# Kylin on Parquet Introduction and Quick start

王汝鹏

2020.04.13

# Agenda

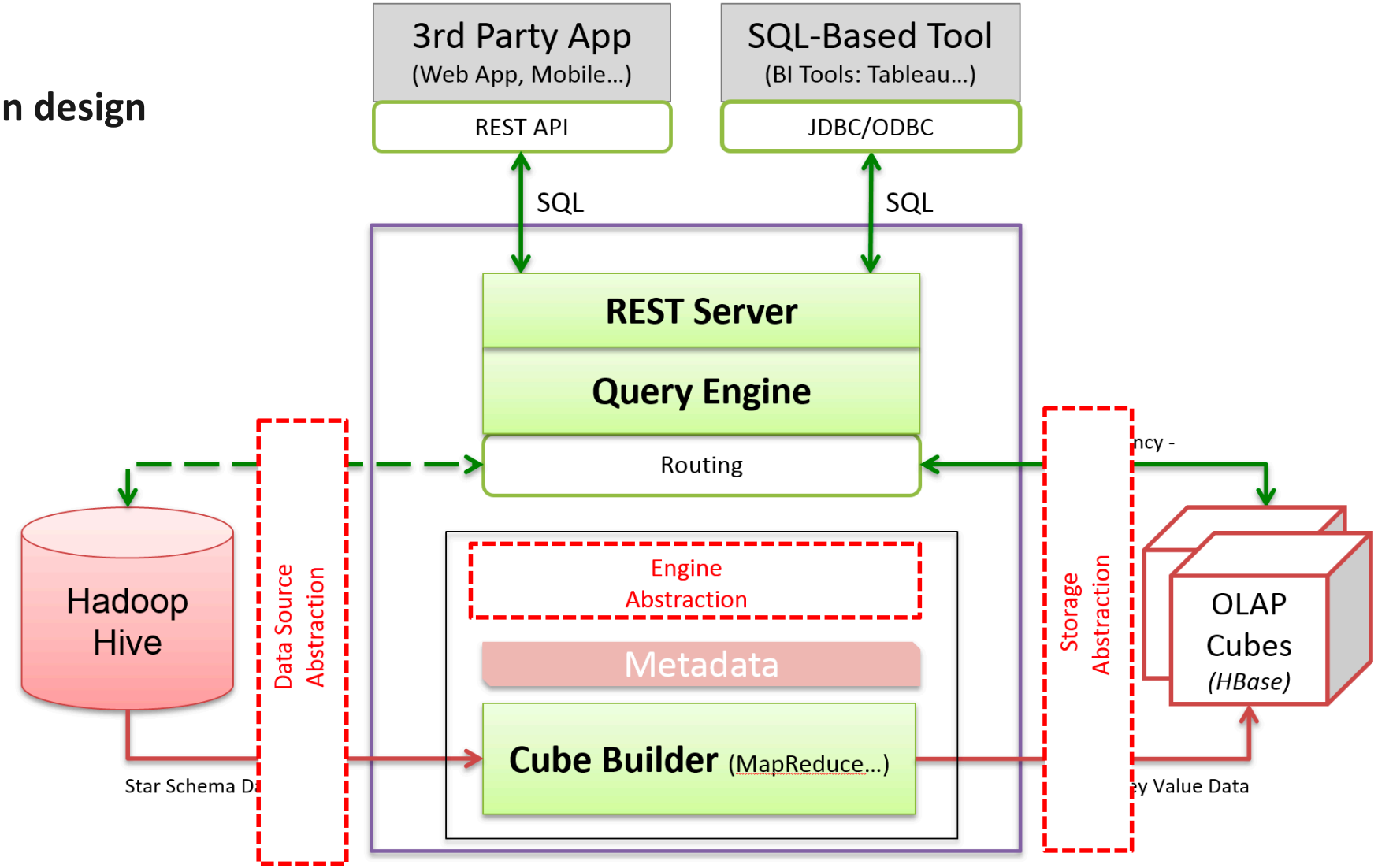
- Architecture
- Why Kylin on Parquet?
- Cube build & Query
- Performance
- Live Demo

# Architecture

1

# Architecture - Apache Kylin

**Plug-in design**

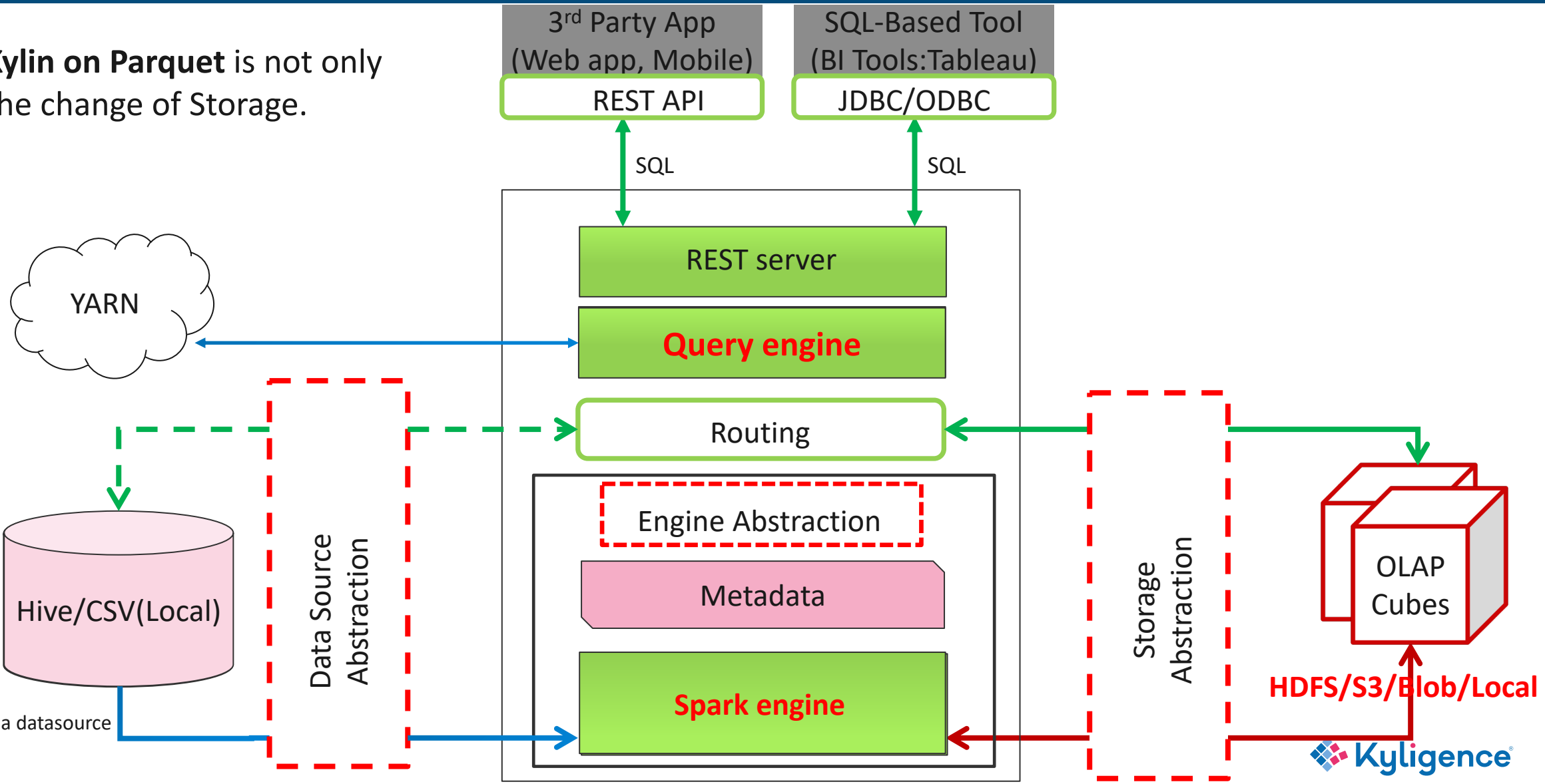





Kyligence

# Architecture - Kylin on Parquet

- **Kylin on Parquet** is not only the change of Storage.



