

Big Telco, Bigger DW Demands: Moving Towards SQL-on-Hadoop

Keuntae Park



- IT Manager of SK Telecom, South Korea's largest wireless communications provider
- Work on commercial products (~'12)
 - T-FS: Distributed File System
 - Windows compatible layer on TimOS
 - T-MR: on-demand MapReduce service like E-MR
- Open source activity ('13~)
 - Committer of Apache Tajo project



Overview

- Background
 - Telco requirements
- Before Tajo
 - Commercial product
 - Open source (Hadoop) outsourcing
- After Tajo
 - Issues & solutions
 - Performance
- win-win between community and company
- Future Works



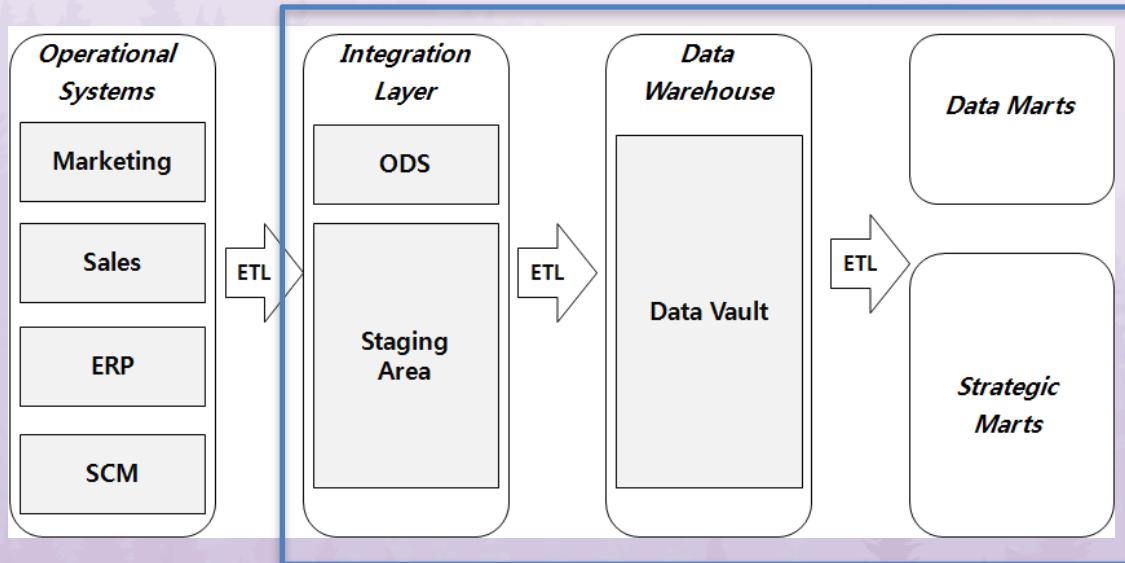
Telco data characteristics

- Huge amount of data
 - 40 TB/day (compressed)
 - 15 PB (estimated, end of 2014)
- Report & OLAP ad-hoc query
 - Filtering
 - Summary
 - BI tools

Requirements - different size, different speed

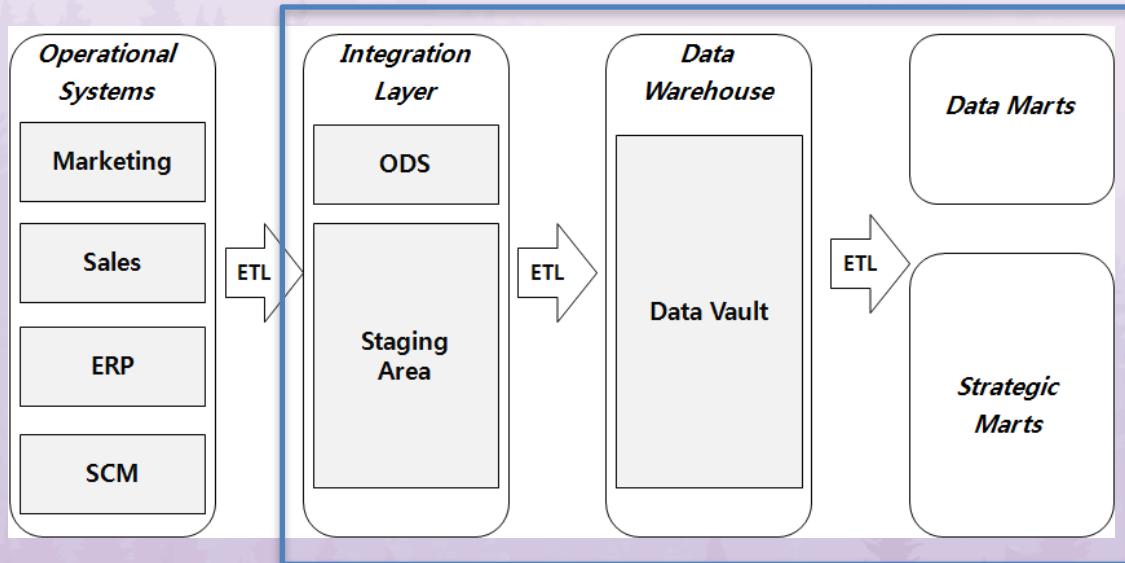
	<i>Filtering & aggregation</i>	<i>Summary</i>	<i>Data re-construction</i>	<i>BI report</i>	<i>Ad-hoc Query</i>
<i>Target</i>	accumulated for 5 minutes	daily sum of filtered data	entire summary data	<i>mart data</i>	<i>summary data</i>
<i>Frequency</i>	every 5 minutes	daily or monthly	non-regularly (rare)	<i>ah-hoc</i>	<i>ah-hoc</i>
<i>Amount of data</i>	terabytes	hundreds of terabytes	petabytes	tens of gigabytes	tens of terabytes
<i>Response time</i>	within a minute	within a hour	no strict deadline	within two seconds	within a hour

