A Principled Technologies report: Hands-on testing. Real-world results.

A Principled Technologies report: Hands-on testing. Real-world results.

142,214 operations per second

Process a large number of transactions

1.64ms average read latency

Drive productivity

95 TB of storage in just 3U

Maximize storage density

Support high-performance business applications with a powerful Dell EMC, Nutanix, and Toshiba solution

A Dell EMC XC Series cluster featuring Nutanix software and powered by Toshiba PX05S SAS SSDs delivered strong database performance with a blend of structured and unstructured data

The document-based MongoDB, one of the most popular database engines available,[1] allows developers to create dynamic, responsive enterprise applications and bring together structured and unstructured data. To work at optimal levels, MongoDB projects require a solution with copious storage and fast processing.

Toshiba, Nutanix®, and Dell EMC™ offer such a solution, supporting high usage environments while minimizing wait times. Short wait times keep users from getting distracted or frustrated and increase the likelihood of repeat usage, making your initiative more successful. In addition, a Toshiba, Nutanix, and Dell EMC solution offers high storage density that not only supports massive datastores but keeps datacenter sprawl in check as well.

In the Principled Technologies datacenter, we configured a Dell EMC XC Series cluster, featuring Nutanix software paired with VMware vSphere® ESXi™ as the hypervisor, with Toshiba PX05S SAS SSDs and ran a heavy workload against a MongoDB database. The solution delivered high performance, which can mean a smooth and quick experience with your app, whether viewing search results, finding a tagged photo, or doing a host of other operations.

Support high-performance business applications with a powerful Dell EMC, Nutanix, and Toshiba solution June 2018 (Revised)

The science behind the report:

Support high-performance business applications with a powerful Dell EMC, Nutanix, and Toshiba solution

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Support high-performance business applications with a powerful Dell EMC, Nutanix, and Toshiba solution.

On March 15, 2018, we finalized the hardware and software configurations we tested. Updates for current and recently released hardware and software appear often, so unavoidably these configurations may not represent the latest versions available when this report appears. For older systems, we chose configurations representative of typical purchases of those systems. We concluded hands-on testing on April 6, 2018.

Our results

The tables below presents our findings in detail.

Latency

Measurement	Host 1	Host 2	Host 3	Average
CPU %	64.76	58.55	72.53	65.28
Read latency (ms)	1.93	1.53	1.45	1.64
Write latency (ms)	1.64	1.31	1.25	1.40

Operations per second

YCSB driver	YCSB operations/sec
Driver 1	47,427.9
Driver 2	47,368.5
Driver 3	47,417.9
Total	142,214.3

System configuration information

The table below presents detailed information on the systems we tested.

Server configuration information	3 x Dell EMC™ PowerEdge™ R640
BIOS name and version	Dell 1.3.7
Non-default BIOS settings	Intel Turbo Boost enabled, Virtualization enabled
Operating system name and version/build number	VMware vSphere® ESXi™ 6.5 and Nutanix® 5.1.3
Date of last OS updates/patches applied	03/15/2018
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 5120
Core count (per processor)	14
Core frequency (GHz)	2.20
Stepping	1
Memory module(s)	
Total memory in system (GB)	192
Number of memory modules	12
Vendor and model	Hynix HMA82GR7AFR8N-VK
Size (GB)	16
Type	PC4-21300R
Speed (MHz)	2,666
Speed running in the server (MHz)	2,400
Storage controller	
Vendor and model	Dell HBA330 Mini (Embedded)
Cache size (GB)	0
Firmware version	13.17.03.05
Driver version	7.700.50.00
Local storage (type A)	
Number of drives	10
Drive vendor and model	Toshiba PX05SRB384Y
Drive size (GB)	3.84
Drive information (speed, interface, type)	12Gb SAS SSD

Server configuration information	3 x Dell EMC™ PowerEdge™ R640
Network adapter	
Vendor and model	Mellanox ConnectX -4 Lx EN
Number and type of ports	2 x 25 GbE
Driver version	14.20.18.22
Cooling fans	
Vendor and model	Delta Electronics GFC0412DS-D
Number of cooling fans	8
Power supplies	
Vendor and model	Dell D750E-S6
Number of power supplies	2
Wattage of each (W)	750

