

Prototyping Data Intensive Apps: TrendingTopics.org



Pete Skomoroch
Research Scientist at LinkedIn
Consultant at Data Wrangling

@peteskomoroch

Talk Outline

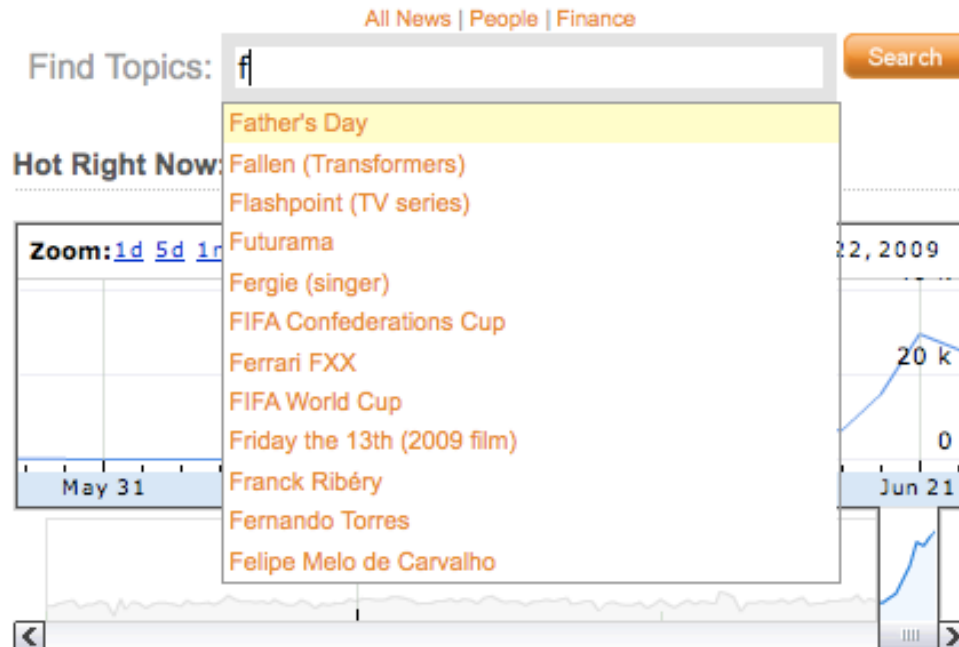
- Trending Topics Overview
- Wikipedia Page View Dataset
- Hadoop on Amazon EC2
- Loading Data on EC2: Amazon EBS & S3
- Daily Timelines with Hadoop Streaming
- Hive Data Warehouse Layer
- Trend Computation with Hive
- Hooking It All Together
- Front End & Visualizations

Data Intensive Web Apps

- Batch data mining or prediction with Hadoop
- Iterate quickly with high level languages & tools
 - Pig, Hive, Clojure, Cascading, Python, Ruby
- EC2: Get running with limited initial capital
- Use external data and APIs in novel ways
- Recent real world example: FlightCaster



TrendingTopics.org



Trending Topics (Last 30 Days)

RANK	VIEWS	TREND	ARTICLE
1	3,036,195		Iran
2	1,486,177		Father's Day
3	4,036,186		Megan Fox
4	447,978		Iranian presidential election, 2009
5	2,910,404		Cristiano Ronaldo

Trending Topics

Hot Wikipedia Topics, Served Fresh Daily.
Built by [Data Wrangling](#) with [Cloudera Hadoop](#)

In the News:

Basij

[Feared Basij militia has deep history](#)

CNN - 22 hours ago

Members of the **Basij** militia enter Tehran University, where students were protesting on Monday. They are the **Basij**, Iran's volunteer paramilitary group that ...

[Amnesty urges Iran to stop using Iranian Women Shot & Killed by](#)
[ABC Online](#) - [New York Times](#) - [Telegraph.co.uk](#) - [Jerusalem Post](#)
[Related Articles »](#)



ABC News

[« Previous](#) [Next »](#)

powered by [Google News](#)

Rising (Last 24 hours):

