Journal of Engineering and Applied Sciences 13 (13): 5084-5089, 2018
ISSN: 1816-949X

A Seven Tier Architecture of Cloud Database Management System

M. Husain, Yazed. ALsaawy and Ali Tufail
Islamic University, Madinah, Saudi Arabia

Abstract: The cloud computing is a leading driving movement for information technology where the major organizations are using more than one cloud provider, hence, computing will certainly become more distributed than it currently is this poses management, architectural and performance challenges. In this study, we propose a unique seven tier architecture of cloud database management system. The first tier elaborates the physical layer and specifies how to physically store the database files on the secondary storage devices. The second tier, physical middleware layer, provides seamless connectivity regardless of the underlying platform. The third tier, conceptual lower layer, supports logic design, database connectivity, data abstraction and partitioning. The fourth tier, conceptual layer, deals with query processing, searching and security of various databases of the system for the internal processing on data. The fifth tier, conceptual middleware layer, provides seamless connectivity amongst various databases. The sixth tier, external layer, focuses on user's prospective where the same database can be viewed by various user's simultaneously. The seventh tier, external upper layer, provides interface interoperability, interface design and security which are the major issues for cloud databases. We argue that our proposed distinct seven tier architecture would not only help to design a distributed database architecture in an efficient manner but would also optimize the querying/searching in a cloud environment.

Key words: Cloud computing, schema design, hybrid cloud, virtualization, DBaaS, cloud environment

INTRODUCTION

In the traditional database approach, data is usually stored on a single server that serves as the source of all the querying and searching. The data is generally accessed directly or through local area network. However, with the advent of cloud computing the whole new concept of distributed database is now being materialized into reality (Kossmann *et al.*, 2010). On one side, cloud computing gives an option to companies to outsource their database on the other side it brings whole new challenges related to management of database, security, query efficiency, consistency, etc. Nevertheless, the concept of cloud computing is flourishing and major player o f industry, like Google, Microsoft, Salesforce. com, Rackspace Amazon EC2, etc. are now competing to provide best cloud platform and related services (Alam *et al.*, 2013; Dinh *et al.*, 2013). Generally, the cloud database management system is deployed in the following three ways (Curino *et al.*, 2011; Singh and Sandhu, 2011).

Virtual machine image: In this approach, the database management system can run only on Virtual Machine (VM) instances and these instances are sold out by the cloud provider and they are responsible for the infrastructure including uploading or purchasing of the DBMS.

Database as a Service (DBaaS): In this approach, the cloud provider is overall responsible for maintaining the database management system. Mainly it is more attractive due to hardware and software costs.

Managed hosting: This approach provides the facility to install, maintain and manage the overall database accomplishment.

Before deploying cloud database management system the organizations should keep in mind about performance, budget, data governance and staffing which directly affect the organizations.

The cloud computing is a leading driving movement for information technology where the major organizations are using more than one cloud provider, hence, computing will certainly become more distributed than it currently is this poses management, architectural and performance challenges (Curino *et al.*, 2011; Bloor, 2011).

In this study, we propose a seven tier architecture of cloud database management system. These tiers are physical layer, physical middleware layer, conceptual

Corresponding Author: M. Husain, Islamic University, Madinah, Saudi Arabia

lower layer, conceptual layer, conceptual middleware layer, external layer and external upper layer. Our architecture, handles the complexity of the distributed database in cloud environment. In order to enhance the efficiency and the optimization of database querying process we have suggested to divide the overall process into seven distinct tiers/layers. Where each layer is responsible to deal with its defined tasks and functionalities while complementing the other layer. We argue that by following our suggested architecture the overall performance of the distributed database will get optimized, efficient and effective.

Literature review: Dimovski (2013) argues that the cloud database has the potential to provide an optimal solution for web and mobile applications where DBaaS and PaaS provides an opportunity to the organization to develop products by not wasting resources. However, he suggests that the cloud is not yet ready and reliable to store highly sensitive enterprise application data but it can be used for testing and development purposes.

Another important aspect of cloud database is the query processing. Stonebraker *et al.* (2005) present query processing technique to enhance the performance of the compressed data. They proposed column oriented algorithm and performed a comparison with the traditional algorithms. Their algorithm created a decision-tree to help the database designers in the decision of compressing a particular column. However, they do not discuss the decompression cost of the various compression algorithms. Also, by Kumar and Bansal (2012) researcher discuss the optimization of database performance in context to lightweight compression schemes. Although, their proposed scheme is well suited for the traditional database, however, it cannot be applied in the cloud database architecture.

