DISTRIBUTED DATA ACCESS CONTROL USING HADOOP

P. Kanaha* & R. Selvakumar**

* PG Scholar, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

** Assistant Professor, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:

Hadoop is an unlock-source software infrastructure for storing data and running applications on clusters of commodity hardware. It provides big storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks. Big data means really a big data; it is a collection of large datasets that cannot be processed using traditional computing techniques. Big data is not merely a data; rather it has become a complete subject, which involves various tools, techniques and frameworks. The danger, of course, in running on commodity machines is how to handle failure. Hadoop is architected with the expectation that hardware will fail and as such, it can delightfully handle lot of failures. Furthermore, its architecture allows it to scale nearly linearly, so as processing capacity demands increase, the only constraint is the amount of budget you have to add more machines to a cluster. At a high-level, hadoop operates on the philosophy of pushing analysis code close to the data it is aim to analyze rather than requiring code to read data across a network. Big data involves the data produced by different devices and applications. Given below are some of the fields that come under the umbrella of Big Data

Index Terms: Big Data Map Reduce, Hadoop & Cloud Computing

1. Introduction:

Big data is not merely a data; rather it has become a complete subject, which involves various tools, techniques and frameworks. The danger, of course, in running on commodity machines is how to handle failure. Hadoop is architected with the expectation that hardware will fail and as such, it can delightfully handle lot of failures. Furthermore, its architecture allows it to scale nearly linearly, so as processing capacity demands increase, the only constraint is the amount of budget you have to add more machines to a cluster. At a high-level, hadoop operates on the philosophy of pushing analysis code close to the data it is aim to analyze rather than requiring code to read data across a network. Big data involves the data produced by different devices and applications. Given below are some of the fields that come under the umbrella of Big Data.

Black Box Data: It is a component of helicopter, airplanes, and jets, etc. It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft.

Social Media Data: Social media such as Face book and Twitter hold information and the views posted by millions of people across the globe.

Stock Exchange Data: The stock exchange data holds information about the 'buy' and 'sell' decisions made on a share of different companies made by the customers.

Power Grid Data: The power grid data holds information consumed by a particular node with respect to a base station.

Transport Data: Transport data includes model, capacity, distance and availability of a vehicle.

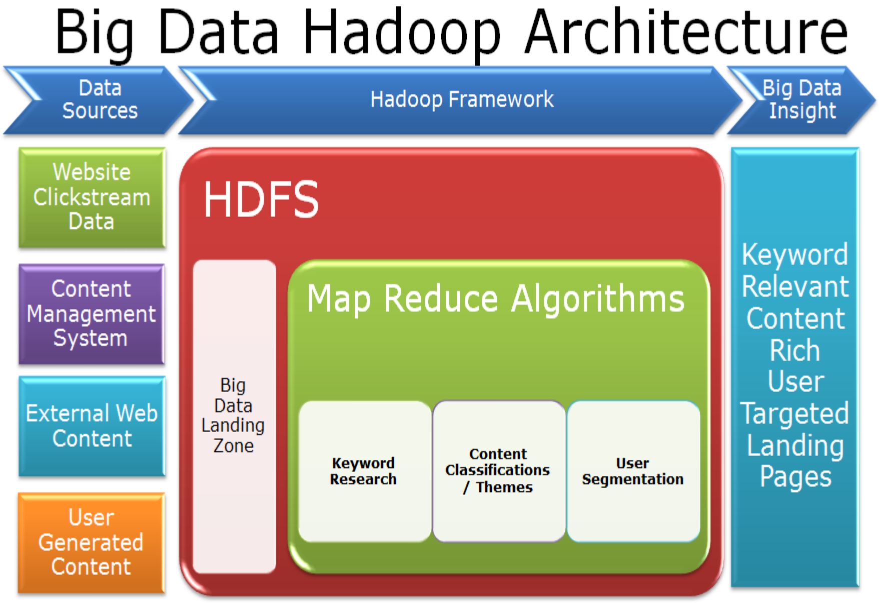
Search Engine Data: Search engines retrieve lots of data from different databases.

Thus Big Data includes huge volume, high velocity, and extensible variety of data. The data in it will be of three types.

Structured Data: Relational data.

Semi Structured Data: XML data.

Unstructured data: Word, PDF, Text, Media Logs.



Online users search for products, services, topics of interest etc. not only in Google and other search engines, but also more importantly on site itself (For example, in eCommerce site Amazon.com, search is the top product finding method used by site visitors). Facilitating searchers by providing relevant search results is something online search providers like Google, Bing and also site search providers continuously optimize and calibrate.

