

Availability and Load Balancing in Cloud Computing

Zenon Chaczko¹, Venkatesh Mahadevan², Shahrzad Aslanzadeh¹ and Christopher Mcdermid¹

^{1,3 & 4} University of Technology Sydney, Australia

² Swinburne University of Technology, Australia

Abstract. Availability of cloud systems is one of the main concerns of cloud computing. The term, availability of clouds, is mainly evaluated by ubiquity of information comparing with resource scaling. In clouds, load balancing, as a method, is applied across different data centers to ensure the network availability by minimizing use of computer hardware, software failures and mitigating recourse limitations. This work discusses the load balancing in cloud computing and then demonstrates a case study of system availability based on a typical Hospital Database Management solution.

Keywords: cloud computing, load balancing and security.

1. Introduction

Availability is a reoccurring and a growing concern in software intensive systems. Cloud systems services can be turned offline due to conservation, power outages or possible denial of service invasions. Fundamentally, its role is to determine the time that the system is up and running correctly; the length of time between failures and the length of time needed to resume operation after a failure. Availability needs to be analyzed through the use of presence information, forecasting usage patterns and dynamic resource scaling [1].

The aim of this paper is to demonstrate and discuss a critical role the load balancing of resources plays in improving and maintaining the availability in cloud systems.

Application of load balancing and redundant mirrored databases in clusters techniques, across multiple availability zones, reduces the chance of outages that could simultaneously affect the services in cloud systems. If an outage affected one system, the load balancer is able to switch to another available resource [4]. Load balancing techniques, in the area of cloud computing, reduces costs associated with document management systems and maximizes availability of resources reducing the amount of downtime that affect businesses during outages. This article discusses possible ways to improve the performance of cloud networks by the introduction of resource load balancing technique that uses the message-oriented middleware within the web service oriented model of software architecture.

2. Load Balancing

Load balancing is a relatively new technique that facilitates networks and resources by providing a maximum throughput with minimum response time [8]. Dividing the traffic between servers, data can be sent and received without major delay. Different kinds of algorithms are available that helps traffic loaded between available servers [2B]. A basic example of load balancing in our daily life can be related to websites. Without load balancing, users could experience delays, timeouts and possible long system responses. Load balancing solutions usually apply redundant servers which help a better distribution of the communication traffic so that the website availability is conclusively settled [8].

There are many different kinds of load balancing algorithms available, which can be categorized mainly into two groups. The following section will discuss these two main categories of load balancing algorithms.

2.1. Static Algorithms

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more imperfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic [5].

2.2. Dynamic Algorithms

Dynamic algorithms designated proper weights on servers and by searching in whole network a lightest server preferred to balance the traffic. However, selecting an appropriate server needed real time communication with the networks, which will lead to extra traffic added on system.

In comparison between these two algorithms, although round robin algorithms based on simple rule, more loads conceived on servers and thus imbalanced traffic discovered as a result [5]. However; dynamic algorithm predicated on query that can be made frequently on servers, but sometimes prevailed traffic will prevent these queries to be answered, and correspondingly more added overhead can be distinguished on network.

3. Load Balancing in Cloud Computing

Cloud vendors are based on automatic load balancing services, which allowed entities to increase the number of CPUs or memories for their resources to scale with the increased demands [6]. This service is optional and depends on the entity's business needs. Therefore load balancers served two important needs, primarily to promote availability of cloud resources and secondarily to promote performance.

According to the previous section Cloud computing will use the dynamic algorithm, which allows cloud entities to advertise their existence to presence servers and also provides a means of communication between interested parties. This solution has been implemented into the IETF's RFC3920 - Extensible Messaging and Presence Protocol abbreviated as XMPP [7].

