

Lots of Data, Little Money. A Last.fm perspective

Martin Dittus, martind@last.fm
London Geek Nights, 2009-04-23

Big Data
Little Money

- You have lots of data
- You want to process it
 - For your product (Last.fm: Charts)
 - For internal use, metrics (to count stuff, do simple analyses)

“Big Data”

- Apache logs
- Transaction logs
- Other application data
- Last.fm: Scrobbles

(How many in the audience are storing >1TB of data?)

“Little Money” for Powerful Hardware

- Open source RDBMS: single node
- Limits to what a single machine can achieve
- Tried clustering, sharding... you’ve heard it before

“Little Money” for Powerful Hardware

- Final realisation:
 - We don't need realtime updates of scrobbles
 - Writing to a flat file is much faster than writing to a DB
 - We just need a good batch processing system

Enter Hadoop

```
public void map(LongWritable key, Text value,
    OutputCollector<Text, IntWritable> output,
    Reporter reporter) throws IOException {

    String line = value.toString();
    StringTokenizer tokenizer = new StringTokenizer(line);
    while (tokenizer.hasMoreTokens()) {
        word.set(tokenizer.nextToken());
        output.collect(word, one);
    }
}

public void reduce(Text key, Iterator<IntWritable> values,
    OutputCollector<Text, IntWritable> output,
    Reporter reporter) throws IOException {

    int sum = 0;
    while (values.hasNext()) {
        sum += values.next().get();
    }
    output.collect(key, new IntWritable(sum));
}
```

```

package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class WordCount {

    public static class Map extends MapReduceBase
        implements Mapper<LongWritable, Text, Text, IntWritable> {

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value,
            OutputCollector<Text, IntWritable> output,
            Reporter reporter) throws IOException {

            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                word.set(tokenizer.nextToken());
                output.collect(word, one);
            }
        }
    }

    public static class Reduce extends MapReduceBase
        implements Reducer<Text, IntWritable, Text, IntWritable> {

        public void reduce(Text key, Iterator<IntWritable> values,
            OutputCollector<Text, IntWritable> output,
            Reporter reporter) throws IOException {

            int sum = 0;
            while (values.hasNext()) {
                sum += values.next().get();
            }
            output.collect(key, new IntWritable(sum));
        }
    }

    public static void main(String[] args) throws Exception {
        JobConf conf = new JobConf(WordCount.class);
        conf.setJobName("wordcount");

        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        conf.setMapperClass(Map.class);
        conf.setCombinerClass(Reduce.class);
        conf.setReducerClass(Reduce.class);

        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

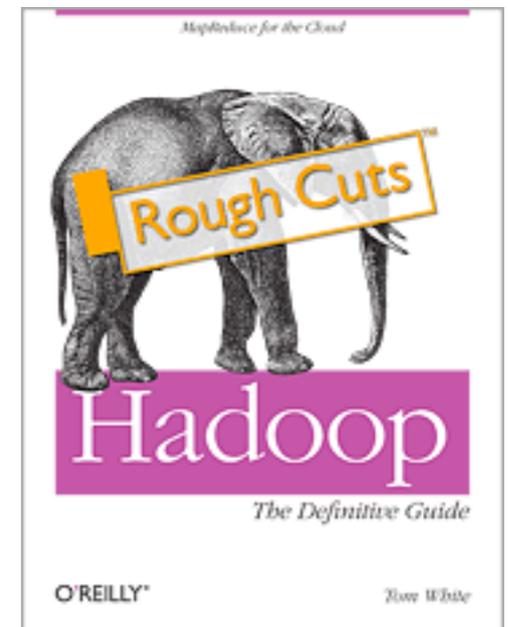
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));

        JobClient.runJob(conf);
    }
}

```

Blam! We're Done! We Have Charts.

(Read more in Tom White's
Book "Hadoop: The
Definitive Guide",
O'Reilly Sept 2009)



<http://oreilly.com/catalog/9780596521998/>

(now back to the
“internal use” part)

