VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DISTRIBUTED CLOUD BROKERAGE: SOLUTION TO REAL WORLD SERVICE PROVISIONING PROBLEMS

Prashant Khanna[1], Sonal Jain[1] and B. V. Babu[2]
[1]JK Lakshmipat University, Jaipur, India
[2]Galgotia University, Greater NOIDA, India
E-Mail: perukhan@gmail.com

ABSTRACT

This research analyzes the performance of a distributed cloud broker in a live cloud environment, utilizing a government owned, private, federated cloud. The paper explores functioning of a distributed cloud broker that assists in the provisioning of services to geographically distributed data centers. The data centers have volunteered to federate and expose their utilization metrics to each other through the cloud broker. The experimental infrastructure utilizes the closed schema for federation. The cloud broker is responsible for match making and bundling/provisioning of services from multiple private cloud providers, through volunteer and federated data centers. This has been tested under multiple load conditions. The proposed distributed cloud broker handles the load on the cloud ecosystem through a strictly controlled mechanism in a private cloud ecosystem, custom routing all overload conditions (cloud bursting scenario) on specific private clouds through a common interface visible to the Amazon Web Services (AWS). The proposed broker mechanism shows high efficiency and lesser cloud bursting instances compared to a pure AWS based ecosystem. The work also analyzes real world issues faced by organizations handling cloud brokerage frameworks in a distributed manner. The research asserts that it is possible to create customized distributed cloud brokers and the perspective of using a hybrid cloud approach using distributed broker in federated clouds is feasible, albeit in a tightly integrated and fine tuned cloud environment.

Keywords: distributed cloud brokerage, cloud federation, cloud broker, live analysis, cloud service, cloud provisioning.

1. INTRODUCTION

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 backed framework has been accepted in academia and business alike. The authors have used the same for presenting this study and discussing an ignored player in the NIST model - the cloud broker. With multiple agencies and business use cases necessitating arrival of multiple cloud offerings, the concept of Intercloud [2, 3] and Federated Cloud [4, 5] is finding increasing visibility. This has also led to highlighting of cloud inter-operability issues. Recent advances made in actual federation of clouds has put the role of a cloud broker as an agent that allows for dissimilar services to interact with each other. While several works in the recent past have highlighted frameworks that make cloud federation a possibility, live testing is not often seen in these papers. This research is aimed at providing a live example of the concept of cloud federation adopting a closed schema where pre-defined SLAs and maximum capacity knowledge is available apriori. Traditionally cloud brokerage and cloud federation are complimentary concepts, but in a limited sense, this research makes them work together to derive certain compelling business cases.

The rest of the paper is structured as under: In section II, we refer to the State of Art on current research in cloud federation. Section III describes, in general, the distributed cloud federated broker and its implication on the present research. Section IV discusses in detail the real life setup for the experiments and the test results. The section also presents the research findings. In section V we conclude and describe future work on the

2. STATE OF ART

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 aspect of brokering services to the end user is finding refereed status only recently [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. The traditional definition of the cloud broker requires a revision in the face of increasing adoption of federation in cloud ecosystems. This adoption has been necessitated due to a strong business case for converged service/data centers that talk and share offerings/services and make the entire ecosystem more efficient. It is however pertinent to note that cloud brokerage and federation are complimentary concepts, but there is a strong use case for allowing the existence of both these concepts in private clouds ecosystems where organizations can achieve highly optimized results while using a broker to manage, schedule and provision services/resources to requesting cloud data centers. Several theoretical works in the recent past have attempted to arrive at a universal cloud interaction mechanism that makes them seamless to messages traversals across clouds. Recent works on the subject include [9, 17] which pertain to interactions in a federated cloud infrastructure and the issues of broker management. A limitation of most of existing studies on cloud federation and broker implementation is that they are simulation oriented and actual broker creation and its implementation in the cloud has not been explored. This research is a remedial step in that direction.

Standardization of interfaces is necessary to realize interoperability objectives and autonomic cloud brokers could act as the repository of such standard interfaces. Federation of the clouds whether for storage, utility optimization or migration of services from once cloud to another provides a compelling business case and thus sees renewed research on the subject. The emergence of the cloud broker as the agent to do cloud resource matching and provisioning amongst cloud providers would be a natural evolution of the cloud broker. Recent work by Buyya [10] also cites the utility of such a use case.

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. Both these efforts 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, coupled by an efficient cloud service brokers are the way ahead. The present research was undertaken to understand and implement such a use case 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.

