NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696
VOLUME 3, ISSUE 1, JAN.-2016

INTEGRATION OF DATABASES WITH CLOUD ENVIRONMENT

Miss Anuradha Gayake
ME Computer, Computer Engineering Department, AVCOE,Sangamner,
Prof. R.L.Paikrao
ME Computer, Computer Engineering Department, AVCOE,Sangamner

ABSTRACT

Cloud is on-demand service, as it provides resources, services as well as required information for clients whenever they want or whenever they request for it. With lots of benefits cloud has some challenges like load balancing in which workload is divided into various servers or virtual machines. Each client need to be served in least expected time. Virtual machines do not saved the state of previous task therefore, required external scheduling of multiple requested resources. In this paper we are multiple tenant handling at cloud end for handling multiple tenants requests at a time to improve the performance and reduce the response time. As cloud is capable to handle multiple requests at a time end user can send upcoming request without waiting for the response of previous request. A scheduling mechanism is proposed at client end using Round robin scheduling algorithm. It will helps to efficient utilization of processor as well as resources of cloud. RR gives assurance of getting services from cloud server to clients with least waiting time. The system performance is evaluated with different environmental conditions and with different data sizes.

Keywords- Virtual Machine, Cloud Sim, Datacenter, VM Load Balancer, Round Robin Virtual Machine Load Balancing Algorithms.

INTRODUCTION

Recently, there is growth in popularity in applications of cloud. There is success in enterprise applications to cloud environments, it also increases force to combine enterprise applications in both on-premise & services in the clouds with other distributed services on clouds. CEAI is the cloud enterprise application combined to bridge the gap between services of cloud and un-precise software. CEAI can work with the properties like, multi-tenancy, cloud level scalability as well as environmental departure helps to differentiate applications those are cloud-based from traditional Enterprise Application Integration (EAI) environments [1]. There is an important component in service-oriented middleware [3], known as ESB i.e. Enterprise Service Bus [2]. It helps the above CEAI properties for bridging gap between the cloud services and un-precise software. In cloud storage, multiple clients or users requests for services as required. Therefore, cloud offers large scale resources for organizations whenever they want. As we, aware of cloud scalability, user/clients have to wait for gaining the required services or from gaining the response from cloud server. Existed systems were implemented on FCFS, SJF or least connection loaded algorithm approaches. These approaches are not much time efficient as user having to wait for the response from server. And hence it results into less utilization of processor. In this paper, we are proposing Round Robine Algorithm [14]. According to our study analysis, it is very active processing algorithm to carry load balancing in cloud system [15]. It allocates resources in cyclic manner with equal priority to all nodes that are available. Round Robine Algorithm maintains a queue for storing incoming requests and then it allocates required resources for clients in scheduling manner [15], for give time period or time quantum. Our approach work well time efficient manner than FCFS or SJB algorithm. It will result into maximum utilizations of CPU as well as resources.

RELATED WORK

Jianwei Yin, Xingjian Lu, Calton Pu, Zhaohui Wu, Hanwei Chen [1], mainly focused on cloud services in SAAS model. It is used for integrating enterprise applications in the cloud with the other distributed services on clouds. CEAI is the cloud enterprise application integration that bridges gap between services of cloud and on-premises software. CEAI have properties such as, multi-tenancy, cloud-level scalability and environmental heterogeneity. These properties help for distinguishing applications that are cloud-based from traditional Enterprise Application Integration (EAI) environments. Cloud Service Bus (CSB) used to support the properties of CEAI and it also helps to bridge the gap between services of cloud and on-premises software. Basically, this system implements JTangCSB, for integration of an efficient as well as cost effective service to gain CEAI. Evaluation of JTangCSB represents an effectiveness and feasibility of CSB approach.

Strauch, Steve, et al[2] and Vaquero, Luis M, et al. [3] , focus on enabling multi-tenancy for one of the important components in service-oriented middleware, i.e. Enterprise Service Bus (ESB). Authors discussed about the prototype realization of a multi-tenant aware ESB by using an open source solution. They represent the scalability of application that is required to manage applications in a holistic manner. They also deal with NaaS.

Georgantas, Nikolaos, et al [4] [5], introduced system integration solution. It is based on orchestration workflow & a high-level data-driven coordination abstraction. It enables application workflows that are agnostic to the underlying middleware platforms. Orchestration workflow preserves native CM's difficulties with integrating heterogeneous systems. The proposed CS, PS, TS, and further GDS abstractions as well as related trade-off is between generic programming interfaces that offers simplicity and outperforms the loss of platform-specific features.

