Final Report

Abstract:

Computer networking is one of the marvels of the modern world. Able to connect PC's in the next room to PC's across the globe, computer networking has changed the way humans do almost every type of business, but unfortunately some business environments are not as conducive to the requirements of a good computer network as others. Namely, a manufacturing environment presents unique challenges in the form of limited spacing, fast moving machinery and people, as well as dust and other contaminants that cause extra wear and tear on the equipment. Additionally, it must be noted that most manufacturing environments feature machines that could be several decades old and have little to no native computer capability.

The goal of our project was to explore ways to improve the core networking functionality of a manufacturing environment, while also providing a live back-up for a Cloud-base server that the manufacturer uses as a database for jobs and related data. We needed to accomplish our task with a limited budget and specific set of guidelines.

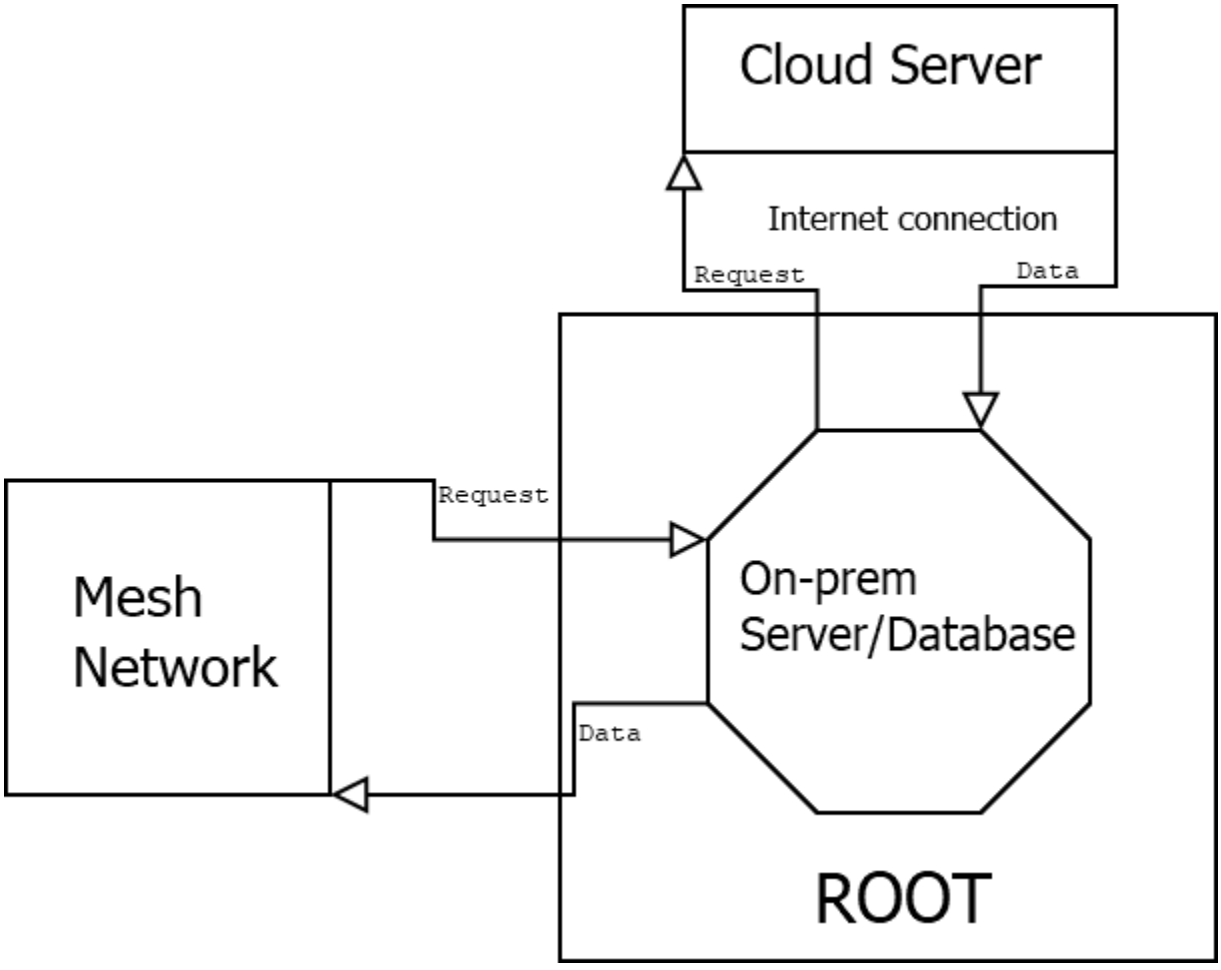
Introduction:

Automatic Spring Products Company has multiple locations with large floors full of spring production workcenters. Each workcenter is manually assigned appropriate jobs by floor workers, who are responsible for setting up the machines and interfacing with Plex. Plex is a cloud based enterprise resource planning software package used by aspc to keep track of transactions, manufacturing and supply chain details, employees, etc.. Manufacturing jobs are distributed to workstations by higher-ups in the company, however the floor workers have autonomy in deciding which jobs are in progress in a given day. Additionally, they are responsible for recording the state of the workstations throughout the production process using a kiosk to interface with Plex.

