eXtremeDB Financial Edition 6.0 on Lucera Compute™ (16 Core SSD v2, SmartOS)

SUT ID: KDB141111

STAC-M3™ BENCHMARKS
(Antuco Suite)

Test date: 11 November 2014
Draft 1.0, November 18 2014

Benchmark specs:
STAC-M3™
(Antuco suite)

These tests followed STAC benchmark specifications proposed or approved by the STAC benchmark council (see www.STACresearch.com). Be sure to check the version of any specification used in a report. Different versions may not yield results that can be compared to one another.
Contents

REFERENCES ........................................................................................................................3
SUMMARY ............................................................................................................................5
REPORT CARD .......................................................................................................................6
STAC REPORT CARD (CONT’D) ...........................................................................................7
STAC REPORT CARD (CONT’D) ...........................................................................................8
STAC REPORT CARD (CONT’D) ...........................................................................................9
CHART VIEW .........................................................................................................................9
1. OVERVIEW OF THE STAC-M3 BENCHMARK SPECIFICATIONS ..................................15
2. PRODUCT BACKGROUND ...............................................................................................20
3. PROJECT PARTICIPANTS AND RESPONSIBILITIES .......................................................23
4. CONTACTS .....................................................................................................................23
5. RESULTS STATUS ..........................................................................................................23
6. SPECIFICATIONS ...........................................................................................................24
7. LIMITATIONS ..................................................................................................................24
8. STACK UNDER TEST .......................................................................................................24
9. VENDOR COMMENTARY .................................................................................................24
ABOUT STAC .......................................................................................................................25
References

[1] Specifications used for this benchmark: STAC-M3 Benchmark Specifications, Antuco Suite, Rev N – 
Disclaimer

This document was prepared by the Securities Technology Analysis Center (STAC®) at the request of McObject. It is provided for your internal use only and may not be redistributed, retransmitted, or published in any form without the prior written consent of STAC. “STAC” and all STAC names are registered trademarks or trademarks of the Securities Technology Analysis Center LLC. All other trademarks in this document belong to their respective owners.

The test results contained in this report are made available for informational purposes only. Neither STAC nor the vendor(s) supplying the information in this report guarantee similar performance results. All information contained herein is provided on an “AS IS” BASIS WITHOUT WARRANTY OF ANY KIND. STAC explicitly disclaims any liability whatsoever for any errors or otherwise.
Summary

STAC recently performed the baseline STAC-M3™ Benchmarks on a stack involving McObject’s eXtremeDB Financial Edition 6.0 hosted on a Lucera Compute™ cloud platform. This report documents the benchmark results. The McObject code used to execute these benchmarks is available from the STAC Vault for inspection and use by qualified STAC Benchmark Council members.

This report documents the first STAC-M3 audit involving a cloud platform.

Unlike STAC-M3 tests with previous versions of eXtremeDB Financial Edition, in which the benchmark operations were implemented completely in C++ and compiled into the database engines, this benchmark implementation was written largely in Python using SQL calls via TCP/IP to the database engines.

In all, the STAC-M3™ specifications deliver dozens of test results, which are presented through a variety of tables and visualizations in this report. Of these, McObject chose to highlight a few, as follows:

As compared to previous STAC-M3 benchmarks run with a non-eXtremeDB database:

- Lower response times than the previously published best results for 5 of the 17 operations
- Over 3x the performance of the previously published best result for 50T.STATS-UI (SUT ID: KDB131007)
- Over 2x the performance of the previously published best result for 100T.STATS-UI (SUT ID: KDB130603)
- Over 2x the performance of the previously published best result for 1T.NBBO (SUT ID: KDB131007).

NOTE: This report can be compared to earlier reports

Benchmark IDs in this report that end with "TIME" can be compared to prior benchmarks that end in "LAT2". Likewise, "MBPS" benchmarks are the same as "BPS" benchmarks in earlier reports. These are ID revisions only. See "Benchmark Identifiers" in Section 1 for details.

A STAC-M3 Report Card appears at the top of the report as a convenience for readers who want to get straight to the results. However, we recommend that readers who are not part of the STAC-M3 Working Group first read Section 1 (Overview) to get a feel for the test cases and metrics.

