2024

CS4440

Hive- A Petabyte Scale Data Warehouse Using Hadoop

Authors: Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu, and Raghotham Murthy
Published 2010 by IEEE

Presenters: Sean Liu, Sanjeev Viswan, Bill Xiao

Background and Motivation

- **Traditional data warehousing is infeasible for enormous business data sets.**
- **Began with Facebook’s need to scale for its volume of data**
- **Hadoop:**
    - Uses map-reduce to mitigate the problem
    - Programming is low-level and hard to maintain.
- **New solution- Hive**
    - HiveQL is compiled into map-reduce jobs, which are executed via Hadoop.
    - Similar to SQL, and as customizable and scalable as Hadoop
    - Used for business intelligence, ML, and product feature support

Related Products

Hadoop

- Open-source project, with speed and scalability from commodity hardware
- Hard for end-users; map-reduce heavy, unintuitive programming

Scope

- SQL-like language using Microsoft's own Cosmos map-reduce

Apache Pig

- uses declarative scripts to process data

Hive

- maintains tables' metadata within the system

Related Literature

- **Chaiken et al. (2008). *SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets*. Proceedings of the VLDB Endowment.**
  - Presents and analyzes syntax, system architecture, compilation, optimization, and scalability of a novel scripting language
  - SCOPE (Structured Computations Optimized for Parallel Execution)
- **Pavlo et al. (2009). *A comparison of approaches to large-scale data analysis*. SIGMOD '09**
  - Compares performance of various DBMS (both parallel and map-reduce) on a 100-node cluster
    - While Hadoop MR was easy to set up, parallel systems outperformed Hadoop MR in data intensive analysis
    - Map-reduce was superior at minimizing work lost upon a hardware failure

Chaiken, R., Jenkins, B., Larson, P.-Å., Ramsey, B., Shakib, D., Weaver, S., & Zhou, J. (2008). Scope. *Proceedings of the VLDB Endowment*, *1*(2), 1265–1276. https://doi.org/10.14778/1454159.1454166

Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., & Stonebraker, M. (2009). A comparison of approaches to large-scale data analysis. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data.* https://doi.org/10.1145/1559845.1559865

Overview of Paper

- Analysis of Hive, and how it streamlines data warehousing and operations
- Integration with Hadoop
- How Hive stores and queries data:
    - Data types, storage, and serialization/deserialization
    - HiveQL queries- comparison to SQL, compilation
- Advantages of Hive
- Applications (in Facebook) and future work

Technical Details: HiveQL Language

- Hive uses SQL's primitive types (e.g. int, decimal, string) and table formats, along with its own complex types (e.g. associative arrays, lists, structs)
- Can support arbitrarily complex information
- Hive query language (HiveQL) is largely similar to SQL (e.g. SELECT, JOIN, CREATE TABLE).
- Important difference- Hive does not have INSERT
- An "insert" overwrites all existing data in a table.
- Can use simple mechanisms, rather than complex locks, to handle concurrency

```
CREATE TABLE t1(st string, fl float, li list<map<string,
struct<p1:int, p2:int>>);
```

Sourced text from original paper:

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Murthy, R., Liu, H., Antony, S., Zhang, N., & Chakka, P. (2010). Hive - A petabyte scale data warehouse using Hadoop | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5447738/

Technical Details: Data Storage

- Data can be stored as tables (directory in file system), partition (subdirectory within table’s directory), bucket (file within partition’s or table’s directory)
- Partition can be created through INSERT or ALTER statements
- Partitions and buckets reduce number of directories that are scanned when processing a query

Partition Example

```
INSERT OVERWRITE TABLE
  test_part PARTITION(ds='2009-01-01', hr=12)
SELECT * FROM t;
```

Sourced text from original paper:

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Murthy, R., Liu, H., Antony, S., Zhang, N., & Chakka, P. (2010). Hive - A petabyte scale data warehouse using Hadoop | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5447738/

Adds a new partition to the table test_part and populates the partition with rows from table t where ds = ‘2009-01-01’ and hr = 12. Overwrites existing data in the partition (if any).

Bucket Example

```
SELECT * FROM t TABLESAMPLE(2 OUT OF 32);
```

Sourced text from original paper:

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Murthy, R., Liu, H., Antony, S., Zhang, N., & Chakka, P. (2010). Hive - A petabyte scale data warehouse using Hadoop | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5447738/

Gets a random sample of 2 out of 32 buckets (or 6.25%) of the data from table t.