# Why Kylin on Parquet?

2

# Kylin on HBase - Limitations

- Kylin Query engine has a **single point problem**
- The code generated by Calcite is hard to debug
- HBase is KV, not a real columnar storage
- HBase operation and maintenance is difficult to do

CUBE 逻辑视图

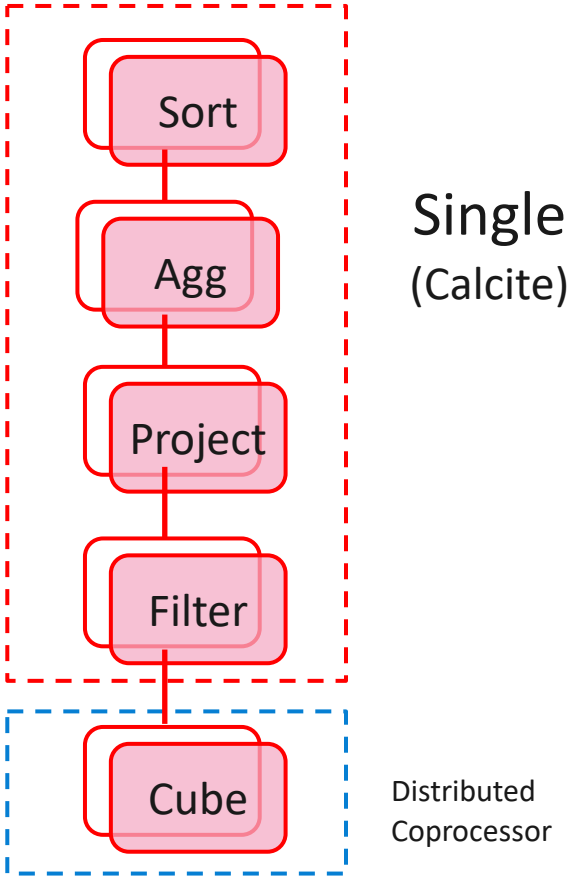
| Dimensions | | Measures | |
|---|---|---|---|
| user_id | name | CNT_ALL | SUM |
| 1 | John | 1000 | 1325 |
| 2 | Tom | 2000 | 1223 |
| 3 | Jack | 1500 | 1232 |

字典视图

| 维度值 | 编码值 |
|---|---|
| John | 1 |
| Tom | 2 |
| Jack | 3 |

存储视图

| Dimensions | Measures |
|---|---|
| 11 | 10001325 |
| 22 | 20001223 |
| 33 | 15001232 |

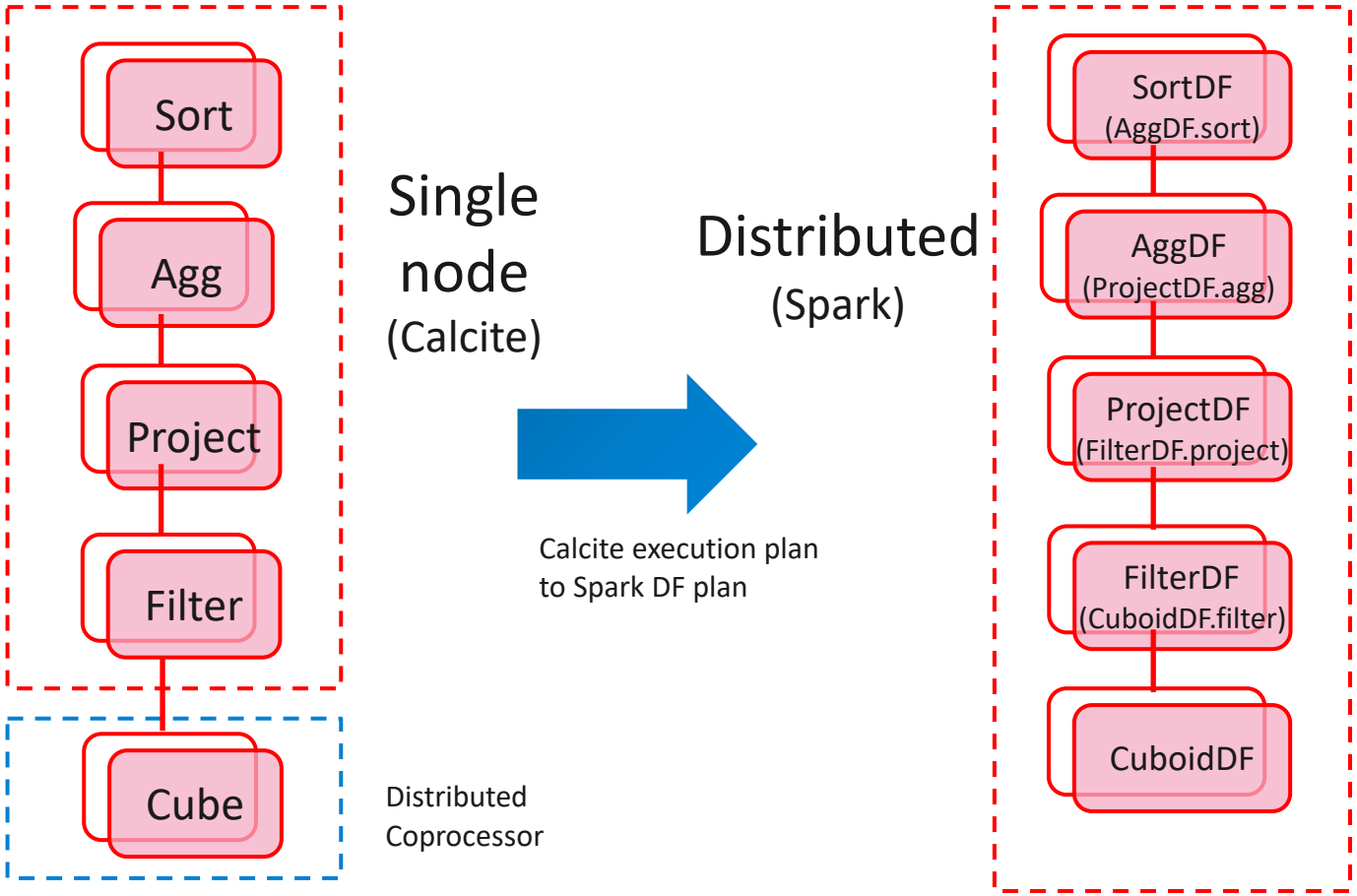


# Kylin on Parquet

- Distributed query execution, less SOF
- Easy to debug by adding breakpoints in each DataFrame
- Real columnar storage
- Compute-storage separation

Kylin Parquet Schema example:

| | | |
|---|---|---|
| 1: | OPTIONAL | INT32 |
| 2: | OPTIONAL | DOUBLE |
| 3: | OPTIONAL | INT64 |
| Measure1: | OPTIONAL | FLOAT |
| Measure2: | OPTIONAL | BINARY |






Kyligence

# Cube Build

3

# Cube Build- Key Features

- All steps in Spark, less dependencies
- Adaptively adjust Spark parameters
- Distributed Global Dictionary generation
- Automatically retry failed job

