Hive* – A Petabyte Scale Data Warehouse Using Hadoop

Authors
Facebook Data Infrastructure Team

CS 743, Fall 2014

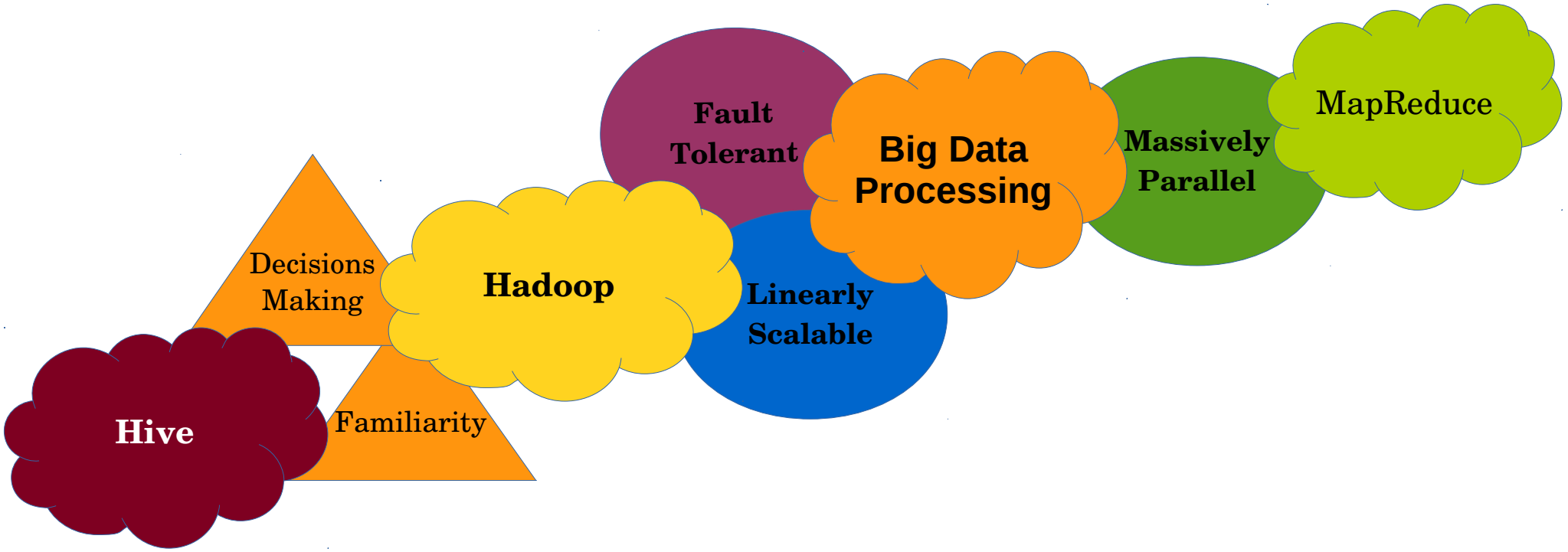
Conference
Data Engineering (ICDE), 2010 IEEE

UNIVERSITY OF
WATERLOO

Presenter
Malek NAOUACH, Nets&Dist Sys

November 13th , 2014

Overview*


Fault Tolerant
Big Data Processing
Massively Parallel
MapReduce
Decisions Making
Hadoop
Linearly Scalable
Hive
Familiarity

Hive Data Structure*

Primitive Data Types

INT | TINYINT | SMALLINT | BIGINT | BOOLEAN | FLOAT

Complex Data Types

Associative arrays | Lists | Structs

Complex Datatypes Composition

list<map<string, struct<p1:int, p2:int>>>

Complex Schema Creation

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

Hive Data Incorporation

+ SerDe Interface
+ ObjectInspector Interface
+ getObjectInspector method

**Serialization

Process of translating data structures or object state into a format that can be stored and reconstructed later.

