Real Time Processing of Web Pages for Consumer Analytics

Authored by:

Team : SYNERGY

Fnu Lavanya Ramani (007529223)
Shweta Tiwari (007496190)
Sravanti Akella (007522801)

Prof. Rakesh Ranjan

Table of Contents

Abstract

In this era of continuous inter-connectivity, huge amount of data is created every second. Just like marketers, consumers are facing information overload, there arises a question, how can we use this information base to produce better, more relevant customer experience. The websites designed to provide/collect the product reviews, encompass useful market trends and customer purchasing options. This information is scattered and lacks a common platform for viewing, analyzing and performing business analytics. However, if brought under the unified umbrella, it can be utilized to deduce clear understanding of the product for both vendor and consumer. This paper will propose an innovative solution that will deliver single source of truth for product reviews of global brands for retail business which operate on massive data volumes.

Map reduce function will facilitate consolidation of information for consumers and retailers to make quick and smart decisions. Hadoop distributed file system will support review processing in real time.

Key Words

Big Data [2], Customer Rating, Hadoop [3], HDFS [3], Map Reduce [3], Predictive Analytics [4], Sentiment Analysis [5], Software Engine [6], Netezza Warehouse Solution [7].

Keyword Description

Big Data
It contains massive volumes of structured and unstructured data collected from various sources of information that grows too large that it becomes un-manageable to work with traditional Relational Database Management System.

Customer Review
It is the online product feedback given by the users through rating.

Hadoop
The Apache Hadoop provides open-source software for reliable, scalable, and distributed computing. It empowers applications to work with thousands of systems inter-connected over distributed network to process petabytes of raw data.

HDFS
Hadoop Distributed File System (HDFS) is the primary information/data storage system offered by Apache Hadoop. HDFS produces multi-replica of data and disseminate them on cluster nodes to enable scalable, reliable and extremely rapid processing.

Map Reduce
Hadoop Map Reduce is a programming paradigm and software architecture for writing applications that rapidly process vast volumes of data concurrently on scads of compute distributed system nodes.

Netezza Warehousing Solutions
The Netezza architecture is a platform that provides superior performance of Data Warehousing and Business Analytics. This framework encompasses the best capabilities of Symmetric Multiprocessing (SMP) and Massively Parallel Processing (MPP) for analyzing petabytes of data quickly.

Predictive Analysis
Predictive analytics is the technique of data mining that focus on the prediction of future possibilities, probabilities and trends.

Software Engine
A Software Engine is a self sustaining automated process that enables software in well defined way for its various components wherein it collaborates to work intuitively such that manual intervention in processing is minimal. It encompasses following features: Domain independence, Longevity, Reusability, Scalability, adaptability and incorporation of enhanced functionality. Other salient features include dynamic analysis, usage of minimal resources and applicability in diversified domains.

Sentiment Analysis
It is a field of business analytics that gives importance to opinion. It can also be termed as opinion mining. The data is created and analyzed by positive or negative feelings/reviews about a product, a person, a brand or any recent activity that is in the news.

Introduction

In this era of buying and advertising products online, there are an overwhelming number of websites selling fast moving consumer goods and an equal number of online resources to check the ratings and reviews. The advantages of online shopping are numerous such as 24x7 availability, location independence and scads of purchasing options. The percentage of online shopping has mounted to a whopping 47% in United States [1]. However, this exponentially growing business has its own challenges namely with more and more online purchase there are an ever increasing customer review updates not consolidated under common roof. Secondly, this critical information of user review is not properly utilized because of scattered and diversified source of reviews. Our Software Engine will provide amalgamation of dispersed users critical viewpoints to carry computations that can provide useful insights for better business. It will enable vendor to enhance his business by getting aligned with the user's choice and for consumer section of society, it offers succinct information to narrow down his choice to the most popular

current buying trend. It will follow the latest trend of giving core importance to the user sentiment, thus making the system act as how a consumer desires.
To build this system, we will be using Apache Hadoop (HDFS, Map-Reduce) which is a popular open source tool to handle Big Data in the enterprise. The objective of the system is to intake diversified product ratings (product name and rating) and store it in the HDFS for map reduce to act upon it. Map reduce will perform aggregated computation to provide top rated and most rated product for a particular category.

