ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Comparative Study of Role Based Access Control in Cloud Databases and NoSQL Databases

Kapadia Gayatri S.
Assistant Professor
Master of Computer Application Department
Sarvajanik College of Engineering & Technology
Surat, Gujarat, India.

Morena Rustom D.
Professor
Department of Computer Science
Veer Narmad South Gujarat University,
Surat, Gujarat, India.

Abstract: Role Based Access Control (RBAC) is a mechanism to restrict access to unauthorized users for security of computer systems. Many organizations have adopted non-relational databases which are generally referred to as NoSQL databases. Security issues are the main concerns, as sensitive data is stored in both Cloud Databases and NoSQL Databases, especially access control to those data by unauthorized users. The working conditions and characteristics of RBAC of different Cloud Databases and NoSQL Databases vary. The main objective of this paper is to give clear-cut idea about RBAC in different Cloud Databases and NoSQL Databases. As a consequence, we have compared the RBACs of the most popular Cloud Databases Oracle 12c, Microsoft SQL Azure and NoSQL Databases MongoDB, Cassandra. We have presented the extensive comparative study of Role Based Access Control Model in Cloud Databases and NoSQL Databases under the specific parameters.

Keywords: RBAC; Security; Cloud Database; Oracle 12c; SQL Azure; NoSQL; MongoDB; Cassandra;

I. INTRODUCTION

In recent years, distributed web applications and cloud computing created the need to store a large amount of data in distributed databases that not only provide high availability and scalability but also 24X7 accessibility. Many organizations have adopted non-relational databases which are generally referred to as NoSQL databases. Different NoSQL Databases have different approaches. The main and common benefit of their approaches (compare to relational databases) is – they handle unstructured data like documents, e-mail, and multimedia data very efficiently.

Database Management Applications are preferable elements for deployment in the cloud. Cloud computing provides an access to very large pools of data and computational resources through access policies defined in RBAC of different databases rather than underlying frame work. [15]

Cloud Database Management System (CDBMS) is a distributed database that delivers computing as a service rather than product. In CDBMS, users can access resources and data available on the cloud through a wide range of interfaces. [16, 17]

Many of today's NoSQL applications are inspired by the Dynamo Technology developed at Amazon [18] and the Bigtable distributed storage system developed at Google. [19]

Oracle 12c offers real-time features and improvements to its users for supporting cloud implementations and manages many databases as one database. It introduces a new multitenant architecture that is implemented in two primary parts 1. A multitenant Container Databases (CDB) and 2. Pluggable Database (PDBs). Access Controls are enforced both at the database and application layers remain in effect. Oracle Label Security authorization can be used with Oracle Advanced Security Redaction and Oracle Database Vault commands rules for creating role-based access policy. [21]

The Windows Azure platform has one of the parts called SQL Azure which is a suite of Data Services, Web Services, infrastructure and Hosted Computing. SQL Azure provide the full relational database functionality of SQL server along with functionality as a cloud computing service, accommodated in Microsoft data centers around the world. There are a few aspects of Collaboration, Scale, Consolidation, Hosted Applications and Cost efficiency which makes SQL Azure be good fit.

MongoDB uses a document-oriented data model. [1] The MongoDB RBAC model is characterized by the concepts of privilege, data resource, action, role, and user. It regulates the access to document collections on the basis of the privileges granted to roles. We enhance this basic scheme introducing fine-grained purpose-based access control at the document level. MongoDB integrates a role-based access control (RBAC) [2] model which supports user and role management and enforces access control at the collection level. However, no support is provided for purpose-based policies.

Cassandra is a distributed storage system which manages the very large volume of structured data spread across many commodity servers which provide high availability service without single point failure. Cassandra supports dynamic control over data layout and format. It was originally designed for Facebook's Index Search Feature. [20]

II. LITERATURE SURVEY

We have surveyed several papers which are enlisted in reference, a few of them have been discussed here.

