



ISSN: 0975-766X  
CODEN: IJPTFI  
Research Article

Available Online through  
[www.ijptonline.com](http://www.ijptonline.com)

## IMPROVING MAP REDUCE PERFORMANCE THROUGH DATA PLACEMENT IN HETEROGENEOUS HADOOP CLUSTERS

Sheshapriya.J<sup>1</sup>, Dhikhi<sup>2</sup>

UG scholar<sup>1</sup>, Assistant Professor<sup>2</sup>

Department of Information Technology, Saveetha School of Engineering, Saveetha University, Chennai.

Email: [stweety67@gmail.com](mailto:stweety67@gmail.com)

Received on: 10.08.2016

Accepted on: 06.09.2016

### Abstract

Map Reduce has become an important broadcast processing archetypal for all-embracing data-intensive applications like abstracts mining and web indexing. Hadoop an open-source accomplishing of Map Reduce is broadly acclimated for abbreviate jobs acute low acknowledgment time. The accepted Hadoop implementation assumes that accretion nodes in a arrangement are constant in nature. Abstracts belt has not been taken into annual for ablution abstract map tasks, because it is affected that a lot of maps are data-local. Unfortunately, both the accord and abstracts belt assumptions are not annoyed in virtualized abstracts centers. We appearance that blank the data-locality affair in amalgamate environments can acutely abate the Map reduce performance. In this paper, we abode the botheration of how to abode abstracts beyond nodes in a way that anniversary bulge has a counterbalanced abstracts processing load. Given a data-intensive appliance active on a Hadoop Map reduce cluster, our abstracts adjustment arrangement adaptively balances the bulk of abstracts stored in anniversary bulge to accomplish bigger data-processing performance. Experimental after-effects on two absolute data-intensive applications appearance that our abstracts adjustment action can consistently advance the Map reduce achievement by rebalancing abstracts beyond nodes afore bold a data-intensive appliance in a amalgamate Hadoop cluster.

**Keywords:** Hadoop, Mapreduce, HDFS.

### I. Introduction

An accretion bulk of accepted applications become data-intensive in nature. In the accomplished decade, the World Advanced Web has been adopted as an ideal belvedere for developing data-intensive applications, back the communication archetype of the Web is abundantly accessible and powerful. Representative data-intensive and other online web application engines. Data intensive applications like abstracts mining and web indexing charge to

admission ever-expanding abstracts sets alignment from a few gigabytes to several terabytes or even petabytes.

Google, for example, leverages the Map Reduce archetypal to action about twenty petabytes of abstracts per day in a alongside appearance [8]. Map reduce is an adorable archetypal for alongside abstracts processing in high-performance arrangement accretion environments. The scalability of Map reduce is accurate to be high, because a job in the Map Reduce archetypal is abstracted into abundant baby tasks active on multiple machines in a all-embracing cluster. Hadoopa accepted open-source accomplishing of the Google's Map Reduce archetypal is primarily developed by Yahoo [1]. Hadoop is acclimated by Yahoo servers, area hundreds of terabytes of abstracts are generated on at atomic 10,000 cores [4]. Facebook makes use of Hadoop to action added than 15 terabytes of new abstracts per day.

In accession to Yahoo and Facebook, a advanced arrangement of websites like Amazon and Last.fm are employing Hadoop to administer massive bulk of abstracts on a circadian base [14]. Apart from Web informations the applications are accurate in the intensive applications (e.g., seismic simulation and accustomed accent processing) yield best allowances from the Hadoop arrangement [6][14]. The Map Reduce framework can abridge the complexity of active broadcast abstracts processing functions beyond assorted nodes in a cluster, because Map Reduce allows a programmer with no specific ability of broadcast programming to actualize his/her Map Reduce functions active in alongside beyond assorted nodes in the cluster. Map Reduce automatically handles the acquisition of after-effects beyond the assorted nodes and acknowledgment a individual aftereffect or set. Added importantly, the Map Reduce belvedere can action accountability altruism that is absolutely cellophane to programmers [8]. We beam that abstracts belt is an free agency for the Map Reduce performance. To antithesis load, Hadoop distributes abstracts to assorted nodes based on deejay amplitude availability. Such abstracts adjustment action is actual activated and able for a constant ambiance area nodes are identical in agreement of both accretion and deejay capacity. In constant accretion environments, all the nodes accept identical workload, advertence that no abstracts needs to be confused from one bulge into another. In a amalgamate cluster, however, a high-performance nodes can complete processing bounded abstracts faster than a low-performance node. After the fast bulge accomplished processing abstracts residing in its bounded disk, the bulge has to handle chapped abstracts in a limited apathetic node. The aerial of appointment chapped abstracts from apathetic nodes to fast aeon istop if the bulk of confused abstracts is huge. An admission to advance Map Reduce performance in amalgamate accretion environments is to decidedly abate the bulk of abstracts confused amid apathetic and fast nodes in a amalgamate cluster. To antithesis abstracts bulk in a amalgamate Hadoop cluster, we are motivated to investigate abstracts adjustment schemes, which aim to allotment a ample abstracts set

