Map Reduce Technique for HIPI

Madhu M Nayak
Assistant Professor,
Department of CSE, GSSSIETW,
Mysuru

Pradeep.S
Assistant Professor,
Department of CSE, GEC,
Kushal Nagar

***Abstract* - Due to the increasing popularity of cheap digital photography equipment, personal computing devices with easy to use cameras, and an overall improvement of image capture technology with regard to quality, the amount of data generated by people each day shows trends of growing faster than the processing capabilities of single devices. It becomes computationally inefficient to analyze such huge data. The amount of raw data available has been increasing at an exponential rate. For the effective handling of such massive data, the use of MapReduce framework has been widely came into focus. Over the last few years, MapReduce has emerged as the most popular computing paradigm for parallel, batch-style and analysis of large amount of data. In this paper, we are going to work around MapReduce, its advantages, disadvantages and how it can be used in integration with other technology.**

Keywords: Hadoop, MapReduce, JobTracker, TaskTracker

I. INTRODUCTION

With the concern of big data, the three main challenges being faced are volume, velocity and variety. Volume refers to the amount of data to be processed, velocity refers to the speed at which the data are processed and variety is the ability to manage different types of data. Veracity is the abnormality or uncertainties of the data. The big data capture, manage and process the large data must be done in an efficient way. Nowadays, large volumes of data are in an unstructured manner. It is very difficult to perform the operation in unstructured data. So the data need to be structured in order to perform some operations. Hadoop Map Reduce is used to structure the data [1].

II. MAPREDUCE TECHNIQUE

Over the last few years, for analysis of large amount of data in parallel and batch style MapReduce has emerged as the most popular paradigm[2]. Map Reduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner. MapReduceis is a technique that can process large data files which are multi structured across massive data sets. It is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

MapReduce Algorithm

Generally MapReduce paradigm is based on sending the computer to where the data resides. MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

Map stage: The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

Reduce stage: This stage is the combination of the Shuffle stage and the Reduce The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS. During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster. The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes. Most of the computing takes place on nodes with data on local disks that reduces the network traffic. After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

A Map Reduce job splits the input data set into independent "chunks" that are processed by map tasks in parallel. The framework sorts the map outputs, which are then input to reduce tasks. Job inputs and outputs are stored in the file system. The MapReduce framework and the HDFS are typically on the same set of nodes, which enables the framework to schedule tasks on nodes that contain data. The Map Reduce framework consists of a single master JobTracker and one slave TaskTracker per node. The master is responsible for scheduling job component tasks on the slaves, monitoring tasks, and re-executing failed tasks.

The slaves execute tasks as directed by the master. Minimally, applications specify input and output locations and supply map and reduce functions through implementation of appropriate interfaces or abstract classes. Although the Hadoop framework is implemented in Java, Map Reduce applications do not have to be written in Java. HDFS uses a master/slave architecture in which one device (the master) controls one or more other devices (the slaves). A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode.

A slave or worker node acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. The job tracker schedules map or reduce jobs to task trackers with an awareness of the data location.

Example
If node A contains data (x, y, z) and node B contains data (a, b, c), the job tracker schedules node B to perform map or reduce tasks on (a, b, c) and node A would be scheduled to perform map or reduce tasks on (x, y, z). This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer.

This technology is much simpler conceptually but very powerful when put along with Hadoop framework. There are two major steps:

Map:
In Map step master node takes input and divides into simple smaller chunks and provides it to other worker node. It is a function that parcels out work to different nodes in the distributed cluster.

Reduce:
In Reduce step it collects al the small solution of the problem and returns as output in one unified answer. Both of these steps use function which relies on Key-Value pairs. This process runs on the various nodes in parallel and brings faster results for framework. The MapReduce framework is fault-tolerant because each node in the cluster is expected to report back periodical y with completed work and status updates. If a node remains silent for longer than the expected interval, a master node makes note and re-assigns the work to other nodes.

III. MAP REDUCE TECHNIQUE FOR LARGE SCALE IMAGES

Big data is often boiled down to a few varieties including social data, machine data, and transactional data. Social media data is providing remarkable insights to companies on consumer behavior and sentiment that can be integrated with CRM data for analysis, with 230 million tweets posted on Twitter per day, 2.7 billion Likes and comments added to Facebook every day, and 60 hours of video uploaded to YouTube every minute. This is what we mean by velocity of data.