Technical Details: Serialization/Deserialization

- Quick definition[1]:
  - Serialization: converts an object's state to a byte stream, which can saved to a file, stored in a database, and used to reconstruct the object
  - Deserialization: converts a byte stream to an object
- In Hive:
  - LazySerDe (default): deserializes rows into internal objects
  - RegexSerDe: lets user specify a regular expression to parse various columns from a row

[1] *What are serialization and deserialization in programming?*. Baeldung on Computer Science. (2024, March 18). https://www.baeldung.com/cs/serialization-deserialization

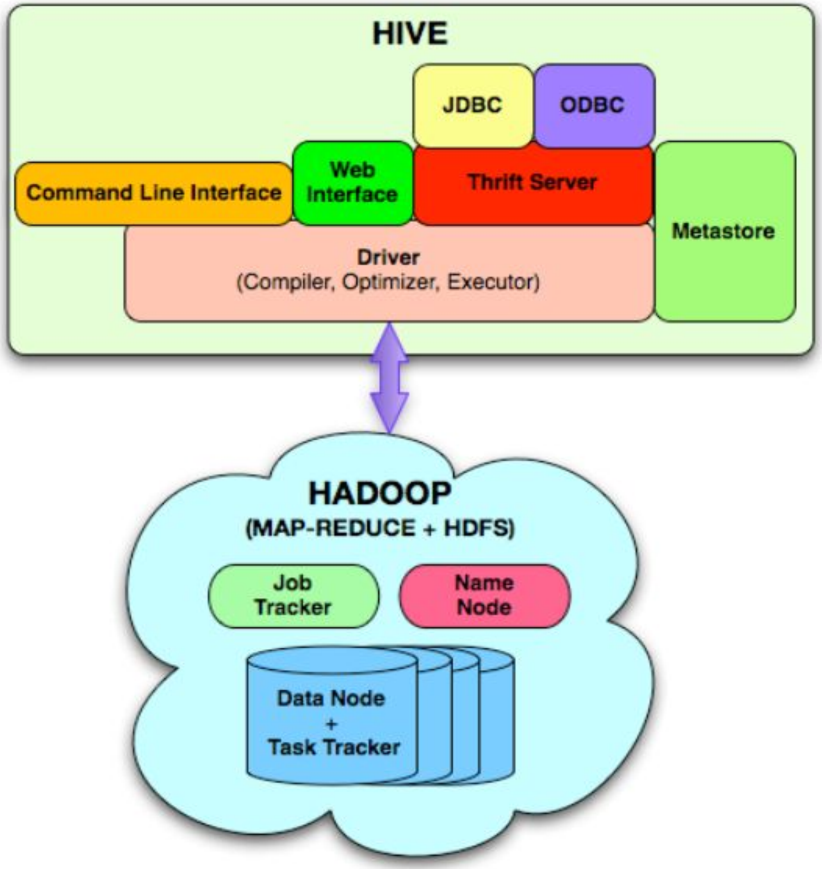

Fig. 1: Hive System Architecture

Technical Details: Metastore

- Metastore: stores metadata about tables, columns, partitions, etc.
  - Metadata can be accessed and modified via Thrift Server
  - Stores metadata in traditional RDBMS for speed (e.g. MySQL in Facebook)
  - Uses DataNucleus, an object relational mapper, to convert object representations into relational schemas (and vice-versa)
  - Metadata musts be backed up regularly, ideally with a duplicate server.

Technical Details: Query Compiler

- Parse query with Antlr to create abstract syntax tree (AST)
- Fetch information from Metastore tables to create a logical plan, while checking for semantic errors in query
  - Creates a directed acyclic graph (DAG)
- Optimize query by performing graph algorithms on DAG, creating a logical plan
- Create physical plan by converting logical plan into map/reduce tasks

Technical Details: Example of Query and Final Plan

```
FROM (SELECT a.status, b.school, b.gender
        FROM status_updates a JOIN profiles b
              ON (a.userid = b.userid
                    AND a.ds='2009-03-20' )) subq1

INSERT OVERWRITE TABLE gender_summary
                              PARTITION(ds='2009-03-20')
SELECT subq1.gender, COUNT(1)
GROUP BY subq1.gender

INSERT OVERWRITE TABLE school_summary
                              PARTITION(ds='2009-03-20')
SELECT subq1.school, COUNT(1)
GROUP BY subq1.school
```

