A SURVEY ON BIG DATA PROCESSING

[1]Santosh Kumar J., [2]Raghavendra B. K., [3]Meenakshi.
[1]Associate Professor, [2]Head & Professor , [3]Assistant Professor
[1]Computer Science and Engineering,
[1]KSSEM, Bengaluru, India

Abstract : Big data is the data which is not able to store and process by our traditional system. There are many tools now a days to store and process big data each and every device generates the lot of data every seconds nearly data generation is growing like anything past many years lot of data generated but not analyzed and used it now companies like amazon use the analysis on stored data and number one company in the world. So data analysis is very important to grow or development big data is not only about huge data along with that variety and velocity of data the speed at which data generated is also one of the challenges to store and manage the streaming data. And data is not of uniform in nature variety of data like structured whose fields and data types are known, semi structured whose data types are not known but fields are known and unstructured whose data types and fields are not known. to process Hadoop had very important components basically it has main two components like HDFS and Map Reduce to store and process big data on top of that the component YARN added for resource management on top of Hadoop framework many components like H-base, hive, pig, zoo-keeper, Oozie, flume and many are added to make user friendly and improve the performance one of the components which overcome drawbacks of Hadoop is spark and spark itself have many drawbacks to overcome one more framework flink added which also have many drawbacks which will be stated and tried to give solutions
Here we ran wordcount for the sample input file with single mapper reducer and multiple reducers with Pig script without orderby and Hive with order by and we achieved enhanced performance with hive with order by along multiple reducers.

IndexTerms **– Hadoop,HDFS,MapReduce,Spark,Flink.**

I. INTRODUCTION

Big data is the Data generated by or for the humans by humans and devices.Every secondPetta bytes of data generating around us, many devices like face book twitter sensors devices of IoTgenerates huge amount of data every seconds.Generated data if we analyze and make a knowledgeable out of it will definitely benefits the society, mankind and organizations. Big data is not only the huge data but not able to store and process by our traditional system. There are many tools available to store and process big data.Each and every IoTdevice generates the lot of data, data generation is growing like anything past many years lot of data generated but not analyzed used it,and now companies like amazon used the data for analysis and became number one company in the world. So data analysis is very important to enhance the organization, big data is not only about huge data along with that variety and velocity of data the speed at which data generation is one of the challenge to store and also manage the streaming data is one more challenge. Data is not of uniform in nature variety of data like Un-structured, semi-structured,and structured whose data types and fields are not known. to process variety of data Hadoop have very important components basically it has main two components like HDFS and Map Reduce to store and process big data on top of that the component YARN added for resource management on top of Hadoop framework many components like Hbase, hive, pig, zoo-keeper, oozie, flume and many are added to make user friendly and improve the performance one of the components which overcome drawbacks of Hadoop is spark and spark itself have many drawbacks to overcome one more framework flink added which also have many drawbacks which will be stated and tried to give solutions. Big data and cloud are the very much influence on current IT industry, which have massive storage and computation power which will give user to deploy user application without using infrastructure. To process huge amount of data map-reduce framework is widely used with cloud computing which is very flexible scalable and cost effective. Examples are cloudex-lab, amazon Web services elastic map reduce components also amazon cloud added many components like mahout Oozie zookeeper pig hive and many more as shown in fig 1. In amazon user can invoke or create cluster to process the computation web services are charged according to usage pay as use. Machine learning is subject of Artificial intelligence which trains the machine and machine will train itself for making more intelligence for specific application, which allows improving their computational output based on previous input or learns with past errors. With thread parallelism big data computing performance may be improved.

Attributes of Big Data

1. Variety – They are collecting data from various sources (human and machine) and include (not exhaustive)-social media, credit card usage, website visits, retail shops, hospitals, mobiles, sensors, log files, security cameras etc.
2. Volume – The volume of data generated every day is humongous. IBM estimates that 2.5 quintillion bytes of data is created each day
3. Velocity – The data is being generated in gigabyte scale every second given the improvement in bandwidth.

