CC5212-1

Procesamiento Masivo de Datos

Otoño 2020

Lecture 4.5

Projects, Practice with Pig/Hadoop

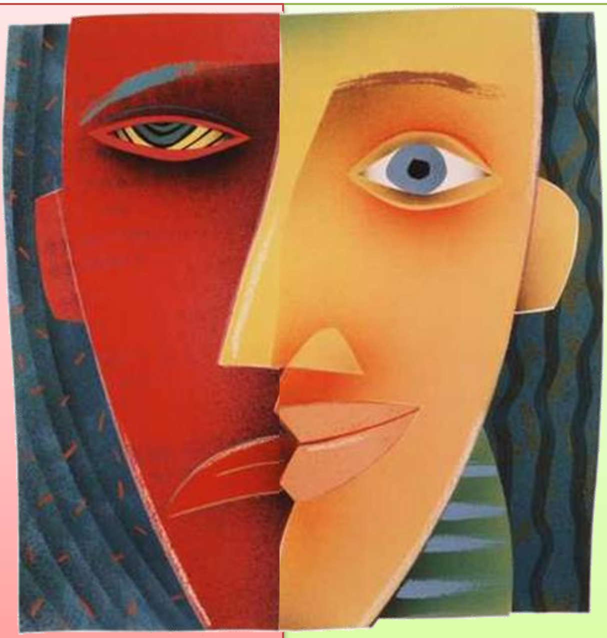
Aidan Hogan
aidhog@gmail.com

Course Marking (Revised)

- 75% for Weekly Labs (~9% a lab)
  - 4/4 obligatory, 4/7 optional
- 25% for Class Project
- Need to pass in overall grade

Assignments each week

Working in groups



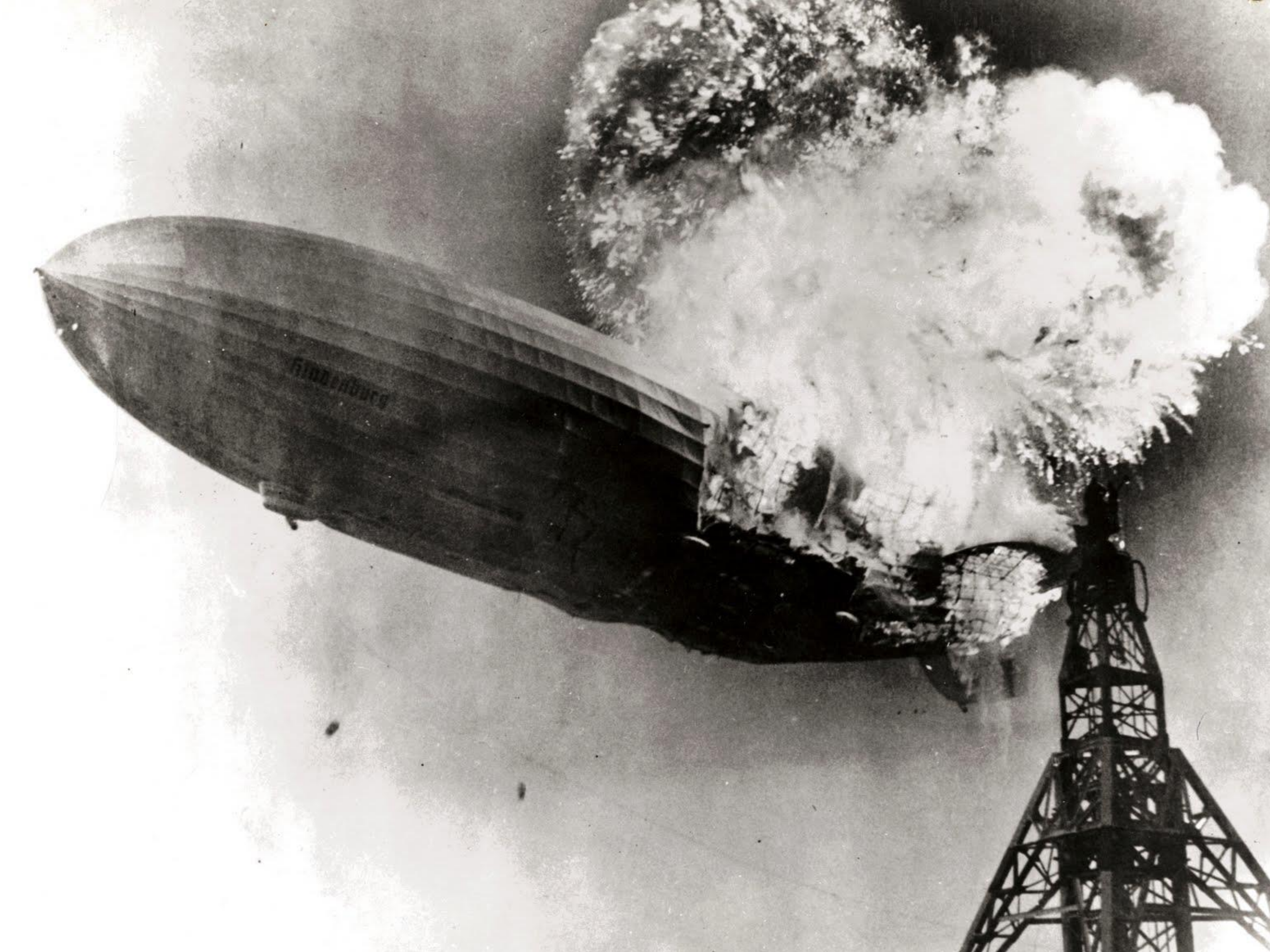
Hands-on each week!

