

facebook

# HIVE

*Data Warehousing & Analytics on Hadoop*

Original Slides by:

Joydeep Sen Sarma, Ashish Thusoo

Facebook Data Team

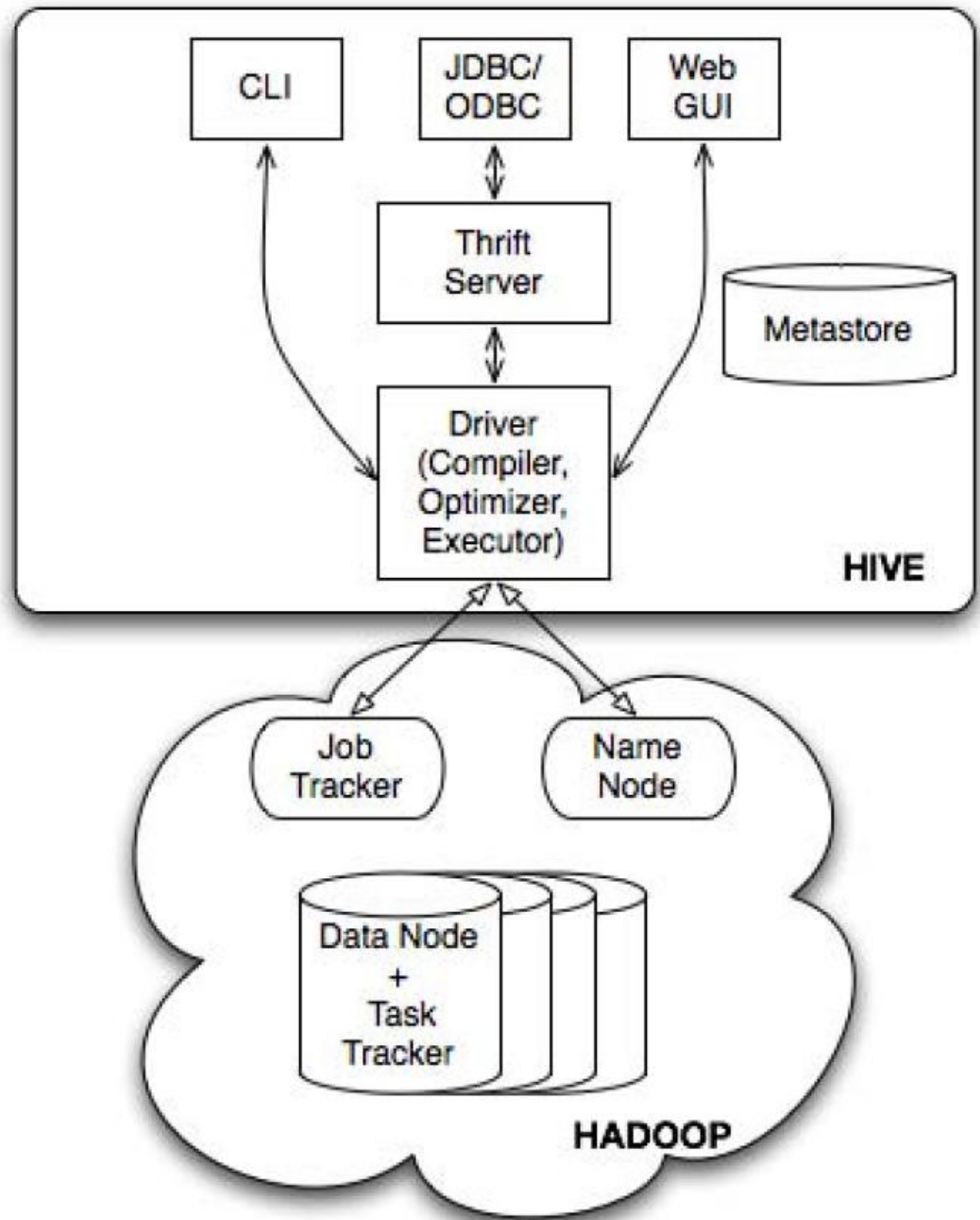
## Why Another Data Warehousing System?