Foundation of proposed System – Sentiment Analysis [5]

In present day scenario of online businesses, virtual currency (user-reviews) plays a vital role to either make or break the business. This urged us to employ Opinion Mining that primarily focuses on classifying an opinion based on a spectrum which can correspond either positively or negatively about a product or a brand.
Our solution is based on sentiment analysis because the customers sentiment for a product is mainly influenced directly by the reviews provided by other users. These reviews are given as input that helps in deducing the output which is cumulative aggregate of all the reviews of that particular product.

Hadoop Map Reduce

This section will briefly describe the Hadoop Map Reduce Architecture and describe its significance in our project.
The fundamental need for Hadoop System arises from the fast multiplication of unstructured data volumes that originate from scads of sources namely web logs, text files, sensor readings, user generated content that necessitate ultra fast computing power. This growth demands new strategies for processing and analyzing information.
Companies that can extract important insights from these humongous volumes of Data can better control processes and costs, can better predict demands and can build better products.

Apache Hadoop provides reliable and inexpensive storage and new options for analysis on structured and unstructured data which motivated us to utilize these capabilities to implement our system.

Hadoop Architecture in the proposed System

It includes a fault-tolerant storage system called the Hadoop Distributed File System, or HDFS. It is able to store huge amounts of information in clusters of nodes interconnected on the network that can scale rapidly and withstand the failure of quintessential parts of the storage infrastructure without data loss.

HDFS stores three full copies of every customer review file by making three replicas and map it to different servers as shown in figure1. HDFS is used for manageable data distribution to divide the work on to several nodes in a cluster. This allows analysis to run concurrently to avoid bottlenecks put forth by monolithic storage systems.

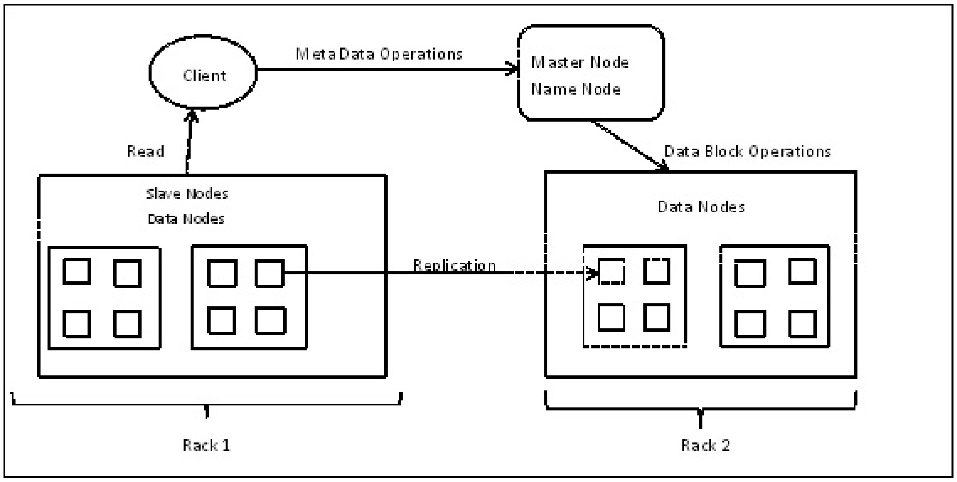


Figure 1: Hadoop Architecture

Hadoop Map Reduce and HDFS use simple, robust techniques on low cost computer systems to offer very high data availability and to process bulk information quickly and efficiently. Thus, Hadoop offers a powerful new tool for managing big data as shown in figure2.

