Benchmarking a Content Services Platform at 11 Billion Documents

→ nuxeo.com

Over the past several years, Nuxeo has conducted a number of large-scale benchmark exercises of the Nuxeo Platform and published the results. This paper explores why we benchmarked the Nuxeo Platform with more than 10 billion documents, why we used a phased approach to this exercise, and key takeaways for enterprise organizations looking to scale out their content-centric applications.

It started at 1 Billion Documents

In 2014, we published our first benchmark, which was done with a single application using ten different repositories using ten PGSQL servers. The tests showed that the responsiveness and performance of the Nuxeo Platform remains strong, thanks to Nuxeo's scale-out architecture.

We ran a similar benchmark using a single repository leveraging a sharded MongoDB cluster..

In 2019, we performed a benchmark study focused on Digital Asset Management. For this exercise, we focused on uploading and downloading large, multimedia files. To allow high-throughput conversions, we successfully scaled worker nodes in the process.

We also perform regular benchmarks as part of our continuous integration chain to monitor how the performance of the Nuxeo Platform evolves. You can read more about the results here.

It's Nuxeo's philosophy to be fully transparent about its benchmark results, both to inform our customers about what's possible on the platform and to share best practices to optimize the system when scaling your solutions and the platform itself.

Setting Higher Goals

When we considered doing a more ambitious benchmark this year, we agreed upon three objectives:

1. Benchmark the platform at ten billion documents. As Nuxeo customers approach this milestone, we wanted to better understand what resources are required and how to configure the Nuxeo Platform to support these high-volume scenarios.

2. Define the architecture blueprints needed to deploy Nuxeo implementations that contain one billion, two billion, three billion, or ten billion documents inside the Nuxeo Cloud as a deliverable for our customers.

3. Develop a benchmark architecture to continuously test and measure Nuxeo Platform performance as the platform itself evolves.

Unlike academic benchmark exercises, we wanted to make the benchmark scenario as realistic as possible. As a result, we are not simply documenting that Nuxeo Platform met certain milestones; rather, we are documenting and sharing details and the learnings from this real-world scenario. This information should be useful in determining architectural strategies, calculating resource requirements, and understanding the best practices for deploying high-volume implementations.

To achieve these goals, we added the following constraints:

- We used a real production infrastructure environment, as we did not want to base our results on test laboratory conditions that could not be reasonably reproduced in a production environment.

- We tested our scalability with real-world use cases and a meaningful set of documents. Again, we wanted to certify our results in realistic conditions and to produce meaningful advice for customers.

-

Production Environment

We leveraged our Nuxeo Cloud infrastructure and automation tools to deploy the Nuxeo application following exactly the same procedures that we follow for our own customers. This meant that we deployed the Nuxeo Platform using Docker containers on EC2. We leveraged PaaS services to support the Nuxeo implementation, including AWS Elasticsearch, AWS MSK for Kafka, AWS S3 for blob storage, and MongoDB Atlas for the database.

We deployed the entire implementation using existing automation, including Terraform for AWS automation. We relied upon the Nuxeo Configuration template system to configure our implementation and leveraged the Nuxeo Cloud build pipeline to create the custom Docker images. We also complied with the following security and production rules:

- Everything would be secured and encrypted, including all communication being TLS-encrypted and encryption at REST for all storage.
- All security systems would be enabled, including antivirus and IDS.
- We relied upon DataDog for monitoring, and we modified the existing dashboard to add metrics to be monitored..

A Two-Phase Project

We identified two distinct phases for this project:

Phase One – maximizing a single repository

In the first phase, we focused on scaling up a single-repository instance of our Nuxeo Cloud service. Simply put, this means one database, without any sharding, one set of indices, and one object store. The goal of this first phase was to identify any bottlenecks to performance that impact ingestion or general usability.

