International Journal of Engineering Research and Modern Education (IJERME)
ISSN (Online): 2455 - 4200
(www.rdmodernresearch.com) Volume I, Issue I, 2016

E-LEARNING: DISTRIBUTED PROCESSING OF LARGE DATASETS WITH A PARALLEL ALGORITHM

C. Kalaiarasi* & K. Adlin Suji**

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

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

Abstract:

Data that are generated from variety of sources with massive volumes, high rates, and different data structure are collectively known as Big Data. Big Data processing and analyzing is a challenge for the current systems because they were designed without Big Data requirements in mind and most of them were built on centralized architecture, which is not suitable for Big Data processing because it results on high processing cost and low processing performance and quality. A Map Reduce framework usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. Map Reduce framework was built as a parallel distributed programming model to process such large-scale datasets effectively and efficiently. Big Data software analysis solutions were implemented on Map Reduce framework, describing their datasets structures and how they were implemented with MongoDB as NoSQL Database. NoSQL encompasses a wide variety of different database technologies that were developed in response to the demands presented in building modern applications. MongoDB stores data using a flexible document data model. Documents contain one or more fields, including arrays, binary data and sub-documents. In this project Fabcoder portal is used in turn which uses the Map reduce Framework. Since the portal is implemented using Map reduce framework, the required information can be obtained effectively and efficiently.

Key Terms: NOSQL, MongoDB, HM, Map reduce & Fabcoder

1. Introduction:

The use and adaption of Big Data within Organization processes is beneficial and allows efficiencies terms of cost, productivity, and innovation. This process does not come without its flows. Data analysis often requires multiple parts of Organization to work in collaboration and create new and innovative processes to deliver the desired outcome.

Organizations today are confronted with the challenge and the opportunity of data growing at unprecedented rates. This data comes from numerous sources – ERP systems, Data Warehouses, Website logs, Web Services, Social Media, Mobile devices, Sensors, etc. - in various forms - Structured, Semi-structured and Unstructured. Big Data" is the catch all phrase for this rapidly changing field. Big Data analytics has the potential to provide great insights and opportunities to organizations in the areas of consumer behavior, marketing, fraud detection and customer service. With the right technical architecture, true real-time decisions are enabled providing organizations with heightened agility. While most organizations recognize the importance and benefits of Big Data analytics, there are challenges arising from the nature of Big Data and limitations of existing technologies that need to be considered. HDFS stores extremely large files containing record-oriented data. It does not split large data files. The size of the files and the number of replications are not configurable.

2. Related Work:

Project Use in Domain:

This project has been deployed based on a Big Data and the algorithm used in the project is known as pattern gathering.

- ✓ Big data usually includes data sets with sizes beyond the ability of commonly used software tools.
- ✓ Big data doesn't sample, it just observes and tracks what happens. It is often available in real-time.
- ✓ Big data draws from text, images, audio, video plus it completes missing pieces through data fusion.
- ✓ Map Reduce algorithm helps in sending the Map & Reduce tasks to appropriate servers in a cluster

Technical Terms:

- ✓ **Term Frequency (TF):** It measures how frequently a particular term occurs in a document. It is calculated by the number of times a word appears in a document divided by the total number of words in that document.
- ✓ **Inverse Document Frequency (IDF):** It measures the importance of a term. It is calculated by the number of documents in the text database divided by the number of documents where a specific term appears.
- ✓ **Collating:** Mapper computes a given function for each item and emits value of the function as a key and item itself as a value. Reducer obtains all items grouped by function value and process or save them. In case of inverted indexes, items are terms (words) and function is a document ID where the term was found.
- ✓ **Sorting:** Sorting in Map Reduce is originally intended for sorting of the emitted key-value pairs by key, but there exist techniques that leverage the implementation specifics to achieve sorting by values.

Algorithm Details:

The Map Reduce algorithm contains two important tasks, namely Map and Reduce.

- ✓ The map task is done by means of Mapper Class
- ✓ The reduce task is done by means of Reducer Class.

Mapper class takes the input, tokenizes it, maps and sorts it. The output of Mapper class is used as input by Reducer class, which in turn searches matching pairs and reduces them.

Map Reduce implements various mathematical algorithms to divide a task into small parts and assign them to multiple systems. In technical terms, Map Reduce algorithm helps in sending the Map & Reduce tasks to appropriate servers in a cluster. These mathematical algorithms may include the following –

- ✓ Sorting
- ✓ Searching
- ✓ Indexing
- ✓ TF-IDF

Sorting: Sorting is one of the basic Map Reduce algorithms to process and analyze data. Map Reduce implements sorting algorithm to automatically sort the output key-value pairs from the mapper by their keys.

- ✓ Sorting methods are implemented in the mapper class itself.
- ✓ In the Shuffle and Sort phase, after tokenizing the values in the mapper class, the Context class (user-defined class) collects the matching valued keys as a collection.

- ✓ To collect similar key-value pairs (intermediate keys), the Mapper class takes the help of Raw Comparator class to sort the key-value pairs.
- ✓ The set of intermediate key-value pairs for a given Reducer is automatically sorted by Hadoop to form key-values (K2, {V2, V2, …}) before they are presented to the Reducer.

Searching: Searching plays an important role in Map Reduce algorithm. It helps in the combiner phase (optional) and in the Reducer phase.

Indexing: Normally indexing is used to point to a particular data and its address. It performs batch indexing on the input files for a particular Mapper. The indexing technique that is normally used in Map Reduce is known as inverted index. Search engines like Google and Bing use inverted indexing technique.

TF-IDF: TF-IDF is a text processing algorithm which is short for Term Frequency – Inverse Document Frequency. It is one of the common web analysis algorithms. Here, the term 'frequency' refers to the number of times a term appears in a document.

3. Proposed Work:

In this project, enterprise system has a centralized server to store and process data. The following illustration depicts a schematic view of a traditional enterprise system. Traditional model is certainly not suitable to process huge volumes of scalable data and cannot be accommodated by standard database servers. Moreover, the centralized system creates too much of a bottleneck while processing multiple files simultaneously. The data will be stored and retrieved from database within the group of organization. In all branch they can store and retrieve the data. Here the data will stored in the common database and the values will be retrieved using Map Reduce framework.

Advantages

- ✓ Map Reduce framework stores files containing record-oriented data.
- ✓ It splits large data files into chunks of 64 MB, and replicates the chunk across three different nodes in the cluster.

4. Experimental Analysis and Results:

System Analysis:

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

Analysis Model:

The model that is basically being followed is the SPIRAL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

- ✓ Requirement Analysis
- ✓ Project Planning
- ✓ System design
- ✓ Detail design
- ✓ Coding
- ✓ Unit testing
- ✓ System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project. The following diagram shows how a spiral model acts like

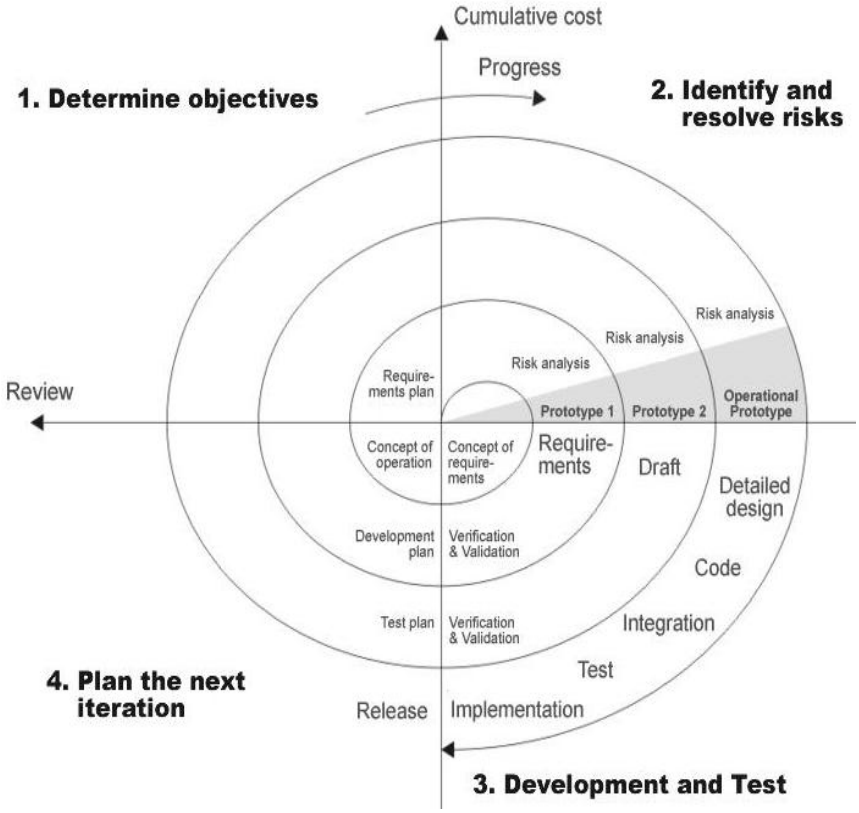

Figure 1: Spiral Model

Feasibility Report:

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- ✓ Technical Feasibility
- ✓ Operation Feasibility
- ✓ Economical Feasibility

Technical Feasibility: The technical issue usually raised during the feasibility stage of the investigation includes the following:

- ✓ Does the necessary technology exist to do what is suggested?
- ✓ Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- ✓ Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- ✓ Can the system be upgraded if developed?
- ✓ Are there technical guara
- ✓ ntees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a browser based user interface for audit workflow. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

Operational Feasibility: The analyst considers the extent the proposed system will fulfill his departments. That is whether the proposed system covers all aspects of the working system and whether it has considerable improvements. We have found that the proposed "Secure transaction" will certainly have considerable improvements over the existing system.

Economic Feasibility: The proposed system is economically feasible because the cost involved in purchasing the hardware and the software are within approachable. Working in this system need not required a highly qualified professional. The operating-environment costs are marginal. The less time involved also helped in its economic feasibility.

5. Conclusion and Future Enhancement:

In this project, dynamic community detection problem is focused in evolving content-based networks. The Map Reduce frame work programming model has been success- fully used at websites for many different purposes. An implementation of Map Reduce that scales to large clusters of machines comprising the data easy to learn. Map Reduce framework has been used for filtering the content for clear knowledge about the portion. The process of filtering into the portal to know the work of Map Reduce framework. Map Reduce framework support to store the large dataset with a split chunks for reuse and clear about the data. The Map Reduce library in the user program first splits the input files into M pieces of typically 16 megabytes to 64 megabytes (MB) per piece to provide the particular topic explanation to the user by using the filtering method within the system.

I plan to examine whether the techniques used to support unstructured data. The main challenge is still the limitation of storage. Similar to the relational case, I need a

layout plan to guide us which part of data should be maintained to maximize the performance. The first step is to discover query patterns. Three most popular workloads on unstructured data are keyword based queries, data mining tasks and machine learning tasks.

6. References:

1. D. J. Abadi, P. A. Boncz, and S. Harizopoulos. Column oriented database systems. PVLDB, 2(2):1664–1665, 2009.
2. N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. S. Manasse, and R. Panigrahy. Design tradeoffs for ssd performance. In USENIX Annual Technical Conference, pages 57–70, 2008.
3. M. Canim, G. A. Mihaila, B. Bhattacharjee, K. A. Ross, and C. A. Lang. An object placement advisor for db2 using solid state storage. Proc. VLDB Endow., 2(2):1318–1329, Aug. 2009.
4. S. Chen. Flashlogging: exploiting flash devices for synchronous logging performance. In SIGMOD, pages 73–86, 2009.
5. S. Chen. Cheetah: A high performance, custom data warehouse on top of map reduce. PVLDB, 3(2):1459–1468, 2010.
6. J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In OSDI, 2004.
7. B. Debnath, S. Sengupta, and J. Li. Skimpystash: Ram space skimpy key-value store on flash-based storage. In SIGMOD, pages 25–36, 2011.
8. J. Dittrich, J.-A. Quian´e-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad. Hadoop++: Making a yellow elephant run like a cheetah (without it even noticing). PVLDB, 3(1):518–529, 2010.
9. J. Do, D. Zhang, J. M. Patel, D. J. DeWitt, J. F. Naughton, and A. Halverson. Turbo charging dbms buffer pool using ssds. In SIGMOD, pages 1113–1124, 2011.
10. A. Floratou, J. M. Patel, E. J. Shekita, and S. Tata. Column-oriented storage techniques for map reduce. PVLDB, 4(7):419–429, Apr. 2011.