A Technical Survey on Optimization of Processing GeoDistributed Data

T YJ Naga Malleswari¹, S.Ushasukhanya², A.Nithyakalyani³, Girija.S⁴

Department of Computer Science and Engineering, SRM University, Chennai
nagamalleswari.t@ktr.srmuniv.ac.in, ushasukhaya.s@ktr.srmuniv.ac.in,
nithyakalyani.a@ktr.srmuniv.ac.in, girija.s@ktr.srmuniv.ac.in

Abstract. With growing cloud services and technology, there is growth in some geographically distributed data centers to store large amounts of data. Analysis of geo-distributed data is required in various services for data processing, storage of essential information, etc., processing this geo-distributed data and performing analytics on this data is a challenging task. The distributed data processing is accompanied by issues in storage, computation and communication. The key issues to be dealt with are time efficiency, cost minimization, utility maximization. This paper describes various optimization methods like end-to-end multiphase, G-MR, etc., using the techniques like Map-Reduce, CDS (Community Detection based Scheduling), ROUT, Workload-Aware Scheduling, SAGE, AMP (Ant Colony Optimization) to handle these issues. In this paper various optimization methods and techniques used are analyzed. It has been observed that end-to-end multiphase achieves time efficiency; Cost minimization concentrates to achieve Quality of Service, Computation and reduction of Communication cost. SAGE achieves performance improvisation in processing geo-distributed data sets.

1 Introduction

In recent years, there is a tremendous growth in Cloud Computing[1], and it has become an important part of our life. It provides services in various fields like astronomy, physics, healthcare, commerce, social networking, web service providers, internet service providers, content delivery networks and various other application domains. The increase of usage of these services, there is a rise in need towards geo-distributed networks. In a case of many new applications and environments, the data sources are widely distributed. Some companies, businesses produce data at various sites situated in various locations. With the increasing users, applications, sensors and devices, to ensure a quality of service, there is a need for analysing a large amount of data produced all over the world. Apart from the data generated being distributed, the resources required for computation which in turn are needed for carrying out the analysis of data can also be distributed. To develop new products and services, improve decisionmaking and risk minimization, this analysed data has shown its worth and importance in discovering the valuable understanding ofthis data. The key objective of our work is to efficiently execute the computations of the distributed data in distributed environments which are an ever-growing need. For efficient processing of this distributed data, the first important question is where the computation is to be carried out. Thus various methods are proposed and are being developed to achieve this efficiency by minimizing the effect of various factors on processing the distributed data. In this paper Section II explains the basic ways of processing geo-distributed data. The detailed description of various techniques is given in Section 3. Section 4 shows the comparison of various
mentioned techniques. In Section 5, the findings from the comparison given in the previous section are discussed. In Section 6, the conclusion is given based on the survey.

2 Processing distributed data
There are two ways in which the data can be processed. Jim Gray has proposed, “one can either move the questions or the data” [2]. The main feature is the tight coupling between the data and computation. The computation tasks cannot be conducted unless the necessary data is available at the required time. Therefore, task assignment [3], data placement and data movement are very important to efficiently process distributed data.

2.1 Moving Data
In this method, the data is collected from various diverse origins. This data is then sent to a centralized data center where the computation tasks are carried out. The problems that incur in this method are it may lead to excessive latency [4],[5], leads to high bandwidth costs and it may not be able to accommodate all the data which is collected at a single data center for computation. The advantage of this centralized approach is that it is less fault tolerant whereas the disadvantage is the failure of centralized data center might lead to the complete shutdown.

2.2 Moving Computation
In this method, we perform computation task at each data center with the input data. The results of these sub-computations contain intermediate data which has to be further processed to produce the final results. If the amount of the intermediate data generated is large, then processing this data becomes an extremely costly affair than moving the data to a centralized data center. The other problems incurred in this method are the different data centers might have different compute capacities, this can lead to imbalanced resource utilization and execution of tasks.

3 Various Techniques

3.1 MapReduce
MapReduce is being extensively used today for various data analysis applications. It is a well-known programming model to process a large amount of data on data center-scale clusters. The scenario where the data and compute resources are present in a single central location is no longer applicable for many applications in various fields like commercial, scientific and social networking domains, where the data generated is geographically distributed. MapReduce[6][7] is effectively used by a wide range of data-intensive applications, but the limitation is that it was developed to run on single-site homogeneous clusters. The research results have shown that the implementation of the HadoopMapReduce[8],[9] in geo-distributed environments where there is high network heterogeneity, the performance levels drop down rapidly. The main reason for this drop in performance is the heavy inter-dependency across different phases of MapReduce. As the tasks are usually allocated to nodes which already host their input data in MapReduce, the data placement and task execution are interlinked with each other strongly. Therefore, the selecting of the systems to which inputs have to be sent effects both the amount of time taken to send data and also the nodes where the intermediate data is generated. This also effects some of the factors influencing the data shuffle to the reducers. This problem severity increases even more in a case of the distributed networks and environments, considering the fact that the heterogeneity of computation capacities of a node and link capacities is huge.