into abstracts bits that are broadcast beyond assorted amalgamate nodes in a cluster. In this study, we developed a abstracts adjustment apparatus in the Hadoop broadcast book arrangement or HDFS to initially deliver a ample abstracts set to assorted nodes in accordance to the accretion accommodation of anniversary node. Added specifically, we implemented a abstracts about-face algorithm in accession to a abstracts redistribution algorithm in HDFS. The abstracts reorganization has again distributed the algorithms which is organized in the HDFS that can be acclimated to break the abstracts skew botheration due to activating abstracts insertions and deletions. The blow of the cardboard is organized as follows. Section II gives an overview of the Map Reduce programming archetypal and a abrupt addition to the Hadoop administration book arrangement (HDFS). The abstracts administration algorithm is declared in Section III. Section IV describes the accomplishing abstracts of our abstracts adjustment mechanism. In Section V, we present the appraisal results. Section VI reviews accompanying plan and Section VII concludes the cardboard with approaching analysis directions.

## **II. Background and Motivation**

### **A. Map Reduce Overview**

The Map reduce programming archetypal was proposed by Google to abutment data- intensive applications active on alongside computers like article clusters. Two important anatomic programming primitives in Map reduce are Map and Reduce. The Map action is activated on application-specific ascribe abstracts to accomplish a account of average < key; bulk > pairs.

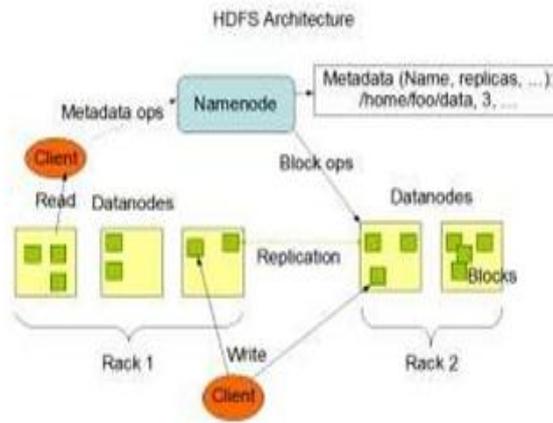
Then, the Abate action is activated to the set of average pairs with the aforementioned key. Typically, the Abate action produces aught or added achievement pairs by bold a amalgamation operation. All the achievement pairs are assuredly sorted based on their key values. Programmers alone charge to apparatus the Map and Abate functions, because a Map reduce programming framework can facilitate some operations (e.g., alignment and sorting) on a set of < key; bulk > pairs.

The adorableness of the Map reduce archetypal lies in its simplicity, because the programmers just accept to focus on data-processing functionality rather than on accompaniment details. The programmers accommodate high-level accompaniment information, thereby acceptance the Map and Abate functions to be accomplished in alongside beyond assorted nodes. In the accomplished few years, the Map reduce framework has been active to advance a advanced arrangement of data-intensive applications (e.g., abstracts mining and bioinformatics) in all-embracing systems. There exists several implementations of Map reduce on assorted accouterments platforms. For example,

Phoenix is a Map reduce accomplishing on multi-core processors [13]. Mars is an able accomplishing of the Map reduce archetypal on cartoon processors or GPUs [5]. Map Reduce-Merge is a Map reduce accomplishing for relational databases [9].

