RESEARCH NOTE

JSON Developers?
Oracle Autonomous Database wants you

Lowering barriers to entry

Executive Summary

Trigger

Oracle is introducing a new variation of the Autonomous Database that is geared directly to developers working with document-oriented databases based on the JSON format. JSON-based, document-oriented databases have grown popular among developers thanks to their flexible schema, developer familiarity with JavaScript and JSON, and the ubiquity of data on the Internet that is exposed as JSON in popular APIs. With Oracle pioneering autonomous databases, will the ease of use promised by self-running databases build the bridge to meet document database developers where they live?

Our Take

Oracle's approach in releasing a special-purpose variation of the Autonomous Database storing documents exposed as JSON; leveraging the ease of use associated with self-driving databases; at a competitive price point is an excellent first step toward building traction with JSON developers. Oracle is not inventing a new silo'ed platform, but instead, is repurposing a specialized subset of its converged multi-model flagship database. But this is just a first step. While Oracle is using a simplified API that will look familiar to many JSON developers, we believe that Oracle should go the last mile by adding support for the original MongoDB API that has become a de facto standard thanks to a broad spectrum third-party support.

JSON Document databases are popular

Developers love JavaScript and JSON. According to Stack Overflow, a popular online developer forum, it has been the most frequently-discussed data interchange format for most of the past decade (see Figure 1). JavaScript has long been the most popular language for designing interactivity in web browsers. Developers embraced JSON, not only because it was already based on a language that was second nature to them, but also because it provided a far more streamlined alternative to XML for describing the layout of web documents.

Document databases emerged in the early 2000s as a response to the need for managing data, such as user profiles that often had variable, nested structures that would have required significant effort to fit into a relational model. Fast forward to the present, MongoDB has catapulted to being the fifth ranked database in popularity as of this writing (August 2020), based on frequency of search engine queries, technical discussions in developer forums, job offers, and degree of mentions on social networks. By the way, those same rankings place Oracle as the Number 1 database – but the results point to the fact that while document databases are not necessarily replacing relational, they are claiming a strong and growing following, with JSON enshrined as the de facto standard data format.

Figure 1. Popular Stack Overflow data interchange format inquiries
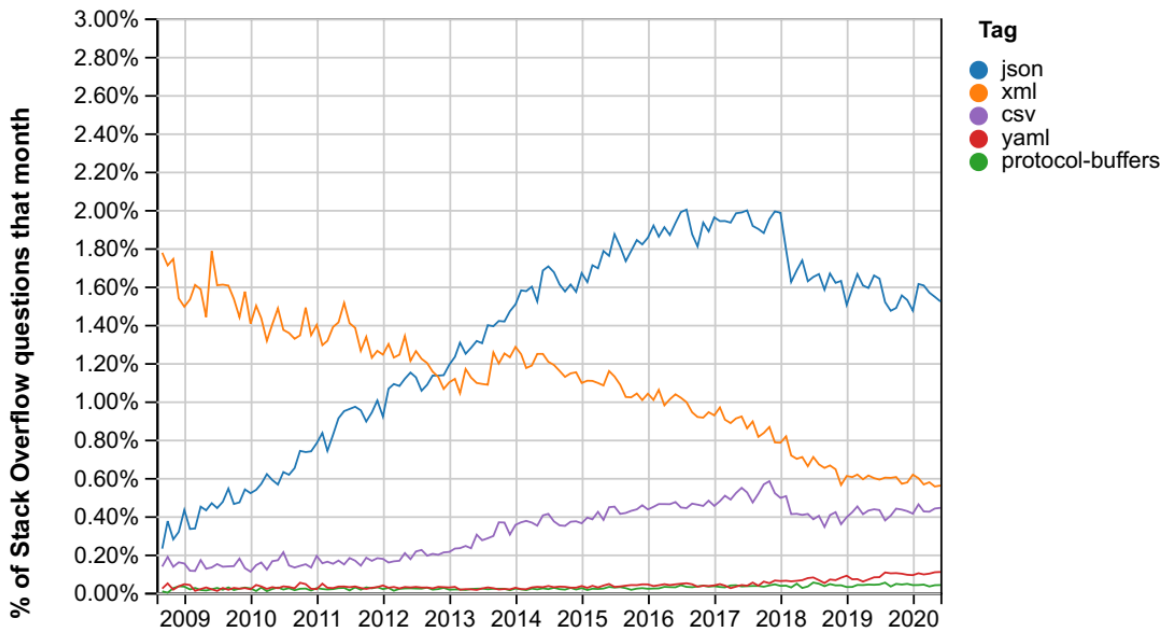
Source: Stack Overflow

Oracle is not "just" a relational database

While Oracle is typically not top of mind to document database developers, it has long had capabilities for storing and querying non-relational data types, initially with XML. Today, Oracle is positioning squarely in the multi-model camp, supporting graph, spatial, plain text, JSON documents, and key-value in addition to relational. Specifically, Oracle positions its flagship database as "converged," which is its term for multi-model. Oracle is clearly not alone; most of its relational rivals, including Db2, Teradata, and SAP HANA also have varying degrees support for non-relational data types. Admittedly, not every major database player embraces the multi-model strategy; for instance, AWS, with its portfolio of 15 databases, very much embraces the fit-for-purpose model, and this is one of the areas of differentiation that Oracle spotlights.

Oracle's challenges winning JSON developers

There are good reasons that Oracle hasn't been well-known to JSON document database developers.

Oracle has not been known for the ease of use of its developer tools – APEX is now the most prominent example of how Oracle is counteracting that. Secondly, until a couple years ago, Oracle's support for JSON did not match that of databases like MongoDB that were specifically built for JSON. Since then it has evolved from treating JSON as a variable character string to a full binary data type.

Designed for stringent enterprise requirements, JSON developers never perceived Oracle as an accessible database for relatively narrow use cases such as user profile tracking. Furthermore, compared to databases like MongoDB or other open source NoSQL databases, Oracle licenses were pricier and typically more complex.

MongoDB became the favorite for JSON developers for web applications such as user profile tracking on the strength of its intuitive tooling, its support of horizontal scaling (through sharding), and the perception that it was a simpler, more bare bones database. Not surprisingly, many used the free community edition that went viral. In fact, basic features, such as aggregation or database security that are core to enterprise databases like Oracle, came later to MongoDB because of its modest origins.

Oracle's new Autonomous JSON Database hits the "easy" button

Oracle is introducing a variation of its Autonomous Transaction Processing Database specialized for JSON. The Oracle Autonomous JSON Database is a subset, not a fork of the Autonomous Database (if the customer desires, they can expand to Autonomous Transaction Processing with a single mouse click). Autonomous JSON Database is accessed via document APIs, a command line interface, or REST. SQL is available as an *option* for querying and aggregating across collections but is not required. The document API is based on the open-source Simple Oracle Document Access (SODA). The offering is serverless, meaning customers only need specify service levels rather than nodes, and within customer SLA specs, will auto-scale. It is built on the same transactional foundation of the Oracle database. That means having the same optimizations for fast reads (avoiding linear scans) and partial updates (reducing redo/undo log sizes), providing low latency CRUD operations and full ACID consistency of the mother ship.

Oracle is pricing the new service very competitively at hourly rates that are similar to Amazon DocumentDB. We didn't compare pricing to MongoDB because its published prices are not

listed by the hour. Oracle claims to underprice MongoDB Atlas by up to 30% based on its own benchmarks, which we show in Figure 2 below.

Oracle compared the Autonomous JSON Database with 8 OCPUs vs. the result of MongoDB's own benchmarking exercise, that in turn, made the case for its superior performance over Amazon DocumentDB running on AWS M60 instances. Note that Oracle's distillation omits DocumentDB to confine its comparison to Atlas.

Oracle can price aggressively because the underlying high-performance autonomous infrastructure makes it cheap to run.