1. The Stig

Pageviews: 1,019,118



2. Greenland

Pageviews: 1,556,561

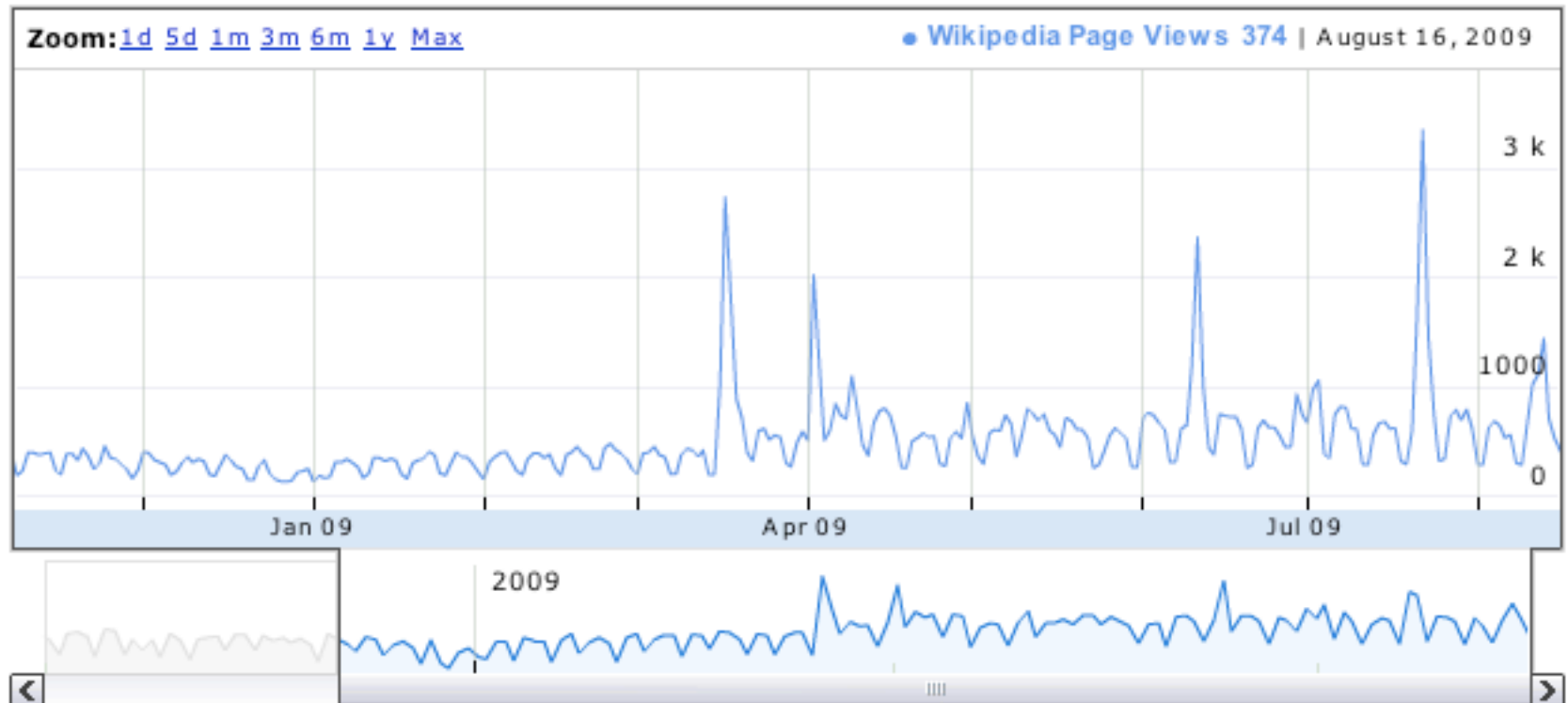


3. Ingushetia



Daily Pageview Timeline Charts

Daily Pageviews for Hadoop



Detects Rising Trends with Hadoop

Trending Topics (Last 30 Days)

RANK	VIEWS	TREND	ARTICLE
1	3,036,195		Iran
2	1,486,177		Father's Day
3	4,036,186		Megan Fox
4	447,978		Iranian presidential election, 2009
5	2,910,404		Cristiano Ronaldo
6	631,136		Ali Khamenei
7	446,097		Mir-Hossein Mousavi
8	4,618,938		Transformers: Revenge of the Fallen
9	572,757		2009 FIFA Confederations Cup
10	968,874		Mahmoud Ahmadinejad
11	1,019,118		The Stig
12	2,899,413		Africa
13	2,576,762		True Blood
14	830,431		Prototype (video game)
15	332,986		Paul Erdős
16	1,015,779		John Dillinger

Rising (Last 24 hours):

1. The Stig

Pageviews: 1,019,118



2. Greenland

Pageviews: 1,556,561



3. Ingushetia

Pageviews: 149,915



4. 2012

Pageviews: 5,085,401



5. Transformers: Revenge of the Fallen

Pageviews: 4,618,938



6. David Duval

Pageviews: 129,127



TrendingTopics is Open Source

- Built as a side project at Data Wrangling
- Core code completed over 2 weeks in June
- Code on Github
- Data released on Amazon Public Datasets

The screenshot shows the GitHub repository page for `datawrangling/trendingtopics`. The repository is described as a "Rails app for tracking trends in server logs - powered by the Cloudera Hadoop Distribution on EC2". It includes links for the homepage, public clone URL, and your clone URL. A commit history table is visible at the bottom.

name	age	message	history
<code>.gitignore</code>	June 06, 2009	update to gitignore [datawrangling]	
<code>Capfile</code>	June 01, 2009	first commit [datawrangling]	
<code>README.textile</code>	June 23, 2009	updated cloudera config in README [datawrangling]	
<code>Rakefile</code>	June 01, 2009	first commit [datawrangling]	

The screenshot shows the Amazon Web Services Public Data Sets page for "Wikipedia Page Traffic Statistics". It includes a table with metadata about the dataset, such as the submitted by, US Snapshot ID, size, creation date, last updated date, license, and source.

Submitted By:	Peter N. Skomoroch
US Snapshot ID (Linux/Unix):	snap-753dfc1c
Size:	320 GB
Creation Date:	06/06/2009
Last Updated:	06/06/2009
License:	GNU Free Documentation License 1.3
Source:	Data Wrangling

