RESEARCH ARTICLE OPEN ACCESS

# Modified Data Analysis of Big Data Using Map Reduce In Hadoop Process

Dr.G.Ravi, V.Sobanadevi
Associate Professor Head Department of Computer Science Jamal Mohamed College (Autonomous) Trichy-20
Research scholar Department of Computer Science Jamal Mohamed College (Autonomous) Trichy-20

**Abstract**

With the advancement of PC innovation, there is a colossal increment in the development of information. Researchers are overpowered with this expanding measure of information handling needs which is getting emerged from each science field. A major issue has been experienced in different fields for making the full utilization of these expansive scale information which bolster basic leadership. Information mining is the strategy that can finds new examples from huge informational indexes. For a long time it has been examined in a wide range of utilization territory and in this way numerous information mining strategies have been produced and connected to rehearse. However, there was a colossal increment in the measure of information, their calculation and investigations as of late. In such circumstance most established information mining strategies wound up distant by and by to deal with such enormous information. Productive parallel/simultaneous calculations and usage procedures are the way to meeting the versatility and execution prerequisites involved in such huge scale information mining investigations. Number of parallel calculations has been executed by making the utilization of various parallelization strategies which can be recorded as: strings, MPI, MapReduce, and blend or work process innovations that yields diverse execution and convenience attributes. MPI demonstrate is observed to be effective in figuring the thorough issues, particularly in reproduction. Be that as it may, it is difficult to be utilized as a part of genuine. MapReduce is created from the information investigation model of the data recovery field and is a cloud innovation. Till now, a few MapReduce structures has been produced for taking care of the enormous information. The most renowned is the Google. The other one having such highlights is Hadoop which is the most well known open source MapReduce programming embraced by numerous enormous IT organizations, for example, Yahoo, Facebook, eBay et cetera. In this paper, we center particularly around Hadoop and its execution of MapReduce for expository handling.

***Keywords***: Big Data, Data Mining, parallelization Techniques, HDFS, MapReduce, Hadoop.

## 1. Introduction

Associations with a lot of multi-organized information think that its hard to utilize customary social DBMS innovation for preparing and breaking down such information. This kind of issue is particularly looked by Web-based organizations, for example, Google, Yahoo, Facebook, and LinkedIn which require to process immense voluminous information in a quickly and in savvy way. An extensive number of such associations have built up their own particular non-social frameworks to beat this issue. Google, for instance, created MapReduce and the Google File System. It likewise assembled a DBMS framework known as BigTable. It winds up conceivable to seek a large number of pages and restore the outcomes in milliseconds or less with the assistance of the calculations that

drive both of these real association look administrations started with Google's MapReduce structure [1]. It is an exceptionally difficult issue of today to dissect the huge information. Huge information is major ordeal to work upon thus it is a challenging task to perform examination on enormous information. Innovations for investigating huge information are advancing quickly and there is noteworthy enthusiasm for new expository methodologies, for example, MapReduce, Hadoop and Hive, and MapReduce augmentations to existing social DBMSs [2]. The utilization of MapReduce structure has been generally came into center to deal with such monstrous information successfully. Throughout the previous couple of years, MapReduce has showed up as the most famous figuring worldview for parallel, bunch style and examination of expansive measure of information [3]. MapReduce picked up its notoriety when utilized effectively by Google. In genuine, it is a versatile and blame tolerant information preparing device which gives the capacity to process colossal voluminous information in parallel with some low-end figuring hubs [4]. By temperance of its straightforwardness, adaptability, and adaptation to non-critical failure, MapReduce is getting to be universal, increasing huge energy from both industry and scholastic world. We can accomplish elite by breaking the preparing into little units of work that can be keep running in parallel over a few hubs in the group [5]. In the MapReduce structure, a disseminated record framework (DFS) at first segments information in numerous machines and information is spoken to as (key, esteem) sets. The MapReduce structure executes the primary capacity on a solitary ace machine where we may preprocess the information before delineate are called or postprocess the yield of decrease capacities. A couple of guide and decrease capacities might be executed once or various circumstances as it relies upon the qualities of an application [6]. Hadoop is a famous open-source usage of MapReduce for the examination of extensive datasets. It utilizes a conveyed client level filesystem to oversee capacity assets over the group [7]. However, the framework yields undesired speedup with less huge datasets, yet creates a sensible speed with a bigger gathering of information that supplements the quantity of figuring hubs and diminishes the execution time by 30% when contrasted with ordinary information mining and other handling strategies [8]. Segment 2 gives the general showing of the development of guide, lessen and Hadoop. Segments 3 give the detail portrayal Big Data and programming model of MapReduce. Segment 4 details the Hadoop design. Segments 5 give the viable approach of MapReduce and Hadoop innovation which is a capable mix of guide and diminish work with the coming of Hadoop.

