

Sauvegarde

Clément Ghnassia
Mathieu Leroux
Edgar Reyes Rodriguez

23 mars 2012

Chapitre 1

Généralités

1.1 Introduction

En Informatique, une sauvegarde consiste à faire des copies de données qui pourront être utilisées après un évènement de pertes de données.

La sauvegarde a deux buts :

- La restauration de données après que celles ci aient été perdues ou endommagées. En effet, la perte de données est courant pour les utilisateurs. 67% d'entre eux ont déjà eu ce type d'expérience.
- La restauration de données plus anciennes. Ici, la sauvegarde peut s'apparenter à de l'archivage.

Le fait d'avoir une bonne stratégie de sauvegarde n'est qu'un élément parmi tant d'autres dans un plan de restauration après un désastre. En effet les administrateurs ont tendance à penser que faire des sauvegardes régulières est suffisant : il n'en est rien. Il faut penser à plein d'autres situations comme la perte totale du système et des applications, la destruction d'une base de données, ... pour envisager une reprise d'activité rapide.

Un système de sauvegarde doit prendre en compte la taille des données qu'il faut sauvegarder. En effet, même si cela dépend du type de données et de la compression plus ou moins efficace, il faut comprendre qu'une sauvegarde du système contiendra au moins une copie de toutes les données. Les capacités de stockages qui en découlent sont donc considérables, et c'est donc un facteur majeur dans le choix d'une stratégie de sauvegarde. Il existe plusieurs modèles pour la sauvegarde de données, chacun étant adapté aux différentes situations, comme la quantité de données à sauvegarder, la capacité de stockage pour les sauvegardes, la redondance des données dans la sauvegarde, la sécurité des données, ou encore la rapidité de la restauration.

Avant que les données soient réellement sauvegardées, elles sont sélectionnées, extraites et manipulées. Lorsque les données sont importantes, cela peut vite devenir fastidieux. Des techniques ont été mises au point pour optimiser ces procédures. Cela peut aller de la résolution de problèmes lors de sauvegardes à chaud, de la compression, du chiffrement ou de la déduplication des données. Il ne faudra pas non plus négliger les facteurs humains dans les processus de sauvegarde et de restauration, qui tendent tout de même à être de plus en plus automatisés (pour les sauvegardes du moins).

1.2 Le stockage

1.2.1 Les modèles d'entrepôts de données

La première étape dans l'élaboration d'une stratégie de sauvegarde est de trouver le ou les modèles qui correspondront le mieux avec la situation, les contraintes et les objectifs recherchés. Le choix d'un modèle est primordial et doit être mûrement réfléchi. Il définira comment les données seront stockées

et organisées. Cela aura donc un impact important en terme d'espace disque nécessaire, de sécurité, ou encore de vitesse et de simplicité lors d'une restauration

Sauvegarde non structurée

Même si cela semble la méthode la plus simple à mettre en oeuvre, elle n'est pas très efficace lors d'une perte de données et n'offre pas un très haut niveau de sécurité en terme de restauration de données. De plus, elle nécessite une organisation méthodique de l'administrateur. Cette méthode consiste simplement à stocker des sauvegardes avec des informations sur la nature de ces sauvegardes et les dates auxquelles elles ont été réalisées.

Images du système

Dans ce type de modèle, il s'agit de stocker une image complète du système à un instant donné. Une des applications courantes consiste à faire une image afin de copier à l'identique la configuration d'un système sur d'autres machines. Pas très performant, surtout en ce qui concerne l'optimisation de l'espace disque, il est de moins en moins utilisé pour la sauvegarde.

Sauvegarde différentielle

Cette méthode consiste à réaliser une sauvegarde complète des données souhaitées, puis d'enregistrer à intervalle régulier toutes les différences depuis la dernière sauvegarde complète. Cela permet de ne pas dupliquer les données qui n'ont pas changées et offre donc un gain d'espace considérable, dépendant bien évidemment du nombre de données qui peuvent changer d'une sauvegarde à l'autre. Pour restaurer ces données, on aura besoin de la dernière sauvegarde complète et éventuellement de la dernière sauvegarde avant la date du système souhaitée pour la restauration.

Sauvegarde incrémentale

Cette méthode repose sur le même principe qu'une sauvegarde différentielle, à la différence que lors d'une sauvegarde incrémentale, seul les données modifiées depuis la dernière sauvegarde, complètes ou non, seront enregistrées. L'avantage par rapport à une sauvegarde différentielle réside dans un gain d'espace disque, car il y'aura moins de données redondantes. En revanche, la restauration s'avèrera plus longue et plus fastidieuse étant donné qu'on aura besoin de rejouer toutes les sauvegardes depuis la dernière sauvegarde complète et éventuellement jusqu'à la dernière sauvegarde incrémentale à la date désirée.

Sauvegarde delta inversé

Ici, le fonctionnement est inversé. En effet, plutôt que de faire une sauvegarde complète et d'enregistrer les modifications par la suite comme c'est le cas avec les sauvegarde différentielles et incrémentales, on aura toujours une sauvegarde complète (un miroir) correspondant à l'état des données à l'instant de la dernière sauvegarde effectuée. Le fonctionnement est trivial : après avoir effectué une sauvegarde complète, chaque sauvegarde inversée aura pour effet une synchronisation du miroir avec l'état réel des données. En même temps, les modifications qui permettront de revenir à un état antérieur seront enregistrées. Ainsi, pour restaurer les données au dernier état, il suffira juste de restaurer le miroir, et une restauration à un état antérieur impliquera de rejouer les sauvegardes de manière décroissante dans le temps depuis le miroir jusqu'au moment souhaité.

Protection des données en continu (CDP)

Aussi appelé sauvegarde en temps réel, c'est la méthode la plus sûre en terme de protection des données. Au lieu d'effectuer des sauvegardes périodiquement, les modifications sont envoyées instantanément vers l'espace de stockage dédié à la sauvegarde à travers le réseau. Bien que faisant penser à de la réplication (pour de la haute disponibilité par exemple), cela est complètement différent car ce n'est pas une copie parfaite des données qui est réalisée, mais un système de logs et de journaux

qui enregistrent les modifications et qui permettront de revenir à un état antérieur dans le cas d'une éventuelle restauration.

Type	Taille	Vitesse de sauvegarde	Vitesse de restauration	Intégrité	Simplicité
Non structurée	N	N	O	N	O
Image système	N	N	O	N	O
Différentielle	O	O	N	O	N
Incrémentale	O	O	N	O	N
Delta inversée	O	O	O	O	N
CDP	O	O	O	O	N

TABLE 1.1 – Tableau récapitulatif sur les modèles d'entrepôts de données

1.2.2 Dispositifs de stockage

Aujourd'hui, les périphériques pour le stockage de données sont nombreux et variés. Il s'agira ici d'analyser chacun de ces dispositifs et d'en comprendre les avantages et les inconvénients, en particulier dans le cas d'une utilisation pour la sauvegarde.

Bandes magnétiques

Aujourd'hui encore, c'est sans doute le support le plus utilisé pour le stockage de données, plus particulièrement pour la sauvegarde et l'archivage. La raison est tout d'abord financière. En effet, même si la tendance est à une réduction des différences de prix entre les bandes magnétiques et les disques durs, c'est encore actuellement la solution la moins chère à capacité de stockage égale. Il faut noter que les bandes magnétiques sont en adéquation avec les contraintes de sauvegarde car elles fournissent un accès séquentiel, ce qui a un impact bénéfique sur les vitesses de lecture et d'écriture. De plus, on sait que les bandes magnétiques, si elles sont correctement conservées, assureront l'intégrité de leur contenu sur plusieurs décennies. Enfin, les bandes magnétiques sont utilisées depuis un certain temps, et leurs caractéristiques sont donc totalement maîtrisées. En revanche, le problème survient au niveau des formats. En effet, la plupart du temps les formats de bandes magnétiques sont nombreux et souvent propriétaires. Utiliser des bandes magnétiques engendrera donc une dépendance forte vis à vis des constructeurs, et ajoute des contraintes en cas de changement de solution. De plus, pour les grosses structures, l'investissement sera d'autant plus important que l'achat d'un automate sera nécessaire.

Disques durs

Ce moyen est de plus en plus utilisé pour la sauvegarde et l'archivage, mais dans bien des cas, les solutions à base de bandes magnétiques lui sont toujours préférées, et ce à juste titre. A première vue, l'utilisation de disques durs pour la sauvegarde et l'archivage de données semblerait intéressante. En effet, le prix des disques durs se rapproche des bandes magnétiques à capacité équivalente. De plus, les disques durs ont des avantages certains par rapport aux bandes magnétiques en terme de temps d'accès, de disponibilité, de capacité, et de simplicité d'utilisation. Ils ne nécessitent pas d'équipement spécifique, et permettent une utilisation dans des contextes différents, s'adaptant plus aux besoins, comme par exemple leur emplacement, aussi bien au niveau local que sur le réseau. On pourra aussi ajouter des procédés permettant de réduire les tailles des sauvegardes, comme des techniques de dé-duplication. Mais l'utilisation de ce type de support dans un contexte de sauvegarde, et surtout d'archivage possède un inconvénient de taille. Tout d'abord la fragilité du support n'est pas adaptée à ce type d'utilisation et n'assure donc pas l'intégrité des données stockées. En effet là où la manipulation et l'externalisation des sauvegardes est courante, la fragilité des disques durs est un point critique. De plus, il n'a pas été établi sur combien de temps les données pourraient être conservées sur le long terme sans risque d'altération de celles-ci. Si cette contrainte n'est pas insurmontable pour des sauvegardes où la conservation de données sur le long terme en vue d'une restauration n'aurait pas beaucoup de sens, elle l'est

en revanche beaucoup plus pour l'archivage, qui nécessite souvent une conservation des données à long terme.

Supports optiques

Si ces supports sont courants et très utilisés par les particuliers pour stocker des données, de par leur prix et leur simplicité d'utilisation, il ne semble pas que ce soit un support adapté dans le cas de sauvegarde et d'archivage de quantités importante de données, comme c'est souvent le cas dans une entreprise. En effet ces support sont généralement d'une faible capacité de stockage et sont souvent de type WORM (écriture mais pas effacement). Toutefois, il existe des solutions qui peuvent correspondre à des petites structures ne nécessitant pas forcément de grande capacité de stockage. Des systèmes de chargeurs permettent une automatisation, ce qui aura pour effet d'éviter le contact des disques avec l'être humain, négatif pour l'intégrité des données.

Disquette

Les disquettes, très à la mode dans les années 1980, n'est évoqué ici que pour des raisons historiques. Très limité en terme de capacité de stockage, ce support est aujourd'hui et depuis un certain temps totalement obsolète

SSD

Ce type de support à base de mémoire flash ont pour particularité d'être extrêmement rapide, aussi bien en lecture, qu'en écriture. Ces éléments ne possèdent pas le défaut de fragilité des disques durs. En revanche le ratio prix /capacité est encore extrêmement élevé, et les capacités sont plus limitées que les disques durs classiques, ce qui rend leur utilisation comme support pour la sauvegarde ou l'archivage impossible.

Services de sauvegardes à distance

Avec la popularisation des accès internet à haut débit, c'est une méthode en vogue et qui se développe de plus en plus. Ces services permettent d'effectuer des sauvegardes en ligne, ce qui semble être la solution la plus fiable en terme d'intégrité des données, puisque ces sauvegardes peuvent être stockées n'importe où dans le monde. En revanche, cela oblige de faire confiance à un tiers puisqu'il aura accès à toutes les données. Même en chiffrant ces données avant de les envoyer, il faudra que ce service soit sûr. En effet les conséquences peuvent être dramatiques si un problème survient chez le tiers alors qu'on doit effectuer une restauration. De plus, les particuliers ont en général un faible débit montant, ce qui est contraignant dans le cas de sauvegarde sur une grande quantité de données.

Type	Taille	Prix	Intégrité	Vitesse	Simplicité
Bandes magnétiques	O	O	O	O	N
Disques durs	O	O	N	O	O
Supports optiques	N	O	N	O	N
SSD	N	N	O	O	O
Sauvegardes à distance	O	N	O	N	N

TABLE 1.2 – Tableau récapitulatif sur les dispositifs de stockages

1.2.3 Gestion du dépôt

Il existe aussi plusieurs types de gestion de dépôt. En effet il faudra choisir une méthode qui correspond aux besoins de sauvegardes en termes d'accessibilité, de sécurité et de coûts. Ces méthodes ne sont pas exclusives et pourront être combinées pour arriver à une gestion des sauvegardes et de leurs dépôts plus fine et plus élaborée.

