

Design and Development of an Adaptive Workflow-Enabled Spatial-Temporal Analytics Framework

Xiaorong Li¹, Rodrigo N. Calheiros², Sifei Lu¹, Long Wang¹, Henry Palit¹, Qin Zheng¹, Rajkumar Buyya²

¹Institute of High Performance Computing
A*STAR Institute, Singapore
{lixr, lus, wangl, henry, qinz}@ihpc.a-star.edu.sg

²Cloud Computing and Distributed Systems Lab
Department of Computing and Information Systems
The University of Melbourne, Australia
{rnc, rbuyya}@unimelb.edu.au

Abstract— Cloud computing is a suitable platform for execution of complex computational tasks and scientific simulations that are described in the form of workflows. Such applications are managed by Workflow Management System (WfMS). Because existing WfMSs are not able to autonomically provision resources to real-time applications and schedule them while supporting fault tolerance and data privacy, we present a highly-scalable workflow-enabled analytics system that manages inter-dependable analytics tasks adaptively with varying operational requirements on a common platform and enables visualization of multidimensional datasets of real world phenomena. In this paper, we present the architecture of such a WfMS and evaluate it in terms of performance for execution of workflows in Clouds. A real world application of climate-associated dengue fever prediction was evaluated on public, private, and hybrid Clouds and experienced effective speedup in all the environments.

I. INTRODUCTION

Cloud computing enables a large amount of computer resources to be transformed into a pool of configurable virtual machines to provide an on-demand computing facility to users. It allows pay-as-you-go for billing so that users can access their purchased resources from anywhere at any time. It is a suitable platform for the execution of complex computational tasks and scientific simulations, which may require a spike of computational resources, e.g., computing nodes and storages. In those data intensive applications, there might be a large amount of input, output, and/or intermediate data with dependencies among each other. In order to handle complex data flows, a Workflow Management System (WfMS) [1] is required to ensure elasticity, reliability and efficiency.

WfMSs are responsible for receiving the workflow input from the user, obtaining resources to execute them, scheduling workflow tasks for execution, managing the data transfer for tasks scheduled to different resources in order to meet tasks dependencies, and managing failures in tasks and resources during workflow execution.

To further complicate management of workflows, the rapid growth in the size of datasets and the increasing complexity involved in data analytics require efficient and flexible tools to enable high performance data-intensive computing. If WfMSs are to be used to manage complex applications like in real-time data analytics in a dynamic Cloud infrastructure [2], the challenge of developing such systems increases. The problem of allocating the right number of machines, of the right type,

and for the right time frame is application-dependent and is usually delegated to users. Moreover, the exact execution time of each task is usually unknown a priori, which challenges the system design to handle the heterogeneity and adapt to variation in performance in real time.

Because existing WfMS are not able to autonomically provision resources to real-time applications and schedule them while supporting fault tolerance and data privacy, we present in this paper a highly-scalable workflow-enabled analytics system that manages inter-dependable analytics tasks adaptively with varying operational requirements on a common platform and enables visualization of multidimensional datasets of real world phenomena. The proposed system is able to:

1. Allocate tasks within the workflow into appropriate resources and scaling up/down resources according to users' requirements – *dynamic resource provisioning*;
2. Optimize the performance at run time to meet various goals such as the shortest possible execution time, the most inexpensive execution cost, or the optimized throughput – *adaptive task/workflow/users scheduling*;
3. Self-heal during resource failure or service disruption – *fault tolerance*;
4. Protect data in the public and private Clouds as it is being transferred, stored, and processed – *data privacy*;
5. Monitor the status information in real-time – *application management*.

The main contributions of this paper are: (i) it introduces the architecture and implementation of a WfMS able to achieve the five goals described above and (ii) it contains an evaluation of the proposed WfMS in terms of performance for execution of workflows in Clouds. The application was executed on public, private, and hybrid Clouds and experienced effective speedup in all the environments.

The rest of this paper is organized as follows. Section II describes related works and positions our approach against existing ones. Section III introduces our proposed system, its requirements and architecture, and an application scenario. Section IV presents a performance evaluation of our proposed system. Finally, Section V presents conclusion and future research directions.

