

Association for Information Systems AIS Electronic Library (AISeL)

PACIS 2014 Proceedings

Pacific Asia Conference on Information Systems
(PACIS)

2014

MIN-COST WITH DELAY SCHEDULING FOR LARGE SCALE CLOUD-BASED WORKFLOW APPLICATIONS PLATFORM

Cheng-Wei Yang

Shandong University of Finance and Economic, yangchengwei2006@163.com

Ji-Dong Guo

Shandong University of Finance and Economic, gjd730210@163.com

Jing Chi*

Shandong University of Finance and Economic, peace_world_cj@126.com

Follow this and additional works at: <http://aisel.aisnet.org/pacis2014>

Recommended Citation

Yang, Cheng-Wei; Guo, Ji-Dong; and Chi*, Jing, "MIN-COST WITH DELAY SCHEDULING FOR LARGE SCALE CLOUD-BASED WORKFLOW APPLICATIONS PLATFORM" (2014). *PACIS 2014 Proceedings*. 359.

<http://aisel.aisnet.org/pacis2014/359>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2014 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

MIN-COST WITH DELAY SCHEDULING FOR LARGE SCALE CLOUD-BASED WORKFLOW APPLICATIONS PLATFORM

Cheng-Wei Yang, College of Management Science and Engineering, Shandong University of Finance and Economic, Shandong Provincial Key Laboratory of Software Engineering, Jinan, China, yangchengwei2006@163.com

Ji-Dong Guo, College of Management Science and Engineering, Shandong University of Finance and Economic, Jinan, China, gjd730210@163.com

Jing Chi*, College of Computer Science and Technology, Shandong University of Finance and Economic, Jinan, China, peace_world_cj@126.com

Abstract

Cloud computing is a promising solution to provide the resource scalability dynamically. In order to support large scale workflow applications, we present Nuts-LSWAP which is implementation for Cloud workflow. Then, a novel Min-cost with delay scheduling algorithm is presented in this paper. We also focuses on the global scheduling including genetic evolution method and other scheduling methods (sequence and greedy) to evaluate and decrease the execution cost. Finally, three primary experiments divided into two parts. One parts of experiment demonstrate the global mapping algorithm effectively, and the second parts compare execution of a large scale workflow instances with or without delay scheduling. It is primarily proved the Nuts-LSWAP is efficient platform for building Cloud workflow environment.

Keywords: task scheduling, cloud computing, workflow.

1 INTRODUCTION

The large scale workflow applications common mean instances intensive applications which are those processes which need to be executed for a large number of times sequentially within a very short period or concurrently with a large number of instances(Matei Zaharia, et al., 2010). Therefore, WfMS needs to process and manage a large amount of data, which require the use of distributed collection of computation and storage facilities. However, such an issue in WfMS has not been well addressed by current computing paradigms. Cloud computing provides different services in each level. Infrastructure level virtualization enables providers offer virtual hardware for computing or data intensive workflow application. Several evaluations have shown that better performance can be achieved at lower cost using Clouds and Cloud technology than based on previous technologies. Since resources are self-contained and organized in a heterogeneous way, resource scalability especially in heavier or lighter workloads is very poor. The difficult issue is that most of resources are idle at most of time, only the peak periods need the busting resource requirement. Dynamically-scalable and managed resources become the critical problem for large scale workflow application.

In our research group, a SaaS based platform that is called Nuts-Platform has been developed and deployed for the application of Textile Industry in September, 2010. Now, Nuts-Platform provides several kinds of SaaS services. Therefore, the final purpose of this research is to present a completed Cloud workflow system infrastructure for non big manufacture enterprises and institute. The project extends the Nuts-Platform to build private Clouds and hybrid Clouds for large scale workflow applications in manufactures. This project is called “Nuts-LSWAP”. (Chengwei Yang, et al., 2011)This plan is that the first step extends the Nuts-platform to solve the collaboration of SaaS applications in Nuts-platform. This step has been accomplished. In order to support efficient scheduling for large scale workflow applications, the second step we have proposed a new algorithm and policies on Nuts-LSWAP. The business process can be arranged and the service can be scheduled by this common workflow engine. The volunteer resources can be utilized to implement the allocated tasks. When the resources are enough for the user, the dynamic resource pool will allocated the resources which are transparent to the users.. The next step is building the completed Nuts-LSWAP and makes the platform harness the private and public Cloud resource. Furthermore, according to the actual manufacture background, we also combine each step with the industry application for the experiment. This paper focuses on the second step of Nuts-LSWAP.