En ligne

Le support de stockage est en permanence connecté. Cela lui donne l'avantage d'être tout le temps accessible et permet d'effectuer une restauration avec le moins de manipulations. Ce système de stockage peut se mettre en place à l'aide de disques internes, ou comme c'est le cas dans les grandes structures, à l'aide de baies SAN. En plus d'être coûteuse, la gestion des dépôts en ligne induit des problèmes de sécurité. En effet, les données étant accessibles en permanence, elles peuvent facilement être altérées, de manière intentionnelle, ou suite à une erreur de manipulation.

Intermédiaire

Moins chers, mais aussi moins accessibles que les systèmes en lignes, ces solutions dites intermédiaires semblent un bon compromis entre accessibilité, sécurité des données et coûts. Cela permet aussi de rester assez réactif en cas de restauration. Un bon exemple serait l'utilisation d'une bibliothèque de bandes magnétiques automatisée, qui tout en offrant une solution déconnectée, permet d'effectuer une restauration rapidement.

Déconnecté

Sauf lorsque les sauvegardes et les restaurations sont réalisées, les sauvegardes sont complètement isolées du point de vue informatique et nécessite une intervention manuelle. Dans le cas de bandes magnétiques, il s'agira d'une insertion dans un lecteur. Dans le cas d'un disque dur, il s'agira de le connecter à un ordinateur par exemple. Cette solution est donc orientée sur la sécurité des données et l'accessibilité dépendra de l'emplacement des sauvegardes (typiquement sur site ou hors site)

1.2.4 Externalisation

Le 11 Septembre l'a montré, le fait d'avoir des sauvegardes n'empêche pas la perte irrémédiable de données. Pour se prémunir contre des désastres de cette ampleur tel que des feux ou des tremblements de terre, il faudra opter pour un stockage d'une partie ou de toutes les sauvegardes hors site. Ces endroits où seront stockées ces données peuvent être plus ou moins évolués, d'un simple bureau chez l'administrateur, jusqu'aux chambres fortes. Dans ce cas, la mise en place d'une restauration peut s'avérer plus longue, et pose donc des problèmes d'accessibilité.

Site de sauvegarde

Si lors d'un événement particulièrement désastreux, les équipements informatiques sont hors d'état de fonctionner, le fait d'avoir des sauvegardes à jour ne permettra pas une restauration, et l'indisponibilité du système sera alors inévitable. Dans le cas de cette éventualité, une solution particulièrement coûteuse consiste de fournir à un autre site des équipements informatiques nécessaires pour prendre le relais. Ainsi, en alimentant le second site avec des sauvegardes à jour provenant du premier site, il pourra fonctionner indépendamment et de manière autonome, jusqu'à un rétablissement.

Type	Accessibilité	Coûts	Sécurité	Vitesse de restauration	Simplicité
En ligne	O	N	N	O	O
Intermédiaire	O	O	O	O	N
Déconnecté	N	O	O	N	N
Externalisation	N	N	O	N	N
Sites de sauvegardes	O	N	O	O	N

TABLE 1.3 – Tableau récapitulatif sur les gestions de dépôt

1.3 Les données

Lors de la conception de stratégies de sauvegardes, une étape importante consiste à établir quelles seront les données qui sont critiques et qui auront besoin d'être sauvegardées. Il faudra bien évidemment considérer les fichiers importants pour l'utilisateur, mais aussi ceux qui sont nécessaires au bon fonctionnement de l'ordinateur. Cette étape est vitale lors de la conception de stratégies de sauvegardes. En effet, choisir de sauvegarder un nombre trop important de fichiers aurait pour conséquences un remplissage de l'espace de stockage et empêcherait l'application de la stratégie de sauvegardes à court terme. Au contraire, ne pas sauvegarder assez de données aurait, comme on peut s'en douter, des conséquences désastreuses lors d'une perte de données.

1.3.1 Fichiers

Aujourd'hui, toute donnée informatique est structurée en fichiers.

Sauvegarde de fichiers

C'est évidemment ce qui sera copié dans le cas de sauvegardes classiques.

Sauvegarde de parties de fichiers

Lorsqu'on modifie un fichier, une sauvegarde de type différentielle ou incrémentale aura pour effet d'inclure toute le fichier dans la nouvelle sauvegarde, même si celui-ci n'a été que peu modifié. Afin de résoudre ce problème de redondance de données, une solution consiste à sauvegarder uniquement les blocs qui ont été modifiés plutôt que tout le fichier. Les procédés de compression et de dé-duplication sont une solution à ce problème.

1.3.2 Systèmes de fichiers

Il peut être judicieux dans certains cas de sauvegarder des éléments relatifs au système de fichiers.

Sauvegarde complète

Plutôt que de copier l'ensemble des fichiers contenus dans le système, une solution consiste à sauvegarder le système de fichiers dans son ensemble. Cette technique s'apparente à la réalisation d'images disque. En copiant tout le système de fichiers à l'aide d'outils de bas niveau (comme dd sous les systèmes UNIX), ce type de sauvegarde est souvent plus rapide, surtout quand les données sont composées de petits fichiers fragmentés. En revanche, la réalisation de ce type de sauvegardes ne peut pas généralement se faire à chaud, c'est à dire que le système sera indisponible durant toute la durée de la sauvegarde. De plus, les avantages de ce type de sauvegarde sur un système de fichier peu rempli seront vite effacés car un outil de bas niveau copiera tous les blocs du système de fichiers, sans faire la distinction entre des blocs contenant des données et des blocs non indexés.

Identificateurs de changement

Certains systèmes de fichiers évolués permettent à chaque fichier d'avoir un bit d'archive qui permettra de savoir si des changements ont été opérés sur le fichier depuis la dernière sauvegarde.

Versionnage du système de fichiers

Lorsque elle est mise en place sur un système de fichiers, une version de chaque fichier modifié est automatiquement conservé. Ainsi, l'utilisateur qui le souhaite peut restaurer un fichier à un instant donné, sans avoir à effectuer aucune manipulation au préalable.

1.3.3 Données en temps réel

Lorsque la sauvegarde est effectuée à chaud, c'est à dire que la sauvegarde a lieu alors que le système est utilisé normalement, il se peut que la sauvegarde de fichiers ouverts et en cours de modification pose des problèmes. En effet, comme c'est généralement le cas lors de sauvegardes de bases de données, les données ne reflètent pas leur état à un instant t puisqu'elles ont été modifiées entre le début et la fin de la sauvegarde. Si des données sont liées, elles peuvent ne plus être cohérentes entre elles, et donc pas exploitables.

Sauvegarde par snapshots

Disponible sur certains systèmes de stockage, cette fonction permet de faire une copie du système de fichier à un instant donné. Lors de la prise d'un snapshot, les activités du système sont très brièvement gelées, et on a donc plus de risque d'incohérence des données.

Sauvegarde de fichier ouvert

Certains logiciels permettent la sauvegarde de fichiers ouverts simplement en vérifiant si les fichiers sont utilisés, ou utilisent des verrous.

Sauvegarde à froid

Ce terme est généralement utilisé pour les bases de données. Lors de ce type de sauvegardes, la base de données devient indisponible le temps de la sauvegarde, et ceci pour éviter toute modification de fichier qui risquerait d'entraîner des incohérences.

Sauvegarde à chaud

Cette fois, ce type de sauvegarde, généralement sur des bases de données, permet d'effectuer des sauvegardes sans pour autant rendre la base indisponible. Un image des données est d'abord réalisée à chaud, pendant que les modifications sont stockées dans des journaux. A la fin de la réalisation de l'image initiale, les journaux sont rejoués pour avoir une copie des données conforme à la date de restauration.

1.3.4 Méta-données

Au-delà des fichiers, il faudra aussi penser à sauvegarder des informations relatives au système.

Secteur de boot

Indispensable pour le démarrage du système, il peut être utile dans certains cas de le sauvegarder.

Schémas de partitions

La aussi, il peut être utile sauvegarder ces informations qui pourraient être utiles lors d'une restauration rapide du système.

Méta-données des fichiers

Ces méta-données stockent, entre autre, les permissions, les ACL et les propriétaires de chaque fichier.

Méta-données du système

Ce sont des informations nécessaires pour le système.

1.4 Optimisation

Beaucoup de méthodes d'optimisation pour la sauvegarde et l'archivage sont utilisées, et ont pour rôle d'améliorer les performances de ce processus, comme par exemple les vitesses de sauvegardes, les vitesses de restauration, ou encore l'intégrité et la sécurité des données.

1.4.1 Compression

Ce mécanisme permet d'influer sur l'espace qu'occuperont les sauvegardes sur l'espace de stockage. En effet, il existe de nombreux algorithmes et de procédés plus ou moins efficaces et dépendant du type de données ciblées qui permettront d'alléger les données en taille et donc d'optimiser l'espace disque.

1.4.2 Dé-duplication

Tandis que la compression n'est ciblée que sur un fichier, voir un groupe de fichiers, la dé-duplication est elle gérée par le support de stockage lui-même. En analysant les blocs déjà présents sur le support, il optimisera l'espace de stockage en préférant indexer ces blocs plutôt que de les réécrire, ce qui aura pour conséquences une redondance nulle et une optimisation de l'espace disque maximale.

1.4.3 Duplication

Les stratégies de sauvegardes ont parfois recourt à de la duplication afin de manipuler et d'optimiser ces données dans le cas d'une éventuelle restauration, ou tout simplement pour stocker les données dans deux endroits différents dans le but d'améliorer la sécurité.

1.4.4 Chiffrement

Comme les données sauvegardées sont souvent des données importantes, voir critiques, elles sont souvent confidentielles. La perte ou le vol de sauvegardes peut alors être lourd de conséquences. En chiffrant les sauvegardes, cela permettra de sécuriser les données. Toutefois cela a un coût : Le chiffrement est un processus qui requiert du CPU et qui augmentera le temps des sauvegardes.

1.4.5 Refactoring

Ce processus consiste à réorganiser les données sauvegardées dans les dépôts de données. Cela est surtout utile pour les stratégies de sauvegarde basées essentiellement sur des sauvegardes incrémentales.

1.4.6 Staging

Cela consiste à faire transiter les sauvegardes par un autre support avant que celles-ci n'atteignent l'espace où elles seront stockées.

1.5 Gestion des processus de sauvegarde

Il faut comprendre que la sauvegarde est un processus. Les sauvegardes sont indispensable dans tout système informatisé. Même si les structures sont différentes en terme de grandeur, les objectifs et les contraintes restent les mêmes.

1.5.1 Objectifs

Objectifs de points de restauration

Cela correspondra au point dans le temps auquel le système sera restauré. Idéalement, il faudrait avoir une sauvegarde correspondant à la date juste avant la perte de données. Plus le délai entre chaque sauvegarde sera court, plus on pourra se rapprocher de la date à laquelle les données ont été altérées. Multiplier les sauvegardes a bien évidemment un coût, et il faudra donc ajuster la stratégie en fonction de la structure et des moyens mis à disposition.

Objectifs de temps de restauration

Lors d'un désastre, la durée nécessaire à la restauration sera un élément capital, car il définira le temps pendant lequel le système sera indisponible.

Protection des données

La protection des données est aussi un élément majeur dans les stratégies de sauvegarde. En effet, il serait paradoxal d'avoir un système ultra-protégé et de laisser les sauvegardes, qui contiennent les mêmes données, accessibles à tous. Le chiffrement des sauvegarde permettra, entre autre, de protéger ces données sensibles.

1.5.2 Limites

Pour la création d'un schéma de sauvegarde, il faudra prendre en compte les limites de l'environnement.

Fenêtre de sauvegarde

On appelle fenêtre de sauvegarde la durée pendant laquelle la sauvegarde est possible, en relation avec l'utilisation plus ou moins importante du système. Bien évidemment, il faudra prendre en compte cette durée dans la mise en place du schéma de sauvegarde.

Performances

Les sauvegardes sont souvent gourmandes en CPU, et ont donc un impact sur les performances du système. C'est un élément qu'il faudra aussi prendre en compte dans l'élaboration d'un plan de sauvegarde.

Investissements

Les stratégies de sauvegardes nécessitent plus ou moins d'espace disque, ce qui aura un certain coût au final. Lors du choix de la stratégie de sauvegarde adoptée, l'espace de stockage nécessaire aux sauvegardes devra être calculé, ceci afin de connaître les besoins, et donc les coûts.

Limites Réseau

Les limitations du réseau par lequel transiteront les données dans le cas de systèmes de sauvegarde distribués sont autant de contraintes à ne pas négliger.

1.5.3 Réalisation

Concevoir la stratégie de sauvegarde tout en essayant d'atteindre au mieux les objectifs fixés et en prenant en compte les contraintes et les limites établies peut s'avérer fastidieux.

