Ceph Benchmark

Hyper-converged infrastructure with Proxmox VE virtualization platform and integrated Ceph Storage.

To optimize performance in hyper-converged deployments, with Proxmox VE and Ceph storage, the appropriate hardware setup is essential. This benchmark presents possible setups and their performance outcomes, with the intention of supporting Proxmox users in making better decisions.

EXECUTIVE SUMMARY

Hyper-converged setups can be deployed with Proxmox VE, using a cluster that contains a minimum of three nodes, enterprise class NVMe SSDs, and a 100 gigabit network (10 gigabit network is the absolute minimum requirement and already a bottleneck). As long as CPU power and RAM are sufficient, a three node cluster can reach reasonably good levels of performance.

- Since by default Ceph uses a replication of three, data will remain available, even after losing a node, thus providing a highly available, distributed storage solution—fully software-defined and 100 % open-source.
- Although it is possible to run virtual machines/containers and Ceph on the same node, a separation makes sense for larger workloads.
- To match your need for growing workloads, a Proxmox VE and Ceph server cluster can be extended with additional nodes on the fly, without any downtime.
- The Proxmox VE virtualization platform has integrated Ceph storage, since the release of Proxmox VE 3.2, in early 2014. Since then, it has been used on thousands of servers worldwide, which has provided us with an enormous amount of feedback and experience.

TABLE OF CONTENTS

TEST BED CONFIGURATION

All benchmarks summarized in this paper were conducted in August and September 2020, on standard server hardware, with a default Proxmox VE/Ceph server installation. The following section describes the testbed configuration.

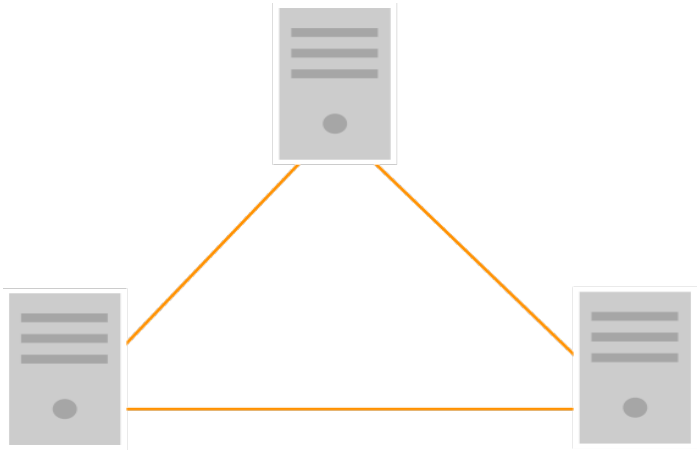
SERVER HARDWARE

For the benchmarks we used 3 identical servers, with the below specifications:

CPU:	Single AMD EPYC 7302P 16-Core Processor
Mainboard:	GIGABYTE MZ32-AR0-00
Case:	2U Supermicro Chassis 8x Hotswap
Dual 1 Gbit NIC:	Intel I350 (on board)
Dual 100 Gbit NIC:	Mellanox MCX456A-ECAT ConnectX-4, x16 PCIe 3.0
Memory:	8 x 16 GB DDR4 FSB3200 288-pin 2Rx8 Samsung M393A2K43DB3-CWE
Disk:	4 x Micron 9300 Max 3.2 TB (MTFDHAL3T2TDR)

NETWORK

All nodes were directly connected with 100 GbE DACs, in a full-mesh topology. This setup allows the nodes to communicate without an additional switch. This benefits the overall cost of the cluster and reduces the latency.



SOFTWARE VERSION (AUG/SEP 2020)

This benchmark was conducted using Proxmox VE 6.2, Ceph Octopus 15.2.4 (Appendix, 2.) and the latest Mellanox OFED + firmware.

SSDS FOR CEPH OSD

It's essential to use reliable, enterprise-class SSDs, with high endurance and power-loss protection. We also recommend testing each SSD's write performance with the Flexible I/O tester (fio), before using them as Ceph OSD devices.

The following table shows fio write results from a traditional spinning disk and from various selected SSDs:

	Bandwidth (KB)	4K IO/s	Latency (ms)
Intel Optane SSD DC P4800X Series 375 GB	322931	80732	0.01
Intel SSD DC P3700 Series 800 GB	300650	75162	0.01
Micron 9300 MAX 3.2 TB	205824	51000	0.02
Samsung SM863 240GB 2.5inch SSD	69942	17485	0.06
Intel DC S3510 120GB	50075	12518	0.08
Intel DC S3500 120GB	48398	12099	0.08
Samsung SSD 850 EVO 1TB	1359	339	2.94
Crucial MX100 512GB	1017	254	3.93
Seagate Constellation 7200.2 500 GB	471	117	8.47

Based on these fio tests, we decided to use 12 x Micron 9300 MAX (MTFDHAL3T2TDR), 2.5", 3.20 TB U.2 SSD. We connected 4 U.2 SSDs per server, using the on board SLink connectors.

We used the following fio command for the device tests:

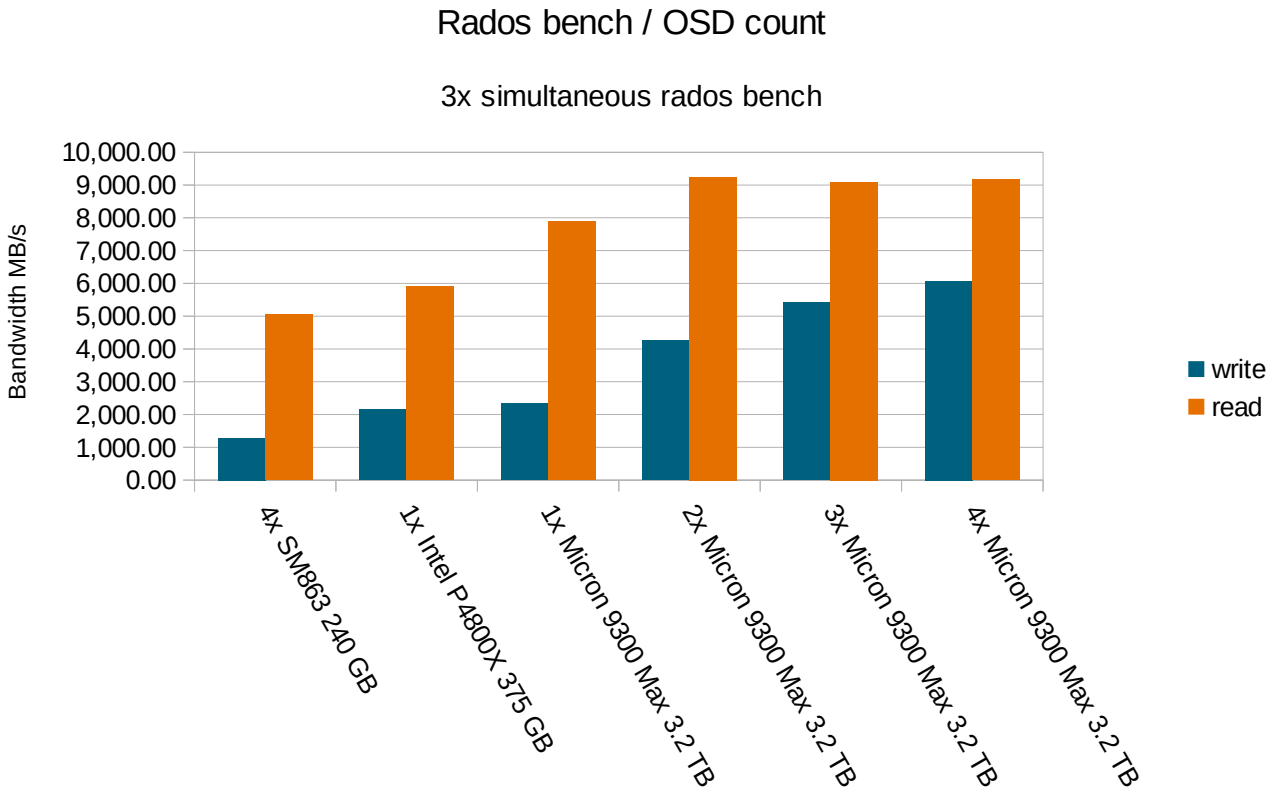
```
fio --ioengine=libaio --filename=/dev/sdx --direct=1 --sync=1 --rw=write --bs=4K
          --numjobs=1 --iodepth=1 --runtime=60 --time_based --name=fio
```

Note: This command will destroy any data on your disk.

