Efficient In-situ Processing of Various Storage Types on Apache Tajo

Hadoop Summit 2015 San Jose

Hyunsik Choi, Gruter Inc.

Agenda

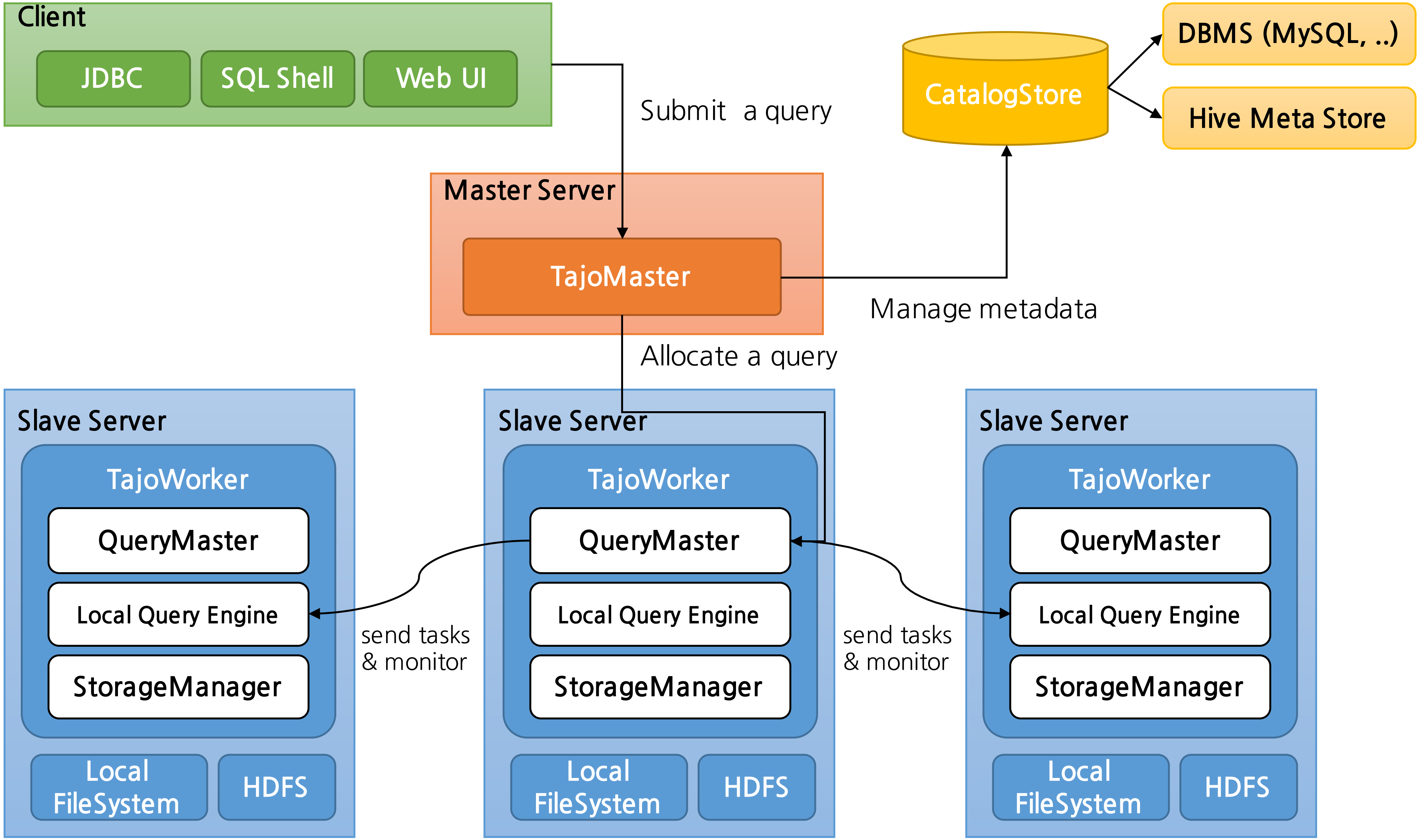
- Tajo Overview
- Various Storage Support
  - Motivation
  - Design Consideration
  - What we did/are doing

