2 Literature Review

Monitoring the cloud performance is an important issue for both the cloud users and the cloud provider. Monitoring facilitates analysis of both the real time and historical data for performance analysis of the cloud. Existing cloud monitoring tools are either provided by the cloud provider itself (such as cloudwatch by Amazon) or user can also use these tools provided by the third parties, e.g. the tools like revealuptime and cloudstatus.

Factors that impact the performance significantly has been demonstrated by Vineetha (2012), who has suggested the importance of cloud performance monitoring tool for both cloud provider and cloud user perspective. Author has monitored the response time, CPU usage and storage usage etc. Reporting for these factors in peak and off peak hours also suggested. Avram (2010) measured the performance of 5 dominant cloud providers (Google, Rackspace, Salesforce.com, Amazon and terremark) for the website and measured the response time and latency time of each of these cloud providers. In other work (Harzog (2010) has suggested a new approach for infrastructure management to determine the performance in virtualized environment by monitoring infrastructure response time. (Bannerman 2010) suggested the performance factors i.e. latency, response time and execution time as one of the greatest risk in cloud computing. Virtual instrument (2012) emphasized to monitor the cloud performance by monitoring for its factor like response time and latency by use of automated tool to enable the cloud user to know about the exact status of the cloud.

Hence, from the above discussion and table it can be inferred that response time, data processing time, throughput and uptime are very important factors which plays vital role in cloud and hence, required to be monitored to determine the cloud performance.

To improve the performance in cloud computing some of the useful work has already been carried out. These works can be broadly categorized as follows:

2.1 Scheduling

An analytical model for interaction aspects between on-demand request and allocation of virtual machine from the infrastructure owner to the user has been proposed by Dot and Rotter(2012). Comparison and Numerical analysis of several scheduling algorithm has been performed
considering the blocking probability of, on demand request. Paper concluded with the possibilities of energy saving in the operational range (where on-demand requests do not face unpleasant blocking probability) with the allocation of virtual machines to physical servers based on the priority of different allocation strategies.

To improve the performance in IaaS based multiple cloud services an algorithm for resource optimization with pre-emptable task execution to increase the utilization of the cloud was proposed by Li et al. (2012). Algorithm was capable to adjust the resource allocation dynamically, based on the updated information of the actual task executions. Efficient and reliable access limitations of traditional CPU-oriented batch schedulers were discussed by (Kosar and Balman, 2009). To improve the performance authors proposed Stork data placement scheduler in data intensive computing.

Lucas-Simarro et al. (2012) considered to improve the performance in multi cloud having fragmented interfaces, pricing schemes and virtual machine offerings. To minimize the cost, work has proposed a broker based on optimization algorithms to interface with multi cloud. Broker's algorithm was capable to decide how to select the cloud to deploy a service and how to distribute the different components of a service among different clouds, or even when to move a given service component from one cloud to another to satisfy some optimization criteria. Paper has considered different scheduling strategies for optimal deployment of virtual services across multiple clouds, based on different optimization criteria (e.g. cost optimization or performance optimization), different user constraints (e.g. budget, performance, instance types, placement, reallocation or load balancing constraints), and different environmental conditions (i.e., static vs. dynamic conditions, regarding instance prices, instance types, service workload, etc.). In web 2.0, applications composed of several components that run indefinitely and need to be available to end users throughout their execution life cycle. To avoid failure Frincu (2012) proposed a solution to find the optimal number of component types needed on nodes so that every type is present on every allocated node. Furthermore nodes cannot exceed a maximum threshold and the total running cost of the applications needs to be minimized. Authors have also given a sub-optimal solution. Both solutions rely on genetic algorithms to achieve their goals. The efficiency
of the suboptimal algorithm is studied with respect to its success rate, i.e., probability of the schedule to provide highly available applications in case all but one node fails.

2.2 Performance
To minimize the end-to-end delay for single-input applications, maximizing frame rate for streaming applications and to arbitrary node reuse Wu & Gu (2011) proposed dynamic programming-based optimal solution and proved the NP-completeness of the problems, for each of which, a heuristic algorithm based on a similar optimization procedure is proposed.