Working in groups!

Class Projects

Class Project

- Done in threes
- Goal: Use what you've learned to do something cool/fun (hopefully)
- Process:
  - Form groups of three (in the forum, before April 30th)
  - On April 30th we will assign the rest automatically
  - Start thinking up topics / find interesting datasets!
  - Register topic (deadline around May 21st)
  - Work on projects during semester
  - Deliverables will due be around week 13
- Deliverables: 4 minute presentation (video) & short report
- Marked on: Difficulty, appropriateness, scale, good use of techniques, presentation, coolness, creativity, value
  - Ambition is appreciated, even if you don't succeed


Hindenburg

Desiderata for project

- Must focus around some technique from the course!

- Expected difficulty: similar to a lab, but without any instructions

- Data not too small:
  - Should have >250,000 tuples/entries

- Data not too large:
  - Should have <1,000,000,000 tuples/entries
  - If very large, perhaps take a sample?

- In case of COVID-19 data, we can make exceptions

Where to find/explore data?

- Kaggle:
  - https://www.kaggle.com/

- Google Dataset Search:
  - https://datasetsearch.research.google.com/

- Datos Abiertos de Chile:
  - https://datos.gob.cl/
  - https://es.datachile.io/

- …

Practice with Hadoop/Pig

Practice with Hadoop

- Optional Assignment 1 (not evaluated):
  - Hadoop: Find the number of good movies in which each actor/actresses has starred.
  - Good movie: ≥ 10001 votes, score ≥ 7.8
  - Separate outputs for actors/actresses
  - Lab 4 in Hadoop basically!

Practice with Hadoop and/or Pig

- Optional Assignment 2 (not evaluated):
  - Hadoop and/or Pig: Find movies with only actors, or only actresses, and order by rating (descending)
  - You can choose if you wish to do only actors, or only actresses, or both

