Open Access

Open Journal of Big Data (OJBD)
Volume 1, Issue 1, 2015

www.ronpub.com/journals/ojbd
ISSN 2365-029X

An Efficient Approach for Cost Optimization of the Movement of Big Data

Prasad Teli, Manoj V. Thomas, K. Chandrasekaran

Department of Computer Science and Engineering,
National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India
{prasad.s.teli, manojkurissinkal, kchnitk}@gmail.com

ABSTRACT

With the emergence of cloud computing, Big Data has caught the attention of many researchers in the area of cloud computing. As the Volume, Velocity and Variety (3 Vs) of big data are growing exponentially, dealing with them is a big challenge, especially in the cloud environment. Looking at the current trend of the IT sector, cloud computing is mainly used by the service providers to host their applications. A lot of research has been done to improve the network utilization of WAN (Wide Area Network) and it has achieved considerable success over the traditional LAN (Local Area Network) techniques. While dealing with this issue, the major questions of data movement such as from where to where this big data will be moved and also how the data will be moved, have been overlooked. As various applications generating the big data are hosted in geographically distributed data centers, they individually collect large volume of data in the form of application data as well as the logs. This paper mainly focuses on the challenge of moving big data from one data center to other. We provide an efficient algorithm for the optimization of cost in the movement of the big data from one data center to another for offline environment. This approach uses the graph model for data centers in the cloud and results show that the adopted mechanism provides a better solution to minimize the cost for data movement.

TYPE OF PAPER AND KEYWORDS

Regular research paper: *cloud computing, big data, data movement, graph model*

1 INTRODUCTION

Looking towards the growth of Internet of Things [21], cloud has become one of the biggest industries of the world. So, large amount of money has been poured into the related activities, the research by big cloud service providers like Google, Amazon, Microsoft as well as various research institutes and the government organizations. For the next 5 to 10 years this trend will go on [9]. With the unprecedented growth of various cloud services, the data is being generated at high velocity, variety and volume which we call as big data.

As the storage services have become cheaper because of low cost storage devices, everybody wants to store each and every aspect of the application process such as logs, text, audio, video etc. Moving this kind of data over the network cannot be done through the traditional LAN technologies. So, for the efficient utilization of the network, the CSP (Cloud Service Provider) uses WAN, specially designed to achieve the high utilization of network with low latency. These design techniques are called as WAN optimization techniques. There has been a lot of work already done to improve the WAN, and many advanced and stable tools and technologies are present in the market. These are called as the WAN Optimizers [19], which improve the bandwidth utilization in WAN, and also improve the performance of the network. But the later part, moving data from one location to another geographically distant location, has not been considered much as compared to this WAN optimization. Even if you use the best technique available to transfer the data, the issue of where to move and how to move the data efficiently remains to be solved.

Whenever an application vendor wants to move on to the cloud, he has to transfer a bulk of data to the cloud. This is a one-time procedure, and hence vendor has to bear this cost. Currently, there are many hardware and software solutions such as WAN optimizers discussed previously. Another scenario is to transfer bulk data between geographically distributed data centers. In cloud environment, as the applications are hosted on multiple, geographically distributed data centers, the data collected at each data center has to be moved to a single assembly site. The reason behind this is performing research operations such as tracking user behavior based on the server logs, using map-reduce like framework will take large amount of bandwidth. In the shuffle and reduce phase of map-reduce operation, large amount of data will be moved between geographically distributed data centers. To improve the performance of the system you need the information of each and every geographically distributed data center hosting the same application. As this is not a small scale data, effective techniques are necessary to finalize the assembly site. The traditional way of moving such large data is through shipping the hard drives physically [2]. This approach is less secure and delays the overall data transfer.

When we talk about the cloud services, we always try to find out which one is the best for our requirements. While choosing the services for an application, we may opt for different service providers for different services for the same application. For example, we want an investment management service and a data analytics service. It may happen that we chose two different service providers for these two requirements. The work [5] suggests to combine these requirements and give both the services together to improve the quality of service and get better results. Authors in the work give example of combination of Return on Investment (ROI) Measurement as a Service (RMaaS) and Risk Analysis as a Service (RAaaS). This approach mainly achieves cost reduction due to single service provider, better way of checking consistency of results and pipelining approach where results of one services are fed as input to the other one. This work is a very good example of integrating different business services. This approach makes things easier in case of big data particularly because processing big data again and again for different business services and at different locations will be very difficult and costly. Combining more business services together and performing at one location will give better results at comparatively lower cost.

Authors in this work give results based on the case studies performed at University of Southampton and Vodafone/Apple, which shows improvement in quality of results. This approach helped in getting more accurate results about return on investment. With the use of results of return on investment, authors calculate the risk using different statistical methods. Authors also represent the results in 3D so that they are more readable. Another approach [3] explaining business intelligence as a service by Victor Chang extends the previous work. This model uses APIs (Application Programming Interface) to calculate risk, pricing using Heston Model. Author also provides business analytics in visualized format. This work gives experimental results to prove that Business Intelligence as a Service (BIaaS) on cloud is better than the desktop versions of the services.

