

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/286583362>

Cloud Broker: Working in federated structures

Conference Paper · September 2014

DOI: 10.1109/ICACCI.2014.6968660

CITATIONS

4

READS

139

3 authors:



Prashant Khanna

Waikato Institute of Technology

11 PUBLICATIONS 35 CITATIONS

[SEE PROFILE](#)



Sonal Jain

JK Lakshmipat University

23 PUBLICATIONS 376 CITATIONS

[SEE PROFILE](#)



Galgotias University

Galgotias University

235 PUBLICATIONS 6,309 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Reactive Extraction [View project](#)



Biomass Gasification [View project](#)

Cloud Broker: Working in Federated Structures

A Hybrid Cloud Live Performance Analysis

Prashant Khanna,
Institute of Engineering and Technology,
JK Lakshmipat University
Jaipur, India
perukhan@gmail.com,

Sonal Jain
Institute of Engineering and Technology,
JK Lakshmipat University
Jaipur, India
sonaljain@jklu.edu.in

BV Babu
Galgotias University
Greater NOIDA, India
vc@galgotiasuniversity.edu.in

Abstract— The research analyzes the functioning of an Autonomic Cloud Broker (ACB) in a real life scenario utilizing a Federated Cloud Infrastructure. The paper analyzes two important use cases that exist while a Cloud Broker enables the provision of services from multiple cloud providers to simultaneously demanding cloud users. Varying, real time load conditions are generated in a live use case on a private cloud with distributed cloud brokers working in a federated manner with cloud bursting handled through a common interface visible to the Amazon Web Services (AWS) cloud infrastructure. The analysis engine is generated using the NewRelic Analysis engine and results on the performance of the broker are derived in real time. The paper highlights real world issues plaguing the Cloud Brokerage framework and indicates ways to mitigate the same using the Federated Cloud Infrastructure combining the Public Cloud and a private cloud. The research asserts that it is possible to create Autonomic Cloud Brokers, albeit in a tightly integrated and fine tuned cloud environment.

Keywords—Autonomic Cloud Broker, Cloud Brokerage, Hybrid Cloud, Federated Cloud Infrastructure, AWS, Responsiveness

I. INTRODUCTION

Cloud computing is alive and kicking. This is despite early naysayers and initial dooms day predictions professing to the contrary. It's a concept that has become the toast of conferences and meetings in the Information Technology domains today and we find increasing evidence of its utility and adoption in the industry as well as the academia. The framework for the cloud is maturing and as it matures and its acceptance increases, there is also a need to review use cases that evolve along with its acceptance. The NIST [1] model proposes a framework which has become increasingly accepted in academia and the authors would be using the same for presenting this study and discussing an ignored player in the NIST model – the Cloud Broker. As the model develops from a single player scenario for cloud provisioning to the existence of multiple cloud offerings, the concept of

Intercloud [2], [3] and Federated Cloud [4], [5] leading to Cloud Interoperability requirements have driven the need for a player that takes up the role of managing the differing services in an automated and efficient manner. The role is a Cloud Broker assumes importance when such a need for management comes to fore. This research studies the role of an autonomic cloud broker implemented in a federated cloud environment with data centers volunteering to part with the usage metrics and the cloud brokers managing the optimal usage of the resources. The study was done on a Pan-India scale and results collated for the study at Jaipur, India.

The rest of the paper is structured as under: In section II, we refer to the State of Art on current relevant research in Intercloud experimentation for autonomic cloud brokers. Section III describes, in general, the Federated structure for cloud computing and its implication on the present the Research structure. Section IV discusses in detail the role of the experimental broker, the real life setup for the install and the test results. We also describe the use case scenarios for cloud broker implemented in a hybrid cloud using the federated cloud environments. The section also presents the research findings. In section V we conclude and describe future work on the subject.

II. STATE OF ART

Though Cloud computing is a highly studied topic today and a large body of research has gone into studying specific standards of interoperability amongst clouds and how they are to be achieved, the aspects of brokering services to the end client from amongst those available is finding refereed status only recently through [6], [7]. Most authors have converged on the definition of a Cloud Broker, using the NIST definition as a baseline, which generically defines a cloud broker as an entity that manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers. Voids in the definition and implementation of autonomic services by a Cloud Broker are

evident in most definitions and have been highlighted by several authors in the recent past. These challenges are compelling the cloud creators to relook at the accepted definitions and the standard use cases for cloud computing adoption and research models are changing.