STORAGE

For the following benchmarks, a single Ceph pool with 512 PGs, distributed on 12x Micron 9300 MAX was created. Ceph was used with default settings. This means the pool had a size of 3 (replica). And librbd uses 32 MiB as rbd cache size.

THE RADOS BENCH



SUMMARY

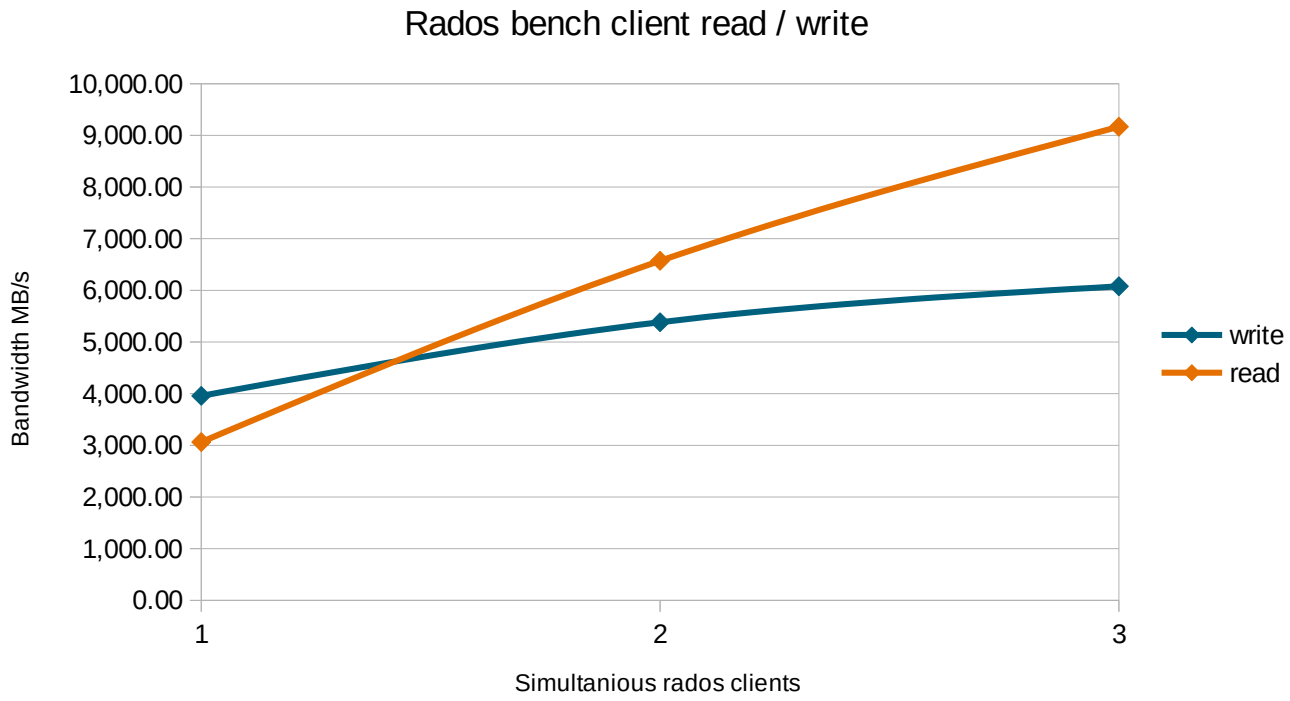
The Rados benchmark shows the read/write bandwidth of three rados bench clients, running simultaneously. The bandwidth topped out with two NVMe U.2s per node. The upper limit on these tests were 9,167 MB/s for reading and 6,078 MB/s for writing.

We used the following rados bench commands for these tests:

```
# write
rados bench 600 write -b 4M -t 16 --no-cleanup

# read (uses data from write)
rados bench 600 seq -t 16
```

A SINGLE CLIENT IS NOT ENOUGH



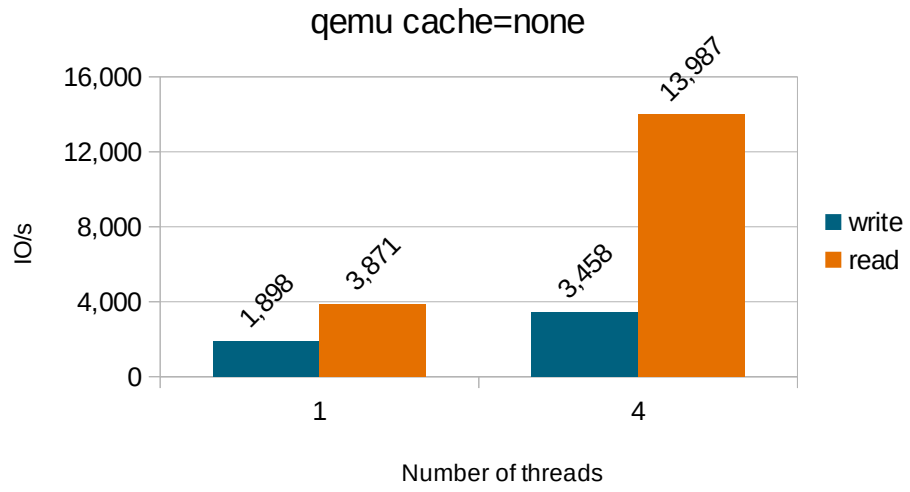
SUMMARY

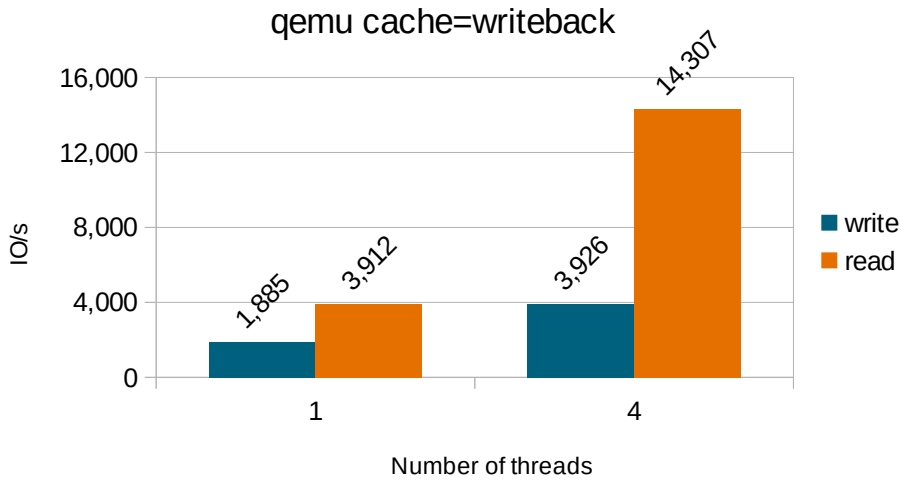
To utilize the bandwidth of 9,167 MB/s, three simultaneously running rados bench clients hat to be used, one on each node. One rados bench client on its own can not produce the amount of IO needed to fully load the cluster or use all of the resources on the running node.

