Research Paper Technology

A Survey on Mapreduce and Data Partitioning in Hadoop Implementation

Ms.S.Ancy	Assistant Professor, Department of Information Technology Jeppiaar Institute of Technology,Sriperumbudhur
S. Srilakshmi Shendurika	Student , Department of Information Technology Jeppiaar Institute of Technology,Sriperumbudhur
B.Krithika	Student , Department of Information Technology Jeppiaar Institute of Technology,Sriperumbudhur
K.Nivetha	Student , Department of Information Technology Jeppiaar Institute of Technology,Sriperumbudhur

ABSTRACT

Big data is a collection of data sets which cannot be partitioned by using traditional data processing applications. The processing and storage of big data although made efficient and easy by hadoop, there are two things to be considered, a storage part (Hadoop Distributed File System) and a processing part (Map reduce). The performance of a map reduce depends on how the data are partitioned and then controlled in parallel manner. The current methods are far from optimal. The skew aware partitioning technology is used to detect the problems related to data partitioning that affects performance. Hadoop Online prototype is a system which supports continuous queries also helps in event monitoring and stream processing. Naive fair sharing algorithm is used to implement map slots and reduce slots efficiently and also various algorithms are used for optimizing the mapreduce and data partitioning. In this paper, the complete study of data partitioning and map reduce has been analyzed.

KEYWORDS Big data, Data partitioning, Map reduce.

1. Introduction

Big data is an evolving term that describes combinations of large amount of structured, semi-structured and unstructured data that has the potential to be extracted for information. The structured data are easily organized whereas unstructured data are not organized in pre-defined manner. International Data Corporation (IDC) defined big data as volume, variety, velocity, variability, veracity and complexity. The feature of big data includes high-performance and inexpensive processing power, data integration and quality capabilities, unstructured text management. Relational database management system is difficult to work with big data .Hadoop is used to overcome this problem. Hadoop is a java based programming framework which supports big data. Hadoop consists of mapreduce and hadoop distributed file system. The hadoop distributed file system consists of two nodes, data node and name node. The advantages of hadoop is that it is cost effective, flexible, fast, scalability, independent.. But to compute and analyze this huge data is not an easy task. Many challenges arise while processing these terabytes of data. The few main issues are Data partitioning, skew computing, scheduling delay, poor system utilization, and low degree of parallelism. Many researchers have promoted their ideas for addressing these issues. This paper portrays the various algorithms and methods that are been employed by various researchers and also facilitate the complete overview, merits and demerits of each of these methodologies. The remaining of the paper is organized as II related works, III evaluation IV conclusion.

2. Related works

Jeffrey Dean proposed that "Data processing is a critical impact on data-parallel performance; a large body of researches have been promoted to address this challenge. The users have to specify map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key is

employed. The run-time system takes care of the details of partitioning the input data, scheduling and processing. This user function allows the programmers to work without any experience in parallel and distributed system and easily utilize the resources of a large distributed system which is the merit of this system. It is also said that this method is highly scalable and large computation of data is easily parallelized [1].

Mohammad Hammoud in his paper has refered that Mapreduce gives a programming model for huge data processing. This mapreduce is admired by many functional style which helps to write their code independently and it is divided automatically into many maps and also distributed over many machines. Hadoop is the open source technology helps in implementation of mapreduce and schedules the map task. This leads to avoid the network traffic and decrease the performance. But hadoop schedules task without considering data locality which degrades the performance. To overcome this problem larts is used to improve the performance efficiently. Larts is described as a locality aware reduces task scheduler. It helps in analyzing all data location, sizes and combines all the required data for computation and increase the performance. So LARTS improves node-local, rack- local and off-rack traffic and increase the performance efficiently.

