# Virtualisation du stockage

## M346 — Concevoir et réaliser des solutions cloud

Jérôme Frossard

EPAI

18 septembre 2024

# Introduction

On peut dater la naissance du cloud à la mise sur le marché des premiers services web d'Amazon (AWS) au milieu des années 2000.

Mais aucune innovation ne surgit du néant. Ces services sont nés d'un besoin et ont été rendus possibles par un ensemble de technologies développées et théorisées au cours des décennies précédentes. Les technologies liées à l'Internet bien sûr, mais aussi celles liées à la virtualisation (calcul, stockage et réseau) que nous nous proposons d'explorer dans cette série de six présentations.

Dans la deuxième présentation, nous avons abordé les deux formes de virtualisation du calcul (machines virtuelles et conteneurs). Le but de cette présentation est d'aborder quelques aspects de la virtualisation du stockage.

# Plan

# Plan

# Qu’est-ce que la virtualisation du stockage ?

La virtualisation du stockage désigne le regroupement (*pooling*) et l’abstraction de ressources de stockage physiques (SSD, disques durs, volumes RAID, NAS, bibliothèque de bandes, etc.) de manière à les présenter comme un espace de stockage unique et cohérent.

Pour cela, un système logiciel intercepte les demandes d’entrée/sortie (E/S) des machines physiques ou virtuelles et envoie ces demandes à l’emplacement physique approprié des dispositifs de stockage du pool.

Pour l’utilisateur, les différentes ressources de stockage qui composent le pool ne sont pas visibles, de sorte que le stockage virtuel apparait comme un seul lecteur physique, partage ou numéro d’unité logique (LUN) qui peut accepter des lectures et des écritures standard.

# Objectifs de la virtualisation du stockage

À travers l'abstraction et le regroupement des ressources, la virtualisation du stockage vise un certain nombre d'objectifs.

Notamment :

- **Optimiser l'utilisation.** Elle permet d'améliorer l'utilisation des ressources de stockage disponibles dans les différentes baies de stockage ou les différents serveurs, en les centralisant et en les allouant aux différentes charges de travail selon leur besoin.
- **Simplifier la gestion.** Elle permet de faciliter l'approvisionnement, la migration et l'allocation du stockage à l'aide d'interfaces utilisateur·rice·s et d'API.
- **Améliorer la disponibilité.** Elle permet d'améliorer la protection des données contre les pannes grâce à de la redondance.
- **Améliorer l'évolutivité.** Elle permet d'étendre les capacités de stockage de manière flexible et sans interruption, en facilitant l'intégration de nouvelles ressources au sein de l'infrastructure existante.

# Inconvénient de la virtualisation du stockage

Ces avantages ne sont pas gratuits. La virtualisation du stockage a également un certain nombre d'inconvénients.

Par exemple :

- **Complexité accrue.** La gestion des couches d'abstraction supplémentaires peut augmenter la complexité de l'infrastructure et nécessiter des compétences spécialisées pour l'administration.
- **Risques de sécurité.** L'unification des ressources et l'utilisation de systèmes logiciels peuvent créer des points de vulnérabilité uniques, et une mauvaise configuration peut exposer l'ensemble du stockage à des risques de sécurité.
- **Problèmes de performance.** L'ajout de couches virtuelles peut introduire une latence supplémentaire et des goulets d'étranglement, surtout si les ressources ne sont pas correctement dimensionnées ou optimisées.

# Plan

# Qu'est-ce que le stockage local

Le stockage local d'un ordinateur est composé de toutes les unités de stockage connectées directement à l'ordinateur (DAS ou *direct attached storage*). Les principales interfaces sont :

- **SATA** (*serial ATA*) est une évolution du standard ATA (*Advanced Technology Attachment*) ou IDE qui était à l'origine une interface parallèle, et supporte des débits théoriques compris entre 1.5 et 6 Gbs selon la version.
- **SAS** (*serial attached SCSI*) est une évolution du standard SCSI (*Small Computer System Interface*) qui était également une interface parallèle, et supporte des débits théoriques compris entre 3 et 22.5 Gb/s selon la version.
- **NVMe** (*non-volatile memory express*) est une interface de communication qui permet d'utiliser au mieux la bande passante du bus PCIe (> 100 Gb/s).

Dans le cas des disques durs et des SSD bas de gamme, le débit est généralement limité par les disques. Les interfaces NVMe permettent d'utiliser pleinement les performances des SSD haut de gamme.

# Qu'est-ce qu'une unité de stockage physique ?

Une unité de stockage physique peut être, par exemple :

- Un disque dur (*hard disk drive* ou HDD)
- Un disque électronique (*solid-state drive* ou SSD)
- Un volume RAID géré par un contrôleur matériel ou logiciel

Ces unités de stockage sont généralement des unités de stockage par bloc.

