Algorithm for Scheduling of Dependent Task in Cloud

Miss.Vijetha[1], Miss.Gangothri R Sanil[2]

[1,2] Asst. Professor, Dept. of Computer Science Engineering, Srinivas Institution of Engineering, Karnataka, India.

---***---

Abstract - *Cloud computing is a group of computing hardware machine connected through a communication network that is internet. In the simplest terms, it means storing and accessing data and programs over the internet instead of computer's hard drive. Cloud computing offer their services according to service model that is Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). IaaS is a service where cloud providers supply resources to the users on demand with pay as you go pricing. Virtual machine is the software based machine that runs on physical machine. Scheduling of the workflow is an important issue to be considered. An algorithm is proposed using cloud simulator for improving the efficiency of execution of task in cloud. Different workflows are experimented with varying virtual machines configuration and task with different cloudlet length. The proposed algorithm is compared with FCFS (First Come First Serve) for different workflows with different configurations of virtual machines and cloudlets. The observation concludes that the proposed algorithm is efficient than FCFS and reduces the overall makespan.*

1.INTRODUCTION

Cloud computing is a recent advancement where in Information Technology infrastructure and applications are provided as "services" to end-users under a usage-based payment model. Cloud computing can be 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". Cloud computing is a means of delivering any and all Information Technology (IT) from computing power to computing infrastructure, applications, business processes and personal collaboration to end-users as a service wherever and whenever they need it. Cloud computing is an emerging style of IT delivery in which applications, data and IT resources are rapidly provisioned and provided as standardized offerings to users over the web in a flexible pricing model or it is way of managing large numbers of highly virtualized resources such that, from a management perspective, they resemble a single large resource. It is a model for enabling convenient and on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing can be viewed from two different perspectives: cloud application and cloud infrastructure as the building block for the cloud application. Most organizations focus on adopting cloud computing model so that they can cut capital expenditure, efforts and control operating costs. Some of the traditional cloud-based application services include social networking, web hosting, content delivery and real time instrumented data processing which has different composition, configuration and deployment requirements. A more viable alternative is the use of cloud simulation tools.

Cloud computing features help to bring agility, transparency and utilization at the datacenter which are self-service that enables an interface for separate authenticated end-users via Role-Based-Access-Controls (RBAC) to select options for deployment. It should have unique policy controls per tenant and user role and also the ability to present unique catalogues per user or group. It must also possess the dynamic workload management which enables the datacenters for automation and it must also have the orchestration software that coordinates workflow requests from the service catalogue or self-service portal for provisioning Virtual Machine (VM). Each provisioned VM is enabled with a life-cycle for deployment expiration thereby increasing the efficiency of utilization of resources. Resource automation feature establishes secure multi-tenancy, isolates virtual resources and helps prevent contention in the load aware resource engine which intelligently does the workload packing or load balancing across hypervisors automatically. Chargeback, showback and metering features facilitates the administrator to bring out the usage reports for cloud infrastructure service consumption which serve as a basis for metering and billing system. Enabling chargeback, showback and metering in any organization would bring in transparency to the business and environment for management to see the usage and to take decision-making steps. Examples for emerging cloud computing platforms are Microsoft Azure, Amazon EC2, Google App Engine and Aneka.

One of the implications of cloud platforms is the ability to dynamically adapt the amount of resources provisioned to an application in order to attend variations in demand that are either predictable or unexpected and that occur due to a subtle increase in the popularity of the application service. This capability of clouds is especially useful for elastic applications, such as web hosting, content delivery and social networks that are susceptible to such behavior.

1.1 Cloud Service Models

Cloud computing providers offer their 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.

