Review Of Fair Job Scheduling Approaches For Map Reduce

Karambir[1], Ambika Rani[2]

[1]*Assistant Professor, Department of CSE, UIET, KUK, India*
[2] *Student of Department of CSE, UIET, KUK, India*

---***---

Abstract - *MapReduce is an emerging paradigm for big data concentrated processing. MapReduce provides convenient programming interfaces to allocate data concentrated works in a cluster environment. The capabilities of MapReduce are fault acceptance, an easy programming format and vast scalability. A selection of applications have accepted MapReduce with scientific assessment, web data handling and high performance computing. The problems of scheduling map-reduce jobs are mostly caused by area and management overhead and there is an essential to schedule various jobs in a common cluster. In this paper, we reviewed the MapReduce job scheduling on storage and the basis of time.*

***KeyWords*: MapReduce, Scheduling, HDFS, NameNode, DataNode, Data Locality**

1. INTRODUCTION

The exponential growth of knowledge initial given challenges to up-to-date businesses like Google. They travel through terabytes and petra bytes of knowledge to work out. Existing system were turning into insufficient method into such massive knowledge sets. This technique stimulated lots of interest as a result of several challenges was facing by business and it was not possible for everyone to rediscover their own proprietary tool [1].

Hadoop may be a hub of the computing communications for several internet corporations, like yahoo etc. Additional ancient businesses, like medium and telecommunication, area unit starting to adopt this technique too. Hadoop is associate free supply structure for writing and organizing distributed applications.

Available—Hadoop operate on massive clusters of trade goods machines or cloud services like amazon's.

Scalable— It stability linearly to work on a more knowledge by adding together a lot of slots to the cluster.

Plain— It permits users to rapidly mark economical analogous code.

Hadoop's accessibility as well as ease provides it a foothold more writing and consecutively massive circulated programs.

1.1 Other parts of Hadoop :

HBase: is associate open supply distributed information that affords low-latency, quick lookups in Hadoop. It permits users to conduct updates, inserts. It's written in Java. It runs on the highest of HDFS. It will function the input and output for the MapReduce.

Pig: Pig Latin may be a Hadoop-based language developed by yahoo .It is quite straightforward to be expressed and really long knowledge pipeline. Pig is high-level platform wherever the MapReduce programs area unit created that is employed with Hadoop. It's a high level processing system wherever the information sets area unit analyzed that happens in high level language.

Hive: may be a Hadoop-based knowledge warehousing-like structure initially developed by Facebook. Hive infrastructure is constructed on the highest of Hadoop that facilitate in providing report, question and analysis

Sqoop: Sqoop may be a command-line interface platform that's used for transferring knowledge between relational databases and Hadoop.

Oozie: Oozie may be a progress process system that lets users outline a series of jobs written in multiple languages like Map reduce, pig and hive that link along. Oozie may be a java based mostly web-application that runs in an exceedingly java servlet. Oozie uses the information to store definition of progress that's a group of actions.

Chukwa: Chukwa may be a knowledge assortment and analysis framework that is employed to method and analyze the large amount logs. it's engineered on the highest of the HDFS and MapReduce framework

Zookeeper: it is a centralized service that has distributed synchronization and providing cluster services and maintains the configuration data etc.

2. LITERATURE REVIEW

Faraz Ahmad[2] proposed an implementation called Tarazu, involving a suite of optimizations to enhanced mapreduce achievement on assorted clusters. The planned optimizations was a communication-aware load balancing method of map calculation diagonally the nodes, a communication aware scheduling of map calculation to avoid bursty network track and a predicted load balancing of Reduce calculation across the nodes. The online measurement-based heuristics to approximate the information needed for making application.

