2011 IEEE International Parallel & Distributed Processing Symposium

Hybrid Heuristic for Scheduling Data Analytics Workflow Applications in Hybrid Cloud Environment

Mustafizur Rahman, Xiaorong Li and Henry Palit

Institute of High Performance Computing (IHPC)
Agency for Science Technology and Research (A*STAR)
Singapore
mmrahman@csse.unimelb.edu.au
lixr@ihpc.a-star.edu.sg
henry@ihpc.a-star.edu.sg

Abstract

Effective scheduling is a key concern for the execution of performance driven applications, such as workflows in dynamic and cost driven environment including Cloud. The majority of existing scheduling techniques are based on meta-heuristics that produce good schedules with advance reservation given the current state of Cloud services or heuristics that are dynamic in nature, and map the workflow tasks to services on-the-fly, but lack the ability of generating schedules considering workflow-level optimization and user QoS constraints. In this paper, we propose an Adaptive Hybrid Heuristic for user constrained data-analytics workflow scheduling in hybrid Cloud environment by integrating the dynamic nature of heuristic based approaches as well as workflow-level optimization capability of meta-heuristic based approaches. The effectiveness of the proposed approach is illustrated by a comprehensive case study with comparison to existing techniques.

1 Introduction

Cloud computing [2] has emerged as a new large-scale distributed computing paradigm that provides a dynamically scalable service delivery and consumption platform facilitated through virtualization of hardware and software with the provision of consuming various services on demand over the Internet. It adopts the market-oriented business model for underlying functionality, where users are charged for consuming Cloud services, such as compute, storage, and network similar to conventional utilities in everyday life (e.g. water, gas, electricity, telephony).

Cloud Workflow Management System (CWMS) is a type of middleware service that facilitates the automation of distributed applications management based on the emerging cloud infrastructure and generally utilized to define, schedule and execute workflow applications on the distributed Cloud services. CWMS uses specific scheduling strategies for mapping the tasks in a workflow to suitable Cloud services (offered by public or private Cloud providers) in order to satisfy user's requirements. However, for large scale data analytics workflow applications, effective scheduling strategies are required to efficiently manage the cost of operation or execution, and at the same time, improve the turn-around time of real-time scientific/business data analytics services, considering the shear dynamism and heterogeneity persisted in Cloud environment.

In this paper, we propose a hybrid heuristic for data analytics workflow scheduling in hybrid Cloud computing environment with the focus on optimizing/minimizing the cost of execution, while satisfying user's requirements or constraints, such as budget, deadline, and data placement. Our proposed approach combines and leverages the advantages of workflow-level optimization capability of meta-heuristic based approaches by generating the initial schedule by using Genetic Algorithm (GA). Then it distributes the user specified workflow-level deadline and budget to task-levels accordingly using the deadline and budget distribution algorithms. Furthermore, it facilitates just-in-time adaptive scheduling by employing the dynamic nature of heuristic based approach for workflow scheduling through Dynamic Critical Path (DCP) algorithm, and also satisfies the user's QoS requirements utilizing the task-level constraints.

1530-2075/11 $26.00 © 2011 IEEE
DOI 10.1109/IPDPS.2011.243

1.1 Motivation

Cloud computing environments are not only dynamic but also heterogeneous with multiple types of services (e.g. infrastructure, platform, and software) offered by various service providers (e.g. Amazon, MS Azure, and Salesforce.com). Scheduling data analytics workflow applications in such environment requires to address a number of issues, including minimizing cost and time of execution, satisfying user's QoS constraints, and considering the temporal behavior of the environment. The majority of scheduling techniques [8][12] proposed to solve these issues are based on meta-heuristics, which produce a good schedule given the current state of Cloud services and reserve the services in advance accordingly.

On the other hand, the existing heuristic based scheduling techniques [5][9] are dynamic in nature and map the workflow tasks to services on-the-fly, but lack the ability of generating schedule considering workflow-level optimization and user QoS constraints, such as deadline and budget. Moreover, most of these techniques do not consider the data placement constraints imposed by the Cloud users, while scheduling the data-intensive workflows.

Thus, in this paper, we endeavor to develop a hybrid heuristic that can effectively integrate most of the benefits of existing approaches to optimize execution cost and time as well as meet the user's requirements through an adaptive fashion in order to efficiently manage the dynamism and heterogeneity of the hybrid Cloud environment.

1.2 Contribution

The key contributions of this paper are as follows:

- proposing an Adaptive Hybrid Heuristic (AHH) for data-analytics workflow scheduling in hybrid Cloud environment by integrating the dynamic nature of heuristic based approaches and workflow-level optimization capability of meta-heuristic based approaches for workflow scheduling.
- designing an algorithm for generating *Comprehensive* workflow from the *Abstract* workflow by taking in to account the service constraints imposed by the users.
- developing a deadline and budget distribution technique to meet the user's QoS contraints in the case of temporal service behaviors.
- analyzing the performance of the proposed hybrid heuristic based scheduling strategy against the existing approaches using a comprehensive case study.

