SailPoint®

SQLite Event Manager File Category Store Replication

Performance Test Results

v8.2.0.3000

Table of Contents

Performance Test Overview

The objective of this testing effort was to measure any performance gains or losses from switching from LiteDB to SQLite.

SQLite Changes

The Event Manager (EM) File Category store replication was updated to use SQLite, replacing LiteDB. Discovery of the possibility of the LiteDB cache files ability to become corrupted necessitated changes to be made. Thus, changes were made to improve reliability and stability of the file category store replication, by utilizing SQLite as the local cache. Enhancements were also made to the design with respect to synchronization of data to minimize event processing blockages and high memory usage due to the event queue backing up.

High-Level Design Summary

At the highest level, the proposed changes will include removing the PersistentDictionary component which uses LiteDB and replacing it with a new SQLite DC Cache Store which will handle updating and querying the DC cache.

All data will be stored in the SQLite DB and only the categories of the specific file that is being enriched will be queried; as opposed to the current design of fetching by BR hash and getting all file categories for all files of that folder.

File category data that is fetched from the SQLite cache will be cached in-memory using MemoryCache(Microsoft.Extensions.Caching.Memory). See the SiqApi project in the website solution for example usages of this cache.

Details of changes are covered In Epic SQISUS-709 and on confluence page "Event Manager - File Category Replicated Store to SQLite - Design".

Performance Test Scenario Details

All performance testing was completed against a Windows File Server (endpoint) which was located in the same domain as FAM core services. The Activity Monitor was located on a separate server than the Event Manager. Please see section 'Performance Environment, Allocation of Resources' below for details. Two scenarios were determined for testing. Each scenario generated activity on the endpoint via a PowerShell script performing a series of file reads on sensitive files, starting with the roots of the folder structure and working its way down. See 'Performance Data Set' for more information. The configuration key <add key="maxDCCacheSizeInMB" value="0"/> was set to 0 to force test throughout to LiteDB (before changes) and SQLite (after changes) exclusively. All performance testing was completed on File Access Manger version 8.2.0.3000 with these specific changes applied. These changes are also Intended for 8.2 SP4, 8.3 SP3 and 8.4. Functional validation testing was also performed. Details are not captured in this document.

1. 8 Hours of Activity scenario - this scenario simulates 8 hours of continuous activity on the Windows File Server - In this scenario the test run ends after 8 hours.
2. 500,000 Activity scenario simulates continuous activity until 500,000 events are generated on the windows file server at a swifter pace.

Performance Environment, Allocation of Resources

The following table lists out the allocation of recourses for the performance environment used in this testing effort. The resource allocation Is based on the standard recommended allocation provided to customers deploying FAM.

Resource Allocation:

VM Name	FAM Services	Processor Cores	RAM (GB)	Disk Space (GB)
FAM-Perf-Core-1	Agent Config Manager / API / Business Website / Sched. Task Handler / User Interface / RabbitMQ	4 @ 2.60 GHz	8.00	60.00

FAM-Perf-Core-2	Activity Analytics / Collector Synchronizer / Crowd Analyzer / Reporting Service / Workflow	4 @ 2.60 GHz	8.00	60.00
FAM-Perf-EvMgmt	Event Manager	4 @ 2.60 GHz	32.00	100
FAM-Perf-ES	Elastic Search	8 @ 2.60 GHz	32.00	C: 60.00 E: 500.00
FAM-Perf-DC-Eng	Central Data Classification	4 @ 2.60 GHz	16.00	60.00
FAM-Perf-Perm-E	Central Permission Collection	4 @ 2.60 GHz	32.00	260.00
FAM-Perf1-SQL	SQL FAM DB	8 @ 2.60 GHz	64.00	C: 60.00 E: 3.4 1 TB
FAM-Perf-FS3-CI	Data Server, Activity Monitor	4 @ 2.60 GHz	8.00	C: 60.00 E: 100

Performance Data Set

All file server data Is located on the Windows File Server FAM-Perf-FS3-CI on drive E:. The drive consists of 90 GB of data, with 90% of the data classified under one or more of the out of the box (OOTB) data classification categories "PHI", "ICD", and "PII".

Performance Test Results Overview

The table below gives an overview of the performance gains from switching to SQLite.

Analyzing performance test results switching to SQLite gains a 49.97% decrease in the event collector processing time; i.e. processing events took half the time previously taken.

How to read the chart below:

Run Duration: The duration of activity performed on the configured endpoint.

> ***Note 1*** - This Is the duration for which activity was being generated on the target Windows File Server at a *consistent* pace, **not** sending events as quickly as they can be processed; therefore we would **not** expect to see the run duration time decrease with a decrease of processing time.

