International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

An Extended User Centric Cloud Computing Architecture

Susheela Hooda[1], A.K.Sharma[2]

[1]M. Tech Student, B. S. Anangpuria Institute of Technology and Management, Faridabad, Haryana, India

[2]Dean- P.G. Research, B. S. Anangpuria Institute of Technology and Management, Faridabad, Haryana, India

Abstract: ***Cloud computing provides dynamically scalable and virtualized resources as a service over the network at minimum cost wherein data centers work as backbone, comprising of a large number of servers, networked to provide the computing and storage needs of the users. The existing work improves the efficiency of data-centers and optimizes the latency by using dynamic load balancing algorithms. In this paper, extended cloud computing architecture is being proposed which is based on computing needs of the users. Data -centers have been divided into sub data centers and based on the computing needs of the users, the request is diverted to a specific favorable sub data center at minimum response time.***

Keywords: Data-centers, Cloud computing, Master data- center, Slave data-center, Sub-Slave data- center, load balancing.

1. Introduction

Cloud computing provides computing resources to a prospective customer over Internet on Pay-as-you-go basis[2].It especially caters to the needs of business and scientific communities which are ready to pay for what they actually use without having to purchase expensive hardware such as servers, storage ,network equipments etc. In fact customers can use resources as and when required from anywhere at any time over the Internet. Moreover one chooses services from the pool of available services and negotiate price through the Service Level Agreement(SLA) from any of the popular cloud service provider such as Amazon[5],Google[7],Microsoft[8] etc. The cloud provides the following three major services such as:

a) **Software as a Service (SaaS):** In this type of service, clients can use ready customized applications and do not need to make changes such as customer resource management, videoconferencing, IT service management, Accounting, Web content management etc.
b) **Platform as a Service(PaaS):** In this type of service, resources that are required to build application are available completely over the Internet .Clients need not to download or installed these resources own his/her computer. PaaS provides services such as database integration, security, testing, development etc.
c) **Infrastructure as a service (IaaS):** In this type of service, clients can create a virtual processing environment by specifying choice of processing power, storage, network parameters etc. and also have control over operating system and application environment.

2. Related Work

Cloud architecture and computing is a potential area of research and it is still in its infancy stage. From the point of view of the some researchers, cloud computing as virtualization of previously existing data centers while some other considers data centers as backend resources. Zhenyu Fang et.[3] correlates cloud architecture with Business process management (BPM).They modeled platform layer and thereafter combined it with application layer. Almost in all cloud computing architectures, there is a common trend of centralized resources at the cloud provider's location. Rajkumar Buyya et.[2] provides the concept of Service-Level-Agreement(SLA) oriented resource allocation. Paton et.[4] proposed a strategy to manage complex and unpredictable workload over clouds. Cloud computing provides various kinds of services to remote users with diverse requirements. Dabas et.[9] proposed a cloud architecture that is based on RFID(radio frequency identification).It is a leading edge technology with some negative issues. Some of these issues are: limited computational capacity, poor resources and inefficient data management. Suraj Pandey[10] GIS is a tool that captures ,stores, analyzes, manages and presents data that are linked to geographical location. Rodrigo et. [11] cloud computing platforms provides the facility to the users to rent computing and storage resources on demand. Rajkumar Sharma[1] proposed cloud computing architecture that is based on master-slave paradigm wherein in this, an intelligent and energy efficient Cloud computing architecture is proposed. It is based on distributes data-centers which forms a client's instance in nearest neighborhood and fulfill client's request in optimized latency. Whenever a user's request for resources, this request is directly received by master data-center and then master data-center creates an instance for the user in the form of Virtual Machine (VM) .This request is then passes to the slave data-center that is nearest to the client's location. In this way, this architecture fulfills the user's request in more efficient way.

A critical look at the available literature indicates that there is no work has been reported about data- centers that support specific computing needs of the users. In this work the existing cloud computing's architecture has been extended by dividing the data-centers into sub-data-centers according to their computing capabilities. Each sub data center provides the specific needs to specific users. Hence forth, request is diverted to the best supporting data-center. Resulted in a user's request is being executed at more favorable destination with minimum response time.

3. Proposed Cloud Architecture

The number of clients on a cloud is increasing day by day and the large number of service requests generated there off lead to latency problems of the service provider end. The reason is being that cloud service providers are physically far away from the client's location and therefore service reaches to the client after a time delay. Some existing cloud providers uses centralized data-center to full fill the needs of the clients which broaden the latency problem .Recently a new architecture is proposed which is based on distributed data-centers. It optimizes the latency problem and also improves the efficiency of data-centers by using dynamic load balancing algorithm. But still there is need to reduce the response time of the user. In this work, an extension in existing cloud architecture is done by dividing the data-center into sub data-centers which provides the quick response to client at minimum latency time. The architecture is followed by an algorithm which forms a client's instance in nearest neighborhood and satisfies the client's request by diverting specific user's request to the specific favorable sub data-center.