Alam *et al.* (2013) have proposed a framework for 5-layered architecture in cloud database management system. Researcher describe functionality of each layer, however, they have overlooked a number of functionalities related to the distributed and the cloud database environment.

MATERIALS AND METHODS

Deployment models: There are various deployment models related to the cloud computing. But generally these models can be divided into four categories. The selection of any of these models takes into account the user requirements (Anonymous, 2017).

Private cloud: This type of cloud infrastructure is deployed for a specific organization where the operation and maintenance is managed by third party at the premises or through in-house.

Community cloud: This type of cloud infrastructure is deployed where same type of organizations exist. It reduces the costs among them and the operation and maintenance is managed by third party at the premises or through in-house.

Public cloud: It is managed by cloud service provider which enables the user to deploy services and develop the system with minimal cost.

Hybrid cloud: This type of cloud infrastructure is deployed for various clouds and they may be different in nature but they should be able to communicate to each other. It is the combination of private and public clouds where the data can be shared and fulfill the requirements (Anonymous, 2017). The two basic concepts are the key features of cloud environment, i.e.

Abstraction: In the cloud environment the user's are unknown about physical data storage locations, application management, administration and the user's identification, etc (Sosinsky, 2010).

Virtualization: The cloud environment provides the facility to share the resources together as and when required from centralized infrastructure and present them as virtual resource.

RESULTS AND DISCUSSION

Architecture of cloud database management system: In this study, we present our seven tier architecture of cloud database system. Following are the seven distinct layers that we propose for our architecture:

- Physical layer
- Physical middleware layer
- Conceptual lower layer
- Conceptual layer
- Conceptual middleware layer
- External layer
- External upper layer (Fig. 1)

Physical layer: This layer deals with the design of the database. It also specifies how to physically store the database files on the secondary storage devices. We can

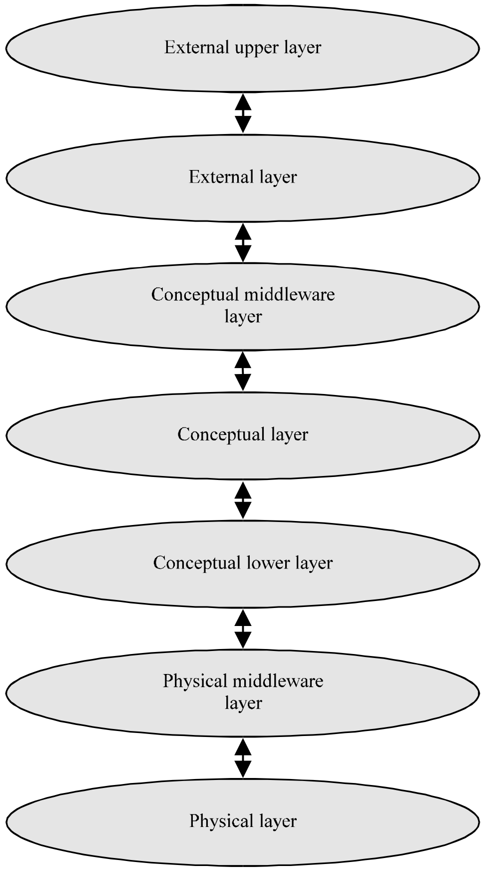

Fig. 1: Architecture of cloud database management system

mention the data characteristics, like data type and data size for a particular attribute. In particular in a cloud environment where the storage can be at multiple locations or servers, this layer can play a pivotal role (Alam *et al*., 2013). Following are the main functionalities that this layer supports (Fig. 2).

Schema design: It deals with the process of creating a real database. The schema design has to be flexible to research in a cloud environment.

Data storage: It provides the mechanism to store the actual data in the database as per the schema design. The storage technique should cater the storage of data at multiple locations or servers. The technique used here can greatly affect the query optimization.

Data manipulation: Manipulation of data in a cloud should be fast and efficient. If any data has to be replaced or updated it should be done reliably regardless of the data location.

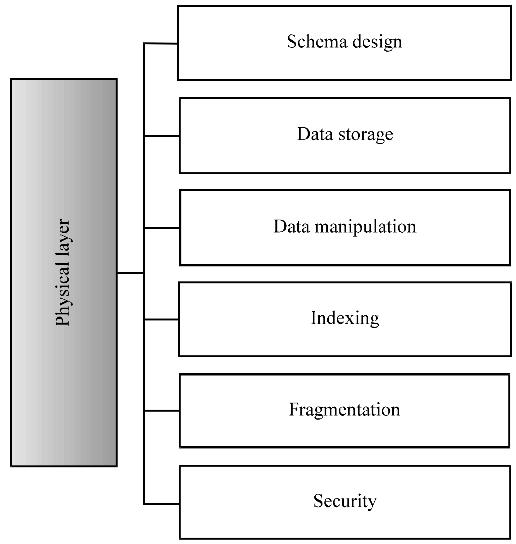

Fig. 2: Physical layer

Indexing: Indexing can be problematic in a cloud environment, since, data can be distributed. An efficient indexing technique should be devised and implemented in order to make an easy access path to the data.