1.3 Organization

The rest of the paper is organized as follows. In the next section, we introduce the basic concepts of data analytics workflow, hybrid Cloud, and workflow scheduling; then provide the definition of workflow scheduling problem in hybrid Cloud environment. Section 3 describes the various components of the system that constitutes the workflow execution environment. The proposed adaptive hybrid heuristic for scheduling budget and deadline constrained data analytics workflows is presented in Section 4 and illustrated with a case study in Section 5. Section 6 presents a brief description of the existing heuristic and meta-heuristic based workflow scheduling techniques utilized in distributed computing systems, such as Grid, Cloud and outlines the uniqueness of proposed hybrid heuristic. Finally, we conclude this paper with future research directions in Section 7.

2 Background and Problem Statement

In this section, we first provide a brief overview of the basic terms related to the scope of this paper including data analytics workflow, hybrid Cloud, and workflow scheduling. Afterwards, we define the problem statement with regards to scheduling data analytics workflow applications in hybrid Cloud computing environment.

2.1 Data Analytics Workflow

Workflow applications comprise a series of large number of structured activities and computations that arise in scientific or business problem solving. Usually, workflows are data or computation intensive and the activities in the workflow have data or control dependencies among them. Data analytics workflow applications are a special class of data intensive workflows, where the activities/tasks are mainly focused on intelligent and intensive processing of data; and the amount of data to be transferred from one task to another is large. As a result, the data processing/transfer time and storage cost are higher for these type of applications. In this paper, we focus on designing algorithms for effective scheduling of data analytics workflows.

2.2 Hybrid Cloud

Clouds are mainly deployed in two ways: public Cloud and private Cloud. Public or external Cloud describes Cloud computing in the traditional main stream sense, where services are dynamically provisioned on demand and self-service basis over the Internet, and charged by the third party providers based on utility. Whereas, private or internal Cloud refers to emulating Cloud computing services on private networks

through virtualization. A hybrid cloud is a combination of a public and a private Cloud aiming at serving the organizational demand of baseline computing through private Cloud and peak computing using public Clouds. For example, an organization may use the private Cloud compute services deployed in local clusters for mission-critical and security concerned applications, while utilize a public Cloud service, such as Amazon Simple Storage Service (Amazon S3) for archiving data as back up.

2.3 Workflow Scheduling

Efficient scheduling technique can induce significant impact on the application runtime as well as system performance. Generally, workflow scheduling is defined as the mapping/allocation of workflow tasks to distributed resources/services and considered as NP-complete problem [4]. Such problems can be solved using exhaustive search techniques; however, the complexity of the method for solving is very high and the optimal solution cannot be generated within polynomial time. Specially, in a dynamic and cost driven environment such as Cloud, scheduling decisions must be made in shortest time possible. Thus, we rely on heuristic and meta-heuristic based scheduling strategies to achieve near optimal solutions within polynomial time. Some of the examples of heuristic based strategy include Min-Min [7], Heterogeneous Earliest Time First (HEFT) [10], DCP [9], and meta-heuristic based strategy include Genetic Algorithm [12], Particle Swarm Optimization [8], Ant Colony Optimization [12].

2.4 Problem Definition

In General, a workflow application is represented as a Directed Acyclic Graph (DAG) in which graph nodes represent tasks and graph edges represent data dependencies among the tasks with weights on the nodes representing computation complexity and weights on the edges representing communication volume. The overall finish/completion time of an application is usually called the schedule length or makespan. The total cost of running the whole application is accrued from various activities: (1) computation cost; (2) storage cost; (3) data transfer (in) cost for each service; and (4) data transfer (out) cost for each service. Moreover, the user may impose various constraints on the execution of application: (1) specifying a particular service for a particular task; (2) archiving intermediate data on a particular storage; and (3) choosing a set of locations for input and output data. Thus, the objective of our workflow scheduling technique is to minimize the total cost of execution, while meeting all the constraints specified by the user including deadline and budget.

In other words, the scheduling problem endeavored to solve in this paper can be stated as: *To find a mapping of all the tasks, T of workflow, W to a set of Cloud services, S such that the estimated total cost incurred for consuming the services is minimized, while Makespan(W) $\leq$ Deadline and Total Cost(W) $\leq$ Budget; and all constraints, $C_{1...k}$ are satisfied.*

Let us assume, data analytics workflow $W(T, E)$ consists of a set of tasks, $T = \{T_1, T_2, \ldots, T_x, .., T_y, T_n\}$ and a set of dependencies among the tasks, $E = \{< T_a, T_b >, ..., < T_x, T_y >\}$, where T_x is the parent task of T_y. $S = \{S_1, S_2, \ldots, S_x, \ldots, S_m\}$ is the set of available compute and storage Cloud services (both public and private) in the hybrid Cloud with varied processing capability delivered at different prices, and only one service can be assigned for a task. Further, $C = \{C_1, C_2, \ldots, C_x, \ldots, C_p\}$ is the set of constraints imposed by the user apart from the stipulated budget, B and deadline, D.

