CHAPTER - I

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The project is entitled as “HARCHE: Huge data Analysis, Reporting and Caching for applications using the MapReduce framework” which deals with Big data, and is developed using PYTHON as front end, HDFS, Spark SQL as back end. This application is basically concerned with fetching the required data and storing the details using HDFS. The main objective of the project is to reduce the effort of the administrator in managing large amount of data.

The data are stored in key value range. Ratings for each person can be analyzed and improvement can be done for the enterprise using the fetched data. Fetched data are analyzed using HIVE and required data are collected using administrator queries.

Data like age division, National tennis rating program details, Tennis League Stats Ratings, National tennis rating program effective level, Scrapy run time, facility, url, section, area, sex and player hash are analyzed and report is generated. The queries given by the administrator are used to retrieve stored data from HDFS, the data is scraped using spider according to the queries. The data are encrypted using Data Encryption Standard. From the scraped data, filtering is done using MapReduce and stored in HDFS. The filtered data is analyzed and report is generated in the desired format.

1.2 NEED OF THE PROJECT

The need of the project is to atomize manual calculations and facilitate the administrator to analyze large amount of data. Since manual processes consume more time for data fetching and report generation. This project is used to reduce manual entries of a particular data and atomization helps to increase efficiency in time. The atomization also helps to increase the efficient usage of data and avoid duplications and high reliability is achieved in transactions.

The data are maintained and analyzed efficiently than doing it in excel sheet. The data analytic report is generated automatically based on the administrator queries. The report can be printed in any format. This efficiency is not provided in manual system. It would take lot of time to provide analysis report in excel sheet.

1.3 OBJECTIVE OF THE PROJECT

The main objective of the project is that to maintain the data and find faster retrievals of search queries. Records can be easily retrieved from the database. It provides valuable information about the users and also provides information about the most accessible data in the website.

Records will be maintained for all data. Records can be updated and easily retrieved from the database. Records are protected with high security and the access right is given only to the administrator. The data is fetched in binary bytes and reserialized after scraping. Data analytics is done on patient details.

Large data sets are maintained using clustering. Similar data are clustered into a single group named data node, name node and secondary name node. Grouped data are analyzed and report is generated using HIVE.

Related clusters get stored in a node, process is done on that done. Queries are executed accordingly to fetch items from theses nodes dynamically.

CHAPTER - II

2. SYSTEM ANALYSIS

2.1 BACKGROUND STUDY

System analysis is concerned with the comparison study about the existing system and the proposed system. The system analysis is essential when the software interfaces with other elements such as software, people and other resources. The essential purpose of this place is to find the need and to define the problem that needs to be solved.

2.1.1 EXISTING SYSTEM

In the existing system the records are maintained in excel sheet. Very essential information is stored. Considering the coverage of large data storage area. It becomes a tedious task to maintain the data based on three V's -volume, velocity and variety. In existing system, the data is analyzed in the server and stored manually in excel.

Drawbacks of Existing System

- Time consuming
- Lack of security.
- slow retrieval of data
- Not designed for collaborative work
- Scales poorly

2.1.2 PROPOSED SYSTEM

The project helps to reduce man power and also consumes less work space. The proposed system is very efficient and highly reliable. Data repetition and inconsistency is reduced. The data is scraped dynamically from the application and it is stored in HDFS. The data is scraped as list of items and the list id appended into a dictionary. Large amount of data is stored, analyzed and report is generated for it. Efficiency by distributing data and logic to process it in parallel on nodes where data is located. Reliability by automatically maintaining multiple copies of data and automatically redeploying processing logic in the event of failures

Advantages of Proposed System

- Faster and efficient in processing of information
- Automation of the processes
- Data Access is very efficient
- High reliability
- High security is provided
- Up-to-date information is provided
- Useful reports can be generated for management to make decisions

2.2 SYSTEM SPECIFICATION

2.2.1 HARDWARE SPECIFICATION

The simple hardware requirements that are required to process the Hadoop based system in their system is as follows:

- Standard PC.
- Standard Server.

2.2.2 SOFTWARE SPECIFICATION

Front-End	: Python, Java
Back-End	: HDFS
Query Language	: Spark SQL
Operating System	: LINUX-Ubuntu
Browser	: Mozilla Firefox (32 and above)
Framework	: Hadoop (2.3 and above)

PYTHON

In this project PYTHON is used for the development. PYTHON is one of the most popular server side scripting languages running today. It is used for creating dynamic web pages that interact with the user offering customized information. PYTHON code is interpreted by a web server with a processor module which generates the resulting web page. PYTHON commands can be embedded directly into a python source document rather than calling an external file to process data.