- Problem: Data, data and more data
  - 200GB per day in March 2008 back to 1TB compressed per day today
- The Hadoop Experiment
- Problem: Map/Reduce is great but every one is not a Map/Reduce expert
  - I know SQL and I am a python and php expert
- So what do we do: HIVE

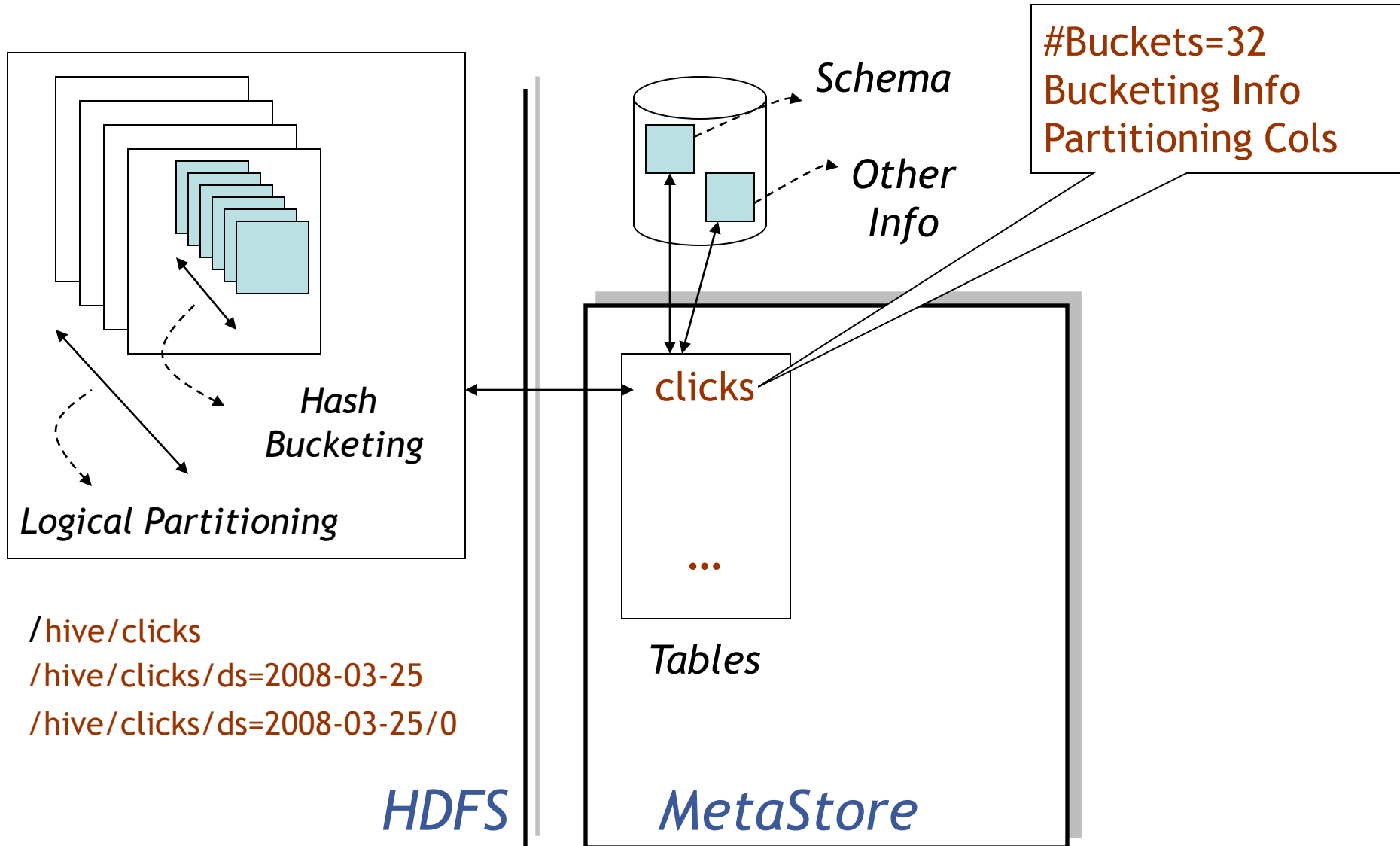
## What is HIVE?

- A system for querying and managing structured data built on top of Map/Reduce and Hadoop
- We had:
  - Structured logs with rich data types (structs, lists and maps)
  - A user base wanting to access this data in the language of their choice
  - A lot of traditional SQL workloads on this data (filters, joins and aggregations)
  - Other non SQL workloads

# Hive Components



# Data Model



# Dealing with Structured Data

- **Type system**
  - Primitive types ~ int, float, strings, dates, booleans
  - Custom defined types
- **Generic (De)Serialization Interface (SerDe)**
  - Defined for all types
  - Custom definable

# MetaStore

- Stores Table/Partition properties:
  - Table schema and SerDe library
  - Table Location on HDFS
  - Logical Partitioning keys and types
  - Other information
- Thrift API
  - Current clients in Php (Web Interface), Python (old CLI), Java (Query Engine and CLI), Perl (Tests)
- Metadata can be stored as text files or even in a SQL backend
  - Not stored in HDFS!