From an Online Marketing perspective, once the searchers click through the search results and arrive at the website (if coming through external search like Google) or arrive at the product or topic page they were searching internally on the site, that page of arrival from a search result, called as landing page in Online Marketing terminology, is very important for: Improving Conversion Rate (%) of the site. Traffic dispersion to subsequent stages of the site. Improving site engagement for the users large websites generate and also need to process, huge volumes of different varieties of data as below: Website click stream data collected through Web Analytics applications like Omniture and from web server logs. The website content such as product content, marketing content, navigation etc. in various formats like text, images, videos etc. which is available in the web content management systems. External web content typically collected by web crawlers, which includes content such as Product content from competitor websites Marketing collaterals from external industry websites etc. User generated content such as product reviews, user survey feedback, social media posts, online discussions, tweets, blog posts, online comments, Wiki articles etc.

Most of the above varieties of data are unstructured or semi-structured, and hence cannot be collected and processed in traditional RDBMS databases like Oracle or MySQL. For large websites, it is not just important to collect large volumes of variety of data as shown above, but it is also important to handle the velocity at which all these data is getting generated online, particularly click stream data and user generated content.

This is where Big Data Analytics solutions come in. In this above example, a typical Architecture to support Big Data Analytics is solution using open source Apache Hadoop framework. In Hadoop architecture - big volumes, variety and velocity of online data are collected and then stored in HDFS file system. Hadoop architecture also provides RDBMS like databases such as HBase, for storing big data in traditional style, particularly useful for beginners and new users of these Big Data Architectures. As we can see in this example, a big data landing zone is set up on a Hadoop cluster to collect big data, which is then stored in HDFS file system.

Using Map-Reduce programming method, Online Marketing Analysts or Big Data Scientists or Analysts develop and deploy various algorithms on a Hadoop cluster for performing Big Data Analytics. These algorithms can be implemented in standard Core Java programming language which is the core programming language used for executing various services for collecting, storing and analyses of big data in Hadoop architecture. Additional programming languages like Pig, Hive, Python or R can be used to implement the same algorithms with less number of lines of code to be deployed. However code written in any of these additional languages would still is compiled into Core Java code by Java Compilers for execution on Big Data Hadoop Architectures.

Some of the use cases of Online Marketing Algorithms which can be implemented on Hadoop Architecture for deriving Analytics are shown in the same example. All these algorithms are deployed using the Map-Reduce programming method.

Keyword Research: Counting the number of occurrences in content and search for hundreds of thousands of keywords across the diverse variety of data collected into Hadoop and stored in HDFS. This algorithm would help identify top keywords by volume, and also the long tail of hundreds of thousands of keywords searched by users. Even new hidden gems among keywords can be discovered using this algorithm to deploy in SEM/SEO campaigns.

Content Classifications / Themes: Classify the user generate content and also web content into specific themes. Due to huge processing capabilities of Hadoop Architecture, huge volumes of content can be processed and classified into dozens of major themes and hundreds of sub themes.

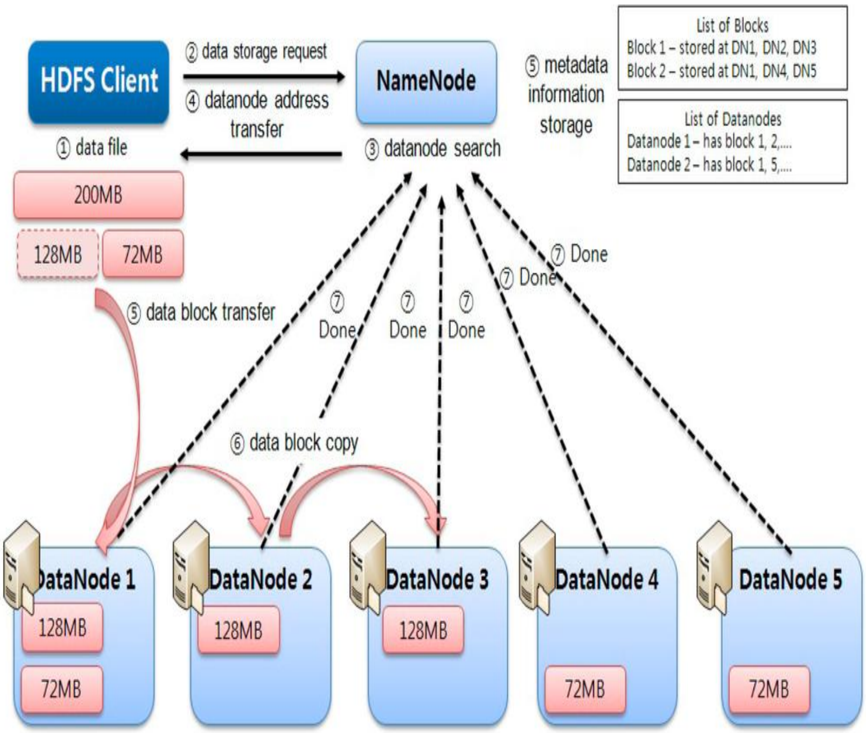
User Segmentation: Individual user behavior available in web click stream data is combined with online user generated content and further combined with user targeted content available in web content management systems to generate dozens of user segments, both major & minor segments. Further this algorithm would identify the top keywords and right content themes targeted for each of the dozens of user segments, by combining the output from other algorithms used for Keyword Research and Content Classifications.

