

Modeling and Designing Integrated Framework for Data Management of Transactional Applications in Cloud

Indu Arora, Dr. Anu Gupta

Abstract—Cloud Computing has been in use for the last few years due to advantages like scalability, high availability and pay-per usage characteristics of Cloud. Though it is still in its infancy, companies are using it for managing and improving tasks like business processes, customer relations etc. Cloud is considered as constructive technology for read-intensive analytical applications than write-intensive transactional applications. Currently traditional licensed applications are used for managing transactional applications. Deploying transactional applications in the Cloud is not considered safe due to stringent consistency requirements of data in the Cloud. Consistency can be provided in Cloud by using middleware between application software and database. It becomes a major challenge to integrate these middleware tools for a transactional application developer. A need was felt to model and design an integrated framework for data management of transactional applications in Cloud to facilitate the development of transactional applications for educational institutes. This paper focuses on the need of such framework comprising Configuration Management and Data Access Management components. It further presents the model and design of Integrated Framework for Data Management of Transactional Applications in Cloud. It also discusses the implementation tools and design verification methodology used in the framework. After its implementation, it will become easier for Cloud developers to use this framework for developing transactional applications.

Index Terms — Cloud Computing, Data Management, Design, In-memory Data Grid, Integrated Framework, Model, Transaction Applications in Cloud

1 INTRODUCTION

Cloud Computing is a buzzword today due to continual growth of Internet in terms of speed and its usage. It is a cost-effective solution for cash-starved organizations [1]. Cloud is mainly used for collaborative services and analytical applications as transactional data management is difficult in Cloud [2]. Replication of data is must in Cloud to ensure high availability of data. Due to replication of same data on different servers, it is difficult to provide ACID guarantees on operational data of RDBMS which are must to keep write-intensive transactional data consistent and accurate. Deployment of transactional applications in Cloud needs special concern in the diverse and distributed environment of Cloud. To ensure ACID compliance data for transactional applications in Cloud, middleware tools like CloudTran over In-memory Data Grid, Oracle Coherence can be used [3]. Integration of such tools with various development tools is a difficult task. So, a need was felt to develop an integrated framework for transactional applications in Cloud which can be used easily by application developers to develop transactional applications so that they can harness the main advantages of Cloud.

This paper has been structured into seven sections. Second section briefs the need and benefits of integrated framework

for transactional applications in Cloud. Third section introduces model of integrated framework. Fourth section presents detailed design of integrated framework and explains the methods designed for meeting the desired requirements. Fifth section reviews the tools used for implementation. Sixth section discusses design verification methodology followed by conclusion.

2. NEED AND BENEFITS OF INTEGRATED FRAMEWORK

The purpose of data management in Cloud is to ensure high availability of data to its users with accuracy and reliability. Analytical applications and transactional applications are generally used to access data. Analytical applications requires data mostly in the form of semi structured or un-structured which is mined mostly for business intelligence purposes. Rick Cattell [4] clarified that NoSQL Cloud databases are used to handle such applications involving read intensive data. These databases generally follow BASE (Basically Available, Soft, Eventual consistency) rules. On the other hand transactional applications are used in almost every sector to manage daily activities of organizations to improve their functionality. Transactional applications have evolved from stand alone applications to browser based applications, but they require data with ACID (Atomicity, Consistency, Isolation and Durability) guarantees to maintain transactional integrity and consistency. Relational databases are managing transactional data for the last four decades successfully. These databases are able to handle vertical scaling, but find it difficult to manage horizontal scaling [5]. Marica Kaufman [6] compares scalability requirement of transactional data with that of analytical data and observed

- Indu Arora is currently pursuing Ph.D. in Computer Science and Applications in Panjab University, Chandigarh, India, indarora@yahoo.co.in
- Dr. Anu Gupta is Associate Professor in Department of Computer Science and Applications, Panjab University, Chandigarh, India, anugupta@pu.ac.in

that its integrity and consistency requirements should be met immediately. Data should persist to disk instantaneously while maintaining the order in which the transactions are executed. So, transactional applications need special consideration during design as they deal with sensitive operational data. Moreover due to the limited scalability of database layer in n-tier architecture of relational databases, existing transactional applications cannot be deployed in Cloud as such. Middleware tools ensuring ACID guarantees and In-memory Data Grid (IMDG) can be used to deploy transactional applications in Cloud to resolve such issues [7], [8], [9]. But usage of IMDGs needs different data modeling skills because data resides in RAM of different servers than storing it on a single data store. Design principles and deployment architectures of in-memory computing systems are more complex than those of conventional systems. In view of the above, a need was felt to design and develop an Integrated Framework for Data Management of Transactional Applications in Cloud.

