TCloud Computing 天云趋势

BigData in Real-time

Impala Introduction

TCloud Computing 天云趋势

孙振南

zhennan_sun@tcloudcomputing.com

2012/12/13

Beijing

Apache Asia Road Show

云基地创造

Created By Cloud Valley

Background (Disclaimer)

- Impala is NOT an Apache Software Foundation project yet
- Impala uses ASLv2
- The speaker (me) is NOT associated with Cloudera

WHY
Impala

The Need For Speed

- **What's wrong with MapReduce?**
  - Batch oriented. Good at complex jobs
  - But slow at startup & shuffle
  - Programmer friendly
- **How about Hive?**
  - SQL friendly. Still slow as …
- **How about HBase?**
  - Slow data import
  - No SQL

The leads from the leader

- **Google BigQuery, the service**
  - Based on Google Dremel, the paper
    - SQL-like interface
    - Interactive analysis of PBs data
    - Query Execution Tree
      - Tasks to sub-tasks, instead of identical distributed tasks
    - Columnar storage based on nested ProtoBuffer data
      - Faster traversing
- **Amazon RedShift is another story…**

Open source alternatives?

- **Apache Drill**
  - No substantial progress
  - Mailing list msg # droped 80% from Sep to Nov
- **Berkeley Shark/Spark**
  - Shared memory based, good at iteration tasks
  - Different component stack
- **Cloudera Impala**

Positioning

- **Compared with MR, it's all about trade-off**
  - Complexity or responsiveness
  - General purpose or ad-hoc
- **MPP-RDB paradigms on top of commodity DFS**
  - On par performance in some cases
  - Extremely cheap
  - Linear scalability

WHAT

is

Impala

Features

- **Distributed SQL on raw HDFS files**
  - Select, where, aggregation, join,
  - Insert into/overwrite
  - Text and Sequence files
- **Hive compatible "meta store" and interface**
  - Reuse Hive's metadata schema, DDL and JDBC/ODBC driver
- **Up to 90x times faster, compared with Hive**
  - Purely I/O bound scenario, 3-4X
  - With joins, 7-45X
  - With memory cached, 20-90X