The same rados bench commands as in the previous tests were used.

SINGLE VM PERFORMANCE (WINDOWS)

SEQUENTIAL IO/S BY NUMBER OF THREADS



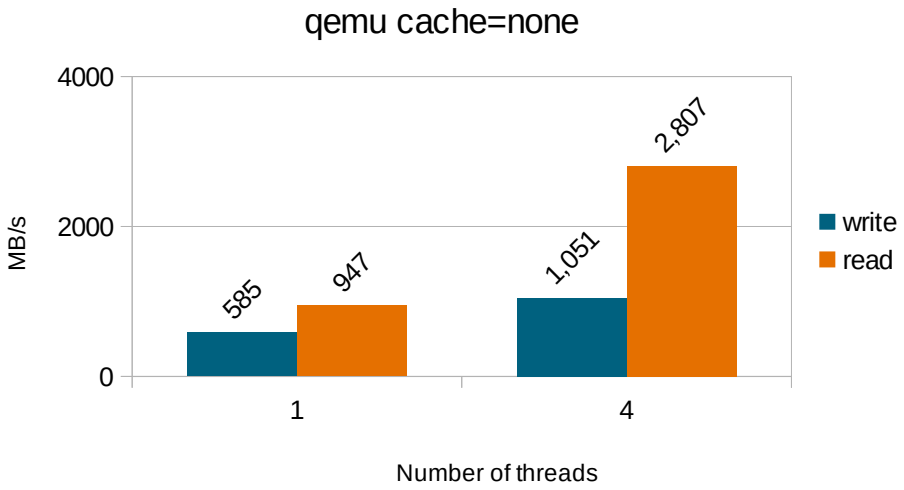


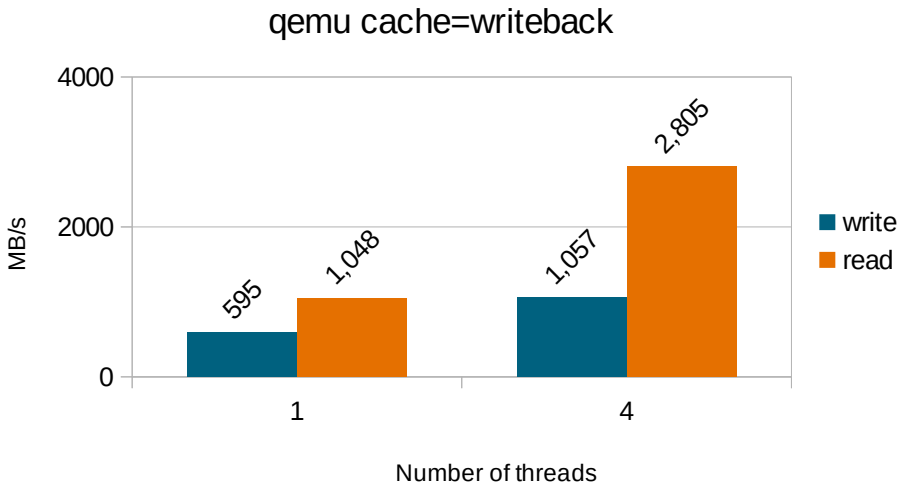
SUMMARY

The IO performance on a single Windows VM on the cluster. While the cache setting *'writeback'* only introduces a slight difference on a single thread, on four threads, it causes a 13.5% increase in write performance and a 2.3% increase in read performance.

The fio command used can be found in the Appendix, 5.1.

SEQUENTIAL BANDWIDTH BY NUMBER OF THREADS





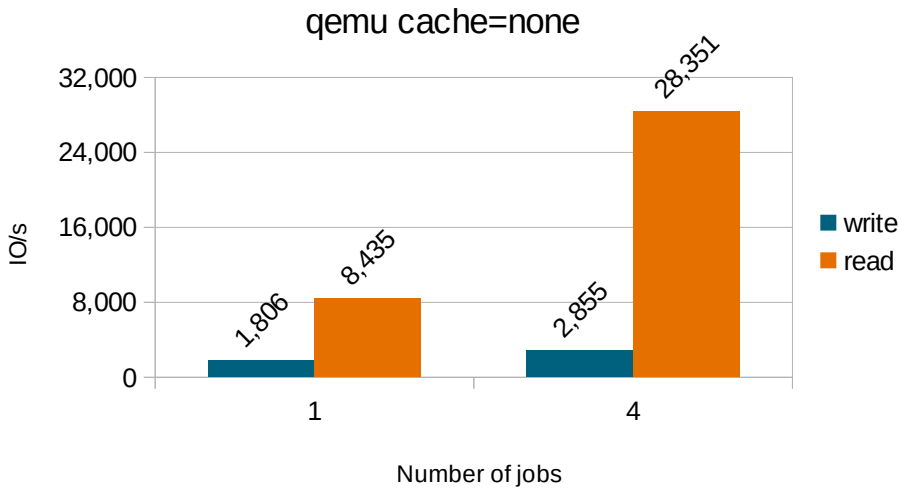
SUMMARY

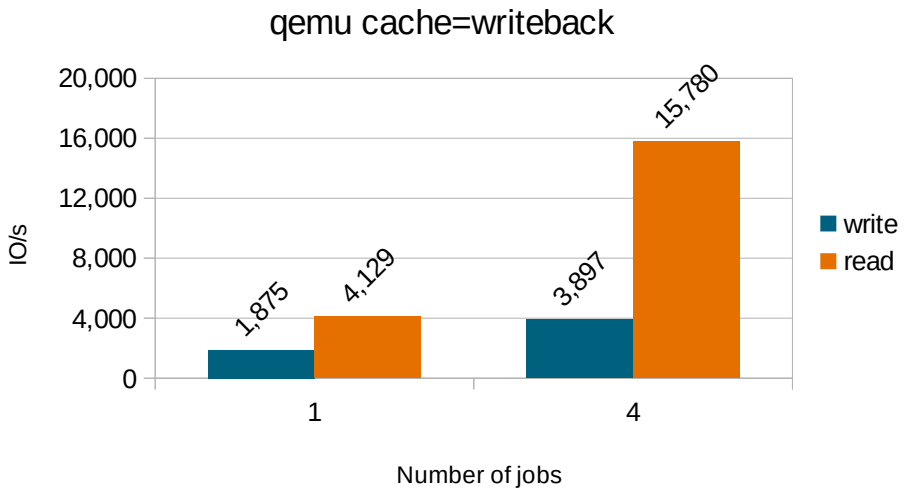
The bandwidth of a single Windows VM on the cluster. The block size used for the read and write benchmark is 4 MB. The cache setting *'writeback'* has no significant impact.

The fio command used can be found in the Appendix, 5.1.

