

## รายงานความก้าวหน้าวิชา CE Project

ครั้งที่ 1

ระหว่างวันที่ 01 ส.ค. 65 ถึงวันที่ 27 ส.ค. 64

1. ชื่อโครงการ (อังกฤษ) Image Processing Application using Task Scheduling on Network System

2. การดำเนินงานมีความก้าวหน้า 2% (ใช้ค่า % **Complete** จาก MS Project)

มีความก้าวหน้าเพิ่มขึ้นจากรายงานความก้าวหน้า ครั้งก่อน 2%

☐ เร็วกว่าแผน 0 วัน ☐ ช้ากว่าแผน 0 วัน

3. รายละเอียดความก้าวหน้า

ในส่วนของการศึกษา ค้นคว้า และ ออกแบบ

สำหรับการพัฒนาในส่วน Front end ของ web application ทางผู้เรียนได้ทำการศึกษาในเรื่องของการทำ SPA(Single-Page Applications) เพื่อต้องการความถี่ในการใช้งานตัว web application โดยได้ทำการ เลือก framework ต่าง ๆ ที่น่าสนใจมาดังนี้

- 1.Django
- 2.React.js
- 3.Vue.js
- 4.Flask
- 5.Angular

โดยหลังจากที่ได้ศึกษาหาข้อมูลเพิ่มเติมและทดลองใช้งานแต่ละ framework ก็พบว่า Vue.js นั้นเหมาะสมกับการนำมาใช้ในการพัฒนา Front end ของ web application มากที่สุด เนื่องจาก สามารถที่จะจัดการทำ element ที่แสดงผลบนหน้าเว็บได้โดยง่าย และ พื้นฐานในการพัฒนามาจากภาษา HTML และ Java Script ที่ทางผู้เรียนมีพื้นฐานอยู่แล้ว

จึงทำให้ผู้เรียนได้เลือกที่จะใช้ Vue.js ในการพัฒนา Front end ของ web application และทางผู้เรียนก็ได้ออกแบบหน้า Front end ของ Web Application ไว้คร่าว ๆ ดังรูปต่อไปนี้



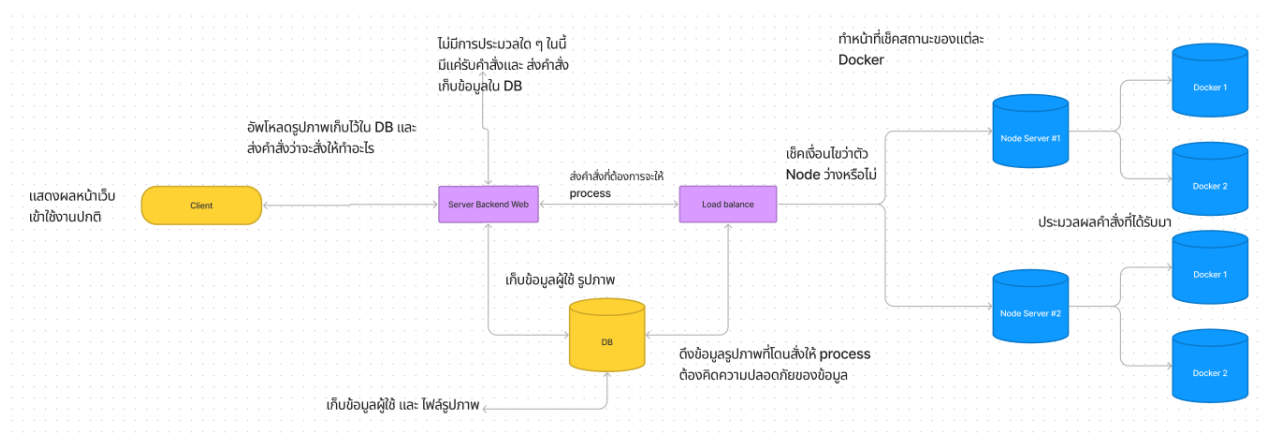
รูปภาพของส่วน Front end ของ Web Application ที่ออกแบบไว้เบื้องต้น

สำหรับการพัฒนาในส่วน Back end ของ web application ทางผู้เรียนนั้นต้องการ framework ที่มีความปลอดภัยและสามารถขยายเพิ่มเติมขึ้นได้หากมีผู้ใช้งานตัว web application จำนวนมากขึ้น เลือก framework ต่าง ๆ ที่น่าสนใจมาดังนี้

1. Django
2. Flask
3. Angular

โดยหลังจากที่ได้ศึกษาหาข้อมูลเพิ่มเติมและทดลองใช้งานแต่ละ framework ก็พบว่า Django นั้นเหมาะสมกับการนำมาใช้ในการพัฒนา Back end ของ web application มากที่สุด เนื่องจาก มีความปลอดภัยที่ดี สามารถที่จะจัดการผู้ใช้ได้ มีสามารถในการขยายเพื่อรองรับผู้ใช้ที่มีมากขึ้น และ พื้นฐานในการพัฒนามาจากภาษา Python ที่ทางผู้เรียนมีพื้นฐานอยู่แล้ว