An overview of Apache Tajo

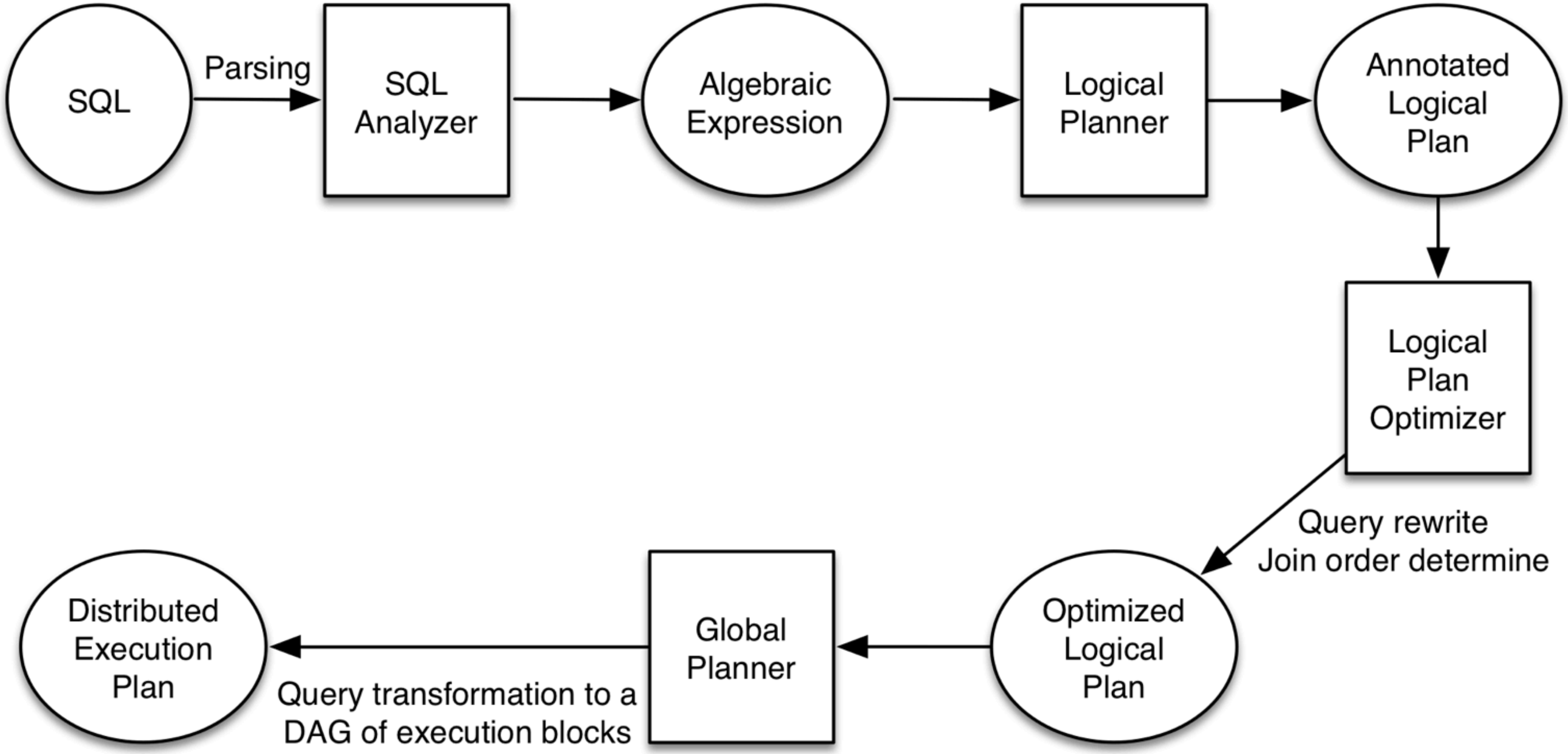
Tajo: A Data Warehouse System

- Data Warehouse System
- Apache Top-level project
- Low latency, and long running batch queries in a single system
  - ~100 ms up to several hours
  - Fault tolerance
- Features
  - ANSI SQL compliance
  - Mature SQL features: Joins, Group by, Sort, Multiple distinct aggregations and Window function
  - Partitioned table support
  - Java/Python UDF support
  - JDBC driver and Java-based asynchronous API
  - SQL data type and Nested type support
  - Direct JSON support