SINGLE VM PERFORMANCE (LINUX)

SEQUENTIAL IO/S BY NUMBER OF JOBS



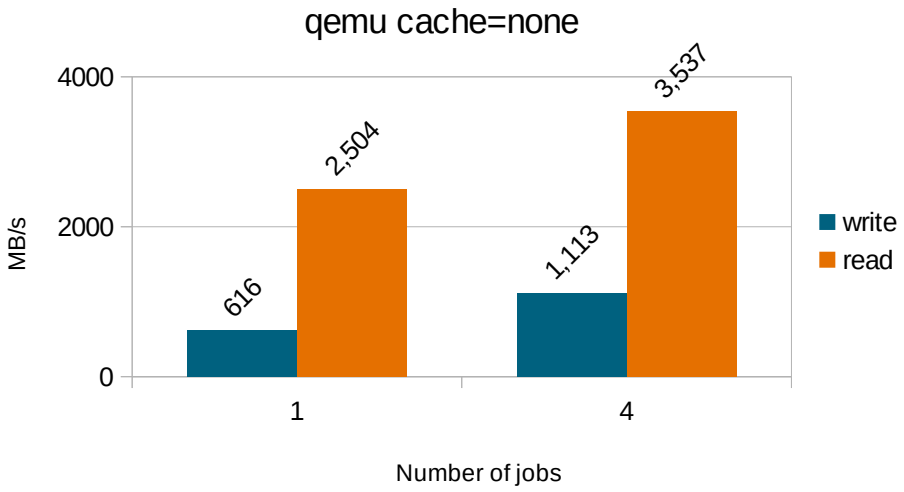


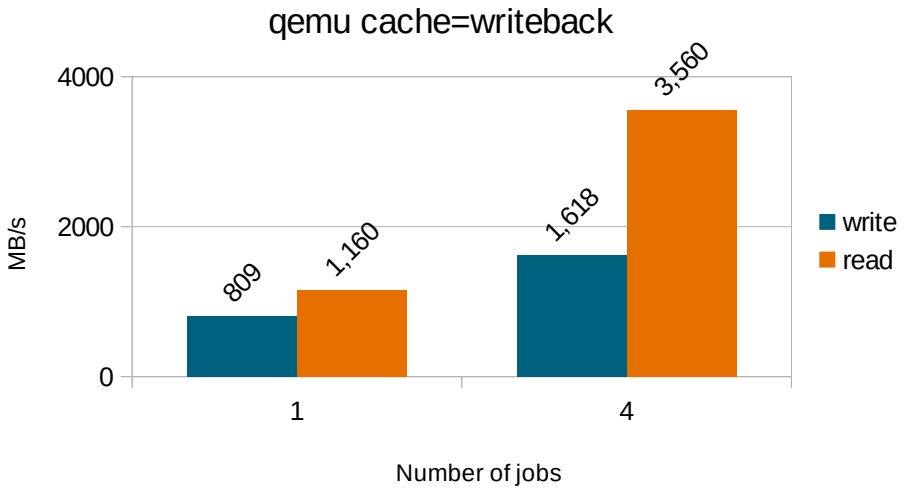
SUMMARY

The IO performance on a single Linux VM on the cluster. The cache setting '*writeback*' introduces a penalty on read. The visible change on the 4x jobs benchmark is an **increase of 36.5% in write** but a **44.4% decrease in read**.

The fio command used can be found in the Appendix, 5.2.

SEQUENTIAL BANDWIDTH BY NUMBER OF JOBS





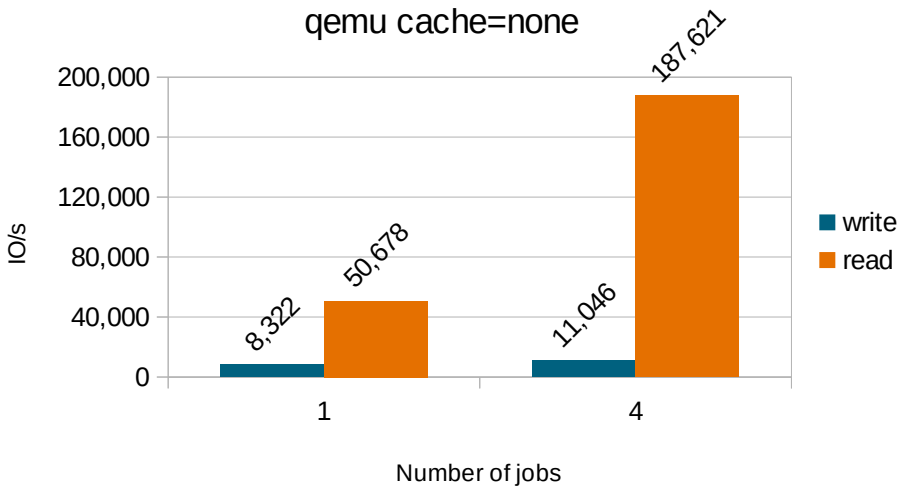
SUMMARY

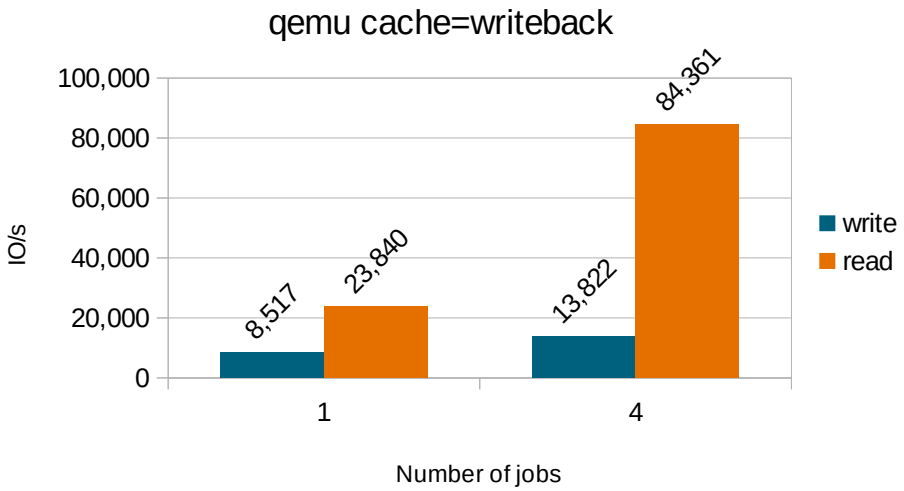
The bandwidth of a single Linux VM on the cluster. The block size used for the read and write benchmark is 4 MB. The cache setting *'writeback'* introduces a small advantage in write bandwidth.

The fio command used can be found in the Appendix, 5.2.

