Multiple DAGs Scheduling with Deadline Driven Coordinator in Grid

Antony Melnyk

Information Systems and Networks Department, Lviv Polytechnic National University, S. Bandery Str., 12, Lviv, 79013, UKRAINE

antony.melnyk@gmail.com

Abstract. *This paper addresses the problem of multiple DAGs scheduling with user deadline constraint. Efficient tasks scheduling in Grid is a complex and important problem. The goal of the task scheduling system is to find an optimal placement of tasks considering user defined requirements. We preset Grid scheduling approach with Deadline Driven coordinator. The objective of this scheduling approach is to minimize the overall completion time and meet user deadline constraint. We use Directed Acyclic Graph (DAG) as an application model. Multiple DAGs scheduling problem is considered in this research. Simulation results showing efficiency of proposed scheduling approach are presented.*

Keywords

Computer science, distributed systems, grid computing, task scheduling, deadline constraint.

1 Introduction

Grids usually denote the sharing of geographically distributed resources. Each resource may have different properties. Each resource provider can have different policies. Independent users submit their tasks to a Grid system. Each user can submit its own requirements. These requirements should be taken into account by the Grid scheduling system.

Tasks scheduling is recognized as an NP-hard problem. Due to the Grids nature, scheduling problem is becoming more complex in Grid. The goal of the Grid task scheduling system is to find an optimal placement of tasks considering Quality of Service (QoS) requirements.

In this paper we present Grid scheduling approach with Deadline Driven coordinator. The objective of this approach is to minimize the overall completion time and meet user deadline constraints. Deadline is the period by which task execution needs to be completed. We use Directed Acyclic Graph (DAG) as an application model. Multiple DAGs scheduling problem is researched. Results of the experiments, executed in simulation are presented as well.

2 Related Works

The problem of task scheduling on multiprocessors is well understood and has been subject of scientific researches. Most studies in heterogeneous distributed systems are addressed to scheduling of a single DAG. Few researches consider scheduling of multiple DAGs but most of them do not take into account user Quality of Service requirements (price, deadline constraint, etc).

In [1] authors proposed a fairness based approach for scheduling of multiple DAGs on heterogeneous systems. Two policies are described. Both policies arrange DAGs in ascending order of their slowdown value. Then ready task from the DAG with minimum slowdown is selected and scheduled using Hybrid.BMCT or Heterogeneous Earliest Finishing Time (HEFT) heuristic. When task is completed the slowdown of each DAG is recalculated.

In [2] First Come First Serve (FCFS) and Serve On Time (SOT) algorithms for online scheduling of multi DAG are proposed. Both algorithms combine all DAGs into single DAG and then use HEFT for scheduling. FCFS appends new-coming DAG to an exit task of the running DAG. SOT appends every arriving DAG in such way as to start it scheduling as soon as possible.

In [3] four strategies for scheduling multiple workflows on Grids are discussed. DAGs can either be scheduled independently by interleaving their execution, or can be merged into a single DAG. Path Clustering Heuristic is used by these four strategies.

In [4], authors considered multiple workflow scheduling with no or estimated execution times. Proposed workflow allocation strategies consist of two and four stages. On tasks labeling stage static properties of each DAG are computed. At the task allocation stage a suitable site for each ready task is selected. Authors introduced MaxAR allocation strategy which does not consider the tasks requirements. At the local queue prioritization and site scheduling stages a local queue ordering and tasks scheduling are performed.

3 Proposed scheduling approach

First Come First Serve (FCFS) approach for job scheduling is one of the most used in Grid scheduling. This approach does not consider Quality of Service requirements. FCFS policy means DAGs are processed in order they have been arrived to Grid.

3.1 Scheduling System with Coordinator

We introduce scheduling system with coordinator component. The purpose of Coordinator is managing DAGs order in a queue. The coordinator rearranges order of DAGs to satisfy user requirements. It can also analyze new-coming DAG with requirements and decline the request if requirements cannot be achieved.

In the scheduling system with coordinator task scheduling is done in two phases: DAGs coordinating and task scheduling. At the DAGs coordinating phase DAGs are arranged in a queue in order to achieve the pursued goal. The DAG that should be processed first is put on top of the queue. At the task scheduling phase resource allocation is performed. DAGs are processed in queue from top to bottom. When all DAG tasks are completed then DAG is removed from the queue.