PYTHON codes are executed on the server, and the result is returned to the browser. PYTHON files have a default extension of ".py".

Why PYTHON

- PYTHON runs on different platforms (Windows, Linux, Unix, etc.)
- PYTHON is compatible with almost all servers used today.
- PYTHON has support for wide range of databases.
- PYTHON is easy to learn and runs efficiently on the server side.
- A popular choice in today's web world is using PYTHON. PYTHON is a general-purpose scripting language that is especially suited to server-side web development where PYTHON generally runs on a web server.
- Stability, flexibility and speed are the chief qualities that attract business owners to choose PYTHON.
- Although PYTHON is already well established, its future prospects are infinite. The keynote is that PYTHON is loosely typed. This makes simple scripts much faster to develop. One has to devote much less energy towards design.
- PYTHON is an interpreter, object-oriented, high level scripting and programming language.

Benefits

- Open source, Simple and very easy to learn
- Programmers of java, PERL, BASIC, and other popular languages can find many parallels to ease transition to PYTHON
- It runs on different operating system
- It's quick to develop in PYTHON

SPARK SQL

Spark SQL allows relational queries expressed in SQL, HiveQL, or Scala to be executed using Spark. At the core of this component is a new type of RDD, SchemaRDD. SchemaRDD are composed of Row objects, along with a schema that describes the data types of each column in the row. A SchemaRDD is similar to a table in a traditional relational database. A SchemaRDD can be created from an existing RDD, a Parquet file, a JSON dataset, or by running HiveQL against data stored in Apache Hive.

Spark SQL supports operating on a variety of data sources through the SchemaRDD interface. A SchemaRDD can be operated on as normal RDDs and can also be registered as a temporary table. Registering a SchemaRDD as a table allows you to run SQL queries over its data. This section describes the various methods for loading data into a SchemaRDD.

Benefits

- ✓ Compact binary in-memory data representation, leading to lower memory usage and reduced reliance on the JVM garbage collector
- ✓ Code generation to convert expressions in SQL to Java byte code
- ✓ Predicate pushdown to reduce I/O
- ✓ Optimal pipelining of operations
- ✓ Optimal join strategy selection (sort-merge, hash, broadcast join, etc.)

JAVA

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA) meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2015, Java is one of the most popular programming languages in use particularly for client-server web applications, with a reported 9 million developers Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Benefits

- ✓ Java is easy to learn.
- ✓ Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
- ✓ Java is object-oriented.
- ✓ This allows you to create modular programs and reusable code.
- ✓ Java is platform-independent.

HDFS

HDFS is an Apache Software Foundation project and a subproject of the Apache Hadoop project. Hadoop is ideal for storing large amounts of data, like terabytes and petabytes, and uses HDFS as its storage system. HDFS lets to connect nodes (commodity personal computers) contained within clusters over which data files are distributed. It can then access and store the data files as one seamless file system. Access to data files is handled in a streaming manner, meaning that applications or commands are executed directly using the MapReduce processing model.

HDFS is fault tolerant and provides high-throughput access to large data sets. HDFS has many goals. Here are some of the most notable: Fault tolerance by detecting faults and applying quick, automatic recovery. Data access via MapReduce streaming Simple and robust coherency model. Processing logic close to the data, rather than the data close to the processing logic. Portability across heterogeneous commodity hardware and operating systems. Scalability to reliably store and process large amounts of data. Economy by distributing data and processing across clusters of commodity personal computers.

Benefits

- ✓ Fault tolerant
- ✓ Runnable on commodity hardware
- ✓ Provides streaming-data access
- ✓ GB- and TB-sized files
- ✓ PB-sized clusters

- ✓ Distributed, scalable, portable
- ✓ Gives control over the number of replicas
- ✓ 3 is a common replication factor

APACHE HADOOP

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework.

The base Apache Hadoop framework is composed of the following modules:

- Hadoop Common – contains libraries and utilities needed by other Hadoop modules;
- Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;
- Hadoop YARN – a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications
- Hadoop MapReduce – an implementation of the MapReduce programming model for large scale data processing.

