Comparison of Workflow Scheduling Algorithms in Cloud Computing

Navjot Kaur
CSE Department, PTU Jalandhar
LLRIET
Moga, India

Taranjit Singh Aulakh
CSE Department , PTU Jalandhar
BGIET
Sangrur, India

Rajbir Singh Cheema
IT Department, PTU Jalandhar
LLRIET
Moga, India

***Abstract*— Cloud computing has gained popularity in recent times. Cloud computing is internet based computing, whereby shared resources, software and information are provided to computers and other devices on demand, like a public utility. Cloud computing is technology that uses the internet and central remote servers to maintain data and applications. This technology allows consumers and businesses to use application without installation and access their personal files at any computer with internet access. The main aim of my work is to study various problems, issues and types of scheduling algorithms for cloud workflows as well as on designing new workflow algorithms for cloud Workflow management system. The proposed algorithms are implemented on real time cloud which is developed using Microsoft .Net Technologies. The algorithms are compared with each other on the basis of parameters like Total execution time, Execution time for algorithm, Estimated execution time. Experimental results generated via simulation shown that Algorithm 2 is much better than Algorithm 1, as it reduced makespan time.**

Keywords- Cloud Computing; Workflows; Scheduling; Makespan; Task ordering; Resource Allocation.

I. INTRODUCTION

Cloud computing is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like a public utility. Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth.

A. Workflows

The WfMC (Workflow Management Coalition) defined a workflow as "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."

WfMC published its reference model in [1], identifying the interfaces within this structure which enable products to interoperate at a variety of levels. This model defines a workflow management system and the most important system interfaces (see Fig 1).

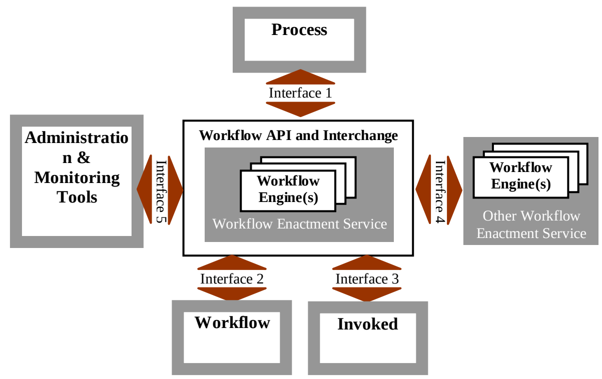

Fig. 1 WfMC's Workflow Reference Model

1) Workflow Engine. A software service that provides the run-time environment in order to create, manage and execute workflow instances.

2) Process Definition. The representation of a workflow process in a form which supports automated manipulation.

3) Workflow Interoperability. Interfaces to support interoperability between different workflow systems.

4) Invoked Applications. Interfaces to support interaction with a variety of IT applications.

5) Workflow Client Applications. Interfaces to support interaction with the user interface.

6) Administration and Monitoring. Interfaces to provide system monitoring and metric functions to facilitate the management of composite workflow application environments.

It can be seen that scheduling is a function module of the Workflow Engine(s), thus it is a significant part of workflow management systems.

The rest of the paper is structured as follows: Related work is discussed in Section II. Then section III describes our Proposed Work. The Implementation is presented in Section IV. And Section V will show the experimental details and simulation results. Finally Section VI includes the future scope of our research work.

II. RELATED WORK

A. Cloud Platforms

A comprehensive survey of cloud computing is defined by number of researchers. There are no. of definitions of cloud computing. According to R. Buyya and S.Venugopal[5] Cloud computing is defined as " *a* type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers".

Sun Microsystems [3] takes an inclusive view that there are many different types of clouds like public cloud, private cloud, hybrid cloud .Many different applications that can be built by using these different clouds.

Recently, several academic and industrial organizations have started investigating and developing technologies and infrastructure for Cloud Computing.

B. Workflow Management Systems

Workflow is concerned with the automation of procedures whereby files and data are passed between Participants according to a defined set of rules to achieve an overall goal. A workflow management system defines, manages and executes workflows on computing resources. Workflow Scheduling: workflow scheduling is a kind of global task scheduling as it focuses on mapping and managing the execution of inter-dependent tasks on shared resources that are not directly under its control. Workflow management includes five dimensions: *time*, *cost*, *fidelity*, *reliability* and *security*.