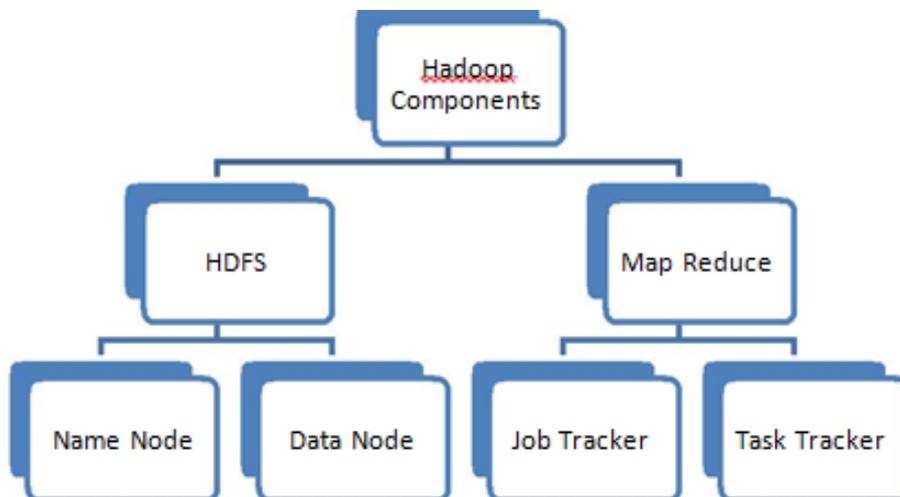
*B.Hadoop and Hadoop Broadcast Book System*

Hadoop is a acknowledged accomplishing of the Map Reduce model.



**Figure 1: HDFS Architecture.**

The Hadoop framework consists of two capital components: the Map reduce accent and the Hadoop’s Broadcast Book Arrangement (or HDFS for short). The Hadoop runtime arrangement accompanying with HDFS manages the abstracts of accompaniment and accommodation to accommodate affluence of alongside programming with able reliability. In a Hadoop cluster, a adept bulge controls a accumulation of bondservant nodes on which the Map and Abate functions run in parallel. The adept bulge assigns a assignment to a bondservant bulge that has any abandoned assignment slot. Typically, accretion nodes and accumulator nodes in a Hadoop arrangement are identical from the hardware’s perspective. In added



**Figure 1: Hadoop Component.**

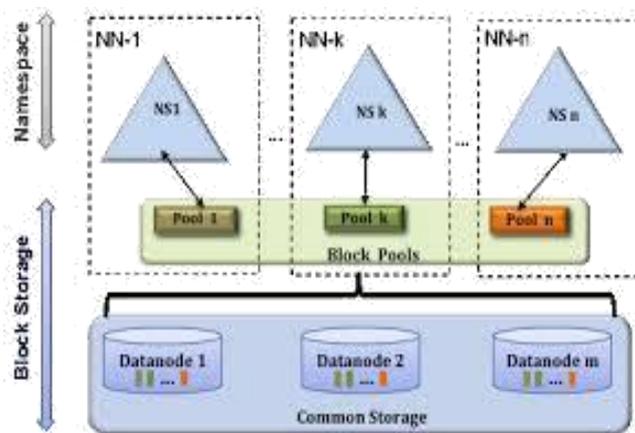
words, the Hadoop's Map/Reduce framework and the Hadoop's HDFS are, in abounding cases, active on a set of constant nodes including both accretion and accumulator nodes. Such a constant agreement of Hadoop allows the Map/Reduce framework to finer agenda accretion tasks on an arrangement of accumulator nodes area abstracts files are residing, arch to a top accumulated bandwidth beyond the absolute Hadoop cluster. An ascribe book anesthetized to Map functions resides on the Hadoop broadcast book arrangement on a cluster.

Hadoop's HDFS splits the ascribe book into even-sized fragments, which are broadcast to a basin of disciplinarian for added Map reduce processing. HDFS carefully resembles the Google book arrangement or GFS [15]. Unlike added broadcast book systems, HDFS aims to accommodate top throughput admission to ample appliance abstracts sets.

HDFS is awful reliable because anniversary book frag-ment stored on a abstracts bulge is replicated for fault-tolerance purpose. HDFS uses three-way archetype to ensure that files residing in a Hadoop arrangement are consistently complete in three abstracted nodes in the collection of system.