Shen, Zhiming, et al [6], describes Cloud Scale system. This system automates fine-grained elastic resource scaling in multi-tenant cloud computing infrastructures. Cloud Scale system consist of three key components such as, combining online resource demand prediction and efficient prediction error handling to meet application SLOs with minimum resource cost, supporting multi-VM concurrent scaling with conflict prediction and predicted migration to resolve scaling conflicts with minimum SLO impact and combining VM resource go up with dynamic voltage and frequency scaling (DVFS) to save energy without affecting application SLOs.

Ni, Jiacai, et al [7] and Pippal S K, Kushwaha D S. [8], studied the problem of adaptive multi-tenant database schema design. In this they identified important attributes, as well as common attributes, star attributes and dependent attributes. Using important attributes they build several dense base tables. Pippal S K, Kushwaha D S. implements the Multi-tenant database for an ad hoc cloud as it is fit to verify SaaS cloud services delivered between multiple clients. Ad hoc cloud derives the data as well as services from fixed cloud then they are connected using ad-hoc link. Multi-tenancy has three approaches.

Nusrat Pasha, Dr. Amit Agarwal and Dr. Ravi Rastogi [9], introduced VM Load balancing algorithm called as, “Round Robin Load Balancing Algorithm”. This algorithm helps to handle service request from user base. Load Balancing is a way to distribute workload on the different computers or a computer cluster via network links. It achieves optimal resource utilization for maximizing throughput and minimizing overall response time. In this paper authors proposes Round Robin VM Load Balancing algorithm to maintain two data structures such as, Hash Map and VM State List. Different numbers of virtual machines are analyzed with the help of Round Robin VM Load Balancing algorithm.

Amandeep Kaur Sidhu , Supriya Kinger[10] & Prof.Meenakshi Sharma, Pankaj Sharma[11] ,represents an issues regarding to cloud computing. Load balancing is the well problem; it can be classified into three categories as, Centralized approach: in this a single node is responsible for managing the distribution within the whole system. Distributed approach: in this, each node individually builds its own load vector by collecting the load information of other nodes. Prof. Meenakshi Sharma, Pankaj Sharma study of various virtual machine load balancing algorithms in cloud computing. They proposed VM load balancing algorithm that is implemented in CloudSim.

Pooja Samal, Pranati Mishra[12]and Ajay Gulati , Ranjeev.K.Chopra[13], studies effect of Round robin technique with dynamic approach by alternating the vital parameters of host such as, bandwidth, cloudlet long length, VM image size and VM bandwidth. By analyzing RR algorithm resource utilization and job response time is improved in the load distribution problem on various nodes of a distributed system. In this approach overloading and under loading situations are avoided.

Randles, M Lamb and Taleb Bendiab[14], discussed various load balancing schemes. They described that static load balancing scheme provide easiest simulation as well as monitoring of environment. But it is fail to model heterogeneous nature of cloud. Whereas, dynamic load balancing algorithm is difficult to simulate, it requires higher degree of replication.

Stuti Dave and Prashant Maheta[15], utilizes the concept of round robin algorithm for load balancing. With RR-algorithm Load balancer allocates a VM to requesting node in cyclic manner equally to all available nodes. RR algorithm simply maintains a queue of incoming requests and allocates them VM in Time scheduling manner. Each request is executed for specific time slice. It is an active monitoring algorithm. They proved that, round robin algorithm is faster processing algorithm on incoming cloud requests that resulting in faster load balancing. It only keeps in mind that, request size and the effect of number of rounds of algorithm.

PROPOSED SYSTEM

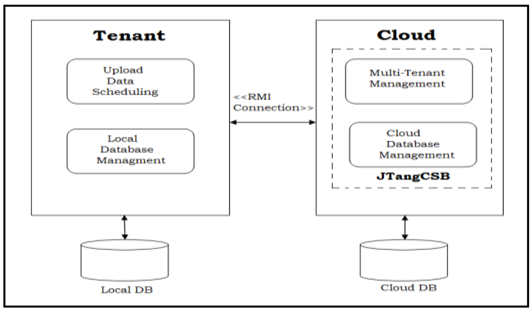


Figure 1: System Architecture

The proposed system works under following four sections:

A] Registration (AOW)

B] Upload Database

C] Multi-Tenant Handling

D] Scheduling Client Request

A] Setup: In this phase, the system is initialized. The Tenant will register itself on cloud and cloud will give access ownership to them. The tenant can access restricted area of the cloud server.

B] Upload Database: In this section, User/tenant will import selected Dataset or Whole Databases present on local machine/ server. Dataset may be large enough to transfer out to cloud server. Client system will automatically read the database details like tables and its constraints, relationship among multiple tables and its data.

C] Multi-Tenant Handling: this is a load balancing service at the cloud end. To handle multi-tenants instance-intensive workflows at the cloud end we propose a Time-Saving-Degree and cost optimization (TSD-Cost) algorithm. This algorithm mainly focuses on increasing users. It minimizes the response time for every user and increases the throughput of system.