Our objective was to identify key components and metrics that should be actively monitored; we also wanted to determine what potential bottlenecks might exist and when to scale out or up to maintain performance. We conducted performance testing along the way, at one billion, two billion, and finally at three billion documents. An important output of this benchmark are reference architectures for single-repository systems running on AWS. We documented these architectures, with requisite hardware, configurations, and expected performance.

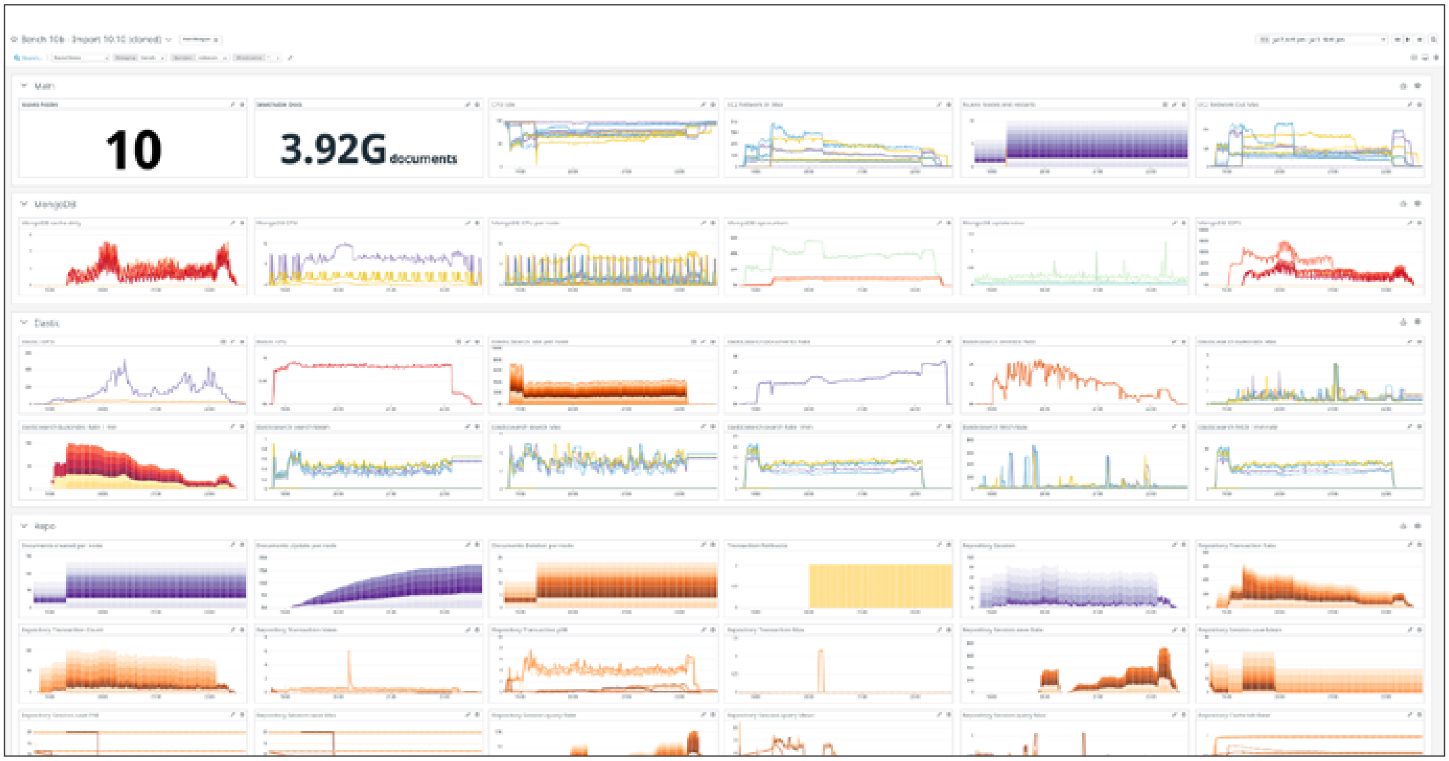

Figure 1 – Phase 1 Dashboard

By the way, for many years now, we have extolled the virtues of NoSQL and MongoDB. As a dramatic proof point, yes, a single instance of a NoSQL database will scale to 3B documents (and beyond with sharding). And for those who might ask, did we stop at three billion documents for any particular reason? The answer is no. Three billion documents isn't, by any means, a hard limit for a single-repository system. We just felt that this number was sufficient for customers that would look to scale out a single-repository system.