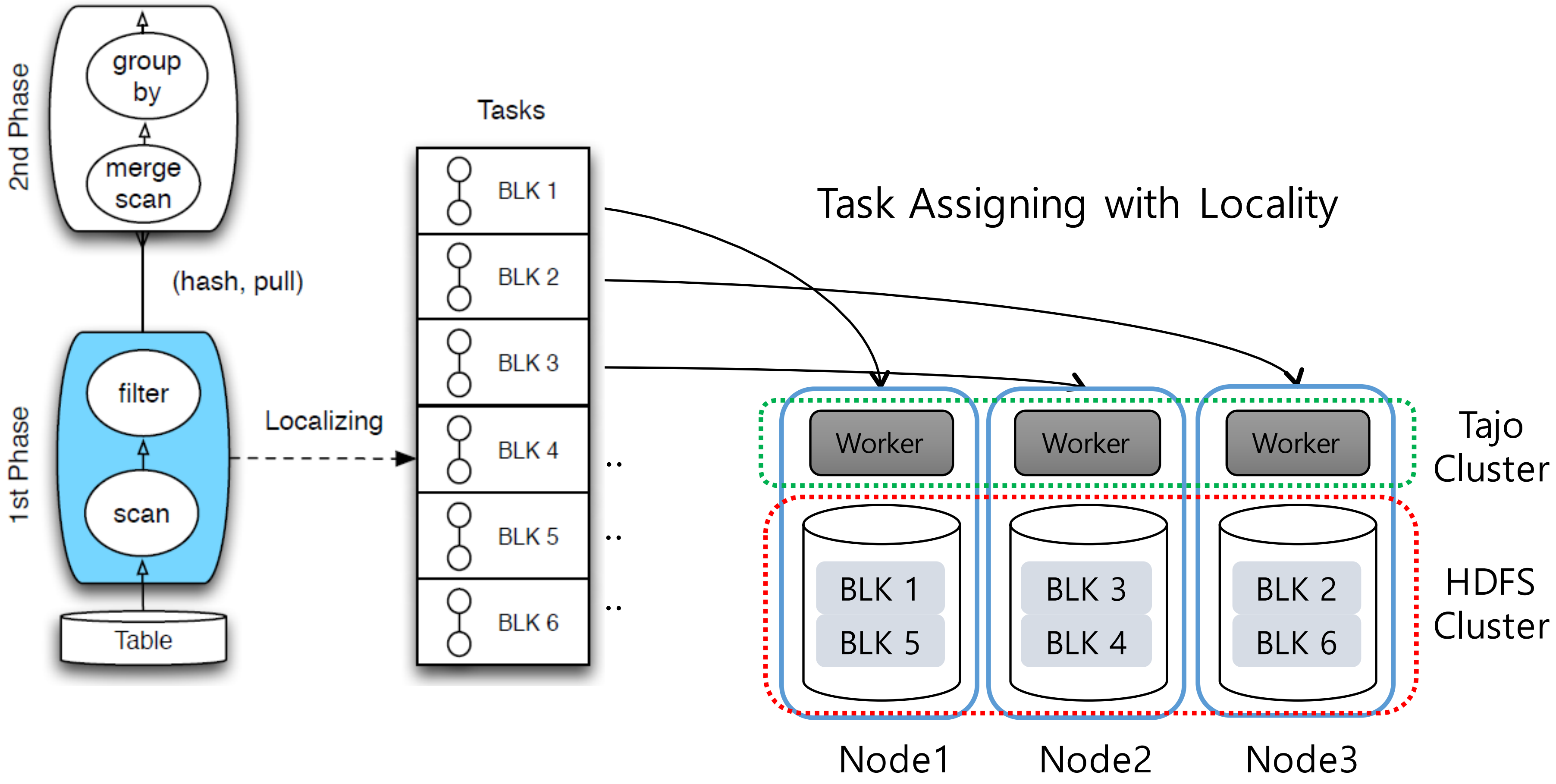
Tajo Overall Architecture



Background: Query Optimization Phases



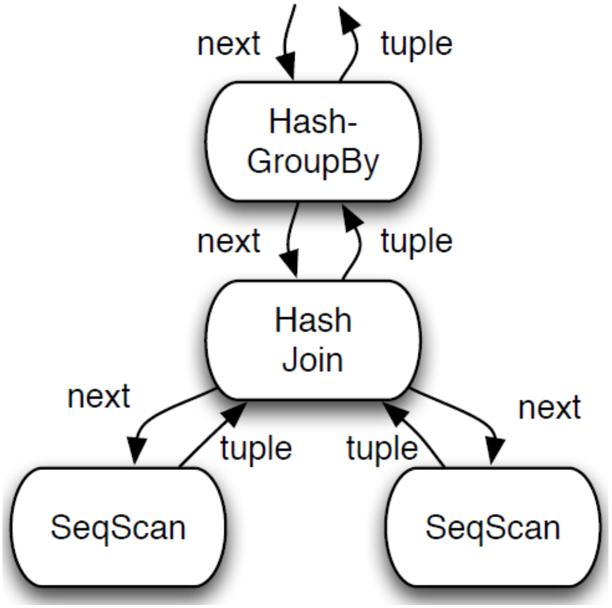
Background: Task Execution



- Each task is assigned to a node according to its locality.

Background: Local Execution

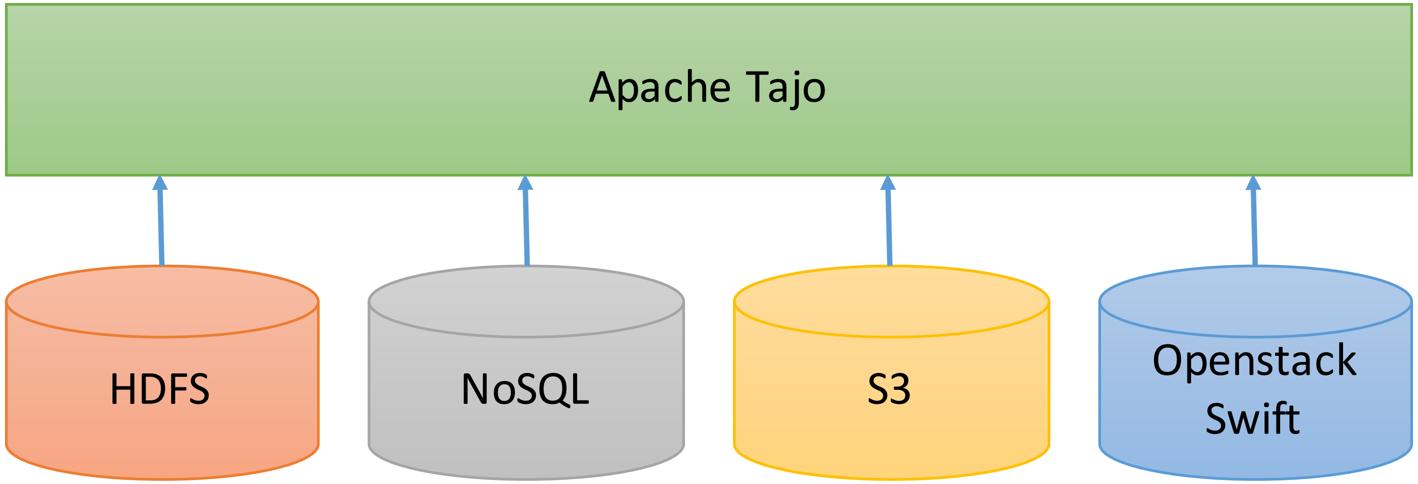
- Physical operators are assembled into a tree and their execution pipelined in the same machine.
- Leaf operators must be scanners.
- Tajo provides abstraction scanner, allowing to read different physical tables.



Various Storage Support

Motivation

- Unified Interface
- Data Integration
- In-situ Processing



Datasets stored in Various Formats/Storages

{JSON}

Parquet

AVRO

Sequence File

Protocol Buffer

RCFile

amazon web services™ S3

APACHE HBASE

Java JDBC

HDFS

elasticsearch.

Design Considerations

- More Storage Properties
  - Splittable, compressible (codecs), indexable, seekable, projectable, aggregatable, …
- Query Optimization
- Pluggable Storage and Data Format
- More operation pushdown