The total execution time, TT of W is composed of two components: (1) computation time; and (2) data transfer time. Thus, TT can be represented as,

$$TT = Makespan(W) = MAX_{1 \leq i \leq n} \mu^{T_i}$$

where, μ^{T_i} is the completion time of task T_i;

Similarly, the total cost, TC of executing W is composed of three components: (1) cost of computation; (2) cost of incoming data transfer; and (3) cost of outgoing data transfer. Thus, TC can be represented as,

$$TC = \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} {C_{comp}}^{T_i}(S_j) + \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} {C_{in}}^{T_i}(S_j) + \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} {C_{out}}^{T_i}(S_j)$$

where, ${C_{comp}}^{T_i}(S_j)$ is the cost generated from computing task T_i on service S_j;

${C_{in}}^{T_i}(S_j)$ is the cost generated from transferring data into service S_j for task T_i; and

${C_{out}}^{T_i}(S_j)$ is the cost generated from transferring data out of service S_j for task T_i .

Therefore, the workflow scheduling problem is to map to , $(T \rightarrow S)$ so that the the following conditions are satisfied:

$TT \leq D$;

$TC \leq B$;

$C_{1....p}$ are incorporated; and

TC is minimized.

3 System Model

Fig. 1 presents the various components of the system that constitutes the workflow execution environment in Cloud. A brief description of these components are as follows:

Service: Cloud service exposes the distributed computing resources including infrastructure, platform, and software to be consumed by the users as a service over the Internet on demand and fine-grained basis through virtualization.

Service Catalogue: Service Catalogue (SC) is crucial to the sharing and utilization of information with regards to Cloud services. The functionality of SC is similar to Grid Information Service (GIS) [3] and it acts as a service directory or registry for publication and discovery of Cloud service providers and consumers.

Service adapter: Service Integration Adapter (SIA) plays a central role in the integration of both private and public Cloud services to the user end and serves as the contact point for accessing/consuming services. An SIA communicates with other components of the system including EE, WE, and SC based on standard communication protocol. SIA registers the services offered by the corresponding provider to EE and updates service status to SC. It also performs other housekeeping operations at the resource site, such as monitoring, accounting, and access control.

Event Engine: We leverage the event-driven mechanism and subscription/notification approach with respect to facilitating coordination among the components in the system. Event Engine (EE) keeps track of and processes the events that are occurred due to the generation of publish or subscribe messages int the system. Thus, all the interaction among SC, SIC, and WE are realized through sending messages to EE and getting corresponding responses from EE.

Workflow Engine: Workflow Engine (WE) aids the users to create, deploy, and monitor the execution of workflows on Cloud services. The service discovery component of WE identifies the suitable services for workflow tasks by querying SC. The scheduler component maps the tasks to these services based on sophisticated scheduling algorithms satisfying users requirements, whereas the dispatcher component deploy the task to the corresponding service using respective SIA. Finally it collects the results and present to user.

4 Proposed Adaptive Hybrid Heuritic Algorithm for Scheduling

Adaptive Hybrid Heuristic (AHH) scheduling algorithm is designed to first generate a task-to-service mapping with minimum execution cost using GA within user's budget and deadline as well as satisfying

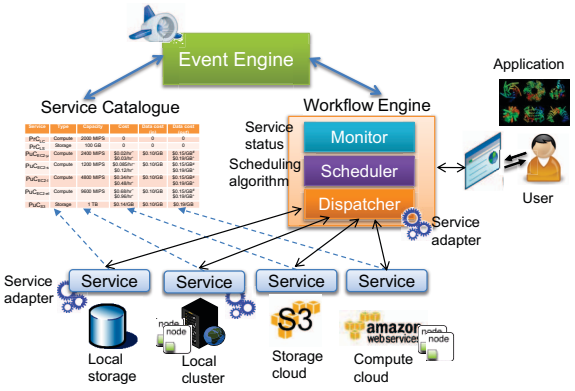


Figure 1: System components facilitating workflow execution in hybrid Cloud.

the service and data placement constraints specified by the user. This initial schedule is then utilized to distribute the workflow-level budget and deadline to task levels. Finally, the DCP heuristic is employed to dynamically schedule the ready tasks level-by-level based on the initial schedule, t_{budget} and $t_{deadline}$, as well as changed status of services. Fig. 2 presents the flowchart that illustrates various segments and flow of logics in the algorithm. Definitions of the terms that are used to formulate the algorithms are given in the following.

Abstract Workflow, W_{abs}: *Abstract* workflow describes a set of computational components in a scientific or business application that are linked by data and control dependencies.

Comprehensive Workflow, W_{com}: *Comprehensive* workflow is an elaborated form of *Abstract* workflow that represents computational components as compute task and data dependencies among these components as input or output task based on user constraints.