Yanfeng Zhang [3] proposed a framework called IMapReduce that used Hadoop to process structured data iteratively. IMapReduce tackled some Hadoop problems to process iterative calculation: the waste of resources to create, plan and wipe out jobs that performed stable functions in each iteration; the performance consequence to load and shuffle fixed data that remains the same during the iterations and the serial execution of jobs in each iteration, resulting in synchronism in reduce and map responsibilities. IMapReduce familiarized the concept of persistent tasks to deal with the problem of waste of resources, avoiding avoidable creation, scheduling and devastation of tasks. This mechanism also avoided constantly data load and shuffle procedure between repetitions. To break the synchronism and allowed the execution of map tasks as soon as possible, IMapReduce implemented a persistent socket connection, keeping alive communication between tasks to store passing data.

Xiaohong Zhang [4] proposed a two-phase execution engine of reduce tasks to handle with massive remote data access delays that may corrupt system performance. The deprivation was related to huge remote I/O operations to copy the middle results of map tasks. In this the engine selected the nodes to run reduce tasks and prefetched middle results for reduce tasks and the selected nodes allocated computing and memory resources to execute the reduce tasks. S. Vikram Phaneendra[5] proposed pruning techniques that was based on voronoi diagrams to decrease the number of predictable distance calculations. The UK-Means algorithm was the first algorithm to grip the ambiguous data or objects vague data distances. To decrease number of expected distance (ED) calculations, introduced an incomplete ED evaluation method and had incorporated the method in VDBiP and by grouping of this algorithm was called as hybrid algorithm. These techniques were analytically established to be more effective than the basic bounding-box-based technique.

Minghong Lin[6] proposed an overlapping model between map and Shuffle phases. The prospects were basis of two paired scheduling algorithms known as MaxSRPT and SplitSRPT. MaxSRPT minimized the average response time of the queue, while SplitSRPT addressed the poor performance of MasSRPT when jobs were more unbalanced. An analytical model proved that the problem of minimizing response time in the proposed NP-hard model. Faraz Ahmad[7] proposed mapreduce with communication overlap (MaRCO), which was directed to overlappe of the shuffle with the Reduce calculation. Hadoop data flow was improved allowing the procedure of reduce tasks on limited data. MaRCO breaks task into many smaller salutation on partial data from some map tasks, and a last reducing step re-reduces all the partial reduce outputs to produce the final output.

Hisham Mohamed [8] proposed to change the Hadoop data flow by using MPI to overlap Map and Reduce phases. Map and Reduce phases were executed in a parallel manner by exchanging partial middle data during a pipeline delivered vided by MPI. In the recommended model Map and Shuffle phases was combined and work as a single phase. MPI and Shuffle enhanced the performance of the prototype. Jiong Xie [9] used a pre-shuffling approach to reduce the network overwork enforced by shuffle exhaustive applications. A push model used in-memory buffer and a two stage pipeline in the pre-shuffling scheme to exchange incomplete data between maps and reduce tasks, was implemented. Designing new shuffling approaches was very engaging for Hadoop clusters where network interconnected performance bottleneck the clusters shared a large number of applications. A Hadoop application execution time was affected by the shuffling phase, where an amount of data was transferred from map tasks to reduce tasks.

Vinod Kumar Vavilapallih [10] summarized the development, progress, and current state of readying of future generation of Hadoop's calculation platform: YARN. The new design introduced decouples the encoding model from the resource management communications, and delegates some programing functions. It considered YARN will function every a concrete making framework and also as a useful playground. They offered the next generation of Hadoop compute platform known as YARN, which departs from its familiar, monumental building. The decoupling of source management and programming framework, YARN provides: i.e. greater scalability, higher efficiency and enables a large number of different frameworks to efficiently share a cluster. Chitharanjan. K [11] planned Apache's Hadoop1 sensible there was a unit scopes of extensions and enhancements. The range of enhancements area unit planned to Hadoop that was associated with open supply implementation of Google's map/reduce framework. It permitted distributed, knowledge intensive and parallel applications by moldering a huge job into smaller tasks and a huge knowledge set into smaller partitions such every task processes a special partition in parallel. Hadoop used hadoop distributed classification system (HDFS) that was associate open supply implementation of the Google classification system (GFS) for storing knowledge. Map/Reduce application used HDFS for storing knowledge. Hadoop enhancements reviewed within the many factors like fault tolerance, quantifiability, knowledge section, load equalization, performance, load time interface amendment to mapreduce, changes to Hadoop framework, classification and layout.