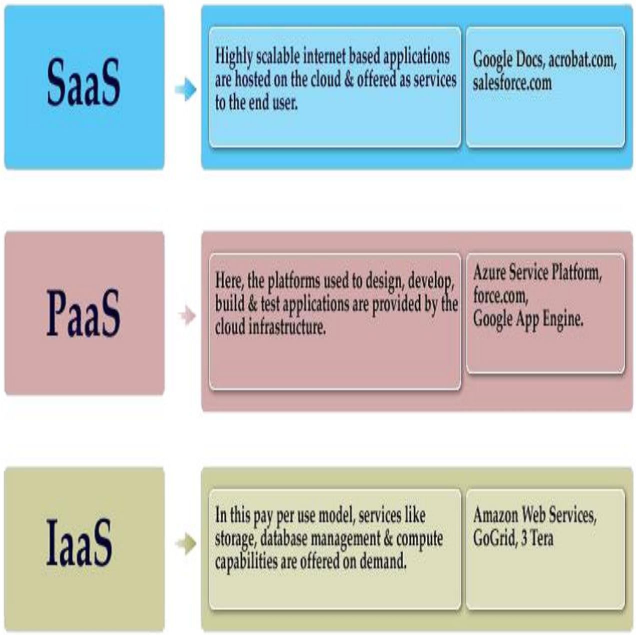

Figure: 1.1: Types of Service Models in cloud computing

1.1.1 Software as a Service (SaaS)

Software as a service (SaaS) saves costs by removing the effort of development, maintenance and delivery of software. SaaS eliminates up-front software licensing and infrastructure costs and it reduces ongoing operational costs for support, maintenance and administration. The time to build and deploy a new service is much shorter than for traditional software development. By transferring the management and software support to a vendor, internal IT staff can focus more on higher-value activities. Software as a Service sometimes referred to as "on-demand software" or "Application-Service-Providers" is a software delivery model in which software and associated data are centrally hosted on the cloud. SaaS providers generally price applications using a subscription fee, most commonly a monthly fee or an annual fee. The SaaS allows the business with an opportunity to reduce the IT operational costs. Example for SaaS is Salesforce.com, Google Docs, acrobat.com.

1.1.2 Platform as a Service (PaaS)

Platform as a Service (PaaS) provide the foundation to build highly scalable and robust Web-based applications in the same way that the traditional operating systems like Windows and Linux have done in the past for software developers. Different about this model is that no longer the platform itself 'sold' to the customer who is then responsible for running and maintaining it. In this model, it is this very operational capability of the platform hosting that it is of primary value which has far-reaching implications to both the business models PaaS vendors as well as their customers. Cloud capabilities can improve the productivity of the developments made by the customer. It provides a catalogue of virtual images, and patterns all ready for immediate use. PaaS significantly improves development productivity by removing the challenges of integration with services such as database, middleware, web frameworks, security, and virtualization.

Software development and delivery times are shortened since software development testing are performed on a single PaaS platform. There is no need to maintain separate development and test environments. PaaS fosters collaboration among developers and also simplifies software project management. This is especially beneficial to enterprises that have outsourced their software development. PaaS provides a software development environment that enables rapid deployment of new applications. Users can access this category of cloud through a Web browser. PaaS provides a platform where the users can develop applications and use services over the internet. The services are provided on the basis of various subscription schemes. In this model, the consumer creates the software using tools and/or libraries from the provider. The consumer also controls software deployment and configuration settings. The provider provides the networks, servers, storage and other services. PaaS offerings facilitate the deployment of applications without the cost and complexity of buying and managing the underlying hardware and software and provisioning hosting capabilities. Services offer varying levels of scalability and maintenance. PaaS offerings may also include facilities for application design, application development, testing and deployment as well as services such as team collaboration, Web service integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation and developer community facilitation. Example for PaaS is Microsoft Azure.

