International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

MAP Reduce In Mobile Clouds Using Hadoop: MDFS (Mobile Distributed File System) Addresses Issues for Big Data Processing in Mobile Clouds

K. Swarupa Rani & K. Anitha
[1]Assistant Professor, Dept of IT, Prasad V Potluri Siddhartha Institute Of Technology, Kanuru, Vijayawada, A.P., India.
[2]Lecturer, Dept. of Computer Science, Sri Durga Malleswara Siddhartha Mahila kalasala,Vijayawada, A.P., India.

Abstract — Hadoop is a scalable platform that provides distributed storage and computational capabilities on clusters of commodity hardware. Building Hadoop on a mobile net-work enables the devices to run data intensive computing applications without direct knowledge of underlying distributed systems complexities. Building Hadoop on a mobile network enables the devices to run data intensive computing applications without direct knowledge of underlying distributed systems complexities. However, these applications have severe energy and reliability constraints (e.g., caused by unexpected device failures or topology changes in a dynamic network). As mobile devices are more susceptible to unauthorized access, when compared to traditional servers,security is also a concern for sensitive data. Hence, it is paramount to consider reliability, energy efficiency and security for such applications. The MDFS (Mobile Distributed File System) addresses these issues for big data processing in mobile clouds. We have developed the Hadoop MapReduce framework over MDFS and have studied its performance by varying input workloads in a real heterogeneous mobile cluster. Our evaluation shows that the implementation addresses all constraints in processing large amounts of data in mobile clouds. Thus, our system is a viable solution to meet the growing demands of data processing in a mobile environment.

Keywords *— Hadoop, MCC, MapReduce, MDFS.*

1.INTRODUCTION

Current mobile applications that perform massive computing tasks (big data processing) o oad data and tasks to data centers or powerful servers in the cloud. There are several cloud service o erings of computing infrastructure to end users for processing large datasets. Hadoop MapReduce is a popular open source programming framework for cloud computing. The framework splits the user job into smaller tasks and runs these tasks in parallel on di erent nodes, thus reducing the overall execution time when compared with a sequential execution on a single node. This architecture however, fails in the absence of external network connectivity, as it is the case in military or disaster response operations. This architecture is also avoided in emergency response scenarios where there is limited connectivity to cloud, leading to expensive data upload and download operations.MCC (Mobile Cloud Computing) is a

combination of Cloud Computing and Mobile technology with the help of the internet, which has progressed over time from reading emails and playing games to learning center on phones with help of applications, tracking a person's health and to entertainment. Increase in the number of mobile phones and usage of internet by every user amounts to large datasets which can help companies to make products better and give benefits to different user as per needs. Features like GPS, Accelerometer, Gyroscope, and Digital Compass are to be considered when we talk about MCC as these features can bring many applications to the table, but there are few challenges that exist in MCC which brings down the efficiency of at ask when deploy ed to it. Such disadvantages are as follows:

• Security: When everyone's data is stored on multiple servers around the globe, it is a big challenge for them to keep it secure.

• Battery: Once the battery is drained from mobile phones, all the features of MCC become void.

• Presentation: The information that can be portrayed on bigger screen compared to the screen on mobile phones are still limited due to it screen size, the accessibility to a graphical user interface such as viewing of an image and working on multiple screens.

• Cross Platform: With so many providers, the information that can be used to analyze is scattered due to business proposition, and the need to bring all the providers into one platform is a task that is difficult to achieve.

Hadoop MapReduce:

Hadoop MapReduce is a software framework for easily writing applications which process big amounts of data inparallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The term MapReduce actually refers to the following two different tasks that Hadoop programs perform:

The Map Task: This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).

The Reduce Task: This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

2. SYSTEM IMPLEMENTATION

Our MDFS framework consists of 18,365 lines of Java code, exported as a single jar le. The MDFS code does not have any dependency with the Hadoop code base. Similar to DistributedFileSystem class of HDFS, MDFS provides MobileDistributedFS class that implements FileSystem, the abstract base class of Hadoop for backwards com-patibility of all present HDFS applications. The user invokes this object to interact with the le system. In order to switch from HDFS to MDFS, the Hadoop user only needs to add the location of jar le to the HADOOP CLASSPATH variable and change the le system scheme to `mdfs'. The parameter `mdfs.standAloneConf'(set to false by default) determines the MDFS architecture to be instantiated. If it is set to false, all the servers are started locally as in the distributed architecture. If it is set to true, the user needs to additionally con gure the parameter `mdfs.nameservice.rpc-

address'in the con guration le. It speci es the location of Name Server in the cluster. In the present implementation, the Fragment Mapper is started in the same node as the Name Server. Since no changes are required in the existing code base for MDFS integration, the user can upgrade to a di erent Hadoop release without any con icts.