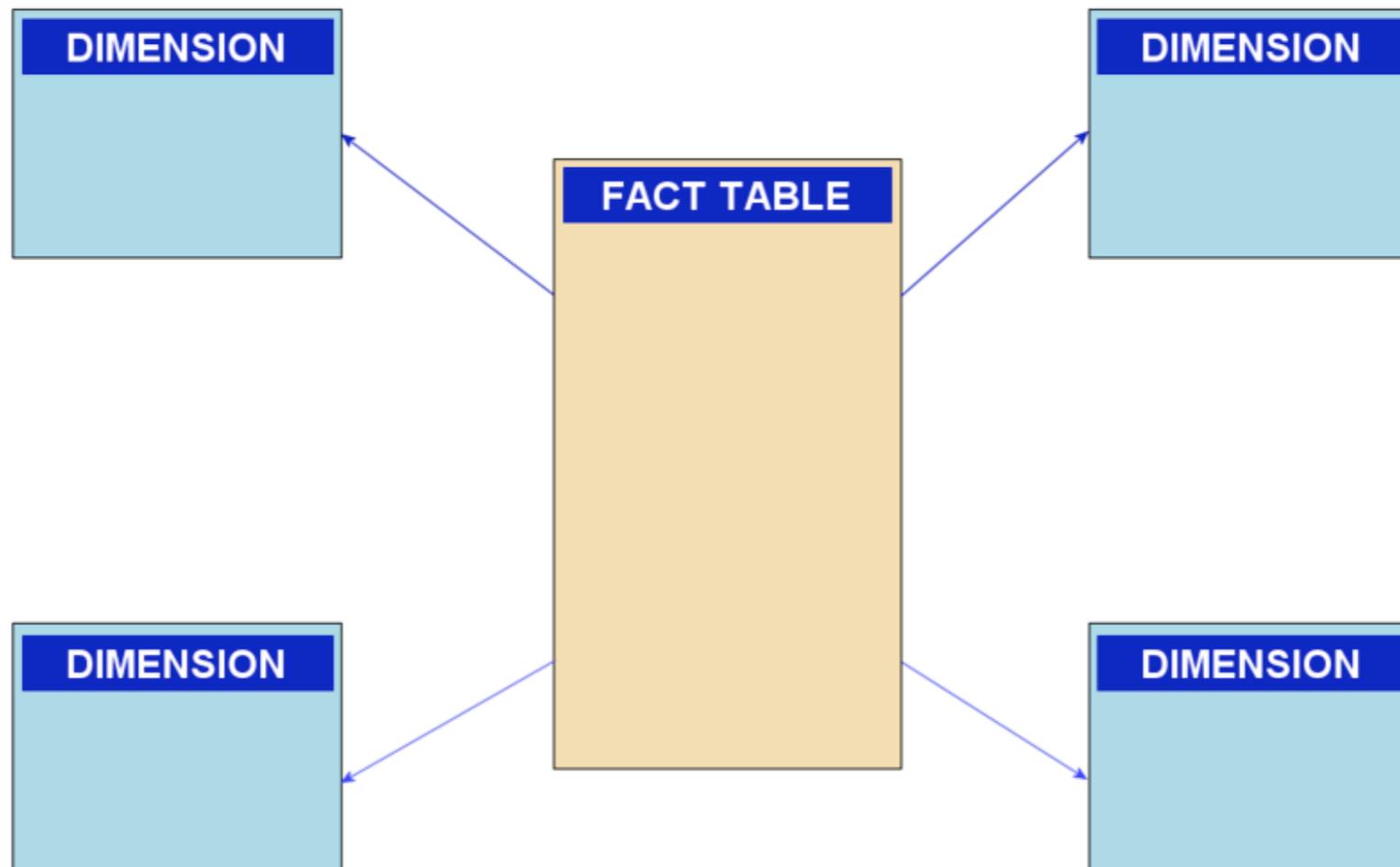
“Little Money” for Analytics

- Can't/don't want to acquire expensive analytics infrastructure
- We do have a couple of spare servers and a team of software developers
- But: not a lot of time for big experiments

Open Source Data Warehousing?

- Back then: Mondrian, Pentaho, ...
- Now: ???
- But: “proper” BI is a lot of work
 - Data cube schema, ETL process, report builders, ...
 - Typically built by dedicated teams

Star Schemas



http://en.wikipedia.org/wiki/Star_schema

So We Want Something Simple

- Based on components we know
- That we can develop/maintain on the side
- That can process a large amount of data efficiently

So we just used
Hadoop

Last.fm Hadoop Applications

- Chart Generation (counting)

Top Artists

⚙ Settings | Paste Your Taste |

[Last 7 days](#) [Last 3 months](#) [Last 6 months](#) [Last 12 months](#) [Overall](#)

1	▶ Alva Noto	162
2	▶ Atom™	106
3	Illa J	68
4	▶ Ras G	61
4	▶ Lukid	61
6	▶ Bullion	46
7	▶ dekstop	42
8	▶ 10-20	41
9	Hudson Mohawke	37
10	▶ Herrmutt Lobby	35

