

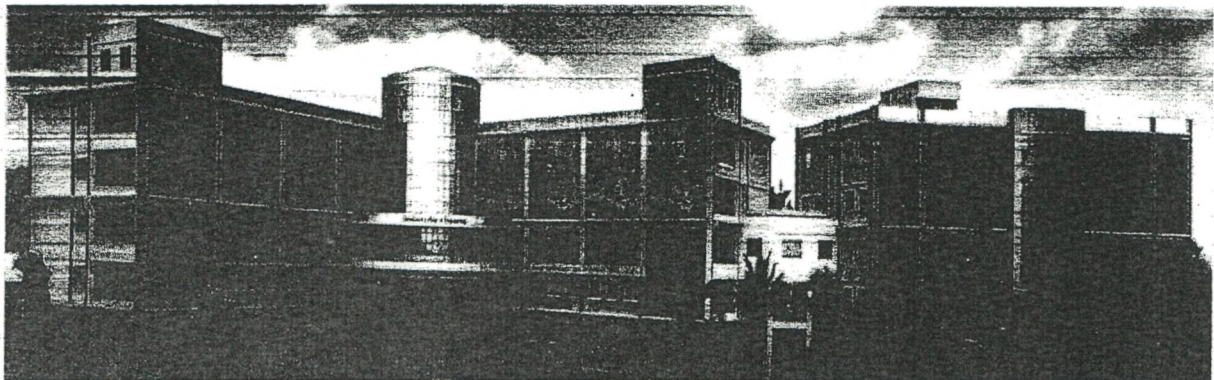
Proceedings of the 2nd National Conference

On 31st May, 2014

*"Sustainable Development through Global Business
and Technological Advancement"*



Organized By
**Department of PG Studies in
Commerce, Management and Computer Science**



Brindavan College

Accredited at the "A" Level by NAAC

Approved by AICTE, Recognized by Govt, Affiliated to Bangalore University and VTU
Dwarakanagar, Bagalur Main Road, Yelahanka, Bengaluru-560063, Karnataka, INDIA

Phone: +91 8028478388/ 8423, Fax: +91 80 28478766

www.brindavancollege.com

VRINDA PUBLICATIONS (P) LTD.

B-5, Ashish Complex (opp. Ahlcon Public School),
Mayur Vihar, Phase-I, Delhi 110091
Tel. : 22755315, 22755316 Fax: 22757220
E-mail— vrinda@ndf.vsnl.net.in
Visit us at : www.vrindaindia.com.

© Author

First Edition : 2015

ISBN: 978-81-8281-545-2

[PROCEEDINGS OF THE 2nd NATIONAL CONFERENCE ON: “*Sustainable Development through Global Business and Technological Advancement*”]

Typeset by: KUMUD PRINTOGRAPHICS—DELHI

Printed at: B.K. OFFSETS, DELHI—

4....

HUMAN RESOURCE MANAGEMENT

- | | | |
|---|--|-----|
| 1 | HR Automation – A Tool to Smooth Success Track for an Organization :
- <i>Prof. Anand Soni</i> | 91 |
| 2 | A Recipe for Business Success - Implementing Change in Organizational Culture :
- <i>Mr. Harris Saseendran, Ms. Shaista Banu Harris</i> | 96 |
| 3 | Innovative Practices in Employee Engagement :
- <i>Mr. Santhosh Kumar</i> | 104 |
| 4 | A Study On HR Ethical Practices: In Indian Unorganized Garment Companies :
- <i>Dr. Senthil Velumurugan & Ms. Malini.T.N</i> | 109 |
| 5 | User Entrepreneurship: An Avenue to Rescue the Dwindling Economy:
- <i>Dr. Sonia Selwin</i> | 113 |
| 6 | Organizational Behaviour and Analysis of Strategic HRM: Organizational Leadership
<i>Rama Krishna. V, Shashidhar.R, Muniraju.M.</i> | 119 |
| 7 | Impact of Social Networking on Emotional Intelligence of Gen-Y Students- A Study on Bangalore's Student Population :
- <i>Ms. Nitu Bose (Ghosh)</i> | 125 |
| 8 | Pioneering Employee Engagement in Teaching Fraternity :
- <i>Ms. Sudarkodi .P</i> | 129 |

