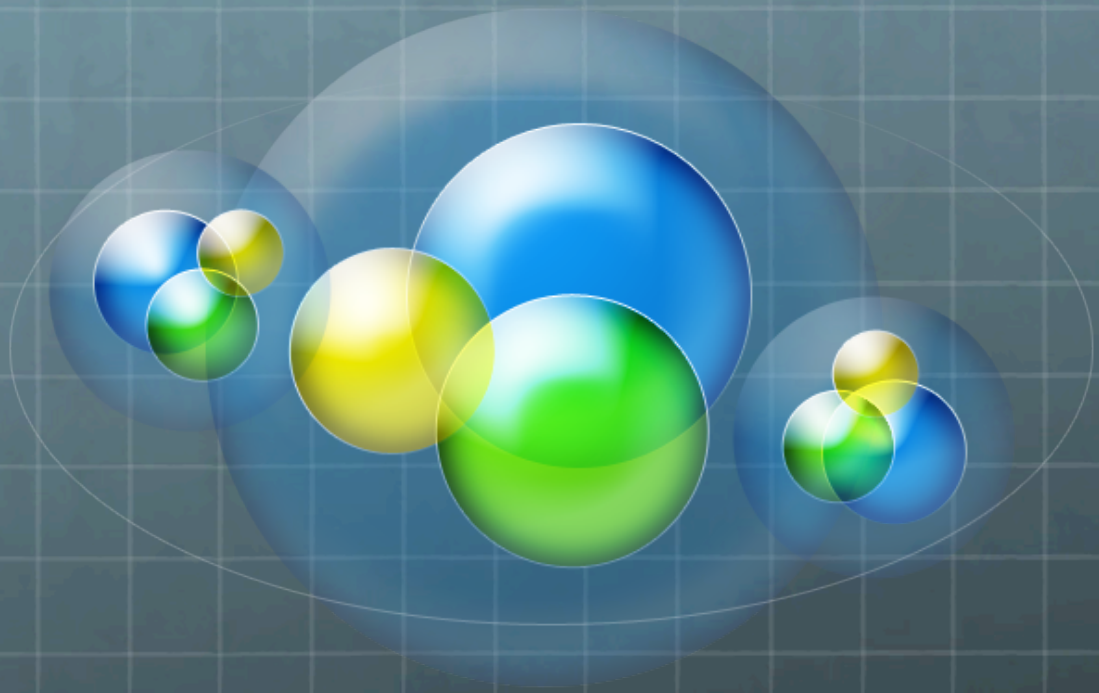


Querying in Druid

Jay Wang
14 April 2013
CPS 394 Research Independent Study

Files Involved

- data.txt
- query.body
- realtime.spec
- start-firehose.sh
- run-queries.sh

data.txt

- **Example in our project:** appevents.txt
- **This is where you specify the structure of your data**
- **You can use a template described here or you can just enter in the raw data you want firehose to push**
- **Sample data stream:**

```
{"appid": 1,"event": "clicked ","event_timestamp": 1366079510443,"country": "CA "}
{"appid": 8,"event": "search ","event_timestamp": 1366079515543,"country": "US"}
{"appid": 4,"event": "like","event_timestamp": 1366079520643,"country": "US"}
{"appid": 1,"event": "search ","event_timestamp": 1366079525743,"country": "US"}
```

query.body

- **Example in our project:** event_counts_query.body
- **This is where you define your query**
- **Here you specify the queryType, dataSource, granularity, dimensions, aggregations, post-aggregations, and intervals**
- **Specific information to be found here**

Trial I

- Firehose has been set to generate @ 10/sec
- Here is what the data set looks like:

```
${name:"sample",type:"combine",endToken:"---",trim:true,separator:""}
{
   "appid": ${type:"minmax", min:1, max:10},
   "event": "${event_name}",
   "event_timestamp": ${type:"time", format:"millis"},
   "country": "${country}"
}
---

${name:"country", type:"weighted",delimiter:","}
80:US
30:CA

${name:"event_name",type:"weighted"}
20:like
25:search
18:clicked
```

Trial I cont.

