Campsite Hot or Not

Blazing-fast weather, powered by Cassandra Charles Chen

Demo

http://campsite-hot-or-not.us/



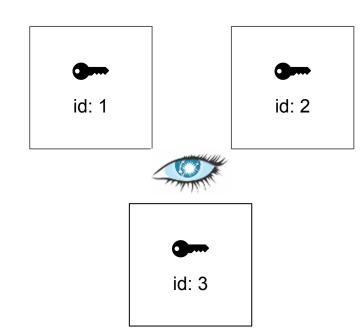


Writing to DB takes more than twice as long as the computation

Cassandra Partition != Spark Partition

Cassandra Partition Key - Part of a row's unique key, determines which partition a row resides on

Cassandra Node - Contains one or more partitions





- Write each row multiple.
 Batch multiple rows together
 Several ways to batch rows Unintuitive batching technique yields best results



id: 1



id: 2





id: 3



{id: 3, timestamp: 1517349935, temp: 25}

{id: 2, timestamp: 1504637523, temp: 40}

{id: 1, timestamp: 1372946023, temp: 30}

{id: 2, timestamp: 1034960223, temp: 25}

{id: 3, timestamp: 1034046294, temp: 30}

{id: 1, timestamp: 1039682384, temp: 25}

{id: 2, timestamp: 1048602453, temp: 40}

{id: 1, timestamp: 1294512359, temp: 2}

{id: 3, timestamp: 1023938102, temp: 5}

{id: 2, timestamp: 1029384642, temp: 30}

{id: 3, timestamp: 1023932852, temp: 23}

{id: 1, timestamp: 1302920492, temp: 33}



id: 1



id: 2





id: 3



{id: 3, timestamp: 1517349935, temp: 25}

{id: 2, timestamp: 1504637523, temp: 40}

{id: 1, timestamp: 1372946023, temp: 30}

{id: 2, timestamp: 1034960223, temp: 25}

{id: 3, timestamp: 1034046294, temp: 30}

{id: 1, timestamp: 1039682384, temp: 25}

{id: 2, timestamp: 1048602453, temp: 40}

{id: 1, timestamp: 1294512359, temp: 2}

{id: 3, timestamp: 1023938102, temp: 5}

{id: 2, timestamp: 1029384642, temp: 30}

{id: 3, timestamp: 1023932852, temp: 23}

{id: 1, timestamp: 1302920492, temp: 33}



id: 1



id: 2





id: 3



{id: 3, timestamp: 1517349935, temp: 25}

{id: 2, timestamp: 1504637523, temp: 40}

{id: 1, timestamp: 1372946023, temp: 30}

{id: 2, timestamp: 1034960223, temp:

{id: 3, timestamp: 1034046294 .cemp: 30}

{id: 1, timestamp: 1039683584, temp: 25}

{id: 2, timestamp: 104 602453, temp: 40}

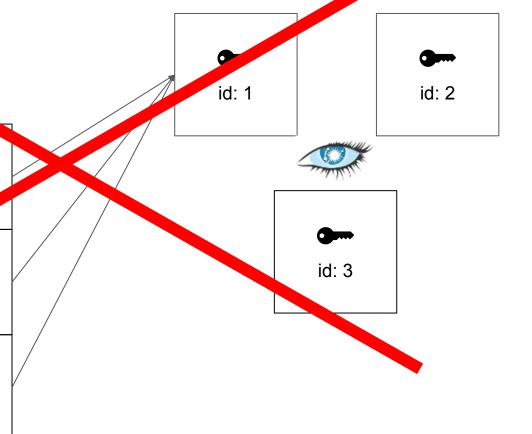
{id: 1, timestamp 1294512359, temp: 2}

{id: 3, tip stamp: 1023938102, temp: 5}

{id: ___timestamp: 1029384642, temp: 30}

{id: 3, timestamp: 1023932852, temp: 23}

{id: 1, timestamp: 1302920492, temp: 33}





{id: 3, timestamp: 1517349935, temp: 25}
{id: 2, timestamp: 1504637523, temp: 40}
{id: 1, timestamp: 1372946023, temp: 30}
{id: 2, timestamp: 1034960223, temp: 25}
{id: 3, timestamp: 1034046294, temp: 30}
{id: 1, timestamp: 1039682384, temp: 25}
{id: 2, timestamp: 1048602453, temp: 40}
{id: 1, timestamp: 1294512359, temp: 2}
{id: 3, timestamp: 1023938102, temp: 5}
{id: 2, timestamp: 1029384642, temp: 30}
{id: 3, timestamp: 1023932852, temp: 23}
{id: 1, timestamp: 1302920492, temp: 33}