The evolving architecture of the cloud, whether public, private, community or hybrid, would make it non trivial to implement an autonomic framework for creating efficient ecosystems in the cloud [8]. Frameworks would fail in the absence of a federated and self sufficient interfacing structure, coupled by accurate measuring and monitoring mechanisms for provision of cloud services, unless standard interfaces are adopted. This is especially true for a use case when an autonomic broker is aggregating and arbitrating services, from multiple cloud service providers, and packaging them as a service bundle for the end user. Several works have in the recent past attempted to arrive at a universal cloud interaction mechanisms that make them seamless to messages traversals across clouds. Recent works on the subject include [9], [10] that pertains to Intercloud interactions in a federated cloud structure and the issues of Broker Management. All these works point to the increasing complexity of designing an autonomic brokering service which can seamlessly interact across a multitude of cloud standards, networking standards and also mobile computing standards. The resulting deductions are that federation of cloud and interoperability – interaction and message passing; in a cloud environment will be necessitated by employing standard interfaces that can understand multiple languages and paradigms or are interconnecting private clouds to public ones using networking paradigms.

Cloud computing is still in its early days and suffers from lack of standardization, whether in interactions that happen between clouds, or, in the way resource management is done[2]. This is especially evident in the way disparate cloud installs interact and try and talk to each other. As the model matures, interoperability issues would be sorted and stability would hopefully be achieved through sustained research and work being done. Brokering would play a very major role in this. By using a service broker, which translates messages between different Cloud interfaces, cloud service users are able to switch between different Clouds and Cloud providers can interoperate. Standardization of interfaces is another common method for realization of interoperability and autonomic cloud brokers act as the standard interfaces[7].

There are numerous benefits of an interconnected and federated cloud environment. These are evident for both the cloud providers and the cloud users. They can greatly assist in avoiding vendor lock-in, permit scalability, higher availability, low access latency, and a more energy efficient solution. Recent work by Toosi et al in [7] highlights the same in great detail. Cloud interoperability requires cloud providers to adopt and implement standard interfaces, protocols, formats and architectural components that facilitate collaboration. Without these provider-centric changes Cloud interoperability is hard to achieve. Among different provider-centric approaches Hybrid Cloud, Cloud Federation and Inter-Cloud are the most prominent scenarios and are highly studied subjects. Several recent works by the CLOUDS Labs, Melbourne and the work

by the IEEE Workgroup P2301 and P2302 on Standards for Intercloud Interoperability and Federation have focused on the way the clouds can interoperate and standards defined. Both these solutions are trying to integrate a solution which heavily derives from the rules of Networking and OSPF with an XMPP blend. However, a universal standard is still some distance away. In such a scenario, federated clouds, which volunteer for establishing a relation, are the way ahead. The present research was undertaken to understand and implement such a use case scenario where the standard interfaces, or, the cloud brokers were defined between the private network and the public one. The efficacy of the implementation was observed and it was seen that through some amount of fine tuning, an autonomic broker is a reality.

III. FEDERATION IN THE CLOUD AND RESEARCH STRUCTURE

A Federation in the cloud is achieved when a set of cloud providers voluntarily interconnect their infrastructures in order to allow sharing of resources among each other. The federation partnership in clouds leads to the establishment process which can be schematized as Discovery, Matchmaking and Authentication of cloud resource requests.

Federation brings to fore new use cases to the cloud computing paradigm. Besides, the traditional scenario where cloud providers offer cloud-based services to their providers, federation bring about a new paradigm where cloud players can exchange cloud services like storage and computational power in a market like environment. This can be done with other clouds providers, or with brokers engaging in the market dynamics. Presently there are established standards which govern the internal cloud management such as Cloud Data Management Interface (CDMI) [11], Open Cloud Computing Interface (OCCI) [12] and Open Virtual Machine Format (OVF) [13].

A host of recent researches on the subject of federation and cloud brokerage have been done using simulations, and mathematical studies based on network structures created using the Distributed Computing analogies, there are referred works existing which have done a real life study of the concept. This is obvious due to spatial and economic reasons of existing truly federated clouds. The economic impact of conducting such experiments is non-trivial and throws serious logistic management problems for the people conducting it.

