A Case for Cooperative and Incentive-Based Coupling of Distributed Clusters

Rajiv Ranjan, Rajkumar Buyya and Aaron Harwood
Department of Computer Science and Software Engineering
University of Melbourne, Australia
{rranjan,raj,aharwood}@cs.mu.oz.au

Abstract— **Interest in Grid and Peer-to-Peer (P2P) computing has grown significantly over the past five years. They provide mechanisms for sharing and accessing the large and heterogeneous collections of remote resources. Management of distributed cluster resources is a key issue in Grid computing while the primary importance of P2P network is sharing and management of distributed data. Central to management of resources is the ability for resource allocation, as it determines the overall utility of the system. In this paper, we propose a new Grid system that facilitates logical coupling of various distributed cluster resources to enable a cooperative environment. The framework uses a computational economy methodology for clusters and their federation that further promotes QoS (Quality of Service) based resource allocation. We show that federation based resource allocation leads to better utilization of underlying resources. Furthermore, economy models driven QoS based resource allocation strategy can help in managing and evaluating resource consumer and resource provider objective functions more efficiently.**

I. Introduction

Clusters of computers have emerged as mainstream parallel and distributed platforms for high-performance, high-throughput and high-availability computing. Grid [19] computing extends cluster computing idea to wide-area networks. The Grid consists of cluster resources that are usually topologically apart in 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 introduces a number of challenges in the domain of security, resource and policy heterogeneity, resource discovery, fault tolerance, dynamic resource availability and underlying network conditions [20]. Resource sharing on Grid involves collection of resource providers (cluster owners) and resource consumers (end users) unified together towards harnessing power of distributed computational resources. Such sharing mechanisms can be master-worker based or P2P [28] where providers can be consumers as well, extending between any subset of participants. These resources and their users may even be located in different time zones.

Existing approach to resource allocation in the Grid environment is non-coordinated in nature. Application schedulers (e.g. Resource Brokering System [4]) view Grid as a large pool of resource to which they hold an exclusive access. They perform scheduling related activities independent of the other schedulers in the system. They directly submit their applications to the underlying resources without taking into account the current load, priorities, utilization scenarios of other application level schedulers. This enforces over-utilization or bottleneck for some resources while leaving others largely underutilized. As these brokering systems do not have a transparent co-ordination mechanism, so they lead to degraded load sharing and utilization of distributed resources.

The resources on the Grid (e.g. clusters, supercomputers) are managed by local resource management systems (LRMS) such as Condor [27] and PBS [7]). These resources can also be loosely coupled to form campus Grids using multi-clustering systems such as SGE [21], LSF [2] that allow sharing of clusters owned by the same organization. This makes the resource pool available for usage very limited and restricts one's ability to access or share external resources. Moreover, these systems do not support the cooperative federation of the autonomous clusters to facilitate transparent sharing and load balancing.

End-users or their application-level schedulers submit jobs to the LRMS without having the knowledge about response time or service utility. Sometimes these jobs are queued in for hours before being actually processed, leading to degraded QoS. To minimize such long processing delay and enhance the value of computation, a scheduling strategy can use priorities from competing user jobs that indicate varying levels of importance and allocates resources accordingly. To perform these tasks effectively, the schedulers require knowledge of how users value their computations and their QoS requirements, which usually varies with time. The Schedulers also need to provide a feedback signal that prevents the user from submitting unbounded amounts of work.

However, the current system-centric [7][12][18][21][27] approaches to batch scheduling used by the LRMS provide limited support for QoS driven resource sharing. The system-centric schedulers, allocate resources based on parameters that enhance system utilization or throughput. The scheduler either focuses on minimizing the response time (sum of queue time and actual execution time) or maximizing overall resource utilization of the system and thus are not good measures of how satisfied the users are with their resource allocations. The system-centric schedulers make decisions that are good for the system as a whole. The users are thus unable to express their valuation of resources and QoS parameters. Further, they do not provide any mechanism for resource owners to define what

is shared, who is given the access and the scenario under which sharing occurs [20].

To overcome these shortcomings of traditional systems, we propose a new model for distributed resource management, in particular cluster resources. A large scale resource sharing system that consists of cooperative federation [35] of distributed clusters based on policies defined by their owners, which we call as Grid-Federation (shown in Fig.1). This would lead to a greater pool of computational resources being available for various commercial and scientific purposes. Our approach considers computational economy metaphor [4][33][34] for clusters and their federation. In this case resource owners can clearly define what is shared, who is given the access and get incentives for leasing their resources to federation users. The resource allocation in the proposed framework is driven by economic based QoS parameters, which focuses on optimizing user-centric performance of the underlying resources while maintaining overall system performance. The user-centric scheduling mechanism [4][16][17][32] use resource allocation policies driven by market based economic models. They focus on increasing the user's perceived value based on QoS level indicators [31] and QoS constraints. In this case the users can express their valuation of resources and expected QoS. The notion of QoS [23][25][30] to a large Grid system is very important, as it consists of various resource owners and resource consumers having diverse objective functions. Resource owners focus on maximizing profit by leasing their resources while resource consumer's explicit goal is to get the best service utility. In this paper, we demonstrate feasibility and effectiveness of Grid-Federation based resource sharing mechanism. We also analyze the QoS based resource allocation methodology in the proposed framework.

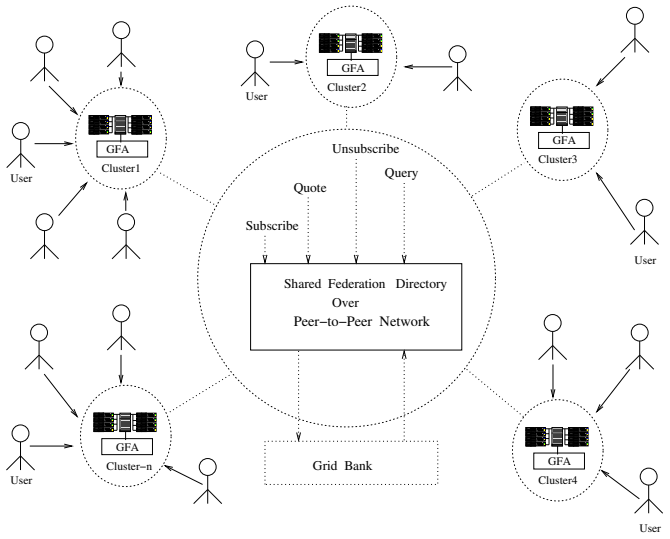

Fig. 1. Grid-Federation over P2P Network

The rest of the paper is organized as follows. Section-II describes some of the related works. In Section III we summarize the underlying proposed Grid system called Grid-Federation. We provide definition of the QoS constraint driven scheduling algorithms in Section III-B. Section IV deals with various experiments that we conducted to prove the utility of our work. We end the paper with some concluding remarks and our future vision in section V.

