การเปรียบเทียบประสิทธิภาพของ Node.js กับ JavaScript โดยใช้โปรแกรมตรวจสอบข้อมูลแรงดันน้ำในท่อประปาแบบเวลาจริงของการประปานครหลวงเป็นกรณีศึกษา

วรพจน์ ไชยพรพัฒนา

DPU

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม คณะวิศวกรรมศาสตร์
มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2560

A Comparative Case Study of Metropolitan Waterworks Authority's Real-time Pressure Monitoring Program between Node.js and JavaScript

Worapoch Chaiyapornpattana

DPU

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Department of Computer and Telecommunication Engineering

Faculty of Engineering, Dhurakij Pundit University

2017

ใบรับรองวิทยานิพนธ์

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อวิทยานิพนธ์ การเปรียบเทียบประสิทธิภาพของ Node.js กับ JavaScript โดยใช้โปรแกรมตรวจสอบข้อมูลแรงดันน้ำในท่อประปาแบบวลาจริงของการประปานครหลวงเป็นกรณีศึกษา

เสนอโดย นายวรพจน์ ไชยพรพัฒนา

สาขาวิชา วิศวกรรมคอมพิวเตอร์และโทรคมนาคม

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.เนื่องวงศ์ ทวยเจริญ

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์แล้ว

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์)

.....กรรมการและอาจารย์ที่ปรึกษาวิทยานิพนธ์

(ผู้ช่วยศาสตราจารย์ ดร.เนื่องวงศ์ ทวยเจริญ)

..... กรรมการ

(อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์)

..... กรรมการ

(อาจารย์ ดร.ก่อพร พันธุ์ยิ้ม)

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว

..... คณบดีวิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

(ผู้ช่วยศาสตราจารย์ ดร.ณรงค์เดช กีรติพรานนท์)

วันที่ 24 เดือน พฤ พ.ศ. 2560

DPU

หัวข้อวิทยานิพนธ์	การเปรียบเทียบประสิทธิภาพของ Node.js กับ JavaScript โดยใช้โปรแกรมตรวจสอบข้อมูลแรงดันน้ำในท่อประปาแบบเวลาจริงของการประปานครหลวงเป็นกรณีศึกษา
ชื่อผู้เขียน	วรพจน์ ไชยพรพัฒนา
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.เนื่องวงศ์ ทวยเจริญ
สาขาวิชา	วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา	2560

บทคัดย่อ

เว็บแอปพลิเคชันเพื่อการตรวจสอบระดับแรงดันน้ำในท่อประธานแบบเวลาจริง (Real-time) ของการประปานครหลวง มีการเขียนโปรแกรมเว็บแอปพลิเคชันเป็นภาษาพีเอชพี (PHP) ภาษาเอชทีเอ็มแอล (HTML) และใช้ภาษาจาวาสคริปต์ (JavaScript) ในการเปลี่ยนแปลงค่าแรงดันน้ำให้เป็นปัจจุบัน (Update) ทุกๆ 60 วินาที ซึ่งวิธีนี้ยังมีข้อบกพร่องในเรื่องของความรวดเร็วในการแสดงข้อมูลมีปริมาณการเข้าถึงฐานข้อมูลมาก มีข้อมูลส่งผ่านระหว่างเครื่องคอมพิวเตอร์แม่ข่ายและเครื่องคอมพิวเตอร์ลูกข่ายมาก ซึ่งส่งผลให้การรับรู้ข้อมูลของผู้ปฏิบัติงานล่าช้า อาจเกิดข้อผิดพลาดในการบริหารจัดการแรงดันน้ำได้

จากกรณีดังกล่าวผู้วิจัยได้ทำการศึกษาซ็อคเก็ตด็อทไอโอ (Socket.io) และโนดด็อทเจเอส (Node.js) เพื่อปรับปรุงโปรแกรมตรวจสอบแรงดันน้ำในท่อประธาน เพิ่มความรวดเร็วในการแสดงผลแรงดันน้ำ จากเดิมที่ใช้เวลาในการแสดงผลทุกๆ 60 วินาที ให้เหลือไม่เกิน 5 วินาที ลดปริมาณการเข้าถึงฐานข้อมูล ลดปริมาณข้อมูลส่งผ่านระหว่างเครื่องคอมพิวเตอร์แม่ข่ายและเครื่องคอมพิวเตอร์ลูกข่าย และยังสามารถปรับการทำงานของซีพียู (CPU Usage) โดยใช้ระบบควบคุมพีไอดี (PID Controller) ให้เหมาะสมกับทรัพยากรของเครื่องคอมพิวเตอร์แม่ข่ายอีกด้วย โดยมีค่าต่างจากเป้าหมาย 11.16 เปอร์เซ็นต์

Thematic Paper Title	A Comparative Case Study of Metropolitan Waterworks Authority's Real-time Pressure Monitoring Program between Node.js and JavaScript
Author	Worapoch Chaiyapornpattana
Thematic Paper Advisor	Asst. Prof. Dr. Nuengwong Tuaycharoen
Department	Computer and Telecommunication Engineering
Academic Year	2017

ABSTRACT

The real-time trunk-main pipe pressure monitoring program of Metropolitan Waterworks Authority (MWA) was created by PHP and HTML. The previous program used JavaScript to update pressure data every 60 seconds. There were many flaws such as spending long time to update the data, taking many requests to database, and using many data transfers between server-side devices and client-side devices. These flaws also made the system's user acquires the data too late and they might affect to their pressure management and trouble shooting.

From the problems, we study the property of "Socket.io" and "Node.js" to improve the program because these languages can reduce updating time from 60 seconds to less than 5 seconds, reduce the requests to database, reduce data access transferring between server-side devices and client-side devices, and the program can automatically control server's CPU-usage by using PID controller algorithm. The difference between the server's actual CPU-usage and the set point is 11.16 percent.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่สามารถจะนำมากล่าวได้ทั้งหมด ซึ่งผู้มีพระคุณท่านแรกที่ผู้จัดทำใคร่ขอกราบขอบพระคุณ คือ ผศ.ดร.เนื่องวงศ์ ทวยเจริญ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ให้ความรู้และแนะนำแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ทุกขั้นตอน กราบขอบพระคุณ ดร.ชัยพร เขมะภาตะพันธ์ และ ดร.ณรงค์เดช กีรติพรานนท์ ตลอดจนอาจารย์และเจ้าหน้าที่ทุกท่านของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์ ที่ให้คำแนะนำและช่วยเหลืออย่างดีเสมอมา

ขอขอบคุณการประปานครหลวงที่ได้สนับสนุนทุนการศึกษา ให้ข้อมูลและทรัพยากรเพื่อใช้ในการศึกษาจนลุล่วงไปได้ด้วยดี

ท้ายที่สุดนี้ ขอกราบขอบคุณพระคุณบิดา มารดา ผู้อยู่เบื้องหลังความสำเร็จ ที่ได้ให้ความช่วยเหลือ ให้กำลังใจ และสนับสนุนในการศึกษาตลอดมา คุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ทั้งหมด ขอมอบให้แด่บุพการีทั้งสองท่านและครูบาอาจารย์ผู้มีพระคุณทุกท่าน

วรพจน์ ไชยพรพัฒนา

DPU

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ฆ

บทคัดย่อภาษาอังกฤษ..... ง

กิตติกรรมประกาศ..... จ

สารบัญตาราง..... ฌ

สารบัญภาพ..... ญ

บทที่

1. บทนำ..... 1

1.1 ที่มาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของงานวิจัย..... 3

1.3 ขอบเขตของงานวิจัย..... 3

1.4 ประโยชน์ที่คาดว่าจะได้รับ..... 4

1.5 อุปกรณ์และเครื่องมือที่ใช้..... 5

1.6 แผนการดำเนินงาน..... 5

1.7 ความรู้ที่ได้จากการวิจัย..... 6

1.8 ผลงานตีพิมพ์..... 6

2. แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง..... 7

2.1 ระบบตรวจสอบแรงดันน้ำในท่อประธานของการประปานครหลวง..... 7

2.2 ระบบโทรมาตร..... 8

2.3 หน่วยควบคุมระยะไกล..... 9

2.4 ระบบจัดการฐานข้อมูลมายเอสคิวแอล..... 13

2.5 การเขียนโปรแกรมภาษาฝั่งแม่ข่าย..... 14

2.6 โครงสร้างภาษาฝั่งลูกข่าย..... 14

2.7 จาวาสคริปต์..... 15

2.8 เจคิวรี่..... 15

2.9 โนดดอทเจเอส..... 16

2.10 ซ็อกเก็ตด็อทไอโอ..... 18

2.11 งานวิจัยที่เกี่ยวข้อง..... 18

DPU

สารบัญ (ต่อ)

บทที่		หน้า
3.	ระเบียบวิธีวิจัย	22
	3.1 สถาปัตยกรรมฮาร์ดแวร์	22
	3.2 แผนการดำเนินงาน	23
	3.3 การออกแบบและพัฒนาระบบ	26
	3.4 แผนภาพแสดงขั้นตอนการทำงานของระบบ	49
	3.5 แผนภาพยูสเคส	50
	3.6 แผนภาพคลาส	56
	3.7 คำอธิบายตัวแปรและฟังก์ชัน	57
4.	การทดลองและผลการวิจัย	59
	4.1 การเปรียบเทียบการทำงานของซีพียูระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ทางด้านฝั่งเครื่องคอมพิวเตอร์แม่ข่าย	59
	4.2 การเปรียบเทียบการทำงานของซีพียูระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ทางด้านฝั่งเครื่องคอมพิวเตอร์ลูกข่าย	61
	4.3 การเปรียบเทียบปริมาณการร้องขอข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์จากทางฝั่งคอมพิวเตอร์ลูกข่าย	62
	4.4 การเปรียบเทียบปริมาณการส่งผ่านข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ ผ่านทางเว็บเบราว์เซอร์	63
	4.5 การเปรียบเทียบค่าเวลาในการตอบสนองของข้อมูลเปรียบเทียบระหว่างจาวาสคริปต์และโนดด็อทเจเอส	64
	4.6 ความสัมพันธ์ระหว่างการทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่าย ปริมาณการเชื่อมต่อจากผู้ใช้งานหน่วยความจำและการส่งข้อมูลภายในเครือข่าย	64
	4.7 การเขียนโปรแกรมเพื่อควบคุมการทำงานของซีพียูของเครื่องแม่ข่ายแบบอัตโนมัติ	67
5.	สรุปผลและข้อเสนอแนะ	70
	5.1 ข้อได้เปรียบของการใช้การเขียนโปรแกรมภาษาโนดด็อทเจเอสแทนภาษาจาวาสคริปต์	71

สารบัญ (ต่อ)

บทที่ **หน้า**

5.2 ข้อเสียเปรียบของการใช้การเขียนโปรแกรมภาษาโนดด็อทเจเอสแทนภาษาจาวาสคริปต์..... 72

5.3 สรุปผล..... 72

5.4 ข้อเสนอแนะ..... 72

บรรณานุกรม..... 73

ประวัติผู้เขียน..... 75

DPU

สารบัญตาราง

ตารางที่ **หน้า**

3.1 แผนการดำเนินงาน..... 26

3.2 ตารางเก็บข้อมูลเปรียบเทียบการทำงานของจาวาสคริปต์และ
โนดด็อทเจเอสฝั่งแม่ข่าย..... 41

3.3 ตารางเก็บข้อมูลเปรียบเทียบการทำงานของจาวาสคริปต์และ
โนดด็อทเจเอสฝั่งแม่ข่าย..... 41

3.4 ตารางเก็บข้อมูลเปรียบเทียบการส่งผ่านข้อมูลของจาวาสคริปต์
และโนดด็อทเจเอสที่เว็บเบราว์เซอร์..... 43

3.5 ตารางเก็บข้อมูลเปรียบเทียบการร้องขอข้อมูลของจาวาสคริปต์
และโนดด็อทเจเอสจากฝั่งลูกข่าย..... 44

3.6 ตารางเก็บข้อมูลเปรียบเทียบการทำงานระหว่างซีพียูของ
เครื่องคอมพิวเตอร์แม่ข่าย และจำนวนการเชื่อมต่อจากผู้ใช้งาน..... 46

3.7 ข้อกำหนดซอฟท์แวร์ส่วนตรวจสอบค่าแรงดันน้ำ..... 52

3.8 ข้อกำหนดซอฟท์แวร์ส่วนตรวจสอบความผิดปกติของระบบ..... 53

3.9 ข้อกำหนดซอฟท์แวร์ส่วนตั้งค่าใช้งานของระบบ..... 55

3.10 คำอธิบายตัวแปรและฟังก์ชัน..... 57

3.11 คำอธิบายตัวแปรและฟังก์ชัน..... 58

4.1 การทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่ายที่คาบเวลาต่างๆ..... 60

4.2 การทำงานของซีพียูของเครื่องคอมพิวเตอร์ลูกข่ายที่คาบเวลาต่างๆ..... 61

สารบัญภาพ

ภาพที่		หน้า
1.1	ภาพรวมของระบบการรับส่งข้อมูลแรงดันน้ำ	2
2.1	ตัวอย่างกราฟความสัมพันธ์ระหว่างแรงดันน้ำและเวลาตลอด 24 ชั่วโมง	7
2.2	แสดงส่วนประกอบหลักของระบบ SCADA	8
2.3	การเชื่อมต่อระหว่างหน่วยควบคุมระยะไกลกับเครื่องมือวัดภาคสนาม	11
2.4	ภาพรวมของระบบสกาด้าเมื่อนำมาใช้กับโปรแกรมตรวจสอบแรงดันน้ำ	12
2.5	หน้าจอระบบสกาด้าที่ศูนย์ควบคุมหรือ สถานีหลัก (Master Station) และฐานข้อมูล	13
2.6	โครงสร้างการทำงานของแม่ข่ายโนดด็อทเจเอส (Node.js Server)	17
2.7	โครงสร้างการทำงานแบบแม่ข่ายมัลติเทร็ด (Multi-Threaded Server)	17
2.8	ภาพรวมโครงสร้างการทำงานของระบบโครวด์ฟิล (Crowd Fill)	19
2.9	ภาพรวมโครงสร้างการทำงานของระบบคลาวด์เบราเซอร์ (Cloud Browser)	20
3.1	ภาพรวมโครงสร้างสถาปัตยกรรมฮาร์ดแวร์ของวิจัย	22
3.2	การเขียนโปรแกรมภาษาโนดด็อทเจเอสแทนภาษาจาวาสคริปต์	23
3.3	หลักการทำงานของภาษาจาวาสคริปต์	27
3.4	แนวคิดการทำการโดยใช้การเขียนโปรแกรมภาษาโนดด็อทเจเอส	28
3.5	หลักการทำงานของการเขียนโปรแกรมภาษาโนดด็อทเจเอส	29
3.6	ส่วนของการเขียนโปรแกรมหลักของ การเขียนโปรแกรมภาษาโนดด็อทเจเอส	30
3.7	การเรียกข้อมูลจากฐานข้อมูลตามคาบเวลา (Polling Loop) ของโนดด็อทเจเอส	30
3.8	การรับค่าตัวแปรของการเขียนโปรแกรมฝั่งลูกข่าย	31
3.9	คำสั่งการติดตั้งส่วนขยายเอ็กซ์เพลสที่ Command Prompt	31
3.10	สคริปต์ส่วนติดต่อกับฐานข้อมูลมายเอสคิวแอล	32
3.11	สคริปต์ส่วนกำหนดคาบเวลาในการรับ-ส่งข้อมูล	33
3.12	สคริปต์คำสั่งเรียกดูข้อมูลจากฐานข้อมูล	33
3.13	สคริปต์การตรวจสอบผลการเชื่อมต่อฐานข้อมูลและสร้างซ็อคเก็ต	34

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.14 สคริปต์การกำหนดซ็อคเก็ตเพื่อส่งข้อมูลไปยังผู้ใช้งาน..... 35
3.15 การอัพเดท (Update) ซ็อคเก็ต เพื่อส่งข้อมูลไปยังผู้ใช้งานอย่างต่อเนื่อง..... 35
3.16 การกำหนดพอร์ตรับ-ส่งข้อมูลระหว่างฝั่งแม่ข่ายและฝั่งลูกข่าย..... 35
3.17 การเรียกใช้ไฟล์ที่จำเป็นของโนดด็อทเจเอส..... 36
3.18 การกำหนดพอร์ต (Port) เพื่อเตรียมการเชื่อมต่อกับเว็บซ็อคเก็ต..... 36
3.19 การกำหนดตัวแปรที่จำเป็นในการรับค่าจากซ็อคเก็ต..... 37
3.20 การสร้างลูป รับค่าที่ได้จากซ็อคเก็ตเพื่อเก็บไว้ในตัวแปร..... 37
3.21 ส่วนการเขียนภาษาเอชทีเอ็มแอล..... 38
3.22 ส่วนการเขียนภาษาจาวาสคริปต์..... 38
3.23 แผนภาพแสดงการติดตั้งฮาร์ดแวร์ที่ใช้ในแบบจำลอง..... 39
3.24 เว็บเบราว์เซอร์ฝั่งผู้ใช้งาน แสดงผลข้อมูลที่รับจากเว็บซ็อคเก็ต..... 40
3.25 การใช้งานแท็บ (Tab) ของเบราว์เซอร์เพื่อจำลอง
การเชื่อมต่อจากหลายผู้ใช้งาน..... 40
3.26 การเก็บข้อมูลซีพียูจากโปรแกรมรีซอสมอนิเตอร์ (Resource Monitor)..... 42
3.27 การเก็บค่าการส่งผ่านข้อมูลภายในเว็บเบราว์เซอร์โดยกูเกิลโครม..... 42
3.28 การเก็บค่าการร้องขอข้อมูล (Request) จากฝั่งลูกข่ายโดยเว็บเบราว์เซอร์..... 44
3.29 การเก็บค่าเวลาการตอบสนองของข้อมูล..... 45
3.30 ระบบเครือข่าย LAN ที่ใช้ในการทดลอง..... 46
3.31 สคริปต์เอ็นพีเอ็มที่ใช้ในการตรวจสอบการใช้งานซีพียู..... 47
3.32 รหัสเทียมของระบบควบคุมพีไอดี..... 48
3.33 การปรับคาบเวลาในการใช้งานซีพียูโดยใช้ระบบควบคุม
แบบสัดส่วน-ปริพันธ์-อนุพันธ์..... 49
3.34 ผังงานแสดงขั้นตอนการทำงานของระบบ..... 50
3.35 แผนภาพผู้ใช้งาน (Use Case Diagram) ส่วนของผู้ใช้งานและผู้ดูแล..... 50
3.36 แผนภาพลำดับขั้นตอนของผู้ใช้งานส่วนตรวจสอบค่าแรงดันน้ำ..... 51

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.37 แผนภาพลำดับขั้นตอนของผู้ดูแลระบบในส่วน
ตรวจสอบความผิดปกติของระบบ..... 53
3.38 แผนภาพลำดับขั้นตอนของผู้ดูแลระบบในส่วนตั้งค่าใช้งานของระบบ..... 54
3.39 แผนภาพคลาส..... 56
4.1 กราฟเปรียบเทียบการทำงานของซีพียูของเครื่องแม่ข่าย..... 60
4.2 กราฟเปรียบเทียบการทำงานของซีพียูของเครื่องลูกข่าย..... 61
4.3 กราฟเปรียบเทียบปริมาณการร้องขอ (Request)
ของโปรแกรมทั้งสองจากฝั่งลูกข่าย..... 62
4.4 กราฟเปรียบเทียบการส่งผ่านข้อมูล (Data Transfer) ของรูปแบบทั้งสอง..... 63
4.5 กราฟเปรียบเทียบเวลาในการตอบสนองของข้อมูล
ระหว่างจาวาสคริปต์และโนดด็อทเจเอส..... 64
4.6 ข้อมูลทรัพยากรทางคอมพิวเตอร์เมื่อมีการเชื่อมต่อที่มากขึ้น..... 65
4.7 ข้อมูลทรัพยากรทางคอมพิวเตอร์เมื่อมีการเชื่อมต่อ
ที่มากขึ้นจากคอมพิวเตอร์อีกเครื่องหนึ่ง..... 66
4.8 การทำงานของซีพียู แบบการปรับคาบเวลาแบบค่าคงที่..... 68
4.9 การปรับคาบเวลาในการใช้งานซีพียูโดยใช้ระบบควบคุม
แบบสัดส่วน-ปริพันธ์-อนุพันธ์..... 68

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

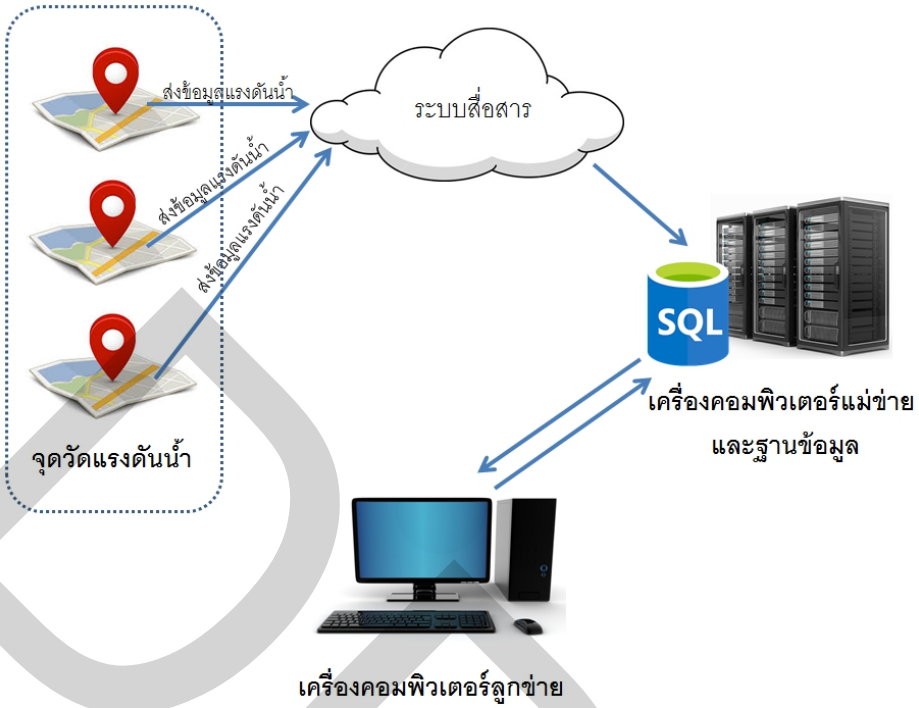
ในการใช้งานโปรแกรมตรวจสอบข้อมูลแบบเวลาจริง (Real-time data monitoring program) เพื่อตรวจสอบแรงดันน้ำในท่อประปาขนาดใหญ่ที่มีขนาดเส้นผ่านศูนย์กลางตั้งแต่ 400 มิลลิเมตร ถึง 1,500 มิลลิเมตร หรือที่เรียกว่าท่อประธาน ของการประปานครหลวงนั้น เป็นการใช้งานแบบเวลาจริง ข้อมูลแรงดันน้ำและอัตราการไหลของน้ำจะมีการเปลี่ยนแปลงให้เป็นปัจจุบัน (Update) ตามคาบเวลาที่กำหนดไว้ ซึ่งในปัจจุบันมีการใช้งานโดยการกำหนดคาบเวลาในการแสดงข้อมูลอยู่ที่ 1 นาที เพื่อไม่ให้เกิดการล่มของระบบ จากการร้องขอข้อมูล (Data Request) พร้อมกันจำนวนมาก และเป็นขีดจำกัดความสามารถทางด้านการเขียนโปรแกรม โดยในปัจจุบันใช้การทำงานของการเขียนโปรแกรมภาษาจาวาสคริปต์[1] (JavaScript) และเจคิวรี่[2] (jQuery) ซึ่งการกำหนดค่าคาบเวลา 1 นาทีนั้น ทำให้การส่งข้อมูลแรงดันน้ำล่าช้า เมื่อเกิดความผิดปกติของแรงดันน้ำ จะทำให้ไม่สามารถแก้ไขปัญหาได้ทันเนื่องจากยังไม่ทราบถึงข้อมูลปัญหาหรือทราบปัญหาล่าช้า เป็นเหตุให้เกิดท่อแตกรั่วหรือระเบิดได้

ภาพรวมของระบบ จะมีส่วนประกอบหลัก อยู่ 4 ส่วน ได้แก่ ส่วนจุดวัดค่าแรงดันน้ำและอัตราการไหลของน้ำจากท่อประธาน ส่วนระบบสื่อสาร ส่วนประมวลผลและฐานข้อมูล และส่วนแสดงผลข้อมูล ดังภาพที่ 1.1

---

[1] JavaScript. “JavaScript.” สืบค้นเมื่อวันที่ 31 กรกฎาคม 2558, จาก https://www.javascript.com/.

[2] JQuery. “jQuery.” สืบค้นเมื่อวันที่ 5 มกราคม 2558, จาก: https://jquery.com/.



ภาพที่ 1.1 ภาพรวมของระบบการรับส่งข้อมูลแรงดันน้ำ

จากภาพที่ 1.1 แสดงให้เห็นถึงการส่งข้อมูลแรงดันน้ำจากจุดวัดแรงดันน้ำ ที่ติดตั้งอยู่ตามตำแหน่งต่างๆ ของท่อประธาน ข้อมูลแรงดันน้ำจะถูกส่งเข้าฐานข้อมูลผ่านระบบสื่อสาร เช่น ระบบ 3G ADSL และอื่นๆ โดยแต่ละจุดวัดจะมีคาบเวลาในการส่งข้อมูลที่ไม่เท่ากันตามการตั้งค่าของแต่ละจุด และข้อมูลจากฐานข้อมูลจะมีการนำมาแสดงผลที่เครื่องคอมพิวเตอร์ลูกข่ายผ่านทางเว็บเซิร์ฟเวอร์เพื่อให้ง่ายต่อการเข้าใช้งาน

ปัญหาอีกอย่างหนึ่งของการเปลี่ยนแปลงค่าให้เป็นปัจจุบันตามกำหนดเวลา คือ การเข้าถึงเครื่องแม่ข่ายและฐานข้อมูลพร้อมกันเป็นจำนวนมากในกรณีที่เปิดดูข้อมูลพร้อมกันหลายๆ ข้อมูล ในปัจจุบันนี้มีจุดวัดแรงดันน้ำที่ต้องทำการตรวจสอบอยู่มากกว่า 200 จุดวัด ส่งผลให้ต้องใช้ทรัพยากรทางคอมพิวเตอร์ค่อนข้างมาก ทำให้เกิดการค้างหรือล่มของหน้าเว็บแอพลิเคชันของเครื่องคอมพิวเตอร์ลูกข่ายที่ใช้งานและในบางครั้งก็เกิดการล่มของเครื่องแม่ข่ายได้เช่นกัน

ข้อมูลแรงดันน้ำในท่อประธานที่นำมาใช้งาน จะได้จากฐานข้อมูลที่เก็บอยู่ในเครื่องเครื่องแม่ข่าย ซึ่งมีปริมาณข้อมูลจำนวนมาก เปลี่ยนแปลงเพิ่มขึ้นอยู่ตลอดเวลา โดยที่ข้อมูลจะมีการเปลี่ยนแปลงไม่พร้อมกัน ขึ้นอยู่กับความสามารถในการส่งค่ามาจากเครื่องวัดแรงดันน้ำที่ใช้ในระบบสกาด้า (SCADA)

จากปัญหาข้างต้น ผู้วิจัยได้มีความสนใจที่จะทำการพัฒนาระบบการตรวจสอบข้อมูลแรงดันน้ำแบบเวลาจริง เพื่อตอบสนองการเปลี่ยนแปลงข้อมูลอย่างทันทีทันใดของข้อมูลแรงดันน้ำ เพื่อลดความล่าช้าในการรับทราบข้อมูล ลดความเสียหายที่อาจจะเกิดขึ้นจากการที่แรงดันน้ำมีการเปลี่ยนแปลงแบบผิดปกติ รวมถึงสร้างแอปพลิเคชันที่เกี่ยวข้องเพื่อใช้เตือนผู้ปฏิบัติงานเมื่อเกิดความผิดปกติได้อย่างทันท่วงที

โดยเทคโนโลยีที่ผู้วิจัยใช้ในการปรับปรุง ได้แก่โนดด็อทเจเอส[3] (Node.js) และซ็อคเก็ตด็อทไอโอ[4] (Socket.io) ที่มีโครงสร้างในการตอบสนองต่อการเปลี่ยนแปลงค่าเป็นปัจจุบันได้แบบเวลาจริง สามารถลดปริมาณการเข้าถึงฐานข้อมูล ลดปริมาณข้อมูลส่งผ่านในเว็บเบราว์เซอร์ (Web browser) ลดข้อผิดพลาดในระบบปฏิบัติการ เป็นผลให้ประสิทธิภาพของระบบดีขึ้น และเพิ่มเสถียรภาพในการทำงานให้กับระบบ[3]

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อทำการปรับปรุงการเขียนโปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำในท่อ ประธานของการประปานครหลวง จากภาษาจาวาสคริปต์เป็นภาษาโนดด็อทเจเอส เพื่อให้โปรแกรมดังกล่าวมีคุณสมบัติดังนี้

1. เพิ่มประสิทธิภาพของโปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำในท่อประธาน เพิ่มความเร็วในการเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบัน จากเดิม 60 วินาที ให้เหลือไม่เกิน 5 วินาที
2. วิจัยและพัฒนาส่วนเสริมโปรแกรมที่มีการปรับสภาพการทำงานซีพียู (CPU Usage) ให้เหมาะสมกับทรัพยากรของเครื่องคอมพิวเตอร์แม่ข่ายแบบอัตโนมัติ โดยสามารถตั้งค่าช่วงการทำงานของซีพียูได้ เพื่อให้การทำงานของโปรแกรมไม่กระทบกับการทำงานของแอพลิเคชันอื่นๆ ภายในเครื่องแม่ข่าย

1.3 ขอบเขตของงานวิจัย

1. งานวิจัยนี้เป็นการใช้งานการเขียนโปรแกรมภาษาโนดด็อทเจเอส ในการปรับปรุงและพัฒนาโปรแกรมตรวจสอบข้อมูลแรงดันน้ำในท่อประปาแบบเวลาจริง ที่มีการใช้กับฐานข้อมูล “มายเอสคิวแอล” (MySQL) เท่านั้น

---

[3] Node.js. “Node.js.” สืบค้นเมื่อวันที่ 1 มกราคม 2558, จาก: https://nodejs.org/.

[4] Socket.IO. “Socket.io.” สืบค้นเมื่อวันที่ 5 มกราคม 2558, จาก: https://socket.io/.

