2017 2nd International Symposium on
Spatiotemporal Computing
Harvard University

Building A Billion Spatio-Temporal Object Search and Visualization Platform

Devika Kakkar, Benjamin Lewis

Goal

Develop a platform to make it easier for researchers to interactively explore large spatio-temporal datasets.

Initial focus on Geo-tweets

(could be any streaming dataset)

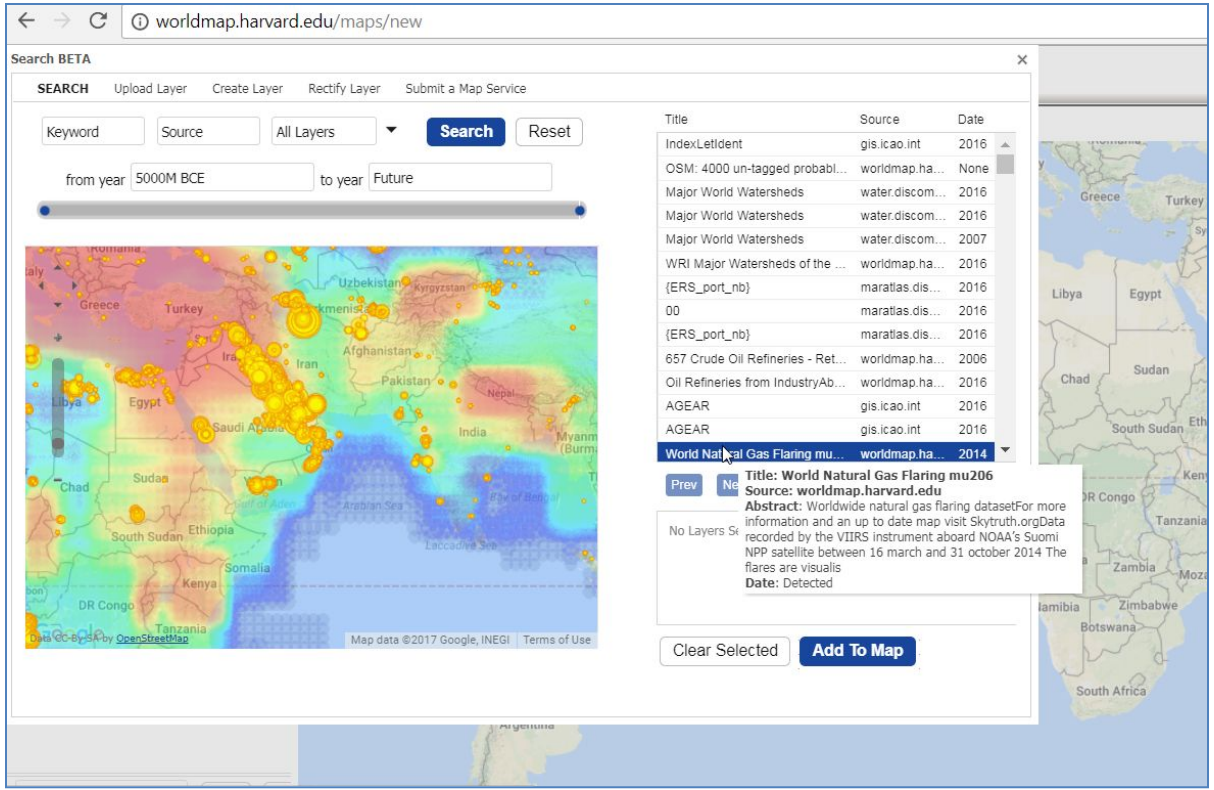
- 1-2% of tweets have GPS coordinates from the user's device, currently about 1 million per day available via the Twitter API
- The CGA has been harvesting geo-tweets since 2012 and has an informal archive of about 8 billion objects
- Northeastern Professor Ryan Qi Wang also harvested during this period and we plan to eventually merge the two datasets to create a more complete version.