Previous approach - DBMS



based on MPP DBMS

Previous approach - DBMS



based on MPP DBMS

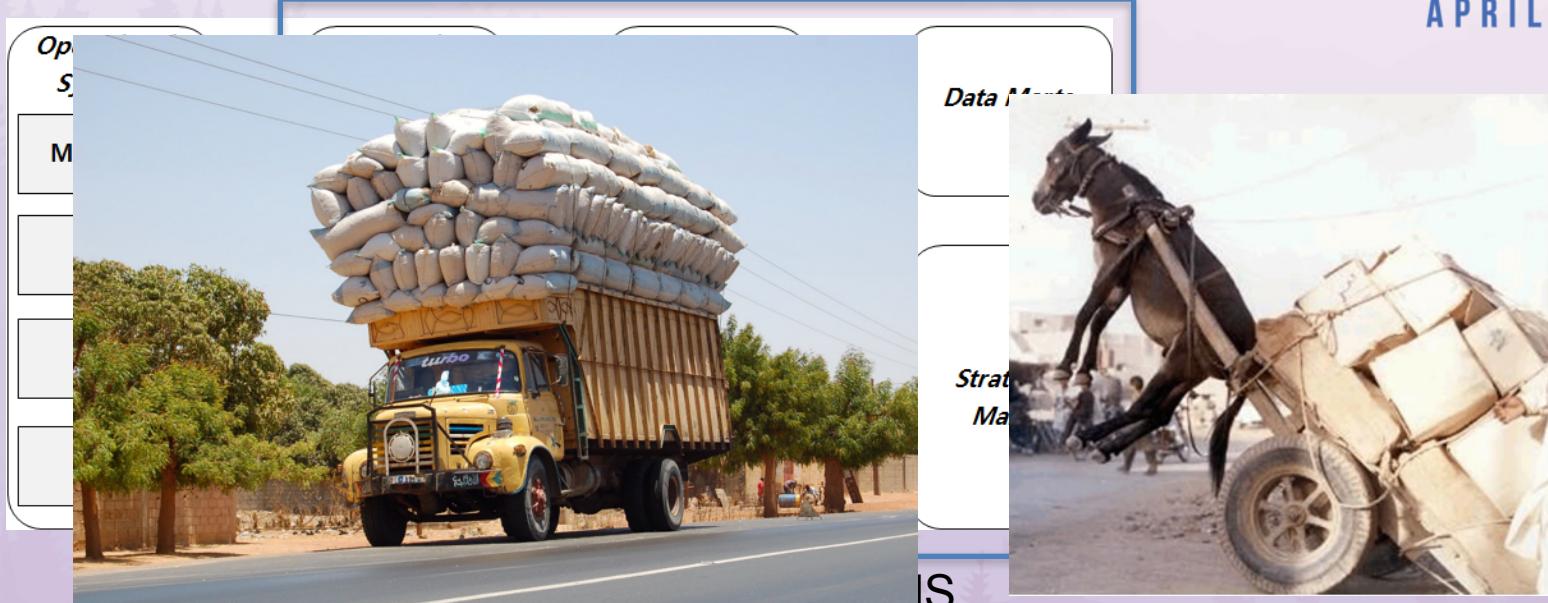
Too Expensive
Not Scalable

Previous approach - DBMS



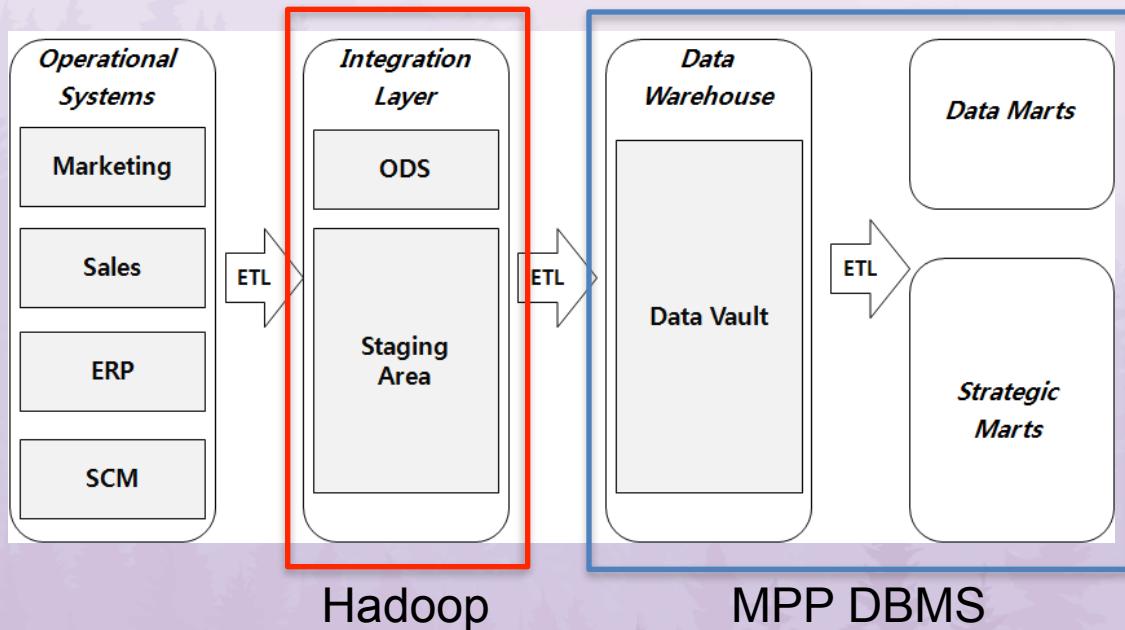
Too Expensive
Not Scalable

Previous approach - DBMS

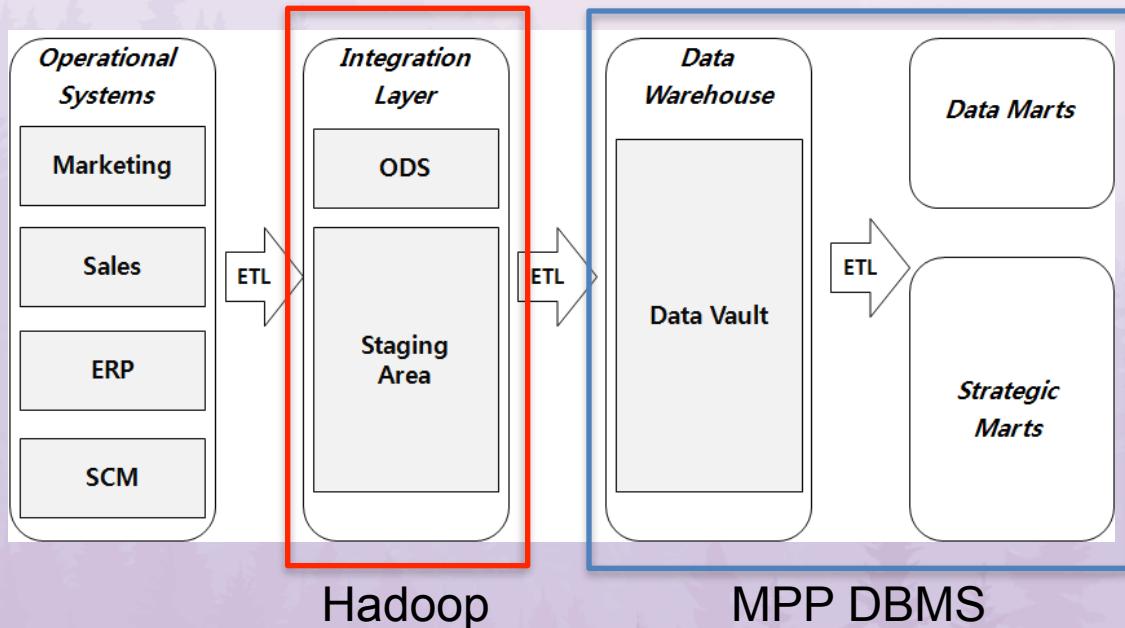


sive
ble

Previous approach - Hadoop(MapReduce, Hive) + DBMS



Previous approach - Hadoop(MapReduce, Hive) + DBMS



Working
(but...)

Still has Problems

- Hadoop outsourcing
 - quality of outcome is not good (actually bad)
 - communication overhead
 - hard to reflect requirements on open source
- Data Warehouse and Mart becomes bigger

Solution - Tajo!!

- It can replace both DBMS and Hadoop
 - High throughput for batch processing
 - Low latency for ad-hoc queries
 - ANSI SQL compatible
- Can do by myself
 - very open community
 - easily make issues about what I really need
 - fast growing
 - issues solved very fast

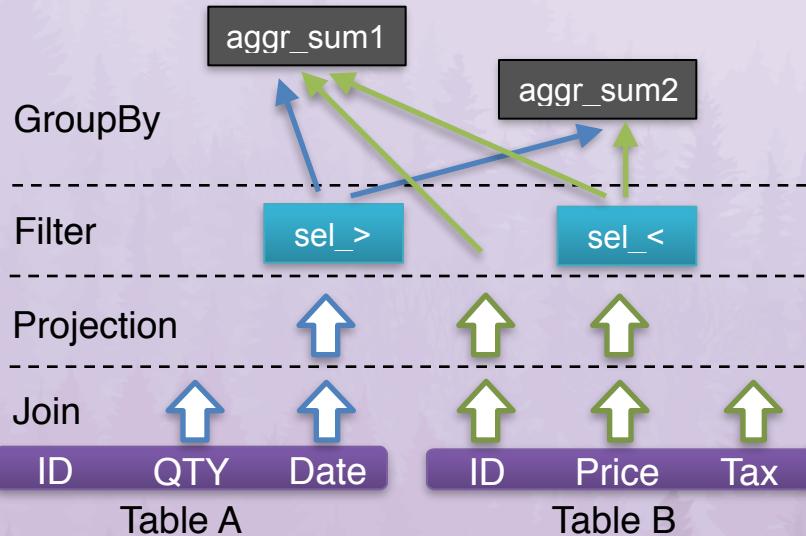
About Tajo

- Tajo (since 2010)
 - Big Data Warehouse System on Hadoop
 - Apache top-level project (entered the ASF in March 2013)
- Features
 - SQL standard compliance
 - Fully distributed SQL query processing
 - HDFS as a primary storage
 - Relational model (will be extended to nested model in the future)
 - ETL as well as low-latency relational query processing (100 ms ~)
- News
 - 0.2-incubating: released November 2013
 - graduation to top-level: April 2014