2. การแสดงผลของโปรแกรมตรวจสอบแรงดันน้ำในท่อประปา แบบเวลาจริงบนเบราว์เซอร์รองรับเฉพาะเบราว์เซอร์ที่มีการรองรับภาษาจาวาสคริปต์เท่านั้น

3. การเก็บค่าผลการทดลอง ใช้โปรแกรมทาสก์เมเนเจอร์ (Task Manager) ของระบบปฏิบัติการวินโดวส์ (Windows OS) ในการเก็บค่าการใช้งานทรัพยากรทางคอมพิวเตอร์ร่วมกับกูเกิลโครม (Google Chrome) ณ ช่วงเวลานั้นๆ จำนวน 10 ครั้งและนำมาหาค่าเฉลี่ย

4. การทดสอบการใช้งานโปรแกรมตรวจสอบแรงดันน้ำในท่อประปาแบบเวลาจริง ใช้ทดสอบกับระบบปฏิบัติการวินโดว์เท่านั้น

5. งานวิจัยนี้เป็นการพัฒนาและปรับปรุง ในส่วนของโครงสร้างการรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์แม่ข่ายและฐานข้อมูล โดยใช้การเขียนโปรแกรมภาษาโนดด็อทเจเอสเท่านั้น

6. งานวิจัยนี้เป็นการทำการปรับปรุงโครงสร้าง โปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำ ในท่อประธานของการประปานครหลวงให้เป็นแบบเวลาจริง โดยมีการเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบันใช้เวลาไม่เกิน 5 วินาที

7. งานวิจัยนี้ทำการวัดค่าประสิทธิภาพระหว่างภาษาจาวาสคริปต์และโนดด็อทเจเอสด้วยข้อมูลดังต่อไปนี้ จำนวนการร้องขอข้อมูล ปริมาณการส่งผ่านข้อมูล การทำงานของซีพียูของเครื่องแม่ข่ายและลูกข่าย และความเร็วในการตอบสนองของข้อมูล (Response Time)

8. ผลการทดลองเป็นการเก็บค่าจากระบบที่สร้างขึ้นมาใหม่ โดยใช้โปรแกรมตรวจสอบข้อมูลแรงดันน้ำในท่อประปาแบบเวลาจริงของการประปานครหลวงเป็นกรณีศึกษา

1.4 ประโยชน์ที่คาดว่าจะได้รับ

การปรับปรุงโครงสร้างโปรแกรมตรวจสอบและแสดงผลแรงดันน้ำในท่อประธานแบบเวลาจริง จะทำให้การเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบันมีความรวดเร็วขึ้น ทำให้การรับทราบข้อมูลแรงดันน้ำของผู้ปฏิบัติงานมีความรวดเร็ว สามารถแก้ไขปัญหาในกรณีที่เกิดความผิดปกติของระบบแรงดัน ลดการแตกรั่วและระเบิดของท่อประธานได้

ทั้งยังสามารถเพิ่มประสิทธิภาพของตัวโปรแกรมโดยมีการปรับตั้งค่าการทำงานของซีพียูให้มีความเหมาะสมกับทรัพยากรของทั้งเครื่องคอมพิวเตอร์แม่ข่าย และเครื่องคอมพิวเตอร์ลูกข่าย ลดการล่มหรือค้างของเครื่องคอมพิวเตอร์แม่ข่ายและเครื่องคอมพิวเตอร์ลูกข่ายอันเนื่องมาจากการทำงานที่หนักเกินไปได้อีกด้วย

1.5 อุปกรณ์และเครื่องมือที่ใช้

ในส่วนของซอฟต์แวร์ (Software) การเขียนโปรแกรมตรวจสอบแรงดันน้ำในท่อประธานแบบเวลาจริง มีการใช้โปรแกรมโน้ตแพดพลัส (Notepad++) เป็นหลักในการเขียนโปรแกรม และพัฒนาโดยใช้การเขียนโปรแกรมภาษาโนดด็อทเจเอส จาวาสคริปต์ พีเอชพี มายเอสคิวแอล เอชทีเอ็มแอลไฟว์ (HTML5) ซีเอสเอส (CSS) เว็บซ็อคเก็ต (Web Socket) และซ็อคเก็ตด็อทไอโอ

ในส่วนของฮาร์ดแวร์ (Hardware) คุณสมบัติเครื่องแม่ข่ายและลูกข่ายที่นำมาทดสอบ คือ Intel® Core™i7-3770 CPU 3.40GHz, 8 GB of Ram, 64-bit Operation System, x64-based processor ระบบปฏิบัติการวินโดวส์แปด (Windows 8 OS) และทำการติดตั้งอะแพชี เอชทีทีพี เซอร์เวอร์ (Apache HTTP Server) ที่เครื่องแม่ข่าย

ในส่วนของการทดสอบ ที่เครื่องลูกข่ายที่ใช้ในการทดสอบ จะใช้เว็บเบราว์เซอร์คือ กูเกิลโครมเป็นหลัก (Google Chrome) และทาสก์เมเนเจอร์ (Task Manager) ในการเก็บผลการทดลอง

1.6 แผนการดำเนินงาน

1. ศึกษางานวิจัยและทฤษฎีที่เกี่ยวข้องกับงานวิจัย
2. ศึกษารูปแบบการเขียนโปรแกรมภาษาโนดด็อทเจเอสและซ็อคเก็ตด็อทไอโอ
3. ทดสอบการใช้งานของการเขียนโปรแกรมภาษาโนดด็อทเจเอสและซ็อคเก็ตด็อทไอโอรูปแบบต่างๆ โดยการสร้างโปรแกรมจำลอง (DEMO) เพื่อใช้ในการทดสอบ
4. ออกแบบขั้นตอนวิธีการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์แม่ข่ายและเครื่องคอมพิวเตอร์ลูกข่าย โดยการใช้หลักการทำงานของการเขียนโปรแกรมภาษาโนดด็อทเจเอสผ่านทางซ็อคเก็ตด็อทไอโอ
5. ทำการปรับปรุงโครงสร้างโปรแกรมตรวจสอบแรงดันน้ำในท่อประธานแบบเวลาจริงด้วยการเขียนโปรแกรมภาษาโนดด็อทเจเอสแทนที่โครงสร้างของระบบเดิม คือ จาวาสคริปต์และเจคิวรี่
6. ทดสอบการใช้งานหลังการปรับปรุงโครงสร้าง โดยการปรับตั้งค่าคาบเวลาการเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบันค่าต่างๆ กัน เพื่อหาประสิทธิภาพในด้านต่างๆ ดังนี้ (1) ปริมาณการร้องขอการเข้าถึงฐานข้อมูล (2) ปริมาณการส่งผ่านข้อมูลภายในเว็บเบราว์เซอร์ (Web browser) (3) การใช้ทรัพยากรทางคอมพิวเตอร์ทั้งฝั่งเครื่องแม่ข่ายและลูกข่าย

7. ทำการปรับตั้งค่าคาบเวลา (Interval time) ที่เหมาะสมในการส่งค่าข้อมูลของเครื่องคอมพิวเตอร์แม่ข่าย โดยทำการเก็บค่าการทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่าย ที่มีผลมาจากคาบเวลาค่าต่างๆ กันและนำมาวิเคราะห์ผล

8. วิเคราะห์และสรุปผลงานวิจัย

1.7 ความรู้ที่ได้จากการวิจัย

ในการทำการศึกษาหลักการทำงานของการเขียนโปรแกรมภาษาโนดด็อทเจเอส ทำให้ทราบถึงขีดจำกัดและความสามารถที่แตกต่างกับการทำงานของภาษาจาวาสคริปต์สามารถนำความรู้ที่ได้ไปต่อยอดพัฒนาโปรแกรมได้อีกมากมาย ซึ่งผู้วิจัยมองว่าโนดด็อทเจเอส ถือเป็นการเขียนโปรแกรมเว็บรูปแบบใหม่ ที่มีความน่าสนใจและน่าศึกษาเพิ่มเติม

1.8 ผลงานตีพิมพ์

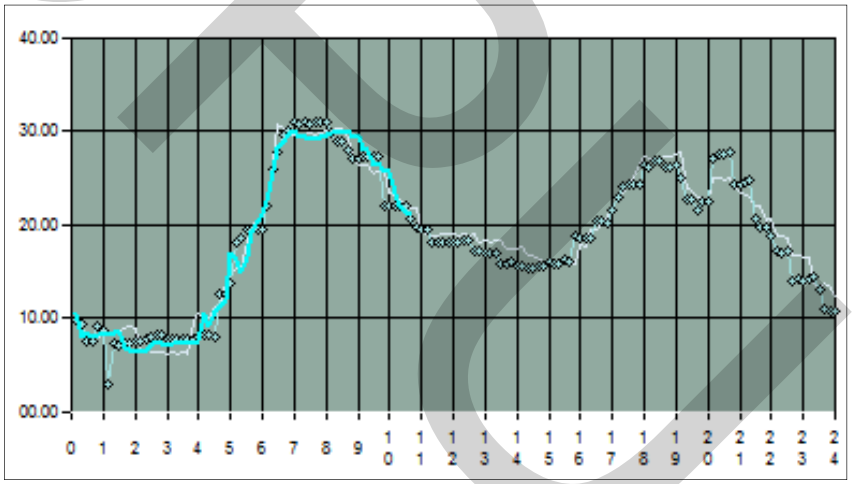
งานวิจัยนี้ได้รับการตีพิมพ์ในงานประชุมวิชาการ งานวิจัย และพัฒนาเชิงประยุกต์ ครั้งที่ 7 (ECTI-CARD 2015) ในวันที่ 8-10 กรกฎาคม พ.ศ. 2558 จัดโดยมหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย วิทยาเขตตรัง และสมาคมวิชาการไฟฟ้าอิเลคทรอนิคส์โทรคมนาคมและสารสนเทศ ประเทศไทยชื่อผลงานตีพิมพ์ “การปรับปรุงโครงสร้างโปรแกรมตรวจสอบข้อมูลแรงดันน้ำในท่อประปาแบบเวลาจริงด้วยสถาปัตยกรรมโนดด็อทเจเอส (Node.js)”

งานวิจัยต่อเนื่องจากนี้ได้รับการตีพิมพ์ในงานประชุมการประชุมเครือข่ายวิชาการด้านวิศวกรรมไฟฟ้า การพัฒนานวัตกรรมเพื่ออุตสาหกรรมและการเกษตรอย่างยั่งยืน ครั้งที่ 9 (EENET2017) ในวันที่ 2-4 พฤษภาคม พ.ศ. 2560 จัดโดยมหาวิทยาลัยเทคโนโลยีราชมงคล ชื่อผลงานตีพิมพ์ “การปรับปรุงอัลกอริทึมสำหรับการจัดการการทำงานของซีพียูแบบอัตโนมัติของโปรแกรมตรวจสอบแรงดันน้ำในท่อประปาของการประปานครหลวง ที่มีการใช้งานโนดด็อทเจเอสในการรับ-ส่งข้อมูล”

บทที่ 2
แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

2.1 ระบบตรวจสอบแรงดันน้ำในท่อประธานของการประปานครหลวง (ก่อนทำการพัฒนา)

การสร้างเว็บแอปพลิเคชัน สำหรับการเฝ้าระวังแรงดันน้ำในท่อประธานแบบเวลาจริง ในปัจจุบัน ตัวโปรแกรมถูกเขียนด้วยโครงสร้างของภาษาจาวาสคริปต์ (JavaScript) และเจคิวรี่ (JQuery) มีการเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบันตามคาบเวลา โดยใช้เวลารอบละ 1-5 นาที และทำการสร้างกราฟความสัมพันธ์ระหว่างแรงดันน้ำและเวลาทุกๆ 1 นาที ตลอด 24 ชั่วโมง โดยที่ข้อมูลแรงดันน้ำจะได้จากฐานข้อมูล ที่มีการจัดเก็บโดยโปรแกรมมายเอสคิวแอล (MySQL) ดังภาพที่ 2.1



ภาพที่ 2.1 ตัวอย่างกราฟความสัมพันธ์ระหว่างแรงดันน้ำและเวลาตลอด 24 ชั่วโมง

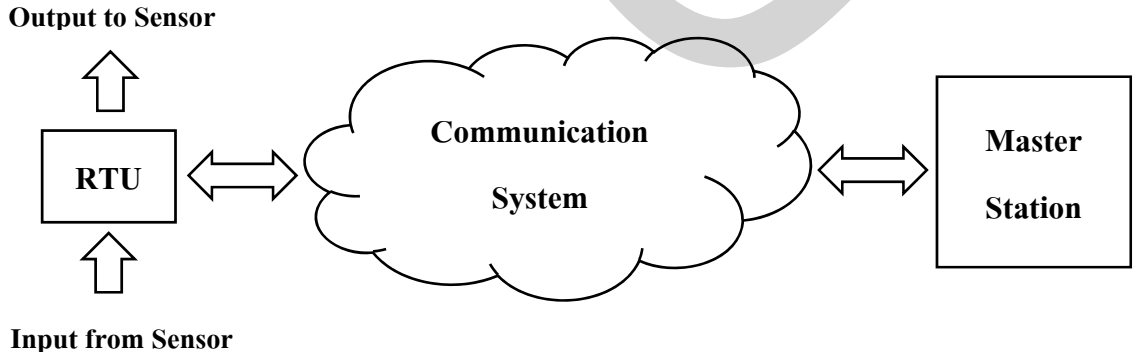
ข้อมูลแรงดันน้ำที่อยู่ในฐานข้อมูล จะมีการรับมาจากแรงดันน้ำที่ได้จากเครื่องมือวัดแรงดันน้ำของท่อประธาน ที่ติดตั้งอยู่ในตำแหน่งต่างๆ ทั่วพื้นที่ให้บริการ ข้อมูลจะถูกส่งผ่านระบบโทรมาตรหรือสกาดา (SCADA : Supervisory Control And Data Acquisition) เข้ามาเก็บไว้ที่ฐานข้อมูลหลัก และนำมาใช้งานกับโปรแกรม

2.2 ระบบโทรมาตร (SCADA: Supervisory Control And Data Acquisition)

ระบบสกาดา เป็นโปรแกรมที่สร้างมาเพื่อใช้ในการจัดการรวบรวมข้อมูลและจัดการข้อมูลภายในฐานข้อมูล รวมถึงการแสดงผลข้อมูลในรูปของ Graphic User Interface (GUI) ตามที่ออกแบบไว้ เพื่อให้เหมาะสมกับความต้องการของผู้ใช้งาน โดยระบบสกาดาจะมีการตรวจรับ-ส่งข้อมูล และควบคุมการทำงานของอุปกรณ์ที่เชื่อมต่อกับระบบ โดยเน้นที่อุปกรณ์ที่ใช้ควบคุมระยะไกลจากสถานที่ปฏิบัติงาน หรือไม่มีผู้ปฏิบัติงานดูแล

เนื่องจากระบบสกาดาเป็นระบบที่ใช้ในการควบคุมและแสดงผลข้อมูลที่ได้จากระยะไกล ดังนั้นระบบสกาดาจึงทำหน้าที่ในการส่งคำสั่งในการควบคุมไปที่หน่วยควบคุมระยะไกล (Remote Terminal Unit : RTU) ที่ติดตั้งอยู่ที่สถานีสนามของระบบ เพื่อให้หน่วยควบคุมระยะไกลทำหน้าที่ในการควบคุมการทำงานของอุปกรณ์และประมวลผลข้อมูลที่สถานีสนามก่อนที่จะทำการส่งข้อมูลไปยังฐานข้อมูลที่อยู่ที่ศูนย์ควบคุม

การกำหนดให้สถานีสนามหรือหน่วยควบคุมระยะไกล เป็นส่วนที่ทำหน้าที่ในการประมวลผลข้อมูลก่อนทำการส่งข้อมูลไปยังฐานข้อมูลหรือศูนย์ควบคุม เนื่องมาจากในการรับส่งข้อมูลระยะไกลนั้น จะต้องใช้ระบบการสื่อสารเป็นตัวกลางในการรับ-ส่งข้อมูล ซึ่งมีข้อจำกัดในด้านปริมาณของข้อมูล หากมีการรับ-ส่งข้อมูลที่มีปริมาณมากเกินไป อาจจะทำให้เกิดความผิดพลาดของข้อมูลได้ ดังนั้น การออกแบบเพื่อลดปริมาณการรับ-ส่งข้อมูล โดยการให้หน่วยควบคุมระยะไกลทำหน้าที่ในการควบคุมอุปกรณ์หรือประมวลผลข้อมูลในการตรวจวัด (จากเครื่องวัด) แล้วจึงส่งข้อมูลมายังศูนย์ควบคุม และศูนย์ควบคุมจะทำหน้าที่เพียงส่งคำสั่งไปที่หน่วยควบคุมระยะไกลให้ดำเนินการเท่านั้น จะทำให้ระบบสกาดาสามารถทำงานได้อย่างมีประสิทธิภาพและมีความน่าเชื่อถือ



ภาพที่ 2.2 แสดงส่วนประกอบหลักของระบบ SCADA

จากภาพที่ 2.2 ระบบสกาดาประกอบด้วย 3 ส่วนหลัก ดังนี้

1. หน่วยควบคุมระยะไกล (Remote Terminal Unit: RTU)
2. ระบบสื่อสาร (Communication System)
3. ศูนย์ควบคุมหรือสถานีหลัก (Master Station)

สำหรับการประปานครหลวงนั้น หน่วยควบคุมระยะไกล จะมีการติดตั้งเครื่องมือวัดแรงดันน้ำ และอัตราการไหลของน้ำ มีการประมวลผลผ่านทางหน่วยควบคุมระยะไกลเพื่อส่งข้อมูลแรงดันน้ำ และอัตราการไหลของน้ำ ผ่านระบบสื่อสารและนำมาเก็บในฐานข้อมูลและแสดงผลที่ศูนย์ควบคุมโดยโปรแกรมสกาดา

2.3 หน่วยควบคุมระยะไกล (Remote Terminal Unit: RTU)

หน่วยควบคุมระยะไกล หรือ Remote Terminal Unit (RTU) เป็นส่วนหนึ่งของระบบสกาดา ซึ่งถูกติดตั้งอยู่ที่สถานีสนามหรือสถานีตรวจวัดข้อมูล โดยหน่วยควบคุมระยะไกลจะถูกต่อกับเครื่องมือวัดข้อมูลที่ต้องการตรวจวัด (ในกรณีนี้ คือ เครื่องมือวัดแรงดันน้ำและอัตราการไหล) และทำการรวบรวมข้อมูลทั้งข้อมูลชนิดอนาล็อก (Analog) และข้อมูลชนิดดิจิตอล (Digital) โดยต่ออุปกรณ์ตรวจวัดข้อมูลเข้ากับส่วนอินพุต (Input Unit) ของหน่วยควบคุมระยะไกล แล้วนำเอาค่าที่ทำการตรวจวัดได้มาทำการประมวลผลและส่งกลับไปแสดงผลที่ศูนย์ควบคุมโดยผ่านระบบสื่อสาร

หน่วยควบคุมระยะไกลจะรับคำสั่งในการควบคุมอุปกรณ์ เครื่องมือวัด จากศูนย์ควบคุม โดยอุปกรณ์เครื่องมือวัดจะถูกติดตั้งเข้ากับส่วนเอาต์พุต (Output Unit) ของหน่วยควบคุมระยะไกล ซึ่งในการควบคุมการทำงานต่างๆ จะมีการสร้างโปรแกรมในการเชื่อมต่อกับส่วนอินพุตเพื่ออ่านค่าที่ตรวจวัดและทำการประมวลผลค่านั้นๆ ให้ออกมาในรูปแบบข้อมูลที่ต้องการโดยมีการติดต่อสื่อสารระหว่างสถานีสนามกับศูนย์ควบคุม โดยผ่านช่องสัญญาณในการสื่อสาร (Communication Port) ของหน่วยควบคุมระยะไกล

หน่วยควบคุมระยะไกล มีส่วนประกอบหลักอยู่ 3 ส่วน ดังนี้

2.3.1 หน่วยประมวลผลกลาง หรือ Central Processing Unit (CPU)

ทำหน้าที่ในการประมวลผลสัญญาณที่ได้รับมาจากเครื่องมือวัดภาคสนาม (Filed Instrument) โดยสัญญาณที่ได้รับมาจากเครื่องมือวัดภาคสนาม จะถูกต่อเข้ากับส่วนรับข้อมูล (I/O Module) ตามมาตรฐานสัญญาณต่างๆ เช่น สัญญาณอนาล็อก (Analog) 4-20 มิลลิแอมป์ หรือสัญญาณดิจิตอลเปิด-ปิด (Digital, On-Off)

เมื่อหน่วยประมวลผลกลางได้รับสัญญาณจากเครื่องมือวัดภาคสนามที่ต่อผ่านส่วนรับข้อมูล (I/O Bus) แล้ว จะทำการประมวลผลโดยการแปลงสัญญาณอนาล็อก เป็นดิจิตอลแล้วนำไปประมวลผลต่อ หรือส่งข้อมูลกลับไปยังศูนย์ควบคุม

นอกจากนี้หน่วยประมวลผลกลาง ยังทำหน้าที่ในการควบคุมระบบการสื่อสารระหว่างหน่วยควบคุมระยะไกลกับศูนย์ควบคุม ผ่านพอร์ตในการสื่อสาร (Communication Port) ซึ่งชนิดของพอร์ตที่ใช้ในการสื่อสารนั้นจะขึ้นอยู่กับสื่อที่ใช้ เช่น สายสัญญาณต่างๆ คลื่นไมโครเวฟ จีพีอาร์เอส (GPRS) ดาวเทียม วิทยุสื่อสาร และอื่นๆ

2.3.2 ส่วนรับ-ส่งข้อมูล หรือ Input/Output Module (I/O Module)

ทำหน้าที่ในการรับ-ส่งสัญญาณจากหน่วยประมลผลกลาง เพื่อส่งไปควบคุมหรืออ่านค่าจากอุปกรณ์เครื่องมือวัดภาคสนาม ซึ่งสามารถแบ่งส่วนรับ-ส่งข้อมูลออกเป็น 4 ชนิด ดังนี้

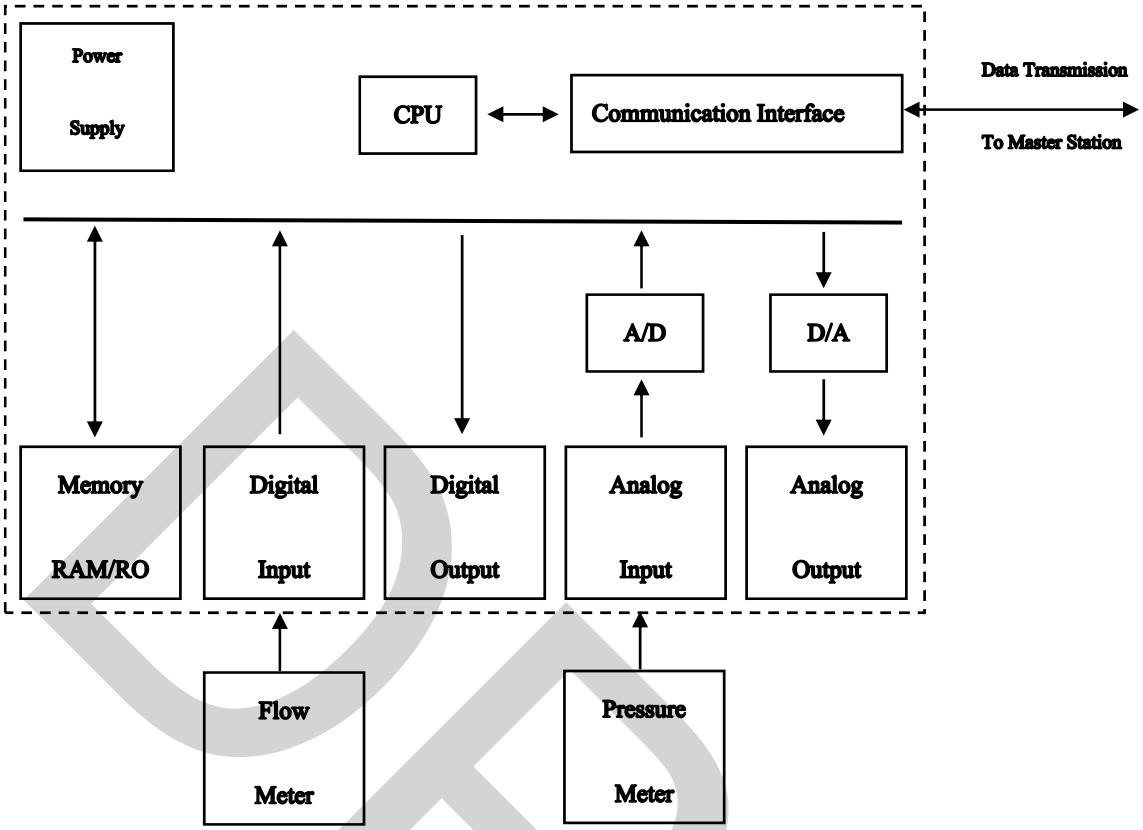
1. ส่วนรับสัญญาณอนาล็อก (Analog Input Module) เป็นส่วนที่รับสัญญาณอนาล็อก ที่เป็นสัญญาณไฟฟ้า 4-20 มิลลิแอมป์ หรือ 0-5 โวลต์ ซึ่งเป็นค่ามาตรฐานทางอุตสาหกรรมจากอุปกรณ์เครื่องมือวัดค่าต่างๆ

2. ส่วนรับสัญญาณดิจิตอล (Digital Input Module) เป็นส่วนที่รับสัญญาณดิจิตอล 0 หรือ 1 ตามลักษณะการปิด-เปิดสวิตช์ (Close or Open Switch)

3. ส่วนส่งออกสัญญาณอนาล็อก (Analog Output Module) เป็นส่วนที่ส่งสัญญาณอนาล็อกที่เป็นสัญญาณไฟฟ้า 4-20 มิลลิแอมป์หรือ 0-5 โวลต์ ซึ่งเป็นค่ามาตรฐานทางอุตสาหกรรม เช่นเดียวกับส่วนรับสัญญาณอนาล็อก

4. ส่วนส่งออกสัญญาณดิจิตอล (Digital Output Module) เป็นส่วนที่ส่งสัญญาณดิจิตอล 0 หรือ 1 ตามลักษณะการปิด-เปิดสวิตช์ (Close or Open Switch)

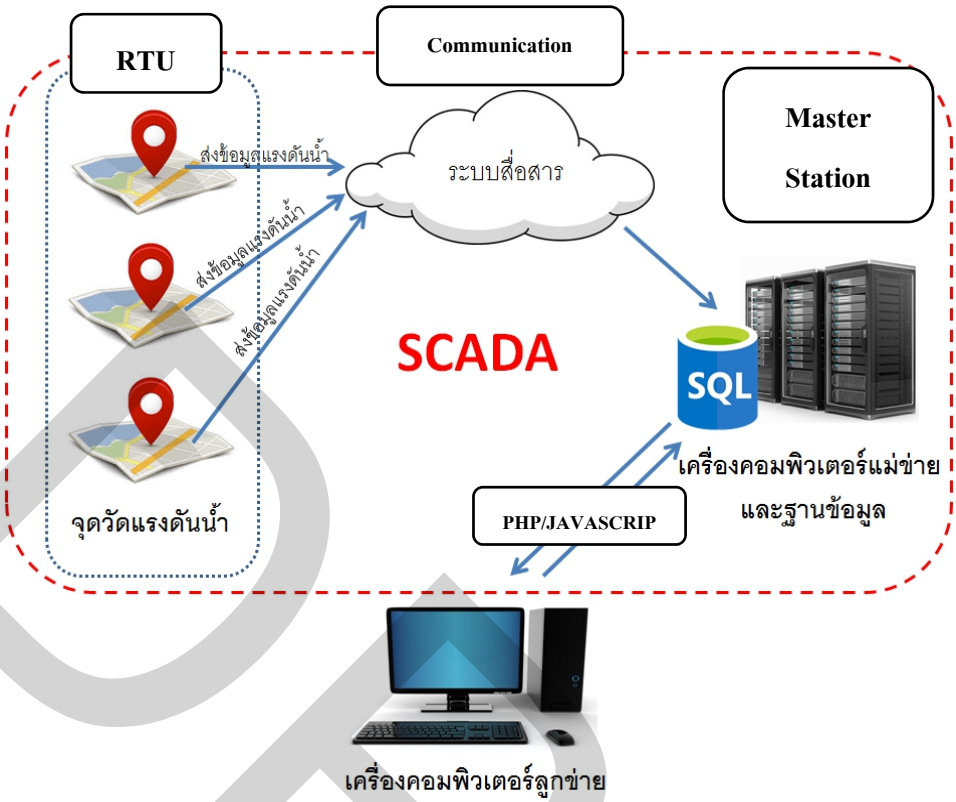
ในการเชื่อมต่อเครื่องมือวัดภาคสนาม เข้ากับส่วนรับข้อมูลอนาล็อกหรือดิจิตอลของหน่วยควบคุมระยะไกล จะต้องต่อเข้ากับส่วนรับข้อมูลอนาล็อกหรือดิจิตอล ให้ตรงกันกับชนิดของสัญญาณของเครื่องมือวัดนั้นๆ ดังภาพที่ 2.3



ภาพที่ 2.3 การเชื่อมต่อระหว่างหน่วยควบคุมระยะไกลกับเครื่องมือวัดภาคสนาม

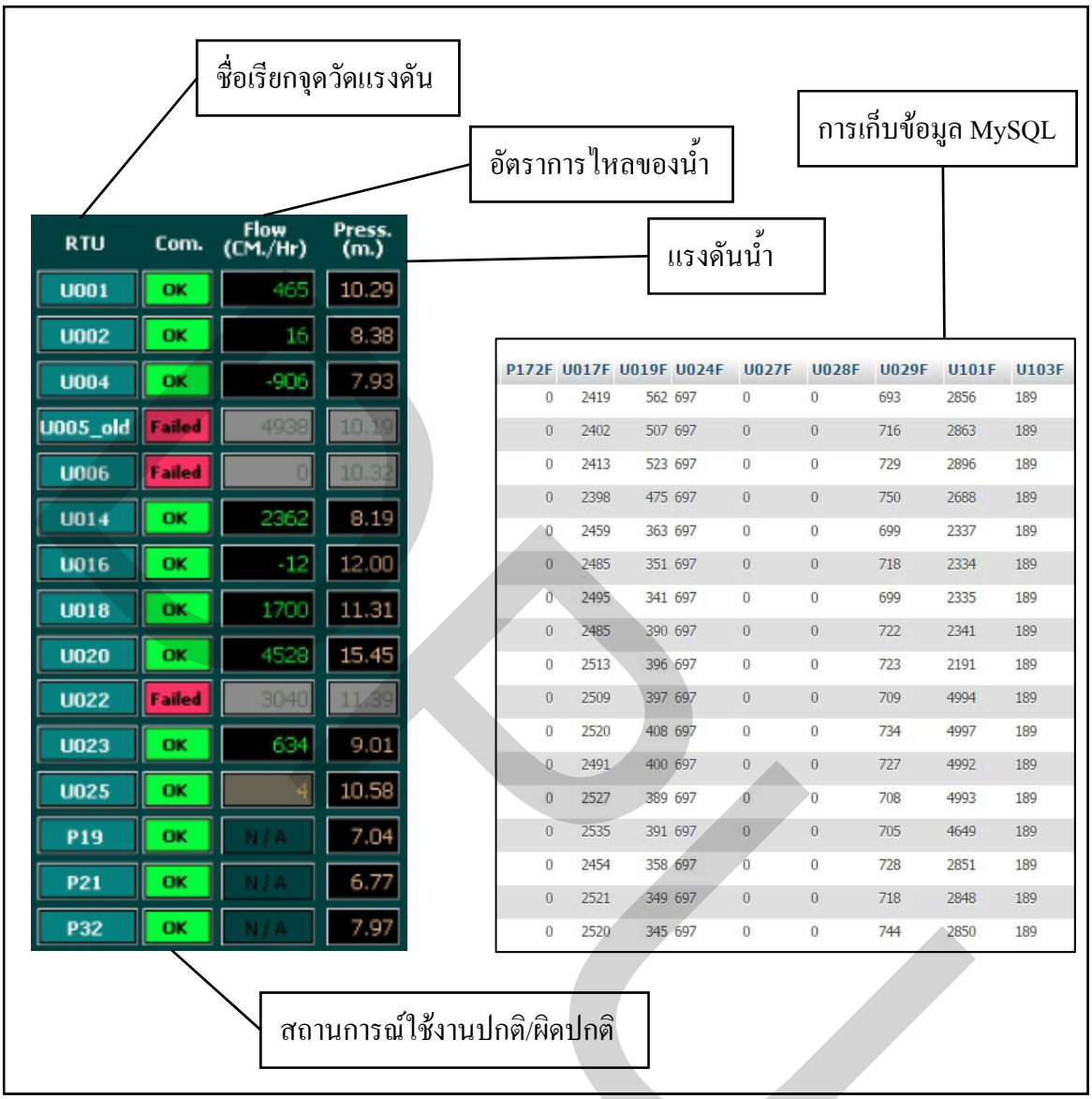
2.3.3 ช่องทางในการสื่อสาร (Communication Port)