Requirements

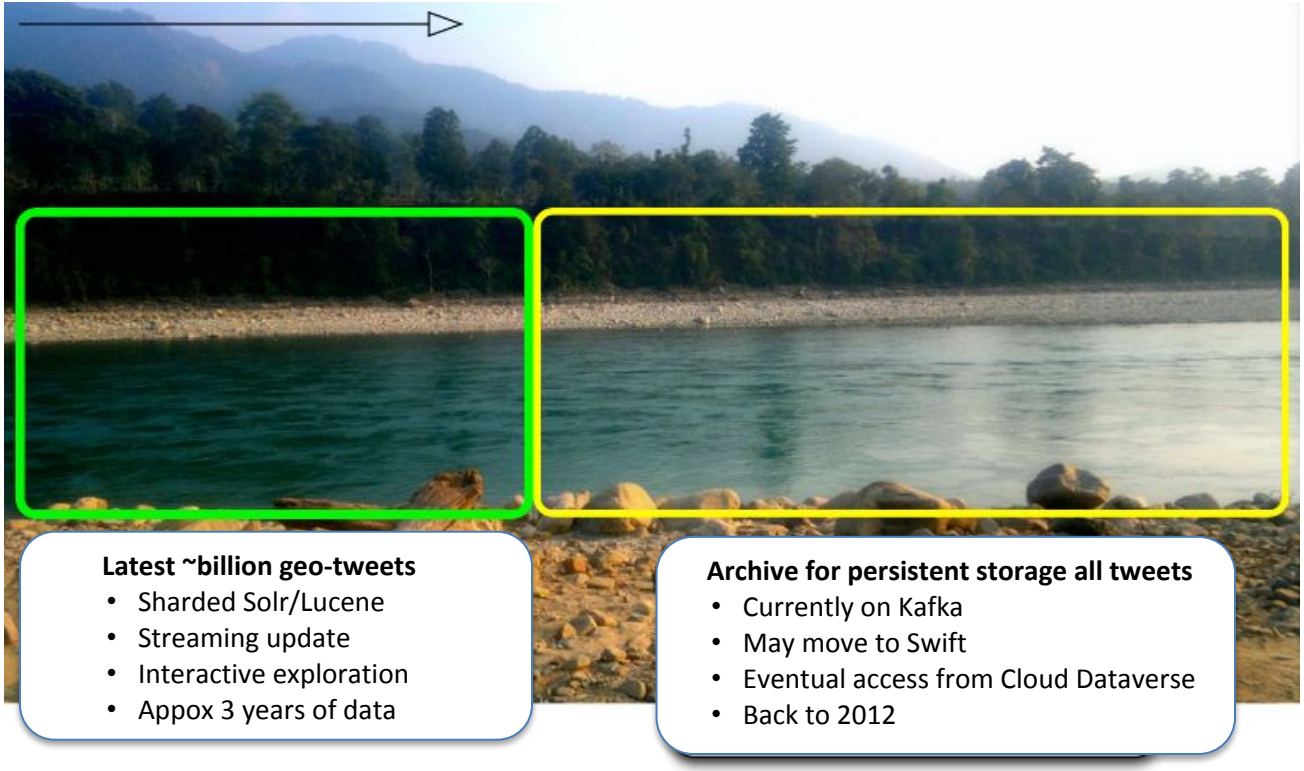
- Develop back end and client to support interactive visualization of a billion point objects
- Support sub-second queries including heatmaps and temporal histograms
- Expose a general purpose RESTful API so other clients could access the data
- System should run on low cost commodity hardware or VMs

Big data visualization built on 2D faceting

Developed for HHypermap in 2015 (layer search)



