World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 3.419

REAL TIME OPTIMIZATION SCHEDULING ON IAAS CLOUD SYSTEM

Deepika Saxena[1]*, Dr. R.K. Chauhan[2] and Shilpi Saxena[3]

[1]Assisstant Professor, Department of Computer Science and Applications, Kurukshetra University Kurukshetra, India.

[2]Professor, Department of Computer Science and Applications, Kurukshetra University Kurukshetra, India.

[3]Assisstant Professor, Dpartment of Computer Science and Applications, Graphic Era Hill University, Dehradun, India.

Article Received on 25/09/2016 **Article Revised on 15/10/2016** **Article Accepted on 07/11/2016**

***Corresponding Author**
Dr. Deepika Saxena
Assisstant Professor, Department of Computer Science and Applications, Kurukshetra University Kurukshetra, India.

ABSTRACT

Cloud Computing has extremely surpassed the whole world of Information Technology. Basically, cloud is a cluster of distributed and interlinked servers providing on-demand services to customers. Broadly, it offers software-as-a-service(SAAS), platform-as-a-service(PAAS), infrastructure-as-a-service(IAAS). Here we are focusing on IAAS cloud system which offers computational resources to remote customers in the form of leases. Here we are defining real-time or online optimized scheduling of requests of various resources arriving simultaneously at data center of IAAS cloud service provider. Being more practical, our algorithm is providing best resource utilization and better results in terms of execution time as compared to DFPT algorithm for task scheduling in cloud computing which do not considers dependency between tasks(requests). Our algorithm proposes dynamic task allocation on IAAS clouds and resource scheduling by utilizing the updated status of various Virtual machines available at real time. We have simulated this experiment using CloudSim toolkit. Surely, there is very beneficial improvement in results as compared to default FCFS scheduling and other available scheduling algorithms.

KEYWORDS: Cloud, CloudSim, Dependency, Resource allocation, Task, Virtual Machine.

INTRODUCTION

In this new digital world of Information Technology and Advancement, the cloud computing has broaden the world of internet to all most everyone in this world. It has transformed the complex internet based services to the simplest laymen requirement of computation. The term ``Cloud`` means a cluster of interconnected distributed datacenters which are expanded worldwide for example, Google, Amazon etc. At each datacenter, there are several computer systems acting as servers and providing enormous variety of services to the customers. Briefly describing, cloud computing is a model for enabling convenient, on-demand network access to shared pool of computing resources for example, networks, servers, storage, applications and several other services that can be rapidly provisioned and released with minimal management effort or cloud service provider interaction.[4] The basic idea of cloud computing is based on a very fundamental principal of reusability of IT capabilities. In this paper, specifically considering IaaS clouds i.e. Infrastructure as a Service which represents lowest level of cloud service providing resources and services on pay-as-per use to its customers. The ultimate goal of this Cloud System is to instantly provide resources whenever customer requires it on their laptops, PCs, mobile phones etc wherever possible. In an IaaS model of cloud computing, Cloud Service Provider provides hardware, software, servers, storage and other infrastructure components to its customers. IaaS clouds also provides various applications to its users and handle task including system maintenance, backup and resiliency planning. IaaS customers pay on a per-use basis typically by hour, week or month. Some Cloud Service Provider also charge customers based on amount of storage space used or VM space used.[5] In recent trends of IT world, cloud computing is emerging with growing popularity of its capability to provide unlimited VM space, storage space and other resources with extreme flexibility so that no adjustment is required at customer end. But practically, there is no datacenter which has unlimited capacity. Infact, to meet real time significant demand of resources, Cloud Service Provider has to make several arrangements and adjustments at their datacenters to concurrently schedule the thousand of requests and demand of resources arriving parallel at datacenter. In case of overflow of workload at datacenter, it can be shared among different datacenters or workload can be share in between public and private clouds with applying extreme security checks.[2] So, this becomes a very challenging issue to schedule several requests simultaneously at datacenter satisfying all the customers, workload balancing, reduction of energy consumption, faster execution of requests at minimum cost etc. Clearly, default FCFS scheduling cannot provide satisfactory results in all the above mentioned respect. Therefore, there is rigorous demand of an