II. RELATED WORK

Several scientific workflow platforms were developed in the last years. Popular ones are Pegasus [3], Taverna [4], and Triana [5]. They were developed to support Grid computing environments, where utilization of resources is based on allocation and resources are accessed via collaboration rather than financial incentives. Cloudbus Workflow Engine [1] was originally developed for Grid infrastructures, but was later extended to operate in Cloud environments.

Kim *et al.* [6] proposed a workflow management system able to deploy workflows in hybrid infrastructures composed of TeraGrid nodes and Amazon EC2 resources. Our proposed system, on the other hand, can also leverage resource from arbitrary Grid systems, desktop Grids and Cloud providers.

Gogouvis *et al.* [7] proposed a Workflow Management System for deploying workflow application on virtualized environments that is able to utilize resources from public Clouds. However, it has no dynamic provisioning capabilities to speedup application execution and to meet real-time application performance requirements as does our approach.

Fernandez *et al.* [8] proposed a Cloud Workflow Management System that applies a concept called chemical programming for the application scheduling. The system, however, does not offer dynamic resource provisioning capabilities and autonomic self-healing features.

III. SYSTEM REQUIREMENTS, ARCHITECTURE, AND APPLICATION SCENARIO

Recent advances in the Cloud computing technology made it a potential candidate to host data intensive, scientific application such as workflows. Clouds add extra complexity to Workflow Management Systems (WfMSs) because the amount of resources that WfMS can provision for executing the workflow is virtually infinite, as long as there is budget available for being spent in the workflow execution process. Therefore, WfMSs operating in Cloud environments must be able not only to perform tasks such as scheduling and mapping of tasks to resources and data movement, but also they have to find an acceptable compromise between budget expenditure, resources utilization, and execution speedup.

Furthermore, data privacy is an important issue when public Cloud infrastructures host workflow applications because the data analytics application may eventually be applied to sensitive data such as data from financial and health industries or from governments. Also, real-time workflows impose limits on the processing time of individual tasks, and therefore timely detection of failed tasks or resources and reaction to such events are paramount for meeting performance requirements of real-time applications.

In summary, the requirements we identified and explored in our proposed WfMS system are:

1. Capability of allocating tasks within the workflow into appropriate resources and scaling up/down resources according to users' requirements – *dynamic resource provisioning*;
2. Capability of optimizing the performance at run time to meet various goals such as the shortest possible execution

time, the most inexpensive execution cost, or the optimized throughput – *adaptive task/workflow/users scheduling*;

3. Capability of *self-healing* during resource failure or service disruption – *fault tolerance*;

4. Capability of protecting data in the public and private Clouds as it is being transferred, stored and processed – *security-conscious data migration and data privacy*;

5. Capability of monitoring the status information in real-time – *application management*.

As none of the existing platforms are able to support all the above requirements, we proposed an architecture to meet them. Our proposed architecture is able to (i) share workflows from multiple users for analytics, (ii) harness a workflow management and scheduling engine for adaptive resource allocation and optimization, (iii) remotely execute data analytics services and auto scale up/down based on the problem size and efficiency, (iv) provide big data visualization, and (v) support iterative optimization.

The proposed architecture, depicted in Figure 1, was implemented as a system that integrates data analytics models and tools such as open source data visualization, GeoServer¹, Amazon EC2 and S3², Eucalyptus private Cloud [9] with Cloud clustering service, and Cloudbus Workflow Engine [1]. The top layer of the architecture acts at the application tier, enabling workflows to be executed by the middleware. This top layer performs application-specific tasks such as application monitoring and execution, creation of workflows, and resource control.

Below the application layer, there is the middleware, which interacts with the hybrid Cloud infrastructure, composed of public private Clouds and in-house private Clouds belonging to the organization managing the system, to enable application execution. This layer includes actual scheduling and dispatching of workflow tasks and management of Cloud resources. Communication among all parts happens via events, which are managed by the *Event Service* component of the architecture.

