Uploading and Processing Big Data to the Cloud Using Online Techniques

M. Lalitha[1] ,
[1] Assistant Professor, Dept.of CSE,
G. Narayanamma Institute of Technology & Science (for Women)
Hyderabad, India

Ch. Divya[2]
[2]Computer Science & Engineering
GNITS,
Hyderabad, India.

***Abstract*--Cloud computing is a new computational paradigm that provides scalable resource access in a utility-like fashion to the users. The primary usage of clouds in practice is processing of massive amounts of data. One important issue is how do we efficiently move the data, into the cloud for processing? The general approach is moving data in hard disks which lacks in flexibility and security. This work studies upload of massive, dynamically generated, data in to the cloud, for processing using a MapReduce like framework. It propose efficient offline and online techniques, which enhance the routes to upload data into the cloud and for choosing the best data center to combine the data for processing, at any given time. Two online techniques are used to guide data migration over time. An online lazy migration (OLM) technique and a randomized fixed horizon control (RFHC) techniques.**

Keywords: Cloud Computing, Big Data, Online Techniques.

I. INTRODUCTION

Nowadays cloud computing is a new computational paradigm that has rapid on demand provisioning of scalable server resources like CPU, storage and bandwidth to users in a utility-like fashion, especially for processing of big data with minimal management efforts. The recent cloud platforms like Microsoft Azure, Google App Engine, Rackspace etc, organize a shared pool of servers from multiple data centers and serve their users using different technologies. This resource provisioning makes a cloud platform more attractive for the execution of different applications like computation-intensive ones [1], data-intensive Internet applications like facebook, twitter, and big data analytics applications are depending on the clouds for processing and analyzing their large scale data sets from various locations over time and parse them with a computing framework such as MapReduce and Hadoop [2].An important issue in big data analytics is: How does one move massive amount of data into a cloud? The existing system is copying the big data in a hard drives for physical transportation to the data center. Such physical transportation has some problems like undesirable delay; service downtime .It is also less secure where hard drives may prone to infection of malicious programs and damages from accidents. The current practice is to copy the static data .The challenge is when we target at dynamically and continuously produced data from different locations; example is usage data from different Facebook Web servers. For large data sets, we want to select the efficient data center to aggregate all the data onto, for processing using mapreduce like framework, which is efficient to process the data within the one data center.

The dedicated effort in the cloud computing literature, it studies timely, cost minimizing migration of massive amounts of dynamically generated data into the cloud, for processing with mapreduce like framework. Detailed cost composition is analyzed and identifies the performance for uploading big data into the cloud and formulates the offline optimal data migration problem. It computes the data routing and aggregation strategies and reduce the cost and transfer delay .Two online algorithms are proposed to guide data migration online lazy algorithm (OLM) and a randomized fixed horizon control algorithm (RFHC).

A. Open Source Cloud Platform

Cloud Computing, is the dream of computing as a utility, which has the potential to convert a large part of the IT industry, developing software even more attractive and efficient as a service. The Cloud Computing refers to both the application delivered as a service over the internet and the hardware and system software in the datacenters that provide those services. Cloud is available in a pay-as-you-go manner to the public, it is a public cloud. Current examples of public cloud are Amazon Web Services, Google AppEngine, and Microsoft Azure. It is a model for providing on-demand network access to a shared pool of computing resources like servers, storage, networks, services and applications that can be rapidly supplied and released with lowest management effort for a service provider interaction [3]. From the view of hardware, three new aspects are there in Cloud Computing.

1. The illusion of vast computing resources is available on demand, so for cloud computing users no need to plan for supply of resources.
2. The deduction of in advance commitment by Cloud users, so there allowing companies to start small and increase their hardware resources only when they are in need.
3. On a short term basis to pay for use of resources as needed (e.g., processors for an hour and storage for the day) and release them after their use, thereby rewarding conservation by letting machines and storage go when they are not useful for longer time.

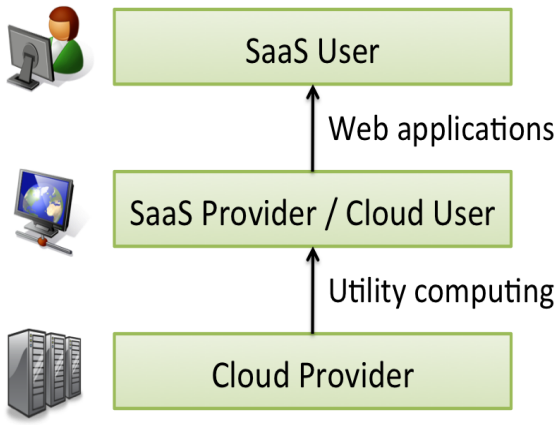

Figure 1: Users and Providers of Cloud Computing.