Medical data which includes medical images also comes under the big data. Size of medical images is large. Victor Chang et al. [4] describe a case study which uses Cloud Computing Adoption Framework (CCAF) along with Storage Area Network (SAN) technologies. This work represents experiments for the data services such as backup automation, data recovery and data migration. This work gives explanation about benefits of moving the medical data to the cloud platform. Results of the work show that execution time for data recovery and data migration is proportional to quantity of recovered data and volume of data migrated respectively. But the failure rate increases exponentially with the growth of size of data. This work explains the benefits of using CCAF, which can be designed according to the requirements of the application and the data. This work also supports the reduction of cost of the data migration with the use of cloud platform. So, instead of using traditional systems for storing this biomedical data, use of private cloud platform gives cost effective, user friendly and time saving solution.

In this paper, we have designed a graph model for the system of geographically distributed data centers. This gives us better understanding of the costs of the links connecting these data centers. In this model, we consider all the required parameters to calculate the complete cost of migrating data from one data center to another. We provide the solution for this issue of big data transfer from one data center to another geographically distributed data center, which gives the optimal cost solution to decide the aggregation site. Proposed approach gives an efficient algorithm for the decision of the site, where big data from all the other data centers will be moved for further processing.

Thus, in this paper we mainly focus on the above mentioned problem of big data movement and propose an efficient approach for the same. The remaining contents of the paper are organized as follows. Section 2 addresses the various challenges involved in moving big data in cloud environment. In section 3, currently available techniques to tackle the challenges are explained. The proposed graph model for the big data movement between data centers is discussed in section 4. Section 5 explains the experimental setup and results. Some characteristic features and key issues related to the proposed work are discussed in section 6. Finally, in section 7, we draw the conclusions and future works.

2 CHALLENGES IN HANDLING THE BIG DATA

Big data provides large number of opportunities such as improving business by analyzing the user behaviors, getting more accurate predictions in scientific research by analyzing large number of previous results etc. As big data provides these opportunities, along with this, there are many challenges while handling big data [6]. Considering the big data, the challenges lie mainly in areas such as data capture, data storage, analysis, representation. To handle these problems, we have many powerful tools like Hadoop [12], which is mainly used for analyzing the big data, and it works on the principles of parallel computing. For data storage, Direct-Attached Storage (DAS), Network-Attached Storage (NAS) and Storage Area Network (SAN) [17] are used. But when size of data is very large, these approaches also have some drawbacks. For efficient storage and retrieval, we have NoSQL [20] databases like HBASE, MongoDB, Cassandra etc. These are mainly built to handle the unstructured data along with the traditional structured data.

The challenge of data handling comes later, and we firstly need to focus on the challenge of the data capture. Data should be available for its further processing. Currently, main Cloud Service Providers are working to provide the services having low latency and high throughput. Figure 1 shows a scenario in which an application is deployed over multiple geographically distributed data centers. Data centers are connected to each other via specially designed high bandwidth WAN links. Each user is connected to one of the data centers. Service providers redirect the user request to the nearest data center hosting the application. This achieves the goal of providing the users a fast service. But, at the same time, if a CSP (Cloud Service Provider) or an application vendor needs to analyze this data for different purposes like user behavior analysis, application data analysis etc., this scenario creates the challenge of not having all the data at one site. Then, it needs to analyze data at each site, get all the results and

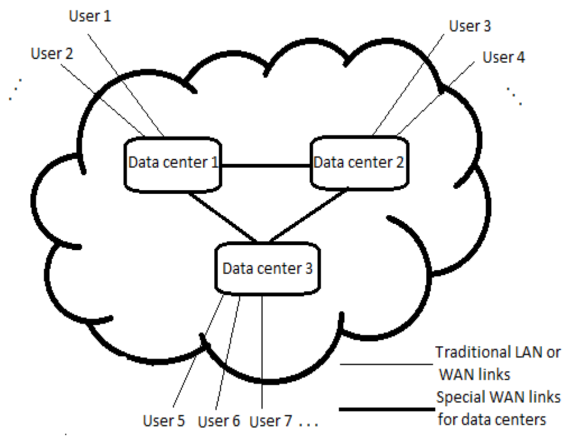

Figure 1: Current System for Applications in the Cloud

further process them and come to final decision. This may not give optimal results, always.

The solution to the above mentioned scenario is to aggregate all the data at one data center, and then do the required processing. As size of this data is not small, it is associated with many challenges. First one is bandwidth which is not uniform across all the links in between data centers. Second one is storage cost, which may vary in different data centers. Third one is processing power and processing cost of data centers can be different as they do not host only one type of application. Hence, there is a need of an efficient algorithm which will move the data from all geographically distributed sites to an aggregation site. We are proposing a solution to the problem of cost effective big data movement.

3 TECHNIQUES OF BIG DATA MOVEMENT IN CLOUD

There are some approaches proposed in the literature for moving big data in the cloud environment. This section gives an overview of the different works done in this area of big data migration. These approaches can mainly be divided into two categories: migration-based and data-placement based approaches.

3.1 Migration-Based Approaches

The migration based approaches consider the migration of the big data from one site to another. This migration includes complete transfer of data from one place to another with minimum cost.

Following are the major approaches proposed in the literature.

3.1.1 Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to cloud

Whenever an enterprise application is to be moved to the cloud, both the CSP and application owner has to decide many things such as security policies, access rights, data movement. The work [13], gives a solution to the enterprise application owners to move the part of their application on to the cloud, and keep the part of it to their local environment. Reason behind keeping some part at the local site is that some data cannot be allowed to be on cloud because the data might involve very sensitive data such as the credit card information or some national security data. Also, there are different types of users such as the internal users and the external users. So access policies should be different for them.

