Mining Public Datasets

using Apache Zeppelin (incubating), Apache Spark and Juju

by Alexander Bezzubov
NFLabs for AppacheCon ’16 NA

Alexander Bezzubov

github.com/bzz

@seoul_engineer

Software Engineer at NFLabs, Seoul, South Korea

Co-organizer of SeoulTech Society

Committer and PPMC member of Apache Zeppelin (Incubating)

Graduated **Maths** at St.Petersburg State University, Russia

Mining Public Data

Mining Public Data

story of 2 data products built on real public datasets,
leveraging practical stack of open source tools

Apache Zeppelin, Spark and Juju
Github Archive, Common Crawl

PUBLIC DATASETS: Number, Size & Growth

Web Crawls
Structured data (RDF, micro-formats, tables)
Hackers News\Reddit\Twitter\StackOverflow\Wikipedia
Reviews (movies, restaurants, beer, wine)
Emails (Enroll, ASF public ML archives)
Census Data (US, UK, UN, Japan, etc)
Transportation (Taxi, Flights, Bicycles)
Genome

PUBLIC DATASETS: Number, Size & Growth

Web Crawls
Structured data (RDF, micro-formats, tables)
Hackers News\Reddit\Twitter\StackOverflow\Wikipedia
Reviews (movies, restaurants, beer, wine)
Emails (Enroll, ASF public ML archives)
Census Data (US, UK, UN, Japan, etc)
Transportation (Taxi, Flights, Bicycles)
Genome

order of Tbs

PUBLIC DATASETS: Number, Size & Growth

Web Crawls
Structured data (RDF, micro-formats, tables)
Hackers News\Reddit\Twitter\StackOverflow\Wikipedia
Reviews (movies, restaurants, beer, wine)
Emails (Enroll, ASF public ML archives)
Census Data (US, UK, UN, Japan, etc)
Transportation (Taxi, Flights, Bicycles)
Genome

order of Tbs

AWS Public Datasets https://aws.amazon.com/public-data-sets/
Yahoo Webscope https://webscope.sandbox.yahoo.com/
Stanford Network Analyser Project http://snap.stanford.edu/data/

Physics Research http://opendata.cern.ch

PUBLIC DATASETS: Number, Size & Growth

Web Crawls
Structured data (RDF, micro-formats, tables)
Hackers News\Reddit\Twitter\StackOverflow\Wikipedia
Reviews (movies, restaurants, beer, wine)
Emails (Enroll, ASF public ML archives)
Census Data (US, UK, UN, Japan, etc)
Transportation (Taxi, Flights, Bicycles)
Genome

order of Tbs

AWS Public Datasets https://aws.amazon.com/public-data-sets/
Yahoo Webscope https://webscope.sandbox.yahoo.com/
Stanford Network Analyser Project http://snap.stanford.edu/data/

Physics Research http://opendata.cern.ch

order of Pbs

PUBLIC DATA = OPPORTUNITY

I. Tools

II. Approach

III. Datasets

I. Tools

II. Approach

III. Datasets

TOOL TO PURSUIT THE OPPORTUNITY:

Overview Big Data eco-system

Spark

HDFS

APACHE PHOENIX

mahout

Map Reduce

APACHE LENS

Flink

kafka

accumulo

apache Ignite

TAJO

GEODE

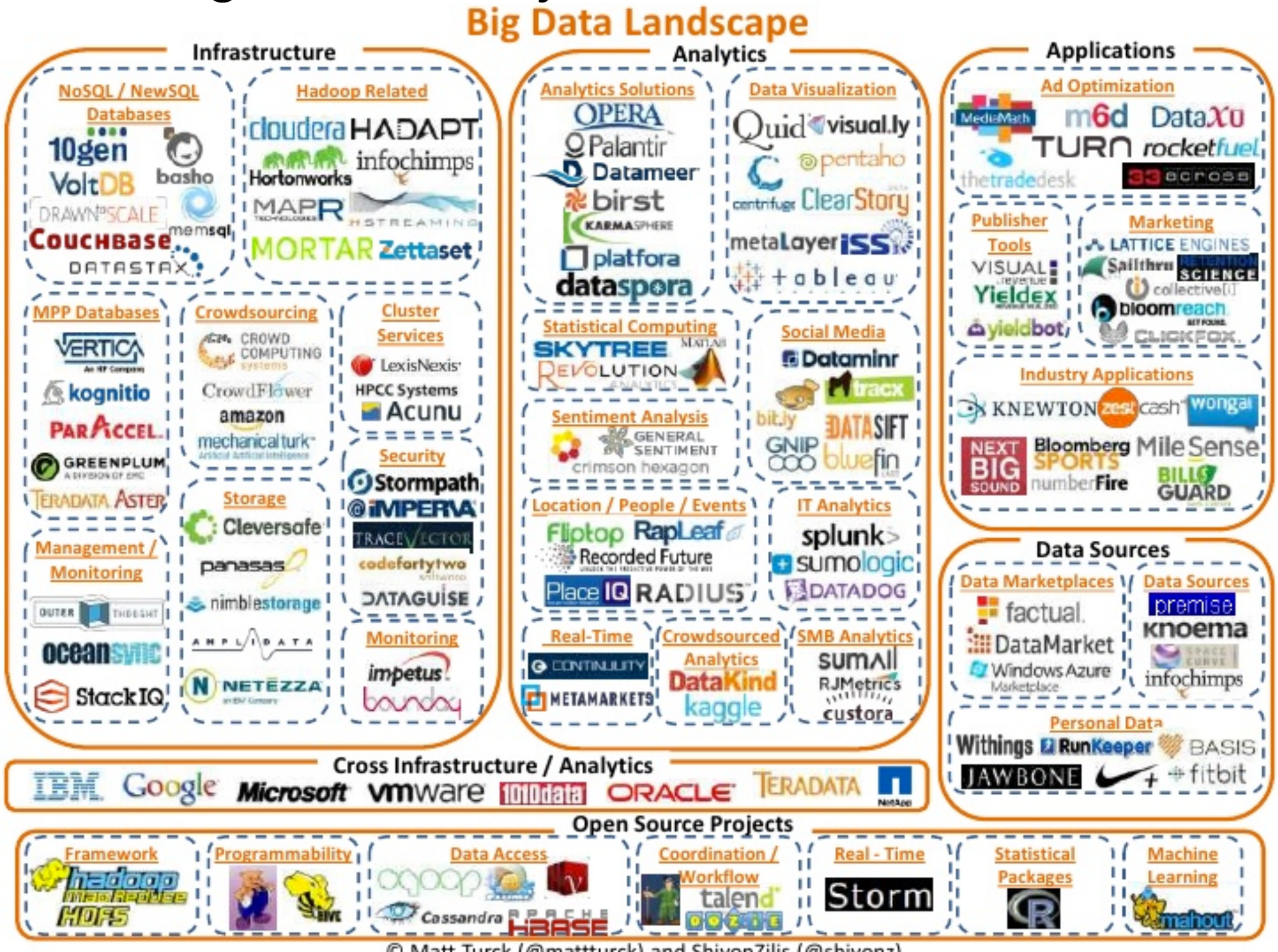
HBASE

Sqoop

APACHE DRILL

TOOL TO PURSUIT THE OPPORTUNITY:

Overview Big Data eco-system



TOOL TO PURSUIT THE OPPORTUNITY:

Todays choice Zeppelin, Spark, Juju

Apache Spark
Scala, Python, R

Apache Zeppelin
Modern Web GUI, plays nicely with Spark, Flink, HAWQ, Elasticsearch, etc.

Warcbase
Spark library for saved crawl data (WARC)

Juju
Scales, integration with Spark, Zeppelin, AWS, GCE

APACHE SPARK

http://spark.apache.org

From Berkeley AMP Labs, since 2010

Joined Apache since 2014

1000+ contributors

REPL + Java, Scala, Python, R APIs

TOOL TO PURSUIT THE OPPORTUNITY:

Todays choice Zeppelin, Spark, Juju

Apache Spark
Scala, Python, R

Apache Zeppelin
Modern Web GUI, plays nicely with Spark, Flink, HAWQ, Elasticsearch, etc.