จึงทำให้ผู้เรียนได้เลือกที่จะใช้ Django ในการพัฒนา Back end ของ web application และ ผู้เรียนได้ออกแบบโครงสร้างการทำงานของ Web Application Server และ Database ไว้ดังนี้



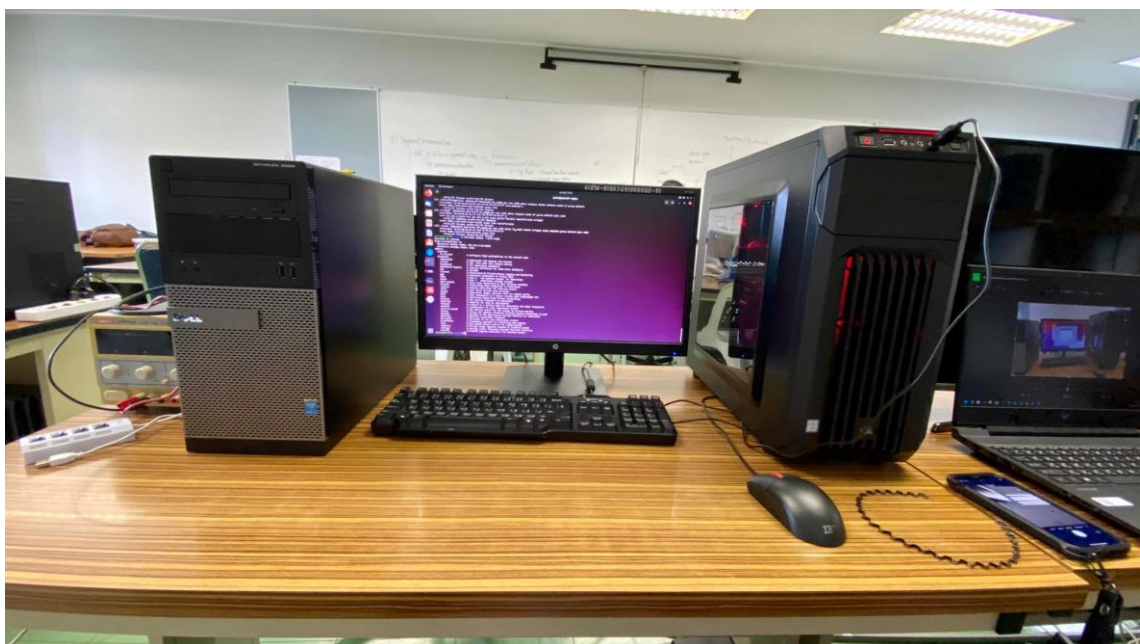
รูปภาพของโครงสร้างการทำงานของ Web Application Server และ Database

สำหรับการพัฒนาในส่วนของ Server ผู้เรียนได้ศึกษาการทำงานของ Cluster network และการทำงานของ Load balancer โดยโปรแกรมที่ใช้ในการศึกษา และ ทดลองมีดังนี้

1. Ubuntu 20.02 สำหรับเป็น ระบบปฏิบัติการ ของ Server
2. Kubernetes(MicroK8S) สำหรับจัดการ Cluster , Monitor และ จัดการ Container
3. Docker สำหรับจัดการ Docker Images
4. Nginx สำหรับจัดการ Load Balancer ตัว Web Service
5. Grafana และ Prometeuse โดยทั้ง 2 เป็นตัวสร้างข้อมูลเป็น time-series data ของตัว Cluster จาก Kubernetes ขึ้นมาและ visualizations ข้อมูล ทรัพยากรที่ใช้ เช่น CPU , Memory และ GPU

จากการศึกษาเนื่องจากเราจะจัดทำเกี่ยวกับ Cluster Computing จึงต้องมีเครื่องอย่างน้อย 2 เครื่องขึ้นไป เพื่อช่วยกันกระจายงานการประมวลผลภาพไปในแต่ละเครื่องจึงได้เตรียมเครื่องคอมพิวเตอร์ 1 ตัวมาทำเป็นเครื่อง Server โดยได้ใช้งานเครื่องคอมพิวเตอร์จาก DIP Lab มีสเปคเครื่องคอมพิวเตอร์ดังนี้

- CPU : i9-9900K 3.60GHz 16 Core
- RAM : 32 GB
- HDD : 4 TB
- GPU : RTX 2060 6 GB



รูปภาพของเครื่องคอมพิวเตอร์จาก DIP Lab

ในส่วนของ Server นั้นต้องการที่จะทำเป็น Cluster Computing จึงได้เตรียมเครื่อง Server ไว้ 1 เครื่อง และ สร้าง VM จำลองเครื่องขึ้นมาอีก 2 เครื่องเป็นเครื่อง Server สำหรับทำ Cluster โดยมีการเตรียมการดังนี้

