Hortonworks

High Volume Updates in Hive

Owen O'Malley

owen@hortonworks.com

@owen_omalley

June 2012

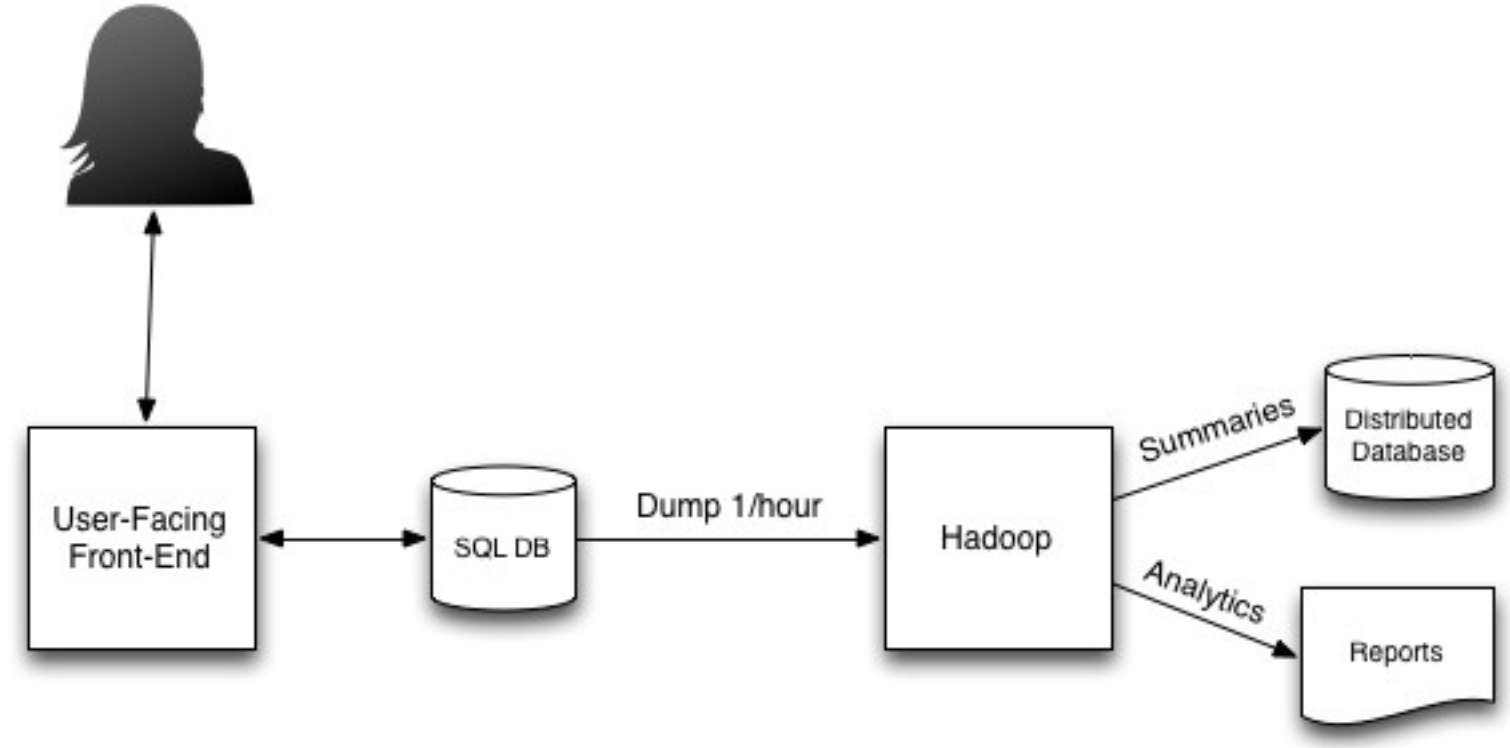
Who Am I?

- Was working in Yahoo Search in Jan 2005, when we decided to work on Hadoop.
- My first patch went in before it was Hadoop.
- Was the first committer added to the project.
- Was the first Apache VP of Hadoop.
- Won the sort benchmark in 2008 & 2009.
- Was the tech lead for:
  - MapReduce
  - Security
- Now I'm working on Hive

A Data Flood

- The customer has:
  - Business critical
  - 1 TB/day of customer data
  - Need to keep all historical data
  - Data size increasing exponentially
  - Doubles year over year
  - Lots of derived reports
  - 15% of traffic is updates to old data

