Deadline and Budget Distribution based Cost-Time Optimization Workflow Scheduling Algorithm for Cloud

Amandeep Verma
University Institute of Engg. & Technology
Panjab University, Chandigarh

Sakshi Kaushal
University Institute of Engg. & Technology
Panjab University, Chandigarh

ABSTRACT
Cloud computing is a rapidly growing area. Cloud Computing offers utility-oriented IT services to the users worldwide over the internet. As compared to grid computing, the problem of resource management is transformed into resource virtualization and allocations. Effective scheduling is a key concern for the execution of performance driven applications, such as workflows in dynamic and cost-driven environment including clouds. In case of Cloud computing, issues such as resource management and scheduling based on users' QoS constraints are yet to be addressed especially in the context of workflow management systems. In cloud, the users submit their workflows along with some QoS constraints like deadline, budget, trust, reliability etc. for computation. In this paper, we are considering the two constraints: deadline and budget. We propose Deadline and Budget distribution-based Cost-Time Optimization (DBD-CTO) workflow scheduling algorithm that minimizes execution cost while meeting timeframe for delivering results and analyze the behavior of the algorithm.

General Terms
Cloud Computing, Workflows, Scheduling Algorithms.

Keywords
Cloud computing, Grid Computing, Workflows, QoS, Deadline and Budget Constraints.

1. INTRODUCTION
Cloud computing has emerged as a new large-scale distributed computing paradigm that provides a dynamically scalable service delivery and consumption platform facilitated through virtualization of hardware and software with the provision of consuming various services on demand over the internet[1]. For cloud computing based services, users consume the services when they need to, and pay only for what they use. Cloud
Computing delivers three kinds of services: Infrastructure as aService (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). These services are available to user in a Pay-per- use –on demand model [2].
Many cloud applications require workflow processing in which tasks are executed based on their control or data dependencies. As workflow scheduling is a well-known NP-complete problem [3], many heuristic and meta-heuristics methods have been proposed for distributed systems like grids [4]. For a utility service like cloud computing, pricing is dependent on the level of Quality of Service (QoS) offered. Typically service providers charge higher prices for higher QoS. Therefore, users may not always need to complete workflows earlier than they require. Instead, they prefer to use cheaper services with lower QoS that are sufficient to meet their requirements [5]. As a result, few of Cloud workflow management systems with scheduling algorithms have been developed by several projects. However, scheduling workflows based on users' QoS requirements (e.g. deadline and budget) has been given very little attention in these existing Cloud workflow management systems [6,7].

In this paper, we propose DBD-CTO workflow scheduling algorithm for cloud environment. The objective of the proposed scheduling algorithm is to develop workflow schedule such that it minimizes the execution cost and yet meet the time constraints imposed by the user. In order to solve scheduling problems efficiently, we partition workflow tasks. Along with workflow partition, the deadline and budget constraint assignment strategy is also discussed to distribute the overall deadline and budget over each partition [8].

The remainder of the paper is organized as follows. Section 2 provides an overview of the workflow management system for cloud. Our proposed workflow scheduling algorithm is described in Section 3. Experimental details and simulation results are presented in Section 4. Finally, we conclude the paper in Section 5.

2. WORKFLOW MANAGEMENT SYSTEM
Figure 1 represents the architecture of a Cloud Workflow Management System (WfMS) [9]. The components of the workflow management system are discussed below:

A. User Interface

The workflow management system allows the user to specify their requirements along with the descriptions of tasks and their dependencies using the workflow specification. The user will submit the workflow along with required QoS parameters through a web portal.

B. Plug-in Components
The plug-ins support workflow executions on different environments and platforms. These are used for querying task and data characteristics, transferring data to and from resources (e.g., transfer protocol implementations, and storage and replication services), monitoring the execution status of tasks and applications (e.g., real-time monitoring GUIs, logs of execution, and the scheduled retrieval of task status), and measuring energy consumption [9]. The broker, which is acts as an intermediate between users and service provider, discovers the available resources from different service providers with the help of these plug-in's. The broker then prepares a list of different resources (services)

for each task of workflow system that are able to execute particular task based upon the QoS provided by the user. The broker gets the estimated time and cost of each task on different services and then decides where to execute a particular task.