Fragmentation: In this layer, we can divide the data horizontally and vertically for an easy access. This can be useful for cloud environment.

Security: In a cloud environment where data can be distributed across many servers or locations, security can be a very big concern for organizations. In this layer, we can provide security by utilizing encryption techniques. This will make sure that the data is not available in its actual form.

Physical middleware layer: This layer deals with various operating systems environment. It provides seamless connectivity regardless of the underlying platform, i.e., Windows OS, Linux OS, Mac OS (Fig. 3 and 4).

Conceptual lower layer

Logic design: It deals with the logical structure of the entire databases, describes the records and how they are related. The entire database is managed by DBA and it describe the structure of all users.

Database connectivity: The ability of this model is to provide the connectivity among the web server and application server which implement logic queries into the database.

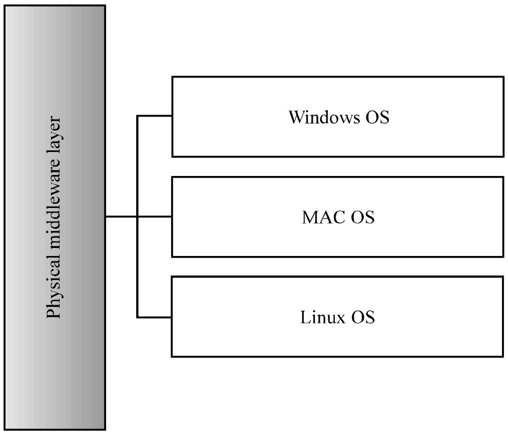

Fig. 3: Physical middleware layer

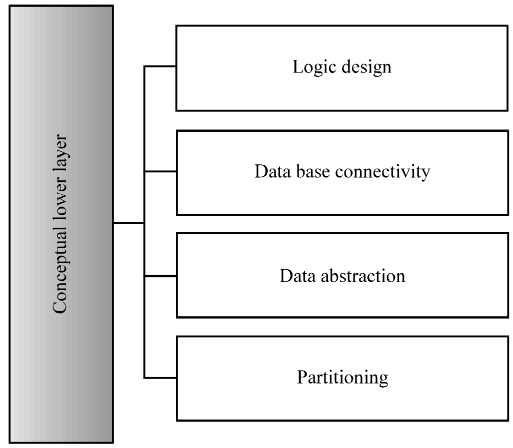

Fig. 4: Conceptual lower layer

Data abstraction: In cloud environment where all the data are not needed to be viewed by the user, it provides the facility to hide the irrelevant data from the users.

Partitioning: In the cloud environment, we tend to have a huge amount of data with large tables. We can use the technique of partitioning by dividing the table into various parts. The idea is to split the large tables to smaller ones, so that, the queries can access only a part of data and information access could be faster.

Conceptual layer: This layer deals with the program, query processing, searching and security of various databases of the system for the internal processing on data. In the cloud various type of data are incorporated to combine the traditional data, so that, the data which is placed on the cloud may be accessed and for this purpose various types of systems are required for cloud databases.

The various languages supported by cloud are such as Microsoft Azure, Google AppEngine, Haskell, Erlang, OpenStack, SQLMR, etc. that gives the results on the basis of their analysis, i.e., we use to design the various forms where the user sends the request and based on that he can get the result after interacting with the corresponding databases which physically resides in some database files on the secondary storage devices. However, it does not matter that how the database has been designed and who is going to use this program irrespective of their qualifications and expertise, language, specialized area, etc (Fig. 5).

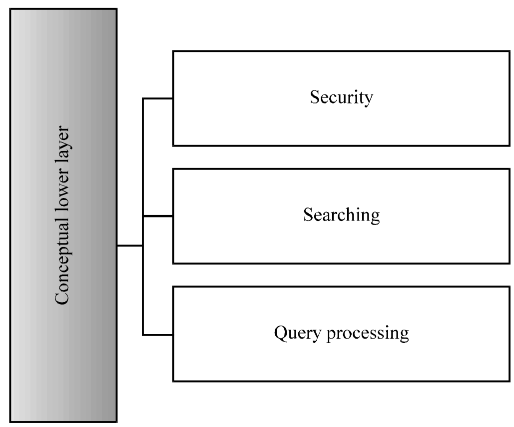

Fig. 5: Conceptual layer

Security: In cloud environment, this layer provides security for the interaction with the database without affecting the other layers and the source code of this layer cannot be changed by any other layer or the system. The threats like privacy, modification and fabrication are fully secured and cannot be changed. The information stealing mobile malware, insecure network, insecure marketplaces and proximity-based hacking are the major threats in the cloud environment.