2. Related Works:

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with undefined (hidden) states. A hidden Markov model can consists a common of a collaborative model where the latent variables, which control the combine component to be selected for each observation, are related through a Markov process rather than independent of each other. Currently, hidden Markov models have been generalized to pair wise Markov models and triplet Markov models which allow consideration of more complex data structures and the modeling of non-stationary data. Attribute to the revolutionary development of web 2.0 technology, individual users have become major contributors of web content in online social media. In light of the growing activities, how to measure a user's influence to other users in online social media becomes increasingly important.

The user influence is computed by combining content-based and network-based approaches Social influence occurs when one's opinions, emotions, or behaviors are affected by others, intentionally or unintentionally. Map Reduce is the heart of hadoop. It is a programming model designed for processing large volumes of data in parallel by dividing the work into a set of independent tasks. The framework possesses the feature of data locality. Data locality means movement of algorithm to the data instead of data to algorithm. When the processing is done on the data algorithm is moved across the Data Nodes rather than data to the algorithm. The architecture is so constructed because Moving Computation is Cheaper than Moving Data. It is fault tolerant which is achieved by its daemons using the concept of replication. The daemons associated with the Map Reduce phase are job-tracker and task-trackers. Informational social influence: to accept information.

System Architecture:



Normative social influence: to conform to the positive expectations of others. Weighted in-degree measure does not take into consideration the global structure of the network. It has following demerits are,

- ✓ These algorithms utilize more memory.
- ✓ Computational complexity is high.
- ✓ Execution time of this algorithm is high.
- ✓ Accuracy and efficiency is less.
- ✓ Ranking accuracy is less.

The upcoming data deluge of semantic data, the fast growth of ontology bases has brought significant challenges in performing efficient and scalable reasoning. Traditional centralized reasoning methods are not sufficient to process large ontologism. Distributed reasoning methods are thus required to improve the scalability and performance of inferences. Proposes an incremental and distributed inference method for large-scale ontologism by using Map Reduce, which realizes high-performance reasoning and runtime searching, especially for incremental knowledge base. By constructing transfer inference forest and effective assertion triples, the storage is largely reduced and the reasoning process is simplified and accelerated. Finally, a prototype system is implemented on a Hadoop framework and the experimental results validate the usability and effectiveness of the proposed approach.

- These algorithms utilize more memory.
- Computational complexity is high.
- Execution time of this algorithm is high.
- Accuracy and efficiency is less.
- Ranking accuracy is less.

3. Proposed Work:

Internet has become a media of communication. Web 2.0 increases its' impact and make it as social media where people can share all the information. Using this people is forming a group, forum for discussion etc. This discussion group becomes a vital source of information. This forum discussion is available for all fields like medical, computer science, engineering and technology. Since millions and millions of people is using net as forum information evolved over a net is huge and decision making based on these huge information is also complex task. Hadoop framework provides a solution for this big data analysis. As a research work, forum of health care community is selected. In the forum, patients can post quires, share therapy followed, prescription prescribed. To make this forum more beneficial for beneficiary, the comments posted are ranked. For accurate ranking and quick access of information Content Based Mining along with Hidden Markov Model (HMM) is used

- These algorithms utilize less memory.
- Computational complexity is less.
- Execution time of this algorithm is less.
- Accuracy and efficiency of algorithm is high.
- Has good ranking accuracy.

Internet has become a media of communication. Web 2.0 increases its' impact and make it as social media where people can share all the information. Using this people is forming a group, forum for discussion etc. This discussion group becomes a vital source of information. This forum discussion is available for all fields like medical, computer science, engineering and technology. Since millions and millions of people is using net as forum information evolved over a net is huge and decision making based on these huge information is also complex task. Hadoop framework provides a solution for this big data analysis. As a research work, forum of health care community is selected. In the forum, patients can post quires, share therapy followed, prescription prescribed. To make this forum more beneficial for beneficiary, the comments posted are ranked. For accurate ranking and quick access of information Content Based Mining along with Hidden Markov Model (HMM) is used.