optimized real-time scheduling technique. Though, many researchers have proposed different algorithms for scheduling in cloud computing which are providing good solutions in one or other respect, but none of them is able to completely satisfy all the different criteria of scheduling in cloud system. Broadly, in IaaS cloud system there is requirement of a scheduling technique which can satisfy all the different terms and conditions of real-time cloud computation. The scheduling technique should consider dependency and non-dependency among tasks or requests, bandwidth utility, data transfer rate, data transfer cost, response-time of VMs, workload balancing, resource sharing, CPU utilization, faster execution-time at minimum cost, how many VMs to be set up at one host computer system and many more aspects like that. Here, in our proposed work we have surely covered all these aspects and designed a satisfactory algorithm that is providing improved results in all respect.

LITRETURE SURVEY

Combined resource provisioning and scheduling strategy for executing scientific workflows on IaaS Clouds was presented which aims to minimize overall execution cost while meeting a user defined deadline.[6] This paper provides an optimized solution to scheduling problem of scientific workflows in cloud system. But it is considering single workflow execution at a time and not the multiple workflows. Various energy efficient strategies were studied which are based on approaches that specific plug-ins and energy control centers for networked large-scale hardware and software can be implemented and that they can reduce software and hardware related energy cost, improve load-balancing, reduce energy consumption due to communications, save CO_2 emissions etc.[1] This paper has considered energy saving aspects only and not noticed the execution time, execution cost and dependency between tasks in cloud computing system. In case of significant client demands, it is necessary to share workload among multiple datacenters, this would provide cheaper resources and would enlarge the capacity of resource pool.[2] This paper is very flexible and is efficient in providing improved results for task scheduling in IaaS clouds. Also, they have designed two online dynamic scheduling algorithms i.e. DCLS and DCMMS for resource allocation mechanism. In DFPT algorithm[3] weighted fair priority queue is applied to schedule task according to deadline based criteria and minimum cost-based criteria. Though results are optimized and improved in this algorithm but it has limitations like it is not suitable dependent tasks, energy saving and load balancing criteria are not defined.

PROPOSED MODEL CONCEPTS

Our proposed model includes the following concepts: Here we assume that several requests are arriving at datacenter simultaneously, these requests are termed as tasks in cloud computing. In reality, most of the request shows dependencies upon one another, so here we have schedule the tasks according to their dependencies, also complexity of computation increases along with increasing dependencies. The set of tasks which includes dependencies represents ``workflows``.

Following steps are followed in our proposed model:

1. **Preparation of list of workflows** (or set of related tasks) submitted by several users at data center**.**
2. **Independent Tasks:** All independent task at same level can execute in parallel separately on selected VMs.
3. **Dependent Tasks:** All tasks which have output dependency on another task, their transfer time and transfer cost is calculated as follows:
   Transfer Time = Data output / bandwidth utilized for data transfer
   Transfer Cost = Data output / cost of bandwidth used.
4. **Bandwidth** at same datacenter will cost same for data transfer, but if data is transferred to some external datacenter it will cost different and it will be high.
5. Whenever a task is selected from the workflow for execution, it is mandatory to execute all its predecessor task first to resolve the dependency between tasks.
6. **Execution Time and Execution Cost** of a single task is calculated as follows:
   Execution Time = Task length / Processing power of selected VM
   *Execution Cost = Cost of selected VM * Task length*
   Here task length is number of instructions of a task to be executed.
7. **Finally, Total Processing time and cost of each workflow is calculated as**:
   Total Processing Time = sum of execution time of each task in the task set + sum of all transfer time related to a task set.
   Total Processing Time = sum of execution cost of each task in the task set + sum of transfer cost of the task set.
8. **Resource allocation** is done using greedy approach of modified Cloud min-min algorithm, which selects suitable resource i.e. VM that execute the selected task at minimum cost and minimum time. Separate list of resource is maintained which keep track of the status of the VM by continuous VM monitoring at each resource allocation.

