International Journal of Advanced Trends in Computer Applications
*www.ijatca.com*

# Big Data Analytic and Efficient Data Storage System on Cloud Computing

**[1]Cho Cho Khaing, [2]Zar Zar Hnin, [3]Ei Ei Mon**
[1,2,3] Computer University, Myanmar
*[1]chokhaing28@gmail.com, [2]zarzarhnin@gmail.com, [3]eieimon80@gmail.com*

**Abstract:** *The widespread popularity of Cloud computing as a preferred platform for the deployment of web applications have resulted in an enormous number of applications moving to the cloud, and the huge success of cloud service providers. The data center storage management plays a vital role in cloud computing environments. Especially the PC cluster-based data storage is necessary to manage data on low cost storage servers in which storage space can be reduced. The paper presents the "Map Reduce" and "Hadoop" as Big Data systems that support the processing of large sets of data in a cloud computing environment. This system presents an efficient data storage approach to push work out to many nodes in a cluster using Hadoop File System (HDFS) with variable chunk size to facilitate massive data processing and introduces the implementation enhancement on MapReduce model with BW Transform to reduce the amount of data redundancy and improves the scalability to keep on working with the amount of existing physical storage capacity when the number of users and files are increased.*

**Keywords:** Big Data, Cloud Computing, Hadoop, Map Reduce, BW Transform.

## I. INTRODUCTION

Information Technology (IT) organizations worldwide are dealing with the tremendous growth of data. With the growth of capacity comes the complexity of managing the storage for that data. The technology has been developed and used in all aspects of life, increasing the demand for storing and processing more data. As a result, several systems have been developed including cloud computing that support big data. While big data is responsible for data storage and processing, the cloud provides a reliable, accessible, and scalable environment for big data systems to function [1]. Big data is defined as the quantity of digital data produced from different sources of technology for example, sensors, digitizers, scanners, numerical modeling, mobile phones, Internet, videos, e-mails and social networks. The data types include texts, geometries, images, videos, sounds and combinations of each. Such data can be directly or indirectly related to geospatial information [2]. Cloud computing refers to on-demand computer resources and systems available across the network that can provide a number of integrated computing services without local resources to facilitate user access. These resources include data storage capacity, backup and self-synchronization [3]. Most IT Infrastructure computing consists of services that are provided and delivered through public centers and servers based on them. Here, clouds appear as individual access points for the computing needs of the consumer. It is generally expected for commercial offers to meet the QoS requirements of customers or consumers, and typically include service level agreements (SLAs) [4]. They are an online storage model where data are stored on multiple virtual servers, rather than being hosted on a specific server, and are usually provided by a third party. The hosting companies, which have advanced data centers, rent spaces that are stored in a cloud to their customers in line with their needs [5].

The relationship between big data and the cloud computing is based on integration in that the cloud represents the storehouse and the big data represents the product that will be stored in the storehouse, since it is not possible to create storehouses without storing any product in them. The traditional databases known as 'relational' are no longer sufficient to process multiple-source data.

This paper presents the efficient storage system on cloud computing. This approach has been designed to use over virtualized storage system. The idea of the proposed system is to exploit the use of virtualization technology to avoid unnecessary storage purchases and reduce storage space for addressing large volumes of

data handling problem.