Hive Query Language*

HiveQL Semantics (SQL)

SUBQUERIES | INNER, LEFT & RIGHT OUTER JOINS | CARTESIAN PROD | GROUP By | AGGREGATION | UNION | CREATE TABLE

NOT HiveQL Semantics

INSERT | UPDATE | DELETE

HiveQL Data Insertion

INSERT OVERWRITE

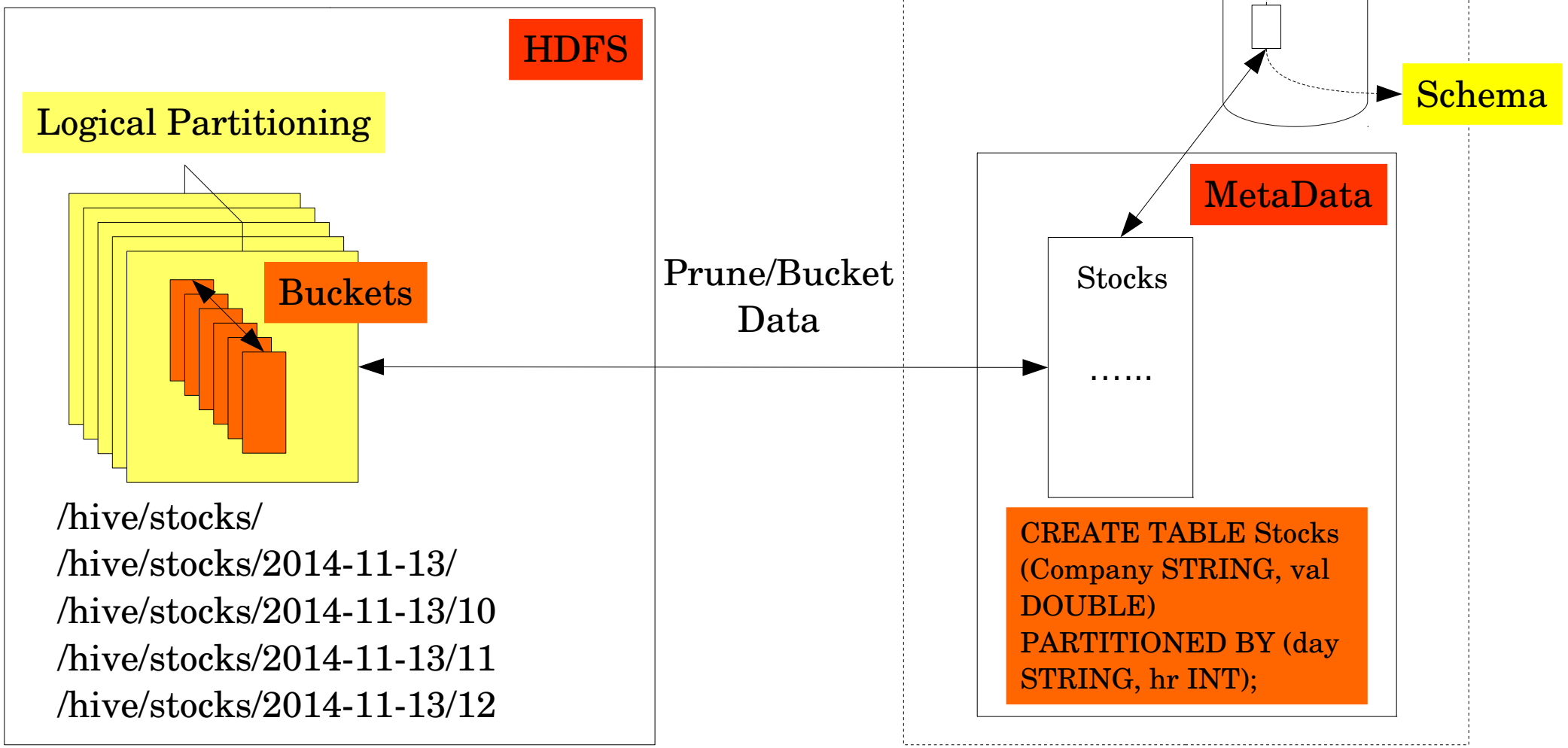
HiveQL Supports Map-Red Programs

```
FROM (
  MAP stocks USING 'python ce_mapper.py'
  AS (company,value)
  FROM stocksStat
  CLUSTER BY value
) a
Reduce company,value USING 'python
ce_reduce.py'
```