Plannification

L'utilisation d'un planificateur de tâches plutôt que de dépendre de l'exécution des tâches manuellement apporte un vrai plus et assure que les sauvegardes seront bien réalisées en temps et en heure. La plupart des logiciels de sauvegardes incluent cette fonctionnalité.

Authentification

Afin d'assurer la protection et la confidentialité des données sauvegardées, un processus d'authentification doit être intégré.

Chaîne de confiance

Les opérations sensibles liées à la sauvegarde tel que la manipulation des supports de sauvegardes ne doit être réalisée que par des personnes dignes de confiance et est un élément important dans la sécurisation des données.

1.5.4 Vérification et supervision

Les sauvegardes sont des données très importantes pour une entreprise. S'assurer que les données sont correctement sauvegardées et avoir des systèmes de vérification, d'alerte et de journaux d'activités est très important.

Validation de la sauvegarde

C'est le processus par lequel les propriétaires des données peuvent avoir des informations sur les sauvegardes concernant leur données.

Rapports

Il est courant qu'un rapport concernant le déroulement de la sauvegarde soit affiché ou envoyé à la personne concernée. Ce rapport contiendra des informations précieuses sur le processus de sauvegarde et permettra de savoir si la sauvegarde s'est bien déroulée.

Journaux

Un système de journaux d'activités qui permettra la création d'historiques sur les sauvegardes est aussi un élément essentiel pour s'assurer du bon fonctionnement des sauvegardes.

Validation

La validation est une étape tout aussi importante qui consiste à vérifier l'intégrité des données. En effet, en effectuant des checksums et hachages sur les fichiers sauvegardés et en les comparant avec les originaux, on pourra aisément s'assurer de l'intégrité de leur contenu. De plus, cela permettra de savoir si les fichiers ont été modifiés depuis la dernière sauvegarde et optimisera ces processus en ne copiant que les données nécessaires.

Supervision

Il est possible de superviser les processus de sauvegardes par un centre de supervision. Ce centre permettra d'alerter les utilisateurs sur les erreurs qui peuvent se produire sur les sauvegardes automatisées.

1.6 Loi

1.6.1 Confusion

A cause du chevauchement de nombreuses technologies, la confusion est souvent faite entre sauvegardes, archives et systèmes à tolérance de panne. Les archives sont des données qui sont stockées car elles ne sont plus utilisées, mais qui doivent être conservées pour des raisons légales et historiques. La sauvegarde consiste en la copie de données actuellement utilisées, et qui seront conservées dans le cas d'une perte ou altération de ces données. Quant aux systèmes à tolérance de panne, ils n'assurent pas l'intégrité des données en cas de panne, mais assure justement qu'il n'y aura pas de panne. On peut donc dire que ces trois technologies, même si elles sont proches techniquement, n'ont pas la même finalité et interviennent à des niveaux différents.

1.6.2 Conseils

- Plus les données sont importantes, plus le besoin de sauvegarde est important.
- Les sauvegardes sont seulement utiles si elles sont correctement intégrées dans une stratégie de sauvegarde.
- Stocker les sauvegardes à côté des données originales n'est pas conseillé.
- Les stratégies de sauvegardés basées sur l'automatisation et la planification sont plus sûres qu'une intervention humaine (l'erreur est humaine).
- Les sauvegardes font intervenir un grand nombre de facteurs externes. C'est donc une opération qui peut échouer pour des raisons diverses et variées. Mettre en place un système de supervision est important.
- Les sauvegardes multiples sur différents médias et stockées à différents endroits est indispensable pour les données les plus critiques.
- Il est préférable que les sauvegardes soient stockées dans un format libre et standard, surtout pour les archives qui sont susceptibles d'être conservées sur le long terme.
- S'assurer du bon fonctionnement des processus de sauvegardes est une tâche primordiale et la non-vérification de celles-ci peut être lourde de conséquences.
- Si cela est possible et envisageable au niveau de l'investissement, il est conseillé de stocker les sauvegardes sur deux supports différents (par exemple bandes magnétiques et disques durs)

1.6.3 Évènements

- En 1996, un feu au siège social du crédit lyonnais a entraîné la perte de nombreuses données car les bandes magnétiques étaient stockées sur site.
- Il a été rapporté que 16 événements de vols de bandes magnétiques appartenant à des organisations majeures ont été recensés au cours des années 2005 et 2006.
- le 3 Janvier 2008, un serveur de mail appartenant à TeliaSonera, une entreprise de telecoms et fournisseur d'accès nordique, est tombé. Il a été découvert au cours de cet événement que la dernière sauvegarde exploitable datait du 15 décembre 2007. Trois cent mille comptes mails ont été irrémédiablement affectés.
- Le 27 Février 2011, un bug applicatif de GMail a eu pour conséquences la perte de l'ensemble des mails pour 0.02% des utilisateurs. Une restauration à partir de bandes magnétiques a permis de récupérer l'ensemble des données perdues.

1.7 Exemple de Mise en place d'une stratégie de sauvegarde

Dans cette partie, nous prendrons l'exemple d'une entreprise qui souhaite adopter une nouvelle stratégie de sauvegarde, plus efficace et répondant aux besoins et aux contraintes de l'entreprise. La mise en place de cette nouvelle stratégie de sauvegarde sera la suivante : après une brève analyse des besoins, et objectifs de l'entreprise, nous verrons quelles sont les différentes contraintes qui peuvent intervenir. Nous nous attarderons ensuite longuement sur le choix d'une stratégie à adopter et les choix qui ont été faits seront justifiés.

1.7.1 Besoins et objectifs

Dans cette partie, il sera défini quels sont les objectifs et les besoins auxquels la stratégie de sauvegarde devra répondre le plus parfaitement possible.

Données à sauvegarder

La première question à se poser est : que faut-il sauvegarder ? Cela permettra de définir l'espace de stockage nécessaire aux données (en relation avec la durée de rétention et les méthodes de sauvegarde). Dans notre exemple, l'entreprise souhaite sauvegarder des fichiers, accessibles depuis un serveur, pour une quantité d'environ 100 Go.

Durée de rétention

Combien de temps faut-il conserver les sauvegardes ? Ceci peut dépendre de l'entreprise, comme de la loi. Dans notre exemple, on supposera que l'archivage est déjà réalisé indépendamment de la sauvegarde, puisque cela ne concerne qu'une petite partie des données à sauvegarder. Il serait souhaitable d'avoir une durée de rétention des sauvegarde de 12 mois, ceci afin de se prémunir contre toute éventualité.

Périodicité des sauvegardes

Combien de temps entre chaque sauvegarde ? Il faut bien comprendre que cela aura un impact direct sur la quantité des données perdues : la durée entre deux sauvegardes correspondra donc à la durée maximale pour laquelle il peut y'avoir une perte de données. Par exemple, en effectuant une sauvegarde toutes les 24 heures, on pourra perdre 24 heures de données au maximum (en admettant que les sauvegardes n'aient pas été détruites dans l'évènement qui a conduit à la perte des données). Dans notre exemple, on considèrera qu'une sauvegarde devra être réalisée toutes les 6 heures au minimum.

Vitesse de restauration

Dans le cas d'une perte de données, quel est le temps maximum d'indisponibilité ? En d'autres termes, combien de temps la restauration des données peut prendre au maximum ? C'est une question à laquelle on ne peut répondre exactement. Toutefois on considèrera que ce temps d'indisponibilité est critique et qu'il devra être réduit au maximum ;

Criticité des données

La question est de savoir si les données sont critiques pour l'entreprise. En général, lorsque l'on souhaite la mise en place d'une stratégie de sauvegarde évoluée (comme c'est le cas ici), la réponse est plus qu'évidente. Toutefois il existe un niveau dans la criticité des données, et qui définira donc un niveau d'investissement. Toutes les données ne se valent pas et on devra donc essayer de définir ce niveau de criticité. Dans notre exemple les données seront définies comme très critiques pour l'entreprise.

Confidentialité des données

Faut-il sécuriser les données pour qu'elles ne soient pas accessibles en cas de vol de sauvegarde ? La question est de savoir si les données sauvegardées devront être chiffrées. On considèrera ici que les données ne sont pas trop sensibles. Les mesures de sécurité élémentaires seront prises, comme l'emplacement de stockage des sauvegardes dans un endroit qui n'est pas accessible à n'importe qui.

Automatisation des sauvegardes

Les sauvegardes devront être lancées manuellement ou réalisées automatiquement aux heures définies ? Dans la plupart des cas, une automatisation est nécessaire. Mieux vaut se fier à un programme informatique qu'à une personne, qui de plus est n'est pas présente tout le temps sur le site. D'ailleurs, dans notre cas où une sauvegarde a lieu toutes les 6 heures, l'automatisation est inévitable.

Responsabilité

A qui donne-t-on la responsabilité des sauvegardes ? Dans le cas d'une automatisation, on pourrait penser que cette responsabilité est fortement réduite : il n'en est rien. Au delà de la mise en place des stratégies de sauvegarde et la réalisation de celles-ci, un nombre important de tâches devront être réalisées, comme tester les sauvegardes, ou encore gérer le stockage et l'externalisation.

Délégation

Lorsqu'un utilisateur souhaite récupérer un fichier qu'il a malencontreusement effacé, il peut être utile de trouver un moyen pour qu'il réalise ces opérations de restauration par lui-même. S'il n'est pas possible de l'intégrer directement à notre stratégie de sauvegarde, ce dispositif, généralement différent, peut être mis en parallèle et donc être indépendant de la stratégie de sauvegarde principale.

Investissement

Pour choisir la stratégie qui sera appliquée, il faudra bien évidemment tenir compte du budget que l'entreprise pourra allouer. Il paraît évident que cet aspect sera un point important, la plupart des entreprises ne pouvant pas se permettre un investissement illimité dans le domaine, d'autant que le Retour sur Investissements (ROI), bien qu'indéniable lors d'événements au cours desquels des données critiques sont perdues ou altérées, est difficile à calculer. En d'autres termes, quel que soit le secteur d'activité de l'entreprise, on n'aura aucune rentabilité directe et immédiate. On comprend donc les entreprises à rechigner dans des investissements pour mettre en place une stratégie de sauvegarde, se limitant au strict minimum. Dans notre exemple, on considèrera que les différentes personnes concernées dans la hiérarchie de l'entreprise ont bien compris l'enjeu et l'importance d'une bonne stratégie de sauvegarde. On aura donc un budget de départ plutôt conséquent, mais il faudra que le quotidien soit viable économiquement. Autrement dit, le budget alloué au fonctionnement ne devra pas être trop élevé.

1.7.2 Stratégie adoptée

Type de sauvegarde

Pour respecter la périodicité de sauvegardes tout en économisant un maximum d'espace, l'idéal est de combiner les trois types de sauvegardes principaux, à savoir réaliser des sauvegardes complètes, des sauvegardes différentielles et des sauvegardes incrémentales. Etant donné la périodicité des sauvegardes particulièrement élevée, et l'espace de stockage à préserver au maximum, on optera pour la configuration suivante :

- Une sauvegarde complète tous les débuts de mois est effectuée
- Une sauvegarde complète est réalisée en parallèle chaque semaine (le dimanche par exemple)
- En dehors du jour de la sauvegarde hebdomadaire complète, une sauvegarde différentielle est réalisée chaque jour de la semaine
- Enfin, pour compléter la stratégie, des sauvegardes incrémentales sont réalisées 4 fois par jour

Support de sauvegarde

Le support de sauvegarde adopté sera celui qui est encore le plus couramment utilisé dans le cadre de stratégies de sauvegardes professionnelles : les bandes magnétiques. En effet, même si un lecteur de bande, voir un robot est un investissement considérable, le coût des bandes magnétiques est lui, très bon marché, compte tenu de l'espace qu'ils offrent.

Cycle et permutation

Compte tenu de la durée de rétention et les types de sauvegardes réalisées, le schéma de permutation choisi sera du type Grand-père / Père / Fils. La sauvegarde mensuelle complète est immédiatement entreposée hors site tandis que la sauvegarde hebdomadaire complète est conservée sur site le temps d'utilisation, c'est à dire toute la semaine. Cette sauvegarde sera ensuite conservée hors site pendant un nombre déterminé de cycles hebdomadaires, généralement de deux à trois semaines. Les sauvegardes différentielles seront elles aussi stockées sur site durant la semaine où la sauvegarde hebdomadaire complète associée est présente, puis seront stockées hors sites parallèlement à la sauvegarde complète à laquelle elle est associée. Enfin les sauvegardes incrémentales seront conservées sur site en permanence, et ceci pour une durée d'une semaine seulement, puis les bandes sur lesquelles elles sont stockées seront réutilisées.