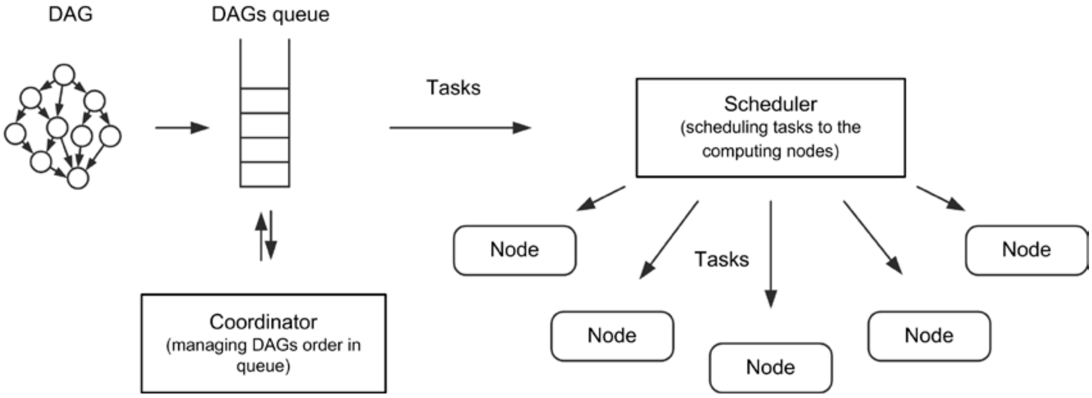

Fig. 1. Scheduling system with coordinator

Different policies can be used by the coordinator depending on the pursued goal. The goal can be maximizing resource utilization, minimizing common schedule length, satisfying user deadline requirement etc. In this paper we consider the Deadline Driven coordinator (DD-coordinator).

3.2 Deadline Driven Coordinator

The DD-coordinator is ordering DAGs considering deadlines specified by users. DAG with earlier deadline is processed first. DAG priority is computed as inversely to deadline value:

$$k_{DD}^{i} = \frac{1}{D^{i}} \tag{1}$$

The DD-coordinator should verify that deadline is realistic. To check whether it will be able to complete the task by the deadline DD-coordinator should check deadline condition:

$$t_{now} + t_{estimate} > D \tag{2}$$

This deadline condition if sufficient to say that deadline cannot be met. The DD-coordinator will decline the request or inform user that deadline cannot be satisfied and suggest defining softer deadline.

The rough DAG run time estimate can be computed as the sum of minimal run time estimate of critical path tasks:

$$t'_{estimate} = \sum_{v_i \in CP} w_{\min}(v_i) \quad (3)$$

If deadline condition check (2) is passed, then DAG will be added to a queue and the scheduling will be started. We need advanced resources reservation policy when scheduling the task. Without reservation resources can be used for another DAGs and DAG complete time will be postponed afterwards.

After reservation coordinator gets more precious estimate $t''_{estimate}$ and checks deadline can be satisfied. If deadline condition check (2) is passed then processing of tasks will started. Otherwise request will be declined and reservation canceled.

3.3 Tasks Scheduler

We used Heterogeneous Earliest Finish Time (HEFT) [5] tasks scheduler. It is the list scheduler which objective is the minimal schedule length. HEFT is an insertion-based algorithm. This means that task can be inserted between two already scheduled tasks. HEFT performs static scheduling so advanced reservation can be applied to the whole DAG before its executing started. Other static schedulers can be used but they are not considered in this paper.

4 Experimental Evaluation

We have used simulation for evaluation. Sesame [6] which is the event driven simulation tool designed for Grid scheduling research has been used. Task Graphs for Free v3.0 (TGFF) [7] have been applied for test DAGs generation.

4.1 Experimental Setup

We generated test graphs of 3125 different types. Deadline was generated as a random value from the range $\left[0; m \cdot \bar{t}_{estimate} / 2\right]$. Where $\bar{t}_{estimate}$ is the average critical path length to average node power ratio; and m is a number of DAGs in a queue. The Grid network topology consists of 20 nodes with randomly generated computational power. The simulation test parameter was the DAGs number in a queue:

$$SET_m = \{5; 10; 15; 20; 25; 30; 35; 40\}$$

We consider two modes of the DD-coordinator: 1) absolute satisfaction and 2) deadlines overdue minimization. In the first mode DAGs are declined if deadline cannot be met but in the second mode DAG is processed anyway.

Totally 320 tests were performed, 160 tests per each mode, 20 tests per each m. In each test random DAGs were selected and scheduling was performed according with each approach. Two approaches were simulated and compared: FCFS and DD. HEFT scheduler was used in both cases.

4.2 Experimental Results

Simulation results are shown in fig. 2 and 3. Average values per each tests group are presented in charts.

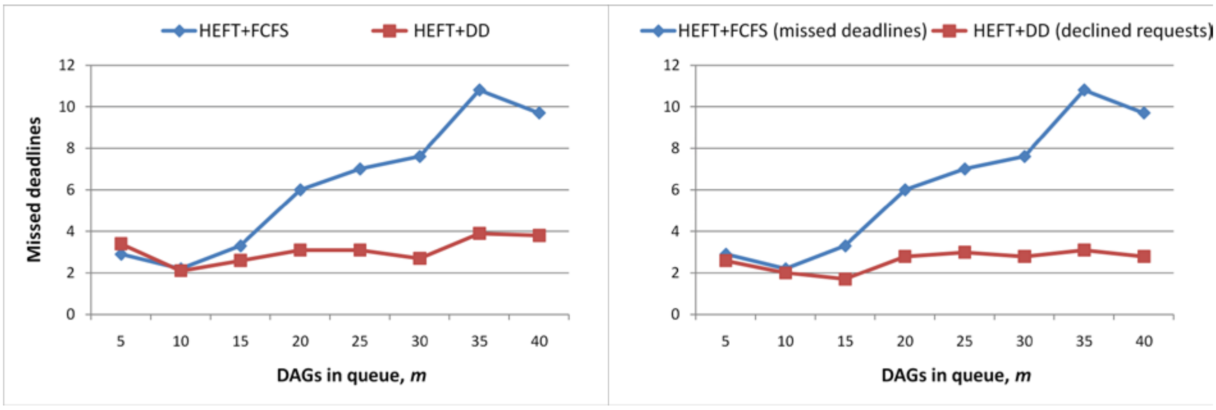

Fig. 2. Missed deadlines/declined requests: **a)** deadlines overdue minimization mode; **b)** absolute satisfaction mode