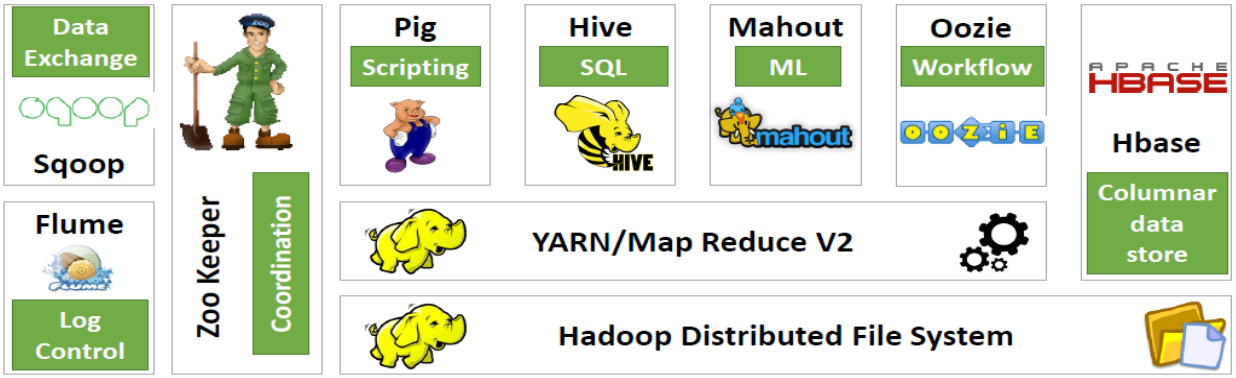

FIG 1: BIG DATA FRAMEWORK OVERVIEW

FLUME AND SQOOP

Sqoop and Flume are the frame work for import and export of Data from various Sources to HDFS and vice versa. Sqoop for the structured data that is From RDBMS to Hadoop and vice versa whereas Flume is for unstructured data and streaming data import and export to and from HDFS to other sources of data.

SPARK OVERVIEW.

Spark is the framework like any other java framework which resides on top of OS to utilize memory of OS and other devices efficiently specially designed framework for big data. Spark has many advantages and disadvantages Memory management is one of the disadvantage of spark whereas processing big data is advantages compared with map reduce framework of Hadoop.

FLINK OVERVIEW.

Flink is one in all. Flink is the frame work for Streaming data, latency which we had in spark for processing is overcome by flink, it processes the data without delay like speed of light, Memory exception problem is also solved by flink.it can also interact with many devices with different storage system to process their data, and it also optimizes the program before execution.

Steps to execute flink program

1. Map will take one input element and gives 1 output element.
2. Flat Map which takes 1 input element and gives many output elements.
3. Then check for Boolean expression for true others are discarded.
4. Then partitions the data in to disjoint and uses Union, Split, Join and select many operators to process the data.

Disadvantages of Flink

1 Memory management in flink have some problem due to pipeline execution of program

2 Data representation in flink is problem due to raw bytes data usage.

Big data has many performance measurement benchmarks programs as soon we install Hadoop we have test the performance of Hadoop with bench mark programs like Terra Gen, Terra sort, Terra validate, Word count, Pi calculation and many more Bench mark applications already installed with Hadoop we just have to Run a Jar file of Bench marks tools to measure the Performance.

II. LITERATURE REVIEW

Author of the paper said about apache Hadoop that it is a framework for processing large distributed data set across cluster of computers and said about scaling the cluster [1].

Due to use of sensors across all devices and network tools of the organizations generating big data, all wanted to store and analyze without investing much cost on managing and service issue of the storage and processing want to deploy everything on cloud so that cloud management organizations will take care of it, these companies can utilize the data for analysis and extract useful knowledge out of it [2].

Map Reduce is the framework which allows large data to be stored across all devices and processed by devices map functions will distribute the data and store across the devices where a reduce will process the query of the client it works on bases of the key value pair. Each line will be treated as key and value that is first word is the key and rest all will be value whenever client request to process the large data first client will approach the name node name node will respond to client with available free nodes after that mapper functions by client will write data to respective data nodes, and whenever client want to process the data it request to name node job tracker then job tracker will communicate to name node to get data information storage then it will assign jobs to task tracker to process the job by name nodes will process the task by their available data then one of the node will aggregate the result and give the result to client[3].