## Spark Jobs (?)

**User:** root
**Total Uptime:** 1.5 min
**Scheduling Mode:** FIFO
**Completed Jobs:** 588

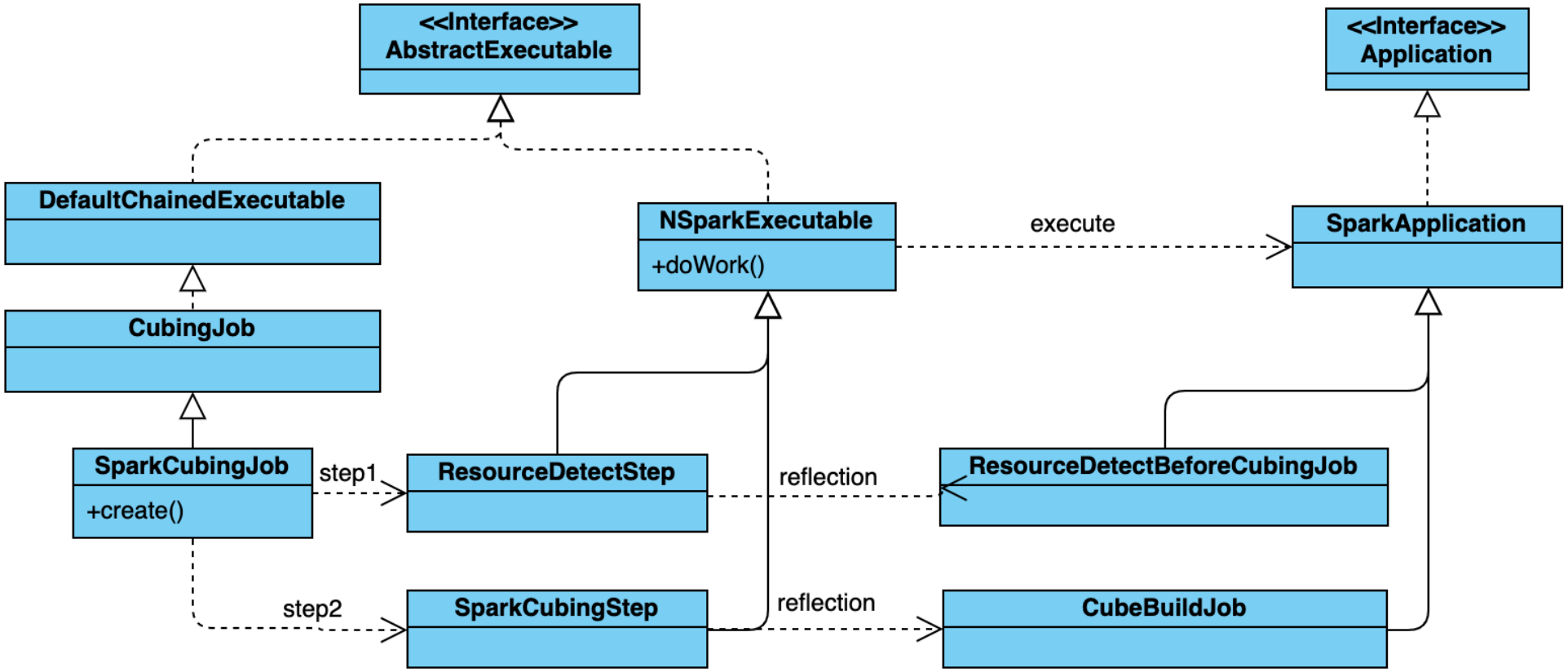
Event Timeline

**Completed Jobs (588)**

Page: 1 2 3 4 5 6 >

| Job Id | Description |
|---|---|
| 587 | build 16384 from parent 16387<br>parquet at ParquetStorage.scala:28 |
| 586 | build 16384 from parent 16387<br>parquet at ParquetStorage.scala:32 |
| 585 | build 16384 from parent 16387<br>parquet at ParquetStorage.scala:28 |
| 584 | parquet at NBuildSourceInfo.java:67<br>parquet at NBuildSourceInfo.java:67 |
| 583 | build 20480 from parent 20483<br>parquet at ParquetStorage.scala:28 |
| 582 | build 20480 from parent 20483<br>parquet at ParquetStorage.scala:32 |
| 581 | build 16544 from parent 16547<br>parquet at ParquetStorage.scala:28 |
| 580 | build 16387 from parent 16467<br>parquet at ParquetStorage.scala:28 |
| 579 | build 17152 from parent 17155<br>parquet at ParquetStorage.scala:28 |
| 578 | build 16544 from parent 16547<br>parquet at ParquetStorage.scala:32 |
| 577 | build 16464 from parent 16467<br>parquet at ParquetStorage.scala:28 |

# Cube Build - Interfaces

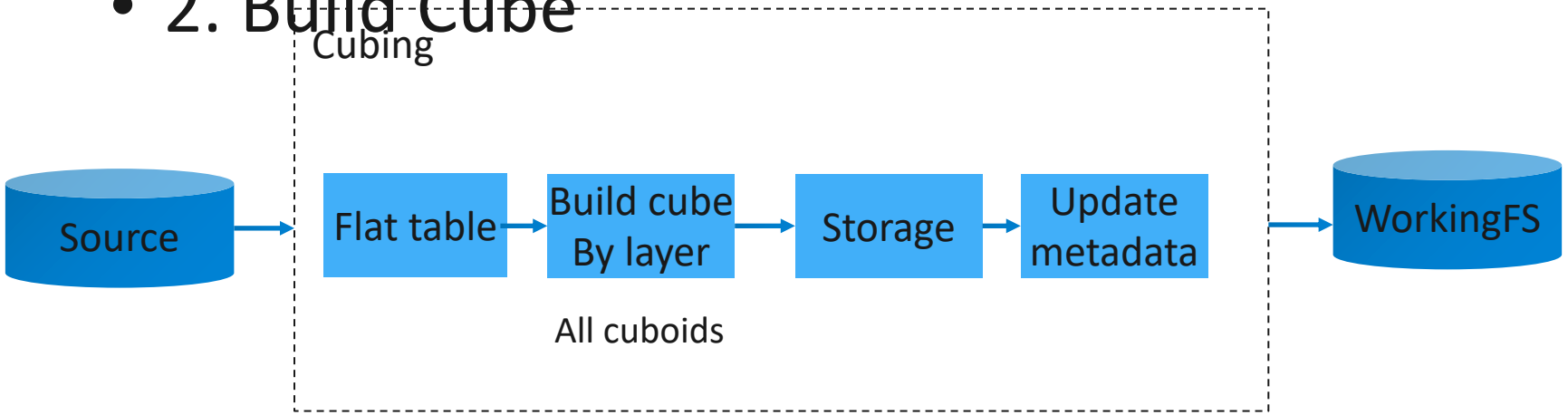


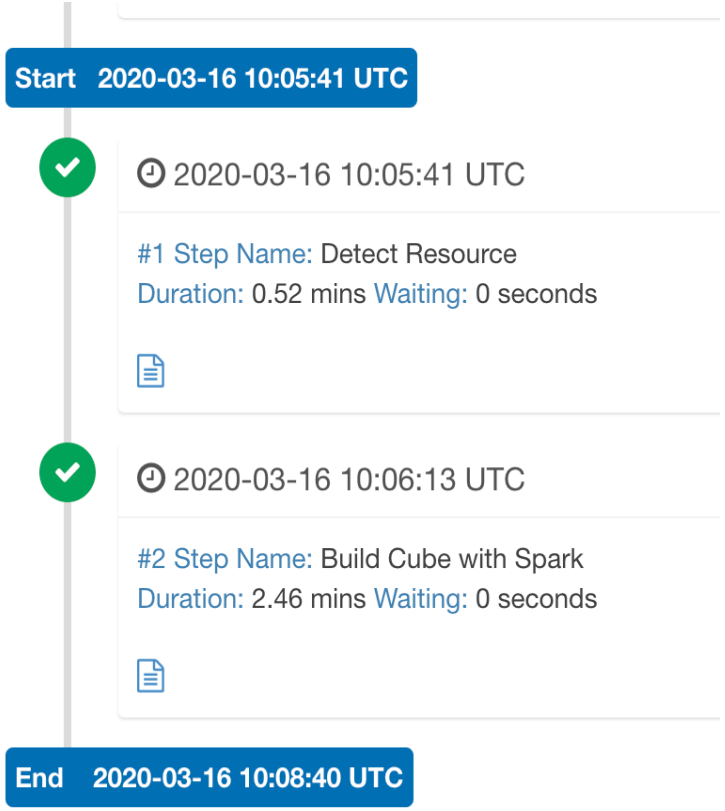



Kyligence

# Cube Build - Steps

- 1. Detect Resource
  - Estimate source table size
  - Adaptively adjust spark parameters
  - Build Global Dictionary (if there is bitmap measure for non-integer column)
- 2. Build Cube