Considering an example of a college website, students of that college will be internal users and anybody else such as alumnus of the college or somebody seeking an admission in the college will be external user. So, how to decide which part of the data should be moved on to the cloud and which should be kept locally, has to be solved. This approach divides the application into three tiers. First, the front end, which mainly contains the web servers serve the client requests, second the servers which perform the business logic part of the application, lastly and most important part is the database servers which hold the data of the application. Based on the requirements, these parts are moved to the cloud or kept locally.

3.1.2 PANDORA: Budget Constrained Bulk Data Transfer

In cloud, there is requirement of transferring bulk data from one data center to another which are geographically at large distance. PANDORA (People And Network moving Data AROund) [7] is a model proposed to transfer large amount of data with budget constraints. Since in the cloud domain, bandwidth is being charged as per the use, we cannot have large bandwidth if we have budget constraints. This increases the data transfer time. That can also take weeks if the data is too big, and also if we have small budget. The solution to this problem can be the shipping of physical disks from one location to another. This approach finds the optimal solution considering both the online as well as physical shipment of data. But this approach also have problems such as security of disks and the time to transfer can vary according to conditions of the transportation medium.

3.1.3 Online Cost Minimization Approach

The approach, proposed by Linquan Zhang et al. [23] mainly suggests how to decide the best site for the data aggregation. This approach suggests two algorithms namely Online Lazy Migration (OLM) and Randomized Fixed Horizon Control (RFHC), which minimizes the cost of the data aggregation of geo-distributed applications. First of all, they have formulated the problem of data migration considering all the parameters which add to the data movement cost such as migration cost, the routing cost and aggregate storage and computation cost. This approach has considered the users using the application as data generators which are connected to the data centers through gateways. Data centers are geographically dispersed. Firstly, they have formulated the problem and given the algorithm for minimizing this cost for offline data, which means all the parameters are fixed and given to the algorithm. Results of this algorithm are considered as the benchmark for the two algorithms namely OLM and RFHC, which work for the online data and run in polynomial time. Here, all the parameters change over the time, as the data is continuously generated at each data center at different rates.

First algorithm, Online Lazy Migration, uses the dynamic programming approach for solving the problem. Second, Randomized Fixed Horizon Control algorithm makes use of an algorithm adapted from the preceding Horizon Control Algorithm [18]. This is similar to Markow Chain Model which predicts the future pattern based on the past events. This algorithm calculates the cost of data aggregation which is upper bounded by the offline cost of aggregation plus the cost of moving data to the aggregation site.

3.1.4 SSH Model - An Efficient Transportation Architecture for Big Data Movement

Work in [15], proposes a new data transfer model on packet network which is an extension to the existing per packet forwarding architecture. In this model, the data is not directly sent as packet by packet immediately as it enters the network. This model makes use of store and forward principle. It proposes Store, Schedule and Switch Model for optical network.

Figure 2 shows, if there are two simultaneous requests for the connections $C_1(1, 4)$ and $C_2(3, 5)$, both cannot be granted at the same time. The two sub requests are created $C_{2Sub1}(3, 4)$ and $C_{2Sub2}(4, 5)$, so that overlapping part is separated. The C_{2Sub2} can be processed along with C_1 and C_{2Sub1} is delayed. For the scheduling of the packets, the SDN (Software Defined

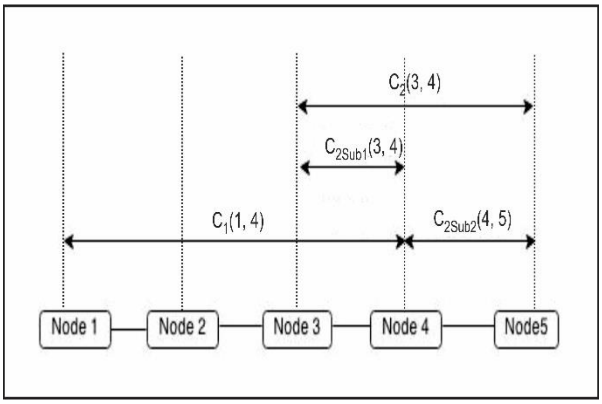

Figure 2: Store, Schedule and Switch (SSH) Architecture

Networking) [14] controller checks the deadline, size of the data, destination and creates a flow. Data is sent to the nearest switching node from the source. The data from multiple sources and going through the same switch is stored. Data packets with close deadlines and for the same destination form a batch. Keeping the deadline in mind, decision is taken whether the batch is to be sent to the next node. As the data is stored in between, it relaxes the timeliness of delivery. This approach does not need any special requirements for the underlying Virtual Private Network (VPN).

3.2 Data Placement-Based Approaches

Data placement strategies try to minimize the data movement instead of the data movement cost. They try to place the data at the sites according to the requirements of data. These strategies provides users with the availability of data at appropriate locations, which are decided based on the proposed algorithms. In this section, some of the data placement strategies which try to minimize the data movement are discussed.

3.2.1 A Data Placement Strategy Based on Genetic Algorithm for Scientific Workflows

One of the important problems of geographically distributed data centers is how the data is placed. The work done in [8], deals with data placement problem which is NP hard. There are many methods to deal with the problem. K-means clustering is one of them, but it suffers from the problem of unbalanced load, because the data sets can get concentrated at a few locations. In K-means clustering, the data sets are placed based on the dependency of the data set on the target data center. But it can happen that more data sets have high dependency on one data center. This will result in overloading of the data center and some data centers may not have any data sets at all. This approach uses a heuristic based genetic algorithm to find the right data center for the data sets.

