Journal of Theoretical and Applied Information Technology
31st July 2012. Vol. 41 No.2

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

NOVEL JOB DISTRIBUTION ARCHITECTURE TOOL FOR GRID COMPUTING

[1]D.KALEESWARAN, [2]Y.SOBIYA, [3]P.R.JOE DHANITH, [4]R.SIVASENTHIL

[1]Asstt Prof., Department of Computer Science and Engineering, Infant Jesus College of Engg & Tech, Triunelveli.

[2]Asst Prof., Department of Computer Science and Engineering Infant Jesus College of Engg & Tech, Triunelveli.

[3]P.G Student, Department of Computer Science nd Engineering Infant Jesus College of Engineering, Triunelveli.

[4]Asst Prof., Department of Computer Science and Engineering, St.Michael College of Engg & Tech, Kalayarkoil.

E-mail: [1]kaleeswaranme@gmail.com, [2]tsofi12@gmail.com ,
[3]joe.dhanith@gmail.com,[4]shivasenthil.10@gmail.com,

ABSTRACT

Grid computing is an emerging field in networking where the network is expected to share the processing among processors with others networks to lighten the load and increase processing power. The whole scenario winds to the selection fo the best algorithm aimed at distributing the jobs evenly among the several processors at its disposal. The proposed architecture supports execution of both the batch jobs and parallel jobs efficiently and brings out the maximum throughput. The newly developed tool is compared with the SUN grid Engine tool both by running the batch parallel jobs. The observation shows that the developed tool performs slightly better than the SGE tool. As it is designed as a layered with enhanced polices/Algorithms and plugged out, modified with enhanced polices/algorithms and plug in back the module to improve its performance.

Keywords: *Grid Schedulert, Service Gridg, Multistage Communication, Parallel Job Executiong, Grid Tool Kit, Queue*

1. INTRODUCTION

Grid computing is a model for enabling convenient, on demand network access to shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cluster is a parallel or distributed processing system which consists of a collection of interconnected computers cooperatively working together as a single, large, integrated computing resource It can be also said as a parallel processing architecture in which CPU resources are shared across a network, and all machines function as one large supercomputer. (Cluster of Cluster).It allows unused CPU capacity in all participating machines to be allocated to one application that is extremely computation intensive and programmed for parallel-processing.

2. NOVEL JOB DISTRIBUTION ARCHITECTURE

An easy way to execute the different type of jobs based on the hardware as well as software requirements. Parallel operation and control heuristic applications (POOCH), Xgrid, globus tool kit, sun grid engine are compared the architecture performance. These tools are providing result is depending on the hardware as well as software requirements. But analysis of result value is sufficient for grid in working environment. More over probably provide high sensitive output. Each and every job is distributed via knowledge based discovery management. Then grid manager control overall contributions about the job allocation. Making high performance computing to find the well performing workload exchange polices for decentralized computational grid[1]. The multistage communication architecture makes the simple packet and job scheduling scheme to provide tree based routing of each LAN's or gateway networks. This scheme can

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

enhance the self handling ability of network, like handling effect of mall function node or computational node[2].

In this infrastructure concentrate the design and implementation principle of end to end user controller interfaces and communications [3]. The scheduling is based on the removing of uncertain communication. Using fuzzy based scheduling operation based on the fuzzy optimization and both computational and node capacity communication is expressed[4]. Some key setting is making good security management for event driven scheme such as the amount of reductions at various location of the network environment[5]. A capacity based node selection algorithm to manage the selection of nodes within the network as well as the best node selection process from different computational network[6].

A. Grid Manager

The grid manager provides the resources for the underlying network. The resource manager which is a part of the grid manager manages the utilization of all the resources in the cluster. It keep the track of the systems, which are all currently running the jobs , by balancing the load among the cpu's at its disposal.

It indicates the scheduler to schedule the output of the job which is collected back by the grid manager. The management software makes the scheduling process and resource allocation process is based on real time scheduling with unpredictable events. The grid manager must be managing a set as follows:

- Grid Scheduler
- Job Submission
- Queue Manager
- File Storage
- Network
- Parallel Job Manager

