

# Performance Tests

## IO Performance Tests

Für IO Performance kann `fiio` genutzt werden.

`fiio` ist für die meisten Betriebssysteme verfügbar und kann daher genutzt werden um vergleichbare Ergebnisse zu bekommen.

```
#!/bin/bash

testfile="FIO-TESTFILE"
filesize=1G

echo "IOPS Write:"
fiio --rw=randwrite --name=IOPS-write --bs=4k --iodepth=32\
  --direct=1 --filename=$testfile --numjobs=4 --ioengine=libaio\
  --refill_buffers --group_reporting --runtime=60 --time_based\
  --size=$filesize --output-format=json | jq .jobs[0].write.iops

echo "IOPS Read:"
fiio --rw=randread --name=IOPS-read --bs=4k --iodepth=32\
  --direct=1 --filename=$testfile --numjobs=4 --ioengine=libaio\
  --refill_buffers --group_reporting --runtime=60 --time_based\
  --size=$filesize --output-format=json | jq .jobs[0].read.iops

echo "Throughput Write (kB/s):"
fiio --rw=write --name=Throughput-write --bs=1024k --iodepth=32\
  --direct=1 --filename=$testfile --numjobs=4 --ioengine=libaio\
  --refill_buffers --group_reporting --runtime=60 --time_based\
  --size=$filesize --output-format=json | jq .jobs[0].write.bw

echo "Throughput Read (kB/s):"
fiio --rw=read --name=Throughput-read --bs=1024k --iodepth=32\
  --direct=1 --filename=$testfile --numjobs=4 --ioengine=libaio\
  --refill_buffers --group_reporting --runtime=60 --time_based\
  --size=$filesize --output-format=json | jq .jobs[0].read.bw

echo "Latency Write (ns):"
fiio --rw=randwrite --name=Latency-write --bs=4k --iodepth=1\
  --direct=1 --filename=$testfile --numjobs=1 --ioengine=libaio\
  --refill_buffers --group_reporting --runtime=60 --time_based\
  --size=$filesize --output-format=json | jq .jobs[0].write.lat_ns.mean

echo "Latency Read (ns):"
fiio --rw=randread --name=Latency-read --bs=4k --iodepth=1\
  --direct=1 --filename=$testfile --numjobs=1 --ioengine=libaio\
  --refill_buffers --group_reporting --runtime=60 --time_based\
```

```
--size=$filesize --output-format=json | jq .jobs[0].read.lat_ns.mean
```

## Testergebnisse von verschiedenen Systemen

	IOPS Write	IOPS Read	Throughput Write	Throughput Read	Latency Write	Latency Read
<b>soquartz eMMC</b>	3264	3295	40.2 MB/s	44.9 MB/s	647us	587us
<b>soquartz NVME</b>	38.2K	54.6K	389.2 MB/s	416.9 MB/s	70us	210us
<b>Olimex Lime2 SATA-SSD</b>	4701	20.6K	132.3 MB/s	280.9 MB/s	222us	248us
<b>PVE Guest (HDD, ZFS Raid-Z)</b>	910	690K	122.7 MB/s	16314.6 MB/s	65us	56us
<b>PVE Guest (NVME, ZFS Raid1)</b>	225K	287K	1469.8 MB/s	11681.5 MB/s	37us	86us
<b>PVE (NVME, ZFS Raid1)</b>	360K	917K	1474.1 MB/s	12081.4 MB/s	13us	63us
<b>HyperV(S2D) Guest (woe)</b>	27.7K	120K	2820.2 MB/s	11549.6 MB/s	530us	158us
<b>HyperV(S2D) Guest (fus) IOPS-Limit 15K</b>	6730	17.8K	120.4 MB/s	120.5 MB/s	854us	256us
<b>HyperV(S2D) Guest (fus) IOPS-Limit 30K</b>	5606	37.4K	116.6 MB/s	240.0 MB/s	121us	360us
<b>Moritz fra-hv01 local Disk</b>	852	1659	71.0 MB/s	138.0 MB/s	190us	402us
<b>Moritz fra-hv02 local Disk</b>	3421	1967	53.8 MB/s	88.4 MB/s	189us	502us
<b>Moritz fra-hv03 local Disk</b>	8903	3605	145.7 MB/s	215.5 MB/s	108us	299us
<b>Moritz fra-hv04 local Disk</b>	7412	5040	184.8 MB/s	208.3 MB/s	144us	204us
<b>Moritz fra-hv06 local Disk</b>	46.8K	51.4K	258.6 MB/s	276.8 MB/s	49us	111us
<b>Moritz debian WSL local</b>	45.3K	90.8K	1574.2 MB/s	1528.0 MB/s	502us	216us
<b>Moritz tgf-nextcloud ZFS Raidz2, kein Cache</b>	43.1K	133.7K	247.2 MB/s	6030.1 MB/s	123us	103us
<b>Moritz Plesk Frankfurt, local Disk on CEPH in fra-hvclu01</b>	10.6K	5680	244.2 MB/s	1311.6 MB/s	147us	2443us
<b>old Plesk Nurnberg, local Disk on CEPH NVMECluster (3 Nodes)</b>	8751	77.2K	1874.1 MB/s	3460.7 MB/s	125us	198us
<b>bookstack LXC Frankfurt, on CEPH in fra-hvclu01</b>	3610	66.0K	354.9 MB/s	1110.3 MB/s	11.4ms	496us

From:

<https://dokuwiki.krumel.moe/> - **Krumels Wiki**

Permanent link:

<https://dokuwiki.krumel.moe/doku.php?id=kb:perftests&rev=1694773516>

Last update: **2023/09/15 10:25**