Genetic Algorithm generally contains the initial population selection, select operation, crossover operation, mutation operation and the calculation of the fitness function. This algorithm contains the placement strategy as a gene. Initially, they consider that a single data set is placed at each data center. After the mutation step, the correctness of the gene is checked to verify if it is according to the fitness function or not. That means some gene code may be invalid or unacceptable, which can overload a single data center. This algorithm takes the help of heuristic information to mutate the illegal gene code. As compared to the K-means algorithm, this algorithm gives less data movements as per the simulation results. So, this algorithm places the data sets in such a way that dependency of the data set with the data center is considered. The load balancing is also taken care of, which results in low data movement at the time of task execution which requires multiple data sets.

3.2.2 Scaling Social Media Application into Geo-Distributed Clouds

With the emergence of social networking applications such as Facebook, large number of social media files are getting uploaded to and downloaded from the internet. The YouTube like applications, which need to provide low latency to their videos because most users will not be satisfied with even some seconds of delay while watching the videos which may be of a few minutes in length. To achieve this, cloud service providers are collaborating with each other. By doing so, CSPs can keep the data to a data center located nearest to the user. But the problem is this social media is very dynamic, and users are also with different trends according to their geographical areas and as well with different likes and dislikes.

To tackle the challenge of distributing the requests to the appropriate sites and dealing with the movement of data to the optimal site, an approach has been proposed by Yu Wu, Chaun Wu et al. [22]. They propose an algorithm to find out the future demand based on the social influence on the users. Also, they give one-shot optimization algorithm to distribute the user request and also for the movement of the data between data centers. Finally, they propose $\Delta(t)$-step look-ahead mechanism to optimize the one-shot algorithm results.

Videos stored in the data center are considered as a graph in which all the videos are interrelated in some manner. So, a user viewing any cricket video is likely to be watching the upcoming cricket tournament videos or videos of recently held matches. Based on this, future needs are predicted. Recommendation system, implemented by the application also suggests some videos. Friends of a user watching a video are also considered to be the potential viewers of the video.

To predict the future access of a video, the probability is considered which is highest for a new video. After calculating the future needs, one shot optimization algorithm gives the minimum cost locations for the data, and optimal distribution of requests. This is optimal for only one time slot at any given point of time, and this solution does not guarantee the optimal performance over the long periods of time. To achieve the optimal solution $\Delta(t)$-step look-ahead mechanism is used, which mainly takes the input of optimal solution at time slot t-1 and calculates for the time slot t. This improves the performance towards the optimal offline algorithm (as discussed in section III subsection 3.1.3, where parameters do not change over the time).

3.2.3 Volley: Automated Data Placement for Geo-Distributed Cloud Services

This work is done by Microsoft Research Team [1]. In order to reduce the latency, the system places data nearest to the primary IP address, and it can cause overloading of a particular data center and also some data centers will be left underutilized. Main challenges considered in placing the data are sharing of the data, data inter-dependencies, application changes, reaching the limits of the data center capacity and most important the mobility of user. Volley is the model which takes data center logs as the input, and gives the optimal results for the placement of data at geographically distributed data centers. This data placement or processing can be done by Map-Reduce like platform. The data center administrator has to specify various parameters such as the locations, cost and capacity model of data centers, bandwidth costs, current trade off between migration cost and current performance, replication level of data and other constraints.

This model studies the geographically distributed users of the data and their access patterns, and calculates the centroid for them. It differentiates the data objects based on their interdependence. The user mobility is considered in such a way that, when a user accesses a particular data, for the next time he may have travelled a long distance, or he may be using a VPN connection which does not reveal his exact location. But this model assumes that no need to do something different for it, even if the user is accessing through a VPN, data is placed nearest to the VPN link access. Considering all these things, Volley takes data center logs as input from different sites, and calculates the migration requirements which can be sent to all the data centers. Volley can be rerun over the time efficiently, so the performance of system is not affected much by Volley.

3.2.4 Cost Minimization for Big Data Processing in Geo-distributed Data Centers

To reduce the data movement inside the data center, and also to save the power consumption, Data Center Resizing (DCR) is used. This includes the task placement on the servers in order to use only those number of servers, which are sufficient to serve the current requests, and shut down the remaining servers. Another approach is using the data locality to improve the response time. But this has not achieved efficient processing of the big data, because if the data locality is used, then the wastage of resources will occur, and the link states are not considered in this approach, which have varying properties over the time. Also, Quality of Service is not taken care of.

The approach proposed by Lin GU et al. [10] gives an efficient method which minimizes the cost of big data processing in cloud considering three parameters such as data placement, task assignment and data routing. This model, which is a graph model, considers all the servers at a data center to be of same computing power and same storage capacity. Data is divided into chunks of equal size which depends on the storage capacity of the server. Number of chunks are P-way [16] replicated. The tasks are considered to access these chunks, and task arrival rate is in Poisson fashion [11]. The problem formulation is done by considering the constraints on data and task placement. The constraints on the data loading include data flow from source nodes through intermediate switches to the destination, where data chunk is required. At last, constraints on QoS satisfaction are modeled using two dimensional Markov model.

