www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 3 March 2017, Page No. 20662-20666
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i3.49

Survey of MapReduce on Big Data

Mr.A.Antony Prakash[1], Dr. A.Aloysius[2]
[1] Asst. Professor in Information Technology
St Joseph's College - Tiruchirappalli- 2
aantonyprakash@gmail.com
[2] Asst. Professor in Computer Science
St Joseph's College - Tiruchirappalli- 2
aloysius1972@gmail.com

Abstract: ***As of recent trends, this is a persistent challenge of integration of cloud computing, data mining and big data mining processes. This paper reveals most recent progress on Hadoop and MapReduce applications. Hadoop and MapReduce can be used for analyzing enormous amount of data. Hadoop is an open source software project used to processing a large data sets. MapReduce is a programming model that associated implementation for parallel processing of large dataset. Big data environment is used to acquire, organize and analyze the various types of data. There is an observation about MapReduce framework. This framework generates large amount of intermediate data.***

Keywords: *component; Big data, Hadoop, HDFS, MapReduce*

1. Introduction

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying and updating information privacy. Big data is a buzzword, or catch-phrase, that is utilized to describe a massive volume of both structured and unstructured data that is so huge that it's complicated to process using traditional database and software techniques. One of the best known methods for turning raw data into useful information is by what is known as MapReduce. MapReduce is a method for taking a large data set and performing computations on it across multiple computers, in parallel. It serves refer to the actual implementation of this model.

In essence, MapReduce consists of two parts ie) Map Function and Reduce Function. The Map function analyzed the data using sorting and filtering techniques. The Reduce function provides a summary of data by combining it all together. While largely credited to research which took place at Google, MapReduce is now a generic term and refers to a general model used by many applications.

1.1 Organization of this paper

This Paper is organized into 4 Sections as follows. Section 2 provides a brief overview of the big data, Hadoop and MapReduce; This Section also reviews and discusses the prior research in the areas of MapReduce frameworks. Section 3 describes the different methodology using Mapping and Reducing in more detail. Section 4 presents and discusses the finding and conclusion for further research.

2. FUNDAMENTAL OF BIG DATA, MAPREDUCE AND HADOOP

2.1 Big Data

Big data contains high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making. Big Data is driving radical changes in traditional data analysis platforms. To perform any kind of analysis on such voluminous and complex data, scaling up the hardware platforms becomes imminent and choosing the right hardware/software platforms becomes a crucial decision if the user's requirements are to be satisfied in a reasonable amount of time.

Researchers have been working on building novel data analysis techniques for big data more than ever before which has led to the continuous development of many different algorithms and platforms. There are several big data platforms available with different characteristics and choosing the right platform requires an in-depth knowledge about the capabilities of all these platforms [1].

2.2 Hadoop

Hadoop plays an important role for storing and retrieving huge set of data. If any system is loses its data, hadoop can be used to retrieve it. Hadoop was designed especially for the analysis of large data sets to build scalable, distributed applications.

To manage sizably voluminous data, Hadoop implements the paradigm called MapReduce defined by Google. Hadoop distributes this data across a set of machines. The real power of Hadoop comes from the fact its competence to scalable to hundreds or thousands of computers each containing several processor cores. Hadoop contains two components that is HDFS (Hadoop distributed file system) and MapReduce [2].

2.3 HDFS (Hadoop Distributed file system)

Hadoop Distributed File System is an open source file system that has been designed specifically to handle large files that traditional file system cannot handle. The large amount of data is split, replicated and scattered on multiple machines. The replication of data facilitates rapid computation and reliability [3] [4].

2.4 MapReduce

MapReduce is an important technology which was proposed by Google. Map Reduce is a software framework for processing large data sets in a distributed fashion over a several machines. MapReduce is a simplified programming model and major component of Hadoop for parallel processing of vast amount of data. It relieves programmers from the burden of parallelization issues while allowing them to freely concentrate on application development. [5] [6]

MapReduce contained two important function for data processing that is Map and Reduce. The core idea behind MapReduce is mapping your data set into a collection of <key, value> pairs, and the reducing overall pairs with the same key. Almost all data can be mapped into <key, value> pairs and your keys and values may be of any type: strings, integers, dummy types and, of course, <key, value> pairs themselves.

3. LITRATURE REVIEW

3.1 Before you Data Aware Caching for Big-Data Applications Using the MapReduce

The author proposes a data aware cache framework for big data application called Dache. In Dache there are using cache manager for intermediate results. A novel cache description scheme and cache request and reply protocol are designed. This paper demonstrates that Dache significantly improves the completion time of MapReduce jobs. Authors' Perspectives: Cache Description, Map Reduce Architecture, Cache Request and Reply, Performance Evaluation [7]. Data-aware caching requires each data object to be indexed by its content [8].

