2013 AASRI Conference on Parallel and Distributed Computing and Systems

5-Layered Architecture of Cloud Database Management System

Bashir Alam[1], M.N. Doja[1], Mansaf Alam[2], Shweta Mongia[1]*

1. Department of Computer Engineering, Jamia Millia Islamia, New Delhi-110025, India
2. Department of Computer Science, Jamia Millia Islamia, New Delhi-110025, India

Abstract

Cloud Database Management System is a new emerging concept recently introduced in the world. In Cloud the concept of Standard architecture of Cloud Database Management System is not yet been implemented. In this paper we are proposing a framework for 5-layered architecture in cloud database management system. First layer introduced is the External Layer, this layer is closest to the user, in which manageability, providing transparency and security are the important issue that should be considered. Second layer is the Conceptual Middleware Layer, as there are heterogeneous databases and clouds are available in the market, so here interoperability is the major issue. Third layer is the Conceptual Layer in which programming techniques, transaction management, query processing and optimization are the issues that should be considered. Forth layer is the Physical Middleware Layer, as there are various platforms available so here also, interoperability between various platforms are the biggest issue and the last layer is the Physical Layer in which how data can be stored so that it can be easily accessible without so much overhead so here data security, storage, backup, load balancing, partitioning, scaling, elasticity, fault tolerance and replication are the important issues that should be considered.

* Corresponding author. Tel.: 9654572859
E-mail address: shweta.mongia@yahoo.com.

1. Introduction

1.1. Cloud Database Management System

Data management applications are potential candidates for deployment in the cloud. A cloud Database is a database that typically runs on a cloud-computing platform such as Google, Microsoft, Salesforce.com, Rackspace, Amazon EC2 etc. [3]

Database-as –a –service (DBaaS) is attractive for two reasons.

- *Hardware cost:* Due to economies of scale, the hardware and energy costs incurred by users are likely to be much lower when they are paying for a share of a service rather than running everything themselves.

- *Software cost:* The cost incurred in a well-designed DBaaS will be proportional to actual usage (pay per use)- this applies to both software licensing and administrative costs [5].

The cloud Database was conceived for the purpose of online data management by using the variety of distributed databases. Cloud computing in general, with the flexible pay-as-you-go pricing models and different plans it presents the one of the best solutions for startup and small companies that are developing new products. In this way, cloud computing is reminiscent of application service provider (ASP) and Database-as-a service (DBaaS) paradigms. DBaaS offerings are tightly integrated with other Platform–as-a Service (PaaS) gives the organization the opportunity to focus on developing their products and do not waste any resources on administration of the platform and gives an opportunity to fully focus on the development of the product. Data centre are used for hosting theses services. Data centres use commodity hardware for computation and storage.

2. Current State of work

The Three important challenges like Efficient multi tenancy, privacy of database and elasticity of scalability have not been address in earlier databases. Curino et al.[5] have presented the key technical feature of Relational cloud as follows:

- A workload aware approach to multi tenancy that identifies the workloads that can be co located on a database server achieving higher consolidation and better performance than existing approaches.

- The use of graph based partitioning to achieve elasticity for even complex transactional workloads.

- An Efficient security scheme that enables Structured Query Language (SQL) queries to run over data encrypted using some efficient encryption algorithm.

These all are based on workload awareness [5].

Technological changes advances have put shared disk performance at par with shared nothing and cloud computing strongly favours the shared disk architecture. Cloud computing economies leveraging the power of multi tenancy delivers extremely fast shared storage at dramatically minimized cost. Virtualization adds these advantages by enabling users to scale elastically and to pay only for the resources they use. Because of all these changes in the technology now shared disk is equally considerable for the clouds [4].

Sometimes data stored in the cloud often need to be combined with the data stored in relational databases. A system named "Bigintegrator"[6] to enable queries that combine data in the cloud based data stores with relational databases, there are several cloud based systems available but with limited query languages.

A system "SQLMR"[1], which is a hybrid approach to fill the gap between SQL-based and MapReduce data processing. Map reduce provides a framework for large data processing and is shown to be scalable and fault tolerant on commodity machines. However it is very difficult to learn than SQL-like languages and maintaining and reuse are not easy. On the other side traditional SQL-based processing is not scalable but

users are familiar with it.

To combine the fault tolerance, heterogeneous cluster and ease of use out of the box capability of MapReduce with the efficiency, performance and tool plug ability of shared nothing parallel database systems, Hybrid solution is needed. This Solution will have a significant impact on the cloud database market [2]. Cloud database system architecture organises cloud architecture into a hierarchy of three level cloud data centre, cloud service provider level, and client level [9].