4. Load Balancing in Distributed Systems

Today more modern software development methodologies are being used to enhance the usability of software embedded on compatible hardware in distributed networks [3]. Attaining this objective and improve software infrastructure, middleware have been applied to foster portability and distributed application component interpretability. Middleware characterized as network services and software components that permit scaling of application and networks [9]. Providing the simple and integrated distributed programming environment, middleware eased the task of designing and programming and managing the distributed applications. Different models and applications have been applied in middleware. Message oriented middleware is a model which is based on using messages. Web services are the popular architecture for developing distributed applications. Communication in network is performed via messaging and optimal resource allocations can run by this architecture in distributed networks. The next section will describe how message oriented middleware have been used in resource allocation and load balancing.

5. Load Balancing and Message Oriented Model

Clusters provide the chance to use distributed applications through different computers on networks. The issue related to clusters raised in network performance. If the workload in distributed network loaded in one computer it will cause the network to be slow [3]. To prevent this situation happening, resource management can be used as software metrics to distribute the traffic between stations in a way that can keep the network performance up. In this section web services as a message oriented middleware model in resource management, will be described.

XMPP is mainly being used in online instant messaging programs. The technology is open for real time communication between various parties; however the key is the application of its availability or presence.

A basic high-level overview of how XMPP works is provided below.

XMPP clients send presence information to XMPP presence servers and XML streams containing details of presence information of clients produced by these servers. XMPP clients can access these XML streams like a directory and receive presence information of other specific XMPP clients.

Presence data distributed in XML streams contained a presence state, addressing information and protocols [7]. Addressing information is normally in the form of IP address and port number or user and domain name. Using presence information in cloud computing supported the availability of cloud entities. XMPP servers handle the storage of presence information and processing of incoming and outgoing requests. Using a load balancer on top of an XMPP server allowed incoming requests to be prioritized and handled by a generic service. Clustered database utilizing a shared-everything architecture that provided a copy of the data on each database node is the example of this fact [10]. Consequently till now, load balancers have been efficient in providing relevant resources based on the resource's availability.

XMPP also served another common function, which is messaging. As mentioned above, resources advertised through an XMPP host as a means for communication. Exchanging messages is similar to SMTP where messages are sent to servers and servers deliver the messages to the intended recipients or the recipient's server.

Messaging in a common format allowed any resources to communicate with each other and exposed how their services can be used. If resources on another cloud network are not implemented by XMPP, there is usually a gateway that can translate XMPP to a foreign protocol, which will help the communication to be established [7].

Leveraging the use of XMPP allowed resources to be monitored, collecting metrics on contained within their presence information. Presence information is standard up to a point; however more information needs to be contained within to provide deeper metrics such as CPU utilization, memory utilization, dynamic scaling capable, and response time and network utilization. All these metrics provide a quality of service and availability view of a cloud resource.

6. Transaction Metrics and Activity Based Costing

In previous sections, comprehensive model based on message-oriented model was described in cloud computing systems.

Resource utilization in clouds can be discussed in context of activity based costing. Activity based costing is an accounting concept through which organizations recoup costs incurred throughout normal business activity cycle. These costs can be then traced back to projects and business groups. The same concept combined with traced analysis, which can be applied to resources in the cloud for a better understanding in the utilization of resources. A cloud resource, whether it is a system, an application or a service, would have some form of metrics collected either to analyze performance or for the purpose of costing [6].

Following is an example of Internet Data Transfer out based on Amazon Ec2.

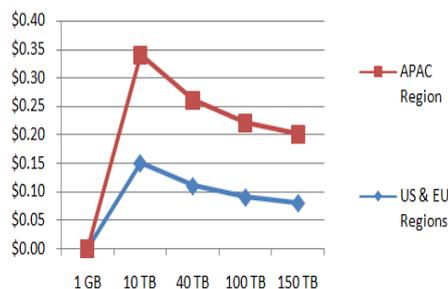


Fig. 1: Internet Data Transfer out based on Amazon Ec2

Each user or system interaction within cloud computing solution causes a resulting transaction to be carefully analyzed. Monitoring of transactions associated with individual and group entities can lead to better utilization of memory, an enhanced CPU usage as well as an improved utilization of the network interface

connecting to the entity that in the end would result in an increased availability and utilization of resource of the entire system.