Latest billion + long term archive



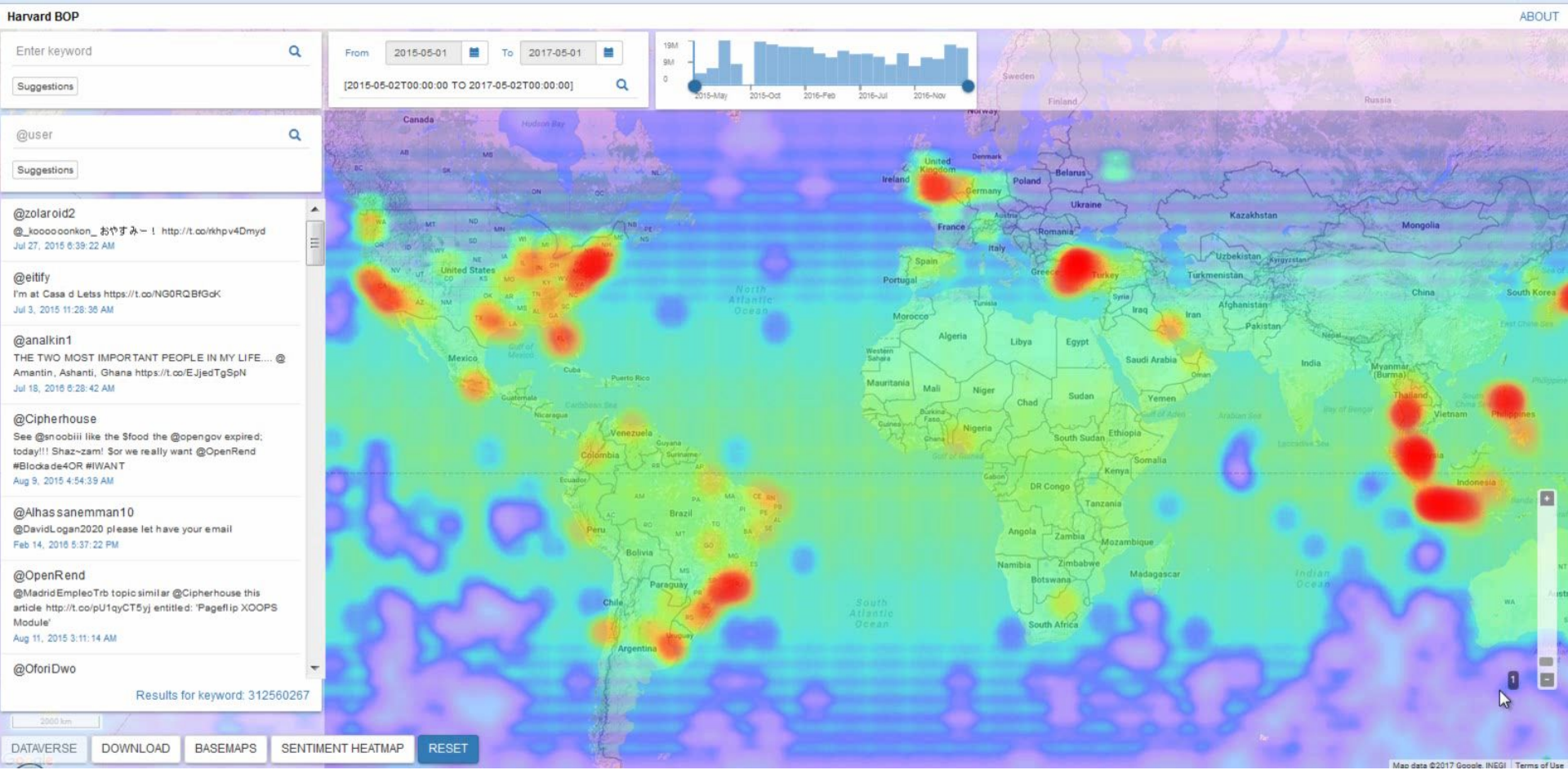
Latest ~billion geo-tweets

- Sharded Solr/Lucene
- Streaming update
- Interactive exploration
- Appox 3 years of data