Hadoop: Multiple Maps, One Reduce

Hadoop: Supermarket Example

ReceiptItems

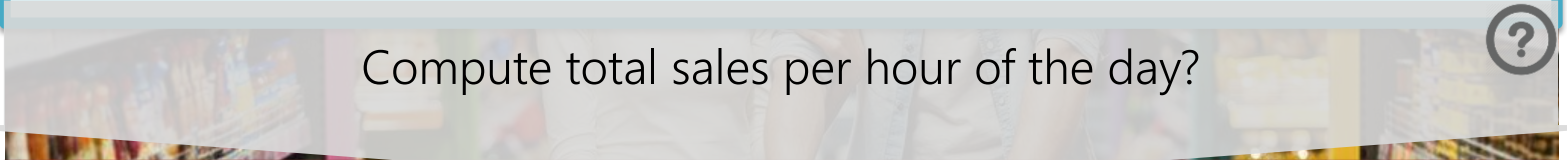
Receipt ID	Item ID
R1401	I306
R1401	I306
R1401	I504
R1402	I007
R1402	I306
R1403	I306
R1403	I504
...	...

ReceiptTimes

Receipt ID	Time
R1403	19:00
R1401	18:59
R1402	19:01
...	...

ItemDetails

Item ID	Name	Price ($)
I306	Zanahoria 500g	500
I504	CocaCola 3L	1400
I007	Comfort	1200
...	...	

