**5.1 Introduction to SIMULATION environment**

**5.1.1 CloudSim**

Cloud user can deploy the large scale application over the real cloud without taking any responsibility for resource management and resource provisioning. CloudSim toolkit provides the modeling and simulation of cloud computing system and application provisioning policy implementation. This Simulation tool provides the repeatable and controlled environment to setup a new virtual cloud computing environment with different cloud component properties. CloudSim Toolkit provides the flexibility to the user to implement his own resource provisioning policy.

CloudSim enables seamless modeling, simulation, and experimenting on Cloud computing infrastructures. It is a self-contained platform that can be used to model datacenters, service brokers, and scheduling and allocation policies of large scale Cloud platforms.

* **Virtual machine:** One of the main aspects that differentiate cloud from other distributed environments is the virtual machine, which creates a virtual environment. All requests are modeled by a simulator in the data centers. These requests are assigned through the VM to the data centers. The VM has a complete lifecycle in the simulator as well as in real cloud environment. The VM lifecycle consists of: provisioning of host to a VM, provisioning of cloudlets to VM, VM creation, VM destruction and VM migration. The VM allocation to the application specific host is the responsibility of the VM provisionary component.
* **Broker:** This acts on the behalf of the consumers to sift through various offerings and decide on what are the best for the consumer. In simple words, it is a third party which creates an understanding environment between the consumer and the CSP.
* **Input file size:** The file size (in Bytes) of the current cloudlet before being submitted to a data centre.
* **Output file size:** This is the file size (in Bytes) of the current cloudlet after execution.
* **Start time:** This is the time at which the VM starts processing the received cloudlet. This time can change for each cloudlet when multiple cloudlets were assigned to a single VM.
* **Finish time:** This is the time that the cloudlet finished being processed by the VM. The finish time depended on the size of the cloudlet.
* **Machine Instructions Per Second (MIPS):** This is the processing power assigned to the VM to execute the instructions according to the specified MIPS

**5.1.2. NetBeans IDE**

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from set of modular software components called modules. Applications based on the NetBeans Platform, including the integrated development environment (IDE), can be extended by third party developers. It is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5. The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications.

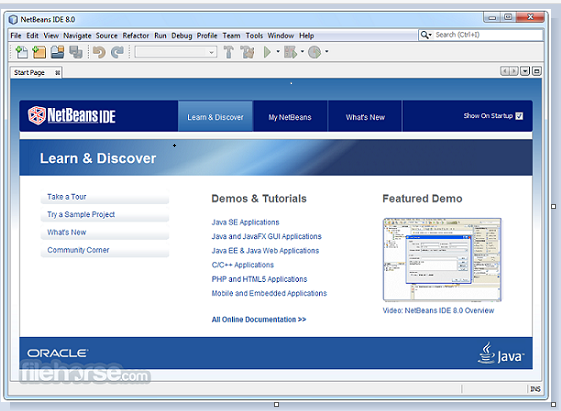
The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plug-ins and NetBeans Platform based applications; no additional SDK is required. NetBeans IDE is a modular developer tool for a wide range of application development technologies. The base IDE includes an advanced multi-language editor, Debugger and Profiler, as well as tools for versioning control and developer collaboration.

The main reusable features and components comprising the NetBeans Platform are outlined below:

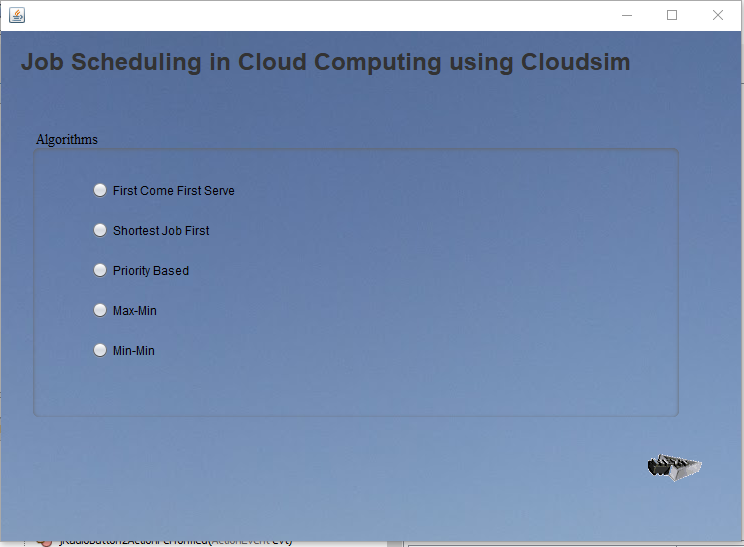
* **Module System**: The modular nature of a NetBeans Platform application gives user the power to meet complex requirements by combining several small, simple, and easily tested modules encapsulating coarsely-grained application features.
* **UI Design:** The award winning Matisse GUI Builder is a standard part of NetBeans IDE. User can use it when prototyping and designing the applications on top of the NetBeans Platform. Its drag-and-drop capabilities and point-and-click features make this an ideal environment for UI design.
* **Build System:** A feature specific to the NetBeans Platform is the fact that its build system is based on Ant. Ant is a standard, non-vendor specific build tool. As a result, users are not locked into NetBeans IDE but can, instead, use the command line to build their NetBeans Platform applications.

**5.2 Results and Discussions**

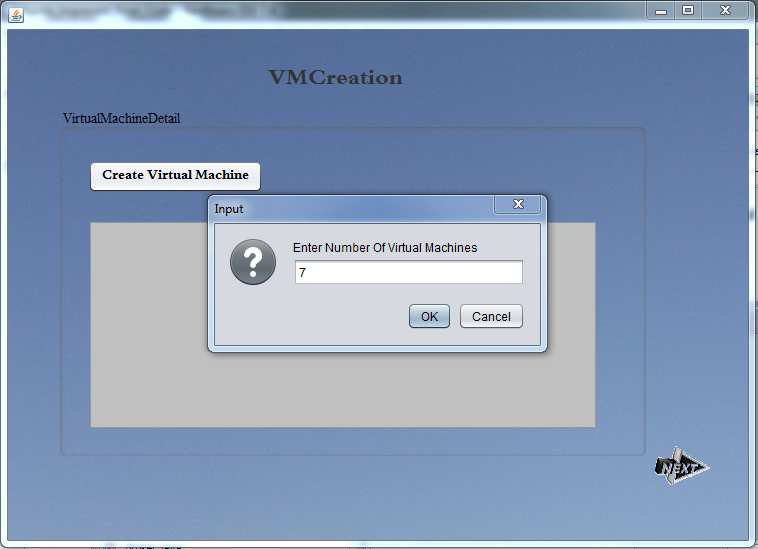
The proposed methodology is implemented with the help of Cloudsim and Net beansIDE8.0.CloudSimis the library that provides the simulation environment of cloud computing and also provide primary classes describing virtual machines, data centers, users and applications.



