Apache Pig

- **Introducing Pig -** describe Pig characteristics and elements
- **Pig Latin Foundations -** Describe Pig features, data flow, and data types
- **Pig Examples -** Examine a common Pig script example

Introducing Pig

- High-level processing layer that runs on Hadoop
- Language for expressing data analysis and infrastructure processes
- Uses both HDFS (read and write files) and MapReduce (execute jobs)

Introducing Pig

- Pig Latin
  - Provides a high-level language that makes it easy to create the MapReduce jobs that run on the Hadoop cluster
  - Requires no metadata or schema
  - Statements translated into series of MapReduce jobs

- Grunt - Interactive Shell

- Piggy Bank - Shared repository for user-defined functions

Pig Latin Foundations

Pig Latin Foundations

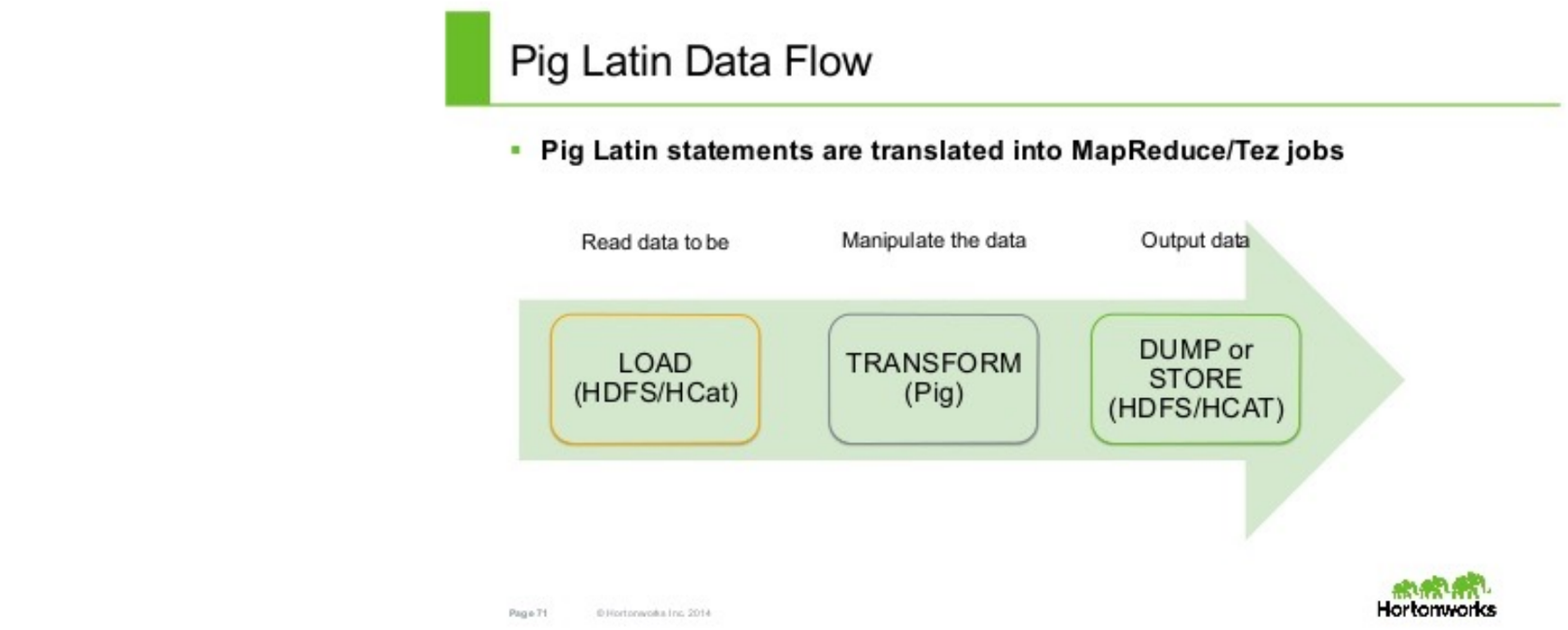
Pig Latin Features

- Language for expressing data analysis and infrastructure processes
- Supports traditional data processing operations (join, sort, filter, etc…)
- Simplifies joining data and chaining jobs together