C. Service Provider/ Resources

The resources are at the bottom layer of the architecture and include clusters, global grids, and clouds. The WfMS has plug-in components for interacting with various resource management systems present [9].

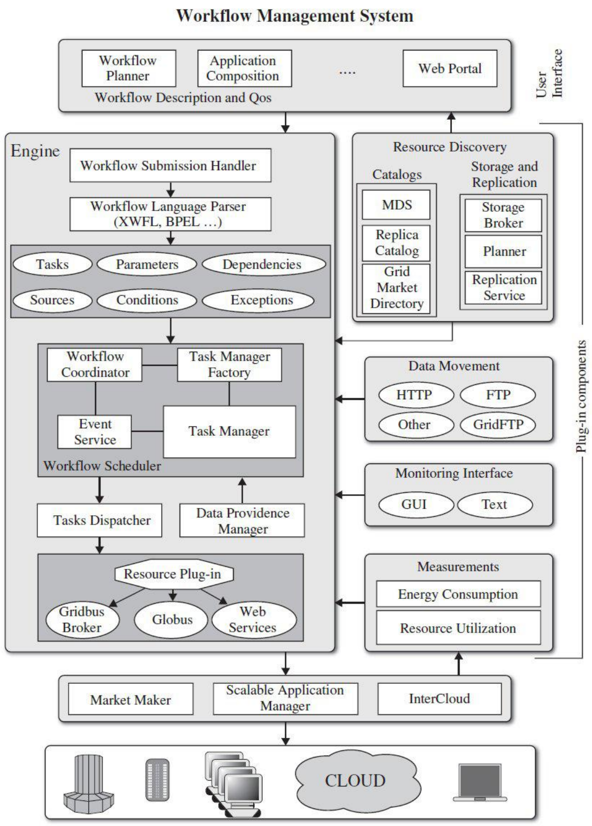

Figure 1: Architecture of Cloud Workflow System [9]

3. WORKFLOW DISCRITION

We model a workflow application as a Directed Acyclic Graph (DAG). Let T be the finite set of tasks Ti and let A be the set of directed arcs of the form (Ti, Tj), where Ti is called the parent task of Tj and Tj is called the child task of Ti . We assume that a child task cannot be executed until all of its parent tasks are completed [10]. Let B be the cost constraint (Budget) and D be the time constraint (Deadline) specified by the users for workflow execution.

Let m be the total number of services available. There is a set of services S_i that are capable of executing a particular task T_i. Each arc A of DAG is labeled with two numbers. First number represents the data transmission time (in sec) and second one represents that data transmission cost (in $). We denote t_i^j as the sum of processing time and data transmission time and c_i^j as the sum of the service price and data transmission cost for processing Ti on the service S_i^j. The scheduling problem is to map every Ti onto a suitable service S_i^j to minimize the execution cost of the workflow ad complete it within the budget B and deadline D given by the user [11].

4. PROPOSED DBD-CTO SCHEDULING ALGORITHM

The steps of the proposed DBD-CTO scheduling algorithm are listed below:

Step1. Discover available services and request QoS parameters of services for every task.

Step2. a) Group workflow tasks into task partitions: We categorize workflow tasks to be either a *synchronization task* or a *simple task*. A synchronization task is defined as a task which has more than one parent or child task. Rests of the task are known as *simple task* .The workflow is partitioned such that a set of simple tasks that are executed sequentially between two synchronization tasks [12].

b) Estimates the minimum execution time and cost for each task from the available set of services.

c) Calculate the total expected completion time by summing the data processing time and minimum execution time.

d) Like in step 2 (c), total expected cost is calculated.

e) If values, calculated in (c) and (d) are less than the deadline (D) and budget (B) provided by the user, only then workflow will be executed otherwise not.

Step3. a) Distribute user's overall deadline and budget into every task partition. T*he overall deadline and budget is divided over task partitions in proportion to their minimum processing time and processing cost respectively calculated in step 2 (b).*

The deadline and budget is distributed according to the following rule:
The execution times of tasks in workflows vary; some tasks may only need 20 minutes to be completed, and some others may need at least one hour. Thus, the deadline and budget distribution for a task partition should be based on its execution time and processing cost. Since there are multiple possible processing times and processing cost for every task, we use the minimum processing time and minimum processing cost to distribute the deadline and budget respectively.

b) Sort the all service lists in decreasing order of their cost.

