Simplified Resource Provisioning for Workflows in IaaS Clouds

Amelie Chi Zhou and Bingsheng He
Nanyang Technological University, Singapore

Abstract—**Resource provisioning is an important and complicated problem for scientific workflows in Infrastructure-as-a-service (IaaS) clouds. Scientists are facing the complexities resulting from the diverse cloud offerings, complex workflow structures and characteristics as well as various user requirements on budget and performance. In this paper, we review the related work on the cost-aware optimizations of workflows in IaaS clouds and summarize the underlying research issues. Existing studies are not effective enough on finding good solutions to workflow optimization problems due to the complexity of workflows and the cloud dynamics. The heuristics proposed in the existing work are specifically designed for certain applications or certain budget and performance requirements. To address those issues, we propose a flexible and effective optimization system to simplify the resource provisioning for scientific workflows in IaaS clouds. The system adopts a probabilistic QoS notion to obtain good optimization results in the dynamic cloud environment and a cloud- and workflow-specific declarative language to specify various workflow optimization problems. We summarize our ongoing work and present some preliminary results on real-world scientific workflows. The experimental results demonstrate the effectiveness of our system on monetary cost optimizations and its capability to solve a wide class of optimization problems for scientific workflows.**

Keywords-**Cloud computing, cloud dynamics, spot instances, resource provisioning, scientific workflows.**

I. Introduction

Workflow models have been widely used by scientists to organize and manage their data analysis jobs in many scientific applications [1]. For example, the astronomical application Montage [2] and the Ligo [3] application for detection of gravitational waves are widely used in many case studies. A scientific workflow is composed of multiple tasks with data dependencies. Each task in the workflow involves managing and processing large data sets and different tasks can have very different I/O and computational behaviours. For example, a task responsible for loading the input data is an I/O-intensive task while another task for processing the input data may have special requirement on the computational resources.

Due to the pay-as-you-go characteristic of the cloud, many real-world scientific workflows are currently deployed and executed in IaaS clouds such as Amazon EC2 [4]. Although the scalability and elasticity of the cloud have brought great opportunities for the workflows, many research problems also arise. Resource provisioning is one important problem for the monetary cost and performance optimizations of scientific workflows in IaaS clouds. Since cloud providers usually offer multiple instance types with different prices and computational capabilities, we need to carefully decide the types of instances that each task of a workflow for performance and monetary cost optimizations. However, making the resource provisioning decisions is non-trivial, involving the complexities from cloud, workflows, and users.

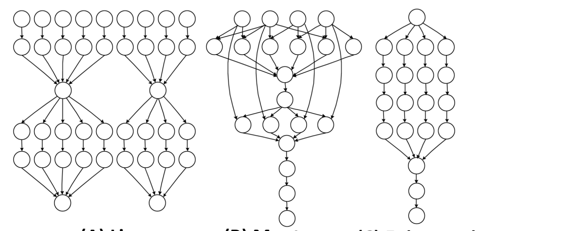


Fig. 1: Workflow structures of Montage, Ligo and Epigenomics.

A. Motivation

The resource provisioning for workflows in IaaS clouds is a complex problem, from the following three aspects.

Diverse cloud offerings. The IaaS clouds usually offer a large number of instance types. For example, Amazon EC2 provides more than 20 types of instances (only counting the latest generation) for the users [5]. Different types of instances have different capabilities and prices. For example, Amazon EC2 offers storage optimized instances to provide very high random I/O performance for I/O-intensive applications. If we consider multiple clouds, the situation is even worse since the cloud providers usually adopt different cloud offerings. For example, Amazon EC2 adopts hourly pricing scheme while Google Compute Engine charges users by minute.

The dynamics in cloud performance and prices make the problem even more complex. The cloud environment is by design a shared infrastructure. The performance of cloud resources, such as I/O and network, is dynamic due to interferences between users. We have observed remarkable dynamics in the I/O and network performances from Amazon EC2 [6]. On another hand, the cloud is an economic market and has dynamic prices [7]. Amazon EC2 offers spot instances, whose prices are determined by market demand and supply. Most existing optimization approaches for scientific workflows in IaaS clouds [8], [9], [10] adopt static notions of performance and cost, which are not suitable for performance and cost optimizations in the dynamic cloud environment. Effective optimization techniques and more rigorous QoS notions are in need to capture the cloud dynamics.

