Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Big Data Management for MMO Games and Integrated Website Implementation

Abdullah Alqwbani[1]

[1] Central Suth University

Received: 9 December 2013 Accepted: 2 January 2014 Published: 15 January 2014

Abstract

With the popularity and success of massively multi-player Games (MMOGs), the development of MMOGS has got a quantum leap on game's contents and entertainment which attract huge number of players making MMOGs these years a big business which increased to billions of dollars revenue each year worldwide. But with this number of players and these game contents, the data volume produced from games has rapidly increased and used by simultaneously game players around the world. This data require high performance, fault tolerance and scalability. Considering all these demands the popular used relational database becomes a big challenge and cannot overcomes the challenges and cannot meet the requirements for MMOGS data storage.This paper focus on using big data technology tools to completely meet the requirement of MMO games. My work can be divided into two parts: the first part we proposed Cassandra database for MMO games data storing and the integration of Hadoop with Cassandra nodes for high performance in operations process. The second part: we implement a new MMO website with new payment methods, new advertisement program by friend's invitations and other enhanced function.By implementing this website and comparisons of results of our database management, we show the applicability of our approach as well as the relative performance benefits of designing new games or website using our architecture.MMO games, big data, electronic websites, NoSQL, Cassandra.

Index terms— MMO games, big data, electronic websites, NoSQL, Cassandra.

1 Introduction

MOG stands for Massive Multiplayer Online Game. MMOG is the kind of online game that allows thousand of players to play simultaneously, from all parts of the world. Players which play these kinds of games are able to stay in the game over longer time of period which is typically 6-12 months. During this time they are able to develop their own game style. Even if a player stops playing the game the servers will still be online with other players. Online games are a blooming market with many opportunities as well as challenges for game developers [1].

In the history of MMOGs of role playing type with poor graphics, and poor contents and easy data management module, this was not because of the non interest from the developers, but because of the nowadays, when broadband connections are increasing by number around the world as well as powerful PCs available for reasonable prices, more computer graphics can be applied and more contents of game could be raised including database models which lead to the huge popularity of MMOGs around the world.

The popularity MMOGs has increased on various platforms in recent years but their development process has not kept pace. Some parts of the development process are still having the manual performance in the field of game content development process. The modern games have turns from simple applications to very sophisticated

information systems. But however the game content is they still have a common of tracking user's actions and response and understanding users' feedback.

In the case of large MMOGs systems it still has difficulty making correct decisions without reliable player feedback, and game providers still have scalability problems to meet growing populations of users. Because the popularity of the gaming environment have attracted more game users, Resulted from that the data generated through such multi-player network games are increasing rapidly, Addressing that The MMOGs struggle to achieve much more than 500 transactions per second [2].

And this transaction rate cannot be improved by adding nodes in cluster as what have been used in some game database systems. With the increasing number of players and concurrent database access, the RDBMS becomes a bottleneck in a large scale MMORPG.

In the age of big data and it's new technologies modules and software we could say MMOGs is also should benefits from these technologies.

Using NoSQL database and integrating it with Hadoop file system will be the proper solution for data management and data handling to improve speed, reliability and scalability of these large gaming applications. In this paper our work focused on two parts: the first we proposed new approach using NoSQL database for MMO log data, game data, and state data for processing, storage, and transmission among databases, servers, and players. The second we implement an MMO games website using traditional database RDMBS to handle account data. This website also has enhanced and new functions can achieve more M bandwidth limits and the rear of technologies. But users and has more flexibility on advertising and payment in this site. In the first part we chose Cassandra for our NoSQL database read and write and Hadoop over Cassandra for analytics work. In the second part the account data does need high data processing performance so we chose to use MYSQL and PHP for our website implementations.

2 a) Motivation and problems

Each and every day, the gaming population as a whole generates terabytes of information. This is due to the multiple platforms that exist for both online and offline gameplay these days. More importantly, each and every time they play, gamers leave massive trails of digital breadcrumbs. These large pools of unstructured data can be analyzed. Coupled with information garnered from social networks revealing a player's reallife preferences, the data can provide valuable insights to help gaming companies tailor the game to more closely match the gamers' profile. There are far fewer vendors focusing on the computer and MMO games, and no single analytics provider appears to focus on delivering game platform analytics **??**3].

The sale of in-game products provides a serious revenue stream for the gaming industry. And big data enables companies to deliver tailored and targeted in-game advertising that speaks to the needs and wants of the individual player. Additionally, through big data, companies can adopt Amazon's successful ecommerce model by recommending virtual products based upon what other gaming customers bought.

The main issue is that the existing architectures of MMORPGs using RDBMS to manage data, which limits the availability and scalability. Also have no enough use for analyzing all these data generated by all users around the game and benefits from them to improve game functions and get more profits.

Happy Farm has 228 million active users and 23 million daily users [4]. World of Warcraft has over 11 million monthly subscribers worldwide. MMOGs pose unique data scale and rate challenges. MMOGs generate and manage massive amounts of information; for example, the database logging user actions for Everquest2, a popular MMOG, stores over 20 new terabytes (TB) of data per year. Other projects such as CERN's Large Hedron Collider or the Sloan Digital Sky Survey produces data orders of magnitude larger than MMOGs, but these projects are using large and preprovisioned (expensive) computational and data infrastructure that game companies cannot afford. Furthermore, the data production rate for these other projects is stable over time spans of days or even weeks, whereas for MMOGs the daily user activity has peaks and may even change hourly [5].

The global games market is this year expected to reach $70.4 Billion, an increase of 6% from 2012. MMO games will account for $14.9 billion, or 21.2%, of these global revenues. The Asia Pacific region (APAC), which is now the largest games market in the world, generates 34% of global game revenues.

APAC accounts for an unprecedented 64% of revenues when focusing on MMO games, highlighting the immense popularity of MMO games in the APAC region relative to the rest of the world. North America (NAM) and Europe, Middle East and Africa (EMEA) account for 17% and 16% of MMO revenues respectively with Latin America (LATAM), which grew by 15% since 2012, accounting for just 3% [6].

3 Big Data and mmo Games

Big data has many definitions we chose the common one for describing big data in our paper "Big data is a collection of data sets so large and complex which becomes difficult to process using traditional database management tools or traditional data processing applications in reasonable amount of time" [7].

For Storage such as Amazon S3, Hadoop Distributed File System. For MapReduce programs such as Hadoop and Hive, Pig, MapR and For Visualization GraphViz, Protovis, Google Fusion Tables. Big Data becomes a use case for multidisciplinary problem solving [8]. In MMO games Variety means Game event logs, user profile data,

social interaction data captured during games between players. And Volume and Velocity means points of data collected from millions of monthly users.

Massively Multi-player Online Games have emerged as a most intensive data application nowadays, being massively used by simultaneously game players around the world. This data require high level of performance, fault tolerance and scalability [9]. The term of game can be categorized in general to Casual/flash game portals, Multiplayer web game portals, Browser-based MMO games, Client-based MMO games, Social games [10]. What makes these MMOGs so addictive is the sense of meaningfully interacting with thousands of people halfway across the globe **??**11].

Big Data approaches to gaming are held to be desirable for several reasons. One is because they afford a numerical (and therefore more "scientific") approach to the study of human behavior. Additionally, it is believed that, because of the character of virtual worlds' design, all players' actions are recorded in at least some form, so the database can be seen to yield an inclusive, more or less complete record of activity. Consequently, some major problems of sampling are believed to be obviated, so that Big Data analyses of traces from virtual worlds make for an easy, complete, and quantitative approach to the understanding of social phenomena [12].

NoSQL systems are designed to capture all data without categorizing and parsing it upon entry into the system, and therefore the data is highly varied. SQL systems, on the other hand, typically place data in welldefined structures and impose metadata on the data captured to ensure consistency and validate data types. MMO games generate huge data that need to be analyzed at different levels and for different purposes, from high-level analysis of the number of players in a MMO game allocation to the detailed analysis of the user mouse clicking and keyboard typing behavior for audit and analysis purposes. Usually, a replica of the data to be analyzed needs to be created, which raises the problem of maintaining consistency between the original and the replica(s). Similar to other cases of information replicas in distributed systems, creating exact copies of the data for analysis purposes may not be only expensive, but also unnecessary [13].

4 a) SQL Database vs. NoSQL Database for MMO games requirements

In this paragraph, I will make a simple comparison between SQL database and NoSQL database considering the most needed features in big applications like MMO Games. There are many differences between SQL and NoSQL database from different aspects. In a RDBMS stored structured table, the data in the table can be queried by using standard SQL language, which NoSQL don't have. NoSQL is a summary term describes a set of non-relational databases, may scale out horizontally to a very large size. NoSQL stands for \not only SQL". So NoSQL is not about to do not using SQL any more, but not only limited to SQL. NoSQL is not a terminator of SQL, but an alternative or enrichment to the SQL World [14].

Generally, SQL is not in minority compare to NoSQL, It is still the first choice for most database problems. SQL database exists for a very long time and almost everyone who works related with programming is familiar with relational database. In addition with the development of many extensions of SQL systems, working with SQL system becomes more and more easy. In the SQL database field, it has a big amount of mature products and a large number of tutorial, support, etc. available. That is why SQL database still remains the first choice position for all database problems. We will now give a short comparison from aspects of scalability, performance and consistency between SQL database and NoSQL database.

i. Scalability Scalability is one important advantage that comes from distributed database of no SQL and came to be used to remove the disadvantage of RDBMS. In the horizontal scaling of a SQL database requires administrative overhead after it scale out to a certain size, the performance of the scaled SQL database will decrease. While on NoSQL don't need a table structure that makes NoSQL particularly suitable for scaling out. The architecture of many NoSQL databases is also running on inexpensive computers for data storage. Normally, the NoSQL system can reach a very high scalability by simple adding new nodes into the cluster, even during the runtime. NoSQL systems can provide well and stable scalability constantly despite of the high volume of data. So the NoSQL database is superior compare to SQL database with the aspect of scalability [15].

The NoSQL databases have less schema model and no joins that makes NoSQL databases are efficient from the SQL databases. One of the reasons why NoSQL exists is that SQL systems have limited performance when it extends to a certain scale. The performance of NoSQL is much better than SQL of writing or reading. When the data volume increase, the contribution of a schema and scalability to the performance of the database is more obvious. But it should be understood that not all NoSQL databases are created alike where performance is concerned [16].

ii. Consistency In order to return the up-to-date value to all users, a number of conditions must be met. Some middle-ware appliances (Such as CloudTPS for Google's BigTable and Amazons SimpleDB) also exist, which are adding full ACID features to some NoSQL systems. In this type of database, it can determine which operation is governed by the ACID and which is by "Eventually consistency". But even so the NoSQL in terms of consistency is still not as strong as SQL, because SQL database has an absolute consistency which means it doesn't allow any inconsistent state exists. Some of NoSQL systems still generally do not provide ACID transactional properties: updates are eventually propagated, but there are limited guarantees on the consistency of reads [17]. So when

choosing database users have to decide the proper one for database to have fully powerful database for their systems.

In here I will summarize some advantage of no SQL database 1. The ability to horizontally scale "simple operation" throughput over many servers, 2. The ability to replicate and to distribute (partition) data over many servers, 4. A weaker concurrency model than the ACID transactions of most relational (SQL) database systems, 5. Efficient use of distributed indexes and RAM for data storage, and 6. The ability to dynamically add new attributes to data records.

We can say the NoSQL provide schema less data model to be faster read and write in the database to SQL. Some of NoSQL database available are Cassandra, HamsterDB, CouchDB, Riak and others. All these databases have advantage and difference of storage, performance and availability.

5 System Analysis and Functions a) Proposed System for data base management

Along the development of gaming industry, the games has gain more success and revenue, but with a huge amount of users and advertisement costs, the game still need to be improved to satisfy users amount and companies demands. The technology brought by big data has confirmed that the game can be improved to hold huge amount of data and can be used easily for analysis works.

The website of MMO is the first stage of users to play the game, The users comes for fun to play games so the website should be more beautiful more social and easy to use.

