PiEdge Cloud

An economic solution to a not so economic problem

Group #4

AGENDA

01 BACKGROUND

, Team member responsibilities timeline of milestones.

02 OUR PROJECT

Objective and scope of the project.

03 RESOURCES

What did we use to make our project?

04 OUR FINDINGS

What results were yielded from our work?

05 CLOSING

Questions
Thank You
Anonymous Survey

01

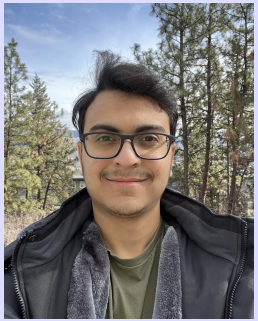
Background

Patrick

Team Lead
Responsible for quality assurance, and maintaining project documentation.

Johan

Lead for machine learning oversees integration of python-based distributed algorithms for real-time video processing



Saad

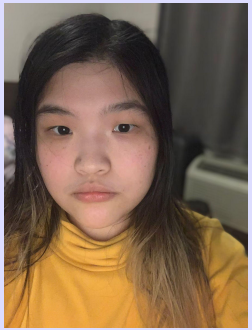
Project Manager, lead for system architecture and cluster configuration, project planning, and milestone tracking

Micheal

Web Development Lead, focusing on the implementation and integration of both frontend and backend of our website

Jia Yi (Joey)

Operation and Maintenance Coordinator , Manages version control and github actions



Ren

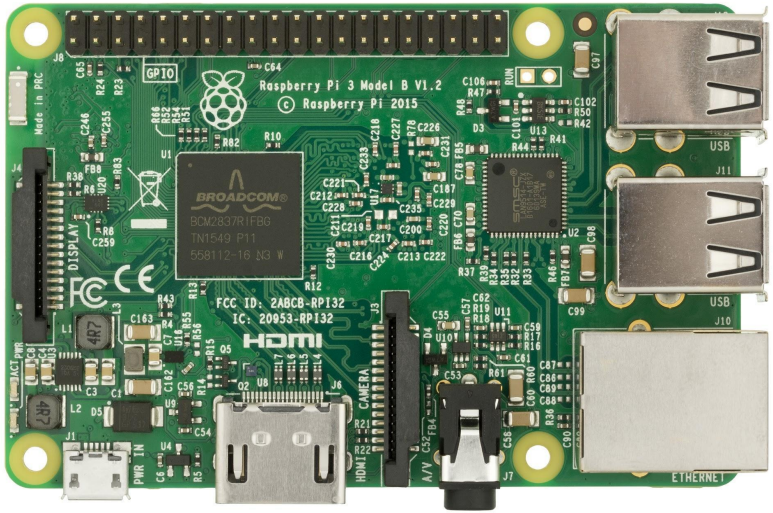
Infrastructure and Hardware Assembly Specialists, handles the overall physical implementation of the Raspberry Pi cluster

Meet The Team

We would like to acknowledge the efforts and time our team has put into the overall development of our project.

BACKGROUND: What is Our Project Anyways?

Our project aims to highlight the strengths of parallel computing when the only options that you would have access to are budget friendly (In this case the Raspberry PI) .