Warcbase
Spark library for saved crawl data (WARC)

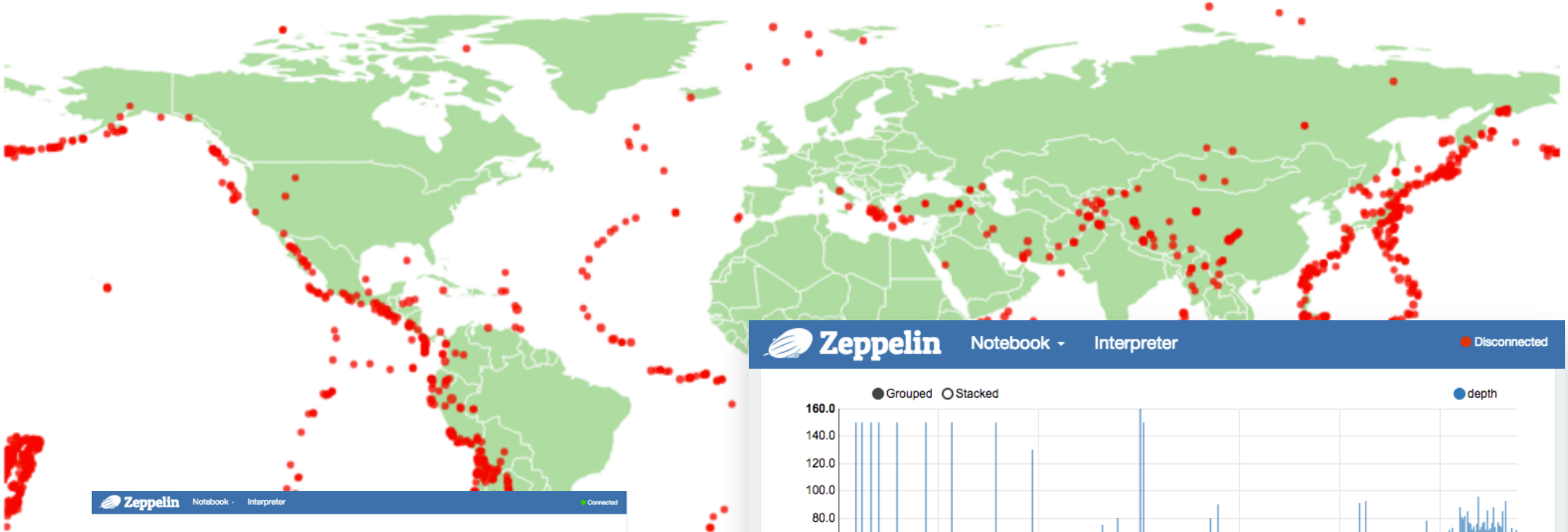
Juju
Scales, integration with Spark, Zeppelin, AWS, GCE

APACHE ZEPPELIN: Brief history

http://zeppelin.incubator.apache.org

12.2012	Commercial App using AMP Lab Shark 0.5
10.2013	Prototype Hive/Shark
08.2013	NFLabs opensource project Hive/Shark/Spark
12.2014	Enters ASF Incubation
01.2016	3 releases
05.2016	Graduation vote passed

Zeppelin Notebook Interpreter Disconnected



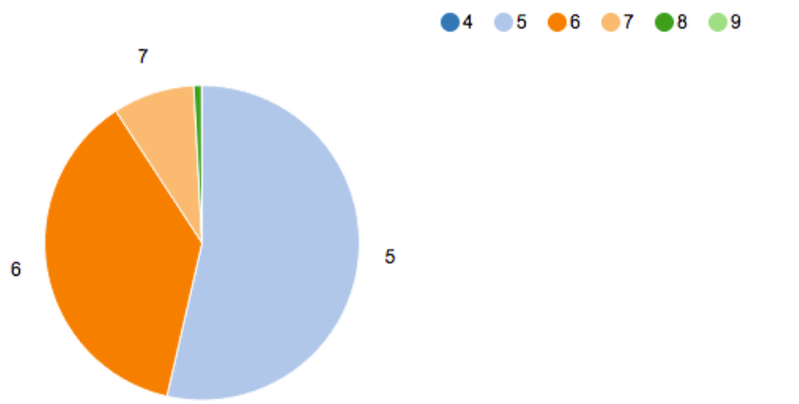
Zeppelin Notebook Interpreter Disconnected

Took 2 seconds.

```
println("%table mag\tcount")
earthQuake.map(s=>(s(5).toString.toDouble.toInt, 1)).reduceByKey(_ + _).collect.foreach(s=>println(s._1
```

FINISHED

settings



Took 1 seconds.

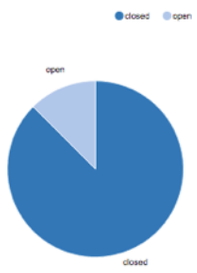
Zeppelin Notebook Interpreter Connected

16 pull requests

8 Contributors

74 comments

4972 additions



number	state	login	commits	comments	additions	deletions	created_at	closed_at	took
20	closed	jongyoul	4	5	34	3	2015-04-01	2015-04-02	null
21	closed	jongyoul	1	1	0	1	2015-04-01	2015-04-02	null
22	closed	RamVenkatesh	4	5	42	7	2015-04-01	2015-04-05	null
23	closed	jongyoul	1	3	24	0	2015-04-01	2015-04-01	null
24	closed	langley	1	1	71	31	2015-04-02	2015-04-05	null
25	closed	jongyoul	2	4	8	12	2015-04-02	2015-04-05	null
26	closed	Leemoonsoo	1	1	39	3	2015-04-03	2015-04-05	null
27	closed	Leemoonsoo	21	24	4,676	203	2015-04-06	2015-04-12	null
30	open	bzz	1	2	1	1	2015-04-07	null	null
29	closed	corneadoug	1	3	4	5	2015-04-07	2015-04-08	null

number	state	login	avatar	created_at	closed_at
20	closed	jongyoul		2015-04-01T01:50:38Z	2015-04-02T13:26:30Z
21	closed	jongyoul		2015-04-01T01:57:09Z	2015-04-02T13:27:53Z
22	closed	RamVenkatesh		2015-04-01T06:09:37Z	2015-04-05T12:55:35Z
23	closed	jongyoul		2015-04-01T07:30:02Z	2015-04-01T23:42:20Z
24	closed	langley		2015-04-02T01:15:03Z	2015-04-05T12:57:48Z
25	closed	jongyoul		2015-04-02T13:59:30Z	2015-04-05T12:59:38Z
26	closed	Leemoonsoo		2015-04-03T05:02:04Z	2015-04-05T13:00:47Z
27	closed	Leemoonsoo		2015-04-06T10:48:45Z	2015-04-12T05:58:50Z
28	closed	bzz		2015-04-07T07:39:56Z	2015-04-08T16:47:28Z
29	closed	corneadoug		2015-04-07T08:04:11Z	2015-04-08T16:41:18Z
30	open	bzz		2015-04-07T09:02:26Z	null
31	closed	RamVenkatesh		2015-04-07T14:04:22Z	2015-04-16T18:02:52Z
32	closed	syepes		2015-04-08T21:31:19Z	2015-04-11T09:48:49Z

TOOL TO PURSUIT THE OPPORTUNITY:

Todays choice Zeppelin, Spark, Juju

Apache Spark
Scala, Python, R

Apache Zeppelin
Modern Web GUI, plays nicely with Apache Spark, Flink, HAWQ, Elasticsearch, etc.

Warcbase
Spark library for saved crawl data (WARC)

Juju
Scales, integration with Spark, Zeppelin, AWS, GCE

WARCBASE

https://github.com/lintool/warcbase

Spark library for WARC (Web ARChive) data processing
 * text analysis
 * site link structure

```
import org.warcbase.spark.matchbox._
import org.warcbase.spark.rdd.RecordRDD._

val r =
RecordLoader.loadArc("/directory/to/arc/file.arc.gz", sc)
.keepValidPages()
.map(r => ExtractTopLevelDomain(r.getUrl))
.countItems()
.take(10)
```

http://lintool.github.io/warcbase-docs

TOOL TO PURSUIT THE OPPORTUNITY:

Todays choice Zeppelin, Spark, Juju

Apache Spark
Scala, Python, R

Apache Zeppelin
Modern Web GUI, plays nicely with Apache Spark, Flink, HAWQ, Elasticsearch, etc.

