ISSN:0975-9646

Minimizing Cost for big data processing using Online Database

Shruthy Y[1], Sreenimol K. R[2]

[1] *PG Scholar,* [2]*Associate Professor,*

Department of Computer Science and Engineering, MG University
Mangalam College of Engineering, Kottayam, Kerala-India

***Abstract*— the growth in the big data is found to be at a very high pace each day. This heavy demand to meet the constraints like storage, computation and communication in the data centers incurs higher expenditures in order to meet their needs. Hence, cost minimization is a very essential and important fact in this era of big data. The operational expenditure of the data centers is deeply influenced by two main factors i.e. task assignment, data placement. In this paper, we have minimized the cost by the joint optimization of these two factors by applying the Mixed Integer Linear Programming that reduced the average completion time and also implemented an online database which improved the efficiency when compared to the traditional database and provided various advantages like processing speed, security etc. The experimental studies have proved on the fact of the cost reduction in case of using the online database. In this paper we have considered the "cyberstudents" as our online database.**

***Keywords*— Big data, cost minimization, data placement, MILP, online database, task assignment.**

I. INTRODUCTION

The big data explosion arising these days needs modern techniques and methods to handle such large amount of data. An example for such is Google whose 13 data centres are being distributed geographically over 8 countries in 4 continents [2]. The analysis of big data has lead to unearthing of valuable data and thus improved on the factors like decision-making, minimizing the risk and developing new products and services.

The existing works to reduce the computation or the communication cost of data centres include Data centre resizing (DCR) that reduces the computation cost by adjusting the number of activated servers by data placement [3]. Some other techniques include the big data service frameworks like MapReduce [4] which comprises of a distributed file system underneath that distributes the data chunks and their replicas across the data centres for fine-grained load balancing and also high parallel data access performance. The above methods have obtained the results but still there's lot of scope for the cost effective big data processing because of the following weaknesses:

First, data locality may result in the wastage of resources. For example those computation resources of a server that have less data in it will remain idle. This low resource utility further leads more number of servers to be activated and hence it leads to higher operating cost.

Second, the links in the network vary on the transmission rate and costs according to their unique features [5] for e.g. the distances and physical optical fibre facilities between data centres. Due to storage and computation capacity constraints, all the tasks cannot be placed on the same server, on which their corresponding data resides hence here the transmission cost (e.g. energy) is nearly proportional to the number of links used while satisfying all the transmission requirements.

Third, the Quality-of-Service (QoS) of the data tasks is not being considered in the existing work. The big data applications exhibit Service-Level-Agreement (SLA) between service provider and the requesters. To observe this, a certain level of QoS is guaranteed and any cloud computing task's QoS is firstly determined by where they are placed and how many computation resources are allocated to them. Like in [3], the general cloud computing task mainly focuses on computation capacity constraints by ignoring their transmission rates.

Fourth, the storage of the data in the traditional databases includes the storage in disks. This increases the time required to fetch and store the necessary information's for processing.

To overcome the above weaknesses, [1] combined the two big data processing factors i.e. data placement and task assignment by introducing MILP (Mixed Integer Linear Programming). Here, we brought into effect the concept of online databases so as to improve the processing speed, provide security etc.

II. RELATED WORK

A. *Server Cost Minimization*

According to Quereshi et al [6], a data centre consists of very large number of servers that consumes megawatts of power. Million dollars are spent on electricity that has posed a big burden to the service providers. Hence reducing the electricity cost has received significance in the field of both industry and academia [6], [7], [8].

DCR and data placements are those techniques that have attracted lots of attention. From then there has been considerable amount of works being carried on to minimize the electricity cost in the server. Rao et al [3] investigated how to reduce electricity cost by routing user requests to geo-distributed data centres with updated sizes that matched the user requests.

B. *Big Data Management*

Many proposals have been proposed to tackle the challenges of effectively managing the big data so as to improve the storage and computation cost.

Sathiamoorthy et al. [9] presented a novel family of erasure codes that are efficiently repairable and also offer higher reliability when compared to the reed Solomon codes. Cohen et al. [10] presented a new design philosophy, techniques and experience providing magnetic, agile and deep data analytics to decide how to allocate the computation resources to the tasks. Chen et al. [11] proposed a practical method to combat the high scheduling complexity of the nodes in the system. He jointly scheduled all three phases i.e., map, shuffle and reduce of the MapReduce process.

