Scientific Journal Impact Factor (SJIF): 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

DATABASE MANAGEMENT CHALLENGES IN CLOUD ENVIRONMENT

Madhukar Shelar[1], Shirish Sane[2] and Vilas Kharat[3]
[1] *Ph.D. Research Scholar, Department of Computer Science, Savitribai Phule Pune University*
[2] *Department of Computer Engg, K.K.Wagh Institute of Engineering Education and Research, Nashik*
[3] *Department of Computer Science, Savitribai Phule Pune University*

Abstract— Recently cloud computing is widely used technology in delivering computing resources as a service. There are variety of web-based applications hosted on cloud computing platforms, majority of them are data driven. Therefore database management is now become critical component and challenging task for the DBMS designers and researchers. Scalability, elastic load balancing, pay-per-use pricing and self-managing are the major reasons for the successful cloud database management. This is the review of research published/presented on databases in cloud platform. Various challenges of managing databases in the cloud and various techniques proposed by researchers to face these challenges are presented in this review. It is based on various parameters like database systems in the cloud, scalability and elasticity, autonomy or self-managing database systems and preserving consistency of database in cloud.

Keywords—Cloud Computing, Cloud Database Management, Scalability, Elasticity, Autonomic Database Management

I. INTRODUCTION

Cloud computing is the latest trend in distributed computing. Majority of cloud applications are data driven, so DBMS software powering these applications form critical component in the cloud software stack [4]. Traditional DBMS software's are not cloud friendly, because cloud services such as web servers and application servers which can scale from few machines to hundreds. DBMS cannot be scaled very easily [1]. However, existing databases resulting in a heavy performance impact during elastic scaling [2]. In Cloud computing environment, DBMS must follow the properties such as scalable, elasticity, autonomic and load balancing in addition to the other well-known properties like high level functionality, consistency, performance and reliability. This review surveyed the existing research papers and literatures using systematic approach. We have searched the major research on database management in cloud computing environment in IEEE-Xplore, ACM Digital Library, Springer Link and Google Scholar. We have decided to select limited papers on the desired properties of cloud databases such as scalability, elasticity, autonomy, reliability, availability and consistency.

II. DATABASES IN THE CLOUD

In cloud environment, data is stored on multiple dynamic servers at data centers rather than on dedicated servers like in traditional data storage. When storing a database, user sees a virtual server, but in reality the data is stored on any one or more of the servers at data centers. Currently cloud platforms have very little support for database design related virtualization enhancement, but in future designing databases specific for cloud is sure possibility [11]. Divyakant Agrawal et.al [12] presented a tutorial which broadly described the challenges faced by the application developers and DBMS designers in developing and deploying internet scale applications.

The components for the cloud service like application servers, web servers are scaled dynamically as need arises. However DBMSs and RDBMSs cannot be scaled very easily, so they are not cloud friendly [1]. Many data management key-value store technologies such as Google's AppEngine, Microsoft's Azure, Amzon's AWS are used today to maintain databases in cloud platform. In these systems, data is represented as Key-Value pairs, and atomic access is provided only at the granularity of single keys. While these properties are insufficient for the web applications such as online gaming, social networks, collaborative editing, and many more which emphasize collaboration[20], because they requires consistent access to groups of keys, scalable and consistent multi key access is critical for such applications. Sudipto Das et.al [20] design and implement G-Store which uses a key-value store as an underlying substrate to provide efficient, scalable, and transactional multi key access.

Key-Value stores have emerged as a preferred choice for scalable and fault tolerant data management, but lack the rich functionality, and transactional guarantees of RDBMS [22]. Sudipto Das et.al [22] presented ElasTraS, an Elastic Transactional relational database, designed to scale out using a cluster of commodity machines while being fault-tolerant and self-managing. Robin Bloor [8] suggested the need of CDBMS for database management in cloud environment. CDBMS is a distributed database that can deliver a query service across multiple distributed database nodes located in multiple data centers, including cloud data centers. Robin has focused on the point that, traditional RDBMS and column store databases have a centralized architecture and such architecture would encounter a scalability limit at some point within as well as between data centers. Hence, they could not fulfill the role of CDBMS. He examined the suitability of Algebraix Data's technology product A2DB to fulfill the role of a cloud database.

