A Scheduling **Algorithm for Multiple** Workflows **in Cloud Computing**

Gyeong-Geun Choe and Bong-Hwan Lee

Department of Information and Communications Engineering, Daejeon University

96-3 Yongun-Dong, Dong-Gu, Daejeon 300-716, Korea

clove38@empal.com, blee@dju.kr

Abstract. Cloud computing is an emerging computing paradigm for sharing resources which includes infrastructures, applications, softwares, etc. A variety of applications that can be used for cloud service are represented in workflows. These applications must be appropriately allocated to resources or services in cloud. In this paper, we propose a scheduling algorithm with multiple workflows for cloud computing environment. Simulation results show that the proposed algorithm can improve the mean makespan comparing with previous scheduling algorithms.

Keywords: Green IT, Cloud Computing, Workflow Scheduling, Workflow Applications, Makespan.

1 Introduction

Recently rapidly increasing energy use driven by the data centers has resulted in enormous IT costs. Thus, "green IT" have become a hot topic and will continue to be an important issue for several years to come [3][6][7]. So cloud computing has interested many enterprises and research institutes since it offers an financial benefit, in that users share a large centrally managed pool of storage and computing resources rather than owning and managing their own systems [3]. Cloud computing also refers to both the workflow applications delivered as services over the Internet, the servers, and software in the data centers [3].

Namely, the workflow applications must be appropriately allocated to resources or services in cloud.

The workflow definition was given in [3][4][5]: the automation of the processes and the movement of information which leads a work process. Thus the workflow scheduling issue is becoming the key point for the efficient use of systems in a cloud environment; the efficient workflow scheduling strategy to consumers such as enterprises, is attracted for eliminating cost. As cloud computing has these features, we consider new points for the workflow scheduling strategy [6][7][8]. First, cloud provides services to multiple users who need the same services at the same time. So the workflow scheduling must schedule multiple workflows with various qualities of service (QoS) parameters. Second, users can access resources whenever they want. Thus, cloud services need to be ready to start for new workflows. Many heuristics have been proposed for workflow scheduling for a single workflow [3]. Namely, those scheduling algorithms do not consider the multiple workflows of cloud computing features and focus on a single QoS parameter such as execution time or cost. To address the problem mentioned above, in this paper, we propose a workflow scheduling strategy for the multiple workflows with multiple QoS parameters considering cloud computing environment. When scheduling multiple workflows in the cloud, those workflows compete for the same resources or services. In this situation, we consider the objective in order to minimize the mean makespan which interests both user and service providers.

The rest of the paper is organized as follows. Section 2 explains the proposed workflow scheduling strategy. And section 3 describes the experiment design and evaluates the performance of the proposed algorithm along with other popular ones. Finally, we conclude our results in Section 4.

2 Scheduling Algorithm

In this section, we describe our new scheduling strategy. In cloud computing system, the scheduling system does schedule the tasks of the each workflow for processing when the workflow applications are submitted in the workflow scheduling system. Afterward, the workflow's tasks are allocated and served in the proper resources

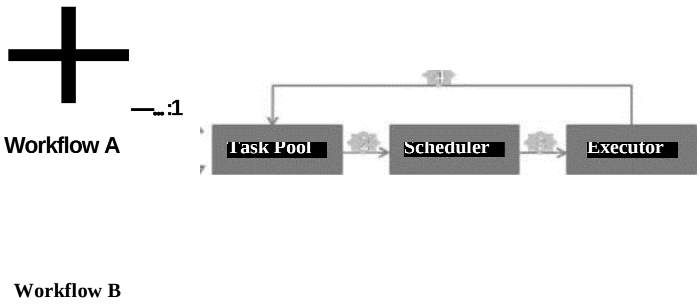

Fig 1. An overview of the scheduling algorithm

2.1 System Overview

In order to schedule tasks of each workflow dynamically and optimize the resource allocation decision, our system consists of three core components as Fig. 1: *Task Pool, Scheduler,* and *Executor*. The task pool is used to manage multiple workflows for dynamic cloud service. Once the workflows are submitted to the task pool (only when they become ready to execute), the task dependence constraint is transparent to the executor. Although these tasks may belong to different workflows, the executor applies our scheduling algorithm to the tasks in the task pool [9].

