Embedded Database Performance Report

Actian Zen more than 100x faster than SQLite

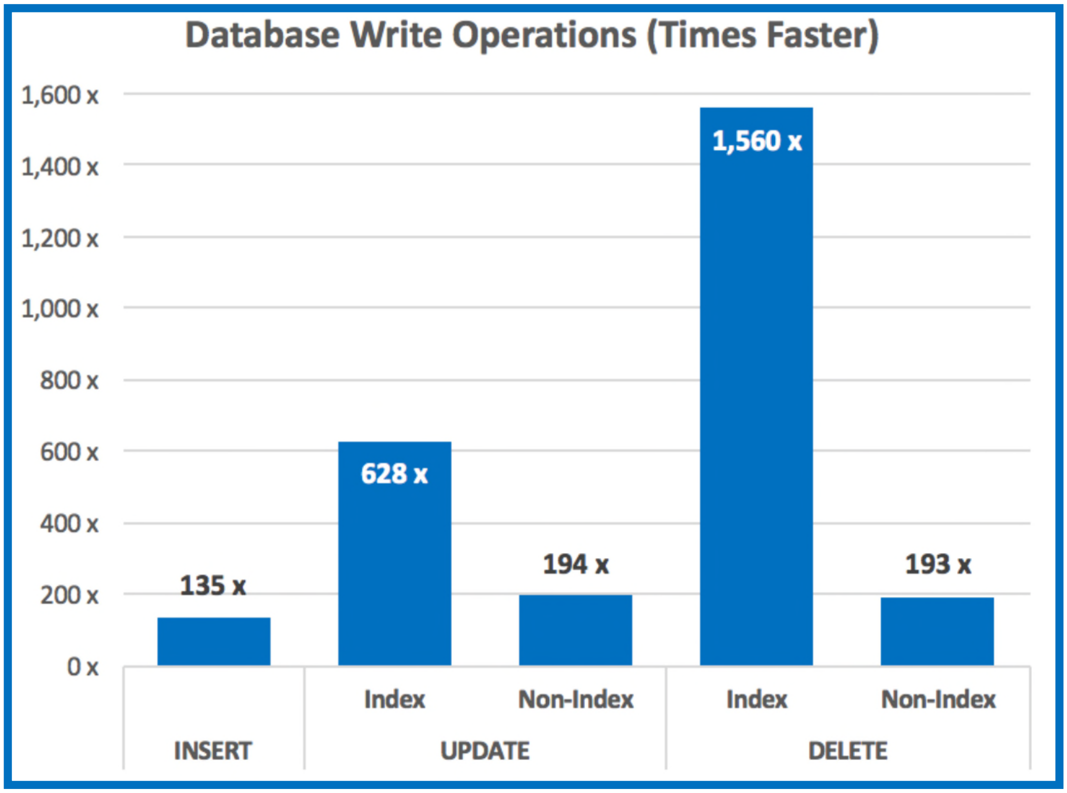
MCG Global Services Benchmark Results

August 2018

actian™

Key insights

- This benchmark compared Actian Zen Core and SQLite, both running on a Raspberry Pi 3, an ARM-based mini-SBC (single board computer)
- Actian Zen Core outperformed SQLite for Indexed and Non-Indexed data management by:
  - more than 100x on inserts
  - up to 1,500x on deletes,
  - over 600x on updates
- Actian Zen Core was faster across the board particularly in write speed—the area where it tends to really matter in embedded Edge applications.



Checkout the Actian Zen performance advantage today!

Visit https://www.actian.com/zen

Embedded Database Performance Benchmark

Product Profile and Evaluation:
Actian Zen Core and SQLite

By William McKnight and Jake Dolezal
McKnight Consulting Group
August 2018

Executive Overview

Embedded databases are built into software, transparent to the application's end user and require little or no ongoing maintenance. Embedded databases are growing in popularity with the rise of internet of things (IoT) giving innumerable devices robust capabilities via their own local database management system (DBMS). Developers can create sophisticated applications right on the IoT or remote device. For these uses, the embedded architecture is preferred over client-server approaches which rely on database servers accessed by client applications via interfaces. Today, to fully harness data to gain a competitive advantage, embedded databases need a high level of performance to provide real-time processing at scale.