Most database systems have scalability limit. Once users reach these scalability limits, data migration and load balancing are the only recourse [5]. Relational cloud, Databases as a Service (DBaaS) model proposed by Carlo Curino et.al [5,9], which has three components – database partitioning, live migration, workload analysis and allocation. It overcomes the challenges like efficient multi-tenancy, elastic scalability and database privacy

III. DATABASE SCALABILITY AND ELASTICITY

Scalability of system indicates its ability to either handle growing amounts of work in graceful manner when additional resources are added. In cloud computing environment, there is need to support virtually unlimited number of users for web-based applications by making it scalable. Distributed database systems for cloud applications emphasize scalability, fault tolerance and availability, sometimes at the expense of consistency or ease of development [4]. Database scalability in cloud can be achieved in two ways- Data Fusion and Data Fission [1]. First approach Data Fusion architecture maintains multi-key atomicity while ensuring scalability. D. Agrawal et.al [1, 20] designed a scalable data store, G-Store which gives multi-key access guarantees than Mega-store designed by Google. Second approach is a Data Fission, where database schema is partitioned instead of keeping individual tables. This approach tries to minimize the distributed transactions. A prototype system ElasTras [1, 22] uses the concept of data-fission with static-partitioning approach.

Luis et.al [13] suggested to combine three mechanisms for handling the database scalability in PaaS – distributed caching, NoSQL databases, and database clustering. Distributed caching provides the same cache across several nodes, which is useful in clustered applications. NoSQL databases refer to the storage solutions for structured data that are different from the traditional relational database systems. Finally clusters can be built to provide better scalability, availability and fault tolerance to DBMS system.

A scalability, in which a system can be scaled-up dynamically by adding more nodes or can scaled down by removing nodes as per the load fluctuations, is referred to as Elasticity [1]. Google's Bigtable and Yahoo's PNUTS are Key-Value store elastic databases. In cloud computing environment, Database Elasticity and pay-per-use pricing model are major factors for the success of the cloud [1]. Database elasticity can be achieved in two ways, live database migration and virtual machine cloning.

Many researchers have proposed techniques for live database migration on different cloud architectures. A technique Zephyr [1, 10] efficiently migrates live database in shared nothing transactional database architecture. Zephyr minimizes service interruption for the tenants being migrated by introducing synchronized dual mode [10]. An Iterative copy [1, 19] technique based on shared disk architecture focuses on transferring the main memory state of the database partition. S.Das et.al [2] proposed a technique Albatross for live data migration in shared storage database architecture. This technique is efficient and low cost technique for live migration of tenant database in multitenant DBMS. Migration is a key component for elasticity and load balancing, and hence, migration should be supported as a first class notion in any multitenant DBMS [21]. Multitenant DBMS uses different multitenancy models for resource sharing viz. shared machine, shared process and shared table. Shared process model provides good balance of effective resource sharing, schema diversity, performance and scaling. Albatross is based on Shared process model. It provides minimal impact on tenant SLA by working in three phases – Initialization, iterative copying and atomic handover.

Virtual Machine cloning is another way to achieve the elasticity scaling in cloud databases. Emmanuel et.al [7] designed a virtual machine cloning technique - Dolly for database provisioning in shared nothing architecture. It assumes that cloud platforms employ a virtualized architecture. Each database replica runs in a separate virtual machine. Dolly clones entire virtual machine of an existing replica. The cloned virtual machine is started on new physical server. It then synchronizes the state of other replicas, which was before the processing of user request. Dolly triggers the replica spawning process well in advance of the increase in workload. It uses the model to estimate the latency to spawn a new replica based on virtual machine snapshot size and the database resynchronization latency.

Carlo Curino et.al [5, 9] designed a Relational Cloud, a DBaaS(Database As A Service) model from scratch to adapt to the peculiarities of the cloud computing environment. It achieves elastic scalability using graph based partitioning method to spread larger databases across many machines.

