CONTENTS

OPEN MAINFRAME PROJECT

What is MongoDB?

MongoDB is an open source database that uses a document-oriented data model rather than traditional Relational Database structures. The database came to life in the mid-2000s under the NoSQL banner. Instead of using tables and rows as in relational databases, MongoDB is built on an architecture of collections and documents. Documents comprise sets of key-value pairs and are the basic unit of data in MongoDB. Collections contain sets of documents and function as the equivalent of relational database tables.

Quick Facts:

• Created by Dwight Merriman and Eliot Horowitz.

• According to Merriman, the name of the database was derived from the word ***humongous*** to represent the idea of supporting large amounts of data.

• Merriman and Horowitz helped form 10Gen Inc. in 2007 to commercialize MongoDB and related software. The company was renamed MongoDB Inc. in 2013.

• According to DB-Engines, Mongo DB ranks amongst the top 5 databases used worldwide.

MongoDB and the Mainframe

For years, the world's largest companies have run critical applications on mainframes. In fact 92 of the top 100 banks run their core mission critical data on the mainframe, as do the top retailers, airlines and government organizations. However this was mainly on IBM's own z/OS operating system with databases such as DB2 and IMS and other vendors such as CA's and their IDMS and Datacom offerings.

While traditional mainframe databases are still growing, a new dynamic has emerged over the last few years, with Linux on the mainframe moving to the mainstream. With this shift, organizations are moving beyond the traditional RDBMS offerings such as Oracle and DB2 and increasingly looking to open source options such as MongoDB.

Not only are Fortune 500 companies looking to open source, academic institutions are increasingly shifting focus, and not just for cost reasons. Academic institutions are sold on the flexibility and the open source communities that surround these solutions. The members of the Open Mainframe Project are also embracing this shift, with member organizations such as ADP, SUSE, CA, Marist College, Velocity Software, RSM Partners and IBM all seeing open source as vital to their success.

In 2013, MongoDB made the move to support Linux running on the mainframe. Organizations that require the utmost security and reliability can now build and run modern applications such as MongoDB and the tools that surround it on proven mainframe technologies. They can combine the innovative features of MongoDB with the unmatched performance of the Linux on the mainframe to create solutions with new levels of availability, security, speed, scale and flexibility.

One example of the scale of the mainframe is that the system is capable of scaling up to 8,000 virtual machines or over 1 million Docker containers on a single box.

MongoDB's NoSQL technology eliminates the overhead of object-relational mapping, allowing for developers to create and deploy modern applications rapidly, without having to define a data schema ahead of time and contend with its restrictions. Main features of MongoDB include: flexible data model, cloud and on-premise cluster management and automation, expressive query language, always-on global deployments, scalability, and performance.

CONTENTS

OPEN MAINFRAME PROJECT

Where MongoDB is being used?

MongoDB is increasingly seen as the go-to alternative for projects where traditional RDBMS options are seen as costly or too expensive, or where flexibility of the data model is seen as paramount. Enterprise deployment examples include:

• Single View: Aggregate structured and unstructured data from disparate data sources, to provide a unified, 360-degree view of enterprise information.

• Internet of Things: Collect data for the persistence to help users respond to market conditions or medical emergencies quickly.

• Combined Analytics: Combine System-of-Record (SOR) data with geospatial and sentiment analysis on news and social media, to achieve deep business insights in real time.

• Mobile: Vertically scale to meet the requirements of dealing with a huge number of mobile users and the queries they create on the backend.

Why MongoDB on Mainframe?

The architecture designed around the Integration of MongoDB and mainframe systems is optimized to provide the best performance for:

High-performance data serving: The system capacities available for MongoDB on the mainframe enable the MongoDB engine with the ability to handle billions of interactions. Servers running Node.js and MongoDB can handle over 30 billion web events per day. The popular MEAN stack runs up to 2x faster than on other platforms.

Better data consistency and reduced overhead: Mainframe systems allow MongoDB to scale vertically with capacity, instead of horizontally by sharding and replicating the database, which ensures that critical data remains consistent and minimizes sharding-related overhead.

