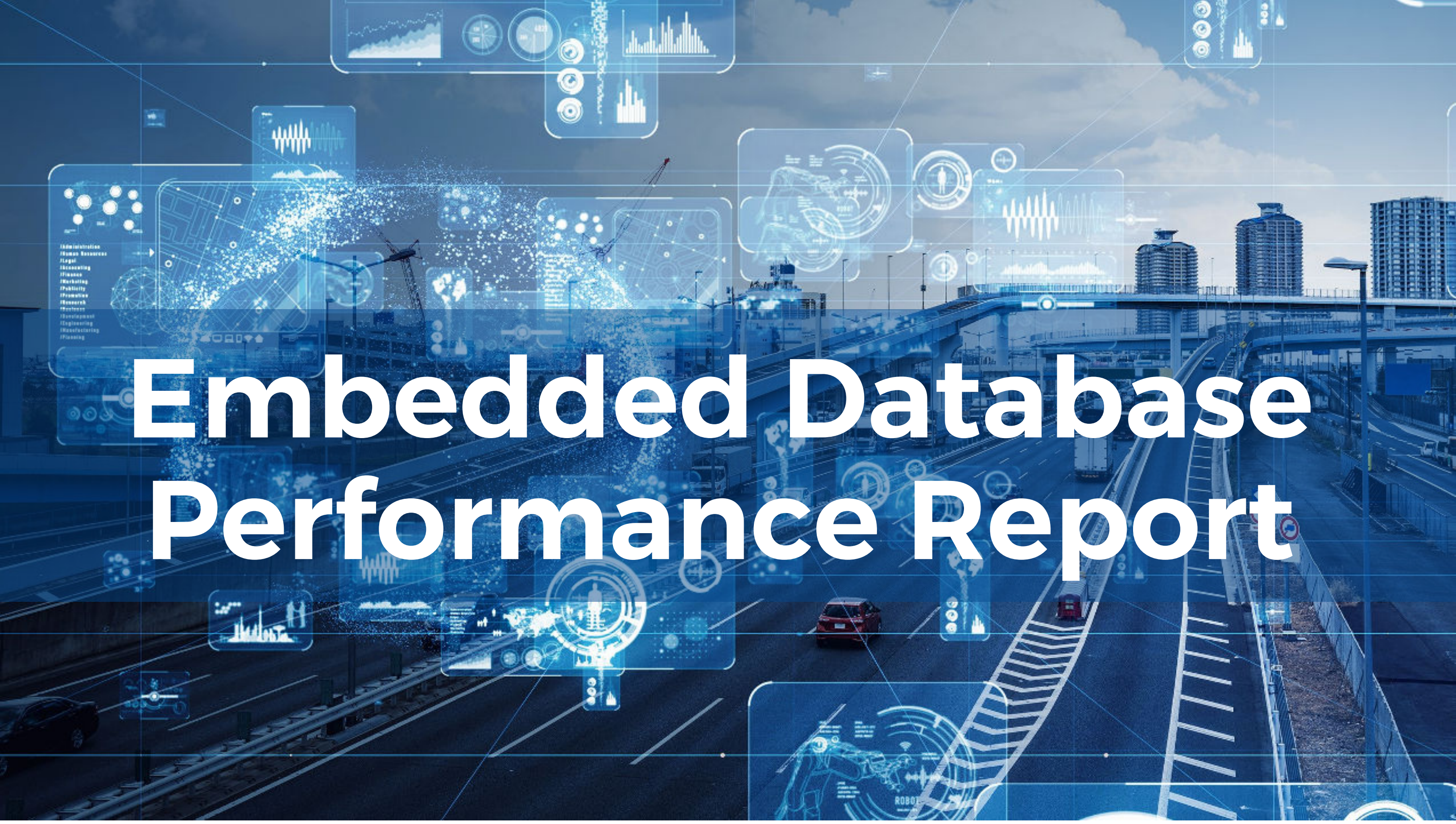


Embedded Database Performance Report

Actian Zen between 2x and 6x faster than MongoDB for Mobile and IoT data management

MCG Global Services Benchmark Results

August 2019

ACTIAN™

Key insights

- This benchmark did a head-to-head comparison of:
  - Actian Zen Enterprise and MongoDB Server for contact information records, both installed on a Lenovo ThinkPad X1 running Windows 10 using their native APIs (NoSQL);
  - Actian Zen Core and MongoDB Mobile for extension to mobile clients, both installed on a Nokia Mobile phone running Android 7 on ARM using their native APIs (NoSQL)
  - Synchronization of Zen Enterprise and Zen Core versus same for MongoDB/MongoDB Mobile

- Actian Zen outperformed MongoDB and its Mobile frontend by a minimum of a factor of two or far more across:
  - Data management operations on Zen Enterprise and MongoDB by 6.3x on inserts, 2x and 8x on queries of documents indexed and non-indexed, respectively
  - Data management operations on Zen Core and MongoDB Mobile by 7x on inserts, 46x and 4x on updates of documents indexed and non-indexed, respectively
  - Zen synchronization was 2.1X faster for single document non-indexed inserts, 4.1X for updates, 5.3X deletes

Checkout the Actian Zen performance advantage today!

Visit https://www.actian.com/zen-embedded-database

NoSQL Database Performance Benchmark

Product Profile and Evaluation:
Actian Zen and MongoDB

By McKnight Consulting Group
July 2019

Sponsored by

Executive Overview

Web, mobile, and IoT applications have generated a new set of technology requirements. Database architecture needs to be far more agile than ever before with the ability to address a wide variety of data requirements and scale down to be embedded right on devices. New approaches to real-time data management must accommodate unprecedented levels of scale, speed, and data flexibility. Sometimes relational databases are unable to meet these new requirements, and application developers are therefore turning to NoSQL database technology. NoSQL use cases abound where the need for flexible schema or schema-less data would trip up conventional, relational databases.

NoSQL embedded databases are growing in ubiquity with the rise of mobile applications and internet of things (IoT) giving innumerable devices robust capabilities via their own local database management system (DBMS). Developers can create sophisticated data-rich serverless applications right on a 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, NoSQL databases need a high level of performance to provide real-time processing at scale.

To quantify embedded NoSQL database performance, we conducted this benchmark study, which focuses on the performance of NoSQL embedded database solutions Actian Zen and MongoDB. The intent of the benchmark's design was to represent a set of basic database transactions that an organization developing applications might encounter.

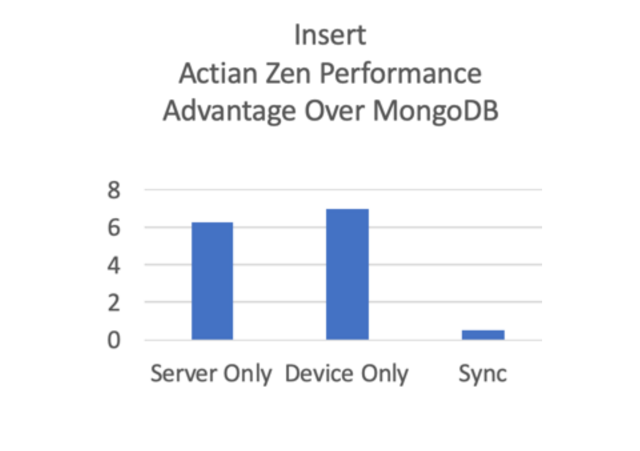
The test methodology was based on and largely followed the Benchmark of Embedded Databases on .NET conducted in 2017 by Christophe Diericx of relational database technologies; however, our own benchmark harness was developed and adapted to a NoSQL use case. We conducted the benchmark on Zen and MongoDB installed on the same laptop computer. In our experience, performance is a very important aspect of database selection, but it is only one aspect and many factors should be considered.

