Reusing Results in Bigdata Frameworks

Nruthya Bojamma K.D
Dept. of CSE
GSSSIETW, Mysuru

Supriya D
Dept. of CSE
GSSSIETW, Mysuru

Supriya P
Dept. of CSE
GSSSIETW, Mysuru

Sushma A.P
Dept. of CSE
GSSSIETW, Mysuru

Rajashekar M B
Associate Professor, CSE
GSSSIETW, Mysuru

***Abstract* – Big Data refers to large and complex data sets. Big Data can use parallel data processing application software and traditional approach is not possible. Gathering a Data, processing and survey are one of the most important factors in organizations, companies and academic institutions. We propose a framework in this paper which can efficiently detect and avoid redundant computations. The important technique used here is the materialized view technique which saves and reuses the result of previous computations[1].**

Keywords –Big Data, reuse results, materialized view technique

I. INTRODUCTION

The amount of information that businesses and organizations can store and analyze is rapidly increasing. It requires processing of large amount of data because there is rapid growth in it, this large processing of data is often referred as Big Data.

Operations, business decisions, product recommendations and numerous other everyday tasks are increasingly relying on processing and analyzing large datasets of diverse formats and heterogeneous sources. The need for using Big Data by the non- experts and others analysts, rapidly led to the improvement and adaptation of high-level, dataflow systems for data analysis[2]. It is getting increasingly popular to use parallel processing systems for large scale data processing. But due to the explosion of data, old approach could not complete the data analysis task efficiently and economically.

Companies such as Yahoo and Microsoft started the research for parallel processing systems, to deal with large datasets. These systems have helped to increase the throughput of data analysis tasks, comparing to traditional solutions. However, the parallel processing systems also consume huge amount of resources as huge amount of nodes and are used as a single task which can be executed. To improve, computation efficiency and resource utilization, it is necessary to optimize the computation tasks. Optimization is always an important topic in data processing systems.

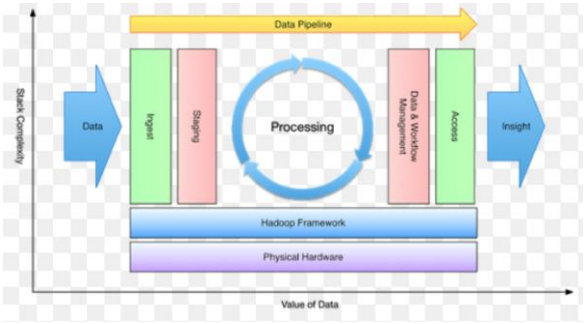

Figure 1.1: Architecture of Big Data[5]

Since the cost for task execution and data storage is very expensive, considerable savings can be obtained by optimization. Avoiding computation redundancy can save a lot of time and money. In past few years Big Data analysis has been a important research.It is hard to execute the data analysis task in older approach than in parallel data processing.Parallel processing platforms and parallel dataflow systems running on top of them are increasingly popular. They have greatly improved the throughput of data analysis tasks. The trade-off is the consumption of more computation resources. Nodes are combined together to execute a single task. However, it might still take hours or even days to complete a task. It is very important to improve resource utilization and computation efficiency. According to research conducted by Microsoft, there exist around 30% of common sub-computations in usual workloads. Computation redundancy has time and resource wastage

II. RELATED WORK

❖ Optimization based on Map Reduce**:**

Map Reduce is one of the most popular parallel processing platform.MR Share is a concurrent sharing framework[3].Functions are similar to multi-query optimization and avoid computation redundancy for a batch of queries executed at the same time.The system architecture consists of the following important steps[4]:

- Upload the medical images to HDFS.
- Take a medical image from HDFS and input it as Mapper.
- Extract the image features. Write the image and features in HBase.

- Complete the image processing in HDFS.

Collect the output of Map Reduce phase[5].

1) The user sends a query image to system, and then the image will be stored temporarily in HDFS.
2) Run a map-reduce job to extract features from query Image.
3) Store image features in HDFS.
4) The similarity/distance between the features vectors of the query image in HDFS and the target images in the HBASE are computed.
5) A reducer collects and combines all the result from all the map function.
6) The reducer stores the result into HDFS.
7) Send the result to the user.

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware[6]. All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines, or racks of machines) are commonplace and thus should be automatically handled in software by the framework. The core of Apache Hadoop consists of a storage part (Hadoop Distributed File System (HDFS)) and a processing part (MapReduce)[7]. Hadoop splits large files into large blocks and distributes them amongst the nodes in the cluster.

Pig is a platform for analyzing large data sets that consists of high level language[2].To process the data, Hadoop MapReduce[8] transfers packaged code for nodes to process in parallel, based on the data each node needs to process. This approach takes advantage of data locality nodes manipulating the data that they have on hand-to allow the data to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel system where computation and data are connected via high-speed networking. The base Apache Hadoop framework[9][13] is composed of the following modules: Hadoop Commonly contains libraries and utilities needed by other Hadoop modules. Hadoop Distributed File System (HDFS)[10] - a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster; Hadoop HBase-HBase[11] is a column-oriented database management system that runs on top of HDFS. Hadoop MapReduce-A programming model for large scale data processing[12].

