International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Approach to Solve NP Complete Problem Using Game Theoretic Scheduling Algorithm and Map-Reduce on Clouds

V. Mogal[1], Shekhar H. Pingale[2]

[1]Professor, Department of Computer Engineering, RMD Sinhgad School of Engineering, Savitribai Phule Pune University, India

[2]Department of Computer Engineering, RMD Sinhgad School of Engineering, Savitribai Phule Pune University, India

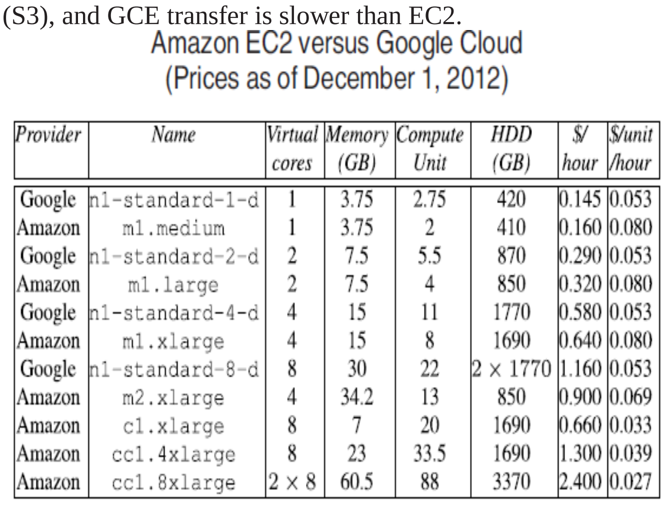
Abstract: ***In case of NP-complete problem, it is curious issue to schedule large scale parallel computing applications on heterogeneous systems like Hybrid cloud. End user want to meet Quality of Service requirement (QoS).To process huge number of Bag-of-Task (BOT) concurrently in such environment with QOS is a Big problem. For that it needs a exact solution. Here we propose Multi-objective scheduling algorithm to schedule particular BOTs. Algorithm optimizes two user objectives such as, Execution time and Economic cost where two constraints to consider are Network Bandwidth and Storage requirement.***

Keywords: Multi-objective scheduling, Map Reduce, Bags-of-tasks, Hybrid clouds, NP Complete.

1. Introduction

Distributed computing systems such as clouds and grids have evolved over decade to support various types of scientific applications with dependable, consistent, pervasive, and inexpensive access to geographically-distributed high-end computational capabilities. To program such a large and scalable infrastructure like hybrid clouds, loosely coupled-based coordination models of legacy software components such as bags-of-tasks (BoT) and workflows have emerged as one of the most successful programming paradigms in the scientific community[1].

Researchers try to address NP complete problems and how to schedule large-scale scientific applications to distributed and heterogeneous resources such that certain objective functions such as total execution time (called from hereafter makespan) in academic Grids or economic cost (in short cost from hereon) in business or market-oriented clouds are optimized, and certain execution constraints such as communication cost and storage requirements are considered and fulfilled. Currently, only a few schemes can deal with both perspectives, such as optimizing user objectives (e.g. makespan ,cost) while fulfilling other constraints, and providing a good efficiency and fairness to all users. On the other hand, many applications can generate huge data sets in a relatively short time, such as the Large Hadron Collider expected to produce 5-6 petabytes of data per year, which must be accommodated and efficiently handled through appropriate scheduling bandwidth and storage constraints. Since there may be many different types of applications or tasks in the clouds which are competitors for use of available resources, many issues arise and need to deal with such as the efficient resource allocation for different applications taking into account their individual performance, cost, bandwidth, storage, and other potential constraint , and the exact use of resources from the system perspective. Comparison of the pricing between EC2 and GCE revealing that Google has better prices overall. Bandwidth between different Amazon and Google cloud services demonstrating that Google cloud storage (GCS) is slower than Amazon simple storage service (S3), and GCE transfer is slower than EC2.


