International Journal of Scientific & Engineering Research, Volume 7, Issue 9, Septmeber-2016
ISSN 2229-5518

On Scheduling Approach for Grid Computing

Hadeer S.Hossam, Hala Abdel-Galil, Mohamed A.Belal

Abstract - Concept of grid computing was appeared in the mid-1990s and is becoming more prevalent and widespread nowadays. Grid computing is used in many areas, including business, e-libraries, e-learning, military applications, medicine, physics, and genetics In this paper, we compare scheduling algorithms techniques based on directed acyclic graph which considers computational cost considering data transfer cost between resources and dependency between tasks.

Keywords - Grid computing, Task Scheduling and directed acyclic graph (DAG).

1. Introduction

Importance of grid computing comes from the need to access resources are geographically distributed and cannot be moved or duplicated to the same location. Grid computing offer approaches to overcome these obstacles.
BY using a grid, distributed resources can be treated as if they are into single place [1]. Assigning tasks to processors /machines is an important issue as it improves the performance of the whole job so our concern will be on scheduling resources in grid computing. Resources can be computers, storage space, instruments, software applications, and data, all connected through the Internet and a middleware layer that provides basic services for security, monitoring, resource management, and so forth as shown in [figure 1].
Resources available on grid are shared under policies that specify who is permitted to access resources, what are the resources that will be available for everyone, and under what conditions they will use these resources [2].

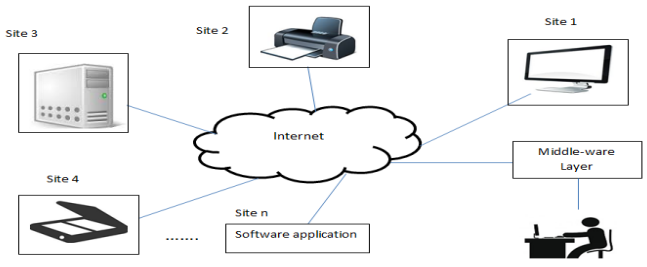

Figure 1 framework of grid computing

Grid computing can increase productivity and performance as it divide applications into tasks and execute these tasks across distributed resources. It also improves efficiency by increasing computational power of available resources.

This paper is structured as follows. Section 2 discusses scheduling process in grid computing. Section 3 outlines preliminaries and definitions of scheduling based on DAG. Section 4 presents related works. Section 5 discusses how the DAG is used in other platforms and section 6 concludes.

2. Scheduling in Grid computing

Scheduler in grids should fetch tasks in job needed to be submitted, defines execution time for each task and dependences between them then assign tasks to available resources as shown in figure 2. The main goal of scheduler is to minimize the completion time of the job and communication time among tasks.

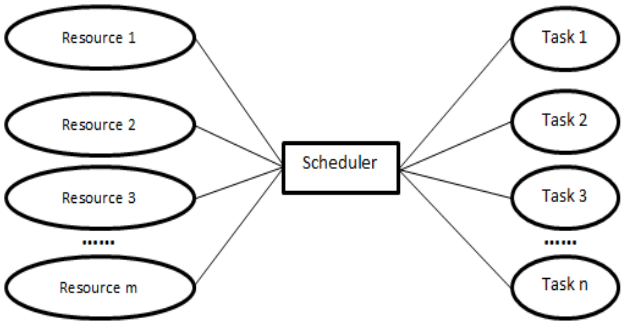

Figure 2 Model of scheduler in grid computing

Scheduler encounters the following obstacles:

1. Heterogeneity obstacle may not come only from the nature of resources but also the way they are

- *Hadeer Samy is Teaching Assistant, computer Science Department. E-mail: Hadeer.samy@gmail.com*
- *Hala Abd-elgalil is Professor of Computer Science. E-mail: aehala2000@yahoo.com*
- *Mohammed Belal is Lecturer of Computer Science. E-mail: dr.mohamedbelal@gmail.com*

All from Faculty of computers and Information system, Helwan university, Egypt

International Journal of Scientific & Engineering Research, Volume 7, Issue 9, Septmeber-2016
ISSN 2229-5518

connected. A lot of algorithms have been emerged to utilize and enhance performance under heterogeneous nature of resources.
2. Dynamic performance for available resources, which happens either from lost connection to a resource or interruption during execution tasks in a job at a certain resource.
3. Mapping tasks to resources is a critical job affects the whole performance of jobs and exploits the concept of parallelism. Assigning tasks to resources done due to status of the resource and is determined before the whole application is executed.
4. In applications that run on the same location cost transferring data between tasks may be neglected. But in a grid computing this is a challenge. Many grid applications costs a lot while transferring data between its tasks so this cost should be taken into account as in these applications computational cost is considered by accessing available resource storage.