Complex workflow structures and characteristics. The tasks within a scientific workflow can have different characteristics, e.g., I/O-intensive tasks and computational intensive tasks. Different workflows can have different structures and parallelism. For example, as shown in Figure 1, Montage has a very complex workflow structure while Ligo and Epigenomics have high parallelism in their workflow structures. There are also different application scenarios of workflows. For example, the workflows can be continuously submitted to the cloud and the optimizations are made for each workflow individually [9], [10]. Users can also group the workflows with similar structure but different input parameters as an ensemble, and submit QoS and optimization requirements for the entire ensemble [8]. We need an effective system that is capable of simplifying the optimizations of different kinds of tasks and workflows. We should also consider how to make use of the different workflow structures for cost and performance optimizations.

Various user requirements. Scientists submit their workflow applications to the IaaS clouds usually with some predefined optimization objectives and QoS requirements. For example, one may desire to finish a workflow execution with a minimum monetary cost before a predefined deadline while another one may desire to execute a workflow as fast as possible with a given budget. Users may also define skyline optimization objectives, e.g., minimizing both of the monetary cost and the execution time of workflows. The users' requirements are also evolving. For example, a user may want to minimize the execution time of a workflow on a cloud $C1$ with a predefined budget. On the other scenario, she may consider running the workflow on multiple clouds besides $C1$. At this point, the optimal solution depends on the offerings of the multiple clouds and the network performance across clouds. Existing optimization algorithms are specifically designed for certain optimization problems and are usually not extensible or flexible to various evolving user requirements.

B. Goals and Objectives

The major goal of our research is to design a flexible and effective optimization system to simplify the optimizations of monetary cost and performance for scientific workflows in IaaS clouds.

Effectiveness. The system should be effective on obtaining good optimization results and capturing the cloud dynamics in performance and prices. There are a good number of studies [9], [8], [10] working on the monetary cost and performance optimizations for scientific workflows. However, none of them have taken the cloud dynamics into account and thus can hardly always satisfy the performance and budget requirements in the dynamic cloud environment. Some heuristics adopted in the previous work, such as the deadline assignment heuristic [9], have been demonstrated less effective than the more comprehensive approach to explore the solution space [6].

Simplification. We are aiming at designing a system that can simplify the optimizations of various resource provisioning problems for workflows in IaaS clouds. Many existing optimization techniques for scientific workflows are only designed for certain optimization goals and constraints, and thus are not suitable for the evolving cloud offerings and user requirements. In our study, we aim to propose a flexible optimization system to solve a wide class of optimization problems for workflows, without modifying the optimization system. We observe that, scientists often have the tedious issues on handling workflow structures, cloud offerings and resource provisioning. Thus, our simplification allows users to focus on their high-level application logics (e.g., the performance and budget requirements), without worrying about those tedious issues.

C. Our Solution

To achieve the above goals and objectives, we propose the following techniques. We propose to use probabilistic distributions to effectively capture the cloud dynamics in network and I/O performances as well as the cloud prices. We adopt a probabilistic deadline notion for users to specify their QoS requirements in the dynamic cloud and propose a scheduling system to meet the probabilistic deadline requirements. We have formulated a number of transformation operations for monetary cost and performance optimizations of workflows. These operations are common for any workflow structures. We also propose a declarative language specifically designed for workflows in the cloud for users to specify their optimization problems and utilize an efficient search engine to solve the problems. To the best of our knowledge, our solution is the first of its kind in developing a flexible declarative optimization framework with the awareness of cloud dynamics for workflows in IaaS clouds.

The organization of this paper is as follows. We introduce the background and review the related work in Section II. We present the underlying research problems in Section III and briefly introduce our initial work and results in Section IV. Section V concludes this paper and highlights some directions for future research.

II. Background and Related Work

We introduce the background on the cloud application scenario and revisit the related work.

A. Background

We consider two typical scenarios of resource provisioning for scientific workflows in the IaaS clouds. In the first scenario, scientists purchase resources from the IaaS clouds to run their workflows. Our goal in this scenario is to design an optimization framework for the scientists to satisfy their performance and cost requirements. Another typical scenario is providing software-as-a-service for workflows in the IaaS clouds. We denote this model as workflow-as-a-service (WaaS). WaaS providers rent resources from the IaaS cloud providers to run the workflow applications. In this scenario, we perform optimizations for the sake of WaaS providers.