Wikipedia Traffic Statistics Dataset

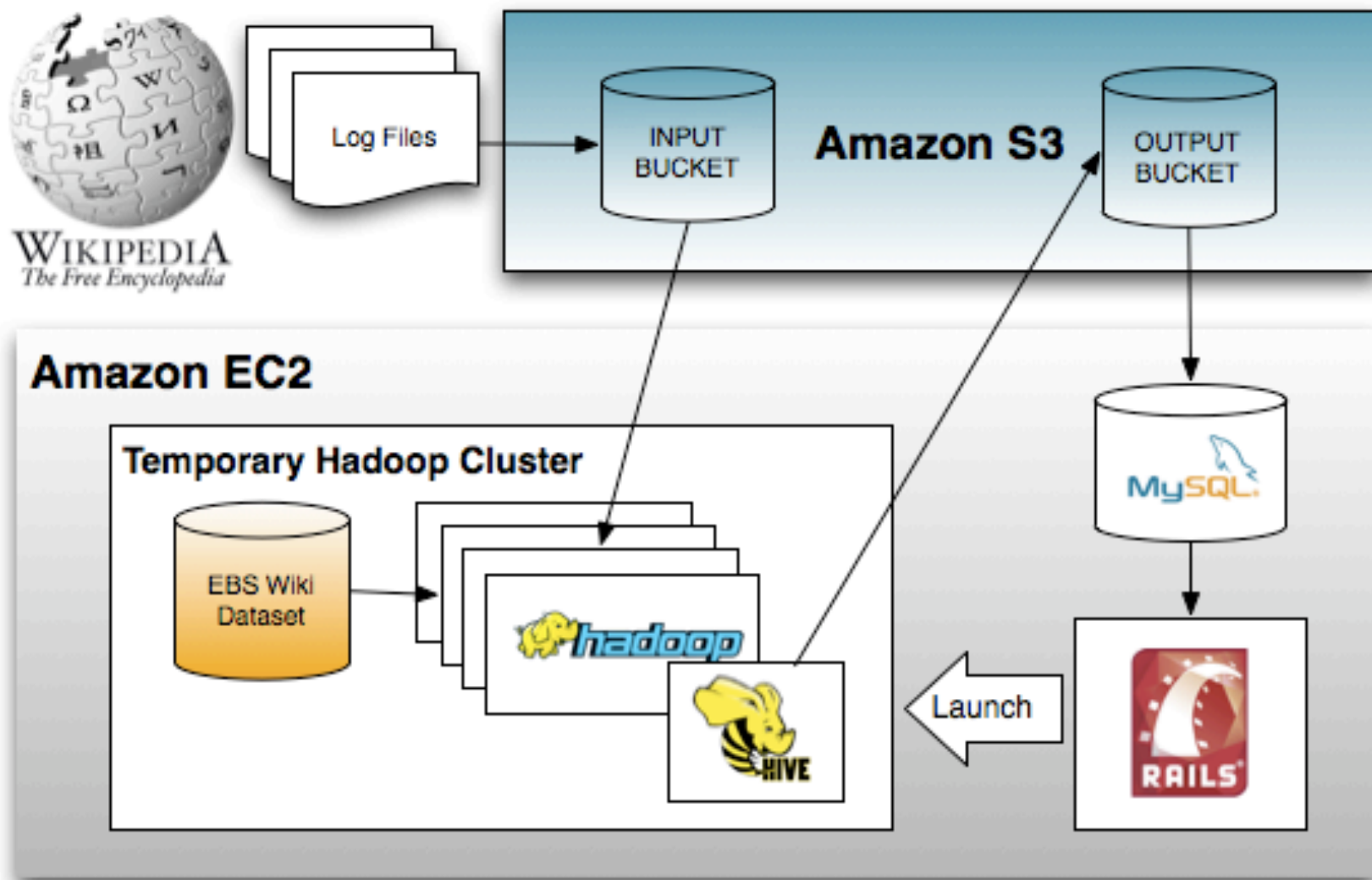
This dataset contains a 320 GB sample of the data used to power `trendingtopics.org`. It includes 7 months of hourly page traffic statistics for over 2.5 Million wikipedia articles (~ 1 TB uncompressed) along with the associated wikipedia content, linkgraph, & metadata.

Compiled by Peter Skomoroch at Data Wrangling, LLC on May, 31, 2009

Technology Stack



Application Data Flow



Hadoop on Amazon EC2

- EC2: Launch servers on demand
- S3: Simple Storage Service
- EBS: Persistent disks for EC2
- Used Cloudera Hadoop Distribution
 - Includes Pig, Hive, HBase, Sqoop, EBS integration
 - Allows customization of Hadoop EC2 instances
 - See EC2 Docs and Tutorials on the Cloudera site



```
% hadoop-ec2 launch-cluster --user-packages 'R git-core' my-hadoop-cluster 10
```

Wikipedia Page View Log Data

- 1 TB of hourly log files from Wikipedia Squid proxy
- Filename contains timestamp
- Columns: project, pagename, pageviews, & bytes

```
$ grep '^en Barack' pagecounts-20090521-100001
```

```
en Barack 8 1240112
```

```
en Barack%20Obama 1 1167
```

```
en Barack_H._Obama 1 142802
```

```
en Barack_H_Obama 3 428946
```

```
en Barack_H_Obama_Jr. 2 285780
```

```
en Barack_Hussein_Obama,_Junior 2 285606
```

```
en Barack_O%27Bama 1 142796
```

```
/mnt/wikidata/wikistats/pagecounts# ls -lh |head
```

```
total 260G
```

```
-rw-r--r-- 1 root root 49M 2009-02-26 13:34 pagecounts-20081001-000000.gz
```

```
-rw-r--r-- 1 root root 46M 2009-02-26 13:34 pagecounts-20081001-010000.gz
```

```
-rw-r--r-- 1 root root 47M 2009-02-26 13:34 pagecounts-20081001-020000.gz
```

```
-rw-r--r-- 1 root root 44M 2009-02-26 13:34 pagecounts-20081001-030000.gz
```

```
-rw-r--r-- 1 root root 45M 2009-02-26 13:34 pagecounts-20081001-040000.gz
```