Warcbase
Spark library for saved crawl data (WARC)

Juju
Scales, integration with Spark, Zeppelin, AWS, GCE

JUJU

https://jujucharms.com/

Service modelling at scale

4+ years old

Deployment\configuration automation
 + Integration with Spark, Zeppelin, Ganglia, etc
 + AWS, GCE, Azure, LXC, etc

JUJU

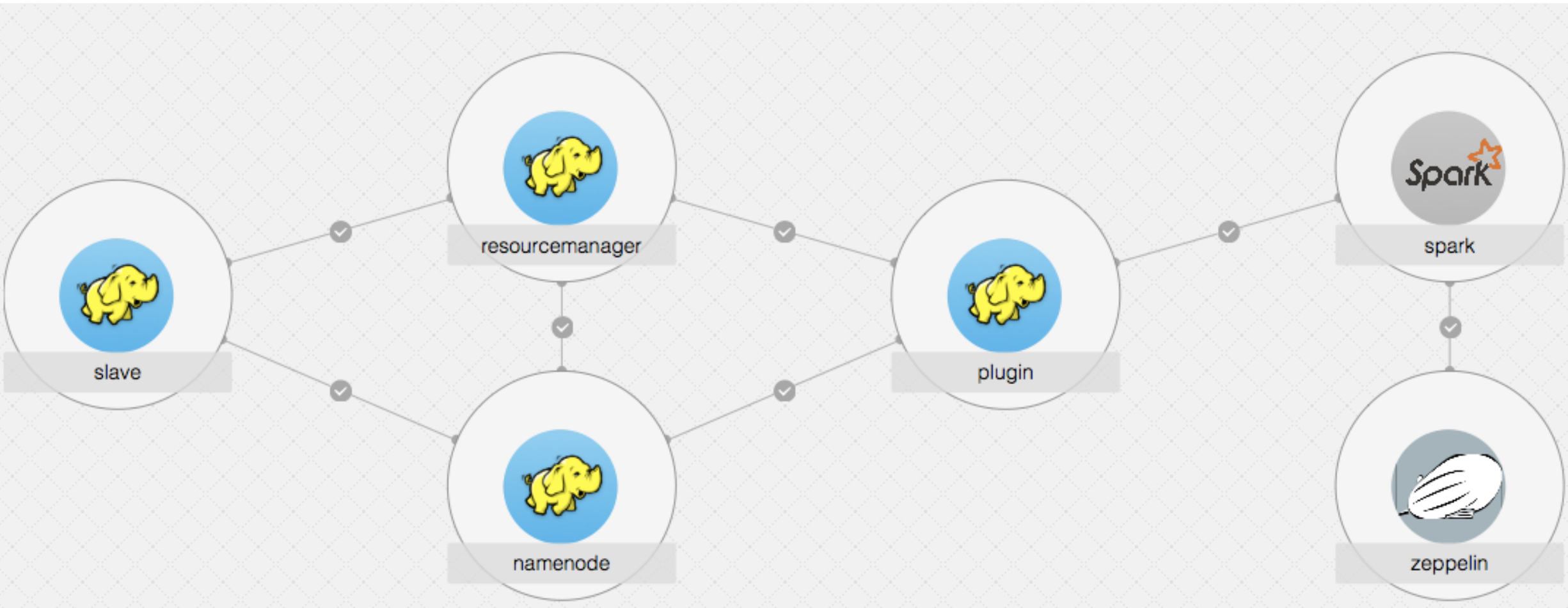
http://bigdata.juju.solutions/getstarted

```
$ apt-get install juju-core juju-quickstart
# or
$ brew install juju juju-quickstart
$ juju generate-config
#LXC, AWS, GCE, Azure, VMWare, OpenStack

$ juju bootstrap
$ juju quickstart apache-hadoop-spark-zeppelin
$ juju expose spark zeppelin
$ juju add-unit -n4 slave
```

JUJU

http://bigdata.juju.solutions/getstarted



7 node cluster designed to scale out

I. Tools

Zeppelin, Spark, Juju

II. Approach

III. Datasets

APPROACH: local, small cluster, big cluster

1 core	Prototype	Your laptop
10s PC	Estimate the cost	AWS\GCE\etc
100s instances	Scale out	Deployment automation

APPROACH: local, small cluster, big cluster

1 core	Prototype	Your laptop
10s PC	Estimate the cost	AWS\GCE\etc
100s instances	Scale out	Deployment automation

described tools do cover all these cases

I. Tools

Zeppelin, Spark, Juju

II. Approach

Prototype, Estimate, Scale out

III. Datasets

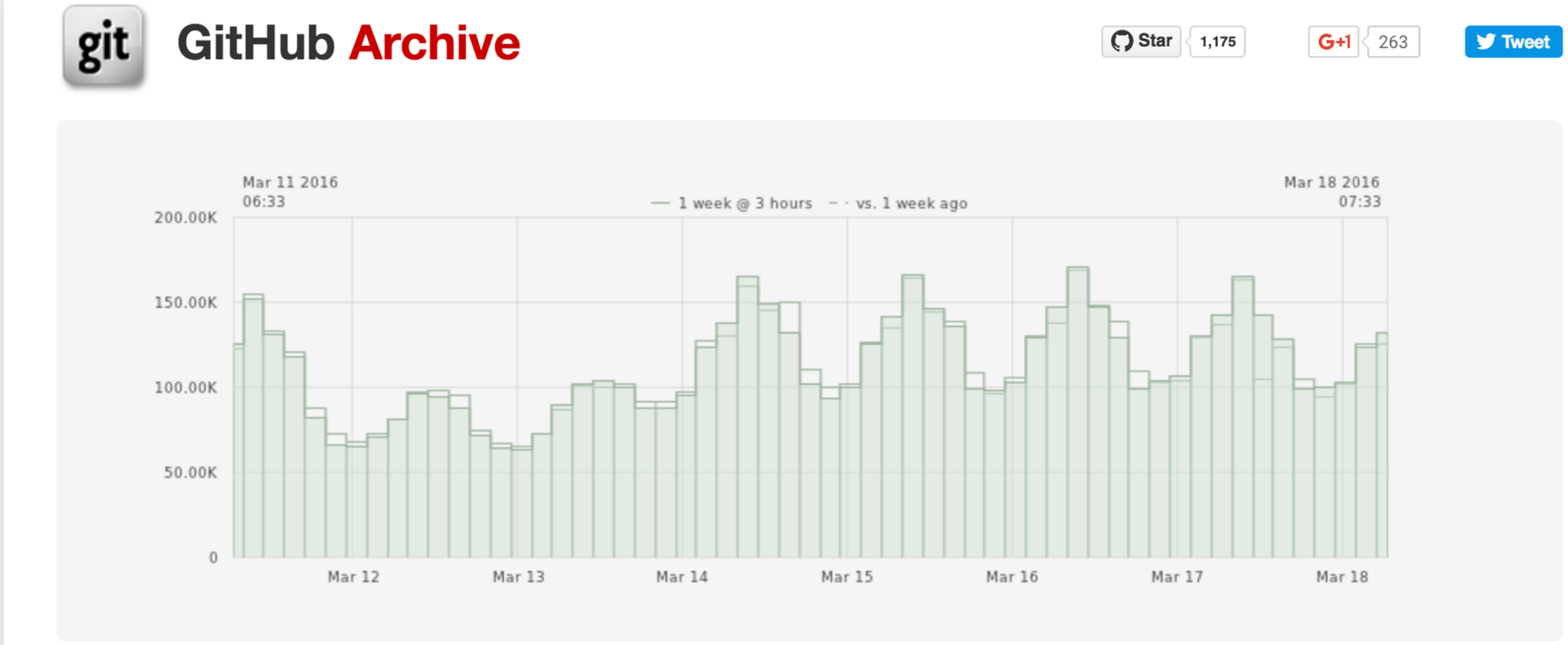
DATA: 2 datasets

Github Archive

Common Crawl

DATA: GitHub

http://githubarchive.org



- +300Gb compressed on google compute
- Collaboration google and github engineers
- Events on PR, repo, issues, comments, etc in JSON