Shekhar Gupta[12] proposed Hadoop was that the de-facto general place for large knowledge analytics applications. They offered schedulers for Hadoop clusters assigned tasks to nodes while not relevancy the possible of the nodes. It condensed the general job achievement time on a cluster of varied nodes by actively programing tasks on nodes supported optimally matching job necessities to node capabilities. The ensuing model was used to optimize allocation of tasks to servers. Initial results ensure that outturn hardware performs higher than the default Hadoop schedulers for varied clusters, and didn't negatively impact performance even on uniform clusters. Weikuan Yu [13] described Hadoop-A, stimulation framework that improves Hadoop through plug-in components for fast data movement. A new network-levitated merge algorithm was lead to combine data without reiteration and disk access. In addition, a full pipeline was designed to join the shuffle, merge and then reduce phases. The experimental results show that Hadoop-A expressively speeds up data movement in mapreduce and pairs the output of Hadoop. In accumulation, Hadoop-A expressively reduces disk accesses produced by intermediate data.

C.P.Saranya M.E[14]analyzed the performance mistreatment reaction time factors in Hadoop distributed classification system. Supported the dataset size and range of nodes that was made in Hadoop cluster, the performance of individual jobs area unit known. By mistreatment Johnson's rule, the optimum answer for individual jobs for various disks area unit been calculated. Further, the potency of the computation task was computed by the datasets taken and range of nodes that's generated in Hadoop distributed classification system. Nidhi Tiwari[15] examined that Map Cut back has become omnipresent for process massive knowledge volume jobs. This presented a survey of a number of the Map cut back programing algorithms planned for such advanced eventualities. Taxonomy was delivered for Map-reduce algorithms supported their runtime nature. Energy potency was generally achieved at the value performance and handiness. Knowledge distribution strategy was done in every of the key factors for rising the Hadoop MapReduce Energy potency. Work load intensity and blend analysis considerably contribute towards energy potency strategy. Apache Hadoop attracted strong care outstanding to its applicability to Big Data handling.

Ivanilton Polato[16] derived the information which retrieved from computer memory unit scale datasets referred to as huge knowledge semiconductor diode to the event of solutions to method data supported parallel and distributed computing. The analysis concluded, i.e. several attention-grabbing solutions developed within the studies were never incorporated into the framework; most publications lack comfortable formal documentation of the experiments, impeding their reproducibility; the systematic reviewed given during demonstration Hadoop had evolved into a solid platform to method massive datasets.

Chanwit Kaewkasi [17] presented Hadoop cluster for process huge knowledge engineered a prime twenty two ARM boards. These examined huge processing with Hadoop had been raising just, each on the computing cloud and enterprise readying. A cluster for large knowledge was totally different from associate MPI-based cluster in terms of the world of applications and also the software package stack. Associate MPI-based cluster focused on CPU-bound procedure tasks, but a giant knowledge cluster performed processing, that was I/O-bound. Several works reportable that associate ARM cluster was roughly 2-9 times slower than associate Intel-based cluster, but higher in terms of power consumption for many benchmarks.

Dan Wang and Wenbing Zhao [18] proposed a framework given Map cut back may be a reasonably software package framework for simply writing applications that method huge amounts of knowledge on massive clusters of trade goods hardware and urge higher allocation of tasks and cargo equalization, the map cut back work mode and task programing rule of Hadoop platform was analyzed. It indicates that it was effective in creating task allocation and achieving sensible balance once it's applied into the Hadoop platform that used solely Job huntsman programing.

Supriya Pati[19] proposed a novel job aware scheduling algorithm. Scheduling Algorithm was necessary for optimal utilization of cluster resources. Mapreduce word count benchmark was used to analyze the performance of scheduling algorithm. This algorithm scheduled job based on three criteria i.e. job execution time, earliest deadline first and workload of the job. Scheduling algorithm was used to increase resource utilization and reduced the average waiting time by 79% in best case and 23% in average case.