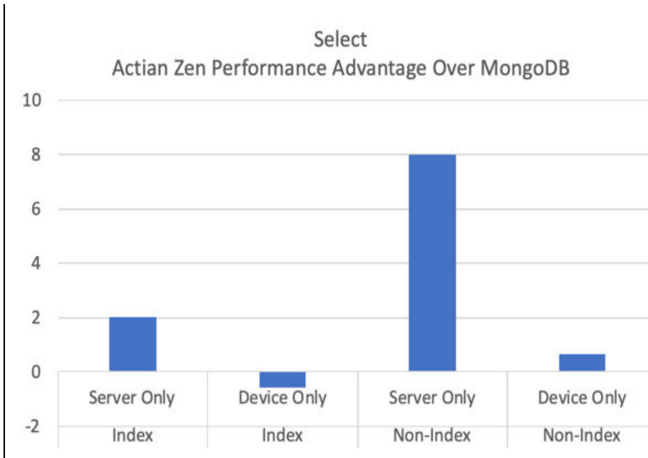
Overall, the benchmark results were insightful in revealing the query execution performance of Actian Zen and MongoDB revealing some of the differentiators in the two products.

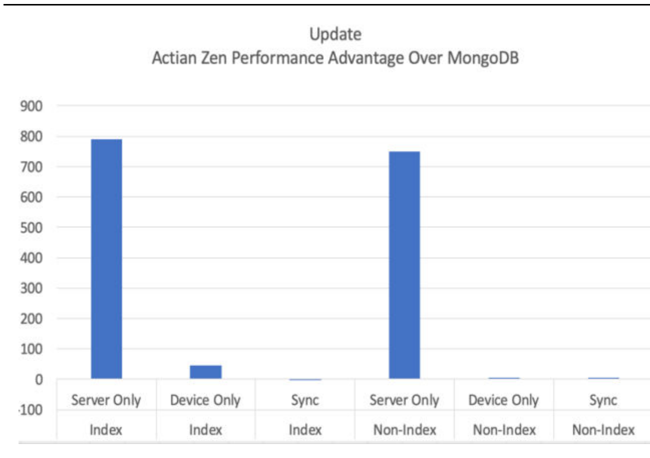
Actian Zen was faster across the board including the area where it tends to really matter in embedded databases—write speed. This is the essential performance metric for IoT data. As a server, Actian Zen outperformed MongoDB by 6.3x on inserts, 2x on queries of documents on an indexed key, 8x on queries of documents on a non-indexed key, 791x on updates of documents on an indexed key, 749x on updates of documents on a non-indexed key, 148x on deletes of documents on an indexed key and 110x on deletes of documents on a non-indexed key. With no synchronization on an Android device, Actian Zen Edge outperformed MongoDB Mobile by 7x on inserts, underperformed by 60% on queries of documents on an indexed key, 66% on queries of documents on a non-indexed key, 46x on updates of documents on an indexed key, 4x on updates of documents on a non-indexed key, 6.3x on deletes of documents on an indexed key and 4x on deletes of documents on a non-indexed key. Also on the same Android device with synchronization back to the server, Actian Zen outperformed MongoDB by 50% on inserts, by 4x on updates of documents on an indexed key, 5.4x on updates of documents on a non-indexed key, outperformed by 43% on deletes of documents on an indexed key and 2.3x on deletes of documents on a non-indexed key.

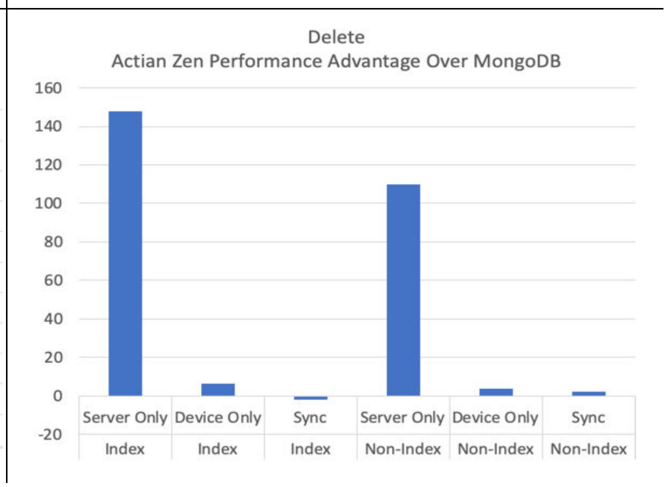
Actian Zen is a mature platform for embedded database applications with over 30 years of engineering and development behind it. Features that contributed to its extremely fast performance include, but are not limited to, the Btrieve API and Turbo Write Accelerator.

These charts represent the times faster for Actian Zen over MongoDB for the respective function on Server Only, Device Only and Sync (Synchronization) runs.









NoSQL Embedded Database Selection

Organizations that utilize application-laden 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 away from the server room is rapidly expanding—placing significant performance demands on local compute 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 databases—SQL and NoSQL alike.

While performance is very important, it is not the only consideration. Developers choosing NoSQL must consider data access, scalability, and availability.

Both MongoDB 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 NoSQL platform to be embedded within OEM environments. Therefore, Actian Zen has features that MongoDB does not—including auto-reconnect networking, automated defragmentation, multi-user support, and concurrent write capabilities.

Actian Zen is natively NoSQL and is flexible enough to be document-based or a key-value store. MongoDB is a BSON document-oriented database. Additionally, Zen exclusively offers the high performance Btrieve 2 API (which is tested in this benchmark.) The Btrieve 2 API supports NoSQL and native development support for Java and C/C++ based devices and SWIG for Python, Perl, and PHP—in addition to its SQL support. MongoDB is exclusively NoSQL, and only offers software development kits for mobile devices, such as iOS, Android, and .NET.

While the subject of this benchmark is embedded applications, Actian Zen edge is part of the overall Zen family of Zen Core, 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.

In a client-server configuration, Actian Zen comes with the capability to automatically synchronize in real time between Zen Core or Edge on a remote device and Zen Enterprise on a server—without ETL. This capability is critical for today’s needs and uses, because the potential number of mobile devices could easily number in the thousands, and all that information may need to funnel into a core database on a server. Actian has real time synchronization capability of Actian Zen Edge or Core to Zen Enterprise via the Btrieve API without an intermediary, which can allow you to achieve scale with simplicity.

Platform maturity is also a consideration. MongoDB was initially released in 2009. 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 NoSQL 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 NoSQL embedded databases. In our benchmark, we chose a simple “schema” for an application that stores peoples’ contact information in the embedded database. The model consists of multiple documents that look similar to the following:

```
{
	"contact": {
		"id": 1,
		"lastname": "Rogers",
		"firstname": "Fred",
		"address": "381 Willinghelm Dr",
		"city": "Pittsburgh",
		"state": "PA",
		"zip": "15106",
		"country": "USA",
		"phone": "412-875-0921"
	}
}
```

The data used in the benchmark was generated randomly in real time by the application during the benchmark execution. The keys 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 key during data generation to ensure there would be enough instances of that value to achieve the document counts required during the Select, Update, and Delete tests.

Configuration

Our benchmark included two different embedded RDBMS—Actian Zen and MongoDB—installed on the same laptop computer.

NoSQL DBMS

Embedded RDBMS	Actian Zen	MongoDB
Version	13.31	4.0.10

Server

Hardware	Lenovo ThinkPad X1 Carbon G6 20BS006UUS x64-based PC
Processor / RAM	2x Intel Core i7-5600U @ 2.60GHz / 8 GB RAM
OS	Microsoft Windows 10 Enterprise 10.0.18362

Android Device

Hardware	Nokia 2 TA-1035 DS
Processor	1.3 GHz 64-bit quad-core ARM Cortex A7
RAM	1 GB (8 GB Storage)
OS	Android 7.1.1 Nougat

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 Diericx. The test involves simple uses cases of the most basic database CRUD operations: selecting, updating, and deleting rows based on indexed and non-indexed columns.