MULTI-VM WORKLOAD (LINUX)

SEQUENTIAL IO/S BY NUMBER OF JOBS



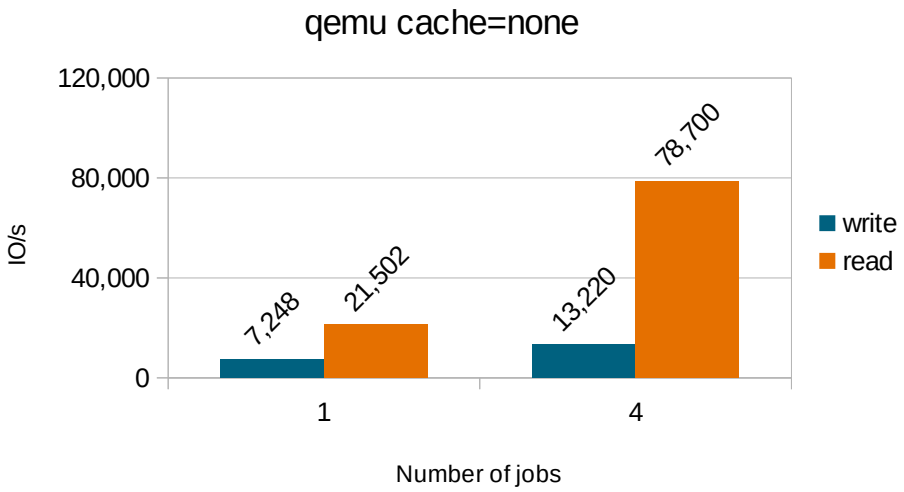


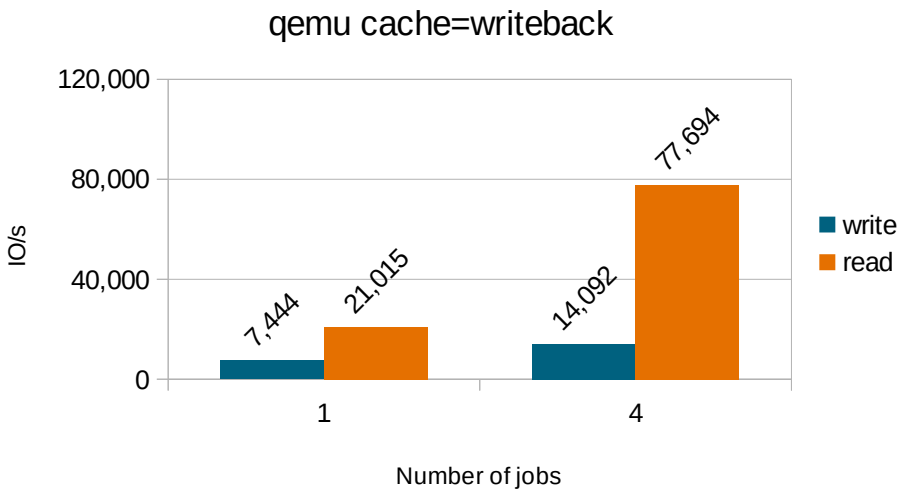
SUMMARY

The IO performance of 6x Linux VMs, evenly distributed on the cluster. As seen in the single VM charts, the cache setting *'writeback'* introduces a heavy penalty on read performance. The visible change on the 4x jobs benchmark is an increase of 25.1% in write, but a 55.1% decrease in read. This is very close in ratio to the single VM charts.

The fio command used can be found in the Appendix, 5.2.

RANDOM IO/S BY NUMBER OF JOBS



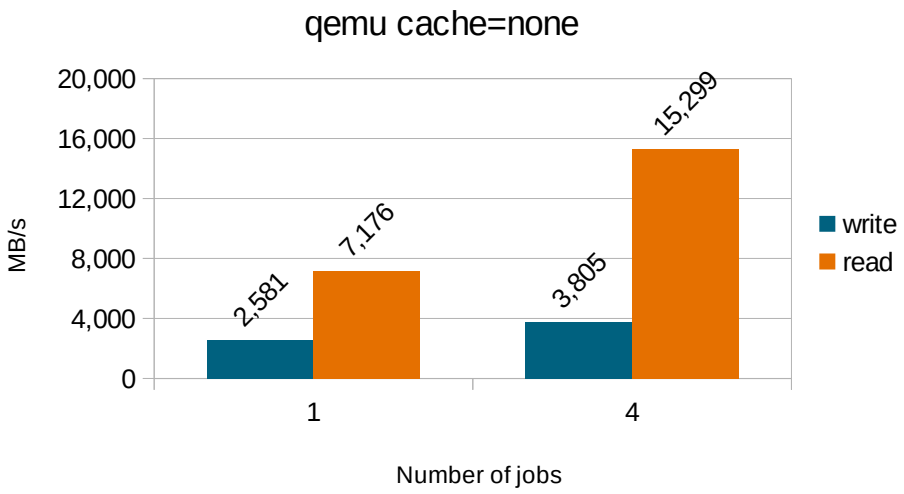


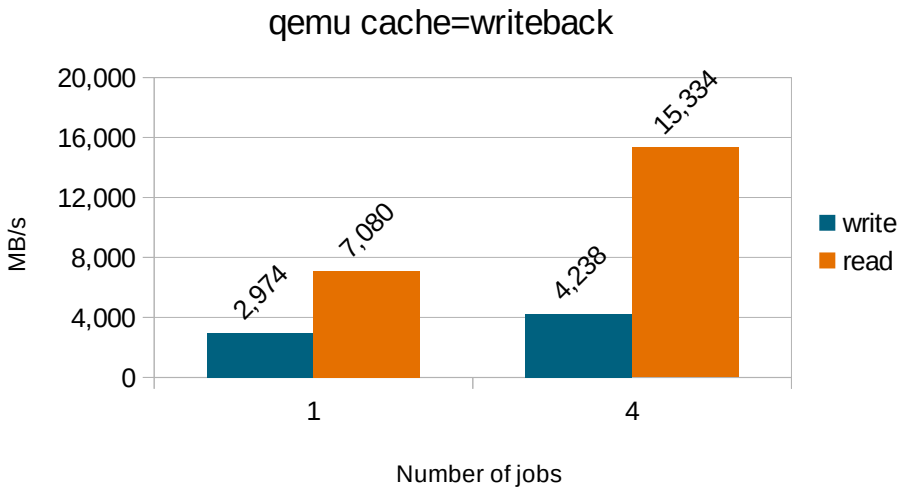
SUMMARY

The IO performance of 6x Linux VMs, evenly distributed on the cluster. Slight gain in write and small loss on read with cache setting *'writeback'* with four number of jobs.

The fio command used can be found in the Appendix, 5.3.