1.1.3 Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) is a provision model in which an organization outsources the equipment used to support operations, including servers, storage, hardware, and networking components. The client typically pays on a per-use basis. The service provider owns the equipment and is responsible for housing, running and maintaining it. Infrastructure as a Service (IaaS) refers to the renting of computer hardware (servers, networking technology, storage, and datacenter space) instead of buying and installing it in own datacenter. Operating systems and virtualization technology to manage these resources can also be rented for use in a company's cloud computing environment. More and more companies are looking to defray costs and gain flexibility by leveraging infrastructure that can be used on demand. IaaS saves costs by eliminating the need to over-provision computing resources to be able to handle peaks in demand. Resources dynamically scale up and down as required, reducing capital expenditure on infrastructure and ongoing operational costs for support, maintenance and administration. Organizations can massively increase their datacenter resources without significantly increasing the number of people needed to support it. The time required to provision new infrastructure resources is reduced from typically months to just minutes- the time required to add the requirements to an online shopping cart, submit it, and have it approved. IaaS platforms are generally open platforms, supporting a wide range of operating systems and frameworks which minimize the risk of vendor lock-in. Infrastructure resources are leased on a pay-as-you-go basis, according to the hours of usage. Example for IaaS is Amazon Ec2, Amazon s3.

1.2 Cloud Deployment Models

Deploying cloud computing can differ depending on requirements, and the following four deployment models have been identified, each with specific characteristics that support the needs of the services and users of the clouds in particular ways.

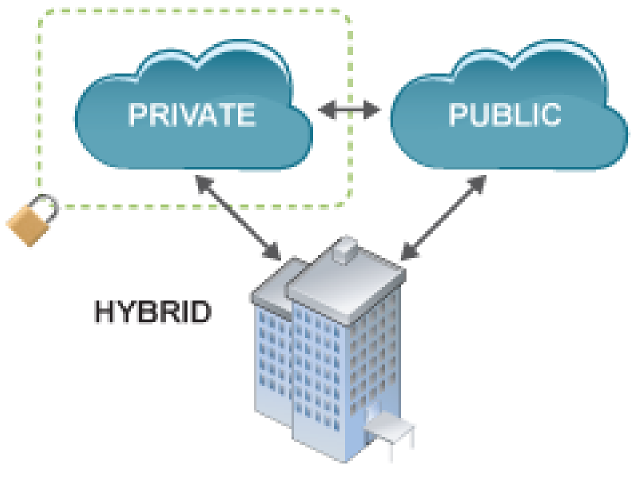

Figure 1.2 Cloud Deployment Models

1.2.1 Private Cloud

The cloud infrastructure has been deployed and is maintained and operated for a specific organization. The operation may be in-house or with a third party on the premises. Public cloud applications, storage, and other resources are made available to the general public by a service provider. These services are free or offered on a pay-per-use model.

1.2.2 Public Cloud

The cloud infrastructure is available to the public on a commercial basis by a cloud service provider. This enables a consumer to develop and deploy a service in the cloud with very little financial outlay compared to the capital expenditure requirements normally associated with other deployment options. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and offer their services which can be only accessed via Internet.

1.2.3 Hybrid Cloud

The cloud infrastructure consists of a number of clouds of any type, but the clouds have the ability through their interfaces to allow data and applications to be moved from one cloud to another. Hybrid cloud can be a combination of private and public clouds that support the requirement to retain some data in an organization and also the need to offer services in the cloud.

2. LITERATURE SURVEY

There has been some work in the field of task scheduling in cloud computing. A system that enables an organization to augment its computing infrastructure by allocating resources from a cloud provider is discussed in paper [1]. Author provides various scheduling strategies that aim to minimize the cost of utilizing resources from the cloud provider. Scheduling strategies are conservative, aggressive and selective backfilling. The author evaluates the proposed strategies, considering different performance metrics, namely average weighted response time, job slowdown, number of deadline violations, number of jobs rejected, and the money spent for using the cloud and evaluates the trade-off between performance improvement and cost. Conservative strategy is where each job is scheduled when it arrives in the system and these requests are allowed to jump ahead in the queue, if they do not delay the execution of the request. In Aggressive strategy the head of the queue is called pivot and it is granted a reservation, other requests are allowed to move ahead if they do not delay the pivot. In selective backfilling, it grants reservation to requests that have waited long enough in the queue. This paper contributes an ideology of using VMs under lease and as a container for deploying applications and also investigates whether an organization operating on its local cluster can benefit from using cloud providers to improve the performance of its users' requests. Seven scheduling strategies suitable for a local cluster that is managed by VM based technology to improve its Service Level Agreements (SLAs) with users are evaluated in this paper. These strategies aim to utilize remote resources from the cloud to augment the capacity of the local cluster. However, as the use of cloud resources incurs a cost, the problem is to find the price at which this performance improvement is achieved.