## II. LITERATURE REVIEWS

Big Data and Cloud computing are a major trend that are rapidly growing and new challenges and solutions are being published every day. In 2014, [6] was published on the integration of big data and cloud computing technologies. The integration of big data and cloud computing has long term benefits of both insights and performance. Due to the large amounts of data collected, they need to analyzed otherwise the data is useless; hence the cloud services can handle these extensive amounts of data with rapid response times and real time processing of the data. There are currently a few integrated cloud environments for big data analytics; Canpaas is an environment that was developed by Vrije Universiteit Amsterdam, the University of Rennes 1, Zuse-Institut Berlin and XLAB to simplify the process of creating scalable cloud applications without the need to put into consideration the complexity of these applications. The environment also provides a handful of resources for hosting web applications written in PHP or java, as well as, managing different database management systems including SQL or NoSQL. Another environment includes MapReduce provided by apache to aid in parallel computing by enabling distributed file storage systems and processing power. Task Framing is another environment used mainly for batch processing, which allows the execution of large number of non-related tasks simultaneously. The benefits of cloud computing include parallel computing, scalability, elasticity, and being inexpensive. In terms of security and privacy, data is encrypted using advanced encryption techniques to secure data but this causes high processing overheads hence other solutions are available. In regards to the performance, some challenges do arise which include data transfer limitations, data retention, isolation management, and disaster recovery. In 2015, [7] shared a paper on the efficiency of big data and cloud computing and why both technologies complement each other. Big data and cloud computing complement each other and are both the fastest growing technologies emerging today. The cloud seems to provide large computing power by aggregating resources together and offering a single system view to manage these resources and applications, so why big data should be placed on the cloud? The following reasons provide a sufficient answer to the question in hand: cost reduction, where organizations can make use of the pay per use model instead of doing a major investment to setup servers and clusters to manage the big data that can become obsolete and require upgrades; reduce overheads, by acquiring any new components needed automatically; rapid provisioning/time to market, where organizations can easily change the scale of the environments easily based on the processing requirements; flexibility/scalability, environments can be set up at any time at any scale in just a few minutes. There has been some recent work on bringing together ideas from MapReduce and HDFS system; however, this work focuses mainly on language and interface issues. A.Verma, N.Zea, B.Cho, I.Gupta, and R.H.Campbell [9] showed that their approach can achieve better performance times than a traditional MapReduce framework. Their experiments with Hadoop demonstrated speedups of up to 87% for well-suited applications, and an average of 25% for more typical applications. HadoopDB [8] built a hybrid system that takes the best features from the parallel DBMS and Hadoop approaches the prototype they built approaches parallel databases in performance and efficiency, scalability, fault tolerance, and flexibility of MapReduce-based systems. This paper [12] presented a multi-GPU parallel volume rendering implementation built using the MapReduce programming model and gave implementation details of the library, including specific optimizations made for our rendering and compositing design. They showed that our system scales with respect to the size of the volume, and (given enough work) the number of GPUs. This paper [14] gave an overview of MapReduce programming model and its applications. It described the workflow of MapReduce process. Some important issues, like fault tolerance, are studied in more detail. It also took a look at the different implementations of MapReduce. In this paper [11], they presented EUCALYPTUS, an open source software framework for cloud computing that implements what is commonly referred to as Infrastructure as a Service (IaaS); systems that give users the ability to run and control entire virtual machine instances deployed across a variety physical resources. The scientific workflow framework [10] that supported streams as first-class data, and is optimized for performance and reliable execution across desktop and Cloud platforms. This paper presented the work flow framework features and its empirical evaluation on the Eucalyptus cloud. In this paper, they showed the need for streaming support in scientific workflows to support the next generation of scientific and engineering applications that respond to events in the environment at real time. They proposed a data model for streams that can coexist with collections and files that are currently supported by workflows.

## III. SYSTEM ARCHITECTURE

In my proposed system, there are four main functions: chunking block, splitting block using HDFS, MapReduce process, big data analytics and data compression. MapReduce includes map function, combine function, partition function and reduce function.

### 3.1 Chunking Blocks

In the chunking blocks, we will divide fixed sized blocks for text files and variable sized blocks for databases.

### 3.2 HDFS

HDFS is a filesystem designed for storing very large files with streaming data access patterns, running on clusters on commodity hardware. There are Hadoop clusters running today that store petabytes of data. Streaming data access HDFS is built around the idea that the most efficient data processing pattern is a write once, read-many-times pattern. A dataset is typically generated or copied from source, and then various analyses are performed on that dataset over time. Each analysis will involve a large proportion, if not all, of the dataset, so the time to read the whole dataset is more important than the latency in reading the first record. Hadoop doesn't require expensive, highly reliable hardware to run on. It's designed to run on clusters of commodity hardware for which the chance of node failure across the cluster is high, at least for large clusters. HDFS is designed to carry on working without a noticeable interruption to the user in the face of such failure. Applications that require low-latency access to data, in the tens of milliseconds range, will not work well with HDFS. HDFS is optimized for delivering a high throughput of data, and this may be at the expense of latency. HBase is currently a better choice for low-latency access. Since the namenode holds filesystem metadata in memory, the limit to the number of files in a filesystem is governed by the amount of memory on the namenode. As a rule of thumb, each file, directory, and block takes about 150 bytes. Files in HDFS may be written to by a single writer. Writes are always made at the end of the file. There is no support for multiple writers, or for modifications at arbitrary offsets in the file.

