A Systematic Way to Analyze the Performance of DBDCTO Algorithm With Respect To DTC, Greedy Cost, Max-Min and Min-Min Workflow Scheduling Algorithms Using Workflow Simulator

Gaganpreet Kaur, and Yogesh Kumar

***Abstract*—**Cloud computing is internet based computing where virtual shared servers provide software, infrastructure, platform, devices and other resources. Clouds are a large pool of easily usable and accessible virtualized resources. These resources can be dynamically re-configured to adjust to a variable load (scale), allowing also for optimum resource utilization. Cloud Computing is the result of evolution and adoption of existing technologies and paradigms. This papers presents a systematic way how to analyse the performance of DBDCTO algorithm with respect to some existing algorithms: DTC, Greedy Cost, Max-Min and Min-Min. It also defines how to minimize execution time and cost using DBDCTO algorithm for workflow scheduling in workflow simulator environment. Workflow simulator compare the different algorithms and give the better results to minimize the execution time and cost rather than cloud simulator.

***Keywords*—** Cloud Computing, Workflow simulator, Scheduling, Deadline, DTC.

I. INTRODUCTION

CLOUD computing is a construct that allows you to access applications that actually reside at a location other than your computer or other internet connected device, most often, this will be a distant datacenter. Cloud Computing offers utility-oriented IT services to the users worldwide over the internet. As compared to grid computing, the problem of resource management is transformed into resource virtualization and allocations. For cloud computing based services, users consume the services when they need to, and pay only for what they use. One implication of Cloud platforms is the ability to dynamically adapt (scale-up or scale-down) the amount of resources provisioned to an application in order to attend variations in demand that are either predictable, and occur due to access patterns observed during the day and during the night; or unexpected, and occurring due to a subtle increase in the popularity of the application service. Cloud computing is internet based computing where virtual shared servers provide software, infrastructure, platform, devices and other resources. Clouds are a large pool of easily usable and accessible virtualized resources. These resources can be dynamically re-configured to adjust to a variable load (scale), allowing also for optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized service level agreements. Cloud Computing is the result of evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. These applications often exhibit transient behavior (usage pattern) and have different QoS requirements depending on time criticality and users' interaction patterns (online/offline). Hence, the development of dynamic provisioning techniques to ensure that these applications achieve QoS under transient conditions is required.

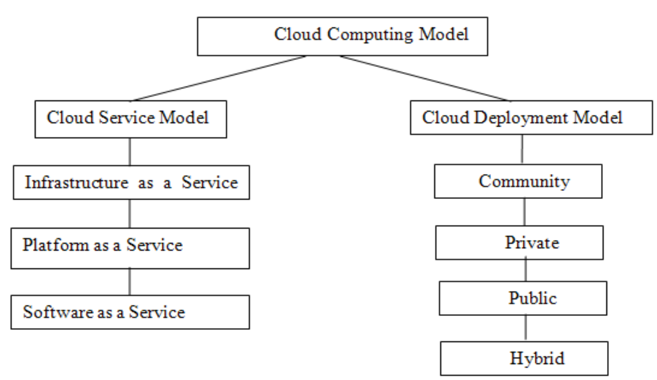

Fig. 1 Cloud Computing Model

Gaganpreet Kaur is a student of Shaheed Udham Singh College of Engineering And Technology, Tangori. Punjab (India).(e-mail: gaganb6@gmail.com)

Er. Yogesh Kumar, Assistant Professor, Computer Science & Engineering Deptt., Shaheed Udham Singh College Of Engineering And Technology, Tangori Punjab (India) (e-mail: yogesh_84kumar@yahoo.com).

Even though Cloud has been increasingly seen as the platform that can support elastic applications, it faces certain limitations pertaining to core issues such as ownership, scale, and locality. For instance, a cloud can only offer a limited number of hosting capability (virtual machines and computing servers) to application services at a given instance of time, hence scaling application's capacity beyond a certain extent becomes complicated. Therefore, in those cases where the number of requests overshoots the cloud's capacity, application hosted in a cloud can compromise on overall QoS delivered to its users. One solution to this problem is to inter-network multiple clouds as part of a federation and develop next-generation dynamic provisioning techniques that can derive benefits from the architecture. Such federation of geographically distributed clouds can be formed based on previous agreements among them, to efficiently cope with variation in services demands. This approach allows provisioning of applications across multiple clouds that are members of a/the federation. This further aids in efficiently fulfilling user SLAs through transparent migration of application service instance to the cloud in the federation, which is closer to the origins of requests.