We considered other benchmark frameworks, such as the Transaction Performance Council (TPC). While, their test use cases have been applied to NoSQL technologies in the past, they are not very applicable to typical mobile device applications. Most IoT devices and mobile applications will not require the sophisticated operations demonstrated by those benchmark frameworks. Therefore, we opted for tests that would demonstrate raw performance that could be found in most embedded database implementations.

Both platforms support a robust set of NoSQL capabilities. For both Actian Zen and MongoDB we used the native APIs to execute the database transactions in order to test its functionality and performance.

Use Case 1: Open and Close Connections in Rapid Succession

NOTE: We did not do this run for the laptop, since it is standard practice for developers to open a database connection and leave it open while the app is running.

Use Case 2: Insert Performance

Devices and their applications will undoubtedly need excellent insert performance. This may the single most important metric for many use cases. For example, consider an application recording measurements at regular intervals. In the case of real-time data, insert performance is critical.

Test 2	Insert 25,000 documents

NOTE: At the beginning of the test, the database contains an empty database. The Insert test provided 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 documents, rather than single documents via a unique identifier. The first variation of the test filters on an indexed key (state). The second test selects fewer documents, but filters on a key that does not have an index (zip).

Test 3a	**Select 10,000 documents on an indexed key**
Test 3b	**Select 5,000 documents on a non-indexed key**

Use Case 4: Update Performance

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

Test 4a	**Update 10,000 documents on an indexed key**
Test 4b	**Update 5,000 documents on a non-indexed key**

Use Case 5: Delete Performance

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

Test 5a	**Delete 10,000 documents on an indexed key**
Test 5b	**Delete 5,000 documents on a non-indexed key**

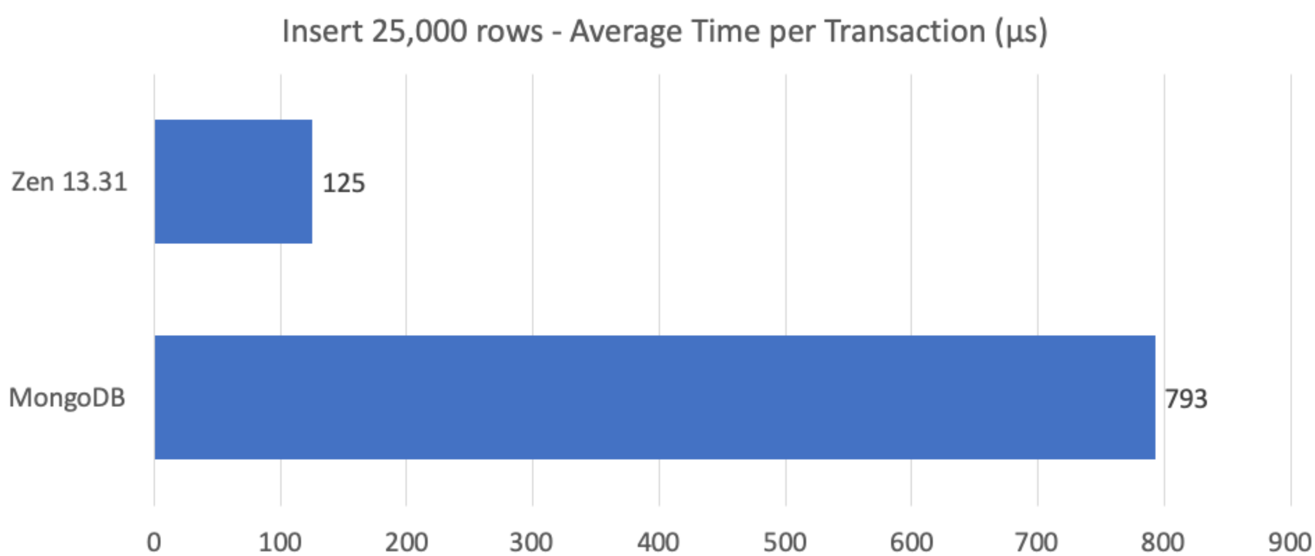
Benchmark Results

Zen 13.31 vs Mongo DB Server

The following figures display the average time elapsed for each database transaction for both Actian Zen and MongoDB. Each test was executed 5 times and the fastest value was used. Please note the values represent the average time per transaction, not for the entire test.

Test 2: Insert 25,000 documents

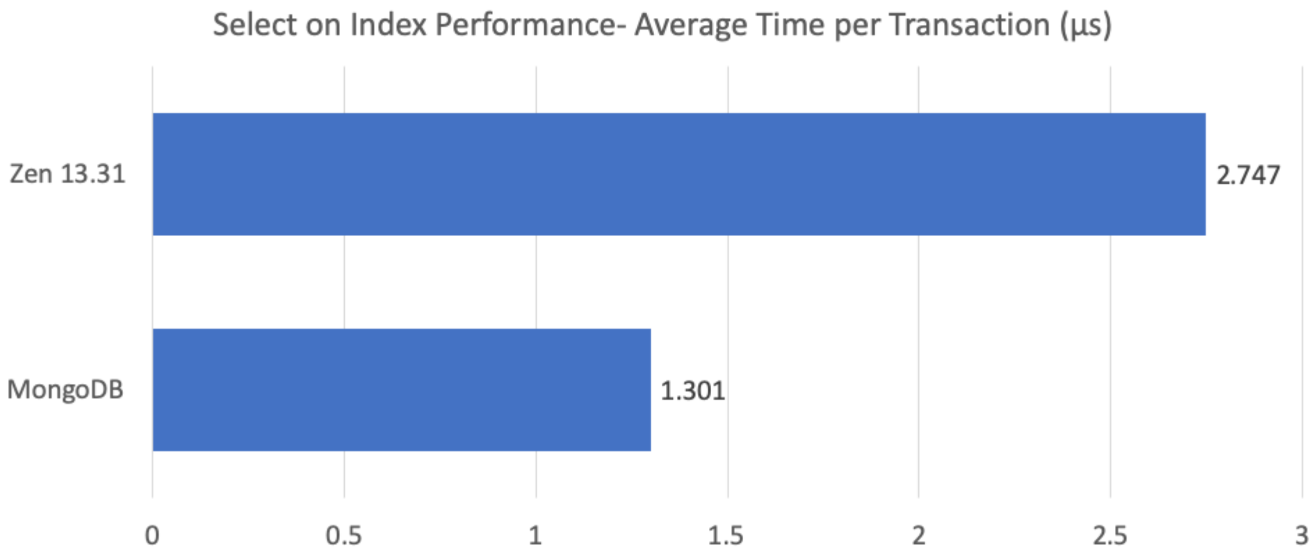
Below are the average times (in microseconds) it took to insert a complete document of randomly-generated data into the Contacts database on Actian Zen and MongoDB.



This test revealed the first major performance differentiator. Actian Zen's average time to insert a single document (taking the average of all 25,000 inserts) was 6.3 times faster than MongoDB inserts.