How we tested

Configuring the Nutanix servers

1. Log onto a server that's on the same network as the Nutanix nodes.
2. Open the Nutanix Foundation applet and run discovery.
3. When you've verified that your nodes are there, launch foundation.
4. In the Discover Nodes step, verify that the Redundancy Factor is set to 2, and click Next.
5. In the Define Cluster step, name your cluster and give it an IP address, then set the general network information, and click Next.
6. In the Setup Node step, enter your hostname and IPs in the top list, then verify that they were propagated correctly to the fields below, and click Next.
7. In the Select Images step, click Skip Imaging.
8. In the Create Cluster step, monitor that things are running properly.
9. When the cluster is finished creating, click on the manage your cluster with Prism prompt.
10. Log into your Prism console with the username/password you defined.
11. Follow the prompts to change your username and password.
12. Select Home→Storage.
13. In the Storage screen, select + Storage Container.
14. In the Create Storage Container window, name your storage container, verify that Mount on all ESXi hosts is selected, and click Save.

Configuring of the VMs for testing

We used the following configuration for the VMs we tested.

VM type	Number of VMs	vCPUs	vRAM	HDDs
Config server	3	8	32 GB	1 x 32 GB (OS) 1 x 200 GB (DB)
Mongos	3	8	64 GB	1 x 32 GB (OS)
Mongod	9	8	64 GB	1 x 32 GB (OS) 1 x 600 GB (DB

Installing and configuring MongoDB on the configuration server VMs

We installed CentOS 7 onto each VM, then performed the following steps on each server:

1. Log into the VM you are configuring.
2. Stop and disable the firewall service by typing the following commands:
```
systemctl stop firewalld
systemctl disable firewalld
```
3. Enter the SELinux configuration file by typing the following command:
```
vim /etc/sysconfig/selinux
```
4. Once inside the SELinux configuration file, change its configuration from enabled to disabled.
5. Create a new tuned config file by typing the following commands:
```
sudo mkdir /etc/tuned/no-thp
vim /etc/tuned/no-thp/tuned.conf
```
6. With the blank tuned.conf file open, write the following into the file, and save:
```
[main]
include=virtual-guest

[vm]
transparent_hugepages=never
```
7. Perform system updates:
```
sudo yum update -y
```

8. Add the Mongo community database to your yum repo, and install MongoDB to your VM by typing the following commands:

```
vim /etc/yum.repos.d/mongodb-org-3.6.repo
yum install -y mongodb-org
```

9. Make mongod start with the operating system by typing the following command:

```
echo “mongod --config=/etc/mongod.conf” >> /etc/rc.d/rc.local
```

10. Add the database drive to the VM, and create the database folders for MongoDB by typing the following commands:

```
mkfs.xfs /dev/sdb
mount /dev/sdb /mnt
vim /etc/fstab
mkdir /mnt/var
mkdir /mnt/var/lib
mkdir /mnt/var/lib/mongo
```

11. Open the mongod configuration file by typing the following command:

```
vim /etc/mongod.conf
```

12. Inside the mongod configuration file, make the following changes:

```
storage:
  dbPath: /mnt/var/lib/mongo
  journal:
    enabled: true
  engine: “wiredTiger”
net:
  port: 27017
  bindIp: [your server’s IP],localhost
replication:
    replSetName: config-replica-set
sharding:
    clusterRole: configsvr
```

13. Save the mongod configuration file.
14. Start your mongod server by typing the following command:

```
mongod --config=/etc/mongod.conf
```

15. Repeat steps 1 through 14 on your remaining two config servers.

Creating the config server replica set

1. Type `mongo` to enter the MongoDB console.
2. In the MongoDB console, create the replica set by typing the following command:

```
rs.initiate(
  {
    _id : “config-replica-set”,
    members: [
      { _id : 0, host : “config-server-1.test.local:27017” },
      { _id : 1, host : “config-server-2.test.local:27017” },
      { _id : 2, host : “config-server-3.test.local:27017” }
    ]
  }
)
```

Installing and configuring MongoDB on the mongos VMs

1. Log into the VM you are configuring.
2. Stop and disable the firewall service by typing the following commands:

```
systemctl stop firewalld
systemctl disable firewalld
```

