A Review on Resource Scheduling Models to Optimize Quality of Service Parameters in Grid Computing using Meta-heuristics

Dinesh Prasad Sahu
SC&SS JNU
New Delhi, India 110067

Karan Singh
SC&SS JNU
New Delhi, India 110067

Shiv Prakash
Department of Chemical Engineering
IIT Delhi, India 110 016

ABSTRACT
Computational Grid (CG) is a wide network of computational resources that provides a distributed platform for high end compute intensive applications. The resources in the computational grid are usually heterogeneous and being a highly heterogeneous system, Computational Grid poses a number of constraints. It is difficult to allocate and schedule the applications properly to achieve the benefit of the grid resources from the applications point of view, as the resources are heterogeneous and dynamic in nature. There are no common scheduling strategies that fulfill all the needs with respect to both, user and the system. The available scheduling implementations consider specific characteristics of the available resources and the application. The complexity of application, user requirements and system heterogeneity prevents any scheduling procedure in achieving its best performance. The aim of a grid scheduling algorithm is to find an appropriate set of resources and maintain its user-demanded Quality of Service (QoS) requirements. Scheduling in CG is an NP-hard problem which requires an efficient solution. The problem, considered in this work, is task scheduling in Computational Grid (CG). Task scheduling in CG is a complex problem as many QoS parameters and system constraints are involved. This paper deliberates over the problem and various tools used in order to solve this problem.

General Terms
Scheduling, Computational Grid, QoS Parameters, Meta-heuristics, Computational Grid

Keywords
Scheduling, Computational Grid, QoS Parameters, Meta-heuristics

1. INTRODUCTION
Parallel systems were designed to execute a job in parallel whereas distributed systems came into existence for exploiting the resources in a better and distributed manner. Eventually, with the introduction of high speed networks, both (parallel and distributed system) began to be used for parallel job execution which was not possible in a uniprocessor system. Consequently, parallel and distributed system evolved. The objective of the parallel system [1, 2], is to execute the submitted jobs/tasks as quickly as possible. Distributed systems [1, 2], allow resource sharing and cooperative engineering to achieve parallel execution with the help of multiple computing nodes. High speed network revolution, during the last decade, was the driving force for the important evolutions in the design of distributed system.

While solving problems, many a times it might not be possible to provide the required computing resources e.g. processors, storage and memory individually to various jobs. It is difficult and costly to afford these resources. A possible and efficient solution, to make these resources available, is to integrate such resources that are distributed at various places in to a single virtual platform so that these can be accessed in a well-defined and transparent manner. This concept is adopted in Grid [3]. One of the largest communities which use technologies of the grid is European Organization for Nuclear Research (CERN) [4]. Future scientific research is to be carried out with larger collaboration of researchers across the globe. Also, size and complexity of research problems are growing with the advancement in research and technology.

When the computing power of the resources is accumulated, it results in Computational Grid (CG) while in data grid the resources are mainly in form of data. Research is not only solution oriented but more focused on efficient solutions and optimizing existing ones. In such scenario, Computational Grids emerged as global cyber infrastructure for sharing computational power to manage and process the big jobs [3, 5]. In this paper, focus is on CG.

Computational Grid (CG) is a wide network of computational resources that provides a distributed platform for high end compute intensive applications. The resources in the computational grid are usually heterogeneous and being a highly heterogeneous system, it poses number of constraints [3, 5, 6]. It is difficult to allocate and schedule the applications properly to achieve the benefit of the grid resources from the application's point of view, as the resources are heterogeneous and dynamic in nature [6]. There are no common scheduling strategies that fulfill all the needs with respect to both; the user and the system. The available scheduling implementations consider specific characteristics of the available resources and the application. The complexity of application, user requirements and system heterogeneity prevents any scheduling procedure in achieving its best performance. The aim of a grid scheduling algorithm is to find an appropriate set of resources and maintain its user-demanded Quality of Service (QoS) requirements. Scheduling in CG is an NP-hard [7, 8, 9] problem which requires an efficient solution.