ทำหน้าที่เป็นช่องทางในการสื่อสารรับ-ส่งข้อมูลระหว่างหน่วยควบคุมระยะไกลกับศูนย์ควบคุมหรือระหว่างหน่วยควบคุมระยะไกลด้วยกัน ในการสื่อสารสามารถใช้ช่องทาง (Port) ในการสื่อสารมากกว่า 1 ช่องทาง โดยจะต้องทำการกำหนดชนิดของช่องทางในการสื่อสาร รวมถึงการกำหนดสื่อที่ใช้ในการสื่อสารด้วย เช่น วิทยุสื่อสาร ดาวเทียม คลื่นไมโครเวฟ และอื่นๆ ดังภาพที่ 2.4



ภาพที่ 2.4 ภาพรวมของระบบสกาด้าเมื่อนำมาใช้กับโปรแกรมตรวจสอบแรงดันน้ำ

หน้าจอการทำงานของระบบสกาด้าจะมีลักษณะเป็นตารางแสดงข้อมูลแรงดันน้ำและอัตราการไหลของน้ำ โดยจะมีการเก็บข้อมูลลงในฐานข้อมูลมายเอ็สคิวแอลทุกๆ 1 นาที ดังภาพที่ 2.5



ภาพที่ 2.5 หน้าจอระบบสกาด้าที่ศูนย์ควบคุมหรือสถานีหลัก (Master Station) และฐานข้อมูล

2.4 ระบบจัดการฐานข้อมูลมายเอสคิวแอล (MySQL Database)

มายเอสคิวแอล (MySQL) เป็นโปรแกรมจัดการข้อมูลแบบมีการสร้างความสัมพันธ์กันของข้อมูล หรือที่เรียกว่า RDBMS (Relational Database Management System) ซึ่งเป็นฐานข้อมูลที่สามารถจัดเก็บ ค้นหา และเรียงข้อมูล โดยเก็บเป็นตารางข้อมูลที่มีความสัมพันธ์กันระหว่างข้อมูลต่างๆ

ข้อได้เปรียบของมายเอสคิวแอล คือ มีการทำงาน เข้าถึงข้อมูล ได้อย่างรวดเร็ว และยังสามารถเข้าใช้งานโดยการดึงข้อมูลออกมาใช้งานได้พร้อมๆ กันจากคอมพิวเตอร์หลายเครื่อง ในปัจจุบันมายเอสคิวแอลได้รับความนิยมเป็นอย่างมาก เนื่องจากใช้งานได้ง่าย และเปิดให้ผู้พัฒนาใช้งานได้อย่างอิสระ (Open source)

2.5 การเขียนโปรแกรมภาษาฝั่งแม่ข่าย (Server-Side Language Structure)

การเขียนโปรแกรมภาษาฝั่งแม่ข่าย หมายถึง ภาษาทางคอมพิวเตอร์ที่ใช้สำหรับการติดต่อ จัดการข้อมูลที่ฐานข้อมูล เพื่อนำมาใช้งาน หรือแสดงผลที่ฝั่งผู้ใช้งาน ได้แก่ ภาษาพีเอชพี (PHP)

ภาษาพีเอชพี (PHP) เป็นภาษาที่ใช้การเขียนโปรแกรมบนเว็บไซต์ โดยที่พีเอชพี จะถูกใช้งานเป็นส่วนในการเขียนฟังก์ชันการคำนวณ ประมวลผล เก็บค่าตัวแปร และทำตามคำสั่งต่างๆ เช่น การรับค่าอินพุตจากแบบฟอร์มรับค่า เพื่อนำไปเก็บในฐานข้อมูล หรือนำไปคำนวณต่อ

ภาษาพีเอชพีจะถูกใช้งานร่วมกับภาษาเอชทีเอ็มแอล ในการสร้างเว็บไซต์ แต่ภาษาพีเอชพี จะแตกต่างกับภาษาเอชทีเอ็มแอล (HTML) ตรงที่ภาษาเอชทีเอ็มแอลจะเป็นส่วนแสดงผลหน้าเว็บไซต์จัดตำแหน่งรูปหรือจัดรูปแบบตัวอักษร ให้ตรงตามที่ออกแบบเท่านั้น จะไม่มีการคำนวณ หรือประมวลผลใดๆ

2.6 โครงสร้างภาษาฝั่งลูกข่าย (Client-Side Language Structure)

โครงสร้างทางภาษาฝั่งลูกข่าย หมายถึง ภาษาทางคอมพิวเตอร์ที่ใช้ในการแสดงผลหรือเปลี่ยนแปลงส่วนแสดงผลที่หน้าเว็บไซต์ฝั่งเครื่องคอมพิวเตอร์ลูกข่ายให้มีการแสดงผลรูปแบบหรือเคลื่อนไหวตามลักษณะการทำงานของฟังก์ชันต่างๆ ที่ผู้ออกแบบต้องการ โดยไม่มีผลกระทบต่อข้อมูลหลักได้แก่ ภาษาจาวาสคริปต์ (JavaScript) เจคิวรี่ (jQuery) วิชวลเบสิกสคริปต์ (VBScript) เอชทีเอ็มแอล (HTML) และซีเอ็สเอ็ส (CSS) เป็นต้น

ในบางครั้งอาจมีการนำข้อมูลจากฝั่งแม่ข่ายเข้ามาอ้างอิงในฟังก์ชัน แต่จะมีการคำนวณและประมวลผลที่ฝั่งลูกข่าย คือ ประมวลผลโดยเว็บเบราว์เซอร์ที่เครื่องลูกข่ายทั้งหมด ดังนั้นประโยชน์หลักของการทำงานฝั่งลูกข่ายก็คือช่วยลดการทำงานที่ฝั่งแม่ข่ายได้นั่นเอง

2.7 จาวาสคริปต์ (JavaScript)

ภาษาจาวาสคริปต์ (JavaScript) คือ ภาษาทางคอมพิวเตอร์ที่ใช้สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต จาวาสคริปต์ เป็นภาษาสคริปต์เชิงวัตถุ (Object) ถูกใช้ในการสร้างและพัฒนาเว็บไซต์ร่วมกับภาษาเอชทีเอ็มแอล (HTML) เพื่อให้เว็บไซต์มีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น จาวาสคริปต์ (JavaScript) สามารถทำให้เว็บเพจมีลูกเล่นต่างๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม

เนื่องจากจาวาสคริปต์ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการ จึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง และได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของจาวาสคริปต์จะมีการแปลคำสั่งโดยเบราว์เซอร์ (Client-side Script) ดังนั้นจาวาสคริปต์จะสามารถทำงานได้เฉพาะบนเบราว์เซอร์ที่สนับสนุน ซึ่งปัจจุบันเบราว์เซอร์เกือบทั้งหมดก็สนับสนุนจาวาสคริปต์แล้ว

2.8 เจคิวรี่ (jQuery)

เจคิวรี่ (jQuery) คือ ไลบรารี่ (Library) ของภาษาจาวาสคริปต์ (JavaScript Library) ซึ่งถูกออกแบบมาเพื่อให้การเขียนจาวาสคริปต์ (JavaScript) นั้นมีความสะดวกและง่ายขึ้น กล่าวคือ เจคิวรี่เป็นการเขียนฟังก์ชันของจาวาสคริปต์เตรียมพร้อมไว้สำหรับการใช้งาน ซึ่งในการเรียกใช้งานเจคิวรี่จะสั้นและซับซ้อนน้อยกว่าการเรียกใช้งานจาวาสคริปต์แบบปกติในการทำงานอย่างเดียวกัน

การนำจาวาสคริปต์ไปประยุกต์ใช้งานจำพวกเว็บไซต์ (Client-side JavaScript) นั้นเป็นสิ่งที่ยุ่งยาก และซับซ้อน ทั้งในเรื่องของความไม่เข้ากันของเว็บเบราว์เซอร์ (Web Browser) แต่ละค่าย ยังรวมถึง DOM (Document Object Model) และ API (Application Programming Interface) ที่แตกต่างกันอีกด้วย ดังนั้นเจคิวรี่จึงมีการรวมเอาฟังก์ชันต่างๆ ที่จำเป็นมารวบรวมไว้ในรูปแบบของไลบรารี่เพื่อให้การใช้งานจาวาสคริปต์นั้นมีความสะดวก กระชับ รวดเร็ว และสามารถใช้งานได้กับทุกเบราว์เซอร์

หลักการทำงานของเจคิวรี่ที่เกี่ยวข้องกับงานวิจัยนี้ จะเป็นการใช้งานไลบลารี่ภาษาจาวาสคริปต์ ในการแสดงข้อมูลเป็นคาบเวลา โดยมีความถี่ตามคาบเวลาที่ทำการตั้งค่าไว้ หรือเรียกว่าเป็นการเปลี่ยนแปลงข้อมูลแบบอัตโนมัติ (Auto Update)

ในการเขียนโปรแกรมลักษณะนี้ จะทำให้เกิดการร้องขอข้อมูลซ้ำๆ ทุกครั้งตามคาบเวลา ซึ่งจะทำให้เกิดการเข้าถึงฐานข้อมูลและส่งผ่านข้อมูล (Data Transfer) ระหว่างเครื่องลูกข่ายและเครื่องแม่ข่ายเป็นจำนวนมาก เกิดเฮดเดอร์ (Header) จำนวนมาก ซึ่งเป็นผลให้เกิดการค้างหรือข้อผิดพลาด (Error) อันเนื่องมาจากการรับ-ส่งข้อมูลที่เกินความจำเป็นขึ้นได้

2.9 โนดด็อทเจเอส (Node.js)

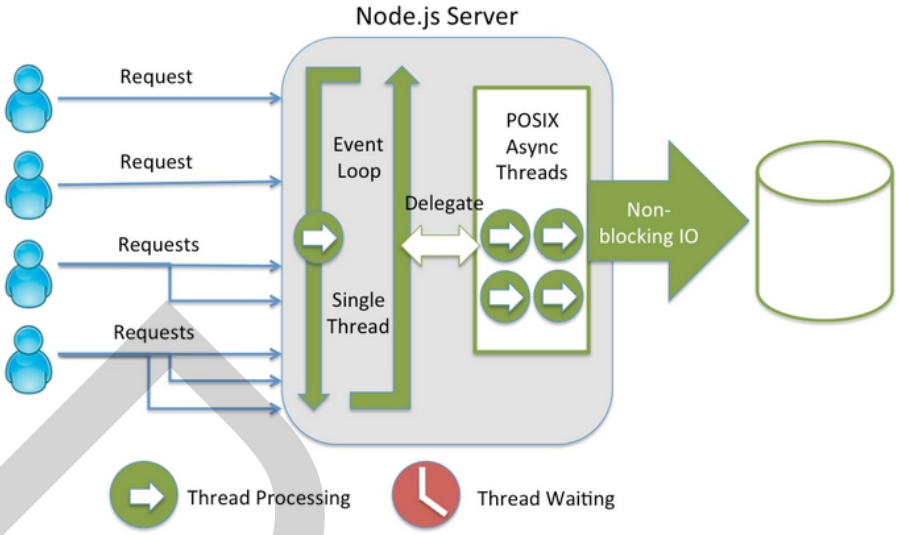
โนดด็อทเจเอส คือ การเขียนโปรแกรมด้วยภาษาจาวาสคริปต์ (JavaScript) ที่ฝั่งเครื่องคอมพิวเตอร์แม่ข่าย (Server Side) แทนการเขียนที่ฝั่งผู้ใช้งาน (Client Side) โนดด็อทเจเอสมีความสามารถในการประมวลผลที่รวดเร็ว ใช้ได้กับระบบเครือข่ายขนาดใหญ่ รวมถึงรองรับการแสดงผลข้อมูลแบบเวลาจริงได้อย่างแท้จริง[1]

โครงสร้างของการเขียนโปรแกรมภาษาโนดด็อทเจเอส มีความสามารถในการรองรับการเชื่อมต่อจากผู้ใช้งานจำนวนมาก (ระดับ 1,000 การเชื่อมต่อขึ้นไป) โปรแกรมจะทำการสร้างลูปของการเชื่อมต่อทั้งหมด แล้วแปลงให้เป็นการทำงานแบบไม่ต้องเรียงตามลำดับ (Asynchronous) และส่งการร้องขอในรูปของน็อนบล็อกกิ้งไอโอ (Non-blocking IO) ทำให้ไม่เกิดการรอคอยของภาระงาน (Thread Waiting) ซึ่งข้อได้เปรียบนี้ทำให้การเขียนโปรแกรมภาษาโนดด็อทเจเอสมีการทำงานที่เร็วกว่าแบบอื่น ดังภาพที่ 2.6

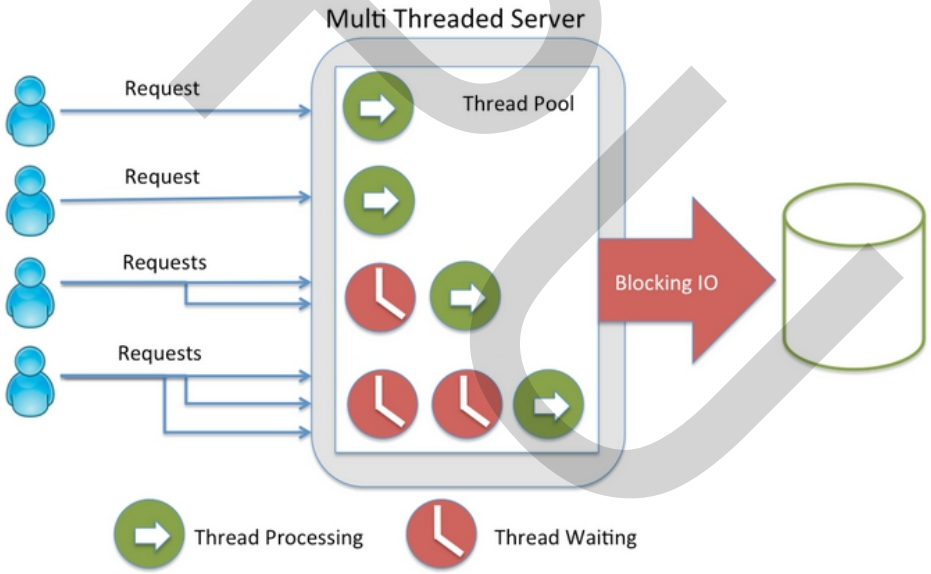
เมื่อทำการเปรียบเทียบกับการทำงานแบบปกติ ดังภาพที่ 2.7 แสดงให้เห็นว่า ในการร้องขอจากผู้ใช้งานเข้าในปริมาณมาก (ปริมาณมากกว่า 1,000 การร้องขอขึ้นไป) เครื่องแม่ข่ายจะทำการจัดลำดับการทำงานขึ้นอยู่ในรูปของบล็อกกิ้งไอโอ (Blocking IO) ซึ่งจะทำให้เกิดการรอเวลาในการทำงานของแต่ละเทร็ด (Thread Waiting)

---

[1] Strongloop. “What Makes Node.js Faster Than Java?” Retrieved October 01, 2015, from https://strongloop.com/strongblog/node-js-is-faster-than-java/



ภาพที่ 2.6 โครงสร้างการทำงานของแม่ข่ายโนดด็อทเจเอส (Node.js Server)[2]



ภาพที่ 2.7 โครงสร้างการทำงานแบบแม่ข่ายมัลติเทร็ด (Multi-Threaded Server)[2]

---

[2] Strongloop. "What Makes Node.js Faster Than Java?" Retrieved October 01, 2015, from https://strongloop.com/strongblog/node-js-is-faster-than-java/

2.10 ซ็อกเก็ตด็อทไอโอ (Socket.io)

ซ็อกเก็ตด็อทไอโอ คือ ไลบรารี่ (Library) ที่ถูกเขียนขึ้นมาโดยภาษาโนดด็อทเจเอส (Node.js Script) สามารถทำงานแบบเวลาจริง (real-time) และรองรับการทำงานหลายๆ อย่างพร้อมกันในเวลาเดียว (Asynchronous) หลักการทำงานของซ็อกเก็ตด็อทไอโอ คือ จะทำการวนลูป (Looping) ตลอดเวลาด้วยความรวดเร็ว เพื่อคอยรับเหตุการณ์ (event) ที่เข้ามาแล้วนำไปทำงานต่อตามการออกแบบของผู้พัฒนาระบบ

ซ็อกเก็ตด็อทไอโอ ถูกสร้างขึ้นมาเพื่อใช้ในการรับ-ส่งข้อมูล ระหว่างเครื่องคอมพิวเตอร์แม่ข่าย (Server) และเครื่องคอมพิวเตอร์ลูกข่าย (Client) ให้มีการทำงานที่ต่อเนื่องและมีการเปลี่ยนแปลงให้เป็นปัจจุบัน (Up-to-Date) แบบเวลาจริง ให้เป็นเรื่องที่ง่าย

ทฤษฎี (method) ที่นิยมใช้งานมี 2 ชนิด ดังนี้

1. socket.on (message, callback) ใช้สำหรับรับข้อความจากเครื่องคอมพิวเตอร์แม่ข่าย (Receive message from server-side)

2. socket.emit (message, args) ใช้สำหรับส่งข้อความไปที่เครื่องคอมพิวเตอร์ฝั่งแม่ข่าย (Send message to server-side)

2.11 งานวิจัยที่เกี่ยวข้อง

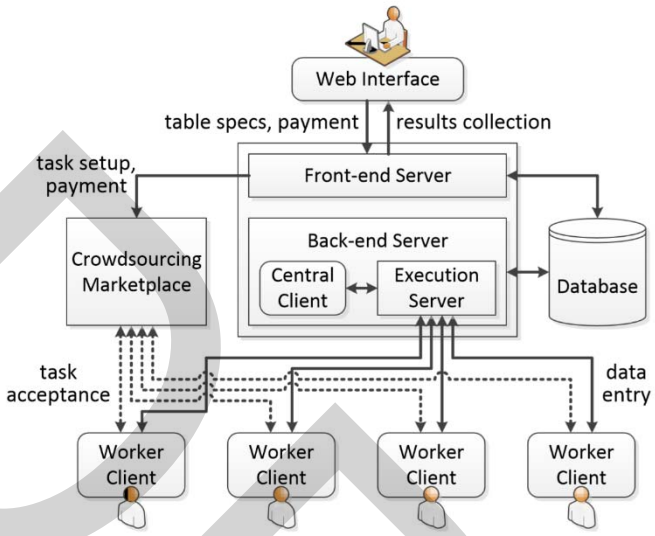
ในการทำการวิจัยเรื่องการปรับปรุงโครงสร้างโปรแกรมตรวจสอบแรงดันน้ำในท่อประปา ผู้วิจัยได้อาศัยหลักการทำการของโครงสร้างการเขียนโปรแกรมภาษาโนดด็อทเจเอส ที่มีความสามารถในการรับ-ส่งข้อมูลจากฐานข้อมูลได้อย่างรวดเร็ว และมีการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ลูกข่าย (ผู้ใช้งาน) กับเครื่องคอมพิวเตอร์แม่ข่าย ผ่านทางซ็อกเก็ตด็อทไอโอและโนดด็อทเจเอส เพื่อให้เว็บเบราว์เซอร์แสดงผลข้อมูลแบบเวลาจริง และได้ทำการศึกษาแนวคิดจากงานวิจัยที่เกี่ยวข้อง ดังต่อไปนี้

2.11.1 Crowd Fill: A System for Collecting Structured Data from the Crowd

งานวิจัยนี้จะเป็นการสร้างเว็บแอปพลิเคชันเกี่ยวกับรายละเอียดต่างๆ ของนักฟุตบอลอาชีพ เช่น ชื่อ-นามสกุล เชื้อชาติ ตำแหน่ง จำนวนประตูที่ทำได้ และสถิติอื่นๆ โดยที่เว็บเบราว์เซอร์จะใช้เป็นส่วนแสดงผลข้อมูลที่ได้จากการรวบรวมข้อมูลของทีมงาน และจะมีการป้อนข้อมูลเข้าสู่เครื่องคอมพิวเตอร์แม่ข่ายทันทีที่ได้รับข้อมูลใหม่ (หมายถึง ทีมงานหลายคน จะทำการป้อนข้อมูลที่ได้รับมาใหม่ เข้าไปยังเครื่องคอมพิวเตอร์แม่ข่ายพร้อมๆ กัน)

โปรแกรมจะทำการ เชื่อมต่อเครื่องคอมพิวเตอร์แม่ข่ายกับเครื่องคอมพิวเตอร์ลูกข่ายผ่านซ็อกเก็ตด็อทไอโอ โดยอาศัยโครงสร้างโนดด็อทเจเอสเพื่อส่งข้อมูลไปแสดงผลที่ฝั่งผู้ใช้งาน

แบบเวลาจริง ข้อมูลที่ผู้ใช้งานได้รับ จะเป็นข้อมูลที่มีการเปลี่ยนแปลงให้เป็นปัจจุบันอยู่เสมอ ดังภาพที่ 2.8



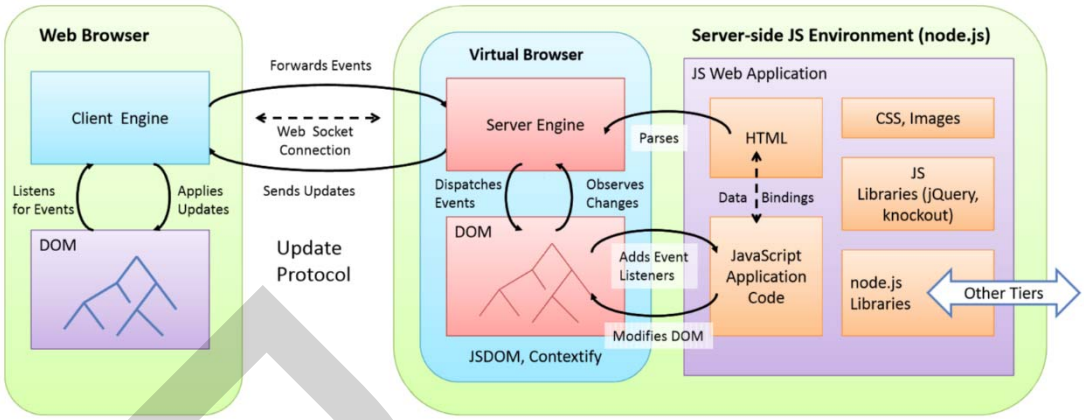
ภาพที่ 2.8 ภาพรวมโครงสร้างการทำงานของระบบโครวด์ฟิล (Crowd Fill)[3]

2.11.2 The Cloud Browser: Web Application Framework

งานวิจัยนี้เป็นการสร้างเว็บเบราว์เซอร์จำลอง (Virtual Browser) ที่ฝั่งเครื่องคอมพิวเตอร์แม่ข่าย โดยมีการติดตั้งโนดด็อทเจเอส เพื่อใช้เป็นส่วนควบคุมและจัดการข้อมูลจากทางฝั่งแม่ข่าย (ใช้ภาษาจาวาสคริปต์ในการกำหนดการทำงาน) ข้อมูลจากฐานข้อมูลจะถูกเปลี่ยนเป็นส่วนแสดงผลจำลอง เพื่อเตรียมพร้อมสำหรับการเชื่อมต่อจากฝั่งผู้ใช้งาน (Client Side)

นอกจากนี้โนดด็อทเจเอส จะสร้างเว็บซ็อกเก็ต (Web Socket) เชื่อมต่อข้อมูลระหว่างเว็บเบราว์เซอร์ของผู้ใช้งานและเว็บเบราว์เซอร์จำลองที่มีการสร้าเตรียมไว้แล้ว เพื่อให้การแสดงผลข้อมูลที่เว็บเบราว์เซอร์ของเครื่องคอมพิวเตอร์ของผู้ใช้งาน แสดงผลได้ตามเวลาจริง ดังภาพที่ 2.9 แสดงโครงสร้างการทำงานภาพรวมของระบบคลาวด์เบราว์เซอร์

[3] Hyunjung Park and Jennifer Widom. (2014). CrowdFill: a system for collecting structured data from the crowd." Retrieved October 01, 2015 from https://dx.doi.org/10.1145/2567948.2577029



ภาพที่ 2.9 ภาพรวมโครงสร้างการทำงานของระบบคลาวด์เบราเซอร์ (Cloud Browser)[4]

2.11.3 CoPerformance: A Rapid Prototyping Platform for Developing Interactive Artist-audience Performances with Mobile Devices

งานวิจัยนี้จะเป็นการสร้างแบบจำลองเพื่อใช้พัฒนาโปรแกรมออกแบบสิ่งต่างๆ ผ่านโทรศัพท์มือถือแบบสมาร์ทโฟน (Smart Phone) เนื่องจากโทรศัพท์เคลื่อนที่ในปัจจุบันจะมีอุปกรณ์อินพุต (Input) ข้อมูลหลากหลายชนิด เช่น ไมโครโฟน กล้อง เข็มทิศ และอื่นๆ โดยผู้วิจัยจะเน้นการออกแบบระบบหรือเครื่องมือที่สามารถโต้ตอบกับผู้ใช้งานที่เป็นนักออกแบบได้ทันทีทันใด โดยที่ระบบจะใช้การป้อนอินพุตจากสื่อสารตามมาตรฐาน 4 ชนิด คือ ซ็อกเก็ตด็อทไอโอ ยูดีพี (UDP) มิดิ (MIDI) และซีเรียล (Serial) โดยมีการใช้งานโนดด็อทเจเอส ในการรับ-ส่งข้อมูลระหว่างอินพุตและฟังก์ชันการทำงาน

การเปรียบเทียบกับงานวิจัยที่เกี่ยวข้อง

เมื่อทำการเปรียบเทียบกับงานวิจัยที่เกี่ยวข้องแล้วนั้น งานวิจัยนี้มีส่วนที่เกี่ยวข้องกับงานวิจัยข้างต้น คือ ในเรื่องของความเร็วในการแสดงผลของข้อมูลจากฐานข้อมูลเมื่อมีการเปลี่ยนแปลงข้อมูลในฐานข้อมูล ก็จะสามารถแสดงผลที่เว็บเบราว์เซอร์ได้ในทันที แต่ลักษณะสำคัญของงานวิจัยนี้ที่แตกต่างไปจากงานวิจัยข้างต้น คือ การหาขีดความสามารถของโนดด็อทเจเอส เมื่อต้องการแสดงผลข้อมูลแรงดันน้ำจากฐานข้อมูล ที่หน้าเว็บเบราว์เซอร์ให้เร็วที่สุดเมื่อมีการเปลี่ยนแปลงข้อมูลที่ได้รับจากเครื่องวัดแรงดัน ที่ส่งเข้าสู่ฐานข้อมูล เนื่องจากในการใช้งานจริง

---

[4] Ariel Ortiz. (2014). “Server-side web development with JavaScript and Node.js”. Retrieved October 01, 2015 from http://doi.acm.org/10.1145/2538862.2539001

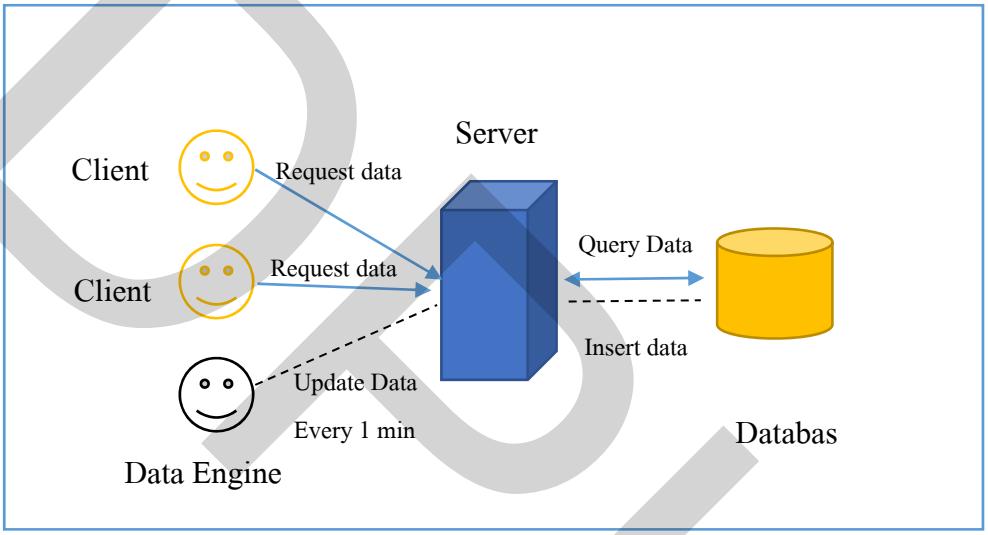
ความเร็วในการแสดงข้อมูลแรงดันน้ำมีความสำคัญมาก เนื่องจากแรงดันน้ำเป็นตัวชี้วัดที่สำคัญที่สุดของระบบการสูบจ่ายน้ำประปา ที่จะสามารถระบุได้ว่าระบบการจ่ายน้ำมีการทำงานผิดพลาดหรือไม่ เพราะถ้าเกิดความผิดปกติของแรงดันน้ำเพียงไม่กี่วินาที แรงดันน้ำที่สูงผิดปกติจะส่งผลให้เกิดท่อแตกและท่อระเบิดได้ ซึ่งจะส่งผลกระทบกับประชาชนผู้ใช้น้ำเป็นวงกว้าง และเป็นการสิ้นเปลืองงบประมาณในการซ่อมแซมอีกด้วย

นอกจากนี้งานวิจัยนี้ยังให้ความสำคัญกับความเสถียรภาพของโปรแกรมเมื่อใช้ในการทำงานอย่างต่อเนื่อง เพื่อไม่ให้เกิดการล่มหรือค้างอันเนื่องมาจากการใช้งานทรัพยากรทางคอมพิวเตอร์ที่เกินขีดจำกัดอีกด้วย

