

A Survey on Infrastructure Platform Issues in Cloud Computing

Er. Kamna Agarwal, Dr. Sugandha Singh

Abstract—Cloud computing providers have setup several data centers at different geographical locations over the Internet in order to optimally serve needs of their customers around the world. However, existing systems do not support mechanisms and policies for dynamically coordinating load distribution among different Cloud-based data centers in order to determine optimal location for hosting application services to achieve reasonable QoS levels. Main focus in this is on creation of federated Cloud computing environment that facilitates just-in-time, opportunistic, and scalable provision of application services, consistently achieve QoS targets under variable workload, resources and network conditions. The Cloud computing providers are unable to predict geographic distribution of users consuming their services, hence the load coordination is required. The overall goal of this work is to create a stable computing environment to support dynamic expansion and contraction of resources to handle abrupt variations in service demands.

Index Terms— Cloud computing, Cloud data Centers, InterCloud Architecture, Quality of service, federated cloud computing, Components, working of module.

1 INTRODUCTION

Cloud Computing is a technical shift when computing is moved away from personal computers or individual enterprise application server into the cloud of computers. A cloud is a virtualized server pool which can provide different computing resources to different clients. Users of system need to be concern about the computing service being asked for. The term cloud is used as a metaphor for the Internet [1]. IEEE computer society defines cloud computing as: A paradigm in which information is permanently stored in servers on the Internet and cached temporarily on clients that include desktops, Entertainment centers, table computers, notebook, wall computers, handhelds, etc [2]. It is a source for the dynamic provisioning of computing services, typically supported by state-of-the-art data centers containing ensembles of networked Virtual Machines. This technology has made information available as subscription-based services in a pay-as-you-go model to consumers. These services in industry are respectively referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). A Berkeley Report in Feb

2009 states “Cloud computing, the long-held dream of computing as a utility has the potential to transform a large part of the IT industry, making software even more attractive as a service” [2].

Clouds aim to power the next generation data centers by architecting them as a network of virtual services (hardware, database, user-interface, application logic) so that users are able to access and deploy applications from anywhere in the world on demand at competitive costs depending on users QoS (Quality of Service) requirements. Providers such as Amazon [3], Google [4], Salesforce [5], IBM, Microsoft [6], and Sun Microsystems have begun to establish new data centers for hosting Cloud computing application services such as social networking and gaming portals, business applications (e.g., Salesforce.com), media content delivery, and scientific workflows. In order to support a large number of application service consumers from around the world, Cloud infrastructure providers (i.e., IaaS providers) have established data centers in multiple geographical locations to provide redundancy and ensure reliability in case of site failures.

2 RELATED WORK

The key Cloud platforms in Cloud computing domain including Amazon Web Services, Microsoft Azure, Google AppEngine, Manjrasoft Aneka [7], Eucalyptus [8], and GoGrid [9] offer a variety of pre-packaged services for monitoring, managing and provisioning resources and application services. However, the techniques implemented in each of these Cloud platforms vary.

-
- Kamna Agarwal, pursuing M.Tech, Computer Science from Rajasthan Technical University, Rajasthan , India, E-mail: kamnaagarwal7@gmail.com
 - Dr. Sugandha Singh, Supervisor, Head M.Tech(CSE), Email-id:sugandha06@gmail.com

The three Amazon Web Services (AWS), Elastic Load Balancer [10], Auto Scaling and CloudWatch [11] together expose functionalities which are required for undertaking provisioning of application services on Amazon EC2. Elastic Load Balancer service automatically provisions incoming application workload across available Amazon EC2 instances. Auto-Scaling service can be used for dynamically scaling-in or scaling-out the number of Amazon EC2 instances for handling changes in service demand patterns. And finally the CloudWatch service can be integrated with above services for strategic decision making based on real-time aggregated resource and service performance information [12].