The desired framework has been modeled and designed and it is named as InFraMegh. The term InFraMegh has been derived from the initial letters of integrated framework and the Megh. Megh is a Hindi word used for Cloud. The framework uses CloudTran over Oracle Coherence to provide ACID guarantees and MySQL as backend database. The framework has GUI (Graphical User Interface) for configuring CloudTran, Oracle Coherence and data connectivity for backend database. It also has APIs such as storing data into cache from main database and persisting it back on the disk. These APIs are designed to facilitate the development process of transactional applications in Cloud environment to interact with data lying in cache and database. The developer of Cloud based transactional applications will use only APIs of the integrated framework while developing their transactional applications. Developers can focus on business logic without bothering about the complexities of operations such as insert/delete.

3. MODEL OF INFRA MEGH

The proposed model of InFraMegh comprises two main components: Configuration Management and Data Access Management. The purpose of Configuration Management is to provide GUI to Cloud application developer to configure the ecosystem required for designing transactional applications. The ecosystem includes setting of database MySQL, middleware CloudTran and IMDG Oracle Coherence. The purpose of Data Access Management is to provide methods for transactional applications to manipulate data stored in IMDG. The Cloud application developer will use these methods like storing data in cache, inserting and deleting data from cache/ database in transactional applications meant for Cloud environment. Fig. 1 shows the model of InFraMegh.

3.1 Configuration Management

Configuration Management is an important part of InFraMegh. This component is required before the start of the development of transactional application meant for Cloud. It

is also required for any configurations change like inserting or deleting a table from the existing database made during the development of transactional applications. It forms the basis for Data access management. It does the following tasks for configuring ecosystem required by CloudTran and Oracle Coherence.

- Updates CloudTran configuration files
- Manages users and their privileges.
- Creates and updates schema class files automatically based on tables in configured database.
- Configures the path for accessing database from backend database, which helps in loading data from database to cache and writing updated data back to database from cache.

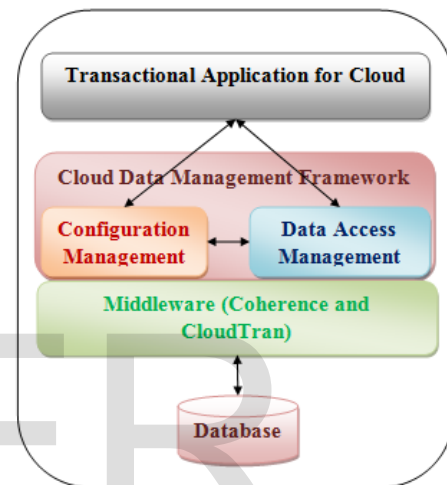


Fig. 1. Model of InFraMegh.

3.2 Data Access Management

Data Access Management provides methods to perform operations on data stored in IMDG for transactional applications deployed in Cloud environment. Based on the requirements of transactional applications, it provides methods for the following tasks.

- Loading data from database to cache for reading and writing.
- Data partitioning and replication is done based on default settings of middleware of CloudTran and Oracle Coherence.
- Supporting Data Manipulation operations like insert, delete, update and select.
- Generating reports based on data available in cache.
- Data backup and restoration
- Evicting data from cache when user logs out of the system or data is no more required.

4. DESIGN OF INFRA MEGH

Keeping in view the model of InFraMegh, an Integrated Cloud Data Management Framework for Transactional Applications in the Cloud has been designed. Designing includes Database Design and Process Design.

