International Journal of Emerging Trend in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967 *Volume 2 , Issue 1(Jan-Feb 2015), PP1-5*

Libra-Service Level Agreement for Resource Management and Provisioning

Mr.Pavan Kulkarni
Computer Engineering Department,TCOER,PUNE

Abstract:- The goal of utility computing is to offer computing services as a utility so that users only pay when they need to use. However, most modern high performance computing sources which constitute combination of computers do not consider user-centric service needs for resource management. They still adopt SCRM approaches that focus on optimizing overall. In this evaluated resource policy like LibraSLA will be used to manage penalties for Service Level Agreement (SLA) based resource management. A fundamental problem in building large scale Grid resource sharing system is the need to improve Quality of Service (QoS) to users' applications. In this paper, I propose a new federated Grid system, called Grid-Federation that aims towards decentralized and coordinated coupling of distributed Grid resources as a part of single cooperative system.

Keywords: Cloud Computing, Grid System, Grid-Federation, LibraSLA, System-Centric.

I. INTRODUCTION

This presents a high-level overview of this paper. It provides the motivation to investigate utility-based resource management for cluster computing. It then identifies the principal Research contributions and outlines the organization of this thesis. Our work on utility-based resource management for cluster computing is inspired by the emerging adoption of cluster computing in industry for high-performance, high-availability, and high throughput processing. We focus on the limitations of existing cluster resource management systems in terms of supporting utility-based resource management. To solve this problem, we propose applying market-based resource management. Applying market-based resource management for utility-based cluster resource management also leads to beneficial implications on both Grid computing and commercial offering.

Next-generation scientific research involves solving Grand Challenge Applications (GCAs) that demand ever increasing amounts of computing power. Recently, a new type of High-Performance Computing (HPC) paradigm, called cluster computing ,has become a more viable choice for executing these GCAs since clusters are able to offer equally high-performance with a lower price compared with traditional supercomputing systems. The emergence of clusters is initiated by number of academic projects, such as Beowulf Berkeley NOW and HPVM that prove the advantage of clusters over traditional HPC platforms. These advantages include less costs to access supercomputing, the ability to track technologies, incrementally upgradeable system, open source development platforms, and vendor independence. As shown in Figure 1.1, clusters have been rapidly increasing their market share of the top 500 supercomputing sites over the last six years since their emergence. Today, about 80% of the top 500 supercomputing

PROBLEMS: A cluster is stand-alone computers working together as a single integrated computing resource It uses middleware to create an illusion of a single system and hide the complexities of the underlying cluster architecture from the users. For example, the cluster Resource Management System (RMS) provides a uniform interface for user-level applications to be executed on the cluster and thus hides the existence of multiple cluster nodes from users. The cluster RMS supports four main functionalities: resource management, job queuing, job scheduling, and job execution. It manages and maintains status information of the resources such as processors and disk storage in the cluster. Jobs submitted into the cluster are initially placed into queues until there are available resources to execute the jobs. The cluster RMS then invokes a scheduler to determine how resources are assigned to jobs. After that, the cluster RMS dispatches the jobs to the assigned nodes and manages the job execution processes before returning the results to the users upon job completion. In cluster computing, a provider or producer is the owner of the cluster that provides resources, while a user or consumer makes use of the resources provided by the cluster and can be either a physical human user or a software agent that represents a human user and acts on his behalf. Examples of resources that can be utilized in a cluster are processor power, memory storage and data storage. Thus, a single cluster can have multiple users submitting job requests that need to be completed.