DPU

บทที่ 3
ระเบียบวิธีวิจัย

3.1 สถาปัตยกรรมฮาร์ดแวร์



ภาพที่ 3.1 ภาพรวมโครงสร้างสถาปัตยกรรมฮาร์ดแวร์ของงานวิจัย

จากภาพที่ 3.1 โครงสร้างการทำงานของโปรแกรมตรวจสอบแรงดันน้ำในท่อประธาน จะประกอบด้วยส่วนหลักๆ ดังนี้

3.1.1 ฐานข้อมูล (Database) เป็นส่วนที่ใช้เก็บข้อมูลแรงดันน้ำ โดยข้อมูลจะมีการเปลี่ยนแปลงให้เป็นปัจจุบันตลอดเวลา ซึ่งได้รับค่ามาจากจุดวัดแรงดันน้ำผ่านระบบสกาด้า แต่สำหรับการวิจัยนี้ผู้วิจัยได้ทำการจำลองข้อมูล (Simulation) โดยการสร้างระบบจำลองข้อมูลแรงดันน้ำในฐานข้อมูลขึ้นมาเอง โดยใช้ภาษาจาวาสคริปต์และพีเอชพี โดยให้ข้อมูลแรงดันน้ำมีการเปลี่ยนแปลงทุกๆ 1 นาที เพื่อสังเกตการทำงานของระบบ

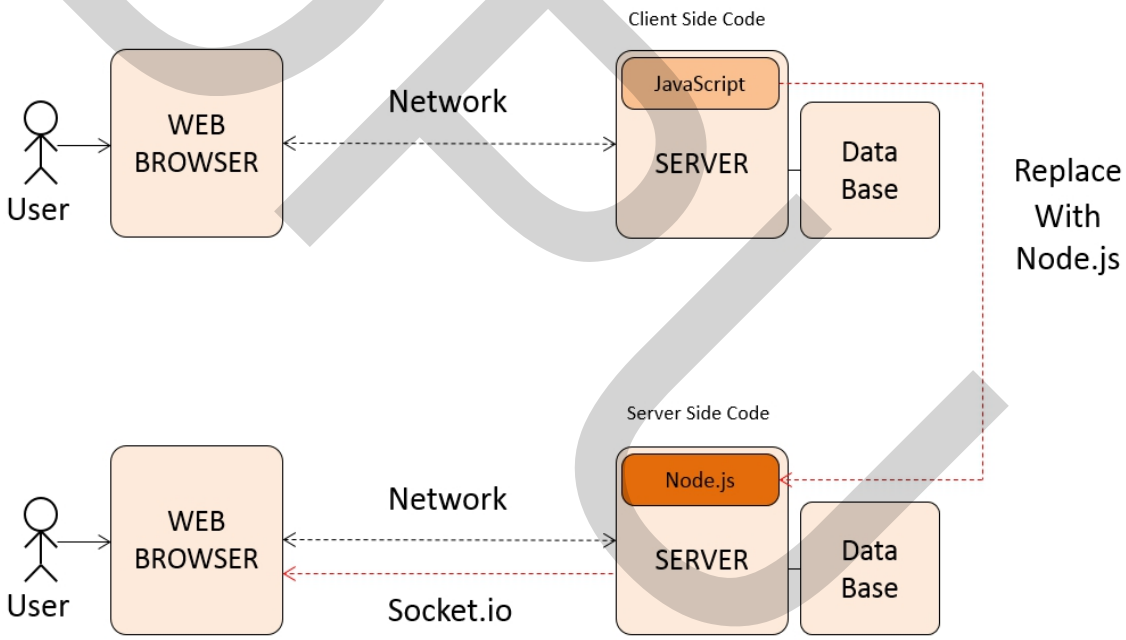
3.1.2 เครื่องคอมพิวเตอร์แม่ข่าย (Server) เป็นส่วนที่ใช้สำหรับประมวลผลแรงดันน้ำ ที่ได้รับจากฐานข้อมูล นำมาสร้างเป็นเว็บแอปพลิเคชัน เพื่อเป็นส่วนติดต่อกับเครื่องคอมพิวเตอร์ลูกข่ายและผู้ใช้งาน

3.1.3 เครื่องคอมพิวเตอร์ลูกข่าย (Client) หรือเครื่องผู้ใช้งาน เป็นส่วนที่ทำหน้าที่แสดงผลเว็บแอปพลิเคชันเพื่อนำไปใช้งานในการควบคุมและจัดการแรงดันน้ำในท่อประธาน

นอกจากนี้ เครื่องคอมพิวเตอร์ลูกข่าย 1 เครื่อง จะถูกนำมาใช้ในการจำลองข้อมูล (Data Engine) เพื่อให้ฐานข้อมูลมีการเปลี่ยนแปลงค่าแรงดันน้ำ ทุกๆ 1 นาที อีกด้วย

3.2 แผนการดำเนินงาน

จุดประสงค์ของงานวิจัยนี้ คือ มุ่งเน้นไปในด้านการปรับปรุงและพัฒนาประสิทธิภาพของโปรแกรมตรวจสอบแรงดันน้ำในท่อประธานแบบเวลาจริง โดยการทำการปรับปรุงโปรแกรมตรวจสอบแรงดันน้ำในท่อประธานที่มีการใช้งานภาษาจาวาสคริปต์อยู่เดิมมาเป็นการใช้งานภาษาโนดด็อทเจเอส ดังภาพที่ 3.2



ภาพที่ 3.2 การเขียนโปรแกรมภาษาโนดด็อทเจเอสแทนภาษาจาวาสคริปต์

ซึ่งมีรายละเอียดขั้นตอนการดำเนินงานดังต่อไปนี้

3.2.1 ศึกษาหลักการทำงานของโปรแกรมตรวจสอบแรงดันน้ำในท่อประธาน (แบบเก่า)

การศึกษาโครงสร้างการทำงานของโปรแกรมตรวจสอบแรงดันน้ำในท่อประธานรูปแบบเดิม นับเป็นจุดเริ่มต้นในการพัฒนาระบบ เนื่องจากการที่จะปรับปรุงและพัฒนาระบบได้นั้น จะต้องรู้จักข้อดี-ข้อเสีย และปัญหาของการทำงานรูปแบบเดิมเสียก่อน จึงจะสามารถพัฒนาระบบให้ดียิ่งขึ้นได้ การทดสอบการใช้งานรูปแบบต่างๆ เพื่อหาขีดจำกัดและข้อบกพร่องของระบบเดิมก็นับเป็นส่วนสำคัญในการได้มาซึ่งข้อมูล เพื่อใช้ในการวิเคราะห์และออกแบบระบบที่ดียิ่งขึ้น

สำหรับการงานวิจัยนี้ โปรแกรมตรวจสอบแรงดันน้ำในท่อประธานแบบเก่านั้น มีการใช้หลักการทำงานของจาวาสคริปต์ ร่วมกับ พีเอชพี และเอชทีเอ็มแอล ในการประมวลผลข้อมูลแรงดันและนำมาแสดงในเว็บเบราว์เซอร์ในส่วนของผู้ใช้งาน ดังนั้น ขั้นตอนนี้จึงเป็นการตั้งปัญหาว่า เราจะสามารถพัฒนาและปรับปรุงระบบเก่าให้ดียิ่งขึ้นได้หรือไม่ และอย่างไร

3.2.2 การศึกษาเทคโนโลยีสมัยใหม่ เพื่อนำมาทดแทน

การศึกษาหาเทคโนโลยีสมัยใหม่ ที่สามารถนำมาแก้ไข ปรับปรุง และทดแทนเทคโนโลยีเดิม เป็นการเพิ่มโอกาสในการปรับปรุงโครงสร้างของระบบให้ดียิ่งขึ้น เนื่องจากในปัจจุบันได้มีการพัฒนาเทคโนโลยีอยู่ตลอดเวลา จึงเป็นความท้าทายในการติดตามเทคโนโลยีสมัยใหม่ เพื่อนำมาปรับปรุงระบบให้ทันสมัย และสามารถรองรับการใช้งานรูปแบบใหม่ๆ ในปัจจุบันได้ นอกจากนี้เทคโนโลยีสมัยใหม่ ยังสามารถเพิ่มขีดจำกัดให้กับระบบได้อีกด้วย

งานวิจัยนี้ ได้ให้ความสนใจในเทคโนโลยีโนดด็อทเจเอส เนื่องจากจากโครงสร้างของโนดด็อทเจเอสมีการทำงานที่รวดเร็ว ทั้งยังมีความสามารถอื่นๆ ที่หลากหลาย และน่าจะมีความเหมาะสมกับการใช้งานในปัจจุบัน

ในขั้นตอนนี้จึงเป็นการตั้งสมมติฐานว่า ความสามารถของการเขียนโปรแกรมภาษาโนดด็อทเจเอสนั้น สามารถนำมาพัฒนาและปรับปรุงระบบเดิมได้

3.2.3 ทำการออกแบบระบบโดยประยุกต์ใช้เทคโนโลยีสมัยใหม่

ในการออกแบบระบบ จะต้องมีการวางแผนปรับปรุงโครงสร้างของโปรแกรม ให้มีการทำงานที่มีประสิทธิภาพสูงขึ้น เพิ่มขีดจำกัดในการทำงานของระบบจากเดิมให้สูงขึ้น โดยการทำเทคโนโลยีสมัยใหม่เข้ามาประยุกต์ใช้งาน ดังนั้น ในขั้นตอนนี้จึงมีความสำคัญเป็นอย่างมาก เนื่องจากระบบจะสำเร็จหรือไม่นั้น ขึ้นอยู่กับการออกแบบตั้งแต่เริ่มแรก

3.2.4 พัฒนาระบบโดยใช้เทคโนโลยีสมัยใหม่

เมื่อทำการศึกษาทั้งระบบเก่าและระบบใหม่จนเข้าใจอย่างถ่องแท้แล้วนั้น จึงนำความรู้ที่ได้มาประยุกต์ใช้งาน โดยการปรับเปลี่ยนโครงสร้างของระบบให้เป็นเทคโนโลยีโนดด็อทเจเอส

และอาศัยหลักการทำงานของเว็บซ็อคเก็ตในการรับ-ส่งข้อมูล ระหว่างฐานข้อมูลและเครื่องคอมพิวเตอร์ลูกข่าย

3.2.5 ทดสอบระบบที่ได้รับการพัฒนาและปรับปรุง

เมื่อทำการพัฒนาและปรับปรุงโปรแกรมด้วยการเขียนโปรแกรมภาษาโนดด็อทเจเอสแล้ว จึงทำการทดสอบการใช้งานตามปัจจัยต่างๆ เช่น ความเร็วในการรับ-ส่งข้อมูล ระหว่างเครื่องคอมพิวเตอร์ลูกข่ายและฐานข้อมูล เป็นต้น และนำข้อมูลที่ได้มาเปรียบเทียบกับโปรแกรมแบบเก่าในหัวข้อต่างๆ กัน เช่น ปริมาณการร้องขอข้อมูล (Request) การรับ-ส่งข้อมูล (Data Transfer) การใช้ทรัพยากรทางคอมพิวเตอร์ของทั้งฝั่งเครื่องคอมพิวเตอร์แม่ข่าย และเครื่องคอมพิวเตอร์ลูกข่าย เป็นต้น

3.2.6 เปรียบเทียบประสิทธิภาพการทำงานของระบบเก่าและใหม่

ทำการเปรียบเทียบการทำงานระหว่างโปรแกรมแบบเก่า และแบบใหม่ โดยใช้เงื่อนไขเดียวกัน เพื่อหาขีดจำกัด เช่น การกำหนดปริมาณข้อมูลที่ต้องการจากฐานข้อมูล การกำหนดคาบเวลาในการรับ-ส่งข้อมูลจากฐานข้อมูล แล้วทำการเปรียบเทียบการใช้งานทรัพยากรทางคอมพิวเตอร์ต่างๆ ได้แก่ การทำงานซีพียูของทั้งแม่ข่ายและลูกข่าย ปริมาณการร้องขอ และเข้าถึงฐานข้อมูล รวมถึงปริมาณข้อมูลที่ส่งผ่านจากฐานข้อมูลสู่เว็บเบราว์เซอร์อีกด้วย

3.2.7 วิเคราะห์และสรุปผลการทำงานของระบบเก่าและใหม่ เพื่อหาข้อสรุป

นำข้อมูลที่รวบรวมมาได้ทั้งหมดมาวิเคราะห์หาข้อสรุป ข้อดี และข้อเสีย ของระบบเก่าและระบบใหม่ เพื่อเป็นประโยชน์ในการวิจัยต่อไป

3.2.8 นำข้อมูลที่วิเคราะห์ได้มาพัฒนาโปรแกรมให้มีความเหมาะสมกับการใช้งาน

เพื่อให้โปรแกรมมีความเหมาะสมกับสภาพของการใช้งานจริง โปรแกรมจึงควรมีการปรับเปลี่ยนการทำงานให้เหมาะสมกับสภาพการใช้งานปัจจุบันแบบอัตโนมัติ โดยมุ่งเน้นไปในเรื่องการปรับปรุงฟังก์ชันการทำงานที่เครื่องคอมพิวเตอร์แม่ข่าย เนื่องจากสามารถควบคุมได้ง่าย เพราะเป็นตัวกำหนดการทำงานของโปรแกรม และไม่เป็นการเพิ่มภาระให้แก่เครื่องคอมพิวเตอร์ลูกข่ายอีกด้วย

ตารางที่ 3.1 แผนการดำเนินงาน

	หัวข้อ	ช่วงเดือน / ปี (พ.ศ.)						
		ม.ค. 2558	ก.พ. 2558	มี.ค. 2558	ก.ย. 2558	ธ.ค. 2558	ม.ค. 2559	ก.พ. 2559
1	ศึกษาหลักการทำงานของโปรแกรมตรวจสอบแรงดันน้ำในท่อประธาน (แบบเก่า)							
2	ศึกษาเทคโนโลยีสมัยใหม่เพื่อนำมาทดแทน							
3	ทำการออกแบบระบบโดยใช้เทคโนโลยีสมัยใหม่							
4	พัฒนาระบบโดยใช้เทคโนโลยีสมัยใหม่							
5	ทดสอบระบบที่ได้รับการปรับปรุงด้วยเทคโนโลยีสมัยใหม่							
6	เปรียบเทียบประสิทธิภาพการทำงานของระบบเก่าและใหม่							
7	วิเคราะห์การทำงานของระบบเก่าและใหม่ เพื่อหาข้อสรุป							
8	ปรับปรุงฟังก์ชันการทำงานที่เครื่องแม่ข่ายให้เหมาะสมกับการใช้งานแบบอัตโนมัติ							

3.3 การออกแบบและพัฒนาระบบ

ในการออกแบบและพัฒนาระบบ มีขั้นตอนในการดำเนินงานตามขั้นตอนที่ได้กล่าวไว้ข้างต้น โดยมีรายละเอียดในการดำเนินงาน ดังนี้

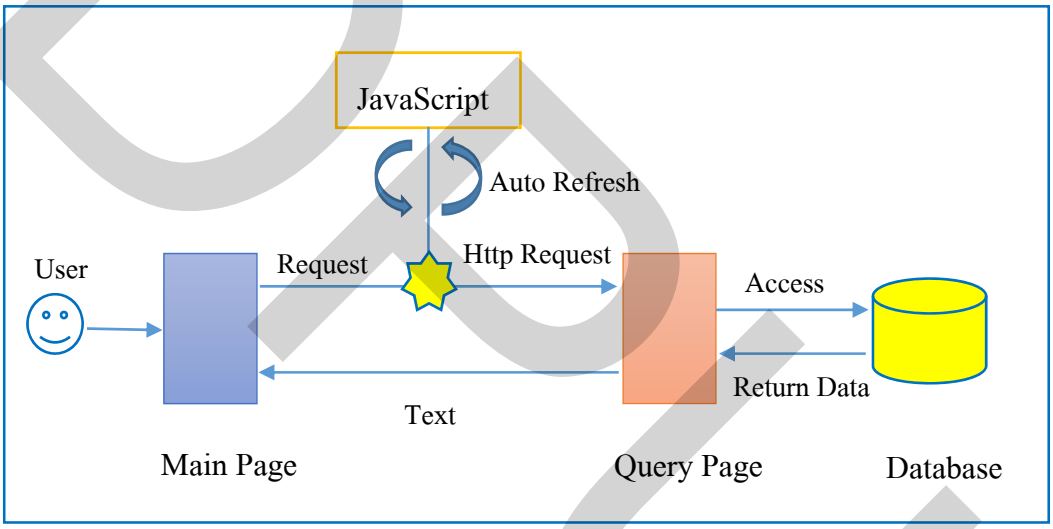
3.3.1 ปัญหาในการใช้งานโปรแกรมตรวจสอบแรงดันน้ำในท่อประธานแบบเดิม

ในการใช้งานโปรแกรมตรวจสอบแรงดันน้ำในท่อประธานทานแบบเก่านั้น จะมีปัญหาหลักๆ อยู่ในเรื่องของการวนลูปรับข้อมูลจากฐานข้อมูล กล่าวคือ โปรแกรมแบบเดิมนั้นเป็นการใช้งานการเขียนโปรแกรมด้วยภาษาจาวาสคริปต์ (JavaScript) ซึ่งมีขนาดใหญ่และซับซ้อน จึงมีผล

ทำให้เกิดการใช้ทรัพยากรทางคอมพิวเตอร์จำนวนมาก ในการรับข้อมูลแบบเวลาจริงทำให้บางครั้งในการใช้งานจะเกิดการล่มของเว็บเบราว์เซอร์ ซึ่งส่งผลเสียในการใช้งานระยะยาว

ปัญหาอีกอย่างก็คือ ความเร็วของคาบเวลา (Interval time) ของการรับข้อมูลจากฐานข้อมูล ถ้ามีการตั้งคาบเวลาไว้น้อยจนเกินไป จะทำให้เว็บเบราว์เซอร์ฝั่งเครื่องคอมพิวเตอร์ลูกข่ายนั้นล่ม รวมถึงปริมาณการร้องขอข้อมูล (Request) ไปยังเครื่องคอมพิวเตอร์แม่ข่ายมีปริมาณมากจนเกินไป หากมีการใช้งานจากหลายๆ เครื่อง ก็จะส่งผลให้เครื่องแม่ข่ายล่มได้เช่นกัน

ดังนั้น ในการทำการวิจัยนี้ ผู้วิจัยจึงเน้นไปในเรื่องของการปรับปรุงโครงสร้างการเขียนโปรแกรมจากเดิมที่เป็นภาษาจาวาสคริปต์ ให้เป็นแบบอื่น ที่สามารถแก้ปัญหาที่กล่าวมาเบื้องต้นได้



ภาพที่ 3.3 หลักการทำงานของภาษาจาวาสคริปต์

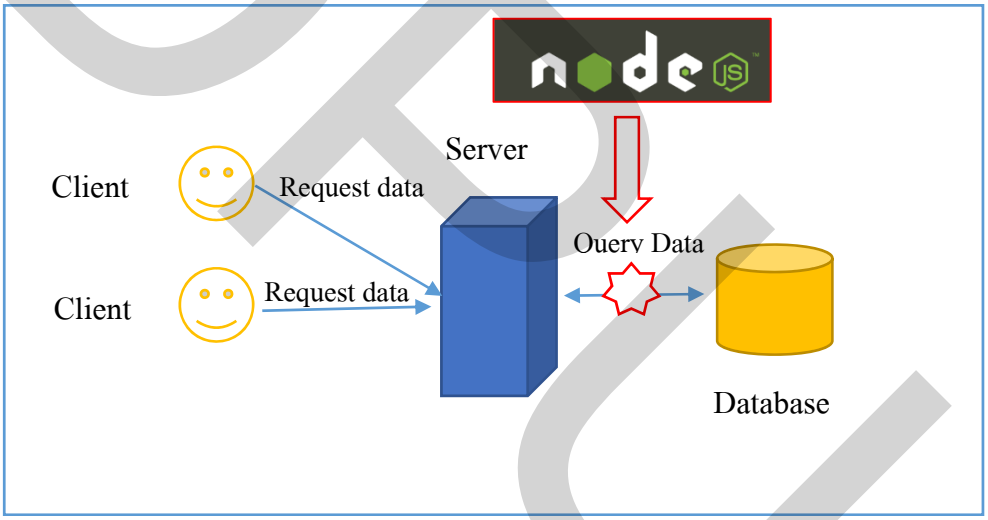
จากภาพที่ 3.3 แสดงให้เห็นถึงหลักการทำงานของภาษาจาวาสคริปต์ เมื่อมีการร้องขอข้อมูลของผู้ใช้งาน จากหน้าเว็บเบราว์เซอร์หลัก (Main Page) จาวาสคริปต์ที่ทำงานเป็นส่วนเปลี่ยนแปลงข้อมูลแบบอัตโนมัติ จะทำการเรียกข้อมูลส่วนที่มีหน้าที่ติดต่อกับฐานข้อมูล (Query Page) เพื่อทำการเข้าถึงฐานข้อมูล เรียกว่า เอชทีทีพีรีเควส (HTTP Request) โดยที่จาวาสคริปต์ในส่วนนี้จะมีการกำหนดคาบเวลา (Interval Time) ในการเรียกข้อมูลอย่างต่อเนื่อง ซึ่งจะทำให้เกิดการเข้าถึงฐานข้อมูลเป็นจำนวนครั้งตามที่กำหนดเวลาไว้

ส่วนที่ติดต่อกับฐานข้อมูล จะทำการดึงข้อมูลจากฐานข้อมูล แล้วส่งค่ากลับ (Return Data) โดยอยู่ในรูปของตัวอักษร (Text) เพื่อนำกลับไปแสดงผลที่หน้าเว็บเบราว์เซอร์หลักเพื่อให้ผู้ใช้งานรับทราบข้อมูลต่อไป

จากการทำงานลักษณะดังกล่าว จะทำให้เกิดการเข้าถึงฐานข้อมูลที่มีปริมาณมาก หากมีการตั้งคาบเวลาในการรับข้อมูลน้อยเกินไป ซึ่งเป็นเหตุในการล่มของเครื่องคอมพิวเตอร์แม่ข่าย รวมถึงมีการส่งผ่านข้อมูลที่มีปริมาณมาก เนื่องจากเป็นการรับข้อมูลจากส่วนที่ติดต่อกับฐานข้อมูลทั้งหมด ถ้าหากมีการรับข้อมูลปริมาณมาก ๆ ก็จะทำให้เว็บเบราว์เซอร์ของเครื่องคอมพิวเตอร์ฝั่งลูกข่ายล่มได้ด้วยเช่นกัน

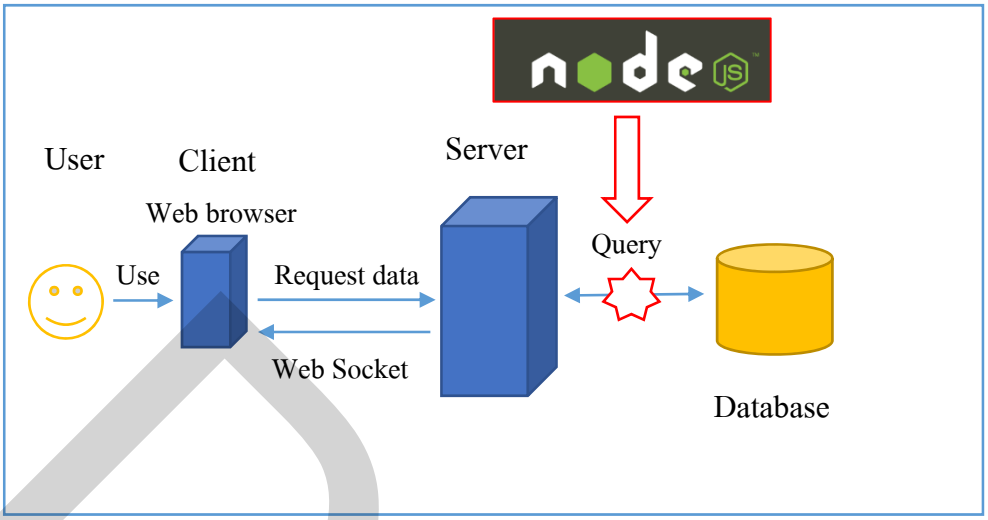
3.3.2 เทคโนโลยีสมัยใหม่ที่สามารถนำมาใช้ในการปรับปรุง

ผู้วิจัยได้ทำการศึกษาการเขียนโปรแกรมภาษาโนดด็อทเจเอสที่สามารถแก้ไขปัญหาปริมาณการร้องขอข้อมูล (Request) ที่มีจำนวนมากเกินไปของการใช้งานภาษาจาวาสคริปต์ โดยการนำโนดด็อทเจเอส มาเป็นส่วนเครื่องจักร (Engine) ในการสร้างช่องทางส่งผ่านข้อมูลจากฐานข้อมูล ดังภาพที่ 3.4



ภาพที่ 3.4 แนวคิดการทำการโดยใช้การเขียนโปรแกรมภาษาโนดด็อทเจเอส

โนดด็อทเจเอสจะเป็นส่วนที่ทำหน้าที่ในการเชื่อมต่อระหว่าง เครื่องคอมพิวเตอร์แม่ข่ายและฐานข้อมูล โดยที่โนดด็อทเจเอสจะทำการวนลูป (Looping) เพื่อรับ-ส่งข้อมูลจากฐานข้อมูลไปยังเครื่องคอมพิวเตอร์แม่ข่าย และเครื่องคอมพิวเตอร์แม่ข่ายจะส่งข้อมูลที่ได้ผ่านเว็บซ็อคเก็ตไปยังเว็บเบราว์เซอร์ของเครื่องคอมพิวเตอร์ลูกข่าย ที่มีผู้ใช้งานร้องขอข้อมูลอยู่ ดังภาพที่ 3.5

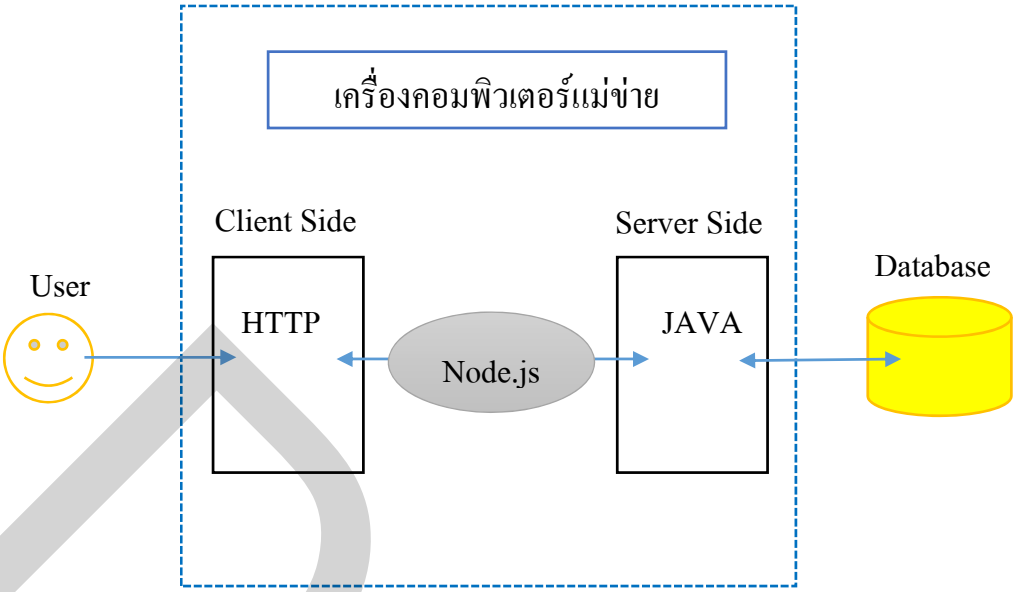


ภาพที่ 3.5 หลักการทำงานของการเขียนโปรแกรมภาษาโนดด็อทเจเอส

การทำงานลักษณะนี้ จะมีการส่งข้อมูลเฉพาะส่วนที่เป็นข้อมูลผ่านทางเว็บซ็อคเก็ตเท่านั้น ซึ่งแตกต่างกับการส่งข้อมูลของภาษาจาวาสคริปต์ในส่วนที่ติดต่อกับฐานข้อมูล (Query Page) อย่างสิ้นเชิง ซึ่งปริมาณการส่งผ่านข้อมูลของโนดด็อทเจเอสจะน้อยกว่า และเนื่องจากเป็นการรับ-ส่งข้อมูลผ่านทางเว็บซ็อคเก็ตที่เครื่องคอมพิวเตอร์แม่ข่ายสร้างขึ้นจนกว่าจะมีการปิดหน้าเว็บเบราว์เซอร์นั้นๆ จึงมีการร้องขอ (Request) และเข้าถึง (Access) ฐานข้อมูลเพียงครั้งแรกครั้งเดียว วิธีนี้จึงช่วยลดการล่มของเครื่องคอมพิวเตอร์แม่ข่าย อันเนื่องมาจากปริมาณการร้องขอที่มีจำนวนมากได้

3.3.3 การออกแบบระบบโดยใช้โครงสร้างการทำงานแบบใหม่

การนำการเขียนโปรแกรมภาษาโนดด็อทเจเอสมาประยุกต์ใช้ในการเชื่อมต่อกับฐานข้อมูลนั้น จะแบ่งออกเป็น 2 ส่วน หลัก คือ ส่วนการเขียนโปรแกรมฝั่งแม่ข่าย และส่วนการเขียนโปรแกรมฝั่งลูกข่าย ดังภาพที่ 3.6

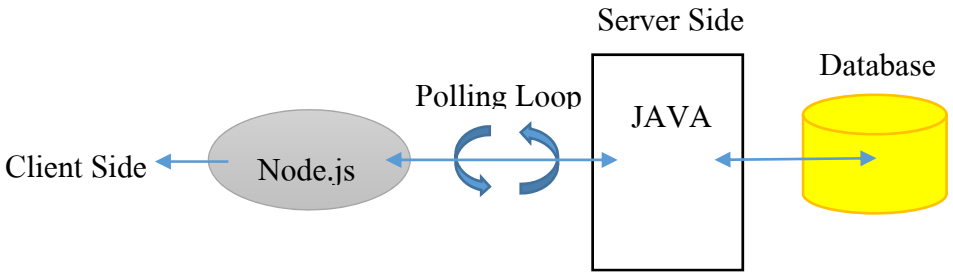


ภาพที่ 3.6 ส่วนของการเขียนโปรแกรมหลักของการเขียนโปรแกรมภาษาโนดด็อทเจเอส

3.3.3.1 ส่วนการเขียนโปรแกรมฝั่งแม่ข่าย (Server Side Code)

เป็นส่วนที่มีหน้าที่ในการเข้าถึงข้อมูลที่อยู่ภายในฐานข้อมูล ในส่วนนี้จะเป็นการเขียนโปรแกรมด้วยภาษาจาวา (JAVA Language) เพื่อทำการเรียกข้อมูล (Query) จากฐานข้อมูล โดยโนดด็อทเจเอสจะทำหน้าสร้างเว็บซ็อคเก็ต แล้วส่งข้อมูลไปยังส่วนการเขียนโปรแกรมฝั่งลูกข่าย (Client Side) ต่อไป

การทำงานของโนดด็อทเจเอส จะทำการเรียกข้อมูลจากฐานข้อมูลตามคาบเวลา (Polling Loop) ที่กำหนดไว้ เพื่อส่งค่าล่าสุดของฐานข้อมูลที่เราทำการเรียกไปแสดงผล ดังภาพที่ 3.7