- Here is what the query looks like:

```
{
    "queryType": "groupBy",
    "dataSource": "appevents",
    "granularity": "all",
    "dimensions": ["appid", "event"],
    "aggregations":[
        {"type":"count", "name":"eventcount"},
        {"type":"doubleSum", "fieldName":"events", "name":"eventssum"}
    ],
    "intervals":["2012-10-01T00:00/2020-01-01T00"]
}
```

- Here is what the schema looks like:

```
[{
  "schema" : { "dataSource":"appevents",
               "aggregators":[ {"type":"count", "name":"events"}
                   ],
               "indexGranularity":"minute"
          },
```

Trial I results

- Here is a sampling of the results:


```
[ {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "1",
    "event" : "clicked ",
    "eventcount" : 4,
    "eventssum" : 9.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "1",
    "event" : "like",
    "eventcount" : 3,
    "eventssum" : 8.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "1",
    "event" : "search ",
    "eventcount" : 3,
    "eventssum" : 11.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "10",
    "event" : "clicked ",
    "eventcount" : 4,
    "eventssum" : 11.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "10",
    "event" : "like",
    "eventcount" : 4,
    "eventssum" : 9.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "10",
    "event" : "search ",
    "eventcount" : 4,
    "eventssum" : 15.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "2",
    "event" : "clicked ",
    "eventcount" : 3,
    "eventssum" : 15.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "2",
    "event" : "like",
    "eventcount" : 3,
    "eventssum" : 6.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "2",
    "event" : "search ",
    "eventcount" : 3,
    "eventssum" : 9.0
  }
}, {
  "version" : "v1",
  "timestamp" : "-292275055-05-16T16:47:04.192Z",
  "event" : {
    "appid" : "3",
    "event" : "clicked ",
    "eventcount" : 2,
    "eventssum" : 5.0
  }
}, {
```

Questions

- **Q: Why will** eventssum **>=** eventcount**?**

  A: It lies in how Druid handles aggregations.

- **In short,** eventssum **is the number of events with the specified dimensions**

- eventcount **is the number of aggregated rows in memory**

- **If Druid gets duplicate data entities, it will aggregate them into one row and assign a count to that row**
  - The count is the number of entities
  - We are using a count aggregator as defined in the schema

Trial II

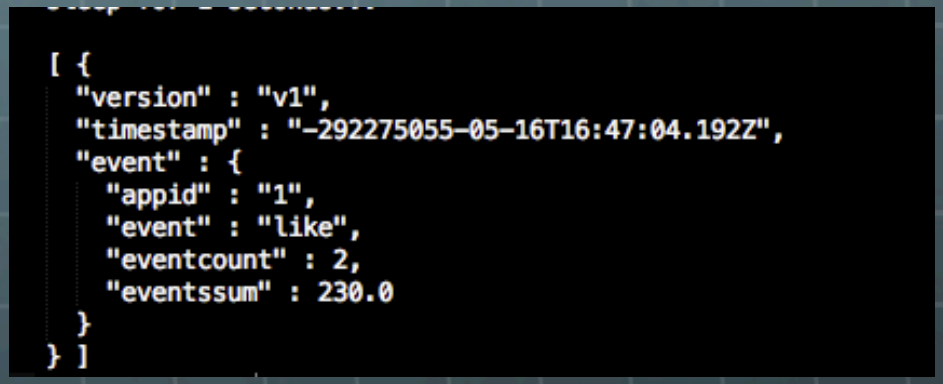
- Objective: Feed Druid the same data to make the eventcount equal 1.
- New data set:

```
${name:"sample",type:"combine",endToken:"---",trim:true,separator:""}
{
   "appid": ${type:"minmax", min:1, max:1},
   "event": "${event_name}",
   "event_timestamp": ${type:"time", format:"millis"},
   "country": "${country}"
}
---

${name:"event_name",type:"weighted"}
20:like

${name:"country", type:"weighted",delimiter:","}
80:US
```

- As can be seen, every event will have an appid of “1”, event of “like” and country of “US”

Trial II results

- The query and schema will remain the same.
- Results:



- Is this what we would expect?
- Why does the eventcount = 2?

Questions

- **The results make sense.**
- **If you go back to the schema for this query, the indexGranularity is set at "minute"**
  - This means at the end of every minute, Druid will perform an aggregation no matter what
  - Ultimately, this means that the timestamps for these data bits span two minutes
- **Note: Druid aggregates on all of the data's fields, NOT on dimensions**
  - If country varied from ["US", "CA", "EN"], Druid would aggregate separately for {appid:1, event:like, country:US} and {appid:1, event:like, country:CA}, etc.

