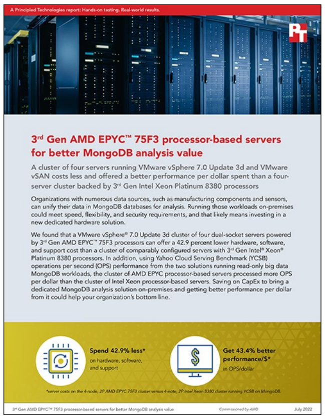

A Principled Technologies report: Hands-on testing. Real-world results.

3rd Gen AMD EPYC™ 75F3 processor-based servers for better MongoDB analysis value

A cluster of four servers running VMware vSphere 7.0 Update 3d and VMware vSAN costs less and offered a better performance per dollar spent than a four-server cluster backed by 3rd Gen Intel Xeon Platinum 8380 processors

Organizations with numerous data sources, such as manufacturing components and sensors, can unify their data in MongoDB databases for analysis. Running those workloads on-premises could meet speed, flexibility, and security requirements, and that likely means investing in a new dedicated hardware solution.

We found that a VMware vSphere® 7.0 Update 3d cluster of four dual-socket servers powered by 3rd Gen AMD EPYC™ 75F3 processors can offer a 42.9 percent lower hardware, software, and support cost than a cluster of comparably configured servers with 3rd Gen Intel® Xeon® Platinum 8380 processors. In addition, using Yahoo Cloud Serving Benchmark (YCSB) operations per second (OPS) performance from the two solutions running read-only big data MongoDB workloads, the cluster of AMD EPYC processor-based servers processed more OPS per dollar than the cluster of Intel Xeon processor-based servers. Saving on CapEx to bring a dedicated MongoDB analysis solution on-premises and getting better performance per dollar from it could help your organization's bottom line.

Spend 42.9% less* on hardware, software, and support

Get 43.4% better performance/$* in OPS/dollar

*server costs on the 4-node, 2P AMD EPYC 75F3 cluster versus 4-note, 2P Intel Xeon 8380 cluster running YCSB on MongoDB.

3rd Gen AMD EPYC™ 75F3 processor-based servers for better MongoDB analysis value   Commissioned by AMD   July 2022

The science behind the report:

3rd Gen AMD EPYC™ 75F3 processor-based servers for better MongoDB analysis value

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 3rd Gen AMD EPYC™ 75F3 processor-based servers for better MongoDB analysis value.

We concluded our hands-on testing on June 22, 2022. 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 21, 2022 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to https://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Yahoo Cloud Serving Benchmark (YCSB) - Workload C results and cost information.

	Cluster of AMD EPYC 75F3 processor-based servers	Cluster of Intel Xeon Platinum 8380 processor-based servers	Percentage difference for AMD solution
Operations per second (OPS)	369,425	450,749	-18%
Percent CPU utilization	75%	77%	N/A
Solution cost (USD)	$199,181.68	$349,269.76	42.97%
Performance per dollar (OPS/hr)/$	1.85	1.29	43.4%

For the VMware® HCI Kit 6 Enterprise and its three-year production support costs, we contacted a third-party reseller who sent us the prices on April 7, 2022. According to these calculations, our estimated three-year cost per license is:

- VMware HCI Kit 6 Enterprise + Production support for three years (Licensed for the max of 32 cores/CPU socket): $14,244.76

Based on these prices, we calculated the licensing cost for each server and cluster, which we show in Table 2.

The table also includes hardware and support cost comparisons. On February 2, 2022, we received a quote from Supermicro for the servers we tested. We added the cost of the 3.84TB drive—which we received in a quote from Supermicro on September 1, 2021—as well as the cost of the 1.92TB drive, which we retrieved from https://www.serversupply.com/SSD/NVMe/1.92TB/SAMSUNG/MZWLJ1T9HBJR_332140.htm?gclid=CjwKCAjw6dmSBhBkEiwA_W-EoAsVdrKju-vf9SUusZ6F6prAdH6Jyfar2n2_RVwHwXUb3TBmI8pJsRoCOpsQAvD_BwE on April 13, 2022.

Table 2: A comparison of licensing, hardware, and support costs in USD for the AMD EPYC and Intel Xeon processor-based solutions.

	Supermicro AS-1124US-TNRP cluster powered by AMD EPYC 75F3 processors	Supermicro SYS-620U-TNR cluster powered by Intel Xeon Platinum 8380 processors	Less cost for Supermicro AS-1124US-TNRP cluster powered by AMD EPYC 75F3 processors
VMware HCI Kit 6 Enterprise + Production support for three years (Licensed for the max of 32 cores/CPU socket)	$28,489.52	$56,979.04	$28,489.52
Per server total	**$28,489.52**	**$56,979.04**	**$28,489.52**
Per cluster total	**$113,958.08**	**$227,916.16**	**$113,958.08**
Hardware and support: Single server	$21,305.90	$30,338.40	$9,032.50
Hardware and support: Cluster	$85,223.60	$121,353.60	$36,130.00
Total cost for three years	**$199,181.68**	**$349,269.76**	**$150,088.08**

CPU utilization statistics

Figures 1 and 2 show the CPU utilization during testing for the two clusters.

