# Map/Reduce

Обзор решений

Алексей Злобин
alexey.zlobin@gmail.com

# Sample job: driver

```
public static void main(String[] a) throws Exception {
  Configuration conf = new Configuration();
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  job.setReducerClass(Reduce.class);
  job.setInputFormatClass(TextInputFormat.class);
  job.setOutputFormatClass(TextOutputFormat.class);
  FileInputFormat.addInputPath(job, new Path(a[0]));
  FileOutputFormat.setOutputPath(job, new Path(a[1]));
  job.waitForCompletion(true);
}
```

# Sample job: mapper

```
class M extends Mapper<LongWritable, Text, Text, IntWritable> {
  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(LongWritable k, Text v, Context ctx) {
    String line = v.toString();
    StringTokenizer tokenizer = new StringTokenizer(line);
    while (tokenizer.hasMoreTokens()) {
      word.set(tokenizer.nextToken());
      ctx.write(word, one);
    }
  }
}
```

# Sample job: reducer

```
class R extends Reducer<Text, IntWritable, Text, IntWritable> {
  public void reduce(Text k, Iterable<IntWritable> v, Context ctx)
{
    int sum = 0;
    for (IntWritable val : v)
      sum += val.get();
    context.write(k, new IntWritable(sum));
  }
}
```

# Pig snippet

```
raw =
  LOAD 'excite.log' USING PigStorage('\t') AS (user, time, qry);
clean1 = FILTER raw BY
  org.apache.pig.tutorial.NonURLDetector(qry);

clean2 = FOREACH clean1
  GENERATE user, time, org.apache.pig.tutorial.ToLower(qr)
  as query;
```

# Hive snippet

```
CREATE TABLE invites
   (foo INT, bar STRING) PARTITIONED BY (ds STRING);

LOAD DATA LOCAL
  INPATH './examples/files/kv2.txt' OVERWRITE
  INTO TABLE invites PARTITION (ds='2008-08-15');

SELECT a.foo
  FROM invites a
  WHERE a.ds='2008-08-15';

INSERT OVERWRITE DIRECTORY '/tmp/reg_5'
  SELECT a.foo, a.bar FROM invites a;
```

# Spark: example

```
val counts = lines.flatMap(line => line.split(“ “))
                  .map(word => (word, 1))
                  .reduceByKey(_ + _)
```

# Shark example

```
CREATE TABLE src(key INT, value STRING);

LOAD DATA LOCAL INPATH '${env:HIVE_HOME}/examples/files/kv1.txt'
  INTO TABLE src;

SELECT COUNT(1) FROM src;

CREATE TABLE src_cached AS SELECT * FROM SRC;

SELECT COUNT(1) FROM src_cached;
```

# Disco example

```
def fun_map(line, params):
        for word in line.split():
                yield word, 1


def fun_reduce(iter, params):
        for word, counts in kvgroup(sorted(iter)):
                yield word, sum(counts)
```

# Disco driver

```
job = Job().run(
  input=["http://discoproject.org/media/text/chekhov.
txt"],
  map=map,
  reduce=reduce)
for word, count in result_iterator(job.wait(show=True)):
  print(word, count)
```

# References I

- “MapReduce: Simplified Data Processing on Large Clusters” Dean, Jeffrey and Ghemawat, Sanjay
- “A Comparison of Join Algorithms for Log Processing in MapReduce” S. Blanas, J. Patel, V. Ercegovac, J. Rao, E. Shekita, Y. Tian
- “Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing” Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica
- “Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters” Matei Zaharia, Tathagata Das, Haoyuan Li, Scott Shenker, Ion Stoica
- “Shark: Fast Data Analysis Using Coarse-grained Distributed Memory” Cliff Engle, Antonio Lupher, Reynold Xin, Matei Zaharia, Haoyuan Li, Scott Shenker, Ion Stoica

# References II

- Disco Technical Overview http://disco.readthedocs.org/en/latest/overview.html
- Disco Distributed Filesystem http://disco.readthedocs.org/en/latest/howto/ddfs.html
- An efficient, immutable, persistent mapping object http://discodb.readthedocs.org/en/latest/