6 ? Account data

Which we used an enhanced platform to perform more useful method for users Account data: this category of data includes user account information, such as user ID, Password, recharges records, and account balance. This data is usually only used when players log in or log out of a game or for accounting purposes. In this part of data we proposed to use RDMS, because the data is not huge amount and does not need a real time analysis and need to be more secured. The figure below shows the architecture we proposed for our system.

7 ? Game data Game data contains: ? Game world data

Data such as world geometry and appearance, object and NPC (Non Player Character) metadata The MMO games data can be divided to two main parts:

(name, race, appearance, etc.), system logs, configuration and game rules/scripts in an MMORPG are generally only modified by game developers. Some significant part of the game data is often stored on the client side to minimize network traffic for unchangeable data.

8 ? State data

As we discussed above, PC (Player Character) metadata, position and state of characters and objects, and inventory in MMORPGs are modified constantly. Currently, the change of state data is executed by an inmemory database in real-time and recorded in a disk resident database periodically.

9 ? Log data

Analyzing user chat history and operation logs in an MMORPG is the most objective and direct way for game providers to evaluate a game, find out the operating habits of players, explore the game development trends, and even supervise the financial system of the game world.

of persistent architectures in use and different game may has its own architecture but there is still a common and standard element since the technology still the same. We did not modify or produce a new architectures model for our system but we have proposed new programs for using in the MMO games architectures model.

As we see in the figure 4.1 we have proposed to use traditional database tools for managing account data. Account data will be integrated through the website system which we developed using MYSQL database since these data still not huge amount and can be managed by using RDBMS efficiently [18]. In the other side game data like log and state data we proposed to use and new technology database system to manage since these data are very huge and need high performance for read/write operation and for analyzing process. We proposed Cassandra for creating database and Cassandra Hadoop for handling database analyzing process. Cassandra is an open source non-relational, NoSQL, column oriented database, and it can store large amounts of unstructured data. Cassandra is peerto-peer model Cassandra's data model offers the convenience of column indexes with the performance of renormalizations and materialized views, and powerful built-in caching. This makes it not only tolerant against single points of failure but also easily horizontally scalable [19] **??** Cassandra is used by Face book as well as other companies, so the code is reasonably mature. Client interfaces are created using Face book's Thrift framework [21].

Cassandra automatically brings new available nodes into a cluster, uses the phi accrual algorithm to detect node failure, and determines cluster membership in a distributed fashion with a gossip-style algorithm. Cassandra adds the concept of a "super column" that provides another level of grouping within column groups. Databases

(called key spaces) contain column families. A column family contains either super columns or columns (not a mix of both). Super columns contain columns. As with the other systems, any row can have any combination of column values. Cassandra uses an ordered hash index, which should give most of the benefit of both hash and B-tree indexes: you know which nodes could have a particular range of values instead of searching all nodes. Cassandra seems to be gaining a lot of momentum as an open source project, as well. For applications where Cassandra's eventual consistency model is not adequate, "quorum reads" of a majority of replicas provide a way to get the latest data. Cassandra writes are atomic within a column family. There is also some support for versioning and conflict resolution.

10 a. Data Model

Model of Cassandra a column is the smallest component of data model, in Cassandra column is the smallest component of data and it is a tuple of name, value and time stamps. Timestamps is to determine the resolution for a multiple version of the same record. Columns are related with certain key can be mentioned as a row, rows may contain a several columns. Column family is stored in separate files by row key order. Keyspace are the owner of column families in Cassandra database.

Replication and Consistency, in Cassandra there is replication factor, which used to ensure no loss from any failing nodes and strategy_options = [{replication_factor:1}]; In this query we used replication_factor:1 which means that any write request will not be considered finished unless at least one server returns success in the entry to it log **??**22]. And for read replica node to return the client request. Cassandra also offers configurable consistency, which provides the flexibility to consciously make trade-offs between latency and consistency. In Cassandra there are two main partitioning ways: random and byteorderd partitioner. The first one is used to in most cases and used hashing to evenly distribute rows across the cluster. Each Cassandra node takes a value that specified the range of keys for which they are responsible. And the second one is used to orders rows by keys.

What we mean is the write and read, in Cassandra to any node in data center for reading or writing the data, these data will automatically partitioned and replicated for them throughout the cluster. Which the write will commit to log durability then to memtable in memory, when the memtable is full, it is turned to an SSTabel (sorted strings table) and writes are atomic in row level, that's mean A write request is sent to all replica nodes, but the consistency level determines how many of them to wait for a write transaction to be considered completed [23]. For a read request, the coordinator contacts the replica nodes specified by the consistency level .

11 iii. Cassandra for MMOGs

Cassandra has a decentralized (peer-to-peer) structure. Each node is identical and is able to initiate reads and writes independently. Data are automatically replicated to multiple nodes [24]. Cassandra has no network bottleneck and single points of failure, which can insure the write performance requirements of state data to MMO game. This is different with RDBMS (e.g., MySQL Cluster) and some other Cloud storage systems such as Google Bigtable which usually adopt a primary/secondary model and may become a challenge. Cassandra provides a column family based data model, which is more efficient than a simple key value store. Every row in a super column family in Cassandra consists of a row key and a dynamical set of super columns, each of which maintains a different number of columns [25]. So we can manage the data of one player in a one row, and partition data based on row key across multiple nodes in the cluster. Hence, there is no more join operation during reading data, and the read performance will be increased. Cassandra adopts a shared-nothing architecture and a simplified data model. So it can scale out easily by adding new hardware, and reach a linearly increasing read and write throughput. That is also the reason that we can manage game data like state and log data in Cassandra. Another advantage is that Cassandra provides a quorum based data replication mechanism. That means as long as a write can receive a quorum responses, it can complete successfully. In this way, Cassandra ensures availability and fault tolerance. Additionally, by controlling the Cassandra employs Read Repair to guarantee data consistency. It means that all replicas must be compared in order to return the up-to-date data to users. In MMO games, state data may have hundreds of attributes and are distributed in multiple data centers. Hence, such a feature will significantly reduce the read performance and increase the network traffic. Note that Cassandra records timestamps in each column, and uses it as version identification. We set all columns in a single row with the same timestamp, so that only one row key and one timestamp are stored for a single row on the server side. In addition, these timestamps are partitioned and managed by access servers in parallel. For these two reasons, accessing these timestamps in the server side is not a challenge.

12 b. Data partitioning

In order to get a high performance of accessing the entire state data of a character or object, we manage these data in a single row and partition them based on row key horizontally. However, this across characters **??**26]. We can partly relieve this problem by creating indexes or semantics.

13 b) MMO Games website and typical features

i. Database design Our system is counted as an e-commerce system with the same database design, so we will not discuss it in detail since the relation tables still the same as any e-commerce site. As we discussed in our proposed system, we will use RDBMS for account database and website implementation. We have used MySQL which is a popular relational database management system. It can provide high performance and stability. It is used by websites developers to build different types web Applications. It was in March 2014 the world's second most widely used open-source relational database management system (RDBMS) [27].

In this section I will introduce the extra tables that will be in the MMO game site to show how the system handle database processes between users. In this figure 4.2 we can see the main tables. The database is designed to handle user or admin queries during the use of websites. The user can log the website and play the game in our site and most of online games site, there is no need for real delivery process, since the product bought by user is delivered to his account in the game. The log table stores user log to the website, the order table stores user's orders and when the user pays successfully the database update this site, in the figure we can see the main function we have developed in this website. We have used popular tools to design and implement our website functions including Ajax, PHP and XML. Ajax stands for Asynchronous JavaScript and XML, which is a collection of many mature technologies The homepage mainly have three functions (login, visit, register) but in details the main page still have other functions like browsing news or products(virtual items) and sending feedback to site managers. After the user login to his account in the main page he can navigate to forum or login directly from the forum login form, he can post threads or make a reply to others and share ideas very easy. He can also manage his profile in the forum like images or signatures and moments. In the website function users can do different functions some of them which we mainly focus on and uniquely introduced in this paper. Users can modify his password or other information, also can apply for password forgets to have an auto mail to let him create new password for his account. User also can browse items in single website of games since the system is able to have more than site to own different games.

14 iii. User's promotions (friend's invitations)

In this section I will introduce unique idea of sharing and making advertising function. From our understanding the more social games or website is the more successful; also nowadays doing advertisement is costing more than the product itself. The idea of this function is in the gaming business and virtual world users can do some activities in order to earn some points (virtual currency). So here we developed a new function for users to earn some points by inviting their friends to the website and games.

The user can copy his unique link and send him to his friends by any communication channel or send this link to any social site, so any registration comes from this link it will be recorded in the database linked to this user who sent the link. By this way the website can In the website page user after logging can click buy button to buy items to his account on the game. User can choose from the general payment methods by credit card or bank transfer or choose our new method using card which can be send to him by email, delivery or company split branches. This function can gain easy success and reduce the cost of the company from revenue share between payment companies. Also it will be easy and convenient way for user to pay by small amount without using banks or credit cards. The more payment methods we have the more users we can attract to pay successfully. The website can have its own distributors for these cards or have any other distributors by sharing some of its revenue among them, which will be for sure less than traditional payment methods share. The figure shows the main process steps of purchasing items on MMO game website.

IV.

15 Design and Implementation

In this chapter we will introduce the proposed idea of using big data technology for game database and using traditional database for accounts data in our a) Game system

For the first point of my proposed idea implementation, we will introduce the database of game which can be used to evaluate the aspect of MMO games. The game as usual uses a server side and client side. The server has id logic, like receiving the commands from clients and sending games states back to clients. I will introduce how we can interact with the new technology introduced in big data.

16 i. Functional Requirements of game system

The focused idea in this paper to have an implementation can show the mean and use of new technologies in big data in MMO games and the difference between the proposed system and traditional systems. The implementation should have a simple architected for game and client server's which can accept high number of clients to log the game. The game data should be stored in distributed database (Cassandra). Game server has side logic to react with client request. The user can connect and communicate with game server via GUI (graphical user interface). The figure 5.2 shows the proposed architecture with Cassandra nodes. In this paragraph I will introduce software used in this implementation in brief. Cassandra has a decentralized (peer-to-peer) structure. i.e., each node is identical and is able to initiate reads and writes independently. Data are automatically replicated to multiple nodes. Therefore, there is no network bottleneck and single points of failure, which can satisfy the

write performance requirements of state data. This is different with RDBMS (e.g., MySQL Cluster) and some other big data storage systems. Cassandra is the best program due to its high performance and scalability, it was written in java and has the most compatibility with Java environment and most of the existing relative mature tools are supported in form of Java libraries. For the communication between server and client we proposed to use darkstar. Darkstar is an Apache project which is designed for MMO game network architecture. To describe our proposed architecture with Cassandra nodes we will show the available hardware resource, we used the architecture below. The figure 5.3 shows the software and architecture proposed in our paper. The client and server communicate via a socket server. The socket server is supported by a project Darkstar. Later we will brief introduce the project Darkstar. Game server stores game logic and manages game state data when the game is running.

The database layer deals with the basic data accessing operations. It is responsible for data querying and inserting in all underlying column family which hosts the game player information, inventory information, user logging, user statistics and game world information in Cassandra cluster. A game player connects to the Darkstar Server by giving their credential information. When the user login is successful, the user can then create or choose a hero to play the game. Darkstar Server is response for calculating the world state and user state in associate with commands sending by players.

iii

17 . MMOGs Database Schema Design in Cassandra

To be clear about game database design in Cassandra and new database system, we will firstly introduce RDMBS of game and secondly explain the difference between Cassandra and RDBMS on design concept and finally present Cassandra as a proposed solution for the game database system.

a

18 . Traditional Game Database Schema

As we know in the RDMBS systems the database introduced in tables for all game data, these tables has a foreign key to refer to related data in other tables. I will introduce the basic entities of MMO games using RDBMS.

19 Table 1 : Simple comparison between relational and

Cassandra models In Cassandra, de normalization is the base. A standard and very efficient way of working with the Cassandra data model is to create one column family for each expected type of query. With this approach, data is denormalized and structured so that one or multiple rows in a single column family are used to answer each query. All data are stored in ordered columns and the columns constitute a row which is uniquely identified by its rowkey. allow you to pre-compute result sets and store them in a single row for efficient data retrieval. Each row is a snapshot of data meant to satisfy a given query, sort of like a materialized view which means each row of a column family could have different set of columns.