The benefits of SaaS to both SaaS users and SaaS providers are well documented, so it focus on Cloud Computing's effects on Cloud Providers and SaaS Providers/Cloud users. The top level can be recursive, in that SaaS providers can also be a SaaS users. For example, a mashup provider of rental maps might be a user of the Craigslist and Google maps services.

B. Analysis of Data

Big data is a word used to describe a massive volume of both structured and unstructured data that is very large which is tough to process using normal databases and software techniques. In the current enterprise scenarios the data is very big and sometimes it exceeds processing capacity. Big Data includes data sets with large sizes beyond the capacity of commonly used software tools to capture, accurate, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target, as of 2012 ranging from a few dozen terabytes to many petabytes of data in a single data set. "Big Data" is defined as "Represents the data sets with sizes beyond the capacity of current method , technology and theory to manage, capture, and process the data within a elapsed time"[4]. The definition of big data given by the Gartner: "Big Data is a high-velocity, big-volume and variety information assets that needs a new types of processing to make enhanced decisions" [5]. According to Wikimedia, "In IT, big data becomes difficult to process using normal database tools because it is a collection of large and complex data sets".

In this technology world the expansion of applications like semantic Web analysis, social network analysis, and bioinformatics network analysis, different data need to be processed continuously to witness a quick increase. Managing and analyzing the large scale of data can be interesting but it is a critical challenge. Nowadays, academic, industry and government has attracted to big data. It gives several processing techniques from both the application and system aspects. The important issues of big data processing, cloud architecture, cloud computing platform, data storage scheme and cloud database are presented from the vision of cloud data management and big data processing mechanisms [6]. Following are uploading data to the cloud, data to the HDFS and Mapreduce parallel processing framework. Finally, we discuss the techniques and challenges, and explore the future research directions on big data processing in cloud computing environments.

II. UPLOADING DATA TO THE CLOUD

Commercial DBMSs are not suitable for uploading extremely large scale data. So cloud is the best platform where it can store large amount of data.

A. System Model

Consider a cloud which has K geo-distributed data centers in a set of regions K, where K=|K|. The cloud user continuously produces massive volumes of data at set D of multiple geographical locations. The user can connect to the data centers from different locations via virtual private networks (VPNs), with G VPN gateways at the user side and K VPN gateways at the user side and K VPN gateways each collocated with a data center. The set of VPN gateways at the user side is taken as G, with G=|G|.
In fig.2 the user side has a private network inter-connects the data generation locations and the VPN gate ways at the user side [7]. This model reflects connection between user and public clouds where private network has been established between a user's premise and the cloud. In user's private network, the data transmission bandwidth between a data generation location d ∈ D and a VPN gateway g ∈ G is large as well. In user's private network, the data transmission bandwidth between a data generation location d ∈ D and a VPN gateway g ∈ G is large as well. we are moving data from different data locations to different data centers in the cloud.

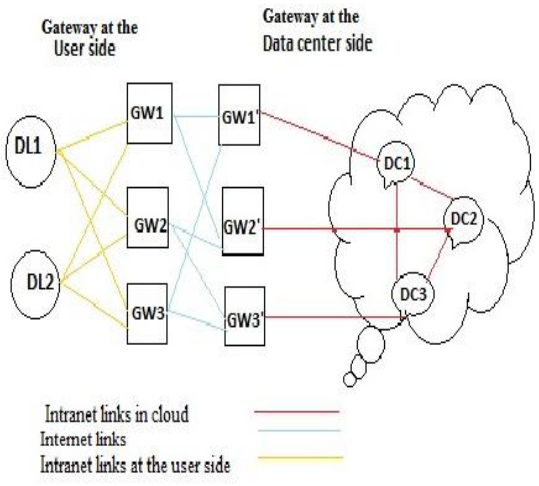

Fig.2. An illustration of the cloud system

III. PROCESSING OF DATA

Commercial DBMSs are not suitable for processing high volume of data. The present architecture's potential bottleneck is the database server while faced with pinnacle workloads. For big data processing has two important goals where every database has some restriction of cost and scalability [8]. Many models have been presented for data processing based on the application. Each provider has different kinds of applications and various business models. For example Google is more interested in small applications with light workloads whereas Azure is interested in affordable service for medium to large services. Most of the cloud service providers are using hybrid architecture which satisfies their service requirements.

