ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Optimized Data Transmission from Cloud to Society by Mapreduce

S.P.Prasanth[1] and P.G.Kathiravan[2]

M.Tech, Department of Information Technology, V.S.B.Engineering College, Tamilnadu, India[1]

Assistant Professor, Department of Information Technology, V.S.B.Engineering College, Tamilnadu, India[2]

ABSTRACT: The Big data is a very big issue in our current world. In every social websites and social communities are using data in the enlarged way. The data may be larger and cannot be controlled at anyway, any situation at the rate of increasing internet users. The big data is a characterized by two types they are structured and unstructured. The structured means data used in the correct way and efficient to use in the medical, Engineering and other fields, unstructured means data which can be used at unwanted issues and social communities and other sources. That type of way we call it as big data and ensure that three dimensions are volume, velocity, velocity. In this paper fully based upon the MapReduceTechnique. In this technique which can be used to optimizing the path, which is an efficient path to sending the data's and retrieve the data's from the cloud**.** In theway to present a various indulged logic to be executed in this data file, which data to be inherited over the atmosphere cannot be controlled over the technique, few of optimized data to be availing the various facilities over the data transmission. The crucial data may be enlarging over the optimized over that geo map reducing technique. The atmospherically data changed at the time by time, in this situation the data may be lose, so to use the big data concept means to control over the losses of data.

KEY WORDS– Job sequences, Geo-distribution, HDFS, Data preprocessing

I. INTRODUCTION

Big data analysis is one of the major challenges of our era. The limits to what can be done are often times due to how much data can be processed in a given time-frame. Big datasets inherently arise due to applications generating and retaining more information to improve operation, monitoring, or auditing; applications such as social networks support individual users in generating increasing amounts of data. Implementations of the popular MapReduce framework [1], such as Apache Hadoop [2], have become part of the standard toolkit for processing large datasets using cloud properties, and are provided by most cloud vendors. In short, MapReduce works by dividing input files into chunks and processing these in a series of parallelizable steps. As suggested by the name, mapping and reducing constitute the essential phases for a MapReduce job. I n the former phase, mappers processes read respective input file chunks and produce key , Val pairs called intermediate data. Each reducer process atomically applies the reduction function to all values of each key assigned to it.

1.1 GeoDistribution:

As the initial hype of cloud computing is wearing off, users are starting to see beyond the illusion of Omni-present .Computing resources and realize that these are implemented by concrete datacenters, whose locations matter. More and more applications relying on cloud platforms are geodistributed, for any (combination) of the following reasons:

(a) Data is stored near its respective sources or frequently accessing entities (e.g., clients) which can be distributed, but the data is analyzed globally;
(b) Data is gathered and stored by different (sub)organizations, yet shared towards a common goal
(c) Data is simulatedacross datacenters for availability, incompletely to limit the overhead of costly updates.

Examples of (a) is data generated by multi-national companies or organizations with sub-datasets (parts of datasets) being managed regionally [3] (e.g., user profiles or customer data) to provide localized fast access for common operations (e.g., purchases), yet that share a common format and are analyzed as a whole. An example of (b) is the US census [4] data collected and stored state-wise (hundreds of GBs respectively) but aiming at global analysis. Similar situations are increasingly arising throughout the US government due to the strategic decision [5] of consolidating

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

independently established infrastructures as well as data in a "cloud-of-clouds". (c) corresponds to most global enterprise infrastructures which replicate data to overcome failures but do not replicate every data item across all datacenters to not waste resources or limit the
Cost of synchronization [6], [7].

1.2 Job Sequences

To make matters worse, MapReduce jobs do not always come alone. Frequently, sequences of MapReduce jobs are accomplished on a given input by applying the first job on the given input, applying the second job on the output of the first job, and so on. An example is the handling of large Web caches by implementing an algorithm such as PageRank [10]. This algorithm constructs a graph that describes the inter-page relationships, which are refined using further MapReduce jobs. Another example is hash-tree generation, performed to verify the integrity of data. Each level of the hash-tree can be generated using a MapReduce job starting from leaves representing the original response. Many times distinct MapReduce jobs are applied in sequence rather than iteratively performing the same job. Whencarrying out a sequence of MapReduce jobs the number of possible execution paths increases dramatically.