C. *Data Placement*

Shachnai et al. [12] investigated on the placement of Video-on-Demand (VoD) file copies on the servers and the amount of load capacity assigned to each file copy to minimize the communication cost while ensuring the user experience. Agarwal et al. [13] came up with an automated data placement mechanism named “Volley”. The cloud services use volley by submitting logs of the datacenter request. Volley analyses these logs using an iterative optimization algorithm based on the data access patterns and client locations. Cidon et al. [14] proposed the idea of MinCopysets, which is a data replication technique that decouples data distribution and data replication to improve data durability in the distributed data centres

III. Online Databases

An online database is a database that is accessible from a network, including from the internet. It differs from a local database, held in an individual computer or its attached storage such as a CD. They differ from the typical traditional databases such as Oracle, Microsoft SQL Server, Sybase etc. These differences include:

1. These online databases are delivered primarily via a web browser
2. They can be purchased based on a monthly subscription
3. They embed common collaboration features such as sharing, email, notifications, etc.

The advantages of these databases can be expressed in different fields based on its various applications like those in administrative, searching and informational.

The administrative online databases are those that deal with the processing of the administration related information at a faster rate. The advantages of using online databases for the administrative purposes includes saving time by improving the processing speed, saves the database memory space by eliminating the redundant values, expenditure cost of building the architecture like that in the traditional databases is cut short. The administrator’s effort is being reduced for the easy and reliable processing of the data. This reduces the working hours of both the administrator and the clients working on the database. Thus increases the efficiency to a considerable amount. The examples include Oracle, SAP Hana etc.

The search related online databases also include the search engines for the effective searching of the user queries. They offer the term searching on different aspects like the search based on user fields, search related to controlled vocabulary and also indexes. These online services also offer the search commands for various purposes like for those in creating the search sets as per user requirements, Boolean operations for just a yes or no result, search related commands for word searching, and also there are commands for search limiting. Such commands provide greater precision for the effective searching purposes.

There are informational advantages too concerned with the online databases. Different kinds of searches are possible to be performed qualitatively , easier access to the paper based on the quality of the information found, processing of the search results by the techniques like sorting, ranking, export etc., highly focused information that is actionable.

We use online databases for complex searches with many search terms and many search criteria’s, for the topics that require comprehensive coverage like the historical searches, all-files searches, cover the topics that are not approachable through a print index that includes the print indexes that are hierarchical and ordered, questions for which the manual search would be time consuming. The questions like cross-disciplinary, multi-file searches, and repeated searches.

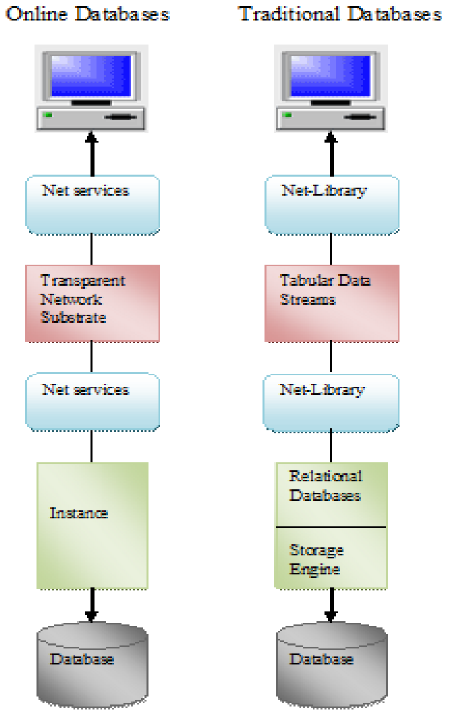

Fig.1: Comparison between online and traditional databases

Fig. 1 gives an idea about the main difference between traditional and online databases. The web interface provided by the online databases accounts for a big advantage of using it rather than the traditional ones.

IV. Architecture

A. Network model

The model consists of all the servers of the same data centre (DC) that are connected to the local switches, and these data centres are connected through switches as shown in Fig 2. There are a set of D datacenters and each datacenter *d* € D consists of a set S_i of servers that are connected to a switch

w_i € W with its transmission cost of $C_{L.}$ Here the transmission cost C_T for inter data centre traffic is greater than $C_{L,}$ i.e., $C_T > C_{L.}$ Here all the computation resource and storage capacity, both of which are normalized to a unit. We use S to denote the set of all servers i.e., $S = S_1 \cup S_2 \cup S_{3\ldots\ldots} \cup S_{|D|.}$

The whole system is modelled as a directed graph G = (N, E). The vertex N = W ∪ S (refer Table 1) which is set W of all switches and the S is the set of all servers, and E is the edge set. All the switches are connected to their local switch by intra-data centre links while the switches are connected by inter-data centre links, which is identified by their physical connection. The weight of each link $l^{(u,v)}$ represents the corresponding communication cost, can be defined as:

$$l^{(u,v)} = \begin{cases} C_T & \text{if } u, v \; € \; W \\ C_{L,} & \text{otherwise} \end{cases}$$

