EMC XTREMIO ARRAY FOR VIRTUALIZED MONGODB

EMC Solutions

Abstract
This guide describes a solution that combines the MongoDB NoSQL database with the EMC® XtremIO® array in a virtualized environment. It also describes the XtremIO building-block design that enables scaling for different types of MongoDB workloads.

August 2016
Chapter 1  Introduction  
Business case ............................................................................................................ 6
Solution overview ....................................................................................................... 6
Key results .................................................................................................................. 7
Document purpose ..................................................................................................... 8
Audience .................................................................................................................... 8
We value your feedback! ............................................................................................. 8

Chapter 2  Technology Overview  
Solution architecture ................................................................................................ 10
Key components ....................................................................................................... 11
Hardware resources .................................................................................................. 13
Software resources ................................................................................................... 13

Chapter 3  Design Considerations and Best Practices  
Storage design ......................................................................................................... 15
Virtualization design ................................................................................................ 16
MongoDB design ...................................................................................................... 17

Chapter 4  Test Tools  
YCSB overview .......................................................................................................... 21
YCSB configuration ................................................................................................... 21

Chapter 5  Test Cases and Results  
Overview .................................................................................................................. 23
Virtualized MongoDB on XtremIO building blocks ..................................................... 24
XVCs for MongoDB .................................................................................................. 30
XtremIO compression with MongoDB Advanced Compression ............................. 32
XtremIO rapid recovery for MongoDB replica set ...................................................... 34

Chapter 6  Conclusion  
Summary ................................................................................................................... 36
Findings .................................................................................................................... 36
Chapter 7  References  37
EMC documentation .................................................................................................38
VMware documentation ............................................................................................38
Other documentation ...............................................................................................38
This chapter presents the following topics:

**Business case** ............................................................................................................ 6

**Solution overview** ...................................................................................................... 6

**Key results** ................................................................................................................. 7

**Document purpose** ..................................................................................................... 8

**Audience** .................................................................................................................... 8

**We value your feedback!** ............................................................................................ 8
Chapter 1: Introduction

Business case

EMC customers are experiencing a more urgent need to capture and store large volumes of data that they want to analyze for patterns, trends, and associations, especially in areas related to human behavior and interactions. These large and complex datasets are frequently referred to as Big Data. Highly critical relational data that is less than 20-30 TB and that is unstructured or complex, or both, can be difficult to manage and analyze without suitable tools and processes.

For these types of environments, MongoDB is one of the fastest-growing NoSQL databases. MongoDB has been downloaded more than 10 million times and has more than 2,000 customers, including over one-third of the Fortune 100 companies.

MongoDB databases are widely deployed in several key vertical markets such as finance and high technology. Companies in these industries deploy complex, high-performance, multinode clusters that require rapid response times. Therefore, many MongoDB deployments today run on solid-state drives (SSDs) in a direct-attached storage (DAS) model.

As IT organizations increasingly deploy MongoDB databases at scale, SSDs in a DAS model present challenges in terms of higher infrastructure and operational costs, and the inability to meet service-level agreements (SLAs).

Many organizations have been taking advantage of virtualization to reduce server cost and gain operational advantages without losing any performance. When it comes to business-critical workloads such as those that are managed by MongoDB, migration to virtualization has demonstrated advantages.

This solution shows the benefits of combining MongoDB with the EMC® XtremIO® all-flash array to enable high performance in a virtualized NoSQL database environment.

Solution overview

This solution delivers high performance with the deployment of a highly efficient SSD storage system that naturally complements the flexible and predictable scale-out architecture of a virtualized MongoDB configuration. XtremIO Virtual Copy (XVC) and disaster-recovery technology greatly simplify MongoDB backup and recovery.

To demonstrate the value of the solution, we use multiple virtualized servers with MongoDB databases running Yahoo Cloud Serving Benchmark (YCSB) with random I/O workloads on an XtremIO and VMware vSphere infrastructure.

We tested and validated the following scenarios for this solution:

- Virtualized MongoDB on XtremIO X-Brick® building blocks
- XVCs for virtualized MongoDB
- XtremIO compression with MongoDB Advanced Compression
- XtremIO rapid recovery of a MongoDB replica set
Key results

This solution provides predictable performance, efficient XVCs, inline deduplication, and compression in a MongoDB virtualized environment. Key test results are as follows:

- The XtremIO all-flash array can support consistent and predictable scale-out performance in a virtualized MongoDB environment. The end-to-end latency can be expected to remain constantly low. Table 1 details the expected performance for common MongoDB workload types with storage configurations of one to four X-Brick building blocks.

<table>
<thead>
<tr>
<th></th>
<th>1 X-Brick configuration</th>
<th>2 X-Brick configuration (expected performance)</th>
<th>4 X-Brick configuration (expected performance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query-dominated</td>
<td>Up to 55,251 operations per second (ops/sec)</td>
<td>Up to 110,502 ops/sec</td>
<td>Up to 221,004 ops/sec</td>
</tr>
<tr>
<td></td>
<td>Up to 2.1 ms average storage latency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query/modify-dominated</td>
<td>Up to 23,778 ops/sec</td>
<td>Up to 47,556 ops/sec</td>
<td>Up to 95,112 ops/sec</td>
</tr>
<tr>
<td></td>
<td>Up to 3.2 ms average storage latency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert-dominated</td>
<td>Up to 18,294 ops/sec</td>
<td>Up to 36,588 ops/sec</td>
<td>Up to 73,176 ops/sec</td>
</tr>
<tr>
<td></td>
<td>Up to 2.4 ms average storage latency</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The profiles in Table 1 are the results of the MongoDB server configuration including the number of CPU cores and the amount of memory. The expected performance does not represent the limitations or capabilities of an XtremIO configuration.

