A Survey on Data Placement in Heterogeneous Cloud Environment for Big Data

[1]Shah Dhairya Vipulkumar, [2]Hinal Somani

[1]PG Scholar, 2Assistant Professor
1Dept. of Computer Engineering,
[1]LJ Institute of Engineering & Technology, Ahmedabad, India

---

***Abstract*—Big-Data is a term for data sets that are so large or complex that traditional data processing tools are inadequate to process or manage them. Apache Hadoop is an open-source software framework for distibuted storage and distributed processing of very large data sets on computer clusters built from commodity hardware. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System(HDFS), and a processing part called MapReduce. The default Hadoop data placement strategy is suitable for homogeneous environment. But it doesn't cater well to the need for heterogeneous data intensive computing in cloud clusters. There are various data placement schemes available for heterogeneous Hadoop clusters, but each of them have their own advantages and disadvantages. This survey paper studies some of the novel data placement strategies which can be applied in heterogeneous cloud environment to speed up big data processing by enhancing the response times.**

***Index Terms*—Big Data, Cloud Computing, Data Placement, Data-Locality, Hadoop, MapReduce.**

I. INTRODUCTION

Cloud Computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort.

Nowadays, the term Big data has become very popular in Information Technology sector. Big-Data is a term for data sets that are so large or complex that traditional data processing tools are inadequate to process or manage them. The challenges include capture, curation, storage, search, sharing, transfer, analysis, and visualization. Big data refers to wide range of datasets which are difficult to be managed by existing conventional applications. Big data can be found in finance and business, banking, online and onsite purchasing, healthcare, astronomy, oceanography, engineering, and many other fields. These datasets are very complex and are growing exponentially day by day in very large amount. Data comes from social media sites, sensors, digital photos, business transactions etc. As data is increasing in volume, in variety and with high velocity, it leads to complexities in processing it. To correlate, link, match, and transform such Big data is a complex process.

Big data being a developing field has many research issues and challenges to address. The main research issues in big data are following: 1) Handling data volume, 2) Analysis of big data, 3) Privacy of data, 4) Storage of huge amount of data, 5) Data visualization, 6) Job scheduling in big data, 7) Fault tolerance. 1) Handling data volume: The large amount of data coming from different fields of science such as biology, astronomy, meteorology, etc make its processing very difficult computing to scientists. 2) Analysis of big data: it is difficult to analyze big data due to heterogeneity and incompleteness of data. Collected data can be in different formats, variety, and structure. 3) Privacy of data in the context of big data: There is public fear regarding the inappropriate use of personal data, particularly through linking of data from multiple sources. Managing privacy is both a technical and a sociological problem. 4) Storage of huge amount of data: it represents the problem of how to recognize and store important information, extracted from unstructured data, efficiently. 5) Data visualization: Data processing techniques should be efficient enough to enable real time visualization. 6) Job scheduling in big data: This issue focuses on efficient scheduling of jobs in a distributed environment. 7) Fault tolerance: is another issue in Hadoop framework in big data. In Hadoop, NameNode is a single point of failure. Replication of block is one of the fault tolerance technique used by Hadoop. Fault tolerance techniques must be efficient enough to handle failure in distributed environment. MapReduce provides an ideal framework for processing of such large datasets by using parallel and distributed programming approaches.

Apache Hadoop is an open-source software framework for distibuted storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Benefits of MapReduce:

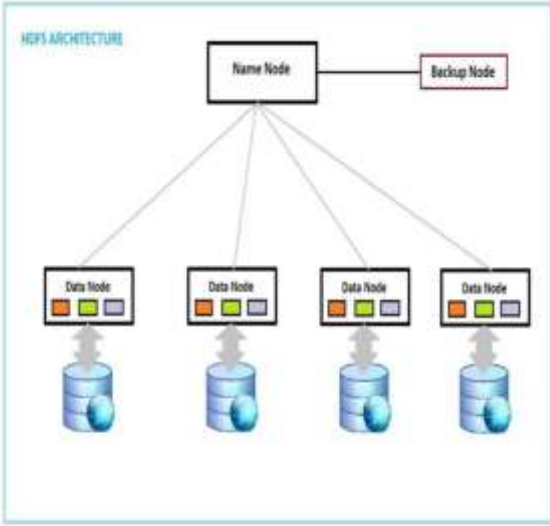

Figure 1. HDFS [1]