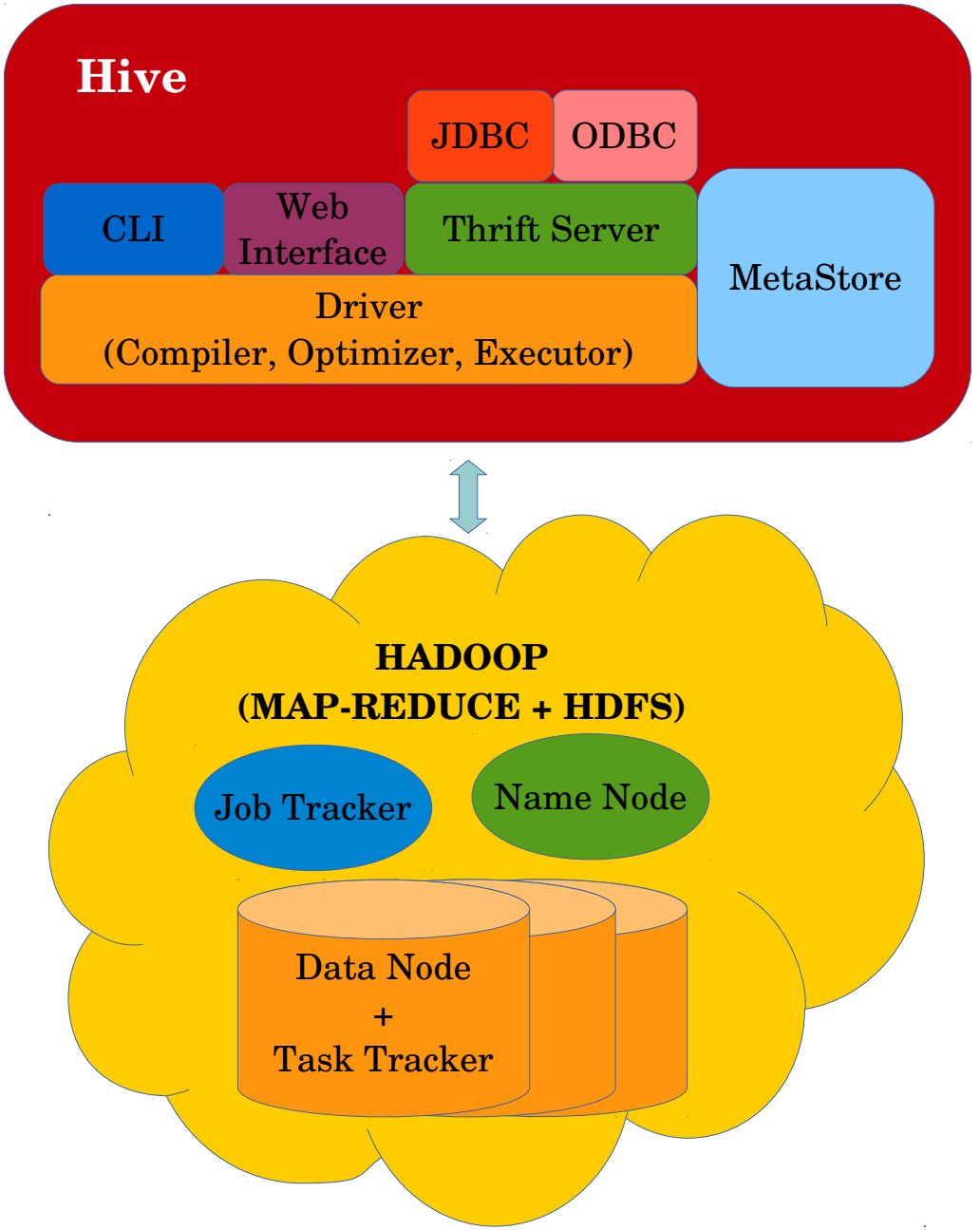
**HQL

Hibernate Query Language

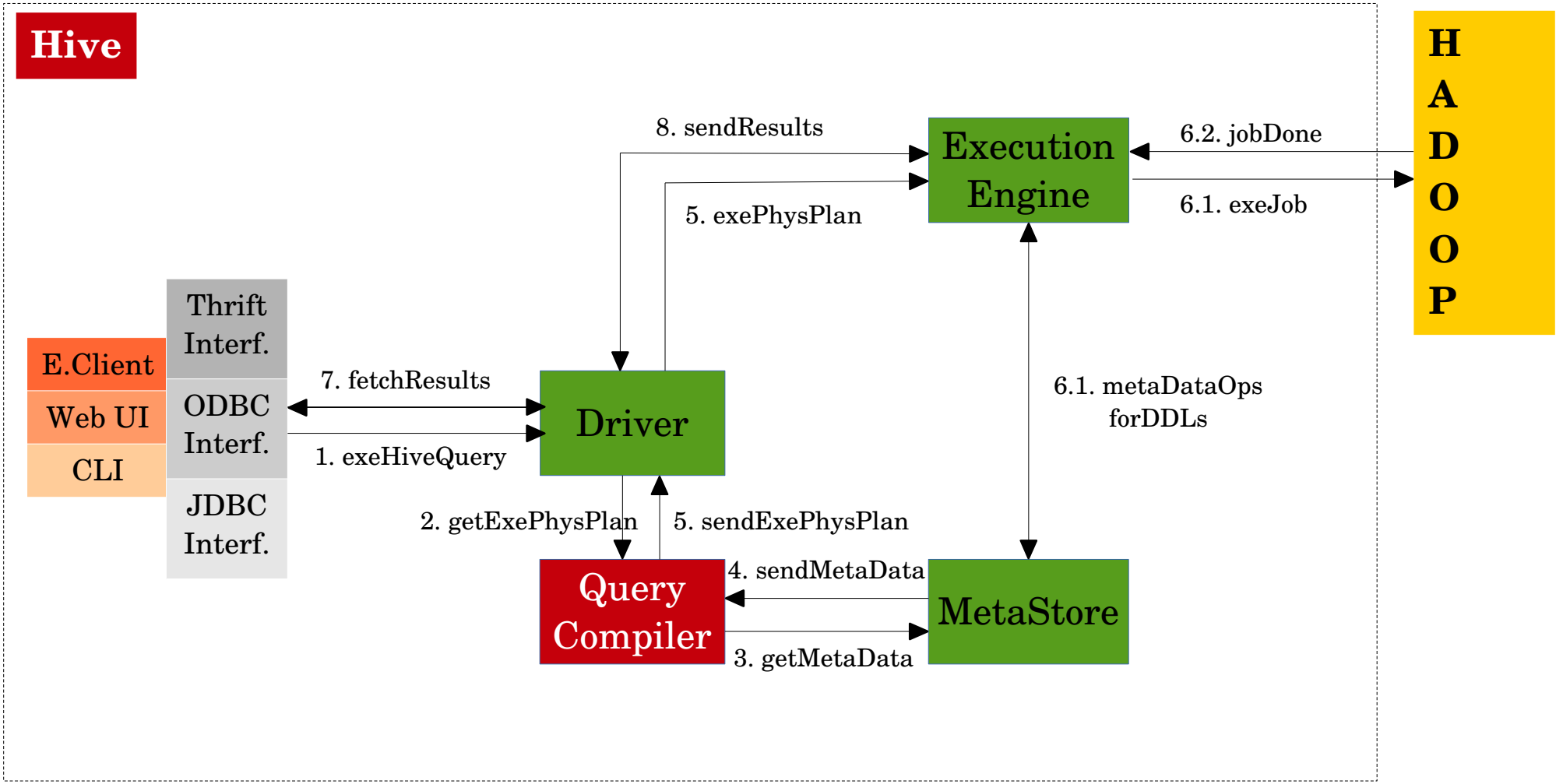