To provide affordability and to improve the performance in 'cloud storage', Broberg, Buyya and Tari, (2009) proposed meta CDN a system that exploits ‘Storage Cloud’ resources, creating an integrated overlay network that provides a low cost, high performance CDN for content creators. MetaCDN removes the complexity of dealing with multiple storage providers by intelligently matching and placing users’ content on to one or many storage providers based on their quality of service, coverage and budget preferences. Word considered the access efficiency of cloud storage during accessibility from anywhere and the capability demanded by the users. Work proposed an efficient cloud storage model for heterogeneous cloud infrastructures to improve the efficiency in cloud.

To evaluate the cloud performance considering the diversity of user requests and nature of cloud data centers. (Khazaei et al.2012) proposed a novel approximate analytical model for cloud server farms. The model allows cloud operators to determine the relationship between the number of servers and input buffer size, on one side, and the performance indicators such as mean number of tasks in the system, blocking probability, and probability that a task will obtain immediate service, on the other.

Considering the heterogeneity and proliferation of cloud eco system Tordsson et al. (2012) proposed brokering approach that optimizes placement of virtual infrastructures across multiple clouds and also abstracts the deployment and management of infrastructure components in these clouds.

To scale the application in intercloud, to improve the performance in case of resource limitations in cloud computing and to provide the quality of service to the user as per the user service level
agreement Calheiros et al. (2012) proposed an architecture for Cloud Coordinator. The proposed co-coordinator enables scaling of application across multiple, independent Cloud data centers, following market-based trading and an extensible design that allows its adoption in different public and private Clouds.

Mauch, Kunze & Hillenbrand (2012) provided an overview of the current state of high performance cloud computing technology, customer’s requirements. Work advocated use of virtual cluster from different cloud providers to provide the effective utilization of the resource and to avoid either underutilization or overloading.

Considering the performance dependency of IaaS clouds on a large set of parameters, e.g., workload, system characteristics and management policies Ghosha, Longo & Naik et al. (2012) proposed a scalable stochastic analytic model for performance quantification of Infrastructure-as-a-Service (IaaS) Cloud. Specifically, a class of IaaS Clouds that offer tiered services by configuring physical machines into three pools with different provisioning delay and power consumption characteristics. To minimize the interactivity work further suggested a multi-level interacting stochastic sub-models approach where the overall model solution is obtained iteratively over individual sub-model solutions and the proposed method compared with a single-level monolithic model to determine its efficiency.

Exposito et al. (2013) considered the dependency of HPC on scalability for efficient support of network communications in virtualized environments. The scalability of High Performance Computing (HPC) applications in IaaS cloud depends heavily on the efficient support of network communications in virtualized environments. Paper analyzes the main performance bottlenecks in HPC application scalability on the Amazon EC2 Cluster Compute platform by evaluating the communication performance on shared memory and a virtualized 10 Gigabit Ethernet network; assessing the scalability of representative HPC codes, the NAS Parallel Benchmarks, using an important number of cores, up to 512; analyzing the new cluster instances (CC2), both in terms of single instance performance, scalability and cost-efficiency of its use; suggested techniques for reducing the impact of the virtualization overhead in the scalability of communication-intensive HPC codes, such as the direct access of the Virtual Machine to the network and reducing the number of processes per instance; and proposing the combination of message-
passing with multithreading as the most scalable and cost-effective option for running HPC applications on the Amazon EC2 Cluster Compute platform.

Multimedia communication research and development often requires computationally intensive simulations in order to develop and investigate the performance of new optimization algorithms.

Depending on the simulations, they may require even a few days to test an adequate set of conditions due to the complexity of the algorithms. The traditional approach to speed up this type of relatively small simulations, which require several develop–simulate–reconfigure cycles, is indeed to run them in parallel on a few computers and leaving them idle when developing the technique for the next simulation cycle. This work proposed a new cost-effective framework based on cloud computing for accelerating the development process, in which resources are obtained on demand and paid only for their actual usage. Issues are addressed both analytically and practically running actual test cases, i.e., simulations of video communications on a packet lossy network, using a commercial cloud computing service. A software framework has also been developed to simplify the management of the virtual machines in the cloud. Results show that it is economically convenient to use the considered cloud computing service, especially in terms of reduced development time and costs, with respect to a solution using dedicated computers, when the development time is longer than one hour. If more development time is needed between simulations, the economic advantage progressively reduces as the computational complexity of the simulation increases.

It is evident from the above work that majority of them are concentrated on improving the performance using some of the scheduling algorithm or balancing the load by distributing the load into different data center or scheduling to reduce the energy consumption in data center.