Cooking.com Recipe Big Data Pipeline

Abhinav Dhiman

-Data Engineer

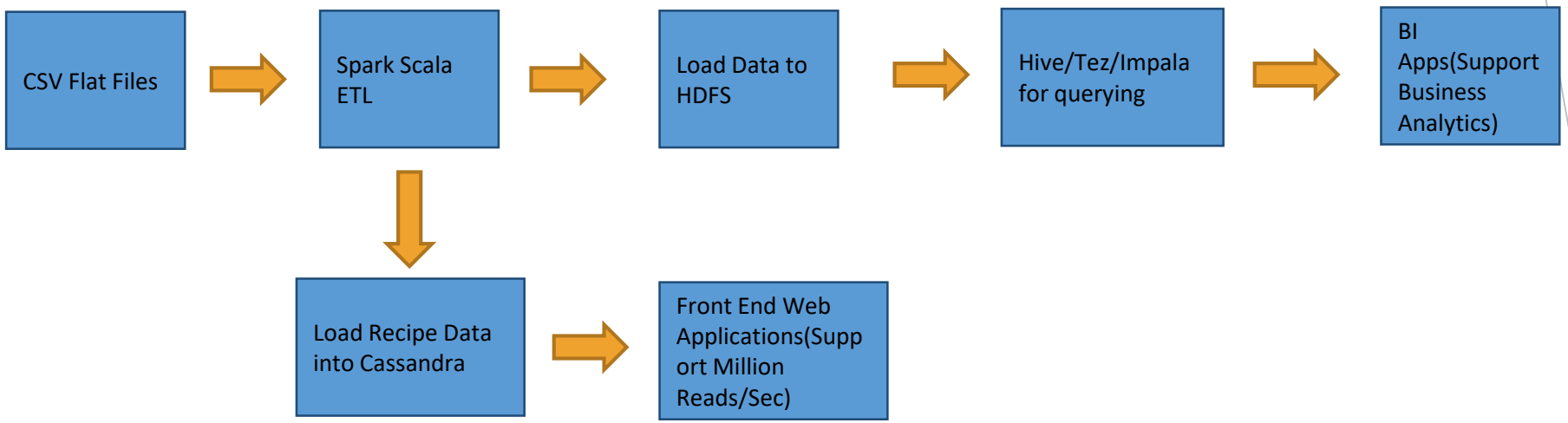
Goals

- Build a reliable big data pipeline to load the cooking.com recipe data from csv so that front end applications can consume recipe data and business can run analytics from this data
- Front End application needs support for million reads/sec
- Present a visualization for how this data pipeline can serve business and front end applications