IV. AUTONOMIC DATA MANAGEMENT

Autonomic or self-managing is another requirement in cloud database management which is related to the scalability and elasticity. In traditional database management system trained DBAs look after the system and takes appropriate actions to improve the performance of database system. In cloud computing environment database is maintained at data centers on many servers, the manual approach of database administration is not feasible. Database autonomy is essential in the cloud for monitoring the behavior and performance of the system, elastic scaling and load balancing dynamically. D. Agrawal et.al [1] suggested for the need of more research on developing an autonomous and intelligent system controller to manage large systems. Self-management of the data by the cloud will be critical to support large number of users with no database expertise [4].

S.Das et.al [2] designed a technique Albatross for live data migration in shared storage architecture to achieve elasticity. They planned to extend the design by adding an intelligent system control that can model the cost of migration to predict its cost as well as the behavior of the entire system. It will leads to the autonomy in database management.

Cloud database management services are part of PaaS (Platform-As-A-Service). PaaS service providers generates revenue from clients as per SLAs (Service Level Agreements) and may pay cost for the rent their resources from IaaS (Infrastructure-As-A-Service) service providers. The cloud database service providers has to meet the clients SLA as well as to maximize own profit. Zephyr [1, 10] technique focuses on minimizing the service interruption and synchronizes the execution of source and destination node transactions, whereas Albatross [3] technique not only minimizes the service interruption but also minimizes the impact on performance of transaction.

P.Xiong et.al [3] presented a cost aware resource management system named SmartSLA to intelligently manage the resources in shared cloud database system. The SmartSLA considers two important issues- local analysis and global analysis. The former issue is to identify the right configuration of system resources to meet the clients' SLA. The later issue is to take decision on how resources among clients are based on current system status. This system captures relationship between system resources and database performance using machine learning technique.

Allocating resources accurately for tenants in cloud data center results in the reduction of extra cost. Jing Bi et.al [18] presented an optimal autonomic virtual machine provisioning technique for cluster based virtualized multi-tier applications. They have presented a self-managing community mechanism to automate the VMs of configuring and tuning the virtualized multitier applications. Ashraf et.al [17] outlined some of the challenges such as placement, resource partitioning, service level objectives, dynamically varying workloads etc. which are associated with deploying database appliances as virtual appliances on IaaS and tuning their performance. To address these challenges, tools and techniques viz. performance models, optimization and control algorithms, tools for system administrators, co-tuning and hint passing are proposed.

V. CONSISTENCY IN CLOUD DATABASE

Data availability and durability is the main requirement of the cloud vendors. It can be achieved by maintaining certain number of replicas of the data at different locations. However to maintain the consistency between these replicas become a complicated issue. Md. Asfakul et.al [14] propose a tree based consistency approach that maximizes the consistency and performance. The tree is formed between primary server to all replica servers in such a way that the maximum reliable path is ensured between them, which reduce the probability of transaction failure.

It is hard to maintain the ACID guarantees if data is replicated over large geographic distances [15]. When data is replicated over a wide area, consistency part of ACID is typically compromised to yield reasonable system availability. Ozsu etl.al [4] concluded that, current solutions of the cloud data management are successful, but developed with specific simple applications in mind. They have sacrificed consistency and ease of programming for the scalability and elasticity.

VI. CONCLUSION

Extensive research work has been published generally on cloud computing and particularly on management cloud databases. Majority of the work is focused on dynamic scalability or elasticity and load balancing using low cost live data migration on shared nothing as well as shared storage architecture. Since DBMS and RDBMS have scalability limit, many researchers have preferred to use key-value store databases. However due to lack of rich functionality and transactional guarantees, some researchers designed self-managing and scalable database systems. Cloud databases have sacrificed consistency and ease of programming for the scalability and elasticity.

Since cloud computing is gaining growing popularity in the IT industry, further research is needed on efficient and effective management of cloud databases by achieving scalability, elasticity with load balancing and autonomy without sacrificing database consistency.

