



Pig Latin

November 2020

Brolinskyi Sergii



Plan of presentation

- Definition Pig Latin
- Pig Latin vs classic MapReduce
- Pig Latin vs Hive
- Demo
- Summary

Pig Latin

- Pig Latin is a language game or argot in which English words are altered, usually by adding a fabricated suffix or by moving the onset or initial consonant or consonant cluster of a word to the end of the word and adding a vocalic syllable to create such a suffix.

"pig" = "igpay"

"latin" = "atinlay"

"banana" = "ananabay"

"will" = "illway"

"butler" = "utlerbay"

"happy" = "appyhay"

"duck" = "uckday"

"me" = "emay"

Pig Latin

- Pig Latin is a Pig's language that allows developers to express data flows (A language that interacts with a Pig tool)
- Pig is application environment used to run Pig Latin and convert Pig Latin scripts into MapReduce jobs

Pig
(Pig Latin)

Hive
(HiveQL)

MapReduce

Why Pig over MapReduce?

○ Fewer
lines of code

○ Quickly
test queries

○ No Java
experience

WordCount.java X

C:\> Users > sebrolin > OneDrive - Microsoft > Presentations > KPI > PigLatin > WordCount.java

```
1 // Example http://wiki.apache.org/hadoop/wordcount
2 package org.myorg;
3
4 import java.io.IOException;
5 import java.util.*;
6
7 import org.apache.hadoop.fs.Path;
8 import org.apache.hadoop.conf.*;
9 import org.apache.hadoop.io.*;
10 import org.apache.hadoop.mapreduce.*;
11 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
12 import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
13 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
14 import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
15
16 public class WordCount {
17
18     public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
19         private final static IntWritable one = new IntWritable(1);
20         private Text word = new Text();
21
22         public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
23             String line = value.toString();
24             StringTokenizer tokenizer = new StringTokenizer(line);
25             while (tokenizer.hasMoreTokens()) {
26                 word.set(tokenizer.nextToken());
27                 context.write(word, one);
28             }
29         }
30     }
31
32     public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
33
34         public void reduce(Text key, Iterable<IntWritable> values, Context context)
35             throws IOException, InterruptedException {
36             int sum = 0;
37             for (IntWritable val : values) {
38                 sum += val.get();
39             }
40             context.write(key, new IntWritable(sum));
41         }
42     }
43
44     public static void main(String[] args) throws Exception {
45         Configuration conf = new Configuration();
46
47         Job job = new Job(conf, "wordcount");
48
49         job.setOutputKeyClass(Text.class);
```

WordCount.java

wordCount.pig X

C: > Users > sebrolin > OneDrive - Microsoft > Presentations > KPI > PigLatin > wordCount.pig

```
1 input_lines = LOAD '/tmp/word.txt' AS (line: chararray);
2 words = FOREACH input_lines GENERATE FLATTEN(TOKENIZE(line)) AS word;
3 filtered_words = FILTER words BY word MATCHES '\\w+';
4 word_groups = GROUP filtered_words BY word;
5 word_count = FOREACH word_groups GENERATE COUNT(filtered_words) AS count, group AS word;
6 ordered_word_count = ORDER word_count BY count DESC;
7 STORE ordered_word_count INTO '/tmp/results.txt';
8
```