Searching: In this layer, the posed query is interact with the relevant database and search the data in that and after query processing it returns the result in front of the user.

Query processing: In this layer, we provide the input as a query which interact with the database and generate the result to the user. The major issues occurs in cloud environment is the protection of data and query between the cloud, user and possessor.

Conceptual middleware layer: This layer provides seamless connectivity amongst various databases like

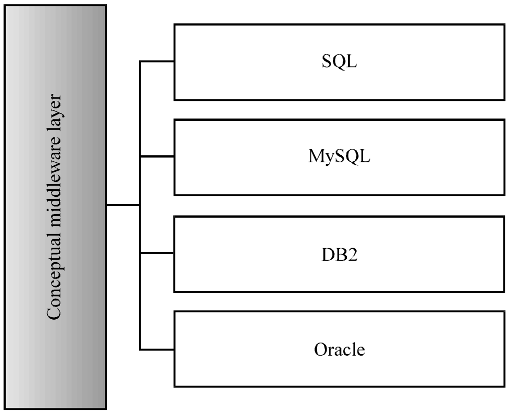

Fig. 6: Conceptual middleware layer

SQL, MySQL, DB2, Oracle. It enables the users to interoperate between any given databases (Fig. 6).

External layer: This layer deals with the user's prospective where the same database can be viewed by various user's simultaneously. It also describes part of the database for particular group of user's. Users can access the by their own customized way according to the need, hence, the same data can be viewed by different users in various ways at the same time.

In this level user interacts with the form/window that appears in front of him and sends the queries to get the result. At this level user does not think that who has designed the form and what was his qualification and when was it designed. The same way at this level the user is not concerned that from where he is getting the answer of his query from how many databases he is getting the result and where they are stored and who was the designer of the database and at what time was it created what was the qualification of the designer, etc (Fig. 7).

DML compiler optimization: In this layer while we wanted to add, delete, change or to perform any other operation onto the database, we need some media to perform it. DML compiler helps to handle these operations in efficient manner. It minimizes resource usage, reduce time complexity and finds near optimum solution.

Multiuser interaction: In the cloud environment when we think about a common architecture that are used to implement multi-user database management systems like, client server architecture, file server and teleprocessing.

Seamless connectivity: External layer will help to connect the users seamlessly regardless of the underlying

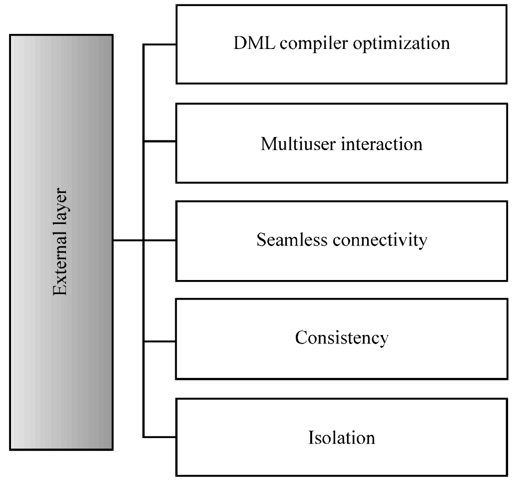

Fig. 7: External layer

database. The users main concern will be to send the query and get the response without worrying about the user device that is sending the query and the underlying architecture and the database responding with the result.

Consistency: In a cloud environment where data can be distributed across many servers or locations and when the transaction process going on unwanted or incorrect data should not insert into the database and should maintain uniformity of the database.

Isolation: In a cloud environment where data can be distributed and if the multiple transactions processing simultaneously then all the transactions should behave like individual transaction and they should not affect to each other. It means each and every transaction executed as if they are independent.

External upper layer

Interface interoperability: Cloud environment provides the facility to interface a system with other systems without any additional features on the part of end user. Several organizations are deploying distributed n-tier client/server applications, most of which require access to data or transactions on existing systems. Standards like TCP/IP, HTP and HTML etc. can serve as a good example of interface interoperability where CORBA and its ORB to provide common interfaces. The interoperability reduces operational cost and complexity, leverages existing investments, rapidly enables deployments (Fig. 8).