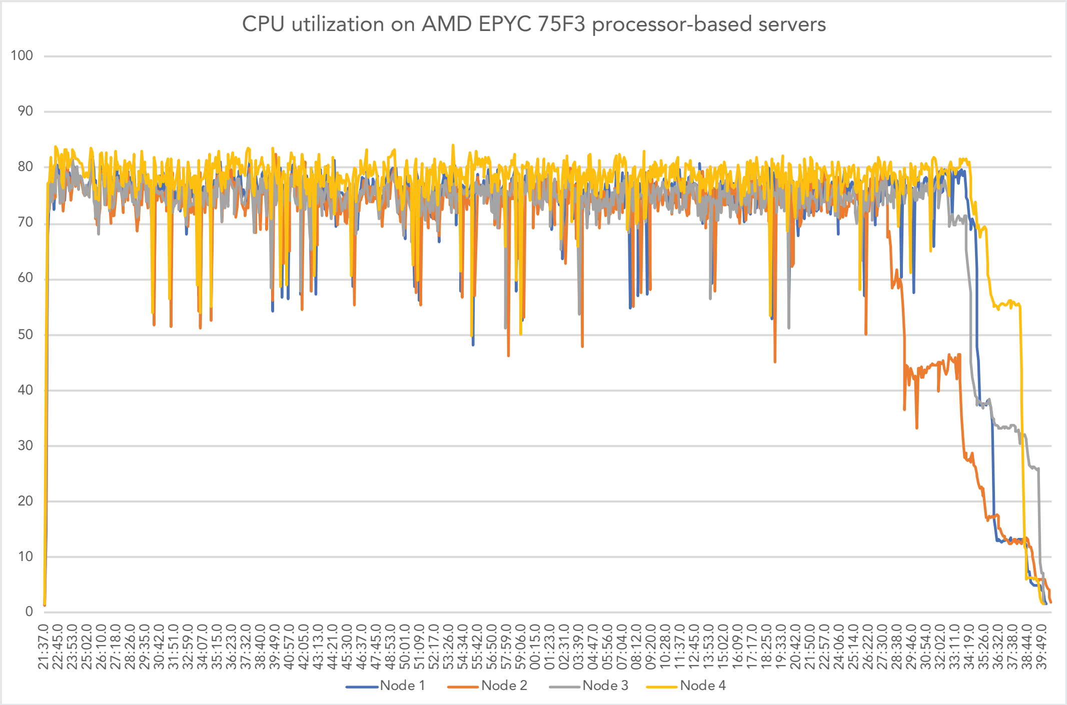

Figure 1: CPU utilization during testing for the AMD EPYC 75F3 processor-based cluster.

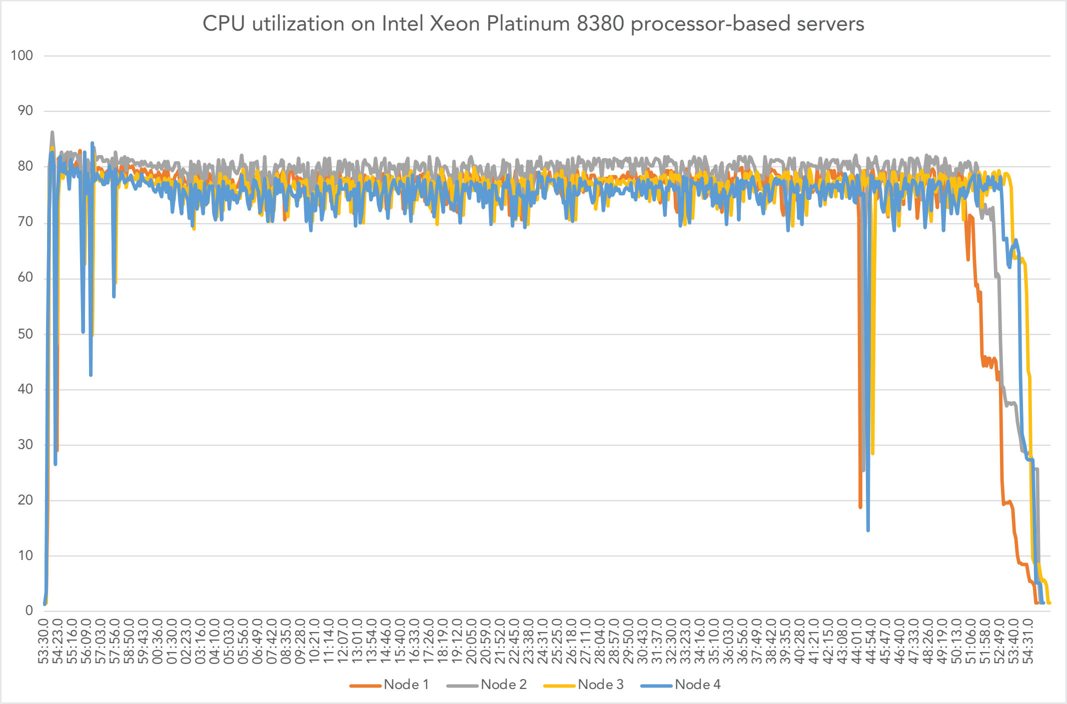

Figure 2: CPU utilization during testing for the Intel Xeon Platinum 8380 processor-based cluster.

System configuration information

Table 3: Detailed information on the systems we tested.

