RESEARCH LETTER OPEN ACCESS

QoS oriented MapReduce Optimization for Hadoop Based BigData Application

Burhan Ul Islam Khan[1], Rashidah F. Olanrewaju[2]
[1,2]Department of Electrical and Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia

Due to increase in data load on cloud infrastructure the maintenance of quality services for BigData applications has become a major issue [1]. The data storage of structured as well as unstructured kinds are also a big concern to be considered while operating with certain search engine or data exploring applications[2]. On the other hand data intensive computation for cloud services is also increasing with vast pace. The predominant requirement for quality services in cloud infrastructure are optimized resource utilization, efficient processing in terms of execution time and data portability [1]. To develop such a cloud framework a number of researches have been done and still going on. Hadoop framework is one of the most successful cloud frameworks for cloud applications. This is the matter of fact that Hadoop is a potential candidate to operate with hundreds of Petabytes of data on cloud, but with increase in further loads, it still needs certain optimization in terms of execution speed and fair load balancing or job scheduling across encompassing nodes in cloud [3]. Hadoop encompasses two components, Hadoop distribution file system (HDFS) and MapReduce. HDFS represents an interface of nodes, task and user assigned jobs. But in fact a revolutionary system optimization can be carried in terms of MapReduce only [4]. HDFS possesses little scope of enhancements. Dealing with unstructured data and data intensive applications, data locality and its mining becomes a very difficult scenario for cloud infrastructure [5]. Therefore, generic load balancing and scheduling cannot be advocated for such huge data processing applications. Even individual enhancements of MapReduce which itself is a combination of Mapping, Sampling and Reducing, might cause higher and of course unwanted overheads on cloud, resulting into QoS degradation. MapReduce can provide certain scope for further enhancement [6]. In Hadoop based cloud compute system, the MapReduce framework functions for collecting data from various clusters or nodes and then processing for Map process which is followed by Reduce phase. Thus in between these two phases along with the resource retrieval and process of resource allocation at proper cluster nodes might be a huge fraction that can be further enhanced to yield better results. The predominant issue with MapReduce is that this framework is fundamentally batch-processing oriented and whenever processing is initialized, it’s updation for input data cannot be done with the expectation of similar output [6]. This is the main reason that makes this framework function poor in case of its real time application. Similarly, the data collection, process for Mapping and the shuffling which is in later stage converted to the intermediate data is a time consuming process, which is required to be optimized. Again, the reduce process of the intermediate data with the key based approach is also a tedious task compared to the conventional table driven or SQL based approach, as the data categorization on the basis of certain rank is a mammoth task. The replacement or allocation of these data efficient nodes is a great deal to be taken care of. So, considering these all aspects of functional MapReduce technique, it can be found that there is a huge space for algorithmic optimization for MapReduce. Therefore a scheme must be developed that could effectively eliminate the issue of resource utilization and latency in cloud infrastructure. The optimization with intermediate storage in MapReduce can give improved results and the selection of right data location for specific data type can reduce the computational complexity and execution time can be enhanced. Thus scheduling of MapReduce components with data location and availability of sensitive mode can be much fruitful. It can further be optimized if along with MapReduce job assignments a fair load balancing across comprising nodes is formed. On the other hand, because of the dynamic characteristics of cloud and its heterogeneous behaviour existing between the central servers and storing disks there must be something like a parallel architecture that could enhance the processing speed and data retrieval rate in MapReduce. Similarly, the load must be distributed uniformly across the network, so that the situation of uneven data distribution can be eliminated [7].

This is pragmatic view of researchers that optimization of MapReduce components (Map, Sample, Reduce) can bring revolutionary enhancements in QoS assurance for Hadoop model which can further optimize performance for BigData applications. Thus, taking into account of factors such as job scheduling and fare scheduling for reduce disks, can be considered for the possible solution [8]. The job scheduling [9] in reduce components with multilayered architecture be more significant, as it would not only provide second layer to employ the outputs generated by first layer MapReduce, but it would also reduce the execution time that in general takes place for exploring and storing big data sets. In case of unstructured data, this extraction issue also becomes worst [5]. So, if the scheduling is done in two consecutive steps, where initially sample MapReduce is performed which would be followed by final MapReduce can give results in minimum time. Even the consideration of Multiprocessor based parallelized scheduling for job assignments across virtual nodes in MapReduce can decrease execution time. The longest processing time (LPT) heuristics can play significant role in accomplishing multiprocessor based scheduling and system optimization [11]. Thus this approach can effectively reduce the overall execution time in cloud applications and especially it can be the precious gift for BigData applications.