3.EXISTING SYSTEM

- Current mobile applications that perform massive computing tasks (big data processing) offload data and tasks to data centers or powerful servers in the cloud. There are several cloud services that offer computing infrastructure to end users for processing large datasets.
- The previous research focused only on the parallel processing of tasks on mobile devices using the MapReduce framework without addressing the real challenges that occur when these devices are deployed in the mobile environment.

DISADVANTAGES:

Fails in the absence of external network connectivity, as it is the case in military or disaster response operations.

- This architecture is also avoided in emergencyresponse scenarios where there is limited connectivity tocloud, leading to expensive data upload and downloadoperations.
- Traditional security mechanisms tailored for static networks are inadequate for dynamic networks.
- Existing ignores energy efficiency. Mobile devices have limited battery power and can easily fail due to energy depletion
- HDFS needs better reliability schemes for data in the mobile environment.

PROPOSED SYSTEM:

- In this paper, we implement Hadoop MapReduce framework over MDFS and evaluate its performance on a general heterogeneous cluster of devices. We implement the generic file system interface of Hadoop for MDFS which makes our system interoperable with other Hadoop frameworks like HBase. There are no changes required for existing HDFS applications to be deployed over MDFS.
- We propose the notion of blocks, which was missing in the traditional MDFS architecture. In our approach, the files are split into blocks based on the block size. These blocks are then split into fragments that are stored across the cluster. Each block is a normal Unix file with configurable block size. Block size has a direct impact on performance asit affects the read and write sizes.

ADVANTAGES OF PROPOSED SYSTEM:

- To the best of our knowledge, this is the first work to bring Hadoop MapReduce framework for mobile cloud that truly addresses the challenges of the dynamic network environment.
- Our system provides a distributed computing model for processing of large datasets in mobile environment while ensuring strong guarantees for energy efficiency, data reliability and security.

4.HADOOP MAPREDUCE FOR TACTICAL CLOUDS

This paper discusses about merging two technologies Hadoop MapReduce and MDFS (Mobile Distributed File System). The paper establishes the theory suggesting traditional or proposed works are not secure like Misco and

Hyrax. In hyrax, Datanode and TaskTracker of Hadoop environment were ported to android for distributed computing among various devices.5 MDFS is similar to HDFS but has focused its implementation on mobile devices. MDFS was initiated for military operations where all the data among soldiers could be collected and be secured in an event of loss of device and faster computation. These security measures are done by adapting k out of n approaches (k storage nodes, n-total number of nodes). In Figure 4 it is shown how every block is encrypted with a secret key. The secret key is not stored in local database and for further secure measurements, the key is divided into parts using Shamir's Key sharing algorithm.

5.RELATED WORK

This paper focuses on combining the task of data collection from various mobile devices and performs analytics on them. MBD (Mobile Big Data) is referred to be a growing opportunity for people to enhance the business aspects, along with healthcare and fraud detection. Since parallel computing is a requirement in a majority of operations taking place in big data, the authors suggest how Spark technology can be used instead of Hadoop among mobile devices combining with Deep Learning. Comparison between Hadoop and Spark has shown Spark to have the upper hand and being faster by approximately 2.5x times7 . Deep Learning: This is a branch of machine learning in which it learns from new inputs that are compared with old inputs of the user. It classifies every input into a different category based on an algorithm and later suggests its result for solutions of other similar options or problems faced by other users. There are some open issues regarding this:

• Volume: With the increase in mobile data, how to handle a number of resources required to process the operations when it varies from time to time is an issue.

• Portability: Since every mobile phone isn't in a stationary location, the collection of samples from various devices becomes a major issue.

• Crowdsourcing: Due to many mobile users the collected data varies, due to this the quality of analytics or result produced in analysis cannot be assured, as it deals with different sets of preferences and usage depending on the its users.

• Training: Algorithms can learn the behavior based on some sample data or training data sets, but the data received from the devices are usually unlabeled and sample data are labeled sets of data.