Kyligence

# Adaptively adjust Spark parameters

Adaptively adjust Spark parameters

- Turned on by default
- Only support cluster mode
- The priority of manual adjustment is higher than that of automatic adjustment (kylin.properties)
- Spark configuration parameters affected
  - spark.executor.memory
  - spark.executor.cores
  - spark.executor.memoryOverhead
  - spark.executor.instances
  - spark.sql.shuffle.partitions

# Cube Build - Global Dictionary

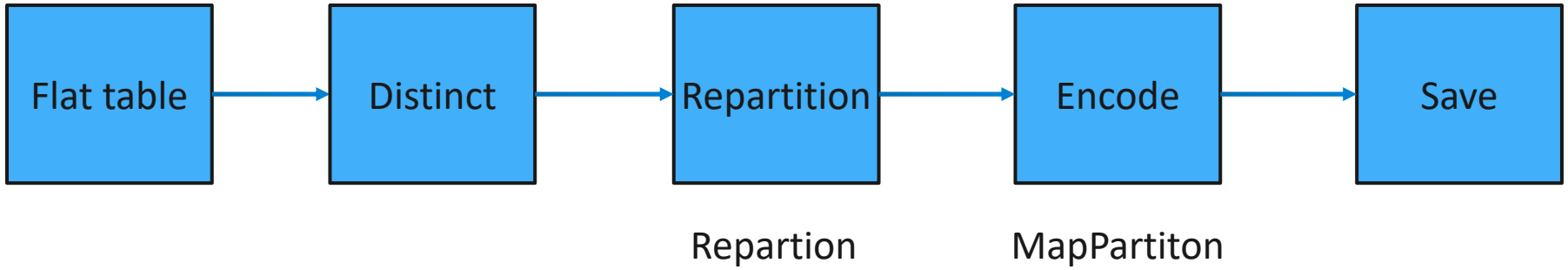
## Improvement

- Distributed generation in Spark
- Support cardinality > Integer.MAX_VALUE (Using Roaring64NavigableMap)

# Cube Build - Global Dictionary

## Steps

- Create flat table and extract distinct values
- Repartion RDD, DictionaryBuilderHelper.*calculateBucketSize()*
- MapPartiton RDD, DictHelper.*genDict()*
- Save dicts and metadata(bucket counts & offsets), NGlobalDictHDFSStore.*writeBucketDict()*