III. RESULTS AND DISCUSSION

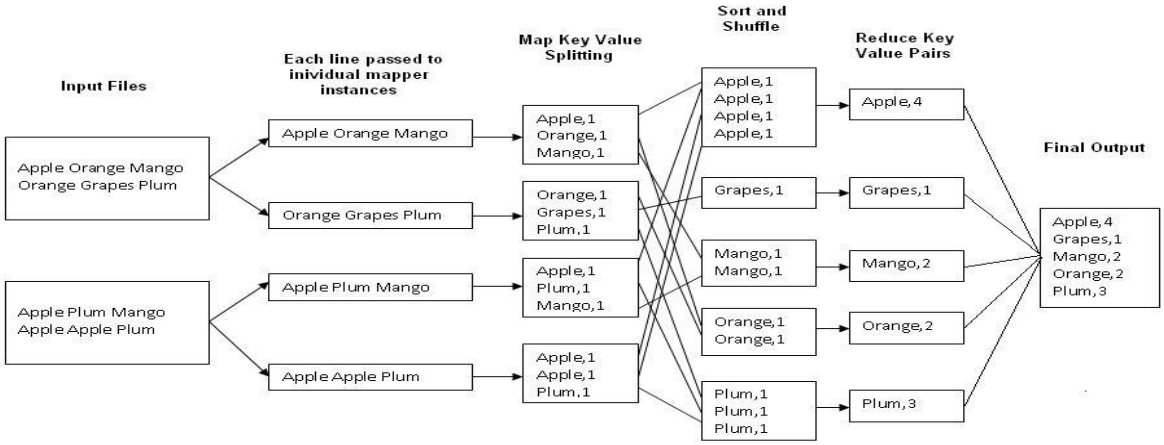

Figure 2: Map Reduce framework for word count.

Fig 2 above is the Map reduce framework for word count where input file is split as lines each lines again separated by spaces to get words then words are shuffled with all name nodes mappers then count the each words in each nodes then using reduces combines the results .

Fig 3,4,5,6 shows the execution of word count Pig script with number of Mapper, Reducers, Order by and sampling methods.

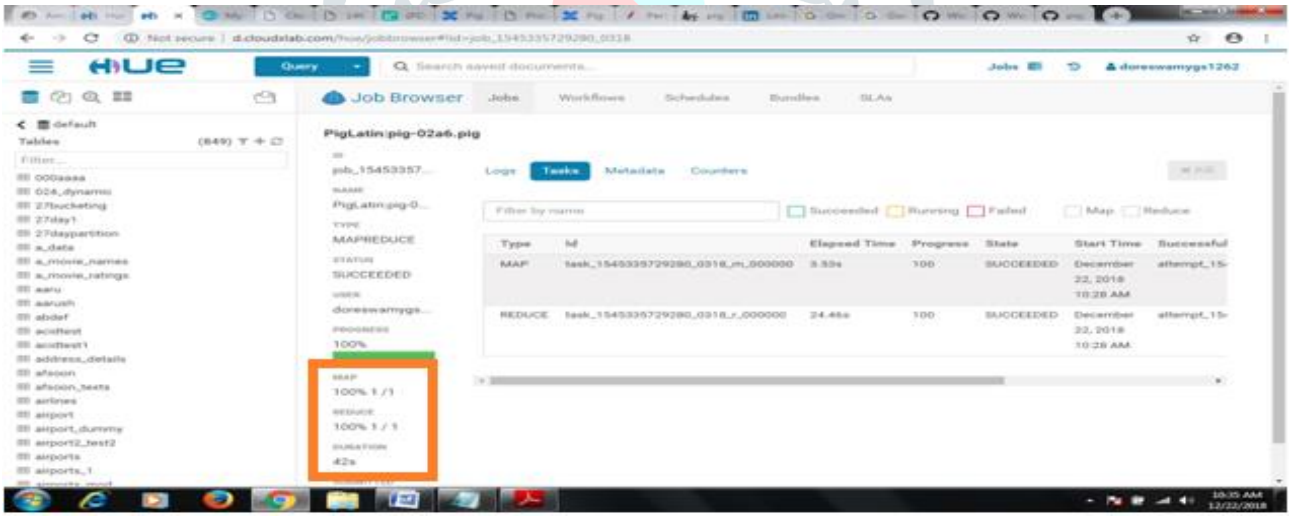

Figure 3: word count Execution time showing 42 Sec.