INFORMATION TECHNOLOGY

- | | | |
|---|---|-----|
| 1 | Research Challenges in Cloud Databases :
- <i>Revina Rebecca.D and Dr. Elizabeth Shanthi.I</i> | 137 |
| 2 | Interacting with any Surface: Taking the Human Interface to the Next Level :
- <i>Mr.Rakshith S, Kumar, Ms.Aditi Singh, Mr. Naidu. R.Y</i> | 143 |
| 3 | Cyber Crimes :
- <i>Prof. Mohan.K</i> | 148 |
| 4 | Cloud Computing: Network/Security Issues and Countermeasures :
- <i>Mr. Shuaib Ahmed Shariff, and Ms. Bharati .B Sali</i> | 151 |
| 5 | Human Computer Interaction Technology :
- <i>Mr. Nandeesh Hugar</i> | 156 |
| 6 | A Study of Energy Efficiency in Wireless Data Communication on Mobile Cloud Environment
<i>Amarnath C.T.K and Dr. Muruganatham S.</i> | 160 |
| 7 | Comparison Between the Cloud Computing and Green Cloud Computing Architecture :
- <i>Srinivasa Rao.T, Kalaivani Y .S, and Prabhu.G</i> | 166 |
| 8 | Digital Library Software: Comparative Study of Greenstone And E-Shelf :
- <i>Ms. Veena V K And Mr. Swamidas Mitra</i> | 171 |

RESEARCH CHALLENGES IN CLOUD DATABASES

Revina Rebecca*,

Associate Professor,
Dept of MCA, REVA
University, Bangalore,

Dr.I.Elizabeth Shanthy**

Associate Professor, Dept of
computer science, Avinashilingam
University for women, Coimbatore

Abstract

Cloud computing is the buzz word of today's IT industry. Databases and storage are an integral part of cloud computing. Cloud computing is slowly pushing out the traditional database technologies which ruled the IT industry for decades, as these technologies is finding it to difficult to cope with the expectation of the cloud technology. The traditional Relational Model(RDBMS) can no longer be the "One Size fits all" solution because of the various limitations when the databases is moved to the cloud. "Traditional database systems do not have the scale, flexibility and power to store, process, analyze, and mine huge volumes of diverse and distributed data in real time. Due to this the IT industry came with many NoSQL (Not Only SQL) and NewSQL databases with various needs in mind. As data variety, velocity, veracity, and volume are fast-growing, different types of database systems such as graph, columnar, text, key-value, etc. (all these are termed as NoSQL) are being explored. Hadoop is a new method for parallel data crunching. NoSQL and Hadoop need a better infrastructure, which is being offered by cloud infrastructure. That is why cloud databases have become hugely popular and pervasive. This paper aims in discussing the various advancements in the recent past in cloud databases and tries to find out the various alternatives available in cloud computing with regard to cloud databases.

Keywords-Cloud Computing, cloud databases, NoSQL databases

Introduction:

The IT industry is in the verge of the next paradigm shift, from client/server computing to cloud computing. The rapid development of hardware has led to a cheaper processing power, storage technologies and availability of computer resource. Also the internet has become very cheaper than ever before. This technological trend has enabled a new computing model to be realized called Cloud computing. This model allows the users to provision resources(CPU and Storage) over the internet without much user expertise. The emergence of Cloud computing has a tremendous impact on

Information Technology (IT) industry. Leading players being Google, Amazon and Microsoft strive to provide more powerful, reliable and cost- efficient cloud platforms. Many business enterprises are trying to gain benefit from this new paradigm by reshaping their business models to. Indeed, cloud computing provides several compelling features that makes the future business and database system to move to the cloud, the reasons as shown below in a database perspective.

Minimal Capital investment: Cloud computing provides a pricing model which converts the initial capital expenses to Operational Expenses. The user need not buy an expensive Database system, a high

end server and also there is no need to hire expert staff (DBA) to handle the Data and Database as the Database system and other infrastructures rented from the service provider.

Lowering operating cost: The resources can be rapidly allocated on demand when needed in a cloud environment, where the user pays only for what he uses. Hence, a Database service provider no longer needs to provision High end Server capacities according to the peak load. The user allocates extra storage space or server space during peak load and de-allocates when not needed. This helps even in lowering the operating costs, by releasing the unused resources.

Highly scalable: The cloud provider makes available any amount of resource needed from data centers and makes them easily accessible. A Database service provider is expected to scale up to handle service demands this model is sometimes called surge computing.