3.1.1 Map Phase Cache Description Scheme:

Map phase Cache refers to the intermediate data produced by worker nodes or processes during the execution of a MapReduce task. A piece of cached data stored in a distributed file system. Content of a cache item is described by the original data and the operations applied. A cache item is described by a 2-tuple: {Origin, Operation}.

3.1.2 Reduce Phase Cache Description Scheme*:*

The input for the reduce phase is a list of key-value pairs, where the value could be a list of values. Original input and the applied operations are required. Original input obtained by storing the intermediate results of the map phase in the distributed file system (DFS).

3.1.3 Findings

Finally the author explain the design and evaluation of a data aware cache framework that requires minimum change to the original MapReduce programming model for provisioning incremental processing for Big data applications using the MapReduce model. Testbed experiment results demonstrate that Dache significantly improves the completion time of MapReduce jobs.Abbreviations and Acronyms

3.2 MATCHMAKING: A NEW MAPREDUCE SCHEDULING TECHNIQUE

The author develops a new MapReduce scheduling technique to enhance map task's data locality and also they develops a new technique to enhance the data locality. The main idea of technique is to assign tasks to a node, local map tasks are always preferred over non-local map tasks, no matter which job a task belongs to, and a locality marker is used to mark nodes and to ensure each node a fair chance to grab its local tasks.

3.2.1 Hadoop default FIFO scheduler:

First the FIFO job order to assign tasks, which means it will not allocate any task from other jobs if the first job in the queue still has an unassigned map task. This scheduling rule has a negative effect on the data locality because another job's local tasks cannot be assigned to the slave node unless the first job has all its map tasks (many of which are non-local to the node) scheduled [9].

Second the data locality is randomly decided by the heartbeat sequence of slave nodes. If we have a large cluster that executes many small jobs, the data locality rate could be quite low.

3.2.2 Delay Scheduling

Zaharia et al. [10] [11] have developed a delay scheduling algorithm to improve the data locality rate of Hadoop clusters. It relaxes the strict job order for task assignment and delays a job's execution if the job has no map task local to the current slave node. To assign tasks to a slave node, the delay algorithm starts the search at the first job in the queue for a local task. If not

successful, the scheduler delays the job's execution and searches for a local task from succeeding jobs.

3.2.3 Findings:

To evaluate the matchmaking scheduling algorithm, they compare it with the Hadoop default FIFO scheduler and the delay scheduling algorithm. Two metrics, i.e., map tasks' data locality rate and average response time, are used for evaluation. Experimental results demonstrate that our matchmaking algorithm can often obtain the highest data locality rate and the lowest average response time for map tasks.

3.3 HADOOP MAPREDUCE FRAMEWORK IN BIG DATA ANALYTICS

The author keeps tabs on MapReduce modifying model, planning undertakings, overseeing and re-execution of the fizzled assignments. Workflow of MapReduce is indicated in this exchange. They explain about the MapReduce framework, workflow in MapReduce, MapReduce functionality, Data organization and scheduling [12].

3.3.1Workflow in MapReduce

The key to how MapReduce works is to take input as, conceptually, a list of records. The records are split among the different computers in the cluster by Map. The result of the Map computation is a list of key/value pairs. Reducer then takes each set of values that has the same key and combines them into a single value [13].

3.3.2 Inputs and Outputs

The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs, and produces a output of the job as set of <key, value> pairs conceivably of distinct types. The key and value classes have to be serializable by the framework and hence need to implement the writable interface[14]. Additionally, the key classes have to implement the Writable comparable interface to facilitate sorting by the framework [15].

3.3.3 Findings:

The MapReduce framework breaks up large data into smaller parallelizable chunks and handles scheduling. If you can rewrite algorithms into Maps and Reduces, and your problem can be broken up into small pieces solvable in parallel, then Hadoop's MapReduce is the way to go for a distributed problem solving approach to large datasets. MapReduce framework is Fault tolerant, decisive and supports thousands of nodes and petabytes of data.

3.4 SURVEY ON MAPREDUCE AND SCHEDULING ALGORITHMS IN HADOOP

MapReduce program is used to collect data as per the request. To process large volume of data proper scheduling is required to achieve greater performance. In this paper explained to study MapReduce and analyze different scheduling algorithms that can be used to achieve better performance in scheduling.

3.4.1 Scheduling Algorithms:

The default scheduling algorithm is based on First in First out (FIFO) order where users jobs are executed. Then after priority criteria to decide the priority of task are added.