A. Structuredness of data

Structured data means the data which is identified and organized in a structure. The common shape of structured data or records is database where each and every information is stored in the form of rows and columns. The easy way to search the structured data is by using data type within content. This Structured representation is easily understood by computers and for human readers. In generally the data which is stored in RDBMS, which have specific organized format.

Semi-structured data is also a structured data that does not have form of models which suits with RDMS or other data tables, but contains tags, markers to separate semantic elements and enforce hierarchies of records and fields within the data. The unstuctured data is also known as self-describing or schema less structure. Semi-structured data is becoming vast because of internet where full-text documents and databases are not in the forms of data and for exchanging this information a medium is required for different application T he forms of semi structured data are XML, other markup languages, email, and EDI. OEM (Object Exchange Model) is a means of self-describing a data structure.

Unstructured Data (or unstructured information) refers to information that either does not have a pre-defined data model and/or does not fit well into relational tables. Unstructured data is used to describe about the information which is not present in the database. Unstructured data can be text or non-text data. The examples of Textual unstructured data are email, PowerPoint presentations, Word documents, and instant messages. Non-textual unstructured data is like MP3 audio files, JPEG pictures, and Flash video files.

The management of unstructured data is recognized as one of the major unsolved problems in the information technology (IT) industry, the main reason being that the tools and techniques that have proved so successful transforming structured data into business intelligence and actionable information simply don't work when it comes to unstructured data. New approaches are necessary. To process the large scale of data we use hadoop and mapreduce framework.

B. Distribution of data in Hadoop

Hadoop infrastructure is used for batch processing and it can be also used on a single machine. The large amount of work is distributed among set of machines by using the hadoop. Data is distributed to all the nodes of the hadoop cluster as it is being loaded in. The main purpose of Hadoop Distributed File System (HDFS) is to split very large data files into chunks and these chunks are managed by different nodes in the hadoop cluster. Each chunk is replicated in three machines because if one system is fails the data can be available in the other system. The active monitoring system deals with the re-replicates of data in response to system failures [9]. Even the file chunks are replicated among several machines, their contents can be universally accessible because they form a single namespace.

HDFS is a distributed file system is designed to hold very large scale amounts of data, the data size can be terabytes or even petabytes, and to access this information HDFS provides high-throughput access. In multiple machines files are stored in a redundant fashion to have their durability to failure and available for parallel applications. For this we need to spread the data in large number of machines.

- The data stored in HDFS should be reliably because if one machine in the cluster malfunctions, the data should be available.
- HDFS should also provide scalable, fast access to the stored information. It should serve a larger number of clients by adding more and more machines to the cluster.
- HDFS should have the possibility of integrating with Hadoop Mapreduce, which allows the data to be read and computed when possible.

But the HDFS design has high performance and also restricts to a particular application and it is very scalable. The HDFS design is based on the Google File System design. HDFS is a block-structured file system where each files are split into blocks having fixed size. These fixed size blocks are stored in a cluster among three machines considering the storage capacity. The clusters in the individual machines are known as Data Nodes. A file can split into many blocks and they can be stored on different machine by choosing randomly on a block by block basis. To access a single file the attention of all machines is requires because the data block of a file is stored in multiple machines. To serve a single file several machines are involved, the file will be unavailable by the loss of any one of the machines. HDFS fight this problem by duplicating each block in various machines (by default, 3). The size of the block-structured file systems size is of 4KB or 8 KB. But the default block size of HDFS is 64MB.

The metadata of a file system should be stored securely. The data of a file is accessed in a write once and read many model and concurrently many clients can modify the metadata structures (e.g., the names of files and directories). The main thing is that information will never desynchronize. It should be handled by a single machine, known as NameNode. If a client wants to open a file he

need to contact the NameNode and takes the list of blocks location where they are stored. The DataNodes in the locations hold each block. Parallel the clients read file data directly from the different DataNode. In this bulk of data transfer the NameNode is not directly involved, because to keep its overhead to a minimum.