Wikipedia Redirect Data

- Many articles in the logs redirect to canonical titles
- Amazon Public Dataset contains a lookup table:

```
mysql> select * from page_lookups where page_id = 534366;
```

id	redirect_title	true_title	page_id	page_latest
219291	Barack_Obama	Barack Obama	534366	276223690
3151538	Barak_Obama	Barack Obama	534366	276223690
3151539	Barack_H._Obama	Barack Obama	534366	276223690
3151540	Barack	Barack Obama	534366	276223690
3151541	Barack_Obama.	Barack Obama	534366	276223690
3151542	Barack_H_Obama	Barack Obama	534366	276223690
3151543	44th_President_of_the_United_States	Barack Obama	534366	276223690
3151544	Barach_Obama	Barack Obama	534366	276223690
3151545	Senator_Barack_Obama	Barack Obama	534366	276223690

Loading Data into Hadoop on EC2

- Hadoop can read directly from Amazon S3

```
-input s3n://trendingtopics-sample/wikistats/pagecounts-200*
```

- Pulling in data from EBS snapshots

Attach the volume to our master node instance:

```
localmachine $ ec2-attach-volume vol-7b49a612 -i i-21301448 -d /dev/sdf
```

Log in to the master node and mount the volume:

```
localmachine $ bin/hadoop-ec2 login my-hadoop-cluster
```

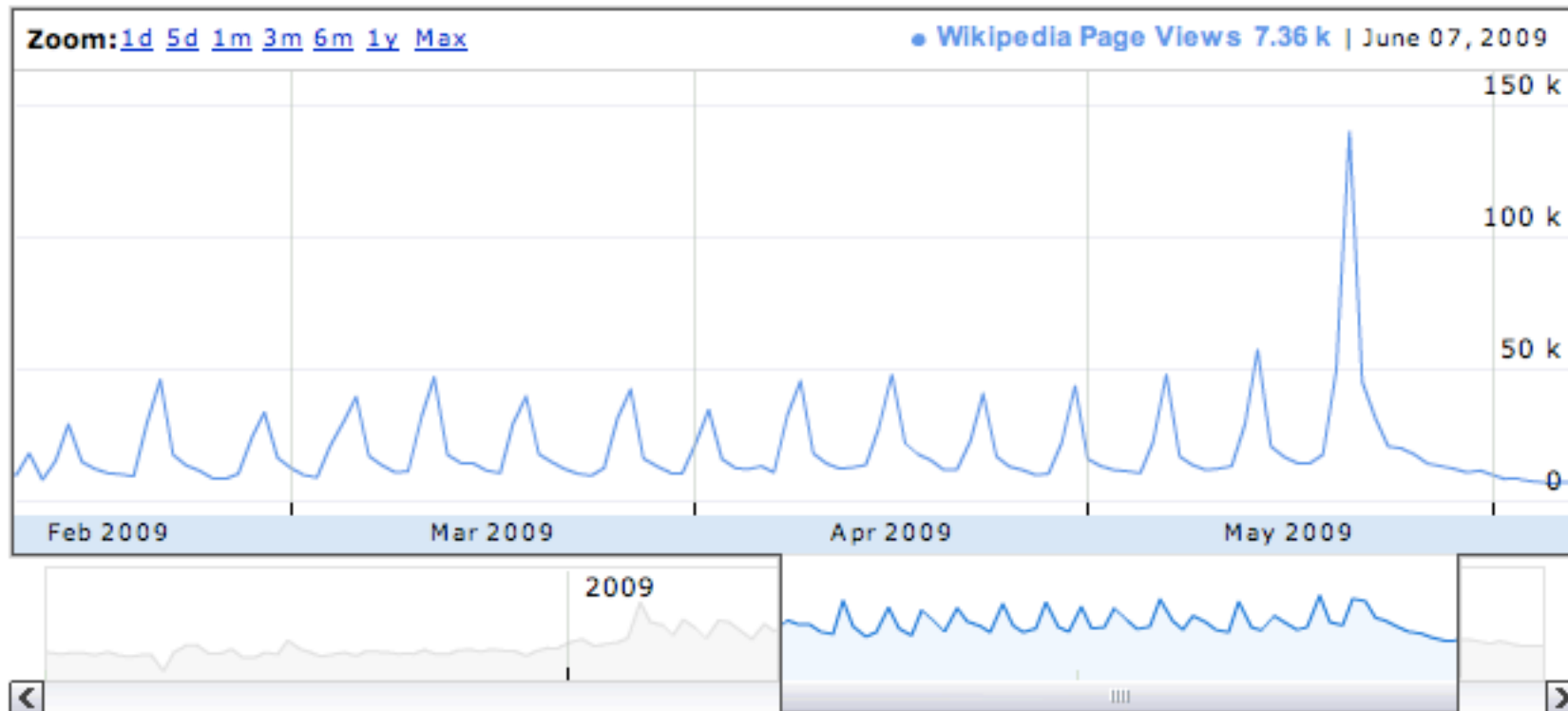
```
root@domU-12-xx-xx-xx-75-81:/mnt# mkdir /mnt/wikidata
```

```
root@domU-12-xx-xx-xx-75-81:/mnt# mount /dev/sdf /mnt/wikidata
```

```
# hadoop fs -put
```