In the basic tables of MMO games database we could have these kind of query figure.

20 Get maps

When the game starts, all maps information should be load to the memory, so when the user logs the game and the game has loaded enough resources, it transitions to a menu screen so the user can present his desired maps and start his transactions. To do this we need a query to scan the Map column family by Rangeslicequery and load its information to the memory.

After determining the needed query pattern of the game or at least the most important query will be needed. We will show how to use them in the best practice. We can also add any query we need in future by using CASSANDRA features like indexes and nested indexes. In the world of MMO games, the character of user is the most essential data. It needs high availability and consistency. some of player data should have transaction, like when the user get certain prestige the lord nobility should increase and get a new features like updated we need to store data in one row. That's because Cassandra does not provide ACID properties so (no complex transactions support), but it still provides some useful atomicity guarantees. More precisely, Cassandra has always provided row-level atomicity of batch mutations. This means that multiple batched writes to the same row are applied by nodes atomically. By this feature we can use "slice query" to get a range from start to end columns.

From the tables of MMO games database, if we want to find a hero created by user we can also "slicequery" using starting and ending case, and then any information that we want to get. And if we want to get users skills in the tech system, the skill table for every user is integrated with tech system so we can avoid join, because every user's technique is stored in one row. This is will have highly performance but it may cause redundancy. We can use queryslice with specifying user's id. In the figure below we can see the To Store the log data, we need a table like log column family. The game event log data are stored in this column family. As shown in Figure 5.7 the row key is ddmmyyhh: eventtype and one row represents events within one hour of a day. The server will generate a new row in every hour and all events occurring within this hour will be recorded in this row. The column name stores the time when the event occurred in a suitable granularity, such as seconds. The column value stores the payload. Many events could happen within one hour so that a row in the log column family can

be very wide. Cassandra's row can hold up to 2 billion columns. So we don't need to worry about the capacity of a row. However, in Cassandra, one row will not be split across nodes and stored together in one node. If we simply use the hour as our row key and keep the event data within an hour together in one row, this will be issue:

Firstly, if we have a super large row with millions of columns, the size of this row can be so large that they cannot host in memory entirely. Second, since the data is handled by only one node in cluster, the entire write request will consequently go to the single node which is holding the row for the current hour, and then this node would be probably a hot spot. The event log data in one MMO Games could be very large, so we have added an event type after the row key so that there will be several rows for different event type within one hour. So the write operations for different event types will go to different nodes in the cluster. By retrieving data, we could use a multi-get for an hour from all of the nodes and merging the results in the application. Game player-Factory, Item Factory which are used to generate the random basic data for testing purpose.

? game.Cassandra.gamestates it contains classes that stores some game logic used information and some game logic.

? game.darkstar.task in this package, we have implemented some tasks which are performed periodically.

? game.login.authenticator this package contains classes which are responsible for the authenticate job.

v. Configuration of Cassandra with Hadoop It is highly recommended to facilitate data management and real-time analysis along with complex data intensive processing. Cassandra can run Hadoop configure a Cassandra cluster for Hadoop integration overlay a Hadoop cluster over your Cassandra nodes. This involves installing a Task Tracker on each Cassandra node, and setting up a Job Tracker and HDFS data node.

When a Hadoop Task Tracker runs on the same servers as the Cassandra nodes, each Task Tracker is sent Tasks for data in the local Cassandra node. This causes tremendous gains in efficiencies and processing times, as the Cassandra nodes receive only queries for which they are the primary replica, avoiding the overhead of the Gossip protocol. Using Cassandra with Hadoop leaves the distribution of data to the partitioning strategy set for the input and output column families within Cassandra. Random Partitioner will evenly split the data among the nodes. This means that with Random Partitioner each Hadoop worker has local access to almost the same amount of data. On the other hand, the Byte Ordered Partitioner causes the data splits to be collected on a small set of nodes, hence, the data locality is restricted to a small set of nodes and the remaining workers have to pick up splits remotely. This would cause an extraneous of data movement in the cluster and in turn affect the turnaround time negatively

21 b) Implementation of MMO Games Website

In the second part of our research we implement a website which can host many games in and play or go for the related y website features Like website forum or blogs.

We illustrate the main function of the site and the use of it in the e-business field of MMO market.

i

22 . Game and Website Integration

To integrate web site with game server we need to provide: Website that the user can browse and sign up and login. Game server login button which the user can click on to login the game.

After the user log in the website he can clicks on games button to browse game available in it and log in easily to the game.

The listing below shows the code behind this integration between website and game server. The main integration steps can be done by three functions: After the user log into his account in the website he can use all website function provided by website like blogs, forum threads, browsing items, sharing friends invitations or log into the MMO game available on the website. To log into the game we need to provide authen -tication to the game side from our site, the listing system. Hadoop/Cassandra Cluster Configuration to shows the function of login the game. If the user has logged in the website the function will direct him to the game server and log easily and enjoy the game.

$UID=$USER->getID(); $utime=time(); $key= $token=md5($uid.$utime.$key.$sid); $game-server='http://alqtest.com/?uid &time='.$utime.'&token='.$token.'&sid ='.$sid."; //after clicking log the website button will redirected to http://alqtest.com/?uid={$uid}&time={ $time}&token=${token}&sid={$sid} Listing 1 : Code of login the game from website After logging the game, the user will have character name and level and other information on the game. This information will be stored in the game database for analyzing and development plan for game. But in the website we still need some basic information of the user to pursuit website activities and sending items, the listing shows how to retrieve character name of the user from game database, and the listing shows how to retrieve the character's level.

23 c. Recharge: buying items from website to the game

The user can recharge his account by click the button of recharge and choose product then pay and receive it in his game account with very easy and simple steps.

The flowchart 14 shows the steps of purchasing process in our implementation website and the figure 15 shows the table structure of order table in our database. **??**tructure In this section we introduced a new easy way for

building and interaction platform between game and website to let users stay linked in to the website and insure the Active degree of them by affecting the website activities for example share ideas, Activity-like and other activity which the user can participate on the website and get gold (game currency) or virtual items to the game directly without using game backend manager. By these three interfaces we can easily interact and retrieve most need data from game data. Users or players can easily login by providing their emails and password to enter the website and the game. Player can enjoy game playing, purchasing, chatting etc. ii. Website users register

In our implementation we chose to create two the website by two methods: The first way is basic way by entering visitor information and clicks submit to become a user.

The second one is by using the user social id like Facebook or Google so the user does not need to provide his name or email to register. Registration can be made by the way chosen by the user, for more explanation we can note and follow the steps of registration method through using Flow Chart Diagram as shown in figure 16. When a user is signed in, we get an OAuth token for making API requests on their behalf, which we can use to let the users register fast and do not need to remember any login OAuth and better understand your user, connect them with their friends, and create a richer and more engaging experience.

In this part we would like notify the importance of users registration on MMO games market field, when any website launch a new game or new game server, will launch advertisements on different sites or search engines and will have a great amount of new browsers to his site or his landing pages, so the more easier way to register the more users we will have and of course more benefits the site can make.

Users can be used to stay linked to users and see who not log for certain amounted of time and send him an email to remind him website activities or new servers or games. Using this function can prevent website to lose users and keep them aware of all new contents of the website. The listing shows the general code for sending emails to websites users.

iii. Friends invitations In this function we developed a new way to advertise our website or game without any costs. Any user will have a unique link which he can use it and send it to his friends and invite him to register from this link.

When the user registers for his new account, the system will record the source where he comes from (search engines, main websites or friend's invitations). when the new user have the TID then the system will record the user and his logging time, the same way we can have any information between these users According our understanding and experience the more social is the the game. The figure 17 shows the main database table structure of friends invitations, so the users can check his friend's level and logging time, to stay linked to them.

24 Figure 17 : User friends table structures

In the website management back-end we check user friends by id or emails and see how many friends they have or what server they logged to analyze their progress and send them game items to stay active in the game or the website.

The figure 18 shows the test id of my email having my friends checked from the website management back-end. And in the website page user after logging can click buy button to bay items to his account on the game. User can choose from the general payment methods by credit card or bank transfer or choose our new method using card which can be send to him by email, delivery or company split branches.

This function can gain easy success and reduce the cost of the company from revenue share between payment companies. Also it will be an easy and convenient way for user to pay by small amount without using banks or credit cards.

The figure shows the system back-end page for creating cards keys. We can choose the unit of package item and set the account we want to create then click create to have all cards key. Then we can use the email to send these cards keys or print them to cards the sending them to company branches or handler to sell these cards to users so they can use them easily on the website. The figure 19 shows the page from website which the user can use to choose payment method and pay for his desired item. game or website the more success it can gain, so by using this function the website can be between friends, will make more social and let the users stay longer on Figure 19 : Cardkey Generating interface And in the website page user after logging can click buy button to bay items to his account on the game. User can choose from the general payment methods by credit card or bank transfer or choose our new method using card which can be send to him by email, delivery or company split branches. This function can gain easy success and reduce the cost of the company from revenue share between payment companies. Also it will be an easy and convenient way for user to pay by small amount without using banks or credit cards. The user can enter his cod key as we see in the figure 20 and make his payment very easy. As we see figure 21 here we can see the test of 5 Cassandra nodes performance, when the number of nodes reaches to 5, the performances of Cassandra cluster is the best in all range of clients' number, both of reading and writing response time with 5 nodes Cassandra cluster are about 15ms. With increasing number of clients in 5 nodes Cassandra, there is no obviously variation of reading and writing response time. Evidently, the performance of 5 nodes Cassandra is stable in our test. 5 nodes cluster is obviously the best one.

25 ? Scalability

To evaluate scalability of a MMO game we have to handle that from two aspects: the scalability of game server and database.

From the proposed system above, we can find that with the constant adding game sever, the maximum number of players increases linearly from 100 to 1500, which proves that the game server has scalability. For database scalability we can see scalability of Cassandra cluster in MMORPG.

If we carry out 5 tests to evaluate the scalability of database, and set the maximum number of concurrent players to 1500, the number of nodes of Cassandra cluster is set from 1 to 5. The number of game server is fixed to 3. Each of the game servers is connected by 100, 200, 300, 400, and 500 clients in turn. That means, the Cassandra cluster handles 300, 600, 900, 1200, 1500 clients separately. Every client sent 500 reading or writing commands. In another word, Cassandra cluster needs to handle 150,000,300,000, 450,000, 6000,000, 750,000 commands in turn. more capably and efficiently than other environments ensuring that your business can scale and be successful. So we did choose Cassandra to run our application and proposed it to be used in MMO games application.

26 Global Journal of Computer Science and Technology

In next paragraph we will show the result of using Cassandra nodes and Cassandra-Hadoop nodes.

Base on the analysis and comparisons above, we can conclude that Cassandra can meet the performance demand of MMOGs in general. The more nodes Cassandra has, the more concurrent players Cassandra can support. With the increasing number of players, the reading performance of Cassandra will be improved a lot; the writing performance stays relatively stable.

27 d) Comparison Using Cassandra And Hadoop

Integrations Using Hadoop with Cassandra cluster is obviously much better than using only Cassandra nodes In figure 5.10 we can see the comparison between operation time of Read/Write mix workload is an indicator of throughput, or transactions per second, that can be achieved by an OLTP database environment. For an environment to handle increased throughput is critical as it shows how well the database will handle growing levels of business. The better a database can handle increasing throughput also informs you about how well the database can scale. Cassandra has been proven to handle growing transactions at rates much Using Cassandra nodes only we can see the result performance as figure 22 shows, by using Hadoop-Cassandra nodes it shows different and better result as figure 23 shows By comparing the test results between only Cassandra nodes and Hadoop-Cassandra nodes in performance, and compare these result to our 5 nodes test using 1500 users we can determine that Hadoop integrated with Cassandra can handle operations faster than any system in term of MMO games analysis. In the end of this chapter we would like to insure by using our proposed database system structures and tools the MMOGs will have better performance and better benefits. And by using our website functions like invitation advertisements, payment methods and flexible registration form, the game's website we have better influence and will get more active users as well as gaining more profits.