3.3 Performance Analysis and Evaluation

Work explained in the section 3.1.1 showed improvement in response time for the application like Thumbnail, where users upload pictures and ERP application run on cloud platform than running it locally. While performing the migration of applications, access policies need to be modeled carefully. The critical part in this approach is applying ACL (Access Control List) rules. These policies need to be modeled in such a way that both the cloud platform and the application owner should be benefited in terms of scalability, complexity and response time. PANDORA (section 3 .1.2) uses both physical as well as online transfer of bulk data. This uses two approaches namely two step binary search and bounded binary search. Bounded binary search approach gives faster results for short time transfer than two step binary search. This approach gives a better way to compute the partial optimal results. Cost of data migration can be considerably reduced using online approach (section 3.2.3). It uses two algorithms for migrating the data to an aggregation site. First algorithm, which is OLM, is reactive approach. First algorithm considers the scenario of frequent change of aggregation site and delays the change of aggregation site. Second algorithm (RFHC) uses prediction to calculate the future size of data at each site and based on that takes the decision of aggregation. But as we all know prediction may not give correct results all the time. Work in section 3.2.4 gives a store, schedule and switch architecture. It stores the data in between source and destination. The data for same destination forms a batch at intermediate node. This approach faces the problem of limited storage space at intermediate nodes and it also considers the previous data to grant the data transfer request which may not give correct result always.

Data placement strategy discussed in section 3.2.1 uses genetic algorithm. This approach gives results same as of K- means clustering in case of data movement. The only benefit is better results in case of load balancing as compared to K- means clustering. Again, accuracy of algorithm depends on initial gene selection and mutation function. For social media applications like YouTube, data placement and request distribution strategy is explained in section 3.2.2. It also predicts the future video requests based on previous video views and recommendation systems. This approach is beneficial in case of particular application hosting videos. But for other social media or the big data generating application, it requires to be modified. Volley, explained in section 3.3.3 gives a data placement solution which can be adapted by diverse set of applications. It gives the liberty of choosing data migration option to service provider itself. Still there is a lot of scope for improvement as big data applications are very diverse. Data Center Resizing (DCR) approach explained in section 3.2.4 gives a data placement solution. This approach takes the server load, data placement and data movement into consideration. This shows that four replicas of each chunk of data give better results with the parameters specified by the authors. The parameters used for decision of task assignment and data placement are system specific; hence decision of the parameters is a tough task for diverse set of applications.

4 Graph-Based Model for the Data Movement

In this section, we propose a graph-based model for the system of geographically distributed data centers. We also propose an efficient algorithm for offline environment.

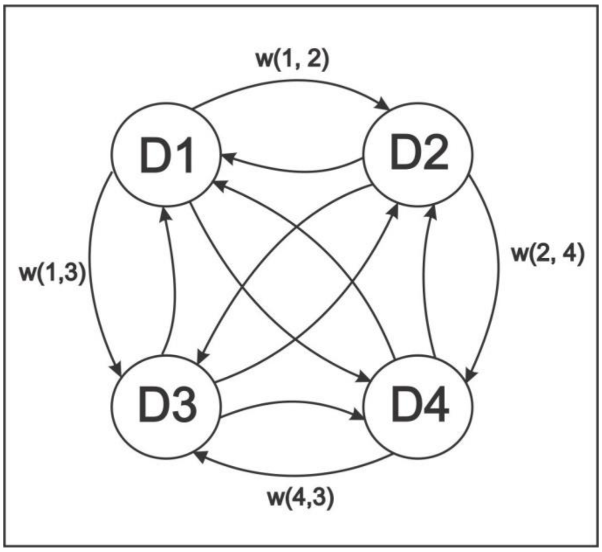

Figure 3: Proposed Graph Model

4.1 Graph Model

This approach considers the system of data centers as a directed complete graph. Each node of the graph is one of the data centers distributed geographically. Each edge is associated with a weight. The weight of each edge shows the cost of migrating data from one node to another. This weight is not same in both directions for a pair of nodes, because available bandwidth between source and destination can vary in case of virtual networks. In case of the use of SDN [14] like technologies, each flow is assigned with its own bandwidth. Also the cost of processing of the data and the size of the data may not be same at each node. This also makes sure that the weights in both directions are not same.

Figure 3 shows the proposed graph model for geographically distributed data centers. Each edge of the graph has weight associated with it. For convenience, in figure, it is shown for only outer four edges. Each of the nodes represents a data center. The links connecting them are of high bandwidth and specially designed intranet links are used between the data centers.

4.2 Problem Formation

We have assumed that all the data generated at a data center by all VMs will be available on a particular VM. We create a cost matrix (*CostMatrix*) of size $n \times n$, where n is the number of nodes of the graph. Each element of this matrix shows the cost of data movement. So, $(i, j)^{th}$ element of matrix will show the cost of data migration from i^{th} data center to the j^{th} data center, where $0 \leq i < n, 0 \leq j < n$.

To calculate this weight or the cost of the data migration we consider the following costs:

1. Cost of Data Transfer (C_{DT}) - This cost includes the cost of bandwidth required to transfer the data between data centers.
2. Cost of Data Storage (C_{DS}) - This is the actual cost of data storage at the destination node. This is the cost of disk space.

We calculate the cost of transferring the data from the i^{th} node to the j^{th} node as

$$CostMatrix(i, j) = C_{DT}(i, j) + C_{DS}(i, j) \quad (1)$$

Now we have to minimize this cost, so that we can find out the optimal cost aggregation site for the data migration.

4.3 Proposed Algorithm

Here, we propose an algorithm to minimize the cost of big data migration in cloud environment.

Step 1 *Creation of Bandwidth Cost Matrix:* To calculate the data transfer cost, first we create a bandwidth cost matrix (*BWCostMatrix*) of same size n × n as of main cost matrix.