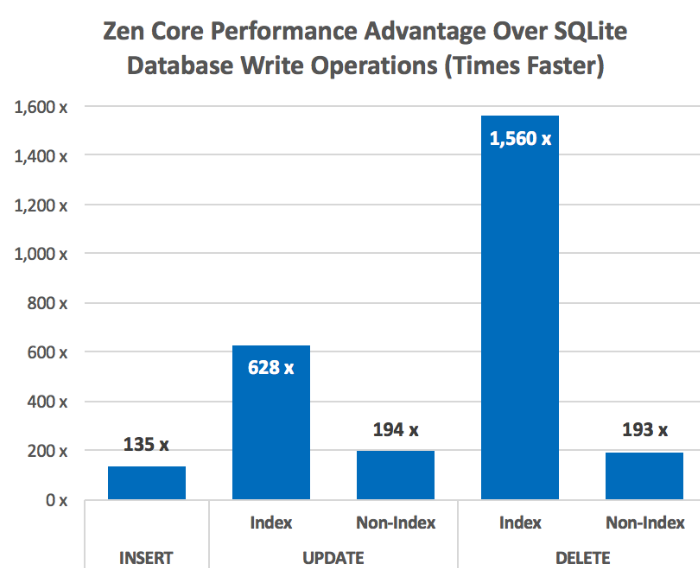
To quantify embedded database performance, we conducted this benchmark study, which focuses on the performance of IoT-enabled, application-ready, relationally-based, embedded database solutions Actian Zen Core and SQLite. The intent of the benchmark's design was to represent a set of basic database transactions that an organization developing edge applications might encounter.

The test methodology was based on and largely followed the Benchmark of Embedded Databases on .NET conducted in 2017 by Christophe Diericx; however, our own benchmark harness was developed. We conducted the benchmark with Zen Core and SQLite installed on the same Raspberry Pi 3 device. In our experience, performance is a very important aspect of an embedded database selection, but it is only one aspect and many factors should be considered.

Overall, the benchmark results were insightful in revealing the query execution performance of Actian Zen Core and SQLite revealing some of the differentiators in the two products.

Actian Zen Core was faster across the board particularly in write speed—the area where it tends to really matter in embedded applications. Actian Zen Core outperformed SQLite by more than 100x on inserts, 1,500x on deletes, and over 600x on updates.

Actian Zen is a mature platform for embedded database applications with over 30 years of engineering and development behind it. Features that contributed to its extremely fast performance include, but are not limited to, the Btrieve API and Turbo Write Accelerator.



Embedded Database Selection

Organizations that utilize IoT and other application-laden smart devices rely on embedded database platforms to process edge data at high speed and bring it in with consistency to harmonize an ecosystem of activity. Volumes for data that can be utilized at the edge is rapidly expanding—placing significant performance demands on embedded architectures. Thus, a key differentiator is the depth by which a database maintains performance to scale with simple queries representative of real world use cases of embedded databases.

While performance is very important, it is not the only consideration.

Both SQLite and Actian Zen were designed to “set it and forget it,” with little-to-no ongoing database administration. However, Actian Zen was engineered purposefully to pare down an enterprise database platform to be embedded within OEM environments. SQLite was only designed as a step up from standard file systems. Therefore, Actian Zen has features that SQLite does not—including auto-reconnect networking, automated defragmentation, multi-user support, and concurrent write capabilities.

Both platforms offer SQL support, Actian Zen SQL is 100% ANSI SQL compliant, while SQLite is not. Additionally, Zen exclusively offers the high performance Btrieve 2 API (which is tested in this benchmark.) The Btrieve 2 API also supports NoSQL and native development support for Java and C/C++ and SWIG for Python, Perl, and PHP—in addition to its SQL support.

While the subject of this benchmark is embedded applications, Actian Zen Core is part of the overall Zen family of Zen Edge, Zen Enterprise, and Zen Reporting Engine. When combined, this suite of products enables not only embedded applications, but client-server (with zero ETL) and cloud deployments as well.

Platform maturity is also a consideration. SQLite was initially released in 2000. Actian Zen was initially designed as Btrieve (and later PSQL) and has been in production with many multi-national organizations with over 30 years of engineering and enhancement.