Manjrasoft Aneka is a platform for building and deploying distributed applications on Clouds. It provides a rich set of APIs for transparently exploiting distributed resources and expressing the business logic of applications by using the preferred programming abstraction [13]. Aneka is also a market-oriented Cloud platform since it allows users to build and schedule applications, provision resources and monitor results using pricing, accounting, and QoS/SLA services in private and/or public (leased) Cloud environments. Aneka [14] also allows users to build different run-time environments such as enterprise/private Cloud by harness computing resources in network or enterprise data centers, public Clouds such as Amazon EC2, and hybrid clouds by combining enterprise private Clouds managed by Aneka with resources from Amazon EC2 or other enterprise Clouds build and managed using technologies such as XenServer.

Eucalyptus is an open source Cloud computing platform [15]. It is composed of three controllers. Among the controllers, the Cluster Controller is a key component to application service provisioning and load balancing. Each Cluster Controller is hosted on the head node of a cluster to interconnect outer public networks and inner private networks together. By monitoring the state information of instances in the pool of server controllers, the Cluster Controller can select the available service/server for provisioning incoming requests.

Windows Azure Fabric has a weave-like structure, which is composed of node (servers and load balancers), and edges (power, Ethernet and serial communications). The Fabric Controller manages a service node through a built-in service, named Azure Fabric Controller Agent, which runs in the background, tracking the state of the server, and reporting these metrics to the Controller [16]. If a fault state is reported, the Controller can manage a reboot of the server or a migration of services from the current server to other healthy servers. Moreover, the Controller also supports service provisioning by matching the services against the VMs that meet required demands.

GoGrid Cloud Hosting offers developers the F5 Load Balancers for distributing application service traffic across servers, as long as IPs and specific ports of these servers are attached. The load balancer [17] allows Round Robin algorithm and Least Connect algorithm for routing application service requests. Also, the load balancer is able to sense a crash of the server, redirecting further requests to other available servers. But currently, GoGrid Cloud Hosting only gives developers programmatic APIs to implement their custom auto-scaling service.

Google App Engine offers developers a scalable platform in which applications can run, rather than providing access directly to a customized virtual machine. Therefore [18], access to the underlying operating system is restricted in App Engine. And load-balancing strategies, service provisioning and auto scaling are all automatically managed by the system behind the scenes. However, at this time Google App Engine can only support provisioning of web hosting type of applications.

In order to support a large number of application service consumers from around the world, Cloud infrastructure providers (i.e., IaaS providers) have established data centers in multiple geographical locations to provide redundancy and ensure reliability in case of site failures. For example, Amazon has data centers in the US and Europe. However, currently they (i) expect their Cloud customers (i.e., SaaS providers) to express a preference about the location where they want their application services to be hosted and (ii) don't provide seamless/automatic mechanisms for scaling their hosted services across multiple, geographically distributed data centers. This approach has many shortcomings, which include:

- (i) It is difficult for Cloud customers to determine in advance the best location for hosting their services as they may not know origin of consumers of their services and
- (ii) Cloud SaaS providers may not be able to meet QoS expectations of their service consumers originating from multiple geographical locations.

However, no single Cloud infrastructure providers have their data centers at all possible locations throughout the world. As a result Cloud application service (SaaS) providers will have difficulty in meeting QoS expectations for all their users. Hence, they would prefer to logically construct federated Cloud infrastructures (mixing multiple public and private clouds) to provide better support for their specific user needs. This kind of requirements often arises in enterprises with global operations and applications such as Internet service, media hosting, and Web 2.0 applications. This necessitates building

technologies and algorithms for seamless federation of Cloud infrastructure service providers for autonomic provisioning of services across different Cloud providers.

To meet these requirements, next generation Cloud service providers should be able to:

(i) Dynamically expand or resize their provisioning capability based on sudden spikes in workload demands by leasing available computational and storage capabilities from other Cloud service providers;

(ii) Operate as parts of a market driven resource leasing federation, where application service providers such as Salesforce.com host their services based on negotiated Service Level Agreement (SLA) contracts driven by competitive market prices; and

(iii) Deliver on demand, reliable, cost-effective, and QoS aware services based on virtualization technologies while ensuring high QoS standards and minimizing service costs. They need to be able to utilize market-based utility models as the basis for provisioning of virtualized software services and federated hardware infrastructure among users with heterogeneous applications and QoS targets.