Benefits

- ✓ Scalable
- ✓ Flexible
- ✓ Cost effective
- ✓ Resilient to failure (Fault Tolerance)
- ✓ Fast

CHAPTER - III

3. DESIGN AND DEVELOPMENT PROCESS

3.1 FUNDAMENTAL DESIGN CONCEPTS

A software design is a meaningful engineering representation of some software product that is to be built. A design can be traced to the customer's requirements and can be assessed for quality against predefined criteria. During the design process the software requirements model is transformed into design models that describe the details of the data structures, System architecture, interface, and components. Each design product is reviewed for quality before moving to the next phase of software development.

3.2 DESIGN NOTATIONS

Design is defining a model of the new system and continues by converting this model to a new system. The method is used to convert the model of the proposed system into computer specification. Data models are converted to a database and processes and flows to user procedures and computer programs. Design proposes the new system that meets these requirements.

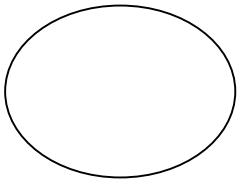
The new system may be built by changing the existing system. The detailed design starts with three activities, database design, user design, and program design. The database design uses conceptual data model to produce a database design. User procedure design uses those parts of the architecture outside the automation boundary to design user procedures.

3.2.1 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identify how the information flows through the processed. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs.

Data flow diagrams are made up of number of symbols.

Squares representing external entities, which are sources or destination of data.

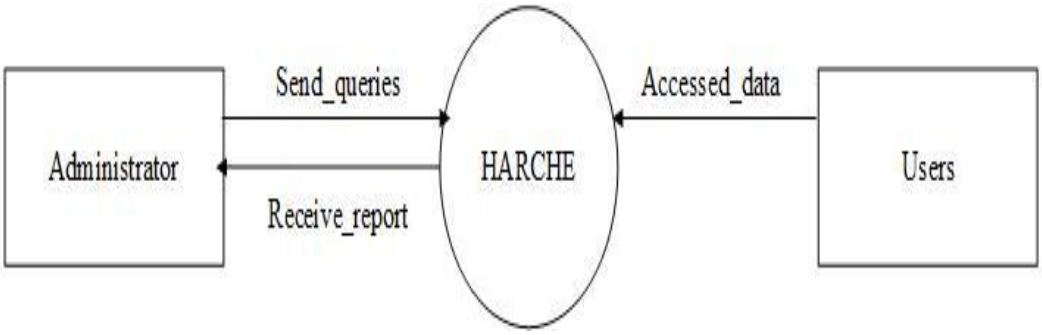


Circle representing processes, which take data as input, do something to it, and output it.

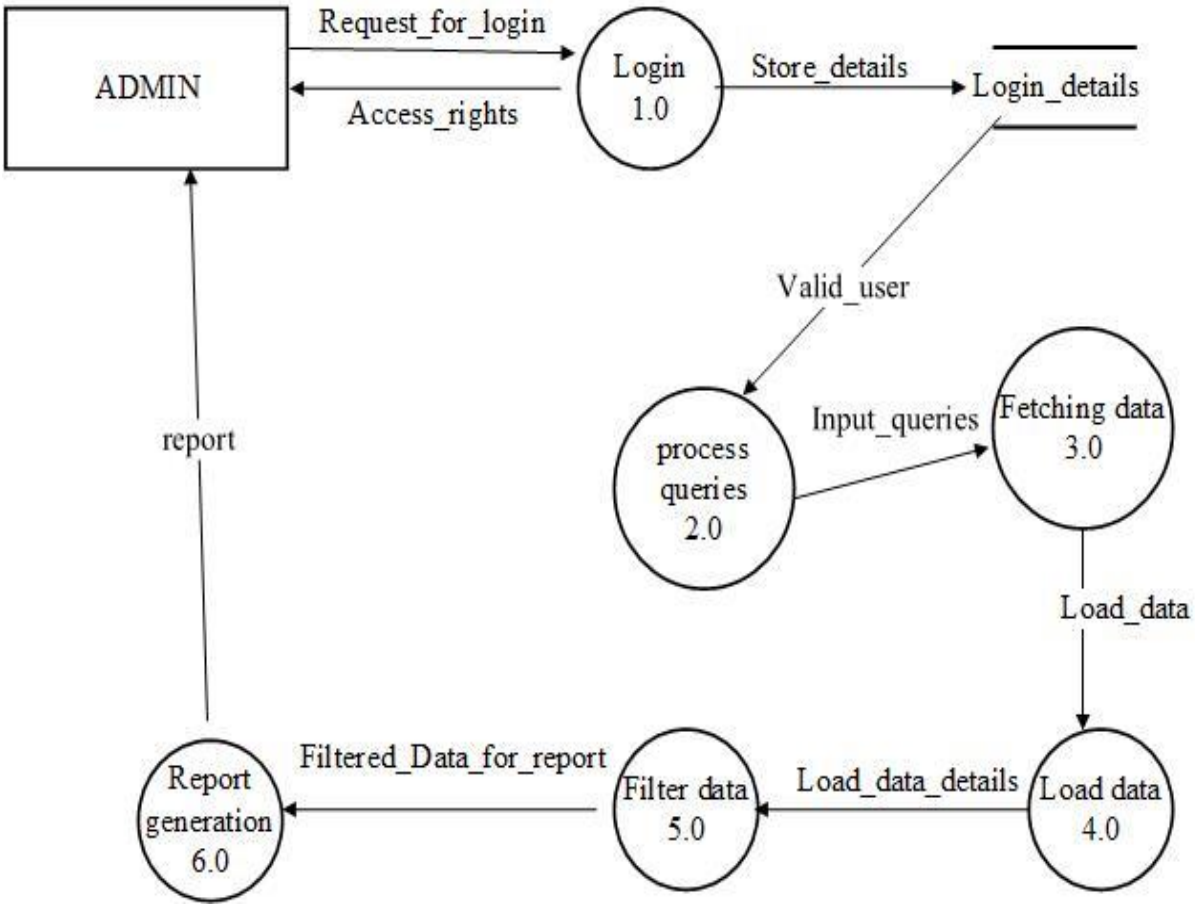
Arrows representing the data flows, which can either, be electronic data or physical terms.