- Collection of data using web crawler
- Construction of social network
- Ranking user based on Hidden Markov Model
- Performance Evaluation

A. Collection of Data Using Web Crawler: In the forum data collecting phase, a crawler was built to collect all threads and replies on the discussion board of the forum of interest. In addition, a parser was built to parse and filter the collected data. For each thread, we generated a formatted thread record which consisted of TID (unique ID for each thread), Thread Title, Thread URL, Thread Initiator ID, Timestamp, List of Replier ID and Thread Content. For each reply of a thread, we created a formatted reply record which was composed of RID (unique ID for each reply), Thread Title, Thread URL, Replier ID, Timestamp and Reply Content. The formatted data was stored in a database which provided inputs to the social network constructing phase.

B. Construction of Social Network:

Social network is a convenient and effective way to represent user interactions. Each vertex of a social network represents a social actor. Two social actors who are interacting with each other are connected by an edge in a social network. Depending on the specific applications and interactions, a social network can be constructed in different way. A forum consists of a number of threads. A forum thread is composed of a number of messages. A social network is constructed by extracting the users and their interactions in a hierarchical tree of a thread based on three observations:

- ✓ Direct reply from a user to another user in a thread represents an interaction. In other words, when a replies to a message posted by B, there should be an edge connecting vertex A and vertex B in the social network to capture their interaction.
- ✓ Indirect reply is B replies to A and then C replies to B, it is possible that C is not only replying to B but also addressing to the message posted by A. In this case, three edges should be created in a social network corresponding to the interactions between A and B, B and C, and A and C.
- ✓ Most of the threads in medical support forum is initialized by the members who are seeking information help. When B is replying the question posted by A, it is considered that B is offering some kinds of support and influence to A. As a result, the direction of an edge should be made from the one who receives the reply to the one who makes the reply to confer authority.

A weight function is incorporated by both content similarity and response immediacy to compute weights, leading to a weighted social network G'=(V,E,W) where the node set V is a set of nodes corresponding to the members of a forum and edge set E is a set of edges corresponding to the interactions between members and the weight set W corresponds to a collection of weights $\{w_{i,j}\}$, for each edge in E.

Given a forum, there are a collection of N threads which consist of messages posted by n users $v_1, v_2, \ldots v_n$. Let M $_{k,l}$ to be the l th message of , V(M $_{k,l}$) to be the user who posts , and time (M $_{k,l}$) to be the timestamp of message . In addition, let ( ) denotes the content similarity between messages M $_{k,a}$ and M $_{k,b}$, and (M $_{k,a}$, M $_{k,b}$) represents the response immediacy between two messages. The weight $W_{i,j}$ of edge e $_{i,j}$ between v_i and v_j is computed.

C. Ranking User Based on Hidden Markov Model:

Given the weighted social network G', proposed two different approaches of computing user influence

- ✓ Weighted in-degree and User Rank
- ✓ Content Based Mining along with Hidden Markov Model

In a directed graph, In-degree of a node is the number of head endpoints adjacent to this node. With edge weights computed, weighted in-degree is a straightforward way of computing user influence. Given a weighted social network, user's influence score is equal to the sum of weights on all in-link edges of the network.

Since the user influence within a social network is similar to the web page popularity in a hyperlink network, User Rank algorithm is used to quantify user influence in a weighed social network that is constructed. To incorporate the content similarity and response immediacy, content base mining with Hidden Markov Model is used. In content base mining, based on the content similar users are identified and the identified similar users are grouped by Hidden Markov Model. By this approach user influence over the medical forum is evaluated and ranked efficiently.

D. Performance Evaluation: Performance a systematic determination of a subject's merit, worth and significance, using criteria governed by a set of standards. It can assist an organization, program, project or any other intervention or initiative to assess any aim, realizable concept proposal, or any alternative, to help in decision-making; or to ascertain the degree of achievement or value in regard to the aim and objectives and results of any such action that has been completed.

4. Experimental Analysis and Results:

System Analysis is the process of gathering and interpreting facts, diagnosing problems, and using the information or recommend improvements to the system. This is the job of the system analyst. In the case of systems analysis, the substance is the business system under investigation and the parts are the various sub-systems, which work together to support the business. Before designing a computer system, which will satisfy the information requirements of a company, it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination will then provide the design teams with the specific data they require in order to ensure that all the client's requirements are fully met. The investigation or study conducted during the analysis phase may build on the results of an initial feasibility study and will result the Production of a document, which specifies the requirements for a new system. This document is usually called the requirements specification or functional specification, and it is also described as a 'target document' because it establishes goals for the rest of the project.