V.

28 Conclusion and Future Work a) Conclusion

Along with the popularity of MMO games and the huge number of players and developer engaging this field, the promising present success will lead to have a bright future of MMO games business. With the growth of MMO games it will face more challenges. The importance analysis of players to allow developers to understand in real time why users are giving up the game and identify others players at risk of leaving the game so can make their strategies and improvement to prevent all games risks. And the main challenge is the huge amount of data produced rapidly in MMO games, nowadays big data technology and tools has come to applications with promising features. Hadoop (HDFS, MapReduce) and NoSQL are in the top of these technologies.

Hadoop is used in maintaining, scaling, error handling, self healing and securing large scale of data. These data can be structured or unstructured. NoSQL is non relational database to handle huge structured and unstructured data. in this paper we chose these technology in order to meet MMO games requirement and overcome the current challenges. Paper has gone through two parts

The first part: by proposing a new methodology for MMO games database using Cassandra NoSQL database system for MMO games. Cassandra data schema must rely on the query pattern so through increasing data redundancy, Cassandra stores all the data that might be queried together in one column family in order to avoid join operation and improve read performance. The results acquired from our paper shows using Cassandra are obviously much better than using traditional database systems. and when Using Cassandra with Hadoop leaves the distribution of data to the partitioning strategy set for the input and output column families within Cassandra and comparing this integration result to other Cassandra nodes result is shows Hadoop-Cassandra can serve much better on MMO games operations especially the read process since Cassandra is optimized for write operations.

The second part: the aims of this part are to design a MMO games website so that the user can visit and choose game and play. The purpose is to attract high number of customers as much as possible, moreover stay active users and facilitate the payment process to increase the revenue to the website and games.

This part can be done through new functions has been developed in this project 1. New payments method using cardkey to increase the flexibility of payment to users and enhance the sales. 2. New advertisement method using friend's invitations to increase the loyal active users and decrease the cost of ads. 3. Social and flexible registration and activity methodology has been implemented using social id to register and auto mail reminder to help users stay active and loyal to the games and website.

29 b) Future work

As we have seen in our paper the system with high partition tolerance and availability (like Cassandra) will have to lose some consistency in order do its job especially in real-time or high speed queries.

In my plan for future work is to design full functional system with big data technology for online applications including MMO games and focus on data analytics using Hadoop, hive and Cassandra. Hive and Hadoop to perform map/reduce engine, and use hive do queries to reinsert data to Cassandra. The goal of using Hive and Hadoop with Cassandra is to have realtime analytics with high performance, fault tolerance and consistency features. [1] [2] [3]

1

Figure 1: Figure 1 :

[1] © 2014 Global Journals Inc. (US)

[2] © 2014 Global Journals Inc. (US) Global Journal of Computer Science and Technology

[3] © 201 4 Global Journals Inc. (US)

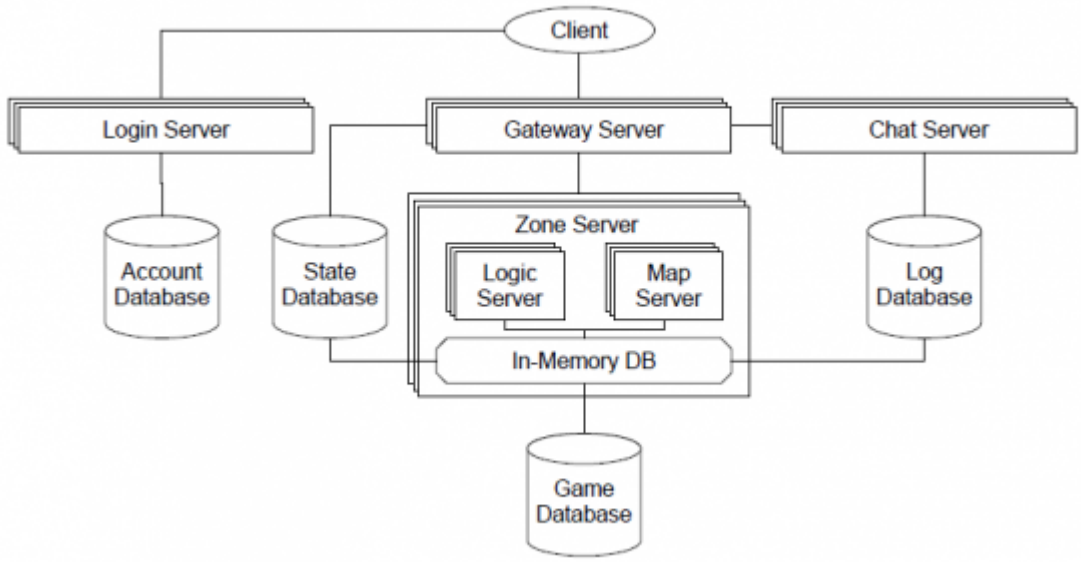
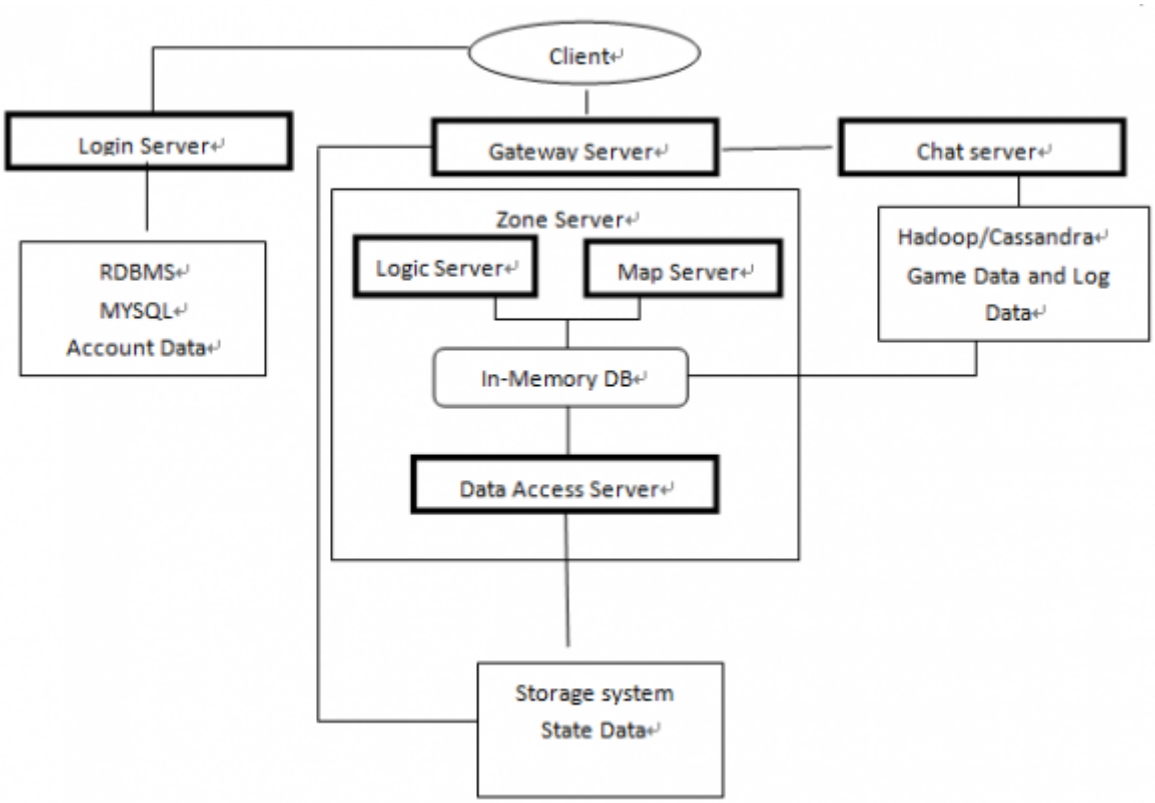


Figure 2:



2

Figure 3: Figure 2 :

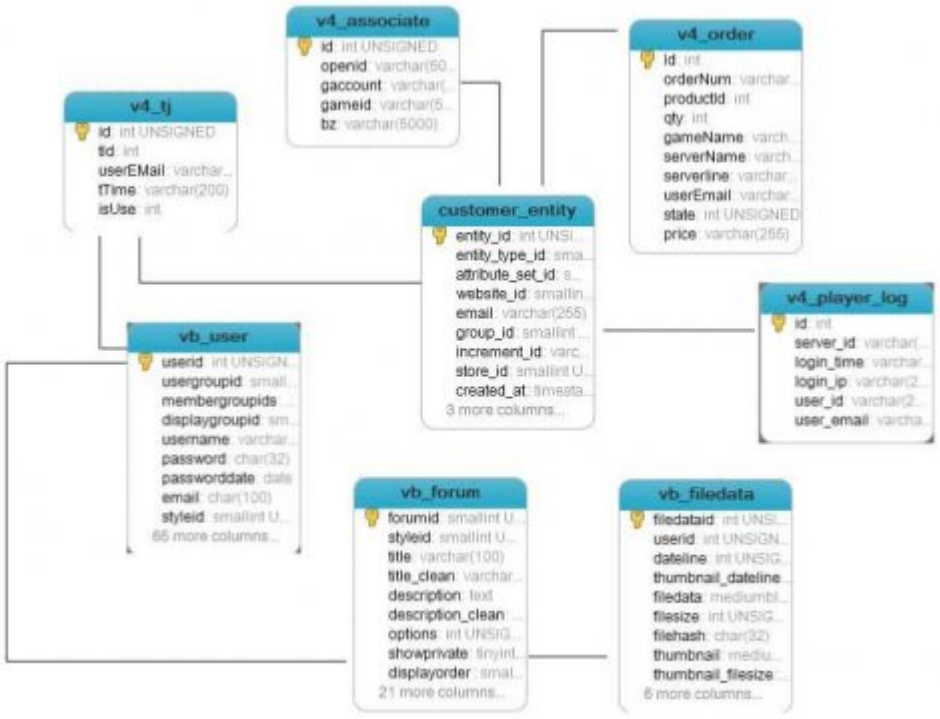


Figure 4:

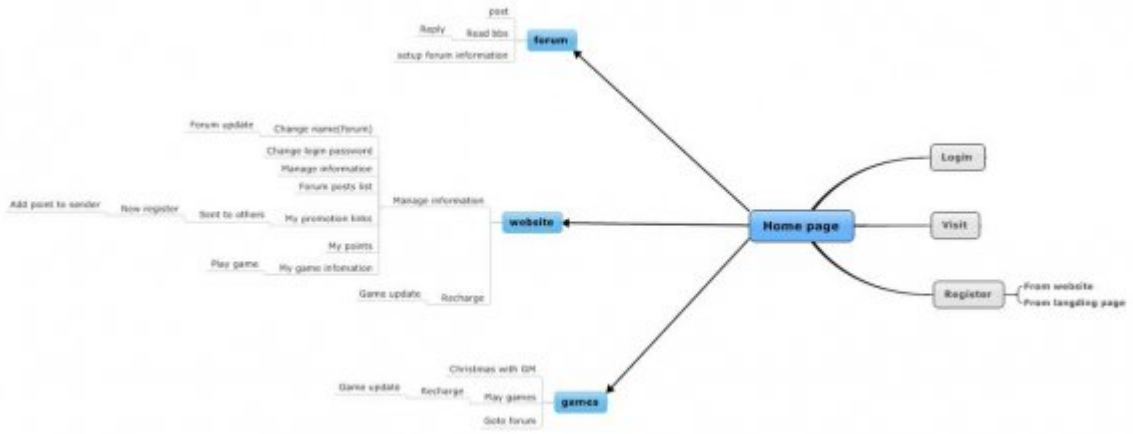
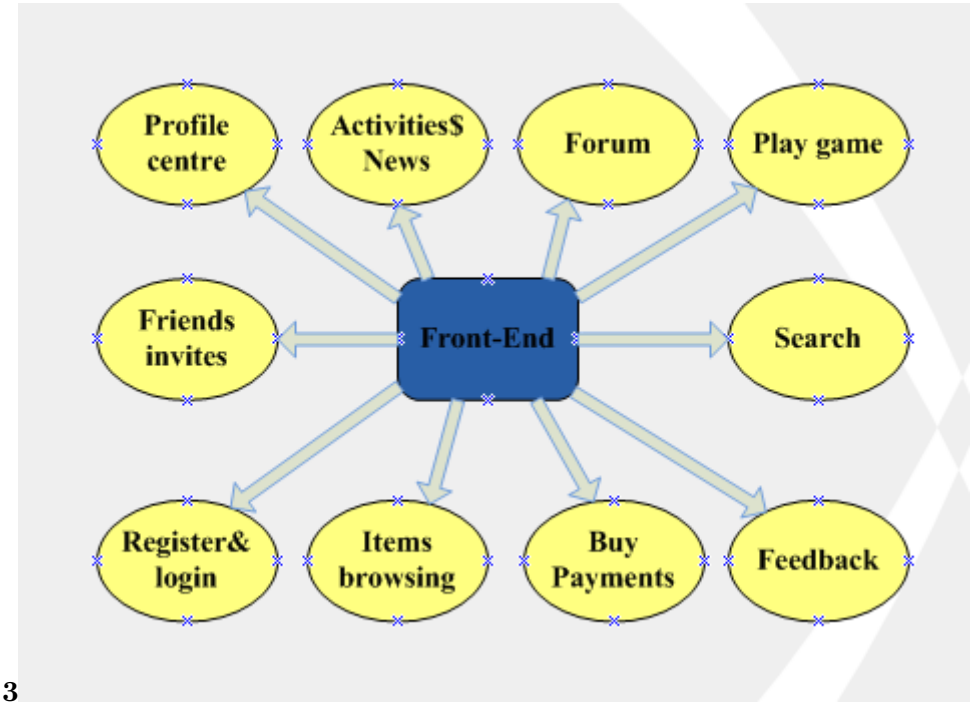