3. DISTRIBUTED FEDERATION IN THE CLOUD AND THE 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 according to three main phases, i.e. Discovery, Match Making of resources, and, Authentication.

Cloud federation presents a compelling use case as it can lead to Internet scale resource optimization opportunities, the ability for cloud users and providers to upscale and down scale efficiently depending upon the utilization metrics/ratio of the data centers, besides energy efficiency while arbitrating services. Presently there is a flux when we come to using established standards governing 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]. In this research the cloud broker acted as a universal interface that achieves transactional integrity and message parsing ability across these popular formats. The experimental setup for this was created across a federation which consisted of private data centers, and which can be emulated in other life scenarios and use cases. Cloud interoperability in real world setup requires cloud providers to adopt and implement standard interfaces, protocols, formats and architectural components that facilitate collaboration. This was achieved in the present research using a common cloud broker abstraction, implemented and hosted on multiple data centers, providing interconnect services. In this research the authors created a scenario for procuring the resources using a closed scope in the federated schema. The cloud instances were already known and a distributed instance of the cloud broker was utilized to derive the metrics from individual cloud centers and do match making of resources. The scenarios for the testing of the broker utilized one of the following:

- **Storage Space Scenario** - The cloud providing storage services to its clients runs out of its storage resources. In order to continue providing cloud-based storage service to its user and adhere to laid down SLAs in place, the cloud requests the federation for additional services and broker decided the allocation of the new resource request, using an identified interface, from other offering clouds.
- **Distributed Availability Scenario** - The cloud scenario was implemented across disparate geographical locations, each hosing a different hardware infrastructure, using virtualization best practices across Xen Hypervisor. When there was a need to deploy a distributed cloud-based service in different geographical locations, the cloud broker assisted by acquiring the resource and instructing the target location to accept the service request.
- **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 when it realizes that its datacenter is under-utilized at a given time.

While setting up the experiment for the research, multiple similar instances 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. The virtual machine (VM) template made use of the following metrics for mapping:

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

The metering and analysis engine for the experiment setup and the broker performance was

VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

implemented using the New Relic [14] utilities. The metrics for the performance were:

- Responsiveness 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 sample use cases and cloud user requests were generated using the automated tool 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. The broker made a request for the availability of VMs and queries the loading status of the data center. The broker collates all the requests in a Hash Table and sends the collated requests to a central premise for the mapping function to be executed. The user-request list contained the desired service needed in the form of a template to the geographically distributed broker. The broker first did a local mapping for availability within the data center and depending upon availability, decided to offer the return template. The transaction was broadcast to all brokers in the federation so as to update the resource availability tables. The authors in this research have restricted the role testing of cloud brokerage to the discovery and matchmaking phase needed for the creation of a federation of clouds. Of the four possible semantics of cloud brokerage in federated schemas, this research tests the Distributed Schema for Brokerage.

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 [16, 4, 17]. The distributed broker implemented and placed at the interface points in the cloud are depicted in Figure- below. The schema has been implemented in a manner that the broker is exposed to both the external environment towards user, as also towards the other clouds that act as super users asking for resources. The schematic illustration is as under:

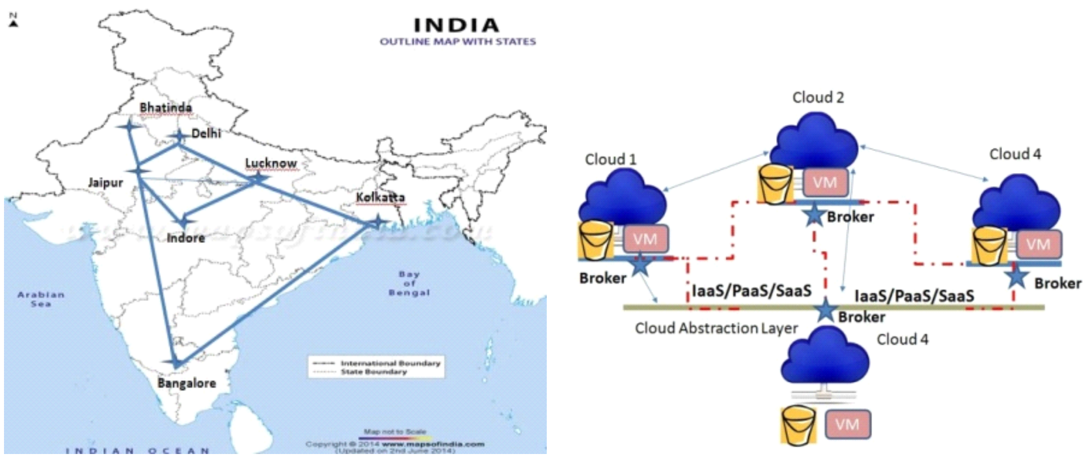

Figure-1. (a) Distributed Federated Broker Schema across Data Centers in India (b) The generic Distributed Brokerage Schema implemented for the experiment.

