ISSN 2047-3338

Optimized Resource Selection to Promote Grid Scheduling Using Hill Climbing Algorithm

N. Krishnamoorthy[1] and R. Asokan[2]

[1]Department of Computer Science and Engineering, Kongu Engineering College, Erode, India
[2]Kongu Nadu College of Engineering and Technology, Thottiyam, Trichy, India

***Abstract*– Grid computing is gaining ground in academia and commerce moving from scientific-based applications to service oriented problem solving environments. Grid is a distributed large-scale computing infrastructure providing dependable, secure, transparent, pervasive, inexpensive, and coordinated resource sharing. Resource selection and use are necessary to enhance application performance. Grid task scheduling is a most important grid system technology being a NP complete problem to schedule tasks on appropriate grid nodes. This study uses hill-climbing heuristics for optimizing the scheduling. Simulations validate the proposed algorithm's performance, and results are evaluated by Makespan. Makespan values of best solutions are recorded throughout optimization iterations and minimum time cost time for all tasks completed.**

***Index Terms*— Grid Computing, Grid Scheduling, Performance and Hill Climbing-Makespan**

I. INTRODUCTION

GRID Computing is an important field visualized as an enhanced form of Distributed Computing. With advent of new technology, it is realized that paralleling sequential applications yielded quicker results at lower cost. GC is next generation IT infrastructure to transform how organizations and individuals communicate, compute and collaborate [1] offering untapped processing cycles from computer networks spanning geographical boundaries.

Some grid characteristics are [2]:

- Large scale: a grid must deal with many resources ranging from a few to millions.
- Geographical distribution: grid's resources location should be at distant places.
- Heterogeneity: grids host software/hardware resources which vary from data, files, software components or programs to scientific instruments, sensors, personal digital organizers, display devices, computers, super-computers and networks.
- Resource sharing: grid resources belong to different organizations permitting other organizations (users) to access them.
- Multiple administrations: each organization may have varied security and administrative policies to access and use their resources.
- Resource coordination: grid resources must be coordinated to provide aggregated computing capabilities.
- Transparent access: a grid should be a single virtual computer.

Grid architecture is layered with the lowest layer (Fabric Layer) implementing low level services and upper most layer (Application Layer) implementing high level services [3]. Low level services are those concerning hardware and complex systems implementation while high level services are linked to user interface and application implementation services. This makes Grid computing attractive to application areas needing different levels of services to accomplish tasks.

Each layer shares underlying component layers behavior. The fabric layer defines interface to local resources, which are shared. Connectivity layer defines basic communication/authentication protocols needed for grid-specific networking services. Resource layer uses communication/ security protocols (defined by connectivity layer) to control initiation, negotiation, monitoring, and payment for function sharing of individual resources [4].

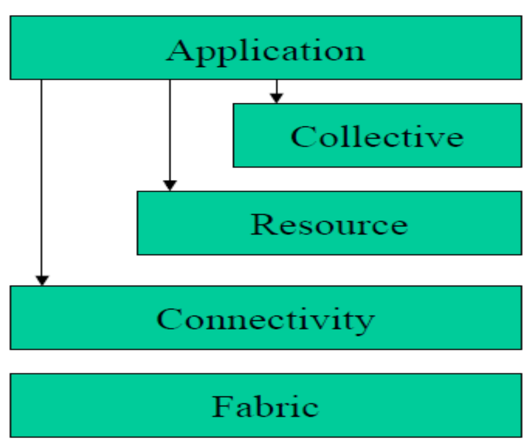

Fig. 1: Layered Grid Architecture

Collective layer, also called protocol layer implements various sharing behaviors using a limited resource and connectivity-layers protocols. Application layer enables resources use in grid environments through collaboration and resource access protocols.

