Database Services for Cloud Computing – An Overview

P. Malliga
Sr. System Analyst
Computer Centre
NITTTR, Chennai

ABSTRACT

Cloud computing has revolutionized the way computing infrastructure is abstracted and used. There is a proliferation of number of applications which leverage various cloud platforms, resulting in a tremendous increase in the scale of data generated as well as consumed by such applications. This paper reviews the features of Cloud computing and then gives a survey of different database architecture for cloud computing. Scalable database management systems are a crucial part of the cloud infrastructure which includes both update intensive workloads and decision support systems. The paper presents an organized picture of the challenges and open issues pertaining to database management systems in developing and deploying internet scale applications in the cloud environment. The typical properties of commercially available databases for cloud computing environment are also brought out.

General Terms

Cloud Computing, Cloud Databases

Keywords

Database as a Service, Cloud computing, Cloud databases, Database architecture, Multitenant Databases

1. INTRODUCTION

Cloud computing is the latest evolution of Internet based computing and it is an extremely successful paradigm of service oriented computing. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources or shared services (e.g., networks, servers, storage, applications, and IT services). The key benefits of cloud computing are reduced costs, reduced complexity, improved quality of service, and increased flexibility when responding to changes in workload.

According to the U.S. National Institute of Standards and Technology, cloud computing consists of five essential characteristics, three distinct service models, and four deployment models [1]. The five essential characteristics are On-demand, Resource Pooling, Rapid Elasticity, Measured Service, Broad Network. The four deployment models are Public Cloud, Private Cloud, Community Cloud and Hybrid Cloud. The three service models namely Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) are the most popular cloud paradigm. The concept is also extended to more granular classifications such as Database-as-a-Service (DBaaS), Storage-as-a-Service, Security-as-a-Service and Testing-as-a-Service. The computing world is shifting from enterprise-centric to data-centric workloads driven by the “Big Data” revolution, while cloud computing is becoming mainstream, reinventing utility/elastic computing as the new mantra for the IT industry.

The major enabling features of cloud computing are elasticity, pay-per-use, low upfront investment, which makes cloud computing a ubiquitous paradigm for deploying novel applications. This has made many SMBs (Small and Medium Business) and SMEs (Small and Medium Enterprises) to deploy the application online which were not economically feasible in traditional enterprise infrastructure settings [2].

Cloud computing is transforming the way data is stored, retrieved and served. Computing resources like servers, storage, network and applications (including databases) are hosted, and made available as cloud services, for a price. Cloud platforms have evolved to offer many IT needs as online services, without having to invest in expensive data centers and worry about the hassle of managing them. Cloud platforms virtually alleviate the need of having their own expensive data-centre. For database environments, the PaaS cloud model provides better IT services than the IaaS model. The PaaS model provides enough resources in the cloud databases which enables the users to create the applications they need. A lot of research has been carried out for more than three decades to address scalable and distributed data management.

2. CLOUD DATABASE MANAGEMENT

2.1 Requirements of Database Management in Cloud

Efficient data processing is a fundamental and vital issue for almost every scientific, academic, or business organization. Therefore the organizations install, manage and maintain database management systems to satisfy different data processing needs. Although it is possible to purchase the necessary hardware, deploy database products, establish network connectivity, and hire the professional people who run the system, as a traditional solution, this solution has been getting increasingly expensive and impractical as the database systems and problems become larger and more complicated. The traditional solution entails different costs. Though the costs of hardware, software, and network more likely to decrease constantly, however people costs do not decrease. In the future, it is likely that computing solution costs will be dominated by people costs [3]. There is also need for database backup, database restore, and database reorganization to reclaim space or to restore preferable arrangement of data. Migration from one database version to the next, without impacting solution availability, is an art still in its infancy [4]. Parts of a database solution, if not the entire solution usually become unavailable during version change.

Enterprises have been using database management systems in their data centre. Initially, it was left to developers to install, manage and use their choice of database instance on the cloud, with the burden of all the database administration tasks being left to the developer. The advantage of this is that you choose your own database and have full control over how the data is managed. In order to simplify the burden on the users of their cloud offerings, many PaaS vendors have started offering database services on the cloud. All physical database administration tasks, such as backup, recovery, managing the logs, etc., are managed by the cloud provider. The responsibility for logical administration of the database, including table tuning and query optimisation, rests on the developer. An organization that provides database service has an opportunity to do these tasks and offer a value proposition provided it is efficient.