Getting the most from these results

Any interested party can analyze public STAC Reports to compare the performance of different systems. However, members of the STAC Benchmark Council are able to put these reports to much greater use. Qualified members may:

- Read the detailed Configuration Disclosure for the system tested in this report
- Read the detailed test specifications
- Access additional reports in the confidential STAC Vault™
- Obtain the materials to run the STAC-M3 Benchmarks on their own systems
- Discuss benchmarks, technologies, and related business issues with their peers.

To join the Council or upgrade your membership, please contact council@STACresearch.com.
Report Card

STAC-M3™ Benchmarks for SUT ID XTR141111:
eXtremeDB Financial Edition 6.0 on
Lucera Compute™ (16 Core SSD v2, SmartOS)

Storage efficiency

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Storage Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.v1.1.STORAGE.EFF</td>
<td>166%</td>
</tr>
</tbody>
</table>

Light-Compute Benchmarks

High Bid
(1 Client Thread Requesting)

Return the high bid for a certain 1% of symbols over varying timeframes. Run the year-high bid a second time (YRHIBID-2) without clearing the cache.

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
<th>MB/sec</th>
</tr>
</thead>
</table>

* Megabytes read per second from persistent media, according to kstat. That is, cache hits do not count as bytes read. See the STAC Configuration Disclosure for a discussion of this method and bytes-read oddities observed on this system.

Write Test

Perform the Basic Data Generation Algorithm for 1 day's data.

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.v1.1T.WRITE.TIME</td>
<td>Mean: 5,571 Max: 6,576</td>
</tr>
</tbody>
</table>
STAC Report Card (cont’d)
STAC-M3™ Benchmarks for SUT ID XTR141111

Post-Trade Analytics Benchmarks

VWAB on 1 Day’s Data
(1 Client Thread Requesting)

Return ~4-hour volume-weighted bid over a single day for certain 1% of symbols

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.v1.1T.VWAB-D.TIME</td>
<td>Query response time</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
</tr>
<tr>
<td></td>
<td>91</td>
</tr>
</tbody>
</table>

Theoretical P&L
(10 Client Threads Requesting)

For each of 10 Client Threads querying a unique set of 100 trades, find the amount of time until 2x, 4x, and 20x the size of each trade was traded in the market, and return the VWAP and total volume over those times intervals.

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.B1.10T.THEOPL.TIME</td>
<td>Query response time</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
</tr>
<tr>
<td></td>
<td>146</td>
</tr>
</tbody>
</table>

Market Snapshot
(10 Client Threads Requesting)

To each of 10 Client Threads querying a unique date, time, and set of symbols (1% of the total symbols), return the price and size information for the latest quote and trade for each symbol.

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.B1.10T.MKTSNAP.TIME</td>
<td>Query response time</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
</tr>
<tr>
<td></td>
<td>387</td>
</tr>
</tbody>
</table>
Research Analytics Benchmarks

Volume Curves
(10 Client Threads Requesting)

To each of 10 Client Threads querying a unique set of 20 dates and set of symbols (10% of the total symbols), return the average proportion of volume traded in each minute interval for each symbol across the date set.

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.1.10T.VOLCURV.TIME</td>
<td>Query response time</td>
<td>6,890</td>
<td>6,717</td>
<td>1,941</td>
<td>11,850</td>
</tr>
</tbody>
</table>

Aggregated Stats
(10 Client Threads Requesting)

For each of 10 Client Threads querying a unique exchange, date, and start time, return basic statistics calculated for the entirety of the 100-minute time range following the start time. Time ranges always cross a date boundary.

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.1.10T.STATS-AGG.TIME</td>
<td>Query response time</td>
<td>29,797</td>
<td>30,271</td>
<td>16,081</td>
<td>133,564</td>
</tr>
</tbody>
</table>

Stats Over Unpredictable Intervals
(Variable Client Threads Requesting)

To each of some number of Client Threads querying a unique exchange, date, and start time, return basic statistics calculated for each minute interval in a 100-minute time range following the start time. Start times are offset from minute boundaries by a random amount. Time ranges always cross a date boundary. Tests must be run with 1, 10, 50, and 100 Client Threads. Tests with other numbers of Client Threads are optional.

