International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2016, pp. 528-532
e-ISSN:2278-621X

SIMULATING JOBS IN GRID COMPUTING

Nishant D. Savardekar[1], Supritha Rai[2], Chaithra MC[3] and Mrs. Annapoorna Shetty[4]

Abstract - Grid computing helps in distributed computing which involves sharing, data storage and computing resource sharing among in a virtualized collection of organizations. This technology leverages a combination of hardware/software resources which includes all elements of computing, including: hardware, software, applications, networking services.This is one technology useful for some of the most innovative and powerful emerging industrial solution approaches. The implementation is done using GridSim toolkit.

Keywords – Grid Computing, GridSim

I. INTRODUCTION

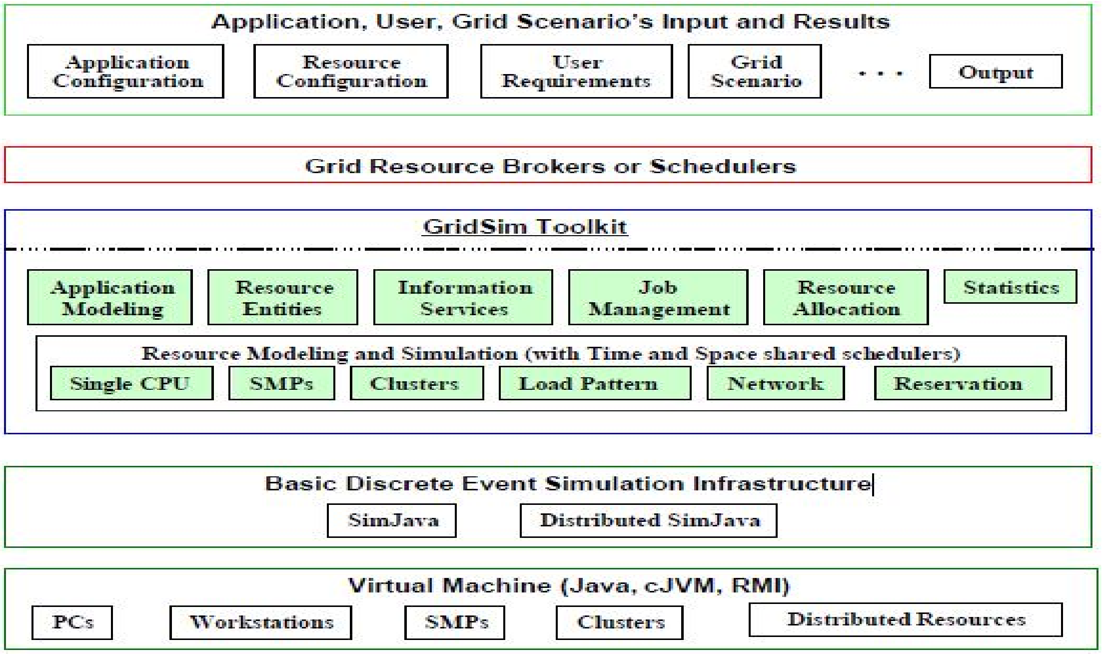
The Grid is emerging as a new technology for solving problems in science, engineering, industry and commerce. Increasing number of applications are using the Grid infrastructure to meet their storage and other needs.However, the effectiveness of a Grid environment is largely dependent on the efficiency and effectiveness of its schedulers, which acts as localized resource brokers.

The multi-layer architecture is showed below. The first layer is concerned with scalable Java's interface, called JVM, whose implementation is available for single and multiprocessor systems including. The second layer is concerned with a basic infrastructure built using the interfaces provided by the first layer. One of the infrastructure implementations available in Java is SimJava.

The third layer is concerned with modelingand simulation of resources, information services which are Grid entities; application model, uniform access interface, and application modeling and framework for creating higher level entities.GridSim toolkit focuses on this layer which simulates system entities offered by the lower-level infrastructure.

The fourth layer is concerned with the simulation of Grid resource brokers or schedulers. The final layer focuses on application and resource modeling with different situations using the services provided by the two lower-level layers.

