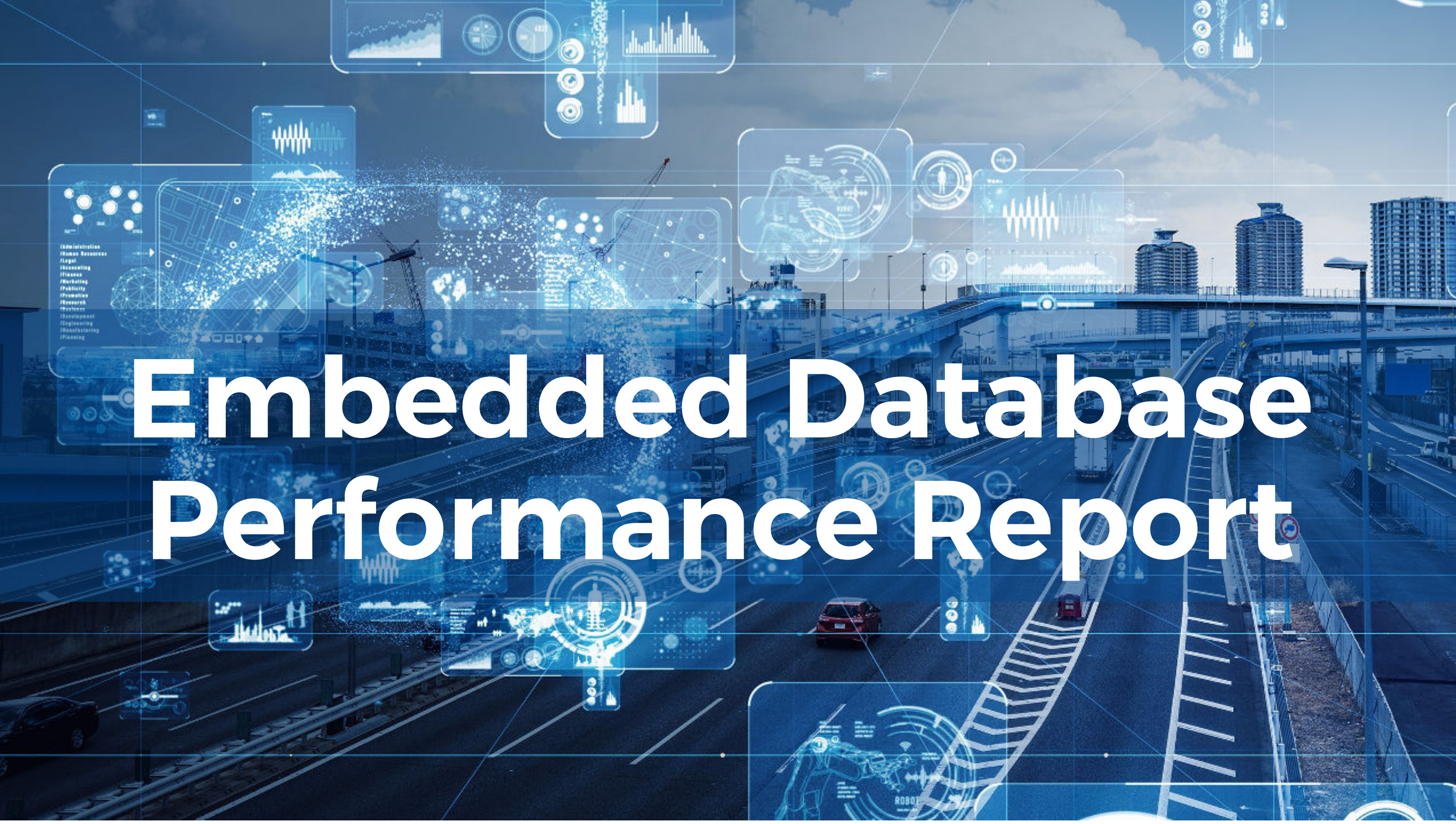


Embedded Database Performance Report

Actian Zen more than 6x faster than Couchbase

MCG Global Services Benchmark Results

September 2018

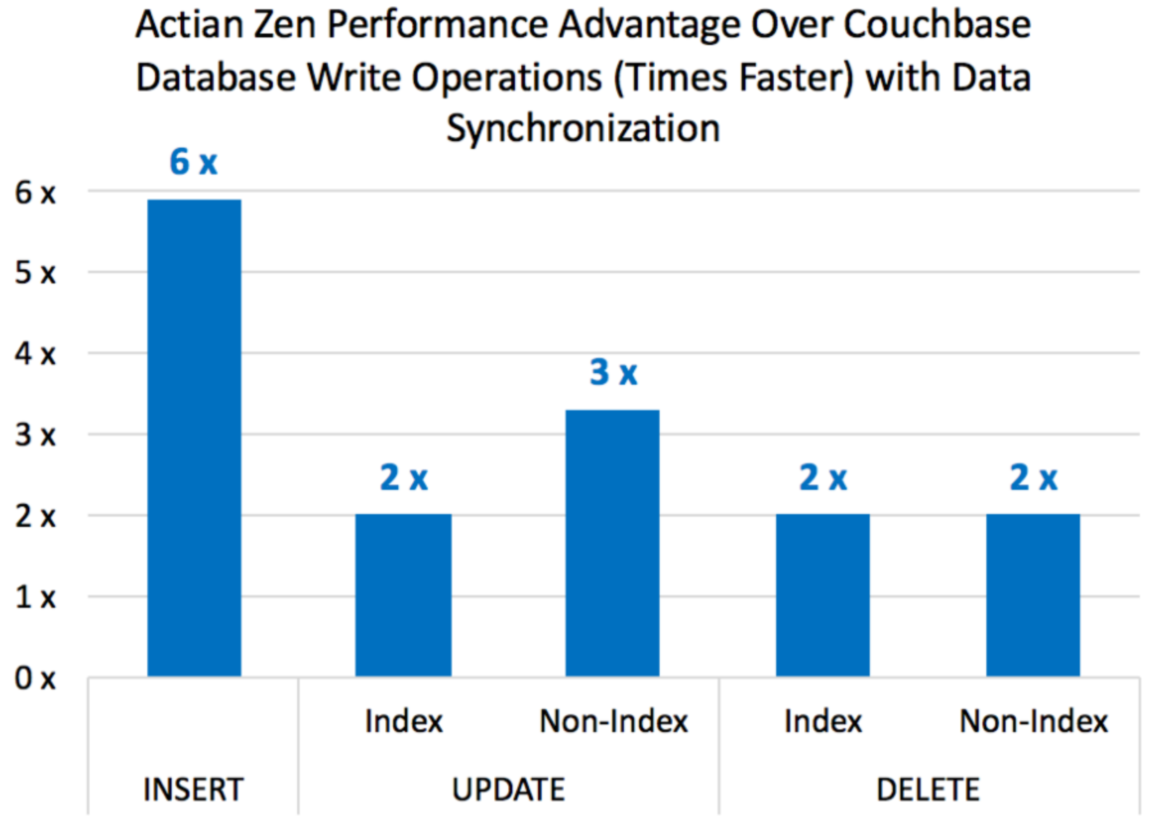
ACTIAN™

Key insights

- This benchmark did a head-to-head comparison of Actian Zen Core and Couchbase Lite, both running on a Nokia 2, Android 7 handset (ARM-based) with 1GB DRAM

- Actian Zen Core outperformed Couchbase Lite for Indexed and Non-Indexed data management by:
  – more than 5x on inserts, up to 2x on deletes, and over 2x on updates

- Actian Zen Core plus Actian Zen Edge was also faster as a client-server combination than Couchbase Lite plus Couchbase Sync Gateway plus Couchbase server, replicating IoT Device or Smartphone to gateway data sharing —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

ACTIAN™

