Secure and Efficient Multitenant Database in SaaS Delivery Model for an Ad hoc Cloud

Basant Kumar Gupta[1], **Kalicharan Sahu**[2]

[1]PG Scholar, Computer Science and Engineering Department, Galgotias University, Greater Noida U.P, India

[2]Assistant Professor, Computer Science and Engineering Department, Galgotias University, Greater Noida U.P, India

Abstract: ***To implement secured SaaS delivery model, save costs, reduced service customizations and improved resource utilization, recently cloud service providers rapidly use the Multitenant database technology. Organizations can collaborate to create a Data Center which does not harm their availability and ability of gaining profit. These organizations are now days competed by spreading locations of their certain edges of others. Here Ad-Hoc cloud helps vendors in remote areas. Here the necessity increases for multitenant database in ad-hoc cloud to help multiple organizations to integrate and compete with each others. Here shared database's shared schema approach has been proposed which offers huge numbers of tenants as per database server. Authentication and Authorization are the noble specifications when we are going to deal with multitenancy. Most secured authentication protocol Kerberos is used on the top of multitenant database for taking participation in educational institutions.***

Keywords: Multitenant Database, SaaS, Ad hoc Cloud, Kerberos, Authentication, Authorization.

1. Introduction

Cloud computing is a computing technique where facilities and data reside in common character in scalable data centers, that are accessible through authentication. Cloud computing facilities can form a obstinate infrastructural and service based framework to avail any type of service oriented computing atmosphere. Ad-hoc clouds [1] enables existing infrastructure as cloud accurateness, the existing resources in the environment are used non-intrusively. An Ad-hoc cloud is the add-on efficient optional appendage happening occurring to troubles confronted by organizations to pretend into unpleasant areas. Educational-cloud, where a cloud computing model is reined to run Information system of rules of an Educational activity institution will be chosen efficient in condition of scalability, accessibility, availability and manageability. An ad-hoc cloud computing paradigm would enable cloud users rein facilities offered by Fixed Education-cloud and facilities created and composed within ad-hoc cloud. The ad hoc cloud as shown in fig-1 derives data and cloud service from fixed cloud, later they are allied using an ad hoc link (V-SAT).

In Multitenancy single instance of an application program gratifies the petitions of far and wide-off along clients. Each individual educational organization is conceives as a tenant and all such administrations cooperate to make and participate in data-descent building process. Multitenancy in Education-cloud, where a cloud computing model is reined to handle Information system of an Educational institution would very huge benefit in terms of comply space, scalability and availability.We propose to utilize our multitenant database for such a scenario, where tenants (Educational Institutions) collaborate to construct the decentralized database and utilize it by authorization. In this scheme the tenants keeps joining and leaving. Providing efficient multi-tenant database with transactional level assure for the distributed database to be utilized as data calm in a cloud is our concern.

Various techniques for multi-tenancy have been suggested depending upon the degree of ferociousness. Four extensive techniques are:

- **Separate database**: It is used to stock the data of an individual tenant (Educational Institution).
- **Shared database, separate schema:** This technique requires multiple tenants to be conciliated into an alone database.
- **Shared database, shared schema:** It implies same database and schema to be traded by all tenants.

The major thinking of this feint is to take occurrence Multitenant database. The specific aims of this paper are:

- To meet the expense of architecture that backing multitenancy in shared database shared schema fashion.
- To run heterogeneity in condition of database attributes and technology are managed.
- To integrate authentication and authorization model for multitenant application program.

To meet above aims we have produced architecture that backing multitenancy. For authentication and attributed approval Kerberos protocol has been used.

