DATABRICKS

Apache Spark & Elasticsearch

Holden Karau - Spark Summit 2014

Who am I?

Holden Karau

- Software Engineer @ Databricks
- I've worked with Elasticsearch before
- I prefer she/her for pronouns
- Author of a book on Spark and co-writing another
- github https://github.com/holdenk
  - Has all of the code from this talk :)
- e-mail holden@databricks.com
- @holdenkarau

What is Elasticsearch?

- Lucene based distributed search system
- Powerful tokenizing, stemming & other IR tools
- Geographic query support
- Capable of scaling to many nodes

Coffee Berkeley, CA

Foursquare Labs, Inc. [US] https://foursquare.com/explore?cat=coffee&mode=url&near=Berkeley%2C%20CA

foursquare | I'm looking for... | Current Map View | 29 | Holden

Suggestions for **Coffee** near **Berkeley**

Show me: Specials | Haven't Been | Friends | Price | Open Now | Saved

1. **Asha Tea House**
   8.7 | 2086 University Ave (at Shattuck Ave)
   Tea Room · $ $ $ $ · View Menu
   2 friends liked this place
   "... latte with **boba** is the bee's knees." (6 tips)
   "... tea, but **matcha latte** ($3.50) was quite..." (4 tips)
   "... else offers **hojicha lattes**." (2 tips)
   Save | have been here

2. **Philz Coffee**
   9.2 | 1600 Shattuck Ave. (at Cedar St.)
   Coffee Shop · 4 · $ $ $ $
   Josh A. liked this place
   "... to your Iced **Mocha Tesora**." (7 tips)
   "The **mint mojito iced coffee** is maybe the best..." (5 tips)
   "... Mojito and **Gingersnap ice** coffee are the nest..." (3 tips)
   Save | +2

3. **Caffe Strada**
   8.9 | 2300 College Ave (at Bancroft Way)
   Coffee Shop · 1 · $ $ $ $
   "grab a **strada bianca mocha** and hang out on the..." (3 tips)
   "... Bianca Mocha + **nut bun**" (2 tips)
   "... mac, but it **wouldn't work**, other people were having..." (2 tips)
   Save | have been here

4. **Starbucks**
   7.6 | 2224 Shattuck Ave
   Coffee Shop · $ $ $ $ · View Menu
   Starbucks • May 21, 2014
   Looking to try something unique? Try one of our Reserve Coffees. Our Brazil Bourbon Rio Verde is a beautifully balanced cup with notes of plum and chocolate.
   People talk about: **venti, peppermint**
   Save | cooper b. has been here.

5. **Au Coquelet Cafe**
   8.0 | 2000 University Ave (at Milvia St)
   Café · $ $ $ $ · View Menu
   "... sure you **charge your laptop** before you get there -..." (2 tips)
   "**Open late**, excellent study/drawing..." (3 tips)
   "... Mousse **Torte**. Worth going out of your way..." (2 tips)
   Save | have been here

6. **Uncommon Café**
   5.9 | 2813 7th St.
   Coffee Shop · $ $ $ $ · View Menu
   Cass M. liked this place

Search this area

ANNEX | SUNSET VIEW CEMETERY | TILDEN PARK | EAST SHORE STATE PARK | ALBANY HILL PARK | THOUSAND OAKS | CRAGMONT | ALBANY BULB | BERKELEY HILLS | TILDEN GOLF COURSE | Albany | UNIVERSITY VILLAGE | NORTHBRAE | LA LOMA PARK | WESTBRAE | NORTHSIDE | CESAR CHAVEZ PARK | OCEAN VIEW | CENTRAL BERKELEY | BERKELEY | PANORAMIC HILL | WEST BERKELEY | BERKELEY MARINA | SOUTHSIDE | POETS CORNER | LECONTE | ELMWOOD | GRIZZLY PEAK | AQUATIC PARK | SOUTH BERKELEY | CLAREMONT | HARMON TRACT | IDORA PARK | ROCKRIDGE | TEMESCAL | CLAREMONT COUNTRY CLUB | MOUNTAIN VIEW CEMETERY | Emeryville | PIEDMONT AVENUE | Piedmont

