป็นฟังก์ชันเสมือนด้วยโปรแกรมโอเพนบาตง และประยุกต์นำข้อมูลที่ผ่านการวิเคราะห์มาแสดงผลผ่านทางโปรแกรมสร้างมโนภาพเชิงโต้ตอบ โดยพัฒนาโปรแกรมด้วยภาษา PHP เพื่อสร้างภาพแสดงความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศในแต่ละพื้นที่ของชั้นที่ 13 อาคารเจริญวิศวกรรม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งมีขั้นตอนการทำงานตอบดังนี้

1. เครื่องเสมือนประมวลผลข้อมูลตัวรับรู้การเคลื่อนไหวของคน ข้อมูลอุณหภูมิ และข้อมูลการบริโภคพลังงานไฟฟ้าในแต่ละวัน จากนั้นนำมาหาค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศตามรูปแบบของงานวิจัย [21] ซึ่งจะประมวลผลเพื่อหาค่าความสูญเปล่าวันละ 1 ครั้ง

2. สร้างภาพแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศที่คำนวณได้ลงสู่แผนผังอาคารที่สร้างไว้และบันทึกเป็นไฟล์รูปภาพ โดยแบ่งเป็น 2 รูปตามผังอาคาร ได้แก่ บริเวณศูนย์เชี่ยวชาญเฉพาะด้านเทคโนโลยีโทรคมนาคม และห้องพักอาจารย์

3. บรรจุไฟล์รูปภาพที่ได้ขึ้นสู่เว็บไซต์ https://www.dropbox.com/

4. โปรแกรมสร้างมโนภาพเชิงโต้ตอบประสานข้อมูลจากเว็บไซต์ dropbox และนำมาแสดงผ่านจอภาพ

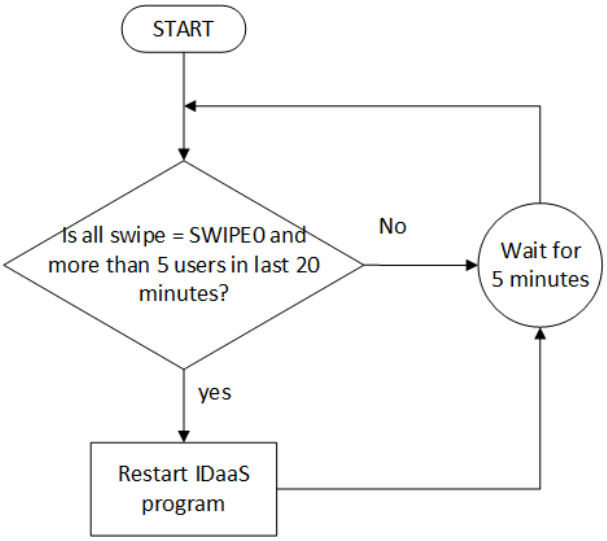
4.3.3 การตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ

เนื่องจากการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟท์ชั้น 13 อาคารเจริญวิศวกรรม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยมีปัญหาบ่อยครั้ง ซึ่งเกิดจากโปรแกรมสร้างมโนภาพเชิงโต้ตอบไม่ตอบสนองต่อการโบกมือของผู้ใช้งานหลังจากเข้าสู่แคนวาสปริยาย ทำให้ผู้ใช้งานไม่สามารถใช้งานโปรแกรมสร้างมโนภาพเชิงโต้ตอบได้ วิทยานิพนธ์นี้จึงพัฒนาฟังก์ชันเสมือนโดยอาศัยข้อมูลพฤติกรรมของผู้ใช้งานที่ถูกบันทึกในหน่วยเก็บข้อมูลของระบบ CU-BEMS มาใช้เพื่อประเมินการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ และแก้ไขปัญหา รวมถึงแจ้งเตือนปัญหาให้ผู้ดูแลระบบรับทราบ โดยขั้นตอนการตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบแสดงในรูปที่ 4.14 และมีรายละเอียดดังนี้

1. เครื่องเสมือนร้องขอข้อมูลจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยาย และจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้าจากระบบ CU-BEMS ภายในระยะเวลา 20 นาทีที่ผ่านมา

2. ในกรณีที่มีจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายมากกว่า 5 ครั้ง แต่ผู้ใช้งานไม่สามารถเข้าใช้งานแคนวาสอื่นได้นอกจากแคนวาสปริยายภายใน 20 นาที จะถือว่าโปรแกรมสร้างมโนภาพเชิงโต้ตอบไม่ตอบสนองต่อผู้ใช้งาน

3. เครื่องเสมือนจะบันทึกสถานะการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบเข้าสู่ระบบ CU-BEMS และส่งคำสั่งให้เริ่มการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบใหม่



รูปที่ 4.14: ขั้นตอนการตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ

4. เครื่องเสมือนส่งข้อความแจ้งเตือนผู้ดูแลระบบผ่านทางโปรแกรมประยุกต์ไลน์และอีเมล์

โดยฟังก์ชันเสมือนนี้จะทำการร้องขอข้อมูลจากระบบ CU-BEMS เพื่อตรวจสอบและประมวลผลทุก 5 นาที

บทที่ 5

ผลการสร้างและการทดสอบฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์

ในบทนี้ นำเสนอผลการทดสอบการสร้างฟังก์ชันเสมือน การวัดสมรรถนะการทำงานของฟังก์ชันเสมือน และประเมินการทำงานของฟังก์ชันเสมือนตามที่ได้อธิบายวิธีการทำงานไว้ในบทก่อนหน้า ซึ่งผลการทดสอบมีดังต่อไปนี้

5.1 ผลการทดสอบฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

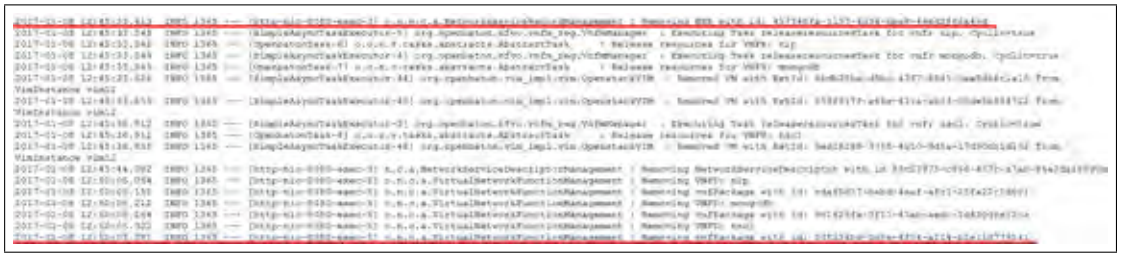
5.1.1 ผลการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

ผลการทดสอบการสร้างฟังก์ชันเสมือนขึ้นบนโปรแกรมโอเพนสแต็กโดยใช้โปรแกรมโอเพนบาตง โดยฟังก์ชันเสมือนนี้สร้างเครื่องเสมือนจำนวน 3 เครื่อง ได้แก่ เครื่องเสมือนเอ็นไอพี เครื่องเสมือนเอ็นเอสซีแอล และเครื่องเสมือนมองโกดีบี และมีขั้นตอนการสร้าง และการปรับแต่งพารามิเตอร์ตามหัวข้อที่ 4.1.1 จากการตรวจสอบระยะเวลาที่ใช้สำหรับการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M จากไฟล์บันทึกข้อมูลการทำงาน (log file) ของโปรแกรมโอเพนบาตง พบว่าสามารถสร้างฟังก์ชันเสมือนนี้ขึ้น และพร้อมใช้งานภายในระยะเวลาเฉลี่ย 262 วินาทีด้วยสมรรถนะของเครื่องคอมพิวเตอร์ที่ใช้งานดังตารางที่ 3.1 และรายละเอียดการสร้างฟังก์ชันเสมือนแสดงในรูปที่ 5.1

รูปที่ 5.1: ไฟล์บันทึกขั้นตอนการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

ในขณะที่การยุติการให้บริการฟังก์ชันเสมือนโดยใช้โปรแกรมโอเพนบาตง จากการตรวจสอบระยะเวลาที่ใช้ยุติฟังก์ชันเสมือน ซึ่งเป็นการทำลายเครื่องเสมือนเหล่านี้ทิ้ง ผลปรากฏว่าสามารถยุติการทำงาน

ของฟังก์ชันเสมือนนี้ภายในระยะเวลาเฉลี่ย 5 วินาทีด้วยสมรรถนะของเครื่องคอมพิวเตอร์ที่ใช้งานดังตารางที่ 3.1 และรายละเอียดการสร้างฟังก์ชันเสมือนแสดงในรูปที่ 5.2



รูปที่ 5.2: ไฟล์บันทึกขั้นตอนการยุติการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

5.1.2 ผลการประเมินฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

หลังจากโปรแกรมโอเพนบาตงสร้างฟังก์ชันเสมือนนี้สำเร็จและพร้อมใช้งาน เครื่องเสมือนเอ็นไอพีส่งข้อมูลร้องขอด้วยโพรโทคอล TRAP เพื่อจดทะเบียนสำหรับการแจ้งเตือนข้อมูลตัวรับรู้ไปที่หน่วยเก็บข้อมูลของระบบ CU-BEMS และระบุอายุของการแจ้งเตือนที่ point ID ตำแหน่งที่ 2 ดังรูปที่ 5.3(a) และส่งข้อมูล subscription เพื่อติดตามการเปลี่ยนแปลงของข้อมูลของเอกสารนี้ในฐานข้อมูลมองโกดีบีที่เครื่องเสมือนมองโกดีบีดังแสดงในรูปที่ 5.4 หลังจากนั้น เมื่อข้อมูลของ point ID ตำแหน่งที่ 2 มีการเปลี่ยนแปลงค่าของตัวรับรู้เป็น ON ที่เวลา 2016-12-27 16:30:26 หน่วยเก็บข้อมูลของระบบ CU-BEMS จะส่งข้อมูลไปที่เครื่องเสมือนเอ็นไอพี ซึ่งในวิทยานิพนธ์ฉบับนี้ทำการเชื่อมต่อระหว่างหมายเลขไอพีสาธารณะกับหมายเลขไอพีส่วนบุคคลของเครื่องเสมือนเอ็นไอพีโดยใช้อุปกรณ์จัดเส้นทางส่งต่อข้อมูลไปยังเครื่องเสมือน เมื่อเครื่องเสมือนเอ็นไอพีได้รับข้อมูลในรูปแบบ XML จากนั้นเปลี่ยนแปลงข้อมูลตามรูปแบบ JSON ดังรูปที่ 5.5 และส่งต่อข้อมูลไปที่เครื่องเสมือนเอ็นเอสซีแอลด้วยโพรโทคอล CREATE เครื่องเสมือนเอ็นเอสซีแอลทำหน้าที่รับ ตรวจสอบข้อมูลและส่งข้อมูลไปบันทึกที่ฐานข้อมูลมองโกดีบีที่เครื่องเสมือนมองโกดีบีที่เวลา 2016-12-27 16:30:28 ตามรูปที่ 5.6 เนื่องจากฐานข้อมูลมองโกดีบีใช้วิธีบันทึกเวลาด้วยเวลาปานกลางกรีนิช (GMT) ข้อมูลจึงถูกบันทึกที่เวลา 2016-12-27 09:30:28 ดังรูปที่ 5.7 หลังจากนั้นข้อมูลที่อัพเดตนี้จะถูกส่งกลับไปเครื่องเสมือนเอ็นไอพีด้วยโพรโทคอล NOTIFY เมื่อเครื่องเสมือนเอ็นเอสซีแอลรับค่ามาแล้วจึงส่งค่ากลับไปที่เครื่องเสมือนเอ็นไอพี และเครื่องเสมือนเอ็นไอพีปรับเปลี่ยนโครงสร้างของข้อมูลให้อยู่ในรูปแบบ XML ดังรูปที่ 5.8 และบันทึกค่าลงที่หน่วยเก็บข้อมูลของระบบ CU-BEMS ที่เวลา 2016-12-27 16:30:28 ดังรูปที่ 5.3(b) ดังนั้นการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M โดยการใช้ระบบ CU-BEMS และสร้างเครื่องเสมือนสำหรับแพลตฟอร์มโอเพนเอ็มทีซีพบว่า ฟังก์ชันเสมือนสามารถประสานข้อมูลระหว่างสองมาตรฐานได้ภายในเวลาเฉลี่ย 2 วินาที

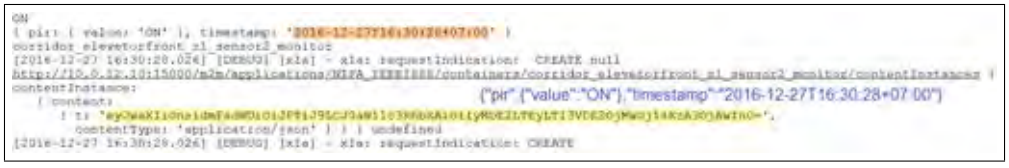
Time	Value	
2016-12-27 16:30:26	ON	1
2016-12-27 16:31:06	ON	
2016-12-27 16:31:48	OFF	2
2016-12-27 16:32:05	OFF	
2016-12-27 16:33:04	OFF	
2016-12-27 16:34:02	OFF	
2016-12-27 16:35:01	OFF	
2016-12-27 16:36:00	OFF	
2016-12-27 16:36:58	OFF	
2016-12-27 16:37:57	OFF	
2016-12-27 16:38:56	OFF	
2016-12-27 16:39:53	OFF	
2016-12-27 16:40:52	OFF	
2016-12-27 16:41:42	ON	3
2016-12-27 16:41:51	ON	
2016-12-27 16:42:24	OFF	4
2016-12-27 16:42:45	ON	5
2016-12-27 16:42:50	ON	
2016-12-27 16:43:29	OFF	6
2016-12-27 16:43:48	OFF	

Time	Value	
2016-12-27 16:30:28	ON	1
2016-12-27 16:31:49	OFF	2
2016-12-27 16:41:43	ON	3
2016-12-27 16:42:26	OFF	4
2016-12-27 16:42:46	ON	5
2016-12-27 16:43:30	OFF	6

(a) ข้อมูลตัวรับรู้การเคลื่อนไหวในระบบ CU-BEMS (b) ข้อมูลตัวรับรู้การเคลื่อนไหวที่เกิดจากฟังก์ชันเสมือน

รูปที่ 5.3: ข้อมูลตัวรับรู้การเคลื่อนไหวที่ถูกบันทึกในหน่วยเก็บข้อมูล CU-BEMS

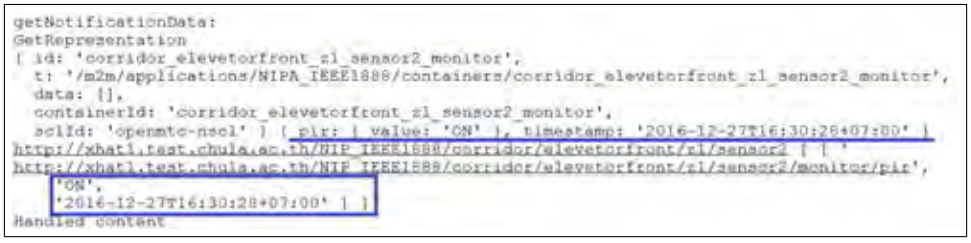
รูปที่ 5.4: ผลการส่งข้อมูล subscription เพื่อติดตามการเปลี่ยนแปลงข้อมูลของเอกสารที่ฐานข้อมูลมองโกดีบี



รูปที่ 5.5: ข้อมูลจากตัวรับรู้การเคลื่อนไหวของคนในระบบ CU-BEMS ที่เครื่องเสมือนเอ็นไอพี

รูปที่ 5.6: ข้อมูลตัวรับรู้การเคลื่อนไหวของคนที่ถูกส่งไปยังเครื่องเสมือนเอ็นเอสซีแอลเพื่อบันทึกลงฐานข้อมูลมองโกดีบีของแพลตฟอร์มโอเพนเอ็มทีซี

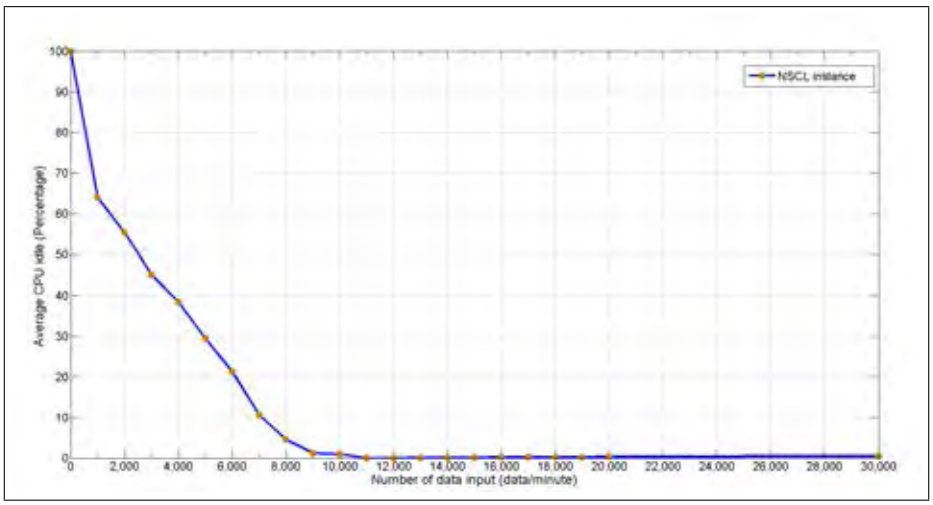
รูปที่ 5.7: ข้อมูลตัวรับรู้การเคลื่อนไหวของคนที่ถูกบันทึกในฐานข้อมูลมองโกดีบีที่เครื่องเสมือนมองโกดีบี



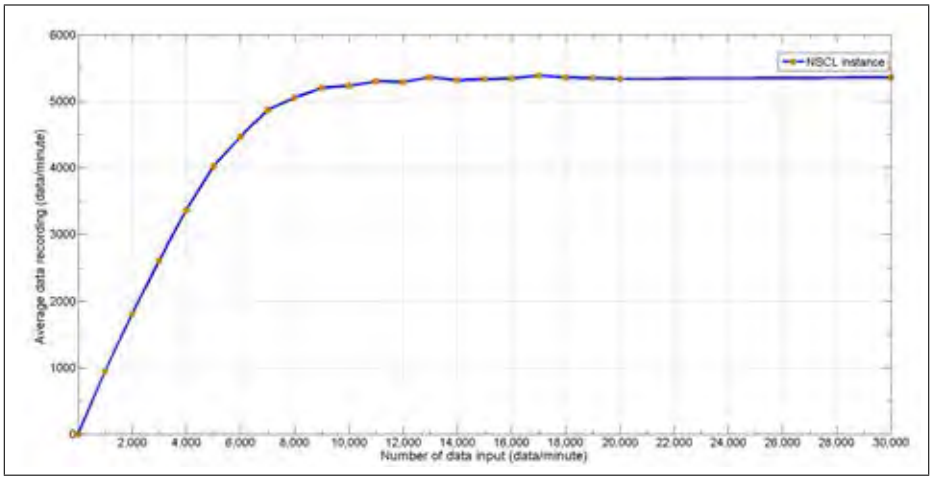
รูปที่ 5.8: ข้อมูลตัวรับรู้ที่ถูกส่งกลับไปยังเครื่องเสมือนเอ็นไอพีด้วยโพรโทคอล NOTIFY

5.1.3 ผลการประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอล

การประเมินสมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอลของแพลตฟอร์มโอเพนเอ็มทีซี กระทำโดยการเปลี่ยนแปลงความถี่การส่งข้อมูลเลียนแบบซึ่งเริ่มตั้งแต่การส่งข้อมูลด้วยความถี่ 1,000 ข้อมูลต่อนาทีจนถึงการส่งข้อมูลด้วยความถี่ 30,000 ข้อมูลต่อนาที โดยใช้ข้อมูลเลียนแบบจำนวน 30,000 ข้อมูล ผลปรากฏว่าเมื่อเพิ่มความถี่การส่งข้อมูลสูงขึ้น ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจะลดต่ำลง จนกระทั่งเมื่อความถี่การส่งข้อมูลมากกว่า 8,000 ข้อมูลต่อนาที ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลมีค่าเข้าใกล้ระดับ 0 เปอร์เซ็นต์ดังแสดงในรูปที่ 5.9 ขณะที่เมื่อประเมินค่าเฉลี่ยการบันทึกข้อมูลเลียนแบบที่ส่งเข้าไปสำเร็จของแพลตฟอร์มโอเพนเอ็มทีซี พบว่าสามารถบันทึกข้อมูลด้วยอัตราเร็วสูงสุด 5,380 ข้อมูลต่อนาทีดังแสดงในรูปที่ 5.10 ซึ่งสอดคล้องกับงานวิจัย [27]



รูปที่ 5.9: ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเมื่อมีการเปลี่ยนแปลงความถี่การส่งข้อมูลเลียนแบบ



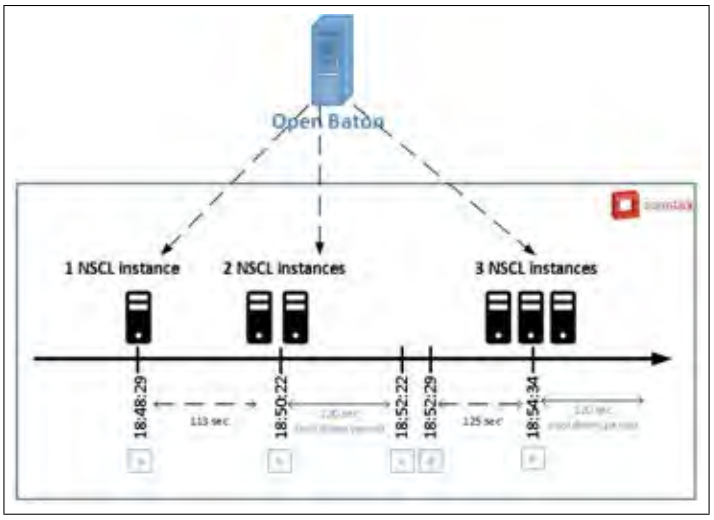
รูปที่ 5.10: ค่าเฉลี่ยการบันทึกข้อมูลเลียนแบบต่อนาทีของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อมีการเปลี่ยนแปลงความถี่การส่งข้อมูลเลียนแบบ

5.1.4 ผลการประเมินความสามารถการปรับจำนวนทรัพยากรแบบอัตโนมัติของแพลตฟอร์มโอเพนเอ็มทีซีโดยใช้โปรแกรมโอเพนบาตง

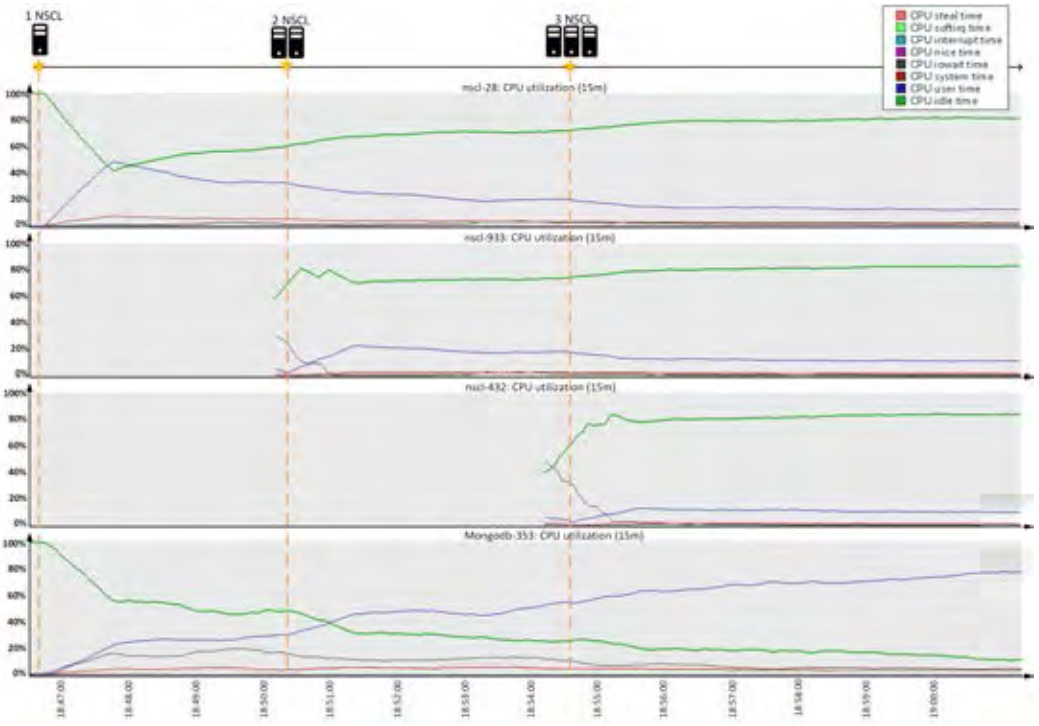
1. ผลการปรับเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติแปรผันตามระดับการทำงานของหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอล ซึ่งสอดคล้องกับปริมาณข้อมูลที่ถูกส่งเข้ามายังแพลตฟอร์มโอเพนเอ็มทีซี การทดสอบนี้ใช้การส่งข้อมูลตัวรับรู้เลียนแบบด้วยความถี่การส่งข้อมูล 12,000 ข้อมูลต่อนาทีไปยังแพลตฟอร์มโอเพนเอ็มทีซี ผลการทดสอบปรากฏว่าเมื่อเครื่องเสมือนเอ็นเอสซีแอลมีระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานลดลงต่ำกว่าระดับ 70 เปอร์เซ็นต์เป็นระยะเวลา 60 วินาที โปรแกรมโอเพนบาตงจึงสร้างเครื่องเสมือนเอ็นเอสซีแอลขึ้นแบบอัตโนมัติจำนวน 1 เครื่อง โดยเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สองติดตั้งระบบปฏิบัติการ ลงทะเบียนหมายเลขไอพีเข้าสู่บริการ LBaaS และพร้อมใช้งาน

ภายในระยะเวลา 113 วินาที เครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สองจึงทำหน้าที่รับข้อมูลตัวรับรู้เลียนแบบที่ส่งเข้ามาเช่นเดียวกับเครื่องเสมือนเอ็นเอสซีแอลเครื่องแรก โดยที่บริการ LBaaS ของโปรแกรมโอเพนสแต็กทำหน้าที่กระจายข้อมูลไปยังเครื่องเสมือนเอ็นเอสซีแอลทั้งสองเครื่อง ผลปรากฏว่าหลังจากเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สองเริ่มทำงานเป็นระยะเวลา 1 นาที ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่หนึ่งเพิ่มขึ้น 11.2 เปอร์เซ็นต์ จากระดับ 49.7 เปอร์เซ็นต์เป็นระดับ 60.9 เปอร์เซ็นต์ และระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนมองโกดีบีมีค่าลดลงประมาณ 19 เปอร์เซ็นต์ จากระดับ 49 เปอร์เซ็นต์เป็นระดับ 30 เปอร์เซ็นต์ ซึ่งแสดงให้เห็นว่าเมื่อเพิ่มเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สองเข้ามา และทำงานร่วมกับบริการ LBaaS ทำให้สามารถเพิ่มประสิทธิภาพการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีได้ และส่งผลให้หน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลทั้งสองเครื่องมีระดับที่ใกล้เคียงกัน