Embedded NoSQL Database Performance Benchmark

Product Profile and Evaluation:
Actian Zen and Couchbase

By William McKnight and Jake Dolezal
McKnight Consulting Group
September 2018

Sponsored by

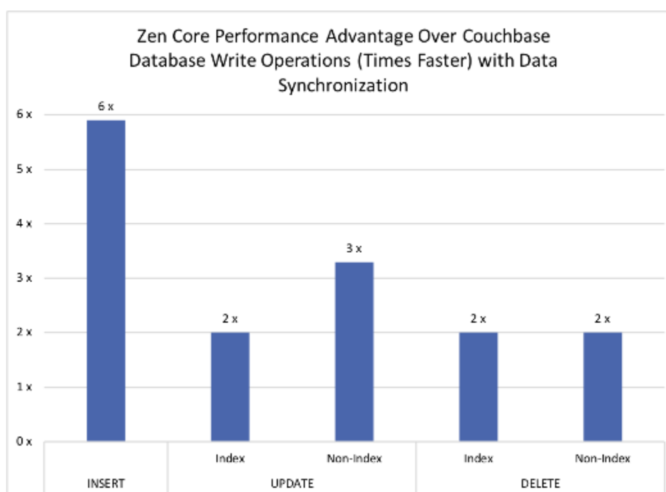
Executive Overview

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 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 applications right on the 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.

All these web, mobile, and IoT applications have generated a new set of technology requirements. Embedded database architecture needs to be far more agile than ever before, and requires an approach to real-time data management that can accommodate unprecedented levels of scale, speed, and data flexibility. Sometimes relational databases are unable to meet these new requirements, and 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.