3.4.2 FIFO Scheduling:

Its preemptive technique that means the scheduler will kill tasks in pools running over capacity in order to give the slots to the pool running under capacity Priority criteria are also assigned to various pools. Here tasks are scheduled in interleaved manner based on priority [17].

3.4.3Capacity Scheduler:

The Capacity Scheduler was designed to allow organizations to share Hadoop clusters in a predictable and simple manner [18].

3.4.4 Delay Scheduling:

This method performs well in Hadoop workloads because Hadoop tasks are short relative to jobs, and because there are multiple locations where a task can run to access each data block [19].

3.4.5 Findings:

MapReduce and various scheduling algorithms to improve the speed of Hadoop system for data retrieval and job processing. Each of the scheduling method tries to utilize resources like Memory, CPU and job deadlines.

3.5 ACCELERATING HADOOP MAPREDUCE USING AN IN-MEMORY DATA GRID

- EGS Hadoop has been widely embraced for its ability to economically store and analyze large data sets. Using parallel computing techniques like MapReduce, Hadoop can reduce long computation times to hours or minutes. This works well for mining large volumes of historical data stored on

disk, but it is not suitable for gaining real-time insights from live operational data.

- Still, the idea of using Hadoop for real-time data analytics on live data is appealing because it leverages existing programming skills and infrastructure – and the parallel architecture of Hadoop itself.

- This paper describes how real-time analytics using Hadoop can be performed by combining an in-memory data grid (IMDG) with an integrated, stand-alone Hadoop MapReduce execution engine. This new technology delivers fast results for live data and also accelerates the analysis of large, static data sets [20].

- IMDGs can employ multiple storage APIs, such as the Named Cache and Named

Map APIs In both named caches and named maps, applications can create, read, update,and delete objects to manage live data. This gives the application developer the choice to store and analyze heavyweight objects with rich metadata or lightweight objects with highly optimized storage, depending on the type of data being analyzed.

3.5.1 Findings:

Using an IMDG with an integrated Hadoop MapReduce engine opens the door to real-time analytics on live, operational data. The IMDG's integrated MapReduce engine also eliminates the need to install, configure, and manage a full Hadoop distribution. Developers can write and run standard Hadoop MapReduce applications in Java, and these applications can be executed stand-alone by the execution engine. Many applications in e-commerce, financial services, logistics, and other areas now can benefit from the power of real-time analytics by leveraging the widespread expertise focused on Hadoop MapReduce.

3.6 CHALLENGES FOR MAPREDUCE IN BIG DATA

The author identifies MapReduce issues and challenges in handling. The identified challenges are grouped into four main categories corresponding to Big Data tasks types: data storage (relational databases and NoSQL stores), Big Data analytics (machine learning and interactive analytics), online processing, and security and privacy [21].

The challenges for MapReduce in big data are Lack of data storage and support capabilities, Lack of application deployment support, Lack of analytical capabilities in database, Issues in online processing, Privacy and Security challenges.

3.6.1 Data Storage

The advanced of the art for Data Storage and MapReduce is a number of challenges remain, such as:

- The lack of a standardized SQL-like query language
- Limited optimization of MapReduce jobs
- Integration among MapReduce, distributed file system, RDBMSs and NoSQL stores.

3.6.2 Interactive Analytics

Interactive analytics can be defined as a set of approaches to allow data scientists to explore data in an interactive way, supporting exploration at the rate of human thought. Most of these approaches are specialized for certain types of datasets and certain queries and thus provide an open research area for a generalized solution [22].

3.6.3 Data Visualization

A large category of interactive analytics is data visualization. There are two primary problems associated with Big Data visualization [23]. MapReduce for data visualization currently performs well in two cases: memory-insensitive visualization algorithms, and inherently parallel visualization algorithms, Vo et al [24].

3.6.4 Findings:

Issues and challenges MapReduce faces when dealing with Big Data are identified and categorized according to four main Big Data task types: data storage, analytics, online processing, and security and privacy [25]. Moreover, efforts aimed at improving and extending MapReduce to address identified challenges are presented.

4. CONCLUSION

This concept entered an era of Big Data. The paper describes the concept of Big Data along with the various MapReduce techniques, algorithm and concepts. It also focuses on Big Data processing problems. These technical challenges must be addressed for efficient and fast processing of Big Data. In this paper, tried to cover all detail of MapReduce and Hadoop component and future scope.

(1) **5. References**

[1] Agneeswaran VS, Tonpay P, Tiwary J: Paradigms for realizing machine learning algorithms. Big Data 2013, 1(4):207-214.