Grid scheduling is mapping of individual tasks to computer resources, respecting service level agreements, etc. Grid schedule systems operate in dynamic environments subject to unforeseen and unplanned events that happen suddenly [5] including computers breakdown, new jobs processing time arrivals which are subject to stochastic variations. It turns out that schedule performance is sensitive to such disturbances. A predictive schedule generated in advance is hard to execute.

An important factor in multi-grid system performance is choice of grid for task execution. While many grid CPUs are critical to obtain high performance for higher complexity massively-parallel tasks, shorter tasks response time depends on the chosen grid [6] execution overheads. The scheduling algorithm's goal is finding proper execution hierarchy level for a complex task with minimum overhead.

A tasks schedule (or schedule) is task assignment to specific time intervals of resources so that no two are on any resource simultaneously, or so that resource capacity is not exceeded by tasks [7]. Task schedule is optimal if it minimizes an optimality criterion (function). Jobs are scheduled to parts of machines (CPUs and memory). Many jobs are scheduled on one machine at any time if resource capacities are not exceeded.

A Grid scheduler/resource broker is an interface between user and distributed resources hiding Grid computing [8] complexities. It discovers resource, negotiates access costs using trading services, maps jobs to resources (scheduling), stages application and data for processing (deployment), starts job execution and gathers results. It monitors and tracks application execution progress with adapting to changes in Grid runtime environment, resource share availability variation and failures.

Scheduler structure is based on resources managed and their location domain. Usually, 3 structure scheduler's models are distinguished: Centralized, Decentralized and Hierarchical [9]. The first manages single/multiple resources located in single/multiple domains and supports uniform scheduling thereby suiting cluster management (or batch queuing) systems. The Decentralized model fits Grid environments better as schedulers interact among themselves to decide resource allocation for job execution. The Hierarchical model fits Grid environments as it permits remote resource owners to enforce own policy on external users by removing single centralization points.

The independent tasks scheduling problem is NP-hard consisting of N tasks and M machines. Each task is processed by each M machine as Makespan is minimized. Task scheduling is part of parallel/distributed computing [10]. Grid scheduling is responsible for resource discovery, selection, assignment and aggregation of resources over decentralized heterogeneous systems; resources belong to many administrative domains.

Grid systems scheduling process for optimization tries to provide better solutions for selection/allocation of resources to tasks. Scheduling optimization is important as scheduling is the main building block to make Grids available more to users [11]. Grid scheduling optimization methods are the subject of this study. Scheduling problem is NP-Complete. So, approximation algorithms which quickly offer solutions, even if only near-to-optimal, are considered.

In grid environment, scheduling is a very complicated. This problem aims how to improve resources efficiency and minimize completion time simultaneously. Heuristic optimization algorithms are used to solve various NP-complete problems [12]. Meta-heuristic techniques are applied to handle NP-complete problems. The rise in solution search space size motivated researchers to use nature-inspired meta-heuristics mechanisms to solve computational grid scheduling problems [13]. Nature-inspired meta-heuristics proved an excellent effectiveness and efficiency in handling combinatorial optimization problems.

Firefly algorithm (FA) is a meta-heuristic algorithm, inspired by fireflies flashing behavior. It is a population-based technique to locate a global optimal solution based on swarm intelligence, investigating fireflies foraging behavior. Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), and heuristics are derived by combining three algorithms [14].

GA and SA are powerful stochastic optimization methods, inspired by nature. Heuristic Task Scheduling Algorithm in Grid computing environment is based on tasks predictive execution time. It gets a scheduling strategy by using mean load as heuristic information and selects maximum-load and minimum-load machines.

The ant algorithm is based on a heuristic approach. It is based on ant's real behavior. Each ant deposes chemical pheromone on a path when searching for food from its nest. When an ant moves in a specific direction, pheromone strength increases. Other ants also trail along with this. This inspired the ACO algorithm which is population based, is applied to many NP-hard optimization problems [15]. A population based search algorithm Particle Swarm Optimization (PSO) has particles flying through multidimensional search space where each particle's position is adjusted based on its experience and that of its neighbors.