Test 3a: Select 10,000 documents on an indexed key

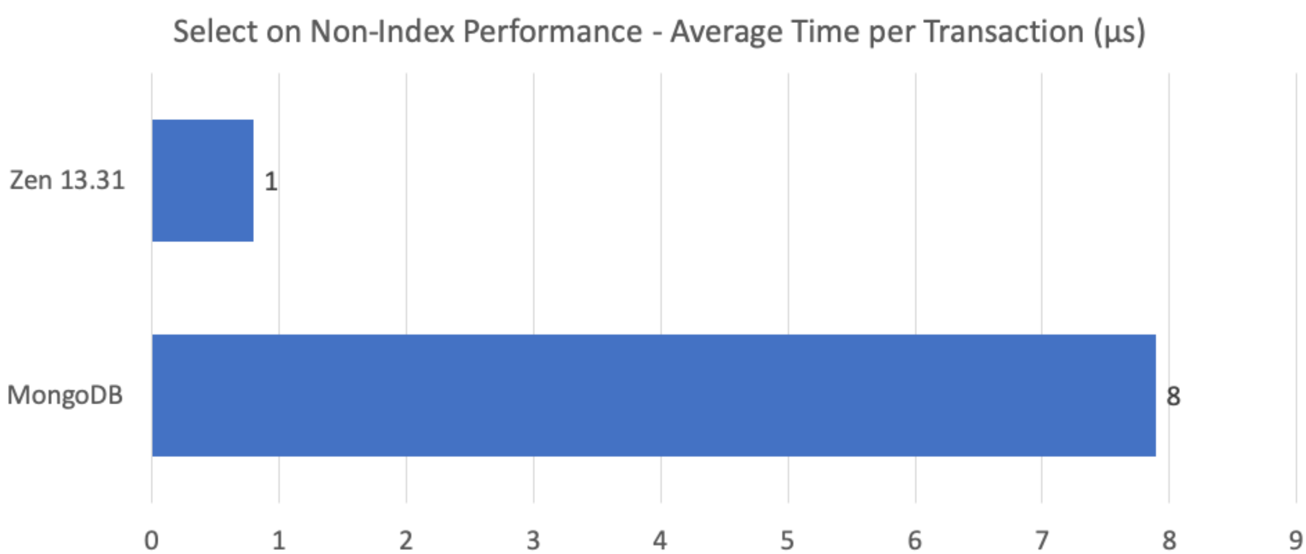
Below are the average times per document (in microseconds) it took to bulk select records from the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



MongoDB's fetch rate per document (taking the average of all 10,000 documents) was better than 2 times that of Actian Zen's.

Test 3b: Select 5,000 documents on a non-indexed key

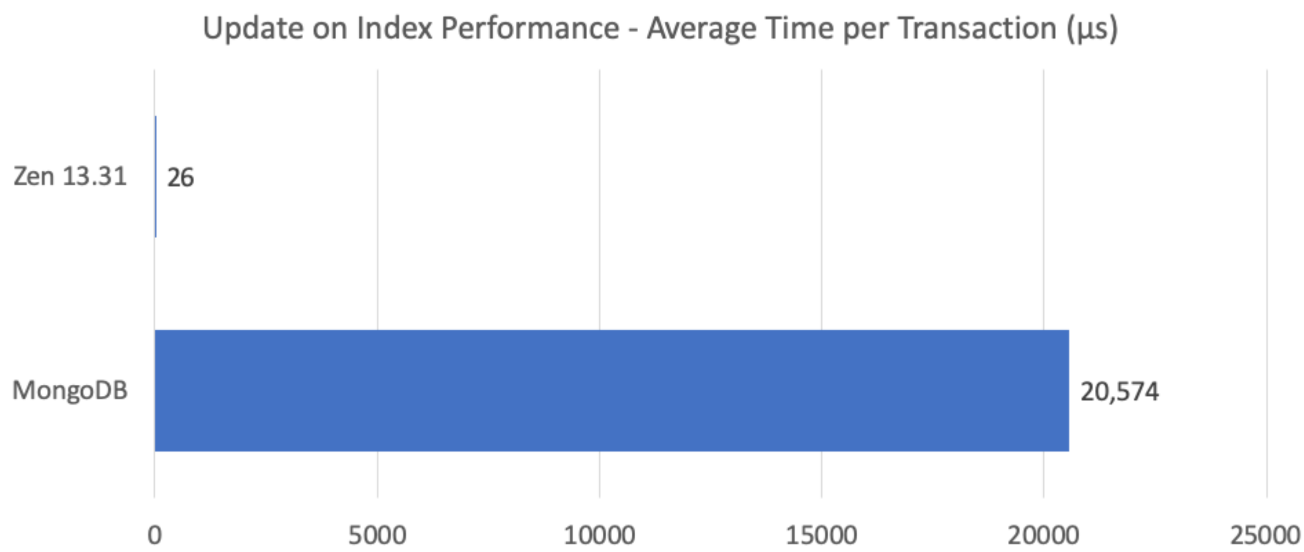
Below are the average times per document (in microseconds) it took to bulk select records from the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



Both platforms responded quickly. MongoDB's fetch rate per document (taking the average of all 10,000 documents) was 8 times that of Actian Zen's.

Test 4a: Update 10,000 documents on an indexed key

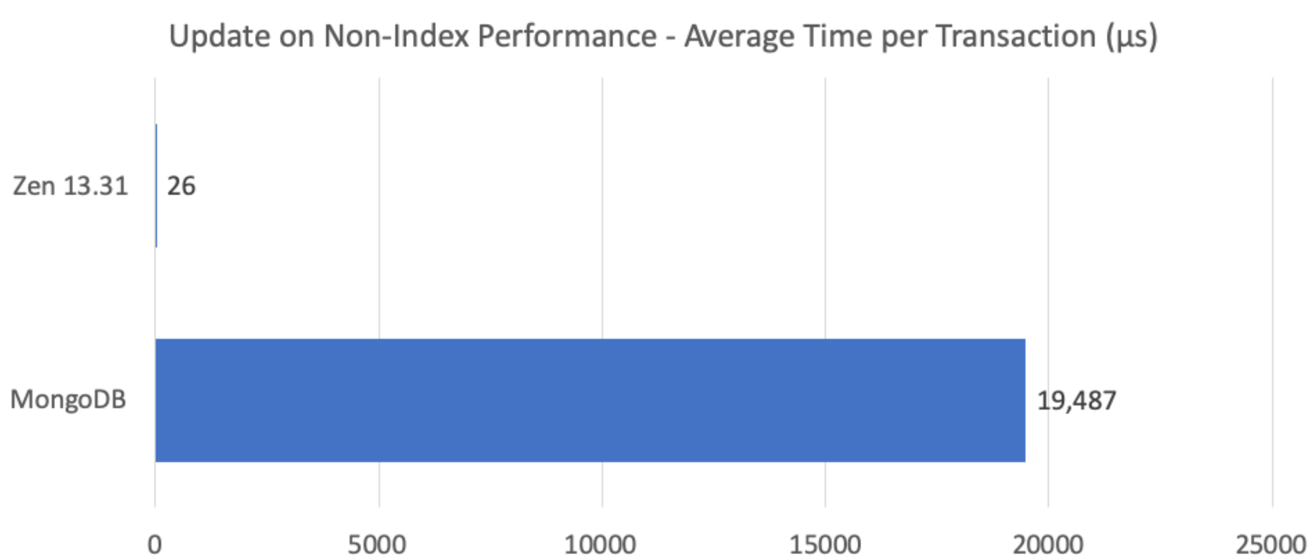
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



This one was not a close test. The average time to update a single key (taking the average of all 10,000 updates) was many times faster than MongoDB updates.