Parallel lines representing data stores, including electronic stores such as databases or XML files and physical stores

CONTEXT LEVEL DIAGRAM



LEVEL 0 DIAGRAM



3.3 DESIGN PROCESS

System design is a detailed concentration of the technical and other specifications that will make the new system. Design goes through logical and physical stages of the development. It is a process or art of defining the hardware and software architecture components and data for a computer and data for a computer system to satisfy specified requirement.

During this phase first, the architectural design of the system was prepared the system was prepared. The different components of the software are individual and entire system is decomposed into processing module and conceptual data structures.

The interconnections among that data structures were also specified. The specific function of the computer component of the overall system was defined and design requirements for the external system inputs were also established.

3.3.1 MODULE DESIGN

PROJECT MODULES AND ITS DESCRIPTION

MODULES:

1. Login
2. Scrapping
3. Query editor
4. Charts
5. Hosts
6. Audit
7. Admin

MODULE DESCRIPTION

1. **Login:**

This module is used to check authentication for administrator by validating their username and password. This module allow administrator to login to HARCHE.

2. **Scrapping:**

This module scraps large amount of data from administrator authenticated website automatically. This scraped data are used for further analysis.

3. **Query editor:**

This module is used to get Query from data analyst and execute those queries using SPARKQL interpreters. The interpreters process using java and return the result for the requested query

4. **Charts:**

The processed queries are displayed in chart format like line, stack area, bar, scatter, Heatmap, histogram, table. History of queries processed can be viewed using recent tab. The result can be visualized using various chart types.

5. **Hosts:**

This module is used to analyze the node storage space, physical memory, load data details, swap space, latest usage. These details are used to check the load for the nodes.

6. **Audit:**

This module is used to analyze and keep track of all recent queries executed using HARCHE. New filter can be created and applied for individual host names and users. Filter can be applied using queries.

7. **Admin:**

This module creates authentication for other users. Administrator subdivides process and provide authentication for one or more users. New workspace can be created with binding to SPARKQL interpreter. Alerts are set for certain limits like displaying of number of fields for all users.

3.3.2 DATABASE & TABLE DESIGN

Designing the database is part of the system. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system. Normalization is done to get an internal consistency of data and to have minimum stability.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of database design is to make the data access easy inexpensive and flexible to the user. Relationships are established between the data items and unnecessary data items are removed.

