Performance Tests

IO Performance Tests

Für IO Performance kann fio genutzt werden.

fio 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:"
fio --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:"
fio --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):"
fio --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):"
fio --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):"
fio --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):"
fio --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
soquartz eMMC	3264	3295	40.2 MB/s	44.9 MB/s	647us	587us
soquartz NVME	38.2K	54.6K	389.2 MB/s	416.9 MB/s	70us	210us
PVE Guest (HDD, ZFS Raid-Z)	910	690K	122.7 MB/s	16314.6 MB/s	65us	56us
PVE Guest (NVME, ZFS Raid1)	225K	287K	1469.8 MB/s	11681.5 MB/s	37us	86us
PVE (NVME, ZFS Raid1)	360K	917K	1474.1 MB/s	12081.4 MB/s	13us	63us
HyperV(S2D) Guest (woe)	27.7K	120K	2820.2 MB/s	11549.6 MB/s	530us	158us
HyperV(S2D) Guest (fus) IOPS-Limit 15K	6730	17.8K	120.4 MB/s	120.5 MB/s	854us	256us
HyperV(S2D) Guest (fus) IOPS-Limit 30K	5606	37.4K	116.6 MB/s	240.0 MB/s	121us	360us
Moritz fra-hv01 local Disk	852.493063	1659.652394	71049	137967	189575.726747	402426.9521
Moritz fra-hv02 local Disk	3421.037766	1967.701077	53799	88412	189259.706921	502043.359959
Moritz fra-hv03 local Disk	8903.40161	3605.51315	145693	215528	108842.985003	299693.106008
Moritz fra-hv04 local Disk	7412.126465	5040.115998	184776	208304	144745.12456	204814.175904
Moritz fra-hv06 local Disk	46782.190594	51395.830208	258558	276785	49581.012909	110651.745375
Moritz debian WSL local	45347.276848	90847.021766	1574197	1528016	502784.817276	215739.029873
Moritz tgf-nextcloud ZFS Raidz2, kein Cache	43088.031866	133705.154914	247242	6030152	123836.536794	102937.000653
Moritz Plesk Frankfurt, local Disk on CEPH in fra-hvclu01	10597.702485	5680.126892	244198	1311596	147824.787995	2443038.964188

	IOPS Write	IOPS Read	Throughput Write	Throughput Read	Latency Write	Latency Read
old Plesk Nurnberg, local Disk on CEPH NVMECluster (3 Nodes)	8751.062447	77185.631437	1874086	3460683	125601.021608	197665.175495
bookstack LXC Frankfurt, on CEPH in fra-hvclu01	3610.192058	66036.780881	354898	1110344	11432765.706634	495618.737156

From:
https://dokuwiki.krumel.moe/ - **Krumels Wiki**

Permanent link:
https://dokuwiki.krumel.moe/doku.php?id=kb:perftests&rev=1692884527

Last update: **2023/08/24 13:42**