ijecs
open access www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242

Volume 3, Issue 11 November, 2014 Page No. 9061-9065

Enhancing Massive Data Analytics with the Hadoop Ecosystem

Misha Sheth[a*], ***Purna Mehta***[b], ***Khushali Deulkar***[c]

[a] Student, Dept. of Computer Engineering, D. J. Sanghvi COE, Vile Parle(West), Mumbai, India

[b]Student, Dept. of Computer Engineering, D. J. Sanghvi COE, Vile Parle(West), Mumbai, India

[c]Asst. Prof, Dept. of Computer Engineering, D. J. Sanghvi COE, Vile Parle(West), Mumbai, India

Abstract

With the advent of the Information Age, organizations are establishing their virtual presence, doing away with static locations and easing into presence on cloud. There is a rise in the volume of data in various industries and distributed storage of this data demands for efficient parallel processing. With the arrival of big data also came the realization that this data can be utilized for intelligent decision making. In addition to the complexity of tools and infrastructures that are required to manage huge volumes of data, there is an urgency to identify and resolve the technologies that can properly take advantage of these volumes. Big data evolution is driven by fast-growing cloud-based applications developed using virtualized technologies. A subsequent development in tools to enable data processing in a distributed environment emerged, leading to the MapReduce framework. In this paper, we will see the various technologies that implement this framework with an emphasis on the components of Apache's Hadoop Ecosystem: Pig, Hive and JAQL and their uses in data analytics.

Keywords: Hadoop Ecosystem; MapReduce; Parallel processing; Big Data; Data Analysis

1. Introduction

Web based data processes handle massive amounts of data. The progresses in data storage and mining technologies adapt to the storage of increasing amounts of data. These data provide opportunities that allow businesses across all industries to gain real-time business insights. Data scientists focus on bringing out coherent conclusions from this massive data. These results can aid strategic choices by providing information regarding past trends or even predict the future possibilities based on specific patterns. Therefore, the uses of Massive Data Analytics are innumerable.

The concepts of cloud computing and big data are inter-linked. New cloud computing technologies are changing the way organizations store, access, and process colossal amounts of data stored at discrete sources. This is achieved by parallel and distributed computing. The technologies that implement this include Hadoop, MapReduce, and BigTable among others.

In order to simplify programming for a distributed environment, several tools have been developed, such as the MapReduce programming model, which has become popular, because of its automatic parallelism and fault tolerance. MapReduce finds applications in index building, article clustering and statistical machine translation. Although MapReduce is a very flexible programming model, many people find that it is incompetent for everyday data analysis tasks.

Hadoop is an open-source Apache Software Foundation project written in Java that enables the distributed processing of large datasets. Pig, Hive, and JAQL are Apache Hadoop components that all translate high-level languages into MapReduce jobs so that the programmer can work at a higher level than they would when writing MapReduce jobs in Java or other lower-level languages supported by Hadoop using Hadoop streaming.

Related Concepts

2.1 MapReduce Framework

It is a model for the processing and generation of large datasets. As is evident by the name, there are two main functions to be defined in any MapReduce implementation: Map and Reduce.Map phase involves division of the computation operation into many fragments and its distribution among the cluster nodes. Individual results evolved at the end of the Map phase will be eventually combined and reduced to a single outputin the Reduce phase, thereby producing the result in a very short span of time. The model consists of support for fault tolerance, data partitioning,and parallel execution.

In relation to High Performance Computing (HPC) platforms states that

theMapReduce model is comparatively strong when nodes in a cluster requiregigabytes of data. In such cases, network bandwidth soon becomes the bottleneckfor alternative HPC architecturesthat use such APIs as Message Passing Interface (MPI), over shared filesystemson the network[1].

2.2 Apache Hadoop

In a distributed environment, Hadoop remains the most popular choice for analysis and manipulation of data in a distributed environment. It was developed by Apache foundation and is well-known for its potential for growth and efficiency. Hadoop preserves the data in its distributed file system HDFS and provides easy access to desired data. HDFS is highly scalable and advantageous in its portability[3]. Along with storage, it also comprises of various tools to apply on the data for processing.Hadoop works on the principle that shifting the task to the data is always more optimal than vice versa. It therefore brings all its processing work to the location of the data.