Pig Latin Foundations

Pig Latin Data Flow

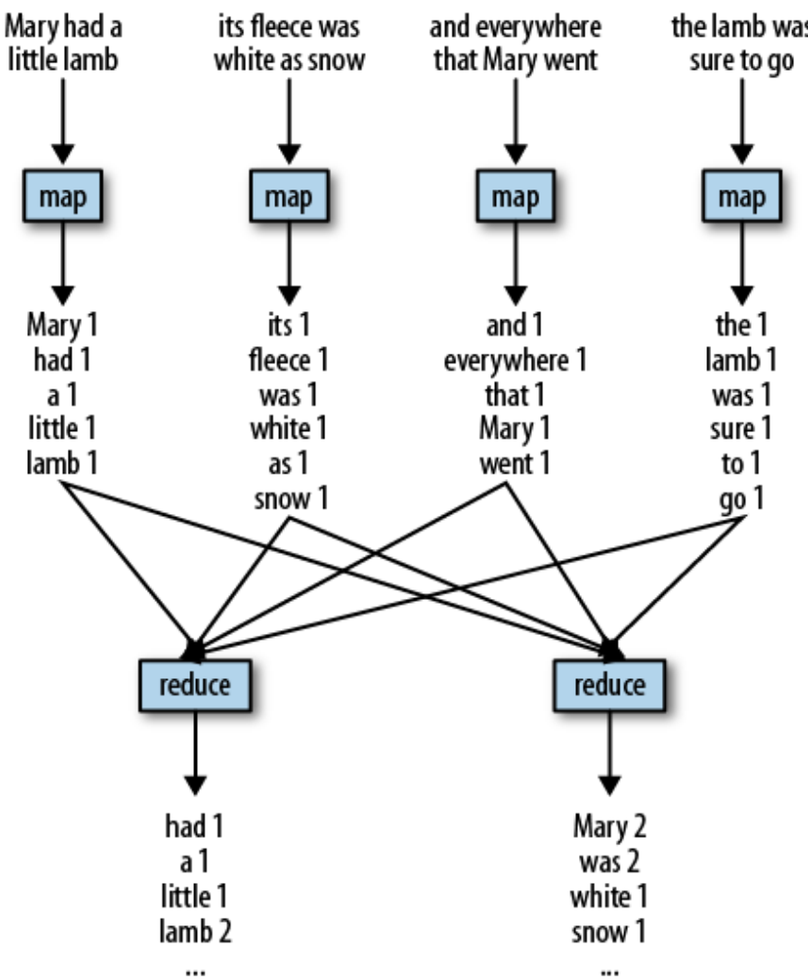
- Pig Latin statements are translated into MapReduce jobs.
- HCatalog can be used to handle metadata and pull in data to be processed by Pig.



Pig Latin Foundations

Pig Latin Data Flow

Pig Latin script describes a DAG (Directed Acyclic Graph)



Pig Latin Foundations

From Script to MapReduce

- Pig interpreter processes each task
- Valid jobs are added to a plan created by interpreter
- Steps in plan are executed only when a DUMP or STORE command is reached

Pig Latin Foundations

From Script to MapReduce

```
grunt> LOGS = LOAD 'sample.log'
grunt> LEVELS = foreach LOGS generate
REGEX_EXTRACT($0,
'(TRACE|DEBUG|INFO|WARN|ERROR|FATAL)',1) as LOGLEVEL;
grunt> dump LEVELS;
```

Pig Relations

- Bag of tuples
- Similar to a table in a Relational Database
- EXCEPT, tuples don't have to have the same "schema"

Pig Data Types

- **Field**: holds piece of data
- **Tuple**: ordered set of fields
  - `("Apples", "ERROR", 27)`
- **Bag**: unorded collection of tuples
  - {
  - ("Apples", "ERROR", 27),
  - ("Cheese", "SUCCESS", 28)
  - }
- **Map**: collection of key-value pairs
  - `[firstName#Cary, lastName#Grant, id#123]`