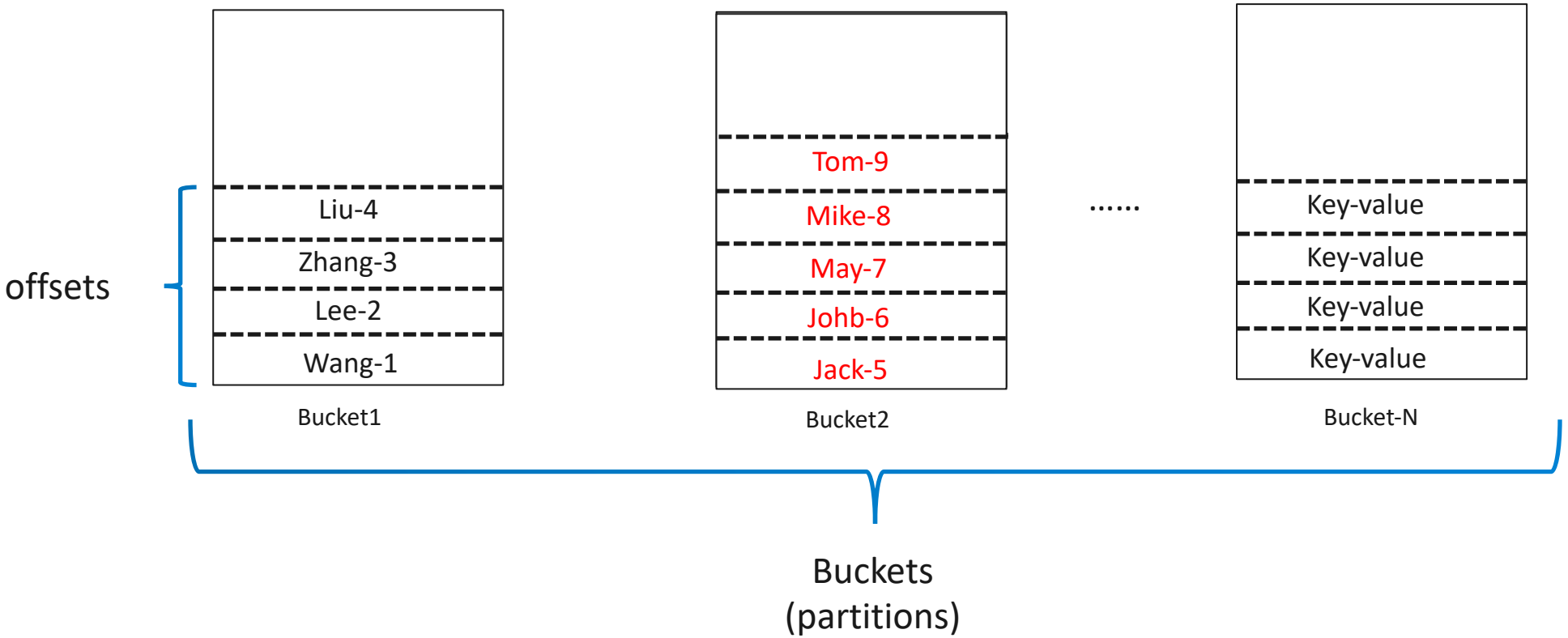





Kyligence

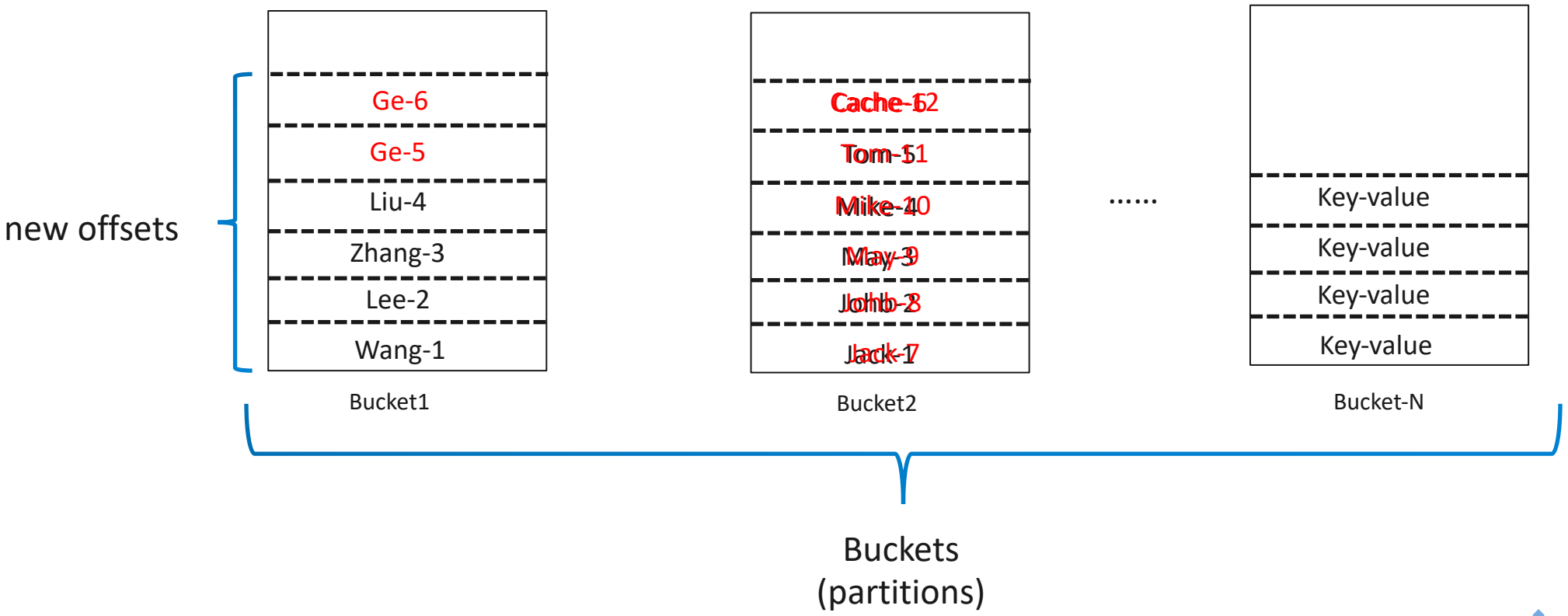
# Cube Build - Global Dictionary

## First build - version_1

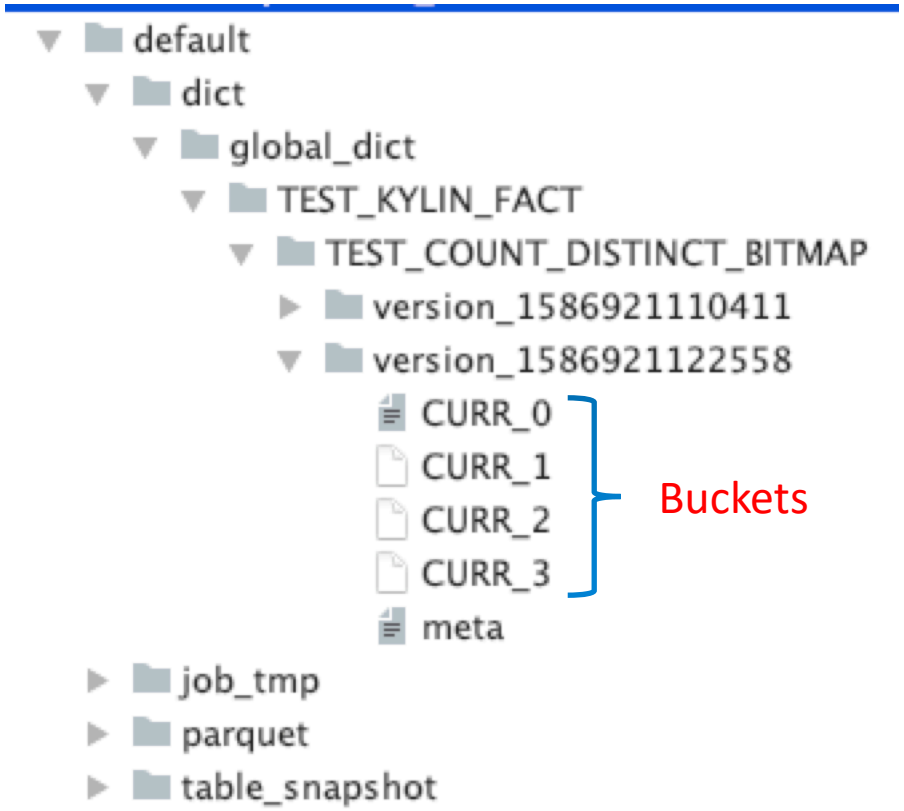


# Cube Build - Global Dictionary

## Second build - version_2



# Global Dictionary - Storage



# Auto retry failed job

Analyze the error message and take corresponding strategy

- OutOfMemoryError
  Disable *AUTO_BROADCASTJOIN_THRESHOLD* and retry
- ClassNotFoundException
  Abort job and throw exception
- Other exceptions
  Adjust cores and memory of executors and resubmit the job

By default retry 3 times, set by “kylin.engine.max-retry-time”

# Cube Build - Measures

Measures processed when cubing (CuboidAggregator.scala)

- Count
- Sum
- Max
- Min
- TopN
- Count Distinct(BitMap,HyperLogLog)
- Percentile

PS： Because of the compatibility of query engine, TopN, Count Distinct and Percentile are still unavailable. Issues have been opened!

# Cube Build - Storage

## Parquet storage

If there is a dimension combination of columns[id, name, price] and measures[COUNT, SUM], then a parquet file will be generated:

Columns[id, name, age] correspond to Dimension[2, 1, 0], measures[COUNT, SUM] correspond to [3, 4]

Cube Logical View

| Dimensions | | | Measures | |
|---|---|---|---|---|
| id | name | price | COUNT | SUM |
| 31 | Mobile | 5000.0 | 1765 | 332500 |
| 32 | Computer | 7000.0 | 2638 | 422300 |
| 33 | Camera | 900.0 | 1923 | 1023200 |
| 34 | Refrigerator | 2000.0 | 2516 | 167400 |

Parquet schema

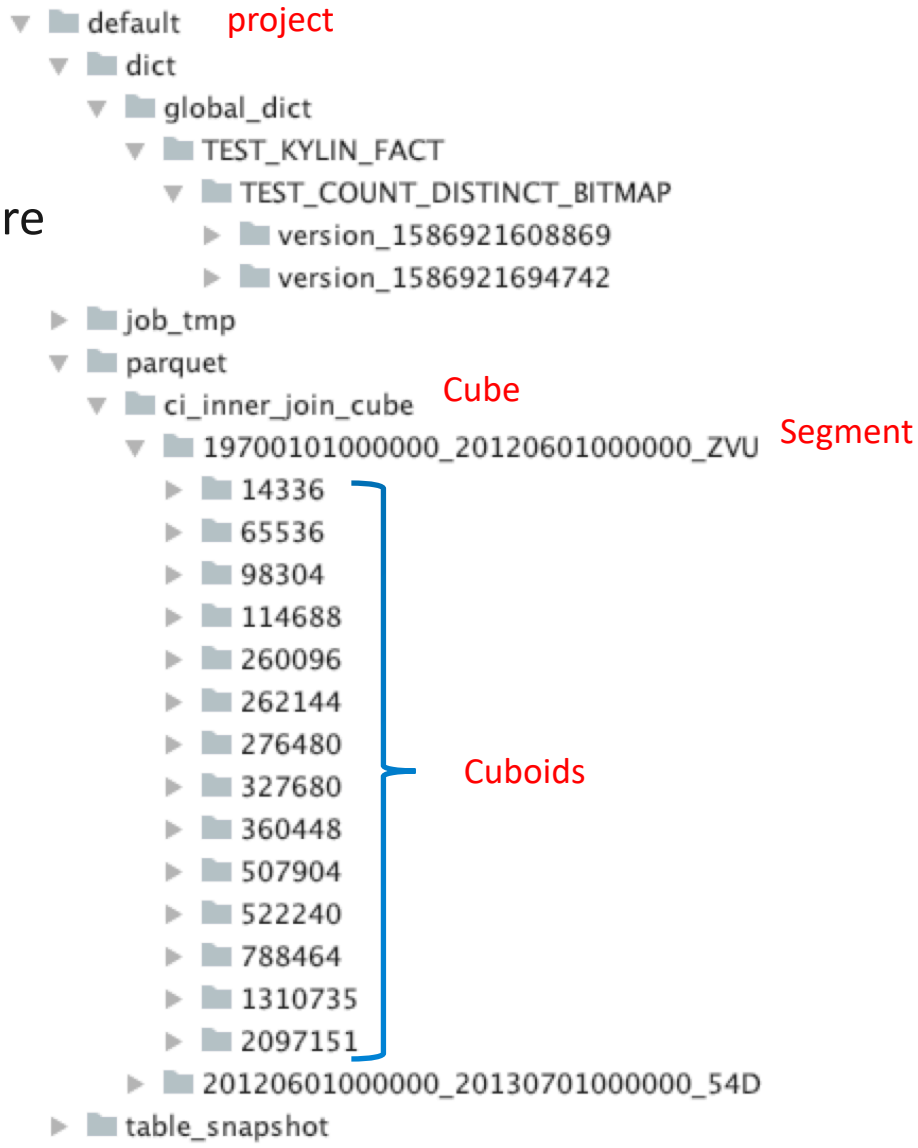
| | | |
|---|---|---|
| 2: | OPTIONAL | INTEGER |
| 1: | OPTIONAL | BINARY |
| 0: | OPTIONAL | DOUBLE |
| 3: | OPTIONAL | LONG |
| 4: | OPTIONAL | LONG |

```
scala> val f = spark.read.parquet(path)
f: org.apache.spark.sql.DataFrame = [2:

scala> f.printSchema
root
 |-- 2: integer (nullable = true)
 |-- 1: string (nullable = true)
 |-- 0: double (nullable = true)
 |-- 3: long (nullable = true)
 |-- 4: long (nullable = true)
```