This study presents a local, anytime algorithm that computes hill-climbing heuristic for grid computing. This algorithm is easily applied to any hill-climbing problem. The remainder of the study is organized as follows; section 2 discusses previous grid computing works. The 3 and 4 sections describe hill climbing method and results respectively. Finally, section 5 concludes the study.

II. RELATED WORKS

A scheduling algorithm with following features to be used in a grid computing environment was proposed by In, et al.,

[16]. The algorithm first supports resource usage constrained scheduling. Next, it performs optimization-based scheduling. Third, it assumes that a resources set was distributed geographically and is heterogeneous. Fourth, scheduling method dynamically adjusts to grid status. It tracks current resources workload. The proposed algorithm's performance was evaluated by predefined metrics. Also, a set of experiment was performed on open science grid to show simulation results for policy-based scheduling out-performance.

An overview of important grid workloads characteristics in the past seven years (2003-2010) was presented by Iosup and Epema [17]. Tough grid user populations range from tens to hundreds of individuals, some users dominate a grid's workload regarding resources consumed and jobs submitted to a system. Real grid workloads include few parallel jobs, but many independent single-machine jobs (tasks) are grouped as single "bags of tasks."

Integrated Processing of Data and Text (IPDT)-FUZZY scheduler, that considers grid applications demands with uncertainties was introduced by Batista and da Fonseca [18]. The scheduler used fuzzy optimization with computational and communication demands being expressed as fuzzy numbers. Performance evaluation revealed it to be attractive when communication requirements were uncertain. Simulation compared its efficacy to a deterministic counterpart scheduler and results reinforced its adequacy to deal with lack of accuracy in communication demands estimation.

A Quality of Experience (QoE)-driven power scheduling in a smart grid context from perspectives of architecture, strategy, and methodology was proposed by Zhou, et al., [19]. It considers QoE requirement when designing power allocation schemes. To obtain QoE requirements, power load fluctuation and transmission delay impact were analyzed. Specifically power allocation was formulated as an optimization problem maximizing the system's social welfare. Based on QoE model, an efficient power scheduling scheme was proposed by considering admission control and QoE expectations. Extensive simulation indicated the proposed scheme could efficiently allocate power based on dynamic QoE requirements in smart grid systems.

Formulation of 2 optimization problems with objectives being minimization of job completion time and minimization of resource usage/cost to satisfy deadline jobs was presented by Liu, et al., [20]. The authors proposed heuristics to deal with applications with optimization objective demonstrating good simulation performances.

A deadline aware algorithm to solve ensuring end-to-end QoS and improving grid resources efficiency was presented by Rashida, et al., [21]. It investigated group requests at start of time slot trying to accept the highest number. Simulations proved that the deadline aware algorithm improved advance reservation resources efficiency in both cases.

A heuristic method, called Gravitational Emulation Local Search (GELS) algorithm to solve scheduling and advance resources was presented by Barzegar, et al., [22]. The algorithm was called Gravitational Emulation Local Search Advanced Reservation Algorithm (GELSAR) and compared with GA to confirm it. Results show that accepted jobs which used GELSAR increased 7.5 percent and computation time was reduced by 50 percent compared to GA.

A de-centralized job scheduling by job migrations between neighboring grid nodes was implemented by Wang, et al., [23].

To optimize a new-submitted job's node selection the job can migrate many times. Here, hill climbing determined migration route. Experiments simulated de-centralized job scheduling, including node adjacencies, grid nodes local scheduling and grid workload. Compared to k-distributed and auction methods, hill climbing-based scheduling enhanced processor use and reduced bounded slowdown.

A tasks scheduling algorithm for grid computing was introduced Fidanova [24]. Based on simulated annealing, the proposed method showed how to search for best tasks when scheduling for grid computing.