1.3Geo-Map Reduce

This paper introduces G-MR, a system for efficiently processing geo-distributed big data. G-MR is a Hadoop based framework that can efficiently perform a sequence of MapReduce jobs on a geo-distributed dataset acrossmultiple datacenters. With current frameworks for "thecloud" operating only in single datacenters, G-MR thus metaphorically speaking acts much like the atmospheresurrounding the clouds.G-MR employs a novel algorithm named *data transformation graph* (*DTG*) *algorithm* that determines an optimized execution path for performing a sequence of MapReduce jobs based on characteristics of the dataset, MapReduce jobs, and the datacenter infrastructure. Our DTG algorithm can be used to optimize for either execution time or (monetary) cost. The optimized execution path determined by G-MR may be different from the optimum possible execution path. Determining the optimum possible execution path for a large dataset is hard since every possible data move has to be considered. As shown in Section 8, G-MR provides a good compromise by determining and executing an optimized execution path that performs better than commonly used execution paths. This execution path is then enforced by G-MR through a geo-distributed chain of operations consisting of geodistributed copy operations and MapReduce executions performed with Hadoop MapReduce clusters deployed in each of the involved datacenters.

II. RELATED WORK

2.1 MapReduce

A MapReduce [1] program consists of map and reducesfunctions with the following signatures.

Map<key1, val 1>→ list (<key2, val2>)

Reduce<key2, list (val2)>→list (val3)

The map function takes an input record and outputs a set of _key, val_ pairs. The reduce function accepts a set of val's for a given key, and emits sets of values (often themselves _key, val_ pairs). Execution of one of thesefunctions is called a phase. A MapReducejob includes map and reduce functions and input data, and its executioncan use any number of mappers and reducers. A set ofinput files are divided into chunks and distributed over themappers. The chunks assigned to a given mapper are calledsplits. Assignment of _key2, val2_ output pairs of mappersto reducers is based on the key key2. More precisely, apartitioning function is used to assign a mapper output tuple to a reducer. Typically, this function hashes keys (key2) to the space of reducers. The MapReduce framework writes the map function's output locally at each mapper and then aggregates the relevant records at each reducer by having them remotely read the records from the mappers. This process is called the randomstage. These entries are unsorted and first buffered at the reducer.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

```
map (String key, String value):

// key: document name; value: document contents;
map (k1, v1) → list (k2, v2)
for each word w in value: EmitIntermediate (w, "1");
(Example: If input string is ("Jesus is only God, he only lord"), Map produces {<"Jesus", 1">, <"is", 1>, <"only", 1>, <"God", 1>, <"he", 1>, <"only", 1><"lord", 1>} }
reduce (String key, Iterator values):
// key: a word; values: a list of counts; reduce (k2, list (v2)) → list (v2)
int result = 0;
for each v in values:
result += ParseInt (v);
Emit (AsString (result));
```

Once a reducer has established*all* its values *val*2, it sorts the buffered entries, effectively grouping them together by key *key*2. The reduce function is applied to each *key*2 assigned to the respective reducer, one key at a time, with the respective set of values. The MapReduce framework parallelizes the execution of all functions and ensures fault-tolerance. A *master* node surveys the execution and spawns new reducers or mappers when corresponding nodes are suspected to have failed.