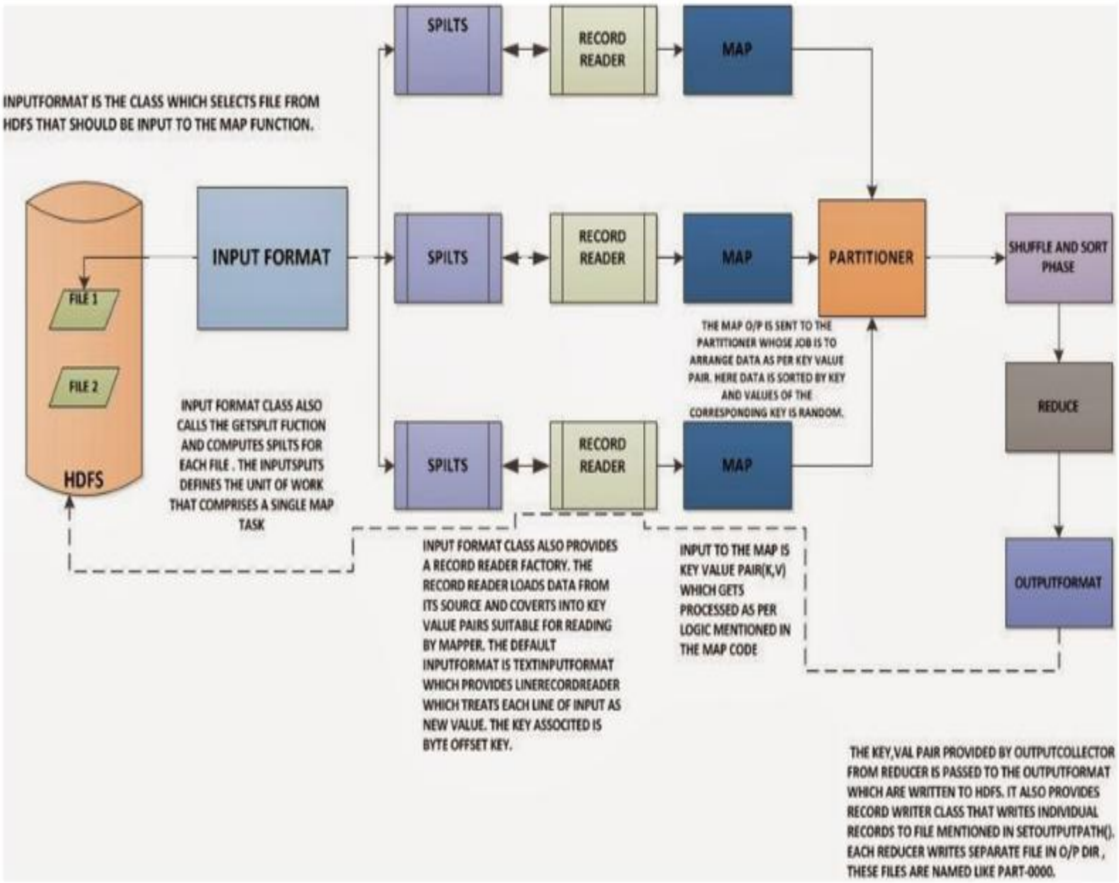
Java is a high-level, third generation programming language, like C, Fortran, Smalltalk, Perl, and many others. You can use Java to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of other things computer software can do. Compared to other programming languages, Java is most similar to C. However although Java shares much of C's syntax, it is not C. Knowing how to program in C or, better yet, C++, will certainly help you to learn Java more quickly, but you don't need to know C to learn Java. Unlike C++ Java is not a superset of C. A Java compiler won't compile C code, and most large C programs need to be changed substantially before they can become Java programs.

What's most special about Java in relation to other programming languages is that it lets you write special programs called applets that can be downloaded from the Internet and played safely within a web browser. Traditional computer programs have far too much access to your system to be downloaded and executed willy-nilly. Although you generally trust the maintainers of various ftp archives and bulletin boards to do basic virus checking and not to post destructive software, a lot still slips through the cracks. Even more dangerous software would be promulgated if any web page you visited could run programs on your system.

Java solves this problem by severely restricting what an applet can do. A Java applet cannot write to your hard disk without your permission. It cannot write to arbitrary addresses in memory and thereby introduce a virus into your computer. It should not crash your system. Object oriented programming is the catch phrase of computer programming in the 1990's. Although object oriented programming has been around in one form or another since the Simulate language was invented in the 1960's, it's really begun to take hold in modern GUI environments like Windows, Motif and the Mac. In object-oriented programs data is represented by objects. Objects have two sections, fields (instance variables) and methods. Fields tell you what an object is. Methods tell you what an object does. These fields and methods are closely tied to the object's real world characteristics and behavior. When a program is run messages are passed back and forth between objects. When an object receives a message it responds accordingly as defined by its methods.

Object oriented programming is alleged to have a number of advantages including:

- ✓ Simpler, easier to read programs
- ✓ More efficient reuse of code
- ✓ Faster time to market
- ✓ More robust, error-free code



INPUTFORMAT IS THE CLASS WHICH SELECTS FILE FROM HDFS THAT SHOULD BE INPUT TO THE MAP FUNCTION.

INPUT FORMAT CLASS ALSO CALLS THE GETSPLIT FUCTION AND COMPUTES SPILTS FOR EACH FILE . THE INPUTSPLITS DEFINES THE UNIT OF WORK THAT COMPRISES A SINGLE MAP TASK

THE MAP O/P IS SENT TO THE PARTITIONER WHOSE JOB IS TO ARRANGE DATA AS PER KEY VALUE PAIR. HERE DATA IS SORTED BY KEY AND VALUES OF THE CORRESPONDING KEY IS RANDOM.

INPUT FORMAT CLASS ALSO PROVIDES A RECORD READER FACTORY. THE RECORD READER LOADS DATA FROM ITS SOURCE AND COVERTS INTO KEY VALUE PAIRS SUITABLE FOR READING BY MAPPER. THE DEFAULT INPUTFORMAT IS TEXTINPUTFORMAT WHICH PROVIDES LINERECORDREADER WHICH TREATS EACH LINE OF INPUT AS NEW VALUE. THE KEY ASSOCITED IS BYTE OFFSET KEY.

INPUT TO THE MAP IS KEY VALUE PAIR(K,V) WHICH GETS PROCESSED AS PER LOGIC MENTIONED IN THE MAP CODE

THE KEY,VAL PAIR PROVIDED BY OUTPUTCOLLECTOR FROM REDUCER IS PASSED TO THE OUTPUTFORMAT WHICH ARE WRITTEN TO HDFS. IT ALSO PROVIDES RECORD WRITER CLASS THAT WRITES INDIVIDUAL RECORDS TO FILE MENTIONED IN SETOUTPUTPATH(). EACH REDUCER WRITES SEPARATE FILE IN O/P DIR , THESE FILES ARE NAMED LIKE PART-0000.

This makes Java very responsive to user input. It also helps to contribute to Java's robustness and provides a mechanism whereby the Java environment can ensure that a malicious applet doesn't steal all of the host's CPU cycles. fortunately multithreading is so tightly integrated with Java, that it makes Java rather difficult to port to architectures like Windows 3.1 or the PowerMac that don't natively support pre-emptive multi-threading. There is a cost associated with multi-threading. Multi-threading is to Java what pointer arithmetic is to C, that is, a source of devilishly hard to find bugs. Nonetheless, in simple programs it's possible to leave multi-threading alone and normally be OK. You do not need to explicitly allocate or deal locate memory in Java. Memory is allocated as needed, both on the stack and the heap, and reclaimed by the garbage collector when it is no longer needed. There's no mallow (), free (), or destructor methods. There are constructors and these do allocate memory on the heap, but this is transparent to the programmer. The exact algorithm used for garbage collection varies from one virtual machine to the next. The most common approach in modern VMs is generational garbage collection for short-lived objects, followed by mark and sweep for longer lived objects. I have never encountered a Java VM that used reference counting.

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any supported hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but are intended to be interpreted by a virtual machine(VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets. Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executable would. Just-in-Time compilers were introduced from an early stage that compiles byte codes to machine code during runtime. Over the years, this JVM built-in feature has been optimized to a point where the JVM's performance competes with natively compiled C code. These core values direct how My SQL works with the My SQL server software: To be the best and the most widely used database in the world. To be available and affordable by all. To be easy to use. To be continuously improved while remaining fast and safe My SQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by My SQL. My SQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access and process data stored in a computer database, you need a database management system such as My SQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities or as parts of other applications.

My SQL is a relational database management system .My SQL software is Open Source. The My SQL Database Server is very fast, reliable, and easy to use. My SQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environment for several years. Although under constant development, My SQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make My SQL Server highly suited for accessing databases on the Internet.MySQL Server works in client/server or embedded systems.

The Mosul Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces. We also provide My SQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage product .A large amount of contributed My SQL software is available. The following list describes some of the important characteristics of the My SQL Database Software. Tested with a broad range of different compilers. Works on many different platforms. APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available. Fully multi-threaded using kernel threads. It can easily use multiple CPUs if they are available. Provider's transactional and non-transactional storage N-memory hash tables, which are implemented using a highly optimized class library and should be as fast as possible.

Usually there is no memory allocation at all after query initialization. The My SQL code is tested with Purify. The server is available as a separate program for use in a client/server networked environment.

A privilege and password systems that is very flexible and secure, and that allows host-based verification. Passwords are secure because all password traffic is encrypted when you connect to server. Software testing is an important element of S/W quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of S/W as a system element and the costs associated with an S/W failure are motivating forces for well planned, through testing. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing. Testing is a set of activities that can be planned in advance and conducted systematically. Testing is a very important stage of a software include Unit Testing, Integration Testing and Deployment testing.

Unit testing focuses verification effort on the smallest unit of S/W design i.e., the module. The unit testing is always white-box oriented and the step can be conducted in parallel for modules. In Online examination system, unit testing is done to uncover the following errors: The module interfaces are tested to ensure that information flows properly into and out of the program and is equal to the number of arguments in stored procedure checking the parameter and argument attributes matching the stored procedures. Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. The objective is to take unit-tested modules and build a program structure that has been dictated by design.

In online examination system, the programs in various modules that are interfacing with other modules are tested thoroughly. Here we followed Top-Down integration and modules are integrated by moving downward through the control hierarchy, beginning with the Project related process, then activity related process and report generation process. In deployment testing us basically check for hard coded links. For smooth transfer of data from one page to another page in the system, we had to be sure there were no hard coded links. The scope of the objects and data was tested when they were transferred to another place. At the end of Integration testing, software is completely assembled as a package, interfacing errors have been uncovered and correction testing begins.

System implementation is the process of making the newly designed systems fully operational. The system is implemented after careful testing. The primary goal of product implementation is development of source code that is easy to read and easy to understand. The term implementation has different meanings, ranging from the conversion of a basic application to a compatible replacement of a computer system. Implementation is used here to mean the process converting a new or a revised system design in to an operational one. During the implementation stage we convert the detailed code in a programming language. Clarity of source code eases debugging, testing and modification of a software product. The difficulties encountered during implementation are caused by inadequate analysis and design.

The major milestone for project implementation is successful integration of source code components into a functioning system. Before a routine can be placed in the evolving system, it may be required that the routine be inspected by an inspection team, or reviewed or tested to a given level of test coverage. The first goal of implementation is to provide a faithful translation of design. The choice of a language should be pragmatic, governed by mixture theoretical needs and practical constraints. Good software should avoid any gap between design and code. This is particularly important for reuse of a component or for maintenance work that will require tracing the connection of design to code.

Abstraction deals with the ability of an implementation to allow the programmer to ignore the portion of detail that is not important at the current level of consideration. Each of the three kinds of abstraction-control, data, process should present in the code. Modularization requires as partitioning the implementation, with each abstraction occupying its own separate and identifiable unit. Assertions used during formal verification of the detailed design should be included as comments in the source code. Implementation is the stage of the project when theoretical design is turned into a working system. Maintenance of the software is one of the major steps in the computer automation. Software, which is developed by the engineer, should undergo maintenance process in a regular interval of time goes on new problems arise and it must be corrected accordingly. Maintenance and enhancements are a long-term process. If the problem is diverted or upgraded, then also the software should be changed.

5. Conclusion and Future Enhancement:

User influence on medical forum is analyzed using link and content based approach. A social network is constructed by collecting data using a crawler which collects all threads related to the user conversation. From the collected data set, social network is constructed. From the constructed network, user behavior is understood by assigning a weight to the links between user and the conversation. User influence is quantified using weighted-in degree and user rank algorithm. Along with this procedure, the content based mining is used with hidden Markova model. Content based mining is used to mine a people of similar behavior and Markova model is used to group similar behavior user. As a future work, ontology can be build for the user and communication relation to make a form. Ontology can be trained to retrieve a relation easier and efficiently than algorithm. Along with this, some other forum features and characteristics can examine. System is developed to accommodate further changes made.