For security purpose, many organizations focus on permission and accessibility given to their employees to use resources and data in the cloud. Neither too many permissions nor too few permissions are reasonable. The former can expose an account and the later will affect the work's efficiency. Azure Role-Based Access Control (RBAC) helps to address this problem by offering fine-grained access management for Azure. In SQL Azure, the permissions are segregated among the team members. [40]

RajkumarBuyya, Chee Shin Yeo, and SrikumarVenugopal et al. [15] throw light on the current scenario of Market-Oriented Cloud Computing utilities which require Service Level Agreement (SLA) between service providers and users to access data and resources in the cloud. They present an idea of creating of global cloud exchange for trading service. Here, access control becomes the most important part of trading in global cloud exchange.

LiorOkman, Nurit Gal-Oz, YaronGonen, Ehud Gudes, JennyAbramov et al. [6] finely discuss the security issues with NoSQL Databases. How to tackle the access to data and resources in the cloud. The authors also extensively discussed parameters authentication, authorization, auditing, and injection attacks of RBAC in NoSQL Databases.

Indu Arora, Dr. Anu Gupta et al. [4] discuss that Cloud databases are better solutions for a large volume of data which is generated by web-based applications. Moreover, Cloud databases are ACID Compliant and capable enough to provide 24X7 access to the data and resources available on the cloud. Cloud databases are the best choice for the organizations which cannot afford to set up their own data centers and databases.

G. Decandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, et al. [18] present the various features of Dynamo Technology used in Cassandra like - highly availability, key-value storage, and always-on performance.

F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber [19] present the features of Bigtable Distributed Storage System used in Cassandra like flexibility and high-performance solution to scaling. We can say Cassandra combines the mentioned features of both Dynamo and BigTable technologies.

PratiyushGuleria et al. [10] presents RBAC of SQL Azure with various features like ACID Properties Support, Fault Tolerance, and Data Encryption which determines the working conditions of RBAC. SQL Azure detects both hardware and software failure.

Manjeet Singh et al. [22] presents different challenges in Cloud Databases and characteristics of Cloud databases. The author has clearly outlined thin line differences between transactional database and analytical database, which gives an idea to us that what parameters can have a diverse effect on RBAC of cloud databases.

III.COMPARISON OF ROLE BASED ACCESS CONTROL MODEL OF CLOUD DATABASES AND NOSQL DATABASES

Table 1 represents an experimental comparison of role-based access in cloud databases (Oracle 12c and SQL Azure) and NoSQL databases (MongoDB and Cassandra). It also illustrates different parameters that can be focused while choosing the cloud database or NoSQL database for role-based access control.

Table 1 Comparison of RBAC in Cloud Databases and NoSQL Databases

Parameters	Role Based Access Control Model			
	Cloud Databases		NoSQL Databases	
	Oracle 12c	SQL Azure	MongoDB	Cassandra
ACID Properties	Supported, but partially not supported in a case when Hive is used.	Supported	ACID Properties Partially Supported, Support BASE Properties	ACID Properties Partially Supported, Support BASE Properties
Sharding	Provides the highest availability of data in case of failure	High Performance	High Performance	Provides high availability of data with no single point of failure. [34]
Scalability	Horizontal Scaling based on demand	Both Horizontal and Vertical	Horizontal	Column Based
Securing Unstructured Data	Complex	Simple	Simple	Simple
Performance	High	High	Medium	High
Efficiency	High	Medium	Medium	High
Fault Tolerance	Provided with IMCU (In Memory Compression Units)	Automated but having issues related to accessibility	Provide through auto resync and auto-shard sync-up	Replication is supported

Heterogeneous Environment	Suitable	Suitable	Suitable	Most Suitable
Data Encryption Support	Excellent	Excellent	Not Supported	Supported for Intercluster Network Communication
Business Intelligence Support	Excellent but Paid	Excellent but Paid	Excellent	Excellent
Authentication for native connection	Available in both sharded and unsharded configuration	Available in both sharded and unsharded configuration	Available only in unsharded configurations	Partially Available
Authorization	Done with WebLogic Framework	Done with stored procedure, data masking	For native connection	Done at Column Family Granularity Level
Auditing	Available	Available	Not Available	Not Available
Injection Attacks	Possible but can be minimized	Possible but cab be minimized	Possible Via Java Script or String concatenation	Possible CQL
Replication Mechanism	Complex	Complex	Simple	Simple

Based on the comparison of RBAC of both Cloud Databases and NoSQL Databases, the details for each parameter are:

1. ACID Properties: Both Oracle 12c and SQL Azure support Atomicity, Consistency, Isolation, and Durability, hence we can say they are ACID Compliant. Oracle 12c provide an API called Transaction Guard using which an application ensures that the transaction executes no more than once. [38] Windows Azure SQL Database systems not only provide rich query semantics, fault-tolerant data storage but also support for ACID transactions along with complex data processing capabilities. [10] NoSQL introduces an approach called BASE (Basically Available Soft-state, Eventual Consistency). Eventual Consistency means that data returned from all read operations may differ from the latest completed write operation. Later when a system goes in a steady state, the last written value will be returned. [23] NoSQL Databases like MongoDB, Cassandra support AID properties, so they are AID Compliant as they sacrifice consistency for performance reasons. [33]

2.Sharding: Sharding is a scalability and availability feature for On-Line Transaction Processing (OLTP) applications which allows distribution and replication of data across a pool of databases that share no hardware or software. Oracle Sharding is envisioned for OLTP applications which must have a well-defined data model and data distribution strategy. Single-Shard transactions provide high performance and multi-shard transactions are supported but reduce the level of performance for routing and access to rows. [12] SQL Azure provide a shard map manager for scaling out databases. It is a special database which manages global information about all databases (shards) in shard set. And every shard in a set contains maps that will track the local shard data. SQL Azure manage multi-shard queries by considering shards as external tables where all queries are processed on their coordinator node. Whereas in Oracle 12c, multi-shard queries are pushed down to each shard. Moreover, Oracle 12c sharding is automatically configured. In contrast, SQL Azure sharding requires the user to configure sharding, in case configuration changes. [8] Sharding in NoSQL databases like MongoDB and Cassandra is mainly envisioned for scalability and availability. In Oracle 12c, multi-shard operations like concurrent reads/updates get a consistent result, whereas, in NoSQL database, this cannot be achieved. NoSQL databases are shaped for simple workloads and therefore they are cheaper compare to Oracle 12c and SQL Azure. [8]

3. Scalability: Horizontal scaling means to add or remove databases in order to adjust capacity or overall performance, called "scaling out". Vertical scaling means to increase or decrease the performance level of an individual database, called "scaling up." In Oracle 12c, multi-shard queries are push down to each shard which improves the performance and reduces the amount of data transmission on a network. We can say Oracle 12c is highly scalable. In SQL Azure, multi-shard queries are processed on their coordinator node which makes scaling the difficult and large amount of data are moved on a network which degrades the query performance. MongoDB and Cassandra are more scalable as multiple coordinators execute multi-shard queries. Oracle 12C is less scalable than NoSQL Databases because single coordinator executes single shard query, multiple coordinators are formed for the next release. [8]

4. Securing Unstructured Data: When many file systems are used, there should be provision for directory services for levels of access control. In Oracle 12c, it may not be possible to restrict access to the individual user i.e., enabling a user access to any content in the directory gives access to all content in the directory. [3] In SQL Azure, there are different types of mechanisms. Each of them has its own data access and control model. For Active Directory Structure, there is Storage Account. Each storage account has Storage Account Key (SAK) and Shared Access Signature (SAS) which is used to control the access to all data. [25] In contrast, it is possible to restrict access to the individual user in Cassandra [29] and in MongoDB, at Collection Level, you can restrict a user.

5. Performance: In Oracle 12c, in memory Column store improves the performance of analytics, online transaction processing (OLTP) applications. [12]In SQL Azure, Transparent Data Encryption (TDE) and Column Level Encryption (CLE) guarantee data protection against any kind of attack. This encryption affects the query optimization performance and access to data. [25] In MongoDB, only one writer can modify database at a time i.e., with either database locking or collection level locking, other readers and writers are locked out. This is the major limitation. Apart from this, when you add or update a field in a document, the entire document is re-written and for this reason, space is pre-allocated to each document. As a consequence, even if you pre-allocate space for each document, access to your document gets slower as your document grows. Cassandra uses advance concurrent structures to support row-level isolation without locking. In this way, there is no re-read and re-write on existing data which makes access to the documents faster even if your dataset grows. NoSQL databases are in memory databases means the whole database resides in RAM. If a failure occurs and a database is no longer available in RAM and as a result, this would not affect the RBAC. [27] In both Cloud databases and NoSQL databases, the backward and forward access to the data is costly. To make this access cost-effective, the concept of a nested object can be used instead of a simple object. [28]