4.1 Database Design

Tools used in the implementation of framework require lots of environment settings. All the required properties of CloudTran, a middleware and Oracle Coherence, an IMDG are stored in XML files. The entities, cloudtran-pof-config, coherence-cache-config, application-pof-config, persistence and properties are used for setting configuration items of CloudTran and Oracle Coherence.

CloudTran-pof-config file is an xml file which contains the settings for classes based on user created tables. It also contains some class references required by CloudTran. Coherence-cache-config.xml file stores the information required by CloudTran for cache implementation. Persistence.xml file contains the settings for persistence units like name of server & client side persistence units, database server address (IP address), user name, password etc. The other main settings in persistence units are classes related to tables present in database. Application-pof-config file contains the settings for classes based on tables created by users.

Another main activity of this design is to store profile and roles of Cloud users who will use transactional applications deployed in Cloud. For this, there are two user management related entities: user and rights. The user entity contains information about the user and rights entity contains the privileges assigned to the user. The Fig. 2 shows Entity Relationship Diagram (ERD) for InFraMegh.

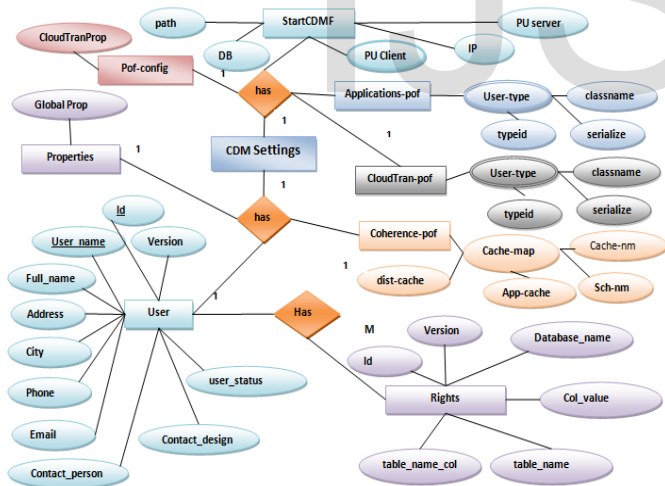


Fig. 2. ERD for InFraMegh.

4.2 Process Design

Process Design includes the logic involved in the framework separately for Configuration Management and Data Access Management. In Configuration Management, two entities are designed. Entity startCDMF interacts with the user. The entity generateClassSchema is designed to provide logics of operations/methods. This entity works at middle layer i.e. between front end and back end layer during implementation. It provides methods which are called and executed automatically by startCDMF. Data Access Management

includes an entity named dbCacheInterface. The purpose of dbCacheInterface is to provide the library for manipulating data in cache. Developer of Cloud based transactional application will use this library for managing data. Fig. 3 shows the class diagram comprising both the layers i.e. Configuration Management and Data Access Management. Table 1 contains the list of methods for entity startCDMF. The purpose of this entity is to interact with developer user. It will ask user either to choose default values or specify values for configuration items stored in all xml files. It will take table names from database and updates tags in files corresponding to tables. Table 2 contains methods related to entity generateClassSchema. It provides methods to create table based entries for entities like cloudtran-pof-config, coherence-cache-config, application-pof-config, persistence. Its methods will be called and executed automatically by startCDMF. Table 3 shows the details of tasks to be performed by methods of entity dbCacheInterface. These methods store/retrieve data to/from the database/cache.

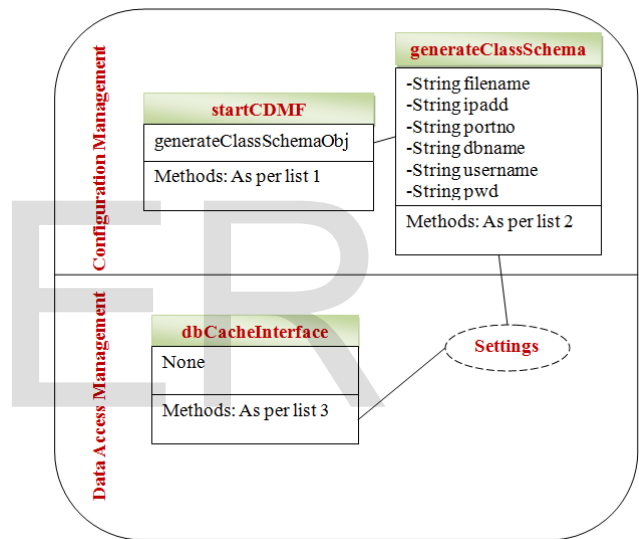


Fig.3. Class Diagram of InFraMegh.

