Scientific Journal Impact Factor (SJIF) : 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

ANALYSING BIG DATA TOOLS

AVINASH KUMAR[1], AVINASH RAJ[2] AND BIPIN KUMAR YADAV[3]
[1,2,3] *Department Of Computer Science, IIMT COLLEGE OF ENGINEERING,GR. NOIDA*

Abstract- Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, and information privacy. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. Our objective was to identify gaps, providing motivation for new research, and outline collaborations to Apache Hadoop and its ecosystem, classifying and quantifying the main topics addressed in the literature. [Results:] Our analysis led to some relevant conclusions: many interesting solutions developed in the studies were never incorporated into the framework.

I. INTRODUCTION

One of the largest technological challenges in software systems research today is to provide mechanisms for storage, manipulation, and information retrieval on large amounts of data. Web services and social media produce together an impressive amount of data, reaching the scale of petabytes daily (Facebook, 2012). These data may contain valuable information, which sometimes is not properly explored by existing systems. Most of this data is stored in a non-structured manner, using different languages and formats, which, in many cases, are incompatible. The assimilation of computing into our daily lives is enabling the generation of data at unprecedented rates. In 2008, IDC estimated that the "digital universe" contained 486 exabytes of data .The MapReduce programming model has emerged as a scalable way to perform data-intensive computations on commodity cluster computers.The success of MapReduce has inspired the creation of Hadoop, a popular open-source implementation. Written in Java for cross-platform portability, Hadoop is employed today by a wide range of commercial and academic users for backend data processing. A key component of Hadoop is the Hadoop Distributed File System (HDFS), which is used to store all input and output data for applications. The efficiency of the MapReduce model has been questioned in recent research contrasting it with the parallel database paradigm for large-scale data analysis.

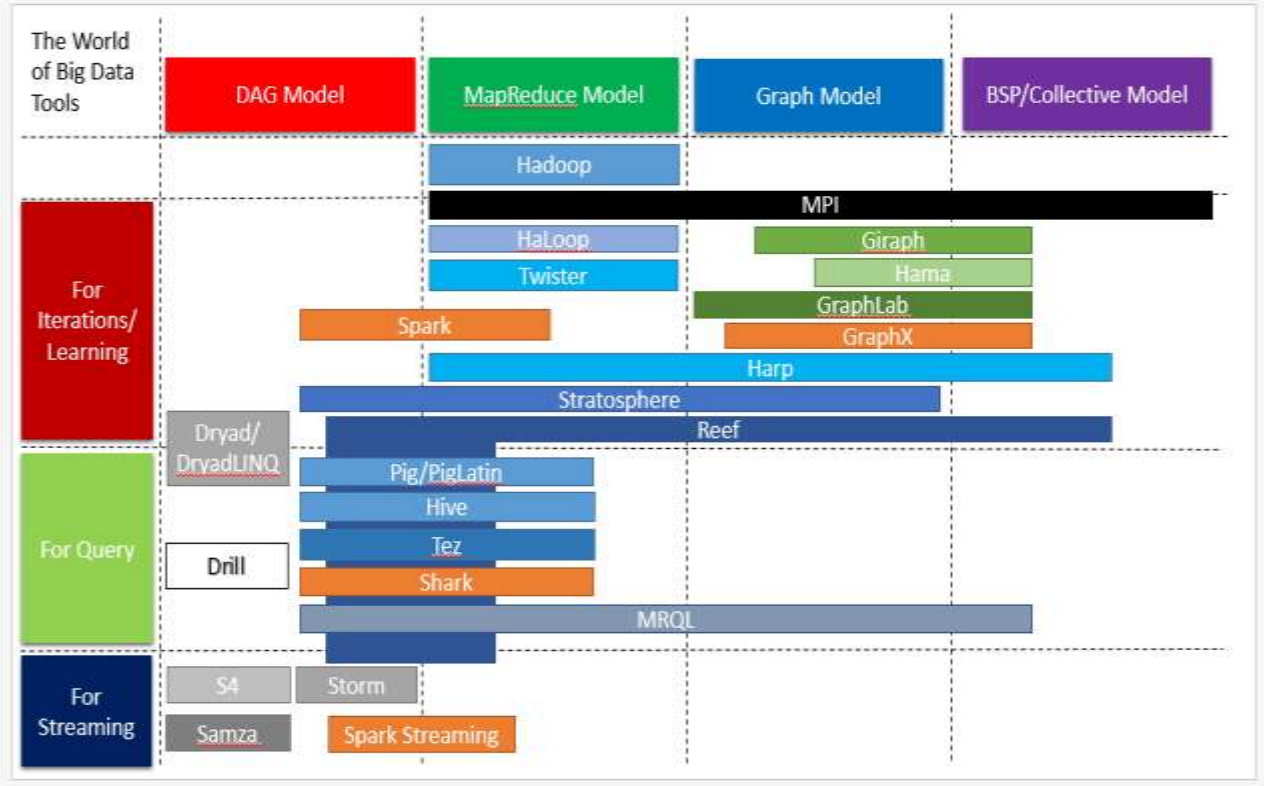
Hadoop applications performed poorly in experiments when compared to similar programs using parallel databases ,However, this work did not perform the profiling necessary to distinguish the fundamental performance of the MapReduce programming model from a specific implementation. We find that it is actually the implementation of the Hadoop storage system that degrades performance significantly.