- Virtualized MongoDB with the XtremIO array is agile and easily managed. vSphere 6.0 enables easy scale out and provisioning of the MongoDB environment. Compared with bare-metal MongoDB deployments, the configuration in this solution results in no obvious performance degradation.

- XVCs provide agile and simplified MongoDB backups with minimal storage footprint and minimal performance impact. When testing this solution, we observed less than 1 percent performance difference. The XVC feature also consolidates the test/development and production environment in the one platform to improve total cost of ownership (TCO).

- XtremIO inline data reduction, including compression, deduplication, thin provisioning, and space-saving XVCs, drives extraordinary data efficiencies with a reduced TCO. XtremIO inline compression can offload 6 percent CPU utilization of the MongoDB server.

- Using XVC technology provides rapid and simplified disaster recovery for the MongoDB replica set. All the data is automatically synchronized among replica nodes.
Chapter 1: Introduction

Document purpose

This guide describes a solution that combines MongoDB with the XtremIO array to deliver a high-performing NoSQL database in a virtualized environment. This guide:

- Introduces the technical components that are required to implement and operate the solution
- Describes the testing and validation of the solution functionality
- Evaluates the technical and business value of the solution

Audience

This guide is intended for NoSQL database administrators, storage administrators, virtualization administrators, system administrators, IT managers, and any others who evaluate, acquire, manage, maintain, or operate database environments.

We value your feedback!

EMC and the authors of this document welcome your feedback on the solution and the solution documentation. Contact EMC.Solution.Feedback@emc.com with your comments.

Authors: Eric Wang, James Shen, Tao Guo, Kirankumar Bhusanurmath, Vincent Shen, Karen Johnson
This chapter presents the following topics:

Solution architecture ........................................................................................................10
Key components .................................................................................................................11
Hardware resources ............................................................................................................13
Software resources ............................................................................................................13
Solution architecture

Overview

This solution includes a validated architecture for MongoDB on the VMware vSphere ESXi 6 virtualization platform and XtremIO all-flash array. The solution configuration provides predictable scale-out performance, XVC efficiency, inline deduplication, inline compression, and disaster recovery capability for MongoDB.

Solution architecture

Figure 1 shows the architecture for the solution.
We deploy multiple MongoDB replica sets and one MongoDB standalone instance over a three-host vSphere High Availability (HA) cluster. We implement each replica set with one primary node and two secondary nodes. The workloads are evenly distributed between two ESXi hosts. We deploy the YCSB Clients in the third ESXi host in the vSphere cluster. For the storage network, we use a dual-port 8 Gb Fibre Channel (FC) link for both optimal performance and high availability.

**Key components**

**EMC XtremIO all-flash array**

The XtremIO all-flash storage array is designed to unlock the full performance potential of flash and deliver array-based capabilities that take advantage of the unique characteristics of SSDs.

The system uses industry-standard components and proprietary intelligent software to deliver performance that ranges from hundreds of thousands to millions of IOPS and consistent low latency. It is also designed to require minimal planning, with an easy-to-use interface for provisioning and managing the array.

The XtremIO all-flash array delivers value across the following main dimensions:

- **Performance**—Regardless of how busy the system is and regardless of storage capacity utilization, latency and throughput remain consistently predictable and constant.

- **Scalability**—The XtremIO array is based on a scale-out architecture. The system begins with a single building block, called an X-Brick. When you require additional performance and capacity, you can scale out the system with additional X-Brick building blocks. Performance scales linearly, with two X-Brick blocks supplying approximately twice the IOPS and four X-Brick blocks supplying approximately four times the IOPS of the single X-Brick configuration. Latency remains consistently low as the system scales out.

- **Efficiency**—The core engine implements content-based inline data reduction. The XtremIO array automatically reduces (deduplicates and compresses) data as it enters the system, reducing the amount of data that is written to flash, improving longevity of the media, and driving down cost. The XtremIO array allocates capacity to volumes on demand in granular data blocks. Volumes are always thin provisioned without any loss of performance, over-provisioning of capacity, or fragmentation. Implementing content-based inline deduplication compresses the remaining data even further, reducing the number of writes to the flash media. The data compression is carried out inline on the deduplicated (unique) data blocks.

**VMware vSphere**

The vSphere virtualization platform increases server utilization so that a business can consolidate its servers and spend less on hardware, administration, energy, and floor space. The components of particular importance in this solution are ESXi and VMware vCenter.
vSphere ESXi

vSphere ESXi is a bare-metal hypervisor. It installs directly on a physical server—the ESXi host—and partitions that server into multiple virtual machines.

vSphere ESXi hosts and their resources are pooled into clusters that contain the CPU, memory, network, and storage resources that are available for allocation to the virtual machines. Clusters scale up to a maximum of 32 hosts and can support thousands of virtual machines.

vSphere vCenter

VMware vCenter Server is management software that runs on a virtual or physical server to oversee multiple ESXi hypervisors as a single cluster. An administrator can interact directly with vCenter Server or use vSphere Client to manage virtual machines from a browser window from any location. For example, an administrator can capture the detailed blueprint of a known, validated configuration that includes networking, storage, and security settings, and then deploy that blueprint to multiple ESXi hosts.