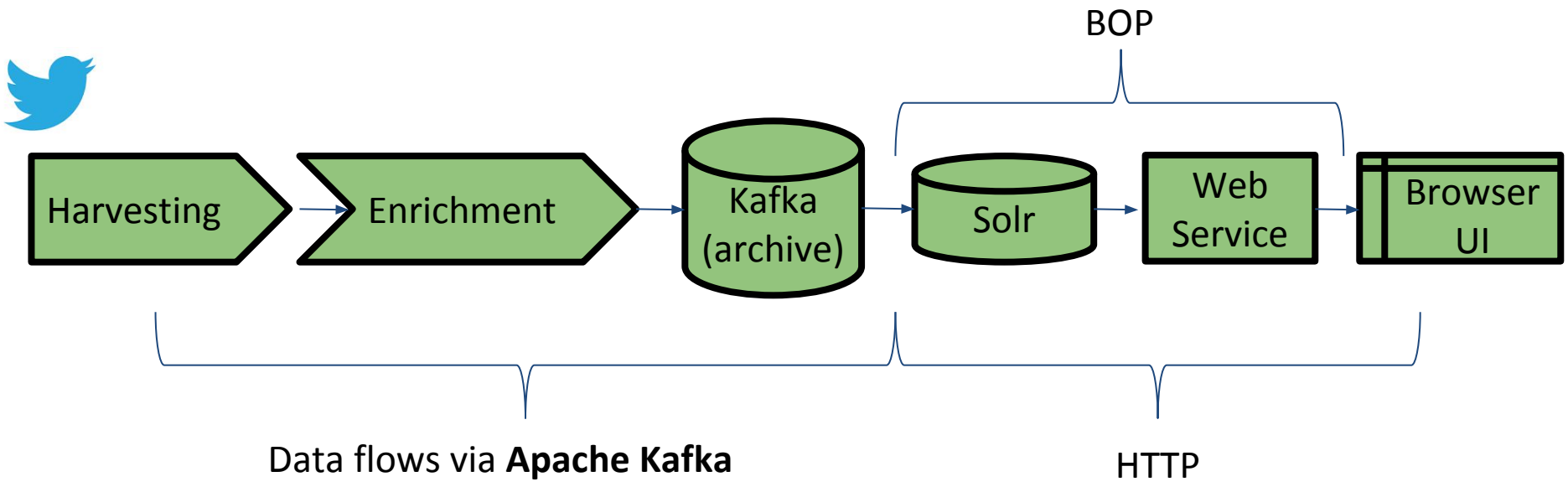
Archive for persistent storage all tweets

- Currently on Kafka
- May move to Swift
- Eventual access from Cloud Dataverse
- Back to 2012

Demo


Harvard BOP
ABOUT
Enter keyword
Suggestions
From
2015-05-01
To
2017-05-01
[2015-05-02T00:00:00 TO 2017-05-02T00:00:00]
19M
9M
0
2015-May
2015-Oct
2016-Feb
2016-Jul
2016-Nov
@user
Suggestions
@zolaroid2
@_kooooonkon_ おやすみー！ http://t.co/rkhpv4Dmyd
Jul 27, 2015 6:39:22 AM
@eitify
I'm at Casa d Letss https://t.co/NG0RQBfGcK
Jul 3, 2015 11:28:36 AM
@analkin1
THE TWO MOST IMPORTANT PEOPLE IN MY LIFE.... @ Amantin, Ashanti, Ghana https://t.co/EJjedTgSpN
Jul 18, 2015 6:28:42 AM
@Cipherhouse
See @snoobiii like the $food the @opengov expired; today!!! Shaz~zam! $or we really want @OpenRend #Blockade4OR #IWANT
Aug 9, 2015 4:54:39 AM
@Alhassanemman10
@DavidLogan2020 please let have your email
Feb 14, 2016 5:37:22 PM
@OpenRend
@MadridEmpleoTrb topic similar @Cipherhouse this article http://t.co/pU1qyCT5yj entitled: 'Pageflip XOOPS Module'
Aug 11, 2015 3:11:14 AM
@OforiDwo
Results for keyword: 312560267
DATAVERSE
DOWNLOAD
BASEMAPS
SENTIMENT HEATMAP
RESET
Map data ©2017 Google, INEGI
Terms of Use