3. Layered Architecture of Cloud Database Management System

There are many challenges to the growth of cloud computing applications like availability of a service, Data Confidentiality, Shared nothing architecture. A successful Cloud Database Management System are designed to satisfy as much as the following goals Availability, Scalability, Elasticity, Multitenancy, Load Balancing, Fault-tolerance, Capability to run in Heterogeneous environment, Flexible query interface [7]. A Federated Cloud Architecture is needed, which can enhance the challenges and meet the goals of Cloud. The Standard Architecture for Cloud Database Management System is not yet been developed so in this paper we are proposing the five layered architecture for Cloud Database Management System as shown in the figure1.
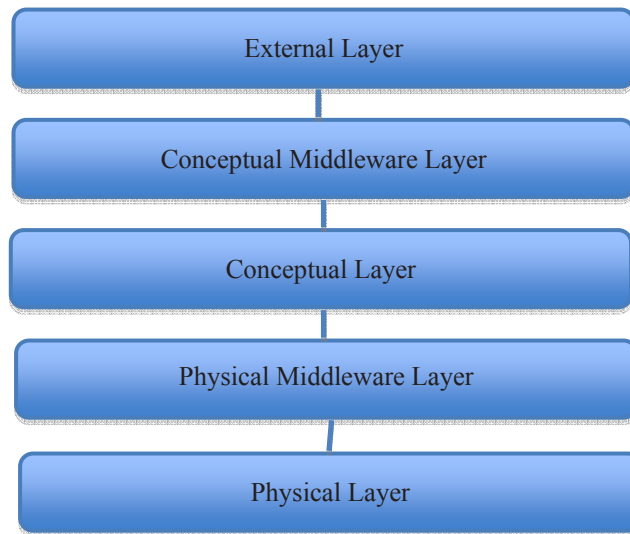

Figure 1: Layered Architecture of Cloud Database Management System

Proposed five layers along with their issues and challenges are described below.

3.1 External Layer

This layer is presentable to the users. More and more companies are moving their data management applications from expensive, high-end servers to the cloud that is composed of cheaper, commodity machines. Here the main function of the service provider is to manage and provide the services with full transparency and Security.

Manageability: This layer deals with managing the various users. This also keeps record of the time a particular user uses the CDMBS (Cloud Data Base Management System). User payment is decided based on this time. This layer also keeps status of payment due to a user. This layer should also generate report like

payment history of an user, use history of an user, user summary of the CDBMS mentioning which user used the CDBMS for how long time . After verifying the user authenticity and validity this layer should transfer control to the next layer.

Security: The external layer provides user authentication mechanism with the help of user id's and passwords. User authentication provides that someone or something should be accepted as being legal one. The legal one is permitted to access their database system in cloud but an illegal one is forbidden.

Transparency: Main function of this layer is to provide transparencies to the users. Transparency is one of the advantages of cloud databases where it means that the physical placement of data is not known to the users. Transparency makes development of various types of real time applications easier. Its support flexibility mobility etc.

3.2 Conceptual Middleware Layer

This layer provides the facility of hiding the conceptual level heterogeneity among different databases like SQL, DB2, Oracle etc. as shown in the figure 2.
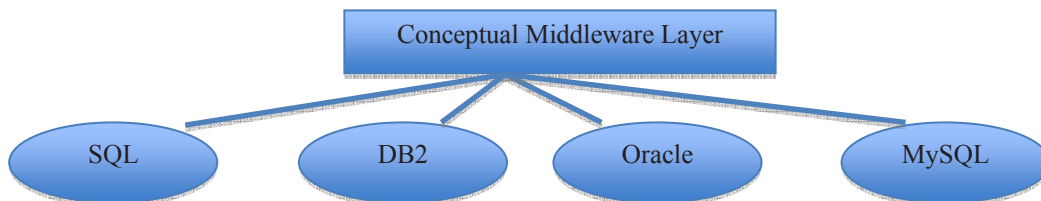

Figure 2: Conceptual Middleware Layer

Interoperability: This layer provides interoperability. Interoperability means operate irrespective of their underlying databases. For example if a customer on 'A' database wants to share data with another customer with 'B' database, they are able to share the data irrespective of their underlying different databases of different vendors with the help of this layer.

3.3 Conceptual Layer

This layer represents the logical structure of the entire database and deals with the internal processing on data. As Cloud deals with various types of data, here users need to combine the traditional data with the data that are placed on the cloud so various types of systems are required for Cloud Databases that provides all these functionalities. Some of the systems or languages like Bigintegrator, SQLMR, Saleforce object query language (SOQL) etc., have been developed for Cloud, that give the results on the basis of their analysis, so users without knowing the logics behind their tasks, gets the results. So, here this layer deals with the following issues- which one of the various types of query languages or which one of the so many available programming techniques, which one of the different query optimization techniques, what transaction management technique should be used for Cloud database Management System. This layer deals with programming techniques, Efficient query processing, Query optimization and security.

Programming Techniques:. This layer resolve the problems such as which query language or programming technique is best suited for cloud databases.

Query Processing and Optimization: The responsibility of this layer is to provide the results to users in minimum time with minimum cost. For this Query Optimizer may be used. Query optimizer takes the query

and finds the cheapest execution plan among the many possible execution plans that give the same answer.
Security: At this layer the main concern is that data should not be destroyed, altered disclosed or copied by any unauthorized user. The conceptual layer provides authorization control methods, which prevents sensitive data being accessed by any unauthorized users. It includes multilevel security policies.