Privacy Terms © Mapbox © OpenStreetMap

DATABRICKS

Talk overview

Goal: understand how to work with ES & Spark

- Spark & Spark streaming let us re-use indexing code
- We can customize the ES connector to write to the shard based on partition
- Illustrate with twitter & show top tags per region
- Maybe a live demo of the above demo*

Assumptions:

- Familiar(ish) with Search
- Can read Scala

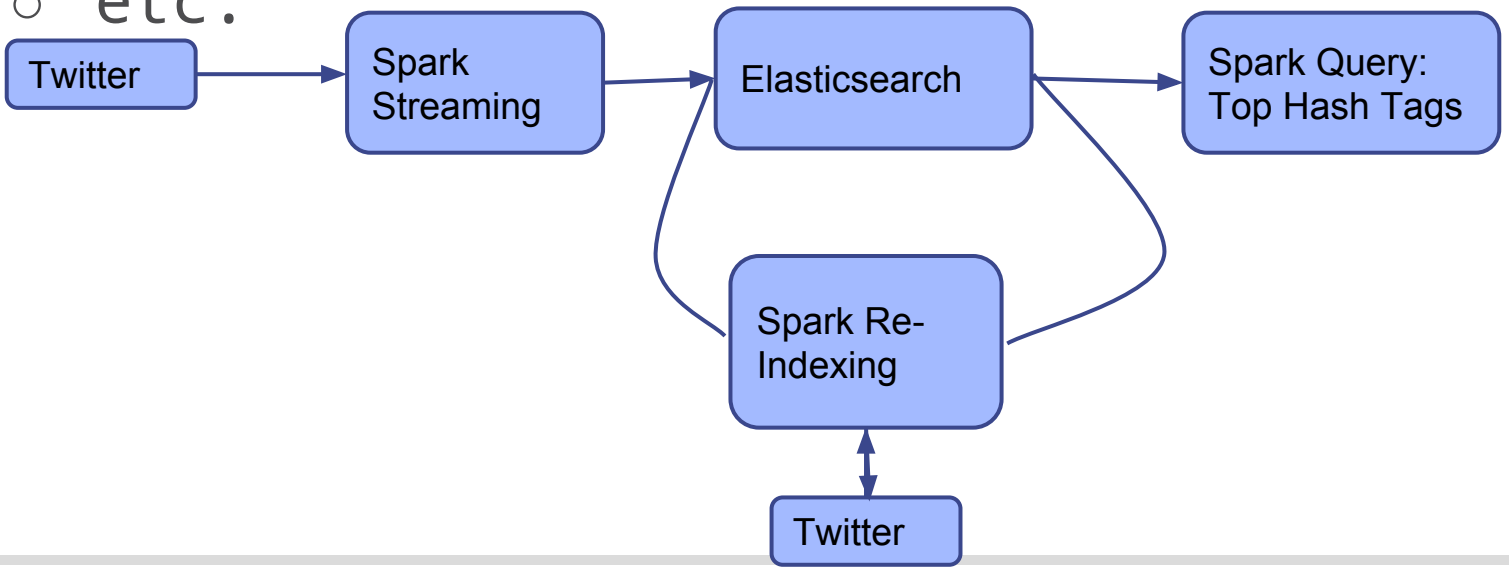
Things you don't have to worry about:

- All the code is on-line, so don't worry if you miss some

*If we have extra time at the end

Spark + Elasticsearch

- We can index our data on-line & off-line
- Gain the power to query our data
  - based on location
  - free text search
  - etc.



Why should you care?

Small differences between off-line and on-line

Spot the difference picture from http://en.wikipedia.org/wiki/Spot_the_difference#mediaviewer/File:Spot_the_difference.png

Cat picture from http://galato901.deviantart.com/art/Cat-on-Work-Break-173043455

Lets start with the on-line pipeline

```
val ssc = new StreamingContext(master, "IndexTweetsLive",
Seconds(1))

val tweets = TwitterUtils.createStream(ssc, None)
```