Described here is the concept of trace analysis and leads to a complete picture of the total resource utilization broken down into its components. The size of increased metrics would be very minute and was only necessary for intensive transactions. However from a cost perspective, all utilized resources would have to be linked back to the related transaction, i.e. the amount of CPU cycles used at each point, as all costs totaled would equal a substantial amount. This mirrors cloud's utility based costing of cloud services at the transaction level. Currently only one proprietary tool, Causticity by JINSIPIRED, performs a costing of activities that monitors and meters various metrics related to service transactions of connection utilizations between entities. There is a large cost associated with the transfer of data that can seem hidden when reviewing the pricing structures of cloud vendors. The services within the Amazon EC2 Internet Data Transfer [11] can be given as a good example of such a case (Fig. 1). The first tier for US and EU regions is \$0.15 per GB. However when you have a cloud application that is data intensive like a document management system, the costs can begin to rise very steeply. Fulfilling the first tier brings a total cost of \$1500 USD in just one month, includes 10TB of data. This assumes that data is transferred between regions that are consistent with clustered and highly available document management systems.

A cloud monitoring should specifically monitor all transactions through all entities such that unusual transactions can be analyzed and reported. This would allow cloud monitoring services to scale resources effectively, thereby increasing availability where necessary and for the right reasons, instead of costing the resource owner large amounts of costs associated with scaling. Cloud monitoring is still a very new concept, and needed to be analyzed at the transaction level within all entities including resources and the connections between those resources. Cloud vendors have the necessary power to do this, however solutions provided don't cover the necessary depth needed for monitoring effectively. Thus the new way of monitoring will look into the "why" of utilization increasing instead of the "if", when scaling up a new cloud instance.

7. The Case Study of HDM Availability

The Hospital Data Management (HDM) system aims to increase the availability and efficiency of data retrieval from multiple databases in the context of cloud infrastructure.

Fig. 2 represents the HDM System as a separate entity to other databases, either in the same cloud or different clouds.

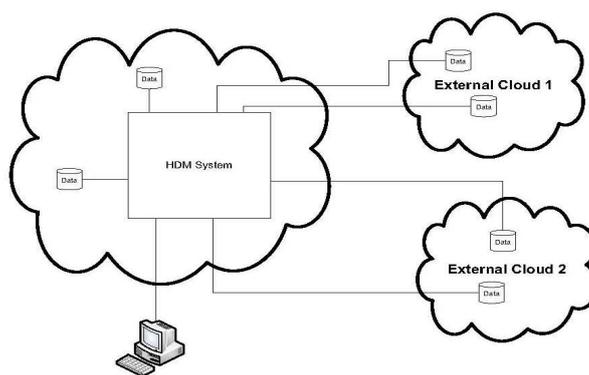


Fig. 2: HDM System

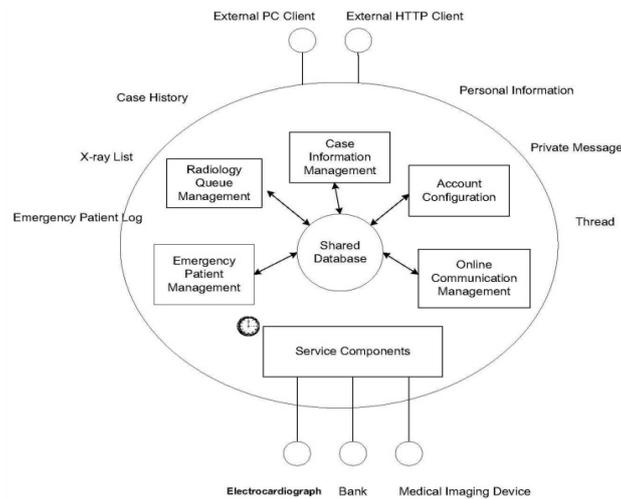


Fig. 3: Conceptual Architecture, Data Centric Model

The HDM system will be responsible for proactively scaling database resources by identifying usage patterns from past data to increase availability. Through the use of compression transfer sizes will decrease, which in turn decreases costs associated with resource usage. The HDM system only simulates data usage inside the hospital, however the presentation of this data will not be included in this system. Only metrics will be collected from the above simulation to measure the performance of the effectiveness of this system.