The workflow of Configuration Management has been depicted in Fig. 4. It has mainly four verticals; setting initial configuration items based on all tables of the database, on the basis of any table added or deleted and creating users and assigning their rights. Inserting and deleting table entries vertical allows to make changes in configuration files on the basis of individual tables. During initial setting, parameters like server address, port, database name, persistence unit-server name, persistence unit client name etc. are set in their respective configuration files. Another major task of this vertical is to create two java classes for each database table created for a transactional application. One class is required for reading and writing data from/to cache. While another class file is required for reading data from the database directly. CloudTran does not provide access to read data directly from the database. Data from the cache or the database is read and written through these two java classes. These classes are invoked when developer starts development of any transactional application. Second vertical is followed

when developer wants to include more database tables into already set project. Third vertical is followed when developer wants to delete any database table from the running object. The fourth vertical creates users and assigns rights.

Data Access Management has commonly used methods required by developer of any transactional application using InFraMegh. These methods will be called in transactional applications as and when required. Therefore, no specific flow exists among them.

TABLE 1
METHODS OF STARTCDMF

```
public void setPara (String filename, String ipadd, String portno, String dbname, String useuname, String pwd, String tname, String pathname)
public String setInitialValues(String iname, String ival)
public String getInitialValues(String ival)
public void createPersistence()
public void createApplicationConfig()
public void addTablePersistence(String tn)
public void removeTablePersistence(String tn)
public boolean checkTablePersistence(String tn)
public void addTableApplicationConfig(String tn)
public void removeTableApplicationConfig(String tn)
public boolean checkTableApplicationConfig(String tn)
public void createCloudtranConfig()
public void addTableCloudtranConfig(String tn)
public boolean checkTableCloudtranConfig(String tn)
public void removeTableCloudtranConfig(String tn)
public void createCoherenceConfig()
public void addTableCoherenceConfig(String tn)
public void removeTableCoherenceConfig(String tn)
public boolean checkTableCoherenceConfig(String tn)
public ArrayList tableListing()
public String userRightsSave(String userRightsFlag, long selectedId, long selectedVersion, String userName, String userDB, String userTB, String userCol, int userVal)
public String userCreateSave(String userRightsFlag, long selectedId, long selectedVersion, String userName, String userFull, String userAdd, String userCity, String userPhone, String userEmail, String userCP, String userCPD, String userStatus )
public String[][] userNamesListing()
public String[][] userRightsListing()
```

TABLE 2
METHODS OF GENERATECLASSSCHEMA

```
public static void addComponentsToPane(Container pane)
private static void createTabbedPane1(JTabbedPane jTabbedMyPane, Container pane)
private static void createTabbedPane2(JTabbedPane jTabbedMyPane, Container pane)
private static void createTabbedPane3(JTabbedPane jTabbedMyPane, Container pane)
private static void createAndShowGUI()
private static void createClasses()
private static void createPer()
private static void createAppIn()
private static void createCloud()
private static void createCoherence()
private static void setValues()
private static void saveValues()
private static void setLabel1(boolean tf)
private static void setLabel2(boolean tf)
private static void setLabel3(boolean tf)
public static void main(String[] args)
```

5. FEATURES OF IMPLEMENTATION TOOLS

To develop the Integrated Framework for Data Management of Transactional Application in Cloud, InFraMegh, following tools will be used.

5.1 Java

Java 1.7 will be used to implement the framework due to its characteristics of JPA (Java Persistence API) that defines mapping of application object to relation database [10]. Java is Object Oriented Language and supports interoperability.

5.2 Oracle Coherence

Oracle Coherence is an IMDG tool which reliably manages data objects in memory across many servers in a cluster. All data is stored in the RAM of the servers. Servers can be added or removed easily to increase the amount of RAM available. Hence distributed nature of IMDGs ensures horizontal scalability. IMDGs use intelligent, distributed caching to give better performance, high scalability and reduced latency than existing databases [11].