Figure 5:



3

Figure 6: Figure 3 :

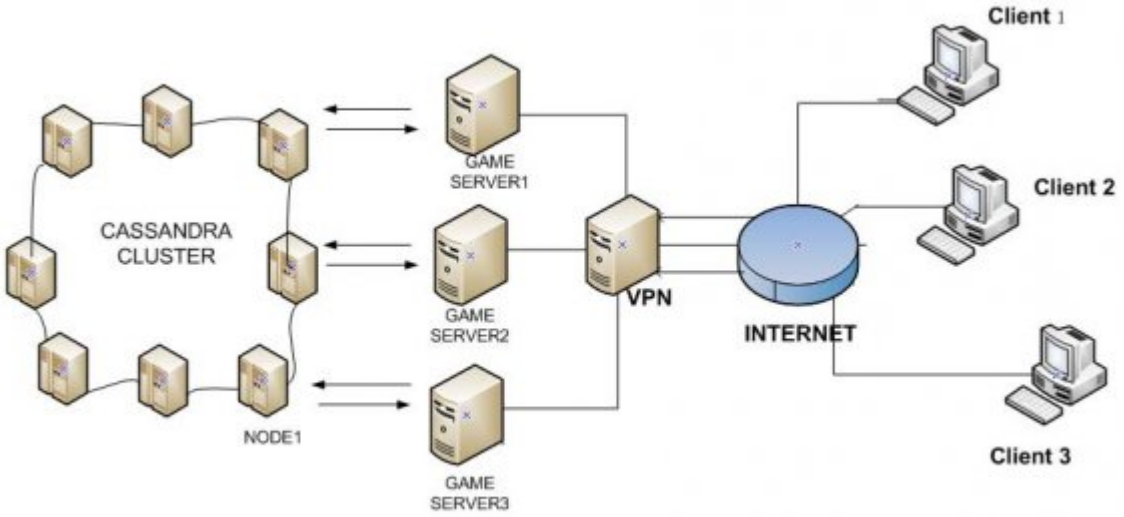
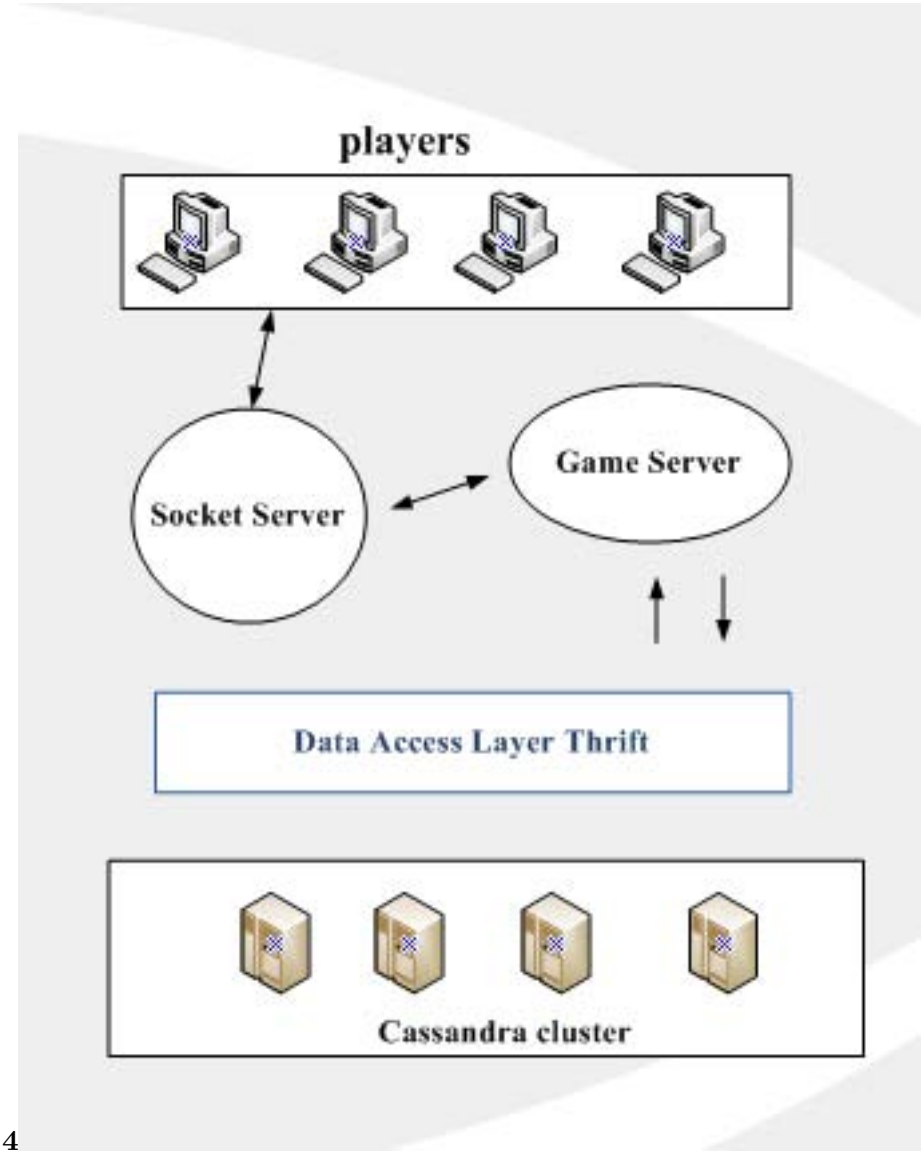


Figure 7:



4

Figure 8: Figure 4 :

Relational Model	Cassandra Model
Database	Keyspace
Table	Column Family (CF)
Primary key	Row key
Column name	Column name/key
Column value	Column value

5

Figure 9: Figure 5 :

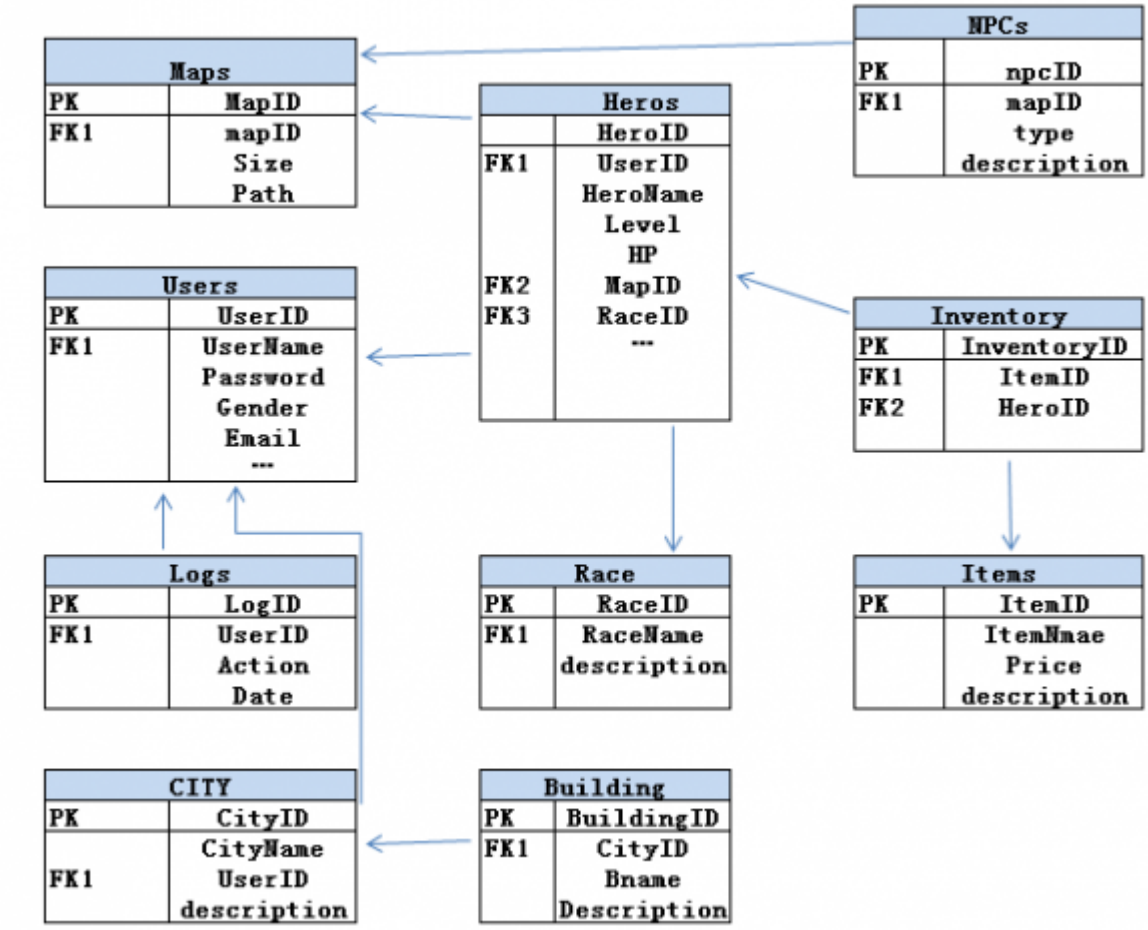
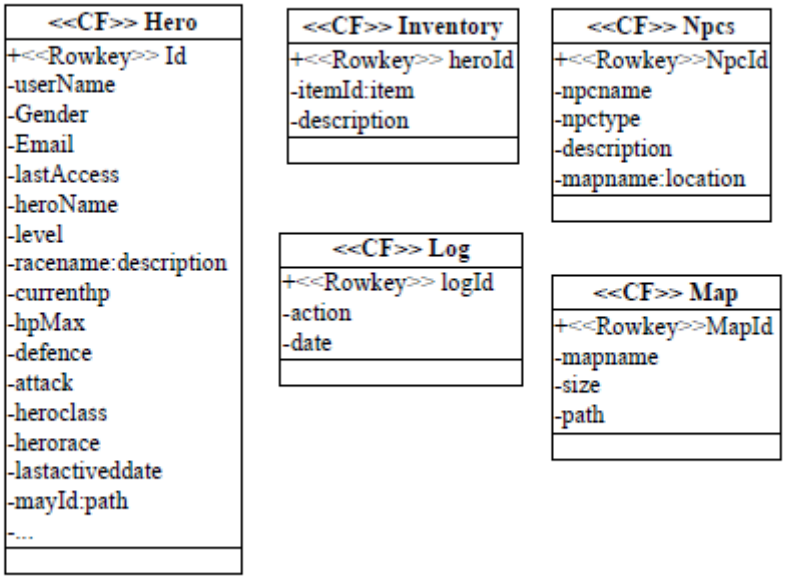