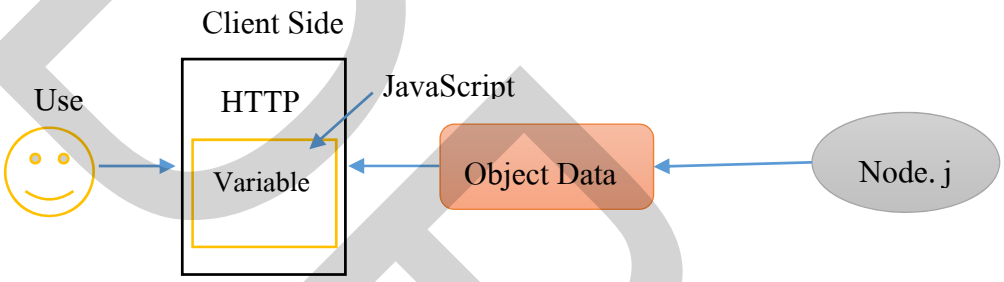


ภาพที่ 3.7 การเรียกข้อมูลจากฐานข้อมูลตามคาบเวลา (Polling Loop) ของโนดด็อทเจเอส

การกำหนดค่าของคาบเวลา จะส่งผลโดยตรงต่อการทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่าย จึงต้องมีการกำหนดให้เหมาะสมกับทรัพยากรทางคอมพิวเตอร์ที่มีอยู่ด้วย

3.3.3.2 ส่วนการเขียนโปรแกรมฝั่งลูกข่าย (Client Side Code)

เป็นส่วนที่มีหน้าที่ในการแสดงผลข้อมูลให้กับผู้ใช้งาน เช่นเดียวกับการแสดงผลเว็บเบราว์เซอร์ทั่วไป แต่ที่พิเศษไปจากเดิมคือ ต้องมีการเขียนส่วนการรับข้อมูลที่รับมาจากโนดด็อทเจเอส ผ่านเว็บซ็อคเก็ต เพื่อรับข้อมูลที่ได้ทำการเรียกมาจากฐานข้อมูล ซึ่งโดยปกติจะอยู่ในรูปของข้อมูลเชิงวัตถุ (Object Data) มาแปลงเป็นค่าข้อมูลแล้วใส่ในตัวแปร (Variable) ของภาษาจาวาสคริปต์ (JavaScript) เพื่อนำไปแสดงผลหน้าเว็บเบราว์เซอร์ตามปกติ ดังภาพที่ 3.8



ภาพที่ 3.8 การรับค่าตัวแปรของการเขียนโปรแกรมฝั่งลูกข่าย

3.3.4 การพัฒนาระบบโดยใช้โครงสร้างการเขียนโปรแกรมภาษาโนดด็อทเจเอส

การพัฒนาโปรแกรม จะมีขั้นหลักๆ ดังนี้

1. การติดตั้งโปรแกรมที่เกี่ยวข้องกับการพัฒนาโปรแกรม ดังนี้

1.1 โนดด็อทเจเอส 64 บิท สำหรับระบบปฏิบัติการวินโดว์ (Node.js for Windows x64) สามารถดาวน์โหลดได้ที่หน้าเว็บไซต์หลักของโนดด็อทเจเอส (“https://nodejs.org/en/”) ภายหลังการดาวน์โหลด ให้ทำการติดตั้งไว้ที่เครื่องคอมพิวเตอร์แม่ข่าย เพื่อใช้เป็นส่วนหลักในการเขียนโปรแกรม

1.2 ทำการติดตั้งส่วนขยายเอ็กซ์เพลส (Express) ของโนดด็อทเจเอส ดังภาพที่ 3.9

```
$ npm install express --save
```

ภาพที่ 3.9 คำสั่งการติดตั้งส่วนขยายเอ็กซ์เพลสที่ Command Prompt

ส่วนขยายเอ็กซ์เพลส (Express) คือ ไลบรารี่ของโนดด็อทเจเอส ที่มีหน้าที่ในการสร้างลูปการรับส่งข้อมูลตามคาบเวลา เพื่อให้มีการเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบัน

1.3 ทำการติดตั้งฐานข้อมูลมายเอสคิวแอล (MySQL) ที่เครื่องคอมพิวเตอร์แม่ข่ายเพื่อใช้ในการเก็บข้อมูล (Database)

1.4 ทำการติดตั้งอะแพชี (Apache) ที่เครื่องคอมพิวเตอร์แม่ข่าย เพื่อให้เครื่องคอมพิวเตอร์แม่ข่ายสามารถเป็นเว็บเซอร์เวอร์ได้ (Web Server) และทำการกำหนดที่จัดเก็บไฟล์ (File Directory) เพื่อใช้สำหรับการเข้าถึงเว็บเซอร์เวอร์ของเครื่องคอมพิวเตอร์ลูกข่าย

2. การเขียนโปรแกรมฝั่งแม่ข่าย (Server Side Structure)

ในการเขียนโปรแกรมฝั่งแม่ข่ายนั้น สำหรับโนดด็อทเจเอส จะใช้ภาษาจาวา (JavaScript) ในการรันโปรแกรม โดยจะต้องทำการสร้างไฟล์นามสกุลวาจาสคริปต์ไว้ที่ตำแหน่งจัดเก็บไฟล์ เช่น server.js และใช้ดอส (DOS) หรือคอมมานด์พรอมต์ (Command Prompt) ในการเรียกไฟล์ที่ทำการสร้างไว้ เพื่อเริ่มการทำการของโปรแกรม

โครงสร้างของภาษาโนดด็อทเจเอสนั้น ประกอบไปด้วยส่วนหลักๆ ดังต่อไปนี้

2.1 ส่วนเชื่อมต่อกับฐานข้อมูล

เป็นส่วนที่ใช้ทำการรับ - ส่งข้อมูล ระหว่างโนดด็อทเจเอสและฐานข้อมูลมายเอสคิวแอล ที่ได้ทำการสร้างไว้ ดังภาพที่ 3.10

```
var app = require('http').createServer(handler),
  io = require('socket.io').listen(app),
  fs = require('fs'),
  mysql = require('mysql'),
  connectionsArray = [],
  connection = mysql.createConnection({
    multipleStatements: true,
    host: 'localhost',
    user: 'root',
    password: 'mwa',
    database: 'nodejs',
    port: 3306
  }),
```

ภาพที่ 3.10 สคริปต์ส่วนติดต่อกับฐานข้อมูลมายเอสคิวแอล

2.2 ส่วนกำหนดคาบเวลาในการรับ – ส่งข้อมูล

ส่วนนี้จะเป็นส่วนสำคัญที่จะทำให้การเปลี่ยนแปลงข้อมูลที่เกิดขึ้นที่ฐานข้อมูล สามารถแสดงผลได้อย่างรวดเร็วที่ฝั่งผู้ใช้งาน หรือเรียกว่าเป็นค่าเวลาจริง ดังภาพที่ 3.11

```
var polling = 1000;

  POLLING_INTERVAL = polling,    //millisec
  pollingTimer;
```

ภาพที่ 3.11 สคริปต์ส่วนกำหนดคาบเวลาในการรับ-ส่งข้อมูล

2.3 ส่วน Query Data จากฐานข้อมูล

เป็นส่วนที่ระบุคำสั่งเอสคิวแอล (SQL) ที่ต้องการ เพื่อใช้ในการเรียกดู (Query) ฐานข้อมูลที่ต้องการนำมาแสดง ดังภาพที่ 3.12

```
var comm = new Array(
                        "SELECT * FROM rtu_1 ORDER BY id DESC LIMIT 200;"
                        );
   var i=1;
   var all_comm = comm[0];      //First command in array comm
   while(i<=comm.length-1){     //Next command in array comm
       all_comm += comm[i];
       i++;
       }
```

ภาพที่ 3.12 สคริปต์คำสั่งเรียกดูข้อมูลจากฐานข้อมูล

2.4 ส่วนสร้างลูปของช่องทางรับ-ส่งข้อมูล (Polling Loop Socket)

เป็นส่วนที่มีหน้าที่ในการตรวจสอบผลลัพธ์ของการเชื่อมต่อกับฐานข้อมูล รวมถึงการนำค่าที่ได้จากฐานข้อมูลมาเก็บไว้ในตัวแปร เพื่อเตรียมการเชื่อมต่อกับซ็อคเก็ต

นอกจากนี้ ส่วนนี้จะเป็นส่วนที่ใช้สำหรับการค้นหาซ็อคเก็ตหรือช่องทางที่ยังว่างของเครื่องแม่ข่าย เพื่อนำไปใช้งาน และจะมีการวนลูปเพื่อหาซ็อคเก็ตใหม่ๆ ในกรณีที่เกิดการเต็มของช่องทาง ดังภาพที่ 3.13

```
var pollingLoop = function() {
  var query = connection.query(all_comm),set_of_data = [];
  query
    .on('error', function(err) {
      console.log(err);
      updateSockets(err);
    })
    .on('result', function(x) {
      set_of_data.push(x);
    })
    .on('end', function() {
      if (connectionsArray.length) {

        pollingTimer = setTimeout(pollingLoop, POLLING_INTERVAL);
        updateSockets({set_of_data: set_of_data});

      } else {
        console.log('No more socket connections')
      }


    });

};
```

ภาพที่ 3.13 สคริปต์การตรวจสอบผลการเชื่อมต่อฐานข้อมูลและสร้างซ็อคเก็ต

2.5 ส่วนส่งผ่านข้อมูลเข้าสู่ซ็อคเก็ต (Transfer Data to Socket)

หลังจากได้ข้อมูลที่ต้องการจากฐานข้อมูลแล้ว ก็ทำการส่งข้อมูลผ่านซ็อคเก็ตเพื่อให้ข้อมูลไปแสดงผลที่ฝั่งผู้ใช้งานอย่างต่อเนื่อง โดยมีสคริปต์ดังภาพที่ 3.14 และภาพที่ 3.15

```
io.sockets.on('connection', function(socket) {
  console.log('Number of connections:' + connectionsArray.length);
  if (!connectionsArray.length) {
    pollingLoop();
  }

  socket.on('disconnect', function() {
    var socketIndex = connectionsArray.indexOf(socket);
    console.log('socketID = %s got disconnected', socketIndex);
    if (~socketIndex) {
      connectionsArray.splice(socketIndex, 1);
    }
  });

  console.log('A new socket is connected!');
  connectionsArray.push(socket);

});
```

ภาพที่ 3.14 สคริปต์การกำหนดซ็อคเก็ตเพื่อส่งข้อมูลไปยังผู้ใช้งาน

```
var updateSockets = function(data) {
  data.time = new Date();
  connectionsArray.forEach(function(tmpSocket) {
      tmpSocket.volatile.emit('notification', data);
  });
```

ภาพที่ 3.15 การอัพเดทซ็อคเก็ต เพื่อส่งข้อมูลไปยังผู้ใช้งานอย่างต่อเนื่อง

นอกจากนี้ ยังต้องมีการกำหนดพอร์ตในการส่งข้อมูลระหว่างฝั่งแม่ข่ายและฝั่งลูกข่าย อีกด้วย ดังภาพที่ 3.16

```
app.listen(8000);
```

ภาพที่ 3.16 การกำหนดพอร์ตรับ-ส่งข้อมูลระหว่างฝั่งแม่ข่ายและฝั่งลูกข่าย

3. การเขียนโปรแกรมฝั่งลูกข่าย (Client Side Structure)

การเขียนโปรแกรมฝั่งลูกข่าย จะเป็นการเขียนโปรแกรมในลักษณะการแสดงผลในรูปแบบเว็บไซต์ ซึ่งมีการใช้ภาษาเอชทีเอ็มแอล (HTML) ร่วมกับการใช้งานเว็บซ็อคเก็ตเพื่อรับข้อมูลจากโนดด็อทเจเอส ที่ได้มีการเขียนโปรแกรมเอาไว้ในส่วนของฝั่งแม่ข่ายแล้ว

ส่วนหลักๆ ของการแสดงผลข้อมูลทางฝั่งลูกข่าย มีดังนี้

3.1 การเรียกใช้งานไฟล์ที่จำเป็น

การเรียกใช้งานไฟล์ที่จำเป็นต่อการทำงานของโนดด็อทเจเอส จะทำการเขียนไว้ในส่วนเฮด (Head) ของหน้าเว็บ ดังภาพที่ 3.17

```
<!doctype html>
<html>
    <head>
    <meta charset="UTF-8">
    <script src="node_modules/socket.io/node_modules/socket.io-client/socket.io.js"></script>
    <script src="http://localhost/thesis/jquery-2.1.3.min.js"></script>
```

ภาพที่ 3.17 การเรียกใช้ไฟล์ที่จำเป็นของโนดด็อทเจเอส

จากภาพที่ 3.17 จะเห็นได้ว่า การใช้งานเว็บซ็อคเก็ตของโนดด็อทเจเอสนั้น จะต้องมีการเรียกใช้งานไฟล์นามสกุลจาวาสคริปต์ (socket.io.js) ก่อนเสมอ จึงจะสามารถใช้งานเว็บซ็อคเก็ตได้

3.2 การกำหนดพอร์ต (Port) เพื่อเตรียมการเชื่อมต่อกับเว็บซ็อคเก็ต

ในส่วนของการกำหนดพอร์ตเพื่อรอรับข้อมูลจากเว็บซ็อคเก็ตนั้น เลขพอร์ตจะตรงกับส่วนของการเขียนโปรแกรมฝั่งแม่ข่าย (ภาพที่ 3.17) ดังภาพที่ 3.18

```
var data_limit=100;
var socket = io.connect('http://localhost:8000');
```

ภาพที่ 3.18 การกำหนดพอร์ต (Port) เพื่อเตรียมการเชื่อมต่อกับเว็บซ็อคเก็ต

3.3 การสร้างฟังก์ชัน (Function) รับค่าจากซ็อคเก็ตและแสดงผลลัพธ์

ฟังก์ชันรับค่าจากซ็อคเก็ตจะประกอบไปด้วยส่วนหลักๆ ดังนี้

ส่วนการกำหนดตัวแปร (Variable) ดังภาพที่ 3.19

```
var data_p = new Array();
var data_f = new Array();
var data_ptc = new Array();
var data_ptc_w = new Array();
var d1=1;
var d2=1;
var d3=1;
var d4=1;
var limit_data=1;
var rtu_list = "rtu";
```

ภาพที่ 3.19 การกำหนดตัวแปรที่จำเป็นในการรับค่าจากซ็อคเก็ต

ส่วนรับค่าจากซ็อคเก็ตเก็บในตัวแปร ดังภาพที่ 3.20

```
var pr="1";
while(pr<=data_limit){
    data_p[pr] = new Array();
    var d1="1";
$.each(data.set_of_data,function(index,set_of_data){
    if(set_of_data["rtu"+pr] != undefined){
    data_p[pr][d1] = set_of_data["rtu"+pr];
    d1++;
    }
});
pr++;
}

var fl="1";
while(fl<=data_limit){
    data_f[fl] = new Array();
    var d2="1";
$.each(data.set_of_data,function(index,set_of_data){
    if(set_of_data["rtu"+fl+"F"] != undefined){
    data_f[fl][d2] = set_of_data["rtu"+fl+"F"];
    d2++;
    }
});
fl++;
}
```

ภาพที่ 3.20 การสร้างลูป รับค่าที่ได้จากซ็อคเก็ตเพื่อเก็บไว้ในตัวแปร

ส่วนแสดงผลข้อมูล

แบ่งออกเป็น 2 ส่วน คือ ส่วนภาษาเอชทีเอ็มแอล และส่วนภาษาจาวาสคริปต์

ส่วนภาษาเอชทีเอ็มแอล คือส่วนที่เป็นโครงสร้างเพื่อใช้แสดงผลที่หน้าเว็บไซต์ โดยจะใช้เป็นตัวกำหนดว่า หน้าเว็บไซต์จะมีรูปร่างอย่างไร และต้องมีการใช้งานคู่กับส่วนภาษาจาวาสคริปต์ เพื่อให้เว็บไซต์มีลูกเล่น หรือมีการเปลี่ยนแปลงข้อมูลได้ตรงตามที่ต้องการใช้งาน ดังภาพที่ 3.21

```
<div id="table1"></div>

<script>
    var tab=1;
    while(tab<=data_limit){
        document.getElementById("table1").innerHTML += "<div id='container"+tab+"'>Loading "+tab+"...</div>";
        tab++;
    }

</script>
```

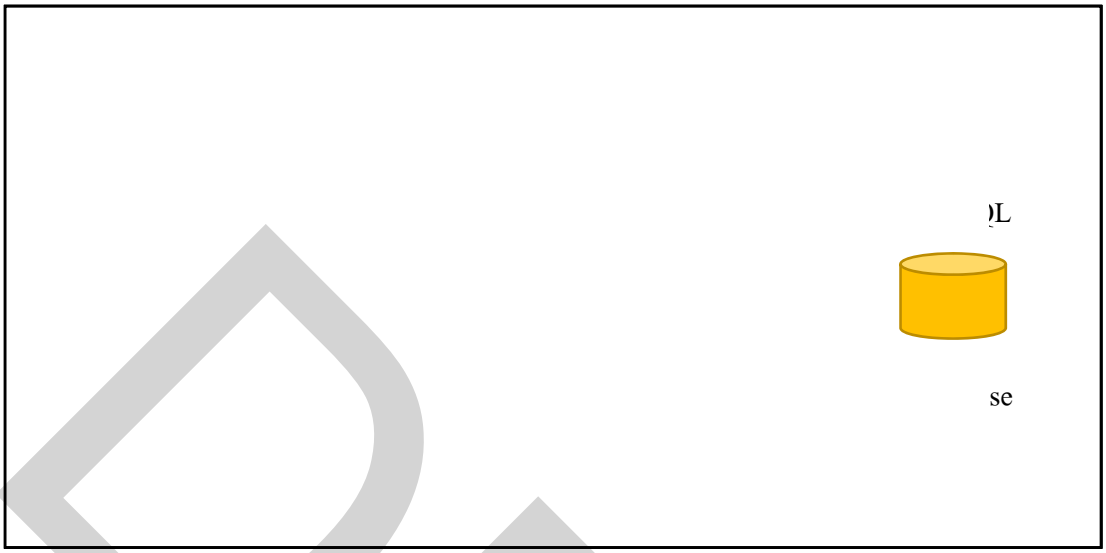
ภาพที่ 3.21 ส่วนการเขียนภาษาเอชทีเอ็มแอล

ส่วนภาษาจาวาสคริปต์ คือส่วนที่มีหน้าที่ในการเปลี่ยนแปลงค่าของข้อมูล ที่ได้รับมาจากฐานข้อมูลผ่านเว็บซ็อตเก็ต เพื่อให้แสดงผลข้อมูลแบบเวลาจริง โดยจะนำไปแสดงในส่วนที่เป็นภาษาเอชทีเอ็มแอลที่ได้เขียนรอไว้แล้ว ดังภาพที่ 3.22

```
    var rrr=1;
    var eee=1;
    while(rrr<=data_limit){
        $('time').html('Last Update:' + data.time);
        $('#container'+rrr).html(data_p[rrr][1]+" : "+data_f[rrr][1]);
        rrr++;
        eee++;
        }
});
```

ภาพที่ 3.22 ส่วนการเขียนภาษาจาวาสคริปต์

3.3.5 การทดสอบระบบที่ได้รับการพัฒนาแล้ว



ภาพที่ 3.23 แผนภาพแสดงการติดตั้งฮาร์ดแวร์ที่ใช้ในแบบจำลอง

จากภาพที่ 3.23 ในการทดสอบระบบ จะมีการทดสอบด้วยเครื่องคอมพิวเตอร์จำนวน 2 เครื่อง ที่มีการเชื่อมต่ออยู่ภายในเครือข่ายเดียวกัน (Local Area Network: LAN) ดังนี้

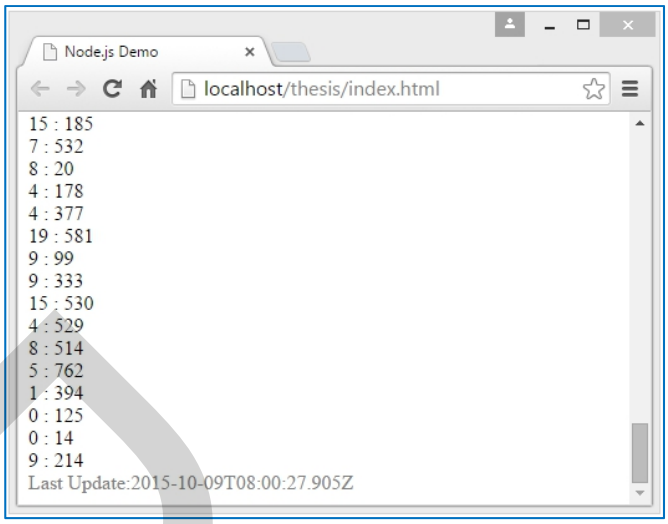
1. เครื่องคอมพิวเตอร์แม่ข่าย (Computer 1)

ใช้เป็นเครื่องแม่ข่ายสำหรับการติดตั้งฐานข้อมูลมายเอสคิวแอล และเว็บเซอร์เวอร์

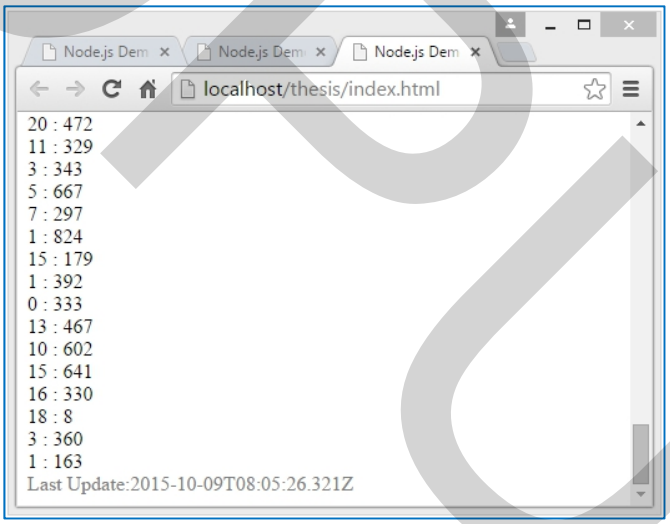
2. เครื่องคอมพิวเตอร์ลูกข่าย (Computer 2)

ใช้เป็นเครื่องสำหรับตรวจดูข้อมูลที่ได้รับมาจากฐานข้อมูล สำหรับการทดสอบนี้จะมีการติดตั้งเว็บเบราว์เซอร์กูเกิลโครม (Google Chrome) เพื่อการทดสอบ และทำการใช้แท็บ (Tab) ของเว็บเบราว์เซอร์ในการจำลองการเชื่อมต่อจากหลายๆ ผู้ใช้งาน

การแสดงผลข้อมูลที่เว็บเบราว์เซอร์ของฝั่งลูกข่าย จะแสดงผลข้อมูลที่ได้จากเว็บซ็อคเก็ตตามสคริปต์ที่เขียนไว้ในเว็บเบราว์เซอร์ โดยจะมีข้อมูลจำนวน 200 ข้อมูลที่แสดงผลพร้อมกันและเวลาล่าสุด (Last Update) เพื่อดูว่าไม่มีการค้างของข้อมูล ดังภาพที่ 3.24 และภาพที่ 3.25



ภาพที่ 3.24 เว็บเบราว์เซอร์ฝั่งผู้ใช้งาน แสดงผลข้อมูลที่รับจากเว็บซ็อคเก็ต



ภาพที่ 3.25 แสดงการใช้งานแท็บ (Tab) ของเบราว์เซอร์เพื่อจำลองการเชื่อมต่อจากหลายผู้ใช้งาน

เมื่อทำการติดตั้งระบบจนเสร็จสิ้นแล้ว จึงทำการรันโปรแกรมต่อเนื่องเป็นเวลา 24 ชั่วโมง เพื่อสังเกตการทำการของโปรแกรม และตรวจสอบการล่มหรือค้าง (Crash) ของโปรแกรม และเครื่องแม่ข่ายอีกด้วย

3.3.6 การทดสอบเปรียบเทียบประสิทธิภาพการทำงานของระบบ

หัวข้อในการเปรียบเทียบประสิทธิภาพของการทำงานของระบบทั้ง 2 แบบ (จาวาสคริปต์และโนดด็อทเจเอส) มีดังนี้

1. การเปรียบเทียบการใช้งานซีพียู (CPU Usage: %) ของจาวาสคริปต์และโนดด็อทเจเอส ที่เครื่องคอมพิวเตอร์แม่ข่าย และเครื่องคอมพิวเตอร์ลูกข่าย มีรายละเอียดดังนี้

1.1 การเปรียบเทียบการทำงานของทั้ง 2 ระบบ ที่ฝั่งแม่ข่าย

โดยมีการเปลี่ยนแปลงค่าของคาบเวลา (Interval Time) ต่างๆ กัน ดังตารางที่ 3.2

ตารางที่ 3.2 ตารางเก็บข้อมูลเปรียบเทียบการทำงานของจาวาสคริปต์และโนดด็อทเจเอสฝั่งแม่ข่าย

Structure	Interval Time (ms)										
	0	100	200	300	400	500	600	700	800	900	1000
Node.js	ทำการเก็บข้อมูลแต่ละคาบเวลา เพื่อนำมาเปรียบเทียบกัน										
JavaScript											

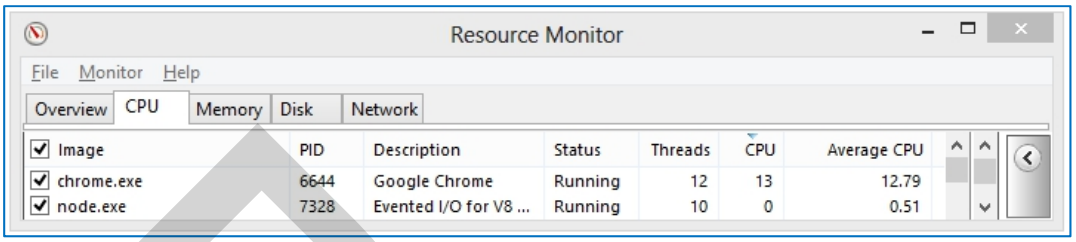
1.2 การเปรียบเทียบการทำงานของทั้ง 2 ระบบ ที่ฝั่งลูกข่าย

โดยมีการเปลี่ยนแปลงค่าของคาบเวลา (Interval Time) ต่างๆ กัน ดังตารางที่ 3.3

ตารางที่ 3.3 ตารางเก็บข้อมูลเปรียบเทียบการทำงานของจาวาสคริปต์และโนดด็อทเจเอสฝั่งแม่ข่าย

Structure	Interval Time (ms)										
	0	100	200	300	400	500	600	700	800	900	1000
Node.js	ทำการเก็บข้อมูลแต่ละคาบเวลา เพื่อนำมาเปรียบเทียบกัน										
JavaScript											

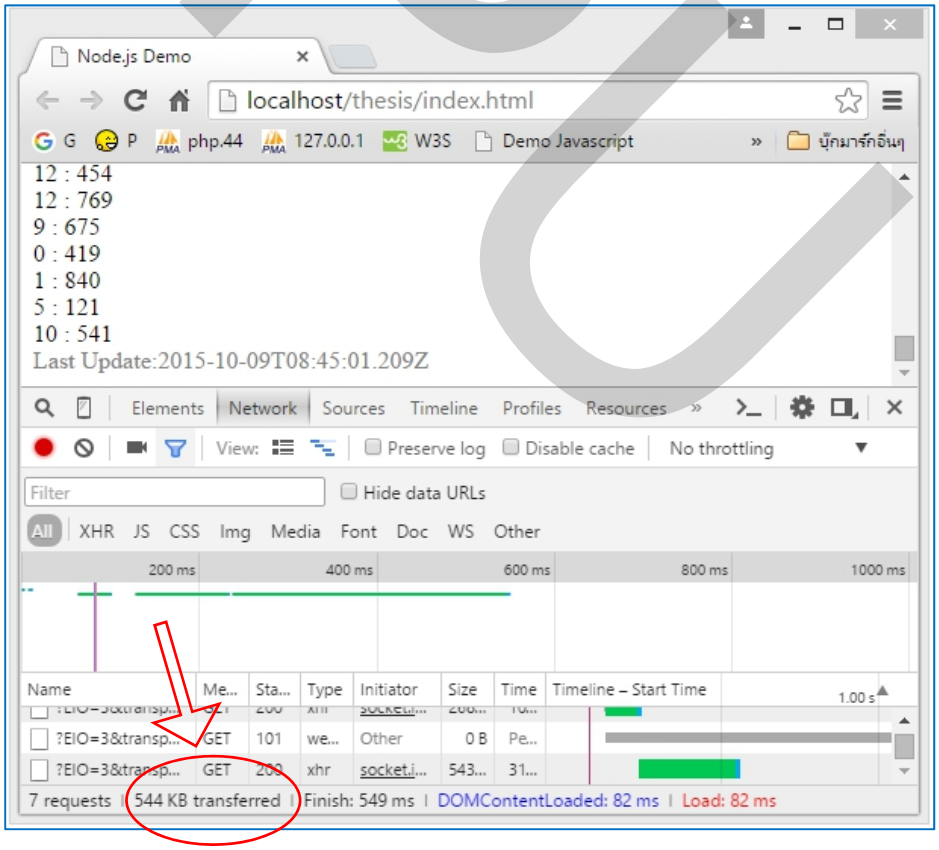
ในการเก็บข้อมูล จะใช้โปรแกรมทาสก์เมเนเจอร์ (Task Manager) และรีซอสมอนิเตอร์ (Resource Monitor) ภายในระบบปฏิบัติการวินโดว์ (OS Windows) ทำการบันทึกจำนวน 10 ครั้ง และหาค่าเฉลี่ย เพื่อนำไปสร้างกราฟความสัมพันธ์ ดังภาพที่ 3.26



ภาพที่ 3.26 การเก็บข้อมูลซีพียูจากโปรแกรมรีซอสมอนิเตอร์ (Resource Monitor)

2. การเปรียบเทียบการดาวน์โหลดข้อมูลภายในเว็บเบราว์เซอร์ (Data Transfer in Web Browser)

เป็นการเปรียบเทียบการส่งผ่านข้อมูลภายในเว็บเบราว์เซอร์ ระหว่างโนดด็อทเจเอส และจาวาสคริปต์ โดยการใช้ฟังก์ชันของกูเกิลโครมในการเก็บค่า ดังภาพที่ 3.27



