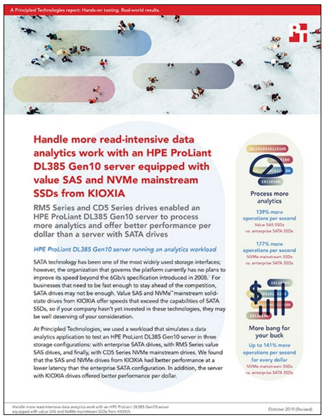

A Principled Technologies report: Hands-on testing. Real-world results.

Handle more read-intensive data analytics work with an HPE ProLiant DL385 Gen10 server equipped with value SAS and NVMe mainstream SSDs from KIOXIA

RM5 Series and CD5 Series drives enabled an HPE ProLiant DL385 Gen10 server to process more analytics and offer better performance per dollar than a server with SATA drives

HPE ProLiant DL385 Gen10 server running an analytics workload

SATA technology has been one of the most widely used storage interfaces; however, the organization that governs the platform currently has no plans to improve its speed beyond the 6Gb/s specification introduced in 2008.[1] For businesses that need to be fast enough to stay ahead of the competition, SATA drives may not be enough. Value SAS and NVMe™ mainstream solid-state drives from KIOXIA offer speeds that exceed the capabilities of SATA SSDs, so if your company hasn't yet invested in these technologies, they may be well deserving of your consideration.

At Principled Technologies, we used a workload that simulates a data analytics application to test an HPE ProLiant DL385 Gen10 server in three storage configurations: with enterprise SATA drives, with RM5 Series value SAS drives, and finally, with CD5 Series NVMe mainstream drives. We found that the SAS and NVMe drives from KIOXIA had better performance at a lower latency than the enterprise SATA configuration. In addition, the server with KIOXIA drives offered better performance per dollar.

Process more analytics

139% more operations per second Value SAS SSDs vs. enterprise SATA SSDs

177% more operations per second NVMe mainstream SSDs vs. enterprise SATA SSDs

More bang for your buck

Up to 141% more operations per second for every dollar NVMe mainstream SSDs vs. enterprise SATA SSDs

Handle more read-intensive data analytics work with an HPE ProLiant DL385 Gen10 server equipped with value SAS and NVMe mainstream SSDs from KIOXIA

October 2019 (Revised)

The science behind the report:

Handle more read-intensive data analytics work with an HPE ProLiant DL385 Gen10 server equipped with value SAS and NVMe mainstream SSDs from KIOXIA

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 Handle more read-intensive data analytics work with an HPE ProLiant DL385 Gen10 server equipped with value SAS and NVMe mainstream SSDs from KIOXIA.

We concluded our hands-on testing on July 15, 2019. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on June 25, 2019 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

The table below presents our findings in detail.

For the total solution cost, we got a quote for a base model of the HPE ProLiant DL385 Gen10 matching our configuration minus the disks. We then got the list prices from HPE for the RM5 and CD5 SSDs. For the SATA SSDs, we viewed the online sources to purchase from as per Intel's site and calculated the average price list. We then added the cost of four drives per configuration to the base server price to get the total solution cost for each drive type.

	HPE ProLiant DL385 Gen10 with enterprise SATA SSDs	HPE ProLiant DL385 Gen10 with KIOXIA RM5 Series value SAS SSDs	HPE ProLiant DL385 Gen10 with KIOXIA CD5 Series NVMe mainstream SSDs
Operations per second	17,603	42,208	48,827
Avg. data read latency (ms)	14.538	6.059	5.238
Total hardware cost of the solution (USD)	32,947.16	36,746.00	37,946.00
Operations per dollar (USD)	0.534	1.149	1.287

System configuration information

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

Server configuration information	HPE ProLiant DL385 Gen10
BIOS name and version	A40 v1.30
Operating system name and version/build number	CentOS Linux release 7.6.1810
Date of last OS updates/patches applied	06/05/2019
Power management policy	Balanced
Processor	
Number of processors	2
Vendor and model	AMD EPYC™ 7601
Core count (per processor)	32
Core frequency (GHz)	2.20
Stepping	ZP-B2
Memory module(s)	
Total memory in system (GB)	256
Number of memory modules	16
Vendor and model	HPE SmartMemory
Size (GB)	16
Type	DDR4-2666V
Speed (MHz)	2,666
Speed running in the server (MHz)	2,400
Storage controller	
Vendor and model	HPE Smart Array E208i-a SR Gen10
Firmware version	1.98
Driver version	Default
Local storage - SATA	
Number of drives	4
Drive vendor and model	Intel® SSDSC2KB960G8
Drive size (GB)	960
Drive information	SATA SSD