Commit Logs From Last Night

because real hackers pivot two hours before their demo

jjsullivan5196
03/18/16 7:18 AM

`Tried to do something smart but fuck it.`

oatterzongit
03/18/16 6:51 AM

`fix shit`

QuentinTorg
03/18/16 6:22 AM

`fixed the shit`

E-vanderHeide
03/18/16 6:20 AM

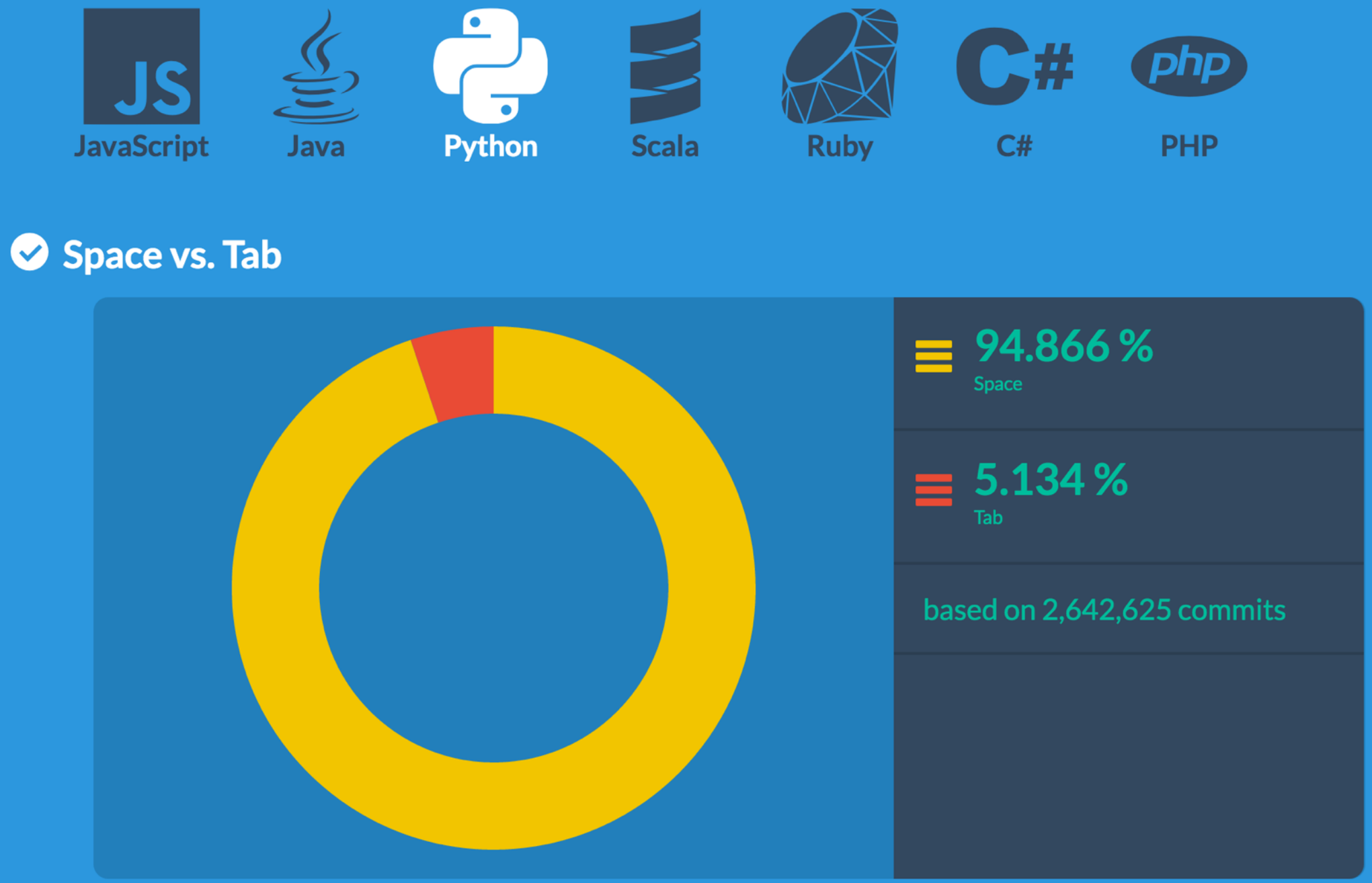
`hacky presentation ready kind of shit`

This thing tweets at @CLFLN

Created by @abestanway

Watch the video!