Database service provider provides seamless mechanisms for organizations to create, store, and access their databases. Users wishing to access data will now access it using the hardware and software at the service provider instead of their own organization's computing infrastructure. The application would not be impacted by outages due to software, hardware and networking changes or failures at the database service provider's site. This would alleviate the problem of purchasing, installing, maintaining and updating the software and administrating the system. Instead of doing these, the organization will only use the ready system maintained by the service provider for its database needs.

Database systems have proven to be wildly successful in many financial, business, and Internet applications. However, they have several serious limitations such as

- Database systems are difficult to scale.
- Database systems are difficult to configure and maintain.
- Diversification in available systems complicates selection.
- Peak provisioning leads to unneeded costs.

These limitations in traditional database systems are addressed through Database as a service in cloud computing environment. At present, there aren't any true DBaaS offerings that satisfy all these requirements. Therefore, these cloud- computing needs will drive the next generation of database evolution [10].

2.2 Deployment of DBaaS

Database as a Service (DBaaS) is an architectural and operational approach enabling IT providers to deliver database functionality as a service to one or more consumers. There are two use-case scenarios by which an organisation's database needs are met by database offerings on the cloud. They are as follows:

- A single large organisation that has many individual databases which can be migrated to a private cloud for the organisation,
- Outsourcing the data management needs of small and medium organisations to a public cloud provider, who caters to multiple small and medium businesses.

In true sense, DaaS offering should satisfy certain requirements such as:

- Freeing the end developer/user from database administration, tuning and maintenance activities, while offering high performance, availability and fault tolerance, as well as advanced features like snapshot, analytics and time travel.
- Elasticity or the ability to dynamically adjust to changing workloads. Elasticity is required to meet user SLAs (Service Legal Agreement) while minimizing the cloud provider's infrastructure, power and administration expenses.
- Security and privacy guarantees, and a pay-as-per-usage pricing model.

A private cloud is an efficient way to deliver database services because it enables IT departments to consolidate servers, storage, and database workloads onto a shared hardware and software infrastructure. Databases deployed on a private cloud offer compelling advantages in cost, quality of service, and agility by providing on-demand access to database services in a self-service, elastically scalable, and metered manner.

Private clouds are a better option than public clouds for many reasons. Public clouds typically provide little or no availability or performance service-level agreements, and there are potential data security risks. In contrast, private clouds enable IT departments to have complete control over the performance and availability service levels they provide, and can easily enforce data governance regulations and auditing policies.

2.3 Architecture models for data management in Cloud

The world is moving to the cloud, and with that move the core database architecture is finally changing. There has been number of proposal in architecture model for data sharing to meet the changing needs of data management in the Cloud. These include:

- Data replication, which creates multiple copies of the databases. The copies can be read-only, with one master copy where updates occur, and then are propagated to the copies — or the copies can be read-write, which imposes the complexity of ensuring the consistency of the multiple copies.
- Memory caching of frequently accessed data, as popularised by the memcached architecture.
- From the traditional "Shared Everything Scale-up" architecture, the focus shifted to "Shared Nothing Scale-out" architectures. The shared-nothing architecture allows independent nodes as the building blocks, with information replicated, maintained and accessed.

2.3.1 Large Multitenant Databases

Wikipedia describes multi-tenancy as "a principle in software architecture where a single instance of the software runs on a server (or cluster), serving multiple client organizations (tenants). In multi-tenant environments, it is essential for the applications to "behave" uniquely by tenant. In other words, there is a need to accommodate the different workflows, business processes, rules, and user interface logic of every tenant (customer) as well as "protect" their individual data. Database multi tenancy is traditionally considered only in the case of SaaS where different tenants share the same database tables. But different models of multi tenancy are relevant in the context of the different cloud paradigms [5].

- ***Virtual Machine Based:*** The database is deployed as a part of Virtual Assembly or Template and several virtual machines share the same physical server. This offers the maximum level of isolation (at OS level) but creates manageability problems owing to VM sprawl. This option is available through the Infrastructure as a Service (IaaS) Self Service Portal.

- ***Shared Cluster:*** The database is deployed on existing clusterware. Typically the Grid Infrastructure (Clusterware, ASM) and Database software are preinstalled. The cloud service entails the deployment of databases on top of that infrastructure. This option is available in the DBaaS Self Service Portal.
- ***Shared Installation:*** The database is deployed as a single instance database on an existing installation. This option is available in the DBaaS Self Service Portal.
- ***Shared Database:*** The database service is a schema deployment on an existing database. It is assumed for purposes of metering, chargeback that each of the consumers of the database uses a different service while accessing the database.