There are the following Cloud Computing Services:

Cloud computing providers offer different types of services according to several fundamental models: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) where IaaS is the most basic and each higher model abstracts from the details of the lower models.

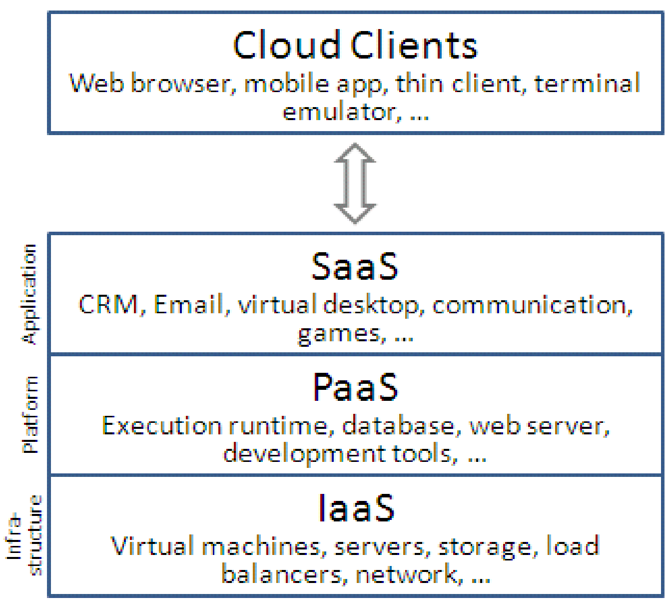

Fig. 2 Cloud Service Levels

A. Infrastructure as a Service (IaaS)

Providers of IaaS offer computers - physical or (more often) virtual machines - and other resources. IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw (block) and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on-demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks). Examples of IaaS providers include: Amazon EC2, Google Compute Engine, HP Cloud, Joyent, Linode, NaviSite, Rackspace, and ReadySpace Cloud Services.

B. Platform as a Service (PaaS)

In the PaaS model, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. Examples of PaaS include: AWS Elastic Beanstalk, Cloud Foundry, Heroku, Force.com, EngineYard, Mendix, OpenShift, Google App Engine, AppScale, Windows Azure Cloud Services, OrangeScape and Jelastic.

C. Software as a Service (SaaS)

In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. Cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. Examples of SaaS include: Google Apps, Microsoft Office 365, Petrosoft, Onlive, GT Nexus, Marketo, Casengo, TradeCard, Salesforceand CallidusCloud. These services are available to user in a Pay-per- use –on demand model..

II. Workflow Scheduling

The advent of Cloud computing as a new model of service provisioning in distributed systems, encourages researchers to investigate its benefits and drawbacks in executing scientific applications such as workflows. There are a mass of researches on the issue of scheduling in cloud computing, most of them, however, are about workflow and job scheduling. The scheduling entails the selection of the services and the appropriate start time for each workflow.

Workflow scheduling is the problem of mapping each task to appropriate resource and allowing the tasks to satisfy some performance criterion. A workflow consists of a sequence of concatenated (connected) steps. Workflow mainly focused with the automation of procedures and also in order to achieve a overall goal thereby files and data are passed between participants according to a defined set of rules. A workflow enables the structuring of applications in a directed acyclic graph form where each node represents the task and edges represent the dependencies between the nodes of the applications.

A single workflow consists of a set of tasks and each task communicates with another task in the workflow. Workflows are supported by Workflow Management Systems. Workflow scheduling discovers resources and allocates tasks on suitable resources. Workflow scheduling plays a vital role in the workflow management. Proper scheduling of workflow can have an efficient impact on the performance of the system. For proper scheduling in workflows various scheduling algorithms is used.

