ISSN 2395-1621

Improved Performance of Hadoop using Efficient Data Aware Caching for Big Data

#1Kunjan Patle, #2Priya Pallavi, #3Shriya Mahajan, #4Mayur Kulkarni,
#5Mrs. Nilam Patil

[1]kunjanpatle@yahoo.com
[2]priya.keshri21@gmail.com
[3]shriyamahajan17@gmail.com
[4]mayurkulkarni16@gmail.com

#12345Department of Computer Engineering,
D.Y.Patil College of Engineering, Akurdi, Pune.

ABSTRACT

New method to improve the performance of MapReduce by using distributed memory cache as a high speed access between map tasks and reduce tasks. Map outputs sent to the distributed memory cache can be gotten by reduce tasks as soon as possible. Experiment results will show that our prototype's performance is much better than that of the original on small scale clusters. To our knowledge, this is the first effort to accelerate MapReduce with the help of distributed memory cache.The main objective is, implemented and integrated with Hadoop works for improving performance. Thus to prove successful in lowering job execution times in the overall system. In traditional system data is issued from disk reducing the overall performance. Therefore an attempt is made to cache data and issue from cache like remote caches. The contributions added to improve the system are basically remote memory caching, more da ta local jobs and reference caching. In remote memory caching, caching of input data at the DataNode level lowers job execution time. Efficient data aware caching system is implemented by using Hadoop incorporated components. Cache manager communicates with task trackers and provides cache items on receiving requests which is implemented in the system. The cache manager uses HDFS, the DFS component of Hadoop, to manage the storage of cache items. In order to access cache items, the mapper and reducer tasks first send requests to the cacher and manager which accepts only key value components.The component TaskTracker class is responsible for managing tasks, understand file split and bypass the execution of mapper classes entirely. Task Tracker also manages reducer tasks and bypass reducer tasks by utilizing the cached results.

ARTICLE INFO

Article History

Received: 28th May 2018

Received in revised form :

28th May 2018

Accepted: 30th May 2018

Published online :

31st May 2018

Keyword: Map Reduce, Hadoop cache, Distributed cache

I. INTRODUCTION

Efficient data aware caching system is implemented by using Hadoop incorporated components. Cache manager communicates with task trackers and provides cache items on receiving requests which is implemented in the system. The cache manager uses HDFS, the DFS component of Hadoop, to manage the storage of cache items. In order to access cache items, the mapper and reducer tasks first send requests to the cache manager. Mapper and Reducer classes only accept key value pairs as the inputs which are fixed by Hadoop interface. An open accessed component Input Format class allows application developers to split the input files of the MapReduce job to multiple file splits and parse data to key value pairs. The component TaskTracker class is responsible for managing tasks, understand file split and bypass the execution of mapper classes entirely. TaskTracker also manages reducer tasks and bypass reducer tasks by utilizing the cached results.

II. LITERATURE SURVEY

Big data is evolving drastically around us. Researchers, scholars defined big data depending on different perspectives. The very common definition of big data is that the datasets that could not be imagined, understood easily, accomplished and processed by conventional information technology software/ hardware tools in an expected time.

The volume of information increasing every day as people create such large data with the help of communications like voice calls, emails, texts, uploaded pictures, video. MapReduce is used by researchers at Google from 2004. Due to limitless features of big data, researchers understood that a single machine is unable to serve all data computation/analytic solutions, and new environment like distributed system is required to process and store data in parallel [9]. An open source implementation Apache Hadoop similar to MapReduce became available with free of cost for large scale data analytics, big-data applications and other major parallel computations in which large input data is required. Hadoop is adopted by several distinguished and renowned companies like Yahoo!, Facebook and became mainstay [10]. The Hadoop computational model has several distinguished attributed properties. It is simple that its API stipulates a few entry points for the application programmer specified mappers, reducers/combiners, partitioners, for formatting input and output[11]. In addition to basic large scale computational models, lot of software tools is built around as Hadoop ecosystem. These tools are such as Apache Hive, Apache Giraph, Apache Hama, Apache Mahout all of which harness Hadoop [12]. Since the last ten fifteen years it is observed that Hadoop clusters size is increased, as well as increase in size of RAM memory supported by each machine. The smaller ¡K, Vj size, high amount of data reusability and Hadoop Job interactive ness make it possible to build a robust caching mechanism preferably in - memory for substantial improvement in performance [13].

III. PROBLEM STATEMENT

In traditional system data is issued from disk reducing the overall performance. Therefore an attempt is made to cache data and issue from cache like remote caches. The contributions added to improve the system are basically remote memory caching, more data local jobs and reference caching. In remote memory caching, caching of input data at the DataNode level lowers job execution time.