We have seen many applications with different workflow structures and optimization objectives in the cloud [4]. We use our project [11], "cloud-assisted water quality management and analysis", as an example to show the background scenario of our problem. The project is developing a cloud processing platform such that users (e.g., water quality related experts and officials) can submit their simulation tasks to predict the water quality for different reservoirs, or perform sensitivity analysis with water quality simulations. Users can also perform data analysis on the water quality history with data mining or machine learning techniques. The workload fluctuates in the system: in the rain-fall season, users run the simulation more often than other times; report generation tasks are more frequent when approaching the end of every week or every month, even the end of season or year. The elasticity feature of cloud computing is attractive to adapt to the workload fluctuation. The monetary cost optimization is an important problem for the research project, since those workflows are executed for many times during the year.

B. Related Work

Monetary cost optimizations is a popular research topic for scientific workflows in IaaS clouds. Many studies have been presented on this topic and we only review the ones most relevant to our research goal.

Cost-aware optimizations for workflows. The pay-as-you-go nature of cloud computing attracts many research efforts in dynamic resource provisioning with performance and budget requirements. Workflow scheduling with deadline and budget constraints (e.g., [12], [9], [8], [10]) has been widely studied.

Many research studies have been conducted for a single cloud provider. Yu et al. [13] proposed deadline assignment technique for workflow scheduling problems with deadline constraint. Mao et al. [9] applied a series of heuristics, including the deadline assignment method, to automatically scale out for the monetary cost optimizations of workflows with deadline constraints. They also considered the performance optimization problem with budget constraints for scientific workflows.Malawski et al. [8] proposed dynamic scheduling strategies for workflow ensembles. These studies all assume static execution time of individual tasks. Buyya et al. [10] proposed an algorithm with task replications to increase the likelihood of meeting deadlines for workflows.

Considering multiple cloud providers, Fard et al. [14] introduced a heuristic for the cost optimization with SLA requirements. Many studies are introducing cloud brokers to deal with the scheduling problem with multiple cloud providers. For example, Simarro et al. [15] proposed a scheduler under the context of cloud broker to minimize the monetary cost. We refer readers to a recent survey [16] for inter-cloud optimizations.

Cloud dynamics. There are generally two categories of research studying the cloud dynamics. The first category is on utilizing and modeling the price dynamics and the second category is on measuring and analyzing the performance dynamics in the cloud.

Amazon EC2 spot instances, which cause price dynamics in the cloud, have attracted many research interests due to their ability on reducing monetary cost. Yehuda et al. [17] proposed a price model consistent with existing spot price traces using reverse engineering. Javadi et al. [17], [18] developed statistical models for different types of spot instances. Existing studies are utilizing spot instances for different applications. Chohan et al. [19] proposed a method to utilize the spot instances to speed up the MapReduce tasks. Yi et al. [20] introduced some checkpointing mechanisms for reducing cost of spot instances. Ostermann et al. [21] utilized spot instances for large workflow applications when the Grid resources are not sufficient. Further studies [22], [23], [24] used spot instances with different bidding strategies and incorporating with fault tolerance techniques such as checkpointing, task duplication and migration. Those studies are with spot instance only, without offering any guarantee on meeting the workflow deadline. Chu et al. [25] proposed a hybrid method to use both on-demand and spot instances for minimizing total cost while satisfying a predefined deadline constraint. However, they did not consider the cloud performance dynamics.

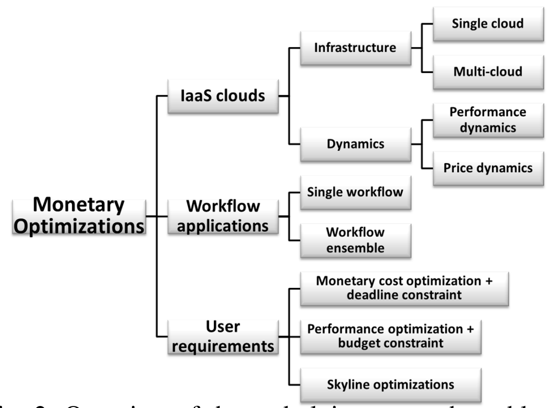


Fig. 2: Overview of the underlying research problems.

