Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Review Paper on Data-aware Caching for Big Data Applications

R. Udendran, Dr. Govindaraj, P. Kannan
Department of Computer Science and Engineering
Bharathidasan University, Tamil Nadu, India

Abstract: This paperreports review on Data-aware Caching for Big data applications. The term big data deals with extensively large sets of data which the data may be structured or unstructured .MapReduce has been the centre piece ofApache Hadoop software tool designed for the distributed processing and storage of these huge datasets. Research is still done in this field to improve the performance of MapReduce framework and many designs, techniques are proposed .The aim of this paper is to expose a data aware cache for Big data applications.

Keywords: Big data , Apache Hadoop, cache, MapReduce, Distributed File System.

I. INTRODUCTION

The main reason of big data existence is when the traditional relational databases management systems were not capable of processing the unstructured data . Normally big data are measured in Zettabytes and Terabytes. The Big data is a large corpse of data on which applications works on eccentrically enormous amount of data. There are several challenges in Big data which includes sharing, storage , visualization, analysis, privacy and security. Big data analytics has become all the rage. The Hadoop MapReduce is an open source software framework developed by Apache which assists in distributed processing of larger data sets across clusters of commodity servers. The three main components of Apache Hadoop 2 are Hadoop Distributed File System (HDFS), Yet Another Resource Negotiator (YARN) and MapReduce framework. The MapReduce is a model for processing huge amount of data sets ,Hadoop MapReduce is a software framework for processing applications which possess vastamounts of data. A MapReduce job usually splits the input data-set intoindependent chunks which are processed by the map tasks ina completely parallel manner. The framework looks after schedulingtasks, re-executes the unsuccessful tasks and also monitors them.

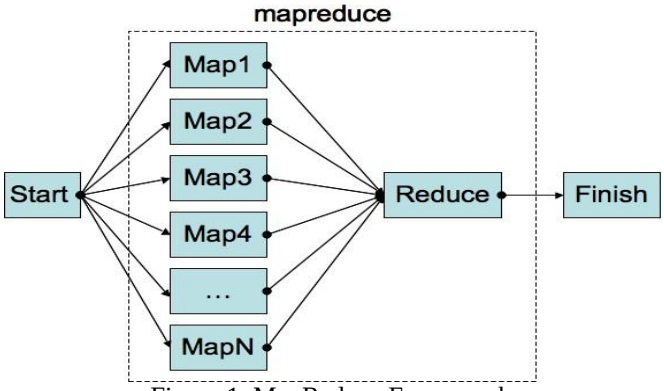

Figure 1. MapReduce Framework

From the Figure 1, it is clear that MapReduce has two phase, namely a map phase and a reduce phase .Map phase is responsible for sorting, filtering whereas the actual operation on data takes place in the reduce phase. Even though MapReduce is an easy tool for processing Big data, it lacks some performance in real-time processing, faster responses, iterations. A novel concept is described in Data-aware caching for big Data which uses MapReduce Framework (Dache).In Dache, the tasks submit their intermediate results to a component known as cache manager. The Dache enhances the working of Google's MapReduce and Apache's Hadoop where these applications produce lots of intermediate results. Dache eliminates these kinds of problems persisting in these applications. The Dache encompasses the following activities: Map phase, Reduce phase, Cache item submission, Life-Time management of cache item, Cache request and reply.

II. PROPOSED SYSTEM

Data-aware cache (Dache) intention is to strengthen and improve the processing of MapReduce framework. It provides a cache layer for finding, accessing and processing cache items in a MapReduce job. First of all, Dache needs its each data object to be graded by its content. In Dache, a cache item represents an intermediate result. The cache manager is the main part, whenever a file is splited the cache manager is responsible to inform the previous file splitting outline used in a cache item.

A). HDFS Architecture:

The Hadoop Distributed File System is a unique file system where it is designed to convert a cluster of servers into huge storage system with self-healing distributed file system and fault tolerant, load balancing and higher security level. The working components of HFDS are NameNode, DataNode, SecondaryNode which are shown in figure 2.

NameNode: It is the most important and centremost part, the purpose of NameNode is that it interacts with the client application for locating, moving, copying, deleting or even adding files. It performs by giving back a listed of related DataNode servers in which a file or a data resides.

DataNode: It is used for storing data, a functional file system consists of more DataNodes.

