7019-Test: Perf Comparison [12/10 LCS]

Setup

3 nodes cluster.
200+GB of data per node.
Single table.
Schema:

```
    --compaction "{'class': 'LeveledCompactionStrategy'}"
    --compression "{'sstable_compression': 'LZ4Compressor'}"
```

QPS: 3K/s.
Read : Write : Delete = 5 : 4 : 1

Timings

Start Time: Thu Dec 10 20:01:14 PST 2020
GarbageCollect Triggering Time: Thu Dec 10 21:41:14 PST 2020
GarbageCollect Duration: ~2 hours

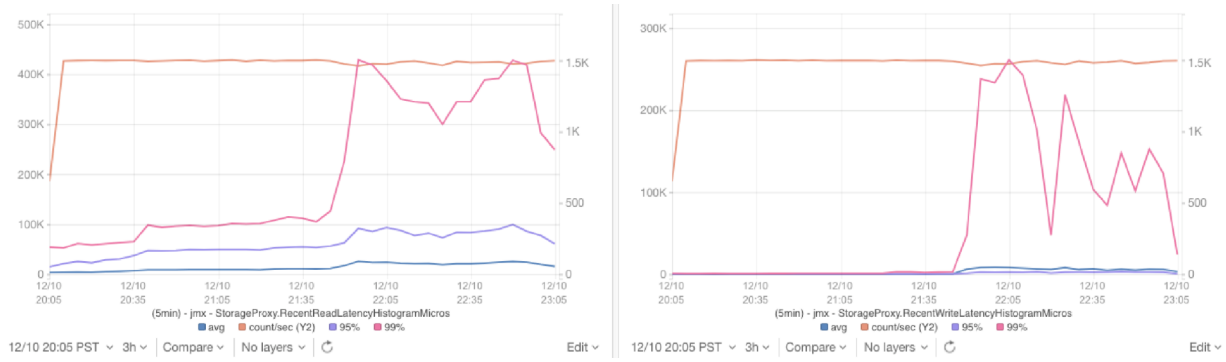
```
timestamp="2020-12-10T21:47:39,598-0800"
message="Starting Remove deleted data for tlp_stress.keyvaluelargeblob_2"

timestamp="2020-12-10T23:43:00,554-0800"
message="Finished Remove deleted data for tlp_stress.keyvaluelargeblob_2 successfully"
```

Result

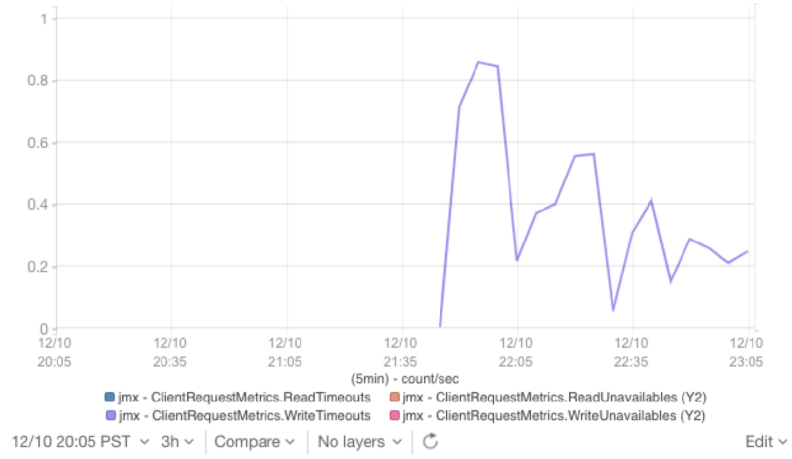
Metric	Steady State	With GarbageCollect
Read Throughput	1.5k/s	1.5k/s
Read Latency avg.	10.28k micros	21.92k micros
Read Latency p95	50.35k micros	87.41k micros
Read Latency p99	102.23k micros	388.14k micros
Write Throughput	1.5k/s	1.5k/s
Write Latency avg.	548.67 micros	3.06k micros
Write Latency p95	894.00 micros	8.05k micros
Write Latency p99	1.40k micros	160.93k micros

Read & Write Throughput and Latencies
Latencies (avg. p95, p99) increases during the GC is running.



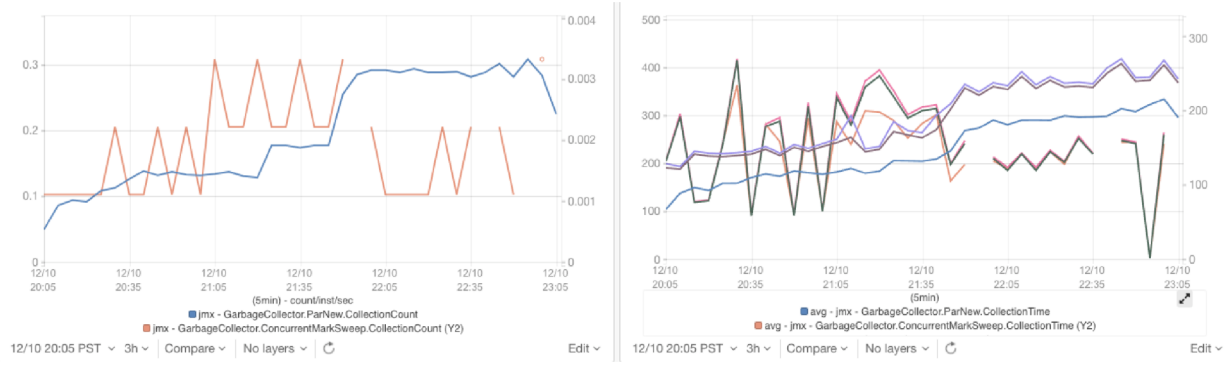
Write Timeouts

Write timeouts start to happen during the GC is running. The rate is low, below 0.9/s.



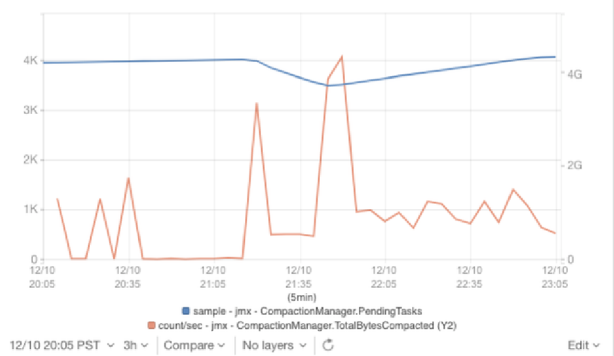
JVM GC Count & Duration

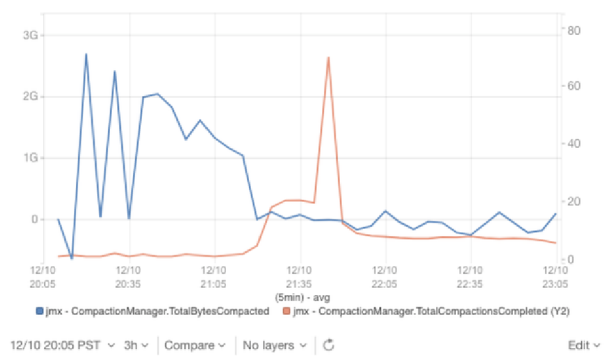
Both count and duration of the JVM GC increases with active `garbagecollect` running.



Compaction Rate & Throughput

When the `garbagecollect` is running, the rate of compaction tasks increases, meanwhile the throughput (i.e. TotalBytesCompacted) decreases. It means the compacted SSTables are smaller comparing with the ones compacted during just the steady state load.





Live SSTable & Unleveled SSTable Count

During `garbagecollect`, the live SSTables count decreases. They got compacted faster.

The unleveled SSTable count increases. GC has a significant impact on increasing the unleveled SSTables.