# CHAPTER 3

# GRID ASSOCIATION BASED RESOURCE ALLOCATION ALGORITHM

## 3.1 INTRODUCTION

Distributed Clusters have emerged as mainstream parallel and distributed platforms for high-performance, high-throughput and high-availability computing. The grid computing extends the cluster computing idea to Wide-Area Networks. The grid consists of cluster resources that are usually distributed over multiple administrative domains, managed and owned by different organizations having different resource management policies. With the large scale growth of networks and their connectivity, it is possible to couple these cluster resources as a part of one large grid system. Such large scale resource coupling and application management is a complex undertaking, as it poses a number of challenges in the aspects of security, resource/policy heterogeneity, resource discovery, fault tolerance, dynamic resource availability and underlying network conditions.

**The proposed Grid-Association** is democratic with complete autonomy for each participant. Further, a participant in the association can behave rationally as the use of economic model is proposed for resource management. Grid-Association users submit their jobs to the local scheduler. In case the local resources are not available or are not able to meet the

requirement, then the job is transparently migrated to a remote resource (site) in the association, although this job migration is a constraint to user's Quality of Service (QoS) requirements. In a Virtual Organization (VO), user jobs are managed by a global scheduler which enforces resource allocation based on VO-wide policies as described by Alexander et al (2010).

**Grid Association:** A mechanism for coordinated sharing of distributed clusters based on computational economy is called Grid-Association (GA), which allows the transparent use of resources from the association. When local resources are insufficient to meet its users' requirements, the use of computational economy methodology in coordinating resource allocation not only facilitates the QoS based scheduling but also enhances the value delivered by resources.

## 3.2 RELATED WORK

Resource management and allocation for parallel and distributed systems have been investigated extensively in the recent past as quoted by Mohammed Khanli and Anoloui (2008). In this chapter, the allocation of jobs and resources which allows scheduling across wide area distributed clusters is mainly focused. Grid scheduling systems, National Aeronautics and Space Administration (NASA) scheduler, Condor-Flock, Legion-based Association and Resource Brokers are considered in the design of current grid scheduling methodology as proposed by Leonardo Kunrath et al (2008).

**The Present approach is distinguished in the following aspects:**

- The job-migration or the load-balancing in the Grid-Association is driven by user specified QoS constraints and resource owners' sharing policies;

- The approach gives a resource owner complete autonomy over resource allocation decision

This work presents a allocation system that consists of Internet-wide Globus work pools. The allocating mechanism is based on system-centric parameters. In comparison to this work, GA is based on decentralized Shared Association Directory (SAD). Further, thus scheduling scheme considers user-centric parameters for job scheduling across the association.

**OurGrid** proposed by Andrade et al (2003) provides a grid scheduling middle-ware infrastructure based on the Peer to Peer (P2P) network paradigm. The Ourgrid community is basically a collection of a number of Ourgrid Peer that communicates using P2P protocols. The scheduling in Ourgrid is primarily driven by the site's reputation in the community. In contrast, more generalized resource sharing system based on real-market model is proposed here. Further, the scheduling system used focuses on optimizing resource owners and consumers objective functions.

**Tycoon** is a distributed market-based resource allocation system. The process of scheduling and resource allocation in Tycoon is based on decentralized isolated auction mechanism. Every resource owner in the system runs its own auction for his local resources. Furthermore, auctions are held independently, thus clearly not lacking any association. The mechanism for cooperative and coordinated sharing of distributed clusters based on computational economy is presented in this work and the demand and auction based market model is applied for regulating supply and demand of resources in the GA.

**Nimrod-G** proposed by Buyya et al (2007) is a Resource Management System (RMS) that serves as a resource broker and supports deadline and budget constrained algorithms for scheduling task-farming applications on the platform. The scheduling mechanism inside the Nimrod-G does not take into account other brokering systems currently present in the system. This can lead to over-utilization of some resources and under-utilization of rest of them. A set of distributed brokers having a transparent association mechanism is proposed here.

## 3.3 ASSOCIATION ARCHITECTURE

### 3.3.1 Grid Association Agent

Grid Association (GA) shown in Figure 3.1 defines a mechanism that enables logical coupling of cluster resources. The Grid-Association supports policy based transparent sharing of resources and GA based job scheduling. A new computational economy metaphor for cooperative association of clusters is proposed here. The computational economy enables the regulation of supply and demand of resources, offers incentive to the resource owners for leasing, and promotes GA based resource allocation.