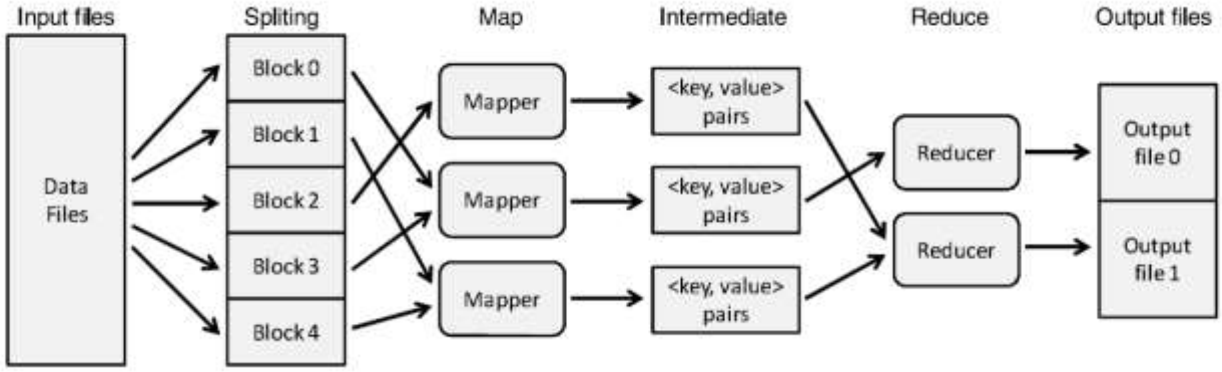

Figure 2. The Overview of MapReduce Model [2]

1)Scalability: even when new machines are added, the system works well without reconstruction or much modification. 2)Fault-Tolerance: can automatically manage failures and mitigate complexity of fault-tolerance mechanisms. 3)Simplicity: Programmers can use the MapReduce model without need to understand thoroughly the details of parallel distributed computing.

Hadoop splits files into large blocks and distributes them across nodes in a cluster. To process data, Hadoop transfers for nodes to process packaged code in parallel based on the data that needs to be processed. A program run using MapReduce model partitons jobs into numerous tasks to be assigned and run on multiple nodes in the cluster, and the program collects the processing results of each node to be returned. This approach takes advantage of data locality. The default Hadoop data placement strategy is suitable for homogeneous environment. But it doesn't cater well to the need for heterogeneous data intensive computing in cloud clusters. There are various novel data placement schemes available for heterogeneous Hadoop clusters, but each of them have their own advantages and disadvantages.

II. Related Work

[1] Suhas V. Ambade, Prof. Priya R. Deshpande have proposed "Heterogeneity-based files placement in Hadoop cluster"

Hadoop which is an open source software framework uses MapReduce framework to process data. But its current implementation there is no direct support for file formats as well as it does not take heterogeneity amongst the cluster into consideration. With default block placement, it distributes data randomly. In heterogeneous environment, it's hard to utilize the power of each node correctly and this may cause performance degradation. In this paper, the authors have proposed a format specific data distribution amongst the nodes by using computational power of each node. By experimentation, results shows there is improvement in the Map performance, MapReduce performance and overall performance of Hadoop in heterogeneous cluster.

The authors have modified the block placement policy that supports for heterogeneous environment. One of the motives in this modification was to improve the performance for processing the unstructured data like pdf, text, etc. formats. As pdf files are more complex, so those are placed on high computational power node, and text are treated as less complex and hence placed on low computational power cluster. Authors call this policy as customized block placement policy. This helped them achieve higher performance. In the proposed solution, at the time of writing data, the DataNodes are selected based on the complexity of data and computation power of each node. Complexity here is dependent on the file type. The Max CPU MHz is considered as the computational power of node for placing the file.