Lets get ready to write the data into Elasticsearch

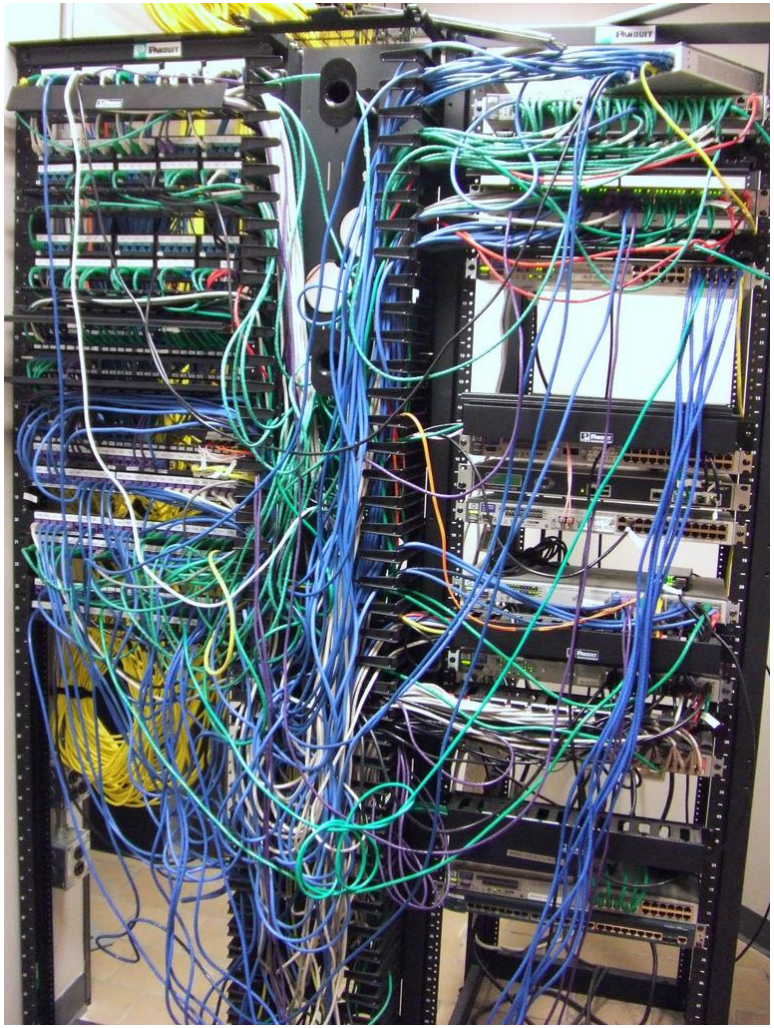


Photo by Cloned Milkmen

Lets get ready to write the data into Elasticsearch

```
def setupEsOnSparkContext(sc: SparkContext) = {
val jobConf = new JobConf(sc.hadoopConfiguration)

jobConf.set("mapred.output.format.class",
"org.elasticsearch.hadoop.mr.EsOutputFormat")

jobConf.setOutputCommitter(classOf[FileOutputCommitter])

jobConf.set(ConfigurationOptions.ES_RESOURCE_WRITE,
“twitter/tweet”)

FileOutputFormat.setOutputPath(jobConf, new Path("-"))
jobconf
}
```

Add a schema

```
curl -XPUT 'http://localhost:
9200/twitter/tweet/_mapping' -d '
{
    "tweet" : {
        "properties" : {
            "message" : {"type" : "string"},
            "hashTags" : {"type" : "string"},
            "location" : {"type" : "geo_point"}
        }
    }
}
'
```

Lets format our tweets

```
def prepareTweets(tweet: twitter4j.Status) = {
  …
  val hashTags = tweet.getHashtagEntities().map(_.getText())
  HashMap(
    "docid" -> tweet.getId().toString,
    "message" -> tweet.getText(),
    "hashTags" -> hashTags.mkString(" "),
    "location" -> s"$lat,$lon"
        )
      }
    }
  // Convert to HadoopWritable types
  mapToOutput(fields)
}
```

And save them...

```
tweets.foreachRDD{(tweetRDD, time) =>
  val sc = tweetRDD.context
  // The jobConf isn't serilizable so we create it here
  val jobConf = SharedESConfig.setupEsOnSparkContext(sc,
                esResource, Some(esNodes))
  // Convert our tweets to something that can be indexed
  val tweetsAsMap = tweetRDD.map(
                    SharedIndex.prepareTweets)
  tweetsAsMap.saveAsHadoopDataset(jobConf)
}
```

Now let's query them!

```
{"filtered" : {
  "query" : {
    "match_all" : {}
  }
  ,"filter" :
    {"geo_distance" :
      {
        "distance" : "${dist}km",
        "location" :
        {
          "lat" : "${lat}",
          "lon" : "${lon}"
}}}}}
```