9. **Workload balancing is done as follows**: First of all, workload is calculated which depends upon the complexity and amount of computation involved in the task set. The complexity can be measured in terms of dependencies and communication among the tasks, amount of expected execution time based upon number of instructions in the task , memory space required for execution of task etc. If the calculated workload is higher than the capacity of single server, then workload is distributed among other servers at same datacenter, or if required among servers at different datacenter to balance the workload. Hence, workload is shared among distributed servers at same or different data center.

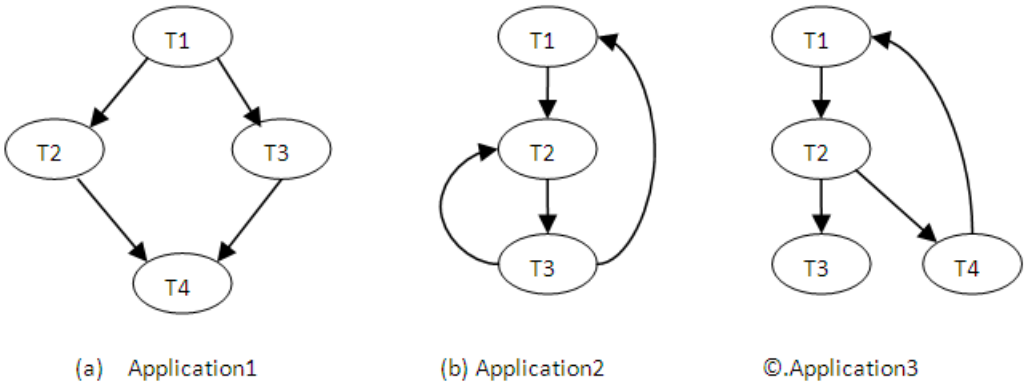

Figure 1: Applications represented as DAG.

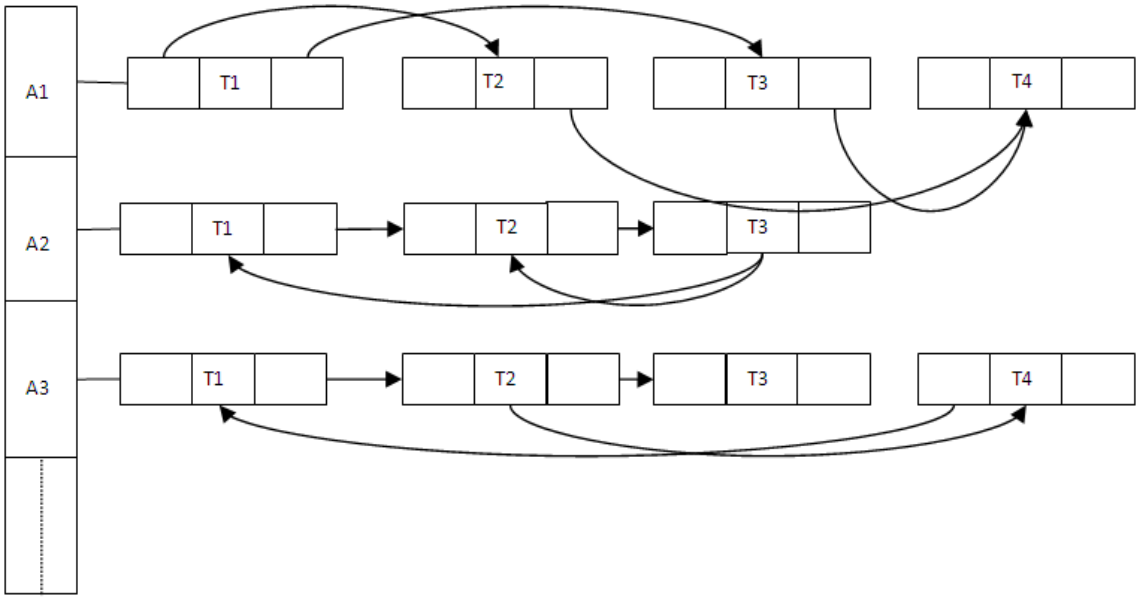

Figure 2: List of Applications.