IV. METHODOLOGY

Big Data is difficult to work with using most relational database management systems, desktop statistics and visualization packages since it requires massive parallel software running on tens, hundreds or even thousands of servers. Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. Hadoop has 2 main components: HDFS (Hadoop Distributed File System) and Map-Reduce. HDFS is a block structured distributed file system for storing large volumes of data. MapReduce are programming model meant for large clusters. It has a parallel computing framework helps in parallelization, fault tolerance, data distribution and load balancing. The computation of Map-Reduce takes a set of input key/value pairs and generates a set of output key/value pairs. The computation of generating the set of output key/value pairs is divided into two functions: Map function and Reduce function.

Block Diagram:

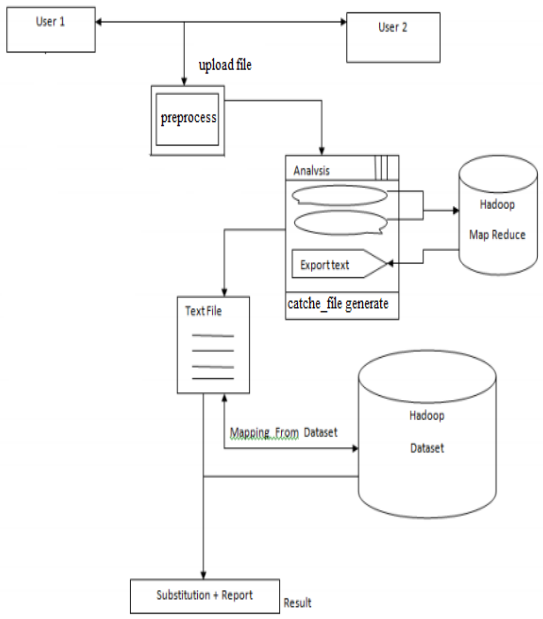
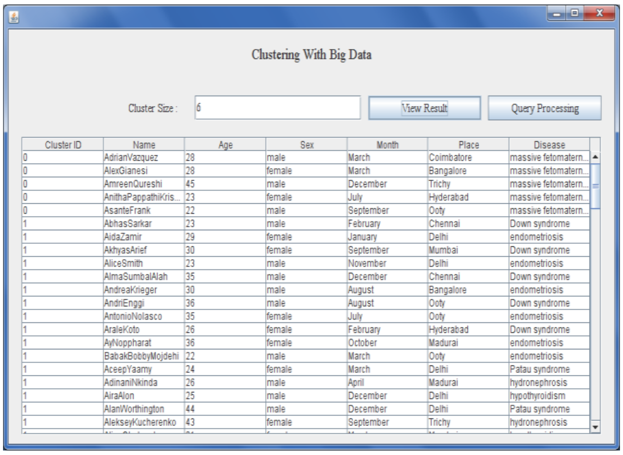


Fig 1. System architecture

V. RESULT



Cluster Size : 6

Cluster ID	Name	Age	Sex	Month	Place	Disease
0	AdrianVazquez	28	male	March	Coimbatore	massive fetomatern...
0	AlexGianesi	28	female	March	Bangalore	massive fetomatern...
0	AmreenQureshi	45	male	December	Trichy	massive fetomatern...
0	AnithaPappathiKris...	23	female	July	Hyderabad	massive fetomatern...
0	AsanteFrank	22	male	September	Ooty	massive fetomatern...
1	AbhasSarkar	23	male	February	Chennai	Down syndrome
1	AidaZamir	29	female	January	Delhi	endometriosis
1	AkhyasArief	30	female	September	Mumbai	Down syndrome
1	AliceSmith	23	male	November	Delhi	endometriosis
1	AlmaSumbalAlah	35	male	December	Chennai	Down syndrome
1	AndreaKrieger	30	male	August	Bangalore	endometriosis
1	AndriEnggi	36	male	August	Ooty	Down syndrome
1	AntonioNolasco	35	female	July	Ooty	endometriosis
1	AraleKoto	26	female	February	Hyderabad	Down syndrome
1	AyNoppharat	36	female	October	Madurai	endometriosis
1	BabakBobbyMojdehi	22	male	March	Ooty	endometriosis
1	AceepYaamy	24	female	March	Delhi	Patau syndrome
1	AdinaniNkinda	26	male	April	Madurai	hydronephrosis
1	AiraAlon	25	male	December	Delhi	hypothyroidism
1	AlanWorthington	44	male	December	Delhi	Patau syndrome
1	AlekseyKucherenko	43	female	September	Trichy	hydronephrosis