6. Efficiency: An RBAC of Oracle 12c is more efficient than RBAC of NoSQL databases, but it is complex. In SQL Azure, there are two types of roles for Cloud Services 1. Web Role 2. Worker Role. A web role is used to provide dedicated IIS (Internet Information Services) to host web application and accessibility to it. Applications in Worker Role run long and asynchronous without user interaction. An RBAC in SQL Azure is also more efficient than RBAC of NoSQL databases. However, access control in Cassandra work efficiently in every condition plus it is simple. RBAC in MongoDB is simple but less efficient in terms of fault tolerance. [36]

7. Fault Tolerance: Oracle 12c has dual-format architecture. 1. In-memory row format and 2. In-memory column format. If a node goes down, data are still accessible via IM (In memory) Column store. Thus, it improves the performance means queries can access both the primary IMCU (In Memory Compression Units) and a backup copy of IMCU. [12] In SQL Azure, databases are automatically moved from heavily accessed machines to other machines and thus maintains the scalability plus RBAC will not be affected by this movement of databases. The issues like latency time while accessing data, unavailability of storage replicas, failure still need to be resolved. [10] Cassandra is highly available and fault tolerant database.Data are automatically replicated on multiple nodes. Multiple data centers support Replication. There is no down time for failed node(s). [37] MongoDB is Partition Tolerance Database. It is not available 100% when a data center is down. If there is a partition between two nodes i.e., both nodes are up but not in communication, then also cluster continues its work. That means access to database or resources is continued even after partition. Features like auto resync and auto-shard sync-up establish a communication between nodes. [36]

8. Heterogeneous Environment: Different users access different applications on any of devices like mobile, tablet, computer and notepad from different locations. Moreover, these devices may have different hardware and system software configuration from different vendors. Apart from these, there is a variation of data (which can be structured or unstructured) and applications, so it is difficult to design and define the system for different users. [4] In this scenario, Cassandra is the best suited. Microsoft Azure supports Availability on Demand (AoD) for a heterogeneous environment. For Enabling AoD Technology, a unified solution says Azure Site Recovery (ASR) is used which provides one-click migration/recovery of virtual and physical workload. In SQL Azure, application Data requires being replicated once, later which can be used for both migration and recovery. The RBAC support remains as it is in a case of migration and/or recovery.

9. Data Encryption Support: Oracle 12c has two critical preventive controls: 1. Transparent Data Encryption and 2. Data Redaction. Unauthorized users cannot have access to sensitive data as Data Redaction provides selective, on-the-fly redaction of sensitive data. [7] SQL Azure supports data encryption at various levels of encryption at data rest, volume level encryption, encryption in transit, platform encryption. Eventually, we need to decide that where we require encryption and/or decryption i.e., in the cloud, to the client side, on-premises, in an application and the like. This decision depends on the level of control you require to maintain plus performance, administration cost. [25] Cassandra provides Internode Encryption, data at rest encryption and client encryption. For Data Encryption especially at Internode, Role Based Access plays a vital role because authentication between the client and the server may be vulnerable to SQL Injection and Denial of Service Attack.[10]In MongoDB, data can be encrypted at the application level or via the external file system, but this will add complexity and cost. [32] By using MongoDB 3.2 (which has Encrypted Storage Engine), administrators reduce both the performance and management overhead of external encryption mechanisms. Compare to Oracle 12c and Cassandra, MongoDB is more mature and suitable for dynamic queries processing requirements and defining indexes. [9]

10. Business Intelligence Support: Oracle Business Intelligence 12c (BI 12c) provides rich Visual analytics portfolio which makes access to the data and resources easy. BI 12c supports extensive enterprise management reporting via direct access to Hyperion Financial Management Application data with single sign-on. [13] SQL Azure supports BI Components like Server Integration Services (SSIS), Power BI, Analysis Services (SSAS), and Reporting Services (SSRS), which benefit the environment (on-premises) and data sources on the cloud for analytics and

scalability. [25] You can manage MongoDB on your own infrastructure with the help of MongoDB Cloud Manager. It is a cloud-based tool that provides fine-grained monitoring, automated provisioning, and continuous backups. [14]

