Research Inventy: International Journal Of Engineering And Science
Vol.5, Issue 4 (April 2015), PP 23-29
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Challenges Management and Opportunities of Cloud DBA

*Dr.Osamah Al-rababah, Dr.khaled Alshraideh and **syhayb abo alshamat
**College of Computing and Information Technology in Alkamil*
University of Jeddah , Jeddah , Saudi Arabia
*** College of Computing and Information Technology -KAU*

ABSTRACT—*This paper discuss the challenges and opportunities of Cloud computing DBA, beginning with building the database in the cloud by comparing between available architectures and choose the suitable one for the cloud. also we compare between traditional relational database (RDBMS) and NoSQL Database and discuss the most important parameters which is important for cloud DBA. we discuss the limitations and opportunities of deploying data management issues on these emerging cloud computing platforms (e.g., Amazon Web Services). We speculate that large scale data analysis tasks, decision support systems, and application specific data marts are more likely to take advantage of cloud computing platforms than operational, transactional database systems. We thus conclude the need for a new database administration system designed especially for cloud computing environments.*
KEYWORDS— *Cloud DBA, DBAMS, NOSQL, Shared-disk, Shared-nothing.*

I. INTRODUCTION

The need for Cloud computing applications & database is coming important for accessing computing resources anywhere. Building and managing database in cloud is one of the most challenges face cloud computing and database administration these days, users should be able to access virtual machines on which they can install and run arbitrary software, including database systems. Users can also deploy database appliances on the clouds, which are virtual machines with pre-installed pre-configured database systems. Cloud has brought in new DBA skill requirements and hence the role of DBA is as important as before in a cloud managed environment. DBA with skills in replication, backup, recovery and clustering have been in high demand over the years due to the way the enterprises have taken care all aspects of the database administration. However, the advent of the Cloud has shifted much of the DBA duties from the Cloud Consumer (Enterprises) to the Cloud Providers (EC2, Azure and more).

II. CLOUD DATABASE ADMINISTRATION

There are two terms that describe the data administration in the cloud, one of them is Data as a service (DaaS) and the other Database as a service (DBaaS). The difference between them is on the basis of how data is stored and managed. Cloud storage enables users to store their data virtually on servers. there are already several example working in this section like Dropbox, iCloud …etc [1]. In Database as a Service, application owners do not have to install and maintain the database by their own. Instead, the database service provider takes responsibility for installing and maintaining the database, and application owners pay according to their usage [2]. For example, Amazon Web Services provides two database services as part of its cloud offering, Simple database which is a NoSQL key-value store, and Amazon Relational Database Service which is an SQL-based database service with a MySQL interface[3]. Cloud Database cannot work without data management services. So we need to depend on Dbaas complete database functionality and allows users to access and store their database at remote disks anytime from any place through Internet.

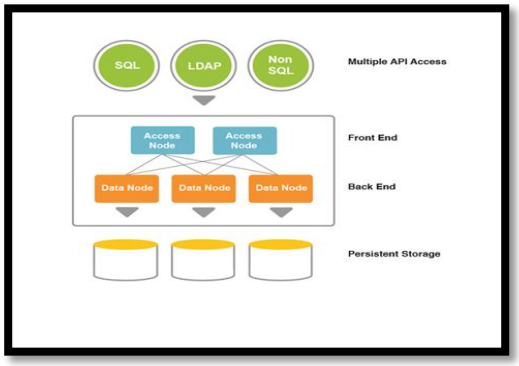


Figure (1) multiple API Access with (SQL,LDAP,NoSQL) Databases

1- Requirements for DBAAS

A- User Requirements

- simple API, with near-zero configuration and administration (i.e., no tuning) .figure(1).
- high-performance (e.g., throughput, latency, scalability)
- high availability and reliability (e.g., hot standby, backup)
- easy access to advanced features (e.g., snapshot, analytics, time travel)

B- Provider Requirements

- meet user service level agreement (potentially under dynamic workloads)
- limit HW and power costs (e.g., intense multiplexing)
- limit administration costs (e.g., personnel costs)