Concrete Workflow, W_{con}: *Concrete* workflow is a mapping of all the tasks (compute, input, and output) in a workflow to available cloud services satisfying user's budget and deadline.

Entry task: A task that does not have any parent task is called *entry* task in a workflow.

Exit task: A task that does not have any child task is called *exit* task in a workflow.

Ready task: At any time of scheduling, the task that has all of its parent tasks have completed execution is called a *ready* task in a workflow.

Input task, IT: An *input* task in a workflow is a set/block of data that are required for the execution of a task and transferred from source to destination.

Output task, OT: An *output* task in a workflow is a set/block of data that are generated from the execution of a task and transferred from source to des-

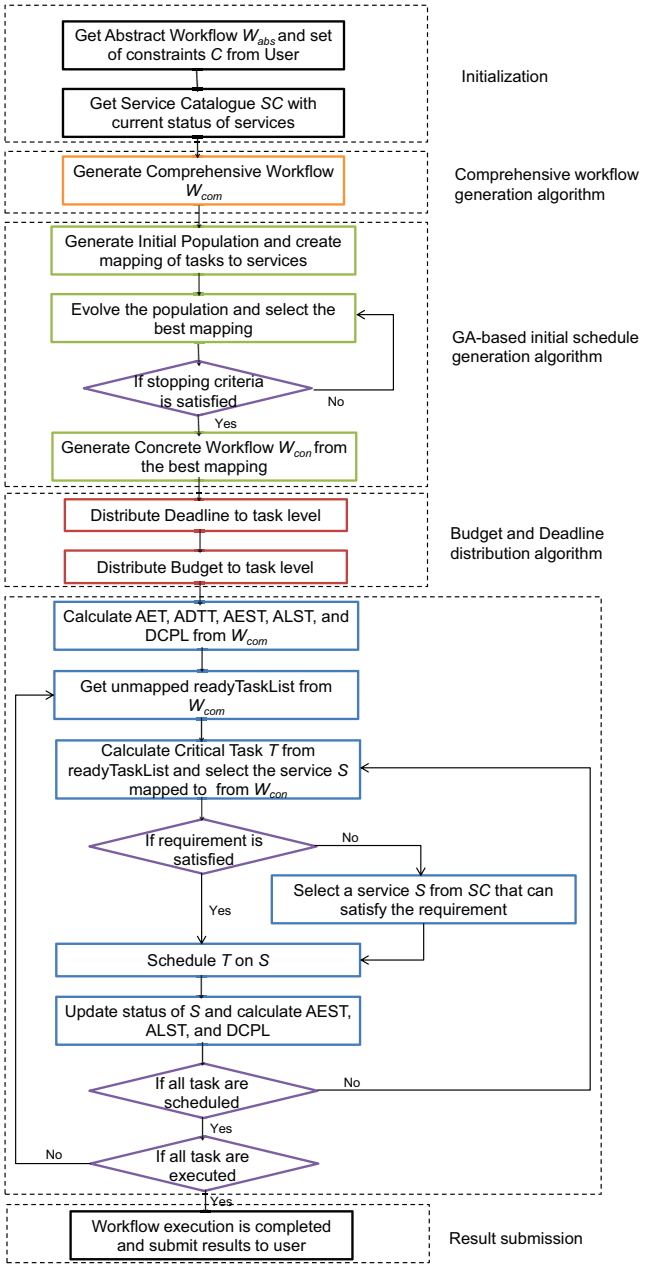


Figure 2: Flowchart for AHH scheduling algorithm.

tination.

Compute task, CT: A *compute* task in a workflow is a set of instructions that are executed as a program on a compute service.

Budget: *Budget* of a workflow is considered as the QoS requirement/constraint imposed by the user on the limit of maximum expenditure for the execution of whole application.

Table 1: Symbols and their meanings

Symbol	Meaning
$AET(t)$	Absolute execution time of task t.
$ADTT(t)$	Absolute data transfer time for task t.
$AEST(t, S)$	Absolute earliest start time of task t on service S.
$ALST(t, S)$	Absolute latest start time of task t on service S.
T_i	i^{th} task.
S_i	i^{th} service.
$DCPL$	Length of dynamic critical path in a workflow.

Deadline: *Deadline* of a workflow is considered as the QoS requirement/constraint imposed by the user on the limit of maximum execution time of whole application.

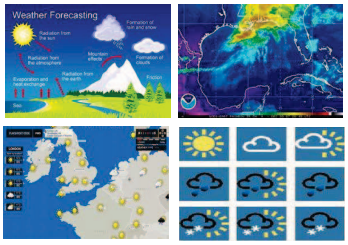
t_{budget}**:** t_{budget} is considered as the constraint imposed by the scheduler on the limit of expenditure for the execution of task t in a workflow.

$t_{deadline}$**:** $t_{deadline}$ is considered as the constraint imposed by the scheduler on the limit of execution completion time of task t in a workflow.