Phase Two – meeting the 10 billion document challenge

To meet the high-volume requirements of the benchmark, we took a different approach in the second phase. We utilized a multi-repository system that accommodated the requirements of large, global customers. Specifically, we deployed three separate repositories:

1. An active, geographic repository (US West)
2. A second, active geographic repository (US East)
3. An archival repository

If you think about this from a global customer perspective, this is a good representation of how they might partition a system. One, they may need to establish different regional repositories or repositories based on different lines of business. Two, they may want to begin to partition active data from archival data and house this information in separate repositories and lower-cost, less-performant storage tiers.

Each repository had its own database instance (MongoDB Atlas), search indices (Elasticsearch) and object store (Amazon S3). While we do support Amazon Glacier, we didn't use a deep archiving service for this exercise. The system supports search across all three repositories seamlessly from within the Nuxeo user interface.

As with phase 1, we defined specific deliverables. Our target with phase 2 was to deliver a series of benchmark results that illustrated the performance of Nuxeo Cloud in scaling up to ten billion documents, identifying the impact of database sharding and multiple repositories.

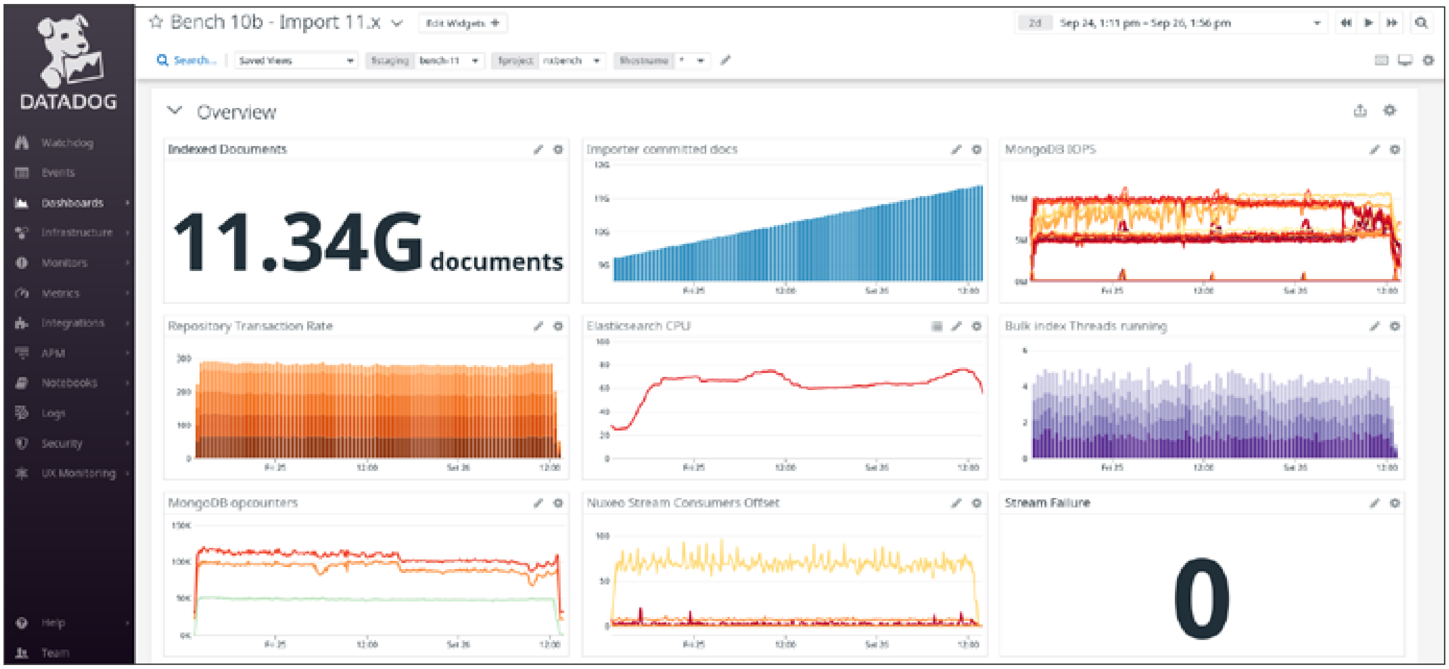

Figure 2 – Phase 2 Dashboard

Testing the application – not the storage service

As we stated earlier, the goal for this exercise was not simply an academic one or to test the cloud and storage infrastructure. The goal was to provide meaningful insights and best practices for Nuxeo customers that are building large-scale information management solutions.

To that end, our objective was to test the conditions that our customers would face when building content-centric business applications. We started with interviewing several of our customers to better understand their requirements and the challenges that they expected as they scale their implementations.

To that end, we constructed scenarios that covered document management use cases that are common for our enterprise customers, especially in the financial services industry.

To support these use cases, we included customer IDs (images), statements, account documents, and even customer correspondence.

In addition to scaling storage to support multiple billions of documents, additional benchmark requirements included:

- **Cost-efficiency:** the solution should be flexible to ensure that costs do not scale proportionately with document volume.
- **Full support for content functionality:** to ensure the results were meaningful, the benchmark included full support for expected content services features, including metadata, search indexing, security controls, and more.
