Caractéristiques et fonctionnalités des systèmes de fichiers locaux
ANF Mathrice 2018
SSA3

Vincent Bayle

AFMB UMR7257

December 4, 2018

Introduction

Pour un système de fichiers, on trouve deux significations :

- Le système de fichier virtuel présenté à l'utilisateur (sous Unix), arbre inversé avec /pour racine
- L'organisation des données sur un espace physique ou logique

Présentation limitée aux systèmes récents, rattachés à des disques durs en local

Méta-données

Du système de fichier : mécanique interne

Pour les fichiers, elles peuvent avoir une utilité pour l'utilisateur :

- Nom
- dates (création, modification, accès)
- droits
  - attributs spécifiques (par exemple read only)
  - propriétaire, groupe
  - type de droits (lecture écriture-modification suppression exécution)
  - ACLs
  - Audit des fichiers (SACL NTFS : qui a copié, regardé, etc.)
- type de fichier (fichier régulier, répertoire, lien, raccourci...)
- indication du type de contenu (extension, type MIME)
- d'autres, éventuellement programmable dans certains systèmes de fichiers

Création, Diffusion

- Quand a été lancé le code pour le système de fichiers (maturité/obsolescence)
- Qui produit le système de fichiers :
  - Société commerciale (associé à un système d'exploitation)
  - Monde libre, et dans ce cas, mode :
    - Cathédrale
    - Bazar
- Diffusion des caractéristiques des systèmes de fichiers :
  - Libre (différents types de licenses)
  - ou pas :
    - caractéristiques non divulguées
    - protégées par des brevets : licenses d'utilisation

Représentation des données sur le disque

- Forme sur le disque
  - Composition
    - en bloc
    - inode
    - domaines (extents)
  - emplacement des méta-données des fichiers
  - Nom des fichiers
    - taille (nombre de caractères)
    - extension
    - jeu de caractères utilisés
- ACLs disponibles
- Quotas disponibles sur le système de fichier :
  - pour un utilisateur
  - pour un sous espace
  - avec limitation de l'espace utilisé et / ou du nombre de fichiers
- Sommes de contrôle
  - pour des transactions
  - pour des méta-données
  - pour les données
- De cette représentation sur le disque (et quelquefois d'une taille de bloc paramétrable), découlent des limitations sur les différentes tailles :
  - Taille maximale du système de fichier
  - Taille maximale d'un fichier
  - Nombre de fichiers total, par répertoire

Journalisation

Dispositif qui permet de s'assurer de la cohérence du système de fichier, par rapport à une opération d'écriture interrompue

- Ecriture dans un journal des opérations
- En cas de redémarrage, les transactions peuvent être rejouées
- Concerne plutôt les méta-données du système de fichiers
- Impact éventuel sur les performances
- Permet d'éviter un fsck (file system check), en cas de redémarrage intempestif
- Deux manières de stocker :
  - logique
  - physique
- Trois manières de se comporter vis-à-vis de la cohérence :
  - Writeback
  - Ordered
  - Data

Ecritures Copy On Write

- Dispositif copié de la gestion de la mémoire virtuelle des ordinateurs (fork)
- Sur les systèmes de fichiers, plutôt Redirect on Write [1] :
  - Quand une écriture est réalisée, on utilise un nouvel espace (bloc p.ex.)
  - Et les pointeurs sont ensuite modifiés
- Les données (et métadonnées) restent donc cohérentes
- Et le nombre d'écritures reste limité

[1] Pour une opération réellement Copy on Write, copie du bloc, puis modification de l'original, soit 2 écritures

Réduction de l'espace utilisé : Compression, Déduplication

Compression : Différents algorithmes disponibles Implémentée par fichier

Déduplication : En comparant avec une base de données des blocs déjà stockés sur le système
Peut-être réalisée :

- online : nécessite beaucoup de mémoire pour une base de blocs
- post-traitement

Les éléments dé-dupliqués peuvent être des blocs, des fichiers, etc.