EMC PowerPath/VE

EMC PowerPath®/VE software provides intelligent, high-performance path management with path failover and load balancing optimized for EMC and selected third-party storage systems.

PowerPath/VE software works with ESXi as a multipathing plug-in (MPP) that provides path management to hosts. It is installed as a kernel module on the vSphere host. PowerPath/VE plugs into the vSphere I/O stack framework to provide advanced multipathing capabilities, including dynamic load balancing and automatic failover, to the vSphere hosts.

MongoDB

MongoDB is a cross-platform document-oriented database that is classified as a NoSQL database with dynamic schemas.

MongoDB replica set

A MongoDB replica set is a group of mongod processes that maintain the same dataset. Replica sets provide redundancy and high availability and are the basis for all production deployments. Replication provides redundancy and increases data availability. With multiple copies of data on different database servers, replication provides a level of fault tolerance against the loss of a single database server.
Hardware resources

Table 2 lists the hardware resources that are used in this solution.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Quantity</th>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage array</td>
<td>1</td>
<td>XtremIO array with one X-Brick building block (7.58 TB of usable physical space)</td>
<td>Volumes that are required for this solution</td>
</tr>
<tr>
<td>Servers</td>
<td>3</td>
<td>20 Intel 3.0 GHz processors, 512 GB RAM, including a 1 x 1 Gigabit Ethernet (GbE) network interface card (NIC)</td>
<td>ESXi hosts for virtual machines</td>
</tr>
<tr>
<td>LAN switches</td>
<td>2</td>
<td>1 GbE</td>
<td>Server connections</td>
</tr>
<tr>
<td>SAN switches</td>
<td>2</td>
<td>8 Gb FC links</td>
<td>Server-to-storage connections</td>
</tr>
</tbody>
</table>

Software resources

Table 3 lists the software resources that are used in this solution.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoDB</td>
<td>3.2</td>
<td>Database for all virtual machine servers</td>
</tr>
<tr>
<td>Red Hat Linux</td>
<td>6.6 (64-bit)</td>
<td>Operating system (OS) for MongoDB servers and YCSB servers</td>
</tr>
<tr>
<td>EMC XtremIO OS (XIOS)</td>
<td>4.0.0-58</td>
<td>OS</td>
</tr>
<tr>
<td>VMware vCenter ESXi</td>
<td>6.0.0 u1</td>
<td>Hypervisor</td>
</tr>
<tr>
<td>VMware vCenter Server</td>
<td>6.0.0 u1</td>
<td>Manager of virtual machine cluster</td>
</tr>
<tr>
<td>EMC PowerPath/VE for VMware</td>
<td>6.0.1</td>
<td>Multipathing software optimizing host I/O to XtremIO array</td>
</tr>
<tr>
<td>YCSB for MongoDB</td>
<td>0.7.0</td>
<td>MongoDB benchmark tool</td>
</tr>
</tbody>
</table>
Chapter 3: Design Considerations and Best Practices

This chapter presents the following topics:

- **Storage design** ................................................................. 15
- **Virtualization design** ...................................................... 16
- **MongoDB design** ............................................................ 17
Storage design

XtremIO connectivity

In this solution, we deploy the XtremIO array in a single X-Brick configuration, which provides 7.58 TB of usable physical space.

We use the best practices for enabling a host and the XtremIO single X-Brick cluster to communicate, using four paths per device. Figure 2 displays the logical connection scheme for the four paths.

![Figure 2. Logical connection scheme for single X-Brick cluster](image)

Two storage controllers are on one X-Brick cluster, and each storage controller has two ports connecting to different switches. Therefore, two links go from one storage controller through different switches to the host. This link redundancy provides the tolerance of one-point switch failure.

In this design, the host can see four paths to the XtremIO array. To manage the paths, we install multipath software, which also provides I/O balancing and optimization. For multipath management, the array supports the vSphere Native Multipathing Plug-In (NMP) technology or, as deployed in this solution, PowerPath/VE.

Storage resource configuration

When creating volumes in XtremIO for a vSphere host, we consider the following best practice: We set the logical block (LB) size on XtremIO as **Normal (512 LBs)**, as shown in Figure 3, because 512 bytes is the only LB size that is supported by vSphere for presenting to ESX.

![Figure 3. Logical block size supported by vSphere](image)

In this solution, we create the thin provisioning volumes on the XtremIO array as shown in Table 4. XtremIO allocates capacity to volumes on demand in granular data blocks.
Table 4. Storage volumes on XtremIO

<table>
<thead>
<tr>
<th>Role</th>
<th>Quantity</th>
<th>Size (TB)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual machine OS volume</td>
<td>1</td>
<td>2</td>
<td>Contains all the virtual machine operating systems</td>
</tr>
<tr>
<td>MongoDB database volume</td>
<td>16</td>
<td>1</td>
<td>Consists of database for one standalone MongoDB instance and five 3-node MongoDB replica clusters</td>
</tr>
</tbody>
</table>

Virtualization design

Virtual machine configuration

In this solution, we install ESXi 6.0 on three physical servers to set up a vSphere HA cluster. We use two servers to host the virtual machines that run the MongoDB environment and one server to host the virtual machines that run the YCSB testing tool.

Table 5 lists the ESXi hosts and their respective virtual machines.