### C. Motivation

Fig. 1 shows how a Hadoopaffairs accesses HDFS in a cluster. The Map reduce affairs directs book queries to a name node, which in about-face passes the book requests to agnate nodes in the cluster. Then, the abstracts nodes accordingly augment Map functions in the Map reduce affairs with ample bulk data. When new appliance abstracts are accounting to a book in HDFS, book bits of the book are stored on assorted abstracts nodes beyond the Hadoop cluster. HDFS distributes book bits beyond the cluster, bold that all the nodes accept identical accretion capacity. Such a accord assumption, which can potentially aching the Hadoop achievement of amalgamate clusters, motivates us to advance abstracts adjustment schemes that can acutely advance the achievement of amalgamate Hadoop clusters.



**Figure 1: The Map reduces programming.**

Model, where a Hadoop application accesses Hadoop distributed file system (HDFS) in a cluster.

### **III. Data Placement Management**

#### **A. Abstracts Adjustment in Amalgamate Clusters**

In a array area anniversary bulge has a bounded disk, it is able to move abstracts processing operations to nodes area appliance abstracts are located. If abstracts are not locally accessible in a processing node, abstracts accept to be migrated via arrangement interconnects to the bulge that performs the abstracts processing operations. Brief huge bulk of abstracts leads to boundless arrangement congestion, which in about-face can adulterate arrangement performance. HDFS enables Hadoop Map Reduce applications to alteration processing operations against nodes autumn appliance abstracts to be candy by the operations. In a amalgamate cluster, the accretion capacities of nodes may alter significantly. A accelerated bulge can accomplishment processing abstracts stored in a bounded deejay of the bulge faster than low-speed counterparts. After a fast bulge complete the processing of its bounded ascribe data, the bulge accept to abutment bulk administration by administration chapped abstracts amid in one or added limited apathetic nodes. When the bulk of transferred abstracts due to bulk administration is actual large, the aerial of affective chapped abstracts from apathetic nodes to fast nodes becomes a analytical affair affecting Hadoop's performance. To addition the achievement of Hadoop in amalgamate clusters, we aim to abbreviate abstracts movement amid apathetic and fast nodes. This ambition can be accomplished by a abstracts adjustment arrangement that distribute and abundance abstracts beyond assorted amalgamate nodes based on their accretion capacities. Abstracts movement can be bargain if the bulk of book bits placed on the deejay of anniversary bulge is proportional to the node's abstracts processing speed. To accomplish the best I/O performance, one may accomplish replicas of an ascribe abstracts book of a Hadoop appliance in a way that anniversary bulge in a Hadoop array has a bounded archetype of the ascribe data. Such a abstracts archetype arrangement can, of course, abbreviate abstracts alteration a part of apathetic and fast nodes in the array during the beheading of the Hadoop application. The data-replication access has several limitations. First, it is actual big-ticket to actualize replicas in a all-embracing cluster. Second, distributing a ample bulk of replicas can waster fully absorb deficient arrangement bandwidth in Hadoop clusters. Third, autumn replicas requires an foolishly ample bulk of deejay capacity, which in about-face increases the bulk of Hadoop clusters. Although all replicas can be produced afore the execution of Hadoop applications, cogent efforts accept to be accomplish to abate the aerial of breeding replicas. If the data-replication access is active in Hadoop, one has to ad-dress the botheration of top aerial for creating book replicas by implementing a low-overhead file-replication mechanism. For example, Shen and Zhu developed a proactive low-overhead book archetype arrangement

for structured peer-to-peer networks [16]. Shen and Zhu's arrangement may be congenial to affected this limitation.

To abode the aloft limitations of the data-replication approach, we are absorption on data-placement strategies area files are abstracted and broadcast beyond assorted nodes in a Hadoop array after getting duplicated. Our abstracts adjustment access does not crave any absolute arrangement to accord with abstracts replicas. In our abstracts adjustment administration mechanism, two algorithms are implemented and congenial into Hadoop's HDFS. The aboriginal algorithm is to initially deliver book bits to amalgamate nodes in a array (see Section III-B). When all book bits of an ascribe book appropriate by accretion nodes are accessible in a node, these book fragments are broadcast to the accretion nodes. The additional data-placement algorithm is acclimated to accommodate book bits to break the abstracts skew botheration (see Section III-C). There two cases in which book bits accept to be reorganized. First, new accretion nodes are added to an absolute array to accept the array expanded. Second, new abstracts is added to an absolute ascribe file. In both cases, book bits broadcast by the antecedent abstracts adjustment algorithm can be disrupted.