System configuration information	4x Supermicro AS-1124US-TNRP	4x Supermicro SYS-620U-TNR
BIOS name and version	American Megatrends Inc. 2.1	1.1A
Operating system name and version/build number	VMware ESXi-7.0U3d-19482537-standard	VMware ESXi-7.0U3d-19482537-standard
Date of last hypervisor updates/patches applied	4/27/2022	4/27/2022
Power management policy	Determinism Slider: Power	Maximum Performance
Non-default BIOS settings	NUMA Nodes Per Socket (NPS) set to 1	N/A
Processor		
Number of processors	2	2
Vendor and model	AMD EPYC 75F3	Intel Xeon Platinum 8380
Core count (per processor)	32	40
Core frequency (GHz)	2.95	2.30
Stepping	N/A	N/A
Memory module(s)		
Total memory in system (GB)	1,024	1,024
Number of memory modules	16	16
Vendor and model	Micron MTA36ASF8G72PZ-3G2E1	Micron MTA36ASF8G72PZ-3G2E1
Size (GB)	64	64
Type	PC4-3200	PC4-3200
Speed (MHz)	3,200	3,200
Speed running in the server (MHz)	3,200	3,200
Local storage (hypervisor)		
Number of drives	1	1
Drive vendor and model	Micron MTFDHBA256TCK	Intel SSD SSDSC2KB240G8
Drive size (GB)	240	240
Drive information (speed, interface, type)	PCIe® 3.0, M2 NVMe™	6Gb, SATA, SSD
Local storage (VMware vSAN™ capacity)		
Number of drives	2	2
Drive vendor and model	KIOXIA KCD6XLUL3T84	KIOXIA KCD6XLUL3T84
Drive size (TB)	3.84	3.84
Drive information (speed, interface, type)	PCIe® 4.0, NVMe™ 1.4, 64GT/s	PCIe 4.0, NVMe 1.4, 64GT/s

System configuration information	4x Supermicro AS-1124US-TNRP	4x Supermicro SYS-620U-TNR
Local storage (vSAN cache)		
Number of drives	2	2
Drive vendor and model	Samsung MZ-WLJ1T90	Samsung MZ-WLJ1T90
Drive size (TB)	1.92	1.92
Drive information (speed, interface, type)	PCIe 4.0, NVMe SSD, 2.4	PCIe 4.0, NVMe SSD, 2.4
Network adapter		
Vendor and model	Supermicro AOC-URG4N4-i4XTS	Supermicro AOC-2UR68G4-I2XT
Number and type of ports	4x 10GbE	2x 10GbE
Driver version	8.10	8.10
Network adapter		
Vendor and model	Supermicro AOC-S25G-M2S-O	Supermicro AOC-S25G-M2S-O
Number and type of ports	2x 25GbE SFP28	2x 25GbE SFP28
Driver version	14.28.2006	14.28.2006
Cooling fans		
Vendor and model	Sunon PF40561BX-Q40U-S9H	Sunon VF80381B1
Number of cooling fans	8	4
Power supplies		
Vendor and model	Supermicro PWS-1K22A-1R	Supermicro PWS-2K08A-1R
Number of power supplies	2	2
Wattage of each (W)	1,200	2,000

How we tested

Testing overview

We installed and configured the latest available version of VMware vSphere 7.0 Update 3 on a cluster of four Supermicro AS-1124US-TNRP servers with two AMD EPYC 75F3 processors each and a cluster of four Supermicro SYS-620U-TNR servers with two Intel Xeon Platinum 8380 processors each. We installed the hypervisor on internal SSD drives. On each cluster, we configured and created a 28TB VMware vSAN™ datastore using four NVMe drives on each server. Each datastore used two disk groups: One 1.92 TB drive for cache and one 3.84TB drive for capacity. The vSAN datastore served as a shared datastore for the virtualized MongoDB cluster. We deployed a MongoDB cluster with three config server VMs, 16 mongos VMs, and 24 mongod VMs. On the AMD processor-based cluster hosts, each mongos VM had 24 vCPUs and 32 GB of memory. On the Intel processor-based cluster hosts, each mongos VM had 32 vCPUs and 32 GB of memory. We configured the config and mongod VMs the same across the AMD and Intel platforms with 8 vCPUs and 8 GB of RAM for the config VMs and 16 vCPU and 128 GB of RAM for the mongod VMs. We ran Workload C from Yahoo Cloud Serving Benchmark and measured OPS and CPU utilization for each cluster.

We used a Dell Networking 1Gb X1052 switch for the VMware management network. We used a Dell 5048-ON 25Gb switch for vMotion®, vSAN, and the VM network. On the 25Gb switch, we configured the following VLANs for the AMD EPYC processor-based cluster: 20 for the vMotion network, and 30 for the vSAN network. We configured the following VLANs for the Intel processor-based cluster: 40 for the vMotion network and 60 for the vSAN. We also configured VLAN 2 for the VM network, which the Intel Xeon processor-based and AMD EPYC processor-based clusters shared for VM traffic. We configured a port channel for each host using the two 25Gb ports from each host. We set the 25Gb switch MTU setting to 9216. We configured a two port LAG on a Distributed vSwitch using two 25Gb ports from each host connected to the 25Gb switch. On the Distributed vSwitch, we created three port groups: vSAN, vMotion, and workload.

The following sections describes the steps we took to configure the test environment and run the test.

Installing vSphere 7.0 Update 3d on a Supermicro server

1. Download ESXi 7.0 Update 3d from the following link: https://my.vmware.com/group/vmware/evalcenter?p=vsphere-eval-7#tab_download.
2. Open a new browser tab, and connect to the IP address of the Supermicro server BMC.
3. Log in with the BMC credentials.
4. In the main screen, click Launch Virtual Console.
5. In the console menu bar, select Virtual Media, and select Virtual Storage.
6. From the drop-down menu in the Virtual Storage pop-up window, select ISO, and click Open Image to browse your local computer and select the image you downloaded in step 1.
7. To mount the ISO image, click Plug In, and click OK.
8. On the console menu bar, click the Power Control, and select Power Reset.
9. The system will boot to the mounted image, and the Loading ESXi installer screen will appear. When prompted, press Enter to continue.
10. To Accept the EULA and Continue, press F11.
11. Select the storage device to target for installation. We selected the internal SD card. To continue, press Enter.
12. To confirm the storage target, press Enter.
13. Select the keyboard layout, and press Enter.
14. Provide a root password, and confirm it. To continue, press Enter.
15. To install vSphere, press F11.
16. To reboot the server upon completion, press Enter.
17. Complete the previous steps for each server under test.