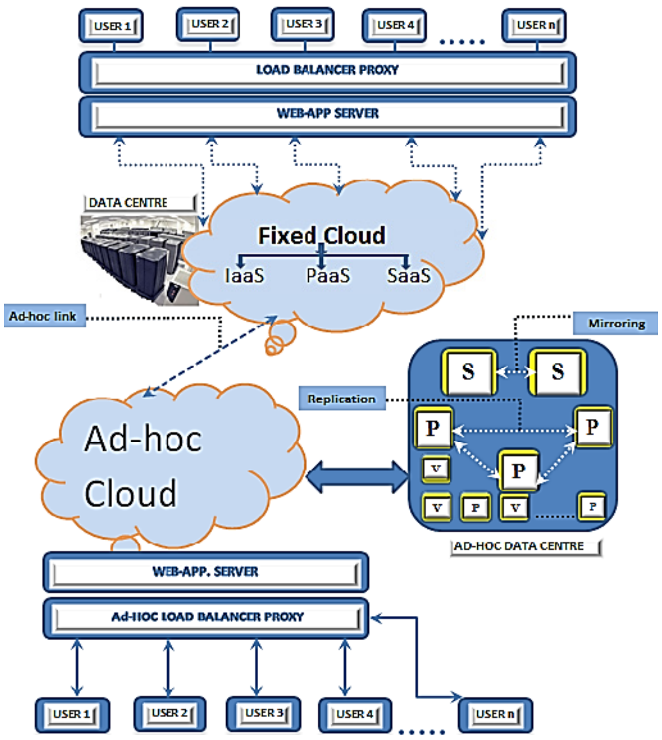


Figure 1: Scenario of cloud Architecture

The content of this paper is managed as follows**:** section 1 tells the introduction and section 2 explains the associated works finished before. The details about the suggested architecture are explained in Section 3. Section 4 presents the implementation details followed by results elaborated in Section 5. Section 6 provides the efficiently ahead scope of the work done and concluding remarks.

2. Related works

Many mechanisms for scalability and data organization favour have been introduced like big table [3] and dynamo [4], but nonattendance in providing transactional level guaranty. We use the concept of Elastras [5] which is a faint burden data storage having ability of availing transactional level assurance. Our data storage would have Higher Level Transaction Manager (HLTM) and Organizational Level Transaction Manager (OLTM) as shown in fig-2. All transactions within an organization will be managed by OLTM and along moreover organizations will be handled by HLTM. Elasticity is important as it will not be upper bound industrial accident for scalabilityat data layer level. Elasticity is provided by decoupling the data base commissioner (DM) subsequent to the transaction manager. Application servers' admittance the data entire quantity through load balancer for data partner. For a transaction demand the OLTM checks his finishing guaranteeing ACID properties for a transaction, if it cannot forward it forwards the request to sudden HTLM. Finally a single or adding together taking place of ODM (Organizations Database Manager) owing the database (data storage calculation in the feel) commits the transaction.

The Metadata Manager (MM) implementation caters decoupling of database and transaction bureaucrat and it in addition to caters mapping of distributed database segmentations into OLTM. Synchronal reverberation of MM is required for anomaly tolerance. Storage grows takes care of replication of data and idiosyncrasy tolerance. Slower nodes can use metadata caching for augmented release commitment. Since HTLM are stateless so to colleague going almost meting out during scalability spawning a count HTLM is easy. Further data base migration among data storage or in cloud can be over and ended together along surrounded by as discussed in Albatross [6].

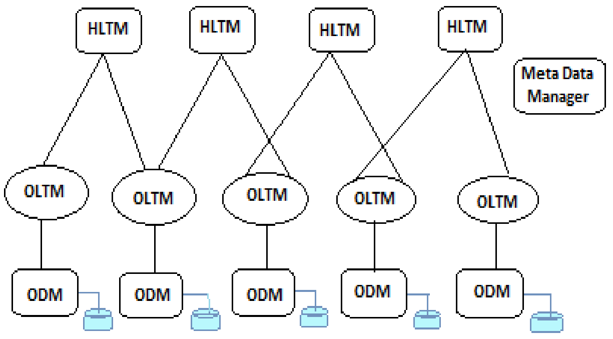


Figure 2: Decoupling of Database and Transaction Managers

The use of multitenant database systems in present scenario of SaaS delivery architecture of Cloud Computing increased multifold [7]. A data center is hosted by an instruction provider and the tenants subscribe to the facilities provided by the encourage provider [8]in a multitenant database.Some of approaches for providing multitenancy are as follows:

2.1 Universal Table Layout

A universal table contains pre defined number of fields [9] that incorporate of a table column, all the data columns and Tenant_id columns that is utilized to single key out the data of a tenant and the table column concern to the id of the table for that tenant. This right to use has been descent from Universal Relation [10] that a table backs every column from all the tables. This mechanism is relatively easy to employ and queries are implemented directly to the table.