Total Events: The total number of events processed by the EM during the timeframe of the test scenario. Same test scenario as Run Duration is applied here. Events sent at a consistent pace, not as fast as possible.

> ***Note 2*** - This can be greater than the simulated activity as there are other window processes which are captured by the Event Collector.

Avg Time Processing Events: This Is a calculated value. Calculated from the averaging "Time Since Last Log" values from the Event Collector statistics log excluding outliers.

- Outliers were defined as either: a) If it is less than 50% of the 25% percentile average, b) If it is 1.5 times the interquartile range greater than the third quartile, or c) If It Is 1.5 times the interquartile range less than the first quartile.

Total EM Time: The total time the Event Collector has been processing events since the Event Manager has started. The Event Manager is restarted at the start of each test providing the total time the Event Collector has been processing events for the specific test scenario.

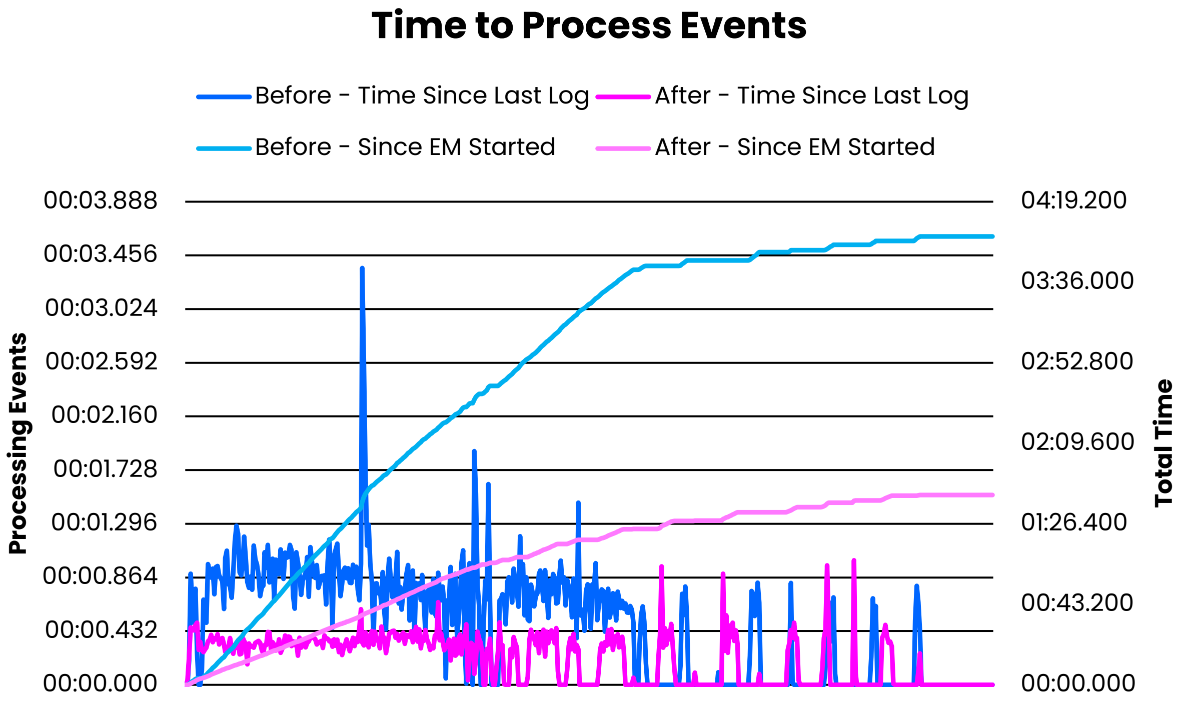
Scenario	Run Duration [1]	Total Events [2]	Avg Time Processing Events	Total EM Time (mm:ss.000)
Before Applying SQLite Changes				
8Hr Activity	8 hrs. 7 mins.	665,809	00:00.785	04:00.557
500K Activity	4 hrs. 51 mins.	503,008	00:00.794	02:56.941
After Applying SQLite Changes				
8Hr Activity with SQLite Changes	8 hrs. 4 mins.	617,983	00:00.350	01:41.856
500K Activity with SQLite Changes	4 hrs. 16 mins.	502,546	00:00.432	01:44.018

Table 1: Results Overview

Test Results Details

We see improved processing time and improvements in the total time It takes to process similar amount of events for each scenario when comparing before and after the SQLite changes. Comparing total events in each scenario we see our ability to process the same amount of events Is not Impeded. Recall, we do not expect to see a decreased run duration as events were created at the same pace for each test.