Chiffrement

Les systèmes de fichiers offrent diverses possibilités quand au chiffrement

- D'un volume ou sous-volume
- Des fichiers
- Eventuellement par utilisateur

Vérification de l'intégrité du système de fichier : fsck, scrub

fsck : File System CheckK consiste à vérifier que le système de fichier est dans un état cohérent
Mais cela ne présume en rien de l'état des données sur le disque

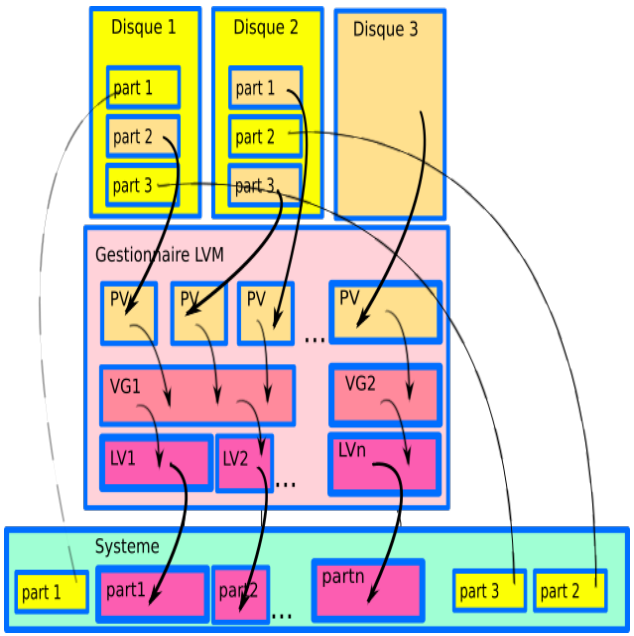
scrub : est une opération de vérification systématique liée à l'utilisation de somme de contrôle

- sur les méta-données
- et sur les données

Le mécanisme :

- le système de fichier est parcouru de manière exhaustive
- la somme de contrôle des éléments lus est comparée à une version initiale
- en cas d'erreur :
  - en cas de redondance suffisante, l'information est récupérée
  - sinon, elle est perdue (et identifiée comme telle)

Gestion de volumes logiques



Sur le modèle du gestionnaire de volume logique LVM de linux :

- Agrégation d'espaces physiques de stockage (couche physique : PV)
- Couche intermédiaire de gestion (zpool, VG)
- Présentation au système de volumes logiques (datasets, LV) : partitions ou datasets

Le système peut intégrer de la répartition des données :

- A des niveaux différents (zpool ou LV)
- Et permettre des répartitions différentes :
  - Stripping ou RAID0
  - Mirroring (RAID1)
  - RAID10
  - Simple parité (RAID5 ou raidz), double parité(RAID6 ou raidz2), triple parité (raidz3)

Gestion de volumes logiques

Notion associées :

Sous-volume :
- Périphérique en mode bloc
- Système de fichier rattaché
  - Sous-répertoire du répertoire parent (hiérarchie possible)
  - Point de montage éventuel dans le système de fichier virtuel
  - Quota
  - Espace réservé

Thin-Provisionning : Partage non-exclusif de l'espace disponible entre plusieurs sous-volumes (pas d'espace réservé)

Tiering : Mise en place de stockage sur des espaces utilisant des technologies différentes, plus ou moins rapide.

