Column Statistics in Hive (WIP)
Shreepadma Venugopalan
Software Engineer, Platform Team
Cloudera
Agenda

- Motivation
  - Why Column Statistics?
- New Statistics
- Computing and Persisting Statistics
  - How to Compute Column Statistics?
  - How to Persist Column Statistics?
- Open Issues
- Summary
- Further Readings
Why Column Statistics?

- Current State of Statistics in Hive
  - Number of rows, size of data etc. on table and partition level
  - Useful in determining the size of inputs to a Join operator
  - Insufficient for implementing a full fledged cost based optimizer
Why Column Statistics?

- **Solution:** Statistics on column level
  - Needed for implementing a cost based optimizer, query progress indicator
  - Useful for implementing Theta Joins (Natural Join) as well!
What are the New Statistics?

- Min Value
- Max Value
- Average Length
- Max Length
- Number of Distinct Values
- Number of Null Values
- Equi-depth Histograms
How to Compute Column Statistics?

- Stats Computation
  - Algorithms follow two phases – collect (Map) and aggregate (Reduce)
  - Requirements: Memory required should scale sub-linearly (preferably logarithmically) with the size of data
  - Problem: Not all statistics are trivial to compute!
    - Number of distinct values
    - Equi-depth Histograms
Hard to Compute Statistics - Example

• Number of Distinct Values
  – Naïve approach: Keep track of all distinct values in column; Impractical to keep in memory given the size of data
  – Flajolet-Martin approach: Use hash functions to estimate the number of distinct values; Memory required is only logarithmic in size of data
How to Compute Column Statistics?

- Stats Computation (contd..)
  - Implemented using Generic UDAF framework
  - Integrate into Hive using new StatsCollector and StatsAggregation Operators

- Explicit Computation
  - Triggered through an explicit “analyze” command

- Implicit Computation (Optional)
  - Incrementally compute and maintain statistics automatically while loading data
How to Persist Column Statistics?

- Use Metastore
  - Extend schema to persist new statistics
  - Provide new Thrift API to retrieve new statistics
Open Issues

• How to aggregate equi-depth histograms constructed by the map tasks?
• Can we improve the estimates of the number of distinct values without increasing the memory footprint?
Summary

• Tracked by JIRA - HIVE-1362
• So far..😊
  – UDAF{s for computing all statistics except histograms
• ToDos 😞
  – Equi-depth histograms
  – Metastore and Thrift API changes to persist and retrieve statistics
  – Integrate with analyze command in Hive
Questions?
Further Reading

- **Learn**
  - **Academic**
    - A. Gruenheid, et. al., Query Optimization using Column Statistics in Hive.
  - Blog: [http://www.cloudera.com/blog](http://www.cloudera.com/blog)

- **Download ’n’ play**
  - [www.cloudera.com/download](http://www.cloudera.com/download)

- **Contribute to the community**
  - [hadoop.apache.org](http://hadoop.apache.org)

- **Expertise input**
  - Join [cdh-user@cloudera.org](mailto:cdh-user@cloudera.org)

Contact me:
shreepadma@cloudera.com