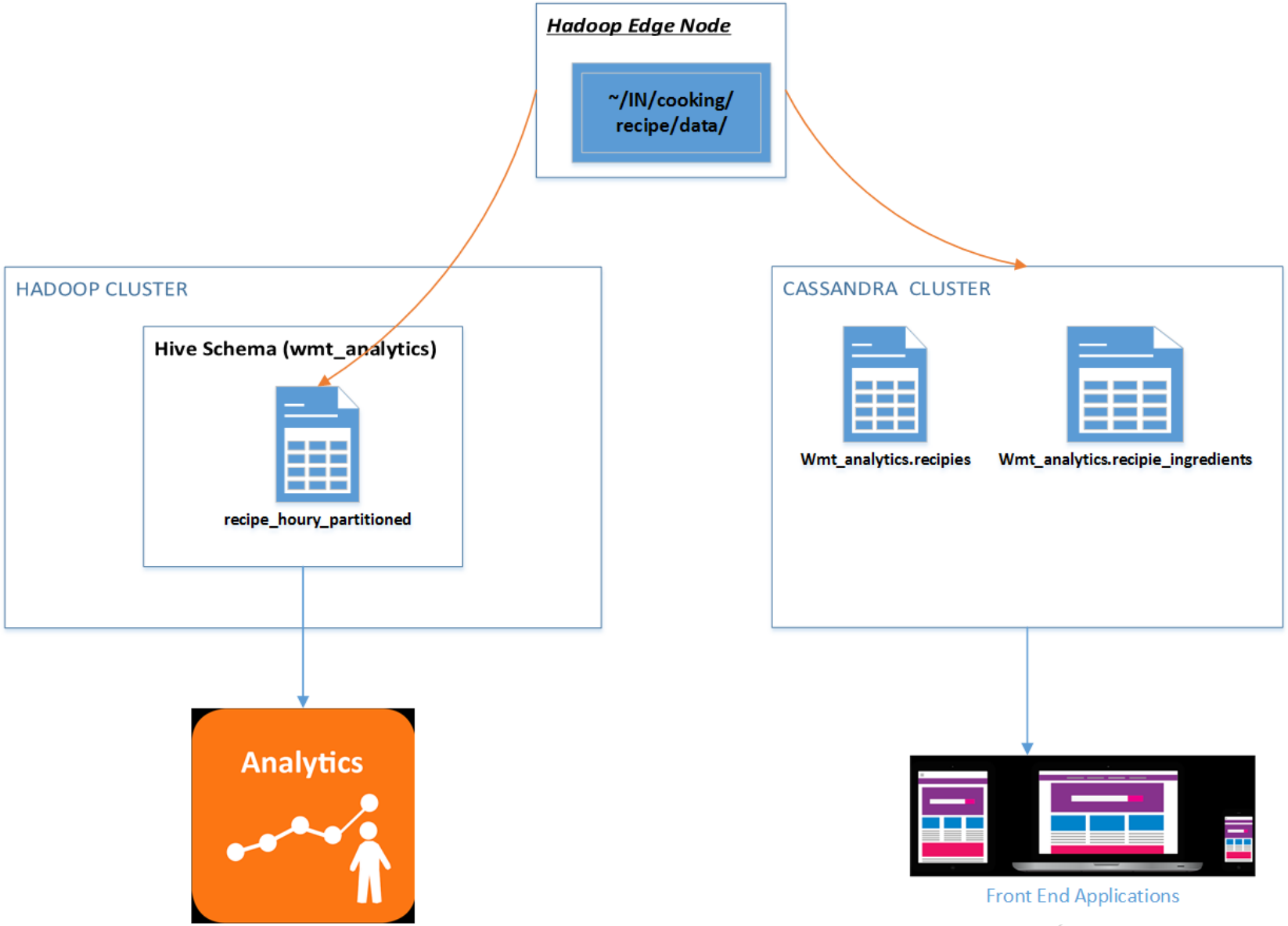
Technology Stack Used

- Cassandra 3.11
- Scala 2.11.8
- Apache Spark 2.3.0
- Apache Hive
- HDFS
- Cloudera CDH 5.12

Logical Data Pipeline Model



Detail Design



Assumptions And Implementation Steps

- Assumptions :
  - Recipe files are dropped on the hadoop edge node at location ~/IN/cooking/recipe/data/
  - Each hourly file contains only incremental load information. The process expects one file on hourly basis
  - File format and naming check is not performed for this implementation and assumed to be correct. Hourly files format is YYYY-MM-DD_HH_recipe_data_raw.csv
  - Hourly files contain latest snapshot of recipe and combination of recipe_name+ingredient is unique
- Step 1 : Spark Scala job to pull the files from edge node and push it into hive table wmt_analytics.recipe_hourly. Each file after successful loading will go to archive location
- Step 2: Spark aggregation runs on the data from csv file. The spark process find all recipe information and also recipe to ingredients association information into two separate dataframes
- Step 3: Spark writes above two dataframe into Cassandra database
- Step 4: Real time front end apps will consume data from Cassandra and BI analytics queries will be performed against data in hive using any kind of BI tool
- Step 5: This pipeline have the ability to process multiple files in case of failure

Challenges

- Cassandra to Spark integration
- Cassandra setup
- Performance Issues

Data Visualization

Cassandra Tables for Front end apps

```
(4 rows)
cqlsh> select * from wmt_analytics.recipes;

 recipe_name
----------------------
               pasta
 chicken tikka masala
       butter chicken
              lasagna

(4 rows)
cqlsh> select * from wmt_analytics.recipes;

 recipe_name
----------------------
               pasta
 chicken tikka masala
       butter chicken
              lasagna

(4 rows)
cqlsh> select * from wmt_analytics.recipe_ingredients;

 recipe_name          | ingredient
----------------------+-----------------
               pasta |    tomato sauce
 chicken tikka masala | roasted chicken
       butter chicken |    tomato sauce
              lasagna |     blue cheese

(4 rows)
cqlsh> select * from wmt_analytics.recipe_ingredients;

 recipe_name          | ingredient
----------------------+-----------------
               pasta |    tomato sauce
 chicken tikka masala | roasted chicken
       butter chicken |    tomato sauce
              lasagna |     blue cheese
```