In this research the authors created a live scenario for procuring the resources through the Intercloud infrastructure on a federated cloud. Studying the real life implementation of a autonomic broker in a federated cloud brought to fore some very novel and interesting issues of experiment and broker management. One of the scenarios implemented during the testing of the broker was when a cloud needed to procure resources from other clouds in the following manner:

- Storage Space - The cloud providing storage services to a user feels a shortage of storage resources. In order to continue providing cloud-based service to its user, the cloud broker decides to initiate a new resource request, using an identified interface, from other clouds.

- **Distributed Availability** - The cloud scenario was implemented across a disparate geographical structure. When there was a need to deploy a distributed cloud-based service in different geographical locations, the cloud broker acquires the resources placed in those target locations.
- **Migration and Stability** - The cloud broker needed to migrate cloud-based service instances in other clouds in order to accomplish service relocation and power saving. At the same time, the cloud broker decided to provide resources to other clouds needing the resource when it realizes that its datacenter is under-utilized for a given service at a given time.

While setting up the experiment for the research, multiple instance of the broker were installed at the interface points to discover the resources, map the requirements and process the acquisition of the same, after authentication of the requirement. The virtual machine (VM) template used made use of the following metrics for making the mapping:

- The number of cores or processes to be assigned to the VM.
- Amount of memory required for the task.
- Pricing per resource.

Based on the mapping interface used, which in this case was a first in first out, best match on least price algorithm, also used by OpenNebula broker, the cloud broker provided the resources to the requests being generated in real time. The federation implementation for this research has been done using seven data centers in a private cloud interfaced to the AWS cloud for a cloud bursting scenario. The metering and analysis engine used for judging the performance of the setup and the broker was done using the New Relic [14] utilities. The metrics for the performance were:

- Responsive of the request completion,
- The loading factor on individual servers,
- The total usage of memory per VM per data center,
- Overall cost per data center per request.

The test use cases and cloud user requests were generated using an automated tool called Loader.io [15] which loaded the servers from the client central premise. The test cases were based on an extensive multi user search on the AWS cloud. As this research is based on live experiments made by the author on a federated cloud environment with multiple data centers talking to each other, authentication of request was limited to a simple token approach by the utility/cloud broker. In the present case, the broker made a request for the availability of VMs and queries the status. The broker collates all the requests in a Hash Table and sends the collated requests to the central premise for the mapping function to be executed. The latency delays were seen to be minimal as the experiment was done on a private network and the VPNs between the data centers were dedicated. The user request list contained the desired service needed in the form of a template to the geographically distributed broker, who in turn first did a local

mapping for availability and then forwarded the same to the central repository.

The authors in this paper would prefer to call brokerage the discovery and matchmaking phase needed for the creation of a federation of clouds. This research studied four possible federation schemas in which this cloud brokerage may take place. These are:-

- Composite Schema
- Hierarchical Schema
- Distributed Schema
- Hybrid or Mixed Schema

While there are numerous use cases which can be extrapolated using the four schemas, for the present research, the authors have studied the Composite and Distributed Schemas and conducted actual experiments based on the scenarios provided. It is pertinent to mention, that earlier work by the authors in the same field were on the implementation of the broker in the AWS cloud [4], [10] and [16].

IV. EXPERIMENT STRUCTURE AND RESULTS

As is illustrated in Fig 1 below, the author used a private network to create a distributed engine based on the Drupal platform which was used to create an application capable of hosting multiple service scenarios. The broker was housed at the local facility of Jaipur, India and multiple request scenarios were generated and pushed through the broker. The analysis engine used was New Relic [14] and Drupal version 7 was used to create the central utility. The PHP library SimpleCloud was used to implement the Cloud paradigm on the Drupal Interface.