<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.1.1T.STATS-UL.TIME</td>
<td>Query response time</td>
<td>941</td>
<td>626</td>
<td>551</td>
<td>1,605</td>
</tr>
<tr>
<td>STAC-M3.1.10T.STATS-UL.TIME</td>
<td>Query response time</td>
<td>2,821</td>
<td>2,774</td>
<td>1,820</td>
<td>4,705</td>
</tr>
<tr>
<td>STAC-M3.1.50T.STATS-UL.TIME</td>
<td>Query response time</td>
<td>2,613</td>
<td>2,601</td>
<td>1,796</td>
<td>4,204</td>
</tr>
<tr>
<td>STAC-M3.1.100T.STATS-UL.TIME</td>
<td>Query response time</td>
<td>4,225</td>
<td>4,230</td>
<td>2,837</td>
<td>5,775</td>
</tr>
</tbody>
</table>
STAC Report Card (cont’d)
STAC-M3™ Benchmarks for SUT ID XTR141111

NBBO Benchmark

```
<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.B1.1T.NBBO.TIME</td>
<td>Time to write all results</td>
</tr>
</tbody>
</table>
```

Multi-day/Multi-User VWAB Benchmark

```
<table>
<thead>
<tr>
<th>Spec ID</th>
<th>Latency (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAC-M3.v1.100T.VWAB-12D-NO.TIME</td>
<td>Query response time</td>
</tr>
</tbody>
</table>
```

Chart view

The charts that follow illustrate or elaborate on the results above:

- Figures 1 through 4 plot the mean response time (TIME) benchmarks for all of the operations.
- Figures 5 and 6 analyze the individual response-time observations for the multi-user/multi-day VWAB benchmark (STAC-M3.v1.100T.VWAB-12D-NO.TIME), first by sorting the results by response time, then by plotting them in a histogram.
- Figure 7 provides a more explicit look at multi-user scaling by plotting the response time for the intervalized statistics benchmark (STAC-M3.B1.[n]T.STATS-UI.TIME) against the number of simultaneously requesting client threads (n).
- Figures 8 and 9 take the 100-client-thread case of Figure 7 and analyze the individual response-time observations, first by sorting the results by response time, then by plotting them in a histogram.

Refer to Section 1 (Overview) and the tables above for explanations of the benchmark IDs used in the charts.

The axes in the bar charts are fixed, so that results from this SUT may be visually compared to those of other SUTs. Because the results of future SUTs are unpredictable, the axes use a log scale.
Figure 1

Figure 2
Figure 3

Figure 4
Figure 5

STAC-M3™ Multi-User/Multi-Day VWAB Mean Response Times, Sorted
STAC-M3.v1.100T.VWAB-12D-NO.TIME -- SUT ID: XTR141111
eXtremeDB Financial Edition 6.0 on
Lucera Compute™ (16 Core SSD v2, SmartOS)

Observation (5 runs for each of 100 threads), sorted by response time

Source: STAC®
www.STACresearch.com
Copyright © 2014 STAC

Figure 6

STAC-M3™ Multi-User/Multi-Day VWAB - Response Time Histogram
STAC-M3.v1.100T.VWAB-12D-NO.TIME -- SUT ID: XTR141111
eXtremeDB Financial Edition 6.0 on
Lucera Compute™ (16 Core SSD v2, SmartOS)

Response time (milliseconds)

Source: STAC®
www.STACresearch.com
Copyright © 2014 STAC
Figure 7

Figure 8
Figure 9
1. Overview of the STAC-M3 Benchmark specifications

Analyzing time-series data such as tick-by-tick quote and trade histories is crucial to many trading functions, from algorithm development to risk management. But the domination of liquid markets by automated trading—especially high-frequency trading—has made such analysis both more urgent and more challenging. As trading robots try to outwit each other on a microsecond scale, they dish out quotes and trades in ever more impressive volumes. This places a premium on technology that can store and analyze that activity efficiently. For example, the faster an algorithm developer can back-test and discard a haystack of unprofitable ideas, the faster he will find the needle of a winning algorithm, leaving more time to exploit it in the market.

The STAC Benchmark Council has developed the STAC-M3 Benchmarks in order to provide a common basis for quantifying the extent to which emerging hardware and software innovations improve the performance of tick storage, retrieval, and analysis.