Installing VMware vCenter Server Appliance 7.0 Update 3d

1. Download vCenter 7.0 Update 3d from the VMware support portal: https://my.vmware.com.
2. Mount the image on your local system, and browse to the vcsa-ui-installer folder. Expand the folder for your OS, and launch the installer if it doesn't automatically begin.
3. When the vCenter Server Installer wizard opens, click Install.
4. To begin installation of the new vCenter server appliance, click Next.
5. Check the box to accept the license agreement, and click Next.
6. Enter the IP address of the infrastructure server with ESXi 7.0 Update 2. Provide the root password, and click Next.
7. To accept the SHA1 thumbprint of the server's certificate, click Yes.
8. Accept the VM name, and provide and confirm the root password for the VCSA. Click Next.
9. Set the size for the environment you're planning to deploy. We selected Medium. Click Next.
10. Select the datastore on which to install vCenter. Accept the datastore defaults, and click Next.

11. Enter the FQDN, IP address information, and DNS servers you want to use for the vCenter server appliance. Click Next.
12. To begin deployment, click Finish.
13. Once Stage 1 completes, click Close. To confirm, click Yes.
14. Open a browser window, and connect to https://[vcenter.FQDN:5480/.
15. On the Getting Started - vCenter Server page, click Set up.
16. Enter the root password, and click Log in.
17. Click Next.
18. Enable SSH access, and click Next.
19. To confirm the changes, click OK.
20. Type `vsphere.local` for the Single Sign-On domain name. Enter a password for the administrator account, confirm it, and click Next.
21. Click Next.
22. Click Finish.

Creating a vSphere cluster in vCenter

1. Open a browser, and enter the address of the vCenter server you deployed. For example: `https://[vcenter.FQDN]/ui`.
2. In the left panel, select the vCenter server, right-click, and select New Datacenter.
3. Provide a name for the new data center, and click OK.
4. Select the data center you just created, right-click, and select New Cluster.
5. Give a name to the cluster, and enable vSphere DRS. Click OK.
6. In the cluster configuration panel, under Add hosts, click Add.
7. Check the box for Use the same credentials for all hosts. Enter the IP address and root credentials for the first host, and enter the IP addresses of all remaining hosts. Click Next.
8. To select all hosts, check the box beside Hostname/IP Address. Click OK.
9. Click Next.
10. Click Finish.

Creating a distributed vSwitch and port group

1. From vSphere client, click Home→Networking.
2. Select your Datastore.
3. In the Actions drop-down menu in the right panel, select Distributed vSwitch→New Distributed vSwitch.
4. Give your vSwitch a name, or accept the default. Click Next.
5. Select 7.0.2 - ESXi 7.0.2 and later as the version, and click Next.
6. Select the number of uplinks per ESXi host you'll give to the vSwitch. We selected 2. Click Next.
7. Click Finish.
8. Select new DSwitch, and click Configure.
9. Select Properties, and click Edit.
10. Under advanced, set MTU (Bytes) to 9,000.
11. Click OK.
12. Select LACP, click + NEW, and set mode to Active.
13. Click OK.
14. Enter an appropriate name, set ports to two, and click OK.
15. Right-click the new DSwitch, and select Add and Manage Hosts.
16. Leave Add hosts selected, and click Next.
17. To add new hosts, click +.
18. To select all the hosts in your target cluster, check the box beside Host. Click OK. Click Next.
19. Select the NIC you want to use for this DSwitch, and click Assign Uplink.
20. Select the first LAG port.
21. Check the box for Apply this uplink assignment to the rest of the hosts. Click OK.
22. Complete steps 16 through 18 for the second NIC, and assign it to the second LAG port.
23. Click Next.
24. Do not assign VMkernel adapters at this time. Click Next.
25. Do not migrate any VM networking at this time. Click Next.
26. Click Finish.
27. Right-click the DvSwitch, and select Distributed Port Group→New Distributed Port Group.
28. Name it `Workload Network`, and click Next.

29. Change the VLAN type to VLAN, and set the VLAN ID to VLAN 2. Click Next.
30. Click Finish.
31. On the AMD EPYC processor-based cluster, complete steps 17 through 20 two times using the following settings:
    - Name: vMotion; VLAN ID 20
    - Name: vSAN; VLAN ID 30
32. For the Intel Xeon processor-based cluster, use the following port group settings:
    - Name: vMotion; VLAN ID 40
    - Name vSAN; VLAN ID 60

Creating a VMkernel Adapter

Note: We completed these steps on each server under test for the private network, vSAN network, and vMotion network using the appropriate IPv4 settings for each network and enabling the corresponding service for each VMkernel adapter.

1. From the vCenter UI in the Hosts and Clusters view, right-click a host, and click Add Networking.
2. Select VMkernel Network Adapter, and click Next.
3. Click Browse, and select existing network.
4. Select the desired Distributed Port Group that you created in the previous section.
5. If necessary, assign a different network label.
6. Select the desired service or services for the VMkernel, and click Next.
7. Select Use static IPv4 settings, and enter the desired IPv4 address, subnet mask, and default gateway.
8. Click Next.
9. Click Finish.

Configuring the vSAN datastore

1. In the left panel of the vCenter, right-click the test cluster, select vSAN, and select configuration.
2. Select single site cluster, and click Next.
3. In the Services screen, click Next.
4. In the Claim disks screen, click Group by, and select Host.
5. Select disks for each host. We selected one drive for cache and two drives for capacity. Click Next.
6. In the Review screen, click Finish.
7. Click the test cluster from vCenter, and in the right panel, click the Configure tab .
8. In the vSAN section, select Services.
9. To create a distributed switch, in the Configure Cluster section, click Configure.
10. In the Physical adapters section, choose two uplink adapters for the distributed switch, and click Next.
11. In the Storage traffic screen, set the VLAN ID assigned for vSAN, choose Static IPs, and fill in the static IP address, Subnet mask, and default gateway. Click Next.
12. In the Advanced options screen, click Next.
13. In the Claim disks screen, click Next.
14. In the Proxy settings screen, click Next.
15. Click Finish.