As per Ganesh Ananthanarayanan Popularity in the mapreduce clusters can be gathered by collecting the information about the jobs that has been submitted for execution. The history of the usage is recorded in order to prioritize the maximum used skew content. The metric used to capture the popularity of the files is the number of concurrent access. The larger files are said to contribute the maximum accesses, so the reduction in the contention leads to the greater improvement in the performance of the cluster. To address the con-

tent of the file, Scarlett introduced the replication of the content at the smallest granularity. The replication is done based on popularity that is predicted. The file replication factor is determined by two approaches namely the priority approach and the round robin approach. After replication the desired numbers of replicas of the blocks are placed in as many as distinct racks and machines as possible with uniformity of the load given.

Shadi Ibrahim said that Mapreduce has become very popular nowadays. In mapreduce, while shuffling, data skew problems are experienced. Though hash partitioning is used default in hadoop, it works well and good only if all the keys are equally present and they are uniformly stored in the data nodes. This blind partitioning leads to network congestion, performance degradation and unfairness in the reducers' input. So LEEN method gives a better way by tracking the intermediate key frequency and their distribution. The keys are sorted out in based on their locality and fairness value. The nodes are sorted out in descending order for each key value based on the frequency that each key has. This partitioning is done by fairness score value. The solution overcomes the partitioning by implementing the locality concept in reducer and achieved a 40% increase in the performance while applying.

Yong Chul Kwon as said that Skew Tune is an approach that is used in automatic skew mitigation in user defined MapReduce programs. Skew Tuning system acts as a replacement for existing Mapreduce. It minimizes the input from the user by identifying the task that has greatest processing time. The data that are unprocessed are repartitioned. This enables the full utilization of the nodes in the cluster as well as the preservation of the order of input such that the original output is reconstructed. Skew Tuning also evaluates the effectiveness of certain applications and makes it transparently compatible with the code that exists already. Even the complex workflow context and the advanced MapReduce algorithm transparency is maintained. The experimentation shows that this process of Skew Tune can improve the Hadoop datasets to a four times larger one.

Tyson Condie referred that Map reduce is a framework provided by hadoop to manage large distributed data. A modified Mapreduce is used to avoid the fault tolerance and reduce the output of map task. In this pipeline concept is introduced to interconnect machines, this leads to reduce completion time and extends the program utilization for all jobs. HOP (Hadoop online Prototype) is an extended version of hadoop map reduces which supports online aggregation and continuous queries. HOP provides a log of all jobs to the user. HOP allows writing programs and supporting processes such as event monitoring and stream process, this leads to retain fault tolerance and run unmodified programs.

As per Alexander Rasmussen**,** Big data is heavily increased on Map-Reduce for large scale data processing. There are many map reduce I/O bound are raised dramatically which degrades the performance, so reducing the bounds is the efficient

task. In order to decrease the map reduce I/O bounds, this paper implemented a themis an efficient I/O map reduce. Themis is a mapreduce concept that reads and writes data records to place exactly twice in the disk, which is the minimum requirement for data sets to fit in memory. This helps to reduce the I/O bounds of Map-Reduce. Themis also performs various mapreduce tasks, which includes log analysis; DNA read sequence alignment, and PageRank.

Hai Jin said that**,** many people focus on shuffling the data, but recently the network traffic is emerging in hadoop. Map reduce is used to avoid failures during partitioning. Many researches have done a research and found that approximately 23% is used as non-local map tasks. To overcome this problem they proposed the algorithm called maestro. Maestro reduces maps upto 34%. Maestro schedules task in two waves, first wave states that when the task is started, it is used to fill each node and to avoid issues like empty node. It fills according to higher capacity. The second is runtime scheduling which helps in mapping the task to the replicas. These two waves results in high performance. So the Maestro design is used to improve the efficiency and to map non local map task.

Matei Zaharia explained that, Data intensive cluster computing systems like Hadoop and Dryad become more popular because there is a need to share cluster between users. But there is some problem between mapreduce and data locality. To overcome this problem delay scheduling algorithm is introduced. This algorithm is used in various scheduling policies. When the job scheduled according to fairness cannot launch any task. By using this delay scheduling algorithm, the task will wait for small amount of time for other jobs to finish. 100% locality is achieved by using this delay scheduling algorithm and also increase the response time for small jobs.