# Cube Build - Storage

- Isolated by project
- Segment with unique signature






Kyligence

# Cube Build - Performance

## Environment:

- 4-nodes Hadoop cluster
- YRAN has 400GB RAM and 128 cores in total
- CDH 5.11

## Apache Kylin version:

version 3.0.1

## Spark version:

spark-2.4.1-os-kylin-r3 (fork version)

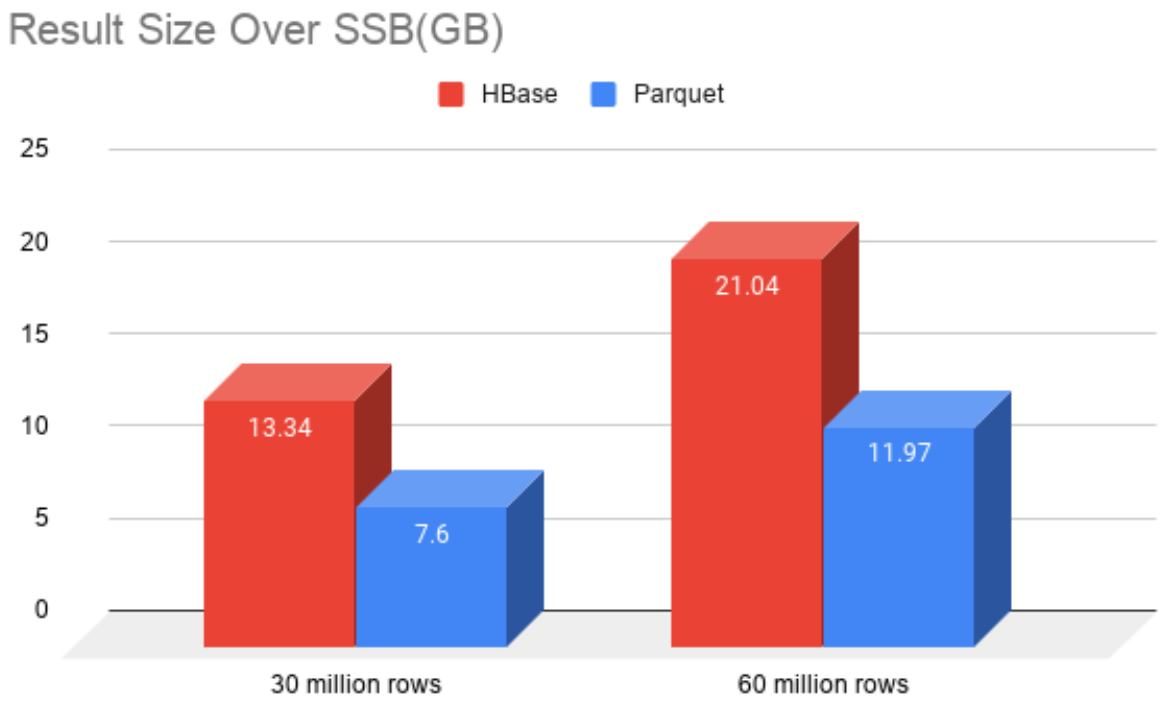
## Test Data:

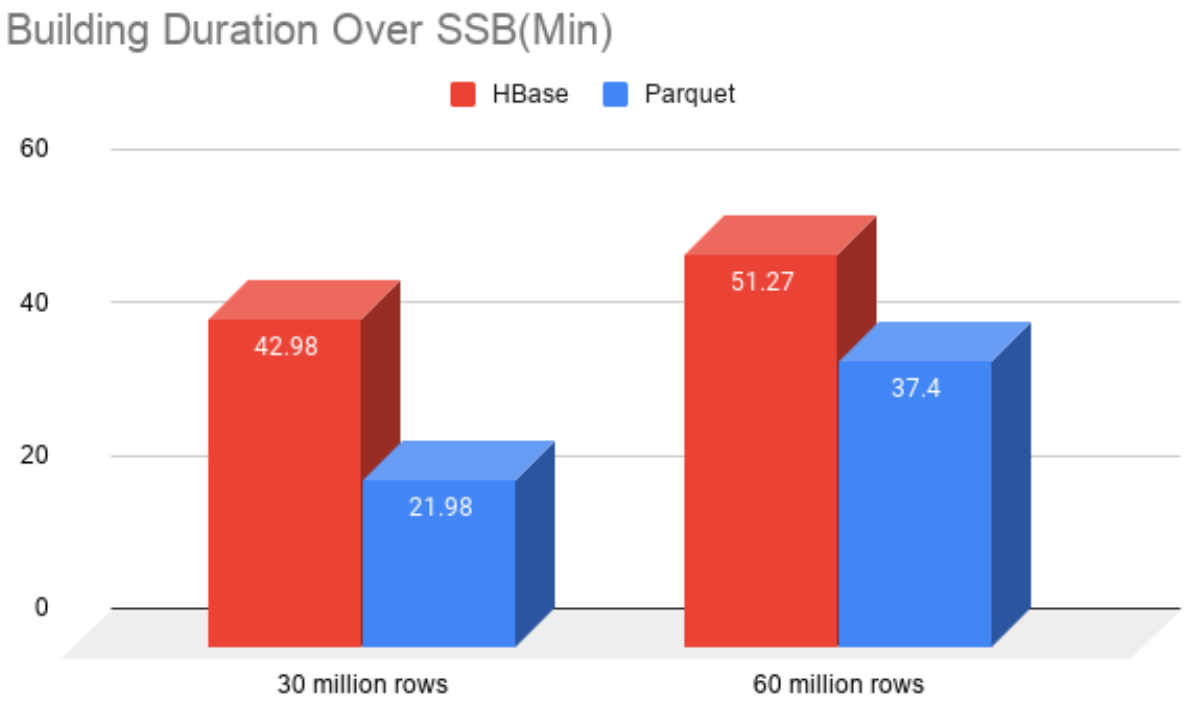
SSB data

Cube: 17 dimensions, 4 measures (COUNT, SUM)