Paper [2] presents a self-adaptive, deadline-aware resource control framework that is implemented in a fully distributed fashion. Nash Bargaining Solutions (NBS) with respect to job priority and deadline is maximized based on Nash Bargaining. The author uses the concept of using military cloud which is a set of computing resources that are located in different geo-Graphical area. The author considers three types of deadline. They are Hard deadline, Soft deadline and No deadline. The policy used in this paper results in better results compared to random scheduling, fair scheduling and Earliest-Deadline-First scheduling. In this model, Hadoop engine is used to model the MapReduce jobs. The model shows significant utility improvement and deadline satisfaction over existing approaches.

Total Cost of Ownership (TCO) model approach for cloud computing Services is presented in paper [3].The author has applied a multi-method approach for the development and evaluation of the formal mathematical model and also verified that the model, which is presented, fits the practical requirements and supports decision making in cloud computing. The TCO model is based on the analysis of real cloud computing Services of author's cloud computing research data base. Furthermore author has conducted a systematic literature review with which he identified important cost types and factors. The TCO model is prototypically implemented on a website for further evaluation steps and is accessible for the general public. A software tool is able to analyze the cost structure of cloud computing Services and thus supports decision makers in validating cloud computing Services from a cost perspective.

In paper [4], the recent emergence of public cloud offerings, outsourcing tasks from an internal data center to a cloud provider in times of heavy load is presented. It has become more accessible to a wide range of consumers. In this paper, the objective is to maximize the utilization of the internal data center and to minimize the cost of running the tasks in the cloud, by fulfilling the quality of service constraints. The author examines optimization problem in a multi-provider hybrid cloud setting with deadline-constrained and preemptible but non provider migratable workloads that are characterized by memory, CPU and data transmission requirements. Linear programming is a general solution for such an optimization problem. Therefore the author analyzes and proposes a binary integer program formulation of the scheduling problem and evaluates the computational costs of this technique with the problem's key parameters. The approach results in a solution for scheduling applications in the public cloud, but the same method becomes much less feasible in a hybrid cloud setting due to very high solve time variances.

Cloud computing scheduler to determine on which processing resource jobs of a workflow should be allocated is discussed in paper [5]. Author says in hybrid clouds, jobs can be allocated either on a private cloud or on a public cloud on a pay per use basis. The capacity of the communication channels connecting these two types of resources impact the makespan and the cost of workflows execution. This paper introduces the scheduling problem in hybrid clouds presenting the main characteristics to be considered when scheduling workflows, as well as a brief survey of some of the scheduling algorithms used in these systems. To assess the influence of communication channels on job allocation, author compares and evaluates the impact of the available bandwidth on the performance of some of the scheduling algorithms. Therefore communication capacity is given importance when scheduling workflows. The scheduling algorithm is developed for efficient use on utility grids and is not efficient on hybrid clouds.

The author in paper [6] describes about Aneka software which is a platform used for building and managing a system. Aneka enables the execution of scientific application in hybrid cloud by efficiently allocating resources from different sources. In this paper author gives a brief overview of Aneka framework which consists of Aneka container containing three services namely execution service which are responsible for scheduling and executing application, foundation services which manages the nodes, metering, allocating resources and service updating and third service that is fabric service whose service is to mainly provisioning the resources. The author also describes about Aneka programming models they are bag of tasks which is a group of tasks which are independent and can be executed in any order, distributed thread which executes threads with shared memory and locks, map reduce model which is map and reduce functions. Hence a deadline mechanism to efficiently allocate resources to reduce execution time is discussed in this paper.