Table -1: Comparison of Schedulers [20]

Name	HBase	Hive	MongoDB	Redis	Cassandra	Drizzle
Description	Wide column store based on Apache Hadoop and on concepts of Big Table	Data Warehouse Software for Querying and Managing Large Distributed Datasets, built on Hadoop	One of the most popular Document Stores	In-memory Database with configurable options performance vs. persistency	Wide-column store based on ideas of BigTable and DynamoDB	MySQL fork with a pluggable micro-kernel and with an emphasis of performance over compatibility
Implementation language	Java	Java	C++	C	Java	C++
Database Model	Wide Column Store	Relational DBMS	Document Store	Key Value Store	Wide Column Store	Relational DBMS
Consistency Concepts	Immediate Consistency	Eventual Consistency	Eventual Consistency, Immediate Consistency	-	Eventual Consistency, Immediate Consistency	-
Concurrency	Yes	Yes	Yes	Yes	Yes	Yes
Durability	Yes	Yes	Yes	Yes	Yes	Yes
Replication Method	Selected Replication factor	Selected Replication factor	Master Slave Replication	Master Slave Replication	Selected Replication factor	Master Master Replication, Master Slave Replication

Table -2: Comparison among Components of Hadoop [20]

Scheduling algorithm	**Key technique**	**Implementation**	**Advantages**	**Disadvantages**	**Area**
FIFO	Non-adaptive	Schedule job based on their properties in first come- first –out	Cost of entire cluster scheduling is less Simple to implement and efficient	Designed only for single type of job Low performance when run multiple type of jobs	Execution time
Fair scheduling	Adaptive	Do a equal distribution of compute resources among the users in the system	Less complex Works well when both small and large cluster	Does not consider the job weight of each node	Execution time
Capacity	Adaptive	Maximize the resource utilization and throughput in multi-tenant cluster environment	Ensure guaranteed access with the potential to reuse unused capacity and prioritize jobs	The most complex among three schedulers	Execution time
LATE	Adaptive	Fault tolerance	robustness to node Heterogeneity.	poor performance due to the static manner in computing the progress of the tasks	Real slow task
SAMR	Adaptive	To improve MapReduce in terms of saving time of the execution and the system resources	Decrease the execution time of map reduce job. Improve the overall mapreduce performance in heterogeneous systems	Do not consider the data locality management for executing backup task	Task back up
Delay scheduler	Adaptive	To address the conflict between locality and fairness	Simplicity of scheduling	No particular	Dead line
Maestro	Adaptive	Proposed for map task, to improve overall performance of mapreduce computation	Provides higher locality in the execution of map task		Data locality

3. CONCLUSIONS

From the above survey, we can conclude that to find the execution time of the scheduling algorithm for the task is more complex, because the technique is quite successful in homogeneous scenario but fails in heterogeneous scenario, also scheduling cannot facilitate the queue management and pipeline system, so for removing the problem we can introduce the pipelining or perhaps designing of a hyper heuristic algorithm for the solution time and queue management system. The main purpose is to find the gap analysis problem and to study and resolve these problems. In the study, it was found that the problem of Replica of data increases the storage space and reduces the utilization of processing. It is very important challenge in previous study of our survey. Other problem , that was found with late scheduling which is related to deadline and time delay optimization problem. It is unfortunate that many of researchers ignore deadline problem for the reason they give efficient time delay which is not real condition scenario.

REFERENCES

[1] TomWhite,"Hadoop:The Definitive Guide",Mapreduce for the C loud,pp:1-525,2009.

