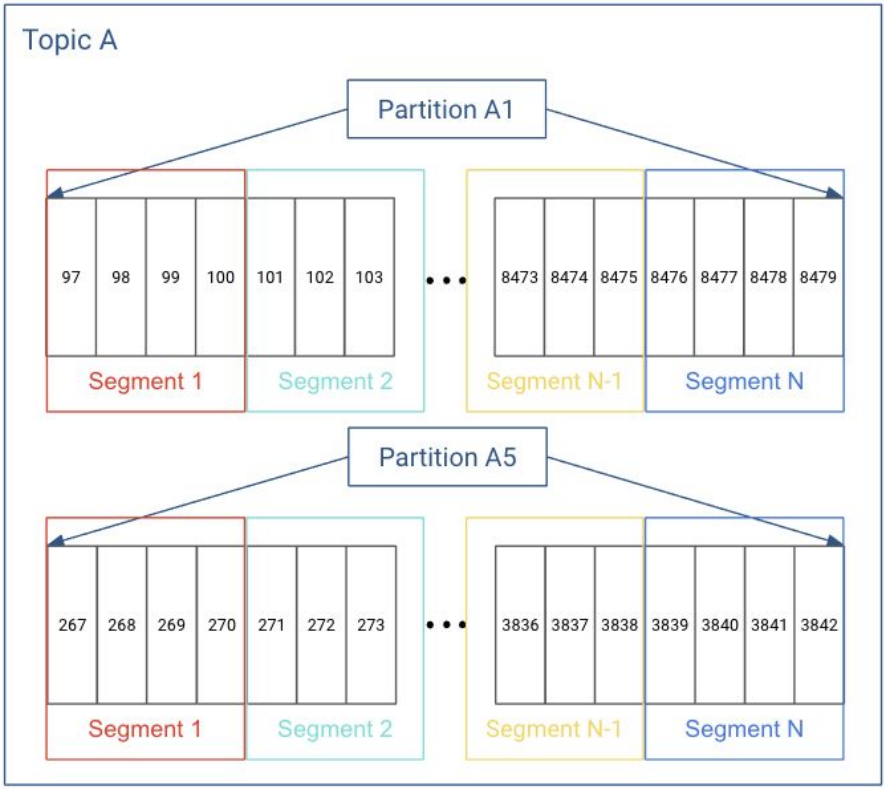
Internal of Kafka Storage

How Partitions are stored on disk



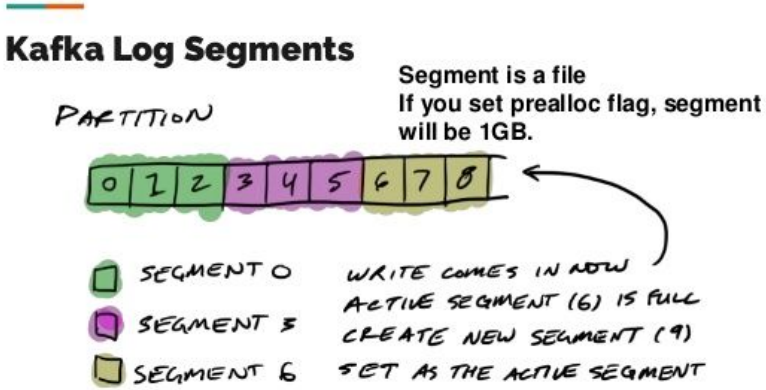
In actuality, each partition does not keep all the records sequentially in a single file. Instead, it breaks each log into log segments. Log segments can be defined using a size limit (for example, 1 GB), as a time limit (for example, 1 day), or both.

Each message has its **value, offset, timestamp, key, message size, compression codec, checksum, and version of the message format.**

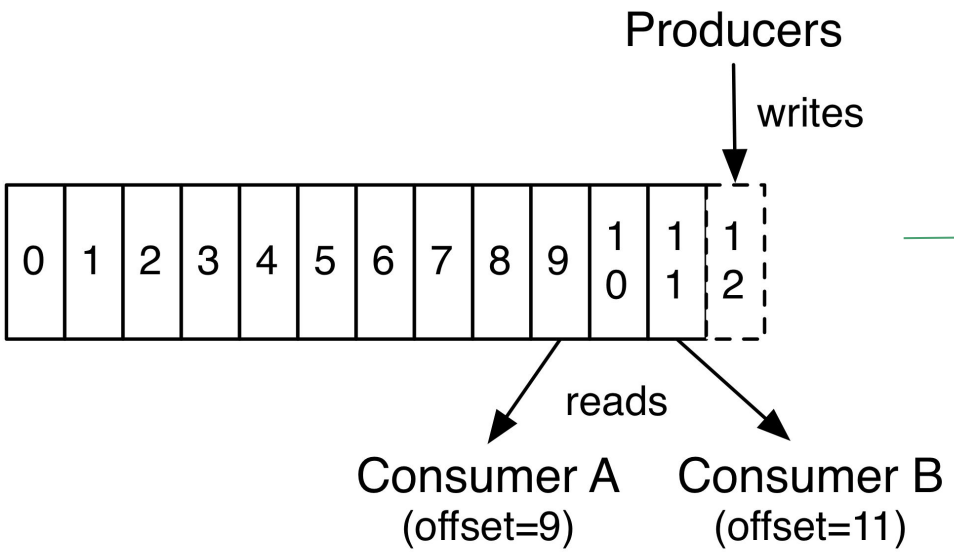
How Partitions are stored on disk

Segments: each partition is split/saved on disk in one or more segments/logs (default size 1GB). There is only one active segment at one time, when the segment reaches a max size or a certain period of time the segment is closed and a new one becomes the active one. **Each segment has an index.**

On disk, a partition is a directory and each segment is an index file and a log file.



Log representation on disk



On disk detailed log view

Offset index

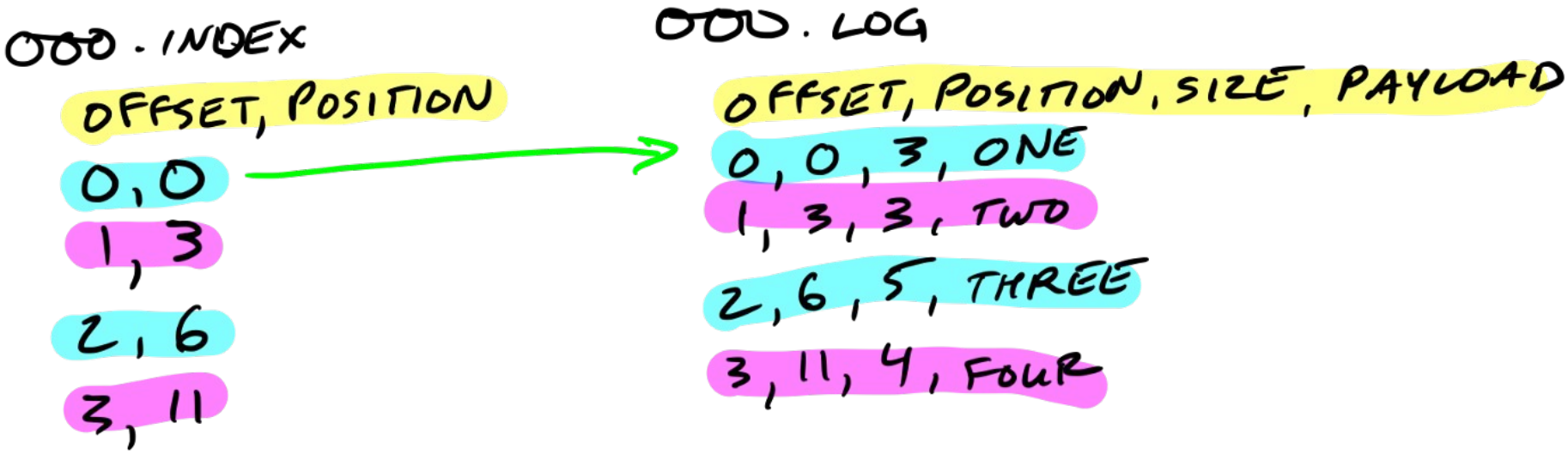
Offset 0 - 10000 (actual messages)

Timestamp index

Kafka 0.10

Each partition is split on disk in segments (1GB or 1 week of data, the smallest) and if the segment limit is reached, we close the file and start a new one. The segment currently writing into = active segment.

Segments



The segment index maps offsets to their message's position in the segment log.

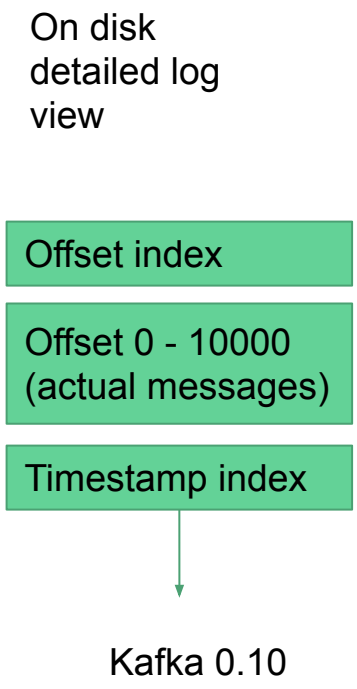
Log representation on disk

Kafka allows consumers to start fetching messages from any available offset. This means that if a consumer asks for messages starting at offset 100, the broker must be able to quickly locate the message for offset 100 (which can be in any of the segments for the partition) and start reading the messages from that offset on.

Kafka maintains an **index for each partition**. The **index maps offsets to segment files and positions** within the file. Indexes are also broken into segments, so we can delete old index entries when the messages are purged.

The **timestamp is given either by the producer when the message was sent or by the broker when the message arrived depending on configuration**.

Timestamp index (Kafka 0.10.1.0) - searchable index for each topic based off of message timestamps, which were added in 0.10.0.0. This allows for finer-grained log retention than was possible previously using only the timestamps from the log segments. It also enables consumer support for offset lookup by timestamp, which allows you to seek to a position in the topic based on a certain time.

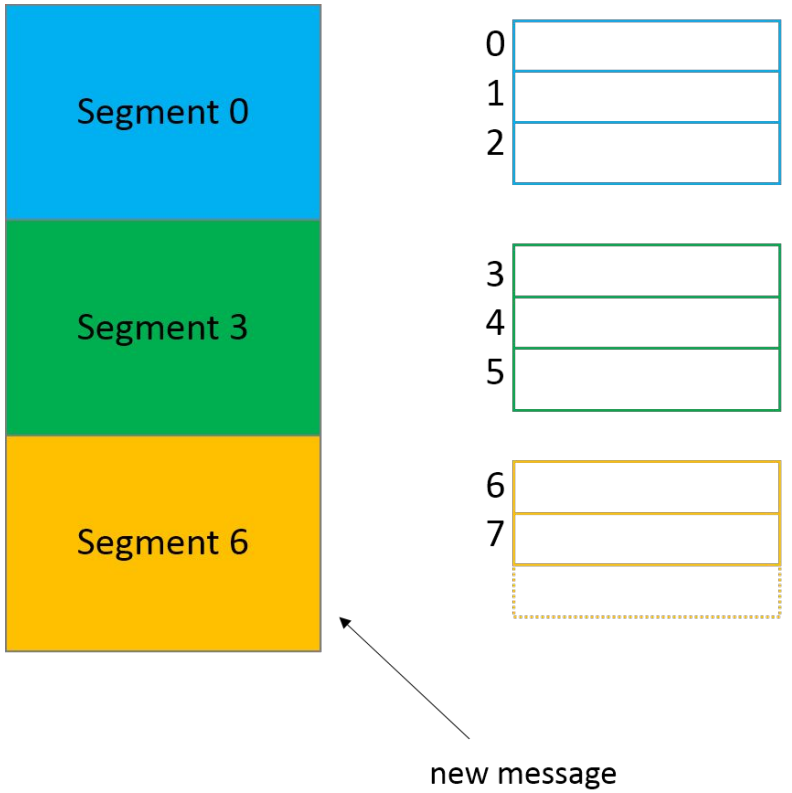


Segments are rolled when

- Size (def 1GB)
- Time
- Offsetindex is full
- Timeindex is full

Kafka writing on disk

Each segment file is created with the offset of the first message as its file name. So, In the above picture, segment 0 has messages from offset 0 to offset 2, segment 3 has messages from offset 3 to 5 and so on. Segment 6 which is the last segment is the active segment.



Exercise

Kafka Storage of data exercise