Dans une telle unité, un bloc physique est la plus petite unité de mémoire adressable. La taille d'un bloc est fixée par le constructeur et est typiquement de 4096 octets dans les disques durs modernes et les SSD. Des unités plus anciennes peuvent encore utiliser des blocs de 512 octets.

# Limite des unités de stockage physiques

Il est possible d'installer un système de fichiers directement sur une unité de stockage vierge.

Toutefois, il est généralement nécessaire de subdiviser une unité en plusieurs partitions. Chaque partition apparait comme une unité de stockage physique indépendante.

On utilise des partitions pour installer un système de fichiers différent (p. ex. FAT32 pour la partition de boot UEFI), ou pour éviter qu'un dépassement de capacité (p. ex., à cause des logs) n'affecte le fonctionnement du système d'exploitation (SE).

Le principal problème des partitions est qu'elles sont de taille fixe et qu'elles ne peuvent généralement pas être étendues en cas de besoin.

Une solution à ce problème est l'utilisation de la virtualisation du stockage dans le SE.

# Virtualisation du stockage au niveau de l'hôte

La virtualisation du stockage au niveau de l'hôte consiste à virtualiser le stockage attaché à un hôte en utilisant son système d'exploitation.

Les systèmes d'exploitation modernes disposent généralement d'outils permettant de virtualiser le stockage.

Par exemple :

- LVM (*logical volume manager*) sous Linux et NetBSD
- Storage Spaces sous Windows
- Core Storage sous macOS
- ZFS sous Linux et FreeBSD

# Principe de la virtualisation du stockage au niveau de l'hôte

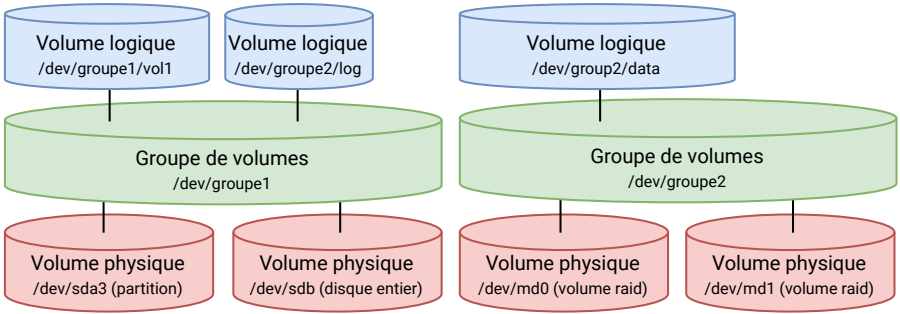
Tous ces systèmes reposent sur des principes similaires :

- Un volume physique (*physical volume* ou PV) est créé à partir d'une unité de stockage physique.
- Les volumes physiques sont regroupés en groupes de volumes (*volume groups* ou VG).
- Un groupe de volume peut être étendu dynamiquement par l'ajout d'un volume physique,
- Les volumes logiques (*logical volume* ou LV) sont des partitions d'un groupe de volume.
- Un volume logique peut être étendu dynamiquement tant qu'il y a de la place sur le groupe de volume.

Ces systèmes permettent également de créer des volumes redondants (RAID) et supportent généralement au moins la création de miroirs (RAID 1). Le ZFS n'utilise pas le RAID, mais dispose de fonctionnalités équivalentes.

# Exemple : Logical Volume Manager sous Linux

Sous Linux, les volumes logiques apparaissent comme des dispositifs de type bloc dans le répertoire **/dev**. Les groupes de volume sont des répertoires (p. ex. **/dev/groupe1**) qui contiennent les volumes logiques (p. ex. **/dev/groupe1/data**).



# Snapshot

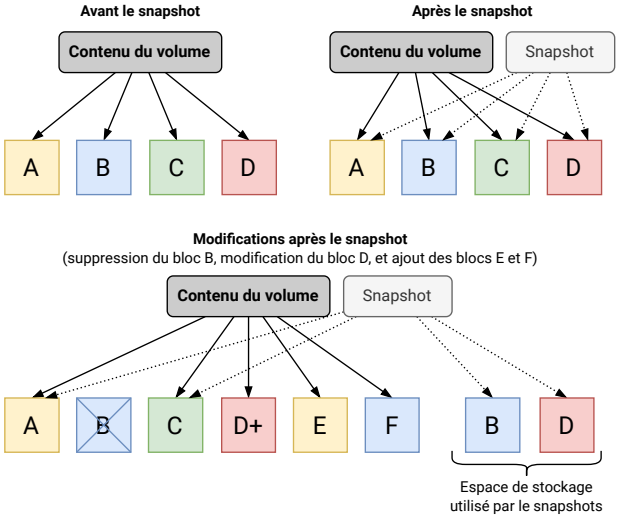
Les instantanés (snapshots) sont une fonctionnalité courante des systèmes de gestion de volumes logiques, qui permet de figer à un instant donné le contenu d'un volume de manière instantanée, grâce à la technique dite du *copy-on-write*. Cela signifie que les blocs ne sont pas copiés immédiatement, mais uniquement lorsque leur contenu est modifié.