Server configuration information	**HPE ProLiant DL385 Gen10**
Local storage - NVMe	
Number of drives	4
Drive vendor and model	HPE VO000960KWVBR (KIOXIA CD5 Series NVMe mainstream)
Drive size (GB)	960
Drive information	NVMe SSD
Local storage - SAS	
Number of drives	4
Drive vendor and model	HPE VO000960RWUEV (KIOXIA RM5 Series value SAS)
Drive size (GB)	960
Drive information	SAS SSD
Local storage - OS	
Number of drives	1
Drive vendor and model	KIOXIA THNSNK256GVN8
Drive size (GB)	256
Drive information	SATA SSD
Network adapter	
Vendor and model	Broadcom Inc. NetXtreme BCM5719 Gigabit Ethernet
Number and type of ports	4 x 1GbE
Driver version	3.137
Cooling fans	
Vendor and model	HPE 867810-B21
Number of cooling fans	6
Power supplies	
Vendor and model	HPE 865414-B21
Number of power supplies	2
Wattage of each (W)	800

How we tested

Configuring the HPE Smart Array P408i-a RAID devices

We used the HPE Smart Array P408i-a RAID controller to build the RAID devices on SATA and SAS drives by performing the following steps:

1. Log onto a server's iLO console.
2. Redirect the screen output to a browser window.
3. Press F9 during boot to get into the System Utilities Menu.
4. Select System Configuration.
5. Select HPE Smart Array P408i-a.
6. Select Array Configuration.
7. Select Create Array.
8. Select drives, and click Proceed to next form.
9. Select RAID10 for the RAID Level, and click Proceed to next form.
10. Name the array, keep the default settings, and click Submit Changes.
11. Save settings, and reboot the system to the OS installation disk in preparation of OS installation.

Installing CentOS 7

1. Insert a CentOS 7.4 bootable USB stick into the server, or mount an ISO image as a virtual CD, and boot to it.
2. Select Install or upgrade an existing system.
3. Choose the language you wish to use, and click Continue.
4. Select Installation Destination.
5. Select the desired disk for the OS.
6. Under Other Storage Options, select I will configure partitioning.
7. Click Done.
8. Select Click here to create them automatically.
9. Click Done.
10. Click Accept Changes.
11. Select Network & Hostname.
12. Enter the desired hostname for the server or VM.
13. Turn on the desired network port, and click Configure.
14. On the General tab, select Automatically connect to this network when it is available.
15. On the IPv4 Settings tab, under Method, select Manual.
16. Under Addresses, click Add, and enter the desired static IP information for the server.
17. Enter the desired DNS information.
18. Click Save, and click Done.
19. Select Date & Time, and ensure the date, time, and time zone are correct.
20. Click the cog next to the Network Time On/Off switch to add your NTP server.
21. Add the IP address of your NTP server, and click +.
22. Uncheck all other NTP servers.
23. Click OK.
24. Click Done.
25. Click Software Selection, and select Server with GUI.
26. In the Add-Ons for Select Environment section, select Compatibility Libraries, and Development Tools.
27. Click Done.
28. Click Begin Installation.
29. Select Root Password.
30. Enter the desired root password, and click Done.
31. When the installation completes, select Reboot to restart the server.

Configuring the OS on the server and the four driver VMs

We installed CentOS 7 on the server and four driver VMs, then performed the following configuration changes:

1. Log into the server as the root user.
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. Disable Hugepages:
```
echo 'never' > /sys/kernel/mm/transparent_hugepage/enabled
echo 'never' > /sys/kernel/mm/transparent_hugepage/defrag
```
5. Perform system updates:
```
sudo yum update -y
```
6. Repeat steps 1–5 on the Driver VMs.

Configuring the MDADM RAID device

Since PERC does not support NVMe drives for hardware RAID, we performed the following to create a MDADM RAID device:

1. Log into the server as the root user.
2. Install mdadm:
```
yum install -y mdadm
```
3. Create NVMe RAID volume (this assumes drives nvme{0,1,2,3}n1):
```
mdadm --zero-superblock /dev/nvme{0,1,2,3}n1
mdadm -C /dev/md/md-nvme -l10 -n4 /dev/nvme{0,1,2,3}n1
```

Preparing the RAID devices

For the purpose of the testing, we labeled and mounted the arrays as follows (note that we could not install both SATA and SAS at the same time, so they had the same raw device location):

- Raw RAID devices:
  - Backup: /dev/sdc, which is the location for the original database
  - SATA: /dev/sdd
  - SAS: /dev/sdd
  - NVMe: /dev/md127, which is the system created soft link to /dev/md/md-nvme
  - For documentation purposes, we will refer to these locations as /dev/<raw>

- Formatted RAID devices:
  - Backup: /dev/sdc1 mounted at /data/md-backkup
  - SATA: /dev/sdd1 mounted at /data/md-sata
  - SAS: /dev/sdd1 mounted at /data/md-sas
  - NVMe: /dev/md127p1 mounted at /data/md-nvme
  - For documentation purposes, we will refer to these locations as /dev/<drive> and /data/<mount>

Installing MongoDB on the server

1. 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-4.0.repo
yum install -y mongodb-org
```

2. Format, build the filesystem, and mount the database drive by typing the following commands:

```
echo -e "o\nn\np\n1\n\n\nw" | fdisk /dev/<raw>
mkfs.xfs /dev/<drive>
mount /dev/<drive> /data/<mount>
mkdir /data/<mount>/db
mkdir /data/<mount>/LOGS
```

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

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

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