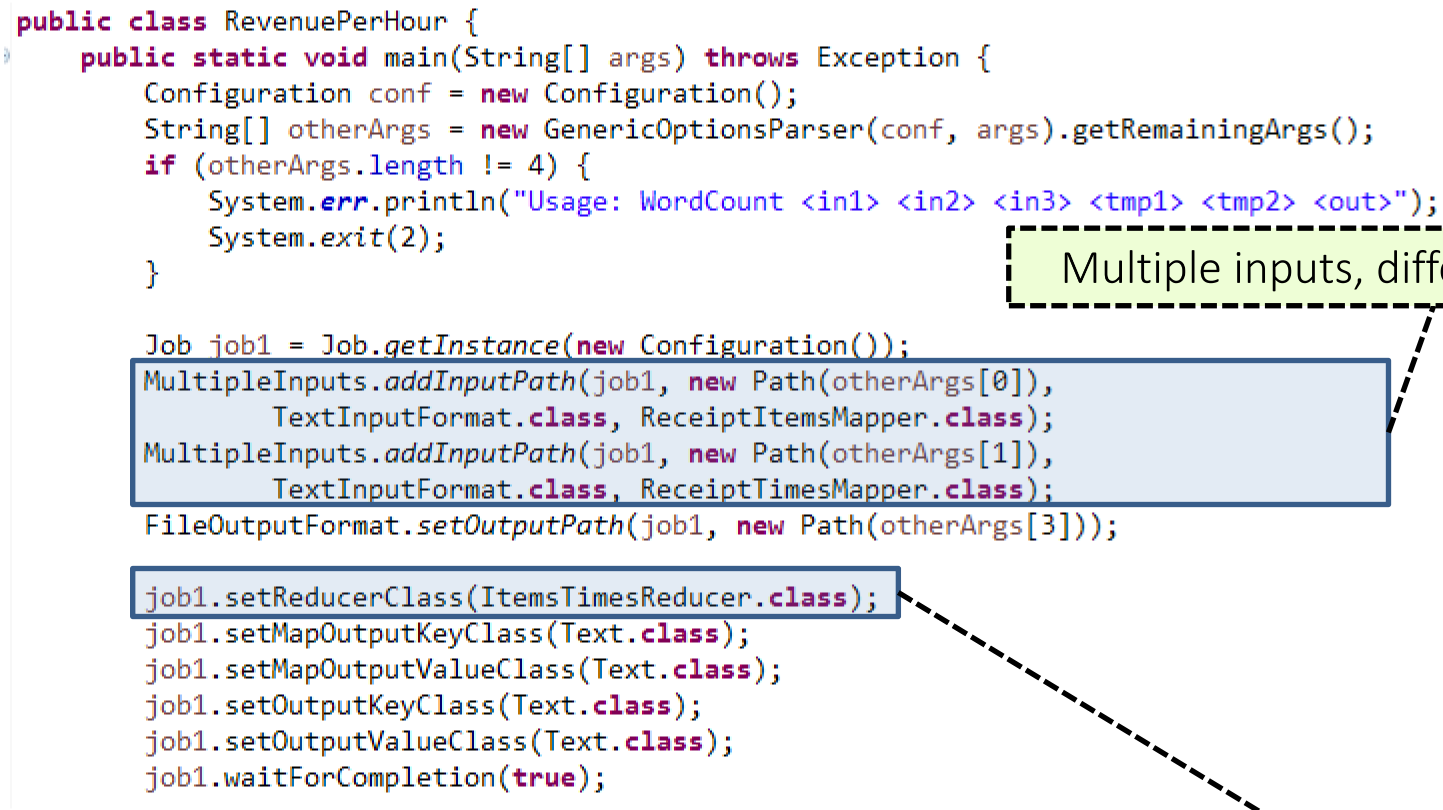


Compute total sales per hour of the day?

Output

Hour	Total
...	...
18:00–18:59	$2400
19:00–19:59	$3600
...	...

More in Hadoop: Multiple Maps, One Reduce

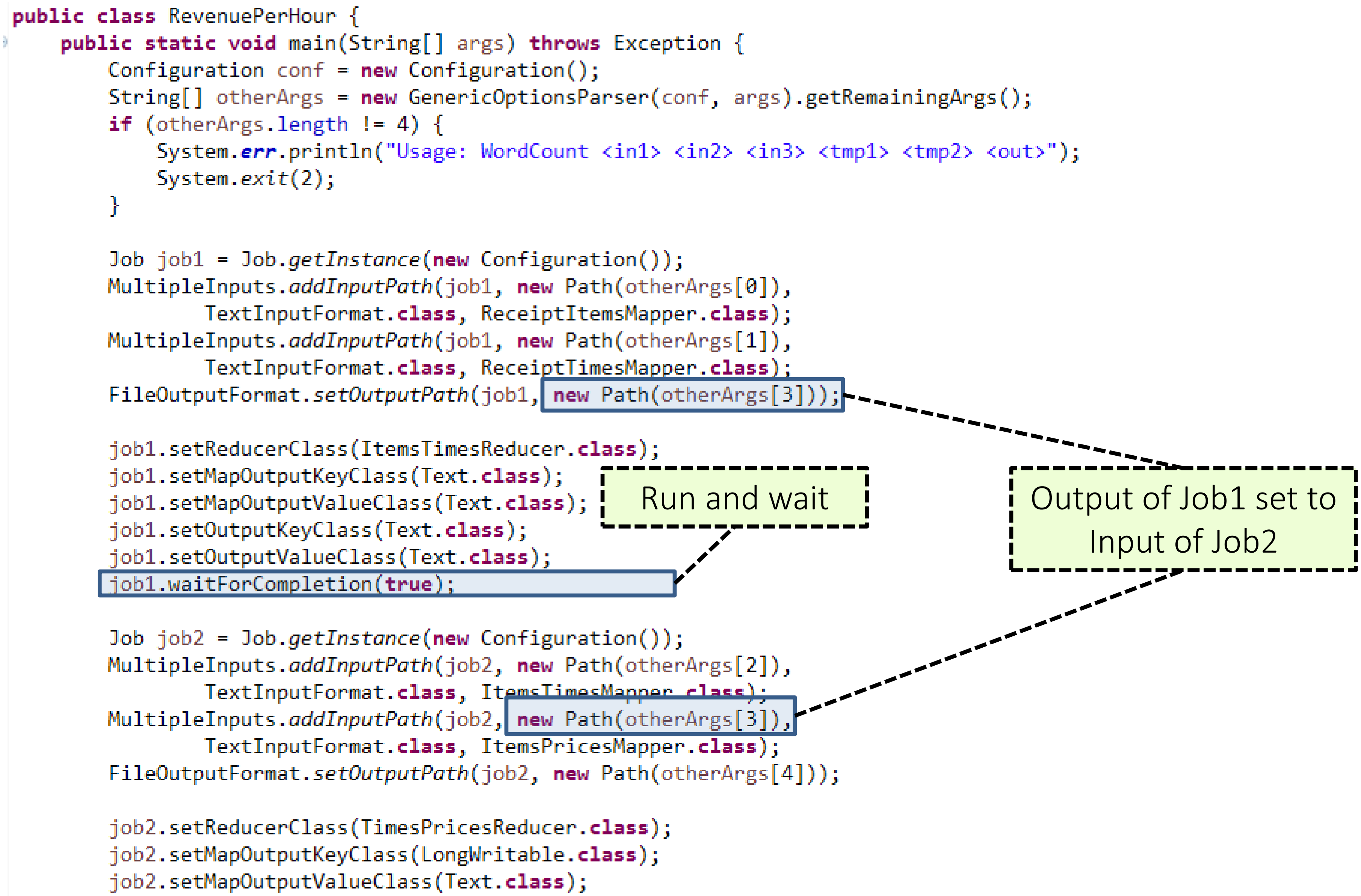


```
public class RevenuePerHour {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        if (otherArgs.length != 4) {
            System.err.println("Usage: WordCount <in1> <in2> <in3> <tmp1> <tmp2> <out>");
            System.exit(2);
        }

        Job job1 = Job.getInstance(new Configuration());
        MultipleInputs.addInputPath(job1, new Path(otherArgs[0]),
                TextInputFormat.class, ReceiptItemsMapper.class);
        MultipleInputs.addInputPath(job1, new Path(otherArgs[1]),
                TextInputFormat.class, ReceiptTimesMapper.class);
        FileOutputFormat.setOutputPath(job1, new Path(otherArgs[3]));

        job1.setReducerClass(ItemsTimesReducer.class);
        job1.setMapOutputKeyClass(Text.class);
        job1.setMapOutputValueClass(Text.class);
        job1.setOutputKeyClass(Text.class);
        job1.setOutputValueClass(Text.class);
        job1.waitForCompletion(true);
```