A new resource characteristic based optimization method capable of being combined with Earlier Gap, Earliest Deadline First (EG-EDF) policy to schedule jobs in a dynamic environment was proposed by Aggarwal, et al., [25]. The new algorithm generated near-optimal solutions, better than those in literature for a specific datasets range. Extensive experimentation and analysis proved the proposed method's efficacy.

The use of epsilon dominance based MOEA approach to solve workflow scheduling problems in Grid was proposed by Navaz and Ansari [26]. In one scheduling problem, two major conflicting objectives called makespan and cost are addressed. The authors consider 3 conflicting objectives like execution time (makespan), total cost and reliability and suggested an evolutionary computing paradigm based multi-objective scheduling algorithm using R-NSGA-II approach. Simulation showed the proposed algorithm generating multiple scheduling solutions near Pareto optimal front with reduced computation overhead.

A new trustworthiness-based scheduling model for grid workflows scheduling was proposed by Tao, et al., [27]. This is a multi-objective optimization problem generally while an integer genetic algorithm (IGA) optimizes scheduling performance. Simulation showed that IGA performance was promising in a robust multi-criteria scheduling algorithm. Scheduling succeeds in open and dynamic grid environments, and trustworthiness and robustness was its result in e-Business Application.

A new stochastic model for grid scheduling and a new evolutionary scheduling algorithm based on this model was presented by Shi and Zhao [28]. Also, optimization methods improved grid QoS. Conventional grid scheduling algorithms use deterministic models, but real world grid environments are subject to uncertainty or randomness, like network status and job execution costs which are unknown precisely in advance. A good scheduling problem model should address these. Simulated experiments revealed the feasibility of the proposed scheduling algorithm.

A new approach to schedule jobs in a grid using Bee Colony algorithm, and discovered resources being reserved in advance for the future was presented by Kaladevi, et al., [29]. Bee Colony is a recent heuristic algorithm used for optimization. Grid has many clusters with resources. Clusters are formed with K-means algorithm, a partitioning method for cluster formation. Results show that resource discovery performance greatly improved through this algorithm and advance resources reservation increased total throughput.

New approaches combining fault tolerance techniques with existing workflow scheduling algorithms was proposed by Zhang, et al., [30]. A study on effectiveness of combined approaches was proposed by analyzing impact on the reliability of workflow execution, workflow performance and resource usage under varied reliability models, failure prediction accuracies and workflow application types.

A new decentralized scheduling algorithm for P2P grid systems was proposed by Chauhan and Nitin [31] where an independent task chose a suitable grid node based on immediate neighbors local information. A vital feature of the proposed method is its scheduling computation and communication intensive tasks to balance grid system's workload.

III. METHODOLOGY

Hill Climbing Algorithm

Hill-climbing is an effective heuristic search technique which assumes a reasonable local optimum can be approached if at every search step it greedily chooses directions that decrease cost most. The method's strength is its simplicity [32], and it was extensively tried for optimum search in exponential domains in problems like GA and clustering.

Hill Climbing is iterative. A current solution determines acceptance of new candidate at each iteration. Hill climbing algorithm starts with a random schedule sequentially making minor changes to a schedule, improving it a little every time. At some point, algorithm reaches where it sees no improvement and so terminates. Then ideally, a schedule close to optimal is found but does not guarantee that hill climbing will come close to optimal solution [33]. Hill climbing optimization has 4 input parameters, which are objective function, starting points, range and search step.

Procedure for Hill climbing

1. Initialization

Set - step k = 0; initial schedule S_0 (randomly generated); best schedule Sbest = S_0, bestCost = $F(S_0)$

2. Select next solution from allowed transitions set for which best current solution is reached and set $S_{best} = S_{k+1}$ and bestCost = $F(S_{k+1})$; continue with step 3; if there is no such transition, terminate.

3. Termination

if conditions for termination are fulfilled, terminate, else set k = k +1 and go to step 2.