Security and resilience = Trusted operations: Mainframe systems achieve availability and deliver good response times even when the system is at its full utilization capacity with many mainframes running constantly at 90%+ utilization. The system also protects databases with the highest level of security accreditation namely, EAL5+. EAL5+ ensures that workloads and data is isolated at every level and that the mainframe delivers the most robust platform for security conscious workloads such as mission critical databases.

On-chip cryptography acceleration and advanced encryption technology built into the platform efficiently protects sensitive data—both in-flight and at rest. This ensures that mission critical data is protected at every level.

Scaling/performance relative to x86

On x86, MongoDB relies on horizontal scaling, which comes with risks such as higher latency for aggregate queries and a lower level of data consistency, and the size of each shard is limited to the size of the servers. Sharding MongoDB on x86 also means:

• More effort required for Extract, Transform and Load (ETL) due to structured and unstructured data residing in different databases.
• Increased developer time and effort is required to use a sharded database. Once a set of data is sharded, it is hard to change the shard key.
• Weak consistency and durability guarantees due to update conflicts between shards or “split brain” situations.
• Run-time overhead in aggregate queries caused by the need to collect results from multiple shards, via high-latency network links. Shard balancing also causes data migration between shards, adding even more overhead.
• Potential security implications because enterprise data is sent across the network.
• High operational costs to design and maintain a distributed cluster due to the number of servers involved, and multiple points of failures.

CONTENTS

OPEN MAINFRAME PROJECT

On the mainframe, to grow a MongoDB database the operational tasks are radically simplified, and extend to the following.

• Dynamically add cores, memory, I/O adapters, devices and network cards, and grow **without disruption** to running environment.

•Provision for peak utilization, unused resources automatically reallocated after peak.

Installing MongoDB on Linux

IBM and MongoDB have done a lot of work to support MongoDB on Linux on the mainframe, to find out more the primary resource can be found here: http://www.w3resource.com/mongodb/installation-Linux.php

If you want test out Mongo DB, check out this pre-built Docker container.

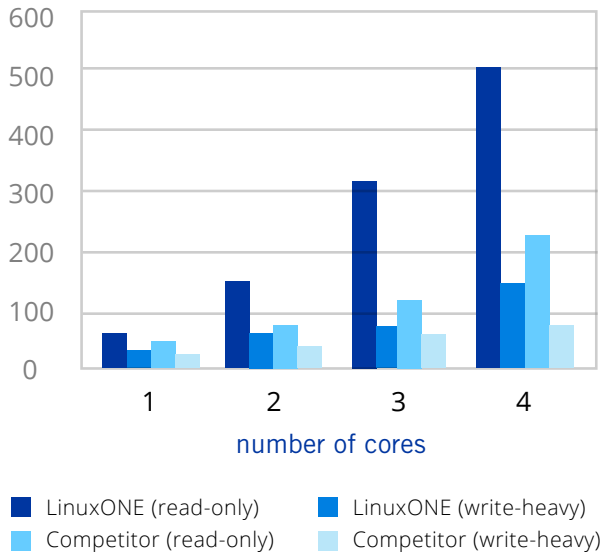
Another key question is which Linux distributions is MongoDB available on for the mainframe? The simple answer is currently:

• SLES 11
• SLES12
• RHEL6
• RHEL7
• Ubuntu 16.04

What Can you Expect?

Typical performance of MongoDB on mainframe is up to 2.1x better throughput than x86 alternatives according to IBM benchmarks. MongoDB also scales up to 2TB with sustained throughput and < 5ms response time, while serving 4+ billion documents, at 460,000 reads/writes per second, with no sharding again according to IBM internal benchmarks. Examples of IBM benchmark results on their LinuxONE servers can be seen below with the YCSB benchmark being used in these cases.

Yahoo! Cloud Serving Benchmark (YCSB) on MongoDB



AcmeAir Throughput vs Data Size in MongoDB