Pig history

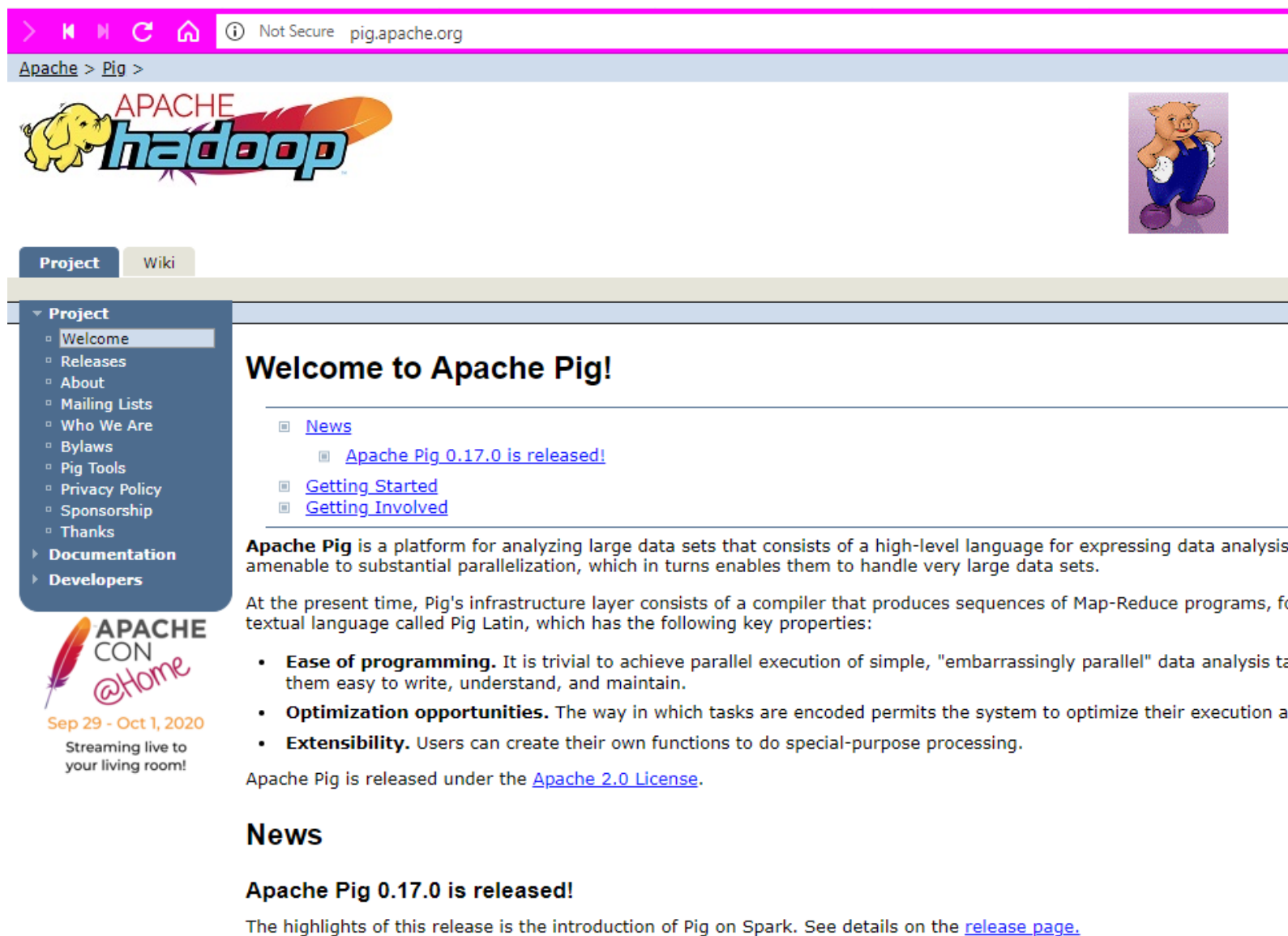
Yahoo

Large Datasets

SQL-Like

Apache

Pig docs (www.pig.apache.org)



The screenshot shows the Apache Pig website. At the top, there is a navigation bar with the Apache Hadoop logo on the left and a cartoon pig character on the right. Below the navigation bar, there are tabs for "Project" and "Wiki". The "Project" tab is active, and a sidebar menu is visible on the left. The main content area features a "Welcome to Apache Pig!" heading, followed by a list of links: "News", "Getting Started", and "Getting Involved". Below this, there is a paragraph describing Apache Pig as a platform for analyzing large data sets. A section titled "News" contains a sub-heading "Apache Pig 0.17.0 is released!" and a paragraph of text.

