Code Review: a8-manthan

Bhanu

November 16, 2017

Details

Commit Info

The code is committed before deadline. Commit cut-off is `78098377631627799d8f6dede277c616efcaf489`

Review

a8-manthan

Reviewer

Bhanu

Execution

- Code runs as mentioned in the `readme` and `Makefile`

Report review

- Report is well organized with great visualizations and analysis.
- Implementations details could have been explained more. There is no mention on the work split between hadoop and R. It would helpful for the reader to understand what operations are performed in which framework.
- It might be good to talk about data, your sanity rules and the thought process behind that. Right now I don't know why you did on what you did. I understand there limited space in the report, but as you have space remaining, including details about data and sanity would make this report even better.
- You could add more concrete findings from your analysis in the Conclusion and ignore comparisons with your previous implementation. These comparisons are already included in `diff.pdf`

Code Review

- Hadoop code is well written and precise.
- (Minor) Comments not up to date. `DelayJobDriver.java` and `DelayMapper.java` talk about `CleanDataWritables`, but there is no such class. Please make sure you have most latest code comments as it creates a lot of confusion.
- `File: a8-manthan/src/jobs/DelayJobDriver.java`

```
52:             job.waitForCompuetion(true);
53:             if(job.isSuccessful()) return 0;
54:             return 1;
```

L53-54 can be avoided as `job.waitForCompletion` returns a status(Boolean). Instead use `return job.waitForCompletion(true) ? 0 : 1;`

- `File: a8-manthan/src/jobs/DelayJobDriver.java`
  ```
  50:             FileInputFormat.setInputDirRecursive(job, true);
  51:             FileOutputFormat.setOutputPath(job, outputPath);
  ```
  You only need L50:`FileInputFormat.setInputDirRecursive(job, true)` when your input files are more than one level inside the given input path. For files `input\<files>`, you can ignore `FileInputFormat.setInputDirRecursive` with input path as `input`

- `File: a8-manthan/src/mappers/DelayMapper.java`
  ```
  29:                 DelayWritable outValue = new DelayWritable();
  30:                 outValue.setDelay(new DoubleWritable(p.getArrDelNew()));
  31:                 outValue.setCount(new LongWritable(1));
  ```
  Use a constructor instead of individually setting values to `DelayWritable` obj. The same thing can be replaced in the reducer.

- I don't think you are using these files any more. You can remove them if no longer needed.

  `a8-manthan/output/*`

- Most of the analysis is moved to R, which is a good thing for the size of data we are dealing in R. The only thing missing is the documentation of R code. It is very difficult to understand your analysis is R.

- In R, not sure where these data files came from.

  ```
  data2 = read.csv("resources/dest_to_full_name.csv", header=T)
  coords = read.csv("resources/coordinates.csv", header=T)
  names(coords) = c("airport", "Latitude", "Longitude")
  ```

  Probably these were manually generated. Do we really need these? Can we auto-generate them through hadoop or java ?

Final Thoughts

- Overall approach is very clean and to the point.
- Great improvement from previous attempt with 3 hadoop jobs reduced 1.
- Report is nicely written and visualizations are well thought of.