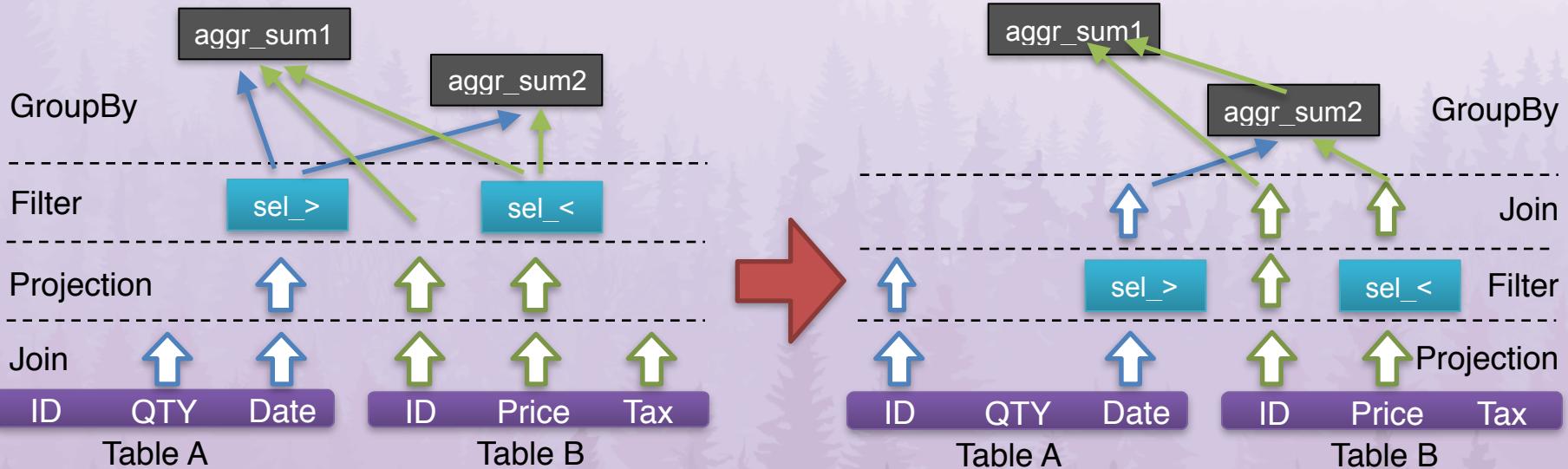
Tajo logical optimizer

- Cost-based join ordering
- Projection/Filter push down & Duplicated expression removal



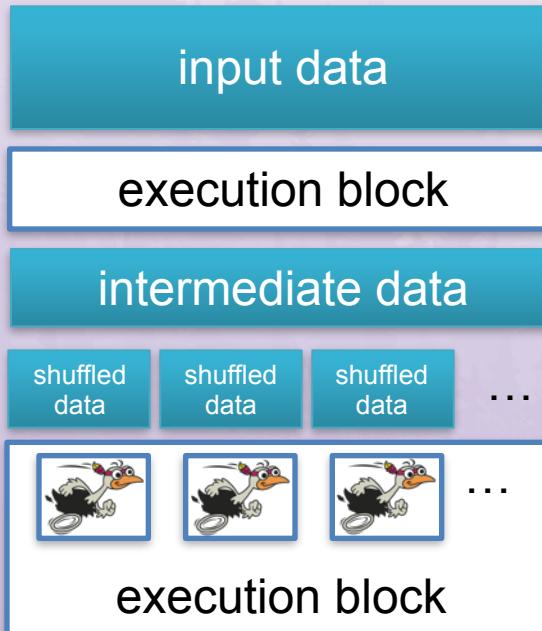
Tajo logical optimizer

- Cost-based join ordering
- Projection/Filter push down & Duplicated expression removal



Tajo progressive optimization

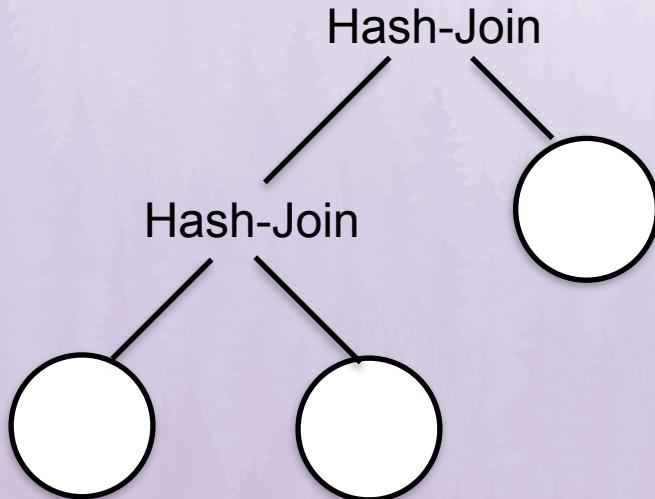
- dynamically adjust number of tasks



- estimate data size at planning time
- check size and adjust plan at execution time
- shuffle intermediate data over workers uniformly

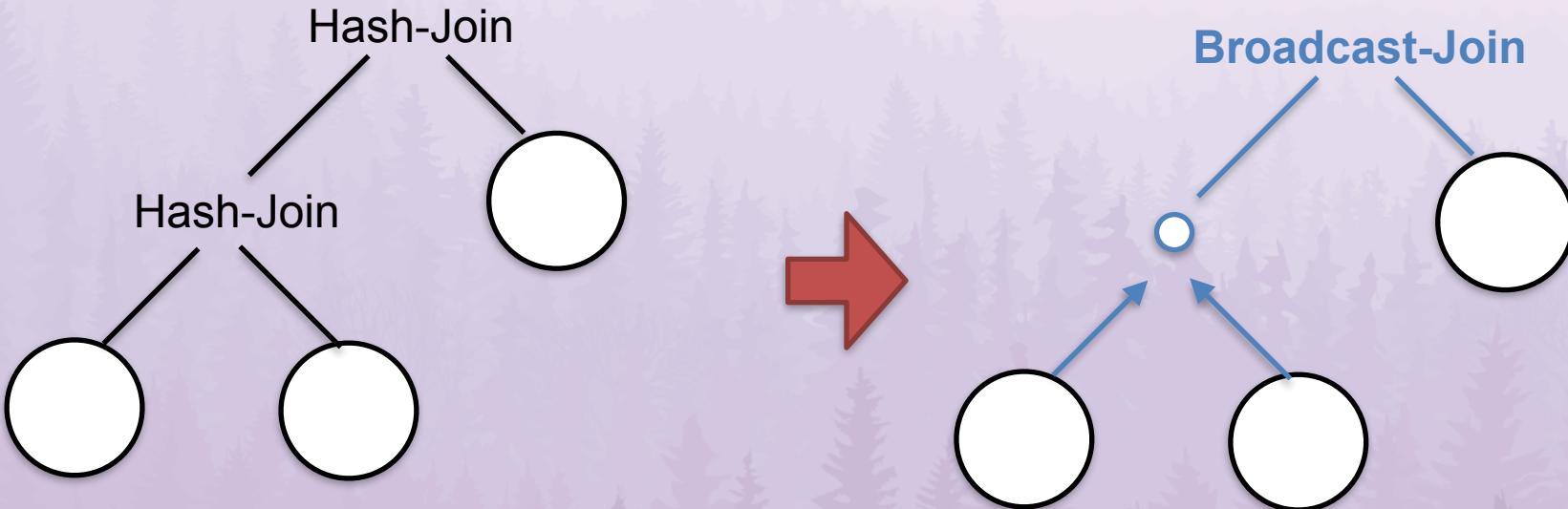
Tajo progressive optimization

- dynamically adjust join order or type



Tajo progressive optimization

- dynamically adjust join order or type



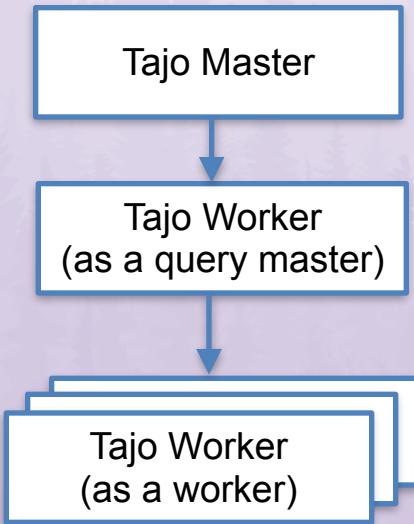
Tajo - what is improved past 9 months ?

