



INFRASTRUCTURE  
WIKIBON

## Oracle Autonomous JSON Takes on Document DBs by, David Floyer

August 24th, 2020

The premise of this research is that large enterprises deploying mission-critical applications on Oracle Databases will generally benefit from lower costs of development and operation by using a unified database strategy as a service. More importantly, enterprise developers and operations can develop and deploy next-generation data-driven applications faster.

An illustration of this premise is an evaluation of how developers should handle JSON documents. One choice is to provide the function in a separate document database such as Amazon DocumentDB or MongoDB. The other choice is to integrate an Autonomous JSON document management database into a unified database. Oracle announced this capability at the Oracle Developer Live virtual conference in August 2020.

Wikibon recommends that large organizations running mission-critical Oracle workloads evaluate Oracle UDBMS as the strategic base for the development of future high-value real-time multi-database automation applications.

Wikibon also recommends the evaluation and adoption of the Autonomous JSON Database as a foundation for future application development-centric environments using Oracle Databases.

## Introduction

The business world is moving to real-time end-to-end automation of business processes, which requires integrating all data from all sources in real-time. Most of the important real-time data is in a database. Many new non-SQL databases have been developed in the past decade—the MongoDB document database, Amazon DocumentDB, and Oracle Autonomous JSON Database are examples of non-SQL document databases.

Oracle is the leading database vendor with the Oracle SQL relational database as the functionally richest tier-1 database. Oracle's strategy is to create a **"Unified Database Management System" (UDBMS\*)** by integrating non-SQL databases as a service on-premises and in the cloud. The Oracle UDBMS has the same performance, automation, fault-tolerance, sharding, and security umbrella for all databases.

An example of this UDBMS direction for Oracle is the recent Autonomous JSON Database availability.

\* Note: Oracle is currently calling this a Converged Database.

## Premise

The premise of this research is that large enterprises deploying mission-critical applications on Oracle Databases will generally benefit from lower costs of development and operation by using a unified database strategy as a service. More importantly, enterprise developers and operations can develop and deploy next-generation data-driven applications faster.

An illustration of this premise is an evaluation of how developers should handle JSON documents. One choice is to provide the function in a separate document database such as Amazon DocumentDB or MongoDB. The other choice is to integrate an Autonomous JSON document management database into a unified database. Oracle announced this capability at the Oracle Developer Live virtual conference in August 2020.

## The Importance of UDBMS

By far the most significant business simplification comes from real-time end-to-end automation of business processes, which requires the ability to integrate any data from any sources in real-time. This type of simplification eliminates complex asynchronous business processes.

These high-value data-driven automation applications require a data platform that will enable developers to blend data from different sources seamlessly. The sources will have different characteristics, come from different databases, and the results are required in real-time or near real-time.

A Unified DataBase Management System (UDBMS) is a data platform that will allow different databases to work together seamlessly. A UDBMS must be able to integrate RDBMS, NoSQL (document, key-value, column-oriented, graph databases), streaming data, ML and analytics, and other specialized databases. Also, a UDBMS is much stronger if all the components share the same management, virtualization, clustering, sharding, replication, in-memory, and security capabilities.

Another UDBMS pre-requisite is a cloud-native implementation of the platform. In a multi-cloud enterprise environment, all cloud implementations must be identical, both on-premises and in the cloud.

The costs will be lower, and availability/security improved if the UDBMS vendor has the responsibility for patching, updating, and upgrading all the database software as a single deployment. The testing of any UDBMS must include low-latency within and between database functionality.

Wikibon assesses that Oracle is the leading UDBSM vendor at the moment.

## Document Databases for Developers

Developers use the intuitive distributed data model within a document database to improve their productivity and code quality. JSON-like schemas are dynamic and self-describing, and developers do not need to pre-define any schema.

### Amazon Document DB

One public cloud document database is Amazon DocumentDB. Amazon DocumentDB is built on top of the AWS Aurora platform, itself a derivative of MySQL. DocumentDB supports a primary node for writes and up to 15 replicas, which can be used to scale read operations. DocumentDB offers good integration with AWS development tools for developers using AWS.

The importance of MongoDB (see below) is shown by the fact that Amazon DocumentDB is a document database that comes with claimed MongoDB API compatibility.