{id: 3, timestamp: 1517349935, temp: 25}



{id: 3, timestamp: 1517349935, temp: 25} {id: 2, timestamp: 1504637523, temp: 40} {id: 1, timestamp: 1372946023, temp: 30} {id: 2, timestamp: 1034960223, temp: 25} {id: 3, timestamp: 1034046294, temp: 30} {id: 1, timestamp: 1039682384, temp: 25} {id: 2, timestamp: 1048602453, temp: 40} {id: 1, timestamp: 1294512359, temp: 2} {id: 3, timestamp: 1023938102, temp: 5} {id: 2, timestamp: 1029384642, temp: 30} {id: 3, timestamp: 1023932852, temp: 23} {id: 1, timestamp: 1302920492, temp: 33}

{id: 2, timestamp: 1504637523, temp: 40}
{id: 3, timestamp: 1517349935, temp: 25}



```
{id: 3, timestamp: 1517349935, temp: 25}

{id: 2, timestamp: 1504637523, temp: 40}

{id: 1, timestamp: 1372946023, temp: 30}

{id: 2, timestamp: 1034960223, temp: 25}

{id: 3, timestamp: 1039682384, temp: 25}

{id: 1, timestamp: 1048602453, temp: 25}

{id: 1, timestamp: 1294512359, temp: 2}

{id: 3, timestamp: 1023938102, temp: 5}

{id: 2, timestamp: 1029384642, temp: 30}

{id: 3, timestamp: 1023932852, temp: 23}

{id: 1, timestamp: 1302920492, temp: 33}
```

{id: 1, timestamp: 1372946023, temp: 30} {id: 2, timestamp: 1504637523, temp: 40} {id: 3, timestamp: 1517349935, temp: 25}



```
{id: 3, timestamp: 1517349935, temp: 25} {id: 2, timestamp: 1504637523, temp: 40} {id: 1, timestamp: 1372946023, temp: 30} {id: 2, timestamp: 1034960223, temp: 25} {id: 3, timestamp: 1034046294, temp: 30} {id: 1, timestamp: 1039682384, temp: 25} {id: 2, timestamp: 1048602453, temp: 40} {id: 1, timestamp: 1294512359, temp: 2} {id: 3, timestamp: 1023938102, temp: 5} {id: 2, timestamp: 10239384642, temp: 30} {id: 3, timestamp: 1023932852, temp: 23} {id: 1, timestamp: 1302920492, temp: 33}
```

```
{id: 1, timestamp: 1372946023, temp: 30}
{id: 2, timestamp: 1504637523, temp: 40}
{id: 2, timestamp: 1034960223, temp: 25}
{id: 3, timestamp: 1517349935, temp: 25}
```



```
{id: 3, timestamp: 1517349935, temp: 25} {id: 2, timestamp: 1504637523, temp: 40} {id: 1, timestamp: 1372946023, temp: 30} {id: 2, timestamp: 1034960223, temp: 25} {id: 3, timestamp: 1034046294, temp: 30} {id: 1, timestamp: 1039682384, temp: 25} {id: 2, timestamp: 1048602453, temp: 40} {id: 1, timestamp: 1294512359, temp: 2} {id: 3, timestamp: 1023938102, temp: 5} {id: 2, timestamp: 1029384642, temp: 30} {id: 3, timestamp: 1023932852, temp: 23} {id: 1, timestamp: 1302920492, temp: 33}
```

```
{id: 1, timestamp: 1372946023, temp: 30}
{id: 2, timestamp: 1504637523, temp: 40}
{id: 2, timestamp: 1034960223, temp: 25}
{id: 3, timestamp: 1517349935, temp: 25}
{id: 3, timestamp: 1034046294, temp: 30}
```

```
stations_df = spark.createDataFrame(time_weighted_temp, station_schema)\
    .write\
    .format("org.apache.spark.sql.cassandra")\
    .mode('append')\
    .options(**station_save_options)\
```



```
{id: 3, timestamp: 1517349935, temp: 25} {id: 2, timestamp: 1504637523, temp: 40} {id: 1, timestamp: 1372946023, temp: 30} {id: 2, timestamp: 1034960223, temp: 25} {id: 3, timestamp: 1034046294, temp: 30} {id: 1, timestamp: 1039682384, temp: 25} {id: 2, timestamp: 1048602453, temp: 40} {id: 1, timestamp: 1294512359, temp: 2} {id: 3, timestamp: 1023938102, temp: 5} {id: 2, timestamp: 10239384642, temp: 30} {id: 3, timestamp: 1023932852, temp: 23} {id: 1, timestamp: 1302920492, temp: 33}
```