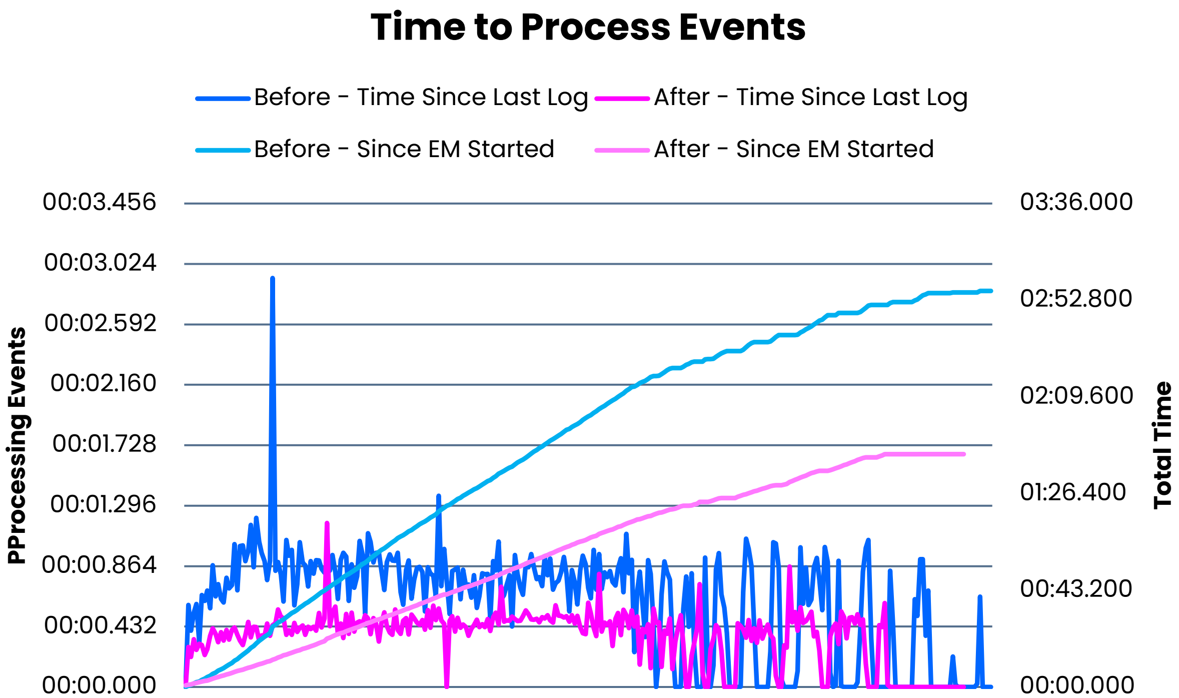
8 Hours of Activity



Graph 1: Time Processing Events – 8 Hours of Activity, comparison of before and after SQLite Changes

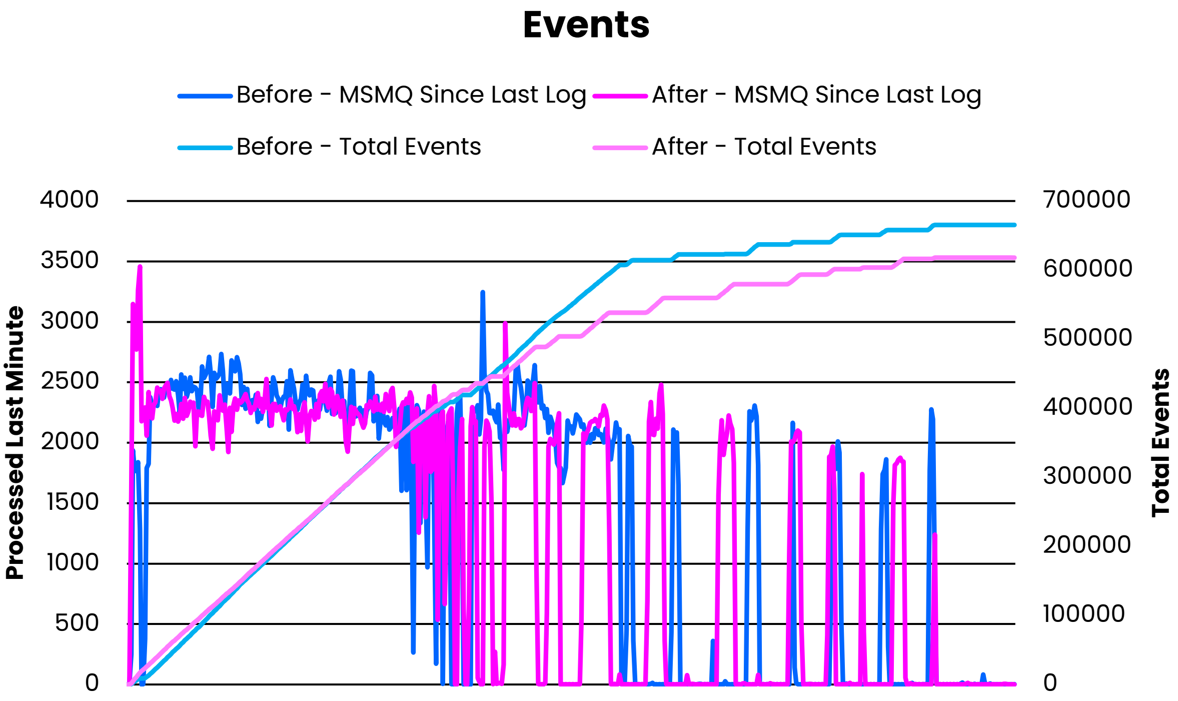
Note: The lulls in activity after ~2 hours Is likely due to the way the scenario Is constructed reading activity further down in the folder structure.