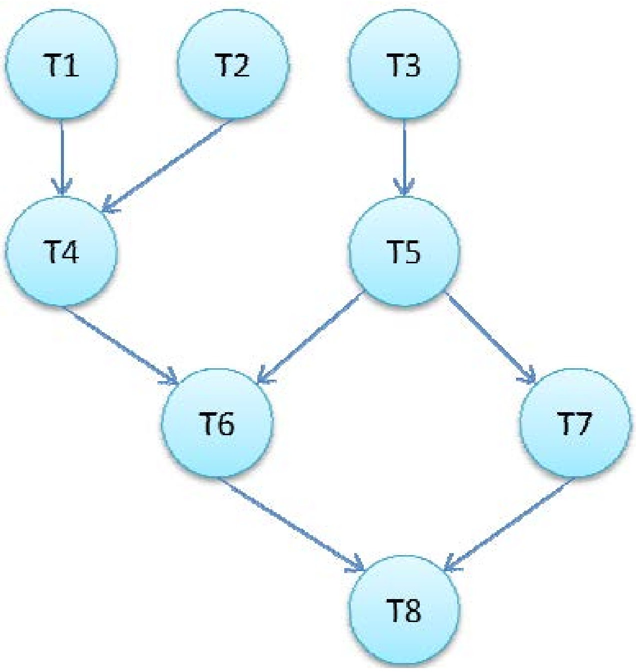

Fig. 3 Workflow Scheduling

Different architecture may support the workflow scheduling infrastructure. In particular, there are three major categories in which these architecture may fall: centralized, hierarchical and decentralized.

In the centralized scheme there is a unique, central scheduler that decides the mapping for every task in the workflow. The hierarchical approach on the other hand, defines a main scheduler which manages a set of low level schedulers in charge of a subset of the workflow tasks. Finally, in the decentralized architecture only a set of independent, decentralized schedulers exist and each of them is responsible of a sub workflow.

The decision to map a task to a resource can be made based on the information available to the scheduler. There are two main types of decisions, local and global. Local decisions are made based solely on the information of the single task (or group of tasks) being handled by a particular scheduler. Conversely, global decisions are those made considering the entire workflow as opposed to a single, isolated task.

III. Related Work

A. *Problem Formulation*

A great deal of research over the past several years has been devoted to the development of methodologies for scheduling the workflows in cloud computing environment. Many workflow scheduling algorithms have been proposed which have reduced the cost and time of execution. The problem is to propose a DBD-CTO algorithm, which is deadline and budget distribution based cost time optimization scheduling algorithm, which minimizes the execution cost and yet meets the time and budget constraints imposed by user. This algorithm will be implemented on workflow simulator which is the extension of cloud simulator. Our approach is to compare the results obtained from the proposed algorithm with the existing one like Greedy algorithm, priority based algorithm, max-min algorithm, min-min algorithm and deadline based algorithm and find that which method is better to minimize the cost and execution time of scheduling..

B. *Objectives*

The following objectives are to be covered under the above title:

1) To implement Cost-time Optimization Workflow Scheduling based on Deadline and Budget Distribution.
2) To enhance the existing technique by simulating the algorithm on workflow simulator which is the extension of cloud sim.
3) Comparison of the results obtained from proposed method with different existing algorithms (Greedy algorithm, priority based algorithm, max-min algorithm, min-min algorithm and deadline based algorithm) for workflow scheduling.

C. *Proposed Algorithm*

We proposed DBDCTO algorithm for dependent task. It is more efficient algorithm that reduce the time and cost of dependent tasks.

1) It discovers available services and request QoS parameters of services for every task.
2) Group workflow tasks into task partitions: We categorize workflow tasks to be either a synchronization task or a simple task. A synchronization task is defined as a task which has more than one parent or child task. Rests of the task are known as simple task .The workflow is partitioned such that a set of simple tasks that are executed sequentially between two synchronization tasks.
3) Estimates the minimum execution time and cost for each task from the available set of services.
4) Calculate the total expected completion time by summing the data processing time and minimum execution time.

We run this algorithm on workflow simulator to achieve better results by comparing proposed DBDCTO algorithm with the various existing algorithms.

IV. Methodology

The method followed to carry out the research work is:

1) Perform the literature survey of the different algorithms used for Scheduling in Cloud computing.
2) Propose the DBD-CTO Scheduling Algorithm which

will minimize the execution cost and time of scheduling.

3) Java will be used as an implementation tool to implement the proposed algorithm.
4) Simulate the DBD-CTO algorithm on workflow simulator.
5) Simulate the existing algorithms like Greedy algorithm, priority based algorithm, max-min algorithm, min-min algorithm and deadline based algorithm on workflow simulator.
6) Analyse the performance of proposed algorithm with respect to different existing algorithms on the basis of parameters like number of cloudlets, status of task, start time, finish time, total time, depth or level or cost.