The Dataflow

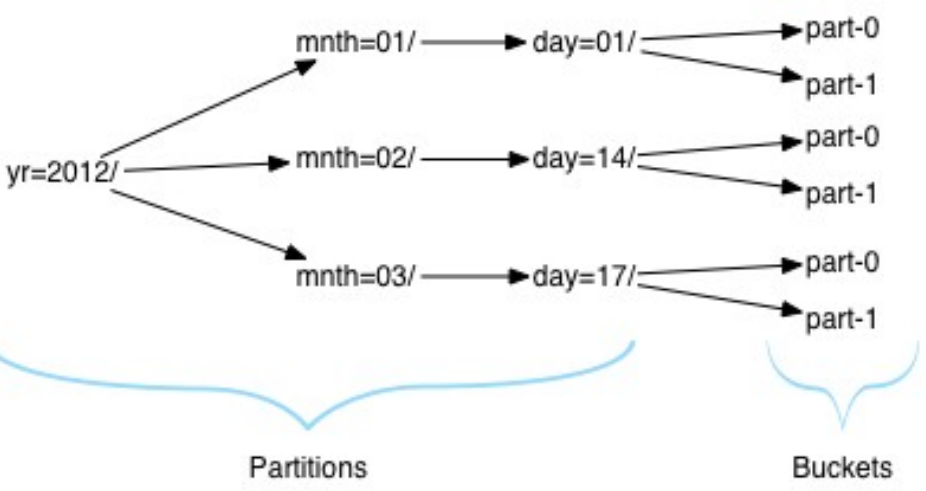


The Approach

- Picked Hive
  - Wanted a higher-level query language
  - Hive's HQL lowers learning curve
  - But, they have those edits & deletes…
- Hive doesn't update things well
  - Reflects the capabilities of HDFS
  - Can add records
  - Can overwrite partitions

Hive Table Layout

CreateDate	UniqueId	Name	Purchase
2012/01/01	1	Anne	Red Fish
2012/01/01	2	Bill	Blue Fish
2012/02/14	3	Christine	Blue Fish
2012/03/17	4	David	Black Fish
2012/03/17	5	Bill	Young Fish



1	Anne	Red Fish
2	Bill	Blue Fish
3	Christine	Blue Fish
4	David	Black Fish
5	Bill	Young Fish

RCFiles

Design

- Hive allows tables to specify an InputFormat and OutputFormat
- We can store each bucket as a base file and a sequential set of edit files.
- To enable efficient reading of a bucket, the base and edits files must be sorted.
- Tag each edit file with a “timestamp” to support repeatable reads.
- Edits must be compacted eventually.

Stitching Buckets Together

Base File

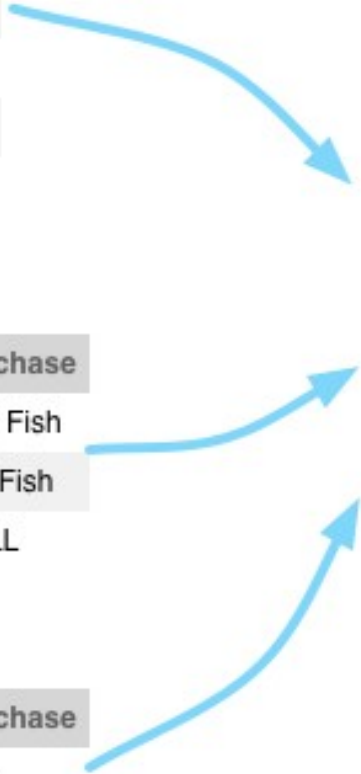
UniqueId	Name	Purchase
1	Anne	Red Fish
2	Bill	Blue Fish
25	Christine	Blue Fish
27	David	Black Fish
30	Bill	Young Fish

Update 1

Op	UniqueId	Name	Purchase
U	1	Ann	Red Fish
I	26	Joe	Old Fish
D	30	NULL	NULL

Update 2

Op	UniqueId	Name	Purchase
U	1	Anne	Star
D	25	NULL	NULL



Squashed File

UniqueId	Name	Purchase
1	Anne	Star
2	Bill	Blue Fish
26	Joe	Old Fish
27	David	Black Fish

Limitations

- Requires active component to manage compaction.
  - Compactions should be scheduled for off-peak hours.
  - Compaction of each partition is independent.
- Requires table be sorted.
- Must use consistent bucketing.

Additional Challenges from Hive

- Hive serializes before the OutputFormat
- Hive defines its own OutputFormat
- Hive forces its own OutputCommitter
- Dynamic Partitions hold open a lot of files
  - Snappy used too much memory
- RCFiles compress well, but no index or bloom filters
- Fixed bucketing across partitions

Why not Hbase?

- Good
  - Handles compaction for us
  - Files are indexed with Bloom filters
- Bad
  - Push down filters only recently done
  - HBase has only a single key
  - Requires key hashing to spread load
  - Tuned for real-time access

A Set Back

- After several months of data loading:
  - Accidental “hadoop fs –rmr”
  - Deleted 450 TB
- Good
  - Shutdown NameNode quickly
  - Copied HDFS fsimage & edits
- Bad
  - Trash wasn’t turned on
  - Brought NameNode back up
  - Didn’t shutdown other servers

Conclusion

- Still in Development
  - The approach is scalable
  - Naturally bumps still hidden in the dark
- Testing on samples as data reloads
- Expect to complete end to end testing in a couple months

Thank You!

Questions & Answers