Another issue considered during scheduling is the dependency between tasks in the same application.

3. Preliminaries and Definitions Scheduling Based on DAG

DAG is a graph G (N, E) that N is a set of nodes which represent tasks and E is a set of edges which represent data cost transfer between tasks as shown in figure 3. Precedence relation from task 1 to task 2 means that task 2 need data from task 1 before being started. If these two tasks are not assigned to the same computing element, a delay cost c12, which represent completions of task 1 and the beginning of task 2, should be considered between resources holds these two tasks. In general, scheduling applications in a distributed system is a NP-hard problem even when the tasks are independent [3]

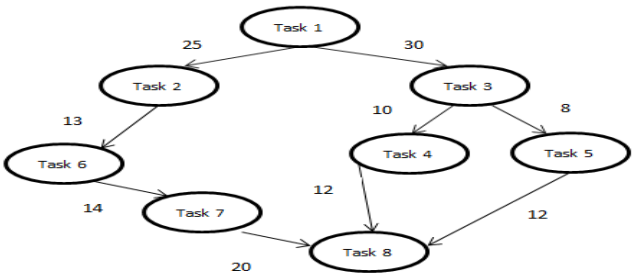

Figure 3 Example of DAG

DAG can be categorized into sequential and parallel structure [4]. Sequential DAG tasks run in serial pattern which means task start execution after the previous one ended. Parallel DAG tasks can be executed concurrently.

Most DAG algorithms are based on list scheduling technique which creates list of tasks and assign each task a priority. There are many approaches used to determine priority of these tasks such as highest level First, Longest Path, and Longest Processing Time [5].

One of the new algorithms used in DAG scheduling is level-based strategy. Level-based methods first organize the DAG into levels, or echelons, such that within each all the nodes are independent and can be scheduled once all the nodes in the previous level have been completed; they then schedule all the nodes in each level, level-by-level from beginning to the end [6].

4. Related Works

Over the past few years, a lot of grid computing algorithms have been introduced. Such algorithms focused on arranging and allocating tasks to resources in a way that minimizes the execution time in order to enhance performance and data transfer cost between these resources.

In this section we present algorithms such as

1. Fully decentralized P2P grid scheduling (FDPGS) algorithm proposed by Piyush Chauhan and Nitin.
2. Sorted nodes in leveled DAG division (SNLDD) algorithm proposed by Nirmeen A. Bahnasawy, Magdy A. Koutb, Mervat Mosa and Fatma Omara.
3. Grid costs and dependence matrix (GCDM) algorithm proposed by Amir M Bidgoli and Zahra Masoudi Nezad
4. CCF (Cluster ready Children First) algorithm proposed by Florin Pop and Valentin Cristea
5. Communication inclusion generational scheduling (CIGS) algorithm proposed by Communication inclusion generational scheduling (CIGS) algorithm

Fully decentralized P2P grid scheduling (FDPGS) algorithm proposed in [7] schedules subtasks of DAG based task. FDPGS schedules the subtasks by taking into consideration three factors. The first two factors are computation cost and communication cost related to the subtasks. Final factor is the waiting time for the subtask because of predecessors and precedence constraints. It proposed a fully decentralized P2P grid scheduling (FDPGS) algorithm for DAG based tasks on the grid. Algorithm consists of the following steps:

1. Priority based task sequence.
2. Selection of subtask for scheduling and predecessor prerequisite.
3. Discovering the most suitable node for execution of selected subtask.
4. Updating information list and finding value of Result Back for next subtask in task sequence.

Algorithm introduces better performance result and minimizes idle time in resources. Simulation using FDPGS

International Journal of Scientific & Engineering Research, Volume 7, Issue 9, Septmeber-2016
ISSN 2229-5518

and genetic algorithm shows that FDPGS takes 98.61% less time than a genetic algorithm to find the schedule. Moreover, schedule of FDPGS algorithm gives results for DAG based task in 6.83% less time than genetic algorithm. However to find a schedule FDPGS algorithm consumes 29.40%, 22.72% more memory than random scheduling and genetic algorithm, respectively.

Sorted nodes in leveled DAG division (SNLDD) proposed by [8] is based on dividing DAG into levels with considering the dependency priority conditions among tasks in the DAG. The tasks will be assigned to the earliest processors according to their priority in the list. Ordering tasks in each level leads to get rid of the heaviest tasks first to reduce the complicated communications dependency between them. If the algorithm finds more than one task is equal in their computation size, it begins executing tasks with large number of communication link first. Disadvantage of this algorithm is that it does not consider communication time between tasks while mapping them to resources. Figure 6 indicates simulation between SNLDD and longest dynamic critical path (LDCP) presented better performance.