C- Public Cloud Requirements

- pricing scheme: cheap, predictable and proportional to actual usage (elasticity)
- security and privacy guarantees
- low-latency (relevant for OLTP and Web application

III. DATABASE ADMINISTRATION FOR A CLOUD

Since cloud infrastructure will be in the center of all systems will concern the integrity and operations of databases. Therefore the organization of Database-as-a-Service (DBaaS) warrants special attention as a separate software discipline. Some Software are now available which simplifies databases administration. This enables oversight of the total environment of hundreds of disk files. Customers can then access a pool of disk drives without intervention by intermediary personnel. The objective is to assure a high level of security, flexibility, control and assurance that the data will comply with policy guidelines. DBaaS automates common database operations by presenting to developers and to operators comprehensive displays, which significantly reduce the management overhead as well as increases the efficiency of memory management. DBaaS manages database provisioning, back up of records and the fail-over cloning for development and testing. The primary objective of DBaaS is to reduce database sprawl through virtualized pooling of disk capacity while providing for sufficient redundancy to assure the continuity of operations. DBaaS installs automated methods for protecting operations and for providing adequate buffer space that assures low latency response time.

DBaaS will reduce database operating and capital expenses by extending virtualization to all data storage components so that capacity is readily scalable. It provides a central view of potential problems while monitoring performance that assures high transaction processing availability. A major concern for moving databases to the cloud is the potential interference caused by multiple databases sharing the same pool of resources, the so-called “noisy neighbor” problem. DBaaS offers isolation at the individual user level, thus eliminating “noisy neighbor” situations. DBaaS restricts data base access to authorized users. Assigned privileges enable IT administrators to control which users can take what actions.

IV. NOSQL DATABASE

Since we know enough information about Relational database system we will present some information about NoSQL database and compare between the two systems later. NoSQL is abbreviation of "not only SQL" is a broad class of database management systems identified by non-adherence to the widely used relational database management system model [4]. NoSQL databases are not built primarily on tables, and generally do not use SQL for data manipulation. NoSQL database systems are often highly optimized for retrieve and append operations and often offer little functionality beyond record storage (e.g. key–value stores). The reduced run-time flexibility compared to full SQL systems is compensated by marked gains in scalability and performance for certain data models. In short, NoSQL database management systems are useful when working with a huge quantity of data when the data's nature does not require a relational model. The data can be structured, but NoSQL is used when what really matters is the ability to store and retrieve great quantities of data, not the relationships between the elements [5]. Usage examples might be to store millions of key–value pairs in one or a few associative arrays or to store millions of data records. This organization is particularly useful for statistical or real-time analyses of growing lists of elements (such as Twitter posts or the Internet server logs from a large group of users). NoSQL databases have the ability to join data across different columns. By removing this great feature of relational databases, they dramatically simplify the underlying implementation. Many of these databases cut corners on what’s called durability. Which make NoSQL databases don’t always flush data to permanent storage. must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified [5].

Advantages of NoSQL database

-NoSQL databases generally process data faster than relational databases.

-NoSQL databases are also often faster because their data models are simpler.

-Major NoSQL systems are flexible enough to better enable developers to use the applications in ways that meet their needs.

V. CLOUD DBA CHALLENGES

1- Data partitioning

Recently we didn't use to worry about database size since Earlier DBA have big servers and databases with unlimited storage at their disposal and most times the databases have been exclusively owned by enterprise vertical domains. The new limitations in the database storage and the need to store multiple tenants require DBA to think of different ways of partitioning data across multiple databases, which will go a long way toward an efficient data organization for Cloud.

2-Monitoring

much more monitoring capabilities need to be brought in by the DBA to make sure how the databases behave to the requests and how they can be managed On-demand instances, since much of the instances are virtual server based and the IP Addresses that connect are virtual and may get changed over time, so traditional DBA needs to know new tools and new concepts behind monitoring a Cloud Database.