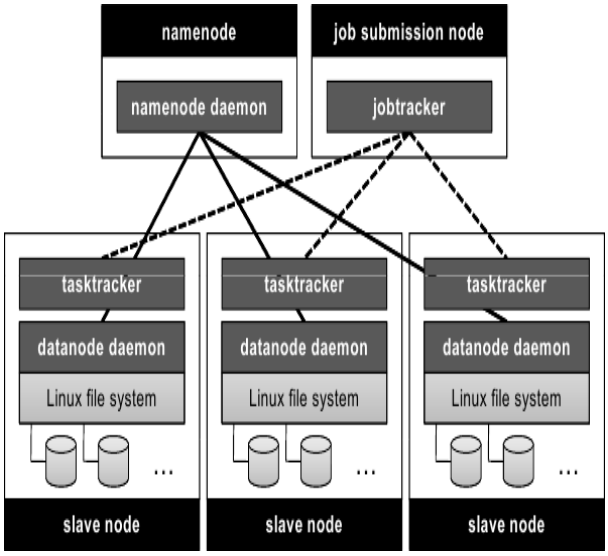

Fig. 3 The architecture of a complete Hadoop cluster is shown in Figure

In Hadoop, HDFS (Hadoop Distributed File System) supports MapReduce.

C. The Execution Framework

MapReduce is an execution framework used for very large amount data processing on clusters and it is also a programming model computation on massive amounts of data. MapReduce is developed by Google and built on well-known principles for distributed processing. MapReduce is a open-source implementation called Hadoop. A vibrant software ecosystem has sprung up around Hadoop. MapReduce and Hadoop have been designed to process huge amounts of data MapReduce can refer to three distinct but related concepts. First, MapReduce is a programming model, which is the sense discussed above. Second, MapReduce can refer to the execution framework (i.e., the "runtime") that coordinates the execution of programs written in this particular style.

A MapReduce job is to be executed whose input is all the data sources. We explicitly use Hadoop, and assume a Hadoop Distributed File System (HDFS) instantiation must be used to complete the job [10]. Therefore the data must be moved from the data sources into HDFS before the job can begin.

Fig.4 shows the general dataflow of a MapReduce job. Prior to the job starting, the delay in starting the job is in the transfer of all data into HDFS (with a given replication factor). If the source data is replicated to distant locations, this could introduce significant delays. After the job starts, the individual Map tasks are usually executed

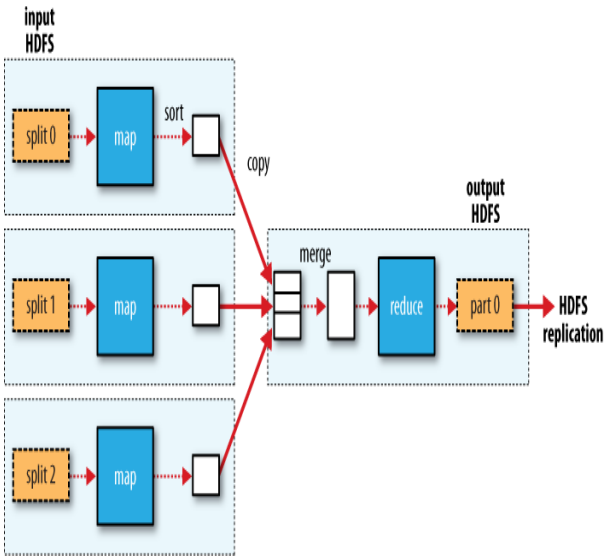

Fig 4: In the traditional MapReduce workflow, Map tasks operate on local blocks, but intermediate data transfer during the Reduce phase is an all-to-all operation.

on machines which have already stored the HDFS data block corresponding to the Map task; these are normal, "block-local" tasks. This stage does not rely on network bandwidth or latency. If a node becomes idle during the Map phase, it may be assigned a Map task for which it does not have the data block locally stored; it would then need to download that block from another HDFS data location, which could be costly depending on which data source it chooses for the download. Finally, and most importantly, is the Reduce phase. The Reduce operation is an all-to-all transmission of intermediate data between the Map task output data and Reduce tasks. If there is a significant amount of intermediate data, this all-to-all communication could be costly depending on the bandwidth of each end-to-end link. In the next section, we will propose architectures in order to avoid these potential performance bottlenecks when performing MapReduce jobs in distributed environments.

