Unit 1

Introduction to Big Data Analytics

Big Data Overview, State of the Practice in Analytics, Data Analytics Lifecycle, Data Analytics Problem. Understanding the features of R language, Understanding different Hadoop modes, Understanding Hadoop features, The HDFS and Map Reduce architecture.

What is Big Data?

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.

What Comes Under Big Data?

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 Facebook 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.

Benefits of Big Data

Big data is really critical to our life and its emerging as one of the most important technologies in modern world. Follow are just few benefits which are very much known to all of us:

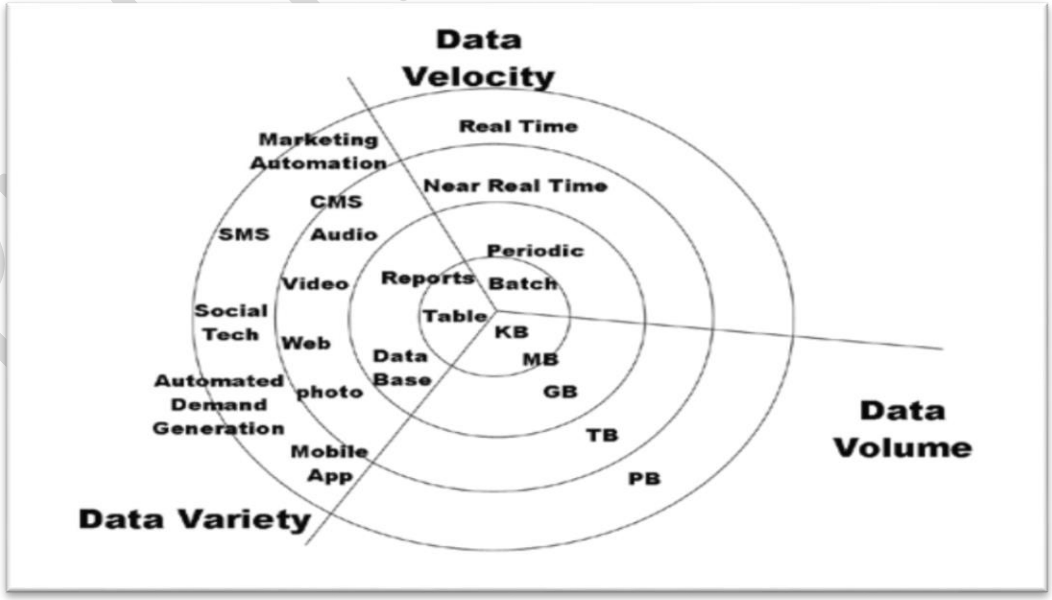
- Using the information kept in the social network like Facebook, the marketing agencies are learning about the response for their campaigns, promotions, and other advertising mediums.
- Using the information in the social media like preferences and product perception of their consumers, product companies and retail organizations are planning their production.
- Using the data regarding the previous medical history of patients, hospitals are providing better and quick service.

Characteristics of Big Data

Big data has to deal with large and complex datasets that can be structured, semi-Structured and unstructured and will typically not fit into memory to be processed.

The main characteristics of big data are

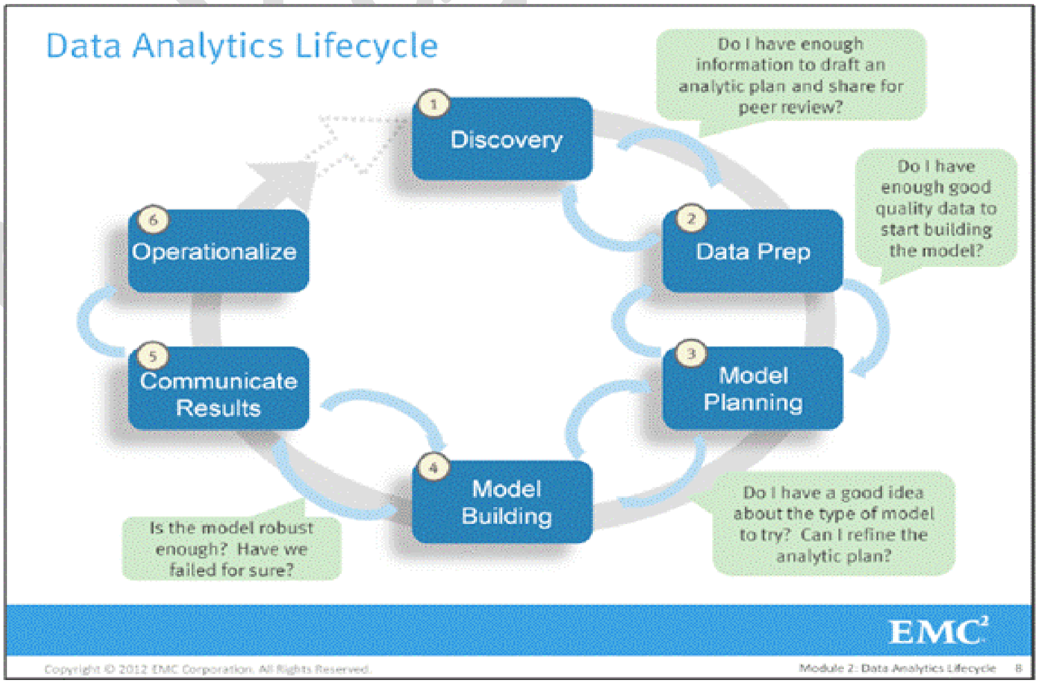
1) Velocity
2) Volume
3) Variety
4) Variability
5) Complexity



- **Volume.** Many factors contribute to the increase in data volume. Transaction-based data stored through the years. Unstructured data streaming in from social media. Increasing amounts of sensor and machine-to-machine data being collected.
- **Velocity.** Data is streaming in at unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time. Reacting quickly enough to deal with data velocity is a challenge for most organizations.
- **Variety.** Data today comes in all types of formats. Structured, numeric data in traditional databases. Information created from line-of-business applications. Unstructured text documents, email, video, audio, stock ticker data and financial transactions. Managing, merging and governing different varieties of data is something many organizations still grapple with.
- **Variability.** In addition to the increasing velocities and varieties of data, data flows can be highly inconsistent with periodic peaks. Even more so with unstructured data involved.
- **Complexity.** Today's data comes from multiple sources. And it is still an undertaking to link, match, cleanse and transform data across systems. However, it is necessary to connect and correlate relationships, hierarchies and multiple data linkages or your data can quickly spiral out of control.

Data Analytics Life Cycle

Data analytics (DA) is the science of examining raw **data** with the purpose of drawing conclusions about that information. **Data analytics** is used in many industries to allow companies and organization to make better business decisions and in the sciences to verify or disprove existing models or theories.



Data Analytics Lifecycle

Do I have enough information to draft an analytic plan and share for peer review?

Do I have enough good quality data to start building the model?

Do I have a good idea about the type of model to try? Can I refine the analytic plan?

Is the model robust enough? Have we failed for sure?

Copyright © 2012 EMC Corporation. All Rights Reserved.

Module 2: Data Analytics Lifecycle 8

Phase 1−Discovery: In Phase 1, the team learns the business domain, including relevant history such as whether the organization or business unit has attempted similar projects in the past from which they can learn. The team assesses the resources available to support the project in terms of people, technology, time, and data. Important activities in this phase include framing the business problem as an analytics challenge that can be addressed in subsequent phases and formulating initial hypotheses (IHs) to test and begin learning the data.

Phase 2−Data preparation: Phase 2 requires the presence of an analytic sandbox, in which the team can work with data and perform analytics for the duration of the project. The team needs to execute extract, load, and transform (ELT) or extract, transform and load (ETL) to get data into the sandbox. The ELT and ETL are sometimes abbreviated as ETLT. Data should be transformed in the ETLT process so the team can work with it and analyze it. In this phase, the team also needs to familiarize itself with the data thoroughly and take steps to condition the data

Phase 3−Model planning: Phase 3 is model planning, where the team determines the methods, techniques, and workflow it intends to follow for the subsequent model building phase. The team explores the data to learn about the relationships between variables and subsequently selects key variables and the most suitable models.

Phase 4−Model building: In Phase 4, the team develops datasets for testing, training, and production purposes. In addition, in this phase the team builds and executes models based on the work done in the model planning phase. The team also considers whether its existing tools will suffice for running the models, or if it will need a more robust environment for executing models and workflows (for example, fast hardware and parallel processing, if applicable).

Phase 5−Communicate results: In Phase 5, the team, in collaboration with major stakeholders, determines if the results of the project are a success or a failure based on the criteria developed in Phase 1. The team should identify key findings, quantify the business value, and develop a narrative to summarize and convey findings to stakeholders.

Phase 6−Operationalize: In Phase 6, the team delivers final reports, briefings, code, and technical documents. In addition, the team may run a pilot project to implement the models in a production environment.

Understanding the feature of R Language

Basic of R Language

R is an open source software package to perform statistical analysis on data. R is a programming language used by data scientist statisticians and others who need to make statistical analysis of data and glean key insights from data using mechanisms, such as regression, clustering, classification, and text analysis. R is registered under **GNU** (**General Public License**).

R provides a wide variety of statistical, machine learning (linear and nonlinear modeling, classic statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. R has various built-in as well as extended functions for statistical, machine learning, and visualization tasks such as:

- Data extraction
- Data cleaning
- Data loading
- Data transformation
- Statistical analysis
- Predictive modeling
- Data visualization

R allows performing Data analytics by various statistical and machine learning operations as follows:

- Regression
- Classification
- Clustering
- Recommendation
- Text mining

Understanding features of R

Let's see different useful features of R:

- Effective programming language
- Relational database support
- Data analytics
- Data visualization
- Extension through the vast library of R packages

Understanding the features of R language

There are over 3,000 R packages and the list is growing day by day. It would be beyond the scope of any book to even attempt to explain all these packages. This book focuses only on the key features of R and the most frequently used and popular packages.

Using R packages

R packages are self-contained units of R functionality that can be invoked as functions. A good analogy would be a .jar file in Java. There is a vast library of R packages available for a very wide range of operations ranging from statistical operations and machine learning to rich graphic visualization and plotting. Every package will consist of one or more R functions. An R package is a re-usable entity that can be shared and used by others. R users can install the package that contains the functionality they are looking for and start calling the functions in the package. A comprehensive list of these packages can be found at http://cran.r-project. org/ called Comprehensive R Archive Network (CRAN).

Performing data operations

R enables a wide range of operations. Statistical operations, such as mean, min, max, probability, distribution, and regression. Machine learning operations, such as linear regression, logistic regression, classification, and clustering. Universal data processing operations are as follows:

· Data cleaning: This option is to clean massive datasets

· Data exploration: This option is to explore all the possible values of datasets

· Data analysis: This option is to perform analytics on data with descriptive and predictive analytics data visualization, that is, visualization of analysis output programming

To build an effective analytics application, sometimes we need to use the online Application Programming Interface (API) to dig up the data, analyze it with expedient services, and visualize it by third-party services. Also, to automate the data analysis process, programming will be the most useful feature to deal with. R has its own programming language to operate data. Also, the available package can help to integrate R with other programming features. R supports object-oriented programming concepts. It is also capable of integrating with other programming languages, such as Java, PHP, C, and C++. There are several packages that will act as middle-layer programming features to aid in data analytics, which are similar to sqldf, httr, RMongo, RgoogleMaps, RGoogle Analytics, and google-predictionapi-r-client.

Increasing community support

As the number of R users are escalating, the groups related to R are also increasing. So, R learners or developers can easily connect and get their uncertainty solved with the help of several R groups or communities.

The following are many popular sources that can be found useful:

· R mailing list: This is an official R group created by R project owners.

· R blogs: R has countless bloggers who are writing on several R applications. One of the most popular blog websites is http://www.r-bloggers.com/where all the bloggers contribute their blogs.

· Stack overflow: This is a great technical knowledge sharing platform where the programmers can post their technical queries and enthusiast programmers suggest a solution. For more information, visit http://stats.stackexchange.com/.

· Groups: There are many other groups existing on LinkedIn and Meetup where professionals across the world meet to discuss their problems and innovative ideas.

· Books: There are also lot of books about R. Some of the popular books are R in Action, by Rob Kabacoff, Manning Publications, R in a Nutshell, by Joseph Adler, O'Reilly Media, R and Data Mining, by Yanchang Zhao, Academic Press, and R Graphs Cookbook, by Hrishi Mittal, Packt Publishing.

Performing data modeling in R

Data modeling is a machine learning technique to identify the hidden pattern from the historical dataset, and this pattern will help in future value prediction over the same data. This techniques highly focus on past user actions and learns their taste. Most of these data modeling techniques have been adopted by many popular organizations to understand the behavior of their customers based on their past transactions. These techniques will analyze data and predict for the customers what they are looking for. Amazon, Google, Facebook, eBay, LinkedIn, Twitter, and many other organizations are using data mining for changing the definition applications.

The most common data mining techniques are as follows:

· **Regression:** In statistics, regression is a classic technique to identify the scalar relationship between two or more variables by fitting the state line on the variable values. That relationship will help to predict the variable value for future events. For example, any variable y can be modeled as linear function of another variable x with the formula y = mx+c. Here, x is the predictor variable, y is the response variable, m is slope of the line, and c is the intercept. Sales forecasting of products or services and predicting the price of stocks can be achieved through this regression. R provides this regression feature via the lm method, which is by default present in R.

· **Classification:** This is a machine-learning technique used for labeling the set of observations provided for training examples. With this, we can classify the observations into one or more labels. The likelihood of sales, online fraud detection, and cancer classification (for medical science) are common applications of classification problems. Google Mail uses this technique to classify e-mails as spam or not. Classification features can be served by glm, glmnet, ksvm, svm, and random Forest in R.

· **Clustering:** This technique is all about organizing similar items into groups from the given collection of items. User segmentation and image compression are the most common applications of clustering. Market segmentation, social network analysis, organizing the computer clustering, and astronomical data analysis are applications of clustering. Google News uses these techniques to group similar news items into the same category. Clustering can be achieved through the knn, kmeans, dist, pvclust, and Mclust methods in R.

Recommendation: The recommendation algorithms are used in recommender systems where these systems are the most immediately recognizable machine learning techniques in use today. Web content recommendations may include similar websites, blogs, videos, or related content. Also, recommendation of online items can be helpful for cross-selling and up-selling. We have all seen online shopping portals that attempt to recommend books, mobiles, or any items that can be sold on the Web based on the user's past behavior. Amazon is a well-known e-commerce portal that generates 29 percent of sales through recommendation systems. Recommender systems can be implemented via Recommender () with the recommender lab package in R.

Understanding different Hadoop modes

Hadoop is used with three different modes:

- **The standalone mode**: In this mode, you do not need to start any Hadoop daemons. Instead, just call ~/Hadoop-directory/bin/Hadoop that will execute a Hadoop operation as a single Java process. This is recommended for testing purposes. This is the default mode and you don't need to configure anything else. All daemons, such as Name Node, Data Node, Job Tracker, and Task Tracker run in a single Java process.

- **The pseudo mode**: In this mode, you configure Hadoop for all the nodes. A separate **Java Virtual Machine** (**JVM**) is spawned for each of the Hadoop components or daemons like mini cluster on a single host.

- **The full distributed mode**: In this mode, Hadoop is distributed across multiple machines. Dedicated hosts are configured for Hadoop components. Therefore, separate JVM processes are present for all daemons.

Understanding Hadoop features

Hadoop is specially designed for two core concepts: HDFS and Map Reduce. Both are related to distributed computation. Map Reduce is believed as the heart of Hadoop that performs parallel processing over distributed data.

Let us see more details on Hadoop's features:

- HDFS
- Map Reduce

HDFS

HDFS is Hadoop's own rack-aware file system, which is a UNIX-based data storage layer of Hadoop. HDFS is derived from concepts of Google file system. An important characteristic of Hadoop is the partitioning of data and computation across many (thousands of) hosts, and the execution of application computations in parallel, close to their data. On HDFS, data files are replicated as sequences of blocks in the cluster. A Hadoop cluster scales computation capacity, storage capacity, and I/O bandwidth by simply adding commodity servers. HDFS can be accessed from applications in many different ways

Characteristics of HDFS

The characteristics of HDFS:

- Fault tolerant
- Runs with commodity hardware
- Able to handle large datasets
- Master slave paradigm
- Write once file access only

Map Reduce

MapReduce is a programming model for processing large datasets distributed on a large cluster. MapReduce is the heart of Hadoop. Its programming paradigm allows performing massive data processing across thousands of servers configured with Hadoop clusters. This is derived from Google MapReduce.

Hadoop MapReduce is a software framework for writing applications easily, which process large amounts of data (multiterabyte datasets) in parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. This MapReduce paradigm is divided into two phases, Map and Reduce that mainly deal with key and value pairs of data. The Map and Reduce task run sequentially in a cluster; the output of the Map phase becomes the input for the Reduce phase. These phases are explained as follows:

- **Map phase**: Once divided, datasets are assigned to the task tracker to perform the Map phase. The data functional operation will be performed over the data, emitting the mapped key and value pairs as the output of the Map phase.

- **Reduce phase**: The master node then collects the answers to all the sub problems and combines them in some way to form the output; the answer to the problem it was originally trying to solve.

The five common steps of parallel computing are as follows:

1. Preparing the Map() input: This will take the input data row wise and emit key value pairs per rows, or we can explicitly change as per the requirement.

    Map input: list (k1, v1)

2. Run the user-provided Map() code

    Map output: list (k2, v2)

3. Shuffle the Map output to the Reduce processors. Also, shuffle the similar keys (grouping them) and input them to the same reducer.

4. Run the user-provided Reduce() code: This phase will run the custom reducer code designed by developer to run on shuffled data and emit key and value.

    Reduce input: (k2, list(v2))
    Reduce output: (k3, v3)

5. Produce the final output: Finally, the master node collects all reducer output and combines and writes them in a text file.

HDFS and MapReduce architecture

HDFS and MapReduce are considered to be the two main features of the Hadoop framework.

HDFS architecture

HDFS can be presented as the master/slave architecture. HDFS master is named as NameNode whereas slave as DataNode. NameNode is a sever that manages the filesystem namespace and adjusts the access (open, close, rename, and more) to files by the client. It divides the input data into blocks and announces which data block will be store in which DataNode. DataNode is a slave machine that stores the replicas of the partitioned dataset and serves the data as the request comes. It also performs block creation and deletion. The internal mechanism of HDFS divides the file into one or more blocks; these blocks are stored in a set of data nodes. Under normal circumstances of the replication factor three, the HDFS strategy is to place the first copy on the local node, second copy on the local rack with a different node, and a third copy into different racks with different nodes. As HDFS is designed to support large files, the HDFS block size is defined as 64 MB. If required, this can be increased.

HDFS components

HDFS is managed with the master-slave architecture included with the following components:

- **NameNode**: This is the master of the HDFS system. It maintains the directories, files, and manages the blocks that are present on the DataNodes.
- **DataNode**: These are slaves that are deployed on each machine and provide actual storage. They are responsible for serving read-and-write data requests for the clients.
- **Secondary NameNode**: This is responsible for performing periodic checkpoints. So, if the NameNode fails at any time, it can be replaced with a snapshot image stored by the secondary NameNode checkpoints.

MapReduce architecture

MapReduce is also implemented over master-slave architectures. Classic MapReduce contains job submission, job initialization, task assignment, task execution, progress and status update, and job completion-related activities, which are mainly managed by the JobTracker node and executed by TaskTracker. Client application submits a job to the JobTracker. Then input is divided across the cluster. The JobTracker then calculates the number of map and reducer to be processed. It commands the TaskTracker to start executing the job. Now, the TaskTracker copies the resources to a local machine and launches JVM to map and reduce program over the data. Along with this, the TaskTracker periodically sends update to the JobTracker, which can be considered as the heartbeat that helps to update JobID, job status, and usage of resources.

MapReduce components

MapReduce is managed with master-slave architecture included with the following components:

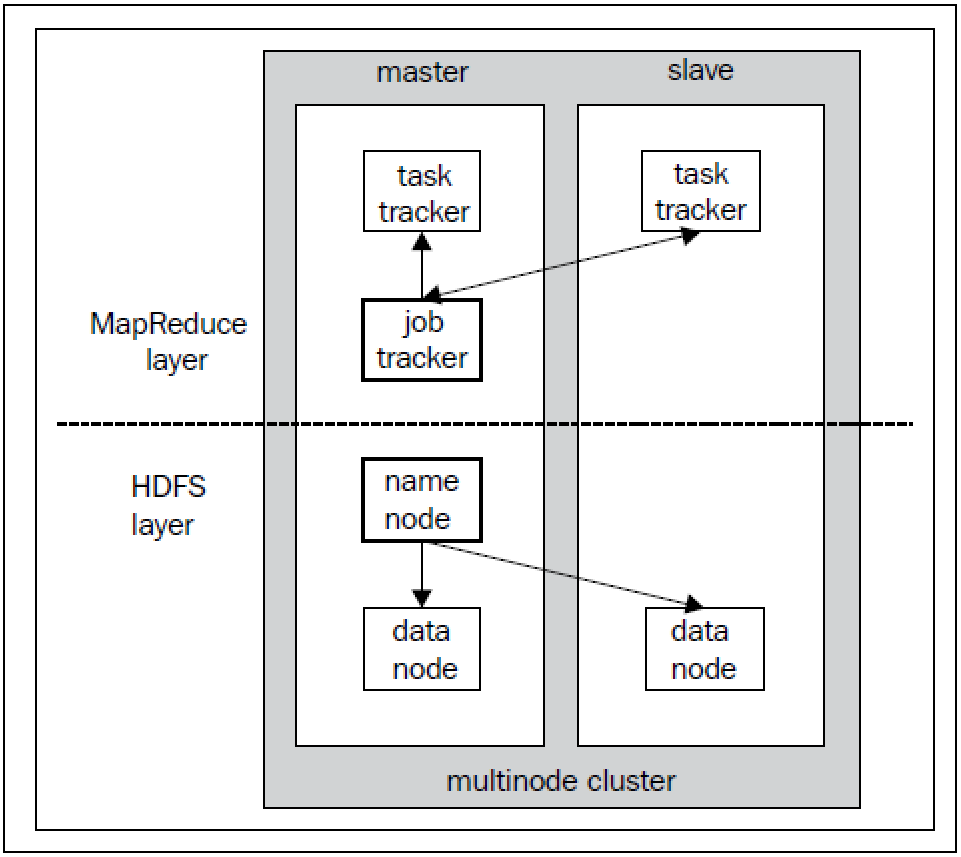
- **JobTracker**: This is the master node of the MapReduce system, which manages the jobs and resources in the cluster (TaskTrackers). The JobTracker tries to schedule each map as close to the actual data being processed on the TaskTracker, which is running on the same DataNode as the underlying block.

- **TaskTracker**: These are the slaves that are deployed on each machine. They are responsible for running the map and reducing tasks as instructed by the JobTracker.

HDFS and MapReduce architecture

Both HDFS and MapReduce master and slave components have been included, where NameNode and DataNode are from HDFS and JobTracker and TaskTracker are from the MapReduce paradigm.

Both paradigms consisting of master and slave candidates have their own specific responsibility to handle MapReduce and HDFS operations. In the next plot, there is a plot with two sections: the preceding one is a MapReduce layer and the following one is an HDFS layer.



The HDFS and MapReduce architecture

Hadoop is a top-level Apache project and is a very complicated Java framework. To avoid technical complications, the Hadoop community has developed a number of Java frameworks that has added an extra value to Hadoop features. They are considered as Hadoop subprojects. Here, we are departing to discuss several Hadoop components that can be considered as an abstraction of HDFS or MapReduce.

Unit 2

Understanding the basic of MapReduce

Syllabus

The Hadoop MapReduce, The Hadoop MapReduce Fundamentals, Writing MapReduce example, learning the different ways to write MapReduce in R. Integrating R and Hadoop –the RHIPE architecture and RHadooop.

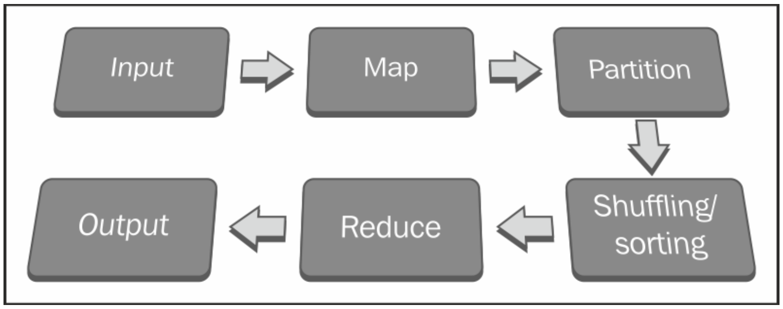
Basics of MapReduce

MapReduce is a programming model that works in a distributed fashion,but it is not the only one that does. It might be illuminating to describe otherprogramming models, for example, MPI and **Bulk Synchronous Parallel** (**BSP**).To process Big Data with tools such as R and several machine learning techniquesrequires a high-configuration machine, but that's not the permanent solution. So,distributed processing is the key to handling this data. This distributed computationcan be implemented with the MapReduce programming model.

MapReduce is the one that answers the Big Data question. Logically, to process datawe need parallel processing, which means processing over large computation; it caneither be obtained by clustering the computers or increasing the configuration of themachine. Using the computer cluster is an ideal way to process data with a large size.

MapReduce implementation runs on large clusters withcommodity hardware. This data processing platform is easier for programmers toperform various operations. The system takes care of input data, distributes dataacross the computer network, processes it in parallel, and finally combines its outputinto a single file to be aggregated later. This is very helpful in terms of cost and isalso a time-saving system for processing large datasets over the cluster. Also, it willefficiently use computer resources to perform analytics over huge data. Google hasbeen granted a patent on MapReduce.

For MapReduce, programmers need to just design/migrate applications into twophases: Map and Reduce. They simply have to design Map functions for processinga key-value pair to generate a set of intermediate key-value pairs, and Reducefunctions to merge all the intermediate keys. Both the Map and Reduce functionsmaintain MapReduce workflow. The Reduce function will start executing the codeafter completion or once the Map output is available to it.



Their execution sequence can be seen as follows:
MapReduce assumes that the Maps are independent and will execute them inparallel. The key aspect of the MapReduce algorithm is that if every Map and Reduceis independent of all other ongoing Maps and Reduces in the network, the operationwill run in parallel on different keys and lists of data.

A distributed filesystem spreads multiple copies of data across different machines.This offers reliability as well as fault tolerance. If a machine with one copy of the filecrashes, the same data will be provided from another replicated data source.

The master node of the MapReduce daemon will take care of all the responsibilitiesof the MapReduce jobs, such as the execution of jobs, the scheduling of Mappers,Reducers, Combiners, and Partitioners, the monitoring of successes as well asfailures of individual job tasks, and finally, the completion of the batch job.

Hadoop MapReduce

The MapReduce model can be implemented in several languages, butapart from that, Hadoop MapReduce is a popular Java framework for easily writtenapplications. It processes vast amounts of data (multiterabyte datasets) in parallel onlarge clusters (thousands of nodes) of commodity hardware in a reliable and fault tolerantmanner. This MapReduce paradigm is divided into two phases, Map andReduce, which mainly deal with key-value pairs of data. The Map and Reduce tasksrun sequentially in a cluster, and the output of the Map phase becomes the input ofthe Reduce phase.

All data input elements in MapReduce cannot be updated. If the input (key,value) pairs for mapping tasks are changed, it will not be reflected in the input files.The Mapper output will be piped to the appropriate Reducer grouped with the keyattribute as input. This sequential data process will be carried away in a parallelmanner with the help of Hadoop MapReduce algorithms as well as Hadoop clusters.

MapReduce programs transform the input dataset present in the list format intooutput data that will also be in the list format. This logical list translation processis mostly repeated twice in the Map and Reduce phases. We can also handle theserepetitions by fixing the number of Mappers and Reducers.

MapReduce entities

The following are the components of Hadoop that are responsible for performinganalytics over Big Data:

- **Client**: This initializes the job
- **JobTracker**: This monitors the job
- **TaskTracker**: This executes the job

- **HDFS**: This stores the input and output data

The Hadoop MapReduce scenario

The four main stages of Hadoop MapReduce data processing are as follows:

- The loading of data into HDFS
- The execution of the Map phase
- Shuffling and sorting
- The execution of the Reduce phase

1. Loading data into HDFS

The input dataset needs to be uploaded to the Hadoop directory so it can be used byMapReduce nodes. Then, **Hadoop Distributed File System** (**HDFS**) will divide theinput dataset into data splits and store them to DataNodes in a cluster by taking careof the replication factor for fault tolerance. All the data splits will be processed byTaskTracker for the Map and Reduce tasks in a parallel manner.

There are some alternative ways to get the dataset in HDFS withHadoop components:

- **Sqoop**: This is an open source tool designed for efficiently transferring bulk data between Apache Hadoop and structured, relational databases. Suppose your application has already been configured with the MySQL database and you want to use the same data for performing data analytics, Sqoop is recommended for importing datasets to HDFS. Also, after the completion of the data analytics process, the output can be exported to the MySQL database.

- **Flume**: This is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data to HDFS. Flume is able to read data from most sources, such as logfiles, sys logs, and the standard output of the Unix process.

Using the preceding data collection and moving the framework can make this datatransfer process very easy for the MapReduce application for data analytics

2. Executing the Map phase

Executing the client application starts the Hadoop MapReduce processes. The Mapphase then copies the job resources (unjarred class files) and stores it to HDFS, andrequests JobTracker to execute the job. The JobTracker initializes the job, retrieves theinput, splits the information, and creates a Map task for each job.

The JobTracker will call TaskTracker to run the Map task over the assigned inputdata subset. The Map task reads this input split data as input (key, value) pairsprovided to the Mapper method, which then produces intermediate (key, value)pairs. There will be at least one output for each input (key, value) pair.

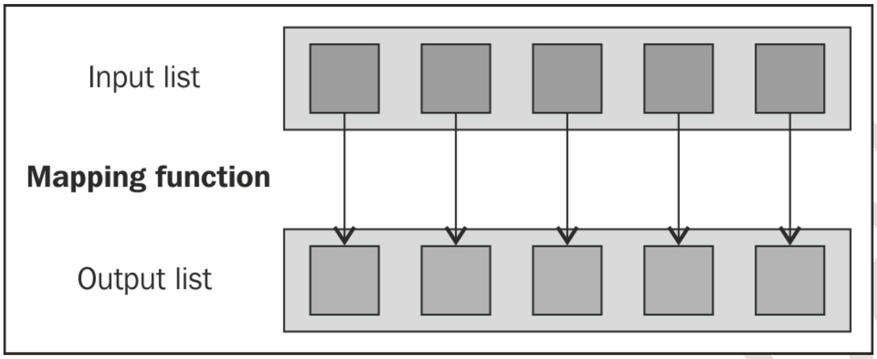


Fig: Mapping individual elements of an input list

The list of (key, value) pairs is generated such that the key attribute will be repeatedmany times. So, its key attribute will be re-used in the Reducer for aggregatingvalues in MapReduce. As far as format is concerned, Mapper output format valuesand Reducer input values must be the same.

After the completion of this Map operation, the TaskTracker will keep theresult in its buffer storage and local disk space (if the output data size is morethan the threshold).

3. Shuffling and sorting

To optimize the MapReduce program, this intermediate phase is very important.As soon as the Mapper output from the Map phase is available, this intermediatephase will be called automatically. After the completion of the Map phase, all theemitted intermediate (key, value) pairs will be partitioned by a Partitioner at theMapper side, only if the Partitioner is present. The output of the Partitioner will besorted out based on the key attribute at the Mapper side. Output from sorting theoperation is stored on buffer memory available at the Mapper node, TaskTracker.

The Combiner is often the Reducer itself. So by compression, it's not **Gzip** or somesimilar compression but the Reducer on the node that the map is outputting thedata on. The data returned by the Combiner is then shuffled and sent to the reducednodes. To speed up data transmission of the Mapper output to the Reducer slot atTaskTracker, you need to compress that output with the Combiner function. Bydefault, the Mapper output will be stored to buffer memory, and if the output sizeis larger than threshold, it will be stored to a local disk. This output data will beavailable through **Hypertext Transfer Protocol** (**HTTP**).

4. Reducing phase execution

As soon as the Mapper output is available, TaskTracker in the Reducer node willretrieve the available partitioned Map's output data, and they will be groupedtogether and merged into one large file, which will then be assigned to a processwith a Reducer method. Finally, this will be sorted out before data is provided to theReducer method.

The Reducer method receives a list of input values from an input (key, list(value)) and aggregates them based on custom logic, and produces the output(key, value) pairs.

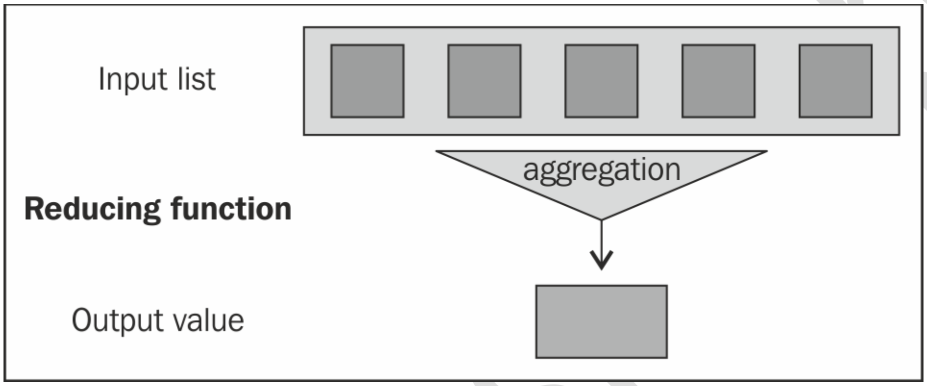


Fig: Reducing input values to an aggregate value as output

Limitations of MapReduce

- The MapReduce framework is notoriously difficult to leverage for transformational logic that is not as simple, for example, real-time streaming, graph processing, and message passing.
- Data querying is inefficient over distributed, unindexed data than in a database created with indexed data. However, if the index over the data is generated, it needs to be maintained when the data is removed or added.
- We can't parallelize the Reduce task to the Map task to reduce the overall processing time because Reduce tasks do not start until the output of the
- A map task is available to it. (The Reducer's input is fully dependent on the Mapper's output.) Also, we can't control the sequence of the execution of the Map and Reduce task. But sometimes, based on application logic, we can definitely configure a slow start for the Reduce tasks at the instance when the data collection starts as soon as the Map tasks complete.
- Long-running Reduce tasks can't be completed because of their poor resource utilization either if the Reduce task is taking too much time to complete and fails or if there are no otherReduce slots available for rescheduling it (thiscan be solved with YARN).

Hadoop MapReduce fundamentals

To understand Hadoop MapReduce fundamentals properly

- Understand MapReduce objects
- Learn how to decide the number of Maps in MapReduce
- Learn how to decide the number of Reduces in MapReduce
- Understand MapReduce dataflow
- Take a closer look at Hadoop MapReduce terminologies

1. Understanding MapReduce objects

MapReduce operations in Hadoop are carried out mainly by threeobjects: **Mapper**, **Reducer**, and **Driver**.

- **Mapper**: This is designed for the Map phase of MapReduce, which starts MapReduce operations by carrying input files and splitting them into several pieces. For each piece, it will emit a key-value data pair as the output value.

- **Reducer**: This is designed for the Reduce phase of a MapReduce job; it accepts key-based grouped data from the Mapper output, reduces it by aggregation logic, and emits the (key, value) pair for the group of values.

- **Driver**: This is the main file that drives the MapReduce process. It starts the execution of MapReduce tasks after getting a request from the client application with parameters. The Driver file is responsible for building the configuration of a job and submitting it to the Hadoop cluster. The Driver code will contain the main() method that accepts arguments from the command line. The input and output directory of the Hadoop MapReduce job will be accepted by this program. Driver is the main file for defining job configuration details, such as the job name, job input format, job output format, and the Mapper, Combiner, Partitioner, and Reducer classes. MapReduce is initialized by calling this main() function of the Driver class.

2. Deciding the number of Maps in MapReduce

The number of Maps is usually defined by the size of the input data and size of thedata split block that is calculated by the size of the HDFS file / data split. Therefore,if we have an HDFS datafile of 5 TB and a block size of 128 MB, there will be 40,960 bmaps present in the file. But sometimes, the number of Mappers created will be morethan this count because of speculative execution. This is true when the input is a file,though it entirely depends on the InputFormat class.

In Hadoop MapReduce processing, there will be a delay in the result of the job whenthe assigned Mapper or Reducer is taking a long time to finish. If you want to avoid this, speculative execution in Hadoop can run multiple copies of the same Map orReduce task on different nodes, and the result from the first completed nodes can beused.

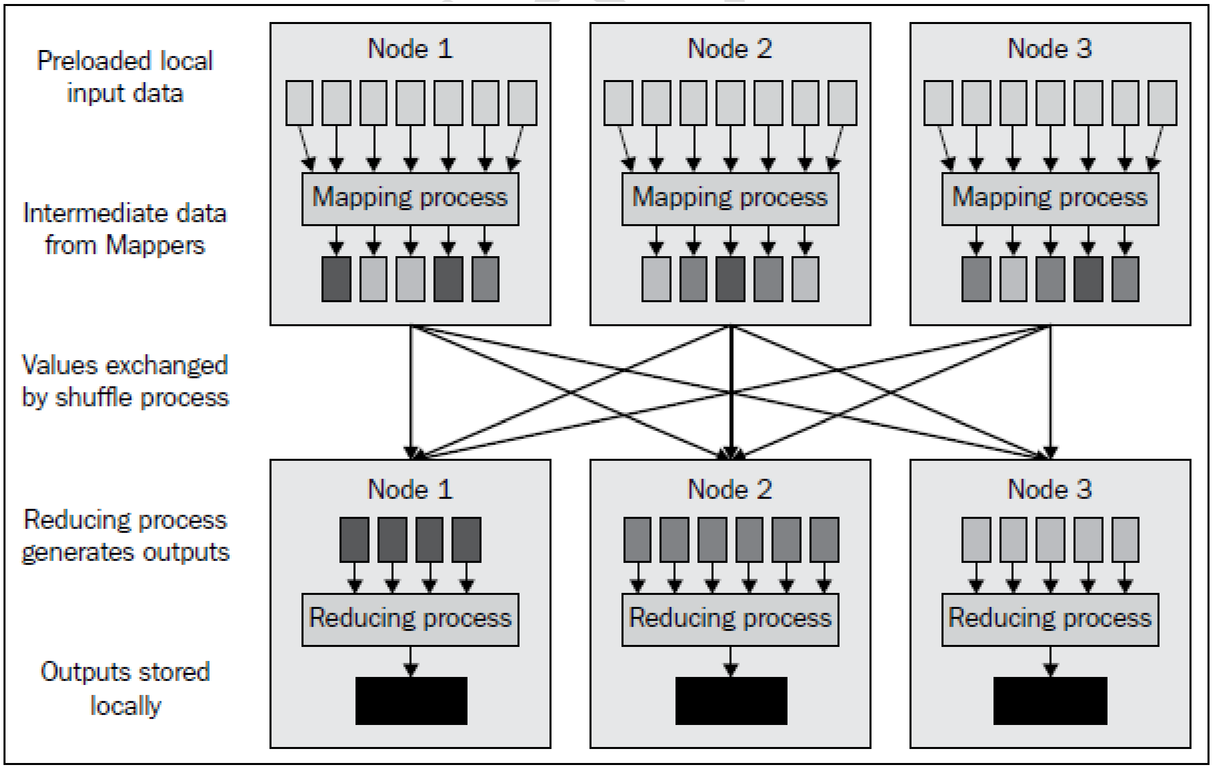
3. Deciding the number of Reducers in MapReduce

A numbers of Reducers are created based on the Mapper's input. However, if youhardcode the number of Reducers in MapReduce, it won't matter how many nodesare present in a cluster. It will be executed as specified in the configuration.

Additionally, we can set the number of Reducers at runtime along withthe MapReduce command at the command prompt -D mapred.reduce.tasks, with the number you want. Programmatically, it can be set via conf.setNumReduceTasks(int).

4. Understanding MapReduce dataflow

From the following diagram, we will understand MapReduce dataflow with multiple nodes ina Hadoop cluster:



MapReduce dataflow

Hadoop data processing includes several tasks that help achieve the final outputfrom an input dataset. These tasks are as follows:

1. Preloading data in HDFS.
2. Running MapReduce by calling Driver.
3. Reading of input data by the Mappers, which results in the splitting of the data execution of the Mapper custom logic and the generation of intermediate key-value pairs.
4. Executing Combiner and the shuffle phase to optimize the overall HadoopMapReduce process.
5. Sorting and providing of intermediate key-value pairs to the Reduce phase.The Reduce phase is then executed. Reducers take these partitioned keyvaluepairs and aggregate them based on Reducer logic.
6. The final output data is stored at HDFS.

Here, Map and Reduce tasks can be defined for several data operations as follows:

- Data extraction
- Data loading
- Data segmentation
- Data cleaning
- Data transformation
- Data integration

Writing a Hadoop MapReduce example

Now we will move forward with MapReduce by learning a very common and easy example of word count. The goal of this example is to calculate how many times each word occurs in the provided documents. These documents can be considered as input to MapReduce's file.

In this example, we already have a set of text files–we want to identify the frequency of all the unique words existing in the files. We will get this by designing the Hadoop MapReduce phase.

In this section, we will see more on Hadoop MapReduce programming using Hadoop MapReduce's old API. Here we assume that the reader has already set up the Hadoop environment as described in Chapter 1, Getting Ready to Use R and Hadoop. Also, keep in mind that we are not going to use R to count words; only Hadoop will be used here.

Basically, Hadoop MapReduce has three main objects: Mapper, Reducer, and Driver.They can be developed with three Java classes; they are the Map class, Reduce class,and Driver class, where the Map class denotes the Map phase, the Reducer class denotes the Reduce phase, and the Driver class denotes the class with the main() method to initialize the Hadoop MapReduce program.

In the previous section of Hadoop MapReduce fundamentals, we already discussed what Mapper, Reducer, and Driver are. Now, we will learn how to define them and program for them in Java. In upcoming chapters, we will be learning to do more with a combination of R and Hadoop.

Understanding the steps to run a MapReduce job

Let's see the steps to run a MapReduce job with Hadoop:

1. In the initial steps of preparing Java classes, we need you to develop a Hadoop MapReduce program as per the definition of our business problem.

In this example, we have considered a word count problem. So, we have developed three Java classes for the MapReduce program; they are Map.

java, Reduce.java, and WordCount.java, used for calculating the frequency of the word in the provided text files.

◦ ◦ Map.java: This is the Map class for the word count Mapper.

```
// Defining package of the class

package com.PACKT.chapter1;

// Importing java libraries

import java.io.*;

importjava.util.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapred.*;

// Defining the Map class
```

```
public class Map extends MapReduceBase implements

Mapper<LongWritable,

Text,

Text,

IntWritable>{

//Defining the map method – for processing the data with //

problem specific logic

public void map(LongWritable key,

Text value,

OutputCollector<Text,

IntWritable> output,
Reporter reporter)
throws IOException {
// For breaking the string to tokens and convert them to
lowercase
StringTokenizer st = new StringTokenizer(value.toString().
toLowerCase());
// For every string tokens
while(st.hasMoreTokens()) {
// Emitting the (key,value) pair with value 1.
output.collect(new Text(st.nextToken()),
new IntWritable(1));
}
}
```

}

Different ways to write Hadoop MapReduce in R

Hadoop Big Data processing with MapReduce is a big deal forstatisticians, web analysts, and product managers who used to use the R toolfor analyses because supplementary programming knowledge of MapReduce isrequired to migrate the analyses into MapReduce with Hadoop. R is a tool that is consistently increasing in popularity; there are many packages/libraries that are being developed for integrating with R. So to develop a MapReducealgorithm or program that runs with the log of R and computation power of Hadoop,we require the middleware for R and Hadoop. **RHadoop**, **RHIPE**, and **Hadoop streaming** are the middleware that help develop and execute Hadoop MapReducewithin R.

1. Learning RHadoop

RHadoop is a great open source software framework of R for performing dataanalytics with the Hadoop platform via R functions. RHadoop has been developedby **Revolution Analytics**, which is the leading commercial provider of software andservices based on the open source R project for statistical computing. The RHadoopproject has three different R packages: rhdfs, rmr, and rhbase. All these packagesare implemented and tested on the Cloudera Hadoop distributions CDH3, CDH4,and R 2.15.0.

These three different R packages have been designed on Hadoop's two main featuresHDFS and MapReduce:

- **rhdfs:** This is an R package for providing all Hadoop HDFS access to R. All distributed files can be managed with R functions.

- **rmr:** This is an R package for providing Hadoop MapReduce interfaces to R. With the help of this package, the Mapper and Reducer can easily be developed.

- **rhbase:** This is an R package for handling data at HBase distributed database through R.

2. Learning RHIPE

R and Hadoop Integrated Programming Environment (**RHIPE**) is a free andopen source project. RHIPE is widely used for performing Big Data analysis with**D&R** analysis. D&R analysis is used to divide huge data, process it in parallel ona distributed network to produce intermediate output, and finally recombine allthis intermediate output into a set. RHIPE is designed to carry out D&R analysis oncomplex Big Data in R on the Hadoop platform. RHIPE was developed by *SaptarshiJoy*

Guha (Data Analyst at Mozilla Corporation) and her team as part of her PhDthesis in the Purdue Statistics Department.

3. Learning Hadoop streaming

Hadoop streaming is a utility that comes with the Hadoop distribution. This utilityallows you to create and run MapReduce jobs with any executable or script as theMapper and/or Reducer. This is supported by R, Python, Ruby, Bash, Perl, and soon. We will use the R language with a bash script.

Also, there is one R package named HadoopStreaming that has been developedfor performing data analysis on Hadoop clusters with the help of R scripts, whichis an interface to Hadoop streaming with R. Additionally, it also allows the runningof MapReduce tasks without Hadoop. This package was developed by *DavidRosenberg*, Chief Scientist at SenseNetworks. He has expertise in machine learningand statistical modeling.

The architecture of RHIPE

The working of the RHIPE library package developed to integrate Rand Hadoop for effective Big Data analytics.

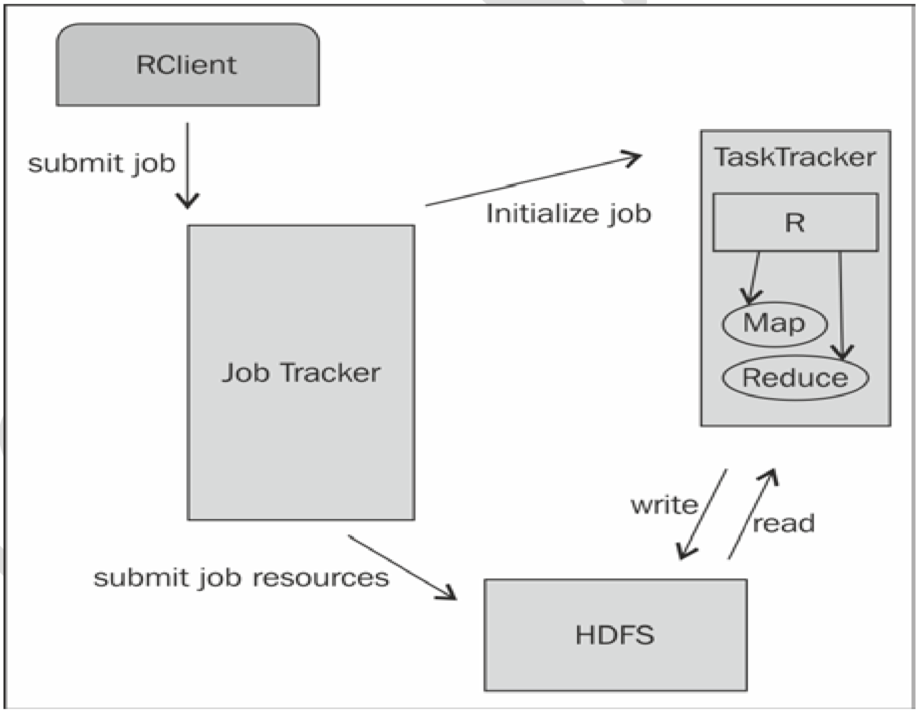


Fig: Components of RHIPE

There are a number of Hadoop components that will be used for data analyticsoperations with R and Hadoop.

The components of RHIPE are as follows:

- **RClient**: RClient is an R application that calls the **JobTracker** to execute the job with an indication of several MapReduce job resources such as Mapper, Reducer, input format, output format, input file, output file, and other several parameters that can handle the MapReduce jobs with RClient.

- **JobTracker**: A JobTracker is the master node of the Hadoop MapReduce operations for initializing and monitoring the MapReduce jobs over the Hadoop cluster.

- **TaskTracker**: TaskTracker is a slave node in the Hadoop cluster. It executes the MapReduce jobs as per the orders given by JobTracker, retrieve the input data chunks, and run R-specific Mapper and Reducer over it. Finally, the output will be written on the HDFS directory.

- **HDFS**: HDFS is a filesystem distributed over Hadoop clusters with several data nodes. It provides data services for various data operations.

RHadoop

RHadoop is a collection of three R packages for providing large data operations withan R environment. It was developed by Revolution Analytics, which is the leadingcommercial provider of software based on R. RHadoop is available with three mainR packages: rhdfs, rmr, and rhbase. Each of them offers different Hadoop features.

- **rhdf**s is an R interface for providing the HDFS usability from the R console. As Hadoop MapReduce programs write their output on HDFS, it is very easy to access them by calling the rhdfs methods. The R programmer can easily perform read and write operations on distributed data files. Basically, rhdfs package calls the HDFS API in backend to operate data sources stored on HDFS.

- **rmr** is an R interface for providing Hadoop MapReduce facility inside the R environment. So, the R programmer needs to just divide their application logic into the map and reduce phases and submit it with the rmr methods. After that, rmr calls the Hadoop streaming MapReduce API with several job parameters as input directory, output directory, mapper, reducer, and so on, to perform the R MapReduce job over Hadoop cluster.

- **rhbase** is an R interface for operating the Hadoop HBase data source stored at the distributed network via a Thrift server. The rhbase package is designed with several methods for initialization and read/write and table manipulation operations.

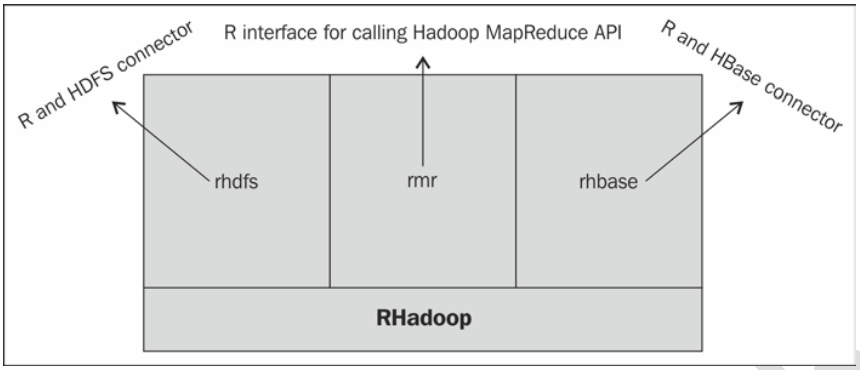


Fig: RHadoop Ecosystem

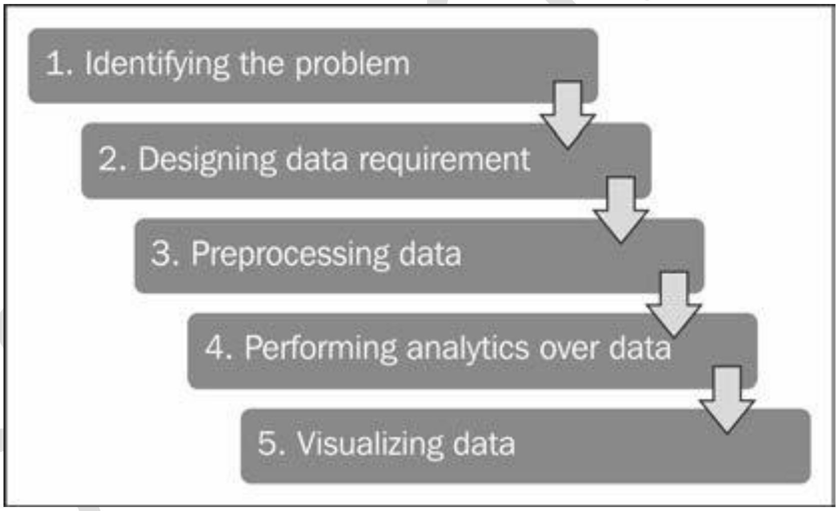
Unit 3
(Learning Data Analytics with R and Hadoop)

The data analytics project cycle, the data analytics problems (web page categorization, stock market change), supervised and unsupervised machine – learning algorithms.

Data analytics project life cycle

While dealing with the data analytics projects, there are some fixed tasks that shouldbe followed to get the expected output. So here we are going to build a data analyticsproject cycle, which will be a set of standard data-driven processes to lead data toinsights effectively. The defined data analytics processes of a project life cycle shouldbe followed by sequences for effectively achieving the goal using input datasets.This data analytics process may include identifying the data analytics problems,designing, and collecting datasets, data analytics, and data visualization.

The data analytics project life cycle stages are seen in the following diagram:


1. Identifying the problem
2. Designing data requirement
3. Preprocessing data
4. Performing analytics over data
5. Visualizing data

1. Identifying the problem

Business analytics trends change by performing data analytics over webdatasets for growing business. Since their data size is increasing gradually dayby day, their analytical application needs to be scalable for collecting insightsfrom their datasets.

With the help of web analytics, we can solve the business analytics problems. Let'sassume that we have a large e-commerce website, and we want to know howto increase the business. We can identify the important pages of our website bycategorizing them as per popularity into high, medium, and low. Based on thesepopular pages, their types, their traffic sources, and their content, we will be able todecide the roadmap to improve business by improving web traffic, as well as content.

2. Designing data requirement

To perform the data analytics for a specific problem, it needs datasets fromrelated domains. Based on the domain and problem specification, the data sourcecan be decided and based on the problem definition; the data attributes of thesedatasets can be defined.

For example, if we are going to perform social media analytics (problemspecification), we use the data source as Facebook or Twitter. For identifying the usercharacteristics, we need user profile information, likes, and posts as data attributes.

3. Preprocessing data

In data analytics, we do not use the same data sources, data attributes, data tools, and algorithms all the time as all of them will not use data in the same format. This leads to the performance of data operations, such as data cleansing, data aggregation, data augmentation, data sorting, and data formatting, to provide the data in a supported format to all the data tools as well as algorithms that will be used in the data analytics.

In simple terms, preprocessing is used to perform data operation to translate data into a fixed data format before providing data to algorithms or tools. The data analytics process will then be initiated with this formatted data as the input.

In case of Big Data, the datasets need to be formatted and uploaded to **HadoopDistributed File System** (**HDFS**) and used further by various nodes with Mappers and Reducers in Hadoop clusters.

4. Performing analytics over data

After data is available in the required format for data analytics algorithms, data analytics operations will be performed. The data analytics operations are performed for discovering meaningful information from data to take better decisions towards business with data mining concepts. It may either use descriptive or predictive analytics for business intelligence.

Analytics can be performed with various machine learning as well as custom algorithmic concepts, such as regression, classification, clustering, and model-based recommendation. For Big Data, the same algorithms can be translated to MapReduce algorithms for running them on Hadoop clusters by translating their data analytics logic to the MapReduce job which is to be run over Hadoop clusters. These models need to be further evaluated as well as improved by various evaluation stages of machine learning concepts. Improved or optimized algorithms can provide better insights.

5. Visualizing data

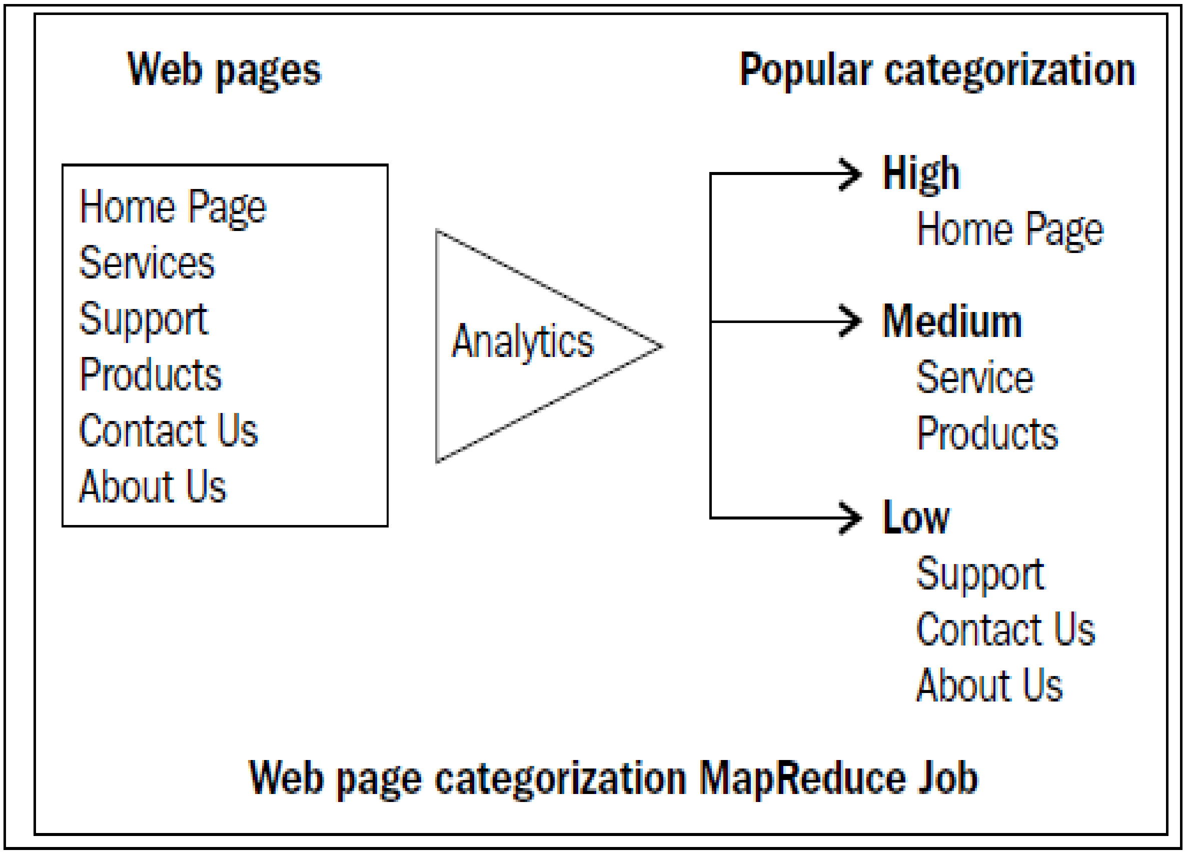
Data visualization is used for displaying the output of data analytics. Visualization is an interactive way to represent the data insights. This can be done with various data visualization softwares as well as R packages. R has a variety of packages for the visualization of datasets. They are as follows:

- **ggplot2:** This is an implementation of the Grammar of Graphics by *r. Hadley Wickham* (http://had.co.nz/).
- **rCharts:** This is an R package to create, customize, and publish interactive JavaScript visualizations from R by using a familiar lattice-style plotting interface by *Markus Gesmann* and *Diego de Castillo*.

Understanding data analytics problems

Exploring web pages categorization

This data analytics problem is designed to identify the category of a web page of awebsite, which may categorized popularity wise as high, medium, or low (regular),based on the visit count of the pages.



Identifying the problem

As this is a web analytics problem, the goal of the problem is to identify theimportance of web pages designed for websites. Based on this information, thecontent, design, or visits of the lower popular pages can be improved or increased.

Designing data requirement

In this section, we will be working with data requirement as well as data collectionfor this data analytics problem. First let's see how the requirement for data can beachieved for this problem.

Since this is a web analytics problem, we will use Google Analytics data source.To retrieve this data from Google Analytics, we need to have an existent GoogleAnalytics account with web traffic data stored on it. To increase the popularity, wewill require the visits information of all of the web pages. Also, there are many otherattributes available in Google Analytics with respect to dimensions and metrics.

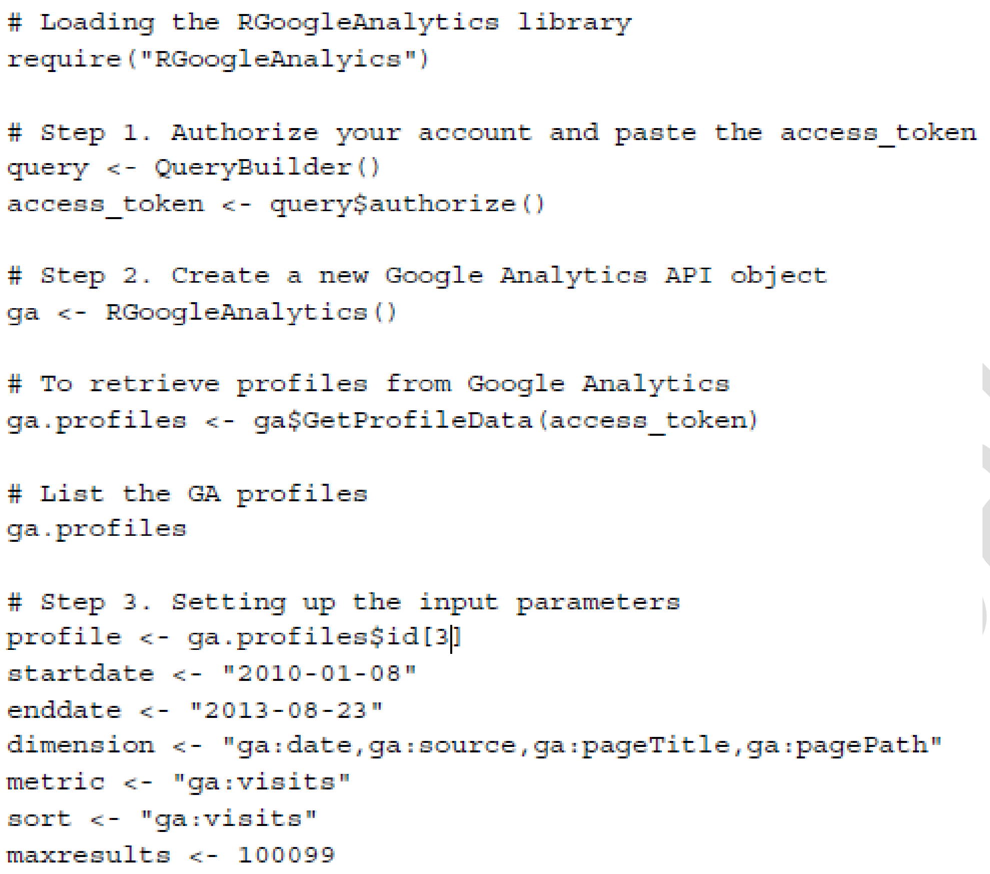
The header format of the dataset to be extracted from Google Analytics is as follows:date, source, pageTitle, pagePath

- date: This is the date of the day when the web page was visited
- source: This is the referral to the web page
- pageTitle: This is the title of the web page
- pagePath: This is the URL of the web page

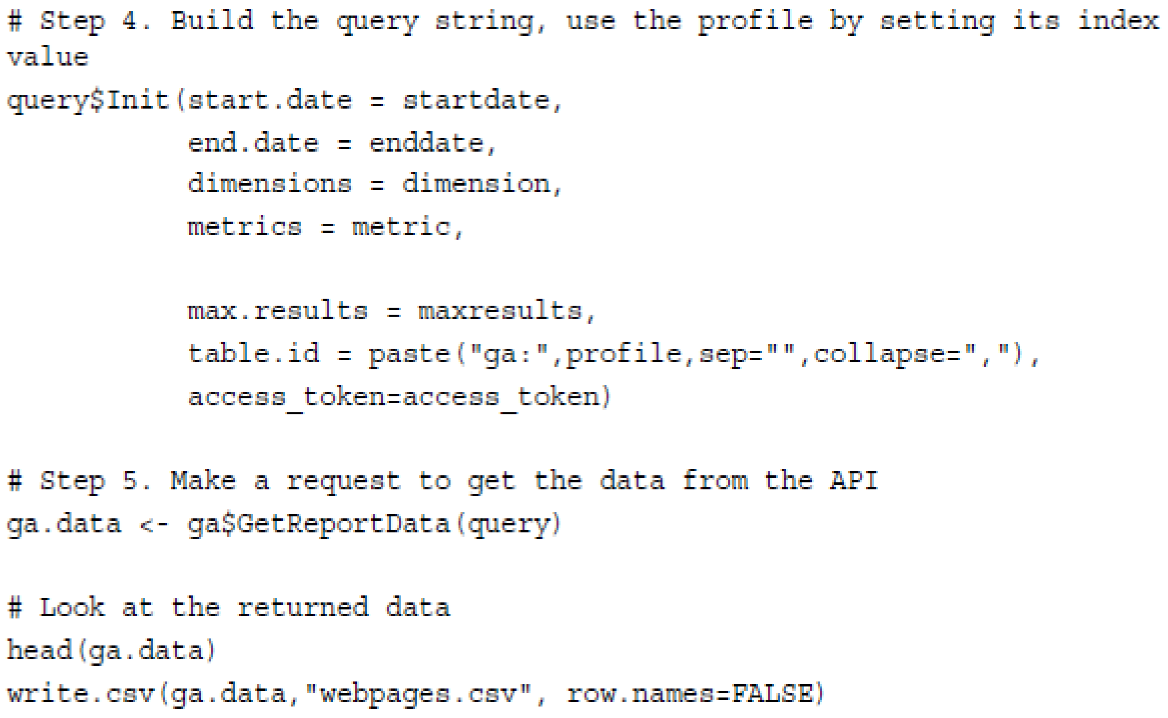
Collecting data

As we are going to extract the data from Google Analytics, we need to useRGoogleAnalytics, which is an R library for extracting Google Analytics datasetswithin R. To extract data, you need this plugin to be installed in R. Then you will be
able to use its functions.

The following is the code for the extraction process from Google Analytics:



```
# Loading the RGoogleAnalytics library
require("RGoogleAnalyics")

# Step 1. Authorize your account and paste the access_token
query <- QueryBuilder()
access_token <- query$authorize()

# Step 2. Create a new Google Analytics API object
ga <- RGoogleAnalytics()

# To retrieve profiles from Google Analytics
ga.profiles <- ga$GetProfileData(access_token)

# List the GA profiles
ga.profiles

# Step 3. Setting up the input parameters
profile <- ga.profiles$id[3]
startdate <- "2010-01-08"
enddate <- "2013-08-23"
dimension <- "ga:date,ga:source,ga:pageTitle,ga:pagePath"
metric <- "ga:visits"
sort <- "ga:visits"
maxresults <- 100099
```

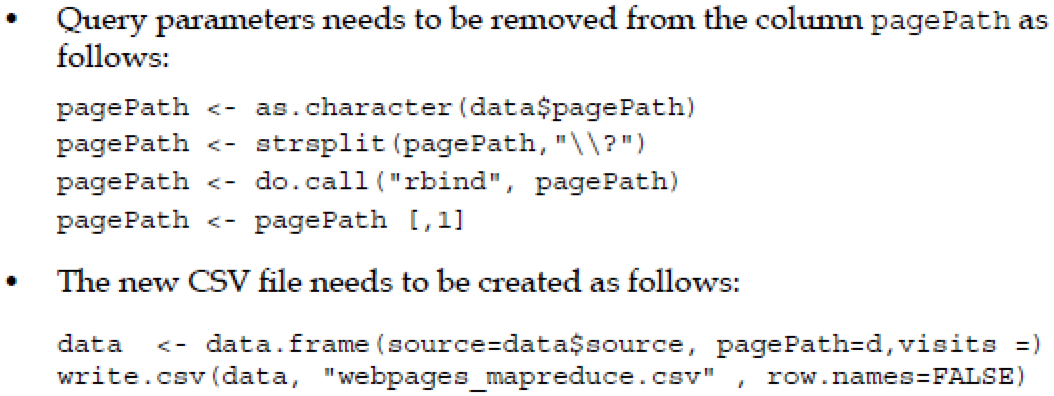


```
# Step 4. Build the query string, use the profile by setting its index
value
query$Init(start.date = startdate,
           end.date = enddate,
           dimensions = dimension,
           metrics = metric,

           max.results = maxresults,
           table.id = paste("ga:",profile,sep="",collapse=","),
           access_token=access_token)

# Step 5. Make a request to get the data from the API
ga.data <- ga$GetReportData(query)

# Look at the returned data
head(ga.data)
write.csv(ga.data,"webpages.csv", row.names=FALSE)
```

Preprocessing data

Now, we have the raw data for Google Analytics available in a CSV file. We need toprocess this data before providing it to the MapReduce algorithm.

There are two main changes that need to be performed into the dataset:



- Query parameters needs to be removed from the column `pagePath` as follows:

```
pagePath <- as.character(data$pagePath)
pagePath <- strsplit(pagePath,"\\?")
pagePath <- do.call("rbind", pagePath)
pagePath <- pagePath [,1]
```

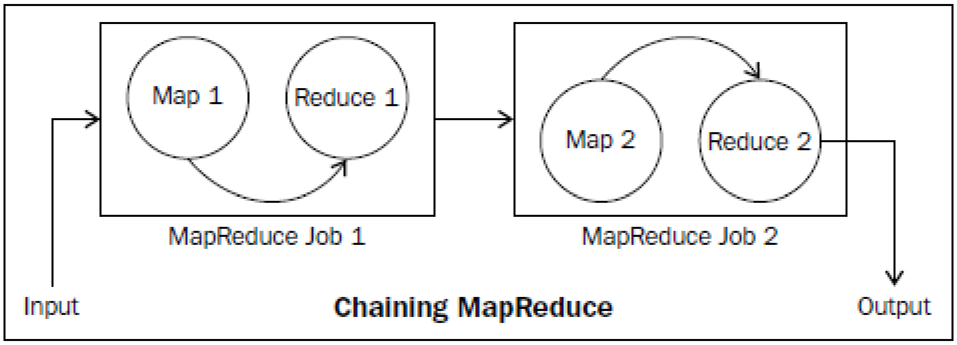
- The new CSV file needs to be created as follows:

```
data  <- data.frame(source=data$source, pagePath=d,visits =)
write.csv(data, "webpages_mapreduce.csv" , row.names=FALSE)
```

Performing analytics over data

To perform the categorization over website pages, we will build and run theMapReduce algorithm with R and Hadoop integration. sometimes we need to use multipleMappers and Reducers for performing data analytics; this means using the chainedMapReduce jobs.

In case of chaining MapReduce jobs, multiple Mappers and Reducers cancommunicate in such a way that the output of the first job will be assigned tothe second job as input. The MapReduce execution sequence is described in thefollowing diagram:



Chaining MapReduce

Visualizing data

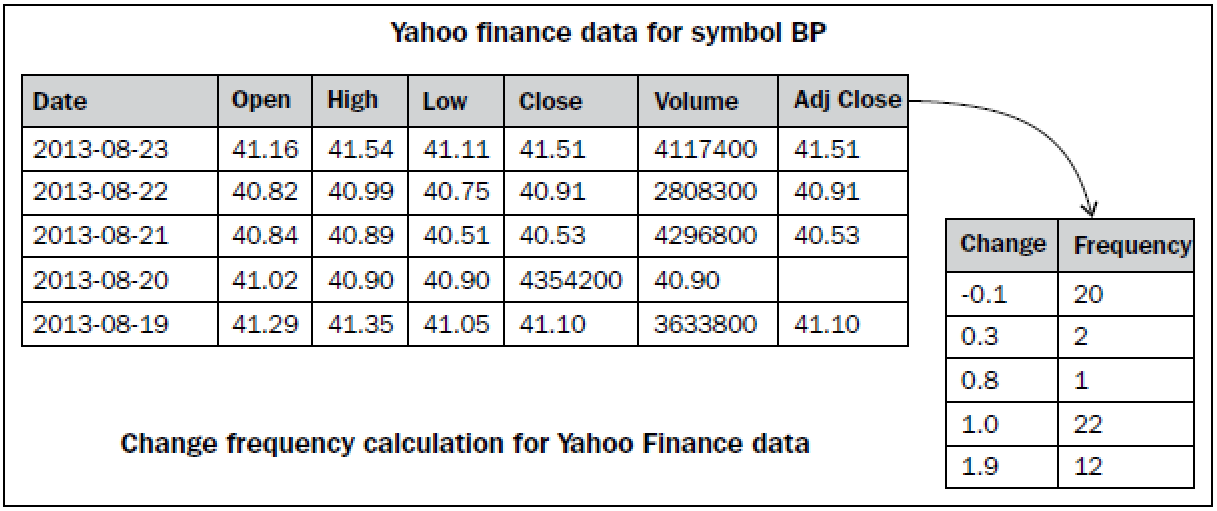
We collected the web page categorization output using the three categories. I thinkthe best thing we can do is simply list the URLs. But if we have more information,such as sources, we can represent the web pages as nodes of a graph, colored bypopularity with directed edges when users follow the links. This can lead to moreinformative insights.

*** Computing the frequency of stock marketchange**

This data analytics MapReduce problem is designed for calculating the frequency ofstock market changes.

Identifying the problem

Since this is a typical stock market data analytics problem, it will calculate thefrequency of past changes for one particular symbol of the stock market, such asa **Fourier Transformation**. Based on this information, the investor can get moreinsights on changes for different time periods. So the goal of this analytics is tocalculate the frequencies of percentage change.



Yahoo finance data for symbol BP

Date	Open	High	Low	Close	Volume	Adj Close
2013-08-23	41.16	41.54	41.11	41.51	4117400	41.51
2013-08-22	40.82	40.99	40.75	40.91	2808300	40.91
2013-08-21	40.84	40.89	40.51	40.53	4296800	40.53
2013-08-20	41.02	40.90	40.90	4354200	40.90	
2013-08-19	41.29	41.35	41.05	41.10	3633800	41.10

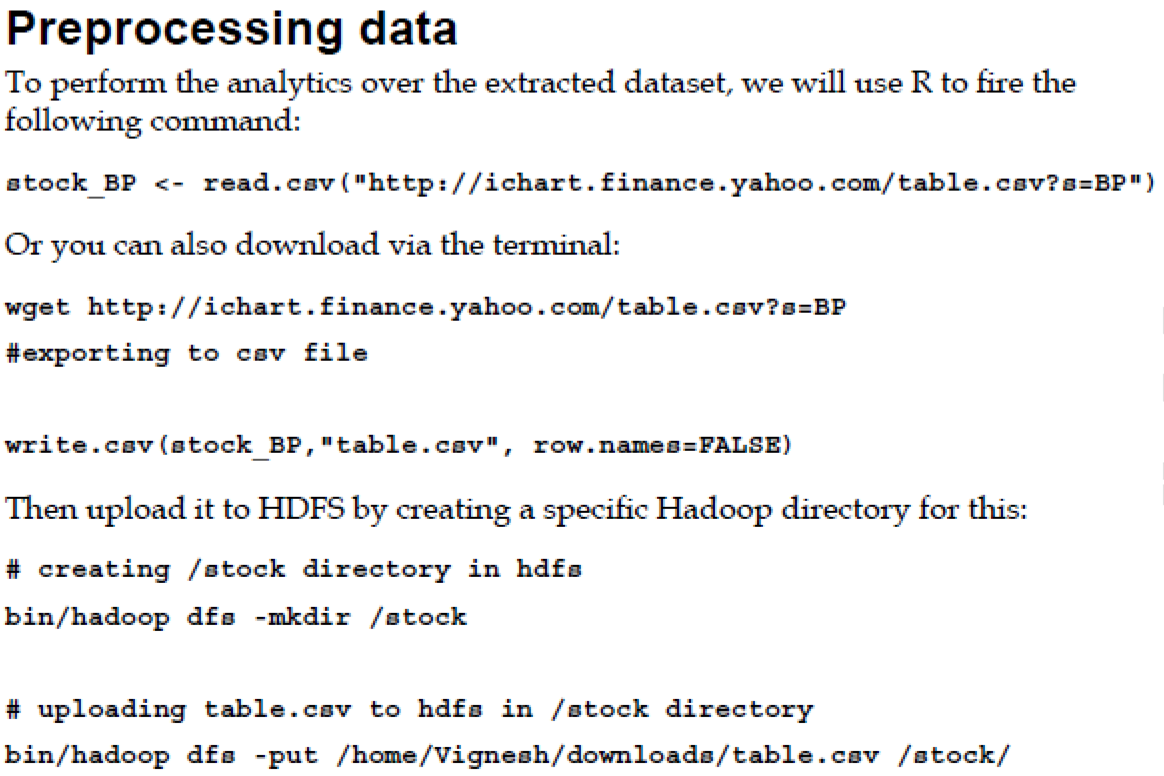
Change	Frequency
-0.1	20
0.3	2
0.8	1
1.0	22
1.9	12

Change frequency calculation for Yahoo Finance data

Designing data requirement

For this stock market analytics, we will use Yahoo! Finance as the input dataset. We need to retrieve the specific symbol's stock information. To retrieve this data, we will use the Yahoo! API with the following parameters:

- From month
- From day
- From year
- To month
- To day
- To year
- Symbol



Preprocessing data

To perform the analytics over the extracted dataset, we will use R to fire the following command:

```
stock_BP <- read.csv("http://ichart.finance.yahoo.com/table.csv?s=BP")
```

Or you can also download via the terminal:

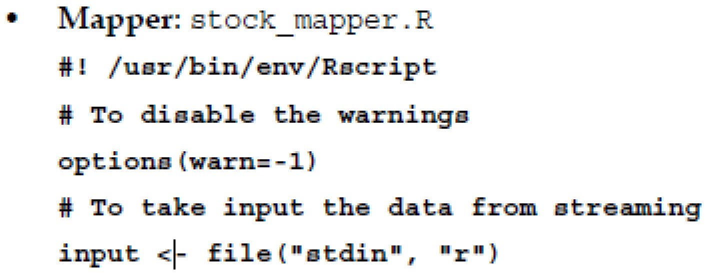
```
wget http://ichart.finance.yahoo.com/table.csv?s=BP
#exporting to csv file


write.csv(stock_BP,"table.csv", row.names=FALSE)
```

Then upload it to HDFS by creating a specific Hadoop directory for this:

```
# creating /stock directory in hdfs
bin/hadoop dfs -mkdir /stock


# uploading table.csv to hdfs in /stock directory
bin/hadoop dfs -put /home/Vignesh/downloads/table.csv /stock/
```

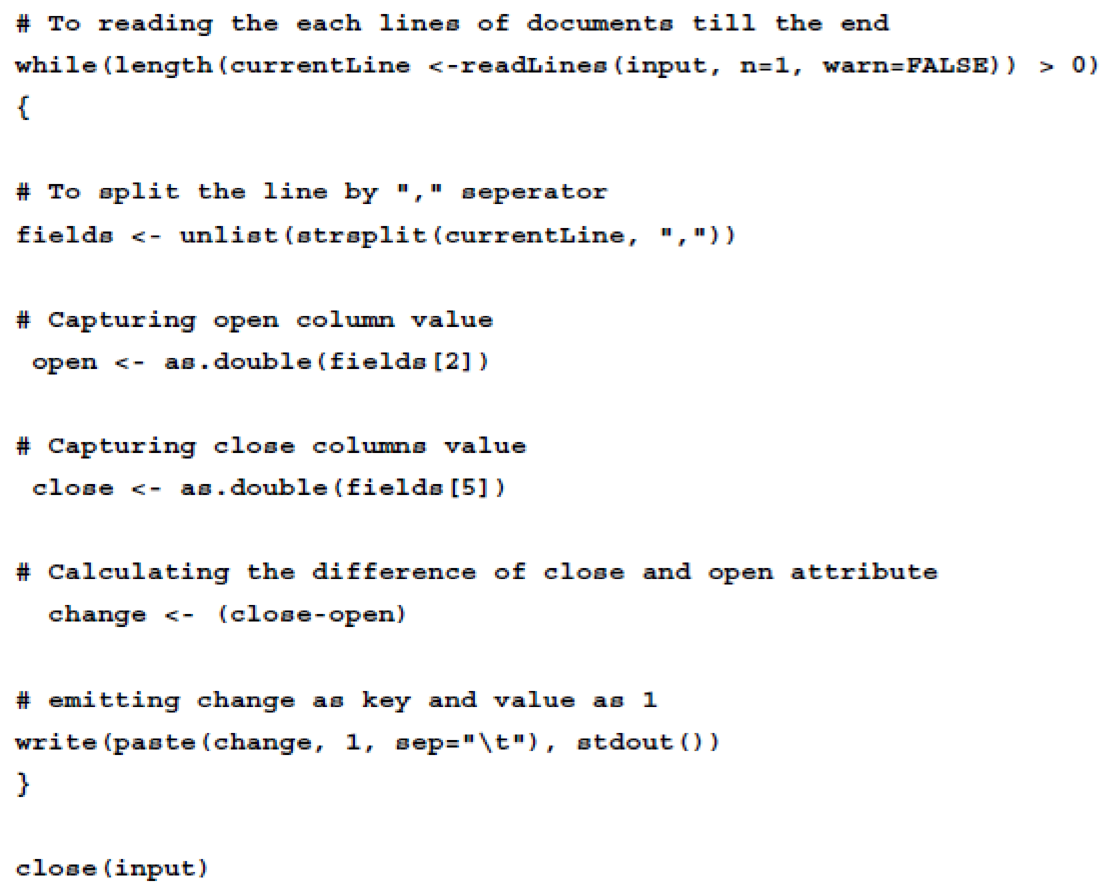
Performing analytics over data

To perform the data analytics operations, we will use streaming with R and Hadoop(without the HadoopStreaming package). So, the development of this MapReducejob can be done without any RHadoop integrated library/package.In this MapReduce job, we have defined Map and Reduce in different R files to beprovided to the Hadoop streaming function.



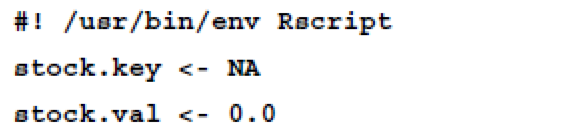
- **Mapper**: `stock_mapper.R`

```
#! /usr/bin/env/Rscript
# To disable the warnings
options(warn=-1)
# To take input the data from streaming
input <- file("stdin", "r")
```

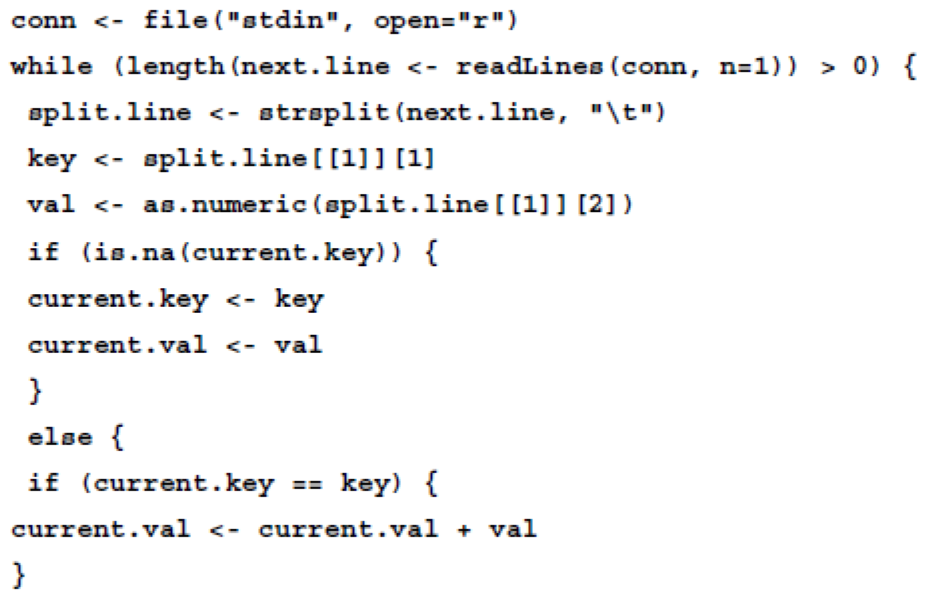


```
# To reading the each lines of documents till the end
while(length(currentLine <-readLines(input, n=1, warn=FALSE)) > 0)
{

# To split the line by "," seperator
fields <- unlist(strsplit(currentLine, ","))

# Capturing open column value
 open <- as.double(fields[2])

# Capturing close columns value
 close <- as.double(fields[5])

# Calculating the difference of close and open attribute
  change <- (close-open)

# emitting change as key and value as 1
write(paste(change, 1, sep="\t"), stdout())
}

close(input)
```

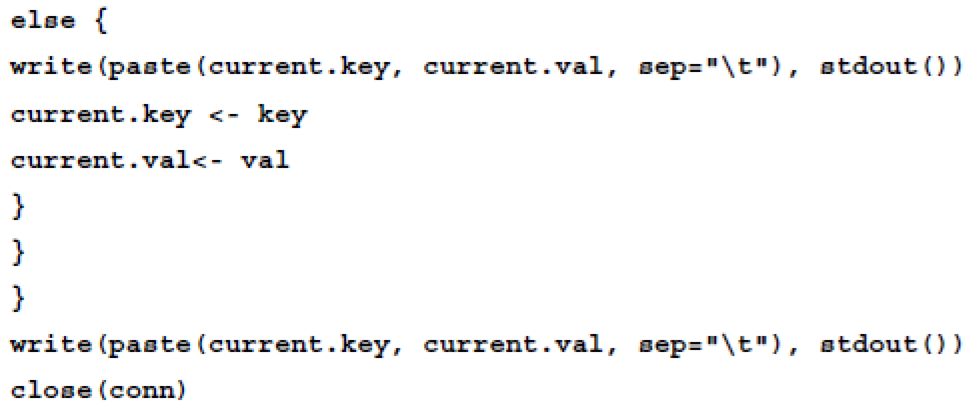
- **Reducer:** stock_reducer.R



```
#! /usr/bin/env Rscript
stock.key <- NA
stock.val <- 0.0
```



```
conn <- file("stdin", open="r")
while (length(next.line <- readLines(conn, n=1)) > 0) {
 split.line <- strsplit(next.line, "\t")
 key <- split.line[[1]][1]
 val <- as.numeric(split.line[[1]][2])
 if (is.na(current.key)) {
 current.key <- key
 current.val <- val
 }
 else {
 if (current.key == key) {
current.val <- current.val + val
}
```



```
else {
write(paste(current.key, current.val, sep="\t"), stdout())
current.key <- key
current.val<- val
}
}
}
write(paste(current.key, current.val, sep="\t"), stdout())
close(conn)
```

Introduction to machine learning

Machine learning is a branch of artificial intelligence that allows us to make ourapplication intelligent without being explicitly programmed. Machine learningconcepts are used to enable applications to take a decision from the availabledatasets. A combination of machine learning and data mining can be used to developspam mail detectors, self-driven cars, speech recognition, face recognition, andonline transactional fraud-activity detection.

There are many popular organizations that are using machine-learning algorithmsto make their service or product understand the need of their users and provideservices as per their behavior. Google has its intelligent web search engine, whichprovides a number one search, spam classification in Google Mail, news labeling inGoogle News, and Amazon for recommender systems. There are many open sourceframeworks available for developing these types of applications/frameworks, suchas R, Python, Apache Mahout, and Weka.

Types of machine-learning algorithms

There are three different types of machine-learning algorithms for intelligent systemdevelopment:

- Supervised machine-learning algorithms
- Unsupervised machine-learning algorithms
- Recommender systems

Supervised machine-learning algorithms

In this section, we will be learning about supervised machine-learning algorithms.

The algorithms are as follows:

- Linear regression

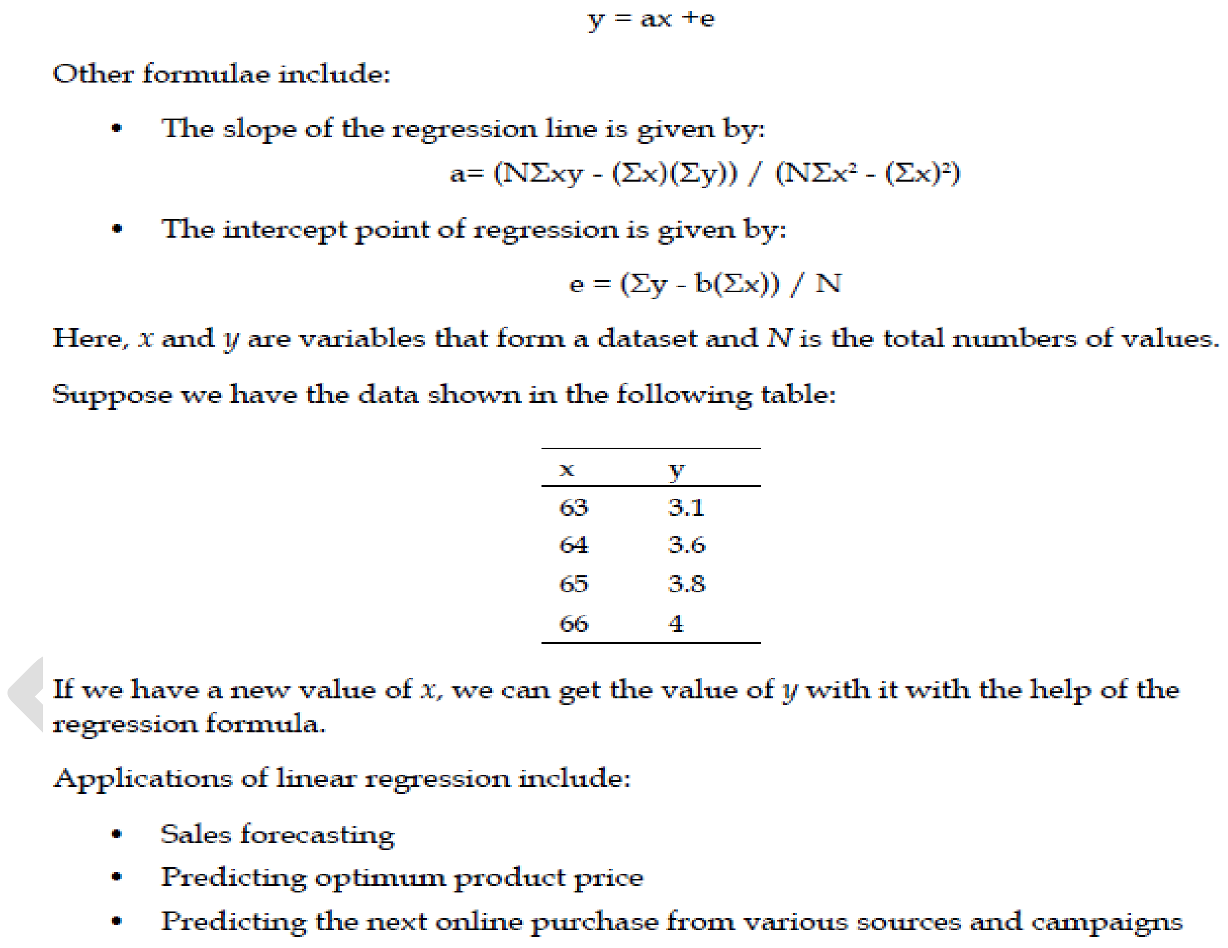
- Logistic regression

Linear regression

Linear regression is mainly used for predicting and forecasting values based onhistorical information. Regression is a supervised machine-learning technique toidentify the linear relationship between target variables and explanatory variables.We can say it is used for predicting the target variable values in numeric form.

In the following section, we will be learning about linear regression with R and linearregression with R and Hadoop.Here, the variables that are going to be predicted are considered as target variablesand the variables that are going to help predict the target variables are calledexplanatory variables. With the linear relationship, we can identify the impact of achange in explanatory variables on the target variable.

In mathematics, regression can be formulated as follows:



$$y = ax + e$$

Other formulae include:

- The slope of the regression line is given by:

$$a= (N\Sigma xy - (\Sigma x)(\Sigma y)) / (N\Sigma x^2 - (\Sigma x)^2)$$

- The intercept point of regression is given by:

$$e = (\Sigma y - b(\Sigma x)) / N$$

Here, *x* and *y* are variables that form a dataset and *N* is the total numbers of values.

Suppose we have the data shown in the following table:

x	y
63	3.1
64	3.6
65	3.8
66	4

If we have a new value of *x*, we can get the value of *y* with it with the help of the regression formula.

Applications of linear regression include:

- Sales forecasting
- Predicting optimum product price
- Predicting the next online purchase from various sources and campaigns

Logistic regression

In statistics, logistic regression or logit regression is a type of probabilistic classification model. Logistic regression is used extensively in numerous disciplines, including the medical and social science fields. It can be binomial or multinomial. Binary logistic regression deals with situations in which the outcome for a dependent variable can have two possible types. Multinomial logistic regression deals with situations where the outcome can have three or more possible types.

Logistic regression can be implemented using logistic functions, which are listed here.

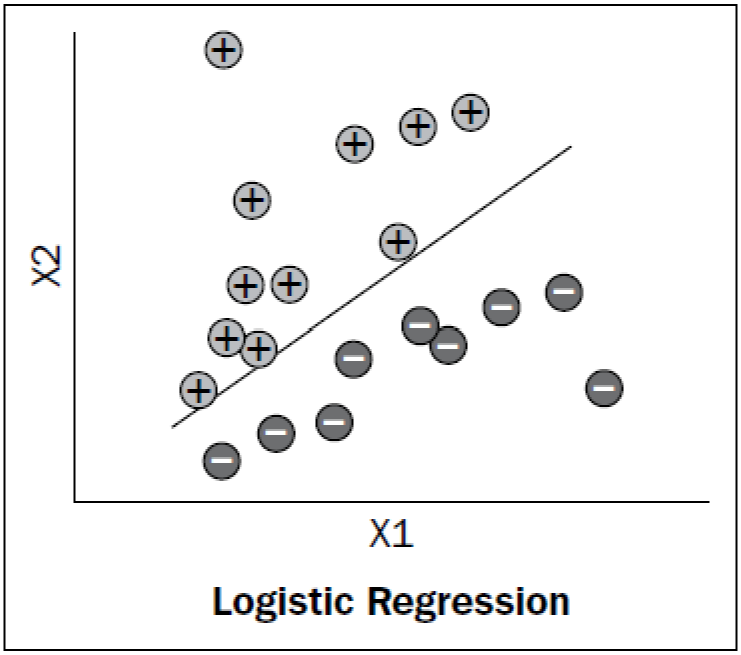
- To predict the log odds ratios, use the following formula:

  $\mathrm{logit}(p) = \beta 0 + \beta 1 \times x1 + \beta 2 \times x2 + \ldots + \beta n \times xn$

- The probability formula is as follows:

  $p = e^{\mathrm{logit}(p)} / 1 + e^{\mathrm{logit}(p)}$

logit(p) is a linear function of the explanatory variable, X (x1,x2,x3..xn), whichis similar to linear regression. So, the output of this function will be in the range 0 to 1. Based on the probability score, we can set its probability range from 0 to 1.In a majority of the cases, if the score is greater than 0.5, it will be considered as 1,otherwise 0. Also, we can say it provides a classification boundary to classify theoutcome variable.



Logistic Regression

The preceding figure is of a training dataset. Based on the training dataset plot, wecan say there is one classification boundary generated by the glm model in R.

Applications of logistic regression include:

- Predicting the likelihood of an online purchase
- Detecting the presence of diabetes

Unsupervised machine learningalgorithm

In machine learning, unsupervised learning is used for finding the hidden structurefrom the unlabeled dataset. Since the datasets are not labeled, there will be no errorwhile evaluating for potential solutions.

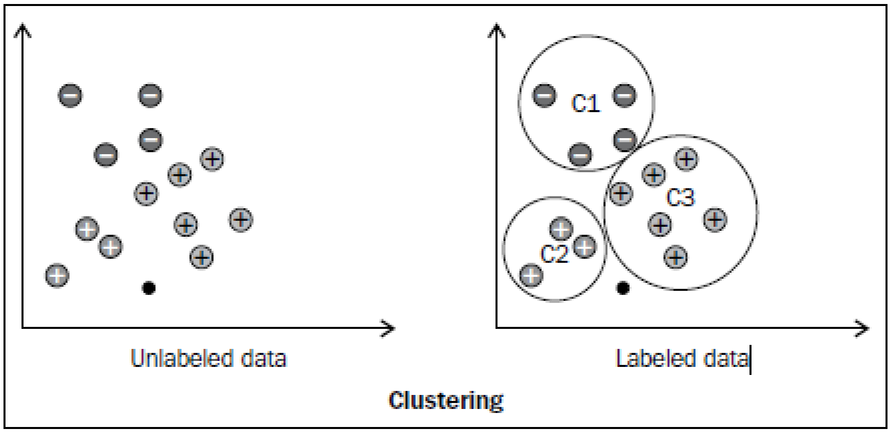
Unsupervised machine learning includes several algorithms, some of which are asfollows:

- Clustering
- Artificial neural networks
- Vector quantization

Clustering

Clustering is the task of grouping a set of object in such a way that similar objectswith similar characteristics are grouped in the same category, but other objects aregrouped in other categories. In clustering, the input datasets are not labeled; theyneed to be labeled based on the similarity of their data structure.

In unsupervised machine learning, the classification technique performs the sameprocedure to map the data to a category with the help of the provided set of inputtraining datasets. The corresponding procedure is known as clustering (or clusteranalysis), and involves grouping data into categories based on some measure ofinherent similarity; for example, the distance between data points.From the following figure, we can identify clustering as grouping objects based ontheir similarity:



There are several clustering techniques available within R libraries, such as k-means, k-medoids, hierarchical, and density-based clustering. Among them, k-means is widely used as the clustering algorithm in data science. This algorithm asks for anumber of clusters to be the input parameters from the user side.

Applications of clustering are as follows:

• Market segmentation
• Social network analysis
• Organizing computer network
• Astronomical data analysis

Recommendation algorithms

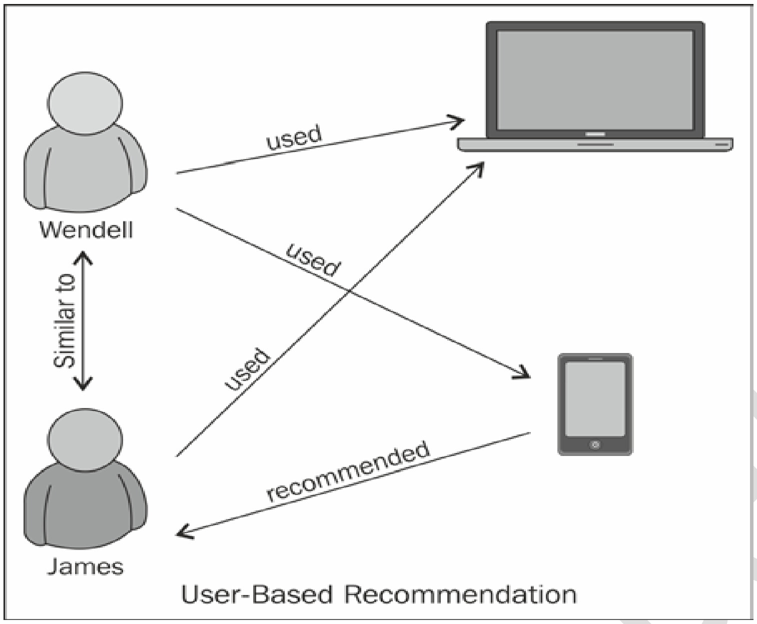
Recommendation is a machine-learning technique to predict what new items a userwould like based on associations with the user's previous items. Recommendationsare widely used in the field of e-commerce applications. Through this flexible dataand behavior-driven algorithms, businesses can increase conversions by helping toensure that relevant choices are automatically suggested to the right customers at theright time with cross-selling or up-selling.

For example, when a customer is looking for a Samsung Galaxy S IV/S4 mobilephone on Amazon, the store will also suggest other mobile phones similar to thisone, presented in the **Customers Who Bought This Item Also Bought** window.

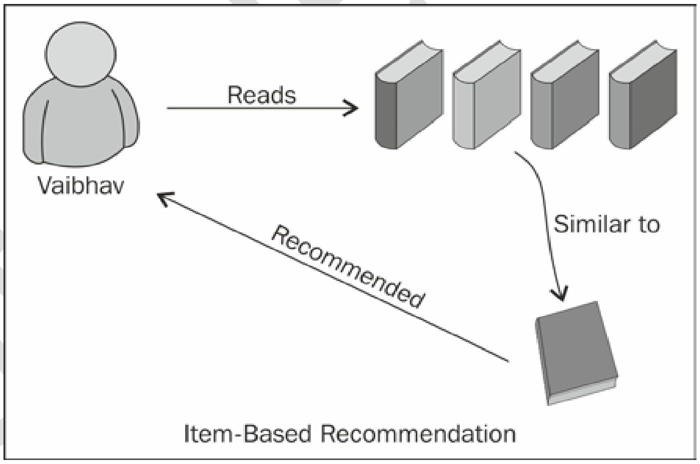
There are two different types of recommendations:

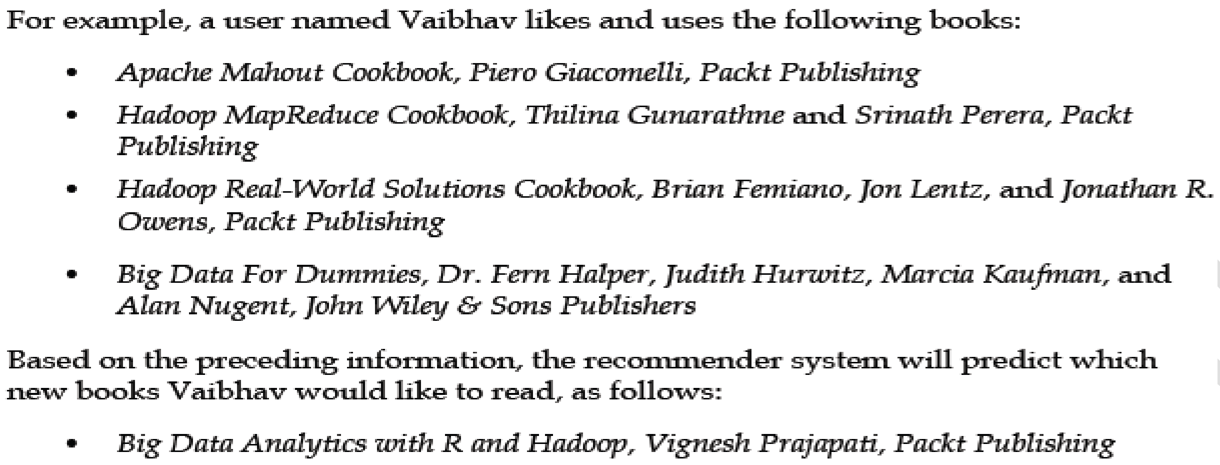
- **User-based recommendations**: In this type, users (customers) similar to current user (customer) are determined. Based on this user similarity, their interested/used items can be recommended to other users. Let's learn it through an example.

  Assume there are two users named Wendell and James; both have a similarinterest because both are using an iPhone. Wendell had used two items, iPadand iPhone, so James will be recommended to use iPad. This is user-basedrecommendation.



- **Item-based recommendations**: In this type, items similar to the items that are being currently used by a user are determined. Based on the item-similarity score, the similar items will be presented to the users for cross-selling and up-selling type of recommendations. Let's learn it through an example.





For example, a user named Vaibhav likes and uses the following books:

- *Apache Mahout Cookbook, Piero Giacomelli, Packt Publishing*
- *Hadoop MapReduce Cookbook, Thilina Gunarathne* and *Srinath Perera, Packt Publishing*
- *Hadoop Real-World Solutions Cookbook, Brian Femiano, Jon Lentz,* and *Jonathan R. Owens, Packt Publishing*
- *Big Data For Dummies, Dr. Fern Halper, Judith Hurwitz, Marcia Kaufman,* and *Alan Nugent, John Wiley & Sons Publishers*

Based on the preceding information, the recommender system will predict which new books Vaibhav would like to read, as follows:

- *Big Data Analytics with R and Hadoop, Vignesh Prajapati, Packt Publishing*

Question Bank

Unit I

1. What is Big Data? Explain Characteristics of Big Data.
2. Explain 3 Vs of Big Data.
3. Explain Data Analytics Lifecycle.
4. Explain the features of R language.
5. Understanding the different modes of Hadoop.
6. Explain the features of Hadoop.
7. Draw and explainthe architecture of HDFS and MapReduce.
8. Explain Data Mining Techniques.
9. List the benefits of Big Data.

Unit II

1. Write short note on Hadoop MapReduce.
2. Explain the fundamental of Hadoop MapReduce.
3. Explain different ways to write MapReduce in R.
4. Explain RHIPE architecture in details.
5. Write short note on RHadoop.

Unit III

1. Explain in details data analytics project cycle.
2. Explain data analytics problem for web page categorization.
3. Explain data analytics problem for Stock market change.
4. Explain supervised machine learning algorithm.
5. Explain unsupervised machine learning algorithm.

Unit IV

Introduction to Business Intelligence : evolution of BI, BI value chain, introduction to business analytics, BI Definitions & Concepts, Business Applications of BI, BI Framework, Role of Data Warehousing in BI, BI Infrastructure Components – BI Process, BI Technology, BI Roles & Responsibilities.

Evolution of Business Intelligence

Now consider the evolution of Business Intelligence (BI). The origins of BI go back to the humble application report, generated in the days of COBOL and assembler processing. The application report purported to tell the organization what was happening in the environment. While the humble COBOL or assembler report served a very real purpose, there were many problems with application reports. Application reports chewed up a lot of paper. Application reports took a long time to run. Application reports were usually out of date by the time they were printed. But application reports were a start.

Soon there were online transactions which also told the organization what was occurring, but in real time. Online transactions took no paper. Online transactions showed data that was up to date as of the moment of access. And online transactions were fast. But online transactions had their limitations, too. Online transactions were not good for showing large amounts of data. Online transactions did not leave an auditable trail. And the data behind online transactions often changed by the second. A person could do an online transaction only to have the information invalidated in the next second. And online transaction processing systems were expensive and fragile.

But the real problem behind online transactions was that online transactions showed only a limited type of data and showed data that was unique to an application. Given enough online systems, it was possible to find the same piece of data with multiple different values coming from multiple different applications. User A looks for some information and finds it. User B looks for the same information in another system, and finds it. However the values shown to user A are not the values shown to user B.

The value of integrating data into a corporate format began to be obvious. Decision making in a world where there was no definitive source of data became a challenging and dicey exercise. For large corporations, it became obvious that a historical, integrated, subject oriented source of data was needed for the corporation. Having multiple overlapping applications simply was not a sound basis for business decisions. Thus born was the data warehouse. With the data warehouse it was now possible to do a whole new style of reporting and analysis. Once there was definitive corporate data in a data warehouse, the world of BI began, at least as we know BI today.

Business Intelligence Value Chain

Business intelligence and the development of an intelligent learning organization represent a popular trend in many public and private sector organizations. Ideally, any manager or knowledge worker should be able to compose information requests without programmer assistance and achieve answers at the speed of thought. Follow-up questions should be immediately asked and answered in order to maintain continuity of thought on a particular topic of importance.

Intelligence is the ability to learn, to understand or to deal with new or trying situations; the skilled use of reason; the ability to apply knowledge to manipulate one's environment or to think abstractly. *Business intelligence* is a set of concepts, methods and processes to improve business decisions using information from multiple sources and applying experience and assumptions to develop an accurate understanding of business dynamics. It is the gathering, management and analysis of data to produce information that is distributed to people throughout the organization to improve strategic and tactical decisions.

Business intelligence involves the integration of core information with relevant contextual information to detect significant events and illuminate cloudy issues. It includes the ability to monitor business trends, to evolve and adapt quickly as situations change and to make intelligent business decisions on uncertain judgements and contradictory information. It relies on exploration and analysis of unrelated information to provide relevant insights, identify trends and discover opportunities.

Business intelligence requires high-quality information which can only be derived from a high-quality data resource. If organizations are moving into business intelligence scenarios as a major initiative, they must understand the need for and value of a high-quality data resource. There is little doubt that the technology will support business intelligence. The real issue is how to clean up disparate data and produce a high-quality data resource that truly supports business intelligence.

Data resource quality is not the same as information quality, though the two are often confused and used interchangeably in this lexically challenged discipline. *Information quality* is how well the demand for business information is met. It includes the data used to produce the information and the information engineering process. The information engineering process includes everything from properly determining the information need to presenting the information.

Data resource quality is how well the data resource supports information engineering in meeting the current and future demand for business information. A high-quality data resource consistently meets the expectations of information engineering. Since the expectations for information constantly change, the expectations for data also change. A high-quality data resource consistently meets those changing expectations.

The foundation of a house is critical to the quality of the entire house. If the foundation is not level and square, the entire house will not be level or square and carpenters will fight it to the last shingle on the peak of the roof. Thus the quality of the house will be lower. The data resource, like the foundation of a house, is the foundation of an information engineering/knowledge environment/business intelligence value chain that supports business strategies, as shown in Figure 1. If the data resource is not "level and square," an organization will fight it clear to the business strategies and the quality of support will be lower. If the data resource is high-quality, that quality will enhance the entire value chain to the benefit of the business.

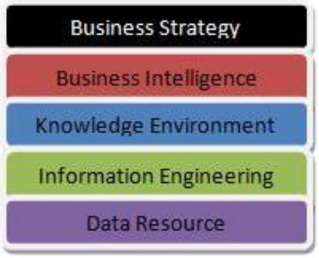


Figure 1: The Business Intelligence Value Chain

The value chain begins with the data resource. Information is developed from the data resource to support the knowledge environment of an intelligent learning organization. Data is the raw material for information which is the raw material for the knowledge environment. Knowledge is the raw material for business intelligence that supports business strategies.

Data is the individual raw facts that are out of context, have no meaning and are difficult to understand. Facts are numbers, characters, character strings, text, images, voice, video and any other form in which a fact may be presented. *Data in context* is facts that have meaning and can be readily understood. It is the raw facts in context with meaning and understanding, but is not yet information because it has no relevance or time frame.

Information is a set of data in context that is relevant to one or more people at a point in time or for a period of time. It is data in context with respect to understanding what the facts mean. Information is data imbued with meaning, relevance and purpose. A set of data in context is a message that only becomes information when one or more people are ready to accept that message as relevant to their needs. A message without meaning, relevance or purpose is simply noise.

Knowledge is cognizance, cognition, the fact or condition of knowing something with familiarity gained through experience or association. It is the acquaintance with or the understanding of something, the fact or condition of being aware of something, of apprehending truth or fact. *Tacit knowledge* is all the knowledge that is in people's heads or the heads of a community of people, such as an organization. It is what makes people smart and act intelligently. *Explicit knowledge* is knowledge that has been rendered explicitly to a community of people, such as an organization, and is what they deem to know.

Organizational knowledge is information that is of significance to the organization, is combined with experience and understanding and is retained. It is information in context with respect to understanding what is relevant and significant to business issues. It is analysis, reflection and synthesis about what information means to the business and how it can be used to advantage. It is the ability to learn, understand and deal with new and trying situations; to apply knowledge and think abstractly. It is the core of an intelligent learning organization that must be accumulated, cultivated and managed. Organizational knowledge is a rational interpretation of information that leads to business intelligence.

An organization has three primary resources: capital, human resource and data resource. The data resource and information engineering, collectively, are the technology resource that supports the human resource in an intelligent learning organization. The knowledge environment and business intelligence, collectively, are the human resource that uses information to support the business strategies. It is the human resource that possesses the business intelligence, the intelligence and the wisdom to support business strategies. Information is the link between the data resource and the human resource.

Knowledge cannot be managed in the sense that data and information are managed. Only an environment that promotes the exchange of information to create knowledge can be managed. Cognition is the application of knowledge, and no technologies today can automate cognition. Knowledge management, as often promoted today, is just another silver bullet that, like so many other silver bullets, will become tarnished with time.

Knowledge management is really the management of an environment where people generate tacit knowledge, render it into explicit knowledge and feed it back to the organization. This forms the base for more tacit knowledge which keeps the cycle going in an intelligent, learning organization. It is the process of creating, institutionalizing and distributing information between people. It is the process of finding bodies of knowledge specific to a need and presenting them in a suitable manner for specific business purposes. Knowledge management matches a knowledge seeker with the best source of knowledge through profiles to share their knowledge. It is an integrated approach to identifying, sharing and evaluating an organization's information. It is a culture for learning where people are encouraged to share information and best practices to solve business problems rather than continually reinventing the wheel.

Knowledge management promotes and relies on information sharing. *Information sharing* is the active sharing and utilization of information in a knowledge environment for specific business advantage. It is the sharing of memories about past situations and solutions, communicating learning experiences and exchanging a deeper understanding of problems and solutions. It is a way of tapping the tremendous hidden knowledge that resides in the human resource for the benefit of the organization.

Data and data in context can be stored in databases. Most databases are just files or tables of facts. Information, as defined above, cannot be stored in databases because relevancy to people at a point in time cannot be stored. If information is stored, it is really stored as data in context. If a report is stored with labels and headings, it is really data in context that is being stored. When that report is retrieved and has relevance and purpose, it becomes information. Data that instructs machines to perform a task or directs the actions of that machine or data that moves between applications is still data, not information. Information pertains to people and must have relevance at a point in time.

Knowledge, as previously defined, cannot be stored any more than information can be stored. Knowledge resides in the human resource of an organization. Knowledge storage is often described as subject matter experts diligently researching and capturing relevant information and storing it for sharing among knowledge workers. Really, it is data in context that is being stored to support information sharing. The data may be hard facts and soft opinions, history of past events, situational evaluations, situations to avoid, alternatives to pursue, and so on, but it is still stored as data in context.

Knowledge is intellectual capital that is retained by the human resource. The knowledge base is in the human resource of an organization. Institutional memory, business knowledge and business experience reside in the human resource of an organization. Knowledge is only a resource with respect to the human resource, not with respect to computer storage and retrieval. *Knowledge quality* is the both the quality of the environment for sharing information and the quality of the human resource that discovers, develops and retains the knowledge. It is the quality with which information is shared for the discovery and accumulation of knowledge.

An organization's data resource is the total of all data within and without the organization that is available to the organization. It includes primitive and derived data; tabular and non-tabular data (spatial, image, textual, voice and so on, referred to as data mega types); elemental and combined data; automated and non-automated data; persistent and non-persistent data; and historical, current and projection data. It includes data in databases as well as data on reports, screens and documents; hard and soft data; internal and external data; global and local data; and enterprise-wide and application-specific data. It includes data used by traditional information systems, expert systems, executive information systems, geographic information systems, data warehouses and object oriented systems. It crosses all business activities, all projects and all information systems regardless of where they reside, who uses them or how they are used. It includes all data regardless of location, origin, form or method of storage.

The quality of the total data resource in most public and private sector organizations is low by reason of its disparity. A low-quality disparate data resource cannot provide high-quality information and cannot adequately support business intelligence. All data must be inventoried, understood and integrated within common data architecture to adequately support a business intelligence initiative.[1] All data means all data mega types, not just tabular data, because comprehensive business intelligence depends on the value of all types of data in the data resource. The data resource is the foundation for business intelligence, and the quality of the support for business strategies from an intelligent learning organization can be no better than the quality of the data resource. There must be a high-quality integrated data resource, high-quality information preparation and sharing and a high-quality human resource to discover and accumulate knowledge to achieve successful business intelligence.

Overview of Business Analytics

INTRODUCTION

Focus on business analytics has increased steadily over the past decade as evidenced by the continuously growing business analytics software market. Business analytics is reaching more organizations and extends to a wider range of users, from executives and line of business managers to analysts and other knowledge workers, within organizations. In an environment of increasingly faster growing data volumes where operating on intuition is no longer an option, business analytics provide the means to both optimize the organization internally and at the same time maintain flexibility to face unexpected external forces.

DEFINITION OF BUSINESS ANALYTICS

Business analytics includes software and business processes that enable organizations to apply metrics-based decision making to all functions ranging from supply chain and financial management to workforce and customer relationship management. Business analytics software comprises tools and applications for tracking, storing, analyzing, and modeling data in support of decision-making processes. This software market includes both application development tools and packaged analytic applications. The tools segments of the market include data warehouse generation, data warehouse management, business intelligence, technical data analysis, and spatial information management tools. The applications segments of the market include CRM, operations, financial and business performance management analytic applications. In 2002 the worldwide business analytics software market stood at $12 billion and is expected to grow at a compound annual growth rate of 6.0% over the next 5 years.

BENEFITS OF BUSINESS ANALYTICS

By implementing transaction-processing systems ranging from ERP, CRM, SCM, and eCommerce applications, organizations have taken a big step towards automating business processes. Business analytics software enables organizations to monitor, capture and analyze the vast amounts of data generated by these applications and provide management and staff at all levels with tools necessary to optimize these processes through strategic and tactical decisions.

"Financial Impact of Business Analytics"

an ROI study conducted by IDC in 2002 evaluated return on investment of business analytics projects at organizations throughout North American and Western Europe. The results showed that the median overall return on investment from business analytics projects was 112% with 49% of the organizations deriving benefits within one year. However, the ROI of business analytics projects impacts also return from other enterprise application projects. A CRM analytic application, for example, has direct ROI implications for the analytic and operational CRM components and is the engine that drives return beyond the productivity gains achieved from initial automation of marketing, sales or customer service processes

Business Intelligence

What is Business Intelligence (BI)?

Def 1:-

The term **Business Intelligence (BI)** refers to technologies, applications and practices for the collection, integration, analysis, and presentation of business information. The purpose of Business Intelligence is to support better business decision making. Essentially, Business Intelligence systems are data-driven Decision Support Systems (DSS). Business Intelligence is sometimes used interchangeably with briefing books, report and query tools and executive information systems.

Def 2:-

Business intelligence (BI) is a technology-driven process for analyzing data and presenting actionable information to help corporate executives, business managers and other end users make more informed business decisions. BI encompasses a variety of tools, applications and methodologies that enable organizations to collect data from internal systems and external sources, prepare it for analysis, develop and run queries against the data, and create reports, dashboards and data visualizations to make the analytical results available to corporate decision makers as well as operational workers.

The potential benefits of business intelligence programs include accelerating and improving decision making; optimizing internal business processes; increasing operational efficiency; driving new revenues; and gaining competitive advantages over business rivals. BI systems can also help companies identify market trends and spot business problems that need to be addressed.

Importance of Business Intelligence tools or software solutions

Business Intelligence systems provide historical, current, and predictive views of business operations, most often using data that has been gathered into a data warehouse or a data mart and occasionally working from operational data. Software elements support reporting, interactive "slice-and-dice" pivot-table analyses, visualization, and statistical data mining. Applications tackle sales, production, financial, and many other sources of business data for purposes that include business performance management. Information is often gathered about other companies in the same industry which is known as benchmarking.

Business Intelligence Trends

Currently organizations are starting to see that data and content should not be considered separate aspects of information management, but instead should be managed in an integrated enterprise approach. Enterprise information management brings Business Intelligence and Enterprise Content Management together. Currently organizations are moving towards Operational Business Intelligence which is currently under served and uncontested by vendors. Traditionally, Business Intelligence vendors are targeting only top the pyramid but now there is a paradigm shift moving toward taking Business Intelligence to the bottom of the pyramid with a focus of self-service business intelligence.

Business Intelligence Concepts

It exist a lot of concepts and terms that it is necessary to know and handle when a team working with Business Intelligence issues. In order to really understand all these concepts and its relationships, it is necessary grouping these terms by functions inside the whole Business Intelligence Design and Implementation.

Instead of established concepts, every organization establishes its own interpretations for every term, so this article contains the most utilized interpretation for every term or concept.

From the functional point of view, we have these groups of components:

- **Transactional or Operational Systems – Source Data Systems:** The main source for business intelligence data to be analyzed is all data captured, processed and reported by all core transactional systems for the company or organization.

- **Data Transfers Processes – Data Interfaces – ETL Processes:** All necessary data must be processed from source data systems to a specialized repositories or to show to final users. These data interfaces are called ETL (Extract, Transform and Load) processes.

- **Data Repositories:** Depending on the size and the reach of this repository, it could be named data warehouse: when the stored data is about all organization or the most of this organization; or could be named datamart when the stored data is about isolated departments or organizational units.

- **Final Users Tools:** For obtaining, querying, analyzing and reporting valuable information, final users have special tools that access datawarehouses and data marts (even transactional data), and these tools access the data dictionaries for document and inform to users what is the accessed data and which is its meaning.

- **Information Distribution and Control:** Regular reports, news and other information must be delivered in a timely and secure fashion to any selected way like email, mobile, web and others to appropriate personnel.

- **Business intelligence applications for better decision-making**

- There is a need for faster decision-making in an environment of increasing complexity and information overload. Business intelligence (BI) applications help enterprises take fact-based decisions rapidly by better utilising and presenting data from within and outside the enterprise.
- This article describes the three broad phases in the evolution of IT applications for enterprises, from office automation to business intelligence. Most large enterprises have passed Phase 1 or 2 and are poised to reap the benefits of Phase 3.

Phase 1: Office automation

- During the first wave of IT enablement of enterprises, various business activities and processes are automated (e.g. invoicing, stock-keeping, accounting, payroll and others). These IT systems commonly known as ERP (enterprise resource planning), MRP (material resource planning), CRM (customer relationship management), HRMS (human resource management system), etc., speed up the business process and provide quick access to information across the enterprise.
- Typically, they maintain records in a database at the lowest level with all details of the transaction. They are used for data entry and operational reporting.

Phase 2: Data management

- Over time as office automation systems mature and became pervasive, it becomes apparent that enterprise data is siloed and fragmented across the enterprise. Data management becomes even more challenging when there are mergers and acquisitions and multiple data sets need to be consolidated.
- These pose several challenges such as poor data quality, incompatibilities between data sets, duplication of data and overheads in managing multiple systems. Consequently, the next phase of IT evolution is targeted at simplifying the information landscape of the enterprise.
- Master data management is an approach to standardise data comprising tools and technologies for classifying, normalising, consolidating and aggregating data across the enterprise to provide a consistent view. Typically, a data warehouse is established to centralise company-wide information on a uniform platform. This warehouse can be accessed by tools for reporting and analysis.

Phase 3: Business intelligence

- Business intelligence is the emerging class of IT applications that use information assets to aid in better decision-making. A variety of tools and techniques such as data mining, predictive analytics and data visualisation are employed to provide valuable insights into past, current and future business metrics.
- BI applications serve a critical function in achieving operational efficiency, integrated planning and coordination and monitoring.
- Here are the highlights of BI applications.

Single version of the truth: It provides consistent information in real time across the enterprise, thereby eliminating debates on the validity of data. It also visualises information through meaningful dashboards to allow for coordinated decision-making.

Metric trees: Business performance metrics are related to KPIs and are computed at various levels within the enterprise. Metrics are linked to each other to create a metric tree, which connects the low-level performance metrics with high-level outcome measures.

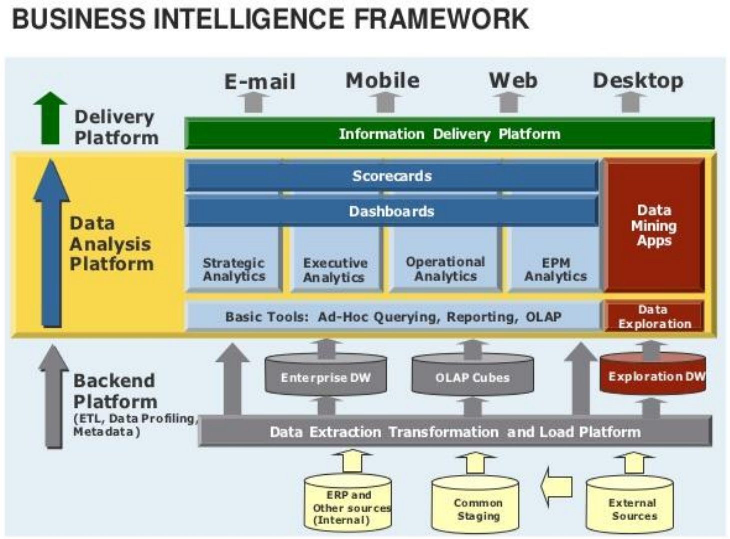
The golden triangle (budget, time and quality): One can foresee impact of changes in specifications and business case. This helps to manage the trade-off between budget, time and quality.

Business modelling: It captures business dynamics in robust and transparent models. These are useful for sensitivity analysis, simulation and scenario-based decision-making.

Looking backward, moving forward: It does not rely only on historic data to look into the future, but integrates external and internal data for better forecasting and predictive analytical capabilities.

- It is vital for enterprises to be well-informed and take quick, fact-based decisions in the dynamic marketplace. Business intelligence offers a ripe set of solutions that plug into existing IT infrastructure and bring out valuable insights.

BI framework



© 2012 These slides may not be copied or reproduced without written permission from Juniper Networks, Inc.

First is a mechanism to compare organization goals and measures with actual at different levels of business. Second is quantification and communication of the benefits in addressing the gap. In a Nutshell it needs to be part of the business or process improvement framework. In this article, I would like to put some examples and best practices I have experienced.

Is this Framework similar to BICC (Business Intelligence Competency Center) or a BI COE? One of the auto parts manufacturing major I have worked with never had both of these centers. But the business leaders in this company were able to show the investors, millions in savings with a successful business intelligence and performance management framework up to the lowest operational units. So what did they have?

The BI framework they have has the following components.

1. Infrastructure to produce consistent and comparable Information of performance indicators

2. Platform to evaluate, rank and collaborate best practices from the top performers

3. Institutionalize improvement program with executive sponsorship

4. Identify new performance monitoring areas and (or) targets.

5. Communicate and share the benefits on improvement.

That’s it? You are right. It is said that simple. But different organizations constitute these parts of framework in isolation and with different vigor. The auto parts manufacturing major I mentioned earlier, for example, has a framework with

1. Technical team to source the operational data and process the performance indicators. They are a group of IT and Data Analyst to provide the information tools and services on the performance indicators with comparisons, ranks, gap with targets, improvement trends etc.

2. Corporate business leadership has constituted a team led by senior vice-president directly reporting to CEO to lead this framework. Let me call this team as Performance management team or PM team. Directly reporting to VP are

Business Analysts from finance, sales, operations, planning etc. to facilitate the business leaders to monitor and report the performance and benefits on a monthly basis.

3. The projects derived to address the gap are integrated through organization’s six-sigma initiatives .

4. A steering committee headed by the business unit heads revisit & revise on a quarterly basis.

5. Very active involvement of corporate communication team and Human resource (HR) team facilitated by PM team. The activities involve communication within the organization, investors, press etc. and employee recognition.

This status is definitely the scenario of a well matured BI framework. The above mentioned corporation started this frame work for one business unit and single geography before rolling out to entire organization worldwide.

What is the role of Data Warehouse in the Business Intelligence

A cube does not strictly require a DW to work, but it does require a DW to work well

The benefit of a DW is that the data is logically designed for reporting, giving you better performance over on OLTP database, that is optimized for DML

With no DW, you face performance issues and a more complex design, with a lot of work to be done in the DSV

If just one OLTP DB then yes you probably need an DW or Data Mart, or atleast a seperate schema with RTA Tables would be beneficial, as I described in you other recent thread.

- Strictly speaking, one can perform BI on an OLTP database. However, the need of DW for BI becomes more acute when your organization scales out and when your OLTP performance becomes an issue. Your OLTP workload may be mission-critical and optimal performance is paramount, and you cannot afford to put your reporting workload on top of your OLTP workload on the same server. This is where you would want to periodically copy your OLTP database to a DW to offload your reporting needs with a DW.

As mentioned, in a sizable DB shop, the OLTP workload is optimized for data consistency, accuracy, and transactional performance (write performance), but it's not optimized for BI performance (read performance). For BI performance, you might want to join tables into extremely large fact tables, for example. This, again, is important only if you have a huge dataset and if you have diverse data sources that you want to merge with your data for BI (like government census data, stock data, etc), not if you're just trying to report off of one small database that never experiences peak load. DWs are tuned to process very large sets of data very quickly to generate analytical metrics. Think about a case where a manager wants to have some analysis done on last month's data to effect business decision for the coming month. If it takes your system 3 days to process the data set, then by the time a business decision can be made, the analysis may already be based on stale data.

So to answer: whether doing BI on an OLTP database is a good idea depends on your current OLTP workload, the DB size, the complexity and size of the dataset for BI, time-sensitivity of the analatyical data, etc, and how all these things will scale in the foreseeable future.

- When you build a cube, although you can build it on an OLTP Source, it does not really make sence to, either logically (from a data perspective) or from a performance angle.

The idea of an OLTP Database, is that it will be very normalized (data redundancy will be removed) but this, by definition, means more tables. The tables you have, the more tables you need to join to answer questions (satisft queries) and the slower the performance will be from a reporting angle.

Also, you would often have multiple OLTP databases that you will wish to join into one for reporting. If this is the case, then you would join them together into a DW.

Even if this is not the case, if you want to build cubes, you are better off creating a Data Mart (small DW) rather than trying to report on the normalized schema, because a star or snowflake schema, as I say, makes more sense and allows you to define much better hierarchies, etc, and also reduces the number of tables that need to be joined.

BUSINESS INTELLIGENCE INFRASTRUCTURE

Business organizations can gain a competitive advantage with a well-designed business intelligence (BI) infrastructure. Think of the BI infrastructure as a set of layers that begin with the operational systems information and meta data and end in delivery of business intelligence to various business user communities. These layers are illustrated in Figure 1.

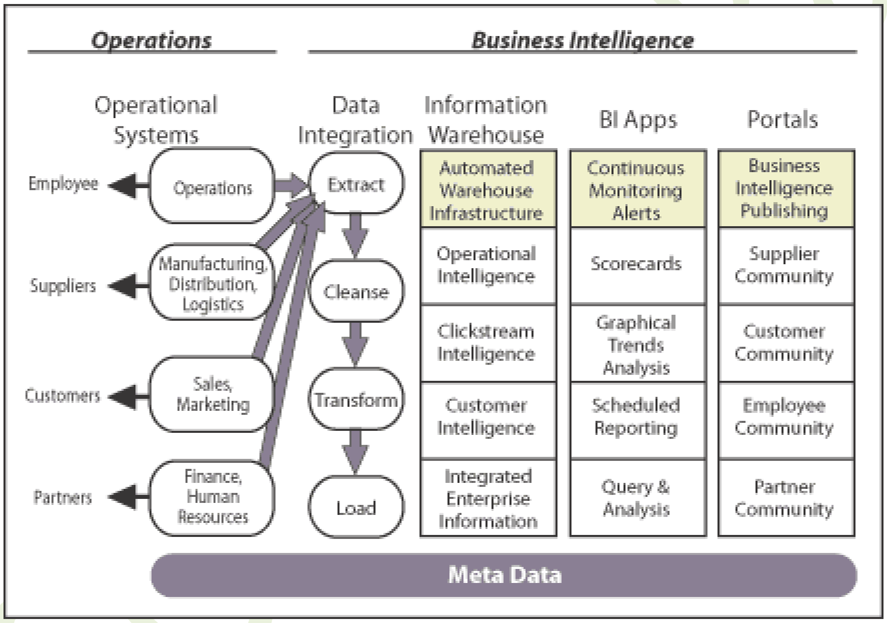


Figure 1: Business Intelligence Infrastructure

Business Benefits

The payback achieved by building the business intelligence infrastructure is a function of how efficiently it operates, how well the infrastructure is supported and enhanced by the business organization as well as its capacity for producing business insight from raw operational data. The business intelligence infrastructure delivers key information to business users. For maximum impact, standards and procedures must be in place to provide key business information proactively. This business intelligence infrastructure enables the organization to unlock the information from the legacy systems, to integrate data across the enterprise and empower business users to become information self- sufficient.

Providing managers and knowledge workers with new tools allowing them to see data in new ways empowers them to make faster and better decisions. Rather than responding to continuous stream of report requests, the business intelligence platform provides business users self-service decision support via the Web or at the desktop. The quantifiable benefits of providing such a business intelligence platform are decisions which increase revenue by identifying and creating up-sell and cross-sell opportunities, improve "valued customer" profitability, decrease costs or expenses by leveraging infrastructure and automating processes, decrease investment in assets such as inventory, or improve productivity with better decision making and faster response-to-market changes or other business events.

The following sections examine more closely each of the layers of the business intelligence infrastructure.

Data Integration. Based on the overall requirements of business intelligence, the data integration layer is required to extract, cleanse and transform data into load files for the information warehouse. This layer begins with transaction-level operational data and meta data about these operational systems. Typically this data integration is done using a relational staging database and utilizing flat file extracts from source systems. The product of a good data staging layer is high-quality data, a reusable infrastructure and meta data supporting both business and technical users. Improved data quality can entail matching against third-party name/address/location databases, merging information from disparate sources into the same information structure and eliminating duplicate, null and outlier values. Building an efficient data integration process is a key component to delivering powerful business intelligence solutions. Often this infrastructure needs to bring data across on a daily basis. In order to accomplish this, the processes need to be efficient and automated. Operator alerts should automatically be generated when exception conditions occur. This layer will generate significant meta data that must be captured and leveraged in the other layers to ensure proper delivery, support and guidance to both system administrators and business intelligence users.

Distinguishing characteristics of business intelligence versus operational systems information are primarily along the lines of purpose, data content, usage and response time requirements. The purpose of operational systems is to support the day-to-day business processes instead of supporting long-term strategic decision-making which is the purpose of business intelligence. Data content for operational systems is current and has real-time values as opposed to historical data that is accurate as of a point in time for business intelligence. Operational data is highly structured and repetitive, whereas business intelligence is highly unstructured, heuristic or analytical. Operation support requires information within seconds, yet business intelligence response time requirements range from seconds to minutes.

Information Warehouse. The information warehouse layer consists of relational and/or OLAP cube services that allow business users to gain insight into their areas of responsibility in the organization. Important in the warehouse design the definition of databases that provide information on confirmed dimensions or business variables that are true across the whole enterprise. The information warehouse is usually developed incrementally over time and is architected to include key business variables and business metrics in a structure that meets all business analysis questions required by the business groups. A common practice is to architect the information warehouse into a sequence of data marts that can be developed within 90 days. The essential responsibility of the first data mart is to build out the conformed dimensions for the enterprise and the business facts for a specific subject area of the business. The infrastructure established during that initial data mart effort can be leveraged in subsequent data mart development efforts.

In order to architect this information warehouse layer correctly, the business requirements and key business questions need to be defined. When this information is available, there will be additional insight into the business derived from the underlying data that cannot be fully anticipated before the data is actually available. Key areas to consider in defining requirements relate to the major functional areas of the organization. There are a few key categories of business intelligence that should be considered: Customer, operational and clickstream intelligence.

Customer Intelligence relates to customer, service, sales and marketing information viewed along time periods, location/geography, product and customer variables. Business decisions that can be supported with customer intelligence range from pricing, forecasting, promotion strategy, competitive analysis to up-sell strategy and customer service resource allocation.

Operational Intelligence relates to finance, operations, manufacturing, distribution, logistics and human resource information viewed along time periods, location/geography, product, project, supplier, carrier and employee. Business decisions that can be supported with operational intelligence include budgeting, investment, pricing, hiring, training, promotion, cost control, scheduling, service levels, defect prevention and capacity planning.

Clickstream Intelligence relates to Web sessions, sales and service information viewed along time periods, products, customer, Web pages and request type. Business decisions that can be supported with clickstream intelligence include website optimization, ad space pricing, promotion communication and up-sell strategy.

Automating the warehouse administration is essential for shortening the cycle time for bringing business intelligence updates into the information warehouse. Features of an automated warehouse administration include operator alerts for exceptions, automated exception handling based on predefined business rules, DBA alerts for key service outages and a production scheduled of tasks run according to business requirements.

BI Applications. The most visible layer of the business intelligence infrastructure is the applications layer which delivers the information to business users. Business intelligence requirements include scheduled report generation and distribution, query and analysis capabilities to pursue special investigations and graphical analysis permitting trend identification. This layer should enable business users to interact with the information to gain new insight into the underlying business variables to support business decisions. Another important application is the balanced scorecard that displays key performance indicator current values and the targets for financial, customer, internal systems and human capital categories. The balance scorecard is a summary of key business analytics rolled up to the appropriate level for the user with capabilities to drill down into more detail. This detail relates to operational, customer and clickstream intelligence described in the previous section.

In order to achieve maximum velocity of business intelligence, continuous monitoring processes should be in place to trigger alerts to business decision-makers, accelerating action toward resolving problems or compensating for unforeseen business events. This proactive nature of business intelligence can provide tremendous business benefits.

Portals. Presenting business intelligence on the Web through a portal is gaining considerable momentum. Web-based portals are becoming commonplace as a single personalized point of access for key business information. All major BI vendors have developed components which snap into the popular portal infrastructure. Portals are usually organized by communities of users organized for suppliers, customers, employers and partners. Portals can reduce the overall infrastructure costs of an organization as well as deliver great self-service and information access capabilities.

Organizational Requirements. There are several organizational considerations that contribute to an efficient business intelligence infrastructure. First is to have a core business intelligence implementation and support team dedicated to the principles of optimizing the business intelligence infrastructure. This core team minimally consists of the following roles:

- BI architect responsible for the overall design and implementation of the business intelligence infrastructure with special focus on the information architecture in the information warehouse layer
- ETL developer responsible for the design and development of the data staging layer
- BI analyst responsible for identifying the key business intelligence questions of the business decision-makers and the key requirements of the BI applications layer
- Database administrator responsible for the physical implementation and support of the information warehouse layer
- Business content managers responsible for delivering the required information to various user communities of the portal layer

Additionally, there needs to be great organizational support for business intelligence across the subject areas comprising the business intelligence infrastructure. This is facilitated by a business intelligence steering committee comprised of the technical managers and business sponsors. This committee ensures that the risks of the business intelligence projects are mitigated with adequate scope control and good communication to the business sponsors.

Finally, there should be objective measures in place to track the effectiveness of the business intelligence architecture. These measures should include areas such as business intelligence usage, ratings of business and technical meta data available, service level metrics, business impacts metrics and efficiency improvement metrics.

BUSINESS INTELLIGENCE TECHNOLOGY

Business intelligence provides organizational data in such a way that the organizational knowledge filters can easily associate with this data and turn it into information for the organization. Persons involved in business intelligence processes may use application software and other technologies to gather, store, analyze, and provide access to data, and present that data in a simple, useful manner. The software aids in Business performance management, and aims to help people make “better" business decisions by making accurate, current, and relevant information available to them when they need it. Some businesses use data warehouses because they are a logical collection of information gathered from various operational databases for the purpose of creating business intelligence.

In order for BI system to work effectively there must be some technical constraints in place. BI technical requirements have to address the following issues:

- Security and specified user access to the warehouse
- Data volume (capacity)
- How long data will be stored (data retention)
- Benchmark and performance targets

People working in business intelligence have developed tools that ease the work, especially when the intelligence task involves gathering and analyzing large quantities of unstructured data. Each vendor typically defines Business Intelligence their own way, and markets tools to do BI the way that they see it. Business intelligence includes tools in various categories, including the following:

- AQL - Associative Query Logic
- Score carding
- Business Performance Management and Performance Measurement
- Business Planning
- Business Process Re-engineering
- Competitive Analysis

- Customer Relationship Management (CRM) and Marketing
- Data mining (DM), Data Farming, and Data warehouses
- Decision Support Systems (DSS) and Forecasting
- Document warehouses and Document Management
- Enterprise Management systems
- Executive Information Systems (EIS)
- Finance and Budgeting
- Human Resources
- Knowledge Management
- Mapping, Information visualization, and Dash boarding
- Management Information Systems(MIS)
- Geographic Information Systems (GIS)
- Online Analytical Processing (OLAP) and multidimensional analysis; sometimes simply called "Analytics" (based on the so-called "hypercube" or "cube")
- Real time business intelligence
- Statistics and Technical Data Analysis
- Supply Chain Management/Demand Chain Management
- Systems intelligence
- Trend Analysis
- User/End-user Query and Reporting
- Web Personalization and Web Mining
- Text mining

BI often uses Key performance indicators (KPIs) to assess the present state of business and to prescribe a course of action. More and more organizations have started to make more data available more promptly. In the past, data only became available after a month or two, which did not help managers to adjust activities in time to hit Wall Street targets. Recently, banks have tried to make data available at shorter intervals and have reduced delays.

For example, for businesses which have higher operational/credit risk loading (for example, credit cards and "wealth management"), a large multi-national bank makes KPI-related data available weekly, and sometimes offers a daily analysis of numbers. This means data usually becomes available within 24 hours, necessitating automation and the use of IT systems.

BENEFITS OF BI

BI provides many benefits to companies utilizing it. It can eliminate a lot of the guesswork within an organization, enhance communication among departments while coordinating activities, and enable companies to respond quickly to changes in financial conditions, customer preferences, and supply chain operations. BI improves the overall performance of the company using it.

Information is often regarded as the second most important resource a company has (a company's most valuable assets are its people). So when a company can make decisions based on timely and accurate information, the company can improve its performance. BI also expedites decision-making, as acting quickly and correctly on information before competing businesses do can often result in competitively superior performance. It can also improve customer experience, allowing for the timely and appropriate response to customer problems and priorities.

The firms have recognized the importance of business intelligence for the masses has arrived. Some of them are listed below.

- With BI superior tools, now employees can also easily convert their business knowledge via the analytical intelligence to solve many business issues, like increase response rates from direct mail, telephone, e-mail, and Internet delivered marketing campaigns.
- With BI, firms can identify their most profitable customers and the underlying reasons for those customers' loyalty, as well as identify future customers with comparable if not greater potential.
- Analyze click-stream data to improve e-commerce strategies
- Quickly detect warranty -reported problems to minimize the impact of product design deficiencies.
- Discover money-laundering criminal activities.
- Analyze potential growth customer profitability and reduce risk exposure through more accurate financial credit scoring of their customers
- Determine what combinations of products and service lines customers are likely to purchase and when.
- Analyze clinical trials for experimental drugs.
- Set more profitable rates for insurance premiums.
- Reduce equipment downtime by applying predictive maintenance.
- Determine with attrition and churn analysis why customers leave for competitors and/or become the customers.
- Detect and deter fraudulent behavior, such as from usage spikes when credit or phone cards are stolen.
- Identify promising new molecular drug compounds

Customers are the most critical aspect to a company's success. Without them a company cannot exist. So it is very important that firms have information on their preferences. Firms must quickly adapt to their changing demands. Business Intelligence enables firms to gather information on the trends in the marketplace and come up with innovative products or services in anticipation of customer's changing demands.

Competitors can be a huge hurdle on firm's way to success. Their objectives are the same as firms' and that is to maximize profits and customer satisfaction. In order to be successful firms must stay one step ahead of the competitors. In business we don't want to play the catch up game because we would have lost valuable market share. Business Intelligence tells what actions our competitors are taking, so one can make better informed decisions

UNIT-5

Unit V: Basics of Data Integration: Concepts of data integration need and advantages of using data integration, introduction to common data integration approaches, data integration technologies, Introduction to data quality, data profiling concepts and applications, the multidimensional data model , star and snowflake schema.

Definition - What does Data Integration

Data integration is a process in which heterogeneous data is retrieved and combined as an incorporated form and structure. Data integration allows different data types (such as data sets, documents and tables) to be merged by users, organizations and applications, for use as personal or business processes and/or functions.

Data integration involves combining data from several disparate sources, which are stored using various technologies and provide a unified view of the data. Data integration becomes increasingly important in cases of merging systems of two companies or consolidating applications within one company to provide a unified view of the company's data assets. The later initiative is often called a data warehouse.

Probably the most well known implementation of data integration is building an enterprise's data warehouse. The benefit of a data warehouse enables a business to perform analyses based on the data in the data warehouse. This would not be possible to do on the data available only in the source system. The reason is that the source systems may not contain corresponding data, even though the data are identically named, they may refer to different entities.

Data Integration Areas

Data integration is a term covering several distinct sub-areas such as:

- Data warehousing
- Data migration
- Enterprise application/information integration
- Master data management

Challenges of Data Integration

At first glance, the biggest challenge is the technical implementation of integrating data from disparate often incompatible sources. However, a much bigger challenge lies in the entirety of data integration. It has to include the following phases:

Design

The data integration initiative within a company must be an initiative of business, not IT. There should be a champion who understands the data assets of the enterprise and will be able to lead the discussion about the long-term data integration initiative in order to make it consistent, successful and benefitial.

Analysis of the requirements (BRS), i.e. why is the data integration being done, what are the objectives and deliverables. From what systems will the data be sourced? Is all the data available to fulfill the requirements? What are the business rules? What is the support model and SLA?

Analysis of the source systems, i.e. what are the options of extracting the data from the systems (update notification, incremental extracts, full extracts), what is the required/available frequency of the extracts? What is the quality of the data? Are the required data fields populated properly and consistently? Is the documentation available? What are the data volumes being processed? Who is the system owner?

Any other non-functional requirements such as data processing window, system response time, estimated number of (concurrent) users, data security policy, backup policy.

What is the support model for the new system? What are the SLA requirements?

And last but not least, who will be the owner of the system and what is the funding of the maintenance and upgrade expenses?

The results of the above steps need to be documented in form of SRS document, confirmed and signed-off by all parties which will be participating in the data integration project.

Implementation

Based on the BRS and SRS, a feasibility study should be performed to select the tools to implement the data integration system. Small companies and enterprises which are starting with data warehousing are faced with making a decision about the set of tools they will need to implement the solution. The larger enterprise or the enterprises which already have started other projects of data integration are in an easier position as they already have experience and can extend the existing system and exploit the existing knowledge to implement the system more effectively. There are cases, however, when using a new, better suited platform or technology makes a system more effective compared to staying with existing company standards. For example, finding a more suitable tool which provides better scaling for future growth/expansion, a solution that lowers the implementation/support cost, lowering the license costs, migrating the system to a new/modern platform, etc.

Testing

Along with the implementation, the proper testing is a must to ensure that the unified data are correct, complete and up-to-date.

Both technical IT and business needs to participate in the testing to ensure that the results are as expected/required. Therefore, the testing should incorporate at least Performance Stress test (PST), Technical Acceptance Testing (TAT) and User Acceptance Testing (UAT) PST, TAT (Technical Acceptance Testing), UAT (User Acceptance Testing).

Data Integration Techniques

There are several organizational levels on which the integration can be performed. As we go down the level of automated integration increases.

Manual Integration or Common User Interface - users operate with all the relevant information accessing all the source systems or web page interface. No unified view of the data exists.

Application Based Integration - requires the particular applications to implement all the integration efforts. This approach is manageable only in case of very limited number of applications.

Middleware Data Integration - transfers the integration logic from particular applications to a new middleware layer. Although the integration logic is not implemented in the applications anymore, there is still a need for the applications to partially participate in the data integration.

Uniform Data Access or Virtual Integration - leaves data in the source systems and defines a set of views to provide and access the unified view to the customer across whole enterprise. For example, when a user accesses the customer information, the particular details of the customer are transparently acquired from the respective system. The main benefits of the virtual integration are nearly zero latency of the data updates propagation from the source system to the consolidated view, no need for separate store for the consolidated data. However, the drawbacks include limited possibility of data's history and version management, limitation to apply the method only to 'similar' data sources (e.g. same type of database) and the fact that the access to the user data generates extra load on the source systems which may not have been designed to accommodate.

Common Data Storage or Physical Data Integration - usually means creating a new system which keeps a copy of the data from the source systems to store and manage it independently of the original system. The most well know example of this approach is called Data Warehouse (DW). The benefits comprise data version management, combining data from very different sources (mainframes, databases, flat files, etc.). The physical integration, however, requires a separate system to handle the vast volumes of data.

Need for Data Integration

- It is done for providing data in a specific view as requested by users, applications, etc.
- The bigger the organization gets, the more data there is and the more data needs integration.
- Increases with the need for data sharing.

Advantages of Using Data Integration

- Benefit to decision - makers, who have access to important information from past studies
- Reduces cost, overlaps and redundancies; reduces exposure to risks
- Helps to monitor key variables like trends and consumer behaviour, etc.

Approaches to Integration

In this section, we apply an architectural perspective to give an overview of the different ways to address the integration problem. The presented classification is based on [12] and distinguishes integration approaches according to the level of abstraction where integration is performed. Information systems can be described using a layered architecture, as shown in Figure 1.1: On the topmost layer, users access data and services through various interfaces that run on top of different applications. Applications may use middleware transaction processing (TP) monitors, message-oriented middleware (MOM), SQL-middleware, etc. to access data via a data access layer. The data itself is managed by a data storage system. Usually, database management systems (DBMS) are used to combine the data access and storage layer. In general, the integration problem can be addressed on each of the presented system layers. For this, the following principal approaches as illustrated in Figure 1.1 — are available:

Manual Integration

Here, users directly interact with all relevant information systems and manually integrate selected data. That is, users have to deal with different user interfaces and query languages. Additionally, users need to have detailed knowledge on location, logical data representation, and data semantics.

Common User Interface

In this case, the user is supplied with a common user interface (e.g., a web browser) that provides a uniform look and feel. Data from relevant information systems is still separately presented so that homogenization and integration of data yet has to be done by the users (for instance, as in search engines).

Integration by Applications

This approach uses integration applications that access various data sources and return integrated results to the user. This solution is practical for a small number of component systems. However, applications become increasingly fat as the number of system interfaces and data formats to homogenize and integrate grows.

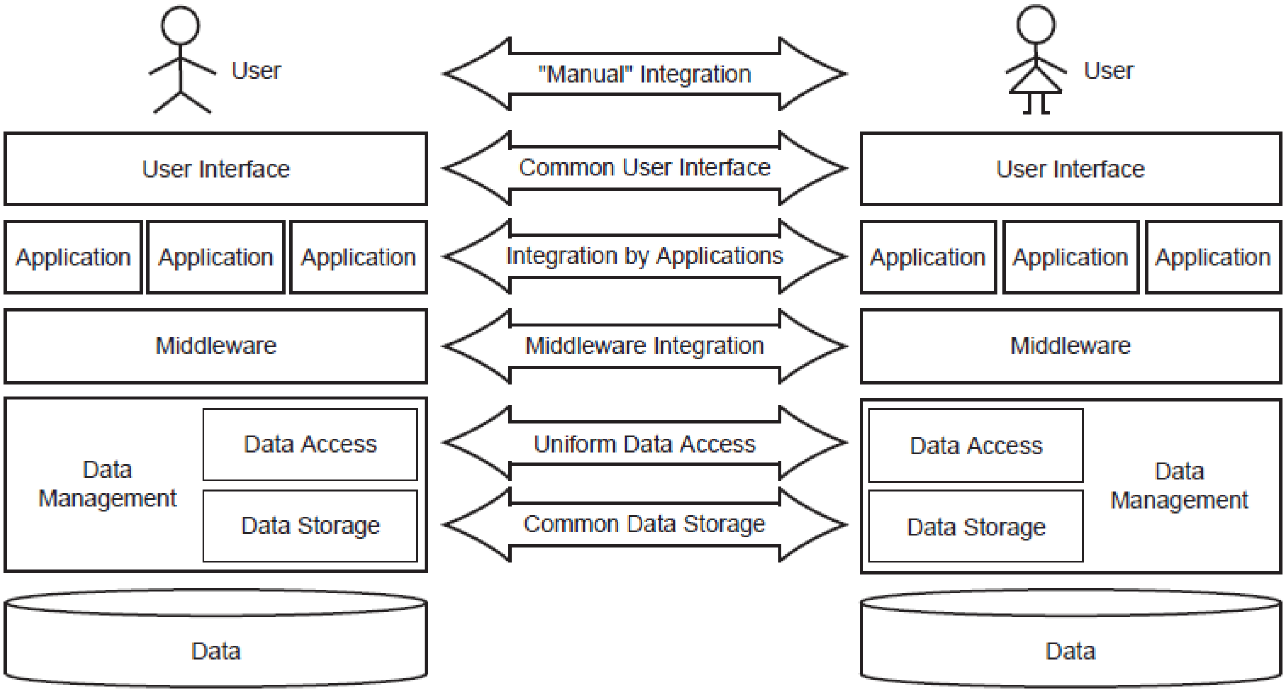


Fig. 1.1. General Integration Approaches on Different Architectural Levels

Integration by Middleware

Middleware provides reusable functionality that is generally used to solve dedicated aspects of the integration problem, e.g., as done by SQL-middleware. While applications are relieved from implementing common integration functionality, integration efforts are still needed in applications. Additionally, different middleware tools usually have to be combined to build integrated systems.

Uniform Data Access

In this case, a logical integration of data is accomplished at the data access level. Global applications are provided with a unified global view of physically distributed data, though only virtual data is available on this level. Local information systems keep their autonomy and can support additional data access layers for other applications. However, global provision of physically integrated data can be time-consuming since data access, homogenization, and integration have to be done at runtime.

Common Data Storage

Here, physical data integration is performed by transferring data to a new data storage; local sources can either be retired or remain operational. In general, physical data integration provides fast data access. However, if local data sources are retired, applications that access them have to be migrated to the new data storage as well. In case local data sources remain operational, periodical refreshing of the common data storage needs to be considered

In practice, concrete integration solutions are realized based on the presented six general integration approaches. Important examples include:

- Mediated query systems represent a uniform data access solution by pro-viding a single point for read-only querying access to various data sources, e.g., as in TSIMMIS. A mediator that contains a global query processor is employed to send sub queries to local data sources; returned local query results are then combined.

- Portals as another form of uniform data access are personalized doorways to the internet or intranet where each user is provided with information according to his detected information needs. Usually, web mining is applied to determine user-profiles by click stream analysis; thereby, information the user might be interested in can be retrieved and presented.

- Data warehouses realize a common data storage approach to integration. Data from several operational sources (on-line transaction processing systems, OLTP) are extracted, transformed, and loaded (ETL) into a data warehouse. Then, analysis, such as online analytical processing (OLAP), can be performed on cubes of integrated and aggregated data.

- Operational data stores are a second example of a common data storage. Here, a “warehouse with fresh data” is built by immediately propagating updates in local data sources to the data store. Thus, up-to-date integrated data is available for decision support. Unlike in data warehouses, data is neither cleansed nor aggregated nor are data histories supported.

- Federated database systems (FDBMS) achieve a uniform data access solution by logically integrating data from underlying local DBMS. Federated database systems are fully-fledged DBMS; that is, they implement their own data model, support global queries, global transactions, and global access control. Usually, the five-level reference architecture by is employed for building FDBMS.

- Workflow management systems (WFMS) allow to implement business processes where each single step is executed by a different application or user. Generally, WFMS support modeling, execution, and maintenance of processes that are comprised of interactions between applications and human users. WFMS represent an integration-by-application approach.

- Integration by web services performs integration through software components (i.e., web services) that support machine-to-machine interaction over a network by XML-based messages that are conveyed by internet protocols. Depending on their offered integration functionality, web services either represent a uniform data access approach or a common data access interface for later manual or application-based integration.
- Model management introduces high-level operations between models (such as database schemas, UML models, and software configurations) and model mappings; such operations include matching, merging, selection, and composition. Using a schema algebra that encompasses all these operations, it is intended to reduce the amount of hand-crafted code required for transformations of models and mappings as needed for schema integration. Model management falls into the category of manual integration.

- Peer-to-peer (P2P) integration is a decentralized approach to integration between distributed, autonomous peers where data can be mutually shared and integrated through mappings between local schemas of peers. P2P integration constitutes, depending on the provided integration functionality, either a uniform data access approach or a data access interface for sub-sequent manual or application-based integration.

- Grid data integration provides the basis for hypotheses testing and pattern detection in large amounts of data in grid environments, i.e., interconnected computing resources being used for high-throughput computing. Here, often unpredictable and highly dynamic amounts of data have to be dealt with to provide an integrated view over large (scientific) data sets. Grid data integration represents an integration by middleware approach.

- Personal data integration systems (e.g., are a special form of manual integration. Here, tailored integrated views are defined (e.g., by a declarative integration language), either by users themselves or by dedicated integration engineers. Each integrated view precisely matches the information needs of a user by encompassing all relevant entities with real-world semantics as intended by the particular user; thereby, the integrated view reflects the user's personal way to perceive his application domain of interest.

- Collaborative integration (e.g.,), another special form of manual integration, is based on the idea to have users to contribute to a data integration system for using it. Here, initial partial schema mappings are presented to users who answer questions concerning the mappings; these answers are then taken to refine the mappings and to expand the system capabilities. Similar to folksonomies, where data is collaboratively labeled for later retrieval, the task of schema mapping is distributed over participating users.

- In Data space systems, co-existence of all data (i.e., both structured and unstructured) is propagated rather than full integration. A data space system is used to provide the same basic functionality, e.g., search facilities, over all data sources independently of their degree of integration. Only when more sophisticated services are needed, such as relational-style queries, additional efforts are made to integrate the required data sources more closely. In general, data space systems may simultaneously use every one of the presented six general integration approaches.

Data Integration tools

- Alteryx
- Analytics Canvas
- Cloud Elements API Integration
- DataWatch
- Denodo Platform
- elastic.io Integration Platform
- HiperFabric
- Lavastorm
- Informatica Platform (www.informatica.com)
- Oracle Data Integration Services
- ParseKit (enigma.io)
- Paxata
- RapidMiner Studio
- Red Hat JBoss Data Virtualization. Community project: teiid.
- Azure Data Factory (ADF)
- SQL Server Integration Services (SSIS)
- TMMData
- Data Ladder

Data quality

Data quality refers to the level of quality of data. There are many definitions of data quality but data are generally considered high quality if "they are fit for their intended uses in operations, decision making and planning." (Tom Redman<Redman, T.C. (2008). Data driven: Profiting from your most important business asset. Boston, Mass.: Harvard Business Press.>). Alternatively, data is deemed of high quality if it correctly represents the real-world construct to which it refers. Furthermore, apart from these definitions, as data volume increases, the question of internal consistency within data becomes significant, regardless of fitness for use for any particular external purpose. People's views on data quality can often be in disagreement, even when discussing the same set of data used for the same purpose.

There are a number of theoretical frameworks for understanding data quality. A systems-theoretical approach influenced by American pragmatism expands the definition of data quality to include information quality, and emphasizes the inclusiveness of the fundamental dimensions of accuracy and precision on the basis of the theory of science (Ivanov, 1972). One framework, dubbed "Zero Defect Data" (Hansen, 1991) adapts the principles of statistical process control to data quality. Another framework seeks to integrate the product perspective (conformance to specifications) and the service perspective (meeting consumers' expectations) (Kahn et al. 2002). Another framework is based in semiotics to evaluate the quality of the form, meaning and use of the data (Price and Shanks, 2004). One highly theoretical approach analyzes the ontological nature of information systems to define data quality rigorously (Wand and Wang, 1996).

A considerable amount of data quality research involves investigating and describing various categories of desirable attributes (or dimensions) of data. These dimensions commonly include accuracy, correctness, currency, completeness and relevance. Nearly 200 such terms have been identified and there is little agreement in their nature (are these concepts, goals or criteria?), their definitions or measures (Wang et al., 1993). Software engineers may recognize this as a similar problem to "ilities".

MIT has a Total Data Quality Management program, led by Professor Richard Wang, which produces a large number of publications and hosts a significant international conference in this field (International Conference on Information Quality, ICIQ). This program grew out of the work done by Hansen on the "Zero Defect Data" framework (Hansen, 1991).

In practice, data quality is a concern for professionals involved with a wide range of information systems, ranging from data warehousing and business intelligence to customer relationship management and supply chain management. One industry study estimated the total cost to the U.S. economy of data quality problems at over U.S. $600 billion per annum (Eckerson, 2002). Incorrect data – which includes invalid and outdated information – can originate from different data sources – through data entry, or data migration and conversion projects.

In 2002, the USPS and PricewaterhouseCoopers released a report stating that 23.6 percent of all U.S. mail sent is incorrectly addressed.

One reason contact data becomes stale very quickly in the average database – more than 45 million Americans change their address every year.

In fact, the problem is such a concern that companies are beginning to set up a data governance team whose sole role in the corporation is to be responsible for data quality. In some organizations, this data governance function has been established as part of a larger Regulatory Compliance function - a recognition of the importance of Data/Information Quality to organizations.

Problems with data quality don't only arise from incorrect data; inconsistent data is a problem as well. Eliminating data shadow systems and centralizing data in a warehouse is one of the initiatives a company can take to ensure data consistency.

Enterprises, scientists, and researchers are starting to participate within data curation communities to improve the quality of their common data.

The market is going some way to providing data quality assurance. A number of vendors make tools for analyzing and repairing poor quality data in situ," service providers can clean the data on a contract basis and consultants can advise on fixing processes or systems to avoid data quality problems in the first place. Most data quality tools offer a series of tools for improving data, which may include some or all of the following:

1. Data profiling - initially assessing the data to understand its quality challenges

1. Data standardization - a business rules engine that ensures that data conforms to quality rules
2. Geocoding - for name and address data. Corrects data to U.S. and Worldwide postal standards
3. Matching or Linking - a way to compare data so that similar, but slightly different records can be aligned. Matching may use "fuzzy logic" to find duplicates in the data. It often recognizes that "Bob" and "Robert" may be the same individual. It might be able to manage "householding", or finding links between spouses at the same address, for example. Finally, it often can build a "best of breed" record, taking the best components from multiple data sources and building a single super-record.
4. Monitoring - keeping track of data quality over time and reporting variations in the quality of data. Software can also auto-correct the variations based on pre-defined business rules.

5. Batch and Real time - Once the data is initially cleansed (batch), companies often want to build the processes into enterprise applications to keep it clean.

There are several well-known authors and self-styled experts, with Larry English perhaps the most popular guru. In addition, IQ International - the International Association for Information and Data Quality was established in 2004 to provide a focal point for professionals and researchers in this field.

ISO 8000 is an international standard for data quality.

Definitions

This list is taken from the online book "Data Quality: High-impact Strategies". See also the glossary of data quality terms.

- Degree of excellence exhibited by the data in relation to the portrayal of the actual scenario.
- The state of completeness, validity, consistency, timeliness and accuracy that makes data appropriate for a specific use.
- The totality of features and characteristics of data that bears on its ability to satisfy a given purpose; the sum of the degrees of excellence for factors related to data.
- The processes and technologies involved in ensuring the conformance of data values to business requirements and acceptance criteria.
- Complete, standards based, consistent, accurate and time stamped.

If the ISO 9000:2015 definition of quality is applied, data quality can be defined as the degree to which a set of characteristics of data fulfills requirements. Examples of characteristics are: completeness, validity, accuracy, consistency, availability and timeliness. Requirements are defined as the need or expectation that is stated, generally implied or obligatory.

What is data quality?

Data quality is a perception or an assessment of data’s fitness to serve its purpose in a given context.

It is described by several dimensions like

- Correctness / Accuracy: Accuracy of data is the degree to which the captured data correctly describes the real world entity.

- Consistency: This is about the single version of truth. Consistency means data throughout the enterprise should be sync with each other.

- Completeness: It is the extent to which the expected attributes of data are provided.

- Timeliness: Right data to the right person at the right time is important for business.

- Metadata: Data about data.

Maintenance of data quality

- Data quality results from the process of going through the data and scrubbing it, standardizing it, and de duplicating records, as well as doing some of the data enrichment.

- 1. Maintain complete data.
- 2. Clean up your data by standardizing it using rules.
- 3. Use fancy algorithms to detect duplicates. Eg: ICS and Informatics Computer System.
- 4. Avoid entry of duplicate leads and contacts.
- 5. Merge existing duplicate records.
- 6. Use roles for security.

What is data profiling?

It is the process of statistically examining and analyzing the content in a data source, and hence collecting information about the data. It consists of techniques used to analyze the data we have for accuracy and completeness.

1. Data profiling helps us make a thorough assessment of data quality.

2. It assists the discovery of anomalies in data.

3. It helps us understand content, structure, relationships, etc. about the data in the data source we are analyzing.

4. It helps us know whether the existing data can be applied to other areas or purposes.

5. It helps us understand the various issues/challenges we may face in a database project much before the actual work begins. This enables us to make early decisions and act accordingly.

6. It is also used to assess and validate metadata.

Data profiling is the process of examining the data available in an existing information data source (e.g. a database or a file) and collecting statistics or small but informative summaries about that data.[1] The purpose of these statistics may be to:

1. Find out whether existing data can easily be used for other purposes
2. Improve the ability to search the data by tagging it with keywords, descriptions, or assigning it to a category
3. Give metrics on data quality including whether the data conforms to particular standards or patterns
4. Assess the risk involved in integrating data for new applications, including the challenges of joins
5. Discover metadata of the source database, including value patterns and distributions, key candidates, foreign-key candidates, and functional dependencies
6. Assess whether known metadata accurately describes the actual values in the source database
7. Understanding data challenges early in any data intensive project, so that late project surprises are avoided. Finding data problems late in the project can lead to delays and cost overruns.
8. Have an enterprise view of all data, for uses such as master data management where key data is needed, or data governance for improving data quality.

How to conduct Data Profiling?

Data profiling involves statistical analysis of the data at source and the data being loaded, as well as analysis of metadata. These statistics may be used for various analysis purposes. Common examples of analyses to be done are:

Data quality: Analyze the quality of data at the data source.

NULL values: Look out for the number of NULL values in an attribute.

Candidate keys: Analysis of the extent to which certain columns are distinct will give developer useful information w. r. t. selection of candidate keys.

Primary key selection: To check whether the candidate key column does not violate the basic requirements of not having NULL values or duplicate values.

Empty string values: A string column may contain NULL or even empty sting values that may create problems later.

String length: An analysis of largest and shortest possible length as well as the average string length of a sting-type column can help us decide what data type would be most suitable for the said column.

Identification of cardinality: The cardinality relationships are important for inner and outer join considerations with regard to several BI tools.

Data format: Sometimes, the format in which certain data is written in some columns may or may not be user-friendly.

Common Data Profiling Software

Most of the data-integration/analysis soft-wares have data profiling built into them. Alternatively, various independent data profiling tools are also available. Some popular ones are:

- Trillium Enterprise Data quality
- Datiris Profiler
- Talend Data Profiler
- IBM Infosphere Information Analyzer
- SSIS Data Profiling Task
- Oracle Warehouse Builder

Benefits

The benefits of data profiling are to improve data quality, shorten the implementation cycle of major projects, and improve understanding of data for users. Discovering business knowledge embedded in data itself is one of the significant benefits derived from data profiling. Data profiling is one of the most effective technologies for improving data accuracy in corporate databases.

Although data profiling is effective and useful for each sector of our daily life, it can be challenging not to slip into "analysis paralysis".

Multidimensional data model, star and snowflake schema. Refer to PDF no 2

Unit VI: BI Project Lifecycle: Typical BI Project Lifecycle, Requirements Gathering and Analysis - Functional and Non- Functional Requirements, Testing in a BI Project, BI Project Deployment, Post Production Support.

An Introduction to Business Intelligence Lifecycle Management

It's no secret that Business Intelligence* (BI) projects are both time consuming and resource intensive, often suffer from poor communication between the business and IT, and are usually inflexible to changes once development has started. This is due, in large part, to the method by which BI projects are traditionally implemented. Regardless of the methodology you employ, at the very least, a successful BI iteration requires:

- Business requirements identification
- Data analysis
- Data architecture and modeling
- Data integration (ETL, ELT, data virtualization, etc)
- Front-end development
- Testing and release management.

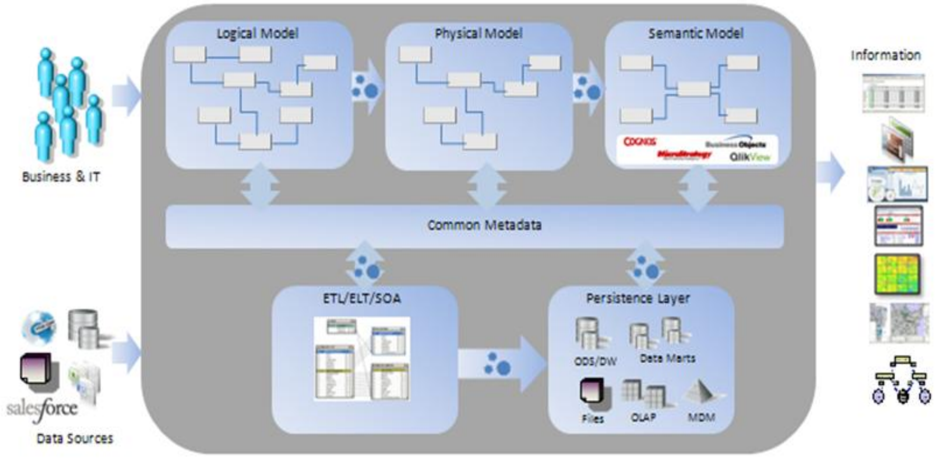
Whether you choose to integrate testing with development or employ prototypes and sprints, it doesn't diminish the fact that design, development and testing are part of the business intelligence lifecycle process. The challenge with the way in which BI projects are usually implemented is that the design and development steps across the architecture aren't integrated and insulated from direct input by the business. In addition, since the tools we usually employed for design and development aren't integrated, both initial development and subsequent changes require a significant level of manual effort and change management. If we want to improve upon the traditional BI development process, we need to start approaching this process as a business driven lifecycle, as well as integrate and automate as much of the development process as possible.

*Note: In this context, BI refers to both the data integration layer as well as the front-end tools for reporting, analyzing and disseminating the information.

Business intelligence lifecycle management is a design, development and management approach to BI that incorporates business users into the design process and focuses on generating the data models, database objects, data integration mappings and front-end semantic layers directly from the business user input. The challenge is that traditional BI projects leverage multiple, disparate tools are used throughout the BI architecture to capture requirements, document business rules, perform source system analysis, model the data warehouse and transform, store, report and analyze the target data. In many environments, metadata management applications, data quality tools and advanced analytic applications are also employed. Rarely do these tools share metadata, which makes it challenging to automate development and difficult to determine the impact when changes are required. In addition, business input is indirect, since it must be captured, disseminated and often re-communicated to the development team.

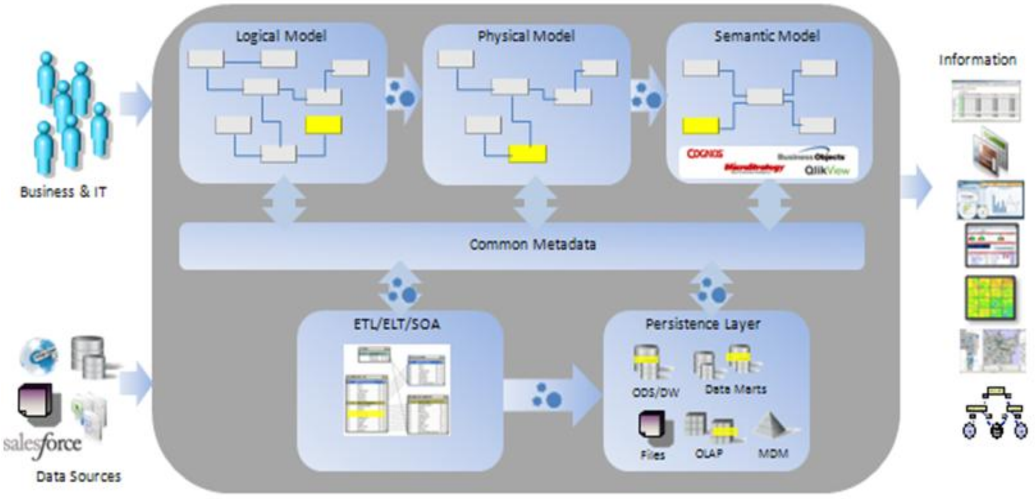
Fortunately, there are a number of business intelligence lifecycle tools in the market that facilitate this approach. With BI lifecycle management tools, users are given the ability to input project requirements, logical entities, relationships, business rules, source attributes, target attributes and business metadata. These act as inputs to the logical model which can be defined and then reviewed by the business. Once approved, a physical model is generated from the logical model, the database objects are generated automatically from the physical model and the data integration (i.e. ETL/ELT/SOA) mappings are created along with the objects in the persistence layer. Some of the tools on the market also generate the semantic models for front-end tool sets from the metadata captured during logical data modeling. Each step in the process may be reviewed by the business and IT through a workflow process, and dependencies ensure that a change is reflected across the architecture.

Figure 1: The Business Intelligence Lifecycle Process



The consistency between layers of the architecture is provided by the common metadata layer. Consider it an inventory of BI architecture assets with defined relationships. Since all the attributes, business rules, physical objects, etc, are being captured, the dependencies can be identified and traced. This enables change impact analysis and facilitates the testing process. The common metadata layer also creates a linkage between all the layers of the architecture so that each layer can be generated dynamically for rapid prototyping, which provides context to the users throughout a BI implementation. In the below example, if a change is required to one entity in your logical model (in yellow), you could also see what objects would require changes across the entire architecture. In addition, with workflow and version management, changes can be controlled, tracked, prototyped, implemented and rolled-back if necessary. You could also view changes to the entire environment over time. Workflow automation ensures that the changes are approved at each step along the way.

Figure 2: Business Intelligence Lifecycle Change Impact Analysis



The business community has long complained that BI takes too long, is too IT focused and doesn't respond easily to change. If BI is to become more agile , then we need to start approaching the implementation process as an integrated business intelligence lifecycle rather than loosely connected steps. While most business intelligence lifecycle tools are still immature (i.e. only addressing a few areas of the complete process) they provide a glimpse into what's possible. However, you don't need to purchase a business intelligence lifecycle tool to get started. BI projects and programs can incrementally add business intelligence lifecycle capabilities into their process, utilizing their existing tools. For example, many ETL applications support mapping definitions that can be generated from MS Excel spreadsheet templates, which can be populated by a data modeling tool. While this doesn't address the entire business intelligence lifecycle, it does help you to begin to view the design and development process as a business driven, integrated lifecycle and begin to change the way in which you implement BI. In a subsequent article, I'll discuss the steps in the BI lifecycle process in more detail as well as how these steps can be integrated across both the design and development of the BI architecture.

Requirements Gathering and Analysis: The Second Step in Software Development

Once the visioning phase is complete, it is time to gather and analyze requirements for your software project, system or whatever else you are planning. At this point, it is often useful to identify a smaller subset of the stakeholder group who can work intensively with you on this (although all stakeholders can be included, if the group is small). This is also a good time to bring together stakeholders and key members of the development and implementation team.

Generally, requirements gathering and analysis happen over a series of meetings (at least two). I find it best to do it freeform and capture ideas on whiteboards; more formal documentation can follow later, or documentation can remain informal if the project doesn't require it.

Remember that this is only the first and most intensive stage of requirements gathering. Ideally, requirements should be continuously collected, documented, analyzed and refined throughout the development process.

These are the general goals for the first couple of requirements meetings:

- Create a context diagram, which is usually a rough drawing of the system and its context, based on previous goal-setting with the stakeholders.
- Define actors and their goals. Actors are the primary users of the system. Ask what specific activities are each actor trying to achieve?
- List initial use cases in brief narrative. Ask how each actor will achieve each goal, but don't try to get too detailed at this point.
- Begin capturing terms to include in the glossary as they come in up in the discussion.
- Identify business rules that provide constraints on the system (based on the givens identified during the vision stage plus others subsequently identified). Ask what we can and can't do according to our organizational policies, laws of the land, requirements of our funders and so forth.
- Identify nonfunctional requirements (based on the givens, high-level goals and others subsequently identified). These requirements won't be captured in the use cases but are important to document. They might include security, technology, system integration, localization, reliability and similar requirements.
- Identify working group members from the stakeholder group to call upon for testing and feedback as development progresses.

As requirements gathering progresses, drill down into detail on the requirements and document them thoroughly. The following goals may be accomplished in subsequent meetings or through a virtual collaborative space where stakeholders and members of the development team can post, read and comment on documentation.

- Write the initial set of high-priority use cases in full form.
- Document functional requirements, or inner operations of the system (calculations, technical details, data manipulation and processing, others) that will satisfy the use cases.
- Organize use cases and requirements into packages (or modules) to help guide development and figure out what should be tackled first.
- Create a project plan that maps out generally when each piece of functionality (or module) will be completed but that can be modified as the project progresses.
- Begin a project control list for documenting new features and areas of redesign and determining if and when they should be implemented as they come up during development.
- Begin an issue list (or bug tracker) for tracking problems and questions that arise and may need to be addressed by stakeholders or in a later development cycle. The issue list is distinguished from the project control list in that a feature or solution has not been determined yet to address the issue; once that is determined, the issue is closed and the item is added to the project control list.
- Create a well-organized project knowledge base with all documentation, including other pieces that may be needed at this point (workflow diagram, database design, network design, prototype, user interface mockup, etc.) that is accessible to everyone working on the project.
- Present the use cases and other relevant documentation to the working group or all stakeholders and receive sign-off to proceed with development.

Requirements gathering and Analysis

1. A requirement is something the product must do or a quality that the product must have
2. Users may not be aware of all requirements
3. Users may voice a perceived need
4. But users do not mention some requirements
    1. assume that the requirement is obvious
    2. good interviews, observation will help to reveal these
    3. some only surface when models are constructed or prototypes are reviewed
5. users also may not appreciate technical possibilities (or impossibilities)
    1. early prototyping helps here

2 types of requirements

1. Functional requirements
2. Non-functional requirements

Functional requirements

What the system must do

1. specifications of the system’s functionality
2. actions that the system must take (verbs)
3. not a quality e.g. ‘fast’
4. take care to be precise & unambiguous

Non-functional requirements

A quality that the system must have (adjectives or adverbs)

1. secure, reliable, quickly, easy to use, compliant with Data Protection legislation
2. often about image in consumer products
3. customer and user may judge product mainly on nonfunctional qualities
4. again must be precise and unambiguous

Different types of nonfunctional requirements

1. look and feel
    1. spirit of the product’s appearance, aesthetics, trendy or corporate
    2. do what users expect
    3. not detailed design
2. usability
    1. specify ease to use **by whom**, **with what skills** or training
    2. also ease of learning
    3. user acceptance
    4. productivity gains
    5. error rates
    6. use by non-(English) speakers
    7. accessibility
3. performance
    1. speed
    2. safety
    3. throughput
    4. accuracy
    5. reliability
    6. availability
4. operational
    1. technical environment
    2. physical environment
5. security
    1. confidentiality
    2. recovery after emergencies
    3. safe storage

Examples

Requirements for PDA hospital system

It is impossible to design a (usable?) system without knowledge of

1. who will use it?
2. where will they use it?
3. what will they use it for?
4. how will they use it?
5. what will reduce their effectiveness in using the system?
6. what will reduce your effectiveness in designing the system?
7. what are the agendas of the people involved, hidden or otherwise?

Some methods used to ascertain the answers to the above questions include

1. ethnography
2. Interviews *
3. Focus Groups *
4. User Logging
5. Questionnaires
6. technology tour
7. task analysis *
8. end user profiling *
9. pact analysis *
10. cultural probes
11. artefact collection
12. activity theory

Methods marked with * are the ones we will concentrate on for this module. Others are available for your information.

Functional vs Non Functional Requirements

If there is any one thing any project must have in order not to be doomed to failure, that is a sensible and comprehensive collection of both the functional and non-functional requirements.

Any project's requirements need to be well thought out, balanced and clearly understood by all involved, but perhaps of most importance is that they are **not dropped or compromised halfway through** the project.

However, what exactly is the difference between 'functional' and 'non functional' requirements? It's not that complex, and once you understand the difference, the definition will be clear.

The official definition of 'a functional requirement' is that it essentially **specifies something the system should do.**

Typically, functional requirements will specify a behaviour or function, for example: "Display the name, total size, available space and format of a flash drive connected to the USB port." Other examples are "add customer" and "print invoice".

A functional requirement for a milk carton would be "ability to contain fluid without leaking"

Some of the more typical functional requirements include:

- Business Rules
- Transaction corrections, adjustments and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data
- Legal or Regulatory Requirements

So what about Non-Functional Requirements? What are those, and how are they different?

Simply put, the difference is that **non-functional requirements describe how the system works**, while **functional requirements describe what the system should do**.

The definition for a non-functional requirement is that it essentially specifies **how the system should behave** and that it is a constraint upon the systems behaviour. One could also think of non-functional requirements as quality attributes for of a system.

A non-functional requirement for a hard hat might be "must not break under pressure of less than 10,000 PSI"

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviours, for example: "Modified data in a database should be updated for all users accessing it within 2 seconds."

Some typical non-functional requirements are:

- Performance – for example Response Time, Throughput, Utilization, Static Volumetric
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability
- Environmental
- Data Integrity
- Usability
- Interoperability

As said above, non-functional requirements specify the system's 'quality characteristics' or 'quality attributes'.

Many different stakeholders have a vested interest in getting the non-functional requirements right particularly in the case of large systems where the buyer of the system is not necessarily also the user of the system.

The importance of non-functional requirements is therefore not to be trifled with. One way of ensuring that as few as possible non-functional requirements are left out is to use non-functional requirement groups. For an explanation on how to use non-functional requirement group.

Functional and nonfunctional requirements:

Basically, functional requirements directly support the user requirements by describing the "processing" of the information or materials as inputs or outputs. Nonfunctional requirements generally support all users in that they describe the business standards and the business environment, as well as the overall user's experience (user attributes).

Functional	Nonfunctional
Product features	Product properties
Describe the work that is done	Describe the character of the work
Describe the actions with which the work is concerned	Describe the experience of the user while doing the work
Characterized by verbs	Characterized by adjectives

The functional requirements specify what the product must do. They relate to the actions that the product must carry out in order to satisfy the fundamental reasons for its existence. Think of the functional requirements as the business requirements. That is, if you speak with a user or one of the business people, they will describe the things that the product must do in order to complete some part of their work. Keep in mind that the requirements specification will become a contract of the product to be built. Thus the functional requirements must fully describe the actions that the intended product can perform. I also relate it to a product you might purchase at a store -- if you look at the bullet features list on the back of the box, it is describing the functionality of the product.

Nonfunctional requirements are the properties that your product must have. Think of these properties as the characteristics or qualities that make the product attractive, or usable, or fast, or reliable. These properties are not required because they are fundamental activities of the product -- activities such as computations, manipulating data, and so on -- but are there because the client wants the fundamental activities to perform in a certain manner. They are not part of the fundamental reason for the product's existence, but are needed to make the product perform in the desired manner.

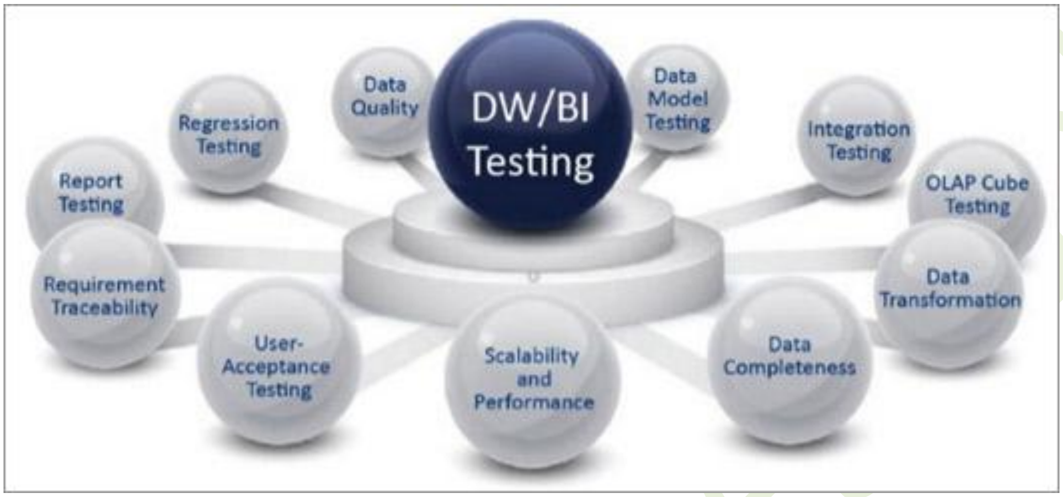
Nonfunctional requirements do not alter the product's functionality. That is, the functional requirements remain the same no matter what properties you attach to them. The non-functional requirements add functionality to the product -- it takes some amount of pressing to make a product easy to use, or secure, or interactive. However the reason that this functionality is part of the product is to give it the desired characteristics. So you might think of the functional requirements as those that do the work, and the nonfunctional requirements as those that give character to the work.

Nonfunctional requirements make up a significant part of the specification. They are important as the client and user may well judge the product on its non-functional properties. Provided the product meets its required amount of functionality, the nonfunctional properties -- how usable, convenient, inviting and secure it is -- may be the difference between an accepted, well-liked product, and an unused one.

Let's take a look at another real example. Anyone who has purchased a car, whether they were aware of it or not, made their final decision based on which car met both their functional and nonfunctional needs.

Functionally, the car had to be able to transport passengers from some starting location to a particular destination (that is, get me from point A to point B). A variety of nonfunctional attributes or characteristics were likely considered: security and safety, maintainability (ease of repair), reliability (probability of failure), scalability (ease of expansion), efficiency and performance (gas mileage, engine size, capacity -- both in number of passengers and cargo space), portability (ease of transport -- can it be towed easily or can it tow a trailer), flexibility (ease of change -- can it adapt to changes in weather/road conditions), and usability (ease of use -- comfort, handling, stereo sound quality).

Testing in a BI Project



Effective integration of testing in the implementation process builds trust and confidence among business users as they make crucial strategic decisions, based on the BI data generated.

Testing of Data Warehouse/Business Intelligence (DW/BI) applications is a little different than testing traditional transactional applications as it requires data-centric testing approach.

The typical challenges an enterprises faces while testing DW/BI implementations include:

- Data volume, variety and complexity
- Data anomalies from disparate data sources
- Data loss during data integration process and handshaking between sources
- Time consuming
- No audit trails, reusability or methodology resulting into high cost of quality
- Specialized skills required to execute data validation and verification process

To ensure data completeness, accuracy, consistency, security and reliability throughout the life cycle, it is important to test all these aspects at each data entry point in the BI architecture and not just at the end through reports or dashboards.

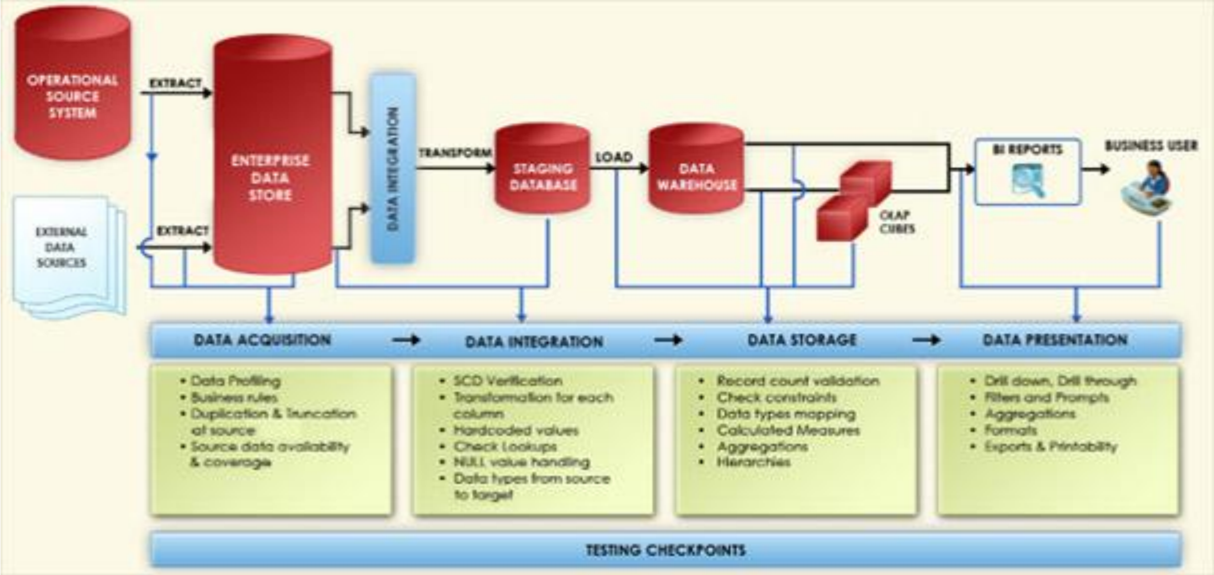
BI Testing Strategy:

The goal of testing BI applications is to achieve credible data. And data credibility can be attained by making the testing cycle effective.

A comprehensive test strategy is the stepping stone of an effective test cycle. The strategy should cover test planning for each stage, every time the data moves and state the responsibilities of each stakeholder e.g. business analysts, infrastructure team, QA team, DBA's, Developers and Business Users. To ensure testing readiness from all aspects the key areas the strategy should focus on are:

- Scope of testing: Describe testing techniques and types to be used.
- Test environment set up.
- Test Data Availability: It is recommended to have production like data covering all/critical business scenarios.
- Data quality and performance acceptance criteria.

The below diagram depicts the data entry points and lists a few sample checks at each stage. – Data Collection, Data Integration, Data Storage and Data Presentation.



Data Acquisition:

The primary aim of data completeness is to ensure that all of the data is extracted that needs to be loaded in the target. During the data acquisition phase it is important to understand the various data sources, the time boundaries of the data selected and any other special cases that need to be considered. The key areas this phase should focus on are:

- Validating the data required and the availability of the data sources from which this data needs to be extracted.
- Data profiling: Embedding data profiling activity helps in understanding the data, especially identifying different data values, boundary value conditions or any data issues at early stages. Identifying data problems early on will considerably reduce the cost of fixing it later in the development cycle.

Data Integration:

Testing within the data integration phase is the crux as data transformation takes place at this stage. Business requirements get translated into transformation logic. Once the data is transformed, thorough testing needs to be executed to ensure underlying data complies with the expected transformation logic. Key areas this phase should focus on are:

- Validating the Data Model: This involves validating the data structure with business specifications. This can be done by comparing columns and their data types with business specifications and reporting column requirements ensuring data coverage at source.
- Reviewing the Data Dictionary: Verifying metadata which includes constraints like Nulls, Default Values, Primary Keys, Check Constraints, Referential Integrity, Surrogate keys, Cardinality (1:1, m: n), etc.
- Validating the Source to Target Mapping: Ensuring traceability throughout will help build the quality aspects like consistency, accuracy and reliability.

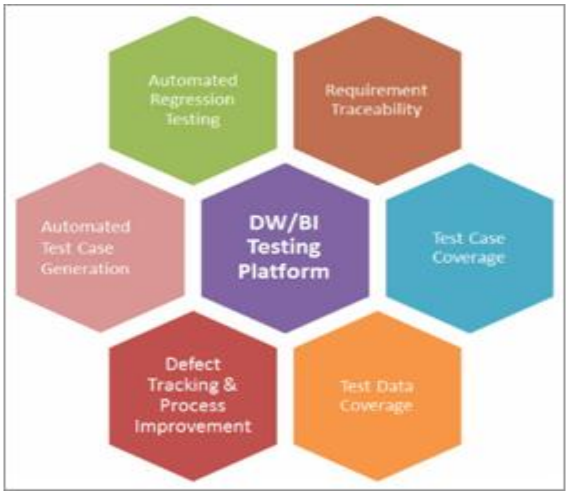
Data Storage: The data storage phase refers to loading of data within the data warehouse/data mart or OLAP cubes. The data loads can be one time, incrementally or in real-time. Key areas this phase should focus on are:

- Validating data loads based on time intervals.
- Performance and Scalability: Testing of initial and subsequent loads with performance and scalability aspect ensures that the system is within acceptable performance limits and can sustain further data growth.
- Parallel Execution and Precedence: Verifying appropriate parallel execution and precedence during ETL process is important as it may impact directly on performance and scalability of the system.
- Validating the Archival and Purge Policy: Ensures data history based on business requirements.
- Verifying error logging, exception handling and recovery from failure points.

Data Presentation:

This is the final step of the testing cycle and has the privilege of having a graphical interface to test the data. Key areas this phase should focus on are:

- Validating the Report Model.
- Report layout validation as per mockups and data validation as per business requirements.
- End to End Testing: Although individual components of the data warehouse may be behaving as expected, there is no guarantee that the entire system will behave the same. Thus execution and validation of end-to-end runs are recommended. Along with data reconciliation discrepancies, issues might surface such as resource contention or deadlocks. The end-to-end runs will further help in ensuring the data quality and performance acceptance criteria are met.


Automated Regression Testing
Requirement Traceability
Automated Test Case Generation
DW/BI Testing Platform
Test Case Coverage
Defect Tracking & Process Improvement
Test Data Coverage

While above considerations are given, one important aspect that still remains to be addressed is the issue of 'Time'. BitWise has created a platform based on DW/BI Testing Best Practices that automates and improves the overall effectiveness of DW/BI Testing. If you're interested in learning more about this platform, please contact us.

With the features and benefits of this platform, the intention is to address most of the DW/BI testing challenges:

- End-to-end traceability achieved right through source extraction to reports.
- 100% requirement and test data coverage through test cases.
- Test case automation of standard checks achieving considerable time savings.
- Up to 50% time savings through automated regression testing.
- Defect or bug tracking involving all the stakeholders.
- Improved testing cycle time through reusability.
- Process improvements with analytical reporting showcasing test data, test cases & defect trends.

Business intelligence deployment

Utilising the DW/BI system is the final step before business users can get access to the information. The first impression the business community gets is when introduced to the BI frontend drives. Because the acceptance from the users is important the deployment has to be thoughtfully planned to ensure that the DW/BI system can perform and is delivering the results it is designed to. To ensure that the implementation can perform and deliver it has to be exposed to extensive end-to-end testing .The process of testing is an ongoing activity along the development process, because defects that should be correct later in the lifecycle are difficult to find and are associated with exponentially increasing costs. A way of securing that the testing is done through the development lifecycle is to follow a methodology. Kimball prescribe that before adding the DW/BI system, it should have passed a mock test that will cover the following procedures;

- Testing procedures
- Data quality
- Operations process testing
- Performance testing
- Deployment testing
- User desktop readiness.

Documentation and Training
Maintenance and Support

Post Production Support Tips for the new Project

Some project managers think that it is the job of the support team to support the software when live and there should not be any phase after project closure. They strongly feels that post production support is not meant for the project team. In reality there is usually more knowledge transfer required than actually done during handshake period. Hence there tends to be an overlap which is sometimes referred to as a warranty or post-implementation operational support period.

The objective of the Post-Implementation Phase is to maintain and enhance the system to meet the ongoing needs of the user community. The Post-Implementation Support Phase begins once the system is in operation, the warranty period has expired, and the production review is complete. Post implementation activity may be the regular warranty support. This includes providing the support necessary to sustain, modify, and improve the operational software of a deployed system to meet user requirements. Shifting of focus from "have we installed the right product" to "have we met the user's expectation"? Post Implementation is the final stage in an application development project.

A process document describing the post-implementation process guides the activities performed in the post-implementation phase, which generally consists of the warranty period as per the contract signed by the client. It also includes helpdesk support, fixing the bugs, and planning for release of the reworked application and all other activities pertaining to the overall support of the system in action. There is a difference between "cutover" and "handover" -Cutovers take as much time as they take dependent on the project and Handover though follows the 80/20 rule of best practice. Immediately after cutover the existing project team controls 80% of the support and the new operations team controls 20%.

Most well-staffed projects do not have dedicated project resources but are made up of a combination of project team, outsourced consultants and assigned resources from operations. The operations resources who were assigned to the project to help build, test and validate the solution would be first line support to operate it so you have some built in resources who would be 100% dedicated or "handed over" on go-live being backed up by project level internal and external expertise.

The main activities in the implementation stage are planning and defining the process for rollout, to deploy the new application, train users on the new system after the rollout has been implemented, and communicate the details of deployment to relevant people.

<u>List of the challenges faced after project Go-Live:</u>

- Pressures from the business community and executive sponsor to meet the Go-Live date: ROI and realization value place enormous demands on the project manager to deliver a go-live.
- Lesser experience than required in post implementation: Often the project manager focuses more on delivering the project through a successful go-live and spend less time in transitioning the project product to the business.
- Need to understand the business more: Not all project managers understand or visualize how the product of the project will run in a departmental/steady state. It is easy to develop the project or product for business users but running the business is a real challenge. The project managers may not have ability to actually work with the system they implemented, their only focus is to deliver the project.
    - Long term benefits over short term gains:At times, the executive sponsor places more pressure by selecting a go-live date for the project manager, resulting in reverse engineering approach opted by the project manager. S/he is forced to plan and work backwards from that

date to meet the go-live. One should try to help sponsors to understand that as per project management triangle (triple constraint) all three on Scope, time and budget all are equally important. On time delivery would be of no use if the business requirements are not met.

How to face these challenges

- Right set of resources in the team: The project team needs to include representatives from business user community. This can help the team in 2 ways-reduction in functionality related surprises and reduction in time required in transition.
- Need to think beyond go-live:
    - Verifying that the project aligns to company objectives
    - Acquiring budget/time approval for post implementation up front
    - Acquiring the complete team including cross-functional team members and calculate the period, these resources are required for.
    - Identifying the real impacts of deferred functionality
    - Development of a proper transition plan
- Understand more about the business:To convince the stakeholders about their business, it is essential for the team to have a sound business knowledge, if not complete, at least pertaining to the area, they are working on. Attend internal trainings conducted by the company related to business.
- Need of training the end user and other key stakeholders:There are times when some functionalities are reported as issues by some users, to minimize this user trainings and user manuals must be provided to the right set of users and all the representatives.