Pig

Igor Yakushin
ivy2@uchicago.edu

Pig: introduction

- High level language - **Pig Latin**
- Compiler translates Pig Latin into MapReduce jobs
- It is a **dataflow language** where you define a data stream and a set of transformations applied to it.
- Operations: load, store, dump, filter, foreach, group, join, order by, distinct, limit, sample, etc.
- You can specify data types to help Pig to optimize a program or you can let it figure it out

Pig: how to run

Pig programs can run in three ways:

- as a script
  - on a local computer

```
pig -x local milesPerCarrier.pig
```

  - on a cluster

```
pig -x mapreduce milesPerCarrier.pig
```

- interactively using Grunt interpreter

```
pig -x local
```

- Embedded in other languages such as Java, Python, JavaScript

Pig: examples

```
records = LOAD 'pig/words.csv' USING PigStorage(',') AS (W, N:int);
mrecs = GROUP records ALL;
tot = FOREACH mrecs GENERATE SUM(records.N);
DUMP tot;
```

```
in = load 'pig/mary.txt' as (line);
-- TOKENIZE splits the line into a field for each word.
-- flatten will take the collection of records returned by
-- TOKENIZE and produce a separate record for each one, calling the single
-- field in the record word.
words = foreach in generate flatten(TOKENIZE(line)) as word;
grpd = group words by word;
cntd = foreach grpd generate group, COUNT(words);
store cntd into 'cntd.out';
```

```
records = LOAD 'words.csv' USING PigStorage(',') AS (W, N:int);
r1 = filter records by N < 10;
dump r1;
r2 = foreach r1 generate N*N as N2, W;
describe r2;
r3 = join r1 by W, r2 by W;
```