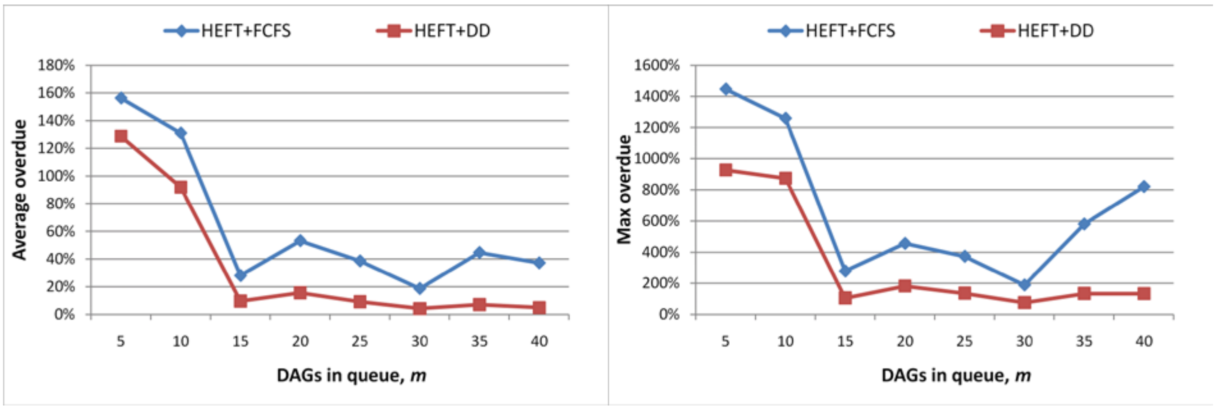

Fig. 3. Deadline overdue in overdue minimization mode: **a)** average; **b)** maximal

These charts show that DD-coordinator using allows to reduce deadlines overdue and number of missed deadlines. Fig. 3 shows that the HEFT+DD approach provides smaller deadline overdue than HEFT+FCFS in all cases. Overdue ratio shown in charts is computed as $\left(t_{competion}-D\right)/\left(D-t_{receive}\right)$. Fig. 2b shows that in absolute satisfaction mode DD-coordinator declines not more than 3 requests to meet deadline of all other requests. Fig. 2a shows that in overtime minimization mode for low number of DAGs in a queue (less than 10) HEFT+DD produces more missed deadlines but deadline overdue is less.

5 Conclusion

We discussed the problem of multiple DAGs scheduling with user deadline constraint. The scheduling system with Deadline Driven coordinator was presented. Two approaches were compared: HEFT-FCFS and HEFT-DD. Simulation results show efficiency of the DD-coordinator. It allows to minimize overtime and to minimize number of DAGs with the missed deadline. Our simulation results consider offline scheduling. Network and computational nodes failures are not considered. Future work focuses on online scheduling.

6 Acknowledgments

We thank Vasyl Lytvyn an Associate Professor in the Information Systems and Networks Department of Lviv Polytechnic National University for support and helpful suggestions.

References

[1] H. Zhao, R. Sakellariou: Scheduling Multiple DAGs onto Heterogeneous Systems. *Proceedings of the 20th international conference on Parallel and distributed processing*, p.159-159, April 25-29, 2006.

[2] L. Zhu, Z. Sun, W. Guo, Y. Jin, W. Sun, W. Hu: Dynamic Multi DAG Scheduling Algorithm for Optical Grid Environment. *Network Architectures, Management, and Applications*, V 6784(1), 2007.

[3] Bittencourt, L.F., Madeira, E.R.M.: Towards the Scheduling of Multiple Workflows on Computational Grids. *Journal of Grid Computing*, 8, pp. 419–441, 2010.

[4] A. Hirales-Carbajal, A. Tchernykh, R. Yahyapour, J. L. González-García, T. Röblitz, J. M. Ramírez-Alcaraz: Multiple Workflow Scheduling Strategies with User Run Time Estimates on a Grid. *Journal of Grid Computing*, 10, pp. 325-346, 2012.

[5] H. Topcuoglu, S. Hariri, and M. Y. Wu: Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3), pp.260-274, March 2002.

[6] A. Melnyk: Sesame – a Simulation Tool for Task Scheduling in Heterogeneous Distributed Systems. *The VII International Scientific and Technical Conference "Computer Science and Information Technologies"*, 2012.

[7] R.P. Dick, D.L. Rhodes, W. Wolf: TGFF: task graphs for free. *Proceedings of the 6th international workshop on Hardware/software codesign*, pp.97-101, March 15-18, 1998.