Separation between Storage and Format

Data Formats

{JSON}
Parquet
AVRO

Sequence File

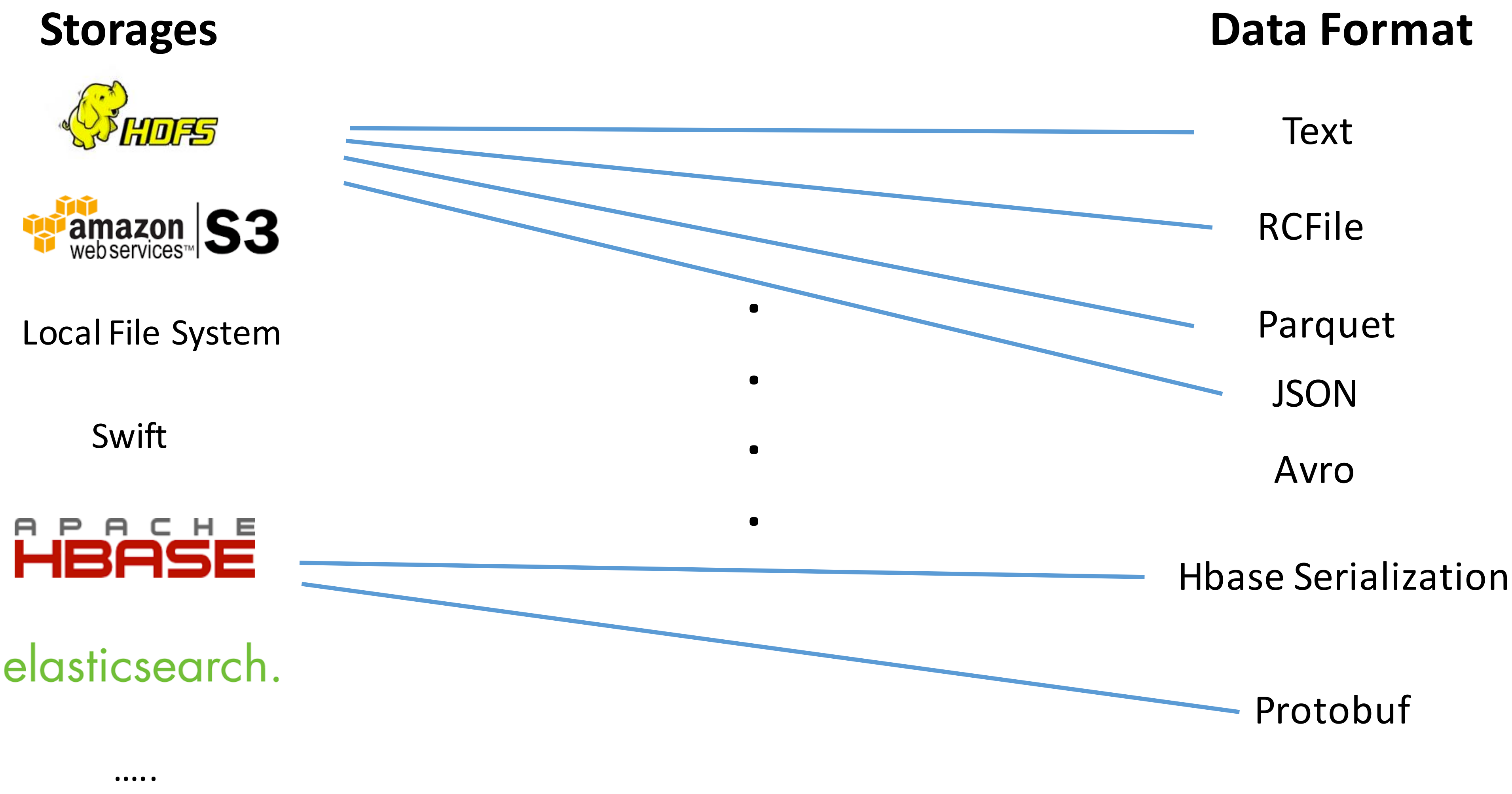
Protocol Buffer

RCFile

Storage Types

amazon web services™ S3
APACHE HBASE
HDFS
elasticsearch.
Java JDBC

Relationships between Storage and Format



Storages

- HDFS
- amazon web services™ S3
- Local File System
- Swift
- APACHE HBASE
- elasticsearch.
- .....

