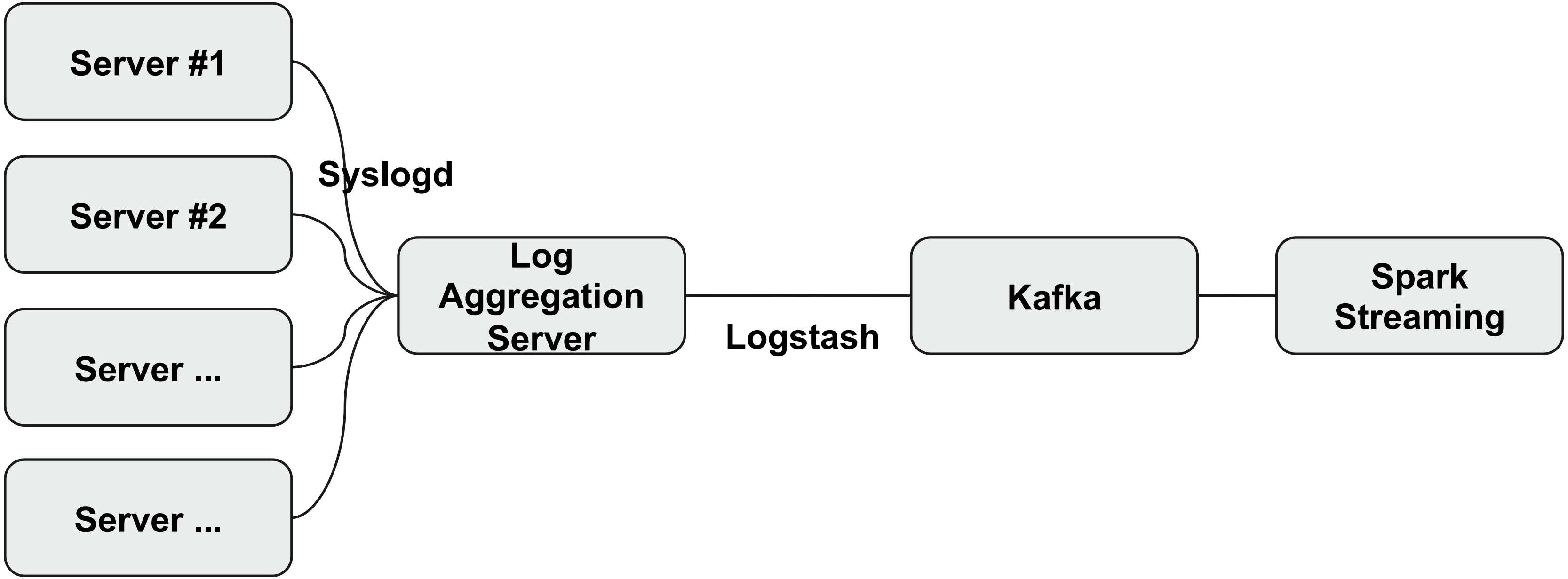
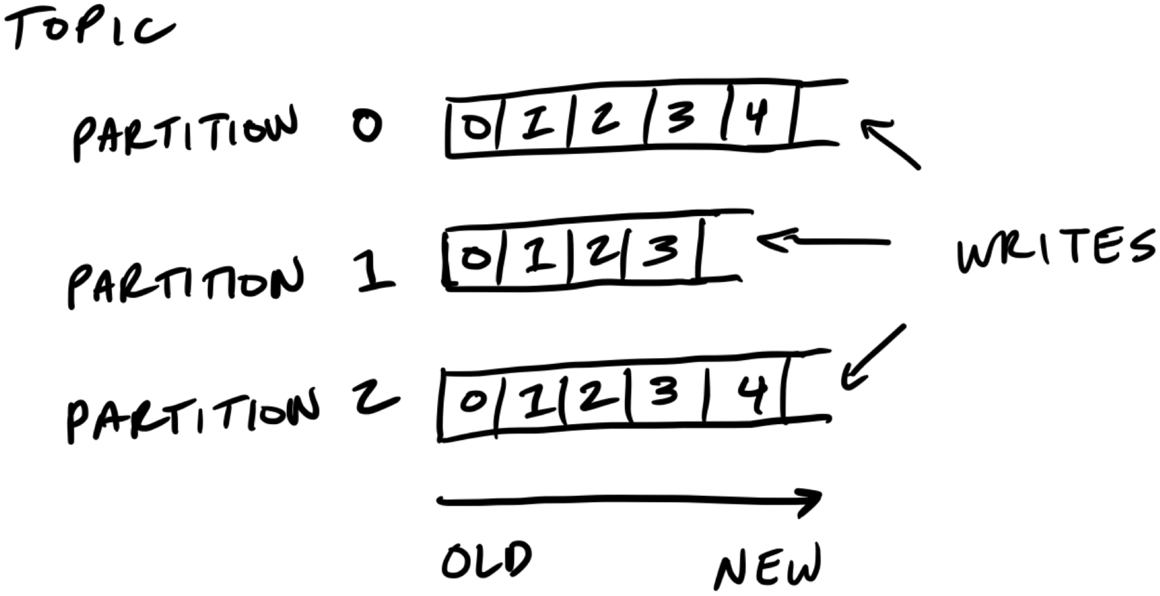
별 쓸모 없는 이야기
Spark, 그리고
Kafka timestamp offset

강대명(charsyam@naver.com)

Common Way: Kafka and Spark Streaming

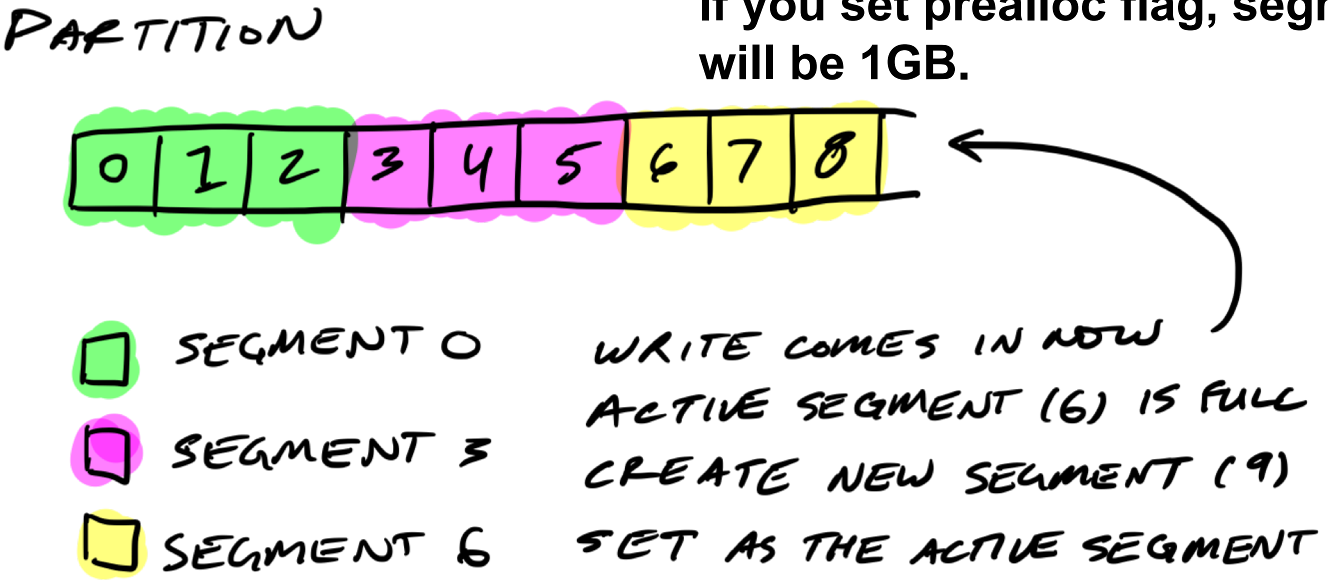


Kafka Log


TOPIC
PARTITION 0
0 1 2 3 4
PARTITION 1
0 1 2 3
PARTITION 2
0 1 2 3 4
WRITES
OLD
NEW

Kafka Log Segments

Segment is a file
If you set prealloc flag, segment will be 1GB.



WRITE COMES IN NOW
ACTIVE SEGMENT (6) IS FULL
CREATE NEW SEGMENT (9)
SET AS THE ACTIVE SEGMENT

What is timestamp Index?

Fetching Kafka Logs by Timestamp.(From Kafka-0.10.0.2)

You can query from specific timestamp range of message.(very useful)

ex) From 2018-09-08 00:00:00 To 2018-09-08-23:59:59

Related KIPs

KIP-32 - Add timestamps to Kafka message : 0.10.0.0

KIP-33 - Add a time based log index : 0.10.1.0

Kafka Log Files

00000000000155593652.log	Kafka Message Data File
00000000000155593652.index	OffsetIndex File
00000000000155593652.timeindex	TimeIndex File

There are many files for 1 segment.
.txnindex, .snapshot, .deleted, .cleaned, .swap, etc

Kafka Index Files

OffsetIndex.scala
TimeIndex.scala

There are different Index Files

When Kafka append Logs.

```
@nonthreadsafe
def append(largestOffset: Long,
           largestTimestamp: Long,
           shallowOffsetOfMaxTimestamp: Long,
           records: MemoryRecords): Unit = {
    ……
    val appendedBytes = log.append(records)
    ……
    offsetIndex.append(largestOffset, physicalPosition)
    timeIndex.maybeAppend(maxTimestampSoFar, offsetOfMaxTimestamp)
    ……
}
```

When Kafka writes Logs #1

- Using MMAP(so OS Page Cache is very important for performance)
- Offset is stored as relative offset
    - Current Offset - Base Offset
- Offset will be returned Absolute Offset

When Kafka writes Logs #2

OffsetIndex

Append

Int(4 bytes)	Int(4 bytes)
Offset	Position

When Kafka writes Logs #3

TimeIndex

Append

Long(8 bytes)	Int(4 bytes)
Timestamp	Position

How to fetch by timestamp #1

```
def fetchOffsetsByTimestamp(targetTimestamp: Long): Option[TimestampOffset] = {
     ......
     val targetSeg = {
       val earlierSegs = segmentsCopy.takeWhile(_.largestTimestamp < targetTimestamp)
       if (earlierSegs.length < segmentsCopy.length)
         Some(segmentsCopy(earlierSegs.length))
       else
         None
     }
     targetSeg.flatMap(_.findOffsetByTimestamp(targetTimestamp, logStartOffset))
   }
 }
```

How to fetch by timestamp #2

```scala
def findOffsetByTimestamp(timestamp: Long, startingOffset: Long = baseOffset): Option[TimestampOffset] = {
  // Get the index entry with a timestamp less than or equal to the target timestamp
  val timestampOffset = timeIndex.lookup(timestamp)
  val position = offsetIndex.lookup(math.max(timestampOffset.offset, startingOffset)).position
```

Using BinarySearch For Search

```scala
  // Search the timestamp
  Option(log.searchForTimestamp(timestamp, position, startingOffset)).map { timestampAndOffset =>
    TimestampOffset(timestampAndOffset.timestamp, timestampAndOffset.offset)
  }
}
```

Simpe Question!!!

- Why offset index is needed?
- How to use binary search in Index File?

How to query by timestamp in spark

Convert timestamp to OffsetRange

Just Create KafkaRDD with OffsetRange

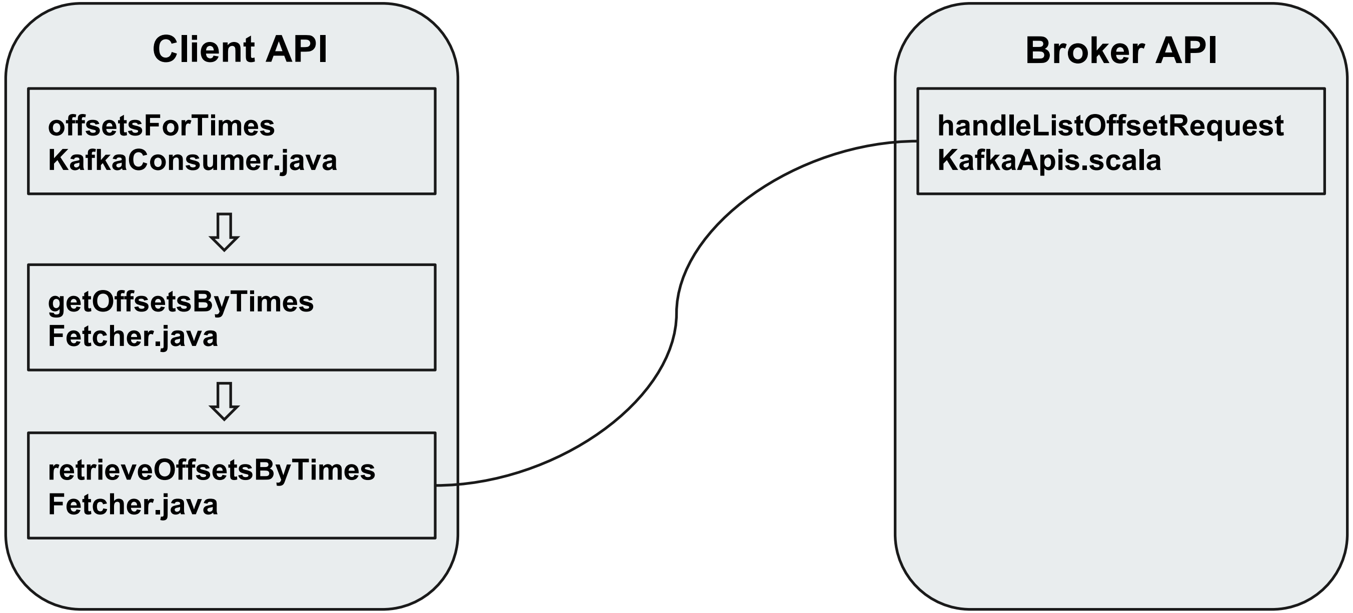
Create KafkaRDD with OffsetRange

```
KafkaUtils.createRDD[K, V](spark.sparkContext, kafkaParamsMap, offsetRanges, PreferConsistent)
```

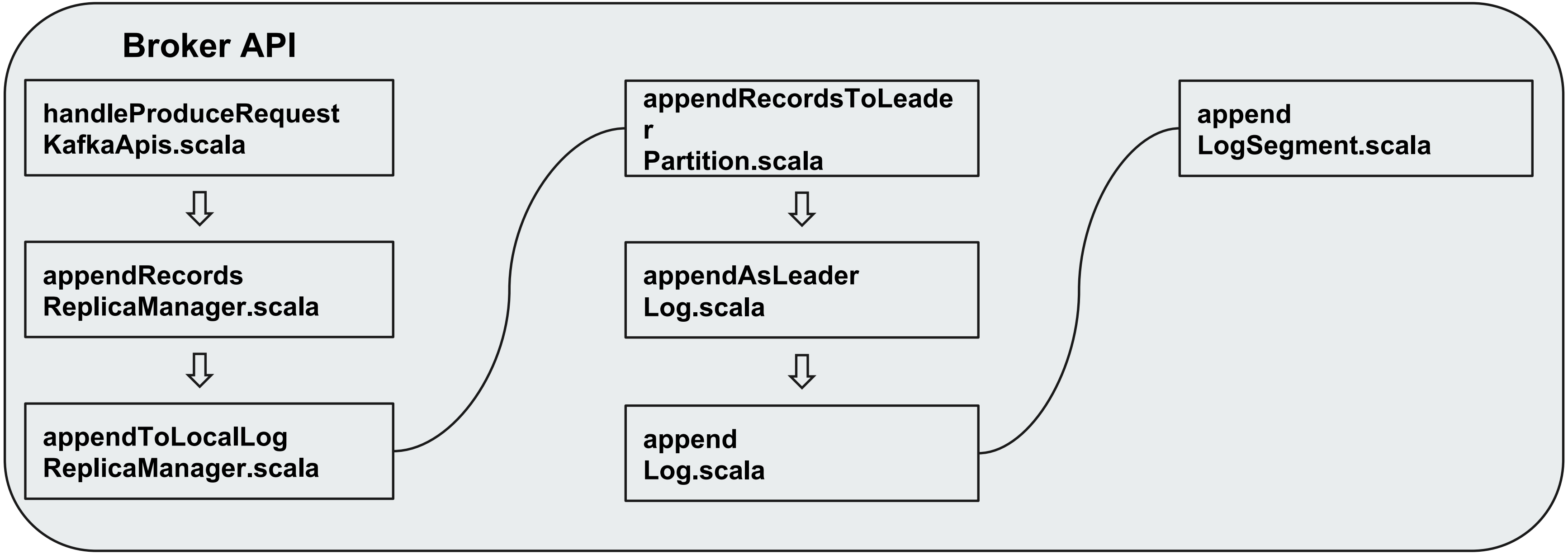
Convert timestamp to OffsetRange

```
val consumer = createKafkaConsumer(props)
val startOffset = consumer.offsetsForTimes(topicMap)
val endOffset = consumer.offsetsForTimes(topicMap)
```

KafkaConsumer.offsetsForTimes



Log Append Flows



Broker API

handleProduceRequest
KafkaApis.scala

appendRecords
ReplicaManager.scala

appendToLocalLog
ReplicaManager.scala

appendRecordsToLeade
r
Partition.scala

appendAsLeader
Log.scala

append
Log.scala

append
LogSegment.scala

When Segmnet is rolled?

```
def shouldRoll(messagesSize: Int, maxTimestampInMessages: Long, maxOffsetInMessages: Long, now: Long): Boolean = {
  val reachedRollMs = timeWaitedForRoll(now, maxTimestampInMessages) > maxSegmentMs - rollJitterMs
  size > maxSegmentBytes - messagesSize ||
    (size > 0 && reachedRollMs) ||
    offsetIndex.isFull || timeIndex.isFull || !canConvertToRelativeOffset(maxOffsetInMessages)
}
```

When Segmnet is rolled?

```
def shouldRoll(messagesSize: Int, maxTimestampInMessages: Long, maxOffsetInMessages: Long, now: Long): Boolean = {
  val reachedRollMs = timeWaitedForRoll(now, maxTimestampInMessages) > maxSegmentMs - rollJitterMs
  size > maxSegmentBytes - messagesSize ||
    (size > 0 && reachedRollMs) ||
    offsetIndex.isFull || timeIndex.isFull || !canConvertToRelativeOffset(maxOffsetInMessages)
}
```

1] size > maxSegmentBytes - messageSize
2] size > 0 && reachedRollMs
3] offsetIndex.isFull
4] timeIndex.isFull
5] canCovertToRelativeOffset is false

One Cent for Using Kafka Timestamp offset

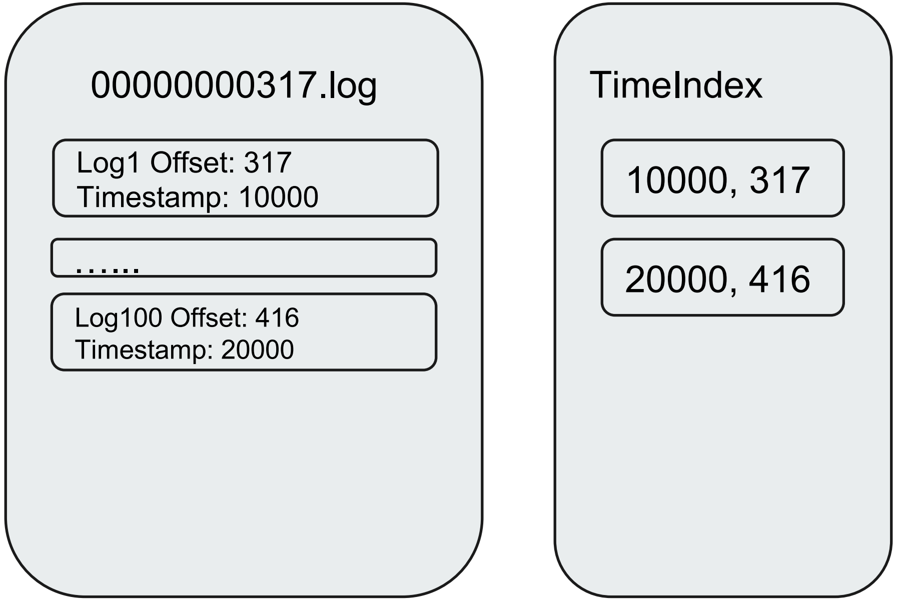
- As a default, timestamp is set as sending time by client.
- So it is not a time that log is created.
  - You should specify to use timestamp as created time of log.

Thank you!

Quiz

- If timestamp is older than last timeIndex
    - How Kafka handles this?

Original



00000000317.log

Log1 Offset: 317
Timestamp: 10000

……

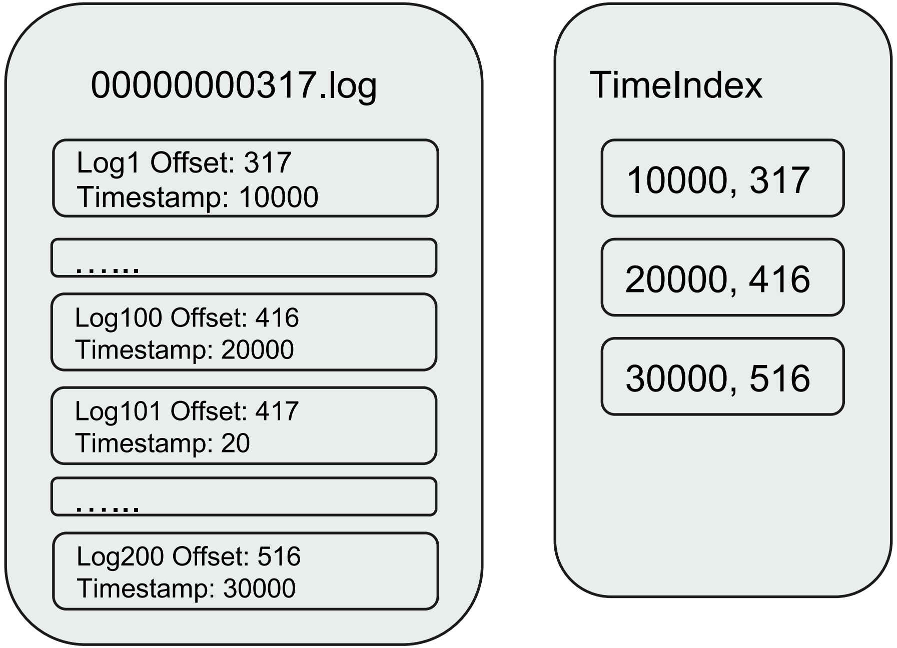
Log100 Offset: 416
Timestamp: 20000

TimeIndex

10000, 317

20000, 416

Append Log with old timestamp



00000000317.log

Log1 Offset: 317
Timestamp: 10000

……

Log100 Offset: 416
Timestamp: 20000

Log101 Offset: 417
Timestamp: 20

……

Log200 Offset: 516
Timestamp: 30000

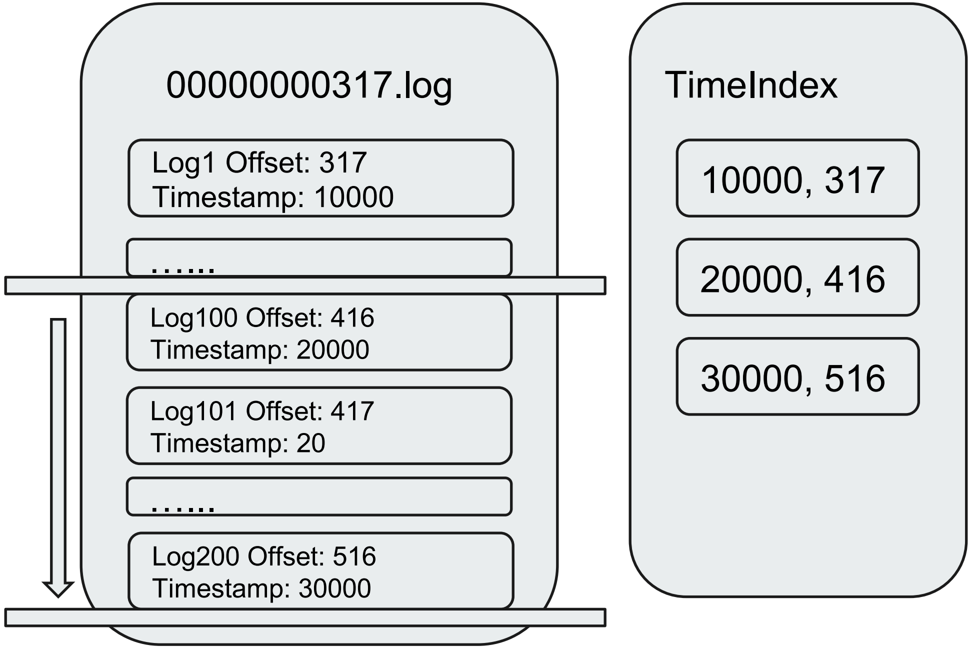
TimeIndex

10000, 317

20000, 416

30000, 516

If you fetch, from timestamp 20000 and you will get Log101 together


00000000317.log
Log1 Offset: 317
Timestamp: 10000
......
Log100 Offset: 416
Timestamp: 20000
Log101 Offset: 417
Timestamp: 20
......
Log200 Offset: 516
Timestamp: 30000
TimeIndex
10000, 317
20000, 416
30000, 516