Multi-tenant is based on subscriber model, so user has freedom to avail the facility as per business requirement or can turnoff. There are different approaches to the advantage out of the multi-tenants database. These are

- Dedicated database (shared nothing): Separate databases per tenant.
- Dedicated table and different schema: Shared database and separate schema.
- Share table/schema (shared everything): Same database and same table.

Different systems target different aspects in the design space, and multiple open problems still remains. In summary a single perfect data management solution for the cloud is yet to be designed

2.3.2 Architecture Principles

Organisations should develop an agile capacity planning strategy as part of their DBaaS architecture [6]. There are varieties of models that can leverage and it is important that the model chosen should be right-sized for the business. Regardless of the format, however, all DBaaS architectural decisions should be traceable back to the set of DBaaS Architecture principles. The following are a few examples of DBaaS architecture principles:

- ***Business Alignment.*** The DBaaS architecture must be in alignment with the business operating model of the enterprise.
- ***Value Tracking***. The value provided by DBaaS architecture will be measured using established key value indicators (KVIs).
- ***Standardized Infrastructure***. The DBaaS architecture must be built on a standardized portfolio of technologies and assets.
- ***Ongoing Architecture Improvement***. The service catalogue and architecture will be continually rationalized to prevent service explosion.
- ***Cost Competitiveness***. The charge-back rates of the database services will be competitive with external providers.
- ***Provider Consumer Collaboration***. Database services, including definitions, attributes and quality of service shall be established and maintained in active collaboration with service consumers.

2.4 Challenges of DBaaS

A DBaaS promises to move much of the operational burden of provisioning, configuration, scaling, performance tuning, backup, data privacy and access control from the database users to the service operator, offering lower overall costs to users. Efficient multi tenancy, elastic scalability and database privacy are the three important challenges which have to be addressed by DBaaS provider [7].

In the database service provider model, user data needs to reside on the premises of the database service provider. Most corporations view their data as a very valuable asset. The service provider would need to provide sufficient security measures to guard the data privacy. At the same time, cloud databases have their share of potential drawbacks, including security and privacy issues a well as the potential loss of or inability to access critical data in the event of a disaster or bankruptcy of the cloud database service provider.

Another challenge facing the database service provider model is that of an appropriate user interface. Clearly, the interface must be easy to use; yet it needs to be powerful enough to allow ease in building applications.

3 EVOLUTION OF DATABASE TECHNOLOGY

In section 2 we have seen different cloud offerings of database services. The underlying technology to provide these services are still traditional SQL based database technology, not specifically reinvented for the cloud.

Database technology evolved from flat file to hierarchical databases and network databases. Later relational databases (RDBMS) have been providing transaction processing with the clarity that emerged from their formal mathematical models, and an elegant way of storing/retrieving data using SQL. With the Big Data explosion combined with the need for massive Web capabilities fuelled by Web 2.0, the industry felt the need for alternatives to traditional RDBMS. There have been several commercial and open source RDBMS products, including IBM's DB2, Oracle Database, Microsoft SQL Server, MySQL and many others.

Due to programming paradigm shift towards object oriented programming, Object Oriented Database Management Systems (OODBMS) evolved in which application data is represented by persistent objects that match the objects used in the programming language. However, object-oriented databases were not very successful, since they were more focused on addressing the programmer needs rather than the business intelligence needs of the organisation.

The proliferation of online data, is not just limited to the Web, but has also occurred at the enterprise. Present database management has to cater to data from multiple external data sources such as customers, GPS, mobile devices, the general public, point-of-sale devices, and sensor data and so on apart from simple in-house data entry feeds. There are new kinds of data "Big Data", such as Web pages, digitised content such as books and records, music, videos, photos, satellite images, scientific data, messages, tweets and sensor data — each with different data-processing requirements [8][9].

Implementations of RDBMS cater to enterprise-centric workloads such as OLTP/OLAP, data warehousing, decision-support regimes etc. However, Big Data has ushered in a whole new set of data-centric workloads, such as Web search, massively multi-player online games, online message systems like Twitter, sensor networks, social network analysis, media streaming, photo processing, etc

A popular columnar database that offers state-of-the-art analytical capabilities is Vertica. It is based on CStore, a column-oriented academic database research project described in the paper [10]. These organise data from the same attribute as columns of values, as opposed to storing it as rows on disk. This results in large I/O savings in analytical and data-warehousing type of data retrieval that largely accesses a set of columns. Columnar databases are relational, and support ACID semantics, as well as provide SQL support. The data management needs of all the data-types mentioned above cannot be met by traditional database architectures. The opinion widely shared by many in the database industry is that, “The real database revolution driven by Big Data and the Cloud is just around the corner.”[9]

3.1 Databases for cloud

Whether the organizations are assembling, managing or developing on a cloud computing platform, they need a cloud compatible database. Shared-nothing databases require data partitioning, which is structurally incompatible with dynamic scalability, a core foundation of cloud computing. The shared-disk database architecture, on the other hand, does support elastic scalability. It also supports other cloud objectives such as lower costs for hardware, maintenance, tuning and support. It delivers high-availability in support of Service Level Agreements (SLAs) [11]. Various SQL operations, such as joins, cannot be implemented at the database layer, since the database is partitioned; they need to be implemented in the application middleware layer. Therefore, supporting both RDBMS and distributed databases, which can scale to the needs of Big Data and the Cloud, have conflicting requirement [12].

The database available for cloud is either SQL-based or NoSQL data model. Amazon Relational Database Service (MySQL), Microsoft SQL Azure (MS SQL), Heroku Postgre SQL, Xeround Cloud Database (MySQL) and Enterprise DB (Postgres SQL) are Database as a Service (DBaaS) based on SQL Data Model.

Amazon Dynamo DB, Amazon Simple DB, Database.com by SalesForce and Google App Engine Datastore are Database as a Service (DBaaS) based on NoSQL Data Model. NoSQL databases support high availability, scalability and low latency needs. They also need to provide state-of-the-art analytics, which can power business intelligence while being elastic to fit the cloud environment. While the NoSQL movement has helped to answer the initial needs of massive data sets of Web 2.0, traditional OLTP and OLAP enterprise applications still depend on RDBMS. For example, Amazon Web Services provides two database services as part of its cloud offering, SimpleDB which is a NoSQL key-value store, and Amazon Relational Database Service which is an SQL-based database service with a MySQL interface. Therefore, both NoSQL and traditional RDBMS would continue to coexist for the next decade.

Traditional row-oriented relational databases are optimized for writes and random reads; columnar databases excel in queries across a large number of records where the values in a small number of columns are accessed. Columnar databases also leverage the commonality of content in the columns to drive substantial compression. A hybrid database seeks to achieve most of the performance advantages of both relational and columnar. Recently published benchmarks suggest that in some circumstances, columnar and/or hybrid technology can achieve order-of-magnitude increases in performance in certain types of large-scale (multi-terabyte) data warehousing.

4 CONCLUSIONS

- Cloud Database services or Database as a service offer organizations new and unique ways to offer, use and manage database services. There are a variety of issues, considerations and choices that organizations must understand before embarking on a DBaaS project.
- Apart from designing scalable, elastic and autonomic multitenant database systems yet another important challenge is ensuring the security and privacy of the data outsourced to the cloud.
- One interesting research question is how to balance the tradeoffs between advantages and disadvantages of relational and columnar databases. The bottom line is that there is both interesting research and engineering work to be done in creating hybrid relational and columnar databases.

5 REFERENCES

[1] "The NIST Definition of Cloud Computing". National Institute of Science and Technology. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Retrieved 24 July 2011

[2] http://www.oracle.com/solutions/business_intelligence/exadata.html

[3] D. Gupta, P. Jalote, and G. Barua.1996, A formal framework for on-line software version change. IEEE Transactions on Software Engineering, 22(2):120–131, 1996.

[4] Tarandeep Singh, Parvinder S. Sandhu, 2011, Cloud computing Databases: Latest Trends and Architectural Concepts, World Aademy of Science, engineering and Technology 73 2011 pg 1043-1045

[5] Oracle White Paper on Enterprise Architecture, 2011 Database as a Service: Reference Architecture – An Overview.

[6] I. Sommerville. Software Engineering. Addison-Wesley, 6th Edition, 2001.

[7] D Agrawal, El Abbadi, S Antony S.Das. Data Management Challenges in Cloud computing Infrastructures, In DNIS, pages 1-10, 2010

[8] http://databasemanagement.wikia.com/wiki/cloud_Database

[9] Saravanan Chidambaram and Sandya Mannarswamy, “Databases in the Era of Cloud Computing and Big Data” on May 1, 2011 in Features, Open gurus, Overview, Technology

[10] Vertica. Performance O-Demand with Vertica Analytic Database for the Cloud. http://wwww.vertica.com/cloud.

[11] http://wiki.cloudcommunity.org/wiki/Cloudcomputing:Incidents_Database

[12] D Agarwal , S Das, El Abbadi , 2011, Big Data and Cloud Computing: Current State and Future Opportunities, EDBT 2011, Uppsala, Sweden, ACM 978-1-4503-0528-0/1/0003