Cost and time optimization by using grid resources is discussed in paper [7]. The author mainly tells about Meta schedulers which work on behalf of users and assigning jobs to resources. In this paper, the author gives brief introduction of existing scheduler which minimizes either cost or time but not both. Meta broker system consists of three main participants they are service provider where CPU time slots are given as a services, users and Meta broker which is used to match jobs to appropriate services. Users are required to pay for their usage based on quality of service such as deadline and makespan.

The author in paper [8] says that Metaschedulers can distribute parts of a Bag-of-Tasks (BoT) application among various resource providers in order to speed up its execution. The expected completion time of the user application is then calculated based on the run time estimates of all applications running and waiting for resources. However, due to inaccurate run time estimates, initial schedules are not those that provide users with the earliest completion time. These estimates increase the time distance between the first and last tasks of a BoT application, which increases average user response time, especially in multi-provider environments. This paper proposes a coordinated rescheduling algorithm to handle inaccurate run time estimates when executing BoT applications in multi-provider environments. The coordinated rescheduling defines which tasks can have start time updated based on the expected completion time of the entire BoT application. The evaluation is done for the impact of system-generated run time estimates to schedule Bag of Task applications on multiple providers. The experiments were performed by simulations and a real distributed platform, Grid'5000. From experiments, the obtained reductions of up to 5% and 10% for response time and slowdown metrics respectively by using coordinated rescheduling over a traditional rescheduling solution is observed. Moreover, coordinated rescheduling requires little modification of existing scheduling systems. System-generated predictions, on the other hand, are more complex to be deployed and may not reduce response times as much as coordinated rescheduling. Metaschedulers can distribute parts of a BoT application among various resource providers in order to speed up its execution. The expected completion time of the user application is then calculated based on the run time estimates of all applications running and waiting for resources.

In paper [9], the author presents Min-min task scheduling. The min-min task scheduling algorithm is a batch mode algorithm. It is mainly used to optimize the execution time of independent tasks with some number of resources. It first completes early completion time of tasks based on resource available to the task. In this algorithm a task which has minimum MCT (Minimum estimate Completion Time) value compared to all task is chosen to be first scheduled. It basically assigns the task on the resource which will finish execution at faster rate. It is expected that smaller makespan is obtained if more tasks are assigned to the machine that complete them the earliest and also execute the tasks faster.

Max-min task scheduling is presented in paper [10]. The max-min task scheduling algorithm is a batch mode algorithm. It is mainly used to optimize the execution time of independent tasks with some number of resources. In this algorithm the task that takes longer execution time are given first priority. Firstly MCT (Minimum Estimate Completion Time) value for each task is computed, then the task having highest or maximum MCT value is selected and scheduled and finally the task is assigned to the resource. This task with maximum MCT is executed at earliest time. Hence the author concludes that this algorithm is expected to get a balanced load across machine and to obtain better makespan.

In this paper [12], the author presents a graph, where each node edge has computation and communication cost. The processing time and data transmission are computed based on the performance of each resource. Relative cost scheduling has three different approaches, they are: Relative Cost of Task and Edge(RCTE), here tasks and edge are separately sorted in two sets, therefore the scheduler receive a set of tasks and edges as inputs are sorted by their weights. The other technique used is Relative Costs with the Computation to Communication Ratio (RC-CCR), where the weight of each task is multiplied to relative weight. Relative Costs Altogether (RCA) is the third method in which tasks and edges are sorted in single set and weights as RCTE. One of the other method discussed in this paper is Relative Heterogeneous Earliest Finish Time (HEFT), where relative cost is input to the scheduling algorithm. It is used to obtain minimum makespan. Rank for each task is calculated. The author says cost can be given as a relation component than numeric values. Hence the three different ways of specifying the relative costs can improve the schedule.

