SERVICE ORIENTED GRID RESOURCE MODELING AND MANAGEMENT

Youcef Derbal
School of Information Technology Management, Ryerson University, 350 Victoria Street, Toronto, ON, M5B 2K3, Canada

Keywords: Grid Resource Model, Computational Grids, Service Oriented Resource Model.

Abstract: Computational grids (CGs) are large scale networks of geographically distributed aggregates of resource clusters that may be contributed by distinct providers. The exploitation of these resources is enabled by a collection of decision-making processes; including resource management and discovery, resource state dissemination, and job scheduling. Traditionally, these mechanisms rely on a physical view of the grid resource model. This entails the need for complex multi-dimensional search strategies and a considerable level of resource state information exchange between the grid management domains. Consequently, it has been difficult to achieve the desirable performance properties of speed, robustness and scalability required for the management of CGs. In this paper we argue that with the adoption of the Service Oriented Architecture (SOA), a logical service-oriented view of the resource model provides the necessary level of abstraction to express the grid capacity to handle the load of hosted services. In this respect, we propose a Service Oriented Model (SOM) that relies on the quantification of the aggregated resource behaviour using a defined service capacity unit that we call servslot. The paper details the development of SOM and highlights the pertinent issues that arise from this new approach. A preliminary exploration of SOM integration as part of a nominal grid architectural framework is provided along with directions for future works.

1 INTRODUCTION

Computational grids are large scale networks of geographically distributed aggregates of resource clusters that may be contributed by distinct providers. These providers may enter into agreements for the purpose of sharing the exploitation of these resources. They may also agree to organize their resources along an economy model to provide commercially viable business services. A comprehensive account of the concepts and issues central to the grid computing framework is given in (Foster and Kesselman, 2004). One central element in CG management is the resource model underlying the various grid decision-making strategies; including resource discovery, resource state dissemination, and job scheduling. Traditionally, the grid resource model is constructed as a dictionary of uniquely identified computing hosts with attributes such as CPU slots, RAM and Disk space, etc. Hence it captures the physical view of the grid resources. Architectural variants of this Physical Resource Model (PRM) have been utilized in the grid management strategies; including resource state dissemination (Iyengar et al., 2004; Krauter et al., 2002; Maheswaran, 2001; Wu et al., 2004); scheduling (Casavant and Kuhl, 1988; He et al., 2003; Spooner et al., 2003; Sun and Ming, 2003; Yang et al., 2003); and resource discovery (Bukhari and Abbas, 2004; Dimakopoulos and Pitoura, 2003; Huang et al., 2004; Ludwig, 2003; Maheswaran, 2001; Zhu and Zhang, 2004). The reliance of the management strategies on the physical view of grid resources entails the need for complex multi-dimensional search strategies and a considerable level of resource state information exchange between the grid management domains. The adoption of the Service Oriented Architecture (SOA) as stipulated by the Open Grid Services Infrastructure (OGSI) recommendation (Tuecke et al., 2003), requires a reformulation of the relationship between the grid resource model and the decision-making mechanisms. In this paper we explore the development of a model of resources that captures a service-oriented logical view of the grid resources (see figure 1).

Derbal Y. (2005).
SERVICE ORIENTED GRID RESOURCE MODELING AND MANAGEMENT.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 146-153
DOI: 10.5220/0001235001460153

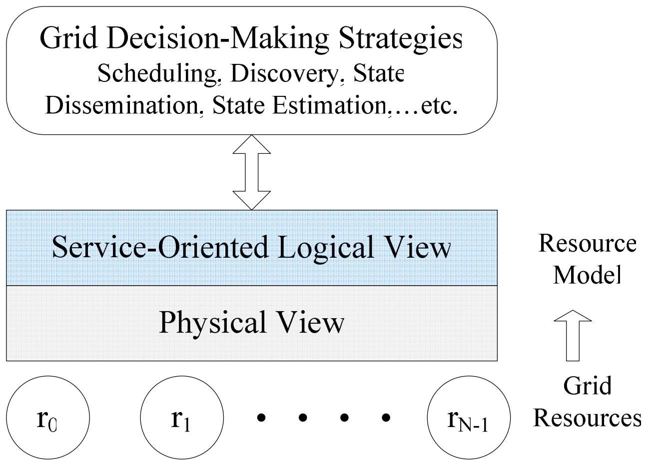

Figure 1: Relationship between the grid resource model and the decision-making strategies

The motivation behind this model is this: the CG ability to handle a service request relies in the end on the available aggregate capacity of the hosting environment to satisfy the resource requirements of the service in question. Hence, it is intuitively more appropriate, for the decision-making strategies, to rely on a capacity model that captures a service-oriented logical view of the aggregated behavior of the resources. To the best of our knowledge, this new approach to grid resource modeling has not been explored with the exception of one related research result reported in (Graupner et al., 2003). In this work a metric, called server share, is introduced to quantify the capacity of a server environment to handle a class of service requests. The server share unit is defined as the maximum server load related to an application deemed to be a benchmark for the class of services in question. This approach has been applied to a single management domain and has resulted in a significant simplification of the management mechanisms (Graupner et al., 2003). Since the definition of the server share unit depends on a chosen benchmark application, an extension of the approach to an open CG environment, which includes distinct management domains, would require a significant standardization effort. Benchmark applications would have to be selected as part of a community standard and thereafter continuously updated to account for new classes of services. Although theoretically conceivable and applicable to a single closed management domain, this approach to the quantification of the aggregated hosting behaviour may not be feasible for the open environment of a commercial or scientific CG. In contrast our proposed quantification of the hosting capacity is independently managed by the provider and therefore will not require any prior normalization among the distinct management domains spanned by a CG.

The paper is organized as follows: Section 2 provides an overview of a nominal architectural organization of CGs. A description of the proposed SOM is given in section 3. Section 4 presents a service management approach associated with the application of SOM. The paper is concluded in section 5.

2 GRID ARCHITECTURAL FRAMEWORK

The framework under consideration views the grid as a dynamic federation of resource clusters contributed by various organizations. Each cluster constitutes a private management domain. It provides a set of grid services which are exposed in compliance with the OGSI recommendation. Clusters may join or leave the grid at any time without any disruption to the grid operation. The effect of this operation is limited to the configuration of neighboring clusters. Each cluster includes a set of agents that host the offered services (figure 2). One agent, labeled the principal, is designated to coordinate inter-cluster operations such as the dissemination of available resource state, and the delegation of service requests. The lines linking the clusters define the pathways of the resource state information exchange. The resulting topology, has the robustness properties of a Power Law network.

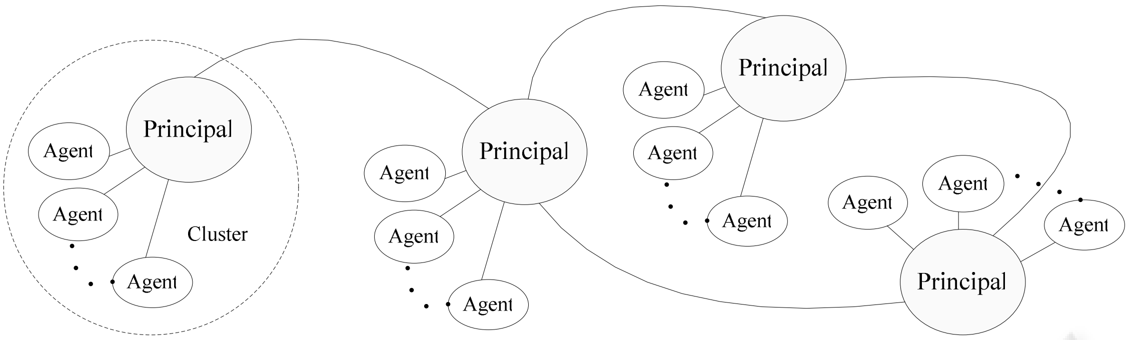

Figure 2: The grid as a federation of cluster

In such network most nodes have few links and few nodes have numerous links (Barabási and Albert, 1999; Faloutsos et al., 1999). In this topology, that we call Grid Neighborhood (GN) topology, there are no restrictions on the IP connectivity between any pair of grid clusters. In fact it is essential that such connectivity be available for the implementation of grid-wide scheduling strategies and service request delegations among peer clusters. Furthermore, we will assume that each resource cluster has the capability to schedule as well as handle service requests for which it has the required resources. Clusters are also assumed to have the capability to delegate the handling of service requests to other clusters in function of some inter-cluster Service Level Agreements (SLAs) (Al-Ali et al., 2004). These agreements would include policies that govern inter-cluster interactions such as the exchange of resource state information.

Many of the assumptions concerning the CG's architectural framework are not strictly necessary. They are stated in order to provide a clearer context for the development of the proposed service-oriented resource model.

3 SERVICE-ORIENTED GRID RESOURCE MODEL

Consider a User Service Request (USR) submitted to a grid cluster. Let us assume that such request requires for its handing the availability of a single grid service. Such availability would necessarily go beyond the assertion that the required grid service is indeed deployed. In particular, the hosting environment has to possess a sufficient resource availability for the instantiation of the grid service in question, the subsequent invocation of its operations, and the maintenance of it state (for a statefull grid service). The required resources may include CPU slots, RAM, other service components, special hardware devices, disk space, swap space, memory cache as well as any required licenses of application software that the service instance may need for its successful operation. If the service needs for its execution a specific OS, some processor architecture, or the presence of a Java Virtual Machine (JVM) and possibly a required heap size, then these would be part of the set of required resources. Let $R = \{r_0, r_1, ..., r_N\}$ be the set of resources owned by a given cluster. Each $r \in R$ takes on a finite or countable number of possible states of availability. The set of all possible states for resource $r \in R$ is denoted by $\Omega^{(r)}$. Then we can define the function $\Phi : (R, \mathbb{N}) \rightarrow \bigcup_{i=0}^{N-1} \Omega^{(r_i)}$ which returns the actual state of a given resource $r \in R$ at a discrete time $n \in \mathbb{N}$. $\mathbb{N}$ is the set of natural numbers. Let $R^{(s)} \subset R$ be the resource subset required for the execution of a service $s \in S = \{s_0, s_1, ..., s_{M-1}\}$, where S is the set of grid services deployed on the cluster in question. These services will have competing needs towards the cluster resources, hence the schematic of figure 3. Let $\Sigma_s^{(n)}$ be the availability state vector associated with the resource set $R^{(s)}$ at time n. Each element of this state vector represents the availability state of a resource $r \in R^{(s)}$, which can be determined using $\Phi(.,.)$. The dynamics of the state vector $\Sigma_s^{(n)}$ reflects the time-varying aggregated capacity of the hosting environment to handle a request targeting service s. For a resource set $R^{(s)}$ of cardinality K, the resource state vector span a set Λ_s defined as the union of all possible ordered K-tuplets $\left(\sigma_x^{(r_0)}, \sigma_x^{(r_1)}, ..., \sigma_x^{(r_{K-1})}\right)$ where $\sigma_x^{(r_i)} \in \Omega^{(r_i)}$, and $r_i \in R^{(s)}, i = 0,.., K-1$. Now consider the point Σ_s^* on the trajectory of $\Sigma_s^{(n)}$ where its elements are at the lowest state of availability that would allow the instantiation, execution and maintenance of a single instance of service s. This is defined as the representative point for the unit service capacity of the hosting environment vis-à-vis service s. This unit will be called a *servslot*. For example, consider a grid service that rely on two physical resources r_a (CPU slots), and r_b (swap memory). Let us assume that an experimental profiling of the aggregated behaviour of r_a and r_b resulted in a trajectory of the state vector for which the hosting environment was able to provide the service (see figure 4). The shaded area represents the regions of Λ_s where the hosting environment was not able to offer the grid service in question for lack of resource availability. Now let us define the subset $\Gamma_s \subset \mathrm{T}_s$ so that for any $z \in \Gamma_s$ we have:

$$d(z, x_0) = \min_{x \in \mathrm{T}_s} d(z, x_0) \tag{1}$$

$$d(x, y) = \frac{1}{K} \sqrt{\sum_{i=0}^{k-1} \left(\eta(\sigma_x^{(r_i)}, \sigma_y^{(r_i)})\right)^2} \tag{2}$$

T_s represents the points on the trajectory of the resource state vector corresponding to sufficient resource availability.

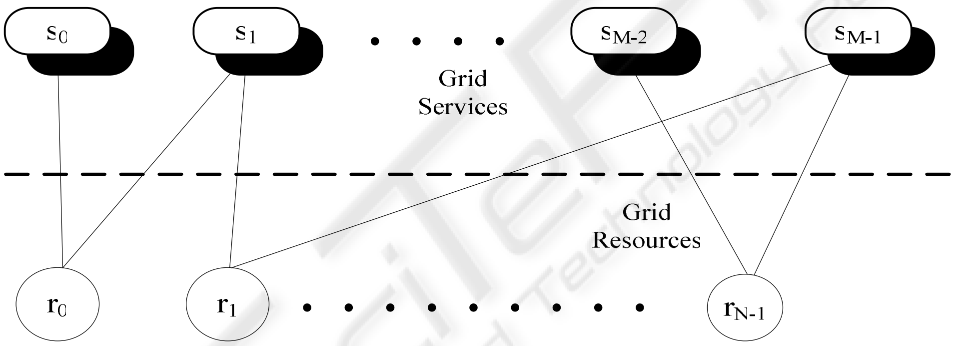

Figure 3: Relationship between the resources and the grid services

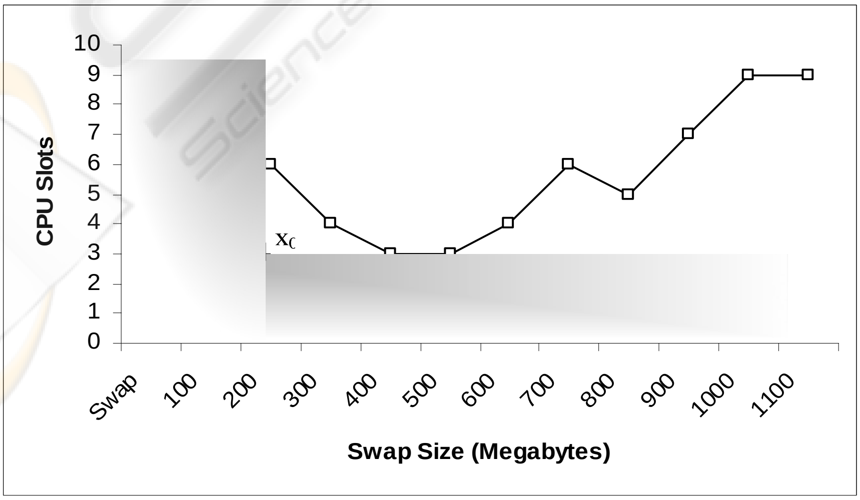

Figure 4: An example of a resource availability profile

In the above relations x and y are two points of the $\Sigma_s^{(n)}$ trajectory, and $\sigma_x^{(r_i)}, \sigma_y^{(r_i)} \in \Omega^{(r_i)}$ are their respective coordinates. $d:(\Lambda_s, \Lambda_s) \rightarrow \mathbb{R}_+$ defines the distance between two points of the state vector. $\mathbb{R}_+$ is the set of non-negative real numbers. The state distance $\eta : (\bigcup_{i=0}^{K-1} \Omega^{(r_i)}, \bigcup_{i=0}^{K-1} \Omega^{(r_i)}) \rightarrow \mathbb{R}_+$ is defined in a fashion that fits the nature of the resource in question. For a resource such as CPU slots, the state distance is simply the difference between the numbers of slots associated with the two respective states. For a resource such as an Operating System (OS) or a CPU architecture, the distance between two states may either be 0 or infinity. In the first case the two states represent the same OS, while in the second case they represent two different Operation Systems. Let $\succ$ be a partial ordering on $R^{(s)}$ where $r_a \succ r_b$, $r_a, r_b \in R^{(s)}$ if and only if r_a is said to be more costly than r_b according to some chosen resource cost function. Without loss of generality let us assume that the ordering on the K dimensional $R^{(s)}$ resulted in the relation $r_0 \succ r_1 \succ r_2 ... \succ r_{K-1}$. Then the previously introduced special point Σ_s^* of the state vector used to define the servslot unit can now be quantitatively determined as the element $y \in \Gamma_s$ that satisfies:

$$\eta(\sigma_y^{(r_i)}, \sigma_{x_0}^{(r_i)}) = \min_{x \in \Gamma_s} \eta(\sigma_x^{(r_i)}, \sigma_{x_0}^{(r_i)}) \quad (3)$$

Where $i = 0,...j$, and $j \le K$ is the smallest value for which the above relation is satisfied. In summary relation (3) allows the selection of the element of the set Γ_s whose most "costly" resource coordinates are the closest to the resource coordinates of the point x_0. If it was found that $j = K$, then the elements of Γ_s are equidistantly located relative to x_0, and hence any element of the set can be used to define the servslot unit. With the definition of a *servslot* as a unit of the aggregated capacity of the hosting environment, each cluster can maintain an updated registry of hosted services and their respective available service capacities expressed in servslots (see Table 1).

Table 1: SOM Registry – Example

Service Name	Service Capacity (servslots)	Description
s_0	4	…
s_1	3	….
…		

The collection of service registries associated with the individual grid clusters make up the service-oriented logical view of the grid resource model. As mentioned earlier, the quantification of the aggregated capacity of the hosting environment simplifies the grid management mechanisms. For instance, the resource state information would be exchanged through the dissemination of the capacities of the hosted services rather than using a dictionary of resource names and their extensive attribute name-value maps. However, since the servslot unit is tied to a service or a class of services, a quantitative description of the servslot unit, i.e. detail description of the Σ_s^* point of the resource state vector, has to be published as part of the service description. For the service discovery and scheduling, the use of a service-oriented resource model (see Table 1), clearly reduces the complexity of the search processes inevitably present in both decision-making mechanisms.

The practical utility of the above model relies on the definition of the servslot unit, itself dependent on the profiling of the dynamics of the resource state vectors associated with the hosted grid services. In practice, such profiling poses a considerable challenge. The number of agent hosts associated with a single cluster is expected to be very high (>100,000). The profiling of the availability of the hosted services (i.e. the dynamics of their associated resource state vectors) requires a service availability monitoring and management mechanism that operates at the agent level as well as the cluster level. Such mechanism would have to be scalable to handle large clusters. It also needs to be sufficiently sophisticated to deal with the fact that different grid services share common resources ($\bigcap_{s \in S} R^{(s)} \neq \{\varnothing\}$). However, because the process is independently carried out by the various management domains we believe it to be practically feasible. Furthermore, it is expected that the rapid advances in autonomic computing will provide the necessary approaches and technologies to address the above challenges of service profiling, monitoring and management mechanisms (Lanfranchi et al., 2003).

The next section explores a service management approach that reflects the integration of the proposed service-oriented resource model with the grid decision-making mechanisms.

4 SERVICE MANAGEMENT

The proposed service-oriented grid resource model is a necessary step in the progression towards the full adoption of the service oriented architecture for grid systems. With the expected reduction in the complexity of grid management processes, the realization of the proposed model requires a service management framework that incorporates the following:

1. A mechanism for the monitoring and management of service availability. As the grid services claim and use the shared resources of their home cluster, their available capacities has to be updated accordingly. In other words the actual availability of the physical resources (CPU, RAM, Swap, Disk space, etc.) has to be regularly translated into service capacity values of the dependent services.

2. A calibration process that should be executed on service deployment in order to determine the Σ_s^* point and hence specify the quantitative definition of the servslot unit associated with the service. During this process, configuration parameters may be provided, as part of the service deployment file, to define the states (i.e. $\Omega^{(r)}$) associated with the individual resources required by the service.

3. A cluster-bound resource registry that maintains an up to date dictionary of the cluster resources and their current attribute values. This may be called the physical model of the grid resources. The model may be maintained as an XML document that complies with a community standard schema. Alternatively, a representation similar to the Microsoft Common Information Model (CIM) (Desktop Management Task Force, 1999.) may be used for the purpose.

4. A cluster-bound SOM registry (see Table 1) that maintains a record of the capacity of the hosted grid services.

Figure 5 illustrates the possible integration of these elements as part of a nominal CG architecture. The Service Manager (SM) is responsible for the calibration mechanism as well as the monitoring and management of the service availability. The resource load accounting maintained by the Cluster Resource Manager is used by the SM to regularly update the available capacities of the hosted services. This requires a well defined mapping between the residual levels of the grid resources and the available service capacity. For a set $S=\{s_0, s_1, ..., s_{M-1}\}$ of services that relies on a K -dimensional set $R^{(s)}$ of resources, the mapping between the service capacities ad the resource states of availability can be written as follows:

$$\begin{bmatrix} c_0(n) \\ . \\ . \\ c_{M-1}(n) \end{bmatrix} = \Theta. \begin{bmatrix} \gamma\left(\Phi(r_0, n)\right) \\ . \\ . \\ \gamma\left(\Phi(r_{K-1}, n)\right) \end{bmatrix} \quad (4)$$

$c_i(n)$, $i=0,...,M-1$ is the service capacity of s_i at the discrete time n . $\gamma\left(\Phi(r_j, n)\right)$, $j=0,...,K-1$, represents the residual level of resource r_j . The function $\gamma(.)$ is a normalization function that translates the availability state of a resource into a non-negative real number which represents the residual level of the resource. The $M \times K$ matrix Θ is called the Resource-Service Mapping (RSM) matrix. The element of Θ are non-negative real numbers. The proposed formulation of the service-resource mapping given in relation (4) illustrates:

1. The coupling between distinct grid services due to their reliance on common resources.

2. The aggregation of resource availability levels into service capacities.

3. The relationship between the physical view of the grid resource model (resources) and the logical view of the model (services).

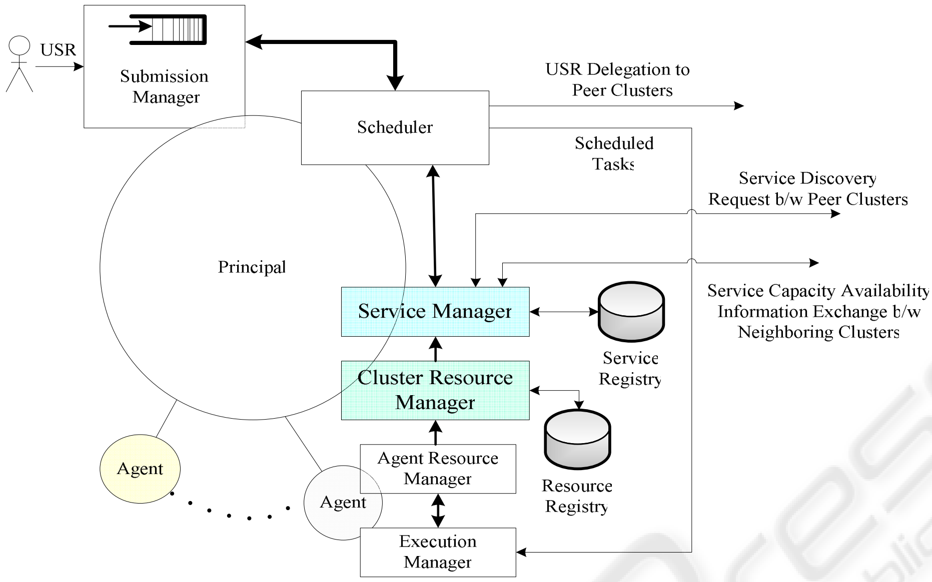

Figure 5: Cluster architecture to reflect the inclusion of the proposed service oriented grid resource model.

The element of the RSM matrix may not necessarily be constant. However, it is reasonable to assume that they are slowly time-varying so that we can write the following relation:

$$\Theta(n) = \Theta_0 + \Delta\Theta(n) \quad (5)$$

Θ_0 may be determined through tests of service requests submitted to the hosting environment. $\Delta\Theta(n)$ is a variable term to be adaptively updated during the operation of the grid. One of the candidate adaptation signals is the discrepancy between the published service capacity and the observed performance data that may be collected from the hosting environment while handling the services in question. There is a large pool of adaptation strategies available from control theory that can be used for this purpose (Cangussu et al., 2004; Zhu and Pagilla, 2003). In (Reed and Mendes, 2005), a dynamic adaptation provides a soft performance assurance for grid applications that rely on shared resources. The offered insight is an encouraging step towards the future development of adaptation strategies in CGs; including the synthesis of adaptive estimation schemes for the problem at hand. Given the limited space of this paper, the adaptive estimation of the RSM matrix will be addressed in future works.

5 CONCLUSIONS

In light of the adopted SOA for CGs, the paper questions the utilization of the physical view of the grid resource model in the grid decision-making mechanisms. A proposed service oriented logical view of the grid resource model is argued to be a more appropriate alternative for the new service oriented grid computing paradigm. The new approach relies on the quantification of the aggregated resource behavior into a service hosting capacity metric of the grid environment. A preliminary analysis of the proposed model reveals numerous challenging issues relevant to its application, implementation and integration within a nominal architecture of CGs. A formulation of a subset of these issues is presented along with directions for future works.

REFERENCES

Al-Ali, R., Hafid, A., Rana, O., and Walker, D., 2004. An approach for quality of service adaptation in service oriented Grids. *Concurrency Computation Practice and Experience.* Vol. 16, no. 5, pp. 401-412.

Barabási, A., and Albert, R., 1999. Emergence of Scaling in Random Networks. *Science.* Vol. 286, no. 5489, pp. 509-512.

Bukhari, U., and Abbas, F., 2004. A comparative study of naming, resolution and discovery schemes for networked environments. In *Proceedings - Second Annual Conference on Communication Networks and Services Research*, pp. 265 - 272.

Cangussu, J. W., Cooper, K., and Li, C., 2004. A control theory based framework for dynamic adaptable systems. In *Proceedings of the ACM Symposium on Applied Computing*, pp. 1546-1553.

Casavant, T. L., and Kuhl, J. G., 1988. A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems. *IEEE Transactions on Software Engineering.* Vol. 14, no. 2**,** pp. 141-155.

Desktop Management Task Force, 1999., Common Information Model (CIM), http://www.dmtf.org/spec/cims.html.

Dimakopoulos, V. V., and Pitoura, E., 2003. A peer-to-peer approach to resource discovery in multi-agent systems. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, pp. 272.

Faloutsos, M., Faloutsos, P., and Faloutsos, C., 1999. On power-law relationships of the Internet topology. *Computer Communication Review.* Vol. 29, no. 4**,** pp. 251-262.

Foster, I., and Kesselman, C., 2004, The grid: blueprint for a new computing infrastructure, Morgan Kaufmann;Elsevier Science, San Francisco, Calif.Oxford, pp. 748.

Graupner, S., Kotov, V., Andrzejak, A., and Trinks, H., 2003. Service-centric globally distributed computing. *IEEE Internet Computing.* Vol. 7, no. 4**,** pp. 36 - 43.

He, X., Sun, X., and Von Laszewski, G., 2003. QoS guided Min-Min heuristic for grid task scheduling. *Journal of Computer Science and Technology.* Vol. 18, no. 4**,** pp. 442-451.

Huang, Z., Gu, L., Du, B., and He, C., 2004. Grid resource specification language based on XML and its usage in resource registry meta-service. In *Proceedings - 2004 IEEE International Conference on Services Computing, SCC 2004*, pp. 467 - 470.

Iyengar, V., Tilak, S., Lewis, M. J., and Abu-Ghazaleh, N. B., 2004. Non-Uniform Information Dissemination for Dynamic Grid Resource Discovery. In, pp.

Krauter, K., Buyya, R., and Maheswaran, M., 2002. A taxonomy and survey of grid resource management systems for distributed computing. *Software - Practice and Experience.* Vol. 32, no. 2**,** pp. 135-164.

Lanfranchi, G., Peruta, P. D., and Perrone, A., 2003. Toward a new landscape of systems management in an autonomic computing environment. *IBM Systems Journal [H.W. Wilson - AST].* Vol. 42, no. 1**,** pp. 119.

Ludwig, S. A., 2003. Comparison of centralized and decentralized service discovery in a grid environment. In *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems*, pp. 12.

Maheswaran, M., 2001. Data dissemination approaches for performance discovery in grid computing systems. In *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS '01)*, pp. 910 - 923.

Reed, D. A., and Mendes, C. L., 2005. Intelligent Monitoring for Adaptation in Grid Applications. *Proceedings of the IEEE.* Vol. 93, no. 2**,** pp. 426 - 435.

Spooner, D. P., Jarvis, S. A., Cao, J., Saini, S., and Nudd, G. R., 2003. Local grid scheduling techniques using performance prediction. *IEE Proceedings: Computers and Digital Techniques.* Vol. 150, no. 2**,** pp. 87-96.

Sun, X.-H., and Ming, W., 2003. Grid Harvest Service: a system for long-term, application-level task scheduling. In *Parallel and Distributed Processing Symposium*, pp.

Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D., and Vanderbilt, P., 2003, Open Grid Services Infrastructure (OGSI), Global Grid Forum.

Wu, X.-C., Li, H., and Ju, J.-B., 2004. A prototype of dynamically disseminating and discovering resource information for resource managements in computational grid. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, pp. 2893-2898.

Yang, L., Schopf, J. M., and Foster, I., 2003. Conservative Scheduling: Using Predicted Variance to Improve Scheduling Decisions in Dynamic Environments. In *Proceedings of Supercomputing 2003*, pp.

Zhu, Y., and Pagilla, P. R., 2003. Adaptive Estimation of Time-Varying Parameters in Linear Systems. In *Proceedings of the American Control Conference*, pp. 4167-4172.

Zhu, Y., and Zhang, J.-L., 2004. Distributed storage based on intelligent agent. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, pp. 297-301.