LOGIN TABLE

Primary Key: id

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
Id	Int	Primary Key	User id number
Username	varchar	NOT NULL	User name
Password	varchar	NOT NULL	Password for username
User_type	varchar	NOT NULL	User type of the user
User_role	varchar	NOT NULL	Role of the user

3.3.3 INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer based format, input data are collected and organized into a group of similar data. Inaccurate input data is the most common cause of data processing errors. Effective input design minimizes errors made by data entry operators. The goal of designing input data is to make data entry as easy, logical and free from errors as possible. In addition to the general form considerations such as collecting only required data, grouping similar or related data, input design requires consideration of the needs of the data entry operator.

Login details

Login form gets username and password from the administrator and validates the information. If valid the user is redirected to the next page. If invalid it asks for appropriate username and password.

Query Editor

Query editor form gets query from the administrator and process it using JAVA, returns java exception if query throws error. The query results are displayed in charts type. The chart types are user defined.

Workspace builder

Workspace builder form is used to create new workspace with specified interpreter for HIVE and SPARKQL.

3.3.4 OUTPUT DESIGN

Output is the important and direct source of information to the user. Efficient, intelligible output design should improve the system's relationships with the user and helps in decision making. The output design is designed in such a way that is viewed and understood by the users easily. It is done in a colorful way by CSS and various options like select options and hyperlinks are used

Login Information

Login form gets username and password from the administrator and validates the information. If valid the user is redirected to the next page. If invalid it asks for appropriate username and password.

Host information

This form contains information about swap space, physical memory, details about latest usage. This provides host allocated for each node like name node, data node, secondary name node.

Administrator profile

This form displays administrator details like their username, password, authentication rights, access role. Administrator can also add sub user to him and he can provide authentication rights to them.

Chart report

This form displays result of the inputted query in chart format. Chart type, description, dimension is user defined.

Audits

This module is used to analyze and keep track of all recent queries executed using HARCHE. New filter can be created and applied for individual host names and users. Filter can be applied using queries.

CHAPTER - IV

4. TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

For any software that is newly developed, primary importance is given to testing of the system. It is last opportunity for the developer to detect the possible errors in the software before handing over it to the customer. Testing is the process by which gives the maximum probability of finding all types of errors that can occur in the software. The various steps of testing the system can be listed as given below.

4.1.1 TESTING METHODOLOGIES

The entire testing process can be divided into phases like

Unit Testing

The first level of testing is called as unit testing. Here login, scraping and query editor modules are tested separately for their correct functionality in this testing step, each module was found to be working satisfactory per the expected output of the module.

Validation testing

The login form had been tested for validation to avoid invalid login to the system. For example, if the user presses the submit button before entering the data into the field a message will be displayed as invalid login. Login form is only valid if correct username and password is entered. If correct then the administrator is directed into the modules form else displayed as invalid login.

Integration Testing

The entire project was split into small programs; each of these programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructions were used. It gives a lot of problem by not functioning in an integrated manner.

The login form and home form is tested invidually. The login is validated with its username and password. If these small programs are valid then the whole form is tested.

Output Testing

After performing the validation testing, next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in to two ways. One is on screen and another is printed format. The output comes as the specified requirements by the user. Hence output testing does not result in any correction in the system.

White Box Testing

This test is conducted during the code generation phase itself. All the errors were rectified at the moment of its discovery. During this testing, it is ensured that

- All independent paths within a module have been exercised at least once
- Exercise all logical decisions on their true or false side.
- Execute all loops at their boundaries.

Black Box Testing

It is focused on the functional requirements of the software. It is not an alternative to White Box Testing; rather, it is a complementary approach that is likely to uncover a different class of errors than White Box methods. It is attempted to find errors in the following categories.

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Performance errors and
- Initialization errors.

TEST CASE

LOGIN FORM

Test Case	**Test Description**	**Test steps**	**Expected output**	**Actual output**	**Status**
1)Login Form	To test the validity	1)Load the Login Form 2)Enter the correct username 3)Enter the correct password 4)Submit	Valid username & password	Valid username & password	Username & password is Correct (Valid). The result is OK.
2)Login Form	To test the validity	1)Load the Login Form 2)Enter incorrect username 3)Enter incorrect password 4)Submit	Invalid username & password	Invalid username & password	Username & password is not Correct (Invalid). The result is OK.

4.2 SYSTEM IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main work load and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause chaos and confusion.

Implementation includes all those activities that take place to convert the old system into new one. The new system may be totally new, replacing an existing manual or automated system or it may be major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organization requirements.

The implementation stage involves following tasks.

Careful planning

The planning of the system implementation has to be done cautiously, since implementation is a critical process. Careful planning is done before implementing the system.

