



Lesson 17&18

Source Systems

One more step backward, think about where data comes from

Learn About



- **Source Systems**
- **Kafka Architecture**
- **CDC**



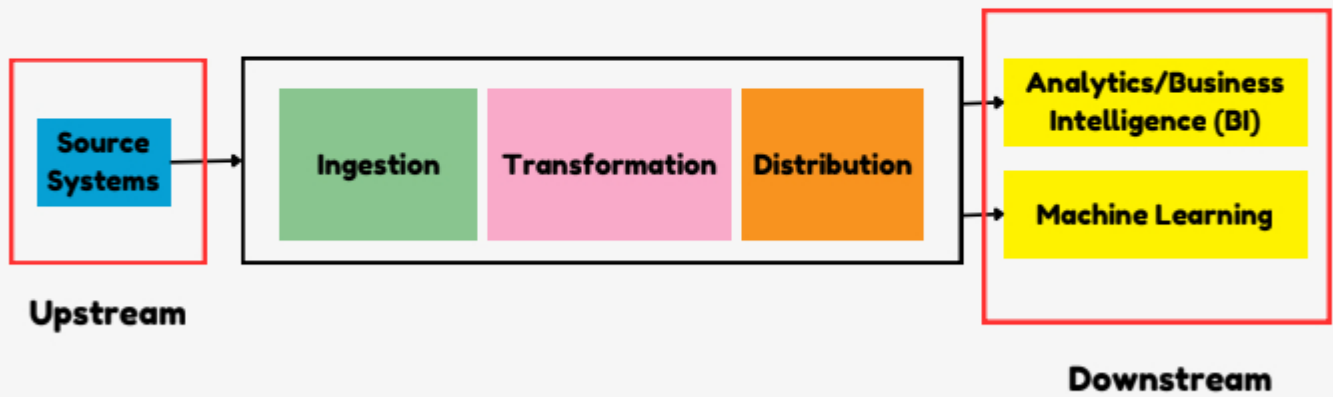


Source Systems




Concept

Source systems generate the data for the rest of the data engineering lifecycle



Files (I)

Types of files:

- **Structured:** Refer to data that can be stored in an SQL database in a table with rows and columns such as Excel, Parquet, and Avro.
 - **Semi-structured:** Data has some organizational properties that make it easier to analyze such as JSON, XML, and CSV. With some further processes, we can store them in a database.
 - **Unstructured:** Data is not organized or does not follow any schema, such as TXT, PNG, PDF, or System Logs.
- 

Files (2)

Row-Based
Storage Layout

String	Int	Date
a	1	2020-01
b	2	2020-02
c	3	2020-03



a 1 2020-01 b 2 2020-02 c 3 2020-03

Store rows sequentially (e.g., **CSV**), optimized for WHERE (**Predicate** process)

Column-Based
Storage Layout

String	Int	Date
a	1	2020-01
b	2	2020-02
c	3	2020-03



a b c 1 2 3 2020-01 2020-02 2020-03

Store columns sequentially (e.g., **ORC**), optimized for SELECT (**Projection** process)

Hybrid-Based
Storage Layout
(row group size = 2)

String	Int	Date
a	1	2020-01
b	2	2020-02
c	3	2020-03



a b 2020-01 2020-02 1 2 c 3 2020-03

Final group only has 1 row

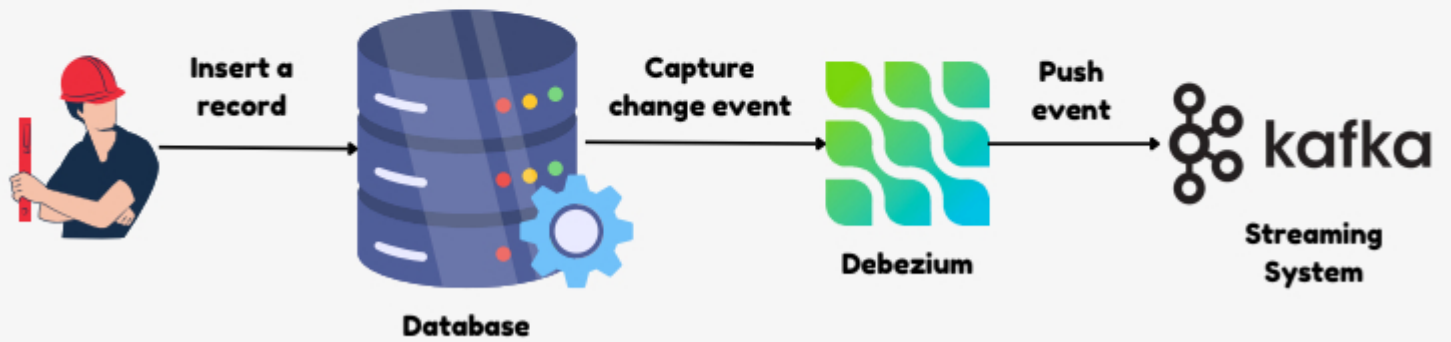
Store chunks sequentially (e.g., **Parquet**), optimized for selecting a subset of rows and columns

Source: <https://towardsdatascience.com/demystifying-the-parquet-file-format-13adb0206705>

CDC

Change Data Capture (CDC): the process of identifying and capturing changes (insert, update and delete) made to data in a database.

According to [Oilk](#)



OLAP vs OLTP Systems

	Online analytical processing (OLAP)	Online transactional processing (OLTP)
Purpose	Analyze large volume of data	Process a large number of real-time transactions
Query types	Complex	Simple
Operations	Mainly select command	Mainly insert, update and delete commands
Response time	Longer, typically seconds, minutes or hours	Shorter, typically milliseconds
Use-cases	Analytics and ML	Online booking and ecommerce

Logs

- **Definition:** Logs capture events occurring in systems, which can be used for downstream tasks such as data analysis and ML.

- **Source sources:**



Docker containers



Servers



IoT devices




Kafka Architecture



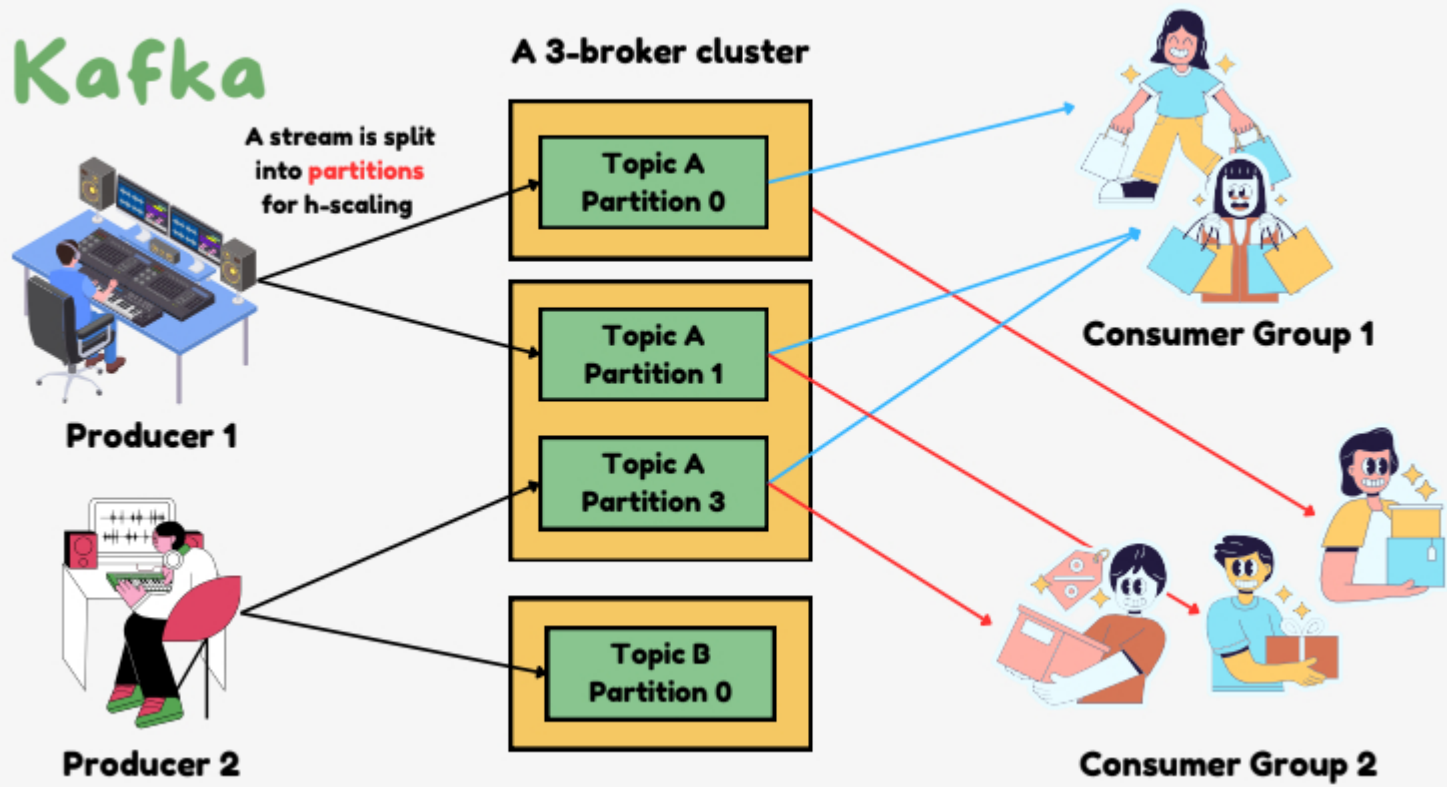
Message broker models

Two basic message distribution patterns

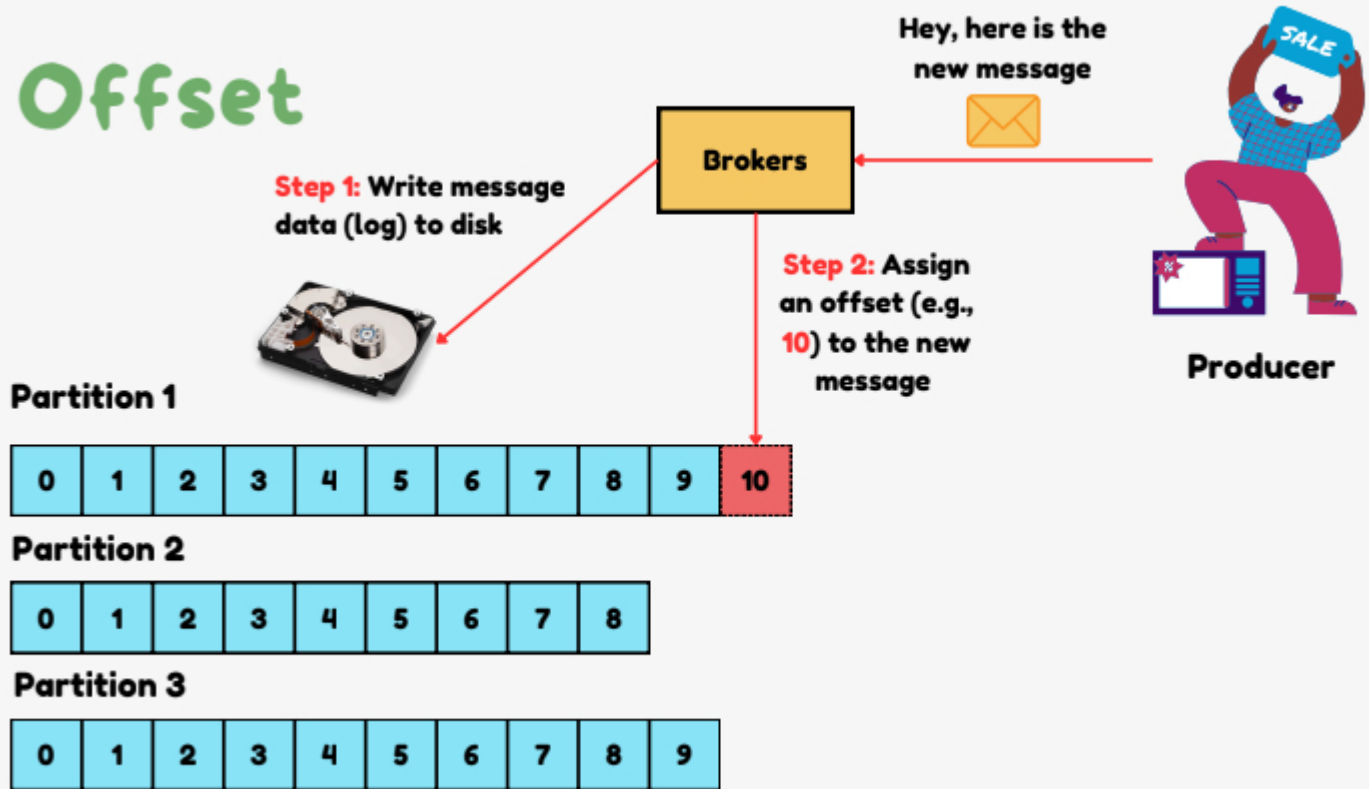
- **Point-to-point (P2P):** Each message in the queue is sent to only one recipient and is consumed only once (**one-to-one** relationship between the sender and the receiver)
 - **Publish/Subscribe (Pub/Sub):** Producer publishes a message on a topic, and multiple consumers can receive this message as long as they subscribe to the topic (**one-to-many** relationship between a sender and receivers)
- 

Kafka

A 3-broker cluster



Offset



Assign message to partitions

Partition 1

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

Partition 2

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

Partition 3

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



Decide partitions
based on parameters
key and **partition**

Hey, here is the
new message



Producer

How to decide?

- **No key, no partition:** random
- **Key-specific:** Use consistent hashing
- **Partition-specific:** Well, this partition
- **Custom partitioner in producer config**

Idle consumer



Producer

Partition 1

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

Partition 2

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

Partition 3

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

Consumer group A

Consumer 1

Consumer 2

Consumer 3

Consumer 4



This will be idle

Rebalance



Producer

Partition 1



Partition 2



Partition 3



Consumer group A

Consumer 1

Consumer 2



Consumer 3

If one or more consumers are removed or added, partitions will be **rebalanced** to consumers.

Multi-topic vs. multi-partition

Should I use one topic or one partition for one entity?

Considerations	Multi-topic	Multi-partition (w/o random)
Different concerns	Yes	No
Different data type	Yes	No
Retention policy	Yes	No



CDC



How Debezium works

There are a variety of methods to capture data change:

- **Audit columns:** Based on the column “last_updated”
- **Table deltas:** Compare with a previous snapshot
- **Trigger-based:** Trigger before or after INSERT, UPDATE, and DELETE commands
- **Log-based:** Trigger by monitoring transaction logs
=> This is how Debezium PostgreSQL works

References

- [Kafka the Definitive Guide](#)
- [Understanding Message Brokers](#)
- <https://www.ibm.com/topics/message-brokers>
- <https://gpcoder.com/6930-su-dung-topic-exchange-publish-subscribe-trong-rabbitmq/>
- <https://viblo.asia/p/tim-hieu-ve-rabbitmq-OeVKB8bMlkWz>
- <https://aws.amazon.com/compare/the-difference-between-olap-and-oltp/>
- <https://www.stitchdata.com/resources/oltp-vs-olap/>
- <https://www.striim.com/blog/change-data-capture-cdc-what-it-is-and-how-it-works/>
- https://access.redhat.com/documentation/ko-kr/red_hat_integration/2020.q1/html/debezium_user_guide/debezium-connector-for-postgresql

Thank You!



Created by
Quan Dang