Cloud computing is a promising solution to provide the resource scalability dynamically. When we began building the data centers, it found the data consolidation provided by a shared cluster highly beneficial. However, when we found the response times starts to suffer due to Hadoop’s FIFO scheduler(Matei Zaharia, et al., 2010).. Nuts-LSWAP was unacceptable to execute the large scale instance intensive tasks. So as to enhance the system resource utilization, We propose a new solution that is called TPWSA in section four. In initial phase, it provides the global users’ QoS guarantee ability. Then, at the execution phase, the scheduler can be real-time perception the situation of service resource according the system loading, and determine the best opportunity to perform the scheduling.

The detailed introduction is described later. The novelty of research-in-progress on this solution lies in the following aspects.

(1) A novel Min-cost with delay scheduling algorithm is presented to facilitate the requirements of large scale workflow applications. with the mainly extension of external commercial Cloud service providers.

(2) Besides some features of scheduling model, this paper focuses on the genetic evolution method and other scheduling methods (sequence and greedy) to evaluate and decrease the execution cost.

(3) Three primary experiments divided into two parts. One parts of experiment is carried out demonstrate the global mapping algorithm effectively and the results for execution of a large scale workflow instances with or without delay scheduling.

(4) This scheduling has been applied to promote the effectiveness of Nuts-LSWAP.

The remainder of the paper is organized as follows. The next section briefly analyzes the motivation and current task execution process of Nuts-LSWAP. The issues for building Cloud manufacture environment to facilitate large scale workflow applications and the requirements are summarized too. Section 3 proposes a more detailed cloud workflow scheduling algorithm, which is called Min-cost with delay scheduling algorithm. The core design and implementation are given in this section. Delay-based on load optimization scheduling model of cloud workflow is presented. Section 4 illustrates the primary experimental results. In section 5, we list representative works as comparison from three phases. Finally, section 6 concludes the paper and overlooks the future research.

2 MOTIVATION AND TASK EXECUTION PROCESS

2.1 Motivation of Large Scale Cloud Workflow Platform, LSWAP

In this section, we present the motivation for Nuts-LSWAP to facilitate large scale workflow applications and the requirements for workflow system. We start review existing solutions for workflow applications and the limitation with respect to scalability and on-demand access. Large scale workflow uses the distributed collection of computation and storage facilities. Therefore, resource scheduling is becoming a critical problem for current workflow system. The Cloud resources allow real-time provisioning of resources to adapt the application execution at runtime. Furthermore, the third benefit which Cloud computing provides is the using of virtualized resources for workflow execution. Opposed to having direct access to the physical resource (e.g. computing, storage or bandwidth), Virtualized Machine (VM) technologies share resource “slice” of physical machine, which are widely reduced the need for securing and harness the resources effectively. In short, for the large scale workflow applications, the Cloud computing environment and technologies will be respected to achieve the better performance at lower cost than based on previous computing paradigms.

2.2 Tasks Execution Process of Cloud Workflow

In Nuts-LSWAP, the application scenario is much the same as it. The data would take the form of a set of files, including the application binaries. This data can be uploaded by the user prior to execution, and stored in storage facilities offered by cloud services for future use. After the process ends, these result files are then staged out to the remote storage server so that they are accessible by other tasks in the workflow managed by the WfMS.

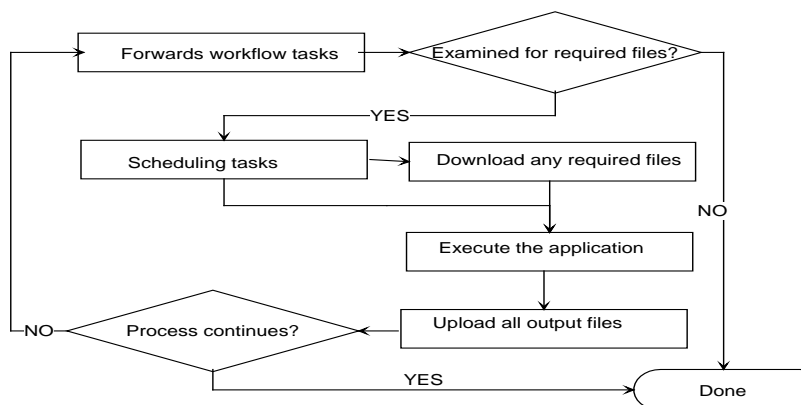


Figure 1. Tasks execution process on cloud workflow platform

A task execution is a single unit of work processed in a node. It is independent from other tasks that may be executed on the same or any other node at the same time. It is also atomic, in the sense that it either executes successfully or fails to produce any meaningful result.

Figure 1 shows the Cloud Workflow Application (CWA) at the execution phase. The process as follows: First of all, the WfMS forwards workflow tasks to the scheduler via the Web Service interface. Secondly, these tasks are subsequently examined for required files and the Storage Service is instructed to stage them in from the remote storage server. The execution begins by scheduling tasks to available execution nodes (also known as worker nodes). The workers download any required files for each task they execute from the storage server. Then the application is executed. Finally, all output files as a result of the execution is uploaded back to the storage server. This process continues until the CWA is completed.

