AYI NEDJIMI Consultants

Optimisation Proxmox VE 9 : CPU, RAM, ZFS, Ceph et HA

Catégorie : Virtualisation Lecture : 7 min Publié le : 22/03/2026 Auteur : Ayi NEDJIMI

Guide expert optimisation Proxmox VE 9 : CPU pinning, hugepages, ZFS ARC, Ceph tuning, SDN réseau, cluster HA, monitoring et recettes par workload.

L'optimisation d'un cluster **Proxmox VE 9** est un processus multi-dimensionnel qui touche au système hôte, aux performances CPU, à la gestion mémoire, au tuning du stockage **ZFS** et **Ceph**, à la configuration réseau SDN et à la résilience HA. Ce guide expert compile les meilleures pratiques d'optimisation avec des recettes concrètes adaptées à chaque type de workload, des outils de monitoring et des métriques de référence pour valider les gains. L'optimisation Proxmox ne se limite pas à augmenter les ressources allouées aux VMs : elle passe par une configuration précise de l'hôte (sysctl, scheduler, IRQ affinity), du stockage (ZFS ARC, prefetch, Ceph CRUSH), du réseau (MTU, offloading, SDN) et du cluster (Corosync, HA fencing). Ce guide couvre chaque couche d'optimisation avec les commandes de mesure avant/après, permettant de quantifier les gains et de prendre des décisions basées sur les données. Les recettes par workload (bases de données, VMs Linux, Windows, Kubernetes) permettent une application directe selon votre contexte.

Points clés à retenir

- L'optimisation Proxmox VE commence par le système hôte : scheduler I/O, IRQ affinity, sysctl réseau et limites ZFS ARC avant tout tuning applicatif.
- Le ZFS ARC doit être limité sur les hôtes Proxmox pour laisser suffisamment de RAM aux VMs : règle des 8-16 Go maximum pour ARC.
- Ceph nécessite une séparation stricte des réseaux public/cluster et un calcul précis des Placement Groups pour des performances optimales.
- Le monitoring proactif (Prometheus/Grafana) est indispensable pour identifier les goulets d'étranglement avant qu'ils n'impactent la production.

Optimisation du Système Hôte Proxmox

Les optimisations système de base à appliquer sur chaque nœud Proxmox VE 9. Dans **/etc/sysctl.conf** :

- **vm.swappiness = 10** : minimiser l'utilisation du swap (désactiver avec 0 si suffisamment de RAM)
- **net.core.rmem_max = 134217728** et **wmem_max** : buffers réseau pour les hauts débits
- **net.ipv4.tcp_congestion_control = bbr** : algorithme de contrôle de congestion moderne (meilleur throughput)

- **kernel.numa_balancing = 0** : désactiver l'auto-balancing NUMA si CPU pinning configuré

Le scheduler I/O : pour les disques SSD/NVMe, utiliser **none** ou **mq-deadline** via **/sys/block/{disk}/queue/scheduler**. Pour les disques rotatifs, **mq-deadline** ou **bfq**. L'IRQ affinity permet de dédier des cœurs CPU spécifiques aux interruptions des interfaces réseau 10/25GbE pour réduire la latence.

Optimisation CPU : Pinning, Topology et Fréquence

Le **CPU governor** de l'hôte Proxmox doit être configuré en mode **performance** pour éliminer les latences de changement de fréquence : **cpupower frequency-set -g performance** (persistant via **/etc/init.d/cpufrequtils**). Désactiver également **C-states** dans le BIOS pour les workloads latence-sensitifs.

Le *CPU pinning (vcpu affinity)* dans Proxmox assigne des vCPUs à des cœurs physiques spécifiques via le paramètre **affinity** dans la configuration VM. Pour une VM DB haute performance sur un processeur 32 cœurs avec NUMA 2 domaines (0-15 et 16-31) : assigner les 8 vCPUs aux cœurs 0-7 (domaine NUMA 0) avec la RAM allouée sur le même domaine. La commande de vérification : **numactl --hardware** pour voir la topologie, **taskset -p {pid}** pour vérifier l'affinity.

Le **Hyper-Threading** doit être considéré selon le workload : bénéfique pour les workloads multi-threads légers (serveurs web), potentiellement problématique pour les workloads HPC sensibles au partage de ressources L1/L2. Pour les VMs critiques nécessitant des performances CPU prévisibles, désactiver HT dans le BIOS ou utiliser uniquement les cœurs physiques (pair ou impair selon la numérotation).