D] Scheduling: Multiple tenants will be scheduled at server end. To schedule data upload request at client end we have used dynamic round robin algorithm. RR algorithm works on time sharing manner in continuous looping manner. It have specific time slice or time quantum that is applicable for each upload request with equal priority. Hence, client can initiate next upload request irrespective of response received from first request.

ALGORITHMS

1. TSD-Cost algorithm

Input:

I: Set of workflow instances sets of each tenant

Output:

TaskQueue[] : Array of sorted task queues.

-Group and sort ready tasks

- Sub-step1 Group ready tasks

Processing:

1. TaskQueue [] ← φ;
2. For each Ii ∈ I do
3. For each Ii j ∈ Ii do
4. While ∃Ti jk ∈ Ii j is ready
5. ReadyTasks ← Ti jk 6.for each task Ti jk in ReadyTasks do
7. T Gk ← Ti jk; Ti jk requires resource R h k ;sub-step2 Sort ready tasks
8. For each task group T Gk do
9. While ∃Ti jk ∈ T Gk do
10. Insert task Ti jk into queue TaskQueue[k] according to the strategy introduced above
11. Return array of sorted task queues TaskQueue[]

2. Round robin algorithm:

This algorithm is used to upload dataset to the cloud server using dynamic round robin algorithm.

Input: Number of database

Output: data uploaded to the server

Processing:

```
        -Initialize all the DB uploads status to AVAILABLE in the db state list
         -Initialize hash map with no entries;
While (new request are received by the Data Centre Controller)
   Do
{
   Data Center Controller queue the requests;
    Data Centre Controller removes a request from the beginning of    the queue;
    If (hash map contain any entry of db corresponding to the current requesting user base && db uploading
status == AVAILABLE)
   {
      DB is uploaded to the user base request;
    }
   Else {
Upload db to the user base request using Round Robin Algorithm;
Update the entry of the user base and the db in the hash map and the db state list;
         }
 }
```

MATHEMATICAL MODEL

S = {I, O, F}

I = {I1, I2, I3, I4} = Set of Inputs, where,

I1 = user details

I2 = user databases

I3= user role

I4 = Selected Databases

O = {O1, O2, O3} = Set of Outputs, where,

O1= user authentication

O2 = User account creation

O3 = Database on cloud

F= {F1, F2, F3, F4, F5, F6, F7} = Set of Functions, where,

F1 = Register user details

F2 = Login

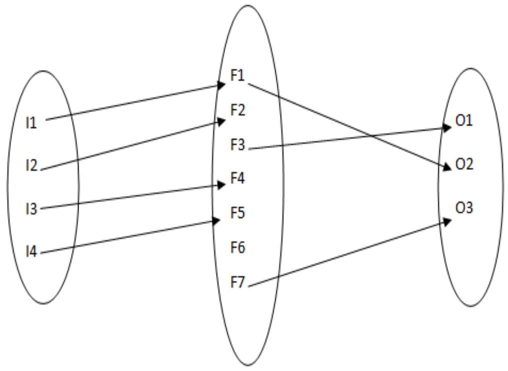
F3 = database creation

F4 =Access privilege check

F5 = Table creation

F6 = Data Insertion

F7 = Data scheduling



EXPERIMENTAL RESULTS

To evaluate the system functionality and performance, we have hosted our system on 4 different nodes. These nodes are connected in a LAN. 1 system is treated as a cloud system. This system has 4GB ram with ore i5 processor. Other 2 nodes are treated as a tenants. These tenants have 2 gb ram with i-3 processor. The communication is done using TCP protocol.

The system is build using java – jdk 1.7. Remote method in vocation(RMI) strategy is used to communicate between tenant and cloud. Mysql database is used to store relation databases.

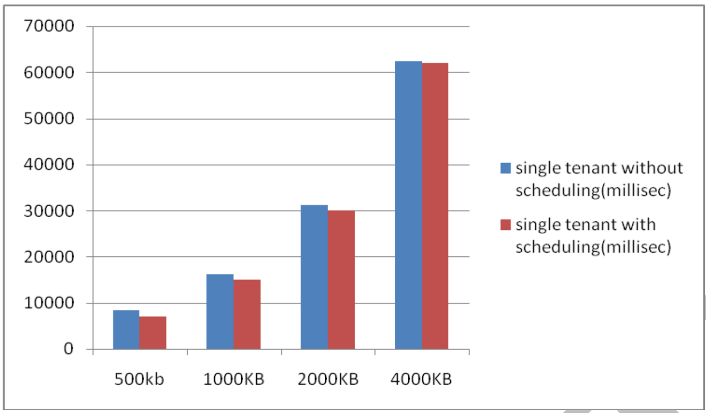
Datasets:

To test the system with different samples we have used following datasets:

Dataset	size in MB	Time
dqflog	2.69	30486
NBA	5.16	77693
Amazon	42.8	650975

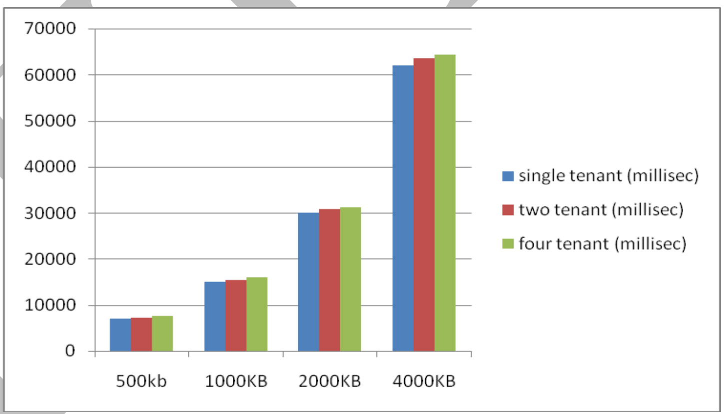
We have tested our system without and with client side scheduling for different sizes of dataset. Following table shows the detailed evaluation result.

File size in KB	single tenant without scheduling (millisecond)	single tenant with scheduling (millisecond)
500kb	8326	7046
1000KB	16112	15003
2000KB	31278	30046
4000KB	62348	62090



System load balancing is tested by connecting multiple nodes at a time to the sever. Following table shows the performance of the system with load balancing.

File size in KB	single tenant (millisecond)	two tenant (millisecond)	four tenant (millisecond)
500KB	7046	7273	7597
1000KB	15003	15470	15906
2000KB	30046	30863	31153
4000KB	62090	63619	64413



CONCLUSION

The growing success in migrating enterprise applications to cloud environments also increases the pressure to integrate enterprise applications (both on premises and services in the clouds) with other distributed services on clouds. We describe a practical implementation (JTangCSB), an integration platform to deliver efficient and cost-effective integration and run-time service. An evaluation of JTangCSB using a realistic application scenario shows the feasibility and effectiveness of the CSB approach. We have also proposed client end scheduling for data migration environment. The est result shows that client side scheduling improve the throughput of system and reduces the response time.

REFERENCES

[1] Jianwei Yin, Xingjian Lu, Calton Pu, Zhaohui Wu, Hanwei Chen, *"JTangCSB: A Cloud Service Bus for Cloud and Enterprise Application Integration"*

[2] Strauch, Steve, et al. "*Implementation and Evaluation of a Multi-tenant Open-Source ESB,"*,ESOCC 2013, pp. 79-93.

[3] Vaquero, Luis M, et al. *"Dynamically scaling applications in the cloud,"* ACM SIGCOMM Computer Communication Review, 2011, 41(1): 45-52.

[4] Georgantas, Nikolaos, et al. *"A coordination middleware for orchestrating heterogeneous distributed systems,"* Advances in Grid and Pervasive Computing 2011, pp. 221-232.

[5] Georgantas, Nikolaos, et al. *"Service-oriented Distributed Applications in the Future Internet: The Case for Interaction Paradigm Interoperability,"* ESOCC 2013, pp. 134-148.

[6] Shen, Zhiming, et al. *"Cloudscale: elastic resource scaling for multi-tenant cloud systems,"* Proceedings of the 2nd ACM Symposium on Cloud Computing. ACM, 2011.

[7] Ni, Jiacai, et al. *"Adapt: adaptive database schema design for multi-tenant applications,"* Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012.

[8] Pippal S K, Kushwaha D S. *"A simple, adaptable and efficient heterogeneous multi-tenant database architecture for ad hoc cloud,"* Journal of Cloud Computing, 2013, 2(1): 1-14.

[9] Nusrat Pasha, Dr. Amit Agarwal and Dr. Ravi Rastogi, *"Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment".*

[10] Amandeep Kaur Sidhu1, Supriya Kinger, *"Analysis of Load Balancing Techniques in Cloud Computing"*

[11] Prof.Meenakshi Sharma, CSE Department SSCET Badhani ,Pankaj Sharma, *"Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm".*

[12] Pooja Samal, Pranati Mishra, *"Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing".*

[13] Ajay Gulati1, Ranjeev.K.Chopra2, *"Dynamic Round Robin for Load Balancing in a Cloud Computing".*

[14] Mayanka Katyal, Atul Mishra, *"A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment".*

[15] Stuti Dave,B H Gardi College of Engineering and Prashant Maheta, *"Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Environment".*