To quantify embedded database performance, we conducted this benchmark study, which focuses on the performance of mobile application-ready, NoSQL, embedded database solutions Actian Zen and Couchbase. 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 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 Couchbase Lite installed on the same Android device, and Zen and Couchbase Server installed on a server. 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 and Couchbase revealing some of the differentiators in the two products.

Actian Zen Edge 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. Without synchronization of data to the server, Actian Zen Edge outperformed Couchbase by 5x on inserts, 7x on queries of 10,000 documents on an indexed key, 6x on queries of 5,000 documents on a non-indexed key, 2x on deletes of 10,000 documents on an indexed key and on deletes of 5,000 documents on a non-indexed key, 2x on updates on the indexed key and 3x on non-indexed key updates.

With synchronization of data to the server, Actian Zen Edge outperformed Couchbase by 6x on inserts, 2x on deletes of 10,000 documents on an indexed key and on deletes of 5,000 documents on a non-indexed key, 2x on updates on the indexed key and 3x on non-indexed key 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 NoSQL Database Selection

Organizations that utilize 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—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 Couchbase 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 Couchbase does not—including auto-reconnect networking, automated defragmentation, multi-user support, and concurrent write capabilities.

Both platforms offer NoSQL support. Zen is natively NoSQL and is flexible enough to be document-based or a key-value store. Couchbase is JSON document-oriented. 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. Couchbase 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, both Actian Zen and Couchbase come with the capability to automatically synchronize in real time between a remote device and 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. With Couchbase, you must utilize a Sync Gateway utility to serve as the proxy between Couchbase Lite on the remote client and Couchbase Server. Actian has real time synchronization capability of Actian Zen Edge to Core via the Btrieve API without an intermediary, which can allow you to achieve scale with simplicity.

Platform maturity is also a consideration. Couchbase was initially released in 2010. 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 Android 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 Couchbase—installed on the same Android device. We also tested a configuration with real-time synchronization. The server had both the latest versions of Actian Zen Core and Couchbase Server installed on the same machine. Also, Couchbase's Sync Gateway was installed to serve as the intermediary between Couchbase Lite and Server. All components were deployed on a local area network.

NoSQL DBMS

Embedded RDBMS	Actian Zen	Couchbase
Version	13.10.030	Server 5.1.1 Community Lite 2.1 Community Sync Gateway 2.1 Community

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

Server

Hardware	Lenovo ThinkPad X1 Carbon G6 20BS006UUS x64-based PC
Processor	2x Intel Core i7-5600U @ 2.60GHz
RAM	8 GB
OS	Microsoft Windows 10 Enterprise 10.0.16299

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 Couchbase Lite we used the native APIs to execute the database transactions in order to test its functionality and performance, rather than SQL (Zen) or N1QL (Couchbase SQL for JSON).

Use Case 1: Open and Close Connections in Rapid Succession

NOTE: We did not do this run for the Android device since it is standard practice for mobile developers to open a database connection and leave it open while the app is running. Also, we did not use this test for the synchronization benchmark, having no applicable use for that workload.

Use Case 2: Insert Performance

Mobile 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 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 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**

NOTE: We did not use this test for the synchronization benchmark, since selecting documents by themselves would not constitute the complete workload.

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**

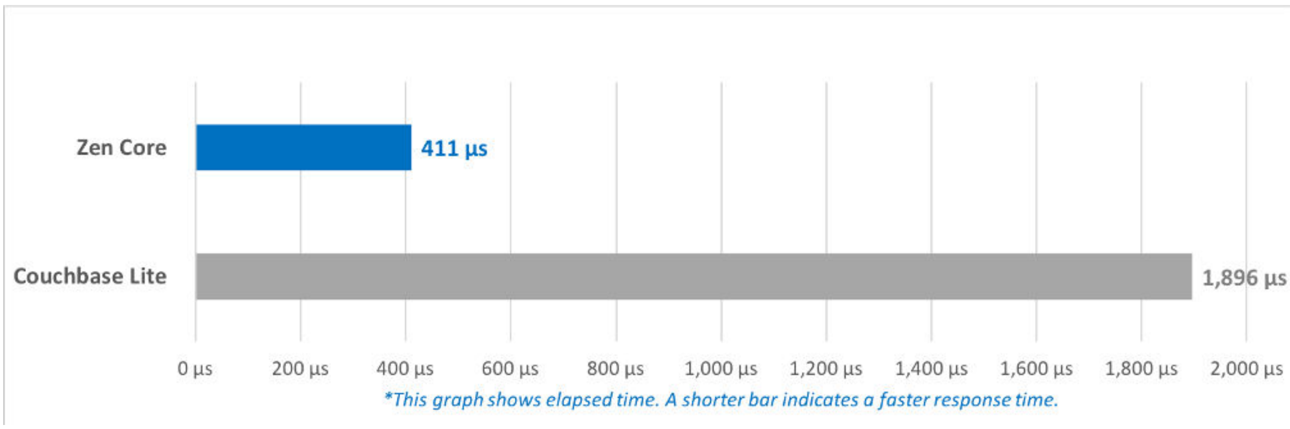
Benchmark Results

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

Local Only (No Synchronization)