Ubiquitous network access: Database Services hosted in the cloud are easily accessible through a variety of devices with Internet connections. These devices not only include desktop and laptop computers, but also cell phones and PDAs.

Reducing business risks and maintenance expenses: By outsourcing the database service to the clouds, the user can rely on the DaaS(Database as a Service)provider and thereby have lesser business risks as the providers often have better expertise and are better equipped for managing these risks. In addition, a service provider can cut down the hardware maintenance and the staff training costs.

The above mentioned advantages make Cloud an inevitable choice for storing Data. A recent Database Trends Survey by embarcadero technologies reveal the fact the databases in the cloud will have a greater impact in the coming years. The future database is predicted to be in the cloud. Data and database system will be delivered as a service over any network to a larger section of people in a variety of devices in a simplified and structured fashion. Due to the exponential growth of data variety, velocity, and volume, big data computing is fast emerging. Cloud is the best infrastructure for accomplishing crunching growing data mountains.

Business intelligence in the enhanced data ecosystem (comprising data from social websites, machine-generated data, people created data, etc.) needs clouds for real-time business analytics where it is possible to process huge amount of data.

DATABASE CLASSIFICATION FOR THE CLOUD

The databases in the cloud can be classified into two broad categories i) Relational Cloud databases ii) Non-relational or NoSQL-databases.

Relational Cloud Databases

The traditional Relational databases such as MySQL, Oracle, and IBM/DB2 are made available in the cloud, and they are called cloud enabled. Microsoft azure is a cloud native relational database developed exclusively for the cloud. Relational model requires that the data should be structured

Non-relational (NoSQL databases)

The requirements with regard to cloud computing and the various limitations of the traditional RDBMS has led to the development of horizontal scalable, distributed, non- relational data stores, called NoSQL(Not only SQL) databases like Google's Bigtable, Amazon's Dynamo, Facebook's Cassandra, MongoDB etc. These databases allow to store unstructured and semi-structured data like documents, images etc. There are many NoSQL databases springing up like mushrooms. Each has its own application and use. The Non-relational databases can be broadly classified into the following categories.

- a) Document Databases
- b) Key/Value databases
- c) Big Table databases
- d) Graph Databases, e) Columnar Databases

Characteristics of NoSQL Databases

Most of the NoSQL Databases do satisfy the requirements

Of cloud computing. Most of the NoSQL databases are open source and they are built exclusively for the cloud environment. They are highly manageable, scalable, and fault-tolerant. Most of the NoSQL database, but these databases tend to have most or all of the following characteristics:

Schema-less: "Tables" don't have a pre-defined schema. Records have a variable number of fields that can vary from record to record. Record contents and semantics are enforced by applications.
Shared nothing architecture: Instead of using a common storage pool each server uses ~~only its own~~ local storage. This allows storage to be accessed at local disk speeds instead of network speeds, and it allows capacity to be increased by adding more nodes.

Elasticity: Both storage and server capacity can be added on-the-fly by merely adding more servers. No downtime is required. When a new node is added, the database begins giving it something to do and requests to fulfill.

Sharing: Instead of viewing the storage as a monolithic space, records are partitioned into shards. Usually, a shard is small enough to be managed by a single server, though shards are usually replicated. Sharing can be automatic

(e.g., an existing shard splits when it gets too big), or applications can assist in data sharing by assigning each

Record a partition ID.

BASE instead of ACID: NoSQL Databases emphasize performance and availability. This requires prioritizing the components of the CAP theorem that makes ACID transactions questionable.

Cloud database requirements and limitations

When it comes to databases i.e. Database as a Service (DaaS), the cloud provider is expected to provide something more along with the other cloud requirements mentioned in the introduction part of this paper. For example the provider has to provide ACID (Atomicity, Consistency, Isolation, and Durability) properties for sensitive transactions like banking, online reservations, Ecommerce etc. The wish list for a DaaS is more when compared with other delivery models. Every property of a conventional RDBMS is expected to be satisfied by a cloud database as well, but when the data is moved to the cloud there are many limitations which prevent to satisfy those demands. Even though the requirements of cloud Databases are many, we restrict our focus on the limitations with respect to the three important aspects, they are listed below.