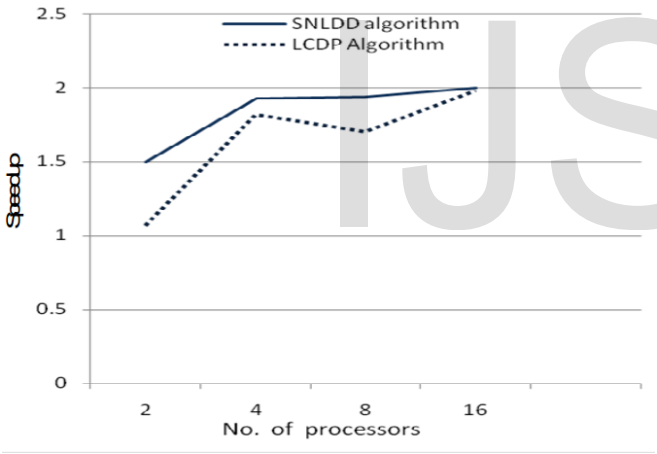

Figure 6: The speedup of two algorithms in cases of 2, 4,8,16 processors with DAG of 40 tasks.

Grid costs and dependence matrix (GCDM) algorithm proposed by [9] which consider dependencies between tasks, data transfer time and execution time for tasks. This algorithm works as the following

1. Represent tasks in DAG graph.
2. Assign each task in the first level (input nodes) to available processors.
3. Next tasks are assigned as the following
   - If task is not shared on the corresponding row between processors, so this Task is assigned to the processor that the tasks dependent on before it in set.
   - If task is shared on the corresponding row between processors, then it adds this task to resource that costs more while transferring data between them.

The purpose of this algorithm is Coupled optimization of time and costs data transfer between tasks. This algorithm does not consider dynamic nature of resources and network bandwidth.

A dynamic DAG scheduling algorithm CCF (Cluster ready Children First) for grid computing proposed by [10], the main phases in this algorithm are resources discovery, DAG tasks scheduling, system selection, and job execution. This algorithm uses two different queues, running queue and children queue. When task is submitted for execution it is inserted in running queue. When this task finished execution all its successors are inserted in children queue. Main functions in the algorithm are assigning Resource, update Resource, suggest Resource, and Monitoring System. The central part of the algorithm is the assign Resource function, which implemented by using genetic algorithm method which works as the following:

1. User submit job for executing.
2. Job tasks are created for scheduling.
3. The batch of tasks is broadcast to all the nodes in the cluster.
4. The tasks are inserted in a queue by its priority according to a fitness function.
   - If the number of tasks in the queue is less than a predefined length of the chromosome, they wait for T units of then padding is added.
   - Else tasks are inserted in the waiting.
5. On each node, up-to-date information on the status of the computers in the Grid on which tasks is sent for execution is kept, by constantly queering a monitoring system.
6. Each node starts with a different, specific initialization of the GA. The subsequent steps of the GA are similar for all the nodes in the cluster, and the same fitness formula is used. In this way, the clients will compute different optima from which the best one will be chosen.
7. The migration of the best current solutions is performed after each step of the GA, thus ensuring that the population finds an optimal solution.
8. The generation of populations ends after a finite, predefined number of steps. At this point, each client in the cluster computes its optimal solution.
9. The same communication procedure as above is used for the final step of the GA.
10. The scheduling obtained is saved in a history file on each node in the cluster.

The purpose of this algorithm is optimizing grid scheduling by the usage of an intelligent and heuristic method. Using

International Journal of Scientific & Engineering Research, Volume 7, Issue 9, Septmeber-2016
ISSN 2229-5518

static CCF algorithm and proposed algorithm simulation results show an improvement of 16% for completion time has been achieved while testing the resource assignment for 25 dependent tasks.

Communication inclusion generational scheduling (CIGS) algorithm [11] used DAG to improve computational grid environment. This algorithm assign priority for each task in the DAG, it gives the source node high priority then span across the remaining nodes. This algorithm followed the following steps:

1. Searches for all source nodes which has highest priority
2. Assign each node to a machine by using heuristic search algorithm such as Min-Min or Max-Min algorithm.
3. By computing expected completion time, all the tasks are scheduled.
4. Update number of node machines and expected completion time for each node.
5. Continue until all nodes are assigned a priority number and scheduled.

Figure 7 indicates that improved CIGS are always small as the number of independent tasks increased.