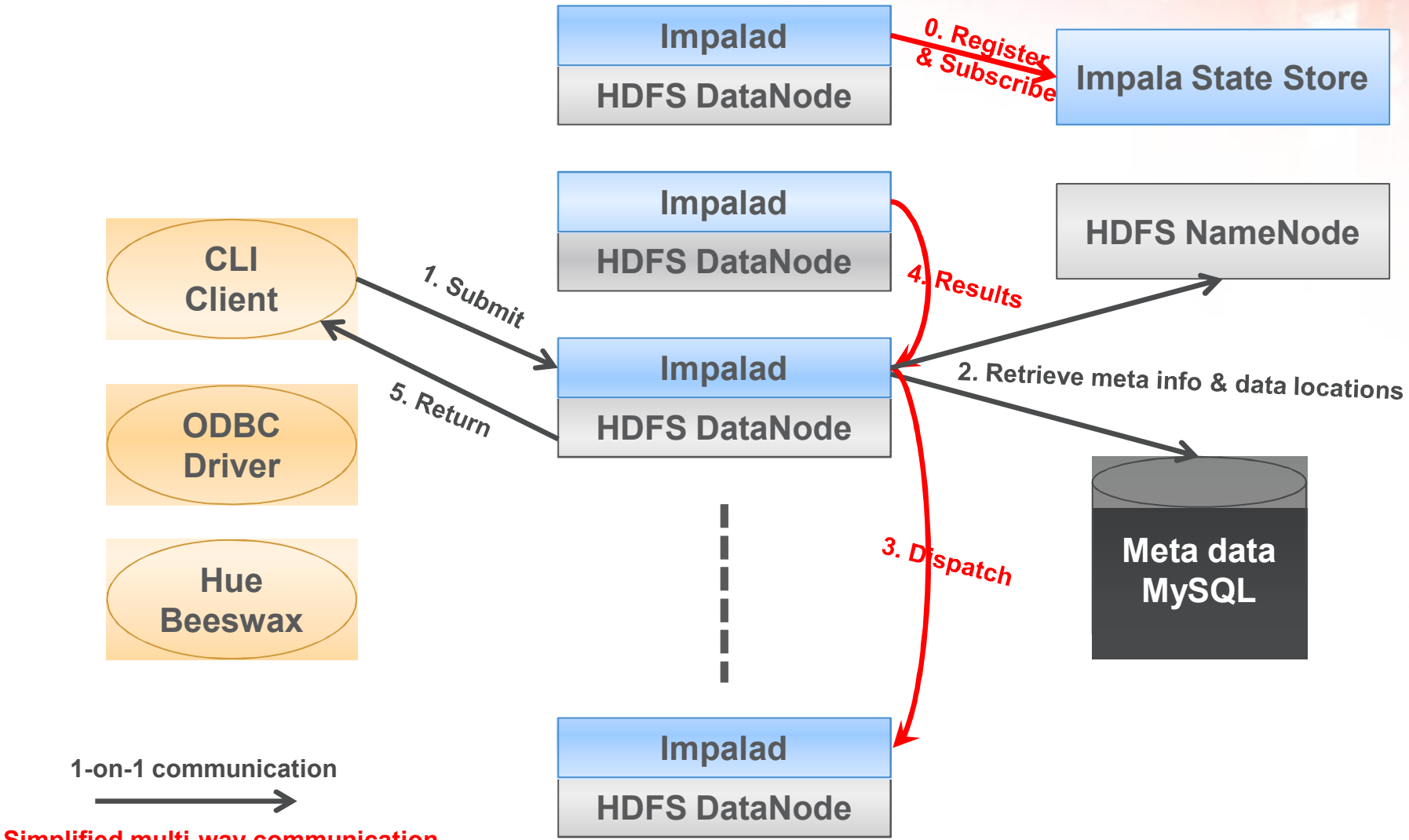
Status

- **Announced at Oct/2012**
  - Now 0.3 at Dec/5
  - Has been private beta for half-year
  - Currently in public beta
  - Target GA @ 2013 Q1
- **Entirely developed by Cloudera (by now)**
  - In the past 2 years, 7 full time engineers
- **Completely open source, ASLv2**

Example

```
$ hive

hive> CREATE TABLE sales (id STRING, item STRING, price int);

hive> load data local inpath '/store/sales.txt' into table sales;

$ impala-shell -impalad=172.16.204.4:21000

[172.16.204.4:21000]> show tables

sales

[172.16.204.4:21000]> select * from sales where price > 100 limit 3

13221   COOKIE  138

38384   DIPER   287

85845   TV      737
```