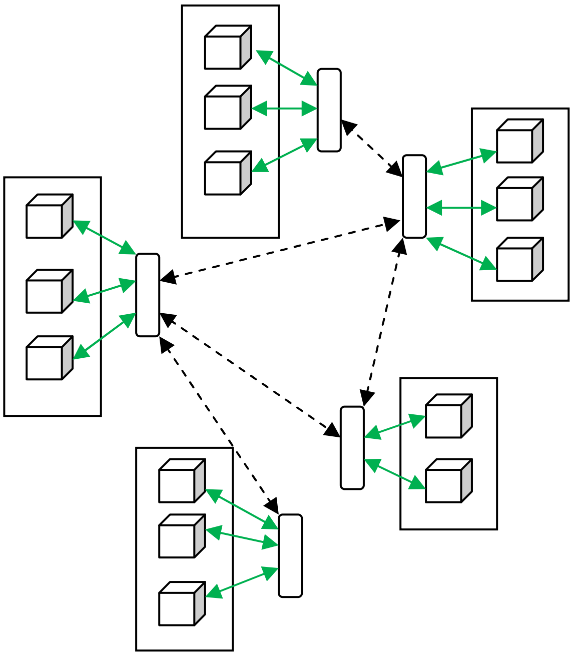

Fig. 2: Data centre topology

TABLE I
NOTATIONS DESCRIBED

Constants	Description
S_i	The set of servers in data center i
w_i	The switch in data center i
$l^{(u,v)}$	The weight of the link (u, v)
D	The set of data centers
H_i	The set of data chunks

The table notations give a brief description about the constants that are used during the processing in the network model.

B. Task model

The data stored in the distributed data centres are divided into a set *H* of data chunks. Each chunk *h* € *H* has the size of

Φ_h ($\Phi_h \leq 1$) which is normalized the server's storage capacity. The idea here is that there are exactly *P* numbers of copies stored in the distributed file system for managing the fault tolerance.

The tasks that arrive to the datacenters during a particular time period can be visualized as a Poisson process [14]. In particular, let λ_h be the average task arrival rate requesting chunk *h*. These tasks will be distributed to the servers with a fixed probability and hence the task arrival in each server can also be considered as a Poisson process. So, here the average arrival rate of the task for chunk *h* on server S λ_{sh} ($\lambda_{sh} \leq 1$).

V. Big Data Processing Constraints

The three constraints on big data processing that have been identified are task placement, data loading and QOS constraints. They have been defined as follows:

A. Constraints on Task Placement

Consider a binary variable y_{sh} defined to denote whether data chunk h is placed on server s as:

$$y_{sh} = \begin{cases} 1, \text{ if chunk h is placed on server S} \\ 0, \text{ otherwise;} \end{cases}$$

B. Constraints on remote data placement

When the server needs a chunk of data, internal or external data transmission can occur. The routing of these chunks is done by the switch connected to the data centres. The nodes in the graph can be categorized into source node, intermediate node and destination node.

Source nodes are nothing but the servers in which the data chunks are stored. Intermediate nodes receive the data from the source node and forwards it to the destination node based on the routing strategies. If the required data chunk is not present in the destination node, it must receive the data chunks in the same rate as the request rate for chunk on server and the CPU usage of the chunk on the server.

C. Constraints on remote data loading

Consider μ_{sh} and γ_{sh} as processing rate and loading rate for data chunks on the server respectively. The processing is then given by a two-Dimensional Markov Chain model, where each state (a, b) where *a* represents pending task and *b* represents the available task. Let θ_{sh} be amount of computation resources that the data chunk h occupies. The processing rate of the task proportional to processing resource usage and it is given by:

$$\mu_{sh} = \alpha_s . \theta_{sh}$$

VI. Performance Evaluation

The experimental setup includes the software's like Net Beans IDE 8.0, HeidiSQL, and JDBC ODBC connections. The processing speed based on the number of records being accessed in lesser time. The large amount of data is thus being stored into the online database rather than the traditional database. This shows considerable improvements in terms of time consumption for processing of this large data.

Fig. 3 shows the result after experimental evaluation. The x-axis gives the rate of arrival of tasks to the server in a given amount of time and the y-axis indicates the operation cost of each task that has arrived to the server. The graph gives the clear picture of the operation cost taken by the server depending on their arrival rates to the server.

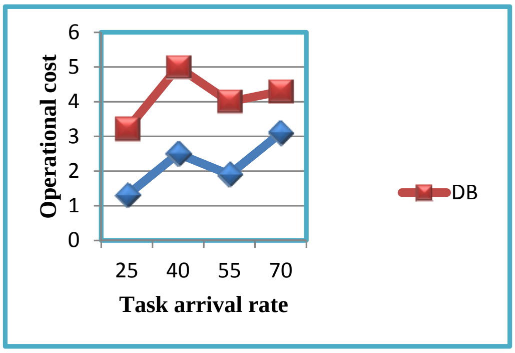

Fig. 3: Graphical representation comparing the two databases based on the rate of task arrivals and the operation cost for each task taken.