A meta-scheduler that is part of the architecture coordinates the execution of interdependent tasks in a highly scalable manner. Various adaptive workflow scheduling algorithms can be incorporated for multi-objective performance optimization [10][11][12]. The meta-scheduler enables application-dependent information to be analyzed collaboratively and simultaneously on different resources and presented as multi-layered graphics to end users.

The *Workflow Scheduler* component of the middleware extends the Cloudbus Workflow Engine by adding adaptive computing methods to process complex analytic workflows. This was enabled with the inclusion of new functionalities to the workflow engine as cost-effective manual and automatic auto-scaling, security-conscious data migration, iterative optimization, and web-based interfaces for geo-data visualization. Fault tolerant capabilities were introduced in the architecture to enable monitoring of failures on both workflow tasks and resources and re-execution of failed processes

¹ <http://geoserver.org>

² <http://aws.amazon.com>

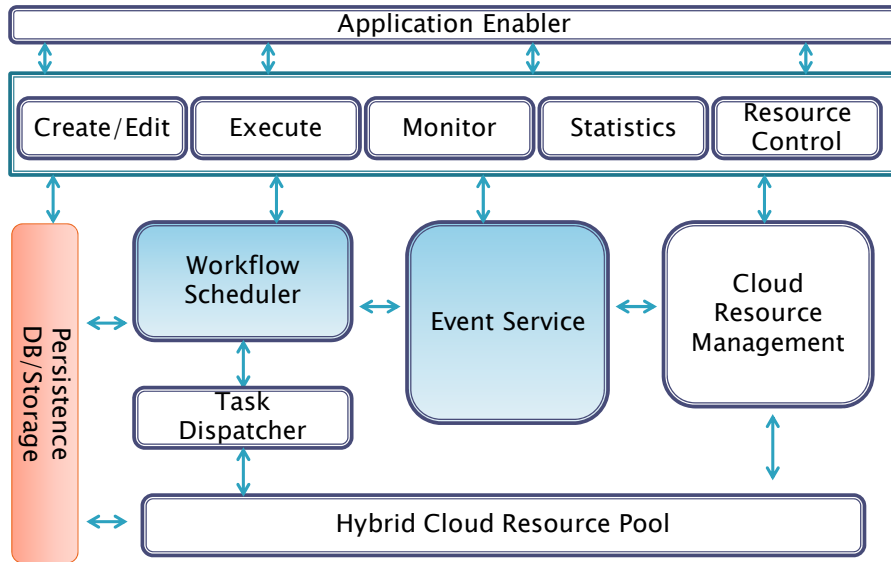


Figure 1. System architecture.

without stopping workflow execution. The *Task Dispatcher* submits workflow tasks to the resources selected by the scheduler for execution.

The *Cloud Resource Management* layer interacts with the physical Cloud infrastructure and with the Application layer to enable features such as adaptive resource allocation and selection of the “best” resource allocation to meet the user time and cost requirements. This component interacts with applications because many of the information it uses to make decision on Cloud resource allocation is application-specific. Therefore, it is important that the *Application Enabler* component contains enough data about the target application and provides means for it to be used by lower layers willing to optimize Cloud resource allocation.

Because optimal results for execution of workflows applications can only be achieved with a strong synergy between the Cloud Resource Manager and the Application Enabler, we discuss next how our proposed architecture enabled the building and deployment of a system for workflow-enabled spatial-temporal analytics for climate-associated dengue fever prediction in Clouds.

A. Application Scenario

Our application scenario is climate-associated dengue fever prediction, a very relevant topic concerning public health in Singapore. Dengue is a mosquito-borne infection disease that occurs in many parts of the world especially tropical regions such as the Southeast Asia. It has become a major international public health concern. According to the World Health Organization (WHO), there are 2.5 billion people in the world living in dengue endemic places. As Singapore is one of

the most densely populated countries in Asia, prediction and control of dengue is an especially important public health issue for the country [13]. Effective methods are needed to analyze the effect of climate variation on dengue disease or the growth of the mosquitoes in a timely manner.

The application takes various factors such as temperature, population density, geographical locations, etc. into consideration, and predicts the distribution patterns of dengue occurrences by analyzing the associations among those factors.