Test 4b: Update 5,000 documents on a non-indexed key

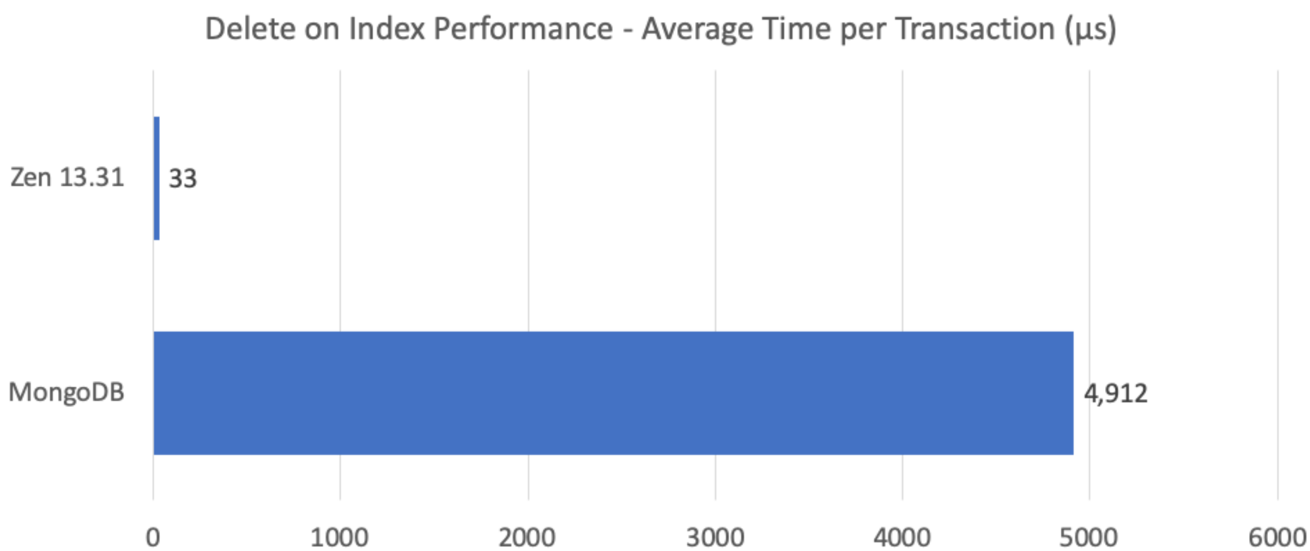
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



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

Test 5a: Delete 10,000 documents on an indexed key

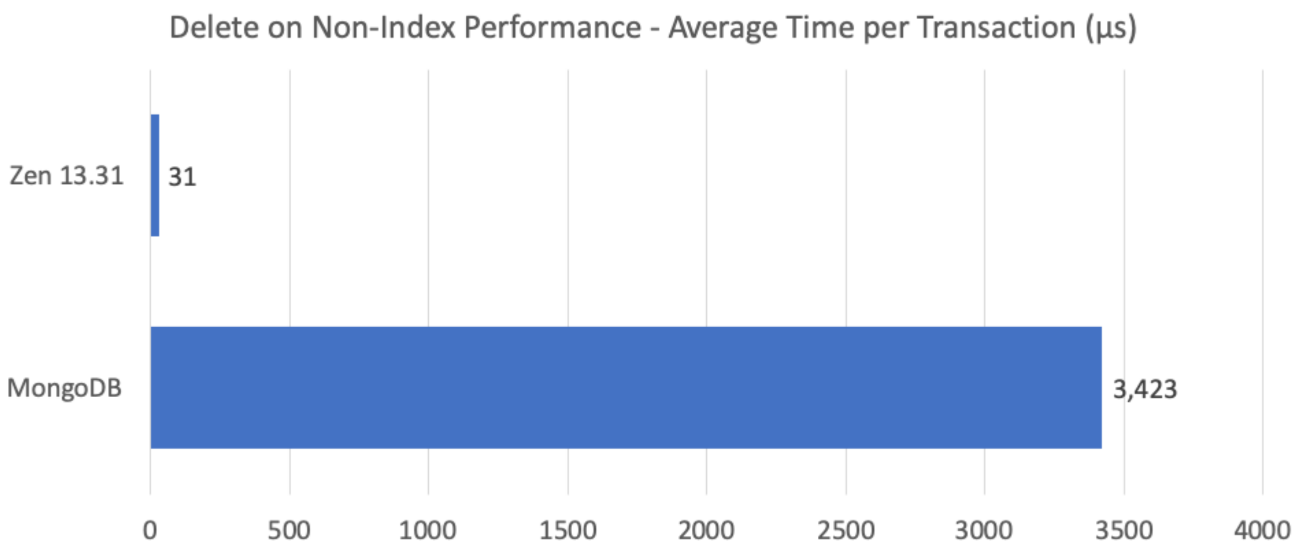
Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



The average time to delete a row (taking the average of all 10,000 deletes) for Actian Zen was many times faster than MongoDB deletes.

Test 5b: Delete 5,000 documents on a non-indexed key

Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



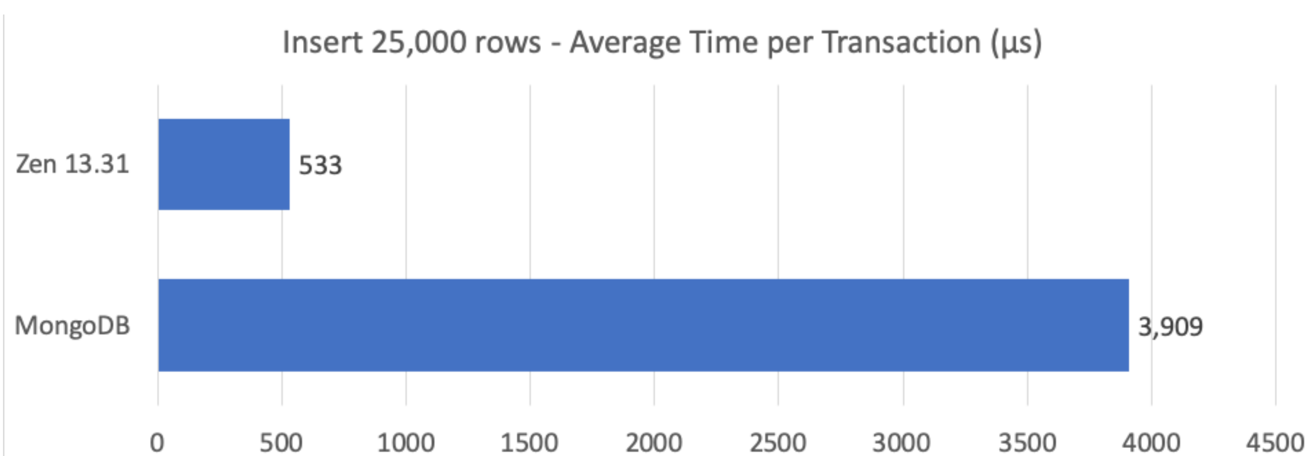
Deleting documents on a non-indexed key produced results consistent with before. Actian Zen's average time to delete a document (taking the average of all 5,000 deletes) was over 100 times faster than MongoDB updates using the same filter.

Android, No Sync

The following figures display the average time elapsed for each database transaction for both Actian Zen and MongoDB Mobile on Android with no synchronization to a server. Each test was executed 5 times and the fastest value was used. Please note the values represent the average time per transaction, not for the entire test.