2.2 Multitenant Shared Table

In this mechanism, common contents from tenant information are separated. This approach proposes the concept of tenants at database intensification so that database engine can choose an appropriate place for warehousing of data for that tenant. A technique into that deals in scalability problem is talked over in [8]. Two primary issues are persistent in [11]. One is resolution of the sparsity of the universal mechanism and second is to avail an indexing schema for multitenant database. Three different techniques shared process; shared table and shared machine are talked over by Jacobs in [12]. A mechanism for multitenant architecture backing the SaaS framework is talked over in [13]. The writers have offered a cloudio software application platform which is related later the elasticity of data model and handling the huge data sets in the database. Dissimilar disputes in Multitenant applications are talked over in [18] such as zero downtime, benefit, scalability and security.

2.3 Pivot Tables

In this mechanism, a pivot table is making for a unique column [11]. This table is shared by various tenants' tables. Each pivot table comprise of a "col" column, a tenant column, a "row" column and a table column. Tenant column refers to the particular tenant. Table concerns to the particular table for that tenant.

2.4 Addressing the Application level security of applications

In [15] scenarios for authentication and authorization have been discussed. A Number of conditions determine grant of resources to a particular user [16].Kerberos [15] is a widely used and fool proof authentication protocol. It provides high speed of communication, a tall degree of mutual authentication and transferrable degree of trust [17].

2.5 Extension Tables

The concept of extension tables came into existence after the development of breaking up storage model talked over in [18]. It is breaking up a table of n-columns into n 2-column tables that are mixed-up together. One matter of this admittance is how to partition the table for that tableso that after joining these tables no added information is generated.

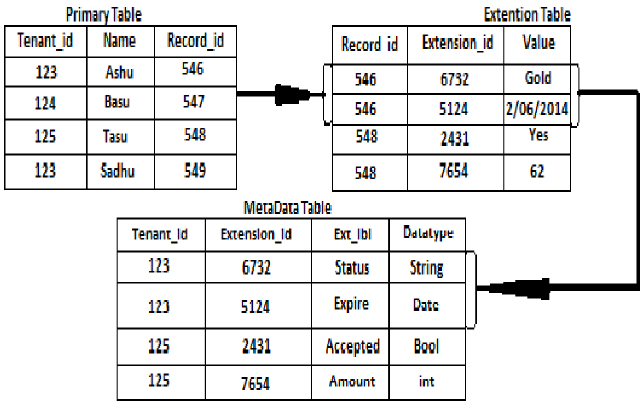
2.6 Chunk Folding

Chunk folding is an approach talked over in [18]. It vertically decomposed the logical tables into chunks and those are creased together into various physical tenants and are allied as needed. One table is used to add together the base account information and other table is utilized to carry the extensions. This admission deeds by carrying the densely utilized sections of the schema into base tables and the remaining part is mapped into the extensions.

3. Proposed Approach

Proposed technique is intended regarding peak of the shared database shared schema technique. This proposed technique creates utilization of extension table.
Why Extension Table?

In the universal table framework it is a hugechallenge to evaluate the amount of impost fields (table containing columns). Making available less number of columns may be bound the tenants that agonized feeling to use a large number of impost fields. A huge number of such fields may results in huge amount of NULL values in the database table. Second difficulty is of dissenting data types of these columns [9].



Primary Table

Tenant_id	Name	Record_id
123	Ashu	546
124	Basu	547
125	Tasu	548
123	Sadhu	549

Extention Table

Record id	Extension_id	Value
546	6732	Gold
546	5124	2/06/2014
548	2431	Yes
548	7654	62

MetaData Table

Tenant_Id	Extension_Id	Ext_lbl	Datatype
123	6732	Status	String
123	5124	Expire	Date
125	2431	Accepted	Bool
125	7654	Amount	int