Capacité nécessaire

Il est souvent difficile d'estimer l'espace de stockage nécessaire pour la mise en place de la stratégie de sauvegarde. Pourtant si on ne fait pas l'effort réaliser cette estimation, on risque de voir être dépassé par la quantité des sauvegardes. L'espace de stockage prend en compte beaucoup de facteurs, qui sont plus ou moins calculables et prévisibles. Les facteurs facilement quantifiables sont la quantité de données sauvegardées, en admettant qu'elle reste toujours identique, la périodicité des sauvegardes, les types de sauvegardes réalisés (complètes, différentielles ou incrémentales), bien évidemment la durée de rétention, et l'optimisation réalisée, comme par exemple la compression de ces données. Dans le cadre de sauvegardes différentielles ou incrémentales, le calcul est plus complexe : leur taille dépend de la quantité de données modifiées depuis la dernière sauvegarde complète/différentielle. Ainsi plus les données seront modifiées, plus les sauvegardes qui leur sont associées nécessiteront un espace de stockage important : La formule suivante permettra de calculer à peu près la quantité nécessaire pour une stratégie de sauvegarde.

$$D \cdot R1 + (D \cdot T1\% \cdot N1 \cdot R2) + (D \cdot T2\% \cdot N2 \cdot R3)$$

Avec :

- D : Quantité de données sauvegardées
- T1 : Taux de modification des données entre une sauvegarde complète et une sauvegarde différentielle
- T2 : Taux de modification des données entre deux sauvegardes incrémentales
- N1 : Nombre de sauvegardes différentielles réalisées par semaine
- N2 : Nombre de sauvegardes incrémentales réalisées par semaine
- R1 : Durée de rétention des sauvegardes complètes, exprimée en semaines
- R2 : Durée de rétention des sauvegardes différentielles, exprimée en semaines
- R3 : Durée de rétention des sauvegardes incrémentales, exprimée en semaines

On notera bien que ce calcul n'est pas exact mais indiquera une approximation concernant l'espace nécessaire à la mise en place d'une stratégie de sauvegarde. Dans notre exemple, on arrive à un peu moins de 1800 GO, ce qui est peut sembler conséquent étant donnée la quantité totale de données sauvegardées, mais qui est du à la périodicité particulièrement élevée et surtout la rétention des sauvegardes mensuelles complètes (les sauvegardes complètes mobilisant à chaque fois un espace égal à la quantité des données à sauvegarder). De plus, on ne prend pas en compte la compression, mais celle-ci est aisément calculable : il suffira seulement de définir le pourcentage de compression obtenue sur les données cibles.

Emplacement de stockage

Cet aspect, souvent négligé dans les entreprises, est pourtant un élément essentiel. En effet, les responsables des sauvegardes s'appliquent à concevoir une stratégie de sauvegarde évoluée, mais oublient parfois que les supports sont, comme tout élément informatique, fragiles et doivent être protégés. On devra donc bien réfléchir à l'emplacement physique de ces sauvegardes, aussi bien quand elles sont stockées sur site que lors d'une externalisation. Pour les sauvegardes stockées sur site, elles devront être éloignées un maximum du serveur sur lequel sont stockées les données cibles. Ainsi, la destruction physique du serveur n'entraînera pas la destruction des données.

1.7.3 Recommandations

On ne le rappellera jamais assez : définir une stratégie de sauvegarde n'est pas suffisant. La stratégie de sauvegarde n'est en réalité qu'un élément parmi beaucoup d'autres, en ce qui concerne le bon fonctionnement de l'entreprise, comme par exemple la disponibilité des données et la marche à suivre en cas d'incident. On mettra l'accent sur la définition d'un plan de reprise d'activité (PRA) qui indiquera justement les procédures à effectuer suivant les différents problèmes qui peuvent intervenir. Afin

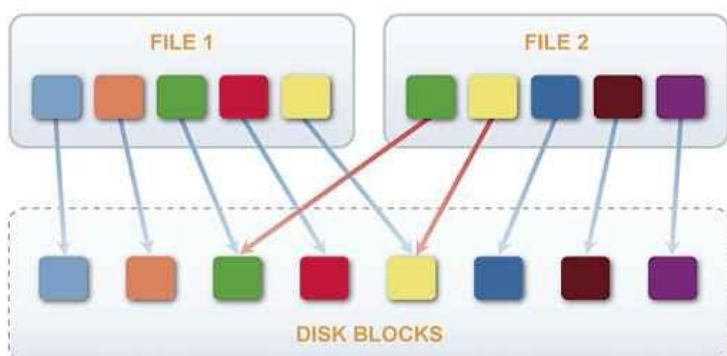
d'augmenter la productivité de l'entreprise, on pourra mettre en place (en parallèle à la sauvegarde principale) un système interne pour la délégation de tâches simples de sauvegarde et de restauration aux utilisateurs finaux.

Chapitre 2

Optimisation

La déduplication de données est une technique qui permet de minimiser de l'espace de stockage. Elle consiste à ne pas répliquer les données déjà existantes sur le disque. Un fichier est décomposé sous forme de blocs de données car des fichiers peuvent avoir des blocs en commun. Le mécanisme de déduplication crée une table avec les index de tous les blocs de données des fichiers présents sur le disque. La taille des blocs peut varier selon les mécanismes utilisés mais plus les blocs sont petits, plus il y aura de chance qu'un autre bloc soit identique et donc, plus la déduplication sera efficace. En général, cette taille ne dépasse pas les 128ko.

Quand un utilisateur dépose un fichier, le mécanisme crée ses index et regarde s'il n'y a pas des blocs déjà existants. Si des blocs sont similaires alors une simple référence aux blocs déjà existants sera créée. Le schéma ci-dessous montre comment la déduplication fonctionne. Les blocs étant de la même couleur sont considérés identiques.



Il existe deux types de déduplication : la déduplication à la volée (à la source) et la déduplication hors ligne (à la destination). La déduplication à la volée analyse les fichiers avant de les stocker pour savoir s'ils n'existent pas déjà sur le disque. Cette technique utilise une forte consommation CPU et mémoire. L'autre technique consiste à copier dans un premier temps le fichier sur le disque avant de tester s'il existe déjà. Cela nécessite de prévoir un espace de stockage tampon plus important.

Dans un contexte de serveur de messagerie et de fichiers centralisés, la déduplication de données peut très rapidement économiser de nombreux gigaoctets d'espace disque ainsi que la diminution de la bande passante qui aurait été utilisée pour la sauvegarde. En effet, dans le cas où un même mail de 1Mo est envoyé à cinquante destinataires alors l'économie du disque sera de 50-1 megaoctets (stockage d'un seul mail). La déduplication est faite pour des fichiers tels que des documents bureautiques ou des machines virtuelles qui ont souvent de nombreux blocs en commun.

Le terme inverse de la déduplication est la réhydratation. Elle fait appel à la table des index afin de renvoyer tous les blocs de données référencés pour un fichier demandé.

Certain outils comme LessFS permettent de dédupliquer et de compresser les blocs de données. Cela permet de gagner encore plus d'octets sur le disque mais nécessite une consommation mémoire et CPU plus importante.

2.1 ZFS

2.1.1 Introduction

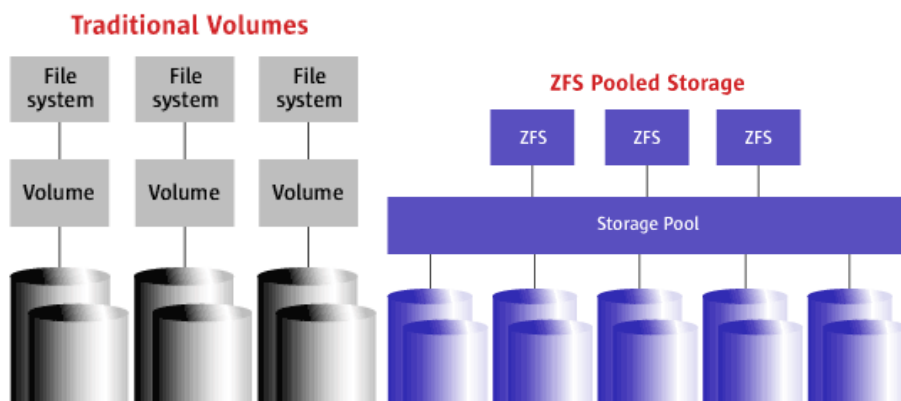
Le système de fichier ZFS (Zettabyte File System) a été conçu par Sun en 2005 et est sous licence CDDL. Il n'était disponible que sous Solaris mais est devenu récemment disponible sous linux. Il est l'un des systèmes de fichiers les plus intéressants du marché. En effet, ZFS intègre de nombreux avantages que d'autres n'ont pas. Voici une liste de ses principaux avantages :

- Pas de limites pratiques (taille des disques, fichiers, ...)
- Garantir la sécurité des données (intégrité, disponibilité)
- Administration simplifiée
- Gestionnaire de volume intégré
- Compression
- Snapshot
- Duplication
- Quotas et réservation d'espace
- Performances élevées
- Indépendant de l'architecture matérielle

ZFS est un système de fichier 128 bits contrairement aux autres systèmes qui sont de 64 bits. Ainsi ses limites sont de 16 milliards de milliards fois plus autant dire qu'il n'a quasi pas de limite. Afin d'optimiser ses performances, ZFS utilise tout l'espace disponible de la RAM pour créer un énorme cache. Ce procédé s'appelle ARC (Adaptive replacement cache). Il peut poser problème aux autres processus qui testent la mémoire inutilisée avant de ce lancer mais cette mémoire est souvent inutilisée. Il peut être partagé via le réseau avec d'autres systèmes de fichiers comme nfs ou samba. Ainsi même depuis des systèmes qui ne le supporte pas, il sera accessible.

2.1.2 Stockage

ZFS fonctionne avec un pool. C'est un ensemble de périphériques qui fournissent de l'espace pour le stockage et la duplication des données comme le raid logiciel ou matériel. Traditionnellement, les systèmes de fichiers classiques étaient restreint à un périphérique par système. Avec la gestion des volumes, il est possible de créer plusieurs systèmes de fichiers sur un périphérique.



Voici les différentes unités de base de stockage de données :

- Disques : entiers ou juste une partition
- Fichiers dans un autre système de fichiers
- Miroirs : 2 (ou plus) disques, partitions ou fichiers
- Raid-z : plusieurs disques, variante de RAID-5

ZMirror est un miroir classique. Il utilise les mécanismes de checksum pour valider les lectures sur un composant et bascule sur le second s'il détecte une erreur puis corrige le composant défaillant (si possible). Le système Raid-z est similaire au procédé Raid 5. Il utilise les checksums (SHA-256 + fletcher) et repose sur le copy on write : supprime le "write-hole".

2.1.3 Garantir la sécurité des données (intégrité, disponibilité)

Avec ZFS, toutes les données et métadonnées sont vérifiées selon un algorithme de somme de contrôle. Lorsque qu'un bloc de données endommagé est détecté, ZFS récupère les données correctes à partir d'une autre copie redondante et répare les données endommagées en les remplaçant par celles de la copie.