The related work done in workflow management system is shown below in tabular form (see Table III):

TABLE III
SURVEY ON WORKFLOW MANAGEMENT SYSTEM

S. No.	Citation	Brief Introduction About Paper
1.	**Authors:** S. Elnikety, E. Nahum, J. Tracey, W. Zwaenepoel **Year:** 2004	This paper [10] consider workflows that are invoked via web requests. The workflows are part of a web application that spans multiple resources in the grid.
2.	**Authors:** Jia Yu, Rajkumar Buyya & Chen Khong Tham **Year:** 2005	In this paper, a cost-based workflow scheduling algorithm is proposed [11] that minimizes execution cost while meeting the deadline for delivering results.
3.	**Author:** E. Deelman, G. Singh, D.S. Katz **Year:** 2005	Pegasus [12], is proposed which is a framework that maps complex scientific workflows onto distributed resources such as the Grid. DAGMan, together with Pegasus, schedules tasks to Condor system.
4.	**Author:** Jia Yu, Rajkumar Buyya **Year:** 2006	A budget constraint based scheduling is proposed [13], which minimizes execution time while meeting a specified budget for delivering results. A new type of genetic algorithm is developed to solve the scheduling optimization problem and the scheduling algorithm is tested in a simulated Grid tested.
5.	**Author:** Patel, Y. Darlaington **Year:** 2006	According to this paper, there are two categories of workflow scheduling [14]. The first category is based on the real time data such as waiting time in the queue or the shortest remaining execution length. The second category is based on average metrics such as mean arrival time, or mean execution length.
6.	**Author:** P. Patala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singal **Year:** 2007	A control system is developed that adjusts the resource sharing among applications to ensure the desired QoS and maintains the high resource utilization [15].
7.	**Author:** J. Yu & R. Buyya **Year:** 2007	The work presented in this paper defines two major types of workflow scheduling [16], best-effort based and QoS constraint based scheduling, primarily for grid workflow management systems.
8.	**Author:** Zhifeng Yu & Weisong Shi **Year:** 2008	In this paper, a planner-guided strategy is presented for multiple workflows [17]. It ranks already tasks and decides which task should be scheduled.
9.	**Author:** Ke Liu, Jin Jun chin, Yun Yang & Hai Jin **Year:** 2008	A throughput maximization strategy is proposed [18] for scheduling transaction intensive workflows. But it is designed for transaction intensive workflows not for multiple workflows.
10.	**Author:** Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi **Year:** 2009	Authors of this paper worked on Multiple workflow and Multiple QOS.A strategy [19] is implemented for Multiple Workflow Management system with multiple QOS. The Scheduling access rate is increased by using this strategy.
11.	**Author:** Boris Mejias, Peter Van roy **Year:** 2010	In this paper, researchers proposed an architecture to organize a set of mini-clouds provided by different institutions, in order to provide a larger cloud that appears to its users as a single one[20].
12.	**Author:** J. Kosinska, J. Kosinski, K. Ziehnski **Year:** 2010	Purpose of this paper is to discuss various forms of mapping cluster topology requirements into cloud environments to achieve higher reliability & scalability of application executed within cloud resources[21].
13.	**Author:** M. Jensen, J.schwenk, J.M. Bohli, L.L. Iacono **Year:** 2011	This paper initiates discussion by contributing a concept which achieves secutrity merits by making use of multiple distinct clouds at the same time[22].

III. PROPOSED WORK

This section presents a set of scheduling algorithms, based on Time management [23]. The aim of the algorithms is to optimize the makespan, which is defined as the maximum time taken for the completion of all the tasks in a given application. The proposed algorithms are implemented using a service based cloud and comparative results are shown.

The problem of scheduling a set of tasks to a set of processors can be divided into two categories:

- Job scheduling
- Job mapping and scheduling

In the former category, independent jobs are to be scheduled among the processors of a distributed computing system to optimize overall system performance. In contrast, the mapping and scheduling problems requires the allocation of multiple interacting tasks of a single parallel program in order to minimize the completion time on the parallel computer system.

To generate the schedule, our technique is based on the traditional list scheduling approach in which we construct a list and schedule the nodes on the list one by one to the processors.

A. Algorithm 1

The design of our algorithm 1 is basis on the following heuristics. It is based on the POSEC method [23]. POSEC is an acronym for Prioritize by Organizing, Streamlining, Economizing and Contributing. The objective of our algorithms is efficient time management and load balancing.