Table 5. Virtual machine hosting

<table>
<thead>
<tr>
<th>Physical host</th>
<th>Number of virtual machines</th>
<th>Virtual machine roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESXi host 1</td>
<td>7</td>
<td>• 1 MongoDB standalone&lt;br&gt;• 2 MongoDB replica sets</td>
</tr>
<tr>
<td>ESXi host 2</td>
<td>9</td>
<td>3 MongoDB replica sets</td>
</tr>
<tr>
<td>ESXi host 3</td>
<td>11</td>
<td>11 YCSB Client servers</td>
</tr>
</tbody>
</table>

Table 6 lists the specifications of the MongoDB virtual machines that are used in this solution.

Table 6. Virtual machine specifications for MongoDB

<table>
<thead>
<tr>
<th>Role</th>
<th>Quantity</th>
<th>vCPU</th>
<th>Memory (GB)</th>
<th>Boot disk (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoDB standalone</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>50</td>
</tr>
<tr>
<td>MongoDB replica sets</td>
<td>15 (5 * 3)</td>
<td>8</td>
<td>8</td>
<td>50</td>
</tr>
</tbody>
</table>

Virtualization configuration

In this solution, we implement the following EMC and VMware best practices to provide the optimal performance for all the MongoDB virtual machines running on ESXi hosts:

- Install PowerPath/VE to provide efficient path management and load balancing.
- Create a vSphere HA cluster to provide a virtualized, high-availability MongoDB environment that is easy to use and cost effective.
- Use a single virtual socket for each virtual machine. With virtual Non-Uniform Memory Access (NUMA) topology, a single virtual socket that has fewer virtual
CPU cores than the physical CPU cores of a socket in the physical ESXi host is recommended.  

- Use a VMware Paravirtual SCSI controller to increase throughput with significant CPU utilization reduction in the SAN environment.  
- Use a VMware VMXNET3 network adapter to optimize network performance.  
- Use Thick Provision Eager Zeroed disk provisioning to optimize virtual-disk performance on the XtremIO array.  
- Set the ESXi host parameters as follows to maximize the MongoDB performance on the XtremIO array:
  - `lpfc_lun_queue_depth=128`
  - `Disk.SchedQuantum=64`
  - `Disk.DiskMaxIOSize=4096`

For detailed information about these best practices, refer to *Performance Best Practices for VMware vSphere 6.0* and the *EMC XtremIO Storage Array Host Configuration Guide*.

### MongoDB design

**MongoDB overall design**

This solution employs the XtremIO array to support a MongoDB environment that can meet enterprise demands for performance, scalability, and data replication.

We create six sets of databases including one standalone and five replica sets to simulate the real-world environment setup and provide sufficient workload on the XtremIO storage array.

All MongoDB configurations are maintained in a configuration file under `/etc/mongo.conf`. The database is maintained at `/var/lib/mongod`, where each MongoDB node is provisioned 300 GB of storage space from the storage array.

In this solution, we apply the following best practices for MongoDB:

- A 64-bit MongoDB build is used, and the MongoDB dbPath directory corresponds to the configured storage engine.
- MongoDB journaling is enabled to ensure that mongod can recover its data files and keep the data files in a valid state after any failure.
- The HTTP interface is disabled in this MongoDB environment. Linux with the ext4 file system is used.
- Sufficient space is provided for MongoDB journaling and the operations log (oplog).

We create 16 virtual machines to host MongoDB instances. We provision each virtual machine with two virtual networks of the standard port group type, one of which is a management network that is connected to a 1 GB full-duplex physical adapter. The other is a virtual-machine port group that is connected to a 10 GB full-duplex physical adapter for high network throughput.
Chapter 3: Design Considerations and Best Practices

**MongoDB design considerations for disaster recovery**

The XtremIO storage array provides efficient and instant XVCs for MongoDB databases. EMC recommends that you run the `db.fsyncLock()` command each time before you take an XVC to avoid data loss, and release the lock by running the `db.fsyncUnlock()` command after you take the XVC.

EMC recommends that you run the `db.collection.validate()` command during disaster recovery when the XVC is mounted to the new MongoDB node in place of a failed MongoDB node. Doing so enables the mongod process to recheck the physical data structure and maintain data consistency by refreshing the metadata and keeping it synchronized with other nodes in the replica set.

**XtremIO building blocks**

This solution is enabled by the predictable scale-out performance of the X-Brick building block of the XtremIO array.

Table 7 shows the performance for a MongoDB query-dominated workload.

**Table 7. Performance for MongoDB query-dominated workload on XtremIO**

<table>
<thead>
<tr>
<th>MongoDB workload type</th>
<th>Performance metric</th>
<th>1 X-Brick configuration</th>
<th>2 X-Brick configuration (expected performance)</th>
<th>4 X-Brick configuration (expected performance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query-dominated</td>
<td>XtremIO IOPS</td>
<td>Up to 64,398</td>
<td>Up to 136,483</td>
<td>Up to 280,653</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Up to 72,085</td>
<td>Up to 144,170</td>
<td>Up to 288,340</td>
</tr>
<tr>
<td></td>
<td>MongoDB ops/sec</td>
<td>Up to 49,852</td>
<td>Up to 105,103</td>
<td>Up to 215,605</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Up to 55,251</td>
<td>Up to 110,502</td>
<td>Up to 221,004</td>
</tr>
<tr>
<td></td>
<td>XtremIO average latency (ms)</td>
<td>Up to 2.1 ms average storage latency</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of ESXi hosts</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

We use YCSB to simulate different types of MongoDB workloads—query-dominated, query/modify-dominated, and insert-dominated—to mimic the majority of real-world scenarios.