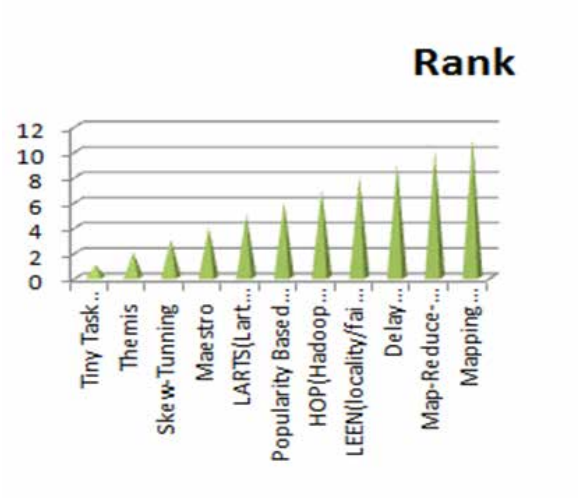
Kay Ousterhout said that, in data-parallel computing frameworks, user have some limitation in running short tasks. To improve the performance of data parallel processing cluster computing has been used. If the cluster is mostly utilized then an interactive job may need to wait for other jobs to complete. To avoid this tiny task is implemented in cluster to break large job into many smaller jobs. So time taken to perform the task is limited. It will improve the scalability in cluster framework.

Hung chin Yang in his paper refered that, Mapreduce is the programming language to process vast amount of data in large cluster. Map and reduce are the two

functions with simple interface to perform parallel implementation of many tasks. But some operations like joins are not supported by this function. To overcome this problem a new model called Map-Reduce-Merge is implemented in this paper. The data which are already partitioned and sorted is merged efficiently. Many join algorithm also implemented by using this Map-Reduce-Merge.

3. Evaluation

TITLE	ALGORITHM/PROCESS	ADVANTAGE
The Case for Tiny Tasks in Complete Cluster	Tiny Task Execution Framework	Large batch jobs can be split into small task that finish quickly
Themis: An I/O- Efficient map reduce	Themis	Wide variety of MapReducing Tasks
Skew-Tune: Mitigating Skew in Map-Reduce Application	Skew-Tunning	Reduce jobs runtime in the presence of skew and reduce overhead
Maestro: Replica-Aware Map Scheduling for MapReduce	Maestro	Higher locality in execution of map task and more balanced intermediate data distribution
Locality-Aware Reduce Task Scheduling for MapReduce	LARTS(Lart Aware Reduce Task Scheduling)	Computes data after recognizing input data network locations and size
Scarlett: Copying with Skewed Content Popularity in MapReduce Clusters	Popularity Based File Replication	Smooth Replicas are created
Online Aggregation and Continuous Query support in MapReduce	HOP(Hadoop Online Prototype)	Supports continuous queries
LEEN: Locality/Fairness-Aware Key Partitioning for MapReduce in the Cloud	LEEN(locality/fairness aware key partitioning)	Partitioning the key according to frequencies and fairness in data distribution among different data nodes
Delay scheduling A simple Technique for Achieving Locality and Fairness in Cluster Scheduling	Delay Scheduling Algorithm	Optimal data locality in a variety of workloads and increase the throughput
Map-Reduce-Merge: Simplified Relational Data Processing on Large Clusters	Map-Reduce-Merge	Parallel implementation of many real world tasks
MapReduce: Simplified Data Processing on Large Clusters	Mapping Functional Programming Model	High scalable and large computation of data



4. Conclusion

To conclude from the survey on mapreduce, there are several methods for efficient mapping and reducing. The data partitioning makes the easy way for mapreduce. Among all the methods, the location aware method is more efficient with respect to performance. As the keys are known with their locality, it is easy for the mapreduce to locate the intermediate keys which in turn indicate the data location paving the way for the decreased job completion time.