Now let's find the hash tags :)

```
// Set our query
jobConf.set("es.query", query)

// Create an RDD of the tweets
val currentTweets = sc.hadoopRDD(jobConf,
  classOf[EsInputFormat[Object, MapWritable]],
  classOf[Object], classOf[MapWritable])

// Convert to a format we can work with
val tweets = currentTweets.map{ case (key, value) =>
  SharedIndex.mapWritableToInput(value) }

// Extract the hashtags
val hashTags = tweets.flatMap{t =>
  t.getOrElse("hashTags", "").split(" ")
}
```

and extract the top hashtags

```
object WordCountOrdering extends Ordering[(String, Int)]{
  def compare(a: (String, Int), b: (String, Int)) = {
    b._2 compare a._2
  }
}

val ht = hashtags.map(x => (x, 1)).reduceByKey((x,y) => x+y)

val topTags = ht.takeOrdered(40)(WordCountOrdering)
```

NYC

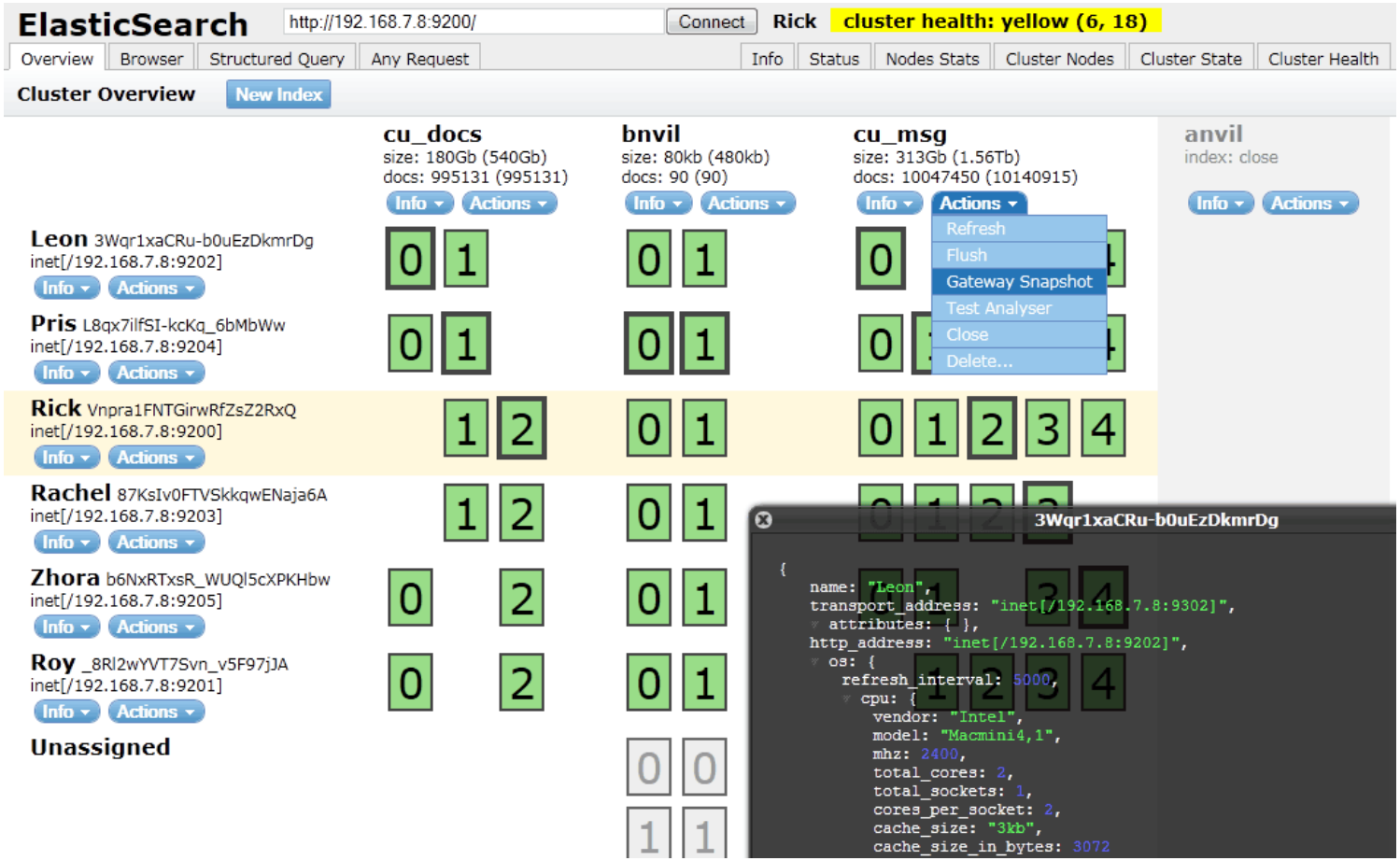
#MEX,11
#Job,11
#Jobs,8
#nyc,7
#CarterFollowMe,7
#Mexico,6
#BRA,6
#selfie,5
#TweetMyJobs,5
#LHHATL,5
#NYC,5
#ETnow,4
#TeenWolf,4
#CRO,4