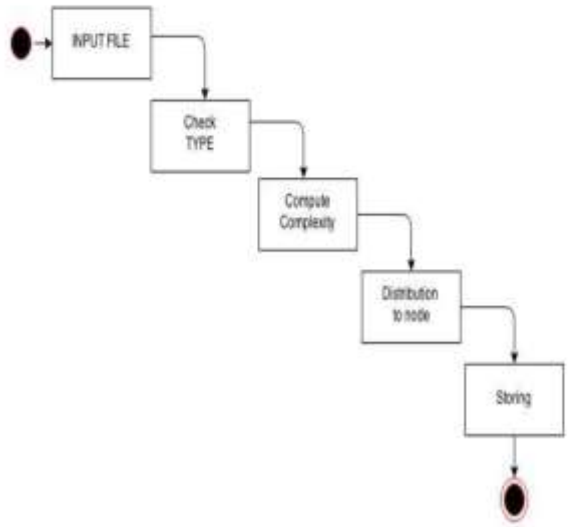

Figure 3. State transition diagram [1]

In one configuration file, they store the information regarding which DataNode should be selected based on specific file type, eg. PDF. Then the type of input file is compared with the configuration file. According to the type or complexity of input file, the DataNode is selected to place the file.

The proposed data block placement strategy based on the computing capacities and the file formats of data for allocating data blocks to DataNode improves the performance of Hadoop. Compared with Hadoop-1.2.1; on which the changes were made, the WordCount and Grep shows improvement by 21.7% and 20.1% respectively. While the performance of Hadoop scales up by 16.2%. As a future work, the authors seek to improve the perfromance of Hadoop further.

[2] Chia-Wei Lee, Kuang-Yu Hsieh, Sun-YuanK Hsieh, Hung-Chang Hsiao have proposed "A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments"

According to the authors, the current implementation of the hadoop assumes that every node in a cluster has the same computing capacity and that the tasks are data-local, which may increase extra overhead and reduce the MapReduce performance.

The authors have proposed a data placement algorithm to resolve the unbalanced node workload problem. The proposed method can dynamically adapt and balance data stored in each node based on the computing capacity of each node in a heterogeneous Hadoop cluster. The proposed method can reduce data transfer time to achieve improved Hadoop performance. The experimental results show that dynamic data placement policy can decrease the time of execution and improve Hadoop performance in a heterogeneous cluster.

The proposed dynamic data placement (DDP) algorithm is based on different computing capacities of nodes to allocate data blocks, thereby improving data locality and reducing the additional overhead to enhance Hadoop performance. From the experiments, regarding WordCount, DDP can improve by up to 24.7% with an average improvement of 14.5%; while regarding Grep, DDP can improve by up to 32.1% with an average improvement of 23.5%.

[3] Vrushali Ubarhande, Alina-Madalina Popescu, Horacio Gonzalez-Velez have proposed "Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments"

The main contribution of this work is a novel methodology for data placement for Hadoop DataNodes based on their computing ratio. Two standard MapReduce applications, WordCount and Grep, have been executed and a significant performance improvement has been observed based on our proposed data distribution technique.

The proposed solution ensures the distribution of data blocks within the Hadoop cluster based on the computing capability of each slave node. First, a *Speed Analyser component is created on NameNode*, and installed an executed on each slave node. Furthermore, the master node reads the response time taken by each slave node from the respective log file, and then creates a file with the computing ratio. This file is fed to the *Data Distribution* component through the Hadoop Distributed File System. Speed Analyser agent is created to measure the processing capacity of each slave node.

The paper has proposed a solution to improve the performance of heterogeneous Hadoop cluster by assigning data blocks on the basis of the processing speed of the DataNodes. The assignment is done based on the computing ratio of the nodes. It has an advantage of understanding the DataNode processing speed and the Data Distribution Technique which assigns blocks as per the computing capacity of DataNodes.