A few studies have evaluated the performance of cloud services from different aspects [26], [27], [28]. We have also performed performance calibration on Amazon EC2 and have consistent observations with the previous work [26]. Some studies are proposed to specifically study the network performance [29], [30] and I/O interference [31], [32] of the cloud.

Simplification of optimization problems. There have been some optimization frameworks proposed to simplify various domain-specific optimization problems in the cloud. Alvaro et al. [33] proposed to use a declarative language called Overlog to reduce the complexity of distributed programming in the cloud. Cologne [34] also proposed a declarative language with extensions for constraints and goals to solve a wide class of constrained optimization problems in distributed systems. Zhang et al. [35] proposed to model cloud service configurations in a structured data model, and to formulate the mapping of users resource requirement to cloud resources with SQL queries. ClouDiA is another system that provides instance deployment solutions for users [36]. Rai et al. [37] proposed a novel bin-ball abstraction for the resource allocation problems. Different from bin-ball abstractions, workflows have more complicated structures with data dependencies. Moreover, bin-ball abstractions are mainly for static resources, which cannot capture dynamic cloud performance. To the best of our knowledge, none of the previous generalized optimization frameworks are specifically designed for workflows in IaaS clouds.

III. Research Problems

IaaS clouds, workflow applications and user requirements are three important design factors in the resource provisioning problems of workflows in IaaS clouds. Figure 2 summarizes the ontology of potential research issues in an ontology form.

A. IaaS Clouds

With the popularity of cloud computing, more and more cloud providers offer various infrastructures as a service. We consider the following design aspects in IaaS clouds.

Infrastructure. From the perspective of infrastructures, we consider the optimization problems in both single cloud and multi-cloud environments. Within a single cloud, the optimization problem is to choose appropriate instance types for tasks from the diverse cloud offerings, considering the performance/cost optimization requirements. For example, when the optimization goal is to minimize the monetary cost, cost-effective instance types are selected for tasks. In the multi-cloud environment, the cloud offerings are much more diverse. Also, we need to consider the data/task migrations between clouds for performance/cost optimizations. We need to decide which cloud to migrate the data/tasks to, considering the network bandwidth and price between clouds for performance/cost optimization requirements.

Dynamics. The cloud dynamics include dynamics on both the cloud performance and prices. In either the single cloud or the multi-cloud environment, we should consider the impact of the cloud dynamics on the workflow execution and the optimization effectiveness. We need a more rigorous notion for QoS requirements in the dynamic cloud and new scheduling algorithms to adapt to the notion. The price dynamics in the cloud are mainly caused by the temporally varying price schemes, such as spot instances provided by Amazon EC2. The major issue in utilizing spot instances is that spot instances may fail at any time due to out-of-bid events (the bidding price is lower than spot price). We need to design fault-tolerant methods to ensure meeting the deadline requirements of workflows when utilizing spot instances.

B. *Workflow Applications*

Single workflow vs. Multiple workflows. In many workflow optimization problems (such as [10]), optimization goals and QoS requirements are defined for a single workflow. Appropriate instance types are chosen for each task in the workflow in order to satisfy the optimization goals and constraints. We should also make use of the characteristics of tasks and consider consolidating tasks/instances to further reduce cost. For example, consider two tasks with different characteristics and similar start time. One task is I/O-intensive but does little computation while another one is computation-intensive but does very little I/O operations. In this case, we can actually schedule the two tasks onto the same instance in order to more efficiently utilize the I/O and computation resources.

Workflow ensembles are an representative application to group multiple workflows to execute. Each workflow is associated with a priority to indicate its importance. QoS requirements are either associated with the entire ensemble (e.g., budget constraint) or each single workflow (e.g., deadline constraint). The goal of such a resource provisioning problem is to maximize the overall priorities of completed workflows in the ensemble, within the budget and deadline constraints. In this scenario, we need to decide which workflow to execute and select the appropriate instance types for each task in the workflow. We should avoid executing workflows that are unlikely to finish before deadline and use the limited budget to execute as many high-priority workflows as possible.