Optimisation Mémoire : ZFS ARC et Hugepages

La gestion de la mémoire est critique sur les hôtes Proxmox car **ZFS ARC**, les VMs QEMU et le système hôte se disputent la RAM disponible. La règle d'or : limiter l'ARC ZFS à **8-16 Go maximum** sur les hôtes avec des VMs, quelle que soit la RAM totale. Configuration dans **/etc/modprobe.d/zfs.conf** :

options zfs zfs_arc_max=8589934592 (8 Go en octets)

Désactiver le *ZFS prefetch* pour les workloads avec des patterns d'accès aléatoires (bases de données) : **options zfs zfs_prefetch_disable=1**. Le prefetch est bénéfique pour les accès séquentiels (media, backups).

Les **hugepages** doivent être configurées selon le nombre de VMs et leur RAM totale. Exemple pour 10 VMs de 8 Go chacune (80 Go de hugepages 2 Mo) : **vm.nr_hugepages = 40960** dans sysctl.conf. La RAM hugepages est allouée statiquement au démarrage : s'assurer que la RAM totale hôte = RAM VMs hugepages + ARC ZFS + 4-8 Go pour l'OS hôte.

Optimisation Stockage ZFS

Le tuning **ZFS** pour la virtualisation inclut plusieurs paramètres clés :

- **recordsize** : 16K pour les bases de données (MySQL InnoDB, PostgreSQL), 128K (défaut) pour les workloads généraux, 1M pour le stockage de gros fichiers (backups, media)
- **compression** : **zstd** recommandé (excellent ratio CPU/compression, meilleur que lz4 pour les VMs OS)
- **atime=off** : désactiver la mise à jour du timestamp d'accès (réduit les écritures)
- **sync=disabled** : UNIQUEMENT pour les VMs non-critiques sur baie SSD (risque de perte de données en cas de crash)

Les **ZVOLs** (ZFS Volumes) sont préférés aux fichiers image pour les disques VM : ils se comportent comme des périphériques bloc et offrent de meilleures performances I/O. Configuration via **zfs create -V 100G -s rpool/vm-100-disk-0** (le flag -s crée un ZVOL thin-provisioned). Pour une analyse complète du dimensionnement ZFS, consultez notre guide de dimensionnement Proxmox VE 9.

Optimisation Ceph : CRUSH, PGs et Réseau

L'optimisation **Ceph** pour Proxmox VE 9 commence par la configuration correcte du *CRUSH Map (Controlled Replication Under Scalable Hashing, algorithme de placement des données)*. Le nombre de **Placement Groups (PGs)** doit être calculé précisément : trop peu de PGs = mauvaise distribution, trop de PGs = overhead de gestion. Formule : PGs par pool = (Total OSDs × 100) / facteur_réplication, arrondis à la puissance de 2 supérieure.

Les paramètres Ceph critiques pour les performances :

- **osd_pool_default_size = 3**, **min_size = 2** : réplication 3x, lecture possible avec 2 OSDs
- **osd_journal_size = 10240** (10 Go sur SSD NVMe dédié) pour les HDDs OSD
- **bluestore_cache_size** : limiter le cache BlueStore à 4 Go par OSD pour éviter la contention mémoire
- Réseau Ceph : MTU 9000 (jumbo frames) sur le réseau cluster pour maximiser le débit de réplication

Pour le diagnostic Ceph, **ceph osd perf** affiche la latence apply/commit par OSD. La latence cible en production est < 1ms pour les NVMe, < 5ms pour les SSD SATA. Des latences élevées indiquent généralement un problème réseau ou de disque.

Optimisation Réseau et SDN

Les optimisations réseau sur les hôtes Proxmox VE 9 :

- **MTU 9000 (Jumbo Frames)** sur le réseau dédié Ceph et migration : réduction du nombre de paquets, meilleur throughput
- **TX/RX offloading** sur les interfaces physiques : **ethtool -K {iface} tso on gso on gro on**

- **Multi-queue NIC** : **ethtool -L {iface} combined 8** pour utiliser 8 files d'attente (= nombre de cœurs CPU)
- **VXLAN MTU** : avec jumbo frames à 9000 sur le physique, MTU VXLAN effectif = 8950 (overhead 50 bytes)

Pour les VMs réseau-intensive, activer le **vhost-net** (accélération KVM du réseau virtuel) et configurer la VM avec **VirtIO Net** + **multiqueue=8** pour les VMs multi-cœurs. Consulter notre guide SDN Proxmox VE 9 pour les configurations réseau avancées.