## 2. Related Work

Big Data alludes to different types of expansive data sets that require extraordinary computational stages with a specific end goal to be examined. A ton of work is required for examining the enormous information. In any case, to investigate such enormous information is an extremely difficult issue today. The MapReduce structure has as of late pulled in a great deal of consideration for such application that takes a shot at broad information. MapReduce is a programming model and a related execution for handling and creating extensive datasets that is receptive to a wide assortment of genuine undertakings [9]. The MapReduce worldview procures the element of parallel programming that gives effortlessness. In the meantime alongside these attributes, it offers stack adjusting and adaptation to non-critical failure limit [10]. The Google File System (GFS) that commonly underlies a MapReduce framework gives a productive and solid disseminated information stockpiling which is required for applications that chips away at vast databases [11]. MapReduce is enthused by the guide

and lessens natives introduce in utilitarian dialects [12]. Some at present accessible usage are: shared-memory multi-center framework [13], topsy-turvy multi-center processors, realistic processors, and bunch of organized machines [14]. The Google's MapReduce method makes conceivable to build up the extensive scale circulated applications in a more straightforward way and with lessened cost. The principle normal for MapReduce display is that it is equipped for handling vast informational indexes parallelly which are dispersed over numerous hubs [15]. The novel Map-Reduce programming is a restrictive arrangement of Google, and along these lines, not accessible for open utilize. In spite of the fact that the appropriated processing is to a great extent streamlined with the thoughts of Map and Reduce natives, the fundamental foundation is non-inconsequential so as to accomplish the coveted execution [16]. A key foundation in Google's MapReduce is the fundamental disseminated record framework to guarantee information area and accessibility [9]. Joining the MapReduce programming strategy and a productive circulated record framework, one can without much of a stretch accomplish the objective of dispersed figuring with information parallelism more than a large number of registering hubs; handling information on terabyte and petabyte scales with enhanced framework execution, advancement and unwavering quality. It was watched that the MapReduce apparatus is much proficient in information improvement and exceptionally solid since it lessens the season of information access or stacking by over half [16]. It was the Google which initially advanced the MapReduce method. [17]. The as of late presented MapReduce strategy has picked up a great deal of consideration from mainstream researchers for its relevance in substantial parallel information investigations [18]. Hadoop is an open source usage of the MapReduce programming model which depends individually Hadoop Distributed File System (HDFS). It doesn't rely upon Google File System (GFS). HDFS recreates information hinders in a dependable way, places them on various hubs and after that later calculation is performed by Hadoop on these hubs. HDFS is like different filesystems, yet is intended to be exceedingly blame tolerant. This dispersed document framework (DFS) does not require any top of the line equipment and can keep running on item PCs and programming. It is likewise versatile, which is one of the essential outline objectives for the execution. As it is discovered that HDFS is free of a particular equipment or programming stage, in this manner, it is effortlessly convenient crosswise over heterogeneous frameworks [19]. The great accomplishment made by MapReduce has empowered the development of Hadoop, which is a well known open-source usage. Hadoop is an open source structure that actualizes the MapReduce[20]. It is a parallel programming model which is made out of a MapReduce motor and a client level filesystem that oversees stockpiling assets over the bunch [9]. For transportability over an assortment of stages — Linux, FreeBSD, Mac OS/X, Solaris, and Windows — the two segments are composed in Java and just require ware equipment.

## 3. THE IMPORTANCE OF BIG DATA

Associations need to fabricate an investigative figuring stage to understand the full estimation of huge information. This empowers business clients to influence use, to structure and investigate huge information to extricate helpful business data that isn't effectively discoverable in its genuine unique plan. The hugeness of Big Data can be described as[21]:

1) Big information is an important term in spite of the buildup

2) It is increasing greater prominence and enthusiasm from both business clients and IT industry.

3) From an examination point of view regardless it speaks to systematic workloads and information administration arrangements that couldn't already be bolstered on account of cost contemplations as well as innovation constraints.