Configuring of the VMs for testing

Table 4: The configuration for the Supermicro AS-1124US-TNRP VMs we tested.

VM type	Number of VMs	vCPUs	vRAM	VMDKs
Config server	3	8	8 GB	1 x 50 GB (OS) 1 x 20 GB (DB)
Mongos server	16	24	32 GB	1 x 50 GB (OS)
Mongod server	24	12	128 GB	1 x 50 GB (OS) 1 x 200 GB (DB)

Table 5: The configuration for the Supermicro SYS-620U-TNR VMs we tested.

VM type	Number of VMs	vCPUs	vRAM	VMDKs
Config server	3	2	8 GB	1 x 50 GB (OS) 1 x 20 GB (DB)
Mongos server	16	32	32 GB	1 x 50 GB (OS)
Mongod server	24	12	128 GB	1 x 50 GB (OS) 1 x 200 GB (DB)

Creating CentOS Linux 8.4 VMs

1. Create a new VM with an appropriate name, and assign it to the proper Host & Datastore.
2. Edit the VM, and connect the CD/DVD drive to a Centos 8 ISO in your local datastore.
3. Power on the VM, and open its console.
4. Select Install CentOS Linux 8.
5. Choose the language you wish to use, and click Continue.
6. Select Installation Destination.
7. Select the desired disk for the OS.
8. Under Other Storage Configuration, select 'Custom'.
9. Click Done.
10. Select "click here to create them automatically".
11. Remove the / partition.
12. Set the swap partition to 20GiB.
13. Assign all remaining free space to the / partition.
14. Click Done.
15. Click Accept Changes.
16. Select Kdump.
17. Uncheck Enable Kdump, and click Done.
18. Select 'Network & Hostname'.
19. Enter the desired hostname for the VM, and click 'Apply'.
20. Turn on the Ethernet port, and click 'Configure'.
21. On the General tab, select 'Automatically connect to this network when it is available'.
22. Click save, and click Done.
23. Select Date & Time, and ensure the correct date, time, and time zone are set.
24. To add your NTP server, click the cog next to 'Network Time' On/Off switch.
25. Add the IP address of your NTP server, and click +.
26. Uncheck all other NTP servers.
27. Click 'OK' then 'Done'.
28. Select 'Root Password'.
29. Enter the desired root password, and click 'Done'.

30. Select ‘Software Selection’.
31. Select ‘Server’ as the Base Environment.
32. From the ‘Additional software for Selected Environment’, choose Windows File Server.
33. Click ‘Begin Installation’.
34. When the installation completes, to restart the server, select Reboot.
35. Back on the VC console, edit the VM’s settings to change the CD/DVD to ‘client device’.

Installing and configuring MongoDB on the config server VMs

We installed Centos 8.4 onto each VM, then performed the following steps on each config server VM:

1. Log into the VM you are configuring.
2. Configure the system to log in without prompting for a password:

```
mkdir -p /root/.ssh
chmod 700 /root/.ssh
cd /root/.ssh
ssh-keygen -t rsa -q
cp id_rsa.pub authorized_keys
echo "StrictHostKeyChecking=no" > config
```

3. Stop and disable the firewall service:

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

4. Set SELinux to disabled:

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
```

5. Disable transparent hugepages:

```
grub2-editenv - set "$(grub2-editenv - list | grep kernelopts) transparent_hugepage=never"
```

6. Add all host information for all systems into the /etc/hosts file.
7. Reboot the server.
8. Perform system updates:

```
dnf update -y
```

9. Create the mogodb-org.repo file:

```
touch /etc/yum.repos.d/mongodb-org-5.0.repo
```

10. Populate the mongodb-org-5.0.repo file:

```
nano /etc/yum.repos.d/mongodb-org.repo
```

11. Add the following:

```
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

12. Install MongoDB:

```
yum install -y mongodb-org
```

13. Add the database drive to the VM, and create the database folders for MongoDB:

```
mkfs.xfs /dev/sdb

mount /dev/sdb /mntmnt/

echo '/dev/sdb  /mnt xfs  noatime  0  0' >> /etc/fstab
chown -R mongod:mongod /mnt/var/lib/mongoEdit the MongoDB config file at /etc/mongod.conf to match the
Config Server mongod.conf file in the Scripts we used in our testing section.
```

14. Edit the mongod config file at /etc/mongod.conf to match the Mongod Server mongod.conf file in the Scripts we used in our testing section.
15. Edit the system limits file at /etc/security/limits.conf to match the limits.conf file in the Scripts we used in our testing section.
16. Edit the sysctl.conf file at /etc/sysctl.conf to match the sysctl.conf file in the Scripts we used in our testing section.
17. Edit the rc.local file at /etc/rc.d/rc.local to match the rc.local file in the Scripts we used in our testing section.
18. Change permissions on rc.local to execute at VM startup

```
chmod +x /etc/rc.d/rc.local
```

19. Install numactl

```
yum install numactl -y
```

20. Edit the mongod service file at /etc/systemd/system/multi-user.target.wants/mongod.service to match the mongod.service file in the Scripts we used in our testing section.
21. Add `iommu=pt` to the end of the line starting with GRUB_CMDLINE_LINUX, and run the following command to create a new grub config file