Fig 2. Clustered result

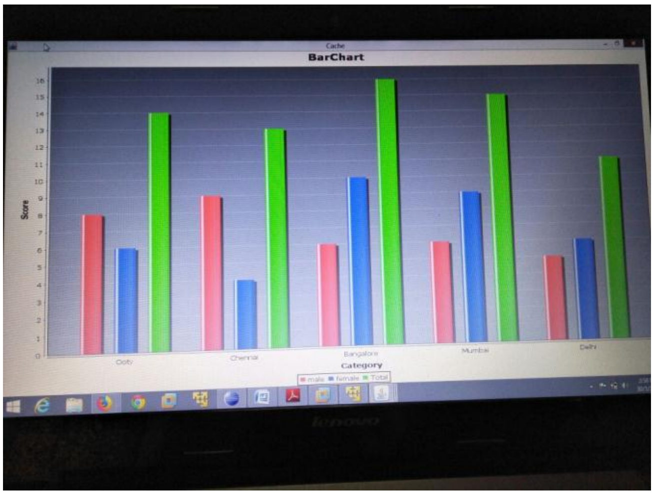


Fig 3. Analysis graph

VI. CONCLUSION

The proposed efficient data aware caching framework is powerful for cache management. The new cache replacement algorithm is implemented and called it as value degree to calculate the value of tuple being replaced. Results show that there is substantial improvement in performance of Hadoop jobs by reducing completion time and storage overhead using efficient data aware caching for big data application. The future development will focus on enhancing the caching mechanism with advanced algorithm using the hadoop and MapReduce.

REFERENCES

[1]. Yaxiong Zhao, Jie Wu, and Cong Liu “Dache: A Data Aware Caching for Big-Data Applications Using the MapReduce Framework” Tsinghua Science and Technology ISSNl l1007-0214l l05/10l lpp39-50 Volume 19, Number 1, February 2014.

[2]. Zhu Xudong, Yin Yang, Liu Zhenjun, and Shao Fang “C-Aware: A Cache Management Algorithm Considering Cache Media Access Characteristic in Cloud Computing”, Research Article, Hindawi Publishing Corporation, Mathematical Problems in Engineering, Article ID 867167, 13 pages, Volume 2013.

[3]. Meenakshi Shrivastava, Dr. Hans-Peter Bischof “Hadoop-Collaborative Caching in Real Time HDFS” Computer Science, Rochester Institute of Technology, Rochester,NY, USA.

[4]. Dachuan Huang, Yang Song, Ramani Routray, Feng Qin “SmartCache: An Optimized MapReduce Implementation of Frequent Itemset Mining” The Ohio State University, IBM Research – Almaden.

[5]. Yingyi Bu, Bill Howe, Magdalena Balazinska, Michael D. Ernst “HaLoop: Efficient Iterative Data Processing on Large Clusters” Department of Computer Science and Engineering University of Washington, Seattle, WA, U.S.A. 36th International Conference on Very Large Data Bases, September 1317, 2010, Singapore.

[6]. Ganesh Ananthanarayanan, Ali Ghodsi, AndrewWang, Dhruba Borthakur, Srikanth Kandula, Scott Shenker, Ion Stoica “PACMan: Coordinated Memory Caching for Parallel Jobs” University of California, Berkeley, Facebook, Microsoft Research, KTH/Sweden.

[7]. Gurmeet Singh, Puneet Chandra and Rashid Tahir “A Dynamic Caching Mechnism for Hadoop using Memcached” Department of Computer Science, University of Illinois at Urbana Champaign.

[8]. Executive Office of the President “ Big Data: Seizing Opportunities, Preserving Values” May 2014.14 Min Chen, Shiwen Mao, Yunhao Liu “Big Data: A Survey” Published online: 22 January 2014, Springer Science+Business Media New York 2014.

[9]. Yaxiong Zhao, Jie Wu, and Cong Liu “Dache: A Data Aware Caching for Big-Data Applications Using the MapReduce Framework” Tsinghua Science and Technology ISSNl l1007-0214l l05/10l lpp39-50 Volume 19, Number 1, February 2014.16 Avraham Shinnar, David Cunningham, Benjamin Herta, Vijay Saraswat “M3R: Increased Performance for InMemory Hadoop Jobs”, roceedings of the VLDB Endowment, Vol. 5, No. 12,38th International Conference on Very Large Data Bases, Istanbul,Turkey, August 27th 31st 2012.

[10]. Tom White “Hadoop: The Definitive Guide”,Third edition,Oreilly, ISBN: 978-1-449-31152-0.

[11]. Venkatesh Nandakumar “Transparent in-memory cache for Hadoop-MapReduce” Athesis submitted in conformity with the requirements for the degree of Master of Applied Science Graduate Department of Electrical and Computer Engineering University of Toronto, 2014.