Figure 1 and 2 are showing applications as submitted by the user, each application is a complex, so it is partitioned into set of task which are interconnected to each other in various respects, to show their relationship we have taken each application as DAG direct acyclic

graph which clearly shows the dependency among tasks as possess by the real request at the cloud server at datacenter.

METHODOLOGY

User submits applications to be executed on cloud server at datacenter as shown in figure 3. Each application is first partitioned into tasks in the form of DAG (Direct Acyclic Graph) based on dependency among tasks. Then these DAGs are placed in the list based on priorities (like deadline, cost, complexity etc) of execution using algorithm 1.

These priorities based list is then submitted at different cloud servers where they are scheduled and assigned to different VMs or resources using real-time resource allocation procedure as described in algorithm 2.

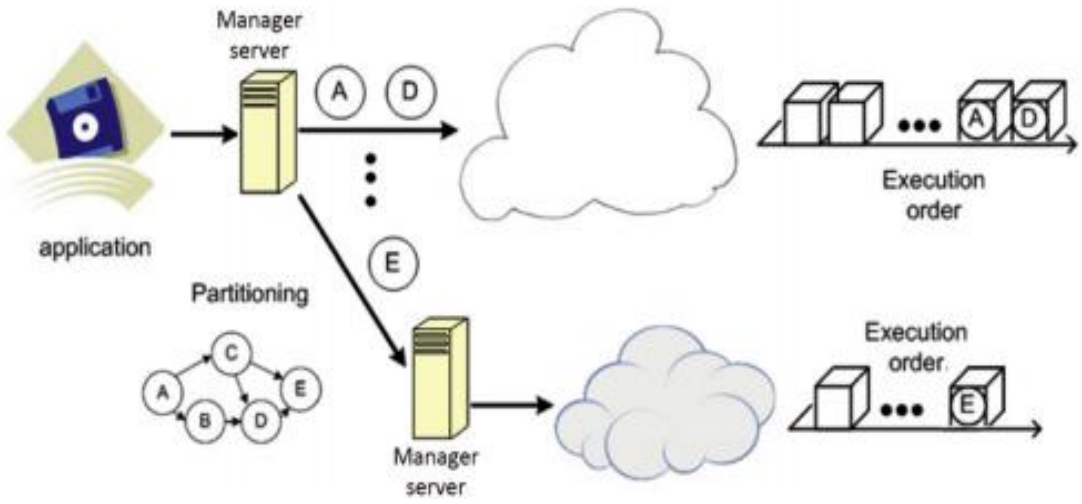

Figure 3: When an application is submitted to the cloud system, it is partitioned, assigned, scheduled and executed in the cloud system.[2]

ALGORITHM 1: Preparation of task list based on priorities.

Requirement – A DAG, expected execution time of each task in DAG

1. Deadline of every task is randomly generated.
2. Initially, all tasks are in list U.
3. Take task one by one from U and push current task node into stack S in decreasing order of their deadline.
4. While stack S is not empty do
5. If top(S) has unstacked immediate predecessor task then
6. Push immediate predecessor task node with least deadline into stack S.
7. Else push top of stack S into list P.

8. Pop top(S).
9. EndIf.
10. EndWhile.

Result- A list of task of P is prepared based on priorities.

ALGORITHM2: Real-time resource allocation Cloud min-min scheduling(RCMMS).

Requirement – Priority based list of task P, n different VMs, execution-time matrix and data transfer-time matrix.

1. Generate mappable task set P using above algorithm 1.
2. While tasks not assigned do
3. T = top(P).
4. Get resource status information from all other DataCenterBroker(DCB) or manager servers.
5. Find resource or VM with minimum ($VM_{min)}$ turnaround time (= waiting time + response time + execution time).
6. Assign task T to $VM_{min.}$
7. Remove T from P.
8. EndWhile.

Result – Selected task T is assigned for execution to VM with earliest finish time.

