**Project 3 PageRank Performance Analysis on Academic Cloud**

Chen Jiang (jiang28@indiana.edu), Ninad Faterpekar(nfaterpe@umail.iu.edu)

**Introduction**

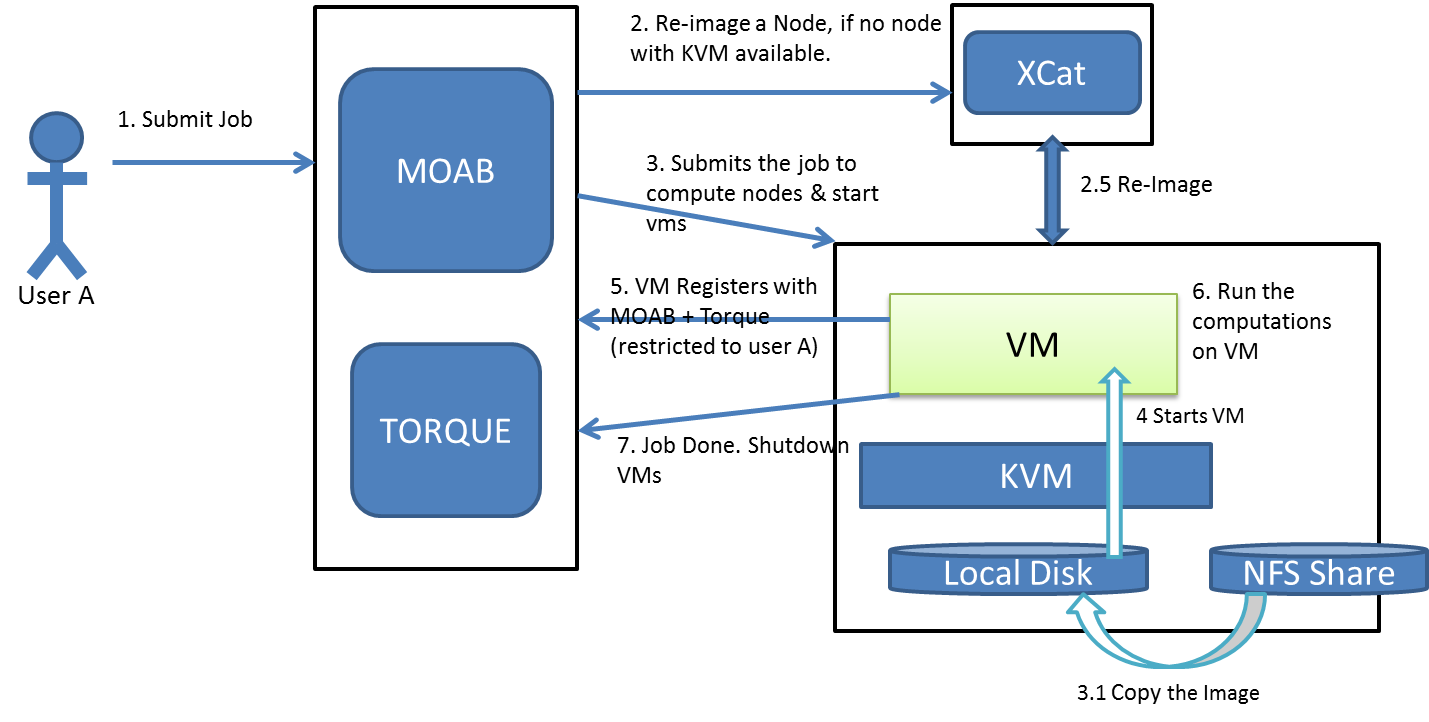
The basic idea of dynamic provisioning is that end users, mainly administrators, can use a client application in the centralized console to deploy or instantiate computing nodes or virtual machines. From this simple client application, an end user is able to easily and dynamically instantiate computing nodes or virtual machines.

The main goal of this project is to use the Cloud environment to run the mpi\_pagerank program in Project1b and to develop simple shell scripts that perform dynamic provisioning as discussed above on FutureGrid. FutureGrid is a nationwide and NSF funded project which provides really good distributed data center that allows us to test pagerank program on cloud. The scripts also automatically run mpi\_pagerank program. Then, remote users can see the result generated by the MPI\_pagerank.

**Main Technologies and Components**

The scripts we developed employ TORQUE Resource Manager to acquire computing resources and set up proper environment. The PBS job scheduler reads the job directives specified in the scripts, and assigned and set up the computing resources accordingly. The scripts execute mpipagerank program, which is detailed in Project 1b.

.



**Fig 1.** User interactions with Dynamic provisioning system (borrowed from project instruction document)

**Overview of Scripts**

Fig 2 shows overview of scripts “run\_pagerank\_bm” and “run\_pagerank\_vm”, performing dynamic provisioning on bare metal nodes and virtual machines, respectively. These two scripts are integrated into one single script

Deamon can be implemented in here in future work

qsub from node i136:

request nodes 1&2

Deamon can be implement in node 2

mpipagerank

mpirun on node 1

mpi\_main processes on node 1

mpi\_main processes on node 2

release node 1&2

run\_pagerank\_bm

**Discussions**

*VM Scripts*

We took a look at the scripts “wait\_for\_vms”, “start\_vms”, and “shutdown\_vms”. The script “start\_vms” basically writes vm IP addresses corresponding to the obtained nodes to an environment variable which is accessible by the requesting user. The script “wait\_for\_vms” checks whether each virtual machine (IP) is accessible through ssh within some timeout, and exists successfully if each vm is accessible. The script shutdown\_vms basically destroys rhels5.5 and removes relevant files.