2.1.4 Snapshots

Un snapshot (ou instantané) est une copie en lecture seul d'un système de fichier ou d'un volume. ZFS permet donc de pouvoir sauvegarder et restaurer l'image du volume désiré. La création d'un instantané est quasi immédiate. Les instantanés utilisent l'espace de stockage du pool. Une seule opération, dite atomique, permet de créer des instantanés récursifs au système désiré. Ils ne sont pas directement accessibles mais peuvent être clonés, sauvegardés ou restaurés. D'une manière simple et rapide, un instantané peut être créé, restauré ou supprimé :

#creation du snapshot nommé "nomSnapshot" du système de fichier systèmeZFS au sein du pool "pool"

```
zfs snapshot pool/systemeZFS@nomSnapshot
```

#Opération atomique (récursif)

```
zfs snapshot pool/systemeZFS@nomSnapshot
```

#Instantané "nomSnapshot" supprimé

```
zfs destroy pool/systemeZFS@nomSnapshot
```

#Restauration de l'instantané "nomSnapshot"

```
zfs rollback pool/systemeZFS@nomSnapshot
```

#Permet de voir les différences entre les deux instantanés

```
zfs diff pool/systemeZFS@nomSnapshot pool/systemeZFS@nomSnapshot2
```

2.1.5 Clones

Un clone est un volume ou un système de fichiers accessible en écriture dont le contenu initial est celui de l'instantané qu'il a créé. En effet, les clones ne sont créés que par des instantanés et une dépendance se crée entre les deux. Néanmoins, les clones n'héritent pas des propriétés de leur instantané mais peuvent être modifiés via les commandes `zfs set` et `zfs get`. Les commandes ci-dessous permettent de créer un snapshot et de l'utiliser pour créer un clone.

creation du snapshot

```
zfs snapshot pool/systemeZFS@nomSnapshot
```

creation du système de fichier /home à l'aide du snapshot

```
zfs clone pool/systemeZFS@nomSnapshot pool/home
```

2.1.6 Compression

La compression est une option de ZFS. Elle peut être activée pour chaque système de fichiers et snapshots via l'option `compression`. Ses valeurs sont soit `on`, `off`, `lzjb` (algorithme tiré de Lempel ziv), `gzip` et `gzip-n`. Son taux de compression sera d'environ 2 suivant le type de données à compresser et l'option choisie. Exemples :

pour le système de fichier

```
zfs set compression=on pool/systemeZFS
```

pour le snapshot

```
zfs set compression=on pool/systemeZFS@nomSnapshot
```

2.1.7 Quotas et reservation d'espace

ZFS permet de limiter la taille de stockage à un système de fichier via l'option "quota" et permet de réserver un espace à un système de fichier via l'option "reservation". Ces propriétés sont très intéressantes quand il s'agit de limiter de l'espace disque à des utilisateurs. ZFS préconise de créer un système de fichiers par utilisateur qui serait monté au sein du même pool. Il est donc possible de créer des cotas par utilisateurs et par groupes via les options `zfs userquotas` et `zfs groupquotas`. Il sera possible de lister l'espace utilisé par utilisateur ou par groupe. Voici quelques exemple d'utilisations :

```
#attribution d'un quota de 10 gigaoctets au système de fichier mat
```

```
zfs set quota=10G pool/home/mat
```

```
#attribution d'un quota de 10 gigaoctets à l'utilisateur mat
```

```
zfs set userquota@mat=10G pool/home
```

```
#attribution de 100 gigaoctets au groupe etudiant
```

```
zfs set groupquota@etudiant=100G pool/etudiant
```

2.2 Compression

Tout comme la déduplication, la compression est une technique qui permet d'économiser de l'espace de stockage. Chaque fichier est constitué d'une succession de millions de bits 0 ou 1. La compression permet de diminuer le nombre de bits que constitue un fichier en changeant la succession de bits de départ. Suivant l'algorithme de codage utilisé, le taux de compression peut différer. Les algorithmes d'encodage sont plus ou moins efficaces selon le type de fichier compressé.

Il existe deux types de compression : la compression avec pertes et sans perte. La compression sans perte signifie qu'après la décompression, le fichier sera identique au fichier compressé. C'est le plus souvent utilisé sur des documents, des fichiers exécutables ou des archives. Ces données étant principalement des caractères texte, ils ne peuvent pas être modifiés. Les formats de documentation tels que txt, doc ou pdf sont donc compressés sans perte. Tant qu'à la compression avec perte, les fichiers décompressés ne seront pas exactement identiques au fichier original mais les informations seront sensiblement les mêmes. Les types de fichiers utilisés par cette compression sont les images, les sons et les vidéos. Cette technique se repose sur la limitation des sens de l'homme comme la vision et l'audition. L'homme ne pourra donc pas identifier les différences entre le fichier original et le fichier après décompression. Les formats de fichiers jpeg, avi ou mp3 sont donc compressés avec pertes.

Pour chaque technique de compression, il existe plusieurs algorithmes de codage.

2.2.1 Compression sans perte

Parmi les algorithmes sans perte, il y a les algorithmes tels que Lempel-Ziv ou le codage RLE (Run-Length Encoding) qui consistent à remplacer des suites de bits utilisées plusieurs fois dans un même fichier. D'autres algorithmes comme l'algorithme de codage Huffman détermine les suites de bits et plus une suite est utilisée souvent, plus la suite qui la remplacera sera courte.

L'algorithme Lempel-Ziv

Cet algorithme se divise en deux versions distinctes : LZ77 et LZ78. Ces algorithmes utilisent un dictionnaire où ils référencent les motifs récurrents. A la rencontre d'un motif du dictionnaire, une simple référence au motif est faite (fenêtre glissante). La déduplication utilise globalement le même procédé.

LZ77 La compression LZ77 encode avec un taux de compression inférieur à d'autres algorithmes comme PPM et CM (voir ci-dessous) mais a le double avantage d'être rapide et asymétrique. Cela lui permet d'utiliser un algorithme de décompression différent de celui de la compression. Ainsi, la compression pourra être rapide et la décompression performante. Les variantes LZSS et LZMA sont basées sur la compression LZ77 et supprime quelques inconvénients de celle-ci tels que le taux de compression assez faible (pour LZMA) ou le problème si aucun motif récurrent n'est rencontré (pour LZSS). Ce problème aura pour conséquence d'augmenter la taille du fichier. La compression LZ77 est la base des algorithmes comme Deflate (ZIP, gzip) et donc LZMA (7-zip).

LZ78 La compression LZ78 ou Lempel-Ziv-Welch utilise aussi un dictionnaire mais au lieu de le remplir au fur et à mesure des motifs rencontrés, il crée un dictionnaire initial de tous les symboles possibles. Cela permet d'améliorer la compression car les données du dictionnaire ne devant plus être envoyées au décompresseur, l'espace utilisé est réduit. L'utilisation de cette technique a été réduite jusqu'en 2003 car elle a été brevetée par UNISYS qui n'avait pas laissé la licence libre.

LZO Lempel-Ziv-Oberhumer (LZO) est un algorithme de compression en temps réel se basant sur les dictionnaires. Ces avantages sont une compression et décompression rapide. L'un des logiciels l'utilisant est lzop.

L'algorithme RLE

Le run-length encoding (codage par plages) est une technique de compression qui s'applique uniquement sur des documents scannés en noir et blanc tels que des fax. Elle consiste à factoriser les termes d'une même couleur. Ainsi la chaîne : NNNNNNBBBBNNNNNNNNNNBB (N étant le nombre de points noirs et B étant le nombre de points blancs) sera encodée par RLE en : 7N4B10N2B. Les formats d'images utilisent cette compression en considérant que toutes les lignes de pixels sont jointes pour former une unique séquence de couleur. Les images BMP utilisent cette compression en 1,4 et 8 bits/pixel (noir et blanc, 16 couleurs et 256 couleurs). Le format PCX utilise aussi cette compression pour les images de 8 et 24 bits/pixels. Celles de 24 bits étant découpées en trois parties de 8 bits chacune.

Codage par modélisation de contexte

Prédiction par reconnaissance partielle (PPM) La prédiction par reconnaissance partielle se base sur une modélisation de contexte pour évaluer la probabilité des différents symboles. Le contexte est un ensemble de symboles déjà rencontrés dans la source de données. Elle utilise les données déjà analysées pour en déduire les données à analyser. Ainsi plus le contexte est long, meilleur sera la prédiction et donc la compression. La prédiction obtenue servira d'entrée à un codage entropique comme le codage Huffman. Elle a l'avantage d'être l'une des plus performantes sur la compression de fichiers texte mais a l'inconvénient de consommer énormément de mémoire si le contexte est très grand. La PPM est un algorithme symétrique contrairement à Lempel-Ziv ce qui signifie qu'il utilise le même pour la compression que pour la décompression. Cela implique un temps d'exécution identique et assez lent.

Pondération de contextes (CM) La pondération de contextes consiste à utiliser plusieurs prédicteurs (par exemple des PPM) pour obtenir l'estimation la plus fiable possible du symbole à venir. A l'image de la prédiction par reconnaissance partielle, les taux de compressions sont très élevés mais proportionnellement aussi lents que la taille du contexte.

L'algorithme de codage Huffman

Cette compression s'apparente à la compression du code morse. Elle consiste donc à coder les séquences fréquentes sur peu de place et ce qui revient rarement sur des séquences plus longues. L'inconvénient avec ce procédé c'est qu'il faut avoir analysé tout le fichier pour créer une table avec les

redondances avant de pouvoir le compresser. Il faut donc envoyer la table pour pouvoir le décompresser ce qui peut être problématique quand le fichier à compresser est petit. Le codage Huffman adaptatif corrige ce problème car il remplit au fur et à mesure la table et démarre la compression avec une table de base.

Ce codage est utilisé en seconde compression après que le premier algorithme (tel que LZ77) est mis en évidence la redondance d'information. Ce codage peut être utilisé pour la compression tels que JPEG, MPEG ou MP3 où les données imperceptibles par l'homme sont supprimées mais on parle donc de compression avec pertes.

2.2.2 Compression avec pertes

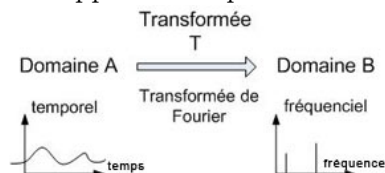
La compression avec pertes s'utilisent donc sur des données perceptibles par l'homme comme les sons, les images ou les vidéos. Elles suppriment les données que l'homme ne perçoit pas ou quasiment pas. Ainsi pour le format JPEG 2000, la compression est de 1 bit/pixels au lieu de 24 bits/pixels. La compression avec pertes est une technique irréversible c'est à dire qu'il ne sera pas possible de retrouver le fichier original. Il existe trois grandes familles de compression avec pertes : la compression par prédiction, par transformation et la compression basée sur les récurrences fractales de motif.

Compression par prédiction

Cet algorithme repose sur un schéma de prédiction et un codage des erreurs entre la prédiction et le signal original. La prédiction consiste à prédire les données à venir en fonction des données analysées. Les erreurs étant souvent de faibles magnitudes, une compression intéressante est possible grâce à la diminution des bits nécessaires à l'opération. Certain formats de codecs de microsoft et d'Apple utilisent cette compression.

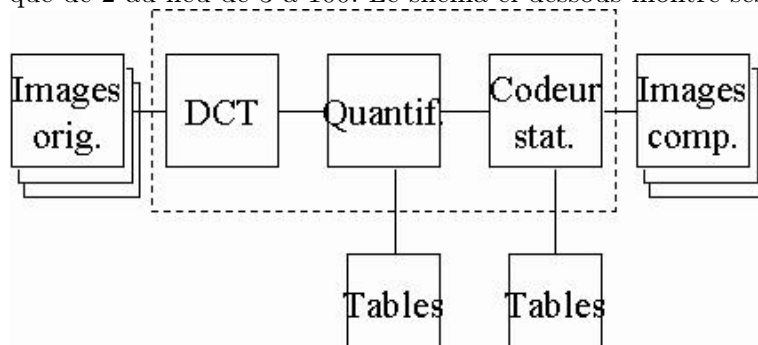
Compression par transformation

Cet algorithme transforme le signal en atténuant les fréquences non décelables par l'homme. Cette technique transforme donc le signal du domaine temporel au domaine fréquentiel afin de déterminer et de supprimer les pixels redondants. Le schéma ci-dessous montre cette transformation.



La compression JPEG, JPEG 2000 ou encore MPEG utilise cette compression. Cette méthode de compression est la plus répandue au vue de ces performances.

La norme JPEG La norme JPEG (Joint Photographic Experts Group) est une norme qui définit le format d'enregistrement et l'algorithme de décodage pour une représentation numérique compressée d'une image fixe. La norme JPEG peut être compressée sans perte mais son taux de compression n'est que de 2 au lieu de 3 à 100. Le schéma ci-dessous montre ses étapes de compression et décompression.



Tout d'abord, le format JPEG commence par découper l'image en blocs de données comme beaucoup d'autres formats compressés avec pertes. Puis JPEG transforme les couleurs de chaque bloc à l'aide

de la transformée DCT (transformée en cosinus discrète) assimilable à la transformée de Fourier qui transforme le signal temporel en signal fréquentiel (DCT). La valeur des fréquences résultantes détermineront leurs importances dans l'image. Une matrice de ces résultats sera générée. La quantification est l'étape qui permet de réduire considérablement la taille de l'image. En effet, elle utilise la DCT pour atténuer les fréquences non perceptibles par l'homme. La matrice résultante par la quantification sera ensuite codée par un algorithme RLE puis par un algorithme d'Huffman afin d'être compressée. Lors de la quantification et du codage, des tables sont créées et envoyées avec le fichier compressé pour la décompression.

Lors de la compression du format JPEG sans perte, l'étape de la quantification n'est pas présente.

Compression par ondelette La compression (ou transformée) par ondelette s'utilise globalement comme la norme JPEG mais génère une image de meilleure qualité avec un taux de compression supérieure (de 15 à 50). Contrairement à la transformée DCT, l'image est analysée plus finement et a un résultat plus proche de la perception humaine. Les codeurs JPEG 2000 et SPIHT utilisent tous deux une transformée en ondelettes dans leur schéma de compression. Les domaines d'utilisation de cette compression est l'imagerie médicale, les empreintes digitales ou encore dans le cinéma.