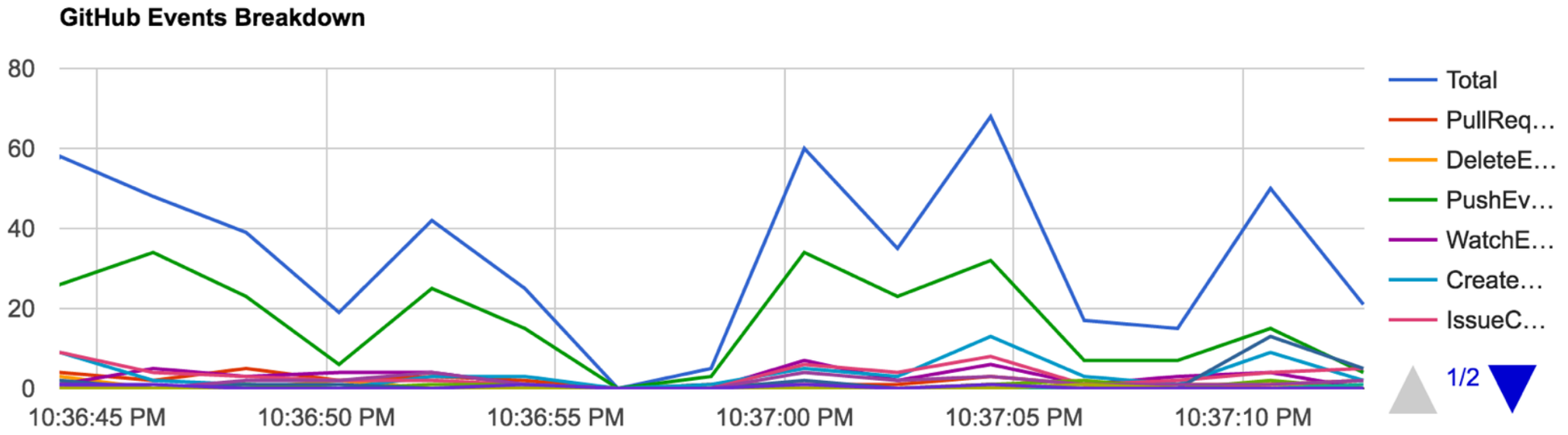
http://www.commitlogsfromlastnight.com/



http://sideeffect.kr/popularconvention/

GitLive

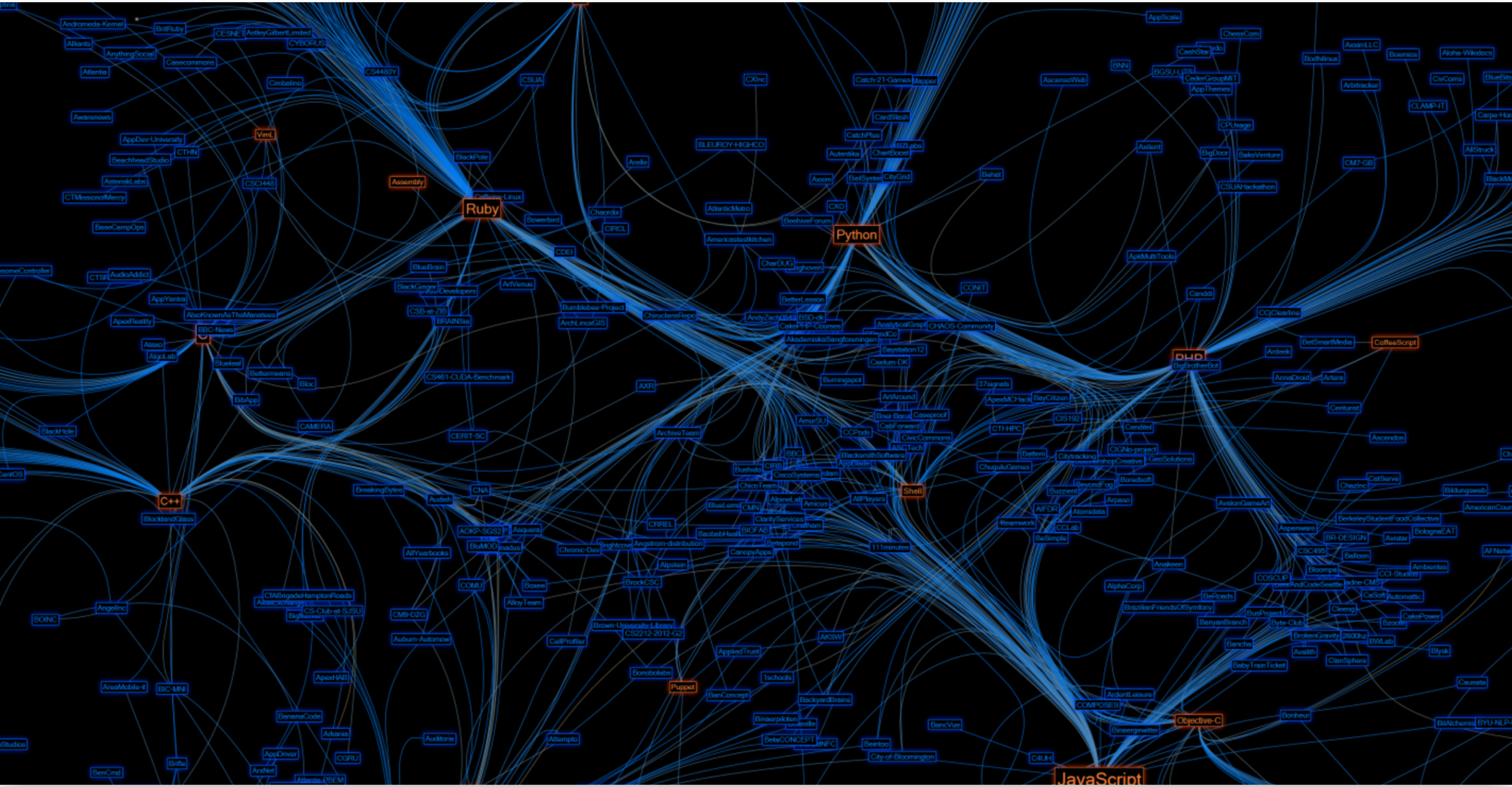
Total GitHub Events



Stats in the past minute

Comments	Commits	Stars	Forks	Pull Requests Opened	PR Merged	Starting Time
39	101	14	5	5	2	22:35:29

https://www.gitlive.net/



http://zoom.it/kCsU

DATA PRODUCT: Exploration

Github public API dump, since 2012

300+ Gb of compressed JSONs in GCE bucket

```
Hourly   2015-01-01-15.json.gz
Daily    2015-01-01-{0..23}.json.gz
Monthly 2015-01-{01..30}-{0..23}.json.gz
```

DATA PRODUCT: Idea

Get notified when a new interesting* project goes Open Source

Definition of “interesting” may vary:
- from a company you like
- similar to other projects you follow
- etc

DATA PRODUCT: Mock

Build an app that sends you a digest email:

Opensourced today

The newest projects on github from the industry leaders

Company	Repository	Language	Description
	google/primarysources	C++	
	facebook/device-year-class	Java	A library that analyzes an Android device's specifications and calculates which year the device would be considered "high end".
	facebook/fresco	Java	An Android library for managing images and the memory they use.
	GoogleCloudPlatform/appengine-multitenancy-python	Python	

DATA PRODUCT: Sketch

We are going to build a Notebook that:

- Downloads the latest data from GitHub Archive
- Read & explore the dataset
- Imports, filters the PublicEvent
- Join logs w/ data from API calls (repo details)
- Shows HTML template, to visualise the list
- Sends email \w this template
- Does all above automatically, once a day

DATA PRODUCT: Full implementation

Demo

http://s.apache.org/zeppelin-github

DATA PRODUCT: Result

Opensourced today

NFLABS

open source today

Alex

Mail

Move to Inbox

More

86 of about 190

COMPOSE

Inbox (3)

Starred

Important

Sent Mail

Drafts (42)

Circles

[Hub Support]

[Imap]/Drafts

[Mailbox]

[OpenSource] (504)

z-manager

Alex

Opensourced today: $date

The newest projects on github from the industry leaders

Company	Repository	Language	Description
	google/primarysources	C++	
	facebook/device-year-class	Java	A library that analyzes an Android device's specifications and calculates which year
	facebook/fresco	Java	An Android library for managing images and the memory they use.

http://s.apache.org/zeppelin-github

DATA PRODUCT: Future steps

Possible future steps:

- buy custom domain
- add static website with email subscription form on
- (optional*) build a startup, rise a seed round

http://s.apache.org/zeppelin-github

* left as an exercise

DATA PRODUCT: Future steps

opensource.sexy C$36.99 Select

+

http://s.apache.org/zeppelin-github

DATA PRODUCT: Next steps

Running the analysis over the full dataset:

- no need to change any code
- just need a GCE cluster
- \w juju 2.0

  edit ~/.juju/environments.yaml, set default: gce

  `$juju deploy apache-hadoop-spark-zeppelin`

- %dep https://storage.googleapis.com/hadoop-lib/gcs/gcs-connector-latest-hadoop2.jar
- run the same notebook \w “gs://data.githubarchive.org/*”

http://s.apache.org/zeppelin-github

* left as an exercise

DATA: 2 datasets

Github Archive

Common Crawl

DATA: Common Crawl

Common Crawl 

https://commoncrawl.org

Nonprofit founded in 2007, by Gil Elbaz of  Factual

“Internet as a dataset”, accessible to everyone for free

- AWS Public Dataset
- S3 bucket, files in WARC, WET, WAT, formats
- since 2013, monthly: ~50Tb compressed, 2+bln ulrs

URL Index by Ilya Kreymer of @webrecorder_io
http://index.commoncrawl.org/

February 2016 Index Info Page

Fork me on GitHub

title: February 2016 Index

Search a url in this collection: (Wildcards -- Prefix: *http://example.com/** Domain: **.example.com*)

wikipedia.org Search

Show Number Of Pages Only

(See the CDX Server API Reference for more advanced query options.)

Back To All Indexes

{"urlkey": "org,wikipedia)/", "timestamp": "20160205193906", "status": "200", "url": "https://www.wikipedia.org/", "filename": "common-crawl/crawl-data/CC-MAIN-2016-07/segments/1454701146196.88/warc/CC-MAIN-20160205193906-00134-ip-10-236-182-209.ec2.internal.warc.gz", "length": "13948", "mime": "text/html", "offset": "873911661", "digest": "Q3INWX2LTYBYB5FYHQJDI3BYX2OYN6ZI"}
{"urlkey": "org,wikipedia)/", "timestamp": "20160206090127", "status": "200", "url": "https://www.wikipedia.org/", "filename": "common-crawl/crawl-data/CC-MAIN-2016-07/segments/1454701146241.46/warc/CC-MAIN-20160205193906-00134-ip-10-236-182-209.ec2.internal.warc.gz", "length": "13954", "mime": "text/html", "offset": "894140307", "digest": "Q3INWX2LTYBYB5FYHQJDI3BYX2OYN6ZI"}
{"urlkey": "org,wikipedia)/", "timestamp": "20160206112239", "status": "200", "url": "https://www.wikipedia.org/", "filename": "common-crawl/crawl-data/CC-MAIN-2016-07/segments/1454701146302.25/warc/CC-MAIN-20160205193906-00134-ip-10-236-182-209.ec2.internal.warc.gz", "length": "13951", "mime": "text/html", "offset": "890420573", "digest": "Q3INWX2LTYBYB5FYHQJDI3BYX2OYN6ZI"}
{"urlkey": "org,wikipedia)/", "timestamp": "20160206132759", "status": "200", "url": "https://www.wikipedia.org/", "filename": "common-crawl/crawl-data/CC-MAIN-2016-07/segments/1454701146550.16/warc/CC-MAIN-20160205193906-00134-ip-10-236-182-209.ec2.internal.warc.gz", "length": "13947", "mime": "text/html", "offset": "890136204", "digest": "Q3INWX2LTYBYB5FYHQJDI3BYX2OYN6ZI"}
{"urlkey": "org,wikipedia)/", "timestamp": "20160206152725", "status": "200", "url": "https://www.wikipedia.org/", "filename": "common-crawl/crawl-data/CC-MAIN-2016-07/segments/1454701146600.56/warc/CC-MAIN-20160205193906-00134-ip-10-236-182-209.ec2.internal.warc.gz", "length": "13950", "mime": "text/html", "offset": "874751947", "digest": "Q3INWX2LTYBYB5FYHQJDI3BYX2OYN6ZI"}
{"urlkey": "org,wikipedia)/", "timestamp": "20160206182942", "status": "200", "url": "https://www.wikipedia.org/", "filename": "common-crawl/crawl-data/CC-MAIN-2016-07/segments/1454701147492.21/warc/CC-MAIN-20160205193907-00134-ip-10-236-182-209.ec2.internal.warc.gz", "length": "13946", "mime": "text/html", "offset": "898718776", "digest":

N-gram Counts and Language Models from the Common Crawl

Christian Buck[†], **Kenneth Heafield**[‡], **Bas van Ooyen**[*]
[†]University of Edinburgh, Edinburgh, Scotland
[‡]Stanford University, Stanford, CA, USA
[*] Owlin BV, Utrecht, Netherlands
christian.buck@ed.ac.uk, heafield@cs.stanford.edu, bas@owlin.com

Abstract

We contribute 5-gram counts and language models trained on the Common Crawl corpus, a collection over 9 billion web pages. This release improves upon the Google n-gram counts in two key ways: the inclusion of low-count entries and deduplication to reduce boilerplate. By preserving singletons, we were able to use Kneser-Ney smoothing to build large language models. This paper describes how the corpus was processed with emphasis on the problems that arise in working with data at this scale. Our unpruned Kneser-Ney English 5-gram language model, built on 975 billion deduplicated tokens, contains over 500 billion unique n-grams. We show gains of 0.5–1.4 BLEU by using large language models to translate into various languages.