3.4 Physical Middleware Layer

This layer provides the facility of hiding the heterogeneity across the different platforms like windows, Mac OS, Linux etc. as shown in the figure 3.
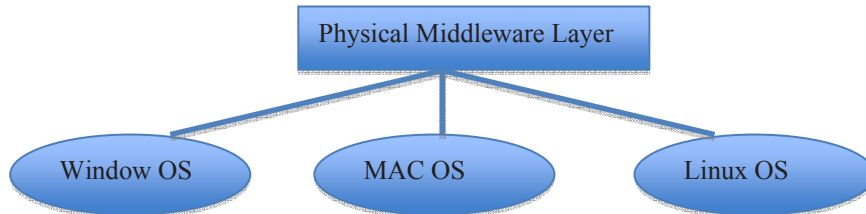
Figure 3: Physical Middleware Layer

Interoperability: Here at this layer, Interoperability means operate irrespective of their underlying Platforms. Customer using Mac OS should not find any difficulty in accessing the data from database of another customer, which is using Windows OS.

3.5 Physical Layer

This layer represents the physical representation of the data. In a cloud database service, the backend is being overseen by a **Physical layer** that's responsible for the continuous monitoring and configuring of the database to achieve optimal scaling, high availability, multi-tenancy and effective resource allocation in the cloud. The issue of Data Security, privacy, partitioning, shared disk architecture or shared nothing architecture are handled on this layer. Various issues of this layer are given below.

Backup& Replication: As data availability is important for Clouds. Restoring and replication techniques should be used to ensure that organizations do not experience any data loss. Data is replicated at various sites.

Partitioning: Partitioning is a technique that is used by cloud providers to balance and share the load. This balancing and sharing improves the performance, scalability and availability of CDBMS in multitenant environment [8].   Where to use horizontal partitioning and where to use vertical partitioning for cloud data, these decisions are to be taken at this layer.

Storage: At this layer, issues like where the data should resides so that it take lesser time for accessing the data should be explored. For Cloud Databases shared disk architecture is best suitable. It should also use sophisticated caches to maintain the most recently used metadata in memory.

Indexing: Indexing provides easy access path to the data. Indexes are constructed for the files that stored in the database. Cloud Applications should index various types of fields to deliver scalable performance.

Load Balancing: It must be able to automatically move load between servers so that most of the hardware resources are effectively utilized and to avoid any resource overloading situation.

Fault Tolerance: Systems for the cloud should be designed to handle failures in order to remain operational (perhaps at a reduced level) in the presence of failures. So, the responsibility of this layer is to provide the concurrency control, recovery techniques and deadlock detection techniques.

Security: This layer provides security by transforming raw data into an unreadable form with the help of encryption/decryption techniques. The threats like privacy, modification and fabrication can be prevented

using these techniques.

4 . Conclusion

The five layered Architecture of Cloud Database Management System have been introduced in this work. Functions of the five layers namely External Layer, Conceptual Middleware Layer, Conceptual Layer, Physical Middleware Layer, Physical Layer have been described. We have also explained various issues and challenges involved with each layer.

References

[1]M. Hsieh, C.Chang, L. Ho, J. Wu and P. Lui. "SQLMR: A Scalable Database Management System for Cloud Computing". International Conference on Parallel Processing (ICPP) 2011, page315-324 .
[2]Daniel J. Abadi. "Data Management in the Cloud: Limitation and Opportunities ". IEEE Data Engineering Bulletin, Volume 32(1), 2009, page 3-12.
[3]R. Aggarwal, Philip A. Bernstein, A. Ailamaki etc., "The Claremont Report on Database Research", record ACM SIGMOD, Volume 37(3), 2008 , page 9-19.
[4]Mike Hogan. Report on "Database Virtualization and the cloud". *Report*
2009(url=http://www.scaledb.com/pdfs/cloud_databases_whitepapers2.pdf).
[5]C. Curino, E.P.C Jones, R. A. Popa, N. Malviya, E.Wu, S. Madden, H. Balakrishnan and N. Zeldovich., "Relational Cloud: A Database-as-a Service for the Cloud ". 5th Biennial Conference on Innovative Data Systems Research (*CIDR)* 2011, page 235-240.
[6]M. Zhu and T. Risch. "Querying Combined Cloud-Based and Relational Databases". International Conference on cloud and service computing (CSC) 2011, page 330-335.
[7]S. Sakr, A.Lui, Daniel M. Batista, and M. Alomari: "A Survey of Large Scale Data Management Approches in Cloud Environment". IEEE Communication Surveys and Tutorials, Volume 13(3), 2011, page 311-336.
[8]Yvette E. Gelogo and S. Lee, "Database Management System as a Cloud Service", International Journal of Future Generation Communication and Networking, Vol. 5(2), 2012,page 71-76 .
[9]Mansaf Alam and Kashish Ara Shakil, "Cloud Database Management System Architecture", UACEE International Journal of Computer Science and its Applications, Volume 3(1), 2013, page 27-31.