This reports focuses on the performance of two embedded database options. It is important to get into the right embedded database early in the development cycle when the stakes are less critical. One is a specialty approach with enterprise software optimized for the embedded architecture, and the latter an open source, multi-purpose database platform.

Benchmark Setup

The benchmark was executed using the following setup, environment, standards, and configurations.

Data Preparation

An aim of the benchmark is to simulate a typical real-world scenario and use case for embedded databases. In our benchmark, we chose a simple data model for an application that stores peoples' contact information in the embedded database. The model consists of one table, Contacts, described by the following:

Contacts
id integer
lastname varchar(25)
firstname varchar(25)
street varchar(30)
city varchar(30)
state varchar(2)
zip varchar(10)
country varchar(20)
phone varchar(13)

The data used in the benchmark was generated randomly in real time by the Python script during the benchmark execution. The columns city, state, and zip were used as selection criteria in the Select, Update, and Delete tests (described below). Therefore, a particular value was randomly seeded into this column during data generation to ensure there would be enough instances of that value to achieve the row counts required during the Select, Update, and Delete tests.

Configuration

Our benchmark included two different embedded RDBMS—Actian Zen Core and SQLite—installed on the same Raspberry Pi 3 single board computer with a 64-bit quad core processor.

Embedded RDBMS	Actian Zen Core	SQLite
Version	13.20.023	3.24.0

Hardware	Raspberry Pi 3+
Processor	1x Broadcom BCM2837B0 SoC 1.4 GHz 64-bit quad-core ARM Cortex-A53 (512 KB shared L2 cache)
RAM	1 GB
OS	Raspbian GNU/Linux 9.4 (stretch)

The Raspberry Pi 3 (shown) was deployed on a local area network and connected to via secure shell.

Test Use Cases

As aforementioned, the test methodology was based on and largely followed the Benchmark of Embedded Databases on .NET conducted in 2017 by Christophe Dierickx[1]. The test involves simple uses cases of the most basic RDBMS operations: open and closing connections and selecting, updating, and deleting rows based on indexed and non-indexed columns.

We considered other benchmark frameworks, such as the Transaction Performance Council (TPC). However, their test use cases were too complex and not very applicable to IoT and device applications. Most IoT devices and other applications will not require sophisticated RDBMS operations like multiple JOINs or subqueries. Therefore we opted for tests that would demonstrate raw performance that could be found in most embedded database implementations.

Both RDBMS platforms support a robust set of SQL capabilities. For Actian Zen Core we used the Btrieve API, rather than SQL, to execute the database transactions in order to test its functionality and performance.

Use Case 1: Open and Close Connections in Rapid Succession

Some IoT device application will be developed to not keep a persistent connection to the database. Therefore, we were interested in seeing the performance of both RDBMS' ability to quickly connect and disconnect from the database in rapid succession.

For this, we had the benchmark test harness establish and close 250 connections to each RDBMS—one immediately after another.

Test 1	**Open and close 250 connections**

Use Case 2: Insert Performance

IoT devices and other applications will undoubtedly need excellent insert performance. This may the single most important metric for many use cases. For example, consider an IoT device is a sensor taking readings at regular intervals. In the case of real-time or rapid sensor readings, insert performance is critical.

Test 2	**Insert 25,000 rows**

[1] The Benchmark of Embedded Databases on .NET found SQLite to be the fastest overall the platforms tested.

At the beginning of the test, the database contains an empty Contacts table. The Insert test provides the test data for the remaining benchmarks.

Use Case 3: Select Performance

Certainly, we must consider both platforms' ability to retrieve data. Our test cases involve selecting bulk rows, rather than single rows via a unique identifier. The first variation of the test filters on an indexed column (state). The second test selects fewer rows, but filters on a column that does not have an index (zip).

Test 3a	**Select 10,000 rows on an indexed column**
Test 3b	**Select 5,000 rows on a non-indexed column**

Use Case 4: Update Performance

We also tested the performance of bulk row updates using the same selection test criteria as Test 3. Our test cases involve selecting bulk rows and updating a single column. The first variation of the test filters on an indexed column (state) and updates zip. The second test selects fewer rows, but filters on a column that does not have an index (zip) and updates state.