```
systemLog:
path: /data/<mount>/LOG/mongod.log
storage:
dbPath: /data/<mount>/db
journal:
enabled: true
engine: "wiredTiger"
net:
port: 27017
bindIp: [your server's IP],localhost
```

5. Save the mongod configuration file.
6. Start your mongod server by typing the following command:

```
systemctl start mongod
```

7. Verify that the mongod server has started correctly by typing the following command:

```
systemctl status mongod
```

Installing YCSB on the server

1. Install Java with the following command:

```
sudo yum install java-devel
```

2. Install Maven with the following commands:

```
wget http://ftp.heanet.ie/mirrors/www.apache.org/dist/maven/maven-3/3.6.1/binaries/apache-maven-
3.6.1-bin.tar.gzsudo 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
```

3. Download and unpack YCSB into your driver with the following commands:

```
run("curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.15.0/ycsb-
0.15.0.tar.gz")
run("tar xfvz ycsb-0.15.0.tar.gz")
```

Installing YCSB on the driver VMs

We installed CentOS 7 on the Driver VMs, then performed the following steps to install YCSB:

1. 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-4.0.repo
yum install -y mongodb-org
```

2. Add the Mongo community database to your yum repo and install Java with the following command:

```
sudo yum install java-devel
```

3. Install Maven with the following commands:

```
wget "http://apache.cs.utah.edu/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-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
```

4. Download and unpack YCSB into your driver with the following commands:

```
run("curl -O --location 5https://github.com/brianfrankcooper/YCSB/releases/download/0.15.0/ycsb-
0.15.0.tar.gz")
run("tar xfvz ycsb-0.15.0.tar.gz")
```

5. Repeat steps 1 through 4 on your remaining 3 YCSB driver VMs.

Creating the YCSB database

1. Log into the MongoDB Server.
2. Copy the <path-to-ycsb>/ycsb-0.15.0/workloads/workloadc to workload_pt.
3. Edit the workload_pt file to correctly reflect the database size:

```
recordcount=1000000000
```

4. Run the following command to create a ycsb table in the database:

```
echo -e 'use ycsb\ndb.dropDatabase()\ndb.createCollection("ycsb")\nexit' | mongo.
```

5. Run the following command to build the 1.2 TB of data:

```
<path-to-ycsb>ycsb-0.15.0/bin/ycsb load mongodb-async -s -P <path-to-ycsb>ycsb-0.15.0/workloads/
workload_pt -p mongodb.writeConcern=strict
```

6. The database will finish building in several hours on the /data/md-backup mount. After the database is finished building, you are ready to test.

Configuring the workload on the YCSB drivers

1. Log into the YCSB Driver VM.
2. Copy the <path-to-ycsb>/ycsb-0.15.0/workloads/workloadc to workload_pt
3. Edit the workload_pt file to correctly reflect the test parameters:

```
recordcount=1000000000
```

Running the YCSB test

1. Log into your four YCSB Driver VMs as root.
2. Reboot the system, and log into your MongoDB server as root.
3. On the MongoDB server, delete any old database from the drives under test, and copy the database from the /data/md-backup mount to the drive type under test:

   ```
   rm -rf /data/$DRIVE/db/*
   cp -a /data/md-backup/db/. /data/$DRIVE/db
   ```

4. On the MongoDB Server, limit the memory to 200GB, and bounce the daemons:

   ```
   systemctl set-property --runtime mongod.service MemoryLimit=200G
   systemctl stop mongod
   systemctl start mongod
   ```

5. On the MongoDB Server, run the following command to clear all caches:

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

6. Verify and set the filesystem read-ahead value for consistency (for this test, we used the value of 256):

   ```
   blockdev –report  #This will list out the current values
   blockdev –setra 256 /dev/<raw>
   ```

7. On all four YCSB Driver VMs, run the following command to warm up the database:

   ```
   /root/ycsb-0.15.0/bin/ycsb run mongodb-async -s -P /root/ycsb-0.15.0/workloads/workload_pt -threads
   64 -p mongo
   db.url=mongodb://<server-ip>:27017/ycsb?w=1 -p operationcount=30000000 -p mongodb.upsert=true > ~/
   output/ycsbtest_64_threads_30000000_operations_workloadc_[your_ycsb_hostname]_runX.warmup
   ```

8. Once the database has warmed up, run the following command to test it:

   ```
   /root/ycsb-0.15.0/bin/ycsb run mongodb-async -s -P /root/ycsb-0.15.0/workloads/workload_pt -threads
   64 -p mongo
   db.url=mongodb://<server-ip>:27017/ycsb?w=1 -p operationcount=30000000 -p mongodb.upsert=true > ~/
   output/ycsbtest_64_threads_30000000_operations_workloadc_[your_ycsb_hostname]_runX.txt
   ```

9. Record the results from the test.
10. Repeat steps 1 through 9 thrice for each drive type.

Read the report at http://facts.pt/2zah70y ▶

This project was commissioned by KIOXIA.

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.