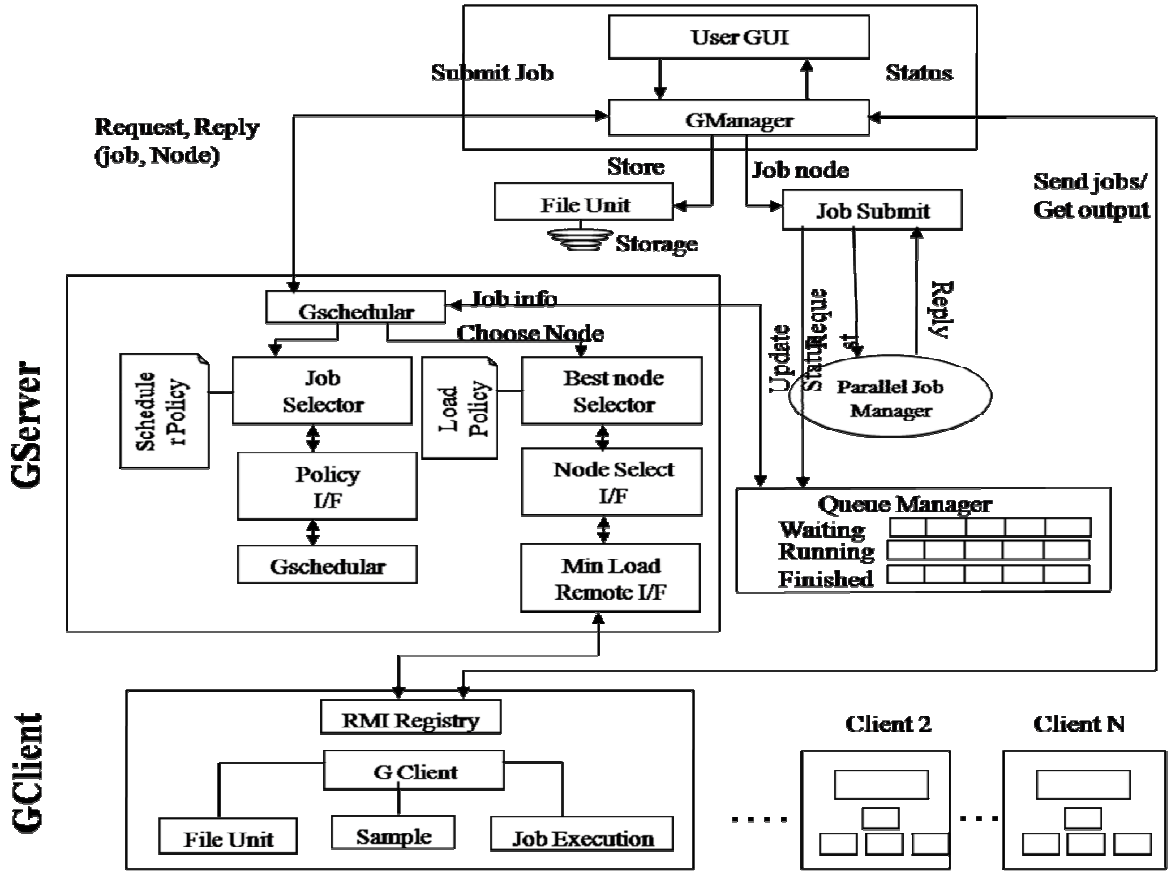

Fig. 1. A novel job distribution architecture tool for grid computing

Journal of Theoretical and Applied Information Technology
31st July 2012. Vol. 41 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

B. Grid Scheduler

A job scheduler is a software application that is in charge of unattended background executions, commonly known as batch processing. Distributed resource management and distributed resource manager provides the batch system with their necessary input. Job scheduling is the assignment of currently running processes to CPU's by the Operating system. There are many concepts that are central to almost every job scheduler implementation and that are widely recognized with minimal variations.

The scheduler evaluates the optimization in dealing with imprecise input value of the scheduler. The weights of the job allocation with security based event driven approach and also the capacity based node allotment. Nodes are randomly chosen from a uniform distribution.

computational grid to support advanced data bank services and reservoirs for data that can be shared among multiple computers and end users on the grid. It is usually containing one are more hard disks, often arranged. These above statements are fully concentrating the performance of parallel job execution in the grid environment.

$$\sum_{j \in V_D} \sum_{s=t}^{[t+I_j TI_k - 1]} X_{j,s,k} \le 1$$

$$for \ \ k \in V_H, t \in T,$$

$$t \le [T_{\max} - I_j TI_k]; \quad (1)$$

$$x_{j,t,k} \in \{0,1\} \quad for \ \ j \in V_D, l \in V_{H,t \in T}$$

To derive the job in the form of task scheduling equ 1 is employed [3[4]]. The given task is once again to split in to weight of the node with random and uniform distribution. Further the required execution time is less than 97% with other scheduling algorithm to evaluate the gain of the optimization with uncertainty in grid computing based on the novel job distribution architecture.

C. Grid Storage

Grid oriented storage is a dedicated data storage architecture which can be connected directly to a logical, redundant storage containers or redundant array of independent disks as do traditional file servers.