II. Related Work

Grid resource management and scheduling has been investigated extensively in the recent past (Apples [6], NetSolve [11], Condor [27], LSF [2], SGE [21], Punch [24], Legion [12]). In this paper, we mainly focus on multi-clustering systems that allow coupling of wide area distributed clusters. We also briefly describe computational economy based cluster and Grid systems as we draw inspiration from them.

Load Sharing Facility (LSF) [2], is a very popular commercial batch queuing system which mainly supports campus grids. It focuses towards coupling of various local clusters for example departmental clusters under same administrative domain. It has the ability to run parallel jobs through the use of parallel virtual machine (PVM). Recently it has been extended to support multi-cluster environment by enabling transparent migration of jobs from one cluster to another. Although resource allocation strategy of LSF includes various priorities and deadlines mechanism, still it does not provide any mechanism for end users to express their valuation of resources and QoS constraints. Our *Grid-Federation* addresses this issue through user-centric resource allocation mechanism, which enable users to have better utility and control for their application scheduling.

Sun Grid Engine (SGE) [21] is a cluster resource management system developed by Sun Micro systems. The SGE enterprise edition allows the users to create campus Grid of clusters by combining two or more clusters in the local enterprise network. Each of these clusters is managed by SGE master manager. It has got a policy module which defines proportional based sharing of resources to the users of campus Grid, which in turn determined by the respective share of the user's cluster in the global share space. The users are assigned Tickets, which are like user's pass to use the campus Grid resources. They also get incentive for preserving their tickets during low computation period by getting more access tickets when they need more computational power. This policy is quite flexible depending on resource usage scenario and suited only to campus Grid environment under same administrative domain. It is not very useful for environment that consists of various resource owners with different resource sharing policies and resource consumers with different objective functions and QoS constraints. Our system supports policy based resource sharing where a resource owner can define how, what or when to share a resource and end user's can express their own resource usage scenario.

Condor [27] is a distributed batch system developed to execute long-running jobs on workstations that are otherwise idle. The emphasis of Condor is on high-throughput computing. Condor presents a single system view of pool of multiple distributed resources including cluster of computers, irrespective of their ownership domain. It provides a job queuing mechanism, scheduling policy, priority scheme, job check-pointing and migration, remote system calls, resource monitoring and resource management facilities. Scheduling

and resource management in Condor is done through matchmaking mechanism [29]. Recently Condor has been extended to work with globus, the new version is called Condor-G, which enables creation of global Grids and designed to run jobs across different administrative domains. In contrast, we propose a more general scheduling system that views multiple clusters as cooperative resources that can be shared and utilized based on computational economy model of resources.

Nimrod-G [4] is a RMS system for wide-area parallel and distributed computing platform called the Grid. The Grid enables the sharing and aggregation of geographically distributed heterogeneous resources such as computers (PCs, workstations, clusters etc.) software and scientific instruments across the Grid and presents them as a unified integrated single resource that can be widely used. Nimrod-G serves as a resource broker and supports deadline and budget constrained algorithms for scheduling task-farming applications on the Grid. It allows the users to lease and aggregate resources depending on their availability, capability, performance, cost, and users QoS constraints. The resource allocation mechanism and application scheduling inside Nimrod-G does not take into account other brokering system currently present in the system. This can lead to over-utilization of some resources while underutilization of others. To overcome this, we propose a set of distributed brokers having a transparent co-ordination mechanism, hence enabling cooperative resource sharing and allocation environment.

Libra [32] is a computational economy based cluster-level application scheduler. This system demonstrates that the heuristic economic and QoS driven cluster resource allocation is feasible since it delivers better utility than traditional system-centric one for independent job model. Existing version of Libra lacks the support for scheduling jobs composed of parametric and parallel models, and does not support inter-cluster federation. REXEC [16] is remote execution environment for a campus-wide network of workstations. It provides command line interface for users to specify the maximum credits per minute he is willing to pay for CPU time. The REXEC client selects a node that fits the user requirements. It allocates resources to user jobs proportional to the user demands. It demonstrates computational economy methodology for clusters.

III. Models

This section provides brief details about our proposed Grid system Grid-Federation.

A. Grid-Federation

The realm of Grid computing is an extension of the existing scalable distributed computing idea: Internet-based networks of topologically and administratively distributed computing resources. Different resource type includes computers, computational clusters, on-line scientific instruments, storage space, data and various applications. These resource can be utilized by resource consumers in order to solve compute-intensive applications. For managing such complex computing environment traditional methodologies to resource allocation that attempt to enhance system-utilization by optimizing system-centric functions is less efficient. They rely on centralized policies that usually need complete system wide state information to enable application scheduling. They do not focus on the realization of objective functions of the resource providers and the resource consumers simultaneously. Therefore, we propose an economy-based methodology for co-operative management of distributed cluster resources in the Grid environment. This approach will enhance both policy and accountability in resource sharing, that would further lead to optimized resource allocation.

Existing Grid systems including (Legion [12], Condor [27] etc.) offer unrestricted access to the Grid resources. This can sometimes lead to ”the tragedy of the commons”–A socioeconomic phenomenon whereby the individually ”rational” actions of members of a population have a negative impact on the entire population [14]. These Grid infrastructure lacks both policy and accountability as regards to distributed resource sharing. Currently, there is no standard mechanism that can limit system usage and protect it from free-riders who can abuse the system like in P2P file sharing networks [26]. Other Grid systems such as brokering mechanism access resources independent of other brokers in the system, which can lead to over-utilization of some resources, while under-utilization of others. They do not have any kind of co-ordination [3] mechanism hence are inefficient and non-scalable. The possible solution to this can be set of distributed brokers that co-operate and seamlessly work together having a transparent co-ordination mechanism, which is the notion of our proposed system.