One of the powerful tools of MapReduce implementation is the abstraction. Therefore, the objective of implementing MapReduce in geo-distributed environments is appreciable despite the various difficulties in its implementation. MapReduce is a very popular technique which is used by different data-intensive computing applications. Hence there are many data scientists who have expertise in MapReduce application development. Therefore, further research in this area can lead to innovations into using MapReduce in geographically distributed environments.
3.2 End-to-End Multiphase
In this end-to-end multiphase optimization[10], the dependency of different phases of MapReduce is considered to overcome the various problems incurred in traditional MapReduce implementations. Considering both the execution cost of an individual task or computational phase and its impact on the performance of upcoming phases, we develop a new method. This method has two techniques. One technique is Map-Aware Push, and the other technique is Shuffle Aware Push.

3.2.1 Map-Aware Push
The Map-Aware Push technique optimizes the push and map phases in MapReduce. The traditional MapReduce follows the push-then map approach. In this approach, all the input data required is collected, and the computation phase starts only when the entire data collection finishes. This approach has two important demerits. First, all the computation tasks that are ready to execute have to wait for the completion of data push to the slowest communication link which introduces wastage of time. Second, map tasks are deprived of a way to demand the amount of work based on their compute capacities if both the push and map phases are separated. Scheduling of the push phase in a map aware manner becomes more difficult due to these issues.

In our technique, we overcome these issues by letting the push phase aware of the map phase. The execution cost of each map phase is made known to the data push based on the source-to-mapper link capacities as well as mapper node computation speeds. In our technique, both the push phase and map phase are overlapped. There are two advantages in this map aware push technique. The first advantage is it reduces the time delay caused due to data push when not overlapped with the map execution. The second advantage is a dynamic awareness between both the phases. Therefore, the mapper nodes with higher computation speeds and faster links can process more amount of data.

3.2.2 Shuffle Aware Map
The map phase can influence the push phase, regarding the volume of data each mapper receives as well as the sources from which each mapper receives its data. In turn, the push determines, in part, when a map slot becomes available for a mapper. Thus, from the perspective of the push and map phases, a set of mappers and their data sources are decided. In MapReduce, the shuffling of intermediate data from mappers to reducers is an all-to-all operation. In a distributed environment, a mapper with a slow outgoing link can, therefore, become a bottleneck in the shuffle phase, slowing down the downstream reducers. We propose map task scheduling based on the estimated shuffle cost from each mapper to enable faster shuffle and reduce.

3.3 CDS (Community Detection Based Scheduling)
One of the major problems in analyzing the data faster in geo-distributed environments is the limited bandwidth availability. To overcome this issue, a scheduling algorithm which minimizes the total completion time is needed. This minimization of time can be achieved by optimizing task scheduling and data transfer over the distributed network. In the proposed method, the task scheduling is considered as a community detection problem. The Community Detection-based Scheduling (CDS) algorithm [11], will be able to minimize the data transfer volume by considering the dependencies among the tasks, data placement, and data centers. Identification of communities and the boundaries of the communities from a group of points based on the structural position of these points within the network is termed as community detection. Grouping of all the tasks requiring same input files as one community is done and the entire community is scheduled to the datacenter containing the maximum input files. In a datacenter, all the corresponding files are arranged as communities, and initially, there are no connections between the communities. The process of placing the tasks into the communities continues throughout the scheduling. This maximizes the measurement of modularity. This process places all the nodes which are connected closely in the same communities. The scheduling reduces the completion time to an extent where it is at least same as the greatest processing time among all the tasks. As the time taken for processing is fixed, the time taken to transfer files could vary highly. This technique implies that the tasks which have long processing times must be provided with more
bandwidth so that its file transfer time reduces. In the case of many data-intensive and compute intensive applications, CDS algorithm minimizes the file transfer volume.

3.4 G-MR System
To process geo-distributed one might have to execute some MapReduce jobs on some geo-distributed data sets. For the optimization of this data processing, we need to analyze numerous possibilities of executing these jobs and discover how the data can be transformed. This knowledge can be used to find how the jobs can be scheduled which optimizes either the total execution time or total cost. G-MR is a system proposed for executing these job sequences, which optimizes the processing of distributed data. G-MR [12] is a Hadoop-based framework which can execute some sequences of MapReduce jobs on distributed data sets placed across multiple data centers. The proposed G-MR system is similar to the atmosphere surrounding the clouds. It is different from the present frameworks that operate only on single data centers. G-MR system uses an ideal algorithm named data transformation graph (DTG) algorithm. Considering the characteristics of the data sets, the infrastructure of the data center and MapReduce jobs, this algorithm detects an execution path for performing a sequence of MapReduce jobs which is optimal. DTG algorithm can optimize either the execution time or the cost. The execution path determined by the application of DTG algorithm in G-MR system need not be the optimum possible execution path. It is very difficult to determine the execution path which is optimum for a large data sets because each and every possible data move need to be considered for it. The benefit with G-MR system is that it provides an optimized execution path that performs better than many of the commonly used execution paths. This execution path determined by G-MR system can be enforced into the geo-distributed environment where the MapReduce executions can be performed in respective data centers.