Workflow Overview



HOW

the Impala speed up

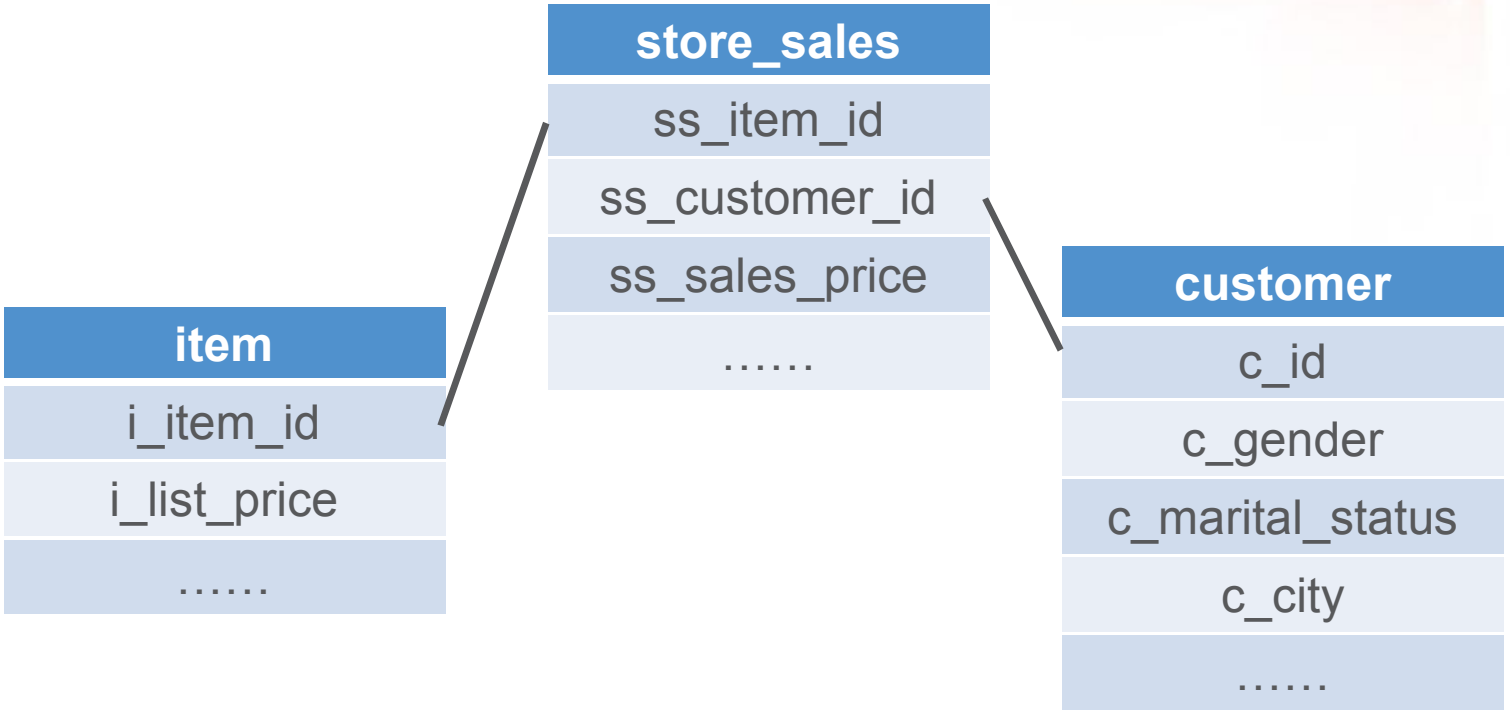
‘MPP’ SQL

- **PlanNode**
  - Node of the Depth-First execution plan tree
  - Various types
    - HDFS_SCAN_NODE, HBASE_SCAN_NODE
    - HASH_JOIN_NODE
    - AGGREGATION_NODE, SORT_NODE
    - EXCHANGE_NODE
- **Fragment**
  - Atomic executable unit, could be distributed
  - Contains one or more PlanNodes
    - Depends on the data distribution and the SQL statement

SQL breakdown sample

- **There's a saying that young single males don't use coupon or discount as much as others, is it true?**
- **We can compare the list price and sales price**
  - Items are a little bit expensive
  - Buyers are young, single, male
  - Live in major city

