www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 4 April, 2013 Page No. 1007 -1011

A Novel Cloud Computing Architecture Supporting E-Governance

[1]M.Shahul Hameed, [2]A.Appunraj, [3]DR.T.Nalini

[1,2]Student
[3]Professor
Department of Computer Science & Engineering
Bharath University

Abstract

In this paper Slave Data center and Master Data Center is a technique used to spread a network service workload between two or more devices. Benefits include scalability, reliability, efficiency, redundancy, and minimized response time. The large number of service requests to fulfill the demands of millions of users will broaden the latency problem. Cloud service provider physically may be far away from the clients, compelling data to travel from several mediums and network equipments. So in this paper we are providing the an intelligent & energy efficient Cloud computing architecture is proposed based on distributed data-centers (DDC) which form a client's instance in nearest neighborhood and fulfill client's request in optimized latency.

KEY WORDS: Cloud computing, Master data center, Slave data center, Latency, DDC (distributed data center), Virtual Machine

INTRODUCTION:

Computer scientists have always been attempting and innovating a new technology that efficiently & effectively utilizes the contemporary underlying hardware resources for the benefit of the science and business community. Starting from mainframes to recent virtual machines on 'Clouds', computational history experienced a trend of alternatively convergent and divergent patterns for the use of computing resources. Mainframe/Mini Computers processed users programs centrally on time sharing concept. The deep penetration of cheap Personal Computers affected almost every corner of computing thus diverging the resources. In this paper we minimize the client request and response time in efficient manner in an intelligent & energy efficient Cloud. In this case we are maintaining Informal Description of Algorithm and Formal Description of Algorithm. Which are used to minimize the client request and response time in cloud scattered to serve user's requests in minimum physical distance in cloud environment.

RELATEDWORK:

Continuous research work is being carried out worldwide to address the issue of Cloud computing architecture. Some researchers identify Cloud computing as virtualization of previously existing datacenters while some others nominate data-centers as backend resources of newly adopted Cloud computing paradigm. Rajkumar Sharma, Priyesh Kanungo [1] correlate Cloud architecture with Business Process Management (BPM). Business process activities are modeled over platform layer and combined with application layer .A common trend of centralized resources at the Cloud provider's location is present in almost all existing Cloud computing architectures leading to increase in latencies.

[2]This keynote paper: presents a 21st century vision of computing; identifies various computing paradigms promising to deliver the vision of computing utilities; defines Cloud computing and provides the architecture for creating market-oriented Clouds by leveraging technologies such as VMs; provides thoughts on market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain SLA oriented resource allocation.[4]Cloud computing provides services to potentially numerous remote users with diverse requirements. Although Predictable performance can be obtained through the provision of carefully delimited services, it is straightforward to identify applications in which a cloud might usefully host services that support the composition of more primitive analysis services or the evaluation of complex data analysis requests. [3]While cloud-based BPM (Business Process Management) shows potentials of inherent scalability and expenditure reduction, such issues as user autonomy, privacy protection and efficiency have popped up as major concerns. Users may have their own rudimentary or even full-fledged BPM systems, which may be embodied by local EAI systems, at their end, but still intend to make use of cloud-side infrastructure services and BPM capabilities, which may appear as PaaS (Platform-as-a-Service) services, at the same time. A whole business process may contain a number of non-compute-intensive activities, for which cloud computing is over-provision.[5]RFID is a leading edge technology with some bleed points. Some of the issues in RFID are preventing it from gaining everyday acceptance. Such performance issues associated with RFID systems are limited computational capacity, poor resources and inefficient data management. Hence there is a demanding urge to address these issues in the light of some mechanism which can make the technology excel. [10] Geographical Information Systems or Geospatial Information Systems (GIS) is a collection of tools that captures, stores, analyzes, manages, and presents data that are linked to geographical locations. GIS plays an essential role in wide range of areas and is extensively adopted nowadays. In the simplest terms, GIS is the merging of cartography, statistical analysis, hardware, software and data. GIS is commonly used as a supporting system for making best possible decisions through spatial and non-spatial data relations, visualization and processing. GIS is beneficial and works well when made available to as many people as possible everywhere and anytime at the expense of very less resources in terms of technology and expenditure. Over a few decades efforts are being made to upgrade the conventional GIS applications in order to provide broad spectrum services to the users across the globe. 'Cloud computing', a term which has become popular in recent years, has been described as "the next natural step in the evolution of on-demand information technology services and products". Cloud Computing can be applied to solve and overcome the challenges in GIS applications.[9]Cloud computing platforms enable users to rent computing and storage resources on-demand to run their networked applications and employ virtualization to multiplex virtual servers belonging to different customers on a shared set of servers. In this paper, we empirically evaluate the efficacy of cloud platforms for running latency-sensitive multimedia applications. Since multiple virtual machines running disparate applications from independent users may share a physical server, our study focuses on whether dynamically varying background load from such applications can interfere with the performance seen by latency-sensitive tasks. We first conduct a series of experiments on Amazon's EC2 system to quantify the CPU, disk, and network jitter and throughput fluctuations seen over a period of several days. Our results reveal that the jitter and the throughput seen by a latency-sensitive application can indeed degrade due to background load from other virtual machines.