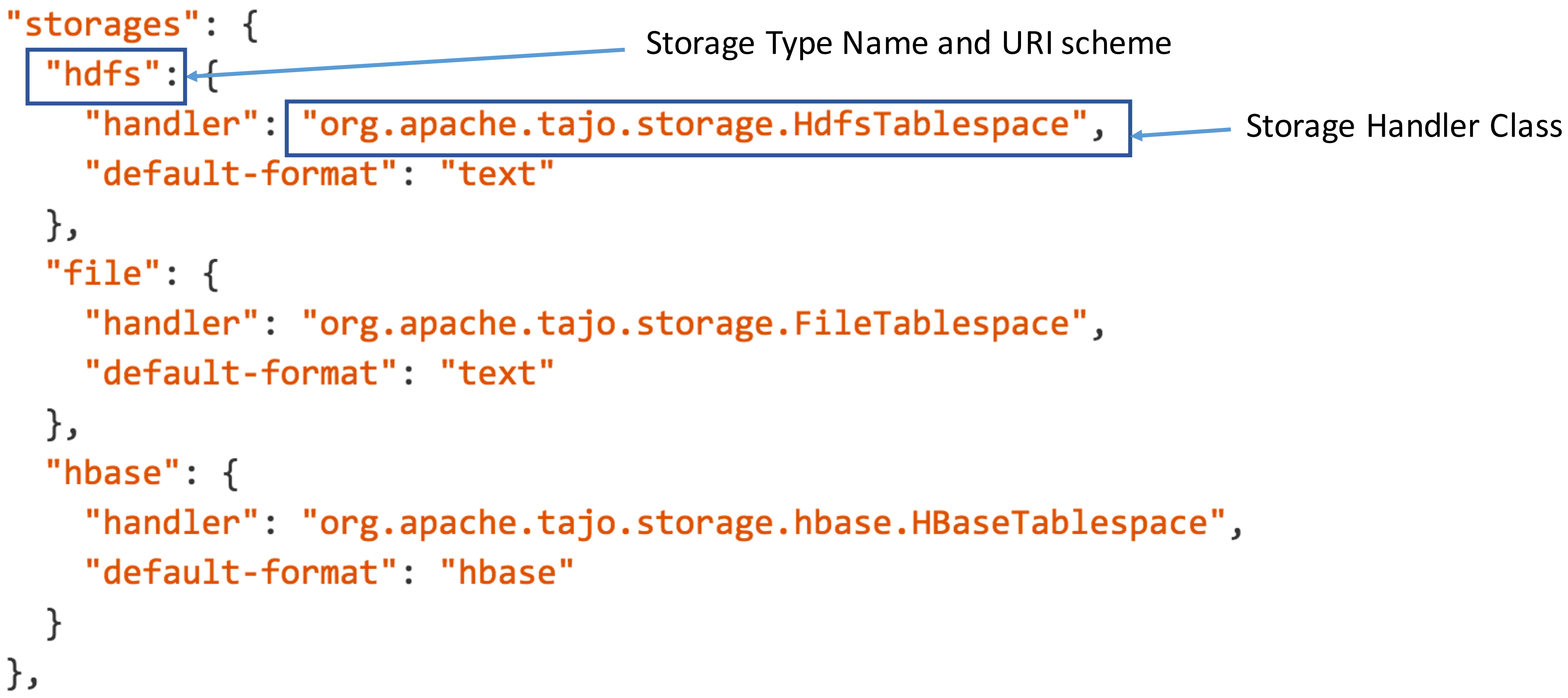
Data Format

- Text
- RCFile
- Parquet
- JSON
- Avro
- Hbase Serialization
- Protobuf

Tablespace

- Tablespace
  - Each table space is identified by a URI.
    - Hdfs://host:port/warehouse, hbase:zk://quorum1:2171, quorum2:2171, …
  - All tables in the same tablespace shares the same physical configuration.
  - URI scheme indicates storage type.
    - Hdfs, hbase, jdbc, …
  - Multiple tablespaces is possible in single storage namespace.
    - HDFS-2832: Enable support for heterogeneous in HDFS.
      - e.g.,
        - /warehouse/ (disk)
        - /today/ (ssd)

Storage Configuration

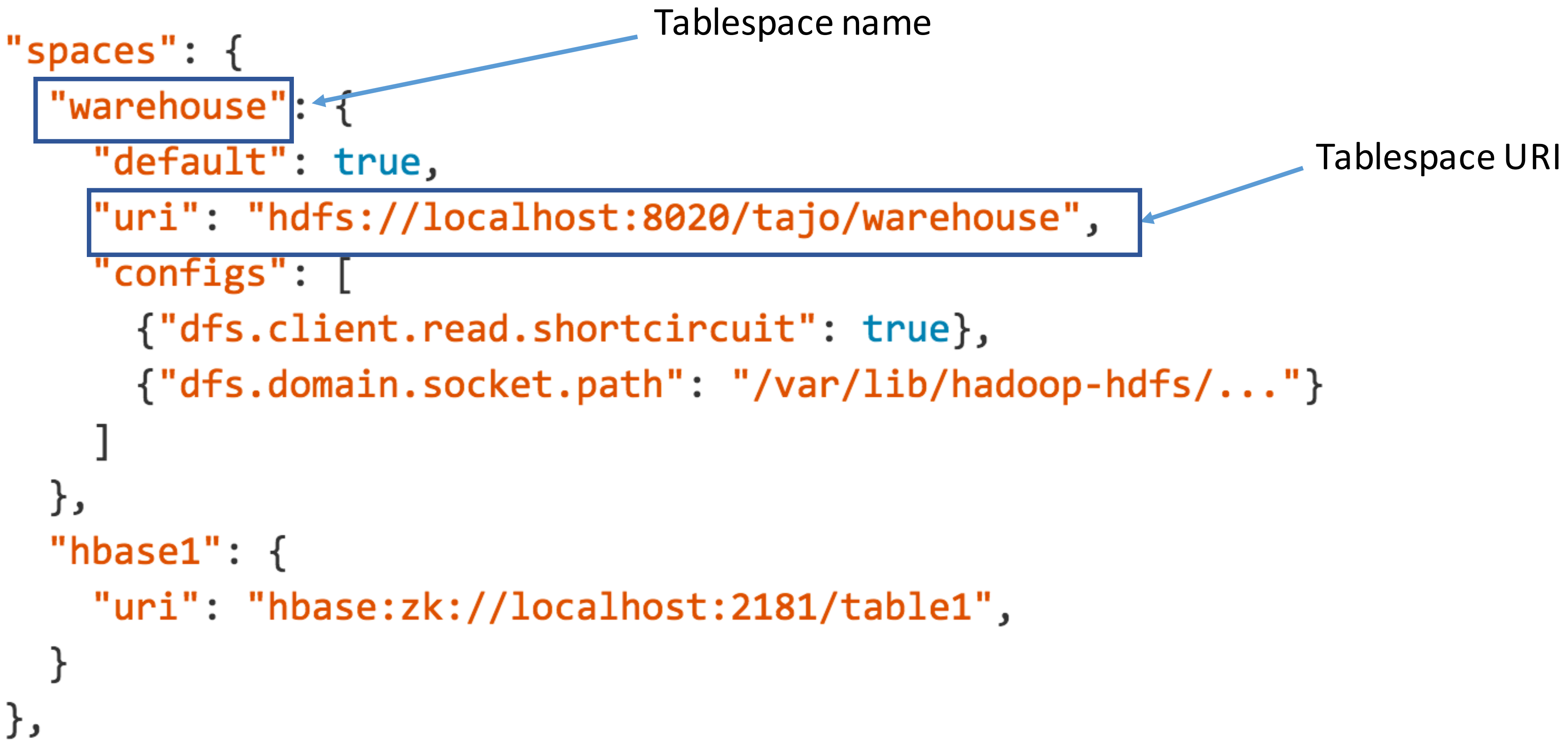


```
"storages": {
  "hdfs": {
    "handler": "org.apache.tajo.storage.HdfsTablespace",
    "default-format": "text"
  },
  "file": {
    "handler": "org.apache.tajo.storage.FileTablespace",
    "default-format": "text"
  },
  "hbase": {
    "handler": "org.apache.tajo.storage.hbase.HBaseTablespace",
    "default-format": "hbase"
  }
},
```

