Research on Workflow Scheduling Algorithms in the Cloud

Congyang Chen[1,2], Jianxun Liu[1,2], Yiping Wen[1,2], and Jinjun Chen[1,2,3]

[1] Key Laboratory of Knowledge Processing and Networked Manufacture,
Hunan University of Science and Technology, Xiangtan, China
[2] School of Computer Science & Engineering,
Hunan University of Science and Technology, Xiangtan, China
[3] Faculty of Engineering and Information Technology,
University of Technology, Sydney, Australia
chencongyangmm@gmail.com

Abstract. Cloud computing owns merits of more efficiency and less cost in fields of information processing and service mode. Algorithms of workflow scheduling in the cloud can contribute to cutting cost and improving the quality of services, therefore, it has been a hot research topic. In this paper, the workflow technology in the cloud and the needs for cloud workflow scheduling are firstly introduced. Then, typical cloud workflow scheduling algorithms are analyzed and classified into three categories. In the end, typical cloud workflow scheduling research tools such as CloudSim, WorkflowSim and SwinFlow-Cloud are evaluated. Besides, we also analyze the existing problems of current workflow scheduling algorithm in the cloud and introduce the directions of the future research.

Keywords: Cloud computing, Workflow, Scheduling algorithm, CloudSim.

1 Introduction

With the promotion of the world's leading companies, cloud computing has achieved significant developments and applications in recent years. Cloud computing can be defined as the epitome of distributed computing, parallel computing, utility computing, pervasive computing and grid computing. We can obtain data storage which it is security, convenient, efficient, and huge amounts of computer power based on the Internet. Regarded as a distributed computing paradigm with rapid growth, it has shown some obvious difference with other. In this regard, the running environment is controlled by user/application program in cloud environment; always the jobs in user level will not be exposed to the scheduling system, the VM will be allocated to users or none. In brief, cloud computing environment provides three forms of services, they are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service(SaaS), which can offer the flexible payment services [1,2]. It can satisfy the various requires of services and bring the possibility of service innovations [3]. It also has made great strides in developing cloud computing simulation tools. Many cloud computing platforms are also implemented by some famous corporations.

F. Bian and Y. Xie (Eds.): PAS 2014, CCIS 495, pp. 35–48, 2014.

Now, a series of problems are exposed with the extensively developed and further used in cloud computing, of which the most outstanding one is how to reduce cost and meanwhile keep or improve the QoS, at last maximize the revenues. Workflow technology can support business process management and business process automation, and enhance flexibility of business process system and the ability of fitting for change. Therefore, it can be used to solve such problems. Generally speaking, by using workflow technology, the business process can be resolved to some manageable small events, and individually model and control the procedural semantics which describe constraint relation between activities. A workflow application can be described by a directed acyclic graph, the nodes denote tasks and the edges denote the internal dependency between activities. Single workflow always contains a series of tasks, and each task may communicate with other task because of the dependency.

A workflow management system (WfMS) can manage the definition and execution control of business processes. Workflow engine is the key component of a WfMS to manage workflow's execution, which can be used to perform workflow scheduling, data transmission and fault tolerance management. The aim of workflow scheduling is to assign appropriate task to the right resource at appropriate time. Data transmission is used for the communication between data resources, and fault tolerance management works when the execution of some task goes wrong. Workflow scheduling is the most critical for a WfMS, because an appropriate scheduling plan has a significant impact on performance of WfMS.

This paper focuses on the typical cloud workflow scheduling algorithms and typical tools for its research. The rest of the paper is organized as follows. Section 2 briefly introduces the state-of-art of workflow technology and its technical features in cloud. Section 3 analyzes various existing workflow scheduling algorithms and classified them into three categories. Section 4 evaluates typical research tools such as CloudSim, WorkflowSim and SwinFlow-Cloud. Section 5 analyzes the problems of current workflow scheduling algorithms in the cloud and proposes the improvement direction. Section 6 concludes the paper.

2 Workflow Technology in Cloud Environment

The promising cloud computing environment brings new challenges to the traditional workflow technology. The research on cloud workflow aims at deploying the business process specified by user automatically in cloud, so that can reduce computational expense and improve the quality of cloud services. It integrates the technical advantages of cloud computing and the workflow system, so regarded as one of optimization solutions for cloud computing, it has been obtained more and more attention and study.