C. *User Requirements*

Performance vs. Monetary cost. Users can have very different performance and monetary cost optimization goals and constraints. One widely studied problem is to minimize the monetary cost of executing workflows while satisfying a

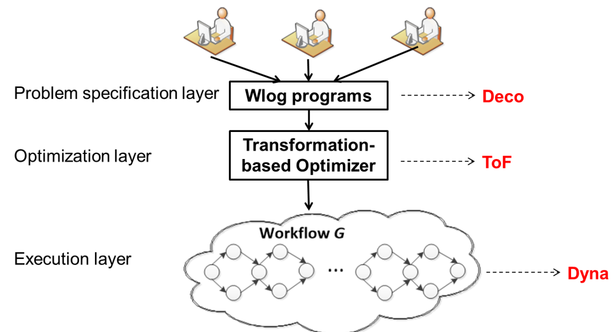


Fig. 3: Overall Design.

predefined deadline constraint. The users may also require optimizing the performance of workflows with budget constraint. Many heuristics have been proposed for the two constrained optimization problems [9], [38]. Another type of optimization problems are skyline optimizations, which target at multiple optimization objectives at the same time, e.g., monetary cost and performance.

Unfortunately, existing studies propose specific heuristics to tackle each one of them. However, we need to design a flexible optimization system applicable to various evolving user requirements. Since a flexible optimization system does not include any customized problem-specific heuristic, an important design issue is to balance the trade-off between simplification and effectiveness.

IV. Preliminary Work and Results

In this section, we briefly introduce our current research projects, and present some preliminary results. Specifically, we introduce our work and results on the optimization effectiveness and simplification aspects.

A. *Overall Design*

Figure 3 shows the overall design of our work. From the bottom up, we have explored three projects (Dyna, ToF and Deco) to achieve the effectiveness and simplification goals. In our Dyna [6] project, we propose probabilistic models to capture the cloud dynamics and a probabilistic QoS notion to effectively minimize the monetary cost of running WaaS for workflows. To simplify the various workflow optimization problems, we have conducted two studies, namely the transformation-based optimization framework (ToF) [39] and the declarative optimization framework (Deco) [40]. ToF embraces several transformation operations that are common for any workflow structures and uses a cost model to guide the selection of transformations during the cost optimization. Deco proposes a workflow- and cloud-specific declarative language called WLog for users to specify their optimization problems. Deco models these optimization problems as search problems by default to be able to solve a wide class of optimization problems.

In Dyna, we consider both the performance and price dynamics in the cloud to effectively minimize the monetary cost for workflows. Specifically, we propose a probabilistic deadline notion for users to specify their QoS requirements in the dynamic cloud environment. A probabilistic deadline requirement of $p_r\%$ means the p_r-th percentile of the workflow execution time distribution is no longer than a predefined

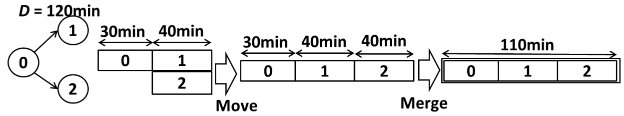


Fig. 4: Example of applying transformation operations on a three node structured workflow.

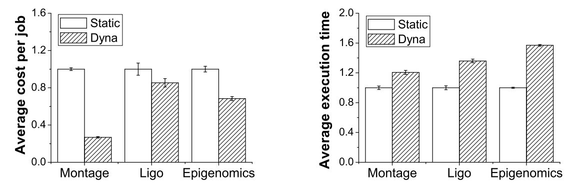
deadline. The system embraces a series of optimization techniques for monetary cost optimizations, which are specifically designed for cloud dynamics. We develop probabilistic models to capture the performance dynamics in I/O and network of instances in IaaS clouds as well as the price dynamics of spot instances. With the captured dynamics, we estimate the execution time of tasks in the dynamic cloud as probabilistic distributions instead of single values. We further propose a dynamics-aware hybrid instance configuration approach to adopt both spot and on-demand instances while satisfying the probabilistic deadline requirements of users. The spot instances are adopted to potentially reduce monetary cost and on-demand instances are used as the last defense to meet deadline constraints.