Storage Type Name and URI scheme

Storage Handler Class

Tablespace Configuration

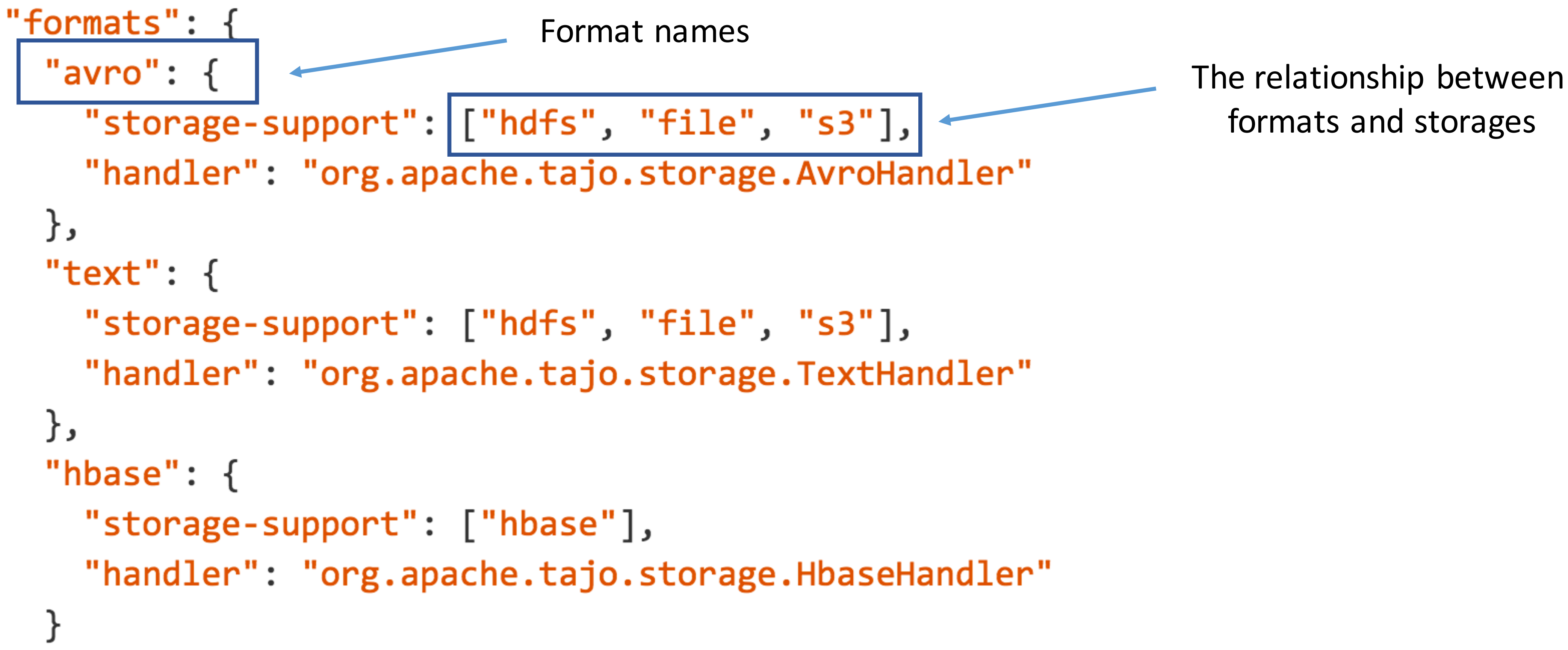


```
"spaces": {
  "warehouse": {
    "default": true,
    "uri": "hdfs://localhost:8020/tajo/warehouse",
    "configs": [
      {"dfs.client.read.shortcircuit": true},
      {"dfs.domain.socket.path": "/var/lib/hadoop-hdfs/..."}
    ]
  },
  "hbase1": {
    "uri": "hbase:zk://localhost:2181/table1",
  }
},
```

Tablespace name

Tablespace URI

Format Configuration

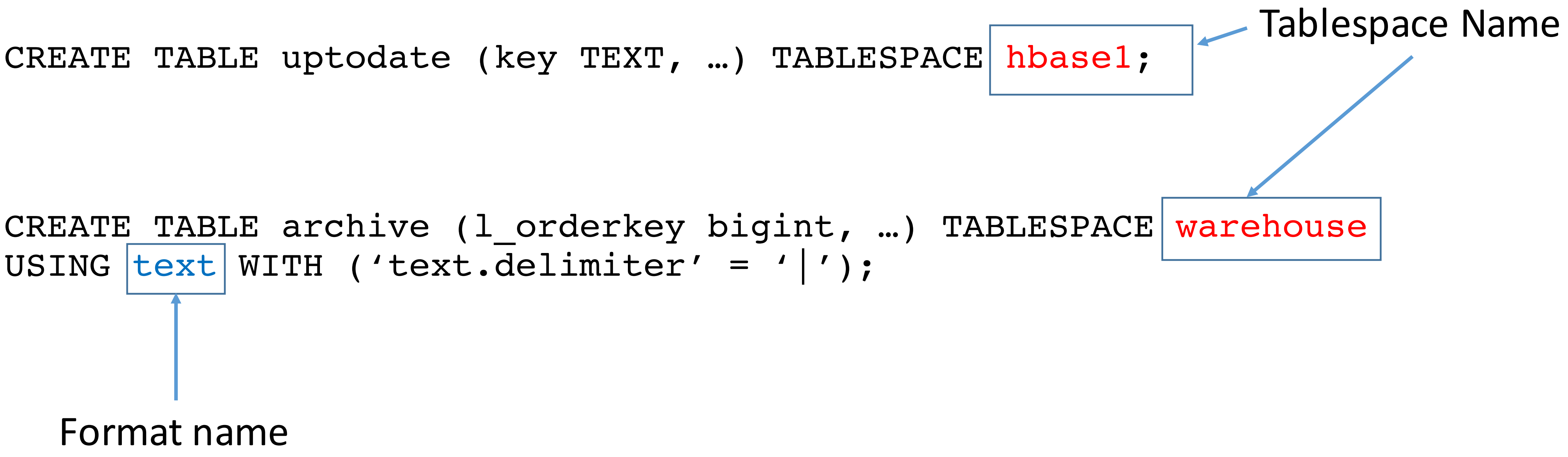


```
"formats": {
  "avro": {
    "storage-support": ["hdfs", "file", "s3"],
    "handler": "org.apache.tajo.storage.AvroHandler"
  },
  "text": {
    "storage-support": ["hdfs", "file", "s3"],
    "handler": "org.apache.tajo.storage.TextHandler"
  },
  "hbase": {
    "storage-support": ["hbase"],
    "handler": "org.apache.tajo.storage.HbaseHandler"
  }
```