[2] J. Applications and organizations using Hadoop". Wiki.apache.org. 2013-06-19. Retrieved 2013-10-2017

[3] Lizhe Wang, Jie Tao, Rajiv Ranjan , Holger Marten, AchimStreit , Jingying Chen,DanChen, "G-Hadoop: MapReduce across distributed data centers for data-intensiveComputing", 2013.

[4] WeijiaXu, Wei Luo, Nicholas Woodward,"Analysis and Optimization of Data Import with Hadoop", 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum.

[5] HUANG Lan, WANG Xiao-wei, ZHAI Yan-dong, YANG Bin,"Extraction of User Profile Based on the Hadoop Framework".

[6] Jeffy Dean,Sanjay Ghemawat. MapReduce,"Simplified Data Processing on Large Clusters", OSDI04: Sixth Symposium on Operating System Design and Implemention,Ssn Francisco,CA,December, 2004.

[7] Yaxiong Zhao , Jie Wu, and Cong Liu ,"Dache: A Data Aware Caching for Big-Data Applications Using the MapReduce Framework", Tsinghua Science And Technology February 2014.

[8] Memcached—A distributed memory object caching system, http://memcached.org/, 2013.

[9] Chen He Ying Lu David Swanson,"Matchmaking: A New MapReduce Scheduling Technique", Third IEEE International Conference on Coud Computing Technology and Science,2011.

[10] M. Zaharia et al. "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling". In EuroSys,2010.

[11] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Job scheduling for multi-user mapreduce clusters," EECS Department, University of California, Berkeley, Tech. Rep.,Apr 2009.

[12] Vidyullatha Pellakuri, Dr.D. Rajeswara Rao," Hadoop Mapreduce Framework in Big Data Analytics", (IJCTT) – volume 8 number 3– Feb 2014

[13] HDFS Users Guide - Rack Awareness", Hadoop.apache.org. Retrieved 2013-10-17.

[14] D. Wegener, M. Mock, D. Adranale, and S. Wrobel, "Toolkit-Based High-Performance Data Mining of Large Data on MapReduce Clusters," Proc. Int'l Conf. Data Mining Workshops (ICDMW '09), pp. 296-301, 2009

[15] "Zettaset Launches Version 4 of Big Data Management Solution, Delivering New Stability for Hadoop Systems and Productivity Boosting Features",Zettaset.com 2011-12-06. Retrieved 2012-05-23.

[16] Harshawardhan S. Bhosle, Devendra P. Gadekar : "Big Data Processing Using Hadoop:Survey on Scheduling",IJRS, Volume 3 Issue 10, 2014.

[17] DeWitt & Stonebraker, "MapReduce: A major step backwards", 2008.

[18] Hadoop'scapacityscheduler:http://hadoop.apache.org /core/docs/current/capacity/scheduler.

[19] Mark Yong, Nitin Garegrat, Shiwali Mohan: "Towards a Resource Aware Scheduler in Hadoop" in Proc. ICWS, 2009, pp:102-109.

[20] By David L. Brinker and William L. Bainc," Accelerating Hadoop MapReduce Using an In-Memory Data Grid", ScaleOut Software, Inc, 2013.

[21] Katarina Grolinger1, Michael Hayes1, Wilson A. Higashino1,2, Alexandra L'Heureux1 David S. Allison1,3,4,5, Miriam A.M. Capretz," Challenges for MapReduce in Big Data",Electrical and Computer Engineering Publications,2014.

[22] J. Heer and S. Kandel, "Interactive analysis of Big Data," XRDS: Crossroads, the ACM Magazine for Students, 19(1), pp. 50-54, 2012.

[23] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar and R. Pasquin, "Incoop: MapReduce for incremental computations," Proc. of the 2nd ACM Symposium on Cloud Computing, 2011.

[24] W. Zeng, Y. Yang and B. Luo, "Access control for Big Data using data content," IEEE International Conference on Big Data, 2013.

[25] J. Fan, F. Han and H. Liu, "Challenges of Big Data analysis," National Science Review, in press, 2014.

Author Profile

A.ANTONY PRAKASH is working as an Assistant Professor in the Department of Information Technology, St.Joseph's college(Autonomous), Tiruchirappalli, TamilNadu, India. I am having 6 years of experience in teaching and 2 years in research

A. Aloysius is working as an Assistant Professor in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has 16 years of experience in teaching and research. He has published many research articles in the National / International conferences and journals. He has acted as a chairperson for many national and international conferences. Currently, eight candidates are pursuing Doctor of Philosophy Programmed under his guidance.