Simple Job Dispatcher For Distributed Systems

From : Edoardo Busano-45757100, Prithivi Sharma- 45705704, Ben Ahmmed-45934029

Introduction :

This project is centred around developing a scheduler for jobs for distributed systems. The first stage of this project focuses on allocating tasks on a server by clients. This is firstly done through a client-server connection, allowing the client to schedule jobs to the server simulator of the distributed system. Through this connection, the main task that is done in this stage is to find the largest server that is present within the distributed systems configuration file and sending all jobs from the client to that server. This is done with the use of a function called “allLargest” created on the client-side where the largest server within the configuration file is determined. Subsequently, the client-side simulator uses this information for transferring all jobs of the client to that server. All in all, the main goal of this stage is to be able to create a stable connection between the client side simulator and the server side simulator and be able to schedule jobs to the largest server within the configuration file that the client side simulator is connected to.

System Overview:

The main system used for this project is classified as Ds-sim. It is an open source distributed system simulator which consists of two elements used to connect a client and server through simulations, both known as ds-client and ds-server. The main purpose of ds-client within this project is to schedule jobs for the nodes which are within ds-server. Furthermore, the ds-client’s responsibilities also include finding the largest server within the configuration file which ds-client is connected to and schedule all jobs to that server. The ds-server contains virtual servers whose specifications are saved in an XML file. Ds-server’s main responsibility is to ensure to run the scheduled job from the users on the virtual servers it has which are fabricated and are a part of a simulation. The ds-server is also responsible for sending back any error messages that occur and to acknowledge a completed job.

System Diagram:

