

MULTIDIMENSIONAL QoS ORIENTED TASK SCHEDULING IN GRID ENVIRONMENTS

Amit Agarwal and Padam Kumar

Department of Electronics & Computer Engineering,
Indian Institute of Technology Roorkee, Roorkee, India
{aamitdec, padamfec}@iitr.ernet.in

ABSTRACT

High availability of resources is the key requirement in grids where computational and network resources are highly dynamic and unpredictable. QoS (Quality of Service) based resource management and scheduling is an important issue in such dynamic and heterogeneous grid environments. The computational resources in the grid vary from dedicated clusters to non-dedicated desktops which are connected over communication networks of diverse bandwidths. The task-resource mapping in such a non-deterministic computing environment leads to concerns over QoS constraints imposed by application tasks to identify suitable resources. In this paper, we consider bandwidth requirement as a QoS of a network and availability requirement as a QoS of a compute resource. An Availability aware QoS oriented Algorithm (AQuA) for task scheduling in grids has been proposed. The algorithm is evaluated and compared with QGMM (QoS Guided Min-Min) algorithm in the simulated grid environment. The experimental analysis and results show that the AQuA algorithm is able to utilize the highly available resources in grid and therefore increases the reliability to successfully execute applications without adversely affecting the makespan.

KEYWORDS

Quality of Service (QoS), Scheduling, Grid Computing, Availability, Bandwidth, Reliability

1. INTRODUCTION

The Grid [1] is a unified computing platform which consists of heterogeneous resources (e.g. processors, data storages, catalogs, networks and sensors etc) over large geographical regions that exhibit different availability patterns over time due to administrative policies of different domains. The Grid consists of several resources that are supposed to work always, still there are lots of problems that cause some failures like hardware (hard disk, memory, network, etc), software, operator errors, network problems etc and the restore time may vary between minutes to several days. The resources are not always available to solve the problems due to launches, nights, weekends and holidays. There exist many high performance applications like healthcare applications, finance management applications, military applications etc which may require resources with high availability as serious performance degradation could occur at specific time when even only one computing resource becomes unavailable.

The basic assumption in scheduling theory is that resources are always available for computation [2]. This assumption might be reasonable in few cases, but it is not valid in grid scenario where resources are not continuously available due to certain maintenance requirements, breakdowns, administrative policies or other constraints. Availability can be defined as the total time a computing resource is functional during a given interval. A large number of grid resources may be unavailable at any time due to wide range of policies for when and how to make these resources available to the Grid. These policies are based on resources'

usage patterns and owners' other preferences, like shutting down the resources during night. To obtain a highly available resource is an issue as resource is inevitably unreliable and dynamic in grid computing environment. Therefore, there must be a mechanism to evaluate and manage the availability levels of grid resources while scheduling the tasks.

The grid network infrastructure consists of diverse sets of heterogeneous communication links between different computing resources and the links may have different bandwidths due to non-dedicated networks. The task scheduling is to map the tasks onto the distributed computing resources in order to obtain a schedule with an optimum objective. In this paper, task scheduling focuses on a set of independent tasks with QoS constraints. For example, some critical tasks may require bandwidth not less than 1.0 Giga bits/sec and availability of computing resources should be greater than 90% for their execution.

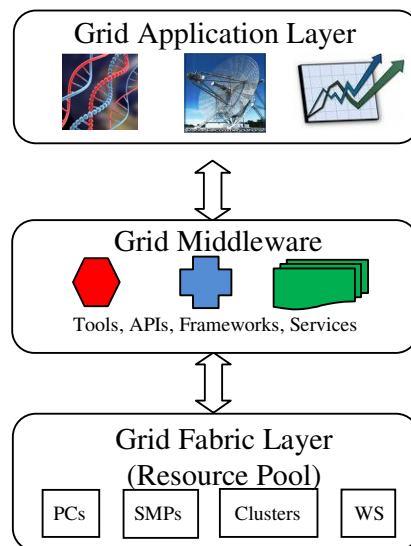


Figure1. Layered Grid Architecture.

The three layer grid architecture is depicted in figure 1. This consists of three layers namely application layer, middleware, and fabric layer. The grid application layer contains grid enabled applications. The grid middleware represents the software infrastructure that facilitates security, resource management, data access, instrumentation, policy, accounting and other services required by application users and resource providers to operate effectively in grid environment. There exist several popular middleware like Globus, Gridbus, Alchemi, gLite, Legion etc. The fabric layer contains the resource pool of different resources joining the grid.

Scheduling is an integral part of grid computing. In traditional scheduling problem, tasks are scheduled across a parallel machine in order to minimize the total turn around time. In a super computer centre, a set of applications from different users is assigned to the parallel machine while maximizing system utilization. In grids, the scheduling problems get amplified while scheduling a number of such centres. The objectives of grid wide scheduling algorithms are to maximize throughput and system utilization while meeting the user requirements/ constraints.

The Grid scheduler needs to consider the QoS requirements of application tasks to discover a set of potential resources that meets QoS requirements. Then, application tasks are assigned over grid resources in order to get a better match between tasks and resources which meets predefined scheduling criteria such as minimizing the makespan. The meaning of QoS is highly dependent on particular applications, from hardware capacity to software existence. Usually, QoS is a constraint imposed on the scheduling process instead of the final objective function [3].

To address the challenges of non-dedicated computing and network resources in the Grid, we adopt the prediction models [4, 5]. A newly proposed long-term application-level prediction model [6] is exploited to predict the task execution time for a task/resource pair in a dynamic running environment for large applications. In this paper, we embed the QoS information into the scheduling algorithm to make a better match among different levels of QoS request/ supply. The grid schedulers and resource brokers need information about resource availability properties and predictions about their future availability.

Most conventional scheduling algorithms have concentrated only on high throughput with the goal of reducing makespan, completely ignoring the availability and bandwidth requirements of tasks. To obtain a resource with high availability is an open issue in the Grid as the resources are inevitably unreliable and dynamic. This motivates us to adopt the availability and bandwidth constraints of tasks while scheduling in grid scenario. Our scheduling model addresses the match of QoS request from application to the QoS provided by the grid resources. Most current QoS concerns at the resource management level [7]. However, we embed the QoS information into the scheduling algorithm to make a better match among different levels of QoS request/ supply. The involvement of QoS has an effect on the resource selection step in the scheduling which influences the final objective function.

This paper is organized as follows: Section 2 presents the related work. Section 3 describes task scheduling problem in grid with multiple QoS constraints. Section 4 and 5 presents the grid resource model and application model respectively. In section 6, the proposed scheduling algorithm (AQuA) is presented. In Section 7, experimental results are presented and discussed. Section 8 concludes the proposed research work.

2. RELATED WORK

QoS resource management and scheduling is an important component of the grid. Currently, QoS-based scheduling is hot research area in the grid community. In literature [8, 9, 10, 11], many heuristics have been proposed to obtain the optimal match between application tasks and resources. The scheduling is performed in multiple steps to solve the problem of matching application requirements with resource availability while providing quality of service. It is challenging to achieve high throughput and high availability requirements simultaneously, because these are two conflicting objectives. The problem of finding the best pairs of application tasks and resources is an NP-complete problem [12]. Braun et al. [8] and Maheswaran et al. [11] proposed simple heuristics for dynamic matching and scheduling of a class of independent tasks onto a heterogeneous computing system. An extended suffrage heuristic was presented by Casanova et al. [9] for scheduling parameter sweep applications based on AppLes [10] framework that takes the advantage of file sharing to achieve better performance under a wide variety of load conditions.

The Min-Min heuristics [13] is becoming the benchmark of such kinds of task-resource scheduling problems. Braun et al. [14] compared 11 static heuristics including Min-Min, Max-

Min, A* and GA. Experimental results show that GA (Genetic algorithm) and Min-Min outperforms the others. The GA consistently gives the best performance. But, if the execution time of mapping is also taken into account, Min-Min can give the best performance with very small execution time. He et al. [15] presented a QoS guided Min-Min (QGMM) heuristic for independent task scheduling problems, but it does not consider the availability requirements of application tasks for scheduling. Foster et al. [16] presented general purpose architecture for reservation and allocation (GARA) that supports flow-specific QoS specification, immediate and advance reservation, and online monitoring and control of both individual resources and heterogeneous resource ensembles. Dogan et al. [17] consider the problem of scheduling a set of independent tasks with multiple QoS requirements, which may include timeliness, reliability, security, version, and priority in a grid in which resource prices can vary with time during scheduling time intervals. In [18], Golconda et al. compare five QoS-based scheduling heuristic from the literature in terms of three performance parameters, namely number of satisfied users, makespan, and total utility of meta-task.

The main contribution of this paper is to consider QoS constraints (i.e. availability of computation resources and bandwidth of network resources) of applications users while selecting the suitable resources in the grid environments. The experimental analysis and results show that the proposed algorithm (AQuA) is able to utilize the highly available resources in grid and therefore increases the reliability to successfully execute applications without adversely affecting the makespan.

3. PROBLEM STATEMENT

The problem of optimally scheduling large, diverse sets of tasks onto resources of a distributed heterogeneous computing environment has been shown to be NP-complete [14] and attracts more and more researcher's attention to propose new heuristics, such as Min-Min, Simulated Annealing and Genetic Algorithm (GA). In the Grid, task scheduling is a process to assign the tasks onto the available distributed computing resources in order to obtain a schedule with optimum objective such as minimization of makespan.

Embedding multi-dimensional QoS into task scheduling is an open problem and an issue which has been addressed in this work. In this paper, the task scheduling focuses on a set of independent tasks with QoS constraints imposed by application tasks. Availability and bandwidth are two QoS constraints in resource selection for successful execution of tasks. The primary objective of any scheduling problem is to obtain a schedule with minimum execution time (or makespan). The makespan is the maximum completion time, i.e. the maximum duration from the starting time of the first task to the finish time of the last task in a metatask (or application) scheduled on grid resources. The makespan of the complete schedule is defined as $\max_{t_i \in K} (CT_i)$. Makespan is the measure of throughput in independent task scheduling in grid computing system. The objective of the grid scheduling algorithm is to minimize the makespan while maximizing its reliability. Reliability is a function of node's availability levels allocated for application tasks in grid.

4. THE GRID RESOURCE MODEL

Most existing scheduling heuristics developed for distributed computing systems do not factor in availability requirements imposed by application tasks. To explore this issue, each computing node in grid is modelled using nodes' computing capacity and availability. A Grid system can be modelled as a finite set of heterogeneous computing resources as:

$$R = \{r_1, r_2, \dots, r_n\} \tag{1}$$

Each resource is characterized by its bandwidth capacity and availability level. These resources are not entirely dedicated to the Grid. Based on resource availability policies, we can classify them in three main categories as in Austrian Grid [19]: (1) the dedicated resources, which are meant to be always available to grid users; (2) the non-dedicated resources, which are available in grid as long as they are turned on; and (3) the on-demand resources, which are made available to grid only on demand from the users for executing large scale applications or experiments. This classification in turn reflects the availability of the resources.

Definition 1: *Availability is the probability that the system is continuously available at any random period of time. It can be computed as the ratio of the total time a computing resource is functional during a given interval.*

Definition 2: *Bandwidth is the network capacity (in bits per second) to transfer data per unit time.*

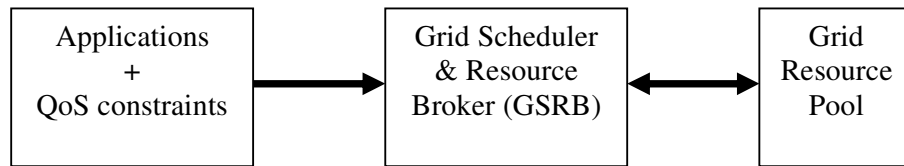


Figure 2. QoS based Grid Scheduling System.

The availability and capacity of grid resources, e.g. number of hosts and network links, fluctuates. Thus, to determine a priori the completion time of a task on a particular resource is difficult. Moreover, a task may fail to complete the execution due to failure of the resource or network link. In this paper, we assume that an entity (or distributed entities) exists that computes and provides prediction information of resources' availability and links' bandwidth in the grid. In case of network bandwidth, we consider the dynamically forecasted prediction information by facilities such as the Network Weather Service (NWS) [20].

The objective of the proposed scheduling algorithm is to obtain a good task allocation decision for application tasks to satisfy their multi-dimensional QoS requirements and maintain an ideal performance in the makespan.

5. APPLICATION MODEL

An application can be modelled as a set of m independent tasks (or meta-task) as follows:

$$T = \{t_1, t_2, \dots, t_m\} \tag{2}$$

The expected execution time for each task on different resource is not identical which is known prior to execution and represented by a *ETC* (expected time to compute) matrix. The completion time of task t_i on resource r_j can be computed as:

$$CT_{ij} = ETC_{ij} + AT(r_j) \tag{3}$$

where $AT(r_j)$ is the earliest time when resource r_j is available to start execution of task t_i . Initially, $AT(r_j) = 0$ for all resources $r_j \in R$.

6. THE PROPOSED ALGORITHM (AQUA)

A QoS based grid scheduling system architecture is illustrated in figure 2. In this model, user's application tasks with QoS requirements are submitted to the grid scheduler where tasks with their QoS requirements are matched with the available resources and scheduled in order to minimize the total execution time. A task set T can be divided into two task sets T_1 and T_2 such that T_1 contains m_1 tasks with QoS requirements (availability and bandwidth) and T_2 contains m_2 tasks without QoS requirements respectively ($m = m_1 + m_2$). It is assumed that there exist few tasks with QoS requirements in an application. The tasks in task set T_1 gets priority in scheduling over those in T_2 . For every task, the completion time is computed on each suitable resource satisfying its QoS requirements. A resource with minimum completion time is selected for scheduling this task. This ensures that the selected resource satisfy QoS requirements of the task scheduled on it. Availability and bandwidth of a resource are QoS requests which need to be matched while scheduling such tasks in the distributed grid environments.

Algorithm: AQUA

All tasks t_i in T are grouped into sets T_1 and T_2 . Set T_1 consists of tasks with QoS requests while set T_2 consists of tasks without QoS requests.

1. **for** all tasks in set T (in an arbitrary order)
2. **for** all resources (in a fixed arbitrary order)
3. $CT_{ij} = ETC_{ij} + AT(r_j)$
4. **end for**
5. **end for**
6. **do until** there is any task in T_1
7. **for** each task
8. **find** a resource in the QoS qualified resource set that obtains the earliest completion time
9. **end for**
10. **find** a task t_k with minimum earliest completion time (ECT)
11. **assign** task t_k to the resource m_i that gives minimum ECT
12. **delete** task from T_1
13. **update** the available time of resource m_i
14. **update** CT_{ij} for all i
15. **end do**
16. **do until** there is any task in T_2
17. **for** each task
18. **find** a resource in the resource set that obtains the earliest completion time
19. **end for**
20. **find** a task t_k with minimum earliest completion time (ECT)
21. **assign** task t_k to the resource m_i that gives minimum ECT
22. **delete** task from T_2
23. **update** the available time of resource m_i
24. **update** CT_{ij} for all i
25. **end do**

Figure 3. The pseudo code of AQUA scheduling algorithm.

The pseudo code of a newly proposed scheduling algorithm (AQuA) is depicted in figure 3. The algorithm is based on QGMM scheduling approach proposed by He et al. [15]. Initially, a set of potential resources meeting the availability and bandwidth requirements of application tasks is identified from a grid resource pool. A task in a task set is assigned to the resource that minimize the earliest completion time. After assignment of tasks (with QoS requests), the scheduler map the remaining tasks (without QoS requests) over the available resources in order of minimizing the earliest completion time on each resource.

7. SIMULATION AND RESULT ANALYSIS

The algorithm (AQuA) has been validated in the simulated grid environment and the results have been compared with the existing QoS Guided Min-Min (QGMM [15]) algorithm. The experimental results have been depicted in figures 4(a-h) that illustrate the comparative analysis of AQuA and QGMM scheduling algorithms for the performance metrics i.e. reliability of resources for executing applications and makespan of the schedules obtained in simulated grid environments. Different scenarios have been used in simulation testing which affect the performance of task scheduling. Reliability of grid resources allocated for executing the application tasks and makespan of the application schedule have been used as performance metrics in the simulation analysis.

In figures 4(a-b), the comparison is made between AQuA and QGMM in simulated grid environment (application size = 1000 tasks, grid size = 100 nodes) where percentage of dedicated nodes (computational resources) are varied in testing. A node with 100% availability (availability level=1) connected over a network of bandwidth no less than 1 Gbps is considered as dedicated node. The results show that our approach produces better reliability when non-dedicated nodes are higher which is true for grids where the number of non-dedicated nodes is very high. Therefore, AQuA is able to obtain more reliable resources in scheduling while compromising in the makespan.

In figures 4(c-d), grid size has been varied (50-1000 nodes) for scheduling an application (application size = 1000 tasks). The results show that AQuA algorithm produces more reliability than QGMM when the grid size is small without increasing the makespan. Similarly, the results shown in figures 4(e-f) have been obtained for scheduling the applications of different task size over grid resources (grid size = 100 nodes). The results show that our algorithm improves reliability with the increase of application size.

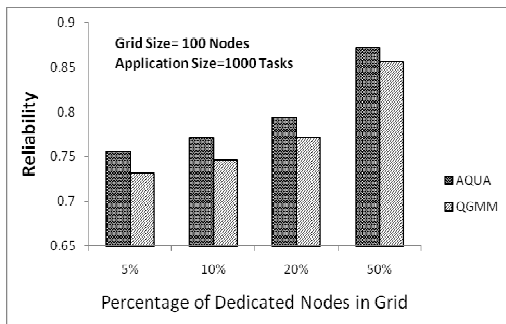
Another experiment is made considering the percentage of tasks in an application having the QoS requirements. Figure 4(g-h) shows the result analysis of this experiment between AQuA and QGMM. The results show that AQuA produces better reliability as number of tasks with QoS requirements increase while sacrificing a little bit in makespan. Thus, the simulated graphs exhibit that the AQuA algorithm generates more reliable schedule than QGMM in highly dynamic grid environment at small cost of makespan which is affordable to lessen the risks associated for executing the applications in grids.

8. CONCLUSIONS

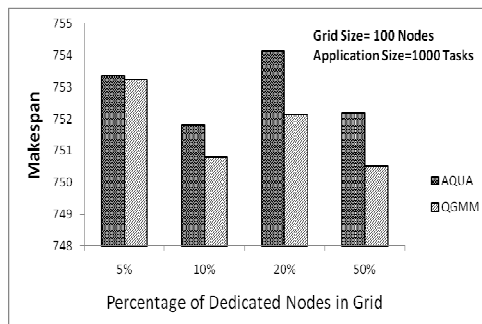
The new challenges have been raised with the evolution of grid computing systems. The objectives need to be revised. To obtain a schedule with minimum makespan is still an objective but the nature of resources in grid is highly dynamic and unreliable. Therefore, selection of highly reliable resources is an important issue while scheduling the tasks. In this paper, we have

proposed an adaptive scheduling algorithm (AQuA) which discovers the resources meeting the QoS requirements of tasks and then schedule the task over the resource which minimizes the overall completion time (makespan).

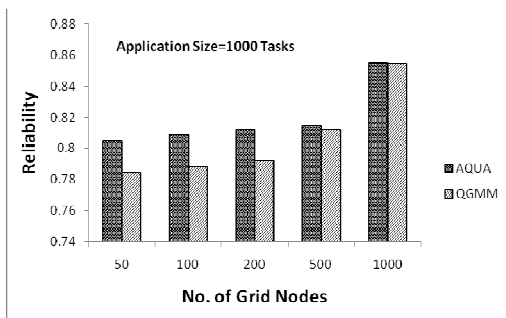
We have addressed two-dimensional QoS issues i.e. availability and bandwidth requirements of tasks. These constraints have major impact on the performance of scheduling applications in grid computing systems. The QoS components are added to the traditional Min-Min heuristic to form the proposed heuristic (AQuA). A simulation system is developed to test the performance of proposed algorithm with the QGMM in the simulated Grid environment. The result analysis shows that AQuA algorithm gives preference to highly available resources in scheduling tasks while meeting the QoS requirements and increases the reliability of grid to successfully execute applications without adversely affecting the makespan.



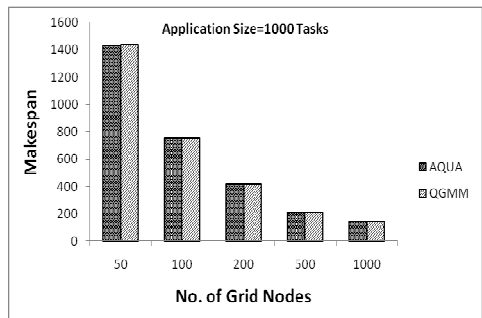
(a)



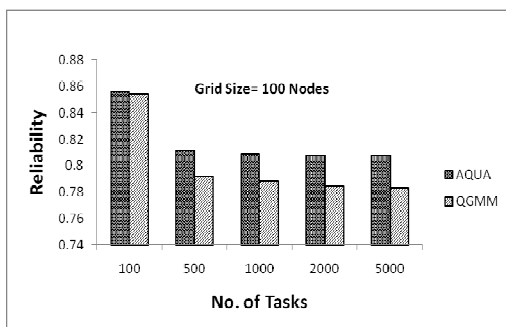
(b)



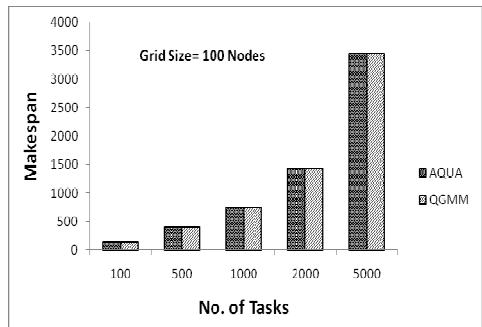
(c)



(d)



(e)



(f)

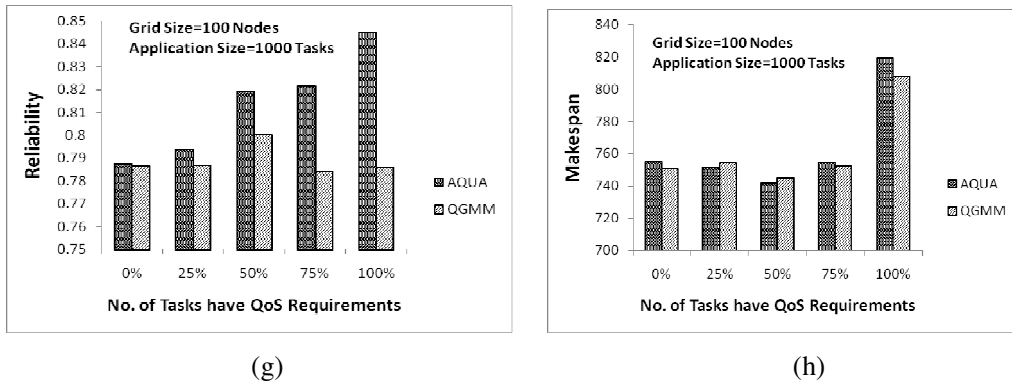


Figure 4 (a-h): Comparative result analysis of AQUA and QGMM in simulated grids

REFERENCES

- [1] I. Foster and C. Kesselman, "The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers", Elsevier Inc., 2004.
- [2] G. Schmidt, "Scheduling with Limited Machine Availability", European J. Operational Research, vol. 121, pp. 1-15, 2000.
- [3] Li Chunlin and Li Layuan, "A Pricing Approach for Grid Resource Scheduling with QoS Guarantees", Fundamenta Informaticae, 76(2007), pp. 59-73, 2007.
- [4] F. Nadeem, R. Prodan, and T. Fahringer, "Characterizing, Modeling and Predicting Dynamic Resource Availability in a Large Scale Multi-Purpose Grid", Proceedings of CCGrid 2008, Lyon, France, 2008.
- [5] Xian-He Sun and Wu Ming, "GHS: A performance prediction and task scheduling system for Grid computing", Proceedings of IEEE International Parallel and Distributed Processing Symposium, Nice, France, 2003.
- [6] L. Gong, X. H. Sun, and E. Waston, "Performance Modeling and Prediction of Non-dedicated Network Computing", IEEE Trans. Computer, 2002, 51(9): 1041-1055.
- [7] I. Foster, A. Roy, and V. Sander, "A Quality of Service Architecture that combines resource reservation and application adaptation", Proc. of the 8th Int. Workshop on Quality of Service, Pittsburgh, PA, USA, pp. 181-188, June 5-7, 2000.
- [8] T. D. Braun, H. J. Siegel, and N. Beck et al, "A Taxonomy for Describing Matching and Scheduling Heuristics for Mixed-Machine Heterogeneous Computing Systems", IEEE Workshop on Advances in Parallel and Distributed Systems, West Lafayette, 1998, pp. 330-335.
- [9] H. Casanova, A. Legrand, D. Zagorodnov, and F. Berman, "Heuristics for scheduling parameter sweep applications in grid environments", Proceedings of the 9th Heterogeneous Computing Work-shop (HCW'2000), Cancun, Mexico, 2000, pp.349-363.
- [10] H. Casanova, G. Obertelii, F. Berman, and R. Wolski, "The AppLeS parameter sweep template: User-level middleware for the grid", Proc. the Super Computing Conference (SC'2000), 2000.
- [11] M. Maheswaran, S. Ali, and H. J. Siegel et al, "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems", Proceedings of 8th IEEE Heterogeneous Computing Workshop (HCW'99), San Juan, Puerto Rico, 1999, pp. 30-44.
- [12] Fernandez-Baca David, "Allocating modules to processors in a distributed system", IEEE Trans. Software Engineering, 15(11), pp. 1427-1436, November, 1989.

- [13] R. F. Freund, M. Gherrity, and S. Ambrosius et al., "Scheduling resources in multi-user, heterogeneous computing environments with SmartNet", Proc. of 7th IEEE Heterogeneous Computing Workshop (HCW '98), Orlando, Florida, USA, 1998, pp.181-199.
- [14] T.D. Braun, H. J. Siegel, and N. Beck, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing, 2001, 61(1): 810-837.
- [15] X. HE, Xian He Sun, and Laszewski Gregorvon, "QoS Guided Min-Min Heuristic for Grid Task Scheduling", J. Comput. Sci. & Technol, 2003, 18(4), pp. 442-451.
- [16] I. Foster, M. Fidler, A. Roy, V. Sander, and L. Winkler, " End-to-end Quality of Service for High-end Applications", Elsevier Comput. Comm. J., 27(14), pp. 1375-1388, 2004.
- [17] A. Dogan and F. Ozguner, "Scheduling Independent Tasks with QoS requirements in Grid Computing with Time-varying Resource Prices", Proc. of GRID 2002, Lecture Notes in Computer Science, vol. 2536, Springer, Berlin, pp. 58-69, 2002.
- [18] K. S. Golconda, F. Ozguner, "A Comparison of Static QoS-based Scheduling Heuristics for a Meta-task with Multiple QoS Dimensions in Heterogeneous Computing", Proc. of the 18th Int. Parallel and Distributed Processing Symposium (IPDPS'04), 2004.
- [19] The Austrian Grid Consortium. <http://www.austriangrid.at>.
- [20] R. Wolski, N. Spring, and J. Hayes, "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing", Future Generation Computing Systems, 1999, 15(5-6):757-768.

Authors



Amit Agarwal born on March 27, 1977 in Roorkee, India. Mr. Agarwal received the B.E. degree (Computer Science and Engineering) from G.B. Pant Engineering College, Pauri in 1999 and the M.Tech. degree in Information Technology from Punjabi University, Patiala in 2005. Presently, he is a PhD scholar in the Department of Electronics and Computer Engineering at the Indian Institute of Technology Roorkee, India. His area of research includes parallel and distributed computing, multiprocessor scheduling and grid computing. He is currently working as an Associate Director & Dean at GRD Institute of Management & Technology, Dehradun (India). He is a member of the IEEE, CSI India, ISoC.



Padam Kumar received the B.Tech and M.Tech degrees in Electrical Engineering from the Indian Institute of Technology, Delhi, India and subsequently, the PhD degree in Computer Science from the University of Roorkee (now, Indian Institute of Technology Roorkee). He has been with the Department of Electronics and Computer Engineering, University of Roorkee since 1974 and is currently working as a professor of Computer Science and Engineering. He is the head of Institute Computer Centre and Information Superhighway Centre of Indian Institute of Technology Roorkee. His current area of interest is computer architectures, multiprocessors, grid computing, fault tolerant computing, and real time multiprocessor systems. He is a member of the IEEE.