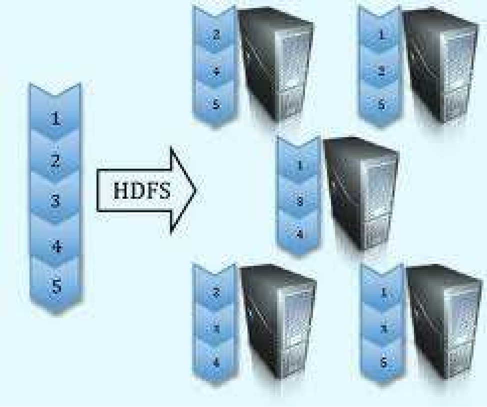


Figure 2: HDFS Mapping

Map Reduce Functionality

Hadoop Map-Reduce is a software framework for formulating applications which process peta bytes of data sets in parallel on clusters containing thousands of nodes of traditional hardware in a reliable manner offering high fault tolerance.

The *Job tracker* is the master job that is responsible to track the slave *task tracker* jobs that in turn run the mapper and reducer on each data node. The mapper job divides the incoming

customer review data-set to HDFS nodes and stores the block-id to disk mapping meta data in the Name Node which is the master node as shown in Figure3. It also handles scheduling, monitoring of the tasks and re-execution of the failed ones.
The framework sorts the outputs of the maps, which are then input to the reduce tasks as shown in the Figure 3. The output of which are stored in a warehouse for example, Netezza warehouse to utilize its fast query processing capabilities and help working directly with the raw de-normalized Map Reduce data without having to predefine and build traditional data warehousing cubes. It will carry out further processing to apply various business logics.

The reason for why we are employing Netezza instead of traditional databases is high query performance on petabytes of data. It pushes the CPU down to the disk level. Each disk is connected to a single CPU, and thus Netezza is able to process data just as fast as the disk can read the data (60MB/second per disk) whereas in a traditional database one has CPUs connected via a network switch to a bunch of disks. The throughput off disk is limited by the number of fiber channels that are connected to the network switch.

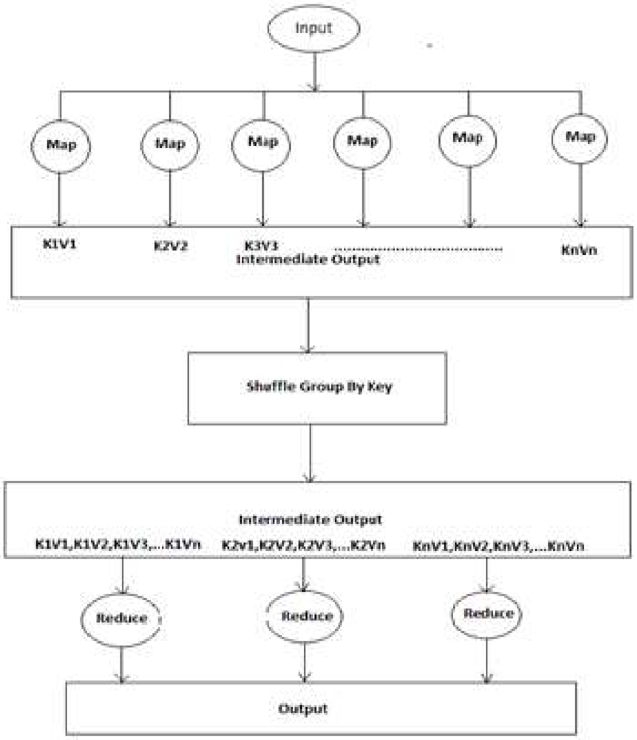


Figure 3: Map Reduce Functionality

Block Diagram

The following section will describe the various blocks employed in our proposed system.

1. Vendor Website: User provides the category or name of the product that he intends to buy. This is the input to Vendor's Inventory Database.

2. Inventory: It will capture the user's search choice and provide all the available items corresponding to that category to the software engine.

3. Software Engine: It operates on the Big data stored at multiple clusters of HDFS. It comprises of three modules:

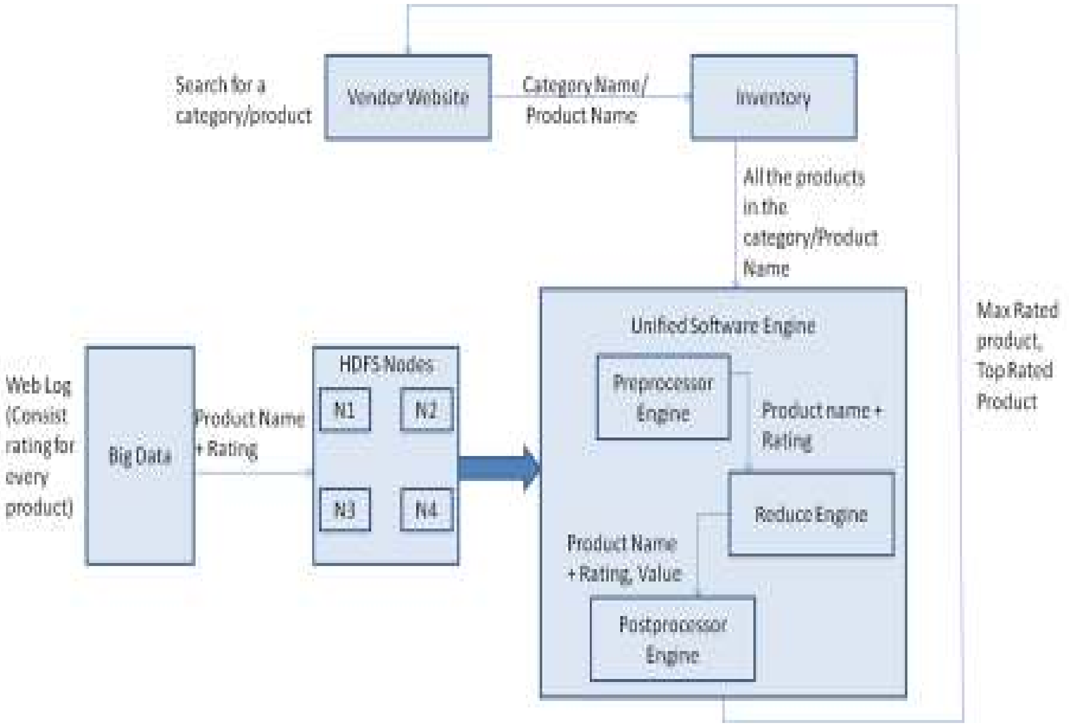
(i) **Preprocessor:** It takes all the item names present in the inventory for that category and concatenates it with the Rating Parameters. The concatenated result will become input to the Map-Reduce Engine.

(ii) **Map-Reduce Engine:** It is the heart of the software engine that performs the core computation on the Big data. This Reduce function will run in parallel for n (n = number of items * Number of Rating Parameter) times. The output of all the parallel threads will become input to Postprocessor Engine.

(iii) **Postprocessor Engine:** It contains all the business logic that is applied on the output of Reduce Engine. This mainly comprises logic for retrieving Top Rated, Most Rated and User Product Rating (where product provided by User as an Input in the vendor website)

4. Hadoop Distributed File System (HDFS): The bigdata is handled through HDFS that stores it into multiple manageable chunks also maintaining multiple replicas of data blocks for risk management. The reduce function runs independently and in parallel on each node.

5. BigData: It comprises of the critical user feedback about the retail product posted on the web.


Search for a category/product
Vendor Website
Category Name/ Product Name
Inventory
All the products in the category/Product Name
Unified Software Engine
Preprocessor Engine
Product name + Rating
Reduce Engine
Product Name + Rating, Value
Postprocessor Engine
Max Rated product, Top Rated Product
Web Log (Consist rating for every product)
Big Data
Product Name + Rating
HDFS Nodes
N1
N2
N3
N4

Software Engine

This section will explain Software Engine in detail. It comprises of the three major functionalities as described below.

Pre-processor Functionality

- The Preprocessor prepares the input for the Map Reduce Engine.
- It fetches the item names of a particular category present in the inventory.
- It concatenates each item with the Rating Parameters namely 1, 2, 3, 4, 5.
- The concatenated result will become input to the Reduce Engine.

Algorithm for Preprocessor Engine

```
1. Start
2.     For items 1..n sent by inventory database
3.         For all <rating_num> from 1 to 5
4.             Concatenate <item_name> and <rating_num> to make
                 <item_name+rating_num>
5.             Send to Hadoop Reduce Engine
6.         End
7.     End
8. Stop
```

Map Reduce Engine Functionality

- It performs the core computation on the HDFS.
- The Job tracker is the master program that monitors and schedules the task tracker.
- The Slave Task Trackers are responsible for running the Mapper and Reducer tasks.
- Mapper job assigns the Customer Review data to HDFS nodes and Reducer tasks perform aggregation on this data.
- The Reducer will run in parallel for n (n = number of items * Number of Rating Parameter) times.
- The output of all the parallel threads will become input to Postprocessor Engine

***Algorithm for Map Reduce Engine*:**

// There will be two distinct algorithms for map and reduce respectively.

Map Algorithm

```
1. Start
2. Initialize thread_count to number of HDFS nodes
3. For every rating 1..5
4.     For each thread 1..thread_count
5.          Find/Match the <key,value> in every node where key :=
            <item_name+rating_num>
            value := 1
6.          Append <key, value> = <item_name+rating_num, 1> in the mapper.out
7.   End
8. End
9. Stop
```

Reduce Algorithm

```
1. Start
2. Input mapper.out file in <key, value> format
3. Spawn a reducer thread
4. Run search algorithm to calculate the total number of occurrences of key.
5. Store the result in value, <value>:= <1+1+1+..n>
6. Append <key,value> to reducer.out
7. Send to postprocessor
8. End
```

Post Processor Functionality

- It contains all the business logic that is applied on the output of Map-Reduce Engine.
- This mainly comprises logic for retrieving Top Rated, Most Rated and User Product Rating of a particular category.
- **Top Rated**: Product with highest rating in the category
- **Most Rated**: Product with maximum number of ratings >=3 in that category.

- **User Product Rating**: The average rating of the product input by user.

Algorithm for Post Processor Engine

```
Start
// Calculate the sum of the ratings and average rating for a particular item
Input Reducer.Out
   For  rating_num from  1.. n
      Initialize sum_rating, count_rating to zero.
      //sum_rating is the cumulative count of rating of a item across all HDFS nodes
      //count_rating is the total count of rating of an item.
       For HDFS_node from 1..n
            count_rating := HDFS_node.count_rating
            sum_rating := sum_rating + count_rating
       End
     sum_rating[rating_num] = sum_rating
    //Store the total count of rating of the item in an array
   End;
 sum_all_rating[item_name] :=  (sum_rating[1]+ ….+ sum_rating[5])
 // sum_all_rating will add all the ratings for a particular item
 avg_rating_item[item_name]:=(1*sum_rating[1]+..n*sum_rating[5])/
 sum_all_rating[item_name]

//Calculate the User Rating for a particular item
If item_name = user_input_item_name
  rating[item_name] := avg_rating_item[item_name]
end if

//Calculation of top_rated item
// max function will calculate the top_rating of all the items
top_rating := max(rating[item_name] );

// Calculation for most_rated item
Initialize most_rated to zero
 For item_name in 1..n
   If avg_rating_item[item_name]  >= 3
      If most_rated < sum_all_rating[item_name]
        most_rated := sum_all_rating[item_name]
     End If
   End If
   return most_rated;
 End
Stop
```

Example

This section will briefly describe the functionality of the proposed system with an example of given below in Figure4. The item name taken here is Pantene in the category of Shampoo.

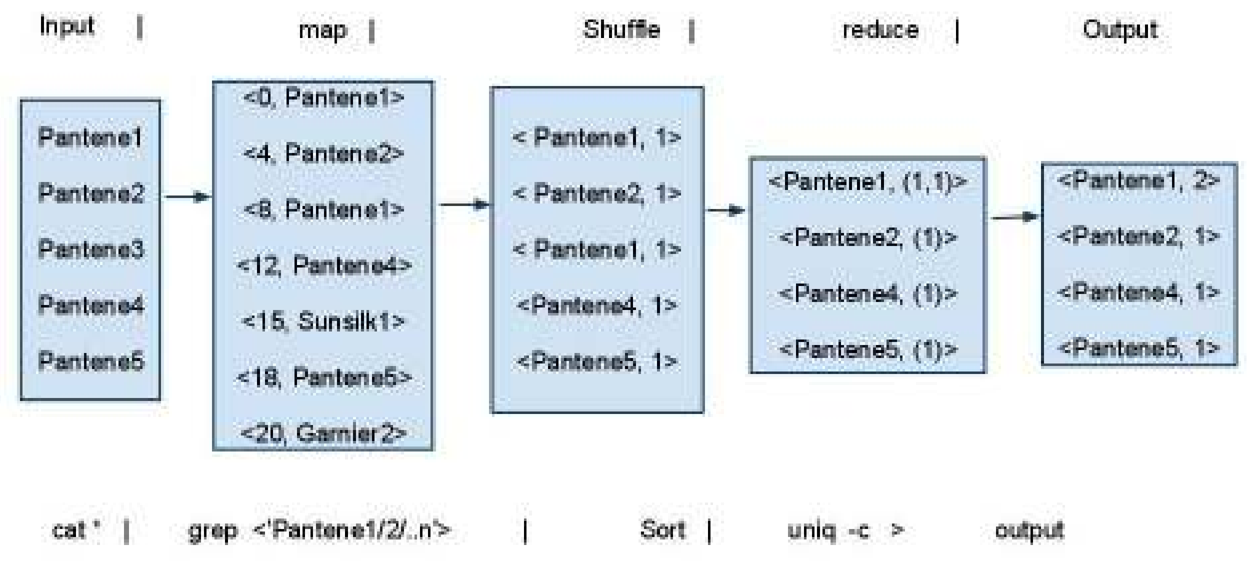


Figure 4: Example of Proposed System

Inputs and Outputs

The Map-Reduce framework operates entirely on <key, value> pairs, it views the input to the task as a set of <key, value> pairs and results a set of <key, value> pairs as the output of the task.

As shown in Figure4, the Input and Output types of a Map-Reduce jobs are as follows:

The Map-reduce consists of three phases. First comes a map phase that takes input records and produces output (key,value) pairs. This is followed by a shuffle phase that groups the (key,value) pairs by common values of the key, and finally a reduce phase that takes all pairs for a given key and produces a new value for the same key and this is the output of the Map-reduce.

(input) <k1, v1> -> **map** -> <k2, v2> -> **combine** -> <k2, v2> -> **reduce** -> <k3, v3> (output)

Example: <Filename, Pantene1> ->input -> <8, Pantene1> -> map-> <Pantene1, 1> -> shuffle> <Pantene1, 1,1,1,1,1,…> -> reduce -> <Pantene1, N>

Fault Tolerance

Our system offers fault tolerance using hadoop map reduce fault tolerant model by replicating each data node three times. In case of crash of a data-node or delay in producing result by the task tracker running on it containing the customer reviews, the job tracker schedules another task tracker to run on its replica to generate outcome. And if the Name Node which is the Master Node collapses, the entire system will be down.

Benefits

The solution to the consolidated reviews under the umbrella of retail store provides benefits for users as well as vendors as described below

User Benefits

1) User can check the reviews and get clear understanding of the current market trends under one roof without have to search for them.
2) It enables the user to get a better knowledge about the product well in advance about the future trends.
3) The power of online retail store is given a social touch enabling user to believe on what a common man's reaction is towards the product.
4) Bad products get poor ratings that help a user not to choose them. The power of sentiment analysis is exposed well.
5) User can remain unmoved by flossy advertisements and believe what a user for the product says in the review.

Vendor Benefits

1) Manufacturer/Vendor gets to know where the user liking is. What is hot in the market and what is not. What is being bashed and what is being appreciated and well accepted.
2) Helps in meeting high supply in case of positive reviews leading to the increased demand.
3) Better supply-chain management leading to better business decisions and better alignment as per market trend.
4) Increased challenge of making products better as customer has total power of accepting or denying it by giving bad/good reviews.
5) Produce competitive products based on the ones which get high reviews.
6) Better revenue and increased sales.

Caveat and Assumptions

Though our proposed system addresses the need of today by providing useful insights for well-informed judgments, however it has the following limitations and assumptions.

1) Tool will not catch the reviews which are not given rating in measurable number. For example, written reviews, tweets, Facebook comments will not be incorporated. The tools needs product to be given a numbered rating.
2) If the category for the product can't be determined, only that product's rating will be given to the customer. Most rated product and Top rated product will not be displayed.
3) All the product names in the big table is assumed to be concatenated with the rating.
4) If the most rated product has a rating of less than 3, it won't be showed to the user.
5) If the most rated product has a rating of less than 3 and that is the product that user entered, only that products rating will be displayed along with the top rated product. Most rated product will not be shown.

Future Enhancements

Our system is scalable and we would make the following feature enhancements in future releases.

1) Once the tool is deployed at the retail stores for online browsing, other future markets such as mortgage, airlines, restaurants and motels can be added.
2) We can integrate this solution with Facebook and Twitter to provide them added advantage for building better social graph.
3) We might add script understanding software also known as natural language text analytics that can interpret the written reviews and convert them to a number for incorporating them.

Conclusion

Sentiment analysis is core of the online social world today. Products that get high reviews and that have high user acceptance are sold well. This helps businesses perform better and produce the products that are the current trend in the market today. Our tool is positioned at the heart of this system to help benefit the customer and the vendor for understanding the product better based on the used reviews. It will provide a common platform to input the reviews and display the top rated and most rated product.

It provides benefits to both sections of the commercial world that is, Customer and Vendor. For Customer, it helps in streamlining their purchasing choice and for the vendor it guides him to get aligned with User's Choice. To consummate, this tool will bring clarity and better understanding of the current market trends.

References

[1] Statistical Report (2011) *US Census Bureau*
http://www.census.gov/compendia/statab/cats/wholesale_retail_trade/online_retail_sales.html
[2] Bryan Thompson, Micheal Personick (2010) *Big Data*
http://www.systap.com/bigdata.htm
[3] Apache Hadoop (2007) *Hadoop 0.17 Documentation*
http://hadoop.apache.org/common/docs/r0.17.2/mapred_tutorial.html#Purpose
[4] SearchCRM.com (2005) *Predictive Analysis*
http://searchcrm.techtarget.com/definition/predictive-analytics
[5] Matthew Russel (2011) *With Sentiment Analysis, context always matters*
http://radar.oreilly.com/2011/03/sentiment-analysis-context.html
[6] *Sofware Engine* (2007)
http://it.toolbox.com/wiki/index.php/Software_engine
[7] IBM *Netezza* (2011)
http://www.netezza.com/documents/whitepapers/Netezza_Appliance_Architecture_WP.pdf