2.3 Apache Hadoop Ecosystem

A number of High Level Query Languages (HLQLs) havebeen constructed on top of the Hadoop MapReduce realization, primarilyPig, Hive, and JAQL[1]. The languages that are supported by the Pig, Hive and JAQL technologies greatly reduce the size of the code as compared to the same written in Java. The extensibility of these languages also allows for definition of functions in Java.Finally, since all thesetechnologies run on top of Hadoop, when they do so, they have the same limitationswith respect to random reads and writes and low-latency queries as Hadoop does[1].

2.4 Pig

It was developed at Yahoo Research around 2006. Pig operates as a layer of abstraction on top of the MapReduce programming model[6]. The language used is called PigLatin which basically allows the user to program by connecting components together.The flexibility of Pig lies in the kind of data structures it can work on. It can function on complex, nested structures, and also processes unstructured data with equal suitability. It can operate without a schema, and if provided, it also can take advantage of it correctly. Like SQL, PigLatin is complete in terms of relational algebra. Turing completeness requires looping constructs, an infinite memory model, and conditional constructs. PigLatin is not Turing complete on its own, but is Turingcomplete when extended with User-Defined Functions[2].

The main advantage of Pig is that programmers can define how data should be analysed using high level statements instead of writing functions for each low level operation.

In addition, since Pig is fundamentally based on the MapReduce framework, it consists of all the basic advantages including fault tolerance and good scalability.

2.5 Hive

Hive is a technology which is developed by Facebook[2] andwhich turns Hadoop into a data warehouse complete with an extension of sql for querying. The language used by Hive is HiveQL. Being a SQL dialect, HiveQL is a declarative language Inpiglatin, we have to describe [4] thedataflow but in Hive results must be described. The data structured by Hive is in traditional format and has familiar structures that are easily understood like tables and partitions. The essential data types are supported by Hive including integers, floats, etc. It also supports complex structures like maps and lists. Hive includes a system catalogue - Metastore – that contains schemas and statistics, which are useful in data exploration, query optimization and query compilation[5].

The main advantage of Hive lies in the fact that users can extend the functionalities of a Hive-based program by defining their own types and functions. Since the language is similar to traditional SQL, almost anyone can easily understand the program. Like PigLatin and the SQL, HiveQL itself is a relationally complete language but it is not a Turing complete language. It can also be extended through UDFs just like Piglatin to be a Turing complete.

2.6 JAQL

JAQL is a functional data query language, which is built upon JavaScript Object Notation Language (JSON)[6]. It is a data flow language that operates on unstructured, structured and semi-structured data and is used for general purposes. It supplies a model for accessing data in traditional formats and provides functions for basic functions like filtering, aggregation and joins. JAQL is extendable with operations written in many programming languages because JSON has a much lower impedance mismatch than XML for example, yet much richer data types than relational tables. Every JAQL program is run in JAQL shell. Initially JAQL shell is run by JAQL shell command[4].

To achieve parallelism, it converts high-level queries into low-level queries consisting of MapReduce jobs. Similar to the flow of data typically present in a MapReduce job, JAQL queries can be thought of as pipelines of data that flow through various operators and end in a sink which is the final destination.

2.7 Massive Data Analytics with Apache Hadoop

To analyse big data, many organizations turn to open-source utilities found in the Apache Hadoop ecosystem. The choice of a particular tool depends on the needs of the analysis, the skill set of the data analyst, and the trade-off between development time and execution time.

Apache Pig provides a data flow language, Pig Latin that enables the user to specify reads, joins and other computations without the need to write a MapReduceprogram[8]. Like Hive, Pig generates a sequence of MapReduce programs to implement the data analysis steps. JAQL is used more often for data processing and querying.

