

FAULT-TOLERANCE AWARE MULTI OBJECTIVE SCHEDULING ALGORITHM FOR TASK SCHEDULING IN COMPUTATIONAL GRID

Dinesh Prasad Sahu¹, Karan Singh² and Shiv Prakash³

^{1,2}School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi-110067, India

¹dinesh.sahu1230@gmail.com, ²karan@jnu.ac.in

³Department of Chemical Engineering,
Indian Institute of Technology, Delhi-110 016, India

³shivprakash@chemical.iitd.ac.in

ABSTRACT

Computational Grid (CG) creates a large heterogeneous and distributed paradigm to manage and execute the applications which are computationally intensive. In grid scheduling tasks are assigned to the proper processors in the grid system to for its execution by considering the execution policy and the optimization objectives. In this paper, makespan and the fault-tolerance of the computational nodes of the grid which are the two important parameters for the task execution, are considered and tried to optimize it. As the grid scheduling is considered to be NP-Hard, so a meta-heuristics evolutionary based techniques are often used to find a solution for this. We have proposed a NSGA II for this purpose. The performance estimation of the proposed Fault tolerance Aware NSGA II (FTNSGA II) has been done by writing program in Matlab. The simulation results evaluates the performance of the all proposed algorithm and the results of proposed model is compared with existing model Min-Min and Max-Min algorithm which proves effectiveness of the model.

KEYWORDS

Computational Grid, Scheduling, NSGA II, Idle Time, MS, Fault-tolerance

1. INTRODUCTION

Primarily goal of computational grid (CG) is to fulfill the computational need of the tasks [1]. The computational grid [1] is combination of software and hardware infrastructure that facilitate right to use to enormous computational capabilities. The enormous computational capabilities are dependable, reliable, pervasive, and inexpensive even though the users and resources are geographically distributed. The fault tolerance is important parameter in terms of environment and cost. The tasks are migrated from idle nodes to busy nodes so that idle nodes operate in recoverable mode. There may be some nodes in grid which are idle and some which are busy.

The challenging issue in computational grid is fault tolerance consumption due to cost and ecological problem [1]. The fault tolerance consumption based scheduling is key issue in this decade, by taking care of fault tolerance consumption. In order to decrease the fault tolerance consumption, the nodes in the grid have different computing power. Faster computing power of the nodes permits less time for task execution, but they consume much higher power. In fault tolerance -aware scheduling fault tolerance consuming at nodes is minimized during the task execution. In this paper, we explore Makespan (MS) and fault tolerance scheduling algorithm by using NSGA II for grid on a set of nodes. The organization of the work is as follows: The related work has been briefed in section 2 and the problem of the system has been described in section 3 and the proposed model is discussed in section 4. Experimental evaluation with respective observation is depicted in section 5 and finally conclusion is made in section 6.

2. RELATED WORK

Makespan, fault tolerance, turnaround time, availability, reliability etc. are some of the important parameters that are often optimized by scheduling the jobs appropriately on the grid nodes. The grid scheduling problem has been broadly deliberated in literature [1, 2, 7, 8]. Genetic Algorithm (GA) is highly used to find the solution of scheduling problem of CG, as the problem is NP-Hard [1, 17, 18, 19, 20, 21, 22]. Taking the effect of IPC into consideration, [9] uses GA for solving the problem of independent task scheduling in computational grid (CG). In [10], in place of a single, more than one task has been considered for allocation with load consideration using GA focusing on minimization of turnaround time in distributed computing systems. In [10], Load balancing, which is also to be taken care of and is an NP Hard problem, has been widely discussed in [26]. Load balancing on the grid nodes that uses GA has been elaborated in [11]. Both these paper considers load distribution and load variation among the nodes in CG. In [12], fault tolerance aware scheduling for independent tasks using GA is discussed. In this paper we modify the work of [12] by incorporating dependencies between tasks. The effectiveness of the model is studied by comparing with Min-Min, Max-Min. Another important parameter, security also find place in many papers. Security aware scheduling in computational grid (CG) is discussed in [22] with focus on security optimization. Parameter availability is discussed in [13] that demonstrate the availability metric [13], deals with the availability maximization for task scheduling problem of computational grid using GA.

3. THE PROPOSED MODEL

The proposed model, that considers MS optimization in computational grid, has been discussed as follows.

3.1 Fitness Function

Fitness function is derived using Queuing Theory [9, 26].