Puisqu'il n'y a généralement pas de copie des données, un snapshot n'est pas à proprement parler une sauvegarde, mais il facilite la réalisation d'une sauvegarde cohérente, à chaud :

- Le système de sauvegarde prend un snapshot.
- Il copie le contenu du snapshot (qui est figé).
- Il supprime le snapshot.

L'espace de stockage nécessaire à un snapshot dépend de la fréquence des écritures et de la durée de vie du snapshot.

# Snapshot — Illustration du *copy-on-write*



# Plan

# Problèmes du stockage local dans les serveurs

Dans le cas des serveurs, le stockage local pose plusieurs problèmes.

Il entraîne, par exemple :

- Une utilisation inefficace des ressources avec des capacités souvent sous-utilisées.
- Une complexité accrue pour la gestion et la sauvegarde des données.
- La nécessité d'une interruption de service lorsque les besoins de stockage dépassent la capacité disponible.

C'est pourquoi le stockage local est le plus souvent réduit à ce qu'il faut pour faire tourner le système d'exploitation et le reste est centralisé.

# Stockage centralisé

La centralisation du stockage permet de résoudre ces problèmes en séparant le stockage des serveurs.

On peut distinguer au moins deux approche :

- La centralisation des dispositifs de stockage dans des baies de stockage, et la mise à disposition de volume logique à travers un réseau spécialisé appelé SAN (*storage area network*).
- La centralisation de la gestion et de l'accès au stockage à l'aide d'un système de stockage distribué.

# Plan

# Qu’est-ce qu’un SAN ?

Un SAN (Storage Area Network) est un réseau dédié qui permet de connecter des serveurs à des unités de stockage partagées. Du point de vue de l’utilisateur·rice, une unité de stockage du SAN à laquelle le serveur est connecté, apparait comme une unité de stockage par bloc locale.

Pour cela, les SAN utilisent des protocoles spécialisés comme FCP (*Fibre Channel protocol*) ou iSCSI. Ces protocoles supportent des vitesses de transfert élevées (jusqu’à plusieurs centaines de gigabits par seconde) et une faible latence.

En permettant la centralisation des ressources de stockage, un SAN facilite la gestion des ressources de stockage et permet d’en optimiser l’utilisation en allouant à chaque serveur ce dont il a besoin.

# Qu'est-ce qu'une baie de stockage ?

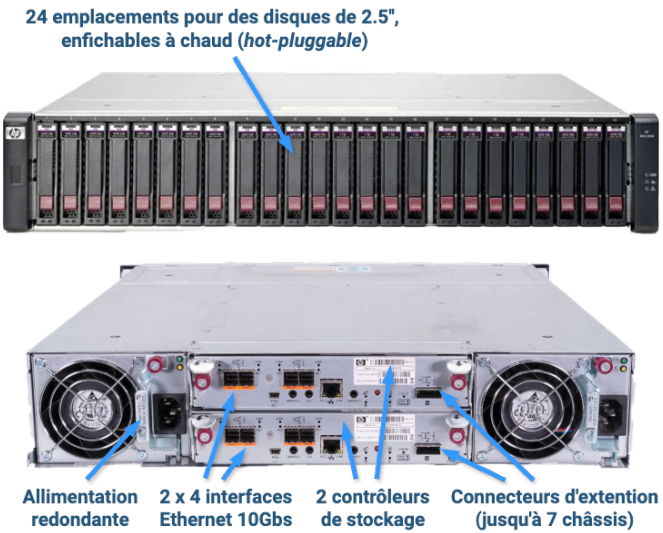
Une baie de stockage (*storage array*) regroupe dans un même châssis, un ou deux serveurs spécialisés, appelés contrôleurs de stockage, et un relativement grand nombre (jusqu'à une centaine) d'unités de stockage physique (disques durs ou SSD).

Le contrôleur de stockage permet de créer et gérer les volumes RAID (typiquement RAID 6 ou RAID 10) et de mettre ces volumes à disposition sur le SAN. Chaque volume est identifié par un numéro appelé LUN (*logical unite number*).

Chaque volume peut être connecté à un ou plusieurs serveurs à travers un SAN.

**Remarque :** Un volume connecté à un serveur apparait comme une unité de stockage par bloc. Si plusieurs serveurs sont connectés à une même unité, il est important d'assurer que seul l'un d'entre eux peut y accéder en écriture, ou alors qu'un système de fichiers spécial appelé *clustered file system* est installé sur cette unité. Par exemple : VMware VMFS (*Virtual Machine File System*) ou Microsoft CSV (*Cluster Shared Volume*).

