Research Paper Engineering

Various Constraint workflow Scheduling Algorithm in Cloud Computing - Simulated Analysis

Senthilkumar P Assistant professor, Department of Computer Science and engineering S.veerasamy chettiar college of engineering and technology ,puliangudi

S. Arogya Swarna Associate Professor & Head of the Department(PG) Department of Computer Science and engineering S.veerasamy chettiar college of engineering and technology ,puliangudi

ABSTRACT

The popularity of the Internet and the availability of computers and high speed network as low-cost commodities are changing the way we use the computers today. These technical opportunities have led to the possibility to provide anything as a service over the Internet. Cloud computing has gained popularity to provide this service to the users in recent time. To achieve this goal, a cloud must provide services to many users at the same time and as different users have different QoS requirements, the scheduling strategy should be developed for multiple workflows with different QoS requirements. Most of the algorithms developed for scheduling applications on cloud computing focus on a single Quality of Service (QoS) parameter such as execution time or cost or total data transmission time. Even in MQMW [1] algorithm used only two QOS parameters such as reducing the make span of workflow and cost. However, if we consider more than one QoS parameter in to single objective function then the problem becomes more challenging .In this paper we introduced Multiple Workflow QOS Scheduling algorithm to address this problem .The scheduler can schedule multiple workflows which are started at any time and the QoS requirements are taken into account. Experimentation shows that our scheduler is able check the Scalability, Reliability, Average Response Time.

KEYWORDS : Cloud Computing, Scheduling, Multiple QoS Requirements, Multiple Workflows

I. Introduction

Modern collaborative scientific experiments in domains such as structural biology, high-energy physics and neuroscience involve the use of distributed data sources. As a result, analysis of their datasets is represented and structured as scientific workflows [2]. These scientific workflows usually need to process huge amount of data and computationally intensive activities. A scientific workflow management system [3] is used for managing these scientific experiments by hiding the orchestration and integration details inherent while executing workflows on distributed resources provided by cloud service providers.

Clouds are a large pool of easily usable and accessible virtualised resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing also for an optimum resource utilisation. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customised service level agreements [4].

Cloud computing is not purely a new technology. It is a combination of grid computing, virtualisation, utility computing.. This fact makes grid computing and cloud computing closely related to each other on one hand, yet different to each other on the other hand.

Cloud computing helps user applications dynamically provision as many compute resources at specified locations (currently US east1a-d for Amazon1) as and when required. Also, applications can choose the storage locations to host their data (Amazon S32) at global locations. In order to efficiently and cost effectively schedule the tasks and data of applications onto these cloud computing environments, schedulers have different QoS policies that vary according to the objective function: minimize total execution time (makspan), minimize total cost to execute, high scalability, reliability and reducing the response time used while meeting the deadline constraints of the application, and so forth. In this paper, we focus on providing high reliability, availability, decreasing average response time and BW utilization minimizing the total execution cost of applications on these resources provided by Cloud service providers, such as Amazon and GoGrid3. We achieve this by a method called Compound QoS Constraint workflow scheduling algorithm (CQCW).

The rest of the paper is structured as follows: Related work is discussed in Section 2. Then section 3 describes our research tool. The scheduling strategy will be presented in Section 4. And Section 5 will show the experimental details and simulation results. Finally Section 6 concludes.

II. RELATED WORK

In this section, we will describe the related work of workflow scheduling. Scheduling in distributed systems is NP-complete in general. Some typical grid workflow scheduling algorithms are introduced in [5] by Jia Yu and Buyya. As we take a look at the scheduling algorithms they use, it is possible to see that many practical algorithms are designed for single workflow and do not consider the QoS requirements, such as the Min-Min Heuristic, the Greedy Randomized Adaptive Search Procedure algorithm [6] and the Heterogeneous Earliest-Finish-Time algorithm [7]. Jia Yu introduces a Cost-based scheduling algorithm [6] which partitions a workflow and assign deadline to every partition. However the algorithm is designed for single workflow. Though it can be used to schedule multiple workflows, the relationship of workflows is not taken into account.

Ke Liu et al. proposed a throughput maximization strategy for scheduling transaction intensive workflows [8]. But it is designed for transaction intensive workflows not for multiple workflows. The difference between transaction intensive and multiple workflows is that transaction intensive workflows are multiple instances of one workflow and multiple workflows are different workflows. In other word, multiple workflows may have completely different structures and transaction intensive workflows have the same structure. As multiple workflows may have thoroughly different structure, this strategy would not perform well.

Zhifeng Yu and Weisong Shi [9] present a planner-guided strategy for multiple workflows. It ranks all ready tasks and decides which task should be scheduled first. However, if there are new lower rank workflows coming continuously, the higher rank task will not be scheduled to execute. In a massive scalable cloud, this situation will become true. On the other hand, this algorithm only considers the execution time, not other QoS requirements, such as cost.