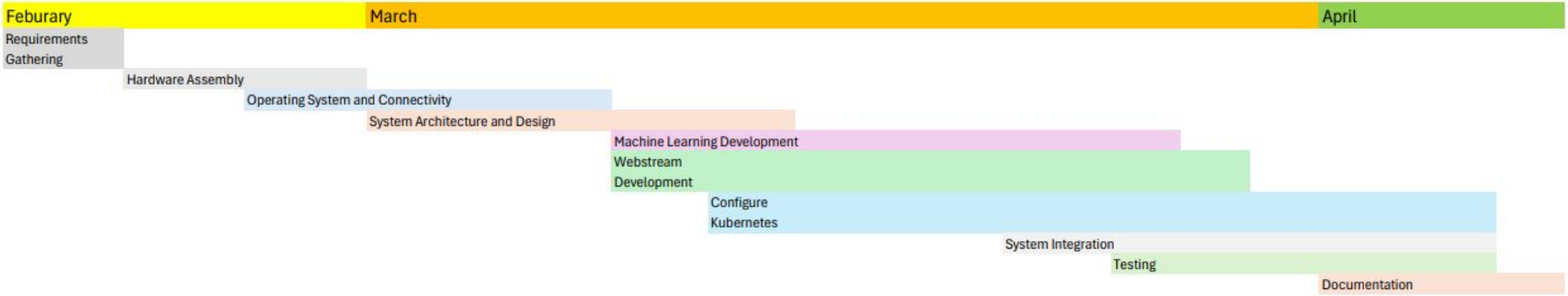


Credit: RobotShop Canada

Credit: sleeklogos.com

This allows other people to develop their own cluster computers to complete computing tasks usually relegated to bigger super centers.

Project Timeline/ Milestone Achievements



02

Our Project

Project Scope

What are our expectations for the project? To build a scalable, low-cost cloud platform using a Raspberry Pi cluster for real-time distributed computing and video streaming.

Hardware	• 3 Raspberry Pis • Network Switch • Camera Module
Software	• Containers running on Raspberry Pi OS • Next.js, Node.js, MongoDB, Nginx • Docker, Kubernetes (K3s), Visual Studio Code, Postman
Libraries	• ZMQ • Pandas • FastAPI, etc
Difficulties	• Software Compatibility issues on Raspberry pi • Deploy production environment on Pi • One of the Raspberry PIs got corrupted (sd card i/o failure)

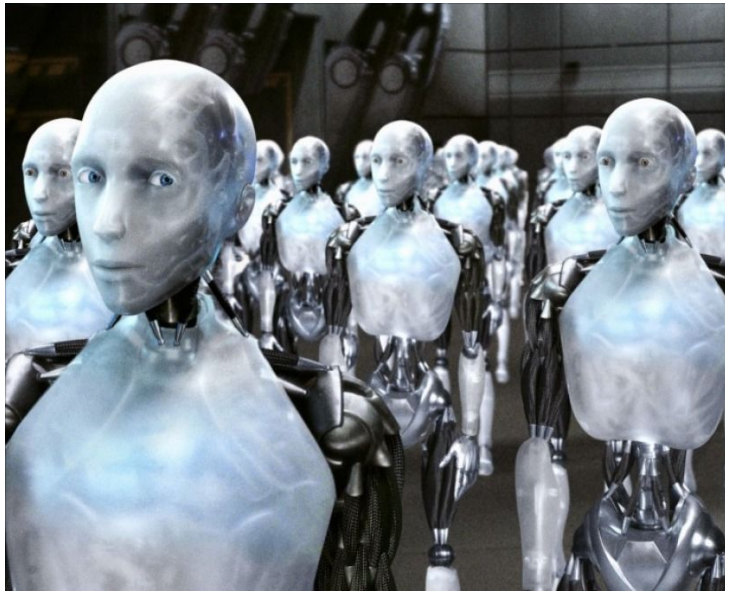
Machine Learning Aspect (The cool stuff)

As previously stated in order to showcase the Capabilities of our cluster, and as the initial prompt for the project proposed, we decided to utilize machine learning to showcase how we can divide tasks among our nodes to produce a singular outcome.

What are the tasks?