The GA consists of the cluster owners as resource providers and the end-users as resource consumers. The end-users are also likely to be topologically distributed, having different performance goals, objectives, strategies and demand patterns. This Chapter focuses on optimizing the resource provider’s objective and resource consumer’s utility functions by using a quoting mechanism. The Grid-Association consists of cluster resources distributed across multiple organisations and administrative domains. To enable policy based coordinated resource sharing between these clusters, it is defined and modeled as a new RMS system, which is called Grid

Association Agent (GAA). A cluster can become one of the members of the association by instantiating a GAA component. GAA acts as a resource co-coordinator in the integrated space, spanning over all the clusters. These GAAs in the association interoperate using an agreed communication primitive over the shared association directory.

This section provides job allocation about the proposed GA, including models used for budget and deadline calculations in the simulations of the next section. This model defines the following functional modules of a document.

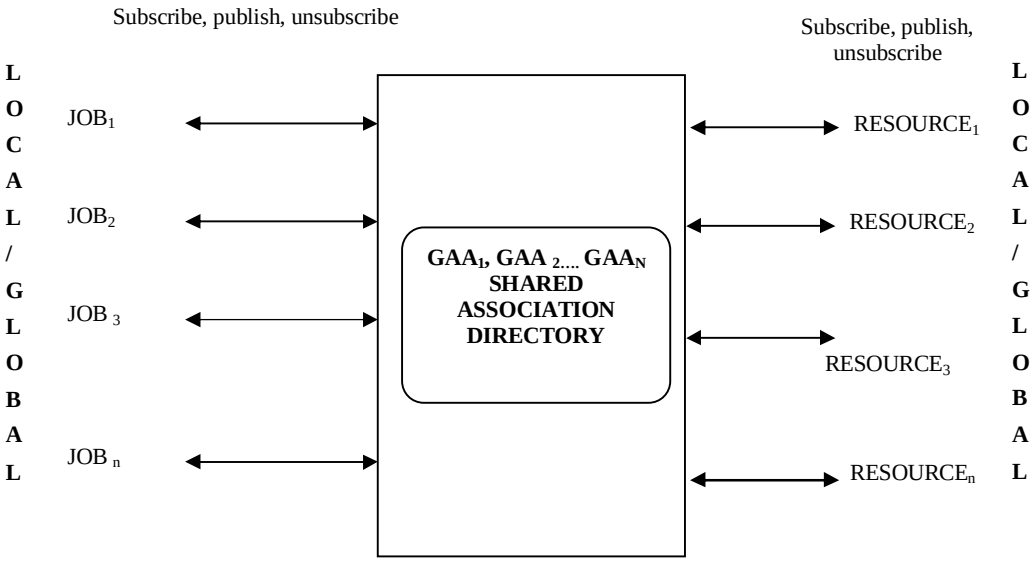


**Figure 3.1 Grid Association**

### 3.3.2 Grid Resource Manager

The Grid Resource Manager (GRM) is responsible for scheduling the locally submitted jobs in the association. Further, it also manages the execution of remote jobs in conjunction with the LRMS on the local resource. A local job refers to the jobs submitted by the local population of users, while

remote jobs refer to the incoming jobs from remote GRMs. A GRM provides admission control facility at each site in the association. Figure 3.2 shows the proposed Grid-Associations Mechanism. The GAA is associated with modules Grid Resource Manager (GRM), Local Resource Management System (LRMS) and Distributed Information Manager (DIM) which is shown in Figure 3.2. The GRM component of GAA is connected to the association queue which accepts the incoming remote jobs (from the association) as well as local jobs. All the remote jobs are transferred to the local queue which is controlled by the GAA's LRMS module. The GRM can also export the locally submitted jobs to other sites in the association depending on the user specified GA requirements.

A local user submits his job to the GRM which in turn places it in the association queue. The GRM analyses the user's GA specification and then sends a query message to the DIM. The DIM returns the $1^{st}$ fastest or $1^{st}$ cheapest machine as specified in the GA requirements. If the returned machine is the local resource, then the job is transferred to the local queue. Otherwise, the job is transferred to a remote site in the association. The GRMs undertake one-to-one negotiation (chapter 4) before submitting the job to the remote site. The GRM to the local job sends admission control negotiate message to the remote GRM requesting a guarantee on the total job completion time. Following this, the contacted GRM queries its LRMS. If the LRMS reports that the job can be completed within the specified deadline, then the admission control acceptance message is sent to the requesting GRM. On receiving the acceptance, the GRM sends the job.