Our Workflow-Enabled Scalable Spatial-Temporal Analytic system processes data about dengue fever cases provided by the Communicable Diseases Division, Ministry of Health (MOH) Singapore. Such data is composed of historical incident data from 1960s to 2011 traced by day, week, and month. This data is combined with Climate variable data obtained from the Singapore Changi Meteorological Station and National Climatic Data Center (NCDC), which is composed of Terabytes of climate-associated data, comprising 300,000 spatial data points multiplied by 365,250 temporal intervals. A simple workflow for this analytics is depicted in Figure 2.

A 1-day dataset requires 30 minutes to be processed, considering the time taken from the initial data extraction to final visualization in a machine with an Intel dual core 2.93GHz CPU and 4GB of memory. In order to reduce such a processing time and enable real-time spatial and temporal analysis with shorter turnaround time, the system must be able to dynamically allocate Cloud resources. Furthermore, such an allocation must be optimized in order to minimize budget expenditure with public Cloud resources.

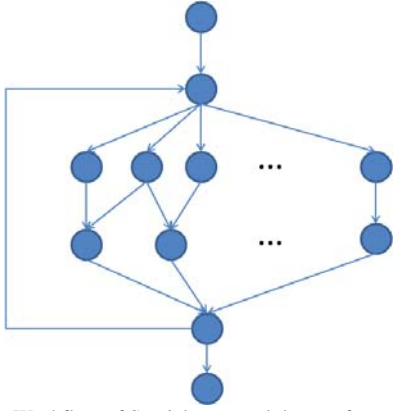


Figure 2. Workflow of Spatial-temporal dengue fever analysis.

In order to achieve the requirements discussed previously in the context of such an application, we developed a system for workflow-enabled scalable spatial-temporal analysis. The performance requirements are achieved by partitioning of the data processing into different parallel tracks and execution of such tracks on multiple virtual machines simultaneously. To achieve that, the system optimizes the performance iteratively and finds the best suitable solutions for performance optimization. The adaptive workflow engine design allows the system to select the best suitable resources according to the user requirements (e.g., update frequency, cost, etc), schedule the privacy-sensitive data in private resources, and tolerate faults when failure happens. Users can define the privacy policy of the data access, and resources are allocated with awareness of the security level.

Security and Privacy: The system requires security-conscious data migration which allows users to specify the security level of each workflow task so that the system will schedule data processing and migration onto highly secured resources to retain data privacy.

Figure 3 shows the security-conscious data migration mechanism in use. In Figure 3(a), users specify the security level of tasks in a workflow, for example high or low. Once specified, the workflow engine will allocate the task to those resources which can meet the security requirements, as depicted in Figure 3(b). For example, the task with high security requirement will only be allocated to resources in private clouds, while those with low security requirements can be allocated to public resources. In this method, the system is able to constraint the data required by a high secured tasks to be migrated to the resources which have relatively low security.

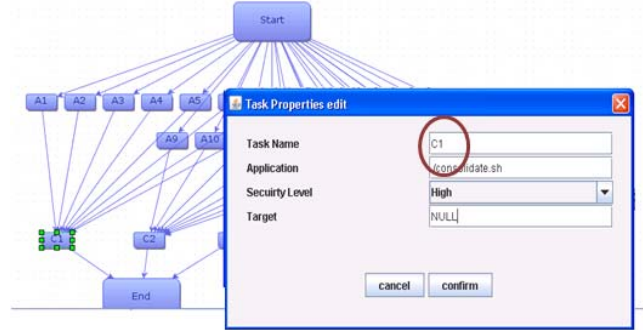
Cost-effective auto-scaling: We consider that resources are allocated to workflow tasks based on a budget constraint, and the system schedules tasks to resources that can optimize the performance in terms of the makespan while satisfying budget requirements. Our previous work [10] [12] [14] on budget constraint scheduling and multi-objective optimization discussed the tradeoff issues of the performance and cost under budget constraints.

The above features of our proposed system enable high scalable, secure, and cost-effective execution of workflows for spatial temporal analysis on Clouds, as demonstrated in the experiments presented in the next section.