Step4. Choose a service to execute particular task from the list so that processing cost and execution time should be less than the partition's deadline and budget value for that task.

Step5. Repeat the step4 until all tasks within all partitions have been scheduled.

5. EXPERIMENT AND SIMULATION RESULT

We had developed our own simulating programme in JAVA and randomly generate the workflows and perform the above said algorithm over them .For the workflow, we created a list of three services for each task of the workflow. The scheduler that is implemented in the broker part calls DBD-CTO to choose a particular service such that overall workflow execution should be in deadline and budget constraints specified by the user.

The figure 2 shows one of the workflow used for simulation. The number present in bracket of each node represent the length of particular task in MI. Each edge is labeled with data transmission time (in Secs) and data transmission cost (in $) between two tasks.

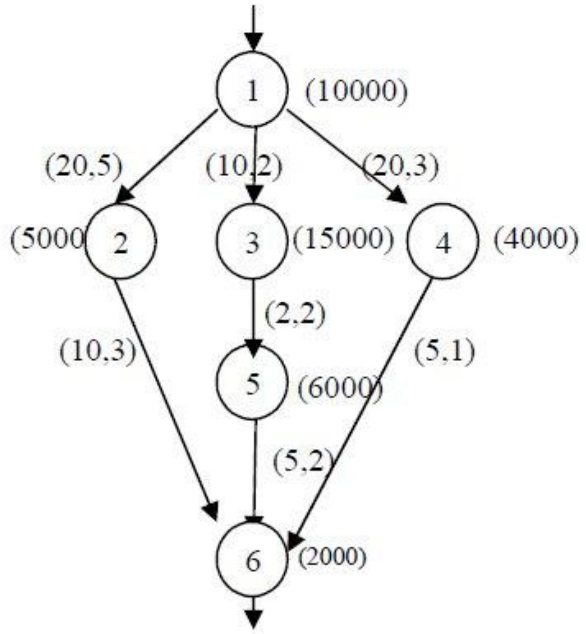

Figure 2: Workflow 1

When DBD-CTO is called, the workflow is partitioned into sub parts. For workflow1, three partitions are created. Partition one consists of task 2, partition 2 consists of task 3 and 5 and partition 3 consists of single task 4.

So, here task 1 and 6 are synchronous tasks and rest are simple one.

The table 1 shows the available set of services for task 1. In the same fashion, the scheduler creates the lists for all tasks.

TABLE 1: Different Services Available for Task 1

Service ID	MIPS rating	Processing Time(Sec)	Cost ($)
1	500	20	6
2	1000	10	12
3	2000	5	18

Table 2 shows the simulation result for five workflows.

TABLE 2: Results of DBD-CTO

Workflow	Execution Time(Sec)	Execution Cost($)	User's Value	
			Deadline	Budget
Workflow 1	29	108	100	150
Workflow 2	37	72	90	100
Workflow 3	58	108	120	150
Workflow 4	53	84	110	120
Workflow 5	30	120	120	200

Figure 3 & 4 shows the graph between varying budget and execution time and execution cost respectively for the workflow 1 for the fixed deadline of 100 sec. The x-axis shows the different budget values and y–axis shows the corresponding execution time and execution cost values. It is clear from the figure 3 & 4, that when budget is very low like 50, the scheduler is choosing the cheaper services that take more time to execute the particular task. But when budget is higher, the same task is executed on costlier services within less time. So, as the budget value increases, the execution time decreases and cost increases within the same deadline.