Then the MS can be compute by equation 3 as it is the maximum time taken for execution by the latest task.

$$MS = \max_{i=1}^m \left[\sum_{j=1}^r \left[\left(\frac{\lambda_i}{\mu_i(\mu_i - \lambda_i)} + \frac{1}{\mu_i} \right) \times \delta_{ji} \times NOI_j \right] \right] \text{-----} (3)$$

Consider a physical nodes of Computational Grid (CG), N is number of CG nodes. For computational simplicity, the nodes are assumed to be connected to each other following mesh topology. The CG offers virtual connections various virtual are mapped to the CG. Let, The number of paths from S to D using the total number of path of equation (1), the maximization function for Fault Tolerance (FT) can be expressed as

$$\text{Maximize Normalize FT} = \max_{i=1,2,\dots,N^p} (F_i^{path}) / \sum_{i=1}^{N^p} F_i^{path} \quad (5)$$

where, F_i^{path} represents fault tolerance of i^{th} path from S to D .

The major components of the original NSGA-II are modified to adjust them into the framework of the optimization problem. The modified components are described below.

$$\text{Minimize MS with MS} \leq \text{MUMS} \quad (6)$$

$$\text{Maximize Normalize FT} \leq 1 \quad (7)$$

The proposed work is a multi-objective optimization problem where the required objectives are maximizing the fault tolerance and minimizing the MS achieved for tasks in CG. Multi-objective optimization is an important decision to find solutions of good quality. Since the conflicting complex nature of CG the multi-criteria makes optimization process complicated. An evolutionary algorithm is NSGA-II, which uses Pareto based evaluation with the capability to find several Pareto-optimal solutions in single iteration of simulation. Optimal fault tolerance of this paper is based on NSGA-II. Initial population is randomly generated and sorted according to fitness and the best half population is selected. The algorithm of NSGA-II is as follows [27]:

- Step 1: A random population is initialized.
- Step 2: Objective functions for all objectives and constraint are evaluated using equation 3 and 7.
- Step 3: Front ranking of the population is done based on the dominance criteria.
- Step 4: Crowding distance is calculated.
- Step 5: Selection is performed using crowded binary tournament selection operator.
- Step 6: Crossover and mutation operators are applied to generate an offspring population.
- Step 7: Parent and offspring populations are combined and a non-dominated sorting is done.
- Step 8: The parent population is replaced by the best members of the combined population.

In Step 3, each solution is assigned a non-domination rank (a smaller rank to a better non-dominated front). In Step 4, for each i^{th} solution of a particular front, density of solutions in its surrounding is estimated by taking average distance of two solutions on its either side along each of the objective [27]. This average distance is called the crowding distance. Selection is done based on the front rank of an individual and for solutions having same front rank, selection is done on the basis of their crowding distances (larger, the better)

3. EXPERIMENTAL EVALUATIONS AND RESULT

The proposed model is evaluated in this section by performing experimentations. The size of the input data set affects the convergence of the solution. The experiments are performed by writing the programs in Matlab version 9.0 and Java and integrated with Gridsim [4] to evaluate the performance on the system 500 GB secondary memory and 4 GB primary memory respectively. Simulation environment uses a random generator between the given ranges with uniform distribution. The following simulation parameters have been used in the experimentations. Input Parameter Values the Number of nodes 16-1000, Number of tasks 50-14000, Range of arrival Rate 1-100 MIPS, Range of task Size 2000-5000 MI and Range of processing speed 101-200 MIPS. Figure 4 contains these parameters and their values.

The performance evaluation is conducted for analysing FAGA (fault tolerance aware genetic algorithm) in the grid. The grid sizes are small (3 to 32 nodes and 10 to 512 tasks), medium (33 to 64 nodes and 513 to 1024 tasks), and large (65 to 128 nodes and 1025 to 2048 tasks). Figure 5 contains the type of grid and corresponding size in terms of the number of nodes and number of tasks. Normal distribution is used to randomly generate the capacity of the resources and the workload of tasks. It is expected that all submitted tasks to the system must be scheduled and all nodes of the system can be used. Above parameters are also used in [4, 12, 14].

For MS minimization minimize MS is taken as fitness which is discussed in [9]. For fault tolerance maximization we minimize TE is taken as fitness function. We run the simulation setup up-to 30 iterations and then average is taken for each result shown in this work.