Meng Xu, Lizhen Cui et al[1]. Proposed a multiple QoS constraint strategies of multiple workflows for cloud computing. It provides the solutions of providing different QoS to different users. The experimental simulator results shows that they consider only three QoS parame-

Volume-5, Issue-2, Feb - 2016 • ISSN No 2277 - 8160

ters such as Time, Cost and Success rate of the scheduling.

In conclusion, although the algorithms mentioned above have their respective benefits in their particular application areas, none of them is particularly designed for multiple workflows with multiple QoS constrained scheduling (i.e not consider reliability, and response time) and the key features of cloud computing are not specifically taken into consideration. Thus, the motivation of our work is to design a strategy to decrease the response time scheduling and increase the scalability and reliability of workflow scheduling for cloud computing platform.

III.MOTIVATION

In this section, we discuss the motivation of our research. There have been extensive comparative studies for static and dynamic algorithms [8], [9], [10], [11] in the context of scheduling single workflow. Cloud is massively scalable. This leads to a fact that the services in cloud may change dynamically.

In this case, the static scheduling algorithm is not a optimal choice. Moreover, all application on cloud will be used by a large number of users. And different user has different QoS requirements naturally. So a challenge for workflow systems on cloud is how to satisfy with multiple QoS requirements of different users.

However, most existing scheduling algorithms are designed for scheduling of a single complicated workflow instance or only one QoS constraints rather than for multiple QoS constraints of multi-workflows. To address this problem, we need to develop new solutions.

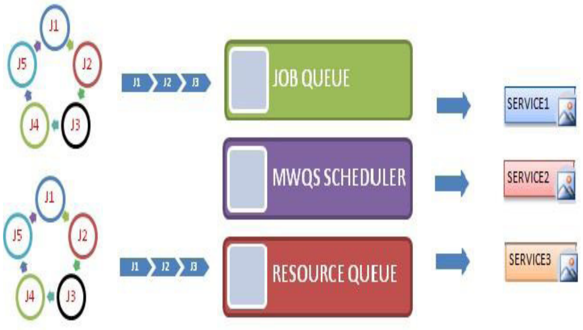
The new algorithms must be able to support the scheduling of multiple workflows, and must be able to meet multiple QoS requirements. Therefore, the new algorithms should not only consider the completion time of each single workflow, but most importantly, the overall performance.

The overall performance can be defined in many aspects. Among them, we focus on the following aspects: Scalability, Reliability and Average Response time with respect to QoS.

IV.THE PROPOSED MWQS ALGORITHM

Workflow Management System

Figure shows the architecture of the workflow management system. Users first submit workflow specifications with their QoS requirements. The system then Discovers appropriate services for processing the workflow tasks and schedules the tasks on the services. Distribute users' overall deadline into every task partition. Schedule the task to particular service Start workflow execution



B. Definition

Before we introduce our scheduling strategy, we first define some term which we will use later.

Definition 1: The Time Quota
The time limit when the task is executed

$$TQ(ti)=Min(T(ti,R))+VT(ti,R)*(QoS(time)-MinT(w))/\Sigma VT(t,R)$$

Where Min(T(ti,R)) is the minimum time of the task ti executed, VT(ti,R) is the time variance which the task ti executed on all services, QoS(time) is the time attribute of QoS requirements, MinT(w) is the minimum time of the whole workflow executed, and

ΣVT(t,R) is the time variance sum of all tasks in the workflow.

Compound QoS Constraint workflow scheduling algorithm (CQCW)in Cloud Computing

In our algorithm directly apply in our workflow management system .Workflows are getting from the job queue, in a job queue maintain a all workflow jobs .In our algorithm sorted the task in according to their attributes.

The number of services in cloud Environment. The number of tasks waiting to be executed is larger than the service number. So a task with minimum available service number should be scheduled first. The reason is that the task would have not available services, if other tasks are scheduled first

The tasks which belong to the workflow with minimum time should be scheduled first.

T: a set of all tasks
S: a set of available services

Procedure: Get Ready Task

```
getReadyTasks()
{while X (t) ε T is ready do ReadyTasks← t
end while}
```

Procedure : Sort The Task

```
sortTask(ReadyTasks, q)
{while X( t) .Ready Tasks insertTask(t,q) end
while }
```

Procedure: Insert The Task

```
insertTask(t,q)
{ insert task t into queue q according to the
strategy introduced above }
```

Procedure: schedule for Task

```
schedule(q,S)
{while q is not empty
t←first task in q
s←getService(t,S)
schedule task t on service s
 q=q-{t}
 S=S-{r} end while}
```

Procedure: Get The Service

```
get Service(t,S)
{select s.S, and the execution time and cost not
exceed the time and cost quota of task t return s }
```

No of Work flows	Total Execution Time							
Test	1	2	3	4	5	6	7	8
10	141	140	141	142	140	141	142	140
20	155	156	155	156	157	156	157	156

V. EXPERIMENTS AND RESULTS

To evaluate the performance of the algorithm, we have developed an experimental simulator. Our cloud environment has 15 services. Every service can execute one task at the same time. We list some of results for MWQS algorithm.