SQL breakdown sample - tables

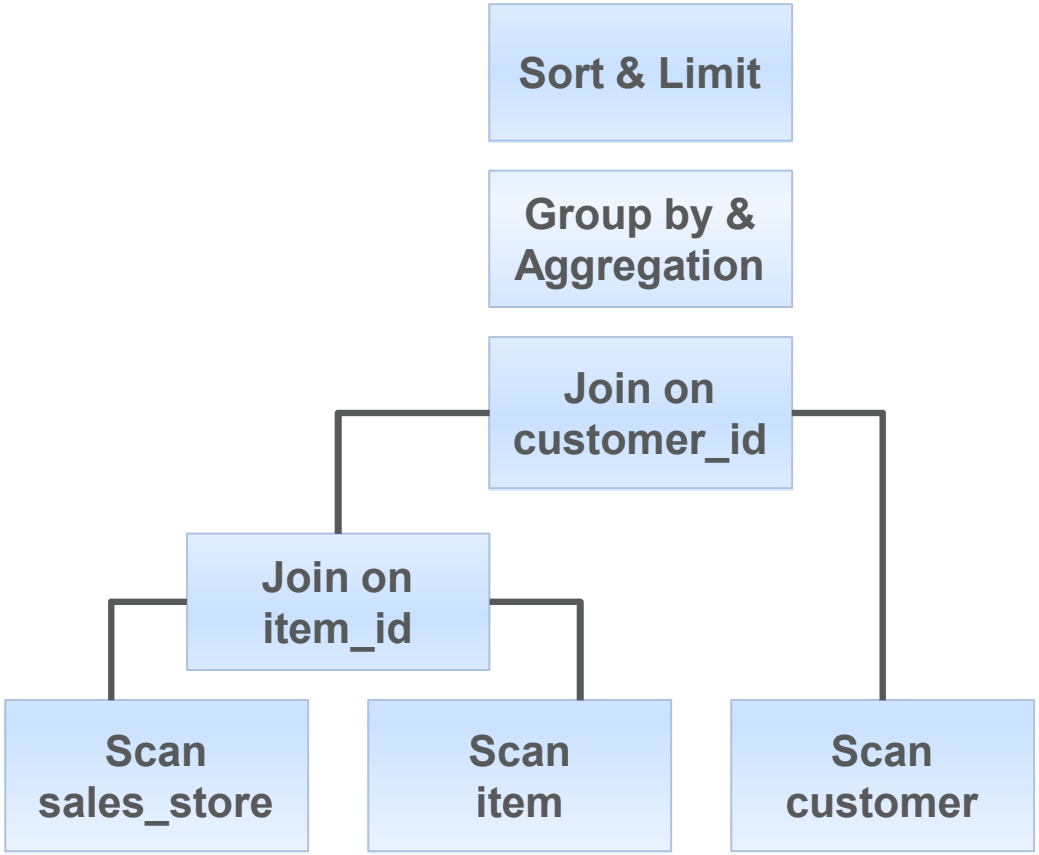


SQL breakdown sample – SQL statement

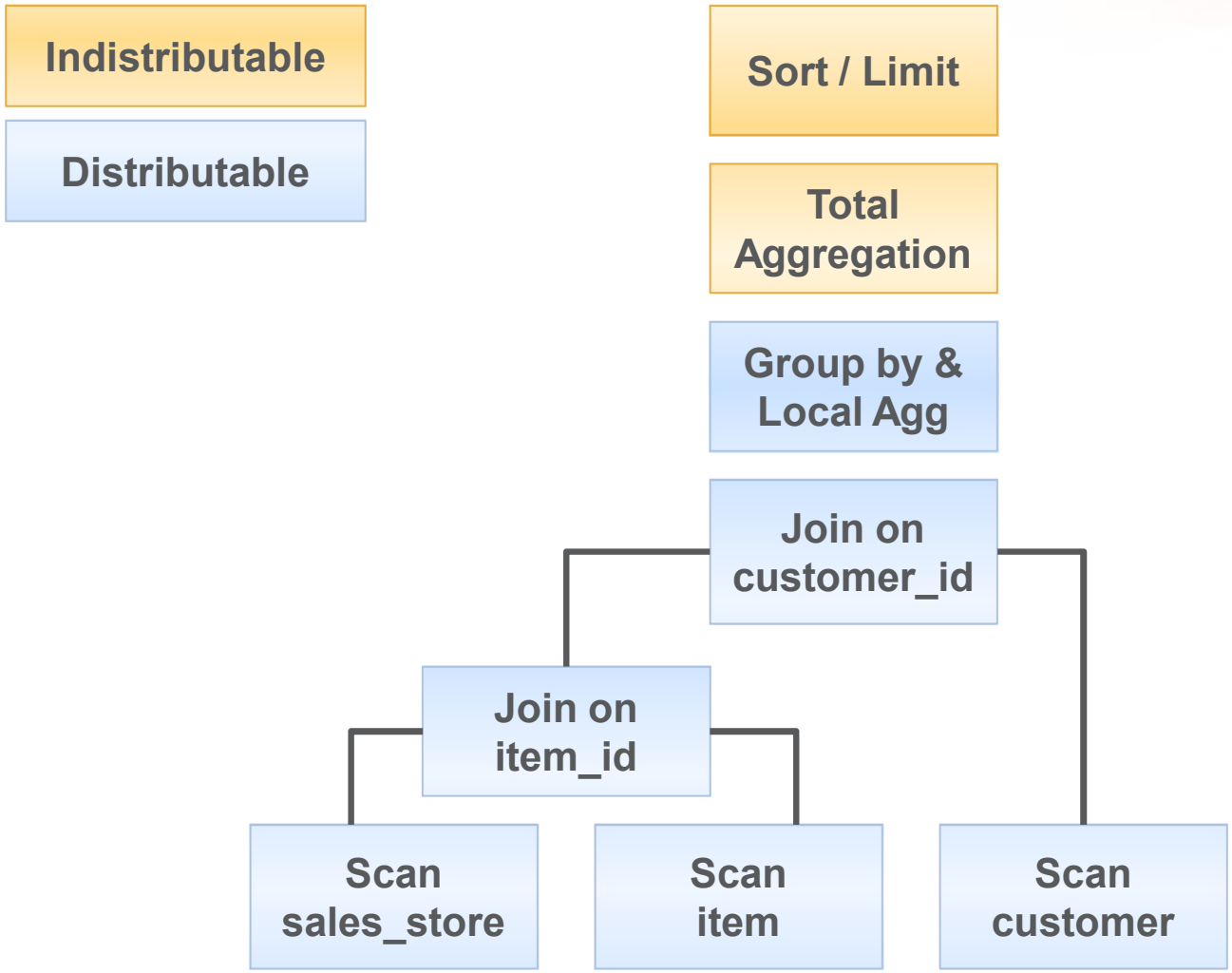
TCloud Computing 云基地创造 Created By Cloud Valley

```
select i_item_id, i_list_price, avg(ss_sales_price) agg1

  FROM store_sales

  JOIN item on (store_sales.ss_item_id = item.i_item_id)
  JOIN customer on (store_sales.ss_customer_id = customer.c_id)

  where

       i_list_price > 1000 and
       c_gender = 'M' and
       c_marital_status = 'S‘ and

       c_city in ('Beijing','Shanghai','Guangzhou')

  group by i_item_id,

  order by i_list_price

  limit 1000
```