Keywords: web corpora, language models, multilingual

1. Introduction

The sheer amount of data in multiple languages makes web-scale corpora attractive for many natural language processing tasks. Of particular importance is language modeling, where web-scale language models have been shown to improve machine translation and automatic speech recognition performance (Brants et al., 2007; Chelba and Schalkwyk, 2013; Guthrie and Hepple, 2010). In this work, we contribute n-gram counts and language models trained on

2. Data Preparation

The Common Crawl[2] is a publicly available crawl of the web. We use the 2012, early 2013, and "winter" 2013 crawls, consisting of 3.8 billion, 2 billion, and 2.3 billion pages, respectively. Because both 2013 crawls are similar in terms of seed addresses and distribution of top-level domains in this work we only distinguish 2012 and 2013 crawls.

The data is made available both as raw HTML and as text

https://about.commonsearch.org

common search: wikipedia EN About

Wikipedia for Schools
schools-wikipedia.org
Welcome to Wikipedia for Schools ! This selection of articles from Wikipedia matches the UK National Curriculum and can be used by school children around the...

Wikipedia Review
www.wikipediareview.com
Putting the wakeup alarmclock to Wikipedia's head since... Oh god, it's been that long?

Wikiwix » Wikipedia
www.wikiwix.com
Search Last visited websites Categories Favorite websites

Wikipedia Toolbar: Home
wikipedia.mozdev.org
Wikipedia Toolbar gives you quick access to usefull commands for the Mediawiki Software. It also provides a backend for other types of Wiki software. It also...

Wikipedia (TheFreeDictionary.com mirror)
www.encyclopedia.farlex.com
Wikipedia is a Web-based, free-content encyclopedia written collaboratively by volunteers and sponsored by the non-profit Wikimedia Foundation. It contains...

Welcome! This is a **demo** of the Common Search interface.

The search results are NOT complete/relevant. For this demo they are restricted to **some homepages** from the Web.

Can you help us improve this interface? We are looking for contributors!

OK, I understand this is a demo.

DATA: Common Crawl - existing projects

Web Data Commons - Hyperlink Graphs

UNIVERSITY OF MANNHEIM

- 2012 - 3.5 billion web pages 128 billion hyperlinks
- 2014 - 1.7 billion web pages, 64 billion hyperlinks

Web Data Commons - Web Tables, 233 mil. tables

Parallel text corpus

N-Grams for language models

GloVe - Unsupervised ML for word meaning representation
http://www-nlp.stanford.edu/projects/glove/

US phone numbers (by Neftflix)
http://engineeringblog.yelp.com/2015/03/analyzing-the-web-for-the-price-of-a-sandwich.html

DATA: Common Crawl - Product Idea

Measuring the impact of Google Analytics

Objective: estimate % of pages/domains that use Google Analytics

Harvard research on 2013 data: 39.7% of web impacted

http://smerity.com/cs205_ga/ by C. Hornbaker and S. Merity

using Hadoop + EMR + 2012 CommonCrawl dataset

DATA: Common Crawl - Data Product

Measuring the impact of Google Analytics

1. **build intuition around the data**
2. experiment\prototype on single machine
3. run on fraction of the data
4. scale out

Demo

DATA: Common Crawl - Warcbase

https://github.com/lintool/warcbase

Spark library for WARC (Web ARChive) data processing
 * text analysis
 * site link structure

```
import org.warcbase.spark.matchbox._
import org.warcbase.spark.rdd.RecordRDD._

val r =
RecordLoader.loadArc("/directory/to/arc/file.arc.gz", sc)
.keepValidPages()
.map(r => ExtractTopLevelDomain(r.getUrl))
.countItems()
.take(10)
```

http://lintool.github.io/warcbase-docs

DATA: Common Crawl - Data Product

Measuring the impact of Google Analytics

1. build intuition around the data
2. **experiment\prototype on single machine**
3. run on fraction of the data
4. scale out

Demo

DATA: Common Crawl - Data Product

Measuring the impact of Google Analytics

1. build intuition around the data
2. experiment\prototype on single machine
3. **run on fraction of the data**
4. scale out

DATA: Common Crawl - Data Product

Measuring the impact of Google Analytics

Deploy small cluster of AWS using Juju: with Apache Zeppelin, Hadoop (YARN), Spark

Same as before:

edit ~/.juju/environments.yaml, set default: amazon

```
$juju deploy apache-hadoop-spark-zeppelin \
  -constraints "cpu-cores=8 mem=32G"
```

DATA: Common Crawl - Data Product

Measuring the impact of Google Analytics

Deploy small cluster of AWS using Juju: with Apache Zeppelin, Hadoop (YARN), Spark

Experiment results:

- 0,0025% (1/400 of Segment, 1Gb)
- 0,1% (1/10 of Segment, 40 Gb)
- 1% (1 segment, 0,4Tb)
- 10% (10 segment, 4Tb)

DATA: Common Crawl - Data Product

Measuring the impact of Google Analytics

Deploy small cluster of AWS using Juju: with Apache Zeppelin, Hadoop (YARN), Spark

Experiment results:

- 0,0025% (1/400 of Segment, 1Gb) 31,3% 11993 out of 38244
- 0,1% (1/10 of Segment, 40 Gb)
- 1% (1 segment, 0,4Tb)
- 10% (10 segment, 4Tb)

follow @ApacheZeppelin for updates

DATA: Common Crawl - Data Product

AWS optimisations:
 - pick spot instances
 - pick instance type wise (max out net throughput)
 - Juju instead of EMR = 2x $$ savings!

Spark optimisations:
 - IO-bound, so get more then 1 executor per-machine: increase and adjust `spark.executor.cores`, `spark.executor.memory`

DATA: Common Crawl - new projects?

Impact of Facebook\Twitter\Google\Any other service

Mine phone numbers, books, bitcoin addresses

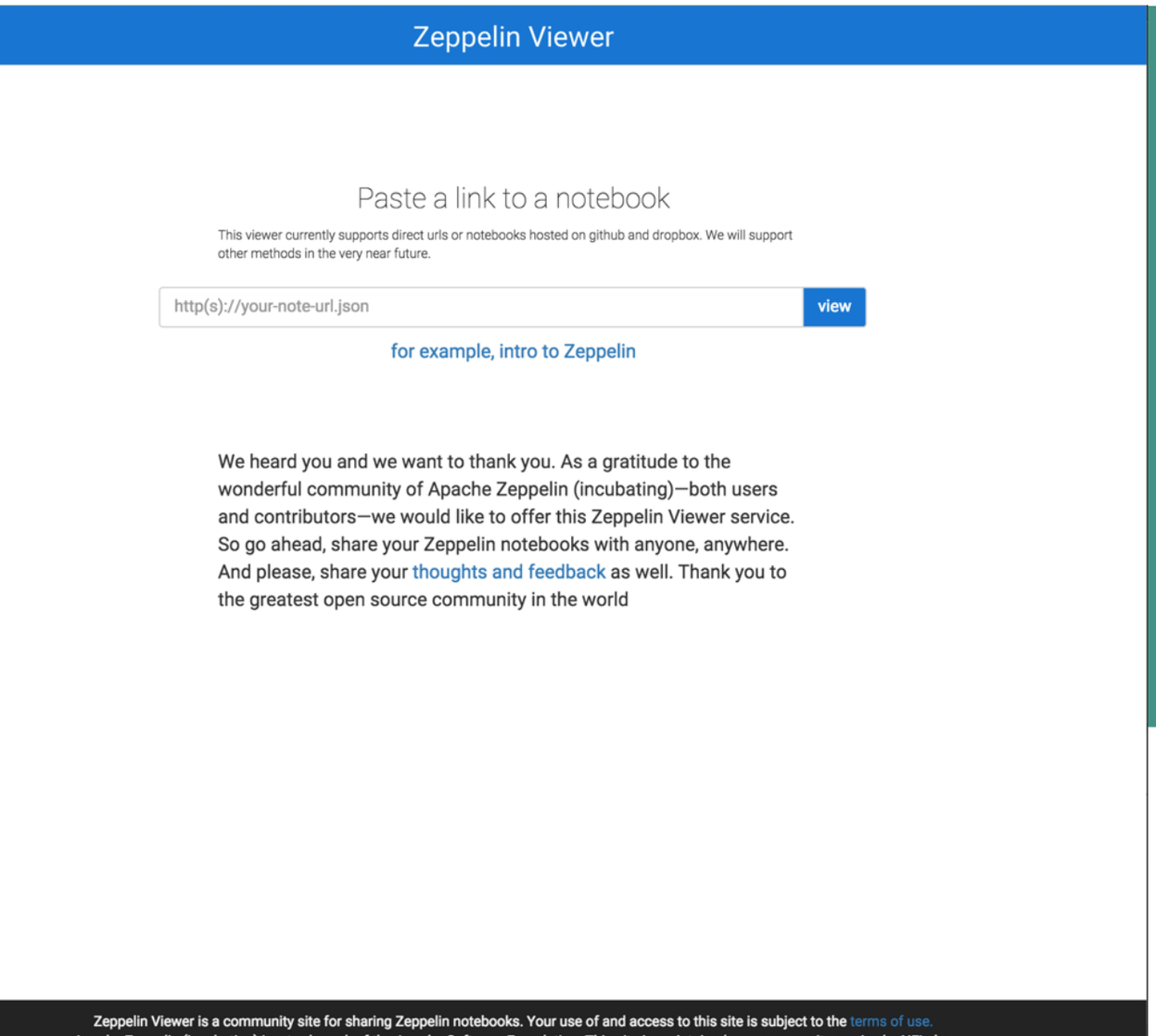
PDF, .xls, robots.txt, apply Apache Ticka to extract

