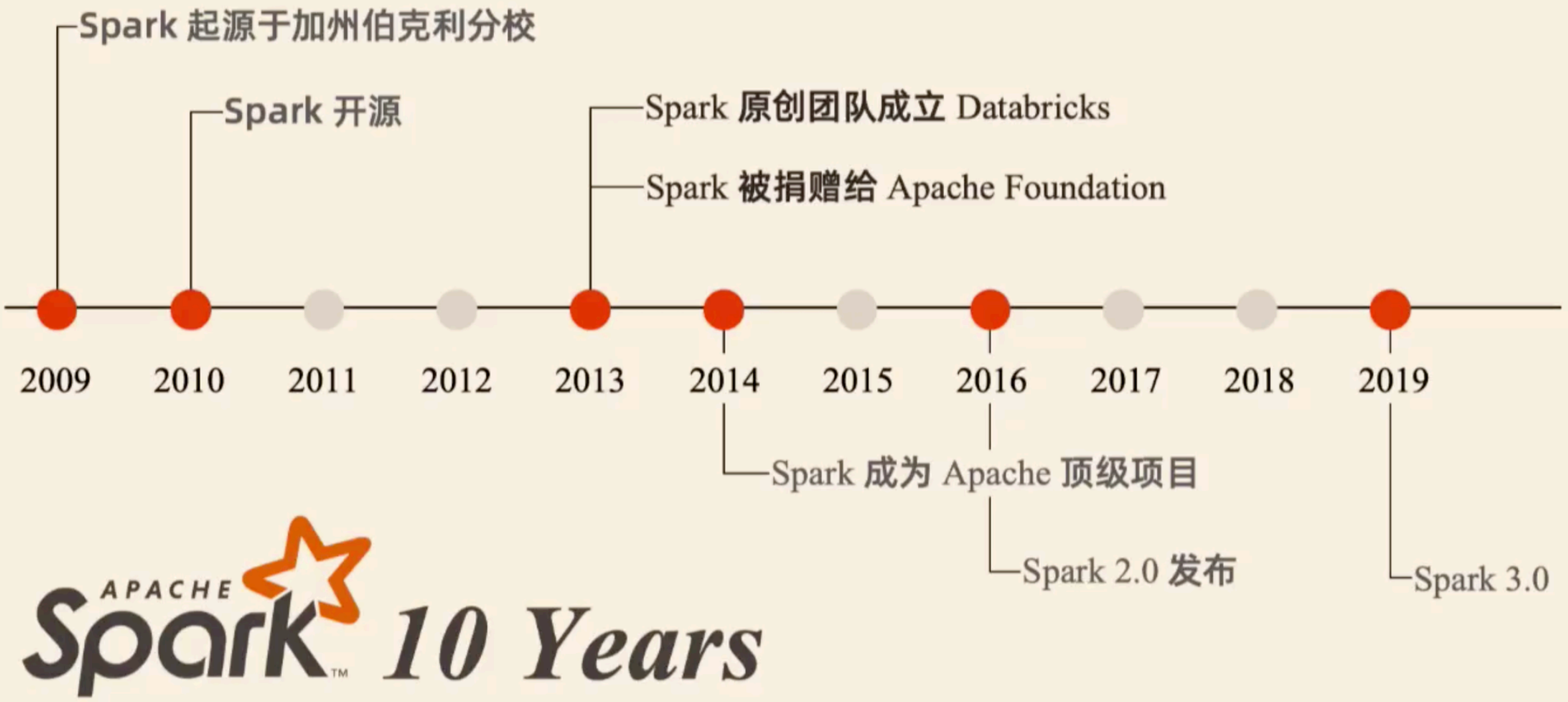
Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks



APACHE Spark™ *10 Years*

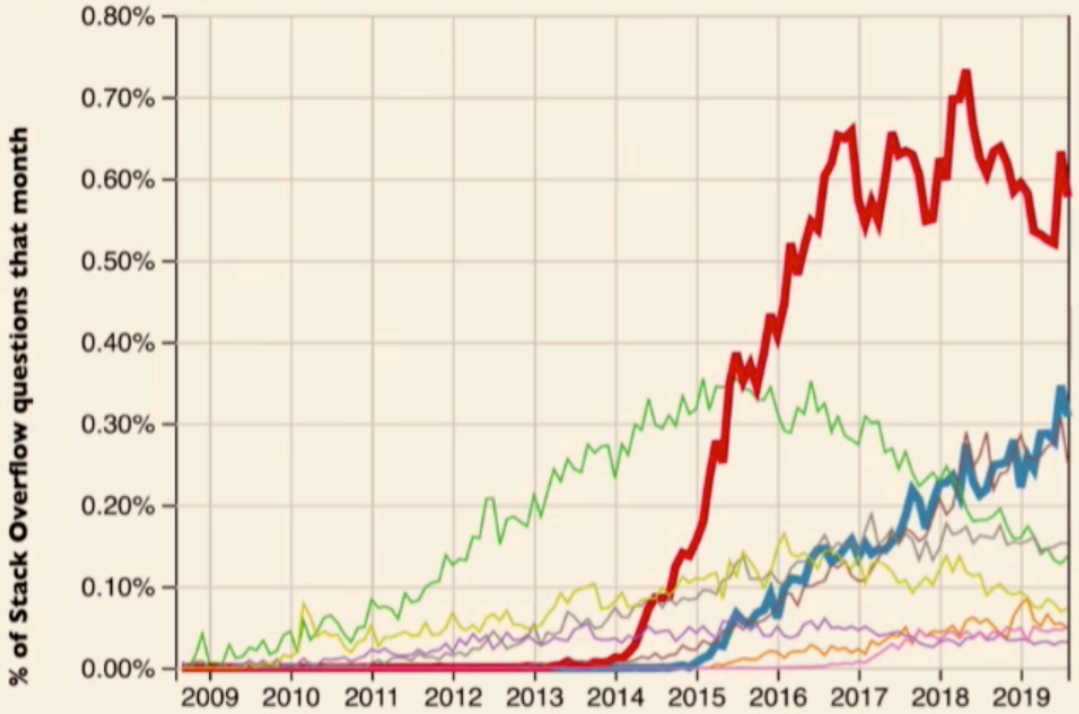
Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

stackoverflow

当前 排名

- **Apache Spark**
- **PySpark**
- Apache Kafka
- Apache Hive
- Apache Hadoop
- Apache Cassandra
- Apache Beam
- Apache Flink
- Apache Hbase



10年 累计排名

- **Apache Spark**
- Apache Hadoop
- **PySpark**
- Apache Kafka
- Apache Hive
- Apache Cassandra
- Apache Flink
- Apache Hbase
- Apache Beam

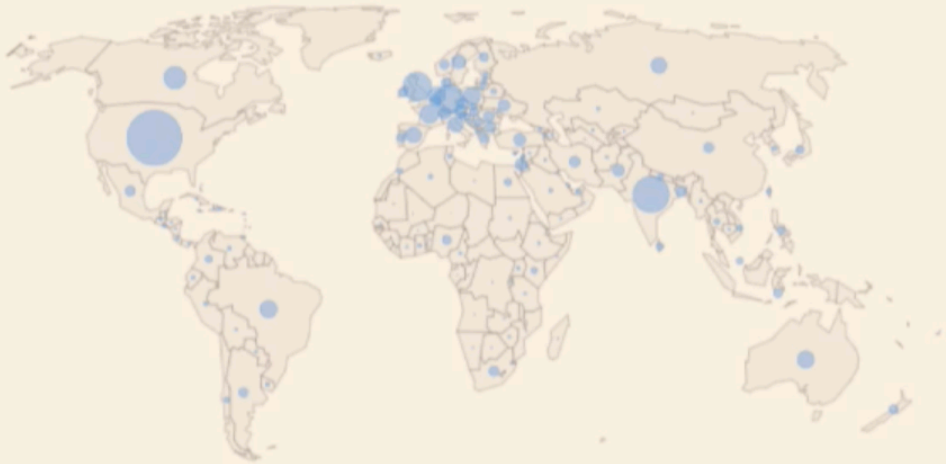
Stackoverflow: Q & A for professional and enthusiast programmers, 9.4 m visits/day

Source: https://insights.stackoverflow.com/trends?tags=pyspark%2Capache-flink%2Chadoop%2Capache-spark%2Chbase%2Capache-kafka%2Capache-beam%2Chive%2Ccassandra

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

stackoverflow

开发者调查报告 2019

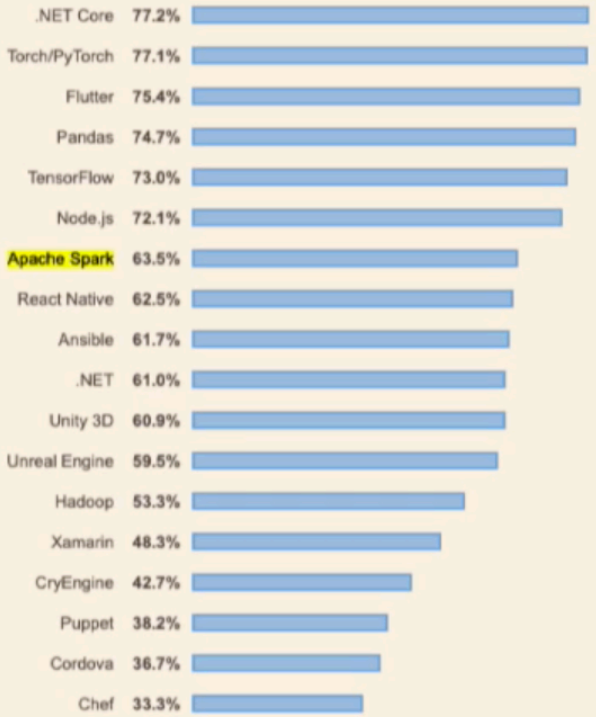


88,615 responses

最受欢迎的平台之一：Apache Spark

Most Loved, Dreaded, and Wanted Other Frameworks, Libraries, and Tools

Loved | Dreaded | Wanted



% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it


Alibaba Group
阿里巴巴集团
APSARA
云 栖 大 会
databricks
APACHE
Spark™ 3.0

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

Will resolve 2000+ JIRAs

Dynamic Partition Pruning

Adaptive Query Execution

Spark Graph

Accelerator-aware Scheduling

Spark on Kubernetes

Data Source API with Catalog Supports

ANSI SQL Compliance

SQL Hints

Vectorization in SparkR

JDK 11

Hadoop 3

Scala 2.12

Query Optimization in Spark 2.X

Missing Statistics (一次性计算 ETL workloads)

Out-of-date Statistics (存储与计算分离)

Misestimated Costs (多样部署环境 + UDFs)

Query Optimization

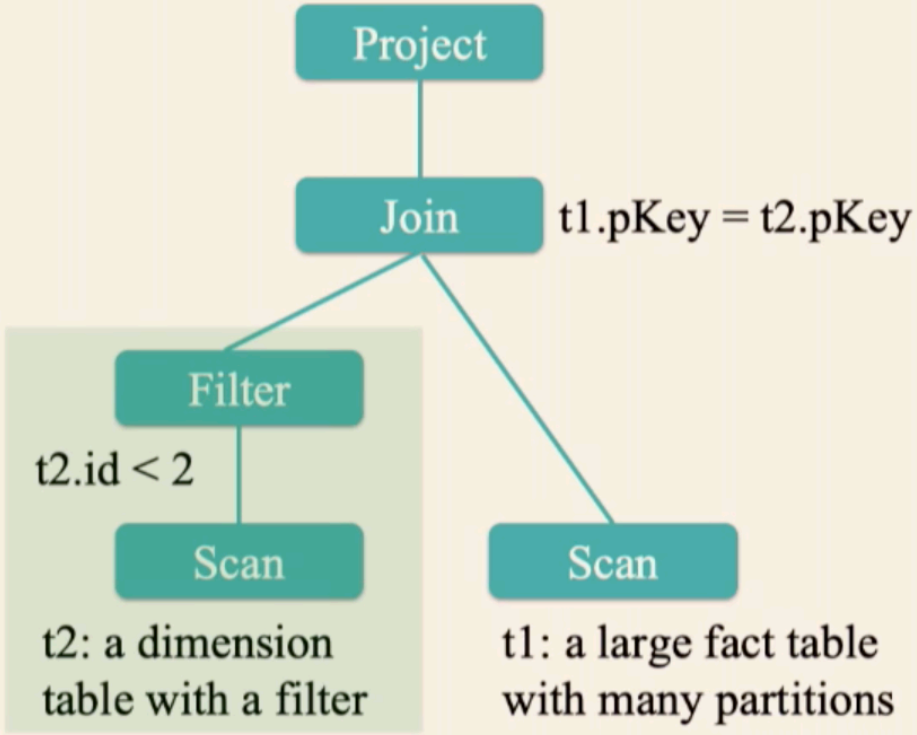
Spark 1.x, Rule

Spark 2.x, Rule + Cost

Spark 3.0, Rule + Cost + Runtime

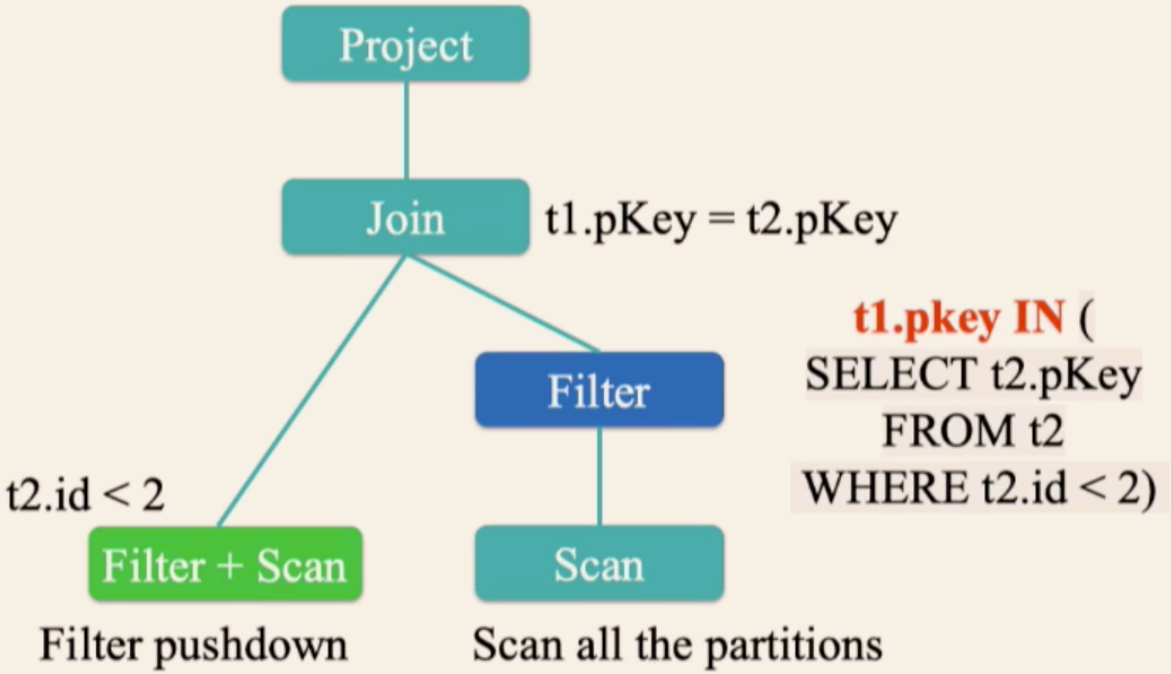
Dynamic Partition Pruning

```
SELECT t1.id, t2.pKey
FROM   t1
  JOIN t2
    ON t1.pKey = t2.pKey
    AND t2.id < 2
```

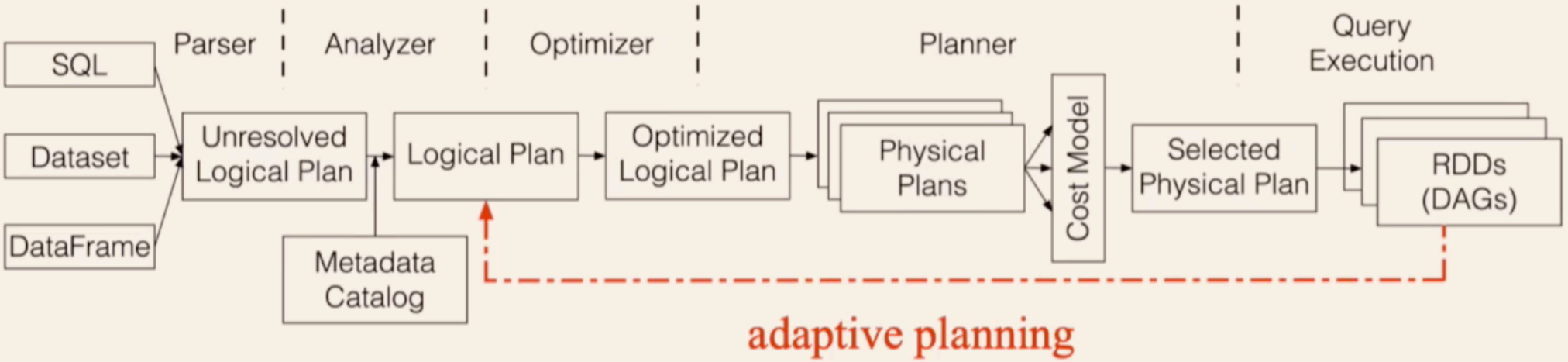


Dynamic Partition Pruning

```
SELECT t1.id, t2.pKey
FROM   t1
  JOIN t2
    ON t1.pKey = t2.pKey
    AND t2.id < 2
```

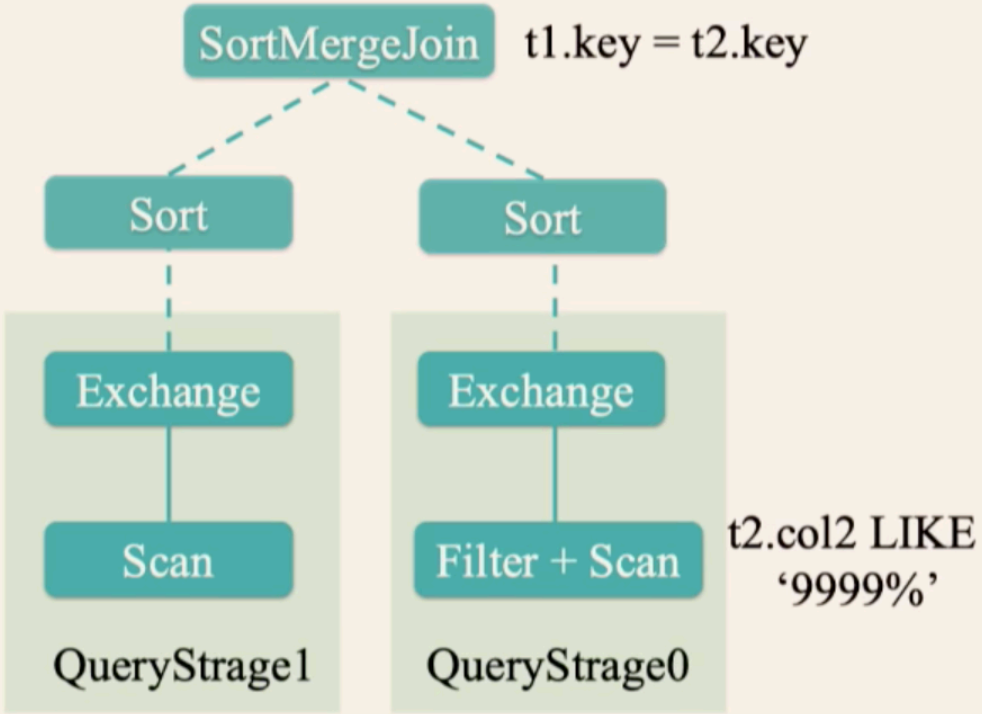


Adaptive Query Execution



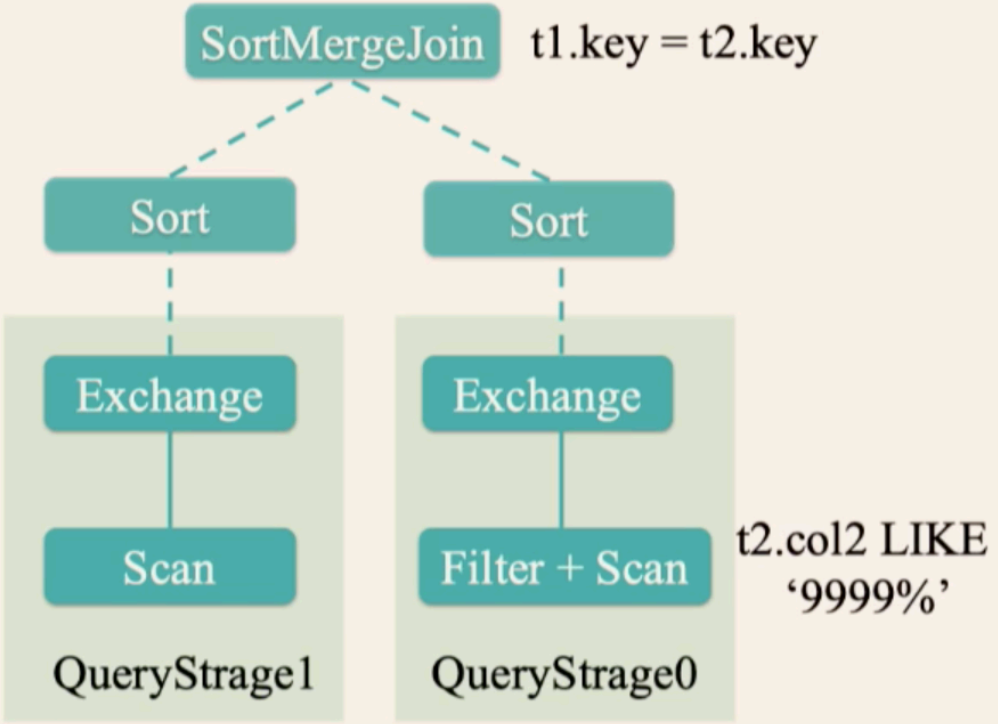
Adaptive Query Execution

```
SELECT *
FROM   t1
  JOIN t2
    ON t1.key = t2.key
    AND t2.col2 LIKE '9999%'
```



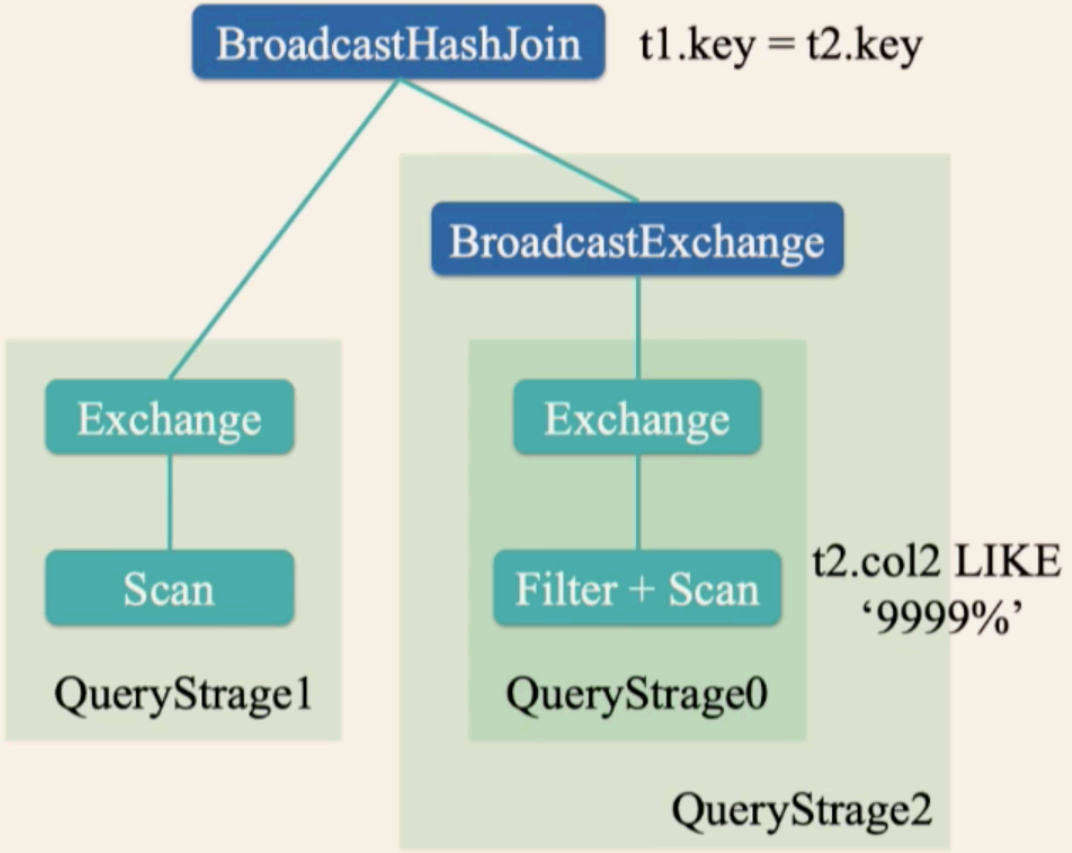
Adaptive Query Execution

```
SELECT *
FROM   t1
  JOIN t2
    ON t1.key = t2.key
    AND t2.col2 LIKE '9999%'
```



Adaptive Query Execution

```
SELECT *
FROM   t1
  JOIN t2
    ON t1.key = t2.key
    AND t2.col2 LIKE '9999%'
```




Alibaba Group
阿里巴巴集团
APSARA
云 栖 大 会
databricks
APACHE
Spark™
生 态

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

DOI:10.1145/2934664

This open source computing framework unifies streaming, batch, and interactive big data workloads to unlock new applications.

BY MATEI ZAHARIA, REYNOLD S. XIN, PATRICK WENDELL, TATHAGATA DAS, MICHAEL ARMBRUST, ANKUR DAVE, XIANGRUI MENG, JOSH ROSEN, SHIVARAM VENKATARAMAN, MICHAEL J. FRANKLIN, ALI GHODSI, JOSEPH GONZALEZ, SCOTT SHENKER, AND ION STOICA

Apache Spark: A Unified Engine for Big Data Processing

THE GROWTH OF data volumes in industry and research poses tremendous opportunities, as well as tremendous computational challenges. As data sizes have outpaced the capabilities of single machines, users have needed new systems to scale out computations to multiple nodes. As a result, there has been an explosion of new cluster programming models targeting diverse computing workloads.[1,4,7,10] At first, these models were relatively specialized, with new models developed for new workloads; for example, MapReduce[4] supported batch processing, but Google also developed Dremel[13] for interactive SQL queries and Pregel[11] for iterative graph algorithms. In the open source Apache Hadoop stack, systems like Storm[1] and Impala[9] are also specialized. Even in the relational database world, the trend has been to move away from "one-size-fits-all" systems.[15] Unfortunately, most big data applications need to combine many different processing types. The very nature of "big data" is that it is diverse and messy; a typical pipeline will need MapReduce-like code for data loading, SQL-like queries, and iterative machine learning. Specialized engines can thus create both complexity and inefficiency; users must stitch together disparate systems, and some applications simply cannot be expressed efficiently in any engine.

In 2009, our group at the University of California, Berkeley, started the Apache Spark project to design a unified engine for distributed data processing. Spark has a programming model similar to MapReduce but extends it with a data-sharing abstraction called "Resilient Distributed Datasets," or RDDs.[15] Using this simple extension, Spark can capture a wide range of processing workloads that previously needed separate engines, including SQL, streaming, machine learning, and graph processing[1,26,5] (see Figure 1). These implementations use the same optimizations as specialized engines (such as column-oriented processing and incremental updates) and achieve similar performance but run as libraries over a common engine, making them easy and efficient to compose. Rather than being specific

» key insights

- A simple programming model can capture streaming, batch, and interactive workloads and enable new applications that combine them.
- Apache Spark applications range from finance to scientific data processing and combine libraries for SQL, machine learning, and graphs.
- In six years, Apache Spark has grown to 1,000 contributors and thousands of deployments.

APACHE Spark™

Dr. Ion Stoica & Dr. Matei Zaharia

BUSINESS INTELLIGENCE | BIG DATA ANALYTICS | UNIFYING DATA + AI

Data Warehouses

Reactive Analytics - What just happened?

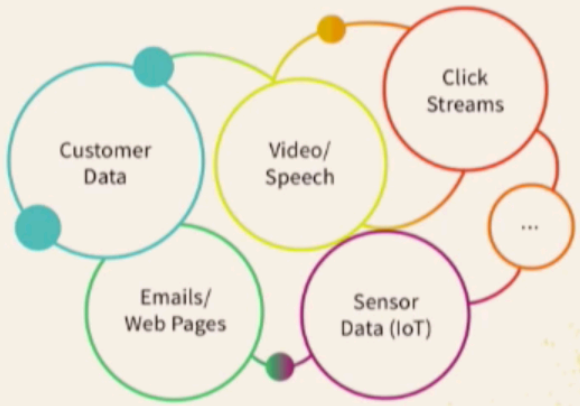
Predictive Analytics

Proactive Analytics - Predict what's going to happen

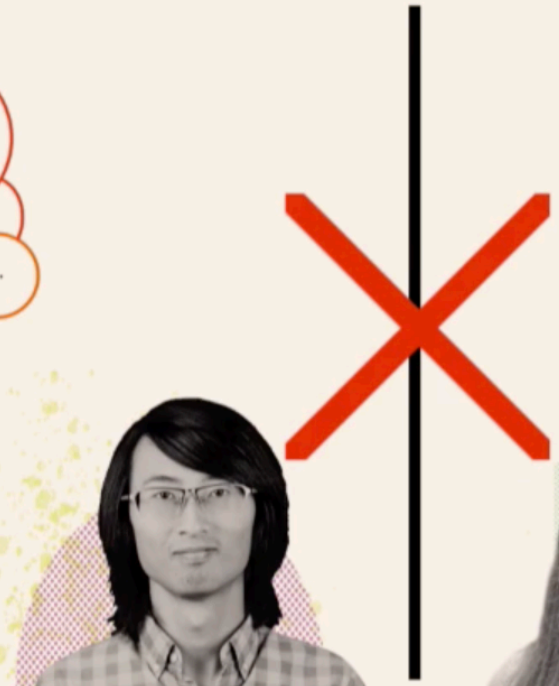
Autonomous Decisions

Advanced ML / AI - Make the decision for me

Spark as a unifying technology - Data processing at scale + analytics + AI


Customer Data
Video/ Speech
Click Streams
...
Emails/ Web Pages
Sensor Data (IoT)

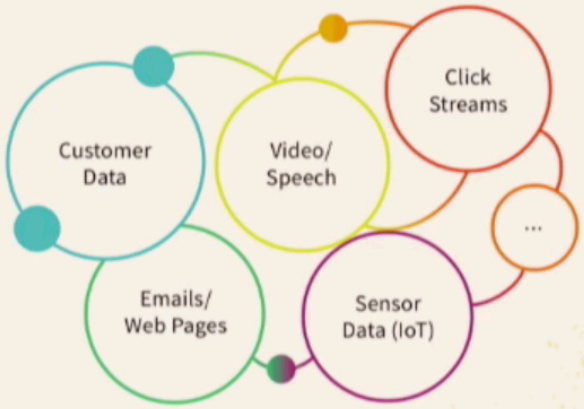
数据工程



数据科学

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks



APACHE Spark™

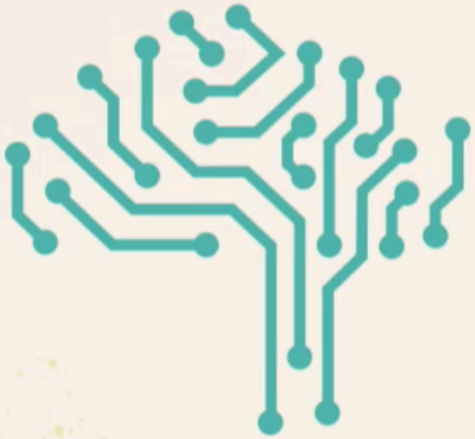
Accelerate innovation by unifying data science and engineering

数据工程

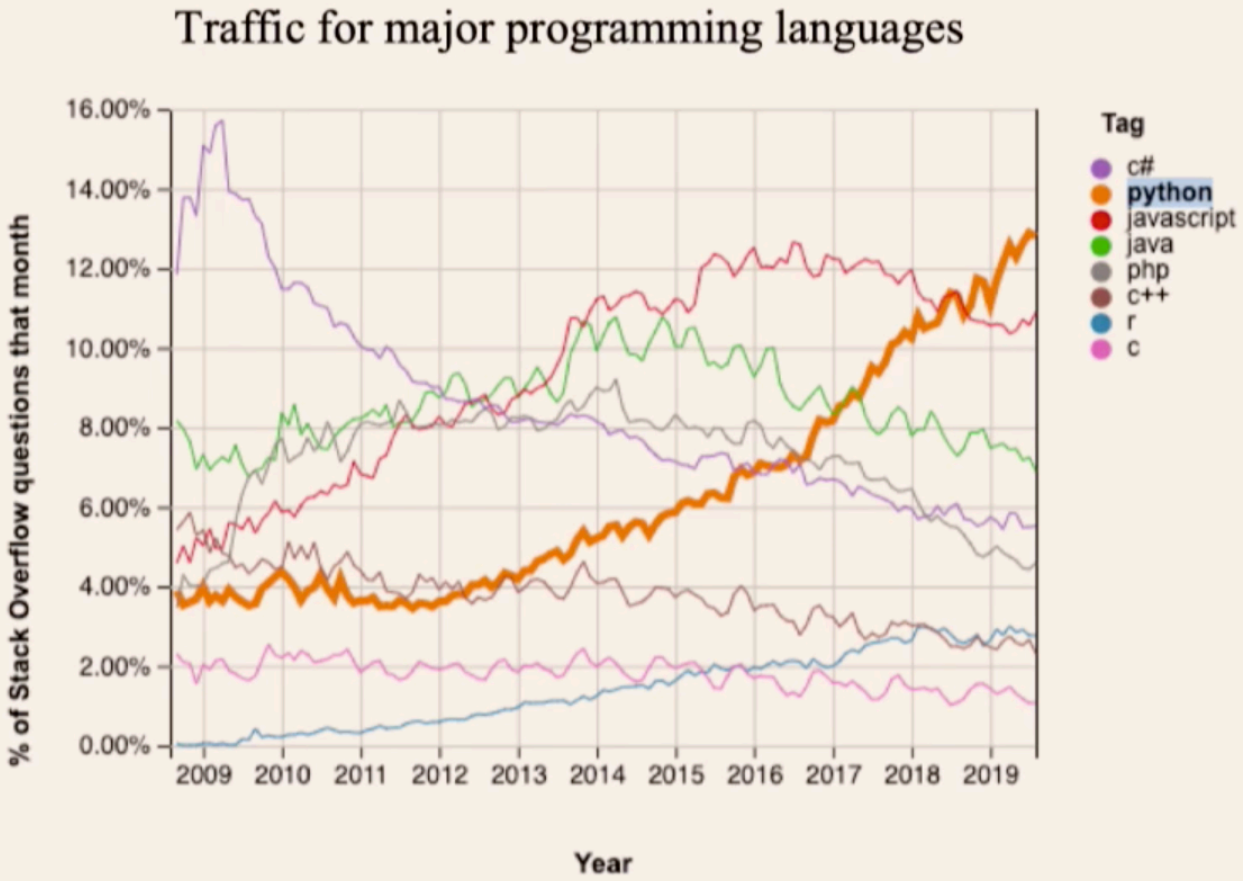
数据科学

Alibaba Group 阿里巴巴集团 | APSARA CONFERENCE 云栖大会

databricks



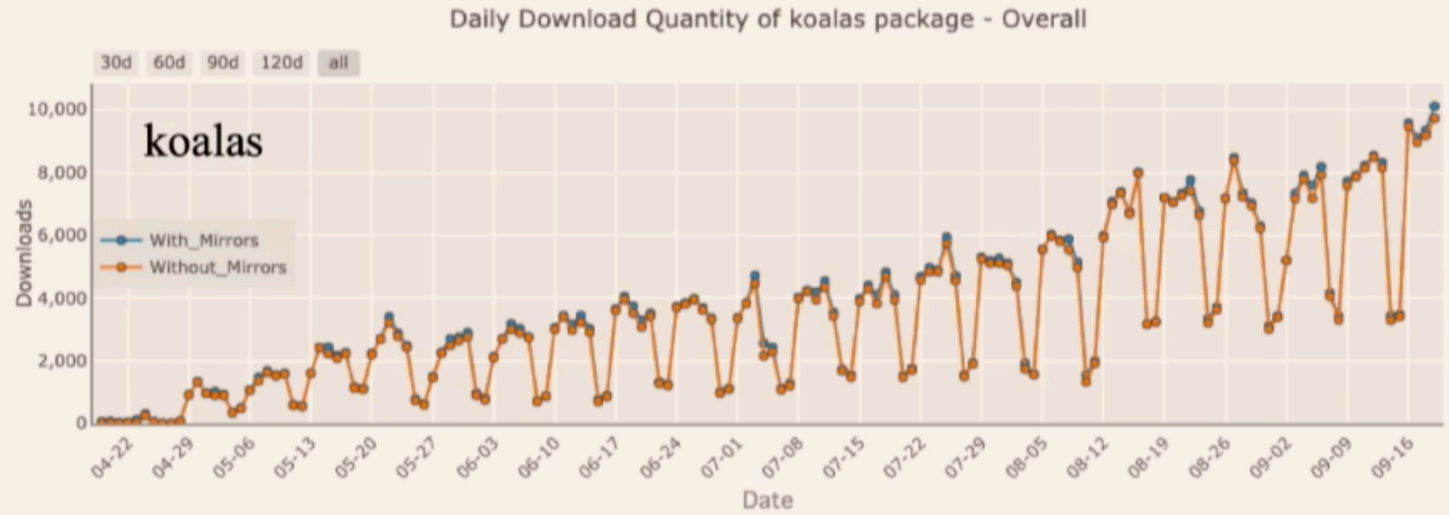
数据科学



pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

Image source: Stack Overflow



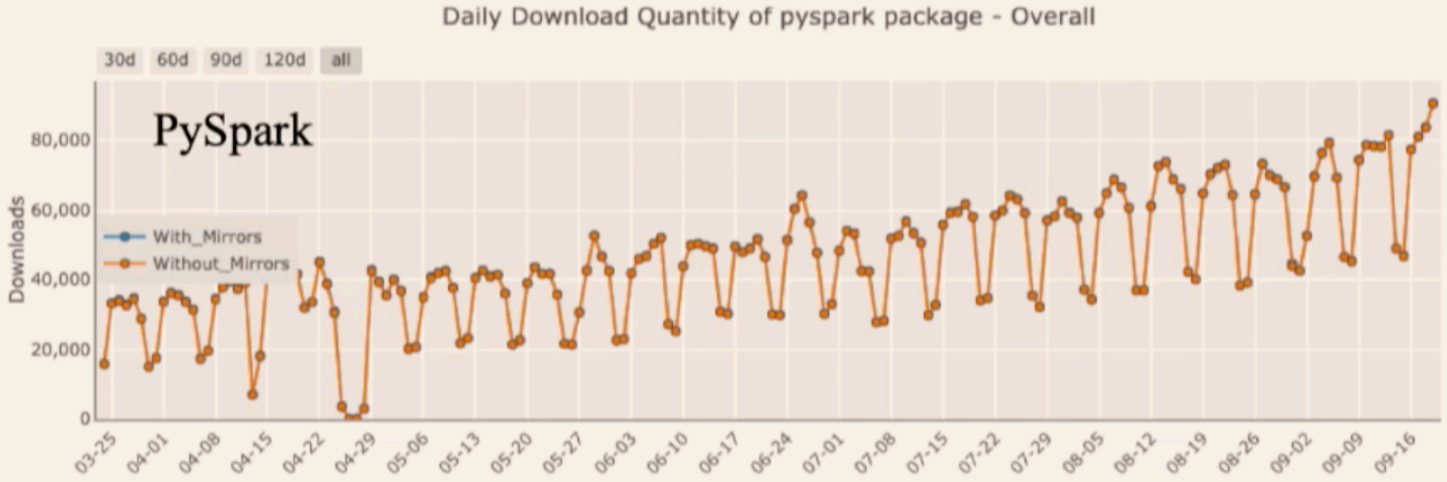


Image source: pypistats.org

koalas

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

	pandas DataFrame	PySpark DataFrame
Column	df['col']	df['col']
Mutability	Mutable	Immutable
Add a column	df['c'] = df['a'] + df['b']	df.withColumn('c', df['a'] + df['b'])
Rename columns	df.columns = ['a','b']	df.select(df['c1'].alias('a'), df['c2'].alias('b'))
Value count	df['col'].value_counts()	df.groupBy(df['col']).count() .orderBy('count', ascending = False)

pandas

```
import pandas as pd
df = pd.read_csv("my_data.csv")
df.columns = ['x', 'y', 'z1']

df['x2'] = df.x * df.x
```

PySpark

```
df = (spark.read
    .option("inferSchema", "true")
    .option("comment", True)
    .csv("my_data.csv"))

df = df.toDF('x', 'y', 'z1')
df = df.withColumn('x2', df.x*df.x)
```

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

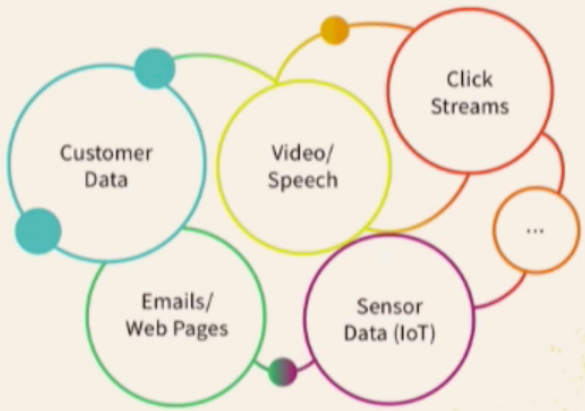
databricks

```
import pandas as pd
df = pd.read_csv("my_data.csv")
df.columns = ['x', 'y', 'z1']

df['x2'] = df.x * df.x
```

```
import databricks.koalas as ks
df = ks.read_csv("my_data.csv")
df.columns = ['x', 'y', 'z1']

df['x2'] = df.x * df.x
```


Customer Data
Video/ Speech
Click Streams
...
Emails/ Web Pages
Sensor Data (IoT)

数据工程

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

A New Standard for Building Data Lakes

Open Format Based on Parquet

With Transactions

Apache Spark™ APIs

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

Instead of parquet …

```
dataframe
.write
.format("parquet")
.save("/data")
```

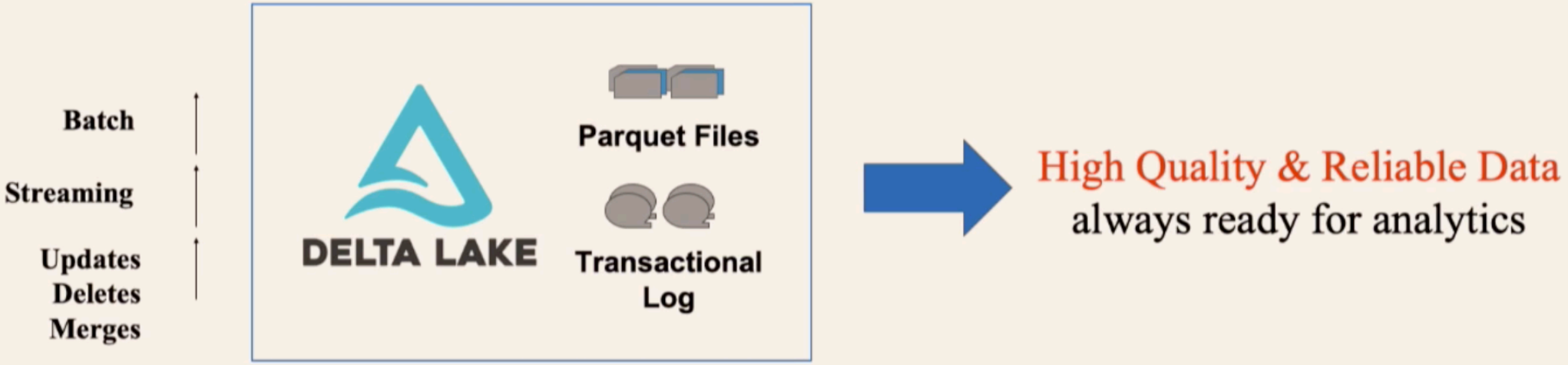
… simply say delta

```
dataframe
.write
.format("delta")
.save("/data")
```

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

Delta Lake ensures data reliability



Key Features

- ACID Transactions
- Schema Enforcement
- Unified Batch & Streaming
- Time Travel/Data Snapshots

COMCAST

databricks

Improved reliability:
Petabyte-scale jobs

10x lower compute:
640 服务器 → 64!

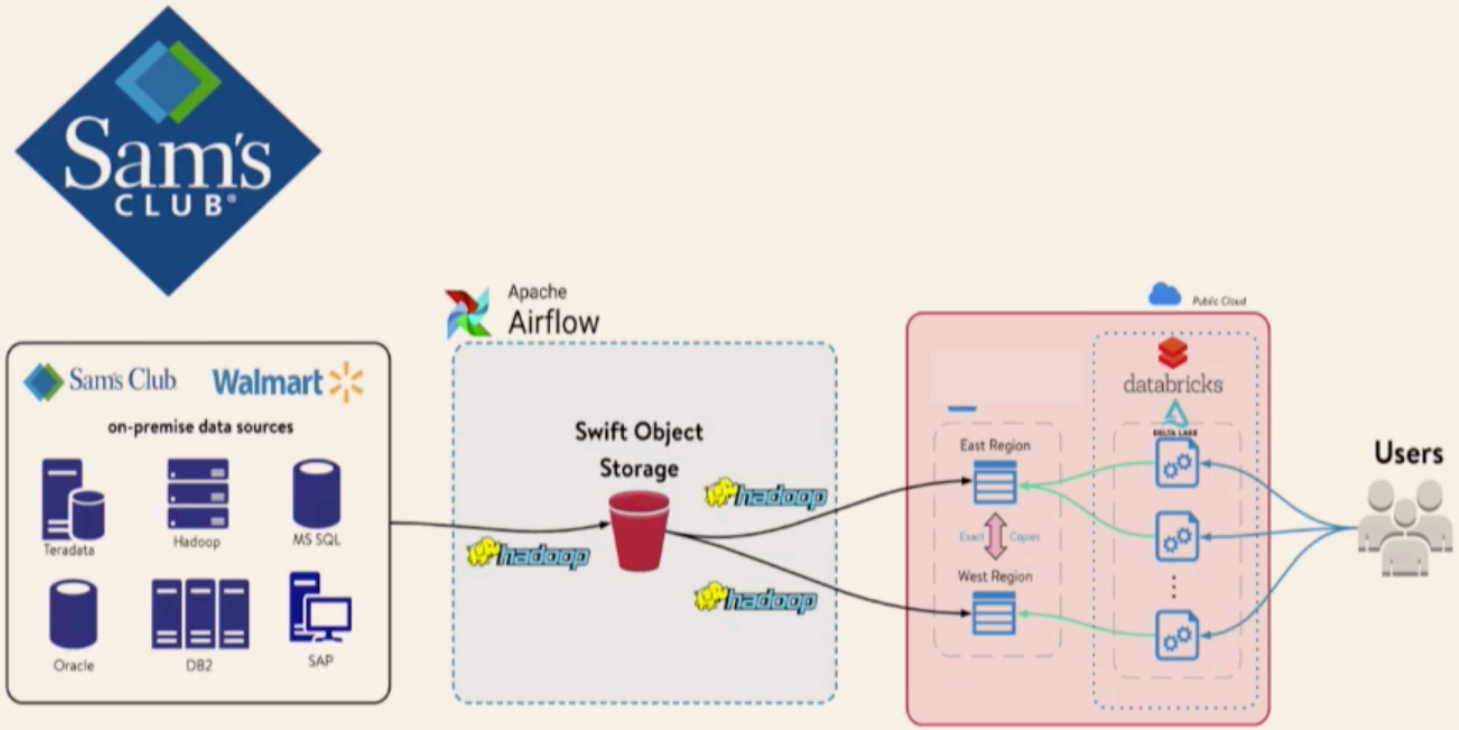
Simpler, faster ETL:
84 jobs → 3 jobs
halved data latency

SESSIONIZATION WITH DELTA LAKE

FASTER QUERIES, RELIABLE PIPELINES, 10X REDUCTION IN COMPUTE!

14

COMCAST



Easier transactional updates:

No downtime or consistency issues!

Simple CDC:

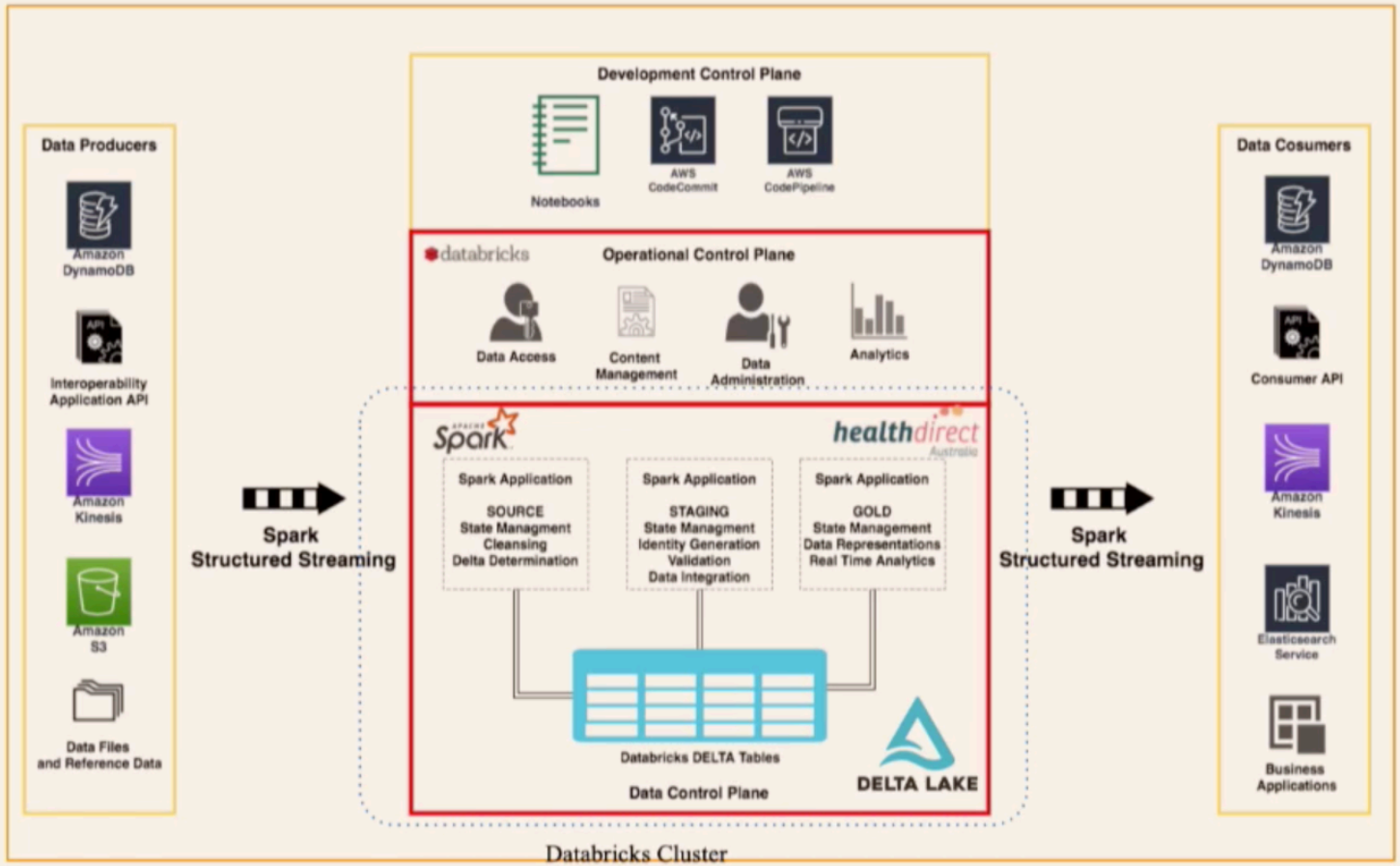
Easy with MERGE

Improved performance: Queries run faster

大于一小时 → 少于六秒

healthdirect Australia

databricks



Data consistency and integrity:

not available before

Increased data quality:

name match accuracy up

80% → 95%

Faster data loads:

一天 → 二十分钟

Alibaba Group 阿里巴巴集团 | APSARA 云栖大会

databricks

Databricks Delta in Numbers!

日均 38+ PB

在客户生产环境
处理数据量

日均 400+ TB

某一大型客户的生产环境
新入库数据量


Alibaba Group
阿里巴巴集团
APSARA
云 栖 大 会

databricks


10
YEARS


APACHE
Spark
TM


Alibaba Group
阿里巴巴集团
APSARA
云 栖 大 会
databricks
DELTA LAKE
koalas
SQL
Java
mlflow
10 YEARS
APACHE
Spark™

Thank You

李潇 (lixiao@databricks.com)

databricks