Last.fm Hadoop Applications

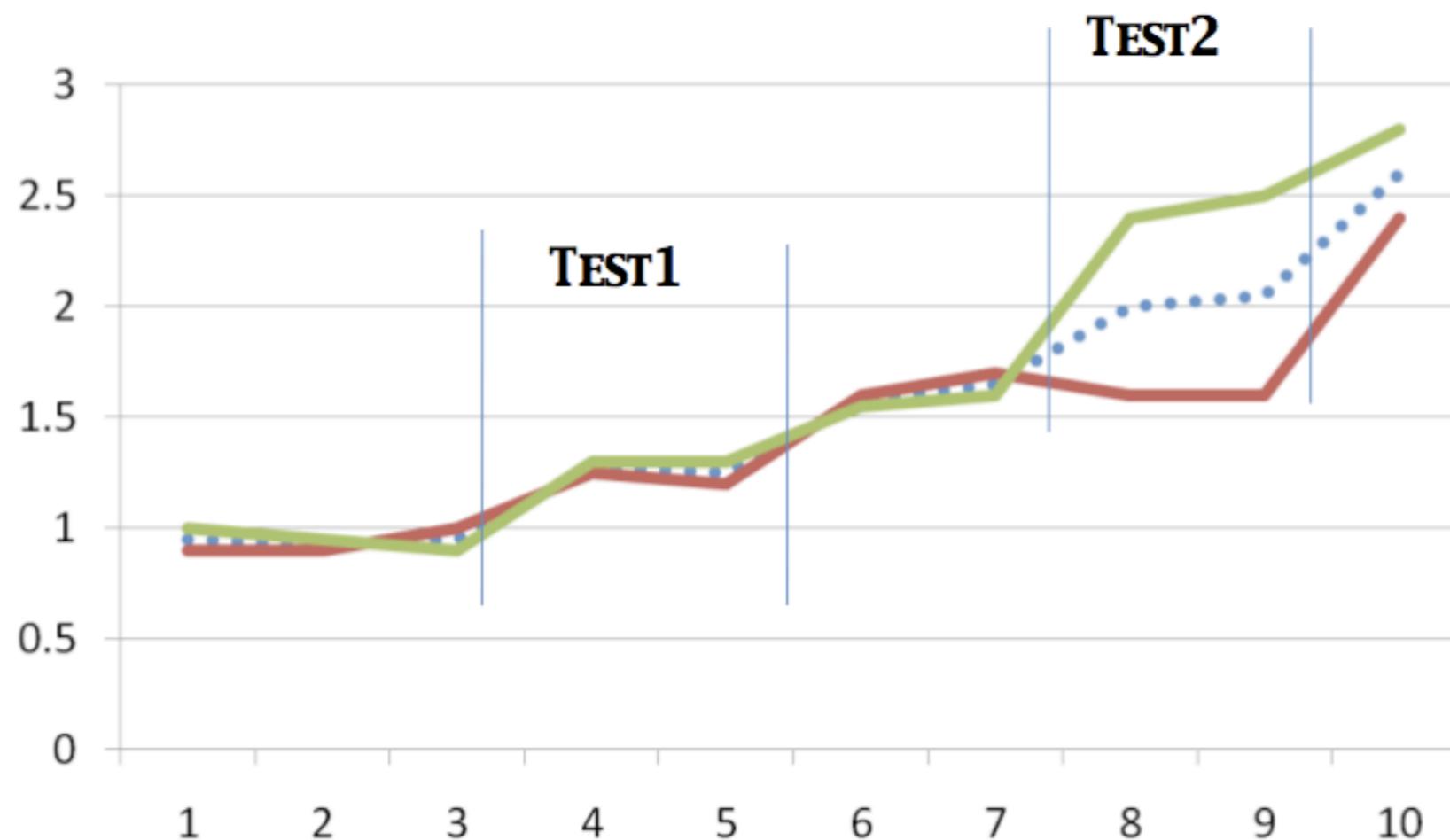
- Internal Metrics (counting)



0.77 %	<div style="width: 0.77%;"></div>
0.00 %	<div style="width: 0.00%;"></div>
0.68 %	<div style="width: 0.68%;"></div>
13.24 %	<div style="width: 13.24%;"></div>