หลังจากนั้นข้อมูลตัวรับรู้เลียนแบบยังถูกส่งเข้าสู่แพลตฟอร์มโอเพนเอ็มทีซี และเมื่อสิ้นสุดระยะเวลาที่ยกเว้นการตรวจสอบค่าพารามิเตอร์ที่กำหนด ปรากฏว่าระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่หนึ่งยังต่ำกว่าระดับ 70 เปอร์เซ็นต์ ดังนั้นโปรแกรมโอเพนบาตงจึงสร้างเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สามขึ้นแบบอัตโนมัติ และพร้อมใช้งานภายในระยะเวลา 125 วินาที หลังจากเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สามทำงานเป็นระยะเวลา 1 นาที ส่งผลให้ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่หนึ่งเพิ่มขึ้น 6.4 เปอร์เซ็นต์ จากระดับ 69.8 เปอร์เซ็นต์สู่ระดับ 76.2 เปอร์เซ็นต์ และระดับของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สองเพิ่มขึ้น 6.6 เปอร์เซ็นต์ จากระดับ 72.3 เปอร์เซ็นต์สู่ระดับ 78.9 เปอร์เซ็นต์ ในขณะที่ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนมองโกดีบีลดลงประมาณ 8 เปอร์เซ็นต์ จากระดับ 28 เปอร์เซ็นต์มาที่ระดับ 20 เปอร์เซ็นต์ โดยช่วงเวลาการเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอลแสดงในรูปที่ 5.11 กราฟแสดงระดับหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลแสดงในรูปที่ 5.12 และหมายเลขไอพีของเครื่องเสมือนเอ็นเอสซีแอลที่ถูกเพิ่มเข้าสู่บริการ LBaaS ของโปรแกรมโอเพนสแต็กแสดงในรูปที่ 5.13



รูปที่ 5.11: ผังเวลาแสดงการเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติซึ่งควบคุมโดยโปรแกรมโอเพนบาตง



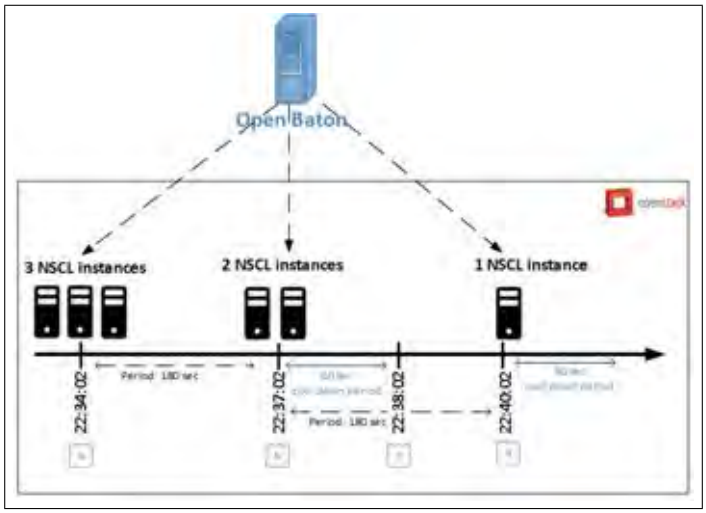
รูปที่ 5.12: ระดับหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลและเครื่องเสมือนมองโกดีบี

```
(neutron) lbaas-member-list pool1
+--------------------------------------+------+---------------+---------------+--------+--------------------------------------+----------------+
| id                                   | name | address       | protocol_port | weight | subnet_id                            | admin_state_up |
+--------------------------------------+------+---------------+---------------+--------+--------------------------------------+----------------+
| 82079220-8c18-43f7-b109-cb6120e58e01 |      | 10.0.12.9     |         15000 |      1 | a0f70b43-714f-45ff-a22b-c37f054dc19c | True           |
| 5222f2b5-05e1-4f56-8261-a493b1b7c27e |      | 10.0.12.7     |         15000 |      1 | a0f70b43-714f-45ff-a22b-c37f054dc19c | True           |
| f87201a7-494f-4835-a40e-196c46f9b64e |      | 10.0.12.5     |         15000 |      1 | a0f70b43-714f-45ff-a22b-c37f054dc19c | True           |
+--------------------------------------+------+---------------+---------------+--------+--------------------------------------+----------------+
(neutron)
```

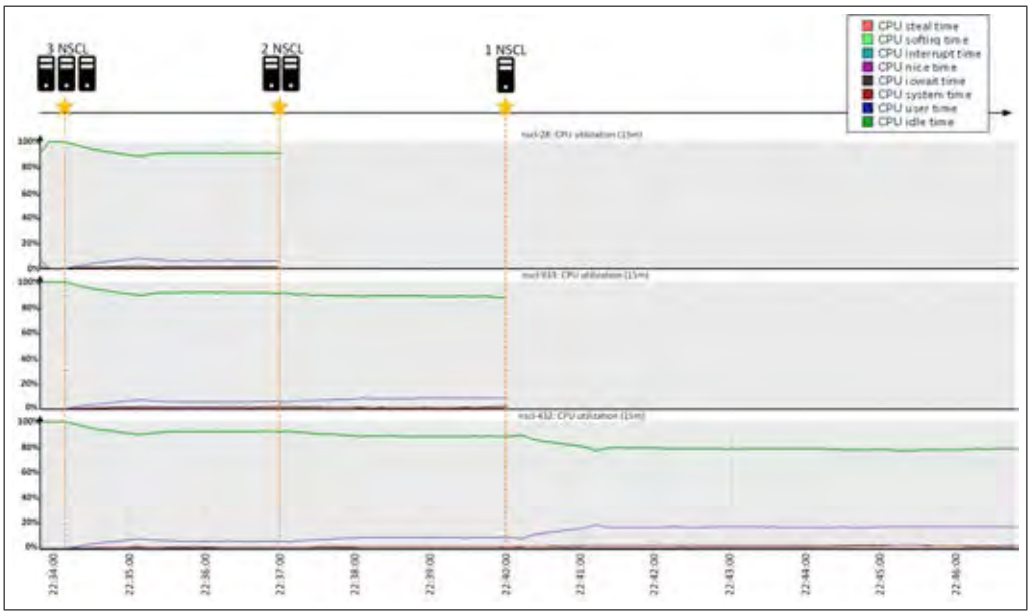
รูปที่ 5.13: หมายเลขไอพีของเครื่องเสมือนเอ็นเอสซีแอลที่ถูกเพิ่มเข้าสู่บริการ LBaaS

2. ผลการปรับลดจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติแปรผันตามระดับการทำงานของหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอล โดยใช้วิธีการส่งข้อมูลตัวรับรู้เลียนแบบที่มีความถี่การส่งข้อมูล 2,000 ข้อมูลต่อนาที ทำให้ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลทั้งสามเครื่องสูงกว่าระดับ 85 เปอร์เซ็นต์ที่กำหนดไว้เป็นระยะเวลา 180 วินาที โปรแกรมโอเพนบาตงจึงทำลายเครื่องเสมือนเอ็นเอสซีแอลทิ้งจำนวน 1 เครื่อง ซึ่งได้แก่ เครื่องเสมือนเอ็นเอสซีแอลเครื่องที่หนึ่ง หลังจากเครื่องเสมือนเครื่องที่หนึ่งถูกทำลาย 1 นาที ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สองลดลง 1.4 เปอร์เซ็นต์ จากระดับ 91.9 เปอร์เซ็นต์มาที่ระดับ 88.5 เปอร์เซ็นต์ และระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สามลดลง 3.3 เปอร์เซ็นต์ จากที่ระดับ 92.2 เปอร์เซ็นต์มาที่ระดับ 88.9 เปอร์เซ็นต์ และเมื่อสิ้นสุดระยะเวลาที่ยกเว้นการตรวจสอบค่าพารามิเตอร์ที่กำหนด ปรากฏว่าระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลที่เหลือทั้งสองเครื่องยังคงสูงกว่าระดับที่กำหนด

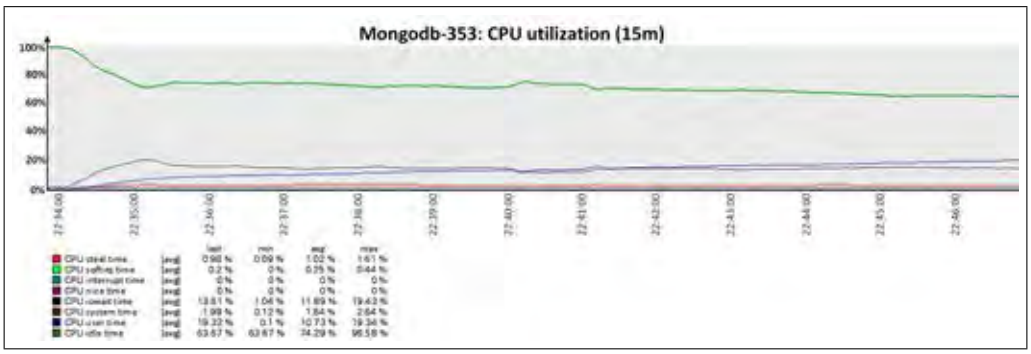
ดังนั้นโปรแกรมโอเพนบาตงจึงทำลายเครื่องเสมือนเอ็นเอสซีแอลทิ้งอีก 1 เครื่อง ได้แก่ เครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สอง ส่งผลทำให้ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลเครื่องที่สามลดลง 9 เปอร์เซ็นต์ จากที่ระดับ 87.9 เปอร์เซ็นต์มาที่ระดับ 79.9 เปอร์เซ็นต์ โดยช่วงเวลาการลดจำนวนเครื่องเสมือนเอ็นเอสซีแอลแสดงในรูปที่ 5.14 และกราฟแสดงระดับหน่วยประมวลผลกลางแสดงในรูปที่ 5.15 ขณะรูปที่ 5.16 แสดงระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนมองโกดีบีที่ระดับ 70 เปอร์เซ็นต์ ซึ่งเปลี่ยนแปลงเพียงเล็กน้อยเมื่อปรับลดจำนวนเครื่องเสมือนเอ็นเอสซีแอลลงแบบอัตโนมัติโดยโปรแกรมโอเพนบาตง



รูปที่ 5.14: ผังเวลาแสดงการลดจำนวนของเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติซึ่งควบคุมโดยโปรแกรมโอเพนบาตง



รูปที่ 5.15: ระดับหน่วยประมวลผลกลางของเครื่องเสมือนเอ็นเอสซีแอลเมื่อทำงานร่วมกับบริการ LBaaS ของโปรแกรมโอเพนสแต็ก



รูปที่ 5.16: ระดับหน่วยประมวลผลกลางของเครื่องเสมือนมองโกดีบีเมื่อเครื่องเสมือนเอ็นเอสซีแอลลดจำนวนลงแบบอัตโนมัติ

3. การทดสอบการปรับเพิ่มจำนวนเครื่องเสมือนมองโกดีบีแบบอัตโนมัติ เพื่อให้การทดสอบการปรับเพิ่มจำนวนเครื่องเสมือนมองโกดีบีสามารถเห็นผลได้ชัดเจน วิทยานิพนธ์ฉบับนี้จึงทำการป้อนไฟล์ข้อมูลไฟล์วิดีโอขนาดใหญ่เข้าสู่ฐานข้อมูลมองโกดีบีในเครื่องเสมือนมองโกดีบีเพื่อให้การทดสอบเป็นไปด้วยความรวดเร็วขึ้น โดยแบ่งผลการทดสอบออกเป็นดังนี้

   (a) เมื่อป้อนข้อมูลเข้าสู่ฐานข้อมูลมองโกดีบีจำนวน 19.84 กิกะไบต์ ดังแสดงในรูปที่ 5.17 ส่งผลให้พื้นที่จัดเก็บข้อมูลของเครื่องเสมือนมองโกดีบีเหลือต่ำกว่า 12 กิกะไบต์ (30 เปอร์เซ็นต์ของพื้นที่จัดเก็บข้อมูลของเครื่องเสมือนมองโกดีบีตามข้อกำหนด) โปรแกรมโอเพนบาตงจึงสร้างเครื่องเสมือนมองโกดีบีเพิ่มขึ้นจำนวน 1 เครื่องแบบอัตโนมัติโดยใช้ระยะเวลา 98 วินาที

```
mongos> db.fs.chunks.getShardDistribution()

Shard MzI at MzI/mongodb-391:27017
 data : 19.84GiB docs : 81598 chunks : 593
 estimated data per chunk : 34.27MiB
 estimated docs per chunk : 137

Totals
 data : 19.84GiB docs : 81598 chunks : 593
 Shard MzI contains 100% data, 100% docs in cluster, avg obj size on shard : 255KiB
```

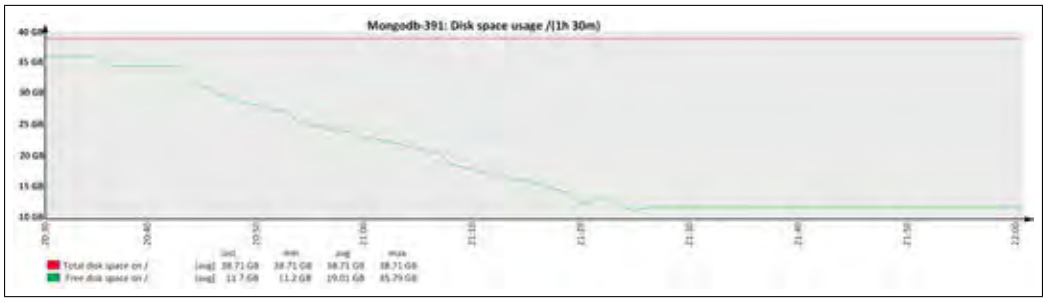
รูปที่ 5.17: การจัดเก็บข้อมูลในฐานข้อมูลมองโกดีบีเมื่อมีเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่อง

   (b) หลังจากเครื่องเสมือนมองโกดีบีเครื่องที่สองพร้อมให้บริการ เครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งทำการถ่ายโอนข้อมูลประมาณ 50 เปอร์เซ็นต์ที่มีอยู่ในฐานข้อมูลมองโกดีบีด้วยวิธีการขยายฐานข้อมูลมองโกดีบี (sharding) [26] ไปยังฐานข้อมูลมองโกดีบีของเครื่องเสมือนมองโกดีบีเครื่องที่สองแบบอัตโนมัติ หลังการถ่ายโอนข้อมูลเสร็จสิ้น จำนวนข้อมูลในฐานข้อมูลมองโกดีบีของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งมีขนาด 10.46 กิกะไบต์ ขณะที่จำนวนข้อมูลในฐานข้อมูลมองโกดีบีของเครื่องเสมือนมองโกดีบีเครื่องที่สองมีขนาด 9.61 กิกะไบต์ดังรูปที่ 5.18 ขณะที่รูปที่ 5.19 และรูปที่ 5.20 แสดงพื้นที่จัดเก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่ง และเครื่องที่สองหลังจากการถ่ายโอนข้อมูลเสร็จสิ้นตามลำดับ ขณะที่รูปที่ 5.21 แสดงกราฟการถ่ายโอนข้อมูลจากเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งไปยังเครื่องเสมือนมองโกดีบีเครื่องที่สอง และรูป

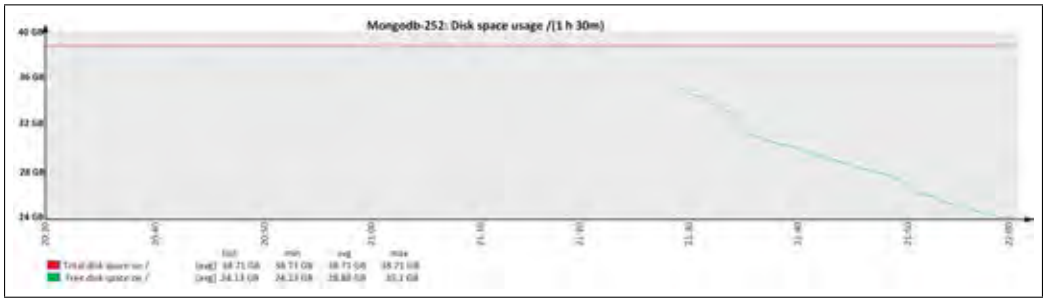
ที่ 5.22 แสดงกราฟการรับถ่ายโอนข้อมูลจากเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งไปยังเครื่องเสมือนมองโกดีบีเครื่องที่สอง จากรูปที่ 5.19 และ 5.20 แสดงให้เห็นว่า แม้ว่าเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งจะถ่ายโอนข้อมูลไปที่เครื่องเสมือนมองโกดีบีเครื่องที่สองเสร็จสิ้นแล้ว แต่พื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งไม่เพิ่มขึ้นแต่อย่างใด นอกจากผู้ดูแลระบบจะปิดการทำงานของฐานข้อมูลมองโกดีบี และทำการจัดเรียงฐานข้อมูลใหม่

```
mongos> db.fs.chunks.getShardDistribution()

Shard MzI at MzI/mongodb-391:27017
 data : 10.46GiB docs : 43049 chunks : 305
 estimated data per chunk : 35.15MiB
 estimated docs per chunk : 141

Shard ZTh at ZTh/mongodb-252:27017
 data: 9.61GiB docs : 39515 chunks : 304
 estimated data per chunk : 32.37MiB
 estimated docs per chunk : 129

Totals
 data : 20.08GiB docs : 82564 chunks : 609
 Shard MzI contains 52.13% data, 52.14% docs in cluster, avg obj size on shard : 255KiB
 Shard ZTh contains 47.86% data, 47.85% docs in cluster, avg obj size on shard : 255KiB
```

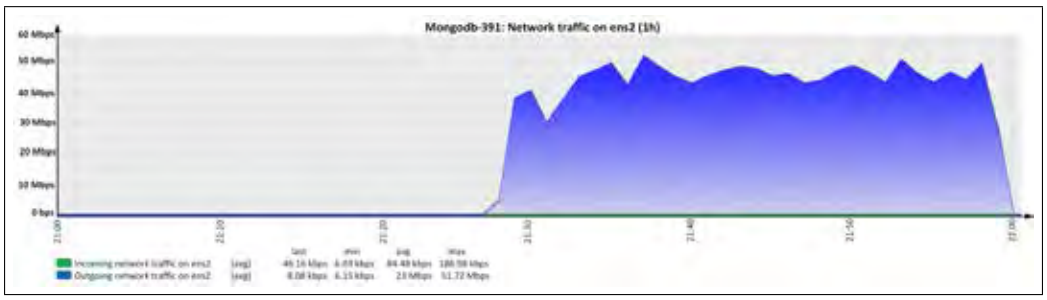
รูปที่ 5.18: การจัดเก็บข้อมูลในฐานข้อมูลมองโกดีบีเมื่อมีเครื่องเสมือนมองโกดีบีจำนวน 2 เครื่อง



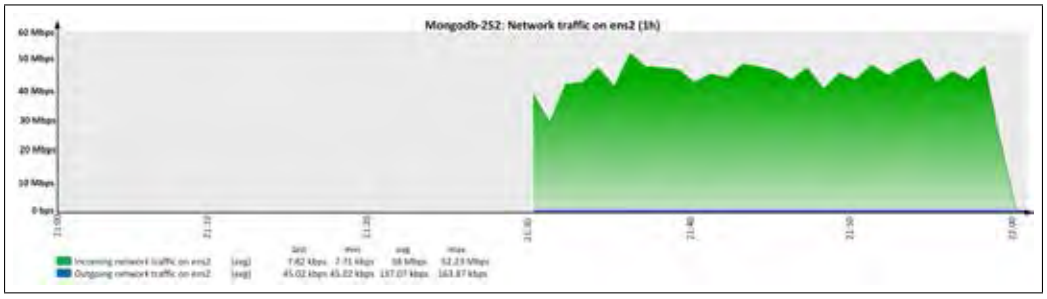
รูปที่ 5.19: ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่ง



รูปที่ 5.20: ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่สอง



รูปที่ 5.21: การถ่ายโอนข้อมูลจากเครื่องเสมือนมองโกดีบีเครื่องแรกไปสู่เครื่องเสมือนมองโกดีบีเครื่องที่สอง

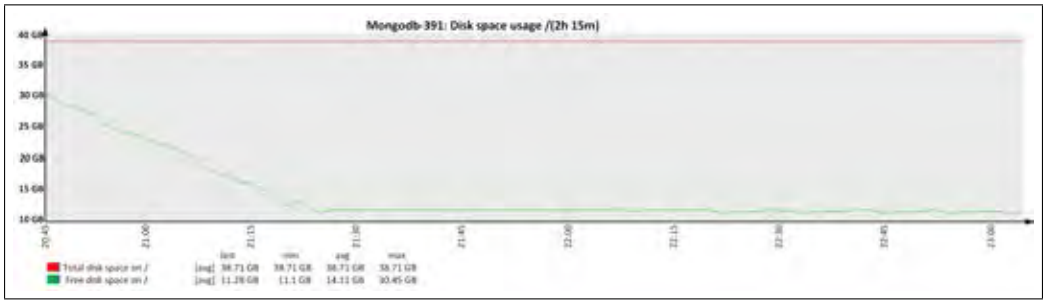


รูปที่ 5.22: การรับถ่ายโอนข้อมูลจากเครื่องเสมือนมองโกดีบีเครื่องแรกไปสู่เครื่องเสมือนมองโกดีบีเครื่องที่สอง

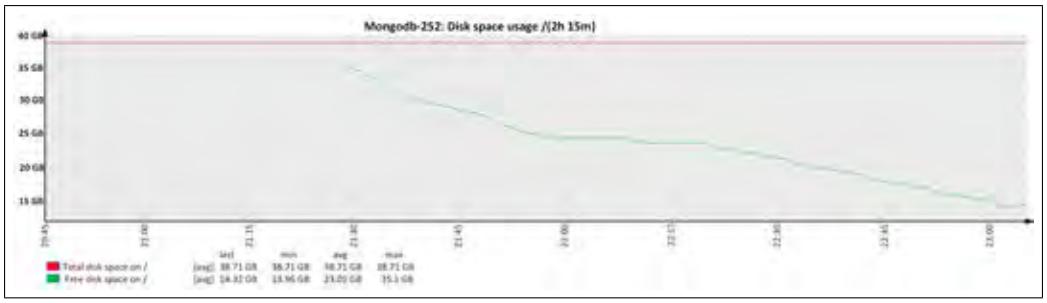
(c) เพื่อทดสอบระบบการขยายฐานข้อมูลมองโกดีบีเพิ่มเติมจากข้อ (b) จึงป้อนข้อมูลเพิ่มเข้าสู่ฐานข้อมูลมองโกดีบีจำนวน 20 กิกะไบต์ ซึ่งข้อมูลเหล่านี้จะถูกกระจายเข้าไปเก็บยังเครื่องเสมือนมองโกดีบีทั้งสองเครื่องด้วยวิธีการขยายฐานข้อมูลมองโกดีบี ทำให้เครื่องเสมือนมองโกดีบีจะบันทึกข้อมูลลงในฐานข้อมูลมองโกดีบีเพิ่มขึ้นเครื่องละประมาณ 10 กิกะไบต์ ดังแสดงในรูปที่ 5.23 อย่างไรก็ตามรูปที่ 5.24 แสดงให้เห็นว่าพื้นที่จัดเก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งเปลี่ยนแปลงเพียงเล็กน้อยเท่านั้น ขณะที่รูปที่ 5.25 แสดงให้เห็นว่ามีการใช้พื้นที่จัดเก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่สองเพิ่มขึ้นประมาณ 10 กิกะไบต์ซึ่งเท่ากับจำนวนข้อมูลที่ส่งเข้ามา จึงสรุปได้ว่า แม้ฐานข้อมูลมองโกดีบีจะไม่คืนพื้นที่เก็บข้อมูลกลับคืนให้ระบบปฏิบัติการหลังการถ่ายโอนข้อมูลให้ฐานข้อมูลมองโกดีบีของเครื่องเสมือนมองโกดีบีเครื่องที่สองเสร็จสิ้น แต่เมื่อมีการส่งข้อมูลใหม่เข้าไป ฐานข้อมูลมองโกดีบีของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งจะบันทึกข้อมูลใหม่ไปยังพื้นที่จัดเก็บข้อมูลที่เคยใช้งานไว้ก่อนจนเต็มแล้วจึงจะไปใช้งานพื้นที่เก็บที่เหลืออยู่ของเครื่องเสมือน

```
mongos> db.fs.chunks.getShardDistribution()

Shard MzI at MzI/mongodb-391:27017
 data : 20.49GiB docs : 84281 chunks : 610
 estimated data per chunk : 34.41MiB
 estimated docs per chunk : 138

Shard ZTh at ZTh/mongodb-252:27017
 data: 19.66GiB docs : 80846 chunks : 609
 estimated data per chunk : 33.06MiB
 estimated docs per chunk : 132

Totals
 data : 40.16GiB docs : 165127 chunks : 1219
 Shard MzI contains 51.04% data, 51.04% docs in cluster, avg obj size on shard : 255KiB
 Shard ZTh contains 48.95% data, 48.95% docs in cluster, avg obj size on shard : 255KiB
```

รูปที่ 5.23: การจัดเก็บข้อมูลในฐานข้อมูลมองโกดีบีเมื่อมีเครื่องเสมือนมองโกดีบีจำนวน 2 เครื่อง หลังการทดสอบส่งข้อมูลครั้งที่สอง



รูปที่ 5.24: ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่ง หลังการทดสอบส่งข้อมูลครั้งที่สอง



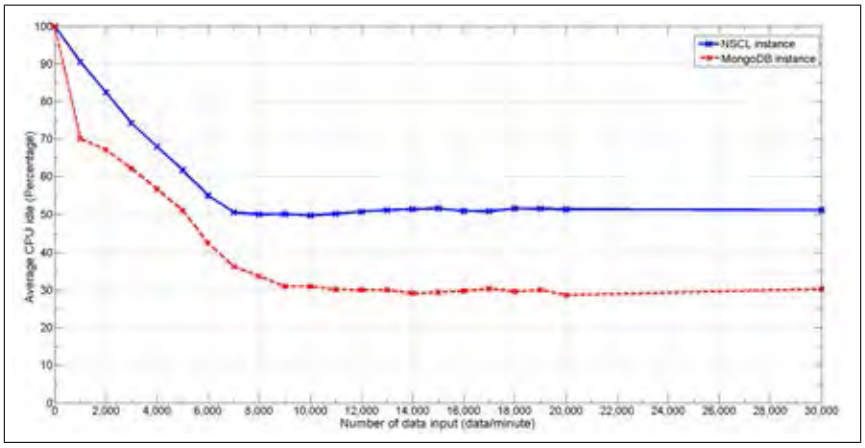
รูปที่ 5.25: ข้อมูลพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีเครื่องที่สอง หลังการทดสอบส่งข้อมูลครั้งที่สอง

5.1.5 ผลการประเมินสมรรถนะการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีหลังจากเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอล

จากการที่โปรแกรมโอเพนบาตงสามารถปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลได้ และทำงานร่วมกับบริการ LBaaS ของโปรแกรมโอเพนสแต็กเพื่อกระจายงานไปยังเครื่องเสมือนเอ็นเอสซีแอล ทำให้แพลตฟอร์มโอเพนเอ็มทีซีสามารถรองรับข้อมูลได้เพิ่มมากยิ่งขึ้น ดังนั้นในหัวข้อนี้จึงเป็นการวัด

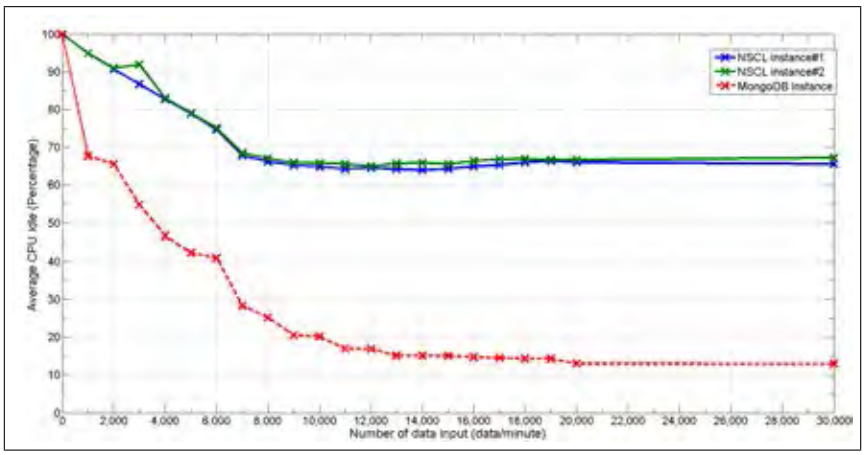
ประสิทธิภาพของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อใช้เครื่องเสมือนเอ็นเอสซีแอลตั้งแต่ 1 เครื่องขึ้นไป โดยแบ่งการวัดประสิทธิภาพได้ดังนี้

1. การวัดระดับของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 1 เครื่อง และเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่องแสดงในรูปที่ 5.26 ผลปรากฏว่าเมื่อส่งข้อมูลเลียนแบบด้วยความถี่ที่สูงกว่า 7,000 ข้อมูลต่อนาที ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจะเริ่มอิ่มตัวที่ระดับ 50 เปอร์เซ็นต์ และไม่มีการเปลี่ยนแปลงอย่างมีนัยะสำคัญ แม้ว่าจะเพิ่มความถี่การส่งข้อมูลสูงขึ้น ขณะที่ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนมองโกดีบีที่จะอิ่มตัวที่ระดับ 30 เปอร์เซ็นต์



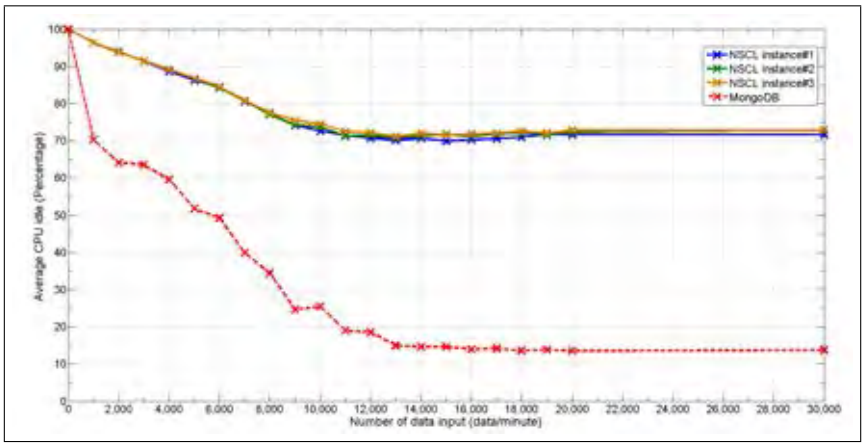
รูปที่ 5.26: ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 1 เครื่องและเครื่องเสมือนมองโกดีบีเมื่อเปลี่ยนแปลงความถี่การส่งข้อมูล

2. การวัดระดับของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 2 เครื่อง และเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่องแสดงในรูปที่ 5.27 ผลปรากฏว่าเมื่อส่งข้อมูลเลียนแบบด้วยความถี่ที่สูงกว่า 11,000 ข้อมูลต่อนาที ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลทั้งสองเครื่องจะอิ่มตัวที่ระดับ 65 เปอร์เซ็นต์ และไม่มีการเปลี่ยนแปลงอย่างมีนัยะสำคัญ แม้ว่าจะเพิ่มความถี่การส่งข้อมูลสูงขึ้น ขณะที่ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนมองโกดีบีจะอิ่มตัวที่ระดับ 15 เปอร์เซ็นต์



รูปที่ 5.27: ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 2 เครื่องและเครื่องเสมือนมองโกดีบีเมื่อเปลี่ยนแปลงความถี่การส่งข้อมูล

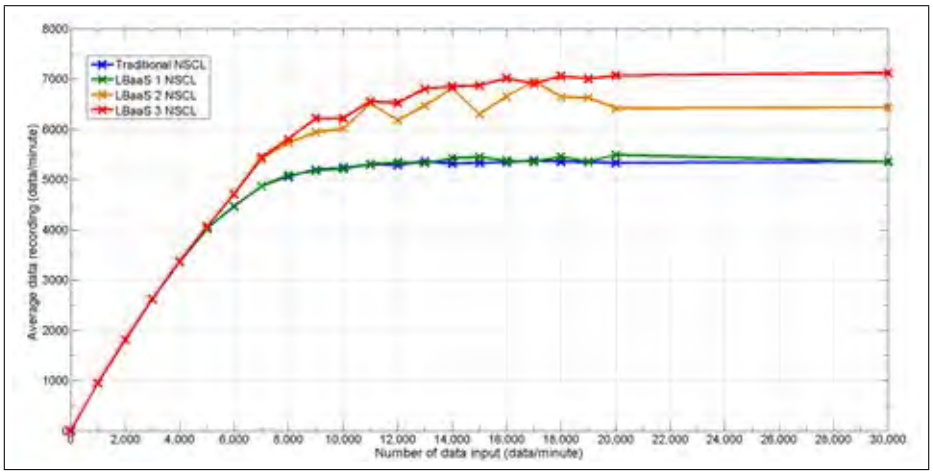
3. การวัดระดับของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 3 เครื่องและเครื่องเสมือนมองโกดีบีจำนวน 1 เครื่องแสดงในรูปที่ 5.28 ผลปรากฏว่าเมื่อส่งข้อมูลเลียนแบบด้วยความถี่ที่สูงกว่า 13,000 ข้อมูลต่อนาที ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลทั้งสามเครื่องจะอิ่มตัวที่ระดับ 70 เปอร์เซ็นต์ และไม่มีการเปลี่ยนแปลงอย่างมีนัยะสำคัญ แม้ว่าจะเพิ่มความถี่การส่งข้อมูลสูงขึ้น ขณะที่ระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนมองโกดีบีจะอิ่มตัวที่ระดับ 15 เปอร์เซ็นต์



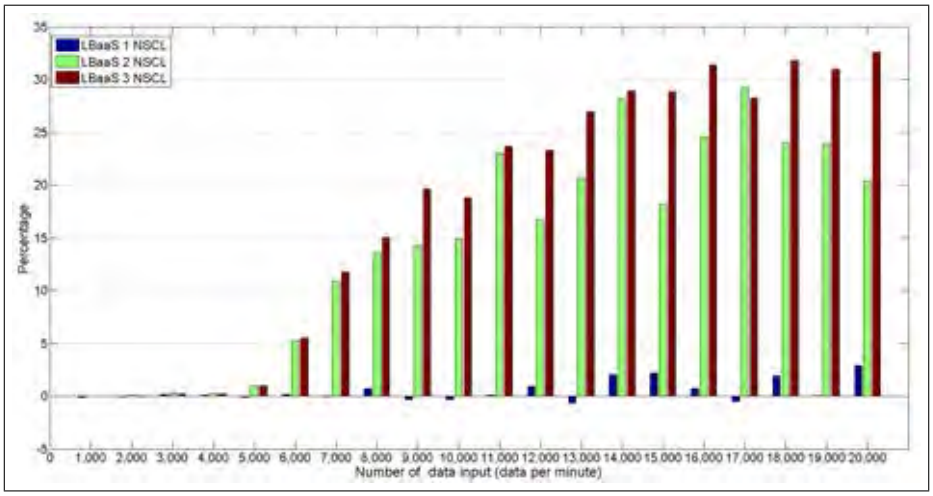
รูปที่ 5.28: ค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องเสมือนเอ็นเอสซีแอลจำนวน 3 เครื่องและเครื่องเสมือนมองโกดีบีเมื่อเปลี่ยนแปลงความถี่การส่งข้อมูล

4. การเปรียบเทียบความสามารถการบันทึกข้อมูลของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อเพิ่มจำนวนเครื่องเสมือนเอ็นเอสซีแอล และทำงานร่วมกับบริการ LBaaS ของโปรแกรมโอเพนสแต็ก ผลปรากฏว่าเมื่อใช้เครื่องเสมือนเอ็นเอสซีแอลจำนวน 3 เครื่องสามารถบันทึกข้อมูลที่อัตราเร็วสูงสุด 7,119 ข้อมูลต่อนาที ขณะที่เครื่องเสมือนเอ็นเอสซีแอลจำนวน 2 เครื่องสามารถบันทึกข้อมูลที่อัตราเร็วสูงสุด 6,959 ข้อมูลต่อนาที และเครื่องเสมือนเอ็นเอสซีแอลจำนวน 1 เครื่อง

สามารถบันทึกข้อมูลที่อัตราเร็วสูงสุด 5,487 ข้อมูลต่อนาที ซึ่งใกล้เคียงกับผลการทดสอบเครื่องเสมือนเอ็นเอสซีแอลในหัวข้อที่ 5.1.3 ที่อัตราเร็วสูงสุด 5,380 ข้อมูลต่อวินาทีดังรูปที่ 5.29 และเมื่อเปรียบเทียบประสิทธิภาพของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อใช้เครื่องเสมือนเอ็นเอสซีแอลตั้งแต่ 1 เครื่องขึ้นไปทำงานร่วมกับบริการ LBaaS ของโปรแกรมโอเพนสแต็กเทียบกับเครื่องเสมือนเอ็นเอสซีแอลแบบดั้งเดิมซึ่งรวมส่วนประมวลผลและส่วนจัดเก็บข้อมูลไว้ในเครื่องเดียวกัน ผลปรากฏว่าเมื่อเพิ่มความถี่การส่งข้อมูลเลียนแบบสูงขึ้น เครื่องเสมือนเอ็นเอสซีแอลจำนวน 2 เครื่องสามารถบันทึกข้อมูลได้เร็วกว่าเครื่องเสมือนเอ็นเอสซีแอลแบบดั้งเดิมถึง 29.2 เปอร์เซ็นต์ที่ความถี่การส่งข้อมูล 17,000 ข้อมูลต่อนาที ขณะที่เครื่องเสมือนเอ็นเอสซีแอลจำนวน 3 เครื่องสามารถบันทึกข้อมูลได้เร็วกว่าเครื่องเสมือนเอ็นเอสซีแอลแบบดั้งเดิมถึง 32.9 เปอร์เซ็นต์ที่ความถี่ในการส่งข้อมูล 20,000 ข้อมูลต่อนาที แต่อย่างไรก็ตามเมื่อเพิ่มความถี่การส่งข้อมูลสูงขึ้น ผลปรากฏว่าไม่พบความแตกต่างที่มากขึ้นแต่อย่างใดดังรูปที่ 5.30



รูปที่ 5.29: ค่าเฉลี่ยของการบันทึกข้อมูลเลียนแบบของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อเปลี่ยนแปลงจำนวนเครื่องเสมือนเอ็นเอสซีแอลและเปลี่ยนแปลงแปลงความถี่การส่งข้อมูล



รูปที่ 5.30: ประสิทธิภาพการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีเมื่อใช้เครื่องเสมือนเอ็นเอสซีแอลทำงานร่วมกับบริการ LBaaS เทียบกับเครื่องเสมือนเอ็นเอสซีแอลแบบดั้งเดิม

5.2 ผลการทดสอบฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

5.2.1 ผลการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

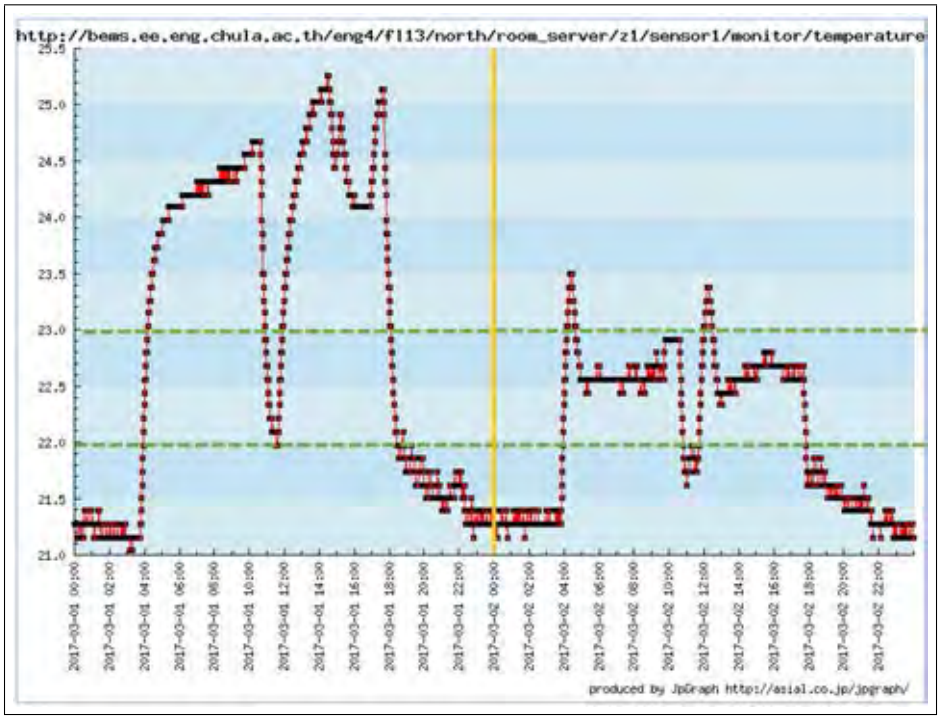
ผลการสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite ด้วยโปรแกรมโอเพนบาตง เครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ที่ถูกสร้างขึ้นจะมีส่วนต่อประสานโครงข่ายจำนวน 2 จุด จุดแรกทำหน้าที่ติดต่อกับหน่วยเก็บข้อมูลของระบบ CU-BEMS และการเข้าใช้งาน จุดที่สองทำหน้าที่ติดต่อกับโครงข่าย ECHONET Lite เพื่อร้องขอข้อมูล และส่งคำสั่งไปยังอุปกรณ์ที่อยู่ในโครงข่ายนี้ดังแสดงในรูปที่ 5.31 ซึ่งโปรแกรมโอเพนบาตงสามารถสร้างฟังก์ชันเสมือนได้ภายในเวลา 210 วินาที และสามารถยุติการทำงานฟังก์ชันเสมือนได้ภายในเวลา 1 วินาที

รูปที่ 5.31: การเชื่อมต่อโครงข่ายของเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์บนแพลตฟอร์มโอเพนสแต็ก

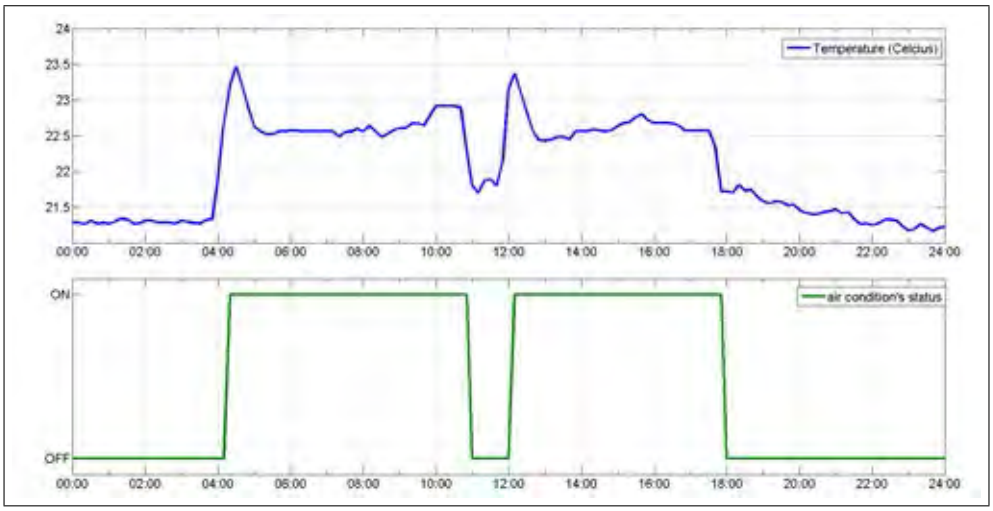
5.2.2 ผลการทำงานของฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

เมื่อเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์เริ่มทำงาน เครื่องเสมือนจะร้องขอข้อมูลอุณหภูมิภายในระยะเวลา 10 นาทีล่าสุดของห้องคลัสเตอร์จากหน่วยเก็บข้อมูลของระบบ CU-BEMS เพื่อนำไปคำนวณค่าเฉลี่ย และพิจารณาการทำงานของเครื่องปรับอากาศที่เชื่อมต่อกับโนด ECHONET Lite ซึ่งถูกติดตั้งอยู่ภายในห้องคลัสเตอร์ตามระดับของอุณหภูมิที่กำหนดไว้ในไฟล์ข้อมูลรายละเอียดบริการโครงข่ายฟังก์ชันเสมือนดังแสดงในรูปที่ 4.9 การทดสอบการทำงานของฟังก์ชันเสมือนนี้เริ่มตั้งแต่เวลา 00.00 นาฬิกา จนถึงเวลา 24.00 นาฬิกาของวันที่ 2 มีนาคม ค.ศ. 2017 รูปที่ 5.32 แสดงการเปรียบเทียบอุณหภูมิที่บันทึกจากตัวรับรู้อุณหภูมิภายในห้องคลัสเตอร์ของวันที่ 1 และวันที่

2 ซึ่งเปรียบเทียบให้เห็นว่าเมื่อฟังก์ชันเสมือนควบคุมการทำงานของเครื่องปรับอากาศในวันที่ 2 โดยเปิดเครื่องปรับอากาศในช่วงที่อุณหภูมิสูงกว่า 23 องศาเซลเซียส ส่งผลให้อุณหภูมิลดต่ำลงกว่า 23 องศาเซลเซียสในช่วงเวลา 04:20 ถึงเวลา 10:50 นาฬิกา และช่วงเวลา 12:10 ถึงเวลา 17:50 นาฬิกา ขณะที่ช่วงเวลาอื่น ๆ เครื่องปรับอากาศถูกปิดเนื่องจากอุณหภูมิเฉลี่ยต่ำกว่า 22 องศาเซลเซียส รูปที่ 5.33 แสดงค่าอุณหภูมิเฉลี่ยทุก 10 นาทีเทียบกับสถานะของเครื่องปรับอากาศภายในระยะเวลา 1 วันที่ฟังก์ชันเสมือนนี้ทำงาน ซึ่งแสดงให้เห็นว่าฟังก์ชันเสมือนนี้สามารถช่วยควบคุมอุณหภูมิของห้องคลัสเตอร์ได้



รูปที่ 5.32: อุณหภูมิของห้องคลัสเตอร์ระหว่างวันที่ไม่เปิดใช้งานและวันที่เปิดใช้งานฟังก์ชันเสมือน

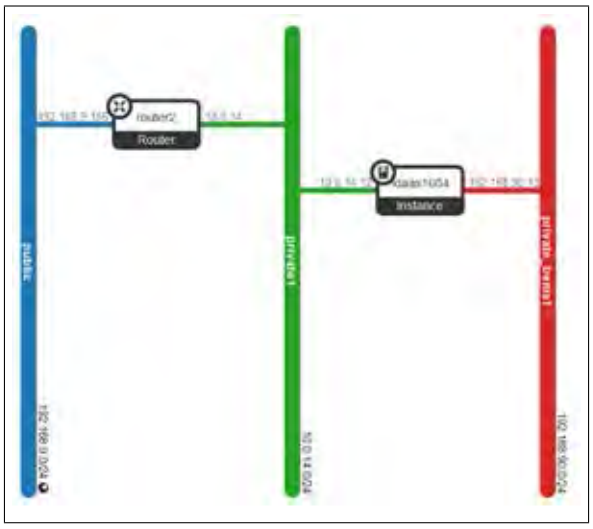


รูปที่ 5.33: อุณหภูมิของห้องคลัสเตอร์และสถานะของเครื่องปรับอากาศเมื่อเปิดใช้งานฟังก์ชันเสมือน

5.3 ผลการทดสอบฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

5.3.1 ผลการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

ผลการสร้างฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 ด้วยโปรแกรมโอเพนบาตง เครื่องเสมือนสำหรับการวิเคราะห์ข้อมูลถูกสร้าง และเชื่อมต่อกับโครงข่ายส่วนบุคคล 2 โครงข่าย ได้แก่ โครงข่ายเสมือน private1 เพื่อใช้เชื่อมต่อกับอินเทอร์เน็ตและการเข้าถึง และโครงข่ายเสมือน private_bems1 เพื่อเชื่อมต่อกับโครงข่ายของระบบ CU-BEMS ดังรูปที่ 5.34 เพื่อร้องขอข้อมูลจากหน่วยเก็บข้อมูลของระบบ CU-BEMS โดยโปรแกรมโอเพนบาตงสามารถสร้างฟังก์ชันเสมือนนี้สำเร็จภายในเวลา 450 วินาทีและสามารถทำลายฟังก์ชันเสมือนนี้ได้ภายในเวลา 1 วินาที

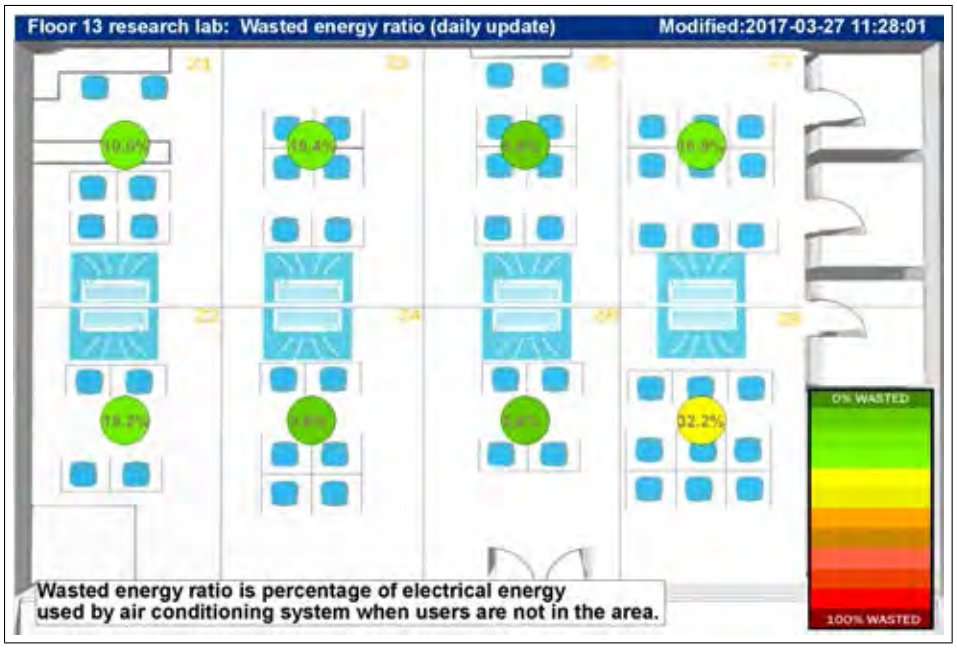


รูปที่ 5.34: การเชื่อมต่อโครงข่ายของเครื่องเสมือนสำหรับการวิเคราะห์ข้อมูลบนแพลตฟอร์มโอเพนสแต็ก

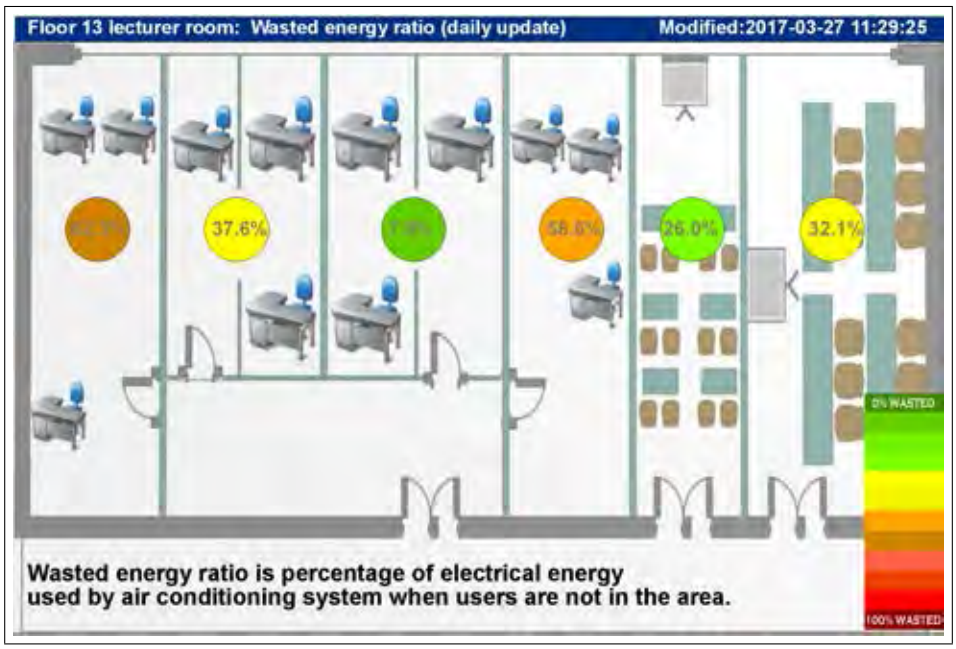
5.3.2 ผลการแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศภายในอาคารผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบ

เครื่องเสมือนนี้ประมวลผลข้อมูลตัวรับรู้การเคลื่อนไหวของคน ข้อมูลอุณหภูมิ รวมถึงข้อมูลการบริโภคพลังไฟฟ้า จากนั้นนำมาหาค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศในแต่ละวัน โดยจะประมวลผลที่เวลา 01:00 นาฬิกาของทุกวัน หลังจากนั้นนำข้อมูลที่คำนวณได้มาวาดลงแผนผังของอาคารชั้น 13 และแสดงผลด้วยโปรแกรมสร้างมโนภาพเชิงโต้ตอบ โดยแบ่งเป็น 2 ภาพตามแผนผังของอาคารประกอบด้วยรูปที่ 5.35 แสดงอัตราส่วนของค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศเทียบกับข้อมูลการบริโภคพลังงานไฟฟ้าของเครื่องปรับอากาศภายในศูนย์เชี่ยวชาญเฉพาะด้านเทคโนโลยีโทรคมนาคม และรูปที่ 5.36 แสดงค่าอัตราส่วนนี้