Data Storage*



System Architecture(1/3)*



System Architecture (2/3)*



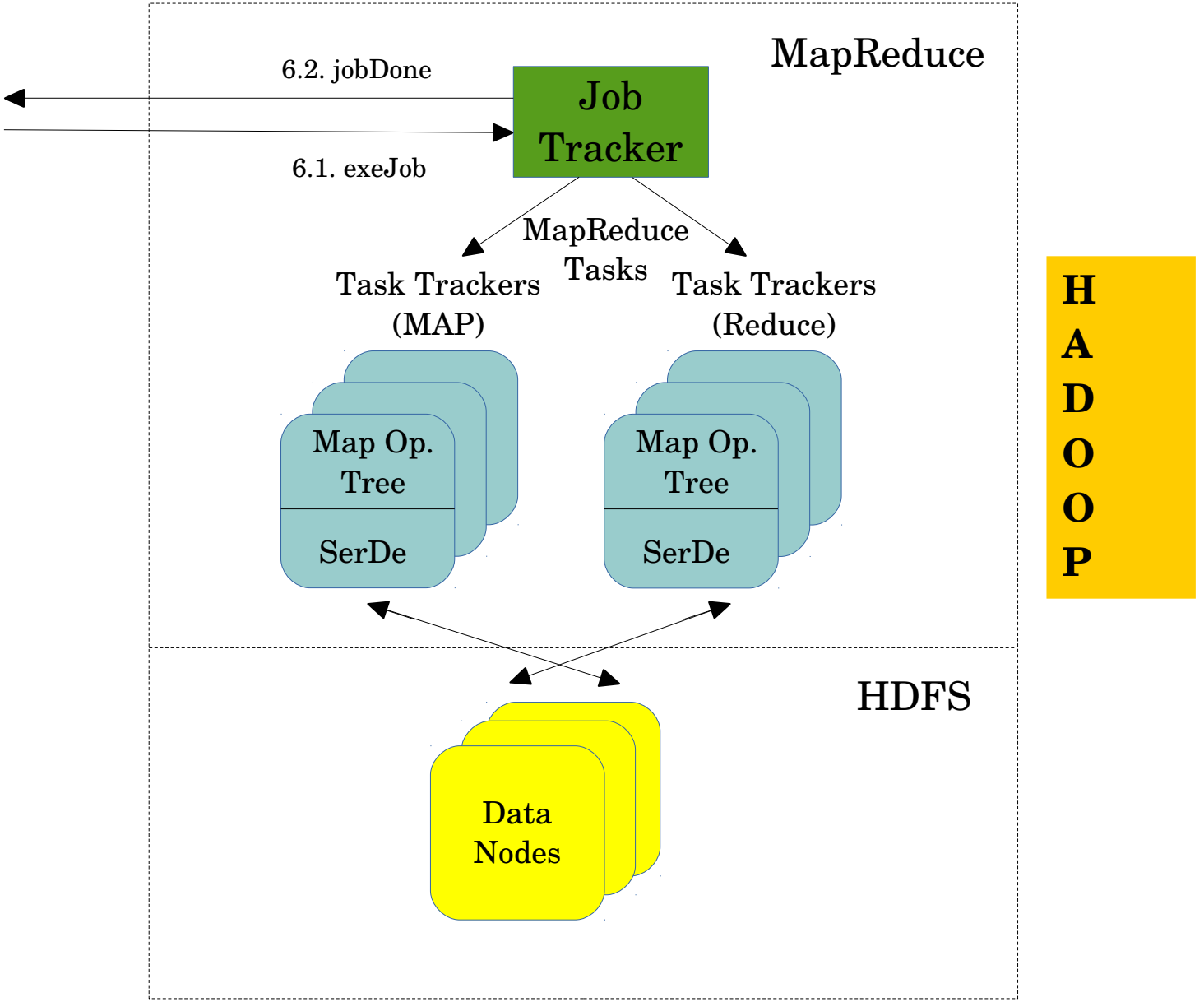
**Interoperability

is the ability of a system to work with other systems without special effort on the customer side.