We define our Grid-Federation (shown in Fig.1) as a architectural framework for P2P [22] logical coupling of cluster resources that are under different organizations, administrative and time domains, and that supports policy based [13] transparent sharing of resources and QoS [30][23] based application scheduling. We draw the analogy of Grid-Federation to the electric power Grids [15], which includes a limited number of power suppliers with large investment size (cluster owners). It has large population of electric power consumers purchasing power from these suppliers (federation users) and are connected through various transmission lines (Internet). It provides seamless policy-based (pricing for power/resource consumption) service to is users. This framework aims towards optimizing the user-centric performance of the underlying resources. We also propose a new computational economy metaphor for co-operative federation of clusters. Computational economy [4][33][34] enables the regulation of supply and demand of resources, offers incentive to the resource owners for leasing, and promotes QoS based resource allocation. This new and emerging framework consists of cluster owners as the resource providers and the end-users as the resource consumers. The end-users are also likely to be topologically distributed, having different performance goals, objectives, strategies and demand patterns. We focus on optimizing resource provider’s objective and resource consumer’s utility functions through quoting mechanism.

We model a underlying P2P (shown in Fig.1) networking infrastructure for Grid-Federation. To model shared database

over P2P [10] network we apply the protocol as proposed in the work (which uses Chord protocol to do resource information sharing). The peer-level logical coupling is facilitated by GFA (Grid Federation Agent) component, which acts as cluster's representative to the federation. It quotes for the jobs to other GFAs with its resource description and pricing policy. A quote consists of a QoS guarantee in terms of resources it has to offer, the price it would charge for those resources evaluated by usage over a fixed period of time. We also model Grid Bank [5] that provides services for accounting in the Grid-Federation. More comprehensive details about Grid-Federation framework can be found in [31], which contains details on various entities such as cluster RMS, GFA, job queuing model, economic models and resource allocation algorithm.

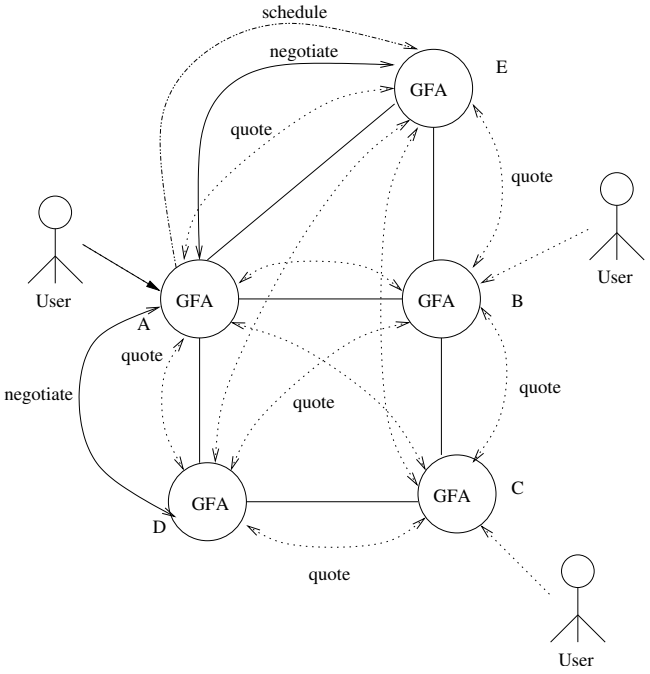

Fig. 2. Quoting Mechanism in Grid Federation

1) Quoting Mechanism between GFAs: This framework aims towards P2P coupling of various clusters thus overcoming the burden of central management and thereby giving autonomous control to individual clusters about their functioning. Each of these clusters are driven by different pricing policies.

In Fig.2, cluster A in the federation does quote broadcast to all other clusters in the federation through P2P shared database. A user who is local to cluster A is making a request while the other clusters are broadcasting their *quote*. A typical quote consists of resource description R_{c_i} and c_i^{price} (price to be paid for using the specified cluster resource), configured by cluster owner. After analyzing all the quotes, cluster A decides whether the request should be serviced locally or forwarded to another cluster. In this way cluster A has the information about all other cluster's service policies.

If the user request can not be served locally then cluster A evaluates all quotes against the user's required QoS. After this cluster A sends negotiate message (Enquire about QoS guarantee in terms of response time) to the matching (In terms of the resource type and the service price) clusters (cluster A sending negotiate message to cluster E and D) one by one untill it finds the cluster on which it can schedule the job (job finally scheduled on cluster E) .

2) Economy Models in Grid-Federation: Existing work in resource management and application scheduling in Grid computing is driven by conventional metaphor where a scheduling component takes decision regarding the site where application will be executed based on some system-centric parameters (Legion [12], Condor [27], Apples [6], NetSolve [11], Punch [24]). They treat all resources with the same scale, as if they worth the same and the results of different applications have the same value, while in reality the resource provider may value his resources differently and has different objective function. Similarly the resource consumer may value various resources differently depending on its QoS based utility functions, may want to negotiate a particular price for using a resource based on demand, availability and its budget. To overcome these shortcomings, we propose an economics-based resource allocation, in this case the scheduling mechanism is driven by resource provider's sharing policy, objective functions and resource consumer's QoS based utility functions. Pricing is primarily based on the demand by the resource consumers and resource availability pattern, in a economic market based resource allocation model.

Some of the commonly used economic model [8] in resource allocation includes the commodity market model, the posted price model, the bargaining model, the tendering/contract-net model, the auction model, the bid-based proportional resource sharing model, the community/coalition model and the monopoly model. We mainly focus on the commodity market model [36]. In this model every resource has a price, which is based on the demand, supply and value in the Grid-Federation. The cost model for the particular cluster depends on the resources it provides to the federation user and is valued accordingly. The initial price of the resources are configured by their owners, it varies between the clusters depending on the hardware configuration, software availability and user's percieveness of QoS.

The relative worth of resources are determined by their comparative supply and demand pattern. If a resource has less demand, then its owner quotes with lower price as compared to previous quote in order to attract more users. Every federation user has to express how much he is willing to pay *(budget)* and expected response time *(deadline)* for his job. User's valuation of resources for his job is directly governed by the job specification and QoS requirements.