บริเวณห้องพักอาจารย์ และห้องเรียนชั้น 13 ผ่านทางโปรแกรมสร้างมโนภาพเชิงโต้ตอบดังรูปที่ 5.37



รูปที่ 5.35: ค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศบริเวณศูนย์เชี่ยวชาญเฉพาะด้านเทคโนโลยีโทรคมนาคม

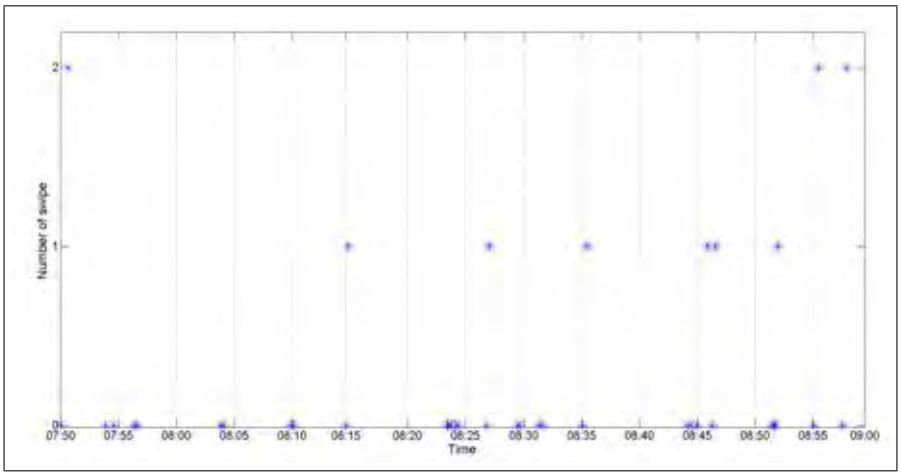


รูปที่ 5.36: ค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศบริเวณห้องพักอาจารย์และห้องเรียนชั้น 13

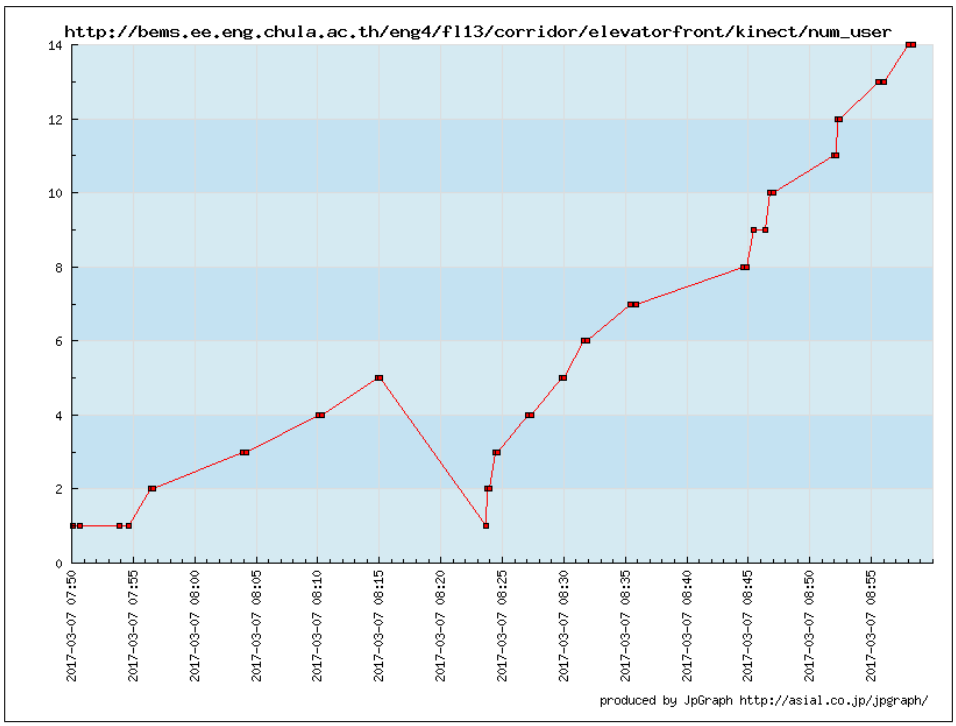
รูปที่ 5.37: การนำเสนอค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศผ่านทางโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟท์ชั้น 13

5.3.3 ผลการประเมินการตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ

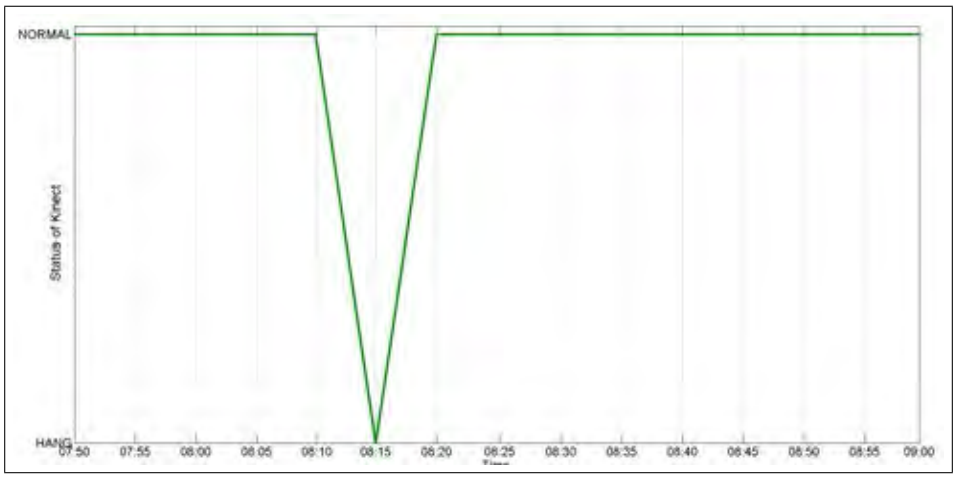
เมื่อนำฟังก์ชันเสมือนนี้มาประยุกต์ใช้ตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบที่ติดตั้งบริเวณหน้าลิฟต์ชั้น 13 เพื่อลดระยะเวลาที่ไม่สามารถใช้งานได้ของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ โดยพิจารณาจากข้อมูลการใช้งานโปรแกรมมโนภาพเชิงโต้ตอบในระบบ CU-BEMS ฟังก์ชันเสมือนนี้ตรวจสอบข้อมูลทุก 5 นาที ปรากฏว่าตั้งแต่เวลา 07:55 ถึง 08:15 นาฬิการวมระยะเวลา 20 นาที ไม่มีผู้ใช้งานใดสามารถโบกมือเพื่อให้โปรแกรมสร้างมโนภาพเชิงโต้ตอบเปลี่ยนไปแคนวาสอื่นได้สำเร็จดังรูปที่ 5.38 แม้ว่ามีผู้ใช้งานกระตุ้นให้โปรแกรมสร้างมโนภาพเชิงโต้ตอบเข้าสู่แคนวาสปริยายจำนวน 7 ครั้งดังรูปที่ 5.39 ซึ่งมากกว่าจำนวน 5 ครั้งที่กำหนดไว้ในฟังก์ชันเสมือน ดังนั้นฟังก์ชันเสมือนนี้จึงปิดโปรแกรมสร้างมโนภาพเชิงโต้ตอบ และเริ่มต้นโปรแกรมใหม่ดังแสดงในรูปที่ 5.40 และส่งข้อความแจ้งเตือนผ่านทางโปรแกรมประยุกต์ไลน์ดังรูปที่ 5.41 และอีเมล์ดังรูปที่ 5.42



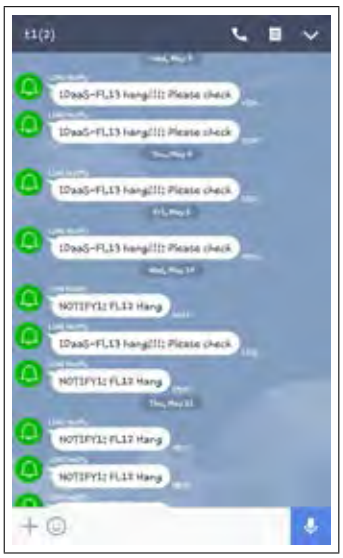
รูปที่ 5.38: จำนวนการโบกมือของผู้ใช้งานผ่านโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟต์ชั้น 13



รูปที่ 5.39: จำนวนผู้ใช้งานที่กระตุ้นให้โปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟต์ชั้น 13 เข้าสู่แคนวาสปริยาย



รูปที่ 5.40: สถานะการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณหน้าลิฟต์ชั้น 13



รูปที่ 5.41: การส่งข้อความแจ้งเตือนผ่านทางโปรแกรมประยุกต์ไลน์

รูปที่ 5.42: การส่งข้อความแจ้งเตือนผ่านทางอีเมล์

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 บทสรุป

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนา และสร้างฟังก์ชันเสมือนสำหรับอินเทอร์เน็ตของสรรพสิ่งในระบบจัดการพลังงานไฟฟ้าภายในอาคารโดยใช้เทคโนโลยีคลาวด์ เพื่อให้สามารถติดตั้งฟังก์ชันได้รวดเร็ว ลดความผิดพลาดที่เกิดจากผู้ติดตั้งระบบ และสามารถใช้ทรัพยากรคอมพิวเตอร์ได้เหมาะสมกับความต้องการของฟังก์ชันนั้น โดยใช้การทำงานร่วมกันระหว่างโปรแกรมโอเพนสแต็ก [16] และโปรแกรมโอเพนบาตง [22] วิทยานิพนธ์ฉบับนี้นำงานวิจัย [19] มาสร้างเป็นฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M รวมถึงการเพิ่ม และการปรับลดทรัพยากรของแพลตฟอร์มโอเพนเอ็มทีซี [14] แบบอัตโนมัติเพื่อให้สอดคล้องกับปริมาณข้อมูลที่เข้าสู่ระบบ รวมถึงนำงานวิจัย [20] มาสร้างเป็นฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite และประยุกต์นำข้อมูลของตัวรับรู้อุณหภูมิในระบบ CU-BEMS ภายใต้มาตรฐาน IEEE1888 มาใช้เพื่อควบคุมการทำงานของเครื่องปรับอากาศภายในห้องคลัสเตอร์ที่อยู่ภายใต้มาตรฐาน ECHONET Lite และนำงานวิจัย [21] มาสร้างเป็นฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 โดยนำข้อมูลอัตราส่วนค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศมานำเสนอผ่านโปรแกรมมโนภาพเชิงโต้ตอบ [12] ฟังก์ชันเสมือนเหล่านี้ถูกสร้างบนแพลตฟอร์มโอเพนสแต็กที่ถูกติดตั้งบนคอมพิวเตอร์ที่มีความเร็วการประมวลผล 3.3 กิกะเฮิรตซ์ จำนวน 4 คอร์ และแรม 32 กิกะไบต์ ผลการสร้างและทดสอบฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M โดยที่แต่ละเครื่องเสมือนมีหน่วยประมวลผลกลาง 1 คอร์ และแรม 2 กิกะไบต์ มีดังนี้

- สามารถสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ขึ้นพร้อมใช้งานภายในระยะเวลาเฉลี่ย 262 วินาที และสามารถยุติการทำงานของฟังก์ชันเสมือนได้ภายในระยะเวลาเฉลี่ย 5 วินาที รวมทั้งสามารถประสานข้อมูลแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ได้ตามงานวิจัย [19]
- สมรรถนะการทำงานของเครื่องเสมือนเอ็นเอสซีแอลบนแพลตฟอร์มโอเพนสแต็กจำนวน 1 เครื่องสามารถบันทึกข้อมูลด้วยความเร็วสูงสุด 5,380 ข้อมูลต่อนาที
- เมื่อนำโปรแกรมโอเพนบาตงมาใช้เพื่อปรับจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติตามระดับของหน่วยประมวลผลกลาง ผลปรากฏว่าสามารถสร้างเครื่องเสมือนเอ็นเอสซีแอลเพิ่มได้ภายในระยะเวลาเฉลี่ย 119 วินาที และสามารถปรับลดเครื่องเสมือนเอ็นเอสซีแอลนี้ได้ภายในระยะเวลาเฉลี่ย 2 วินาที
- เมื่อใช้เครื่องเสมือนเอ็นเอสซีแอลจำนวน 2 เครื่องร่วมกับบริการ LBaaS ส่งผลให้แพลตฟอร์มโอเพนเอ็มทีซีสามารถบันทึกข้อมูลต่อนาทีเพิ่มขึ้น 29.2 เปอร์เซ็นต์
- เมื่อใช้ครื่องเสมือนเอ็นเอสซีแอลจำนวน 3 เครื่องร่วมกับบริการ LBaaS ส่งผลให้แพลตฟอร์มโอเพนเอ็มทีซีสามารถบันทึกข้อมูลต่อนาทีเพิ่มขึ้น 32.9 เปอร์เซ็นต์

- การขยายพื้นที่จัดเก็บข้อมูลของเครื่องเสมือนมองโกดีบี เมื่อพื้นที่จัดเก็บข้อมูลเหลือน้อยกว่าระดับที่กำหนด ผลปรากฏว่าโปรแกรมโอเพนบาตงสามารถสร้างเครื่องเสมือนมองโกดีบีเพิ่มขึ้น 1 เครื่องได้อัตโนมัติ และพร้อมถ่ายโอนข้อมูลด้วยวิธีการขยายฐานข้อมูลของมองโกดีบีภายในระยะเวลา 98 วินาที

การนำบริการ LBaaS มาใช้งานทำให้ข้อมูลถูกส่งเข้าสู่เครื่องเสมือนเอ็นเอสซีแอลแต่ละเครื่องแบบวนรอบ ส่งผลให้เครื่องเสมือนเหล่านี้มีระดับของหน่วยประมวลผลกลางที่ใกล้เคียงกัน ในการทดสอบนี้ โปรแกรมโอเพนบาตงตรวจสอบข้อมูลการใช้งานทรัพยากรของเครื่องเสมือนจากโปรแกรมแซบบิกซ์ ทำให้สามารถตรวจสอบได้เฉพาะข้อมูลการใช้งานทรัพยากรจากระบบปฏิบัติการ ไม่สามารถตรวจสอบการใช้งานทรัพยากรของแต่ละโปรแกรมประยุกต์ได้โดยตรง

นอกจากนี้ โปรแกรมโอเพนบาตงสามารถสร้างฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite ที่สามารถวิเคราะห์ข้อมูลอุณหภูมิในห้องคลัสเตอร์จากระบบ CU-BEMS และควบคุมการทำงานของเครื่องปรับอากาศที่ใช้มาตรฐาน ECHONET Lite ส่งผลให้อุณหภูมิในห้องคลัสเตอร์นี้อยู่ในระดับที่กำหนดไว้ได้ ขณะที่ฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 สามารถนำเสนอข้อมูลอัตราส่วนค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศในแต่ละพื้นที่ของชั้น 13 อาคารเจริญวิศวกรรมผ่านทางโปรแกรมมโนภาพเชิงโต้ตอบ ส่งผลให้ผู้ใช้สามารถรับรู้ถึงการใช้พลังงานไฟฟ้าภายในอาคารได้มากขึ้น นอกจากนี้ได้นำข้อมูลการใช้งานโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณชั้น 13 ในระบบ CU-BEMS มาวิเคราะห์ และแก้ไขปัญหาการทำงานที่ผิดพลาดของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ ซึ่งผู้ใช้งานไม่สามารถเข้าถึงแคนวาสอื่นนอกจากแคนวาสปริยายได้เนื่องจากโปรแกรมไม่ตอบสนองต่อการโบกมือของผู้ใช้งาน รวมถึงแจ้งเตือนผู้ดูแลระบบเมื่อเกิดปัญหา ทำให้ผู้ดูแลระบบสามารถนำข้อมูลนี้ไปใช้เพื่อปรับปรุง และแก้ไขในอนาคตได้ ระยะเวลาการสร้างฟังก์ชันเสมือนแต่ละฟังก์ชันขึ้นกับเครื่องคอมพิวเตอร์ที่นำมาใช้งานและขั้นตอนการติดตั้งโปรแกรมที่กำหนดขึ้น อย่างไรก็ตาม โปรแกรมโอเพนบาตงรุ่น 3.1 ที่นำมาใช้งานสามารถทำงานร่วมกับโปรแกรมโอเพนสแต็กได้เท่านั้น ไม่สามารถทำงานร่วมกับระบบโครงสร้างพื้นฐานแบบแบ่งปันทรัพยากรผ่านโครงข่ายเชิงพาณิชย์อื่นได้

6.2 ข้อเสนอแนะ

การสร้างฟังก์ชันเสมือนด้วยโปรแกรมประยุกต์ Docker [28] ทำให้สามารถสร้างฟังก์ชันเสมือนบนระบบปฏิบัติการต่าง ๆ เช่น ระบบปฏิบัติการวินโดวส์ ระบบปฏิบัติการลินุกซ์ ระบบปฏิบัติการ macOS รวมถึงระบบโครงสร้างพื้นฐานแบบแบ่งปันทรัพยากรผ่านโครงข่าย ดังนั้นการสร้างฟังก์ชันเสมือนสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารด้วย Docker ส่งผลให้ฟังก์ชันเสมือนมีความกะทัดรัด และมีความยืดหยุ่นสำหรับการใช้งานเพิ่มมากขึ้น ทำให้สามารถใช้ทรัพยากรคอมพิวเตอร์ได้อย่างมีประสิทธิภาพมากขึ้น รวมถึงนักวิจัยสามารถสร้างฟังก์ชันเสมือนได้สะดวก โดยแต่ละฟังก์ชันสามารถทำได้อย่างเป็นอิสระต่อกัน ส่งผลให้นักวิจัยสามารถพัฒนาฟังก์ชันการทำงานต่าง ๆ ขึ้นมาโดยใช้ทรัพยากรคอมพิวเตอร์ร่วมกันได้

รายการอ้างอิง

[1] Sahoo, J., Mohapatra, S., and Lath, R. Virtualization: a survey on concepts, taxonomy and associated security issues. 2010 Second International Conference on Computer and Network Technology (ICCNT), pp.222-226, Bangkok, July 1-4, 2010.

[2] Smith, J. E., and Nair R. The architecture of virtual machines. Computer, vol. 38, no. 5, (May 2005): 32-38.

[3] Duffield, N,G., Goyal, P., Greenberg, A., Mishra, P., Ramakrishnan K. K., and Van der Merive, J. E. A flexible model for resource management in virtual private networks. ACM SIGCOMM Computer Communication Review, Vol. 29, No. 4, (1999) : 95-108.

[4] Jain, R., and Paul, S. Network virtualization and software defined networking for cloud computing: a survey. IEEE Communications Magazine, vol. 51, no. 11, (November, 2013) : 24-31.

[5] Mell, P. and Grance, T. The NIST definition of cloud computing. National Institute of Standards and Technology, SP800-145, (September, 2011).

[6] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. Internet of Things (IoT): a vision, architectural elements, and future directions. Future Generation Computer Systems, vol. 29 (2013): 1645-1660.

[7] Evans, D. The Internet of Things : How the next evolution of the internet is changing everything. Cisco Internet Business Solutions Group white paper, April, 2011.

[8] รายงานฉบับสิ้นสุดโครงการ, โครงการวิจัยและพัฒนาเทคโนโลยีระบบโครงข่ายไฟฟ้าอัจฉริยะเพื่อบริหารจัดการการใช้พลังงานไฟฟ้าของอาคาร, จุฬาลงกรณ์มหาวิทยาลัย (สถาบันวิจัยพลังงาน) , เสนอกองทุนเพื่อส่งเสริมการอนุรักษ์ พลังงานสำนักงานนโยบายและแผนพลังงาน, กันยายน 2557.

[9] IEEE1888-2011. Standard for Ubiquitous Green Community Control Network, 2011.

[10] Le, D. H., and Pora, W. Development of smart meter for building energy management system based on the IEEE 1888 standard with Wi-Fi communication. 2014 International Conference on Electronics, Information and Communications (ICEIC), pp.1-2, Kota Kinabalu, January 15-18, 2014.

[11] Khawsa-ard, P., and Aswakul, C. Application of simple computer board game with gesture sensor input for increasing awareness in electrical energy consumption. The 29th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2014), Phuket, July, 2014.

[12] Khawsa-ard, P., and Aswakul, C. IEEE1888 interactive display as a service (IDaaS): example in building energy management system. The 39th Annual Computer Software and Applications Conference, pp.517-522, Taichung, July 1-5, 2015.

[13] ETSI Technical Committee Machine-to-Machine communications (M2M). ETSI TS 102 690 V1.2.1 (2013-06). Machine-to-Machine Communications (M2M): Functional Architecture [Online]. 2013. Available from : http://www.etsi.org/ [2017,March]

[14] Corici, M. et al. OpenMTC: prototyping machine type communication in carrier grade operator networks. 2012 IEEE Globecom Workshops, pp. 1735-1740, Anaheim, December 3-7, 2012.

[15] Corici, A.A., Elmangoush, A., Magedanz, T., Steinke, R., Mwangama, J., and Ventura, N. An OpenMTC platform-based interconnected European–South African M2M Testbed for Smart City Services. The first International Conference on the use of Mobile Informations and Communication Technology, pp. 35-39, 2014.

[16] Sefraoui, O., Aissaoui, M., and Eleuldj, M. OpenStack: toward an open-source solution for cloud computing. International Journal of Computer Applications, vol. 55, no. 3, (October 2012).

[17] UNIFI [Online]. Avialable from : http://www.daadunifi.org/ [2017, February].

[18] Klinpratum, T., Saivichit, C., Elmangoush, A., and Magedanz, T. Toward interconnecting M2M/IoT standards: interworking proxy for IEEE1888 standard at ETSI M2M platform. The 29th International Technical Conference on Circuit/Systems Computers and Communications (ITC-CSCC 2014), pp. 763-766, Phuket, July 1-4, 2014.

[19] Kosolworrawattanakul, N., Elmangoush, A., Magedanz, T., and Aswakul, C. Development of real-time data synchronization for IEEE1888 and ETSI M2M standards. Workshop on Internet Architecture and Applications (IA 2014), pp.79-84. 2014.

[20] ชญานนท์ แสงอำไพ และเชาวน์ดิศ อัศวกุล. การพัฒนาการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite สำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคาร. การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 39, เพชรบุรี, 2-4 พฤศจิกายน 2559.

[21] เขตต์นนท์ ชูพุทธิพงศ์. และเชาวน์ดิศ อัศวกุล. การพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบการจัดการพลังงานภายในอาคารด้วยการวิเคราะห์พลังงานสูญเปล่าโดยใช้ตัวรับรู้การเคลื่อนไหว. การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 39, เพชรบุรี, 2-4 พฤศจิกายน 2559.

[22] Carella, G. A., and Magedanz, T. Open Baton: A Framework for virtual network function management and orchestration for emerging software-based 5G networks. IEEE Softwarization eNewsletter, July 2016.

[23] Tader, Paul. Server monitoring with Zabbix. Linux Journal, vol. 195, no. 7, (July 2010).

[24] Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG). ETSI GS NFV-MAN 001 v1.1.1 (2014-12): Network Functions Virtualisation (NFV); Management and Orchestration, [Online]. 2014. Available from : http://www.etsi.org/ [2017,March]

[25] Corici, M., Reichel, B., Bochow, B., and Magedanz, T. An SDN-based solution for increasing flexibility and reliability of dedicated network environments, The 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1-6, Berlin, September 6-9, 2016.

[26] Chodorow, K. Scaling MongoDB: Sharding, Cluster Setup, and Administration. O'Reilly Media, Inc., 2011.

[27] Wantamanee, A., Watarakitpaisarn, S., Carella, G., Aswakul, C., and Magedanz, T. Virtualising machine to machine (M2M) application using Open Baton as NFV-compliant framework for building energy management system. The 11th International Conference on Computer Science & Education, pp. 199-204, Nagoya, August 23-25, 2016.

[28] Merkel, D. Docker: lightweight linux containers for consistent development and deployment. Linux Journal, vol. 239, no. 2, (March 2014).