SEQUENTIAL BANDWIDTH BY NUMBER OF JOBS



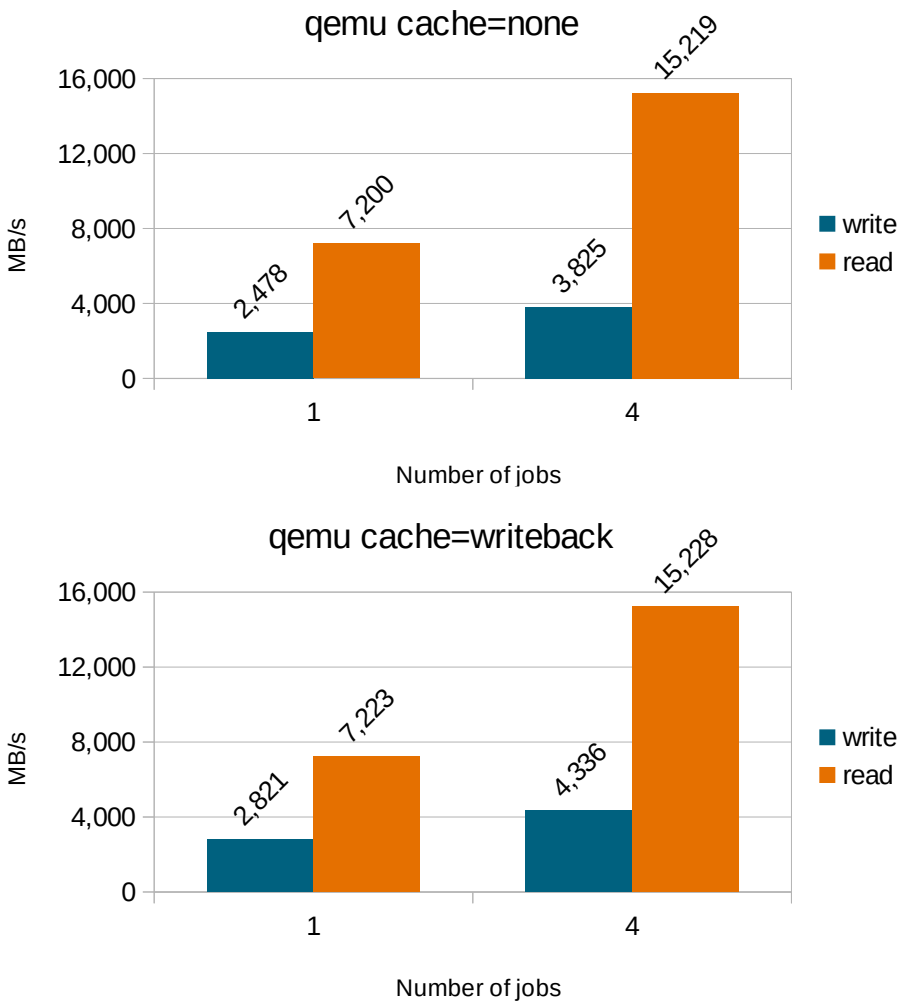


SUMMARY

The bandwidth of 6x Linux VMs, evenly distributed on the cluster. The block size used for the read and write benchmark is 4 MB. As in the single VM bandwidth chart, the cache setting '*writeback*' introduces a small benefit on write bandwidth.

The fio command used can be found in the Appendix, 5.2.

RANDOM BANDWIDTH BY NUMBER OF JOBS



SUMMARY

The bandwidth of 6x Linux VMs, evenly distributed on the cluster. The block size used for the read and write benchmark is 4 MB. As in the sequential VM bandwidth charts, the cache setting '*writeback*' introduces a small advantage on write bandwidth.

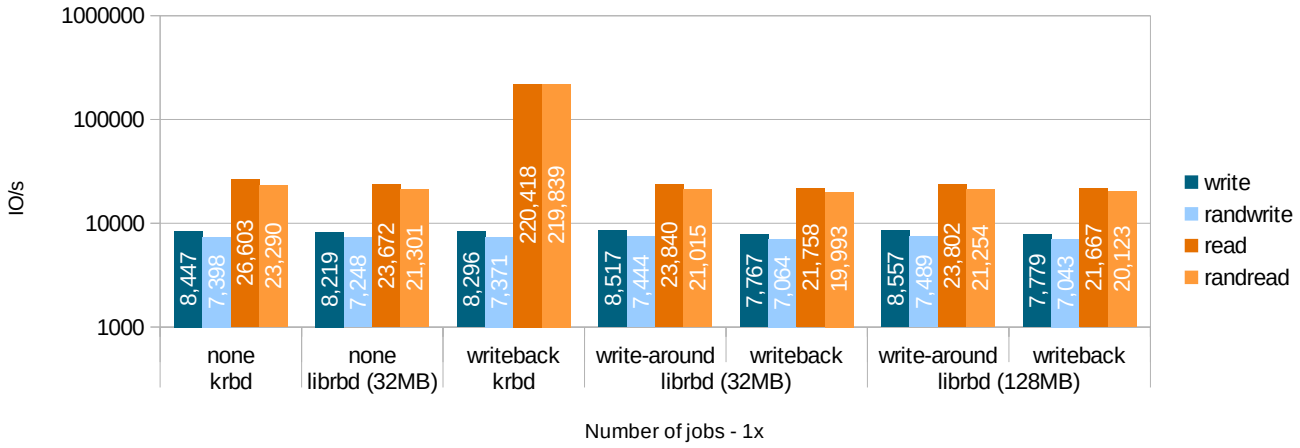
The fio command used can be found in the Appendix, 5.3.

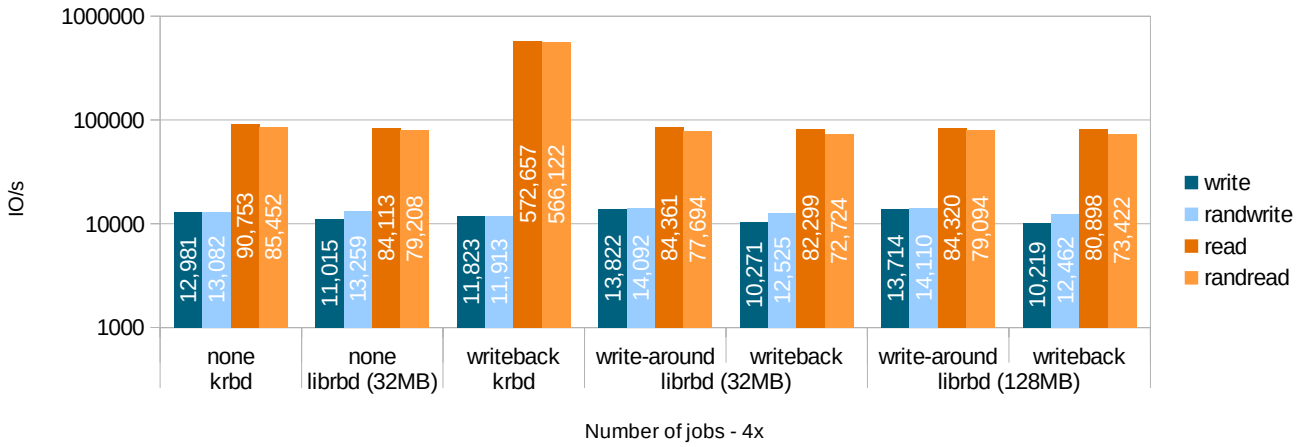
KRBD VS LIBRBD (LINUX)

krbd is the kernel driver for Ceph's Rados Block Devices (RBD). It can use the host's page cache to improve performance. The RBD image is mapped to a local block device.

librbd is the userspace library used by Qemu to connect RBD images. Since it is a userspace library, it needs its own cache. By default this is 32 MiB.