Quality is the totality of features of a service that influences its ability to satisfy the given needs. Quality of service evaluations are considered to be driven by a comparison of consumer expectations with their perceptions of the actual quality received. QoS is a guaranteed level of performance delivered to the customer, which is part of service level agreement (SLA) between the service providers and the end-users. The QoS can be characterized by several basic performances criteria including availability, performance, response time and throughput. Service providers may guarantee a particular level of the QoS as defined in the SLA. In our

proposed framework the SLA is part of quoting process, in which the cluster owners are committed towards providing the services they define in their subsequent quotes. The focus of user-centric resource allocation is towards maximizing the end-users satisfaction in terms of QoS constraints. Our Grid-Federation economy model defines the cluster owners, $C_{Gowner} = \{c_1^{owner}, c_2^{owner}, ..., c_n^{owner}\}$ that owns resources $R_G = \{R_{c_1}, R_{c_2}, ..., R_{c_n}\}$. Every cluster in the federation has its own Resource set R_{c_i} which contains the definition of all resource owned by the cluster and ready to be offered. R_{c_i} includes information about the CPU architecture, number of processors, RAM size, Secondary storage size and Operating system type. Every resource in federation has a price, which we represent by $P_{Gcost} = \left\{c_1^{price}, c_2^{price}, ..., c_n^{price}\right\}$. The resource owner c_i^{owner} charges c_i^{price} per unit time or price per unit of Million Instructions (MI) executed e.g. per 1000 MI. There is mapping function from set of federation resources (R_G) to cluster price model (P_{Gcost}).

$$\Pi : R_G \rightarrow P_{Gcost} \tag{1}$$

Let $U_G = \{c_1^{user}, c_2^{user}, ..., c_n^{user}\}$ contains the federation users belonging to various clusters. c_i^{user} represents the users belonging to cluster i. Every cluster owner c_i^{owner} requires jobs J_u to use its resource power. A user owns a job $J_i \in J_u$. Every federation user u_i is modeled as having a resource allocation utility function $Q_{oS}(Constraint)$ for each job which indicates QoS value delivered to the user as a function of specified QoS constraints (deadline and budget). Each job J_i consumes some power of particular type of cluster resource R_{c_i}.

For every job J_i, federation user u_i determines a *budget*, which he is ready to spend in order to get his job done. This is a mere user's assumption which can be feasible or unfeasible. If this assumption is unfeasible then it is quite likely that user's job would get rejected from the federation, in that case the user may have to increase the budget constraint. In addition to budget, user may also give his preference about the response time it expects from the system *(deadline)*. When users submit their jobs to the GFA, they express maximum value of both budget and deadline constraints with one of the two optimization strategy that should be adopted during scheduling.

Every federation user $u_i \in c_i^{user}$ can express the optimization strategy he intends for his job J_i. We propose two optimization strategies that a user can opt for. Starting with the *Time Optimization* [4] strategy, where the focus is on getting the work done as fast as possible. In this case the users specify the maximum budget (c_{budget}) and the deadline ($t_{deadline}$) for their job. In this optimization strategy the user might get his job done within the deadline limit but he may have to invest maximum budget. This signifies as the user invests more budget, it is likely that he will get better response time from the system.

$$Response - Time \propto 1/Budget \tag{2}$$

The federation user can also specify *Cost Optimization* [4] strategy for his job, in this case focus is on getting the work done in minimum possible cost, but within the time constraint. This strategy will get the user's job done in minimum possible cost while maximizing the response time within the deadline limit.

B. QoS Driven Resource Allocation Algorithm for Grid-Federation

(Our algorithm is an extension of basic Nimrod-G [4] algorithm)

We consider a deadline and budget constrained (DBC) Grid-federation scheduling algorithm, or cost-time optimization scheduling. The federation user can specify any one of the following optimization strategies for their job.

1) Optimize for time: give minimum possible response time to the federation user, but within the budget limit.
2) Optimize for cost: produces results by deadline, but reduces cost within a budget limit.

As jobs arrive at a GFA, the following is done:

1) Analyze Quotes: Identify the resource type, characteristics, configuration, capability and the usage cost per unit time or job length by analyzing the quotes advertised by various clusters in the federation. Store these statistics for future job scheduling in federation resource set R_G.
2) Accept, Analyze and Schedule Local Jobs.

Assignment of job J_i to the resources in the federation can be formally described by the function

$$\Delta : J_i \longrightarrow R_{c_i} \tag{3}$$

from the set of jobs J_i to the set of federation resource R_{c_i}.

At any time t given m jobs J_1,J_2,..,J_m and p clusters resources R_{c_1},R_{c_2},...,R_{c_p} that matches jobs resource and QoS requirements, it is possible to assign them in p^m ways. Each job J_i has c_{budget} and $t_{deadline}$ associated with it. The problem is to find an a resource, which minimizes both c_{budget} and $t_{deadline}$ in accordance with the optimization strategy sought by the owner of the job J_i. Further the assignment strategy should lead to efficient utilization of federation resources and minimize the job starvation rate.

Resource allocation for job J_i can be optimized of any of the two user specified QoS constraints. We define R_{cost} as a function which determines the processing cost of resource R_{c_i} (service price) and R_{power} as a function which determines the processing power of resources R_{c_i}.

$$R_{cost} : R_{c_i} \longrightarrow Q \tag{4}$$

$$R_{power} : R_{c_i} \longrightarrow Q \tag{5}$$

If user seeks cost optimization for his job then, allocate resource R_{c_k}, $k < p$, such that,

$$R_{cost}(R_{c_k}) = min(R_{cost}(R_{c_i})) \quad i = 1...p \tag{6}$$

If user seeks time optimization for his job then, allocate resource R_{c_k}, $k < p$, such that,

$$R_{power}(R_{c_k}) = max(R_{power}(R_{c_i})) \quad i = 1...p \tag{7}$$

The following holds true for both optimization strategy. Let the start time of J_i is s_i, (we assume that the s_i's are integer, and that min $\{s_i\}$=0)