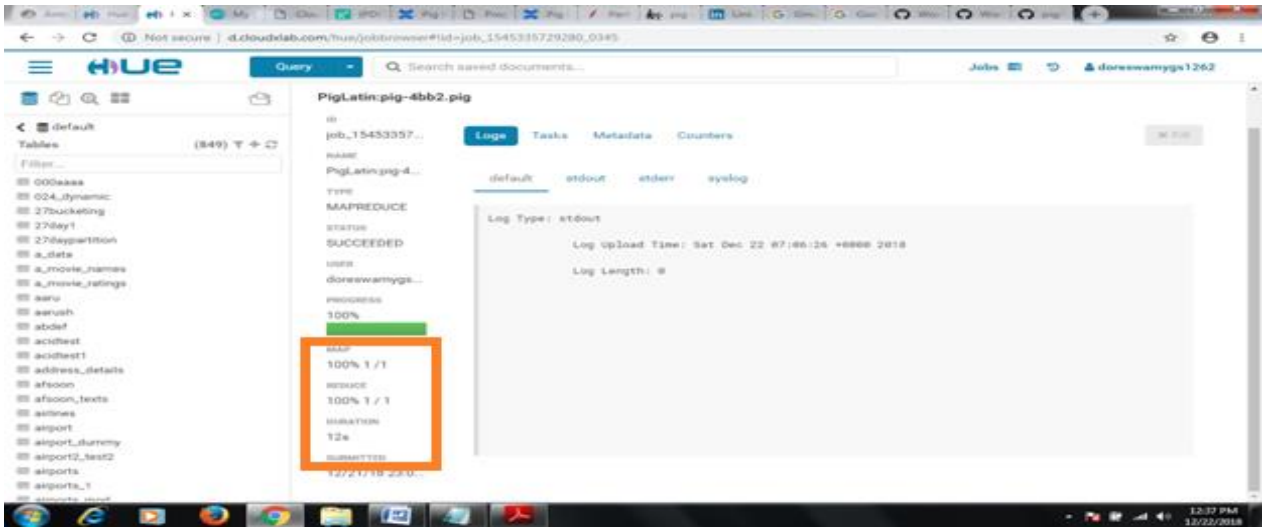

Figure 4: word count Execution time 12 Sec.

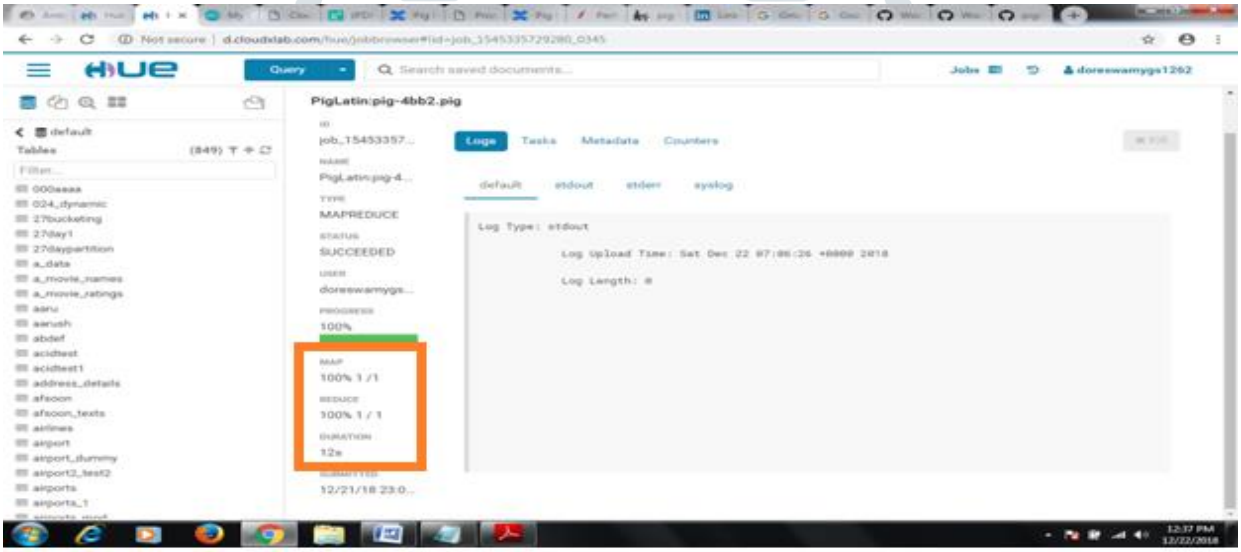

Figure 5: Word count execution time 12 Sec.