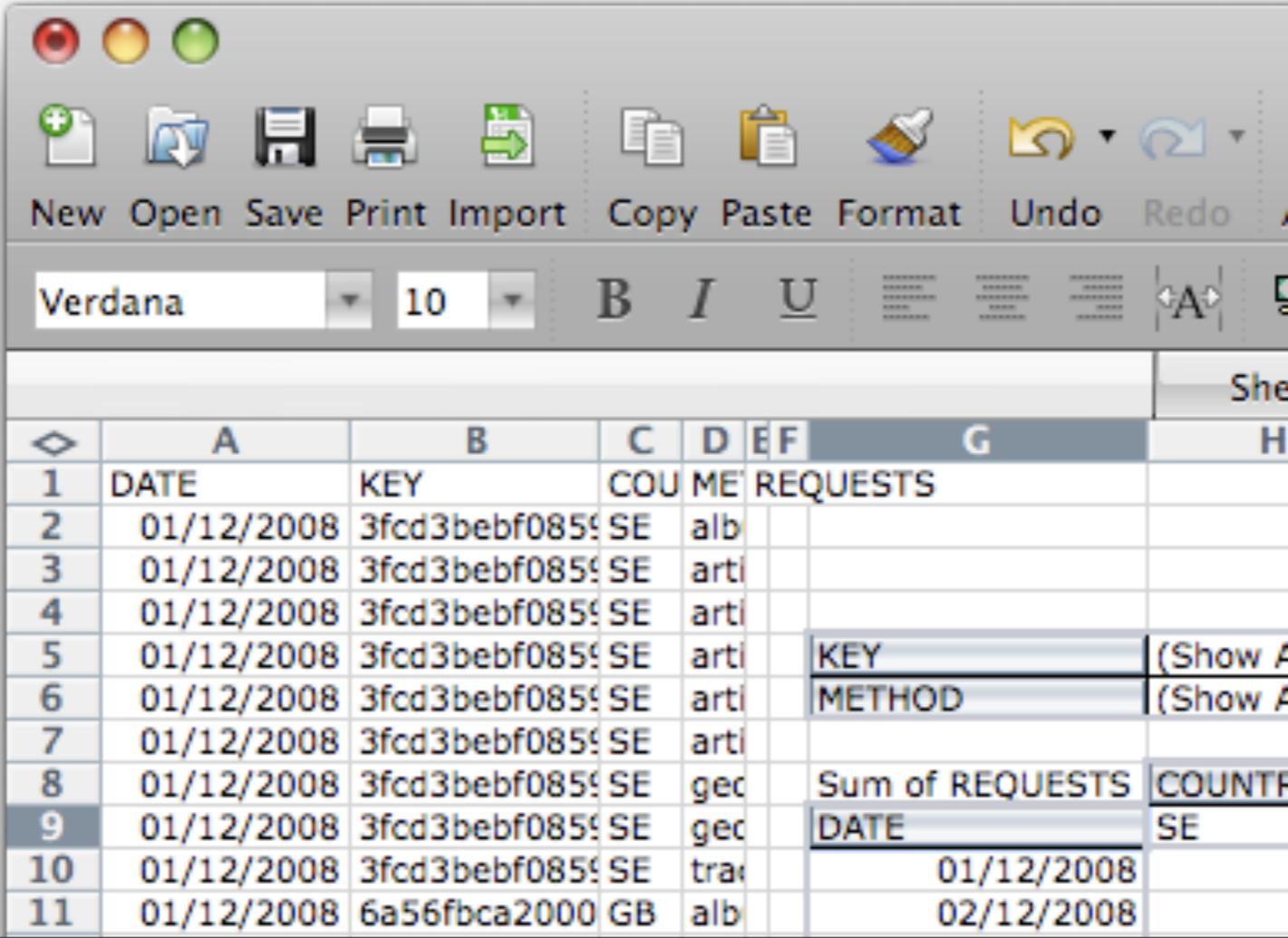
Last.fm Hadoop Applications

- A/B Test Evaluation (counting, then basic statistics)



Last.fm Hadoop Applications

- Misc Ad Hoc Tasks



The screenshot shows a spreadsheet application window with a menu bar and a toolbar. The menu bar includes options like New, Open, Save, Print, Import, Copy, Paste, Format, Undo, and Redo. The toolbar shows icons for these actions. Below the toolbar, the spreadsheet is visible with columns labeled A through H. The data is organized into rows, with columns A through G containing specific data points. Column G is highlighted, and a summary row is visible at the bottom of the data range.