Interface design: It provides the interface of the system to varied users. Some examples of interfaces are as follows, menu based interfaces for web clients form Based

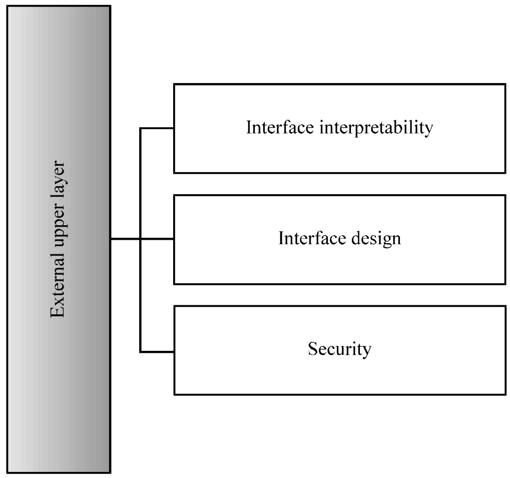

Fig. 8: External upper layer

interfaces, graphical user interfaces, natural language interfaces, interfaces for parametric users and interfaces for the DBA.

Security: Provides a powerful and flexible security mechanism by hiding parts of the databases from certain users. The user is not aware of the existence of any attributes that are missing from the view.

CONCLUSION

Cloud computing is an innovative concept and has been turning in to reality on a fast pace. With cloud computing there comes the challenge of distributed database and the service of distributed querying. In this study, we proposed a distinct seven tier architecture of cloud database management system. The main focus of our proposed architecture is to simplify the complexity of distributed querying of databases spread across different nodes of the cloud. We have defined specific functionality for each layer of the architecture that would help to manage the cloud database efficiently and effectively. We have proposed seven tiers namely physical layer, physical middleware layer, conceptual lower layer, conceptual layer, conceptual middleware layer, external layer and external upper layer. All these tiers cover a vast range of the functionality supporting the cloud architecture. Some of the functionalities are searching, indexing, optimizing, portability, seamless connectivity, interoperability, security, etc. We have also, high lighted various issues and challenges involved with each tier.

REFERENCES

Alam, B., M.N. Doja, M. Alam and S. Mongia, 2013. 5-layered architecture of cloud database management system. AASRI. Procedia, 5: 194-199.

Anonymous, 2017. Introduction to cloud computing. Dialogic Corporation, Milpitas, California.

Bloor, R., 2011. The Suitability of Algebraix Data's Technology to Cloud Computing. The Bloor Group, Austin, Texas,.

Curino, C., E.P. Jones, R.A. Popa, N. Malviya and E. Wu *et al*., 2011. Relational cloud: A database-as-a-service for the cloud. Proceedings of the 5th Biennial Conference on Innovative Data Systems Research (CIDR'11), January 9-12, 2011, Asilomar Conference Grounds, Asilomar, California, USA., pp: 235-240.

Dimovski, D., 2013. Database management as a cloud-based service for small and medium organizations. Master Thesis, Masaryk University, Brno, Czech Republic.

Dinh, H.T., C. Lee, D. Niyato and P. Wang, 2013. A survey of mobile cloud computing: architecture, applications and approaches. Wirel. Commun. Mob. Comput., 13: 1587-1611.

Kossmann, D., T. Kraska and S. Loesing, 2010. An evaluation of alternative architectures for transaction processing in the cloud. Proceedings of the ACM International Conference on Management of Data (SIGMOD'10), June 06-10, 2010, ACM, Indianapolis, Indiana, USA., ISBN:978-1-4503-0032-2, pp: 579-590.

Kumar, N. and K.K. Bansal, 2012. Different compression techniques and their execution in database systems to improve performance. Intl. J. Adv. Res. Comput. Sci. Software Eng., Vol. 2,

Sakr, S., A. Liu, D.M. Batista and M. Alomari, 2011. A survey of large scale data management approaches in cloud environments. Commun. Surv. Tutorials IEEE., 13: 311-336.

Singh, T. and P.S. Sandhu, 2011. Cloud computing databases: Latest trends and architectural concepts. Intl. J. Comput. Electr. Autom. Control Inf. Eng., 5: 86-89.

Sosinsky, B., 2010. Cloud Computing Bible. John Wiley & Sons, Hoboken, New Jersey, ISBN:9781118023990, Pages: 532.

Stonebraker, M., D.J. Abadi, A. Batkin, X. Chen and M. Cherniack *et al*., 2005. C-store: A column-oriented DBMS. Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05), August 30-September 02, 2005, ACM, Trondheim, Norway, pp: 553-564.