Performance Evaluation

In this section we have presented the experiments conducted in order to evaluate the performance of the proposed work. We utilized the CloudSim 3.1 tool to simulate the cloud environment and task sets are generated randomly using rand() function. The proposed work is compared with sequential, DFPT, EWSA algorithm for cloudlet scheduling. The configuration of the datacenter are as follows:

Table 1 shows the configuration of host and table 2 shows the configuration of VMs used in this proposed work. Here, we have used:

Number of processing element – 1

Number of host on each processor – 8 on each processor.

Table 1: Configuration of host.

RAM (MB)	2080
Processing Power (MB)	220000
VM Scheduling	Time-shared

Table 2: Configuration of VMs.

VM	RAM	Processing Power(MIPS)	Processing Element
VM1	5024	22000	1
VM2	1048	11000	1
VM3	3308	22000	1
VM4	4604	32000	1
VM5	8028	55000	1
VM6	4000	41000	1
VM7	6024	44000	1
VM8	7028	62000	1

Table 3: System Configuration.

Processor	Pentium Dual-core CPU T4400@2.20GHz
RAM	3.00GB
System Types	64 bit operating system
Operating System	Windows 7

Performance with respect to time: The experimental results shows the remarkable improvement in time over sequential, DFPT and EWSA algorithms.

Table 4: Performance of proposed methodology with respect to time.

No. of task set	FCFS	DFPT	EWSA	PRPOSED METHODOLOGY
10	803.6499	366.2881	359.4416	280.6669
20	3202.9668	1188.6538	1146.1112	660.5355
30	5564.0266	3766.8554	2881.6667	1044.6681
40	7004.3999	5044.6669	3331.1161	1846.4401
50	8963.3897	6887.4331	5842.556	3566.6664
60	10468.26666	8113.9988	7046.1134	5442.1448

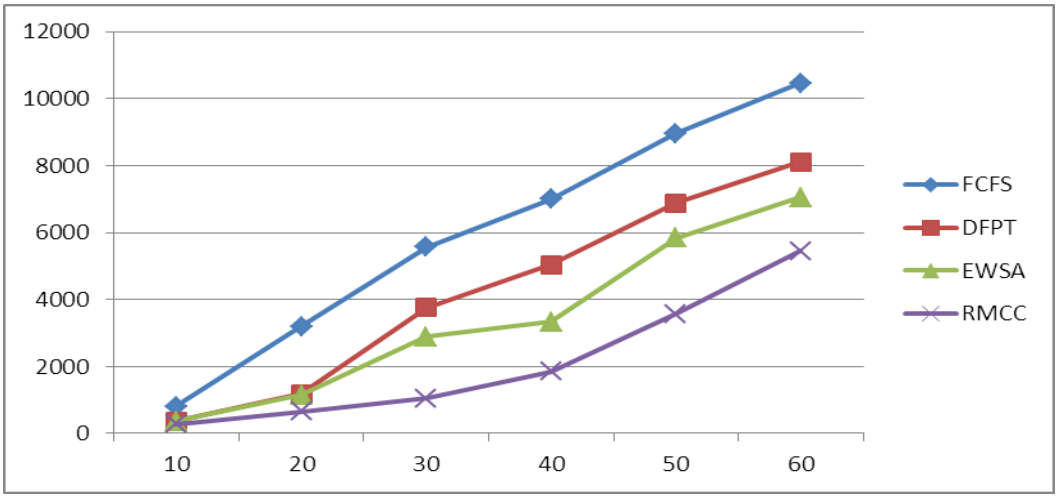

Figure 4: Graph showing comparision of execution time of proposed work RMCC with existing algorithms.

Performance with respect to cost: The experimental results shows the immense improvement in cost over sequential, DFPT and EWSA algorithms.

Table 5: Comparision of Cost of Execution of Proposed methodology V/s existing algorithms.