6. References:

1. M. S. Marshall et al., "Emerging practices for mapping and linking life sciences data using RDF—A case series," J. Web Semantics, vol. 14, pp. 2–13, Jul. 2012.
2. M. J. Ibanez, J. Fabric, P. Olivarez, and J. Ezpeleta, "Model checking analysis of semantically annotated business processes," IEEE Trans. Syst., Man, Clyburn. A, Syst., Humans, vol. 42, no. 4, pp. 854–867, Jul. 2012.
3. V. R. L. She, "Correctness in hierarchical knowledge-based requirements," IEEE Trans. Syst., Man, Clyburn. B, CIBER, vol. 30, no. 4, pp. 625–631, Aug. 2000.
4. J. Guo, L. Xu, Z. Gong, C.-P. Che, and S. S. Chaudhry, "Semantic inference on heterogeneous e-marketplace activities," IEEE Trans. Syst., Man, Cyber. A, Syst., Humans, vol. 42, no. 2, pp. 316–330, Mar. 2012.
5. J. Cheng, C. Liu, M. C. Zhou, Q. Zen, and A. Ylä-Jääski, "Automatic composition of Semantic Web services based on fuzzy predicate Petri nets," IEEE Trans. Autos. Sci. Eng., Nov. 2013, to be published.
6. D. Kurtosis, J. M. Alvarez-Rodriguez, and I. Para kakis, "Semantic based Quos management in cloud systems: Current status and future challenges," Future Genre. Computer Syst., vol. 32, pp. 307–323, Mar. 2014.
7. Linking Open Data on the Semantic Web [Online]. Available: http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/Datasets/Statistics

8. M. Nagy and M. Vargas-Vera, "Multi agent ontology mapping framework for the Semantic Web," IEEE Trans. Syst., Man, Clyburn. A, Syst., Humans, vol. 41, no. 4, pp. 693–704, Jul. 2011.
9. J. Weaver and J. Handler, "Parallel materialization of the finite RDFS closure for hundreds of millions of triples," in Proc. ISWC, Chantilly, VA, USA, 2009, pp. 682–697.
10. J. Urban, S. Kotoulas, J. Massed, F. V. Hamelin, and H. Bal, "Weepier: A web-scale parallel inference engine using map reduce," J. Web Semantics, vol. 10, pp. 59–75, Jan. 2012.
11. J. Urbanity, S. Kotoulas, E. Oren, and F. Hameln, "Scalable distributed reasoning using map reduce," in Proc. 8th Int. Semantic Web Conf., Chantilly, VA, USA, Oct. 2009, pp. 634–649.
12. J. Dean and S. Ghemawat, "Map Reduce: Simplified data processing on large clusters," Communed. ACM, vol. 51, no. 1, pp. 107–113, 2008.
13. C. Anagnostopoulos and S. Hadjiefthymiades, "Advanced inference in situation-aware computing," IEEE Trans. Syst., Man, CIBER. A, Syst., Humans, vol. 39, no. 5, pp. 1108–1115, Sep. 2009.
14. H. Paulheim and C. Baser, "Type inference on noisy RDF data," in Proc. ISWC, Sydney, NSW, Australia, 2013, pp. 510–525.
15. G. Antoniou and A. Bikakis, "DR-Prologue: A system for defensible reasoning with rules and ontology's on the Semantic Web," IEEE Trans. Known. Data Eng., vol. 19, no. 2, pp. 233–245, Feb. 2007.
16. V. Mila, F. Frasincar, and U. Kayak, "OWL: A temporal web ontology language," IEEE Trans. Syst., Man, Cyber. B, Clyburn, vol. 42, no. 1, pp. 268–281, Feb. 2012.
17. D. Lopez, J. M. Sempere, and P. García, "Inference of reversible tree languages," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 34, no. 4, pp. 1658–1665, Aug. 2004.
18. Skylight and H. Stuckenschmidt, "Map Resolve," in Proc. 5th Int. Conf. RR, Galway, Ireland, Aug. 2011, pp. 294–299.
19. B. C. Graz, C. Halaschek-Wiener, and Y. Kasyanov, "History matters: Incremental ontology reasoning using modules," in Proc. ISWC/ASWC, Bus an, Korea, 2007, pp. 183–196.
20. RDF Semantics [Online]. Available: http://www.w3.org/TR/rdf-mt/
21. RDF Schema [Online]. Available: http://en.wikipedia.org/wiki/RDFS
22. SPARQL 1.1 Overview [Online]. Available: http://www.w3.org/TR/sparql11-overview/
23. Hardtop [Online]. Available: http://hadoop.apache.org/
24. Abase [Online]. Available: http://hbase.apache.org/
25. Billion Triples Challenge 2012 Dataset [Online]. Available: http://km.aifb.kit.edu/projects/btc-2012/
26. Y. Guo, Z. Pan, and J. Heflin, "LUBM: A benchmark for OWL knowledge base systems," J. Web Semantics, vol. 3, nos. 2–3, pp. 158–182, Oct. 2005.