11. Authentication for Native Connection: Authentication means what identity (specifically username and password) is given to the user to access the resources and data in the cloud. Oracle 12c performs special database operations for authentication and for that it encrypts passwords during transmission to ensure the network authentication security. [30] Azure Active Directory (Azure AD) is used for authentication. It is a mechanism which connects Microsoft Azure SQL Database and SQL Data Warehouse. Azure AD Authentication allows the Database Administrator to manage database users' identity and Microsoft Services in one central location. In this way, it simplifies permission management. [39]
Cassandra provides an IAuthentication. The available solution is not ready to use. So, implement the custom provider. By default, authentication is turn off. You can allow authentication by setting up username and password via a Java Properties file which has sets of password properties. The password can be in plain text or as an unsalted MD5 Hash. [5] The MD5 hash is not secured, hence it is easy to trap password.Cassandra uses apache thrift framework for client communications. An attacker that is capable of monitoring the database traffic will be able to see all of the data as clients see it. The client interface supports a login() operation, but both username and password are sent across the network as clear text. Nodes on a cluster can communicate freely and no encryption or authentication is used. An interface called fat client is used to load bulk data into Cassandra from different nodes in the cluster. No protection or authentication is provided for fat clients at all.

In MongoDB, authentication is not supported in Sharded mode. In unsharded mode i.e., replica-set or standalone mode, the authentication is based on a pre-shared secret which is provided using key file parameter in configuration file. The user is added in admin database is considered as a DBA user with special privileges. The password is kept in the MD5 hash of string <username>: mongo: <password>, and can easily be read from admin data-files. So, an attacker who has access to these admin data files can easily recover all users' passwords defined in a database. Here, the MD5 algorithm is not counted very secure. However, restful API with a reverse proxy can provide fine-grain permissions for both authentication and authorization. [6]

12. Authorization: Authorization means what the user can do with the data and resources in the cloud with existing permission and roles. Authorization in Oracle 12c is provided with the help of WebLogic Security Framework where Authorization Providers interact with each other using this framework. [31] SQL Azure supports various features for authorization like impersonation, module-signing, Stored Procedure, Data Masking and Row-Level Security. [39][40] In Cassandra, Authorization is handled at Column Family (CF) Granularity level. [9] In MongoDB, there is no support of Authorization in Sharded Mode. In Unsharded Mode, both read-only and read-write permissions are assigned to users. Users having read-only permission can access everything in Database whereas Users having read-write permission have full access to the Database and is considered as Database Administrator (DBA). For stronger authentication, in Restful API with reverse proxy, a fine-grained permissions can be defined. [6]

13. Auditing: For supporting Auditing in Oracle 12c, Integrigy's Framework is used, which provides auditing of not only database users but also application users. [11] With Unified Auditing Policy, you can audit several activities like user accounts, object actions, application context values and activities from Oracle Label Security, Oracle Recovery Manager, Oracle Database Real Application Security, Oracle Data Mining, Oracle Data Pump. Moreover, in Oracle 12c, Role auditing is also available which allows you to audit all system privileges that are directly granted to the role. [35]

There are two audition methods in SQL Azure: 1. Blob Auditing and 2. Table Auditing. These methods allow to analyze, retain and report unusual activities, suspicious events on data and resources on the cloud. Inline auditing is not supported by Cassandra. Whenever auditing is required then custom implementation of IAuthority (for full auditing of all operations) and IAuthenticate (for a full audit trail of all login success/failure occurrence) can be written. [6] In a case of MongoDB, whenever new namespace is created, then one line in the log file for data file creation is written. Afterward, nothing is written in a log file on subsequent insertion, updates or queries. Hence, we can say Oracle 12c Auditing is the strongest.