Figure 3: Use of extension table

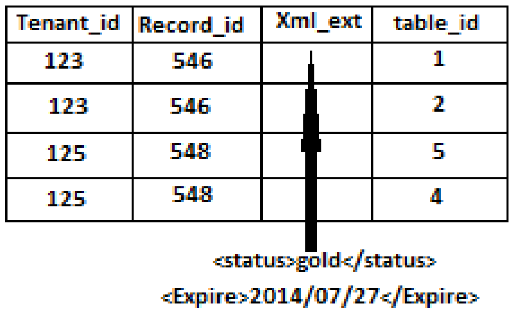
Fig-3 shows three tables used in the basic mechanism, which makes use of extension table. The principle table preserves the record_id and tenant_id and some different fields. Record_id field uniquely identifies the dealing made by a particular tenant. By taking out the value of record_id field, one can take out the values from the extension table. For an alone record_id, there are lots of rows in the involve ahead table. Lots of rows for an individual record_id are lots of columns in the severe table of same tenant. The Meta Data table shows as regards the data types of these fields. Any time when a tenant enters data into their table, Meta Data table is approaches to have the same guidance the miserable values calm just on the subject of-right to use-easy to use to the data types of the Meta data table. An ext_id in extension table is linked with an extension_id field of Meta Data table. This extension_id is alone for all columns and is utilized to recognize the data type and uncovered tag of the same field in the logical table for same tenant. An extension table carries the actual data for each and every tenant. In case of universal table structure, columns, that are not utilized by a particular tenant, having the NULL value, this effect in use wastage of space. Extension table thought defeat this matter. Fig-4 and Fig-5 demonstrate an extension table and a Meta Data table. In the basic technique of extension table discussed above, behind corns can be noticed:

1. Extension table carry the numbers of information for Meta Data i.e. for an alone row of table of a tenant which have four columns. The record_id and extension_id are iterated four times this information proposes a type of redundancy.
2. Whenever a query for insertion, subtraction or update is applied three tables are admittance that makes the increments in the query dispensation time.

In our proposed technique following conceptions are proposed and applied-

1. Conception of XML motility into a database is utilized which helps to shorten the size of showing off table as astern ease as eliminates the way of a primary table.
2. A mechanism that achieves multiple tables making for a tenant is introduced and successfully implemented.
3. Kerberos authentication protocol is used for authentication and authentication.

A table which sustains the information about each of the tables of every tenant is created. This table The table_id field of the extension table is represented by that table to the name of the table that a tenant is cited to.



Tenant_id	Record_id	Xml_ext	table_id
123	546		1
123	546		2
125	548		5
125	548		4

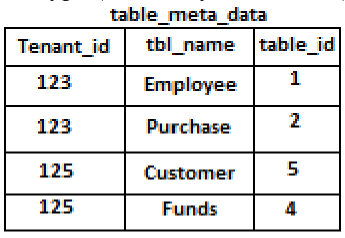
<status>gold</status>
<Expire>2014/07/27</Expire>

Figure 4: Modified Extension Table

Fig-4 shows the proposed position where extension table having a table_id, an XML attribute, a record_id and a tenant_id. Record_id and tenant_id distinctively describe an individual record. A record_id is used to coordinate each transaction with a distinctive record number. XML object carries the data for whole row of a Tenant's logical table. Labels in a unique XML object mention to the name of an individual field in the associating table. Table_id field symolizes the id of the table in which an individualrecord is enclosed for the defined tenant. The tenant defines the name of the table and our introduced system develops single id for that table for that tenant.

3.1 Creating the Tables

A tenant is practicable to utilize any number of impost areas presuming that submit to serve to provider has created enough number of field in the primary database schema. A tenant is reprieve to make whatever number of tables and use whatever data type (backed by that DBMS) for its fields.



table_meta_data

Tenant_id	tbl_name	table_id
123	Employee	1
123	Purchase	2
125	Customer	5
125	Funds	4

Figure 5: Table table_meta_data

Whenever a tenant defines a new table name, this name is saved in the table_meta_data table. Fig-5 shows the shape of table_meta_data table.

3.2 Inserting Elements into the Table

A tenant defines the name of the table and appends the values. Our introduced architecture follows a succession of guide lines in the primary extension table as describes below:

1. The table id of that table for that tenant is taken out from the table_meta_data table.
2. Meta_data table is admittance to recognize the data types of the areas.
3. A record id, describing this individual dealing, is developed and is entered into the record id column.
4. An XML document with pushed values are developed whose labels are the column names in the table.
5. This XML document is pushed into ext_xml column of the extension table.
6. Table_id taken out from the table_meta_data is entered into the table_id column of extension table.