- Server#1 OS Ubuntu 20.02 CPU 8 Core, Ram 16 GB, GPU RTX 2060
- Server#2 OS Ubuntu 20.02 CPU 4 Core, Ram 8 GB
- Server#3 OS Ubuntu 20.02 CPU 4 Core, Ram 8 GB

เพื่อให้ระบบ Cluster มีประสิทธิภาพสูงสุดจึงได้ศึกษาทำให้ Cluster ทำงานในรูปแบบของ HA หรือ High Availability เพื่อแก้ปัญหาในกรณีที่เครื่องตัวใดตัวหนึ่งขัดข้องไป และ ยังได้ทำให้ทั้ง 3 เครื่องนั้นอยู่ในวงเครือข่ายเดียวกัน (LAN) สำหรับการเชื่อมต่อเป็น Cluster และ ได้มีการจำลองตัว shared folder ขึ้นมาเพื่อให้ไว้สำหรับทั้ง 3 เครื่องมาจัดเก็บไฟล์ข้อมูล และ ประมวลผลภาพภายใน Folder เดียวกัน

โดยเนื่องจากต้องการที่จะให้ทำงานกระจายงานประมวลผลไปตามที่รับงานมาจากส่วนของ Web application จึงได้เตรียมส่วนประมวลผลของงานที่ได้รับมาไว้เป็น Python Script จึงได้หาวิธีการจัดทำทั้ง 3 เครื่องเป็น Cluster และ สามารถช่วยกระจายงานจากคำสั่งของ Python Script ได้จึงได้นำตัวของ Docker เข้ามาใช้งาน

จากการศึกษาการใช้ตัวของ Docker ในการจัดการ Python Script และ web application ให้อยู่ในรูปของ Docker Images เพื่อจัดการเรื่องของ Environment ต่าง ๆ ของ Python Script และ web application ให้ทำงานได้อย่างมีประสิทธิภาพ และ ใช้งานทรัพยากรได้เหมาะสมที่สุด และ เนื่องจากการที่จะส่งงาน Docker Images และ ตัวเครื่องอยู่ในรูปแบบของ Cluster จึงได้ใช้ Cluster management software ที่ชื่อ Kubernetes

ในส่วนของ Kubernetes นั้นไว้สำหรับจัดการ Cluster, Containers และ Monitor Node ต่าง ๆ ที่ได้ทำเป็น Cluster ไว้ ซึ่งตัว Kubernetes ตัวสมบูรณ์นั้นมีการใช้ทรัพยากรพื้นที่ของเครื่อง และ ใช้ทรัพยากรภายในเครื่องมากเกินไป สำหรับตัว Project นี้ จึงได้ใช้ตัว MicroK8S ซึ่งเป็น ตัว Kubernetes ที่มีขนาดเล็ก ใช้ทรัพยากรค่อนข้างน้อย และ ยังสามารถติดตั้งใช้งานในส่วนของการ Monitor อื่น ๆ ได้เทียบเท่าตัวของ Kubernetes ตัวหลักทางเราจึงเลือกใช้ตัวของ MicroK8S ในการจัดการ Cluster ,Containers และ Monitor Node ต่าง ๆ เพื่อให้ใช้ทรัพยากรมาจัดการงานประมวลผลได้อย่างมีประสิทธิภาพ

#### 4. ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

ปัญหาที่เกิดขึ้นในส่วนการส่วนของการศึกษา ค้นคว้า และ ออกแบบ คือ การที่ไม่ทราบ Keyword ในการค้นหาเริ่มต้นจึงทำให้การเริ่มต้นของการศึกษา ค้นคว้า นั้นค่อนข้างทำได้ยากลำบาก จึงได้แก้ปัญหาโดยการปรึกษากับทางอาจารย์ที่ปรึกษา และ รุ่นพี่ที่มีความถนัดในด้านการเขียน Web Application ในรูปแบบ SPA และ การทำ Infrastructure

## 5. สิ่งที่จะดำเนินการต่อไป

สิ่งที่จะดำเนินการต่อไปคือ

- ออกแบบหน้า Front end ของ Web Application เพิ่มเติมจากเดิมให้ครอบคลุมทั้งหมด
- ออกแบบฟังก์ชันต่าง ๆ ที่จะใช้ภายในส่วน Back end ของ Web Application
- เชื่อมต่อ Server ให้เป็นระบบ Cluster
- ทำระบบ Cluster ให้อยู่ในรูปของ HA หรือ High Availability
- จำลองพื้นที่เก็บข้อมูลในรูปแบบของ NAS ขึ้นมา
- ทดสอบระบบ Server และ วัดผลประสิทธิภาพการประมวลผลบน Cluster และ Server ธรรมดา