CS 6847: Cloud Computing
**Assignment 4**: Setting Up Vishwa

--Mahesh Gupta (CS12M018)

## 1. Vishwa Configuration:

Vishwa Grid Configuration, used for program execution was

- **Client Node:** 10.6.9.239
- **Zonal Server:** 10.6.9.97
- **Grid Nodes:** 10.6.9.98, 10.6.9.156, 10.6.9.30, 10.6.9.93

## 2. Program Execution:

Vishwa middleware sample code for **prime number generation** was downloaded from the site, and program was modified for the above mentioned nodes configuration.

The code is supposed to generate all the prime numbers from a given interval. It divides the number interval into 10 equal number interval tasks and executes them. Each task executes the interval independenty.

As the client sample code was written to write on the same client file, results were getting overwritten. We modified the code to create the independent temporary file for result so that all the result values can be appended to the client file.

For the execution, we store the zonal server log and result and client execution log in differnet log files. Each file is stored and is attached in the submission.

The files are:

- zonal.log : log file of zonal server
- primelog.txt : log file generated on client side upon client code execution
- primeresult.txt: result upon code execution.

## 3. Vishwa vs Apache Hadoop:

Vishwa and Apache Hadoop both can be used for distributed task execution. But there are fair amount of differences between both:

| Vishwa | Apache Hadoop |
|---|---|
| 1. It is designed as **Peer to peer**, so there is no single point of failure. | 1. It is designed as **master-slave architecture** with Application Manager (master) and Node Manager(slave). Although namenode is executed in active passive node configuration, but active name node failure affects the performance. |
| 2. It takes care of **node capability** while dividing the task, highly loaded nodes get less amount of task to process than other nodes. Based on capbility, task can be migrated to other node. | 2. All the nodes are considered **homogenous** in terms of capability and load and tasks are equally divided among them. Task cannot be migrated from one node unless node itself fails. |
| 3. Data are considered independent and data with similar feature cannot be aggregated. i.e. Data processed on different nodes are totally different unless on client site some processing is | 3. **Aggregation is supported** in Hadoop. All the values with similar key are processed together. But it handles grouping on one level. Second level of group if to be handled, will be done |

| | |
|---|---|
| performed or another task is created to take care of them. Thus, **agregation operation is not supported** by Vishwa. | manually. |
| 4. Vishwa supports **code and data movement** during execution. | 4. Hadoop supports **data movement** only. |
| 5. Vishwa is designed to handle relatively smalll amount of data. (few GB of data.) | 5. Hadoop is desgined specifically for handling very large amount of data (few TB of data). |