ภาคผนวก

ภาคผนวก ก
ชุดคำสั่งสร้างทรัพยากรเสมือนขึ้นบนโปรแกรมโอเพนสแต็ก

init_openstack.sh

ไฟล์ init_openstack.sh สำหรับการสร้างทรัพยากรบนแพลตฟอร์มโอเพนสแต็กมีรายละเอียดการทำงานของโปรแกรมมีดังนี้

- บรรทัดที่ 2: การประกาศตัวแปรสภาวะแวดล้อมของโปรแกรมโอเพนสแต็ก
- บรรทัดที่ 3-5: การบรรจุไฟล์ระบบปฏิบัติการ Ubuntu cloud 16.04 ที่ถูกปรับแต่งเข้าสู่บริการ Glance ของโปรแกรมโอเพนสแต็ก
- บรรทัดที่ 6-8: การสร้างโครงข่ายเสมือนชื่อ private และการกำหนดช่วงของหมายเลขไอพีให้กับเครื่องเสมือน
- บรรทัดที่ 9-11: การสร้างอุปกรณ์จัดเส้นทางเสมือนเพื่อเชื่อมต่อระหว่างโครงข่ายเสมือน private กับโครงข่ายจริง
- บรรทัดที่ 12-13: การสร้างกุญแจ (key pair) เพื่อให้ผู้ใช้งานสามารถเข้าถึงเครื่องเสมือนด้วยโพรโทคอล secure shell
- บรรทัดที่ 14-27: การกำหนดข้อกำหนดความปลอดภัยให้กับเครื่องเสมือน
- บรรทัดที่ 28-32: การสร้างหมายเลข floating ip ให้กับเครื่องเสมือน ทำให้เครื่องเสมือนมีหมายเลขไอพีที่อยู่ในโครงข่ายจริง
- บรรทัดที่ 33-50: การสร้างบริการ LBaaS โดยที่กำหนดพอร์ตใช้งานหมายเลข 15000 สำหรับโพรโทคอลทีซีพีและใช้วิธีการวบรอบ โดยบรรทัดที่ 38-50 เป็นการตรวจสอบสถานะของบริการ LBaaS

```
1  #!/bin/bash
2  . ${HOME}/admin-rc
3  openstack image create "openmtcv8" --file ./openmtcv82.qcow2 \
4  --disk-format qcow2 --container-format bare --public
5  openstack image list
6  openstack network create private
7  openstack subnet create --network private --subnet-range \
8  10.0.12.0/24 --gateway 10.0.12.1  private_subnet
9  openstack router create router1
10 neutron router-gateway-set router1 public
11 neutron router-interface-add router1 private_subnet
12 openstack keypair create --public-key ${HOME}/id_rsa12.pub \
13 openstack_key12
14 openstack keypair list
```

```
15 openstack security group create openstack_sec12
16 openstack security group rule create --protocol icmp openstack_sec12
17 openstack security group rule create openstack_sec12 \
18 --protocol tcp --dst-port 22:22 --remote-ip 0.0.0.0/0
19 openstack security group rule create openstack_sec12 \
20 --protocol tcp --dst-port 80:80 --remote-ip 0.0.0.0/0
21 openstack security group rule create openstack_sec12 \
22 --protocol tcp --dst-port 1000:1000 --remote-ip 0.0.0.0/0
23 openstack security group rule create openstack_sec12 \
24 --protocol tcp --dst-port 8081:8081 --remote-ip 0.0.0.0/0
25 openstack security group rule create openstack_sec12 \
26 --protocol tcp --dst-port 15000:15000 --remote-ip 0.0.0.0/0
27 openstack security group rule create openstack_sec12 \
28 --protocol tcp --dst-port 10050:10050 --remote-ip 0.0.0.0/0
29 for i in {1..15}
30 do
31   openstack floating ip create public
32 done
33 openstack floating ip list
34 neutron lbaas-loadbalancer-create --name loadbalancer1 \
35 private_subnet
36 STATUS1=$(neutron lbaas-loadbalancer-show loadbalancer1 \
37 | awk '$2 == "provisioning_status" {print $4}')
38 sleep 30
39 COUNTER=0
40 while [  $COUNTER -lt 10 ];
41 do
42   if [ $STATUS1 = "ACTIVE" ]; then
43     neutron lbaas-listener-create --loadbalancer loadbalancer1 \
44     --protocol HTTP --protocol-port 15000 --name listener1
45     neutron lbaas-pool-create --lb-algorithm ROUND_ROBIN \
46     --listener listener1 --protocol TCP --name pool1
47     break
48   else
49     sleep 30
50     COUNTER=$(($COUNTER+1))
51   fi
52 done
```

ภาคผนวก ข
ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่าง มาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

ข1. nsd_etsi_ieee1888.json

ไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2Mมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 2-4: การระบุชื่อไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M รวมถึงชื่อผู้สร้าง และรุ่นของฟังก์ชันเสมือน
- บรรทัดที่ 5-17: การกำหนดชื่อเครื่องเสมือนเอ็นเอสซีแอล ประเภทของเครื่องเสมือน และการกำหนดพารามิเตอร์ให้กับเครื่องเสมือน
- บรรทัดที่ 18-38: การกำหนดเงื่อนไขการสร้างเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติ โดยบรรทัดที่ 22 กำหนดระยะเวลาการตรวจสอบพารามิเตอร์ที่กำหนดเป็นระยะเวลา 60 วินาที บรรทัดที่ 23 กำหนดระยะเวลาการยกเว้นการตรวจสอบพารามิเตอร์ที่กำหนดหลังจากการสร้างเครื่องเสมือนสำเร็จแล้วเป็นระยะเวลา 120 วินาที บรรทัดที่ 26-32 ระบุเงื่อนไขให้แจ้งเตือนเมื่อค่าเฉลี่ยของระดับหน่วยประมวลผลกลางที่ไม่ได้ใช้งานต่ำกว่า 70 เปอร์เซ็นต์ และบรรทัดที่ 33-36 กำหนดให้สามารถสร้างเครื่องเสมือนเอ็นเอสซีแอลขึ้นครั้งละ 1 เครื่องเมื่อเป็นไปตามเงื่อนไขที่กำหนด
- บรรทัดที่ 39-59: การกำหนดเงื่อนไขการทำลายเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติ โดยบรรทัดที่ 43 กำหนดระยะเวลา 180 วินาทีสำหรับการตรวจสอบพารามิเตอร์ที่กำหนด บรรทัดที่ 44 กำหนดระยะเวลาการยกเว้นการตรวจสอบพารามิเตอร์หลังจากทำลายเครื่องเสมือนแล้วเป็นระยะเวลา 60 วินาที บรรทัดที่ 47-53 ระบุเงื่อนไขการแจ้งเตือนเมื่อค่าเฉลี่ยของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานมีระดับสูงกว่า 85 เปอร์เซ็นต์ บรรทัดที่ 54-59 กำหนดให้ทำลายเครื่องเสมือนเอ็นเอสซีแอลครั้งละ 1 เครื่องเมื่อเป็นไปตามเงื่อนไขที่กำหนด
- บรรทัดที่ 61: การกำหนดชื่อไฟล์ระบบปฏิบัติการที่มีอยู่บนแพลตฟอร์มโอเพนสแต็กให้ติดตั้งในเครื่องเสมือนเอ็นเอสซีแอล
- บรรทัดที่ 62: การระบุโนดโอเพนสแต็กที่ลงทะเบียนไว้ในโปรแกรมโอเพนบาตง เพื่อใช้สำหรับสร้างเครื่องเสมือนเอ็นเอสซีแอล
- บรรทัดที่ 63: การระบุจำนวนเครื่องเสมือนเอ็นเอสซีแอลสูงสุดที่สามารถสร้างขึ้นได้
- บรรทัดที่ 64-70: การระบุหมายเลขไอพีของโครงข่ายเสมือนให้กับเครื่องเสมือนเอ็นเอสซีแอล
- บรรทัดที่ 71: การระบุชื่อโครงข่ายเสมือนให้กับเครื่องเสมือนเอ็นเอสซีแอล

- บรรทัดที่ 72-76: การระบุชื่อชุดคำสั่งที่ใช้งานสำหรับเหตุการณ์ INSTANTIATE เพื่อติดตั้งโปรแกรมที่จำเป็นสำหรับการสร้างฟังก์ชันเสมือนนี้และปรับแต่งค่าพารามิเตอร์เช่น หมายเลขไอพี ชื่อของเครื่องเสมือนภายในเครื่องของตัวเอง รวมทั้งการประสานเวลาจาก ntp.chula.ac.th
- บรรทัดที่ 77-82: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ CONFIGURE เพื่อแลกเปลี่ยนและบันทึกค่าพารามิเตอร์ของแต่ละเครื่องที่อยู่ในฟังก์ชันเสมือนนี้ได้แก่ หมายเลขไอพี และชื่อของเครื่องเสมือนที่อยู่ในฟังก์ชันเสมือนนี้
- บรรทัดที่ 83-87: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ START เพื่อลงทะเบียนหมายเลขไอพีของเครื่องเสมือนเอ็นเอสซีแอลในบริการ LBaaS ของแพลตฟอร์มโอเพนสแต็กและเริ่มต้นการทำงานของฟังก์ชันเสมือนนี้
- บรรทัดที่ 88-92: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ TERMINATE เพื่อยุติการทำงานของฟังก์ชันเสมือน โดยหมายเลขไอพีของเครื่องเสมือนเอ็นเอสซีแอลจะถูกลบออกจากบริการ LBaaS และเครื่องเสมือนเอ็นเอสซีแอลจะถูกทำลาย
- บรรทัดที่ 93-98: การระบุชื่อโปรแกรมสำหรับเหตุการณ์ SCALE_IN เพื่อทำลายเครื่องเสมือนเอ็นเอสซีแอลในกรณีที่ทำลายจำนวนเครื่องเสมือนเอ็นเอสซีแอลแบบอัตโนมัติตามเงื่อนไขที่กำหนด
- บรรทัดที่ 99: การกำหนดรายละเอียดของเครื่องเสมือนเอ็นเอสซีแอลซึ่งประกอบด้วยจำนวนหน่วยประมวลผลเสมือน แรมและพื้นที่เก็บข้อมูลของเครื่องเสมือน
- บรรทัดที่ 100: ยูอาร์แอลระบุตำแหน่งที่เก็บไฟล์ข้อมูลสำหรับการสร้างฟังก์ชันเสมือนนี้
- บรรทัดที่ 102-114: การกำหนดชื่อเครื่องเสมือนเอ็นไอพี ประเภทของเครื่องเสมือนและการกำหนดพารามิเตอร์ให้กับเครื่องเสมือน
- บรรทัดที่ 116: การกำหนดชื่อไฟล์ระบบปฏิบัติการที่มีอยู่บนแพลตฟอร์มโอเพนสแต็ก เพื่อให้ติดตั้งลงในเครื่องเสมือนเอ็นไอพี
- บรรทัดที่ 117: การระบุโนดโอเพนสแต็กที่ลงทะเบียนไว้ในโปรแกรมโอเพนบาตง เพื่อใช้สร้างเครื่องเสมือนเอ็นไอพี
- บรรทัดที่ 118: การระบุจำนวนเครื่องเสมือนเอ็นไอพีสูงสุดที่สามารถสร้างขึ้นได้
- บรรทัดที่ 119-125: การระบุหมายเลขไอพีของโครงข่ายเสมือนให้กับเครื่องเสมือนเอ็นไอพี
- บรรทัดที่ 126: การระบุชื่อโครงข่ายเสมือนให้กับเครื่องเสมือนเอ็นไอพี
- บรรทัดที่ 127-131: การระบุชื่อชุดคำสั่งที่ใช้งานสำหรับเหตุการณ์ INSTANTIATE เพื่อติดตั้งโปรแกรมที่จำเป็นสำหรับการสร้างฟังก์ชันเสมือนนี้และปรับแต่งค่าพารามิเตอร์เช่น หมายเลขไอพี ชื่อของเครื่องเสมือนภายในเครื่องของตัวเอง รวมทั้งการประสานเวลาจาก ntp.chula.ac.th

- บรรทัดที่ 132-137: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ CONFIGURE เพื่อแลกเปลี่ยนและบันทึกค่าพารามิเตอร์ของแต่ละเครื่องที่อยู่ในฟังก์ชันเสมือนนี้ได้แก่ หมายเลขไอพี และชื่อของเครื่องเสมือนที่อยู่ในฟังก์ชันเสมือนนี้
- บรรทัดที่ 138-143: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ START เพื่อเริ่มการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M
- บรรทัดที่ 144: การกำหนดรายละเอียดของเครื่องเสมือนเอ็นไอพีซึ่งประกอบด้วยจำนวนหน่วยประมวลผลเสมือน แรมและพื้นที่เก็บข้อมูลของเครื่องเสมือน
- บรรทัดที่ 145: ยูอาร์แอลระบุตำแหน่งที่เก็บไฟล์ข้อมูลสำหรับการสร้างฟังก์ชันเสมือนนี้
- บรรทัดที่ 146-159: การกำหนดชื่อเครื่องเสมือนมองโกดีบี ประเภทของเครื่องเสมือนและการกำหนดพารามิเตอร์ให้กับเครื่องเสมือน
- บรรทัดที่ 160-178: การกำหนดเงื่อนไขการสร้างเครื่องเสมือนมองโกดีบีเพิ่มแบบอัตโนมัติ โดยบรรทัดที่ 164 กำหนดระยะเวลาการตรวจสอบพารามิเตอร์ที่กำหนดเป็นระยะเวลา 90 วินาที บรรทัดที่ 165 กำหนดระยะเวลาการยกเว้นการตรวจสอบพารามิเตอร์ที่กำหนดในกรณีที่มีการสร้างเครื่องเสมือนขึ้นแล้วเป็นระยะเวลา 300 วินาที บรรทัดที่ 167-173 ระบุเงื่อนไขให้แจ้งเตือนเมื่อค่าเฉลี่ยของพื้นที่เก็บข้อมูลของเครื่องเสมือนมองโกดีบีน้อยกว่า 30 เปอร์เซ็นต์ บรรทัดที่ 174-176 กำหนดให้สร้างเครื่องเสมือนมองโกดีบีขึ้นครั้งละ 1 เครื่องเมื่อเป็นไปตามเงื่อนไขที่กำหนด
- บรรทัดที่ 180: การกำหนดชื่อไฟล์ระบบปฏิบัติการที่มีอยู่บนแพลตฟอร์มโอเพนสแต็ก ให้ติดตั้งในเครื่องเสมือนมองโกดีบี
- บรรทัดที่ 181: การระบุโนดโอเพนสแต็กที่ลงทะเบียนไว้ในโปรแกรมโอเพนบาตง เพื่อใช้สร้างเครื่องเสมือนมองโกดีบี
- บรรทัดที่ 182: การระบุจำนวนเครื่องเสมือนมองโกดีบีสูงสุดที่สามารถสร้างขึ้นได้
- บรรทัดที่ 183-189: การระบุหมายเลขไอพีของโครงข่ายเสมือนให้กับเครื่องเสมือนมองโกดีบี
- บรรทัดที่ 190: การระบุชื่อโครงข่ายเสมือนให้กับเครื่องเสมือนมองโกดีบี
- บรรทัดที่ 191-195: การระบุชื่อชุดคำสั่งที่ใช้งานสำหรับเหตุการณ์ INSTANTIATE เพื่อติดตั้งโปรแกรมฐานข้อมูลมองโกดีบีและปรับแต่งค่าพารามิเตอร์เช่น หมายเลขไอพี ชื่อของเครื่องเสมือนภายในเครื่องของตัวเอง รวมทั้งการประสานเวลาจาก ntp.chula.ac.th
- บรรทัดที่ 196-201: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ CONFIGURE เพื่อแลกเปลี่ยนและบันทึกค่าพารามิเตอร์ของแต่ละเครื่องที่อยู่ในฟังก์ชันเสมือนนี้ได้แก่ หมายเลขไอพี และชื่อของเครื่องเสมือนที่อยู่ในฟังก์ชันเสมือนนี้ และปรับแต่งพารามิเตอร์ให้พร้อมสำหรับการขยายฐานข้อมูลมองโกดีบี
- บรรทัดที่ 202-207: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ START เพื่อเริ่มการทำงานของฐานข้อมูลมองโกดีบี

- บรรทัดที่ 208: การกำหนดรายละเอียดของเครื่องเสมือนเอ็นไอพีซึ่งประกอบด้วยจำนวนหน่วยประมวลผลเสมือน แรมและพื้นที่เก็บข้อมูลของเครื่องเสมือน
- บรรทัดที่ 209: ยูอาร์แอลระบุตำแหน่งที่เก็บไฟล์ข้อมูลสำหรับการสร้างฟังก์ชันเสมือนนี้
- บรรทัดที่ 212: การระบุโครงข่ายทอพอโลยีที่ใช้สำหรับการสร้างฟังก์ชันเสมือนนี้
- บรรทัดที่ 213-267: การกำหนดการแลกเปลี่ยนพารามิเตอร์ของเครื่องเสมือนในขั้นตอน CONFIGURE ได้แก่ หมายเลขไอพีและชื่อของเครื่องเสมือนแต่ละเครื่อง

```
1  {
2     "name":"openmtc-scale14",
3     "vendor":"NRG-team",
4     "version":"0.4",
5     "vnfd":[{
6        "vendor":"NRG-team",
7        "version":"0.1",
8        "name":"nscl",
9        "type":"server",
10       "endpoint":"generic",
11       "configurations":{
12         "name":"config_name",
13         "configurationParameters":[{
14                 "confKey":"publicip",
15                 "value":"161.200.90.78"
16         }]
17     },
18     "auto_scale_policy": [{
19       "name":"scale-out-nscl",
20       "threshold":100,
21       "comparisonOperator":">=",
22       "period":60,
23       "cooldown":120,
24       "mode":"REACTIVE",
25       "type":"WEIGHTED",
26       "alarms": [{
27         "metric":"system.cpu.util[,idle]",
28         "statistic":"avg",
29         "comparisonOperator":"<=",
30         "threshold":70,
31         "weight":1
32       }],
33       "actions": [{
34         "type":"SCALE_OUT",
35         "value":"1",
36         "target":"server"
37         }]
```

```
38        },
39        {
40        "name":"scale-in-nscl",
41        "theshold":100,
42        "comparisonOperator":">=",
43        "period":180,
44        "cooldown":60,
45        "mode":"REACTIVE",
46        "type":"WEIGHTED",
47        "alarms": [{
48          "metric":"system.cpu.util[,idle]",
49          "statistic":"avg",
50          "comparisonOperator":">=",
51          "threshold":85,
52          "weight":1
53        }],
54        "actions": [{
55          "type":"SCALE_IN",
56          "value":"1",
57          "target":"server"
58        }]
59      }],
60      "vdu":[{
61        "vm_image":["openmtcv8"],
62        "vimInstanceName":["vim12"],
63        "scale_in_out":3,
64        "vnfc":[{
65          "connection_point":[{
66            "floatingIp":random,
67            "virtual_link_reference":"private1"
68            }]
69        }]
70      }],
71      "virtual_link":[{"name":"private1"}],
72      "lifecycle_event":[{
73        "event":"INSTANTIATE",
74        "lifecycle_events":[
75          "install_nscl.sh"
76        ]},
77        {
78        "event":"CONFIGURE",
79        "lifecycle_events":[
80          "client_configns cl.sh",
81          "db_config.sh"
82        ]},
83        {
```

```
      "event":"START",
      "lifecycle_events":[
        "start_nscl.sh"
      ]},
      {
      "event":"TERMINATE",
      "lifecycle_events":[
        "terminate_lbaas.sh"
      ]},
      {
      "event":"SCALE_IN",
      "lifecycle_events":[
        "scale_in_lbaas.sh"
      ]}
    ],
    "deployment_flavour":[{"flavour_key":"m1.small"}],
    "vnfPackageLocation":"https://github.com/chart2023/NIP2.git"
  },
  {
    "vendor":"NRG-team",
    "version":"0.1",
    "name":"nip",
    "type":"client",
    "endpoint":"generic",
    "configurations":{
      "name":"config_name",
      "configurationParameters":[{
        "confKey":"publicip",
        "value":"161.200.90.78"
        }]
    },
    "vdu":[{
      "vm_image":["openmtcv8"],
      "vimInstanceName":["vim12"],
      "scale_in_out":1,
      "vnfc":[{
        "connection_point":[{
          "floatingIp":"192.168.9.120",
          "virtual_link_reference":"private1"
          }]
        }]
    }],
    "virtual_link":[{"name":"private1"}],
    "lifecycle_event":[{
      "event":"INSTANTIATE",
      "lifecycle_events":[
```

```
        "install_nip.sh"
      ]},
      {
      "event":"CONFIGURE",
      "lifecycle_events":[
        "server_configNip.sh",
        "db_configNip.sh"
      ]},
      {
      "event":"START",
      "lifecycle_events":[
        "start_nip.sh"
      ]}
    ],
    "deployment_flavour":[{"flavour_key":"m1.small"}],
    "vnfPackageLocation":"https://github.com/chart2023/NIP2.git"
  },
  {
    "vendor":"NRG-team",
    "version":"0.1",
    "name":"mongodb",
    "type":"db",
    "endpoint":"generic",
    "configurations":{
      "name":"config_db",
      "configurationParameters":[{
        "confKey":"publicip",
        "value":"161.200.90.78"
      }]
    },
    "auto_scale_policy": [{
      "name":"scale-out-db",
      "threshold":100,
      "comparisonOperator":">=",
      "period":90,
      "cooldown":300,
      "mode":"REACTIVE",
      "type":"WEIGHTED","alarms": [{
        "metric":"vfs.fs.size[/,pfree]",
        "statistic":"avg",
        "comparisonOperator":"<=",
        "threshold":30,
        "weight":1
      }],
      "actions": [{
        "type":"SCALE_OUT",
```

```
        "value":"1"
      }]
    }],
    "vdu":[{
      "vm_image":["openmtcv8"],
      "vimInstanceName":["vim12"],
      "scale_in_out":2,
      "vnfc":[{
        "connection_point":[{
          "floatingIp":random,
          "virtual_link_reference":"private1"
          }]
        }]
    }],
    "virtual_link":[{"name":"private1"}],
    "lifecycle_event":[{
      "event":"INSTANTIATE",
      "lifecycle_events":[
        "install_db.sh"
      ]},
      {
      "event":"CONFIGURE",
      "lifecycle_events":[
        "server_config.sh",
        "client_configdb.sh"
      ]},
      {
      "event":"START",
      "lifecycle_events":[
        "start_mongodb.sh"
      ]}
    ],
    "deployment_flavour":[{"flavour_key":"m1.medium"}],
    "vnfPackageLocation":"https://github.com/chart2023/NIP2.git"
  }],
  "vnffgd":[],
  "vld":[{"name":"private1"}],
  "vnf_dependency":[{
    "source":{"name":"nscl"},
    "target":{"name":"nip"},
    "parameters":[
      "private1",
      "private1_floatingIp",
      "hostname"
    ]},
    {
```

```
222       "source":{"name":"nip"},
223       "target":{"name":"nscl"},
224       "parameters":[
225         "private1",
226         "private1_floatingIp",
227         "hostname"
228       ]},
229       {
230       "source":{"name":"nscl"},
231       "target":{"name":"mongodb"},
232       "parameters":[
233         "private1",
234         "private1_floatingIp",
235         "hostname"
236       ]},
237       {
238       "source":{"name":"mongodb"},
239       "target":{"name":"nscl"},
240       "parameters":[
241         "private1",
242         "private1_floatingIp",
243         "hostname"
244       ]},
245       {
246       "source":{"name":"nip"},
247       "target":{"name":"mongodb"},
248       "parameters":[
249         "private1",
250         "private1_floatingIp",
251         "hostname"
252       ]},
253       {
254       "source":{"name":"mongodb"},
255       "target":{"name":"nip"},
256       "parameters":[
257         "private1",
258         "private1_floatingIp",
259         "hostname"
260       ]},
261       {
262       "source":{"name":"nscl"},
263       "target":{"name":"nscl"},
264       "parameters":["hostname"]
265       }
266     ]
267   }
```

ข2. install_nip.sh

ไฟล์ install_nip.sh สำหรับเหตุการณ์ INSTANTIATE ของเครื่องเสมือนเอ็นไอพีมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 5-8: การตั้งค่าและประสานเวลาของเครื่องเอ็นไอพีกับ ntp.chula.ac.th เพื่อให้เครื่องเสมือนเอ็นไอพีมีเวลาตรงกับเวลาของระบบ CU-BEMS
- บรรทัดที่ 9-12: การบันทึกค่าหมายเลขไอพีส่วนบุคคลของเครื่องเสมือนเอ็นไอพี เพื่อให้ชุดคำสั่งต่าง ๆ สามารถเรียกใช้ได้
- บรรทัดที่ 13-18: การตรวจสอบการเชื่อมต่ออินเทอร์เน็ตของเครื่องเสมือนเอ็นไอพี
- บรรทัดที่ 19-21: การกำหนดให้เครื่องเสมือนเอ็นไอพีทำงานร่วมกันแบบทันทีระหว่างหน่วยเก็บข้อมูลของระบบ CU-BEMS และแพลตฟอร์มโอเพนเอ็มทีซีทุกครั้งที่เปิดเครื่อง
- บรรทัดที่ 22-28: การนำชุดคำสั่งสำหรับเหตุการณ์ต่าง ๆ ของเครื่องเสมือนเอ็นไอพีไปไว้ในตำแหน่งที่กำหนดไว้

```
1  #!/bin/bash
2  echo "STEP1: INSTANTIATE NIP"
3  echo "START at:" $(date)
4  myhome=${HOME}
5  dpkg-reconfigure -f noninteractive tzdata
6  apt-get update
7  apt-get install --reinstall tzdata -y
8  ntpq -p
9  echo ${private1} | sudo tee ./ipaddress.txt
10 ipfile="./ipaddress.txt"
11 ipaddress=$(head -1 $ipfile)
12 bash -c "echo $ipaddress `cat /etc/hostname` >> /etc/hosts"
13 wget -q --tries=10 --timeout=20 --spider  http://archive.ubuntu.com
14 if [[ $? -eq 0 ]]; then
15         echo "Server can connect to Internet"
16 else
17         echo "Server cannot connect to Internet"
18 fi
19 cp /opt/openbaton/scripts/start-nip.sh /etc/init.d/start-nip.sh
20 chmod ugo+x /etc/init.d/start-nip.sh
21 update-rc.d start-nip.sh defaults
22 sudo rm -rf /OpenMTC-Chula/openmtc/settings/ipserv.js
23 echo "exports.ipopenstack='${publicip}';" | tee --append /OpenMTC-
      Chula/openmtc/settings/ipserv.js
24 cp /opt/openbaton/scripts/openstack.info /home/ubuntu/openstack.info
25 cp /opt/openbaton/scripts/add-iplbaas.sh /home/ubuntu/add-iplbaas.sh
26 cp /opt/openbaton/scripts/del_iplbaas_scale.sh /home/ubuntu/
      del_iplbaas_scale.sh
```

```
27 chmod 755 /home/ubuntu/add-iplbaas.sh /home/ubuntu/del_iplbaas_scale
       .sh
28 echo "STOP at:" $(date)
```

ข3. server_configip.sh
ไฟล์ server_configip.sh สำหรับเหตุการณ์ CONFIGURE ของเครื่องเสมือนเอ็นไอพีมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-12: การบันทึกหมายเลขไอพีและชื่อของเครื่องเสมือนเอ็นเอสซีแอลลงสู่เครื่องเสมือนเอ็นไอพี

```
1  #!/bin/bash
2  echo "START server configure at:" $(date)
3  ipfile="./ipaddress.txt"
4  ipaddress=$(head -1 $ipfile)
5  echo "nscl_ip=$server_private" | tee --append ${HOME}/nscl_info.conf
6  echo "nscl_fip=$server_private_floatingIp" | tee --append ${HOME}/
       nscl_info.conf
7  echo "nscl_hostname=$server_hostname" | tee --append ${HOME}/
       nscl_info.conf
8  echo "exports.ipnscl='$server_private';" | tee --append /OpenMTC-
       Chula/openmtc/settings/ipserv.js
9  echo "exports.ipnip='$ipaddress';" | tee --append /OpenMTC-Chula/
       openmtc/settings/ipserv.js
10 echo "exports.fipnscl='$server_private_floatingIp';" | tee --append
       /OpenMTC-Chula/openmtc/settings/ipserv.js
11 echo "exports.ipopenstack='$publicip';" | tee --append /OpenMTC-
       Chula/openmtc/settings/ipserv.js
12 echo "STOP at:" $(date)
```

