RESEARCH ARTICLE OPEN ACCESS

Effective Time and Cost Based Task Scheduling In Grid Computing

Shashi Bhushan Semwal [1], Mr. Amit Das [2]
M.Tech Research Scholar [1], Professor [2]
Department of Computer Science and Engineering
JBIT (Jai Baghwan Institute of Technology), Uttarakhand
Dehradun - India

ABSTRACT

This Paper titled **"EFFECTIVE TIME AND COST BASED TASK SCHEDULING IN GRID COMPUTING"** presents the effective time and cost scheduling technique followed by the scheduler decide the Grid system throughput and consumption of the source in to the grid. At the present time parallel and distributed systems are modify in the association and the idea of Grid computing, a group of dynamic and heterogeneous resources linked via Internet and linked by many and many clients, is currently suitable a certainty. The Grid system is dependable for the implementation of task submits to it. The superior Grid system will contain a job scheduler which mechanically finds the most suitable machines on which a specified job is to run. This source range is very significant in dropping the total implementation time and cost of processing the jobs which depends on the job scheduling algorithm.

Keywords:- Grid Computing, Effective time, Effective weight, Processor Speed

I. INTRODUCTION

Grid is a novel communications for system computing on home or geographical scales that can energetically represent heterogeneous computing resources. Grid computing is generally worked in much scientific and engineering application area. Grid application addresses collaboration, data sharing, and cycle sharing and extra modes of communication that involves distributed resources and services [1].

The idea of grid computing is receiving popular day by day with the appearance of the internet as an everywhere media and the wide extend accessibility of controlling computers and system as little cost product part. Resources can be computational method (such as conventional computers, clusters, or even powerful skill), exacting class of devices (such as sensors) and smooth group devices. A number of applications require extra calculate power than can be presented by a single resource in arrange to answer them inside a possible/reasonable time and cost. Geographically circulated resources require to be reasonably joined together to make them work as a combined resources.This explains to the popularization of a ground describe Grid Computing. The area of Grid Computing is a demonstration of the development of scattered and cluster computing situation. The idea of a computational grid was first planned with Ian Foster and Carl Kesselman in the mid 1990s.Because then field has mixed up into one of the most stimulating areas of learn in the computing society[2].

II. EXISTING SCHEDULING TECHNIQUE

1.Non Duplication Technique

List development is mostly worked development algorithms. In list development, a load is allocate to every job and edge, based on which a prepared task. A list is constructing by assigning priority for every task. Then, the tasks are selected priorities and scheduled based and takes a lowest cost function. We have two types of the list development heuristics; HEFT (Heterogeneous Earliest Finish Time) and CPOP (Significant Path on a Processor) be study in [3]. The upward rank and downward position of each task are computed at the start. HEFT algorithm is selected firstly highest upward rank of each step. After that chosen job is allocate to a host that preserve less its first finish time. In difference, CPOP algorithm all the time chooses the job with the maximum entire rank (upward rank + downward rank) value. A CPOP schedules all important every jobs onto a particular host with the finest presentation and reduced the total processing time. For the period of processing, if an elected job is non-critical, it will be plot a host which might decrease its first end time, as in HEFT. HEFT and CPOP contain small complexity, i.e., lower algorithm implementation time [4].

2. Duplication Technique

The projected algorithm is a duplication-based fixed scheduling algorithm and it change from the previous algorithms by deal through the minimization of the schedule extent plus the integer of processors used as divide problems to be optimized in two different phases.

Real world difficulties in grid are multipurpose means they essential more than one idea. For e.g. total processing time or makespan, economy, dependability, loyalty, etc. We projected a scheduling algorithm stand on multi-use namely total economy cost/execution time.

III. METHODOLOGY

GAMDL is XML language support and the GAMDL syntax is developed as a position of XML-Schema [5]. XML is the mainly usually worked modeling language for workflow details in network computing and has a very successful set of development tools.

