KIP-228 Negative record timestamp support

Status

Current state: "Voting in progress"

Discussion thread: on mail-archives.apache.org

JIRA: KAFKA-6048

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Kafka does not support negative record timestamps, and this prevents the storage of historical data in Kafka. In general, negative timestamps are supported by UNIX system timestamps:

From https://en.wikipedia.org/wiki/Unix_time

> *The Unix time number is zero at the Unix epoch, and increases by exactly 86,400 per day since the epoch. Thus 2004-09-16T00:00:00Z, 12,677 days after the epoch, is represented by the Unix time number 12,677 × 86,400 = 1095292800. This can be extended backwards from the epoch too, using negative numbers; thus 1957-10-04T00:00:00Z, 4,472 days before the epoch, is represented by the Unix time number 4,472 × 86,400 = 386380800.*

Writing data generated nowadays wouldn't result in the negative timestamp. This is useful if clients are setting record timestamp manually in order to do a lookup by that timestamp later.

Public Interfaces

- org.apache.kafka.common.record
- org.apache.kafka.clients.producer
- org.apache.kafka.streams.processor

Proposed Changes

Allow negative timestamps.

1 value for timestamps stays a special value in Kafka as "no timestamp", which means we do not have a valid way to express Wednesday, December 31, 1969 11:59:59.999 PM UTC.

If you do need to express Wednesday, December 31, 1969 11:59:59.999. PM UTC consider shifting it by one millisecond.

In general, there should be is no reason to fail on broker/client side working with negative timestamps.

Compatibility, Deprecation, and Migration Plan

What impact (if any) will there be on existing users?

No impact on current users, they should update their infrastructure in that order: Broker, Consumers, Producers.

If we are changing behavior how will we phase out the older behavior?

We are not changing the behavior in the way we would need to phase out the older behavior.

If we need special migration tools, describe them here.

No need.

When will we remove the existing behavior?

No need.

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.

Message flag "hasTimestamp"

1. add a special boolean flag to message record "`hasTimestamp`",
2. write a migration tool that adds this flag to message with the negative timestamp to legacy messages,
3. make sure clients know about that field and check them

Timestamp delta

In case it's known that you would need no more than 200 years back before Unix epoch, the possible solution is:

- producers should add that "delta" to an original timestamp,
- all consumers who do lookups by timestamp should know about that "delta",
- keep 1 semantics as it is now.

It may seem the easiest thing to do, nevertheless changing 1 semantics is a cleaner solution.

Topic property "may have a valid negative timestamps"

Add a topic property that specifies that topic may have a valid negative timestamps.

First, we need to remove all checks for negative timestamps across the code:

- client should be able to publish record with a negative timestamp (to the topics that support that),
- broker should accept and serve that record,
- streams should not drop a record with a negative timestamp.

Changes from producer perspective

	Old Broker Behaviour	**New Broker Behaviour**
Old producer sends NO_TIMESTAMP (1L) `isTimestampExtended` = 0	✅ Records this as NO_TIMESTAMP (1L)	✅ Records this as NO_TIMESTAMP (Long. MIN_VALUE)
New producer sends NO_TIMESTAMP (Long. MIN_VALUE) `isTimestampExtended` = 1	❌ Error or Records this as NO_TIMESTAMP (1L)	✅ Records this as NO_TIMESTAMP (Long. MIN_VALUE)
New producer sends 1L ms `isTimestampExtended` = 1	❌ Error or Records this as NO_TIMESTAMP (1L)	✅ Records this as 1L ms

So broker should be updated first, before producers.

Changes from consumer perspective

	Old Consumer	**New Consumer**
Record has TS 1L `isTimestampExtended` = 0	✅ Interpret as NO_TIMESTAMP (1L)	✅ Interpret as NO_TIMESTAMP (Long.MIN_VALUE)
Record has TS 1L `isTimestampExtended` = 1	❌ Error or NO_TIMESTAMP	✅ Interpret as 1L ms

Record has TS Long.MIN_VALUE `isTimestampExtended` = 1	✗ Error or NO_TIMESTAMP	✓ Interpret as NO_TIMESTAMP (Long.MIN_VALUE)

So only new consumers will read records correctly.

Changes in binary message format

Current binary format:

```
XXXX XXXX = 8 bits

 1. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - baseOffset
    XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 2. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - batchLength
 3. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - partitionLeaderEpoch
 4. XXXX XXXX                               - magic (current magic value is 2)
 5. XXXX XXXX XXXX XXXX                     - attributes

    Compression
    000 - no comporession
    001 - gzip
    010 - snappy
    011 - lz4

       Timestamp
       0 - create time
       1 - log append time

         X - isTransactional
          X - isControlBatch
            XX XXXX XXXX - unused

 6. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - lastOffsetDelta
 7. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - firstTimestamp
    XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 8. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - maxTimestamp
    XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 9. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - producerId
    XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
10. XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX - producerEpoch
11. ...                                     - baseSequence
12. ...                                     - records
```

Proposed change: let's use one of the reserved bits to indicate that timestamp can be negative.

```
 5. XXXX XXXX XXXX XXXX - attributes

    Compression
    000 - no comporession
    001 - gzip
    010 - snappy
    011 - lz4

       Timestamp
       0 - create time
       1 - log append time

         X - isTransactional
          X - isControlBatch
           X - isTimestampExtended
            X XXXX XXXX - unused
```

That `isTimestampExtended` bit should be `1` for all new records.

Broker should convert old `NO_TIMESTAMP=1L` to new `NO_TIMESTAMP_EXTENDED=Long.MIN_VALUE`.