STAC-M3 tests the ability of a solution stack such as columnar database software, servers, and storage, to perform a variety of operations on a large store of market data. The STAC-M3 Working Group designed these test specs to enable useful comparisons of entire solution stacks (i.e., to gauge the state of the art) as well as comparisons of specific stack layers while holding other layers constant. Comparisons can include (but are not limited to):

- Different storage systems, including SSD, DRAM, interconnects, and file systems
- Different server products, processors, chipsets, and memory
- Different tick-database products

As shown below, the test setup for STAC-M3 consists of the “stack under test” (SUT) and client applications. No restrictions are placed on the architecture of the SUT or clients (though members of the STAC-M3 Working Group frequently provide input on architectures they would like to see tested). Threads within the clients take in Randomized Reference Data (RRD) such as dates and symbols, submit requests for the required operations, receive responses, and store the timings and results from these queries. Vendor-supplied code for the operations and latency calculations are subjected to a combination of source-code inspection and empirical validation.
Dataset

STAC-M3 draws from client experience with equities and FX use cases. The database is synthetic, modeled on NSYE TAQ data (US equities). While it is also desirable to test with real data, synthetic data has three advantages that make it compelling for this STAC-M3 suite:

- Synthetic data allows us to control the database properties exactly, which in turn allows us to randomize elements of queries from project to project while keeping the resulting workload exactly the same (for example, we control how much volume is associated with each symbol).
- Synthetic data does not incur fee liability from a third party such as an exchange.
- Synthesizing the data makes it easy to scale the database to an arbitrarily large size and run benchmarks against projected future data volumes.

The dataset consists of high-volume symbols and low-volume symbols in proportions based on observed NYSE data. The data volume per symbol was based on doubling the typical volume in NYSE TAQ in 1Q10. The resulting database is considerably smaller than databases in use at customer sites. This was a deliberate choice by the STAC-M3 Working Group to minimize the cost of running benchmarks while still yielding valuable results. Benchmarks that scale the database to the size of existing customer footprints and well beyond are contained in the Kanaga suite of STAC-M3 Benchmark specifications.

Metrics

The key metric in STAC-M3 is query response time. This measurement is performed in the client. A client thread gets a local timestamp (t_submit) just before submitting a query. When it receives the complete results of the query (sorted appropriately), the client immediately gets a second timestamp (t_last). Query response time is (t_last) - (t_submit).

```
Client code
Query submitted
SUT Interface
```

Some of the I/O-focused benchmarks also measure the bytes read per second from persistent storage (i.e., excluding server cache), which is computed from the output of appropriate system utilities.

The algorithms in all benchmarks are defined so as to keep the result sets small. This ensures that network I/O between the test clients and server(s) is negligible compared to back-end processing times.

Test cases

The current tests in the STAC-M3 suite are listed in the Summary Table below. These benchmarks operate by symbol on many fields of underlying tick data for both trades and quotes across varying time windows. The table classifies each test case as relatively heavy on I/O, compute, or both.

The tests require a client application that is written to a product API and is capable of submitting requests from 10 independent threads. As detailed in the table, some of the benchmarks call for one client instance making requests...
from a single thread, while others call for one client using 10 threads, and still others require 10 clients each using 10 threads (100 total requesting threads). One set of benchmarks (using the STATS-UI operation) tests multi-user scaling by running with 1, 10, 50, and 100 client threads. In all cases, benchmark results refer to per-request response times. For example, the mean of 10T.MKTSNAP.TIME is the mean time to satisfy a market-snapshot request from one of the threads, not the total time to satisfy requests from all 10 client threads. (Note, however, that a single request typically requires access to multiple fields, symbols, dates, and/or times.)

The range of dates eligible for querying depends on the benchmark. For example, some algorithms operate on dates randomly chosen throughout the year, some stick to a recent date range, and some always run on the most recent date (see the “Input Date Range” column of the table). The purpose of this differentiation is to provide a “recency bias” for those workloads where such bias is observed in the real world, while preventing such bias for those workloads that do not exhibit it in the real world.

Benchmark identifiers

The STAC-M3 Report Card and accompanying charts identify each benchmark unambiguously, as follows:

```
STAC-M3.ß1.100T.STATS-UI.TIME
```

In charts, the ID is sometimes decomposed, with part of it in the chart title or labels. Each individual STAC Benchmark™ specification has its own version number. The same version of a given spec may appear in multiple benchmark suites. Thus, the code names of the suites are irrelevant when making comparisons. Versioning individual specs enables the reader to compare a discrete result from this “stack under test” (SUT) to the corresponding result from another SUT. When making comparisons, be sure that the identifiers match exactly. If they do not, the benchmark results may not be capable of fair comparison.

