Research Methodology

Every year, students and researchers that are exploring computational-heavy research such as artificial intelligence may be facing issues to complete their research. Throughout the recent years artificial intelligence research may require processing vast amount of data and depending on the dataset, require a high-end machine to be able to handle certain computations to train the neural network. This could impede the researcher as 1) Their machine may not be powerful enough to handle certain computations; 2) Results may take a long time to be produced as more than one neural network model need to be tested to get accurate results; However, according to (Goscniski et al.,2016) this issue could be solved by using school computers as some academic institutions are equipped with low-to-medium end computers are not used after school hours. These computers could then be clustered together to potentially raise more workload and allow researchers to produce better results due to having more processing power and results are produced in a shorter time span.

From previous research conducted by (Agius & Inguanez, 2019). A Dispy and Beowulf Raspberry Pi 3B Cluster that consisted of seven nodes was created for academic purpose research. In their research they compared which cluster was better in terms of scalability, messaging interface and ease of maintenance. The next phase of their research would be implementing such a prototype of school computers to allow students to access these systems for research. (Doucet & Zhang, 2019) created a low-cost Beowulf Raspberry Pi cluster that contained up twenty-one Raspberry Pi Model 3B nodes. This cluster enabled to students in their academy to learn how distributed computing works while also allowing them to access the system for research purposes. Moreover, (Cox et al., 2014) implemented a sixty-four Raspberry Pi 2 nodes that allowed students in that academy to utilize Hadoop for data processing purposes.

**(create pipeline that explains the upcoming processes)**

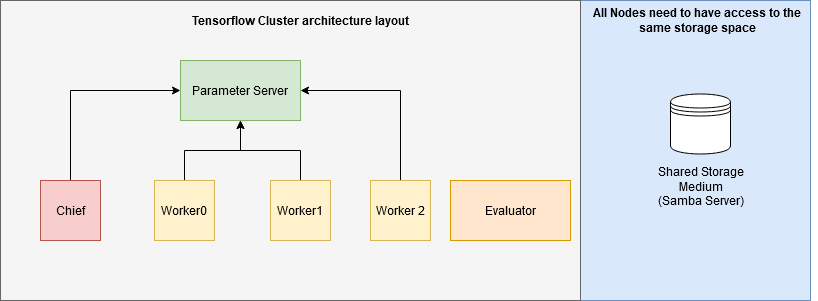
In this research, we propose a solution that by clustering several low-to-medium end computers, a platform as a service with a considerable high processing power can be provided to researchers who might not have the budget nor the necessary computation power that is offered by higher end systems.

Therefore, several research questions needed to be addressed: 1) How can an academic institution offer a platform as a service to its staff and student population for research purposes; 2) How does research in an academic institution benefit from such a solution; 3) How can a good quality of service be guaranteed for such a solution to be run in parallel with the day-to-day running of an academic institution.

1. **Cluster Architecture (DRAFT)**

The first step into creating a distributed system is to explore which framework can be used that allows a model to be trained on multiple machines. TensorFlow, is an open-sourced API that has become one of the most used frameworks for artificial intelligence in recent years. Apart from allowing support for a large variety of neural network layers, activation function, optimizers and analytical tools. It also handles heavy kernels such as convolutions and dense matrix multiplications.

Apart from allowing local implementations, the TensorFlow framework also supports distributed implementations by utilising their distributed training library. Hence, we have decided to use this framework to be able to train concurrently a neural network model in each cluster. The distributed TensorFlow interface allows multiple strategies that can be implemented on a cluster-based system. However, since the aim of this research is to utilize low-medium end computers, we opted to use the Asynchronous Parameter Server Strategy approach as it allows both multi-CPU and multi-GPU computations.



**Figure 1.1: TensorFlow Cluster Architecture**

For this strategy to work in a clustered environment, several steps need to be taken in consideration.

1. Specific cluster setup where each node in cluster had to access the same storage space, as shown in Figure 1.1
2. Ability for all nodes in the cluster to be able to communicate with each other, they do not need to be on the same network however, they must be able to be able to communicate with each other via Secured Shell Protocol.
3. Each Node requires a specific TF\_CONFIG file so that when TensorFlow is initialised it awaits the initialisation of the chief, parameter server and worker nodes before being able to start data distribution and start the model training.

**1.2: Prototype**

In high-performance computing, the concept of having a computer cluster is that the main-master machine in that cluster is required to be able to distribute job to its worker nodes. To be able to produce results, a prototype to demonstrate this concept will be created that will enable the user to concurrently train multiple neural network models in a distributed environment.