❖ Optimization based on Dryad /DryadLINQ[14]:
Dryad/DryadLINQ is a parallel processing system developed by Microsoft. Dryad is a distributed execution engine to process data-parallel applications and DryadLINQ is a high-level language. A Dryad application is expressed as a directed acyclic graph[12] where each vertex is a program and each edge represents a data channel. DryadLINQ would translate the queries into plans that can be executed by Dryad[14]. It processes a query with the following steps: (1) Translate the query into a logical plan ;(2) Transform logical plan to physical plan; (3) Encapsulate physical plan to Dryad execution graph. Microsoft has conducted a series of research on query optimization. Two kinds of redundant computations are identified[15]: input data scans and common sub-query computations. Redundant scanning contributes to around 33% of the total I/O; while 30% of the sub-queries are detected to have a match[16][13](have same input and common computation).
The computation redundancies are also detected[17]. To solve this problem, Microsoft developed the DryadInc system, the Nectar system and the Comet System for query optimization. They are all built upon the Dryad/DryadLINQ System. And they assume the file system is append-only[18].

III. METHODOLOGY

The basic objective of this work is building a framework using the technique called materialized view technique which saves and reuses the results of previous computations[1][7].
The main reason of using materialized view technique is that basically views logically exist unlike tables. If we want to hide certain columns to users we cannot do using tables, creating a view we can achieve security.

i. Proposed Work

In Proposed system, we are building a framework to identify nouns and pronouns of Kannada language. It includes access to user and admin. User functions includes analyze NER, generate pdf, send mail, upload doc and admin functions include check users, approve users, approve document.
Also the user will give lot of large files as input to the system which will put all those to database. When the query is executed in order to identify certain things like noun/ pronoun in this project, it will check if it has been run previously.
The input in this project is paragraph of text which will be given by the user and the output is identifying the noun and pronoun out of given the paragraph of text.

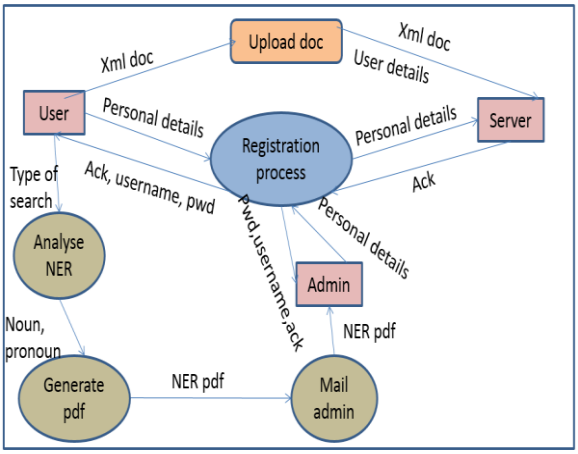

Figure 1.2:Architecture of proposed work[1][20]

The above architecture comprises of three main entities that are User, Admin and Server.

User: Initially the user register by giving his/her personal details like name, password and Email-Id in the registration process. After validating the name, password and Email-Id the user gets an acknowledgement for the registration. The user enter the type of search that is whether noun or pronoun. After the generation of the analyzed pdf the user is required to send it to the admin for verification. If the admin approve/allow that pdf to upload then user need to upload that respective pdf file to the server for future use.

Admin: Initially the admin also register by giving the personal details as user and gets an acknowledgement for the registration. The admin verify the pdf sent by the user, if it is valid or correct then the admin allow user to upload otherwise it will not allow the user to upload.

Server: The server verifies the personal details of user/admin and sends an acknowledgment. The server analyzes the given paragraph of text in order to identify the given type of search it may be either noun/pronoun. It also generates a result file which is in pdf format called NER pdf. The file which is uploaded by the user after taking the approval for the same from the admin is stored in the server which can be reused for future purpose.
Also in the diagram there is an indirect connection from the server to the Analyze NER and Generate pdf.

ii. Installation Details

For implementing this project the server that is required is WAMP:
W: Windows is an operating system.
A: Apache is a server.
M: MySQL is the database that is used to store the data.
P: PHP is a programming language used to write the code.
Design tools: Back end is designed using the eclipse and frond end is designed using HTML, CSS, Java script and JQuery. The database that is used to store all the uploaded documents/files is PostgreSQL.

iii. Requirements

Functional Requirements:

Software Requirements

- Operating system: Windows
- Technology :Java

Hardware Requirements

- Processor: Pentium
- RAM: 2GB minimum
- Hard disk drive: 40GB

Non Functional Requirements:

- Security
- Adaptability
- Performance
- Scalability
- Manageability
- Capacity
- Availability

IV. CONCLUSION AND FUTURE ENHANCEMENT

This project enables to identify the nouns and pronouns of Kannada language which are part of parts of speech in Kannada grammar from a large dataset which is given by the user. The technique that we are using is materialized view technique which provides more security. It avoids computation redundancy thereby lot of time and money can be saved.
Future work: This project can be enhanced to other language also like Hindi, English and so on. Also can search other parts of speech like verb, adverb, adjective, preposition, conjunction and interjection from the given paragraph of text or large set of data.