There are Four Quadarnts of Descion Making :

a) Level 1: Low Urgency & Low Importance

b) Level 2: Low Urgency & High Importance

c) Level 3: High Urgency & Low Importance

d) Level 4: High Urgency & High Importance

There are Four Quadarnts of Descion Making : It needs two types of Priority Scores to take descion , Urgency Score and Importance Score. Urgency Score given by Cluster Member of cloud. Importance Score is given by Cloud Resources Manager .

Urgency Score is Calculated on the scale of 10 on the basis of the following table.

TABLE IIV
CLASSIFICATION OF URGENCY LEVELS

Level of Severity	**Description of Severity**	**Initial Response Within**	**Score**
Level I	Production application down or major malfunction causing business revenue loss resulting in majority of users unable to perform their normal functions	1 hour	7.6<=Score<=10
Level II	Critical loss of application functionality or performance resulting in high number of users unable to perform their normal functions	4 hours	5.1<=Score<=7.5
Level III	Moderate loss of application functionality or performance resulting in multiple users impacted in their normal functions	8 hours	2.6<=Score<=5.0
Level IV	Minor loss of application functionality or product feature question	24 hours	1>=Score<=2.5

Importance Score is Calculated by the Resource manager and its also on the scale of 10. The various parameter of resource cheking are CPU time. Threades etc. we have use resource monitor program to generate the importance score. High Importance means the Resources are available. Low Importance means based the Resources are Not available .

It is assumed that job consist of tasks. The cloud scheduler assigns these tasks to resources. Also it is assumed that each computational resource can run one application at a time, and must run that application to completion.

Let T be a set of n tasks and m is the number of computational resources in a cloud. We define a schedule of T as follows: A schedule S of T onto a cloud with m resources is a finite set of tuples<v, p, t> where v is the schedule, t is the starting time, and p is the resource.

To generate the schedule, our technique is based on the traditional list scheduling approach in which we construct a list and schedule the nodes on the list one by one to the processor. The list is constructed by ordering the jobs according to their urgency score s. The list is static therefore the order of nodes on the list will not change during the resource allocation process.

We restrict ourselves to non-preemptive schedules where a job once started has to run to completion on the same machine.

Scheduler has information about all resources such as processing speed (in MIPS), processing cost per second, baud rate(communication rate) and resource load during peak hours and off peak hours.

After gathering the details of user jobs, the system calculated the importance score. The jobs are executed on the values of urgency and importance score.

The time management parameters used by the algorithms are:

1) Total Execution Time: The total time consumed by the algorithm to execute all the jobs.

2) Execution Time of Algorithm: This is the time taken by the algorithm to execute.

3) Estimated Execution Time: Based on the average of total execution time parameters of previous jobs.

The proposed algorithm comprises of two parts as explained below.

A. Task Ordering Procedure, to get the schedule list

B. Resource Allocation Procedure, which allocates resources to the jobs contained in scheduling list, generated by task ordering procedure.

a) Task Ordering Procedure

Begin

Step 1: The list is initialized to be an empty list. The cloud clients calculate the urgency score according to the severity of jobs.

Step 2: The urgency score is calculated. The urgency score is based on the scale of 10.

There are 4 cases to determine the urgency score of the job.

If Level 1: 7.6 <=Score<=10

If Level II: 5.1 <=Score<=7.5

If Level III: 2.6 <=Score<=5.0

If Level IV: 1>=Score<=2.5

Step 3. According to the urgency score , the alert templates have set up

b) Resource Allocation Procedure:

Begin

Step 1: The Cloud Scheduler collects resources and their characteristics like processing speed, processing cost per second, resource load during peak hours and off peak hours.

Step 2: It generates importance score according to these characteristic on the scale of 10.

Step 3: After collecting the information about the job parameters like urgency and score, the jobs are executed according to the following case.

If Urgency High, Importance High: The email alert sent immediately.

If Urgency High, Importance low: whenever the resources are free, the email is sent on high priority basis.

If Urgency Low, Importance High: The email alert is sent after emptying the job Queue.

If Urgency low, Importance low: whenever the resources are free, and the job queue is empty, the email is sent with lower priority basis.

B. Algorithm 2

The second algorithm is based upon the Pareto Analysis [23].

Pareto Analysis

This is the idea that 80% of tasks can be completed in 20% of the disposable time. The remaining 20% of tasks will take up 80% of the time. This principle is used to sort tasks into two parts. According to this form of Pareto analysis it is recommended that tasks that fall into the first category be assigned a higher priority.