Python Streaming for Daily Timelines

Daily Pageviews for American Idol



Streaming: Filter & Aggregate Logs

```
en Barack 8 1240112
en Barack&20Obama 1 1167
en Barack_H._Obama 1 142802
en Barack_H_Obama 3 428946
en Barack_H_Obama_Jr. 2 285780
```



```
Barack_Obama 20090422 129
Barack_Obama 20090419 143
Barack_Obama 20090421 163
Barack_Obama 20090420 152
```



```
'Barack_Obama\t[20090419,20090420,20090421,20090422]\t[143,152,163,129]\t587'
```

```
# hadoop jar /usr/lib/hadoop/contrib/streaming/hadoop-*-streaming.jar \
-input s3n://trendingtopics-sample/wikistats/pagecounts-200* \
-output stagel-output \
-mapper "daily_timelines.py mapper1" \
-reducer "daily_timelines.py reducer1" \
-file '/mnt/trendingtopics/lib/python_streaming/daily_timelines.py' \
-jobconf mapred.reduce.tasks=10 \
-jobconf mapred.job.name=daily_timelines_stagel
```

Hive Data Warehouse Layer on EC2

- Easy for analysts familiar with SQL
- Familiar syntax (SELECT, GROUP BY, SORT...)
- Hide the MR muck, for JOIN, UNION, etc.
- Supports streaming UDFs
- Sampling: generate datasets for R&D
- Used on Timeline and Redirect data



Fixing Wiki Redirects with Hive JOIN

```
hive> INSERT OVERWRITE TABLE daily_timelines
      SELECT redirect_table.page_id,
             raw_daily_stats_table.dates,
             raw_daily_stats_table.pageviews,
             raw_daily_stats_table.total_pageviews
      FROM redirect_table JOIN raw_daily_stats_table
      ON (redirect_table.redirect_title = raw_daily_stats_table.redirect_title);
```

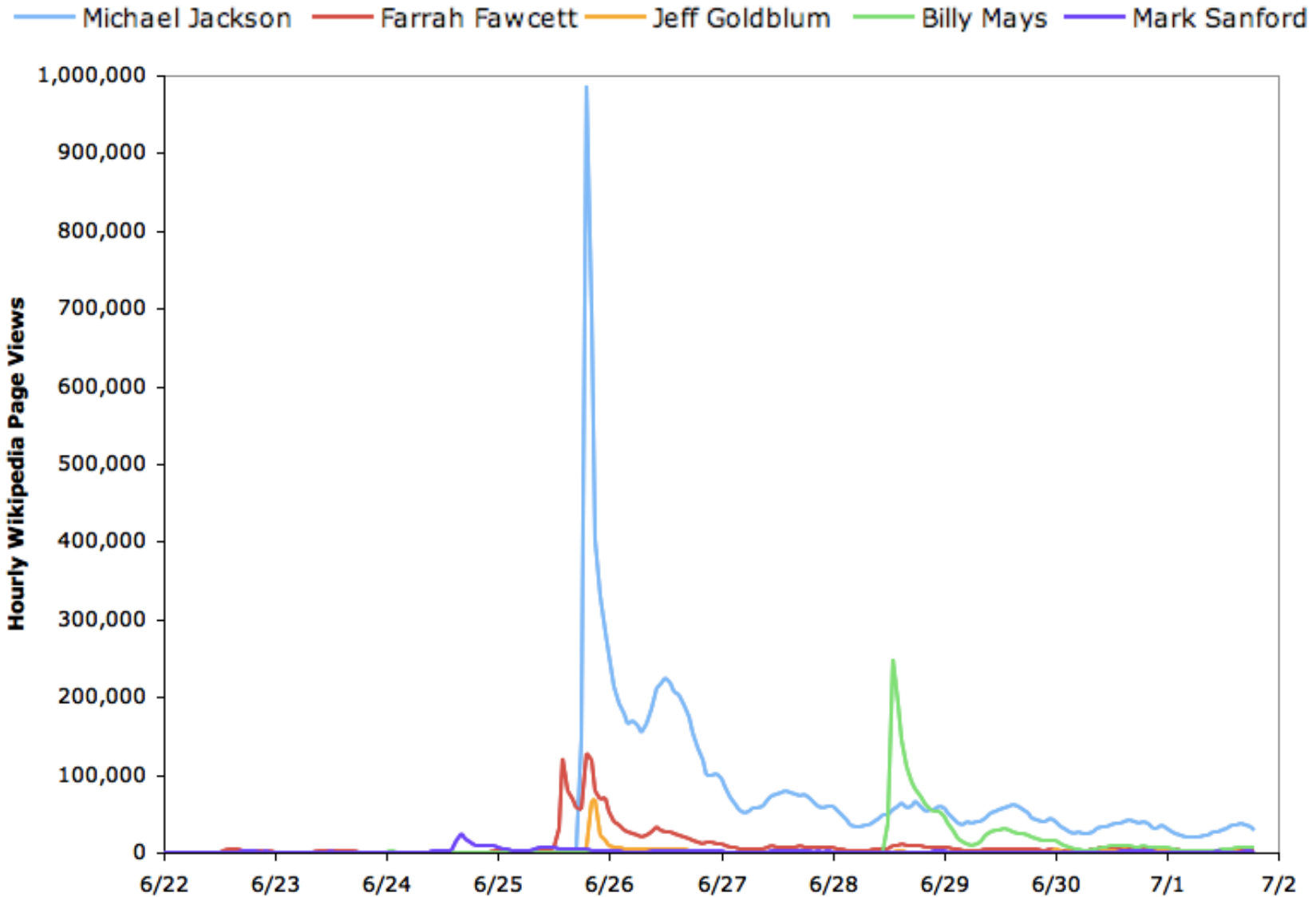
```
hive> select redirect_title, monthly_trend from raw_daily_stats_table sort by
monthly_trend desc LIMIT 10;
```

...