The inter-site GRM-to-GRM negotiation scheme prevents the GRMs from submitting unlimited amount of jobs to the resources. Further, this approach allows autonomy for every resource domain, as they have

capability to perform per job-basis admission control decision. All migrated jobs are queued in the association queue, and then subsequently transferred to the local queue for final execution process.

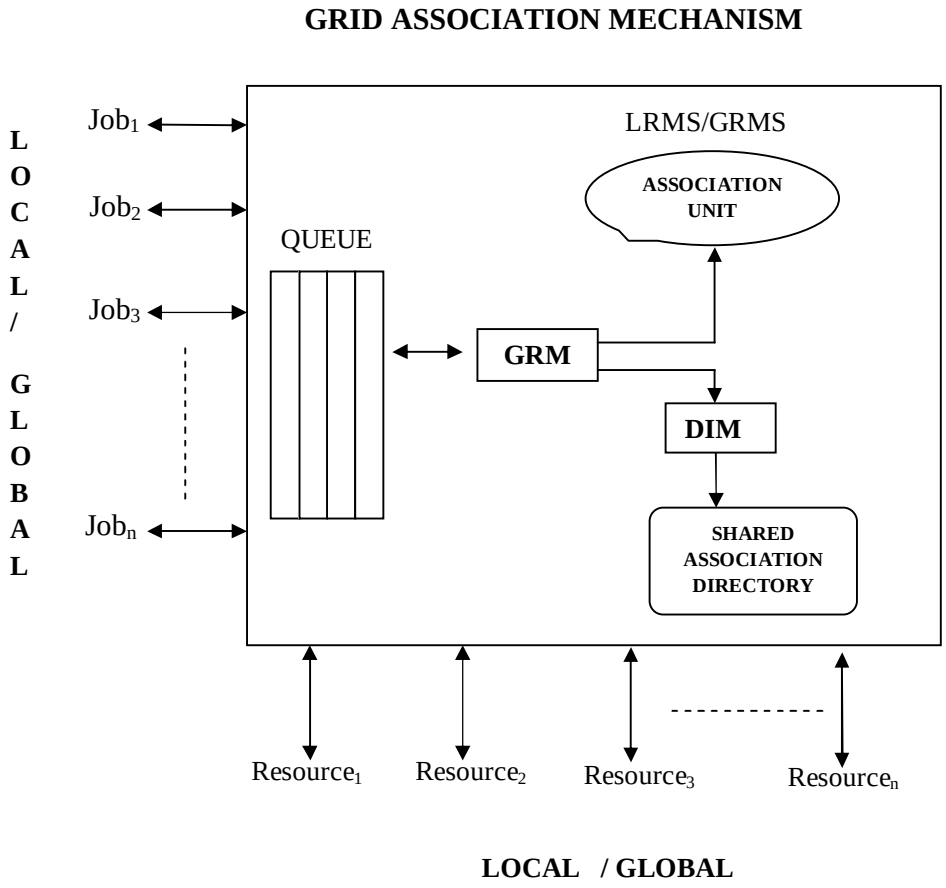


**Figure 3.2 Grid Association Mechanism**

The proposed Grid-Association mechanism can leverage services of Grid-Bank Alexander et al (2010) for credit management. The participants in the system can use grid-bank to exchange grid rupees.

### 3.3.3 Local Resource Management System

In the proposed GA distributed resource sharing system, it is assumed that every cluster has a generalized RMS, such as a Sun Grid Engine (SGE) or Portable Batch System (PBS) that manages cluster wide resource allocation and application scheduling. Most of the available RMS packages have a decentralized organization similar to the master-worker pool model. In the decentralized organization, there is multi scheduling controller (master node) which coordinates system-wide decisions. Grid resource manager queries LRMS to acquire information about local job queue size, expected response time for a job, and resource utilization status.

### 3.3.4 Distributed Information Manager

The Distributed Information Manager performs tasks like resource discovery and advertisement through well defined primitives. It interacts with an underlying SAD as shown in Figure 3.1. It may be recalled that the directory information is shared using some efficient protocol (e.g. P2P protocol). In this case, the P2P system provides a decentralized database with efficient updates and range query capabilities. The individual GAAs access the directory information using the interface shown in Figure 3.1, i.e. subscribe, publish, and unsubscribe. In this chapter, the specification of the interface is of no concern. The resource discovery function includes searching for suitable cluster resources while resource advertisement is concerned with advertising resource capability (with pricing policy – chapter 4) to other clusters in the association. The association directory maintains quoted or advertised costs from each GAA in the association.