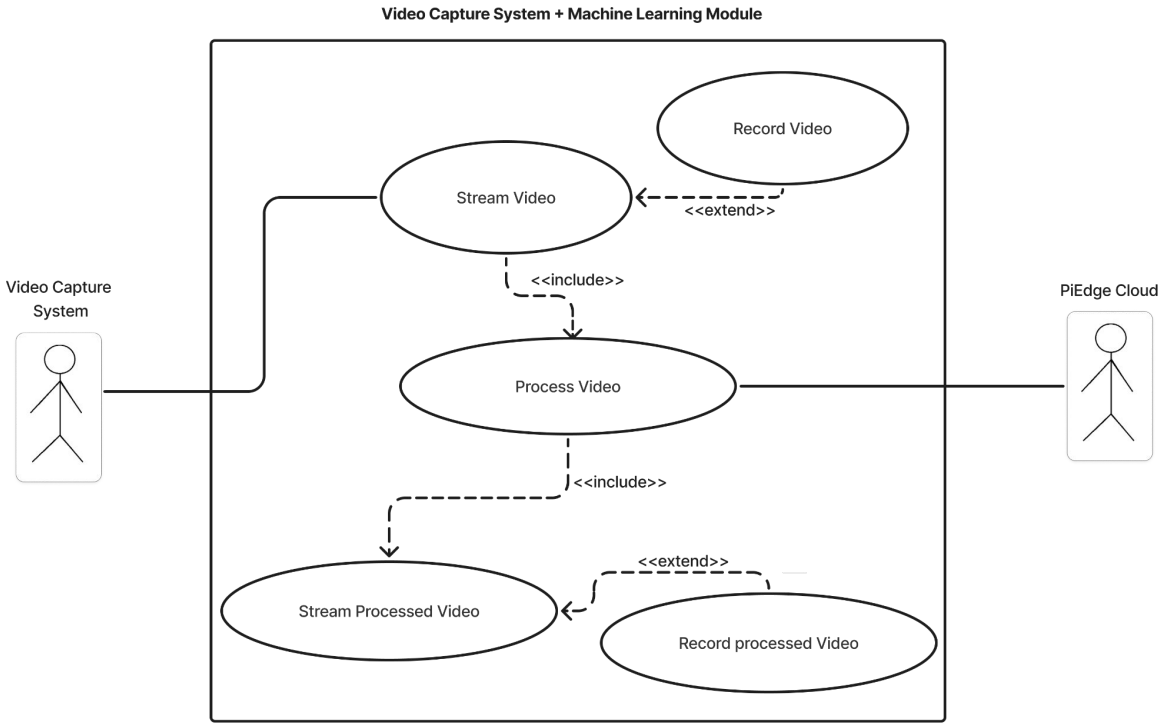
In our 3 node system, the tasks were divided as such:

- HLS Fetcher
  - A node that detects input from the streamer to allow for the process to start. It then records segments of the stream to be worked on by the inference worker
- Inference-Worker
  - Run the actual machine learning model itself allowing the model to draw boxes around things it recognizes (Faces, phones, ect..)
- Visualiser
  - The least resource impactful node acts as a bridge for the segments to be sent from the inference worker to the Mongodb.



Credit: pluggedin.com

Use Case Diagram for the Machine Learning Model*



*Some features were left out due to time restrictions

How The MLThing Works With The Project?

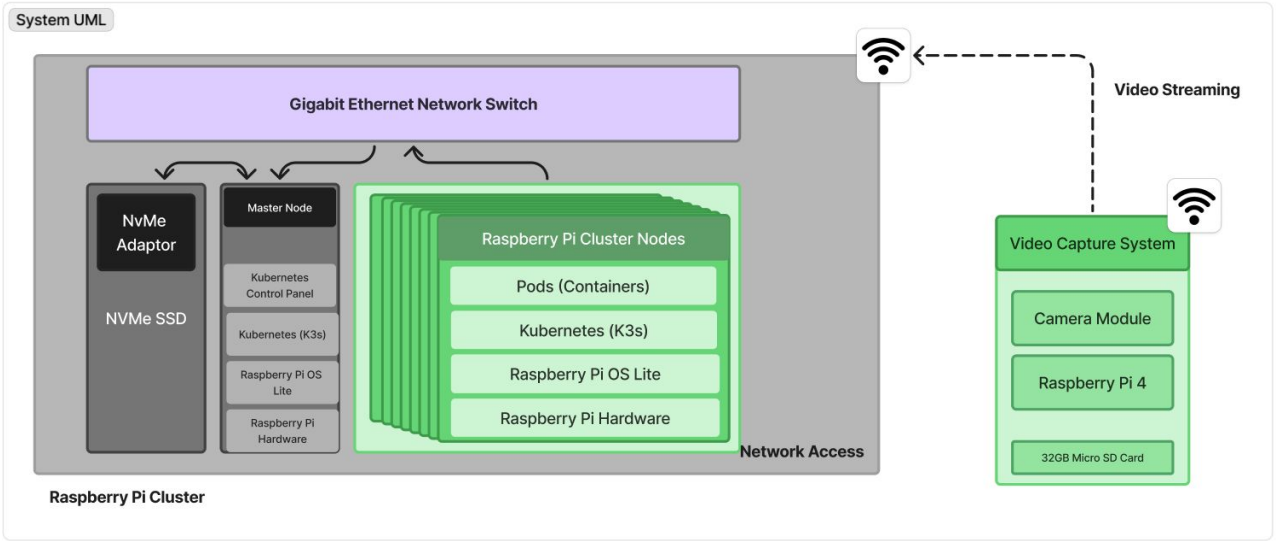
Our Project utilizes a cluster computer to work on a real time stream to efficiently complete tasks in the machine learning model process.