Test 4a	**Update 10,000 rows on an indexed column**
Test 4b	**Update 5,000 rows on a non-indexed column**

Use Case 5: Delete Performance

We also tested the performance of bulk row deletes—again, using the same selection test criteria as Test 3. Our test cases involve selecting bulk rows and deleting them. The first variation of the test filters on an indexed column (state) and deletes those rows. The second test selects fewer rows, but filters on a column that does not have an index (zip) and deletes the rows.

Test 5a	**Delete 10,000 rows on an indexed column**
Test 5b	**Delete 5,000 rows on a non-indexed column**

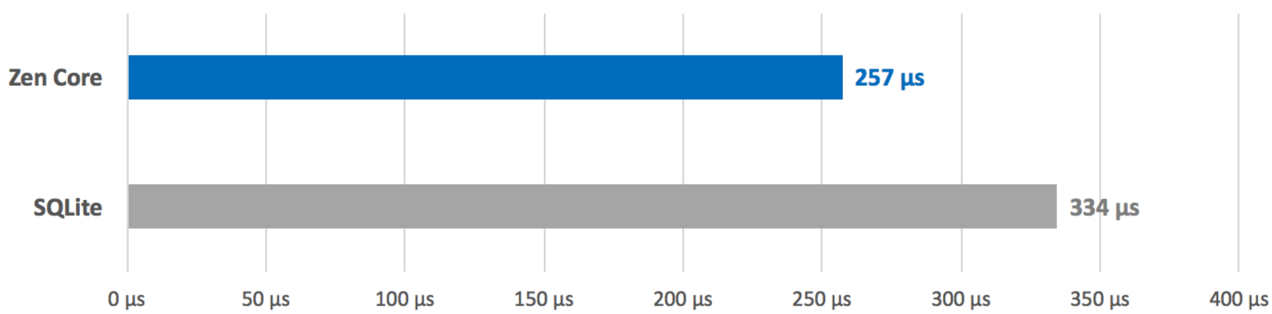
Benchmark Results

The following figures display the average time elapsed for each database transaction for both Actian Zen Core and SQLite. Each test was executed 5 times and the median value was used.

Test 1: Open and close 250 connections

Below are the average times (in microseconds) it took to open and close a connection to the Actian Zen Core and SQLite databases.

1. Open/Close 250 Connections - Average Time per Transaction (µs)



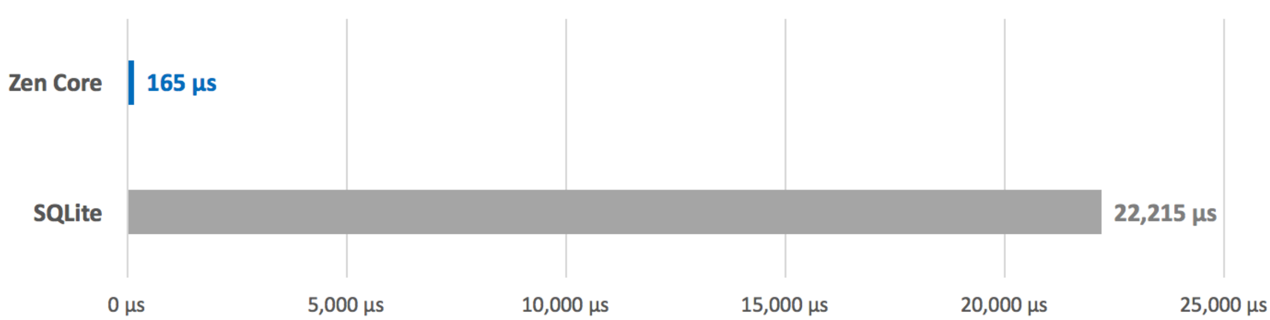
**This graph shows elapsed time. A shorter bar indicates a faster response time.*

After 250 attempts, Actian Zen's average time to open and close a connection took 23% less time than SQLite.

Test 2: Insert 25,000 rows

Below are the average times (in microseconds) it took to insert a complete row of randomly-generated data into the Contacts table on the Actian Zen Core and SQLite databases.