TABLE 3
METHODS AND THEIR TASKS

Methods and Their Tasks	
Method public Collection getTableData(String tname, String cri, Object o)	Task This method retrieves data from the database based on parameters passed and returns records as Collections to the calling method.
Method public Collection getTableDataFromCache(String tname, String cri, Object o)	Task This method retrieves data from the cache based on parameters passed and returns records as Collections to the calling methods.
Method public void put_in_cache(String tname, Object o, Long v)	Task This method puts data based on table name passed into cache.
Method public String remove_object(String tname, Long v, Class type)	Task This method removes records from table based on a key. Class type contains java scheme file.
Method public String saveData(Object obj, String clsName, String tname)	Task This method saves records in cache.
Method public Collection getTableData(String tname, String fld, String fldValue)	Task This method retrieves data from table based on a particular field and its value.

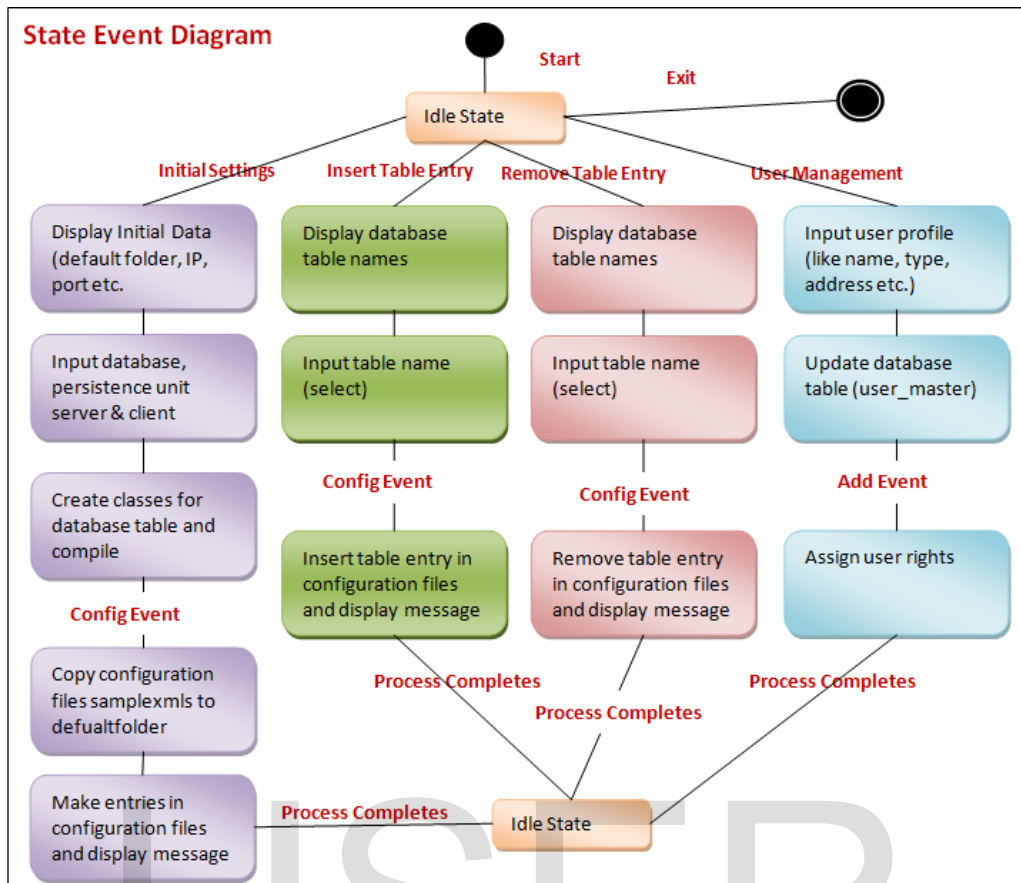


Fig. 4. Work Flow of Configuration Management.