14. Injection Attacks: In Oracle 12c, SQL Injection Attacks can be narrowed with the help of runtime conditions in access policies. [7] In SQL Azure, both user and application use separate accounts to authenticate and access to the data and resources in the cloud. The concept of a contained database user is used where in place of creating a server login for a user and granting permission, contained database user is created to segregate app account and user. Contained database users can directly authenticate to the database instead of make an extra hop to the master database. This will not allow malicious activity and/or injection attacks to access data and resources in the cloud. [26] Cassandra Query Language (CQL) is a parsed query language which is vulnerable to injection attacks. On the other hand, MongoDB strongly uses JavaScript. Moreover, JavaScript functions can be stored in the database which is accessible to the database users. There is a high possibility of injection attacks because JavaScript is interpreted Language. [6]

15. Replication Mechanism: For High Availability of data, MongoDB and Cassandra have Simple Replication Mechanism compare to Oracle 12C i.e., in MongoDB and Cassandra direct writes to each replica rather than replication from a master to its replicas. [8] SQL azure has features like Geo-Restore, Standard-Geo Replication, and

Active-Geo Replication which provide high availability of data in case of failure.

IV.CONCLUSION

In this paper, we addressed various parameters supported in Cloud Databases and NoSQL Databases in terms of role-based access. We tried to classify role based access in different Cloud Databases and NoSQL Databases available in the literature and showed how different parameters can be considered among those databases. At present, no such role-based access exists which provides complete security to the data and resources in the cloud. Still, sufficient work is required to ensure role based access for applications in Cloud Databases and NoSQL Databases based on various parameters discussed in the paper.

V.REFERENCES

[1] Pietro Colombo Elena Ferrari, "Enhancing MongoDB with Purpose-based Access Control", (Volume: PP, Issue: 99), November 2015.IEEE Transactions on Dependable and Secure Computing DOI: 10.1109/TDSC.2015.2497680 PrintISSN: 1545-5971

[2] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. Vol. 4, No. 3, August 2001, Pages 224–274.ACM Transactions on Information and System Security (TISSEC).

[3] Unstructured Data Management with Oracle Database 12c, November 2016,Oracle White Paper, pp. 3.

[4]Indu Arora, Dr. Anu Gupta, "Cloud Databases: A Paradigm Shift in Databases", Vol. 9, Issue 4, No 3, July 2012.IJCSI International Journal of Computer Science Issues, pp. 80 ISSN (Online): 1694-0814

[5]R. Rivest, "The md5 message-digest algorithm." RFC 1321, April 1992.MIT Laboratory for Computer Science and RSA Data Security, Inc, Tech. Rep. Available Online: https://tools.ietf.org/html/rfc1321

[6]LiorOkman, Nurit Gal-Oz, YaronGonen, Ehud Gudes, Jenny Abramov, "Security Issues in NoSQL Databases", 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11,pp. 541-547, DOI 10.1109/TrustCom.2011.70

[7] Encryption and Redaction in Oracle Database 12c with Oracle Advanced Security, March 2017, Oracle White Paper, pp. 3-13.

[8] Oracle Sharding FAQ, February 2017,Oracle White Paper, pp. 6 & 13.

[9] Ganesh Chandra Deka, Chap. 7 "Cloud Database Security Issues and Challenges",Year 2014, Book on "Handbook of Research on Securing Cloud-Based Databases with Biometric Application, pp.167-169.

[10] PratyushGuleria, "ACID Support and Fault-Tolerant Database Systems on Cloud:A Review", Volume 1, Issue 8, October 2011. International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), pp. 18-20.

[11] Integrigy – Oracle 12c Unified Auditing, April 2016.Oracle White Paper, pp. 4-24.

[12] Oracle Database In-Memory with Oracle Database 12c Release 2 Technical Overview, March 2017. Oracle White Paper, pp. 3 & 24.

[13] Powerful Visual Analytics for the Entire Organization – Oracle Business Intelligence 12c, Oracle Data Sheet, pp. 1-5.

[14] MongoDB: Bringing Online Big Data to Business Intelligence & Analytics,June 2016. A MongoDB White Paper, pp. 16-18.

[15] RajkumarBuyya, Chee Shin Yeo, and SrikumarVenugopal, "Market-Oriented Cloud Computing: Vision, Hype and Reality for Delivering IT Services as Computing Utilities," 2008. pp. 1-9 Online: www.cloudbus.org/papers/hpcc2008_keynote_cloudcomputing.pdf

[16] Tuan V., "Cloud Data Management," IFSIC, IRISA, Ker Data Project-Team. Jan, 2010.