Amazon EC2 versus Google Cloud
(Prices as of December 1, 2012)

Provider	Name	Virtual cores	Memory (GB)	Compute Unit	HDD (GB)	$/ hour	$/unit /hour
Google	n1-standard-1-d	1	3.75	2.75	420	0.145	0.053
Amazon	m1.medium	1	3.75	2	410	0.160	0.080
Google	n1-standard-2-d	2	7.5	5.5	870	0.290	0.053
Amazon	m1.large	2	7.5	4	850	0.320	0.080
Google	n1-standard-4-d	4	15	11	1770	0.580	0.053
Amazon	m1.xlarge	4	15	8	1690	0.640	0.080
Google	n1-standard-8-d	8	30	22	2 × 1770	1.160	0.053
Amazon	m2.xlarge	4	34.2	13	850	0.900	0.069
Amazon	c1.xlarge	8	7	20	1690	0.660	0.033
Amazon	cc1.4xlarge	8	23	33.5	1690	1.300	0.039
Amazon	cc1.8xlarge	2 × 8	60.5	88	3370	2.400	0.027

Figure 1: Comparison on the basis of Price

In this paper, we address these issues by proposing a communication and storage-aware multi-objective scheduling scheme for an important class of workflow applications characterized by large sets of independent and homogeneous BoTs, interconnected through data flow dependencies:

1) Multi-objective scheduling minimizes the expected execution time and economic cost of applications based on a sequential cooperative game theoretic algorithm.
2) Communication and storage-aware scheduling minimizes the makespan and cost of applications while taking into account their bandwidth and storage constraints for transferring the produced data.

The main advantages of our game theoretic algorithm are its faster convergence by using competitors and environment information to determine the most promising search direction by creating logical movements, its minimum requirements regarding the problem formulation, and its easy customisation for new objectives. We compare the performance of our approach with six related heuristics and show that, for the applications with large BoTs, our algorithm is superior in complexity (orders-of-magnitude improvement), quality of result (optimal in certain known cases). Our algorithm may not be suited for applications with complex dependencies between BoTs (such as the hydrological Invmod, meteorological MeteoAG or the bioinformatics association test workflows), for which the scheduling problem cannot be properly formulated as a typical and solvable game, consisting of phases specifically defined so that game players can bargain with limited dependencies between each other.

2. Background

- Utility computing has emerged as a new service provisioning model and is capable of supporting diverse computing services such as storage, network, servers and e-Business applications and e-Science over a global network. For utility computing based services, users consume the services when they need to, and pay only for what they use. Utility computing encourages organizations to offer their specialized applications and other computing utilities as services.
- Many Grid applications in areas such as bioinformatics and astronomy require workflow processing in which tasks are executed based on their control or data dependencies. A number of Grid workflow systems with scheduling algorithms have been developed. It provide the execution of workflow applications and minimize their execution time on Grids. Given this motivation, we focus on developing workflow scheduling based on user's QoS constraints such as deadline and budget. In general, processing time and execution cost are two typical QoS constraints for the workflow execution on "pay-per-use" services. We have presented a cost based scheduling , it minimizes workflow execution cost within a certain deadline. we develop a genetic algorithm based scheduling heuristic to minimize execution time while meeting user's budget constraint[2].
- Grid Computing is a type of distributed computing to solve critical and big computations where the resources in the remote places are accessed by connecting all the resources together in a network. The resources of computers owned by individuals or by organizations from several countries are connected to form a single, vast super computer. A Grid gathers together resources and makes them accessible in a secure manner to users and applications. Grid computing is needed when there is a necessity for huge computing of data and the data is stored in different institutions. Based on the needs, grid is classified into Private vs. Public, Regional vs. Global, All-purpose vs. Particular scientific problem. Vast applications from the users are divided into a set of subtasks and sent to the several resources connected to the main server which is freely available. There are four different classes of Grid users such as end-users of applications, developer, system administrators, and manager. After the computations are over at the particular resources, the results are sent back to the main global server. As computations are received by the global server, the result is then delivered to the user.
- Grid scheduling plays the major role to schedule the tasks onto the processor efficiently. There are three phases in grid scheduling such as Resource discovery, job execution and System selection . At execution of the job, the resources should be reliable so that the job can execute successfully. The reliability of the grid system is affected by several factors such as hardware or software failure. Reliability of computational hardware, software and data resources that comprise the grid and provide the means to execute user applications and reliability of grid networks for messaging and data transport are important and should be met [3].
- Recently, a number of cost-aware workflow scheduling heuristics have been proposed and evaluated. Even though multiple criteria have been considered, their aim was to optimize a single objective. For example, they either minimize execution cost while meeting users' deadline or minimize execution time while meeting users' budget. However, in many situations users may need more information such as the range of QoS levels available and associated costs before making decisions. Furthermore, QoS levels and prices offered by service providers may be highly diverse and may not be directly correlated with the utility perceived by the users. For example, users may prefer some assignments which have slightly longer execution times but offer large savings in execution cost [4].
- Particle Swarm Optimization (PSO) is a self-adaptive global search based optimization technique introduced by Kennedy and Eberhart . The algorithm is similar to other population-based algorithms like Genetic algorithms but, there is no direct re-combination of individuals of the population. Instead, it relies on the social behavior of the particles. In every generation, each particle adjusts its trajectory based on its best position (local best) and the position of the best particle (global best) of the entire population. This concept increases the stochastic nature of the particle and converge quickly to a global minima with a reasonable good solution [5].
- The load balancing problem is formulated as a cooperative game among the computers and the communication subsystem and game theory offers a suitable modeling framework. The several decision makers (e.g. computers and the communication subsystem) cooperate in making decisions such that each of them will operate at its optimum. Based on the Nash Bargaining Solution (NBS) which provides a Pareto optimal and fair solution, we provide an algorithm for computing the NBS for our cooperative load balancing game [6].
- Thus, it is an important research problem to model the Grid and its constituents by taking into account the potential non cooperativeness at various levels. With such modeling, we can then study the impact of selfishness and subsequently design proper strategies to avoid its adverse impacts. This can in turn lead to a much more efficient utilization of the Grid processing resources. However, despite that there have been several recent attempts in scrutinizing the Grid from a so-called "market" oriented

