# CSCI 5105 PA2 Phase 1

**Submitted by: Poorna Bengaluru Shivaji Rao and Prajwal Umesha**

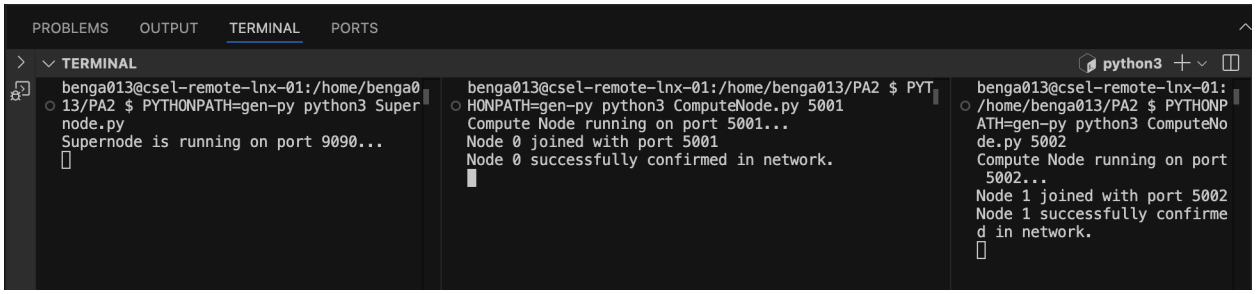
For Phase 1, we have developed a distributed system consisting of a Supernode and Compute Nodes using Thrift-based RPC. The objective was to develop a Supernode which can be run as a server and allow at least one Compute Node to integrate into the network successfully. The Supernode is also responsible for assigning unique IDs to the nodes and checking if their registration has been successful.

Our system consists of two main parts: the Supernode and the Compute Node. The Supernode maintains an active list of nodes and handles join requests using the **request_join** and **confirm_join** methods. It runs on port **9090** and uses a threaded server design to support multiple concurrent connections. The Compute Node, on the other hand, talks to the Supernode in order to join the network through the said port, receives a unique ID, and confirms registration. The Compute Node is also a threaded server in order to support concurrent executions.

We confirmed the system by running the Supernode followed by the Compute Node. The Supernode correctly gave an ID to the node and signed the join. The Compute Node showed its ID and connection details, which confirms that it joined the network successfully. The following screenshot shows the successful execution of the Supernode and Compute Node, which confirms that the requirements of Phase 1 have been met.

Everything has been working fine till now, and the system is behaving as predicted. In the next phase, we will be focusing on handling multiple Compute Nodes simultaneously and improving error handling so that the system is more robust and scalable. In addition to this, we will also be working on compute nodes to maintain a **finger-table** for maintaining the network. To ensure the complete functionality, we will have a client that will select the node and pass the files to the compute nodes for training. Client will then aggregate the results from all the compute nodes and check for the validation error.

Here is a snapshot of the Supernode in action and a Compute Node successfully being added to the network.


PROBLEMS
OUTPUT
TERMINAL
PORTS
TERMINAL
python3
benga013@csel-remote-lnx-01:/home/benga013/PA2 $ PYTHONPATH=gen-py python3 Supernode.py
Supernode is running on port 9090...
benga013@csel-remote-lnx-01:/home/benga013/PA2 $ PYTHONPATH=gen-py python3 ComputeNode.py 5001
Compute Node running on port 5001...
Node 0 joined with port 5001
Node 0 successfully confirmed in network.
benga013@csel-remote-lnx-01:/home/benga013/PA2 $ PYTHONPATH=gen-py python3 ComputeNode.py 5002
Compute Node running on port 5002...
Node 1 joined with port 5002
Node 1 successfully confirmed in network.