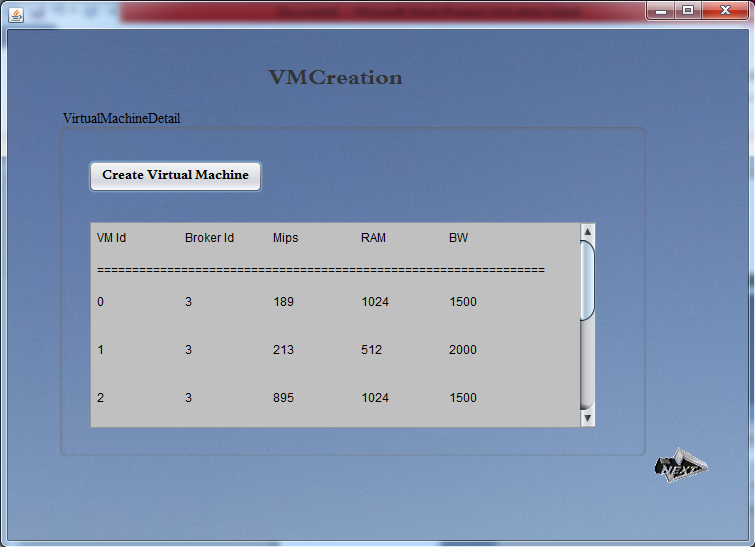
**Figure 1: CloudSim and Netbeans IDE**



**Figure 2: Selection of scheduling algorithm**

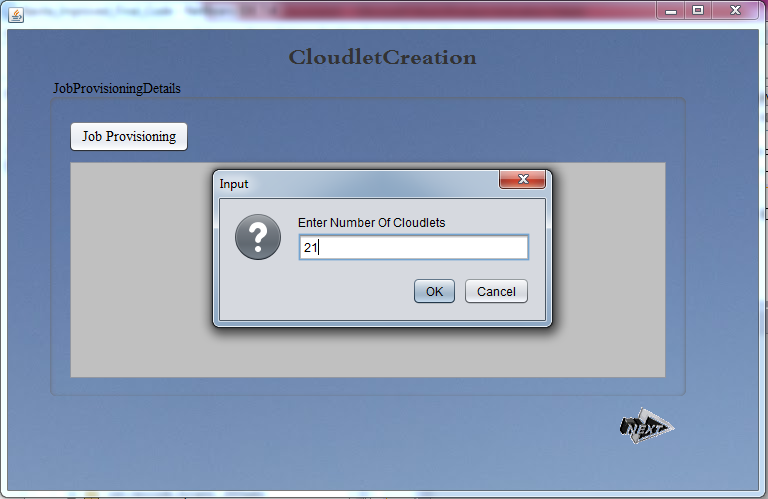


**Figure 3: Shows the virtual cloud environment of Virtual Machine creation**



**Figure 4: Configuration created of Virtual Machines with respective parameters**

In the above figures, Cloudsim is initialized using three parameters that are no. of users, calendar object and one Boolean flag in order to get the status during simulation. After the initialization, the datacenter servers are created and then virtual machines are created in the respective host of the datacenter.