The next graph i.e. Fig. 4**,** gives the comparison between number of records accessed against the time or the speed taken for processing these records in milliseconds. The graph clearly shows how the online database is proved to be faster when compared to that of the traditional database.

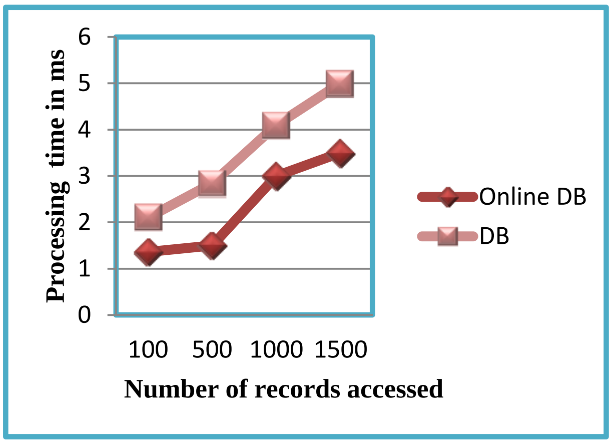

Fig. 4: Graphical representation comparing the number of records accessed against the time taken for processing in milliseconds.

VII. Conclusion

The paper minimizes the cost that occurs during the big data processing by considering two main big data service issues i.e., data placement and task assignment by proposing the MILP. We have also introduced the online databases database in the paper to overcome the drawbacks of using the traditional database. The example we have considered here is cyberstudents. Cyberstudents is one of the online database for maintaining the all the students information, generally used by an organization or an academic institution. Thus, through the experimental evaluation we have found that there is much increase in the processing speed of the big data by the implementation of the online databases as it provides additional advantages like security, less storage space, cost effective by not worrying on the infrastructure part etc. Thus, this is found to be of much use in case of big data processing. In the future works, we try to implement any of the In-memory databases for handling the time taken during the disk accesses.

References

[1] Lin Gu, Deze Zeng, Peng Li, and Song Guo "Cost minimization for big data processing in geo-distributed data centers" IEEE Transactions on emerging topics in computing, volume 2, no.3, September 2014.

[2] (2013) Data Center Locations [Online]. Available: http://www.google.com/about /datacenters/inside/locations/index.html.

[3] L. Rao, X. Liu, L. Xie, and W.Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *Proc 29th Int. Conf. Comput. Commn.* 2010, pp1-9.

[4] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commn. ACM,* vol. 51, no. 1, pp. 107-113, 2008.

[5] I. Marshall and C. Roadknight, "Linking cache performance to user behavior," *Comput. Netw. ISDN Syst.,* vol. 30, no. 223, pp..2123-2130, 1998.

[6] A. Quershi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proc. ACM Special Interest GroupData Commun., 2009, pp.123-134.*

[7] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramaniam "Optimal power cost management using stored energy in data centers," in *Proc. Int. Conf. Meas. Model. Comput. Syst.,* 2011, pp. 221-232.

[8] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar, "Benefits and limitations of tapping into stored energy for datacenters," in *Proc. 38th Annu. ISCA, 2011, pp.341-352.*

[9] M. Sathiamoorhthy *et al.,* "Xoring elephants: Novel erasure codes for big data," in in *Proc.39th Int. Conf.* very Large databases, 2013, pp.325-336.

[10] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstien and C. Welton, "Mad skills: New analysis practices for big data," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1481_1492, 2009.

[11] F. Chen, M. Kodialam, and T. V. Lakshman, ``Joint scheduling of processing and shuffle phases in mapreduce systems," in *Proc. 29th Int. Conf.Comput. Commun.* 2012, pp. 1143-1151.

[12] H. Shachnai, G. Tamir, and T. Tamir, ``Minimal cost reconfiguration of data placement in a storage area network," *Theoretical Comput. Sci.*, vol. 460,pp. 42-53, 2012

[13] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan,``Volley: Automated data placement for geo-distributed cloud services," in Proc. 7th USENIX Symp. NSDI, 2010, pp. 17-32.

[14] A. Cidon, R. Stutsman, S. Rumble, S. Katti, J. Ousterhout, and M. Rosenblum, ``MinCopysets: Derandomizing replication in cloud storage," in *Proc. 10th USENIX Symp. NSDI*, 2013, pp. 1-5.

[15] S. Gunduz and M. Ozsu, ``A Poisson model for user accesses to web pages," in Computer and Information Sciences-ISCIS. Berlin, Germany: Springer-Verlag, 2003,pp. 332-339.

[16] Simon Buckle "Introduction to VoltDB," Independent Consultant Freelance 11 December 2012.