Each quote consists of a resource description Ri, for cluster i, and a cost Ci for using that resource configured by respective cluster owners. Using Ri and Ci, a GAA can determine the cost of executing a job on cluster i and the time taken, assuming that the cluster i has no load. The actual load of the cluster needs to be determined dynamically and the load can lead to changes in time taken for job completion.

In this work, it is assumed that ci remains static throughout the simulations. Each GAA can query the association directory to find the $K^{th}$ fastest cluster or the $K^{th}$ cheapest cluster. It is assumed that the query process is optimal, i.e. that it takes O(logn) message to query the directory, when there are "n" number of GAAs in the system. The number of additional messages that are used to satisfy the Grid-Association scheduling process are considered.

## 3.4 DECENTRALISED MARKET PLACE AND GRID ASSOCIATION

Grid computing assembles resources that are well managed, powerful and well connected to the Internet. The grid presents a platform for Grid Participants (GPs) to collaborate and coordinate resource management activities. The Key GPs include the producers (Grid resource-owners) and consumers (Grid users). The GPs have different goals, objectives, strategies, and supply and demand functions. GPs are topologically distributed and belong to different administrative domains. The Controlled administration of Grid resources gives an ability to provide a desired GA in terms of computational and storage efficiency, software or library upgrades. However, such controlled administration of resources give rise to various social and political issues on which these resources are made available to the outside world.

A resource owner invests a significant amount of money in establishing the resource such as, initial cost of buying, setting up, maintenance cost including hiring the administrator and expense of timely software and the hardware upgrades. There is a complex diversity in terms of resources' usage policies, loads and availability. The resource owners in the grid behave as rational participants having distinct stake holdings with potentially conflicting and diverse utility functions. In this case, resource owners apply resource sharing policies that tend to maximize their utility functions as proposed by Grosu and Chronopoulos (2004). Similarly, the resource consumers in a grid associate GA based utility constraints to their applications and expect that the constraints are satisfied within the acceptable limits. Every resource owner makes the policy related decision independently which best optimizes his objective function. Likewise, resource consumers have diverse GA based utility constraints, priorities and demand patterns which will be detailed in chapter 4.

To capture the above dynamics and complexity of Grid resource sharing environment, Grid-Association applies market based economy principles for resource allocation and application scheduling. In particular, two pricing mechanism is adopted. In this model, every resource owner sets up a fixed price based on the demand for his resources in the decentralised market place. The Resource owner advertises his resource access cost through his local GAA service. Analyzing different pricing algorithm based on supply and demand function has become a vast research area. Investigating how the cluster owners determine the price as quoted by Cheng and Wellman (1998) for their commodity is a subject of future work.

## 3.5 IMPLEMENTATION

In this section, general Grid-Association scheduling technique is described. In Figure 3.3 a user who is local to GAA submits his job to the GAA.

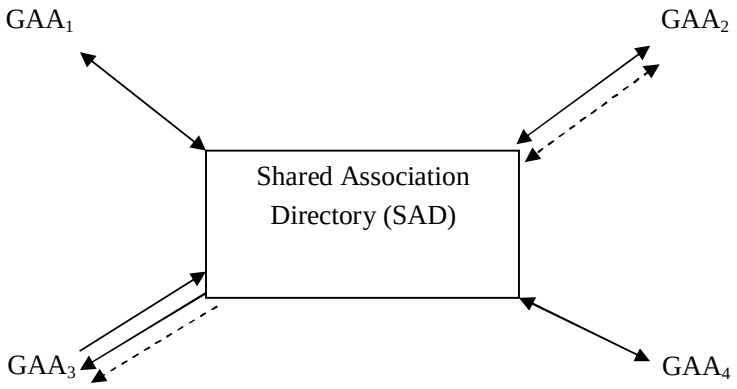


**Figure 3.3 Resource Sharing Mechanism**

If the user's job cannot be satisfied locally then GAA queries the SAD to obtain the quote of the fastest or the cheapest cluster. The SAD which processes the quoting information of all GAAs connected in the cluster sends the quote advertised by GAA2, which is the cheapest or fastest GAA to process the job submitted by GAA. However if the GAA2 has too much load to complete the job within the deadline, then the 2nd cheapest or fastest GAA3 is picked up as an alternative to GAA2 and it is negotiated for the completion of the job. If GAA3 is able to process the job then the job is completed by it otherwise the GAA3 queries SAD for the next level GAA to complete the job. Thus the query negotiate process is repeated until a suitable GAA has found for completing the job submitted by GAA.

