Big Telco, Bigger real-time demands: Moving towards Real-time analytics

Jung Ryong Lee

- IT Manager of SK Telecom, South Korea's largest wireless communications provider

Real-time analytics

Jung Ryong Lee

SK telecom

- IT Manager of SK Telecom, South Korea's largest wireless communications provider
- Work on commercial products (~ '12)
    - Has worked with email archived solutions
    - Has worked with IDS using In-stream processing engine
- Open source activity ('14 ~)
    - Contributor of REEF

Overview

- Background
- Real-time analytics in Telco

- Has worked with IDS using In-stream processing engine

- Open source activity ('14 ~)
  - Contributor of REEF

Overview

- Background
- Real-time analytics in Telco
- Project 1 - High speed data processing
  - Issues & solutions
  - Performance
- Project 2 - In-stream processing
- Lessons Learned

Background

- Telco data characteristics
  - Huge amount of data daily

  - Performance
- Project 2 - In-stream processing
- Lessons Learned

Background

- Telco data characteristics
  - Huge amount of data daily
    - 40 TB/day
    - 15 PB (estimated by the end of 2014)
- Active user of Hadoop
  - Involved with 10+Hadoop clusters
    - The largest one has 500+nodes and a total of 900+nodes altogether.
- Uses various commercial MPP databases for analytics

10+Clusters

Real-time analytics in Telco

Introduce 2 projects using Spark

- Involved with 10 + Hadoop clusters
  - The largest one has 500 + nodes and a total of 900 + nodes altogether.
- Uses various commercial MPP databases for analytics

Real-time analytics in Telco

Introduce 2 projects using Spark

1. The first being high speed data processing
   - Replacement of MPP database
2. The second being In-Stream data processing
   - Replacement of Hive batch job

Previous approach

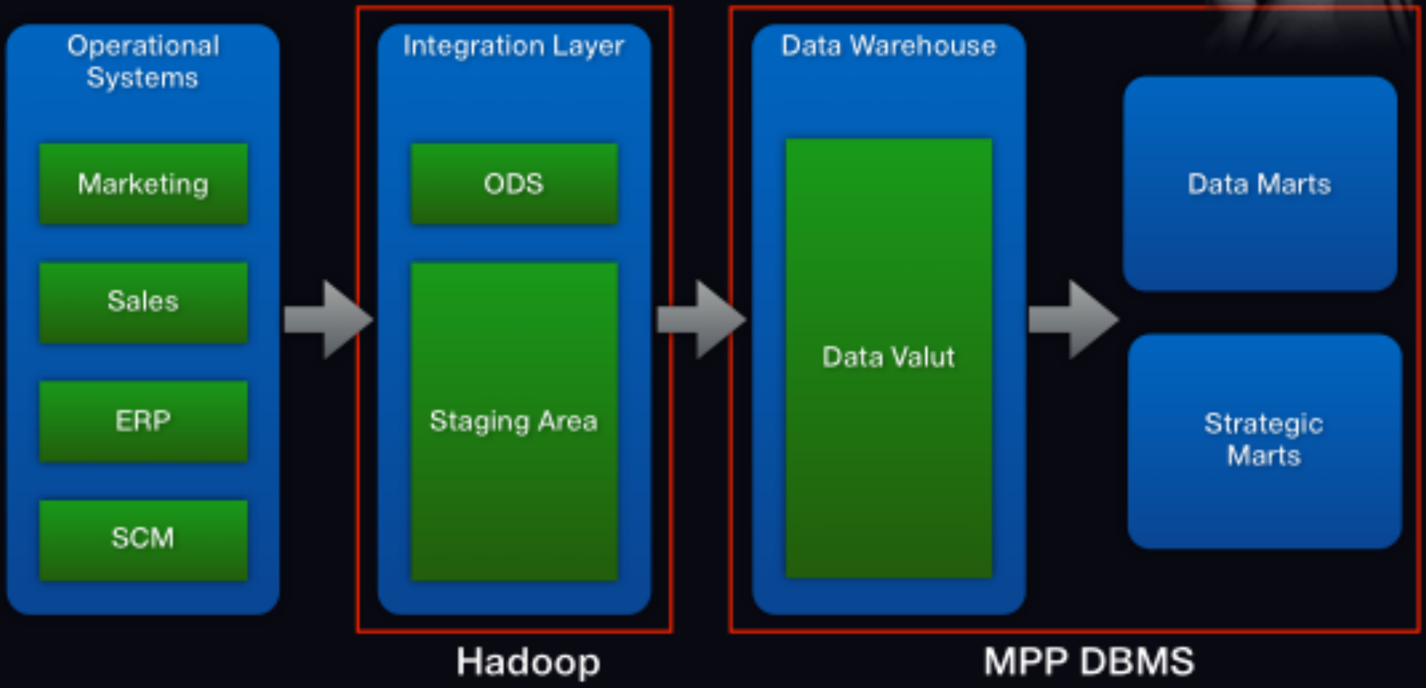
Working, But···

2. The second being In-Stream data processing

- Replacement of Hive batch job

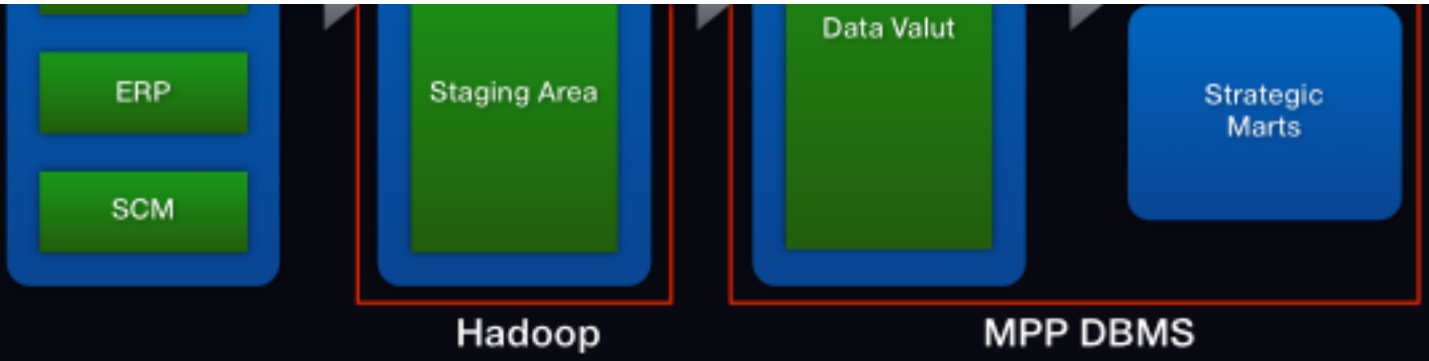
Previous approach

Working, But…



Previous approach

have to load too much data into MPP DBMS



ERP

SCM

Staging Area

Data Valut

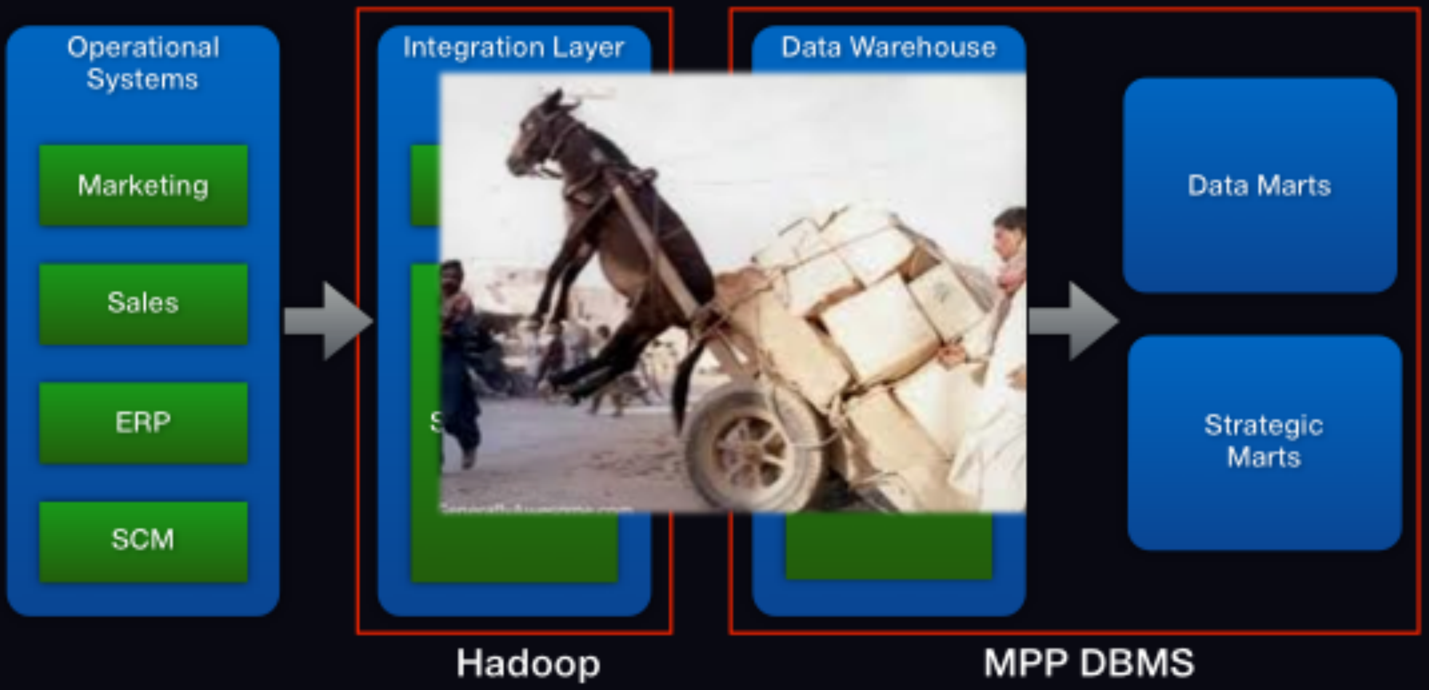
Strategic Marts

Hadoop

MPP DBMS

Previous approach

have to load too much data into MPP DBMS



Operational Systems

Marketing

Sales

ERP

SCM

Integration Layer

Data Warehouse

Data Marts

Strategic Marts

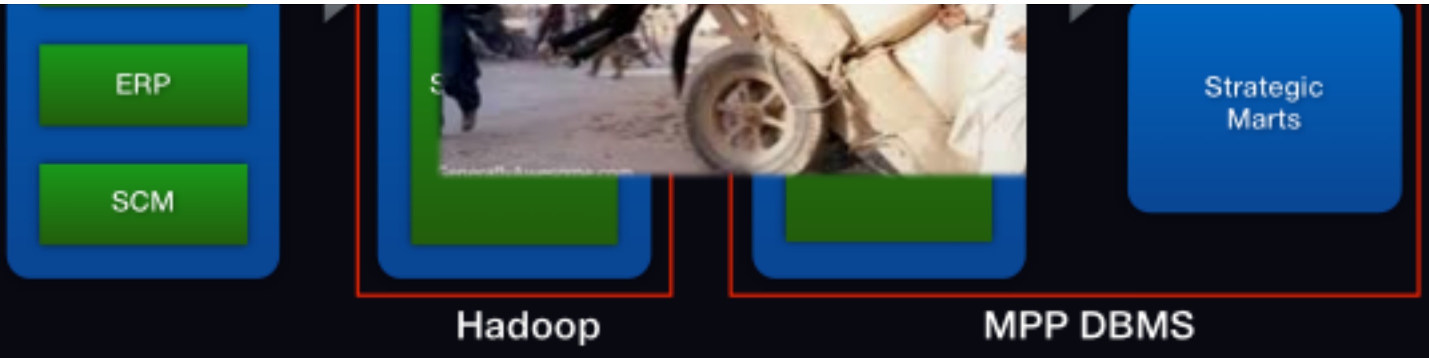
Hadoop

MPP DBMS

New approach

Support High speed ETL data processing

Real-time query processing for web client users



ERP

SCM

Strategic Marts

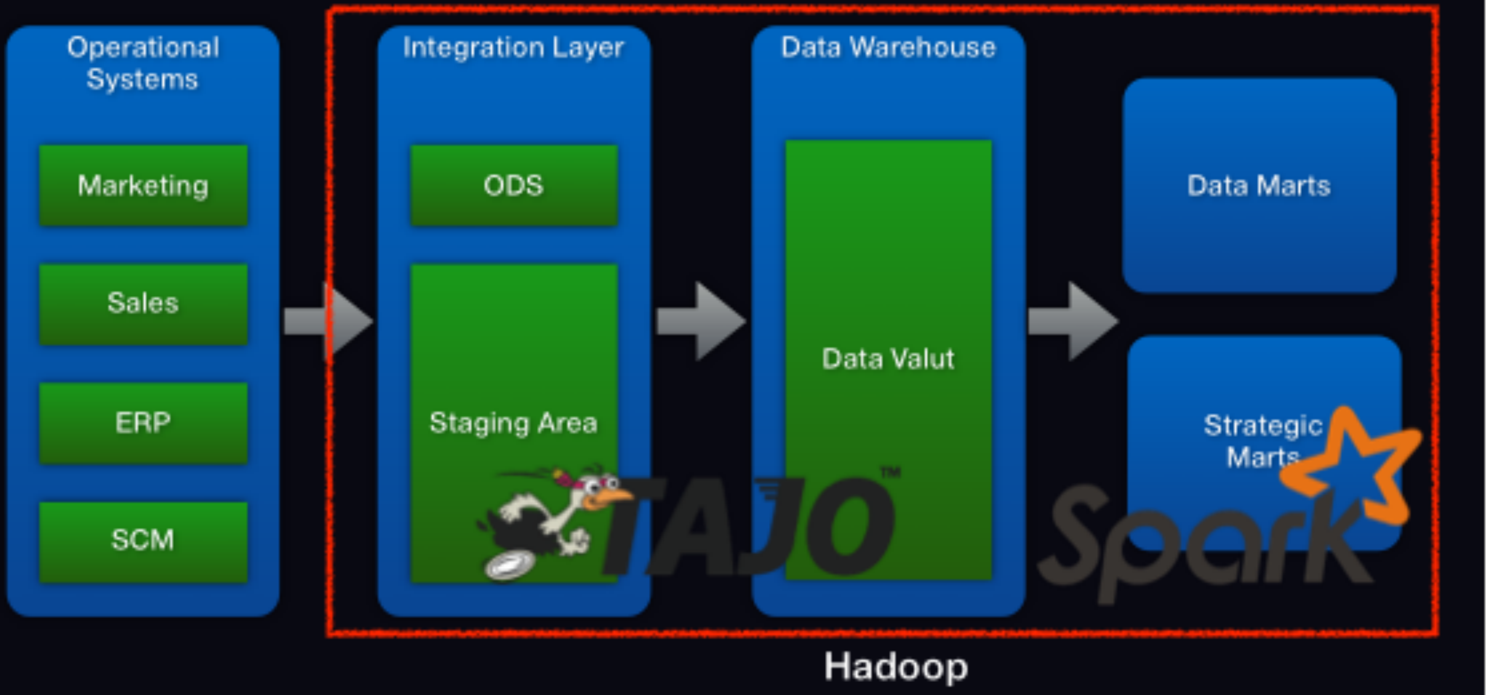
Hadoop

MPP DBMS

New approach

Support High speed ETL data processing

Real-time query processing for web client users



Operational Systems

Marketing

Sales

ERP

SCM

Integration Layer

ODS

Staging Area

Data Warehouse

Data Valut

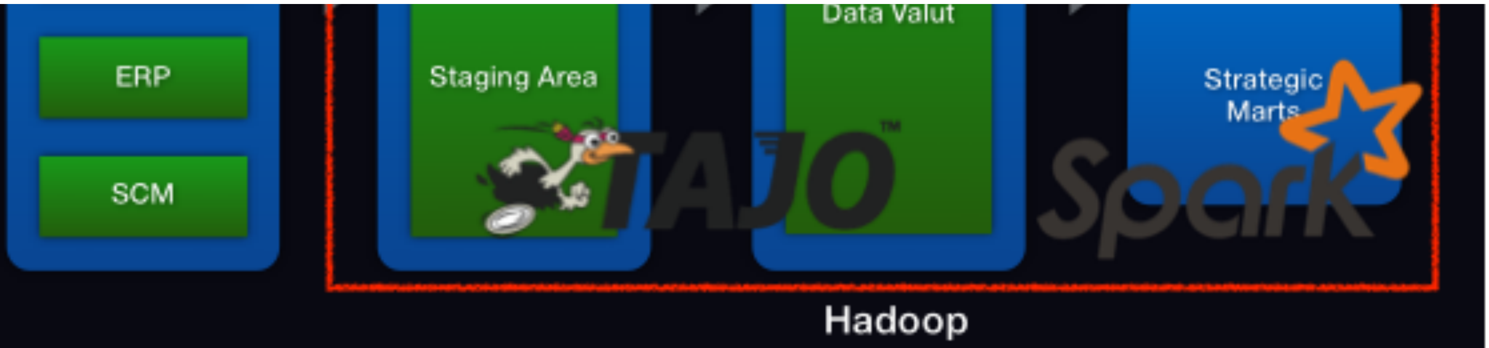
Data Marts

Strategic Marts

Hadoop

System Requirements

- Low latency ad-hoc query(< 2secs)



System Requirements

- Low latency ad-hoc query(< 2secs)
- ANSI SQL support
  (no need for Insert/Update/Delete)
- JDBC support
- Support concurrent users(10 users per sec)
- High availability

Shark on Spark

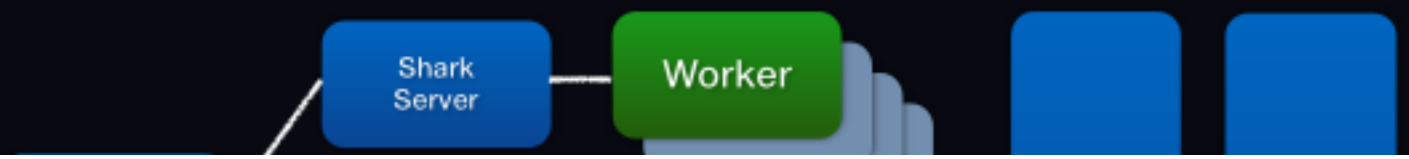
- It can replace RDBMS

- Support concurrent users(10 users per sec)
- High availability

Shark on Spark

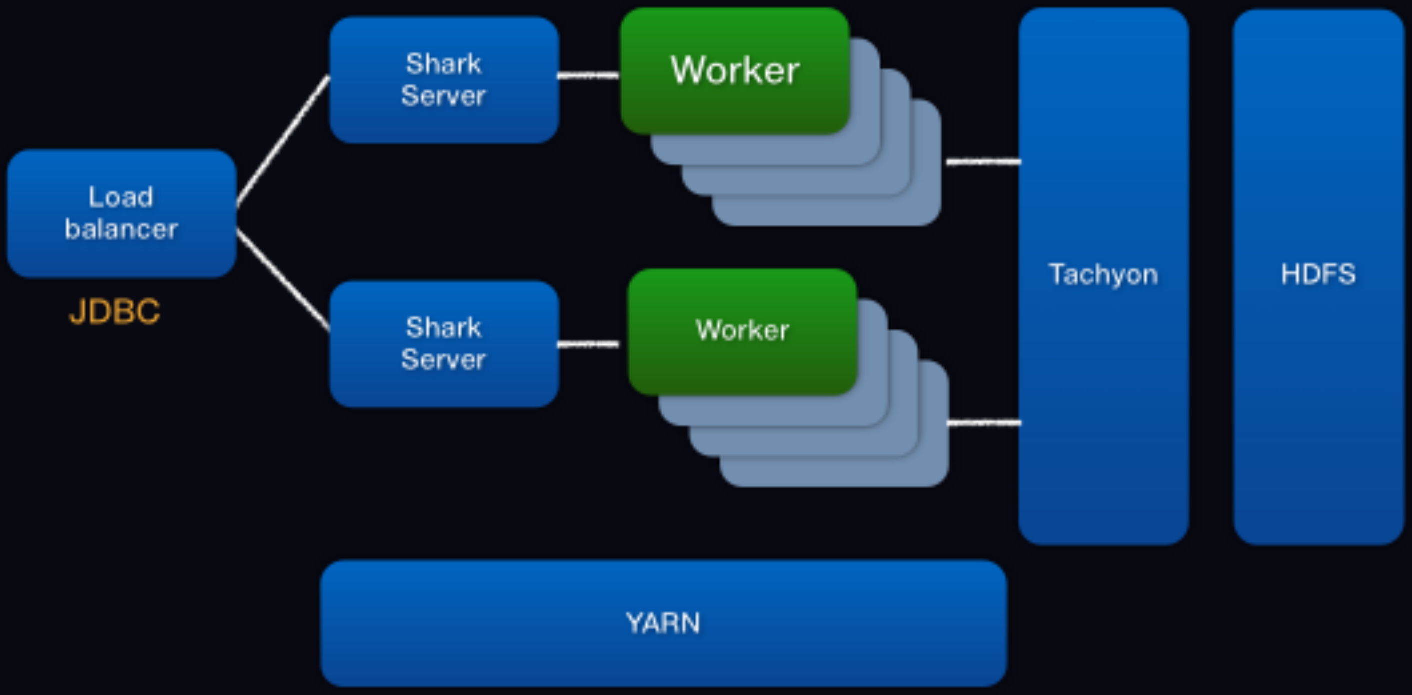
- It can replace RDBMS
  - Low latency for ad-hoc query with caching tables
  - HiveQL support
  - JDBC support
- Support concurrent users(10users per sec)
- High availability

Shark on Spark



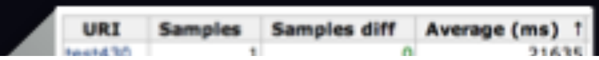
- Support concurrent users(10users per sec)
- High availability

Shark on Spark



Function & Performance

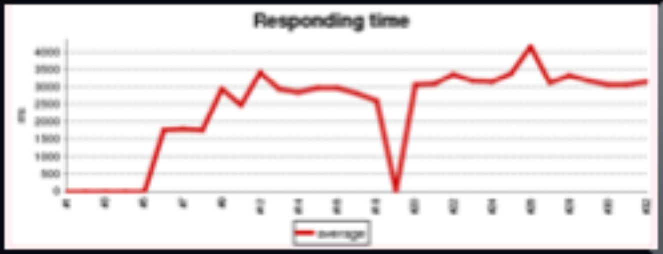
- All queries work good with some modifications of queries

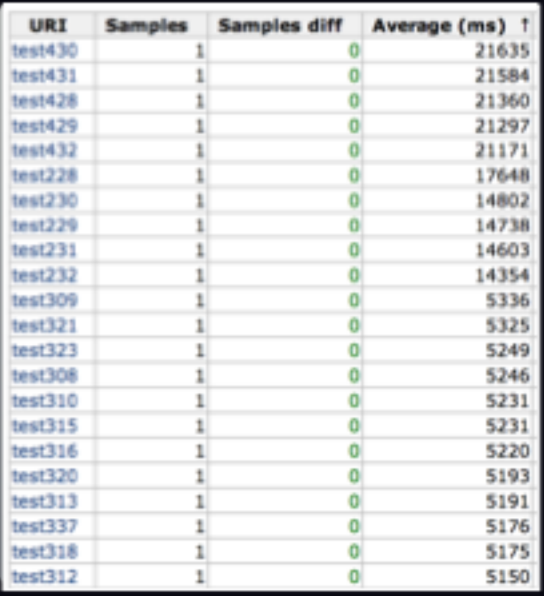


URI	Samples	Samples diff	Average (ms) ↑
test430	1	0	21635

Function & Performance

- All queries work good with some modifications of queries

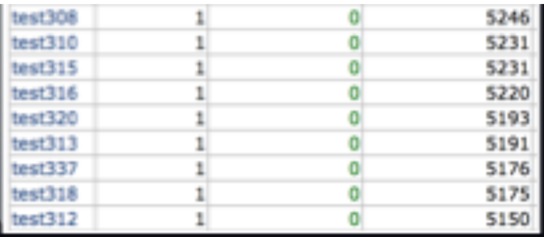




URI	Samples	Samples diff	Average (ms) ↑
test430	1	0	21635
test431	1	0	21584
test428	1	0	21360
test429	1	0	21297
test432	1	0	21171
test228	1	0	17648
test230	1	0	14802
test229	1	0	14738
test231	1	0	14603
test232	1	0	14354
test309	1	0	5336
test321	1	0	5325
test323	1	0	5249
test308	1	0	5246
test310	1	0	5231
test315	1	0	5231
test316	1	0	5220
test320	1	0	5193
test313	1	0	5191
test337	1	0	5176
test318	1	0	5175
test312	1	0	5150

Improvement

- Improvement and Bug fix
  - Bug fix of Hive 0.11 / Support Non ASCII table name
  - Implement UDF(Oracle like functions)



test308	1	0	5246
test310	1	0	5231
test315	1	0	5231
test316	1	0	5220
test320	1	0	5193
test313	1	0	5191
test337	1	0	5176
test318	1	0	5175
test312	1	0	5150

Improvement

- Improvement and Bug fix
  - Bug fix of Hive 0.11 / Support Non ASCII table name
  - Implement UDF(Oracle like functions)
  - Change Correlation Sub Query to Left Outer Join
- Performance Improvement
  - Control reducer count
  - Material view can reduce query time
  - Caching query result

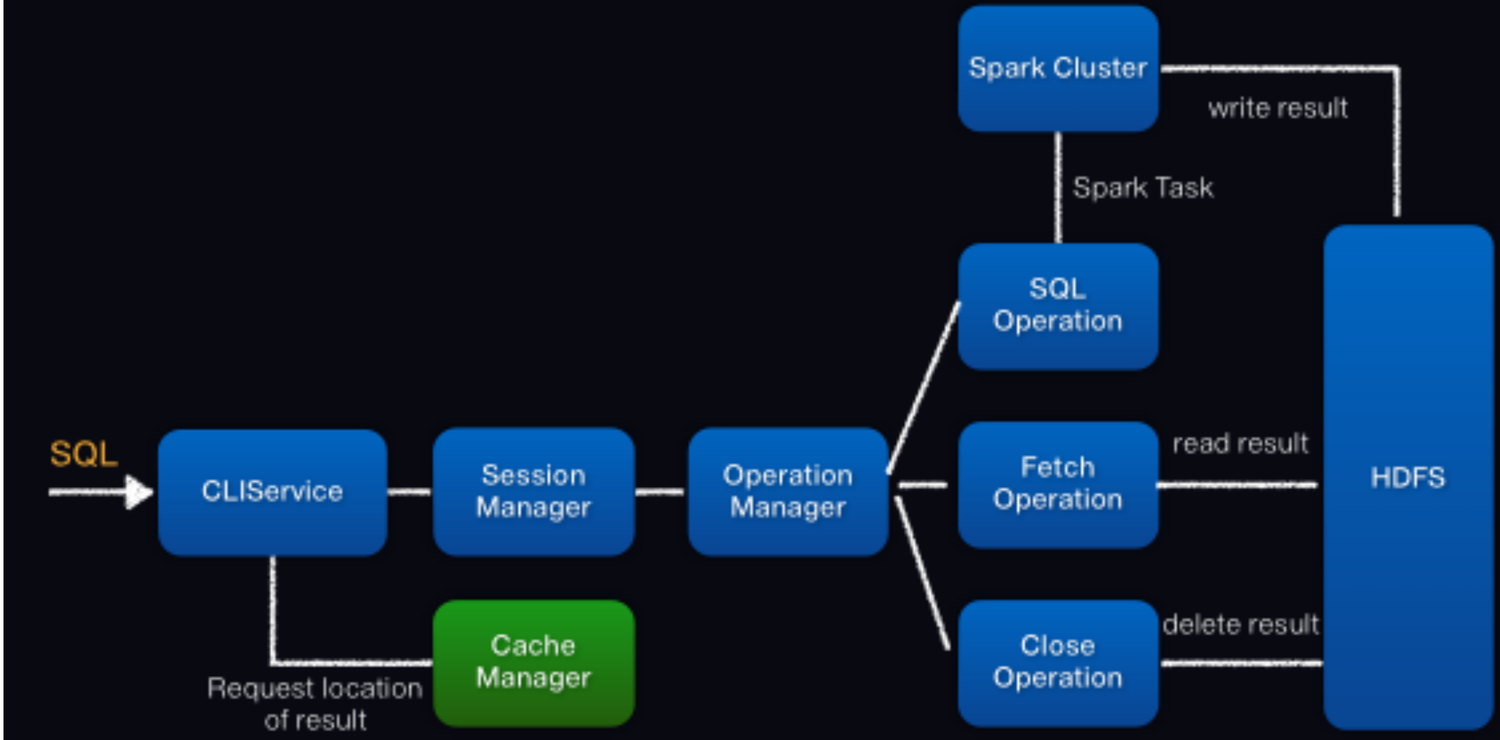
Improvement - Example

- Can reduce query time with a simple understanding of Shark and Hive

- Control reducer count
- Material view can reduce query time
- Caching query result

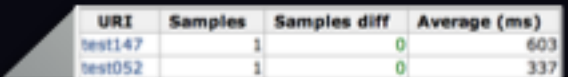
Improvement - Example

- Can reduce query time with a simple understanding of Shark and Hive

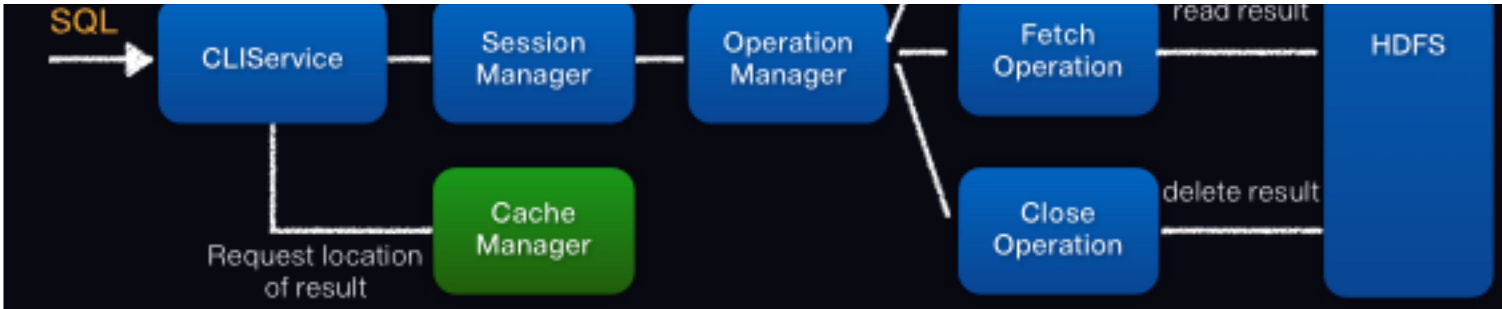


Result of Caching query

- Most of queries complete in 0.3 second.

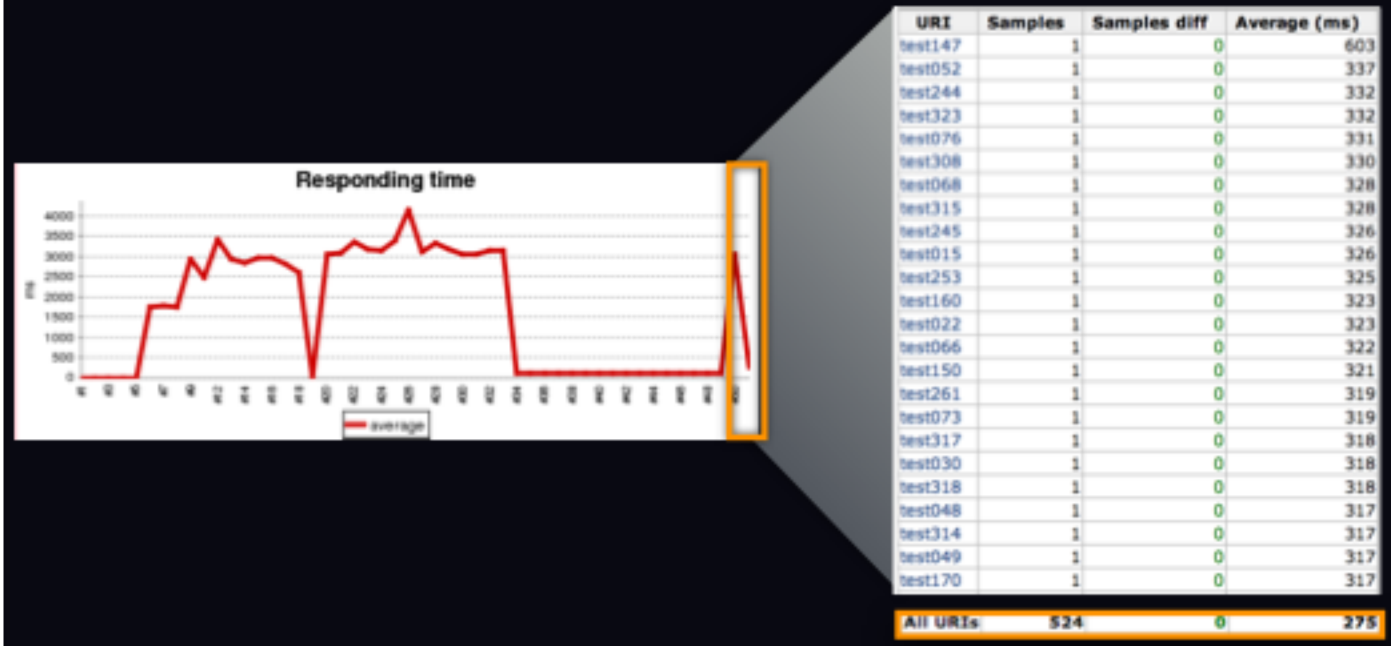


URI	Samples	Samples diff	Average (ms)
test147	1	0	603
test052	1	0	337



Result of Caching query

- Most of queries complete in 0.3 second.



URI	Samples	Samples diff	Average (ms)
test147	1	0	603
test052	1	0	337
test244	1	0	332
test323	1	0	332
test076	1	0	331
test308	1	0	330
test068	1	0	328
test315	1	0	328
test245	1	0	326
test015	1	0	326
test253	1	0	325
test160	1	0	323
test022	1	0	323
test066	1	0	322
test150	1	0	321
test261	1	0	319
test073	1	0	319
test317	1	0	318
test030	1	0	318
test318	1	0	318
test048	1	0	317
test314	1	0	317
test049	1	0	317
test170	1	0	317
All URIs	524	0	275

In-Stream data processing

- Replace Hive batch job with Spark
  - One hour batch job -> 5 sec batch job in 1 minute of window time
- Calculate top 100 keywords and applications

test261	1	0	319
test073	1	0	319
test317	1	0	318
test030	1	0	318
test318	1	0	318
test048	1	0	317
test314	1	0	317
test049	1	0	317
test170	1	0	317
All URIs	524	0	275

In-Stream data processing

- Replace Hive batch job with Spark
    - One hour batch job -> 5 sec batch job in 1 minute of window time
- Calculate top 100 keywords and applications
- Processing data with 530MB/s
  -> 1 mil records / sec
- Must be implemented within one month

The answer is Spark Streaming!!

- Very similar to Spark
    - DStream is very similar to RDD

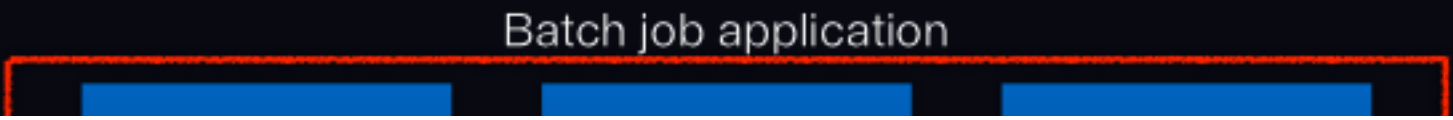
- Must be implemented within one month

The answer is Spark Streaming!!

- Very similar to Spark
    - DStream is very similar to RDD
    - Can use most functions that RDD provides(groupByKey, sortByKey)
    - Easily change batch application to in-stream processing application
- Support local environment
    - Can evaluate the application with laptop

Implementation process

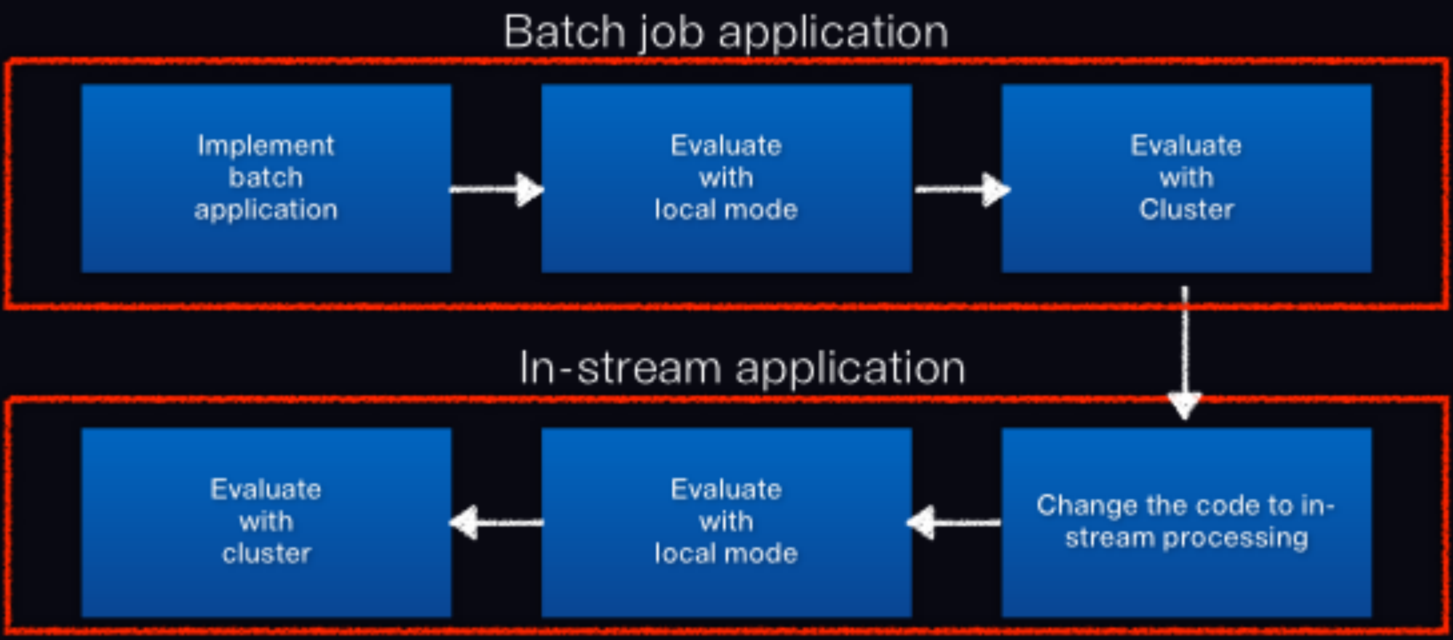
- It only takes one week to complete the process



- Support local environment
  - Can evaluate the application with laptop

Implementation process

- It only takes one week to complete the process



Batch job application

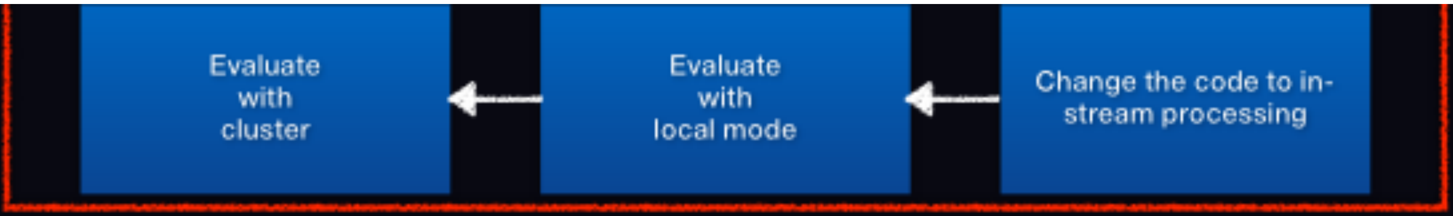
Implement batch application → Evaluate with local mode → Evaluate with Cluster

In-stream application

Change the code to in-stream processing → Evaluate with local mode → Evaluate with cluster

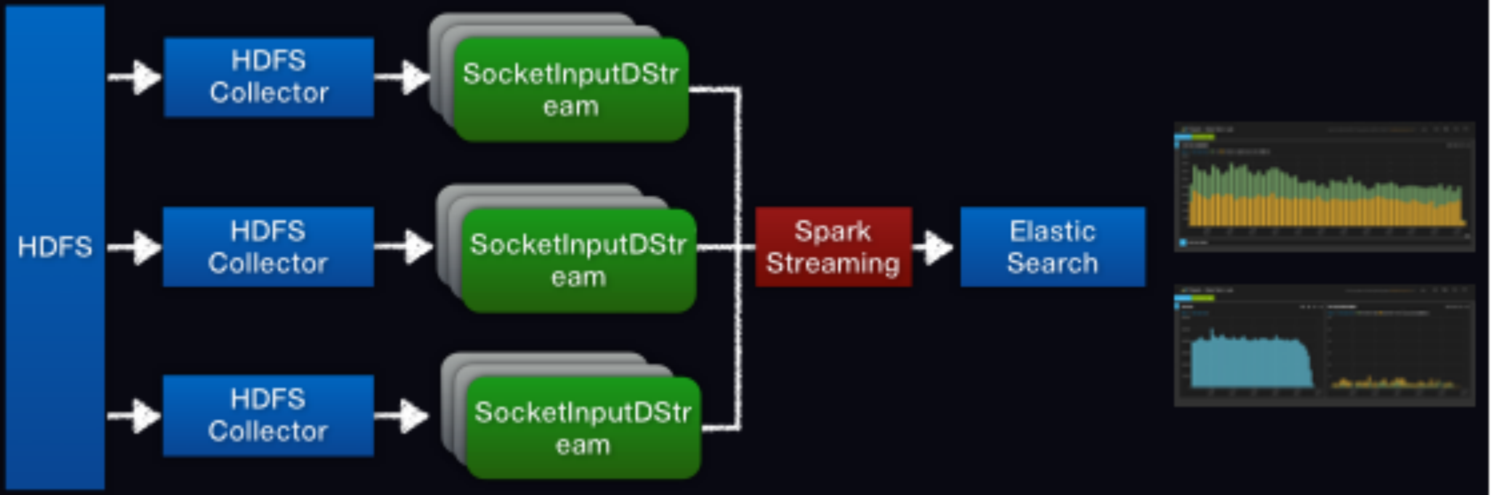
Data processing architecture

- Only “84 **lines of code”** are needed!



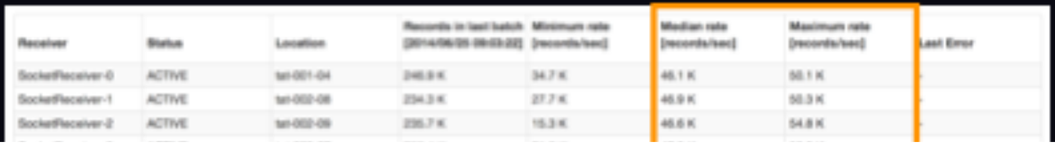
Data processing architecture

- Only “84 **lines of code”** are needed!

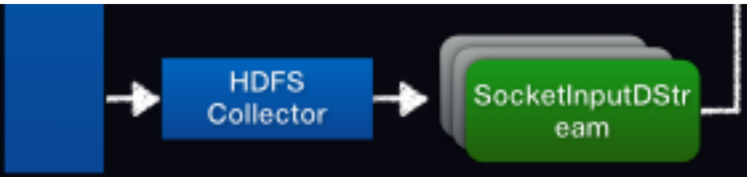


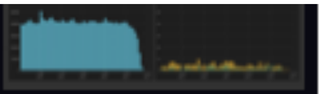
Performance

- About 3 times faster than other In-streaming processing engines.



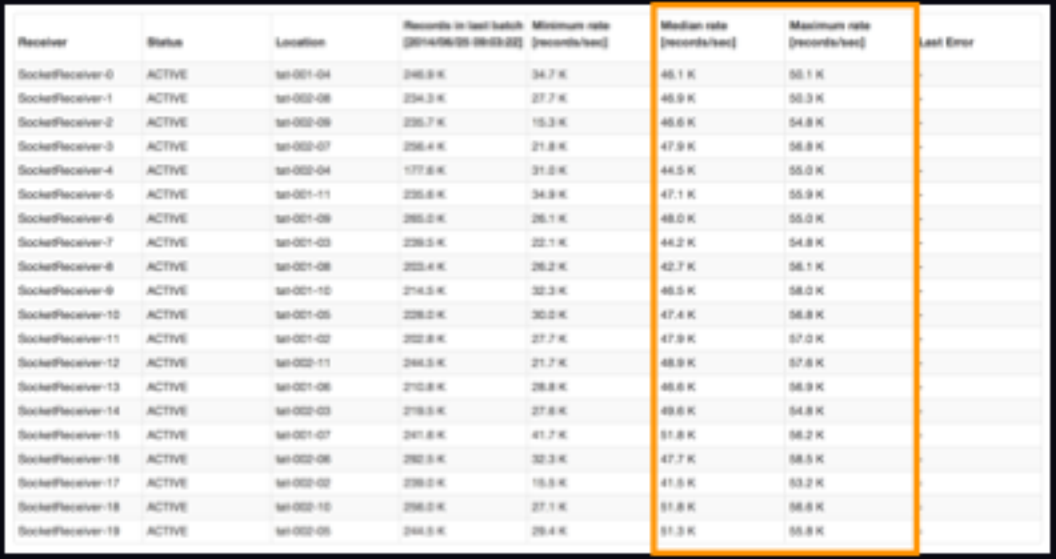
Receiver	Status	Location	Records in last batch [2014/06/25 09:03:22]	Minimum rate [records/sec]	Median rate [records/sec]	Maximum rate [records/sec]	Last Error
SocketReceiver-0	ACTIVE	tat-001-04	246.9 K	34.7 K	46.1 K	50.1 K	-
SocketReceiver-1	ACTIVE	tat-002-08	234.3 K	27.7 K	46.9 K	50.3 K	-
SocketReceiver-2	ACTIVE	tat-002-09	235.7 K	15.3 K	46.6 K	54.8 K	-



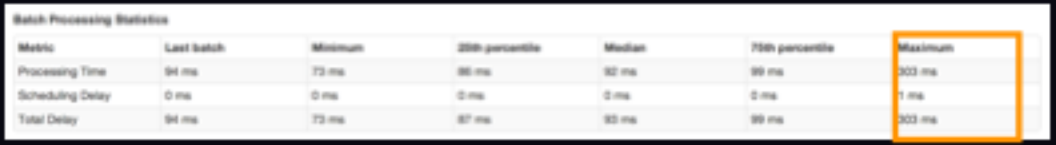


Performance

- About 3 times faster than other In-streaming processing engines.



Receiver	Status	Location	Records in last batch [2014/06/25 09:03:22]	Minimum rate [records/sec]	Median rate [records/sec]	Maximum rate [records/sec]	Last Error
SocketReceiver-0	ACTIVE	tat-001-04	246.9 K	34.7 K	46.1 K	50.1 K	-
SocketReceiver-1	ACTIVE	tat-002-08	234.3 K	27.7 K	46.9 K	50.3 K	-
SocketReceiver-2	ACTIVE	tat-002-09	235.7 K	15.3 K	46.6 K	54.8 K	-
SocketReceiver-3	ACTIVE	tat-002-07	256.4 K	21.8 K	47.9 K	56.8 K	-
SocketReceiver-4	ACTIVE	tat-002-04	177.6 K	31.0 K	44.5 K	55.0 K	-
SocketReceiver-5	ACTIVE	tat-001-11	235.6 K	34.9 K	47.1 K	55.9 K	-
SocketReceiver-6	ACTIVE	tat-001-09	265.0 K	26.1 K	48.0 K	55.0 K	-
SocketReceiver-7	ACTIVE	tat-001-03	239.5 K	22.1 K	44.2 K	54.8 K	-
SocketReceiver-8	ACTIVE	tat-001-08	203.4 K	26.2 K	42.7 K	56.1 K	-
SocketReceiver-9	ACTIVE	tat-001-10	214.5 K	32.3 K	46.5 K	58.0 K	-
SocketReceiver-10	ACTIVE	tat-001-05	228.0 K	30.0 K	47.4 K	56.8 K	-
SocketReceiver-11	ACTIVE	tat-001-02	202.8 K	27.7 K	47.9 K	57.0 K	-
SocketReceiver-12	ACTIVE	tat-002-11	244.5 K	21.7 K	48.9 K	57.6 K	-
SocketReceiver-13	ACTIVE	tat-001-06	210.8 K	28.8 K	46.6 K	56.9 K	-
SocketReceiver-14	ACTIVE	tat-002-03	219.5 K	27.6 K	49.6 K	54.8 K	-
SocketReceiver-15	ACTIVE	tat-001-07	241.6 K	41.7 K	51.8 K	56.2 K	-
SocketReceiver-16	ACTIVE	tat-002-06	262.5 K	32.3 K	47.7 K	58.5 K	-
SocketReceiver-17	ACTIVE	tat-002-02	239.0 K	15.5 K	41.5 K	53.2 K	-
SocketReceiver-18	ACTIVE	tat-002-10	256.0 K	27.1 K	51.8 K	56.6 K	-
SocketReceiver-19	ACTIVE	tat-002-05	244.5 K	29.4 K	51.3 K	55.8 K	-

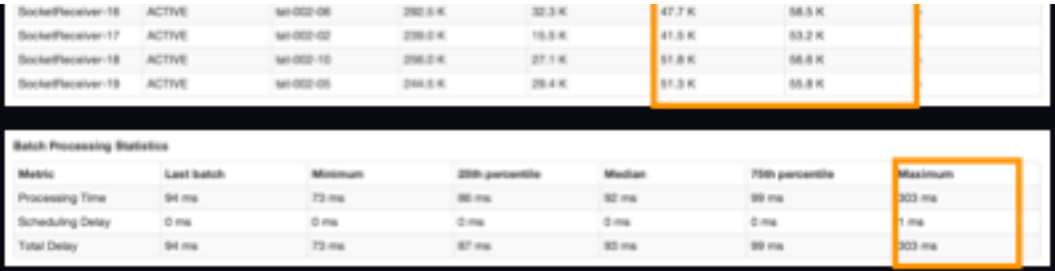


Batch Processing Statistics

Metric	Last batch	Minimum	25th percentile	Median	75th percentile	Maximum
Processing Time	94 ms	73 ms	86 ms	92 ms	99 ms	303 ms
Scheduling Delay	0 ms	0 ms	0 ms	0 ms	0 ms	1 ms
Total Delay	94 ms	73 ms	87 ms	93 ms	99 ms	303 ms

Lessons learned

- We can implement **OLTP** style systems and **In-stream data processing** with Spark and Shark.
- Win-win between community and company



SocketReceiver-16	ACTIVE	tat-002-06	292.5 K	32.3 K	47.7 K	58.5 K
SocketReceiver-17	ACTIVE	tat-002-02	239.0 K	15.5 K	41.5 K	53.2 K
SocketReceiver-18	ACTIVE	tat-002-10	256.0 K	27.1 K	51.8 K	58.6 K
SocketReceiver-19	ACTIVE	tat-002-05	244.5 K	29.4 K	51.3 K	55.8 K

Batch Processing Statistics

Metric	Last batch	Minimum	25th percentile	Median	75th percentile	Maximum
Processing Time	94 ms	73 ms	86 ms	92 ms	99 ms	303 ms
Scheduling Delay	0 ms	0 ms	0 ms	0 ms	0 ms	1 ms
Total Delay	94 ms	73 ms	87 ms	93 ms	99 ms	303 ms

Lessons learned

- We can implement **OLTP** style systems and **In-stream data processing** with Spark and Shark.
- Win-win between community and company
  - Test in real working cluster
    - Finding some bugs and function requirement, as well.
    - mainly focusing on low latency query.

Q
&
A