The HDM system scope includes the following:

- Collects usage patterns
- Forecasts future usage patterns
- Scales database requirements based on the future usage patterns incorporating some contingency
- Compresses data before storing/sending
- Uncompressed data after retrieving
- Collects metrics such as cpu usage (cpu hour/minute), data transfer in/out (MB) – used for cost benefit analysis

The Hospital Data Management (HDM) system will be the context of where the system shall be applicable. Every doctors and nurses will have access to the HDM system, used for retrieval of patient data, which will include various types of images such as x-ray, scans, audio and videos. Fig.3 showed the system in context.

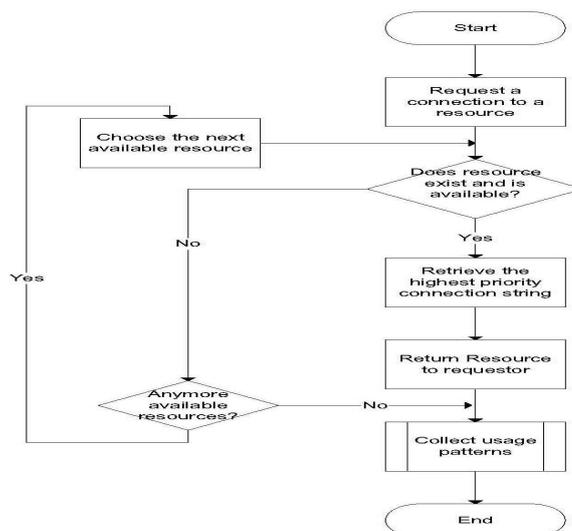


Fig. 4: Process for Load Balancing

The HDM system will simulate a cloud application, which will be used globally with data not stored locally, but retrieved from different global nodes. Thus the system must be able to access data from different databases situated on different nodes in an efficient manner. Methods of improving efficiency can be broken down into:

- Proactive scaling
- Compression

Planning the usage of pattern from past data, proactive scaling made specific popular resources available at their most common usage times. Accordingly faster access to a large set of data will appear. Compression has the advantage of minimizing the size of large files, which will lead to higher data transfer rates. The only disadvantage is the need to compress before transfer or storage, which requires memory.

In this system Resource Manager is planned as the load-balancer of the system, which can be thought of as a resource injector for finding a highly available resource to use. The process for load balancing is displayed in Fig. 4. Resource Manager maintains an array of resources, which can be requested on demand by users. The flow chart bellow shows the logic of Resource Manager when a user requests a Resource. The load-balancing package successfully chooses the highest resource based on the resource priority. Resources are continually monitored checking availability and performance. Based on these two factors, the resource table is updated calculating new priorities for resources that have significant changes in their availability and performance. High performing resources are favored and are shifted to a higher priority, whilst non-available resources are given a priority of 100, which means they will never become utilized.