ข4. db_configip.sh
ไฟล์ db_configip.sh สำหรับเหตุการณ์ CONFIGURE ของเครื่องเสมือนเอ็นไอพีมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-7: การบันทึกหมายเลขไอพีและชื่อของเครื่องเสมือนมองโกดีบีลงสู่เครื่องเสมือนเอ็นไอพี
- บรรทัดที่ 8-18: การตรวจสอบหมายเลขไอพีและชื่อของเครื่องเสมือนมองโกดีบีเครื่องที่หนึ่งและเครื่องเสมือนมองโกดีบีที่กำลังส่งข้อมูลมายังเครื่องเสมือนเอ็นไอพีนี้
- บรรทัดที่ 19-34: ในกรณีที่เครื่องเสมือนมองโกดีบีที่ส่งข้อมูลมานี้เป็นเครื่องที่หนึ่ง เครื่องเสมือนเอ็นไอพีจะส่งชุดคำสั่งไปติดตั้งระบบการขยายฐานข้อมูลให้กับฐานข้อมูลมองโกดีบี แต่ในกรณีที่เครื่องเสมือนมองโกดีบีที่ส่งข้อมูลมานี้เป็นเครื่องที่สองเป็นต้นไป เครื่องเสมือนเอ็นไอพีจะส่งชุดคำสั่งให้ระบบฐานข้อมูลมองโกดีบีเพิ่มหมายเลขและชื่อของเครื่องเสมือนมองโกดีบีเครื่องนี้เข้าสู่ระบบการขยายฐานข้อมูลของฐานข้อมูลมองโกดีบีที่มีอยู่แล้ว

```
1  #!/bin/bash
2  echo "db configure at NIP"
3  echo "START at:" $(date)
4  DBFILE=/home/ubuntu/db_info.conf
5  echo "$db_private $db_hostname" | sudo tee --append $DBFILE
6  echo "exports.ipdb='$db_private';" | tee --append /OpenMTC-Chula/
       openmtc/settings/ipserv.js
7  echo "$db_private" | sudo tee --append /home/ubuntu/maindb_info.conf
8  NUM_DB=$(cat $DBFILE | wc -l)
9  echo $NUM_DB
10 MAINDB_IP=$(awk -F' ' '{ print $1}' $DBFILE | head -1)
11 echo $MAINDB_IP
12 MAINDB_NAME=$(awk -F' ' '{ print $2}' $DBFILE | head -1)
13 echo $MAINDB_NAME
14 MAINDB_INFO=$(head -1 $DBFILE)
15 echo $MAINDB_INFO
16 THISDB_INFO=$(tail -1 $DBFILE)
17 echo $THISDB_INFO
18 THISDB_NAME=$(awk -F' ' '{ print $2}' $DBFILE | tail -1)
19 if [ $MAINDB_IP = $db_private ];
20 then
21   echo "THIS IS MAIN DB"
22   expect /opt/openbaton/scripts/setup-shard.exp $db_private
23 else
24   echo "THIS IS EXTENDED DB"
25   ##Add new DB hostname to MAIN DB
26   ssh -o StrictHostKeyChecking=no -i /openstack_key.pem -l ubuntu
       $MAINDB_IP "echo $THISDB_INFO | sudo tee --append /etc/hosts"
27   ##Add Main DB hostname to extend DB
28   THISDB_IP=$(awk -F' ' '{ print $1}' $DBFILE | tail -1)
29   ssh -o StrictHostKeyChecking=no -i /openstack_key.pem -l ubuntu
       $db_private "echo $MAINDB_INFO | sudo tee --append /etc/hosts"
30   REPLSET=$(ssh -o StrictHostKeyChecking=no -i /openstack_key.pem -l
        ubuntu $db_private "head -1 /home/ubuntu/db_info.conf")
31   REPLSETID="$REPLSET/$THISDB_NAME:27017"
32   mongo --host $MAINDB_IP --port 27020 --eval "sh.addShard( '
       $REPLSETID' )"
33 fi
34 echo "FINISED at:" $(date)
```

ข5. start_nip.sh

ไฟล์ start_nip.sh สำหรับเหตุการณ์ START ของเครื่องเสมือนเอ็นไอพีมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-6: การกำหนดพารามิเตอร์ที่ใช้งานสำหรับชุดคำสั่งนี้
- บรรทัดที่ 7-23: การตรวจสอบการทำงานของแพลตฟอร์มโอเพนเอ็มทีซีของเครื่องเสมือน

เอ็นเอสซีแอล เมื่อเครื่องเสมือนเอ็นเอสซีแอลพร้อมให้บริการแพลตฟอร์มโอเพนเอ็มทีซีแล้ว เครื่องเสมือนเอ็นไอพีจะเริ่มทำงานร่วมกันแบบทันทีระหว่างหน่วยเก็บข้อมูลของระบบ CU-BEMS และแพลตฟอร์มโอเพนเอ็มทีซี

```
1  #!/bin/bash
2  echo "START NIP"
3  echo "START at:" $(date)
4  source /root/nscl_info.conf
5  NSCL_IP=$nscl_ip
6  echo $NSCL_IP
7  sleep 30
8  for i in {1..10}
9  do
10   nc -z -v $NSCL_IP 15000
11   if [ $? -eq 0 ]; then
12     sudo nohup  node /OpenMTC-Chula/openmtc-NIP/ProxyGateway/
          NIP_IEEE1888_ETSI.js >/home/ubuntu/ieee.log 2>/home/ubuntu/
          ieee.err &
13     sleep 10
14     sudo nohup node /OpenMTC-Chula/openmtc-NIP/ProxyGateway/
          NIP_ETSI_IEEE1888_nscl.js >/home/ubuntu/etsi.log 2>/home/
          ubuntu/etsi.err &
15     echo "NSCL started:$i on start_nscl"
16     break
17   else
18     echo "NSCL stopped:$i on start_nscl"
19     sleep 15
20     fi
21 done
22 echo "STOP at:" $(date)
```

ข6. install_nscl.sh

ไฟล์ install_nscl.sh สำหรับการทำงานในเหตุการณ์ INSTANTIATE ของเครื่องเสมือนเอ็นเอสซีแอลมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 6-9: การตั้งค่าและประสานเวลาของเครื่องเอ็นเอสซีแอลกับ ntp.chula.ac.th เพื่อให้เครื่องเสมือนเอ็นเอสซีแอลมีเวลาตรงกับเวลาของระบบ CU-BEMS
- บรรทัดที่ 10-13: การบันทึกค่าหมายเลขไอพีส่วนบุคคลของเครื่องเสมือนเอ็นเอสซีแอลเพื่อให้ชุดคำสั่งต่าง ๆ สามารถเรียกใช้ได้
- บรรทัดที่ 14-19: การตรวจสอบการเชื่อมต่ออินเทอร์เน็ตของเครื่องเสมือนเอ็นเอสซีแอล
- บรรทัดที่ 20-26: การกำหนดให้เครื่องเสมือนเอ็นเอสซีแอลให้บริการแพลตฟอร์มโอเพนเอ็มทีซีทุกครั้งที่เปิดเครื่อง

```
1  #!/bin/bash
2  echo "STEP1: INSTANTIATE NSCL"
3  echo "START at:" $(date)
4  NSCL_START_TIME=$(date)
5  myhome=${HOME}
6  dpkg-reconfigure -f noninteractive tzdata
7  apt-get update
8  apt-get install --reinstall tzdata -y
9  ntpq -p
10 echo ${private1} | sudo tee ./ipaddress.txt
11 ipfile="./ipaddress.txt"
12 ipaddress=$(head -1 $ipfile)
13 bash -c "echo $ipaddress `cat /etc/hostname` >> /etc/hosts"
14 wget -q --tries=10 --timeout=20 --spider  http://archive.ubuntu.com
15 if [[ $? -eq 0 ]]; then
16         echo "Server can connect to Internet"
17 else
18         echo "Server cannot connect to Internet"
19 fi
20 echo "nscl_ip=${private}" | sudo tee --append ${HOME}/nscl_info.conf
21 echo "nscl_fip=${private_floatingIp}" | sudo tee --append ${HOME}/
       nscl_info.conf
22 cp /opt/openbaton/scripts/start-nscl.sh /etc/init.d/start-nscl.sh
23 chmod ugo+x /etc/init.d/start-nscl.sh
24 update-rc.d start-nscl.sh defaults
25 cp /opt/openbaton/scripts/openstack.info /home/ubuntu/openstack.info
26 echo "FINISH at:" $(date)
```

ข7. client_configNSCL.sh

ไฟล์ client_configNSCL.sh สำหรับเหตุการณ์ CONFIGURE ของเครื่องเสมือนเอ็นเอสซีแอลมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-7: การบันทึกหมายเลขไอพีและชื่อของเครื่องเสมือนเอ็นไอพีลงสู่เครื่องเสมือนเอ็นเอสซีแอล

```
1  #!/bin/bash
2  echo "client configure at NSCL"
3  echo "START at:" $(date)
4  echo "nip_ip=$client_private" | sudo tee --append ${HOME}/nip_info.
       conf
5  echo "nip_fip=$client_private_floatingIp" | sudo tee --append ${HOME
       }/nip_info.conf
6  echo "nip_hostname=$client_hostname" | sudo tee --append ${HOME}/
       nip_info.conf
7  echo "STOP at:" $(date)
```

ข8. db_config.sh

ไฟล์ db_config.sh สำหรับเหตุการณ์ CONFIGURE ของเครื่องเสมือนเอ็นเอสซีแอลมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-8: การบันทึกหมายเลขไอพีและชื่อของเครื่องเสมือนมองโกดีบีลงสู่เครื่องเสมือนเอ็นเอสซีแอล

```
1 #!/bin/bash
2 echo "START at:" $(date)
3 echo "DB configure at " $(hostname)
4 echo "db_ip=$db_private" | sudo tee --append ${HOME}/db_info.conf
5 echo "db_fip=$db_private_floatingIp" | sudo tee --append ${HOME}/
    db_info.conf
6 echo "db_hostname=$db_hostname" | sudo tee --append ${HOME}/db_info.
    conf
7 echo "exports.ipdb='$db_private';" | tee --append /OpenMTC-Chula/
    openmtc/settings/ipserv.js
8 echo "FINISHED at:" $(date)
```

ข9. start_nscl.sh

ไฟล์ start_nscl.sh สำหรับเหตุการณ์ START ของเครื่องเสมือนเอ็นเอสซีแอลมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-7: การกำหนดพารามิเตอร์ที่ใช้งานสำหรับชุดคำสั่งนี้
- บรรทัดที่ 8-20: การตรวจสอบการทำงานของฐานข้อมูลมองโกดีบีที่เครื่องเสมือนมองโกดีบี บรรทัดที่ 13-16 ในกรณที่ฐานข้อมูลมองโกดีบีพร้อมให้บริการ เครื่องเสมือนเอ็นเอสซีแอลจะเริ่มทำงาน หลังจากนั้นเครื่องเสมือนเอ็นเอสซีแอลจะเพิ่มหมายเลขไอพีของตัวเครื่องเสมือนนี้เข้าสู่บริการ LBaaS ของโปรแกรมโอเพนสแต็ก
- บรรทัดที่ 21-28: การร้องขอโทเค็นจากโอเพนสแต็กในโครงการชื่อ admin
- บรรทัดที่ 29-31: การร้องขอข้อมูลของโครงข่ายชื่อ private1_subnet ที่สร้างไว้
- บรรทัดที่ 32-37: การร้องขอข้อมูลของบริการ LBaaS ชื่อ loadbalancer1 ที่สร้างไว้
- บรรทัดที่ 38-46: การเพิ่มหมายหมายเลขไอพีของตัวเครื่องเสมือนนี้เข้าสู่บริการ LBaaS ของโปรแกรมโอเพนสแต็ก
- บรรทัดที่ 47-59: การบันทึกการทำงานในกรณีที่เครื่องเสมือนเอ็นเอสซีแอลหรือเครื่องเสมือนมองโกดีบีไม่สามารถให้บริการได้

```
1 #!/bin/bash
2 echo "START at:" $(date)
3 source /root/db_info.conf
4 dbhost=$db_ip
5 ipfile="./ipaddress.txt"
```

```
6  ipaddress=$(head -1 $ipfile)
7  source /root/nip_info.conf
8  LINK1="http://$ipopenstack:9696/v2.0"
9  for i in {1..5}
10 do
11   nc -z -v $dbhost 27017
12   if [ $? -eq 0 ]; then
13     echo "Mongos have already started"
14     nohup node /OpenMTC-Chula/nscl >/home/ubuntu/nscl.log 2>/home/
           ubuntu/nscl.err &
15     sleep 15
16     expect /opt/openbaton/scripts/init-shard.exp $dbhost
17     for j in {1..5}
18     do
19     nc -z -v $ipaddress 15000
20     if [ $? -eq 0 ];  then
21       echo "NSCL has been started"
22       source /home/ubuntu/openstack.info
23       curl -s -X POST $OS_AUTH_URL/tokens -H "Content-Type: \
24        application/json" -d '{"auth": {"tenantName": "admin", \
25        "passwordCredentials": {"username": "'"$OS_USERNAME"'", \
26        "password": "'"$OS_PASSWORD"'"}}}' \
27        | python -m  json.tool > token.info
28       TOKEN=$(cat token.info |  jq '.access.token.id' | tr -d '"')
29       curl -s -X GET $LINK1/subnets?name=private1_subnet \
30        -H "X-Auth-Token: $TOKEN" | python -m json.tool > subnet.info
31       subnet=$(cat subnet.info | jq '.subnets[].id' | tr -d '"')
32       curl -s -X GET $LINK1/loadbalancers?name=loadbalancer1  \
33        -H "X-Auth-Token: $TOKEN" | python -m json.tool > lbaas.txt
34       vip=$(cat lbaas.txt | jq '.loadbalancers[].vip_address' \
35        | tr -d '"')
36       pools=$(cat lbaas.txt | jq '.loadbalancers[].pools[].id' \
37        | tr -d '"')
38       curl -s -X POST $LINK1/lbaas/pools/$pools/members  \
39        -H "X-Auth-Token: $TOKEN" \
40        -d '{"member": {"address": "'"$ipaddress"'", \
41       "name": "'"$ipaddress"'","subnet_id": "'"$subnet"'", \
42       "protocol_port": "15000"}}'
43       curl -s -X GET $LINK1/lbaas/pools/$pools/members \
44        -H "X-Auth-Token: $TOKEN" | python -m json.tool
45       ssh -o StrictHostKeyChecking=no -i /openstack_key.pem \
46        -l ubuntu $nip_ip "${HOME}/add-iplbaas.sh $ipaddress"
47       break
48       else
49       echo "NSCL is not working on NSCL"
50       sleep 5
```

```
51        fi
52      done
53      break
54    else
55      echo "Mongos stopped:$i on start_nscl"
56          sleep 15
57    fi
58  done
59  echo "FINISH at:" $(date)
```

ข10. terminate_lbaas.sh

ไฟล์ terminate_lbaas.sh สำหรับการทำงานในเหตุการณ์ TERMINATE ของเครื่องเสมือนเอ็นเอสซีแอลมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-8: การกำหนดพารามิเตอร์ที่ใช้งานสำหรับชุดคำสั่งนี้
- บรรทัดที่ 9-30: เครื่องเสมือนเอ็นเอสซีแอลจะลบหมายเลขไอพีของตัวเครื่องเสมือนนี้ออกจากบริการ LBaaS ของโปรแกรมโอเพนสแต็กในกรณีที่ยุติฟังก์ชันเสมือน

```
1  #!/bin/bash
2  echo "TERMINATED iplbaas at" $(hostname)
3  echo "START at:" $(date)
4  source /home/ubuntu/openstack.info
5  source /root/nip_info.conf
6  ipfile="./ipaddress.txt"
7  ip=$(head -1 $ipfile)
8  LINK1="http://$ipopenstack:9696/v2.0"
9  curl -s -X POST $OS_AUTH_URL/tokens \
10  -H "Content-Type: application/json" \
11  -d '{"auth": {"tenantName": "admin", "passwordCredentials": \
12  {"username":"'"$OS_USERNAME"'","password":"'"$OS_PASSWORD"'"}}}' \
13  | python -m  json.tool > token.txt
14 TOKEN=$(cat token.txt |  jq '.access.token.id' | tr -d '"')
15 curl -s -X GET $LINK1/lbaas/loadbalancers?name=loadbalancer1 \
16  -H "X-Auth-Token: $TOKEN" | python -m json.tool > lbaas.txt
17 pools=$(cat lbaas.txt | jq '.loadbalancers[].pools[].id'|tr -d '"')
18 curl -s -X GET $LINK1/subnets?name=private1_subnet \
19  -H "X-Auth-Token: $TOKEN" | python -m json.tool > subnet.info
20 subnet=$(cat subnet.info | jq '.subnets[].id' | tr -d '"')
21 curl -s -X GET $LINK1/lbaas/pools/$pools/members?address="$ip" \
22  -H "X-Auth-Token: $TOKEN" | python -m json.tool > lbaasmember.info
23 MEMBER_ID=$(cat lbaasmember.info | jq '.members[].id' | tr -d '"')
24 curl -s -X DELETE $LINK1/lbaas/pools/$pools/members/$MEMBER_ID \
25  -H "X-Auth-Token: $TOKEN"
26 curl -s -X GET $LINK1/lbaas/pools/$pools/members \
27  -H "X-Auth-Token: $TOKEN" | python -m json.tool
28 ssh -o StrictHostKeyChecking=no -i /openstack_key.pem \
```

```
29  -l ubuntu $nip_ip "/opt/openbaton/scripts/del-iplbaas.sh"
30 echo "FINISH at:" $(date)
```

ข11. scale_in_lbaas.sh
ไฟล์ scale_in_lbaas.sh สำหรับเหตุการณ์ SCALE_IN ของเครื่องเสมือนเอ็นเอสซีแอลมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-6: การกำหนดพารามิเตอร์ที่ใช้งานสำหรับชุดคำสั่งนี้
- บรรทัดที่ 7-32: เครื่องเสมือนเอ็นเอสซีแอลจะลบหมายเลขไอพีของตัวเครื่องเสมือนนี้ออกจากบริการ LBaaS ของโปรแกรมโอเพนสแต็กในกรณีที่ลดจำนวนเครื่องเสมือนลงเมื่อเป็นไปตามเงื่อนไขของการปรับลดทรัพยากรระบบ
- บรรทัดที่ 33-35: การตรวจสอบเครื่องเสมือนเอ็นเอสซีแอลที่คงเหลือในบริการ LBaaS

```
1  #!/bin/bash
2  source /home/ubuntu/openstack.info
3  ipaddress=$1
4  LINK1="http://$ipopenstack:9696/v2.0"
5  echo "Perform by" $ipaddress "at" $(date) >> log_iplbaas.log
6  iplbaas=$(cat ${HOME}/iplbaas.info)
7  for ip in $iplbaas
8  do
9    nc -z -v -w 1 $ip 15000
10   if [ $? -eq 0 ]; then
11     echo $ip "is reachable at" $(date) >> log_iplbaas.log
12   else
13     echo $ip "is unreachable at" $(date) >> log_iplbaas.log
14     curl -s -X POST $OS_AUTH_URL/tokens \
15      -H "Content-Type: application/json" \
16      -d '{"auth": {"tenantName": "admin", "passwordCredentials": \
17      {"username": "'"$OS_USERNAME"'", "password": "'"$OS_PASSWORD"'
            "}}}' \
18      | python -m  json.tool > token.txt
19     TOKEN=$(cat token.txt |  jq '.access.token.id' | tr -d '"')
20     curl -s -X GET $LINK1/lbaas/loadbalancers?name=loadbalancer1 \
21      -H "X-Auth-Token: $TOKEN" | python -m json.tool > lbaas.txt
22     pools=$(cat lbaas.txt | jq '.loadbalancers[].pools[].id' | tr -d
           '"')
23     curl -s -X GET $LINK1/lbaas/pools/$pools/members?address="$ip" \
24      -H "X-Auth-Token: $TOKEN" | python -m json.tool > lbaasmember.
           info
25     MEMBER_ID=$(cat lbaasmember.info | jq '.members[].id' | tr -d '"
          ')
26     curl -s -X DELETE $LINK1/lbaas/pools/$pools/members/$MEMBER_ID \
27      -H "X-Auth-Token: $TOKEN"
28      sed -i -e "/$ip/d" ${HOME}/iplbaas.info
```

```
29      echo "DELETE:" $ip at $(date) >> log_iplbaas.log
30      sleep 5
31    fi
32  done
33  curl -s -X GET $LINK1/lbaas/pools/$pools/members \
34  -H "X-Auth-Token: $TOKEN" | python -m json.tool
35  echo "FINISH at:" $(date)
```

ข12. install_db.sh

ไฟล์ install_db.sh สำหรับเหตุการณ์ INSTANTIATE ของเครื่องเสมือนมองโกดีบีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 6: การกำหนดชื่อจำนวน 3 ตัวอักษรแบบสุ่มให้กับฐานข้อมูลมองโกดีบีในเครื่องเสมือนมองโกดีบี
- บรรทัดที่ 6-10: การตั้งค่าและประสานเวลาของเครื่องมองโกดีบีกับ ntp.chula.ac.th เพื่อให้เครื่องเสมือนมองโกดีบีมีเวลาตรงกับเวลาของระบบ CU-BEMS
- บรรทัดที่ 11-14: การบันทึกค่าหมายเลขไอพีส่วนบุคคลของเครื่องเสมือนมองโกดีบีเพื่อให้ชุดคำสั่งต่าง ๆ สามารถเรียกใช้ได้
- บรรทัดที่ 15-20: การตรวจสอบการเชื่อมต่ออินเทอร์เน็ตของเครื่องเสมือนมองโกดีบี
- บรรทัดที่ 21-25: การนำชุดคำสั่งสำหรับตั้งค่าให้ฐานข้อมูลมองโกดีบีไปยังตำแหน่งที่กำหนดไว้

```
1  #!/bin/bash
2  echo "STEP: INSTANTIATE MONGODB"
3  echo "START at:" $(date)
4  MYHOME=${HOME}
5  THISHOST=$(hostname)
6  date +%s |sha256sum|base64|head -c 3 >> /home/ubuntu/db_info.conf
7  dpkg-reconfigure -f noninteractive tzdata
8  apt-get update
9  apt-get install --reinstall tzdata -y
10 ntpq -p
11 echo ${private1} | sudo tee ./ipaddress.txt
12 ipfile="./ipaddress.txt"
13 ipaddress=$(head -1 $ipfile)
14 bash -c "echo $ipaddress $THISHOST >> /etc/hosts"
15 wget -q --tries=10 --timeout=20 --spider  http://archive.ubuntu.com
16 if [[ $? -eq 0 ]]; then
17         echo "Server can connect to Internet"
18 else
19         echo "Server cannot connect to Internet"
20 fi
21 cp /opt/openbaton/scripts/initshard.js /home/ubuntu/initshard.js
```

```
22 cp /opt/openbaton/scripts/initshard_bigfile.js /home/ubuntu/
       initshard_bigfile.js
23 chown ubuntu:ubuntu /home/ubuntu/initshard.js /home/ubuntu/
       initshard_bigfile.js
24 chmod 755 /home/ubuntu/initshard.js /home/ubuntu/initshard_bigfile.
       js
25 echo "FINISHED at:" $(date)
```

ข13. server_config.sh
ไฟล์ server_config.sh สำหรับเหตุการณ์ CONFIGURE ของเครื่องเสมือนมองโกดีบีมีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-20: การบันทึกหมายเลขไอพีและชื่อของเครื่องเสมือนเอ็นเอสซีแอลลงสู่เครื่องเสมือนมองโกดีบี

```
1  #!/bin/bash
2  echo "START server configure at:" $(date)
3  ipfile="./ipaddress.txt"
4  echo $(hostname)
5  ipaddress=$(head -1 $ipfile)
6  echo "nscl_ip=$server_private" \
7   | tee --append ${HOME}/nscl_info.conf
8  echo "nscl_fip=$server_private_floatingIp" \
9   | tee --append ${HOME}/nscl_info.conf
10 echo "nscl_hostname=$server_hostname" \
11  | tee --append ${HOME}/nscl_info.conf
12 echo "exports.ipnscl='$server_private';" \
13  | tee --append /OpenMTC-Chula/openmtc/settings/ipserv.js
14 echo "exports.ipnip='$ipaddress';" \
15  | tee --append /OpenMTC-Chula/openmtc/settings/ipserv.js
16 echo "exports.fipnscl='$server_private_floatingIp';" \
17  | tee --append /OpenMTC-Chula/openmtc/settings/ipserv.js
18 echo "exports.ipopenstack='$publicip';" \
19  | tee --append /OpenMTC-Chula/openmtc/settings/ipserv.js
20 echo "STOP at:" $(date)
```

ข14. client_configdb.sh
ไฟล์ client_configdb.sh สำหรับเหตุการณ์ CONFIGURE ของเครื่องเสมือนมองโกดีบีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 1-12: การกำหนดพารามิเตอร์ที่ใช้งานสำหรับชุดคำสั่งนี้
- บรรทัดที่ 13-25: การตรวจสอบหมายเลขไอพีและสถานะของเครื่องเสมือนมองโกดีบีเครื่องนี้
- บรรทัดที่ 26-50: ในกรณีที่เครื่องเสมือนมองโกดีบีเครื่องนี้เป็นเครื่องลำดับที่หนึ่ง ชุดคำสั่งจะกำหนดให้เครื่องเสมือนมองโกดีบีเครื่องนี้ติดตั้งระบบขยายฐานข้อมูลของฐานข้อมูลมองโกดีบี

- บรรทัดที่ 51-75: ในกรณีที่เครื่องเสมือนมองโกดีบีเครื่องนี้เป็นเครื่องเสมือนที่ถูกสร้างขึ้นมาแบบอัติโนมัติเพื่อเพิ่มพื้นที่จัดเก็บข้อมูลให้ฐานข้อมูลมองโกดีบี ชุดคำสั่งกำหนดให้เครื่องเสมือนมองโกดีบีเครื่องนี้ลงทะเบียนในระบบขยายฐานข้อมูลของฐานข้อมูลมองโกดีบีที่มีอยู่แล้ว

```
1  #!/bin/bash
2  THISHOST=$(hostname)
3  ipfile="./ipaddress.txt"
4  THISIP=$(head -1 $ipfile)
5  echo "client configure at DB-SERVER START at:" $(date)
6  echo "nip_ip=$client_private" \
7   | sudo tee --append ${HOME}/nip_info.conf
8  echo "nip_fip=$client_private_floatingIp" \
9   | sudo tee --append ${HOME}/nip_info.conf
10 echo "nip_hostname=$client_hostname" \
11  | sudo tee --append ${HOME}/nip_info.conf
12 MAINDB_IP=''
13 for i in {1..5}
14 do
15   MAINDB_IP=$(ssh -o StrictHostKeyChecking=no \
16    -i /openstack_key.pem  \
17    "sudo head -1 /home/ubuntu/maindb_info.conf")
18    if [ -z "$MAINDB_IP" ]; then
19     echo "MAINDB_IP=NULL"
20     sleep 15
21   else
22     echo "MAINDB_IP:" $MAINDB_IP
23     break
24   fi
25 done
26 SERVICE='mongos'
27 if [ $MAINDB_IP = $THISIP ]; then
28   echo "STEP: REGISTER MAIN SHARD"
29   REPLSET=$(head -1 /home/ubuntu/db_info.conf)
30   sudo rm -rf /var/lib/mongod
31   MYHOST="$REPLSET/$THISHOST:27017"
32   for i in {1..5} do
33     nc -z -v $MAINDB_IP 27020
34     if [ $? -eq 0 ]; then
35       echo "$SERVICE is running!!!"
36        mkdir /var/lib/mongod
37        mongod --shardsvr --replSet $REPLSET --dbpath /var/lib/mongod
              \
38         --fork --syslog --port 27017
39        sleep 3
40        mongo --port 27017 --eval "rs.initiate()"
```

```
41          sleep 3
42          mongo --host $MAINDB_IP --port 27020 \
43           --eval "sh.addShard( '$MYHOST' )"
44          break
45        else
46          echo "$service stopped:$i on client_configuredb.sh"
47          sleep 15
48        fi
49      done
50      echo "FINISHED at:" $(date)
51    else
52      echo "STEP: REGISTER extended SHARD"
53      REPLSET=$(head -1 /home/ubuntu/db_info.conf)
54      rm -rf /var/lib/mongod
55      MYHOST="$REPLSET/$THISHOST:27017"
56      for i in {1..5} do
57        nc -z -v $MAINDB_IP 27020
58        if [ $? -eq 0 ]; then
59          echo "$SERVICE is running!!!"
60          mkdir /var/lib/mongod
61          mongod --shardsvr --replSet $REPLSET --dbpath /var/lib/mongod
              \
62           --fork --syslog --port 27017
63          sleep 3
64          mongo --port 27017 --eval "rs.initiate()"
65          sleep 5
66          mongo --host $MAINDB_IP --port 27020
67           --eval "sh.addShard( '$MYHOST' )"
68          break
69        else
70          echo "$service stopped:$i on client_configuredb.sh"
71          sleep 15
72        fi
73      done
74      echo "FINISHED at:" $(date)
75    fi
```