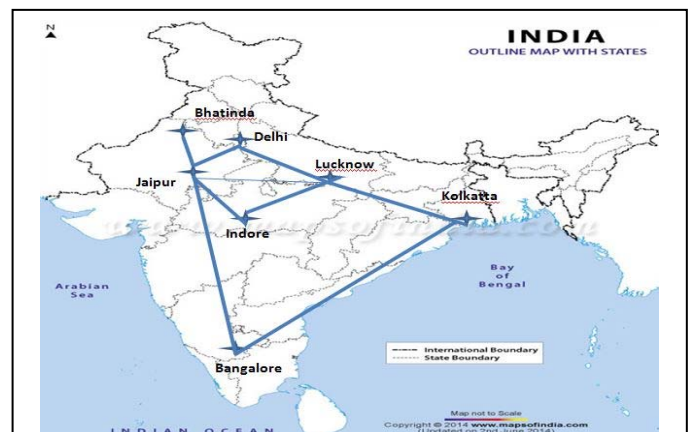


Fig 1. Live experimentation setup with distributed brokers across seven cities in India

In the case of the Composite Federated Schema, private data centers at Jaipur, Delhi, Lucknow and Indore were used to create the cloud infrastructure. Kolkata was used as a limb to test certain externalities. The base tests were around the adoption and creation of new VMs across the server stacks and the usage of additional memory beyond the baseline memory usage available for standard servers. Different use

case scenarios depicted in the tables below indicate the different conditions and load tests generated.

Use Case I : Composite Federated Schema.

The Composite Federated Schemas are for a use case where a single broker, common to all clouds, is in charge of establishing the federation. Such schemas are often visible in private clouds which are disparate and distributed across geographical locations. The schema was tested across the private network of the author and the figure 1 illustrates the physical disposition of the data centers housing the Servers.

TABLE I. SCENARIO OF LOAD TESTS FOR COMPOSITE FEDERATED SCHEMA

Scenario	Load Test Requests	Servers in Use	Time Duration	Type of Requests
Scenario 1	100 Searches on the Servers	5 Concurrent Servers at Three Locations	60 - 400 secs	Singular Jobs
Scenario 2	60 Searches on the Server + 40 across servers	5 Concurrent Servers at Two locations	60 - 400 secs	Batch Jobs
Scenario 3	50 + 50 Search and Swaps on the Server	4 Concurrent Servers at Three Locations	60- 400 secs	Periodical Jobs (50% singular mixed)

The tests cases illustrated in Table I were conducted across a seven day period and to normalize the results, 10 separate sets of similar experiments were conducted for each use case to eliminate network latency issues and to ensure that latency errors were reduced to a minimum. Three types of jobs, singular, batch and periodical jobs, were sent to the cloud broker through the network and the response of the broker was analyzed against in terms of successful job execution. The test results for the load under the Scenario depicted above using the data centrals that are geographically distributed, as shown in Table II are as under:

TABLE II. TEST RESULTS FOR COMPOSITE FEDERATED SCHEMA

Scenario	Total Req-uest	Function	Test Duration	Req per Sec	Time per Req	CPU Load	ME M
Scenar io 1	100	Search	377 Sec	0.27	3.7 (3.7)	10%	1.7 GB used
Scenar io 2	100	Search Across Servers	141 Sec	0.71	1.4 (2.8)	13%	1.7 GB used
Scenar io 3	100	Search + Swap	125 Sec	0.79	1.2 (3.7)	18%	1.7 GB used

The CPU and memory load using the cloud broker in the two main scenarios were as depicted in the graphs. It is pertinent to note that the load conditions on the servers were severe but the resultant distribution and accomplishment of the requests was very high when the composite scenario was used with the proposed cloud broker acting in a centralized manner from Jaipur, India. In the scenarios depicted, the broker initially used a linear model for request distribution in a centralized manner by using a simple FIFO queue to push the cloud request to the cloud stack.

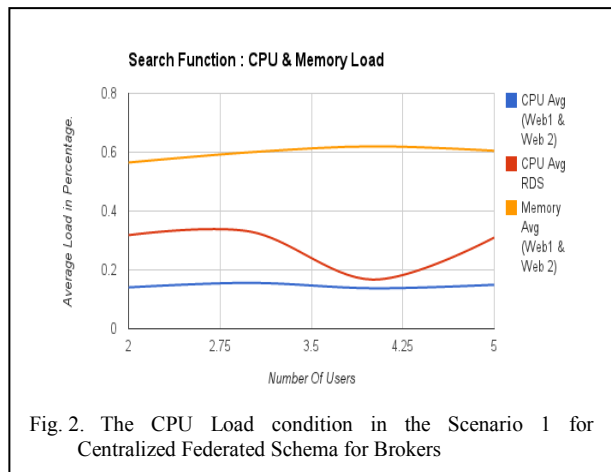


Fig. 2. The CPU Load condition in the Scenario 1 for Centralized Federated Schema for Brokers

Fig. 2 illustrates the CPU load and the Memory Load on the overall Datacenter metrics analyzed through the New Relic engine. While the memory requirements did peak at around 60% of the server capacity, the CPU load was consistently lower than 20% of peak load.