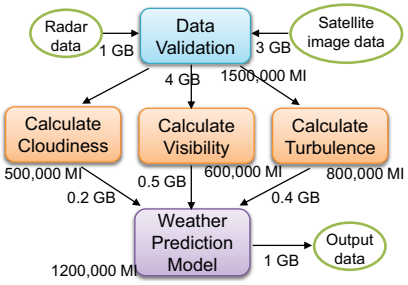
5 Case Study

This section provides a case study of scheduling and execution of a sample data analytics workflow application in hybrid Cloud computing environment under various user's constraints. In order to incorporate the heterogeneity of Cloud environment, we consider different Windows or Linux compute instances (e.g. small, medium, extra large for EC2) with different capability measured in terms of Millions of Instructions Per Second (MIPS), where the size of the tasks in the workflow is measured in Millions of Instructions (MI).

We generate the mapping of workflow tasks to Cloud services and calculate the total execution time and cost using the following scheduling techniques: (1) DCP heuristic focused on cost optimization under user constraints; (2) DCP heuristic focused on time optimization under user constraints; (3) DCP heuristic focused on time optimization without user constraints, assuming that all the tasks are executed on the Amazon Elastic Compute Cloud (EC2) [11], all the data are stored in the Amazon Simple Storage Service (S3), and data transmission are carried out through the Amazon Cloud Front; (4) DCP heuristic focused on time optimization without user constraints, assuming that all the tasks are executed on and data are stored in private Cloud (i.e. local cluster); (5) GA meta-heuristic under user constraints with budget and deadline; and (6) hybrid heuristic under user constraints with budget, deadline, and temporal service behavior.



(a)

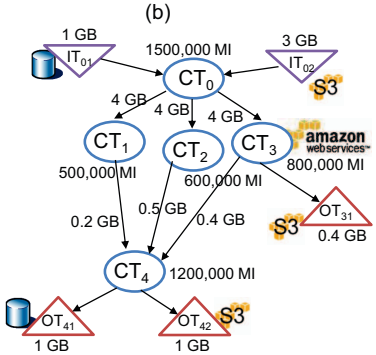


(b)

Task	Constraint Type	Service
T_0	Input	PrC_{LC}
T_0	Input	PuC_{S3}#
T_3	Output	PuC_{S3}+
T_4	Output	PuC_{S3}+
T_4	Output	PrC_{LC}
T_3	Compute	PuC_{EC2}*

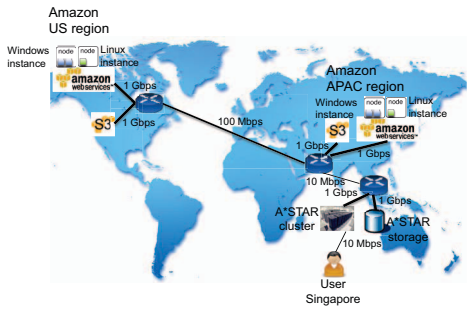
* Windows environment # US-Standard
\+ APAC-Singapore

(c)



(d)

Figure 3: Example of workflow scheduling using AHH algorithm: (a) Weather prediction workflow application; (b) Abstract workflow; (c) User constraints ; (d) Comprehensive workflow.



(a) network configuration

Service	Type	Capacity	Cost	Data cost (in)	Data cost (out)
PrC_{LC}	Compute	2000 MIPS	0	0	0
PrC_{LS}	Storage	100 GB	0	0	0
$PuC_{EC2\text{-}µ}$	Compute	2400 MIPS	\$0.02/hr~ \$0.03/hr*	\$0.10/GB	\$0.15/GB# \$0.19/GB^
$PuC_{EC2\text{-}s}$	Compute	1200 MIPS	\$0.085/hr~ \$0.12/hr*	\$0.10/GB	\$0.15/GB# \$0.19/GB^
$PuC_{EC2\text{-}l}$	Compute	4800 MIPS	\$0.34/hr~ \$0.48/hr*	\$0.10/GB	\$0.15/GB# \$0.19/GB^
$PuC_{EC2\text{-}el}$	Compute	9600 MIPS	\$0.68/hr~ \$0.96/hr*	\$0.10/GB	\$0.15/GB# \$0.19/GB^
PuC_{S3}	Storage	1 TB	\$0.14/GB	\$0.10/GB	\$0.19/GB

* Windows usage ~ Linux usage # US-Standard ^ APAC-Singapore

(b) service configuration

Figure 4: Testbed setup.

5.1 Testbed Setup

Fig. 3(a) presents a sample data analytics workflow application: Weather Forecasting. The abstract workflow, W_{abs} for this weather forecasting application as outlined by the user is shown in Fig. 3(b), whereas compute and data placement constraints given by the users are listed in Fig. 3(c). For instance, one constraint can be executing task T_3 (turbulence calculation) on a windows instance of public Cloud and other constraints could be storing the final output (weather forecasting information) in both public and private Cloud. Once W_{abs} is submitted by the user with the specified constraints, Workflow Engine transforms it into comprehensive workflow, W_{com} (refer to Fig. 3(d)) using the Comprehensive Workflow Generation algorithm.