500K Activity



Graph 2: Time Processing Events - 500K Activity, comparison of before and after SQLite Changes

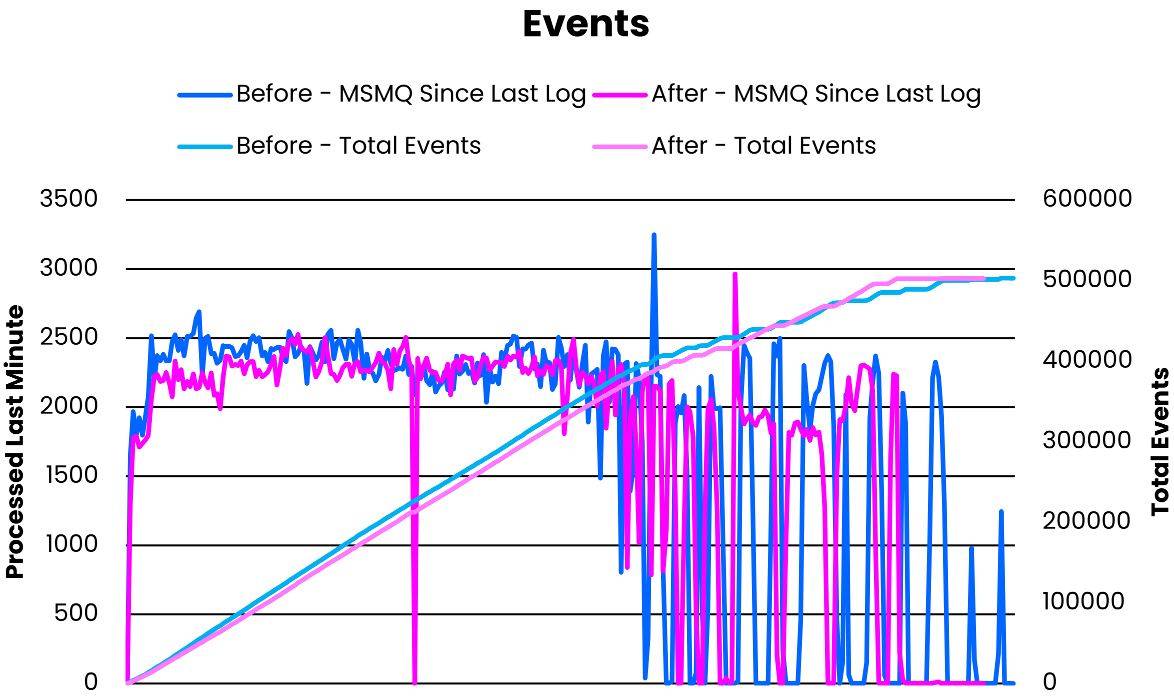
8 Hours of Activity



Graph 3: Events - 8 Hours of Activity, comparison of before and after SQLite Changes

Note: The increased total events processed is due to additional system events occurring on the endpoint during the testing period.

500K Activity



Graph 4: Events - 500K Activity, comparison of before and after SQLite Changes

Conclusion

In conclusion, with the transition to SQLite, we not only improve the reliability of the event manager data classification file category cache, but also improve the overall performance and efficiency of event processing.

