Kafka Retention Policy By Size

By Goutam Chowdhury

Common Configurations or set up for all below test

1. Create a file with 250000000 (two hundred fifty million ) lines.
2. One line = one Kafka message
3. Size of one line = Size of one Kafka message = 10 byte.
4. One topic with one partition

So we will be ingesting 250000000 (two hundred fifty million ) messages (2.5 GB of data)  into Kafka where each message size is 10 byte

To understand the Kafka retention policy we have to understand the segment file behavior in Kafka.

**Test1: Understand Segment and its size with default configuration:**

Configuration:

Topic with default configuration, i.e

segment.bytes : 1 GB

Result:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Starting offset | Log | Index | Timeindex | Total Size(B) | Total size in MB | Number of messages |
| **Segment 1** |  |  |  |  |  |  |  |
|  | 0 | 1073731931 | 526592 | 756252 | 1075014775 | ~1 GB | 63172307 |
| **Segment 2** |  |  |  |  |  |  |  |
|  | 63172307 | 1073732539 | 525888 | 748932 | 1075007359 | ~1 GB | 63172346 |
| **Segement 3** |  |  |  |  |  |  |  |
|  | 126344653 | 1073732950 | 526288 | 751476 | 1075010714 | ~1 GB | 63172371 |
| **Segment 4** |  |  |  |  |  |  |  |
|  | 189517024 | 1028021674 | 10485760 | 10485756 | 1048993190 | ~1 GB | 60482976 |
|  | 250000000 | 4249219094 | 12064528 | 12742416 | 4274026038 | ~4 GB | 250000000 |

**Observation:**

1. Four segments created and each of them has three files in it.
   1. X.log [Will only have messages with offset and metadata]
   2. X.index [ offset searching index which will used to search an offset inside .log file ]
   3. X.timestamp [ Offset searching index by timestamp which will be used to search an offset inside .log file]

Where X = starting offset of the file.

1. Each segment size is about 1 GB but it not the same for all the same segment.
2. To ingest 2.5 GB data we need  around 5 GB disk space(more than 4 GB)
3. Each of the segment will have messages of 63172300 which is not again same in all log file (maximum difference 70 messages)
4. Each Segment file takes around 1 min time to be created

**Test2: Understand Segment and its size with manual configuration:**

Configuration:

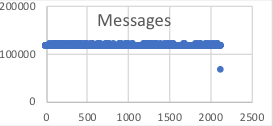
Topic configuration,

segment.bytes : "2000000" [2 MB] [ which is 500 times lesser than default(previous)]

Result:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  | Count | Byte | MB/GB |
| **Total File Count** | | | 6395 |  |  |
| **Total File Size** | | |  | 4275318358 | ~ 4 GB |
| **Log File Count** | | | 2131 |  |  |
| **Total Log File Size** | | |  | 4249219310 | ~4 GB |
| **Smallest Segment Size** | | |  | 1985988 | <2 MB |
| **Largest Segment Size (Exclude: last)** | | |  | 2003269 | >2MB |
| **Last Segment Size** | | |  | 22110656 | >22 MB |
| **Smallest Log File Size  (Excluding last)** | | |  | 1983636 | <2 MB |
| **Largest Log File Size** | | |  | 1999987 | ~2 MB |
| **Last Log Size** | | |  | 1139130 | ~1 MB |
| **Smallest Number of Messages in a Log File(except last)** | | | 116706 |  |  |
| **Largest Number of messages in a log file** | | | 117668 |  |  |
| **Number of messages in a Last log file** | | | 67020 |  |  |
| **Max. number of messages diff among log files** | | | 962 |  |  |

Messages inside log files



**Observation**:

1.        We need more than 4 GB to ingest 2.5 GB od data

2.        In a segment around 99% size will be allocated by log file (very small size for index and timeindex)

3.        Last log file size is smallest log file but due to much bigger index file and timestamp file size (inside same segment) , last Segment file size will be largest one.

4.        All log files will not have same number of messages in it. Number of messages between largest log file and smallest log file is around 1000( not considering last log file).

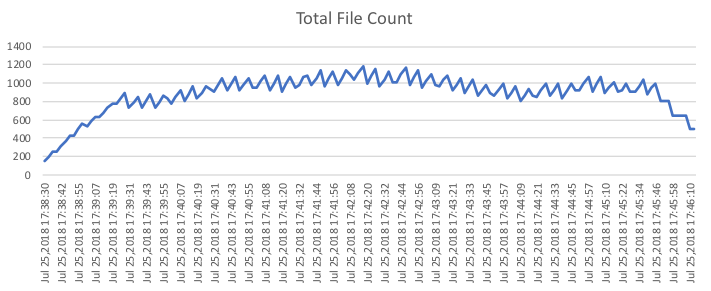
**Test3: Understand Retention policy by size**

1. Topic configuration
   1. segment.bytes : "2000000" [2 MB] [ by default 1 GB]
   2. retention.bytes:    “1997381” [<2MB] [ By default -1 means not set any value]
   3. [delete.retention.ms](http://delete.retention.ms/) : “15000” [15 sec] [ by default 1 day]

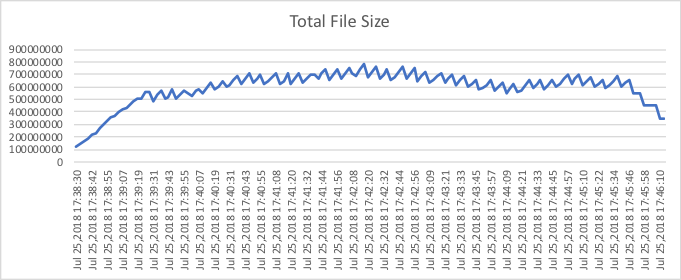
Files have been monitored every 3 secs

Result:

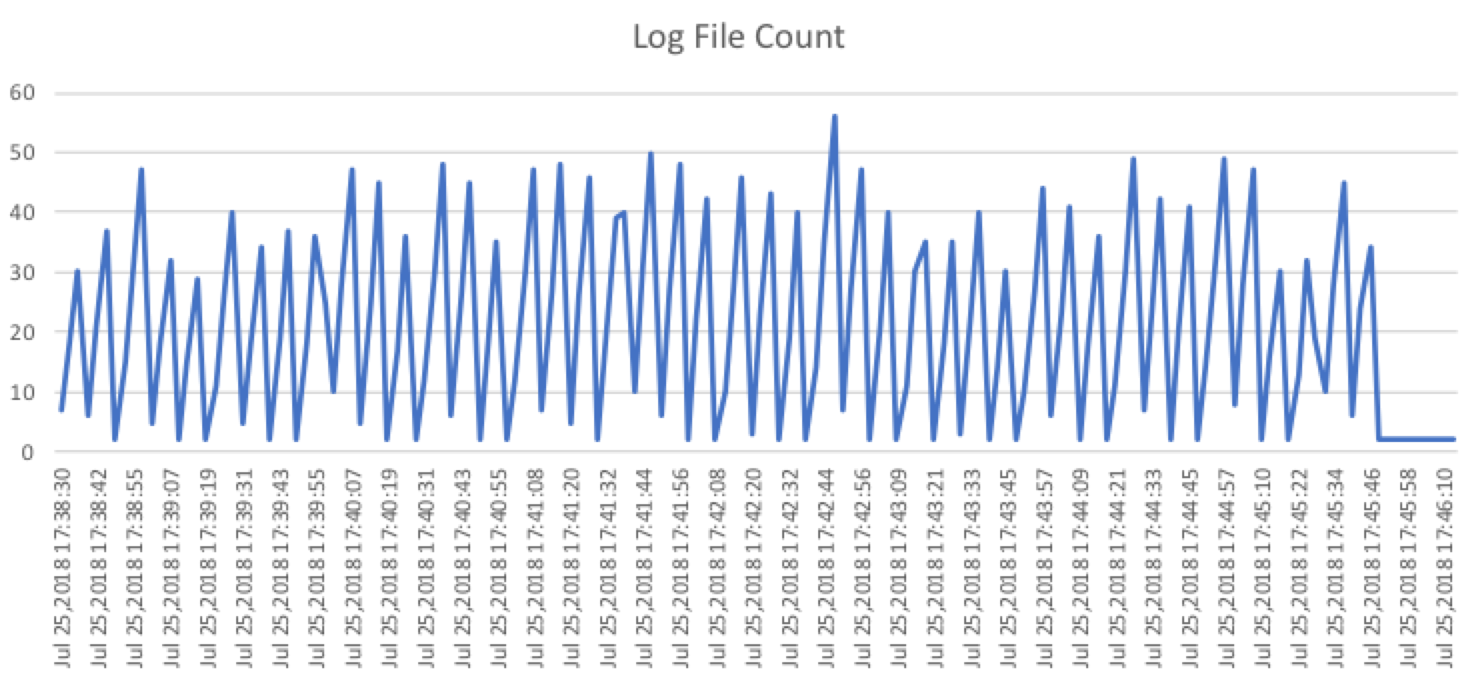
1. “Total file count” over “Time”



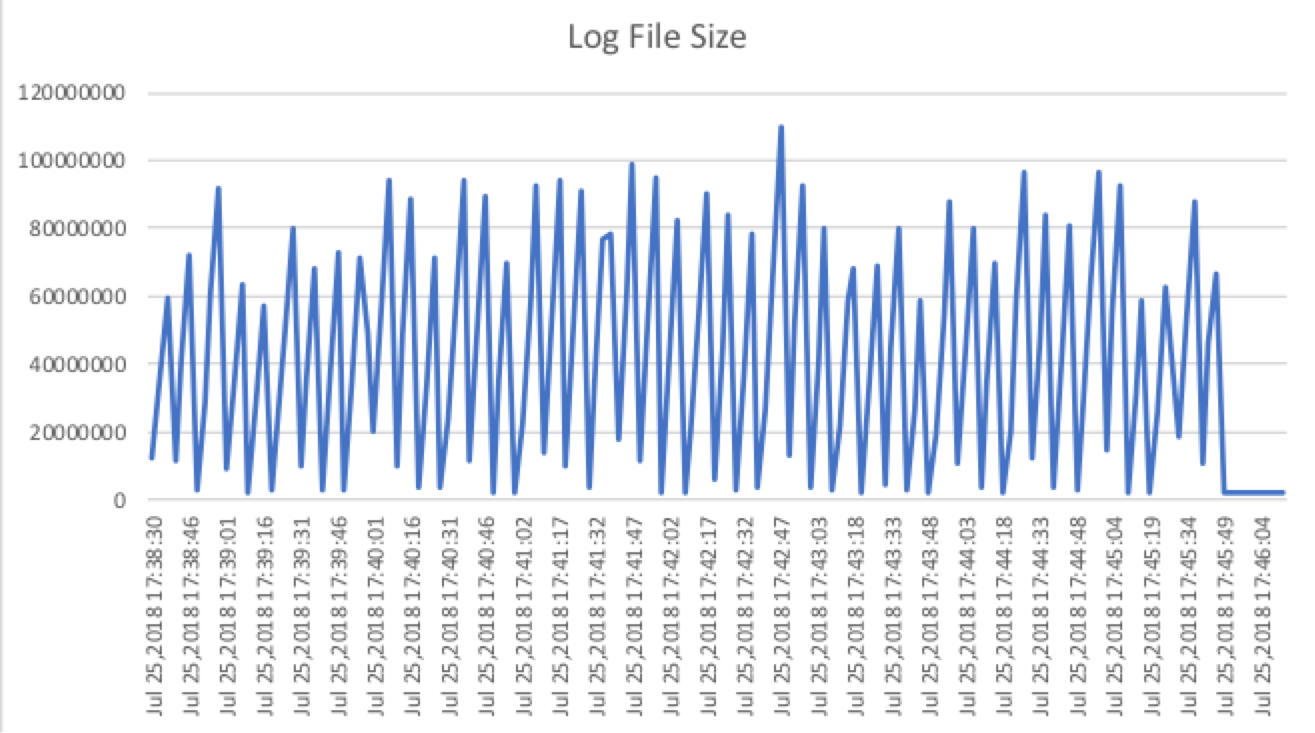
1. “Total File Size” over “Time”



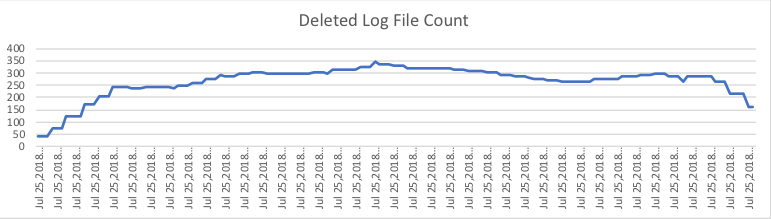
1. “Log File Count” over “Time”



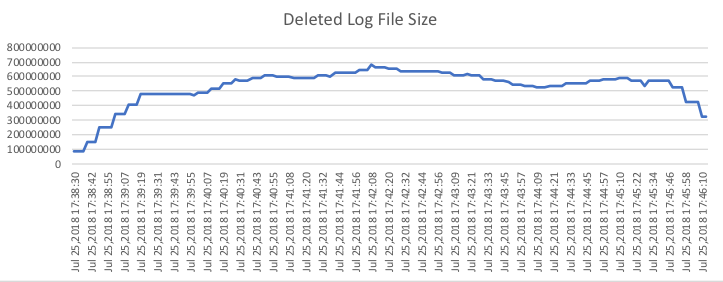
1. “Log File Size” over “Time”



1. “Deleted Log File Count” over “Time”



1. “Deleted Log File Size” over “Time”



**Observation**:

At any point of time  to ingest 2.5 GB of data

1. Maximum total file count <1200
2. Maximum total file size < 800 MB
3. Maximum log file count < 60
4. Maximum log file size < 200 Mb
5. Maximum deleted file count < 334
6. Maximum log file size < 350 Mb

So to ingest 2.5 GB of data we need 800MB disk space if we use below delete configuration

1. segment.bytes : "2000000" [2 MB] [ by default 1 GB]
2. retention.bytes:    “1997381” [<2MB] [ By default -1 means not set any value]
3. [delete.retention.ms](http://delete.retention.ms/) : “15000” [15 sec] [ by default 1 day]

But if we don’t use the we need 4 GB of disk space.

**So using above retention policy we can save more than 3 GB disk space while ingesting 2.5 GB of data.**

**BUT at the same time we will surely loose performance due to frequently accessing hard disk to delete files.**