3. Enter the SELinux configuration file by typing the following command:

```
vim /etc/sysconfig/selinux
```

4. Once inside the SELinux configuration file, change its configuration from enabled to disabled.
5. Create a new tuned config file by typing the following commands:

```
sudo mkdir /etc/tuned/no-thp
vim /etc/tuned/no-thp/tuned.conf
```

6. With the blank tuned.conf file open, write the following into the file, and save:

```
[main]
include=virtual-guest
[vm]
transparent_hugepages=never
```

7. Perform system updates:

```
sudo yum update -y
```

8. Add the Mongo community database to your yum repo, and install MongoDB to your VM by typing the following commands:

```
vim /etc/yum.repos.d/mongodb-org-3.6.repo
yum install -y mongodb-org
```

9. Make mongod start with the operating system by typing the following command:

```
echo "mongos --config=/etc/mongod.conf" >> /etc/rc.d/rc.local
```

10. Open the mongod configuration file by typing the following command:

```
vim /etc/mongod.conf
```

11. Inside the mongod configuration file, make the following changes:

```
#storage:
  #dbPath: /var/lib/mongo
  #journal:
    #enabled: true
net:
  port: 27017
  bindIp: [your server's IP],localhost
sharding:
    configDB: config-replica-set/config-server-1.test.local:27017,config-server-2.test.local:27017,config-
server-3.test.local
:27017
```

12. Save the mongod configuration file.
13. Start your mongod server by typing the following command:

```
mongos --config=/etc/mongod.conf
```

14. Repeat steps 1 through 13 on your remaining two mongos servers.

Installing and configuring MongoDB on the mongod VMs

1. Log into the VM you are configuring.
2. Stop and disable the firewall service by typing the following commands:

```
systemctl stop firewalld
systemctl disable firewalld
```

3. Enter the SELinux configuration file by typing the following command:

```
vim /etc/sysconfig/selinux
```

4. Once inside the SELinux configuration file, change its configuration from enabled to disabled.
5. Create a new tuned config file by typing the following commands:

```
sudo mkdir /etc/tuned/no-thp
vim /etc/tuned/no-thp/tuned.conf
```

6. With the blank tuned.conf file open, write the following into the file, and save:

```
[main]
include=virtual-guest
[vm]
transparent_hugepages=never
```

7. Perform system updates:

```
sudo yum update -y
```

8. Add the Mongo community database to your yum repo, and install MongoDB to your VM by typing the following commands:

```
vim /etc/yum.repos.d/mongodb-org-3.6.repo
yum install -y mongodb-org
```

9. Make mongod start with the operating system by typing the following command:

```
echo “mongod --config=/etc/mongod.conf” >> /etc/rc.d/rc.local
```

10. Add the database drive to the VM, and create the database folders for MongoDB by typing the following commands:

```
mkfs.xfs /dev/sdb
mount /dev/sdb /mnt
vim /etc/fstab
mkdir /mnt/var
mkdir /mnt/var/lib
mkdir /mnt/var/lib/mongo
```

11. Open the mongod configuration file by typing the following command:

```
vim /etc/mongod.conf
```

12. Inside the mongod configuration file, make the following changes, filling in the # with the appropriate number:

```
storage:
  dbPath: /mnt/var/lib/mongo
  journal:
    enabled: true
  engine: “wiredTiger”
net:
  port: 27017
  bindIp: [your server’s IP],localhost
replication:
    replSetName: shard-replica-set-#
sharding:
    clusterRole: shardsvr
```

13. Save the mongod configuration file.
14. Start your mongod server by typing the following command:

```
mongod --config=/etc/mongod.conf
```

15. Repeat steps 1 through 14 on your remaining eight mongod servers.

Creating the mongod replica sets

1. Type mongo to enter the MongoDB console.
2. In the MongoDB console, create the replica set by typing the following command, filling in the # in the command with the appropriate number:

```
rs.initiate(

  {

    _id : “shard-replica-set-#”,

    members: [

      { _id : 0, host : “mongod-#.test.local:27017” },

      { _id : 1, host : “mongod-#.test.local:27017” },

      { _id : 2, host : “mongod-#.test.local:27017” }

    ]

  }

)
```