Using workflow technology in the cloud can take advantage of various cloud services which facilitate execution of workflow greatly. Unlike other computing environments, the resource in cloud is shared and offered to user on demand, meanwhile, also can use the resource dynamically according to the estimated requirements of execution. It makes workflow system satisfy the request of applications more easily.

In addition cloud services are mostly expensive for computing, storage and bandwidth, etc. The business model of pay as you go also can reduce the cost of workflow execution [4, 5]. Therefore, optimization of the workflow execution in cloud computing has become one of the research hotspots in the research of workflow and cloud computing recently.

Generally speaking, it is necessary to map tasks to the appropriate execution resource by some workflow scheduling algorithms in cloud, which will directly influence the success rate of cloud workflow scheduling and the execution efficiency. Besides, unlike traditional workflow scheduling, cloud workflow scheduling should consider not only the optimal combination and utilization of the resources but also the constraint of time sequence and causation of each task to obtain the final result. As a consequence, the cloud workflow scheduling problem is usually a NP-hard problem. The implications of cloud workflow scheduling research are as follows:

- It can promote the user's QoS request of gratification. It not only promotes the user's gratification of workflow execution cost, but also attracts the users to use cloud services, thus help to achieve maximum profit.
- It can improve the resource utilization of cloud services provided by cloud service provider. By taking the characteristic of workflow instance into account, the resource utilization involved in workflow scheduling will be significantly improved.
- It can promote the development and application of cloud computing and workflow technology, especially in the areas of biomedicine, chemistry, gene expression data analysis, astrophysics and the instance-intensive applications such as e-commerce, etc.

3 Typical Cloud Workflow Scheduling Algorithms

This section makes the typical cloud workflow scheduling algorithms with principle, merit and demerit sorting and analysis according to the scheduling difference of existing cloud workflows. We can divide the algorithms into three categories: single-objective optimization algorithms, multi-objective optimization algorithms, and heuristic algorithms for scheduling strategy and heuristic algorithms used.

3.1 Single-Objective Optimization Workflow Scheduling Algorithms

Cloud Workflow Scheduling Algorithm Oriented to Dynamic Price Changes
Min Zheng et al. [9] proposed a cloud workflow algorithm based on dynamic planning to solve the scheduling overhead optimization of cloud workflow in dynamic resource prices environment. Firstly, they define the model of workflow, resource and the target of task. Then, the cloud workflow tasks are divided into groups. Next, using the dynamic algorithm to dispatch each task and making the task links of workflow get better result. After grouping, the overall deadline will be allocated to each task group, and sort them topologically. At last, the dynamic algorithm is used to dispatch each task, and that the lowest overhead scheduling scheme in certain time is calculated.

Instance-Intensive Cost-Constrained Workflows Scheduling Algorithm in a Cloud
Mukute et al. [10] proposed an algorithm based on Job Shop to solve the issue of dynamic scheduling in cloud computing with a special attention to the case of instance-intensive cost-constrained workflows. They first consider the classification and combinatorial optimization of the concurrent tasks which need specified resources with a certain number. Then they specify the priority for tasks and make user minimize the overall cost in the end.

Meeting Deadlines of Scientific Workflows with Tasks Replication
In order to address limitations of ignoring costs related to utilization of the infrastructure and he capacity of taking advantage of elastic infrastructures and other. Rodrigo N. Calheiros and Rajkumar Buyya [17] proposed an algorithm, called EIPR, which uses idle time of provisioned resources to replicate workflow tasks in order to mitigate effects of performance variation of resources so that the soft deadlines can be met. The experiment result showed that the EIPR algorithm increases the chance of deadlines being met and reduces the total execution time of workflow.