• Time Utilization: The algorithm running on a mobile phone is slower compared to bigger machines which are dedicated for such purposes due to the volatile nature of MBD. Learning patterns or deep models with mobile big data is slow in real time, as to rely on mobile resources for computing some algorithm which a standalone machine can do within a day compared to mobile which will take days is not reasonable. The paper therefore uses a technology called Spark which has shown improvement and growth in big data computation. Spark supports many features such as fault tolerance and recovery of operations. In Cloud computing, the computational and storage resources are often under-utilized, thereby increases further adoption of cloud services. A volunteer cloud infrastructure, called Ad-hoc Cloud Computing will allow cloud services to an existing heterogeneous hardware and also improves the resource utilization and yields significant cost savings. In mobile adhoc network, mobile nodes self-organize themselves to create a

network without the support of any infrastructure like base stations. Users will make use of the mobile phones as a personal information processing tool. But some mobile lacks in resources when compared to smart phones. To overcome this limitation, the adhoc mobile clouds are implemented, which allows the cloud computing resources in the smart phones to be used by resource-poverty or resource hungry mobile phones. The adhoc mobile clouds are created using the mobile devices in the vicinity of users. Thus, in contrast to the data center cloud model, resource provisioning level are not established a priori, nor are resource committed exclusively to the cloud while in use but for a small proportion of the time.

Research challenges :

this section describes the challenges involved in the implementation of MapReduce framework over MDFS.

1. Traditional MDFS architecture only supports a flat hierarchy. All files are stored at the same level in the file system without the use of folders or directories. But the MapReduce framework relies on fully qualified path names for all operations.

2. The capabilities of traditional MDFS are very limited. It supports only a few functionalities such as read(), write() and list(). A user calls the write() function to store a file across the nodes in the network and the read() function to read the contents of a file from the network. The list() function provides the full listing of the available files in the network. However, MapReduce framework needs a fairly generic file system that implements wide range of functions. It has to be compatible with available HDFS applications without any code modification or extra configuration.

3. MapReduce framework needs streaming access to their data but, MDFS reads and writes are not streaming operations.

4. During the job initialization phase of Hadoop, JobTracker queries the NameNode to retrieve the information of all the blocks of the input file (blocks and list of DataNodes that store them) for selecting the best nodes for task execution. JobTracker prioritizes data locality for TaskTracker selection. It first looks for an empty slot on any node that contains the block.

CONCLUSION

In this paper, The Hadoop MapReduce framework over MDFS demonstrates the capabilities of mobile devices to capitalize on the steady growth of big data in the mobile environ-ment. Our system addresses all the constraints of data processing in mobile cloud -energy e_ciency, data reliability and security. The evaluation results show that our system is capable for big data analytics of unstructured data like media _les, text and sensor data. This paper focuses on trying to connect to a maximum number of small devices to distribute the workload and not be dependent on larger machines when an idle resource can be utilized around the world. But it also has some open issues that need to be resolved for better usage and adaptation. Due to rise in privacy matters, people don't like sharing information about themselves as they believe it will be used against them in the future if the information is acquired by an unknown party

REFERENCE:

1.Johnu George, Chien-An Chen, Radu Stoleru, *Member, IEEE,* Geoffrey G. Xie *Member, IEEE*, "Hadoop MapReduce for Mobile Clouds", IEEE Transactions on Cloud Computing, 2017.

2. Sindia S, Gao S, Black B, Lim A, Agrawal V, Agrawal P. Waco, TX, USA: Baylor University: MobSched: Customizable Scheduler for Mobile Cloud Computing 45th Southeastern Symposium on System Theory, 2013. 2013 March; p. 129-34.

3. George J, Chen C, Stoleru R, Xiey G, Sookoorz T, Bruno D. Hadoop MapReduce for Tactical Clouds 2014 IEEE 3rd International Conference on Cloud Networking. 2014; p. 340-46.

4. Marinelli EE. Carnegie Mellon University: Hyrax: Cloud Computing on Mobile Devices using MapReduce, Master Thesis. 2009

5. Elespuru et al., ―MapReduce System over Heterogeneous Mobile Devices,‖ Software Technologies for Embedded and Ubiquitous Systems, 2009.

6. E. E. Marinelli, ―Hyrax: Cloud Computing on Mobile Devices using MapReduce,‖ Master Thesis, Carnegie Mellon University, 2009.

[7] Sanjay Ghemawat, Howard Gobio_, and Shun-Tak Leung. The google _le sys-
tem. ACM SIGOPS Operating Systems Review, 37(5):29{43, 2003.

[8] Jean-Pierre Hubaux, Levente Butty_an, and Srdan Capkun. The quest for secu-
rity in mobile ad hoc networks. In Proceedings of the 2nd ACM International
Symposium on Mobile Ad Hoc Networking and Computing, pages 146{155, 2001.

[9] Scott Huchton, Geo_rey Xie, and Robert Beverly. Building and evaluating a k-resilient mobile distributed _le system resistant to device compromise. In
Military Communications Conference, pages 1315{1320. IEEE, 2011.

[10] Gonzalo Huerta-Canepa and Dongman Lee. A virtual cloud computing provider
for mobile devices. In Proceedings of the 1st ACM Workshop on Mobile Cloud
Computing and Services: Social Networks and Beyond, page 6, 2010.