The federation of clouds in the experimental setup has been created such that individual clouds volunteer with the metrics of their performance and availability of resources in real time. This is done through the cloud broker which interprets the present requirement at the data center and provides information to the other data centers on the availability of excess resources or the need for more resources, as the case may be. This abstraction works in a distributed manner and polls the statistics from other connected or federated clouds to take a decision on the resource utilization.

4. EXPERIMENTAL STRUCTURE AND RESULTS

The authors used a private cloud ecosystem to host a distributed broker schema based on the Drupal 7 platform. The PHP library SimpleCloud was used to implement the Cloud paradigm on the Drupal Interface. In the case of the Distributed Federation Schema, the private data centers at seven locations across India (Jaipur, Delhi, Lucknow, Kolkata, Bhatinda, Bangalore and Indore) were used to create the cloud infrastructure. Kolkata was used as a limb to test externalities. There were three zones created for allocation preferences. Different use case scenarios depicted in the tables below indicate the different conditions for load tests generation. The base tests were conducted with the adoption, creation and migration of new VMs across the cloud and the usage of additional memory beyond the baseline available for standard 2GHz, Intel servers. The Xen Project version 4.3 was used as the Hypervisor.

The tests cases illustrated in Table-1 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 errors due to

ARPN Journal of Engineering and Applied Sciences

network latency and standard deviations have been taken as 1% on the results. Three types of jobs, singular, batch and periodical jobs were generated and pushed to the cloud brokers. The data centers are seamlessly talking to each other and offering metrics of their resource availability - thus enforcing a federated structure.

Table-1. Scenario of load test for brokerage schemas.

Scenario	Load test	Servers in use	Time duration	Type of request
Scenario 1	100 Searches on the Servers sent by all registered users	5 Concurrent servers at three locations	60 - 400 secs	Singular jobs
Scenario 2	60 Searches on the Server + 40 across servers sent by all registered users	5 Concurrent servers at two locations	60 - 400 secs	Batch jobs
Scenario 3	50 + 50 Search and swaps on the Server sent by all registered users	4 Concurrent servers at three locations	60- 400 secs	Periodical jobs (50% singular mixed)

The broker was tested to autonomically resolve the requirements and provision the services using available resources in the federation. 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 proposed brokers used FIFO as the preferred choice for resource allocation, but are also capable of a weighted mean assignment. The provisioning of the service was transparent to the client utilizing the service.

Table-2 depicts 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. The load was generated on live servers and the concurrent client requests were created on each of the seven data centers using the application of Loader.io.

Table-2. Results of distributed federated schema using multiple clients at distributed data centers for periodic jobs over 10 experiments.

No. of concurrent clients per sec.	CPU Avg (Web server) without broker enabled and standard deviation of 1%	CPU Avg (Web server) with broker enabled and standard deviation of 1%	CPU Avg (RDS) without broker enabled and standard deviation of 2%	CPU Avg (RDS) without broker enabled and standard deviation of 2%	Memory avg (Web 1)
1	16.07%	12.05%	32.5%	28.50%	57%
2	19.01%	13.35%	32.1%	31.70%	59%
3	21.03%	13.85%	33.2%	23.90%	60%
10	27.10%	17.87%	37.8%	24.10%	53%
30	39.09%	23.80%	42.4%	31.30%	73%
50	49.88%	30.90%	49.6%	39.10%	76%

The feature of a broker to initially appreciate the local load requirements at the data center in terms of storage, VMs or CPU load distribution and then deciding the rationale and amount of resources to offer to the others is critical to the efficient functioning of the distributed broker. The result is illustrated in Table-3 below as the resources available at the Kolkata data center in Table-3 were reserved for local use and then offered to the Lucknow data center when requested. While the data centers were provisioned adequately, several conditions of load were generated where it was seen that the broker was short of local resources and requisitioned additional storage and transfer of CPU loads to the federation. In the case of the Bhatinda Data Center, this is very clearly visible. The final cost and storage overruns at the seven data centers used during the experiment are also depicted in the table below.

