

Benchmark des stockages de Proxmox

Introduction

L'idée de cette page est de déterminer quel est la meilleure solution pour le stockage des Machines Virtuelles, afin de sortir quel système de stockage local, et quelles options sont les plus judiciables d'un point de vue performance mais aussi en terme de fonctionnalité. J'essaie d'être le plus exhaustif en terme de configuration, même si certaines options ont pu être ignorées/oubliées.

Afin de réaliser ce benchmark, je vais utiliser la machine suivante :

Type	Equipement
CPU	Intel Core i5 7600K
Carte mère	Asus Z170 PRO GAMING
Mémoire	2*8GB Corsair 2133MHz DDR4
Disque Dur	KINGSTON SSD A400 480GB Samsung SSD 970 EVO Plus NVMe 250GB

Proxmox¹⁾, ainsi que la racine de la VM sont stocké sur le SSD Kingston, et tous les benchmarks seront faits sur le SSD NVMe Samsung qui est dédié uniquement à ceux-ci.

Ce choix permet de limiter l'impact de limitation de bande passante d'IO, ni par d'autres usages.

La VM, elle est configuré comme ceci :

Type	Equipement
Mémoire	8GB
CPU	1 socket, 2 cores (host)
Bios	Seabios
Disque principal	20GB sur du LVM-Thin avec discard en virtio
OS	Debian 10 Buster sans UI ²⁾

Afin d'effectuer ces tests je vais lancer sur le disque (sans système de fichier) avec les commandes suivantes :

- Pour le test d'écriture :

```
# dd if=/dev/zero of=/dev/${DISQUE} count=10 bs=1000M conv=fdatasync
```

- Pour le test de lecture :

```
# sync; echo 3 > /proc/sys/vm/drop_caches; dd if=/dev/${DISQUE} of=/dev/null count=10 bs=1000M
```

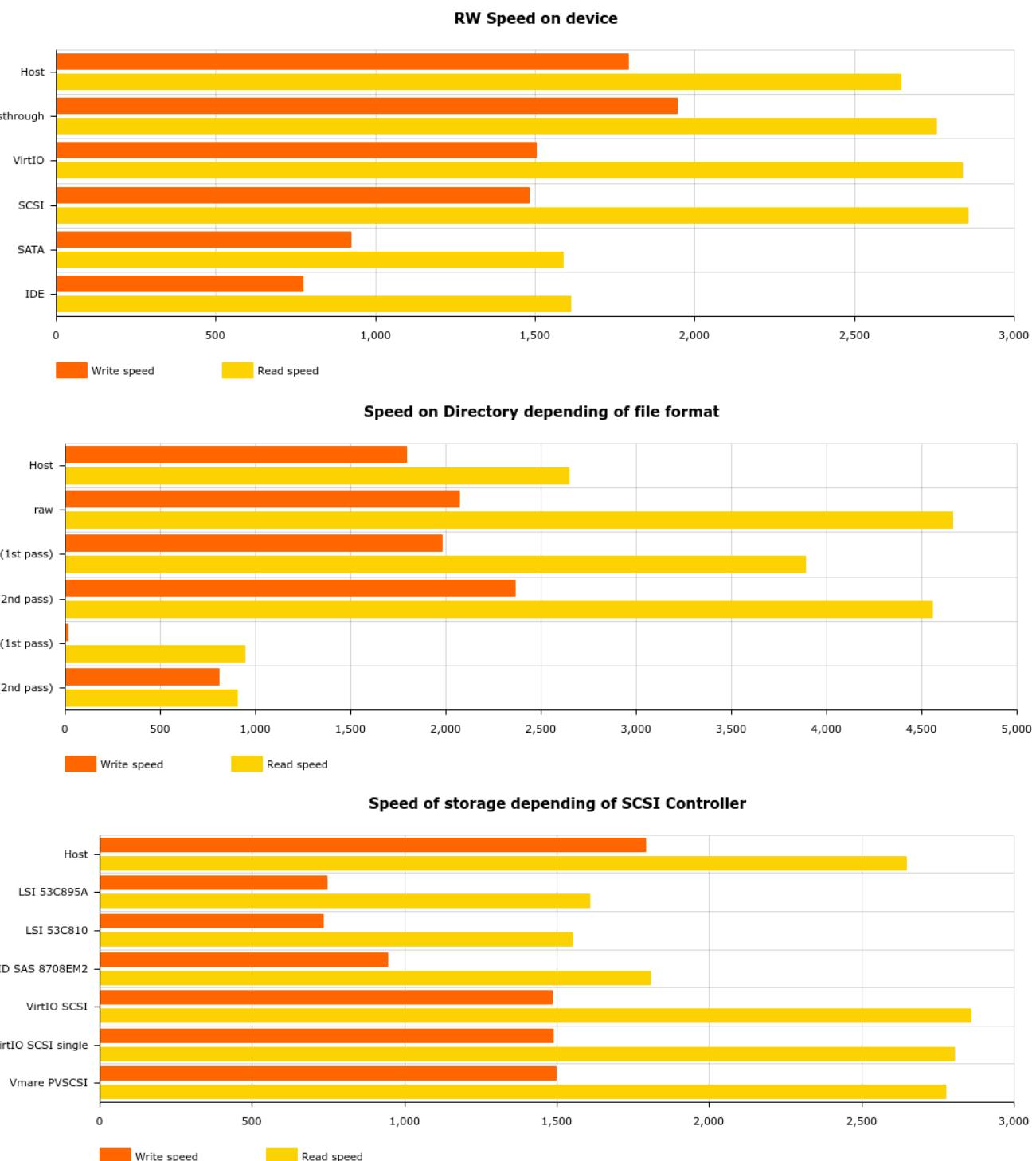
- Pour le test de latence :