IO/S BY NUMBER OF JOBS



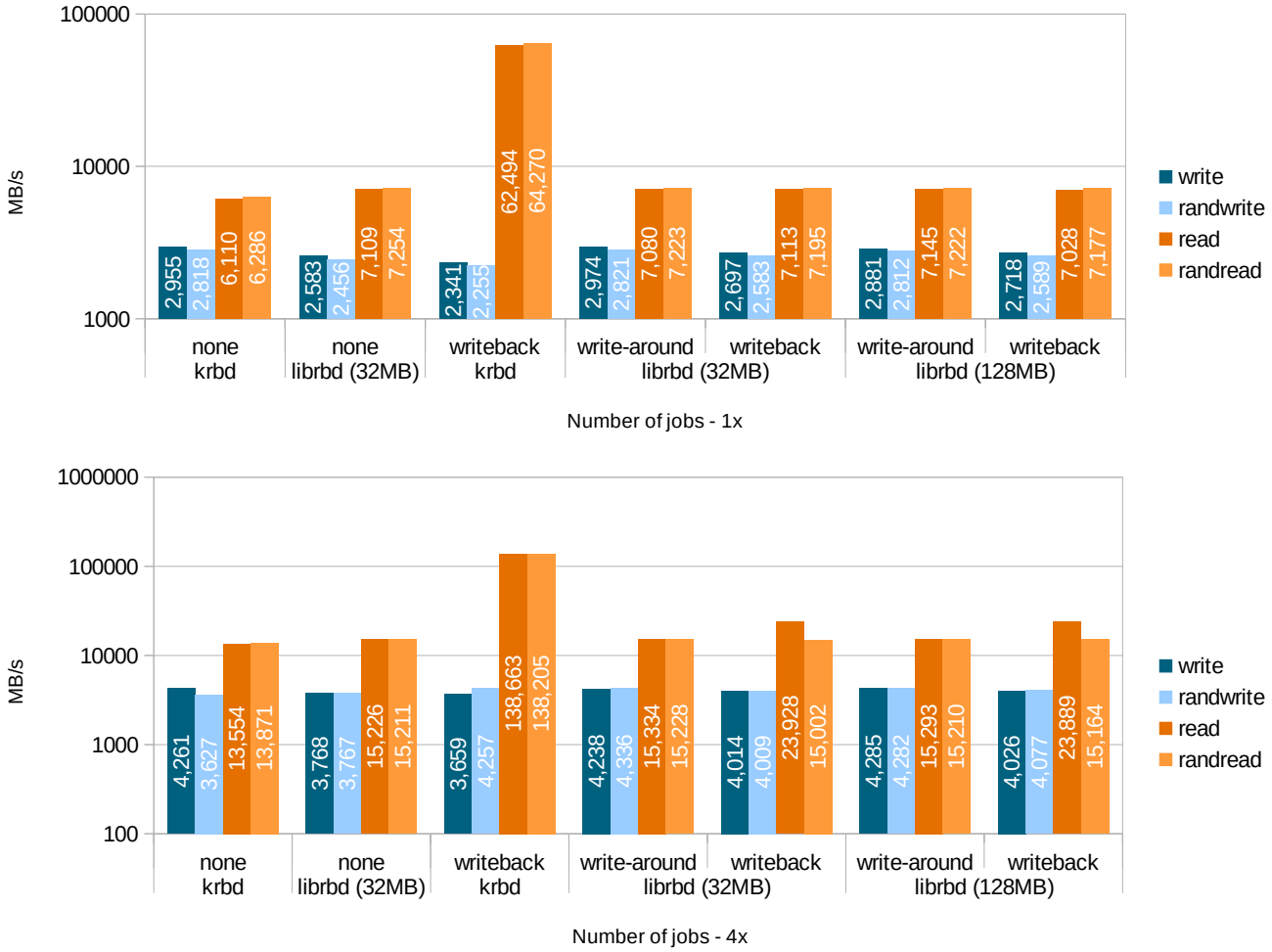


SUMMARY

The IO performance of 6x Linux VMs, evenly distributed on the cluster. Due to the ~100 GB of memory, the krbd client can achieve over 500k IO/s with writeback enabled. librbd shows a slight advantage on write performance compared to krbd.

The fio command used can be found in the Appendix, 5.3.

BANDWIDTH BY NUMBER OF JOBS



SUMMARY

The bandwidth of 6x Linux VMs, evenly distributed on the cluster. The block size used for the read and write benchmark is 4 MB. The bandwidth has the same spike as seen on the previous IO/s chart. It should be noted that the write bandwidth will always be lower since Ceph can read in parallel.

The fio command used can be found in the Appendix, 5.3.

HARDWARE FAQ

Can I use NVMe SSDs, for example M2 or PCI-Express cards?

Yes, but the U.2 NVMe SSDs provide better disk performance compared to M.2 variants. It also allows for a more space-saving build than with PCI-Express cards.

Can I create a fast pool with NVMe SSDs and a semi-fast pool with SSDs?

Yes, building several pools can help in situations where budget is limited, but a lot of storage is needed.

Which CPUs are better: More cores or a higher frequency?

CPUs with both a lot of cores and a high frequency are the best choice. This is true for Intel Xeon and AMD Epyc CPUs.

Should I use NUMA or UMA systems?

The use of *Non-Uniform Memory Access* (NUMA) or *Uniform Memory Access* (UMA) is a secondary factor. It can be compared to better speed through locality vs easier configuration and management.

How much RAM do I need per server?

Every OSD needs memory for caching. Also VM/CT workloads will need memory. So the amount depends on the number of OSDs and your VM workload. In general, the best recommendation is using as much RAM as possible.

Why did you use 3.20 TB U.2 NVMe SSDs for your tests?

To utilize the AMD Epyc Zen2 platform, it needs multiple modern U.2 disks.

Why do you test with 100 Gbit? Isn't 25 Gbit enough?

While both the 25 Gbit and the 100 Gbit technologies are a good choice (as they have the same low latency properties), running 1x OSD on each node will already demand more than 60 Gbit/s during read operations.

Can I use a mesh network, instead of an expensive 100 Gbit switch?

Yes, a mesh network is okay if you have dual NICs and just 3 nodes.

Can I use consumer or pro-sumer SSDs, as these are much cheaper than enterprise-class SSDs?

No. Never. These SSDs wont provide the required performance, reliability or endurance. See the fio results from before and/or run your own fio tests.

Can I mix various disk types?

It is possible, but the cluster performance will drop to the performance of the slowest disk.

Can I mix different disk sizes?

No, it's not recommended to use different disk sizes in small clusters, because this will provoke unbalanced data distribution.

APPENDIX

1 - RADOS BENCHMARK ON 100 GBIT NETWORK

	rados bench 600 write -b 4M -t 16 --no-cleanup		
	3x PVE (EPYC) 3x4 Micron 9300 Max		
	Single Client	Two Clients	Three Clients
Total time run	600.01	600.02	600.02
Total writes made	593,805.00	807,521.00	911,808.00
Write size	4,194,304.00	4,194,304.00	4,194,304.00
Object size	4,194,304.00	4,194,304.00	4,194,304.00
Bandwidth (MB/sec)	3,958.62	5,383.28	6,078.49
Stddev Bandwidth	73.33	58.78	50.54
Max bandwidth (MB/sec)	4,124.00	5,708.00	6,504.00
Min bandwidth (MB/sec)	3,516.00	4,912.00	5,504.00
Average IOPS	989.00	1,345.00	1,518.00
Stddev IOPS	18.33	14.69	12.63
Max IOPS	1,031.00	1,427.00	1,626.00
Min IOPS	879.00	614.00	1,376.00
Average Latency(s)	0.0162	0.0238	0.0316
Stddev Latency(s)	0.0051	0.0100	0.0147
Max latency(s)	0.1191	0.2347	0.2674
Min latency(s)	0.0062	0.0069	0.0074

	rados bench 600 seq -t 16 (uses 4M from write)		
	3x PVE (EPYC) 3x4 Micron 9300 Max		
	Single Client	Two Clients	Three Clients
Total time run	600.03	491.63	399.09
Total reads made	459,616.00	807,521.00	911,808.00
Read size	4,194,304.00	4,194,304.00	4,194,304.00
Object size	4,194,304.00	4,194,304.00	4,194,304.00
Bandwidth (MB/sec)	3,063.95	6,571.09	9,167.38
Average IOPS	765.00	1,641.00	2,290.00
Stddev IOPS	26.50	24.60	22.05
Max IOPS	844.00	1,752.00	2,483.00
Min IOPS	685.00	1,487.00	2,029.00
Average Latency(s)	0.0205	0.0191	0.0205
Max latency(s)	0.1518	0.1220	0.1146
Min latency(s)	0.0045	0.0047	0.0050