Stress Tests

- **Trying to write a stress test for Druid**
- **"stress" is based on two things:**
  - the amount of data to process over
  - the complexity of the query
- **Problem:**
  - there is a cap as to how fast firehose can push data to kafka (it's generally around <1.5K/sec
  - i.e. you can't feasibly push 1M tweets/sec

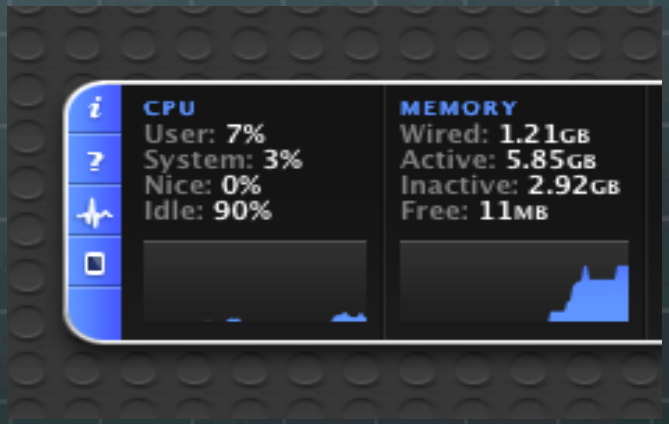
Trial III

- Firehose rate set to 1M/sec (won't actually reach that speed, it peak out much sooner)
- I will let firehose run for 30 min
- Data file is tweets.txt
  - Sample Tweet:


```
{"time_zone":"Alaska","event_timestamp":${type:"time",
format:"millis"},"id":306771720193339393,"id_str":"306771720193339393","text":"Geeee","source":"\u003ca href=\"http:\/\/twitter.
com\/download\/iphone\" rel=\"nofollow\"\u003eTwitter for iPhone\u003c\/a\u003e","truncated":false,"in_reply_to_status_id":null,"in_reply_to_status_id
_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":274990882,"id_str":"274990882","name"
:"Anne Curtis-Smith ","screen_name":"Pawlavableeee","location":"Sa ref","url":"http:\/\/johannapawla.tumblr.com\/","description":"Fil- Aussie
Actress\/Host and Pursuer of dreams","protected":false,"followers_count":423,"friends_count":145,"listed_count":2,"event_timestamp":"Thu Mar 31
12:26:48 +0000 2011","favourites_count":5063,"utc_offset":-32400,"geo_enabled":true,"verified":false,"statuses_count":23038,"lang":"en","contributors_
enabled":false,"is_translator":false,"profile_background_color":"FFFFFF","profile_background_image_url":"http:\/\/a0.twimg.
com\/profile_background_images\/779953914\/44cd05b6ca02282caecf5656ed80dd16.png","profile_background_image_url_https":"https:\/\/si0.twimg.
com\/profile_background_images\/779953914\/44cd05b6ca02282caecf5656ed80dd16.png","profile_background_tile":true,"profile_image_url":"http:\/\/a0.twimg
.com\/profile_images\/3300953457\/fce4b8629cf7c31aa7547c83723f8097_normal.jpeg","profile_image_url_https":"https:\/\/si0.twimg.
com\/profile_images\/3300953457\/fce4b8629cf7c31aa7547c83723f8097_normal.jpeg","profile_banner_url":"https:\/\/si0.twimg.com\/profile_banners\/2749908
82\/1361611207","profile_link_color":"5950FA","profile_sidebar_border_color":"FFFFFF","profile_sidebar_fill_color":"FFFFFF","profile_text_color":"0000
00","profile_use_background_image":true,"default_profile":false,"default_profile_image":false,"following":null,"follow_request_sent":null,"notificatio
ns":null},"geo":{"type":"Point","coordinates":[14.77285192,120.88028117]},"coordinates":{"type":"Point","coordinates":[120.88028117,14.77285192]
},"place":null,"contributors":null,"retweet_count":0,"entities":{"hashtags":[],"urls":[],"user_mentions":[]
},"favorited":false,"retweeted":false,"filter_level":"medium"}
```

Trial III cont.

- Running the same query, except my dimensions will be [“id”, “id_str”, “text”]
- Schema will remain the same
- I ended up running firehose for 15.4 minutes (my computer began running out of memory)



Stress Success

- **Firehose ended up pushing** 1.2M tweets

```
generated 1251192 lines in 923895ms (using the System Clock), 1354/s.
bytes/sec: 3733494, bytes/min: 224009673
```

- **Queries ended up taking almost** 14 seconds

```
real    0m13.941s
user    0m0.006s
sys 0m0.004s
bin/run_client.sh finished
```

- firehose.log **and** queries.log **file available**
- **Conclusion:** definitely possible to “stress druid”

Issues

- Druid has limited ability to query over nested dimensions
- If still running and the data file has finished processing, firehose will start over (should be easy fix)
- No way to pause firehose when data file is finished and then restart firehose when more data is added to data file

Moving Forward

- For these examples, I have used:
  - queryType: groupBy
    - there also exist: search, timeSeries, timeBoundary
  - aggregations: count and doubleSum
    - there also exist: minMax, longSum
- There are other cool features as well all detailed here