# Cube Build - Performance

Kylin 3.0 MR engine VS Kylin 4.0 Parquet+spark engine






Kyligence

# Query

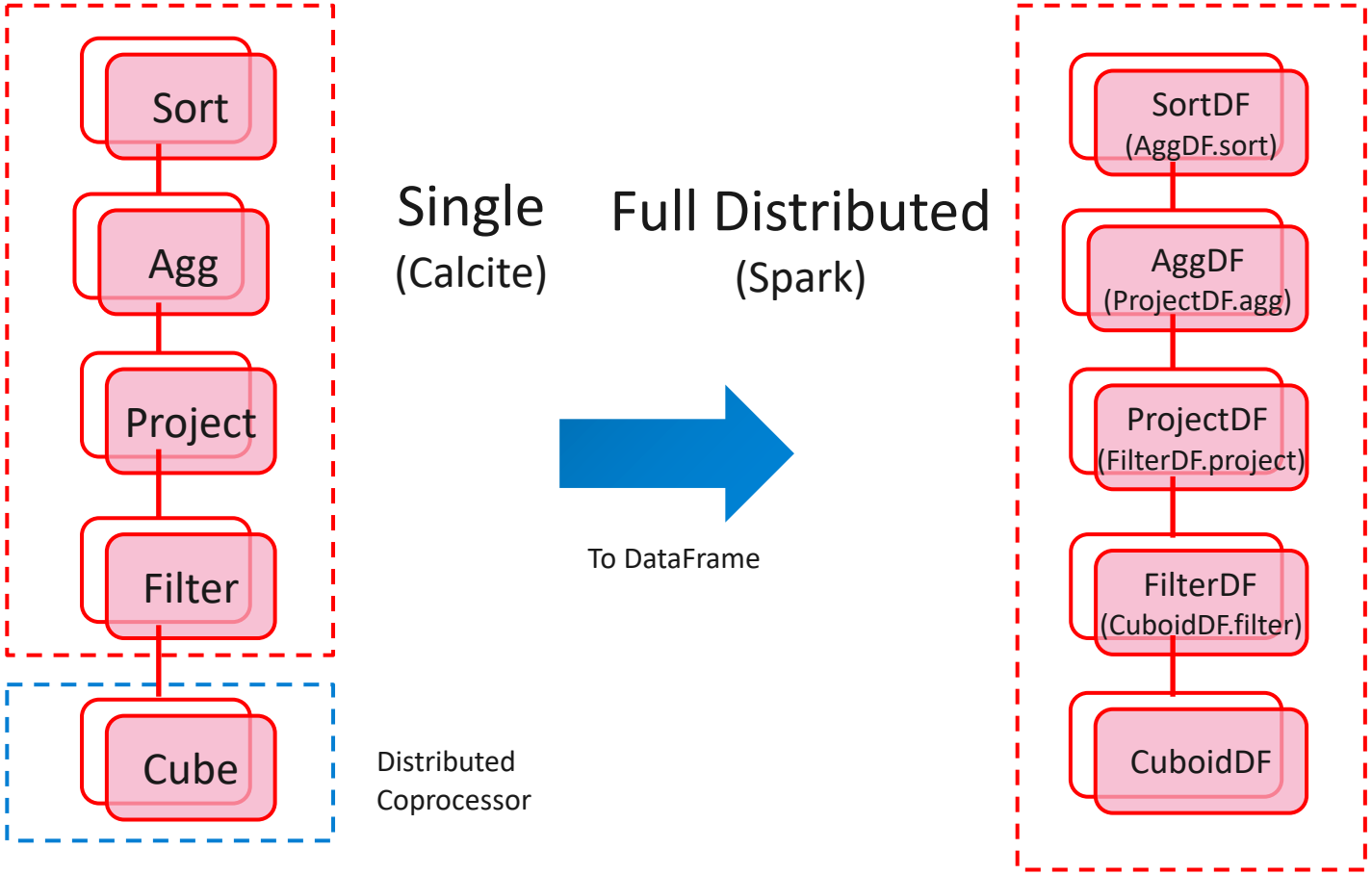
4

# Query - Steps

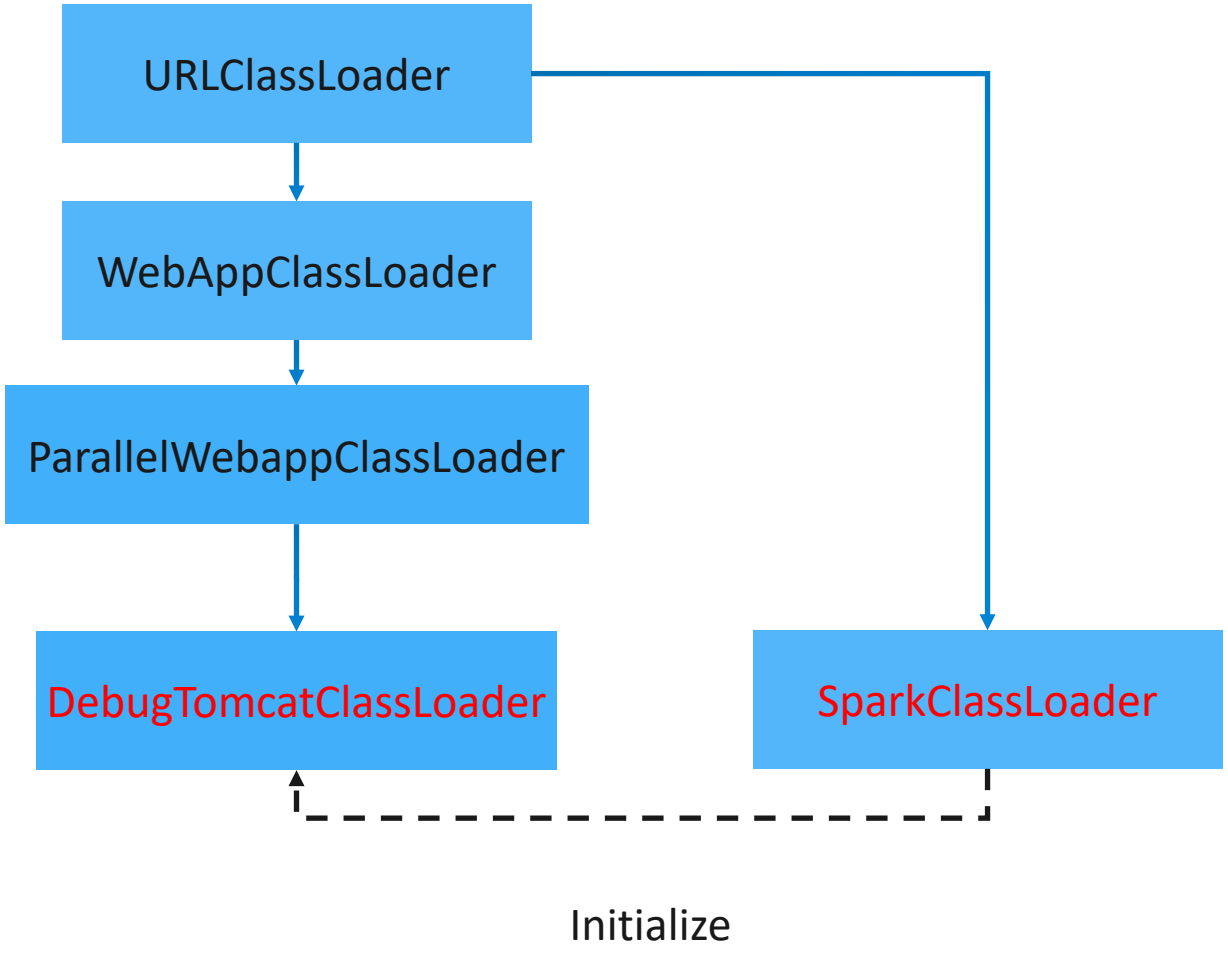
- Previous Process
  - Calcite analysis SQL to generate Abstract Syntax Tree(AST)
  - Calcite validate, optimize AST, trans to RelNodes
- Parquet query engine transform RelNodes to SparkPlan
- Distributed calculation and response

# Query - Steps

- Easy to to debug
- Get values in each Data Frame by adding breakpoint
- Spark-context on YARN (Lazy initial)