Test 2: Insert 25,000 documents

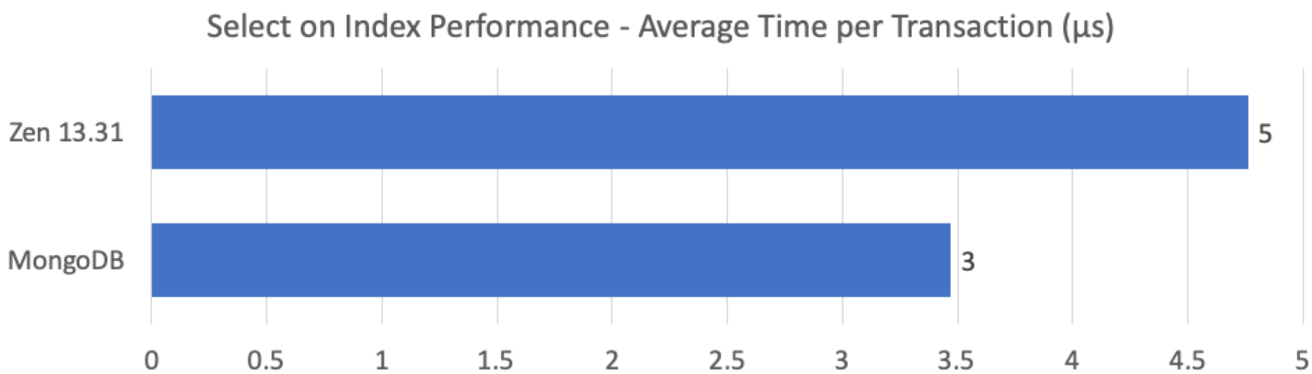
Below are the average times (in microseconds) it took to insert a complete document of randomly-generated data into the Contacts database on Actian Zen and MongoDB.



This test revealed the first major performance differentiator. Actian Zen's average time to insert a single document (taking the average of all 25,000 inserts) was over 7 times faster than MongoDB inserts.

Test 3a: Select 10,000 documents on an indexed key

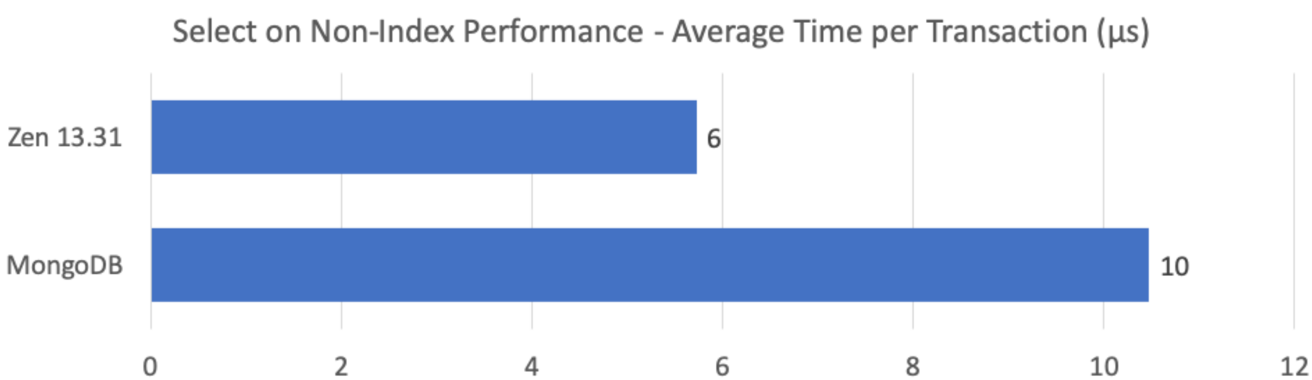
Below are the average times per document (in microseconds) it took to bulk select records from the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



MongoDB's fetch rate per document (taking the average of all 10,000 documents) was 40% better than that of Actian Zen's.

Test 3b: Select 5,000 documents on a non-indexed key

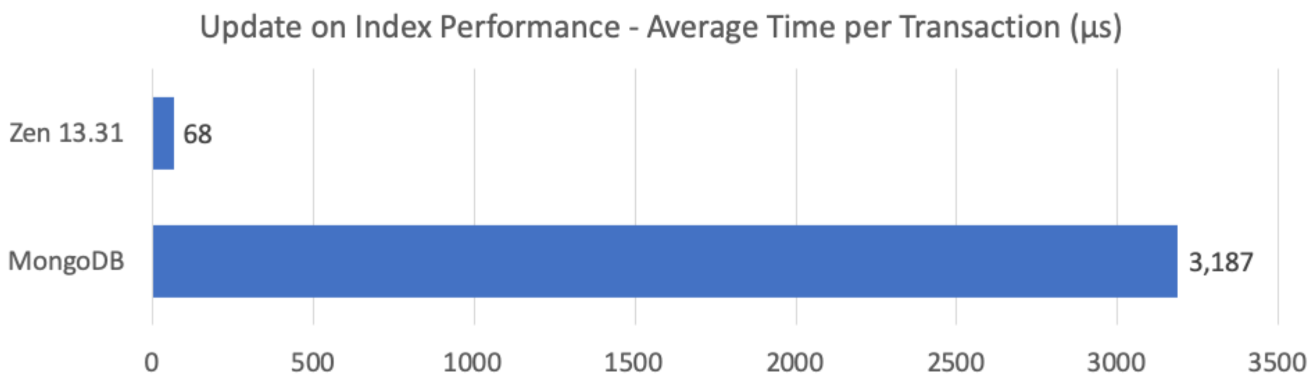
Below are the average times per document (in microseconds) it took to bulk select records from the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



Both platforms responded quickly. MongoDB's fetch rate per document (taking the average of all 10,000 documents) was 66% more than that of Actian Zen's.

Test 4a: Update 10,000 documents on an indexed key

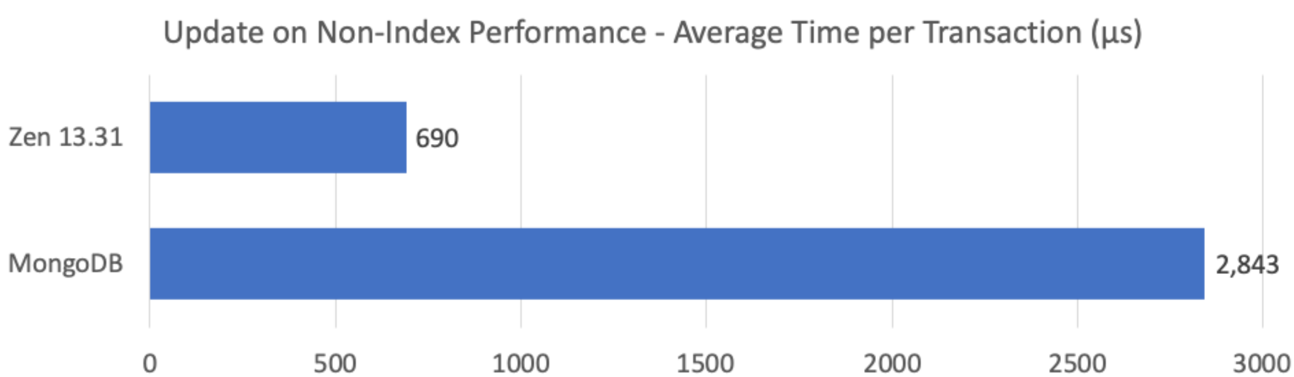
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



This one was not a close test. The average time to update a single key (taking the average of all 10,000 updates) was many times faster than MongoDB updates.