- **Support real-world requirements:** to ensure that the benchmark provided meaningful information for large-scale implementations, we tested for several real-world situations, including mixed workloads (e.g., bulk import simultaneous with daily usage, ability to fully reindex, bulk operations across large numbers of documents).

Key Takeaways

In summary, this benchmark exercise successfully loaded 11.34 billion documents, with metadata, without degrading system performance.

We confirmed that for volumes up to 3 billion documents, we were able to configure the Nuxeo system with a single repository. During this first phase, we were also able to provide insight into when configuration changes, such as adding more Nuxeo worker nodes or scaling up Elasticsearch clusters to support bulk import or re-indexing.

For larger systems, we were able to confirm support for over 11 billion documents in a multiple repository system. For cost and performance reasons, we found that implementing the larger system with multiple repositories was a better approach: not only is it more cost-efficient, customers that provided input agreed that partitioning active content from archival content was the right approach. While most customers do not have multiple billions of documents that they work on regularly, they do need these billions of documents to be fully searchable and available when they need them. By splitting content across three repositories, we maintained strong performance while still providing consistent navigation and search functionality.

In addition, we determined best practices for a number of configuration decisions, including when and how to shard the MongoDB database to optimize infrastructure costs, and what Elasticsearch strategies and configurations enable the system to scale up without dramatically increasing costs. In our tests, for example we were able to support scaling from one billion to over 11 billion without doubling our Elasticsearch hardware configuration.

Finally, using AWS Elasticsearch and MongoDB Atlas PaaS services allowed us to resize the infrastructure dynamically – without affecting system availability – as we added more and more content to the system.

Below are some key takeaways from this exercise::

- The Nuxeo Platform scaled to support over 11 billion documents without degrading key performance metrics.
- Insight from this exercise has resulted in a reference sharding architecture for our customers' use, as well as corresponding guidance and best practices for scaling out a multi-repository system.
- Nuxeo has established a documented process for continuously benchmarking the platform's performance.

If you'd like to see for yourself how Nuxeo Cloud performs with over ten billion documents, please contact us to schedule a demo.