[4] Jia-xuan WU, Chang-sheng ZHANG, Bin ZHANG, Peng WANG have proposed "A New Data-Grouping-Aware Dynamic Data Placement Method that Take into Account Jobs Execute Frequency for Hadoop"

The authors observe that many data-grouping-aware data placement schemes are static, without taking MapReduce job execution frequency into consideration. Such data placements scheme will lead to severe performance degradation that is way below the potential efficiency of optimal data distribution when executing MapReduce jobs that are executed frequently.

The authors propose a new data-grouping-aware dynamic (DGAD) data placement method based on the job execution frequency. Firstly, a job access correlation relation model is built among the data blocks according to the relationships provided by the records about historical data block access. Then clustering algorithm is used to divide data blocks into clusters according to the job access correlation relation model among the data blocks and propose a data placement algorithm based on data block clusters in order to put correlated data blocks within a cluster on the different nodes. Finally, a series of experiments are carried out in order to verify the method proposed in this paper.

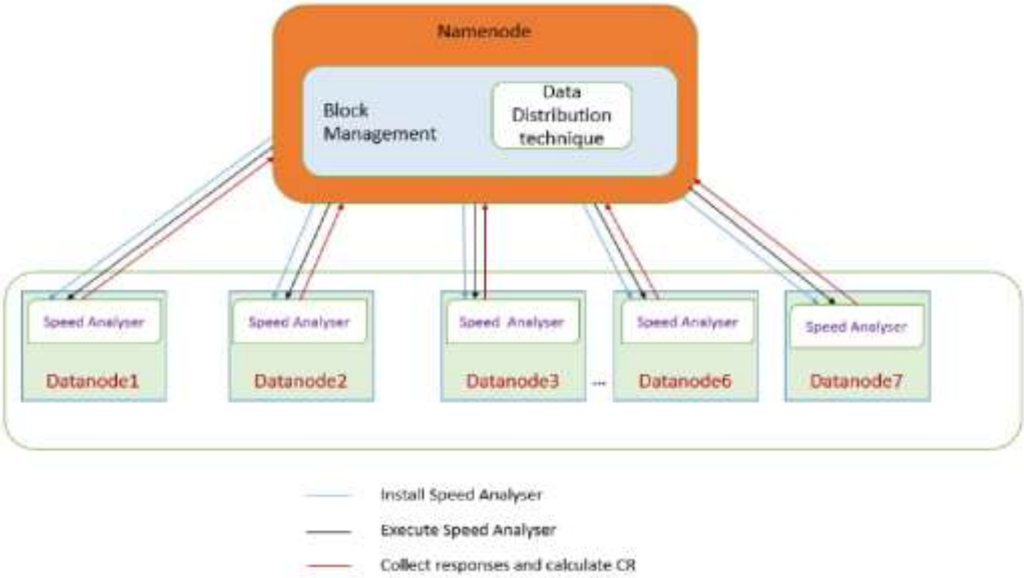

Figure 4. Proposed Data-Distribution Architecture [3]

There are three phases in DGAD: building a job access frequency correlation model from system logs, clustering the data grouping matrix based on access frequency correlation degree, and re-organizing the clustered data. Experimental results show that, for two representative MapReduce jobs—MaxTo1949 and MaxTo1988—DGAD can significantly improve the execution efficiency of MapReduce.

[5] S. Sangavi, A. Vanmathi, R. Gayathri, R. Raju, P. Victer Paul, P. Dhavachelvan have proposed "An Enhanced DACHE model for the MapReduce Environment"

The unstructured data are processed using MapReduce framework and Hadoop as an environment, providing distributed file storage for absolute blocks of data. The intermediate data generated are left unstored and hindered in MapReduce framework.

In this paper, authors propose an effective technique for enhancing the performance of Dache by introducing a cache optimization algorithm and a better page replacement technique. Dache refers a cache imbibing the data that is accessed through a request and reply protocol.