V. RESULTS

The task consolidation that is also known as server/workload consolidation problem is the process of assigning a set of n tasks (that can be service requests or simply services) to a set of m workflow resources, without violating time constraints. The main purpose is to develop workflow schedule such that it minimizes the execution cost and yet meet the time constraints imposed by the user. In order to solve scheduling problems efficiently, we partition workflow tasks. Along with workflow partition, the deadline and budget constraint assignment strategy is also discussed to distribute the overall deadline and budget over each partition.

TABLE I
RESULTS OF DBDCTO

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Execution Cost($)	Deadline in seconds(User Value)	Budget $(User Value)
92	SUCCESS	2	0	16.59	62.06	78.65	19.91	550	1000
4	FAILED	2	0	534.18	171.31	705.49	641.02	550	1000
2	FAILED	2	1	534.29	171.31	705.6	641.15	550	1000
0	FAILED	2	2	534.61	171.31	705.91	641.53	550	1000
3	FAILED	2	3	534.71	171.31	706.02	641.66	550	1000
1	SUCCESS	2	4	535.14	171.31	706.44	642.16	550	1000
8	FAILED	2	5	533.87	179.41	713.28	640.65	550	1000
6	SUCCESS	2	6	534.2	179.41	713.61	641.05	550	1000
5	FAILED	2	7	535.05	179.41	714.46	642.06	550	1000
9	FAILED	2	8	536.3	179.41	715.71	643.56	550	1000
7	FAILED	2	9	538.83	179.41	718.24	646.6	550	1000

TABLE II
RESULTS OF DTC

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Execution Cost($)	Deadline in seconds(User Value)	Budget $(User Value)
92	SUCCESS	2	0	17.66	62.06	80.08	19.91	550	1000
4	FAILED	2	4	534.34	171.31	705.87	639.6	550	1000
0	FAILED	2	3	534.72	171.31	706.68	640.21	550	1000
3	SUCCESS	2	2	536.87	171.31	707.97	642.29	550	1000
2	FAILED	2	1	537.6	171.31	708.76	642.82	550	1000
1	SUCCESS	2	0	538.32	171.31	710.13	644.62	550	1000
8	FAILED	2	9	535.05	179.41	714.62	640.29	550	1000
5	FAILED	2	8	534.96	179.41	714.94	640.5	550	1000
9	FAILED	2	7	535.6	179.41	715.13	640.56	550	1000
7	FAILED	2	6	536.2	179.41	714.8	641.09	550	1000
6	SUCCESS	2	5	535.54	179.41	715.15	641.42	550	1000

TABLE III
RESULTS OF GREEDY COST

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Execution Cost($)	Deadline in seconds(User Value)	Budget $(User Value)
92	SUCCESS	2	0	20.1	62.06	80.47	19.91	550	1000
1	FAILED	2	1	536.73	171.31	707.07	640.27	550	1000
2	FAILED	2	2	535.05	171.31	708.26	640.62	550	1000
4	FAILED	4	4	536.42	171.31	709.78	641.69	550	1000
3	FAILED	2	3	536.44	171.31	709.99	641.97	1000	1000
0	FAILED	2	0	540.9	171.31	711.05	645.1	550	1000
8	FAILED	2	8	537.26	179.41	717.78	641.35	550	1000
5	FAILED	2	5	536.74	179.41	715.76	641.58	550	1000
9	FAILED	2	9	537.74	179.41	715.4	641.62	550	1000
7	FAILED	2	7	538.84	179.41	718.66	642.45	550	1000
6	FAILED	2	6	539.5	179.41	718.46	642.86	550	1000

TABLE IV
RESULTS OF MAX-MIN

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Execution Cost($)	Deadline in seconds(User Value)	Budget $(User Value)
92	SUCCESS	2	0	18.22	62.06	79.02	19.91	550	1000
3	FAILED	2	4	535.35	171.31	706.32	640.15	550	1000
2	FAILED	2	3	535.16	171.31	705.62	640.77	550	1000
4	FAILED	2	2	534.1	171.31	706.96	640.89	550	1000
0	FAILED	2	1	535.72	171.31	706.71	641.66	550	1000
1	FAILED	2	0	536.82	171.31	708.71	643.2	550	1000
5	FAILED	2	9	534.58	179.41	714.5	640.04	550	1000
9	FAILED	2	8	534.33	179.41	179.41	640.53	550	1000
6	FAILED	2	7	536.85	179.41	715.04	642.74	550	1000
8	FAILED	2	6	536.83	179.41	716.84	643.17	550	1000
7	SUCCESS	2	5	537.15	179.41	716.27	644.03	550	1000