There are several problems with this setup. Firstly, employees are constantly moving between the workstations and the Plex kiosk to ensure that the cloud is updated to reflect the status of the workstation. Secondly, some material is used during machine setup that is not currently being recorded and sent to the cloud. This results in inaccurate records of available raw materials, causing unexpected shortages and production delays.

In order to solve the problems outlined earlier the client requested that two GVSU teams collaborate to set up a network of sensors in order to automate the integration between the workstations and the cloud. This achieves dual goals of increasing worker productivity and ensuring correct raw materials records.

Our group was assigned to develop a root server for the purpose of coordinating and communicating with the sensors as well as sending and receiving data from the cloud. We act as a hub for the sensors and an intermediary between the workcenters and cloud.



Root - Overview

The Root team working for Automatic Spring Products Corporation (ASPC) was responsible for developing a prototype server to act as an intermediary between the cloud and the sensors developed by the Branch & Leaf team. This project was largely exploratory and involved a lot of experimentation with different solutions. Our server is designed to act as a wifi hotspot to allow for the coordinated transmission of data between the sensors, the cloud, and our own SQLite database. The database is meant to store backup data from the cloud and sensors in case of a wifi outage. Information about spring production jobs will be pulled from the cloud and forwarded to the appropriate production stations, and data relating to material usage will be received from the sensors and pushed to the cloud. The entire server prototype runs on a Raspberry Pi 4b 2GB and is meant to interface with the Plex ERP software used by ASPC. However the most important element of this project was not necessarily the functioning code itself. The lessons we learned about the company's poorly documented ERP platform, as well as our exploration of various solutions will be very useful to ASPC as they extend the project in the coming years. Our documentation will include important details about the limits of the PLEX API, failed solutions, and advice for future developers.

Hardware

One of the most important decisions the team had to make was to decide which make and model PC we were going to use for our various tasks. An ideal setup would involve purpose built hardware of the highest quality but since we had constricting monetary and logistical requirements, we decided to use the same hardware our Branch and Leaf sub-team was using, namely Raspberry PI 4s. We felt this gave us flexibility with our hardware to do all the tasks on one machine or split them onto duplicate machines of the same type easily and cheaply if it was found that one PI could not handle the multiple tasks it was being given.

Wifi Access Point

One of the core functionalities of the 'Root' box was to act as a Wifi access point and the start of our Wifi mesh network. To accomplish this, we modified the configuration files of the 'Root' box and set it up to function as a DHCP/DNS server that provides routing and IP masquerading between the Wifi network and the internet. In this configuration, the Raspberry PI will dynamically acquire an IP from the LAN/Internet, it will create it's own Wifi LAN, set itself up with a static IP, and then service that network with a range of IP addresses.

Server

The Root node has to access as a server for Branch & Leaf to connect to. This involved constantly listening for connections and to be able to serve multiple Leaves at a time. The server was coded in python and uses threading to handle multiple connections at once. Both teams worked on a message format together so it can be easily parsed when sending data to and from nodes. One issue with Root server is that it is on a raspberry pi and may not be able to handle the upwards of hundred(s) of connections that can be going on at a single time. Our solution to this problem was recommending the sponsor use a higher quality device for the final product.

API

Our initial plan was to use Plex's Developer Portal API to communicate with the cloud server. However, the API did not have the ability to push scrap or update a workcenter's

status. We considered using Plex's Data Source API to fill in the missing functionality, but access was granted too late into development for us to integrate it into the project.

Database

Another function of the root machine is to locally store important data about the company's assets, including workcenters, production jobs, and part types. During certain procedures, such as a new workcenter being added to the network or a new job being loaded into a kiosk, the database can be used to backup data that it receives from leaf nodes and can also be used to send data to leaf nodes on the network. In addition to providing an interface for basic insert, get, update, and delete operations on an SQLite database file, the database module also provides several convenience methods to improve usability. The database also performs validation on input data to prevent errors and malicious attacks.

Teamwork Reflections

Trevor - Although development was rough, this class was definitely a valuable learning experience for me, and I feel as though my skills in both professional development and teamwork have improved as a result of this class. The ongoing pandemic naturally made this class more difficult than it would normally be, but we were able to use Discord and other online platforms to help us coordinate with each other. We also experienced some difficulty from having to coordinate with the Branch and Leaf team and making sure that our project met all of their requirements in addition to our sponsor's requirements.