IV. PERFORMANCE EVALUATION

We conducted extensive experimental tests to examine the performance of our proposed system in terms of processing time, speedup, and scalable efficiency in private, public, and hybrid Clouds.

The experimental testbed is composed of a private Cloud and a public Cloud that hosted our system individually and together, in the form of a hybrid Cloud.



(a)



(b)

Figure 3. Security-conscious data transfer mechanism in use. (a) Definition of security level of workflow tasks (b) Resource mapping.

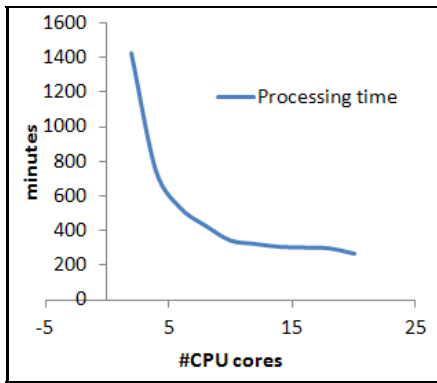
The private Cloud is composed of 4 nodes, each of which has 24 cores (hyper-threaded) 2.93 GHz processor and 96 GB of memory. On top of this infrastructure, we deployed 48 virtual machines. Each virtual machine (VM) is a Linux CentOS 5.8 with 2 or 4 cores and 4 GB of memory.

The public Cloud is composed of 25 Amazon EC2 large instance (2 cores with 4 ECU and 7.5 GB of memory) machines deployed in the region of Asia Pacific (South East).

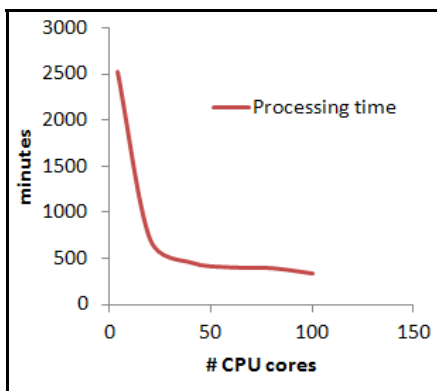
We repeated the workflow execution with different number of cores, from 1 to the maximum number of cores available in the specific Cloud, in order to analyze the system's scalability.

The metrics we used to evaluate our system are speedup and efficiency. We define the speedup, $S=T1/Tn$, as the ratio between processing time on 1 CPU core and the processing time on n CPU cores and the efficiency, $E=S/n$, as the speedup over the number of CPU cores.

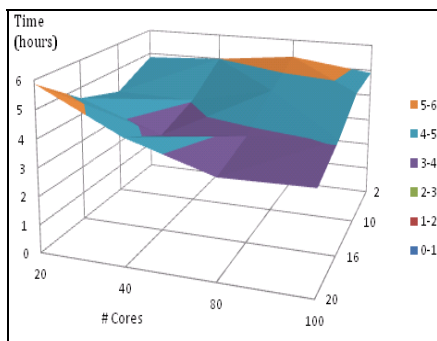
Figures 4-6 show respectively the execution time, speedup, and efficiency of the data analytics application running on each of the three environments. For the public and private Cloud results, the number of cores is presented in x-axis, and the metric is presented in the y-axis. For the hybrid Cloud experiment, separate axis for CPU cores from public and private Clouds are used. Colors in the planes group combinations of public and private Clouds with similar performance for the measured metric.



(a)



(b)



(c)

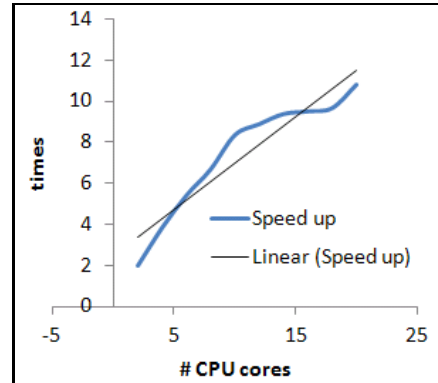
Figure 4. Execution time for different Cloud scenarios. (a) Private Cloud (b) Public Cloud (c) Hybrid Cloud.