Figure 10:



6

Figure 11: Figure 6 :

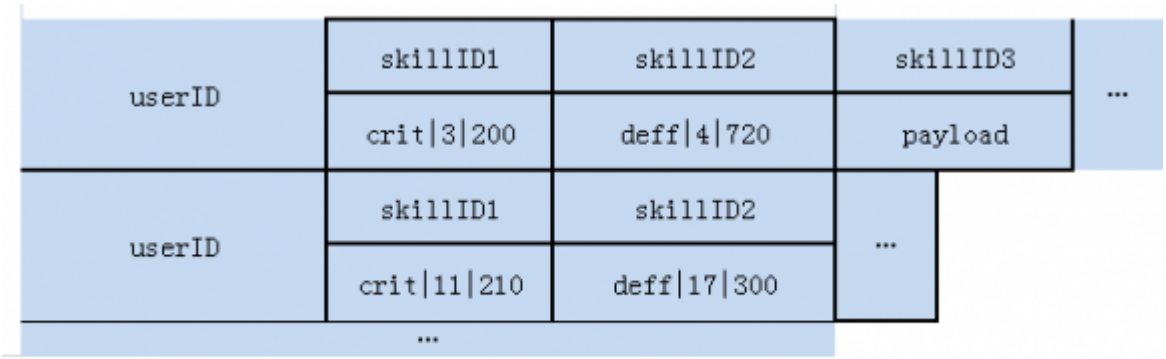
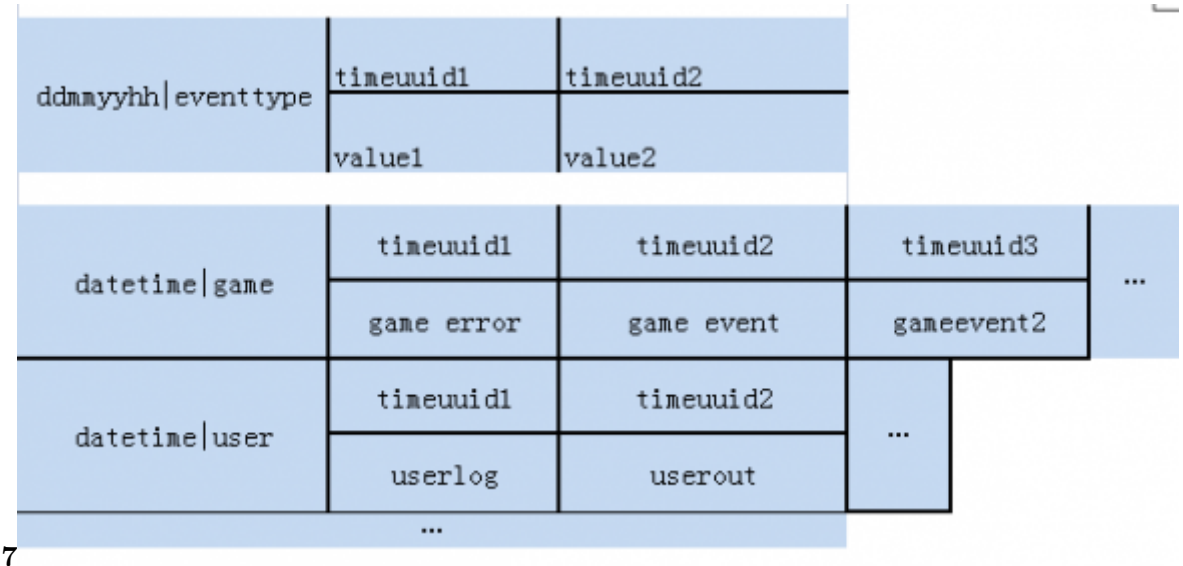
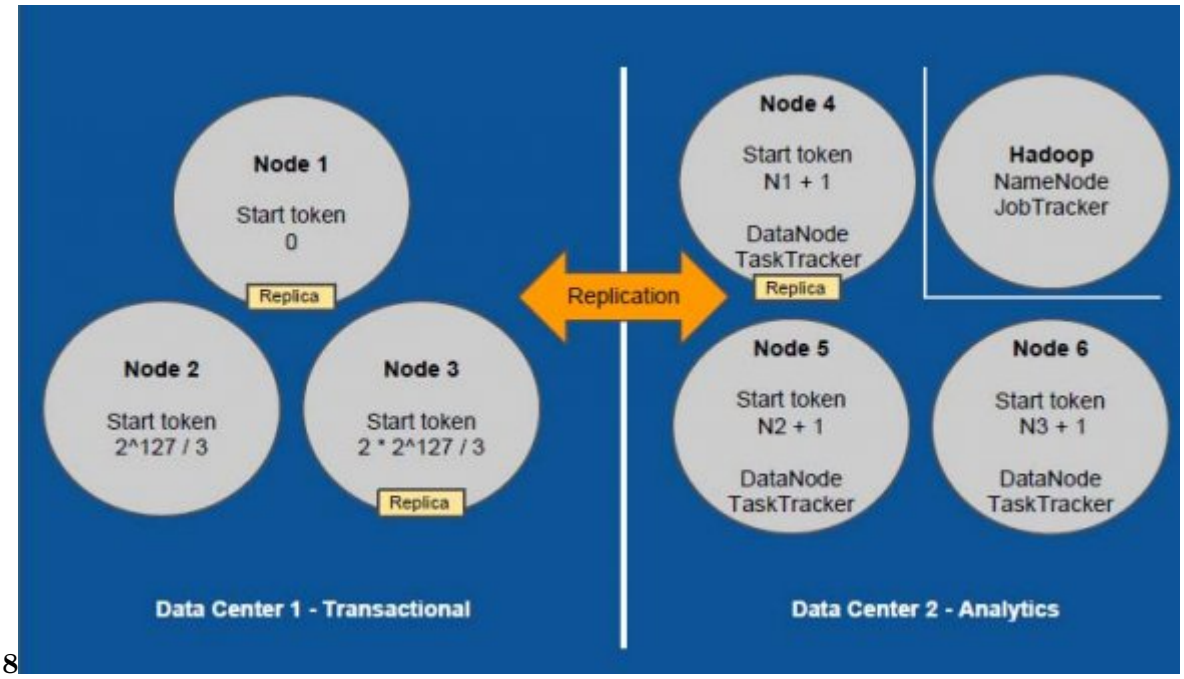


Figure 12:



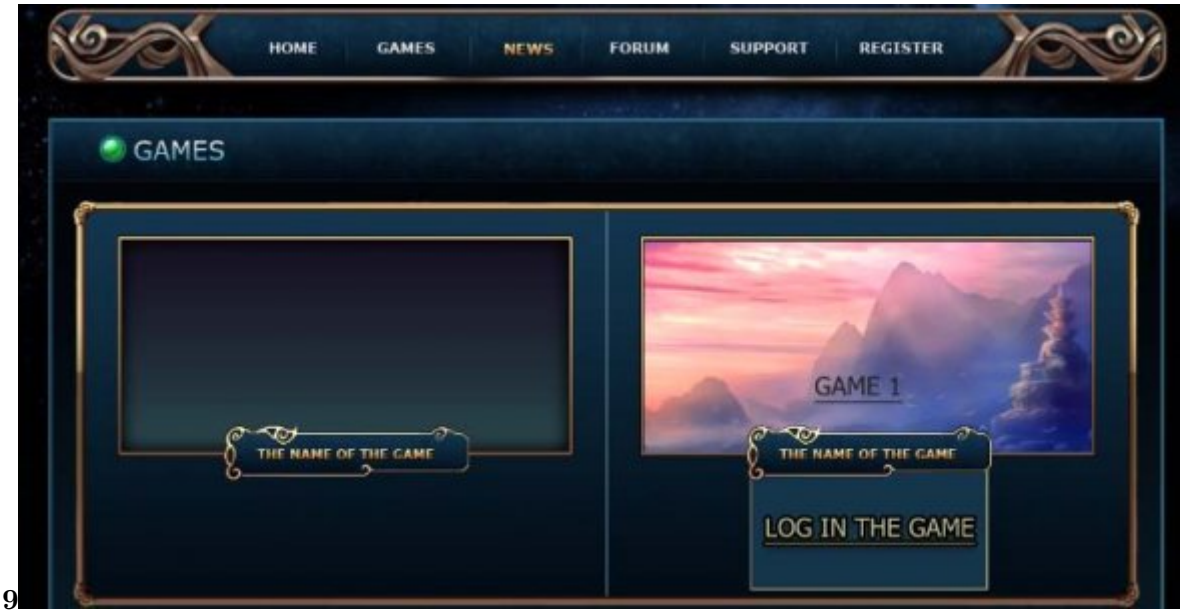
7

Figure 13: Figure 7 :



8

Figure 14: Figure 8 :



9

Figure 15: Figure 9 :

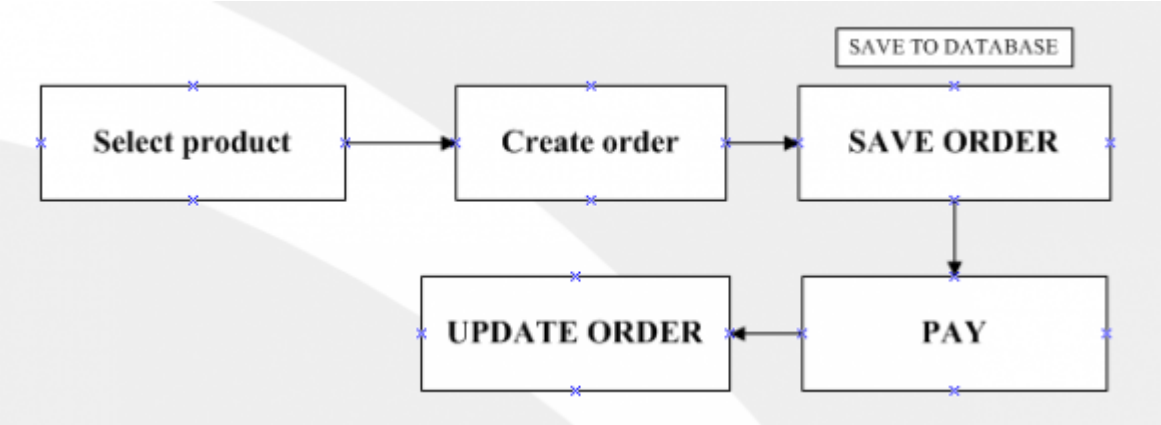
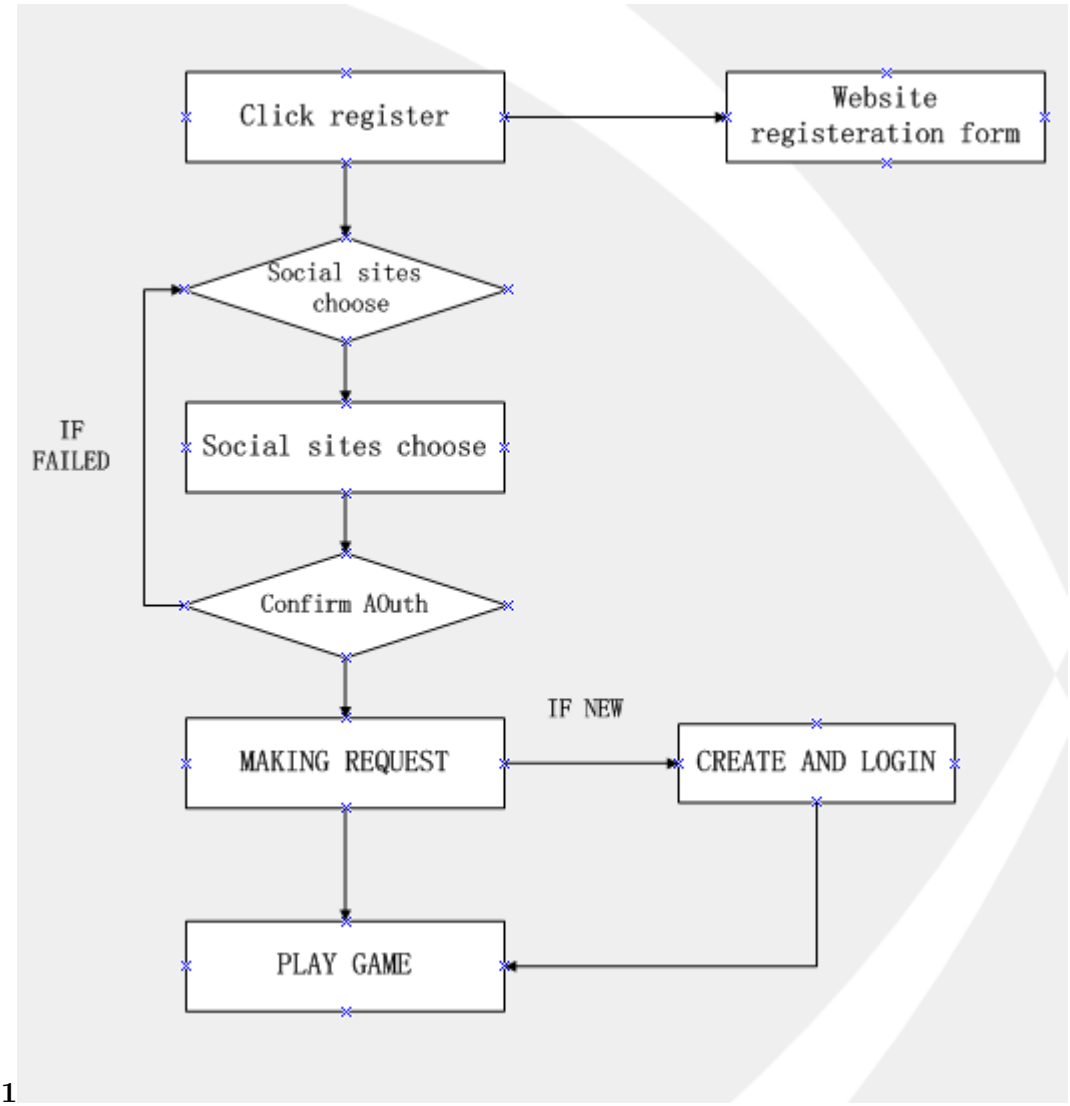