- **Dynamic Scalability:** This is the ability to scale dynamically, i.e. databases that can be dynamically expanded. That is, additional servers and/or storage can be added without taking the DBMS software down. This is the ability to scale dynamically, say from 100 users to 1 million users instantaneously
- Transaction processing

- Ensure secure transactions.
- Multi-Tenancy

This is the ability to share a single instance of the virtualized resource with many users; The Cloud computing environment should and must support multi-tenancy for cost effectiveness.

Now we will discuss the limitations with respect to the conventional i) RDBMS ii) NoSQL databases.

Limitations of relational databases in the cloud

Most of the commercially available RDBMS fail to meet the needs of cloud computing. There are many limitations to be faced when an RDBMS operate in the cloud. The traditional relational database will be a bad fit in cloud environments. There are problems in both, whether it may be a cloud enabled RDBMS or cloud native RDBMS developed exclusively for the cloud (MS Azure), the limitations of a relational database with respect to cloud does not diminish. MS Azure does not support distributed transactions. There are other limitations as well. Now we will discuss the limitations with respect to the three major issues mentioned below

A. Dynamic Scalability: To meet the challenges laid by the cloud, the Database applications can be deployed to a virtualized environment. But for dynamic scaling it is necessary that data must be replicated and that requires data to be copied or moved to the new node. Since this data shipping is a time-consuming and expensive process, relational databases are unable to be dynamically and efficiently provisioned on demand. Cloud computing requires scaling up instantaneously which is achieved by allocating additional virtual machines. Scaling can be achieved through

- i) Shared-nothing architecture ii) Shared-disk architecture.

i) Shared-nothing architecture

Shared nothing architecture is best suitable for the cloud as resources are added on the fly when needed. Web servers and applications scale out using shared-nothing clustering and hence resources can be added or removed from these layers simply by starting or stopping members of the cluster, but it's much harder to cluster databases. Shared-nothing architecture works on the principle that each node in the cluster is the sole owner of the data on that node. Each node literally shares no data with the other nodes of the cluster, hence the term shared-nothing. When data is moved from a single server to multiple servers in a shared-nothing cluster, the data is divided across the servers. This process of splitting the data across servers is called partitioning. Data can be partitioned vertically or horizontally. In a shared-nothing-database cluster, data must either be replicated across the cluster members, or partitioned between them. In either case, adding a machine to the cluster requires data to be copied or moved to the new node. Since this data shipping is a time consuming and expensive process, shared nothing databases are unable to be dynamically and quickly provisioned on demand.

ii) Shared-Disk Architecture

In Shared disk architecture, also known as shared-everything, works on the principle that an on array of disks is present, that holds all of the data in that database. Each server or node in the cluster access on that single collection of data in real time. Oracle's Real Application Clusters (RAC) in particular has the potential to scale on demand, since an instance can be added to the cluster without the need to shift any data. RAC technology has legitimate advantages in this respect, but as yet no widely used RAC provisioning solution exists: instances are not created or removed based on demand but instead must be

provisioned manually. No other credible shared-disk database architecture is apparent. Unfortunately, most relational databases cannot smoothly scale up in this manner. Even though Oracle's RAC cluster database provides a database scale-out capability, RAC is also relatively high-maintenance, expensive and proprietary and hence less attractive to those trying to push down licensing and maintenance costs. Its licensing and administrative overhead currently make it an unattractive fit for zero-maintenance, pay-as-you-go cloud computing. The common MySQL "sharing" model and the use of distributed object caches such as memcached can be used to establish a large virtual database from many small database servers, but this solution is insufficiently elastic and automatic when demand scales rapidly. Also shared nothing architecture is suitable for the cloud.

B. Transaction processing

Typically, traditional relational databases (RDBMS) ensure ACID (Atomicity, Consistency, Isolation and Durability) for every transaction only when the transaction runs on a single hardware, but here in cloud computing since we scale horizontally using sharing i.e. partitioning data into multiple machines or optimistic locking i.e. allowing multiple transactions complete without affecting each other often has problems with normalization and thereby problems in satisfying ACID compliance as well. As databases become distributed across multiple hosts, maintaining ACID consistency becomes increasingly difficult. In a transaction that spans multiple independent databases; complex two-phase commit protocols must be employed. In the case of a truly clustered distributed database even more complex protocols are required, since the state of data in memory and the state of data in various transaction logs and data files must be maintained in a consistent state (cache fusion in Oracle RAC for instance). This makes RDBMS irrelevant and only applications that does not require ACID can be used in the Cloud.