3 SCHEDULING MODEL AND ALGORITHM DESIGN

3.1 Delay-based on load optimization scheduling model

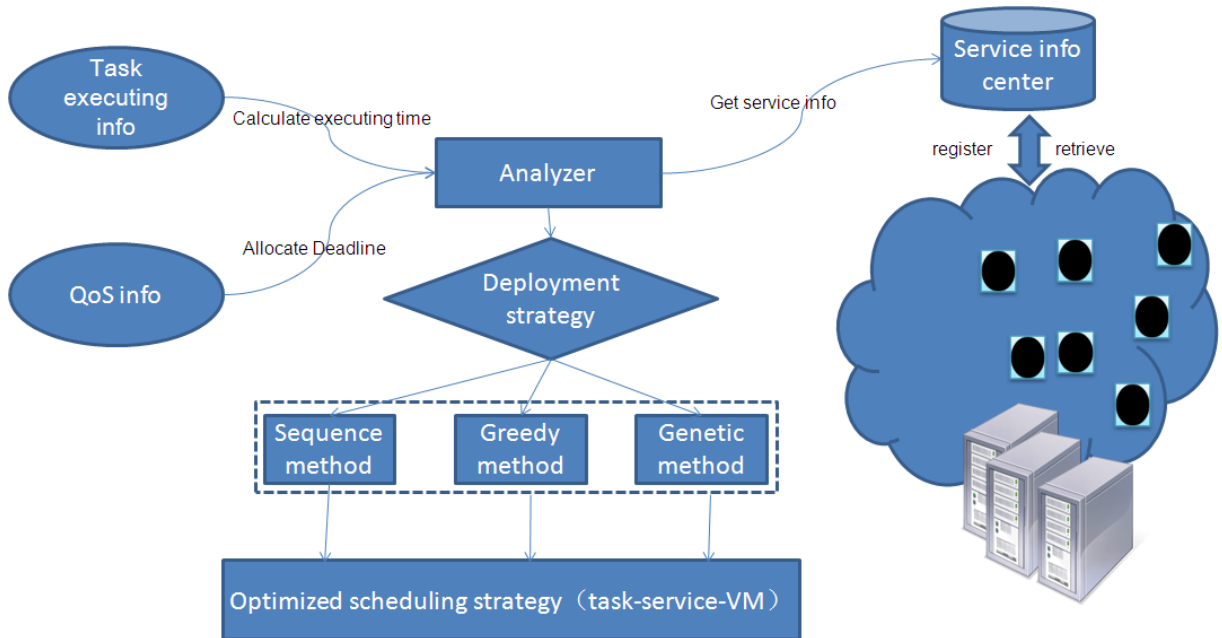


Figure 2. Delay-based on load optimization scheduling model

As shown in Figure 2, delay-based on load optimization scheduling model is mainly composed of three core parts, including information service centre, analyzer, and the scheduler. Information service centre takes charge of managing the resources and services under cloud environment. The main functions include registration service and retrieve service. Registration service is to encapsulate computing and storage resources through virtualization technology and register the interface information and service functionality information. LSWAP adopts service bus technology to manage service resources. Retrieval services get the required service resources and objects through user's service matching. Retrieval service is a reverse process of lookup service resources. First of all, it must be used the intelligent recommendation algorithm to get the adapted service information, which can quickly find the required service in the information center.

Analyzer is mainly responsible for the workflow task priority scheduling according to user's QoS constraints, select the best scheduling strategies, and determine the mapping between task allocation and distributed resource. Hence, the design of analyzer guarantees the performance of global optimization scheduling. In the phase of local execution, scheduler gets the current load of the available resources, and computing the current scheduling cost. So as to reduce the cost, we use delay

scheduling algorithm to determine the scheduling opportunity. Delay scheduling can ensure that the current service can make full use of resources, improve resource utilization, and reduce the probability of failure scheduling.

3.2 Genetic based Min-cost Loading Mapping

Genetic evolution algorithm is through the fitness function and evolution for the next generation of selection. We assign a task through a two-step process: First, we create a sorted list of jobs according to our hierarchical scheduling policy. Second, we scan down this list to find a job to launch a task from, applying delay scheduling to skip jobs that do not have data on the node being assigned for a limited time. It is to begin to execute the current schedule instance or wait for the next through analyzing scheduling cost, which can be calculated in the creation of instance. We assume cloud workflow jobs $J = \{j_1, j_2, \dots, j_n\}$, n is the number of jobs, set $G = \{g_1, g_2, \dots, g_m\}$ is the execution nodes, $T = [t_i]$ is the execution time matrix. Let $P = \{p_1, p_2, \dots, p_M\}$ denote the price set, $D = \{d_1, d_2, \dots, d_n\}$ is the deadline set.