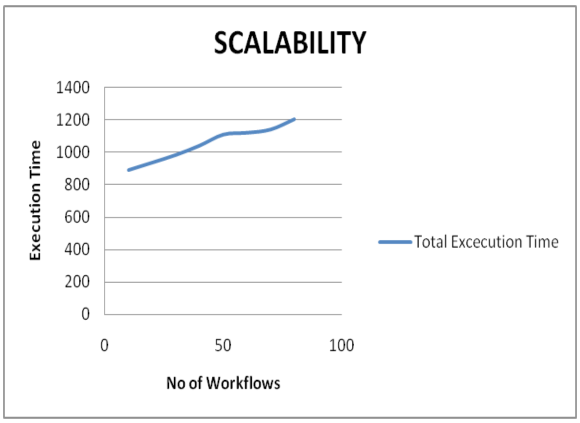
In table 1 shown in result of scalability using MWQS algorithm, in this system increase the number of workflows does not affect execution workflows

In table 2 shown in result of Reliability using MWQS algorithm. We list the 8 times execution of MWQS algorithm with 10and 20 workflows .the result is variation of one or two milliseconds

In table 3 shown in result of Response time calculation using MWQS algorithm. We list 5 results and get the average response time of system

No of Workflows	Total Execution Time
10	891
20	937
30	984
40	1041
50	1109
60	1120
70	1140
80	1203

TABLE 1



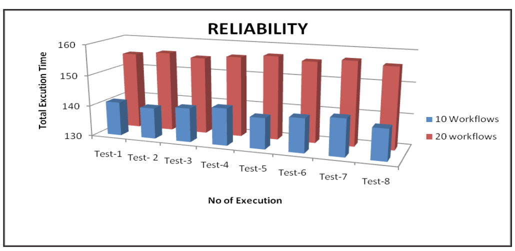


Figure 3

No of Workflows	Response Time(ms)	Total Execution Time(sec)
10	2.234	1644.2
20	2.235	1648.2
30	2.24	1651.1
40	2.355	1654
50	2.375	1679
Average	2.2878	1655.3

TABLE 3

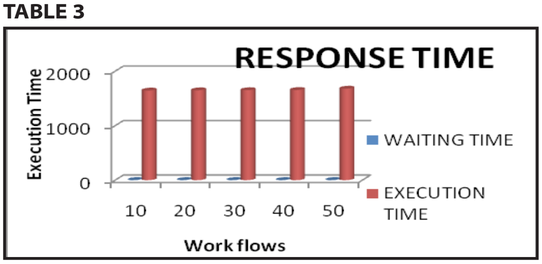


Figure 4

VI .CONCULUSION

The workflows on cloud computing platform have multiple QoS requirements. It is a main challenge of cloud workflow system to scheduling the multiple QoS constrained workflows. However, most existing scheduling algorithms are not designed to address this problem. To address this problem, we have proposed a multiple QoS constrained scheduling strategy of multiple workflows for cloud computing.

In future work, we plan to add more QoS constrained (availability etc) to workflows.

Reference

[1] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi: A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing, 2009 (IEEE International Symposium on Parallel and Distributed Processing with Applications)

[2] H. Topcuouglu , S. Hariri and M. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing ,"IEEE Transactions on Parallel and Distribution Systems, vol. 13, no. 3, pp. 260–274, 2002.

[3] S. Pandey, W. Voorsluys, M. Rahman, R. Buyya, J. Dobson, and K. Chiu. A grid workflow environment for brain imaging analysis on distributed systems. *Concurrency and Computation: Practice & Experience*, 21(16):2118–2139, November 2009.

[4] Deelman E. Mapping abstract complex workflows onto Grid environments. Journal of Grid Computing 2003; 1:25–39.

[5] Jia Yu, Rajkumar Buyya and Chen Khong Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids", In 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, Dec. 5-8, 2005.

[6] R. Sakellariou and H. Zhao, A Hybrid Heuristic for DAG Scheduling on Hetero-geneous Systems, The 13th Heterogeneous Computing Workshop (HCW 2004), Santa Fe, New Mexico, USA, April 26, 2004.

[7] G. C. Fox and D. Gannon, "Workflow in Grid Systems", Concurrency and Computation: Practice and Experience, vol. 18, no. 10, 2006, pp. 1009-1019.

[8] A. Mandal et al., "Scheduling strategies for mapping application workflows onto the grid," in Proceeding of the 14th International Symposium on High Performance Distributed Computing (HPDC'05). IEEE, 2005, pp. 125–134.

[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.

[10] 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.

[11] 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, 2006, pp. 199–207.

[12] J. Blythe et al., "Task scheduling strategies for workflow-based applications in grids," in Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05), Cardiff, UK, 2005.

[13]. L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, A Break in the Clouds: Towards a Cloud Definition, ACM SIGCOMM Computer Communication Review, 39(1), 50-55, December 2008.

[14] J. Yu and R. Buyya, "Workflow Scheduling Algorithms for Grid Computing", Metaheuristics for Scheduling in Distributed Computing Environments, F. X. a. A. Abraham, ed., Springer, 2008.