3.5 Rout
To perform data processing tasks across various datacenters, different geo-distributed data structures and operations are introduced by Rout technique. Cloud computing serves as one of the solutions to tackle the data challenges. To implement a cloud, we require datacenters that are geographically distributed because a single datacenter will not be able to host all the data in the world reliably and efficiently. Therefore in this technique, we are proposing Rout [13], a data processing language which can be used to express programs with datasets from various datacenters and also individual datasets that are partitioned across the datacenters. Experimental results have shown that the execution time to process geo-distributed data has reduced to half in the case of Rout as compared to other scheduling techniques where all the data is consolidated first. One more feature of Rout technique is that only a small portion of geo-distribution is visible to programmers.

SAGE
A cloud architecture is proposed for performing data analysis in this approach. This is implemented by composing services which are distributed among various cloud data centers. Exploiting the geographical data locality, the computation tasks are distributed across multiple data centers. For efficiently processing geographically distributed data in the cloud, SAGE[14] proposes the effective solution. This technique proposed a service-oriented architecture, where the tasks are distributed among multiple cloud data centers exploiting the data locality. To aggregate the global results, data-driven services are composed in this technique.

AMP(Ant Colony Optimization)
In this technique[15], the Ant-Colony Optimization (AMP) is used to propose a customer satisfaction-aware algorithm for geo-distributed data centers. In this model, a customer satisfaction model is introduced. To ensure customer satisfaction[16], we consider utility (or net profit) maximization as an issue and optimize this problem. In this method, the net profit of the service providers is maximized by sending service requests to appropriate data centers, and the right amount of systems are generated to provide customer satisfaction.
In this model, we take into account the two parameters customer service price and service time to ensure customer satisfaction. Based on the complexity of the problem, we formulate the problem as a binary integer programming problem. The Ant-Colony Optimization used in this algorithm helps to achieve net profit maximization while considering the capacity constraint of various data centers. The results of various experiments show that there is a significant improvement in net profit ensuring customer satisfaction by dispatching requests to data centers.

**Cost Minimization**

In this technique [17], we are motivated to study the cost minimization problem via a joint optimization of three factors, i.e., task assignment, data placement and data movement for big data services in geo-distributed data centers. In general, the task completion time is based on the time taken for both data transmission and computation tasks. In this method, we are using a two-dimensional Markov chain to derive the average task completion time. This method also models the whole problem as a mixed-integer non-linear programming (MINLP) to linearize it and achieve an efficient solution. This is further linearized into an MILP problem to tackle the high computational complexity of solving our MINLP. Various experimental results have shown that the proposed optimization solution has the substantial advantages over various optimization solutions.

4 Comparison

The comparison of various techniques used for optimization of geo distributed data processing is discussed in Table 1 given below.

| S.No | Optimization Technique Used | Year | Used & Proposed Methods | Issues Addressed | Findings and Results |
|------|-----------------------------|------|-------------------------|------------------|---------------------|
| 1    | MapReduce                   | 2004 | Hadoop Map Reduce       | Performance and time delay | Remarkably effective in single site homogenous clusters. |
|      |                             |      | Implementation on geo-  |                  | Reduction in overall runtime by 7%-18% considering the | |
|      |                             |      | Map-Aware Push and      |                  | Reduction in total completion time and total data transfer | |
|      |                             |      | Shuffle-Aware Map       |                  | volume                                                                 | |
| 2    | End-to-end Multiphase       | 2013 | Community Detection     | Fast data analysis, WAN bandwidth | Significant Improvement in processing time and cost on geo-distributed data sets. |
|      |                             |      | based scheduling        |                  |                                                       | |
| 3    | CDS                         | 2016 | DTG(Data Transformation | Execution time and monetary cost for Map Reduce job implementations | Two times faster response times at distributed data centers. |
|      |                             |      | Graph) Algorithm        |                  |                                                       | |
| 4    | G-MRSystem                  | 2014 | Geo-Distributed Data    | Response times at distributed data centers. | Two times faster response times at distributed data centers. |
| 5    | ROUT                        | 2013 | Structures and Operations. |                  |                                                       |
5 Findings
The optimization technique is chosen based on the issue to be addressed from the comparisons made in section IV. For performance improvement, MapReduce technique proves to be remarkably effective in case of homogenous clusters, while SAGE technique is effective in case of highly distributed multi clusters improving performance by a factor of 3.3. For time efficiency end-to-end multiphase technique efficiently reduces runtime by 7% to 18%. CDS technique reduces total completion time by 35% along with the data transfer volume by 40% compared to other scheduling algorithms in this scenario. For both time and cost optimization G-MR system technique provides significant improvement. ROUT technique optimizes time by reduction in time responses by almost two times. Ant Colony Optimization provides with net profit improvement by utility maximization. Cost Minimization technique assures quality of service and cost minimization by optimized placement of data, data movement and task movement. This knowledge can be used to further develop new techniques as future work.