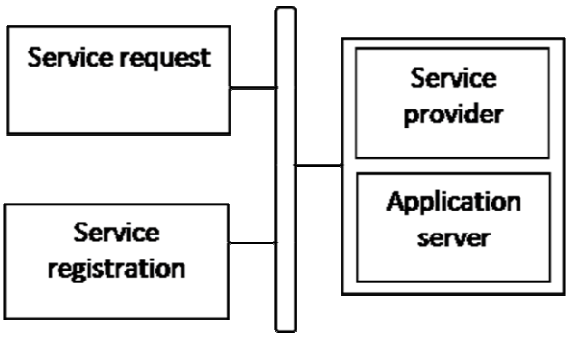

Fig. 2. A file unit storage architecture for grid computing

D. Queue Manager

The queue makes the arrangement of jobs status in the form of three levels. Because of data structure in first in first out concept implementation of higher levels.

- Waiting
  Job is waiting to enter the concurrent execution of the environment.
- Running
  Job is running under the concurrent execution of the grid environment.
- Finished
  Job was finished under the concurrent execution of the grid environment.

IF queue is long and job highly parallel THEN decline job

: : :
: : :
: : :

IF queue is empty and job not parallel THEN accept job

E. User GUI

A graphical user interface is employed to select the job and specify its properties. It provides facility for the user to submit his job and helps to view the status grid environment. The GUI is refreshed automatically whenever a queue is updated in the server. At the same time grid manager is only visible for overall administrative works of grid management environment.

F. Grid Client(Remote Manager)

Remote manager resides at the execution of nodes. The grid manager communicates with remote manager by making remote calls. Remote manager calculates the cpu loads of the local system and send it to server every second. Since server knows the entire execution node's current

Journal of Theoretical and Applied Information Technology
31st July 2012. Vol. 41 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

load, server restricts itself in overloading the execution nodes.

- Node Selection :

The capacity based node selection algorithm is used to select the node in form of 1 or o. Node value = 1 is denotes the node is accepted for handling the job. At the another condition node value = 2 denotes for not suitable for handling the jobs Ref [6].

Maximum flow = capacity matrix(C)-closure matrix(c*)

At the same time node is selected via local job and remote job manager under the certain polices Ref. [1].

$$R_i = IF_{x1} \text{ is } g_i^{(1)} \text{ and…and } x_{N_f} \text{ is } g_i^{(N_f)} \quad (2)$$

$$THEN\ y_i = b_{i0} + ... + b_{iN_f} x_{N_f}$$

The fuzzy in the rule to make the job allocation .Check weather is best or worst node. Selection process is based on the equation 2,3,4,5. Finally get the optimized selected node from the equation 6.

$$y_D(\vec{x}) = \frac{\sum_{i=1}^{Nr} \varnothing_i(\vec{x}) y_i}{\sum_{i=1}^{Nr} \varnothing_i(\vec{x})} \quad (3)$$

Where $y_D(\vec{x})$ is the Set of fuzzy based local output variable.

$$\varnothing_i(\vec{x}) = g_i^{(1)}(x_1) \wedge g_i^{(2)}(x_2) \wedge .... g_i^{(N_f)}(x$$

Where $\varnothing_i(\vec{x})$ is the degree of membership (4)

$$g_i^{(h)}(x) = \exp\left\{ \frac{\left(x_h - \gamma_i^{(h)^2} \right)}{\sigma_i^{(h)^2}} \right\} \quad (5)$$

get the degree of membership from the equation 4.

$$y_i \begin{cases} 1, & \text{job is accepted} \\ -1, & otherwise \end{cases} \quad (6)$$

Select whether a job is selected or rejected

$$\varnothing_i(\vec{x}) = \bigwedge_{h=1}^{N_f} g_i^{(h)}(x_h) = \prod_{h=1}^{N_f} \exp\left\{ \frac{\left(x_h - \gamma_i^{(h)^2} \right)}{\sigma_i^{(h)^2}} \right\} \quad (7)$$

When a positive number again represents the acceptance of the job and a negative value is decline. This optimized with an evolutionary algorithm for number of optimization algorithm.

The local and remote jobs are constructed under the set of policies.

Node Policy

- Contains information about the list of polices.
- Currently chosen policy for selecting the best node.

Pet Policy:

- Contains permissions that will be allowed to the Grid manager.

It is used at the time of launching the remote objects at the execution node.