In Fig. 3, the load condition for a combination of searches and record swaps (a multiple, mixed load) shows improved results even in the case of increased number of requests (higher number of users per second.) when using a Composite Schema Broker. Its is also visible that the servers tend to peak at close to 80% when the number of concurrent requests reach a figure of 50 per second, which is an application peak in terms of the load it can handle per server.

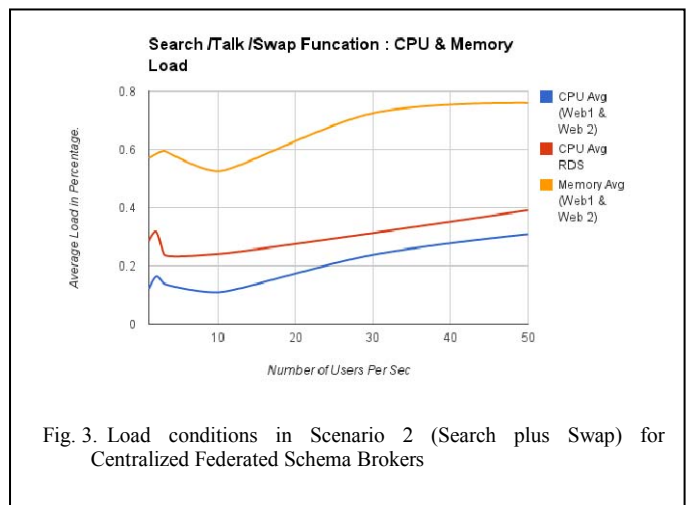


Fig. 3. Load conditions in Scenario 2 (Search plus Swap) for Centralized Federated Schema Brokers

Use Case II – Distributed Federated Schema

In the Distributed Federated Schema tested by the authors, the broker is acting as medium to integrate the services which are available across distributed servers/data centers which seamlessly talk to each other – thus enforcing a federated structure. However, it is possible to simulate a broken federation structure when required. This implies that there can be occasions when some members of the federation of clouds do not talk to each other or provide resource templates. The Broker would be required to implement suitable mechanisms to make the service providers talk to each other and often acts

as a third party to implement the resource request. There is a use case here for a broker to act as an intermediary as well as an aggregator of multiple existing or new resources available from the cloud service providers. The schema was implemented across Jaipur, Indore, Kolkata, Bangalore and Bhatinda data centers in the present research. The data centers across the locations were given a base line install of 200 VMs per Data Center and the traffic requests were routed through the broker available at Jaipur and Kolkata. The brokers available for the test were using a serial FIFO logic to implement the request. The General Meta-Broker Service, as described by [17], was utilized for describing the requested service and as far as concerned, the provisioning of the service was transparent to the client utilizing the service.

To maintain consistent conditions of load, the test case scenarios for the Distributed Federated Schema under live traffic conditions were similar to the ones used for the Composite one. The metrics for obtaining the results was the response time a request took to be completed. The number of request which were not delivered as a result of over loading was also a metric which was checked again default conditions and against the special case of implementation of the broker.

The results of the Distributed Schema tests are in Table III and show the CPU averages on the web server and the database server located at Jaipur main facility.

TABLE III. RESULTS OF DISTRIBUTED FEDERATED SCHEMA USING THE LINEAR BRKER ON A LOCAL DATA CENTER.

No of clients per Sec	CPU Avg (Web Server)	CPU Avg RDS	Memory Avg (Web 1)
2	14.05%	31.80%	57%
3	15.60%	33.00%	60%
4	13.75%	16.70%	62%
5	14.95%	31.00%	61%

While the results in Table III are indicative of a performance that matches local brokers in a linear distributive role, where the requets are processes in a FIFO manner, initially from the local data center and then from the external ones, Table IV is clearly depicting the way the hybrid cloud system behaves as a whole when severe load is posted on the system from multiple clients having concurrent requests to take the same resource.

