WQD7007 Big Data Management

Apache Pig

Agenda

- Pig - performing a long series of data operations

Apache Pig

- Apache Pig allows Apache Hadoop users to **write complex MapReduce transformations** using a simple scripting language called Pig Latin.
  - Pig translates the Pig Latin script into MapReduce so that it can be executed within YARN for access to a single dataset stored in the Hadoop Distributed File System (HDFS).
  - Pig was designed for performing a long series of data operations, making it ideal for three categories of Big Data jobs:
    - **Extract-transform-load (ETL)** data pipelines,
    - **Research** on raw data, and
    - **Iterative data processing.**

Apache Pig

- MapReduce allows allows you to specify map and reduce functions, but working out how to fit your data processing into this pattern may sometimes require you to write multiple MapReduce stages.
  - With Pig, data structures are much richer and the transformations you can apply to data are much more powerful.

Pig

CHARACTERISTIC	BENEFIT
Extensible	Pig users can create custom functions to meet their particular processing requirements
Easily Programmed	Complex tasks involving interrelated data transformations can be simplified and encoded as data flow sequences. Pig programs accomplish huge tasks, but they are easy to write and maintain.
Self-Optimizing	Because the system automatically optimizes execution of Pig jobs, the user can focus on semantics.

Apache Pig

- Pig runs on Apache Hadoop YARN and makes use of MapReduce and the Hadoop Distributed File System (HDFS).
- Pig Latin is abstracts from the Java MapReduce idiom into a form similar to SQL.
  - While SQL is designed to query the data, Pig Latin allows you to **write a data flow** that describes how your data will be transformed (such as aggregate, join and sort).
  - Since Pig Latin scripts **can be graphs (instead of requiring a single output) it is possible to build complex data flows involving multiple inputs, transforms, and outputs.**
  - Users can extend Pig Latin by writing their own functions, using Java, Python, Ruby, or other scripting languages.
  - Pig Latin is sometimes extended using UDFs (User Defined Functions), which the user can write in any of those languages and then call directly from the Pig Latin.

Apache Pig

- The user can run Pig in two modes, using either the "pig" command or the "java" command:
  1. **MapReduce Mode.** This is the default mode, which requires access to a Hadoop cluster. The cluster may be a pseudo- or fully distributed one.
  2. **Local Mode.** With access to a single machine, all files are installed and run using a local host and file system.

Difference between Hive and Pig

Pig	Hive
Procedural Data Flow Language	Declarative SQLish Language
For Programming	For creating reports
Mainly used by Researchers and Programmers	Mainly used by Data Analysts
Operates on the client side of a cluster.	Operates on the server side of a cluster.
Does not have a dedicated metadata database.	Makes use of exact variation of dedicated SQL DDL language by defining tables beforehand.
Pig is SQL like but varies to a great extent.	Directly leverages SQL and is easy to learn for database experts.
Pig supports Avro file format.	Hive does not support it.

Source: https://www.dezyre.com/article/difference-between-pig-and-hive-the-two-key-components-of-hadoop-ecosystem/79

Extra: Pig VS Hive

Pig	Hive
Procedural Data Flow Language	Declarative SQLish Language
For Programming	For creating reports
Mainly used by Researchers and Programmers	Mainly used by Data Analysts
Does not have a dedicated metadata database.	Makes use of exact variation of dedicated SQL DDL language by defining tables beforehand.
Pig is SQL like but varies to a great extent.	Directly leverages SQL and is easy to learn for database experts.
Examples: data cleaning e.g. web log analysis, web crawling, process a million songs	Examples: data warehousing, ad hoc queries for retail analysis

Source: https://www.dezyre.com/article/difference-between-pig-and-hive-the-two-key-components-of-hadoop-ecosystem/79

Online reference

- https://hortonworks.com/tutorial/beginners-guide-to-apache-pig/
- https://pig.apache.org/docs/r0.12.0/basic.html#store

Run pig script within command line

```
[hive@sandbox ~]$ cat tmp.pig
truck_events = LOAD '/user/maria_dev/truck_event_text_partition.csv' USING PigStorage(',');
DESCRIBE truck_events;
```

```
[hive@sandbox ~]$ pig -f tmp.pig
18/04/17 03:54:16 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
18/04/17 03:54:16 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
18/04/17 03:54:16 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2018-04-17 03:54:17,130 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0.2.5.0.0-1245 (rexported) compiled Aug 26 2016, 02:
07:35
2018-04-17 03:54:17,130 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/hive/pig_1523937257116.log
2018-04-17 03:54:19,589 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/hive/.pigbootup not found
2018-04-17 03:54:20,109 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system
at: hdfs://sandbox.hortonworks.com:8020
2018-04-17 03:54:21,704 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-tmp.pig-651ca0b0-b75f-4636-82d3-3750
092d698a
2018-04-17 03:54:23,043 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://sandb
ox.hortonworks.com:8188/ws/v1/timeline/
2018-04-17 03:54:23,354 [main] INFO  org.apache.pig.backend.hadoop.PigATSClient - Created ATS Hook
Schema for truck_events unknown.
2018-04-17 03:54:25,751 [main] INFO  org.apache.pig.Main - Pig script completed in 9 seconds and 477 milliseconds (9477 ms)
[hive@sandbox ~]$
```

Please follow hortonworks beginner guide to understand more about the coding style

Important keywords

- LOAD
- DESCRIBE
- LIMIT
- DUMP
- ORDER
- FILTER
- GROUP

Concluding Remarks

- Pig – design a set of pre-defined operations for large scale processing