Execution plan tree

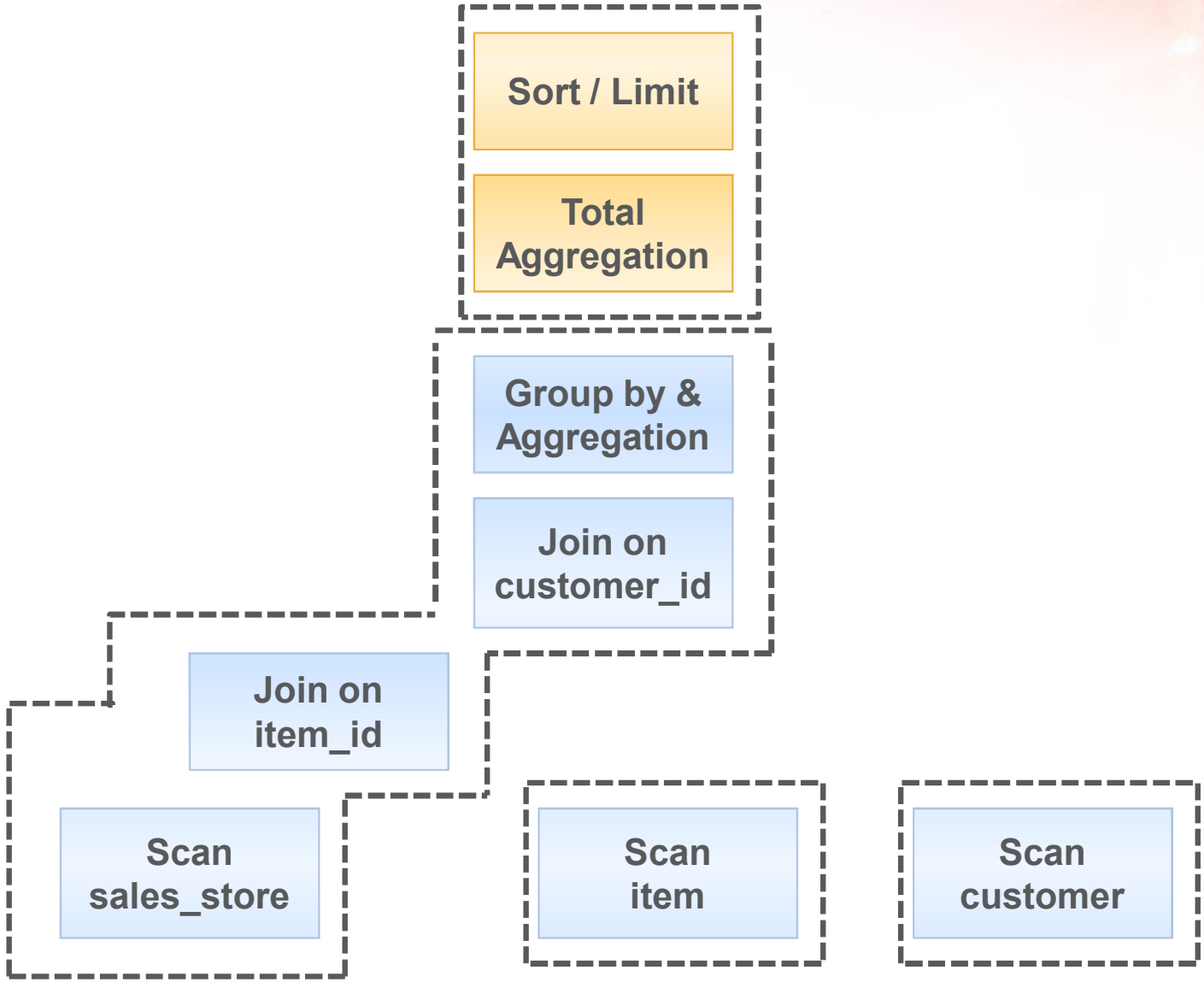


Execution plan tree – Distributed!

