IBM z Systems
Introduction
May 2017

MongoDB and IBM z Systems

Frequently Asked Questions

Worldwide

ZSQ03083-USEN-01

MongoDB and IBM z Systems

What is MongoDB and how does it deliver value?

MongoDB is a cross-platform document-oriented database. Released under a combination of the GNU Affero General Public License and the Apache License, MongoDB is free and open-source software. Since its initial release in 2009, MongoDB has been adopted as back-end software by a number of major websites and services. Today, MongoDB is the fastest-growing database ecosystem, with over 10 million downloads, thousands of customers, and more than 1,000 technology and service partners.

Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents without fixed schemas. The schema-less nature of MongoDB not only makes it better than relational databases at handling unstructured and sparse data, it also facilitates rapid prototyping and fast evolution of programs without paying the constant costs of updating database schemas and refactoring tables.

By adopting JSON/BSON as the document format, MongoDB allows developers to write data queries in JavaScript, a language with which they are already familiar, so that the need to mix SQL statements into the application code can be eliminated. It also eliminates the cost of ORM (object-relational mapping), because JSON documents can contain complex types and are by nature JavaScript objects. Because JSON is widely supported as a de-facto standard for information exchange, it is easy to consolidate data from multiple disparate sources into one MongoDB collection.

MongoDB also supports geospatial and time-series analytics, as well as scalability and high availability through sharding and replication, making it a highly capable engine for processing the vast amount of data generated online. All these factors help make programs easier to develop and maintain, boost programmer productivity, and improve business agility.

How are clients using MongoDB?

Through our conversations with various customers, we have identified these typical scenarios in which MongoDB is used:

- Aggregating data from multiple sources into a central repository to create a single 360-degree view of enterprise customers. Complex queries can be performed over the aggregated data efficiently.
- Exploiting the flexible data model and multiple options for scaling—including range-based, hash-based and location-aware sharding—to develop applications quickly, and to support the constantly changing requirements. Schemas can be radically changed while running in production, with zero impact on the user experience.
- Creating a read-forward cache of System of Record data for fast viewing or manipulation by a front-end system, such as a web application or a mobile application, which could be in use by a large number of concurrent users.

How is IBM working with the community and MongoDB, Inc.?

IBM is in talks with MongoDB, Inc. to enable the company to introduce their enterprise-level products on IBM z Systems® and to include z Systems as a supported platform. Our development team is also working closely with the community to port bug fixes and new code changes to the platform.

What IBM products and services work with MongoDB?

MongoDB is currently available on IBM Bluemix via MongoLab, a fully-managed cloud database service featuring high availability, automated backups, web-based tools, monitoring and support. Since version 10.5, IBM DB2® has provided JSON support. One of the ways in which DB2 users can interact with JSON data is to deploy the MongoDB wire listener, which intercepts the MongoDB wire protocol and acts as a gateway in front of DB2. It allows DB2 to accept and respond to requests from applications written against the MongoDB API. Any modern language with a driver that supports the MongoDB protocol can be used, including Node.js, PHP, Python and Ruby, as well as more traditional languages such as C, C++, Java® and Perl. For more information, see *DB2 JSON capabilities, Part 1: Introduction to DB2 JSON* (http://www.ibm.com/developerworks/data/library/techarticle/dm-1306nosqlforjson1/).

How does MongoDB work with z Systems?

MongoDB 3.0.4 is now available on Linux on z Systems as a technology preview from IBM. All features are available including sharding, replication and compression. Clients can take their JSON data and create MongoDB collections on—or migrate existing ones to—the mainframe. Applications that use any of the existing MongoDB drivers work without modification. With the wealth of corporate data that either resides on or originates from the mainframe, running MongoDB on Linux® on z Systems enables faster and more secure access to data sources for aggregation purposes. The performance and virtualization capabilities of the new IBM z13™ (z13) also make it ideal for scaling out as well as scaling up NoSQL workloads.

The MongoDB port for z Systems is available as open source code. Instructions for building and installing MongoDB on z Systems are available in *Building MongoDB* (https://github.com/linux-on-ibm-z/docs/wiki/Building-MongoDB).

MongoDB drivers for various languages, such as C/C++, Node.js, and Ruby, either have been verified to work on Linux on z Systems or are being ported to the platform. IBM is working to contribute all the changes upstream to the latest version of the product.

What benefits do clients gain by running MongoDB on z Systems?

By deploying MongoDB on z Systems, a client gains all the strengths, security and reliability of the mainframe. z Systems platforms are equipped with some of the fastest general-purpose processors in the world, and they are well-known for data processing throughput. The large number of cores available in a mainframe and its high input/output bandwidth mean that MongoDB can scale up as well as scale out. The enterprise-grade virtualization capabilities of z/VM® support very high virtual machine density and resource over-commit, which allows MongoDB instances to achieve much better utilization than on other platforms. Co-located MongoDB instances (different shards, or replica set members within the same shard) can leverage HiperSockets™ to reduce network communication overhead. Proximity to enterprise data that resides on the mainframe,

such as z/OS®, allows MongoDB to perform ETL (extracting, transforming, and loading) on such data with low latency and high security.

What is IBM's competitive differentiation regarding MongoDB?

Currently, the MongoDB 3.0.4 port for z Systems is in beta-testing stage. It is fully compatible with the community version, and will be kept current with fixes that the community is delivering to the stable branch.

For more information, see *MongoDB* (https://www.mongodb.org/).

IBM Corporation
New Orchard Road
Armonk, NY 10504
U.S.A.
Produced in the United States of America,
05/2017

ZSQ03083-USEN-01