We scale out the topology by deploying multiple MongoDB replica sets to achieve the maximum IOPS and ops/sec.

The ops/sec values in Table 7 do not represent the limits of the whole system but are the outcome of the following assumptions:

- The configuration of each MongoDB virtual machine includes eight virtual cores and 8 GB memory, as specified in Virtual machine configuration.
- The configuration of each ESXi host includes 20 CPUs and 512 GB memory, as specified in Hardware resources.
Table 8 and Table 9 respectively show the performance for a query/modify-dominated workload and an insert-dominated workload.

**Table 8. Performance for MongoDB query/modify-dominated workload on XtremIO**

<table>
<thead>
<tr>
<th>MongoDB workload type</th>
<th>Performance metric</th>
<th>1 X-Brick configuration</th>
<th>2 X-Brick configuration (expected performance)</th>
<th>4 X-Brick configuration (expected performance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query/modify-dominated</td>
<td>XtremIO IOPS</td>
<td>Up to 52,516</td>
<td>Up to 105,032</td>
<td>Up to 210,064</td>
</tr>
<tr>
<td></td>
<td>MongoDB ops/sec</td>
<td>Up to 23,778</td>
<td>Up to 47,556</td>
<td>Up to 95,112</td>
</tr>
<tr>
<td></td>
<td>XtremIO average latency (ms)</td>
<td>Up to 3.2 ms average storage latency</td>
<td>Up to 3.2 ms average storage latency</td>
<td>Up to 3.2 ms average storage latency</td>
</tr>
<tr>
<td></td>
<td>Number of ESXi hosts</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 9. Performance for MongoDB insert-dominated workload on XtremIO**

<table>
<thead>
<tr>
<th>MongoDB workload type</th>
<th>Performance metric</th>
<th>1 X-Brick configuration</th>
<th>2 X-Brick configuration (expected performance)</th>
<th>4 X-Brick configuration (expected performance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert-dominated</td>
<td>XtremIO IOPS</td>
<td>Up to 42,144</td>
<td>Up to 84,288</td>
<td>Up to 168,576</td>
</tr>
<tr>
<td></td>
<td>MongoDB ops/sec</td>
<td>Up to 18,294</td>
<td>Up to 36,588</td>
<td>Up to 73,176</td>
</tr>
<tr>
<td></td>
<td>XtremIO average latency (ms)</td>
<td>Up to 2.4 ms average storage latency</td>
<td>Up to 2.4 ms average storage latency</td>
<td>Up to 2.4 ms average storage latency</td>
</tr>
<tr>
<td></td>
<td>Number of ESXi hosts</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

For detailed results, see Chapter 5, Test Cases and Results.
This chapter presents the following topics:

- **YCSB overview** ........................................................................................................21
- **YCSB configuration** ................................................................................................21
YCSB overview

YCSB is an open-source tool that was developed by Yahoo to stress-test database systems. It is used for a common set of random I/O workloads to evaluate the performance of key-value and cloud serving stores. YCSB has two parts:

- **YCSB Client**—An extensible workload generator
- **Core workloads**—A set of workload scenarios to be executed by the generator

The core workloads provide a well-rounded picture of system performance. The YCSB Client is extensible so that you can define additional workloads to examine system aspects or application scenarios that are not covered by the core workload. The Client can also be extended to benchmark different databases. YCSB ships with bindings for numerous databases including HBase, Cassandra, Apache Accumulo, MongoDB, and Voldemort. Support for other data stores can be added by writing an interface layer.

**Note:** For more information about YCSB, refer to How to benchmark MongoDB with YCSB and Quick Start for running YCSB on MongoDB.

YCSB configuration

For this solution, we create two YCSB virtual machines for each database to generate about 300 GB of data on each node. We then run the three workloads that are described in Chapter 5, Test Cases and Results. We run the workloads in asynchronous mode with the write concern parameter set to the primary node to ensure the performance.

For each YCSB virtual machine, we launch eight YCSB Client processes to use 12 threads to reach maximum performance.
This chapter presents the following topics:

Overview ..................................................................................................................................................23
Virtualized MongoDB on XtremIO building blocks........................................................................24
XVCs for MongoDB .............................................................................................................................30
XtremIO compression with MongoDB Advanced Compression .....................................................32
XtremIO rapid recovery for MongoDB replica set.............................................................................34
Overview

Table 10 provides an overview of the scenarios that we tested and validated for this solution.

Table 10. Test scenarios

<table>
<thead>
<tr>
<th>Test scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtualized MongoDB on XtremIO X-Brick building blocks</td>
<td>Validates different MongoDB workload types and scales on the XtremIO array in a virtualized environment</td>
</tr>
<tr>
<td>XVCs for virtualized MongoDB</td>
<td>Demonstrates that XtremIO provides no performance impact and storage footprint for the MongoDB deployment</td>
</tr>
<tr>
<td>XtremIO compression with MongoDB Advanced Compression</td>
<td>Demonstrates XtremIO inline data reduction, including compression, deduplication, thin provisioning, and space-saving XVCs that drive significant data efficiencies with a reduced TCO</td>
</tr>
<tr>
<td>XtremIO rapid recovery for a MongoDB replica set</td>
<td>Simulates replica set failures and the use of XVCs for recovery</td>
</tr>
</tbody>
</table>

Notes: Benchmark results are highly dependent on workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, the workloads should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data that is contained in this report was obtained in a rigorously controlled environment. Results that are obtained in other operating environments may vary significantly.