No. of task set	FCFS	DFPT	EWSA	PROPOSED METHODOLOGY
10	363.44	200.448	197.576	160.001
20	539.61	369.866	311.667	271.904
30	944.68	677.007	661.112	511.008
40	1248.56	721.481	624.081	441.169
50	1554.81	1098.224	976.116	771.816
60	1768.66	1544.616	1496.672	981.197

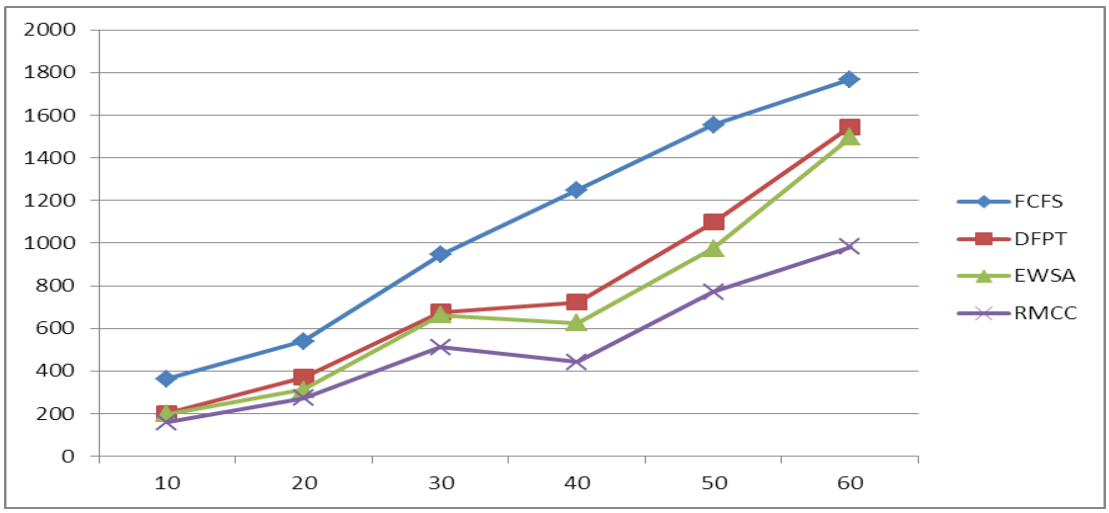

Figure 5: Graph showing comparision of cost of execution of proposed RMCC method with existing algorithms.

Table 4 and Table 5 are showing performance evaluation of various number of task set against execution time and execution cost. Here, a task set can have one task, two tasks or up to five tasks having interdependency. A task set is generated at run time and dependency is also decided at execution time randomly. So we have calculated, the results of various execution time of task set as the arithmetic mean of execution time obtained on running the task set twenty times. Each time result may somewhat vary depending upon dependency and number of task in task set.

CONCLUSION AND FUTURE WORK

IAAS cloud computing offers immense research work to do like data security, virtualization license management etc. But the scheduling is the topic of research in this field. Many well

defined algorithms are proposed by many researchers, but none of them is suitable in all the way. In contrast, our proposed work is more dynamic and pragmatic that meets all the demands of perfect scheduling in IAAS. We have proposed RCMMS and workload balancing concepts and resolved dependency among task sets while minimizing execution time and execution cost. We have provided complete scenario of real time IAAS cloud scheduling and implemented it using CloudSim. The outcome results are showing tremendous improvement as compared to FCFS, DFPT[3] and EWSA[6] cloudlet scheduling algorithms.

In future more sophisticated and real algorithms can be research and work out for IAAS cloud system including better techniques for workload balancing, energy-saving techniques and considering contention problem at datacenters.

REFRENCES

1. Andreas Berl, Erol Gelenbe, Marco Di Girolamo, Giovanni Giuliani, Hermann De Meer, Minh Quan Dang and Kostas Pentikou sis, "Energy-Efficient Cloud Computing", Advance Access publication on August 19, 2009.
2. Jiayin Li, Meikang Qiu, Zhong Ming, Gang Quan, Xiao Qin, Zonghua Gu, "Online optimization for scheduling preemptable tasks on IaaS cloud system", J.Parallel Distributed Computing journal, 2012; 72: 666-677.
3. Deepika Saxena, Dr. R.K. Chauhan, Dr. Ramesh Kait,"Dynamic Fair Priority Optimization Task Scheduling Algorithm in Cloud Computing: Concepts and Implementation", I.J. Computer Network and Information Security", 2016; 2: 41-48.
4. P.Mell and T.Grance, "The NIST definition of cloud computing", National Institute of Standard and Technology, 2009; 53.
5. Infrastructure as a Service, http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-service.
6. Shilpi Saxena, Deepika Saxena,"Enriched Workflow Scheduling Algorithm", IEEE International Conference on Computing, Communication and Security(ICCCS), 2015 10.1109/CCCS.2015.7374202
7. H.J.M.Y. Chi, H. Hacigumus, Performance evaluation of scheduling algorithmsfor database services with soft and hard SLAs, in: International Workshop onData Intensive Computing in the Clouds, 2011; 1–10.
8. J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters,Communications of the ACM, 2008; 1: 107–113.