Test 2: Insert 25,000 documents

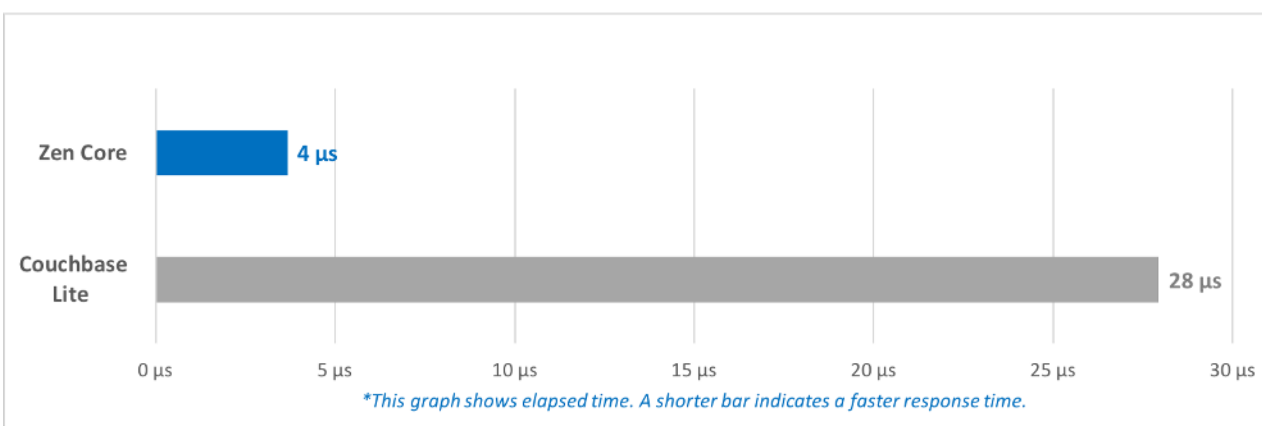
Below are the average times (in microseconds) it took to insert a complete document of randomly-generated data into the Contacts database on Actian Zen and Couchbase.



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 4.6 times faster than Couchbase inserts.

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

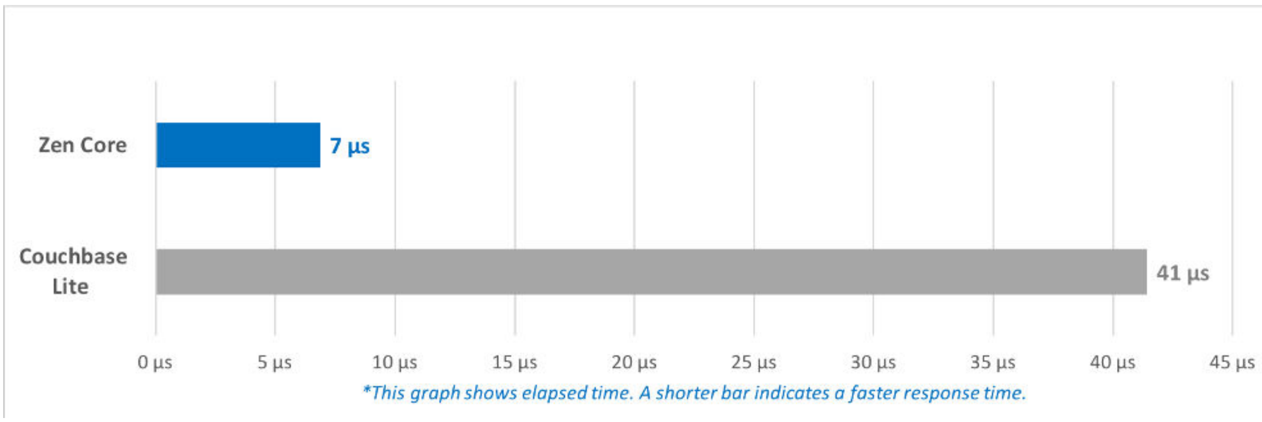
Below are the average times per document (in microseconds) it took to bulk select records from the Contacts database applying a filter on an indexed key for both Actian Zen and Couchbase.



Both platforms responded very quickly. Couchbase's fetch rate per document (taking the average of all 10,000 documents) was 7 times that of Actian Zen's.