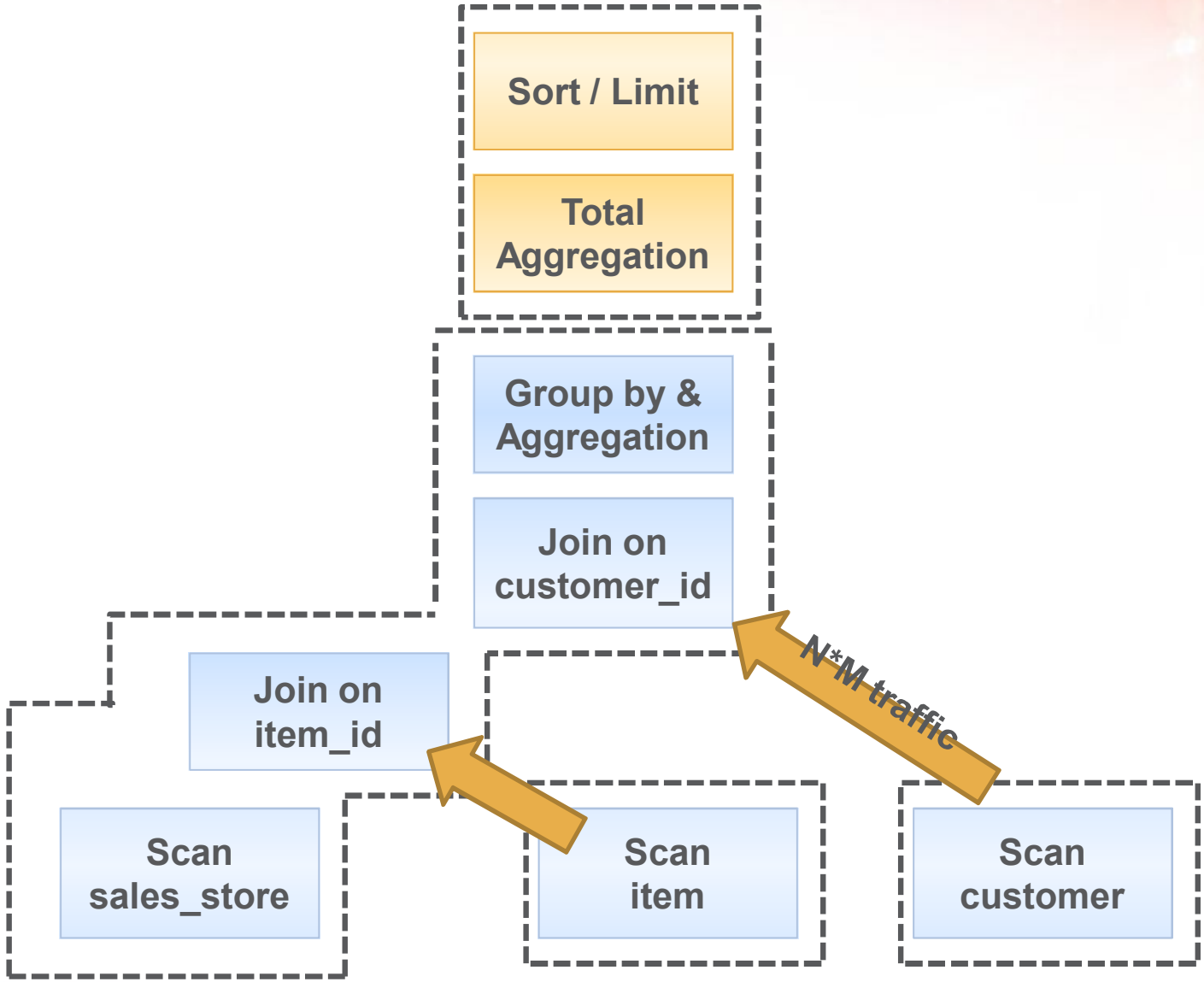


From execution plan to fragment

From execution plan to fragment



From execution plan to fragment



THAT'S IT?

