Mutable Data in Hive's Immutable World

Lester Martin – Hortonworks

2015 Hadoop Summit

Connection before Content

Lester Martin – Hortonworks Professional Services

lmartin@hortonworks.com || lester.martin@gmail.com

http://lester.website *(links to blog, twitter, github, LI, FB, etc)*

“Traditional” Hadoop Data

~~**Time-Series Immutable (TSI) Data**~~ – Hive’s *sweet spot*

Going beyond web logs to more exotic data such as:

Vehicle sensors (ground, air, above/below water – *space!*)

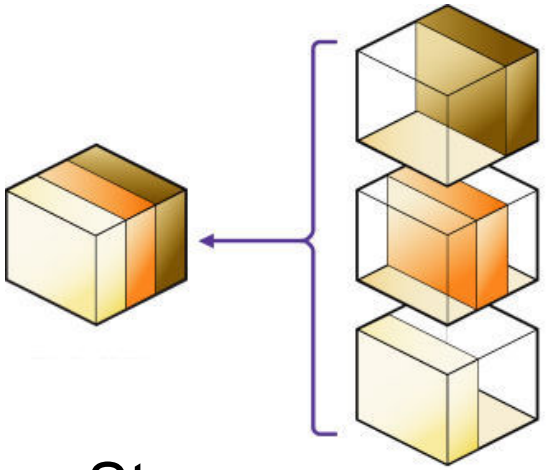
Patient data (to include the atmosphere around them)

Smart phone/watch (TONS of info)

Good TSI Solutions Exist

Hive partitions

- Store as much as you want
- Only read the files you need



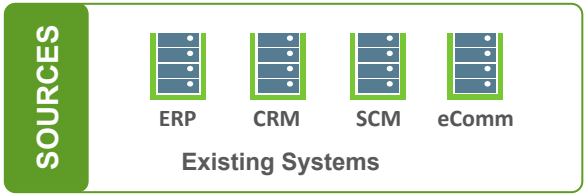
Hive Streaming Data Ingest from Flume or Storm

Sqoop's `--incremental` mode of `append`

- Use appropriate `--check-column`
- ~~"Saved Job" remembering `-last-value`~~

Use Case for an Active Archive

Evolving Domain Data – Hive *likes immutable data*



Need exact copy of mutating tables refreshed periodically

- Structural replica of multiple RDBMS tables
- The data in these tables are being updated
- Don’t need every change; just “as of” content

Start With a Full Refresh Strategy

The epitome of the KISS principle

- Ingest & load new data
- Drop the existing table
- Rename the newly created table

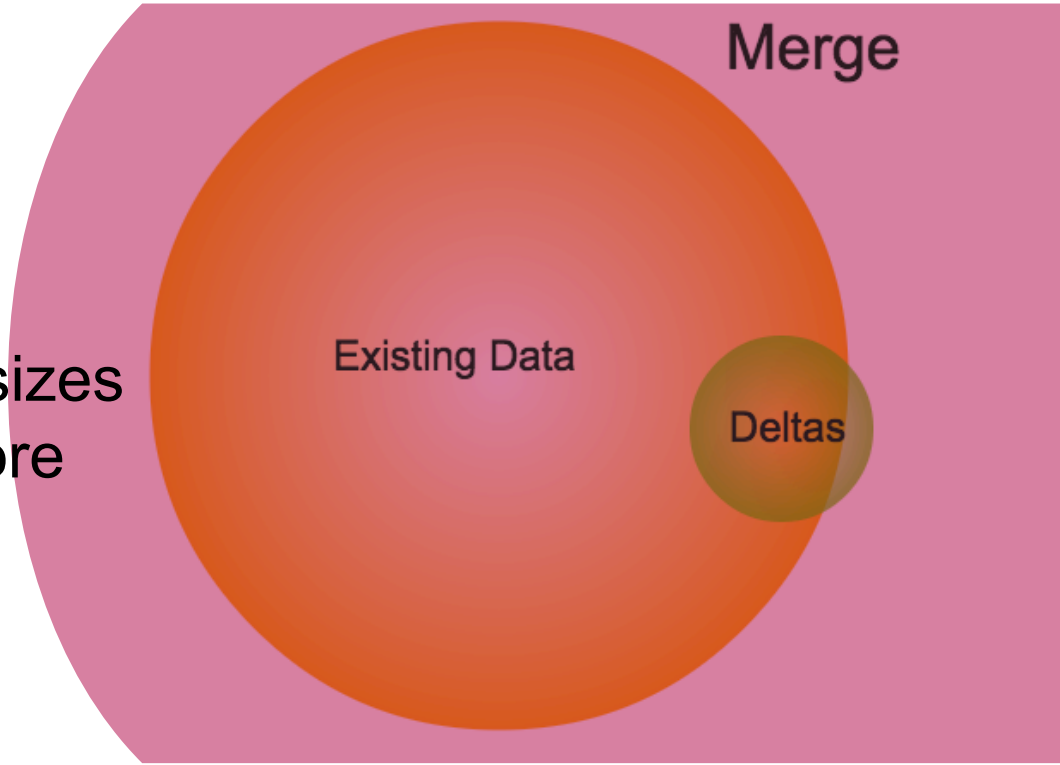
Surely not elegant, but solves the problem until the reload takes longer than the refresh period