.save()

```
{id: 1, timestamp: 1372946023, temp: 30}
{id: 1, timestamp: 1039682384, temp: 25}
{id: 1, timestamp: 1294512359, temp: 2}
{id: 1, timestamp: 1302920492, temp: 33}
{id: 2, timestamp: 1504637523, temp: 40}
{id: 2, timestamp: 1034960223, temp: 25}
{id: 2, timestamp: 1048602453, temp: 40}
{id: 2, timestamp: 1029384642, temp: 30}
{id: 3, timestamp: 1517349935, temp: 25}
{id: 3, timestamp: 1034046294, temp: 30}
{id: 3, timestamp: 1023938102, temp: 5}
{id: 3, timestamp: 1023932852, temp: 23}
```

```
stations_df = spark.createDataFrame(time_weighted_temp, station_schema)\
    .sort("id")\
    .write\
    .format("org.apache.spark.sql.cassandra")\
    .mode('append')\
    .options(**station_save_options)\
    .save()
```

```
{id: 3, timestamp: 1517349935, temp: 25}
                                                             {id: 1, timestamp: 1372946023, temp: 30}
{id: 2, timestamp: 1504637523, temp: 40}
                                                             {id: 1, timestamp: 1039682384, temp: 25}
{id: 1, timestamp: 1372946023, temp: 30}
                                                             {id: 1, timestamp: 1294512359, temp: 2}
{id: 2, timestamp: 1034960223, temp: 25}
                                                             {id: 1, timestamp: 1302920492, temp: 33}
{id: 3, timestamp: 1034046294, temp: 30}
                                                             {id: 2, timestamp: 1504637523, temp: 40}
{id: 1, timestamp: 1039682384, temp: 25}
                                                             {id: 2, timestamp: 1034960223, temp: 25}
{id: 2, timestamp: 1048602453, temp: 40}
                                                             {id: 2, timestamp: 1048602453, temp: 40}
{id: 1, timestamp: 1294512359, temp: 2}
                                                             {id: 2, timestamp: 1029384642, temp: 30}
{id: 3, timestamp: 1023938102, temp: 5}
                                                             {id: 3, timestamp: 1517349935, temp: 25}
{id: 2, timestamp: 1029384642, temp: 30}
                                                             {id: 3, timestamp: 1034046294, temp: 30}
{id: 3, timestamp: 1023932852, temp: 23}
                                                             {id: 3, timestamp: 1023938102, temp: 5}
{id: 1, timestamp: 1302920492, temp: 33}
                                                             {id: 3, timestamp: 1023932852, temp: 23}
```



{id: 1, timestamp: 1372946023, temp: 30} {id: 1, timestamp: 1039682384, temp: 25} {id: 1, timestamp: 1294512359, temp: 2} {id: 1, timestamp: 1302920492, temp: 33} {id: 2, timestamp: 1504637523, temp: 40} {id: 2, timestamp: 1034960223, temp: 25} {id: 2, timestamp: 1048602453, temp: 40} {id: 2, timestamp: 1029384642, temp: 30} {id: 3, timestamp: 1517349935, temp: 25} {id: 3, timestamp: 1034046294, temp: 30} {id: 3, timestamp: 1023938102, temp: 5} {id: 3, timestamp: 1023932852, temp: 23}





id: 1



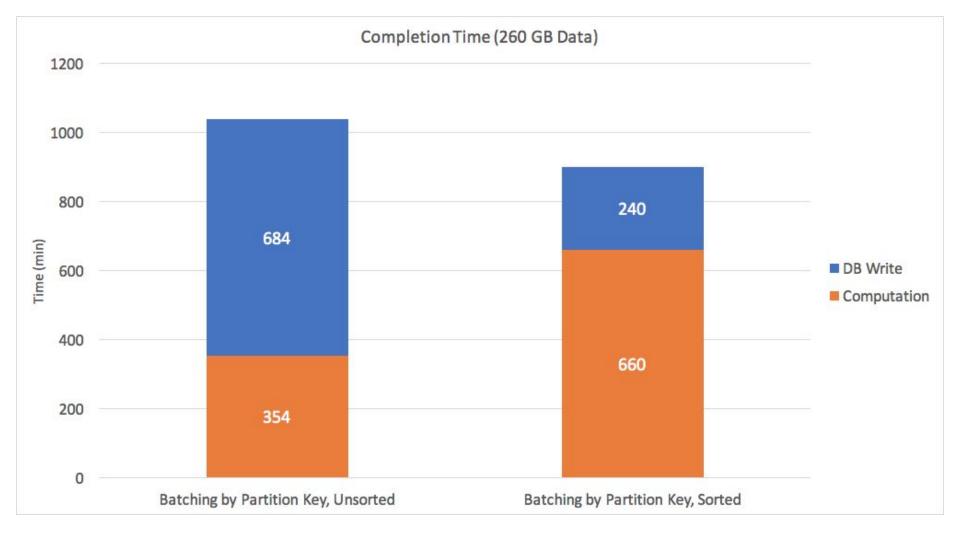
id: 2



id: 3

{id: 1, timestamp: 1372946023, temp: 30} {id: 1, timestamp: 1039682384, temp: 25} {id: 1, timestamp: 1294512359, temp: 2} {id: 1, timestamp: 1302920492, temp: 33} {id: 2, timestamp: 1504637523, temp: 40} {id: 2, timestamp: 1034960223, temp: 25} {id: 2, timestamp: 1048602453, temp: 40} {id: 2, timestamp: 1029384642, temp: 30} {id: 3, timestamp: 1517349935, temp: 25} {id: 3, timestamp: 1034046294, temp: 30} {id: 3, timestamp: 1023938102, temp: 5} {id: 3, timestamp: 1023932852, temp: 23}





