**Content:**

* **Kafka Producer config**
  + **compression.type**
  + **batch.size**
  + **linger.ms**
  + **buffer.memory**
  + **max.request.size**

1. **compression.type:**

* Used to compress the messages.
* The compression type for all data generated by the producer.
* **Default** value is none. i.e. **no compression**.
* Specify the final compression type for a given topic.
* Valid values are **none, 'gzip', 'snappy', 'lz4', 'zstd'.**
* The compression is of full batch of data so the efficiency of batching will also impact the compression ratio.**(More batching means better compression.)**
* It additionally accepts **'uncompressed'** which is equivalent to **no compression**;
* The parameters are as given below:
* **Type: string**
* **Default: producer**
* **Valid Values: [uncompressed, zstd, lz4, snappy, gzip, producer]**
* **Importance: high**
* **Update Mode: cluster-wide**

1. **batch.size**

* The producer will attempt to **batch records together** into fewer requests whenever multiple records are sent to the **same partition**.
* This helps performance to both client and the server.
* The configuration controls the default batch size in **bytes**.
* No attempt will be made to batch records larger than this configured size.

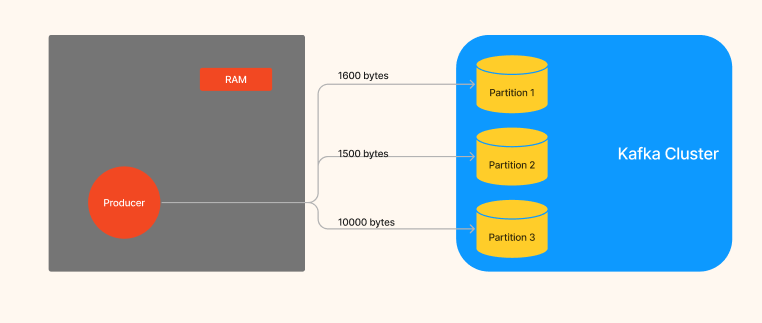


Figure 1 Batch size

* Request sent to brokers will contain multiple batches, one for each partition with data available to be sent. Means if we have 3 partitions then for each partition there will be different batch.
* A small batch size will make batching less common and may reduce throughput.
* A batch size of zero will disable batching entirely.
* A very **large batch size** may use memory a bit **wastefully**, as we will always allocate a buffer of the specified batch size in anticipation of additional records.
* The parameters are as given below:
  + **Type: int**
  + **Default: 16384**
  + **Valid Values: [0,...]**
  + **Importance: medium**

1. **batch.size**

* Every producer has memory allocated in RAM, which is called buffer memory.
* The total bytes of memory the producer can use to buffer records waiting to be sent to the server.
* If messages sent faster than it could be deliver to the server, the producer blocks for max.block.ms after which it can throw an exception.
* This memory corresponds to the memory which is used by the producer.
* But producer used some other memory for compression, and it is not hard bound.
* The parameters are as given below:
  + **Type: long**
  + **Default: 33554432**
  + **Valid Values: [0,...]**
  + **Importance: high**

1. **Linger.ms**

* Producer waits until defined linger.ms time to group the messages into single batch.
* Rather than sending messages immediately, producer waits until given delay i.e. linger.ms
* Producer batches the records until batch.size is not getting completed or linger.ms time is not elapsed.
* The parameters are as given below:
  + **Type: long**
  + **Default: 0**
  + **Valid Values: [0,...]**
  + **Importance: medium**

1. **Max.request.size**

* **Whenever producer will send message to cluster, the maximum size of one batch in bytes is max.request.size**
* **It is used to avoid huge request.**
* The parameters are as given below:
  + **Type: int**
  + **Default: 1048576**
  + **Valid Values: [0,...]**
  + **Importance: medium**