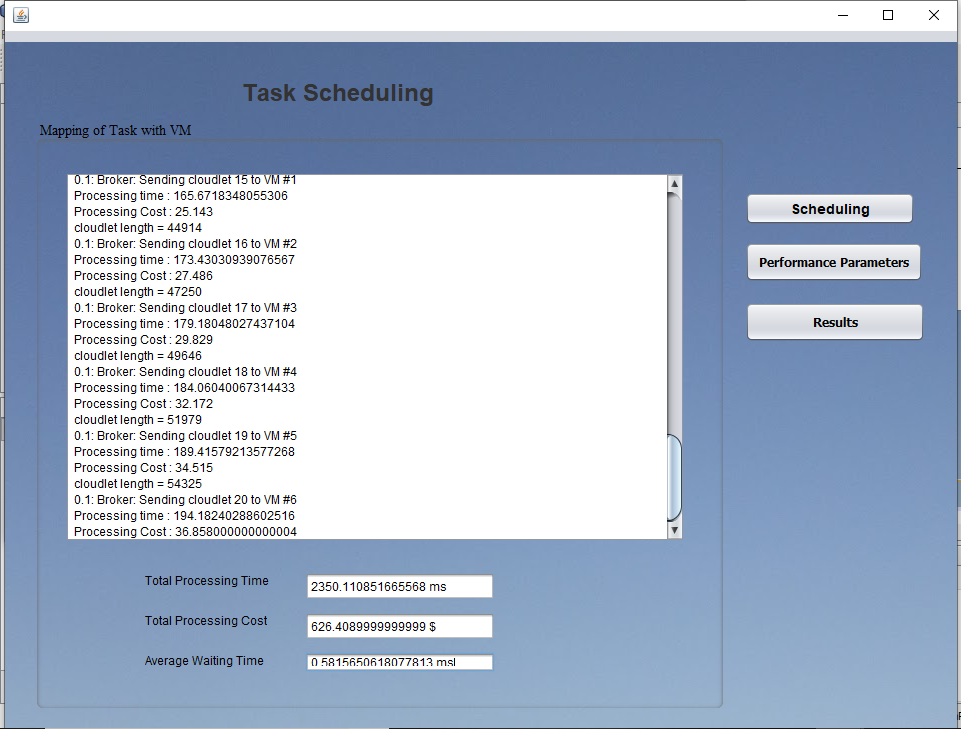


**Figure 5: Shows the virtual cloud environment of Cloudlet creation**

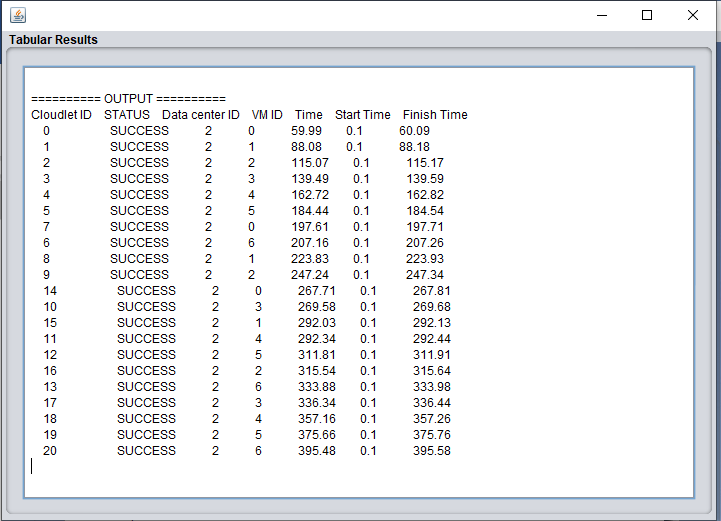


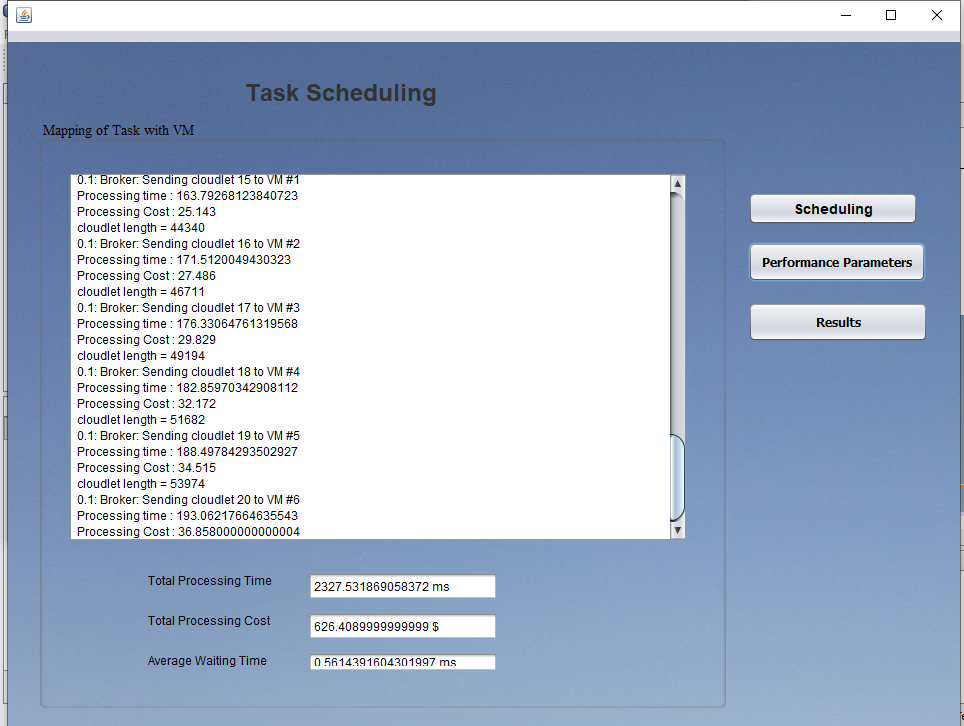
**Figure 6: Shows the configuration creation of Cloudlets with respective parameters**

After the creation of virtual machines in the datacenter servers; the figure 5 and 6 shows the creation of cloudlets in the cloud sim simulation environment with parameters of that are cloudlet length, file size and file output size.