BWCostMatrix (i, j) = cost of bandwidth to transfer unit data from the i^{th} *node to the* j^{th} *node*

This matrix contains the $(i, j)^{th}$ element as bandwidth cost of transferring unit data from i^{th} node to the j^{th} node, which is calculated per GBPS per unit time.

Step 2 *Minimizing the Bandwidth Cost:* Now using Dynamic programming, we find the all source shortest paths from every node to all other remaining nodes.

Here, shortest path is the path with least bandwidth cost. This includes the end to end delays also. So suppose the cost of transferring a unit data from from i^{th} node to j^{th} node is greater than the sum of the costs of transferring a unit data from from i^{th} node to the k^{th} node and transferring a unit data from k^{th} node to the j^{th} node, where k^{th} node is an intermediate node. Then, we update the bandwidth cost matrix with the cost of transferring unit data from i^{th} node to the j^{th} node through k^{th} node. The reason behind doing this is, if we have an alternate path to the destination with lower cost, we use that path, even if it is not the direct one. We consider that the data is transferred through the edge switch. So, extra cost added is just the latency of the switch. While updating this cost, we also consider the scenario where one of the links of this low cost path is having less bandwidth. In this scenario, we calculate the total time required to transfer the data over these two links. Considering this time and also the costs of both the links, if it is still lower than the direct path, then only we update the cost in bandwidth cost matrix. *BWCostMatrix(i, j)* will contain the minimum bandwidth cost for transferring the data form i^{th} node to the j^{th} node.

Algorithm 1 Proposed Algorithm

```
Initialize BWCostMatrix(i, j)
for all i = 0 to n − 1 do
    for all j = 0 to n − 1 do
        minimize BWCostMatrix(i, j)
    end for
end for
Calculate CostMatrix(i, j)
```

Step 3 *Minimizing the Total Migration Cost:* Now cost of data transfer C_{DT} will be calculated as follows,

$$C_{DT}(i, j) = BWCostMatrix(i, j) \times size \quad (2)$$

where, *size = size of data in GBs to be transferred from* i^{th} *node to the* j^{th} *node*

Now, we have the *CostMatrix* with final optimal values of the costs. We need to minimize these cost values. From this cost matrix, we can calculate the cost of aggregating data from all other nodes to a particular node. Minimum of all these gives us the optimal solution for the decision of aggregation of the data.

5 Experimental Results

In this section, we discuss the experimental setup used to simulate our experiment and also the analysis of the results obtained from the proposed algorithm is carried out. We have simulated the data center environment using JAVA programming language. We setup a system of data centers and then we calculate the cost of data migration with a fixed size approach, and then with the proposed approach. Later we analyze the results and the algorithm.

5.1 Experimental Setup

We have carried out the simulation on a machine with Intel i7 3.4GHz quad core processor with 8 GB RAM. In our simulation, we have considered 4 data centers having data of different sizes stored on each. Each data center is having processing power in range of [1000, 1500] MIPS. Each data center is connected to each of the remaining three data centers with the links having different available bandwidths in the range of [5, 20] MBPS and different bandwidth costs in the above ranges. Cost of data processing and data storage is also considered as different at each data center.

We consider the cost of storing 1 GB data at a server is in between [0.01, 0.1]$/h. Bandwidth cost of 1GBPS is considered in between [0.05, 0.2]$/h.

5.2 Results and Analysis

Results are collected from two different approaches. First, we aggregate the data at a site having largest size of data already collected. Second approach is using the proposed algorithm.

Figure 4 shows the comparison between the costs of data migration. X-axis shows the size of data aggregated in TBs. Y-axis shows the cost of data migration in $. In the figure, line with circles shows the cost of aggregating data at the site having the largest data size. The line with boxes, shows the costs of data migration using the proposed algorithm which finds the minimum cost of data aggregation based on the previously discussed parameters such as the cost of bandwidth, the cost of the data storage and the cost of the data processing. If only size of data is considered, it does not give the optimal cost solution for the data migration. The cost of data aggregation changes because of the data storage cost and the bandwidth cost, even if the size of data originally stored at the aggregation site is largest among all the sites. These results show that the aggregation of the data with proposed algorithm gives optimal cost solution compared to aggregating data on the site having largest sized data.

Figure 5 shows the cost of aggregation of the data at a site with lowest storage cost against the aggregation cost with the proposed algorithm. This clearly shows that parameters such as data size and storage cost affect the decision of finding the aggregation site. Even if the storage cost at the aggregation site is lowest, the size of data at each site varies, which leads to the result of different aggregation site. In some cases, the site with lowest storage cost resulted as the aggregation site. But this may not happen always. In the first experiment, it did not happen. We conducted these experiments with random data sizes at each location, random bandwidth cost and random storage cost in the range specified in section 5.1.