Apache > Pig >

APACHE HADOOP

Project Wiki

Project

- Welcome
- Releases
- About
- Mailing Lists
- Who We Are
- Bylaws
- Pig Tools
- Privacy Policy
- Sponsorship
- Thanks
- Documentation
- Developers

APACHE CON @Home
Sep 29 - Oct 1, 2020
Streaming live to your living room!

Welcome to Apache Pig!

- [News](#)
 - [Apache Pig 0.17.0 is released!](#)
- [Getting Started](#)
- [Getting Involved](#)

Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis amenable to substantial parallelization, which in turns enables them to handle very large data sets.

At the present time, Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs, for textual language called Pig Latin, which has the following key properties:

- Ease of programming.** It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks that are easy to write, understand, and maintain.
- Optimization opportunities.** The way in which tasks are encoded permits the system to optimize their execution.
- Extensibility.** Users can create their own functions to do special-purpose processing.

Apache Pig is released under the [Apache 2.0 License](#).

News

Apache Pig 0.17.0 is released!

The highlights of this release is the introduction of Pig on Spark. See details on the [release page](#).

Documentation (eg case sensitivity)

Case Sensitivity

The names (aliases) of relations and fields are case sensitive. The names of Pig Latin functions are case sensitive. The names of parameters (see [Parameter Substitution](#)) and all other Pig Latin keywords (see [Reserved Keywords](#)) are case insensitive.

In the example below, note the following:

- The names (aliases) of relations A, B, and C are case sensitive.
- The names (aliases) of fields f1, f2, and f3 are case sensitive.
- Function names PigStorage and COUNT are case sensitive.
- Keywords LOAD, USING, AS, GROUP, BY, FOREACH, GENERATE, and DUMP are case insensitive. They can also be written as load, using, as, group, by, etc.
- In the FOREACH statement, the field in relation B is referred to by positional notation (\$0).

```
grunt> A = LOAD 'data' USING PigStorage() AS (f1:int, f2:int, f3:int);
grunt> B = GROUP A BY f1;
grunt> C = FOREACH B GENERATE COUNT ($0);
grunt> DUMP C;
```

Hive vs Pig

HiveQL

- Declarative language based on SQL and schema bound



Pig Latin

- Procedural or data flow programming language with ability to declare schema at runtime





DEMO TIME

```
a = LOAD 'cereal.csv' AS (name:chararray, calories:int);  
b = FOREACH a GENERATE name;  
DUMP b;
```

Example Expression

```
Int 32-bit           → 5
Long 64-bit          → 5L
Float 32-bit float   → 5.5f
Double 64-bit        → 5.5
```

Numeric Types

4 different numeric types

Inherited from Java

Chararray character string → “some text”

Text Data Type

java.lang.string


```
datetime → 1981-07-26T00:00:00.000+00:00
```

DatetimeType

```
bytearray Byte array (blob)
```

Binary Data Type

Java class DataByteArray

```
tuple ordered list of fields → (7,26)
Bag collection of tuples → {(7,26), (9,5)}
Map set of key value pairs → [somekey#somevalue]
```

Complex Data Type

Addition $\rightarrow + \rightarrow a + b$

Subtraction $\rightarrow - \rightarrow a - b$

Multiplication $\rightarrow * \rightarrow a * b$

Division $\rightarrow / \rightarrow a / b$

Arithmetic Operators

Equal $\rightarrow a == b$

Not Equal $\rightarrow a != b$

Greater than $\rightarrow a > b$, $a >= b$

Less than $\rightarrow a < b$, $a <= b$

Comparison Operators

AND → `a == 10 and b == 12`

OR → `a == 10 or b == 12`

Boolean Operators

NASDAQ 100 Index

Date	Open	High	Low	Close	Volume	Adj Close
2015-03-06	44	45	42	45	190000	45
--	--	--	--	--	--	--
--	--	--	--	--	--	--
--	--	--	--	--	--	--
--	--	--	--	--	--	--

Relational operators

- Limit
- Group
- Filter
- Foreach

Limit

```
x = Limit stock 10;
```

Group

```
x = GROUP stock BY high;
```

Filter


```
x = FILTER stock BY closing > 43;
```

Foreach

```
x = FOREACH stock GENERATE (high, low, close);
```

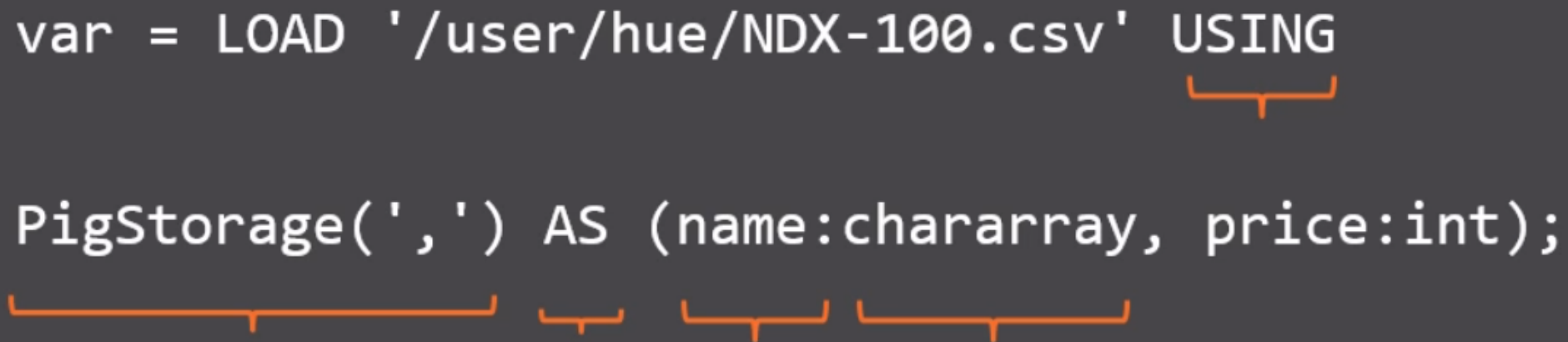
Load

```
var = LOAD '/tmp/datafile.csv' USING PigStorage(',')  
      (name:chararray, price:int);
```




Load Using PigStorage

```
var = LOAD '/user/hue/NDX-100.csv' USING  
PigStorage('','') AS (name:chararray, price:int);
```



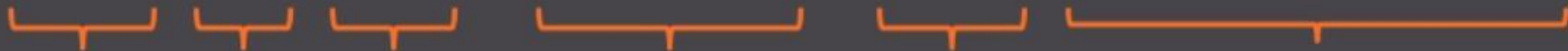
Load Using PigStorage

```
var = LOAD '/user/hue/NDX-100.csv' USING  
PigStorage(',') AS (name:chararray, price:int);
```

The image shows a Pig Latin script snippet with orange brackets highlighting parts of the PigStorage function call. The first line is 'var = LOAD '/user/hue/NDX-100.csv' USING'. The second line is 'PigStorage(',') AS (name:chararray, price:int);'. An orange bracket is under 'USING' in the first line. In the second line, orange brackets are under the opening parenthesis '(', the closing parenthesis ')', the field definition 'name:chararray', and the field definition 'price:int'.

Store Using PigStorage

```
STORE var INTO 'filename' USING PigStorage(',');
```



Describe

```
var = LOAD '/tmp/datafile.csv' AS
```

```
(name:chararray, price:int);
```

```
DESCRIBE var;
```

Explain

```
var = LOAD '/tmp/datafile.csv' AS
```

```
(name:chararray, price:int);
```

```
EXPLAIN var;
```

Illustrate

```
var = LOAD '/tmp/datafile.csv' AS
```

```
(name:chararray, price:int);
```

```
ILLUSTRATE var;
```

Summary

Pig allows analysts with no Java experience, but some SQL background easily run MapReduce jobs on a Hadoop cluster.

Pig latin is an ETL (extract transform load) script language to be run on Pig