[1] *Department of MCA, AIMIT, St Aloysius College, Mangalore, Karnataka, India*
[2] *Department of MCA, AIMIT, St Aloysius College, Mangalore, Karnataka, India*
[3] *Department of MCA, AIMIT, St Aloysius College, Mangalore, Karnataka, India*
[4] *Department of MCA, AIMIT, St Aloysius College, Mangalore, Karnataka, India*



II.EXISTING ALGORITHM

A. Scheduling in Time-Shared Resources -

GridSim resource simulator uses internal events to simulate the execution and allocation of Processing Elements (PE) shareto Gridlet jobs. When jobs arrive, time-shared systems start their execution immediately and shareresources among all jobs. Whenever a new job arrives, update the processing time of existing Gridlets and then add this newly arrived job to the execution set. Schedule an internal event to be delivered at the earliest completion time of smallest job in the execution set. It then waits for the arrival of events.

Algorithm: Time-Shared Grid Resource Event Handler ()

1. Wait for an event
2. If external and Gridlet arrives event, then:
BEGIN
a. Allocate Processing Elements (PE) Share for Gridlets Processed so far
b. Add arrived Gridlet to Execution_Set
c. Project completion time of all Gridlets inExecution_Set
d. Schedule an event to be delivered at thesmallest completion time
END
3. If event is internal and its tag value is sameas recently scheduled internal event tag,
BEGIN
a. Allocate PE Share of all Gridlets processed so far
b. Update finished Gridlet's PE and Wall clock time parameters and send it back to the broker
c. Remove finished Gridlet from theExecution_Set and add to Finished_Set
d. Project completion time of all Gridlets inExecution_Set
e. Schedule an event to be delivered at the smallest completion time

END
4. Repeat the above steps until the end of simulation event is received

GridSim allocates appropriate amount of PE share to all Gridlets for the event duration using the algorithm shown below. It should be noted that Gridlets sharing the same PE would get an equal amount of PE share. The completed Gridlet is sent back to its originator (broker or user) and removed from the execution set. GridSim schedules a new internal event to be delivered at the forecasted earliest completion time of the remaining Gridlets.

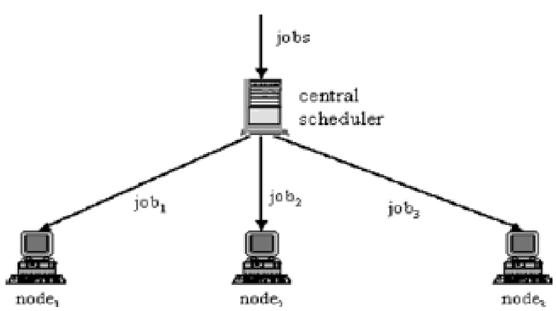
```
Algorithm: PE_Share_Allocation (Duration)
1. Identify total MI per PE for the duration and the number of PE that process one extra Gridlet
        TotalMIperPE = MIPSRatingOfOnePE()*Duration
        MinNoOfGridletsPerPE = NoOfGridletsInExec / NoOfPEs
        NoOfPEsRunningOneExtraGridlet = NoOfGridletsInExec % NoOfPEs
2. Identify maximum and minimum MI share that Gridlet get in the Duration
   If (NoOfGridletsInExec <= NoOfPEs), then:
        MaxSharePerGridlet = MinSharePerGridlet = TotalMIperPE
        MaxShareNoOfGridlets = NoOfGridletsInExec
   Else /* NoOfGridletsInExec > NoOfPEs */
        MaxSharePerGridlet = TotalMIperPE/ MinNoOfGridletsPerPE
        MinSharePerGridlet = TotalMIperPE/(MinNoOfGridletsPerPE+1)
        MaxShareNoOfGridlets = (NoOfPEs - NoOfPEsRunningOneExtraGridlet)*MinNoOfGridletsPerPE
```

III. EXISTING MODELS

There are 2 scheduling models: centralized and distributed.

A. Centralized scheduling

In a centralized scheduling environment, a central machine serves as a resource manager to schedule jobs to all the other nodes that are in the environment. It is used in situations like a computing centre where resources have similar characteristics and usage policies. Figure below shows the architecture of centralized scheduling.