2 – PROXMOX VE SOFTWARE VERSIONS

proxmox-ve	6.2-1 (running kernel 5.4.55-1-pve)
pve-manager	6.2-11 (running version: 6.2-11/db10c37a)

pve-kernel-5.4	6.2-5	lxc-pve	4.0.3-1
pve-kernel-helper	6.2-5	lxcfs	4.0.3-pve3
pve-kernel-5.4.55-1-pve	5.4.55-1	novnc-pve	1.1.0-1
ceph	15.2.4-pve1	proxmox-backup-client	0.8.11-1
ceph-fuse	15.2.4-pve1	proxmox-mini-journalreader	1.1-1
corosync	3.0.4-pve1	proxmox-widget-toolkit	2.2-10
criu	3.11-3	pve-cluster	6.1-8
glusterfs-client	5.5-3	pve-container	3.1-12
ifupdown	0.8.35+pve1	pve-docs	6.2-5
ksm-control-daemon	1.3-1	pve-edk2-firmware	2.20200531-1
libjs-extjs	6.0.1-10	pve-firewall	4.1-2
libknet1	1.16-pve1	pve-firmware	3.1-2
libproxmox-acme-perl	1.0.4	pve-ha-manager	3.0-9
libpve-access-control	6.1-2	pve-i18n	2.1-3
libpve-apiclient-perl	3.0-3	pve-qemu-kvm	5.1.0-1
libpve-common-perl	6.2-1	pve-xtermjs	4.7.0-1
libpve-guest-common-perl	3.1-2	qemu-server	6.2-13
libpve-http-server-perl	3.0-6	smartmontools	7.1-pve2
libpve-storage-perl	6.2-6	spiceterm	3.1-1
libqb0	1.0.5-1	vncterm	1.6-2
libspice-server1	0.14.2-4~pve6+1	zfsutils-linux	0.8.4-pve1
lvm2	2.03.02-pve4		

3 – BIOS SETTINGS GIGABYTE MZ32-AR0-00)

Advanced:			
PCI Subsystem Settings			
PCIE_7 Lanes	[x4 x4 x4 x4]	##	bifurcation for U.2 SSD
AMD CBS:			
CPU Common Options			
Performance			
Custom Pstate2			
Custom Core Pstates	[Auto]	##	fix P-state, leave sub-settings default
Custom Pstate0	[Disabled]		
Custom Pstate1	[Disabled]		
Global C-state Control	[Disabled]		
Local APIC Mode	[x2APIC]		
DF Common Options			
Memory Addressing			
NUMA nodes per socket	[NPS0]	##	check as last option, will be set automatically by other options
UMC Common Options			
DDR4 Common Options			
Enforce POR			
Overclock	[Enabled]	##	fixed memory speed
Memory Clock Speed	[1600MHz]		(uncoupled from Infinity Fabric)
Security			
TSME	[Disabled]	##	Transparent Secure Memory Encryption root (5-7 ns)
NBIO Common Options			
SMU Common Options			
Power Policy Quick Setting	[Best Performance]		
Determinism Control	[Manual]		
Determinism Slider	[Performance]	##	identical performance for all cores
APBDIS	[1]	##	fixed Infinity Fabric P-state control
DF Cstates	[Disabled]	##	Infinity Fabric power states
Fixed SOC Pstate	[P0]		
CPPC	[Disabled]	##	Allows OS to make performance/power optimization requests using ACPI CPPC

4 – PROXMOX VE HOST SETTINGS

```
# /etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT="quiet pcie_aspm=off amd_iommu=on iommu=pt
mitigations=off"


# 100 GbE ConnectX-4
# /etc/network/interfaces
MTU 9000
```

5 – FIO TESTS

5.1 - fio command for “VM Performance (Windows)”

```
fio --ioengine=windowsaio --filename=test_fio --size=9G --time_based --name=fio
--group_reporting --runtime=600 --direct=1 --sync=1 --rw=<write|read> --threads
--bs=<4K|4M> --numjobs=<1|4> --iodepth=<1|32>
```

5.2 - fio command for “VM Performance (Linux)”

```
fio --ioengine=psync --filename=/dev/mapper/test_fio --size=9G --time_based
--name=fio --group_reporting --runtime=600 --direct=1 --sync=1 --rw=<write|read>
--bs=<4K|4M> --numjobs=<1|4> --iodepth=<1|32>
```

5.3 - fio command for “Multi VM workload (Linux)”

```
fio --ioengine=psync --filename=/dev/mapper/test_fio --size=9G --time_based
--name=fio --group_reporting --runtime=600 --direct=1 --sync=1
--rw=<randwrite|randread> --bs=<4K|4M> --numjobs=<1|4> --iodepth=<1|32>
```

6 – VM CONFIGURATION

qemu cache=writeback

qm config 101 (Proxmox VE 6.2)

agent:	1
bios:	ovmf
bootdisk:	scsi0
cores:	4
efidisk0:	rbd:vm-101-disk-1,size=4M
machine:	q35
memory:	16384
name:	PVE6
net0:	virtio=CE:F4:46:8C:FF:95,bridge=vmbr0,firewall=1
numa:	0
ostype:	l26
scsi0:	rbd:vm-101-disk-0,cache=writeback,discard=on,iothread=1,size=132G,ssd=1
scsihw:	virtio-scsi-single
smbios1:	uuid=65f3eff6-cbeb-4f12-871d-4e14cd42e1db
sockets:	1
vga:	virtio
vmgenid:	72fdba72-015f-489d-bb4e-6921111982ab
Edition	Proxmox VE 6.2
OS build	proxmox-ve:6.2-1 (running kernel: 5.4.60-1-pve)
Version	pve-manager:6.2-11 (running version: 6.2-11/22fb4983)

qm config 100 (Windows 2k19)

agent:	1
bios:	ovmf
bootdisk:	scsi0
cores:	4
efidisk0:	rbd:vm-100-disk-1,size=4M
machine:	q35
memory:	16384
name:	Win2019
net0:	virtio=56:43:AC:12:39:4E,bridge=vmbr0,firewall=1
numa:	0
ostype:	win10
scsi0:	rbd:vm-100-disk-0,cache=writeback,discard=on,iothread=1,size=132G,ssd=1
scsihw:	virtio-scsi-single
smbios1:	uuid=e83e6d64-bd8a-4814-b221-72dea14e2199
sockets:	1
vga:	virtio
vmgenid:	ed59f395-e7c1-4564-b577-567a51d81258
Edition	Windows Server 2019 Standard Evaluation
OS build	17763.1457
Version	1809 (latest updates)

LEARN MORE

Wiki: https://pve.proxmox.com

Community Forums: https://forum.proxmox.com

Bugtracker: https://bugzilla.proxmox.com

Code repository: https://git.proxmox.com

HOW TO BUY

Find an authorised reseller in your area: www.proxmox.com/partners or

Visit the Proxmox Online Shop to purchase a subscription: https://shop.maurer-it.com

SALES AND INQUIRIES

https://www.proxmox.com

Proxmox Customer Portal

https://my.proxmox.com

TRAINING PROXMOX VE

Learn Proxmox VE easily, visit https://www.proxmox.com/training

ABOUT PROXMOX

Proxmox Server Solutions GmbH is a privately held company based in Vienna, Europe.