Work explained in section 3.1.2 finds out the optimal cost solution for transferring data from one geographical location to the other. The problem with this approach is that it also considers the physical shipment of the hard drives. As compared to this work, proposed approach gives a solution which does not face the difficulties such as weather conditions and reliability of the transport medium. Also, in comparison with the work explained in section 3.1.4, proposed approach overcomes the difficulty of storage capacity which arises due to storing

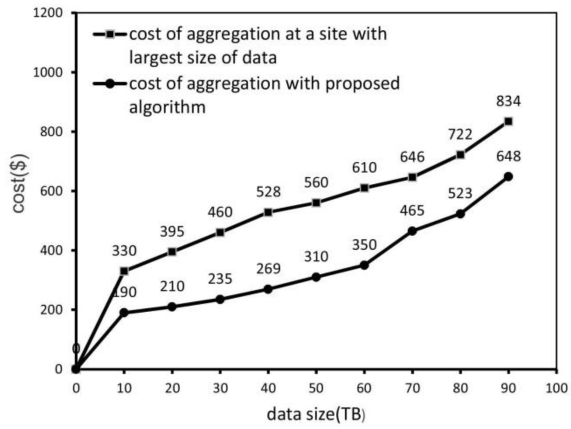

Figure 4: **Costs of the data transfer using fixed aggregation site with highest data size and proposed algorithm**

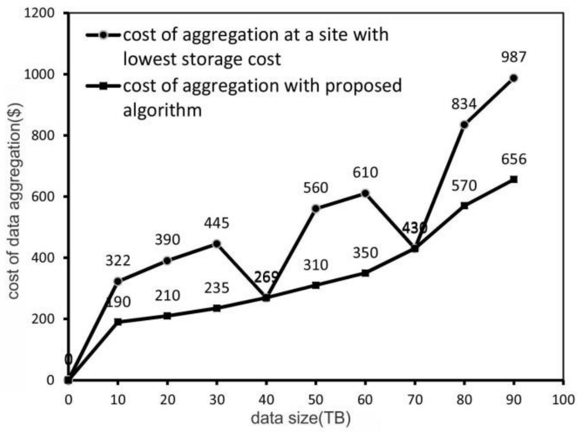

Figure 5: **Costs of the data transfer using fixed aggregation site with lowest aggregation cost and proposed algorithm**

data in between and then forwarding. Data placement based approaches explained in section 3.2 focus on the live application data and not the log data. These approaches work well for the applications data such as social media applications. Data placement based approaches for data migration focus on minimizing the data movements of the application data in between the geographically distributed data centers. Proposed approach gives a solution for the data which is mainly useful for service providers to improve their services.

In comparison of the processing of data at the geographically distributed data with the processing of data at one site, latter results into simple and low cost solution in case of the data such as log files. While processing data at geographically distant sites with Map Reduce like frameworks, data will be moved between the sites. This data movement costs more bandwidth as compared to the bandwidth cost of aggregating data at one site. Web applications are very dynamic in nature. The load on the applications is varying and the size of data generated at each site depends on various parameters such as time of the day, geographical location of users, social interests of the users for social networking sites, festival times or special occasions for the e-commerce applications etc. Aggregating data at a fixed site will not give optimal cost results always. It requires an optimal cost solution which considers all the parameters as explained previously. Proposed approach gives a model to minimize the cost of data aggregation.

Complexity of the proposed algorithm is $O(n^3)$, where n is the number of nodes in the graph. This algorithm considers the migration of the data in offline environment. So, the parameters such as bandwidth are not changing over the time. The proposed algorithm is best suited for the data which needs to be collected at a site over a particular period such as daily at the end of working hours of a particular firm.

6 DISCUSSION

Some of the characteristics of the proposed work are discussed in this section.

6.1 Size of Data

Cloud platform creates huge opportunities as well as challenges for the data intensive research works. As size of data grows, data storage, data recovery and data migration become the important areas for both the cloud service providers and the cloud service users. To handle such a large amount of data, it requires advanced techniques of both hardware and software. Advancement in hardware solutions is affected by multiple factors such as energy consumption, availability of materials with which hardware is made, the cost of the material etc. Software solution is a wide area which is open and attracting many researchers. A lot of work needs to be done in the area of data migration and data handling in cloud environment.

6.2 Challenges in Data Migration

As discussed previously in the Introduction section, data migration itself is a challenge. While migrating big data from one location to the other, parameters such as bandwidth, storage cost, storage capacity etc. need to be taken care of. Along with these basic parameters there are some more complex issues as well. This includes live migration. Our proposed approach as of now does not consider this. But, current web applications cannot afford the downtime. Hence, live migration is a big challenge and hot topic in the cloud environment. Data Migration to the cloud has to be done in parallel to the running application. Variety of data also plays a vital role in the data migration. Since the structured and unstructured data are nearly inseparable in most of the current systems, data migration process becomes more complex.

6.3 Data Processing

Because of the geographically distributed data centers, we need to use Hadoop like distributed frameworks. As the size of data is very large and it is distributed at geographically distant sites, processing this data consumes a lot of bandwidth. One of the solutions to this problem is data aggregation. Our work mainly focuses on reducing the cost of data processing by means of reducing the data aggregation cost. Because of the geographically distributed data centers, we need to use Hadoop like distributed frameworks. As the size of data is very large and it is distributed at geographically distant sites, processing this data consumes a lot of bandwidth. One of the solutions to this problem is data aggregation. Our work mainly focuses on reducing the cost of data processing by means of reducing the data aggregation cost.

6.4 Limitations

This proposed work considers the data in offline cloud environment. Total size of data depends on the storage capacity of the aggregation site. As of now our approach mainly considers the data such as server log data. Log file size depends on the application. Generally these files can be in GBs, if we consider for the time span of a day. We assume that each site has enough space for storing data from each of the remaining sites in case the proposed algorithm results the given site as the aggregation site.