Table 1. Makespan and tolerance by NSGA II

No. of nodes	No. of tasks	Makespan NSGA II	Makespan MIN-MIN	Makespan MAX-MIN	Tolerance by NSGA II
3	10	223.3	448.2955	448.2955	0.9981942
8	256	5578.8	10000000	3.5E+11	0.9826984
16	256	2674.2	6792966	6792966	0.9951092
32	256	10003.9	15782.03	15782.03	0.9991123
40	600	57297.3	2736774	2736774	0.9974828
80	1200	11123.5	2364124	2364124	0.9967923
50	700	50023.9	1086594	1086594	0.9769240
128	2048	153625.5	4105532	4105532	0.9860082

4. CONCLUSIONS

A huge number of developments has been introduced in Grid Computing in recent few years. Still, being an extremely heterogeneous system, grid poses a number of constraints. The research in the area of grid computing especially resource scheduling is going on. This work focuses on the proposal of scheduling in CG with emphasizing on the MS, and fault tolerance optimization using a NSGA II technique. NSGA II, a meta-heuristic, is a well-known procedure for solving complex multi-objective optimization problem. Performance of the proposed NSGA II method has been studied by carrying out the number of experiments and it is found that it performs well. Also, its comparative study with NSGA II model shows that it has an edge over Min-Min and Max-Min algorithms. The effectiveness of the model is also studies with scaled input and found that model is performing better than both Min-Min and Max-Min algorithms.

ACKNOWLEDGEMENTS

I would like to thank my guide Dr. Karan Singh for his kind suggestions and guidance. I like to thank Dr. Shiv Prakash for his help and guidance in this paper.

REFERENCES

- [1] Foster and C. Kesselman, *The Grid 2, Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers is an Imprint of Elsevier, 2004.
- [2] F. G. Berman, Fox Anthony and J.G. Hey, *Grid Computing, Making the Global Infrastructure a Reality*, John Wiley and Sons, 2003.
- [3] M.R. Garey and D.S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.
- [4] R. Buyya and Murshed M., "Gridsim a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing", *Concurrency and Computation: Practice and Experience*; vol. 14(13-15), pp.1175–1220, 2002.
- [5] R. Buyya, D. Abramson and J. Giddy, *An architecture for resource management and scheduling system in global computational grid*. High Performance Computing Asia, China, IEEE CS Press: USA, vol. 1, pp. 283–289, 2000.
- [6] Z. Shi and J. Dongarra, "Scheduling workflow applications on processors with different capabilities". *Future Generation Computer Systems*, vol. 22, pp. 665–675, 2006.
- [7] S. Prakash and D.P. Vidyarthi, "A Novel Scheduling Model for Computational Grid using Quantum Genetic Algorithm", *Journal of Supercomputing Springer*, Vol. 65(2), pp.742-770, 2013.
- [8] R. Buyya, D. Abramson and J. Giddy, "Nimrod/G: an architecture for resource management and scheduling system in a global computational grid", In :*Proc. of the High Performance Computing Asia, China, IEEE, USA, vol. 1, pp. 283-289, 2000.*
- [9] S. Prakash and D. P. Vidyarthi, "Observations on Effect of IPC in GA Based Scheduling on Computational Grid". *Int. J. of Grid and High Performance Computing* vol. 4(1), pp. 67-80, 2012.
- [10] S. Prakash and D.P. Vidyarthi, "Load Balancing in Computational Grid Using Genetic Algorithm", *International Journal of Advances in Computing, Scientific and Academic Publishing*, vol. 1(1), pp. 8-17, 2011.
- [11] S. Prakash and D. P. Vidyarthi, "A model for load balancing in computational grid", In: *Proc. of 18th annual international conference on High Performance Computing (HiPC11) Bangalore, Student Research Symposium* , pp. 1-5, 2011.
- [12] R. Kashyap and D.P. Vidyarthi, "Fault tolerance -aware scheduling model for computational grid". *Concurrency and Computation: Practice and Experience*, vol. 24(12), pp. 1377-1391, 2012.
- [13] I. Koren and C.M. Krishna, *Fault tolerant systems*. Morgan Kaufmann is an imprint of Elsevier, 2007.
- [14] S. Prakash and D. P. Vidyarthi, "Maximizing availability for task scheduling in computational grid using GA", *Concurrency and Computation: Practice and Experience, Wiley*, vol. 27(1), pp. 193–210, 2015.
- [15] 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*, vol. 61, pp. 810–837, 2001.
- [16] J.H. Abawajy, "Automatic Job Scheduling Policy for Grid Computing", LNCS, Springer-Verlag Berlin Heidelberg, vol. 3516, pp. 101-104, 2005.
- [17] F. Xhafa and A. Abraham, "Meta-heuristics for Grid Scheduling Problems", *Studies in Computational Intelligence Series, Springer*, vol. 146, pp. 1–37, 2008.
- [18] F. Xhafa and A. Abraham, "Computational models and heuristic methods for Grid Scheduling problems", *Future Generation Computer Systems, Elsevier*, vol. 26, pp. 608-621, 2010.
- [19] F. Xhafa and A. Abraham, "A Genetic Algorithm Based Schedulers for Grid Computing Systems" *International Journal of Innovative Computing, Information and Control*, vol. 3(6), pp.1–19, 2007.
- [20] W. N. Chen and J. Zhang, "An ant colony optimization approach to grid work flow scheduling problem with various QoS requirements", *IEEE Transactions on Systems, Man and Cybernetics--Part C: Applications and Reviews*, vol. 39(1), pp. 29-43, 2009.

