Building Applications with YugabyteDB and Spark

Wei Wang

Amey Banarse

Who we are

Wei Wang

Principal Pre Sales Architect

Rackspace • Accenture • HP Enterprise

University of Oklahoma

@wwang0825

Amey Banarse

VP of Field Engineering, Yugabyte, Inc.

Pivotal • FINRA • NYSE

University of Pennsylvania (UPenn)

@ameybanarse

about.me/amey

Workshop Agenda

- Overview of Yugabyte Architecture
- Yugabyte's YCQL API
- YugabyteDB Spark Connector
- Hands-on Workshop

yugabyteDB

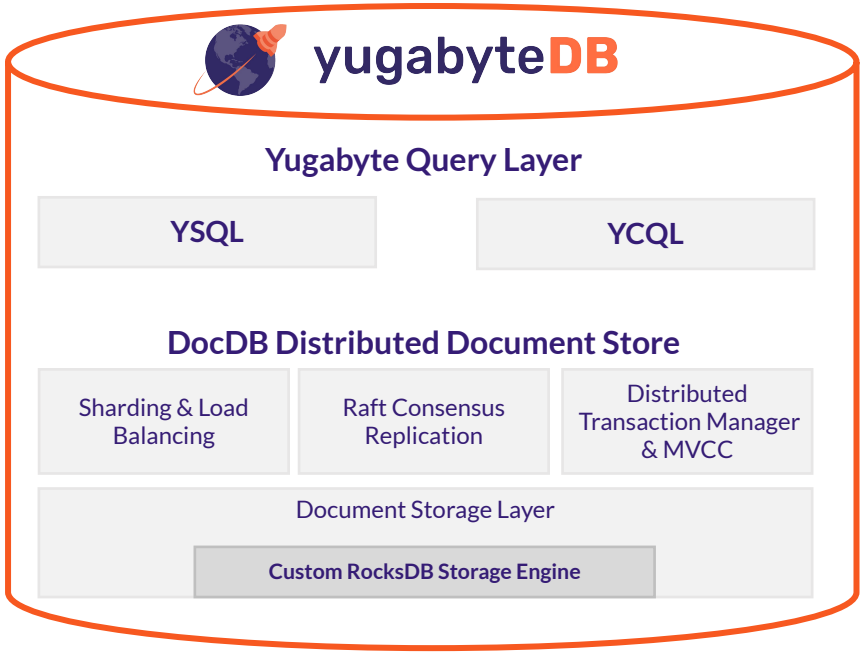
Transactional, distributed SQL database designed for **resilience** and **scale**

100% open source, PostgreSQL compatible, enterprise-grade RDBMS

…..built to run across all your cloud environments

Designed for cloud native microservices.

	PostgreSQL	Google Spanner	YugabyteDB
SQL Ecosystem	✓ Massively adopted	✗ New SQL flavor	✓ **Reuse PostgreSQL**
RDBMS Features	✓ Advanced *Complex*	✗ Basic *cloud-native*	✓ **Advanced** *Complex and cloud-native*
Highly Available	✗	✓	✓
Horizontal Scale	✗	✓	✓
Distributed Txns	✗	✓	✓
Data Replication	Async	Sync	Sync + Async



Yugabyte Query Layer - YCQL

In addition to Cassandra CQL API support, YCQL adds the following enhancements:

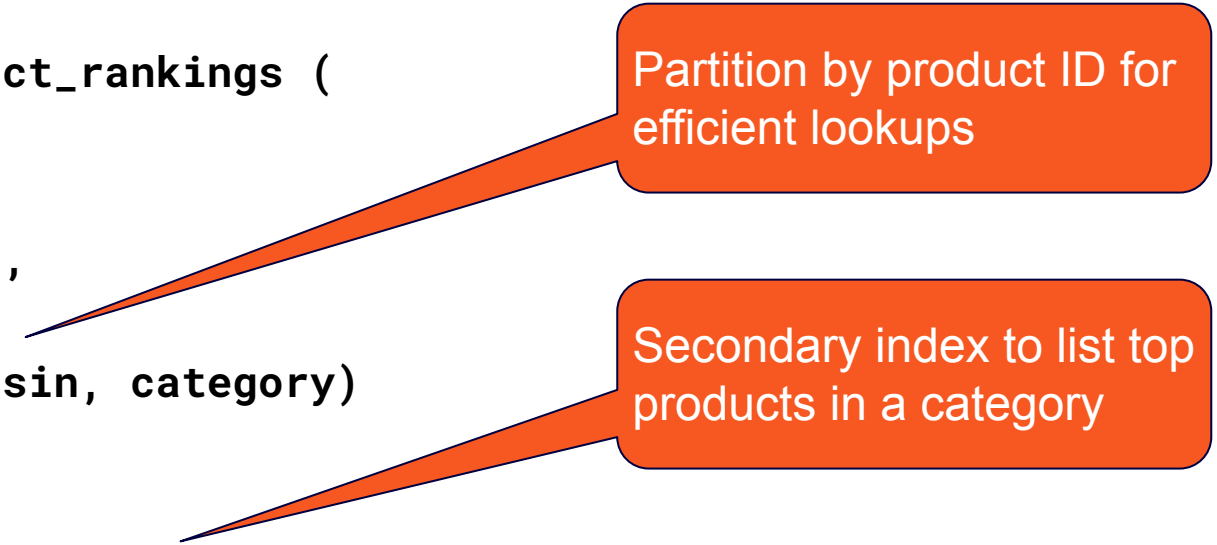
- Strongly-consistent secondary indexes
- Native JSON support
- Geographic location hints to optimize cost and latency
- Distributed ACID transactions
- Geo-distributed
- Supports Cassandra CQL 3.9.x and 4.x

Features	Cassandra	YCQL
Clustering and sharding	✓	✓
Keyspaces and tables	✓	✓
Indexes & unique constraints	✓	✓
Transactions	✓	✓
Strongly consistent secondary indexes	×	✓
Native JSON	×	✓
Geographic location hints	×	✓

Secondary Indexes - YCQL

```
CREATE TABLE product_rankings (
    asin text,
    category text,
    sales_rank int,
    ...
    PRIMARY KEY (asin, category)
);

CREATE INDEX top_products_in_category
    ON product_rankings (category, sales_rank, asin);
```



Secondary Indexes - YCQL

```
SELECT * FROM product_rankings
    WHERE asin = '0684841363';
```

Query by ID for product summary data

```
SELECT * FROM product_rankings
    WHERE category = 'Books'
    LIMIT 10
    OFFSET 20;
```

Query by index to list top products in a category by sales rank (best sellers)

Global Transactions

Multi-Row/Multi-Shard Operations At Scale

```
CREATE TABLE orders (
    order_id text PRIMARY KEY,
    user_id  uuid,
    order_details jsonb,
    ...
) WITH transactions = {'enabled': 'true'};
```

Update inventory and orders tables atomically.

Use JSON type for flexible schema objects.

Simple to enable global transactions on any table

Global Transactions

Multi-Row/Multi-Shard Operations At Scale

```
BEGIN TRANSACTION

  UPDATE product_inventory
      SET quantity = quantity - 2
      WHERE asin = '0684841363';

  INSERT INTO orders (order_id, user_id, order_details, … )
      VALUES ('<order-id>', '<user-id>',
              '{id: "0684841363", quantity: 2, … }', … );

END TRANSACTION;
```

Decrease inventory count for products fulfilled.

Add to orders table for products purchased.

Query and index by JSON attributes if needed!

Native JSON

```
cqlsh> SELECT * FROM store.books;
```

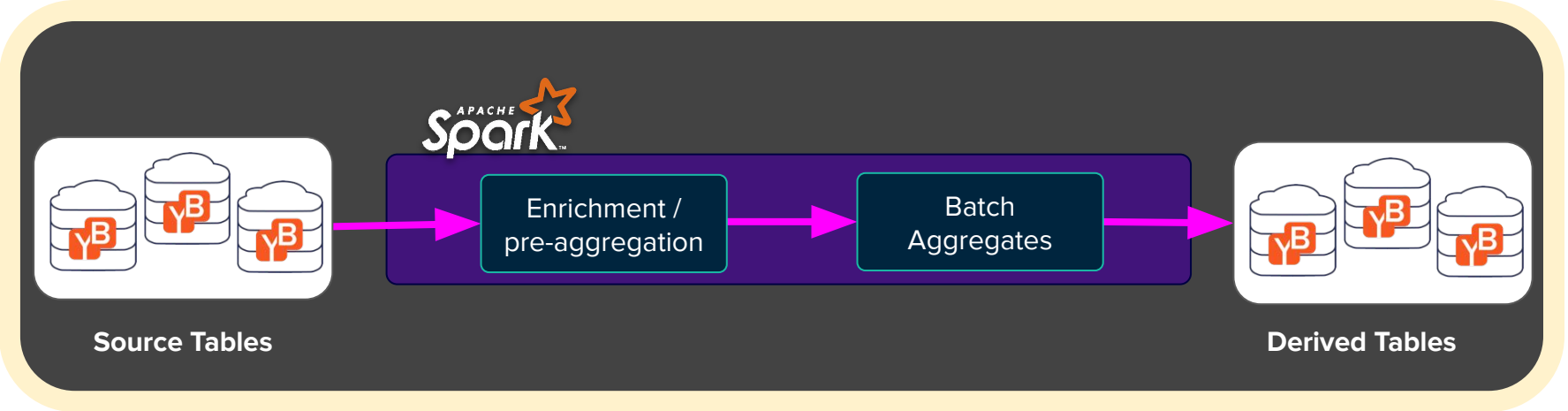
```
 id | details
----+-----------------------------------------------------------------------------------------------------------------------
  5 | {"author":"Stephen Hawking","genre":"science","name":"A Brief History of Time","year":1988}
  1 |                                    {"author":"William Shakespear","name":"Macbeth","year":1623}
  4 |          {"author":"Charles Dickens","genre":"novel","name":"Great Expectations","year":1950}
  2 |                                     {"author":"William Shakespear","name":"Hamlet","year":1603}
  3 |                {"author":"Charles Dickens","genre":"novel","name":"Oliver Twist","year":1838}

(5 rows)
```

YugabyteDB Spark Connector

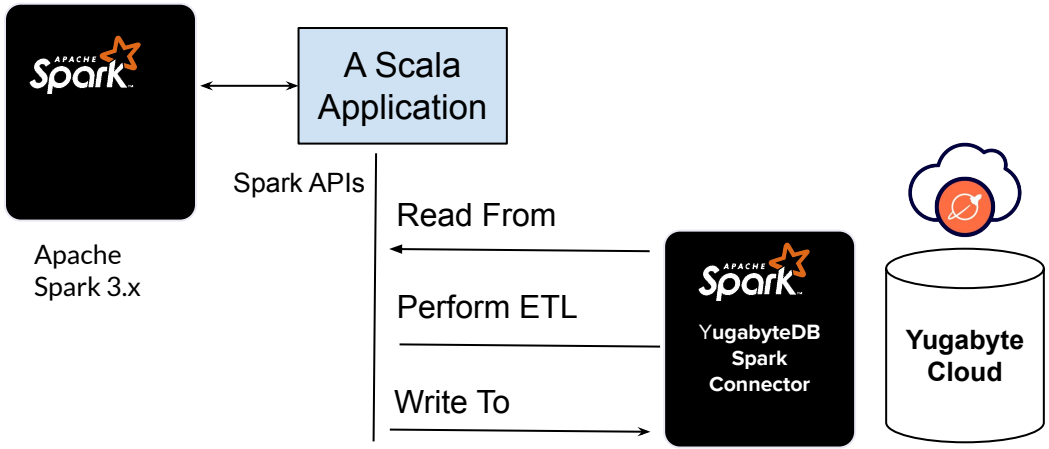
Key features:

- Native JSONB support - ex. Column pruning
- Performance optimizations with predicate pushdowns
- Cluster topology and partition awareness



What we will be building in this workshop?

Integrating Apache Spark with Yugabyte Cloud



YugabyteDB Spark Connector 3.0-yb-8

- Compatible with Spark 3.0 and Scala 2.12
- Compatible with Yugabyte YCQL 3.7+
- Exposes YCQL tables as Spark RDDs and Datasets/DataFrames
- Saves RDDs /DataFrames back to Cassandra by implicit saveToCassandra call
- Allows for execution of arbitrary CQL statements

Yugabyte Cloud

- Data source and target for Spark application
- Namespace: test
- Table: Employees_json

Spark application: Native support of JSON

- Read from and write to Yugabyte cloud
- Perform sample ETL operation: Window function
- Process JSON data type

yugabyteDB

Hands-on Lab

github.com/yugabyte/yugabyte-spring-workshop

yugabyteDB

Thank You

Join us on Slack: yugabyte.com/slack

Star us on Github: github.com/yugabyte/yugabyte-db