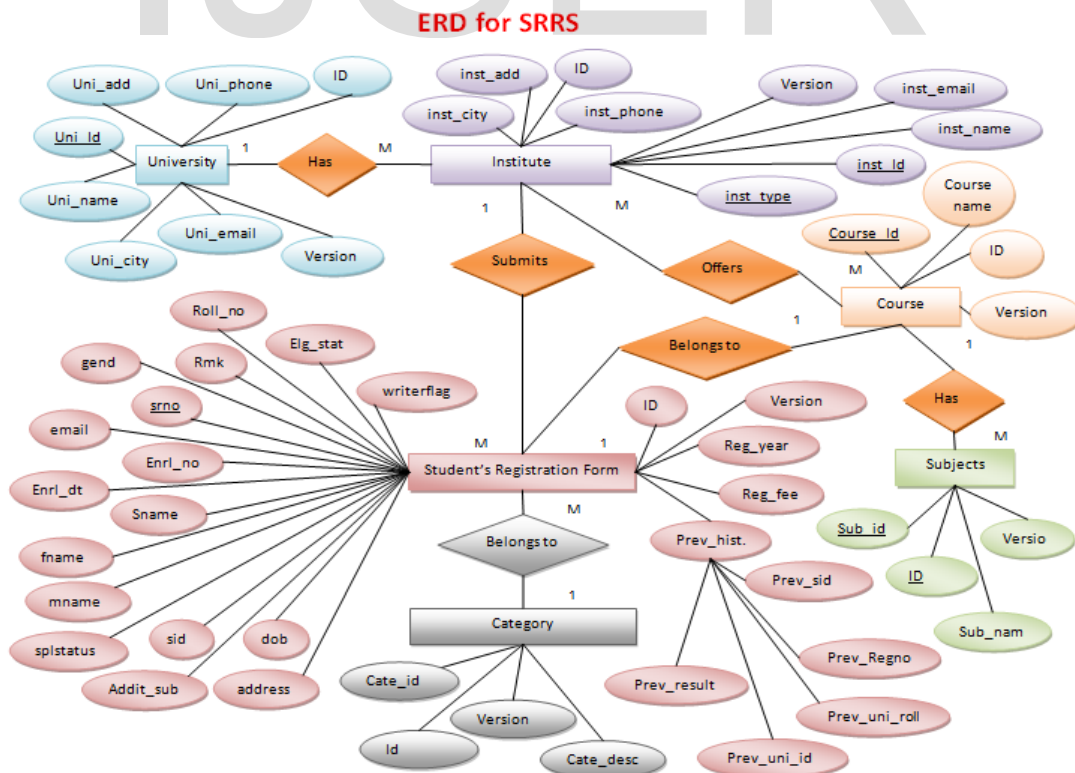


Fig. 5. ERD for SRRS Application.

5.3 CloudTran

CloudTran is a middleware product that enables Java developers to build scalable commercial applications for the Cloud quickly and easily. CloudTran works on the top of Oracle Coherence. It also uses the features of Oracle Coherence. Generally, the data size is bigger than the memory size of cache node; hence data is partitioned into multiple sets. The complete set of data is stored across the nodes in a cluster. CloudTran stores all the data required for an application in IMDG [12], [13], [14].

5.4 MySQL

Popular Relational Database Management System, MySQL will be used as backend database for implementation of proposed framework due to its advantages over other RDBMSs like no licensing cost, open source and It is not confined to a particular operating system. It runs over more than 20 operating systems [15].

6. DESIGN VERIFICATION

To verify the design of InFraMegh, a pilot application of educational institutes has been taken and designed. Education sector like any other sector uses transactional applications for the purpose of fast, efficient and reliable data processing. Student Information System, Fee Management, Library Management, Payroll and Personnel Information, Financial Accounting and Store Keeping applications are the commonly used transactional applications in educational institutes. An application of Student Registration Return System (SRRS) commonly used in educational institutes is used as a pilot application to verify the design for Integrated Cloud Data Management Framework for Transactional Applications in the Cloud. Most of the universities have affiliated colleges and departments where students are enrolled and their registration data is sent to the respective universities. Currently, affiliated colleges submit data of enrolled students for a particular session in the prescribed format as hard or soft copy to their respective universities for further processing. Few universities are also using browser based software to get this data from the affiliated colleges. A general discussion was held with lecturers and administrative staff of colleges of different universities. This interaction helped in understanding the flow of information and data exchanged between universities and colleges. Based on study, an ERD has been prepared and shown in Fig. 5.

