Michael Sanzo

Database Management

Big Data Paper Summary

Main Idea-Pig (Slide 3)

In order to allow people who were using Hadoop to focus more time on analyzing large data sets and less time writing mapper and reduce programs. This is so that the user can focus on what they want to do, rather than how it gets done. With that said, the main idea of "The Pig Experience" that pig programming language is designed to handle any kind of data with much more efficiency. In order for this to work there are two components that create this programming language distinctive which are a language; pig Latin, and a runtime environment. It provides the necessary parallel programming constructs and also gives sufficient control back to the programmer.

Implementation-Pig (Slide 4)

Pig was developed in a much different way than other projects related to parallel databases. This is because it is a collaboration of research and development engineering teams at Apache that drew an interesting perspective of what a data set needed when dealing with big data. With that, it is also an open source project which means it is highly combatable throughout various companies.

The pig programming language has two components to it, being the language itself and the runtime environment. First, the programming language, is significantly easier to the user rather than writing mapper and reducer programs. The language consists of three steps that create the much simpler method of parallel database programming which are LOAD, transformations, and DUMP or STORE.

1.LOAD; the objects are stored in HDFS which is a Java-based file system that provides scalable and reliable data storage. This is how it is capable of spanning to large clusters of servers.

2.Transformations; Pig Mix. Nested Elements, SPLIT, FILTER the rows out that are not of interest, JOIN multiple data sets, GROUP the data to construct solutions. Etc. There are 12 programs designed to collectively test these features and more.

3.DUMP/STORE; It is required that you identify the DUMP or STORE command otherwise the data will not be generated. DUMP-Output to screen when debugging program. STORE-Results from running program are stored in a file for further processing or analysis.

Streaming

Allows data to be pushed through external implementation processes as a part of a Pig data processing pipeline which is much more efficient.

UDFs

Flow control by acting in a similar behavior overtime. Must be written in Java and conform to UDF interface. This provides one route for users to include customer code whenever necessary in the data processing pipeline. (Pipeline is a set of data processing elements connected in series, where the output of one element is the input of the next one).

Analysis-Pig (Slide 5)

Open source which means that it is highly combatable to various companies.

Pig Mix was developed to measure performance and efficiency of the language in order to make sure that the user would be able to understand the functionality. This is typically done on a regular basis through algorithms.

Data Generator in Pig yields proprietary data sets at fast speeds with max customization for the user.

Results; the correspondence between execution speed and code maintence. Pig is becoming much more powerful and robust.

Main Ideas-LSDP (Slide 6)

All DBMSs need the data to adapt to a definite schema, whereas MapReduce authorizes data to be in any random format. Other differences also include how each system provides indexing and compression optimizations, programming models, the way in which data is distributed, and query execution strategies.

1.MapReduce; Simplicity. Map and reduce are the two functions that the program consists of. The input data set is stored in a collection of dividers in a distributed file system. Then the program is put into a distributed processing structure and performed.

2. Parallel DBMSs; Support standard relational tables and SQL. In addition, the program has multiple machines to store the data that is transparent to the end user. Most tables are portioned over the nodes in a cluster. The system uses and optimizer that translates SQL commands into a query plan where it is then divided into multiple nodes.

Architectural Elements

The nature of MapReduce is well suited for a development environment with a small number of programmers and limited domain. Because of this it constraints the programs capabilities, unlike how we saw the Pig programming language. This lacks its efficiency when dealing with longer and larger sized projects which can create a lack in reliable. However, it does have key elements that that are important in order to structure large scale data analysis.

Schema Support; Parallel DBMSs require data to fit into the relational paradigm of rows and columns. In contrast, the MR model does not require that data files adhere to a schema defined using the relational data model. That is, the MR programmer is free to structure their data in any manner or even to have no structure at all.

Indexing; Hash or B-tree indexes are most common when being used to accelerate access to specific data. If one is looking to analyze specific information within a dataset, then using a proper index eliminates the constraints within the search drastically. Database systems support multiple indexes per table. The query optimizer is beneficial because it can decide which index to use for each query or see if it should complete a sequential search.

Modeling; Written by starting what you want, rather than presenting an algorithm for how to get it. Giving an algorithm for data access. Was not a repetitive tasks system.

Data Distribution Strategy; Main objective is to send the computation of data rather than the opposite.

Flexibility; SQL. Incredible aptitude within the language and it can be embedded into the program.

Fault Tolerance; Very sophisticated failure model for parallel DBMSs.

Implemented-LSDP (Slide 7)

Hadoop is an open source project that requires the use of a Java programming structure that cares for the processing and storage of big data in a distributed computing environment. DBMS-X is a parallel SQL database management system for important relational database merchants that stores its data in a row based arrangement. Each set is hash split across all nodes on the noticeable feature for that particular data set and then sorted and indexed accordingly. Lastly, Vertica is an analytical database management software founded by Michal Stonebraker to manage massive data sets, rapidly expanding capacities of data that arranges very fast efficiency when used.

Analytical Tasks

Data Loading; Hadoop needs to directly load document files into its internal storage. Meanwhile DBMS-X and Vertica execute a UDF that processes document files on each node at runtimes and loads this data into a temporary set.

Selection Task; An important reason for why the parallel DBMSs are able to outperform Hadoop is that both Vertica and DBMS-X use an index on this rankings table. It is sorted by column and stored in the Rankings table already sorted.

Aggregation Task; This will determine the time it takes for the program to execute a command. It is looked at as an advantageous aspect to use a column-store system when processing fewer groups for a task. This is because the when accessing columns that consist of small data compared to large data there are few groups that need to merge. This helps efficiency and cost. Vertica is thus able to outperform the other two systems from not reading unused parts.

Join Task; Grabs data from two different data sets and join them together.

UDF Aggregation Task; This task includes the count for each document in the dataset. Self-references, Nodes connect the HTML documents into larger files when storing them in HDFS. This was found to improve Hadoop's performance.

Analysis-LSDP (Slide 8)

B-Tree indices are used to speed the execution of specific operations within the program. In the past, storage and memory was an issue for many programs. Now it has adapted to a much better mechanism. Compression techniques zone in on specific commands that need to be performed in the data set which is advantageous because it is direct. Lastly, the need of parallel algorithms id important especially when dealing with large sets of data.

Main Ideas-Michael Stonebraker (Slide 10)

https://www.youtube.com/watch?v=9K0SWs1mOD0 (video link did not work, used this one)

Data Warehouse Market; all major vendors have column stores, which are much more common and faster today.

OLTP Market; Transaction process has relatively small databases. Use main memory and it will be a very different implementation. Lightweight system that uses unique techniques.

NoSQL Market; No standards. With that, there are many vendors of data models and architectures.

Complex Analytics; Business Intelligence. Use of data warehouses, like graphical front end to sequel analytics. Replace business analysts with data scientists.

Streaming Market; development of stream processing engines. We can see that this has become huge in the last decade. (FB live stream, YouTube live stream)

Graph Analytics Market; programs that analyze graphs to pull necessary data you request.

There is a very big diversity of engines because there are no standards. There are many great opportunities for new ideas. Times have changed and it is an interesting time to be a database researcher!

Advantages and Disadvantages-Pig (Slide 11)

Summary of key points that are an advantages and disadvantages to the Pig Programming Language.

Advantages are that pig Latin works on algorithms where the process is repeated multiple times to check efficiency. Faith in optimizers influence speeds that the data can be processed. User interaction would be much more complex because you can nest within the program for maximum customization and compatibility reasons. Lastly, DAGs, which are directed acyclic graphing which helps all these perform better because it is the special ordering of data.

Disadvantages are that data schemas are not enforced openly but indirectly. This will cause extra steps and complexity. Another disadvantage would be the mass diversity of alternate programs that could be used to show data efficiently. This is a disadvantage because it opens up the opportunity that an easier much less complex platform will be used.

Sources

Gates, Alan. “Building a High Level Dataflow System on Top of MapReduce.” *Yahoo!, Inc.*. N.p., 24 Aug. 2009. Web. 18 Oct. 2016.

Pavlo, Andrew. "A Comparison of Approaches to Large Scale Data Analysis." *SIGMOD*. N.p., 24 Sept. 2007. Web. 18 Oct. 2016.

Stonebraker, Michael. "IEEE ICDE 2015 Ten-Year Influential Talk." *YouTube*. YouTube, 20 June 2015. Web. 18 Oct. 2016.