Logical High-Level Architecture

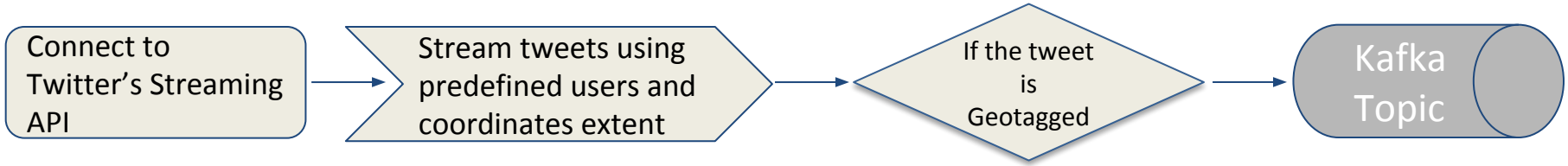


Docker, Kontena, OpenStack Hosting: Mass OpenCloud

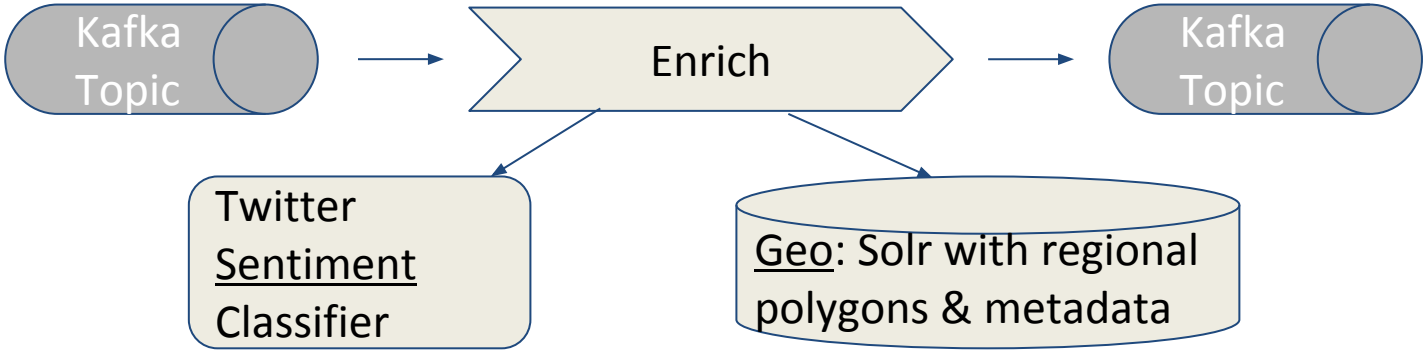
Apache Kafka

- Kafka: a scalable message/queue platform
- See new Kafka Streams & Kafka Connect APIs
- No back-pressure; can be a challenge
- Non-obvious use:
  - For storage; time partitioning
    - Lots of benefits yet serious limitations