<table>
<thead>
<tr>
<th>TIME = LAT2, and MBPS = BPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark IDs in this report with the suffix “TIME” are the same as those in prior reports with the suffix “LAT2”. Thus, these benchmarks may be fairly compared. Prior to 2014, benchmarks in the STAC-M3 Antuco suite had two metrics: LAT1 (time to receive first result) and LAT2 (time to receive all results). Given that LAT1 and LAT2 results were identical for all systems reported from 2011 to 2013, LAT1 was eliminated in 2014. In addition, LAT2 was redesignated TIME in order to clarify that the measurement represents a response time at the application level and to avoid confusion with micro-level storage latency.</td>
</tr>
</tbody>
</table>

Likewise, benchmark IDs ending in "MBPS" are the same as previous benchmarks ending in "BPS". The suffix now reflects the fact that data throughput is reported in megabytes per second.

Comparing STAC-M3 Antuco results with STAC-M3 Shasta results

As explained in the Appendix, the benchmark specs in the STAC-M3 Shasta suite are nearly the same as those in the STAC-M3 Antuco suite, but they are not the same. STAC-M3 Antuco forces the SUT to access storage, while STAC-M3 Shasta allows the SUT to substitute memory for storage. This means that it is not fair to compare Shasta and Antuco results (unless the results are for the same SUT and the purpose is to quantify the benefit provided by fully exploiting memory). Hence, Shasta and Antuco benchmarks have different IDs: Shasta root IDs contain a “/s”
Summary Table – STAC-M3 Benchmarks in the Antuco Suite

The table below gives a brief overview of each test in this STAC-M3 suite. Version numbers of 1 or greater indicate benchmark specs that have been approved. Versions less than 1 are proposed by the STAC-M3 Working Group but not yet voted on by the full STAC Benchmark Council.