7. CONCLUSION AND SCOPE FOR FUTURE WORK

This paper proposes the model and design of an Integrated Framework for Data Management in Cloud, InFraMegh. The middleware tools like CloudTran and Oracle Coherence form the basis for the designed framework. After implementation, the framework, InFraMegh will reduce the complexities in developing transactional applications for Cloud and will provide immense help to developers in developing transactional applications for Cloud for any domain with minor modifications. The future work will include implementation of designed framework in Java with proposed tools. Different

tools will be used like CloudTran, Oracle Coherence, MySQL and Java. Integration of these tools will be a major concern during implementation of the proposed design of InFraMegh.

REFERENCES

- [1] M. Mircea and A. I. Andreescu, "Using Cloud Computing in Higher Education: A Strategy to Improve Agility in Current Financial Crisis," *International Business Information Management Association (IBIMA)*, vol. 2011, 2011
- [2] Daniel J. Abadi, "Data Management in the Cloud: Limitations and Opportunities," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol.32, no.1, pp.3-12, 2009
- [3] Indu Arora and Dr. Anu Gupta, "Usage of In-memory Data Grids in Development of Transactional Applications in Cloud," *International Conference on Advanced Computing and Communication Technologies (ICACCTTM-2013)*, pp. 62-66, 2013
- [4] Rick Cattell, "Scalable SQL and NoSQL Data Stores," *ACM Special Interest Group on Management of Data (ACM SIGMOD)*, New York, USA, vol. 39, issue 4, pp.12-27,2011
- [5] R.D. Johnson and D.Reimer,"Issues in the Development of Transactional Web Applications," *IBM Systems Journal*, vol. 43, issue. 2, pp. 430-440, 2004
- [6] Marcia Kaufman, "The Challenge of Managing On-line Transaction Processing Applications in the Cloud Computing World," *A Hurwitz Whitepaper*, <http://www.cloudtran.com/pdfs/CloudOLTP.pdf> 2014
- [7] Barkha Bahl, Vandana Sharma and Navin Rajpal, "Boosting Geographic Information System's Performance using In-Memory Data Grid," *BIJIT, BVICAM's International Journal of Information Technology*, ISSN 0973 - 5658, vol. 4, no. 2, pp.468-471, 2012
- [8] "Oracle Optimized Solution for WebLogic Suite: An Optimal In-Memory Data Grid Architecture," *An Oracle White Paper*, March 2011 <http://www.oracle.com/technetwork/articles/systems-hardware-architecture/oo-soln-weblogic-11g-170238.pdf>, 2013
- [9] Steven Robbins, "RAM is the new disk...," <http://www.infoq.com/news/2008/06/ram-is-disk>, 2013
- [10] M.Sasi and Dr. L.Larance, "Analysis and Design of Mobile Cloud Computing for Online banking Application System," *SSRG International Journal of Mobile Computing and Application (SSRG-IJMCA)*, vol. 1, issue 1, pp. 21-25, 2014.
- [11] www.oracle.com/technetwork/middleware/coherence [last accessed on October 2, 2013]
- [12] CloudTran, "CloudTran Technology Genesis," *A Technology Whitepaper by CloudTran Inc*, 2011, <http://www.Cloudtran.com/pdfs/CloudTranTechnologyBrief.pdf>, 2013
- [13] Robert Herber, "Distributed Caching in A Multi-Server Environment," <http://www.idt.mdh.se/utbildning/exjobb/files/TR1080.pdf>, 2013
- [14] Paul Colmer, "In-Memory Data Grid Overview", <http://highscalability.com/blog/2011/12/21/in-memory-data-grid-technologies.html>, 2013
- [15] Ion-Sorin Stroe, "MySQL as a Part of the Online Business using a Platform based on Linux", *Database Systems Journal*, vol. II, no. 3, pp.33-12, 2011.