3.3 Updating the Information in the Database

The name of the field and the table name are specified, and in the middle of guide lines are applied:

1. It admittance the table_meta_data table from where it retrieves the id of the table for that tenant
2. In extension table, it obtains the rows associating to that table id and tenant.
3. From the XML documents, that are corresponding to the table_id, it makes an Xquery [19] that generates pitch cites.
4. It updates the value in the XML document and saves it sentient ahead in the gathering table in addition to the similar folder id.

3.4 Deleting the Guidance

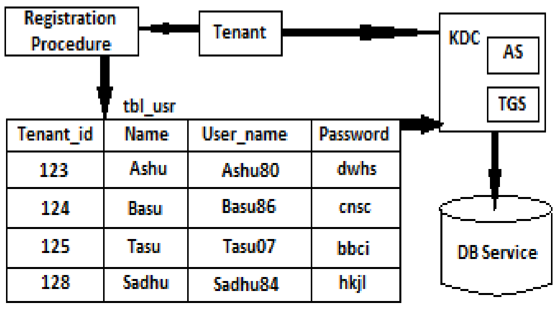
The table name and a value for a defined field is provided. To recognize the table_id of the table in update process, the table_meta_data table is admittance as well as latter on to know the rows associating to that table_id, corresponding table is accessed. The whole argument that having the XML document in which the defined value for the given field is discovered, is deleted.

3.5 Authentication and Authorization

A new tenant registers itself with the cloud service provider. It provides the IP address, user name, password and other details. At the period of registering a tenant defines the time of the subscription. It also defines the kind of users and authorities given to a particular useri.e. manager level user is allowed to all permissions whether employee level user can be restricted to only some permissions like are restricted to perform modification and deletion operations in the databases. Kerberos authentication protocol is needed for authentication and authorization.

4. Implementation

To implement the introduced mechanism, .NET platform and SQL server 2005 have been used. To generate the keys and encrypt the messages, cryptography class of .NET platform has been used. As in fig-6 the table tbl_usr contains the information about the tenants. When a subsidiary tenant registers with the cloud service provider, the registration procedure is called to check if this is a new user by traversing the list of its existing users and its IP address.



tbl_usr

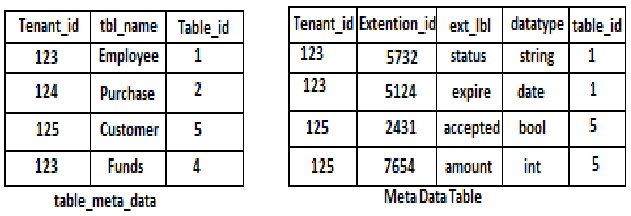
Tenant_id	Name	User_name	Password
123	Ashu	Ashu80	dwhs
124	Basu	Basu86	cnsc
125	Tasu	Tasu07	bbci
128	Sadhu	Sadhu84	hkjl

Figure 6: Proposed Architecture for authentication

It once assigns a Tenant id after registration process. All this way of information is kept in tbl_usrtable. This tenant id is used for identifying a particular tenant and his tables.

4.1 Creating the Tables

Each table name is given a unique table id. Meta data informationis also maintained in Meta_data_table table. This table contains the information like data type and external label for a column that a tenant is using. Each column is coordinated with aext_id. Fig-7 shows these two tables.



Tenant_id	tbl_name	Table_id
123	Employee	1
124	Purchase	2
125	Customer	5
123	Funds	4

table_meta_data

Tenant_id	Extention_id	ext_lbl	datatype	table_id
123	5732	status	string	1
123	5124	expire	date	1
125	2431	accepted	bool	5
125	7654	amount	int	5

Meta Data Table

Figure 7: Tables involved in Creation of a new table.

4.2 Inserting the Values in the Table

The actual values for all the tenants are stored in the extension table. The value is entered with the help of two tables i.e. Meta_data table and table_meta_data table as shown in fig. 8.