[17] Chris G. et al., "The Eucalyptus open source cloud computing system,", Year 2009, IEEE International Symposium on Cluster Computing and the Grid, pp. 124-131. DOI 10.1109/CCGRID.2009.93

[18] G. Decandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store," Oct. 14-17, 2007, in Proceedings of the 21st ACM Symposium on Operating Systems Principles, Stevenson, WA. pp. 205-220

[19]F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," vol. 26, pp. 4:1–4:26, June 2008.ACM Transactions on Computer Systems (TOCS), [Online]. Available: http://doi.acm.org/10.1145/1365815.1365816

[20]Hamilton, James (July 12, 2008). "Facebook Releases Cassandra as Open Source" Retrieved April 26, 2017.

[21] Security and Compliance with Oracle Database 12c,April 2014. White Paper, pp. 8 & 15.

[22] Manjeet Singh "Study on Cloud Computing and Cloud Database", Year 2015, International Conference on Computing, Communication and Automation (ICCCA2015) ISBN:978-1-4799-8890-7/15/$31.00 ©2015 IEEE pp. 708-713

[23] Ayman E. Lotfy, Ahmed I. Saleh, Haitham A. El-Ghareeb, Hesham A. Ali, "A middle layer solution to support ACID properties for NoSQL Databases", Year 2016, Journal of King Saud University – Computer and Information Sciences (2016) 28, pp. 133–145. http://dx.doi.org/10.1016/j.jksuci.2015.05.003

[24] Protecting Data in Microsoft Azure, August 2014,White Paper, pp. 5-37

[25]Joseph D'Antoni, StaciaMisner, "Microsoft SQL Server,Using Power BI in a Hybrid Environment", August 2014. Microsoft White Paper, pp. 14-32.

[26] Joseph D'Antoni, StaciaVarga, "Security and Azure SQL Database", Technical White Paper, October 2015.

[27] Priti M. Tailor, Rustom D. Morena, "A Survey of Database Buffer Cache Management Approaches", Volume 8, No. 3, March – April 2017.International Journal of Advanced Research in Computer Science, ISSN No. 0976-5697

[28] Bharti Joshi, Rustom D. Morena, "A novel approach for queries on nested objects", Vol. 1, No. 1, 2015. Int. J. Data Science, pp. 84-95

[29] Retrieved from https://www.datastax.com/dev/blog/role-based-access-control-in-cassandra on April 23, 2017.

[30] Retrieved from https://docs.oracle.com/database/121/DBSEG/authentication.htm#DBSEG0032 on April 23, 2017.

[31] Retrieved from https://docs.oracle.com/cd/E24329_01/web.1211/e24486/atz.htm#DEVSP296 on April 23, 2017.

[32]Retrieved from https://www.mongodb.com/blog/post/securing-mongodb-part-3-database-auditing-and-encryption on April 23, 2017.

[33] Retrieved from http://www.javaworld.com/article/2078841/enterprise-

java/datastax-ceo--let-s-clear-the-air-about-nosql-and-acid.html on April 23, 2017.

[34] Retrieved from http://www.datastax.com/nosql-databases/benchmarks-cassandra-vs-mongodb-vs-Hbase on April 23, 2017.

[35] Retrieved from http://docs.oracle.com/database/121/DBSEG/audit_config.htm#DBSEG355 on April 26, 2017.

[36] Retrieved from https://phisymmetry.wordpress.com/2015/03/05/mongodb-fault-tolerance-strategy/ on April 26, 2017.

[37] Retrieved from http://cassandra.apache.org/ on April 26, 2017.

[38] Retrieved from https://docs.oracle.com/database/121/CNCPT/transact.htm#CNCPT89319 on April 27, 2017.

[39] Rick Byham, "Use Azure Active Directory Authentication for authentication with SQL Database or SQL Data Warehouse", March 2017. Retrieved from https://docs.microsoft.com/en-us/azure/sql-database/sql-database-aad-authentication on May 2, 2017.

[40] Rick Byham, "Azure SQL Database Access Control", February 2017. Retrieved from https://docs.microsoft.com/en-us/azure/sql-database/sql-database-control-access on May 2, 2017.