In this paper [13], the author briefs about techniques of list and graph scheduling. Its main challenge is to achieve minimum job accomplishing time and high resource utilization efficiency with fault tolerance. DAGMap (Directed Acyclic Graph) has two phases, they are static mapping and dependable execution. There are four salient features in this paper which are task grouping is based on dependency relationship and task upward priority, Critical tasks are scheduled first, Max-Min and Min-Min selective scheduling are used in independent tasks, Check point servers are used for dependable execution. Static mapping has two steps they are task grouping which includes determining priorities of upward, downward and over all, Independent tasks scheduling which schedules group by group in ascending order. In task grouping, firstly the upward, downward and over all priority of each task are determined. Accordingly these collections of critical tasks are obtained and then tasks are grouped by upward priority and dependency relationship. Tasks in the same group are kept independent. Procedure for task grouping includes two algorithm, they are DAGMap heuristic which has following steps. Firstly all tasks are sorted in descending order and given upward priority. Then entry task are added to group according to each level.

Later successive tasks are independent of already existing group are added to particular group else form another group. This is repeated until all tasks are grouped. Another algorithm is independent task scheduling, which includes Max-Min and Min-Min algorithm. Max-Min executes all short tasks first and then larger tasks, Min-Max execute tasks in vice versa manner. Therefore better makespan is obtained and better resource is utilized. Dependable execution reduces fault tolerance.

In this paper [14] a new scheduling algorithm to schedule task by considering several parameters like machine capacity, priority of task and history log is presented. In CloudSim, there are two default allocation policies, they are Round Robin algorithm in a time shared system and First Come First Serve (FCFS) algorithm in space shared system. This is paper is based on the concept of multi-queue. The algorithm used is queue based task scheduling algorithm which minimizes completion time of job being scheduled. In this paper it takes a global queue which is used for store all the incoming tasks. There are three local queues for three different VMs. This paper proposes a scheduling algorithm to detect the status of VM. In algorithm, it first computes Completion Time (CT) of a job in three VMs, if maximum completion time is shorter than average arriving time and average processing time. First of all completion time of each VM must be compared and select the one which has minimum completion time.

In this paper [15] the CloudSim simulation layer provides support for modeling and simulation of virtualized cloud-based data center environments including dedicated management interfaces for VMs, memory, storage, and bandwidth are presented. The fundamental issues, such as provisioning of hosts to VMs, managing application execution, and monitoring dynamic system state, are handled by this layer. A cloud provider, who wants to study the efficiency of different policies in allocating its hosts to VMs (VM provisioning), would need to Implement his strategies at this layer. Such implementation can be done by programmatically extending the core VM provisioning functionality.

The top-most layer in the CloudSim stack is the user code that exposes basic entities for hosts (number of machines, their specification, and so on), applications (number of tasks and their requirements), VMs, number of users and their application types, and broker scheduling policies. By extending the basic entities given at this layer, a cloud application developer can perform the following activities: (i) generate a mix of workload request distributions, application configurations; (ii) model cloud availability scenarios and perform robust tests based on the custom configurations; and (iii) implement custom application provisioning techniques for clouds and their federation.

The infrastructure-level services (IaaS) related to the clouds can be simulated by extending the data center entity of CloudSim. The data center entity manages a number of host entities. The hosts are assigned to one or more VMs based on a VM allocation policy that should be defined by the cloud service provider. Here, the VM policy stands for the operations control policies related to VM life cycle such as: provisioning of a host to a VM, VM creation, VM destruction, and VM migration. Similarly, one or more application services can be provisioned within a single VM instance, referred to as application provisioning in the context of cloud computing. In the context of CloudSim, an entity is an instance of a component. A CloudSim component can be a class (abstract or complete) or set of classes that represent one CloudSim model (data center, host). A data center can manage several hosts that in turn manage VMs during their life cycles. Host is a CloudSim component that represents a physical computing server in a cloud: it is assigned a pre-configured processing capability expressed in Millions of Instructions Per Second (MIPS), memory, storage, and a provisioning policy for allocating processing cores to VMs. The Host component implements interfaces that support modeling and simulation of both single-core and multi-core nodes.