3.1 Cloud Computing Model

In the proposed Cloud architecture, data centers are divided into three levels such as master data-center(MDC),slave data-centers(SDC) and sub-slave data-centers(SSDC).Master data center resides at the service provider's premises and remaining two (SDC and SSDC) are scattered in various geographical locations.

Figure1: Proposed Cloud Architecture with Master, Slave and Sub-Slave Data Centers

In existing cloud architectures, user perceived latency is the main problem. This paper, we attempt to optimize the latency problem by dividing data-centers into sub data-centers and each sub data-center caters to the specific request made by the user. Since, user's request is directly transferred to the specific data-center and the response time of the system is drastically reduced.

In the proposed Cloud architecture, the user requests to the Master data-center (MDC), for a potential service. The MDC in turn passes to the nearest Slave data-center (SDC). Thereafter, slave data-center (SDC) identifies the type of user's request and accordingly sends it to the specific sub slave data-center(SSDC).Since SSDC is designed to handle specific tasks, it is tailor make to caters the user's request thereby resolving the latency problem to some extent.

The main entities involved in proposed architecture, as shown in Figure 1, are as given below with brief description:

- **Master Data-Center (MDC):** Master Data-Center is located at cloud provider's premises. User's accounting on pay-as-you-go basis is completed here.
- **Slave Data-Center (SDC):** These are scattered in various geographical locations to serve the user's request. Here user's instance is created in the form of Virtual Machines (VMs).
- **Sub-Slave Data-Center (SSDC):** Slave data-centers are divided into sub-slave data-centers that serve the specific needs of the users. Some specialize data-centers are given below:-
- **Database Data-Center:-**Cloud based database data - center converts data that is coming from different sources into one standardized format. These are high performance data-centers capable of handling Data Warehouse (DW) and Online Transaction Processing (OLTP).One such type of cloud database is Oracle Exadata database Machine.
- **Graphics Data-Center:-**Cloud graphics provides various services such as Gaming as a service (GaaS), medical imagery photo, media editing, advance media delivery etc.
- **SQL Data-Center: -**SQL cloud is a MySQL database that is known as Google's cloud. It has all the capabilities and functionalities of MySQL. Google Cloud SQL is easy to use, doesn't require any software installation or maintenance, and is well suited to small and medium-sized applications.
- **Simple storage Data-Center:-**Cloud storage means "the storage of data online in Clouds" wherein a company's data is stored in and accessible from multiple distributed and connected resources that comprise a cloud. Cloud storage providers are Google, Reckspace, Mezeo, Amazon etc.

- **Computational Data-Center:-**High performance computing (HPC) allows scientists and engineers to solve complex science, engineering and business problems using applications that require high bandwidth, low latency in networking and high computing capabilities. Amazon Web Service (AWS) is one of the service providers of it.
- **Users/Brokers**:- Users can directly communicate or via brokers with the Master Data- Center.MDC creates user instance at appropriate SDC.Then SDC passes the user's request to the SSDC after identify the computing needs of the users. In this way, user's request full fill in optimized way and reduces the response time of the user.
- **Service Level Agreement (SLA):** Here clients settled pricing. Master data-center scans SLA each time to host needs of the users.

The algorithmic detail of the working of proposed architecture is given below:

```
Algorithm findSDC()
{
 SDCfound=0; //this is a flag to find whether SDC is found
or not.
 Request MDC to create an instance;
 MDC checks the availability of SDC at user's location;
 If SDC is available at user's location
 {
 Assign a Virtual Machine (VM) to the user;
 Identify the needs of the user and accordingly assign SSDC;
 Load_Balance();
 SDCfound=1;
 }
 else
 {
 Search the nearest SDC from the user;
 Assign a Virtual Machine (VM) to the user;
 Identify the needs of the user and accordingly assign SSDC;
 Load_Balance();
 SDCfound=1;
 }
 If (SDCfound==0) findSDC();

}
```

3.3 Proposed Algorithm for Assigning SSDC (Sub-slave data-center) to user

To maximize the performance of Cloud's application in distributed computing environment, it is necessary to bring a balance between the overloaded data-centers and the idle data-centers by distributing load across the network. In proposed algorithm, a load table is maintained by every SDC which containing the information about all of its sub-slave data-center's (SSDCs) load. Whenever SDC (slave data-center) receives the request, it locates available SSDCs as per the user's location and request's type. In fact, it consults its load table and if load value of SSDC is below the threshold value then it is assigned to the user otherwise the process is repeated for a suitable SSDC in nearest neighborhood from the user's location until it finds desired lightly loaded data-center.

