ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

Optimization of Storage and Retrieval of Big Data

Rajesh B K [1], Puja Kumari [2], Prema C.R [3]

P.G. Student, Department of Information Science and Engineering, Sri Krishna Institute of Technology, Bangalore, Karnataka, India[1]

P.G. Student, Department of Information Science and Engineering, Sri Krishna Institute of Technology, Bangalore, Karnataka, India[2]

Assistant Professor, Department of Information Science and Engineering, Sri Krishna Institute of Technology, Bangalore, Karnataka, India[3]

ABSTRACT: In the present, the world has moved to the cloud for storage and retrieval of data. Every second millions of users access cloud to upload or download files which effect the rate of data transfer. Many mechanisms has been introduced to optimize data transfer rate but still the expected results has not been achieved. In this paper we propose to optimize data transfer rate through pipelining, parallelism and concurrency along with MapReduce Model. This work proposed an efficient traffic aware separation and aggregation to minimize network traffic, hence provide a better traffic management . Implementation of this technique will outperform the most popular data transfer tools like global online and UDT in majority of the cases.

KEYWORDS: Pipelining, Parallelism, Concurrency, MapReduce Model, Network Traffic, Traffic Management.

I. INTRODUCTION

Big Data is a combination of enormous and complex data that is difficult to capture, store, process, analyze and retrieve. Big Data is all about combination of large amount of structured, unstructured and semi-structured data that has the potential to be extracted for information.

Every second millions of data is uploaded by users and enterprises. This creates huge network traffic. Downloading from cloud is also a challenging task as retrieving data from Big Data is not easy. Downloading is more difficult in case of the data is unstructured or worse, semi-structured.

Earlier work proposes techniques such as Data Intensive distributed computing[9], which provides hints on how to manage low level data handling issues where performing data intensive distributed computing. But this technique's efficiency in terms of performance is not optimal. Hadoop has the ability to analyze the data very quickly and in a very cost effective way. Hadoop is suitable for structured and semi-structured data.

In the proposed system, the file will be uploaded through optimized pipelining, parallelism and concurrency. The file uploaded through these parameters will be the input into the server via gateway. The gateway consists of a log manager that preserves all the user derails who are uploading and retrieving the file from the server. Other than the main server the remaining three servers will be created stores the duplicate files. So , whenever the main server will be busy or file is blocked due to data traffic, duplicate file that were stored in the other servers are retrieved and given to the user as an output. Thus using the combined optimization technique for both uploading and downloading the data transfer rate will efficiently enhanced.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

II. RELATED WORK

A set of models were proposed, which aim to approximate the optimal number of streams with least history information and lowest prediction overhead[1].it suffered from Inefficient handling of data. The other papers [2]provided hints on how to manage low level data handling issues when performing data intensive distributed computing and [3] calculates the optimal number of streams as well as optimal number of CPU stripes to increase efficient data transfer . But the performance was close to optimal and overhead was present.[4]Calculates the smallest amount of sampling data required to find the file transfer throughput, but these parameters were poorly tuned.[5]Different parameters play a significant role in affecting achievable transfer throughput. Different algorithms are used to tune protocol parameters for data transfer in wide area networks.

Most of the current works use MapReduce performance by optimizing data transmission. The proposed MapReduce with communication overlap (MaRCO) to achieve nearly full overlap via the novel idea of including the reduce in the overlap In Map Reduce. Unfortunately, this overlap is insufficient in shuffle-heavy MapReductions.MapReduce lazily performs reduce computation only after receiving all the map data. The Eager reduce processes partial data from some map tasks while overlapping with other map tasks' communication. MaRCO's approach of hiding the latency of the inevitably high shuffle volume of shuffle- heavy Map Reductions is fundamental for achieving performance. MaRCO achieves 23% average speedup over Hadoop for shuffle-heavy MapReductions[8]. The new method of computation is characterized by the heavy use of user-defined functions for data processing. The data-shuffling stages help to prepare them for data partitions in order to compute in parallel. They have assessed this optimization opportunity on over 10,000 data parallel programs used in production SCOPE clusters, and a framework is designed that is implemented in the production system. Experiments with real SCOPE programs on real production data have shown that this optimization can save up only to 47% in terms of disk and network 110 for shuffling, and up to 48% in terms of cross-pod network traffic. To defeat this inadequacy issue of enormous information [6] . They have composed a calculation to calendar operations in view of the key dissemination of moderate key/esteem sets to enhance the heap parity [7].