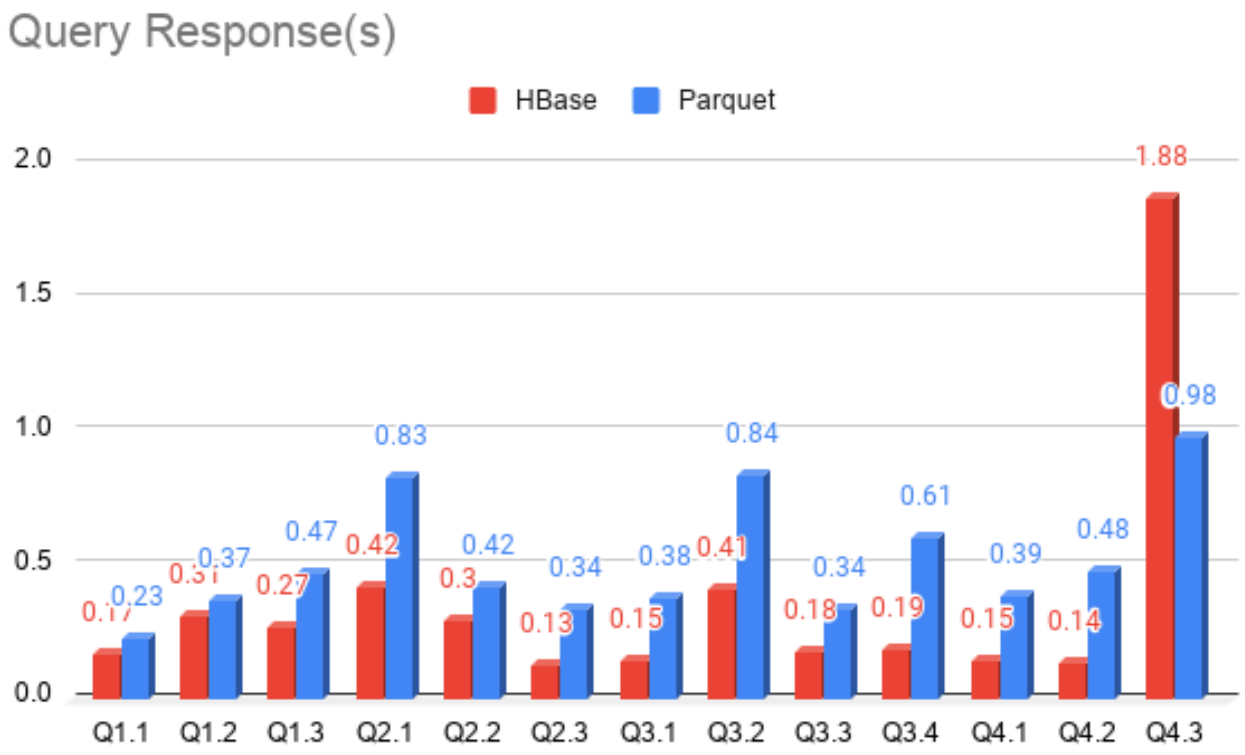


# Query - Dependency Isolation



# Query - Performance

Over SSB – 60 million rows (lower is better)

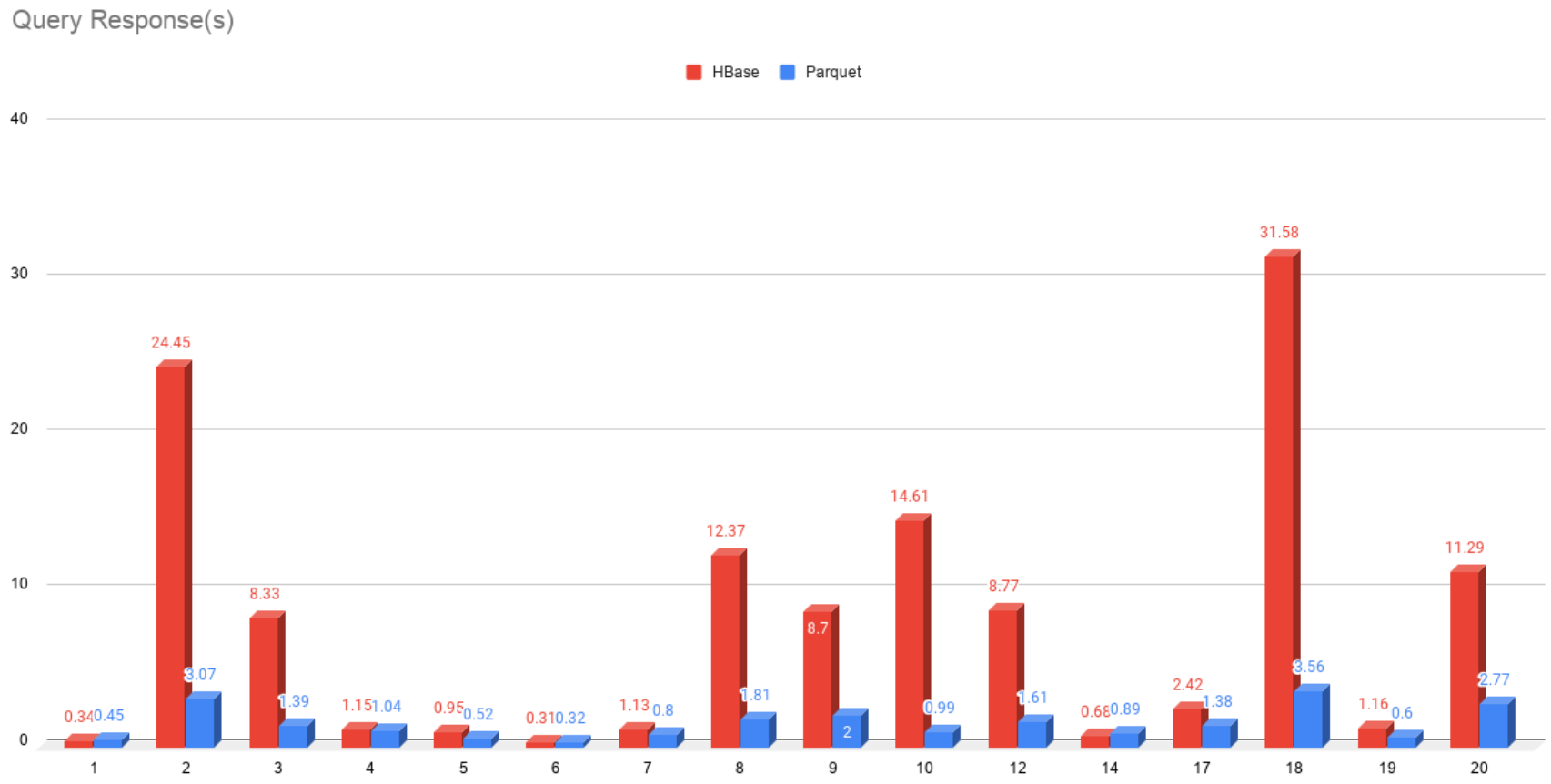





Kyligence

# Query - Performance

Over TPCH – 12 million rows (lower is better)






Kyligence

# Live Demo

5

# Live Demo

## Repository (temporary, will move to Apache Kylin soon):

https://github.com/Kyligence/kylin-on-parquet-v2

## Document (temporary)

https://github.com/Kyligence/kylin-on-parquet-v2/wiki

## Debug (Local mode & Tomcat):

https://github.com/Kyligence/kylin-on-parquet-v2/wiki/Development-document

## Build a Package:

${KYLIN_HOME}/build/dist/package.sh
${KYLIN_HOME}/build/dist/download-spark.sh

## Test：

mvn test

# Live Demo

Run/Debug Configurations

Application
DebugTomcat
JUnit
Remote
Templates

Name: DebugTomcat
Share
Allow parallel run

Configuration | Code Coverage | Logs

Main class: org.apache.kylin.rest.DebugTomcat

VM options: -Dspark.local=true

Program arguments:

Working directory: $MODULE_WORKING_DIR$

Environment variables:

Redirect input from:

Use classpath of module: kylin-server

Include dependencies with "Provided" scope

JRE: Default (1.8 – SDK of 'kylin-server' module)

Shorten command line: user-local default: none – java [options] classname [args]

Enable capturing form snapshots

Before launch: Build, Activate tool window

Cancel | Apply | OK

# Source modules

- docker
- engine-flink [kylin-engine-flink]
- engine-mr [kylin-engine-mr]
- engine-spark [kylin-engine-spark]
- examples
- jdbc [kylin-jdbc]
- job
- kylin-it
- kylin-spark-project
  - kylin-spark-classloader
  - kylin-spark-common
  - kylin-spark-engine
  - kylin-spark-metadata
  - kylin-spark-query
  - kylin-spark-test
  - target
  - kylin-spark-project.iml
  - pom.xml
- metrics-reporter-hive [kylin-metrics-reporter-hive]
- metrics-reporter-kafka [kylin-metrics-reporter-kafka]
- odbc
- parquet-assembly
- query [kylin-query]
- server [kylin-server]
- server-base [kylin-server-base]
- source-hive [kylin-source-hive]
- source-jdbc [kylin-source-jdbc]

# TODO

6

# TODO List

- More Measures (TopN, CountDistinct, Percentile)
- Cube Planner
- System Cube
- JDBC data source (to be discussed)
- File Pruning with shard by columns

Summary of features currently supported by Kylin on Parquet
https://github.com/Kyligence/kylin-on-parquet-v2/issues/156

# How to contribute?

## We need your Star, and PRs!

### Repository:

https://github.com/apache/kylin
https://github.com/Kyligence/kylin-on-parquet-v2

### Issues:

https://issues.apache.org/jira/browse/KYLIN-4188
https://github.com/Kyligence/kylin-on-parquet-v2/issues

WeChat Group

4.18 Kylin on Parquet线上分享②

该二维码7天内(4月29日前)有效，重新进入将更新



Kyligence

Kyligence

THANK YOU