The **80-20-rule** can also be applied to increase productivity: it is assumed that 80% of the productivity can be achieved by doing 20% of the tasks. If productivity is the aim of time management, then these tasks should be prioritized higher.

For example, look at your to do list**-** if you have 10 tasks on there then two of those tasks will yield 80% of your results. Alternatively, 80% of income is owned by 20% of people **-** it works both ways!

The Pareto principle holds across business, academia, politics, and a number of other areas. The foundation of this time management skill is that: **20% of tasks yield 80% of results**

This algorithm is also comprised of two parts.

1. Task Ordering Procedure, to get the schedule list

2. Resource Allocation Procedure, which allocates resources to the jobs contained in scheduling list, generated by task ordering procedure.

a) Task Ordering Procedure

Begin

Step 1: The list is initialized to be an empty list. The cloud clients send the jobs to the cloud manager according to their priority.

Step 2: The urgency score is calculated. The urgency score is based on the scale of 10.

There are 4 cases to determine the urgency score of the job.

If Level 1: 7.6 <=Score<=10

If Level II: 5.1 <=Score<=7.5

If Level III: 2.6 <=Score<=5.0

If Level IV: 1>=Score<=2.5

Step 3. According to the urgency score , the alert templates have set up

b) Resource Allocation Procedure:

Step 1: From the previous set of jobs, importance score is calculated. The score is based upon the following method:

Let t = (time to execute jobs/estimated time) %

If t = 100, then the cloud resources are utilized properly. A high score of importance is send by the algorithm.

It t <100 and t>80, the cloud manages resources are overloaded; but according to 80:20 rule by paleto, a high importance score is generated.

If t<80, a low importance score is generated.

If t> 100, the cloud manager resources are underutilization, a high importance score is generated by the cloud.

Step 2: According to the importance score and urgency score, calculated, the jobs are executed by the cloud manager.

If Urgency High, Importance High: The email alert sent immediately.

If Urgency High, Importance low: whenever the resources are free, the email is sent on high priority basis.

If Urgency Low, Importance High: The email alert is sent set after emptying the job Queue.

If Urgency low, Importance low: whenever the resources are free, and the job queue is empty, the email is sent with lower priority basis.

IV. IMPLEMENTATION

The management and scheduling of resources in Cloud environment is complex, and therefore demands sophisticated tools for analysis the algorithm before applying them to the real system. But there are no good tools are available that serve our needs. So we develop a service based cloud using Microsoft .Net Technologies. The proposed algorithms are implemented upon this real time cloud.

Feature of This cloud:

a) To send real time email alerts to the cloud clients members like Bank, Insurance, and Hospital etc.

b) The algorithms are tested on real time cloud.

c) Google's SMTP server is used to send the mails.

d) The Database is saved on the Web server.

e) The cloud is working online. You need no special software to test the results.

f) Visual studio 2008 is used as frontend and SQL 2005 is used as Backend.

Cloud Architecture:

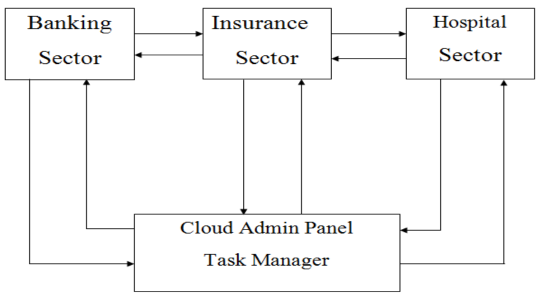

Fig. 2 Architecture of service based cloud

The cloud architecture is based upon the real time email alert system. It sends email alerts to its cluster client members.

Feature of the cloud architecture are:

a) This scenario based cloud has real life application of sending email alerts to Bank clients, Hospital clients, and Insurance company clouds. This cloud takes jobs from the all other clients with their urgency score. The cloud manager executes the jobs according to the importance score based on cloud resources.

b) The data Flow between all the clouds is using XML.

c) XML is hardware and software free technology

d) It's widely suited for cloud application.

e) There are 4 domains used in this service based cloud.

f) Moreover it's three tier architecture, the database is stored on the other server and web services execute on the other server.

g) FileZilla client FT\P application is used to upload/download data from the server.

V. Experimental Results

This section describes the experiment results obtained after implementing the scheduling algorithms. The algorithms are implemented in Microsoft .Net framework using a service based cloud. It takes as input the required set of resources and a set of tasks. The algorithms are compares with each other on set of parameters like Total execution Time, Execution time for Algorithm, Estimated Execution Time.