Migrating Different Applications into Clouds

The recent years have witnessed significant interest in migrating different applications onto the cloud platform. Hajjat et al. [11] develop an optimization model for migrating enterprise IT applications onto a hybrid cloud, to benefit the unlimited computing resources with a specific security guarantee. Cheng et al. [12] and Wu et al. [13] advocate deploying social media applications into clouds, for leveraging the rich resources and pay-as-you-go pricing. These projects focus on workflow migration and application performance optimization, by carefully deciding the modules to be moved to the cloud and the data caching/replication strategies in the cloud. The very first

question of how to move large volumes of application data into the cloud is not explored.

IV. ONLINE TECHNIQUES

An Online technique is one that the input events are given one by one while the output decisions are generated on the fly, based on the current input history, without any knowledge of the future input. Each decision will affect the following decisions, so that it has an significant impact on the overall performance, i.e., an online algorithm faces a situation that current optimal decision may later turn out not to be optimal due to lack of future information. Ski-rental problem and Paging problem give typical examples for the application of online technique

In contrast, an Offline Technique is given the complete input data from the beginning to the end, and produces an output to solve the whole problem at one time [19]. Competitive analysis, which compares the performance of the online algorithm to that of optimal offline algorithm, is widely used to measure the quality or performance of decision-making in online technique. Competitive ratio is essential for competitive analysis.

Two online techniques are proposed to practically guide data routing and aggregation over time. At any given time to make the choice of data centers for data processing and routes for transmitting the data there.

- Online lazy migration (OLM) Technique
- Randomized fixed horizon control (RFHC) Techniques,

The OLM relies only on the current and historical information. We design a more judicious online solution by exploring the inter-slot dependencies for data center selection.

RFHC is an advanced control technique based on the prediction of future requests. RFHC further exploits predicted information from the future. In practical applications, near-term future data generation patterns can often be estimated from history, e.g., using a time series forecasting model. This technique is that exploits such future information.

Competitive analysis [14], which compares the performance of the online algorithm to that of optimal offline algorithm, is widely used to measure the quality or performance of decision-making in online algorithm. Competitive ratio is essential for competitive analysis

V. CONCLUSION

This paper describes a flow of survey for uploading and processing of big data in the context of cloud computing and also the key issues, including cloud storage and processing of data. Two online techniques are stated to practically guide data migration in to the cloud in online fashion, for processing the data using mapreduce and hadoop framework.OLM deals with current information and RFHC deals with future information. Nowadays big data is very challenging and it is not a new concept. It calls for scalable index storage and a distributed approach to retrieve real time results. The fact is that data is too large to process conventionally. Nevertheless, during all big challenges big data will be very complex and exist continuously, which are the big opportunities for us. Significant and complex challenges need to be faced and solved in future by industry and academia.

REFERENCES

1. M. Armbrust, A. Fox, R. Grifth, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. P. A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," EECS, University of California, Berkeley, Tech. Rep., 2009.
2. Hadoop — Facebook, http://www.facebook.com/note.php?note id=1612 1578919
3. "Cloud Computing," National Institue of Standards and Technology, http://www.nist.gov/itl/cloud/index.cfm.
4. "Big Data Processing in Cloud Computing Environments" Changqing Ji∗†, Yu Li‡, Wenming Qiu‡, Uchechukwu Awada‡, Keqiu Li‡ College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China †College of Physical Science and Technology.
5. "Big data: science in the petabyte era," Nature 455 (7209): 1, 2008.
6. Douglas and Laney, "The importance of 'big data': A definition," 2008.
7. L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. M. Lau, "Move My Data to the Cloud: an Online Cost-Minimizing Approach," http://i.cs.hku.hk/~cwu/movedata.pdf, Tech. Rep.
8. D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the cloud," in *Proceedings of the 2010 international conference on Management of data*. ACM, 2010, pp. 579–590.
9. Big Data & Hadoop : Use Cases, Case Studies & Glossary
10. Hadoop: The Definitive Guide, Third Edition by Tom White.
11. M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, and S. Rao, "Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud," in Proceedings of ACM SIGCOMM, August 2010.
12. X. Cheng and J. Liu, "Load-Balanced Migration of Social Media to Content Clouds," in Proceedings of ACM NOSSDAV, June 2011.
13. Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. Lau, "Scaling Social Media Applications into Geo Distributed Clouds," in Proceedings of IEEE INFOCOM, March 2012.
14. L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. M. Lau, "Move My Data to the Cloud: an Online Cost-Minimizing Approach," http://i.cs.hku.hk/~cwu/movedata.pdf, Tech. Rep.