Figure 16:

1	**Id**	int(11)		
2	**orderNum**	varchar(255)	latin1_swedish_ci	
3	**productId**	int(11)		
4	**qty**	int(11)		
5	**gameName**	varchar(255)	latin1_swedish_ci	
6	**serverName**	varchar(255)	latin1_swedish_ci	
7	**serverline**	varchar(255)	latin1_swedish_ci	
8	**userEmail**	varchar(255)	latin1_swedish_ci	
9	**state**	int(10)		UNSIGNED ZEROFILL
10	**update_time**	timestamp		
11	**price**	varchar(255)	latin1_swedish_ci	
12	**payment**	varchar(50)	utf8_general_ci	

10

Figure 17: Figure 10 :



11

Figure 18: Figure 11 :

id	int(8)	UNSIGN
tId	int(8)	
userEMail	varchar(200)	utf8_general_ci
tTime	varchar(200)	utf8_general_ci
fromurl	varchar(500)	utf8_general_ci
game_name	varchar(500)	utf8_general_ci
game_level	varchar(500)	utf8_general_ci
game_paid	varchar(500)	utf8_general_ci
isUse	int(8)	

12

Figure 19: Figure 12 :

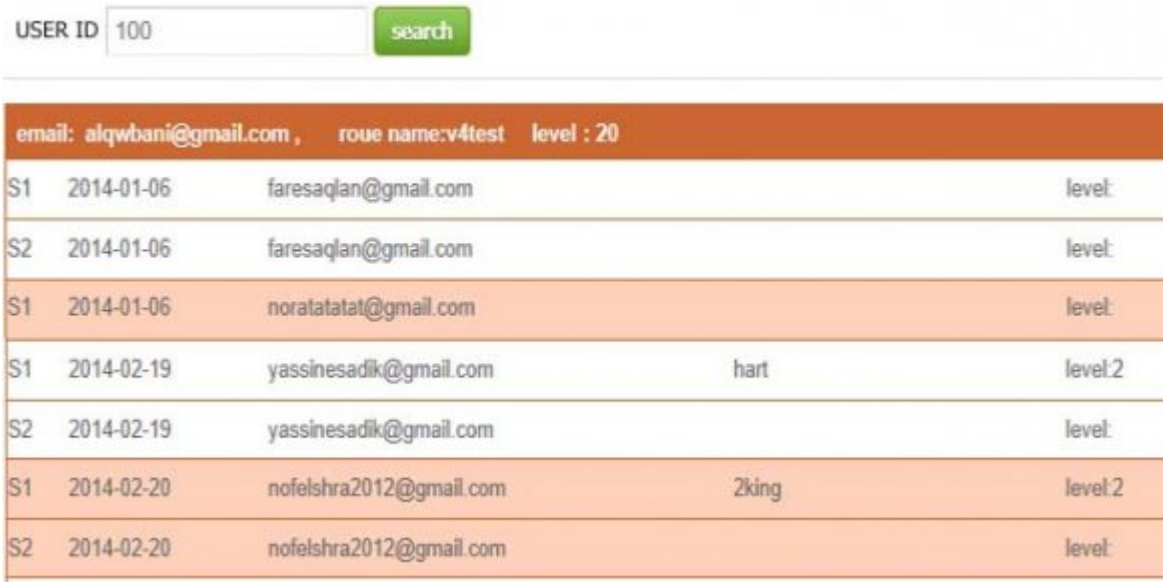
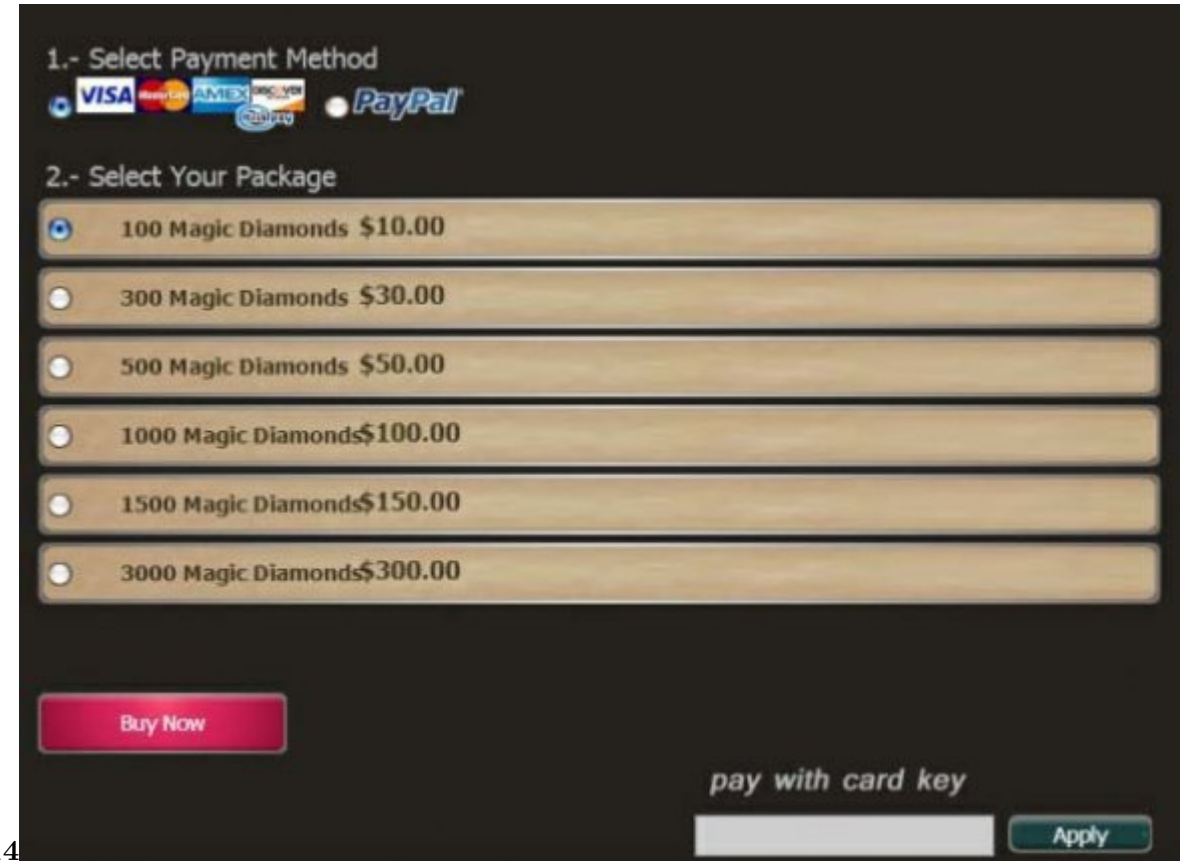


Figure 20:

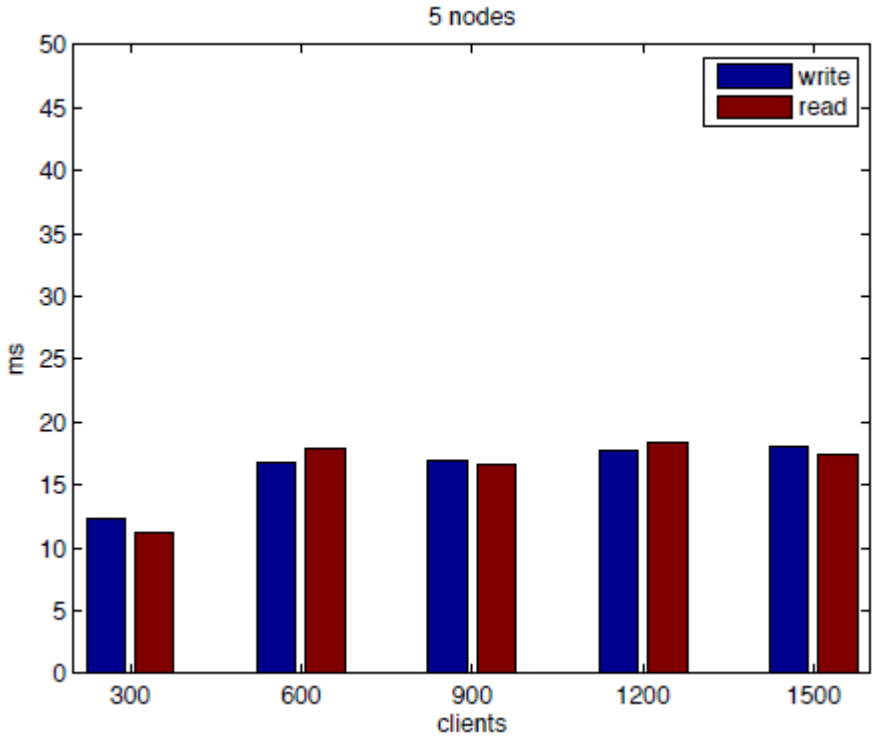
13

Figure 21: Figure 13 :