4) The arrangements gave empower more quick witted and quicker basic leadership, and enable associations to accomplish speedier time to an incentive from their interests in expository preparing innovation and items.

5) Analytics on multi-organized information empower more astute choices. Up till now, these sorts of information have been hard to process utilizing customary diagnostic preparing advancements.

6) Rapid choices are empowered in light of the fact that enormous information arrangements bolster the fast investigation of high volumes of nitty gritty information.

7) Faster time to esteem is conceivable on the grounds that associations would now be able to process and break down information that is outside of the undertaking information distribution center.

The developers utilize the programming model MapReduce to recover valuable data from such enormous information. The primary highlights and issues related in giving diverse sorts of huge informational collections are compressed in the table beneath. It gives précises data how Big Data advancements can help settle them [22].
Table 1: Summarizes the main features, challenges and technology responses connected to handing different types of large data sets

| Attribute | Features | Challenges and Skill responses |
|---|---|---|
| **Volume** | Amount of generated data has increased tremendously the past years. However, this is the less challenging aspect in practice. | Internet has created tremendous increment in the global data production. A response to this situation has been through the generalization of the cloud based solutions.<br><br>The noSQL database approach is a response to store and query huge volumes of data heavily distributed. |
| **Velocity** | Production of data is growing with high speed and such produced data must be collected in shorter time frames. | Millions of connected devices (smartphones) are getting added daily which results in the increase of not only the volume but also velocity.<br>To get a competitive edge, global companies considered the Real-time data processing platforms as a requirement. |

| | | |
|---|---|---|
| **Variety** | There came the explosion of data formats that range from structured information to free text with the multiplication of data sources. | The current way to collect and analyse non-structured or semi-structured data is just opposite from the manner the traditional relational data model and query languages does. This reality has resulted in the evolution of new kinds of data stores that gives the ability to support flexible data models. |
| **Value** | Until recently, there was more focus on recording the large volumes of data but not bothered how to conquer them. | Big Data technologies are deeping their roots in creating, capturing and exploiting large volumes of data. In principle, the challenge comes while transforming underdone data into information that contains value and can be used in decision making or other business requirements. |

### 3.1 MapReduce: A Programming Model

MapReduce is designed to be used by programmers, rather than business users. It is a programming model, not a programming language. It has gained popularity for its easiness, efficiency and ability to control "Big Data" in a timely manner. The steps involved in working of MapReduce can be shown in as:

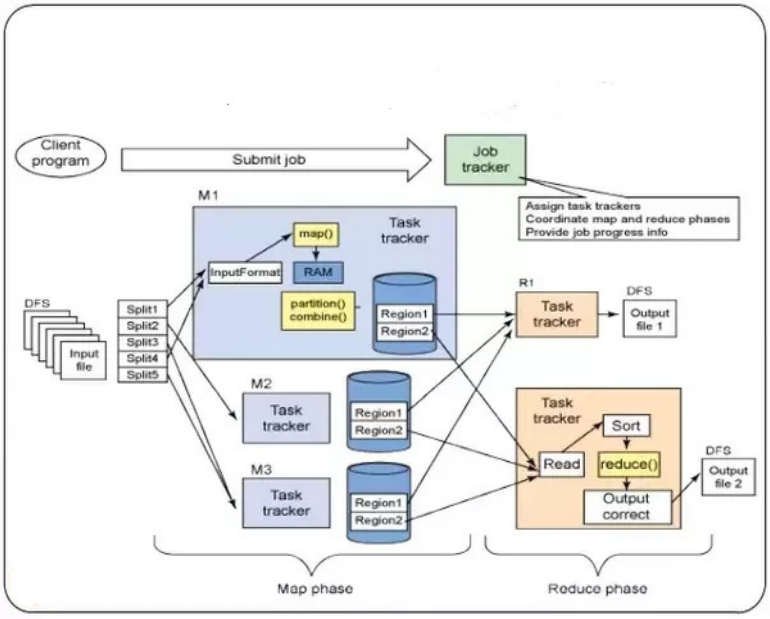


**Fig. 1: Steps in MapReduce to process the database**

The applications which include indexing and search, graph analysis, text analysis, machine learning, data transformation and many more, are not easy to implement by making the use of

standard SQL which are employed by relational DBMSs. In such areas the procedural nature of MapReduce makes it easily understood by skilled programmers. It also has the advantage that developers do not have to be concerned with implementing parallel computing – this is handled transparently by the system. Although MapReduce is designed for programmers, non-programmers can exploit the value of prebuilt MapReduce applications and function libraries [3]. The architecture of MapReduce can be depicted as:

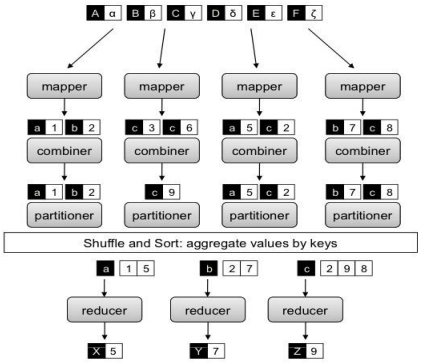


**Fig. 2: MapReduce with combiners, partitioners**

Table 2: Description of mappers, reducers, partitioners and combiners

| | |
|---|---|
| **Mappers** | Required to generate an arbitrary number of intermediate pairs. |
| **Reducers** | Applied to all intermediate values associated with the same intermediate key. |
| **Partitioners** | Its main job is to divide the intermediate key space, and then to assign the intermediate key-value pairs to reducers. |
| **Combiners** | Combiners are an (optional) optimization.  Before performing the phase of shuffle and sort, it allows the local aggregation of data. Essentially, combiners are used to save bandwidth, e.g.: word count program. |

MapReduce programs are usually written in Java. They can also be coded in other languages such as C++, Python, Ruby, R, etc. These programs may process data stored in different file and database systems. At Google, for example, MapReduce was implemented on top of the Google File System (GFS).

## 4. Problem Formulation

Hadoop: Yahoo! became the primary contributor in 2006

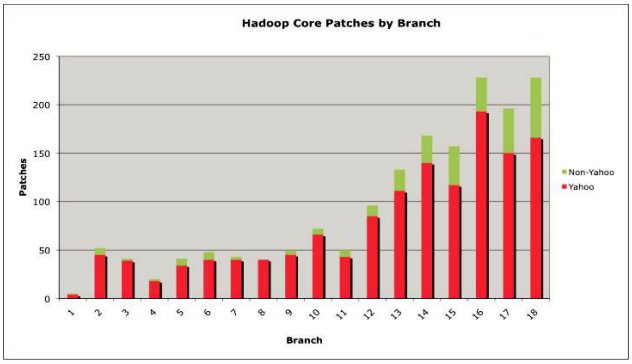


Fig. 3: Primary contribution of Hadoop

Apache Hadoop consists of several components. The ones that of are of interest from a database and analytical processing perspective are [23]:

Hadoop Distributed File System (HDFS), MapReduce, Pig, Hive, HBase, SqoopHDFS can be a source or target file system for MapReduce programs. It is best suited to a small number of very large files. Use of data replication make possible to achieve data availability in HDFS. But it results into the rise in storage required to cope the data. The HadoopMapReduce framework helps in distributing the map program processing so that the required HDFS data is local to the program. To process all of the output files created by the mapping process, the Reduce program performs more movement and access to internode data. At the time of execution, both the map and reduce programs write the accomplished data to the local file system so as to reduce or even avoid the overhead of HDFS replication. HDFS supports multiple readers and one writer (MROW). The index mechanism is not available in HDFS, hence, it is best suited to read-only applications that need to scan and read the complete contents of a file. In HDFS, the actual location of the data is transparent to applications and external software.

**HDFS architecture**

The architecture of HDFS includes the master and the slave nodes, where the master is called a *NameNode* and the slaves are called *DataNodes.* HDFS contains only a single NameNode (master) and have many DataNodes (slaves) across the cluster, usually one per node. HDFS assigns a *namespace* (similar to a package in Java) to store the users data. A file might be split into one or more data blocks, and these data blocks are kept in a set of DataNodes. The NameNode will have the necessary metadata information on how the blocks are mapped to each other and which blocks are being stored in which of the NameNodes. The request made by the client to read and write the filesystem get processed directly by the DataNodes, whereas namespace operations like the opening, closing, and renaming of

directories are performed by NameNodes. The responsibilities of NameNode for DataNodes are to issue instructions regarding certain activities, such as, data block creation, replication, and deletion [20]. HDFS (Hadoop distributed file system) architecture is shown as [23]:

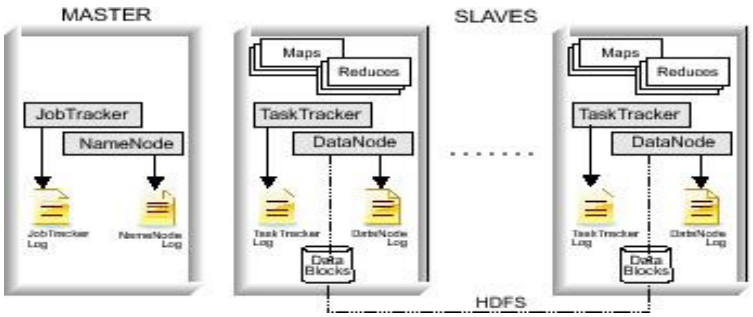


Fig. 4: A simple model of a multi-node Hadoop cluster

A typical deployment of HDFS has a dedicated machine that runs only the NameNode. Each of the other machines in the cluster typically runs one instance of the DataNode software, though the architecture does allow you to run multiple DataNodes on the same machine. The NameNode is concerned with metadata repository and control, but otherwise never handles user data. The NameNode uses a special kind of log, named *EditLog,* for the persistence of metadata.

**Deploying Hadoop**

Though Hadoop is a pure Java implementation, we can use it in two different ways. We can either take advantage of a streaming API provided with it or use Hadoop pipes. The latter option allows building Hadoop apps with C++. Here, we will focus on the former. Hadoop's main design goal is to provide storage and communication on lots of homogeneous commodity machines. The implementers selected Linux as their initial platform for development and testing; hence, if interested to work with Hadoop on Windows, it is required to install separate software to mimic the shell environment.

Hadoop can run in three different ways, depending on how the processes are distributed [24]:

- **Standalone mode**: This is the default mode provided with Hadoop. Everything is run as a single Javaprocess.
- **Pseudo-distributed mode**: Here, Hadoop is configured to run on a single machine, with different Hadoopdaemons run as different Java processes.
- **Fully distributed or cluster mode**: Here, one machine in the cluster is typically labelled as the NameNodeand another machine is designated as the JobTracker. Only one NameNode is placed in each cluster, which manages the namespace, filesystem metadata, and access control. An optional SecondaryNameNode can also be placed for periodic handshaking with NameNode for fault tolerance. The rest of the machines within the cluster act as both DataNodes and TaskTrackers. The DataNode holds the system data; each data node manages its own locally scoped storage, or its local hard disk. The TaskTrackers carry out map and reduce operations.

## 5. Our Contribution

Recently, in some experiments it has been discovered that applications using Hadoop performed poorly compared to similar programs using parallel databases. Our main objective is to optimize HDFS and provide significant impact on the overall performance of a MapReduce framework which will result in the boosting of overall efficiency ofMapReduce applications in Hadoop. There may be no change in the ultimate conclusions of the MapReduce versus parallel database

debate but this new approach of Hadoop and MapReduce will certainly allow a fairer comparison of the actual programming models. Though Hadoop provides built-in functionality to profile Map and Reduce task execution but there are no built-in tools to contour the framework itself, that can allow performance hurdles to remain unexposed. This paper has retrieved the interactions between Hadoop and storage. Here, we explained how many performance blockages are not directly attributable to application code (or the MapReduce programming style), but rather are caused by the task scheduler and distributed filesystem underlying all Hadoop applications. HDFS performance under concurrent workloads can be significantly improved through the use of application-level I/O scheduling while preserving portability. Further improvements can be done by reducing fragmentation and cache overhead which are also possible at the expense of reducing portability. The portability in Hadoop support users by making the development simper and reduce installation complexity. This results in the widespread of this parallel computing paradigm.