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

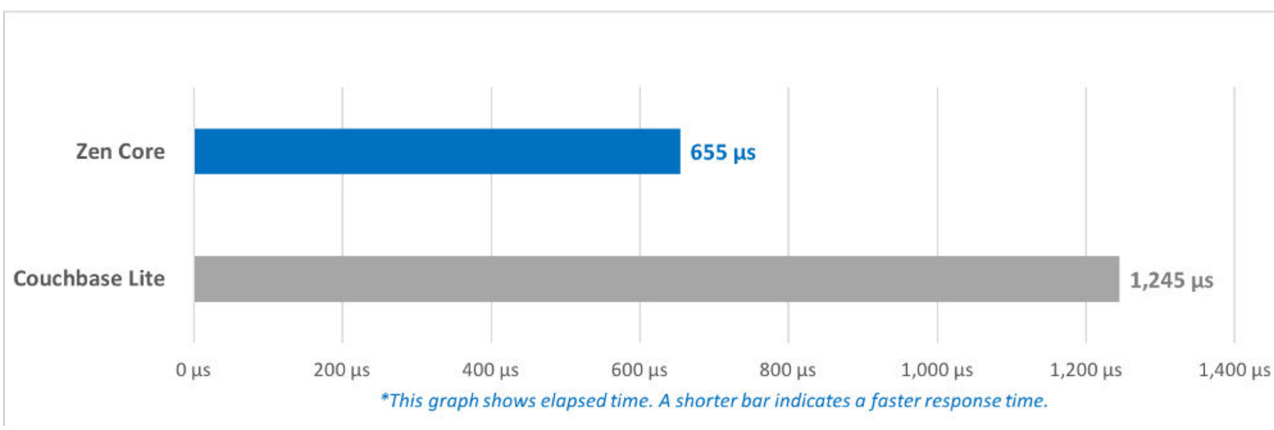
Below are the average times per document (in microseconds) it took to bulk select records from the Contacts database applying a filter on a non-indexed key for both Actian Zen and Couchbase.



Again both platforms responded very quickly. Couchbase's fetch rate per document (taking the average of all 10,000 documents) was 5.8 times that of Actian Zen's.

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

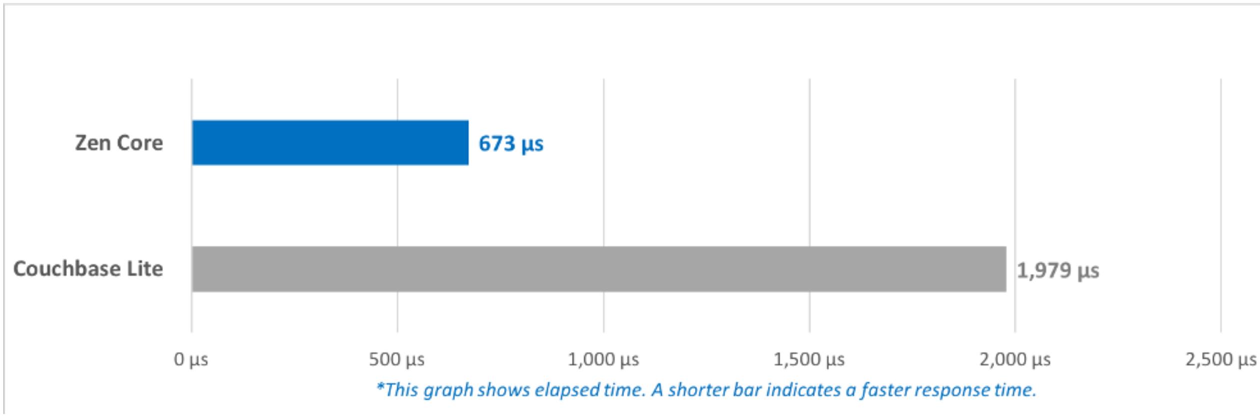
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on an indexed key for both Actian Zen and Couchbase.



This one was a close test. The average time to update a single key (taking the average of all 10,000 updates) was only 1.9 times faster than Couchbase updates.

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

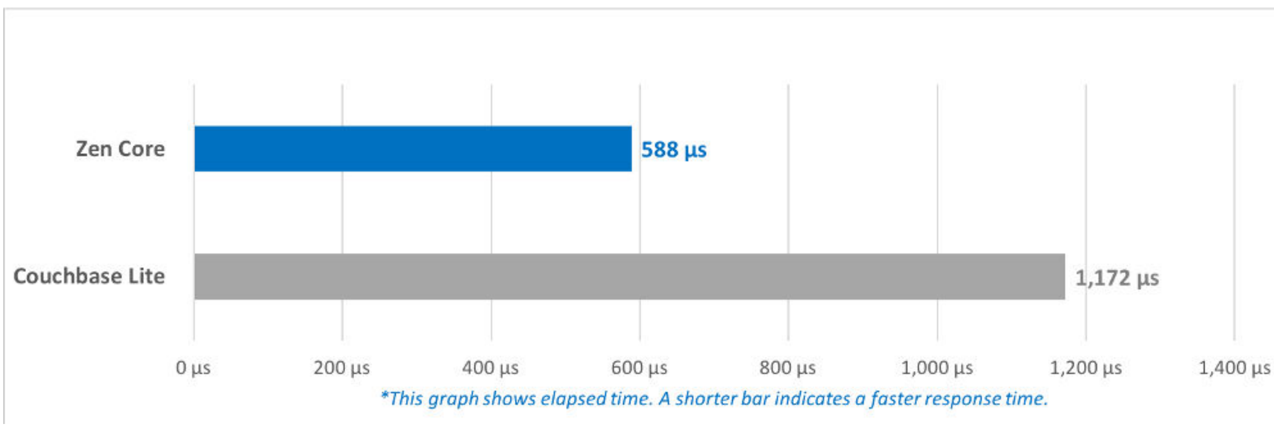
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on a non-indexed key for both Actian Zen and Couchbase.