perspective, the modeling problem of the Grid with realistic selfishness concepts is relatively unexplored [7].

- In the global approach, a centralized job allocator minimizes the expected system response time over all jobs. For single class jobs, the job allocation problem is formulated using nonlinear optimization

and polymatroid optimization . Also, for multiclass jobs, most solutions are based on nonlinear optimization . In the cooperative approach, the computing nodes cooperate, resulting in an optimal job allocation that minimizes the expected execution time of jobs per node. This is implemented by complete information cooperative game theory. In the non cooperative approach, each job attempts to minimize its own response time by playing a non cooperative game with the other jobs . Such non cooperative games are applicable to many game-theory-based resource allocation problems [8].

3. Architecture

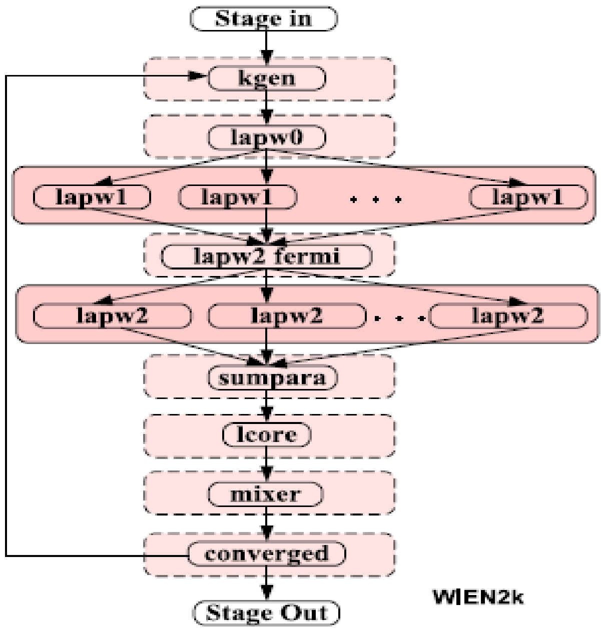


Figure 2: Real-world application examples –WIEN2K

WIEN2k is a program package for performing electronic structure calculations of solids using density functional theory based on the full-potential (linearized) augmented plane-wave ((L)APW) and local orbital method. We have ported this application onto the hybrid clouds by splitting the monolithic code into several coarse-grain tasks coordinated in a simple workflow. The lapw1 and lapw2 BoTs can be solved in parallel by a fixed number of homogeneous tasks called k-points. A final task named converged applied on several output files tests whether the problem convergence criterion is fulfilled.