TABLE IV. RESULTS OF DISTRIBUTED FEDERATED SCHEMA USING MULTIPLE CLIENTS AT DISTRIBUTED DATA CENTERS

No of clients per Sec	CPU Avg (Web1)	CPU Avg RDS	Memory Avg (Web1)
1	12.05%	28.50%	57%
2	16.35%	31.70%	59%
3	13.85%	23.90%	60%
10	10.87%	24.10%	53%
30	23.80%	31.30%	73%
50	30.90%	39.10%	76%

Fig. 4 shows the response time of the client at Jaipur as the number of concurrent requests increase over time. The test was conducted using a response time as a metric for uploading a high resolution image on a web page and concurrent search and swap requests being made on the servers.

It is evident from the results that the two schemas which have been tested depict a very clear business case for different scenarios. While the Composite Federation Schema for Cloud Broker can work very efficiently in a private network with a single broker taking the onus of managing local requirements and passing the additional requirements to the central server at a different location using a request template, the Distributed one works in cases when there is a significant time delay or even break in the federation structure.

In the present set of experiments, the breaking of association between members of a federation was depicted by closing off of multiple communication channels between data centers and then checking the performance of the broker. In a real life scenario, this can frequently occur for a multitude of reasons. The joining and leaving of clouds providers might lead to reconfiguration of the cloud federated structure, the present set of experiments depict that the hybrid model used prompted the broker to use the cloud bursting use case to transfer the load to the AWS engine.

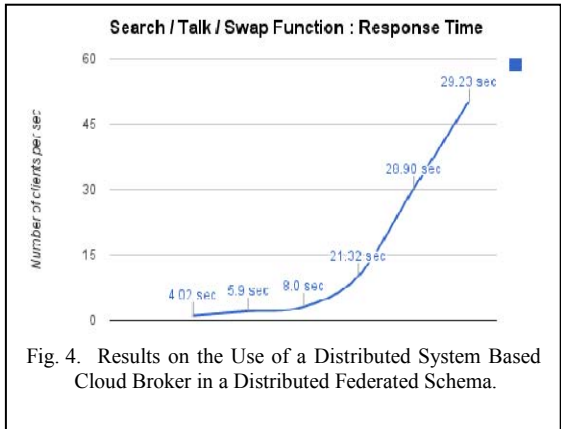


Fig. 4. Results on the Use of a Distributed System Based Cloud Broker in a Distributed Federated Schema.

Distributed Federated schemas however added a new layer of complexity in federation management as multiple relationships need to be governed and managed. In the composite schema, the overheads of management are significantly lower, but the response times of the request being executed may increase due to network latency.

Over provisioning was also observed with relationships between data centers were decided apriori. When decided apriori, reservation was done in a probabilistic manner and intelligence generated over a period of time on the requirements to further reduce response time for a request to be executed. However, it was seen that this also led to resource wastage and over provisioning for both schemas. However, depending upon the application being services, this is a contingency that will be catered for in a more dynamic manner in future.

The real life scenarios depicted during the course of this experiment provided enough evidence that an efficient Autonomic Cloud Broker is a reality and it can exist in a hybrid cloud environment with a high level of performance, when fine tuned. This assurance gives confidence to the authors to further experiment on the use cases and derive results to specific requirements in a real world scenario.

However, some aspects that would need to be further analyzed and catered for in a real life scenario and in future experiments are :

- There is a need for the handling of the contingency when there is no global knowledge of the state of the cloud resource availability in a decentralized cloud. In particular, it may happen that some of the brokers involved in the federation do not have a complete view of the system state (e.g. existing clouds, resources offered, etc.). In such a scenario these broker serviced data centers incur the risk of taking suboptimal choices during the matchmaking phase because of their limited knowledge about other clouds.
- Timely propagation of the partial state of the cloud to the central server in the case of a composite broker besides handling of misinterpreted requests.
- There is a need to design customized network topologies for efficient cloud interactions. As the present set of experiments used a structured network or a Distributed Hash Table analogy for the underlying network, it was evident that the network latencies were minimized. Queries amongst brokers had a deterministic execution time. However, in cases of a more dynamic cloud, this may not be the case and significant network latency issues as well as decision time delays may occur for both the decentralized as well as the composite schemas.

Resolution and analysis of these issues remain work for future for the authors and will be covered in subsequent work being done. An improved version of the broker using a methodology to remember the cloud usage in previous requests and utilizing the knowledge to provide better services for future implementation is also a work in progress for the authors.