- [21] S. 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.
- [22] D.A. Menasce and E. Casalicchiop, Quality of service aspects and metric in grid computing. In: Proc. of Computer Measurement Group Conference, pp. 511-532, 2004.
- [23] A. Rajni and I. Chana, “Formal QoS policy based grid resource provisioning framework”, J. Grid Computing, vol. 10(2), pp. 249–264, 2012.
- [24] R.B. Cooper, Introduction to Queuing Theory, 2nd Edition, Elsevier North Holland Publications, 1981.
- [25] S.Prakash and D.P. Vidyarthi, “Immune Genetic Algorithm for Scheduling in Computational Grid”, Journal of Bio-Inspired Computing, vol. 6(6), pp. 397-408, 2014.
- [26] C. Kumar, S. Prakash, T. Kumar and D. P. Sahu, “Variant of genetic algorithm and its applications”, International Journal of Artificial Intelligence and Neural MCGs, vol. 4(4), pp. 8-12, 2014.
- [27] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.
- [28] S. Prakash, D.P. Sahu, Karan Singh, “Maximizing Availability and Minimizing Markesan for Task Scheduling in Grid Computing using NSGA II”, 2nd international conference for computer and communication technology, LNCS, Springer, 1-6, 2015.
- [29] D. P. Sahu, K. Singh and S. Prakash“ A Review on Resource Scheduling Models to Optimize Quality of Service Parameters in Grid Computing using Meta-heuristics”, International Journal of Computer Applications 114(8):pp.1-4, 2015.

AUTHORS

Dinesh Prasad Sahu received the Master degree (Computer Science & Application) M.Tech (Computer Science & Application) from Jawaharlal Nehru University, New Delhi, India. Currently, He is doing Ph.D. (Computer Science & Engineering) under the guidance of Dr. Karan Singh, from Jawaharlal Nehru University, New Delhi, India. Currently, he is working in school of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi. His primary research interests include parallel and distributed system and Grid Computing. He has published 3 papers in proceedings of peer-reviewed Conferences.



Karan Singh (M-1981) received the Engineering degree (Computer Science & Engineering) from Kamala Nehru Institute of Technology, Sultanpur, UP, India and the M.Tech (Computer Science & Engineering) from Motilal Nehru National Institute of Technology UP, India. He is Ph.D. (Computer Science & Engineering) from MNNIT Allahabad deemed university. Currently, he is working in school of Computer & Systems Sciences, JawaharlalNehru University, New Delhi. His primary research interests are in computer network and computer network security. He is reviewer of IEEE & Elsevier conferences and reviewer of International journals. I have research papers (8 Journal, 9 IEEE conferences, 2 Elsevier Conference, 15 National and International conference) and 2 are accepted. He2 is organizer of various workshop, Conference and training. Recently, Dr. Karan work as General Chair of Qshine 2013. Dr. Singh has been joined as Professional member of IEEE, ACM, CSTA, CSI, IACSIT, ICST, IAENG, ACEEE, ISOC and IEEE computer society.



Shiv Prakash received his M.Tech and PhD in Computer Science from School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi, India in 2010 and 2014. He has published around 15 papers in various international journals and around 5 papers in proceedings of peer-reviewed conferences. He is a member of the IEEE and ACM. His research interest includes parallel/distributed system and grid computing.