6 Conclusion
A large number of data-intensive applications are highly distributed because their input data is stored and generated at various geographical locations. Processing data in such geo-distributed environments is a challenging task as it is to be dealt with some issues. Thus for optimization of processing the geo-distributed data, many techniques dealing with various issues are being developed. Some techniques deal with the reduction of time, some other with cost optimization while some deal with the overall optimization is ensuring the quality of service and customer satisfaction. The survey of various techniques and the issues dealt with them and the extent to which they optimize the factor about the issues are proposed in this paper. It is concluded from section IV and V that the best optimization technique is chosen based on scenario and constraints.

References
[1] T.Y.J.Naga Malleswari, D.Rajeswari, Dr.V. Jawahar Senthil Kumar, “A Survey of Cloud Computing, Architecture & Services Provided by Various Cloud Service Providers,” in Proceedings of International Conference on Demand Computing, 978-93-5087-502-5 201, Bangalore, 2012
[2] T. Hey, S. Tansley, and K. Tolle, Eds., The Fourth Paradigm: Data Intensive Scientific Discovery. Redmond, Washington: Microsoft Research,2009.http://scholar.google.co.in/scholar?hl=en&q=Live+Virtual+Machine+Migration+Techniques+A+Technical+Survey+&btnG=.
[3] L. Rao, X. Liu, L. Xie, and W. Liu, “Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a MultiElectricity-Market Environment,” in Proceedings of the 29th International Conference on Computer Communications (INFOCOM). IEEE, 2010, pp. 1–9. Qinlu He, Zhanhuai Li, Xiao Zhang, “Data Deduplication Techniques,” in International Conference on Future Information Technology and Management Engineering, China, 2010, pp.43-433.
[4] “A conversation with Jim Gray,” Queue, vol. 1, no. 4, pp. 8–17, Jun. 2003. [Online]. Available: http://doi.acm.org/10.1145/864056.864078

[5] A. Rabkin, M. Arye, S. Sen, V. Pai, and M. J. Freedman, “Making every bit count in wide-area analytics,” in Proc. of HotOS, 2013.

[6] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,” in Proc. of OSDI, 2004, pp. 137–149.

[7] TYJ NagaMalleswari, G. Vadivu, “MapReduce: A Technical Review”, Indian Journal of Science and Technology, 10.17485/ijst/2016/v9i1/78964, Vol.9 Issue1, 2016.

[8] M. Cardosa, C. Wang, A. Nangia, A. Chandra, and J. Weissman, “Exploring MapReduce efficiency with highly-distributed data,” in Proceedings of MapReduce, 2011, pp. 27–33.

[9] “Hadoop,” http://hadoop.apache.org.

[10] "End-to-end Optimization for Geo-Distributed MapReduce” Benjamin Heintz, Student Member, IEEE, Abhishek Chandra, Member, IEEE, Ramesh K. Sitaraman, Member, IEEE, Jon Weissman, Senior Member, IEEE, IEEE Transactions on Cloud Computing, 2013 IEEE

[11] “Fast Big Data Analysis in Geo-Distributed Cloud” Yue Li*, Laiping Zhao†§, Chenzhou Cui‡, Ce Yu∗∗School of Computer Science and Technology, Tianjin University, Tianjin, China †School of Computer Software, Tianjin University, Tianjin, China {liyue, laiping, yuce}@tju.edu.cn ‡National Astronomical Observatories, CAS (NAOC), Beijing, China, 2016 IEEE International Conference on Cluster Computing.

[12] “From the Cloud to the Atmosphere: Running MapReduce across Data Centers” Chamikara Jayalath, Julian Stephen, and Patrick Eugster, IEEE Transactions On Computers, VOL. 63, NO. 1, January 2014.

[13] L. Chen, C. Wang, B.B. Zhou, L. Sun, Y.C. Lee and A.Y. Zomaya. Tradeoffs between Profit and Customer Satisfaction for Service Provisioning in the Cloud. In: Proc. of Symposium on High-Performance Parallel and Distributed Computing (HPDC), San Jose, California, USA, ACM, 2011: 229-238.