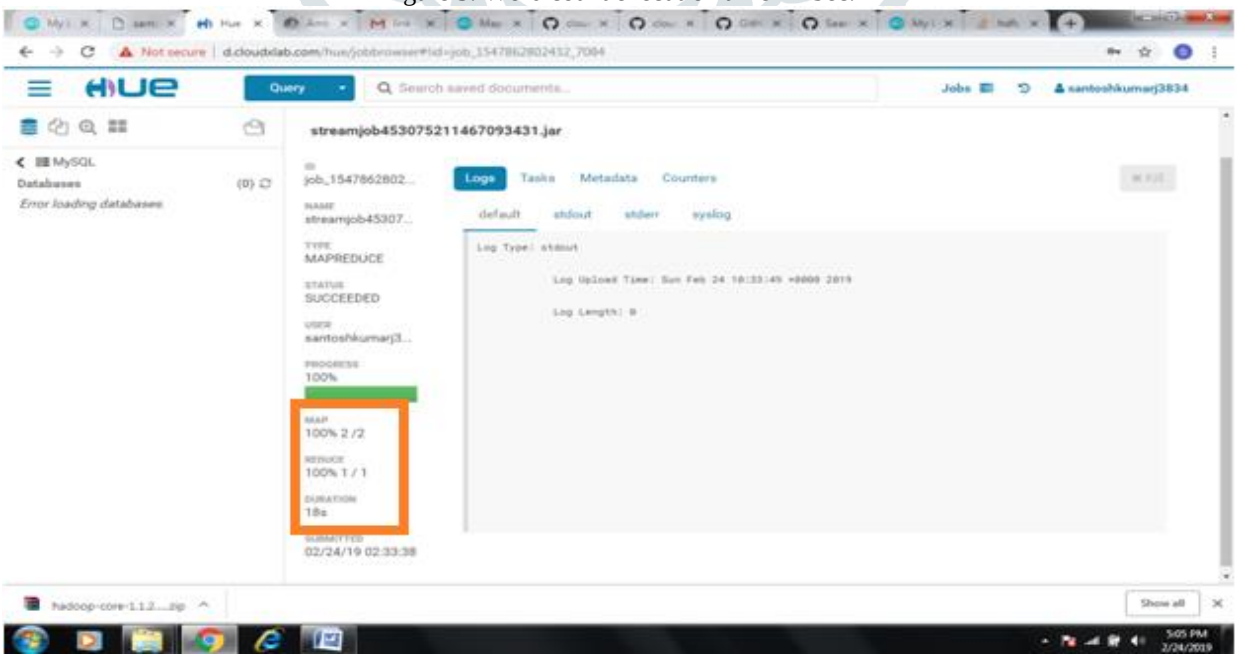

Figure 6: Word count execution time 18 Sec.

.

IV. CONCLUSION

Hadoop is the framework for big data processing, companies like Amazon have framework for processing the big data, also provide cloud infrastructure to store and process big data, with single mapper,reducer we achieved 18sec of execution time and where as with multiple mappers,reducesrs 12 sec of execution time for the word count data. Along with Order by, Sampling with PIG and Hive Query from results we say that by using multiple reducers and order by with Hive query one can achieve improved performance rather than single reducer and without order by of pig script.

V. ACKNOWLEDGMENT

I would like express my deep gratitude to the Principal, HoD and Staff of Computer Science and Engineering department of KSSEM, Bangalore for supporting me in doing this research work.

REFERENCES

[1] Stephen Kaisler , Frank Armour, J.Alberto Espinosa, William Money "Big Data Issues and Challenges Moving Forward" in Proceedings of the 46th Hawaii International Conference on System Sciences. 2013.

[2] OpenStack, http://openstack.org/, accessed on: June 10, 2012. P. Bhatotia, A. Wieder, R. Rodrigues, U.A. Acar and R. Pasquin, “Incoop: Mapreduce for Incremental Computations,” in Proceedings of Proceedings of the 2nd ACM Symposium on Cloud Computing (SoCC'11), pp 1-14, 2011.

[3] Ismail Ari, ErdiOlmezogullari, ÖmerFarukÇelebi†”Data Stream Analytics and Mining in the Cloud”, 2012 IEEE 4th International Conference on Cloud Computing Technology and Science, 2012.

[4] A. Abouzeid, K. B. Pawlikowski, D. J. Abadi, A. Rasin, and A. Silberschatz. HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. PVLDB, 2(1): pp 922–933, 2009.

[5] D. Agrawal, S. Das, and A. E. Abbadi. Big data and cloud computing: New wine or just new bottles? PVLDB, 3(2):pp 1647–1648, 2010.

[6] D. Agrawal, A. El Abbadi, S. Antony, and S. Das. Data Management Challenges in Cloud Computing Infrastructures. In DNIS, pp 1–10, 2010.

[7] Big data: The next frontier for innovation, competition, and productivity. James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. McKinsey Global Institute. May 2011.

[8] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” Commun. ACM, vol. 51, no. 1, pp 107–113, January 2008.

[9] Dean, Jeffrey, and Sanjay Ghemawat, MapReduce: Simplified data processing on large clusters, Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December 2004.

[10] F. N. Afrati and J. D. Ullman. Optimizing Joins in a Map-Reduce Environment. In EDBT, pp 99–110, 2010.