This algorithm is shown below:

Algorithm 1: Genetic based Min-cost Loading Mapping

Input: Integrated Task-to-VM list Tasks ; Virtual machines(VM_j,Price_j,Speed_j)

Output: Optimised task-level scheduling plan

Get ready status tasks list of cloudlets and VMs

do genetic encoding and init Population of fixed size;

for condition of convergence is not met **do**

select the next pops;

calculate the fitness value ;

do crossover pops and setting the crossover rate equals 25%

execute mutation pops operation;

replacement pops before;

end for

record the fitness result;

deployment of the tasks ;

The fitness function directly affects the convergence speed. Based on the definition of total execution cost, the fitness function can be defined(Xiao Liu, et al., 2010):

$$f(i) = \sum (t_{ij}p_j + \max(0, l_i(s_i + t_{ij} - d_i)))$$

In order to select the next generation individual, first of all c probability must be choose P as the probability of individual selection. By this selection, individuals in a population with shorter completion time will be evolve as the next generation of more excellent individual, which can provide good genes. So, we set $P_i(i)$ is:

$$P_i(i) = f(i) / \sum_{j=1}^{SCALE} f_i(j)$$

Crossover mutation imitates natural biological principle of genetic recombination. It will pass the good genes to the next generation, and generate a new individual. We assume the biggest fitness value is f_{max} . Then, in order to cross the larger of the two individual fitness value, we set k_1, k_2 are individual of fitness value of crossover probability and mutation probability.

$$P_{cross} = k_1(f_{max} - f') / (f_{max} - f_{avg})$$

$$P_{mutation} = k_2(f_{max} - f') / (f_{max} - f_{avg})$$

In order to compare, the sequence strategy and greedy strategy have been implemented for experiment in section 4.

3.3 Delay scheduling opportunity

In this section, we consider a new version of delay scheduling which is called Mini-Cost with Delay Scheduling. Pseudocode for this algorithm is shown below:

Algorithm 2: Mini-Cost with Delay Scheduling

- 1) For all the ready scheduling instance t_i {
 - 2) For all the task requests from ready scheduling instance {
 - 3) For all the ready scheduling machine m {
 - 4) Compute value of F_M^{start} , F_M^{wait} and $\Delta F_{r,M}$
 - 5) If ($\Delta F_{r,M} < 0$) {
 - 6) $\min_{r,M} = \{r, M\}$;
 - 6) If ($F_{r,M}^{penalty} < minCost$) {
 - 7) $minCost = F_{r,M}^{penalty}$;
 - }
 - 8) Assign task request to minimumCost VM;
 - }
-

Note that the core of this algorithm is to determine min-cost scheduling under system loading. We have explained the rationale for this design in our analysis of delay scheduling. Therefore, we give the formula directly. Some detail description may be found (Nieolas G, et al., 2009).

At the time t_i , set scheduling cost of a request r^* is :

$$F_M^{start} = \sum_{r \in MU\{r^*\}} F_{w(r)} \left(\frac{f_{w(r)}^{\wedge} + \Delta t^{start} \left(\frac{1}{e^{start}} - \frac{1}{e^{wait}} \right) - d_{w(r)}}{d_{w(r)} - b_{w(r)}} \right)$$

At the time t_i , set delay scheduling cost of a request r^* is:

$$F_M^{wait} = F_{w(r^*)} \left(\frac{f_{w(r^*)}^{\wedge} + \Delta t^{wait} - d_{w(r^*)}}{d_{w(r^*)} - b_{w(r^*)}} \right) + \sum_{r \in M} F_{w(r)} \left(\frac{f_{w(r)}^{\wedge} - d_{w(r)}}{d_{w(r)} - b_{w(r)}} \right)$$

For any request r and candidate machine M , the total scheduling cost is :

$$\Delta F_{r,M} = F_M^{\text{start}} - F_M^{\text{wait}}$$

As a result, when the $\Delta F_{r,M} \geq 0$, the request r is delay scheduled, or if $\Delta F_{r,M} < 0$, the request will be scheduled at once.

4 EXPERIMENTAL EVALUATION AND RESULTS

4.1 Experimental environment

In this section, two kinds of primary experiments are made to evaluate the algorithm strategies that proposed in first phase and the system performance evaluation with Delay based on load optimization in the second phase. The deployment environment is depicted in Table 1:

Characteristics	Master/ Worker	Workflow Engine
OS	Windows 2000 Server	Windows 2000 Server
Runtime platform	Nuts-platform	Nuts-platform
Database persistence	MySQL 5.5	MySQL 5.5
CPU	P8700 2.53GHz	P8700 2.53GHz
Memory	2 GB	4 GB
Instance storage	80 GB	120 GB
Number of instances	3	1
Application Server	Microsoft .Net Framework 2.0	Tomcat 6.0
Bandwidth	100Mbit/s	100Mbit/s

Table 1. Parts of computing resources used in our experiments

4.2 Algorithm Comparison and Evaluation

In the simulation experiment, we makes a preliminary experiment, the experiment algorithm selected three kinds of typical scheduling strategies, including sequential scheduling method, greedy scheduling method, and genetic scheduling method. We let MI (Million Instruction) denotes the length of task and the MIPS (Million Instruction Per Second) denotes the capacity of VM requesting service.