Multiple inputs, different map for each

One reducer

More in Hadoop: Chaining Jobs



```java
public class RevenuePerHour {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        if (otherArgs.length != 4) {
            System.err.println("Usage: WordCount <in1> <in2> <in3> <tmp1> <tmp2> <out>");
            System.exit(2);
        }

        Job job1 = Job.getInstance(new Configuration());
        MultipleInputs.addInputPath(job1, new Path(otherArgs[0]),
                TextInputFormat.class, ReceiptItemsMapper.class);
        MultipleInputs.addInputPath(job1, new Path(otherArgs[1]),
                TextInputFormat.class, ReceiptTimesMapper.class);
        FileOutputFormat.setOutputPath(job1, new Path(otherArgs[3]));

        job1.setReducerClass(ItemsTimesReducer.class);
        job1.setMapOutputKeyClass(Text.class);
        job1.setMapOutputValueClass(Text.class);
        job1.setOutputKeyClass(Text.class);
        job1.setOutputValueClass(Text.class);
        job1.waitForCompletion(true);

        Job job2 = Job.getInstance(new Configuration());
        MultipleInputs.addInputPath(job2, new Path(otherArgs[2]),
                TextInputFormat.class, ItemsTimesMapper.class);
        MultipleInputs.addInputPath(job2, new Path(otherArgs[3]),
                TextInputFormat.class, ItemsPricesMapper.class);
        FileOutputFormat.setOutputPath(job2, new Path(otherArgs[4]));

        job2.setReducerClass(TimesPricesReducer.class);
        job2.setMapOutputKeyClass(LongWritable.class);
        job2.setMapOutputValueClass(Text.class);
```

Run and wait

Output of Job1 set to Input of Job2

More in Hadoop: Number of Reducers

```
job.setNumReduceTasks(1);
```

Set number of parallel reducer tasks for the job

Why would we ask for 1 reduce task?

Output requires a merge on one machine
(for example, sorting, top-*k*)

Questions?