3.2 Multi-objective Optimization Workflow Scheduling Algorithms

Multi-objective Workflow Scheduling
Juan et al. proposed [11] a multi-objective workflow scheduling method called Multi-Objective Heterogeneous Earliest Finish Time (MOHEFT) for multi-objective optimization problem in Amazon EC2 Cloud which offer heterogeneous types of resources at different prices and with different performance. MOHEFT is a Pareto-based list scheduling heuristic that provides the user with a set of tradeoff optimal solutions from which the one that better suits the user requirements can be manually selected. Finally, the experiments revealed that MOHEFT was able to meet the constraints imposed by current commercial Clouds in terms of the maximum amount of instances.

Minimum Total Cost Under User-Designated Total Deadline Algorithm
Jing Yan et al. [8] proposed a scheduling algorithm, named Minimum Total Cost Under User-designated Total Deadline (MCUD), based on multiple instances. For the workflow instances of the same type, after classification, MCUD algorithm distributes the user-designated overall deadline into each task with a new distribution method. In addition MCUD algorithm adjusts the sub-deadline of successive task dynamically during the scheduling process. Instances of the same nature are given the sub-deadline distribution results of some difference, which can avoid the fierce competition of cheaper services and increase the efficiency of resource utilization.

Auto-Scaling to Minimize Cost and Meet Application Deadlines
Ming Mao [14] et al. proposed a new auto-scaling mechanism for deadline to avoid the faults which the traditional "auto-scaling" mechanisms only support simple resource utilization indicators and do not specifically consider both user performance requirements and budget concerns. What the auto-scaling mechanism they have

implemented is finishing all jobs by user specified deadlines in a cost-efficient way. The method based a monitor-control loop adapts to dynamic changes such as the workload bursting and delayed instance acquisitions. In experiments, it has shown great performance.

Scaling and Scheduling to Maximize Application Performance with Budget Constraints

Ming Mao [15] et al. proposed two auto-scaling mechanisms to solve the issue that how to maximize the return from the cloud investment. They have implemented two algorithms: the scheduling-first algorithm which distributes the application-wide budget to each individual job, determines the task scheduling plan first and then acquires the VMs, while the scaling-first algorithm determines the size and the cloud resources first and then schedules the workflow jobs on the acquired instances. The results show good tolerance to inaccurate parameters.

Compromise-Time-Cost Scheduling Algorithm

Liu Ke [6] et al. presented a novel compromised-time-cost scheduling algorithm which focus on the trade-off of time and cost throughout the scheduling process. They pay attention to the feature of cloud computing, the cost of execution and average execution time and make the trade-off dynamically under user preferences, to resolve the scheduling problem of instance-intensive cost-constrained workflows. The algorithm can be further decomposed into two sub-algorithms: CTC-MC (Compromised-Time-Cost algorithm Minimizing execution Cost) algorithm which minimizes the execution cost with user designated and CTC-MT (Compromised-Time-Cost algorithm Minimizing execution Time) algorithm which minimizes the execution time within user designated budget.

3.3 Heuristic Based Workflow Scheduling Algorithms

A Particle Swarm Optimization (PSO)-Based Heuristic for Scheduling Workflow Application in Cloud Computing Environments

In addition to optimizing execution time, the cost arising from data transfers between resources as well as execution costs must also be taken into account. Suraj Pandey et al. [12] proposed a particle swarm optimization (PSO) based scheduling heuristic for data intensive applications that take into account both computation cost and data transmission cost. They use the heuristic to minimize the total cost of execution of scientific application workflows on Cloud computing environments. They vary the communication cost between resources, the execution cost of compute resources and compare the results against "Best Resource Selection" (BRS) heuristic. The experiments show that PSO based task-resource mapping can achieve at least three times cost saving as compared to BRS based mapping.

Market-Oriented-Hierarchical Scheduling

Zhangjun Wu [7] et al. proposed a cloud workflow scheduling strategy based on intelligence algorithm and adaptation-aware of cloud services composition strategy which has been developed to scheduling the two-level cloud workflow tasks. The two-level schedulings are service level scheduling which selects suitable cloud service for task units called package-based scheduling and task level scheduling which allocates tasks to the virtual machine of data center dynamically. Service-level scheduling should satisfy the QoS constrain of each task and the dependency between tasks. The task-level scheduling research genetic algorithm, ant colony algorithm and particle swarm algorithm under QoS of each task and the total cost reduced. In brief, this strategy selects suitable cloud provider and books the service resource; allocates and optimizes tasks to virtual computing resource using intellectual algorithms in task-level.