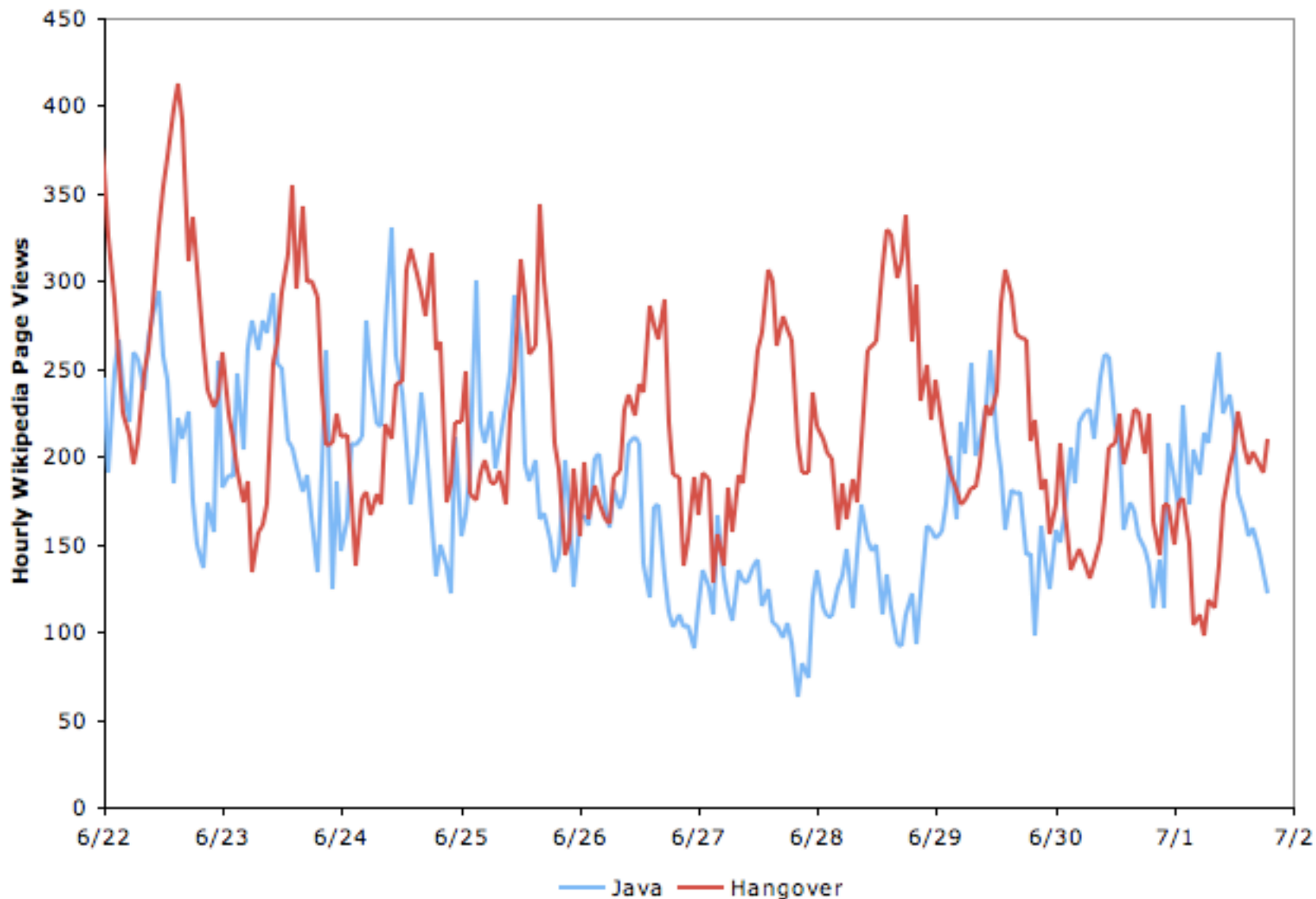
OK

```
Swine_influenza      3635994.0
Swine_flu            2154449.0
Samuel_F._B._Morse   1760945.0
Pandemic             704871.0
Beatrice_Arthur      681941.0
Earth_Day            616544.0
420_(cannabis_culture) 612281.0
Waterboarding        560519.0
Amanda_Holden        394859.0
Britain's_Got_Talent 364008.0
Time taken: 43.772 seconds
```