Test 4b: Update 5,000 documents on a non-indexed key

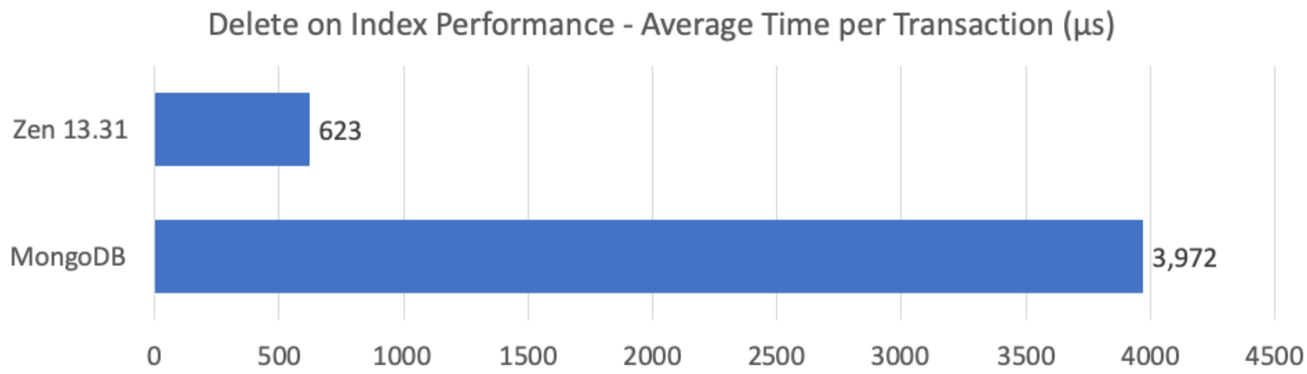
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



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

Test 5a: Delete 10,000 documents on an indexed key

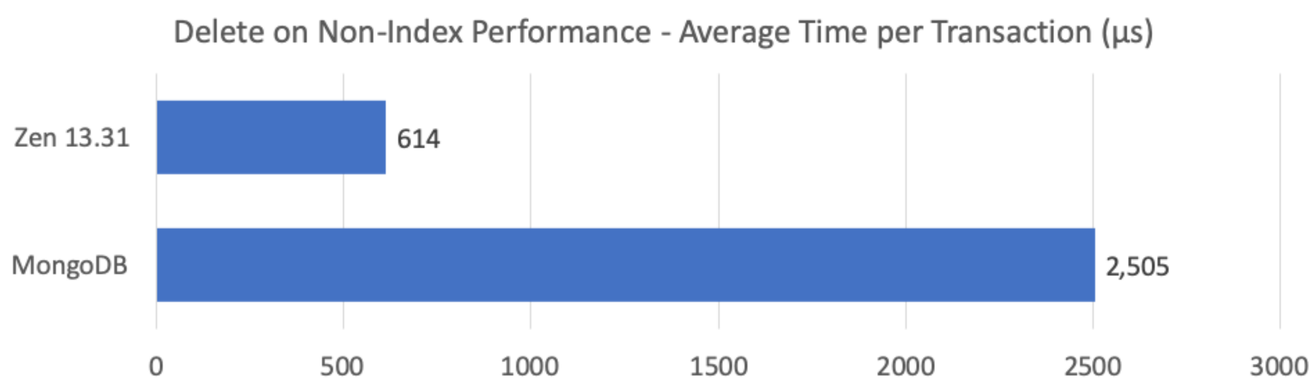
Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



The average time to delete a row (taking the average of all 10,000 deletes) for Actian Zen was many times faster than MongoDB deletes.

Test 5b: Delete 5,000 documents on a non-indexed key

Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



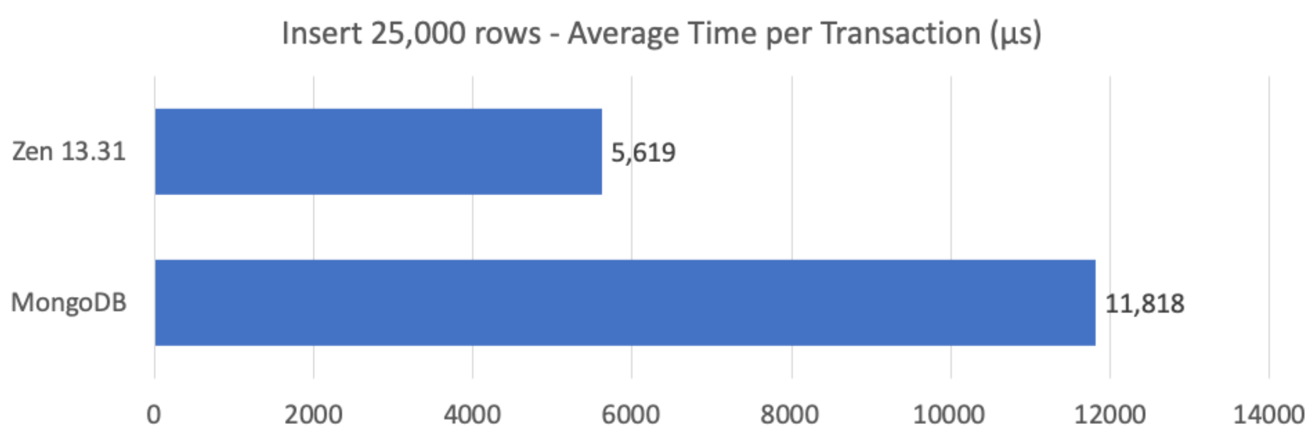
Actian Zen's average time to delete a document (taking the average of all 5,000 deletes) was over 4 times faster than MongoDB deletes using the same filter.

Android, Sync

The following figures display the average time elapsed for each database transaction for both Actian Zen and MongoDB Mobile with synchronization back to both Actian Zen Server and MongoDB Server, respectively. Each test was executed 5 times and the fastest value was used. Please note the values represent the average time per transaction, not for the entire test. No selects test were performance due to the synchronization.

Test 2: Insert 25,000 documents

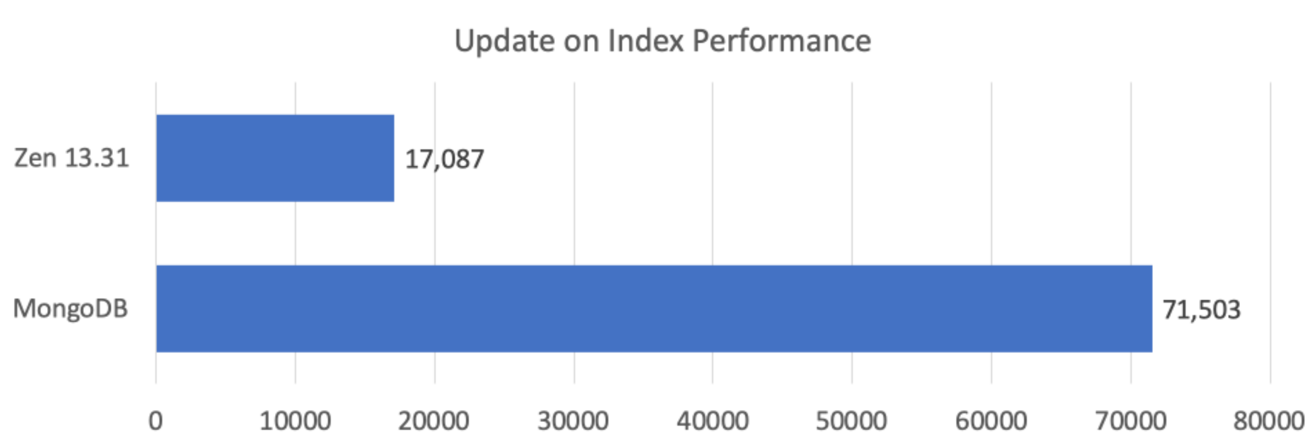
Below are the average times (in microseconds) it took to insert a complete document of randomly-generated data into the Contacts database on Actian Zen and MongoDB.



Actian Zen's average time to insert a single document (taking the average of all 25,000 inserts) was almost 50% faster than MongoDB inserts.