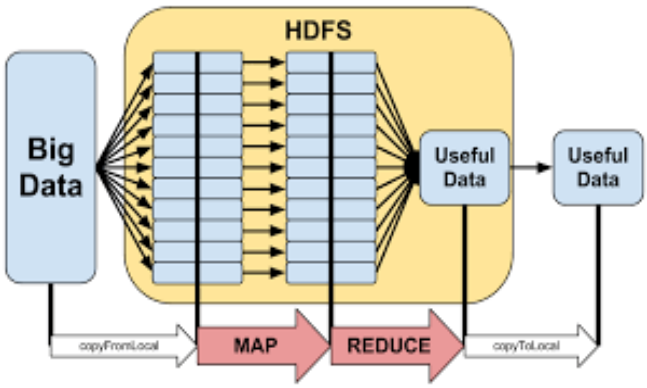


**Figure 1**: HDFS Architecture

### 3.3 MapReduce

MapReduce is a programming model for data processing. The model is simple, yet not too simple to express useful programs in. Hadoop can run MapReduce programs written in various languages: Java, Ruby, Python, and C++. Most importantly, MapReduce programs are inherently parallel, thus putting very large-scale data analysis into the hands of anyone with enough machines at their disposal. One of the most significant advantages of MapReduce is that it provides an abstraction that hides many system-level details from the programmer. Therefore, a developer can focus on what computations need to be performed, as opposed to how those computations are actually carried out or how to get the data to the processes that depend on them. Like OpenMP and MPI, MapReduce provides a means to distribute computation without burdening the programmer with the details of distributed computing (but at a different level of granularity). However, organizing and coordinating large amounts of computation is only part of the challenge. Large data processing by definition requires bringing data and code together for computation to occur no small feat for datasets that are terabytes and perhaps petabytes in size. MapReduce addresses this challenge by providing a simple abstraction for the developer, transparently handling most of the details behind the scenes in a scalable, robust, and efficient manner. Instead of moving large amounts of data around, it is far more efficient, if possible, to move the code to the data. This is operationally realized by spreading data across the local disks of machines in a cluster and running processes on machines that hold the data. The complex task of managing storage in such a processing environment is handled by the distributed file system that underlies MapReduce.

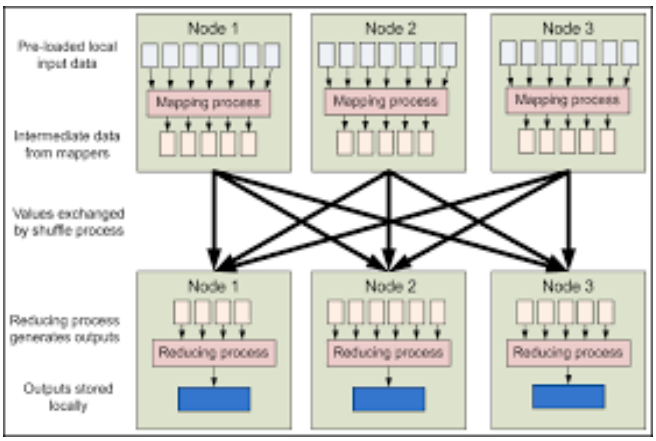


**Figure 2:** MapReduce Framework

### 3.4 Big Data

Big data is a term that refers to data sets or combinations of data sets whose size (volume), complexity (variability), and rate of growth (velocity)

make them difficult to be captured, managed, processed or analyzed by conventional technologies and tools, such as relational databases and desktop statistics or visualization packages, within the time necessary to make them useful. While the size used to determine whether a particular data set is considered big data is not firmly defined and continues to change over time, most analysts and practitioners currently refer to data sets from 30-50 terabytes(10 12 or 1000 gigabytes per terabyte) to multiple petabytes (1015 or 1000 terabytes per petabyte) as big data. Figure No. 1 gives Layered Architecture of Big Data System. It can be decomposed into three layers, including Infrastructure Layer, Computing Layer, and Application Layer from top to bottom. The characteristics of big data are as follows:

1. **Volume:** Volume refers to amount of data volume represent the size of the data how the data is large. The size of the data is represented in terabytes and petabytes.

2. **Variety:** Variety makes the data too big. The files comes in various formats and of any type, it may be structured or unstructured such as text, audio, videos, log files and more.

3. **Velocity:** Velocity refers to the speed of data processing. The data comes at high speed. Sometimes 1 minute is too late so big data is time sensitive.

4. **Value:** The potential value of Big data is huge. Value is main source for big data because it is important for businesses, IT infrastructure system to store large amount of values in database.

5. **Veracity:** Veracity refers to noise, biases and abnormality when we dealing with high volume, velocity and variety of data, the all of data are not going 100% correct, there will be dirty data.