Then Evolve to a Merge & Replace Strategy

Typically, deltas are…

- Small % of existing data
- Plus, some totally new records

In practice, differences in sizes of circles is often much more pronounced



Requirements for Merge & Replace

An immutable unique key

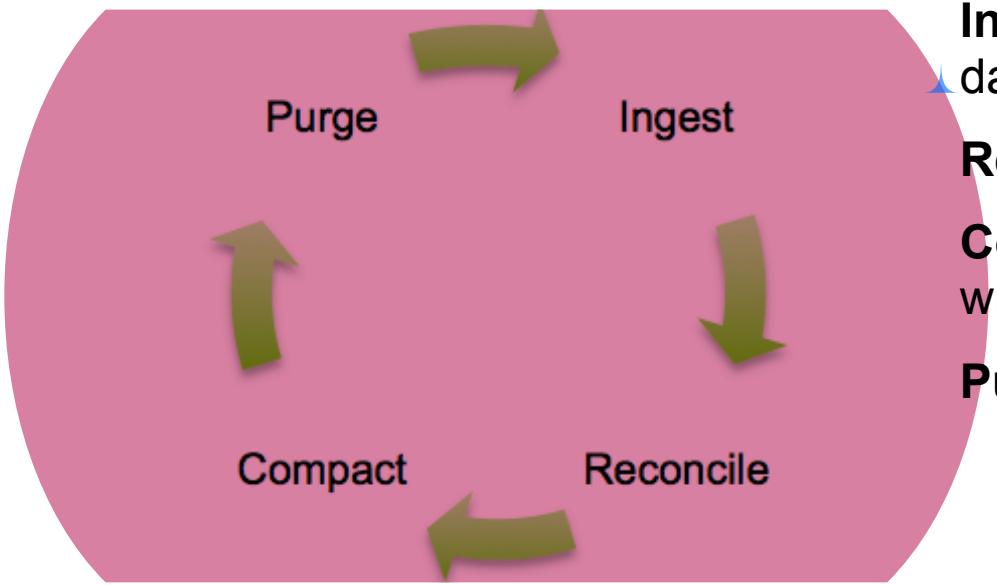
- To determine if an addition or a change
- ~~The source table's (natural or surrogate) PK is perfect~~