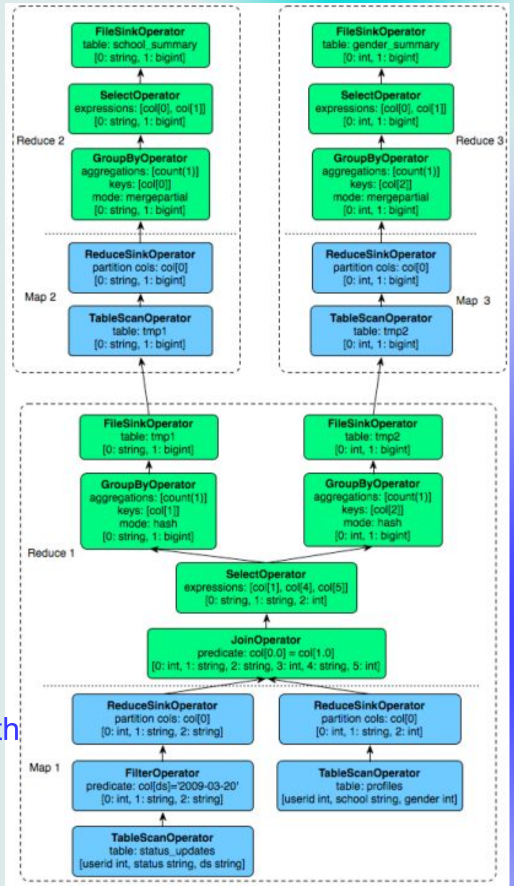

Fig. 3: Query plan for multi-table insert query with 3 map/reduce jobs

Technical Details: Execution Engine

- **Executes tasks with respect to order of dependencies (i.e. execute all of a task's prerequisites before executing such task)**
- **Steps in executing a task:**
  1. **Task serializes its part of the plan into plan.xml file**
  2. **The plan.xml file is added to the task's job cache**
  3. **Hadoop creates instances of ExecMapper and ExecReducers**
  4. **ExecMapper and ExecReducers objects deserialize the plan.xml to execute the corresponding part in the DAG**
  5. **Store the results in a temporary location.**
  6. **Repeat steps 1-5.**
  7. **After the entire query is finished, move the final results to the desired location.**

Evaluation

- Authors’ hypothesis: Hive’s query language (HiveQL) and architecture/data storage are superior to those of other data warehouses, including Hadoop
- Baseline: comparison to Hadoop, as well as systems such as Scope and Apache Pig
- Results of experiments:
  - On-par performance with Hadoop (but still has room for improvement)
  - Analysis of Hive used in Facebook- better data management and processing than Hadoop alone, broad applications of Hive

Hive Application in Facebook

- 700 TB in data warehouse
- Over 75 TB of data processed daily
- Work includes specific, customized queries and reporting dashboards
  - Wide variety of tasks, from data summary to advanced ML
  - Hard to share resources because customized queries are unpredictable
  - Simplified by Hive, operating on the Hadoop cluster
  - Far more efficient than traditional data warehouses

Conclusion and Future Work

- Shortcomings of the approach:
    - Could focus more on quantitative comparisons of Hive vs Hadoop
    - Helpful to have diagrams illustrating tables, partitions, and buckets
- Further work on Hive:
    - Expand Hive to accept all SQL
    - Create cost-based and adaptive optimizers
    - Integrate Hive’s database connection interfaces with business intelligence tools

Study Questions

What are some ways in which regular SQL and HiveQL languages differ, and in what cases might it be advantageous to use one over the other?

What are all the steps required to execute a HiveQL query, from the moment it is written by the user to the moment the final result is stored?

Sources

Chaiken, R., Jenkins, B., Larson, P.-Å., Ramsey, B., Shakib, D., Weaver, S., & Zhou, J. (2008). Scope. *Proceedings of the VLDB Endowment*, *1*(2), 1265–1276. https://doi.org/10.14778/1454159.1454166

Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., & Stonebraker, M. (2009). A comparison of approaches to large-scale data analysis. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. https://doi.org/10.1145/1559845.1559865

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Murthy, R., Liu, H., Antony, S., Zhang, N., & Chakka, P. (2010). Hive - A petabyte scale data warehouse using Hadoop | IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5447738/

What are serialization and deserialization in programming?. Baeldung on Computer Science. (2024, March 18). https://www.baeldung.com/cs/serialization-deserialization/