EMC Corporation does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute.
Virtualized MongoDB on XtremIO building blocks

Test scenario
To validate virtualized MongoDB on the XtremIO array, we use YCSB workloads to simulate real-world usage profiles, as shown in Table 11.

Table 11. Workload type descriptions

<table>
<thead>
<tr>
<th>MongoDB workload type</th>
<th>YCSB workload type</th>
<th>Workload profile</th>
<th>I/O characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query-dominated</td>
<td>YCSB workload B</td>
<td>Read-mostly workload with a 95% read to 5% update ratio</td>
<td>Combination of 32 KB, 16 KB, 8 KB, and 4 KB of I/O size</td>
</tr>
<tr>
<td>Query/modify-dominated</td>
<td>YCSB workload F</td>
<td>Read-modify-write workload with a 50% read to 50% read-modify-write ratio</td>
<td></td>
</tr>
<tr>
<td>Insert-dominated</td>
<td>YCSB custom workload</td>
<td>Write-mostly workload with a 90% insert to 10% read ratio</td>
<td></td>
</tr>
</tbody>
</table>

For each workload type, we scale out from one standalone MongoDB database to five MongoDB replica sets to achieve maximum system capability. We collect the following performance metrics:

- MongoDB ops/sec
- XtremIO IOPS
- XtremIO latency (ms)
- XtremIO CPU utilization (%)
- ESXi CPU utilization (%)

To achieve steady-state operations, each test runs for 2 hours. We disregard the first 15 minutes of warm-up time.

Test results and analysis

Workload B test results
Figure 4 shows the IOPS and CPU utilization on a single X-Brick building block when we scale out the mixed MongoDB databases.
Chapter 5: Test Cases and Results

Figure 4. XtremIO IOPS and CPU utilization with workload B

Figure 4 indicates that a single X-Brick building block can support 72,085 IOPS with the scale out of virtualized MongoDB replica sets.

When the virtualized MongoDB deployment is scaled out to five replica sets with a standalone instance, the latency is still extremely low, as shown in Figure 5.

Figure 5. MongoDB ops/sec and average latency on XtremIO with workload B

In this solution, we deploy two ESXi servers to host MongoDB virtual machines, maximizing flexibility of the scale-out topology and consolidating the compute resources. As shown in Figure 6, the first three deployments, from standalone to two replica sets with standalone, use only one ESXi server. Subsequently, the MongoDB virtual machines are spread to both ESXi servers to balance the CPU utilization of ESXi.
Chapter 5: Test Cases and Results

Figure 6. ESXi CPU utilization with workload B

Table 12 summarizes the workload B configuration and the performance results.

<table>
<thead>
<tr>
<th>Storage configuration</th>
<th>MongoDB workload type</th>
<th>Number of ESXi servers (20 CPUs, 512 GB memory)</th>
<th>XtremIO IOPS</th>
<th>XtremIO latency</th>
<th>MongoDB ops/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>XtremIO single X-Brick building block</td>
<td>Query-dominated</td>
<td>1</td>
<td>Up to 64,398</td>
<td>Less than 1.4 ms</td>
<td>Up to 49,852</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Up to 72,085</td>
<td>Less than 2.1 ms</td>
<td>Up to 55,251</td>
</tr>
</tbody>
</table>

Due to the predictable performance of the XtremIO array, we can easily scale out the topology to support additional workload with constant low latency.

**Workload F test results**

Table 7 shows the IOPS and its CPU utilization when we scale out the MongoDB topology running YCSB workload F on one X-Brick building block.
Figure 7.  **XtremIO IOPS and CPU utilization with workload F**

The figure shows that, due to the CPU utilization of the XtremIO array, the IOPS remains flat when we deploy more than three replica sets with a standalone MongoDB.

As indicated by Figure 7, the maximum IOPS, 52,516, occurs when we deploy three MongoDB replica sets with one standalone virtual machine.

Figure 8 reflects a similar pattern in MongoDB ops/sec and latency.

Figure 8.  **MongoDB ops/sec and average latency on XtremIO with workload F**

Table 13 shows the ESXi CPU utilization. As indicated, a single server is sufficient to handle all types of MongoDB deployments in this test scenario.
Table 13. ESXi CPU utilization on workload F

<table>
<thead>
<tr>
<th>Standalone (SL)</th>
<th>1 MongoDB replica set with SL</th>
<th>2 MongoDB replica sets with SL</th>
<th>3 MongoDB replica sets with SL</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.01%</td>
<td>31.69%</td>
<td>62.26%</td>
<td>77.7%</td>
</tr>
</tbody>
</table>

Table 14 summarizes the workload F configuration and the performance results.

Table 14. Workload F: Configuration and performance results

<table>
<thead>
<tr>
<th>Storage configuration</th>
<th>MongoDB workload type</th>
<th>Number of ESXi servers (20 CPUs, 512 GB memory)</th>
<th>XtremIO IOPS</th>
<th>XtremIO latency</th>
<th>MongoDB ops/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>XtremIO single X-Brick building block</td>
<td>Query/modify -dominated</td>
<td>1</td>
<td>Up to 52,516</td>
<td>Up to 3.2 ms</td>
<td>Up to 23,778</td>
</tr>
</tbody>
</table>

Due to the predictable performance of the XtremIO array, we can easily scale out the topology to support additional workload with constant low latency.