XML-Schema is worked to simplify a position of rules to which an XML file must match in order to be measured "valid". As a W3C normal, it supplies a prosperous information model that permit us to state complicated organization and limitation worked in GAMDL. The use of XML-Schema for GAMDL helps us mostly develop a GAMDL parser by the unlock basis XML development records, Apache XML Beans [6]. XML Beans binds XML information with Java objects during the schema of the information spoken in XML-Schema. In our example, following we have calculated the GAMDL XML-Schema, the XML Beans compiler find the GAMDL schema and produce Java codes that contact a GAMDL file. All the information types, XML papers and parts in GAMDL are plotted to Java classes. Using these mechanically produced codes, we can simply expand a GAMDL parser in clean Java language.

In the situation of network computing, there exist numerous applications for example bioinformatics, Financial analysis etc. that can be billed as workflows. A scheduling difficulty can be clear as the job of various network services to various workflow jobs. Represented by $W = (N, E, T, C)$, where N is a place of n computational jobs T is a place of job computation capacity (one unit of computation capacity is one million instructions), E is a place of communication arcs or edges that explain precedence limitation between the jobs and C is the place of message information from close relative tasks to child tasks (one unit of communication data is one Kbyte). The value of $\tau_I \in T$ is the computation capacity for the job $n_i \in N$. The value of $c_{ij} \in C$ is the message information move along the edge e_{ij}, $e_{ij} \in E$ from job n_i to job n_j, for n_i, $n_j \in N$. A job node with no any parent node is called 'start nodes' and a job node with no any child node is called 'end node'[7].

In this work, a quite permanent methodology has been accepted for significant the weights of the computational jobs and communicating edges[8]. In this study, we explain the 'processing time (makespan)' as the total time connecting the end tasks and create time of the start task in the given DAG. Equally, the 'economic cost' (EC) is the summary of the economic costs of all workflow jobs scheduled on different source which can be defined as:

$$EC = \sum_{j=1}^{m} Dj$$

Where m is the total number of services (source) presented in grid and Dj is the finishing cost due to jobs scheduled on source j which can be calculated as:

$$Dj = PBTj \times \alpha(pj) \times Mj$$

Where $\alpha(pj)$ is the executing ability of source j , Mj is the processing cost per MIPS (in grid dollar or g$) of perform task on source j and $PBTj$ is the total busy time addicted by jobs scheduled on source j (PBT is three for source P1 and eight for source P2)[9]. In our representation, the cost of the at rest slots between the listed jobs on any source is also careful in economic cost as it is complicated for the network scheduler to schedule other workflow jobs in these at rest slots. After transferred of all tasks in a DAG on grid resources, the makespan of the schedule will be the real finish time of the end task which can be computed as:

$$\text{makespan} = FT(task_{end}) - ST(task_{open})$$

Where FT and ST are the finish and start time of end task and open task respectively. In general, ST is zero except some cases where open task has to stay for source due to some local tasks listed on it. Since a big place of task graphs with various properties is used, it becomes needed to simply the schedule length (makespan) to a lower bound, called the normalized schedule length (NSL) of a schedule which can be calculated as:

$$NSL = \frac{makespan}{\sum_{n \in CPmin} \min pj \in P\{\omega ij\}}$$

The denominator is the summary of the smallest implementation costs of jobs on the *CP*min where calculation cost of task n_i on source p_j is ω_{ij}. The average values of NSL over numerous task graphs are worked in the simulation.

IV. RESULT

In this part we will explain the calculation of makespan or total completing time followed by cost of implement the DAG. We can apply the DAG of changed sizes. This algorithm is developed in JAVA for approximation of time and cost of various random task graph or DAG of various graph size (100,200,300,400,500).The algorithm have been processed in a grid of mixed cluster of various sizes(5,10,15,20,25) with four sources in each cluster. We have got eight nodes and information of processor is four. On compute the over java code we have next production. We got changed output on modify Node Load **(weight in million instructions)** and modify on CPU Speed **(mips).** We experiential that when weight is similar and processor speed varies, the machine cost stay similar there is only modify in whole time occupied by nodes. One significant issue to be pointed out is the time occupied by processor will be constant after a sure quantity of execution of code whether there is modify in node weight or in processor speed.