[2] Faraz Ahmad, Srimat T. Chakradhar, AnandRaghunathan and T. N. Vijaykumar,"Tarazu: optimizing MapReduce on heterogeneous clusters",7th International conference on architectural support for programming languages and operating systems,pp:61-74, 2012

[3] Yanfeng Zhang, QixinGao, LixinGao, and Cuirong Wang, "Imapreduce: A distributed computing framework for iterative computation", Journal of grid computing vol1, issue(1),pp: 47-68, 2012

[4] Xiaohong Zhang, Guowei Wang, Zijing Yang, and Yang Ding, "A two-phase execution engine of reduce tasks in Hadoop MapReduce", IEEE International conference on systems and informatics (ICSAI), pp. 858-864, 2012

[5] N.Sateeesh and S.Vikram Phaneendra, "Efficient clustering uncertain data using hybrid algorithm", International journal of computer science and management research, vol 1,issue(3),pp:556-561, 2012

[6] Minghong Lin, Li Zhang,Adam Wierman, and Jian Tan, "Joint optimization of overlapping phases in reduce",Performance Evaluation,vol70,issue(10),pp: 720-735, 2013.

[7] Faraz Ahmad, Seyong Lee, MithunaThottethodi, and T. N. Vijaykumar, "MaRCO- MapReduce with Communication Overlap", Journal of parallel and distributed computing vol73,issue(5) ,pp: 608-620, 2013

[8] Hisham Mohamed and Stéphane Marchand-Maillet,"MRO-MPI:MapReduce overlapping using MPI and an optimized data exchange policy",Parallel Computing vol39,issue(12),pp: 851-866, 2013.

[9] Jiong Xie, Yun Tian, Shu Yin, Ji Zhang, XiaojunRuan, and Xiao Qin, "Adaptive preshuffling in Hadoop clusters", Procedia Computer Science 18,pp:2458-2467, 2013

[10] Vavilapalli and Vinod Kumar, "Apache hadoop yarn: Yet another resource negotiator",ACM 4th annual Symposium on Cloud Computing, 2013.

[11] A.Kala Karun and K. Chitharanjan, "A Review on Hadoop-HDFS Infrastructure Extensions", IEEE Conference on Information and Communication Technologies (ICT), pp. 132-137, 2013

[12] Shekhar Gupta, Christian Fritz, Bob Price, Roger Hoover, and Johan deKleer, "ThroughputScheduler: Learning to Schedule on Heterogeneous Hadoop Clusters",10th International Conference On Autonomic Computing ICAC, 2013

[13] Weikuan Yu, Yandong Wang and XinyuQue, "Design and evaluation of network-leviated merge for Hadoop Acceleration", IEEE Transactions on Parallel and Distributed Systems,vol25,issue(3),pp:602-611, 2014

[14] R. Manopriya and C.P.Saranya, "Optimal Resource Allocation and Job Scheduling to Minimise the Computation Time under Hadoop Environment", International journal of latest trends in engineering and technology (ijltet),vol 3,issue (3),pp:75-82, 2014

[15] Nidhi Tiwari, Santonu Sarkar, Umesh Bellur, and Maria Indrawan,"An empirical study of hadoop's energy efficiency on a HPC cluster",14th International Conference on Computational Science,pp:62-72, 2014

[16] Ivanilton Polato, Reginaldo R, Alfredo Goldman and Fabio Kon, "A comprehensive view of Hadoop research-A systematic literature review", Journal of Network and Computer Applications,pp: 1-25, 2014

[17] Kaewkasi Chanwit, and Wichai Srisuruk, "A study of big data processing constraints on a low-power Hadoop cluster",IEEE International Conference on Computer Science and Engineering (ICSEC), 2014

[18] Meira Dania and Naomi Osato, "A Comparative Study of Scalable Implementations of the Alternating Least Squares Algorithm for Collaborative Filtering Recommendation", 2015

[19] Supriya Pati and Mayuri A Mehta,"Job Aware Scheduling in Hadoop For Heterogeneous Cluster",IEEE international Advance Computing Conference(IACC),pp: ; ml778-783, 2015

[20] Bincy P Andrews1 and Binu A,"Survey on Job Schedulers in Hadoop Cluster",International Organization of Scientific Research (IOSR) Journal of Computer Engineering (IOSR-JCE),pp: 46-50, 2013