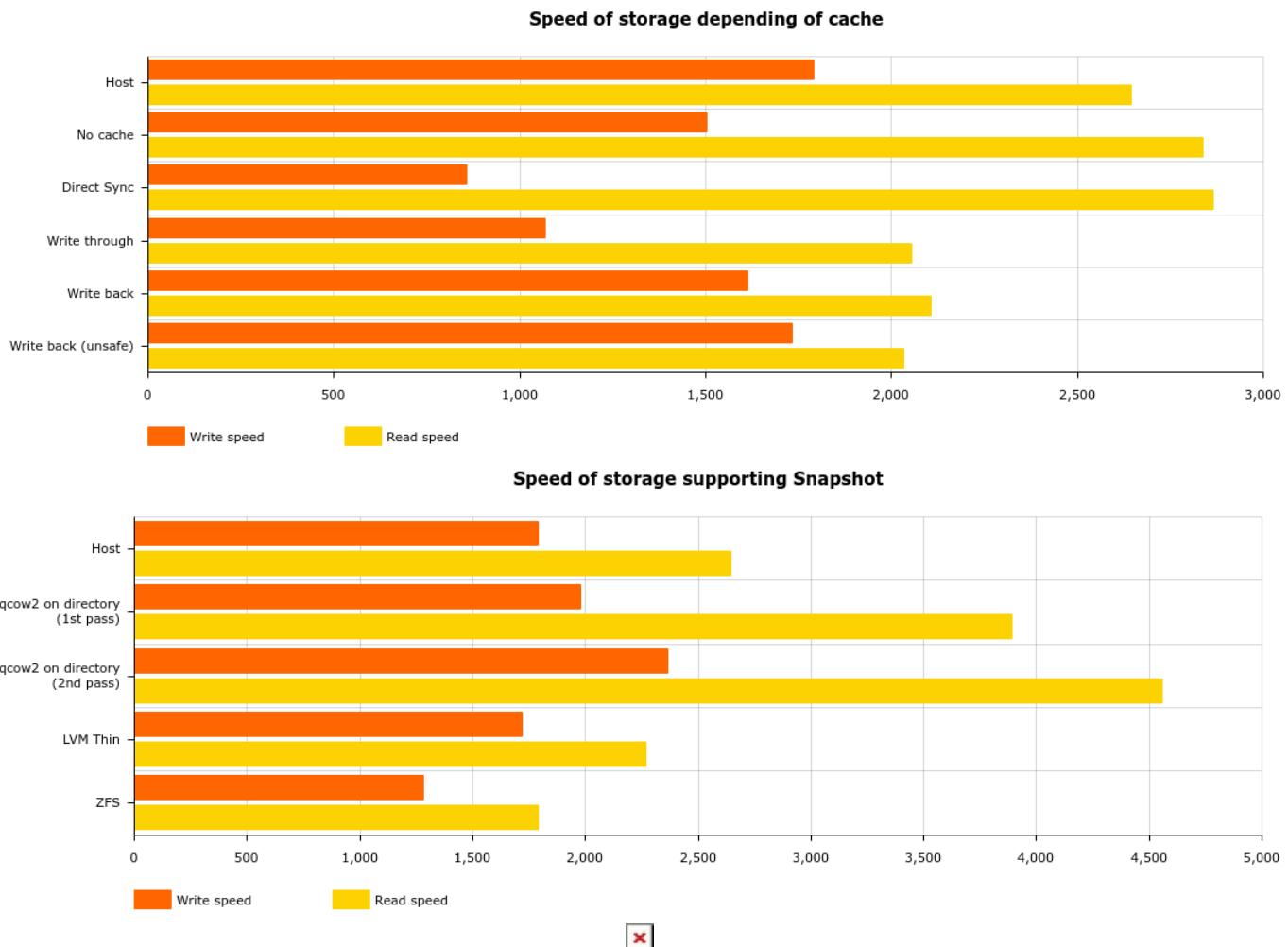
```
# ioping -B -c 10 /dev/${DISQUE} | awk '{print $6}'
```