3- Cloud Databases architecture

In our research we need to define the suitable cloud database architecture since there are many types of databases architectures to use. have to compare between the most popular types of cloud database architecture as follows:

A- Shared-nothing Storage Architecture

Shared-nothing Storage architecture mean to partition data into independent sets. The data sets resulted from partition are physically stored on different database servers. Each server processes and maintains its portion of the database exclusively which makes the shared-nothing databases architecture scalable. Due to inherent scalability, applications designed to work on shared-nothing storage architecture are suitable for Cloud. But data partitioning used in this architecture does not work well with cloud. It is very difficult to virtualize a shared-nothing database as it becomes very complex and difficult to maintain due to data partitioning. It needs a device in the middle to route database requests to the suitable server. As more servers are added, data has to be repartitioned[6]. Data partitioning should be done very carefully, otherwise data passing from one machine to the other machine for processing and joining will become difficult. More data passing means more latency and network bandwidth bottlenecks. These disadvantages reduce database performance badly.

Figure(2) show the difference between virtual classic storage and shared-nothing storage architecture .

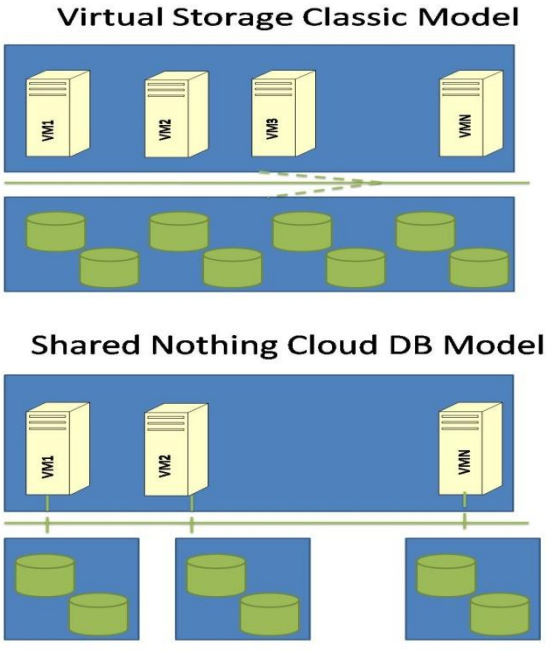
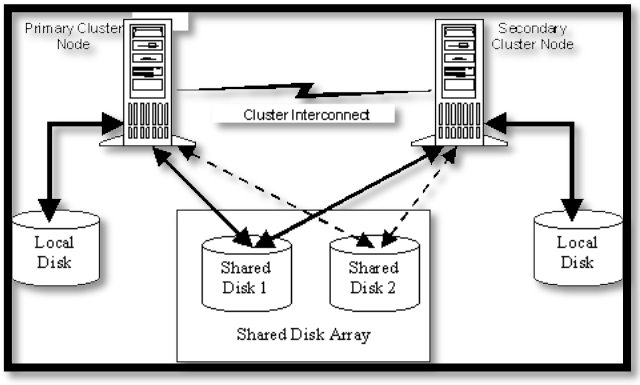


Figure (2) difference between classic storage and shared nothing DB

B-Shared-disk Database architecture:

Database architecture called shared-disk, which eliminates the need to partition data, will be ideal for cloud databases. Shared-disk databases allow clusters of low-cost servers to use a single collection of data, typically served up by a Storage Area Network or Network Attached Storage . All of the data is available to all of the servers, there is no partitioning of the data. As a result, if we were using two servers, and the query takes 0.5 seconds, we can dynamically add another server and the same query might now take 0.35 seconds. In other words, shared-disk databases support elastic scalability[7]. Figure (3) illustrate the shared-disk architecture.



Figure(3) shared-disk architecture

The shared-disk DBMS architecture has other important advantages in addition to elastic scalability that make it very appealing for deployment in the cloud we conclude them as follows:

Reduce the number of servers: Since shared-nothing databases break the data into distinct pieces, it is not sufficient to have a single server for each data set, you need a back-up in case the first one fails. This is called a master-slave configuration. In other words, you must duplicate your server infrastructure. Shared-disk is a master-master configuration, so each node provides fail-over for the other nodes. This reduces the number of servers required by half when using a shared-disk database.

