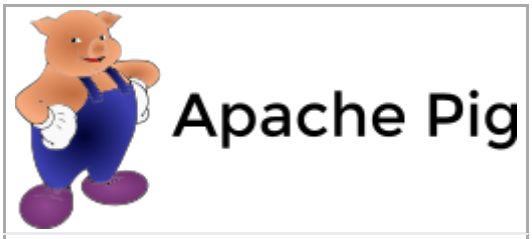
Apache Pig

Apache Pig[1] is a high-level platform for creating programs that run on Apache Hadoop. The language for this platform is called **Pig Latin**.[1] Pig can execute its Hadoop jobs in MapReduce, Apache Tez, or Apache Spark.[2] Pig Latin abstracts the programming from the Java MapReduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for relational database management systems. Pig Latin can be extended using user-defined functions (UDFs) which the user can write in Java, Python, JavaScript, Ruby or Groovy[3] and then call directly from the language.

Apache Pig



Developer(s)	Apache Software Foundation, Yahoo Research
Initial release	September 11, 2008
Stable release	0.17.0 / June 19, 2017
Repository	svn.apache.org/repos/asf/pig/ (http://svn.apache.org/repos/asf/pig/)
Operating system	Microsoft Windows, OS X, Linux
Type	Data analytics
License	Apache License 2.0
Website	pig.apache.org (https://pig.apache.org)

Contents

History

Apache Pig was originally[4] developed at Yahoo Research around 2006 for researchers to have an ad hoc way of creating and executing MapReduce jobs on very large data sets. In 2007,[5] it was moved into the Apache Software Foundation.

Version	Original release date	Latest version	Release date[6]
0.1	2008-09-11	0.1.1	2008-12-05
0.2	2009-04-08	0.2.0	2009-04-08
0.3	2009-06-25	0.3.0	2009-06-25
0.4	2009-08-29	0.4.0	2009-08-29
0.5	2009-09-29	0.5.0	2009-09-29
0.6	2010-03-01	0.6.0	2010-03-01
0.7	2010-05-13	0.7.0	2010-05-13
0.8	2010-12-17	0.8.1	2011-04-24
0.9	2011-07-29	0.9.2	2012-01-22
0.10	2012-01-22	0.10.1	2012-04-25
0.11	2013-02-21	0.11.1	2013-04-01
0.12	2013-10-14	0.12.1	2014-04-14
0.13	2014-07-04	0.13.0	2014-07-04
0.14	2014-11-20	0.14.0	2014-11-20
0.15	2015-06-06	0.15.0	2015-06-06
0.16	2016-06-08	0.16.0	2016-06-08
0.17	2017-06-19	0.17.0	2017-06-19
Legend: Old version	Older version, still maintained	**Latest version**	Latest preview version		

Example

Below is an example of a "Word Count" program in Pig Latin:

```
input_lines = LOAD '/tmp/my-copy-of-all-pages-on-internet' AS (line:chararray);

-- Extract words from each line and put them into a pig bag
-- datatype, then flatten the bag to get one word on each row
words = FOREACH input_lines GENERATE FLATTEN(TOKENIZE(line)) AS word;

-- filter out any words that are just white spaces
filtered_words = FILTER words BY word MATCHES '\\w+';

-- create a group for each word
word_groups = GROUP filtered_words BY word;

-- count the entries in each group
word_count = FOREACH word_groups GENERATE COUNT(filtered_words) AS count, group AS word;

-- order the records by count
ordered_word_count = ORDER word_count BY count DESC;
STORE ordered_word_count INTO '/tmp/number-of-words-on-internet';
```

The above program will generate parallel executable tasks which can be distributed across multiple machines in a Hadoop cluster to count the number of words in a dataset such as all the webpages on the internet.

Pig vs SQL

In comparison to SQL, Pig

1. has a nested relational model,
2. uses lazy evaluation,
3. uses extract, transform, load (ETL),
4. is able to store data at any point during a pipeline,
5. declares execution plans,
6. supports pipeline splits, thus allowing workflows to proceed along DAGs instead of strictly sequential pipelines.

On the other hand, it has been argued DBMSs are substantially faster than the MapReduce system once the data is loaded, but that loading the data takes considerably longer in the database systems. It has also been argued RDBMSs offer out of the box support for column-storage, working with compressed data, indexes for efficient random data access, and transaction-level fault tolerance.[7]

Pig Latin is procedural and fits very naturally in the pipeline paradigm while SQL is instead declarative. In SQL users can specify that data from two tables must be joined, but not what join implementation to use (You can specify the implementation of JOIN in SQL, thus "... for many SQL applications the query writer may not have enough knowledge of the data or enough expertise to specify an appropriate join algorithm."). Pig Latin allows users to specify an implementation or aspects of an implementation to be used in executing a script in several ways.[8] In effect, Pig Latin programming is similar to specifying a query execution plan, making it easier for programmers to explicitly control the flow of their data processing task.[9]

SQL is oriented around queries that produce a single result. SQL handles trees naturally, but has no built in mechanism for splitting a data processing stream and applying different operators to each sub-stream. Pig Latin script describes a directed acyclic graph (DAG) rather than a pipeline.[8]

Pig Latin's ability to include user code at any point in the pipeline is useful for pipeline development. If SQL is used, data must first be imported into the database, and then the cleansing and transformation process can begin.[8]

See also

- Apache Hive
- Sawzall — similar tool from Google

References

1. "Hadoop: Apache Pig" (http://pig.apache.org/). Retrieved Sep 2, 2011.
2. "[PIG-4167] Initial implementation of Pig on Spark - ASF JIRA" (https://issues.apache.org/jira/browse/PIG-4167). *issues.apache.org*. Retrieved 2018-12-29.
3. "Pig user defined functions" (http://pig.apache.org/docs/r0.11.1/udf.html). Retrieved May 3, 2013.
4. "Yahoo Blog:Pig – The Road to an Efficient High-level language for Hadoop" (https://web.archive.org/web/20160203181220/https://developer.yahoo.com/blogs/hadoop/pig-road-efficient-high-level-language-hadoop-413.html). Archived from the original (https://developer.yahoo.com/blogs/hadoop/pig-road-efficient-high-level-language-hadoop-413.html) on February 3, 2016. Retrieved May 23, 2015.
5. "Pig into Incubation at the Apache Software Foundation" (https://web.archive.org/web/20160203162733/https://developer.yahoo.com/blogs/hadoop/pig-incubation-apache-software-foundation-393.html). Archived from the original (https://developer.yahoo.com/blogs/hadoop/pig-incubation-apache-software-foundation-393.html) on February 3, 2016. Retrieved May 23, 2015.

6. "Apache Pig Releases" (https://pig.apache.org/releases.html). *Apache*. Retrieved 2019-03-13.
7. "Communications of the ACM: MapReduce and Parallel DBMSs: Friends or Foes?" (https://web.archive.org/web/20150701205317/http://database.cs.brown.edu/papers/stonebraker-cacm2010.pdf) (PDF). Archived from the original (http://database.cs.brown.edu/papers/stonebraker-cacm2010.pdf) (PDF) on July 1, 2015. Retrieved May 23, 2015.
8. "Yahoo Pig Development Team: Comparing Pig Latin and SQL for Constructing Data Processing Pipelines" (https://web.archive.org/web/20150530103839/https://developer.yahoo.com/blogs/hadoop/comparing-pig-latin-sql-constructing-data-processing-pipelines-444.html). Archived from the original (https://developer.yahoo.com/blogs/hadoop/comparing-pig-latin-sql-constructing-data-processing-pipelines-444.html) on May 30, 2015. Retrieved May 23, 2015.
9. "ACM SigMod 08: Pig Latin: A Not-So-Foreign Language for Data Processing" (http://infolab.stanford.edu/~olston/publications/sigmod08.pdf) (PDF). Retrieved May 23, 2015.

External links

- Official website (https://pig.apache.org)