EXTRACT SESSION USING PIG

Seema Sultana* and Sunanda Dixit**

*Dayananda Sagar College of Engineering/Department of Information Science and Engineerig, Bangalore, India
E-mail: seemasultana17@gmail.com
**Dayananda Sagar College of Engineering/Department of Information Science and Engineerig, Bangalore, India
E-mail: sunanda.bms@gmail.com

ABSTRACT: There is a growing need for ad-hoc analysis of extremely large data sets, especially at internet companies where invocation critically depends on being able to analyze terabytes of data collected everyday Parallel database products prohibitively expensive at the scale. The success of map-reduce programming model and implementations on hardware is the evidence for all. But the map-reduce is too low-level and rigid and the maintenance of user code is very difficult and it is hard to reuse. Here it describes a new language called Pig Latin that spots a difference between declarative style of SQL (Structure Query Language) and low-level procedural style of map-reduce. Thus, Pig is fully implemented and complies Pig Latin into physical plans that are executed over Hadoop, an open-source, map-reduce implemented.

KEYWORDS: Ad-hoc analysis, Map-Reduce, Pig-Latin, SQL, Hadoop.

INTRODUCTION

In 2006, Apache Pig was developed as research project at Yahoo, especially to create and execute MapReduce jobs on every dataset. In 2008, the first release of Apache Pig came out. In 2010, Apache Pig graduated as Apache top-level project. Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze large sets of data representing them as data flows. Pig is generally used with Hadoop, can perform all the data manipulation operations in Hadoop using Apache Pig.
For example, an operation that would require to type 200 lines of code (LoC) in Java can be easily done by typing as less as just 10 LoC in Java can be easily done by typing as less as just 10 LoC in Apache Pig. Ultimately Apache Pig reduces the development time by almost 16 times.
Apache Pig comes with the following features:
Rich set of operators: It provides many operators to perform operations like join, sort, filter, etc.
Ease of programming: Pig Latin is similar to SQL and it is easy to write a Pig script if are good at SQL
Optimization opportunities: The tasks in Apache Pig optimize their execution automatically, so the programmers need to focus only on semantics of the language.
UDF's: Pig provides the facility to create user-defined Functions in other programming languages such as Java and invoke or embed them in Pig Scripts.

Apache Pig vs Hive

Table 1: Apache Pig vs Hive

Pig	Hive
Apache Pig uses the language called Pig Latin. It was originally created by Yahoo.	Hive uses a language called HiveQL.It was originally created at Facebook.
It is a data flow language.	It is a query processing language.
It is a procedural language.	It is a declarative language.
It can handle structured, unstructured, semi-structured	It can handle only structured data.

LITERATURE SURVEY

Javid Ali and Anandhamala G S describes in [1] that user's use the web for different purposes. The access time and level will be different from one user to another. The main objective was to control the session flow based on the user access level in conjunction with fault committed by them.

On extracting unit test from interactive live programming sessions [2] says that programs run by software engineering methodology like unit testing should captured in repeated and automated format. Here it is proposed that leverage these exploratory testing bursts by automatically extracting scripted tests from a recording of live programming sessions.

Ruhma Tahir et al in [3] informs that cryptography has become an essential for providing security in embedded system applications. Here it is investigated and analyzed ICMetrics and its counterpart scheme for generation of strong high entropy ICMetrics session key pairs.

Budi Thomas Jap et al in [4] describes that train accidents have a massive impact on general community. Most train accidents can be attributed to fatigue countermeasure devices that can warn drivers of fatigue status and prevent accidents can greatly benefit train drivers, passengers, society and general community.

An adjustment strategy on multi-session EEG data for online left/right hand imagery classification [5] describes that electroencephalography (EEG) is used as an interface to communicate between patients and doctors. Here it has been proposed that an adjustment strategy that can be applied online to all features by subtracting "estimated means" and dividing "estimated interquartile range" which are obtaining by using exponentially weighted moving average.

Synchronizing multimedia data such as audio and video with 3D depth [6] skeletal data of a patient performing therapy at home is a challenging task. This is because the media characteristics of video, audio, and skeletal stream data are different. At the end of the first-tier synchronization, all the media streams have to be further synchronized with respect to a model therapy skeletal stream.

Predictive user behavior through sessions using the web log mining [7] paper says that initially, each user is identified according to IP address specified in the log file and corresponding user sessions are extracted. Server-side logs can be automatically generated by web servers. Client-side logs can capture accurate, comprehensive usage data for usability analysis.

Ricardo Dias, Manue J. Fonseca describes in [8], the usage of temporal context and session diversity in session-based collaborative filtering properties and session diversity, to group and compare the similarity of sessions, which is able to implicitly model temporal patterns.

Non-stationarity of electroencephalograph (EEG) [9] data from session-to-session transfer is one of the challenges for EEG-based brain-computer interface systems. An adaptive extreme learning machine (AELM) is proposed to update the initial classifier from the calibration session by using chunks of EEG data from the evaluation session whereby the common spatial pattern (CSP) algorithm is used to extract the most discriminative features.

PIG ARCHITECTURE

Pig Latin is the language used to analyze data in Hadoop using Pig. It provides a rich set of rich set of operators to perform various operations on the data.

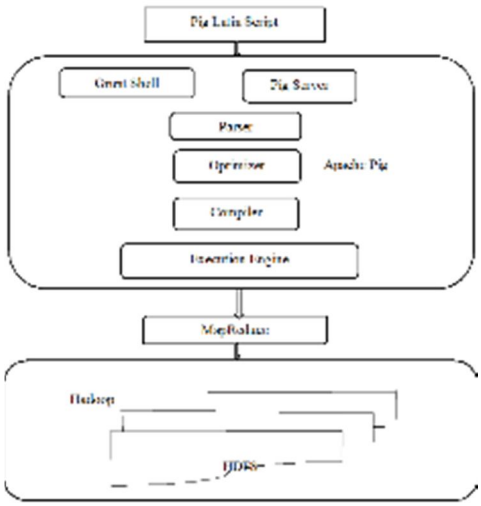

Figure 1. Pig Architecture

Parser: Pig scripts are handled by the parser. It checks the syntax of the script, does type checking, and other miscellaneous checks.
Optimizer: Logical plan is passed to the logical optimizer, which carries out the logical optimizations such as projection and pushdown.
Compiler: It compiles the optimized logical plan into series of Map Reduce jobs.
Execution engine: Finally the MapReduce jobs are submitted to the Hadoop. Then MapReduce jobs are executed on Hadoop producing the desired results.

EXECUTION OF APACHE PIG

Local mode

In this mode, all the files are installed and run from local host and local file system. This mode is used for testing purpose. Here there is no need of Hadoop or HDFS.
Local mode: $ ./pig –x local

MapReduce

MapReduce mode is where load or process the data that exists in the Hadoop File System (HDFS) using Apache Pig. In this mode MapReduce job is invoked in the back-end to perform a particular operation on the data that exists in the HDFS.
MapReduce mode: $ ./pig –x mapreduce

PIG LATIN

It focuses on the language primitives, and not on how they can be implemented to execute in parallel over a cluster.

Specifying input data: LOAD

In Pig Latin program the first step is to specify what input data files are and how the files contents are to be deserialized i.e, converted into pig data model. An input file is assumed to contain a sequence of tuples i.e, bag. This step is carried out by the LOAD command.

```
Example:    queries = LOAD ‘query_log.txt’
                    USING myLoad() AS(userId, queryString, timestamp);
```

The input file is a query_log.txt. The loaded tuples have three fields named userId, queryString, timestamp.Both USING clause and the AS clause are optional. The return value of LOAD command is handled to bag which will be assigned to the variable queries.

Filter: Discarding the unwanted data

The operation of FILTER command is to retain only some subset of the data that is of interest and discarding the rest.
Example: real_query = FILTER queries BY userId neq ‘bot’;
The operator neq in the above example is used to signify string comparison, as opposed to numeric comparison which is specified via ==.

Map-Reduce in Pig Latin:

With the GROUP and FOREACH statements, it is trivial to express a map-reduce program in Pig Latin.

```
Example:   Map_result = FOREACH input GENERATE FLATTEN(map(*));
           Key_groups = GROUP map_result BY $0;
            output = FOREACH key_groups GENERATE reduce(*);
```

STORE: Asking for output

The user can ask for the result of Pig Latin expression sequence to be materialized to a file, by issuing the STORE command.

```
Example: STORE query_revenues INTO ‘mtput’ USING myStore();
```

CONCLUSION

Pig is meant for online, ad-hoc, scan-centric workloads. Amazon Dynamo, Google's BigTable are the examples. But these systems are not meant for data analysis. BigTable does have hooks through which data residing in BigTable can be analyzed using a map-reduce job, but in the end analysis is through map-reduce only. PigLatin has a higher level primitives like filtering and aggregation, an arbitrary number of them can be extensibily chained together in a PigLatin program and all primitives can use user-defined functions.

REFERENCES

[1] Javid Ali and Anandhamala G S, "Efficient fault tolerance technique for controlling the user session flow over online transaction applications," International conference on Control Instrumentation, Communication and Computational Technologies, pp. 109-114, May 2016.
[2] Adrian Kuhn, "On extracting unit tests from interactive live programming sessions," International Conference on Software Engineering, pp. 1241-1244, September 2013.
[3] Ruhma Tahir et al, "Resilience against brute force and rainbow table attacks using strong ICMetrics session key pairs," International Conference on Communications, Signal Processing and their Applications, pp. 1-6, March 2013.
[4] Budi Thomas Jap et al, "Using spectral analysis to extract frequency components from electroencephalography: application for fatigue countermeasure in train drivers," International Conference on Wireless Broadband and Ultra Wideband Communications, pp. 13-13, September 2007.
[5] Sitthiphong Muthonh et al, "An adjustment strategy on multi-session EEG data for online left/right hand imagery classification," International Conference on Knowledge and Smart Technology, pp. 179-183, March 2016.
[6] Mohamed Abdur Rahman, Abdulhameed Alelaiwi, "A synchronized multimedia in-home therapy framework in big data environment," International Conferece o Multimedia & Expo Workshops, pp. 1–6, September 2016.
[7] G. Neelima, Sireesha Rodda, "Predicting user behavior through sessions using the web log mining," International Conference on Advances in Human Machine Interaction, pp. 1-5, April 2016.
[8] Ricardo Dias, Manue J. Fonseca, "Improving music recommendation in session-based collaborative filtering bby using temporal context," International Conference on Tools with Artificial Intelligence, pp. 783-788, February 2014.
[9] Atieh Bamdadian et al, " Non-stationarity of electroencephalograph (EEG)," Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2188-2191, September 2013.