II. PROPOSAL

This thesis advocates the use of market-based resource management or computational economy to achieve utility-based resource management and allocation in clusters since system centric management for shared resources is not effective due to the lack of economic accountability. Market-based approaches can support utility-based computing within a cluster where the utility or value is the monetary payment paid by the users for accessing cluster resources. Market-based resource management is expected to regulate the supply and demand of limited cluster resources at market equilibrium, provide feedback in terms of economic incentives for both cluster users and providers, and promote QoS-based resource allocation that differentiate service requests based on their utility and thus caters to users' needs. A provider and a user have to agree on a Service Level Agreement (SLA) which serves as a contract outlining the expected level of service performance such that the provider is liable to compensate the user for any service under-performance. Cluster RMSs thus need to support SLA-based resource allocations that not only balance competing user needs, but also enhance the profitability of the provider while delivering the expected level of service performance. Although market-based resource management have long been proposed, there are yet any actual implemented market-based RMSs for enabling technologies. But, with numerous recent technological advances that can aid actual deployments of market-based RMSs, it is now timely to examine how market-based solutions can be applied effectively even though there still remain some key challenges that need to be overcome first. This thesis builds upon an earlier work which explores the use of market-based resource management in a cluster RMS via a deadline-based proportional processor share strategy with job admission control called Libra. Libra allocates a minimum proportion of processor share to each job depending on the expected remaining runtime of the job and the amount of time left till the user's specified deadline. The job admission control of Libra accepts jobs only when their deadlines can be met. However, Libra uses a static pricing function that is inflexible and not adaptive. Hence, this thesis proposes an enhanced pricing function called Libra+$ to better support utility-based resource management. Libra also does not consider the possibility of being penalized for failing to meet the required deadline of a job. Thus, this thesis describes a job admission control called LibraSLA to take into account the penalty that may be incurred for accepting new jobs. In addition, Libra requires accurate runtime estimates to ensure that the deadline of accepted jobs is met. But, runtime estimates may be rather inaccurate. Therefore, this thesis proposes a job admission control called LibraRiskD to consider the risk of deadline delay due to the inaccuracy of runtime estimates.

III. GRID COMPUTING

The last few years have seen the emergence of a new generation of distributed systems that scale over the Internet, operate under decentralized settings and are dynamic in their behavior, where participants can leave or join the system at any time. One such system is referred to as Grid Computing and other similar systems include P2P Computing, Semantic Web, Pervasive Computing and Mobile Computing. Grid Computing provides the basic infrastructure required for sharing diverse sets of resources including desktops, computational clusters, supercomputers, storage, data, sensors, applications and online scientific instruments. Grid Computing offers its vast computational power to solve highly challenging problems in science and engineering such as protein folding, high energy physics, financial modeling, earthquake simulation, cli mate/weather modeling, aircraft engine diagnostics, earthquake engineering, virtual observatory, bio-informatics, drug discovery, digital image analysis, astrophysics and multiplayer gaming. The notion of Grid Computing goes well beyond the traditional Parallel and Distributed Computing Systems (PDCS) as it involves various resources that belong to different administrative domains and are controlled by domain specific resource management policies. Furthermore, grids in general have evolved around complex business and service models where various small sites (resource owners) collaborate for computational and economic benefits. The task of resource management and application scheduling over a grid is a complex undertaking due to resource heterogeneity, domain specific policies, dynamic environment, and various socio-economic and political factors.

IV. PROJECT MOTIVATION

The fundamental objective behind the emergence of Grid computing systems is to facilitate a coordinated resource and problem sharing environment among collaborative administrative domains. Grid resource brokering or super scheduling activity is defined as scheduling of jobs across resources that belong to distinct administrative domains. Brokering in computational grids is facilitated by specialized resource brokers such as NASA-Scheduler, Nimrod-G and Condor-G . The main challenges involved with Grid brokering include: (i) GRISfor locating resources that match the job requirements; (ii) coordinating and negotiating SLAs; and (iii) job scheduling. The Grid resources are managed by their respective Local Resource Management System (LRMS) such as Condor, PBS, SGE, Legion, Alchemi and LSF . The LRMS manages job queues, initiating and monitoring their execution. A key consideration involved with the Grid brokering is the *distributed ownership*. As a result, brokers do not have any control over the resources. Further, there is *incomplete system-wide state information* available at brokers, in particular about the arrival pattern, service pattern and composition of jobs across the system.