Every Association User is supported to quote his budget (willing payment) and deadline (required response time) to express how much he is

willing to pay, for his job number j. In this work, a job is considered to be satisfied if it is only completed within the budget and deadline quoted by the Associate User. Every cluster in the association has its own resource set $R_i$ which contains the definition of all resources owned by the cluster and is ready to be offered. The $R_i$ can include information about the CPU architecture, number of processors, capacity of RAM, secondary storage device and type of Operating System. In this work, $R_i = (p_i, e_i , h_i )$ is considered where $p_i$ = Processor, $e_i$ = Speed and $h_i$ = underlying interconnect network bandwidth . It is assumed that there are enough memory space and flexible OS to execute the jobs and the resource utilization Cost Ci is charged per unit time or per unit of Million Instruction (MI).

$J_i$ represents **the i-th job from the j-th user of the k-th resource.** A job consists of the number of processors required, $p_i$ , the job length, $l_i$ (in terms of instructions), the budget, $b_i$ , the deadline or maximum delay, $d_i$ and communication overhead, $g_i$ .

Realizing the nature of parallel execution with message passing overhead involved in the real world applications, a part of total execution time is considered as the communication overhead and the remaining as the computational time. In this work, it is considered that the network communication overhead $g_i$ occurs for a parallel job $J_i$ which is to be randomly distributed over the processes. The communication overhead parameter $g_i$ would scale up or down for all the clusters depending on how $h_i$ is assumed. The total data transfer involved during a parallel job execution is given by

$$H (J_i , R_i) = g_i \, h_i \qquad (3.1)$$

The time for job $J_{i,}$ = ( $p_i$ , $l_i$ , $b_i$ , $d_i$ , $g_i$ )to execute on resource $R_m$ is

$$D(J_i, R_m) = \frac{l_i}{e_m p_i} + \frac{H(J_i, R_i)}{h_m} \quad (3.2)$$

$$D(J_i, R_m) = \frac{l_i}{e_m p_i} + \frac{g_i, h_i}{h_m} \quad (3.3)$$

And the associated cost is

$$B(J_i, R_m) = C_m \frac{l_i}{e_m p_i} \quad (3.4)$$

If $s_i$ is the system time that $J_i$ is the job submitted to the system then the job must be completed by time $s_i + d_i$ .

### 3.5.1 Cost and Time Factors

Considering the Deadline and Budget Constraint (DBC) scheduling algorithm or Cost-Time Optimization scheduling, the association user can specify anyone of the following optimization strategies for job completion.

- Optimization For Time (OFT) – It gives minimum possible response time within the budget limit
- Optimization For Cost (OFC) – It gives minimum possible cost within the deadline.

When each job arrives at local GAA, the following procedure is carried out:

1. Set r = 1.
2. If OFT is required for the job then query the association directory for the $r^{th}$ fastest GAA; else if OFC is required then the query is made for the $r^{th}$ cheapest GAA. Reference is made to the result of the query as the remote GAA.
3. The local GAA sends a message to the remote GAA requesting for a guarantee of the time to complete the job.
4. If the remote GAA confirms the guarantee, then the job is sent, otherwise r = r+1 and the process iterates through step2.

It is assumed that each query takes O (log n) messages and hence in this work the simulation is done to study how many times the iteration has been performed based on the execution of per job basis and per GAA basis. The remote GAA makes the decision immediately upon receiving the request as to whether it can accept the job or not. If the job's QoS parameters cannot be satisfied (after iterating up to the greatest r such that GAA could feasibly complete the job) then the job is dropped.