- Resource Manager
- Scheduler & Storage Manager
- Data types & Functions
- SQL Interface
- Management



Tajo resource manager

- Fine resource allocation

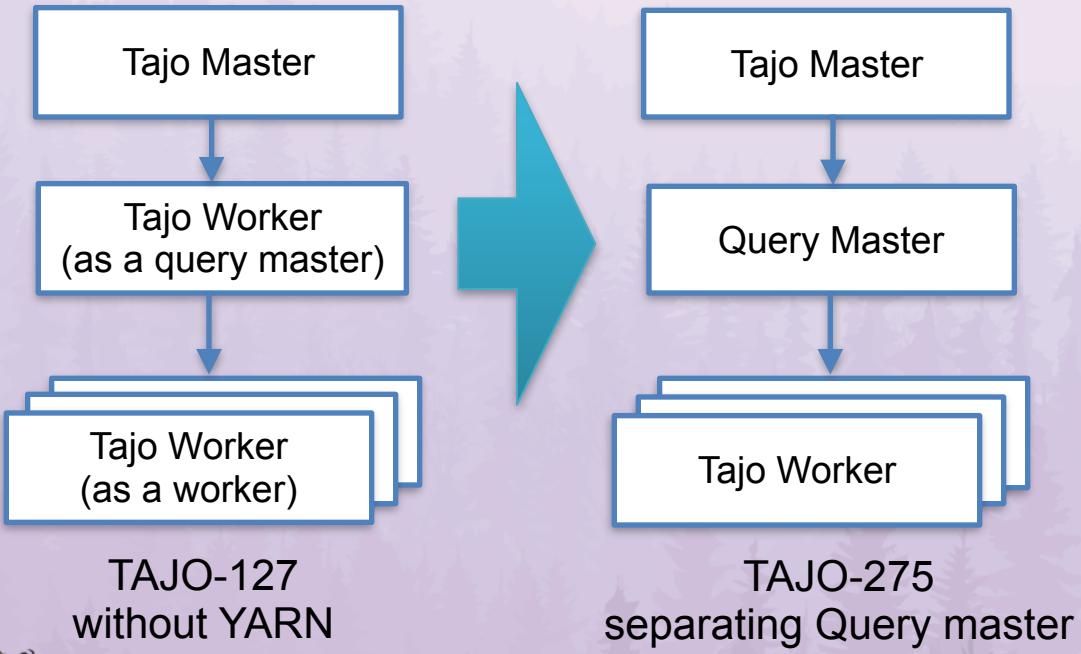


TAJO-127
without YARN



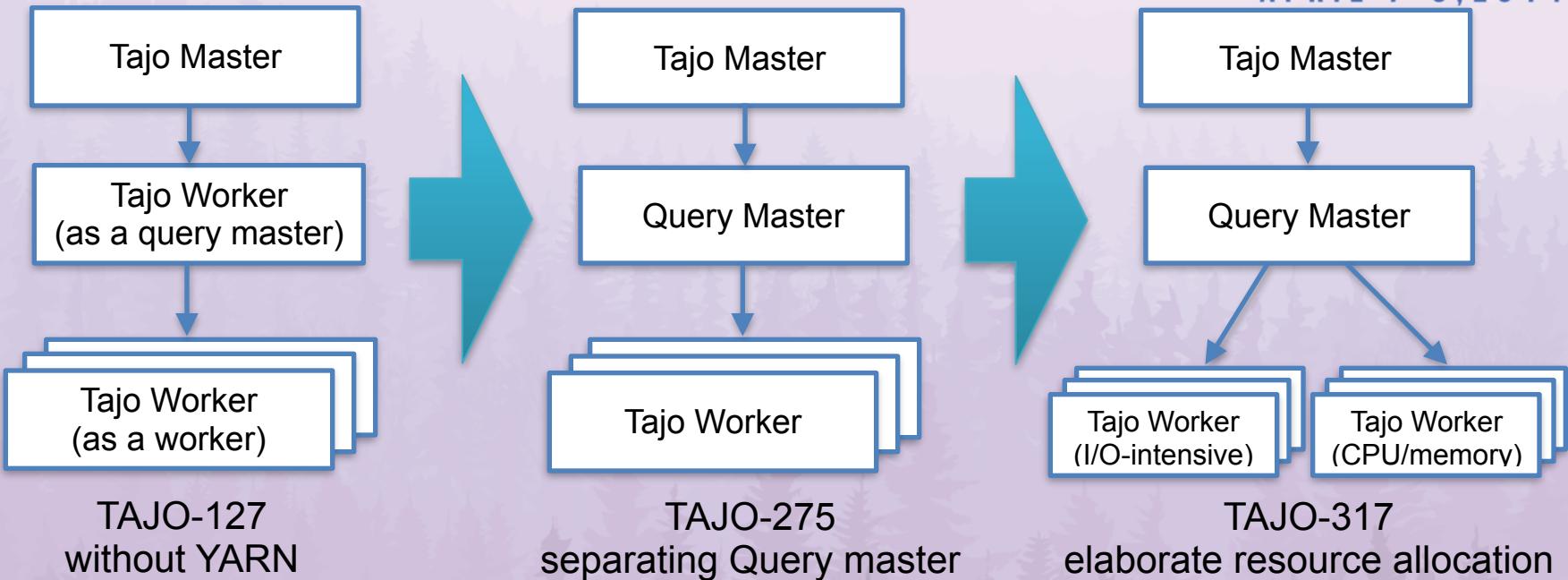
Tajo resource manager

- Fine resource allocation



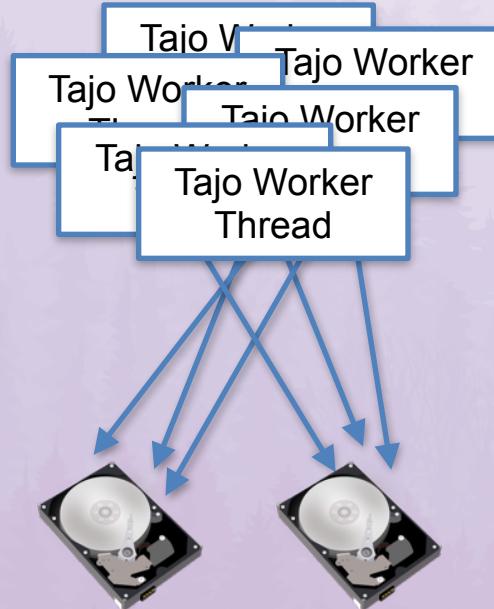
Tajo resource manager

- Fine resource allocation



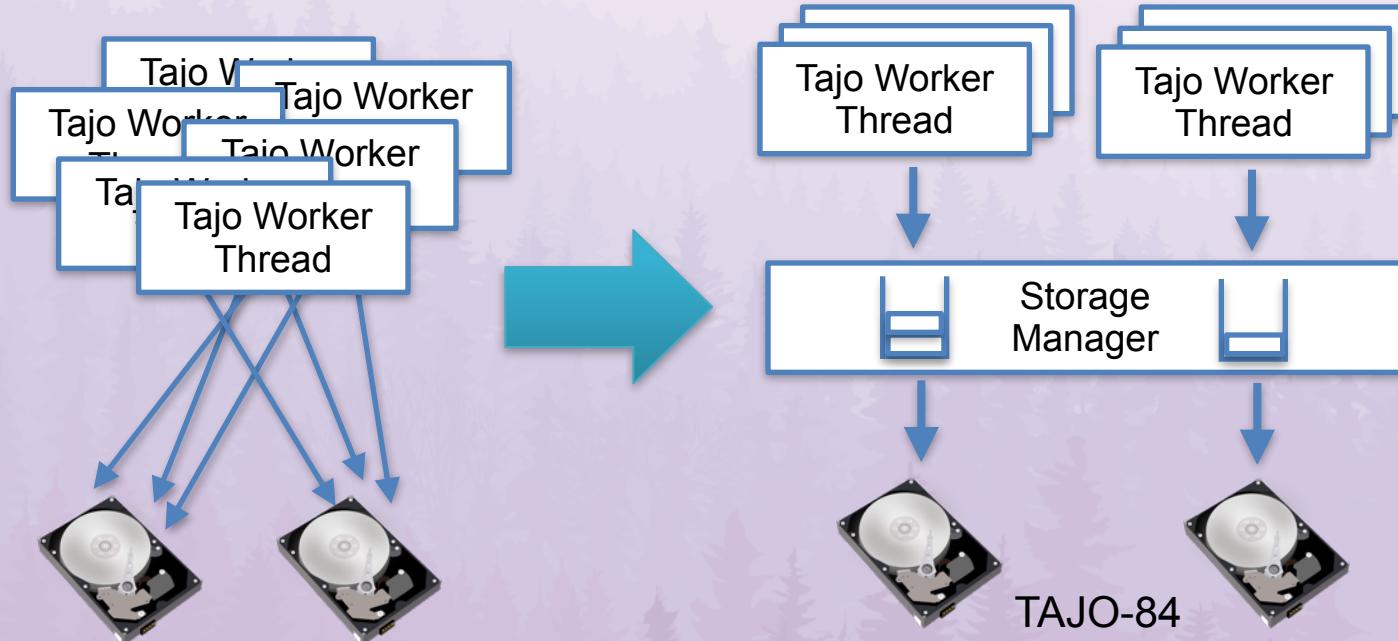
Scheduler & Storage manager

- disk-aware scheduling (volume info from HDFS-3672)



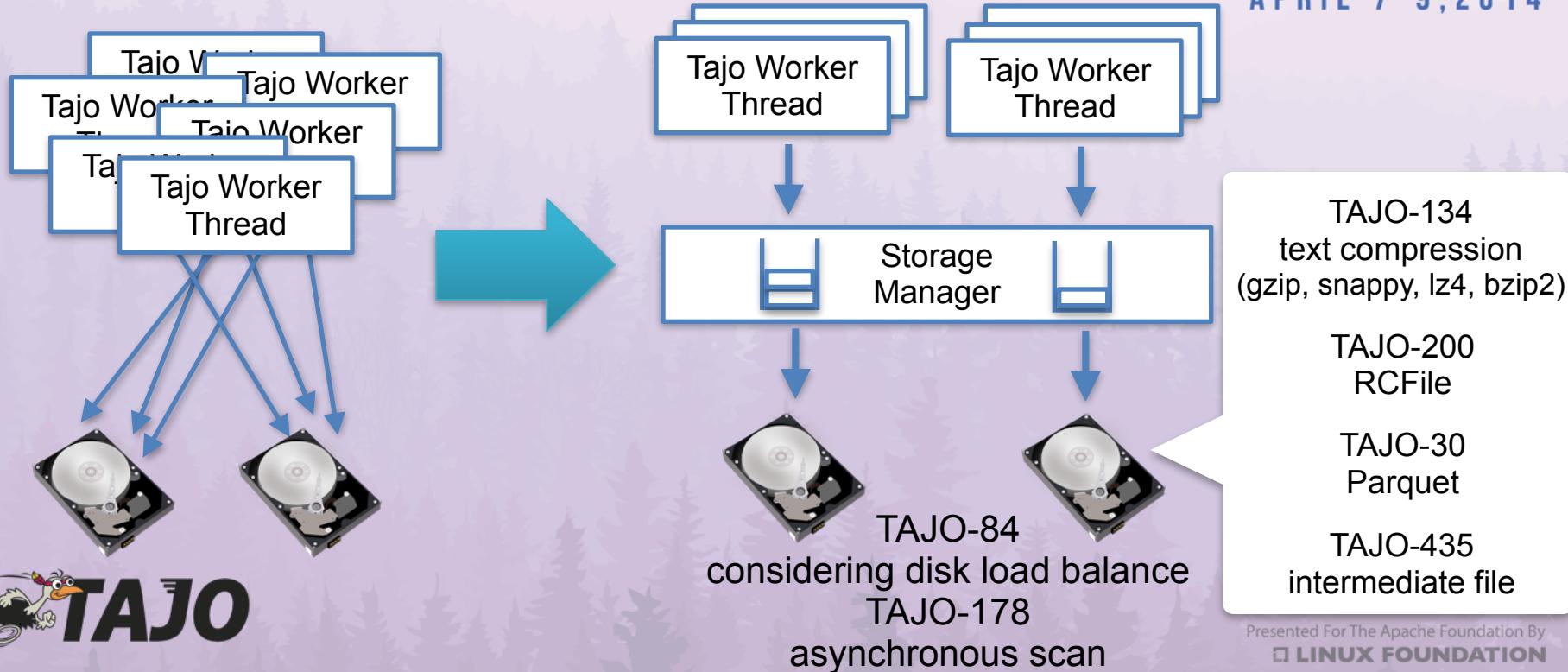
Scheduler & Storage manager

- disk-aware scheduling (volume info from HDFS-3672)



Scheduler & Storage manager

- disk-aware scheduling (volume info from HDFS-3672)



Functions & data types

- supporting more functions and UDFs

function1

function2

function3

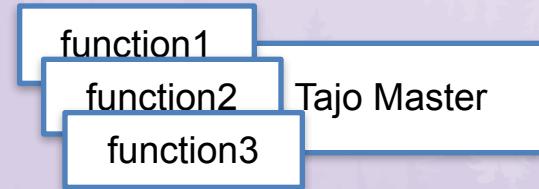
Tajo Master

registered at startup
(class name is coded in source)

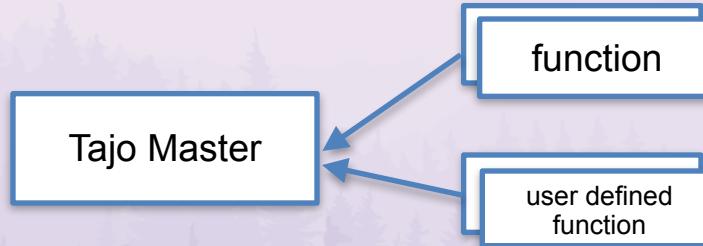


Functions & data types

- supporting more functions and UDFs