3. Repeat steps 1 and 2 for the other two replica sets.

Installing the YCSB driver VMs

1. Log into the VM you are configuring.
2. Stop and disable the firewall service by typing the following commands:
   ```
   systemctl stop firewalld
   systemctl disable firewalld
   ```
3. Enter the SELinux configuration file by typing the following command:
   ```
   vim /etc/sysconfig/selinux
   ```
4. Once inside the SELinux configuration file, change its configuration from enabled to disabled.
5. Perform system updates:
   ```
   sudo yum update -y
   ```
6. Add the Mongo community database to your yum repo, and install MongoDB to your VM by typing the following commands:
   ```
   vim /etc/yum.repos.d/mongodb-org-3.6.repo
   yum install -y mongodb-org
   ```
7. Install Java with the following command:
   ```
   sudo yum install java-devel
   ```
8. Install Maven with the following commands:
   ```
   wget "http://apache.cs.utah.edu/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz"
   sudo tar xzf apache-maven-*-bin.tar.gz -C /usr/local
   cd /usr/local
   sudo ln -s apache-maven-* maven
   echo export M2_HOME=/usr/local/maven >> /etc/profile.d/maven.sh
   echo export PATH=${M2_HOME}/bin:${PATH} >> /etc/profile.d/maven.sh
   ```
9. Download and unpack YCSB into your driver with the following commands:
   ```
   run("curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.12.0/ycsb-
   0.12.0.tar.gz")
   run("tar xfvz ycsb-0.12.0.tar.gz")
   ```
10. Repeat steps 1 through 9 on your remaining two YCSB drivers.

Creating the YCSB database

1. Log into a YCSB VM.
2. Run the following command to create a roughly 1.2 TB database for MongoDB:
   ```
   /root/ycsb-0.12.0/bin/ycsb load mongodb -s -P /root/ycsb-0.12.0/workloads/workloadb -threads 64 -p
   mongodb
   .url=mongodb://mongos-1.test.local:27017/ycsb?w=1 -p recordcount=1000000000
   ```
3. As the database is being created, log into one of your shard VMs.
4. Enter MongoDB by typing the following command:
   ```
   mongo
   ```
5. Type the following commands to shard and chunk your MongoDB database:
   ```
   use ycsb
   sh.enableSharding("ycsb")
   sh.shardCollection("ycsb.usertable", { _id: 1 }, true )
   ```
6. The database will finish building in several hours. After the database is finished building, run the following command to check on the progress of the shard distribution:
   ```
   sh.status()
   ```
7. Once all shards have an equivalent number of chunks distributed, you are ready to test.

Running the YCSB test

1. On each mongod VM, run the following command to clear all caches:

```
echo 3 > /proc/sys/vm/drop_caches
```

2. Log into your three YCSB drivers.
3. On all three drivers, run the following command to warm up the database:

```
/root/ycsb-0.12.0/bin/ycsb run mongodb -s -P /root/ycsb-0.12.0/workloads/workloadb -threads 64 -p
mongo
db.url=mongodb://mongos-1.test.local:27017,mongos-2.test.local:27017,mongos-3.test.local:27017/
ycsb?w=1 -p operationcount=50000000 -p mongodb.upsert=true > ~/output/ycsbtest_64_threads_50000000_
operations_workloadb_[your_ycsb_hostname]
.output
```

4. Once the database is warmed up, run the following command to test it:

```
/root/ycsb-0.12.0/bin/ycsb run mongodb -s -P /root/ycsb-0.12.0/workloads/workloadb -threads 64 -p
mongo
db.url=mongodb://mongos-1.test.local:27017,mongos-2.test.local:27017,mongos-3.test.local:27017/
ycsb?w=1 -p operationcount=50000000 -p mongodb.upsert=true > ~/output/ycsbtest_64_threads_50000000_
operations_workloadb_[your_ycsb_hostname]
.output
```

5. Record the results from the test.

Read the report at http://facts.pt/esCvbq ▶

This project was commissioned by Dell Technologies.

Principled Technologies®

Facts matter.®

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:
Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.