However, no single Cloud infrastructure providers have their data centers at all possible locations throughout the world. As a result Cloud application service (SaaS) providers will have difficulty in meeting QoS expectations for all their users. Hence, they would prefer to logically construct federated Cloud infrastructures (mixing multiple public and private clouds) to provide better support for their specific user needs.

2.1 System Architecture and Elements of InterCloud

Fig.1 shows the high level components of the service-oriented architectural framework consisting of client's brokering and coordinator services that support utility-driven federation of clouds: application scheduling, resource allocation and migration of workloads. The architecture cohesively couples the administratively and topologically distributed storage and computes capabilities of Clouds as parts of single resource leasing abstraction. The system will ease the cross-domain capabilities integration for on demand, flexible, energy-efficient, and reliable access to the infrastructure based on emerging virtualization technologies [19][20].

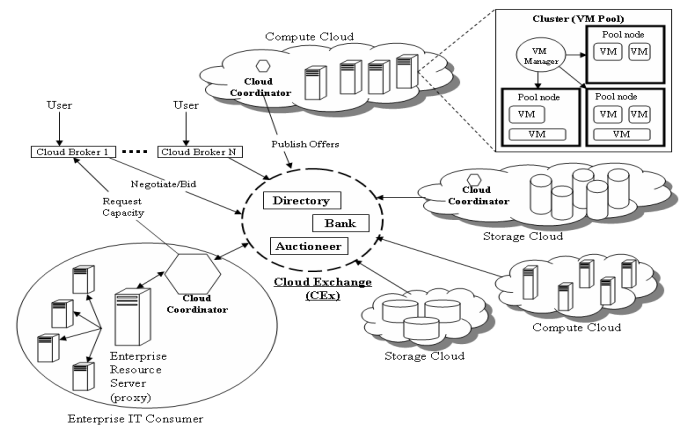


Fig.1: Federated network of clouds mediated by a Cloud exchange

The Cloud Exchange (CEX) acts as a market maker for bringing together service producers and consumers. It aggregates the infrastructure demands from the application brokers and evaluates them against the available supply currently published by the Cloud Coordinators. It supports trading of Cloud services based on competitive economic models [21] such as commodity markets and auctions. CEX allows the participants (Cloud Coordinators and Cloud Brokers) to locate providers and consumers with fitting offers. Such markets enable services to be commoditized and thus, would pave the way for creation of dynamic market infrastructure for trading based on SLAs. An SLA specifies the details of the service to be provided in terms of metrics agreed upon by all parties, and incentives and penalties for meeting and violating the expectations, respectively. The availability of a banking system within the market ensures that financial transactions pertaining to SLAs between participants are carried out in a secure and dependable environment. Every client in the federated platform needs to instantiate a Cloud Brokering service that can dynamically establish service contracts with Cloud Coordinators via the trading functions exposed by the Cloud Exchange.

2.1.1 Cloud Coordinator (CC)

The Cloud Coordinator service is responsible for the management of domain specific enterprise Clouds and their membership to the overall federation driven by market-based trading and negotiation protocols. It provides a programming, management, and deployment environment for applications in a federation of Clouds. Fig.2 shows a detailed depiction of resource management components in the Cloud Coordinator service.