9. A. Dogan, F. Ozguner, Matching and scheduling algorithms for minimizingexecution time and failure probability of applications in heterogeneous computing, IEEE Transactions on Parallel and Distributed Systems, 2002; 13(3): 308–323.
10. W. Emeneker, D. Stanzione, Efficient virtual machine caching in dynamic virtual clusters, in: SRMPDS Workshop of International Conference on Parallel and Distributed Systems, Hsinchu, Taiwan, 2007.
11. N. Fallenbeck, H. Picht, M. Smith, B. Freisleben, Xen and the art of cluster scheduling, in: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, Tampa, Florida, USA, 2006; 4.
12. T. Forell, D. Milojicic, V. Talwar, Cloud management challenges and opportunities, in: IEEE International Symposium on Parallel and Distributed, 2011; 881–889.
13. G. Garzoglio, T. Levshina, P. Mhashilkar, S. Timm, ReSS: a resource selection service for the open science grid, in: International Symposium on Grid Computing, 2007; 1–10.
14. C. Germain-Renaud, O. Rana, The convergence of clouds, grids, and autonomics, IEEE Internet Computing 2009; 13(6): 9.
15. Apache Hadoop, http://wiki.apache.org/hadoop/.
16. T. Hagras, J. Janecek, A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems, Parallel Computing, 2005; 31(7): 653–670.
17. H. Huang, L. Wang, P&p: a combined push–pull model for resource monitoring in cloud computing environment, in: IEEE International Conference on Cloud Computing, Miami, Florida, USA, 2010; 260–266.
18. O.H. Ibarra, C.E. Kim, Heuristic algorithms for scheduling independent tasks on nonidentical processors, Journal of the ACM., 1977; 24(2): 280–289.
19. D. Irwin, J. Chase, L. Grit, A. Yumerefendi, D. Becker, K.G. Yocum, Sharing networked resources with brokered leases, in: USENIX Annual Technical Conference, Boston, Massachusetts, USA, 2006.
20. W. Jiang, Y. Turner, J. Tourrilhes, M. Schlansker, Controlling traffic ensembles in open cirrus, https://www.hpl.hp.com/techreports/2011/HPL-2011-129.pdf.
21. K. Keahey, T. Freeman, Contextualization: providing one-click virtual clusters, in: IEEE International Conference on eScience, Indianapolis, Indiana, USA, 2008.
22. M.A. Kozuch, M.P. Ryan, R. Gass, S.W. Schlosser, D. O'Hallaron, et al. Cloud management challenges and opportunities, in: Workshop on Automated control for datacenters and cloud, 2009; 43–48.

