APACHE PIG OVERVIEW

Apache Pig Overview

- **Apache Pig Overview**
- Pig Latin
- Data Analysis with Pig
- Pig Execution Modes
- Grunt

Apache Pig Overview

- Apache Pig is a platform of analyzing large datasets:
  - with a high-level language (Pig Latin) for expressing data analysis programs
  - with a compilation layer that produces a sequence of MapReduce jobs from Pig Latin statements
- Data analysts write Pig Latin statements to load a distributed dataset, assign a schema to it, apply relational operations, and store the result
- The Pig Latin statements are then compiled into a series of MapReduce jobs, and run in the defined execution mode (currently; local, MapReduce, Tez)

Introduction to Apache Pig

- Apache Pig Overview
- **Pig Latin**
- Data Analysis with Pig
- Pig Execution Modes

Pig Latin

- The key properties of Pig Latin is its:
  - Ease of programming: Complex data analysis tasks of multiple interrelated transformations (dataflow) are easy to write, understand, and maintain
  - Optimization opportunities: Pig can optimize the execution of Pig scripts automatically, allowing its users focus on the semantics
  - Extensibility: Out-of-box Pig capabilities can be extended via user-defined functions

Pig Latin

- A Pig Latin statement is an operator that takes a relation as an input (except the **LOAD** operator) and produces another relation as output (except the **STORE** and **DUMP** operators)
- Pig operators such as **FILTER**, **FOREACH … GENERATE**, … are used to apply relational operations to an input relation
- A series of Pig Latin statements ending with a **DUMP** or **STORE** are executed in a **multi-query execution** manner (allowing optimizations)

Pig Latin

- A Pig script (a batch of Pig Latin statements) is organized as:
  - A LOAD statement describing the **location**, and using a **load function**; the **record interpretation** and the **schema** of the input relation
  - A series of transformations to process data such as: **FILTER, FOREACH … GENERATE, LIMIT, ORDER, JOIN, GROUP … BY**, …
  - A **DUMP** statement to view the results, or a **STORE** statement to save the results into the output **location**, and using a **store function**; in a described **format**

Pig Latin

- A Pig Latin statement ends with a semi-colon (;)
- A Pig operator is applied to a relation alias
- Pig scripts written in a file can be executed in batch mode using:

```
$ pig script.pig
```

- Pig statements can also be written into the interactive Pig shell, Grunt

```
$ pig

grunt> A = LOAD …;
grunt> DUMP A;
```

Pig Latin

```
-- example.pig: An example Pig scripts
-- run with the command: pig example.pig

-- Loading the data
A = LOAD ‘/data/students/all’ USING PigStorage(‘,’) AS
(name:chararray, age:int, gpa:float);

-- A projection operator
B = FOREACH A GENERATE name, gpa;

-- A selection operator
C = FILTER B BY gpa>=3.0;

-- Storing the data
STORE C into ‘/data/students/high_gpa’ USING PigStorage
```

Data Analysis with Pig

- Apache Pig Overview
- Pig Latin
- **Data Analysis with Pig**
- Pig Execution Modes

Data Analysis with Pig

- A LOAD operator should include a load function for Pig to interpret the input data as a collection of records
  - The default load function is PigStorage, which is used for input data (in HDFS) of lines of records, each of which are interpreted as tuples of a relation, where the fields within a record are delimited by a character (\t, by default)
  - There are many built-in load/store functions, such as JsonLoader/JsonStorage, TextLoader, HBaseStorage, …
  - Users can define their own load functions

Data Analysis with Pig

- Some Pig operators to transform data:
  - FILTER: Selection operator
  - FOREACH … GENERATE: Projection operator
  - GROUP, COGROUP
  - UNION, INTERSECTION, SPLIT
  - JOIN, OUTER JOIN
  - …

Data Analysis with Pig

- Some Pig operators to debug the Pig scripts:
  - DUMP: Displays a relation
  - DESCRIBE: Returns the schema of a relation
  - EXPLAIN: Displays the execution plan
  - ILLUSTRATE: Illustrates the execution of individual statements

Data Analysis with Pig

- Apache Pig Overview
- Pig Latin
- Data Analysis with Pig
- **Pig Execution Modes**

Pig Execution Modes

- The MapReduce jobs created from Pig statements can be executed in:
  - Local mode

```
$ pig –x local script.pig
```

  - MapReduce mode

```
$ pig script.pig
```

  - Tez mode

```
$ pig –x tez script.pig
```

Grunt

- Grunt is Pig's interactive shell
- It can be run in different execution modes:
  - Local mode

```
$ pig –x local

grunt>
```

  - MapReduce mode

```
$ pig

grunt>
```

  - Tez mode

```
$ pig –x tez

grunt>
```

Grunt

- The Grunt shell also provides access to the underlying filesystem (e.g. You can run run commands interacting with the HDFS within the grunt shell)
  - It also has the notion of working directory , so for example, `cd <dir>` works
- Of course, Pig operations can be performed within grunt shell, again with multi-query execution
  - That is, until a **STORE** or **DUMP** operation is performed, nothing actually is executed

Demo

Exploring a Pig Script

Demo

Using the Grunt Shell

Apache Pig Overview

End of Chapter