ToF formulates six common structural transformation operations for the performance and monetary cost optimizations of workflows, including Merge, Demote, Promote, Move, Split and Co-Scheduling. The first two operations are main schemes and the rest are auxiliary schemes. The main schemes can directly reduce cost while the auxiliary schemes are used to help main schemes reduce cost. Consider the example shown in Figure 4, a simple workflow has three tasks and the execution time of Tasks 0, 1 and 2 on the assigned instance types are 30, 40 and 40 minutes respectively. The Move operation only changes the start and end time of tasks and thus is an auxiliary scheme. The Merge operation reduces the charging hours from three to two and thus is a main scheme. We divide the six operations into different categories to reduce the optimization space and optimization overhead. During the optimization process, we adopt a light-weight cost model to guide the selection of transformations periodically. In each plan period, we select two operations, each from main schemes and auxiliary schemes, with the lowest estimated cost. ToF is applicable to any workflow structure and can also be extended by users with their customized transformation operations.

Deco is a declarative optimization framework especially designed for the optimization problems of workflows. Deco is able to serve a wide class of optimization problems for workflows in IaaS clouds. It adopts the probabilistic QoS notion in Dyna to capture the performance dynamics in the cloud and incorporates the transformation operations designed in ToF to efficiently reduce the monetary cost for workflows. Deco designs a declarative language called WLog for users to specify their workflow optimization problems. WLog is extended from Prolog with workflow and cloud specific extensions. Table I gives several examples of such extensions and explains their functionality. Given a WLog program, Deco formulates the problem of finding a good solution as a generic search problem or even more efficient $A^{\star}$ search problem (if users can offer some application specific heuristics to prune optimization space). Users can easily enable the $A^{\star}$ search using the $\mathrm{enabled(astar)}$ keyword shown in Table I. Deco

TABLE I: Workflow and cloud specific built-in functions and keywords in WLog.

Function/Keyword	Remark
$\mathrm{import(daxfile)}$	Import the workflow-related facts generated from a DAX file.
$\mathrm{import(cloud)}$	Import the cloud-related facts from the cloud metadata.
$\mathrm{enabled(astar)}$	The $A^{\star}$ heuristic is enabled for efficiently finding solutions.



(a) Average monetary cost (b) Average execution time

Fig. 5: The average monetary cost and execution time of the compared algorithms on Ligo, Montage and Epigenomics workflows.

performs a series of optimizations to automatically improve the effectiveness of finding a good solution for provisioning workflows. Moreover, Deco leverages the power of the GPU to find the solution in a fast and timely manner.

In the following, we present the preliminary results.

B. Preliminary Results

Optimization Effectiveness. We compare Dyna with the state-of-the-art algorithm [9] (denoted as Static) on three different workflow applications shown in Figure 1. The detailed experimental setup can be found in our technical report [6]. Figure 5 shows the average monetary cost and execution time of the compared algorithms. Dyna saves monetary cost over Static by 15–73% when the probabilistic deadline requirement is 96%. Although the average execution time of Dyna is longer than Static, it can guarantee the probabilistic deadline requirements under all settings.

Optimization Simplification. We compared ToF with the state-of-the-art algorithm [9] on the Montage and Ligo workflows. ToF outperforms the state-of-the-art algorithm by 30% for monetary cost optimization, and by 21% for the execution time optimization. Please refer to our previous work [39] for experimental details.

We use Deco to solve three different workflow optimization problems. Specifically, we formulate a workflow scheduling problem (single workflow and single cloud), a workflow ensemble optimization problem (multiple workflows and single cloud) and a workflow migration problem (multiple workflow and multiple clouds). These use cases have covered a large part of the research problems mentioned in Section III. Our experimental results show that, Deco is able to obtain better optimization results than heuristic based methods in all use cases [40].

Many Workflow Management Systems (WMSes), such as Pegasus [41] and Kepler [42], are widely used to manage the execution of workflows in the cloud. Several tools such as Wrangler [43] and cloudinit.d [44] are developed for automatic resource provisioning in the cloud. We have developed a prototype which integrates Deco into Pegasus to schedule and execute the workflows and a script written with Amazon APIs to acquire and release instances on Amazon EC2.

V. Conclusion and Future Work

Scientific workflows are emerging on IaaS clouds, and resource provisioning has been an important research problem for monetary/performance optimizations of the workflows. In this paper, we review the related work on this problem and the underlying research issues to be addressed. We have conducted several projects aiming at designing a flexible and effective optimization framework to simplify the workflow optimization problems in IaaS clouds. Our preliminary experimental results demonstrate the effectiveness of our system and its capability to solve a wide class of optimization problems for scientific workflows.