Effectively, for job $J_{i,}$ that requires OFC then GAA "m" with $R_m$ is chosen such that

$$B(J_i,R_m) = \min_{1<m\leq n}\{B(J_i,R_m)\} \text{ and } D(J_i,R_m)\leq s_i+d_i$$

where $B(J_i,R_m)$ = Associative Cost, $D(J_i,R_m)$ = Associative Time, $s_i$ = time of job submission and $d_i$ = Execution time. As the cost is considered as the prominent factor, the criteria, the Associative Cost should be minimum i.e $\min_{1<m\leq n}\{B(J_i,R_m)\}$ and the criteria the Associative time should be less

than the deadline $D(J_i,R_m) \leq s_i+d_i$ . In this context the cost will be minimum and it completes the job within the deadline $(s_i+d_i)$

Similarly, for OFT then GAA m is chosen such that

$$D(J_i,R_m) = \min_{1<m\leq n}\{D(J_i,R_m)\} \text{ and } B(J_i,R_m) \leq b_i$$

where $B(J_i,R_m)$= Associative Cost, $D(J_i,R_m)$ = Associative Time and $b_i$=user's budget. As the time is considered as the prominent factor, the criteria the Associative Time should be minimum i.e $\min_{1<m\leq n}\{D(J_i,R_m)\}$ and the criteria the Associative Cost will be well within the budget of the user i.e $B(J_i,R_m) \leq b_i$ .

### 3.5.2 Quote Value

It is assumed that $c_i$ remains constant throughout the simulations. In this work, the only interest is studying the effectiveness of the Grid-Association scheduling algorithm based on the static access charge $c_i$. In simulation, $c_i$ is configured using the function:

$$c_i = f(e_i) \tag{3.5}$$

where, $$f(e_i) = \frac{c}{e} e_i \tag{3.6}$$

c is the access price and e is the speed of the fastest resource in the Grid-Association.

### 3.5.3 User Budget and Deadline

While the simulations in the next section is used to trace data for job characteristics, the trace data does not include user specified budgets and deadlines on per basis. In this case, fabrication of these quantities and inclusion of the models here are forced as discussed by Yang et al (2008).

For a user j, each job is allowed from that user to be given a budget (using Equation (3.4))

$$b_i = 2\ B(J_{i\,,}\ R_k). \tag{3.7}$$

In other words, the total budget of a user over simulation is unbounded and it is interested to compute the budget that is required to schedule all of the jobs. Also, Let the deadline for job I (using Equation (3.2)) be

$$d_i = 2\ D(J_{i\,,}\ R_k). \tag{3.8}$$

The values of total budget and deadline are assigned intime for the given job, as compared to the budget spent and response time on the originating resource.

**Table 3.1 Load and Resource Configuration**

| Index | Resource | Processors | No. of Jobs |
|---|---|---|---|
| 1 | R1 | 512 | 29,000 |
| 2 | R2 | 1024 | 63,000 |
| 3 | R3 | 2048 | 73,000 |
| 4 | R4 | 512 | 31,000 |
| 5 | R5 | 1024 | 59,000 |

## 3.6 PERFORMANCE AND EVALUATION

### 3.6.1 Workload and Resource Methodology

The grid simulation is used to evaluate the effectiveness of the proposed system and the GA provided by the proposed scheduling algorithm is used. The workload trace data are obtained from Feitelson et al (2006). The trace contains real time workload of various supercomputers/resources that are deployed at the virtual organizations R1, R2, R3, R4, and R5 as shown in Table 3.1. The workload trace is a record of usage data for parallel jobs that are submitted to various resource facilities.

Every job that arrives, is allocated to one or more processors for a period of time, and then leaves the system. Furthermore, every job in the workload has an associated arrival time, indicating when it is submitted to the scheduler for consideration. As the experimental trace data do not include details about the network communication overhead involved with different jobs, the artificial communication overhead element as 10% of the total parallel job execution time is included as given by Alexander Folling et al (2010).

The simulator is implemented using the GridSim toolkit that allows modeling and simulation of distributed system entities for evaluation of scheduling algorithms by Buyya et al (2002). The gridSim offers a concrete base framework for simulation of different kinds of heterogeneous resources, brokering systems and application types. This toolkit can simulate resource brokering for resources that belong to a single administrative domain like a cluster or multiple administrative domain like a grid. The core of simulation is based on a discrete event simulation package implemented in java. The main classes of GridSim include GridResource, GridSim, Gridlet, AllocPolicy and

Grid Information-Service. These classes communicate using discrete message passing events. To enable parallel workload simulation with GridSim, the existing AllocPolicy and SpaceShared entities is extended.

The simulation environment models the following basic entities in addition to existing entities in GridSim:

- Local user population – models the workload obtained from trace data;
- GAA – generalized RMS system;
- GAA queues (association and local) – placeholder for incoming jobs from local user population and the association;
- GAA shared association directory – simulates an efficient distributed query process such as P2P.