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 2.9 times faster than Couchbase updates using the same filter.

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

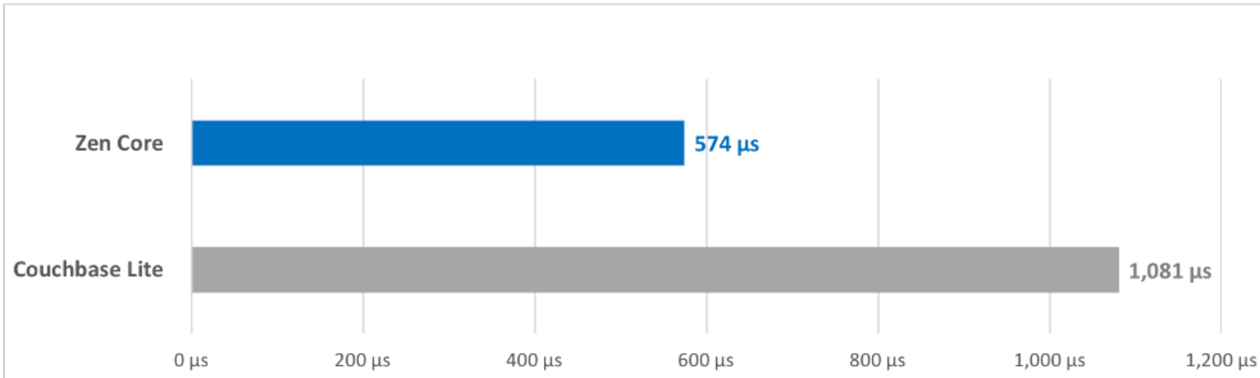
Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on an indexed key for both Actian Zen and Couchbase.



Both were very fast. The average time to delete a row (taking the average of all 10,000 deletes) for Actian Zen was simply 2 times faster than Couchbase 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 Couchbase.



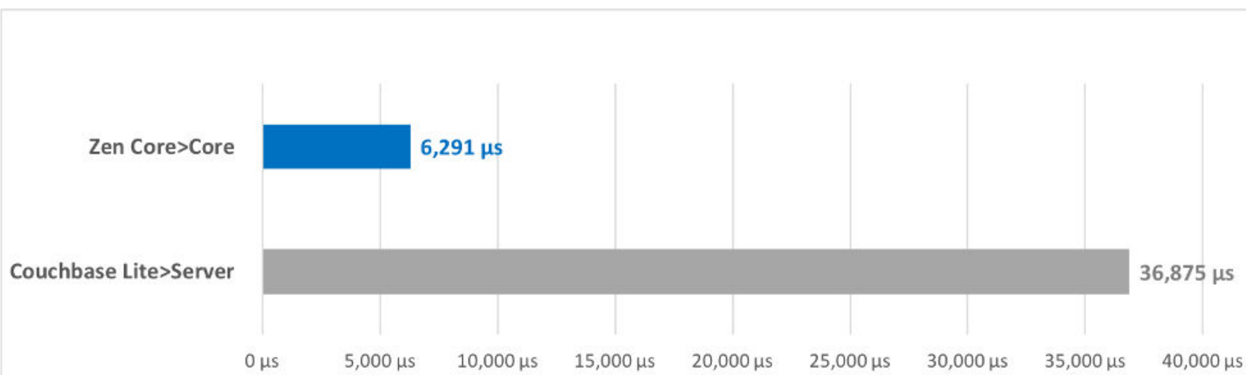
This graph shows elapsed time. A shorter bar indicates a faster response time.

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 1.88 times faster than Couchbase updates using the same filter.