ภาพที่ 3.27 การเก็บค่าการส่งผ่านข้อมูลภายในเว็บเบราว์เซอร์โดยกูเกิลโครม

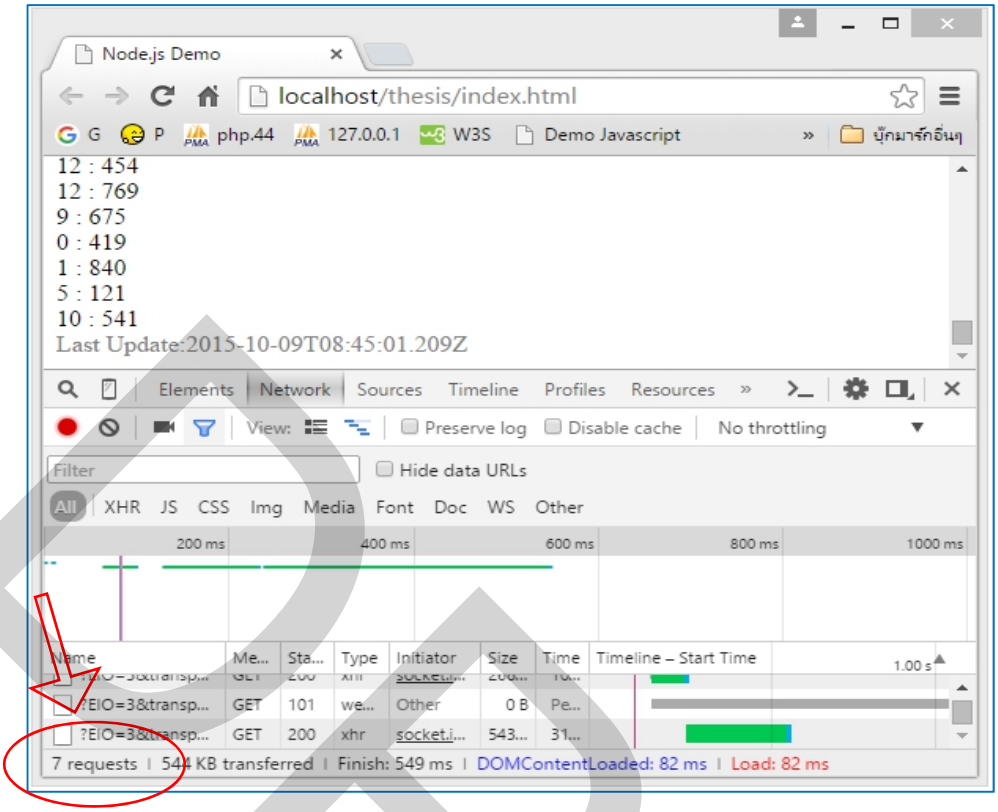
ในการทดสอบเก็บข้อมูล จะทำการตั้งค่าคาบเวลาการเปลี่ยนแปลงข้อมูล (Interval Time) ไว้คงที่ที่ 1000 ms และทำการเก็บค่าจำนวน 30 ค่า เพื่อสร้างกราฟแสดงความสัมพันธ์ ดังตารางที่ 3.4

ตารางที่ 3.4 ตารางเก็บข้อมูลเปรียบเทียบการส่งผ่านข้อมูลของจาวาสคริปต์และโนดด็อทเจเอสที่เว็บเบราว์เซอร์

Structure	Interval (times)						
	1	2	3	4	5	…	30
Node.js	ทำการเก็บข้อมูลแต่ละคาบเวลา เพื่อนำมาเปรียบเทียบกัน						
JavaScript							

3. การเปรียบเทียบปริมาณการร้องขอข้อมูล (Requests) ที่ฝั่งลูกข่าย

เป็นการเปรียบเทียบปริมาณการร้องขอข้อมูลจากเว็บเบราว์เซอร์ทางฝั่งลูกข่ายระหว่างโนดด็อทเจเอสและจาวาสคริปต์ โดยการใช้ฟังก์ชันของกูเกิลโครมในการเก็บค่า ดังภาพที่ 3.28



ภาพที่ 3.28 การเก็บค่าการร้องขอข้อมูล (Request) จากฝั่งลูกข่ายโดยเว็บเบราว์เซอร์

ในการทดสอบเก็บข้อมูล จะทำการตั้งค่าคาบเวลาการเปลี่ยนแปลงข้อมูล (Interval Time) ไว้คงที่ที่ 1000 ms และทำการเก็บข้อมูลจำนวน 30 ค่า เพื่อสร้างกราฟแสดงความสัมพันธ์ ดังตารางที่ 3.4

ตารางที่ 3.5 ตารางเก็บข้อมูลเปรียบเทียบการร้องขอข้อมูลของจาวาสคริปต์และโนดด็อทเจเอสจากฝั่งลูกข่าย

Structure	Interval (times)						
	1	2	3	4	5	…	30
Node.js	ทำการเก็บข้อมูลแต่ละคาบเวลา เพื่อนำมาเปรียบเทียบกัน						
JavaScript							

4. การทดสอบหาค่าเวลาในการตอบสนองของข้อมูลเปรียบเทียบระหว่างจาวาสคริปต์และโนดด็อทเจเอส

เป็นการหาเวลาที่ใช้ตั้งแต่การส่งการร้องขอข้อมูลไปยังเครื่องแม่ข่ายจนถึงได้รับข้อมูลเสร็จสิ้นโดยการใช้ฟังก์ชันของกูเกิลโครมในการเก็บค่า ดังภาพที่ 3.29

ภาพที่ 3.29 การเก็บค่าเวลาการตอบสนองของข้อมูล

โดยจะทำการเก็บค่าตั้งแต่เริ่มส่งการร้องขอจนได้รับข้อมูลทั้งหมดเสร็จสิ้น (ตั้งแต่ Request sent จนถึง Content Download) แล้วนำข้อมูลมาทำการเปรียบเทียบกันระหว่างการใช้งานภาษาจาวาสคริปต์และภาษาโนดด็อทเจเอส

5. การเปรียบเทียบการทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่าย เมื่อมีการเชื่อมต่อจากผู้ใช้งานจำนวนมาก

เป็นการหาความสัมพันธ์ระหว่างการทำงานของซีพียูที่เครื่องคอมพิวเตอร์แม่ข่าย และจำนวนการเชื่อมต่อจากผู้ใช้งานตั้งแต่ 1 การเชื่อมต่อ ถึง 100 การเชื่อมต่อ เปรียบเทียบกันระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง ที่มีสเปคแตกต่างกัน และทำการเก็บข้อมูลลงในตารางที่ 3.6

ตารางที่ 3.6 ตารางเก็บข้อมูลเปรียบเทียบการทำงานระหว่างซีพียูของเครื่องคอมพิวเตอร์แม่ข่าย และจำนวนการเชื่อมต่อจากผู้ใช้งาน

Structure	Connection Number						
	1	2	3	4	5	…	30
Node.js	ทำการเก็บข้อมูลแต่ละคาบเวลา เพื่อนำมาเปรียบเทียบกัน						
JavaScript							

สเปคของเครื่องคอมพิวเตอร์เครื่องที่ 1 ที่นำมาใช้ทดลอง

Intel(R) Core(TM)2 Duo CPU E8400 @3.00 GHz 3.00 GHz 4.00 GB of RAM

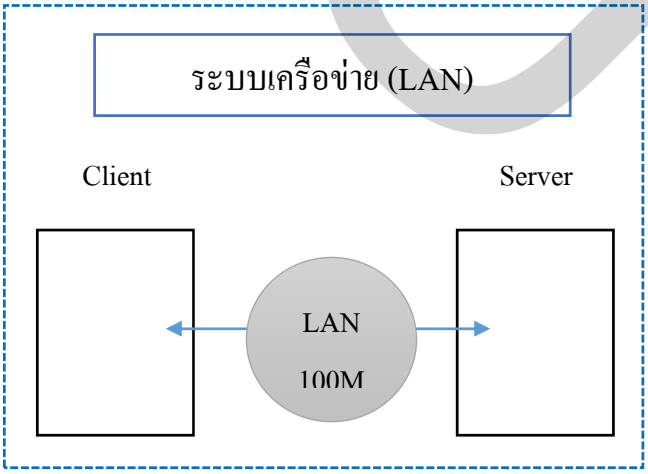
32-bit Operation System, x64-based processor

สเปคของเครื่องคอมพิวเตอร์เครื่องที่ 2 ที่นำมาใช้ทดลอง

Intel(R) Core(TM) i5-3210M CPU @2.50GHz 2.49GHz 4.00 Gb of RAM

64-bit Windows Operation System, x64-based processor

ระบบเครือข่ายที่ใช้ในการทดลอง จะเป็นแบบ LAN (Local Area Network) ระบบปิด ที่มีความเร็วพอร์ต 100 Mbps เชื่อมต่อกันลักษณะดังภาพที่ 30.3



ภาพที่ 3.30 ระบบเครือข่าย LAN ที่ใช้ในการทดลอง

ทำการควบคุมการทำงานของระบบเบื้องหลัง (Background Process) ที่ไม่เกี่ยวข้องโดยการปิดการใช้งาน ตลอดการทดลอง

6. ทำการเขียนโปรแกรมเพื่อควบคุมการทำงานของซีพียูของเครื่องแม่ข่ายแบบอัตโนมัติ

เป็นการเขียนโปรแกรมเพิ่มเพื่อทำการควบคุมและกำหนดช่วงการทำงานของซีพียูของเครื่องแม่ข่ายแบบอัตโนมัติ โดยทำการเปรียบเทียบระหว่างการปรับค่าคาบเวลาของการเปลี่ยนแปลงข้อมูลแบบคงที่กับการนำระบบควบคุมพีไอดีเข้ามาคำนวณค่าคาบเวลาที่จะทำการปรับ โดยมีขั้นตอนดังนี้

6.1 แนวคิดของการออกแบบการเขียนโปรแกรม

โนดด็อทเจเอสมีเอ็นพีเอ็ม (npm) ที่ใช้ในการตรวจสอบสถานะการใช้งานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่าย ในการออกแบบฟังก์ชันการปรับค่าคาบเวลาจะอาศัยเอ็นพีเอ็มดังกล่าวในการกำหนดเงื่อนไข หรือนำค่าการใช้งานของซีพียูเป็นตัวแปรในการปรับค่าคาบเวลา ดังภาพที่ 3.31

```
var cpu = require('windows-cpu');
```

ภาพที่ 3.31 สคริปต์เอ็นพีเอ็มที่ใช้ในการตรวจสอบการใช้งานซีพียู

ขั้นตอนในการทำงานของสคริปต์ มีขั้นตอนหลักๆ ดังนี้

1. กำหนดค่าพื้นฐาน (Default) ของคาบเวลา (Loop Time) เริ่มต้นไว้ที่ 0 ms
2. ทำการตั้งค่าการใช้งานซีพียูเป้าหมาย (Set Point) ที่ยอมรับได้สำหรับเครื่องคอมพิวเตอร์แม่ข่าย (ในการทดสอบกำหนดให้อยู่ที่ 50%) อ่านค่าโดยสคริปต์เอ็นพีเอ็ม (npm)
3. กำหนดให้สคริปต์มีการตรวจสอบค่าการใช้งานซีพียูทุกๆ 100 มิลลิวินาที
4. คำนวณค่าการปรับคาบเวลาโดยใช้ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ (PID Controller)
5. เก็บค่าการใช้งานของซีพียูทุกๆ 100 มิลลิวินาที อย่างต่อเนื่อง เพื่อนำมาวิเคราะห์ผลประสิทธิภาพของโปรแกรม
6. เปรียบเทียบประสิทธิภาพกับการปรับค่าคาบเวลาแบบคงที่

ทำการเปรียบเทียบการอัลกอริทึมที่ใช้ในการปรับค่าคาบเวลาไว้อยู่ การปรับ แบบ คือ 2 ค่าคาบเวลาแบบค่าคงที่ และการปรับค่าคาบเวลาด้วยวิธีระบบควบคุมพีไอดี

การปรับค่าคาบเวลาแบบค่าคงที่ อาศัยการปรับค่าคาบเวลาครั้งละเท่าๆ กัน ในหน่วยมิลลกรัมวินาทีและเก็บค่าการทำงานของซีพียูทุกๆ มิลลิวินาที เมื่อโปรแกรมได้รับค่าการทำงานของ 100 ซีพียูป้อนกลับ CPU Usage Feedback) แล้วทำการเปรียบเทียบกับค่าซีพียูเป้าหมาย (CPU Usage Set Point) ก็จะนำไปปรับค่าคาบเวลาแบบอัตโนมัติ

การปรับค่าคาบเวลาแบบระบบควบคุมพีไอดี เป็นการปรับค่าคาบเวลาโดยใช้น้ำหนักของผลต่างระหว่างค่าการทำงานของซีพียูป้อนกลับ และค่าซีพียูเป้าหมาย นำมาปรับตั้งน้ำหนักของแต่ละส่วนประกอบให้เหมาะสม เพื่อให้โปรแกรมสามารถปรับค่าคาบเวลาได้อย่างรวดเร็ว และสอดคล้องกับค่าซีพียูเป้าหมายแบบอัตโนมัติ ดังแสดงในรหัสเทียม (Pseudocode) ดังภาพที่ 3.32

```
previous_error = setpoint – actual_cpu
start:
        error = setpoint – actual_cpu
        integral = integral + (error*dt)
        derivative = (error – previous_error)/dt
        output = (Kp*error) + (Ki*integral) + (Kd*derivative)
        previous_error = error
wait(dt)
goto start
```

ภาพที่ 3.32 รหัสเทียมของระบบควบคุมพีไอดี

DPU

จากนั้นจึงนำรหัสเทียมไปทำการเขียนโปรแกรมเพิ่มเติมที่โนดด็อทเจเอส ดังภาพที่ 3.33

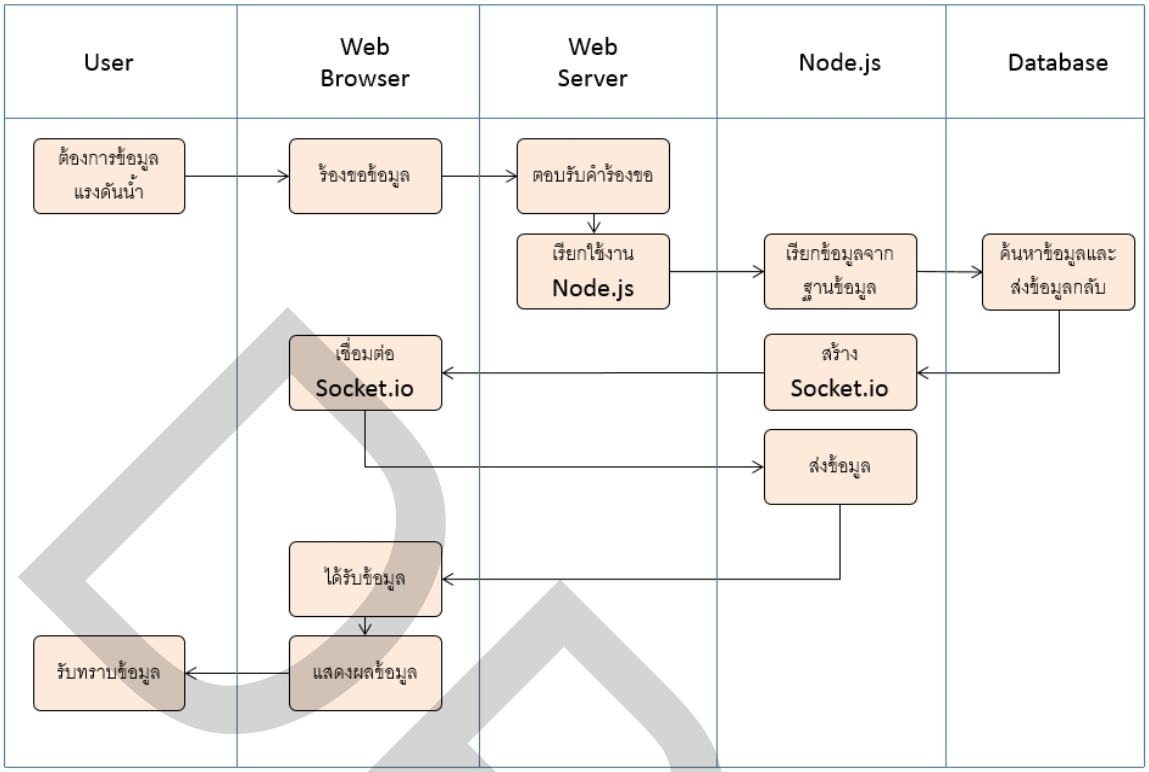
```
var Kp = 1;
var Ki = 1;
var Kd = 1;

var actual_position = results;
var setpoint = 50;
var integral = 0;
var previous_error = 0;
var error = setpoint - actual_position;
var dt = 100;
var integral = integral + (error*dt);
var derivative = (error - previous_error)/dt;
var output = (Kp*error) + (Ki*integral) + (Kd*derivative);
    polling = output;
    if(polling  <= 0){
       polling = 0;
    }
    previous_error = error;

POLLING_INTERVAL = polling;
pollingTimer = setTimeout(pollingLoop, POLLING_INTERVAL);

console.log('' + results +' % ' + polling);
```

ภาพที่ 3.33 การปรับคาบเวลาในการใช้งานซีพียูโดยใช้ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์

แล้วจึงทำการเก็บค่าการทำงานของซีพียู และค่าคาบเวลาที่ทำการปรับ เพื่อนำไปสร้างกราฟแสดงความสัมพันธ์ และคำนวณผลต่างของการทำงานของซีพียูจริงกับค่าเป้าหมาย

3.4 แผนภาพแสดงขั้นตอนการทำงานของระบบ

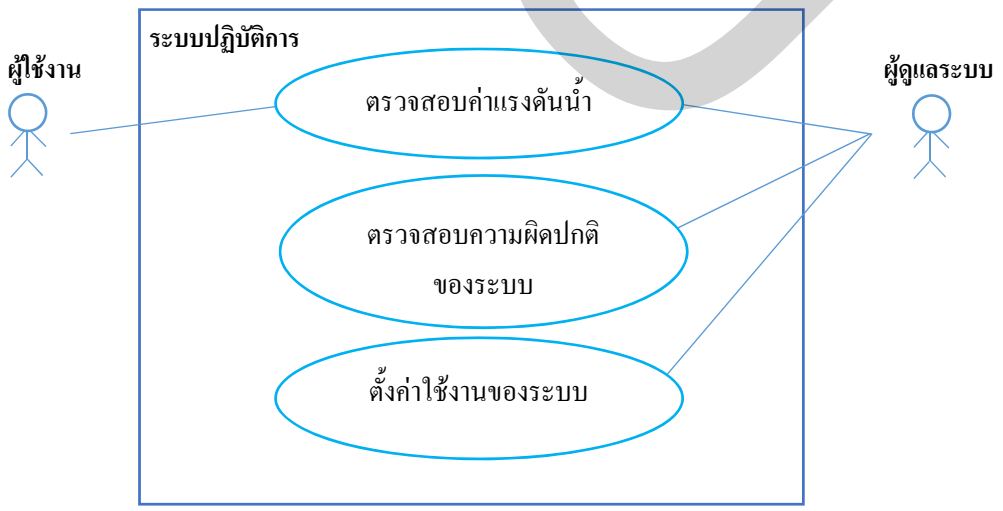
ผังงานแสดงขั้นตอนการทำงานของระบบในรูปแบบ Swim Lane Activity Diagram ดังภาพที่ 3.34



ภาพที่ 3.34 ผังงานแสดงขั้นตอนการทำงานของระบบ

3.5 แผนภาพยูสเคส (Use Case Diagram)

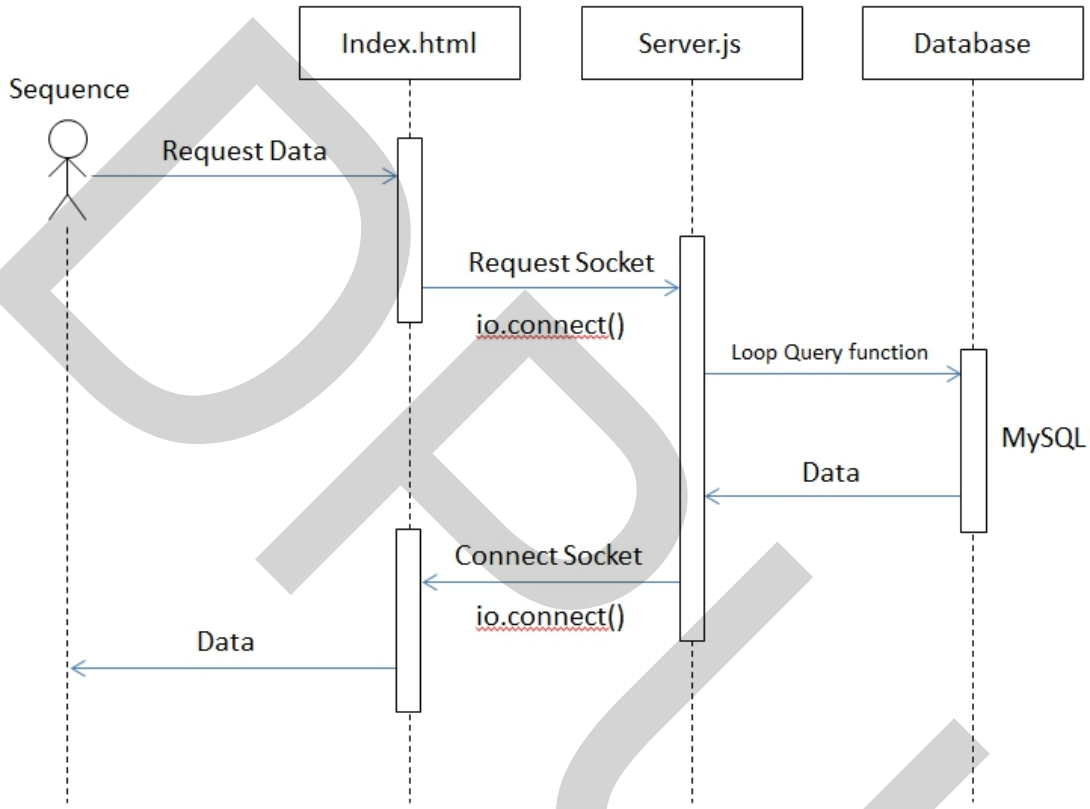
โปรแกรมตรวจสอบแรงดันน้ำในท่อประธาน จะแบ่งผู้ที่มีส่วนในการใช้งานโปรแกรมออกเป็น 2 ส่วน คือ ผู้ใช้งานและผู้ดูแลระบบ ดังภาพที่ 3.35



ภาพที่ 3.35 แผนภาพผู้ใช้งาน (Use Case Diagram) ส่วนของผู้ใช้งานและผู้ดูแล

3.5.1 ผู้ใช้งาน

ผู้ใช้งานจะมีสิทธิ์ในการเข้าใช้งานได้เพียงการตรวจสอบข้อมูลแรงดันน้ำในท่อประธาน โดยผ่านการใช้งานผ่านเว็บเบราว์เซอร์ได้เพียงอย่างเดียวเท่านั้น โดยจะไม่สามารถปรับแต่งระบบได้ ดังภาพที่ 3.36



ภาพที่ 3.36 แผนภาพลำดับขั้นตอนของผู้ใช้งานส่วนตรวจสอบค่าแรงดันน้ำ

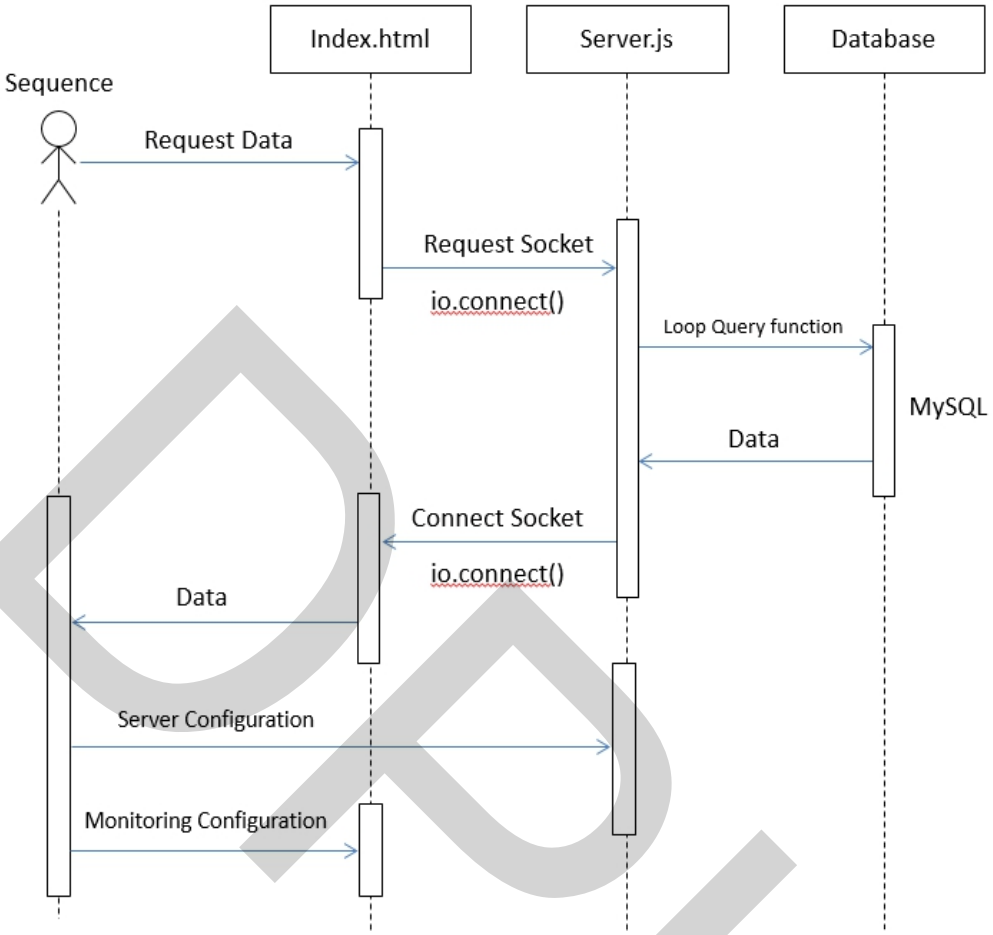
ตารางที่ 3.7 ข้อกำหนดซอฟต์แวร์ส่วนตรวจสอบค่าแรงดันน้ำ

System	โปรแกรมตรวจสอบแรงดันน้ำในท่อประปาแบบเวลาจริง
Use Case	ตรวจสอบค่าแรงดันน้ำ
Actor	ผู้ใช้งาน, ผู้ดูแลระบบ
Action	• ผู้ใช้งานตรวจสอบความถูกต้องของข้อมูลแรงดันน้ำและอัตราการไหลของน้ำแต่ละจุดวัด โดยอ้างอิงความถูกต้องจากข้อมูลวันที่ที่มีการเปลี่ยนแปลงให้เป็นปัจจุบันและข้อมูลแรงดันน้ำบริเวณใกล้เคียง • ผู้ใช้งานตรวจสอบความผิดปกติของระบบแรงดันน้ำจากข้อมูลที่แสดงบนโปรแกรมและทำการแก้ไข
Stimulus	คาบเวลาในการแสดงผลข้อมูล
Response	โปรแกรมแสดงข้อมูลแรงดันน้ำและอัตราการไหลของน้ำที่เว็บเบราว์เซอร์ตามคาบเวลา

การเข้าถึงข้อมูลแรงดันน้ำของผู้ใช้งานนั้น สามารถทำได้โดยผ่านการใช้งานเว็บเบราว์เซอร์ เว็บเบราว์เซอร์จะทำการร้องขอ (Request) ข้อมูลไปยังเครื่องคอมพิวเตอร์แม่ข่าย และเครื่องแม่ข่ายจะทำการเรียกใช้งานฟังก์ชันโนดด็อทเจเอส โดยที่โนดด็อทเจเอสจะทำการเข้าถึงฐานข้อมูล เพื่อนำข้อมูลแรงดันน้ำส่งออกมาที่เครื่องคอมพิวเตอร์แม่ข่าย โดยการทำงานของโนดด็อทเจเอสจะเป็นการส่งข้อมูล (Push Data) ออกมาอย่างต่อเนื่อง และเครื่องคอมพิวเตอร์แม่ข่ายจะส่งข้อมูลที่ได้ผ่านช่องทางเว็บซ็อคเก็ตไปยังเว็บเบราว์เซอร์ เพื่อให้ผู้ใช้งานรับทราบข้อมูลอย่างต่อเนื่อง

3.5.2 ผู้ดูแลระบบ

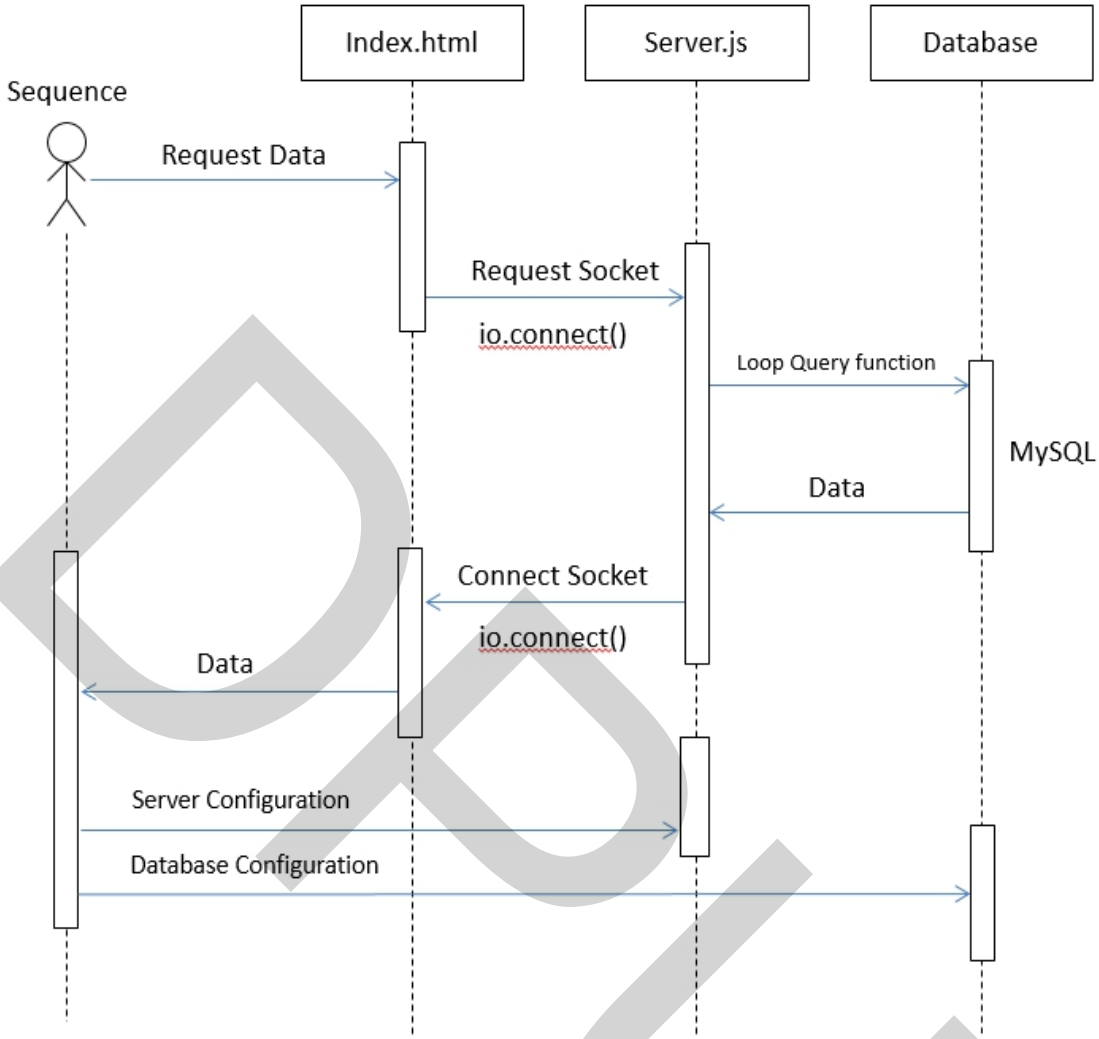
ผู้ดูแลระบบจะมีสิทธิ์ในการเข้าถึงการตั้งค่าต่างๆ ของระบบ เพื่อให้มีความเหมาะสมกับทรัพยากรทางคอมพิวเตอร์ของเครื่องคอมพิวเตอร์แม่ข่ายและคอมพิวเตอร์ลูกข่าย รวมถึงมีหน้าที่ในการตรวจสอบการทำงานของระบบว่า ยังสามารถทำงานได้อย่างเป็นปกติหรือไม่ และทำการแก้ไขในกรณีที่เกิดความผิดปกติขึ้น ดังภาพที่ 3.37 และ 3.38