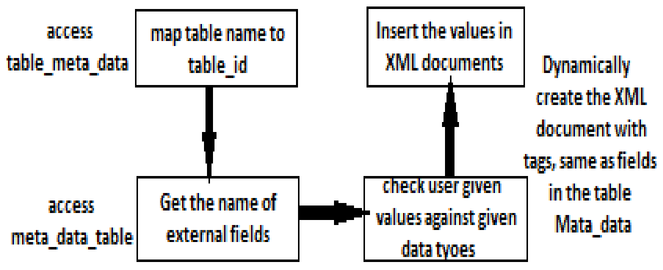

access table_meta_data
map table name to table_id
Insert the values in XML documents
Dynamically create the XML document with tags, same as fields in the table Mata_data
access meta_data_table
Get the name of external fields
check user given values against given data tyoes

Figure 8: Process showing the insertion in extension table

4.3 Creating the View for a Tenant

Following procedure in fig-9 shows the steps and implementation of this achieve:

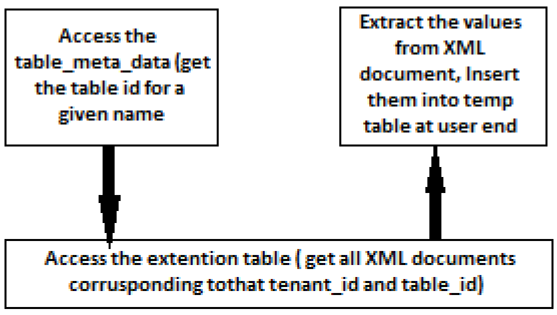

Access the table_meta_data (get the table id for a given name
Extract the values from XML document, Insert them into temp table at user end
Access the extention table (get all XML documents corrusponding tothat tenant_id and table_id)

Figure 9: Extracting information for a particular Tenant

4.4 Implementing Kerberos

Following procedure has been implemented for authentication and authorization**:**

1) Getting the TGS
2) Getting the SGT
3) Getting entrance to the server

5. Results

The introduced technique has been triumphantly implemented and queries like Subtraction, selection and insertion have been experimented. For more additional attributes in a table saves much more space as compared to the extension table technique in fig-10 and performance became slightly better than the previous. Due to the use of XML in the attribute, it gets this type of profit.

5.1 Comparison with the Pivot Table Mechanism

It is observed that number of tables in this mechanism depends upon the number of fields, even though in the introduced mechanism the number of tables is fixed.

5.2 Comparison with Basic Extension Table

The number of rows in the native extension table be dependent upon the number of fields in a tenant's table. But in modified extension table contains only a single ablaze for a row.

5.3 Comparison Subsequent To Universal Table Approach

The problem of Null value is solved. A tenant can use any fieldfor any type of data.

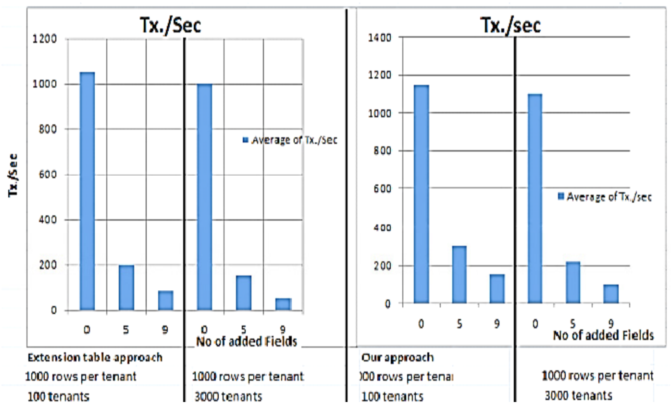


Figure 10: Comparison b/w Extension table and our approach [2]

6. Conclusions

An attempt has been made to implement the Multitenant database in this paper that proposes operational leads over the surviving ones. It suitable very well in schemes wherever SaaS cloud facilities are to be delivered among collective clients (institutions) with authentication. The proposed multitenant database makes arrangement of huge number of tenants due to a single database instance is utilized to add replacement of the data of perplexing tenants. Another advantage of our combat is that tenants are allowed to create disturb happening tables which associated flexibility from tenants mitigation of view.

