```
hdparm -W
zfs set atime=off local-ssd-zfs
zfs set compression=lz4 local-ssd-zfs

zpool iostat  (affiche les stats cumulées...)
zpool iostat 2 (pour rafraichir toutes les 2 secondes)

#ajouter un disque log et cache
zpool add -f test-zfs log /dev/sdxx cache /dev/sdxxx
```

Trouvé sur le net

Symptom: Copying anything—even an ISO—caused I/O delay to spike to 40–90%, the VM froze, and the whole node choked. Even with only one VM and plenty of CPU/RAM. I know, I know…

“You’re not supposed to use consumer SSDs in production.” Totally agree. But sometimes a client chooses the budget they choose and the job is to make it work as safely as possible. Anyway… ✔ The Root Cause ZFS synchronous writes + consumer SSDs = absolute misery. Consumer SATA SSDs have:

```
  slow fsync latency
  tiny SLC caches
  no power-loss protection
  awful random write performance once the cache fills
  controllers that can stall under ZFS write patterns
```

Even with a 2-disk mirror, copying a file would hit the end of the SLC cache → SSD latency would jump → ZFS TXG flushes stalled → Proxmox I/O delay went crazy. I also spun up another test box at home using Intel DC enterprise SSDs and none of these issues showed up — so the hardware difference was the smoking gun. ✔ The Fix These ZFS dataset settings instantly stabilized the system: zfs set sync=disabled rpool/data zfs set atime=off rpool/data zfs set recordsize=64K rpool/data What each does (short version):

```
  sync=disabled → stops ZFS from forcing every tiny write to hit the SSD
immediately.
  (Yes, slight risk during an unexpected power loss. We have a UPS and BDR.)
  atime=off → stops ZFS from doing metadata writes for every read.
  recordsize=64K → better block size for VM workloads.
```

After these changes:

```
  ISO copies completed instantly
  I/O delay dropped from 90% → 1–5%
  Windows VM became responsive
  No more host freezing
```

Night and day.