For evaluating the GA driven resource allocation algorithm, a synthetic GA specification is assigned to each resource including the Quote value (price that a cluster owner charges for service), having varying MIPS rating and underlying network communication bandwidth. The simulation experiments are conducted by utilizing workload trace data over the total period of 2 days (in simulation units) at all the resources. Hence, effectively the simulation considers only a fraction of jobs per computing site as compared to the total number of jobs that are submitted. For example, originally 29000 jobs are submitted to R1 over a period of 1 year, while the simulation considers only 417 jobs (that is no. of jobs submitted over 2 days). Considering of the following resource sharing environment, the experiments are carried out.

- Independent resource:-Experiment 1
- Association without considering economy:-Experiment 2

### 3.6.2 Experiment 1 – Independent Resources

In this experiment the resources are modeled as an independent entity (without association). All the workload submitted to a resource is processed and executed locally (if possible). In Experiment 1 (and 2) it is considered that if the user request cannot be served within requested deadline, then it is rejected otherwise it is accepted. In original trace, as jobs are supposed to be scheduled on the local resource so they are queued in until the required number of processors become available.

Effectively, no job is rejected in the original trace. In Table 3.2 and Figure 3.4 the performance of a resource is evaluated in terms of average resource utilization (amount of real work that a resource does over the simulation period excluding the queue processing and idle time), job acceptance rate (total percentage of jobs accepted) and conversely the job rejection rate (total percentage of jobs rejected). The average resource utilization can be noted down through monitoring the utilization of resource in the simulation. The result of this experiment can be found in Table 3.3. The experiment 1 is essentially the control experiment that is used as a benchmark for examining the affects of using associated (with and without economy) sharing of resources.

**Table 3.2 Average Resource Utilization of Independent Resources**

| Index | Resource | Average Resource Utilization (%) | Total Jobs | Accepted (%) | Rejected (%) |
|---|---|---|---|---|---|
| 1 | R1 | 53.492 | 417 | 96.642 | 3.357 |
| 2 | R2 | 50.06438 | 163 | 93.865 | 6.134 |
| 3 | R3 | 47.103 | 215 | 83.72 | 16.27 |
| 4 | R4 | 44.55013 | 817 | 93.757 | 6.24 |
| 5 | R5 | 62.347 | 535 | 100 | 0 |
| 6 | R6 | 48.17991 | 189 | 98.941 | 1.058 |
| 7 | R7 | 82.08857 | 215 | 57.67 | 49.54 |

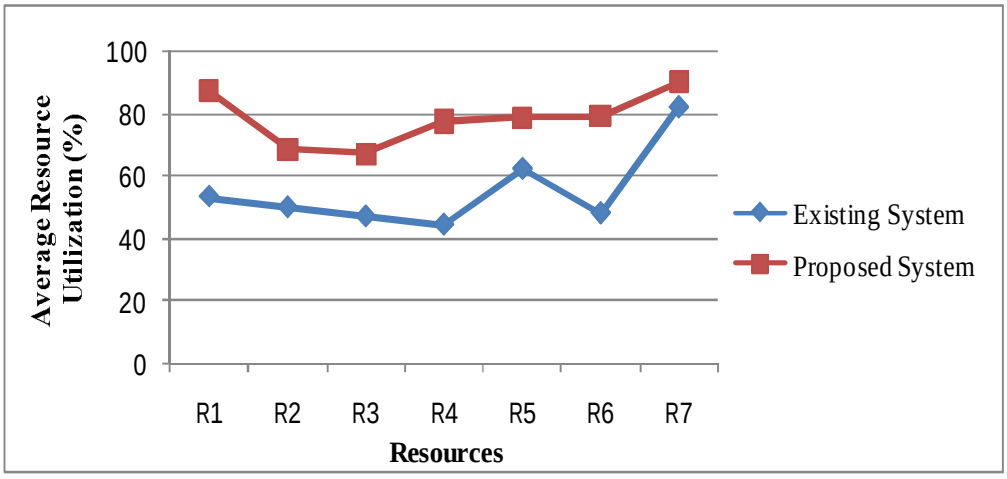


**Figure 3.4 Average Resource Utilization of Independent Resources**

### 3.6.3 Experiment 2 – With Association

In this experiment, the workload processing statics of various resources are analyzed by considering a part of the grid association and without considering the economic model. In this case the workload assigned to a resource can be processed locally. In case a local resource is not available then online scheduling is performed that considers the resources in the association in decreasing order of their computational speed. Quantifying the jobs depending on whether they are processed locally or migrated to the association is performed. In Table 3.3 and Figure 3.5 describe the result of this experiment.