V. CONCLUSIONS

The experimental research presented in this paper was done on a geographically distributed setup with individual clouds and data centers federating and volunteering to feed resources to achieve overall efficiency in the system. The research also brought to fore certain issues which distributed as well as centralized federated schema for cloud brokers bring to fore. The centralized and distributed federated schema was tested using seven data centers and under a deterministic execution time paradigm. This research clearly illustrates that there is a realistic scope for creation and implementation of real life, autonomic cloud brokers, that can use a variety of delivery schemes for efficient performance. Cloud bursting is a live issue and if enabled in an efficient manner, can be

handled by a public cloud. The same was demonstrated in this research when the AWS cloud infrastructure was integrated with the private cloud to handle peak loads. The results showcase that an autonomic broker is a reality and if implemented in a hybrid cloud can be used to derive a highly efficient network/cloud manager. The research is presently a work in progress as it has indicated aspects of cloud brokerage which would need further analysis and deliberation, especially those pertaining to initiation of a broker and issues of overall system knowledge management in case of misinterpreted requests. This would be taken up as a work for future and analysis on the same would be done either through a revised experimental setup or thorough simulations.

REFERENCES

- [1] P. Mell and T. Grance, "The NIST definition of Cloud Computing," NIST Special Publication, Washington, 2011.
- [2] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," Fourth International Conference on Internet and Web Applications and Services, Venice/Mestre, 2009.
- [3] Y. Demchenko, M. X. Makkes, R. J. Strijkers and C. de Laat, "Intercloud Architecture for Interoperability and Integration," CloudCom, 2012.
- [4] P. Khanna and S. Jain, "Cloud Broker: Application Brokering in a Federated Clouds," Shodh 2014, JKLU, Jaipur, Jun 2014.
- [5] R. Buyya, R. Ranjan and R. N. Calheiros, "InterCloud- Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services," 10th International Conference on Algorithms and Architectures for Parallel Processing, Busan, Korea, May 2010.
- [6] Z. Sanaei, S. Abolfazli, G. A. and R. Buyya, "Heterogeneity in mobile Cloud Computing: Taxonomy and Open Challenges," Communications Surveys & Tutorials, vol. 16, no. 1, pp. 369-392, 2014.
- [7] A. N. Toosi, N. C. Rodrigo and R. Buyya, "Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey," ACM Computing Surveys, NY, USA, 2014.
- [8] M. Hamze, N. Mbarek and O. Togni, "Autonomic Brokerage Service for an End-to-End Cloud Networking Service Level Agreement," IEEE 3rd Symposium on Network Cloud Computing and Applications (NCCA), Cambridge, 2014.
- [9] F. Fowley, P. Claus and Z. Li, "A Comparison Framework and Review of Service Brokerage Solutions for Cloud Architectures," in Service-Oriented Computing" ICSOC Workshops, 2013.
- [10] P. Khanna and S. Jain, "Cloud Brokerage: Autonomic Service Provisioning Challenges and Handling of Complex Architectures," in Complex Adaptive Systems 2014, Atlanta, 2014.
- [11] "Cloud Data Management Interface," SNIA, [Online]. Available: <http://www.snia.org/cdmi>. [Accessed 3 August 2014].
- [12] "GFD.183 Open Cloud Computing Interface - Core," [Online]. Available: <http://www.ogf.org/documents/GFD.183.pdf>. [Accessed 3 Aug 2014].
- [13] "Open Virtualization Format," DMTF, [Online]. Available: <http://www.dmtf.org/standards/ovf>. [Accessed 3 Aug 2014].
- [14] "New Relic," 10 Jun 2014. [Online]. Available: <http://newrelic.com/>.
- [15] "Loader," [Online]. Available: www.loader.io. [Accessed 10 Jul 2014].
- [16] P. Khanna and B.V. Babu, "Cloud Computing Brokering Service: A Trust Framework," Third International Conference on Cloud Computing, GRIDs, and Virtualization, Nice, 2012 doi:10.1109/FTDCS.2003.1204317.
- [17] A. Maros, G. Kecskemeti, A. Kertesz and P. Kacsuk, "FCM: an Architecture for Integrating IaaS Cloud Systems," Proceedings of the Second International Conference on Cloud Computing, GRIDs, and Virtualization, Rome, 2011.