REFERENCES

[1] D. Agrawal, A.E.Abbadi, S.Das and A.J.Elmore, "Database Scalability, Elasticity and Autonomy in Cloud",
[2] S.Das, S.Nishimura, D.Agrawal, A.E.Abbadi, "Albatross: Lightweight Elasticity in Shared Storage Databases for the Cloud Using Live Data Migration", Proc.VLDB Endowment, Vol.4 No.8, 2011.
[3] P.Xiong, Yun Chi, S.Zhu, H.J.Moon, Calton Pu, Hakan H., "Intelligent Management of Virtualized Resources for Database Systems in Cloud Environment", IEEE 2011, Proc. ICDE Conf.2011.
[4] Özsu M. Tamer, Valduriez, Patrick
[5] 3rd Edition. Springer, 2011
[6] Carlo Curino, Evan Jones, Yang Zhang, Eugene Wu, "Relational Cloud: The Case for a Database Service", ACM, 2010.
[7] Herald Kllapi, Eva Sitaridi, M.M.Tsangaris, "Schedule Optimization for Data Processing Flows on the Cloud", SIGMOD'11, ACM, 2011.
[8] Emmanuel Cecchet, Rahul Singh, Upendra Sharma, Prahant Shenoy, "Dolly: Virtualization- driven Database Provisioning for the cloud", VEE'11, ACM, 2011.
[9] Robin Bloor, "What is a cloud database? The suitability of Algebraix Data's Technology to cloud computing", www.BloorGroup.com, The Bloor Group, 2011.
[10] Carlo Curino, Evan P.C.Jones, Raluca Ada Popa et.al., "Relational Cloud: A Database-as-a-Service for the Cloud", CIDR, 2011.
[11] Aaron J. Elmore, Sudipto Das, Divyakant Agrawal, Amr El Abbadi: Zephyr: Live migration in shared nothing databases for elastic cloud platforms, SIGMOD Conference 2011: 301-312.

[12] Arptia Mathur, Mridul Mathur, Pallavi Upadhyay, "Cloud Based Distributed Databases: The Future Ahead", IJCSE, ISSN:0975-3397, Vol.3 No.6 June 2011.
[13] Divyakant Agrwal, Suditpto Das, Amr El Abbadi, "Big Data and Cloud Computing: Current State and Future Opportunities", EDBT 2011, Mar 22-24, 2011 ACM.
[14] Luis M.Vaquero, Luis Rodero-Merino, Rajkumar Buyya, "Dynamically Scaling Applications in the Cloud", ACM SIGCOMM, Vol 41, Number 1, Jan 2011.
[15] Md. Asfakul Isalam, Susan V. Vrbsky: Tree-based Consistency Approach for Cloud Databases, 2nd Intl. IEEE Conference on Cloud, 2011, 401-404.
[16] Daniel J. Abadi: Data Management in the Cloud : Limitations and Opportunities, Bulletin of IEEE Computer Society, Technical Committee on Data Engineering, 2009.
[17] Sean Marston, Zhi Li, Subhajyoti, Anand Ghalasi : Cloud Computing : The Business Perspective, Proceedings 44th Hawaii Intl. Conference, IEEE, 2011.
[18] Ashraf Aboulnaga, Kenneth Salem, Ahmed Soror, Umar F.Minhas, Peter Kokosielis, Sunil Kamath: Deploying Database Appliances in the Cloud, Bulletin IEEE Computer Society Tech.Committee on Data Engineering, 200
[19] Jing Bi, Zhiliang Zhu, Ruixiong Tian, Quingbo Wang : Dynamic Provisioning Modeling for Virtualized Multitier Applications in Cloud Data-Center, IEEE 3rd Intl.Conf.on Cloud Computing, 2010.
[20] Sudipto Das, Shoji Nishimura, Divyakant Agrawal, Amr El Abbadi : Live Database Migration for Elasticity in Multitenant Database for Cloud Platforms, UCSB Comp Sci Tech Rep 2010-09.
[21] Das, S., Agrawal, D., El Abbadi, A.: G-Store: A Scalable Data Store for Transactional Multikey Access in the Cloud. In: ACM SoCC. pp. 163–174 (2010)
[22] Aaron J. Elmore, Sudipto Das, Divyakant Agrawal, Amr El Abbadi: Towards an Elastic and Autonomic Multitenant Database, NetDB'11, June 12, 2011, ACM, 2011.
[23] Sudipto Das, Shashank Agarwal, Divyakant Agrawal, Amr El Abbadi : ElasTraS: An Elastic, Scalable, and Self Managing Transactional Database for the Cloud, UCSB Computer Science Technical Report 2010.