TABLE V

RESULTS OF MIN-MIN

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	Execution Cost($)	Deadline in seconds(User Value)	Budget $(User Value)
92	SUCCESS	2	0	17.74	62.06	81.58	19.91	550	1000
1	FAILED	2	0	535.46	171.31	707.23	640.47	550	1000
4	FAILED	2	1	536.75	171.31	707.88	641.37	550	1000
3	FAILED	2	2	535.79	171.31	707.42	641.64	550	1000
0	FAILED	2	3	538.63	171.31	709.57	643.56	550	1000
2	FAILED	2	4	539.28	171.31	711.08	644.39	550	1000
9	SUCCESS	2	5	535.23	179.41	715.61	640.02	550	1000
8	FAILED	2	6	536.85	179.41	717.11	641.79	550	1000
5	SUCCESS	2	7	536.42	179.41	717.46	642.07	550	1000
6	FAILED	2	8	537.15	179.41	716.56	642.51	550	1000
7	SUCCESS	2	9	538.45	179.41	717.7	642.95	550	1000

VI. CONCLUSION

This thesis have concluded that the proposed approach leads to the better results in making the complete cloud computing resources that represents a Deadline and Budget distribution based Cost and Time optimization workflow scheduling algorithm that minimizes the cost and time of execution while meeting the deadline and budget constraint specified by the user. In clouds, DBDCTO algorithm is more efficient algorithm that reduce the time and cost of dependent tasks. It discovers available services and request QoS parameters of services for every task. Group workflow tasks into task partitions: We categorize workflow tasks to be either a synchronization task or a simple task. A synchronization task is defined as a task which has more than one parent or child task. Rests of the task are known as simple task .The workflow is partitioned such that a set of simple tasks that are executed sequentially between two synchronization tasks.

VII. FUTURE SCOPE

The proposed approach help us to give the better result with respect to DTC, Greedy Cost, MaxMin and MinMin workflow scheduling algorithm to minimize the cost and time of execution while meeting the deadline and budget constraint specified by the user . In future, we can develop a new algorithm that minimizes the execution cost and execution time and give better result than proposed DBDCTO algorithm. It can be adjusted depending on the system load. Changes can be made in the some parameters of the proposed algorithm.

REFERENCES

[1] Junwei, G.E, Yongsheng, Y., (2013), "Research Of Cloud Computing Task Scheduling Algorithm Based On Improved Genetic Algorithm", International Conference on Computer Science and Electronics Engineering.

[2] Xavier, S., Lovesum, S.P.J., (2013), "A Survey of Various Workflow Scheduling Algorithms in Cloud Environment", International Journal of Scientific and Research Publications, Volume 3, Issue 2, ISSN 2250-3153.

[3] Salot, P., (2013), "A Survey Of Various Scheduling Algorithm In Cloud Computing Environment", IJRET, Volume 2, Issue 2, ISSN 2319 – 1163, page no. 131 – 135.

[4] Bessai, K., Youcef, S., Oulamara, A., Godart, C., Nurcan, S., (2012), "Bi-criteria Workflow Tasks Allocation and Scheduling in Cloud Computing Environments", IEEE Fifth International Conference on Cloud Computing.

[5] Elghoneimy, E., Bouhali, O., Alnuweiri, H., (2012), "Resource Allocation and Scheduling in Cloud Computing", Workshop on Computing, Networking and Communications.

[6] Malik, S., Huet, F., Caromel, D., (2012), "Reliability Aware Scheduling in Cloud Computing", International Conference for Internet Technology and Secured Transactions.

[7] Bardsiri, A. K., Hashemi, S. M., (2012), "A Review of Workflow Scheduling in Cloud Computing Environment", International Journal of Computer Science and Management Research, Vol. 1, Issue 3, ISSN 2278-733X.

[8] Srinivasan, P.K., (2012), "Time-Cost Scheduling Algorithm", International Conference on Computing and Control Engineering.