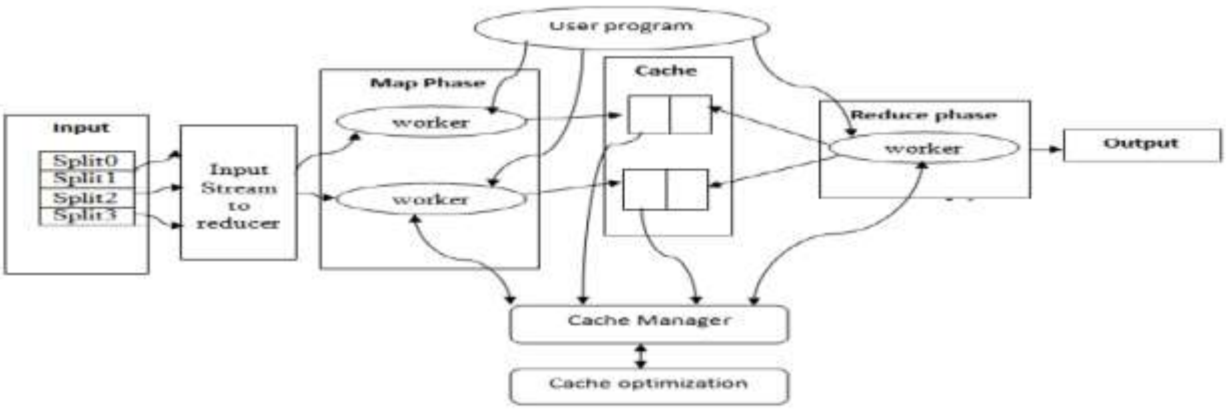

Figure 5. Architecture of Enhanced Dache [5]

In the proposed technique, the dache request the source input from which the cache item is obtained and various operations are applied over the input such that the cache item is indexed properly in the map phase and the intermediate results are generated. In the reduce phase, the generated intermediate results are shuffled and sorted. Then it is fed to the worker nodes in the reduce phase. The final results are computed and given to the user. The dache is enhanced by providing a better page replacement and a cache optimization technique. The page replacement technique used here is WSClock algorithm. It effectively replaces the pages by embedding a reference bit to each page.

The enhancement of Dache is proposed by a better cache optimization algorithm and a page replacement technique which is represented as three modules- map phase, reduce phase and cache manager which greatly improves the performance of the MapReduce framework, making the searching process more easy and efficient than the existing technique and eliminating the duplicate task that co-exists with the incremental MapReduce jobs.

[6] B Ganesh Babu, Shabeera T P, Madhu Kumar S D have proposed "Dynamic Colocation Algorithm For Hadoop"
The current data placement policy of Hadoop distributes the data among DataNodes using random placement policy for simplicity and load balance. This simple data placement is good for Hadoop applications that used to access data from a single file. But if any application needs data from different files simultaneously, the performance normally degrades. Identifying the related files and placing them in the same DataNode or in adjacent DataNodes reduces network overhead and reduces the query span. Authors propose a Dynamic Colocation Algorithm, where the average number of machines that are involved in processing a query decreases by colocating the datasets, that are frequently accessed together and hence reduces the network overhead. Our technique checks the relations between datasets dynamically and rearrange the datasets according to their relations. Experimental results show that, after colocation there is a significant reduction on the execution time of MapReduce programs.
In this paper, authords present a novel Dynamic Colocation Algorithm, for colocating related files in HDFS. In Hadoop, default data placement policy does not consider any data relatioships while placing the datasets into HDFS. By making use of the data characteristics like interdependency, they propose an efficient algorithm for dynamically colocating the related datasets. We studied the performance of our dynamic colocation algorithm and compared with the existing partition algorithm i.e BEA algorithm. The simulation results show that Dynamic colocation algorithm colocates the related files efficiently.
Authors considered homogeneous environment only. They plan to extend this to heterogeneous clusters. Their algorithm chooses DataNodes having more free space to place the datasets after partitioning. For better load balancing one can choose the DataNodes optimally.