2.2Hadoop Tool

The Hadoop tool is based upon the java platform which is a MapReduce implementation for large clusters. In this tool we can use only the file name as Hadoop Distributed File System (HDFS), which is used to efficient batch workloads such as those of MapReduce. In this tool follows a master-slave model where the master is implemented in Hadoop's job tracker. The master only responsible for accepting jobs and rejecting the jobs, dividing those into tasks which encompass mappers or reducers, and assigning those tasks to slave worker nodes. Each worker node runs a Task Tracker that manages its assigned tasks. In the Hadoop file system only saved by the extension of HDFS, the master and slaves work like a client and server model. What the client request to server, the server replies the sufficient of the client response, in which as workout as vice-versa. Sometimes the node has been missed the data and again request from the server, server needs the explanation for the missing node, then only server response the client request. The data request from the server to mapped out the logical expressions used to adapt the several cluster based items. In the cluster having any script error means the MapReduce technique correct the errors and replied to the efficient search results.

2.2.1 Hadoop Map/Reduce – Goals:

- Process large data sets
- Cope with hardware failure
- High throughput

2.3 Existing Work on Big Data Handling

Volley[3] is a system for automatically geo-distributing databased on the needs of an application, presupposing

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

knowledge on the placement of its data, client access at-terns, and locations. Based on this information, data will be dynamically migrated between datacenters to maximize the efficiency. Services like Volley can be used to optimize the placement of data *before* handling them with our proposed solution, thus only solving a part of the problem. Other storage systems can support geo-distributed big data in cloud environments. Walteris a key-value storage system that can hold massive amounts of geo-distributed data. COPS[6] are another storage system for geo-distributed data. These do not address actual computations over stored data. Other recently proposed storage systems like RAM Cloud can efficiently handle large amounts of data buthaveto are deployed within a data center.

In the above architecture result fully based upon the reducing techniques, the key 1, val 1, cannot be used at any situation, which the results based upon the geo-distributed data values in the sequence of jobs. The system architecture explains what are all the functions happening in the overall system, how to reduce the data lose and optimized the data transmission of the cloud computing technology. In this will be used in the architecture means easily to find the key and values.

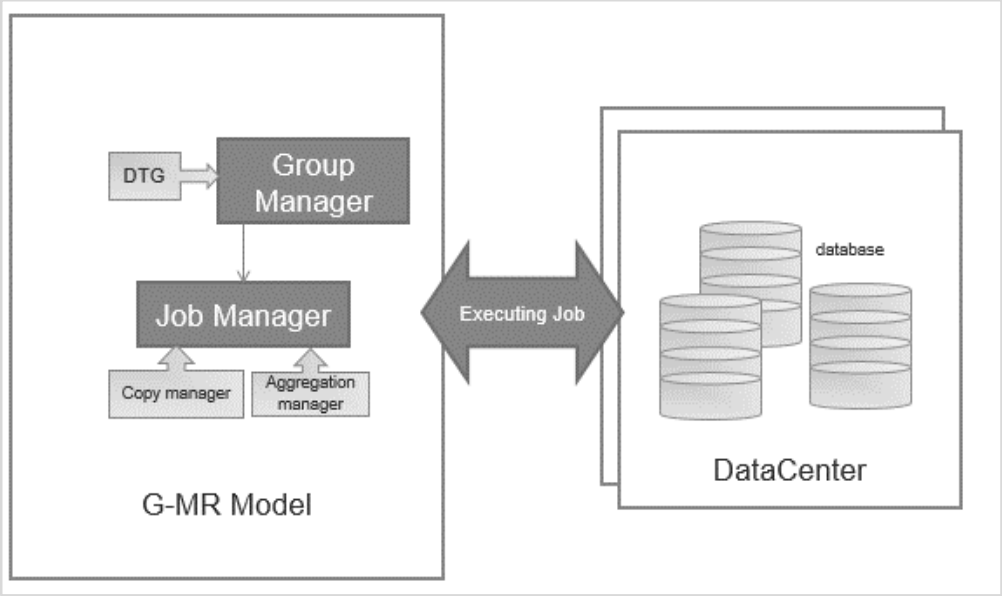

Figure 1 System Architecture of Optimized Data Transmission

III. MODEL

3.1Hadoop Tool Installation

In this model to install the Hadoop tool with the help of terminal in the Linux platform, to extract all the data's incredible of several accesses. If the data is single node means to install the double way installation, otherwise two nodes means, and single installation only needed.