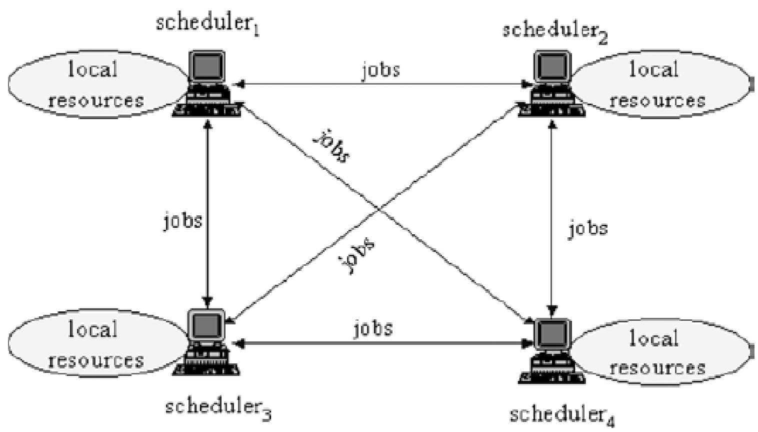
Jobs are first sent to the central scheduler, which then sends the jobs to the appropriate nodes. The jobs that cannot be started on a node are stored in a central job queue for a later start.

One advantage of centralized scheduling system is that the scheduler may produce better scheduling decisions because it has all necessary, and up-to-date, information about the available resources. However, centralized scheduling obviously does not scale well with the increasing size of the environment that it manages. If there is a problem i.e. a failure, it presents a single point of failure in the environment.

B. Distributed scheduling

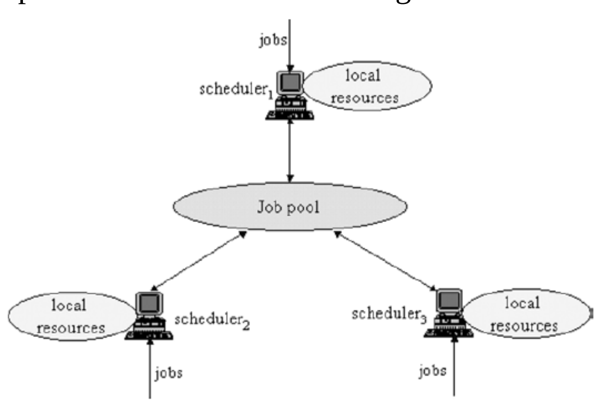
In this, there is no central scheduler responsible for managingall the jobs. Instead, distributed scheduling involves multiplelocalized schedulers, which interact with each other in order to dispatchjobs to the participating nodes. There are two waysfor a scheduler to communicate with other schedulers – direct orindirect communication.Distributed scheduling solves scalability problems, whichare incurred in the centralized paradigm; in addition it can offerbetter fault tolerance and reliability.

1. Direct communication:
In this, each scheduler has remote schedulers that they can interact with. The below figure shows the architecture of direct communication in the distributed scheduling model.



If a job that cannot be sent to its local resources, its scheduler will communicate with other to find resources accurately and make available for executing its job.

2. Communication via a central job pool:
In this scenario, jobs that cannot be executed quickly are sent to a job pool. Compared with direct communication, the local schedulers can choose jobs to schedule resources. Guidelines are required so that the jobs in the pool are executed at some time. The below figure shows the architecture of using a job pool for distributed scheduling.



IV.CONCLUSION

We have used GridSim toolkit for simulation. GridSim resource simulator uses internal events to simulate the execution and allocation of PEs shareto Gridlet jobs and demonstrate the suitability of GridSim for developing simulators for scheduling in parallel and distributed systems.

REFERENCES

[1] Satheesh Kumar N, Ilayaraja AP and PNV Pavan, A Scheduling in Heterogeneous Computing Environment using Alea simulator based on Gridsim

[2] Harshadkumar B. Prajapati and Vipul A. Shah,Scheduling in Grid Computing Environment

[3] Farhad Soleimanian Gharehchopogh, Majid Ahadi, Isa Maleki, Ramin Habibpour, and Amin Kamalinia, Analysis of Scheduling Algorithms in Grid Computing Environment

[4] Pinky Rosemarry, Payal Singhal and Ravinder Singh, A Study of Various Job & Resource Scheduling Algorithms in Grid Computing

[5] GridSim: A Toolkit for Modeling and Simulation of Grid Resource Management and Scheduling