

The Distributed Database Based on Kudu

Shunda Lin

Outline

- Motivation
- Introduction of Kudu
- Deployment and Configuration
- Query Test
- Conclusion

Outline

- Motivation
- Introduction of Kudu
- Deployment and Configuration
- Query Test
- Conclusion

Motivation



Outline

- Motivation
- Introduction of Kudu
- Deployment and Configuration
- Query Test
- Conclusion

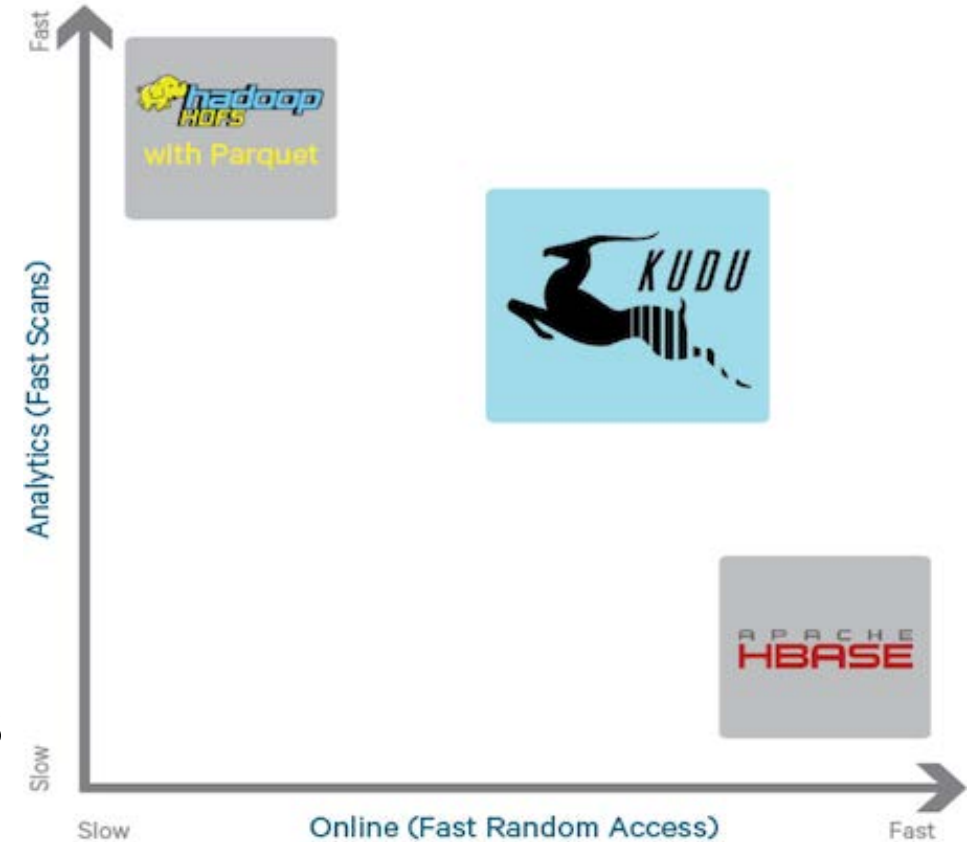


Traditional System

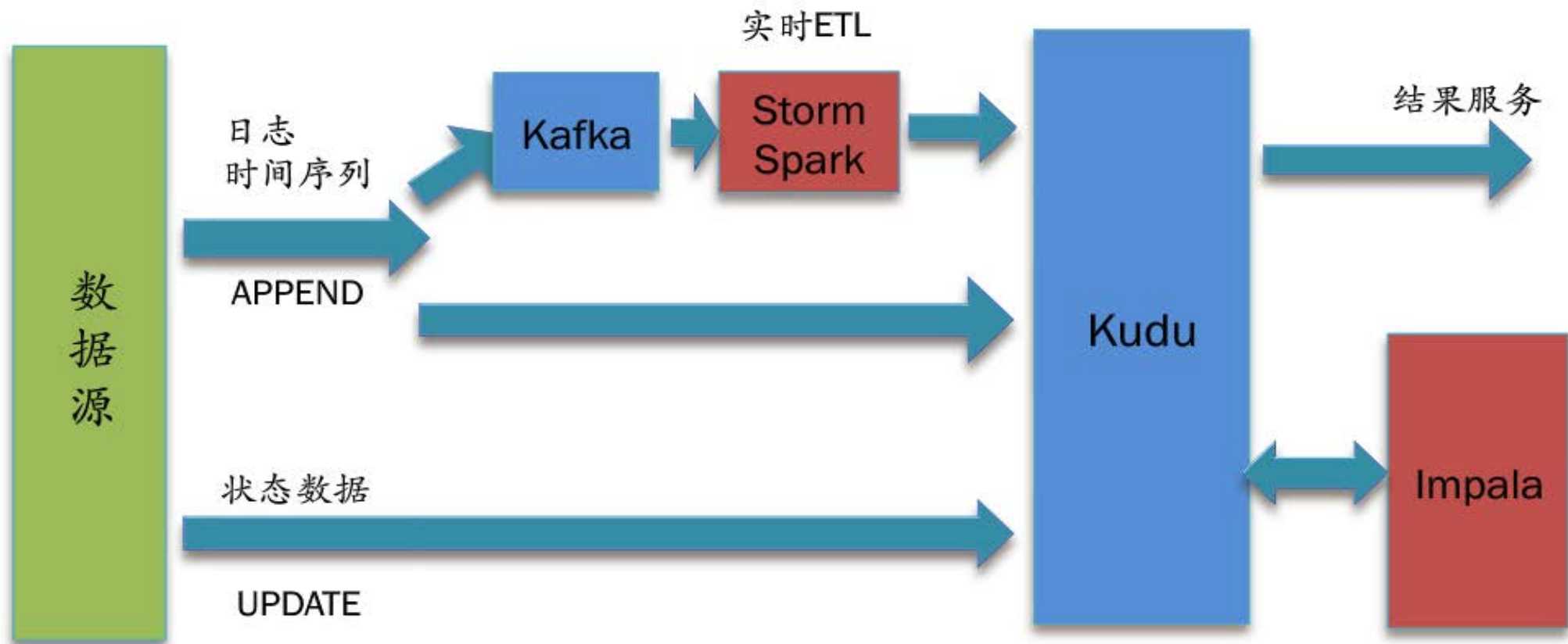
- The application system needs to reverse the data between the real-time and offline systems, and write a complex code.
- Systems are complex, need various backups, security policies, and monitoring systems
- There is a delay in the transformation from real-time system to offline system for OLAP analysis
- It requires expensive price to change or rewrite the backward data when data in the past has been filed

Kudu-Fast Analytics on Fast Data

- Released by Cloudera in 2015
- Used for OLAP
- High performance for both data scanning and random access
- Simplifying complex hybrid architectures