### MongoDB

The leading developer document database is MongoDB, which comes in two flavors. MongoDB Enterprise Advanced is the original on-premises database. MongoDB Atlas is a managed cloud database platform available on AWS, Microsoft Azure, and GCP. Developers highly regard both the stability and functionality of MongoDB.

### Oracle Autonomous JSON Database

Oracle is the new kid on the block. Oracle chose to integrate an Autonomous JSON document management into its UDBSM platform and make it simple for developers to use JSON documents. The same JSON data format is used in the application and the database. JSON and relational data can be freely mixed or joined. Any JSON element can be indexed to improve OLTP performance. An application can be built with or without SQL. The programmer can define full ACID properties if required.

Oracle Autonomous JSON is a cloud service built for JSON-centric application development and provides indexed native JSON storage, which is accessed using document APIs. It's now available on Oracle Autonomous Database in Oracle Cloud Infrastructure (OCI), Oracle Exadata Cloud@Customer, and Dedicated Region Cloud@Customer.

## Document Database Comparative Performance

### Document Database Benchmarks

One of the most respected benchmarks for NoSQL databases is the open-source YCSB (Yahoo! Cloud Serving Benchmark), written in Java. This section analyses the results of two YCSB benchmarks run by Mongo and Oracle.

The first [YCSB](#) benchmark was run by MongoDB between Amazon DocumentDB and MongoDB Atlas in 2019. The second YCSB benchmark was run by Oracle in 2020 using Autonomous JSON Database and compares the MongoDB Atlas results from the first YCSB benchmark.

YCSB uses primary key queries. In the benchmarks, three YCSB workloads were run. The first workload was 95% find 5% write, the second 50% find 50% write, and the third was 5% find 95% write. There were two data sets, one with 4 million documents that could fit entirely into DRAM, and one with 81 million documents much larger than the DRAM available. Both data sets used 2.5Kb documents and contained 25 fields.

The Amazon DocumentDB cluster used three (3) AWS r4.4 large instances, the MongoDB Atlas cluster used

# Oracle Autonomous JSON Takes on Document DBs

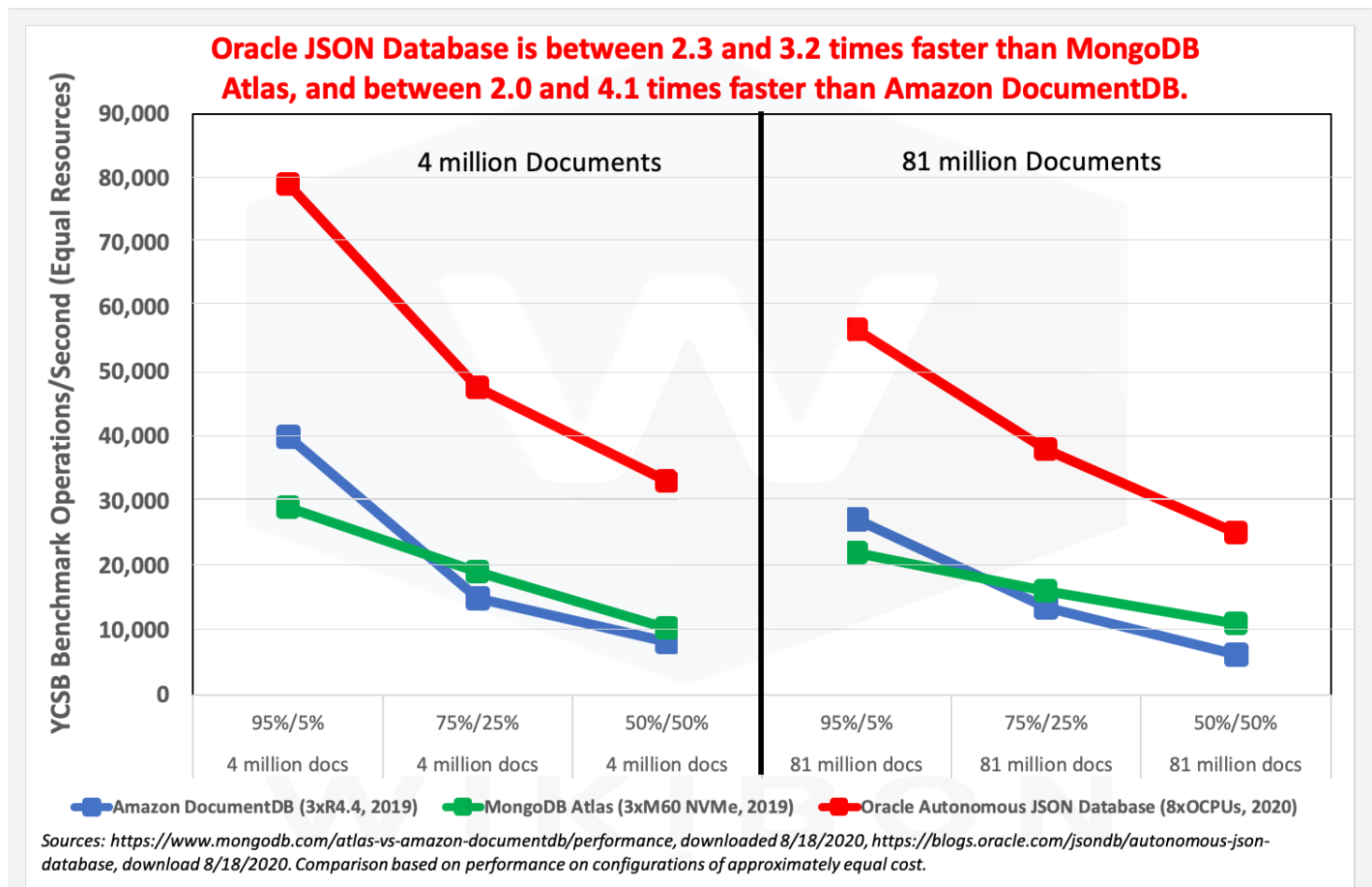
three (3) M60 NVMe instances, and the Oracle Autonomous JSON Database used eight (8) OCPUs in OCI. The cost of running the three (3) MongoDB Atlas clusters was \$3.95 per hour, and the cost of the Autonomous JSON Database on eight (8) OCPUs was \$2.74 per hour.