Every job J_i has deadline $t_{deadline}$ and budget c_{budget} so,

$$s_i + \tau_i \leq t_{deadline} \tag{8}$$

$$\tau_i = Total\ CPU\ Time\ required\ by\ the\ job \tag{9}$$

and,

$$J_i^{p-cost} = R_{cost}(R_{c_i}) \ . \ \tau_i \leq c_{budget} \tag{10}$$

$$J_i^{p-cost} = c_i^{price} \ . \ \tau_i \leq c_{budget} \tag{11}$$

J_i^{p-cost} denotes processing cost of job J_i on the resource R_{c_i}

IV. Experiment and Analysis

We used trace based simulation to evaluate the effectiveness of the proposed system and the QoS provided by the resource allocation algorithm. The simulator was implemented using GridSim [9] toolkit that allows modeling and simulation of distributed system entities for evaluation of scheduling algorithms. Our simulation environment models the following basic entities in addition to existing entities in GridSim:

1) Local user population, which basically models the local user population.
2) GFA, generalized RMS system that we model for Grid-Federation.
3) GFA queue, placeholder for incoming jobs from local user population and the federation.
4) GFA shared federation directory over Peer-to-Peer network, for distributed information management.

A. Workload and Resource Modeling

We based our experiments on real time workload trace data obtained from [1] various resources/supercomputers (See Table-I). The trace data was composed of parallel applications. To enable parallel workload simulation with GridSim, we extended existing GridResource, Alloc Policy and Space Shared entities. For evaluating the QoS driven resource allocation algorithm, we assigned synthetic QoS specification to each resource including the Quote value (Price that cluster owner charges for service) and having varying MIPS rating. The simulation experiments were conducted by utilizing workload trace data over the total period of two days (in simulation units) at all the resources. In experiment 1 and 2 we consider, if the user request can not be served when requested, then it is rejected otherwise it is accepted. During experiment-1 and experiment-2 we measure the following

1) Average resource utilization (Amount of real work that resource does over the simulation period excluding the queue processing and idle time).
2) Job acceptance rate (total percentage of job accepted).
3) Job rejection rate (total percentage of job rejected).
4) Number of jobs locally processed.
5) Number of local jobs migrated to federation.
6) Number of remote jobs processed.

B. Experiment-1 (Independent Resource)

In this experiment the resources were modeled as an independent entity (without federation). All the workload submitted to the resources was processed locally. We evaluate the performance of a resource in terms of average resource utilization, job acceptance rate and job rejection rate. The result of this experiment can be found in (refer to Table-II). We observed that about half of the resources including CTC, KTH SP2, LANL Origin, NASA iPSC, and SDSC Par96 were utilized less than 50%.

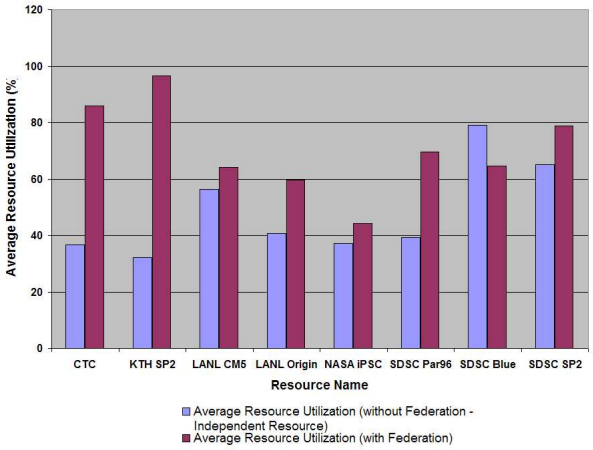

Fig. 3. Average Resource Utilization (%) Vs. Resource Name

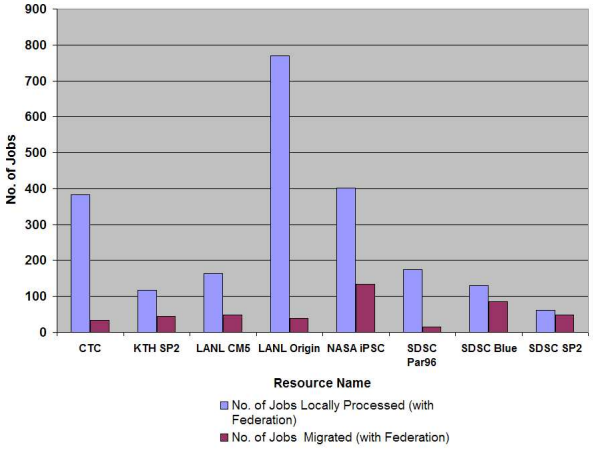

Fig. 4. No. of Jobs Vs. Resource Name

TABLE I
WORKLOAD AND RESOURCE CONFIGURATION

Index	Resource / Cluster Name	Trace Date	Nodes	MIPS (rating)	Jobs	Quote(Price)
1	CTC SP2	June96-May97	512	850	79,302	5.0
2	KTH SP2	Sep96-Aug97	100	900	28,490	5.2
3	LANL CM5	Oct94-Sep96	1024	700	201,387	3.6
4	LANL Origin	Nov99-Apr2000	2048	630	121,989	3.5
5	NASA iPSC	Oct93-Dec93	128	930	42,264	5.3
6	SDSC Par96	Dec95-Dec96	416	710	38,719	3.6
7	SDSC Blue	Apr2000-Jan2003	1152	730	250,440	3.7
8	SDSC SP2	Apr98-Apr2000	128	920	73,496	4.5

C. Experiment-2 (With Federation)

In this experiment we analyzed the workload processing statistics of various resources when they are part of the Grid-Federation, in this case the workload assigned to the resource can be processed locally or may be migrated to any other resource in the federation depending on the availability pattern. Table-III describes the result of this experiment.

D. Experiment-3 (With Federation and Economy)

In this experiment, we study the computational economy metaphor in the Grid-Federation. We assigned QoS parameters (budget and deadline) to all the jobs across the resources. We performed the experiment under three scenarios having different user population profile.

1) All users seek cost-optimization.
2) Even Distribution (50% seeking cost-optimization 50% seeking time-optimization).
3) All users seek time-optimization.