By Graphical Analysis of Experimental results, we analysed the simulation results using graphs. Graphical data consist of 3 cloud clients' data that are scheduled by main cloud manager. Each algorithm is run by 8 times to conclude the results:

Experiment Results of Algorithm1:

TABLE V
EXPERIMENTAL RESULTS OF ALGORITHM 1

Jobs	Total Execution Time	Total Algorithm Time	Estimated Time
1	22419.5038	432.2809	16624
2	17782.3016	575.8664	17204
3	18689.8762	868.9554	17248
4	48342.7388	19863.5854	17385
5	21839.9645	51.0341	19449
6	21720.2652	45.0113	19599
7	28639.296	329.7595	19723
8	22714.3025	123.876	20509

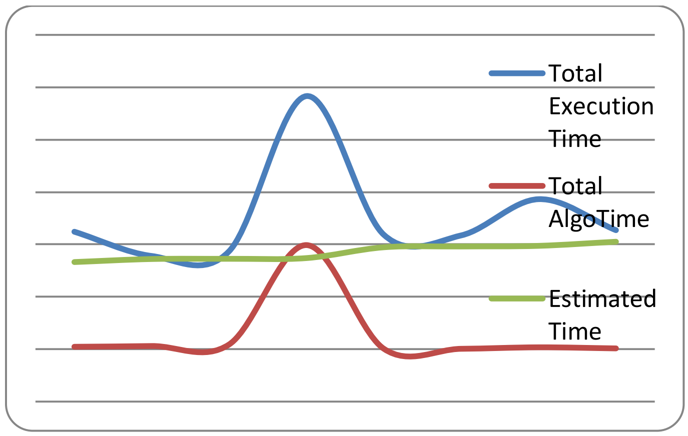

Fig. 3 Line Chart results of Algorithm 1

Experimental Results of Algorithm2:

TABLE VIII
EXPERIMENTAL RESULTS OF ALGORITHM II

Jobs	Total Execution Time	Total Algorithm Time	Estimated Time
1	15048.9774	219.5119	17120
2	14730.2754	255.9244	16861
3	17153.3982	1454.7952	17256
4	17731.0295	319.7149	17359
5	23109.6739	489.1577	20219
6	19565.4316	76.3057	20371
7	22035.7285	286.8222	20331
8	22558.6223	230.3577	20412

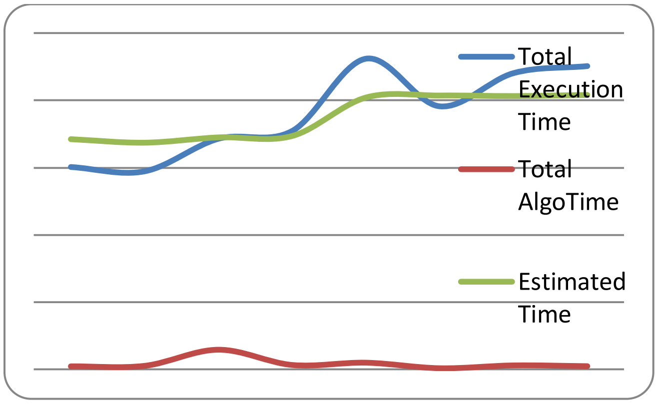

Fig. 4 Line Chart results of Algorithm 2

Makespan comparison of Algorithm 1 and Algorithm 2:

The following Graph shows the comparison of makespan time between the two proposed algorithms.

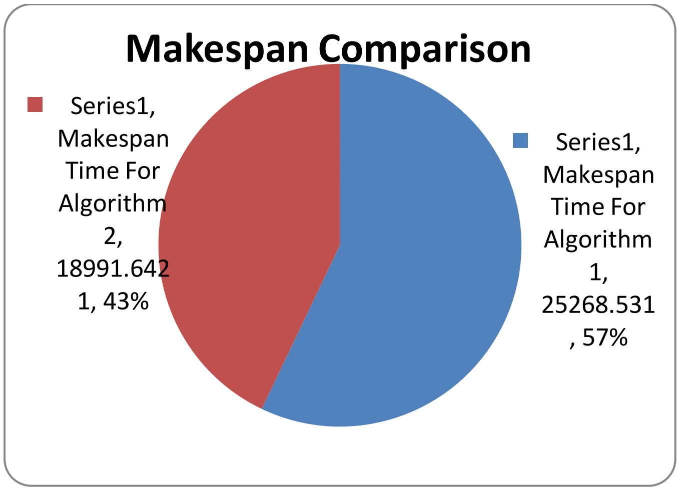

Fig. 5 Pie Chart with Makespan Comparison

VI. Future Scope

We would like to extend these algorithms to include various parameters like options for advance reservation, preemptive jobs as well. Also, in the Future we can add more clouds to this main cloud, to distribute the load work. Currently this cloud provides services like email alerts, we can also extend to store online data and providing the synchronization mechanism in this.