E-GOVERNANCE:

E-Governance is use of IT infrastructure to ease governance activity such as administration, revenue services, various services to citizens, policy formation etc. e-Governance improves the efficiency of government functioning by removing redundancy at different levels. citizens get advantage from several eservices like income tax, pension, services related to municipal corporation and agriculture etc. The four main categories of e-Governance applications and the Services fall under these categories are :

1. Government to Government (G2G): Administration, Policy formation etc.

2. Government to Business (G2B): Taxation, Tender etc.

3. Government to Consumer (G2C):Land record, Birth certificate etc.

4. Government to Employees (G2E): Income tax, Pension etc. Transforming ongoing national e-Governance plan of governments to Cloud architecture would yield the following benefits:

5. Uniform e-Governance architecture all over the country as against the heterogeneous

Architecture due to procurement of IT infrastructure from autonomous government Agencies.

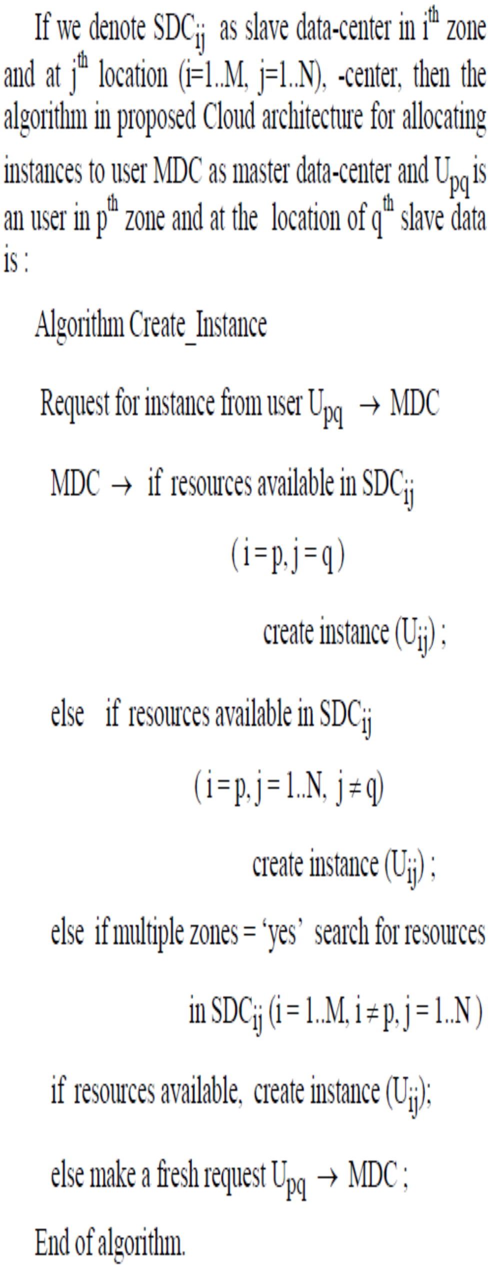
6. Governments need not have expensive IT Setups, pay for software licenses and maintain them leading to a substantial cut in the government's annual budget for IT infrastructure.