The budget and deadline distribution for the user having the job J_i, seeking cost-optimization is given by $c_{budget} = processingcoston(J_i, R_{c_m})$ (cost of executing the job J_i on the resource R_{c_m}), $m < n$ such that

$$R_{cost}(R_{c_m}) = \frac{\sum_{i=1}^{n}(R_{cost}(R_{c_i}))}{n} \tag{12}$$

where n is the total number of resources in the federation.

$t_{deadline} = executiontimeon(J_i, R_{c_m})$ (Execution time of the job J_i on the resource R_{c_m}) , $m < n$, such that

$$R_{power}(R_{c_m}) = min(R_{power}(R_{c_i})) \quad i = 1...n \tag{13}$$

where n is the total number of resources in the federation.

The budget and deadline distribution for the user having the job J_i, seeking time-optimization is given by $c_{budget} = processingcoston(J_i, R_{c_m})$ (cost of executing the job J_i on the resource R_{c_m}), $m < n$, such that

$$R_{cost}(R_{c_m}) = max(R_{cost}(R_{c_i})) \quad i = 1...n \tag{14}$$

where n is the total number of resources in the federation.

$t_{deadline} = executiontimeon(R_{c_m})$ (Execution time of the job J_i on the resource R_{c_m}), $m < n$, such that

$$R_{power}(R_{c_m}) = \frac{\sum_{i=1}^{n}(R_{power}(R_{c_i}))}{n} \tag{15}$$

where n is the total number of resources in the federation.

E. Results and Observations

During experiment-2, we observed that overall resource utilization of most of the resources increased as compared to experiment-1 (when they were not part of the federation), for instance resource utilization of CTC increased from mere 36.71% to 85.85%. Same trends can be observed in case of other resources too (refer to Fig.3). There was an interesting observation regarding migration of the jobs between the resources in the federation (load-sharing). This characteristic was evident at all the resources including CTC, KTH SP2, NASA iPSC etc. At CTC, which had total 417 jobs to schedule, we observed that 383 (refer to Table-III) of them were executed locally while the remaining 34 jobs migrated and executed at some remote resource in the federation. Also, this resource executed 80 remote jobs, which came from other resources in the federation.

The federation based load-sharing also led to decrease in the total job rejection rate, this can be observed in case of resource LANL CM5 whose job rejection rate decreased from 18.83% to 0.093%. Thus, we conclude that the federation based resource allocation promotes transparent load-sharing between various participant resources, which further helps in enhancing their overall resource utilization and the job acceptance rate.

In experiment-3, we measured the computational economy related behavior of the system in terms of supply-demand pattern, resource owner's incentive (earnings) and end-user's QoS constraint satisfaction (average response time and average budget spent) with varying user population distribution profiles. We study the relationship between resource owner's total incentive and end-user's population profile. Total incentive earned by different resource owners with varying user population profile can be seen in Fig.6. Result shows that the owners (across all the resources) got more incentive when users sought time-optimization (Total Incentive 1.79E+09 Grid Dollars) (scenario-3) as compared to cost-optimization (Total Incentive 1.57E+09 Grid Dollars) (scenario-1). During time-optimization, we observed that there was a uniform distribution of the jobs across all the resources (refer to Fig.5) and every resource owner got some incentive. While during cost-optimization, we observed non-uniform distribution of the jobs in the federation (refer to Fig.5). We observed that some resource owners do not get any incentive (e.g. CTC, KTH SP2, NASA iPSC and SDSC SP2). This can also be

TABLE II
WORKLOAD PROCESSING STATISTICS (WITHOUT FEDERATION - INDEPENDENT PROCESSING/RESOURCE)

Index	Resource / Cluster Name	Average Resource Utilization (%)	Total Job	Total Job Accepted(%)	Total Job Rejected(%)
1	CTC	36.71	417	98.32	1.678
2	KTH SP2	32.132	163	98.15	1.875
3	LANL CM5	56.22	215	81.86	18.83
4	LANL Origin	40.64	817	91.67	8.32
5	NASA iPSC	37.22	535	100	0
6	SDSC Par96	39.30	189	99.4	0.59
7	SDSC Blue	79.16	215	76.2	23.7
8	SDSC SP2	65.18	111	66.66	33.33

TABLE III
WORKLOAD PROCESSING STATISTICS (WITH FEDERATION)

Index	Resource / Cluster Name	Average Resource Utilization (%)	Total Job	Total Job Accepted(%)	Total Job Rejected(%)	No. of Jobs Processed Locally	No. of Jobs Migrated to Federation	No. of Remote jobs processed
1	CTC	85.85	417	100	0	383	34	80
2	KTH SP2	96.50	163	100	0	118	45	44
3	LANL CM5	64.19	215	99.06	0.093	164	49	35
4	LANL Origin	59.61	817	98.89	1.10	769	39	38
5	NASA iPSC	44.16	535	100	0	401	134	69
6	SDSC Par96	69.50	189	100	0	175	14	30
7	SDSC Blue	64.55	215	100	0	130	85	57
8	SDSC SP2	78.80	111	100	0	62	49	96

observed in their resource utilization statistics (refer to Fig.5) which indicates 0% utilization. These resources offered faster (response time) services but at a higher price. This is worst case scenario in terms of resource owner's incentive across all the reosurces.

This also indicates imbalance between the resource supply and demand pattern. As the demand was for the cost-effective resources as compared to the faster one, so these faster but expensive resources remained underutilized. All the jobs in this case were scheduled on other resources (LANL CM5, LANL Origin, SDSC Par96 and SDSC Blue), as they provided cost-effective solution to the users. With even user population distribution (during scenario-2) all the resource owners across the federation got incentive (Total Incentive 1.77E+09 Grid Dollars) and had better resource utilization (refer to Fig.5). This scenario shows balance in the resource supply and demand pattern. Thus, we conclude that resource supply (No. of resource providers) and demand (No. of resource consumers and QoS constraint preference) pattern determines the resource owners overall incentive and the resource usage scenario.