Monitoring et Métriques de Référence

Le monitoring avec **Prometheus + Grafana** est essentiel pour valider les optimisations et détecter les régressions. Métriques clés à surveiller :

- **Latence I/O ZFS** : **zpool iostat -v 1**, cible < 1ms pour NVMe, < 5ms pour SSD
- **Ceph OSD latency** : **ceph osd perf**, cible < 2ms apply latency
- **CPU steal time** : indique la contention CPU entre VMs (cible < 5%)
- **RAM balloon** : monitoring du ballooning (utilisation mémoire effective des VMs)
- **Corosync ring latency** : **corosync-cfgtool -s**, cible < 2ms

La documentation officielle Proxmox VE et le wiki Performance Tweaks complètent ce guide avec des ajustements spécifiques aux versions. Pour les outils de monitoring complets, consultez notre panorama des outils Proxmox VE.

Couche	Paramètre clé	Valeur optimale	Impact
Système hôte	CPU governor	performance	Latence réduite
Mémoire	ZFS ARC max	8-16 Go	RAM disponible VMs
ZFS	compression	zstd	Espace + performances
Ceph	Réseau cluster MTU	9000 (jumbo)	Débit réplication
Réseau VM	VirtIO multiqueue	= nb vCPUs	Bande passante VM

Questions fréquentes

Comment limiter le ZFS ARC pour optimiser la mémoire disponible aux VMs Proxmox ?

Par défaut, **ZFS ARC** peut utiliser jusqu'à 50% de la RAM disponible, ce qui sur un hôte avec 256 Go représente 128 Go potentiellement soustrait aux VMs. La limitation se configure dans **/etc/modprobe.d/zfs.conf** avec le paramètre **zfs_arc_max** en octets. Pour limiter à 16 Go : **options zfs zfs_arc_max=17179869184**. Après modification, mettre à jour le initramfs : **update-initramfs -u -k all** et redémarrer. La valeur en runtime peut être modifiée sans redémarrage via **echo**

17179869184 > /sys/module/zfs/parameters/zfs_arc_max. La taille optimale dépend du workload : plus d'ARC bénéficie aux VMs qui lisent fréquemment les mêmes données du stockage ZFS.

Quels paramètres Ceph optimiser en priorité pour améliorer les performances I/O des VMs Proxmox ?

Les trois optimisations Ceph avec le plus grand impact sur les performances I/O des VMs : 1) **Séparation réseaux public/cluster** sur des interfaces dédiées 10/25GbE (évite la congestion entre trafic client et réplication). 2) **Calcul correct des PGs** : sous-dimensionner les PGs crée des hot spots, surdimensionner génère de l'overhead de gestion. 3) **Déploiement des WAL/DB BlueStore sur SSD NVMe dédiés** séparés des OSDs HDD pour accélérer les opérations d'écriture. En complément, activer les jumbo frames (MTU 9000) sur les réseaux Ceph et s'assurer que les OSDs utilisent BlueStore (défaut depuis Ceph Nautilus) plutôt que FileStore.

Comment mesurer et valider les gains d'optimisation sur un cluster Proxmox VE 9 ?

La validation des optimisations nécessite des mesures avant/après avec des outils standardisés. Pour le stockage : **fio** (flexible I/O tester) mesure les IOPS et latences avec des patterns représentatifs du workload cible (4K random read/write pour les bases de données, 128K sequential pour les backups). Pour le réseau : **iperf3** mesure le débit entre nœuds. Pour ZFS : **zpool iostat -v 1** pendant un test de charge. Pour Ceph : **rados bench**. Les dashboards Grafana avec les métriques Prometheus permettent de comparer les performances avant/après optimisation sur des périodes représentatives de la charge réelle de production.

Sources et références : Proxmox VE Wiki · ANSSI

Articles connexes

- Proxmox VE 9.1 : Paramètres Avancés VM et Nested Virt

Conclusion

L'optimisation de **Proxmox VE 9** est un processus itératif qui passe par la mesure, l'ajustement et la validation à chaque couche : hôte, CPU, mémoire, ZFS, Ceph et réseau. Les gains peuvent être significatifs : 20-50% d'amélioration des performances I/O avec un tuning ZFS correct, 2-3× de débit réseau VM avec VirtIO multiqueue et jumbo frames, et une réduction de la latence de 30-50% avec le CPU pinning sur les workloads critiques.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr