Apache Pig

Thaveewat Khanan

Introduction

A high-level platform for creating MapReduce programs Using Hadoop

Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

Pig Components

<u>Two Compnents</u>

Language (Pig Latin)

Compiler

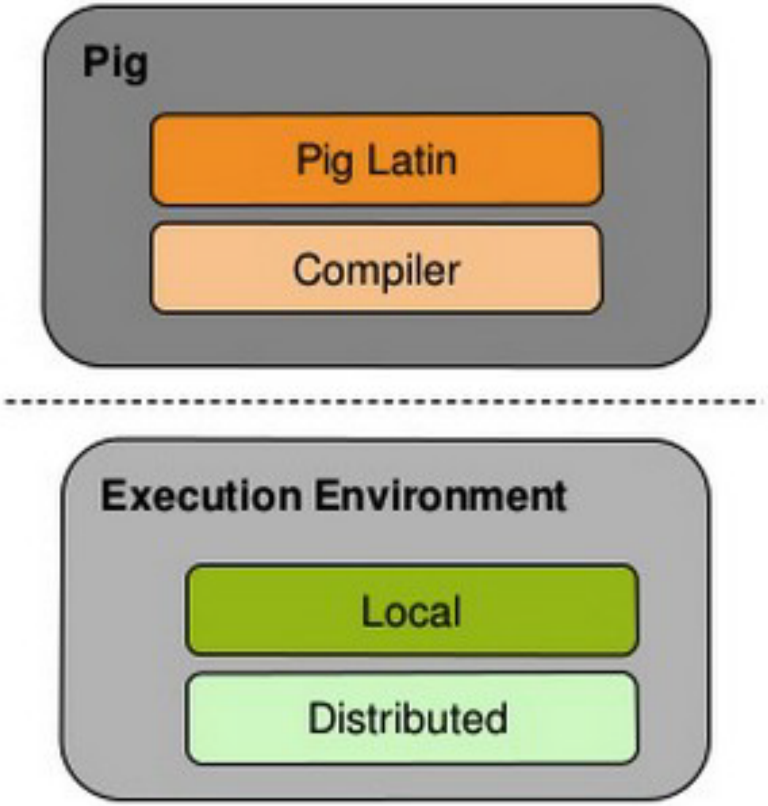
<u>Two Execution Environments</u>

Local

pig -x local

Distributed

pig -x mapreduce



Thaveewat Khanan

Running Pig

Script

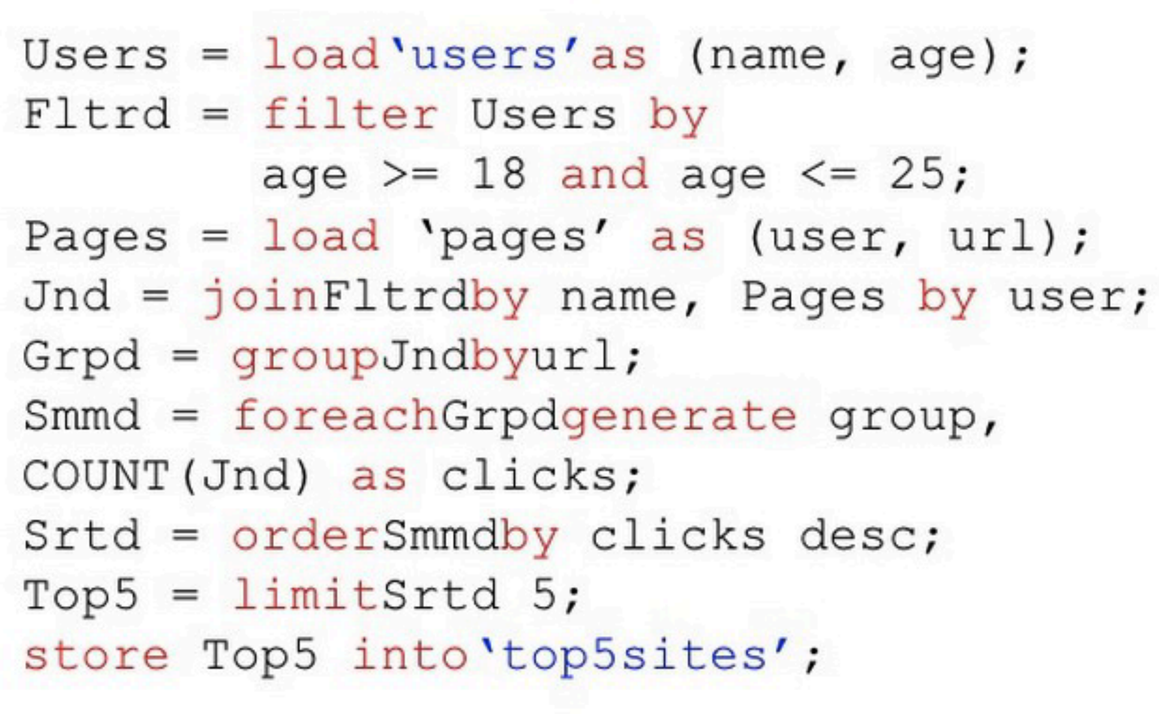
pig myscript

Command line (Grunt)

pig

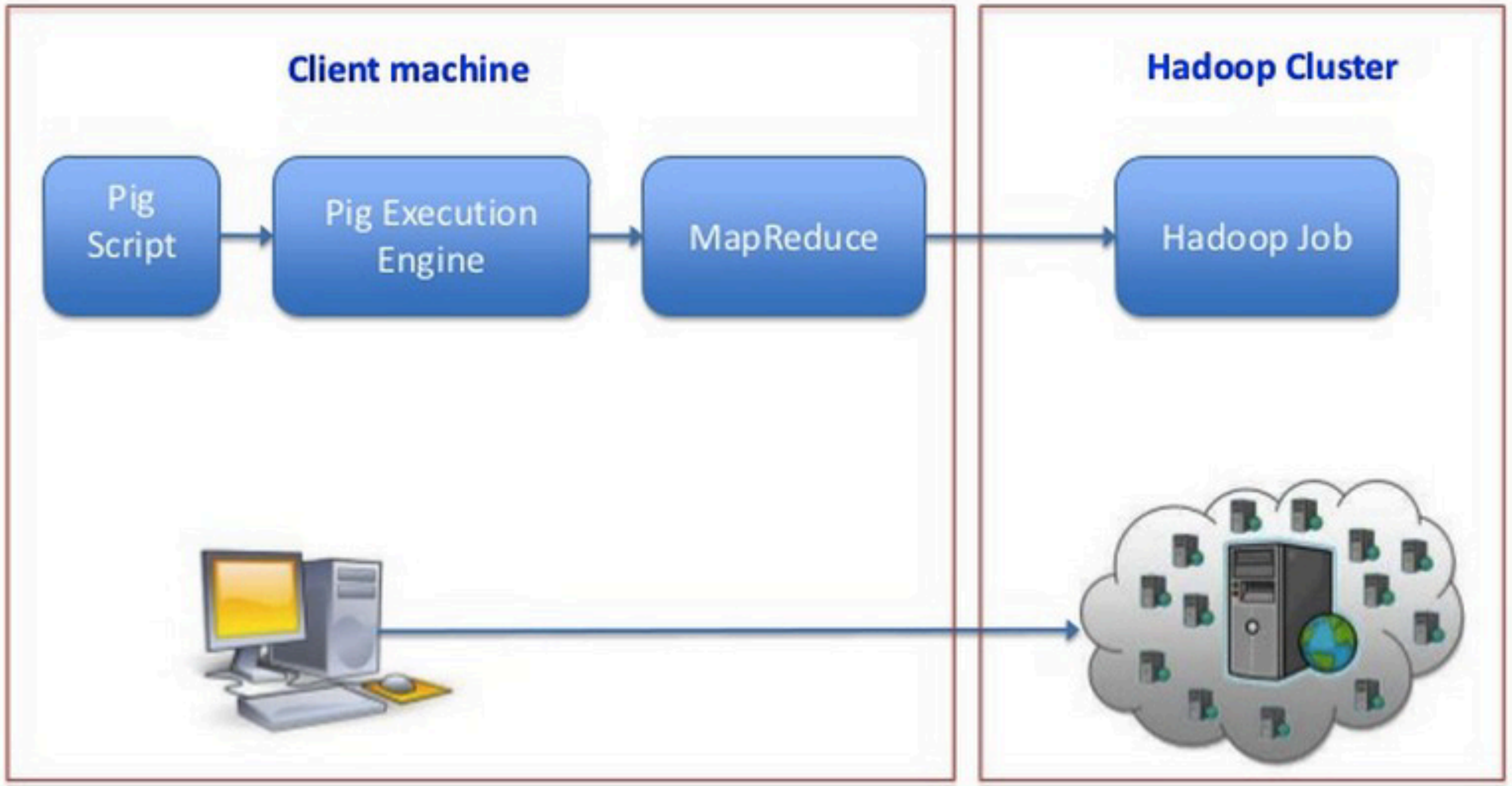
Embedded

Writing a java program



```
Users = load‘users’as (name, age);
Fltrd = filter Users by
          age >= 18 and age <= 25;
Pages = load ‘pages’ as (user, url);
Jnd = joinFltrdby name, Pages by user;
Grpd = groupJndbyurl;
Smmd = foreachGrpdgenerate group,
COUNT(Jnd) as clicks;
Srtd = orderSmmdby clicks desc;
Top5 = limitSrtd 5;
store Top5 into‘top5sites’;
```

Pig Execution Stages



Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

Why Pig?

Makes writing Hadoop jobs easier

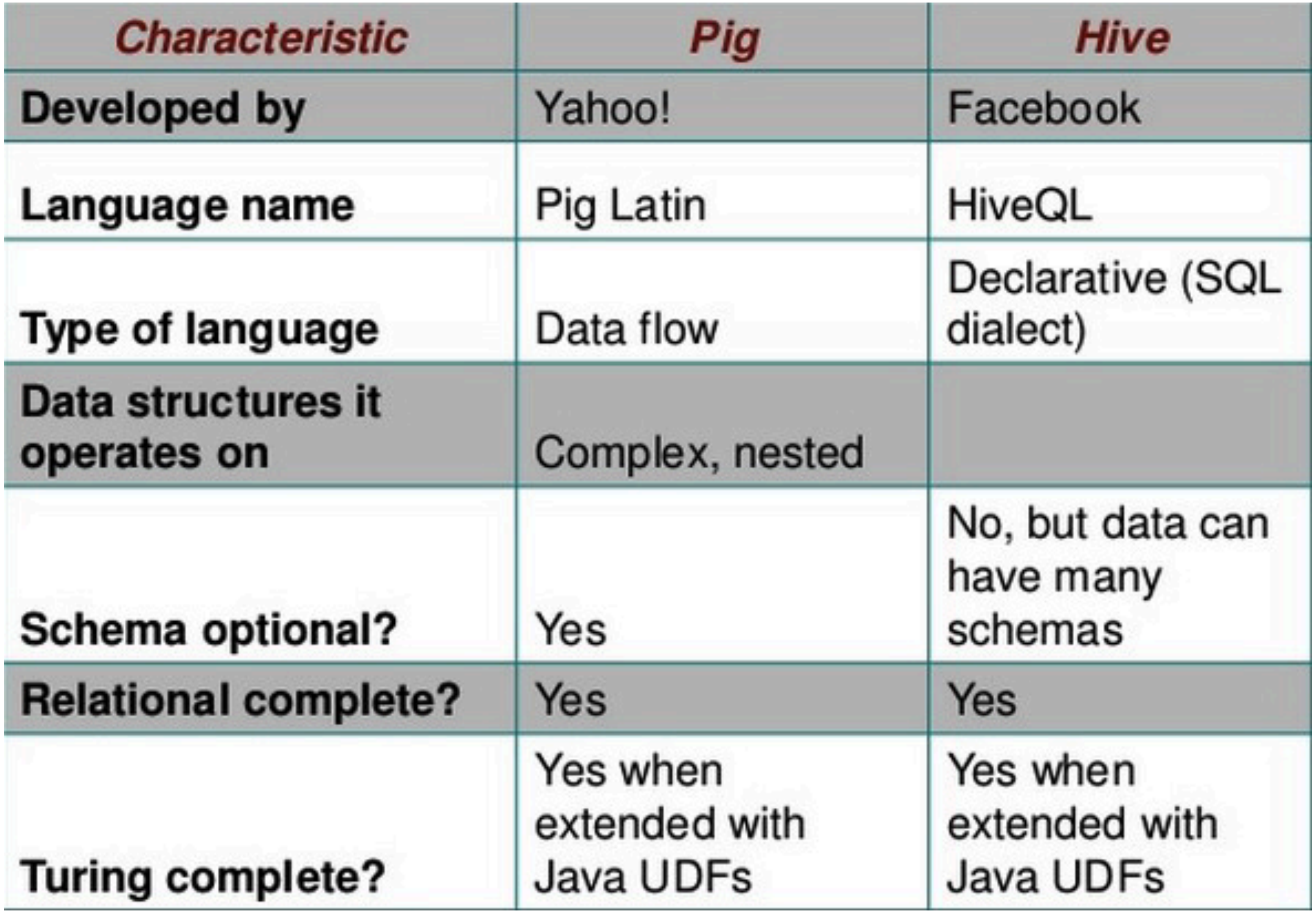
- 5% of the code, 5% of the time
- You don't need to be a programmer to write Pig scripts

Provide major functionality required for DatawareHouse and Analytics

- Load, Filter, Join, Group By, Order, Transform

User can write custom UDFs (User Defined Function)

Pig v.s. Hive



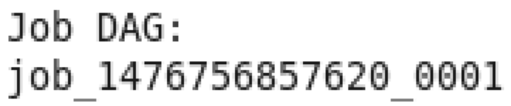
Characteristic	*Pig*	*Hive*
Developed by	Yahoo!	Facebook
Language name	Pig Latin	HiveQL
Type of language	Data flow	Declarative (SQL dialect)
Data structures it operates on	Complex, nested	
Schema optional?	Yes	No, but data can have many schemas
Relational complete?	Yes	Yes
Turing complete?	Yes when extended with Java UDFs	Yes when extended with Java UDFs

Running a Pig script

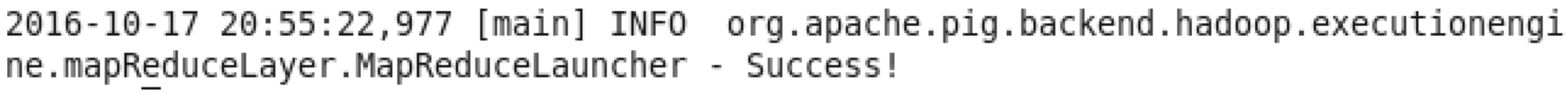
```
$ pig -x mapreduce
```

Writing a Pig Script for wordcount

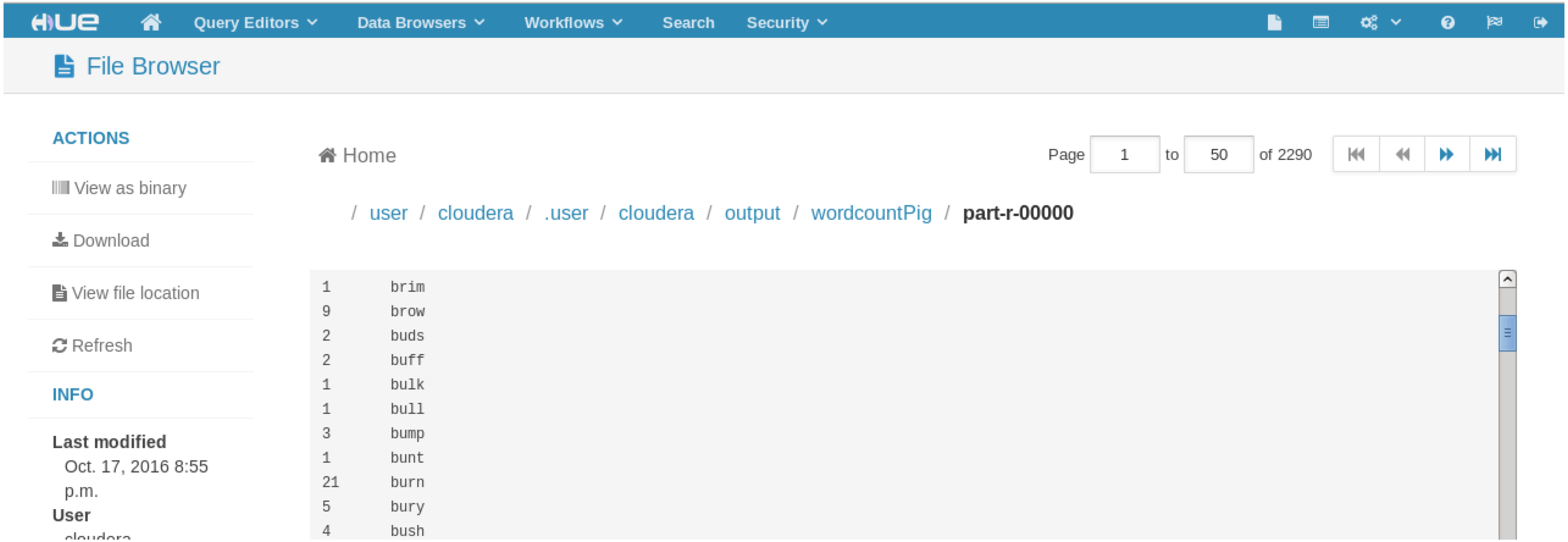
```
A = load '/user/cloudera/input/*';
B = foreach A generate flatten(TOKENIZE((chararray)$0)) as word;
C = group B by word;
D = foreach C generate COUNT(B), group;
store D into '.user/cloudera/output/wordcountPig';
```



```
Job DAG:
job_1476756857620_0001
```



```
2016-10-17 20:55:22,977 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
```



Page 1 to 50 of 2290

/ user / cloudera / .user / cloudera / output / wordcountPig / **part-r-00000**

```
1       brim
9       brow
2       buds
2       buff
1       bulk
1       bull
3       bump
1       bunt
21      burn
5       bury
4       bush
```

Last modified
Oct. 17, 2016 8:55 p.m.
User
cloudera