III. SYSTEM MODEL

3.1 UPLOAD

- The user first registers their particulars in the cloud.
- The user then uploads the heterogeneous file to transfer.
- Using recursive chunk algorithm, it check the file size. If the file size is large it chooses either parallelism or concurrency. If the file size is small it chooses pipelining.
- During transferring, it checks the file size for each file. If the file size is small, it chooses pipelining.
- In pipelining, the files are transferred sequentially i.e. it transfers file one by one to avoid bottleneck problem.
- If the chunk size is large it either chooses parallelism or concurrency.
- In parallelism, a single file is split into small packets. After splitting, it transfers the packets parallel.
- In concurrency, it chooses different file size and transfers the file simultaneously. The designated user receives the heterogeneous file through pipelining, parallelism and concurrency.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

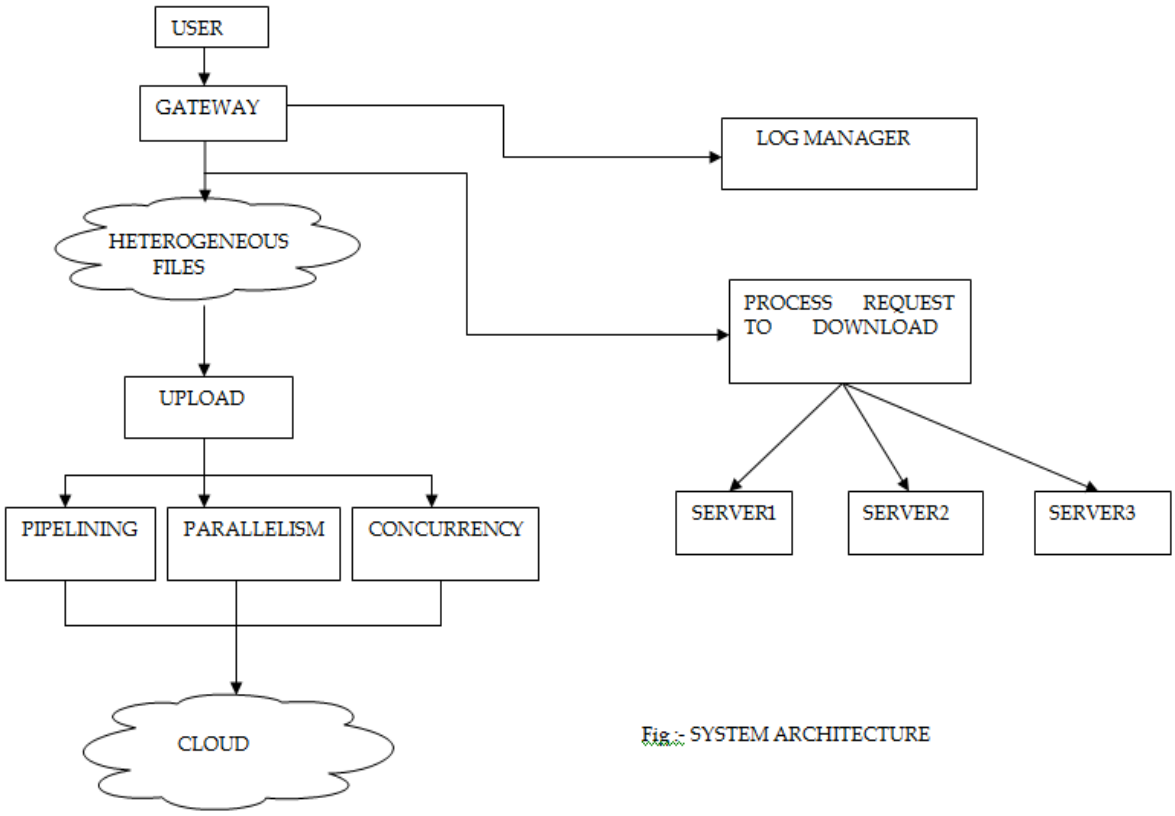

Fig :- SYSTEM ARCHITECTURE

3.2 DOWNLOAD-

- The user inputs the file into the server via the gateway.
- The gateway consists of a log manager that preserves all the user details who are uploading as well as retrieving the data from the servers.
- Other than the main server we create another two servers in which the duplicated data gets stored automatically.
- The Hadoop distributed file system approach is used here.
- The user sends the message to the gateway requesting for a particular file.
- The main server is the place from where the file gets retrieved.
- When the main server from which the file has to be retrieved gets blocked due to the data traffic that occurs, the other servers play the role of the main server.

Hence the duplicate file that were stored in the other servers are retrieved and given to the user as an output

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 12, July 2017

IV. CONCLUSION