Experiment

We are interested in finding the difference between running pagerank with only mpi and running mpi\_pagerank on futureGrid. Our focus point is the difference in running time, and our assumption is that it takes much longer to run mpi\_pagerank on the cloud rather than running only on mpi.

We have another obvious investigation is that the running time will grow exponentially as the input file size grows, in order to test it. We have four datasets, the first one is the 1000 database, the second one is the 10000 database, the third one is the 100000 database and the last one is the 1M database. The relation between running time and the file size is as follows.

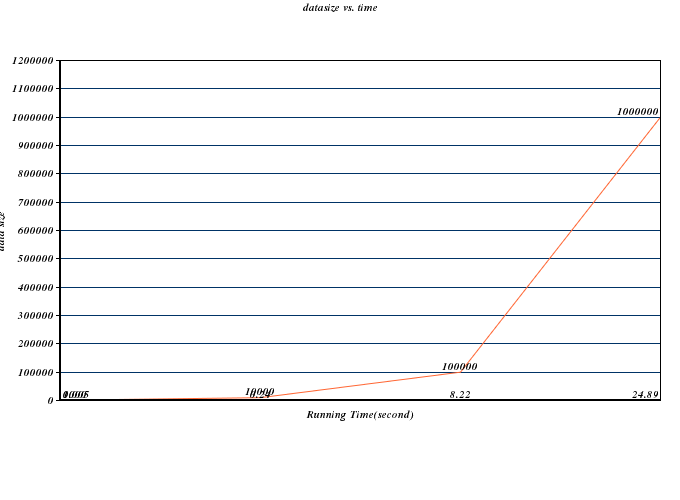
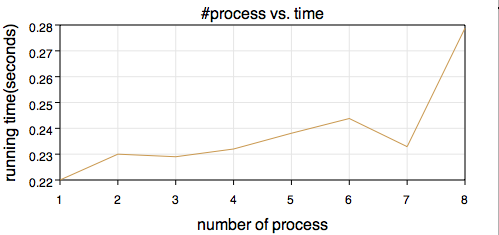


Figure 2. Input\_file size versus. Running time in second(s)

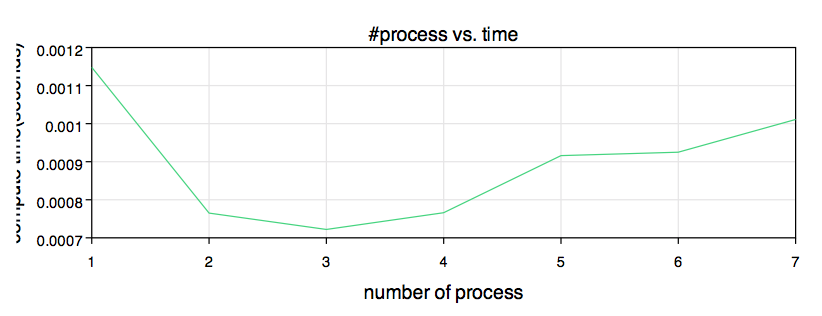
We are also curious about how number of process(node) effect the running time, our initial assumption is that the running time will decrease as the number of process increases because the simple fact that the more nodes we have the less the computation load on each node. In order to justify our assumption, we have designed the experiment with 10000 nodes as input data, 30 iterations and 0.01 as the threshold. We increase the number of process by 1 each run, and we got the following result.



We found that the result is not what we expect. From the figure above, we can see that with process increases. The running time increases (7 process is an exception, but overall, it is increasing). Consequently, the more nodes we assign to this mpi\_pagerank, the slower it is. We conclude that it is maybe because the dataset is not large enough so that the increase in process actually increase the time of communication between each nodes.

Then we measured the computation time in mpi\_pagerank and try to see what would happen to computation time when the number of process changes, and we found out that the computation time will decrease when we increase the number of nodes, however, when we add up to 3 nodes, the computation time starts to increase. The conclusion is, for our test data setting (10k nodes, 30 iterations and 0.01 threashold), the optimization is to use 3 nodes/processes to compute the pagerank. After that, the cost of communication between nodes increases.

Speed up data from 1 million



Speed up data for Million Urls

Number of Iterations / Number of Process on Futuregrid

10k Urls

100kUrls

Feedback on Using Baremetal and Eucalyptus:

This assignment gives us an unique opportunity to work in the academic cloud environment and execute programs.We used PBS script to submit job to job scheduler in bare metal mode. Thus, we learned about the eucalyptus and bare metal in futuregrid.

Acknowledgments

We would like to Thank Prof. Judy Qiu for teaching the concept of Page Rank and AI Stephen Wu and Ila Jogaikar for their help and support.

References:

http://portal.futuregrid.org

Wikipidea