3. CONCLUSIONS

This thesis presents an algorithm for executing workflow in a cloud. The workflow consists of set of task where tasks are dependent on each other. In order to evaluate the algorithm using cloud simulator, experiments are conducted by varying the different set of workflow with different VM configurations and cloudlet length. The result is compared with the FCFS. It is observed that proposed algorithm results in minimum makespan compared to FCFS.

REFERENCES

[1] Marcos Dias de Assuncao, Alexandre di Costanzo and Rajkumar Buyya, "A Cost-benefit Analysis of using Cloud Computing to Extend the Capacity of Clusters", In Proceedings of IEEE International Symposium on Cluster Computing, September pp.335-347, 2010.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] Yu Xiang, Bharath Balasubramanian, Michael Wang, TianLan , Soumya Sen and Mung Chiang, "Self-adaptive, Deadline-aware Resource Control in Cloud Computing", Carlson School of Management, University of Minnesota, MN 55455.

[3] Benedikt Martens, Marc Walterbusch and Frank Teuteberg, "Costing of Cloud Computing Services: A Total Cost of Ownership Approach", In Proceedings of

Hawaii International Conference on System Science (HICSS), 2012, pp.1563-1572.

[4] Ruben Van den Bossche, Kurt Vanmechelen and Jan Broeckhove, "Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads", Department of Mathematics and Computer Sciences University Antwerp, Antwerpen Belgium.

[5] Luiz F.Bittencourt, Edmundo R.M.Madeira, Member IEEE, and Nelson L. S. da Fonseca, Senior Member, IEEE, "Scheduling in hybrid Clouds", Institute of Computing – University of Campinas – UNICAMP Av. Albert Einstein, 1251, 13083-852.

[6] Christian Vecchiola, Rodrigo N.Calheiros, Dileban Karunamoorthy and Rajkumar Buyya, "Deadline-driven Provisioning of Resources for Scientific Applications in Hybrid Clouds with Aneka", Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia, Manjrasoft Private Limited, Melbourne, Australia.

[7] Saurabh Kumar Garg, Rajkumar Buyya and H. J. Siegel, "Scheduling Parallel Applications on Utility Grids: Time and Cost Trade-off Management", Grid Computing and Distributed Systems Laboratory Department of Computer Science and Software Engineering, The University of Melbourne, Victoria 3010, Australia, Electrical and Computer Engineering Department and Computer Science Department Colorado State University Fort Collins, USA.

[8] Marco A.S.Netto and Rajkumar Buyya, "Coordinated Rescheduling of Bag-of-Tasks for Executions on Multiple Resource Providers", Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering the University of Melbourne, Parkville, VIC 3010, Australia.

[9] T.D.Braun, H.J.Seigel, and N.Beck, "A Comparison of Eleven static Heuristics for Mapping a Class of Independent Tasks on to Heterogeneous Distributed Computing System", Journal of Parallel and Distributed Computing, 2001.

[10] Upendra Bhoi and Purvi N.Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", Department of Computer Science & Technology, L.D.College of Engineering, Gujarat Technological University, Ahmedabad.

[11] M.Maheshwaran, Shoukat Ali, Howard Jay Siegel, Debra Hensgen, and Richard F. Freund, "Dynamic Matching and Scheduling of a Class of Independent Tasks on to Heterogeneous Computing Systems", The 8th Heterogeneous Computing Workshop, San Juan, Puerto Rico, Apr. 12 1999.

[12] Luiz F. Bittencourt, Rizos Sakellariou, Edmundo R. M. Madeira, "Using Relative Cost with Workflow Scheduling to cope with Input Data Uncertainity", Montreal, Quebec, Canada, December 3, 2012.

[13] Haijun Cao, Hai Jin, Xiaoxin Wu, Song Wu and Xuanhua Shi, "DAGMap: Efficient and dependable scheduling of DAG workflow job in Grid", Spring Science, Business Media,LLC, May 6, 2009.

[14] Wanqing You, Kai Qian and Ying Qian, "Hierarchial Queue-Based Task Scheduling", Journal of Advances in Computer Networks, Vol. 2, No. 2, June.