III. COMPARATIVE ANALYSIS

Sr. No.	Paper Title	Methods/ Techniques	Advantages	Limitations
1.	Heterogeneity-based files placement in Hadoop cluster	Customized format specific Placement Policy/ Modified Hadoop	Improved MapReduce and Hadoop performance.	Format specific policy might not always make correct decisions.
2.	A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments	Dynamic Data Placement Strategy, Computing Capacity based Placement	Can dynamically adapt and balance data stored in each node based on the computing capacity of each node, reduce execution time, improve Hadoop performance.	Other parameters like memory, power consumption, etc. are not considered.
3.	Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments	Speed Analyzer, Processing Capability based Distribution	Understands processing capability and assigns data block as per processing speed .	Considering only processing speed is not enough. Other factors need to be considered.
4.	A New Data-Grouping-Aware Dynamic Data Placement Method that Take into Account Jobs Execute Frequency for Hadoop	Data-Grouping Aware Dynamic Placement, Frequency Aware, Clustering based Data Placement	Improves Hadoop performance considerably.	Data block which are not being accessed frequently but is important might be overlooked.
5.	An Enhanced DACHE model for the MapReduce Environment	Enhanced Dache, Page Replacement Mechanism – WSClock algorithm	Better caching and page replacement mechanism.	Caching is useful for small files only. Large data blocks need a more efficient data placement technique.

6.	Dynamic Colocation Algorithm For Hadoop	Dynamic Colocation Algorithm	Data relationships and inter-dependency considered, co-locates related files efficiently	Heterogeneity not considered, optimization can be further done.

Table 1. Literature Comparision

IV. Conclusion & Future Work

The default Hadoop data placement strategy is suitable for homogeneous environment. But it doesn't cater well to the need for heterogeneous data intensive computing in cloud clusters. Hadoop which is an open source software framework uses MapReduce framework to process data. But its current implementation there is no direct support for file formats as well as it does not take heterogeneity amongst the cluster into consideration. With default block placement, it distributes data randomly. There are various novel data placement schemes available for heterogeneous Hadoop clusters, but each of them have their own advantages and disadvantages. In this survey, the papers studied suggest various novel algorithms and/or methods to improve data placement in heterogeneous cloud environment and thereby helping in speeding up the big data processing.

V. References

[1] Suhas V. Ambade, Prof. Priya R. Deshpande, "Heterogeneity-based files placement in Hadoop cluster", 2015 IEEE, Pages: 876 - 880, DOI: 10.1109/CICN.2015.325

[2] C.-W. Lee et al., A Dynamic Data Placement Strategy for Hadoop in Heterogroneous Environments, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.002

[3] Vrushali Ubarhande, Alina-Madalina Popescu, Horacio Gonzalez-Velez, "Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments", 2015 IEEE, Pages: 217 - 224 DOI 10.1109/CISIS.2015.37

[4] Jia-xuan Wu , Chang-sheng Zhang , Bin Zhang , Peng Wang , A New Data-Grouping-Aware Dynamic Data Placement Method that Take into Account Jobs Execute Fre-quency for Hadoop, Microprocessors and Microsystems (2016), DOI: 10.1016/j.micpro.2016.07.011.

[5] Sangavi, A. Vanmathi, R. Gayathri, R. Raju, P. Victer Paul, P. Dhavachelvan, "An Enhanced DACHE model for the MapReduce Environment", Pages: 579 - 584, 2015 Elsevier B.V., DOI: 10.1016/j.procs.2015.04.087

[6] B Ganesh Babu, Shabeera T P, Madhu Kumar S D have proposed, "Dynamic Colocation Algorithm For Hadoop", 2014 IEEE, Pages: 2643 - 2647, DOI: 10.1109/ICACCI.2014.6968384

[7] https://dzone.com/articles/how-hadoop-mapreduce-works, Accessed: 23/11/2016, 4:40 PM

[8] http://www.slideshare.net/xqin74/performance-issues-on-hadoop-clusters, Accessed: 23/11/2016, 3:50 PM