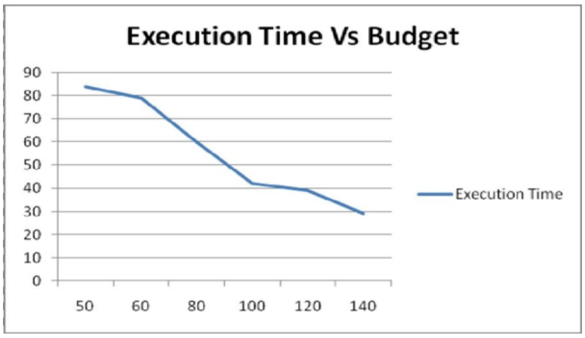

Figure 3: Execution Time vs Budget at deadline 100s

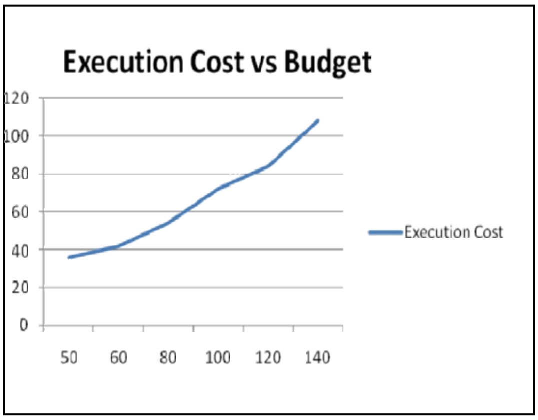

Figure4: Execution cost vs Budget at deadline 100s

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a Deadline and Budget distribution based Cost and Time optimization workflow scheduling algorithm that minimizes the cost and time of execution while meeting the deadline and budget constraint specified by the user. We also described workflow partitioning and overall deadline and budget partitioning optimized execution planning and efficient run-time rescheduling. In the future, we will extend our algorithm to support real time workflows and will compare our algorithm with the existing available techniques.

7. REFERENCES

[1] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., and Brandic, I. 2009. “Cloud computing and Emerging IT platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility”, Journal of Future Generation Computer Systems, vol.25, no. 6, 599–616.

[2] Verma, A., and Kaushal, S. 2011. “Cloud Computing Security Issues and Challenges: A Survey”, in proceeding of Springer International Conference on

Advances in Computing and Communication, Kochi, India, 445-454.

[3] Yu, J., and Buyya, R. 2008. "Workflow Scheduling Algorithms for Grid Computing", In: Xhafa F, Abraham A (eds) Metaheuristics for scheduling in distributed computing environments. ISBN: 978-3-540-69260-7. Springer, Berlin.

[4] Liu, K. 2009. "Scheduling Algorithms for Instance Intensive Cloud Workflows", Ph.D Thesis, Swinburne University of Technology, Australia.

[5] Le, K., Chen, J., Jin, H., and Yang, Y. 2009. "A Min-Min Average Algorithm for Scheduling Transaction Incentive Grid Workflows", 7th Australasian symposium on Grid Computing and e-Research (AusGrid), Australia 41-48.

[6] Zhangjun, W., Xiao, L., Zhiwei, N., Dong, Y., and Yun, Y. 2011. "A Market-Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems", Journal of Supercomputing, DOI 10.1007/s11227-011-0578-4.

[7] Ke, L., Hai, J., Jinjun, C., Xiao, L., Dong, Y., and Yun, Y. 2010. "A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on Cloud Computing Platform", International Journal of High Performance Computing Applications, 1-16.

[8] Yu, J., and Buyya, R. 2005. "Cost Based Scheduling of Scientific Workflow Application on Utility Grid", 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, 8p-147.

[9] Pandey, S., Buyya, R., and Dileban, K. 2011. "Workflow Engine for Clouds", Cloud computing Principles and Paradigms, John Wiley & Sons, Inc., 321-344.

[10] Saeid, A., and Mahmoud, N. 2010. "Cost-driven Scheduling of Grid Workflows Using Partial Critical Paths", 11th IEEE/ACM International Conference on Grid Computing, Brussels, 81-88.

[11] Meng, X., Lizhen, C., Haiyang, W., and Yanbing, B. 2009. "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing", IEEE International Symposium on Parallel and Distributed Processing with Applications, Chengdu, 629-634.

[12] Yu, J., and Buyya, R. 2006. "A Budget Constraint Scheduling of Workflow Application on Utility Grid Using Genetic Algorithm", 15th IEEE international symposium on High Performance Distributed Computing (HPDC'06), Paris, 1-10.