ภาคผนวก ค
ฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่าง มาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite

nsd_echonet_ieee1888.json

ไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite มีรายละเอียดการทำงานดังนี้

- บรรทัดที่ 2-4: การระบุชื่อไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับฟังก์ชันเสมือนการทำงานร่วมกันแบบทันทีระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ECHONET Lite รวมถึงชื่อผู้สร้าง และรุ่นของฟังก์ชันเสมือน
- บรรทัดที่ 5-10: การกำหนดชื่อเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ ประเภทของเครื่องเสมือนและการกำหนดพารามิเตอร์ให้กับเครื่องเสมือน
- บรรทัดที่ 11-25: การกำหนดพารามิเตอร์สำหรับเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ ซึ่งพารามิเตอร์ที่กำหนดให้กับฟังก์ชันเสมือนนี้ ได้แก่ พารามิเตอร์ TempUpper และ TempLower ใช้สำหรับการกำหนดระดับอุณภหูมิที่ใช้ควบคุมการทำงานของเครื่องปรับอากาศภายในห้องคลัสเตอร์ และพารามิเตอร์ TempSet ใช้สำหรับการกำหนดอุณหภูมิของเครื่องปรับอากาศ
- บรรทัดที่ 27: การกำหนดชื่อไฟล์ระบบปฏิบัติการที่มีอยู่บนแพลตฟอร์มโอเพนสแต็กให้ติดตั้งในเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์
- บรรทัดที่ 28: การระบุโนดโอเพนสแต็กที่ลงทะเบียนไว้ในโปรแกรมโอเพนบาตง เพื่อใช้สำหรับสร้างเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์
- บรรทัดที่ 29: การระบุจำนวนเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์สูงสุดที่สามารถสร้างขึ้นได้
- บรรทัดที่ 30-41: การระบุหมายเลขไอพีของโครงข่ายเสมือนให้เครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ สำหรับฟังก์ชันเสมือนนี้ เครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์จะเชื่อมต่อกับโครงข่ายเสมือนจำนวน 2 โครงข่าย
- บรรทัดที่ 42-45: การระบุโครงข่ายเสมือนสำหรับเชื่อมต่อกับเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ ได้แก่ โครงข่ายเสมือน private1 และโครงข่ายเสมือน private_echonet
- บรรทัดที่ 46-50: การระบุชื่อชุดคำสั่งสำหรับการทำงานในเหตุการณ์ INSTANTIATE เพื่อติดตั้งโปรแกรมที่จำเป็นสำหรับการสร้างฟังก์ชันเสมือนนี้และปรับแต่งค่าพารามิเตอร์เช่น หมายเลขไอพี ชื่อของเครื่องเสมือนภายในเครื่องของตัวเอง และการเชื่อมต่อกับโครงข่ายเสมือน

- บรรทัดที่ 51-55: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ CONFIGURE เพื่อตั้งค่าพารามิเตอร์ให้กับเครื่องเสมือนและกำหนดช่วงเวลาการทำงานของฟังก์ชันเสมือน
- บรรทัดที่ 56-61: การระบุชื่อชุดคำสั่งสำหรับการทำงานในเหตุการณ์ START เพื่อเริ่มต้นการทำงานของฟังก์ชันเสมือนนี้
- บรรทัดที่ 62: การกำหนดรายละเอียดของเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ซึ่งประกอบด้วยจำนวนหน่วยประมวลผลเสมือน แรมและพื้นที่เก็บข้อมูลของเครื่องเสมือน
- บรรทัดที่ 63: ยูอาร์แอลระบุตำแหน่งที่เก็บไฟล์ข้อมูลสำหรับการสร้างฟังก์ชันเสมือนนี้
- บรรทัดที่ 66: การระบุโครงข่ายทอพอโลยีที่ใช้สำหรับการสร้างฟังก์ชันเสมือนนี้

```
1  {
2    "name":"echonet",
3    "vendor":"NRG-team",
4    "version":"0.4",
5    "vnfd":[{
6      "vendor":"NRG-team",
7      "version":"0.1",
8      "name":"echonet-gw",
9      "type":"server",
10     "endpoint":"generic",
11     "configurations":{
12       "name":"config_echonet",
13       "configurationParameters":[{
14         "confKey":"TempUpper",
15         "value":"23"
16       },
17       {
18         "confKey":"TempLower",
19         "value":"22"
20       },
21       {
22         "confKey":"TempSet",
23         "value":"21"
24       }]
25     },
26     "vdu":[{
27       "vm_image":["ubun1604"],
28       "vimInstanceName":["vim14"],
29       "scale_in_out":1,
30       "vnfc":[{
31         "connection_point":[{
32           "floatingIp":"random",
33           "virtual_link_reference":"private1",
34           "interfaceId":0
```

```
35          },
36          {
37          "virtual_link_reference":"private_echonet",
38          "interfaceId":1
39          }]
40        }]
41      }],
42      "virtual_link":[
43        {"name":"private1"},
44        {"name":"private_echonet"}
45      ],
46      "lifecycle_event":[{
47        "event":"INSTANTIATE",
48        "lifecycle_events":[
49          "install_echonet.sh"
50        ]},
51        {
52        "event":"CONFIGURE",
53        "lifecycle_events":[
54          "server_configure.sh"
55        ]},
56        {
57        "event":"START",
58        "lifecycle_events":[
59          "start_echonet.sh"
60        ]}
61      ],
62      "deployment_flavour":[{"flavour_key":"m1.small"}],
63      "vnfPackageLocation":"https://github.com/chart2023/echonet1.git"
64      }],
65    "vnffgd":[],
66    "vld":[{"name":"private1"}]
67  }
```

ภาคผนวก ง
ฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888

ง1. nsd_analytic.json

ไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 มีรายละเอียดการทำงานดังนี้

- บรรทัดที่ 2-4: การระบุชื่อไฟล์ข้อมูลรายละเอียดบริการโครงข่ายสำหรับฟังก์ชันเสมือนโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐานมาตรฐาน IEEE1888 รวมถึงชื่อผู้สร้าง และรุ่นของฟังก์ชันเสมือน
- บรรทัดที่ 5-10: การกำหนดชื่อเครื่องเสมือนสำหรับการวิเคราะห์ข้อมูล ประเภทของเครื่องเสมือนและการกำหนดพารามิเตอร์ให้กับเครื่องเสมือน
- บรรทัดที่ 11-29: การกำหนดค่าพารามิเตอร์ให้กับเครื่องเสมือนสำหรับการวิเคราะห์ข้อมูล ซึ่งพารามิเตอร์ที่กำหนดให้กับฟังก์ชันเสมือนนี้ ได้แก่ พารามิเตอร์ยูอาร์แอลสำหรับข้อมูลการใช้งานโปรแกรมสร้างมโนภาพเชิงโต้ตอบบริเวณชั้น 13 ในระบบ CU-BEMS และโทเค็นของโปรแกรมประยุกต์ไลน์เพื่อใช้สำหรับการแจ้งเตือน
- บรรทัดที่ 31: การกำหนดชื่อไฟล์ระบบปฏิบัติการที่มีอยู่บนแพลตฟอร์มโอเพนสแต็กให้ติดตั้งในเครื่องเสมือนสำหรับการวิเคราะห์ข้อมูล
- บรรทัดที่ 32: การระบุโนดโอเพนสแต็กที่ลงทะเบียนไว้ในโปรแกรมโอเพนบาตง เพื่อใช้สำหรับสร้างเครื่องเสมือนสำหรับวิเคราะห์ข้อมูล
- บรรทัดที่ 33: การระบุจำนวนเครื่องเสมือนสำหรับวิเคราะห์ข้อมูลสูงสุดที่สามารถสร้างขึ้นได้
- บรรทัดที่ 34-45: การระบุหมายเลขไอพีของโครงข่ายเสมือนและลำดับของโครงข่ายให้กับเครื่องเสมือนสำหรับวิเคราะห์ข้อมูล สำหรับฟังก์ชันเสมือนนี้ เครื่องเสมือนสำหรับวิเคราะห์ข้อมูลจะเชื่อมต่อกับโครงข่ายเสมือนจำนวน 2 โครงข่าย
- บรรทัดที่ 46-49: การระบุชื่อโครงข่ายเสมือนที่ใช้เชื่อมต่อเครื่องเสมือนสำหรับวิเคราะห์ข้อมูล ได้แก่ โครงข่ายเสมือน private1 และโครงข่ายเสมือน private_bems1
- บรรทัดที่ 50-54: การระบุชื่อชุดคำสั่งสำหรับการทำงานในเหตุการณ์ INSTANTIATE เพื่อติดตั้งโปรแกรมที่จำเป็นสำหรับการสร้างฟังก์ชันเสมือนนี้และปรับแต่งค่าพารามิเตอร์เช่น หมายเลขไอพี ชื่อของเครื่องเสมือนภายในเครื่องของตัวเอง และการเชื่อมต่อกับโครงข่ายเสมือน
- บรรทัดที่ 55-59: การระบุชื่อชุดคำสั่งสำหรับเหตุการณ์ CONFIGURE เพื่อตั้งค่าพารามิเตอร์ให้กับเครื่องเสมือนและกำหนดช่วงเวลาการทำงานของฟังก์ชันเสมือน

- บรรทัดที่ 60-65: การระบุชื่อชุดคำสั่งสำหรับการทำงานในเหตุการณ์ START เพื่อเริ่มต้นการทำงานของฟังก์ชันเสมือนนี้
- บรรทัดที่ 66: การกำหนดรายละเอียดของเครื่องเสมือนอินเตอร์เวิร์กกิงพร็อกซีเกตเวย์ซึ่งประกอบด้วยจำนวนหน่วยประมวลผลเสมือน แรมและพื้นที่เก็บข้อมูลของเครื่องเสมือน
- บรรทัดที่ 67-68: ยูอาร์แอลระบุตำแหน่งที่เก็บไฟล์ข้อมูลสำหรับการสร้างฟังก์ชันเสมือนนี้
- บรรทัดที่ 70: การระบุโครงข่ายทอพอโลยีที่ใช้สำหรับการสร้างฟังก์ชันเสมือนนี้

```
1  {
2    "name":"idaas",
3    "vendor":"NRG-team",
4    "version":"0.4",
5    "vnfd":[{
6      "vendor":"NRG-team",
7      "version":"0.1",
8      "name":"idaas-serv",
9      "type":"server",
10     "endpoint":"generic",
11     "configurations":{
12       "name":"config_idaas",
13       "configurationParameters":[{
14         "confKey":"num_swipeid",
15         "value":"http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/
               elevatorfront/kinect/num_swipe"
16         },
17         {
18         "confKey":"num_userid",
19         "value":"http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/
               elevatorfront/kinect/num_user"
20         },
21         {
22         "confKey":"kinect",
23         "value":"http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/
               elevatorfront/kinect/status"
24         },
25         {
26         "confKey":"linenotify",
27         "value":"LF6aderIPii6RUoB4EDxoGOCKU9vQ4B8wkcHVi4A8df"
28         }
29     ]},
30     "vdu":[{
31       "vm_image":["ubuntu1604"],
32       "vimInstanceName":["vim14"],
33       "scale_in_out":1,
34       "vnfc":[{
```

```
35          "connection_point":[{
36            "floatingIp":"random",
37            "virtual_link_reference":"private1",
38            "interfaceId":0
39          },
40          {
41            "virtual_link_reference":"private_bems1",
42            "interfaceId":1
43          }]
44        }]
45      }],
46      "virtual_link":[
47        {"name":"private1"},
48        { "name":"private_bems1"}
49      ],
50      "lifecycle_event":[{
51        "event":"INSTANTIATE",
52        "lifecycle_events":[
53          "install_idaas.sh"
54        ]},
55        {
56        "event":"CONFIGURE",
57        "lifecycle_events":[
58          "server_configure.sh"
59        ]},
60        {
61        "event":"START",
62        "lifecycle_events":[
63          "start_idaas.sh"
64        ]}
65      ],
66      "deployment_flavour":[{"flavour_key":"m1.small"}],
67      "vnfPackageLocation":"https://github.com/chart2023/checkidaas.
           git"
68    }],
69    "vnffgd":[],
70    "vld":[{"name":"private1"}]
71  }
```

ง2. create_wastedpic.php

ไฟล์ create_wastedpic.php สำหรับการสร้างรูปภาพแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศในแต่ละพื้นที่ของอาคารผ่านทางโปรแกรมมโนภาพเชิงโต้ตอบ มีรายละเอียดการทำงานของโปรแกรมดังนี้

- บรรทัดที่ 2-3: การเรียกใช้โมดูลสำหรับการสร้างกราฟชื่อ jpgraph
- บรรทัดที่ 4-5: การกำหนดวันที่ปัจจุบันและวันก่อนหน้าเพื่อใช้สำหรับการตรวจสอบข้อมูล

- บรรทัดที่ 6: การกำหนดรูปภาพผังของอาคารเพื่อนำมาเป็นพื้นหลังของภาพแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศ
- บรรทัดที่ 7-13:การกำหนดตำแหน่ง point ID ในระบบ CU-BEMS ที่ต้องใช้สำหรับกราฟที่จะสร้างขึ้น
- บรรทัดที่ 14-15: การเรียกใช้ฟังก์ชัน uuid เพื่อใช้สำหรับการร้องขอข้อมูลจากระบบ CU-BEMS
- บรรทัดที่ 16-19: การร้องขอข้อมูลค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศล่าสุดจากระบบ CU-BEMS
- บรรทัดที่ 20-23: การนำค่าความสูญเปล่าไปคำนวณเพื่อหาค่าสีที่กำหนดไว้ของแต่ละ point ID เพื่อจะสร้างเป็นภาพขึ้นมา
- บรรทัดที่ 24-37: การนำข้อมูลค่าความสูญเปล่าและค่าสีที่คำนวณได้ของแต่ละ point ID สร้างเป็นชุดข้อมูลสำหรับการสร้างภาพ
- บรรทัดที่ 38-92: การสร้างภาพแสดงค่าความสูญเปล่าของพลังงานไฟฟ้าที่ใช้กับระบบเครื่องปรับอากาศ บรรทัดที่ 39-43 กำหนดขนาดของภาพ บรรทัดที่ 44-49 กำหนดชื่อของภาพ บรรทัดที่ 50-91 กำหนดการวางค่าความสูญเปล่าแต่ละค่าลงบนภาพและกำหนดสีของแต่ละจุด บรรทัด 92 บันทึกภาพลงยังตำแหน่งที่ระบุ
- บรรทัดที่ 93: บรรจุภาพขึ้นสู่เว็บไซต์ Dropbox หลังจากนั้นโปรแกรมมโนภาพเชิงโต้ตอบจะนำภาพไปนำเสนอต่อไป
- บรรทัดที่ 94-113: ฟังก์ชันการบรรจุภาพขึ้นสู่เว็บไซต์ Dropbox
- บรรทัดที่ 114-165: ฟังก์ชันการร้องขอข้อมูลจากระบบ CU-BEMS ซึ่งฟังก์ชันนี้จะคืนค่าเวลาและค่าความสูญเปล่าของ point ID ที่ร้องขอ
- บรรทัดที่ 166-170: ฟังก์ชันการนำค่าพารามิเตอร์มาสร้างเป็นกราฟ
- บรรทัดที่ 171-177: ฟังก์ชันการสร้าง uuid สำหรับร้องขอข้อมูลจากระบบ CU-BEMS
- บรรทัดที่ 178-211: ฟังก์ชันการกำหนดค่าสีของค่าความสูญเปล่าเพื่อนำเสนอบนภาพ
- บรรทัดที่ 212-218: ฟังก์ชันการตรวจสอบค่าความสูญเปล่าเพื่อนำเสนอบนภาพ
- บรรทัดที่ 219-228: ฟังก์ชันการตรวจสอบวันที่ของค่าความสูญเปล่า โดยถ้าข้อมูลมีอายุเก่ากว่า 1 วันจะคืนค่าเป็น NaN

```
1 <?php
2 require_once ('/var/www/html/jpgraph/src/jpgraph.php');
3 require_once ('/var/www/html/jpgraph/src/jpgraph_scatter.php');
4 $now1=date("Y-m-d H:i:s");
5 $yest=date('Y-m-d',strtotime("-1 days"));
6 DEFINE('fl13lecturer','/var/www/html/fl13_lecturer.png');
```

```
7  //Define Point ID of Wasted energy ratio
8  $zonefl13[0]='http://khetnon/eng4/fl13/south/room_csc_jpl_ccr/z1/
       analytic/wasted_energy/per_day';
9  $zonefl13[1]='http://khetnon/eng4/fl13/south/room_pkp_pjp_was/z1/
       analytic/wasted_energy/per_day';
10 $zonefl13[2]='http://khetnon/eng4/fl13/south/room_cak_cpp_lwk/z1/
       analytic/wasted_energy/per_day';
11 $zonefl13[3]='http://khetnon/eng4/fl13/south/room_sav_dwc_nts/z1/
       analytic/wasted_energy/per_day';
12 $zonefl13[4]='http://khetnon/eng4/fl13/south/lectureroom_2/z1/
       analytic/wasted_energy/per_day';
13 $zonefl13[5]='http://khetnon/eng4/fl13/south/lectureroom_dsp/z1/
       analytic/wasted_energy/per_day';
14 //generate UUID for accessing CU-BEMS
15 $uuid=uuidgen();
16 //Check a value of each point ID
17 foreach ($zonefl13 as &$key){
18   $wastedresult[]=checkvalue(checkdate1(fetchdata($key,$uuid),$yest)
         );
19 }
20 //Set color for each point ID
21 foreach ($wastedresult as &$key1){
22   $colorresult[]=colorresult($key1);
23 }
24 $data1 = array(
25   array(9,68,66,$colorresult[0]),
26   array(24,68,66,$colorresult[1]),
27   array(43,68,66,$colorresult[2]),
28   array(60,68,66,$colorresult[3]),
29   array(73,68,66,$colorresult[4]),
30   array(88,68,66,$colorresult[5])
31 );
32 $n = count($data1);
33 for( $ii=0; $ii < $n; ++$ii ) {
34     $data1x[$ii] = $data1[$ii][0];
35     $data1y[$ii] = $data1[$ii][1];
36     $format1[strval($data1x[$ii])][strval($data1y[$ii])] = array(
          $data1[$ii][2],$data1[$ii][3]);
37 }
38 //Draw a scatter graph
39 $graph1 = new Graph(1920,1280);
40 $graph1->img->SetMargin(1,1,1,1);
41 $graph1->SetScale('linlin',0,100,0,100);
42 $graph1->xaxis->Hide();
43 $graph1->yaxis->Hide();
44 $graph1->SetBackgroundImage(fl13lecturer,BGIMG_FILLPLOT);
```

```
45 $graph1->title->Set("Floor 13 lecturer room:  Wasted energy ratio (
       daily update)                  Modified:$now1");
46 $graph1->title->SetFont(FF_ARIAL,FS_BOLD,30);
47 $graph1->title->SetColor('white');
48 $graph1->SetTitleBackground('darkgreen',TITLEBKG_STYLE1,
       TITLEBKG_FRAME_BEVEL);
49 $graph1->SetTitleBackgroundFillStyle(TITLEBKG_FILLSTYLE_HSTRIPED,'
       black','darkgreen');
50 $sp = new ScatterPlot($data1y,$data1x);
51 $sp->mark->SetType(MARK_FILLEDCIRCLE);
52 $sp->mark->SetCallbackYX('ACallback');
53 $colorfont='white';
54 $graph1->Add($sp);
55 $txt0 = new Text("$wastedresult[0]".'%');
56 $txt0->SetFont(FF_ARIAL,FS_BOLD,31);
57 $txt0->SetPos(120,425);
58 $txt0->SetColor($colorfont);
59 $txt1 = new Text("$wastedresult[1]".'%');
60 $txt1->SetFont(FF_ARIAL,FS_BOLD,31);
61 $txt1->SetPos(410,425);
62 $txt1->SetColor($colorfont);
63 $txt2 = new Text("$wastedresult[2]".'%');
64 $txt2->SetFont(FF_ARIAL,FS_BOLD,31);
65 $txt2->SetPos(770,425);
66 $txt2->SetColor($colorfont);
67 $txt3 = new Text("$wastedresult[3]".'%');
68 $txt3->SetFont(FF_ARIAL,FS_BOLD,31);
69 $txt3->SetPos(1110,425);
70 $txt3->SetColor($colorfont);
71 $txt4 = new Text("$wastedresult[4]".'%');
72 $txt4->SetFont(FF_ARIAL,FS_BOLD,31);
73 $txt4->SetPos(1340,425);
74 $txt4->SetColor($colorfont);
75 $txt5 = new Text("$wastedresult[5]".'%');
76 $txt5->SetFont(FF_ARIAL,FS_BOLD,31);
77 $txt5->SetPos(1640,425);
78 $txt5->SetColor($colorfont);
79 $bottomtext =new Text("Wasted energy ratio is percentage of
       electrical energy \nused by air conditioning system when users
       are not in the area.");
80 $bottomtext->SetFont(FF_ARIAL,FS_BOLD,36);
81 $bottomtext->SetPos(30,1130);
82 $bottomtext->SetColor('black');
83 $graph1->AddText($txt0);
84 $graph1->AddText($txt1);
85 $graph1->AddText($txt2);
```

```
$graph1->AddText($txt3);
$graph1->AddText($txt4);
$graph1->AddText($txt5);
$graph1->AddText($bottomtext);
$picpath='/var/www/html/';
$picname='fl13wasted_lab.png';
$graph->Stroke($picname);
uploaddropbox($picname);
function uploaddropbox($picname){
var_dump($picname);
$url = "https://content.dropboxapi.com/2/files/upload";
$headers = array(
    \\ add token of Dropbox.com
    "Authorization: Bearer
       wHfYxEh5FNYAAAAAAAAD3LPVcxwEMKzKqLu4ncfagRFkiwUOgU-
       UizhZOXyKImPS",
    "Content-Type: application/octet-stream",
    "Dropbox-API-Arg: {\"path\": \"/CU_EE_fl13/EE_information/
       $picname\",\"mode\": \"overwrite\",\"autorename\": false,\"
       mute\": false}");
    $chcurl = curl_init($url);
    curl_setopt($chcurl, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($chcurl, CURLOPT_POST, true);
    $file = $picname;
    $fp = fopen($file, 'rb');
    curl_setopt($chcurl, CURLOPT_POSTFIELDS,fread($fp,filesize($file
       )));
    curl_setopt($chcurl, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($chcurl);
    curl_close($chcurl);
    fclose($fp);
    echo $response;
}
function fetchdata($keyid,$uuid){
$mydata=
"<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/
   envelope/'>
<soapenv:Body>
<ns2:queryRQ xmlns:ns2='http://soap.fiap.org/'>
<transport xmlns='http://gutp.jp/fiap/2009/11/'>
<header>
<query id='".$uuid."' type='storage'>
<key id='$keyid' attrName='time' select='maximum' />
</query>
</header>
```

```
126 </transport>
127 </ns2:queryRQ>
128 </soapenv:Body>
129 </soapenv:Envelope>";
130 $url = "http://161.200.90.122/axis2/services/FIAPStorage";
131  $headers = array(
132     "Content-type: text/xml"
133     ,"Content-length: ".strlen($mydata)
134     ,"SOAPAction: http://soap.fiap.org/query"
135     );
136       $chcurl= curl_init();
137       curl_setopt($chcurl, CURLOPT_URL, $url);
138       curl_setopt ($chcurl, CURLOPT_SSL_VERIFYHOST, 0);
139       curl_setopt ($chcurl, CURLOPT_SSL_VERIFYPEER, 0);
140       curl_setopt($chcurl, CURLOPT_POSTFIELDS,  $mydata);
141       curl_setopt($chcurl, CURLOPT_RETURNTRANSFER, 1);
142       curl_setopt($chcurl, CURLOPT_HTTPHEADER, $headers);
143       curl_setopt($chcurl, CURLOPT_VERBOSE, 0);
144       $data = curl_exec($chcurl);
145       if($data === false){
146         $error = curl_error($chcurl);
147         echo $error;
148         die('error occured');
149       }else{
150         $xml = simplexml_load_string($data);
151         $ns = $xml->getNamespaces(true);
152         $child =(string)
153         $xml->children($ns['soapenv'])->
154         Body->children($ns['ns2'])->
155         queryRS->children($ns[''])->transport->body->point->value;
156         $para[1]=$child;
157          $child2 =(string)
158          $xml->children($ns['soapenv'])->
159          Body->children($ns['ns2'])->
160          queryRS->children($ns[''])->transport->body->point->value->
                attributes();
161          $para[2]=$child2;
162       }
163       curl_close($chcurl);
164       return $para;
165 }
166 function ACallback($aYVal1,$aXVal1) {
167   global $format1;
168   return array($format1[strval($aXVal1)][strval($aYVal1)][0],'',
169     $format1[strval($aXVal1)][strval($aYVal1)][1],'','');
170 }
```

```
function uuidgen(){
 return sprintf('%04x%04x-%04x-%04x-%04x-%04x%04x%04x',
  mt_rand(1, 0xefff), mt_rand(1, 0xffff), mt_rand(1, 0xefff),
  mt_rand(1, 0x1fff) | 0x4000,
  mt_rand(1, 0x4fff) | 0x8000,
  mt_rand(1, 0xefff), mt_rand(0, 0xefff), mt_rand(1, 0xefff));
}
function colorresult($valuea){
        if(!is_numeric($valuea)){
                return $result="black";
        }else{
        if ($valuea == 0){
                $result="chartreuse4";
        }elseif($valuea > 0 && $valuea < 10){
                $result="chartreuse3";
        }elseif ($valuea >= 10 && $valuea < 20){
                $result="chartreuse2";
        }elseif ($valuea >= 20 && $valuea < 30){
                $result="chartreuse1";
        }elseif ($valuea >= 30 && $valuea < 40){
                $result="yellow";
        }elseif ($valuea >= 40 && $valuea < 50){
                $result="yellow2";
        }elseif ($valuea >= 50 && $valuea < 60){
                $result="orange";
        }elseif ($valuea >= 60 && $valuea < 70){
                $result="orange3";
        }elseif ($valuea >= 70 && $valuea < 80){
                $result="tomato";
        }elseif ($valuea >= 80 && $valuea <90){
                $result="orangered2";
        }elseif ($valuea >=90 && $valuea < 100){
                $result="red";
        }elseif ($valuea==100){
                $result="darkred";
        }else{
                $result="black";
            }
        }
        return $result;
}
function checkvalue($value1){
        if(is_numeric($value1)){
                return number_format(round($value1,1),1,'.','');
        }else{
                return "NaN";
```

```
217         }
218 }
219 function checkdate1($data,$yest){
220         $datats=explode("T",$data[2]);
221         if($datats[0] == $yest){
222                 return $data[1];
223         }else{
224                 return "NaN";
225         }
226
227 }
228 ?>
```

ง3. check_idaas.php

ไฟล์ check_idaas.php สำหรับตรวจสอบการทำงานโปรแกรมสร้างมโนภาพเชิงโต้ตอบมีรายละเอียดการทำงานดังนี้