Trend Detection With Hadoop

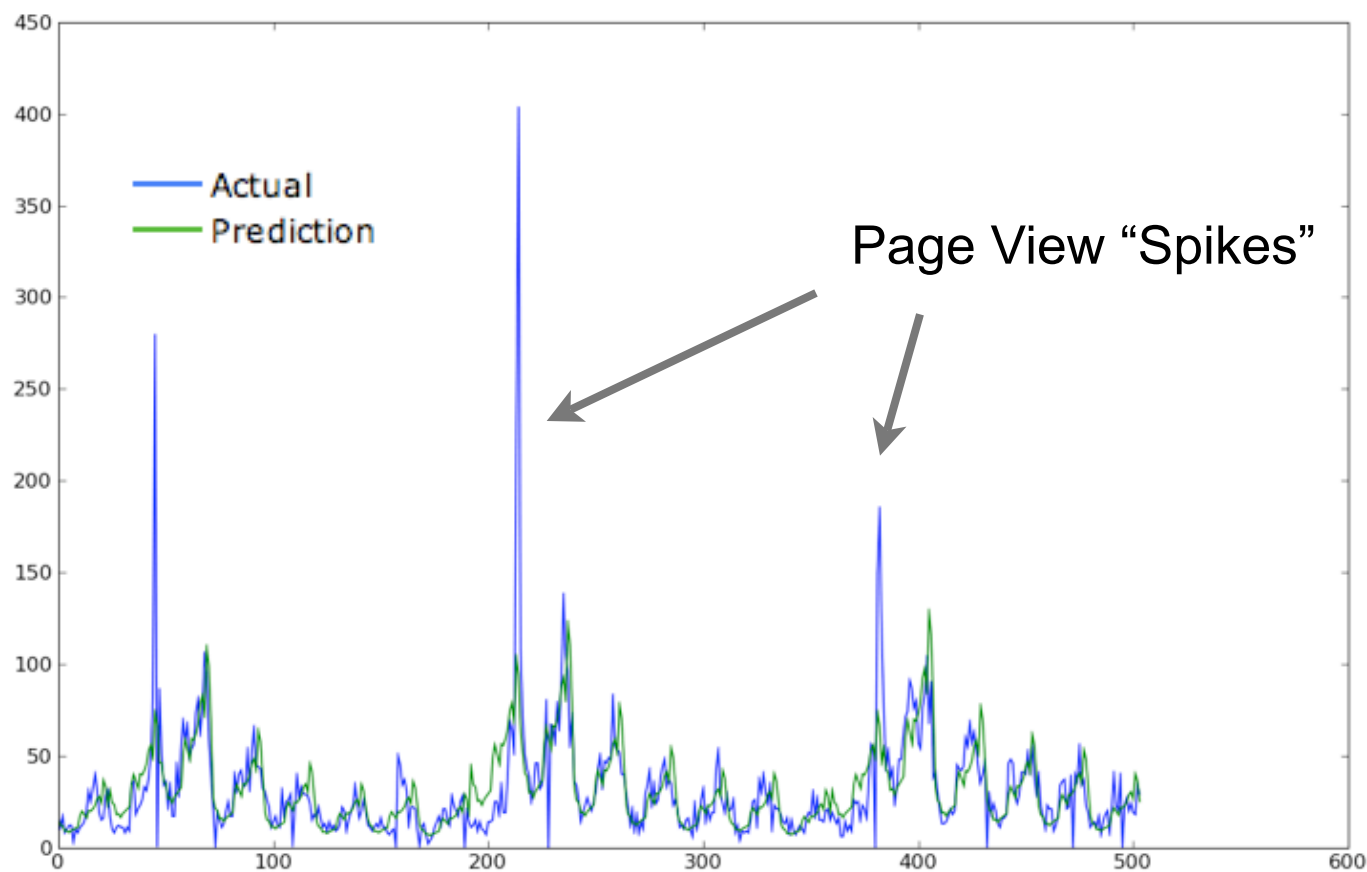


Hourly Data: “Java” vs. “Hangover”



Robust Regression with R & Hadoop

- Fit R model to hourly data for 2.3M Wiki articles
- “Trending” if article views $\gg$ model prediction



Call Python Trend Mapper from Hive

```
# chmod +x /mnt/hive_trend_mapper.py
# hive
hive> add FILE /mnt/hive_trend_mapper.py;

hive> INSERT OVERWRITE TABLE daily_trends
      SELECT u.page_id, u.daily_trend, u.error
      FROM ( FROM daily_timelines ndt
              MAP ndt.page_id, ndt.dates, ndt.pageviews, ndt.total_pageviews
              USING 'python hive_trend_mapper.py'
              AS page_id, daily_trend, error) u;

hive> SELECT * FROM daily_trends LIMIT 10;
OK
16253409      -20.2249832214      0.0800640769025
4015758       -171.306064226      0.0325300024316
151802        6.37331978958      0.0413449115297
5307519       24.9532985002      0.0442373955204
1137274       -148.768754864      0.0289642223182
21895234      -19.2255863732      0.0406222231851
```

Simple Trend Calculation in Python

```
def calc_monthly_trend(dates, pageviews):
    dts, counts = zip( *sorted( zip (dates, pageviews)))
    trend_2 = sum(counts[-15:])
    trend_1 = 1.0*sum(counts[-30:-15])
    monthly_trend = trend_2 - trend_1
    return monthly_trend

def calc_daily_trend(dates, pageviews, total_pageviews):
    '''
    Dead simple trend algorithm used for demo
    Only needs the last 10 days of data
    '''
    # Today's pageviews
    y2 = pageviews[-1]
    # Last Week's pageviews
    y1 = pageviews[-8]
    # Significance factor based on previous week's pageviews
    weekly_pageviews = sum(pageviews[-8:-1])
    # Simple baseline trend algorithm
    slope = y2 - y1
    trend = slope * (1.0 + log(1.0 + int(weekly_pageviews)))
    error = 1.0/sqrt(int(total_pageviews))
    return trend, error
```

