ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (861-871), Month: April - June 2015, Available at: www.researchpublish.com

Technologies Used In Trending Database Model

[1]ABHIJIT D. NAGVENKAR, [2]ROSHAN S. DUMBRE

[1, 2] MCA Department, Mumbai University, Thane, India

Abstract: **Traditional applications had a common platform that captured business transactions. The emerging class of applications requires new functionality that closes the loop between incoming transactions and the analytics that drive action on those transactions. Adopting a longer term and more abstract perspective, it is likely that new database management technologies and application areas will continue to emerge. As with any sufficiently fashionable technology, users should expect data management marketplace to yield solutions that are increasingly application/environment specific.**

Keywords: **Emerging Database, Multimedia, Temporal, Technology, Big Data, GIS.**

I. INTRODUCTION

The term database refers to the collection of related records, and the software should be referred to as the database management system or DBMS. Database management systems are usually categorized according to the data model that they support: relational, object-relational, network, and so on. The data model will tend to determine the query languages that are available to access the database. The world of database boasts many kinds of technologies, which cater to the need of many kinds of organizations. Since 1970 different models and methods have been developed to describe, analyze and design computer based files and databases. The existing relational DBMS technology has been successfully applied to many application domains. RDBMS technology has proved to be an effective solution for data management requirements in large and small organizations, and today this technology forms a key component of most information systems. However, Applications in domains such as Multimedia, Geographical Information Systems, mobile database etc. demand a completely different set of requirements in terms of the underlying database models. The conventional relational database model is no longer appropriate for these types of data. Furthermore the volume of data is significantly larger than in classical database systems. Finally, indexing, retrieving and analyzing these data types require specialized functionality that are not available in conventional database systems. This paper will cover some requirements of these emerging databases such as multimedia database, spatial database, temporal database, biological/genome database, mobile database, big data, their underlying technologies, data models and languages. These trends have resulted into the development of new database technologies to handle new data types and applications.

II. MULTIMEDIA DATABASE

A multimedia computer system is a computer system that can create, import, integrate, store, retrieve, edit, and delete two or more types of media materials in digital form, such as audio, image, full-motion video, and text information. Multimedia computer systems also may have the ability to analyze media materials (e.g., counting the number of occurrences of a word in a text file). A multimedia computer system can be a single- or multiple-user system. Networked multimedia computer systems can transmit and receive digital multimedia materials over a single computer network or over any number of interconnected computer networks. As multimedia computer systems evolve, they may become intelligent systems by utilizing expert system technology to assist users in selecting, retrieving, and manipulating multimedia information.

This paper surveys four possible types of multimedia computer systems: hypermedia, multimedia database, multimedia message, and virtual reality systems. It examines the potential benefits and problems associated with the use of multimedia computer systems as public-access computer systems, which can be employed directly by library patrons.Without question, multimedia computer systems will have a profound impact on library systems that are used for internal purposes; however, this area is beyond the scope of the present paper. This paper also does not attempt to survey the wide array of supporting hardware (e.g., CD-I, CD-ROM XA, and DVI) and software products (e.g., BASISplus and HyperCard) that will be used to build multimedia computer systems. Several recent papers introduce the reader to these topics.Rather, it primarily focuses on advanced multimedia system development projects and theoretical efforts that suggest long-term trends in this increasingly important area.

Multimedia Database Systems:

Multimedia database systems are analogous to contemporary database systems for textual and numeric data; however, multimedia database systems have been tailored to meet the special requirements of dealing with different types of media materials. Multimedia database systems can create, import, integrate, store, retrieve, edit, and delete multimedia information. They may incorporate some hypermedia capabilities. Multiuser multimedia database systems are likely to perform these functions in a manner that reduces redundant data storage, permits different views of data by users, and provides secure access to data. Multimedia database systems are in an early stage of development; however, both theoretical and experimental work have been done.

The multimedia database would utilize what is known as an "object-oriented architecture." Object orientation is a powerful and increasingly popular concept, albeit a somewhat complex one. In brief, objects are organized into hierarchical classes based on their common characteristics.Lower-lower objects "inherit" the characteristics of their antecedents. Objects may be composed of many different components. The specific details of how objects are implemented are hidden from the system as a whole; however, objects will respond to specific structured messages and perform appropriate actions. Objects have unique "identities" that transcend their temporary characteristics.

Based on textual or verbal search keys, the system would retrieve objects and present the user with image or voice representations of these objects. The user would then select the object of interest. The system would permit the user to browse within the multimedia object using several techniques:

1) By "page" (a page could be text, combined text and image, or audio);

2) By marked subunits of the object, such as section or chapter; and

3) By pattern matching. When an object was retrieved, one media presentation mode (i.e., visual or audio) would be dominant;

However, information in another mode could be attached to it. Visual pages, for instance, could have audio annotations. Objects could, in hypermedia fashion, also be linked to external objects. In visual mode, images could be viewed in reduced form and portions of them could be selected for close-up inspection. Images also could be presented superimposed over each other like overhead transparencies. A pre-defined sequence of visual pages could be shown automatically, permitting the author of the multimedia object to imitate a slide-tape presentation or to create basic animation effects. Finally, executable programs, which could be embedded in multimedia objects, could accept user input and perform certain pre-defined functions.

Major points in a multimedia sequence, which would be viewed on the left-hand side of the screen, would be described by a vertical row of icons in the right-hand side of the screen. Shaded icons would provide hypermedia links to different multimedia sequences. In addition to symbols, still images from a multimedia sequence or words could be employed as icons. These icons could be browsed to move to different points in the multimedia sequence. Horizontal lines between icons, much like the markings on a ruler, would indicate the time intervals between icons. The marked points between icons also could be directly accessed. Above the vertical row of icons would be a "context icon" and an "elapsed time indicator." The context icon would identify the broader unit of information that the user is browsing; these icons also could be browsed. The indicator would show the number of elapsed seconds from the beginning of the unit of information represented by the context icon, and the user could enter a different elapsed time to move to that point in the presentation. The elapsed time indicator also could be used to freeze or re-start a presentation. It would be necessary to stop the presentation in order to perform the various browsing functions of the system.

In the object-oriented HMD system, a network controller would analyze the user's database command, identify the server or servers that housed needed multimedia information, designate one server as the "master" server where the majority of processing would occur, decide how to process information from multiple servers, and perform general network management functions. The master multimedia server, using the services of its local controller, would, if required, integrate the multimedia information from all participating servers for delivery to the user workstation. Each multimedia server, which would be a multiprocessor system with substantial memory, could house multiple database management systems, each oriented towards a particular type of data (e.g., image). A broadband optical fiber network, operating at speeds as high as 2-5 gigabits per second, would provide data transmission services for the HMD system. The need for a high-speed network is shown by the projected transmission speeds for two types of media:

1) still image--50 kilobits per second to 48 megabits per second, contingent on the resolution and color characteristics of the image; and

2) full-motion video in the High-Definition Television format--1.2 gigabits per second without compression and 200-300 megabits per second with compression.

In Germany, the BERKOM (Berliner Communications system) project is developing the Multi-Media Document Model, a standard for providing access to multimedia documents via Broadband Integrated Services Digital Network (B-ISDN) systems. By focusing on B-ISDN technology, the BERKOM project is bypassing contemporary Integrated Services Digital Network (ISDN) technology in order to achieve the higher speeds required to transmit a full range of digital multimedia materials. While ISDN systems provide users with 64 kilobits-per-second data channels, the evolving B-ISDN standard is likely to support 135.168 megabits-per-second data channels.

The BERKOM project's Multi-Media Document Model, which is based on the Open Systems Interconnection (OSI) Reference Model, has two components: the Data Model and the Communication Model. The Data Model describes different types of information: text, graphic, audio/speech, raster image, video/movie, modelling data, special forms (e.g., mathematical and chemical formulas), and transparent (i.e., additional data that is not apparent to the user). The Communication Model describes the telecommunications services required to deliver multimedia documents.

III. TEMPORAL DATABASE

Time is an important aspect of real world phenomena. Events occur at specific points in time. Objects and relationships among objects exist over time. The ability to model this temporal dimension of real world is essential to many computer applications such as econometrics, inventory control, airline reservations, medical records, accounting, law, banking, land and geographical information systems. In contrast, existing database technology provides little support for managing such data. A temporal database is formed by compiling and storing temporal data. The difference between temporal data and non-temporal data is that a time period is appended to data expressing when it was valid or stored in the database. The data stored by conventional databases consider data to be valid at present time as in the time instance ―now. When data in such a database is modified, removed or inserted, the state of the database is overwritten to form a new state. The state prior to any changes to the database is no longer available. Thus, by associate time with data, it is possible to store the different database states. In essence, temporal data is formed by time-stamping ordinary data (type of data we associate and store in conventional databases). In a relational data model, tuples are time-stamped and in an object oriented data model, objects/attributes are time stamped. Each ordinary data has two time values attached to it, a start time and an end time to establish the time interval of the data. In a relational data model, relations are extended to have two additional attributes, one for start time and another for end time.

Different Forms of Temporal Databases:

Transaction Time – TT

A flight data recorder collects and records various metrics during a flight to allow the reconstruction of the past. The transaction or system time in a data model is comparable to the functionality of such a flight data recorder. A table with a transaction time axis allows to query the current and the past state, but changes in the past or in the future are not possible.

Example: Scott becomes a manager. The change of the job description from “Analyst” to “Manager” is entered into the system on the April, 15 2013 at 15:42:42. The previous description Analyst is terminated at this point in time and the new description Manager becomes current at exactly the same point in time.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (861-871), Month: April - June 2015, Available at: www.researchpublish.com

Oracle supports the transaction time with Flashback Data Archive (formally known as Total Recall). Using Flashback Data Archive you may query a consistent state of the past

Valid Time – VT

The valid time describes the period during which something in the real world is considered valid. This period is independent of the entry into the system and therefore needs to be maintained explicitly. Changes and queries are supported in the past as well as in the future.

Example: Scott becomes a manager. The change of the job description from "Analyst" to "Manager" is valid from January, 1 2014. The previous description Analyst is terminated at this point in time and the new description Manager becomes valid at exactly the same point in time. It is irrelevant when this change is entered into the System.

Decision Time – DT

The decision time describes the date and time a decision has been made. This point in time is independent of an entry into the System and is not directly related to the valid time period. Future changes are not possible.

Example: Scott becomes manager. The decision to change the job description from "Analyst" to "Manager" has been made on March, 24 2013. The previous job description Analyst is terminated on the decision time axis at this point in time and the new description Manager becomes current at exactly the same point in time on the decision time axis. It is irrelevant when this change is entered into the System and it is irrelevant when Scott may call himself officially a manager.

Multi-temporal Features in Oracle 12c

Oracle 12c has a feature called Temporal Validity. With Temporal Validity you can add one or more valid time dimensions to a table using existing columns, or using columns automatically created by the database. This means that Oracle offers combined with Flashback Data Archive native bi-temporal and even multi-temporal historization features.

Semantics and Granularity of Periods

In Flashback Data Archive Oracle defines periods with a half-open interval. This means that a point in time x is part of a period if x >= the start of the period and x < the end of the period. It is no surprise that Oracle uses also half-open intervals for Temporal Validity. The following figure visualizes the principle:

Fig. 1: Semantics and Granularity of Periods

The advantage of a half-open interval is that the end of a preceding period is identical with the start of the subsequent period. Thus there is no gap and the granularity of a period (year, month, day, second, millisecond, nanosecond, etc.) is irrelevant. The disadvantage is that querying data at a point in time using a traditional WHERE clause is a bit more verbose compared to closed intervals since BETWEEN conditions are not applicable.

IV. MOBILE DATABASE

Mobile computing is increasingly becoming more and more popular as people need information even on the move in this rapid changing information world. This unit is an attempt to highlight the concepts and basic issues relating to mobile computing. This unit is not attempting the details but just an introduction of various issues.

What is mobile Databases?

Traditionally, large-scale commercial databases were developed as centralized database systems. However, this trend changed as more and more distributed applications stated to emerge. The distributed database applications involved usually a strong central database and powerful network administration. However, the newer technology trends have changed this paradigm because of the following technological trends:

- The notebook and laptop Computers are being used increasingly among the Business Community
- The development and availability of a relatively low-cost wireless digital communication infrastructure. This infrastructure is based on wireless local-area networks, cellular digital packet networks, and other technologies

The rapid advancements of wireless communication technology and computer miniaturizing technology have enabled users to utilize computing resources anywhere in the computer network. For example, you can even connect to your Intranet from an aero plane. Mobile database are the database that allows the development and deployment of database applications for handheld devices, thus, enabling relational database based applications in the hands of mobile workers. The database technology allows employees using handheld to link to their corporate networks, download data, work offline, and then connect to the network again to synchronize with the corporate database. For example, with a mobile database embedded in a handheld device, a package delivery worker can collect signatures after each delivery and send the information to a corporate database at day's end.

The current database systems do not provide special facilities for specific update operations in a mobile computing environment. Some of the commercially available Mobile relational Database systems are:

IBM's DB2 Everywhere 1.0

Oracle Lite

Sybase's SQL

These databases work on Palm top and hand held devices (Windows CE devices) providing a local data store for the relational data acquired from enterprise SQL databases. The main constraints for such databases are relating to the size of the Program as the handheld devices have RAM oriented constraints. The commercially available mobile database systems allow wide variety of platforms and data sources. They also allows users with handheld to synchronize with Open Database Connectivity (ODBC) database content, and personal information management data and email from Lotus Development's Notes or Microsoft's Exchange. These database technologies support either query-by-example (QBE) or SQL statements.

Mobile computing has proved useful in many applications. Many business travelers are using laptop computers to enable them to work and to access data while traveling. Delivery services may use/ are using mobile computers to assist in tracking of delivery of goods. Emergency response services may use/ are using mobile computers at the disasters sites, medical emergencies, etc. to access information and to provide data pertaining to the situation. Newer applications of mobile computers are also emerging.

One of the issue relating to wireless computing is that creates a situation where machines no longer have fixed locations and network addresses. This may complicate query processing for the cases where location plays a key role, since it becomes difficult to determine the optimal location at which to materialize the result of a query. This may happen only for the cases where the location of the user is a parameter of the query. For example, if a traveler information system provides data on hotels, roadside services, etc. to motorists; queries about services that are ahead on the current route must be processed based on knowledge of the user's location, direction of motion, and speed.

Another issue relating to mobile computing is the energy (battery power). It is a scarce resource for mobile computers. This limitation influences many aspects of system design. Can we reduce the requirements of data transfer for the sake of energy efficiency? Yes, by doing scheduled data broadcasts, we may reduce the need for mobile systems to transmit queries.

But on the other side it will increase the amount of data residing on machines administered by users, rather than by database administrators. In addition, these machines may, at times, be disconnected from the network; thus, raising the question about the consistency of data

V. GEOGRAPHIC INFORMATION SYSTEMS

A geographic information system, or GIS, is a computerized data management system used to capture, store, manage, retrieve, analyze, and display spatial information. Data captured and used in a GIS commonly are represented on paper or other hard-copy maps. A GIS differs from other graphics systems in several respects. First, data are geo-referenced to the

coordinates of a particular projection system. This allows precise placement of features on the earth's surface and maintains the spatial relationships between mapped features. As a result, commonly referenced data can be overlaid to determine relationships between data elements. For example, soils and wetlands for an area can be overlaid and compared to determine the correspondence between hydric soils and wetlands. Similarly, land use data for multiple time periods can be overlaid to determine the nature of changes that may have occurred since the original mapping. This overlay function is the basis of change detection studies across landscapes.

Second, GIS software use relational database management technologies to assign a series of attributes to each spatial feature. Common feature identification keys are used to link the spatial and attribute data between tables. A soil polygon, for example, can be linked to a series of database tables that define its mineral and chemical composition, crop yield, land use suitability, slope, and other characteristics.

Third, GIS provide the capability to combine various data into a composite data layer that may become a base layer in a database. For example, slope, soils, hydrography, demography, wetlands, and land use can be combined to develop a single layer of suitable hazardous waste storage sites. These data, in turn, may be incorporated into the permanent database of a local government and used for regulatory and planning decisions.

GIS software generally allow for two types of data. Some use raster data (i.e., discrete cells in a rigid row by column format), such as satellite imagery or aerial photography, while others use vectors (points, lines and polygons) to represent features on the earth's surface. Most systems allow for full integration of both types of data. In either case, a fully functioning GIS allows the user to enter or digitize data that are geo-referenced; link specific attributes to each feature using relational database management system technology; analyze relationship between various geographic features using a wide range of spatial operations and functions; and produce high-resolution images or graphics on color monitors or plotters.

How is a GIS used?

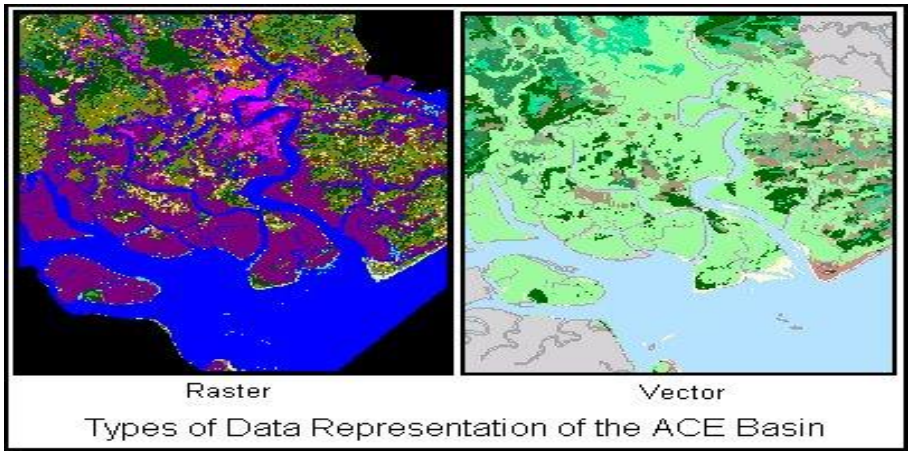
A GIS can be used to answer basic locational questions such as: What is located at a given point on the earth; or where is a specific feature located? For example, using a mouse-driven cursor, a specific point on a map can be queried to determine its land use, vegetation, soil type, elevation, and land ownership characteristics. Similarly, soils data across an entire watershed can be queried to determine the distribution of areas with hydric soils of greater than 100 acres and are adjacent to a major river system. In the first case, a specific, known point was identified and queried to determine preselected attributes. In the second case, however, specific locations were not known. Rather, the database was searched by the GIS to determine where specific conditions were satisfied (hydric class, size restrictions, and neighboring or adjacent feature characteristics).

One of the more powerful functions of a GIS is that it allows users to synthesize or combine different layers of information to identify resource distribution patterns that may otherwise not be obvious. For example, using various map overlay techniques, threatened and endangeredspecies data may be combined with wetland information to determine if any of the freshwater tidal wetlands in an area provide habitat for sensitive or critical species. This information could be used to develop specialized resource management plans that protect critical wetlands or it could be used to identify areas where the reintroduction of a threatened or endangered species might be successful. This information also can be used in the design of survey strategies and methods to focus on areas of potential threatened and endangered species locations.

A GIS also can be used for complex modeling to answer a wide range of "what if" and ecosystem simulation questions. These may be cartographic models designed to document the co-occurrence or interrelationship of multiple data layers or they may be hypothetical research models designed to mimic natural ecological systems. Similarly, modeling with GIS can be used to predict the impacts that one set of parameters will have on another. For example, wetlands, soils, hydrography, climatology and elevation data can be combined to model flooding within a river system. Upstream changes in land use within the same system can be modeled to determine the potential impact of conversion of a forested floodplain to residential development or to agriculture. As a result, both natural system responses to storm events and the impact of human land use decisions can be assessed prior to the proposed action.

Regardless of the application in which GIS technology is used, these systems provide rapid data access and multidimensional analysis and graphical output capabilities that can result in more effective resource management decisions.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (861-871), Month: April - June 2015, Available at: www.researchpublish.com



VI. GENOME DATA

The biological sciences encompass an enormous variety of information. Environmental science gives us a view of how species live and interact in a world filled with natural phenomena. Biology and ecology study particular species. Anatomy focuses on the overall structure of an organism, documenting the physical aspects of individual bodies. Traditional medicine and physiology break the organism into systems and tissues and strive to collect information on the workings of these systems and the organism as a whole. Histology and cell biology delve into the tissue and cellular levels and provide knowledge about the inner structure and function of the cell. This wealth of information that has been generated, classified, and stored for centuries has only recently become a major application of database technology. Genetics has emerged as an ideal field for the application of information technology. In a broad sense, it can be taught of as the construction of models based on information about genes – which can be defined as units of heredity – and population and the seeking out of relationships in that information. The study of genetics can be divided into three branches:

- Mendelian genetics. This is the study of the transmission of traits between generations.
- Molecular genetics. This is the study of the chemical structure and function of genes at the molecular level.
- Population genetics. This is the study of how genetic information varies across populations of organisms.

Google's Cloud-Based Genomics Database a Boon to Autism Research

Google's massive servers will store and analyze 10,000 complete genomes of autism patients and related clinical data, the largest project of its kind, in order to make it easier for researchers to devise better treatments or find a possible cure for autism. Scientists are using genomics - the study of genetic material consisting of chromosomes, genes, and DNA - to find clues on how certain diseases originate and how they can be treated.

By mapping and sequencing the human genome, researchers can someday find the cure for genetically-linked conditions such as cancer, Alzheimer's disease, and autism spectrum diseases, to name a few.The information about a whole genome can take up about 100 gigabytes of memory. But it takes many genomes to glean meaningful research insights. Sequencing a million genomes would generate more than 100 petabytes of data. Storing that amount of data presents a challenge for most researchers, because universities, hospitals and research facilities have finite computer resources. As part of its Google Genomics initiative, which was launched earlier this year, Google's solution is to host this data using a cloud-based, open-access platform that can be used by researchers from multiple locations without having to deal with limited storage issues, and for them to leverage the computing power of the company's massive data centers. The technology giant recently announced that it is collaborating with Autism Speaks, the autism research and advocacy organization, to store and sequence 10,000 complete genomes of autistic kids and their families, plus related clinical data collected over the past 15 years.The data from the Autism Speaks Ten Thousand Genomes Program (AUT10K) amounts to roughly 100 TB, the largest, private repository of genome data of autism patients by far.Scientists from anywhere in the world who have limited IT resources can access this valuable data via the cloud, and with the help of Google's computing prowess and analytical tools, they can uncover breakthroughs in understanding and treating autism.

VII. BIG DATA

Big data is data that exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or doesn't fit the strictures of your database architectures. To gain value from this data, you must choose an alternative way to process it.

Big data has become viable as cost-effective approaches have emerged to tame the volume, velocity and variability of massive data. Within this data lie valuable patterns and information, previously hidden because of the amount of work required to extract them. To leading corporations, such as Walmart or Google, this power has been in reach for some time, but at fantastic cost. Today's commodity hardware, cloud architectures and open source software bring big data processing into the reach of the less well-resourced. Big data processing is eminently feasible for even the small garage startups, who can cheaply rent server time in the cloud.

- Volume. Rather than just capturing business transactions and moving samples and aggregates to another database for analysis, applications now capture all possible data for analysis.
- Velocity. Traditional transaction-processing applications might have captured transactions in real time from end users, but newer applications are increasingly capturing data streaming in from other systems or even sensors. Traditional applications also move their data to an enterprise data warehouse through a deliberate and careful process that generally focuses on historical analysis.
- Variety. The variety of data is much richer now, because data no longer comes solely from business transactions. It often comes from machines, sensors and unrefined sources, making it much more complex to manage.

Apache HBase:

Apache HBase is a distributed, scalable data store that runs on top of Apache Hadoop's file system, the Hadoop Distributed File System (HDFS). HBase is a key component of an enterprise data hub (EDH), as its design caters to applications that require fast, random access tosignificant data sets. HBase, which is modeled after Google's BigTable, can handle massive data tables containing billions of rows and millions of columns.

HBase for the Enterprise:

Serving data to many users or applications:

Apache HBase is built to scale. Traditional relational databases are not inherently distributed, and as the number of users interacting with the database (i.e. reading and writing data) grows, the storage, memory and CPU requirements can quickly grow beyond what a single machine can accommodate. Scaling traditional systems can be costly to build and cumbersome to operate. HBase is distributed by design; the system is architected to leverage the cost-effective capabilities of Hadoop and an EDH and utilize the storage, memory, and CPU resources of any number of servers within a cluster so that the database scales horizontally as load and performance demands increase. Users can query data in HBase using a number of computing engines offered by an EDH, including interactive SQL with Cloudera Impala and full-text, faceted search with Cloudera Search.

Providing fast, random read/write access to users and applications:

HDFS is a "write once read many" (WORM) file system that is well suited for batch processing and interactive SQL and search operations. HDFS emphasizes high throughput computing rather than low latency I/O. HBase augments HDFS by providing record-based storage layer that users and applications use to perform fast, random reads and writes to data. Changes are efficiently cataloged in memory to achieve maximum access while the data is persisted to HDFS. This design enables a Hadoop-based EDH to serve random reads and writes to users and applications in real time yet still enjoy the fault-tolerance and durability of HDFS.

Key Features of Apache HBase

- **Scale-out Architecture** - add servers to increase capacity
- **Full Consistency** - Guard against node failures or simultaneous writes to the same record
- **High Availability** - Multiple master nodes ensure continuous access to data

- **Automatic Sharding** - Transparently and efficiently scale out your data across machines in the cluster
- **Active-active Replication** - Stream data across locations for disaster recovery and data protection
- **Security** - Secure table and column family-level access via Kerberos
- **SQL Access** - Query data interactively with Cloudera Impala and for batch processing with Apache Hive
- **Full-text, Faceted Search** - Give non-technical users and your applications a familiar yet powerful, interactive search experience

Hadoop is the foundation of most big data architectures. The progress from a Hadoop 1's more restricted processing model of batch oriented MapReduce jobs, to more interactive and specialized processing models of Hadoop 2 will only further position the Hadoop ecosystem as the dominant big data analysis platform.

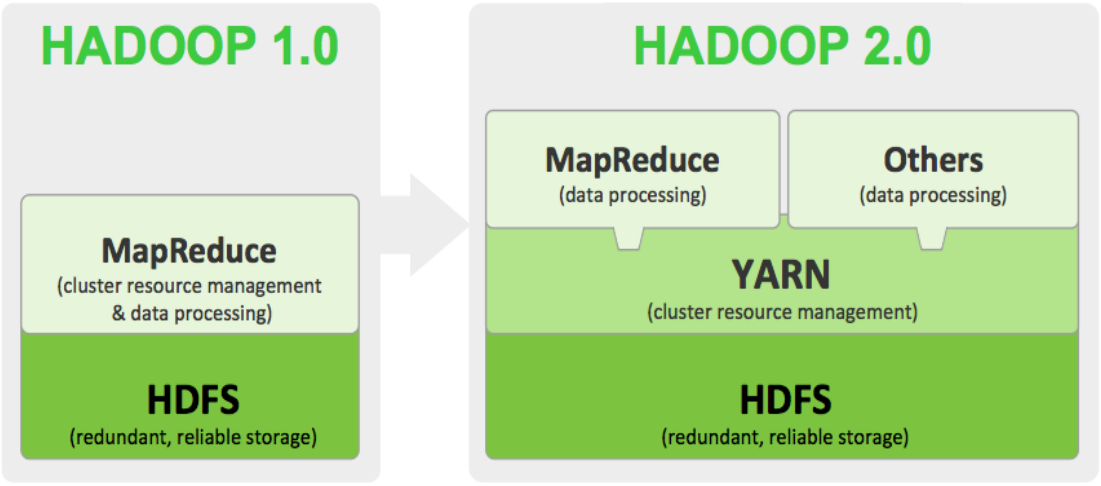
Hadoop 1 popularized MapReduce programming for batch jobs and demonstrated the potential value of large scale, distributed processing. MapReduce, as implemented in Hadoop 1, can be I/O intensive, not suitable for interactive analysis, and constrained in support for graph, machine learning and on other memory intensive algorithms. Hadoop developers rewrote major components of the file system to produce Hadoop 2. To get started with the new version, it helps to understand the major differences between Hadoop 1 and 2.

Two of the most important advances in Hadoop 2 are the introduction of HDFS federation and the resource manager YARN.

Hadoop 2: HDFS

HDFS is the Hadoop file system and comprises two major components: namespaces and blocksstorage service. The namespace service manages operations on files and directories, such as creating and modifying files and directories. The block storage service implements data node cluster management, block operations and replication.

In Hadoop 1, a single Namenode managed the entire namespace for a Hadoop cluster. With HDFS federation, multiple Namenode servers manage namespaces and this allows for horizontal scaling, performance improvements, and multiple namespaces. The implementation of HDFS federation allows existing Namenode configurations to run without changes. For Hadoop administrators, moving to HDFS federation requires formatting Namenodes, updating to use the latest Hadoop cluster software, and adding additional Namenodes to the cluster.



Hadoop Architecture

Hadoop 2: YARN:

HDFS federation brings important measures of scalability and reliability to Hadoop. YARN, the other major advance in Hadoop 2, brings significant performance improvements for some applications, supports additional processing models, and implements a more flexible execution engine.

YARN is a resource manager that was created by separating the processing engine and resourcemanagement capabilities of MapReduce as it was implemented in Hadoop 1. YARN is often called the operating system of Hadoop because it is responsible for managing and monitoring workloads, maintaining a multi-tenant environment, implementing security controls, and managing high availability features of Hadoop.

Like an operating system on a server, YARN is designed to allow multiple, diverse user applications to run on a multi-tenant platform. In Hadoop 1, users had the option of writing MapReduce programs in Java, in Python, Ruby or other scripting languages using streaming, or using Pig, a data transformation language. Regardless of which method was used, all fundamentally relied on the MapReduce processing model to run.

YARN supports multiple processing models in addition to MapReduce. One of the most significant benefits of this is that we are no longer limited to working the often I/O intensive, high latency MapReduce framework. This advance means Hadoop users should be familiar with the pros and cons of the new processing models and understand when to apply them to particular use cases.

VIII. NOSQL DATABASES

The term NoSQL has been around for just a few years and was invented to provide a descriptor for a variety of database technologies that emerged to cater for what is known as "Web-scale" or "Internet-scale" demands. In computing, NoSQL (commonly interpreted as "not only SQL") is a broad class of database management systems identified by nonadherence to the widely used relational database management system model. NoSQL databases are not built primarily on tables, and generally do not use SQL for data manipulation. NoSQL database systems are often highly optimized for retrieval and appending operations and often offer little functionality beyond record storage (e.g. key–value stores). The reduced run-time flexibility compared to full SQL systems is compensated by marked gains in scalability and performance for certain data models. In short, NoSQL database management systems are useful when working with a huge quantity of data when the data's nature does not require a relational model. The data can be structured, but NoSQL is used when what really matters is the ability to store and retrieve great quantities of data, not the relationships between the elements.

MongoDB is one of the vendors that leads the pack among NoSQL Document Databases. “The Leaders we identified offer mature, high-performance, scalable, flexible, secure, and robust NoSQL document database solutions that are enterprise-ready,” writes Noel Yuhanna, Principal Analyst at Forrester in the report. “MongoDB is the most popular NoSQL document database … because it’s easy to rapidly iterate its schema when doing iterative development and deployment.”

As cited in the report, MongoDB was given the highest scores for current offering and market presence, including development, operations, adoption and partnership. Each of the four vendors evaluated had established, enterprise-class functionality and a standalone database solution that is currently being used by 20 or more enterprises in more than one major geographical region.

NoSQL is gaining strong momentum and offering new use cases over relational databases according to the Forrester report. It states, “As data structures get more complex and data volume grows, traditional relational databases … are falling short. [Users] seek solutions that are designed to scale-out horizontally on lower-cost commodity servers, manage semi-structured data more efficiently, and provide a flexible data model to support content and document management system and next-generation web, mobile and cloud applications.”

“The database market is changing quickly. We see MongoDB chosen time and again by leading enterprises over relational database solutions as MongoDB enables organizations to develop, scale and manage modern applications they could never build before,” said Dev Ittycheria, President and CEO of MongoDB. “We believe MongoDB’s designation by Forrester as a leader of NoSQL Document Databases is a testament to the highly innovative work we do and further confirms MongoDB as a scalable and high-performance NoSQL database solution.”

Applications and use cases suitable for NoSQL document databases include:

-- Supported real-time analytics;
Applications that need horizontal scale-out across many servers;
-- Mobile apps that support documents;

Embedded database applications for independent software vendors and value-added resellers; and
-- Document and content management systems.

About MongoDB Inc.

MongoDB is the next-generation database that helps businesses transforms their industries by harnessing the power of data. The world's most sophisticated organizations, from cutting-edge startups to the largest companies, use MongoDB to create applications never before possible at a fraction of the cost of legacy databases. MongoDB is the fastest-growing database ecosystem, with over 8 million downloads, thousands of customers, and over 650 technology and service partners

IX. CONCLUSIONS

Applications in domains such as Multimedia, Geographical Information Systems, big data demand a completely different set of requirements in terms of the underlying database models which conventional relational database can no longer handle. The conventional relational database model is no longer appropriate for these types of data. Furthermore the volume of data is typically significantly larger than in classical database systems. Finally, indexing, retrieving and analyzing these data types require specialized functionality, which is not available in conventional database systems. Hence, a new direction, such as described above, in DBMS is necessary.

REFERENCES

[1] Trends and Challenges in Database Development -Ellen Munthe-Kaas.

[2] Temporal Information Processing Technology and Its Applications.

[3] http://answers.yahoo.com/question/index?qid=20070705045634AAcuhyF

[4] http://www.gislounge.com/what-is-gis/

[5] http://www.tech-faq.com/multimedia-database.html

[6] http://www.genenames.org/useful/genome-databases-and-browsers NoSQL Market Forecast 2013-2018

[7] http://www.marketresearchmedia.com/?p=568

[8] http://www.zdnet.com/article/2015-interesting-big-data-and-analytics-trends/

[9] http://www.infoq.com/research/nosql-databases NoSQL Database Adoption Trends [10]

[10] http://en.wikipedia.org/wiki/GIS

[11] https://www.mongodb.com/nosql-explained

[12] http://www.tech-faq.com/multimedia-database.html [13]Planning for Big Data – O'Relly Radar Team

[13] https://www.gartner.com/doc/2628418/mobile-database-right- Mobile Databases by Niloofar Banivaheb.

[14] https://classes.soe.ucsc.edu/bme110/Spring09/BME110-Lect1-Mar31.pdf.