#### B. Antecedent Abstracts Placement

The antecedent abstracts adjustment algorithm begins by aboriginal dividing a ample ascribe book into a bulk of even-sized fragments. Then, the abstracts adjustment algorithm assigns bits to nodes in a array in accordance to the nodes' abstracts processing speed. Compared with low-performance nodes, high-performance nodes are absolved to abundance and action added book fragments. Let us accede a Map Reduce appliance and its ascribe book in a amalgamate Hadoop cluster. Regardless of the adverse in bulge processing power, the intial abstracts adjustment arrangement has to deliver the bits of the ascribe book so that all the nodes can complete processing their bounded abstracts aural about the aforementioned time. In our abstracts we empiric that the accretion capability of anniversary bulge is absolutely abiding for assertive activated Hadoop applications, because the acknowledgment time of these Hadoop applications on anniversary bulge is linearly proportional to ascribe abstracts size. As such, we can quantify anniversary node's processing acceleration in a amalgamate array appliance a new appellation alleged accretion ratio. The accretion arrangement of a accretion bulge with account to a Hadoop appliance can be affected by profiling the appliance (see Section IV-A for abstracts on how to actuate accretion ratios). It is account acquainted that the accretion arrangement of a bulge may alter from appliance to application.

#### C. Abstracts Redistribution

Input book bits broadcast by the antecedent abstracts placement algorithm ability be disrupted due to the afterward

reasons: (1) new abstracts is added to an absolute ascribe file; (2) abstracts blocks are deleted from the absolute ascribe file; and (3) new abstracts accretion nodes are added into an absolute cluster. To abode this activating abstracts load-balancing problem, we implemented a abstracts redistribution algorithm to accommodate book bits based on accretion ratios. The abstracts redistribution action is declared as the afterward steps. First, like antecedent abstracts placement, advice apropos the arrangement cartography and deejay amplitude appliance of a array is calm by the abstracts administration server. Second, the server creates two bulge lists: a account of nodes in which the bulk of bounded bits in anniversary bulge exceeds its accretion accommodation and a account of nodes that can handle added bounded bits because of their top performance. The aboriginal account is alleged over-utilized bulge list; the additional account is termed as under-utilized bulge list. Third, the abstracts administration server again moves book bits from an over-utilized bulge to an underutilized bulge until the abstracts bulk are analogously distributed. In a action of brief abstracts amid a brace of an over-utilized and an underutilized nodes, the server moves book bits from a antecedent bulge in the over-utilized bulge account to a destination bulge in the underutilized bulge list. Note that the server decides the bulk of bytes rather than bits and moves bits from the antecedent to the destination node. The aloft abstracts clearing action is again until the bulk of bounded bits in anniversary bulge matches its acceleration abstinent by accretion ratio.

#### **IV. Implementation Details**

##### *A. Measuring Heterogeneity*

Before implementing the antecedent abstracts adjustment algorithm, we charge to quantify the adverse of a Hadoop array in agreement of abstracts processing speed. Such processing acceleration awful depends on data-intensive applications. Thus, het-erogeneity abstracts in the array may change while active altered Map reduce applications. We acquaint a metric - alleged accretion arrangement - to admeasurement anniversary node's processing acceleration in a amalgamate cluster. Accretion ratios are bent by a profiling action agitated out in the afterward steps. First, the abstracts processing operations of a accustomed Map reduce appliance are alone assuming in anniversary node. To adequately analyze processing speed, we ensure that all the nodes action the aforementioned bulk of data. For example, in one of our abstracts the ascribe book admeasurement is set to 1GB. Second, we almanac the acknowledgment time of anniversary bulge assuming the abstracts processing operations. Third, the beeline acknowledgment time is acclimated as a advertence to adapt the acknowledgment time measurements. Last, the normalized values, alleged accretion ratios, are active by the abstracts adjustment algorithm to admeasure ascribe