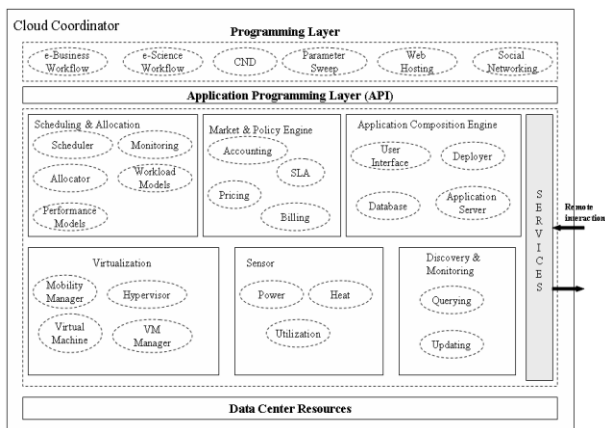


Fig.2: Cloud Coordinator software architecture

The Cloud Coordinator exports the services of a cloud to the federation by implementing basic functionalities for resource management such as scheduling, allocation, (workload and performance) models, market enabling, virtualization, dynamic sensing/monitoring, discovery, and application composition as discussed below:

i) Scheduling and Allocation: This component allocates virtual machines to the Cloud nodes based on user's QoS targets and the Clouds energy management goals. On receiving a user application, the scheduler does the following:

- (a) Consults the Application Composition Engine about availability of software and hardware infrastructure services that are required to satisfy the request locally,
- (b) Asks the Sensor component to submit feedback on the local Cloud nodes' energy consumption and utilization status
- (c) Enquires the Market and Policy Engine about accountability of the submitted request.

A request is termed as accountable if the concerning user has available credits in the Cloud bank and based on the specified QoS constraints the establishment of SLA is feasible. In case all three components reply favorably, the application is hosted locally and is periodically monitored until it finishes execution. Data center resources may deliver different levels of performance to their clients; hence, QoS-aware resource selection plays an important role in Cloud computing. Additionally, Cloud applications can present varying workloads. It is therefore essential to carry out a study of Cloud services and their workloads in order to identify common behaviors, patterns, and explore load forecasting approaches that can potentially lead to more efficient scheduling and allocation. In this context, there is need to analyse sample applications and

correlations between workloads, and attempt to build performance models that can help explore trade-offs between QoS targets.

ii) Market and Policy Engine: The SLA module stores the service terms and conditions that are being supported by the Cloud to each respective Cloud Broker on a per user basis. Based on these terms and conditions, the Pricing module can determine how service requests are charged based on the available supply and required demand of computing resources within the Cloud. The Accounting module stores the actual usage information of resources by requests so that the total usage cost of each user can be calculated. The Billing module then charges the usage costs to users accordingly.

Cloud customers can normally associate two or more conflicting QoS targets with their application services. In such cases, it is necessary to trade off one or more QoS targets to find a superior solution. Due to such diverse QoS targets and varying optimization objectives, we end up with a Multi-dimensional Optimization Problem (MOP). For solving the MOP, one can explore multiple heterogeneous optimization algorithms, such as dynamic programming, hill climbing, parallel swarm optimization, and multi-objective genetic algorithm.

iii) Application Composition Engine: This component of the Cloud Coordinator encompasses a set of features intended to help application developers create and deploy [18] applications, including the ability for on demand interaction with a database backend such as SQL Data services provided by Microsoft Azure, an application server such as Internet Information Server (IIS) enabled with secure ASP.Net scripting engine to host web applications, and a SOAP driven Web services API for programmatic access along with combination and integration with other applications and data.

iv) Virtualization: VMs support flexible and utility driven configurations that control the share of processing power they can consume based on the time criticality of the underlying application. However, the current approaches to VM-based Cloud computing are limited to rather inflexible configurations within a Cloud. This limitation can be solved by developing mechanisms for transparent migration of VMs across service boundaries with the aim of minimizing cost of service delivery (e.g., by migrating to a Cloud located in a region where the energy cost is low) and while still meeting the SLAs. The Mobility Manager is responsible for dynamic migration of VMs based on the real-time feedback given by the Sensor service. Currently, hypervisors such as VMware and Xen have a limitation that VMs can only be migrated between hypervisors that are within the same subnet and share common storage. Clearly,

this is a serious bottleneck to achieve adaptive migration of VMs in federated Cloud environments. This limitation has to be addressed in order to support utility driven, power-aware migration of VMs across service domains.

