VictoriaMetrics vmstorage CPU Profiling Analysis

Bad Profile Analysis (100% CPU)

High CPU Usage (~100%) - Root Cause:
The profiling revealed that over 80% of CPU time was consumed by storage part merging and sorting:

- mergePartsInternal
- mergeBlockStreams
- mergeBlockStreamsInternal
- blockStreamWriter.WriteExternalBlock
- sort.pdqsort (recursive sorting)

These operations were triggered frequently, suggesting overly aggressive or inefficient merging logic. Background merge loops were also consuming significant CPU constantly.

Result: CPU saturated, little room for handling queries or other tasks.

Good Profile Analysis (36% CPU)

Improved CPU Usage (~36%) - Findings:
After improvements, the merging and sorting functions dropped significantly in sample frequency:

- merge functions: from ~37% to ~12%
- sort.pdqsort: from ~40% to ~33%
- smallPartsMerger: from ~17% to ~4%

Query-handling functions became more prominent in the CPU profile, indicating freed CPU resources.

Result: Balanced workload, reduced background CPU load, more CPU available for queries and

ingestion.

Summary of Actions

Recommended Actions to Reduce CPU Usage:

1. Increase merge thresholds:
   - -storage.minMergeMultiplier=2.0-3.0
   - -storage.maxPartsInMerge=10-20

2. Reduce creation of small parts:
   - -insert.maxQueueDuration=1-2s
   - -storage.maxInmemoryPartSize=64MB

3. Control background merger:
   - Try: -storage.disableBackgroundMerge

4. Reduce sort pressure:
   - Batch ingestion data
   - Deduplicate upstream if possible

5. Monitor:
   - Merge metrics (duration, frequency)
   - Profile using pprof or speedscope regularly