Fig. 4(a) shows the locality of different sites considered in this case study and the connectivity among these sites with regards to network configuration, where we assume that location of the user and the corresponding private Cloud is in Singapore. On the other hand, Fig. 4(b) lists the Cloud services that are considered to execute the workflow application in this case study. The configuration (i.e. price, processing speed) of these services is compiled from the real world offerings by various Cloud service providers [1]. The initiation time of an instance in EC2 is assumed to be

90 seconds as evidenced in real experiment.

5.2 Schedule Generation

The schedules (tasks-to-services mapping) generated by the DCP and GA under various conditions are illustrated in Fig. 5. From Fig. 5(a) and Fig. 5(b), it is evident that due to the constraint of importing input data (IT_{02}) from and exporting intermediate data (OT_{31}) to S3 as well as execution of CT_3 on EC2, selection of EC2 instances for executing computing tasks can result in faster completion time. Whereas, if we want to minimize the cost by choosing private Cloud instances for computing tasks, execution time increases enormously. However, if the user does not impose any constraint, choosing public Cloud services for all the tasks (see Fig. 5(c)) gives very short execution time at a minimal cost. On the other hand mapping all the tasks to private Cloud services (refer to Fig. 5(d)) can facilitate reducing the cost to \$0.0, but the execution time is not decreased heavily.

Next, the schedule generated by GA for specified budget and deadline under the given user constraints is presented in Fig. 5(e). The total execution time and cost achieved by this schedule is then distributed to task levels as shown in Fig. 5(f) and Fig. 5(g) respectively. Further, the step by step calculation of t_{budget} and $t_{deadline}$ for all the tasks is listed in Fig. 6.

5.3 Discussion

Fig. 7 illustrates the schedule generated by AHH as well as the corresponding execution time and cost for the sample weather forecasting workflow. AHH initially maps the tasks to available services based on the schedule generated by GA (refer to Fig. 5(e)), and assigns t_{budget}, $t_{deadline}$ to all these tasks. However, during the execution of workflow, if status of the mapped services is changed and user's requirements are not being able to satisfy, AHH reassigns suitable services to the corresponding tasks so that the execution can be completed within specified budget and deadline. Thus, as we can see from Fig. 7, when IT_{01} is delayed by 200 seconds, an extra-large instead of large EC2 instance is chosen for CT_3 provided that the time and cost increments are within surplus deadline and budget.

Further, DCP can either minimize the cost or time of execution, but cannot generate a schedule satisfying workflow-level budget and deadline through the process of dynamic task-to-service mapping. Whereas, the static schedule generated by GA could not be able to adapt the schedule to meet deadline requirement dynamically.

6 Related Work

The section focuses to compare the novelty of the proposed work with respect to existing approaches.