NODES WEIGHT	**5000,4000,2000,1000,2000,4000,6000,3000**
PROCESSOR SPEED	**25,10,2,1,2**

Table 1.1 Input the nodes Weight and Processor speed

TOTAL TIME TAKEN	**2600**
TOTAL MACHINE COST	**3600**

Table 1.2 Show the Result Machine Cost and Total Time

If the performance is best, enter the various types of processor speed the cost remain constant [10].

V. CONCLUSION AND FUTURE WORK

Computational Grids permit the creation of effective computing surroundings for contribution and aggregation of extend resources for solve significant problems in science, engineering and commerce. The sources in the network are geographically scattered and owned by many association with changed procedure and cost policies. They include a large figure of self-interested entity (distributed owners and users) with different objectives, precedence and aim that vary from time to time. The organization of resources in such a big and scattered environment is a complicated job. Inside this, a novel bi-criteria workflow arrangement approach has been obtainable and analyzed. We have proposed a proficient scheduling algorithm called "**EFFECTIVE TIME AND COST BASED TASK SCHEDULING IN GRID COMPUTING**" which accepted the makespan and productive cost of the schedule and minimize the needs of processors. The algorithms contain executed to schedule different random DAGs onto various grids of mixed clusters of different sizes. The schedule produced by ETCTSGC algorithm is improved than other joined bi-criteria algorithms in admiration of together execution time and cost-effective.

Future work would engage just beginning a development system which also thinks about the weight estimation or other objectives during which we reduce the processing time and the cost-effective.

REFERENCES

[1] Fost, Ian, Carl Kesselman, Steve Tuecke, "The Aanatomy of the Grid: Enabling Scalble Virtual Organization", International Journal of Supercomputer Application, 2001.

[2] Klaus Krauter, Rajkumar Buyya, and Muthucumaru Maheswaran. "A Taxonomy and Survey of Grid Resource Management System for Distributed Computing", Software: Practice and Experience (SPE) journal, Wiley Press, USA, 2001.

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

[4] R. Sakellariou and H. Zhao, "A Hybrid Heuristic for DAG Scheduling on

Heterogeneous Systems", in Proceedings of the 13th Heterogeneous Computing Workshop (HCW'04),Santa Fe, New Mexico, USA, pp.26-30, 2004.

[5] W3C XML Schema, http://www.w3.org/XML/Schema.

[6] The XMLBeans, http://xmlbeans.apache.org/.

[7] M. Maheswaran and H. J. Siegel, "A Dynamic Matching and Scheduling Algorithm for Heterogeneous Computing Systems", in Proceedings of the Seventh Heterogeneous Computing Workshop, 1998.

[8] Abramson D, Giddy J, Kotler L. "High performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid Computing", Mexico, 1–5May 2000. IEEE Computer Society Press: Los Alamitos, CA, 2000.

[9] Buyya R, Abramson D, Giddy J. A Case for Economy Grid Architecture for Service Oriented Grid Computing, 23 April 2001 San Francisco, CA. IEEE Computer Society Press: Los Alamitos, CA, 2001.

[10] Sullivan III WT, Werthimer D, Bowyer S, Cobb J, Gedye D, Anderson D. "A New Major SETIPproject Based on Project Serendip Data and 100,000 Personal Computers" Proceedings of the 5th International Conference astronomy 1997

[11] Buyya R, Abramson D, Giddy J. Nimrod/G: "Architecture for a Resource Management and Scheduling System in a Global Computational Grid". 14–17 May 2000. IEEE Computer Society Press: Los Alamitos, CA, 2000.

[12] Buyya R, Abramson D, Giddy J. "An Economy Driven Resource Management Architecture for Global Computational Power Grids". 26–29 June 2000, Las Vegas. CSREA Press, 2000.

[13] Buyya R, Giddy J, Abramson D. "An Evaluation of Economy-Based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep Applications".Pittsburgh, PA, 1 August 2000. Kluwer Academic Press, 2000.

[14] Chapin S, Karpovich J, Grimshaw A, "The Legion Resource Management System. Proceedings of the 5th Workshop on Job Scheduling Strategies for Parallel Processing". San Juan, Puerto Rico, 16 April 1999. Springer: Berlin, 1999.