Revision History

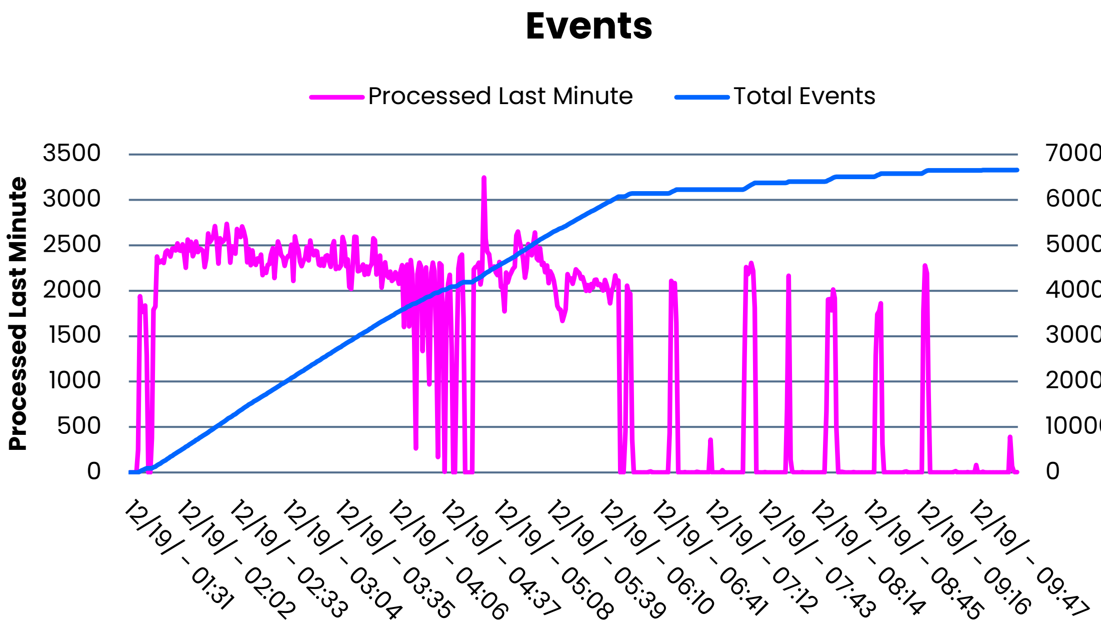
Version	Author	Date	Notes
Rev 1.0	Alejandro Cabrera-Salazar	12/21/2022	Initial Draft
Rev 1.1	Barbara Hodgkin	12/27/22	Review

Appendix

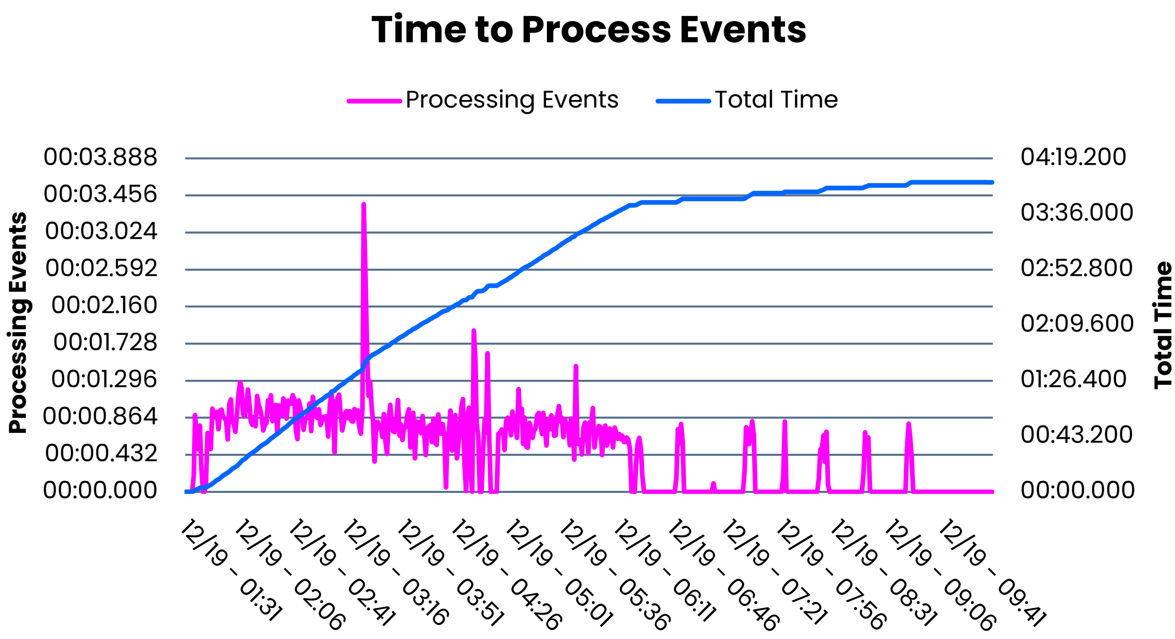
Before SQLite Changes

The following charts outline test results for running both scenarios prior to applying the SQLite changes.