Compression basée sur les récurrences fractales de motif

La compression basée sur les récurrences fractales de motif aussi appelée compression fractale est utilisée pour la compression d'image. Son principe est de détecter les récurrences de motifs et de supprimer les informations redondantes de l'image. Plusieurs méthodes existent mais la plus connue est la méthode Jacquin. Deux étapes composent cette méthode. Dans un premier temps, deux segmentations sont réalisées : une segmentation de figure source et destination. Ensuite pour chaque figure Source, une figure destination est cherchée afin de créer un couple pour minimiser une erreur. Cette erreur est le résultat de leur soustraction après avoir dimensionné le couple de manière identique. A ce stade, des transformations comme la rotation peuvent être réalisées.

2.3 Mise en place d'un serveur de fichiers ZFS + NFS

Nous allons mettre en place un serveur de fichiers sur un système de fichiers ZFS pour les nombreux avantages cité ci-dessus. Il ne sera pas possible d'y ajouter des programmes de déduplication comme lessFS et openDedup car lessFS intègre un système de fichiers et Opendedup fonctionne avec SDFS. Néanmoins, le système de fichier intègre au même titre que la compression une propriété permettant la déduplication (présente uniquement sur Solaris). Il sera donc question de tester les principales fonctionnalités de ZFS comme la compression, l'intégrité des données (via raid-z), les sauvegardes (via les snapshots) et leurs restaurations. Je créerai ainsi 25 espaces utilisateurs de 1Go chacun où je stockerai divers types de fichiers comme des documents (.odt, .txt, .pdf), des images (jpeg, gif, png) et une base de données mysql. Afin de partager les systèmes de fichiers dédiés aux utilisateurs, ZFS permet d'activer le partage en nfs avec l'option sharenf à on.

2.3.1 Création des systèmes de fichiers

Nous allons créer un pool de stockage de 60Go en raidz2 qui est équivalent au raid 6. Pour ce faire, on utilisera sept fichiers de 10Go que nous allons créer par la commande ci-dessous :

```
for i in `seq 0 6`;
do dd if=/dev/zero of=./disque_$i bs=1M count=10000;
done;
```

Parmi les sept fichiers utilisés, un sera utilisé pour les redondances d'informations du au raidz2. Ensuite, nous allons créer un pool de stockage nommé poolAsrall à l'aide de ces fichiers où seront installer les systèmes de fichiers. Le pool sera installé dans /media/. Voici la commande :

```
zpool create poolAsrall raidz2 /media/Windows7/disque_{0,1,2,3,4,5,6} -m /me
```

Puis on va créer un système de fichier /home/ au sein de ce pool et les cinquantes systèmes de fichiers dédiés aux utilisateurs.

```
zfs create poolAsrall/home
for i in `seq 0 49`;
do zfs create poolAsrall/home/utile_$i;
done;
```

Afin de tester la compression de ses systèmes de fichiers, nous allons activer l'option qui sera récursive à tout ses sous systèmes de fichiers. Nous allons tester l'option on, off, lzjb, gzip, gzip-1, gzip-9 pour tout les espaces utilisateurs.

```
zfs set compression=on poolAsrall/home
```

Puis on va les remplir de 1,42Go de fichiers divers et variés. Pour cela, nous avons copié le contenu de notre dossier Documents qui fait cette taille et qui contient :

- 40 pourcents de fichiers textes (sh, php, html, txt,..)
- 30 pourcents d'images (png, gif, jpeg)
- 15 pourcents de pdf
- 15 pourcents d'autres types de fichiers

Par le script suivant, nous allons remplir ces espaces utilisateurs.

```
for i in `seq 0 24`;
do cp -R /home/mathieu/Documents poolAsrall/home/utile_$i;
done;
```

Concernant la base de données mysql, il faut changer le repertoire de stockage des données dans le fichier /etc/mysql/my.conf puis copier le dossier /var/lib/mysql et le coller dans le dossier où nous voulons stocker les données. L'essentiel est de garder les droits du dossier original.

Dans notre test, nous allons créer un système de fichier bin puis mysql où allons stocker les données de la base.

```
zfs create poolAsrall/bin
zfs create poolAsrall/bin/mysql
cp -a /var/lib/mysql /media/poolAsrall/bin/mysql
cd /media/poolAsrall/bin/mysql && tar -xvf mysql.tgz
```

2.3.2 Etat du système

Nous allons visualiser l'état du système mise en place. Pour commencer, nous allons regarder l'état du pool.

```
zpool status
```

```
pool: poolAsrall
```

```
state: ONLINE
```

```
scrub: none requested
```

```
config:
```

NAME	STATE	READ	WRITE	CKSUM
poolAsrall	ONLINE	0	0	0
raidz2	ONLINE	0	0	0
/media/Windows7/disque_0	ONLINE	0	0	0
/media/Windows7/disque_1	ONLINE	0	0	0
/media/Windows7/disque_2	ONLINE	0	0	0
/media/Windows7/disque_3	ONLINE	0	0	0
/media/Windows7/disque_4	ONLINE	0	0	0
/media/Windows7/disque_5	ONLINE	0	0	0


```
/media/Windows7/disque_6  ONLINE          0          0          0
```

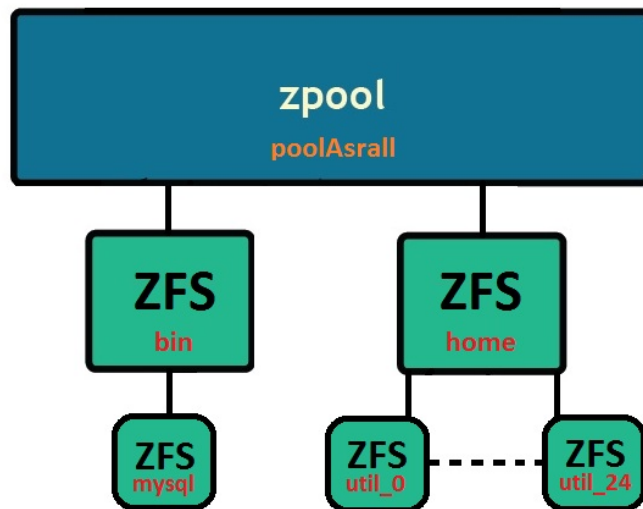
```
errors: No known data errors
```

Les disques créés étant situés dans ma partition dédiée à Windows. Par la commande ci-dessous, on peut visualiser l'espace utilisé et libre du pool. Comme aucun quota ou reservation n'a été précisé, les systèmes de fichiers créés à l'intérieur du pool utiliseront tous son espace jusqu'à saturation.

```
zpool list
```

NAME	SIZE	USED	AVAIL	CAP	HEALTH	ALTROOT
poolAsrall	68G	48,8G	19,2G	71%	ONLINE	—

Schéma de l'installation



Nous allons visualiser l'espace utilisé par chaque système de fichiers.

```
df -h
```

Sys de fichiers	Tail.	Occ.	Disp.	Occ.	Monte sur
/dev/sda7	54G	34G	18G	66%	/
none	1,5G	332K	1,5G	1%	/dev
none	1,5G	116K	1,5G	1%	/dev/shm
none	1,5G	244K	1,5G	1%	/var/run
none	1,5G	0	1,5G	0%	/var/lock
none	1,5G	0	1,5G	0%	/lib/init/rw
none	54G	34G	18G	66%	/var/lib/ureadahead/debugfs
/dev/sda2	187G	149G	39G	80%	/media/Windows7
/dev/sda5	1,1G	149M	892M	15%	/media/b07f0352-66e0-4503-ad13-9c
poolAsrall	13G	49K	13G	1%	/media/poolAsrall
poolAsrall/home	13G	102K	13G	1%	/media/poolAsrall/home
poolAsrall/bin	13G	22M	13G	1%	/media/poolAsrall/bin
poolAsrall/bin/mysql	13G	22M	13G	1%	/media/poolAsrall/bin/mysql
poolAsrall/home/util_0	15G	1,4G	13G	10%	/media/poolAsrall/home/util_0
poolAsrall/home/util_1	15G	1,4G	13G	10%	/media/poolAsrall/home/util_1
poolAsrall/home/util_2	15G	1,4G	13G	10%	/media/poolAsrall/home/util_2

```

poolAsrall/home/util_3
    15G  1,4G   13G  10% /media/poolAsrall/home/util_3
poolAsrall/home/util_4
    15G  1,5G   13G  10% /media/poolAsrall/home/util_4
poolAsrall/home/util_5
    15G  1,5G   13G  10% /media/poolAsrall/home/util_5
poolAsrall/home/util_6
    15G  1,5G   13G  10% /media/poolAsrall/home/util_6
poolAsrall/home/util_7
    15G  1,5G   13G  10% /media/poolAsrall/home/util_7
poolAsrall/home/util_8
    15G  1,5G   13G  10% /media/poolAsrall/home/util_8
poolAsrall/home/util_9
    15G  1,4G   13G  10% /media/poolAsrall/home/util_9
poolAsrall/home/util_10
    15G  1,4G   13G  10% /media/poolAsrall/home/util_10
poolAsrall/home/util_11
    15G  1,4G   13G  10% /media/poolAsrall/home/util_11
poolAsrall/home/util_12
    15G  1,4G   13G  10% /media/poolAsrall/home/util_12
poolAsrall/home/util_13
    15G  1,4G   13G  10% /media/poolAsrall/home/util_13
poolAsrall/home/util_14
    15G  1,4G   13G  10% /media/poolAsrall/home/util_14
poolAsrall/home/util_15
    15G  1,4G   13G  10% /media/poolAsrall/home/util_15
poolAsrall/home/util_16
    15G  1,4G   13G  10% /media/poolAsrall/home/util_16
poolAsrall/home/util_17
    15G  1,4G   13G  10% /media/poolAsrall/home/util_17
poolAsrall/home/util_18
    15G  1,4G   13G  10% /media/poolAsrall/home/util_18
poolAsrall/home/util_19
    15G  1,4G   13G  10% /media/poolAsrall/home/util_19
poolAsrall/home/util_20
    15G  1,4G   13G  10% /media/poolAsrall/home/util_20
poolAsrall/home/util_21
    15G  1,4G   13G  10% /media/poolAsrall/home/util_21
poolAsrall/home/util_22
    15G  1,4G   13G  10% /media/poolAsrall/home/util_22
poolAsrall/home/util_23
    15G  1,4G   13G  10% /media/poolAsrall/home/util_23
poolAsrall/home/util_24
    15G  1,4G   13G  10% /media/poolAsrall/home/util_24
/dev/sda3
    125G  116G   8,5G  94% /media/Data

```

Compression

Les tests sur les différentes compressions ont été fait de cette manière :

- poolAsrall/home/util_0 - 3 -> compression On : 1,39Go
- poolAsrall/home/util_4 - 8 -> compression Off : 1,42Go
- poolAsrall/home/util_9 - 13 -> compression lzjb : 1,40Go
- poolAsrall/home/util_14 - 18 -> compression gzip : 1,36Go
- poolAsrall/home/util_19 - 21 -> compression gzip-1 : 1,37Go

– poolAsrall/home/util_22 - 24 -> compression gzip-9 : 1,36Go

Nous pouvons constater que la compression à la volée du système de fichiers ZFS permet de gagner environ quatre pourcents grâce à l'algorithme de compression gzip. En effet, contrairement à l'algorithme lzjb qui est tiré de Lempel ziv, gzip compresse grâce à Lempel ziv et Deflate. Le taux de compression peut paraître assez faible mais ce gain de place sera beaucoup plus élevé sur du contenu textes. De plus, la compression est juste une option parmi les nombreuses que compte ZFS. Malheureusement, la consommation mémoire qui est très élevée pose des problèmes lors du redémarrage qui parfois, ne réussit à monter tout les systèmes fautes de places mémoires.

Snapshots et restaurations de ZFS

Sous ZFS, il est possible de faire des snapshots sur tous les systèmes de fichiers d'un pool. Ainsi, il sera possible de sauvegarder séparément les systèmes de fichiers récursivement avec leurs fils ou non. Leurs restaurations se feront elles aussi de manière individuelle. Après avoir réaliser le snapshot désirer, il sera judicieux d'envoyer la sauvegarde sur un support sécurisé. Afin de restaurer la sauvegarde réalisée, il faudra supprimer ce système de fichiers puis le restaurer par la commande "zfs receive". Nous allons réaliser un snapshot du système de fichier mysql, le supprimer et le restaurer par les commandes suivantes :

```
# Création du snapshot
zfs snapshot poolAsrall/bin/mysql@snapMysql.snap
# Sauvegarde du snapshot
zfs send poolAsrall/bin/mysql@snapMysql.snap > /home/mathieu/Documents/snapM
# Suppression du système de fichier poolAsrall/bin/mysql
zfs destroy poolAsrall/bin/mysql
# Restauration de la sauvegarde
zfs receive poolAsrall/bin/mysql < /home/mathieu/Documents/snapMysql.snap
# Changement du point de montage car celui par défaut sera son nom
zfs mountpoint=/media/poolAsrall/bin/mysql poolAsrall/bin/mysql
```

Après m'être connecter à une base de données mysql nommé wiki, j'ai créé et sauvegardé un snapshot du système poolAsrall/bin/mysql qui contenait les données de mysql. La sauvegarde ayant pour nom snapMysql.snap, et aura la même taille que le système de fichier sauvegardé. Après avoir supprimé ce système, je l'ai donc restaurer via la sauvegarde. La connexion à cette base se refait parfaitement, ainsi que l'accès à ces données.