7. Governments can concentrate on making e-Governance convenient for the intended users Instead of mere focus on technical and operational overheads to maintain IT Infrastructure.

INFORMAL DESCRIPTION OF ALGORITHM:

An algorithm is any well- defined computational procedure that takes some values, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into output. In this case first client will send the request to server then the server will receive the request from the client. After receiving a request for creating an instance, master data-center look for the availability of resources in the user's local data-center. If desired resources are available, then user gets his required instance and run his application with minimum latency. Master Datacenter searches other slave data-centers of same zone for resources if they are not available on the location of the user. If resources are not available even in same zone and user has opted for multiple zones then master data-center looks for resources in other zones.

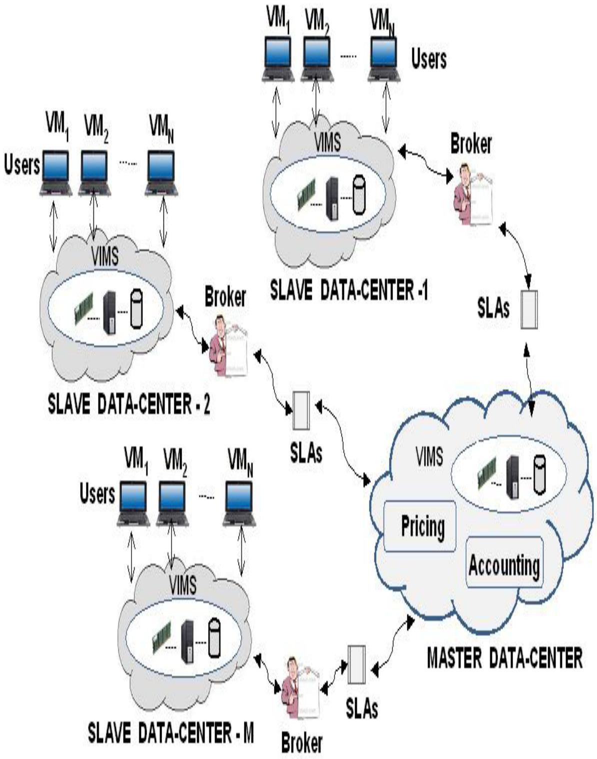
FORMAL DESCRIPTION OF ALGORITHM:



If we denote SDC_{ij} as slave data-center in i^{th} zone and at j^{th} location (i=1..M, j=1..N), -center, then the algorithm in proposed Cloud architecture for allocating instances to user MDC as master data-center and U_{pq} is an user in p^{th} zone and at the location of q^{th} slave data is :

```
Algorithm Create_Instance
 Request for instance from user $U_{pq}$ → MDC
  MDC → if resources available in $SDC_{ij}$
                ( $i = p, j = q$ )
                      create instance ($U_{ij}$) ;
  else  if resources available in $SDC_{ij}$
               ( $i = p, j = 1..N, j \neq q$ )
                      create instance ($U_{ij}$) ;
  else  if multiple zones = 'yes'  search for resources
               in $SDC_{ij}$ ( $i = 1..M, i \neq p, j = 1..N$ )
  if resources available, create instance ($U_{ij}$);
  else make a fresh request $U_{pq}$ → MDC ;
End of algorithm.
```

SYSTEM ARCHITECTURE DIAGRAM:



MODULES:

- Clients and Cloud Server
- Client Request Processing
- MDC and SDC Communication
- Broker Interaction
- Cloud Server Response

CLIENTS AND CLOUD SERVER:

In this module we create a user page using GUI, which will be the media to Connect user with the cloud and through which client can able to give request to the cloud and cloud server can send the response to the client, through this module we can establish the communication between client and cloud.

In this page user able to know about the overview of the whole application and having some link to get better knowledge about the whole application. Then In this case the cloud server receives request from the client and it will get the data from the data base depending on client request finally the cloud server send the response to client.