[15] Litzkow M, Livny M, Mutka M. Condor "A Hunter of Idle Workstations. Proceedings 8th International Conference of Distributed Computing Systems (ICDCS 1988)", San Jose, CA, January 1988. IEEE Computer Society Press: Los Alamitos, CA, 1988.

[16] Berman F, Wolski R. The AppLeS Project: "A status report. Proceedings of the 8th NEC Research Symposium", Berlin, Germany, May 1997.

[17] Casanova H, Dongarra J. NetSolve: "A Network Server for Solving Computational Science Problems". International Journal of Supercomputing Applications and High Performance Computing 1997; 11(3):212–223.

[18] Kapadia N, Fortes J. PUNCH: "Architecture for Web-Enabled Wide-Area Network- Computing". Cluster Computing: The Journal of Networks, Software Tools and Applications 1999; 2(2):153–164.

[19] P. A. Dinda, "Design, Implementation, and Performance of an Extensible Toolkit for Resource Prediction in Distributed Systems", IEEE Trans. Parallel Distrib. Syst., 2006,3317(2), pp. 160-173.

[20] P. A. Dinda, and D. R. O'Hallaron, "Host Load Prediction Using Linear Models", Cluster Computing, 2000, 3(4), pp. 265-280.

[21] D. M. Swany, and R. Wolski, "Multivariate Resource Performance Forecasting in the Network Weather Service", In Proceedings of the 2002 ACM/IEEE conference on Supercomputing Baltimore, Maryland, November 2002, pp. 1-10.

[22] R. Wolski, L. Miller, G. Obertelli, and M. Swany, "Performance Information Services For Computational Grids", In: Resource Management for Grid Computing, J.

Schopf Nabrzyski, and J. Weglarz, editors, Kluwer Publishers, Fall, 2003.

[23] E. Caron, A. Chis, F. Desprez, and A. Su, "Design of Plug-In Schedulers for a GRIDRPC Environment",Future Generation Computer Systems,2008,24(1),pp.46-57.

[24] E. Caron, and F. Desprez, "DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid", International Journal of High Performance Computing Applications, 2006, 20(3), pp. 335-352.

[25] M.Wu, and X.-H. Sun, "Grid Harvest Service: A Performance System of Grid Computing", J. Parallel Distrib. Comput., 2006, 66(10), pp. 1322-1337.

[26] L.-T. Lee, D.-F. Tao, and C. Tsao, "An Adaptive Scheme for Predicting the Usage of Grid Resources", Comput. Electr. Eng. Pergamon Press, Inc., Tarrytown, NY, USA, 2007, 33(1), pp. 1-11.

[27] Miss. Hemangi Joshi, Prof. Viraj Daxini "An Effective Load Balancing Grouping Based Job & Resource Scheduling Algorithms for Fine Grained Jobs in Distributed Environment" IJCSMC, Vol. 3, Issue. 4, April 2014, pg.1209 – 1220

[28] Soheil Anousha,Mahmoud Ahmadi "An Improved Min-Min Task Scheduling Algorithm in Grid Computing" (2013) 34

[29] Zahra Pooranian, Mohammad Shojafar, Jemal H. Abawajy, Mukesh Singhal "A New Job Scheduling Algorithm for Grid Computing" (2013)

[30] Rajiv Ranjan, Suya Nepal, Raj kumar Suyya "Model-Driven Provisioning Of Application Services in Hybrid Computing Environments" (2013)

[31] Pankti Dharwa, "Light-Weight Communication System in Grid" Vol. 2, Issue 4, July-August 2012, pp.972-976

[32] Sanjay Patel, Madhuri Bhavsar, "Qos Based User Driven Scheduler for Grid Environment" An International Journal (ACIJ), Vol.2, No.1, January 2011

[33] Q. Huang, N. Xiao, and B. Liu, "Grid Load Forecasting Based on Least Squares Support Vector Machine,Computer Technology and development, 2007, 17(6), pp. 32-35.

[34] B. Vandy, and G. Bruno, "Brokering Strategies in Computational Grids using Stochastic Prediction Models", Parallel Computing, May 2007, 33(4-5), pp. 238-249.