References

[1] Chandra, Abhishek and WeissmanJon.Nebulas: Using distributed voluntary resources to build clouds. In Proceedings of the 2009 conference on Hot topics in cloudcomputing, ACM id 1855535 ,USENIX Association, 2009.

[2] Pippal, Sanjeev, et al. "Secure and efficient multitenant database for an ad hoc cloud." *Securing Services on the Cloud (IWSSC), 2011 1st International Workshop on*.IEEE, 2011.

[3] 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. In *OSDI*, pages 205–218, 2006.

[4] 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. In *SOSP*, pages 205–220, 2007.

[5] S. Das, S. Agarwal, D. Agrawal, and A. El Abbadi.ElasTraS: An Elastic, Scalable, and Self Managing Transactional Database for the Cloud. Technical Report 2010-04, CS, UCSB, 2010.

[6] Sudipto Das, Shoji Nishimura, DivyakantAgrawal, Amr El Abbadi, "Albatross: Lightweight Elasticity in Shared Storage Databases for the Cloud using Live Data Migration", In the 37th International Conference on Very Large Data Bases (VLDB) 2011.

[7] H. Hacigumus, B. Iyer, S. Mehrotra, "Providing database as a service", In 18th International Conference on Data Engineering, pp. 29-38, 2002.

[8] Mei Hui, Dawei Jiang, Guoliang Li, Yuan Zhou, "Supporting Database Applications as a service", In IEEE 25th International Conference on Data Engineering, pp. 832-843, 2009.

[9] Stefan Aulbach, TorstenGrust, Dean Jacobs, Alfons Kemper, Jan Rittinger, "Multi-Tenant Databases for Software as a Service:Schema-Mapping Techniques", In the proc. of International Conference on Management of Data - SIGMOD, pp. 1195-1206, 2008.

[10] D. Maier and J. D. Ullman, "Maximal objects and the semantics of universal relation databases", In ACM Trans. Database Syst., pp.114, 1983.

[11] M. Grund, M. Schapranow J. Kruege, J. Schaffner, A. Bog. In IEEE Symposium on Advanced Management of Information for Globalized Enterprises,pp 1-5, 2008.

[12] Dean Jacobs,StefanAulbach, "Ruminations on Multi-Tenant Databases", Datenbanksysteme in Bro, TechnikundWissenschaft(German Database Conference) – BTW , pp. 514-521, 2007.

[13] E.J.Domingo, J.T.Nino, A.L.Lemos, M.L.Lemos, R.C. Palacios, J.M.G. Berbis, "A Cloud Computing-oriented Multi-Tenant Architecture for Business Information Systems", In IEEE 3rd International Conference on Cloud Computing, pp. 532-533, 2010.

[14] Cor-Paul Bezemer, Andy Zaidman, "Challenges of Reengineering into Multi-Tenant SaaS Applications", Report TUD-SERG-2010-012, Delft University of Technology Software Engineering Research Group Technical Report Series.

[15] http:www:ibm:com/developerworks/library/arsaassec/index.html. Last accessed on June 4, 2011.

[16] Peter C. Chapin, Christian Skalka and X. SeanWang, "Authorization in trust Management: Features and Foundations", In ACM Computing Surveys, Vol. 40, Issue 3, pp. 1-48,August 2008.

[17] Ching Lin and Vijay Varadharajan, "Trust based risk management for distributed system security- a new approach, In the Proc. of the IEEE First International Conference on Availability, Reliability and Security, 2006.

[18] G. P. Copeland , S. N. Khoshafian. "A decomposition storage model", In the Proc. of 2005 ACM SIGMOD International conference on Management of data, pages 268279. ACM, 2005.

[19] www. Consortium. XQuery: The W3C query language for XML W3C working draft. Available at http /www:w3:org/TR/xquery/;2001

Author Profile

Basant Kumar Gupta is M. Tech. CSE Student at Galgotias University. He has received Bachelors' degree in Information Technology in 2012 from K.N.I.P.S.S. Sultanpur, U.P.

Mr. Kalicharan Sahu is working as Assistant Professor, School of Computing Science, Galgotias University. He has received degree of M. Tech. in CSE (JNU, New Delhi, India) & BE in IT (B.I.T.S. an autonomous institute under R.G.P.V. Bhopal, India)