Figure 2. Oracle's benchmark
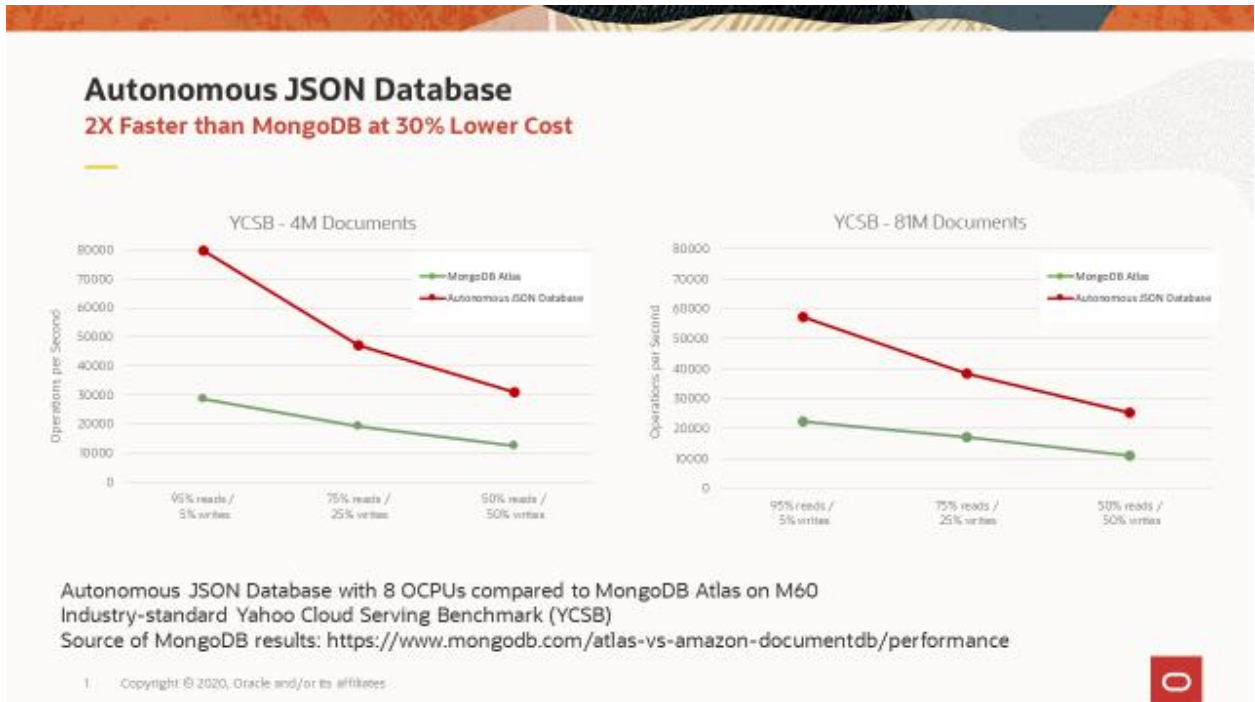
Source: Oracle

There's a good reason that Oracle is entering this market: JSON databases are popular and having a self-running database can materially differentiate it from the crowd. Outside the MongoDB orbit are Couchbase and IBM Cloudant (both of which share a common lineage).

But the MongoDB-compatible world is even more crowded because that's where the largest chunk of JSON document database developers live. Examples include Microsoft Azure Cosmos DB, Amazon DocumentDB, and Percona Server for MongoDB that all use the pre-4.0

MongoDB API. Admittedly, the API available to third parties is not the same as the newer one for MongoDB 4.x and its Atlas cloud service; but the old API supports the core features and functions of the original MongoDB platform.

Takeaways

Oracle Autonomous JSON is a good start

Oracle's introduction of the Autonomous JSON Database is a well thought-out first step for meeting document database developers because it hits several important hot buttons:

- It is a subset of the Oracle Autonomous Database, targeted at storing JSON document data, accessed via document APIs. It gives JSON database developers only as much database as they want, but without silo'ing them on a separate platform. As a subset of the full Oracle Autonomous Database, JSON development is future-proofed in case they need to upgrade or have compatibility with Oracle Databases elsewhere in the enterprise.

- It is autonomous, making it operationally simpler to use than the leading JSON document database DBaaS services, such as MongoDB Atlas. Oracle is still the only database provider to offer a fully self-running database.

- It is serverless, making it well-suited for use cases with variable or less predictable workload patterns.

- It has the same strong ACID transaction support and fast performance that customers expect from Oracle databases.

- It is priced very competitively.

Oracle presents a compelling case for JSON developers – it offers a price-friendly JSON document database that is easy to use based on an API that is similar to, but not the same as MongoDB's. Because the SODA API was developed by Oracle, we expect the Autonomous JSON Database will initially mostly appeal to Oracle shops seeking to consolidate database development under a common umbrella. But we want to see Oracle go further and add the older MongoDB AI that has become a de facto standard to broaden appeal to the bulk of document database developers.

Oh, and by the way, we believe that Autonomous JSON will be the start of more developer-focused services deployed on the Autonomous Database, and expect announcements in coming months.

Author

Tony Baer, Principal, dbInsight

tony@dbinsight.io

Twitter @TonyBaer

About dbInsight

dbInsight LLC provides an independent view on the database and analytics technology ecosystem. dbInsight publishes independent research, and from our research, distills insights to help data and analytics technology providers understand their competitive positioning and sharpen their message.

Tony Baer, the founder and principal of dbInsight, is a recognized industry expert on data-driven transformation. *Analytics Insight* named him one of the 2019 Top 100 Artificial Intelligence and Big Data Influencers. His combined expertise in both legacy database technologies and emerging cloud and analytics technologies shapes how technology providers go to market in an industry undergoing significant transformation. His regular ZDnet *"Big on Data"* posts are read 25,000 – 30,000 times monthly.