7 CONCLUSION

Cloud computing is a continuously evolving area, and so is the big data processing. Traditional approach of moving the data via physical shipments is not the best solution, even if we have cost constraints. This paper gives an efficient algorithm for moving big data in cloud environment. One thing is quite certain that even if the hardware technology is getting better and storage is becoming cheaper every day, there will always be the scope for reducing the cost of data migration by finding optimal solution to the data movement and storage. Hence, in future work of this paper, we mainly focus on, the design and development of an algorithm for the data movement in online environment, where system parameters are changing over a period of time.

REFERENCES

[1] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated data placement for geo-distributed cloud services," *7th USENIX conference on Networked systems design and implementation*, 2010.

[2] Amazon, "Aws import/export," http://aws.amazon.com/importexport/, accessed May 27, 2015.

[3] V. Chang, "The business intelligence as a service in the cloud," *Future Generation Computer Systems*, vol. 37, pp. 512-534, 2014.

[4] V. Chang, R. J. Walters, and G. Wills, "Cloud storage and bioinformatics in a private cloud deployment: Lessons for data intensive research," *Cloud Computing and Service Science, Springer Lecture Notes Series, Springer Book*, 2013.

[5] V. Chang, R. J. Walters, and G. B. Wills, "Business integration as a service," *Business Integration as a Service*, vol. 2, pp. 16-40, 2012.

[6] C. L. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information Sciences*, vol. 275, pp. 314-347, 2014.

[7] B. Cho and I. Gupta, "Budget-constrained bulk data transfer via internet and shipping networks," *ICAC11*, 2011.

[8] Z. Er-Dun, Q. Yong-Qiang, X. Xing-Xing, and Y. Chen, "A data placement strategy based on genetic algorithm for scientific workflows," *Eighth International Conference on Computational Intelligence and Security*, 2012.

[9] Forbes, "10 critical tech trends for the next five years," http://www.forbes.com/sites/ericsavitz/2012/10/22/gartner-10-critical-tech-trends-for-the-next-five-years/, October 2012.

[10] L. Gu, D. Zeng, P. Li, and S. Guo, "Cost minimization for big data processing in geo-distributed data centers," *IEEE Transactions on Emerging Topics in Computing*, 2014.

[11] S. Gunduz and M. Ozsu, "A poisson model for user accesses to web pages," *Computer and Information Sciences - ISCIS 2003*, ser.Lecture Notes in Computer Science. Springer Berlin Heidelberg, vol. 2869, pp. 332-339, 2003.

[12] Hadoop, "Welcome to apache hadoop," http:// hadoop.apache.org/, accessed May 27, 2015.

[13] M. Hajjat, X. Sun, Y. W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani, "Cloudward bound: Planning for beneficial migration of enterprise applications to the cloud," *SIGCOMM10*, 2010.

[14] F. Hu, Q. Hao, and K. Bao, "A survey on softwaredefined network (sdn) and openflow: From concept to implementation," *Communications Surveys & Tutorials, IEEE*, 2014.

[15] W. Hu, W. Sun, Y. Jin, W. Guo, and S. Xiao, "An efficient transportation architecture for big data movement," *ICICS*, 2013.

[16] R. Kaushik and K. Nahrstedt, "T*: A data-centric cooling energy costs reduction approach for big data analytics cloud," *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1-11, 2012.

[17] D. Leong, "A new revolution in enterprise storage architecture*," IEEE Potentials*, vol. 28(6), pp. 32-33, 2009.

[18] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, "Online algorithms for geographical load balancing," in *Green Computing Conference (IGCC)*, 2012 International, pp. 1–10, 2012

[19] M. B. Nirmala, "Wan optimization tools, techniques and research issues for cloud-based big data analytics," *World Congress on Computing and Communication Technologies*, 2014.

[20] NOSQL, "List of nosql databases," http://nosql-database.org/, accessed May 27, 2015.

[21] Rolf H. Weber, Romana Weber, "*Interent of Things*," Springer Berlin Heidelberg, 2010.

[22] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. M. Lau, "Scaling social media applications into geodistributed clouds," in *IEEE INFOCOM*, pp. 684 – 692, 2012

[23] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. M. Lau, "Moving big data to the cloud: An online cost-minimizing approach," IEEE Journal on Selected Areas in Communications, vol. 31, no. 12, pp. 2710 - 2721, 2013.

AUTHOR BIOGRAPHIES

Prasad Teli is currently pursuing Master of Technology in the Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, India. He obtained his Bachelor of Engineering from Govt. College of Engineering Karad, Maharashtra. He has 2 years of Industry experience as a Software Developer. His areas of interests include Cloud Computing and Big Data.A biography for each author should be supplied here.

Manoj V. Thomas is currently pursuing Ph.D in the Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore, India. He obtained his Bachelor of Technology from RIT, Kottyam, Kerala and Master of Technology from NITK, Surathkal with First Rank and Gold Medal. He has more than 10 years of teaching experience and he is a Life Member of Computer Society of India. His areas of interests include Computer Networks, Cloud Computing and Cloud Security.

Dr. K. Chandra Sekaran is currently Professor in the Department of Computer Science and Engineering, National Institute of Technology Karnataka, having 26 years of experience. He has more than 120 research papers published by various reputed International journals, conferences which include IEEE, ACM, Springer etc. He has received best paper awards and best teacher awards. He serves as a member of various reputed societies including IEEE (Senior Member), ACM (Senior Member), CSI, ISTE and Association of British Scholars (ABS). He is also a member in IEEE Computer Society's Cloud Computing STC (Special Technical Community). His areas of interest include Computer Networks, Distributed Computing (includes Cloud Computing and Security) and Business Computing and Information Systems Management.