Our Project uses the YOLO V5 model to provide a low steaks processing stream to minimize the delay between the input and output of our model. The worker nodes work in tandem to streamline this process even more efficiently by performing specific tasks depending on the overall processing power of said machine.

The project utilizes a website with our ML model as a way to test the capabilities of our cluster computer with it being tasked with balancing an ensemble of tasks.

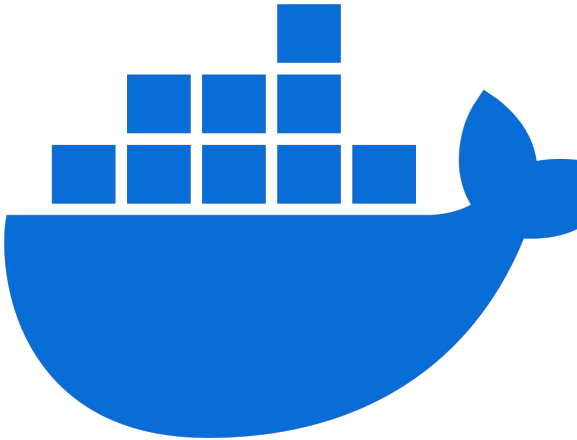
How Is Our Cluster Constructed?

Our cluster has a relatively simple
Construction process, each raspberry
Pi node is stacked on top of eachother
Using a simple cluster case. The master node is then connected to a
NvMe SSD. These nodes are then interconnected
Utilizing a network gigabit switch.