Pig Example

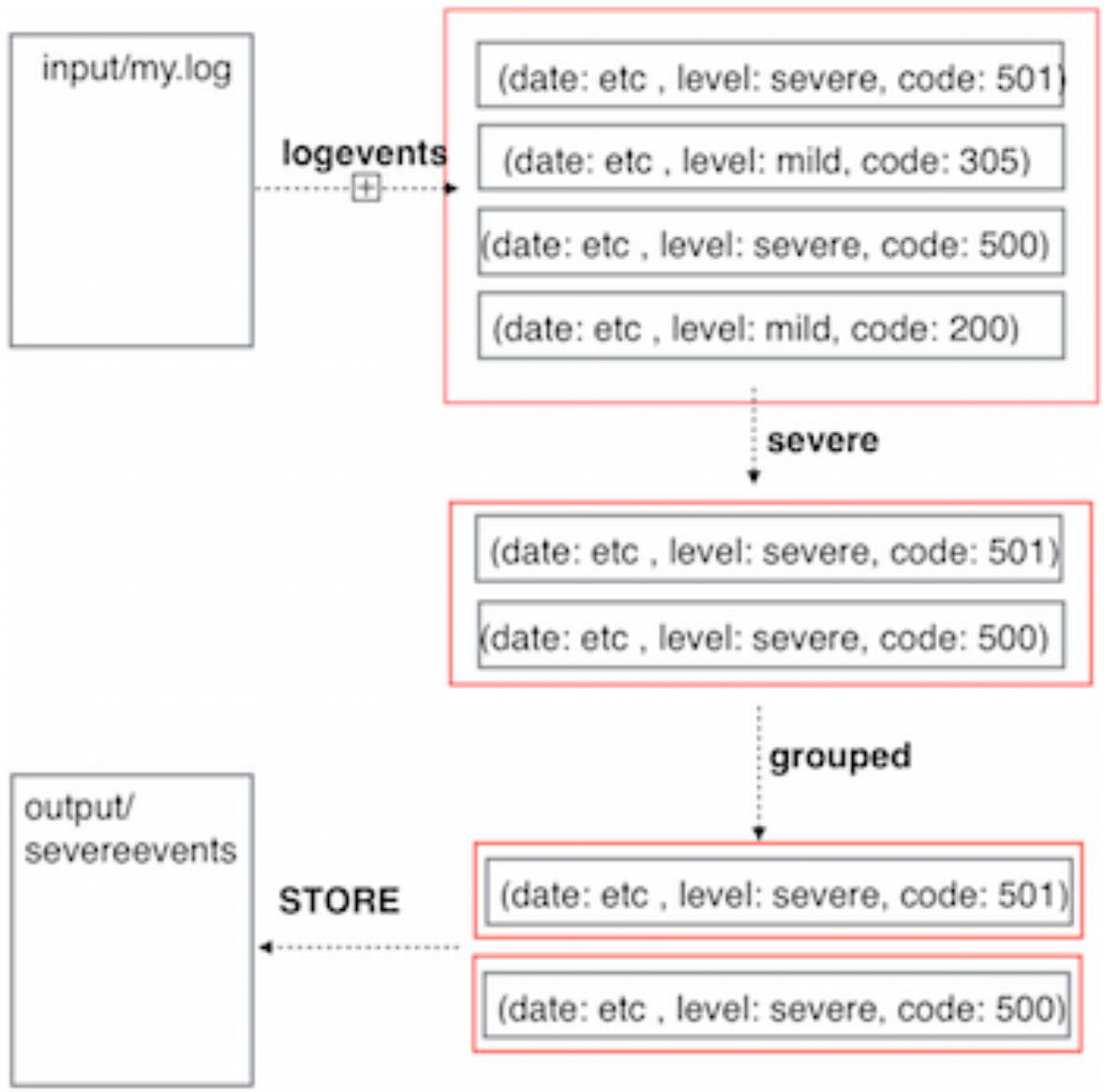
Pig Example: FILTER and GROUP

```
~> logevents = LOAD 'input/my.log' AS
(date:chararray,
    level:chararray, code:int, message:charray);

~> severe = FILTER logevents BY (level == 'severe'
                                 AND code >= 500);

~> grouped = GROUP severe BY code;

~> STORE grouped INTO ‘output/severeevents';
```

- Super straight-forward, easy peasy

Pig Example: FILTER and GROUP

What's happening from last slide



References

Author: Hortonworks. (2013, June 3rd) *Hadoop Tutorial: Apache Pig.*
Retrieved From: https://www.youtube.com/watch?v=PQb9l-8986s