registered at startup
 (class name is coded in source)



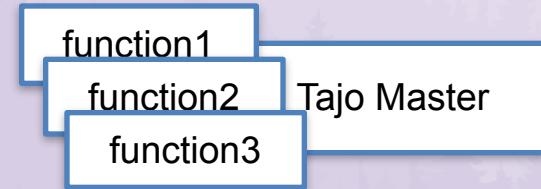
```

@Description(
  functionName = "to_timestamp",
  description = "Convert UNIX epoch to time stamp",
  example = "> SELECT to_timestamp(1389071574);\n"
            + "2014-01-07 14:12:54",
  returnType = TajoDataTypes.Type.TIMESTAMP,
  paramTypes = {@ParamTypes(paramTypes = {TajoDataTypes.Type.INT4}),
                @ParamTypes(paramTypes = {TajoDataTypes.Type.INT8})}
)
  
```

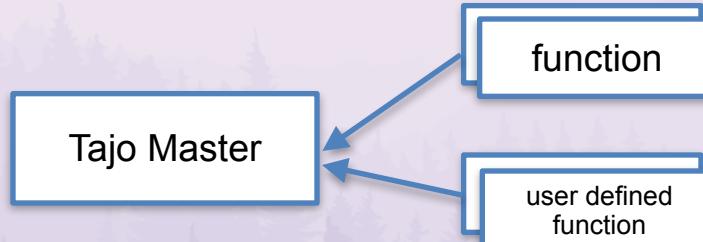
TAJO-408
 Improve function system

Functions & data types

- supporting more functions and UDFs



registered at startup
 (class name is coded in source)



```

@Description(
  functionName = "to_timestamp",
  description = "Convert UNIX epoch to time stamp",
  example = "> SELECT to_timestamp(1389071574);\n"
            + "2014-01-07 14:12:54",
  returnType = TajoDataTypes.Type.TIMESTAMP,
  paramTypes = {@ParamTypes(paramTypes = {TajoDataTypes.Type.INT4}),
                @ParamTypes(paramTypes = {TajoDataTypes.Type.INT8})}
)
  
```

TAJO-408
 Improve function system

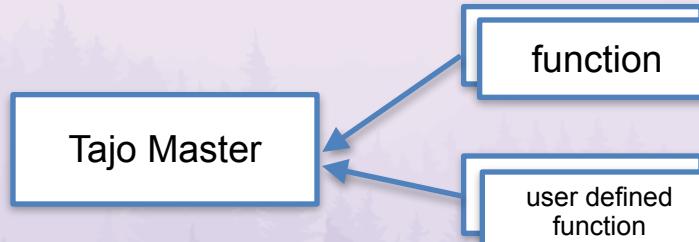


Functions & data types

- supporting more functions and UDFs



registered at startup
 (class name is coded in source)



```

@Description(
    functionName = "to_timestamp",
    description = "Convert UNIX epoch to time stamp",
    example = "> SELECT to_timestamp(1389071574);\n"
              + "2014-01-07 14:12:54",
    returnType = TajoDataTypes.Type.TIMESTAMP,
    paramTypes = {@ParamTypes(paramTypes = {TajoDataTypes.Type.INT4}),
                  @ParamTypes(paramTypes = {TajoDataTypes.Type.INT8})}
)
  
```

TAJO-408
 Improve function system

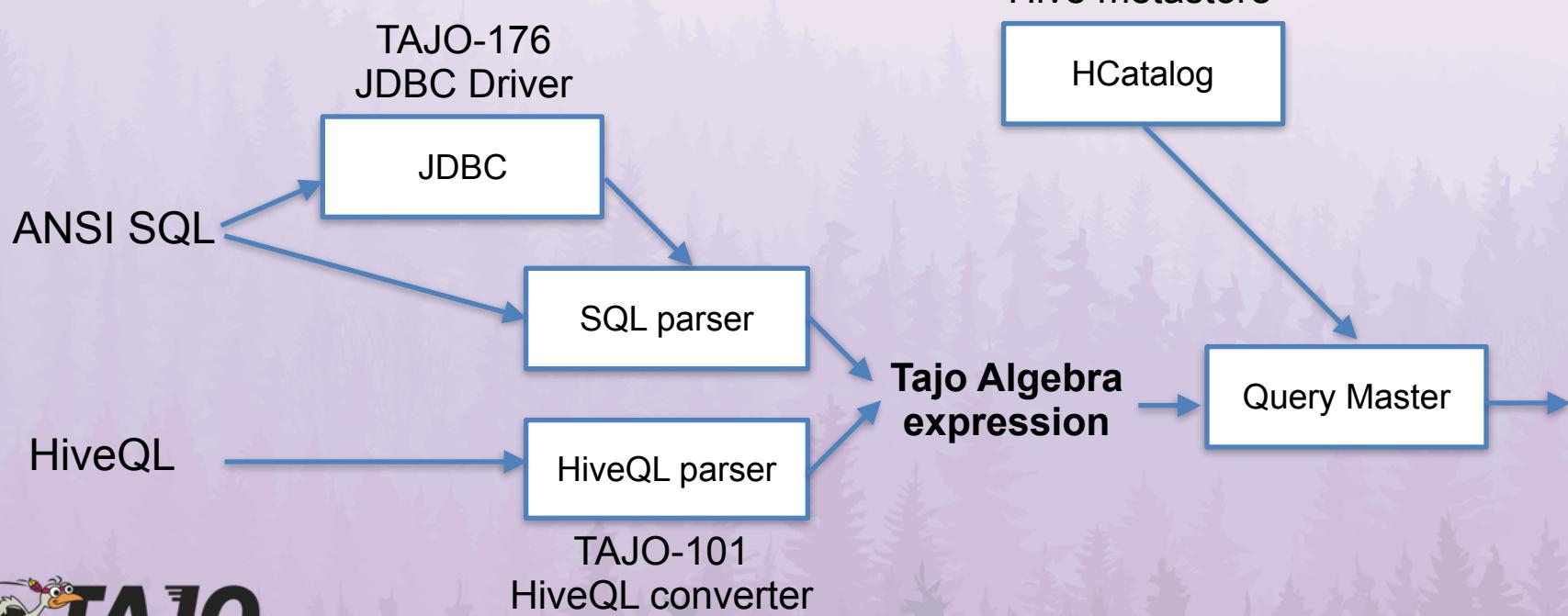
automatic
 registration

runtime
 registration

description

TAJO-52
 standard SQL
 data types

JDBC Driver, HCatalog



Management

TAJO Home Cluster Query Catalog Execute Query

Tajo Master: 50.1.102.122:26001

Query Master

Live:1, Dead: 0, QueryMaster Tasks: 0

Live QueryMasters

No	QueryMaster	Client Port	Running Query	Heap(free/total/max)	Heartbeat	Status
1	ceo-tajo02:28093	28092	0	847/921/921 MB	9.0 sec	LIVE

Worker

Live:6, Dead: 0

Live Workers

No	Worker	PullServer Port	Running Tasks	Memory Resource (used/total)	Disk Resource (used/total)	Heap (free/total/max)	Heartbeat	Status
1	ceo-tajo02:28091	45921	0	0/12288	0.0/6.0	47135/59882/59882 MB	.0 sec	LIVE
2	ceo-tajo03:28091	51067	0	0/12288	0.0/6.0	50460/59928/59928 MB	.0 sec	LIVE
3	ceo-tajo04:28091	37758	0	0/12288	0.0/6.0	54665/59877/59877 MB	.0 sec	LIVE
4	ceo-tajo05:28091	52609	0	0/12288	0.0/6.0	55691/59933/59933 MB	.0 sec	LIVE
5	ceo-tajo06:28091	48598	0	0/12288	0.0/6.0	41604/59890/59890 MB	9.0 sec	LIVE
6	ceo-tajo07:28091	51590	0	0/12288	0.0/6.0	54461/59876/59876 MB	.0 sec	LIVE

Dead Workers

No Dead Workers

TAJO-239 Improving Web UI



Management

Tajo Master: 50.1 Tajo Home Cluster Query Catalog Execute Query

Tajo Worker: [ceo-tajo02:28093](#)

Query Master

Live:1, Dead: 0, QueryMaster: q_1395132712581_0015 [\[Query Plan\]](#)

Live QueryMasters

No	Query	ID	State	Started	Finished	Running time	Progress	Tasks
1	ceo-tajo02:28093	eb_1395132712581_0015_000001	SUCCEEDED	2014-03-18 17:55:35	2014-03-18 17:55:35	.0 sec	100.0%	1/1

Logical Plan

Worker

Query Block Graph

```
|--#ROOT
```

Live Workers

Optimization Log:

No	Worker
1	ceo-tajo02:28091
2	ceo-tajo03:28091
3	ceo-tajo04:28091
4	ceo-tajo05:28091
5	ceo-tajo06:28091
6	ceo-tajo07:28091

Distributed Query Plan

Execution Block Graph (TERMINAL - eb_1395132712581_0015_000002)

```
|--eb_1395132712581_0015_000002
 |--eb_1395132712581_0015_000001
```

Dead Workers

No Dead Workers

Block Id: eb_1395132712581_0015_000001 [ROOT]

TAJO-564
Execution block progress



Management

Tajo Master: 50.1.102.123

Tajo Worker: ceo-tajo02:28093

Query Master

Live:1, Dead: 0, QueryMaster: q_1395132712581_0015

Live QueryMasters

No	Que	ID
1	ceo-tajo02:28093	eb_1395132712581_0015_000001

Logical Plan

```
SCAN(0) on table1
=> target list: table1.id (INT4), table1.name (TEXT), table1.score (FLOAT4), table1.type (TEXT)
=> out schema: ((4) table1.id (INT4),table1.name (TEXT),table1.score (FLOAT4),table1.type (TEXT))
=> in schema: ((4) table1.id (INT4),table1.name (TEXT),table1.score (FLOAT4),table1.type (TEXT))
```

Worker

Live:6, Dead: 0

|--#ROOT

Live Workers

No	Worker
1	ceo-tajo02:28091
2	ceo-tajo03:28091
3	ceo-tajo04:28091
4	ceo-tajo05:28091
5	ceo-tajo06:28091
6	ceo-tajo07:28091

Distributed Query Plan

Execution Block Graph (

|--eb_1395132712581_0015

|--eb_1395132712581_0

Dead Workers

No Dead Workers

Block Id: eb_1395132712581_0015_000001_000000

Status: ALL Filter

No	Id	Status	Progress	Started	Running Time	Host
1	t_1395132712581_0015_000001_000000	SUCCEEDED	100.0%	2014-03-18 17:55:35	34 ms	50.1.102.123

TAJO-589
 Task progress



Management

The screenshot shows the Apache Tajo Management UI interface. It includes several tabs at the top: Home, Cluster, Query, Catalog, and Execute Query. Below these are sections for Tajo Master, Tajo Worker, and Live Workers.

- Tajo Master:** Shows a table of Query Masters with one entry: "Live:1, Dead: 0, QueryMaster: q_1395132712581_0015".
- Tajo Worker:** Shows a table of Live QueryMasters with one entry: "ID: eb_1395132712581_0015, Que: 1, Tajo Worker: ceo-tajo02:28093".
- Logical Plan:** Displays the execution plan for the query, showing a scan on table1 followed by target list, output schema, and input schema.
- Worker:** Shows a Query Block Graph with a single node labeled "#ROOT".
- Live Workers:** Shows a table of workers with seven entries: "ceo-tajo02:28091", "ceo-tajo03:28091", "ceo-tajo04:28091", "ceo-tajo05:28091", "ceo-tajo06:28091", "ceo-tajo07:28091", and "ceo-tajo08:28091".
- Distributed Query Plan:** Shows an Execution Block Graph with a single node labeled "eb_1395132712581_0015".
- Dead Workers:** Shows a table with one entry: "No Dead Workers".
- Block Id:** Shows the value "eb_1395132712581_0015".

On the right side, there is a detailed view of a task named "eb_1395132712581_0015_000001". The task details include:

	Value
ID	t_1395132712581_0015_000001_000000
Progress	100.0%
State	SUCCEEDED
Launch Time	2014-03-18 17:55:35
Finish Time	2014-03-18 17:55:35
# Tasks	1
Running Time	34 ms
Host	50.1.102.123
Shuffles	# Shuffle Outputs: 1, Shuffle Key: 0, Shuffle file: DataLocation(host=ceo-tajo06, volumeId=0) DataLocation(host=ceo-tajo07, volumeId=2) DataLocation(host=ceo-tajo04, volumeId=5)
Input Bytes	6
Actual Processed Bytes	6
Data Locations	
Input Rows	5
Output Bytes	6
Fragment	"fragment": {"id": "table1", "path": "hdfs://ceo-tajo01:9000/tables/table1/data.csv", "start": 0, "length": 60}
Output Rows	5
Input Statistics	TotalBytes: 60 B (60 B), ReadBytes: 60 B (60 B), ReadRows: 5
Output Statistics	TotalBytes: 60 B (60 B), ReadBytes: 60 B (60 B), ReadRows: 5
Fetched	[{"numRows": 5, "numBytes": 60, "numBlocks": 0, "numShuffleOutputs": 0, "avgRows": 0, "readBytes": 0, "columnStatsets": [{"column": {"name": "table1.id", "dataType": ["type": "INT4"]}, "numDistVals": 0, "numNulls": 0}, {"column": {"name": "table1.name", "dataType": ["type": "TEXT"]}, "numDistVals": 0, "numNulls": 0}, {"column": {"name": "table1.score", "dataType": ["type": "FLOAT4"]}, "numDistVals": 0, "numNulls": 0}, {"column": {"name": "table1.type", "dataType": ["type": "TEXT"]}, "numDistVals": 0, "numNulls": 0}]}]

TAJO-468
 Task detail info



Management

Tajo Master: 50.1

Tajo Worker: ceo-tajo

Query Master

Live:1, Dead: 0, QueryMaster: q_1395132712581_0015

Live QueryMasters

No	QueryId
1	ceo-tajo02:28093

Logical Plan

Worker

Live:6, Dead: 0

|--#ROOT

Live Workers

No	Worker
1	ceo-tajo02:28091
2	ceo-tajo03:28091
3	ceo-tajo04:28091
4	ceo-tajo05:28091
5	ceo-tajo06:28091
6	ceo-tajo07:28091

Distributed Query Plan

Execution Block Graph (eb)

|--eb_1395132712581_0015
|--eb_1395132712581_0016

Dead Workers

No Dead Workers

Block Id: eb_1395132712581_0015

Tajo Worker: ceo-tajo

Tajo Worker: ceo-tajo

```
[hadoop@ceo-tajo01 ~]$ tajo admin -list
QueryId          State   StartTime
eb_1395132712581_0015
```

SCAN(0) on table1
=> target list: table1
=> out schema: {((4) ta
=> in schema: {((4) tab

Status:	Si
Started:	21
# Tasks:	1
Progress:	11
# Shuffles:	1
Input Bytes:	61
Actual Processed Bytes:	61
Input Rows:	5
Output Bytes:	61
Output Rows:	5

Status: ALL Filter

No	QueryId
1	t_1395132712581_0015

•Tajo management

Tajo Master: 50.1

TAJO-474
Task admin utility



And lots of Performance enhancement

TAJO-725 Broadcast JOIN should supports multiple tables

TAJO-717 Improve file splitting for large number of splits

TAJO-601 Improve distinct aggregation query processing

TAJO-584 Improve distributed merge sort

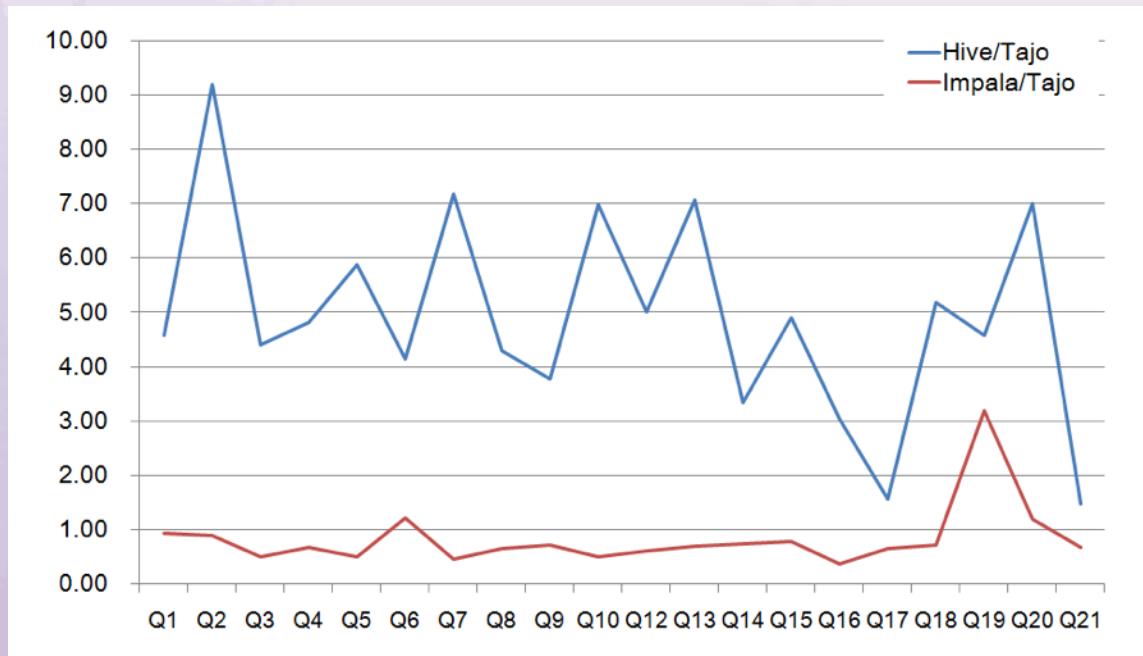
TAJO-36 Improve ExternalSortExec with N-merge sort and final pass omission

TAJO-345 MergeScanner should support projectable storages

...