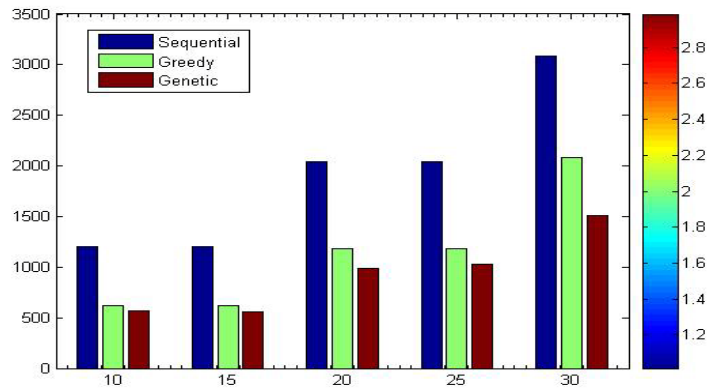


Figure 3. Algorithm Comparison Result with Service Quantity Unchanged

As shown in Figure 3, we let the service resources quantity are 5, the workflow tasks number are 10,15,20,25, 30 cases. We select the proposed scheduling strategy and the genetic algorithm to get the best performance. At the same time, the greedy scheduling algorithm and genetic scheduling algorithm is very close on the condition of task number less than 30. However, with the increase of the numbers of task, the efficiency of genetic scheduling optimization is better than others.

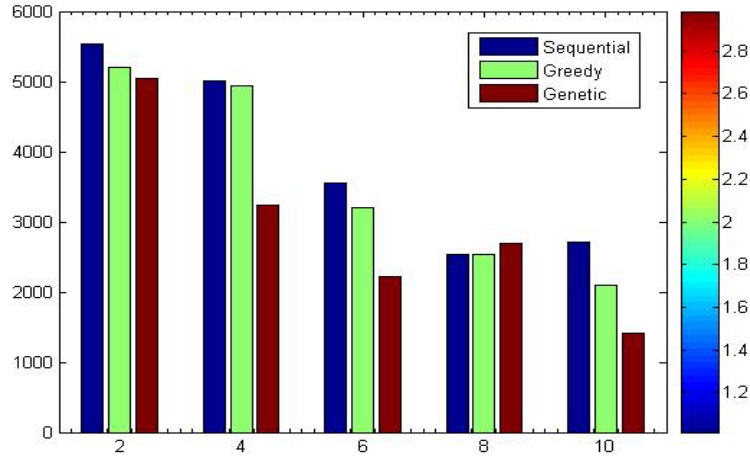


Figure 4. Algorithm Comparison Result with Tasks Number Unchanged

As shown in Figure 4, we let the tasks number are 50, service resources quantity are 10,15,20,25, 30 cases. While increasing the number of resources, the genetic algorithm to optimize scheduling strategy has obvious advantages. This shows the similar results with the previous experiment. However, when the number of service resources are 8, using genetic scheduling algorithm, the resource consumption slightly more than the other two algorithms at the conditions of service resource unchanged. The reason for this phenomenon also shows the uncertainty weaknesses for the genetic algorithm.

In order to compare and evaluate the execution effectively influence on instances intensive jobs, we design three kinds of primary experiments are made to compare the workflow jobs executed time between cloud workflow environment without delay scheduling and cloud workflow environment with Min-Cost delay scheduling. The application scenario demonstrates as follows:

Experiment 1: 1000 Workflow Serial Tasks Were Executed. In this experiment, we start executing 1000 serial workflow tasks in Cloud workflow environment with one master node and two worker nodes. After the 1000 instances intensive jobs (eg. an complicated computing task) ends, the elapsed time is recorded and illustrated in Figure 5.

Experiment 2: 1000 Workflow Parallel Tasks Were Executed. In this experiment, we start executing 1000 parallel workflow tasks in Cloud workflow environment with one master node and two worker nodes. After the 1000 instances intensive jobs (eg. an complicated computing task) ends, the elapsed time is recorded and illustrated in Figure 5.

Execution 3: 1000 Workflow Hybrid Tasks (serial and parallel tasks) are executed. In this experiment, we start executing 1000 serial workflow tasks in Cloud workflow environment with one master node and two worker nodes. After the 1000 instances intensive jobs (eg. a complicated computing task) ends, the elapsed time is recorded and illustrated in Figure 5.