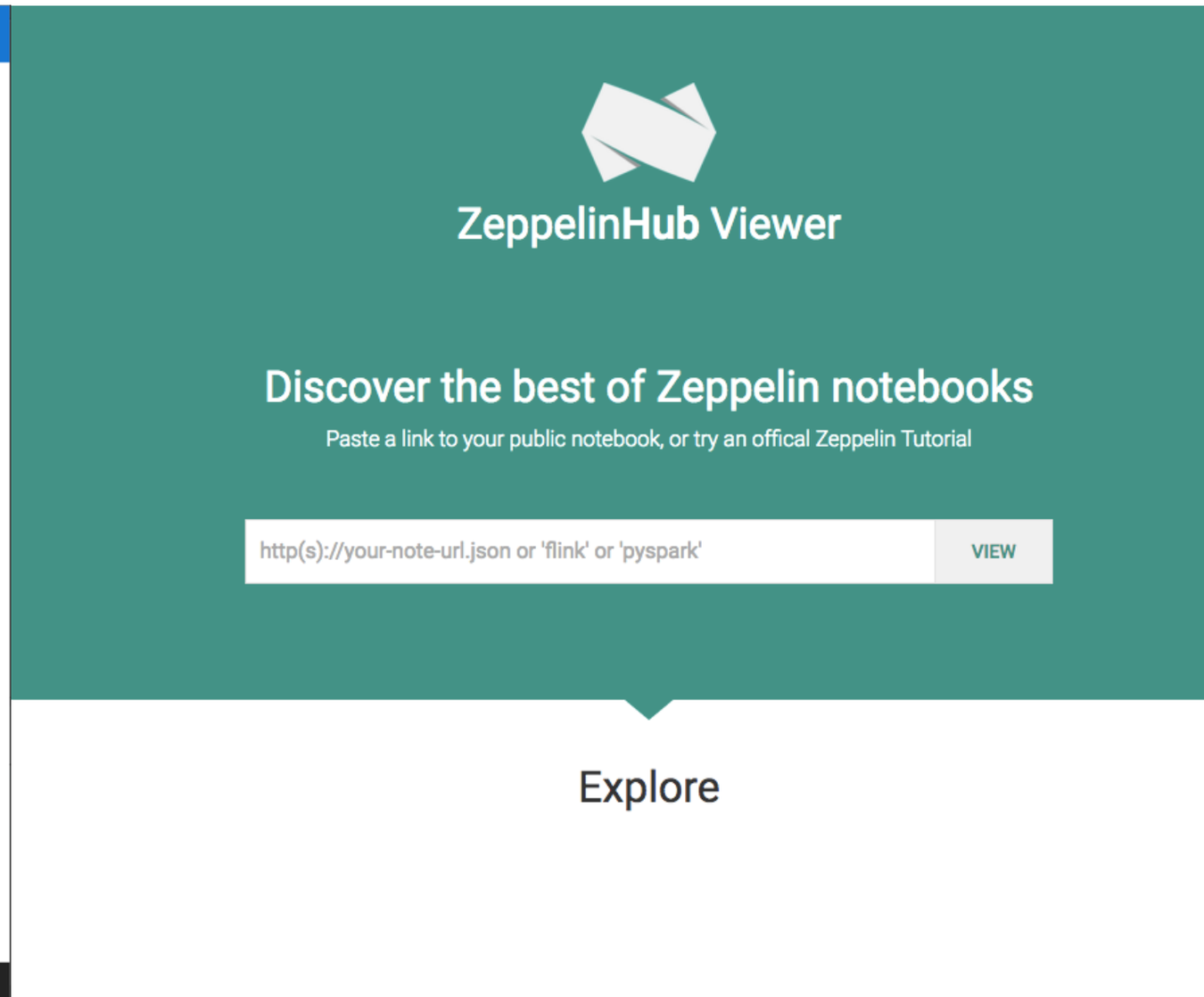
Which CMS is more popular? web framework? web server?

or just build distributed search engine!

Zeppelin Viewer: share your notebooks

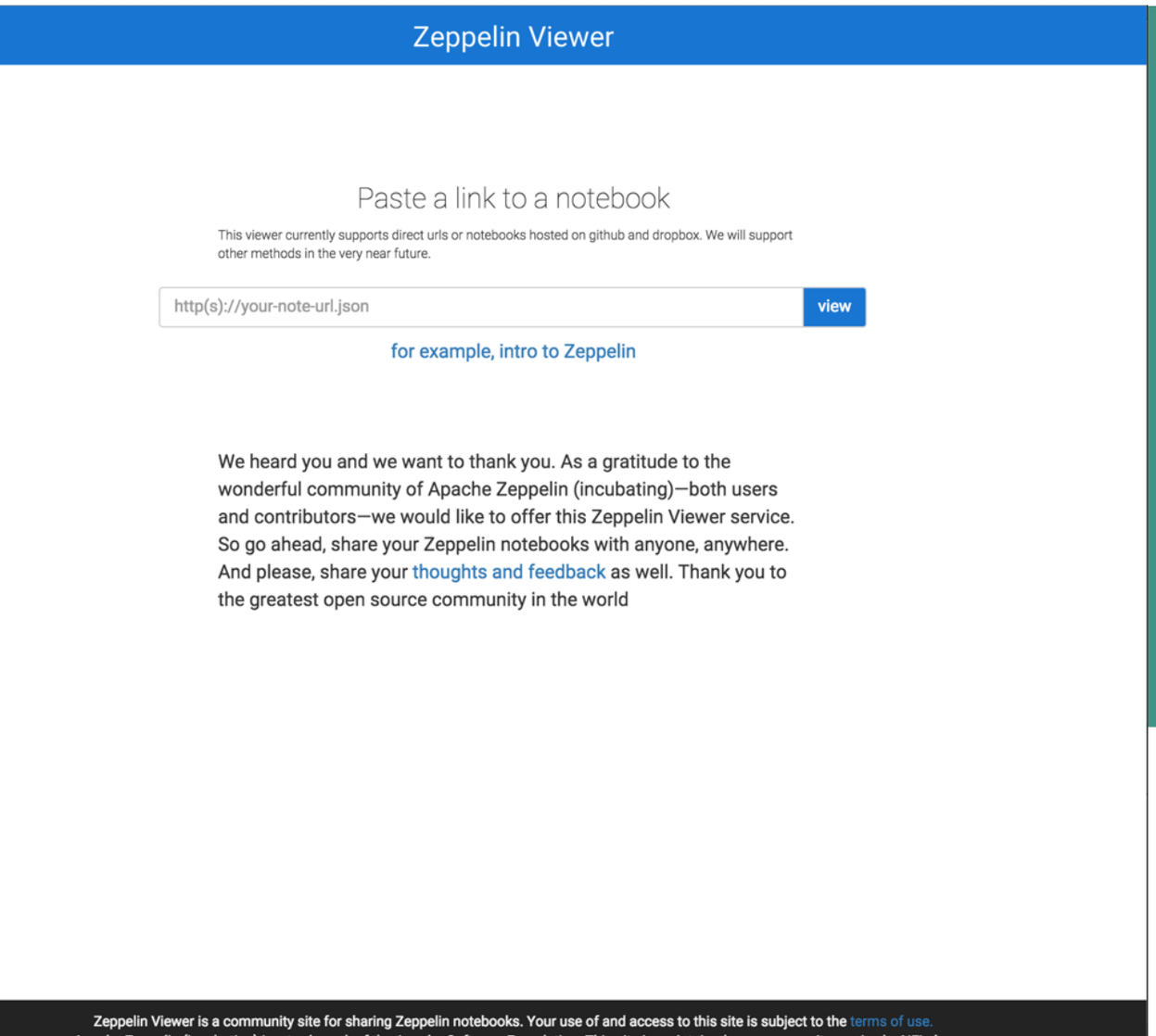
ApacheCon'15 community service for publishing notebooks was launche
http://zeppelinhub.com/viewer