[9] J. Tordsson, R. Montero, R. Moreno-Vozmediano,and I. Llorente, (2012), "Cloud Brokering Mechanisms For Optimized Placement Of Virtual Machines Across Multiple Providers", Future Generation Computer Systems, Volume 28, Issue 2, pp. 358-367.

[10] Yuqi, W.U., Guosun, Z., Yuan, Z., (2012), "Trend Analysis for Scheduling Algorithm in Cloud Computing", Journal of Microelectronics & Computer, vol. 29, no. 9, pp.103-107.

[11] Xiaoyong, X., Yu, P., (2012), "Power-Aware Resource Scheduling Under Cloud Computing Environment", Journal of Computer Applications vol. 32, no. 7, pp.1913-1915.

[12] Genez, T.A.L., Bittencourt, L.F., Madeira, E.R.M., (2012), "Workflow Scheduling for SaaS / PaaS Cloud Providers Considering Two SLA Levels", IEEE Network Operations and Management Symposium (NOMS): Mini-Conference, pp. 906-912.

[13] Lee, Z., Wang, Y., Zhou, W., (2011), "A Dynamic Priority Scheduling Algorithm on Service Request Scheduling in Cloud Computing", International Conference on Electronic & Mechanical Engineering and Information Technology.

[14] Zhangjun, W., Xiao, L., Zhiwei, N., Dong, Y., and Yun, Y. (2011). "A Market-Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems", Journal of Supercomputing, DOI 10.1007/s11227-011-0578-4.

[15] Bossche, R.V.D., Vanmechelen, K. and Broeckhove, J., (2011), "Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds", on Cloud Computing Technology and Science, IEEE International Conference, pp. 320-327.

[16] Liu, H., Xu, D., Miao, H., (2011), "Ant Colony Optimization Based Service flow Scheduling with Various QoS Requirements in Cloud Computing", First ACIS International Symposium on Software and Network Engineering, pp. 53-58.

[17] Lin, C., Lu, S., (2011), "Scheduling Scientific Workflows Elastically for Cloud Computing", in IEEE 4th International Conference on Cloud Computing, pp. 246-247.

[18] Selvarani, S., Dr. Sadhasivam, G.S., (2010), "Improved Cost-Based Algorithm For Task Scheduling in Cloud Computing", IEEE conference.

[19] G. Juve and E. Deelman, (2010), " Scientific Workflows in the Cloud", In Grids,Clouds and Virtualization, M. Cafaro Eds. Springer, pp. 71-91.

[20] Ke, L., Hai, J., Jinjun, C., Xiao, L., Dong, Y., and Yun, Y. (2010). "A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on Cloud Computing Platform", International Journal of High Performance Computing Applications, 1-16.

[21] Bossche, R.V.D., Vanmechelen, K. and Broeckhove, J., (2010), "Cost-Optimal Scheduling In Hybrid Iaas Clouds For Deadline Constrained Workloads", In Proceedings of the 3rd IEEE international conference on cloud computing, pp. 228-235,

[22] Pandey, S., Wu, L., Guru, S. and Buyya, R., (2010) ''A Particle Swarm Optimization Based Heuristic For Scheduling Workflow Applications In Cloud Computing Environments'', 24th IEEE Int'l Conference on

Advanced Information Networking and Applications (AINA), Perth, Australia, pp. 400–407.

[23] Salehi, M.A. and Buyya, R., (2010) ‘‘Adapting Market-Oriented Scheduling Policies For Cloud Computing’’, Proceedings of the 10th Int’l Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010), Busan, Korea,pp. 351–362 .

[24] Han, Y.H, Kim, C.M, Gil, J.M. (2010) “A Greedy Algorithm for Target Coverage Scheduling in Directional Sensor Networks”, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 1, number: 2/3, pp. 96-106.

[25] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., and Brandic, I. (2009). “Cloud computing and Emerging IT platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility”, Journal of Future Generation Computer Systems, vol.25, no. 6, 599–616.

[26] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I., (2009), ‘‘Cloud computing and emerging IT platforms: vision, hype and reality for delivering computing as the 5th utility’’, Future Gener. Comput. Syst., 25(6), pp. 599–616.

[27] Juve, G., Deelman, E., Vahi, K., Mehta, G., Berriman, B., Berman, B.P. and Maechling, P., (2009), ‘‘Scientific workflow applications on Amazon EC2’’, 5th IEEE International Conference on e-Science, Oxford, UK.