Concurrent Level Based Workflow Scheduling Algorithm
Due to Deadline Bottom Level (DBL) hasn't considered the concurrence during the real executing process that cause much more shatter time. In order to solve such problem, Guangzhen Lu [13] et al. proposed a novel heuristic workflow scheduling algorithm CLWS, which it distributes task levels by their concurrence, and adopts the efficiency algorithm MDP to optimize the sequential tasks with time dependency. It not only can decrease the time pieces, but also can optimize the total executing cost. The experiments demonstrate that CLWS has better performance than DBL and Deadline Min-Cost.

Adaptive Workflow Scheduling
Mustafizur Rahman [16] et al. developed a hybrid heuristic that can effectively integrate most of the benefits of both heuristic and metaheuristic-based approaches to optimize execution cost and time as well as meet the user's requirements through an adaptive fashion. They proposed Adaptive Hybrid Heuristic scheduling algorithm, which is designed to first generate a task-to-service mapping with minimum execution cost using GA with user's budget and deadline as well as satisfying the service and data placement constraints specified by the user.

Table 1. The typical of clouding workflow algrithms

Scheduling Algorithm	Scheduling Method	Scheduling Parameters	Findings	Environment	Tools
Cloud workflow scheduling algorithm oriented to dynamic price changes	Batch /dependency mode	Execution cost, Deadline time	It decrease 5% cost compared to state space algorithm with considering the changing price	Cloud Environment	Java
Instance-intensive cost-constrained workflows scheduling algorithm	Batch /dependency mode	Resource Utilization, Total cost	The algorithm decrease the cost compared the built-in algorithm and has better performance	Cloud Environment	CloudSim

Table 1. (*continued*)

Enhanced IC-PCP with Replication (EIPR) algorithm	Dependency mode	Deadline and total cost	Reduce execution time by 59% compared to the IC-PCP algorithm	Cloud Environment	CloudSim
Multi-objective workflow scheduling	Batch /dependency mode	Execution cost, Total time	It saved 5% time compared to SPEA2* and HEFT with the cost decreased by half	Cloud Environment	Amazon EC2
Minimum Total Cost Under User-designated Total Deadline algorithm	Batch /dependency mode	Cost, Deadline time	It decrease 17% cost and 20% time compared to Deadline-MDP algorithm	Cloud Environment	CloudSim
Auto-Scaling to Minimize Cost and Meet Application Deadlines	Dependency mode	Deadlines, cost and lag of instance	Save from 9.8% to 40.4% compared to other approaches	Cloud Environment	Other
Scaling and Scheduling to Maximize Application Performance with Budget Constraints	Dependency mode	Budget , workload and job turnaround time	Reduce the job turnaround time by 9.6%-45.2% and good tolerance	Cloud Environment	CloudSim
Compromise-Cost-Time Algorithm	Batch mode	Cost, Time	Reduce the time and cost	Cloud Environment	SwinDeW-C
A Particle Swarm Optization (PSO)-based Heuristic for Scheduling	Dependency mode	Computational and data transmission cost	It saved three times cost compared to BRS and has good distribution workload	Cloud Environment	Amazon EC2

Table 1. (*continued*)

Market-Oriented-Hierarchical Scheduling	Virtual clusters	Total cost, CPU time	Minimize the total execution cost	Cloud Environment	CloudSim
Concurrent Level based Workflow Scheduling(CLWS)	Dependency mode	Cost with user designated	Decrease the time pieces and optimize the total executing cost.	Cloud Environment	Other
Adaptive Hybrid Heuristic scheduling algorithm	Dependency mode	Budget, deadline and execution cost	Identifie dynamic scheduling approaches	Cloud Environment	Other

As Table 1 shows, we make a comprehensive comparison with these algorithms, and make analysis based scheduling model, scheduling parameter, scheduling result, the tools and the environment they used. We can find that all algorithms used above are cloud-based environment. Most of algorithms are batch-based and dependency-based, and the simulation based CloudSim or other simulators. The batch-based is clustering some relatively separate instances to dispatch; and the dependency-based is that the single instance composed of multi-activity instances; and what the dispatch of Virtual clusters is that using integrated tasks and virtual machines to define the local task and virtual machine list which are the input of scheduling algorithm. Most of algorithms focus on the execution cost and time which are the two hot research areas. In addition, all the scheduling results of algorithms have better performance than the non-optimized algorithms.