Types of tools used in Big-Data

- Where processing is hosted?
  - Distributed Servers / Cloud (e.g. Amazon EC2)
- Where data is stored?

  - Distributed Storage (e.g. Amazon S3)
- What is the programming model?
  - Distributed Processing (e.g. MapReduce)
- How data is stored & indexed?
  - High-performance schema-free databases (e.g. MongoDB)
- What operations are performed on data?
  - Analytic / Semantic Processing

The world map of big data tools



Big data tools for HPC and supercomputing

MPI(Message Passing Interface, 1992)

-Provide standardized function interfaces for communication between parallel processes.

Collective communication operations

-Broadcast, Scatter, Gather, Reduce, Allgather, Allreduce, Reduce-scatter.

Popular implementations

-MPICH (2001)

-OpenMPI (2004)

MapReduce Model

Google MapReduce (2004)

-Jeffrey Dean et al. MapReduce: Simplified Data Processing on Large Clusters. OSDI 2004.

Apache Hadoop (2005)

Apache Hadoop 2.0 (2012)

-Vinod Kumar Vavilapalli et al. Apache Hadoop YARN: Yet Another Resource Negotiator, SOCC 2013.

-Separation between resource management and computation model.

Key Features of MapReduce Model

Designed for clouds

-Large clusters of commodity machines

-Designed for big data

-Support from local disks based distributed file system (GFS / HDFS)

-Disk based intermediate data transfer in Shuffling

MapReduce programming model

-Computation pattern: Map tasks and Reduce tasks

-Data abstraction: KeyValue pairs

Iterative MapReduce Model

Twister (2010)

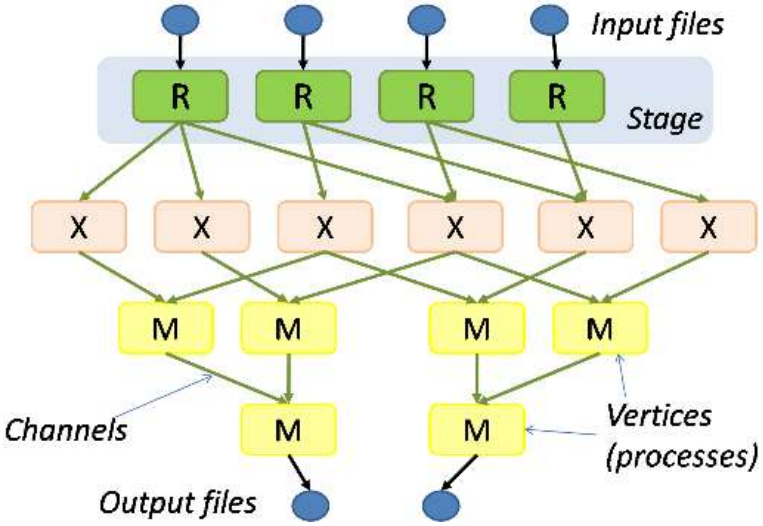
Jaliya Ekanayake et al. Twister: A Runtime for Iterative MapReduce. HPDC workshop 2010.
http://www.iterativemapreduce.org/
Simple collectives: broadcasting and aggregation.

HaLoop (2010)

Yingyi Bu et al. HaLoop: Efficient Iterative Data Processing on Large clusters. VLDB 2010.
http://code.google.com/p/haloop/
Programming model $R_{i+1} = R_0 \cup (R_i \bowtie L)$
Loop-Aware Task Scheduling
Caching and indexing for Loop-Invariant Data on local disk

DAG (Directed Acyclic Graph) Model

- Dryad and DryadLINQ (2007)
  - Michael Isard et al. Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks, EuroSys, 2007.
  - http://research.microsoft.com/en-us/collaboration/tools/dryad.aspx



Collective Model

- Harp (2013)
  - https://github.com/jessezbj/harp-project
  - Hadoop Plugin (on Hadoop 1.2.1 and Hadoop 2.2.0)
  - Hierarchical data abstraction on arrays, key-values and graphs for easy programming expressiveness.