3.2 Data Preprocessing

A Geo- distributed data such as earth or water related data is taken as input for processing. The dataset is analyzed and preprocessed to remove noise and unwanted fields from the dataset before applying into Hadoop for execution. Noise

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

data like invalid data, null values, etc. are identified and removed from the dataset. Our target is fixed for job execution and based on the target (output) needed, fields will be selected for processing.

3.3 HDFS Upload

Hadoop Distributed File System (DFS) will be configured for uploading the preprocessed geo data into hadoop. The configuration includes setting VM (hadoop platform) IP and port for connection. FSDataInputStream and FSDataOutputStream are used to upload and download data from hadoop. Different datacenters are analyzed for data execution across different datacenters.

3.4 Job Execution

The Group Manager executes the DTG algorithm (Novel Algorithm), this group manager is executed for determining path for performing the MapReduce job. After this the Group Manager starts to execute the determined optimized Multi-execution path. Executing individual MapReduce jobs in each datacenter on corresponding inputs and then aggregating results is defined as MULTI execution path. This path used to execute the jobs effectively.

3.5 Executing jobs

Job Managers perform the jobs accordingly using the Hadoop MapReduce clusters deployed in corresponding datacenters. The Group-Manager may also instruct a Job Manager to copy data to a remote datacenter or aggregate multiple sub-datasets copied from two or more remote datacenters. The effective job execution can be achieved using the G-MR Model.

IV. IMPLEMENTATION OF G-MR

In the analysis of data which is important one to be chosen, then only do the project effectively, the loss of data to adjusted and the involvement of errors to be corrected and the enhancement could be achieved at the end of the analysis part. I trying to enhance the data transmission and with high secure manner. And using several techniques for uploading files by splitting algorithm and merging algorithm. The multiple data centers used to store the data. And going to implement map reduce algorithm for quick transaction of data. While transferring data the data retrieved from several data centers and uploaded. The acknowledgement process used to send receives the acknowledgement for secure transaction.