Nous allons tester la sauvegarde incrémentale de zfs en insérant une ligne dans la base wiki et en comparant avec l'ancien snapshot :

```
# Ajout d'une ligne dans la base mysql "wiki"
# Création du second snapshot
zfs snapshot poolAsrall/bin/mysql@snapMysql2.snap
# Enregistrement de la difference entre les deux snapshots
zfs send -i poolAsrall/bin/mysql@snapMysql.snap poolAsrall/bin/mysql@snapMys
# Taille du snapShot snapMysql.snap soit 42Mo
ls -ali snapMysql.snap
```

```
1591678 -rw-r--r-- 1 root root 44766176 2012-03-11 16:53 ../snapMysql.snap
```

```
# Taille de l'incrément diff.snap soit 1.1Mo
ls -ali diff.snap
```

```
1582651 -rw-r--r-- 1 root root 1137672 2012-03-12 22:23 diff.snap
```

Comme précédemment, nous allons supprimer le système de fichier poolAsrall/bin/mysql avant de pouvoir le restaurer via le snapShot et l'incrément. Avant de pouvoir restaurer un incrément, il faut

d'abord réaliser un rollback depuis le snapshot. La commande `zfs rollback` permet, comme son nom l'indique, de retourner aux données d'un snapshot donné.

```
# Destruction du système de fichier
zfs destroy poolAsrall/bin/mysql
# Restauration du snapshot
zfs receive poolAsrall/bin/mysql < /home/mathieu/Documents/snapMysql.snap
# Retour au snapshot
zfs rollback poolAsrall/bin/mysql@snapMysql.snap
# Insertion de l'incrément au système
zfs receive poolAsrall/bin/mysql < diff.snap
```

Concernant les sauvegardes à chauds, la restauration posera des problèmes si une modification de la base est en cours. Afin de corriger ce problème, il faudra utiliser les commandes de `mysql lock` et `unlock` avant et après le snapshot.

Rsnapshot

Introduction Rsnapshot est un logiciel libre qui permet de réaliser des sauvegardes et restaurations. Ces sauvegardes sont incrémentales : si elles sont réalisées tout les jours et qu'elles contiennent des fichiers identiques alors des liens en durs seront fait de cette sauvegarde à la sauvegarde précédente. Le nom des sauvegardes auront le noms de l'option de l'intervalle (`hourly.x`, `daily.x`, `monthly.x` et `yearly.x`) avec `x` le numéro de la sauvegarde. La sauvegarde la plus récente portant le numéro 0, et dès qu'une nouvelle sauvegarde sera réalisée alors la sauvegarde 0 deviendra 1 et ainsi de suite.

Installation et configuration On peut l'installer via le gestionnaire de paquet :

```
sudo apt-get install rsnapshot
```

Rsnapshot n'a qu'un fichier de configuration `/etc/rsnapshot.conf` où est précisé le répertoire de destination des sauvegardes, les emplacements à sauvegarder, les scripts à exécuter ou la périodicité des sauvegardes à exécuter. Rsnapshot n'a pas prévu un traitement spécifique pour la sauvegarde des bases de données. Ainsi, il utilise les programmes propres aux sauvegardes de bases de données comme `pg_dumpall` pour `postgresql` et `mysqldump` pour `mysql`. Des scripts pour l'utilisation de ces programmes sont mis à disposition dans un des dossiers sources de `rsnapshot`. Ils contiennent deux lignes : l'une pour utiliser le programme propre de sauvegarde de la base de données et l'autre pour changer les droits du fichier résultant. Dans mon cas, je vais donc sauvegarder ma base `mysql` via ce script. Voici les principales configurations du fichier `rsnapshot.conf`.

```
# Spécification du répertoire de destination des sauvegardes
snapshot_root /home/mathieu/.snapshots/
#Pour utiliser les sauvegardes via ssh
cmd_ssh /path/to/ssh
# Spécification de l'intervalles des sauvegardes
interval hourly 6
interval daily 7
# Spécification du repertoire à sauvegarder et de l'emplacement dans le repertoire de destination
backup /media/poolAsrall/home/util_0 localhost/home/
# Spécification du script à executer et de l'emplacement dans le repertoire de destination
backup_script /usr/local/bin/backup_mysql.sh localhost/mysql/
```

Sauvegardes Afin d'automatiser la sauvegarde, il faudra ajouter le lancement de `rsnapshot` dans le cron. Dans l'exemple suivant, une sauvegarde est réalisée toutes les 4h et tout les jours à 23h.

```
crontab -e
```

```
0 */4 * * * /usr/local/bin/rsnapshot hourly
30 23 * * * /usr/local/bin/rsnapshot daily
```

Pour tester rsnapshot, nous avons exécuter la commande `"/usr/bin/rsnapshot hourly"` trois fois de suite. La première fois, la sauvegarde mettra quelques dizaines de secondes à se terminer alors que les sauvegardes suivantes ne mettront que deux à trois secondes. Pourtant, la taille des dossiers créés sera la même. En effet, les dossiers sauvegardés n'étant pas modifier entre les sauvegardes, juste des liens en durs seront effectuer de la première sauvegarde aux suivantes.

```
# on test l'espace utilisé par le dossier de sauvegarde .snapshots/
du -hs .snapshots/
```

```
1,5G .snapshots/
```

La taille du dossier de l'espace utilisateur `util_0` de ZFS fait toujours 1,42Go. On peut constater que l'espace du dossier `.snapshots/` est légèrement plus volumineux car il contient aussi la sauvegarde des bases mysql. Voyons le contenu de ce dossier et l'espace utilisé pour chacun des sous-dossiers hourly.

```
du -hs .snapshots/*
```

```
1,5G ./hourly.0
868K ./hourly.1
868K ./hourly.2
```

Comme on peut le constater, l'espace de la sauvegarde la plus récente contient les données et les anciennes sauvegardes font justes des liens vers ces fichiers. A chaque modification d'un de ses fichiers, la réécriture du fichier devrait etre effectué. Nous allons copié un fichier iso de 417Mo dans le repertoire `util_0`, réalisé une sauvegarde, supprimer ce fichier et réaliser une nouvelle sauvegarde.

```
du -ms .snapshots/*
```

```
1481 .snapshots/hourly.0
417 .snapshots/hourly.1
1 .snapshots/hourly.2
1 .snapshots/hourly.3
```

On peut constater que l'or de la modification du repertoire à sauvegarder, Rsnapshot rajoute le fichier entier au snapshot. Puis si le fichier est toujours présent alors il réalisera juste un lien vers le snapshot qui contient ce fichier. Maintenant, nous allons tester si Rsnapshot analyse juste la derniere sauvegarde ou toutes les sauvegardes pour voir si des fichiers existent déjà afin de réaliser les liens. Pour ce faire on va réaliser une sauvegarde, le rajout du même fichier de 417Mo puis une nouvelle sauvegarde. En testant la taille des dossiers, on verra si le fichier apparait une ou deux fois.

```
du -ms .snapshots/*
```

```
1897 .snapshots/hourly.0
1 .snapshots/hourly.1
417 .snapshots/hourly.2
1 .snapshots/hourly.3
1 .snapshots/hourly.4
```

Comme on peut le voir ci-dessus, le fichier entier de 417Mo apparait deux fois dans les snapshots. On peut donc en conclure que Rsnapshot compare juste le dossier à sauvegarder au dernier snapshot réalisé.

Restaurations Avec rsnapshot, il n'y a pas de façon automatique incorporé pour restaurer des fichiers ou des bases de données. Il faudra donc effectuer les restauration manuellement en se déplaçant dans l'arborescence des sauvegardes et puis, en choisissant le ou les fichiers à restaurer. Afin de partager les sauvegardes avec NFS, il faudra créer un repertoire de sauvegarde propres à chaque utilisateur

avec des droits de lecture où ils pourront choisir leurs fichiers sans les modifier, et les rapatrier à l'emplacement désiré. Concernant les bases de données, on pourra les restaurer en utilisant, de la même manière que pour leurs sauvegardes, les programmes propres aux SGBD.

2.4 Déduplication avec LessFS

2.4.1 Introduction

LessFS est un système de fichier permettant la déduplication. Lessfs s'utilise avec le logiciel FUSE (Filesystem in Userspace) et Tokyocabinet (jeu de librairie permettant d'utiliser une base de données). Contrairement au système de fichier normal comme ext3, lessfs stock uniquement les blocks de données qui n'existent pas déjà. Quand un block de données existent déjà, il fait une simple référence à ce fichier et stock cette référence dans les tables de TokyoCabinet.

2.4.2 Tests

Afin de tester lessfs sur un cas concret, nous avons choisi de mettre en place ce logiciel sur les données générées par un serveur mail. Ces données étant des fichiers textes, la déduplication devrait atteindre son maximum. La taille des données non dédupliquées du serveur mail est de 20Go. Dans un premier temps, tous les mails sont stockés dans des fichiers différents. Par défaut la taille des blocks de données générés par lessfs est de 128ko (le maximum). Nous allons donc tester avec cette taille de blocks puis avec une taille de blocks de 64ko et pour finir une taille de 4ko (le minimum). Plus la taille des blocks est faible, plus la déduplication devrait être élevée car il y aura plus de chances de trouver un block identique mais cela générera énormément d'entrées/sorties et donc de consommation mémoire et CPU. Lessfs a un fichier de configuration où on précise le répertoire pour stocker les blocks de données, la base de données tokyocabinet, la taille du cache ou la taille des blocks (voir en annexe). Aussi, lessfs permet de compresser ces blocks de données. Les algorithmes possibles sont qlz (par défaut), qlz15, snappy, Deflate, bzip et lzo. Nous allons tester l'algorithme qlz, bzip, lzo et Deflate. Après avoir copié les données du serveur mail dans le système de fichier lessfs monté sur /media/lessfs. Dans notre premier test, l'architecture du serveur de mail est en maildir. Chaque mail représente un fichier et sont répertoriés par utilisateurs.

```
du -hs /media/lessfs/
```

```
20G /media/lessfs/
```

On constate que le répertoire fait bien une taille de 20 Go. Regardons maintenant la taille des répertoires stockant les blocks de données et la base de données. Celle-ci devrait être très largement inférieure. J'ai choisi de stocker ces données dans /data/, les blocks de données dans /data/dta/ et la base de données dans /data/mta.

Taille des blocks=4ko et Compression=qlz

```
du -hs /data/*
```

```
11G /data/dta
694M /data/mta
```

On peut constater que la déduplication a pas très bien fonctionné. Le nombre de blocks stockés dans la base est très important du aux très grands nombres de blocks référencés. Le taux de déduplication est inférieur à deux et cela a mis 4h10 pour copier tout le contenu.

Taille des blocks=4ko et Compression=bzip

```
du -hs /data/*
```

```
8,4G /data/dta  
509M /data/mta
```

La déduplication fonctionne légèrement mieux que le test avec la compression qlz mais le temps reste très important. Cela a mis 3h30.

Taille des blocks=4ko et Compression=lzo

```
du -hs /data/*
```

```
9,3G /data/dta  
559M /data/mta
```

Comme dans le test précédent, la déduplication est faible mais le temps imparti est très largement inférieur. Cela a mis 1h49.

Taille des blocks=4ko et Compression=Deflate

```
du -hs /data/*
```

```
7,8G /data/dta  
559M /data/mta
```

La déduplication est légèrement supérieur et cela mis un temps a peu près identique, c'est à dire 1h51.

Taille des blocks=64ko et Compression=qlz

```
du -hs /data/*
```

```
2,4G /data/dta  
99M /data/mta
```

Avec la taille des blocks de 64ko, la déduplication est très largement supérieur et dans un temps lui aussi largement inférieur, c'est à dire 18 minutes. On constate aussi que la taille de la base de données est lui aussi inférieur car le nombre de blocks a référencé est moins important.

Taille des blocks=64ko et Compression=bzip

```
du -hs /data/*
```

```
2,5G /data/dta  
99M /data/mta
```

Cette fois, la taux de deduplication est quasiment identique avec 8x mais cela a mis plus de deux fois plus de temps, c'est à dire 35 minutes. La compression bzip est donc bien moins rapide que qlz.