References

[1] Workflow Management Coalition, Workflow Management Coalition Terminology & Glossary, February 1999.

[2] Ian Foster, Yong Zhao, Ioan Raicu and Shiyong Lu, "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop 2008(GCE '08).

[3] Sun Microsystems,Inc."Introduction to Cloud Computing Architecure" Whitepaper,Ist Edition,june 2009.[Online] Availabe: www.sun.com

[4] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", Future Generation Computer Systems, Elsevier Science, Amsterdam, June 2009, Volume 25, Number 6, pp. 599-616

[5] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC 2008, IEEE CS Press, Los Alamitos, CA, USA), Sept. 25-27, 2008, Dalian, China.

[6] Microsoft Azure. http://www.microsoft.com/azure/ [30 Oct. 2010]

[7] Sun network.com (Sun Grid). http://www.network.com [18 July 2010]

[8] Amazon Elastic Compute Cloud (EC2). http://www.amazon.com/ec2/ [18 July 2010]

[9] Google App Engine. http://appengine.google.com [18 July 2010]

[10] S. Elnikety, E. Nahum, J. Tracey and W. Zwaenepoel, "A method for transparent admission control and request scheduling in e-commerce web sites," in WWW '04: Proceedings of the 13th International Conference on World Wide Web, 2004, pp. 276-286.

[11] Jia Yu, Rajkumar Buyya and Chen Khong Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids", In 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, Dec. 5-8, 2005.

[12] 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., 13(3):219–237, 2005.

[13] Jia Yu and Rajkumar Buyya, "A Budget Constrained Scheduling of Workflow Applications on Utility Grids using Genetic Algorithms",Proceedings of the 15th IEEE International symposium on High Performance Distributed Computing (HPDC 2006), IEEE CS Press,Los Alamitos, CA, USA, June 19-23, 2006, Paris, France.

[14] Patel, Y. Darlington, J., "A novel stochastic algorithm for scheduling QoS-constrained workflows in a web service-oriented grid," in Web Intelligence and International Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on, pp. 437-442.

[15] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant and K. Salem, "Adaptive control of virtualized resources in utility computing environments," SIGOPS Oper. Syst. Rev., vol. 41, pp. 289-302, 2007.

[16]J. Yu and R. Buyya, Workflow Schdeduling Algorithms for Grid Computing, Technical Report, GRIDS-TR-2007-10, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, May 2007.

[17] Zhifeng Yu and Weisong Shi, "A Planner-Guided Scheduling Strategy for Multiple Workflow Applications," icppw, pp.1-8, International Conference on Parallel Processing - Workshops, 2008.

[18] Ke Liu, Jinjun Chen, Yun Yang and Hai Jin, "A throughput maximization strategy for scheduling transaction-intensive workflows on SwinDeW-G", Concurrency and Computation:Practice and Experience, Wiley, 20(15):1807-1820, Oct.2008.

[19] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing," ispa, pp.629-634, 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, 2009.

[20] Boris Mejias, Peter Van Roy, "From Mini-clouds to Cloud Computing",2010

[21] J. Kosinska, J. Kosinski, K, Zielinski, "The Concept of Application Clustering in Cloud Computing Environments", 2010

[22] M. Jensen, J. Schwenk, J.M. Bohli, L.L. Iacono, "Security prospects through cloud computing by adopting Multiple Clouds", 2011

[23] Wikipedia .http://en.wikipedia.org/wiki/Time_management [25 march 2011].

[24] Rao, N. M. (2010). Cloud Computing Through Mobile-Learning. IJACSA - International Journal of Advanced Computer Science and Applications, 1(6).

[25] Goel, M. S., Kiran, R., & Garg, D. (2011). Impact of Cloud Computing on ERP implementations in Higher Education. IJACSA - International Journal of Advanced Computer Science and Applications, 2(6), 146-148.