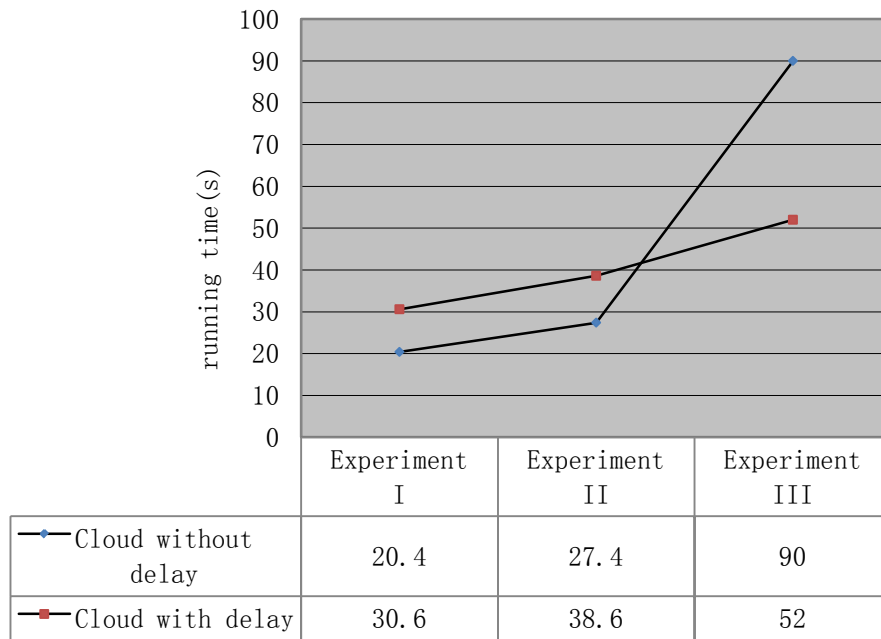


Figure 5. Workflow scheduling in cloud environment

In experiments, the initial workflow jobs are the same as 1000. But the types of workflow are differences in the three experiments. As shown in Figure5, the experimental results illustrate that in a instance intensive applications, the number of tasks and the task types have the influence on the execution effectively in cloud computing. Less quantity or singleness workflow tasks more adapt to run in cloud workflow environment without delay scheduling. However, the complicated and large scale instances intensive workflow tasks need to be scheduled and will be benefit of the Min-Cost delay scheduling algorithm that proposed in this paper.

5 RELATED WORKS

Management Systems for Cloud-Based Workflow: Paper(Ling Shang, et al., 2010)extends the YML to build science private Clouds and hybrid Clouds for non big enterprises and research institutes. This project is called by YML-PC. It presents a reference architecture based on YML for building science private Clouds. CloudWF is described by paper(Chen Zhang, et al., 2009), which is a scalable and lightweight computational workflow system for Cloud on top of Hadoop. It presents a simple workflow description language that encodes workflow blocks and block-to-block dependencies separately as standalone executable components: a new workflow storage method that uses Hadoop HBase sparse tables to store work. The works in paper(Xiao Liu, et al.,2010) shows the SwinDeW-C system, which is being conduct by faculty of Information and Communication Technologies, Swinburne University of Technology, Hawthorn, Melbourne, Australia. One of their working driving force is the demand of large scale instance and data/computation intensive workflow application. Paper(Suraj Pandey, etc al., 2011) propose a wider vision that incorporates an inter Cloud architecture and a market-oriented utility computing model, which is called Cloudbus (a solution for using Cloud infrastructure). The Cloudbus workflow engine scales workflow applications on Clouds using market-oriented computing. This Cloud workflow system comprises the workflow engine, a resource broker, and plug-ins for communicating with various technological platforms.

The above reference architectures based on workflow for building Clouds environment can be divided into three types: private Clouds workflow, public Clouds workflow, and hybrid Clouds workflow. Private Cloud workflows, such as(Ling Shang, et al., 2010; Chen Zhang, et al., 2009) are commonly built in libratory to provide the services to internal user who do not need use public for the issues of

security, custom, confidence, policy, law and so on. Hybrid Clouds workflow presents by (Xiao Liu, et al., 2010; Suraj Pandey, et al., 2011), shares the private and public Cloud resources, and become the trends of large scale workflow application. Therefore, the building aim of Nuts-LSWAP is a Hybrid Clouds workflow for the requirements of manufacture & industry information applications.