Format names

The relationship between formats and storages

CREATE Table using Tablespace



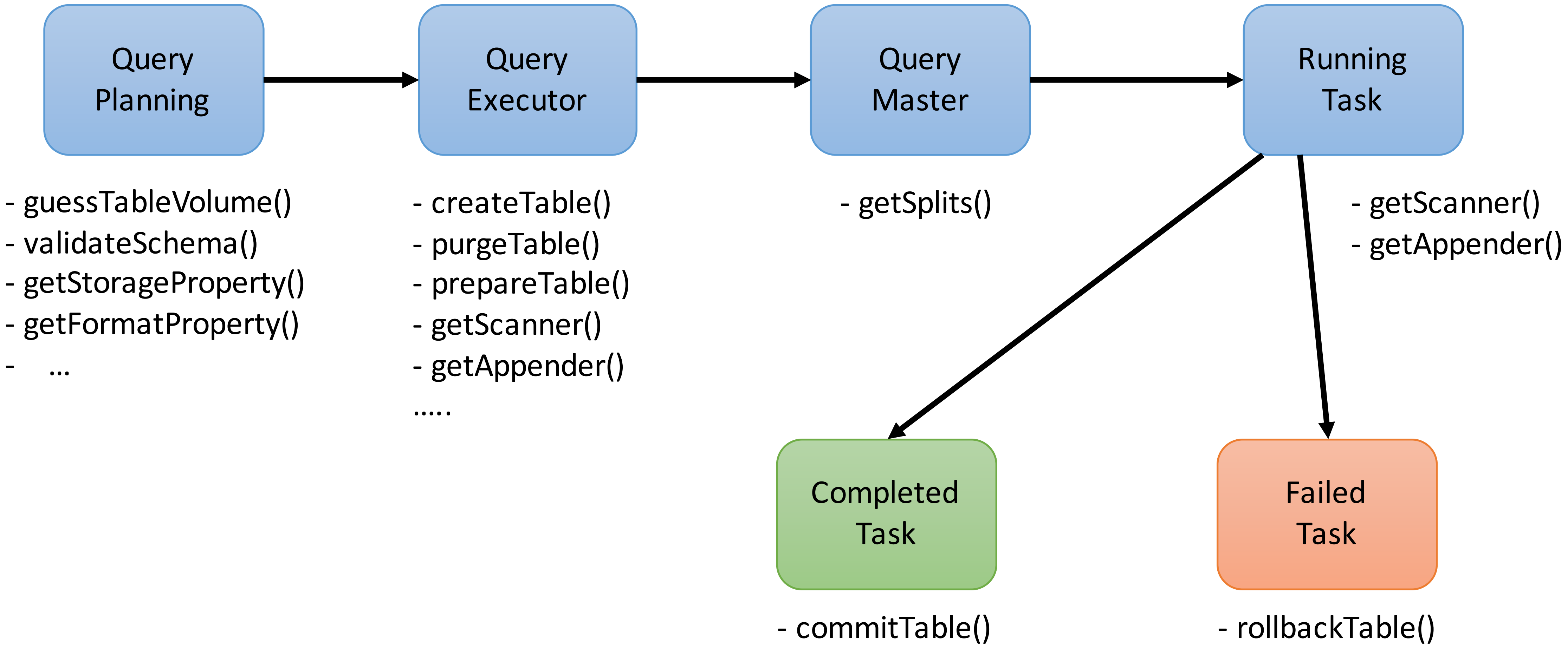
```
CREATE TABLE uptodate (key TEXT, …) TABLESPACE hbase1;
```

```
CREATE TABLE archive (l_orderkey bigint, …) TABLESPACE warehouse
USING text WITH (‘text.delimiter’ = ‘|’);
```

Tablespace Name

Format name

Storage Layer Access over Query Lifecycle



Query Planning

- guessTableVolume()
- validateSchema()
- getStorageProperty()
- getFormatProperty()
- …

Query Executor

- createTable()
- purgeTable()
- prepareTable()
- getScanner()
- getAppender()

…..