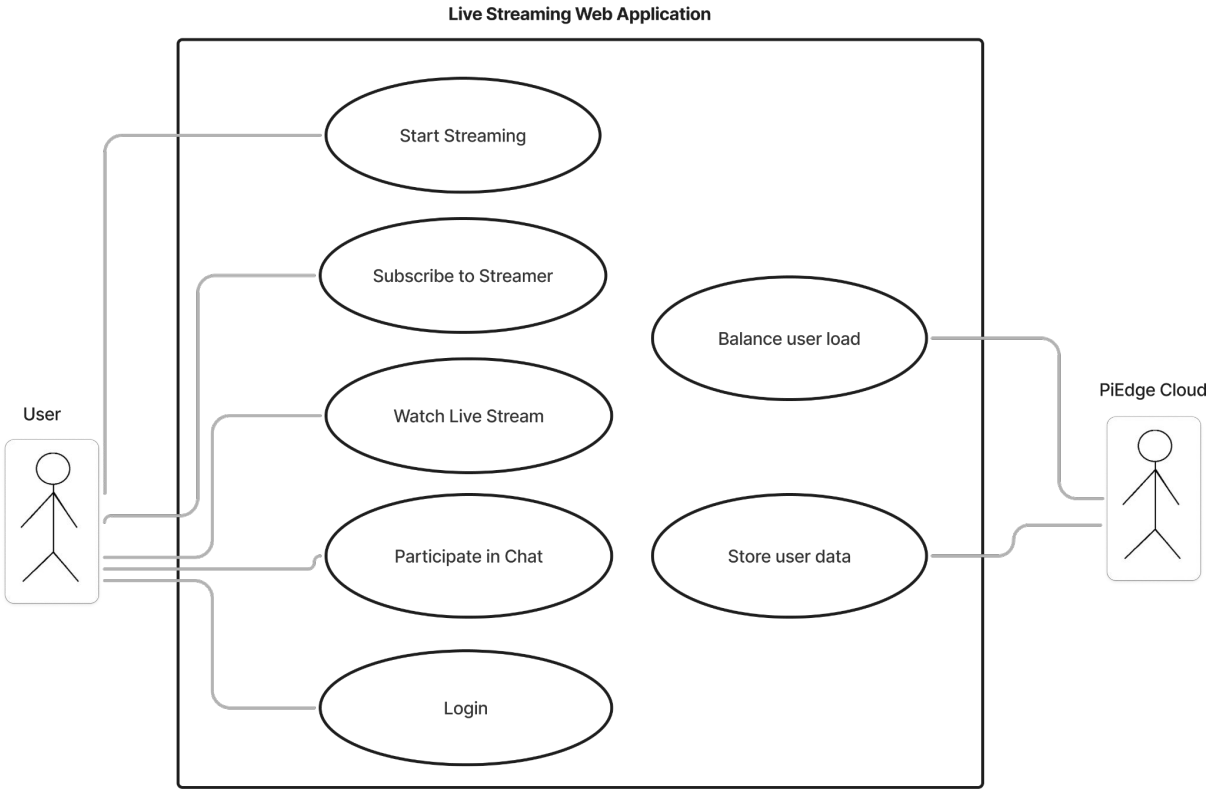
Our Website

Our website acts as a pseudo twitch clone that's able to host stream of IRL video along with a chat room. We used a combination of Kubernetes' light weight model (K3s), Docker containers (to act as our local server), and Mongodb as our database. There able to start streams of their webcam and interact with people in real time through the chat room. Additionally, Node.js was used to design the front end.



Credit: Kubernetes, Docker Swarm, mongodb

Use Case Diagram For Our Website*



*Some features were left out due to time restrictions

Video:

03

Resources
(What did we use?)

Libraries (Machine Learning in Python)

For our machine learning side of our project we utilized the following python libraries:

-ZMQ (Communication between Containers)
-Struct (Used to decode video segments)
-OS (For file handling)
-Torch (Used to run out yoloV5n model)
-PyMongo (Used to save output to Mongodb)
-Gridfs (Storing large files on MongoDB)
-Pandas (yoloV5n Dependencies)
-FastAPI (Used to create HTTP endpoints)
-Pydantic (Used with FastAPI to manage data validity)
-Subprocess (Lets you run shell commands)
-Threading (Allows for sub tasks to be ran in tandem)
-Time (used for delays)

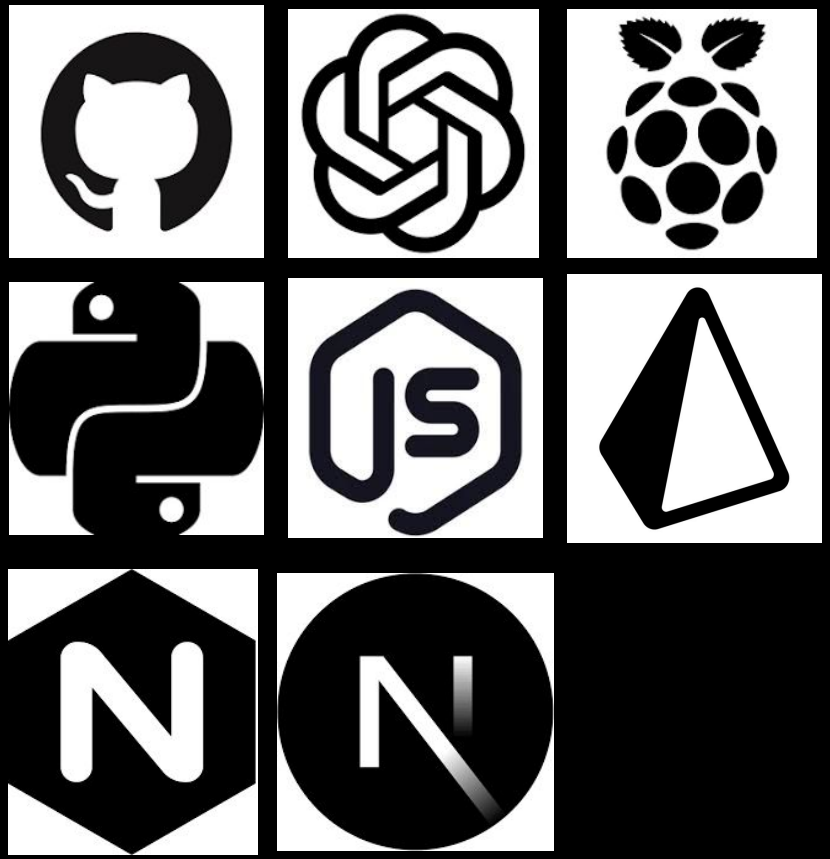
Libraries (Machine Learning) cont.