Secondary NameNode: It acts as a frontier for the NameNode and it's a backup for NameNode.

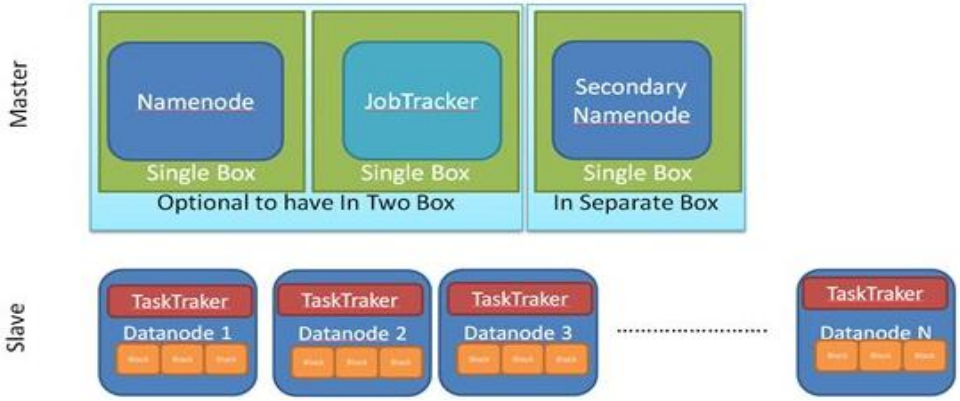

Figure 2: Apache Hadoop Components.

B). MapReduce Architecture:

The MapReduce consists of Job client, Job tracker, Task tracker as shown in figure 3. Each of these components interacts with each other and mainly also with the HFDS to produce the required outcome.

Job client: This is the client where jobs are submitted for processing.

Job Tracker: The Job tracker helps in co-ordinating jobs which is flawless and keeps track on various jobs or process taking place.

Task Tracker: It helps to execute job tasks.

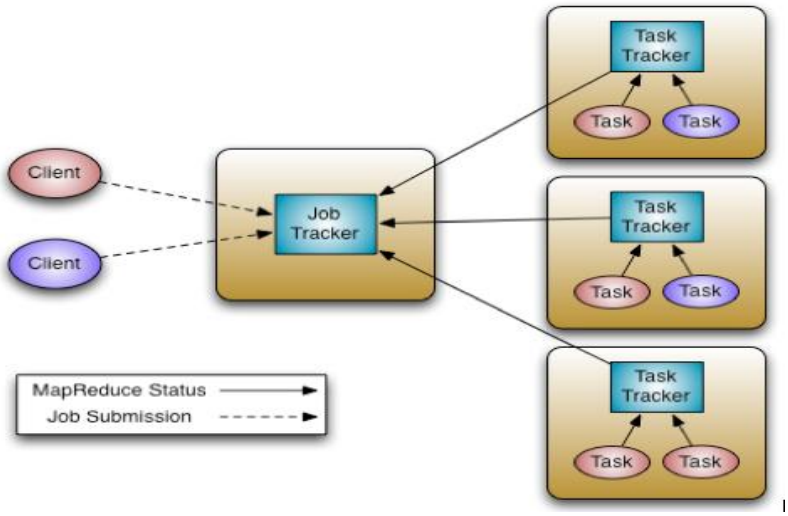

Figure 3: MapReduce Architecture

III. PROTOCOL

Dache has classified cache items into types namely, map cache and reduce cache. These two types interact with each other in different scenarios and there are some complexities when it comes to sharing. In map cache, sharing becomes effortless since the operations enforced are well known but in reduce phase sharing becomes a complex task.

A). Map phase description:

The cached item which is stored in HDFS is represented by 2-tuple :{ origin, operation} where origin is the name of the file and the liner lists of operations performed are:

Item count**:** counts all records of the input.

Sort**:** sorts the record of a file.

Selection**:** selects an item that meets a requirement.

Transform**:** changes each item in the file into a varied item.

Classification**:**splits the input items into multiple groups.

B). Reduce phase description:

Here the reduce phase employs partition method in which partition occurs according to criteria, the partition method helps in determining similar data records which are processed together. The input is a list of key value pairs , the final input is the cached results of the processed data obtained from map phase.