ภาพที่ 3.37 แผนภาพลำดับขั้นตอนของผู้ดูแลระบบในส่วนตรวจสอบความผิดปกติของระบบ

ตารางที่ 3.8 ข้อกำหนดซอฟต์แวร์ส่วนตรวจสอบความผิดปกติของระบบ

System	โปรแกรมตรวจสอบแรงดันน้ำในท่อประปาแบบเวลาจริง
Use Case	ตรวจสอบความผิดปกติของระบบ
Actor	ผู้ดูแลระบบ
Action	• ผู้ดูแลระบบตรวจสอบการแสดงผลข้อมูลแรงดันน้ำและอัตราการไหล ว่ามีการแสดงผลครบถ้วนและเปลี่ยนแปลงให้เป็นปัจจุบันหรือไม่ แล้วทำการแก้ไข • ผู้ดูแลระบบตรวจสอบการแสดงผลข้อมูลแรงดันน้ำและอัตราการไหล ว่ามีการแสดงผลข้อมูลตามคาบเวลาที่กำหนดไว้หรือไม่ แล้วทำการแก้ไข
Stimulus	โปรแกรมแสดงข้อมูลแรงดันน้ำและอัตราการไหลของน้ำที่เว็บเบราว์เซอร์ตามคาบเวลา
Response	จำนวนข้อมูลที่แสดงและคาบเวลาในการแสดงผลข้อมูล



ภาพที่ 3.38 แผนภาพลำดับขั้นตอนของผู้ดูแลระบบในส่วนตั้งค่าใช้งานของระบบ

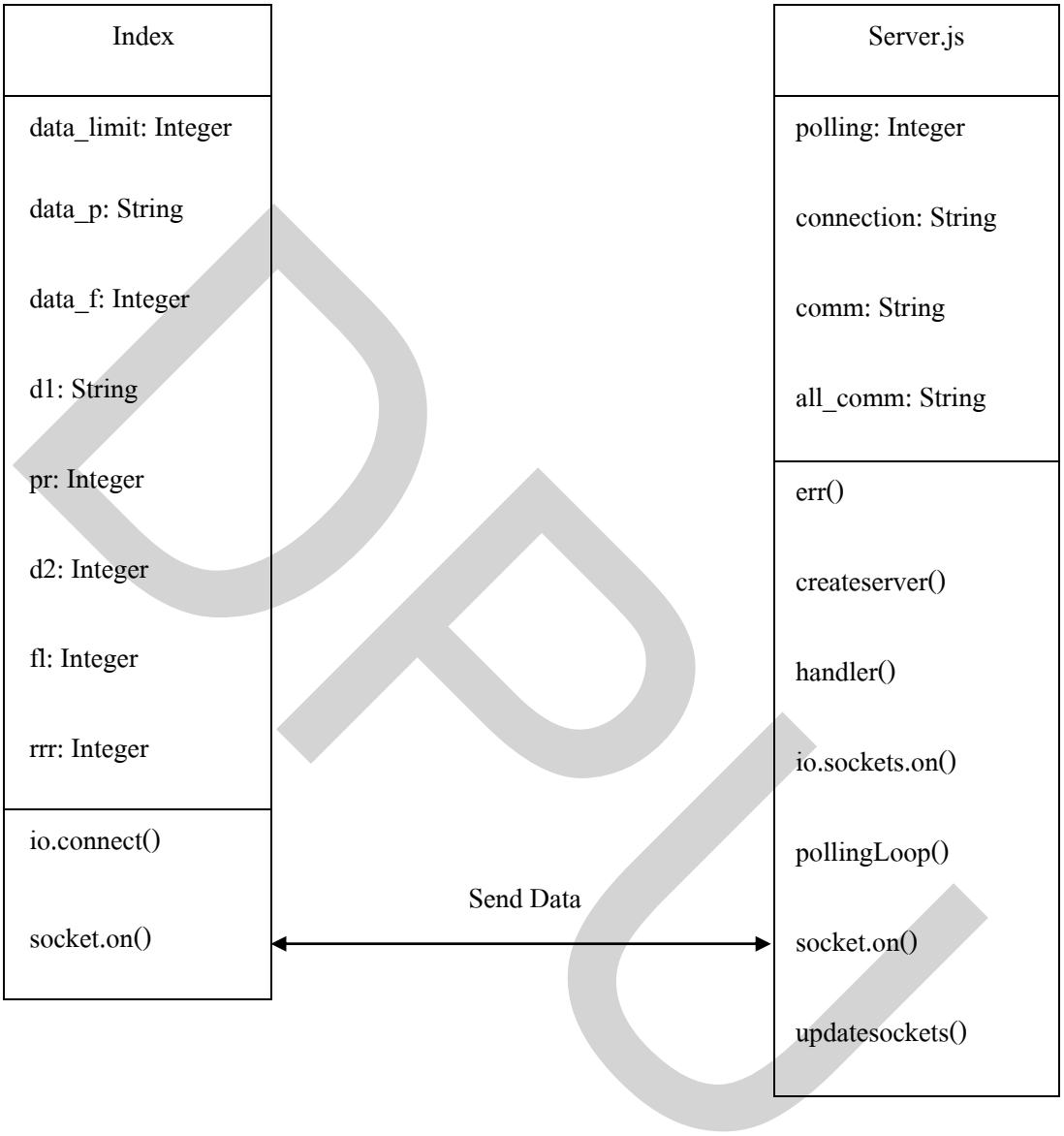
DPU

ตารางที่ 3.9 ข้อกำหนดซอฟต์แวร์ส่วนตั้งค่าใช้งานของระบบ

System	โปรแกรมตรวจสอบแรงดันน้ำในท่อประปาแบบเวลาจริง
Use Case	ตั้งค่าใช้งานของระบบ
Actor	ผู้ดูแลระบบ
Action	• ผู้ดูแลระบบทำการกำหนดค่าคาบเวลาที่เหมาะสมกับการแสดงผลข้อมูลแรงดันน้ำและอัตราการไหลของโปรแกรม • ผู้ดูแลระบบทำการกำหนดช่วงการทำงานของซีพียูของเครื่องแม่ข่ายให้เหมาะสมกับทรัพยากรทางคอมพิวเตอร์ของเครื่องแม่ข่าย • ผู้ดูแลระบบทำการออกแบบและตั้งค่าฐานข้อมูลให้เหมาะสมกับโปรแกรม
Stimulus	โปรแกรมแสดงข้อมูลแรงดันน้ำและอัตราการไหลของน้ำที่เว็บเบราว์เซอร์ตามคาบเวลา
Response	• คาบเวลาในการแสดงผลข้อมูล • การทำงานของซีพียูของเครื่องแม่ข่าย

โดยปกติ ผู้ดูแลระบบจะมีหน้าที่เช่นเดียวกันกับผู้ใช้งานคือ การตรวจสอบระบบแรงดันน้ำ แต่จะมีอีกหนึ่งหน้าที่ในการปรับระบบให้มีเสถียรภาพมากยิ่งขึ้น รวมทั้งดูแลระบบให้มีการทำงานที่ปกติตลอดเวลาอีกด้วย

3.6 แผนภาพคลาส (Class Diagram)



ภาพที่ 3.39 แผนภาพคลาส

DPU

3.7 คำอธิบายตัวแปรและฟังก์ชัน

ลักษณะการทำงานของฟังก์ชันต่างๆ มีคำอธิบาย ดังตารางที่ 3.10 และ 3.11

ตารางที่ 3.10 คำอธิบายตัวแปรและฟังก์ชัน

Index	
data_limit	กำหนดจำนวณข้อมูลที่ต้องการจากฐานข้อมูล
data_p	ข้อมูลแรงดันน้ำจากฐานข้อมูล
data_f	ข้อมูลอัตราการไหลจากฐานข้อมูล
d1	แจกแจงลำดับของ data_p ภายใน array
pr	แจกแจงลำดับของ data_p ภายใน array
d2	แจกแจงลำดับของ data_f ภายใน array
fl	แจกแจงลำดับของ data_f ภายใน array
rrr	แจกแจงลำดับของ data_p และ data_f ที่หน้า web browser
io.connect()	ฟังก์ชันสร้างการเชื่อมต่อจาก Client กับ Server
socket.on()	ฟังก์ชันรับข้อมูลจาก Server

ตารางที่ 3.11 คำอธิบายตัวแปรและฟังก์ชัน

Server.js	
polling	กำหนดคาบเวลาในการส่งข้อมูลไปยัง Client
connection	ข้อมูลเชื่อมต่อกับฐานข้อมูล
comm	คำสั่งการ Query ข้อมูลจากฐานข้อมูล
all_comm	ข้อมูลที่ได้จากการ Query
err()	ตรวจสอบความผิดพลาดของการเชื่อมต่อ
createserver()	สร้าง Server เพื่อรับ-ส่งข้อมูล
handler()	ตรวจสอบสถานะการสร้าง Server รับ-ส่งข้อมูล
io.sockets.on()	สร้างช่องทางการเชื่อมต่อกับ Client
pollingLoop()	สร้าง Loop การรับ-ส่งข้อมูลระหว่าง Client และ Server
socket.on()	ส่งข้อมูลไปยัง Client
updatesockets()	ตรวจสอบและจัดหา Socket ที่ว่างอยู่ เพื่อทำการเชื่อมต่อ

DPU

บทที่ 4
การทดลองและผลการวิจัย

หลังจากที่ได้ทำการพัฒนาโปรแกรมตรวจสอบแรงดันน้ำในท่อประปาแบบเวลาจริงแล้วนั้น ผู้วิจัยได้ทำการทดลองใช้งานโปรแกรมในลักษณะต่างๆ เพื่อให้ทราบถึงข้อจำกัดของระบบ และประเด็นในการพัฒนาระบบต่อเนื่อง เพื่อทำการนำไปประยุกต์ใช้งานจริง โดยมีหัวข้อในการวิจัย ดังนี้

1. การเปรียบเทียบการทำงานของซีพียูระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ ทางด้านฝั่งเครื่องคอมพิวเตอร์แม่ข่าย
2. การเปรียบเทียบการทำงานของซีพียูระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ทางด้านฝั่งเครื่องคอมพิวเตอร์ลูกข่าย
3. การเปรียบเทียบปริมาณการร้องขอข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์จากทางฝั่งคอมพิวเตอร์ลูกข่าย
4. การเปรียบเทียบปริมาณการส่งผ่านข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ ผ่านทางเว็บเบราว์เซอร์
5. การเปรียบเทียบค่าเวลาในการตอบสนองของข้อมูลเปรียบเทียบระหว่างจาวาสคริปต์และโนดด็อทเจเอส
6. ความสัมพันธ์ระหว่างการทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่ายและปริมาณการเชื่อมต่อจากผู้ใช้งาน
7. การเขียนโปรแกรมเพื่อควบคุมการทำงานของซีพียูของเครื่องแม่ข่ายแบบอัตโนมัติ

4.1 การเปรียบเทียบการทำงานของซีพียูระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ทางด้านฝั่งเครื่องคอมพิวเตอร์แม่ข่าย

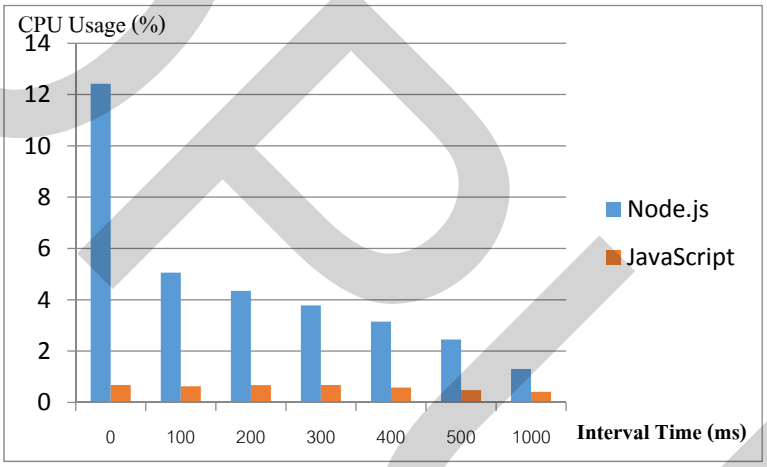
4.1.1 รายละเอียดการทดลอง

การทดลองเปรียบเทียบการทำงานของซีพียูนั้น จะทำการทดลองที่ค่าคาบเวลาในการรับ-ส่งข้อมูลที่แตกต่างกัน ดังนี้ 0 100 200 300 400 500 และ 1000 ms โดยทำการทดลองจำนวน 10 ครั้ง เพื่อนำมาหาค่าเฉลี่ยและนำมาวาดกราฟเพื่อหาแนวโน้มของของข้อมูล ดังตารางที่ 4.1 และภาพที่ 4.1

ในการทดลองมีการกำหนดค่าตัวแปรควบคุม คือ จำนวนการเชื่อมต่อจากผู้ใช้งาน โดยกำหนดให้มีจำนวนการเชื่อมต่อเพียง 1 การเชื่อมต่อ

ตารางที่ 4.1 การทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่ายที่คาบเวลาต่างๆ

Structure	Interval Time (ms)						
	0	100	200	300	400	500	1000
Node.js	12.42	5.06	4.35	3.78	3.15	2.45	1.3
JavaScript	0.68	0.63	0.67	0.68	0.58	0.48	0.41



ภาพที่ 4.1 กราฟเปรียบเทียบการทำงานของซีพียูของเครื่องแม่ข่าย

4.1.2 วิเคราะห์ผลการทดลอง

จากรูปที่ 4.1 เมื่อทำการกำหนดคาบเวลาในการรับ-ส่งข้อมูลให้อยู่ในระดับต่ำ (ช่วงประมาณ 0-100 ms) การทำงานของซีพียูของการเขียนโปรแกรมภาษาโนดด็อทเจเอสฝั่งเครื่องคอมพิวเตอร์แม่ข่าย จะมีค่ามากกว่าการทำงานของภาษาจาวาสคริปต์ และลดลงอย่างต่อเนื่องเมื่อทำการเพิ่มคาบเวลาให้นานขึ้น ในขณะที่ภาษาจาวาสคริปต์นั้นจะมีการทำงานของซีพียูที่ค่อนข้างจะคงที่และมีค่าน้อยกว่า

ผลลัพธ์ดังกล่าว เกิดจากลักษณะการทำงานของโนดด็อทเจเอส ที่เป็นการผลักข้อมูลจากเครื่องคอมพิวเตอร์แม่ข่ายไปยังเครื่องคอมพิวเตอร์ลูกข่าย จึงมีการทำงานของซีพียูที่ต่อเนื่อง

ตลอดเวลา ในกรณีที่ทำการตั้งค่าคาบเวลาน้อยเกินไป จะทำให้เกิดการใช้งานทรัพยากรสูง ซึ่งต่างกับการทำงานของภาษาจาวาสคริปต์ที่รอการร้องขอข้อมูลจากฝั่งลูกข่ายจึงจะมีการส่งข้อมูลออกไป

4.2 การเปรียบเทียบการทำงานของซีพียูระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ทางด้านฝั่งเครื่องคอมพิวเตอร์ลูกข่าย

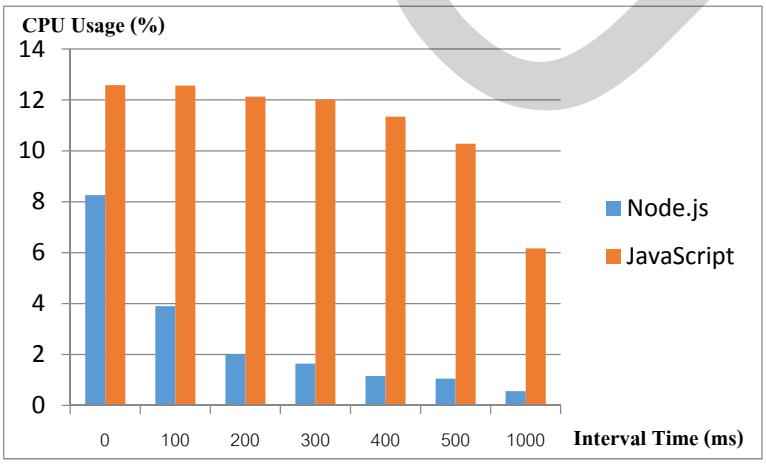
4.2.1 รายละเอียดการทดลอง

การทดลองเปรียบเทียบการทำงานของซีพียูนั้น จะทำการทดลองที่ค่าคาบเวลาในการรับ-ส่งข้อมูลที่แตกต่างกัน ดังนี้ 0 100 200 300 400 500 และ 1000 ms โดยทำการทดลองจำนวน 10 ครั้ง เพื่อนำมาหาค่าเฉลี่ยและนำมาวาดกราฟเพื่อหาแนวโน้มของข้อมูล เช่นเดียวกันกับหัวข้อ 4.1 เพียงแต่หัวข้อที่ 4.2 จะเป็นการทดลองที่ฝั่งเครื่องคอมพิวเตอร์ลูกข่าย ดังตารางที่ 4.2 และภาพที่ 4.2

ในการทดลองมีการกำหนดค่าตัวแปรควบคุม คือ จำนวนการเชื่อมต่อจากผู้ใช้งาน โดยกำหนดให้มีจำนวนการเชื่อมต่อเพียง 1 การเชื่อมต่อ

ตารางที่ 4.2 การทำงานของซีพียูของเครื่องคอมพิวเตอร์ลูกข่ายที่คาบเวลาต่างๆ

Structure	Interval Time (ms)						
	0	100	200	300	400	500	1000
Node.js	8.26	3.9	2	1.64	1.16	1.05	0.56
JavaScript	12.58	12.56	12.13	12.01	11.34	10.28	6.17

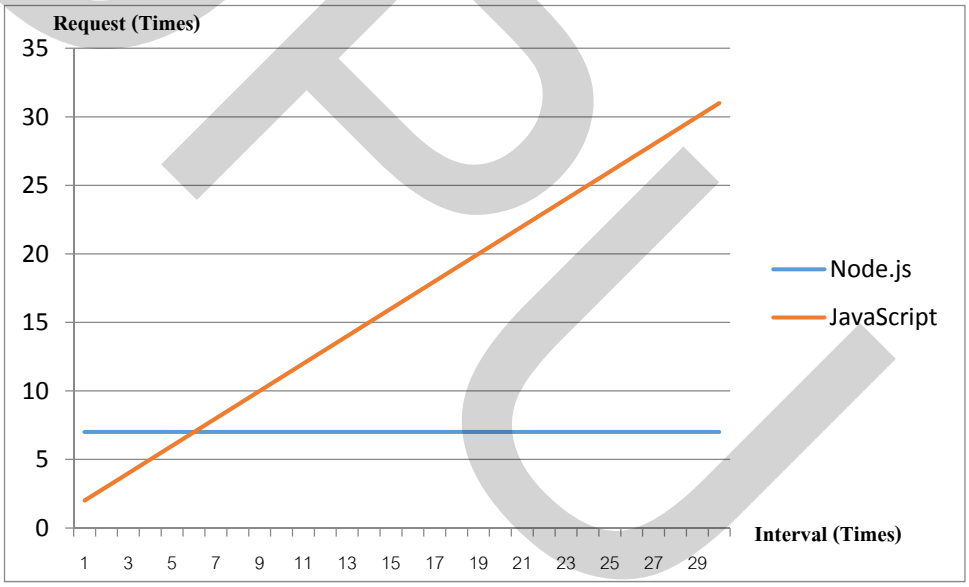


ภาพที่ 4.2 กราฟเปรียบเทียบการทำงานของซีพียูของเครื่องลูกข่าย

4.1.2 วิเคราะห์ผลการทดลอง

จากรูปที่ 4.2 เมื่อพิจารณาการทำงานของซีพียูที่ฝั่งเครื่องคอมพิวเตอร์ลูกข่าย การทำงานของซีพียูของภาษาจาวาสคริปต์จะมีค่ามากกว่าการทำงานของโนดด็อทเจเอส เนื่องจากภาษาจาวาสคริปต์จะมีปริมาณข้อมูลในส่วนของเฮดเดอร์ (Header) จำนวนมาก ที่ส่งมาพร้อมกับข้อมูลที่ทำการร้องขอ และจะมีการประมวลผลใหม่ทุกๆ ครั้งที่ได้รับข้อมูลตามคาบเวลา (ดาวน์โหลดหน้าเว็บเพจใหม่ทั้งหมด) ในขณะที่โนดด็อทเจเอส จะมีเพียงการแสดงผลข้อมูลที่เปลี่ยนแปลงเพียงเล็กน้อย เนื่องจากเป็นข้อมูลที่มีการส่งผ่านทางเว็บซ็อคเก็ต จะไม่มีเฮดเดอร์เหมือนภาษาจาวาสคริปต์ จึงไม่ต้องทำการโหลดหน้าเว็บเพจใหม่ทุกครั้ง

4.3 การเปรียบเทียบปริมาณการร้องขอข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์จากทางฝั่งคอมพิวเตอร์ลูกข่าย

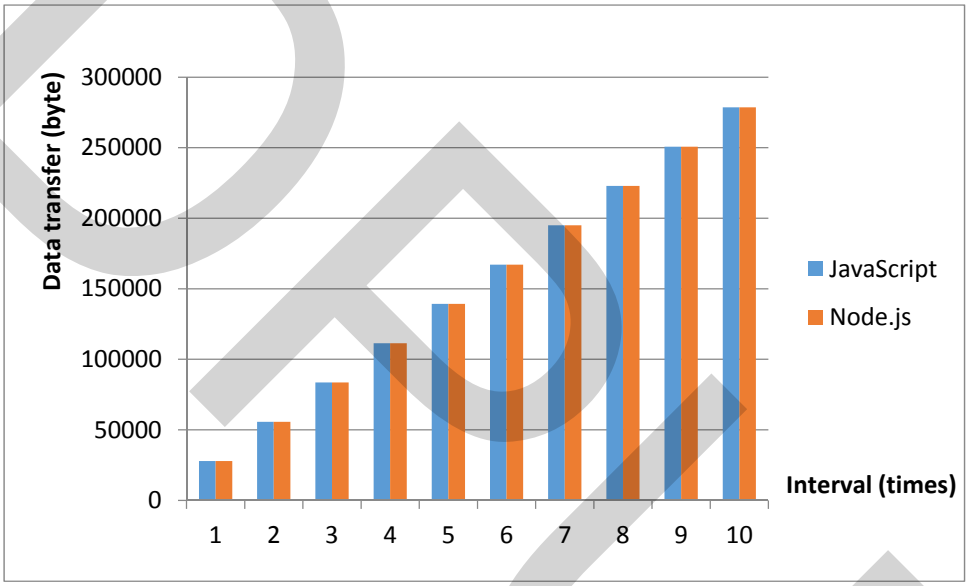


ภาพที่ 4.3 กราฟเปรียบเทียบปริมาณการร้องขอ (Request) ของโปรแกรมทั้งสองจากฝั่งลูกข่าย

จากภาพที่ 4.3 แสดงให้เห็นถึงจำนวนครั้งของการร้องขอ (Request) ของรูปแบบการเขียนโปรแกรมภาษาโนดด็อทเจเอส ที่จะมีการร้องขอข้อมูลเริ่มต้นเพียงค่าเดียวและจะคงที่ไปตลอดการใช้งาน ซึ่งสอดคล้องกับลักษณะโครงสร้างของโนดด็อทเจเอส ที่จะมีการร้องขอข้อมูลเพียงครั้งแรกครั้งเดียวและจะสร้างการเชื่อมต่อผ่านพอร์ตโดยใช้ซ็อตเก็ตไอโอ (Socket.io) เพื่อทำการส่งผ่านข้อมูลระหว่างเครื่องคอมพิวเตอร์แม่ข่ายและเครื่องคอมพิวเตอร์ลูกข่าย ทำให้ช่วยลด

โอกาสที่จะเกิดการล่มของฝั่งแม่ข่ายอันเนื่องมากจากปริมาณการร้องขอข้อมูลที่มากจนเกินไปได้ ในขณะที่ภาษาจาวาสคริปต์จะมีจำนวนการร้องขอเพิ่มขึ้นทุกๆ คาบเวลา เนื่องจากตัวภาษาจาวาสคริปต์นั้น จะมีการตั้งคาบเวลาในการร้องขอข้อมูลใหม่ทุกครั้ง เพื่อนำข้อมูลที่ได้มาแสดงผล ทำให้เกิดปริมาณการร้องขอที่เพิ่มอย่างต่อเนื่องและคงที่ เพราะมีการร้องขอข้อมูลซ้ำๆ เดิมตลอด

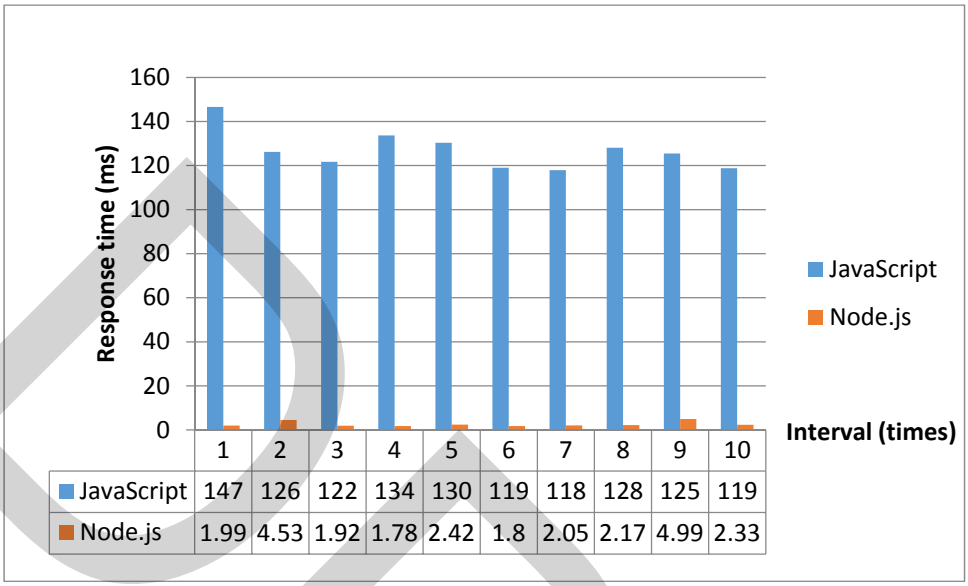
4.4 การเปรียบเทียบปริมาณการส่งผ่านข้อมูลภายในเว็บเบราว์เซอร์ระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ ผ่านทางเว็บเบราว์เซอร์



ภาพที่ 4.4 กราฟเปรียบเทียบการส่งผ่านข้อมูล (Data Transfer) ของรูปแบบทั้งสอง

จากภาพที่ 4.4 แสดงให้เห็นถึงปริมาณการส่งผ่านข้อมูลภายในเว็บเบราว์เซอร์ของการเขียนโปรแกรมทั้งสองแบบที่มีขนาดของข้อมูลส่งผ่านระบบเท่ากัน สอดคล้องกับจำนวนตัวอักษร (Plain Text) ที่เกิดจากการออกแบบเพื่อใช้งานในลักษณะของเจสัน (JSON: JavaScript Object Notation) ดังนั้น ในการเขียนโปรแกรมด้วยภาษาโนดด็อทเจเอส จึงสามารถทดแทนการใช้ภาษาจาวาสคริปต์ในเรื่องของการส่งผ่านข้อมูลภายในเว็บเบราว์เซอร์ได้อย่างสมบูรณ์

4.5 การเปรียบเทียบค่าเวลาในการตอบสนองของข้อมูลเปรียบเทียบระหว่างจาวาสคริปต์และโนดด็อทเจเอส



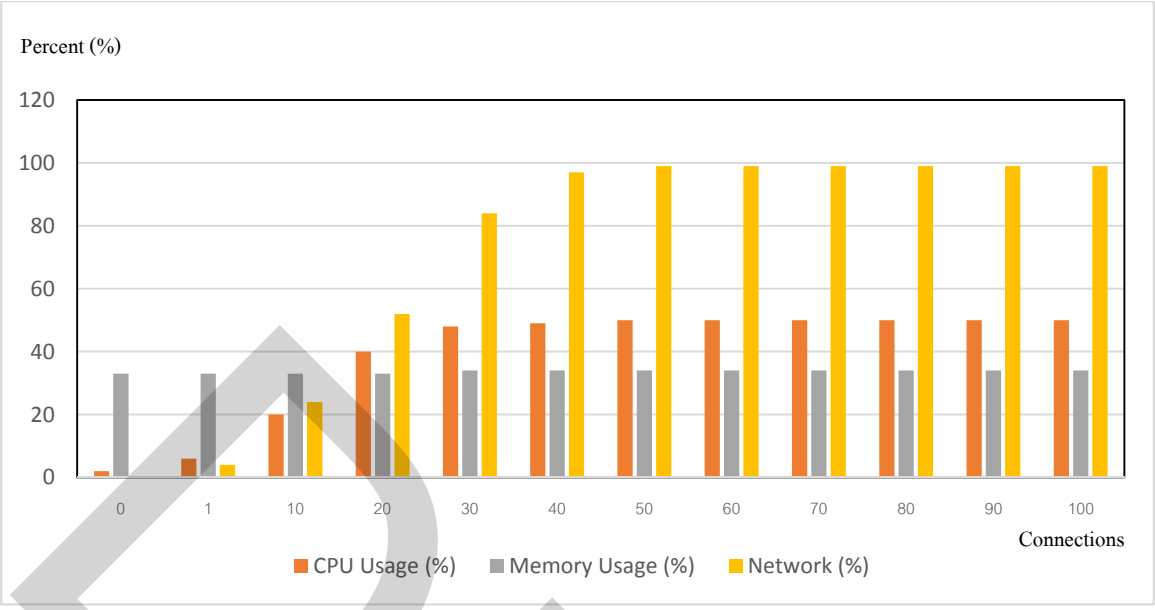
	1	2	3	4	5	6	7	8	9	10
JavaScript	147	126	122	134	130	119	118	128	125	119
Node.js	1.99	4.53	1.92	1.78	2.42	1.8	2.05	2.17	4.99	2.33

ภาพที่ 4.5 กราฟเปรียบเทียบเวลาในการตอบสนองของข้อมูลระหว่างจาวาสคริปต์และโนดด็อทเจเอส