Weighted Round Robin

Weighted Round Robin (WRR) scheduling discipline was implemented in various mission critical computing and communication systems. Tasks are performed cyclically in a WRR scheduler at which time a task can execute in each round is proportional to weight assigned to it. WRR schedulers have 2 major advantages:

- Ability to improve system robustness, in a minimum service rate guarantee for each task. For weighted round robin schedulers, maximum service every task receives in each round is allocation upper-bounded.
- Ability for distributed implementation. A WRR scheduler is realized easily in a distributed environment by a logic ring like timed token protocol [34].

IV. EXPERIMENTAL RESULTS

Simulations are done with various grid nodes/jobs cluster numbers. Two different scheduling algorithms are implemented with weighted round robin algorithm and Fish Swarm Optimization algorithm. Makespan parameter and standard deviations are used in results comparison.

It is observed from Fig. 2 that the Makespan of the proposed Hill climbing decreases when compared to the WRR. The proposed hill climbing reduces the Makespan by 4.97%.

V. CONCLUSION

Grid computing is becoming popular daily due to the emergence of Internet as a ubiquitous media and the availability of powerful computers and networks as low-cost commodity components. The Grid provides ability to access, use and manage various virtual organizations heterogeneous resources across multiple domains and institutions where requests are served from external users with local users. Scheduling was a key challenge and widely studied in enabling computational

Table 1: Experimental Results

Number of clusters / Number of jobs	16 /120	24/180	32/240	40/300
Make span	97.47	151.3	197.5	254.7
Make span – FSO	97.47	151.3	197.52	254.66
Make span- Weighted round robin	100.82	155.2	207.3	261.9

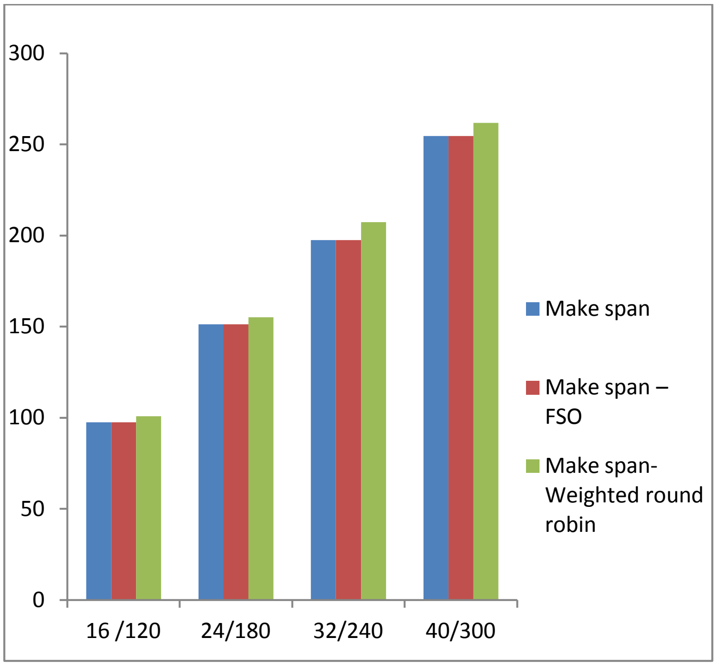

Fig. 2: Comparative Analysis

grid systems over ten years. Task scheduling algorithms improve grid performance by reducing scheduling length. Simulations were undertaken with varying resource clusters and sufficient jobs. The result is evaluated by Makespan with many resources and jobs.

REFERENCES

[1]. Bansal, B., & Bawa, S. (2005, November). Design and Development of Grid Portals. In TENCON 2005 2005 IEEE Region 10 (pp. 1-5). IEEE.

[2]. Bote-Lorenzo, M. L., Dimitriadis, Y. A., & Gómez-Sánchez, E. (2004, January). Grid characteristics and uses: a grid definition. In Grid Computing (pp. 291-298). Springer Berlin Heidelberg

[3]. Goteng, G., Tiwari, A., & Roy, R. (2007). Grid computing for engineering design optimisation: Evolution and future trends.