We also measured the end-users QoS satisfaction in terms of average response time and average budget spent under two different optimization scenario (cost and time). We observed that end-users got better average response time (refer to Fig.7) when they sought time optimization (scenario-3) for their jobs as compared to cost-optimization (scenario-1). At LANL Origin (refer to Fig.7) the average response time for the users was 6243.6 simulation seconds (scenario-1) which reduced to 4709.4 during time-optimization. The end-users spent more budget in case of time-optimization as compared cost-optimization (refer to Fig.8). This shows that users get more utility for their QoS constraint parameter response time, if they are ready to spend more budget. Thus, we conclude that in user-centric resource allocation mechanism users have more control over the job scheduling activities and they can express their priorities in terms of QoS constraints.

More experiments related to the Grid system QoS level indicator, resource owner's incentive and end-user's QoS parameter can be found in [31].

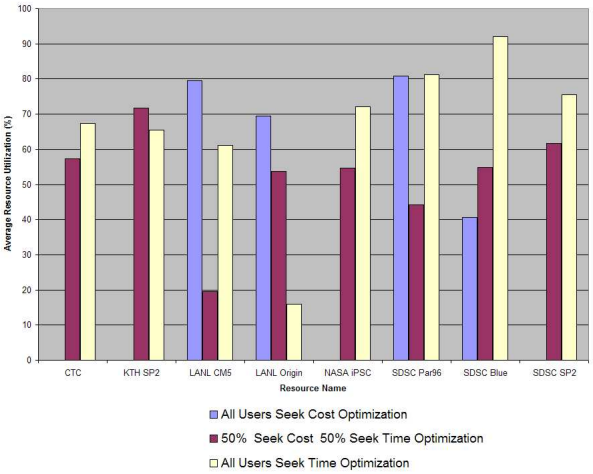

Fig. 5. Average Resource Utilization (%) Vs. Resource Name

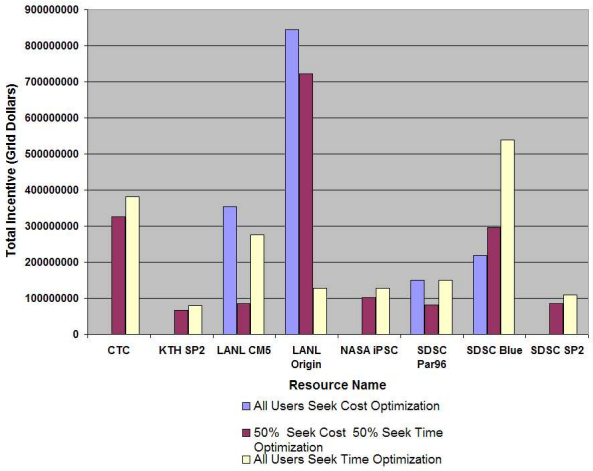

Fig. 6. Total Incentive (Grid Dollars) Vs. Resource Name

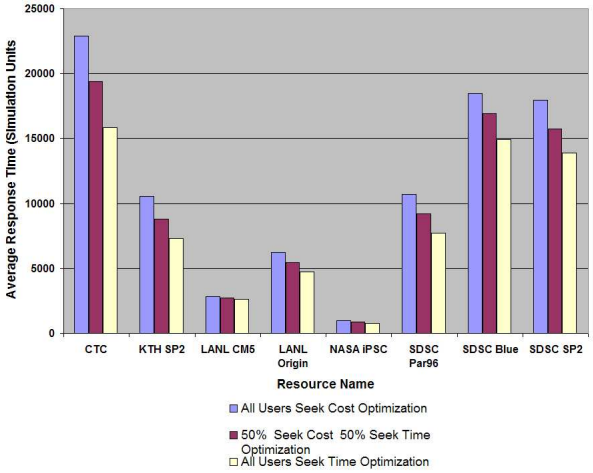

Fig. 7. Average Response Time (Simulation Units) Vs. Resource Name

V. Conclusion

In this paper we proposed a new computational economy based distributed cluster resource management system called Grid-Federation. The experiments prove the effectiveness of the proposed framework, as it leads to better overall resource utilization and load-sharing. The result of the QoS based resource allocation algorithm indicates that the resource supply and demand pattern affects resource provider's overall incentive. We also show that user-centric resource allocation mechanism give users more control on their application scheduling and enable them to express their priorities in terms of QoS constraints. Our future work aims towards investigating co-ordinated QoS of service mechanism in the proposed framework and measuring the network complexity of such a system with large population density of resource providers and consumers. We also intend to look into new QoS constraint based algorithms for scheduling jobs containing parallel applications like MPI or PVM.

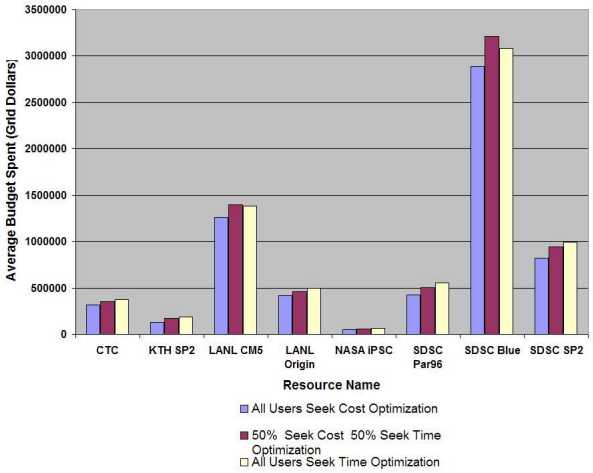

Fig. 8. Average Budget Spent (Grid Dollars) Vs. Resource Name

References

[1] *Parallel Workload Trace, http://www.cs.huji.ac.il/labs/parallel.*

[2] *Platform, http://www.platform.com/products/wm/LSF.*

[3] J. H. Abawajy and S. P. Dandamudi. Distributed hierarchical workstation cluster co-ordination scheme. *(PARELEC'00) August 27 - 30, Quebec, Canada*, 2000.

[4] D. Abramson, R. Buyya, and J. Giddy. A computational economy for grid computing and its implementation in the Nimrod-G resource broker. *Future Generation Computer Systems (FGCS) Journal, Volume 18, Issue 8, Pages: 1061-1074, Elsevier Science, The Netherlands, October*, 2002.

[5] B. Alexander and R. Buyya. Gridbank: A grid accounting services architecture for distributed systems sharing and integration. *Workshop on Internet Computing and E-Commerce, Proceedings of the 17th Annual International Parallel and Distributed Processing Symposium (IPDPS 2003), IEEE Computer Society Press, USA, April 22-26 Nice, France*, 2003.