Dynamic Resource Management Strategies for Distributed Environment: Paper (Vincenzo D, et al., 2009) have been put forward the resource management method, including the user's QoS volunteer resources collection, dynamic resources provision, and service resources migration. When the QoS violation occurred, the engine will deal with conflict according to the configuration which is set up before. Resource management of YML system proposes in paper (Ling Shang, et al., 2010), which put forward the method of using volunteer resources for heterogeneous environment. If the resources provided by service providers are not able to satisfy the user's QoS requirements, the user can have the chance to replace the original service by selecting other services. Paper (Bowers, S, et al., 2005) proposes a new dynamic resource extension model. It has a negotiation mechanism among multiple third party service providers and automatic chooses the best matching resources. Paper (J. Yu and R. Buyya, et al., 2006) puts forward the service migration methods. When a server loading is overweight, it will move parts of the request to the other idle servers for the purpose of reducing the server pressure.

The above reference works meet the needs of users QoS in the running system. They all need to provide sufficient service resources for computing environment. However, the differences among them are the resource selection, allocation and recovery strategy.

Scheduling for Instance-Intensive Cluster Application: Paper (Jing Yan, et al., 2010) puts forward a instances of intensive tasks scheduling algorithm (MCUD). The algorithm specifies a deadline assigned to each task. So as to optimize addition execution time, it can dynamically adjust follow-up task scheduling time at the running time. Paper (Wenhao Li, et al., 2010) pays close attention to intensive scheduling problem of community cloud. In order to solve the problem of the cloud cooperate with community members. The USS -I scheduling strategy has put forward. This strategy consists of three components: cloud scheduling algorithm among members of the community (QCOAL), cloud scheduling algorithm in members of the community (QCPCSA) and QoS decomposition algorithm (QD). For the problem of instance intensive tasks scheduling, the preview of scheduling algorithm (PCSA) is presented in paper (Ke Liu, et al. 2008) Its main characteristic is the tasks are ranked by the degree of urgency. It will choose the idle computing resources to carry out, so that the execution results will be near-optimality.

The above reference scheduling method tackled the conflict between locality and fairness in scheduling. Paper (Jing Yan, et al., 2010; Ke Liu, et al. 2008) represents the scheduling problem as an optimization problem, which is very closet work with us. The difference is that tasks must be matched to nodes and different assignments have different cost based on locality and fairness also be concerned. Besides, delay scheduling is simpler than the other works, which makes it easy to use with scheduling policies other than fair sharing.

6 CONCLUSION AND FUTURE WORKS

With the emergence of Cloud computing paradigm, it provides a promising new solution for large scale sophisticated workflow applications. In order to realize the project of building Cloud workflow environment, we divided the project of "Nuts-LSWAP" into three steps. The paper focuses on the second step. Then, a novel Min-cost with delay scheduling algorithm is presented in this paper. We also focuses on the global scheduling including genetic evolution method and other scheduling methods (sequence and greedy) to evaluate and decrease the execution cost. Finally, three primary experiments divided into two parts. One parts of experiment is carried out demonstrate the global mapping algorithm effectively and the results for execution of a large scale workflow instances with or without delay scheduling. It is primarily promoted the Nuts-LSWAP is efficient platform for building Cloud workflow environment.

Since the aim of this paper discusses the scheduling policy of Nuts-LSWAP, some detailed research works such as loading balance policy, and QoS management & negotiation mechanism is out of this paper, which will be discussed as the next step work. Furthermore, the scalable resource and decentralized management will be further discussed in Nuts-LSWAP.

Acknowledgement

This paper is supported in part by National Natural Science Foundation of China under Grant 61303088, and part by A Project of Shandong Province Higher Educational Science and Technology Program under Grant J14LN19, and part by the Fundamental Research Funds for Shandong Provincial Key Laboratory of Software Engineering under Grant 2013SE05, The second author is the corresponding author, and his e-mail address is peace_world_cj@126.com.