Synchronization

Test 2: Insert 25,000 documents and sync

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

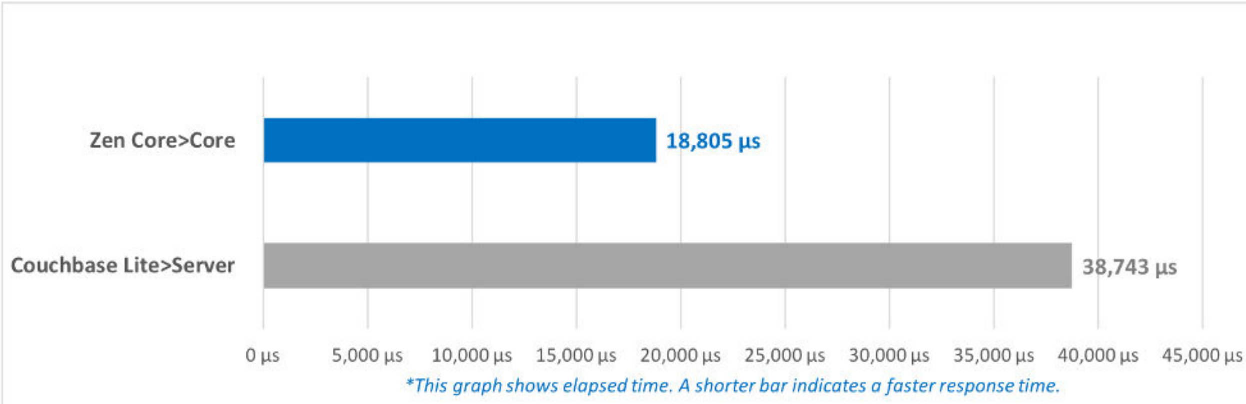


This graph shows elapsed time. A shorter bar indicates a faster response time.

This test revealed the first major performance differentiator. Actian Zen's average time to insert a single document (taking the average of all 25,000 inserts) was 5.9 times faster than Couchbase inserts.

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

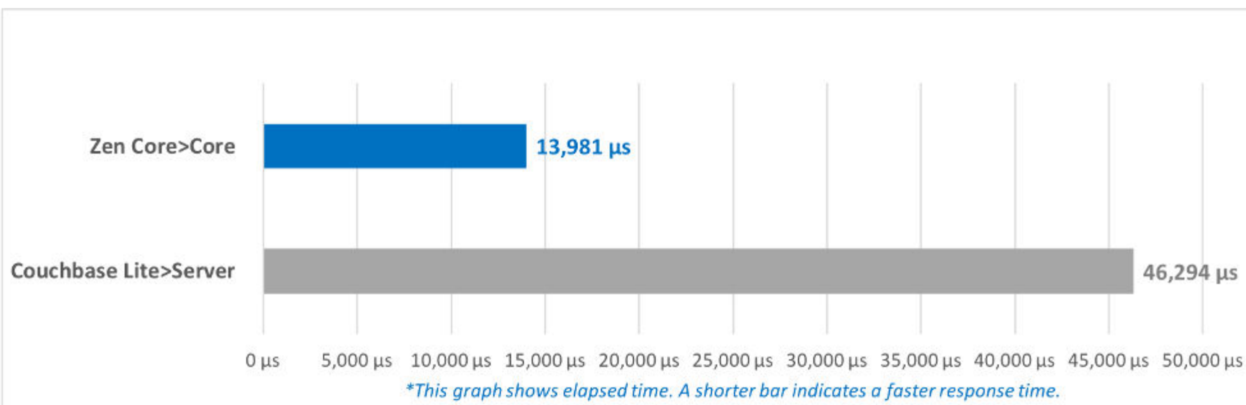
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on an indexed key for both Actian Zen and Couchbase.



Actian Zen's average time to update a single key (taking the average of all 10,000 updates) was 2 times faster than Couchbase updates.

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

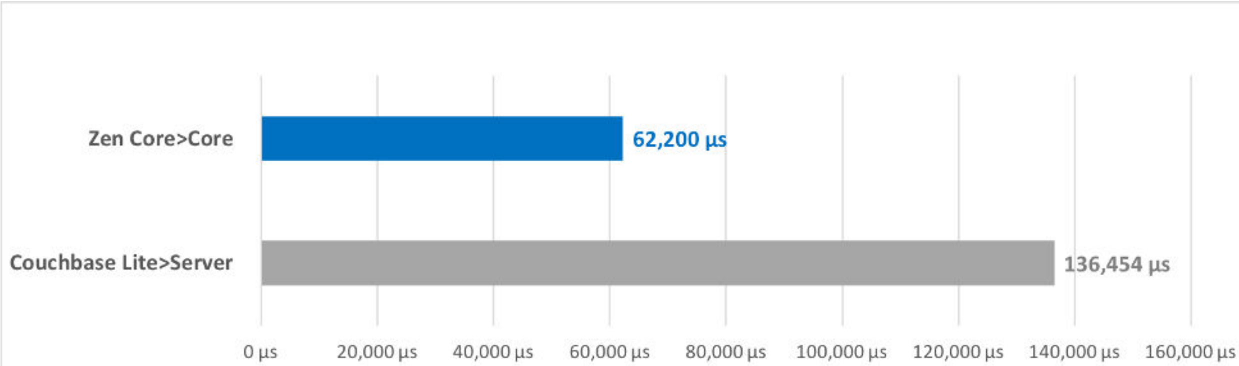
Below are the average times (in microseconds) it took to update a single key in the Contacts database applying a filter on a non-indexed key for both Actian Zen and Couchbase.