We have identified several directions for future research. Firstly, we are looking into the issues of designing an energy-efficient cloud with energy-efficient hardware/software. Secondly, we plan to discover the optimization opportunities in multi-cloud environments.

VI. Acknowledgement

We thank the anonymous reviewers for their constructive comments. This work is partly supported by a MoE AcRF Tier 1 grant (MOE 2014-T1-001-145) in Singapore.

References

[1] A. C. Zhou, B. He, and S. Ibrahim, "A Taxonomy and Survey on eScience as a Service in the Cloud," *arXiv preprint arXiv:1407.7360*, 2014.

[2] Montage Workflow, *http://montage.ipac.caltech.edu/docs/download2.html*, accessed on July 2014.

[3] D. A. Brown, P. R. Brady, A. Dietz, J. Cao, B. Johnson, and J. McNabb, "A Case Study on the Use of Workflow Technologies for Scientific Analysis: Gravitational Wave Data Analysis," in *Workflows for e-Science*, 2007, pp. 39–59.

[4] Amazon Case Studies, "http://aws.amazon.com/solutions/case-studies/."

[5] Amazon EC2 Instance Types, *http://aws.amazon.com/ec2/instance-types/*, accessed on August 2014.

[6] A. C. Zhou, B. He, and C. Liu, "Probabilistic Scheduling of Scientific Workflows in Dynamic Cloud Environments," *CoRR*, vol. abs/1306.6410, 2013.

[7] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou, "Distributed Systems Meet Economics: Pricing in the Cloud," in *HotCloud '10*.

[8] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Cost- and Deadline-Constrained Provisioning for Scientific Workflow Ensembles in IaaS Clouds," in *SC '12*, pp. 22:1–22:11.

[9] M. Mao and M. Humphrey, "Auto-scaling to Minimize Cost and Meet Application Deadlines in Cloud Workflows," in *SC '11*, pp. 49:1–49:12.

[10] R. Calheiros and R. Buyya, "Meeting Deadlines of Scientific Workflows in Public Clouds with Tasks Replication," *IEEE TPDS*, vol. 25, no. 7, pp. 1787–1796, July 2014.

[11] Cloud-Assisted Real-Time and Large-Scale Monitoring and Analysis for Water Quality, *http://pdcc.ntu.edu.sg/camawq/home.html*, accessed on July 2014.

[12] Y. Gong, B. He, and D. Li, "Finding Constant from Change: Revisiting Network Aware Optimizations on IaaS Clouds," in *SC '14*.

[13] J. Yu, R. Buyya, and C. K. Tham, "Cost-Based Scheduling of Scientific Workflow Application on Utility Grids," in *e-Science '05*, pp. 140–147.

[14] U. Bellur and A. Malani, "Cost Optimization in Multi-site Multi-cloud Environments," in *UCC '13*, pp. 115–122.

[15] J. L. Lucas-Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Dynamic Placement of Virtual Machines for Cost Optimization in Multi-cloud Environments," in *HPCS*, 2011, pp. 1–7.

[16] N. Grozev and R. Buyya, "Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey," *Softw., Pract. Exper.*, vol. 44, no. 3, pp. 369–390, 2014.

[17] O. A. Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing Amazon EC2 Spot Instance Pricing," in *CloudCom '11*, pp. 304–311.

[18] B. Javadi, R. K. Thulasiram, and R. Buyya, "Characterizing Spot Price Dynamics in Public Cloud Environments," *Future Gener. Comput. Syst.*, vol. 29, no. 4, pp. 988–999, 2013.

[19] N. Chohan, C. Castillo, M. Spreitzer, M. Steinder, A. Tantawi, and C. Krintz, "See Spot Run: Using Spot Instances for MapReduce Workflows," in *HotCloud '10*.

[20] S. Yi, A. Andrzejak, and D. Kondo, "Monetary Cost-Aware Checkpointing and Migration on Amazon Cloud Spot Instances," *IEEE TSC*, 2011.