## Hive CLI

- DDL:
  - create table/drop table/rename table
  - alter table add column
- Browsing:
  - show tables
  - describe table
  - cat table
- Loading Data
- Queries



# Hive Query Language

- **Philosophy**
  - SQL like constructs + Hadoop Streaming
- **Query Operators in initial version**
  - Projections
  - Equijoins and Cogroups
  - Group by
  - Sampling
- **Output of these operators can be:**
  - passed to Streaming mappers/reducers
  - can be stored in another Hive Table
  - can be output to HDFS files
  - can be output to local files

## HiveQL Example

```
FROM page_view pv JOIN user u ON (pv.userid = u.id)
INSERT INTO TABLE pv_users
SELECT pv.*, u.gender, u.age
WHERE pv.date = 2008-03-03;
```

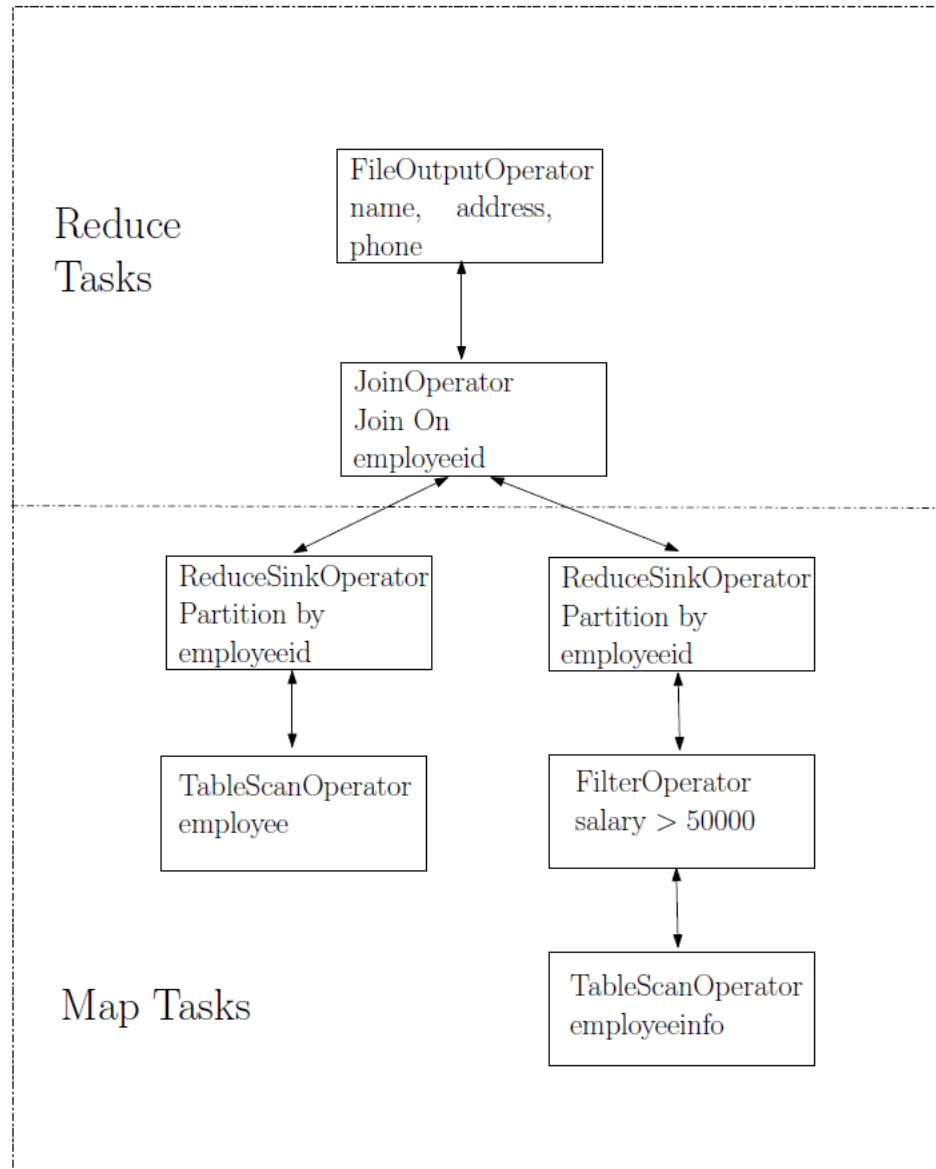
# HiveQL with Custom Map/Reduce Scripts

```
FROM (  
    FROM pv_users  
    SELECT TRANSFORM(pv_users.userid, pv_users.date) USING  
        'map_script'  
    AS(dt, uid)  
    CLUSTER BY(dt)) map  
INSERT INTO TABLE pv_users_reduced  
    SELECT TRANSFORM(map.dt, map.uid) USING 'reduce_script'  
    AS (date, count);
```

# HiveQL Example

```
SELECT a.name, b.address, b.phone  
FROM employees a JOIN employeeinfo b  
ON ( a.employeeid = b.employeeid  
      and b.salary > 50000 )           (Q1)
```

## HiveQL Translated to Query Plan



# Conclusion

The End.