<table>
<thead>
<tr>
<th>Root ID</th>
<th>Operation name</th>
<th>Ver</th>
<th>Number of requesting Client Threads</th>
<th>Algorithm performed on behalf of each requesting Client Thread</th>
<th>Algorithm I/O intensity</th>
<th>Algorithm compute intensity</th>
<th>Input date range*</th>
</tr>
</thead>
<tbody>
<tr>
<td>VWAB-D</td>
<td>VWAB-Day</td>
<td>1</td>
<td>1</td>
<td>4-hour volume-weighted bid over one day for 1% of symbols (like VWAP but operating on quote data, so much higher input volume).</td>
<td>Heavy read</td>
<td>Light</td>
<td>Last 30 days</td>
</tr>
<tr>
<td>VWAB-12D-NO</td>
<td>VWAB-12DaysNoOverlap</td>
<td>1</td>
<td>100</td>
<td>4-hour volume-weighted bid over 12 days for 1% of symbols. No overlap in symbols among client threads.</td>
<td>Heavy read</td>
<td>Light</td>
<td>Full year</td>
</tr>
<tr>
<td>YRHIBID</td>
<td>Year High Bid</td>
<td>β1</td>
<td>1</td>
<td>Max bid over the year for 1% of symbols.</td>
<td>Heavy read</td>
<td>Light</td>
<td>Full year</td>
</tr>
<tr>
<td>YRHIBID-2</td>
<td>Year High Bid Re-run</td>
<td>β1</td>
<td>1</td>
<td>Re-run of YRHIBID (same symbols) without clearing the cache.</td>
<td>Heavy read†</td>
<td>Light</td>
<td>Full year</td>
</tr>
<tr>
<td>QTRHIBID</td>
<td>Quarter High Bid</td>
<td>β1</td>
<td>1</td>
<td>Max bid over the quarter for 1% of symbols.</td>
<td>Heavy read</td>
<td>Light</td>
<td>Most recent quarter</td>
</tr>
<tr>
<td>MOHIBID</td>
<td>Month High Bid</td>
<td>β1</td>
<td>1</td>
<td>Max bid over the month for 1% of symbols.</td>
<td>Heavy read</td>
<td>Light</td>
<td>Most recent month</td>
</tr>
<tr>
<td>WKHIBID</td>
<td>Week High Bid</td>
<td>β1</td>
<td>1</td>
<td>Max bid over the week for 1% of symbols.</td>
<td>Heavy read</td>
<td>Light</td>
<td>Most recent week</td>
</tr>
<tr>
<td>STATS-AGG</td>
<td>Aggregate Stats</td>
<td>β1</td>
<td>10</td>
<td>One set of basic statistics over 100 minutes for all symbols on one exchange. Each 100-minute range crosses a date boundary.</td>
<td>Heavy read</td>
<td>Heavy</td>
<td>Full year</td>
</tr>
<tr>
<td>Operation</td>
<td>Type</td>
<td>Read</td>
<td>Write</td>
<td>Year</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>------</td>
<td>-------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATS-UI</td>
<td>Stats - Unpredictable Intervals</td>
<td>B1</td>
<td>1, 10, 50, 100 (more optional)</td>
<td>Per-minute(†) basic statistics over 100 minutes for all high-volume symbols on one exchange. Each 100-minute range crosses a date boundary.</td>
<td>Heavy read</td>
<td>Heavy</td>
<td>Full year</td>
</tr>
<tr>
<td>MKTSNAP</td>
<td>Market Snapshot</td>
<td>B1</td>
<td>10</td>
<td>Most recent trade and quote information for 1% of symbols as of a random time.</td>
<td>Heavy read</td>
<td>Heavy</td>
<td>Full year</td>
</tr>
<tr>
<td>VOLCURV</td>
<td>Volume Curves</td>
<td>B1</td>
<td>10</td>
<td>Create an average volume curve (using minute intervals aligned on minute boundaries) for 10% of symbols over 20 days selected at random.</td>
<td>Light read</td>
<td>Heavy</td>
<td>Full year</td>
</tr>
<tr>
<td>THEOPL</td>
<td>Theoretical P&amp;L</td>
<td>B1</td>
<td>10</td>
<td>For a basket of 100 trades on random dates, find the future times at which 2X, 4X, and 20X the trade size traded in each symbol. Trade sizes cause up to 5 days of forward searching. Calculate the corresponding VWAP and total volume traded over those periods.</td>
<td>Light read</td>
<td>Heavy</td>
<td>Full year</td>
</tr>
<tr>
<td>NBBO</td>
<td>NBBO</td>
<td>B1</td>
<td>1</td>
<td>Create the NBBO across all 10 exchanges for all symbols on the most recent day. Write to persistent storage.</td>
<td>Heavy read and write</td>
<td>Heavy</td>
<td>Most recent day</td>
</tr>
<tr>
<td>WRITE</td>
<td>Write</td>
<td>1</td>
<td>1</td>
<td>Write one day's quote data to persistent storage, following the same algorithm used to generate the randomized dataset used in the other Operations.</td>
<td>Heavy write</td>
<td>Light</td>
<td>n/a</td>
</tr>
<tr>
<td>STORAGE.EFF</td>
<td>Storage efficiency</td>
<td>1.1</td>
<td>n/a</td>
<td>Reference Size of the Dataset divided by size of the Dataset in the SUT format used for the performance benchmarks. Expressed as as percentage.</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

\(^{†}\) In some cases, one or more dates at the end of the year were excluded from eligibility to prevent an algorithm that crosses days from running out of input data.

\(^{†}\) Typically this will be reads from DRAM cache.

\(^{‡}\) In this case, interval start times are offset from minute boundaries by a consistent random amount per test run, so that the SUT cannot rely on pre-calculated minute statistics.
2. Product background

The stack under test (SUT) included the following:

- eXtremeDB Financial Edition 6.0
- Lucera Compute™, 16 Core SSD v2, SmartOS

McObject submitted the following information and claims about its products:

**eXtremeDB** has been used for over 13 years in high performance applications in Capital Markets, Military, Defense, Avionics, Netcom, and other markets. There are hundreds of companies using the development software and over 30 million copies deployed in applications around the world. McObject developed the eXtremeDB Financial Edition database management system to break through the financial IT data management bottleneck. The volume of data flowing through today’s automated capital markets is skyrocketing, and success for financial technology hinges on acting on this information instantly. But alternatives today either lack the performance for low latency applications or are too hard to use and expensive.