**Custom workload test results**

The test results of a custom workload are similar to the results for workload F. When we deploy more than two MongoDB replica sets with one standalone virtual machine, the performance remains flat due to the CPU utilization of the XtremIO array. Figure 9 shows the XtremIO IOPS and CPU utilization, and Figure 10 shows the MongoDB ops/sec and average latency.

---

**Figure 9.** XtremIO IOPS and CPU utilization with custom workload
Table 15 shows the ESXi CPU utilization. As indicated, a single server is sufficient to handle all types of MongoDB deployments in this test scenario.

Table 15. ESXi CPU utilization on custom workload

<table>
<thead>
<tr>
<th>Storage configuration</th>
<th>MongoDB workload type</th>
<th>Number of ESXi servers (20 CPUs, 512 GB memory)</th>
<th>XtremIO IOPS</th>
<th>XtremIO latency</th>
<th>MongoDB ops/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>XtremIO single X-Brick building block</td>
<td>Insert-dominated</td>
<td>1</td>
<td>Up to 42,144</td>
<td>Less than 2.4ms</td>
<td>Up to 18,294</td>
</tr>
</tbody>
</table>

Table 16 summarizes the custom workload configuration and the performance results.

Table 16. Custom workload: Configuration and performance results

Due to the predictable performance of the XtremIO array, we can easily scale out the topology to support additional workload with constant low latency.
**XVCs for MongoDB**

### Test scenario

In this scenario, we test the impact of XVCs by running the following procedure:

1. **Configure the XVC Scheduler:**
   - a. Create one Consistency Group including the three volumes of replica cluster 04 on the XtremIO array.
   - b. Create one Scheduler on the Consistency Group and configure it to create 33 XVC Snapshot Sets within 20 minutes.

2. Run the custom workload, which is 90 percent writes and 10 percent reads, on replica cluster 04.

3. Start the Scheduler to create 33 Snapshot Sets on each database volume within 20 minutes.

4. Stop the custom workload and collect performance data from the XtremIO array.

5. Create another Snapshot Set and mount it to the MongoDB nodes to verify XVC validity.

### Test results and analysis

This test shows that XVCs have minimum performance impact on the production database. The capacity footprint increases minimally after XVCs, and the XVC database instances can be restored instantly.

According to the test results, the XVC volumes are proved to be valid and consistent. With XVC technology, the XtremIO array can support quick disaster recovery in the data center and consolidate test/development and production environments into one platform.

Figure 11 shows the overall efficiency increases to over 27:1 from 6:1 after 99 XVCs were created. These results demonstrate that XtremIO space-saving XVC services and thin provisioning produce extraordinary data efficiencies.
The XtremIO dashboards in Figure 11 show volume capacity of 138.9 TB after XVCs compared to 30.0 TB before XVCs. However, the physical capacity change is minimal.

**Physical Space Used** in the **Volume Capacity** section of the dashboards show that the physical space used increased less than 0.2 TB when we created more than 100 TB XVCs on the array. Approximately 500 times the storage capacity requirement was reduced inline.

The physical space used resulted primarily from the data that was inserted in the database as we used the custom workload. XVCs have minimal impact on the storage footprint.

To validate the production impact of XVCs, we checked the performance metrics when we ran the custom workload against the cluster. The results show that XVCs have limited or no impact on the production environment in terms of IOPS and latency.

Figure 12 shows the IOPS metrics on the database volumes of the MongoDB replica set. We started the XVC Scheduler in the middle of the workload and stopped it after 20 minutes. No obvious degradation or improvement occurred when the workload was run with or with XVCs in the background.

![Figure 12. IOPS of workload with and without XVCs being taken](image)

As shown in Figure 13, the latency metrics show no obvious degradation or improvement when the workload was run with and without XVCs in the background. Those metrics show that XVCs have minimal impact on storage performance and the MongoDB application.
Chapter 5: Test Cases and Results

As shown by the test results, XVCs can provide customers with parallel environments, generating as many copies as wanted without any impact on production SLAs. These XVCs improve TCO, accelerate workflow, and even enable test/development copies to be tested in the actual production environment.

XtremIO compression with MongoDB Advanced Compression

This test scenario monitors the storage footprint of XtremIO inline data reduction with MongoDB snappy and zlib compression. We focus on storage capacity and database server resource utilization with MongoDB snappy and zlib compression turned on and off. The overall testing procedure is as follows:

1. On a MongoDB replica set:
   a. Set the MongoDB compression option to None.
   b. Initialize dataset testing with 260 million records.
   c. Run the YCSB custom workload to generate 6.5 million new records.
   d. Collect the performance metrics.
2. Repeat the preceding step for MongoDB snappy and zlib compression options.