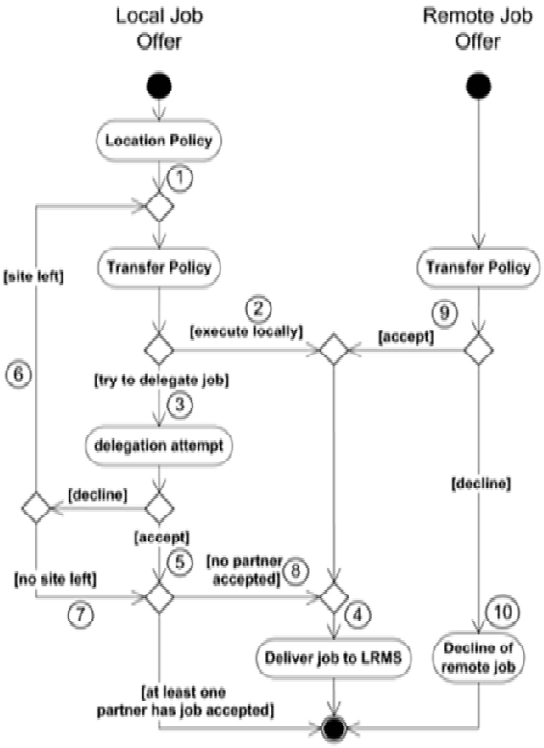

Fig. 3. Job selection algorithm for grid computing Scheduled Policy File

Journal of Theoretical and Applied Information Technology
31st July 2012. Vol. 41 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Contains information about the scheduling policies.
- Currently used scheduling policy
  The entry will be like
  "POLICY_NAME YES (OR) NO"
  Default policy will be the top in the list.

Pet Policy:

- Contains permissions that will be allowed to the Grid manager.
- It is used at the time of launching the remote objects at the execution node.

3. NOVEL JOB DISTRIBUTION ALGORITHM

```
/* when it receives the job */
Upon receiving a job(user_GUI)
If(user_GUI=Parallel job) continue request to
parallel job manager
{
    When receive a reply to G_manager;
/* Node selection  in scheduler*/
    If(select node=best & min loadnode)
    {
        Reply to G_manager ;
        Then submited job to Queue manager;
         move to "Waiting Queue"
     }return to G_manager
/*when job is enter the execution*/
     move to to "Running queue"
     G_manager  send job to the node  selected
/* When job finish */
     Replies the output of given job(G_manager)
& job move to "Finished queue"
     Output to user_GUI;
}
Else
{
 /* Slected job "Batch job */
Continue step 3 to 11
}.
```

4. CONFIGURING FILE

Node File:

It contains the information about the list of address of all execution nodes in the cluster environment.

Input jobs File:

- This directory is used by the grid manager to store the incoming jobs and their input data.
- It can transfer them to the nodes for their execution.

5. EVALUATION

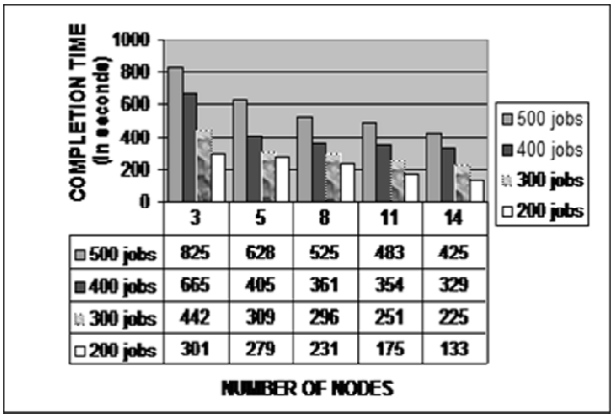

Fig. 4. Performance evaluation for node Vs time

The application used for evaluating the cluster is the 800x800 matrix multiplication application. The program runs for 9700 milliseconds in a standalone computer. The clusters were created using our tool. 3 nodes 5 nodes, 8 nodes, 11 nodes and 14 node clusters were created and tested using the developed tool. The application submitted for 200 numbers of times in each cluster and node the job completion time taken for that number of jobs to get completed.