For our machine learning side of our project we utilized the following python libraries:

-Streaming Response (used to stream data in response to an hls request)
-IO (provides memory-stream handling)

Other Resources

- Github for documentation and code storage
- ChatGPT for debugging and clarification on concepts
- Raspberry Pi OS for the Pis themselves
- Python for backend Construction
- Node.Js and Prisma for front end services
- Nginx serves as a proxy for secure traffic
- Next.js serves as the fullstack framework



Credit (In Order): Seek Logo, Github, SVG Repo, Wikimedia Commons, UX Wing, World Vector Logo, Creazilla, UXwing

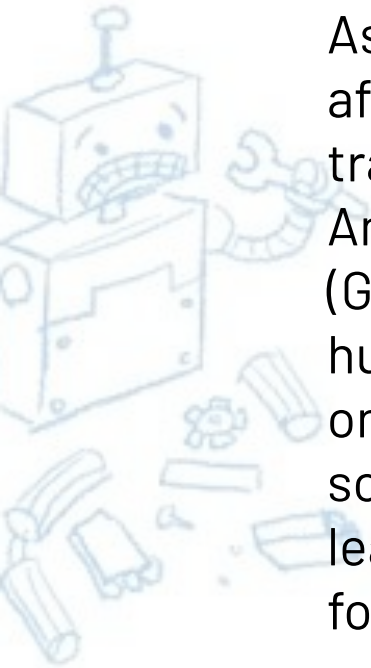
04

Our Findings

What Were Our Findings?

We found that it is possible to construct a raspberry pi cluster, and we were able to allocate resources towards accessing web pages and docker construction. We also learned that it's possible to send information over the internet to receive things like live video feed, and be able to parse segments to be put through a machine learning model. We also now have a comprehensive understanding of Kubernetes, Docker, and CI/CD pipelines.

What Are The Applications?



As stated previously our goal is to provide an affordable solution to a world that is rapidly transitioning to cloud and edge computing. Think of Amazon Web Services (AWS), or Google Cloud Platform (GCP). The annual fees can accumulate to several hundreds of dollars, so having something you purchase once to do all the things you could want is a viable solution. The application with the website and machine learning system, are both stand alone and can be used for entertainment, and surveillance.