On the other hand, in cloud infrastructures there are general occurrences of inter node fluctuations in input and output performance [12]. Therefore it is required to develop such a scheduling scheme for load balancing that can effectively reduce these fluctuations by reducing communication overheads, iterative functional use to reduce computation etc. It is also important that the scheme ought to reduce the probability of single point failure in cloud infrastructure. Similarly the computation should be scalable to reduce overheads. The reduction in overheads caused due to inter-process communication [13] can make the system operational with minimum overheads. Thus keeping these all needs into consciousness there is a call for a novel load balancing or load distribution scheme to be developed for inter-node load balancing in cloud infrastructure. An iterative programming scheme ought to be proposed that can effectively reduce computational cost even with higher performance. A multilayered scheduling for MapReduce, where in first case the cache based data samples are generated that would be helpful for final MapReduce process, and therefore computational overheads can also be removed.

A number of researches have been done for load distribution and Hadoop optimization, but unfortunately, majority of existing approaches considers a single point optimization [16][17][8], that is not sufficient for overall QoS optimization in BigData applications. Some works advocate for capacity scheduling [14] then some emphasize on issues of load distribution [15] among cloud nodes. For MapReduce applications, major works have emphasized on wither Mapping or sampling of data with key, value generation [18]. However if this optimization is supplemented with certain multiprocessor based scheme with optimization for load distribution across comprising nodes, then the overall performance of Hadoop can be obtained.

As of now the Hadoop considers a homogenous network but in fact the BigData has to be operational with heterogeneous kind of network also. The final solution should consider the locality of data obtained by implementing multilayered scheduling, where the first layer process MapReduce with sample data which if further processed for final MapReduce function. As an alternative it can also be considered hypothetically that most Mapping tasks might access the local data abruptly. In existing approaches the data movement has also not taken seriously that causes latency and delay in performance [19] [20]. Such ignorance in the existing approaches causes the reduction in overall quality of service of the cloud network. The emphasis has to be done on uniform data distribution across the comprising nodes in the network. In case of data placement consideration, certain pre-fetching scheme can be considered with predictive scheduling approach, which can efficiently assist Hadoop model in loading data files from remote or even local server to the main memory bank.

For a competitive scenario in multiple VMs communication, congestion could raise, so for eliminating such problems, incorporating pre-shuffling approach in scheduling itself can effectively exhibits processing on intermediate data files existing between Map and Reduction/ Reduce phase. This would enhance the overall throughput. In the scheduling approach researchers should try to incorporate the strengths of algorithms like pre-shuffling, pre-processing, load distribution, pre-fetching approaches etc., so that the overall balance of system functionality can be obtained and eventually the performance of the system can be enhanced. A sub-component for scheduling would have to be developed while taking into account of load balancing between Reduce tasks. Researchers need to keep in mind that the input to the Reduce Job/tasks is not known till the entire Mapping task has been done. And thus the roles of the Reduce component are issued that results into certain imbalance in load distribution between numerous tasks, therefore this is required to be considered while developing scheduling algorithm. Furthermore for realization of the optimum load distribution, consideration of a certain network monitoring module is must so that the real performance of

Source-to-Mapper link might be retrieved and can be conveyed to scheduler. It would make the system functional better even with minimum overheads.

The predominant philosophy behind Hadoop optimization is that the optimization of MapReduce which is a dominant programming platform that can bring many functional enhancements as per scheduling algorithms developed and implemented. On the other hand, the implementation of multilayered scheduling will facilitate second scheduler of MapReduce to use the details extracted from first layer of MapReduce, thus the overall execution time in exploring entire datasets would be saved, and the Map-Reduce function already done in layer one implementation would be helpful for second layer of Map-reduce. The parallelized scheduling may cause the reduction in unwanted overheads and the system functionality would be enhanced in terms of execution process and optimum resource optimization. The implementation of predictive kind of load scheduling can utilize experiences to remove or eliminate the extreme or bursty conditions in cloud infrastructure as it would be employing experiences for decision making. On the other hand the integration of multiple scheduling for load distribution as well as MapReduce's reducer scheduling can not only enhance resource utilization but will also reduce delay cost that would optimize QoS delivery for BigData applications.

REFRENCES

[1] Demirkan, Haluk, and DursunDelen. "Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud."*Decision Support Systems* 55, no. 1 (2013): 412-421.