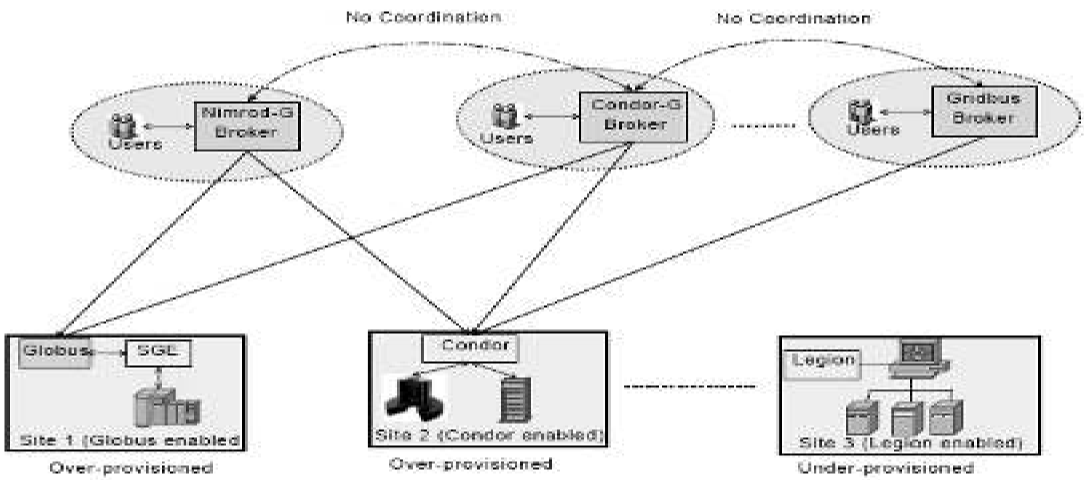

Fig 1.1: Non-Coordinated resource brokering in Grids.

In a non-coordinated system, application schedulers perform scheduling related activities independent of the other schedulers in the system. They directly submit their applications to the responsible LRMS managing the resources *without* taking into account the current load, priorities and utilization scenarios of other application level schedulers. Clearly, this can lead to over-utilization or bottleneck of some valuable resources while leaving others largely underutilized. Furthermore, these brokering systems do not have a coordination (or cooperative) mechanism and this exacerbates the load sharing and utilization problems of distributed resources because sub-optimal schedules are likely to occur. Fig. 1.1 shows such a scenario in which the Nimrod-G, Condor-G and Grid bus resource brokers are over-provisioning the resources at the Site-1 and Site-2 due to lack of coordination. At the same time, resources at Site-3 are left under-provisioned. Another competing approach to the ad-hoc Grid resource assembling is the creation of a distributed Virtual Organization (VO) environment that includes scientific research groups working on collaborative scientific projects. Fig. 1.2 shows a VO based Grid architecture involving three research institutes. Membership to a specific VO is subject to the particular problem solving or project domain. An individual research institute can organize its own resources using a centralized broker which connects to the global VO wide broker. Various VO brokers connect in hierarchy to form a distributed scheduling architecture. A VO user can submit jobs to either its centralized local resource broker or to the VO broker. This solution is fault-tolerant, but a hierarchic connection between VO's can impose serious performance limitations as the number of VOs increases. Further, there is no defined topology in which various institutes can connect with each other. The autonomy of an institute is dependent on the VO broker to which it connects. Any VO specific job migration and resource allocation admission control decision is also taken care of by the VO broker.

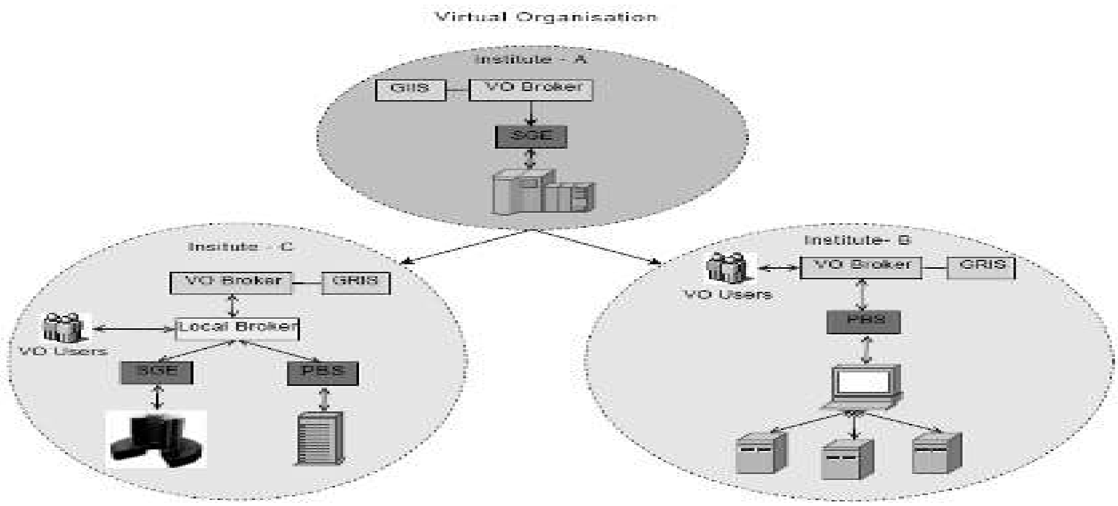

Fig. 1.2: Virtual Organization based Grid organization.