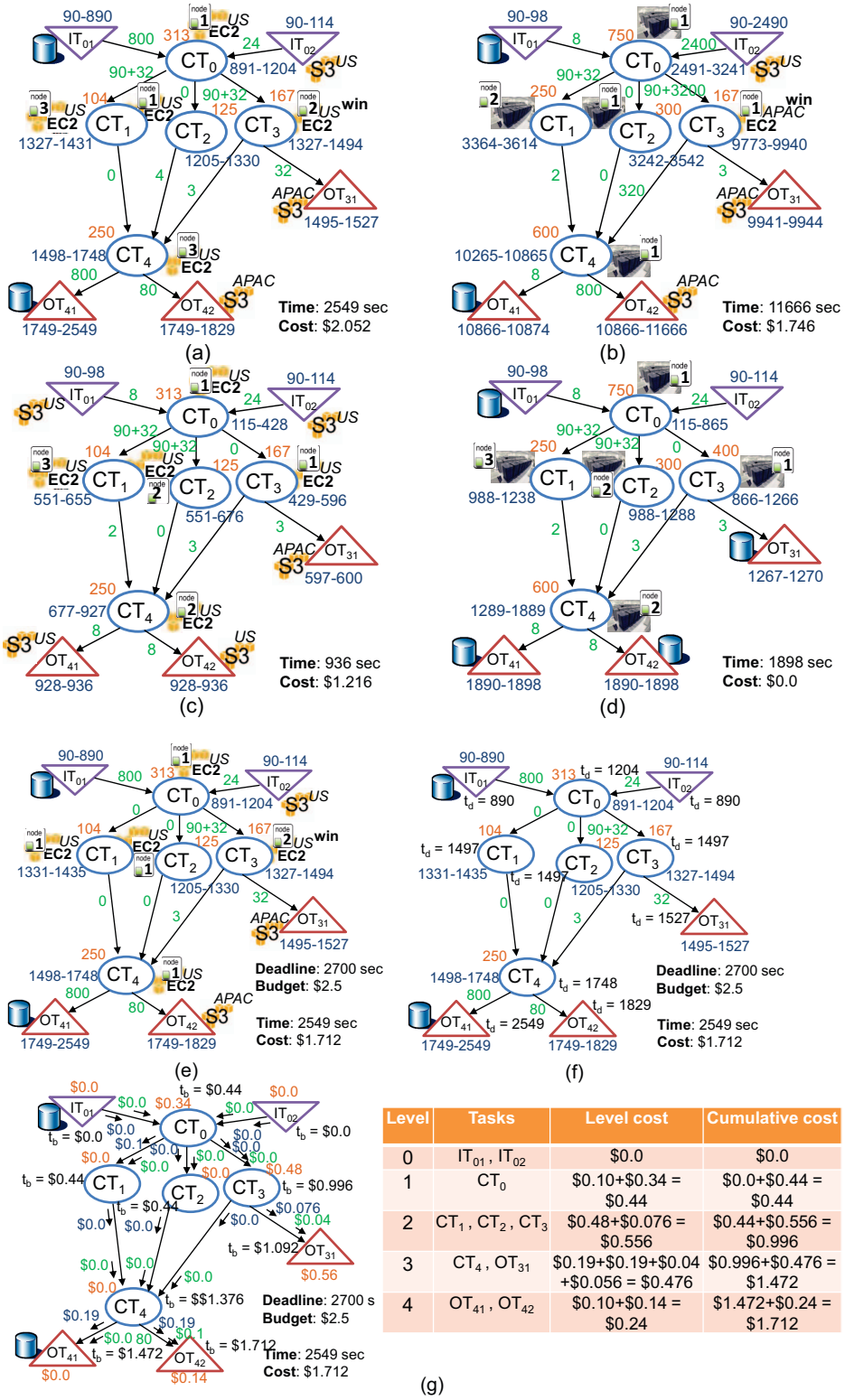
Yu et al. [13] proposed a Genetic Algorithm (GA) based workflow execution planning approach to solve multi-objective (cost and time optimization) scheduling problem for Grid computing environment. It attempts to achieve the trade-off between execution time and cost, while meeting budget and deadline constraint. Rahman et al. [9] investigated a Dynamic Critical Path (DCP) based adaptive workflow scheduling approach for Grids that determines the mapping of tasks to resources dynamically by calculating the critical path in the workflow task graph at every step. The heuristic applied by them tries to optimize execution time of a workflow. K. Liu et al. [6] presented Min-Min-Average (MMA) algorithm for scheduling transaction-intensive Grid workflows. It adapts to changing network transmission speed automatically with an aim to optimize the execution time of workflows.

Wu et al. [12] analyzed meta-heuristic based workflow scheduling strategies (GA, ACO, and PSO) in market-oriented Cloud workflow systems aiming at satisfying the QoS requirements imposed by the users, while decreasing the overall running cost of the workflow system. Pandey et al. [8] contributed on developing a PSO-based meta-heuristic to schedule workflow applications to Cloud resources by taking into account both computational and data transmission cost. X. Liu et al. [5] proposed a MMA algorithm for scheduling instance-intensive workflows on a service-based Cloud environment. This approach mainly focuses on the communication time between the services with an attempt to increase the utilization rate of both execution and transmission unit of workflow execution in order to reduce the mean execution time of all workflows.

On the other hand, proposed workflow scheduling approach leverages the advantages of both heuristic and meta-heuristic based scheduling techniques by combining them into a hybrid heuristic. It supports scheduling in a hybrid Cloud environment (comprising of public and private Cloud providers) as well as takes into account the service constraints imposed by the users and generates the comprehensive workflow accordingly.

7 Conclusion

This paper presents an adaptive hybrid heuristic (AHH) for data-analytics workflow scheduling in hybrid Cloud environment.The performance of the proposed heuristic based scheduling strategy is evaluated against the existing approaches using a comprehensive case study. The results demonstrate that AHH based



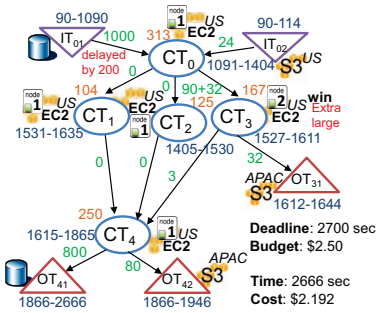
Level	Tasks	Level cost	Cumulative cost
0	IT_{01} , IT_{02}	$0.0	$0.0
1	CT_0	$0.10+$0.34 = $0.44	$0.0+$0.44 = $0.44
2	CT_1 , CT_2 , CT_3	$0.48+$0.076 = $0.556	$0.44+$0.556 = $0.996
3	CT_4 , OT_{31}	$0.19+$0.19+$0.04 +$0.056 = $0.476	$0.996+$0.476 = $1.472
4	OT_{41} , OT_{42}	$0.10+$0.14 = $0.24	$1.472+$0.24 = $1.712