Data Visualization

Hive Table for Business Apps

```
Time taken: 30.473 seconds, Fetched: 4 row(s)
hive> show partitions wmt_analytics.recipe_hourly;
OK
source_file_date=2018-01-01/hour=10
source_file_date=2018-01-09/hour=10
source_file_date=2018-01-09/hour=11
Time taken: 0.14 seconds, Fetched: 3 row(s)
hive> select * from wmt_analytics.recipe_hourly;
OK
1       pasta   Italian pasta   tomato sauce    true    2018-01-09 11:00:57     2018-01-10 13:00:57     2018-01-01      1
0
3       butter chicken  indian style butter chicken     tomato sauce    true    2018-01-09 11:10:57     2018-01-11 11:00
:57     2018-01-01      10
4       chicken tikka masala    british style tikka masala      roasted chicken true    2018-01-09 11:00:57     2018-01-
10 13:00:57     2018-01-01      10
4       lasagna layered lasagna blue cheese     true    2018-01-09 11:00:57     2018-01-10 13:00:57     2018-01-01      1
0
1        pasta   Italian pasta   tomato sauce   NULL    2018-01-09 10:00:57     2018-01-10 13:00:57     2018-01-09      1
0
1        pasta   null    cheese NULL    2018-01-09 10:10:57     2018-01-10 13:00:57     2018-01-09      10
2        lasagna         layered lasagna         cheese NULL    2018-01-09 10:00:57     2018-01-10 13:00:57     2018-01-
09      10
2        lasagna         layered lasagna         blue cheese    NULL    2018-01-09 10:00:57     2018-01-10 13:00:57     2
018-01-09       10
1       pasta   Italian pasta   tomato sauce    true    2018-01-09 11:00:57     2018-01-10 13:00:57     2018-01-09      1
1
3       butter chicken  indian style butter chicken     tomato sauce    true    2018-01-09 11:10:57     2018-01-11 11:00
:57     2018-01-09      11
4       chicken tikka masala    british style tikka masala      roasted chicken true    2018-01-09 11:00:57     2018-01-
10 13:00:57     2018-01-09      11
4       lasagna layered lasagna blue cheese     true    2018-01-09 11:00:57     2018-01-10 13:00:57     2018-01-09      1
1
Time taken: 0.153 seconds, Fetched: 12 row(s)
```

Data Visualization

Average number of recipes which are updated per hour

```
Time taken: 35.995 seconds, Fetched: 4 row(s)
hive> select recipe_name,hour,count(*) from wmt_analytics.recipe_hourly group by recipe_name,hour;
Query ID = cloudera_20180521091414_6dd6adaf-5620-419f-9ffd-fcb93acc7177
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1520459735422_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1520459735422_00
02/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1520459735422_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-05-21 09:14:28,057 Stage-1 map = 0%,  reduce = 0%
2018-05-21 09:14:35,863 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.56 sec
2018-05-21 09:14:45,831 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.86 sec
MapReduce Total cumulative CPU time: 3 seconds 860 msec
Ended Job = job_1520459735422_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.86 sec   HDFS Read: 10352 HDFS Write: 70 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 860 msec
OK
butter chicken  10      1
chicken tikka masala    10      1
lasagna 10      1
pasta   10      1
Time taken: 30.473 seconds, Fetched: 4 row(s)
```

Data Visualization

Number of recipes which got updated at 10:00 clock in the entire year.

```
hive> select recipe_name
    > ,count(*) as NumberOfTimesUpdated
    > from wmt_analytics.recipe_hourly
    > where hour=10
    > and to_date(update_date)>= add_months(current_date,-12)
    > group by recipe_name;
Query ID = cloudera_20180521115050_b33137fd-a398-4b2f-88fa-5b34353b48ce
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1520459735422_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1520459735422_00
04/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1520459735422_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-05-21 11:50:58,097 Stage-1 map = 0%,  reduce = 0%
2018-05-21 11:51:12,836 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.96 sec
2018-05-21 11:51:29,345 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.63 sec
MapReduce Total cumulative CPU time: 7 seconds 630 msec
Ended Job = job_1520459735422_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.63 sec   HDFS Read: 13047 HDFS Write: 78 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 630 msec
OK
 lasagna        2
 pasta  2
butter chicken  1
chicken tikka masala    1
lasagna 1
pasta   1
```

Future Enhancements

- Add unit tests
- Add load control framework integration for lineage information
- Code documentation
- Performance Improvements
- Miscellaneous code enhancements

Current Status + Demo

- The code was completed is now been committed to GitHub
- https://github.com/dhimanabhinav87/wmt_analytics
- Readme.md explains all the different files associated with project
- The data pipeline was tested on Cloudera hadoop VM and produced expected results

Questions ?