v) Sensor: Sensor infrastructure will monitor the power consumption, heat dissipation, and utilization of computing nodes in a virtualized Cloud environment. To this end, we will extend our Service Oriented Sensor Web [22] software system. Sensor Web provides a middleware infrastructure and programming model for creating, accessing, and utilizing tiny sensor devices that are deployed within a Cloud. The Cloud Coordinator service makes use of Sensor Web services for dynamic sensing of Cloud nodes and surrounding temperature. The output data reported by sensors are feedback to the Coordinator's Virtualization and Scheduling components, to optimize the placement, migration, and allocation of VMs in the Cloud. Such sensor-based real time monitoring of the Cloud operating environment aids in avoiding server breakdown and achieving optimal throughput out of the available computing and storage nodes.

vi) Discovery and Monitoring: In order to dynamically perform scheduling, resource allocation, and VM migration to meet SLAs in a federated network, it is mandatory that up-to-date information related to Cloud's availability, pricing and SLA rules are made available to the outside domains via the Cloud Exchange. This component of Cloud Coordinator is solely responsible for interacting with the Cloud Exchange through remote messaging. The Discovery and Monitoring component undertakes the following activities:

- (a) Updates the resource status metrics including utilization, heat dissipation, power consumption based on feedback given by the Sensor component
- (b) Facilitates the Market and Policy Engine in periodically publishing the pricing policies, SLA rules to the Cloud Exchange
- (c) Aids the Scheduling and Allocation component in dynamically discovering the Clouds that offer better optimization for SLA constraints such as deadline and budget limits
- (d) Helps the Virtualization component in determining load and power consumption; such information aids the Virtualization component in performing load-balancing through dynamic VM migration.

Further, system components will need to share scalable methods for collecting and representing monitored data. This leads us to believe that we should interconnect and monitor system components based on decentralized

messaging and information indexing infrastructure, called Distributed Hash Tables (DHTs) [23]. However, implementing scalable techniques that monitor the dynamic behaviors related to services and resources is non-trivial. In order to support a scalable service monitoring algorithm over a DHT infrastructure, additional data distribution indexing techniques such as logical multi-dimensional or spatial indices [17] should be implemented.

2.1.2 Cloud Broker

The Cloud Broker acting on behalf of users identifies suitable Cloud service providers through the Cloud Exchange and negotiates with Cloud Coordinators for an allocation of resources that meets QoS needs of users. The architecture of Cloud Broker is shown in Fig.3 and its components are discussed below:

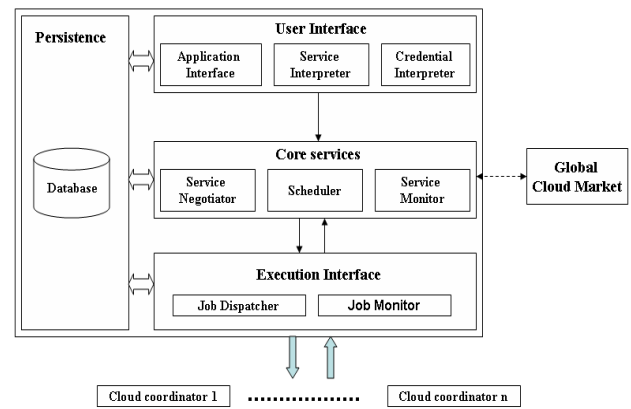


Fig.3: High level architecture of Cloud Broker service

i) User Interface: This provides the access linkage between a user application interface and the broker. The Application Interpreter translates the execution requirements of a user application which include what is to be executed, the description of task inputs including remote data files (if required), the information about task outputs (if present), and the desired QoS. The Service Interpreter understands the service requirements needed for the execution which comprise service location, service type, and specific details such as remote batch job submission systems for computational services. The Credential Interpreter reads the credentials for accessing necessary services.