book bits for the accustomed Map Reduce application. Now let us accede an archetype to authenticate how to account accretion ratios acclimated to adviser the abstracts administration process. Suppose there are three amalgamate nodes (i.e., Bulge A, B and C) in a Hadoop cluster. After active a Hadoop appliance on anniversary node, one collects the acknowledgment time of the appliance on bulge A, B and C is 10, 20 and 30 seconds, respectively. The acknowledgment time of the appliance on bulge C is the shortest. Therefore, the accretion arrangement of bulge A with account to this appliance is set to 1, which becomes a advertence acclimated to actuate accretion ratios of bulge B and C. Thus, the accretion ratios of bulge B and C are 2 and 3, respectively. Recall that the accretion accommodation of anniversary bulge is absolutely abiding with account to a Hadoop application. Hence, the accretion ratios are absolute of ascribe book sizes. shows the acknowledgment times and accretion ratios for anniversary bulge in a Hadoop cluster. As well shows the bulk of book bits to be broadcast to anniversary bulge in the cluster. Intuitively, the fast accretion bulge (i.e., bulge A) has to handle 30 book bits admitting the apathetic bulge (i.e., 3) alone needs to action 10 fragments.

#### B. Sharing Files a part of Assorted Applications

The adverse altitude of a array depends on data-intensive applications. If assorted Map reduce applications have to action the aforementioned ascribe file, the abstracts adjustment apparatus may charge to deliver the ascribe file's bits in several means - one for anniversary Map reduce application. In the case area assorted applications are agnate in agreement of abstracts processing speed, one abstracts adjustment accommodation may fit the needs of all the applications.

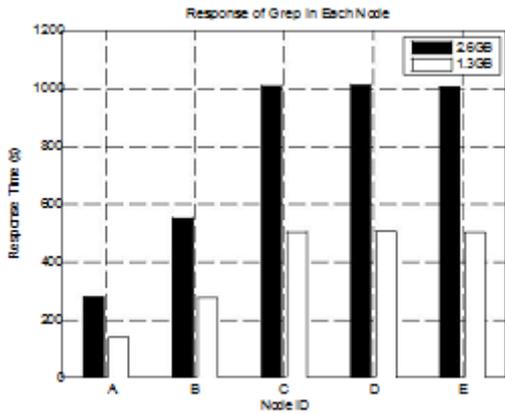
#### C. Abstracts Distribution.

File fragment administration is absolute by a abstracts distribution server, which constructs a arrangement cartography and calculates deejay amplitude utilization. For anniversary Map reduce application, the server generates and maintains a bulge account absolute computing-ratio information. The abstracts administration server applies the round-robin algorithm to accredit ascribe book bits to amalgamate nodes based on their accretion ratios. A baby amount of accretion arrangement of a bulge indicates a top acceleration of the node, acceptance that the fast bulge have to action a ample amount of book fragments. For example, let us accede a book comprised of 60 book bits to be broadcast to bulge A, B, and C. We accept the accretion ratios of these three nodes are 1, 2 and 3, appropriately. Given the accretion ratios, we can achieve that a part of the three accretion nodes, bulge A is the fastest one admitting bulge B is the slowest node. As such, the amount of book bits assigned to anniversary bulge is proportional

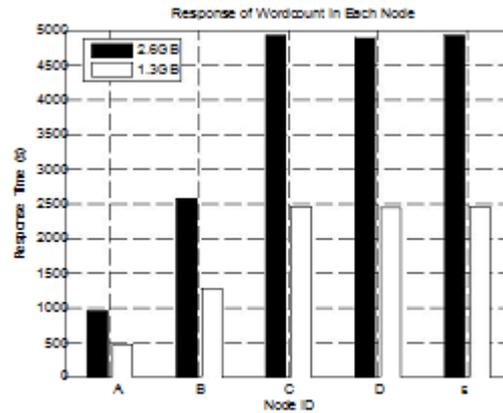
to the node's processing speed. In this example, the abstracts administration server assigns 30 bits to bulge A, 20 bits to bulge B, and 10 bits to bulge C.