Using our proposed method, the file will be uploaded through by the use of various optimized methods like pipelining, parallelism and concurrency. The file uploaded through these parameters will be the input into the server via gateway. By the use of MapReduce method other than the main server the remaining three servers will be created to store the duplicate files. Hence , whenever the main server will be busy or file is blocked due to data traffic, duplicate file that were stored in the other servers are retrieved and given to the user as an output. The log manager at the gateway stores the details of the users who have uploaded the file as well as downloaded the file. Thus, using the combined optimization technique for both uploading and downloading the data transfer rate is efficiently improvised.

REFERENCES

[1] E. Yildirim, D. Yin, and T. Kosar, "Prediction of optimal parallelism level in wide area data transfers," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 12, 2011

[2] E. Yildirim, M. Balman, and T. Kosar, "Data-aware distributed computing," in Data-Intensive Distributed Computing: Challenges and Solutions for Large-Scale Information Management. Hershey, PA, USA: IGI Global, 2012.

[3] E. Yildirim and T. Kosar, "End-to-end data-flow parallelism for throughput optimization in high-speed networks," J. Grid Comput., vol. 10, no. 3, pp. 395–418, 2012.

[4] E. Yildirim and T. Kosar J.Kim, "Modeling throughput sampling size for cloud hosted data scheduling and optimization service," future generation computer system, vol.29,no.7,pp.1795-1807,2013.

[5] E. Arslan, B. Ross, and T. Kosar, "Dynamic protocol tuning algorithms for high performance data transfers," in Proc. 19th Int. Conf. Parallel Process.,2013, pp. 725–736.

[6] Jiaxing Zhang, Hucheng Zhou, Rishan Chen, Xuepeng Fan ,Zhenyu a Guo,Haoxiang Lin,Jack Y. Li ,Wei Lin_ Jingren Zhou_ Lidong Zhou "Optimizing Data Shuffling in Data-Parallel Computation by Understanding User-Defined Functions "In Microsoft Research Asia Microsoft Bing tPeking University &Huazhong University of Science and Technology_ §Georgia Institute of Technology

[7] L. Fan, B. Gao, X. Sun, F. Zhang, and Z. Liu, "Improving the load balance of mapreduce operations based on the key distribution of pairs," arXiv preprint arXiv:1401.0355,2014.

[8] F. Ahmad, S. Lee, M. Thottethodi, and T. Vijaykumar, "Mapreduce with communication overlap," pp. 608-620, 2013.

[9] S.-C. Hsueh, M.-y' Lin, and Y.-C. Chiu, "A load-balanced mapreduce algorithm for blocking-based entity- resolution with multiple keys," Parallel and Distributed Computing 2014, p. 3,2014.