2. THE PROBLEM
The problem, considered in this work, is scheduling the job/task on Computational Grid (CG) resources. Task scheduling in CG is a complex problem as many QoS parameters and system constraints are involved [10, 11]. This section deliberates over the problem and various tools used in order to solve this problem.

Computational Grid [3, 5, 6], primarily used for processing of compute intensive tasks, has emerged as a global next generation computing infrastructure. Research communities are utilizing CG to share, manage and process the large computational tasks. CG [3, 5, 6] is a collection of

hardware/software infrastructure which facilitates pervasive, consistent, dependable, and economical access to high end computational power in spite of geographical distribution of both the resources and the users. CG users may demand compute intense task execution. CG [5, 6] middleware explores the relevant resources from the pool of grid resources and based on task requirements and execution policies, schedules the task on suitable computing resources.

Grid scheduling defines how the jobs are assigned to run on suitable computing nodes in a manner that aims to optimize some scheduling parameters. The scheduling parameters [8, 9] may be system utilization, throughput, turnaround time, fairness, waiting time, response time etc. The scheduling parameters, mostly, are QoS parameters. Because of heterogeneous nature, scheduling whatever is its type (low level, middle level and user level) in general is NP-hard [6, 7, 8].

Some of the QoS parameters that have been addressed in this work are MS, load balancing, availability and energy as detailed in the next section.

3. ADDRESSED QoS PARAMETERS

The QoS parameters, discussed in this paper, are as follows.

3.1. Make-span (MS):

For a schedule consisting of jobs, MS is defined as the time required in finishing the latest job [6, 8].

3.2. Load Balancing:

Load balancing is a process of distributing the tasks in a fair manner within the grid nodes [13, 14]. Load balancing completely depends upon the scheduling algorithm [13]. Better scheduling algorithm results in optimum load balancing eventually leading to improved grid performance [14]. If a node is heavily loaded, some computation needs to be migrated to other suitable nodes. Load balancing can be static or dynamic.

3.3. Availability:

Availability is defined as a percentage of time resource/sites are available [15] for job processing. It can also be described as fraction of time for which a resource is available for the application. Availability of the system is the percentage of time the system is operational [15]. It is measured as a factor of reliability, so when the reliability of the system increases, availability also increases.

3.4. Energy:

To maintain and provide high computing power, energy is a major constraint. Although, present available devices have the crunch of both the computational and electrical energy, it is believed that in near future with advancement in technology these devices will have better computational power within same amount of electrical energy [17]. High computing device consumes more energy. Thus, there is a tradeoff between the computational power and energy. To save energy, the design of the software systems should be energy efficient and should be tailored accordingly [16].

4. RELATED WORK

The paper proposes few scheduling models for Computational grid with the objective to optimize QoS parameters as mentioned above. Paper one is an introductory paper and in paper two the addressed problem has been defined. The first proposed work appears in paper 3 of the paper. It is a scheduling model using GA that studies the effect of IPC [9] in the overall objective of the scheduling. Three cases have been considered for observing the effect of IPC on the (MS). These are; job allocation without considering the IPC, with variable IPC and with constant IPC. It is observed that MS increases when IPC is introduced amongst the tasks. With variable communication (IPC) there is little increase in MS, whereas increase is substantive with constant communication. It is to note that in constant IPC, maximum of variable IPC has been taken. The conclusion drawn is that unless there is an execution of a dedicated system in which usually there is constant IPC, MS is not affected much. It is a good sign for the parallel execution in CG. Another work, in paper three, deals with the load balancing [13, 14] aspect in CG scheduling with focus on load variation and load distribution amongst the nodes. Three cases have been considered to observe the load variation and load distribution. First two are with fixed load and varying nodes for both fine grain and coarse grain task. Last experiment deals with the coarse grain tasks having fixed load and varying nodes. It is concluded that when the number of nodes are increased, for fixed load, the distribution is better. In comparison to coarse grain tasks, load distribution is better for fine grain tasks for fixed number of nodes. It is because, fine grain tasks gives more possibility of load distribution. Overall, proposed model attains better load distribution for both fine grain as well as coarse grain tasks. The load distribution schemes can be incorporated with any scheduling algorithms to deduce better load balancing and therefore better system utilization.