Real-Time Harvesting



Enrichment

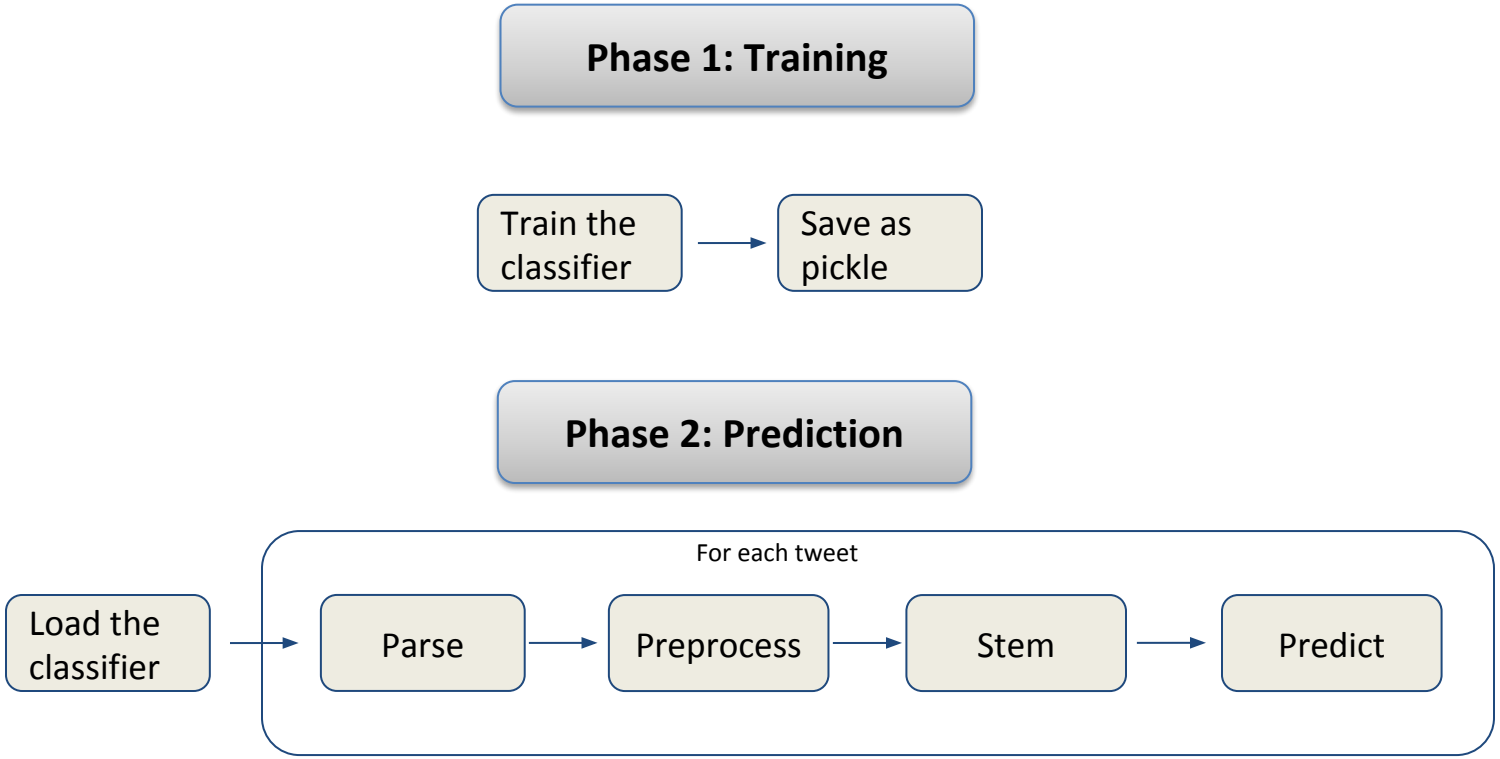


Geo: Query Solr via spatial point query; attach related metadata to tweet

Sentiment Analysis

Component	Description
Classifier	Support Vector Machine with Linear Kernel
Source Code	Python
Libraries	Scikit-learn, numpy, NLTK, scipy
Classes of sentiment	Positive (1) and Negative (0)
Training Corpus	Stanford Sentiment140, Polarity dataset v2.0, University of Michigan
Preprocessing	Lower case, URLs, @user, #tags, trimming, repeating characters, emoticons
Stemming	Porter Stemmer
Precision, recall and f1-score	0.82 (82%)
Processing speed	20ms/tweet (no emoticon), 5ms/tweet(emoticon)

Sentiment Analysis



Sentiment Analysis

- Classifier: Support Vector Machine (SVM) with Linear Kernel
- Source code in Python
- Uses scikit-learn, numpy, scipy, NLTK
- Two classes of sentiment: Positive (1), Negative (0)
- Training Corpus: Sentiment140, Polarity dataset v2.0, University of Michigan
- Preprocessing: Lower case, URLs, @user, #tags, trimming, repeating characters, emoticons
- Stemming: Porter stemmer
- Precision, Recall, F1 score: 0.82 (82%)
- Processing speed: 20ms/tweet (no emoticon), 5ms/tweet (emoticon)

Solr for Geo Enrichment

"Reverse Geocoding"

- Tweets (docs) can have a geo lat/lon
- Enrich tweet with Country, State/Province, …
  - Gazetteer lookup (point-in-polygon)

Data Set	Features	Raw size	Index time	Index size
Admin2	46,311	824 MB	**510 min**	892 MB
US States	74,002	747 MB	4.9 min	840 MB
Massachusetts Census Blocks	154,621	152 MB	5.9 min	507 MB

Apache Solr

- Search / analytics server, based on Lucene
- Custom add-ons:
  - Time sharded routing (index + query)
  - LatLonPointSpatialField – in Solr 6.5
    - Faster/leaner search & sort for point data
  - HeatmapSpatialField – in Solr 6.6 TBD
    - Faster/leaner heatmaps at scale