ii) Core Services: They enable the main functionality of the broker. The Service Negotiator bargains for Cloud services from the Cloud Exchange. The Scheduler determines the most appropriate Cloud services for the user application based on its application and service requirements. The Service Monitor maintains the status of Cloud services by periodically checking the availability of known Cloud services and discovering new services

that are available. If the local Cloud is unable to satisfy application requirements, a Cloud Broker lookup request that encapsulates the user's QoS parameter is submitted to the Cloud Exchange, which matches the lookup request against the available offers. The matching procedure considers two main system performance metrics: first, the user specified QoS targets must be satisfied within acceptable bounds and, second, the allocation should not lead to overloading (in terms of utilization, power consumption) of the nodes. In case the match occurs the quote is forwarded to the requester (Scheduler). Following that, the Scheduling and Allocation component deploys the application with the Cloud that was suggested by Cloud market.

iii) Execution Interface: This provides execution support for the user application. The Job Dispatcher creates the necessary broker agent and requests data files (if any) to be dispatched with the user application to the remote Cloud resources for execution. The Job Monitor observes the execution status of the job so that the results of the job are returned to the user upon job completion.

iv) Persistence: This maintains the state of the User Interface, Core Services, and Execution Interface in a database. This facilitates recovery when the broker fails and assists in user-level accounting.

2.1.3 Cloud Exchange (CEx)

As a market maker, the CEx acts as an information registry those stores the Cloud's current usage costs and demand patterns. Cloud Coordinators periodically update their availability, pricing, and SLA policies with the CEx. Cloud Brokers query the registry to learn information about existing SLA offers and resource availability of member Clouds in the federation. Furthermore, it provides match-making services that map user requests to suitable service providers. Mapping functions will be implemented by leveraging various economic models such as Continuous Double Auction (CDA) as proposed in earlier works.

As a market maker, the Cloud Exchange provides directory, dynamic bidding based service clearance, and payment management services as discussed below.

i) Directory: The market directory allows the global CEx participants to locate providers or consumers with the appropriate bids/offers. Cloud providers can publish the available supply of resources and their offered prices. Cloud consumers can then search for suitable providers and submit their bids for required resources. Standard interfaces need to be provided so that both providers and consumers can access resource information from one another readily and seamlessly.

ii) Auctioneer: Auctioneers periodically clear bids and asks received from the global CEx participants. Auctioneers are third party controllers that do not represent any providers or consumers. Since the auctioneers are in total control of the entire trading process, they need to be trusted by participants.

iii) Bank: The banking system enforces the financial transactions pertaining to agreements between the global CEx participants. The banks are also independent and not controlled by any providers and consumers; thus facilitating impartiality and trust among all Cloud market participants that the financial transactions are conducted correctly without any bias. This should be realized by integrating with online payment management services, such as PayPal, with Clouds providing accounting services.

3 EXPERIMENT AND RESULT

The experiment aims at proving that federated infrastructure of clouds has potential to deliver better performance and service quality as compared to existing non-federated approaches. To this end, a simulation environment that models federation of three Cloud providers and a user (Cloud Broker) is modeled. Every provider instantiates a Sensor component, which is responsible for dynamically sensing the availability information related to the local hosts. Next, the sensed statistics are reported to the Cloud Coordinator that utilizes the information in undertaking load-migration decisions. Load-migration police is done only when the origin provider does not have the requested number of free VM slots available [24]. The migration process involves the following steps:

- i) Creating a virtual machine instance that has the same configuration, this is supported by the different provider.
- ii) Migrating the applications assigned to the original virtual machine to the newly instantiated virtual machine at the destination provider. The federated network of Cloud providers is created based on the topology shown in Fig.4.