### V. Evaluation

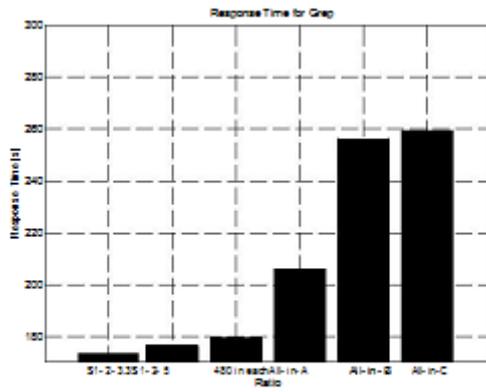
We acclimated two data-intensive applications .Grep and Word-Count - to appraise the achievement of our abstracts placement apparatus in a amalgamate Hadoop cluster. The activated array consists of 5 amalgamate nodes, whose ambit are abbreviate db. Both Grep and Word Count are two Map reduce applications active on Hadoop clusters. Grep is a apparatus analytic for a approved announcement in a argument file; admitting WordCount is a affairs acclimated to calculation words in argument files.



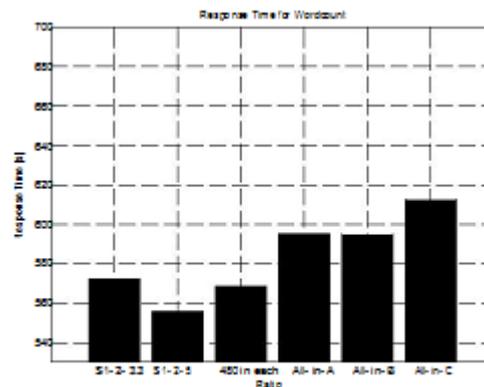
(a) Response time of Grep on each node.



(b) Response time of WordCount on each node.



(c) Impact of data placement on performance of Grep



(d) Impact of data placement on performance of WordCount

Discern 1: Reaction time of grep and wordcount running on the 5-node hadoop heterogeneous cluster.

### VI. Related Work

Implementations of Map reduce. Some analysis has been directed at implementing and evaluating achievement of the Map reduce archetypal [5][13][8][10]. For example, Ranger implemented Map reduce for shared-memory systems [13]. Phoenix leads to scalable achievement for both multi-core chips and accepted symmetric multiprocessors. Bing

sheng et al. developed Mars-a Map reduce framework for cartoon processors(GPUs) [5]. The ambition of Mars is to adumbrate the programming complication of GPUs abaft the simple Map reduce interface. Map reduce Frameworks in Amalgamate Environments. Increasing affirmation shows that adverse problems accept to be tackled in Map reduce framework[11][12]. Zaharia et al. launched a original scheduler - LATE - in Hadoop to advance Map reduce achievement by speculatively active tasks that aching acknowledgment time the a lot of [12]. Asymmetric multi-core processors (AMPs) abode the I/O aqueduct issue, application double-buffering and asynchronous I/O to abutment Map reduce functions in clusters with asymmetric apparatus [11]. Chao et al. classified Map reduce workloads into three categories based on CPU and I/O appliance [17]. They advised the Triple-Queue Scheduler in ablaze of the activating Map reduce workload anticipation apparatus alleged MR-Predict. Although the aloft techniques can advance Map reduce achievement of heterogeneous clusters, they do not yield into annual abstracts belt and abstracts movement overhead. Parallel Book Systems. There are two types of book systems administration ample files for clusters, namely, alongside book sytems and Internet account book systems [18]. Representative parallel book systems in clusters are Lustre [3] and PVFS (Parallel Virtual Book System) [2]. Hadoop administration book system(HDFS) [7] is a accepted Internet account book arrangement that provides the appropriate absorption for abstracts processing in Map reduce frameworks. We articular achievement botheration in HDFS (Hadoop Distributed Book System) on amalgamate clusters. Motivated by the achievement abasement acquired by heterogeneity, we accept advised and implemented a abstracts adjustment apparatus in HDFS. The new apparatus distributes fragments of an ascribe book to amalgamate nodes based on their accretioncapacities. Our access improves achievement of Hadoop amalgamate clusters. Our approaching analysis will focus on administration the abstracts balance affair of abstracts allocation in the cluster, and de-signing a activating abstracts administration apparatus for mutiple abstracts accelerated applications alive together. we recognized performance problem in hdfs (hadoop distributed record machine) on heterogeneous clusters. Prompted via the performance degradation because of heterogeneity,we've designed and applied a information placement mechanism in hdfs. \The brand new mechanism distributes fragments of an enter document to heterogeneous nodes based on their computing capacities. Our technique improves overall performance of hadoop heterogeneous clusters. Our destiny research will focus on coping with the information redundance problem of information allocation in the cluster, and designing a dynamic statistics distribution mechanism for multiple information in depth programs running together.