Test 4a: Update 10,000 documents on an indexed key

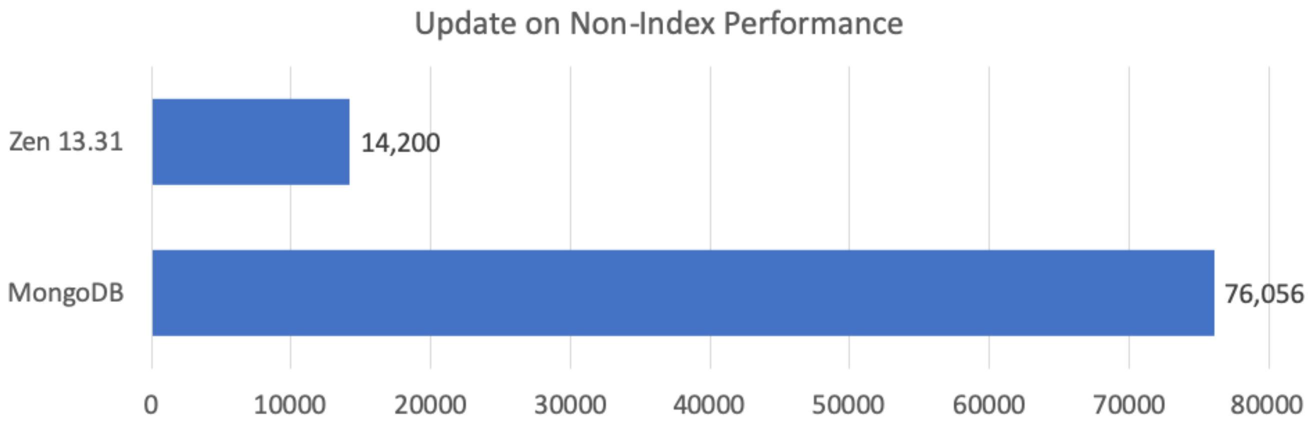
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



The average time to update a single key (taking the average of all 10,000 updates) for MongoDB was slower than Actian Zen updates.

Test 4b: Update 5,000 documents on a non-indexed key

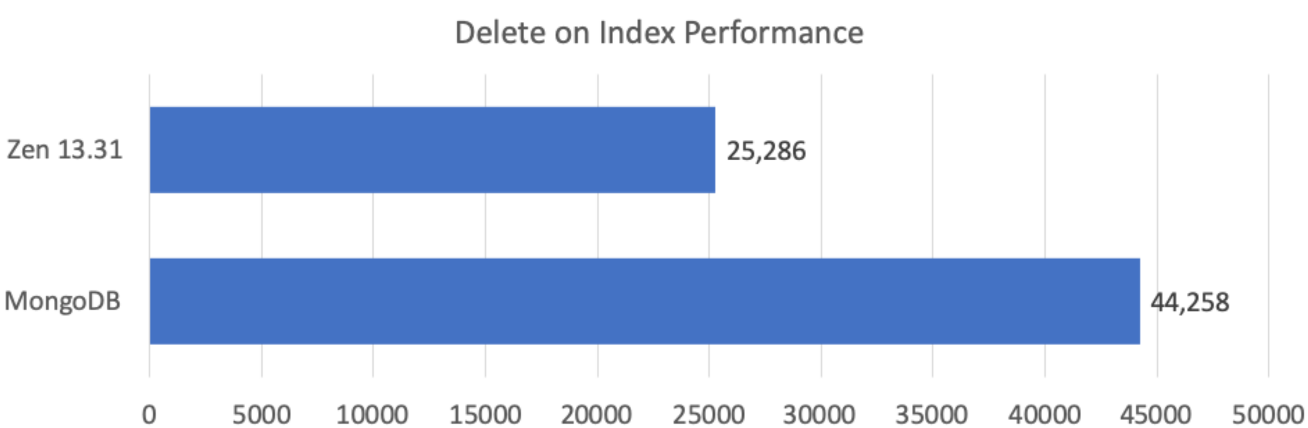
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



Actian Zen's average time to update a single key (taking the average of all 5,000 updates) was over 5 times faster than MongoDB updates using the same filter.

Test 5a: Delete 10,000 documents on an indexed key

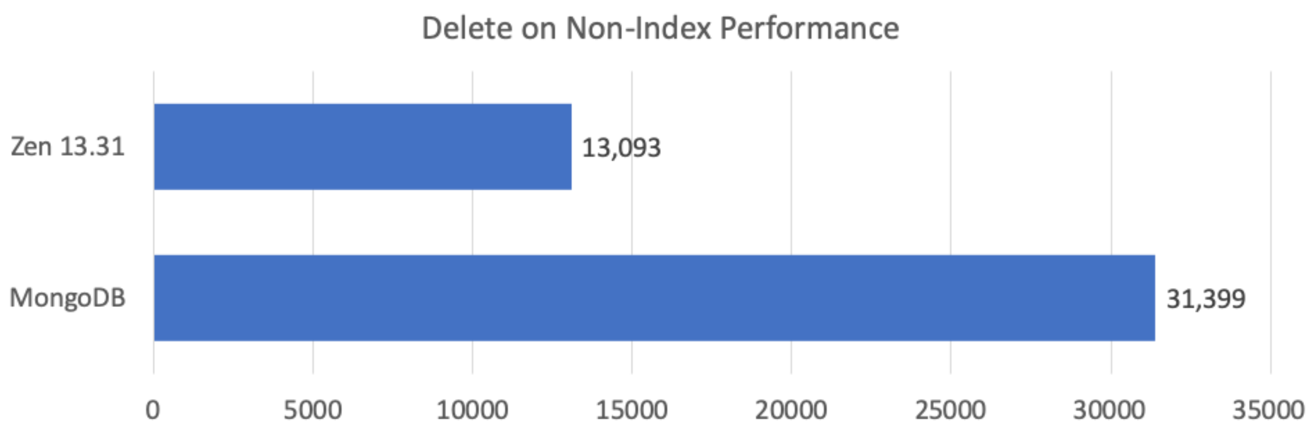
Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on an indexed key for both Actian Zen and MongoDB.



The average time to delete a row (taking the average of all 10,000 deletes) for Actian Zen was faster than MongoDB deletes.

Test 5b: Delete 5,000 documents on a non-indexed key

Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on a non-indexed key for both Actian Zen and MongoDB.



Deleting documents on a non-indexed key produced results consistent with before. Actian Zen's average time to delete a document (taking the average of all 5,000 deletes) was faster than MongoDB deletes using the same filter.

Conclusion

In a server configuration on the same hardware, Actian Zen outperformed MongoDB by 6.3x on inserts, 2x on queries of documents on an indexed key, 8x on queries of documents on a non-indexed key, 791x on updates of documents on an indexed key, 749x on updates of documents on a non-indexed key, 148x on deletes of documents on an indexed key and 110x on deletes of documents on a non-indexed key.

With no synchronization, Actian Zen Edge on Android outperformed MongoDB by 7x on inserts, underperformed by 60% on queries of documents on an indexed key, 66% on queries of documents on a non-indexed key, 46x on updates of documents on an indexed key, 4x on updates of documents on a non-indexed key, 6.3x on deletes of documents on an indexed key and 4x on deletes of documents on a non-indexed key.

With synchronization, Actian Zen Edge outperformed MongoDB by 50% on inserts, by 4x on updates of documents on an indexed key, 5.4x on updates of documents on a non-indexed key, outperformed by 43% on deletes of documents on an indexed key and 2.3x on deletes of documents on a non-indexed key.

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 MongoDB. 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, shows a marked, and sometimes astonishing, performance advantage to Actian Zen. This is especially true in the important write operations insert, update and delete.

Overall, Actian Zen is an excellent choice for companies needing high performance in a scalable NoSQL 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, visit www.actian.com.