**Project of CS 644:**

**Introduction to Big Data Flight Data Analysis**



Under the Guidance of: Professor Chase Q. Wu

**TEAM MEMBERS**

Jigyasa Kohli (jk423)

Samtha Reddy (sr852)

**Oozie WorkFlow Diagram**

**START**

**KILL**

**MapReduce 1 Scheduler**

**Error**

**OK**

**MapReduce 2**

**TaxiTimeInOut**

**Error**

**Error**

**Ok**

**STOP**

**Ok**

**MapReduce 3**

**Cancellation Reason**

**A performance measurement plot that compares the workflow execution time in response to an increasing number of VMs used for processing the entire data set (22 years) and an in-depth discussion on the observed performance comparison results**

From the above graph it can be inferred that when we ran our workflow with MapReduce Jobs to analyze the entire data set (total 22 years from 1987 to 2008) at one time on two VMs first and then gradually increased the system scale to the 8 VMs and measured each corresponding workflow execution time; the time taken to execute the whole workflow will reduce.

The maximum time taken to ran the workflow is 50.687 minutes (which is approximately 1 hour) to run on 2 VMs, but as we increased our VMs to 8 , the time decreased. Hence, performance of the workflow increased gradually by increasing number of VMs.

**Note : Number of time taken to run the workflow α 1/Number of VMs.**

**Performance α Number of VMs. <- Performance is directly proportional to increase in VMs**

**A performance measurement plot that compares the workflow execution time in response to an increasing data size (from 1 year to 22 years) and an in-depth discussion on the observed performance comparison results.**

In this experiment, we ran our workflow to analyze the data in a progressive manner with an increment of 1 year, i.e. the first year (1987), the first 2 years (1987-1988), the first 3 years (1987-1989), …, and the total 22 years (1987-2008), , and measured each corresponding workflow execution time. We wanted to find out performance with respect to varying input data. We are using 8 Virtual machines throughout this experiment.

We see that execution completes in total 50.687 minutes, that is approximate 1 hour of time.

When we increment our data by one year for every run and record the execution time; we observe that the execution time gradually increases as the amount of input data increases.

Hence, we can conclude that Time taken is directly proportional to the amount of data . Performance deceases with the amount of data.