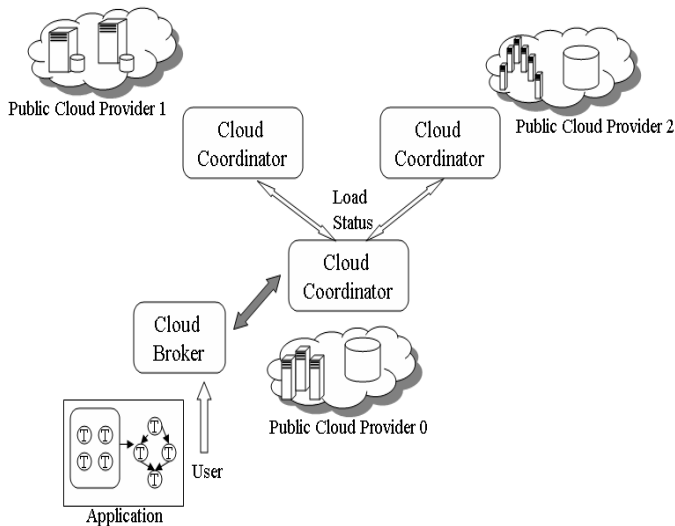


Fig.3.4: A network topology of federated Data Centers

Every Public Cloud provider in the system is modeled to have 50 computing hosts, 10GB of memory, 2TB of storage, 1 processor with 1000 MIPS of capacity, and a time-shared VM scheduler. Cloud Broker on behalf of the user, request a VM that requires 256MB of memory, 1GB of storage, 1 CPU, and time-shared Cloudlet scheduler. The broker requests instantiation of 25 VMs and associates one Cloudlet (Cloud application abstraction) to each VM to be executed. These requests are originally submitted with the Cloud Provider 0. Each Cloudlet is modeled to have 1800000 MIs. The simulation experiments were run under the following system configurations:

- i) A federated network of clouds is available; hence data centers are able to cope with peak demands by migrate the excess of load to the least loaded ones.
- ii) The data centers are modeled as independent entities (without federation).

All the workload submitted to a Cloud provider must be processed and executed locally. Table.3.1 shows the average turn-around time for each Cloudlet. A user application consists of one or more Cloudlets with sequential dependencies. The simulation results reveal that the availability of federated infrastructure of clouds reduces the average turn-around time by more than 50%. It shows that, even for a very simple load-migration policy, availability of federation brings significant benefits to user's application performance.

TABLE 1
RESULT

Performance Metrics	With Federation	Without Federation	% Improvement
Average Turn Around Time (Secs)	2221.13	4700.1	>50%

Average Turn Around Time (Secs)	2221.13	4700.1	>50%
---------------------------------	---------	--------	------

4 CONCLUSION

Development of fundamental techniques and software systems that integrate distributed clouds in a federated fashion is critical to enabling composition and deployment of elastic application services. This experiment will make significant scientific advancement in understanding the theoretical and practical problems of engineering services for federated environments. The resulting framework facilitates the federated management of system components and protects customers with guaranteed quality of services in large, federated and highly dynamic environments. The different components cloud federation offer powerful capabilities to address both services and resources management, but their end-to-end combination aims to dramatically improve the effective usage, management, and administration of Cloud systems. This will increase degrees of scalability, flexibility, and simplicity for management and delivery of services in federation of clouds.

ACKNOWLEDGMENT

It gives me immense pleasure to put forward this practical venture. But surely, it would not have been possible without proper guidance and encouragement. So I would like to thank all those people without whose support this dissertation would not have been a success.

I owe a special thanks to Prof. Sugandha Singh (HOD-CSE) for her able guidance and valuable suggestions for choosing and shaping this thesis dissertation topic. Her contribution has helped a great deal in the timely completion of this report.

REFERENCES

- [1] Jonesrayen, Jan 10, 2011, Tech news, cloud computing.
- [2] Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. University of California at Berkley, USA. Technical Rep UCB/EECS-2009-28 (2009)
- [3] Guohui Wnag: The Impact of Virtualization on Network performance of Amazon EC2 Data Center. In INFOCOM, Proceedings IEEE, San Diego, CA (March 2010)
- [4] Sean Lynch on Announcing Google App Engine for business, May, 19, 2010
- [5] Goslin. L.N on New Product Development: Salesforce and Cycle Time. In International Conference on Management and Technology, Portland (July 1997)