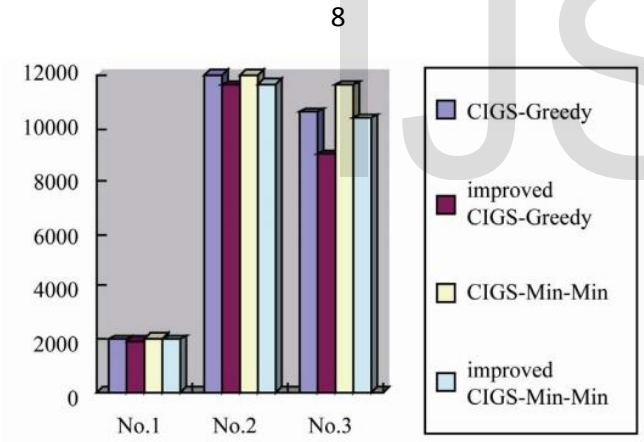

Figure 7: Scheduling performance comparison.

The purpose of this algorithm is to adapt computational grid environment by predicting data transmission time, expected completion time for the ancestors, and preparation time for the machine.

5. DAG USAGE IN OTHER PLATFORMS

DAG is used in a wide range of platforms and introduces good results as committed by the researchers. For example DAG in cloud computing proposed in [12] is called when the user want to delete task, the algorithm checks subtasks related to this task then cost of deletion task is computed then algorithm update DAG. Representing tasks in DAG improve usage of cloud computing resources.

Also Earliest finish time duplication (EFTD) algorithm in heterogeneous cloud computing proposed in [13] allow the user to submit tasks which are represented in DAG and the resource provider to submit the available resources. This model has four main modules which are resource processing, priority processing, task allocation and task duplication.

Another usage of DAG appears also in parallel computing systems, [14] explains also application of DAG such as electronic circuit design, Dataflow programming languages, and in compilers.

6. CONCLUSION

Many obstacles facing grid computing scheduling and nowadays a lot of researches have been submitted trying to overcome these obstacles. Algorithms which submitted nowadays are trying to call algorithms used in other platforms such as first-come-first-served and round robin for example which are used mainly in operating systems architecture to enhance scheduling process. Also there is a trend toward using work-steeling algorithm. Grid computing architecture lack well resource utilization and how task will be divided and distributes among resources which are the main objectives to increase performance.

7. REFERENCES

[1] 7 things you should know about grid computing, Educause65 learning initiative, January 2006.
[2] Fangpeng Dong and Selim G. Akl, Scheduling Algorithms for Grid Computing: State of the Art and Open Problems, School of Computing, Queen's University Kingston, Ontario January 2006.
[3] Meddeber Meriem Department of Computer Science, University of Mascara, ALGERIA and Yagoubi Belabbas Department of Computer Science, University of Oran, ALGERIA, Tasks assignment for grid computing.
[4] Jia Yu and Rajkumar Buyya , The University of Melbourne, Australia,"A Taxonomy of Workflow Management Systems for Grid Computing".
[5] Yu-Kwong Kwok and Ishfaq Ahmad, "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", ,July 1998
[6] Yang Zhang, Charles Koelbel, and Ken Kennedy, Computer Science Department, Rice University, Houston, TX 77005: "Relative Performance of Scheduling Algorithms in Grid Environments".
[7] Piyush Chauhan and Nitin, "Decentralized Scheduling Algorithm for DAG Based Tasks on P2P Grid", Hindawi Publishing Corporation Journal of Engineering Volume 2014, Article ID 202843,14 pages, January 2014.
[8] Nirmeen A. Bahnasawy, Magdy A. Koutb, Mervat Mosa and Fatma Omara, "A new algorithm for static task scheduling for heterogeneous distributed computing systems" African Journal of Mathematics and Computer Science Research Vol. 4(6), pp. 221-234,

International Journal of Scientific & Engineering Research, Volume 7, Issue 9, Septmeber-2016
ISSN 2229-5518

June 2011

[9] Amir M Bidgoli and Zahra Masoudi Nezad, "A new scheduling algorithm design for grid computing tasks", 5th SASTech 2011, Khavaran Higher-education Institute, Mashhad, Iran. May 12-14.

1) Florin Pop, Valentin Cristea, "INTELLIGENT STRATEGIES FOR DAG SCHEDULING OPTIMIZATION IN GRID ENVIRONMENTS", 16th International Conference on Control Systems and Computer Science (CSCS16'07).

2) Liang Yu, Gang Zhou, Yifei Pu ,"An Improved Task Scheduling Algorithm in Grid Computing Environment", February 21, 2011.

3) Taoshen Li, and Xixiang Zhang, "On the Scheduling Algorithm for Adapting to Dynamic Changes of User Task in Cloud Computing Environment", nternational Journal of Grid Distribution Computing Vol.7, no.3 (2014), pp.31-40

4) Zhaobin Liu, Wenyu Qu*,† , Weijiang Liu, Zhiyang Li and Yujie Xu, "Resource preprocessing and optimal task scheduling in cloud computing environments", Accepted 8 December 2013.

5) Rafiqul Zaman Khan, Javed Ali, "Use of DAG in Distributed Parallel Computing", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 11, November 2013.