TABLE I: EXECUTION PERFORMANCE FOR MACHINE VS PROGRAM

Machine	**Execution with different System**			
	Matrix multiplication	**CFD**	*TSP*	Net time
2 RHEL	13:15:20 to 13:15:34	13:15:20 to 13:17:18	13:15:20 to 13:16:43	1 min. 23 sec.
2 APPLE	14:12:51 to 14:14:13	14:12:51 to 14:20:33	14:14:18 to 14:17:16	3 min. 02 sec.
2 RHEL 1 APPLE	13:06:20 to 13:06:32	13:06:20 to 13:12:57	13:06:20 to 13:08:06	2 min. 14 sec.
1 RHEL 2 APPLE	14:35:47 to 14:35:56	14:35:47 to 14:42:24	14:35:47 to 14:39:51	4 min. 04 sec.
4 APPLE	8:53:08 to 8:54:17	8:53:08 to 8:54:17	8:53:12 to 8:55:11	2 min. 01 sec.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. RESULT

The Formation of Cluster with 14 nodes using our tool. Submit the matrix application for 50,100,200,300,400,500 times and noted the total turnaround time. At the same time the cluster 14 nodes using the SGE tool and we submitted the same application for the same number of times and noted the total turnaround
time.

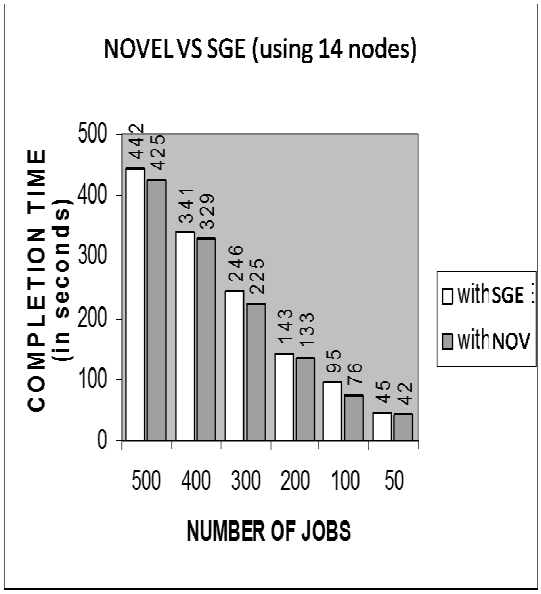

Fig. 5 . Performance evaluation of jobs Vs time

The developed tool achieves better throughput than SGE tool. Because, the developed tool implicitly has its own module to transfer the jobs between the nodes. But SGE and NFS servers are transferring the jobs between nodes. In SGE scripts are submitted first at the execution nodes. After the execution nodes contact the NFS server which is running at the submission node for getting their jobs for execution. In case some delay in transferring the jobs and their input data to the execution nodes.

TABLE III
ANALYSIS FOR NOVEL VS SGE

No of Nodes	Execution time in seconds(NOVEL)	Execution time in seconds(SGE)
1	22	28
2	16	21
3	13	15
4	11	12
5	10	11

7. CONCLUSION

This new architecture supports execution of both the batch jobs and parallel jobs efficiently and bring out the maximum throughput. As it is designed as a layered architecture, a module can be plugged out, modify with enhanced polices/algorithms and plug in back the modified module to improve its performance. This newly developed tool is compared with the sun grid engine tool both by running the batch jobs and parallel jobs. The observation shows that the developed tool performs slightly better than the SGE tool.

8. FEATURE ENHANCEMENT

In feature, add some module like knowledge base discovery. Each and every network selection is depending on the query selection process. Knowledge in the form of node selection with quality management works. Failure of node arrangement is based on unreachable path selection.

REFERENCES

[1] Alexander Folling, Christian Grimme, Joachim Lepping, Alexander Papaspyrou, Roubust Load Delegation in Service Grid Environments, IEEE parallel and Distribution systems, Sept. 2010 Vol. 21.

[2]Hamid Gharavi, Bin Hu, Multistage communication Network for Smart Grid, IEEE , June2011, vol. 99.

[3] Preben Nyeng and Jacob Ostergaard , "Information and communication Systems for controlled by price of Distributed Energy Resources and Flexible Demand," *IEEE Smart Gridt.*, vol. 2, pp. 334–341, Jun. 2011.

[4] Daniel M.Batista and Nelson L.S. da Fonsecs, "Scheduling Grid Task in Face of Uncertain Communication Demands," IEEE Network and Service Management , vol. 2, pp. 92-103, Jun 2011.

[5] Yunfei wang, Iraj Rahimi Pordanjani, and Wilsun Xu, "An Event Driven Demand Response Scheme for Power System Security Enhancementr", IEEE Smart Grid, vol. 2, pp. 23-29, Mar. , 2011.

[6] A.s .Asohan, " Linear Network Fractional Routing", The IJCSE website. [Online]. Available: http://www.ijcse.org/,vol. 2, Issue, 4, 2011

[7] Matteo Gaeta, Pierluigi Ritrovato, Domenico Talia,
"Grid-Enabled Virtual Organizations for Next-Generation Learning Environments ", IEEE Systems, Man, Cybernetics. Vol. 41, no. 4, July 2011.