4 Typical Tools for Cloud Workflow Scheduling Research

4.1 CloudSim[1]

Cloud computing can satisfy the different service requests with different configuration, deployment condition and service resources of various user at different time point. With the influence of multidimensional factors, it is unreality to test with different parameters in actual cloud computing center. So CloudSim, WorkflowSim and so on have been used for simulating with cloud workflow scheduling.

CloudSim is a toolkit (library) for simulation of cloud computing scenarios. It provides basic classes for describing data centers, virtual machines, applications, users, computational resources, and policies for management of diverse parts of the system (e.g., scheduling and provisioning) [18].

[1] `http://www.cloudbus.org/cloudsim`

The main features of CloudSim are following:

1. Support for modeling and simulation of large scale Cloud computing data centers;
2. A self-contained platform for modeling Clouds, service brokers, provisioning, and allocation polices;
3. Support for simulation of network connections among the simulated system elements;
4. Facility for simulation of federated Cloud environment that inter-networks resources form both private and public domains.

Above these, the researchers can evaluate the hypothesis prior to a real deployment in an environment with CloudSim, where one can reproduce tests, while they can also test the new developmental methodologies and policies in cloud computing environment. CloudSim can bring us some benefits: (Ⅰ) to test the performance of their provisioning and service delivery policies in a repeatable and controllable environment free of cost; and (Ⅱ) to tune the performance bottlenecks before real-world deployment on commercial Clouds.

4.2 WorkflowSim[2]

WorkflowSim extends the CloudSim simulation toolkit by introducing the support of workflow preparation and execution with an implementation of a stack of workflow parser, workflow engine and job scheduler. It supports a multi-layered model of failures and delays occurring in the various levels of the workflow management systems. The architecture of WorkflowSim is shown in Figure 1.

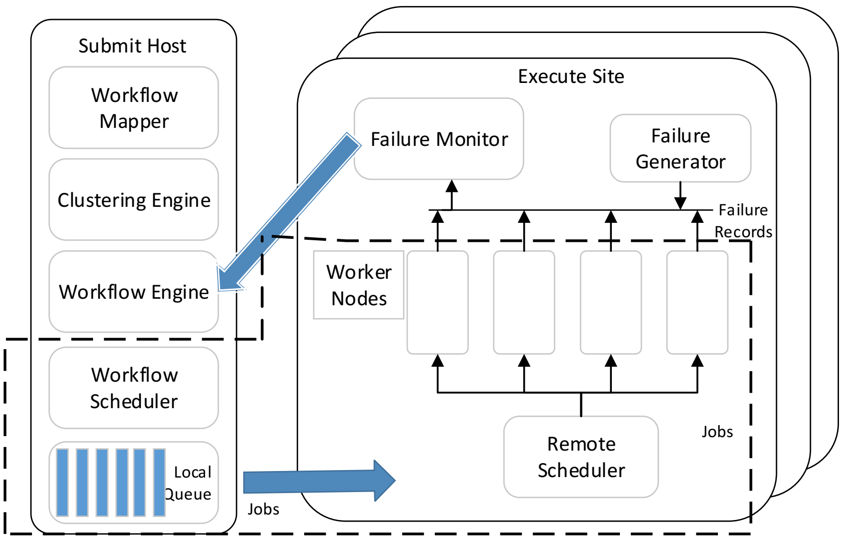

Fig. 1. WorkflowSim Overview. The area surrounded by dotted lines is supported by CloudSim.

[2] `http://www.workflowsim.org`

As it shows, the submit host consists of Workflow Mapper which maps abstract workflows to concrete workflows that are dependent on execution sites, Clustering Engine which merges small tasks into a large job such that the scheduling overhead is reduced, and Workflow Engine which handles the data dependencies and local scheduler and that only releases free tasks to Clustering Engine. The Execution Site consists of Remote Scheduler which is used to match jobs to a worker node based on the criteria selected by users, Worker Nodes, Failure Generator which is introduced to inject task/job failures at each execution site during the simulation and Failure Monitor which collects failure records to return these records to Clustering Engine to adjust the scheduling strategies dynamically [19].