# Exemples de baie de stockage



# Protocoles des SAN

Aujourd'hui, les débits élevés et la faible latence de l'Ethernet permettent sa mise en œuvre dans un SAN, mais cela n'a pas toujours été le cas. C'est pourquoi beaucoup de SAN utilisent plutôt le **Fibre Channel (FC)**, un type de réseau spécialement conçu pour le stockage.

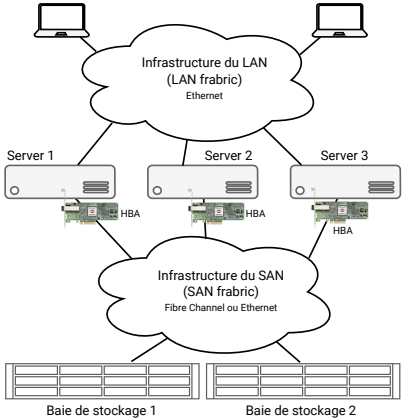
Les principaux protocoles utilisés pour connecter un LUN à un serveur sont :

- **FCP** est une implémentation du protocole SCSI pour le réseau FC. Ce protocole peut également être utilisé sur un réseau Ethernet avec FCoE (*Fibre Channel over Ethernet*).
- **iSCSI** qui permet d'utiliser le protocole SCSI sur un réseau TCP/IP.
- **NVMe-oF** (*NVMe over fabric*) qui permet d'utiliser le protocole NVMe au-dessus d'un réseau FC ou Ethernet.

Pour utiliser ces protocoles, un serveur peut être équipé d'une carte réseau spéciale appelée HBA (*host bus adapter*) qui prend en charge une grande partie du protocole pour épargner les ressources du serveur.

# LAN et SAN

Dans tous les cas, LAN et SAN sont des réseaux distincts pour éviter les interférences entre les différents type de trafic.



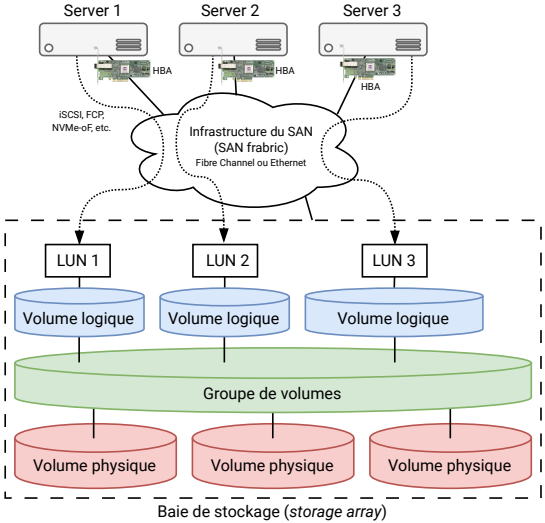
# Virtualisation au niveau de la baie de stockage

Avec un SAN, la virtualisation du stockage peut être mise en œuvre dans la baie de stockage. Puisqu'un contrôleur de stockage n'est rien d'autre qu'un serveur, le principe n'est pas très différent de celui de la virtualisation au niveau de l'hôte.

Dans un contrôleur de stockage, le logiciel de virtualisation permet typiquement :

- D'agréger plusieurs volumes RAID (les volumes physiques) en un **groupe de volumes**.
- De créer des **volumes logiques** de taille variable dans un groupe de volumes.
- D'étendre la capacité de ce volume logique tant qu'il reste de la place dans le groupe de volume.
- D'étendre la capacité d'un groupe de volume en ajoutant un volume physique à ce groupe.
- De réaliser des snapshots (instantané) d'un volume, pour conserver une copie de son contenu à un instant donné.

# Schéma de principe



# Avantage et inconvénients

En plus de faciliter grandement la gestion des unités logiques, par rapport à l'utilisation de volumes physiques de taille fixe, comme dans le cas de la virtualisation du stockage au niveau de l'hôte, la virtualisation du stockage au niveau de la baie de stockage offre les avantages suivants :

- Centralisation de la gestion du stockage.
- Performance du logiciel optimisé pour le matériel.

Les principaux inconvénients de cette solution sont qu'elle repose sur du logiciel propriétaire, et qu'elle n'offre pas d'interface de gestion unifiée.

De plus, certaines fonctionnalités avancées, comme le stockage hiérarchisé (*tired storage*), peuvent être verrouillées et nécessiter l'achat de licence supplémentaire pour être activées.

# Plan

# Qu'est-ce qu'un système de stockage distribué ?

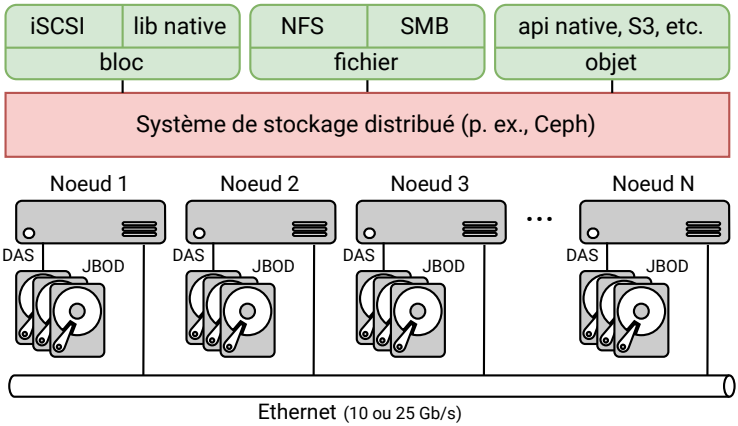
Plutôt que de centraliser les disques dans des baies de stockage connectées avec un réseau spécialisé, une autre option est d'utiliser un système logiciel pour rendre la capacité des disques des différents serveurs, disponible sous la forme de volume, de pool d'objets, de partage, etc., accessible des moyens tels que iSCSI, API REST, SMB, NFS, etc.

Le stockage distribué est rendu possible par les débits importants et la faible latence des réseaux Ethernet.

Parmi les implémentations de stockage distribué, on peut mentionner :

- Red Hat Ceph (stockage par objet, par bloc, et par fichier)
- Red Hat Gluster (stockage par bloc et par fichier)
- SUSE Longhorn (stockage par bloc)
- StarWind VSAN (stockage par bloc)
- VMware vSAN (stockage par bloc)

# Schéma de principe d'un système de stockage distribué



# Grappe de serveurs et redondance

Pour assurer la disponibilité des données, un système de stockage distribué ne repose pas sur le RAID, mais sur la copie des données sur au moins trois hôtes différents d'une grappe de serveur (*cluster*). Il est possible d'augmenter le nombre de copies pour augmenter la tolérance de panne, mais on ne devrait pas utiliser moins de trois copies en production.

Pour réaliser une grappe de stockage distribuée avec trois copies, il est nécessaire d'avoir au moins trois nœuds (serveurs) équipés d'au moins un disque. Pour cinq copies, il est nécessaire d'avoir au moins cinq nœuds.

- Avec trois copies, la grappe continue à fonctionner en cas de panne d'un nœud.
- Avec cinq copies, la grappe continue à fonctionner en cas de panne de deux nœuds.
- Au-delà, il n'y a aucune perte de données, mais la grappe n'est plus en mesure de traiter de nouvelles requêtes.

# Contrôleur de la grappe

Le système de stockage distribué nécessite un processus de contrôle, qui assure que les données sont bien répliquées dans les différents hôtes et qui maintient un index pour en mesure de retrouver ces données. Pour que la grappe fonctionne correctement, il doit y avoir à tout instant un et un seul contrôleur actif dans la grappe.

Pour avoir une grappe à haute disponibilité (*high availability cluster* ou *HA cluster*), qui continue à fonctionner en cas de panne, il est nécessaire d'avoir au moins un autre nœud avec un contrôleur secondaire qui maintient un réplica de l'index du contrôleur actif et qui est prêt à prendre le relai.

Pour assurer qu'il n'y ait qu'un contrôleur actif, ce contrôleur doit être élu, et pour assurer la cohérence des données, cette élection ne peut avoir lieu que s'il y a un nombre minimal de votants. Ce nombre minimal de votants est appelé quorum.

Dans un système distribué, le plus petit quorum est de deux. Pour assurer la disponibilité en cas de panne d'un nœud, il doit donc y avoir trois contrôleurs répartis sur trois nœuds.

# Théorème CAP

Une grappe de serveurs est une forme de système distribué, c'est-à-dire un système qui utilise des ressources de différentes machines pour produire un résultat.

Selon le théorème CAP, un tel système ne peut assurer que deux des trois propriétés suivantes :

- Cohérence (*Consistency*)
- Disponibilité (*Availability*)
- Tolérance au partitionnement (*Partition tolerance*)

Dans le cas d'un système de stockage distribué, la cohérence est primordiale. Il s'en suit que le système ne sera pas disponible (ou disponible en mode dégradé) dans le cas d'un partitionnement à cause de la panne d'un serveur ou du réseau.

# Split-brain

Dans le cas d'une défaillance d'un équipement ou d'un câble réseau, il est possible que le réseau soit partitionné.

Faisons les hypothèses suivantes :

- Chaque partition contient une partie de la grappe
- Chaque partie de la grappe est accessible par une partie des clients

Pour chaque partie de la grappe, les nœuds de l'autre partie sont hors-ligne.

Si chaque partie de la grappe procède à l'élection d'un nouveau nœud actif, on se retrouve avec deux contrôleurs actifs. Mais comme ces contrôleurs ne peuvent pas communiquer entre eux, la cohérence des données n'est plus assurée.

Cette situation est appelée un *split-brain* et doit absolument être évitée.

# Quorum

Comme nous l'avons déjà dit, un quorum est le nombre minimal de votants pour qu'un vote puisse avoir lieu.

Pour éviter un *split-brain* il faut :

- un nombre impair de nœuds dans la grappe
- un quorum correspondant à la moitié des nœuds + 1.

Avec trois nœuds et un quorum de deux :

- Avec un groupe de deux nœuds et un nœud isolé, seul le groupe de deux pourra élire un contrôleur actif.
- Avec trois nœuds isolés, aucune élection n'est possible.

De cette manière, on assure qu'il n'y a jamais plus d'un nœud actif.

# Avantage et inconvénients

En plus des avantages d'un SAN avec des baies de stockage, un système de stockage distribué présente les avantages suivants :

- Réduction du risque d'enfermement propriétaire par l'utilisation de matériel courant (serveurs, ssd, disques durs) et hétérogène.
- Haute disponibilité et sécurité élevée.
- Support d'API standards comme l'API de S3 en plus d'interfaces plus traditionnelles comme iSCSI.
- Extensibilité quasi illimitée. Par exemple, en 2023, le CERN possède 17 cluster pour un total de 100 Po

Mais il peut également présenter quelques inconvénients :

- Coût élevé pour un petit cluster (seulement 1/3 du stockage disponible avec 3 nœuds).
- Plus complexe à mettre en œuvre et à gérer qu'un solution propriétaire.

# Exemple d'un cluster Ceph

L'image ci-contre montre une partie d'un cluster Ceph qui se trouve au CERN. Dans chaque rack, on voit deux châssis de 4 serveurs, et un châssis de 24 disques par serveur. Un groupe de 24 disques, appelé JBOD (*just a bunch of disk*), est connectés directement à chaque serveur (DAS ou *direct attached storage*).

# Plan

# Différents types de services de stockage

La centralisation et la virtualisation du stockage permettent d'optimiser l'allocation de la bonne capacité de stockage pour chaque charge de travail de l'infrastructure.

Mais cette capacité de stockage peut être mise à disposition de différente manières. On peut en distinguer au moins trois type de services :

- Service de stockage par bloc (*block storage*).
- Service de stockage par fichier (*file storage*).
- Service de stockage par objet (*object storage*).

# Service de stockage par bloc

Un service de stockage par bloc présente une capacité de stockage sous la forme d'une unité de stockage par bloc qui apparait au système d'exploitation d'un serveur ou d'une machine virtuelle (VM) comme un disque local.

- Disque de démarrage (*boot drive*). Un serveur démarre sur disque local dans lequel est généralement installé le système d'exploitation.
- Ce type de stockage est également celui qui offre les meilleures performances (débit et opération d'E/S par seconde) et la plus faible latence.

Une unité de stockage par bloc ne peut pas être utilisée simultanément par plusieurs serveurs. Si une unité est connectée à tous les nœuds d'une grappe (*cluster*), il faut :

- Installer un système de fichiers pour cluster (*clustered file system*) sur cette unité.
- Assurer qu'à tout instant un seul serveur de la grappe ait le droit d'écrire sur cette unité.

# Machine virtuel et disque virtuel

Comme pour un serveur physique, une machine VM démarre sur un disque local.

- Pour assurer de bonnes performances pour une VM, l'accès au disque doit être rapide et sûr.
- L'émulation n'est donc pas une option.
- Lorsque c'est possible, un accès direct à un volume physique ou virtuel de l'hôte est la meilleure option.
- Lorsque ce n'est pas le cas, la paravirtualisation est une très bonne seconde option. Le principe consiste à installer dans l'SE invité, un pilote qui accède à une image du disque virtuel en communiquant directement avec l'hyperviseur.

# Image de disque virtuel

Une image de disque virtuel prend la forme d'un fichier stocké dans un système de fichiers accessible par l'hyperviseur.

Les principaux formats d'image de disque virtuel sont :

- VMDK (*Virtual Machine Disk*) : VMware WorkStation et ESXi
- VHD et VHDX (*Virtual Hard Drive*) : Microsoft Hyper-V
- VDI (*Virtual Disk Image*) : Oracle VirtualBox
- Qcow2 (*Qemu copy-on-write*) et raw : KVM

# VM et disque virtuel dans le cloud

Par défaut, dans le cloud, le disque de démarrage d'une VM est en général une unité de stockage temporaire.

- Cela signifie que le contenu du disque peut être réinitialisé au redémarrage de la VM.
- Ce stockage temporaire convient pour des charges de travail sans état comme un serveur Web, par exemple.
- Mais il ne convient pas pour des charges de travail avec état comme un serveur de base de données.

Pour avoir du stockage permanent, on utilise un service de stockage par bloc pour ajouter un disque supplémentaire. Le tarif dépend de paramètres tels que :

- La capacité.
- La vitesse et la latence du disque (SSD, HDD haut de gamme, HDD bas de gamme, etc.)
- Le nombre et la localisation des réplicas.

# Système de fichiers

En dehors de quelques cas particuliers, principalement des SGBD (Oracle, DB2, SQL Server, etc.), qui utilisent des partitions 'raw' pour optimiser les performances des E/S, il est assez rare qu'une application utilise directement une unité de stockage par bloc.

Le plus souvent, on installe un système de fichiers (p.ex., ext4 sous Linux, NTFS sous Windows, etc.) sur l'unité de stockage pour en faciliter l'utilisation :

- Un fichier est un ensemble de blocs, pas forcément contigu, associé à des métadonnées (nom, date de création, date de modification, etc.) et des attributs.
- Un répertoire est un fichier spécial dont le contenu est une liste de liens vers d'autres fichiers et répertoires, qui permet d'organiser les fichiers de manière hiérarchique.
- Les métadonnées et les attributs sont stockés de manière centralisée, indépendamment des données, dans une structure maintenue par le système de fichiers, par exemple, la *inode table* dans ext4 ou la *master file table* dans NTFS.

# Avantages d'un système de fichiers

Pami les avantages d'un système de fichier, on peut mentionner :

- **Facilite le nommage des fichiers.** Le nom d'un fichier doit être unique dans le répertoire où il se trouve, mais il n'a pas à être unique dans tout le système de fichier. Le chemin complet d'un fichier permet de l'identifier de manière unique dans le système de fichier.
- **Facilite la gestion des blocs**. Le système de fichier gère l'allocation et la libération des blocs pour augmenter ou réduire la taille d'un fichier lorsqu'on le modifie, ainsi que l'enchaînement des blocs non contigus.
- **Interface universelle**. Même si chaque système de fichiers a ses spécificités, l'interface d'accès est essentiellement toujours la même. Cela permet d'exposer tout ou partie d'un système de fichier à l'aide de protocole standard tel que SMB ou NFS à travers un réseau.

# Serveur de fichiers et NAS

Un serveur de fichiers (*file server*) est un serveur qui partage un ou plusieurs répertoires de son système de fichier à travers un réseau à l'aide d'un protocole tel que SMB ou NFS.

Par rapport à une unité de stockage par bloc, un répertoire partagé (*shared folder*) :

- Peut être utilisé simultanément par plusieurs serveurs.
- A des débit souvent plus faible et une latence généralement plus élevé.
- Ne peut pas être utilisé comme disque de démarrage.

Un NAS (Network Attached Storage) est un serveur optimisé pour le stockage et le partage des fichiers sur un LAN.

À la maison ou dans une petite entreprise, un NAS se présente souvent sous la forme d'un serveur « tout en un » avec un SE propriétaire géré via une interface web, utilisé pour exécuter diverses charges de travail supplémentaires (sauvegarde, serveur multimédia, etc.).

# Service de stockage par fichier

Dans le cloud, un service de stockage par fichier (*file storage service*) permet de présenter une capacité de stockage de la même manière qu'un répertoire partagé sur un serveur de fichier ou un NAS.

Cela permet de migrer des données sur le cloud sans changer les habitudes des utilisateurs ou en maintenant la compatibilité avec des applications existantes.

Parmi les principaux cas d'utilisation des services de stockage par fichier, on peut mentionner :

- Collaboration, partage et diffusion de fichiers.
- Stockage de fichiers pour une application Web ou un système de gestion de contenu.
- Stockage de fichiers log.
- Sauvegarde de base de données.

# Plan

# Limites du stockage par fichier

Le stockage par fichier, bien que largement utilisé et pratique, présente certaines limitations dans des environnements à grande échelle ou pour certaines applications modernes :

- **Utilisabilité.** Si le nombre de documents et la profondeur de la structure deviennent trop importants, il peut devenir difficile de nommer et retrouver facilement les fichiers.
- **Mise à l'échelle.** Bien qu'un système de fichier puisse théoriquement gérer des Po, en pratique, la limite est plutôt d'une centaine de To, principalement à cause de la centralisation des métadonnées et de l'indexation.
- **Performance.** La gestion centralisée des métadonnées et de la table d'allocation des blocs peut devenir un goulet d'étranglement avec un grand nombre d'accès simultanés.
- **Manque de flexibilité.** Les systèmes de fichiers ne sont pas toujours adaptés aux applications modernes qui manipulent des fichiers de données non structurées (images, vidéos, logs, etc.) pouvant atteindre de très grandes tailles.

# Qu’est-ce que le stockage par objet ?

Le stockage par objet est un modèle de stockage qui gère les données sous forme d’objets, plutôt que de blocs (comme dans le stockage par bloc) ou de fichiers (comme dans le stockage par fichier).

Chaque objet est constitué de trois éléments :

- **Un identifiant unique.** Un identifiant global unique (comme un hash ou un UUID) qui permet de localiser l’objet dans le système de stockage.
- **Les données.** Les données à stocker, par exemple, une image, un document, etc.
- **Des métadonnées.** Des informations associées à l’objet, comme la date de création, le type de données, les permissions d’accès, ou d’autres attributs spécifiques définis par l’utilisateur ou le système.

# Objets et bucket

Avec un service de stockage par fichier, on accède à un répertoire partagé via un protocole réseau comme SMB ou NFS.

Avec un service de stockage par objet, on accède à un *bucket* via une API REST au-dessus du protocole HTTP.

- Dans le jargon du stockage par objet, un bucket est l'équivalent d'un répertoire.
- Un bucket peut contenir un nombre illimité d'objets.
- Mais contrairement à un répertoire qui peut contenir des sous-répertoires, il n'est pas possible de créer un bucket dans un bucket.
- Les objets sont donc stockés « à plat » (sans structure hiérarchique) et doivent avoir un nom unique dans le bucket.

# Limitation du stockage par objet

Avec un service de stockage par objet, la création ou la suppression d'un objet est similaire à la création d'un fichier. En revanche, il n'est pas possible de modifier une partie du contenu ou des métadonnées d'un objet.

Pour modifier un objet, il faut :

- Télécharger l'objet (*download*)
- Modifier les données de cet objet,
- Téléverser (*upload*) l'objet en entier.

Le stockage par objet n'est pas adapté pour le stockage de fichier dont le contenu doit être souvent modifié.

# Cas d'utilisation du stockage par objet

Parmi les principaux cas d'utilisation du stockage par objet, on peut mentionner :

- Stockage et diffusion de contenu *riche media* (image, musique, vidéo).
- Analyse de big data.
- Sauvegarde et archivage.
- Enregistrement des données d'objets connectés (IoT).

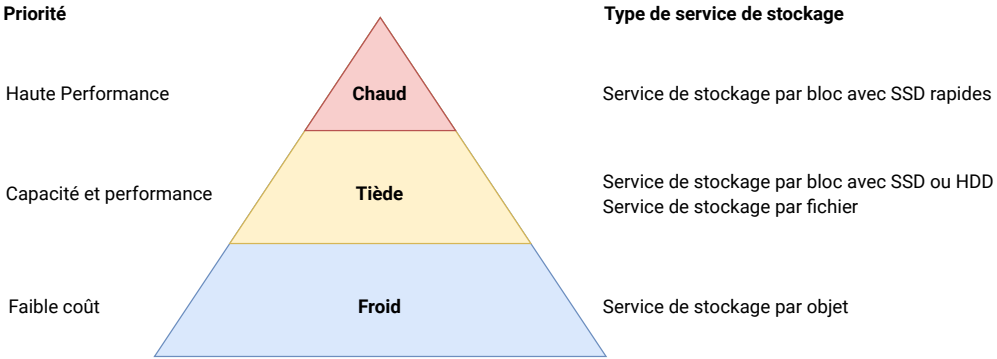
Les services de stockage par objet sont généralement des services serverless et s'intègrent donc très bien avec d'autres services serverless comme le FaaS.

# Température des données

On peut catégoriser les données en fonction de la fréquence d'accès à ces données :

- **Données chaudes (*hot data*).** Des données qui sont activement utilisées et régulièrement sollicitées. Ces données sont essentielles pour dans le traitement des transactions en ligne.
- **Données tièdes (*warm data*).** Des données moins sollicitées que les données chaude, mais qui doivent rester facilement accessibles. Ces données sont utilisée pour l'analyse ou la création de rapports périodique.
- **Données froides (*cold data*).** Des données rarement consultées. Ces données sont souvent stockées par obligation réglementaire, ou pour la conservation historique.

# Température des données et types de service



# Classes de stockage

Le stockage par objet est toujours utilisé pour des données froides.

Toutefois, cette catégorie couvre des données encore un peu tièdes, jusqu'à des données gelées qui ne sont plus du tout utilisées.

C'est pourquoi il existe plusieurs classes de stockage par objet, selon que les données doivent être rapidement accessibles ou non en cas de besoins.

Par exemple, pour des données particulièrement froides, un service peut utiliser des bandes magnétiques pour réduire le coût du stockage. Dans ce cas, un accès aux données reste possible, mais cela peut alors demander plusieurs heures, et le coût de transfert est généralement sensiblement plus élevé.

Enfin, certains services peuvent mesurer la fréquence d'accès aux données et modifier dynamiquement la classe de stockage (*dynamic tiering*).