- Collective communication model to support various communication operations on the data abstractions.
- Caching with buffer management for memory allocation required from computation and communication
- BSP style parallelism
- Fault tolerance with check-pointing

Free Big Data Analytics Tools That You Can Use Today

Talend Open Studio for Big Data, free to download and use under an Apache License, provides all you need to easily design and implement big data transfer and big data analytics jobs using Hadoop technologies like HDFS, HBase, Hive, Pig, and Sqoop. The studio features an Eclipse graphical development environment complete with business modeler, meta-data repository, and a palette of configurable components that let you graphically create big data migration and transformation jobs without writing code. Behind the scenes, Talend Open Studio for Big Data generates the underlying code which can be deployed as a scheduled job, as a stand-alone executable, or as a service.

With this fully-functional, feature-rich open source solution, you can quickly get to work with big data and Hadoop. Load data from diverse sources into Hadoop HDFS, HBase, or Hive; execute big data analytics using Hadoop Hive or Pig – all without having to learn or write Hadoop languages or commands.

From Big Data Analytics to Enterprise Big Data Management

For organizations striving for comprehensive management of big data throughout the enterprise, Talend supports seamless migration from Talend Open Studio for Big Data to the subscription-based Talend Platform for Big Data. Talend Platform for Big Data extends Open Studio's big data analytics capabilities with end-to-end enterprise data management functionality including:

- Support for advanced data migration methodologies like change data capture.
- Innovative FileScale technology for running big data analytics on flat files on a single node.
- Time and event-based job scheduling.
- Comprehensive big data quality control components for data profiling, matching, cleansing, and enrichment.
- Support for SOA-enablement of big data services.
- Technical support services from Talend, the world's leading provider of open source data integration technologies.

II. CHARACTERISTICS

Big data can be described by the following characteristics:

Volume – The quantity of data that is generated is very important in this context. It is the size of the data which determines the value and potential of the data under consideration and whether it can actually be considered Big Data or not. The name ‘Big Data’ itself contains a term which is related to size and hence the characteristic.

Variety - The next aspect of Big Data is its variety. This means that the category to which Big Data belongs to is also a very essential fact that needs to be known by the data analysts. This helps the people, who are closely analyzing the data and are associated with it, to effectively use the data to their advantage and thus upholding the importance of the Big Data.

Velocity - The term 'velocity' in the context refers to the speed of generation of data or how fast the data is generated and processed to meet the demands and the challenges which lie ahead in the path of growth and development.

Variability - This is a factor which can be a problem for those who analyse the data. This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

Complexity - Data management can become a very complex process, especially when large volumes of data come from multiple sources. These data need to be linked, connected and correlated in order to be able to grasp the information that is supposed to be conveyed by these data. This situation, is therefore, termed as the 'complexity' of Big Data.

III. ARCHITECTURE

In 2004, Google published a paper on a process called MapReduce that used such an architecture. The MapReduce framework provides a parallel processing model and associated implementation to process huge amounts of data. With MapReduce, queries are split and distributed across parallel nodes and processed in parallel (the Map step). The results are then gathered and delivered (the Reduce step). The framework was very successful, so others wanted to replicate the algorithm. Therefore, an implementation of the MapReduce framework was adopted by an Apache open source project named Hadoop.

Big Data Analytics for Manufacturing Applications can be based on a 5C architecture (connection, conversion, cyber, cognition, and configuration). Big Data Lake - With the changing face of business and IT sector, capturing and storage of data has emerged into a sophisticated system. The big data lake allows an organization to shift its focus from centralized control to a shared model to respond to the changing dynamics of information management. This enables quick segregation of data into the data lake thereby reducing the overhead time

IV. APPLICATIONS

Big data has increased the demand of information management specialists in that Software AG, Oracle Corporation, IBM, Microsoft, SAP, EMC, HP and Dell have spent more than $15 billion on software firms specializing in data management and analytics. In 2010, this industry was worth more than $100 billion and was growing at almost 10 percent a year: about twice as fast as the software business as a whole

V. CONCLUSION

The availability of Big Data, low-cost commodity hardware, and new information management and analytic software have produced a unique moment in the history of data analysis. The convergence of these trends means that we have the capabilities required to analyze astonishing data sets quickly and cost-effectively for the first time in history. These capabilities are neither theoretical nor trivial. They represent a genuine leap forward and a clear opportunity to realize enormous gains in terms of efficiency, productivity, revenue, and profitability. The Age of Big Data is here, and these are truly revolutionary times if both business and technology professionals continue to work together and deliver on the promise.

REFERENCES

[1] www.google.com
[2] www.wikipedia.com