In actual operation process, as the workflow is so numerous, even contains tens of thousands of tasks, while we usually have only dozens of computing nodes, task clustering needed to use for polymerizing similar tasks at this moment, and form clustered job accordingly, which generally called job. Each job include several tasks, and integrally submit to operating environment, in this way can save a lot of submission delay, and open and execute the clustered job separately when a certain computing node is available.

WorkflowSim is used for validating Graph algorithm, distributed computing, workflow scheduling, resource provisioning and so on. In addition, WorkflowSim is an open source workflow simulator that has been hosted on GitHub[3]. Compared to CloudSim and other workflow simulators, WorkflowSim provides support of task clustering that merges tasks into a cluster job and dynamic scheduling algorithm that jobs matched to a worker node whenever a worker node become idle. A series of popular workflow scheduling algorithm (e.g., HEFT, Min-Min, and Max-Min) and task clustering algorithms have been implemented in WorkflowSim. Users can specify different criteria to optimize the overall performance.

4.3 SwinDeW-C and SwinFlow-Cloud

Instance-intensive application is one ubiquitous workflow application in real life, but traditional workflow systems cannot give an enough support to these applications. Therefore, the group led by Yun Yang professor has proposed the concepts of instance-intensive workflow, and focused on research of the design of system architecture, scheduling algorithms. The cloud workflow systems have gained rapid development, the typical of which are SwinDeW-C based on SwinDeW-G from Swinburne University of Technology, Cloud based on Hadoop from University of Waterloo, Cloudbus Engine based on Gridbus from University of Melbourne. The design of cloud workflow system architecture is following:

The workflow system architecture has improved from centralized architecture to decentralized architecture. But centralized architecture is still the most popular paradigm in today's workflow community because it is simple and easy to rapidly implement a prototype or product to support workflows. The client-server model also is a typical

---

[3] `https://github.com/WorkflowSim/WorkflowSim-1.0`

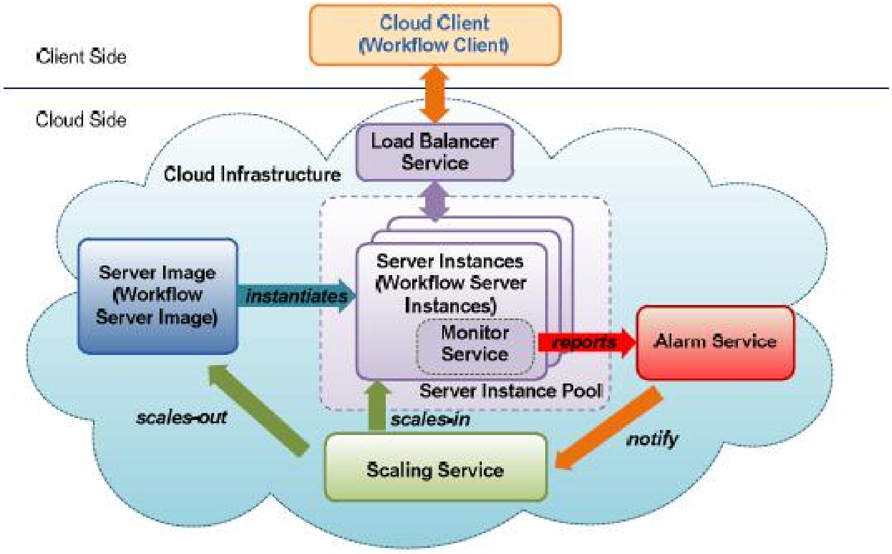

Fig. 2. Cloud workflow system architecture of supporting instance-intensive application

centralized architecture, such as IBM FileNet, TIBCO iProcess Suite and so on. In addition, although Amazon launched the Simple Workfolw (SWF) which can set up the extensible and elastic workflow applications, and can coordinate each step in application, it cannot effectively support the high throughput workflow. While there are a huge amount of user service requests needed to handle at one time in instance-intensive workflow, what the most important is how to maximize the system throughput.