Investigation of system and constraints

The system in which the project is to be implemented has to be analyzed thoroughly. The existing system is analyzed and checked if changes is needed to implement the proposed system.

Design if methods to achieve the changeover

The system can start for designing if planning and constraint investigation are successful.

Training of the staff in the changeover phase

The training is given to the Data analyst, who uses the project for the organization.

Evaluation of the changeover method

After the training, the feed backs are collected from them, evaluated and used for further enhancement.

4.3 SYSTEM MAINTENANCE

After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affects a system, which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only through proper system maintenance procedures, the system can be adapted to cope up with these changes.

Software maintenance activities can be classified into

- Corrective maintenance.
- Adaptive maintenance.
- Perceptive maintenance.
- Preventive maintenance.

Corrective maintenance

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called corrective maintenance.

Perceptive maintenance

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore, adaptive maintenance- an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

Adaptive maintenance

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancements are received from users. To satisfy requests in this category, perfective maintenance is performed. This activity accounts for the majority of all effort extended on software maintenance.

CHAPTER - V

5. CONCLUSION

The "**HARCHE**: Huge data Analysis, Reporting and Caching for applications using the MapReduce framework**"** is done to make the process easily and efficiently. This particular project is a solution developed to comfort the manual process that is automated.

The system main objective is to minimize manual power for the organization. The particular project is a solution developed to comfort the enterprise or organization. In excel processing system if any flaw occurs in one phase it may reflect till the end of the whole process and large data cannot be maintained using SQL or other server. This system avoids the chaos and provides the administrator to easily retrieve the data from the administrator website and store in any format. It also provides administrator to generate report easily.

Data analysis is done on age division, National tennis rating program details, Tennis League Stats Ratings, National tennis rating program effective level, Scrapy run time, facility, url, section, area, sex and player hash.

The system developed manages large amount of data and retrieval is done easily. The forms are very user friendly and also to handle even by the beginners with very little effort and guidance.

CHAPTER – VI

6. SCOPE FOR FURTHER ENHANCEMENT

The project title "**HARCHE:** Huge data Analysis, Reporting and Caching for applications using the MapReduce framework" is developed successfully from the perspective of all the modules. Only administrator has the access rights to login or logout the system, adding or modifying any other users. In future, it can be enhanced by using bio metrics recognition pattern. The project can also be extended by adding new modules as per the user requirements.

The system has some flaws like it is not portable for the administrator. In future the system may be enhanced as a simple app in android or IOS to manage large data efficiently.

Some of the enhancements that can be done to the system

- A module from where entities can communicate
- Future prediction of data using past and present data

BIBLIOGRAPHY

TEXT BOOKS

- David M.Beazly, **"Python essential reference",** Addison-Wesley professional, Fourth Edition in 2009
- Wes McKinney, **"Python for data analysis",** O'Reilly Media.Inc. Fourth Edition in 2011.
- Tom White **"Hadoop: The Definitive Guide"**, Manning Publications, fifth edition in 2012
- Chuck Lam **"Hadoop in Action"** O'Reilly Press publications, sixth edition 2014
- Dean Wampler **"Programming Hive"** O'Reilly Press publications**,** seventh Edition 2013

WEBSITES

- http://www.lovepython.blogspot.in/2010/09/python-code-to-retreive-links
- http://hortonworks.com/hadoop-tutorial/hello-world-an-introduction-to-hadoop-hcatalog-hive-and-pig/
- http://www.coreservlets.com/hadoop-tutorial/
- https://cwiki.apache.org/confluence/display/Hive/Tutorial
- https://www.youtube.com/watch?v=0r8ut2jF16o
- https://pig.apache.org/docs/r0.7.0/tutorial.html
- https://pig.apache.org/docs/r0.11.1/start.html
- http://hortonworks.com/hadoop-tutorial/hands-on-tour-of-apache-spark-in-5-minutes/

D. ABBREVIATIONS

HARCHE	-	Huge Data Analysis, Reporting and Caching for applications using the MapReduce framework
HiveQL	-	Hive Query language
SQL	-	Structured query language
JSP	-	Java Servlet page
ORC	-	Optimized Row Columnar
CSS	-	Cascading Style Sheet
ORM	-	Object Relational Mapping
HDFS	-	Hadoop Distributed File System
PY	-	Python
UTF	-	Unicode Transformation Format
BWT	-	Burrows–Wheeler transform
UDF	-	Universal Disk Format