KUDU数据流

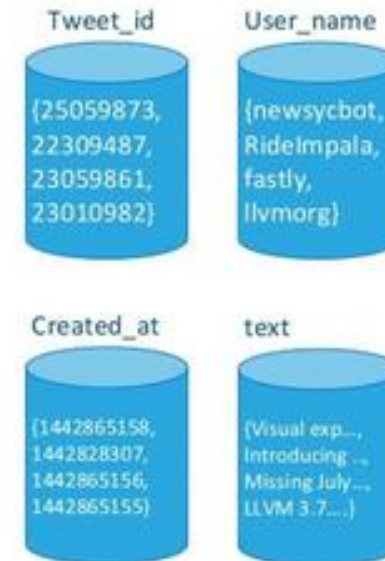


Architectures and Design

- Super-fast Columnar Storage

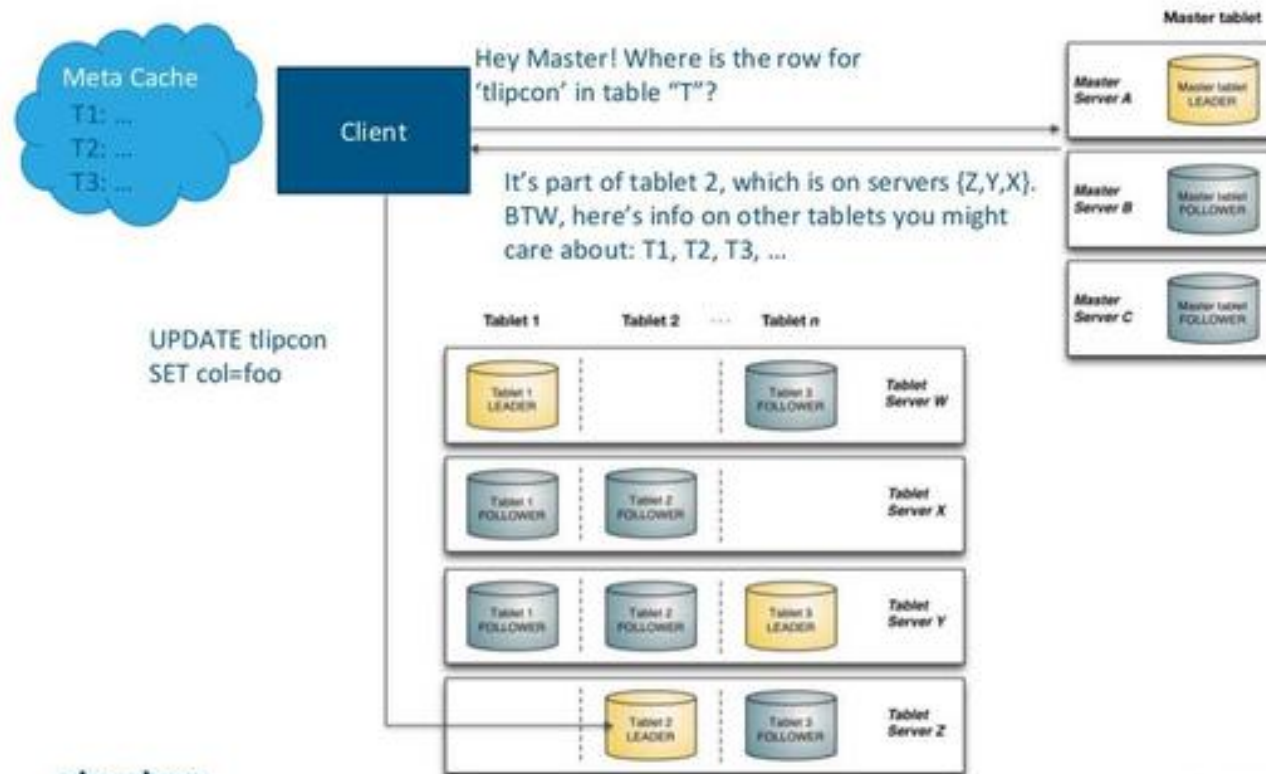
Columnar Storage

Twitter Firehose Table			
tweet_id	user_name	created_at	text
INT64	STRING	TIMESTAMP	STRING
23059873	newsycbot	1442865158	Visual Explanation of the Raft Consensus Algorithm http://bit.ly/1DOUac0 (cmts http://bit.ly/1HKmjfc)
22309487	Ridelpala	1442828307	Introducing the ibis project: for the Python experience at Hadoop Scale
23059861	fastly	1442865156	Missed July's SF @papers_we_love? You can now watch @el_bhs talk about @google's globally-distributed database: http://fastly.us/1eVz8MM
23010982	llvmorg	1442865155	LLVM 3.7 is out! Get it while it's HOT! http://llvm.org/releases/download.html#3.7.0



Architectures and Design

- Distribution and Fault Tolerance



Outline

- Motivation
- Introduction of Kudu
- Deployment and Configuration
- Query Test
- Conclusion

Deployment

- master : slave2 (192.168.0.134)
- tserver : slave1 (192.168.0.135)
 slave2 (192.168.0.134)
 slave3 (192.168.0.100)

```
sudo apt-get install kudu                # Base Kudu files
sudo apt-get install kudu-master        # Service scripts for managing kudu-master
sudo apt-get install kudu-tserver      # Service scripts for managing kudu-tserver
sudo apt-get install libkuduclient0    # Kudu C++ client shared library
sudo apt-get install libkuduclient-dev # Kudu C++ client SDK
```

Data Persistence

- MySQL->HDFS->Kudu

- Sqoop

a command-line interface application for transferring data between relational databases and Hadoop

- Spark

an open-source cluster-computing framework

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月10日 20:38:43	0	0 B	.sparkStaging
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月21日 21:03:45	0	0 B	.temp
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月17日 10:56:58	0	0 B	AuthorFieldCount
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月8日 22:49:01	0	0 B	Authors
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月21日 21:03:45	0	0 B	FOSReferencesCount
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月8日 23:07:09	0	0 B	FieldsOfStudy
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月17日 9:56:09	0	0 B	Linsd
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月24日 12:07:44	0	0 B	Liuyuezhou
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月4日 15:54:03	0	0 B	PaperAuthorAffiliations
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月8日 22:05:45	0	0 B	PaperKeywords
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月8日 23:08:57	0	0 B	PaperRecommenderList
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月4日 12:09:37	0	0 B	PaperReferences
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月8日 21:23:47	0	0 B	PaperSciReferencesCount
drwxr-xr-x	hadoop	supergroup	0 B	2017年4月1日 23:51:59	0	0 B	Venues_Network

Data Persistence on Kudu

- spark-shell
- design table
- create table
- insert data



MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

- FOSReferencesCountSmall
- intro
- Journals
- NewKeywordNameInfo
- PaperAuthorAffiliations
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- PaperCitationCount
- PaperIDTmp
- PaperKeywords
- PaperRecommenderList
- PaperReferenceCount
- PaperReferences

Information

Table: PaperAuthorAffiliations

Columns:

PaperID varchar(10)
AuthorID varchar(10)
AffiliationID varchar(10)
 OriginalAffiliationName varchar(500)
 NormalizedAffiliationName varchar(500)
 AuthorSequenceNumber int(4)

Object Info Session

1 SELECT count(*) FROM `mag-new-160205`.PaperAuthorAffiliations

Limit to 1000 rows

Result Grid

Filter Rows: Export: Wrap Cell Content:

count(*)

338222414

Result 2 x

Output

Action Output

Time Action

1 18:00:02 SELECT * FROM `mag-new-160205`.PaperAuthorAffiliations LIMIT 0, 1000

2 20:12:37 SELECT * FROM `mag-new-160205`.FOSReferencesCount LIMIT 0, 1000

3 20:12:42 select PaperReferences.PaperID,count(*) from PaperReferences inner join

4 20:16:03 SELECT * FROM `mag-new-160205`.PaperAuthorAffiliations LIMIT 0, 1000

5 20:16:24 SELECT count(*) FROM `mag-new-160205`.PaperAuthorAffiliations LIMIT 0, 1000



ssh://202.120.36.28:23

要添加当前会话, 点击左侧的箭头按钮。

1 新建会话

```
at org.apache.spark.sql.Dataset$$anonfun$foreachPartition$1.apply$mcV$sp(Dataset
at org.apache.spark.sql.Dataset$$anonfun$foreachPartition$1.apply(Dataset.scala
at org.apache.spark.sql.Dataset$$anonfun$foreachPartition$1.apply(Dataset.scala
at org.apache.spark.sql.execution.SQLExecution$.withNewExecutionId(SQLExecution
at org.apache.spark.sql.Dataset.withNewExecutionId(Dataset.scala:2546)
at org.apache.spark.sql.Dataset.foreachPartition(Dataset.scala:2126)
at org.apache.kudu.spark.kudu.KuduContext.writeRows(KuduContext.scala:192)
at org.apache.kudu.spark.kudu.KuduContext.insertRows(KuduContext.scala:145)
... 60 elided
```

```
scala> sqlContext.read.options(kuduOptions).kudu.registerTempTable("kuduTable")
warning: there was one deprecation warning; re-run with -deprecation for details
17/04/25 20:15:35 ERROR TabletClient: [Peer master-192.168.0.134:7051] Tablet ser
17/04/25 20:15:35 WARN TabletClient: [Peer master-192.168.0.134:7051] sending an
terRegistration, tablet=null, attempt=0, DeadlineTracker(timeout=0, elapsed=0), T
2.168.0.134:7051)
```

```
scala> val startTime = System.currentTimeMillis()
startTime: Long = 1493122536065
```

```
scala> sqlContext.sql("SELECT count(*) FROM kuduTable").show
+-----+
| count(1)|
+-----+
|338222355|
+-----+
```

```
scala> System.currentTimeMillis() - startTime
res5: Long = 36858
```

scala>

scala>

scala> :quit

hadoop@iiot-cluster1-master1:~/Linsd\$

仅将文本发送到当前选项卡

ssh://202.120.36.28:23

...	1000 row(s) returned	160.234 sec / 0.000 sec
...	1000 row(s) returned	0.031 sec / 0.047 sec
...	1 row(s) returned	0.015 sec / 0.000 sec

Outline

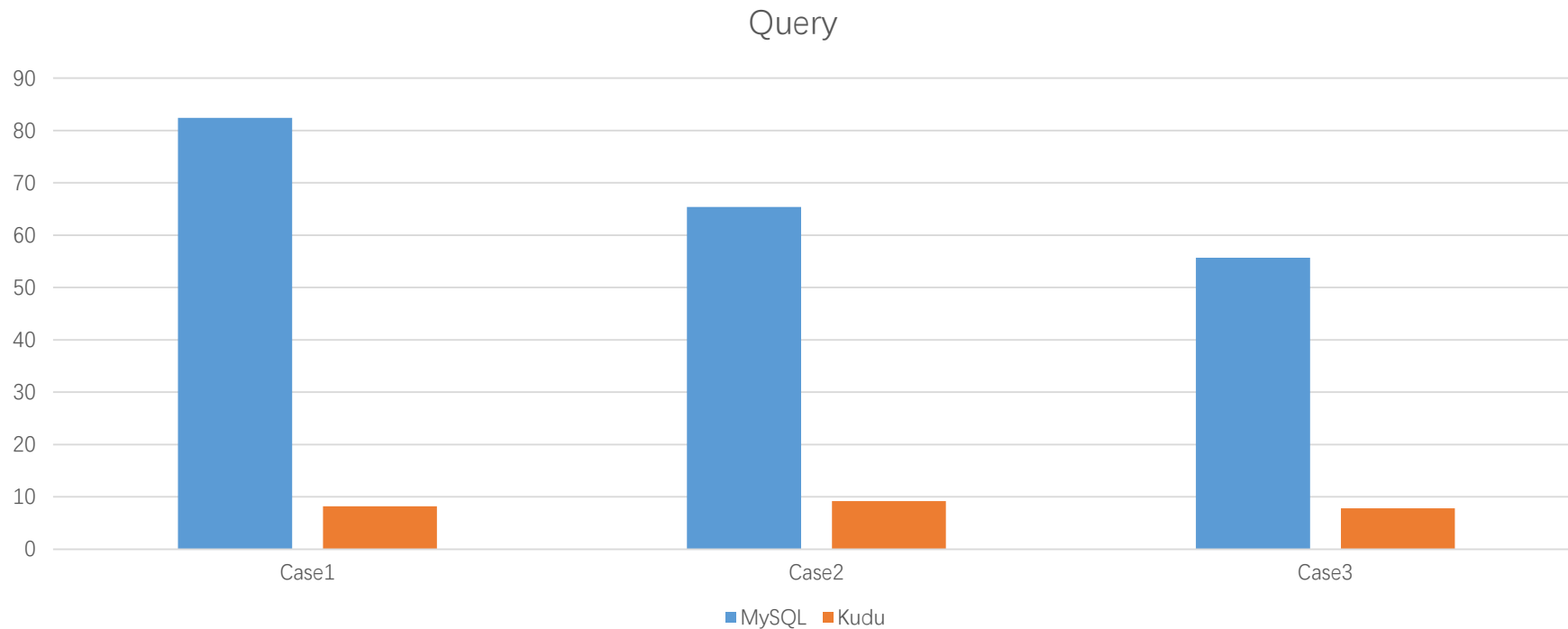
- Motivation
- Introduction of Kudu
- Deployment and Configuration
- Query Test
- Conclusion