Above these faults in instance-intensive workflow, Swinburne Decentralized Workflow for Cloud which evolved from a series of SwinDeW projects has been set up by the group of Yun Yang. It is a cloud workflow prototype system built on SwinCloud, and contains four basic layers: Application Layer (User application), Platform Layer (cloud services of middleware, deploy user application), Uniform Resources Layer (abstract/encapsulate the virtual resources), and Physical Layer (hardware resources) [20].

The researchers can describe the cloud workflow application as cloud workflow description which include definition, flowage structure and QoS constraint by modeling tool of SwinDew-C. Then they can make a static verification with the flowage structure and QoS constraint. Next, previous description will be submitted the coordinating nodes. In the stage of workflow instantiation, coordinating nodes allocate the description to suitable nodes. At last, that is workflow execution, tasks will be scheduled on VMs of SwinDew-C to execute. SwinDeW-C also support outside business IaaS, such as offered by Amazon, Google, and Microsoft.

Soon after, Dahai Cao et al. [21] proposed a novel client-cloud architecture for scalable workflow system which takes advantages of cloud computing to support instance-intensive workflow and built the SwinFlow-Cloud prototype system. Client-cloud means that the client communicates with an elastic pool of workflow server instances on the cloud side, which is different from the traditional client-server model where the client communicates with one static or a cluster of physical workflow servers. The high throughput, elastic scalability and cost-effectiveness have been taken account into the system, so it can achieve a dynamic, elastic, and sustainable scalability.

5 Problem Analysis and Prospect

5.1 Existing Problems

The researchers have made great progress in Workflow Scheduling Algorithms so far. But there are also some problems in them, as follows:

1. Most workflow scheduling algorithms focus on the cost and the deadline of workflow, but other scheduling parameters have obtained little attention, for example, like resource utilization, resource reliability, fault-tolerant and so on. The resource in cloud is huge, not only we should pay attention to meeting the deadline but also the improvement of singe resource utilization. At this time, we should solve the problem when the execution happens faults, too.
2. Lack of the workflow task model in cloud. Always, it is more practicability when workflow scheduling algorithms have been made verification. But in the current cloud environment, most researchers just use simple mathematical model or directed acyclic graph because of lacking task log. Most task models are not made according to the specific operational tasks in cloud, the experimental results are least persuasion, so the problem of making appropriate model in cloud needs to be resolved urgently.
3. Most workflow scheduling algorithms have been implemented in cloud simulators, besides the scientific workflow. We must apply workflow scheduling algorithms in real-life environment so that can solve more real practical problem.

5.2 Prospect

1. The reliability should been considered in workflow scheduling algorithms, the robustness can have a significant impact on experiment. Taking the reliability and robustness into account can close to real situation.
2. Duplication based on workflow scheduling algorithm. The parts of tasks can be automatically duplicated to the free computing nodes which have some savings.
3. To be combined with the practical application. Today, most workflow scheduling are used in simulation environment, while it still has some deficiencies compared with real environment.

6 Conclusion

The optimization of cloud workflow scheduling can make full use of various cloud services, which will greatly promote the execution of a workflow. Workflow scheduling is the most important problem in the execution of workflow management in cloud. In this paper, we research several typical cloud computing workflow scheduling algorithms and make a detailed analysis and comparison for them. Meanwhile, we also give a detail about cloud computing tools, like CLoudSim, WorkflowSim and other cloud simulation tools. CloudSim can facilitate the research of the cloud computing simulation and make it easy for researchers to set up their experimental platform; while WorkflowSim and other systems can help researchers process the workflow scheduling optimization according to their own requirements and parameters. At last, we point out the existing problems of workflow scheduling algorithms and put forward some opinions and the direction of improvement., In conclusion, the significant findings of workflow scheduling algorithms in the cloud have been obtained but more progress is still need.

Acknowledgment. This paper was supported by Nature Science Fund of China, under grant number 61272063, 61402167, 61202111, 61402168, the Planned Science and Technology Project of Hunan Province under grant number 13FJ4048, 2014GK3004, and Scientific Research Fund of Hunan Provincial Education Department under grant number 13C160.