- [6] Subramanian, V; Liqiong Wang; En-Jui Lee; Chen, P: on Rapid Processing on Synthetic Seismograms using Windows Azure Cloud. In IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom). (December 2010)
- [7] Vecchiola, C., Chu, X., Buyya, R.: Aneka: A Software Platform for .NET-based Cloud Computing. In: Gentsch, W., Grandinetti, L., Joubert, G. (eds.) High Speed and Large Scale Scientific Computing, pp. 267–295. IOS Press, Amsterdam (2009), ISBN: 978-1-60750-073-5
- [8] Nurmi, D., Wolski, R., Grzegorzczak, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The Eucalyptus Open-Source Cloud-Computing System. In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2009), Shanghai, China, May 18-May 21 (2010)
- [9] Michael Sheehan: Understanding “Clouded” Terms of Cloud Computing, June, 6, 2008.
- [10] James Halminton: AWS Ships Monotoring, Auto Scaling and, Elastic load Balancing, May, 18, 2009.
- [11] Jui-Hao Chiang: Physical Machine State Migration. In International Conferencr on Parallel and Distributed System (ICPADS), Tainan. (Dec 2011).
- [12] Lee, K.; Murray, D.; Hughes, D. and Joosen, W.: Extending Sensor Network into the Cloud using Amazon web Services, in International Conference on network Embedded System for Enterprise Applications, Suzhou, China (Nov 2010)
- [13] C. Vecchiola, X. Chu, M. Mattess, and R. Buyya, Aneka - Integration of Private and Public Clouds, Cloud Computing: Principles and Paradigms, ISBN-13: 978-0470887998, Wiley Press, New York, USA, 2010.
- [14] D. Duncan, X. Chu, C. Vecchiola, and R. Buyya, The Structure of the New IT Frontier: Cloud Computing, Strategic Facilities Magazine, Issue 9, Pages: 67-72, Pacific & Strategic Holdings Pte Ltd, Singapore, August/September 2009.
- [15] Waqar, Adeela Raza, Abbas, Haider: User Privacy Issues in Eucalyptus: A private cloud Computing Environment. In International Conference on Trust, Security and Privacy in Computing & Communication (TrustCom), Changsha, China (Nov 2011)
- [16] Rabi Prasad Padhy; Manas Ranjan; Suresh Chandra Satapathy: WINDOWS AZURE PAAS CLOUD: AN OVERVIEW. In International Journal of Computer Application, Issues 2, Volume 1 (Feb 2012)
- [17] Michael Sheehan: GoGrid Exchange: CohesiveFT VPN-cubed Available on GoGrid Cloud, (Jan, 2010)
- [18] Malawski, M; Kuzniar, M; Wojcik, P; Bubak, M; How to Use Google App Engine for free Computing, in AGH University of Science & Technology University of Notre Dame, Krakow Notre Dame (Nov 2011)
- [19] Barham, P., et al.: Xen and the Art of Virtualization. In: Proceedings of the 19th ACM Symposium on Operating Systems Principles. ACM Press, New York (2003)
- [20] Buyya, R., Abramson, D., Giddy, J., Stockinger, H.: Economic Models for Resource Management and Scheduling in Grid Computing. Concurrency and Computation: Practice and Experience 14(13-15), 1507–1542 (2002)
- [21] Chu, X., Buyya.: Services Oriented Sensor Web. In: Mahalik, N.P. (ed.) Sensor Network and Configuration: Fundamental, Standards, Platforms, and Applications, January 2007. Springer, Berlin (2007)
- [22] Lua, K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A Survey and Comparison of Peerto-Peer Overlay Network Schemes. In: Communications Surveys and Tutorials, Washington,DC, USA, vol. 7(2) (2005)
- [23] Ranjan, R.: Coordinated Resource Provisioning in Federated Grids. Ph.D. Thesis, The University of Melbourne, Australia (March 2009)
- [24] R. Buyya, R. Ranjan, R. Calheiros: InterCloud: Utility-oriented federation of cloud Computing envirmnts for scaling of application service. In Lecture notes on Algorithms and Parallel Processing in Computer Science, 2011. (vol. 6081).