Furthermore, end-users or their application-level schedulers submit jobs to a LRMS without having the knowledge about response time or service utility. The main reason for this being the lack of admission control decision making support between brokers and LRMSes. Sometimes these jobs are being actually processed, leading to degraded QoS. To mitigate such long processing delays and enhance the value of computation, a scheduling strategy can use priorities from competing user jobs that indicate varying levels of importance. LRMS can provide feedback information that minimizes the user from submitting unbounded amounts of work.

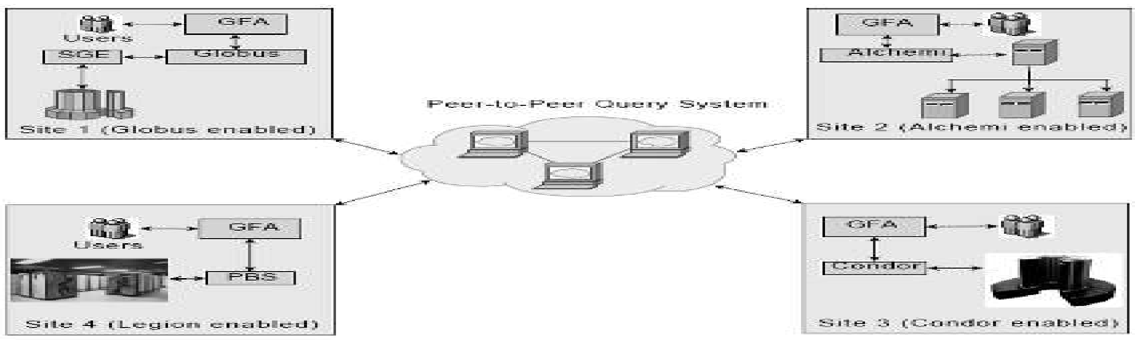

Fig. 1.3: Proposed solution: GF resource sharing system.

V. OBJECTIVE FUNCTION

Grid resources are dynamic in nature whose state can change in very small interval of time, hence it warrants scheduling and resource allocation policies that can adapt to changing conditions. Resources belong to different domains and are controlled by diverse resource management policies. Further, the Grid Participants (GPs) including resource providers and resource consumers associate diverse objective functions with resource allocation and scheduling processes. The resource owners in a grid form a group of participants who make rational choices independently or based on the strategic analysis of what others in the group might do . They like to dictate the access privilege for their resources through diverse sharing policies. Thus, a resource owner enforces the pricing policy, admission control policy and domain specific resource allocation strategy. Similarly, the resource consumers in a grid associate QoS-based utility constraints to their applications and expect that the constraints are satisfied within the acceptable limits. Likewise, resource consumers have diverse QoS-based utility constraints, priority and demand patterns. Exact composition of a GP's objective function is determined by the mechanism design principles. A Grid system based on system centric mechanism defines relatively simple objective function. A system centric scheduler focuses on maximizing