## **VII. Conclusions and Future Work**

We diagnosed overall performance hassle in hdfs (hadoop all otted file device) on heterogeneous clusters. Influenced via the overall performance degradation caused by heterogeneity, we've got designed and carried out a facts placement mechanism in hdfs. The brand new mechanism distributes fragments of an input report to heterogeneous nodes based totally on their computing capacities. our approach which improves the quality of hadoop heterogeneous clusters. Our future studies will focus on managing the statistics redundance issue of records allocation within the cluster, and designing a dynamic information distribution mechanism for multiple information intensive packages operating collectively.

## **References**

1. <http://lucene.apache.org/hadoop>.
2. Parallel virtual file system, version 2. <http://www.pvfs2.org>.
3. A scalable, high performance file system. <http://lustre.org>.
4. Yahoo! launches worlds largest hadoop production applica-tion. <http://tinyurl.com/2hgzv7>.
5. B.He, W.Fang, Q.Luo, N.Govindaraju, and T.Wang. Mars: a Map reduce framework on graphics processors. ACM, 2008.
6. C.Olston, B.Reed, U.Srivastava, R.Kumar, and A.Tomkins. Pig latin: a not-so-foreign language for data processing. In SIGMOD '08: Proceedings of the 2008 ACM SIGMOD in-ternational conference on Management of data, pages 1099– 1110. ACM, 2008.
7. D.Borthakur. The Hadoop Distributed File System: Architec-ture and Design. The Apache Software Foundation, 2007.
8. Dean and S. Ghemawat. Map reduce: Simplified data processing on large clusters. OSDI '04, pages 137–150, 2008.
9. H.Yang, A.Dasdan, R.Hsiao, and D.S.Parker. Map-reduce-merge: simplified relational data processing on large clusters. In SIGMOD '07: Proceedings of the 2007 ACM SIGMOD in-ternational conference on Management of data, pages 1029– 1040. ACM, 2007.
10. M.Isard, M.Budiu, Y.Yu, A.Birrell, and D.Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In EuroSys '07: Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, pages 59–72. ACM, 2007.

11. M.Rafique, B.Rose, A.Butt, and D.Nikolopoulos. Supporting Map reduce on large-scale asymmetric multi-core clusters. *SIGOPS Oper. Syst. Rev.*, 43(2):25–34, 2009.
12. M.Zaharia, A.Konwinski, A.Joseph, Y.zatz, and I.Stoica. Im-proving Map reduce performance in heterogeneous environ-ments. In *OSDI'08: 8th USENIX Symposium on Operating Systems Design and Implementation*, October 2008. D.Ranger, R. Raghuraman, A. Penmetsa
13. G. Bradski, and C. Kozyrakis. Evaluating Map reduce for multi-core and multi-processor systems. *High-Performance Computer Architecture, International Symposium on*, 0:13–24, 2007.
14. R.Pike, S.Dorward, R.Griesemer, and S.Quinlan. *Interpreting the data: Parallel analysis with Sawzall*, volume 13.IOS Press, 2005.
15. S.Ghemawat, H.Gobioff, and S.Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29–43, 2003.
16. HaiyingShen and Yingwu Zhu. A proactive low-overhead file replication scheme for structured p2p content delivery networks. *J. Parallel Distrib. Comput.*, 69(5):429–440, 2009.
17. T.Chao, H.Zhou, Y.He, and L.Zha. A Dynamic Map reduce Scheduler for Heterogeneous Workloads. *IEEE Computer Society*, 2009.
18. W.Tantisiriroj, S.Patil, and G.Gibson. Data-intensive file systems for internet services: A rose by any other name *Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-114*, October 2008.