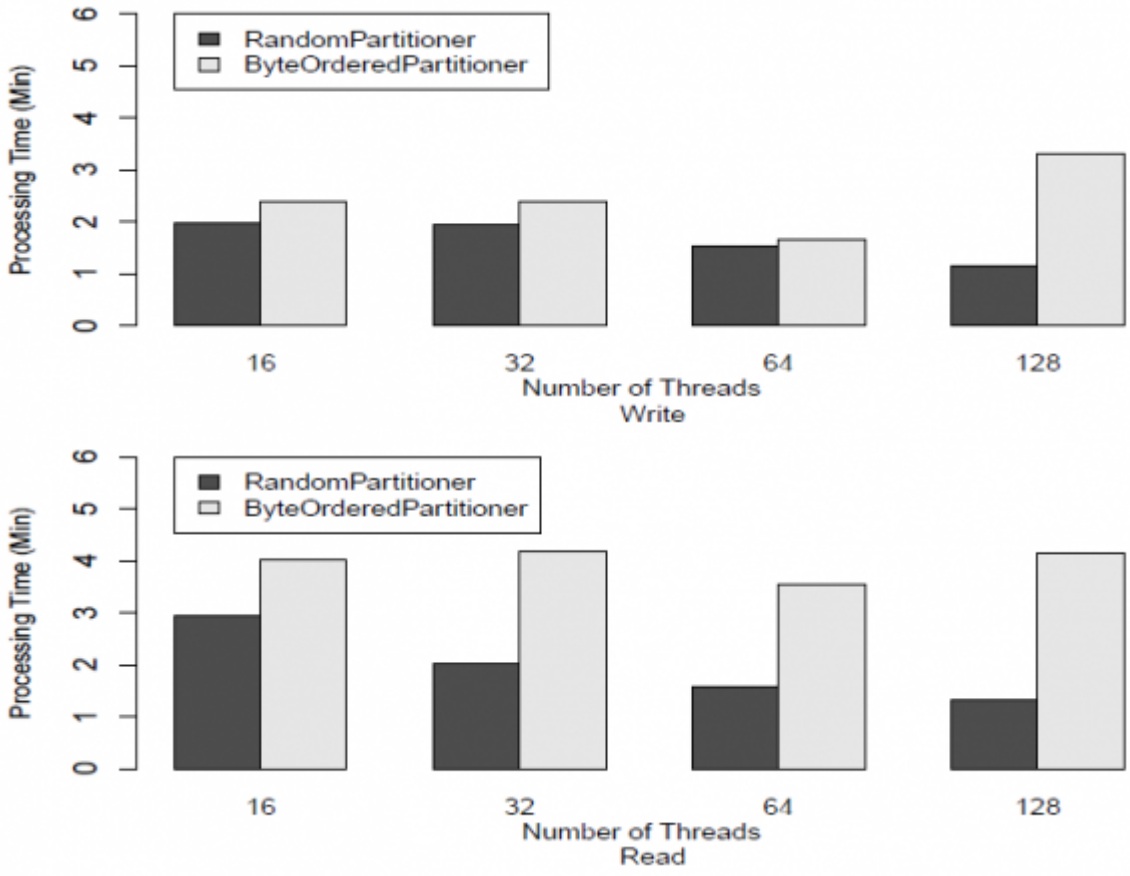
14

Figure 22: Figure 14 :



15

Figure 23: Figure 15 :



16

Figure 24: Figure 16 :

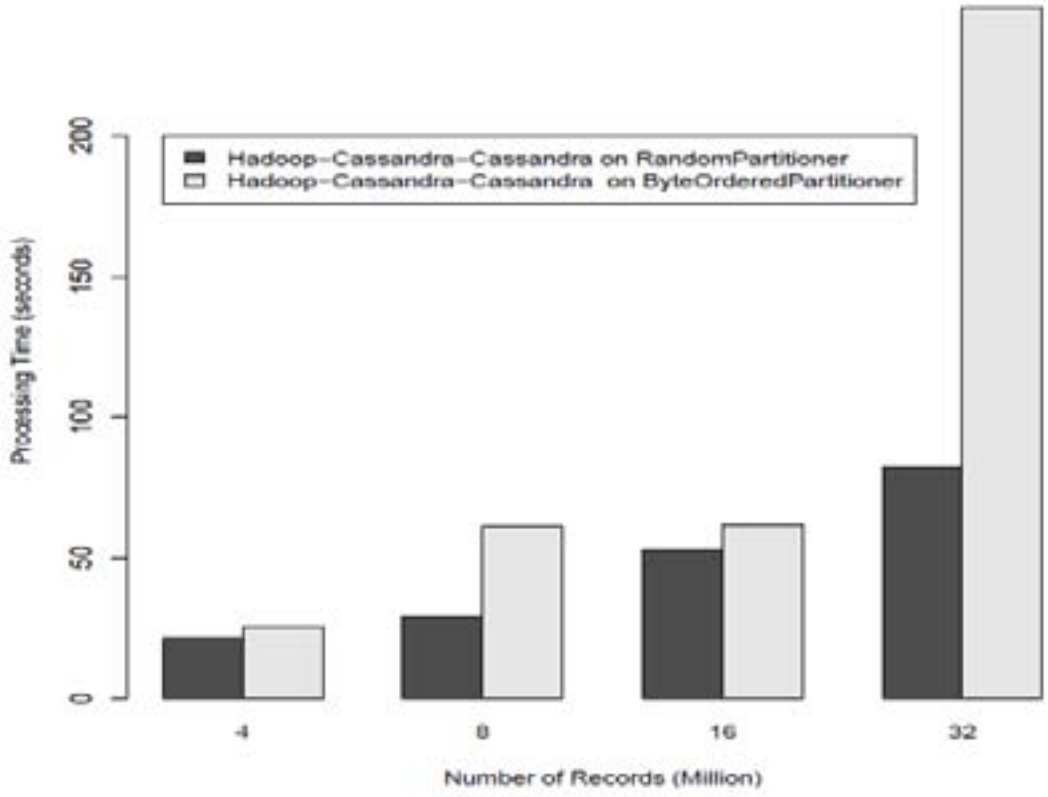
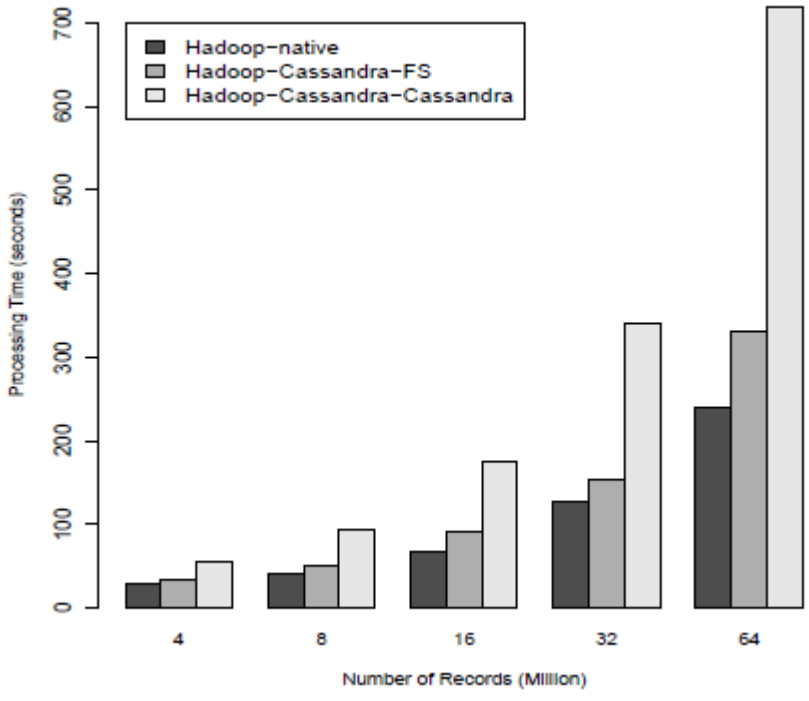


Figure 25:



18

Figure 26: Figure 18 :

[Note: 3. Complexity problems: Data in an RDBMS are stored in different tables using foreign keys to link them. In new and big MMO Games, the state data are so much and more complex, so in this highly concurrent data accessing, RDBMS is getting hard to handle the operations efficiently. 4. Structured problems: with the fixed schema in RDMBS, structure of a table must be pre-defined. And in MMO Games is always need to fix bugs and need to maintain data according these improvement features and game development process which will need to alter existing structure frequently. 5. Cost problems: expensive license from the database vendor for RDBMS cluster.II.]

Figure 27:

.1 Acknowledgements

This project was supported by our School of Information science & Engineering and v4games technique department. Thanks for all support and help and special thanks to school teachers who helped in this project to come to this stage.

[Arash Akhlaghi] , http://incubator.apache.org/thrift/22.Wa'elBelkasim *Arash Akhlaghi*

[Klopfer et al. ()] , Eric Klopfer , Scot Osterweil , Katie Salen . 2009.

[Avinash Lakshman. Cassandra A Decentralized Structured Storage System Operating Systems ()] *Avinash Lakshman. Cassandra A Decentralized Structured Storage System Operating Systems*, Reviewed 2010.

[Karun and Surendran (1956)] 'BigTable, Dynamo & Cassandra -A Review'. Kala Karun , A , Subu Surendran . *International Journal of Electronics and Computer Science Engineering* -2277. 1956 December 2012.

[Bogojevic and Lund (2003)] Sladjan Bogojevic , Mohsen Kazemzadeh Lund . *The Architecture of Massive Multiplayer Online Games*, September 8, 2003.

[Iosup ()] *CAMEO: Continuous Analytics for Massively Multiplayer Online Games on Cloud Resources*, Alexandru Iosup . 2009.

[Brodie et al. (2013)] *Databases and AI: The Twain Just Met, 2011 STI 26. Datastax Apache Cassandra 1.1 Documentation, Hadoop Integration A New Approach to Web Application*, M L Brodie , M Greaves , J A Hendler . http://www.adaptivepath.com/publications/-essays/archives/000385.php 2013. February 18.

[DataStax documentations Apache Cassandra 1.1 Documentation Data modeling] *DataStax documentations Apache Cassandra 1.1 Documentation Data modeling*,

[DataStax Enterprise 3.0 Documentation (2014)] *DataStax Enterprise 3.0 Documentation*, February 21. 2014.

[Nae et al. ()] 'Efficient management of data center resources for massively multiplayer online Games'. V Nae , A Iosup , S Podlipnig , R Prodan , D H J Epema , T Fahringer . *ACM/IEEE Supercomputing IEEE/ACM*, 2008.

[Evaluation and Implementation of Distributed NoSQL Database for MMO Gaming Environment, Exam ensarbete 30 hp Oktobe *Evaluation and Implementation of Distributed NoSQL Database for MMO Gaming Environment, Exam ensarbete 30 hp Oktober*, 2011.

[White (2012)] *Hadoop: The Definitive Guide. O'Reilly Media*, Tom White . 10 May 2012. p. 3.

[Hovard and Berg (2010)] Alexander Hovard , Berg . *the Computer Game Industry*, June 2010.

[Introducing Cassandra Architecture (2013)] *Introducing Cassandra Architecture*, http://www.datastax.com/docs/1.1/cluster_architecture/index/ March 2013.

[B M Moniruzzaman and Syed Akhter Hossain ()] 'NoSQL Database: New Era of Databases for Big data Analytics -Classification, Characteristics and Comparison'. A B M Moniruzzaman , Syed Akhter Hossain . *International Journal of Database Theory and Application* 2013. 6 (4) .

[Dalton (2007)] 'Online gaming architecture: Dealing with the real-time data crunch in MMOs'. B Dalton . *Proc. Austin GDC*, (Austin GDCAustin, TX) September 2007.

[Cattell (2011)] *Scalable SQL and NoSQL Data Stores Originally published in 2010, last revised*, Rick Cattell . December 2011.

[R ()] R . *Scalable SQL and NoSQL data stores SIGMOD Record*, 2010. 39 p. .

[Semantic Summit ()] *Semantic Summit*, (Riga, Latvia) July 6-8, 2011.

[Stonebraker (2010)] 'SQL databases v. NoSQL databases'. Michael Stonebraker . *Commun. ACM* April 2010.

[The Global MMO games market Payments, Intelligence and Trends Global Collect Video Gaming Payments: Knowledge Series E *The Global MMO games market Payments, Intelligence and Trends Global Collect Video Gaming Payments: Knowledge Series Edition*, 2 July 2013.

[Datastax and Benchmarking ()] *Top NoSQL Databases A Performance Comparison for Architects and IT Managers White Paper*, Datastax , Benchmarking . FEBRUARY 2013.

[Why MySQL? Oracle Retrieved (2012)] *Why MySQL? Oracle Retrieved*, 17 September 2012. (Market Share)

[Gamasutra and Com] *World Of War craft Reaches 12 Million Subscribers Worldwide*, Gamasutra , Com . http://www.gamasutra.com/view/news/30845/WorldOfWarcraftReaches12MillionSubscribersWorldwide.php#.UFPdIciZaJ4