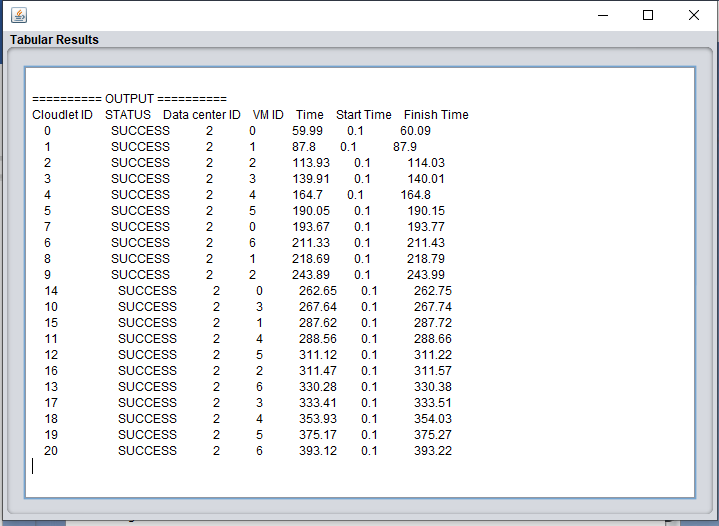
****

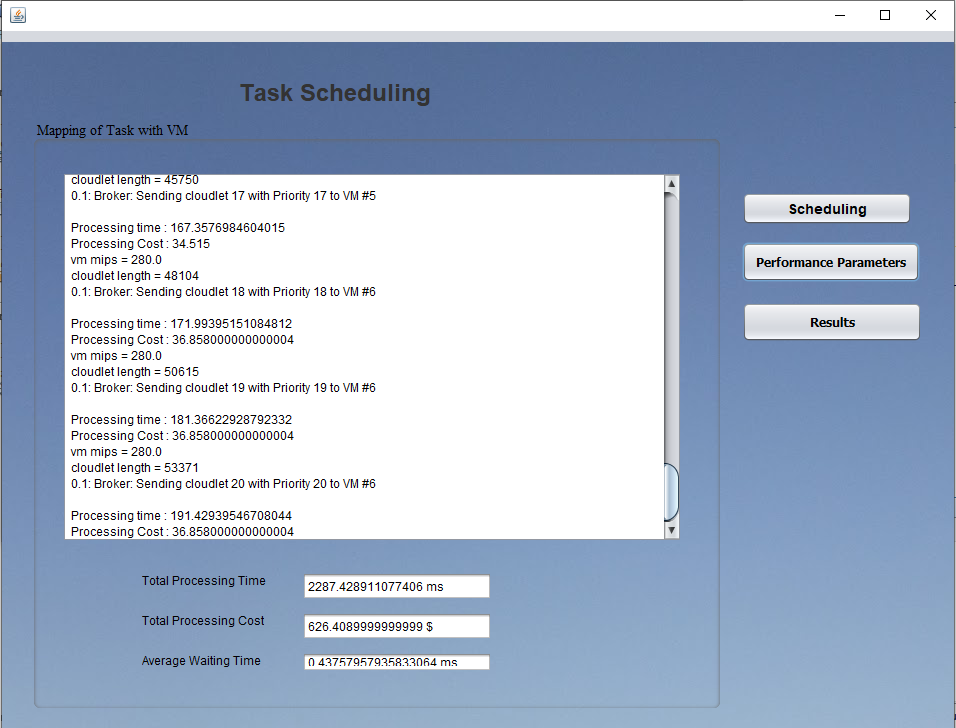
**Figure 7: First Come First Serve Scheduling algorithm results**