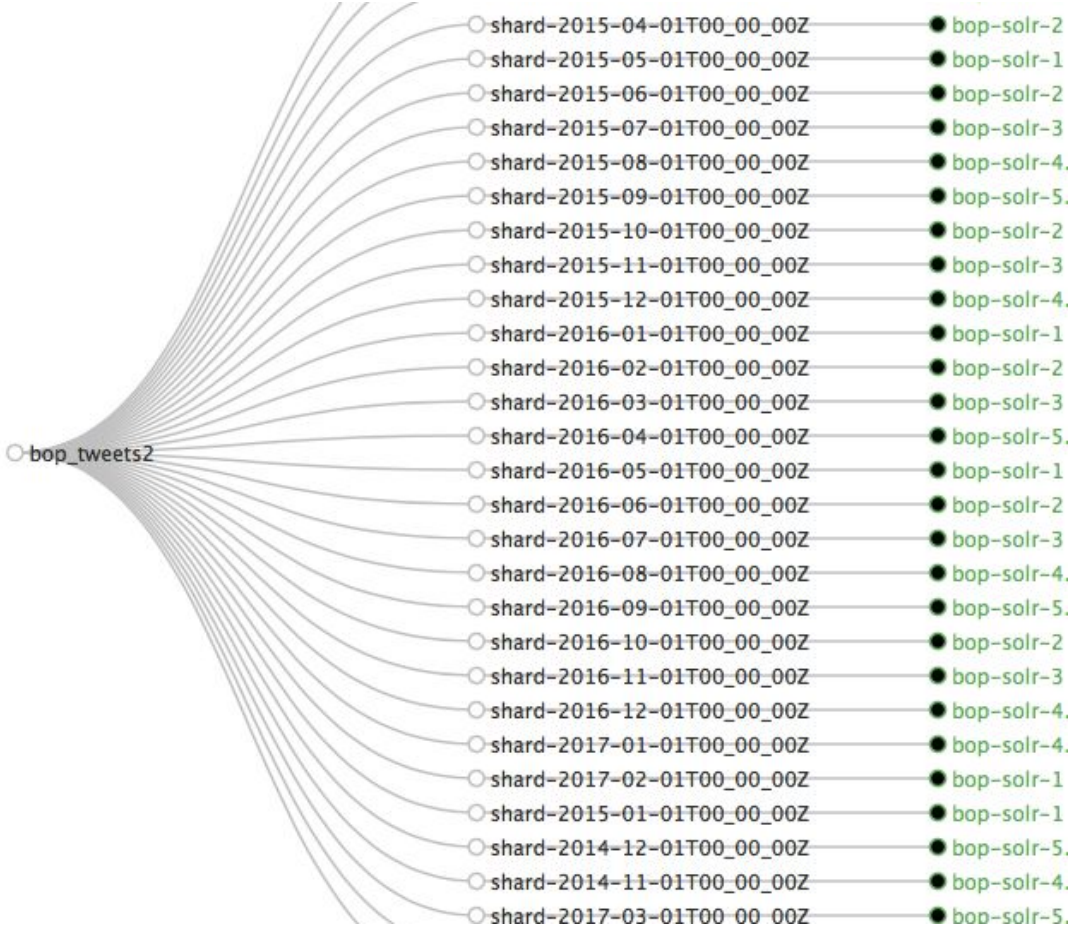
Time “Sharding”

Solr has no built-in time based sharding.

A Solr custom “URP” was developed to route tweets to the right by-month shard. It auto creates and deletes shards.

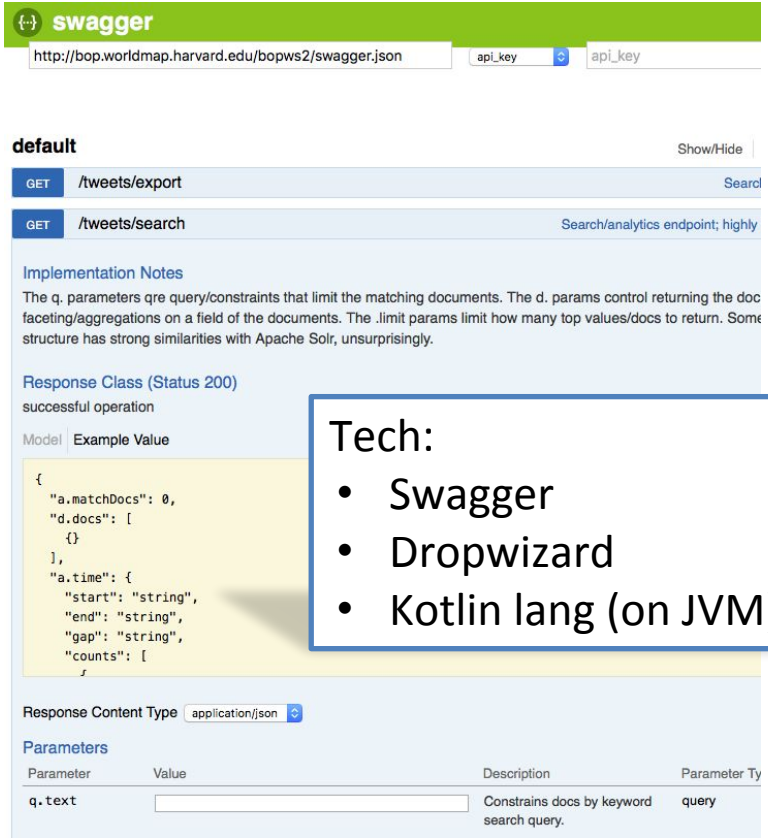
A Solr custom “SearchHandler” was developed to decide which subset of shards to search based on custom parameters sent by the web-service.