Paper four proposes two scheduling models to optimize QoS parameters in Computational Grid. First work deals with the proposal of a scheduling in CG with emphasis on the availability as a QoS parameter. This model uses GA [12] for the purpose and is called AGA model. Availability [15] is an important QoS parameter in CG to be optimized and has been given consideration while scheduling a job. It uses a meta-heuristic technique GA for this purpose. Seven cases have been considered to observe node availability in CG during execution. In the first case, it is concluded that if MTTF increases, availability increases [16]. Second case concludes that with the increase in MTTR, availability decreases. Third case exhibits that the increase in task size leads to the decrease in availability. Observation in the fourth case is when the number of tasks increases, availability decreases. Fifth case says that when the number of tasks increases, availability decreases. Sixth case depicts that when number of nodes in CG environment increases, availability increases. Conclusion of the seventh case is that with the increase in processing speed of the nodes, availability increases and with the decrease in load on the node, availability increases. Finally, a comparative study of the proposed AGA based model with simple GA based model for MS and availability epitomes that if availability is maximized, MS increases though increase is not much substantial.

Another work deals with the proposal of a scheduling model using a GA variant, Quantum GA or QGA, with focus on MS as QoS parameter. QGA has recently been developed for solving such problems and has been quite effective too. QGA performs better than GA as it explores in 0-1 hyperspace whereas GA explores directly in search space [17, 18, 19]. A novel method of task scheduling using QGA has been proposed and experimental study reveals that the performance of QGA based model is better than GA based model. MS minimization is highly impressive in QGA. It is because in QGA [17, 18], *m* Q-bit representation has better population diversity. Further, the solution converges quickly in

comparison to GA based methods. The model performs well even on scaled data showing that the method is robust, rigorous and scalable. Thus, the method performs better for high scale of grid and tasks.

In paper five two hybrid approaches towards the grid scheduling problem are described. First work deals with the proposal of a grid scheduling algorithm optimizing MS using a hybrid GABFO technique [20, 21]. GABFO has the advantages of both the GA [12, 22] and BFO algorithms [20] in terms of its ability to find feasible solutions, avoid premature convergence and the ability to conduct fine-tuning in the search space. Meanwhile, heuristics are embedded into the GA as a local search to improve the search ability. GA operators do not have decision capability and is based on random search. BFO works like an intelligent system which can tackle the system in an efficient manner. BFO has the decision capability but it cannot navigate through the search space in an efficient manner. Therefore, both approaches are combined together for effective and efficient results. Performance of the proposed GABFO based model has been studied by carrying out number of experiments and it is found that it performs very well. Also, its comparative study with another GA based model shows that it has an edge over GA based model. The effectiveness of the model is also studied with scaled input and it is found that proposed GABFO based model performs well.

Another work, presented in paper five, proposes a grid scheduling algorithm applying another hybrid approach; Immune based Genetic Algorithm (IGA) to optimize MS [22]. Six cases have been considered for observing the value of the MS. In first case, it is concluded that if number of tasks increases, MS increases. In second case, if number of computing nodes increases, MS decreases. In third case, if task size increases, MS increases. In fourth case, if processing speed of the node increases, MS decreases and when load in the node decreases, MS decreases. In each case, comparative study of IGA based model with GA and GABFO based model is done where it has been observed that the MS produced by IGA is much better than produced by GA and GABFO based models. The performance of the proposed model has also been observed for large set of input jobs and large number of computing nodes in the grid. Results indicate that the model performs better for larger data sets. The study concludes that MS is minimized faster in IGA based model than GA and GABFO based models.

Energy utilization is a major concern and work in paper six deals with the proposal of a scheduling algorithm in CG with emphasis on the energy minimization using a GA technique. Energy consumption, MS and utilization are major issues in this model. Performance study of the proposed model has been done by carrying out experiments with three grid sizes; small, medium and large. Result reveals that the proposed model performs well. Also, the comparative study reveals that the proposed model has an edge over other contemporary models viz. Min-Min, Max-Min, HEFT and EAMM for energy optimization [17, 23, 24, 25, 33].