REFERENCES

[1] Ananthanarayanan, G.Agarwal, S. Kndula,S. Greenberg, A.Stoica, I.Harlan, D.Andharrise, E. Scarlett (2011). "Coping with Skewed Content Popularity in Map-Reduce Clusters. In Proc. EuroSys ". | [2] J. Dean,S. Ghemawat, (2010). "MapReduce a flexible data processing tool," Commun. ACM, vol(53), pp 72–77. | [3] Y. Bu, B. Howe, M. Balazinska,M. D. Ernst, (2010). "Hadoop: E□cient Iterative Data Processing on Large Clusters. In VLDB". | [4] S. Chen,S. W. Schlosser,(2008). "MapReduce Meets Wider Varieties of Applications," IRP-TR-08-05, Intel Research. | [5] Y. Xu, P. Kostamaa, X. Zhou,L. Chen,(2008) "Handling data skew in parallel joins in shared-nothing systems. In Proc of the SIGMOD Conf". | [6] Yu et. al. DryadLINQ,(2008). "A system for general-purpose distributed data-parallel computing using a high-level language. In Proc. of the 8th OSDI Symp". | [7] M. Isard, M. Budiu, Y. Yu, A. Birrell,D. Fetterly, (2007). "Dryad: distributed data-parallel programs from sequential building blocks," in Proceedings of the 2nd ACM European Conference on Computer Systems (EuroSys '07), Lisbon, Portugal, pp59–72. | [8] M. Isard, M. Budiu, Y. Yu, A. Birrell, D. Fetterly,Dryad,(2007). "Distributed data-parallel programs from sequential building blocks. In EuroSys",pages 59–72. | [9] S. F. Goldsmith, A. S.Aiken,D. S.Wilkerson, (2007). " Measuring empirical computational complexity. In ESEC/FSE". | | | [10] G. Candea, S. Kawamoto, Y. Fujiki, G. Friedman,A, (2004). Fox. Microreboot "A Technique for Cheap Recovery.In OSDI". | [11] Y. Yu, P. K. Gunda, M. Isard, (2009). "Distributed aggregation for data-parallel computing: interfaces and implementations. In Proc. of the 22nd SOSP Symp". | [12] A.Szalay, A. Bunn, J. Gray, I. Foster, I. Raicu, (2006). "The Importance of Data Locality in Distributed Computing Applications" NSF Workflow Workshop. | [13] Ousterhout, K., Wendell, P., Zaharia, M.Sto- Ica, I. Sparrow, (2013). "Scalable scheduling for sub-second parallel jobs. Tech. Rep. UCB/EECS-2013-29" EECS Department, University of California, Berkeley. | [14] B. Hindman, A. Konwinski, M. Zaharia,I. Stoica, (2009). "A common substrate for cluster computing. In Workshop on Hot Topics in Cloud Computing (HotCloud)". | [15] E. A. Brewer. Combining Systems and Databases: A Search Engine Retrospective. In J. M. Hellerstein and M. Stonebraker, (2005), editors, Readings in Database Systems, Fourth Edition, Cambridge, MA. MIT Press. | [16] F. Chang etal, Bigtable, (2006), A Distributed Storage System for Structured Data. In OSDI, pages 205–218. | [17]B. Chattopadhyay, L. Lin, W. Liu, S. Mittal, P. Aragonda, V. Lychagina, Y. Kwon, and M. Wong. Tenzing, (2011). A SQL Implementation On The MapReduce Framework. In Proc. VLDB Endowment. | [18] E. Bauer, X. Zhang, and D. Kimber, (2009). Practical System Reliability (pg. 226). Wiley-IEEE Press. | [19] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, (2009). Detecting large-scale system problems by mining console logs. In SOSP. | [20] B. Hindman, A. Konwinski, M. Zaharia, and I. Stoica, (2009). A common substrate for cluster computing. In Workshop on Hot Topics in Cloud Computing (HotCloud).