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

22. Reboot.
23. Start your mongod server:

```
systemctl start mongod
```

24. Enable the mongod process to start automatically when the server restarts:

```
systemctl enable mongod
```

25. Repeat steps 1 through 25 on your remaining two Config Server VMs.

Creating the config server replica set on the config server VMs

1. Log into all three config servers and verify that all three are “active (running)”.
2. Type mongo to enter the MongoDB console on all three Config Servers.
3. In the MongoDB console of ONLY ONE of the Config Servers, create the replica set:

```
rs.initiate(
  {
    _id : "config-replica-set",
    members: [
      { _id : 0, host : "mongo-config-1:27017" },
      { _id : 1, host : "mongo-config-2:27017" },
      { _id : 2, host : "mongo-config-3:27017" }
    ]
  }
)
```

4. Verify that the config-replica-set has been created by clicking the enter key several times on all three config servers. The Command line should change to one of the following:

```
config-replica-set:PRIMARY>
config-replica-set:SECONDARY> (2 of these will be present)
```

Installing and configuring MongoDB on the Mongos server VMs

We installed Centos 8.4 onto each VM, then performed the following steps on each Mongos server VM:

1. Log into the VM you are configuring.
2. Configure the system to log in without prompting for a password:

```
mkdir -p /root/.ssh
chmod 700 /root/.ssh
cd /root/.ssh
ssh-keygen -t rsa -q
cp id_rsa.pub authorized_keys
echo "StrictHostKeyChecking=no" > config
```

3. Stop and disable the firewall services:

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

4. Set SELinux to disabled:

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
```

5. Disable transparent hugepages:

```
grub2-editenv - set "$(grub2-editenv - list | grep kernelopts) transparent_hugepage=never"
```

6. Add all host information for all systems into the /etc/hosts file.
7. Reboot the server.
8. Perform system updates:

```
dnf update -y
```

9. Create the mogodb-org.repo file.

```
touch /etc/yum.repos.d/mongodb-org-5.0.repo
```

10. Populate the mongodb-org-5.0.repo file:

```
nano /etc/yum.repos.d/mongodb-org.repo
```

11. Add the following:

```
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

12. Install MongoDB.

```
yum install -y mongodb-org
```

13. Edit the mongod config file at /etc/mongod.conf to match the Mongos Server mongod.conf file in the Scripts we used in our testing section.
14. Edit the rc.local file at /etc/rc.d/rc.local to match the mongos rc.local file in the Scripts we used in our testing section.Edit the system limits file at /etc/security/limits.conf to match the limits.conf file in the Scripts we used in our testing section.
15. Edit the sysctl.conf file at /etc/sysctl.conf to match the sysctl.conf file in the Scripts we used in our testing section.
16. Add `iommu=pt` to the end of the line starting with GRUB_CMDLINE_LINUX, and run the following command to create a new grub config file

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

17. Reboot.Login and type `mongosh` to enter the mongosh console.
18. Verify that the MongoDB console starts in the mongos console. The Command line should look like the following:

```
[direct: mongos] test>Repeat steps 1-20 on your remaining 7 Mongos Server VMs.
```

Installing and configuring MongoDB on the Mongod server VMs

We installed Centos 8.4 onto each VM, then performed the following steps on each Mongod server VM:

1. Log into the VM you are configuring.
2. Configure the system to log in without prompting for a password:

```
mkdir -p /root/.ssh
chmod 700 /root/.ssh
cd /root/.ssh
ssh-keygen -t rsa -q
cp id_rsa.pub authorized_keys
echo "StrictHostKeyChecking=no" > config
```

3. Stop and disable the firewall services:

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

4. Set SELinux to disabled:

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
```

5. Disable transparent hugepages:

```
grub2-editenv - set "$(grub2-editenv - list | grep kernelopts) transparent_hugepage=never"
```

6. Add all host information for all systems into the /etc/hosts file.
7. Reboot the server.
8. Perform system updates:

```
dnf update -y
```

9. Create the mogodb-org.repo file.

```
touch /etc/yum.repos.d/mongodb-org-5.0.repo
```

10. Populate the mongodb-org-5.0.repo file:

```
nano /etc/yum.repos.d/mongodb-org.repo
```

11. Add the following:

```
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

12. Install MongoDB.

```
yum install -y mongodb-org
```

13. Add the database drive to the VM, and create the database folders for MongoDBs:

```
mkfs.xfs /dev/sdb
mnt

mkdir -p /mnt/var/lib/mongo
echo '/dev/sdb  /mnt xfs  noatime  0  0' >> /etc/fstab
chown -R mongod:mongod /mnt/var/lib/mongo
```

14. Edit the mongod config file at /etc/mongod.conf to match the Mongod Server mongod.conf file in the Scripts we used in our testing section.
15. Edit the system limits file at /etc/security/limits.conf to match the limits.conf file in the Scripts we used in our testing section.
16. Edit the sysctl.conf file at /etc/sysctl.conf to match the sysctl.conf file in the Scripts we used in our testing section.
17. Edit the rc.local file at /etc/rc.d/rc.local to match the rc.local file in the Scripts we used in our testing section.
18. Change permissions on rc.local to execute at VM startup.

```
chmod +x /etc/rc.d/rc.local
```

19. Install numactl.

```
yum install numactl -y
```

20. Edit the mongod service file at /etc/systemd/system/multi-user.target.wants/mongod.service to match the mongod.service file in the Scripts we used in our testing section.
21. Add `iommu=pt` to the end of the line starting with GRUB_CMDLINE_LINUX, and run the following command to create a new grub config file.