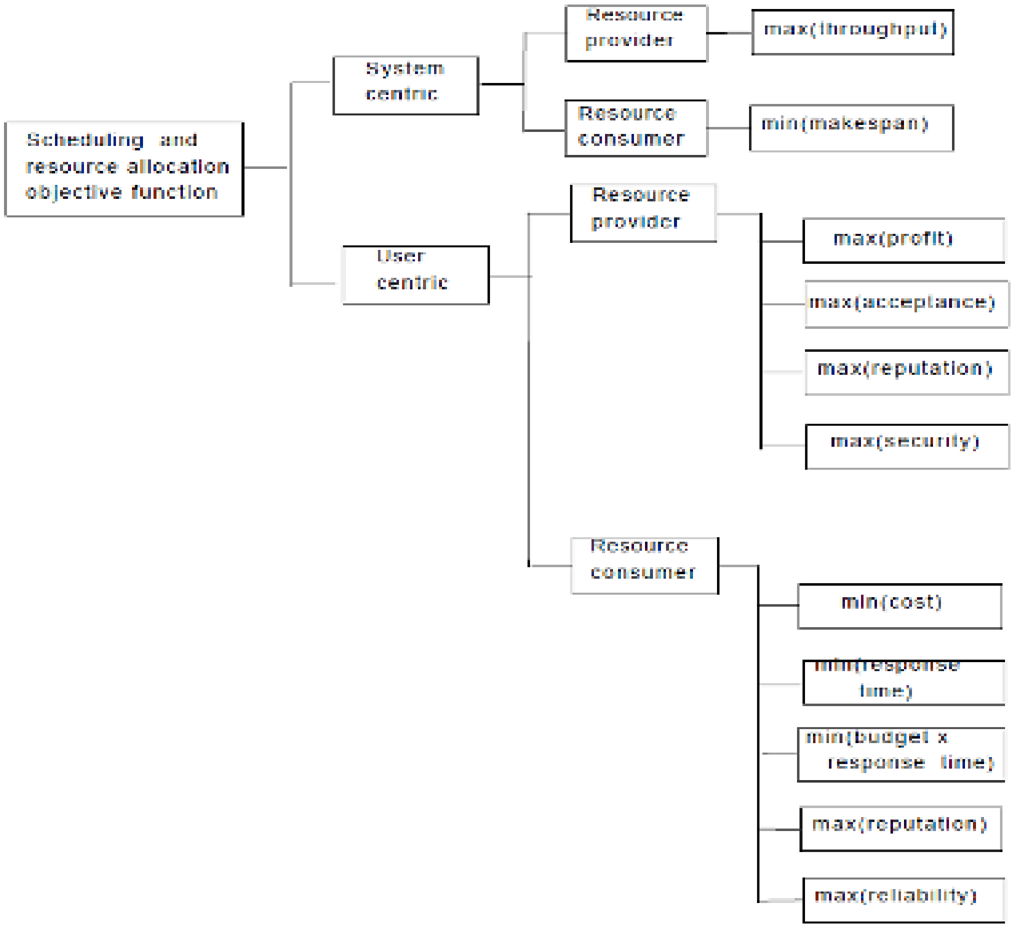

Fig1.4 Grid scheduling and resource allocation objective function taxonomy

re source throughput on the owner side while minimizing overall user's application make span. Grid and Planet Lab systems including Tycoon, Bellagio, Our Grid, Sharp, and Nimrod- G apply market-based economic mechanism for resource management and applicationScheduling. Market driven scheduling mechanisms define user's objective functions based on QoS parameters. These QoS parameters include reputation, budget spent, response time or combination of all. Exact combination of QoS parameters is determined by the applied economic model. Some of the commonly used economic models in resource allocation include the commodity market structure, the posted price structure, the bargaining structure, the tendering/contract-net model, the auction model, the bid-based proportional resource sharing model, the bartering model and the monopoly model. Systems containing Our Grid and Sharp are based on bartering of resources among cooperative domains. In this case, the focus of each participant is on maximizing its bartering reputation in the system. In bartered system, a participant is the consumer as well as the provider at the same time. In cooperative market model, such as the bartered economy, there is singleton objective function shared by both consumer and provider. Competitive market models including commodities market, bid-based proportional sharing, and auction warrants separate objective functions for providers and consumers. Resource owners define objective function with focus on maximizing profit. For this purpose, they can adjust the resource prices dynamically based on supply and demand pattern.

VI. DECENTRALIZED RESOURCE INDEXING

Recently, proposals for decentralizing a GRIS have gained significant momentum. The decentralization of GRIS can overcome the issues related to current centralized and hierarchical organizations. An early proposal for decentralizing Grid information services was made by Iamnitchi and Foster. The task featured a P2P based concept for maintaining the MDS directories in a flat, dynamic P2P network. It envisages that every VO maintains its information services and makes it available as part of a P2P based network. Application schedulers in various VOs initiate a resource look-up query which is forwarded in the P2P network using flooding. However, this approach has a more capacity of network information generated because of flooding. To avoid this, a Time to Live (TTL) field is associated with every message. To an extent, this approach can limit the network message traffic, but the search query results may not be deterministic in all cases. Thus, the proposed approach cannot guarantee to find the desired resource even though it exists in the network.