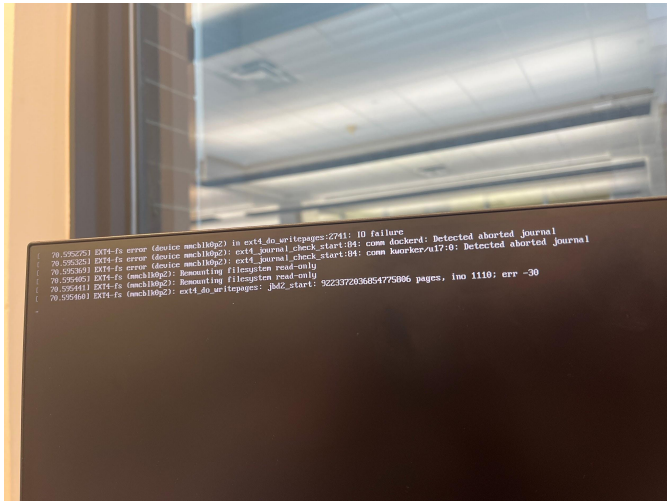
Credit: Google

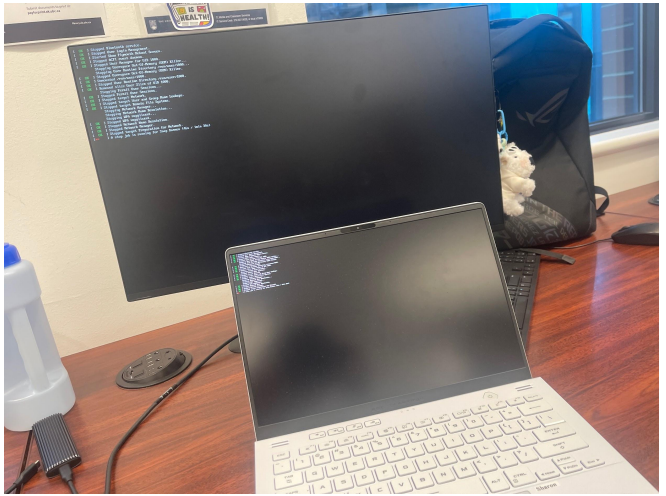
Some Unexpected Results

- During integration with software and hardware, we ran into a problem with the master node Raspberry Pi.

- Due to its weak processing power, it corrupted the OS files on its SD card and stopped booting.

- This is why we are unable to present our findings on our cluster.





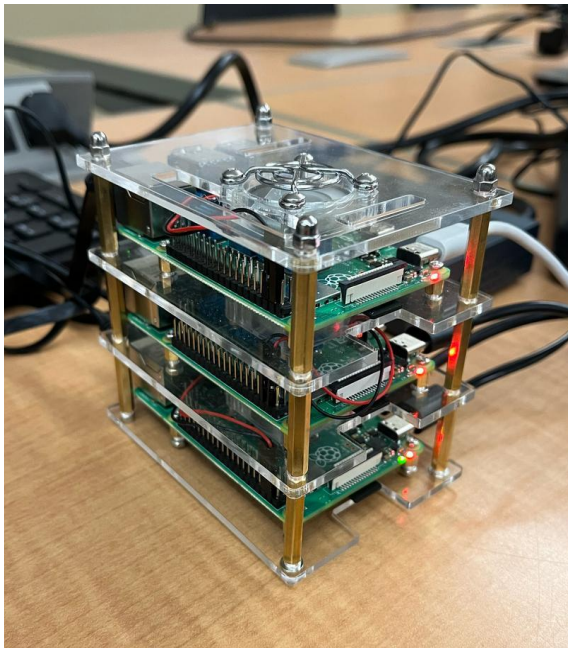
How Can We Improve In The Future?

Resources

- Higher budget
- More Time

We believe that the project was a bit out of scope for the time allocated, and if we were given at least a couple months we would be successful with our project. This, along with limited budget, forced us to downsize compared to our original design. This limited us in our processing power, which was the main reason our cluster stopped working.

Our Demo, and GitHub Pages



Website link:
https://superbolt08.github.io/PiEdgeWebsite/

05

Closing Remarks

Before that.

Questions?

Reflections

What Lessons did we learn?

Patrick

I am thankful to have people that are willing to put in the extra effort to accomplish everything they have up to now, and I am excited to see the opportunity to grow from this experience.

Johan

I think that for allowing yourself to really delve deep into the content you're responsible for can inspire motivation to continue to work on improving the quality of your work.

Saad

The better you know what you are building. The easier it gets

Michael

Translate complex technical challenges into clear, user-focused solutions. Balance system design with usability, plan detailed implementations, and communicate my ideas effectively—skills that will shape my future projects.

Joey

Give yourself ample time for work, as things may go wrong. Communicate consistently to every team member so everyone know what needs to be done

Ren

I've improved my team communication skills and enhanced my hands-on abilities through assembling a Kubernetes cluster. I'm fortunate to be part of a supportive and skilled team, which has significantly contributed to my growth.

Our Repositories

GitHubs and Website

Public Website Repository

https://github.com/superbolt08/OpenStream

Feel free to fork the repository and share your feedback!

Main Project Repository

https://github.ubc.ca/CMPE-O-246-001-02024W2/PiEdgeML

(UBC Access only)

Main Website Repository

https://github.ubc.ca/CMPE-O-246-001-02024W2/OpenStream

THANK YOU For Listening!

Qualtrics Survey

PiEdge

https://ubc.ca1.qualtrics.com/jfe/form/SV_5bip4qGicfBHEEe