~~A last-updated timestamp to find the deltas~~

Leverage Sqoop's `--incremental` mode of `lastmodified` to identify the deltas

- Use appropriate `--check-column`
- "Saved Job" remembering `-last-value`

Processing Steps for Merge & Replace



Ingest – bring over the incremental data

Reconcile – perform the merge

Compact – replace the existing data with the newly merged content

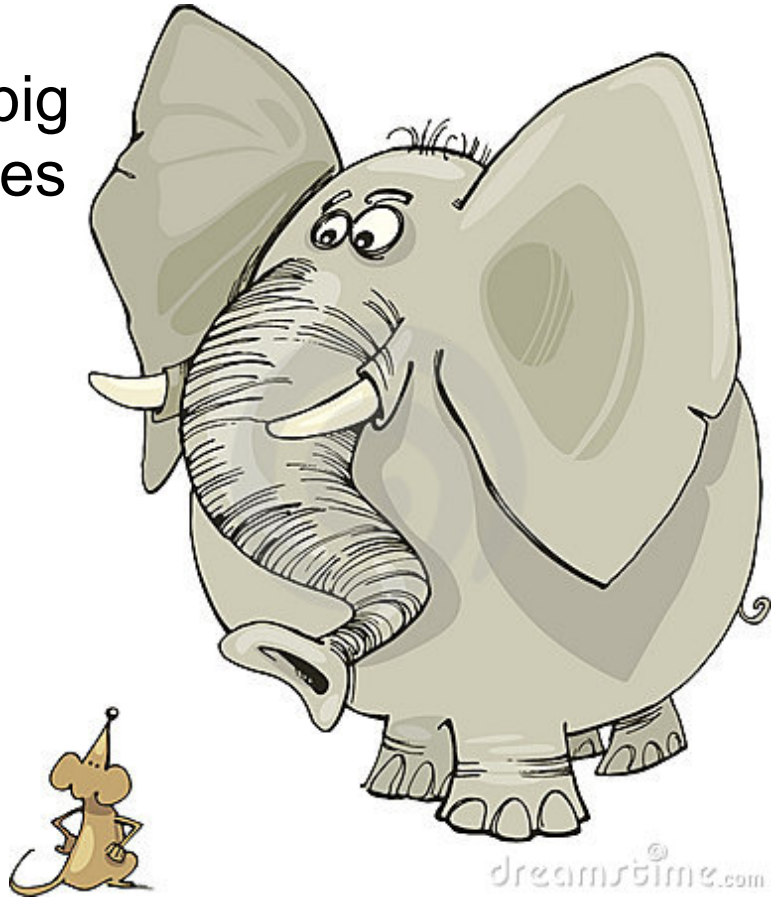
Purge – cleanup & prepare to repeat

See blog at http://hortonworks.com/blog/four-step-strategy-incremental-updates-hive/, *but note that merge can be done in multiple technologies, not just Hive*

Full Merge & Replace Will NOT Scale

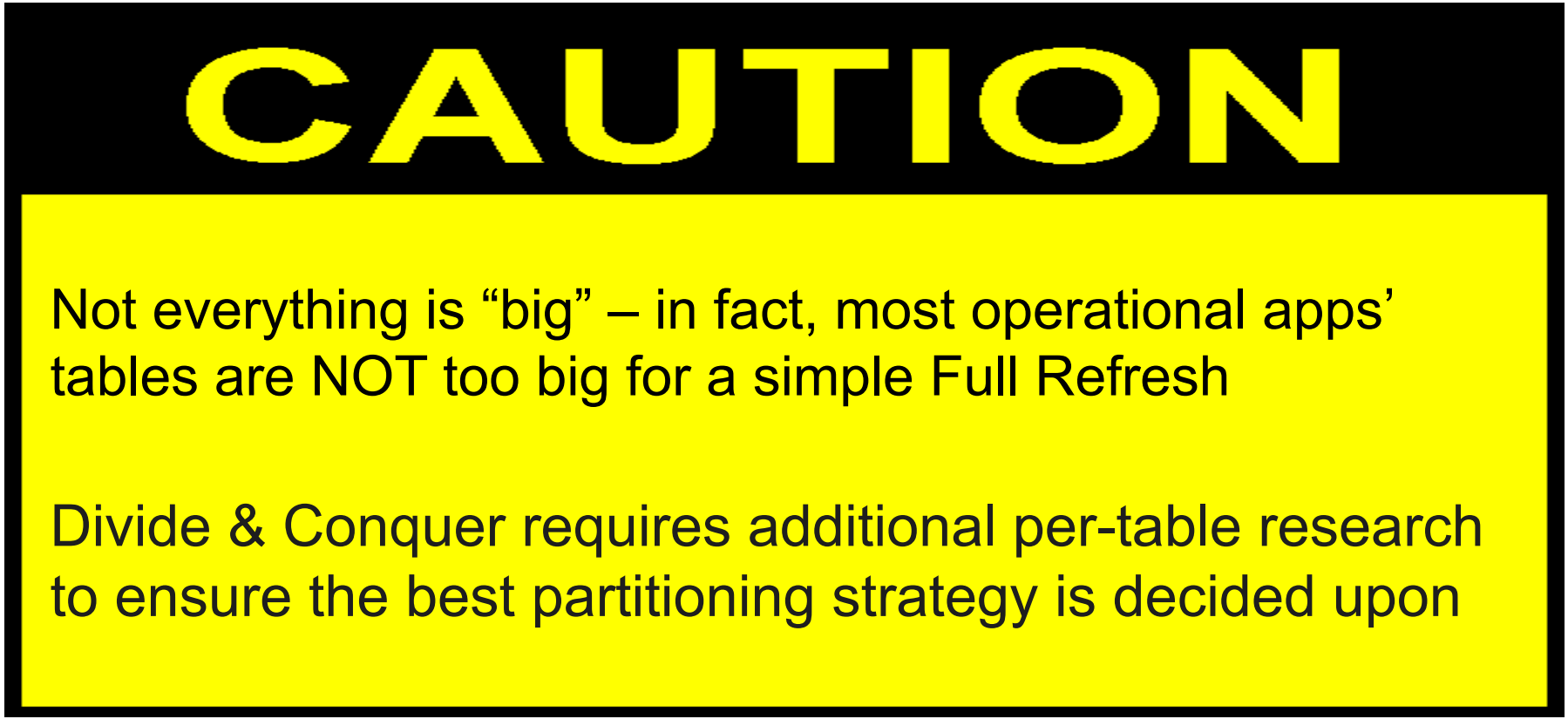
The “elephant” eventually gets too big and merging it with the “mouse” takes too long!