Les résultats

J'ajoute quelques précisions pour mieux comprendre les résultats. Les valeurs "Host" présentes sur la totalité des graphs sont les valeurs de référence, en l'occurrence ce que le Proxmox lui-même obtient comme résultat (pas dans un VM donc).

Ah, et les valeurs sont en MB/s





Voici quelques remarques intéressantes :

- Les options **Discard**, **IO Thread** et **SSD Emulation** n'ont pas d'impacts sur les performances.³⁾
- Le **VMDK** est extrêmement lent lors de l'écriture de nouveaux blocks à cause du Thin Provisionning, d'où la présence de deux passes
- Il n'y a pas de différence flagrante sur le système de fichiers entre **ext4** et **xfs**
- La librairie compression sur le **ZFS** n'a peu d'impact sur les performances. Mais si vous désactivez la compression, les performances sont plus faibles⁴⁾
- Le support de stockage **LVM** n'est pas affiché dans les précédents graphes car ne rentrant dans aucune catégorie. Pour faire simple, il n'a aucun avantage par rapport au Directory + RAW ou LVM-Thin. (1474MB/s Read, 2679MB/s Write)

Pour approfondir le sujet, voici le document contenant tous les résultats aux tests effectués.



Conclusion

Quel système de stockage je recommande ?

Même si il y a beaucoup de options possibles, je pense que d'utiliser la combinaison :

- Contrôleur SCSI VirtIO
- disques VirtIO⁵⁾

- Pas de cache
- Discard si compatible avec l'OS guest (a partir de Linux 5.0)
- IO Thread si plusieurs disques lourdement utilisé

Après pour le support, j'en recommande 2, dépendant des besoins/possibilités:

- **Directory** avec images en **RAW⁶⁾** car étonnamment le plus performant, et permet simplement de partager ce volume avec les sauvegardes, les ISO, et autres....
- **LVM-Thin** pour simplement le thin-provisionning, et le support des Snapshots.

¹⁾ , ²⁾

a jour au 2020-04-01

³⁾

IO Thread peut être utile dans le cas où vous avez plusieurs disques

⁴⁾

Lorsque la compression était désactivée, le proxmox avait tendance à redémarrer tout seul pendant les tests.

⁵⁾

SCSI si pas compatible

⁶⁾

qcow2 n'a plus vraiment d'intérêt, si ce n'est les snapshots, car le thin-provisionning n'est plus supporté nativement par Proxmox

From:

<https://wiki.virtit.fr/> - VirtIT

Permanent link:

https://wiki.virtit.fr/doku.php/kb:linux:proxmox:benchmark_des_systemes_de_stockage?rev=1585906211

Last update: **2020/04/03 09:30**