## Document Database Benchmark Results

Figure 1 below shows the combined results of the YCSB benchmarks discussed in the previous section. The y-axis is the number of operations per second, assuming approximately equal resources as defined by the cost of the resources in a cloud service environment.

The left-hand part of the x-axis shows the YCSB results for a small 4 million document database. The right-hand part of the x-axis shows the results for a much larger database with 81 million documents. Within each part are three benchmarks with different IO configurations (95% find/5% write, 50% find/50% write, 5% find/95% write).

The Oracle Autonomous JSON Database is shown in red, MongoDB is in green, and Amazon DocumentDB is in blue.



**Figure 1: Document Database Performance Comparisons between Amazon DocumentDB, MongoDB Atlas, and Oracle Autonomous JSON Database.**

Source: *YCSB Benchmark between MongoDB Atlas & Amazon DocumentDB, & YCSB Benchmark between MongoDB Atlas and Oracle Autonomous JSON Database. Comparison based on Operations/second on configurations of approximately equal cloud hardware and costs.*

Wikibon concludes that Oracle Autonomous JSON Database is between 2.3 and 3.2 times faster than

MongoDB Atlas and between 2.0 and 4.1 times faster than Amazon DocumentDB for workloads represented by the YCSB benchmark. In general, the higher the percentage of IO writes, the better the Autonomous JSON Database performed. Wikibon also concludes that the Autonomous JSON database used for Oracle application development is overall at least as functional as the MongoDB database. Performance matters in the cloud because the faster the workload is completed, the quicker the enterprise stops paying the cloud service provider.

Wikibon analysis suggests that the functionalities in the JSON Database that mainly contributed to the improved benchmark performance are Serverless Auto-scaling, the complete independence of data provisioning and server provisioning for finds and writes, and the JSON Search Index.

Wikibon also concludes that the Oracle Autonomous JSON Database integrates well with other Oracle databases, and offers best-of-breed functionality and performance in developer-centric Oracle database environments.

## MongoDB Socialite Benchmark vs. Amazon DocumentDB

Mongo has an internal Document Database benchmark used for regression testing named Socialite. Socialite simulates a social networking application using the MongoDB API and uses all the access patterns and complex queries supported in MongoDB. [Mongo has created a publicly available harness](#) to run alternative Document Databases and compared the performance of MongoDB and Amazon DocumentDB on the Socialite benchmark.