References

- Andrzejak A, Kondo D, Anderson DP (2010) Exploiting non-dedicated resources for Cloud computing. In the 12th IEEE/IFIP (NOMS 2010), Osaka, Japan, 19–23 April 2010.
- Bowers, S, Ludaescher, B. Actor-Oriented Design of Scientific Workflows. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, O, LNCS, vol. 3716, pp. 369 – 384, 2005.
- Bowers, S., Ludaescher, B.: Actor-Oriented Design of Scientific Workflows. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, O. (eds.) ER 2005. LNCS, vol. 3716, pp. 369 – 384. Springer, Heidelberg (2005)
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599–616.
- Chen Zhang, Hans De Sterck. CloudWF: A computational workflow system for Clouds based on hadoop. CloudCom 2009, LNCS 5931, pp.393-404, 2009.
- Chengwei Yang, Lu Wang, Chenglei Yang, Shijun Liu, Xiangxu Meng, The Personalized Service Customization Based on Multimedia Resources in Digital Museum Grid, The 3rd International Conference on U-media , Zhejiang Normal University, China, June, 298-304 2010.
- Chengwei Yang, Lei Wu, Shijun Liu, Xiangxu Meng, Applying Service-oriented Composition Process in TPMS. The 3rd International Colloquium on Computing, Communication, Control, and Management, Yangzhou University, China, July, 298-304 2010.
- Chengwei Yang, Lei Wu, Shijun Liu, Xiangxu Meng*, LSWAP: A Large Scale Workflow Applications Platform for Cloud Environment. The 6th ChinaGrid Annual Conference ChinaGrid 2011. Dalian University of Technology, China, August 2011.
- Chengwei Yang, shijunLiu, Xiangxu Meng. The Application of Cloud Computing in Textile-order Service, JDCTA, Vol. 9, No. 3, pp. 304 ~ 309, 2011.
- Chen, J., & Yang, Y.. Temporal dependency based checkpoint selection for dynamic verification of temporal constraints in scientific workflow systems. *ACM Transactions on Software Engineering and Methodology*, to appear. Retrieved 1st February 2010.
- Chen, J., & Yang, Y. Adaptive selection of necessary and sufficient checkpoints for dynamic verification of temporal constraints in grid workflow systems. *ACM Transactions on Autonomous and Adaptive Systems*, 2(2), Article 6.
- Chen, J., & Yang, Y. Temporal dependency based checkpoint selection for dynamic verification of fixed-time constraints in grid workflow systems. *Proceedings of the 30th International Conference on Software Engineering (ICSE 2008)*, Leipzig, Germany, 141–150
- De Assuncao, M. D., di Costanzo, A., & Buyya, R. (2009). Evaluating the cost-benefit of using Cloud computing to extend the capacity of clusters. *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*, Garching, Germany, 1–10.
- David P Anderson, Gilles Fedak: the computational and storage potential of volunteer computing. *CCGRID 2006*, pp 73–80.

- Rui Wang, Chengwei Yang, Chenglei Yang, Xiangxu Meng. A XML-based Interface Customization Model in Digital Museum . IEEE Transactions on Edutainment III, LNCS 5940,190-202, 2009.
- J.Yu and R.Buyya. Scheduling Scientific Workflow Applications with Deadline and Budget Constraints using Genetic Algorithms, Scientific Programming, 14(3-4):217-230, IOS Press, Amsterdam, The Netherlands,2006.
- JingYan , GuikaiWu. Scheduling Algorithm for instance intensive workflow[J].Computer Applications.2010(11),2864-2866.
- JIAO YuChang, LIU ShiJun,Meng Xiangxu, Yang Chenglei, ASaaS: an approved Architecture for SaaS Service Composition, CIDE2008
- Jie Hou, Shijun Liu, Xiangxu Meng. “Research on the UI Integration Architecture of Service System”, International Conference on Communications, 2008.
- Ke Liu, Jinjun Chen, Yun Yang and Hai Jin. A Throughput Maximization Strategy Scheduling Transaction Intensive Workflows on SwinDeW-G Concurrency and Computation: Practice and Experience, pp.1807-1820, 2008.
- K.Liu, H.Jin, J.Chen, X.Liu, D.Yuan and Y Yang. 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,2010.
- Ludscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.,Tao, J., Zhao, Y.: Scientific Workflow Management and the Kepler System. In: Concurrency and Computation: Practice and Experience, vol. 18, pp. 1039–1065(2006)
- Ling Shang, Serge Petiton, Nahid Emad, and Xiaolin Yang. YML-PC: A Reference Architecture Based on Workflow for Building Scientific Private Clouds. pp. 247–252, 2010.
- Lei Wu and Chengwei Yang. A Solution of Manufacturing Resources Sharing in Cloud Computing Environment. CDVE 2010, LNCS 6240, pp. 247–252, 2010.
- Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma. Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling. EuroSys’10.
- Nieolas G. Grounds, JohnK.Antonio. "Cost Minimizing Scheduling of Workflows on a Cloud of Memory Managed Multicore Machines", CloudCom2009,LNCS5931, pp. 435-450 , 2009.
- Suraj Pandey, Dileban Karunamoorthy and Rajkumar Buyya. Cloud Computing: Principles and Paradigms. Workflow Engine for Clouds. 2011.
- Vincenzo D ,Cunsolo, Salvatore Distefano, Antonio Puliafito, Marco Scarpa: Cloud@ Home: bridging the gap between volunteer and Cloud computing. ICIC (1):423–432, 2009.
- Vincenzo D (2009) Cunsolo, Salvatore Distefano, Antonio Puliafito, Marco Scarpa: Cloud@ Home: bridging the gap between volunteer and Cloud computing. ICIC (1):423–432.
- Wenhao Li. Research on instance intensive workflow scheduling for community cloud[J]. Shandong University,2010.
- Xiao Liu, Dong Yuan, Gaofeng Zhang, Jinjun Chen, and Yun Yang .SwinDeW-C: A Peer-to-Peer Based Cloud Workflow System. B. Furht, A. Escalante (eds.), Handbook of Cloud Computing, 2010.