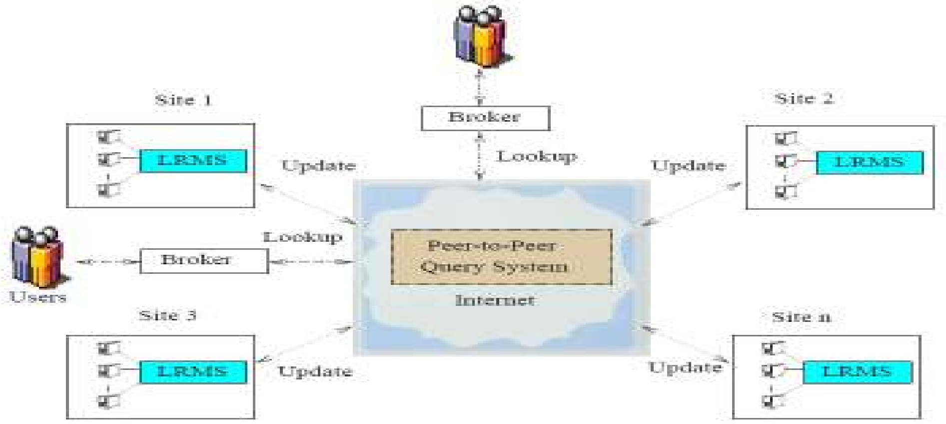

Fig.1.5.Brokering and resource queries

Recently, organizing a GRIS over structured P2P networks has been widely explored. Structured P2P networks offer deterministic search query results with logarithmic bounds on network message complexity. Structured P2P look-up systems including Chord, CAN, Pastry and Tapestry are primarily based on Distributed Hash Tables (DHTs). DHTs provide hash table like functionality at the Internet scale. A DHT is a data structure that associates a key with a data. Entries in the distributed hash table are stored as a (key, data) pair. A data can be looked up within a logarithmic overlay routing hops if the corresponding key is known. Fig. 3.1 shows an abstract model for organizing resource brokering systems over a P2P query system. The brokers access the resource information by issuing lookup queries. The resource providers register the resource information through update queries. It is widely accepted that DHTs are the building blocks for next-generation large scale decentralized systems. Some of the example distributed systems that utilizes DHTRouting substrate include distributed databases, group communication, E-mail services, resource discovery systems [and distributed storage systems Current implementations of DHTs are known to be efficient for 1- dimensional queries such as "find all resources that match the given search point". In this case, distinct attribute values are specified for resource attributes. Extending DHTs to support d-dimensional range queries such as finding all resources that overlap a given search space is a complex problem. Range queries are based on range of values for attributes rather than on a specific value. Current works including have studied and proposed different solutions to this problem.

VII. CONCLUSION

The vision of utility computing is to offer computing services as a utility so that users only pay when they need to use most current high performance computing resources which constitute clusters of computers do not consider user-centric service needs for resource management A fundamental problem in building large scale Grid resource sharing system is the need for efficient and scalable techniques for discovery and provisioning of resources for delivering expected Quality of Service (QoS) to users' applications.

REFERENCES

1)Abawajy, J. H. and Dandamudi, S. P. (2000). Distributed hierarchical workstation cluster co- ordination scheme. In *PARELEC '00: Proceedings of the International Conference on Parallel Computing in Electrical Engineering*, page 111. IEEE Computer Society, Los Alamitos, CA, USA.

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

3)Business Process Execution Language for Web Services version 1.1, July 2005. http://www.ibm.com/developerworks/library/specification/ws-bpel/.

4) V. Akcelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E. J. Kim,J. opez, D. O'Hallaron, T. Tu, and J. Urbanic. High Resolution Forward And Inverse Earthquake Modeling on Terascale Computers. In Proceedings of the 16th Supercomputing Conference (SC 2003), Phoenix, AZ, USA, Nov. 2003. IEEE Computer Society: Los Alamitos, CA, USA.

5)Y. Yemini, A. Dailianas, D. Florissi, and G. Huberman. MarketNet: Protecting Access to Information Systems through Financial Market Controls. Decision Support Systems, 28(1– 2):205–216, Mar. 2000