จากภาพที่ 4.5 การเขียนโปรแกรมด้วยภาษาจาวาสคริปต์จะมีการใช้เวลาในการตอบสนองของข้อมูลมากกว่าการเขียนโปรแกรมด้วยภาษาโนดด็อทเจเอส 50-70 เท่า แสดงว่าประสิทธิภาพของการเข้าถึงข้อมูลของโนดด็อทเจเอสนั้นมากกว่าจาวาสคริปต์ ดังนั้นเมื่อกำหนดค่าคาบเวลาในการเปลี่ยนแปลงข้อมูลที่น้อยๆ การใช้งานภาษาโนดด็อทเจเอสจึงแสดงข้อมูลได้รวดเร็วกว่า

4.6 ความสัมพันธ์ระหว่างการทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่าย ปริมาณการเชื่อมต่อจากผู้ใช้งาน หน่วยความจำและการส่งข้อมูลภายในเครือข่าย

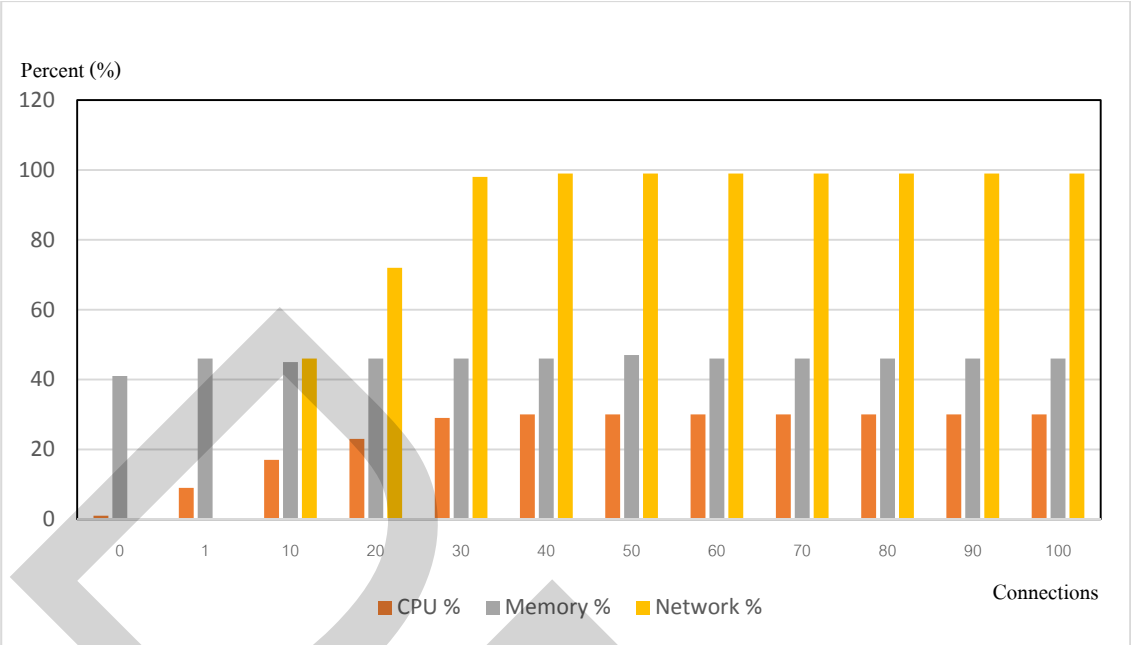
ทำการทดสอบเพิ่มจำนวนการเชื่อมต่อจากผู้ใช้งานตั้งแต่ 0 ถึง 100 การเชื่อมต่อ และทำการเก็บข้อมูลการใช้ทรัพยากรทางคอมพิวเตอร์ ดังต่อไปนี้ การใช้งานซีพียู หน่วยความจำ และการส่งข้อมูลภายในเครือข่าย แล้วจึงนำข้อมูลที่ได้มาเขียนกราฟแสดงความสัมพันธ์ ดังภาพที่ 4.6



ภาพที่ 4.6 ข้อมูลทรัพยากรทางคอมพิวเตอร์เมื่อมีการเชื่อมต่อที่มากขึ้น

จากรูปที่ 4.6 เมื่อมีการเชื่อมต่อเพิ่มมากขึ้น เครื่องคอมพิวเตอร์แม่ข่ายจะมีการใช้งานทรัพยากรมากขึ้นเรื่อยๆ ตามจำนวนการเชื่อมต่อ จนถึงจำนวนการเชื่อมต่อ 40 การเชื่อมต่อจะพบว่า การใช้งานซีพียูจะมีค่าประมาณ 50% ในขณะที่การส่งข้อมูลภายในเครือข่ายนั้นถูกใช้งานไป 99% แล้ว และเมื่อได้ทำการเพิ่มจำนวนการเชื่อมต่อไปจนถึง 100 การเชื่อมต่อ จะพบว่าค่าของซีพียูจะมีค่าคงที่ ที่ 50% ในขณะที่หน่วยความจำของเครื่องคอมพิวเตอร์แม่ข่ายมีการใช้งานคงที่ตลอดการทดลอง

ข้อสังเกตจากผลการทดลอง คือ ในขณะที่มีการใช้งานการส่งข้อมูลภายในเครือข่ายจนถึง 99% แล้วนั้น อาจจะทำให้ขีดความสามารถของโนดด็อทเจเอสถูกจำกัด จึงไม่สามารถที่จะเพิ่มการทำงานของซีพียูต่อไปได้ตามจำนวนการเชื่อมต่อที่เพิ่มขึ้น ทั้งนี้ผู้วิจัยทดลองกับระบบเครื่องข่ายแบบ LAN (Local Area Network) ที่มีขีดจำกัดอยู่ที่ 100Mbps ตามการใช้งานจริง จึงมิได้มีการทดลองในส่วนของระบบเครือข่ายเพิ่มเติม แต่ผู้วิจัยได้ทำการทดลองแบบเดียวกันกับเครื่องคอมพิวเตอร์ที่มีสเปคแตกต่างกันเพื่อนำมาเปรียบเทียบว่าเกิดจากระบบเครือข่ายจริงหรือไม่



ภาพที่ 4.7 ข้อมูลทรัพยากรทางคอมพิวเตอร์เมื่อมีการเชื่อมต่อที่มากขึ้นจากคอมพิวเตอร์อีกเครื่องหนึ่ง

จากรูปที่ 4.7 จะเห็นได้ว่าการทำงานของซีพียูของเครื่องคอมพิวเตอร์แม่ข่ายเครื่องใหม่จะเริ่มมีค่าคงตัวที่ 30% เมื่อมีรับ-ส่งข้อมูลของเครือข่ายที่ 99% เช่นเดียวกันกับการทดลองก่อนหน้านี้ แสดงว่าการทำงานของซีพียูจะถูกจำกัดเนื่องจากไม่สามารถรับ-ส่งข้อมูลภายในระบบเครือข่ายเพิ่มเติมได้ เนื่องจากมีการใช้งานเครือข่ายจนถึงขีดสุดแล้ว

ดังนั้น จากข้อมูลเบื้องต้นสรุปได้ว่า จำนวนการเชื่อมต่อที่เพิ่มมากขึ้น มีผลต่อการทำงานของซีพียูของเครื่องแม่ข่าย โดยที่การทำงานของซีพียูจะสูงขึ้น ตามจำนวนการเชื่อมต่อ และจะคงที่เมื่อการใช้งานเครือข่ายมีการใช้งานที่ 100% แล้ว

4.7 การเขียนโปรแกรมเพื่อควบคุมการทำงานของซีพียูของเครื่องแม่ข่ายแบบอัตโนมัติ

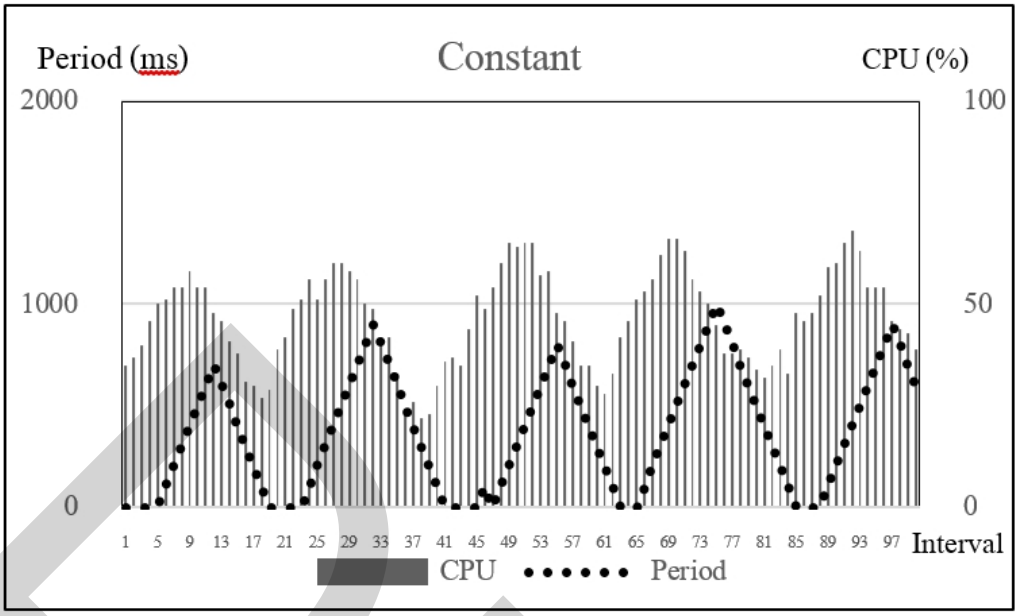
จากการวิเคราะห์ผลการทดสอบการใช้งานทรัพยากรทางคอมพิวเตอร์ที่ฝั่งเครื่องคอมพิวเตอร์แม่ข่าย จะเห็นได้ว่า การใช้งานซีพียูของโนดด็อทเจเอสสูงกว่าของภาษาจาวาสคริปต์ ดังนั้นผู้ดูแลควรมีการปรับแต่งการตั้งค่าการรับ-ส่งข้อมูลที่ฝั่งเครื่องคอมพิวเตอร์แม่ข่ายให้มีความเหมาะสมกับสภาพของเครื่องคอมพิวเตอร์แม่ข่าย เพื่อให้เกิดประสิทธิภาพที่ดีขึ้น และไม่เกิดการล่มของระบบอันเนื่องมาจากการใช้งานทรัพยากรคอมพิวเตอร์ที่มากเกินความจำเป็น

ปัจจัยที่มีผลโดยตรงต่อการใช้งานทรัพยากรทางคอมพิวเตอร์ ก็คือค่าของคาบเวลา เมื่อทำการตั้งค่าคาบเวลาที่น้อยเกินไป จะทำให้เกิดการใช้งานซีพียูที่สูงเกินไปทั้งฝั่งแม่ข่ายและลูกข่าย ทั้งนี้ผู้วิจัยได้ทำการเขียนโปรแกรมเพิ่มในส่วนของการปรับแต่งค่าคาบเวลาที่ฝั่งแม่ข่าย เพื่อให้มีความเหมาะสมกับสภาพการใช้งานแบบเวลาจริง คือ ถ้ามีการใช้งานซีพียูที่สูงขึ้นมากอันเนื่องมากจากจำนวนการเชื่อมต่อที่เพิ่มขึ้น โปรแกรมจะทำการลดคาบเวลาลงเพื่อลดการใช้งานของซีพียู และจะปรับเข้าสู่ระบบเดิมเมื่อมีการใช้งานซีพียูที่ลดลงแล้วเพื่อให้เกิดประสิทธิภาพสูงที่สุด

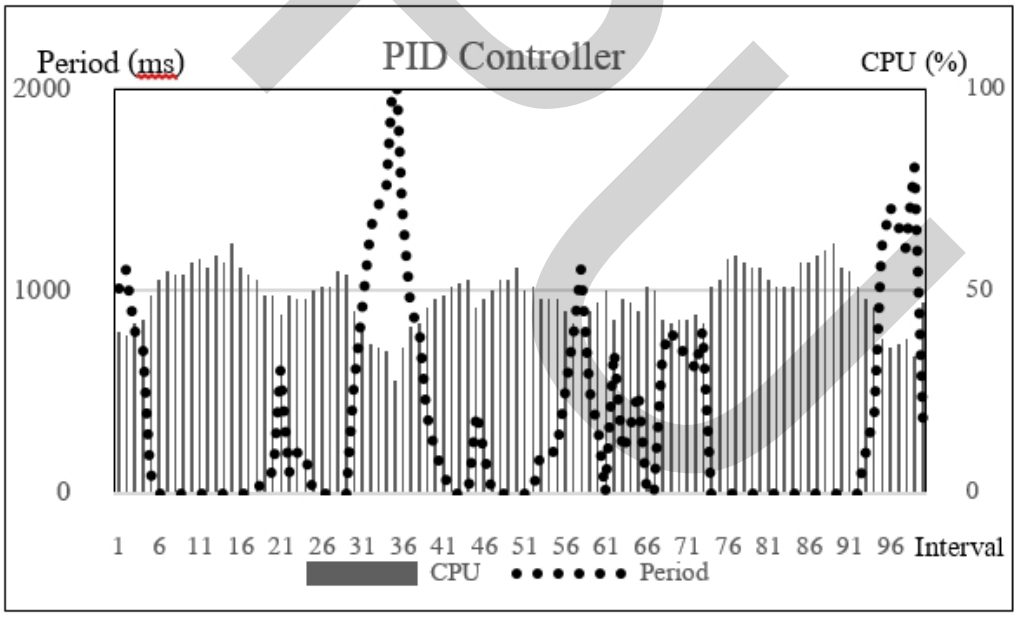
โดยสาเหตุของการปรับแต่งที่เครื่องแม่ข่ายเพียงอย่างเดียวนั้น ผู้วิจัยได้ทำการทดสอบการปรับแต่งการรับค่าข้อมูลของเว็บเบราว์เซอร์ผ่านทางโนดด็อทเจเอสที่เครื่องคอมพิวเตอร์ลูกข่าย โดยมีการใช้ตัวนับ (Counter) n ครั้งก่อนการเริ่มประมวลผล เพื่อหน่วงการทำงานของเว็บเบราว์เซอร์ ซึ่งก็ช่วยให้การทำงานของซีพียูลดลงมาได้ระดับหนึ่ง แต่ในการที่จะทำให้ตัวโปรแกรมสามารถทำงานโดยอัตโนมัตินั้น จะต้องมีการตรวจสอบการทำงานของซีพียูที่เครื่องคอมพิวเตอร์ลูกข่ายแบบเวลาจริง โดยจะต้องทำการเขียนโปรแกรมเพิ่มเติมเพื่อคำนวณการทำงานของซีพียูแบบเวลาจริง ซึ่งเป็นการเพิ่มภาระให้แก่เครื่องคอมพิวเตอร์ลูกข่ายโดยไม่จำเป็น ผู้วิจัยจึงได้ทำการเน้นการปรับปรุงที่เครื่องคอมพิวเตอร์แม่ข่ายเป็นหลักแทน

ค่าผลลัพธ์ (Output) ที่ได้ คือ ค่าของคาบเวลาในการส่งข้อมูล ที่โปรแกรมจะต้องทำการปรับเปลี่ยน เพื่อให้ค่าของการทำงานของซีพียูขณะนั้น (Actual CPU) เข้าใกล้กับค่าการทำงานของซีพียูเป้าหมาย (Set Point) นั่นเอง

ผลการทดลองเปรียบเทียบการปรับค่าคาบเวลาทั้ง 2 แบบ คือ การปรับโดยการใช้ค่าคงที่ และการปรับโดยการใช้ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ เพื่อให้ค่าซีพียูเป้าหมาย เป็นไปตามที่กำหนดไว้ ได้ผลการทดลอง ดังภาพที่ 4.8 และภาพที่ 4.9



ภาพที่ 4.8 การทำงานของซีพียู แบบการปรับคาบเวลาแบบค่าคงที่



ภาพที่ 4.9 การทำงานของซีพียู แบบระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์

วิเคราะห์ผลการทดสอบ

การปรับค่าคาบเวลาแบบค่าคงที่ จะมีผลต่อการเปลี่ยนค่าคาบเวลาอย่างต่อเนื่องไปอย่างช้าๆ ทำให้การเปลี่ยนแปลงของซีพียูเป็นไปอย่างช้าๆ ด้วยเช่นกัน จากภาพที่ 4.8 จะเห็นได้ว่า การทำงานของซีพียูมีการเปลี่ยนแปลงห่างไปจากค่าเป้าหมายอยู่มาก เฉลี่ยอยู่ที่ 19.62%

การปรับค่าคาบเวลาโดยใช้ระบบควบคุมพีไอดี จากภาพที่ 4.9 จะเห็นได้ว่ามีการเปลี่ยนแปลงค่าคาบเวลาอย่างรวดเร็ว เนื่องจากมีการนำค่าผิดพลาดมาใช้ในการคำนวณด้วย จึงทำให้การควบคุมการทำงานของซีพียูเป็นไปได้อย่างราบรื่นกว่าการปรับค่าคาบเวลาแบบค่าคงที่ แต่ในบางครั้งอาจทำให้เกิดการปรับค่าคาบเวลาที่สูงมากเช่นกัน โดยมีค่าห่างจากค่าเป้าหมายเฉลี่ยอยู่ที่ 11.16%

DPU

บทที่ 5
สรุปผล และข้อเสนอแนะ

จากวัตถุประสงค์ของการวิจัยที่ได้กำหนดไว้ ผู้วิจัยได้ทำการวิจัยมีรายละเอียด ดังนี้

1. เพื่อทำการปรับปรุงการเขียนโปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำในท่อประธานของการประปานครหลวง จากภาษาจาวาสคริปต์เป็นภาษาโนดด็อทเจเอส

จากผลการทดลองเรื่อง (1) การเปรียบเทียบการทำงานของซีพียูระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ ทางด้านฝั่งเครื่องคอมพิวเตอร์แม่ข่ายและลูกข่าย (2) การเปรียบเทียบปริมาณการร้องขอข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์จากทางฝั่งคอมพิวเตอร์ลูกข่าย (3) การเปรียบเทียบปริมาณการส่งผ่านข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ ผ่านทางเว็บเบราว์เซอร์ และ (4) การเปรียบเทียบค่าเวลาในการตอบสนองของข้อมูลเปรียบเทียบระหว่างจาวาสคริปต์และโนดด็อทเจเอส สามารถสรุปได้ว่า การนำภาษาโนดด็อทเจเอสเข้ามาใช้งานแทนจาวาสคริปต์ สามารถใช้งานแทนกันได้อย่างสมบูรณ์ คือ ปริมาณข้อมูลส่งผ่านเว็บเบราว์เซอร์มีค่าเท่ากันเมื่อมีการใช้ข้อมูลเดียวกัน จำนวนการร้องขอข้อมูลของภาษาโนดด็อทเจเอสน้อยกว่าของจาวาสคริปต์ การทำงานของซีพียูฝั่งลูกข่ายของภาษาโนดด็อทเจเอสมีค่าน้อยกว่าจาวาสคริปต์ และเวลาในการตอบสนองของข้อมูลของภาษาโนดด็อทเจเอสน้อยกว่าจาวาสคริปต์

ดังนั้น การปรับปรุงการเขียนโปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำในท่อประธานของการประปานครหลวง จากภาษาจาวาสคริปต์เป็นภาษาโนดด็อทเจเอส จึงสำเร็จตามจุดประสงค์ในข้อนี้

2. เพื่อทำการเพิ่มประสิทธิภาพการทำงานของโปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำในท่อประธาน ได้แก่ การลดการร้องขอการเข้าถึงฐานข้อมูล การลดปริมาณข้อมูลส่งผ่านภายในเว็บเบราว์เซอร์ และเพิ่มความเร็วในการเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบัน จากเดิม 60 วินาที ให้เหลือไม่เกิน 5 วินาที

จากผลการทดลองเรื่อง (1) การเปรียบเทียบปริมาณการร้องขอข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์จากทางฝั่งคอมพิวเตอร์ลูกข่าย (2) การเปรียบเทียบปริมาณการส่งผ่านข้อมูลระหว่างการเขียนโปรแกรมภาษาโนดด็อทเจเอสและจาวาสคริปต์ ผ่านทางเว็บเบราว์เซอร์ และ (3) การเปรียบเทียบค่าเวลาในการตอบสนองของข้อมูลเปรียบเทียบระหว่างจาวาสคริปต์และโนดด็อทเจเอส สามารถสรุปได้ว่า ปริมาณข้อมูลส่งผ่านเว็บเบราว์เซอร์มีค่าเท่ากัน

เมื่อมีการใช้ข้อมูลเดียวกัน จำนวนการร้องขอข้อมูลของภาษาโนดด็อทเจเอสน้อยกว่าของจาวาสคริปต์ การทำงานของซีพียูฝั่งลูกข่ายของภาษาโนดด็อทเจเอสมีค่าน้อยกว่าจาวาสคริปต์ และเวลาในการตอบสนองของข้อมูลของภาษาโนดด็อทเจเอสน้อยกว่าจาวาสคริปต์

ดังนั้น การเขียนโปรแกรมภาษาโนดด็อทเจเอสมีประสิทธิภาพมากกว่าภาษาจาวาสคริปต์ ซึ่งสำเร็จตามจุดประสงค์ในข้อนี้

3. เพื่อทำการวิจัยและพัฒนาส่วนเสริมโปรแกรมที่มีการปรับสภาพการทำงานซีพียู (CPU Usage) ให้เหมาะสมกับทรัพยากรของเครื่องคอมพิวเตอร์แม่ข่ายแบบอัตโนมัติ โดยสามารถตั้งค่าช่วงการทำงานของซีพียูได้ เพื่อให้การทำงานของโปรแกรมไม่กระทบกับการทำงานของแอปพลิเคชันอื่นๆ ภายในเครื่องแม่ข่าย

โปรแกรมมีการปรับปรุงอัลกอริทึมที่ใช้ในการควบคุมการทำงานของซีพียูของเครื่องแม่ข่ายให้อยู่ในช่วงที่ต้องการและมีความเหมาะสมกับทรัพยากรทางคอมพิวเตอร์ของฝั่งแม่ข่ายโดยนำทฤษฎีของระบบควบคุมพีไอดีมาใช้ในการคำนวณ ให้ผลลัพธ์ต่างจากค่าเป้าหมายเฉลี่ย 11.16 เปอร์เซ็นต์ ซึ่งมีประสิทธิภาพดีกว่าการปรับแบบคงที่ ที่มีผลลัพธ์ต่างจากค่าเป้าหมายเฉลี่ย 19.62 เปอร์เซ็นต์ จึงทำให้การวิจัยนี้ได้ผลสำเร็จตามวัตถุประสงค์ในข้อนี้

ดังนั้น การปรับปรุงโปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำในท่อประธานด้วยการเขียนโปรแกรมภาษาโนดด็อทเจเอสแทนภาษาจาวาสคริปต์ จึงสำเร็จตามจุดประสงค์ที่ตั้งไว้ทุกหัวข้อ

5.1 ข้อได้เปรียบของการใช้การเขียนโปรแกรมภาษาโนดด็อทเจเอสแทนภาษาจาวาสคริปต์

5.1.1 สามารถใช้งานภาษาโนดด็อทเจเอสแทนภาษาจาวาสคริปต์ได้อย่างสมบูรณ์ ในเรื่องของการส่งผ่านข้อมูลผ่านเว็บเบราว์เซอร์ที่เท่ากัน ปริมาณการร้องขอข้อมูลที่ลดลง การทำงานของซีพียูของเครื่องลูกข่ายที่ลดลง และเวลาในการตอบสนองของข้อมูลลดลง

5.1.2 ประสิทธิภาพการทำงานสูงกว่าการใช้ภาษาจาวาสคริปต์ เนื่องจากมีปริมาณการร้องขอข้อมูลที่ลดลง การทำงานของซีพียูของเครื่องลูกข่ายที่ลดลง และเวลาในการตอบสนองของข้อมูลลดลง

5.1.3 มีความเหมาะสมกับโปรแกรมที่มีการใช้งานฟังก์ชันของจาวาสคริปต์ เนื่องจากการรับส่งข้อมูลจะอยู่ในรูปของตัวแปรจาวาสคริปต์เป็นหลัก

5.1.4 สามารถกำหนดค่าคาบเวลาในการรับ-ส่งข้อมูลได้รวดเร็วกว่าการใช้ภาษาจาวาสคริปต์ เนื่องจากการเขียนโปรแกรมภาษาโนดด็อทเจเอสมีโครงสร้างที่สามารถรับ-ส่งข้อมูลได้อย่างรวดเร็วกว่าภาษาจาวาสคริปต์มาก

5.2 ข้อเสียเปรียบของการใช้สถาปัตยกรรมโนดด็อทเจเอสแทนภาษาจาวาสคริปต์

5.2.1 การทำงานของซีพียูที่เครื่องคอมพิวเตอร์ฝั่งแม่ข่ายสูง เนื่องจากมีการวนลูปเพื่อส่งข้อมูลตลอดเวลา

5.2.2 มีการติดตั้งระบบที่ซับซ้อนกว่าภาษาจาวาสคริปต์แบบปกติ เนื่องจากต้องทำการเขียนโปรแกรมเพื่อรับ-ส่งข้อมูล ทั้งฝั่งแม่ข่ายและลูกข่าย ให้สอดคล้องกัน

5.3 สรุปผล

โปรแกรมตรวจสอบและแสดงผลข้อมูลแรงดันน้ำในท่อประธานของการประปานครหลวง ที่มีการเขียนโปรแกรมภาษาโนดด็อทเจเอสเพื่อปรับปรุงการแสดงผลแรงดันน้ำในท่อประปาแบบเวลาจริง สามารถรับ-ส่งข้อมูลได้ภายใน 5 วินาที อย่างต่อเนื่อง สามารถลดปริมาณการร้องขอข้อมูล ลดการทำงานของซีพียูของเครื่องลูกข่าย และลดเวลาในการตอบสนองของข้อมูล จึงทำให้โปรแกรมมีประสิทธิภาพที่สูงขึ้น

นอกจากนี้การปรับปรุงอัลกอริทึมที่ใช้ทฤษฎีของระบบควบคุมพีไอดี มาควบคุมการทำงานของซีพียูของเครื่องแม่ข่ายให้อยู่ในช่วงที่ต้องการส่งผลให้โปรแกรมแสดงผลและตรวจสอบแรงดันน้ำในท่อประปาสามารถกำหนดช่วงการทำงานของซีพียู เพื่อให้เหมาะสมกับทรัพยากรทางคอมพิวเตอร์ของฝั่งแม่ข่าย สามารถทำให้โปรแกรมมีประสิทธิภาพมากขึ้นอีกด้วย

5.4 ข้อเสนอแนะ

ในการใช้งานสถาปัตยกรรมโนดด็อทเจเอสในงานจริง การกำหนดค่าคาบเวลาในการรับ-ส่งข้อมูลควรมีความเหมาะสมกับกับเครื่องคอมพิวเตอร์แม่ข่ายและเครื่องคอมพิวเตอร์ลูกข่ายที่ใช้งาน เนื่องจากการใช้งานโนดด็อทเจเอส จะเป็นการผลักภาระไปให้ซีพียูในการประมวลผลข้อมูลทั้งฝั่งแม่ข่ายและลูกข่าย ถ้าเครื่องที่นำมาใช้งานมีค่าการทำงานของซีพียูที่ต่ำเกินไป จะส่งผลให้เกิดการทำงานของซีพียูเกินขีดจำกัด (CPU Usage 100%) ซึ่งจะทำให้การทำงานของโปรแกรมมีความผิดพลาด และส่งผลทำให้การแสดงผลข้อมูลผิดพลาดอีกด้วย

ข้อเสนอแนะของการใช้อัลกอริทึมเข้ามาปรับปรุงค่าคาบเวลาในการรับ-ส่งข้อมูลนั้น จะทำให้เครื่องคอมพิวเตอร์มีการใช้งานซีพียูในการคำนวณการปรับค่าคาบเวลาเพิ่มขึ้นมาอีกด้วยเช่นกัน ดังนั้น ในการใช้งานจริงจึงควรมีการตั้งค่าระบบให้เหมาะสมกับขีดความสามารถของเครื่องคอมพิวเตอร์ที่ใช้ตั้งแต่แรกจึงจะให้ผลลัพธ์ที่ดีที่สุด

บรรณานุกรม

DPU

บรรณานุกรม

ภาษาไทย

ความรู้ความเข้าใจในหลักการของระบบโทรมาตร (SCADA).
สำนักบริหารจัดการน้ำและอุทกวิทยา พ.ศ. 2558
http://water.rid.go.th/hyd/scada/scada_uc.htm

ภาษาต่างประเทศ

Ariel Ortiz. 2014. “Server-side web development with JavaScript and Node.js (abstract only). In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14)”. ACM, New York, NY, USA, 747-747. DOI=10.1145/2538862.2539001 http://doi.acm.org/10.1145/2538862.2539001

Bohdan Anderson, Symon Oliver, and Patricio Davila. 2014. CoPerformance: a rapid prototyping platform for developing interactive artist-audience performances with mobile devices. In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI '14). ACM, New York, NY, USA, 605-607. DOI=10.1145/2628363.2645699 http://doi.acm.org/10.1145/2628363.2645699

Hyunjung Park and Jennifer Widom. 2014. CrowdFill: a system for collecting structured data from the crowd. In Proceedings of the companion publication of the 23rd international conference on World wide web companion (WWW Companion '14). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 87-90. DOI=10.1145/2567948.2577029 http://dx.doi.org/10.1145/2567948.2577029

JQuery. (2015). “JQuery” Retrieved January 5, 2015 from https://jquery.com/.

Node.js. (2015). “Node.js” Retrieved January 1, 2015 from https://nodejs.org/.

Socket.IO. (2015). “Socket.IO” Retrieved January 5, 2015 from https://socket.io/.

Strongloop. (2015). “What Makes Node.js Faster Than Java?” Retrieved October 01, 2015 from https://strongloop.com/strongblog/node-js-is-faster-than-java/

ประวัติผู้เขียน

ชื่อ-นามสกุล	วรพจน์ ไชยพรพัฒนา
ประวัติการศึกษา	พ.ศ. 2550 วิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ตำแหน่งและสถานที่ทำงานปัจจุบัน	วิศวกรไฟฟ้า การประปานครหลวง
ประสบการณ์ ผลงานทางวิชาการ	พ.ศ. 2558 การปรับปรุงโครงสร้างโปรแกรมตรวจสอบข้อมูลแรงดันน้ำในท่อประปาแบบเวลาจริงด้วยสถาปัตยกรรมโนดด็อทเจเอส (Node.js) งานประชุมวิชาการ งานวิจัย และพัฒนาเชิงประยุกต์ ครั้งที่ 7 (ECTI-CARD 2015) พ.ศ. 2560 การปรับปรุงอัลกอริทึมสำหรับการจัดการการทำงานของซีพียูแบบอัตโนมัติของโปรแกรมตรวจสอบแรงดันน้ำในท่อประปาของการประปานครหลวง ที่มีการใช้งานโนดด็อทเจเอสในการรับ-ส่งข้อมูล งานประชุมการประชุมเครือข่ายวิชาการด้านวิศวกรรมไฟฟ้า การพัฒนานวัตกรรมเพื่ออุตสาหกรรมและการเกษตรอย่างยั่งยืน ครั้งที่ 9 (EENET2017)

DPU