5. CONCLUSION AND FUTURE SCOPE

The work in this paper proposes few grid scheduling models considering the QoS parameters from the users as well as systems point of view. For this, a meta-heuristic GA, its variant and few hybrid GA techniques have been applied. GA, a search technique based on the evolutionary computation, is found to be quite efficient for solving a class of complex optimization problems. GA has the potential to solve scheduling problem of computational grid and therefore GA and its variant have been considered in most of the proposed work. The paper is organized in seven papers.

The future work will consider the resource scheduling problem with consideration on some other QoS parameters such as reliability, perform-ability and security [23] etc. Also, in all the proposed work, at a time only one parameter has been considered. Sometime, there is a need to consider more than one parameter while scheduling a job on grid. It is possible to work on multi-objective [24] optimization and to handle more than one QoS parameter simultaneously to solve the grid scheduling problem. Also there are many other appealing meta-heuristic techniques that can be explored with various QoS parameters and will be taken as future work. Resources scheduling models may be applied in various other new areas like Cloud computing, Cluster computing etc. Resource scheduling models may also be applied for real life scheduling problems like project portfolio management. It is possible to apply the developed models on newer areas where the problem is nonlinear and finding a good solution is difficult.

6. ACKNOWLEDGEMENTS:

The assistance for this work is provided by the University Grant Commission and JNU New Delhi, India. Authors would like to thanks to Prof. D. P. Vidyarthi and anonymous reviewers for their valuable suggestions.

7. REFERENCES

[1] M.J. Quinn, Parallel Computing: Theory and Practices, Tata McGraw Hill, India, 2nd edition, 2002.

[2] A.S. Tanenbaum, Distributed Systems: Principles and Paradigms, Prentice Hall of India, 2nd edition, 2002.

[3] I. Foster and C. Kesselman, Grid 2: Blueprint for a New Grid Computing Infrastructure, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA an Imprint of Elsevier, 2nd edition, 2003.

[4] Grid-Café, http://www.gridcafe.org, visited on February 2014.

[5] F. Berman, G. Fox and T. Hey, Grid Computing: Making the Global Infrastructure a Reality, John Wiley and Sons, New York, 2002.

[6] F. Xhafa and A. Abraham, Meta-heuristics for Scheduling in Distributed Computing Environments Studies in Computational Intelligence, Springer, 146:1–37, 2008.

[7] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Co., New York, 1979.

[8] F. Xhafa and A. Abraham, "Computational Models and Heuristic Methods for Grid Scheduling Problems", Future Generation Computer Systems, Elsevier, 26(4):608-621 2010.

[9] P.K. Tiwari and D.P. Vidyarthi, "Observing the Effect of Inter Process Communication in Auto Controlled Ant Colony Optimization based Scheduling on Computational Grid", Concurrency and Computation: Practice and Experience, Wiley, 26(1):241–270, 2014.

[10] L. Ferreira, N. Bieberstein, V. Berstis and J. Armstrong Introduction to Grid Computing with Globus, Redbook, IBM Corporation, 2003.

[11] B. Jacob, M. Brown, K. Fukul and J. Armstrong, Introduction to Grid Computing, Redbook, IBM Corporation, 2005.

[12] Z. Raza and D.P. Vidyarthi, "GA Based Scheduling Model for Computational Grid to Minimize Turnaround Time", International Journal of Grid and High Performance Computing, IGI Global, 1(4):70-90, 2009.

[13] K. Li, "Optimal Load Distribution in Non-Dedicated Heterogeneous Cluster and Grid Computing Environments", Systems Architecture, Elsevier, 54(1-2):111–123, 2008.

[14] Y. Li, Y. Yang, M. Ma and L. Zhou, "A Hybrid Load Balancing Strategy of Sequential Jobs for Grid Computing Environments", Future Generation Computer Systems, Elsevier, 25(8):819-828, 2009.

[15] I. Koren and C.M. Krishna, Fault Tolerant Systems, Morgan Kaufmann is an imprint of Elsevier, New York, 2007.