Service Id	Service Type	Host	Port	Health	Response Time	Utilization %	Data rate mbps
65	MySQL	testServer4	3306	Green	10	0.23	5
23	IIS	testServer4	3066	Green	4	0.12	0.021
34	PostgreSQL	testServer4	5432	Red	400	0.7	1
35	Sybase ASE	testServer5	20003	Green	5	0	3
38	Sybase ASE	testServer5	20004	Green	5	0.02	3
40	MySQL	10.11.3.65	3306	Green	10	0.02	5
41	MySQL	10.11.3.65	3307	Green	10	0.1	5
42	MySQL	10.11.3.65	3308	Green	10	0.11	5
20	Sybase ASE	10.11.3.20	20003	Red	600	0.86	3
15	Sybase ASE	10.11.3.20	20004	Yellow	300	0.65	3
31	MySQL	testServer1	3306	Yellow	300	0.21	3

Fig. 5: Resource Manager GUI Table

The IP address traffic management could be incorporated into this project application, its function will be to determine the closest IP address to serve the requestor. IP address, availability and performance metrics could all be weighted with meaningful importance to calculate which resources should be sent to the requestor to use. The Resource Manager also provides a visual representation of all resources currently monitored. A snapshot of the Resource Manager's resource table is depicted in Fig. 5. Health is based on a combination of response time, utilization and data rate, which displays a color based on traffic light indicators. Green health announces a highly available resource, which red health signifies low availability. A white health (not shown in diagram), indicates that the resource is currently unavailable, which can be due to the resource being offline or not connectable. By utilizing the load balancer functionality of this project, the most available resources can be passed onto users, which they can continue to maximize efficiency through the use of their systems.

8. Conclusions

This paper demonstrated the applicability of using load balancing techniques to obtain measurable improvements in resource utilization and availability of cloud-computing environment. On that basis, the proposed approach could bring major cost advantages to cloud vendors who are concerned with utility costs and who are searching for efficiencies that can be relatively easily achieved. Various models and rules can be applied to load balancers, however these should be based on the scenario the load balancer will be applied for. The network structure or topology should be taken into account when creating the logical rules for the load balancer. This is due to the pricing of transfer between regions, availability zones and cloud vendors, which all constitute different pricing strategies. Message oriented architecture as a middleware model has been pointed out to improve load balancing in distributed networks. Based on messaging techniques XMPP allowed resources to be monitored and provide availability of cloud resources.

9. References

- [1] M. Armbrust, *et al.*, A view of cloud computing. *Commun. ACM.* vol. 53 (2010), pp. 50-58
- [2] A. Brian. Load Balancing in the Cloud: Tools, Tips, and Techniques. A Technical white paper in Solutions Architect, Right Scale.
- [3] R. L. Carter, *et al.*. Resource allocation in a distributed computing environment, in Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC, AIAA/IEEE/SAE, Vol. 1 (1998), pp. C32/1-C32/8.
- [4] D. Durkee. *Why Cloud Computing Will Never Be Free.* Queue, vol. 8 (2010), pp. 20-29
- [5] R. X. T. and X. F. Z.. A Load Balancing Strategy Based on the Combination of Static and Dynamic, in Database Technology and Applications (DBTA), 2010 2nd International Workshop (2010), pp. 1-4.
- [6] G. Reese. *Cloud Application Architectures.* O'Reilly Media, 1st Ed., Inc. Sebastopol, CA, US (2009).
- [7] IETF, 2010, Extensible Messaging and Presence Protocol (XMPP):CoreRFC3920,<http://datatracker.ietf.org/doc/rfc3920/>, viewed on 25th of Nov (2010)
- [8] R. Shimonski. *Windows 2000 & Windows Server 2003 Clustering and Load Balancing.* Emeryville. McGraw-Hill Professional Publishing, CA, USA (2003), p 2, 2003.
- [9] R. E. Schantz and D. C. Schidt. Middleware for Distributed System: Evolving The Common Structure for Network-centric Applications. *Encyclopedia of Software Engineering.* New York, Wiley & Sons (2001), pages 801-813
- [10] Y. Wu, *et al.*, A Load Balancing Strategy in Web Cluster System. Third International Conference on Natural Computation, ICNC 2007 (2007) pp. 809-813
- [11] Amazon, Amazon EC2 Pricing, Viewed on 25th of Nov. 2010, available at <http://aws.amazon.com/ec2/pricing/>, Amazon Web Services LLC (2010).