## 6. Conclusion

Big data and the technologies associated with it can bring significant benefits to the business. But the tremendous uses of these technologies make difficult for an organization to strongly control these vast and heterogeneous collections of data to get further analysed and investigated. There are several impacts of using the Big Data. For facing the competitions and strong growth of individual companies, it supports by providing them a huge potential. Certain aspects are needed to be followed so that we can get timely and productive results from Big Data because the precise use of Big Data can give the proliferation to throughput, modernization, and effectiveness for entire divisions and economies. To be able to extract the benefits of Big Data, it is crucial to know how to ensure intelligent use, management and re-use of Data Sources, including public government data, in and across country to build useful applications and services. It is crucial to evaluate the best approach to use for filtering and/or analyzing the data. For the optimized analytic processing, Hadoop with MapReduce can be used. In this paper, we've presented the basics of MapReduce programming with the open source Hadoop framework. This outstanding framework of Hadoop speeds-up the processing of large amounts of data through distributed processes and thus, provides the responses very fast. It can be adopted and customized to meet various development requirements and can be scaled by increasing the number of nodes available for processing. The extensibility and simplicity of the framework are the key differentiators that make it a promising tool for data processing.

## References

1.J R Swedlow, G Zanetti, C Best. Channeling the data deluge. Nature Methods, 2011, 8: 463-465

2.G C Fox, S H Bae, et al. Parallel Data Mining from Multicore to Cloudy Grids. High Performance Computing and Grids workshop, 2008

3. Maitrey S, Jha. An Integrated Approach for CURE Clustering using Map-Reduce Technique. In Proceedings of Elsevier, ISBN 978-81-910691-6-3,2nd August 2013].

4.D. DeWitt and M. Stonebraker. MapReduce: A major step backwards. The Database Column, 1, 2008.

5.Apache. Apache Hadoop. http://hadoop.apache.org, 2010.

6.Y. Kim and K. Shim. Parallel top-k similarity join algorithms using MapReduce. In ICDE, 2012.
7.Jeffrey Shafer, Scott Rixner, and Alan L. Cox. The Hadoop Distributed Filesystem: Balancing Portability and Performance. DOP is March 30,2010.
8.Moturi, Maiyo. Use of MapReduce for Data Mining and Data Optimization on a Web Portal.Published in International Journal of Computer Applications (0975 – 8887) Volume 56– No.7, October 2012].
9. Jeffrey Dean et al. Mapreduce: Simplified data processing on large clusters. In Proceedings of the 6th USENIX OSDI, pages 137–150, 2004.
10. S. Ghemawat et al. The google file system. ACM SIGOPS Operating Systems Review, 37(5):29–43, 2003.
11. C. Ranger et al. Evaluatingmapreduce for multi-core and multiprocessor systems. In Proceedings of the 2007 IEEE HPCA, pages 13–24, 2007.
12. Yoo, R. M., Romano, A.K. and Kozyrakis, C. 2009. Phoenix Rebirth: "Scalable MapReduce on a Large-Scale Shared-Memory System". Proceedings of the 2009 IEEE International Symposium on Workload Characterization, pp. 198-207.
13. Rafique, Mustafa. M. 2009. "Supporting MapReduce on Large-Scale Asymmetric Multi-Core Clusters". ACM SIGOPS Operating Systems Review, Vol. 43, 2, pp. 25-34.
14. J. Dean et al. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1):107– 113, 2008.
15. Kyong, Lee, Choi, Chung, Moon. Parallel Data Processing with MapReduce: A Survey. Published in SIGMOD Record, December 2011 (Vol. 40, No. 4).
16.B. Panda, J. Herbach, S. Basu, and R. J. Bayardo, "Planet: Massively parallel learning of tree ensembles with mapreduce,"PVLDB, vol. 2, no. 2, pp. 1426–1437, 2009.
17. J. Dean et al. MapReduce: a flexible data processing tool. Communications of the ACM, 53(1):72–77, 2010.
18. JaliyaEkanayake, ShrideepPallickara, and Geoffrey Fox, MapReduce for Data Intensive Scientific Analyses. In Fourth IEEE International Conference on eScience (978-0-7695-3535-7/08) eScience, 2008.
19. "GFS, MapReduce, and Hadoop" (Geeking with Greg, June 2006).
20.http://hadoop.apache.org/common/docs/current/hdfs design.html, 2009.
21. MapReduce and the Data Scientist Colin White, BI Research January 2012.
22. Big Data: A New World of Opportunities. NESSI White Paper, December 2012
23. Tomasz WiktorWlodarczyk, Yi Han, ChunmingRong: Performance Analysis of Hadoop for Query Processing. AINA Workshops 2011:507-513.
24. W. Tantisiriroj, S. Patil, and G. Gibson. Data-intensive file systems for internet services: A rose by any other name. Technical report, Carnegie Mellon University, 2008.