Query Master

- getSplits()

Running Task

- getScanner()
- getAppender()

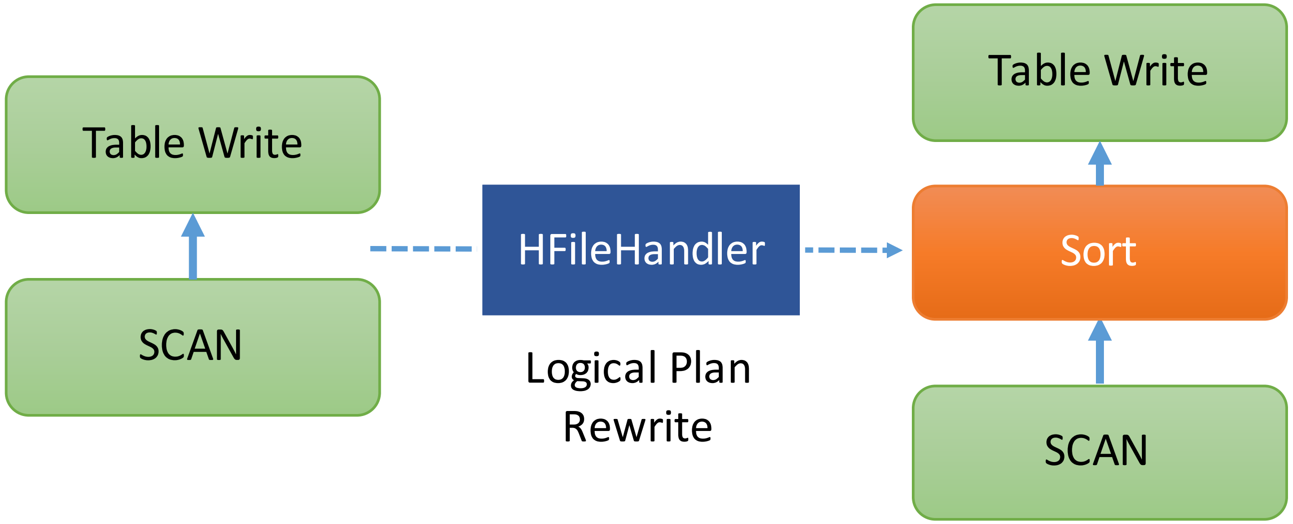
Completed Task

- commitTable()

Failed Task

- rollbackTable()

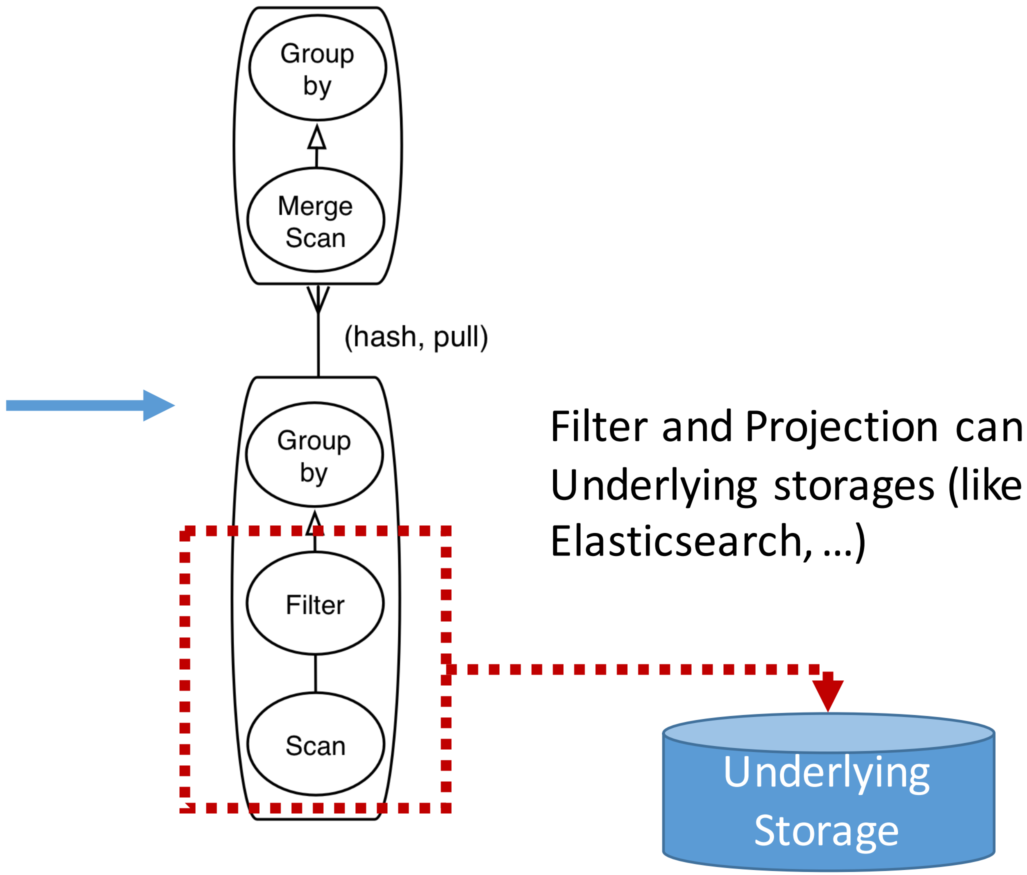
Query Rewrite for Specific Storages



```
CREATE TABLE hbase_table (key TEXT, …)
INSERT INTO hbase_table SELECT id, name, …
```

Operation Push Down

```
SELECT
  X,
  SUM(Y)
FROM
  table1
WHERE
  x > 100
GROUP BY
  x
```



Filter and Projection can be pushed down into Underlying storages (like RDBMS, Hbase, Elasticsearch, …)

Current Status

- Storages:
  - HDFS support
  - Amazon S3 and Openstack Swift
  - Hbase Scanner and Writer - Hfile and Put Mode
  - JDBC-based Scanner and Writer (Working)
  - Kafka Scanner (Patch Available)
  - Elastic Search (Patch Available)
- Data Formats
  - Text, JSON, RCFile, SequenceFile, Avro, Parquet, and ORC (Patch Available)

Get Involved!

- We are recruiting contributors!

- General
  - http://tajo.apache.org

- Getting Started
  - http://tajo.apache.org/docs/0.10.0/getting_started.html

- Downloads
  - http://tajo.apache.org/downloads.html

- Jira – Issue Tracker
  - https://issues.apache.org/jira/browse/TAJO

- Join the mailing list
  - dev-subscribe@tajo.apache.org
  - issues-subscribe@tajo.apache.org

Q&A