REFERENCES

[1] J. Yang, K. Karlapalem, and Q. Li, "Algorithms for materialized view design in data warehousing environment," in Proceedings of the 23rd International Conference on Very Large Data Bases, San Francisco CA, USA: Morgan Kaufmann Publishers Inc.
[2] A. F. Gates, O. Natkovich, S. Chopra, P. Kamath, S. M. Narayanamurthy, C. Olston, B. Reed, S. Srinivasan, and U.Srivastava "Building a high-level dataflow system on top of map-reduce: the pig experience," Proc. VLDB Endow., vol. 2, no. 2, pp. 1414–1425
[3] I. Elghandour and A. Aboulnaga, "Restore: reusing results of map reduce jobs," Proc. VLDB Endow., vol. 5, no. 6, pp. 586–597, Feb. 2012
[4] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin "Incoop: Mapreduce for incremental computations," in Proceedings of the 2nd ACM Symposium on Cloud Computing, ser. SOCC '11. New York, NY, USA: ACM, 2011, pp. 7:1–7:14.
[5] Y. Chen, S. Alspaugh, and R. Katz, "Interactive analytical processing in big data systems: a cross-industry study of map reduce workloads" Proc. VLDB Endow., vol. 5, no. 12, pp. 1802–1813, Aug. 2012
[6] Prof.DeeptiChikmurge," Implementation of CBIR Using Map Reduce Over HADOOP", International Journal of Computer, Information Technology Bioinformatics (IJCITB) June 2014
[7] WichianPremchaiswadi, AnuchaTungkatsathan, SarayutIntarasema, NuchareePremchaiswadi, "Improving Performance of Content-Based Image Retrieval Schemes using Hadoop Map Reduce." (IJCITB) June 2014. 2008.
[8] Dr. Siddaraju, Sowmya C L, Rashmi K, Rahul M ―Efficient Analysis of Big `Data Using Map Reduce Framework‖ International Journal of Recent Development in Engineering and Technology (ISSN 2347-6435(Online) Volume 2, Issue 6, June 2014
[9] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In OSDI, 2004.
[10] J. Dean and S.Ghemawat. Map Reduce: Simplified data processing on large clusters. In Proc. OSDI, 2004.
[11] H. Herodotou and S. Babu. Pro_ling, What-if Analysis, and Cost-based Optimization of Map Reduce Programs. In VLDB 2011.
[12] GONZALEZ, J. E., LOW, Y., GU, H., BICKSON, D., AND GUESTRIN, C. Power graph: Distributed graph-parallel computation on natural graphs. OSDI'12, USENIX Association, pp. 17–30.
[13] Hu Yafei, Li Fangmin, Liu Xinhua. CPS: Network System Framework and Key Technologies [J]. JOURNALOF COMPUTER RESEARCH AND DEVELOPMENT.2010,47**(z2)**: 304-311.

[14] DEWITT, D., AND GRAY, J. Parallel database systems: The future of high performance database processing. Communications of the ACM *36*, 6, 1992.
[15] ISARD, M., BUDIU, M., YU, Y., BIRRELL, A., AND FETTERLY,D. Dryad: Distributed data-parallel programs from sequentialbuilding blocks. In Proceedings of European Conferenceon Computer Systems (EuroSys), 2007.
[16] YU, Y., ISARD, M., FETTERLY, D., BUDIU, M., ERLINGSSON,Ú., GUNDA, P. K., CURREY, J., MCSHERRY, F., AND ACHAN,K. Some sample programs written in DryadLINQ. Tech. Rep.MSR-TR-2008-74, Microsoft Research, 2008
[17] Y. Yu, M. Isard, D. Fetterly, M. Budiu, U. Erlingsson, P. K. Gunda, and J. Currey, "Dryadlinq: a system for general-purpose distributed data parallel computing using a high-level language," in Proceedings of the 8th USENIX conference on Operating systems design and implementation ,ser. OSDI'08. Berkeley, CA, USA: USENIX Association, 2008,pp. 1–14.
[18] L. Popa, M. Budiu, Y. Yu, and M. Isard. DryadInc: Reusing work in large-scale computations. In HotCloud, 2009.
[19] Y. Yu, M. Isard, D. Fetterly, M. Badiu, U. Erlingsson, P. K. Gunda, and J. Currey. DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language. In Proc. OSDI, 2008.
[20] A. Gupta and I. S. Mumick, "Materialized views," A. Gupta and I. S. Mumick, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Maintenance of materialized views: problems, techniques, and applications, pp. 145-157.