**Figure 8: Tabular results of FCFS showing 395.58 milliseconds as a makespan time of algorithm**

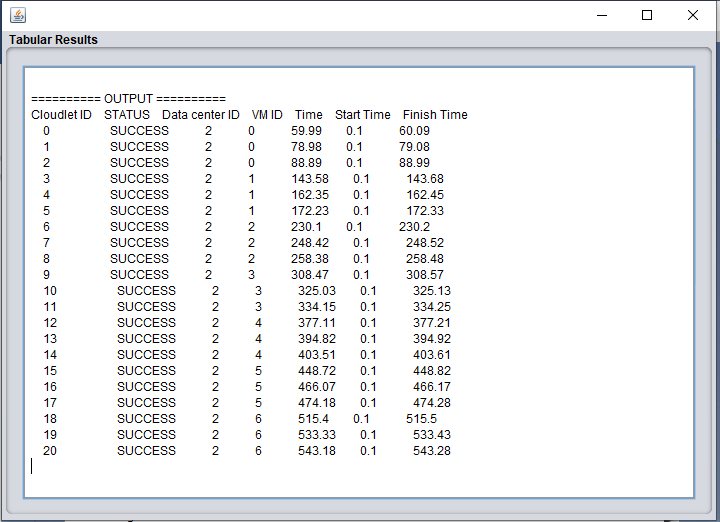


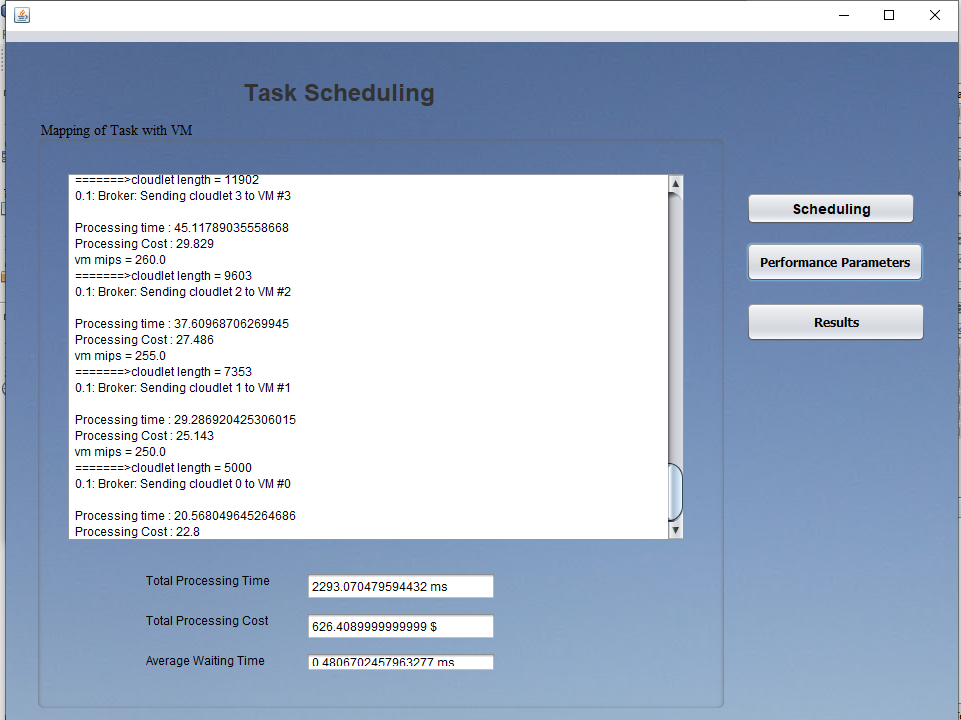
**Figure 9: Shortest Job First Scheduling algorithm results**