SF

#Job,6
#Jobs,5
#MEX,4
#TweetMyJobs,3
#TeenWolfSeason4,2
#CRO,2
#Roseville,2
#Healthcare,2
#GOT7COMEBACK,2
#autodeskinterns,2

Indexing Part 2

(electric boogaloo)

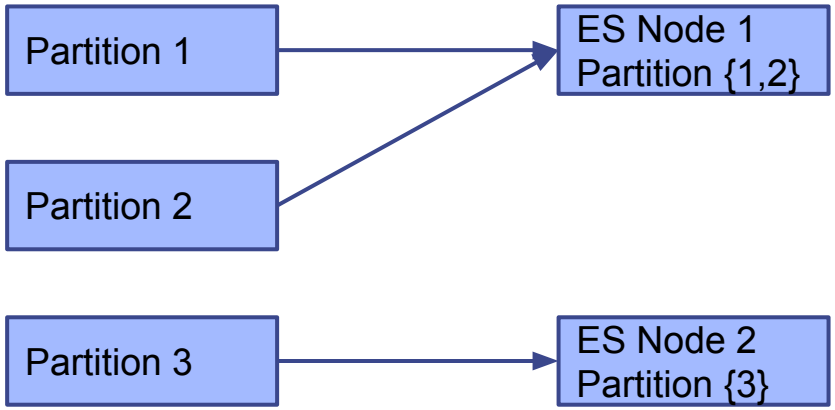


Writing directly to a node with the correct shard saves us network overhead

Screen shot of elasticsearch-head http://mobz.github.io/elasticsearch-head/

So what does that give us?

Spark sets the filename to part-[partition number]
We can modify the connector to parse this! :D
If we have same partitioner we write directly
Likely the best place to use this is in re-indexing data



Re-index all the things*

```
// Fetch them from twitter
val t4jt = tweets.flatMap{ tweet =>
  val twitter = TwitterFactory.getSingleton()
  val tweetID = tweet.getOrElse("docid", "")
  Option(twitter.showStatus(tweetID.toLong))
}

t4jt.map(SharedIndex.prepareTweets)
  .saveAsHadoopDataset(jobConf)
```


*Until you hit your twitter rate limit…. oops

“Useful” links

- Feedback: holden@databricks.com
- Customized ES connector*: https://github.com/holdenk/elasticsearch-hadoop
- Demo code: https://github.com/holdenk/elasticsearchspark
- Elasticsearch: http://www.elasticsearch.org/
- Spark: http://spark.apache.org/
- Spark streaming: http://spark.apache.org/streaming/
- Elasticsearch Spark documentation: http://www.elasticsearch.org/guide/en/elasticsearch/hadoop/current/spark.html

So what did we cover?

- Indexing data with Spark to Elasticsearch
- Sharing indexing code between Spark & Spark Streaming
- Using Elasticsearch for geolocal data in Spark
- Making our indexing aware of Elasticsearch
- Lots* of cat pictures

* There were more before.

Cat photo from https://www.flickr.com/photos/deerwooduk/579761138/in/photolist-4GCc4z-4GCbAV-6Ls27-34evHS-5UBnJv-TeqMG-4iNNn5-4w7s61-6GMLYS-6H5QWY-6aJLUT-tqfrf-6mJ1Lr-84kGX-6mJ1GB-vVqN6-dY8aj5-y3jK-7C7P8Z-azEtd/