Query Test

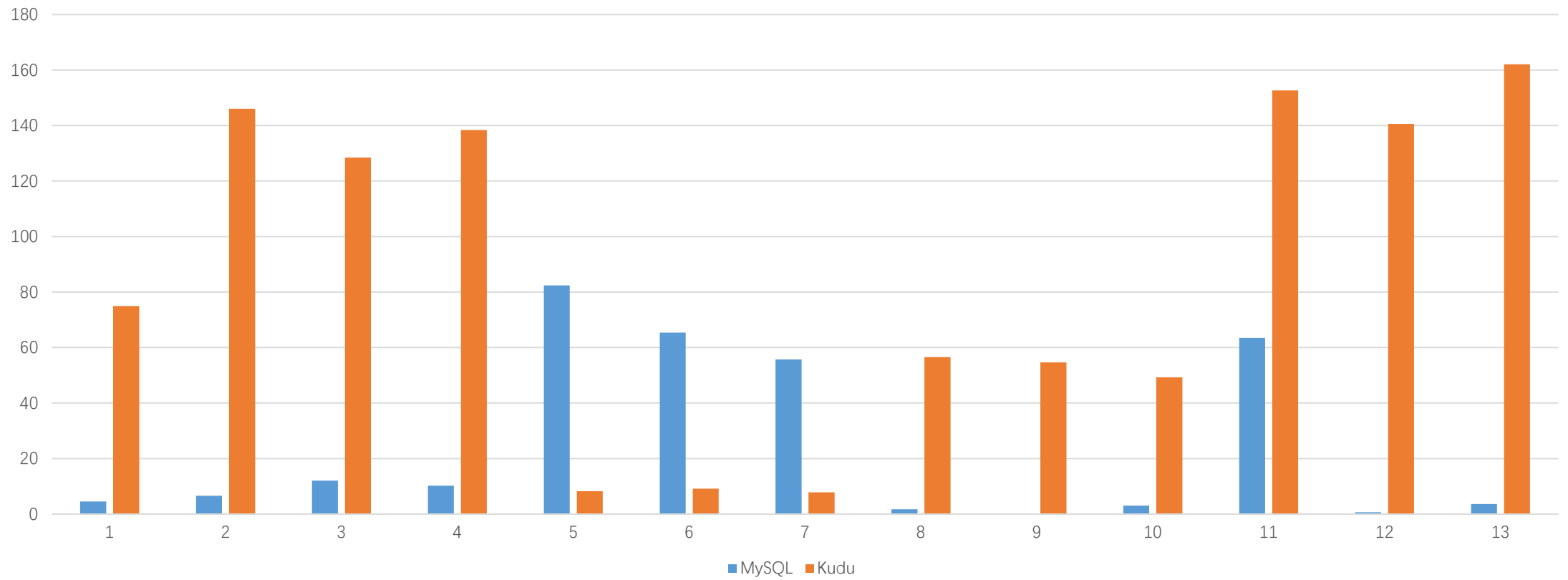
- 从领域相关表中提取出1000个与某领域最为相关的领域之间的相关关系

```
select FOSID as Source, FOSReferencesCount.FOSReference as Target, Similarity/10000000 as Weight from
(select FOSReference
  from `FOSReferencesCount`
  where `FOSID` = '0271BC14'
  order by `Similarity` desc
  limit 1000) e1,
(select FOSReference
  from `FOSReferencesCount`
  where `FOSID` = '0271BC14'
  order by `Similarity` desc
  limit 1000) e2,
FOSReferencesCount
where e1.`FOSReference` = `FOSReferencesCount`.FOSID and e2.`FOSReference` = `FOSReferencesCount`.FOSReference;
```

FOSID	Computer Science (0271BC14)	Ethnic studies (03D2C4FF)	Data Structure (09ACCB7D)
MySQL	82.4s	65.4s	55.7s
Kudu	8.23s	9.175s	7.821s



Query Test



Outline

- Motivation
- Introduction of Kudu
- Deployment and Configuration
- Query Test
- Conclusion

Q&A