Results show that for the three environments, the application presents a region where a high speedup is achieved, followed by a region of smaller speedup. For the

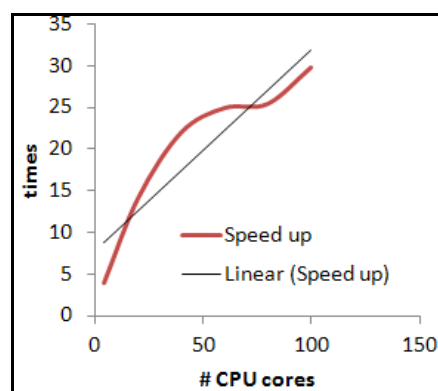
private Cloud, speedup is more significant for up to 5 cores, against around 20 cores from the public Cloud. We observe that a trend of sub-linear speedup is verified for up to 100 cores in the public Cloud, with efficiency above 30% even in the case of this high scale.

V. CONCLUSIONS AND FUTURE WORK

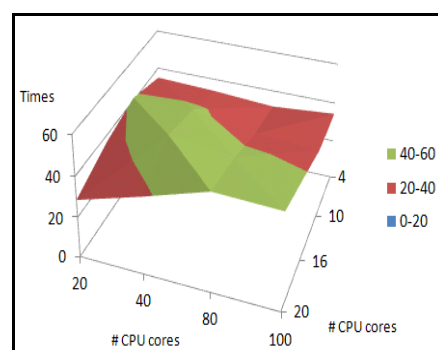
In this paper, we presented a highly-scalable workflow-



(a)



(b)



(c)

Figure 5. Speedup for different Cloud scenarios. (a) Private Cloud (b) Public Cloud (c) Hybrid Cloud.

enabled analytics system that manages inter-dependable analytics tasks adaptively with varying operational

requirements on a common platform and enables visualization of multidimensional datasets of real world phenomena.

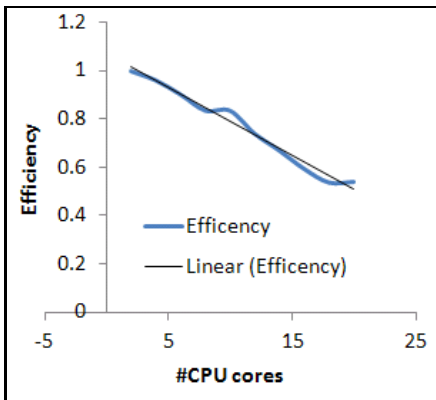
Our proposed system fills the gaps of existing similar systems to allow collaborative workflow analytics in Clouds in which workflows can be contributed by different users, and extends the capability of current spatial-temporal analytics tools to enable complex analytics workflows. It can handle large volume of multidimensional data sets and allows parallelizing the analytics by partitioning the geo-data into multiple trunks and visualizing multi-layered information in a map-like display. A real world application of climate-associated dengue fever prediction was chosen as a use case to demonstrate the capability of such a system in assisting

analysts for real-time spatial-temporal analysis. Executed on public, private, and hybrid Clouds, the application experienced effective speedup in all of the environments.

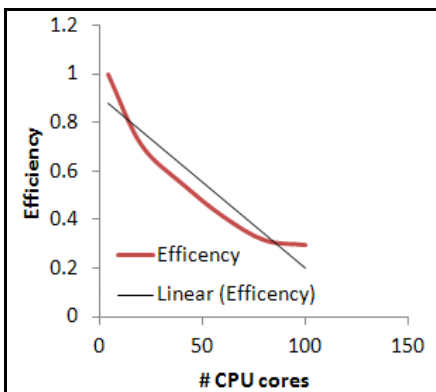
One improvement to the system we plan to investigate is allowing the evolution of the analytics workflows and optimizing the workflows by selecting/replacing certain portions of the workflows on the fly. Other future work includes improving the partitioning of multidimensional data for parallel analytics, in order to enable tasks to split or merge to improve the workflow scalability and efficiency.