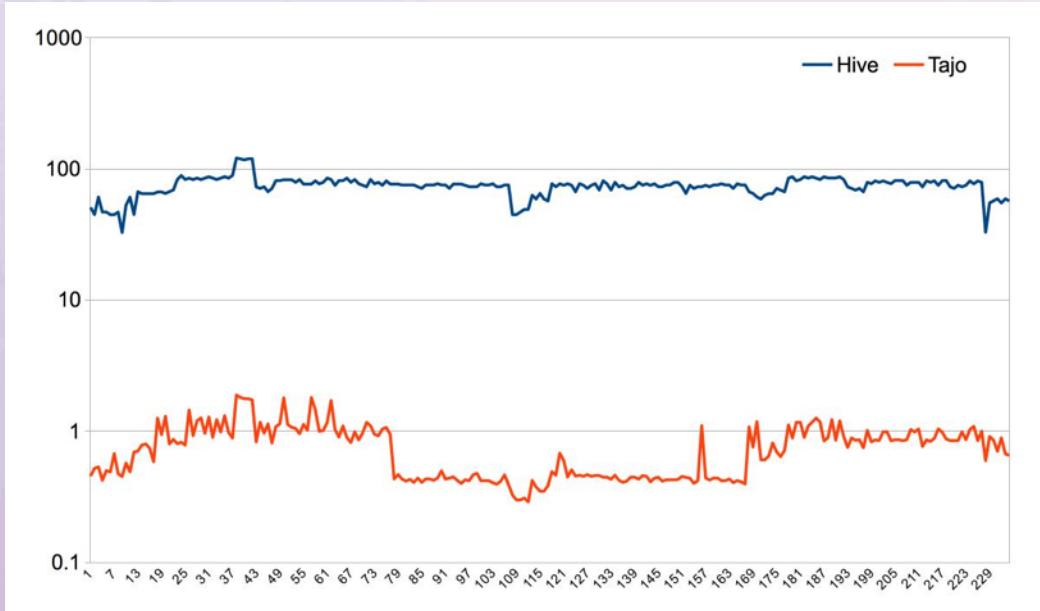
Performance

- TPC-H



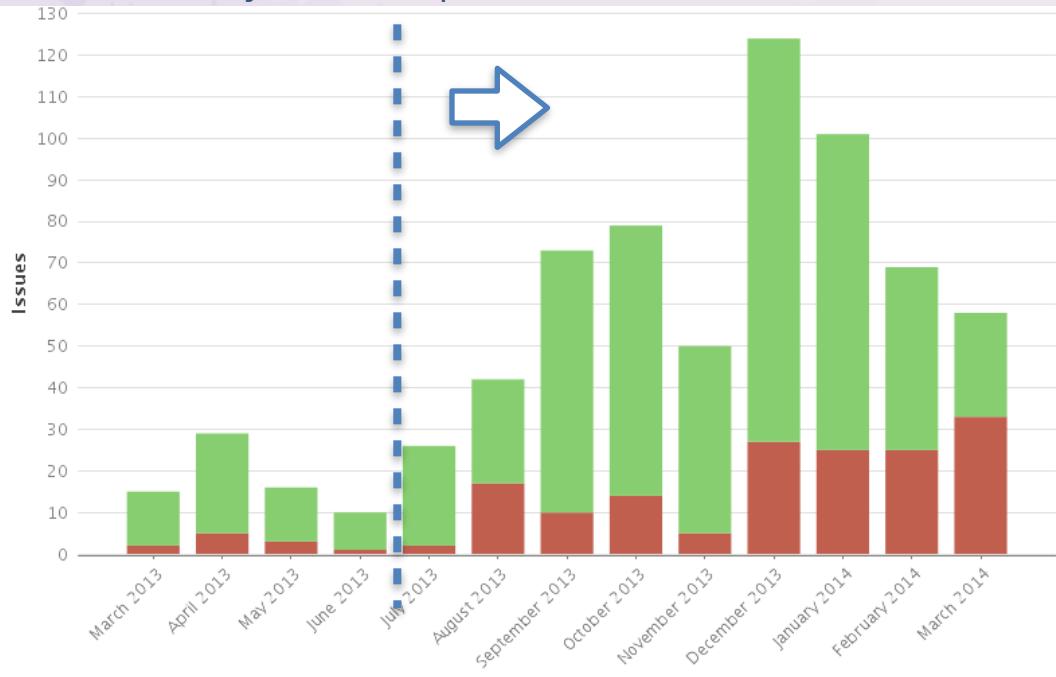
Performance

- OLAP reporting - relatively small data



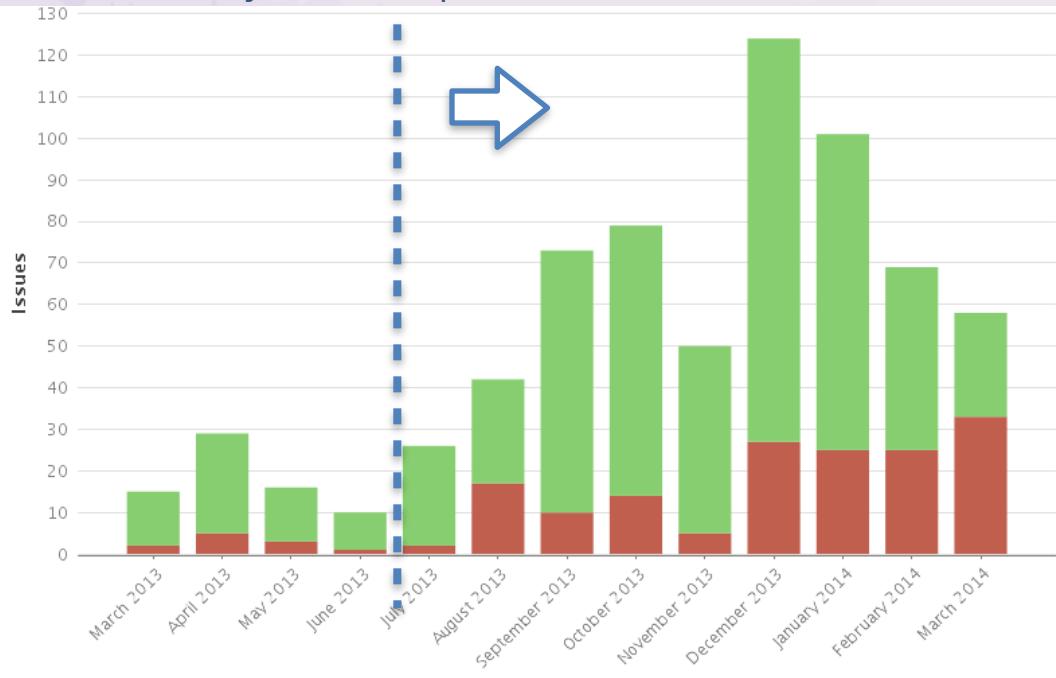
win-win between company and community

- Community boom up



win-win between company and community

- Community boom up



13 → 30

win-win between company and community

- Test in real working cluster
 - Mainly focusing on the scalability test & integration with existing IT systems
 - Finding bugs and function requirements, also

win-win between company and community

- Test in real working cluster
 - Mainly focusing on the scalability test & integration with existing IT systems
 - Finding bugs and function requirements, also

TAJO-691 HashJoin or HashAggregation is too slow if there is many unique keys

TAJO-675 maximum frame size of frameDecoder should be increased

TAJO-673 Assign proper number of tasks when inserting into partitioned table

TAJO-650 Repartitioner::scheduleHashShuffledFetches should adjust the number of tasks

TAJO-647 Work unbalance on disk scheduling of DefaultScheduler

TAJO-292 Too many intermediate partition files

TAJO-283 Add table partitioning

TAJO-592 HCatalogStore should supports RCFile and default hive field delimiter.

...

win-win between company and community



APACHECON
DENVER
WESTIN DENVER DOWNTOWN
APRIL 7-9, 2014



Presented For The Apache Foundation By
□ LINUX FOUNDATION

win-win between company and community



APACHE CON
DENVER
WESTIN DENVER DOWNTOWN
APRIL 7-9, 2014

- efficient development and operation
- human networking
- brand value up - recruiting

Future Works

- Nested data model (parquet model)
- more SQL compatible
 - window functions, IN, EXIST
- Multi-tenancy
- push shuffle (no materialization)
 - use selectively between push and pull shuffle
 - push shuffle: performance
 - pull shuffle: resilience, schedulability

Q & A

- Getting Started
 - <http://tajo.apache.org/tajo-0.2.0-doc.html#GettingStarted>
- Checkout the development branch
 - <http://tajo.apache.org/downloads.html>
- Jira - Issue Tracker
 - <https://issues.apache.org/jira/browse/TAJO>
- Join the mailing list
 - dev@tajo.apache.org