**eXtremeDB Financial Edition** leverages proven eXtremeDB embedded database strengths – including a streamlined hybrid in-memory/persistent storage database system design, multi-core optimization, maximum developer flexibility and high scalability – and adds specialized features to address key financial data management challenges. These features include:

- Columnar data layout for fields of type ‘sequence’. Sequences can be combined to form a time series, ideal for working with tick streams, historical quotes and other sequential data
- A rich library of vector-based math functions that accelerate management of time series data by maximizing L1/L2 cache use. Functions can be pipelined to form an assembly line of operations on sequences in support of statistical/quantitative analysis
- Use of common languages including SQL, C/C++, JAVA, C#, Python, ODBC/JDBC
- Distributed query support

These enhancements, and eXtremeDB’s rich core features and many specialized options, combine to make eXtremeDB Financial Edition the fastest database system with the most flexibility for financial applications.

**Architecture for Minimizing Latency**

eXtremeDB’s roots are in real-time embedded systems. How does it deliver a response that is fast and predictable enough for mission-critical avionics and defense electronics, industrial controllers and telecom/networking gear, as well as for ultra-low latency trading?

**In-memory and/or persistent tables.** eXtremeDB’s design is based on a core in-memory database system that eliminates performance-draining I/O, cache management, data transfer, and other sources of latency that are hard-wired into traditional disk-based relational database management systems (RDBMSs). McObject’s relentless focus on efficiency extends to persistent table management.

**Embedded or Client/Server.** eXtremeDB can run entirely within the application process, eliminating inter-process communication (IPC) between client and server modules. To exploit multiple CPUs/cores and distributed/parallel query execution, eXtremeDB can run in client/server mode.

**Short execution path.** eXtremeDB is written in C/C++, and its code path is very short: a “footprint” of approximately 150K for the core database system points to McObject’s unrelenting focus on eliminating even small potential sources of latency.
Programming interfaces. While eXtremeDB supports standard SQL, developers can use a direct access C/C++ API for faster and more predictable performance (i.e. to minimize latency spikes). eXtremeDB also provides the ability to store records directly as C/C++ data types (including structures, vectors and arrays), eliminating the overhead of conversion to SQL data types. Date, time and timestamp (down to the nanosecond) are fully supported with all eXtremeDB Financial Edition programming interfaces.

Memory management. eXtremeDB performs its own memory management instead of relying on the programming language run-time. This contributes to its performance advantage and scalability for multi-core configurations.

CPU cache efficiency. Keeping code and data in CPU (L1/L2) cache eliminates costly (in performance terms) fetches. eXtremeDB Financial Edition’s column-based data layout for sequences maximizes L1/L2 cache efficiency when working with tick and quote data, and its small code size maximizes the likelihood that the entire code path for a given operation is loaded into the cache at once.

<table>
<thead>
<tr>
<th></th>
<th>Symbol</th>
<th>Exchange</th>
<th>SEQ Open</th>
<th>SEQ Close</th>
<th>SEQ Volume</th>
<th>SEQ Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row1</td>
<td>IBM</td>
<td>NYSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RowN</td>
<td>ORCL</td>
<td>NASDAQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Traditional DBMSs bring rows of data into L1/L2 cache for processing. But financial data – such as ticks, trades and quotes – are better handled by a column-based layout that maximizes efficiency in fetching needed information. The result is higher performance: the database system benefits from L1/L2 cache speed and avoids excess fetching of data from RAM.
STAC-M3 Implementation

As shown in the diagram below, the core of the STAC-M3 implementation using eXtremeDB (the STAC-M3 Pack for eXtremeDB) is a set of Python language scripts that utilize eXtremeDB SQL to access a distributed eXtremeDB database, distributed over multiple physical storage locations.

From the standpoint of database access, the STAC benchmark is logically divided into client and server parts. The client part creates 1 to 10 Python processes and each process creates 1 to 10 Python threads (based on the benchmark requirements). Python threads, in turn, make calls to the eXtremeDB SQL engine client (“SQL client”) that connects to the eXtremeDB SQL server for a database partition (“Database shard”) through the TCP/IP-based network channel integrated with the SQL engine.

User-defined functions written in C++ are processed by SQL for four STAC-M3 test cases. The rest simply made standard eXtremeDB SQL calls.

The configuration used in this STAC-M3 implementation creates 64 shards distributed over 4 cloud servers. Once the client-server connection is established, the Python applications send SQL requests to servers and receive results sets. Each client sends SQL requests to all SQL servers, and consolidates the results received from them.