For multiple workflows scheduling algorithm based on cost, our description is divided into prioritization tasks and selection services. The prioritization tasks are to decide rank of tasks. The decision services are to allocate tasks in cloud service or resource according to such priority.

2.2 Prioritization Tasks

We consider and modify the formula proposed by Zhifeng Yu and Weisong Shi [9] using our multi-workflow scheduling approach based on cost:

$$rank(Task_i) = i, + \max_{Task \, \in succ(Task_i)} \overline{(C(i,j) + Task_i)}, \; i = 1, A, n \qquad (1)$$

where succ (Task ,) QUOTE **SUM Meet)** is the set of immediate successors ofTask„ $C\overline{(i, j)}$ is the average communication cost of edge (i, j), and T_i is the average execution time of Task,. In (1), we do not consider c(i, j) , which is the average communication cost of edge (i, j), but consider only the priority between tasks of workflow. We can modify (1) into (2).

$$\text{rank}(\text{Task}) = i_i + \max_{\text{Task}_j \text{ E succ}(\text{Task})} (\text{Task}_j), \quad i = 1, \tag{2}$$

Our workflow scheduling algorithm based on cost is a dynamic algorithm for the multiple workflows. Thus, we need to define an equation which will be used at the scheduler. We compute cost using throughput in (3). The throughput is computed according to the estimated execution time $\mathbf{t_i}$ in (3):

$$\text{Cost}(\text{Task},) = \text{Length}(\text{Task},) \text{ x } \text{Tput}(\text{Task},),$$

$$\text{Tput}(\text{Task}_i) = \frac{1}{\mathbf{T}_i} \quad i = 1, A, n \tag{3}$$

where Cost (Task ,) defines the cost for each tasks of workflow, Tput (Task ,) is the individual throughput for each task of workflow, and Length (Task,) is the size of each task.

2.3 Selecting Services or Resources

Our algorithm is to schedule all ready tasks according to their attributes defined in section IV and allocate the resources or services in cloud as follow:

1) *Workflow Submission:* When any new workflow comes in the system, it is submitted to the Task Pool. After calculating the attributes of all tasks in the workflow, the computed tasks are inserted into the queue of the Task Pool. The first tasks in the queue will be submitted in the Scheduler. Afterwards, whenever there are completion notifications of the tasks by the Executor, the Task Pool will determine if any successor tasks get ready and submit them in the Scheduler. The task attributes information such as T, , or waiting time is

submitted along with the task.

2) *Task Scheduling:* Whenever there are services available and the ready tasks in the queue, the scheduler will compute the cost of all tasks currently in the queue and schedule all tasks and then repeatedly executes as follow:

a) Remove the first tasks from the Task Pool.

b) Schedule the tasks with the lower cost into the executor. If the cost value is the same, the tasks with the higher waiting time are scheduled in advance.

c) Allocate the task in the proper service.

d) Put the task into the next round queue if there are no services in process.

3) *Task completion notification:* When a task is finished successfully, the executor will notify the Task Pool of the task completion status and remove the corresponding tasks from the Task Pool.

We name the proposed scheduling algorithm based on cost as MWSC (multiple workflow scheduling algorithm based on cost).

3 Experimental Results

In this section, we present the experiment results for evaluating the effectiveness of MWSC. To evaluate the performance of our algorithm, we have developed an experimental simulator. We comparatively evaluate *RANK HF* and *RANK HYBD* in [3][9] . To guarantee fairness of the simulation, we conduct 30 iterations for each parameter set and each algorithm, and calculate the average values of the results. Our cloud environment simulator has 10 services. Every service can execute one task at the same time.

We presume another set of attributes to model the dynamic workflow:

- *Number of concurrent workflows.* This is the total number of workflows concurrently executed in a cloud service. In the experiment, we simulate 5, 10, 15, 20, and 25 numbers of concurrent workflows, respectively.
- *Task number of workflow.* This is the total number of tasks executed at the same time. It consists of randomly generated scenario. For example, when we generate 20 of concurrent workflows, each workflow is in random scenario.

So the total task number is different individually.

- *Arrival interval.* We are interested in the arrival interval at which the workflows are submitted into the cloud service. In the simulation, we assume the arrival interval follows a Poisson distribution.
- *Task Size.* This is the size of each task in workflows. The size is randomly generated and ranges from 50 to 1000 units respectively, since every applicant's information or credit history is individually different.