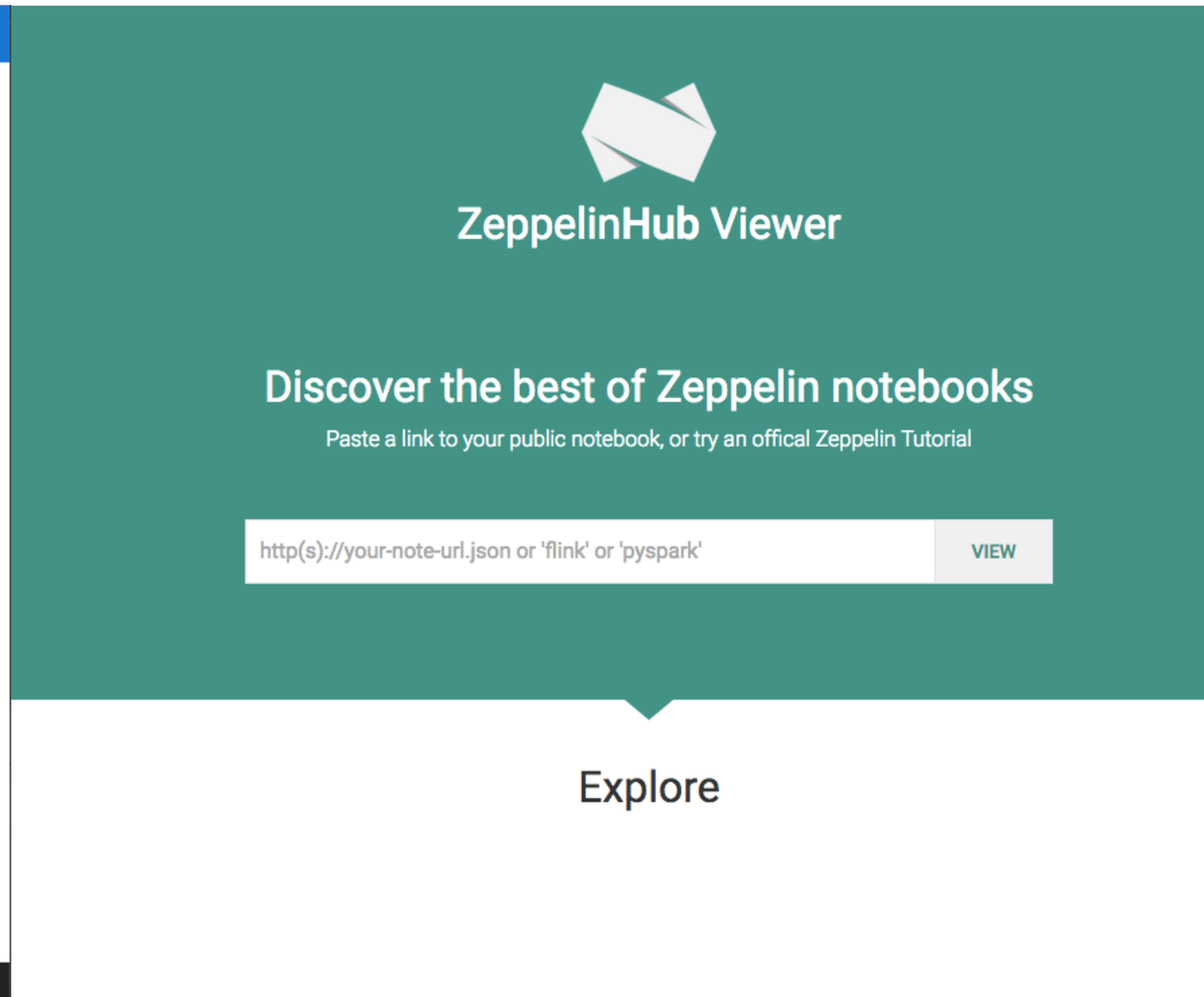




Zeppelin Viewer: share your notebooks

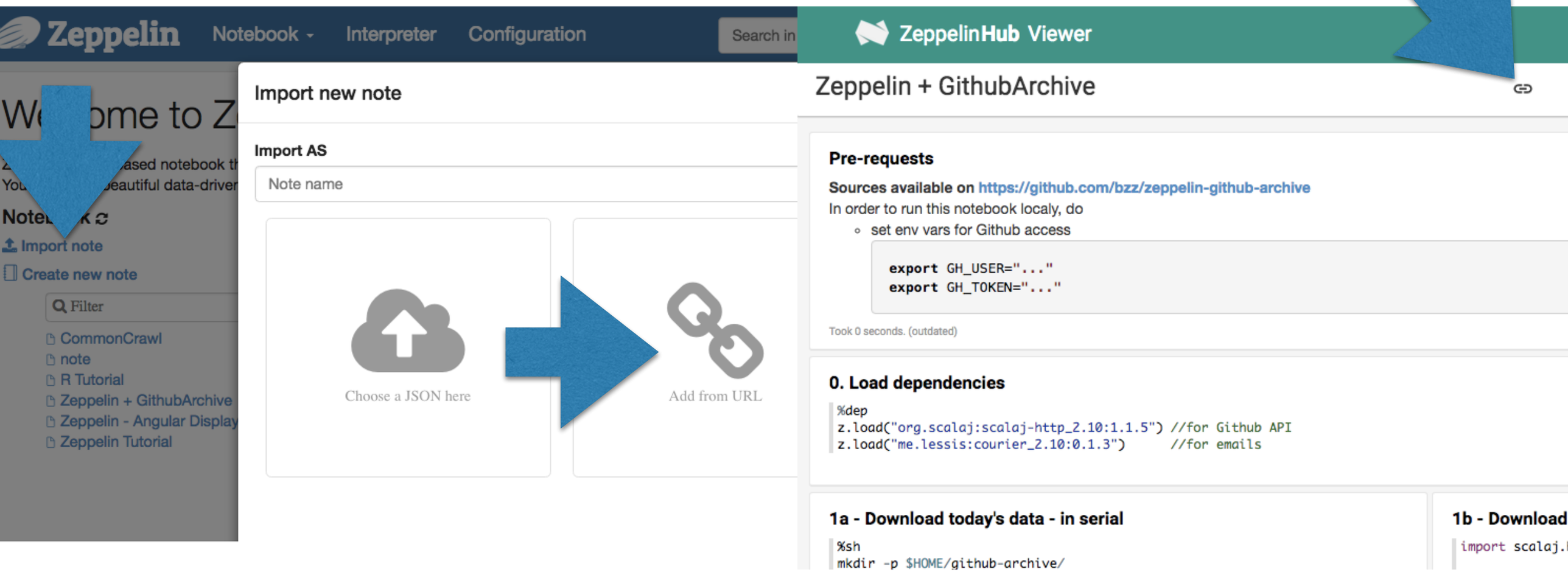
ApacheCon ’16 it is updated,
http://zeppelinhub.com/viewer





Zeppelin Viewer: publish your notebooks

Import notes from http://s.apahce.org/zeppelin-github
or https://github.com/bzz/zeppelin-github-archive



TAKEAWAY

There are plenty of free tools

To crunch the data for fun and profit

are easy to grasp and generic enough to be useful

3 tools: Apache Zeppelin, Spark, Juju
2 dataset: CommonCrawl, Github Archive

Thank you

Alexander Bezzubov

NFLabs, Seoul (we are hiring!)

Questions?

Alexander Bezzubov

@seoul_engineer

github.com/bzz