Implement change to spark-cassandra-connector

- Looked at DataStax's JIRA, open issue regarding this
- Could add Catalyst rule to automatically sort on partition key prior to save
- Engaging with Russell Spitzer on DataStax Slack
 - DataStax employee, Spark contributor, speaker at Spark Summit

About Me - Charles Chen

B.S./M.S. Aerospace Engineering, USC

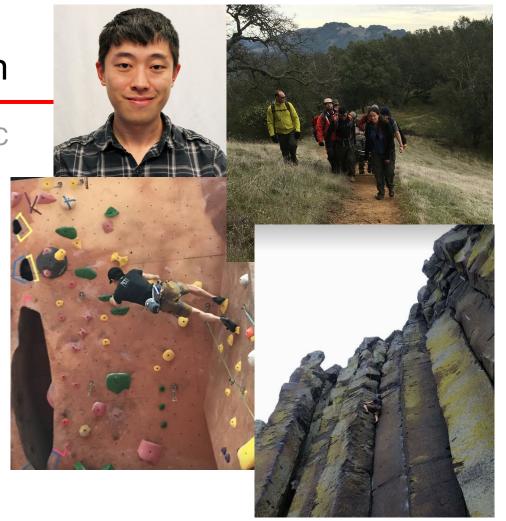
B.S. Computer Science, Oregon State

Aerodynamics Engineer @ Boeing

linkedin.com/in/charles-l-chen

github.com/CCInCharge

Interests: Rock Climbing, Hiking, Backpacking, Search and Rescue Volunteer



Q & A

Backup

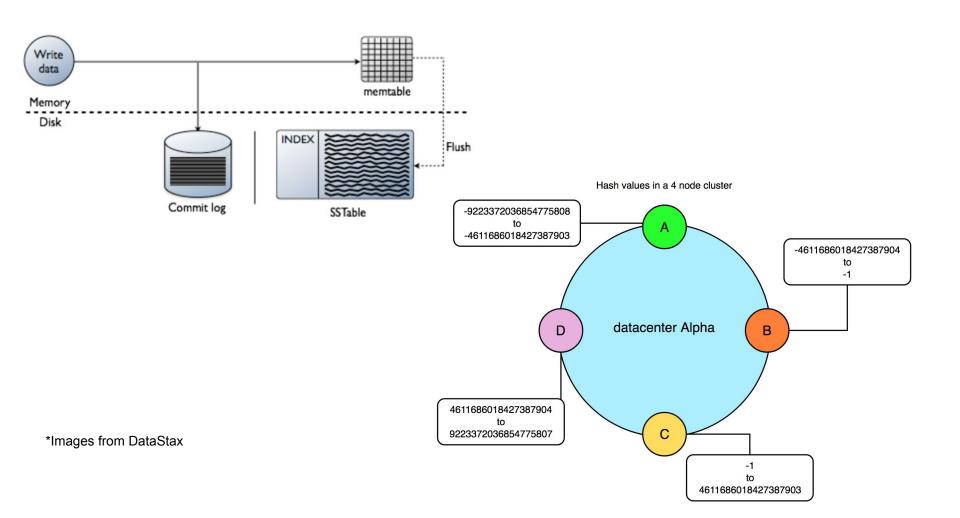
Algorithm

Inverse Weighted Average

$$u(\mathbf{x}) = egin{cases} rac{\sum_{i=1}^N w_i(\mathbf{x}) u_i}{\sum_{i=1}^N w_i(\mathbf{x})}, & ext{if } d(\mathbf{x}, \mathbf{x}_i)
eq 0 ext{ for all } i, \ u_i, & ext{if } d(\mathbf{x}, \mathbf{x}_i) = 0 ext{ for some } i, \end{cases}$$

where

$$w_i(\mathbf{x}) = rac{1}{d(\mathbf{x},\mathbf{x}_i)^p}$$











express

Implementation of distributed weighting algorithm

Storage of time-series data

Implementation Details for Connector Patch

- Create a Catalyst rule to automatically sort the DataFrame on partition key,
 prior to saving to database
 - Catalyst is a very low-level API

Motivation

Identify historical weather patterns at campsites

Consumer interest:

Identify good places to go camping, based on weather patterns

Trail impact:

100% increase in permit applications for JMT from 2011 - 2015