References

1. Chai, X., Cao, J.: Cloud Computing Oriented Workflow Technology. Journal of Chinese Computer Systems 33(1), 90–95 (2012)
2. Fan, Y.: Fundamentals of Workflow Management Technology. TUP, Beijing (2001)
3. Yao, H., Tian, S.: Cloud Computing. Electronics Industry Publisher (2013)
4. Buyya, R., Broberg, J., Goscinski, A.: Cloud Computing: Principles and Paradigms. John Wiley & Sons (2010)
5. Bala, A., Chana, I.: A survey of various workflow scheduling algorithms in cloud environment. In: Proc. of the 2nd National Conference on Information and Communication Technology, pp. 26–30 (2011)
6. Liu, K., Yang, Y., Chen, J., et al.: A Compromised-Cost Scheduling Algorithm in SwinDeW-C for Instance-intensive Cost-Constrained Workflows on Cloud Computing Platform. International Journal of High Performance Computing Applications 24(4), 445–456 (2010)
7. Wu, Z., Liu, X., Ni, Z., Yuan, D., Yang, Y.: A market-oriented hierarchical scheduling strategy in cloud workflow systems. The Journal of Supercomputing 63(1), 256–293 (2013)
8. Yan, J., Wu, G.: Scheduling Algorithm for Instance-Intensive Cloud workflow. Journal of Computer Applications 30(243), 2864–2866 (2010)
9. Zheng, M., Cao, J., Yao, Y.: Cloud Workflow Scheduling Algorithm Oriented to Dynamic Pric Changes. Computer Integrated Manufacturing Systems 19(8), 1849–1858 (2013)
10. Mukute, S., Hapanyengwi, G., Mapako, B., et al.: Scheduling in Instance-Intensive Cost-Constrained Workflows in a Cloud. International Journal of Scientific & Engineering Research 4, 755–760 (2013)
11. Durillo, J.J., Prodan, R.: Multi-objective workflow scheduling in Amazon EC2. Cluster Computing 17(2), 169–189 (2014)

12. Pandey, S., Wu, L., Guru, S.M., Buyya, R.: A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In: Proc. of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 400–407 (2010)
13. Lu, G., Tan, W., Sun, Y., Zhang, Z., Tang, A.: QoS Constraint Based Workflow Scheduling for Cloud Computing Services. Journal of Software 9(4), 926–930 (2014)
14. Mao, M., Humphrey, M.: Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In: Proc. of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, p. 49. ACM (2011)
15. Mao, M., Humphrey, M.: Scaling and Scheduling to Maximize Application Performance within Budget Constraints in Cloud Workflows. In: Proc. of the 27th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2013), pp. 67–78 (2013)
16. Rahman, M., Hassan, M.R., Ranjan, R., Buyya, R.: Adaptive workflow scheduling for dynamic grid and cloud computing environment. Concurrency and Computation: Practice and Experience 25(13), 1816–1842 (2013)
17. Calheiros, R.N., Buyya, R.: Meeting Deadlines of Scientific Workflows in Public Clouds with Tasks Replication. IEEE Transactions on Parallel and Distributed Systems 25(7), 1787–1796 (2013)
18. Buyya, R., Ranjan, R., Calheiros, R.N.: Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In: Proc. of the IEEE International Conference on High Performance Computing & Simulation (HPCS 2009), pp. 1–11 (2009)
19. Chen, W., Deelman, E.: Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In: Proc. of the 8th IEEE International Conference on E-Science, pp. 1–8 (2012)
20. Liu, X., Yuan, D., Zhang, G., et al.: SwinDeW-C: a peer-to-peer based cloud workflow system. In: Handbook of Cloud Computing, pp. 309–332. Springer US (2010)
21. Cao, D., Liu, X., Yang, Y.: Novel Client-Cloud Architecture for Scalable Instance-Intensive Workflow Systems. In: Lin, X., Manolopoulos, Y., Srivastava, D., Huang, G. (eds.) WISE 2013, Part II. LNCS, vol. 8181, pp. 270–284. Springer, Heidelberg (2013)