David - I think that we had somewhat of a disadvantage in the sense that the ongoing pandemic made it difficult to regularly meet with the rest of the team, and maybe more importantly the client. I certainly think the project would have been more successful if we had more time to be on location together talking with the client in person. However with the situation we were given I think we performed well as a team. We worked within the given limitations and made sure to regularly communicate and coordinate among our own team and also with Branch & Leaf. Communication with the client was more difficult because we couldn't find a time that worked with everyone, but at the end of the day I know that we all put a lot of time and effort into the project.

Logan: I'm really proud of how hard our team worked on this project despite all of the obstacles we faced. Everyone pitched in when needed and we were able to meet a lot of our goals and deadlines. It's unfortunate that we weren't able to finish the project, even from an exploration aspect, but I feel like our group did everything we could do with what we were given. Hopefully it was a learning experience for everyone else, because it certainly was for me.

Ayden: Covid and the online learning environment made group projects such as this harsher than it could have been, but overall the group dynamics have been very nice. Discord has been a wonderful form of communication for us and the inter group communication with branch and leaf. Near the end of the project we had some in person meet ups with all of us, and those were great towards meeting and getting to know each other as well as working far more efficiently.

Conclusions:

We faced many issues over the course of this project but at the end our team developed a prototype for what the sponsor envisioned. We created a Root node for a mesh network, this Root can create its own hotspot to help spread a WLAN across the workplace while also acting as a DHCP server to assign ip addresses.. The Root node acts as a database and server for the various Leafs to interact with and gather information from. Finally, the Root node can act as a medium through which PLEX can get workstation and job development updates automatically.

Development was delayed throughout the project due to needing various access credentials and API keys, without which we could not complete a large portion of the project, since we rely on API's to receive and send information with PLEX. Another difficulty we encountered could have been resolved multiple times, which was miscommunication. Both Root and Branch and Leaf teams had different visions on how this project would work for a long portion of the project, if we had better communication between the two teams since the start of the project and shared developments more often some of this confusion could have been delayed. Examples of confusion were differences in

who was sending what information and how we would receive information. Both teams resolved this issue near the end, where we had multiple in person meetings going over the general and specific information regarding both sides and how to work with each other.

The AGILE method helped our development process a lot, if all that happened was assigning parts of the project to team members at the beginning and having little to no communication between each other, we would have been stuck multiple times. We constantly talked with each other to problem solve aspects of the project and communicated multiple times with each other during every week. This was very helpful when big issues came up, an example of which would be how to assign a Leaf to a workstation. Since the Leaf doesn't know what workstation it is when it requests data we need some way to remember that Leaf-workstation combination. Our group got together as soon as possible after this issue was noticed and created a solution to it through the use of a DHCP server and assigning ip addresses to Leafs.

Some items we would like to work on in the future would be proper API integration, the PLEX api has tens of api calls that deal with each aspect of its service and documentation is minimal so it can be hard to figure out the best API call without testing each one. We would also like to stress test the PI even more to see exactly how many connections it can handle, so we can put a limit or recommend a more efficient piece of hardware.

Overall the CS curriculum has prepared our team very well for this project. The project can be broken down into three parts, the database, the server, PLEX communication. Two of these aspects had classes directly talking about them, with database and data communication. The only aspect that isn't talked about a lot in CS curriculum is third party API usage, but that is pretty self explanatory most of the time.

Software Engineering Code of Ethics and Professional Practice:

1.01. Accept full responsibility for their own work.

We tried very hard to make sure to always do the best we can do and if something that was achievable wasn't achieved to remember that we could have worked harder.

2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.

- We mentioned our limitations such as machine learning to the sponsor to not give false expectations on what we are capable of.

2.03. Use the property of a client or employer only in ways properly authorized, and with the client's or employer's knowledge and consent.

- We talked with the sponsor about proper usage and used the PLEX database and ASPC accounts given to us properly and without abusing them.

2.05. Keep private any confidential information gained in their professional work, where such confidentiality is consistent with the public interest and consistent with the law.

- Some access we were given had private information in it and we did not disclose this information to any other party.

2.09. Promote no interest adverse to their employer or client, unless a higher ethical concern is being compromised; in that case, inform the employer or another appropriate authority of the ethical concern.

- All suggestions we gave to the sponsor we believed were in their best interest towards making a final product.
- Some suggestions we gave we to purchase better hardware to act as the root node, how to handle the mesh networks mode of communication, to let the leaf nodes have information stored inside of them.

3.07. Strive to fully understand the specifications for software on which they work.

- Had multiple meetings with the sponsor and the other ASPC team to make sure we were on the right track or going in the right direction that was wanted of us.
- Meetings and discussions on the overall project increased near the end, especially when we realized there were some discrepancies between the two teams.

6.05. Not promote their own interest at the expense of the profession, client or employer.

- We didn't promote anything that would solely help us and not the sponsor, suggestions made were towards the betterment of the project itself.
- It would have been easy to just tell the sponsor that we don't have enough time to implement new APIs but we tried to work and finish the project.

6.06. Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest.

- No laws were broken when working on this project, and we tried to always keep in mind not to do anything unethical or unlawful