Use Trend Scores to Rank Articles

Trending Topics (Last 30 Days)

RANK	VIEWS	TREND	ARTICLE
1	3,036,195		Iran
2	1,486,177		Father's Day
3	4,036,186		Megan Fox
4	447,978		Iranian presidential election, 2009
5	2,910,404		Cristiano Ronaldo
6	631,136		Ali Khamenei
7	446,097		Mir-Hossein Mousavi
8	4,618,938		Transformers: Revenge of the Fallen
9	572,757		2009 FIFA Confederations Cup
10	968,874		Mahmoud Ahmadinejad
11	1,019,118		The Stig
12	2,899,413		Africa
13	2,576,762		True Blood
14	830,431		Prototype (video game)
15	332,986		Paul Erdős
16	1,015,779		John Dillinger

Rising (Last 24 hours):

1. The Stig

Pageviews: 1,019,118



2. Greenland

Pageviews: 1,556,561



3. Ingushetia

Pageviews: 149,915



4. 2012

Pageviews: 5,085,401



5. Transformers: Revenge of the Fallen

Pageviews: 4,618,938



6. David Duval

Pageviews: 129,127



Exporting Data from our EC2 Cluster

- Output files are delimited text for bulk DB load
- Distcp outputs to S3:
 - `hadoop distcp /user/output s3n://trendingtopics/exports`
- Hive can create tables directly over S3 filesystem:
 - `create external table foo ... location 's3n://trendingtopics/hiveoutput'`

Hooking It All Together



Ruby on Rails Front End

- Run a development version of the app locally:

```
git clone git://github.com/datawrangling/trendingtopics.git
```

Create the database:

```
$ rake db:create  
$ rake db:migrate
```

Populate the app with 2 months of demo data from 100 wiki articles:

```
$ rake db:develop
```

Launch the Rails app itself:

```
$ script/server  
=> Booting Mongrel  
=> Rails 2.3.2 application starting on http://0.0.0.0:3000  
=> Call with -d to detach  
=> Ctrl-C to shutdown server
```

Automate & Deploy with EC2onRails

- EC2onRails: Ubuntu server images for EC2, runs instances of Ruby on Rails, MySQL, Apache
- Daily cron job on App server launches Hadoop
- Hadoop EC2 boot script checks out latest trend detection code from Github
- Capistrano tasks deploy new application code, maintain DB, cache, archives

Cron Triggers Hadoop Jobs on EC2

Modify Cloudera hadoop-ec2-init-remote.sh boot script

```
function run_trendingtopics_batch() {
  MYBUCKET=trendingtopics
  MYSERVER=db.trendingtopics.org
  MAILTO=pete@datawrangling.com
  NUMREDUCERS=35
  cd /mnt
  git clone git://github.com/datawrangling/trendingtopics.git
  bash trendingtopics/lib/scripts/run_daily_merge.sh \
  $MYBUCKET $MYSERVER $MAILTO $NUMREDUCERS
}
# self terminate if job takes > 108 minutes
shutdown -h +108 >/dev/null &
if $IS_MASTER ; then
  setup_web
  start_hadoop_master
  run_trendingtopics_batch
else
  start_hadoop_slave
fi
```

Bash Scripts Automate Daily Run

- ***run_daily_merge.sh*** kicks off Hadoop/Hive jobs
- Queries MySQL for last run date
- Generates timelines with Hadoop streaming
- Hive operates on timeline data, computes Trends with Python
- Hive output copied to S3
- Calls remote script to load MySQL staging tables
- Hadoop cluster self terminates
- Remote script swaps staging and prod after load

Capistrano Tasks

- Capistrano: Ruby tool for automating tasks on one or more remote servers (www.capify.org)
- Deploys Rails code and cron jobs to App server
- Cleanup tasks are triggered at end of load:
 - Flags featured Wikipedia articles
 - Indexes MySQL tables, Flushes cache

Visualization APIs

- Google Viz API: Annotated Timeline for Rails

```
@data = {  
  1.day.ago => { :foo=>123, :bar=>100 },  
  2.day.ago => { :foo=>345, :bar=>200 }  
}
```

- Google Charts sparklines: gc4r

```
dataset = GoogleChartDataset.new :data => [10,50,4,10,16]  
data = GoogleChartData.new :datasets => dataset  
@chart = GoogleLineChart.new :width => 300, :height => 200  
@chart.data = data
```

- Yahoo BOSS image search in Rails

```
def picture  
  boss = BOSSMan::Search.images(self.title, { :filter => "-porn"})  
  if boss.totalhits.to_i > 0  
    result = boss.results[0]  
    url = result.thumbnail_url  
  else  
    url = "White_square_with_question_mark.png"  
  end  
end
```

Ideas & Next Steps

- Try out the code on Github
- Show related articles using link graph data
- Extract trending phrases from article text
- Boost Twitter trending topics with Wikipedia logs
- Update top trends hourly
- Use date partitioned tables, incremental updates
- Add a real search engine (uses MySQL now)

LinkedIn Analytics Team



LinkedIn

We're Hiring

Questions?

- Contact

- Blog: datawrangling.com
- Twitter: @peteskomoroch



- Code

- <http://github.com/datawrangling/trendingtopics>
- Hadoop blog posts & tutorials on the Cloudera site
- Amazon EMR tutorial on Finding Similar Artists with Hadoop & Last.FM data <http://bit.ly/EMRsimilarity>