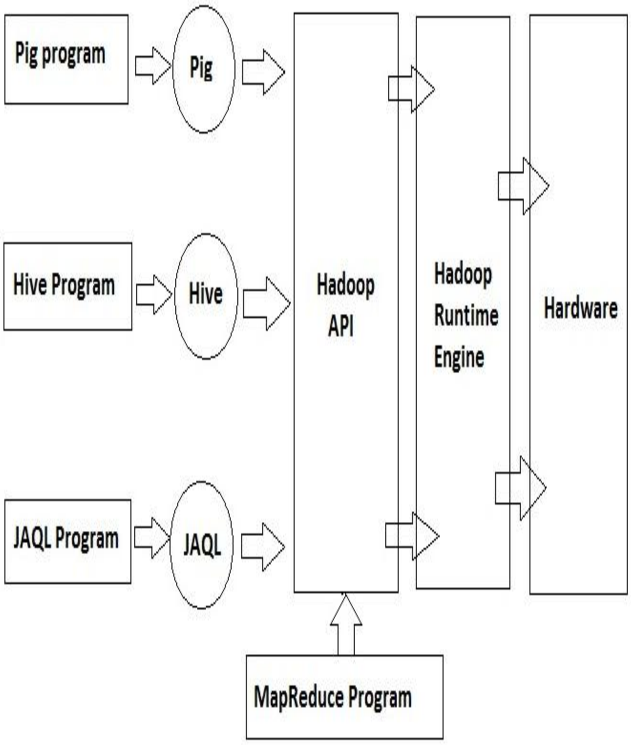

Fig. 1. Apache Hadoop Ecosystem

Table 1.Comparative Study between Pig, Hive and JAQL

PARAMETER	PIG	HIVE	JAQL
1. Description	A high level data flow language and execution framework for parallel computing	A data warehouse infrastructure that provides data summarization and ad hoc querying	A data processing and querying language
2. Language name	PigLatin	HiveQL	JAQL
3. Language type	Procedural data flow	Declarative (SQL Dialect)	Data flow
4. Language design	A data flow language influenced by both the declarative style of SQL and the more procedural MR	An SQL like language, presenting a declarative language	A functional, higher order programming language where functions maybe assigned as variables
5. Evaluation	During compilation	During compilation	At runtime
6. Developed by	Yahoo!	Facebook	IBM
7. Works on	Structured data	Unstructured data	Deeply nested semi-structured data as well as heterogeneous data
8. Source lines of code (mean ratio with Java)	5.2%	6.5%	9%
9. Most widely used by	Programmers and researchers	Analysts	Analysts
10. What it does?	Analyze large data sets that consist of high-level language for expressing data analysis programs, coupled with infrastructure or for evaluating them	Ability to filter, select, do equi-joins between two tables, evaluate aggregations, store the results, manage tables and partitions and plug in custom scripts	Access and load data from different sources, query data (databases), transform, aggregate and filter data, write data into different places
11. User Defined Functions	Extendable	Extendable	Extendable
12. Data structures it operates on	Complex, nested		JSON

2. Conclusion

With the onset of large scale businesses storing large amounts of data, parallel processing techniques become the most important methods to handle this data. Various enterprises are opting for Hadoop to store, manage, and analyse large amounts of data and thereby make informed decisions about their business and customers. The need of selecting the correct method becomes surmount as it is directly proportional to the cost, time and space needed. Pig, Hive and JAQL are such technologies that expedite the process of analysing big data with efficient storage and access.

References

1. R.J. Stewart, P.W. Trinder, and H-W. Loidl. Comparing High Level MapReduceQuery Languages. Mathematical And Computer Sciences Heriot Watt University.

2. Sanjeev Dhawan, Sanjay Rathee. Big Data Analytics using Hadoop Components like Pig and Hive. American International Journal of Research in Science, Technology, Engineering & Mathematics.13- 131; 2013

3. Anjali P P and Binu A. Comparative Survey Based on Processing Network Traffic Data Using Hadoop Pig and Typical Mapreduce. International Journal of Computer Science & Engineering Survey (IJCSES) Vol.5, No.1, February 2014.

4. Sanjay Rathee. Big Data and Hadoop with components like Flume, Pig, Hive and Jaql. International Conference on Cloud, Big Data and Trust 2013, Nov 13-15, RGPV

5. Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu and Raghotham Murthy. Hive – A Petabyte Scale Data Warehouse Using Hadoop. *Facebook Data Infrastructure Team.*

6. Satish Srirama, PelleJakovits.Large Scale Data Analysis Using Apache Pig.UNIVERSITY OF TARTU, 2011

7. Dave Jaffe.Three Approaches to Data Analysis with Hadoop. A Dell Technical White Paper; November 2013.

8. Sam Madden, “From Databases to Big Data”, IEEE Computer Society, 2012.

9. Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Google, Inc.; 2004