[4]. Joseph, J., Ernest, M., & Fellenstein, C. (2004). Evolution of grid computing architecture and grid adoption models. IBM Systems Journal, 43(4), 624-645.

[5]. Andrieux, A., Berry, D., Garibaldi, J., Jarvis, S., MacLaren, J., Ouelhadj, D., & Snelling, D. (2003). Open issues in grid scheduling. UK e-Science Report UKeS-2004-03, April 2004.

[6]. Silberstein, M., Geiger, D., Schuster, A., & Livny, M. (2006, June). Scheduling mixed workloads in multi-grids: the grid execution hierarchy. In High Performance Distributed Computing, 2006 15th IEEE International Symposium on (pp. 291-302). IEEE.

[7]. Fibich, P., Matyska, L., & Rudová, H. (2005). Model of grid scheduling problem. Exploring Planning and Scheduling for Web Services, Grid and Autonomic Computing, 17-24.

[8]. Buyya, R., & Murshed, M. (2002). A deadline and budget constrained cost-time optimisation algorithm for scheduling task farming applications on global grids. ArXiv preprint cs/0203020.

[9]. Pop, F., & Cristea, V. (2009). Optimization of Schedulig Process in Grid Environments. UPB Sci. Bull., Series C.

[10]. Radha & Dr. V. Sumathy. (2013). A Hybrid Genetic Algorithm with Elitist Ant System in Grid Scheduling.

[11]. Murugesan, G., & Chellappan, C. (2010). An economic-based resource management and scheduling for grid computing applications. ArXiv preprint arXiv: 1004.3566.

[12]. Zhang, L., Chen, Y., Sun, R., Jing, S., & Yang, B. (2008). A task scheduling algorithm based on pso for grid computing. International Journal of Computational Intelligence Research, 4(1), 37-43.

[13]. Yousif, A., Abdullah, A. H., Nor, S. M., & Abdelaziz, A. A. (2011). Scheduling Jobs on Grid Computing using Firefly Algorithm. Journal of Theoretical and Applied Information Technology, 33(2), 155-164.

[14]. Maruthanayagam, D., & Rani, R. U. Improved Ant Colony Optimization for Grid Scheduling.

[15]. Selvakrishnan, S., & Perumal, V. Optimization Of Grid Resource Scheduling Using Particle Swarm Optimization Algorithm

[16]. In, J. U., Lee, S., Rho, S., & Park, J. H. (2011). Policy-based scheduling and resource allocation for multimedia communication on grid Computing environment. Systems Journal, IEEE, 5(4), 451-459.

[17]. Iosup, A., & Epema, D. (2011). Grid computing workloads. Internet Computing, IEEE, 15(2), 19-26.

[18]. Batista, D. M., & da Fonseca, N. L. (2011). Scheduling Grid Tasks in Face of Uncertain Communication Demands. Network and Service Management, IEEE Transactions on, 8(2), 92-103.

[19]. Zhou, L., Rodrigues, J. J., & Oliveira, L. M. (2012). QoE-driven power scheduling in smart grid: Architecture, strategy, and methodology. Communications Magazine, IEEE, 50(5), 136-141.

[20]. Liu, X., Qiao, C., Wei, W., Yu, X., Wang, T., Hu, W., & Wu, M. Y. (2009). Task scheduling and lightpath establishment in optical grids. Journal of Light wave Technology, 27(12), 1796-1805.

[21]. Rashida, S. Y., TeshnehLab, M., & Rahmani, A. M. (2010, October). An efficient negotiation based algorithm for resources advanced reservation using hill climbing in grid computing system. In Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on (pp. 71-77). IEEE.