2. Insert 25,000 Rows - Average Time per Transaction (µs)

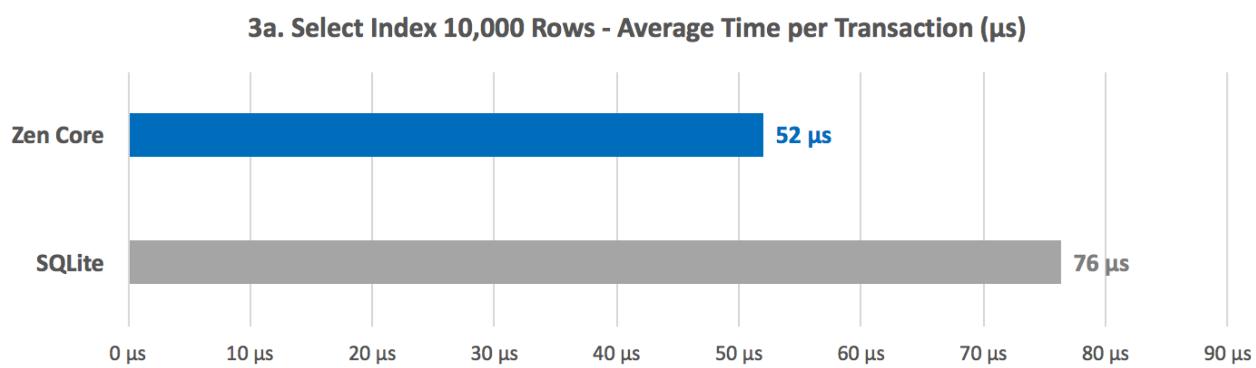


**This graph shows elapsed time. A shorter bar indicates a faster response time.*

This test revealed the first major performance differentiator. Actian Zen's average time to insert a single row (taking the average of all 25,000 inserts) was a 135 times faster than SQLite inserts.

Test 3a: Select 10,000 rows on an indexed column

Below are the average times per row (in microseconds) it took to bulk select records from the Contacts table applying a filter on an indexed column for both the Actian Zen Core and SQLite databases.

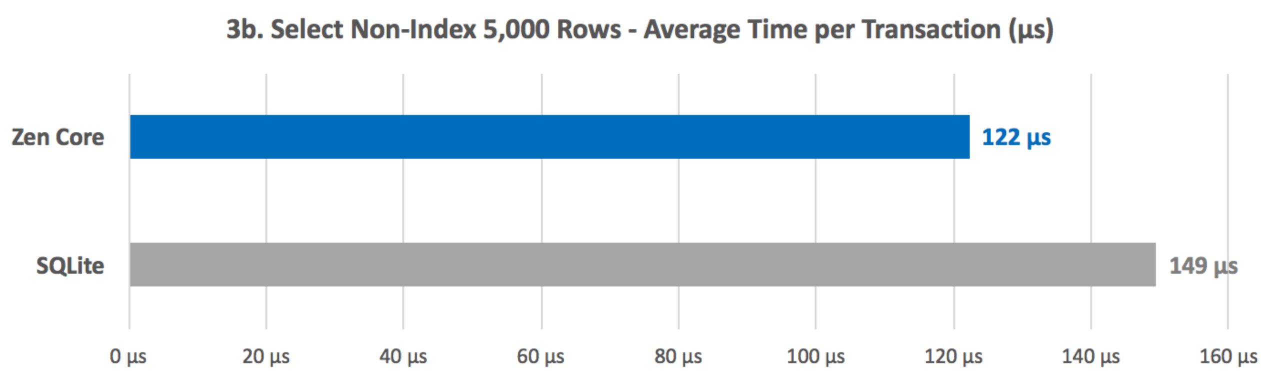


**This graph shows elapsed time. A shorter bar indicates a faster response time.*

Both platforms responded very quickly. Actian Zen's fetch rate per row (taking the average of all 10,000 rows) took 32% less time than SQLite.

Test 3b: Select 5,000 rows on a non-indexed column

Below are the average times per row (in microseconds) it took to bulk select records from the Contacts table applying a filter on a non-indexed column for both the Actian Zen Core and SQLite databases.

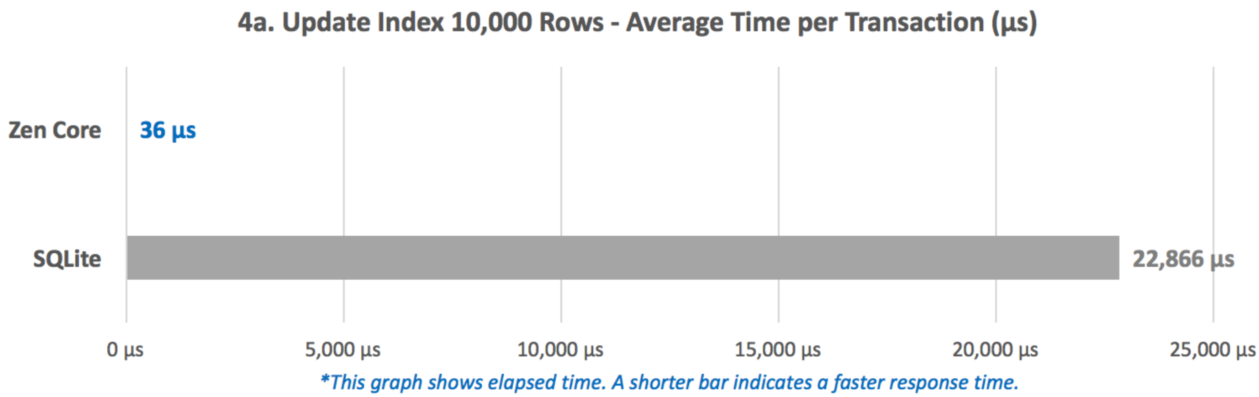


**This graph shows elapsed time. A shorter bar indicates a faster response time.*

Again both platforms responded very quickly. Actian Zen's fetch rate per row (taking the average of all 5,000 rows) took 18% less time than SQLite.

Test 4a: Update 10,000 rows on an indexed column

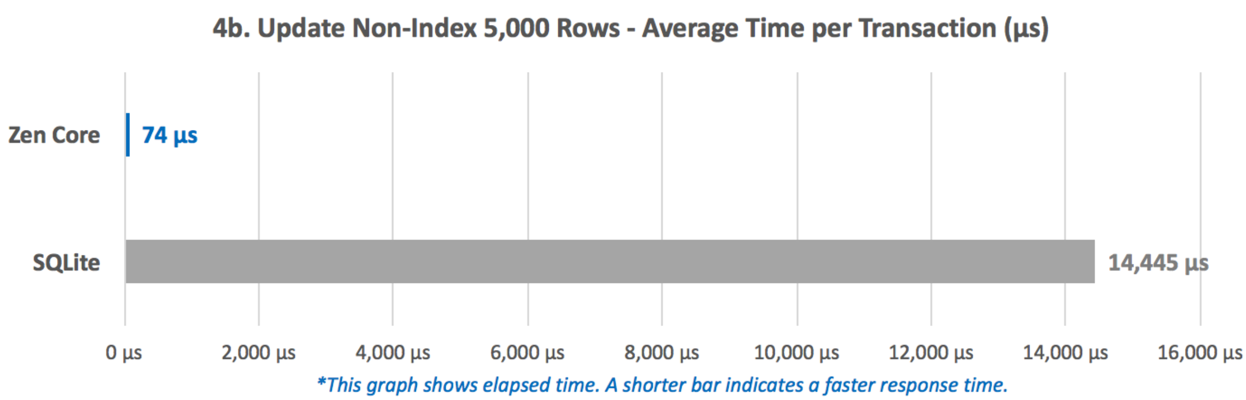
Below are the average times (in microseconds) it took to update a single column in the Contacts table applying a filter on an indexed column for both the Actian Zen Core and SQLite databases.



Actian Zen's average time was so fast; it barely registers on this graph. Its average time to update a single column (taking the average of all 10,000 updates) was an impressive 628 times faster than SQLite updates!

Test 4b: Update 5,000 rows on a non-indexed column

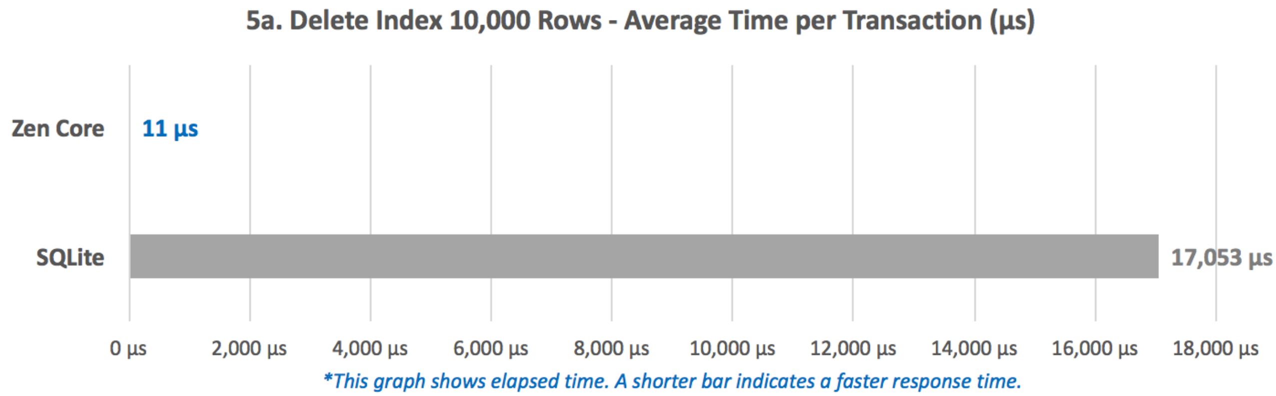
Below are the average times (in microseconds) it took to update a single column in the Contacts table applying a filter on a non-indexed column for both the Actian Zen Core and SQLite databases.



This test had similar results as test 4a. Actian Zen's average time to update a single column (taking the average of all 5,000 updates) was 194 times faster than SQLite updates using the same filter.

Test 5a: Delete 10,000 rows on an indexed column

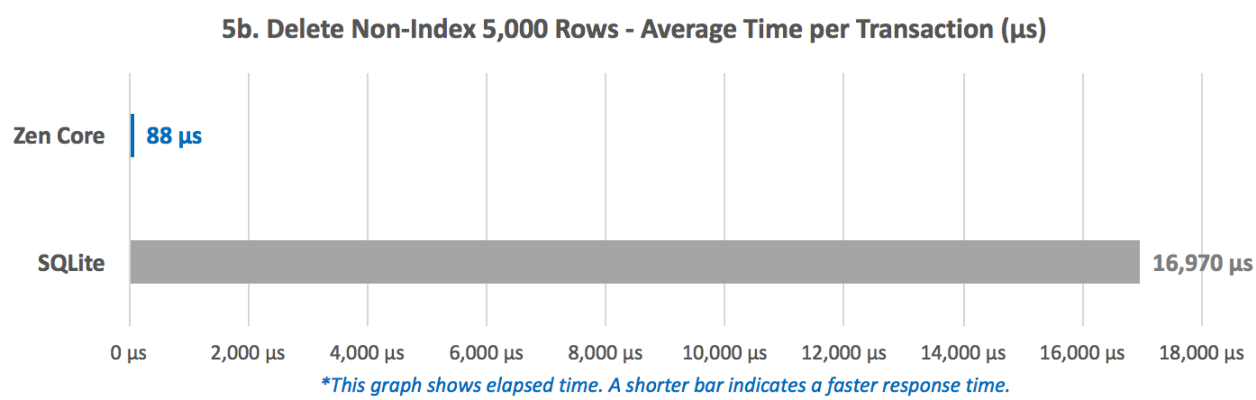
Below are the average times (in microseconds) it took to delete a row in the Contacts table applying a filter on an indexed column for both the Actian Zen Core and SQLite databases.



Actian Zen was very fast. Its average time to delete a tow (taking the average of all 10,000 deletes) was an overwhelming 1,560 times faster than SQLite deletes!

Test 5b: Delete 5,000 rows on a non-indexed column

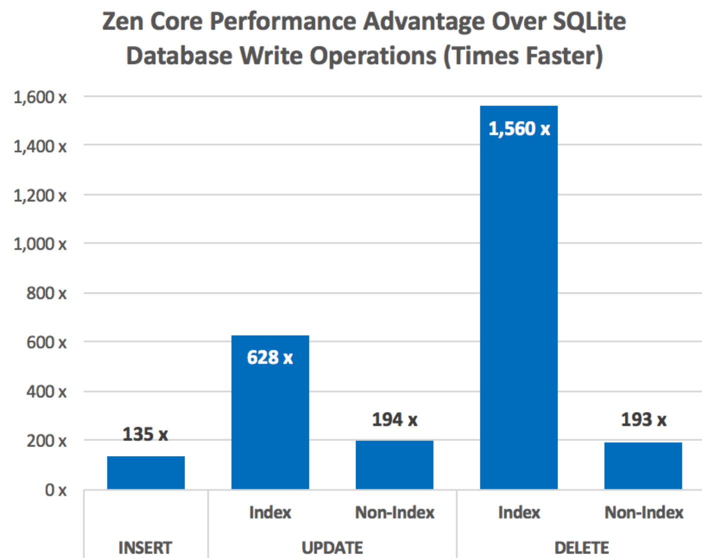
Below are the average times (in microseconds) it took to delete a row in the Contacts table applying a filter on a non-indexed column for both the Actian Zen Core and SQLite databases.



Deleting rows on a non-indexed column produced results consistent with before. Actian Zen's average time to delete a row (taking the average of all 5,000 deletes) was 193 times faster than SQLite updates using the same filter.

Conclusion

Actian Zen Core was faster across the board particularly in write speed—the area where it tends to really matter in embedded applications. Actian Zen Core outperformed SQLite by more than 100x on inserts, 1,500x on deletes, and over 600x on updates.



Moreover, Zen Core also took between 18% and 32% less time to open and close rapid connections and select data, with and without indices.

Actian Zen Core outperformed SQLite in all of the fundamental database operations. These tested operations underlie nearly all operations that occur on an embedded database for an IoT implementation, so it is unlikely more complex operations would have a different result.

Actian Zen is a mature platform for embedded database applications with over 30 years of engineering and development behind it. The Btrieve 2 API had clear performance advantages without the overhead of SQL-bound SQLite. Also, Zen's Turbo Write Accelerator could also shed light into its performance advantages. Since it costs much less to continue writing than to stop and restart, contiguous writes are significantly faster than non-contiguous writes. The Turbo Write Accelerator (TWA) pre-allocates open slots within the physical file so that multiple pages can be written as a single coalesced page—improving I/O performance and reducing the overhead of interaction with the operating system.

The result of the application of the methodology to the architecture, both explained herein and replicable, show a marked, and sometimes astonishing, performance advantage to Actian Zen Core. This is especially true in the important write operations insert, update and delete.

Overall, Actian Zen Core is an excellent choice for IoT companies needing high performance and a scalable embedded database.

About McKnight Consulting Group

William McKnight is President of McKnight Consulting Group (MCG) (http://www.mcknightcg.com). He is an internationally recognized authority in information management. His consulting work has included many of the Global 2000 and numerous midmarket companies. His teams have won several best practice competitions for their implementations and many of his clients have gone public with their success stories. His strategies form the information management plan for leading companies in various industries.

Jake Dolezal has two decades of experience in the Information Management field with expertise in business intelligence, analytics, data warehousing, statistics, data modeling and integration, data visualization, master data management, and data quality. Jake has experience across a broad array of industries, including: healthcare, education, government, manufacturing, engineering, hospitality, and gaming. He has a doctorate in information management from Syracuse University.

MCG services span strategy, implementation, and training for turning information into the asset it needs to be for your organization. We strategize, design and deploy in the disciplines of Master Data Management, Big Data Strategy, Data Warehousing, Analytic Databases and Business Intelligence.

About Actian

Actian, the hybrid data management, analytics and integration company, delivers data as a competitive advantage to thousands of customers worldwide. Through the deployment of innovative hybrid data technologies and solutions Actian ensures that business critical systems can transact and integrate at their very best – on premise, in the cloud or both. For more information about Actian Vector and the entire Actian portfolio of hybrid data management, analytics and integration solutions on-premise or in the cloud.

Find out more about Actian Vector for single servers and for Hadoop clusters, or get links to downloads for on-premise deployment or cloud instances.