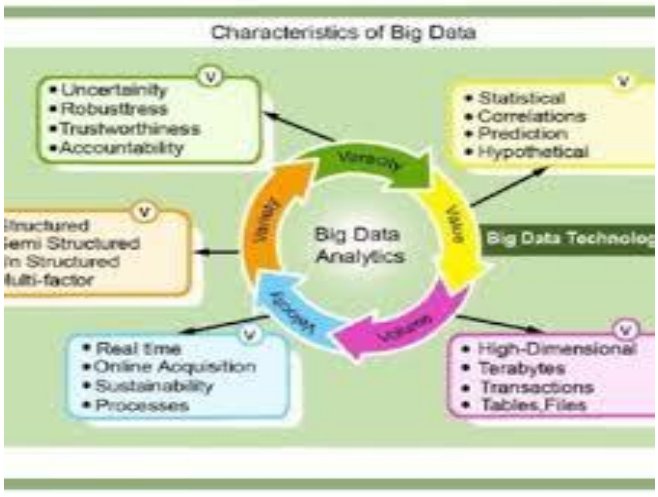


**Figure 3:** Characteristics of Big Data

### 3.5 Burrows-Wheeler Transform

The Burrows-Wheeler transform is applied on blocks of input data (symbols). It is usually the case that larger blocks result in greater compressibility of the transformed data at the expense of time and system resources. One of the effects of BWT is to produce blocks of data with more and longer runs (strings of identical symbols) than those found in the original data. The increasing the number of runs and their lengths tends to improve the compressibility of data. By additionally applying Move-to-Front coding, the data will be in a format which is generally more compressible by even zero order statistical encoders such as traditional implementations of Huffman coding.

## IV. CONCLUSION

When the amount of data is growing in existing storage capacity, the proposed system will improve the storage utilization and efficiency. By using MapReduce model with BW Transform, we will reduce the amount of data redundancy and will expect to save storage space and increase speed. The proposed system will reduce the total amount of data required to be moved over the network and get significant performance benefits as well as high throughput. This proposed system will provide the scalability to keep on working with the amount of existing physical storage capacity when the number of users and files increase. Moreover, this system will enable the service provider to add a new data centers as needed, while existing data centers continue functioning without interruption.

## References

[1] Neves, Pedro Caldeira, Bradley Schmerl, Jorge Bernardino, and Javier Cámara. "Big Data in Cloud Computing: features and issues."
[2] Lopez, Xavier. "Big data and advanced spatial analytics." In Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications, p. 5. ACM, 2012.
[3] Kshetri, Nir. "Cloud computing in developing economies." Computer 43, no. 10 (2010): 47-55.
[4] https://en.wikipedia.org/wiki/Cloud_computing
[5] Klous, Sander, and Nart Wielaard. We are Big Data: The Future of the Information Society. Springer, 2016.
[6] Chandrashekar, R., Kala, M., & Mane, D. (2015). Integration of Big Data in Cloud computing environments for enhanced data processing capabilities. International Journal of Engineering Research and General Science, 240-245.
[7] James Kobielus, I., & Bob Marcus, E. S. (2014). Deploying Big Data Analytics Applications to the Cloud: Roadmap for Success. Cloud Standards Customer Council.
[8] A.Abouzeid1,K.B.Pawlikowski,D.Abadi,A.Silbersc hatz, A.Rasin:“ HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical

Workloads", VLDB '09, August 24-28, 2009, Lyon, France Copyright 2009 VLDB Endowment, ACM .

[9] A.Verma, N.Zea, B.Cho, I.Gupta, and R.H.Campbell: " Breaking the MapReduce Stage Barrier", University of Illinois at Urbana Champaign, 2009.

[10] D.Zinn, Q.Hart, T.McPhillips, B.L.ascher, Y. Simmha n, M.Giakkoupis, V.K.Prasanna: "Towards Reliable, Performant Workflows for Streaming-Applications on Cloud Platforms", University of California,2010.

[11] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov:" The Eucalyptus Open-source Cloud-computing System", Computer Science Department, University of California, Santa Barbara, 2008.

[12] J.A. Stuart, C.K. Chen, K.L. Ma: "Multi-GPU Volume Rendering using MapReduce", MAPREDUCE 2010 Chicago, Illinois USA Copyright 2010 ACM.

[13] S. Scully and W.Benjamin: "Improving Storage Efficiencies with Data Deduplication and Compression" May 2010.

[14] T. Aarnio: "Parallel data processing with MapReduce", Helsinki University of Technology, TKK T-110.5190 Seminar on Internetworking, 2009.

## Author Profile

**Cho Cho Khaing** received post-graduate degree in computer science. She currently serves at Faculty of Computer Science, Computer University (Myanmar). Her interested research areas are Cloud Computing, Cluster-based Storage System, Big Data analytics and Data Compression.