The performance of three algorithms is compared with respect to various workflows which are consisted of randomly generated scenarios. When generating each workflow concurrently, we consider mean makespan for each workflow. Fig. 2 shows the mean makespan with different number of concurrent workflows. As a result, *RANK HF* and *RANK HYBD* have almost identical performance with respect to mean makespan. MWSC is different from others with respect to mean makespan. The mean makespan rate of MWSC over *RANK HF* and *RANK HYBD* increases from 8% to 6%.

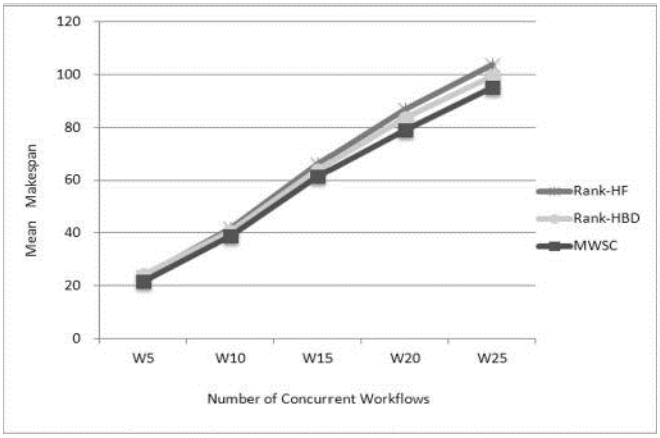

Fig 2. Mean Makespan

Conclusively, we confirmed that our MWSC algorithm outperforms than *RANK HF* and *RANK HYBD algorithm* with respect to the mean makespan, which is interesting to user and service provider's perspective.

4 Conclusion

The multiple workflow scheduling is a very important problem in cloud computing. Many workflow scheduling algorithms have been proposed for a single workflow. Also, some workflow scheduling algorithms have considered multiple workflows. However, those algorithms do not also consider the various QoS parameters. To address this problem, in this paper, we have proposed a multiple workflows scheduling algorithm based on cost (MWSC) in cloud computing environment. Our new algorithms should not only consider the multiple workflows, but the various QoS parameters and metrics for evaluating services.

The evaluation results show that our algorithm outperforms significantly over *RANK HF* and *RANK HYBD* algorithm with respect to mean makespan.

In future work, we will study the mean availability from service provider's perspective and the mean time to recovery (MTTR) in metrics for measuring service reliability. We will also apply the practical workflow applications to evaluate the algorithms.

ACKNOWLEDGMENT

This research was financially supported in part by the Ministry of Education, Science and Technology (MEST) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation.

References

1. D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, Cost-benefit analysis of cloud computing versus desktop grids, In Proc. IEEE Int. Symp. Parallel Distrib. Process, Rome, Italy, May (2009).
2. J. Yu and R. Buyya, "Workflow Scheduling Algorithms for Grid Computing," Tech. Rep., GRIDS-TR- 2007-10, University of Melbourne, Australia(2007).
3. M. Armbrust, Above the Clouds: A Berkeley View of Cloud Computing, EECS Department, University of California, Berkeley (2009).

4. M. Wieczorek, R. Prodan, and T. Fahringer,"Scheduling of scientific workflows in the ASKALON grid environment," *SIGMOD Record*, vol. 34, no. 3, pp. 56-62 (2005).
5. M. Lopez, E. Heymann, and M. Senar, "Analysis of dynamic heuristics for workflow scheduling on grid systems," In Proceedings of the 5th International Symposium on Parallel and Distributed Computing (ISPDC' 06). IEEE, pp. 199-207 (2006).
6. Ian Foster, Yong Zhao, loan Raicu, and Shiyong Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop (GCE '08) (2008).
7. L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," vol 39, pp.50-55, Jan. (2009).
8. Meng Xu, Lizhen Cui, Haiyang Wang, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing," IEEE International Symposium on Parallel and Distributed Processing with Applications, pp. 629-634, (2009).
9. Zhifeng Yu and Weisong Shi, "A Planner-Guided Scheduling Strategy for Multiple Workflow Applications," ICPPW, pp.1-8, International Conference on Parallel Processing - Workshops, (2008).