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 3.3 times faster than Couchbase updates using the same filter.

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

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 Couchbase.

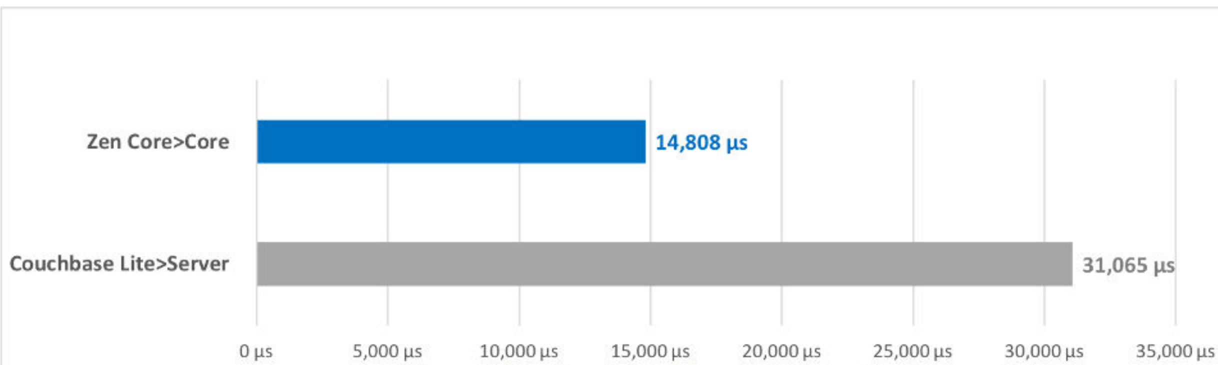


*This graph shows elapsed time. A shorter bar indicates a faster response time.

Both were very fast. Its average time to delete a row (taking the average of all 10,000 deletes) was a little less than half the time Couchbase took.

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

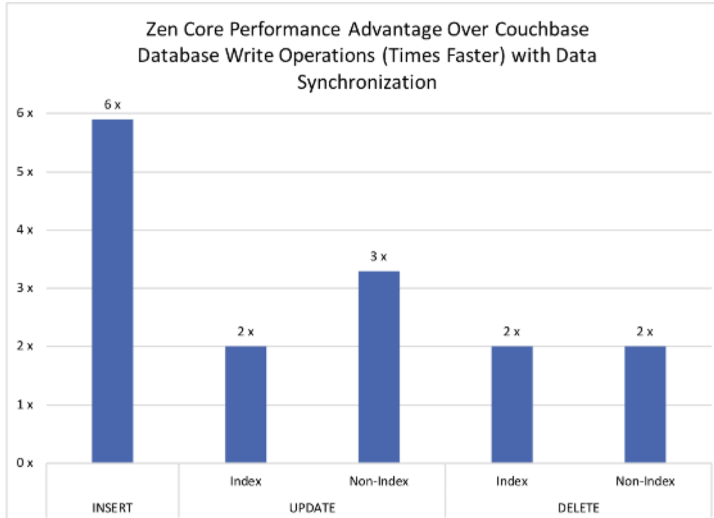
Below are the average times (in microseconds) it took to delete a document in the Contacts database applying a filter on a non-indexed key for both Actian Zen and Couchbase.



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

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

Conclusion



Without synchronization of data to the server, Actian Zen Edge outperformed Couchbase by 5x on inserts, 7x on queries of 10,000 documents on an indexed key, 6x on queries of 5,000 documents on a non-indexed key, 2x on deletes of 10,000 documents on an indexed key, 2x on deletes of 5,000 documents on a non-indexed key, 2x on updates on the indexed key and 3x on non-indexed key updates.

With synchronization of data to the server, Actian Zen Edge outperformed Couchbase by 6x on inserts, 2x on deletes of 10,000 documents on an indexed key, 2x on deletes of 5,000 documents on a non-indexed key, 2x on updates on the indexed key and 3x on non-indexed key updates.

Actian Zen outperformed Couchbase in all of the fundamental database operations. These tested operations underlie nearly all operations that occur on an embedded database for an IoT or mobile 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 Couchbase. 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. This is especially true in the important write operations insert, update and delete.

Overall, Actian Zen is an excellent choice for IoT or mobile 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.