The results for MongoDB were between 7,000 and 16,000 operations per second. The results of Amazon DocumentDB were a maximum of 200 operations/second. Overall, MongoDB was over 80 times faster. MongoDB claims that Amazon DocumentDB uses collection scans in preference to indexes for complex queries, leading to much poorer performance results.

Wikibon concludes MongoDB Atlas is superior in performance and functionality compared to Amazon DocumentDB for all except find-only simple query environments.

Wikibon observes the Oracle JSON Database does not support the MongoDB APIs and has not been compared with MongoDB using the Socialite Database harness. There may be Document DB applications where MongoDB is a better strategic fit outside of application development for Oracle environments that we analyzed in depth.

## Integrating Autonomous JSON Document Management Functionality

In the premise section above, we defined the choice as providing a separate database for JSON documents such as MongoDB, or integrating it into a UDBSM environment.

Wikibon's assessment of the Oracle Autonomous JSON is well integrated into the Oracle UDBMS ecosystem. For example, the programmer can add relational data capabilities simply and transform it into an Autonomous Transaction Processing database.

Wikibon's assessment is that the UDBMS integration can usually improve both Oracle programmer and operational productivity more than any additional functionality from MongoDB.

In yesterday's announcement, the target of much of Oracle's commentary was centered around simplicity, ease-of-use, and performance advantages compared to MongoDB. For MongoDB, this is an acknowledgment that they have achieved the status of a primary target for other vendors.

Wikibon expects that Oracle will provide additional developer-centric services to the Autonomous JSON Database over the next eighteen months.

## Autonomous JSON Conclusions

At the moment, Wikibon believes that Oracle is the leading UDBMS vendor. As an illustration, the announcement of a document management system for Oracle Database developers using JSON is competitive in function and price with MongoDB. Wikibon's analysis is that the Autonomous JSON functionality is a strong addition to the Oracle UDBMS, as shown in the benchmark analysis above. This environment provides faster end-to-end development cycles, and operations are simplified with a machine-learning-powered autonomous environment.

There is a stark contrast in the database philosophies of AWS and Oracle. AWS has taken a series of different open-source databases optimized for a specific function, running on the same infrastructure (PaaS) services. The above performance analysis shows that the Oracle JSON Database and the MongoDB are superior to Amazon DocumentDB.

IT organizations spend significant time analyzing different databases for different projects. For a distributed stand-alone IoT project deploying a best-of-breed time-series database makes business sense. However, businesses will also require integrated real-time applications across different database types, and Wikibon concludes that enterprise development must have access to UDBMS technology.

AWS and Microsoft have most of the pieces to develop a UDBMS but will need to work hard to provide an integrated UDBMS platform for the next generation of real-time automation applications. Both will also need to improve individual databases either by significant development commitments or by acquisition. There are startups such as [Splice Machine](#) who are building connections between different databases. These startups may well be acquisition targets for cloud providers.

Microsoft is in a better position than AWS to develop a UDBMS, as they can build on their tier-1 SQL Server Database. AWS Aurora is not a tier-1 database at the moment. It will need significant time and effort to develop the performance and recoverability umbrella to compete against the IBM, Microsoft, and Oracle tier-1 relational databases. Also, some of the AWS non-SQL databases such as Amazon DocumentDB will need significant upgrading in function and performance, or be replaced by acquisition.

Wikibon concludes that a unified approach is a strategically better platform for developing large-scale integrated data-driven applications that will differentiate large enterprises. In particular, Oracle Database users will have a faster time-to-market to deliver real-time automation applications that can radically improve enterprise productivity more than MongoDB or AWS DocumentDB.

## Action Item

**Wikibon recommends that large organizations running mission-critical Oracle workloads evaluate Oracle UDBMS as the strategic base for the development of future high-value real-time multi-database automation applications.**

**Wikibon also recommends the evaluation and adoption of the Autonomous JSON Database as a foundation for future application development-centric environments using Oracle Databases.**

David Floyer spent more than 20 years at IBM, holding positions in research, sales, marketing, systems analysis and running IT operations for IBM France. He worked directly with IBM's largest European customers, including BMW, Credit Suisse, Deutsche Bank and Lloyd's Bank. Floyer was a Research Vice President at International Data Corporation (IDC) and is a recognized expert in IT strategy, economic value justification, systems architecture, performance, clustering and systems software.



**David Floyer**

@dfloyer

david.floyer@wikibon.org