Generally useful for others. Need more work for contribution to Solr itself.



The BOP Web-Service

- HTTP/REST API
  - Keyowrd search
  - Faceting
    - Heatmaps
  - CSV export
- Why not Solr direct?
  - Define a supported API
  - Ease of use for clients
  - Security

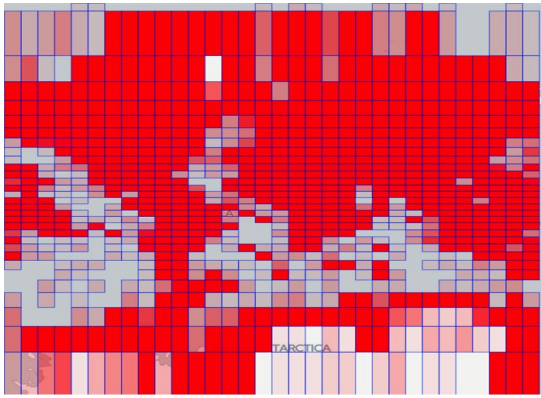


Tech:

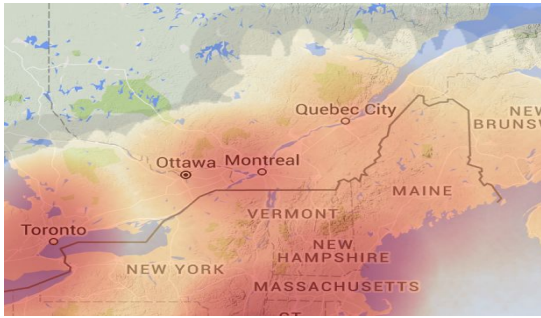
- Swagger
- Dropwizard
- Kotlin lang (on JVM)

Heatmaps: Spatial Grid Faceting

- Spatial density summary grid faceting, also useful for point-plotting search results
- Lucene & Solr APIs
- Scalable & fast *usually*…



- Usually rendered with a gradient radius ->
- See: http://spacemansteve.github.io/leaflet-solr-heatmap/example/index.html



UI Stack

- BOP's UI uses the following technologies:
  - Angular JS
  - OpenLayers 3
  - npm (dependencies, script minification, development)

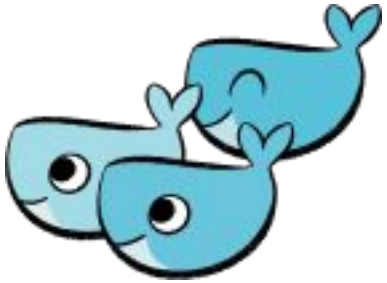
Deployment / Operations

- MassOpenCloud “MOC”
  - OpenStack based cloud (mimics Amazon EC2)
- CoreOS
- Kontena & Docker
- Admin/Ops tools:
  - Kafka Manager (Yahoo!)
  - Solr’s admin UI

Stats:

- 12 nodes (machines)
  - 5 to Solr
  - 3 to Kafka
  - 3 to enrichment, …
- 217 GB RAM
- 3500 GB disk
- 17 services (software pieces)
  - 133 containers

Docker



- Easy to find/try/use software
  - No installation
  - Simplified configuration (env variables)
  - Common logging
  - Isolated
- Ideal for:
  - Continuous Int. servers
  - Trying new software
  - Production advantages too
    - but “new”

Docker in Production

- We use “Kontena”
- Common logging, machine/proc stats, security
  - VPN to secure network; access everything as local
- No longer need to care about:
  - Ansible, Chef, Puppet, etc.
  - Security at network or proxy; not service specific
- Challenges: state & big-data

Next steps

- Persistent archive in Swift object storage
- Exploring adding analytic capabilities using GeoMesa
- Support faceting on numeric values (in addition to counts) to support other types of visualizations.

Thank you

Devika Kakkar
kakkar@fas.harvard.edu
Ben Lewis
blewis@cga.harvard.edu