This mechanism improves the efficiency of data-centers, which results in reducing better response time for the application and it also improves the efficiency of the resource utilization.

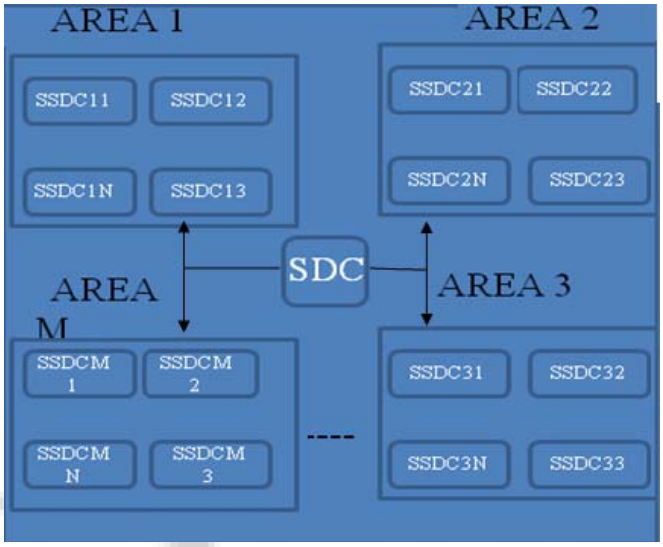

Figure 2: Create User's Instance based on Geographical Location

In fact, entire geographical locations are divided into m number of area locations i.e. Area1, Area2,--,AreaM where each area consists of number of SSDCs (sub-slave data-centers) as shown in figure 2.After receiving a user's request, SDC identifies the user's location and also looks for the availability of resources in user's local SSDCs.If desired resources are available at user's location then user gets required resources and his requested application is executed. If resources are not available at user's location then Slave data-center looks for other SSDCs of the same area i.e.SDC looks for resources in other geographical locations.

The algorithmic detail of load balancing in proposed architecture is given below:

```
Algorithm: Load_Balance(Area i)
{
 SSDCfound=0; // this is a flag which keeps track whether
SSDC is found or not.
 J=1;
 //Match the user's location and resource request with SSDC
 While (SSDCfound==0 && j<=m)
 {
 if (load (SSDC) <max_val) //Check the load of SSDC
 {
 SSDCfound=1; // if desired SSDC is found then
SSDCfound set to 1 and
 Assign SSDC to the user; // assigned to the user.
 exit ();
 }
 J++; // move to the next nearest area by incrementing in j
 }
 i=i+1;
 Load_Balance(i);
}
```

4. Conclusion

In this paper, a user-centric cloud computing architecture has been proposed which divides data-center into sub data-centers depending upon the computing capabilities of data-centers. It is an extension of the existing cloud architecture that could create client's instance in nearest neighborhood of the user which satisfies the specific client's request by diverted to the specific sub data-center .Additionally the issue of load balancing in distributed environment has also been addressed which reduces the number of migrations of user's request within a specific area and it reduces the network congestion by creating user's instance in nearest user's geographical location.

References

[1] Rajkumar Sharma and PriyeshKanungo "Vikram University", Ujjain, India rksujn@rediffmail.com," Patel College of Sc. & Technology", Indore, India,sept 2011.
[2] RajkumarBuyya, Chee Shin Yeo, and SrikumarVenugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC 2008), Dalian, China,* Sept. 25-27, 2008.
[3] Zhenyu Fang and Changqing Yin, "BPM Architecture Design Based on Cloud Computing," *Online Journal on Intelligent Information Management*, Vol 2, May 2010, pp 329-333.
[4] Norman W. Paton, Marcelo de Aragao, Kevin Lee, Alvaro Fernandes and RizosSakellariou, "Optimizing Utility in Cloud Computing through Autonomic Workload Execution," retrieved from http://research.microsoft.com/pub/debull/A09mar.
[5] Amazon Elastic Compute Cloud (EC2), http://www.amazon.com/ec2/
[6] S.MadhavaReddy,E.Mrithyunjaya and j.Srikanth ,"cloud computing architecture supporting e-governance",in IJAR,vol-2,issue 8,august 2012.
[7] Google App Engine, http://appengine.google.com/
[8] Microsoft Live Mesh, http://www.mesh.com/
[9] ChetnaDabas and J.P Gupta, "A Cloud Computing Architecture Framework for Scalable RFID," *Proceeding of the International Multi Conference of Engineers and Computer Scientsts (IMECS 2010,) Hong Kong*, Vol 1, March 2010, pp 217-220.
[10] SurajPandey, "Cloud Computing Technology & GIS Applications," *Asian Symposium on Geographic Information Systems From Computer & Engineering View (ASGIS 2010), Chine,* April 2010
[11] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and RajkumarBuyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Volume 41, Number 1, Pages: 23-50, New York, USA, January, 2011.