[2] Herodotou, Herodotos, Harold Lim, Gang Luo, NedyalkoBorisov, Liang Dong, FatmaBilgen Cetin, and ShivnathBabu. "Starfish: A Self-tuning System for Big Data Analytics." In *CIDR*, vol. 11, pp. 261-272. 2011.

[3] Prekopcsák, Zoltán, Gabor Makrai, TamasHenk, and Csaba Gaspar-Papanek. "Radoop: Analyzing big data with rapidminer and hadoop." In *Proceedings of the 2nd RapidMiner Community Meeting and Conference (RCOMM 2011)*, pp. 1-12. 2011.

[4] Dittrich, Jens, and Jorge-ArnulfoQuiané-Ruiz. "Efficient big data processing in Hadoop MapReduce."*Proceedings of the VLDB Endowment* 5, no. 12 (2012): 2014-2015.

[5] Abadi, Daniel J. "Data Management in the Cloud: Limitations and Opportunities." *IEEE Data Eng. Bull.* 32, no. 1 (2009): 3-12.

[6] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." *Communications of the ACM* 51, no. 1 (2008): 107-113.

[7] Kolb, Lars, Andreas Thor, and Erhard Rahm. "Load balancing for mapreduce-based entity resolution." In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pp. 618-629. IEEE, 2012.

[8] Zaharia, Matei, DhrubaBorthakur, J. SenSarma, KhaledElmeleegy, Scott Shenker, and Ion Stoica. "Job scheduling for multi-user mapreduce clusters."*EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-55* (2009).

[9] Condie, Tyson, Neil Conway, Peter Alvaro, Joseph M. Hellerstein, KhaledElmeleegy, and Russell Sears. "MapReduce Online."In *NSDI*, vol. 10, no. 4, p. 20. 2010.

[10] Vieira, Kleber, AlexandreSchulter, Carlos Westphall, and Carla MerkleWestphall. "Intrusion detection for grid and cloud computing." *It Professional* 12, no. 4 (2010): 38-43.

[11] Jiang, Benli, and Jianjun Wu. "Research on Data Block Distribution Optimization on Cloud Storage."In *Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 3*, pp. 733-738.Springer Berlin Heidelberg, 2014.

[12] Gunarathne, Thilina, Tak-Lon Wu, Judy Qiu, and Geoffrey Fox. "MapReduce in the Clouds for Science."In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pp. 565-572.IEEE, 2010.

[13] Rabl, Tilmann, Michael Frank, HatemMoussellySergieh, and HaraldKosch. "A data generator for cloud-scale benchmarking."In *Performance Evaluation, Measurement and Characterization of Complex Systems*, pp. 41-56.Springer Berlin Heidelberg, 2011.

[14] Sandholm, Thomas, and Kevin Lai. "Dynamic proportional share scheduling in hadoop."In *Job scheduling strategies for parallel processing*, pp. 110-131.Springer Berlin Heidelberg, 2010.

[15] Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services." In *Algorithms and architectures for parallel processing*, pp. 13-31.Springer Berlin Heidelberg, 2010.

[16] Yan, Jinshuang, Xiaoliang Yang, RongGu, Chunfeng Yuan, and Yihua Huang. "Performance Optimization for Short

MapReduce Job Execution in Hadoop." In *Cloud and Green Computing (CGC), 2012 Second International Conference on,* pp. 688-694. IEEE, 2012.

[17] Raj, Aparna, KamaldeepKaur, UddipanDutta, V. VenkatSandeep, and ShrishaRao. "Enhancement of Hadoop Clusters with Virtualization Using the Capacity Scheduler."In *Services in Emerging Markets (ICSEM), 2012 Third International Conference on*, pp. 50-57.IEEE, 2012.

[18] Bhattacharjee, Ratnadeep. "An analysis of the cloud computing platform."PhD diss., Massachusetts Institute of Technology, 2009.

[20] Kurazumi, Shiori, Tomoaki Tsumura, Shoichi Saito, and Hiroshi Matsuo. "Dynamic processing slots scheduling for I/O intensive jobs of Hadoop MapReduce." In *Networking and Computing (ICNC), 2012 Third International Conference on*, pp. 288-292. IEEE, 2012.

[21] Yu, Xiao, and Bo Hong. "Bi-Hadoop: Extending Hadoop To Improve Support For Binary Input Applications." In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pp. 245-252. IEEE, 2013.