Machine data consists of information generated from industrial equipment, real-time data from sensors that track parts and monitor machinery (often also called the Internet of Things), and even web logs that track user behavior online.

Regarding transactional data, large retailers can generate multitudes of data on a regular basis considering that their transactions consist of one or many items, product IDs, prices, payment information, manufacturer and distributor data, and much more. Major retailers like Amazon.com, which posted $10B in sales in Q3 2011, and restaurants like US pizza chain Domino's, which serves over 1 million customers per day, are generating petabytes of transactional big data. In defining big data, it's also important to understand the mix of unstructured and multi-structured data that comprises the volume of information.
Unstructured data comes from information that is not organized or easily interpreted by traditional databases or data models. Metadata, Twitter tweets, and other social media posts are good examples of unstructured data.
Multi-structured data refers to a variety of data formats and types and can be derived from interactions between people and machines, such as web applications or social networks. A great example is web log data, which includes a combination of text and visual images along with structured data like form or transactional information. As digital disruption transforms communication and interaction channels—and as marketers enhance the customer experience across devices, web properties, face-to-face interactions and social platforms—multi-structured data will continue to evolve.

Advantages of MapReduce Technique
The following are the advantages of MapReduce [4]:
1. Simple and easy to use- The MapReduce model is simple but expressive. It is easy even for the programmers without any experience in parallel and Distributed Computing, since it hides the details of parallelization, fault tolerance, and locality of data. With MapReduce, a programmer only needs to define his job with only Map and Reduce Functions, without having to specify physical distribution of his job across the nodes.
2.Flexible- MapReduce does not have any dependency on data model and schema. With MapReduce a programmer can deal with irregular or unstructured data more easily than they do with DBMS.
3. Independent of the storage- MapReduce is basically independent of the storage layers. Thus, MapReduce can work with different storage layers.
4. Fault tolerance- MapReduce is highly fault tolerant. It detects the failed Map and Reduce tasks of failed nodes and reassigns them to other idle nodes of the cluster.
5. High scalability- MapReduce has been designed in such a way that it can scale up to large clusters of machines. It supports runtime scheduling which enables dynamic adjusting of resources during job execution. Hence, offering elastic scalability.

6. Supports data locality by collocating the data with the compute node, so it reduces network communication cost
7. Ability to handle data for heterogeneous storage system, since MapReduce is storage independent, and it can analyze data stored in different storage system.
8. High level language support which was not there earlier. Microsoft scope, Apache Pig and Apache Hive all aim at supporting declarative query languages for the MapReduce Framework

Disadvantages of MapReduce Technique
The following are the pitfalls in the MapReduce framework compared to DBMS [3]:
1. Earlier there was no high level language support like SQL in DBMS and any query optimization technique.
2. MapReduce is schema free and index free. An MR Job can work right after its input is loaded into its storage.

3. A single fixed dataflow which don't support for algorithms that require multiple inputs. MapReduce is originally designed to read a single input and generate single output.
4. Low efficiency- With fault tolerance and scalability as its primary goals, MapReduce operations are not always optimized for I/O efficiency. In addition, MapReduce are blocking operations. A transition to the next stage cannot be made until all the tasks of the current stage are finished. Also, MapReduce has a latency problem that comes from its inherent batch processing nature.
5. Very young compared to 40 years of DBMS.

Challenges in MapReduce Technique
The two major challenges in MapReduce technique are [6]:
1. Due to frequent checkpoints and runtime scheduling with speculative execution, MapReduce reveals low efficiency. Thus, how to increase efficiency guaranteeing the same level of scalability and fault tolerance is a major challenge. The efficiency problem is expected to be overcome in two ways: Improving MapReduce itself or leveraging new hardware.
2. Second challenge is how to efficiently manage resources in the clusters which can be as large as 4,000 nodes in multi user environment and achieving high utilization of MR clusters. Another challenge is the energy issue. Since the energy cost of data centers hits 23% of the total amortized monthly operating expenses. It is necessary to devise an energy efficient way to control nodes in data center when they are idle.

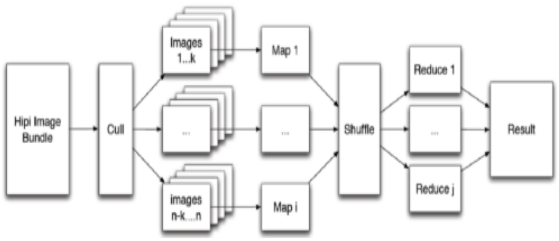
Strategies in MapReduce Technique
Two strategies are proposed:
1. Covering Set Approach- It designates in advance some nodes that should keep atleast a replica of each data block and the other nodes are powered down during low utilization periods.

2. All-In strategy- It saves energy in all or nothing fashion. In this, all the MR jobs are queued until it reaches a threshold, then all the nodes are run to finish MR jobs and then all the nodes are powered down until new jobs are queued.

Applications of MapReduce Framework
The following are the applications of MapReduce framework [5]:
1) Processing crawled documents, web request logs in order to compute various kinds of derived data, such as inverted indices.
2) Machine learning task, scientific simulation and large scale image processing tasks.
3) Opinion mining. It is technique for extracting estimation from the internet. It is known as sentiment classification.
4) MapReduce can be for clustering very large-moderate to-high dimensionality dataset.
5) MapReduce can also be used for performing join task: Two way joins- Joins which involve only two datasets. The two way join is classified into Map side join, Reduce side join and broadcast join. Multi way joins- Where joins involve more than two datasets.

IV. HADOOP IMAGE PROCESSING INTERFACE

HIPI is an image processing library designed to be used with the Apache Hadoop MapReduce parallel programming framework. HIPI facilitates efficient and high-throughput image processing with MapReduce style parallel programs typically executed on a cluster. It provides a solution for how to store a large collection of images on the Hadoop Distributed File System (HDFS) and make them available for efficient distributed processing.



This diagram shows the organization of a typical MapReduce/HIPI program. The primary input object to a HIPI program is a HIPI Image Bundle (HIB). A HIB is a collection of images represented as a single file on the HDFS. The HIPI distribution includes several useful tools for creating HIBs, including a MapReduce program that builds a HIB from a list of images downloaded from the Internet. The first processing stage of a HIPI program is a culling step that allows filtering the images in a HIB based on a variety of user-defined conditions like spatial resolution or criteria related to the image metadata. This functionality is achieved through the Culler class. Images that are culled are never fully decoded, saving processing time.

The images that survive the culling stage are assigned to individual map tasks in a way that attempts to maximize data locality, a cornerstone of the Hadoop MapReduce programming model. This functionality is achieved through the HibInputFormat class. Finally, individual images are presented to the Mapper as objects derived from the HipiImage abstract base class along with an associated HipiImageHeader object. For example, the ByteImage and FloatImage classes extend the HipiImage base class and provide access to the underlying raster grid of image pixel values as arrays of Java bytes and floats, respectively. These classes provide a number of useful functions like cropping, color space conversion, and scaling.

The records emitted by the Mapper are collected and transmitted to the Reducer according to the built-in MapReduce shuffle algorithm that attemps to minimize network traffic. Finally, the user-defined reduce tasks are executed in parallel and their output is aggregated and written to the HDFS.

V. CONCLUSION

The need to process enormous quantities of data has never been greater. Not only are terabyte- and petabyte-scale datasets rapidly becoming commonplace, but there is consensus that great value lies buried in them, waiting to be unlocked by the right computational tools. Big Data analysis tools like

Map Reduce over Hadoop and HDFS, promises to help organizations better understand their customers and the marketplace, hopefully leading to better business decisions and competitive advantages. For engineers building information processing tools and applications, large and heterogeneous datasets which are generating continuous flow of data, lead to more effective algorithms for a wide range of tasks, from machine translation to spam detection. The ability to analyze massive amounts of data may provide the key to unlocking the secrets of the cosmos or the mysteries of life. MapReduce can be exploited to solve a variety of problems related to text processing at scales that would have been unthinkable a few years ago.

REFERENCES

[1] Subramaniyaswamy V, "Unstructured Data Analysis on Big Data using Map Reduce", 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15). 2015.

[2] Seema Maitrey, "Handling Big Data Efficiently by using MapReduce Technique", 2015 IEEE International Conference on Computational Intelligence & Communication Technology.

[3] SaloniMinocha and Hari Singh, "Mapreduce Technique: Review and S.W.O.T Analysis", International Journal of Engineering Research, No.5, Issue No.6, pp : 531-533

[4] Lee KH, et al, "Parallel data processing with MapReduce: a survey", AcMsIGMoD Record. 2012 Jan 11;40(4):11-20.