**Figure 10: Tabular results of SJF showing 393.22 milliseconds as a makespan time of algorithm**

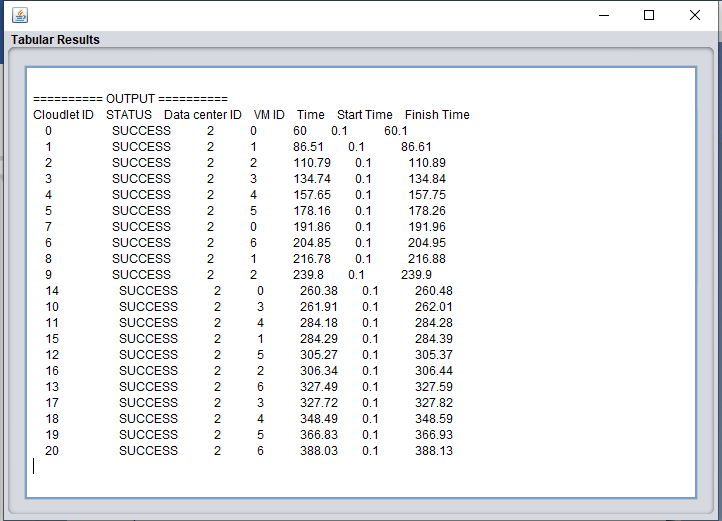


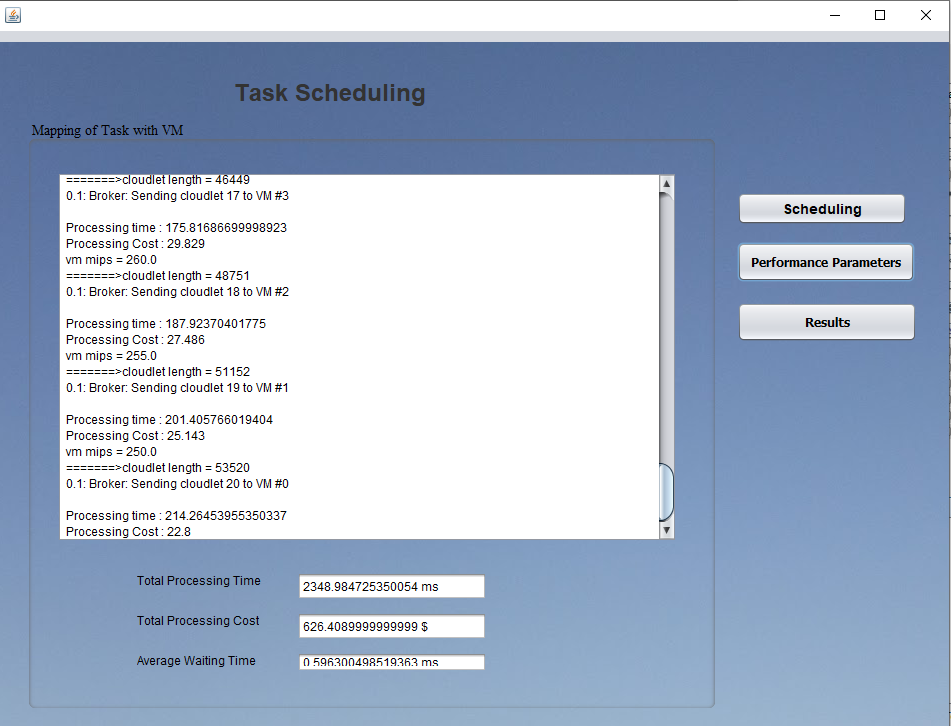
**Figure 11: Priority based Scheduling algorithm results**

**Figure 12: Tabular results of Priority based scheduling showing 543.28 milliseconds as a makespan time of algorithm**

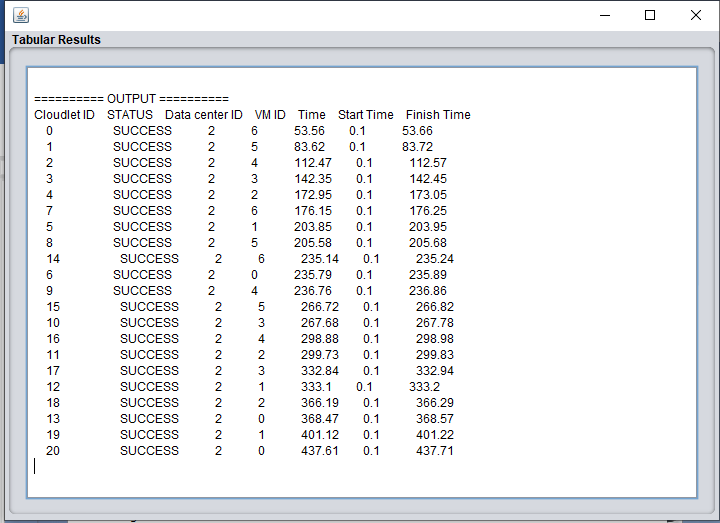


**Figure 13: Max Min Scheduling algorithm results**

**Figure 14: Tabular results of Max-Min showing 388.13 milliseconds as a makespan time of algorithm**



**Figure 15: Min-Min Scheduling algorithm results**

**Figure 16: Tabular results of Min-Min showing 437.71 milliseconds as a makespan time of algorithm**