Snapshots

Un snapshot (instantané en français) est une version du système de fichier, dans un état cohérent

- Dépendant de la technologie sous-jacente
- En lecture seule, ou lecture-écriture
  - Système copy on write : ensemble de pointeurs vers les blocs à l'instant ou le snapshot est pris.
    - Taille mobilisée par un snapshot minimale (principalement les modifications)
    - Ecriture de quelques méta-données supplémentaires
    - Destruction des snapshots instantanée
    - Possibilité d'envoi de différentiels entre snapshot, pour diminuer les tailles des sauvegardes
  - Système non copy on write (LVM) : taille initiale (maximale) figée.
    - Quand il y modification d'un bloc, réécriture du bloc initial dans le snapshot, et du nouveau à la place
    - Multiplication des écritures (i.e. 1 snapshot = 2 écritures, 8 snapshots = 9 écriture) : perte de performance
    - Une fois rempli, un snapshot est complètement inutilisable
    - Une destruction de snapshot (merge) n'est pas instantanée (attention au reboot...)

En conséquence :

- Le nombre d'instantané peut être limité
- La consommation de ressources peut être conséquente : espace disque, ou autre (cas BTRFS p.ex.)
- Impact (ou pas) sur les performances
- Possibilité de différentiels entre 2 snapshots, incrémental

Clones

Pour un système de fichiers, un clone est un fork d'un système de fichier. Mais en lecture écriture :

- Possibilité de basculer d'un clone à l'autre
- Consommation de ressources
- Impact sur les performances

Certains systèmes de fichiers permettent de créer un clone d'un fichier :

- Lié à des systèmes copy on write
- Différent d'un lien permanent :
  - Création d'un nouvel inode
  - Seuls les nouveaux blocs des 2 fichiers seront différents

Dispositifs de cache

Il existe des dispositifs de cache à plusieurs niveau :

- En lecture
- En écriture

Les dispositifs vont utiliser plusieurs supports :

- Mémoire
- Disques ou éléments de stockage plus rapides (SSD, NVME, mémoire)

HFS, HFS+

- Développé par Apple
- Intégré aux systèmes Mac Os et Mac OsX
- Hierarchical Filesystem depuis 1981, remplacé par HFS+ en 1998
- Journalisation apparue avec OsX 10.3,
- ACLs de type NFSv4 à partir de OsX 10.4
- Compression à partir de OsX 10.6

APFS :

- Création, Diffusion : développé par Apple, depuis 2014
  - Première version diffusée en 2017 (IOS 10.3)
  - Système de fichier par défaut depuis OsX 10.13 (disponible depuis 10.12)
- Stockage des données sur le disque :
  - Utilisation de B-Tree (contenu des répertoires, allocation des fichiers, ...)
  - Somme de contrôle pour les meta-données
  - ACLs standards
  - Quelques chiffres :
    - Taille max du nom de fichiers : 255 caractères en UTF16
    - Taille max de volume théorique : 256 To en pratique
    - Taille max de fichiers théorique : 8 Eio
    - Nombre max de fichiers : 2^{63}
- Gestion de l'espace disque et de la vitesse
  - Partition, gestion de ≪ sous-volume ≫ : via un partage d'espace
  - Quotas
  - Compression : implémentée
  - Déduplication : non
  - Cache / Tiering : non

APFS :

- Gestion de l'intégrité, de la sécurité, possibilité de conserver des états antérieurs / différents.
  - Journalisation : dispositif plus intelligent
  - Copy On Write : le système est dôté d'un dispositif copy on write (révolutionnaire, permet d'éviter la double écriture du journal)
    - Snapshots read only,
    - Clones de fichiers disponibles
  - Vérification de l'intégrité des méta-données (somme de contrôle)
  - Chiffrement : disponible
- Autres éléments : Alternate Data Stream, gestion des fichiers creux (sparse files)

FAT (12-16-32) / VFat / ExFat

- Système mono-utilisateur
- Diffusé par microsoft depuis MS-DOS, taille des indices a grandi avec la taille des périphériques
- Initialement, limitation des noms à 8 + 3 caractères
- VFAT pour parer à cette limitation
- (Dé)Fragmentation : certains ont vu bouger des carrés de couleur...
- ExFat depuis 2006
  - Soumis à une license microsoft, protégé par des brevets
  - Répandu dans les dispositifs mobiles
  - Redevance microsoft dans les appareils mobiles