**Table 3.3 Average Resource Utilization of Association**

| Index | Resource | Average Resource Utilization (%) | Total JOBS | Accepted Jobs (%) | Rejected Jobs (%) | Number of Jobs Processed Locally | Number of Jobs Migrated to Remote with Association | Number of Remote Jobs Processed |
|---|---|---|---|---|---|---|---|---|
| 1 | R1 | 87.15 | 417 | 100 | 0.00 | 324 | 93 | 72 |
| 2 | R2 | 68.69 | 163 | 99.38 | 0.61 | 110 | 52 | 35 |
| 3 | R3 | 67.20 | 215 | 90.69 | 9.30 | 145 | 50 | 70 |
| 4 | R4 | 77.62 | 817 | 98.89 | 1.10 | 733 | 75 | 81 |
| 5 | R5 | 78.73 | 535 | 99.81 | 0.18 | 428 | 106 | 129 |
| 6 | R6 | 79.17 | 189 | 100 | 0.00 | 143 | 46 | 129 |
| 7 | R7 | 90.009 | 215 | 98.60 | 1.39 | 105 | 107 | 77 |

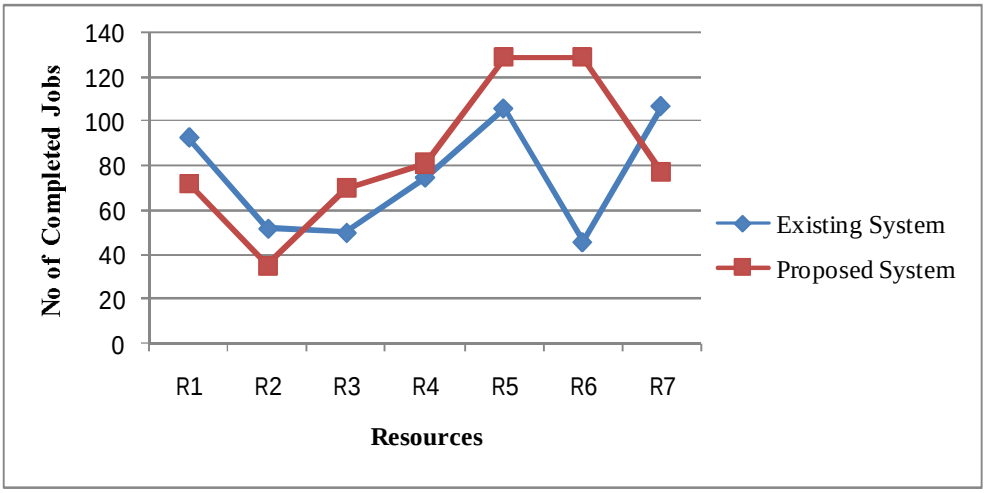


**Figure 3.5 Comparison among Grid Association Considering Job Migration**

During experiment 1, it is observed that 5 out of 8 resources remained underutilized (less than 60%). During experiment 2, it is observed that overall resource utilization of most of the resources get increased as compared to experiment 1 (when they are not part of the association), for instance resource utilization of R1 gets increased from 53.49% to 87.15%. The same trends can be observed for other resources also.

## 3.7 CONCLUSION

A new globally decentralized distributed Cluster Resource Management System namely Grid Association, has been proposed in the view of optimizing the utilization of resources in terms of time or cost or the both by accessing the resource information from SAD effectively. The results show that the average resource utilization of grid associates has tremendously improved over that of the traditional method and the number of jobs completed by the resources of grid association has an edge over that of the traditional one.

The proposed system provides an improvement in average resource utilization and number of completed jobs by establishing the proposed scheme of Grid Association as shown in Figures 3.4 and 3.5.

The methodology of categorizing the jobs as time constraint and cost constraint allows the jobs to be allocated to appropriate resources optimistically and it indirectly balances the supply and demand pattern to satisfy the need of the both resource user and provider in large scale grid systems.

This proposed approach provides a control mechanism for the resource providers to make decisions on resource allocation and offers flexibility to the resource users in selecting the resources according to their requirement whenever it should be cost effective or time effective to balance the supply and demand pattern amicably.