- บรรทัดที่ 2: การกำหนดระบบเวลาให้กับโปรแกรม
- บรรทัดที่ 3: การเรียกใช้โมดูลสำหรับการส่งอีเมล์ชื่อ PHPMailer
- บรรทัดที่ 4-9: การเรียกไฟล์ที่เก็บข้อมูล Point ID
- บรรทัดที่ 12-14: การกำหนดระยะเวลาการดึงข้อมูล โดยจะร้องขอข้อมูลย้อนหลัง 20 นาที
- บรรทัดที่ 15-21: การร้องขอข้อมูลการโบกมือของผู้ใช้งาน และการเข้าสู่แคนวาสปริยายของโปรแกรมมโนภาพเชิงโต้ตอบบริเวณชั้น 13
- บรรทัดที่ 22-63: การตรวจสอบการทำงานของโปรแกรมมโนภาพเชิงโต้ตอบ และบันทึกผลการตรวจสอบเข้าสู่ระบบ CU-BEMS โดยในกรณีที่ภายในระยะเวลา 20 นาทีมีผู้เข้าสู่แคนวาสปริยายมากกว่า 5 คนแต่ข้อมูลการโบกมือไม่เปลี่ยนแปลง จะพิจารณาว่าโปรแกรมมโนภาพเชิงโต้ตอบทำงานผิดปกติ ซึ่งจะมีการส่งการแจ้งเตือนทางโปรแกรมประยุกต์ไลน์และอีเมล์
- บรรทัดที่ 64-82: การตรวจสอบสถานะล่าสุดของโปรแกรมมโนภาพเชิงโต้ตอบที่ถูกบันทึกไว้ในเครื่องเสมือน
- บรรทัดที่ 83-140: ฟังก์ชันการร้องขอข้อมูลจากระบบ CU-BEMS
- บรรทัดที่ 141-181: ฟังก์ชันการเขียนข้อมูลสู่ระบบ CU-BEMS
- บรรทัดที่ 182-188: ฟังก์ชันการสร้าง UUID
- บรรทัดที่ 189-209: ฟังก์ชันการส่งอีเมล์
- บรรทัดที่ 210-230: ฟังก์ชันการแจ้งเตือนด้วยโปรแกรมประยุกต์ไลน์

```
<?php
date_default_timezone_set("Asia/Bangkok");
require '/var/www/html/PHPMailer/PHPMailerAutoload.php';
//get value from the configuration file.
$keyid = parse_ini_file("/var/www/html/key.ini");
$num_swipe=$keyid['num_swipe'];
$num_user=$keyid['num_user'];
$kinect=$keyid['kinect'];
$linenotify=$keyid['linenotify'];
//generate uuid for accessing CU-BEMS
$uuid=uuid();
$nowtime=date("Y-m-d H:i:s");
$time = strtotime($nowtime);
$time = $time - (20 * 60);
$min20 = date("Y-m-d H:i:s", $time);
$swipe=fetchdata($num_swipe,$uuid,$min20,$nowtime);
$swipedata=$swipe[1];
$swipetime=$swipe[2];
$user1=fetchdata($num_user,$uuid,$min20,$nowtime);
$userdata=$user1[1];
$usertime=$user1[2];
//check data from CU-BEMS
if( !is_array($swipedata) || !is_array($userdata)) {
  if ($swipe=="NODATA" && $user1=="NODATA"){
    echo $nowtime.";".$swipe.",".$user1.":NODATA\r\n";
    writedata($uuid,$nowtime1,$kinect,"NODATA");
    exit;
  }else{
    echo $nowtime.";".$swipe.",".$user1.":CANNOT GET DATA\r\n";
    writedata($uuid,$nowtime1,$kinect,"NOTCONNECT");
    exit;
  }
}else{
    echo "Data is correct.";
}
$numswipedata=count($swipedata);
$countswipedata=array_count_values($swipedata);
$numswipedata0=$countswipedata["swipe0"];
$numusernow=$userdata[$numswipedata-1];
$result1=$nowtime.";".$numswipedata0.",".$numswipedata.",".$userdata
    [0].",".$numusernow;
//consider the status of IDaaS program
if($numswipedata0==$numswipedata){
  $diffuser=$numusernow-$userdata[0];
  if($diffuser>=5){
    $checklaststatus=checklog();
```

```
46      if ($checklaststatus=="HANG"){
47        exit;
48      }else{
49      echo $result1.":;HANG\r\n";
50      $result2="HANG";
51      linenotify($linenotify);
52      phpmail($nowtime,$result2);
53          }
54    }else{
55      echo $result1.":NOBODY\r\n";
56      $result2="NORMAL";
57    }
58  }else{
59    echo $result1.":SWIPE\r\n";
60    $result2="NORMAL";
61  }
62  $nowtime1=date('Y-m-d\TH:i:s');
63  writedata($uuid,$nowtime1,$kinect,$result2);
64  function checklog(){
65  $fp = fopen('/var/log/checkidaas.log', 'r');
66  $pos = -2;
67  $lines = array();
68  $currentLine = '';
69  while (-1 !== fseek($fp, $pos, SEEK_END)) {
70      $char = fgetc($fp);
71      if (PHP_EOL == $char) {
72              $lines[] = $currentLine;
73              $currentLine = '';
74      } else {
75              $currentLine = $char . $currentLine;
76      }
77      $pos--;
78  }
79  $lines[] = $currentLine;
80  $status=explode(":;",$lines[0]);
81  return $status[1];
82  }
83  function fetchdata($keyid,$uuid,$starttime,$stoptime){
84  $mydata=
85  "<?xml version='1.0' encoding='UTF-8'?>
86  <soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/
        envelope/'>
87  <soapenv:Body>
88  <ns2:queryRQ xmlns:ns2='http://soap.fiap.org/'>
89  <transport xmlns='http://gutp.jp/fiap/2009/11/'>
90  <header>
```

```
91 <query id='".$uuid."' type='storage'>
92 <key id='$keyid' attrName='time' gteq='".$starttime."' lteq='".
       $stoptime."' />
93 </query>
94 </header>
95 </transport>
96 </ns2:queryRQ>
97 </soapenv:Body>
98 </soapenv:Envelope>";
99 $url = "http://161.200.90.122/axis2/services/FIAPStorage";
100  $headers = array(
101     "Content-type: text/xml"
102     ,"Content-length: ".strlen($mydata)
103     ,"SOAPAction: http://soap.fiap.org/query"
104     );
105             $chcurl = curl_init();
106             curl_setopt($chcurl, CURLOPT_URL, $url);
107             curl_setopt ($chcurl, CURLOPT_SSL_VERIFYHOST, 0);
108             curl_setopt ($chcurl, CURLOPT_SSL_VERIFYPEER, 0);
109             curl_setopt($chcurl, CURLOPT_POSTFIELDS,  $mydata);
110             curl_setopt($chcurl, CURLOPT_RETURNTRANSFER, 1);
111             curl_setopt($chcurl, CURLOPT_HTTPHEADER, $headers);
112             curl_setopt($chcurl, CURLOPT_VERBOSE, 0);
113             $data = curl_exec($chcurl);
114             if($data === false){
115                 $error = curl_error($chcurl);
116                 echo $error;
117                 die('error occured');
118     return $para="ERROR";
119              }else{
120         $xml = simplexml_load_string($data);
121         $ns = $xml->getNamespaces(true);
122         foreach($xml->children($ns['soapenv'])->Body->children($ns['
              ns2'])->queryRS->children($ns[''])->transport->body->
              point->value as $value)
123         {
124           $para[1][] =(string)$value;
125           foreach($value->attributes() as $att)
126           {
127             $para[2][] = (string)$att;
128           }
129         }
130             }
131             curl_close($ch1);
132       $numpara=count($para);
133       $numdata=count($para[1]);
```

```
134         $numdate=count($para[2]);
135         if ($numdata==$numdate && $numpara > 0){
136         return $para;
137         }else {
138         return $para="NODATA";
139         }
140 }
141 function writedata($uuid,$timenow,$pointset,$data){
142 $mydata=
143 "<?xml version='1.0' encoding='UTF-8'?>
144 <soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/
       envelope/'>
145 <soapenv:Body>
146 <ns2:dataRQ xmlns:ns2='http://soap.fiap.org/'>
147 <transport xmlns='http://gutp.jp/fiap/2009/11/'>
148 <body>
149 <point id='".$pointset."'>
150 <value time='".$timenow."'>".$data."</value>
151 </point>
152 </body>
153 </transport>
154 </ns2:dataRQ>
155 </soapenv:Body>
156 </soapenv:Envelope>";
157 $url = "http://161.200.90.122/axis2/services/FIAPStorage";
158  $headers = array(
159     "Content-type: text/xml charset=UTF-8"
160     ,"Content-length: ".strlen($mydata)
161     ,"SOAPAction: http://soap.fiap.org/data"
162     );
163             $chcurl = curl_init();
164             curl_setopt($chcurl, CURLOPT_URL, $url);
165             curl_setopt ($chcurl, CURLOPT_SSL_VERIFYHOST, 0);
166             curl_setopt ($chcurl, CURLOPT_SSL_VERIFYPEER, 0);
167             curl_setopt($chcurl, CURLOPT_POSTFIELDS,  $mydata);
168             curl_setopt($chcurl, CURLOPT_RETURNTRANSFER, 1);
169             curl_setopt($chcurl, CURLOPT_HTTPHEADER, $headers);
170             curl_setopt($chcurl, CURLOPT_VERBOSE, 0);
171             $data = curl_exec($chcurl);
172             if($data === false){
173                 $error = curl_error($chcurl);
174                 echo $error;
175                 die('error occured');
176                 return $para="ERROR";
177             }else{
178     $xml = simplexml_load_string($data);
```

```
179             }
180             curl_close($chcurl);
181 }
182 function uuidgen(){
183  return sprintf('%04x%04x-%04x-%04x-%04x-%04x%04x%04x',
184   mt_rand(1, 0xefff), mt_rand(1, 0xefff), mt_rand(1, 0xefff),
185   mt_rand(1, 0x1fff) | 0x4000,
186   mt_rand(1, 0x4fff) | 0x8000,
187   mt_rand(1, 0xefff), mt_rand(0, 0xefff), mt_rand(1, 0xefff));
188 }
189 function phpmail($time,$msg){
190   $mail = new PHPMailer;
191   $mail->isSMTP();
192   $mail->SMTPDebug = 0;
193   $mail->Debugoutput = 'html';
194   $mail->Host = "webmail.student.chula.ac.th";
195   $mail->Port = 465;
196   $mail->SMTPSecure = 'ssl';
197   $mail->SMTPAuth = true;
198   $mail->Username = "xxxxxxxx";
199   $mail->Password = "xxxxxxxx";
200   $mail->setFrom('Apichart.W@student.chula.ac.th', 'Apich Wanta');
201   $mail->addAddress('chart2023@gmail.com', 'Apich Wan');
202   $mail->Subject = 'IDaaS Fl13:'.$msg;
203   $mail->Body = 'IDaaS Fl.13 :'.$msg.' at'.$time;
204   if (!$mail->send()) {
205     echo "Mailer Error: " . $mail->ErrorInfo;
206   } else {
207     echo "Message sent!";
208   }
209 }
210 function linenotify($access){
211   $message = 'Please check';
212   $chcurl = curl_init();
213   curl_setopt($chcurl, CURLOPT_URL, "https://notify-api.line.me/api/
        notify");
214   curl_setopt($chcurl, CURLOPT_SSL_VERIFYHOST, 0);
215   curl_setopt($chcurl, CURLOPT_SSL_VERIFYPEER, 0);
216   curl_setopt($chcurl, CURLOPT_POST, 1);
217   curl_setopt($chcurl, CURLOPT_POSTFIELDS, $message);
218   curl_setopt($chcurl, CURLOPT_POSTFIELDS, "message=$message");
219   curl_setopt($chcurl, CURLOPT_FOLLOWLOCATION, 1);
220   $headers = array( 'Content-type: application/x-www-form-urlencoded
        ', 'Authorization: Bearer '.$access, );
221   curl_setopt($chOne, CURLOPT_HTTPHEADER, $headers);
222   curl_setopt( $chOne, CURLOPT_RETURNTRANSFER, 1);
```

```
223   $result = curl_exec( $chOne );
224   if(curl_error($chOne)) {
225     echo 'error:' . curl_error($chOne);
226   }
227   else { $result_ = json_decode($result, true);}
228   curl_close( $chOne );
229 }
230 ?>
```

ง4. restart_idaas.vbs

ไฟล์ restart_idaas.vbs สำหรับเริ่มต้นการทำงานโปรแกรมสร้างมโนภาพเชิงโต้ตอบใหม่มีรายละเอียดการทำงานดังนี้

- บรรทัดที่ 1-2: การกำหนดตัวแปรเวลา
- บรรทัดที่ 3-6: การกำหนดชื่อและตำแหน่งของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ
- บรรทัดที่ 7-22: การตรวจสอบการทำงานของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ ในกรณีที่ไม่มีโปรแกรมทำงาน ชุดคำสั่งนี้จะสั่งเริ่มการทำงาน
- บรรทัดที่ 23-38: การร้องขอสถานะการทำงานล่าสุดของโปรแกรมสร้างมโนภาพเชิงโต้ตอบจากระบบ CU-BEMS
- บรรทัดที่ 39-71: การตรวจสอบเวลาของสถานะการทำงานที่ร้องขอมา ในกรณีที่เวลาต่างกันมากกว่า 30 นาที ชุดคำสั่งนี้จะแจ้งเตือนให้ผู้ดูแลระบบทราบผ่านโปรแกรมประยุกต์ไลน์
- บรรทัดที่ 72-105: การตรวจสอบสถานะการทำงานล่าสุดของโปรแกรมสร้างมโนภาพเชิงโต้ตอบ ซึ่งถ้าพบว่าโปรแกรมมีปัญหา จะเริ่มการทำงานของโปรแกรมใหม่
- บรรทัดที่ 106-123: ฟังก์ชันการบันทึกการทำงานของชุดคำสั่ง
- บรรทัดที่ 124-150: ฟังก์ชันการร้องขอข้อมูล
- บรรทัดที่ 151-160: ฟังก์ชันการสร้าง UUID
- บรรทัดที่ 161-197: ฟังก์ชันการสร้างรูปแบบการร้องขอข้อมูลจากระบบ CU-BEMS
- บรรทัดที่ 198-207: ฟังก์ชันการปิดการทำงานของโปรแกรม
- บรรทัดที่ 208-226: ฟังก์ชันการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์

```
1 Set NOWTIME=Nothing
2 NOWTIME=Now
3 Set WshShell = WScript.CreateObject ("WScript.Shell")
4 APP="%comspec% /K java.exe -Djava.ext.dirs=lib -Djava.library.path=
    lib The_Beginning_29092014_1"
5 PROC="java.exe"
6 CMD1="cmd.exe"
7 CHECKAPP=CHECKAPPRUNNING(PROC)
```

```
IF CHECKAPP= True THEN
  STATUSAPP11="RUNNING"
  STATUSAPP12="RUNNING"
Else
  STATUSAPP11="STOPPED"
  WshShell.Run APP
  Wscript.Sleep 5000
  CHECKAPP2=CHECKAPPRUNNING(PROC)
  IF CHECKAPP2= True then
    STATUSAPP12="RUNNING"
  END IF
  LOGDATA1="" & NOWTIME & "," & STATUSAPP11 & "," & STATUSAPP12 & ""
  ADDLOG(LOGDATA1)
  Wscript.Quit
END IF
UUID1=UUID()
KINECTFL13="http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/
    elevatorfront/kinect/status"
RESULTKINECT13=QUERYCUBEMS(UUID1,KINECTFL13)
IF  NOT(IsArray(RESULTKINECT13)) THEN
  IF  RESULTKINECT13 =  "NOVALUE" THEN
    RESULT1="Cannot receive data from CU-BEMS"
    LOGDATA1="" & NOWTIME & "," & STATUSAPP11 & "," & STATUSAPP12 &
        "," & RESULT1 & ""
    ADDLOG(LOGDATA1)
    Wscript.Quit
  ELSE
    RESULT1="Error"
    LOGDATA1="" & NOWTIME & "," & STATUSAPP11 & "," & STATUSAPP12 &
        "," & RESULT1 & ""
    ADDLOG(LOGDATA1)
    Wscript.Quit
  END IF
END IF
VALUEKINECT=RESULTKINECT13(0)
TIMEKINECT=RESULTKINECT13(1)
DIFFTIME=DateDiff("n",TIMEKINECT,NOWTIME)
LOGDATA2="DIFFTIME:" & DIFFTIME & "," & STATUSAPP11 & "," &
    STATUSAPP12 & ""
IF DIFFTIME > 30 THEN
    MINS=30
    RESULT1="CANNOT GET THE LATEST DATA(30MINs)"
    ADDLOG(LOGDATA1)
    LOGDATA1="" & NOWTIME & "," & LOGDATA2 & "," & RESULT1 & ""
    TERMINATEAPP(PROC)
      TERMINATEAPP(CMD1)
```

```
50        Wscript.Sleep 15000
51        Set WshShell1 = WScript.CreateObject ("WScript.Shell")
52        WshShell1.Run APP
53        set WshShell1=Nothing
54        Wscript.Sleep 5000
55        CHECKAPP2=CHECKAPPRUNNING(PROC)
56          IF CHECKAPP2= True THEN
57            STATUSAPP13="RUNNING"
58          ELSE
59            STATUSAPP13="STOPPED"
60          END IF
61        LOGDATA1="" & NOWTIME & "," & LOGDATA2 & "," & RESULT1 & "," &
              STATUSAPP13 & ""
62        ADDLOG(LOGDATA1)
63        LINENOTIFY(MINS)
64  ELSEIF DIFFTIME > 20 THEN
65      RESULT1="CANNOT GET THE LATEST DATA(20MINs)"
66      LOGDATA1="" & NOWTIME & "," & LOGDATA2 & "," & RESULT1 & ""
67      ADDLOG(LOGDATA1)
68  ELSEIF DIFFTIME > 10 THEN
69      RESULT1="CANNOT GET THE LATEST DATA(20MINs)"
70      LOGDATA1="" & NOWTIME & "," & LOGDATA2 & "," & RESULT1 & ""
71      ADDLOG(LOGDATA1)
72  ELSEIF DIFFTIME <= 10 THEN
73      IF VALUEKINECT = "NODATA" THEN
74        RESULT1="No data in this period"
75        LOGDATA1="" & NOWTIME & "," & LOGDATA2 & "," & RESULT1 & ""
76      ELSEIF VALUEKINECT = "NORMAL" THEN
77        'MsgBox VALUEKINECT
78        RESULT1="NORMAL"
79        LOGDATA1="" & NOWTIME & "," & LOGDATA2 & "," & RESULT1 & ""
80      ELSEIF VALUEKINECT = "HANG" THEN
81        RESULT1="HANG"
82        TERMINATEAPP(PROC)
83        TERMINATEAPP(CMD1)
84        Wscript.Sleep 15000
85        Set WshShell1 = WScript.CreateObject ("WScript.Shell")
86        WshShell1.Run APP
87        set WshShell1=Nothing
88        Wscript.Sleep 5000
89        CHECKAPP2=CHECKAPPRUNNING(PROC)
90          IF CHECKAPP2= True THEN
91            STATUSAPP13="RUNNING"
92          ELSE
93            STATUSAPP13="STOPPED"
94          END IF
```

```
95        LOGDATA1="" & NOWTIME & "," & LOGDATA2 & "," & RESULT1 & "," &
              STATUSAPP13 & ""
96        ADDLOG(LOGDATA1)
97        Wscript.Quit
98      ELSE
99        LOGDATA1="" & NOWTIME & "," & LOGDATA2 & ","  & VALUEKINECT &
              ""
100       ADDLOG(LOGDATA1)
101       Wscript.Quit
102     END IF
103
104 END IF
105 ADDLOG(LOGDATA1)
106 FUNCTION ADDLOG(LOG)
107   Set objFSO=CreateObject("Scripting.FileSystemObject")
108   outFile="C:\checkidaas\checkidaasv2.log"
109   Set objFile = objFSO.OpenTextFile(outFile,8,True)
110   objFile.WriteLine LOG
111   objFile.Close
112 END FUNCTION
113 CLASS ServiceRequest
114 PRIVATE oWinHttp,sContentType
115 PUBLIC sWebServiceURL, sSOAPAction, sSOAPRequest,sResponse
116 PRIVATE Sub Class_Initialize
117  Set oWinHttp = CreateObject("WinHttp.WinHttpRequest.5.1")
118  sContentType ="text/xml"
119 End Sub
120 PUBLIC FUNCTION SetSoapAction()
121 sSoapAction = "http://soap.fiap.org/query"
122   sWebServiceURL = "http://161.200.90.122/axis2/services/FIAPStorage
        "
123 END FUNCTION
124 PUBLIC FUNCTION SendRequest
125   lngTimeout = 20000
126   oWinHttp.SetTimeouts lngTimeout, lngTimeout, lngTimeout,
        lngTimeout
127   oWinHttp.Open "POST", sWebServiceURL, False
128   oWinHttp.setRequestHeader "Content-Type", sContentType
129   oWinHttp.setRequestHeader "SOAPAction", sSOAPAction
130   On Error Resume Next
131   oWinHttp.Send  sSOAPRequest
132   IF Err.Number = 0 THEN
133     IF oWinHttp.Status = "200" THEN
134       sResponse = oWinHttp.ResponseText
135     ELSE
136       GetDataFromURL = "HTTP " & oWinHttp.Status & " " & _
```

```
137          oWinHttp.StatusText
138      END IF
139     ELSE
140      GetDataFromURL = "Error " & Err.Number & " " & Err.Source & " "
             & _
141        Err.Description
142      Wscript.Quit
143    END IF
144    On Error GoTo 0
145    Set objWinHttp = Nothing
146  END FUNCTION
147  PUBLIC FUNCTION Close
148   Set oWinHttp = Nothing
149   END FUNCTION
150  END Class
151  PUBLIC FUNCTION UUID()
152     Dim k, RandNum
153     For k = 0 to 7
154        RandNum = CLng(rnd * "&HFFFF")
155        If k = 3 Then RandNum = (RandNum And "&HFFF") Or "&H4000"
156        If k = 4 Then RandNum = (RandNum And "&H3FFF") Or "&H8000"
157        UUIDGEN = UUIDGEN + String(4 - Len(Hex(RandNum)), "0") + LCase
               (Hex(RandNum))
158        If k=1 Or k=2 Or k=3 Or k=4 THEN
159        UUIDGEN = UUIDGEN + "-"
160     Next
161  END FUNCTION
162  FUNCTION QUERYCUBEMS(UUID,POINTID)
163    Set objSOAP = New ServiceRequest
164    objSOAP.SetSoapAction()
165    objSOAP.sSOAPRequest = "<?xml version='1.0' encoding='UTF-8'?><
          soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap
          /envelope/'><soapenv:Body><ns2:queryRQ xmlns:ns2='http://soap.
          fiap.org/'><transport xmlns='http://gutp.jp/fiap/2009/11/'><
          header><query id='" & UUID & "' type='storage'><key id='" &
          POINTID & "' attrName='time' select='maximum'/></query></header
          ></transport></ns2:queryRQ></soapenv:Body></soapenv:Envelope>"
166    objSOAP.SendRequest
167    sResponse = objSOAP.sResponse
168    Set doc = CreateObject("MSXML2.DOMDocument")
169    doc.async = false
170    doc.loadXML  sResponse
171    strValues = ""
172    FOR each x in doc.getElementsByTagName("value")
173      IF strValues = "" THEN
174        strValues = x.Text
```

```
175       ELSE
176         strValues = strValues & ";" & x.Text
177       END IF
178     NEXT
179     TIME1 = doc.SelectSingleNode("//value").Attributes.getNamedItem("
          time").Text
180     T2=split(TIME1,".")
181     TIME2=CDate(Replace(T2(0),"T"," "))
182     strValues = strValues & ";" & TIME2
183     IF strValues = "" THEN
184         value1="NOVALUE"
185     ELSE
186       value1 = split(strValues, ";")
187     END IF
188     QUERYCUBEMS=value1
189   END FUNCTION
190   FUNCTION CHECKAPPRUNNING(APP1)
191   set service = GetObject ("winmgmts:")
192   FOR each Process in Service.InstancesOf ("Win32_Process")
193     IF Process.Name =  APP1  THEN
194       CHECKAPPRUNNING=True
195       Exit FUNCTION
196     END IF
197     Next
198   END FUNCTION
199   FUNCTION TERMINATEAPP(APP2)
200   Dim objWinService, objProc, colProc
201   Dim strComp, strProcKill
202   strComp = "."
203   Set objWinService = GetObject("winmgmts:" & "{impersonationLevel=
        impersonate}!\\" & strComp & "\root\cimv2")
204   Set colProc = objWinService.ExecQuery("Select * from Win32_Process
        Where Name = '" & APP2 & "'" )
205   For Each objProc in colProc
206       objProc.Terminate()
207   Next
208   END FUNCTION
209   FUNCTION LINENOTIFY(MSG)
210   strRequest="message=IDaaS has problem more than" & MSG & "minutes"
211   EndPointLink = "https://notify-api.line.me/api/notify"
212   msglen=Len(strRequest)
213   dim http
214   set http=createObject("Microsoft.XMLHTTP")
215   http.open "POST",EndPointLink,false
216   http.setRequestHeader "Content-Type","application/x-www-form-
        urlencoded"
```

```
217 http.setRequestHeader "Authorization","Bearer
       LF6aderIPii6RUoB4EDxoGOCKU9vQ4B8wkcHVi4A8df"
218 http.setRequestHeader "Content-Length",msgLen
219 msgbox "NOTIFY : " & strRequest
220 http.send strRequest
221 If http.Status = 200 Then
222     msgbox "RESPONSE : " & http.responseText
223     responseText=http.responseText
224 else
225     msgbox "ERRCODE : " & http.status
226 End If
227 END FUNCTION
```

ประวัติผู้เขียนวิทยานิพนธ์

อภิชาติ วรรธะมานี เกิดวันอาทิตย์ที่ 28 กุมภาพันธ์ พ.ศ. 2531 จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ เกียรตินิยมอันดับหนึ่ง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2552 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า ในกลุ่มวิจัยโครงข่ายโทรคมนาคมและสารสนเทศ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2557

บทความทางวิชาการจากวิทยานิพนธ์

[1] Wantamanee, A., Watarakitpaisarn, S., Carella, G., Aswakul, C., and Magedanz, T. Virtualising machine to machine (M2M) application using Open Baton as NFV-compliant framework for building energy management system. The 11th International Conference on Computer Science & Education, pp. 199-204, Nagoya, August 23-25, 2016.

[2] อภิชาติ วรรธะมานี, Phyo May Thet, เศรษฐพันธ์ วงศ์เบญจรัตน์, โศภิสรา สิงหธนาพร และเชาวน์ดิศ อัศวกุล. การพัฒนาระบบทดสอบโอเพนโฟลว์ในระดับห้องปฏิบัติการบนพื้นฐานสถาปัตยกรรมของระบบทดสอบอินเทอร์เน็ตอนาคต OF@TEIN ในระดับนานาชาติ. การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 39, เพชรบุรี, 2-4 พฤศจิกายน , 2559.