Extend servers life span: In a shared-nothing database, each server must be run at low CPU utilization in order to be able to accommodate spikes in usage for that server's data. This means that you are buying large (expensive) servers to handle the peaks. Shared-disk, on the other hand, spreads these usage spikes across the entire cluster. As a result, each system can be run at a higher CPU utilization. This means that with a shared-disk database you can purchase lower-cost commodity servers instead of paying a large premium for high-end computers. This also extends the lifespan of existing servers, since they needn't deliver cutting-edge performance.

Scale-in: The scale-in1 model enables cloud providers to allocate and bill customers on the basis of how many instances of a database are being run on a multi-core machine. Scale-in enables you to launch one instance of MySQL per CPU core. For example, a 32-core machine could support a cluster-in-a-box of 32 instances of MySQL. Simplified maintenance/upgrade process: Servers that are part of a shared-disk database can be upgraded individually, while the cluster remains online. You can selectively take nodes out of service, upgrade them, and put them back in service while the other nodes continue to operate. You cannot do this with a shared-nothing database because each individual node owns a specific piece of data. Take out one server in a shared-nothing database and the entire cluster must be shut down.

High-availability: Because the nodes in a shared-disk database are completely interchangeable, you can lose nodes and your performance may degrade, but the system keeps operating. If a shared-nothing database loses a server the system goes down until you manually promote a slave to the master role. In addition, each time you (re)partition the database, you must take the system down. In other words, shared-nothing involves more scheduled and unscheduled downtime than shared-disk systems.

Reduced partitioning and tuning services: In a shared-nothing cloud database, the data must be partitioned. While it is fairly straightforward to simply split the data across servers, thoughtfully partitioning the data to minimize the traffic between nodes in the cluster—also known as function or data shipping—requires a great deal of ongoing analysis and tuning. Attempting to accomplish this in a static shared-nothing cluster is a significant challenge, but attempting to do so with a dynamically scaling database cluster is a Sisyphean task.

Reduced support costs: One of the benefits of cloud databases is that they shift much of the low-level DBA functions to experts who are managing the databases in a centralized manner for all of the users. However, tuning a shared-nothing database requires the coordinated involvement of both the DBA and the application programmer. This significantly increases support costs. Shared-disk databases cleanly separate the functions of the DBA and the application developer, which is ideal for cloud databases. Shared-disk databases also provide seamless load-balancing, further reducing support costs in a cloud environment.

4-Cloud Database Deployment

Deploying Database to the Cloud face several challenges we discuss them as follows:

A-Localization:

Which mean to give distinct address or name to an identity we need to give the virtual machine a MAC address, an IP address, and a host name. We also need to adapt (or localize) the database instance running on this virtual machine to the virtual machine's new identity. For example, some database systems require every database instance to have a unique name, which is sometimes based on the host name or IP address. The underlying operating system and networking infrastructure may help with issues such as assigning IP addresses, but there is typically little support for localizing the database instance. The specific localization required varies from database system to database system, which increases the effort required for creating database appliances.

B-Routing:

We need to be able to route application request to the suitable virtual machine. This includes the IP-level routing of packets to the virtual machine, but it also includes making sure that database requests are routed to the correct port and not blocked by any firewall, that the display is routed back to the client console if needed, that I/O requests are routed to the correct virtual storage device if the "compute" machines of the IaaS cloud are different from the storage machines, and so on.

C-Authentication:

The virtual machine must be aware of the credentials of all clients that need to connect to it, independent of where it is run in the cloud.

5-Balancing between Virtual machine and physical ones

One of the challenges face the cloud DBA is to run a user's virtual machine on any available physical machine. Connecting between virtual machines and physical machines can have a significant impact on performance. So we need to decide how many virtual machines we can run on each physical machine [8]. The cloud provider would like to use the minimum number of physical machines, and run more Virtual machine on the physical one degrades the performance of these Virtual machines. So the challenge is to balance these conflicting objectives: minimizing the number of physical machines used while still has acceptable performance for users.

6-Resource allocation:

Another challenge is to choose the best way of partition the resources of each physical machine between the virtual machines that are running on it. Most Virtual machines provide tools for controlling the way that physical resources are allocated. Scheduling parameters can be used to distribute the total physical CPU capacity between the Virtual machines. Some other tools can be used to control the amount of physical memory that is available to each Virtual machine. To reach the best performance, we need to put in consideration the characteristics of the application running in the virtual machine so that we can allocate resources where they will provide the maximum benefit [9]. So now we need to minimize the cloud resources required while still having good performance for the database appliance. A database system is usually belonging to a stack of multi-layer software that is used to serve application requests. As we work on end-to-end application there is no need to care how much of performance budget is available to the database system and how much is available to other layers of the software stack. Since an application request can make a number of database requests, and these database requests can vary in complexity regarding to the SQL statements being executed. Therefore cloud environment requires developing practical and intuitive ways of expressing database service level objectives.

VI. CLOUD DBA LIMITATION AND FUTURE OPPORTUNITIES

In addition to challenges cloud DBA face we will explore the advantages and disadvantages of deploying database systems in the cloud from Database administration view. We look at how the typical properties of available cloud computing platforms affect the choice of data management applications to deploy in the cloud in the cloud, showing why currently available systems are not ideally-suited for cloud deployment, and arguing that there is a need for a newly designed DBA, architected especially for cloud computing platforms in the virtual environment. We will decide which data management applications are best suited for deployment on top of cloud computing infrastructure.

VII. DATABASE ADMINISTRATION TOOLS IN THE CLOUD

A-Transactional DBA

Transactional applications typically rely on the ACID guarantees (atomicity, consistency, isolation, durability) that databases provide, and tend to be fairly write-intensive, systems like airline reservation, online e-commerce, and supply chain management applications use this kind of application, however we expect that transactional data management applications are not likely to be deployed in the cloud, at least in the near future, for the following reasons:

- Transactional data management systems do not typically use a shared-nothing database architecture[10]
- It is hard to maintain ACID guarantees in the face of data replication over large geographic distances.
- There are enormous risks in storing transactional data on an entrusted host

B- Database analytical

Analytical tools are applications which query a data store for use in business planning, problem solving, and decision support. Historical data along with data from multiple operational databases are all typically involved in the analysis. Consequently, the scale of analytical data management systems is generally larger than transactional systems, Furthermore, analytical systems tend to be read-only with occasional batch inserts. Analytical data management account for 27% of data market and is growing at a rate of 10.3% annually [11]. We suggest that systems are well-suited to run in a cloud environment, and will be among the first data management applications to be deployed in the cloud, for the following reasons:

- Shared-nothing architecture is a good match for analytical data management [12].
- ACID guarantees are typically not needed
- Particularly sensitive data can often be left out of the analysis

VIII. CONCLUSIONS

Here we can present a comparison between the most important parameters in cloud DBA:

Shared-Disk:

In this architecture the single node will coordinate the changes itself, so there is no delay of waiting for the slowest node. This results in much faster performance comparing with shared-nothing.

Data Access Speeds

Shared-disk systems tolerate a small amount of latency in data access from storage area network or network attached storage, while shared-nothing databases access data from a local disk at faster bus speeds. Given high-speed interconnects such as Fiber Channel and gigabit Ethernet, the latency difference between a local disk and a shared disk can be trivial.

1- Comparing shared-disk database with shared nothing storage architectures:

Architecture	Shared-nothing	Shared-disk
Partitioning	Yes	No
Distributed	Yes	Yes
Scalability	Yes	Yes
Reliability	No	Yes
Online transaction processing	No	Yes
Analytical	Yes	Yes
Maintenance Cost	High	Low
Useful for cloud	Yes	Yes
Initial Set-up Effort	Partitioning & routing tables	No extra effort
Evolving Performance	Re-partitioning may be required to handle evolving usage	Adapts to evolving requirements via load balancing
Load Balancing	Fixed load balancing based upon the partitioning scheme	Dynamic load balancing
Parallel Processing Across Nodes	Partitioning trade-offs, parallel vs. unified view, problematic	Processes can be parallelized without additional effort
Total Cost of Ownership	Higher ongoing/maintenance costs for partitioning, tuning, slave replication, etc.	Software can be more expensive, but set-up and maintenance costs are lower

2- Comparing RDBMS and NoSQL databases

Database	RDBMS	NoSQL
Independency	No	Yes
Centralization	Yes	No
Backup	Static	Dynamic
Scalability	Difficult	Easy
Query data	SQL	API
Online transaction	Online transaction	Support Web2 application
Example	ORACLE, MySQL, SQL Server etc.	Amazon SimpleDB, Yahoo's PNUTS, CouchDB etc.

IX. RECOMMENDATIONS

Cloud computing make the DBA service need to be in a new form. Apart from the skills that traditional database administrators handle, DBA services need to include more skills pertaining to cloud in their portfolio. DBA service providers require new skills such as improving NOSQL database usage and skills, capacity planning in terms of on-demand usage cost, Multi-tenancy and data partitioning on Cloud, new tools and concepts behind monitoring a Cloud Database, and data in and out, between cloud and data centers, techniques. Cloud is a technology to safe-guard the databases at minimum costs. It also encourages database expansion to no limits. We recommend use NoSQL Database instead of relational database and to build cloud databases depending on shared-nothing architecture s which is designed for high availability and high performance for massive read and write operations.

X. SUMMARY

Cloud has brought in new DBA several requirements and hence there is several important parameters to care about. We explain the Database as service (DBAAS) term at first and present the requirement for users and providers to use this technique, then we choose NoSQL Database against traditional relational database to suit cloud requirements, after that we discuss the most important challenges face the cloud DBA like database portioning and monitoring , then we concentrate on database architecture and compare between shared-nothing and shared-disk architectures. We also address the localization , routing and authentication as challenges face the deployment of database to cloud.

REFERENCES

[1] Jiyi Wu et al, "Recent Advances in Cloud Storage", in Third International Symposium on Computer Science and Computational Technology(ISCSCT '10), Jiaozuo, P. R. hina, 14-15,August 2010, pp. 151-154.

[2] Klint Finley, "7 Cloud-Based Database Services", ReadWriteWeb, Retrieved 2011-11-9.
Database as a Service: Reference Architecture – An Overview, An Oracle White Paper on Enterprise Architecture September 2011 Daniel J. Abadi et al., "Column-oriented Database Systems",VLDB '09.

[3] Arpita Mathur et al., "Cloud Based Distributed Databases: The Future Ahead", International Journal on Computer Science and Engineering (IJCSE) Vol. 3, No. 6, 2011.

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

[5] Donald Kossmann, Tim Kraska, Simon Loesing, "An Evaluation of Alternative Architectures for Transaction Processing in the Cloud", SIGMOD'10, June 2010.

[6] Ahmed A. Soror, Ashraf Aboulnaga, and Kenneth Salem. Database virtualization: A new frontier for database tuning and physical design. In Proc. Workshop on Self-Managing Database Systems (SMDB), 2007.

[7] Pradeep Padala, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, andKenneth Salem. Adaptive control of virtualized resources in utility computing environments. In Proc. European Conf. on Computer Systems (EuroSys), 2007.

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

[9] D. Vesset. Worldwide data warehousing tools 2005 vendor shares. Technical Report 203229, IDC, August 2006.

[10] S.Madden, D. DeWitt, andM. Stonebraker. Database parallelismchoices greatly impact scalability. DatabaseColumn Blog. http://www.databasecolumn.com/2007/10/database-parallelism-choices.html.