C). Cache item submission:

The cache item is recorded by the mapper and reducer nodes. As soon as an operation is completed, the cache item is promoted to the cache manager. Inorder to improve the data locality, cache item and the worker node are put on the same machine. Each time before processing an input data file, the worker node contacts the cache manager and sends the file name and the plans for operations that will be performed on that file. Upon receiving, the cachemanagerstarts to compare the plans with the stored mapping data and when the moment it finds the precise match, a tentative description is sent by the cache manager to the worker node and the worker node fetches the cache item.

D). Life-time management of cache item:

The cache manager should be able to find how much time a cache item can reside in HDFS. If a cache item is kept for a longer period, will eventually result in storage wastage. Dache provides two types of policies in life-time management of a cache item:-

Fixed storage quota: For a cache item, dache provides a fixed amount of storage space , older cache items are replaced by new cache items. The Least Recent Used method is employed in fixed storage quota.

Optimal Utility: Inorder to find the optimal space for storing cache items, a utility based measurement is used. The following equations show the method to find expenses and saved expenses for storing a cache item.

$$\text{rage}^{\times S}\text{cache}^{\times t}\text{s} \qquad (1)$$

$$_{\text{tt}}\text{ation}\times\mathbf{R}\text{duplicate}\times\text{ts} \qquad (2)$$

e).Cache request and reply:

The cache request and reply takes place in both map cache and reduce cache.

Map cache: - Before commencing the file splitting phase, the job tracker is responsible for sending out cache requests to the cache manager. In return, the cache manager responses with a list of cache descriptions.

Reduce cache: - The requested cache item is compared with the cached items in the cache manager's database. Cache manager identifies the overlaps of the original input files of the requested cache and stored cache and for this purpose linear scan method is used.

IV. PROPOSED FRAMEWORK TEST BED AND REQUIREMENTS

The cache manager is deployed in separate server. Hadoop which is pseudo-distributed mode executed on a server which the server should have an 8-crore CPU with a running of 3GHz, 16GB of memory and SATA disk. The number of mappers are fixed which is 16 throughout the experiments, but the count of reducers varies. Inorder to measure the speed of Dache, word-count and tera-sort applications are used.

V. CONCLUSION

The data-aware cache for big data applications reduces execution time, CPU utilization, and alters the MapReduce programming model, so that there is only less modification to it. The dache main improvement is that it eliminates duplicate tasks in incremental processing. But Dache also has its own drawbacks, where it requires maximum amount of cache and it also needs a better life-time management of cache.

ACKNOWLEDGMENT

The author would like to thank Mr K MuthuRamalingam- Assistant Professor of Bharathidasan University and Dr.Akila-Professor of Bharathidasan University for their valuable suggestions during the planning and development of this review work. Their willingness to give their time is appreciated.

REFERENCES

[1] KudakwasheZvarevashe and Dr A VinayaBabu, "Towards Mapreduce performance optimization: A look into the optimization techniques in Apache Hadoop for big data Analytics". International Journal of Science and Research, ISSN:2319-7064 , 2012.

[2] Z. Zheng, J. Zhu, M. R. Lyu. "Service-generated Big Data and Big Data-as-a-Service: An Overview," in Proc. IEEE Big Data, pp.403-410, October 2013.

[3] Hadoop, http://hadoop.apache.org/,2013

[4] Yaxiong Zhao, Jie Wu, and Cong Liu, "Dache: A Data Aware Caching for Big- Data Applications Using the MapReduce Framework" , TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 05/10 Volume 19, Number 1, pp 39- 50, February 2014

[5] SheetalThokal and Prof.VrundaBhusari, "Review Paper on Clustering Based Collaborative Filtering",International Journal of Advance Research inComputer Science and Management Studies,ISSN:: 2321-7782

[6] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica, "Improving Mapreduce performance in heterogeneous environments",in Proc. of OSDI' 2008, Berkeley, CA, USA, 2008.

AUTHORS PROFILE

Mr. Udendhran is pursuing M.tech (integrated) in cse at Bharathidasan University. He has presented research papers in International conference and national conference and also published papers in reputed journals.

Prof Govindaraj M (M Sc., M. Tech) is a lecturer in Bharathidasan University. He has published many international research papers in reputed journals and also conducted many workshops in colleges and universities.

Mr. Kannan is pursuing M.tech (integrated) in cse at Bharathidasan University. He has also presented research papers and published papers in reputed journals

.