[22]. Barzegar, B., Rahmani, A. M., Zamanifar, K., & Divsalar, A. (2009, November). Gravitational Emulation Local Search Algorithm for Advanced Reservation and Scheduling in Grid Computing Systems. In Computer Sciences and Convergence Information Technology, 2009. ICCIT'09. Fourth International Conference on (pp. 1240-1245). IEEE.

[23]. Wang, Q., Gao, Y., & Liu, P. (2006, June). Hill climbing-based decentralized job scheduling on computational grids. In Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on (Vol. 1, pp. 705-708). IEEE.

[24]. Fidanova, S. (2006, October). Simulated annealing for grid scheduling problem. In Modern Computing, 2006. JVA'06. IEEE John Vincent Atanasoff 2006 International Symposium on (pp. 41-45). IEEE.

[25]. Aggarwal, A., Du, P., & Robert, D. (2010). Grid Scheduling Optimization Based on Resource Characteristics. Journal of Computational Information Systems, 6(14), 4609-4616.

[26]. Navaz, S., & Ansari, U. An Evolutionary Algorithm in Grid Scheduling by multi-objective Optimization using variants of NSGA.

[27]. Tao, Q., Chang, H., Yi, Y., & Gu, C. (2010, May). A grid workflow scheduling optimization approach for e-Business

application. In E-Business and E-Government (ICEE), 2010 International Conference on (pp. 168-171). IEEE.

[28]. Shi, X., & Zhao, Y. (2010, August). Stochastic model and evolutionary optimization algorithm for grid scheduling. In Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on (Vol. 1, pp. 424-428). IEEE.

[29]. Kaladevi, A. C., Srinath, M. V., & Prabhakar, A. (2013, January). Reserved Bee Colony optimization based grid scheduling. In Computer Communication and Informatics (ICCCI), 2013 International Conference on (pp. 1-6). IEEE.

[30]. Zhang, Y., Mandal, A., Koelbel, C., & Cooper, K. (2009, May). Combined fault tolerance and scheduling techniques for workflow applications on computational grids. In Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on (pp. 244-251). IEEE.

[31]. Chauhan, P., & Nitin, N. (2012, March). Decentralized Computation and Communication Intensive Task Scheduling Algorithm for P2P Grid. In Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on (pp. 516-521). IEEE.

[32]. Krivitski, D., Schuster, A., & Wolff, R. (2007). A local facility location algorithm for large-scale distributed systems. Journal of Grid Computing, 5(4), 361-378.

[33]. Martincová, P., & Zábovský, M. (2007). Comparison of simulated Grid scheduling algorithms. Systemova Integrace, 4, 69-75.

[34]. Wu, J., Liu, J. C., & Zhao, W. Schedulability Bound of Weighted Round Robin Schedulers for Hard Real-Time Systems.

Mr. N. Krishnamoorthy received his B.E degree in Computer Science and Engineering from Periyar University and M.E degree in Computer Science and Engineering from Anna University Chennai. Currently he is pursuing his Ph.D in Information Communication Engineering from Anna University Chennai. He has published more than 12 papers in the National Conferences, 3 papers in the International Conferences and 3 papers in the International Journals. He has 11 years of teaching experience. He is member of various scientific and professional societies. At present, he is working as Assistant Professor (Senior Grade), in Kongu Engineering College, Erode India. His areas of interest include Grid Computing and Operating Systems.

Dr. Asokan Ramsamy received his B.E. degree in Electronics and Communication from Bharathiar University and M.S. degree in Electronics and Control from Birla Institute of Technology. He obtained M.Tech. Degree in Electronics and Communication from Pondicherry Engineering College with distinction. He obtained PhD in Information and Communication Engineering from Anna University Chennai. He has published more than 80 papers in National and International journals and conferences. He has 25 years of teaching experience. He is member of various scientific and professional societies. At present, he is working as Principal, Kongunadu College of Engineering and Technology, Thottiyam, India. His areas of interest include wireless networks, network security and image processing.