Figure 5: Final schedule with total cost and makespan generated by different scheduling approaches: (a) DCP-time optimization with constraints; (b) DCP-cost optimization with constraints; (c) DCP-time optimization (public cloud only) without constraints; (d) DCP-time optimization (private cloud only) without constraints; (e) GA with constraints (initial schedule/Concret workflow); (f) Deadline distribution; (g) Budget distribution.

scheduling approach is not only capable of adapting to the changes in the Cloud environment but also able to meet users budget and deadline. In future, we endeavor to evaluate the performance of AHH through extensive simulation and real world prototype implementation. We plan to devise specific policies for different workload and particular user requirements in order to better utilize the features of hybrid Cloud computing environment through this evaluation. Moreover, we intend to incorporate other QoS parameters, such as reliability of a service for task-to-service mapping by integrating a multi-objective optimization technique into our proposed algorithm.

Task	Cloud Service	Start time (sec)	End time (sec)	Start time of child tasks (sec)	$t_{deadline}$ (sec)	Cost (compute/st orage)	Data cost (in)	Data cost (out)	Previous level cost	t_{budget}
IT_1	PrC_{LS}	90	890	891	890	$0.0 (s)	--	$0.0	$0.0	$0.0
IT_2	PuC_{S3}^{US}	90	114	891	890	$0.0 (s)	--	$0.0	$0.0	$0.0
CT_0	PuC_{EC2-s}	891	1204	1331,1205, 1327	1204	$0.34 (c)	$0.1	$0.0	$0.0	$0.44
CT_1	PuC_{EC2-s}	1331	1435	1498	1497	$0.0 (c)	$0.0	$0.0	$0.44	$0.44
CT_2	PuC_{EC2-s}	1205	1330	1498	1497	$0.0 (c)	$0.0	$0.0	$0.44	$0.44
CT_3	PuC_{EC2-s}^{Win}	1327	1494	1495, 1498	1494	$0.48 (c)	$0.0	$0.076	$0.44	$0.996
OT_{31}	PuC_{S3}^{APAC}	1495	1527	--	1527	$0.056 (s)	$0.04	--	$0.996	$1.092
CT_4	PuC_{EC2-s}	1498	1748	1749, 1749	1748	$0.0 (c)	$0.0	$0.38	$0.996	$1.376
OT_{41}	PrC_{LS}	1749	2549	--	2549	$0.0 (s)	$0.0	--	$1.472	$1.472
OT_{42}	PuC_{S3}^{APAC}	1749	1829	--	1829	$0.14 (s)	$0.1	--	$1.472	$1.712

Figure 6: Calculation of $t_{deadline}$ and t_{budget}.



Term	Difference
Surplus budget (B_s)	$2.50 - $1.712 = $0.788
Surplus deadline (D_s)	2700 - 2549 = 151 sec
Cost increase	$2.192 - $1.712 = $0.48 < B_s
Time increase	2666 - 2549 = 117 sec < D_s

Figure 7: Final schedule generated by AHH algorithm.

References

[1] Amazon elastic compute cloud (ec2). `http://aws.amazon.com/`.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616*, 2009.

[3] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC'01), USA*, June, 2001.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.

[5] K. Liu. *Scheduling Algorithms for Instance-Intensive Cloud Workflows*. PhD Thesis, Swinburne University of Technology, Australia, 2009.

[6] K. Liu, J. Chen, H. Jin, and Y. Yang. A min-min average algorithm for scheduling transaction-intensive grid workflows. In *Proceedings of 7th Australasian Symposium on Grid Computing and e-Research (AusGrid'09), New Zealand*, January, 2009.

[7] A. Mandal and et al. Scheduling strategies for mapping application workflows onto the grid. In *Proceedings of the 14th IEEE International Symposium on High Performance Distributed Computing (HPDC'05), USA*, July, 2005.

[8] S. Pandey, L. Wu, S. Guru, and R. Buyya. A particle swarm optimization (pso)-based heuristic for scheduling workflow applications in cloud computing environments. In *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA'10), Australia*, April, 2010.

[9] M. Rahman, S. Venugopal, and R. Buyya. A dynamic critical path algorithm for scheduling scientific workflow applications on global grids. In *Proceedings of the 3rd IEEE International Conference on e-Science and Grid Computing (eScience'07), India*, December, 2007.

[10] H. Topcuoglu, S. Hariri, and M. Y. Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 3, pp. 260-274*, 2002.

[11] J. Varia. *Architecting Applications for the Amazon Cloud, Cloud Computing: Principles and Paradigms, R. Buyya et al. (eds.)*. Wiley Press, New York, USA, 2010.

[12] Z. Wu, X. Liu, Z. Ni, D. Yuan, and Y. Yang. A market-oriented hierarchical scheduling strategy in cloud workflow systems. *Journal of Supercomputing*, 2011.

[13] J. Yu and R. Buyya. A budget constrained scheduling of workflow applications on utility grids using genetic algorithms. In *Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing (HPDC'06), France*, June, 2006.