	A	B	C	D	E	F	G	H
1	DATE	KEY	COU	ME	REQUESTS			
2	01/12/2008	3fcd3bebf0859	SE	alb				
3	01/12/2008	3fcd3bebf0859	SE	arti				
4	01/12/2008	3fcd3bebf0859	SE	arti				
5	01/12/2008	3fcd3bebf0859	SE	arti			KEY	(Show A
6	01/12/2008	3fcd3bebf0859	SE	arti			METHOD	(Show A
7	01/12/2008	3fcd3bebf0859	SE	arti				
8	01/12/2008	3fcd3bebf0859	SE	gec			Sum of REQUESTS	COUNTF
9	01/12/2008	3fcd3bebf0859	SE	gec			DATE	SE
10	01/12/2008	3fcd3bebf0859	SE	tra			01/12/2008	
11	01/12/2008	6a56fbca2000	GB	alb			02/12/2008	

data analysis often with **Small
Specialised Tools**, not
General-Purpose BI Environment

E.g. Dumbo

```
def mapper(key, value):  
    key.isprimary = "hostnames" in key.body[0]  
    key.body, value = value.split("\t", 1)  
    yield key, value
```

```
class Reducer:  
    def __init__(self):  
        self.hostname = "unknown"  
    def __call__(self, key, values):  
        if key.isprimary:  
            self.hostname = values.next()  
        else:  
            key.body = self.hostname  
            self.hostname = "unknown"  
            for value in values:  
                yield key, value
```

```
def mapper(key, value):
    key.isprimary = "hostnames" in key.body[0]
    key.body, value = value.split("\t", 1)
    yield key, value

class Reducer:
    def __init__(self):
        self.hostname = "unknown"
    def __call__(self, key, values):
        if key.isprimary:
            self.hostname = values.next()
        else:
            key.body = self.hostname
            self.hostname = "unknown"
            for value in values:
                yield key, value

def runner(job):
    job.additer(mapper, Reducer)

def starter(prog):
    prog.addopt("addpath", "yes")
    prog.addopt("joinkeys", "yes")

if __name__ == "__main__":
    from dumbo import main
    main(runner, starter)
```

Hadoop Output...

- Load in spreadsheet, Matlab
- Store in RDBMS
 - Often in star schemas
 - Only store + compute what's necessary
 - Mostly aggregated data

(Still almost overflowed a signed int PK in a reporting table with 2B+ rows)

Last.fm Open Source Infrastructure

- Dumbo (Klaas Bostels)
- BashReduce (Erik Frey), just for fun
- “Zohmg” (?!), in progress

Other Tools

- **Misc custom (non-Hadoop) data stores** some with their own query languages
- **Python. Great libraries: matplotlib, numpy, ...**
- **Matlab/Excel/R/...**
- **PHP, MySQL, Postgres** (curious about Greenplum)
- **Linux shell, TextMate** (yes, those are data processing tools)
- **Interns!**

In Other Companies...

Facebook

- Proper (custom) analytics platform with Hadoop, Hive, Oracle DB, BI tools, ...
- UI using Facebook site components
- See also: Facebook Data Team video

http://developer.yahoo.net/blogs/theater/archives/2008/01/nextyahoonet_big_data_viewpoints_from_the_fac.html

Etc

- **Document search to build data warehouses**
blog.foofactory.fi/2008/08/interactive-query-reporting-with-lucene.html
- **SQLite (Google contributions) + Excel**
blog.gobansaor.com/2009/03/14/sqlite-as-the-mp3-of-data/
- **Solr+Hadoop**
highscalability.com/how-rackspace-now-uses-mapreduce-and-hadoop-query-terabytes-data
- **Also check startuptools.pbwiki.com**

Even Big Companies Are Replacing Their Tools Now

Further Reading

- “MapReduce”, “BigTable” papers
- “Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals”
- “The Data Warehouse Toolkit”
- “Hadoop: The Definitive Guide”
- “The Visual Display of Quantitative Information”

Blogs & Novelty Papers

- Cloudera blog
- Dumbotics blog
- “Big Data:Viewpoints from the Facebook Data Team” video
- “Exploiting database join techniques for analytics with Hadoop” video
- “Map-Reduce-Merge” paper
- “MAD Skills” paper
- “The Manga Guide to Statistics”

Summary

- This is not BI in the classic sense
- Common theme: use what you have (and know)
- Build things that others in your company will want to pick up

“ Build things that others will want to pick up”

- **Dumbo: simplified data processing in Hadoop.** *Used by data team, MIR team, web team*
- **For non-developers: web UIs, forms, query languages, ...**
- **Access to all information without middleman** *(we're still far away from that)*

- *What are you using?*