8 Hours of Activity

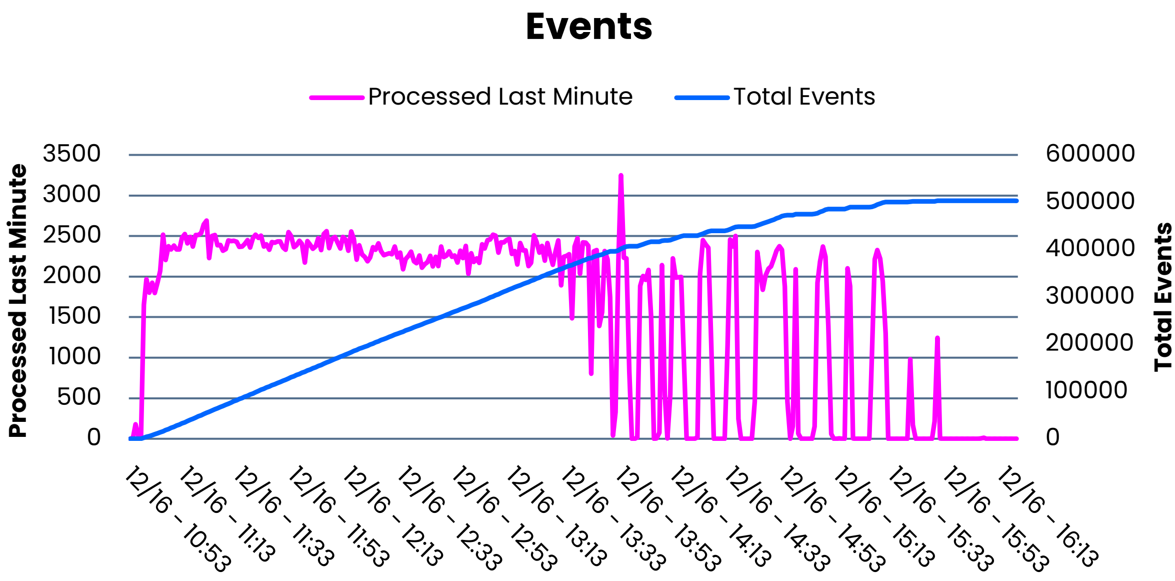


Graph 5: Events Processed - 8 Hours of Activity, before SQLite Changes

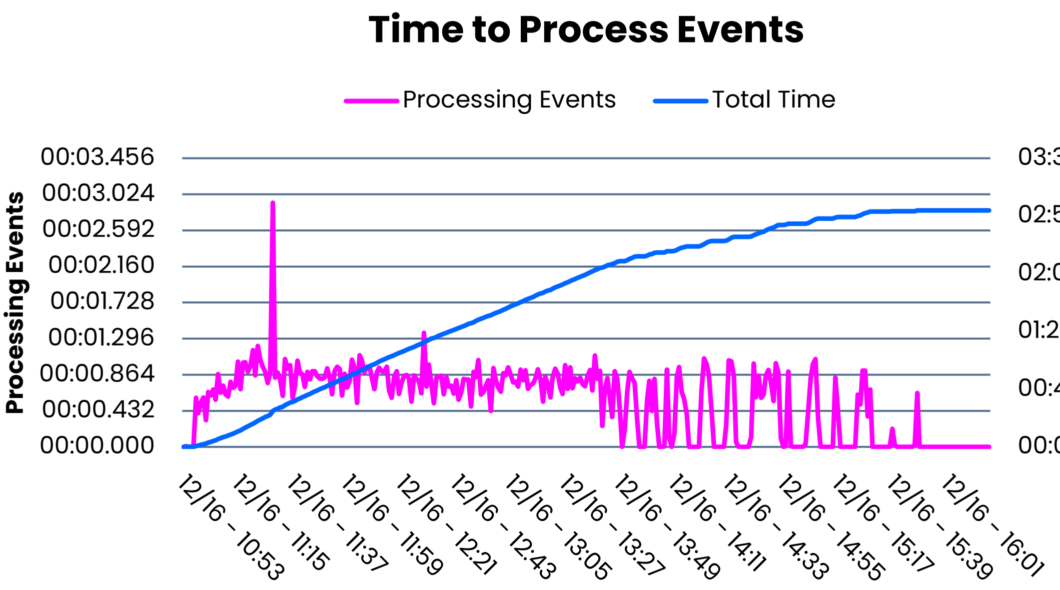


Graph 6: Time Processing Events - 8 Hours of Activity, before SQLite Changes

500K Activity



Graph 7: Events Processed - 500,000 Activities, before SQLite Changes

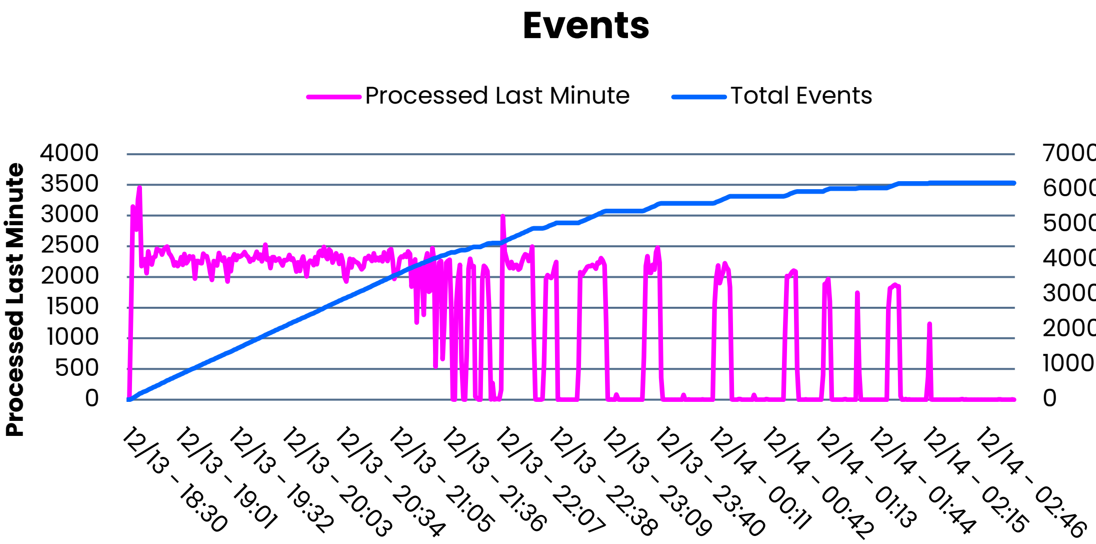


Graph 8: Time Processing Events - 500,000 Activities, before SQLite Changes

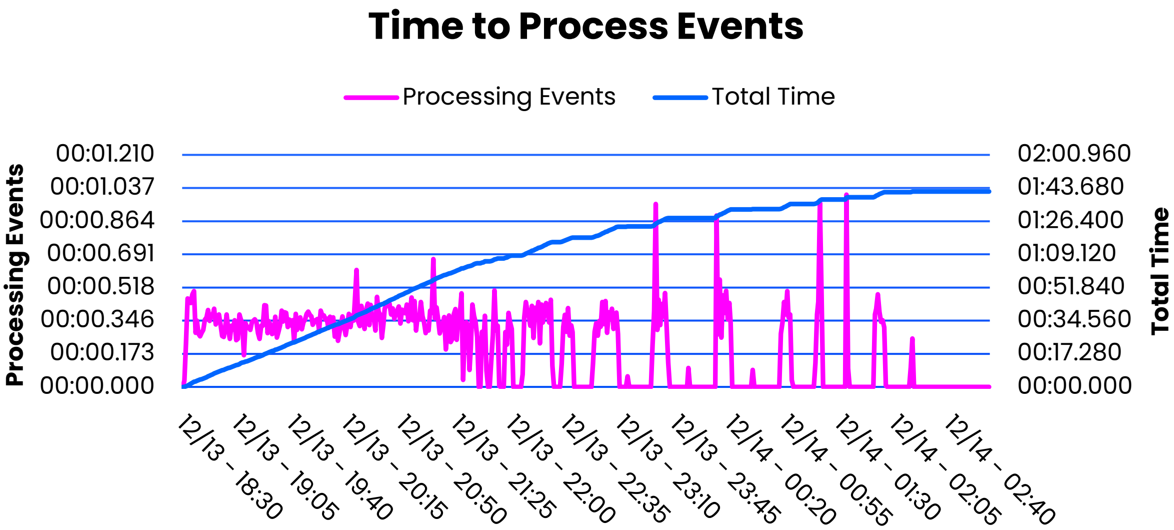
After SQLite Changes

The following charts outline test results for running both scenarios after applying the SQLite changes.

8 Hours of Activity

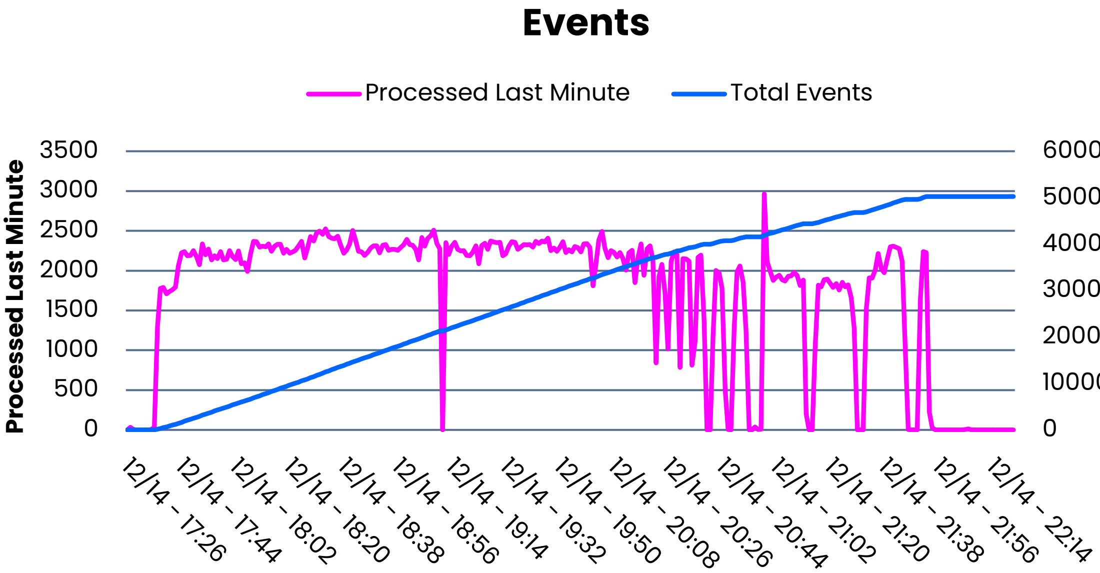


Graph 9: Events Processed – 8 Hours of Activities, with SQLite Changes

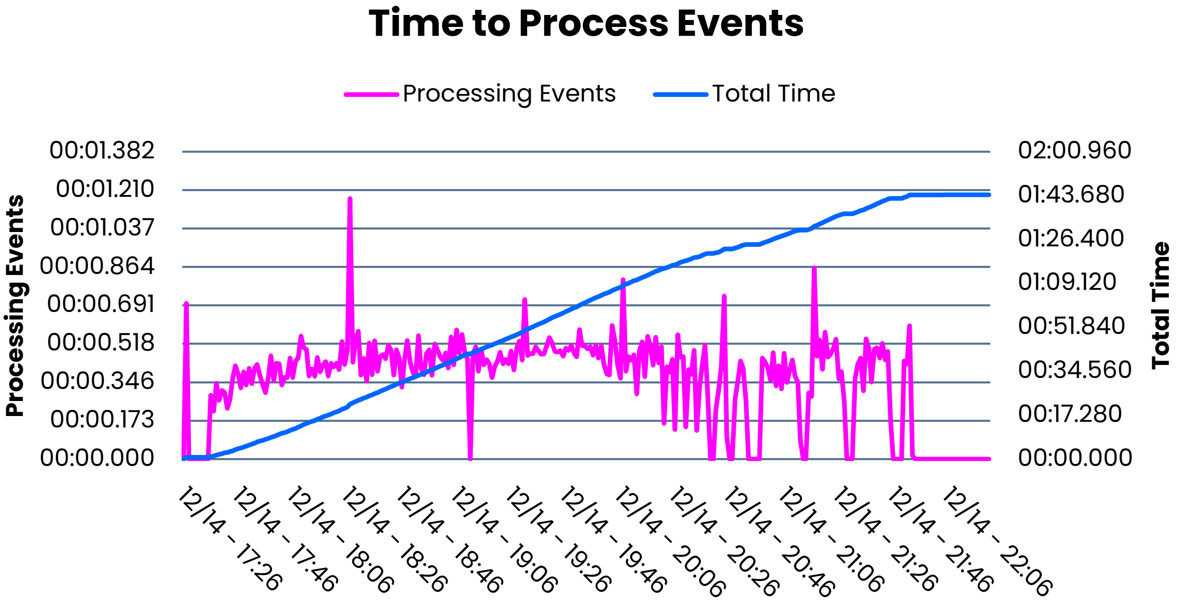


Graph 10: Time Processing Events – 8 Hours of Activities, before SQLite Changes

500K Activity



Graph 11: Events Processed - 500,000 Activity, with SQLite Changes



Graph 12: Time Processing Events - 500,000 Activity, with SQLite Changes