V. EXPERIMENT

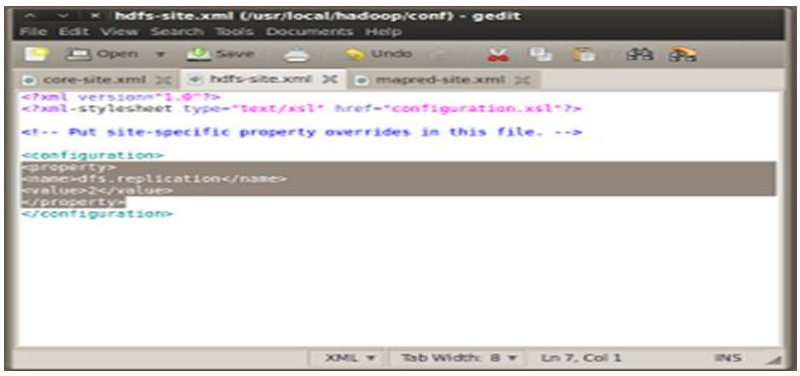

Figure 2 Hadoop Configuration

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

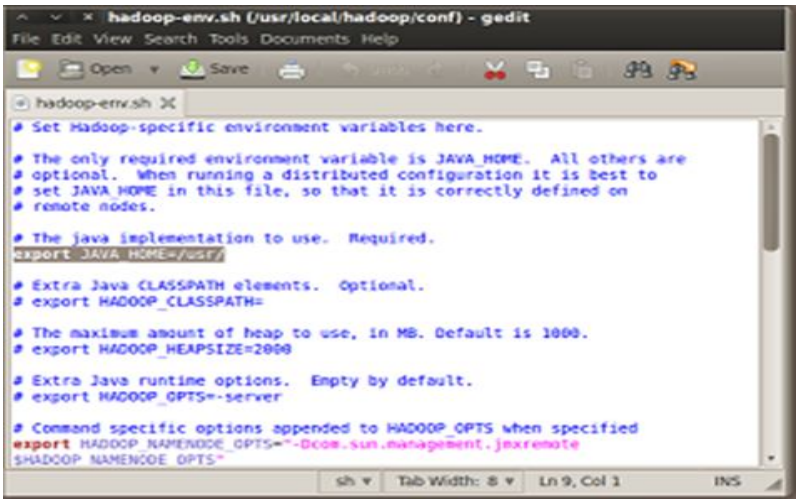

Figure 3 Data Preprocessing

VI. FUTURE WORK

For enhancement, we will implement PACT (Parallelization Contracts).The programming abstraction for writing tasks are Parallelization Contracts (PACTs), consisting of Input and Output Contracts. The programming model has the right level of abstraction for writing data processing tasks. In the analysis of data which is important one to be chosen, then only do the project effectively, the loss of data to adjusted and the involvement of errors to be corrected and the enhancement could be achieved at the end of the analysis part.

VII. CONCLUSION

This system proposes the G-MR model which effectively optimizes the MapReduce jobs. In This System MapReduce framework that can efficiently execute a sequence of MapReduce jobs on geo-distributed datasets.
The algorithm used which looks for the most suitable way to perform a job sequence, minimizing either execution time or cost. Future using this system to establish an efficient data enhancement with the MapReduce jobs.

REFERENCES

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in OSDI, 2004.

[2] Apache Software Foundation, "Hadoop," http://hadoop.apache.org.

[3] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated Data Placement for Geo-Distributed Cloud Services," in NSDI, 2010.

[4] "Data from Year 2000 US Census," http://aws.amazon.com/datasets/Economics/2290.

[5] "Department of Defense Information Enterprise Strategic Plan 2011- 2012," http://dodcio.defense.gov/docs/DodIESP-r16.pdf

[6] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen, "Don't Settle for Eventual: Scalable Causal Consistency for Wide- area Storage with COPS," in SOSP, 2011.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

[7] D. Cutting and E. Baldeschwieler, ``Meet Hadoop,'' OSCON, Portland, OR, USA, 25 July 2007 (Yahoo!)

[8] R. E. Brayant, "Data Intensive Scalable computing: The case for DISC," Tech Report: CMU-CS-07-128, http://www.cs.cmu.edu/~bryant

[9] D. Agrawal, S. Das and A. E. Abbadi, "Big Data and Cloud Computing: Current State and Future Opportunities" ETDB 2011, Uppsala, Sweden

[10] D. Agrawal, S. Das and A. E. Abbadi, "Big Data and Cloud Computing: New Wine or Just New Bottles?" VLDB 2010, Vol. 3, No. 2

[11] F. J. Alexander, A. Hoisie and A. Szalay, "Big Data" IEEE Computing in Science and Engineering journal 2011

[12]ChamikaraJayalath, Julian Stephen, Patrick Eugster"From the Cloud to the Atmosphere: Running MapReduce across Datacenter" IEEE Computing in Parallel and distributed Systems, IEEE 2013.

,

[14] Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman "Volley: Automated Data Placement for Geo-Distributed Cloud Services"www.usenix.org/event/nsdi10/tech/full_papers/agarwal.pdf,

[15] Wyatt Lloyd, Michael J. Freedman, Michael Kaminskyy , and David G. Andersenz "Don't Settle for Eventual: Scalable Causal Consistency for Wide-Area Storage with COPS" http://www.cs.cmu.edu/~dga/papers/cops-sosp2011.pdf,

[16] Michael Isard ,"MihaiBudiu, Yuan Yu, Andrew Birrell, Dennis Fetterly, "Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks" research.microsoft.com/pubs/63785/eurosys07.pdf,