NTFS : New Technology File System

- Création, Diffusion : développé par Microsoft, première version en 1993.
- Développeurs débauchés de chez DEC
  - Plusieurs versions jusqu'en 2001(Version 3.1, livrée avec Windows XP)
- Stockage des données sur le disque :
  - Utilisation de B+Tree (contenu des répertoires, allocation des fichiers)
  - ACLs standards (DACL) et d'audit : SACL
  - Défragmentation : possible
  - Quelques chiffres :
    - 256 caractère par nom de fichier (UTF16)
    - Taille max de volume théorique : 16 EiO (256 To en pratique)
    - Taille max de fichiers théorique : 16Eio (16To en pratique)
    - Nombre max de fichiers : $2^{32} - 1$

  Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : via le Logical Disk Manager ou Storage Spaces
  - Modification possible de la taille : depuis Windows Vista
  - Partition, gestion de $\ll$ sous-volume $\gg$: non
  - Quotas : implémentés, depuis la version 3.0 (Windows 2000)
  - Compression : implémentée, alogrithme LZNT1
  - Déduplication : non
  - Cache / Tiering : non

NTFS : New Technology File System

- Gestion de l'intégrité, de la sécurité, possibilité de conserver des états antérieurs / différents.
  - Journalisation : implémentée depuis l'origine
  - Copy On Write : le système semble être doté de dispositif copy on write
    - Snapshots disponibles via un service Volume Shadow Copy Service (VSS)
    - Persistent depuis vista (accessible Previous Version / System Restore)
  - Vérification de l'intégrité (fsck / scrub). Utilitaire ChkDSK, équivalent fsck
  - Chiffrement : disponible depuis la version 3.0, via le service EFS
- Autres éléments : Alternate Data Stream
- Manipulation retardée sur d'autres OS (la structure n'est pas documentée)
- Remplaçant en cours de création : Sytème ReFS
  - Utilisation de sommes de contrôle (sur les méta-données ou les données)
  - Allocation on write
  - Disponible depuis Windows Server 2012

Liste des systèmes de fichiers décrits sur les systèmes Unix

- Suite ext :
  - ext2
  - ext3
  - ext4
  - btrfs
- reiserfs
- xfs
- jfs
- UFS
- ZFS

ext2

- Lancé en 1994, développé par Rémi Card,
- Stockage des données sur le disque :
  - découpage en blocs (de 1k à 8k), regroupés en groupes (de taille fixe)
  - inodes : intermédiaires (logiques et physiques) entre les répertoires, fichiers et les blocs
- Quelques chiffres :
  - Taille maximale du système de fichiers : 16 TB pour une taille de bloc de 4 ko (2TB avant le noyau 2.4)
  - Taille maximale d'un fichier : 2TB
  - Liste seulement pour le contenu des répertoires, lenteurs au-delà de 10000 fichiers / répertoire
- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : non
  - Modification possible de la taille : via le programme resize2fs (e2fsprogs)
  - Partition, gestion de ≪ sous-volume ≫ : aucune
  - Quotas : gestion par le système
  - Compression : possible via un module complémentaire
  - Déduplication : non
  - Cache / Tiering : système éventuellement
- Gestion de l'intégrité, de la sécurité, possibilité de conserver des états antérieurs / différents.
  - Journalisation non
  - Copy On Write : non, pas de snaphshots, ni clones non plus
  - Vérification de l'intégrité : fsck (e2fsck), seulement sur un système de fichier non-monté
  - Chiffrement : non

ext3

- Développement depuis 1999, disponible dans le noyau à partir de 2001. Compatible avec les systèmes ext2
- Stockage des données sur le disque :
  - découpage en blocs (de 1k à 8k), regroupés en groupes
  - Index HTree pour les répertoires
- Quelques chiffres
  - Taille maximale du système de fichiers : 16 TB pour une taille de bloc de 4 ko
  - Taille maximale d'un fichier : 2TB
- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : non
  - Modification possible de la taille (réduction, agrandissement) : via le programme resize2fs (e2fsprogs)
  - Partition, gestion de ≪ sous-volume ≫, thin-provisionning : non
  - Quotas : gestion par le système
  - Compression : via un module supplémentaire
  - Déduplication : non
  - Cache / Tiering : non
- Gestion de l'intégrité, de la sécurité, gestion de snapshots, clones
  - Journalisation : fonctionnalité ajoutée, pour les données, ou seulement les méta-données
  - Copy On Write : non
  - Vérification de l'intégrité : fsck, sur un système de fichier non-monté
  - Chiffrement : non

ext4

- Développement depuis 2006, disponible dans le noyau à partir de 2008. Rétro-compatible avec les systèmes ext3
- Stockage des données sur le disque :
  - découpage en blocs (de 1k à 8k), regroupés en domaines (extents) qui sont des blocs contigües (128MiB pour une taille de bloc de 4k)
  - utilisation d'inodes : intermédiaires entre les fichiers et les blocs
  - delayed allocation (pour réduire la fragmentation)
- Défragmentation en ligne
- Quelques chiffres
  - Taille maximale du système de fichiers : 1EiB
  - Taille maximale d'un fichier : 16TB
  - Index HTree pour les répertoires
- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : non
  - Modification possible de la taille (réduction, agrandissement) : via le programme resize2fs (e2fsprogs)
  - Partition, gestion de ≪ sous-volume ≫, thin-provisionning : non
  - Quotas : gestion par le système
  - Compression : via un module supplémentaire
  - Déduplication : non
  - Cache / Tiering : non

ext4

- Gestion de l'intégrité, de la sécurité, gestion de snapshots, clones
  - Journalisation : oui, plus un checksum sur le journal
  - Copy On Write : non
  - Vérification de l'intégrité : fsck, sur un système de fichier non-monté
  - Chiffrement : à partir du noyau 4.1 en Juin 2015

btrfs

- Développement depuis 2007. Inclus dans le noyau linux en 2009. Depuis 2012, considéré comme stable pour certaines distributions
- En 2017, fin du support RedHat
- Stockage des données sur le disque :
  - B-Tree. Utilisation de sommes de contrôle pour les méta-données et les données
  - Extents (regroupement de blocs)
- Défragmentation : online, depuis 2011
- Quelques chiffres
  - Taille maximale du système de fichiers : 16EiB
  - Taille maximale d'un fichier : 16EiB
- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : Raid0, 1 et 10. Autres niveaux en cours de développement. Ajout possibles de disques à chaud.
  - Modification possible de la taille online
  - Gestion de sous-volume possibles, montage à des emplacements différents sur le système de fichiers
  - Quotas : gestion par sous-volume, hiérarchique (pas par utilisateur)
  - Compression possible par zlib, lzo
  - Déduplication possible
  - Cache / Tiering : non

btrfs

- Gestion de l'intégrité, de la sécurité, gestion de snapshots, clones
  - Journalisation : non
  - Système de fichiers Copy On Write.
  - Permet de gérer des snapshots, et, des send/receive différentiels, clones de fichiers
  - Vérification de l'intégrité : fsck, plusieurs autres méthodes. Scrub des disques
  - Chiffrement : non

xfs

- Développement par SGI en 1993, pour les systèmes IRIX. Porté (sauf le gestionnaire de volumes) dans le noyau linux en 2001, disponible actuellement dans toutes les distributions
- Stockage des données sur le disque :
  - Système 64-bit, B-tree, allocation par extents, delayed allocation possible, taille de blocs fixe de 512o à 64ko
  - Attributs étendus (root ou user), sauvegardés par xfsdump/xfsrestore
  - Utilisation de write barriers
- Défragmentation : online
- Quelques chiffres
  - Taille maximale du système de fichiers : 8EiB
  - Taille maximale d'un fichier : 8EiB

xfs

- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : le système est adossé à un gestionnaire de volumes. Qui n'a pas été porté sous Linux
  - Modification possible de la taille : augmentation possible online, réduction via dump/restore
  - Gestion de sous-volume possibles : Non
  - Quotas : Option disponible par défaut
  - Compression : non, pas nativement
  - Déduplication : Non
  - Cache / Tiering : non
- Gestion de l'intégrité, de la sécurité, gestion de snapshots, clones
  - Journalisation : logique des méta-données. Possibilité de placer le journal sur un système de fichier différent
  - Système de fichiers Copy On Write : Non
  - Snapshots disponibles : pas sous linux, renvoyés au gestionnaire de volume
  - Vérification de l'intégrité : fsck. Pas de sommes de contrôle contre la corruption silencieuse des données
  - Chiffrement : à partir du noyau 4.1 en Juin 2015

jfs

- Développé par IBM depuis 1990 Version 1, 1999 pour la V2. Disponibles sur systèmes AIX, OS/2, porté sous linux (sous licence GPL) depuis 2001.
- Stockage des données sur le disque :
  - Système 64-bit, B-tree, allocation par extents
  - Efficace pour certaines bases de données
- Défragmentation : online
- Quelques chiffres
  - Taille maximale du système de fichiers : 32 PB
  - Taille maximale d'un fichier : 4PB

jfs

- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : Non
  - Modification possible de la taille du système de fichier : augmentation possible online
  - Gestion de sous-volume possibles : Non
  - Quotas : disponibles
  - Compression : Non (seulement sur JFS1 sous AIX)
  - Déduplication : Non
  - Cache / Tiering : non
- Gestion de l'intégrité, de la sécurité, gestion de snapshots, clones
  - Journalisation : logique des méta-données seulement. Déport sur un autre périphérique possible
  - Système de fichiers Copy On Write : Non
  - Snapshots disponibles : pas sous linux, renvoyés au gestionnaire de volume. Nécessité d'utiliser un mécanisme de lock
  - Vérification de l'intégrité : fsck. Pas de sommes de contrôle contre la corruption silencieuse des données
  - Chiffrement : Non

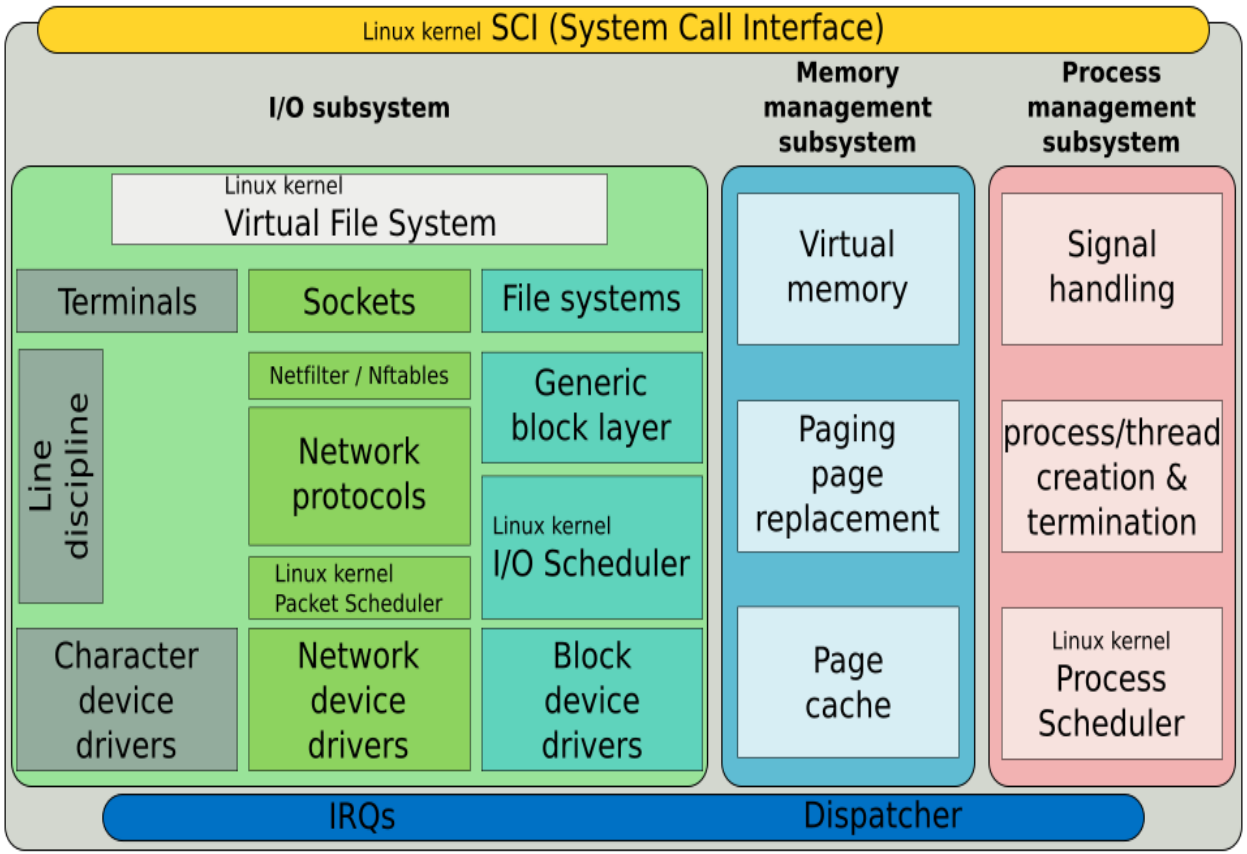
UFS

- Unix File System
- Aussi appelé FFS (Fast File System)
- Développement initial depuis 1979... Voir https://www.usenix.org/system/files/login/articles/584-mckusick.pdf
- Différentes implémentations
- Stockage des données sur le disque :
  - Système 64-bit, Tables pour le contenu des répertoires
  - Augmentations successives de la taille des blocs,
  - Dispositif "Soft updates", plutôt qu'un journal
- Défragmentation : Non
- Quelques chiffres
  - Taille maximale du système de fichiers : 8ZiB (2^{73} octets)
  - Taille maximale d'un fichier : 8ZiB (2^{73} octets)
  - Nom : 255 octets

UFS

- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques : Non
  - Modification possible de la taille du système de fichier : augmentation possible online
  - Gestion de sous-volume possibles : Filesystem Stacking
  - Quotas : disponibles
  - Compression : Non
  - Déduplication : Non
  - Cache / Tiering : non
- Gestion de l'intégrité, de la sécurité, gestion de snapshots, clones
  - Journalisation : Non
  - Système de fichiers Copy On Write : Non
  - Snapshots disponibles depuis 1999
  - Vérification de l'intégrité : fsck (en arrière plan en cas d'utilisation de "Soft updates")
  - Chiffrement : Non

ZFS / OpenZFS


Linux kernel SCI (System Call Interface)
I/O subsystem
Memory management subsystem
Process management subsystem
Linux kernel
Virtual File System
Terminals
Sockets
File systems
Line discipline
Netfilter / Nftables
Network protocols
Linux kernel
Packet Scheduler
Generic block layer
Linux kernel
I/O Scheduler
Character device drivers
Network device drivers
Block device drivers
Virtual memory
Paging page replacement
Page cache
Signal handling
process/thread creation & termination
Linux kernel
Process Scheduler
IRQs
Dispatcher

ZFS / OpenZFS

- Zetabytte File System
- Développement initial en 2001, par Sun Microsystems pour Solaris, le code source est libéré avec OpenSolaris (2005), puis illumos (2010), première version stable pour linux en 2013, lancement d'OpenZFS en 2013
- licence CDDL (problème de compatibilité avec la license GNU)
- Stockage des données sur le disque :
  - Système 128-bit
  - intégration de différentes composantes : I/O Scheduler, Logical Volume Manager, jusqu'à des programmes (serveur NFS)
  - Basé sur des pools d'espaces (zpool), et met à disposition des datasets (systèmes de fichiers) ou des volumes (zvol, périphérique en mode bloc)
  - Utilisation systématique et hiérarchisée de sommes de contrôle
  - Taille de blocs dynamique (128ko par défaut)
- Défragmentation : ???
- Quelques chiffres
  - Taille maximale du système de fichiers : 2^{128} octets
  - Taille maximale d'un fichier : 16 EiB 2^{64} octets
  - Nombre de fichiers 2^{48} par répertoire, illimité par système de fichier
  - Nom : 255 caractères ASCII

ZFS / OpenZFS

- Gestion de l'espace disque et de la vitesse
  - Gestion de plusieurs disques intégrée et native. Plusieurs possibilité d'agrégation, du RAID0 au raidz3
  - Modification possible de la taille du système de fichier : augmentation possible online
  - Gestion de sous-volume possibles : très fine, paramétrable pour chaque branche, délégation possible de droits
  - Quotas : disponibles, au niveau utilisateur ou au niveau d'un sous-volume
  - Compression : Oui, plusieurs algorithmes disponibles
  - Déduplication : Oui, online (mais consomme beaucoup de ressources)
  - Cache : Oui. 2 niveaux de cache en lecture (ARC, en mémoire et L2ARC, sur disque), et un cache disponible en écriture (ZIL)
- Gestion de l'intégrité, de la sécurité, gestion de snapshots, clones
  - Journalisation : pas nécessaire
  - Système de fichiers Copy On Write : Oui
  - Utilisation systématique et hiérarchisée de sommes de contrôle (pour les méta-données, les données)
  - Snapshots : Oui, nombre illimité
  - Vérification de l'intégrité : scrub sur les volumes
  - Chiffrement : au niveau des volumes (datasets)

Conclusion

Evolution des systèmes de fichiers :

- Plus de sécurité : journalisation, gestion de snapshots, vérification de l'intégrité
- Nouvelles fonctionnalités : clones, chiffrement
- Suit l'évolution des supports et des capacités de traitement

Limites : capacités d'une machine

- Agglomération de disques
- Bande-passante (Bus, réseau)
- Calcul

En local, le système ZFS est un cran au-dessus (maturité, fonctionnalités disponibles), pour passer à l'échelle, systèmes distribués

Sources

- Un article technique et cependant très drôle, sur l'histoire de quelques systèmes d'exploitations : `https://arstechnica.com/gadgets/2008/03/past-present-future-file-systems/`
- Un rapport en anglais (ne pas s'arrêter aux premières pages) sur les mécanismes de cache sur les systèmes de fichiers Linux ext4 et XFS, et de tests `https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=158453`
- Un rapport qui compare les systèmes de fichiers CoW, BTRFS et ZFS en vue de l'implantation dans le système FenixOS (galaxie Tanenbaum, Minix, qui a lancé Linus) `http://sakisk.me/files/copy-on-write-based-file-systems.pdf`
- Une thèse à propos de l'intégrité de données dans les systèmes de fichiers, par rapport à la journalisation (Systèmes NTFS, ext3, xfs, jfs, reiserfs). Détaille les modes de journalisation (writeback, ordered, data), Explique la distinction entre Physical et Logical journaling. `http://pages.cs.wisc.edu/~vijayan/vijayan-thesis.pdf`
- Différents articles wikipedia, sur les différentes technologies.