Example: A Hive structure with 100 billion rows, but only 100,000 delta records



What Will? The Classic Hadoop Strategy!

But… One Size Does NOT Fit All…



Criteria for Active Archive Partition Values

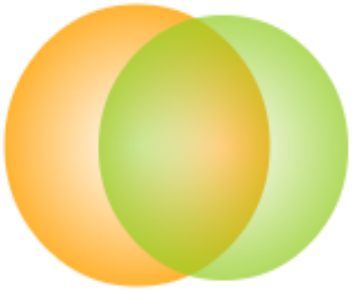
Non-nullable & immutable

Ensures sliding scale growth with new records generally creating new partitions

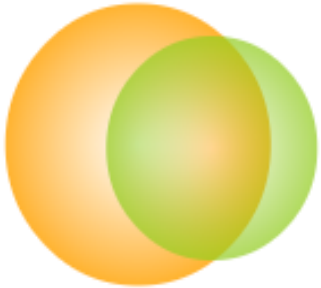
Supports delta records being skewed such that the percentage of partitions needing merge & replace operations is relatively small

Classic value is *(still)* “Date Created”

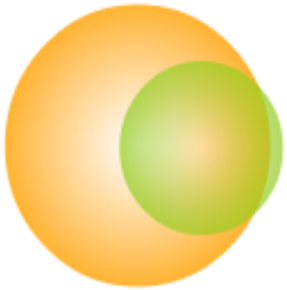
Work on (FEW!) Partitions in Parallel



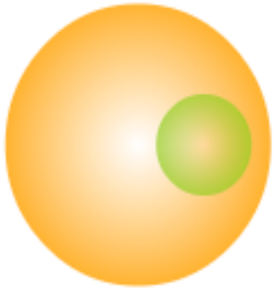
Partition N



Partition N - 1



Partition N - 2



Partition N - 3

Partition-Level Merge & Replace Steps

Generate the delta file

Create list of affected partitions

Perform merge & replace operations for affected partitions

1. Filter the delta file for the current partition
2. Load the Hive table's current partition
3. Merge the two datasets
4. Delete the existing partition
5. Recreate the partition with the merged content

Interested in the Rest of this deck?

Then check out the following links for the slides and video

- http://www.slideshare.net/Hadoop_Summit/mutable-data-in-hives-immutable-world-49979357

- https://www.youtube.com/watch?v=EUz6Pu1lBHQ