Table-3. Comparison of provisioning in the seven data centers during the experiment.

Data center	VMs available	Storage available (GB/VM)	Cost overruns per instance	VM overshoot per instance	Storage overruns per instance	Cloud bursting without broker	Cloud bursting with broker
Jaipur	200	200	Nil	Nil	Nil	No	No
Bhatinda	15	20	$220	7	17	Yes	Yes
Bangalore	150	200	Nil	Nil	12	Yes	No
Kolkata	36	100	Nil	Nil	Nil	Yes	No
Indore	23	50	$35	1	5	Yes	Yes
Lucknow	11	50	$315	6	12	Yes	Yes
Delhi	55	50	Nil	Nil	Nil	No	No

Over provisioning was observed at the Jaipur and Kolkata data centers as relationships and communication channels between data centers were decided apriori. When decided apriori, reservation was done in a probabilistic manner and intelligence generated over a period of time for analyzing future requirements. This was done to reduce the response time for a request to be executed. However, it was seen that this also led to resource wastage and over provisioning in the data centers. It is anticipated that a more dynamic approach to resource management within the federation might lead to more optimization. The cloud bursting or apparent lack of resource matching was also lesser in the instances where the optimized broker was in place. The success of the experiments has also given the authors a great deal of confidence for developing more evolved brokers. 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, when fine tuned.

5. CONCLUSIONS

The present sets of experiments have been done in a geographically distributed manner with individual clouds federating and volunteering to join resources. The distributed federated cloud broker schema was tested using seven data centers and under a deterministic execution time paradigm. The AWS cloud service engine was integrated and used to derive the cloud bursting scenario, while the New Relic engine was used for generating the statistics on the tests. The results are indicative of a successful implementation of broker instances at multiple data centers. Cloud bursting is a live issue and if handled in an efficient manner, can be offset through 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 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] Mell P and Grance T. 2011. The NIST definition of cloud computing, NIST Special Publication, Washington.

[2] Bernstein D, Ludvigson E, Sankar K, Diamond S and Morrow M. 2009. Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability, Fourth International Conference on Internet and Web Applications and Services, Venice/Mestre.

[3] Demchenko Y, Makkes MX, Strijkers RJ and C. de Laat. 2012. Intercloud Architecture for interoperability and integration, CloudCom.

[4] Khanna P and Jain S. 2014. Cloud Broker: Application Brokering in a Federated Clouds, Shodh 2013, Jaipur.

[5] Buyya R, Ranjan R and Calheiros RN. 2010. InterCloud- Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services, 10th International Conference on Algorithms and Architectures for Parallel Processing.

[6] Sanaei Z, Abolfazli S, Buyya R. 2014. Heterogeneity in mobile cloud computing: taxonomy and open challenges, Communications Surveys and Tutorials, vol. 16, no. 1, pp. 369-392.

[7] Toosi AN, Rodrigo NC and Buyya R. 2014. Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey, ACM Computing Surveys.

[8] Hamze M, Mbarek N and Togni O. 2014. Autonomic Brokerage Service for an End-to-End Cloud

Networking Service Level Agreement, IEEE 3rd Symposium on Network Cloud Computing and Applications (NCCA), Cambridge.

[9] Fowley F, Claus P and Li Z. 2013. A comparison framework and review of service brokerage solutions for cloud architectures, Service-Oriented Computing-ICSOC 2013 Workshops.

[10] Rodriguez MA and Buyya R. 2014. Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds, IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 1-22.

[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 August, 2014].

[13] Open Virtualization Format, DMTF, [Online]. Available: http://www.dmtf.org/standards/ovf. [Accessed 3 August, 2014].

[14] New Relic, 10 Jun 2014. [Online]. Available: http://newrelic.com/.

[15] Loader.io [Online]. Available: www.loader.io. [Accessed 10 July, 2014].

[16] Khanna P and Babu BV. 2012. Cloud Computing Brokering Service: A Trust Framework, Third International Conference on Cloud Computing, GRIDs, and Virtualization, Nice.

[17] Khanna P, Jain S and Babu BV. 2014. Cloud Broker: Working in Federated Structures - A Hybrid Cloud: Live Performance Analysis, 3rd International Conference on Advances in Computing, Communication and Informatics, NOIDA.