CLIENT REQUEST PROCESSING:

In this module after registration the client will be valid client then client will send the request to server. The client request like e-Governance components like online tax payment and income certificate, business to business process. After receiving request from the client then Cloud server(Master Data Center) will match the nearest Virtualization slave data center(SDC) then SDC(Slave Data center) will receive the clients request. Then SDC (Slave Data Center) will response to client. The request and response between the client and MDC (Master Data Center) the process will be perform as Virtualization way the process will perform between services Level Agreement (SLA).

MDC AND SDC COMMUNICATION:

In this module after sending the request from the client MDC (Master Data Center) will get the request then MDC Master Data center is located at Cloud provider’s administrative premises it perform like clients pay and go process. If client paid for the required process MDC will precedes the request. Then MDC will search the clients nearest Slave Data Center. After the finding nearest SDC. MDC will transmit the clients request to SDC. After receiving the request from the MDC the SDC will performs as Virtualization process. Then SDC will response to the client request.

BROKER INTERACTION:

In this module the initially the client will send the request the initial request the broker will receive the initial request from the client. After the receiving the client request the broker will automatically forwards the request to MDC (Master Data Center). After that Broker will performs the pay process. The pay process will perform between SDC (slave Data Center) and Broker. SDC will forward the pay process to Broker then the Broker ill forward to SLA (Services level Agreement) finally MDC will get the money from the client.

CLOUD SERVER RESPONSE:

In this module after sending the request from the client MDC (Master Data Center) will get the request then MDC Master Data center will search the nearest zone after finding the nearest zone. The MDC (Master Data Center) will search the nearest SDC (Slave Data Center) After finding the SDC the request will transmit to SDC then SDC will sends the response to regarding client. If zone is not able measure means the MDC (master Data Center) will search the nearest SDC after that the nearest SDC will response to regarding client.

EXPERIMENTAL AND RESULTS:

Experiments were carried out of the proposed architecture on Cloud simulator [8] which has provision for forming different data centers, virtual machine instance migration, energy on assumption model etc. By running an application initially from central data center and then from several geographic locations on different number of data centers, User's instances created as virtual machines are migrated across physical servers of data center for load balancing purposes. Less number of virtual machine migrations were reported as compare to central data center as computing resources are distributed across geographic locations. Maximum migrations occurred within zone of user's locality than inter-zone migrations. Barker & Shenoy show data of read/write experiments for a latency sensitive multimedia application running on a single data center. We simulated a multimedia application running on platform consists of distributed data centers which outperforms about 12% the results Data center usually consists of computing, storage & network devices almost in thousands of numbers. In addition to huge amount of electricity consumed by these devices, huge electric power is also needed for cooling systems. As a result data center emits more carbon dioxide (CO2) in a locality than permissible emission standard of the government. Cloud architectures proposed by several authors including [3] and [4] are based on central data center. On the other hand, in the proposed distributed data centers carbon emission would be within permissible limit as the resources are scattered in wide geographical area.

CONCLUSION:

We proposed innovative approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. Our approach allows the data can easily retrieve from the cloud server with minimum latency. Our approach creates an energy efficient & eco-friendly green to the client. Client will get effective and efficient quick response from the cloud server.

REFERENCE OR BIBLIOGRAPHY:

[1] Rajkumar Sharma and Priyesh Kanungo
"Vikram University", Ujjain, India rksujn@rediffmail.com," Patel College of Sc. & Technology", Indore, India,sept 2011.

[2]Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal, "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 Rizos Sakellariou, "Optimizing Utility in Cloud Computing through Autonomic Workload Execution," retrieved from *http://research.microsoft.com/pub/debull/A09mar*.

[5] Chetna Dabas 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.

[6] Amazon Elastic Compute Cloud (EC2), *http://www.amazon.com/ec2/*

[7] Google App Engine, *http://appengine.google.com/*

[8] Microsoft Live Mesh, *http://www.mesh.com/*

[9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, 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.

[10] Suraj Pandey, "Cloud Computing Technology & GIS Applications," *Asian Symposium on Geographic Information Systems From Computer & Engineering View (ASGIS 2010), Chine*, April 2010.