There's more…

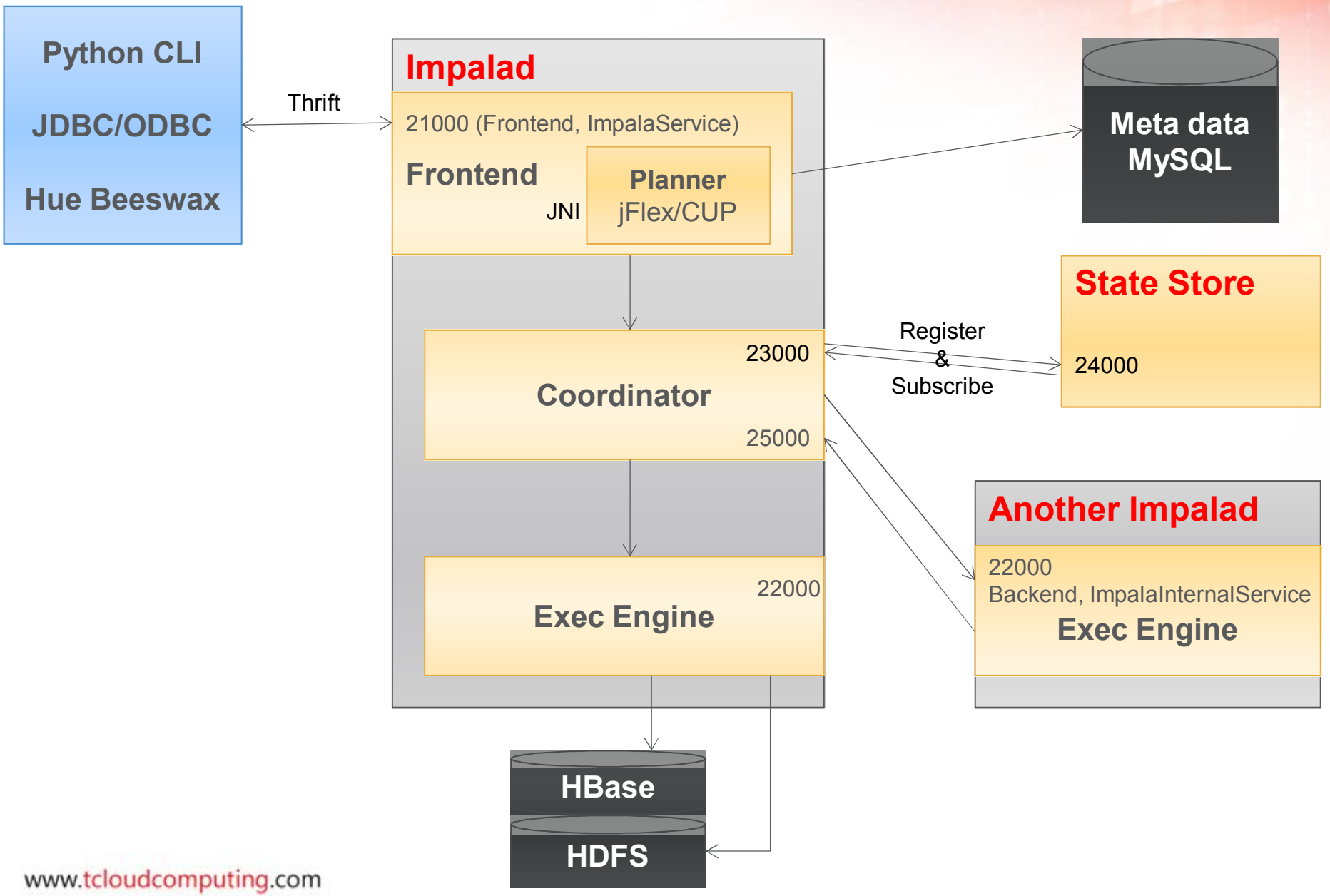
- **Written in C++**
  - Only used jFlex/CUP to parse the SQL statement
- **Local compilation of fragments**
  - LLVM is used
- **Disk awareness**
  - Not just host awareness
  - dfs.datanode.hdfs-blocks-metadata.enabled
  - "40% faster"
- **Direct read**
  - Not via HDFS NameNode then DataNode then …
  - dfs.client.read.shortcircuit, dfs.client.read.shortcircuit.skip.checksum

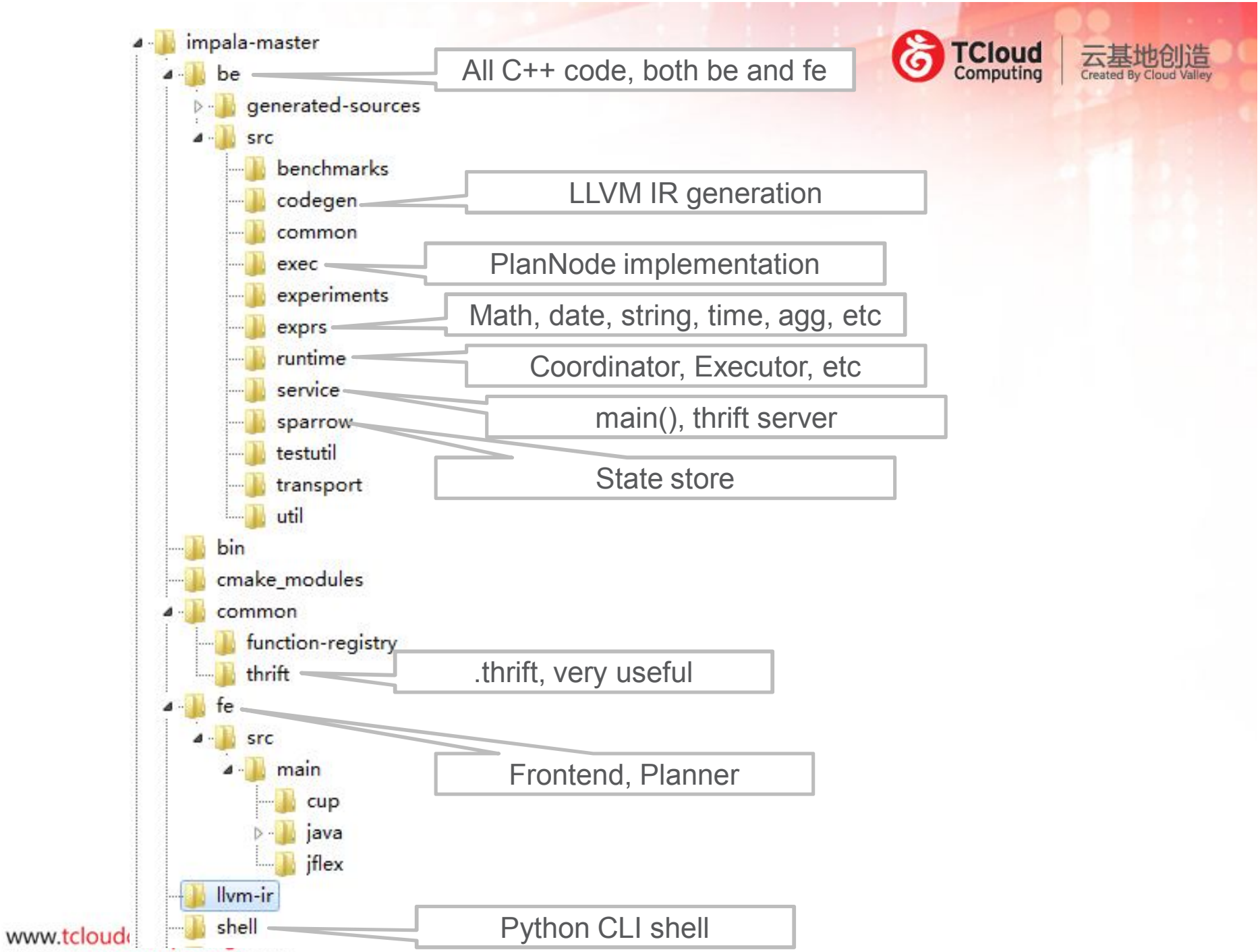
GOOD ENOUGH?