23. H. Liu, D. Orban, GridBatch: cloud computing for large-scale data-intensive batch applications, in: IEEE International Symposium on Cluster Computing and the Grid, 2008; 295–305.
24. Open computing facility - ocf, https://computing.llnl.gov/?set=resources&page=OCF-resource.
25. R. Moren-Vozmediano, R.S. Montero, I.M. Llorente, Elastic management of cluster-based services in the cloud, in: Workshop on Automated control for datacenters and cloud, 2009; 19–24.
26. C. Moretti, J. Bulosan, P.J. Thain, D. Flynn, All-pairs: an abstraction for dataintensive cloud computing, in: IEEE International Symposium on Parallel and Distributed Processing, 2008; 1–11.
27. A. Pavlo, E. Paulson, A. Rasin, D.J. Ababi, D. DeWitt, S. Madden, M. Stonebraker, A comparison of approaches to large-scale data analysis, in: SIGMOD, 2009; 1–14.
28. Parallel workloads archive, www.cs.huji.ac.il/labs/parallel/workload/.
29. M. Qiu, M. Guo, M. Liu, C.J. Xue, L.T. Yang, E.H.-M. Sha, Loop scheduling and bank type assignment for heterogeneous multi-bank memory, Journal of Parallel and Distributed omputing (JPDC) 2009; 69(6): 546–558.
30. M. Qiu, E.H.-M. Sha, Cost minimization while satisfying hard/soft timing constraints for heterogeneous embedded systems, ACM Transactions on Design Automation of Electronic Systems (TODAES) 2009; 14(2): 1–30.
31. B.P. Rimal, E. Choi, I. Lumb, A taxonomy and survey of cloud computing systems, in: International Joint Conference on INC, IMS and IDC, 2009; 44–51.
32. P. Ruth, P. McGachey, D. Xu, Viocluster: virtualization for dynamic computational domains, in: Proceedings of the IEEE International Conference on Cluster Computing, Boston, Massachusetts, USA, 2005; 1–10.
33. P. Ruth, J. Rhee, D. Xu, R. Kennell, S. Goasguen, Autonomic live adaptation of virtual computational environments in a multi-domain infrastructure, in: IEEE International Conference on Autonomic Computing, 2006; 5–14.
34. W. Smith, I. Foster, V. Taylor, Scheduling with advanced reservations, in: IEEE International Parallel and Distributed Processing Symposium, Cancun, Mexico, 2000; 127–132.
35. B. Sotomayor, K. Keahey, I. Foster, Combining batch execution and leasing using virtual machines, in: Proceedings of the 17th International Symposium on High Performance Distributed Computing, Boston, Massachussets, USA, 2008; 87–96.

36. B. Sotomayor, R. Llorente, I. Foster, Resource leasing and the art of suspending virtual machines, in: 11th IEEE International Conference on High Performance Computing and Communications, Seoul, Korea, 2009; 59–68.
37. B. Sotomayor, R. Montero, I. Llorente, I. Foster, Virtual infrastructure management in private and hybrid clouds, IEEE Internet Computing 2009; 13(5): 14–22.
38. Amazon AWS, http://aws.amazon.com/.
39. GoGrid, http://www.gogrid.com/.
40. RackSpace, http://www.rackspacecloud.com/.
41. Microsoft cloud, http://www.microsoft.com/en-us/cloud/.
42. IBM cloud, http://www.ibm.com/ibm/cloud/.
43. Google apps, http://www.google.com/apps/intl/en/business/index.html.
44. HP cloud, http://www8.hp.com/us/en/solutions/solutionsdetail.html? compURI=tcm:245-300983\&pageTitle=cloud.
45. Eucalyptus, http://www.eucalyptus.com/.
46. RESERVOIR, www.reservoir-fp7.eu.
47. E. Walker, J. Gardner, V. Litvin, E. Turner, Dynamic virtual clusters in a grid site manager, in: Proceedings of Challenges of Large Applications in Distributed Environments, Paris, France, 2006; 95–103.
48. X. Wang, J. Zhang, H. Liao, L. Zha, Dynamic split model of resource utilization in mapreduce, in: International Workshop on Data Intensive Computing in the Clouds, 2011; 1–10.
49. M. Wilde, M. Hategan, J.M. Wozniak, B. Clifford, D.S. Katz, I. Foster, Swift: a language for distributed parallel scripting, Parallel Computing 2011; 37(9): 633–652.
50. T. Wood, A. Gerber, K. Ramakrishnan, J. van der Merwe, The case for enterpriseready virtual private clouds, in: Workshop on Hot Topics in Cloud Computing, San Diego, Califoria, USA, 2009.
51. S. Zanikolas, R. Sakellariou, A taxonomy of grid monitoring systems, Future Generation Computer systems, 2005; 1: 163–188.