**Logical/Physical Plan

Abstract Syntax Tree (AST) for the query, Query Block Tree, Involved Interfaces, Directed Acyclic Graph

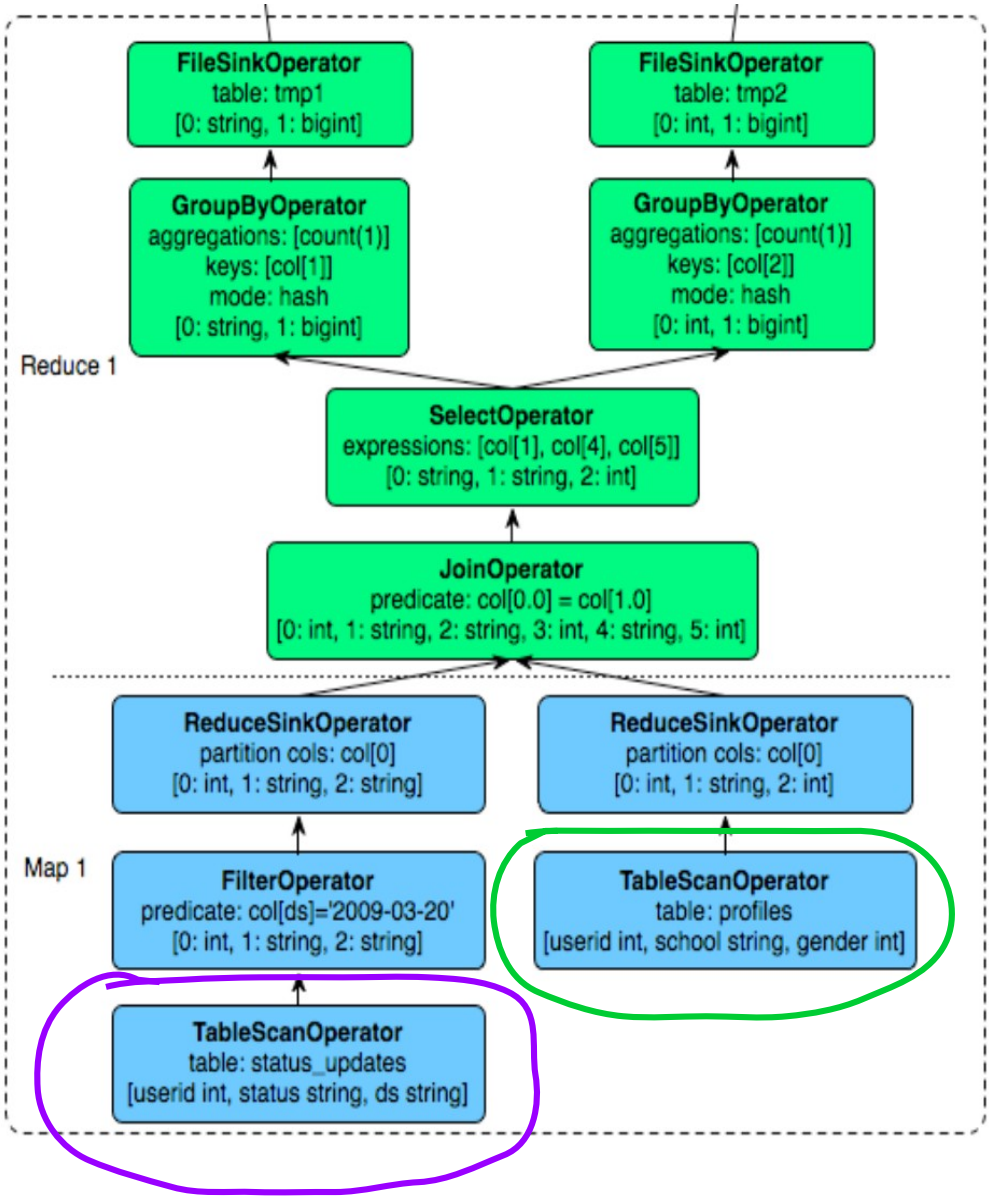
System Architecture (3/3)*



HiveQL to Phys. Plan Exp. (1/3)*

```
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
```

HiveQL to Phys. Plan Exp. (2/3)*



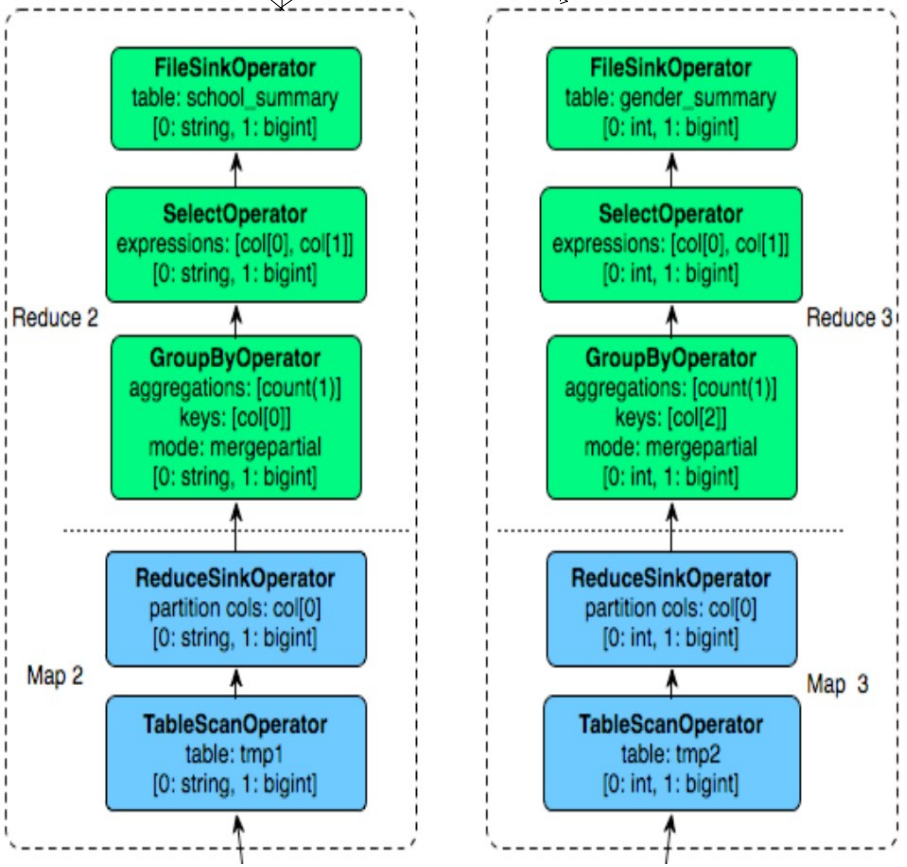
status_updates
(userid, status, ds)

profiles
(userid, school, gender)

HiveQL to Phys. Plan Exp. (3/3)*

SELECT subq1.school, COUNT(1)
GROUP BY subq1.school

SELECT subq1.gender, COUNT(1)
GROUP BY subq1.gender



Brief Recap.*

- Hive is created to simplify big data analysis. (1hour for new users to master)
- Hive is improving the performance of Hadoop. (+20% efficiency)
- Hive enables data processing at a fraction of the cost of more traditional WD.
- Hive is working towards to subsume SQL syntax.
- Hive is enhancing the Query Complier and the interoperability.

http://hadoop.apache.org/

http://hive.apache.org/

Thanks!*

Questions?