NoSQL databases use only BASE (Basically Available Soft-state Eventually Consistent) Principle in transactions.

Basically Available: This means that most data is available most of the time. A failure could cause some data to not be available, but only catastrophic failures cause everything to be down.

Soft state: This means the DB provides a relaxed view of data in terms of consistency. Data Updates are not reflected immediately.

Eventually consistent: As data is replicated throughout the system, such as when new storage nodes are added, it is eventually copied to all applicable nodes. But there is no requirement for all nodes to have identical copies of any given data all the time. When a change is made to a data item, it will be visible eventually, but not immediately, in all locations where a copy of that data is maintained.

Elastic databases that emphasize BASE principles necessarily must give-up the traditional paradigm of ACID transactions: Atomicity, Consistency, Isolation, and Durability. However, various implementations can prioritize certain requirements such as availability over other requirements such as consistency. CAP theorem was proposed in a keynote by Eric Brewer in 2000 says "you cannot have all", you can have at most only two of the following characteristics:

- *Consistency:* This is essentially the same as "Atomicity" in ACID transactions. It is the principle that, for example, prevents two customers from both buying the last copy of a book in inventory. Atomicity guarantees that only one user can (a) lock the book object, (b) decrement its inventory count, and (c) add the book item to the user's shopping cart, all in one transaction.
- *Availability:* This means that the database, or more importantly the services that use it, are entirely available. Ensuring availability requires technologies such as replication and parallel processing so that changes in demand can be met while maintaining a minimum response time.

- *Partition Tolerance*: In a distributed network, “partitions” form when certain failures occur, such as a network cable being unplugged. This gives rise to conflicts and ambiguities as to where data lives and who has control over it. Partition tolerance is defined as: “No set of failures less than total network failure is allowed to cause the system to respond incorrectly”.

The CAP theorem requires the applications to understand their priorities carefully and optimize their distributed database solution carefully, for optimal survival. The industry is waiting for a better methodology other than CAP, because of its limitations.

C. Multitenancy

Some of the NoSQL databases do support multi-tenancy, but not all support multi-tenancy by itself, but we need to do some corrections and customization in order to make it multi-tenant. So it is possible to build multi-tenancy on these databases, based on the tenant or user requirements. But this too have some limitations. Some of the limitation faced may be a) Security and privacy of data b) Per tenant data customization c) Per tenant space utilization d) Disaster recovery e) Single-to-multi-tenant migration. Etc.

Conclusion:

This paper has discussed the complexity in choosing the right database for the cloud. It is obvious that the cloud database can be chosen based on the need and the type of application. If the data is too structured then it is better to go for relational databases in the cloud with some trade-offs. If the data is unstructured and if it is possible to forgo ACID compliance a NoSQL can be chosen with customization to the need. But that is not the solution, the solution lies where the industry is beckoning for a Blended and optimal solution, where it is neither completely relational, nor NoSQL, but somewhere between, where the goodness of the relational model is retained and also it satisfies the requirement of the cloud database environment.

References:

- [1] Daniel J. Abadi, Data Management in the Cloud: Limitations and opportunities, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering (2009)
- [2] Steve Bobrowski, Optimal Multitenant Designs for Cloud Applications, 2011 IEEE 4th International Conference on Cloud Computing.
- [3] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska. Building a Database on S3. In Proc. of SIGMOD, 2008
- [4] M. Stonebraker, S. R. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland. The end of an architectural era (it's time for, a complete rewrite). In VLDB, Vienna, Austria, 2007.
- [5] Divyakant Agrawal, Amr El Abbadi, Sudipto Das, and Aaron J. Elmore, Database Scalability, Elasticity, and Autonomy in the Cloud
- [6] <http://www.embarcadero.com/images/dm/technical-papers/database-survey-report.pdf>
- [7] <http://msdn.microsoft.com/en-us/library/aa479086.aspx>
- [8] <http://www.nosql-database.org> [9] <http://www.odbms.org/download/OnNoSQL.pdf> [10] http://wiki.toadforcloud.com/index.php/Survey_distributed_databases [11] <http://www.dbta.com/Articles/Editorial/Trends-and-Applications/Is-the-Next-DBMS-Revolution-Looming-51615.aspx>