[6] F. Berman and R. Wolski. The apples project: A status report. *Proceedings of the 8th NEC Research Symposium, Berlin, Germany*, 1997.

[7] B. Bode, D. Halstead, R. Kendall, and D. Jackson. PBS: The portable batch scheduler and the maui scheduler on linux clusters. *Proceedings of the 4th Linux Showcase and Conference, Atlanta, GA, USENIX Press, Berkley, CA, October*, 2000.

[8] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. Economic models for resource management and scheduling in grid computing. *Special Issue on Grid computing Environment, The Journal of concurrency and Computation:Practice and Experience (CCPE), Volume 14, Issue 13-15, Wiley Press*, 2002.

[9] R. Buyya and M. Murched. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Journal of Concurrency and Computation: Practice and Experience;14(13-15), Pages:1175-1220*, 2002.

[10] M. Cai, M. Frank, J. Chen, and P. Szekely. Maan: A Multi-atribute addressable network for grid information services. *Proceedings of the Fourth IEEE/ACM International workshop on Grid Computing*, 2003.

[11] H. Casanova and J. Dongara. Netsolve: A network server solving computational science problem. *International Journal of Supercomputing Applications and High Perfomance Computing;11(3); Pages:212-223*, 1997.

[12] S. Chapin, J. Karpovich, and A. Grimshaw. The legion resource management system. *Proceedings of the 5th Workshop on Job Scheduling Strategies for Parallel Processing, San Juan, Puerto Rico, 16 April, Springer:Berlin*, 1999.

[13] J. Chase, L. Grit, D. Irwin, J. Moore, and S. Sprenkle. Dynamic virtual clusters in a grid site manager. *In the Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), June*, 2003.

[14] G. Cheliotis, C. Kenyon, and R. Buyya. *Grid Economics: 10 Lessons from Finance*. Peer-to-Peer Computing: Evolution of a Disruptive Technology, Ramesh Subramanian and Brian Goodman (editors), Idea Group Publisher, Hershey, PA, USA. (in print), 2004.

[15] M. Chetty and R. Buyya. Weaving computational grids: How analogous are they with electrical grids? *Computing in Science and Engineering (CiSE), The IEEE Computer Society and the American Institute of Physics, USA, July-August*, 2002.

[16] B. Chun and D. Culler. A decentralized, secure remote execution environment for clusters. *Proceedings of the 4th Workshop on Communication, Architecture and Applications for Network-based Parallel Computing, Toulouse, France*, 2000.

[17] B. Chun and D. Culler. User-centric performance analysis of market-based cluster batch schedulers. *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02)*, 2002.

[18] A. B. Downey. Using queue time predictions for processor allocation. *3rd Workshop on Job Scheduling Strategies for Parallel Processing which took place in conjunction with IPPS.*, 1997.

[19] I. Foster and C. Kesselman. The grid: Blueprint for a new computing infrastructure. *Morgan Kaufmann Publishers, USA*, 1998.

[20] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications, Vol. 15, No.3*, 2001.

[21] W. Gentzsh. Sun grid engine: Towards creating a compute power grid. *Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2002.

[22] A. Iamnitchi and I. Foster. On fully decentralized resource discovery in grid environments. *International Workshop on Grid Computing, Denver, CO*, 2001.

[23] J. In, P. Avery, R. Cavanaugh, and S. Ranka. Policy based scheduling for simple quality of service in grid computing. *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, 2004.

[24] N. Kapadia and J. Fortes. Punch: An architecture for web-enabled wide-area network computing. *Cluster computing: The Journal of Networks, Software Tools and Applications;2(2) Pages:153-164*, 1999.

[25] C. Lee, C. Kesselman, J. Stepanek, R. Lindell, S. Hwang, B. Michel, J. Bannister, I. Foster, and A. Roy. Qualis: The quality of service component for the globus metacomputing system. *International Workshop on Quality of Service (IWQoS '98), Pages:140-142*, 1998.

[26] M. Li, X. Sun, and Q. Deng. Authentication and access control in p2p network. *Grid and Cooperative Computing: Second International Workshop, GCC 2003, Shanhai, China, December 7-10*, 2003.

[27] J. Litzkow, M. Livny, and M. W. Mukta. Condor- a hunter of idle workstations. *IEEE*, 1988.

[28] D. Moore and J. Hebeler. *Peer-to-Peer:Building Secure, Scalable, and Manageable Networks*. McGraw-Hill Osborne, 2001.

[29] R. Raman, M. Livny, and M. Solomon. Matchmaking: distributed resource management for high throughput computing. *High Performance Distributed Computing, 1998. Proceedings. The Seventh International Symposium on , 28-31 July*, 1998.

[30] R. Al-Ali F. Rana, D. Walker, S. Jha, and S. Sohail. G-QoSM: Grid service discovery using qos properties. *Concurrency and Computation: Practice and Experience Journal, 16 (5)*, 2004.

[31] R. Ranjan, A. Harwood, and R. Buyya. Grid federation: An economy based distributed resource management system for large-scale resource coupling. *Technical Report, GRIDS-TR-2004-10, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia*, 2004.

[32] J. Sherwani, N. ALi, N. Lotia, Z. Hayat, and R. Buyya. Libra: An economy driven job scheduling system for clusters. *Proceedings of 6th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia'02)*, 2002.

[33] M. Stonebraker, R. Devine, M. Kornacker, W. Litwin, A. Pfeffer, A. Sah, and C. Staelin. An economic paradigm for query processing and data migration in maiposa. *Proceedings of 3rd International Conference on Parallel and Distributed Information Systems, Austin, TX, USA, September 28-30, IEEE CS Press*, 1994.

[34] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering , Vol. 18, No.2, IEEE CS Press, USA, February*, 1992.

[35] J. B. Weissman and A. Grimshaw. Federated model for scheduling in wide-area systems. *Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing (HPDC), Pages:542-550, August*, 1996.

[36] R. Wolski, J. S. Plank, T. Bryan, and J. Brevik. G-commerce: Market formulations controlling resource allocation on the computational grid. *International Parallel and Distributed Processing Symposium (IPDPS), San Francisco, CA, April*, 2001.