[21] S. Ostermann and R. Prodan, "Impact of Variable Priced Cloud Resources on Scientific Workflow Scheduling," in *Euro-Par '12*, pp. 350–362.

[22] M. Mazzucco and M. Dumas, "Achieving Performance and Availability Guarantees with Spot Instances," in *HPCC '11*, pp. 296–303.

[23] W. Voorsluys and R. Buyya, "Reliable Provisioning of Spot Instances for Compute-intensive Applications," in *AINA '12*, pp. 542–549.

[24] Y. Gong, A. C. Zhou, and B. He, "Monetary Cost Optimizations for HPC Applications on Amazon Clouds: Checkpoints and Replicated Execution," in *SC '14 (Poster)*.

[25] H.-Y. Chu and Y. Simmhan, "Cost-efficient and Resilient Job Life-cycle Management on Hybrid Clouds," in *IPDPS '14*.

[26] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 460–471, Sep. 2010.

[27] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," *IEEE TPDS*, vol. 22, no. 6, pp. 931–945, Jun. 2011.

[28] M. S. Rehman and M. F. Sakr, "Initial Findings for Provisioning Variation in Cloud Computing," in *CloudCom '10*, pp. 473–479.

[29] S. Suakanto, S. H. Supangkat, Suhardi, and R. Saragih, "Performance Measurement of Cloud Computing Services," *CoRR*, vol. abs/1205.1622, 2012.

[30] G. Wang and T. S. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," in *INFOCOM '10*, pp. 1163–1171.

[31] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, and C. Pu, "Understanding Performance Interference of I/O Workload in Virtualized Cloud Environments," in *CLOUD '10*, pp. 51–58.

[32] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, C. Pu, Y. Cao, and L. Liu, "Who is Your Neighbor: Net I/O Performance Interference in Virtualized Clouds," *IEEE TSC*, vol. 99, 2012.

[33] P. Alvaro, T. Condie, N. Conway, K. Elmeleegy, J. M. Hellerstein, and R. Sears, "Boom Analytics: Exploring Data-centric, Declarative Programming for the Cloud," in *EuroSys '10*, pp. 223–236.

[34] C. Liu, L. Ren, B. T. Loo, Y. Mao, and P. Basu, "Cologne: A Declarative Distributed Constraint Optimization Platform," *Proc. VLDB Endow.*, vol. 5, no. 8, Apr. 2012.

[35] M. Zhang, R. Ranjan, S. Nepal, M. Menzel, and A. Haller, "A Declarative Recommender System for Cloud Infrastructure Services Selection," in *GECON'12*, pp. 102–113.

[36] T. Zou, R. Le Bras, M. V. Salles, A. Demers, and J. Gehrke, "ClouDiA: a Deployment Advisor for Public Clouds," in *PVLDB'13*, pp. 121–132.

[37] A. Rai, R. Bhagwan, and S. Guha, "Generalized Resource Allocation for the Cloud," in *SoCC '12*, pp. 15:1–15:12.

[38] M. Mao and M. Humphrey, "Scaling and Scheduling to Maximize Application Performance within Budget Constraints in Cloud Workflows," in *IPDPS '13*, pp. 67–78.

[39] A. C. Zhou and B. He, "Transformation-based Monetary Cost Optimizations for Workows in the Cloud," *IEEE TCC*, 2013.

[40] A. C. Zhou, B. He, X. Cheng, and C. T. Lau, "A Declarative Optimization Framework for Scientific Workflows in IaaS Clouds," Nanyang Technological University, Singapore, http://pdcc.ntu.edu.sg/xtra/tr/2014-TR-103-Deco.pdf, Tech. Rep. 2014-TR-103.

[41] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems," *Sci. Program.*, vol. 13, no. 3, pp. 219–237, Jul. 2005.

[42] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific Workflow Management and the Kepler System: Research Articles," *Concurr. Comput. : Pract. Exper.*, vol. 18, no. 10, pp. 1039–1065, Aug. 2006.

[43] G. Juve and E. Deelman, "Automating Application Deployment in Infrastructure Clouds," in *CloudCom '11*, pp. 658–665.

[44] J. Bresnahan, T. Freeman, D. LaBissoniere, and K. Keahey, "Managing Appliance Launches in Infrastructure Clouds," in *TeraGrid '11*, pp. 12:1–12:7.