For more information on eXtremeDB Financial Edition, please visit http://financial.mcobject.com
Lucera submitted the following information and claims about the Lucera products in the SUT:

Lucera is a leading provider of on-demand infrastructure solutions for financial services. Lucera’s on-demand Infrastructure as a Service solution allows customers to scale their technology infrastructure to adapt to market demand while maintaining competitive performance in one or more of three collocated data center spaces. Lucera Connect™ provides on demand software defined networking capacity to connect to the financial ecosystem over direct fiber cross connects and across dedicated low latency transatlantic links. Lucera Compute™ is a paradigm shifting approach to high performance compute needs and employs a container based on demand solution which includes bare metal performance, in depth analytics and real time deployment and scale which allows customers to scale compute resources on demand to power trading, risk management, tick database offerings, development or any scale up or scale out financial workload.

Lucera allows companies to scale each layer of their infrastructure (compute, networking and storage) as market and business demands dictate, in near real time. This decoupling of infrastructure use (OpEx) from the legacy planning and deployment model (CapEx) allows for new levels of efficiency and scale. The next generation of financial infrastructures will need to reach more customers, more markets, with lower latency and better economics. Lucera provides exchange grade infrastructure solutions, which scale as your business grows.

For more information, please visit https://Lucera.com

3. Project participants and responsibilities

The following firms participated in the project:

- McObject
- Lucera
- STAC

The Project Participants had the following responsibilities:

- McObject implemented the STAC-M3 Clients and Operations using the STAC-M3 Benchmark specifications.
- Lucera provided the host and storage for the SUT.
- McObject sponsored the Audit.
- STAC conducted the STAC-M3 Benchmark Audit, which included creating the database, validating the database; inspecting the STAC Pack source-code; validating the Operation results; executing the tests, and documenting the results.

4. Contacts

- McObject: Chris Mureen, COO, chris.mureen@mcobject.com, +1 425 888 8505 x211
- Gavan Corr, Dir. of Sales, gcorr@lucera.com +1 (888) 979-0253

5. Results status

- These benchmark specifications were developed by the STAC-M3 Working Group of the STAC Benchmark Council. Benchmarks with a “v1” or higher have been approved by the full Council. Those with a “ß” designation have been proposed by the STAC-M3 Working Group but have not yet been approved.
• These test results were audited by STAC or a STAC-certified third party, as indicated in the Responsibilities section above. As such, they are official results. For details, see www.STACresearch.com/reporting.
• The vendors attest that they did not modify the SUT during the Audit.

6. Specifications
This project followed the Antuco suite of STAC-M3 Benchmark specifications. Qualified members of the STAC Benchmark Council can access these specifications at www.STACresearch.com/m3 and download the programs used in these benchmarks in order to run the same tests on systems in the privacy of their own labs.

7. Limitations
• As discussed in Section 1, this suite of STAC-M3 Benchmarks was designed to test operations on a limited amount of purely historical data. Tests involving much larger amounts of historical data are available in the Kanaga suite of STAC-M3 Benchmark specifications.
• As discussed in Section 1, the dataset used in this version of STAC-M3 is synthetic. The algorithm to generate the dataset creates random values for prices and sizes that can vary widely from tick to tick. In the real world, by contrast, there is significant correlation of successive prices (i.e., large differences from tick to tick are relatively rare). Compression algorithms often take advantage of this fact, such as by focusing on deltas between successive values. Hence, the storage efficiency of a SUT may be higher when working with real data than with the synthetic dataset of this version of STAC-M3.

8. Stack under test
Detailed configuration information is available to qualified members of the STAC Benchmark Council at: www.STACresearch.com/XTR141111.

9. Vendor Commentary
None.
About STAC

STAC® provides technology research and testing tools that are based upon community-source standards. STAC facilitates the STAC Benchmark™ Council (www.STACresearch.com/council), an organization of leading financial institutions and technology vendors that specifies standard ways to assess technologies used in finance. The Council is active in an expanding range of low-latency, big-compute, and big-data workloads.

STAC helps end-user firms relate the performance of new technologies to that of their existing systems by supplying them with STAC Benchmark reports as well as standards-based STAC Test Harnesses™ for rapid execution of STAC Benchmarks in their own labs. User firms do not disclose their results. Some STAC Benchmark results from vendor-driven projects are made available to the public, while those in the STAC Vault™ are reserved for qualified members of the Council (see www.STACresearch.com/vault).

To be notified when new STAC Reports™ become available, please sign up for free at www.STACresearch.com.