Table 17 provides the overall test results, which show that XtremIO inline data reduction provides excellent compression and performance compared to MongoDB default snappy compression. XtremIO inline data reduction also supports offloading 6 percent to 7 percent of MongoDB virtual-machine-server CPU utilization compared to MongoDB zlib compression.
Table 17. Test results of XtremIO inline compression

<table>
<thead>
<tr>
<th>Metric</th>
<th>None</th>
<th>Snappy</th>
<th>Zlib</th>
</tr>
</thead>
<tbody>
<tr>
<td>XtremIO storage footprint (GB)</td>
<td>348.43</td>
<td>348.07</td>
<td>247.70</td>
</tr>
<tr>
<td>MongoDB database performance (ops/second)</td>
<td>11,996</td>
<td>11,891</td>
<td>9,246</td>
</tr>
<tr>
<td>MongoDB virtual machine server CPU utilization (percent)</td>
<td>22.99</td>
<td>22.44</td>
<td>29.84</td>
</tr>
</tbody>
</table>

Figure 14 shows that the storage footprint of XtremIO inline data reduction alone and that of snappy compression are nearly the same, while zlib compression has the smallest storage footprint.

Figure 14. XtremIO storage footprint with and without MongoDB compression

Figure 15 shows zlib compression introduced a 20 percent IOPS degradation compared to snappy compression and XtremIO inline data reduction.

Figure 15. MongoDB performance with and without MongoDB compression

Figure 16 shows that MongoDB virtual-machine-server CPU utilization of XtremIO inline data reduction alone and that of snappy compression are nearly the same. The figure also shows that zlib compression has an additional 6 percent to 7 percent overhead.
Chapter 5: Test Cases and Results

Figure 16.  MongoDB virtual machine server CPU utilization with and without MongoDB compression

XtremIO rapid recovery for MongoDB replica set

Test scenario and results

In this scenario, we test how XVCs enable rapid recovery of MongoDB replica sets.

We use following steps to test this scenario:

1. Start YCSB workloads against a MongoDB replica set.
2. Create an XVC for each MongoDB host.
3. Separately emulate two unexpected MongoDB cluster failure cases:
   ▪ Network partition failure
   ▪ Single node failure in the MongoDB replica set
4. Remount the XVCs that are created before the cluster failure.
5. Restart the MongoDB replica set.
6. Check the data consistency in the MongoDB replica set.

In the network partition failure scenario, we emulate step-by-step interconnection losses between nodes in the MongoDB replica set, resulting in the cluster failure.

In the single-node failure scenario, which occurs frequently in production environments, we power off the nodes in the MongoDB replica set one by one, resulting in the cluster failure.

In both failure cases, the XVCs are restored instantly in the MongoDB replica set and the restored data is consistent on each node.
This chapter presents the following topics:

- **Summary** ................................................................. 36
- **Findings** ............................................................... 36
Summary

This solution demonstrates the benefits of deploying XtremIO all-flash storage with the MongoDB NoSQL database on the VMware virtualization platform. The solution provides high scalability, high performance, and high compression and deduplication along with data-protection features.

Storage performance can scale linearly in response to an increase in the number of MongoDB hosts. Customers can dynamically provision MongoDB virtual machines on VMware vSphere ESXi hosts to meet changing requirements.

The XtremIO array, with its advanced compression algorithm and unique deduplication technology, provides cost efficiencies by enabling customers to manage more data with less storage.

In addition, the XVC feature is a supplementary solution for MongoDB native backup and recovery. Creation of XVCs, a block-level backup and recovery method that is recommended for MongoDB, has no resource impact on production. Further, in the event of a failure, MongoDB can access the data profile from any specific XVC-creation time.

Findings

Testing of this solution showed the following:

- MongoDB with the XtremIO array on a virtualized infrastructure can easily and flexibly provision MongoDB instance nodes as required. VMware vSphere simplifies the MongoDB server management and maintenance workflow.
- XtremIO scalability from one X-Brick to six X-Brick building blocks supports large scale-out of a MongoDB cluster, with predictable, linear performance and low latency.
- XVCs on MongoDB replica sets have no impact on the capacity footprint or performance.
- Advanced inline compression and deduplication features of the XtremIO array bring dramatic storage savings. Native thin provisioning helps customers and stakeholders dramatically improve storage usability and cost efficiency. On the other hand, MongoDB snappy is a lightweight compression method with balanced CPU and storage utilization, while zlib brings dramatic compression capability but heavy CPU overhead.
- XVCs enable rapid, simple, and consistent data backup and recovery. Consistency Group and Scheduler features enable automatic XVC creation. XVCs enable the quick recovery of MongoDB data to any previous XVC-creation time.
This chapter presents the following topics:

EMC documentation .................................................................................................38
VMware documentation ...........................................................................................38
Other documentation ..............................................................................................38
Chapter 7: References

**EMC documentation**

The following documentation on [EMC.com](https://www.emc.com) or [EMC Online Support](https://www.emc.com) provides additional and relevant information. Access to these documents depends on your login credentials. If you do not have access to a document, contact your EMC representative.

- [EMC High-Performance Flash for MongoDB Databases Solution Guide](https://www.emc.com)
- [Introduction to XtremIO Virtual Copies White Paper](https://www.emc.com)
- [XtremIO Data Protection (XDP) White Paper](https://www.emc.com)
- [Introduction to the EMC XtremIO Storage Array White Paper](https://www.emc.com)
- [EMC XtremIO Storage Array Host Configuration Guide](https://www.emc.com)
- [EMC AppSync User and Administration Guide](https://www.emc.com)

**VMware documentation**

The following documentation on the [VMware website](https://www.vmware.com) provides additional and relevant information:

- [VMware vSphere 6.0 Documentation Center](https://www.vmware.com)
- [Performance Best Practices for VMware vSphere 6.0](https://www.vmware.com)

**Other documentation**

The following documentation provides additional and relevant information:

- [The MongoDB 3.2 Manual](https://www.mongodb.com)