REFERENCES

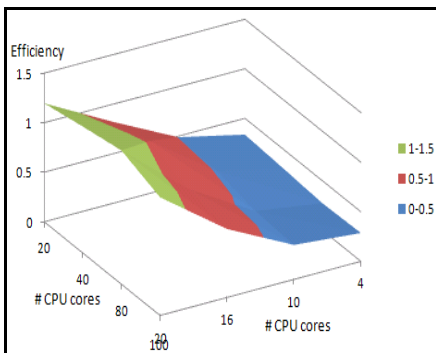
- [1] Suraj Pandey, Dileban Karunamoorthy and Rajkumar Buyya, "Workflow Engine for Clouds", Cloud Computing: Principles and Paradigms, R. Buyya, J. Broberg, A.Goscinski (eds), ISBN-13: 978-0470887998, Wiley Press, New York, USA, February 2011.
- [2] Mustarfizur Rahman, Xiaorong Li, Henry Palit, "Hybrid Heuristic for Scheduling Data Analytics Workflow Applications in Hybrid Cloud Environment", in Proc. High-Performance Grid and Cloud Computing Workshop 2011, in conjunction with International Parallel and Distributed Processing Symposium (IPDPS 2011), 2011.
- [3] Ewa Deelman *et al.*, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems", Scientific Computing, 13(3):219-237, 2005, IOS Press.
- [4] Tom Oinn *et al.*, "Taverna: lessons in creating a workflow environment for the life sciences", Concurrency and Computation: Practice and Experience, 18(10):1067-1100, 2006, Wiley.
- [5] Ian Taylor, Matthew Shields, Ian Wang, Andrew Harrison, "The Triana Workflow Environment: Architecture and Applications", Workflows for E-Science, Ian J. Taylor, Ewa Deelman, Dennis B. Gannon, Matthew Shields (Eds), ISBN-13: 978-1-84628-519-6, Springer, 2007.
- [6] Hyunjo Kim, Yaakoub el-Khamra, Ivan Rodero, Shantenu Jha, Manish Parashar, "Autonomic management of application workflows on hybrid computing infrastructure", Scientific Computing 19(2-3):75-89, 2011, IOS Press.
- [7] Spyridon Gogouvitis *et al.*, "Workflow management for soft real-time interactive applications in virtualized environments", Future Generation Computer Systems, 28(1):193-209, 2012, Elsevier.
- [8] Hector Fernandez, Cedric Tedeschi, Thierry Priol, "A Chemistry-Inspired Workflow Management System for Scientific Applications in Clouds", Proc. 7th International Conference on e-Science (e-Science'11), 2011.
- [9] Daniel Nurmi *et al.*, "The Eucalyptus Open-source Cloud-computing System", Proc. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid'09), 2009.
- [10] Zeng Linfang, Xiaorong Li, Bharadwaj Veeravalli, "ScaleStar: Budget Conscious Scheduling Precedence-Constrained Many-task Workflow Applications in Cloud", in Proc. 26th IEEE International Conference on Advanced Information Networking and Applications (AINA'12), 2012.
- [11] Orachun Udomkasemsub, Sarunya Pumma, Onpreeya Mongkolsermporn, Tiranee Achalaku, "A Multiple-Objective Workflow Scheduling Framework for Cloud Data Analytics", in Proc. 9th International Joint Conference on Computer Science and Software Engineering (JCSSE 2012), 2012.
- [12] Jia Yu, Rajkumar Buyya, and Chen Khong Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids", Proc. 1st International Conference on e-Science (e-Science'05), 2005.
- [13] R. E. Morales Vargas, P. Ya-umphan, N. Phumala-Morales, N. Komalamisra, and J. P. Dujardin, "Climate associated size and shape changes in *Aedes aegypti* (Diptera: Culicidae) populations from Thailand," Infection, Genetics and Evolution, 10(4): 580-585, 2010.
- [14] Rubing Duan, Radu Prodan, Xiaorong Li, "Sequential Cooperative Game Theoretic Approach to Storage-aware Scheduling of Multiple Large-scale Workflow Applications in Grids", in Proc. 13th ACM/IEEE International Conference on Grid Computing (Grid '12), 2012.



(a)



(b)



(c)

Figure 6. Efficiency for different cloud scenarios. (a) Private Cloud (b) Public Cloud (c) Hybrid Cloud.