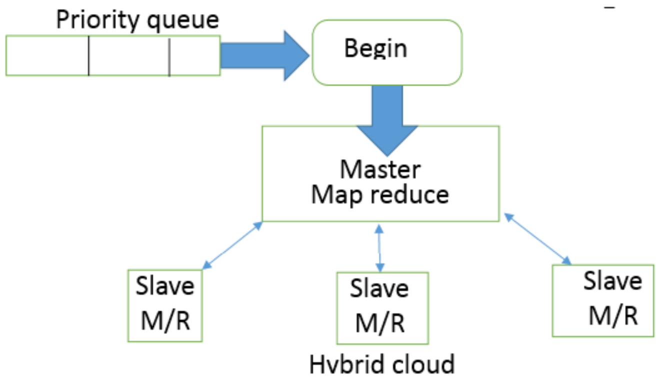


Figure 3: Proposed System Architecture

4. Multi Objective Scheduling Algorithm

The goal of this paper is to design a new algorithm for scheduling a set of applications defined according to Definition and consisting of a huge number of tasks (for which existing algorithms do not scale) in an environment modeled. This Multi Objective Scheduling Algorithm aims to optimize two objectives: aggregated cost and aggregated makespan, also optimize constraints such as bandwidth and storage constraints.

```
1. BEGIN
2. S= {{I},{O}}
 I= Input , O=Output
 Where I={data for operation,AS,K,M,ℵk,bl,br};
 Output: O= {Solution to Problem };
AS : set of applications
K : No. of BOTs
M : No. of sites
ℵk : No. of task of BOTs
Bl : Bandwidth limit
Br : Bandwidth requirement of BOT

3. Phase 1: Status updation of BOT matrix
4. Phase 2: Scheduling And appropriate resource allocation
 for particular job
5. Phase 3: Remove completed BoTs, release resources and
 start new game by repeating Phase 1 and Phase 2.
6. END
```

5. Conclusion

- With increasing focus on large-scale applications on clouds in hybrid environment, it is important to manage a workflow services to efficiently and effectively schedule and dynamically steer execution of applications.
- Here we are preferring to do analysis on bottlenecks of a class of applications with bags-of tasks characterized by a large number of homogeneous tasks, and presented a communication- and storage-aware multi objective scheduling solution based on a sequential cooperative game algorithm for four important metrics: makespan, cost, storage resource and network bandwidth.

References

[1] Multi-Objective Game Theoretic Scheduling of Bag-of-Tasks Workflows on Hybrid Clouds, IEEE transaction on cloud computing, vol .2 , no. 1, January-March 2014.
[2] J. Yu and R. Buyya, "A Budget Constrained Scheduling of Workflow Applications on Utility Grids Using Genetic Algorithms," Proc. First Workshop on Workflows in Support of Large-Scale Science, June 2006.
[3] A Reliable Schedule with Budget Constraints in Grid Computing International Journal of Computer Applications (0975 – 8887) Volume 64– No.3, February 2013
[4] H.M. Fard, R. Prodan, J.J.D. Barrionuevo, and T. Fahringer, "A Multi-Objective Approach for Workflow Scheduling in Heterogeneous Environments," Proc. 12th IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing (CCGrid '12), 2012.
[5] S. Pandey, L. Wu, S.M. Guru, and R. Buyya, "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," Proc. 24th IEEE Int'l Conf. Advanced Information Networking and Applications, pp. 400-407, 2010.
[6] S. Penmatsa and A.T. Chronopoulos, "Cooperative Load Balancing for a Network of Heterogeneous Computers," Proc. 21st IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS '06), Apr.2006.
[7] Y. Kwok, S. Song, and K. Hwang, "Selfish Grid Computing: Game-Theoretic Modeling and Nas Performance Results," Proc. Fifth IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing (CCGrid '05), citeseer.ist.psu.edu/kwok05selfish.html, 2005.
[8] P. Ghosh, K. Basu, and S.K. Das, "A Game Theory-Based Pricing Strategy to Support Single/Multiclass Job Allocation Schemes for Bandwidth-Constrained Distributed Computing Systems," IEEE Trans. Parallel Distributed System, vol. 18, no. 3, pp. 289-306, Mar.2007.

Author Profile

Prof. V. Mogal received the B.E. and M.E. Degrees in Computer engineering. He is working as Assistant Professor in Department of Computer Engineering, RMD Sinhgad School of Engineering Pune, India.

Shekhar Pingale Reseach Scholar at RMD Sinhgad School of Engineering, Savitribai Phule Pune University. He has received B.E. in Computer Engineering from Savitribai Phule Pune University, Pune. Currently he is pursuing M.E. in Computer Engineering from RMD Sinhgad School of Engineering, Savitribai Phule Pune University, Pune, Maharashtra, India