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

22. Reboot.
23. Start your mongod server:

```
systemctl start mongod
```

24. Enable the mongod process to start automatically when the server restarts.

```
systemctl enable mongod
```

25. Repeat steps 1-24 on your remaining 23 Mongod Server VMs.

Creating the config server replica set on the Mongod server VMs

1. Log into the three Mongod servers that will make up a three-node replica set, and verify that all three are "active (running)".
2. Type `mongo` to enter the MongoDB console on all three Mongod servers.
3. In the MongoDB console of only one of the Mongod servers, create the replica set:

```
rs.initiate(
  {
    _id : "shard-replica-set-##",
    members: [
      { _id : 0, host : "mongod-##-a:27017" },
      { _id : 1, host : "mongod-##-b:27017" },
      { _id : 2, host : "mongod-##-c:27017" }
    ]
  }
)
```

4. Verify that the config-replica-set has been created by clicking the enter key several times on all three Mongod servers. The Command line should change to one of the following:

```
config-replica-set:PRIMARY>
config-replica-set:SECONDARY> (2 of these will be present)
```

5. Repeat steps 1 through 4 until all 24 servers are configured, using three Mongod servers per shard-server-set for a total of eight shard-server-sets.

Enabling sharding operations on the Mongos server VMs

You must enable sharding on the Mongos server(s) prior to building the database.

1. Log into the first Mongos server VM you are configuring.
2. Type `mongosh` to enter the MongoDB console on the Mongos server.
3. Configure two shards on the first Mongos server:

```
sh.addShard("shard-replica-set-1/<mongod-1-a:27017, mongod-1-b:27017, mongod-1-c:27017")
```

4. Repeat steps 1 through 3 for the remaining seven mongos VMs and mongod replica sets.

Installing and configuring YCSB driver VMs

We installed CentOS 8.4 onto each VM, then performed the following steps on each YCSB VM:

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

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

3. Set SELinux to disabled:

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
```

4. Disable transparent hugepages:

```
grub2-editenv - set "$(grub2-editenv - list | grep kernelopts) transparent_hugepage=never"
```

5. Add all host information for all systems into the /etc/hosts file.
6. Reboot the server.
7. Perform system updates:

```
dnf update -y
```

8. Create the mogodb-org.repo file:

```
touch /etc/yum.repos.d/mongodb-org-5.0.repo
```

9. Populate the mongodb-org-5.0.repo file:

```
nano /etc/yum.repos.d/mongodb-org.repo
```

10. Add the following:

```
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

11. Install MongoDB:

```
yum install -y mongodb-org
```

12. Start your mongod server:

```
systemctl start mongod
```

13. Enable the mongod process to start automatically when the server restarts:

```
systemctl enable mongod
```

14. Install prerequisites for Maven:

```
dnf install nano wget curl nmon java-devel -y
```

15. Install Maven:

```
wget https://mirrors.sonic.net/apache/maven/maven-3/3.8.1/binaries/apache-maven-3.8.1-bin.tar.gz
tar xzf apache-maven-3.8.1-bin.tar.gz -C /usr/local
cd /usr/local/
ln -s apache-maven-* maven
echo "export M2_HOME=/usr/local/maven" >> /etc/profile.d/maven
echo "export PATH=${M2_HOME}/bin:${PATH}" >> /etc/profile.d/maven
```

16. Download and install YCSB:

```
curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.17.0/
ycsb-0.17.0.tar.g
tar xfvz ycsb-0.17.0.tar.gz
```

17. Repeat steps 1 through 16 on your remaining seven YCSB VMs.

Creating the YCSB database

1. Log into one of the YCSB driver VMs.
2. Create a database with 500 million records:

```
./bin/ycsb load mongodb -s -P /root/ycsb-0.17.0/workloads/workloadc -threads 32 -p mongodb.
url=mongodb://mongos1:27017/ycsb?w=1 -p recordcount=500000000
```

3. As the database is being created, log into one of your Mongos Server VMs.
4. Enter MongoDB mongos console:

```
mongosh
```

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, 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

We used a script to kick off the run on all eight YCSB VMs simultaneously.

1. Log into the first YCSB VM.
2. To kick off the test, run the following script (see the Scripts we used in our testing section).

```
./run_legacy_mongodb.sh
```

Scripts we used in our testing

We used the following scripts to configure and tune our MongoDB deployment. The config-sysctl.conf, conifg-limits.conf, and config-grub files apply to the config, mongod, and mongos VMs. The config-rc.local and config-fstab files apply to only the config and mongod VMs. The mongos-rc.local file applies only to the mongos VMs.

limits.conf

```
# /etc/security/limits.conf
#
#This file sets the resource limits for the users logged in via PAM.
#It does not affect resource limits of the system services.
#
#Also note that configuration files in /etc/security/limits.d directory,
#which are read in alphabetical order, override the settings in this
#file in case the domain is the same or more specific.
#That means for example that setting a limit for wildcard domain here
#can be overriden with a wildcard setting in a config file in the
#subdirectory, but a user specific setting here can be overriden only
#with a user specific setting in the subdirectory.
#
#Each line describes a limit for a user in the form:
#
#<domain>        <type>  <item>  <value>
#
#Where:
#<domain> can be:
#        - a user name
#        - a group name, with @group syntax
#        - the wildcard *, for default entry
#        - the wildcard %, can be also used with %group syntax,
#                 for maxlogin limit
#
#<type> can have the two values:
#        - "soft" for enforcing the soft limits
#        - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#        - core - limits the core file size (KB)
#        - data - max data size (KB)
#        - fsize - maximum filesize (KB)
#        - memlock - max locked-in-memory address space (KB)
#        - nofile - max number of open file descriptors
#        - rss - max resident set size (KB)
#        - stack - max stack size (KB)
#        - cpu - max CPU time (MIN)
#        - nproc - max number of processes
#        - as - address space limit (KB)
#        - maxlogins - max number of logins for this user
#        - maxsyslogins - max number of logins on the system
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#        - nice - max nice priority allowed to raise to values: [-20, 19]
#        - rtprio - max realtime priority
#
#<domain>      <type>  <item>         <value>
#

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4

# End of file

*        -        nofile        64000
*        -        cpu           unlimited
*        -        nproc         64000
*        -        fsize         unlimited
*        -        memlock       unlimited
*        -        as         unlimited
*        -        rss           unlimited
```

sysctl.conf

```
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).

vm.swappiness=1
vm.zone_reclaim_mode=0 vm.dirty_ratio = 15
vm.dirty_background_ratio = 5
vm.max_map_count=128000
net.core.somaxconn = 4096
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 120
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_keepalive_probes = 6
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.core.rmem_default=16777216
net.core.wmem_default=16777216
net.core.optmem_max=40960
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
```

mongod.service

```
[Unit]
Description=MongoDB Database Server
Documentation=https://docs.mongodb.org/manual
After=network-online.target
Wants=network-online.target

[Service]
User=mongod
Group=mongod
Environment="OPTIONS=-f /etc/mongod.conf"
EnvironmentFile=-/etc/sysconfig/mongod
ExecStart=/usr/bin/numactl --interleave=all /usr/bin/mongod $OPTIONS
ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb
ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb
ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb
PermissionsStartOnly=true
PIDFile=/var/run/mongodb/mongod.pid
Type=forking
# file size
LimitFSIZE=infinity
# cpu time
LimitCPU=infinity
# virtual memory size
LimitAS=infinity
# open files
LimitNOFILE=64000
# processes/threads
```

```
LimitNPROC=64000
# locked memory
LimitMEMLOCK=infinity
# total threads (user+kernel)
TasksMax=infinity
TasksAccounting=false
# Recommended limits for mongod as specified in
# https://docs.mongodb.com/manual/reference/ulimit/#recommended-ulimit-settings

[Install]
WantedBy=multi-user.target
```

rc.local (Config & Mongod VMs)

```
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local

ethtool -G ens192 rx 4096 tx 4096
echo 1024 > /sys/block/sdb/queue/nr_requests
echo 0 > /sys/class/block/sdb/queue/rotational
```

rc.local (Mongos VMs)

```
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local

mkdir /var/run/mongodb
chown -R mongod:mongod /var/run/mongodb
mongos --config=/etc/mongod.conf

ethtool -G ens192 rx 4096 tx 4096
```

mongod.conf (Config VMs)

```
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /mnt/var/lib/mongo
  journal:
    enabled: true
  engine: "wiredTiger"
#  wiredTiger:

# how the process runs
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile
  timeZoneInfo: /usr/share/zoneinfo

# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0 #0.0.0.0 binds all


#security:

#operationProfiling:

replication:
    replSetName: config-replica-set

sharding:
   clusterRole:      configsvr

## Enterprise-Only Options

#auditLog:

#snmp:
```

Mongod.conf (Mongos VMs)

```
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
#storage:
#  dbPath: /var/lib/mongo
#  journal:
#    enabled: true
#  engine:
#  wiredTiger:
```

```
# how the process runs
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile
  timeZoneInfo: /usr/share/zoneinfo

# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0


#security:

#operationProfiling:

#replication:

sharding:
    configDB: config-replica-set/mongo-config-1:27017,mongo-config-2:27017,mongo-config-3:27017

## Enterprise-Only Options

#auditLog:

#snmp:
```

Mongod.conf (Mongod VMs)

```
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /mnt/var/lib/mongo
  journal:
    enabled: true
  engine: "wiredTiger"
#  wiredTiger:

# how the process runs
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile
  timeZoneInfo: /usr/share/zoneinfo

# network interfaces
net:
  port: 27017
  bindIp: 10.210.100.151,localhost


#security:

#operationProfiling:

replication:
  replSetName:       shard-replica-set-1

sharding:
  clusterRole:       shardsvr
```

```
## Enterprise-Only Options

#auditLog:

#snmp:
```

ycsb{1-16}-1.sh

Called by run_legacy_mongodb.sh to kickoff all VMs simultaneously

```
#!/bin/bash

/usr/local/ycsb-0.17.0/bin/ycsb run mongodb -s -P /usr/local/ycsb-0.17.0/workloads/workloadc -threads
48 -p mongodb.url="mongodb://mongos1:27017/ycsb?readPreference=secondary&compressors=snappy" -p
operationcount=100000000 > ./ycsb1-1.out
```

run_legacy_mongodb.sh

```
#!/bin/bash

screen -d -m ~/ycsb1-1.sh

ssh ycsb2 'screen -d -m ~/ycsb2-1.sh'
ssh ycsb3 'screen -d -m ~/ycsb3-1.sh'
ssh ycsb4 'screen -d -m ~/ycsb4-1.sh'
ssh ycsb5 'screen -d -m ~/ycsb5-1.sh'
ssh ycsb6 'screen -d -m ~/ycsb6-1.sh'
ssh ycsb7 'screen -d -m ~/ycsb7-1.sh'
ssh ycsb8 'screen -d -m ~/ycsb8-1.sh'
ssh ycsb9 'screen -d -m ~/ycsb9-1.sh'
ssh ycsb10 'screen -d -m ~/ycsb10-1.sh'
ssh ycsb11 'screen -d -m ~/ycsb11-1.sh'
ssh ycsb12 'screen -d -m ~/ycsb12-1.sh'
ssh ycsb13 'screen -d -m ~/ycsb13-1.sh'
ssh ycsb14 'screen -d -m ~/ycsb14-1.sh'
ssh ycsb15 'screen -d -m ~/ycsb15-1.sh'
ssh ycsb16 'screen -d -m ~/ycsb16-1.sh'
```

Read the report at https://facts.pt/b20e3Vm ▶

This project was commissioned by AMD.

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.