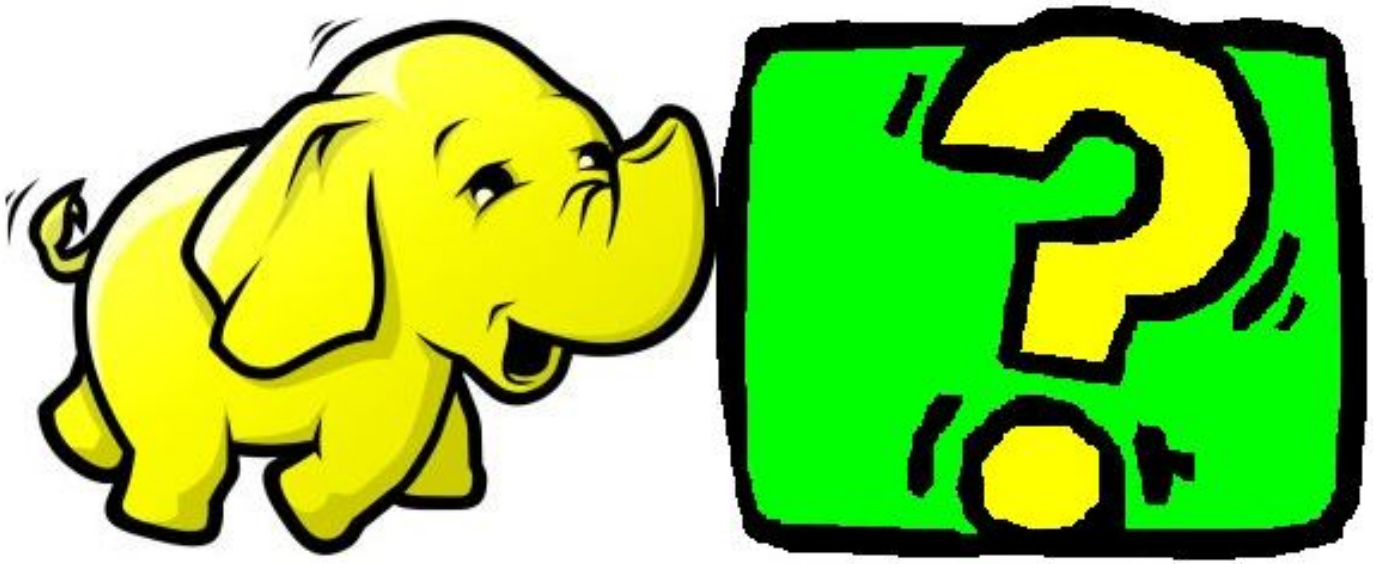
TODOs

- No Data Definition Language yet
- No User Defined Function yet
- No fault tolerance yet
- Avro, RCFile, LZO, Trevni support is on the way
  - Impala + Trevni will introduce another performance boost
    - With more SQL functions compared with BigQuery
- In memory Join only
  - Will be fixed in GA
- Partition before join, reduce traffic
- Support Hive partitions, but not buckets yet

INSIDE


Python CLI
JDBC/ODBC
Hue Beeswax
Thrift
Impalad
21000 (Frontend, ImpalaService)
Frontend
JNI
Planner
jFlex/CUP
Meta data
MySQL
State Store
24000
Register
&
Subscribe
23000
Coordinator
25000
Exec Engine
22000
Another Impalad
22000
Backend, ImpalaInternalService
Exec Engine
HBase
HDFS





新浪微博 @少年振南