[16] S. Nesmachnow, B. Dorronsoro, J. Pecero and P. Bouvry, "Energy-aware Scheduling on Multicore Heterogeneous Grid Computing Systems", Journal of Grid Computing, Springer, 11(4):653-680, 2013.

[17] Q. Niu, F. Zhou and T. Zhou, "Quantum Genetic Algorithm for Hybrid Flow Shop Scheduling Problem to Minimize Total Completion Time", Lecture Notes in Computer Science, Springer, 6329(2):21-29, 2010.

[18] Y. Mingscheng, "Quantum Computation, Quantum Theory and AI", Artificial Intelligence, Elsevier, 174(2):162-176, 2010.

[19] J. Gu, X. Gu and M. Gu, "A Novel Parallel Quantum Genetic Algorithm for Stochastic Job Shop Scheduling", Journal of Mathematical Analysis and Applications, Elsevier, 355(1):63-81, 2009.

[20] K.Vivekanandan and D. Ramyachitra, "Bacteria Foraging Optimization for Protein Sequence Analysis on the Grid", Future Generation Computer Systems, Elsevier, 28(4):647-656, 2012.

[21] S.K. Nayak, S.K. Padhy and S.P. Panigrahi, "A Novel Algorithm for Dynamic Task Scheduling", Future Generation Computer System, Elsevier 28 (5):709-717, 2012.

[22] M.E. Moghaddam and R. Bonyadi, "An Immune-based Genetic Algorithm with Reduced Search Space Coding for Multiprocessor Task Scheduling Problem", International Journal of Parallel Programming, Springer, 40(2):225-257, 2012.

[23] R. Kashyap and D.P. Vidyarthi, "Security-aware Scheduling Model for Computational Grid", Concurrency and Computation: Practice and Experience, Wiley, 24(12):1377-1391, 2012.

[24] R. Kashyap and D.P. Vidyarthi, "Security Driven Scheduling model for Computational Grid using NSGA II", Journal of Grid Computing, Springer, 11(4):721-734, 2013.

[25] T.D. Braun, H.J. Sigel and N. Beck, "A Comparison of Eleven Static Heuristic for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing, Elsevier, 61(6):810–837, 2001.

[26] S. Prakash and D. P. Vidyarthi, "Load Balancing in Computational Grid Using Genetic Algorithm", International Journal of Advances in Computing, Scientific and Academic Publishing, US, 1(1):8-17 2011.

[27] S.Prakash and D.P. Vidyarthi, "A Model for Load Balancing in Computational Grid", 18th IEEE Annual International Conference on High Performance Computing (HiPC'11) Bangalore, India, pp. 1-5, 2011.

[28] , S.Prakash and D.P. Vidyarthi, "Observations on Effect of IPC in GA Based Scheduling on Computational Grid", International Journal of Grid and High Performance Computing (IJGHPC), IGI Global, US, 4(1): 66-79, 2012.

[29] S.Prakash and D.P. Vidyarthi, "A Novel Scheduling Model for Computational Grid using Quantum Genetic Algorithm", Journal of Supercomputing, 65(2):742-770, Springer US, 2013.

[30] S.Prakash and D.P. Vidyarthi, "Maximizing Availability for Task Scheduling in Computational Grid using GA", Concurrency and Computation: Practice and Experience, 27(1),197-210, Wiley, UK, 2015.

[31] S.Prakash and D.P. Vidyarthi, "Immune Genetic Algorithm for Scheduling in Computational Grid", Journal of Bio-Inspired Computing, 6(6), 397-408, 2014.

[32] Prakash and D. P. Vidyarthi "A Hybrid GABFO Approach for Scheduling in Computational Grid", International Journal of Applied Evolutionary Computation (IJAEC) vol. 5(3), pp. 57-83, 2014.

[33] C. Kumar, S. Prakash, T. Kumar and D. P. Sahu, "Variant of genetic algorithm and its applications", International Journal of Artificial Intelligence and Neural Networks, vol. 4(4), pp. 8-12, 2014.