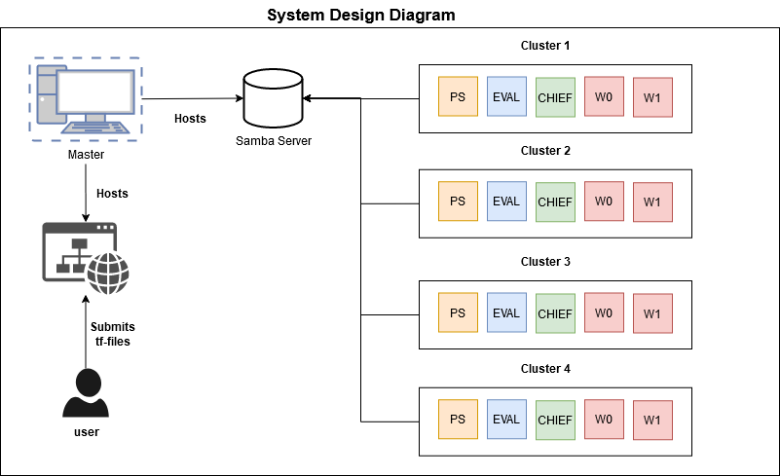
****

Figure 1.2: System Design Diagram

Therefore, we propose a solution as shown in figure 1.3 that follows the concept of high-performance computing. Four clusters will be created where each cluster will have a master node assigned that will distribute tasks to its workers and Main-Master node that will host services and distribute the neural network models to each cluster.

This machine will be required to handle and process the requests done to the users and distribute the neural network models between each cluster. Therefore, a front-end and back-end applications hosted on the main-master node will be used to allow the user to submit a file containing their code to the system and retrieve the neural network models. Also, a samba storage server is needed to be hosted so that each cluster can be able to access this storage space.

To create these clusters, we will be utilizing Google Cloud services as due to the current issue with COVID-19, the prototype could not be implemented using school computers as we did not have access to use them on-site. The Google cloud service platform will allow us to create virtual machines with a custom setup, create them in a specific region so that each machine can be on the same network subnet and be able to create custom firewall rules that can be applied on each virtual machine.

|  |  |
| --- | --- |
| System | Google Cloud Virtual Machine |
| Machine Amount | 21 |
| CPU Type | N1-standard-4 (2 vCPU) |
| RAM | 15 GB |
| Zone deployment | Europe-west3-c |
| Operating System | Ubuntu 18.04 |

Table 1: Machine specification and setup

**Integration and User Access**

Another issue that needed to be addressed is to identify what type of user is accessing our system. Apart from only allowing students and staff into the system, we also need to identify and assign different roles such as ‘researcher’ or ‘admin’ on the system when they login. To solve this issue, we opted to use OAuth 2.0 technologies as it is an industry-standard protocol that is used for authorization. As our academic institution makes use of Microsoft Accounts, we utilise that API to be able to retrieve user information through client details and check if they are students that attend our academy. Afterwards, we make use of a database that checks the roles of the logged in users and allow certain access to the platform depending on their role.

(add diagram)

**2. How does research in an academic institution benefit from such a solution (DRAFT)**

(Add diagram here for distributed TensorFlow and pipeline for training.)

To test how researchers will benefit from a such a system, a neural network model with a dataset will be created and timed by training these models on one machine and then distribute these models between multiple clusters and train them concurrently. As an approach, we have opted to make use of the MNIST Dataset that consists of handwritten digits and contains a training set of 60,000 different images and a testing set of 10,000. For neural network models, we opted to use the Keras Library and create four different neural networks models. Each model will be configured differently by 1) splitting the data training and testing set following the ratio of 60-30, 70-20, 80-10; 2) Adding more hidden layers and activations; and finally, set a Max Step configuration starting from 10000, 20000, 30000 and 40000 steps.

On the other hand, apart from providing a platform as a service to students, we also needed to address the costs that the academic institution will require to spend to implement such a system to students. Therefore, we will be comparing how much it will cost to implement a physical computer cluster and compare how many hours of google cloud services can be redeemed against that cost to see if in the long term would it be better to host cloud computing services instead of building a private cloud.

**3) Text Extraction of timetables (DRAFT)**

**- mention problems caused by computer clusters on students making use of that classroom.**

**- Text extraction from pdf to json.**

**- Database to store information**

**- Algorithm that enables ‘Turns on and off’ cluster training during classroom hours.**

**- show pipeline of process**

For this approach, we can take the school’s timetable and extract important information to see a classroom will be in use and when it will be available. Then create a program that checks if there is currently a pending task in that cluster. If there is a pending task and there is currently a lesson in the class, the process will be paused and will be activated again once the classroom is free again. For tests, we can submit a process and test if the process does indeed pause itself during classroom hours and continues its own when the classroom is labelled as free.