Taille des blocks=64ko et Compression=lzo

```
du -hs /data/*
```

```
5,6G /data/dta  
99M /data/mta
```

La déduplication avec lzo est moins performante que celle avec bzip ou qlz mais elle se termine dans un temps largement inférieur, c'est à dire 19 minutes.

Taille des blocks=64ko et Compression=Deflate

```
du -hs /data/*
```

```
1,7G /data/dta  
99M /data/mta
```

Comme dans le test avec les blocks a 4ko, la compression la plus performante est Deflate. Dans ce cas, la compression a très bien fonctionné car elle a mis que 18 minutes avec un taux de 11,7.

Taille des blocks=128ko et Compression=qlz

```
du -hs /data/*
```

```
1,7G /data/dta
84M /data/mta
```

La taille de blocks deux fois supérieur, la déduplication et la compression avec qlz ont atteint les performances de la compression Deflate du test précédent. Malgré le taux de déduplication égale, le temps quand a lui, a mis 32 minutes.

Taille des blocks=128ko et Compression=bzip

```
du -hs /data/*
```

```
1,6G /data/dta
84M /data/mta
```

Cette fois, la déduplication a bien fonctionné meme mieux que le premier test. La compression avec bzip fonctionne donc mieux que la compression qlz. Le taux de déduplication est de 12,5x en 36 minutes.

Taille des blocks=128ko et Compression=lzo

```
du -hs /data/*
```

```
5,7G /data/dta
84M /data/mta
```

Dans ce test le taux de déduplication n'est que 3.5x mais la compression lzo se rattrape dans la rapidité car le temps n'a été que de 19 minutes. Ce temps a été plus rapide que sur le système de fichier ext3 non dédupliqué.

2.4.3 Taille des blocks=128ko et Compression=Deflate

```
du -hs /data/*
```

```
1,3G /data/dta
84M /data/mta
```

On constate que la déduplication par Deflate est la plus performante car elle atteint un taux de 15.3x. De plus, la copie a mis un temps de 17 minutes.

Tableau récapitulatif

Taille des blocks / Compression	qlz	bzip	lzo	Deflate
4Ko	1.8x/4h10	2,4x/3h30	2.15x/1h49	2.56x/1h51
64Ko	8.5x/18m	8x/35m	3.5x/19m	11.7x/18m
128Ko	11.7x/32m	12.5x/36m	3.5x/19m	15.3x/17m

La déduplication avec lessFs est donc très performante mais cette performance est très liée aux performances de la machine l'utilisant. En théorie, on pouvait penser que plus la taille des blocks était faible, plus lessFs pourrait trouver des blocks similaires. Mais on a pu constater que dans le cas d'un serveur mail et d'une machine assez peu performante, la déduplication était la meilleure quand la taille des blocks était la plus élevée. Un test a été effectué pour savoir le temps que mettait la copie du meme contenu sur un système de fichier ext3 non dédupliqué et le résultat a été de 23 minutes. LessFs est donc plus rapide de 6 minutes tout en diminuant la taille de plus de quinze fois. Parmi les différents

tests, la compression avec Deflate a été la plus élevée car cette compression utilise l'algorithme LZ77 et le codage Huffman qui, sur du contenu qui revient souvent comme sur des mails est très performant. Si on désire sauvegarder ces données, c'est possible de "geler" la base tokyocabinet via une connexion telnet afin de réaliser une sauvegarde et une autre option permet de la "dégeler". Après l'avoir testé avec Rsnapshot, la restauration et l'accès aux données se réalise sans problème quand aucune écriture n'est en cours. Dans le cas contraire, la restauration s'effectuera sans problème mais aucun fichier ne sera accessible. Les données seront donc corrompues.

Testons maintenant une architecture du serveur mail en mailBox. Cette fois, les mails de chaque utilisateur sont stockés dans un fichier. Il y aura donc un fichier par utilisateur mais les données seront identiques.

Taille des blocks=128ko et Compression=Deflate

```
du -hs /data/*
```

```
2,0G /data/dta  
83M /data/mta
```

Taille des blocks=4ko et Compression=Deflate

```
du -hs /data/*
```

```
14G /data/dta  
695M /data/mta
```

Les performances avec lessFs peuvent donc varier suivant le nombre de fichiers traités malgré des données identiques mais mises sous une forme différente. La déduplication a moins bien fonctionné. Il a déduplicé environ deux fois moins mais dans un temps lui aussi deux fois moins pour des blocks d'une taille de 128ko. Pour l'autre test, le temps a été exactement la même durée et la déduplication a été aussi deux fois moins performante. Cette différence de déduplication peut s'expliquer car dans l'architecture en mailDir les mails sont séparés et LessFS peut plus facilement déterminer les similitudes entre des fichiers et donc les blocks étaient identiques. LessFS crée un fichier de stat dans le quel il écrit le ratio de compression qu'il a réalisé sur chaque fichier (voir annexes). Pour le dernier test effectué, on constate que le taux de compression est d'environ 1,5x pour chaque fichier.

Pour conclure, la déduplication alliée à la compression à la volée comme le fait lessFS permet d'économiser énormément d'espace. Le gain d'espace variera selon le nombre et le type de fichiers mais il aura pour la plupart du temps, de meilleures performances que sur un système non déduplicé. L'autre avantage est l'accès aux données qui sera transparent pour les utilisateurs contrairement à une archive tar qui faudrait décompresser avant de pouvoir l'exploiter.

Chapitre 3

Logiciels

Dans le cadre du projet, nous avons entrepris de regarder quelles étaient les solutions logicielles pour la sauvegarde. Nous avons décidé de nous orienter vers des solutions compatibles avec les système UNIX et le plus souvent libre, ce qui n'empêche pas que cela soit aussi des solutions envisageables pour des systèmes WINDOWS. Nous avons essayé de faire le tour des solutions existantes les plus connues, chacun avec leur spécificité, leurs avantages et leurs défauts. On rappellera que il a délibérément été fait un choix concernant les logiciels examinés et ceux testés, et nous n'avons aucune prétention d'exhaustivité.

3.1 Bacula

Bacula (Pour « Backup-Dracula ») est une solution logicielle pour la sauvegarde très complète. Ce logiciel est sûrement l'un des plus complet dans sa catégorie. Cette solution que l'on qualifiera de professionnelle est particulièrement bien adaptée lorsqu'il s'agit d'effectuer des sauvegardes sur plusieurs postes clients avec un serveur qui aura la tâche d'effectuer et de superviser l'ensemble des opérations. Cette solution nécessite une base données relationnelle pour fonctionner, comme par exemple MySQL

Le logiciel Bacula est un peu difficile à prendre en main, notamment car il permet une configuration très fine. En réalité le logiciel se décompose en trois éléments :

- Un superviseur (director) qui sera installé sur le serveur et aura la tâche d'effectuer et de superviser les sauvegardes
- Un démon installé sur le client sur lequel sont stockés les données à sauvegarder
- Un démon installé sur le serveur qui aura la tâche de stocker les données sauvegardées (ou de les enregistrer sur bande magnétique, par exemple)

Du à sa configuration, ce logiciel permet de s'adapter à quasiment n'importe quelle stratégie de sauvegarde. Il permettra de faire tout type de sauvegarde, aussi bien complète que différentielle, ou incrémentale. Il permettra aussi le choix du support de sauvegarde, comme un disque dur ou l'utilisation de bandes magnétiques.

Il bénéficie d'outils permettant à l'administrateur de communiquer avec le logiciel, comme une console dédiée à cet effet. Cette console permet non seulement de lancer des sauvegardes et de les planifier, mais il permettra aussi d'effectuer des opérations de restauration.

Bacula est entièrement utilisable en ligne de commande. C'est compréhensible puisqu'il est souvent utilisé sur des serveurs ne disposant pas d'interface graphique. Toutefois des solutions permettant de configurer et d'utiliser Bacula graphiquement existent.

En résumé, on pourra dire que Bacula est un logiciel de sauvegarde adapté et performant dès que cela se passe dans un cadre professionnel ou que la stratégie de sauvegarde concerne des postes clients

(et que l'on souhaite une centralisation), ou encore lorsque l'on souhaite configurer cette stratégie de sauvegarde finement. En contrepartie, il faudra un peu de temps à l'administrateur pour comprendre le fonctionnement de Bacula, le mettre en place, et surtout le configurer puis l'utiliser correctement. De plus il semble qu'aucune optimisation de sauvegarde ne soit nativement intégré, comme de la compression à la volée. Ainsi une sauvegarde complète sur 100 Go de données nécessitera réellement 100 Go d'espace de stockage.

3.2 rsnapshot

rsnapshot est une application pour la sauvegarde. Contrairement au logiciel précédent qui nécessite une base de données pour fonctionner, ce logiciel ne se base que sur le programme rsync pour la réalisation de sauvegardes. Comme son nom l'indique, il permet de faire des instantanés de systèmes de fichiers. Il est entièrement utilisable en ligne de commande.

rsnapshot permettra de réaliser des sauvegardes complètes et différentielles, mais chaque sauvegarde peut être traitée indépendamment l'une de l'autre, car ce sont en réalité des liens qui sont établis entre les fichiers non changés d'une sauvegarde à l'autre. Ainsi, même si de premier abord, chaque sauvegarde semble contenir l'ensemble des données sauvegardées à chaque fois, il n'en est rien. En d'autres termes, une restauration ne nécessitera la manipulation que d'une sauvegarde (même si en interne, des liens pointeront vers une sauvegarde antérieure).

rsnapshot permet de faire des sauvegardes localement, mais peut aussi faire des sauvegardes à travers le réseau (via le protocole ssh par exemple). Dans ce cas, le logiciel est seulement installé sur le serveur qui gère la sauvegarde, un serveur ssh fonctionnel étant tout de même nécessaire sur le client. Cela permettra un allègement des procédures de configuration sur chaque client lorsque l'on met en place une nouvelle stratégie de sauvegarde.

A l'heure actuelle, aucune compression des données n'est réalisée, ce qui fait qu'une sauvegarde complète nécessitera un espace de stockage équivalent à la quantité de données sauvegardées. On notera aussi que la restauration n'est pas intégrée. Ceci est compréhensible : étant donné qu'une seule sauvegarde est nécessaire, la restauration est triviale. Cela passera le plus souvent par une simple copie des fichiers, à l'aide d'outils tel que cp sous les système UNIX.

Beaucoup moins configurable et moins évolué que bacula, cette solution est avant tout intéressante pour sa facilité de configuration et d'utilisation. Dans un contexte privé, lorsqu'il s'agit de sauvegarder un ordinateur personnel, cette solution semble appropriée. Elle pourra être utilisée dans un contexte professionnel, mais sera au final plus compliquée à gérer que d'autres applications tel que celle vue précédemment.

3.3 git

GIT est avant tout connu pour être l'un des logiciels de gestion de versions les plus utilisés actuellement. Cependant, celui ci dispose de toutes les fonctionnalités nécessaires pour l'utiliser en tant que logiciel de sauvegarde. En pratique celui-ci ne sera utilisé que dans des cas très particuliers.

Sa configuration est plus que triviale, et permettra de faire des sauvegardes localement, mais aussi à travers le réseau, et ceci à l'aide de différents protocoles, les plus courants étant ssh et http. Il peut donc être très rapidement mis en place.

On remarquera qu'au delà des fichiers textes et pour des tailles réduites, GIT n'est pas très performant pour la sauvegarde de données à grande échelles. GIT gère très mal les fichiers binaires. De plus l'utilisation de dépôts locaux et distants occupera de la place aussi bien localement qu'à distance et les sauvegardes prendront plus de temps qu'avec un outil classique.

Comme on pourrait s'en douter, on n'a aucune optimisation, du moins en ce qui concerne la sauvegarde. La restauration se limite aux commandes habituelles de git (à savoir des synchronisations et des rollbacks), et correspond moins aux optiques de sauvegardes. La planification et l'automatisation n'est logiquement pas intégrée, même si le recours à des outils par défaut tel que cron pourrait être effectué.

Même si git ne semble pas être une bonne solution pour les sauvegardes, elle peut être appropriée dans certains cas bien particuliers. Par exemple, on pourrait utiliser ce logiciel pour stocker les données de configuration (qui sont généralement le contenu du répertoire `/etc`). Ainsi on pourrait ajouter des fichiers de configuration semblables à une multitude de clients sans trop d'effort.

Au final, git ne semble pas être une solution viable en ce qui concerne la sauvegarde de données en général. Particulièrement adapté aux fichiers texte (puisque'il a été conçu pour le partage de code source entre membres d'un projet), il faudra se cantonner à cette utilisation, au risque d'avoir des fenêtres de sauvegarde particulièrement importantes et occuper une forte quantité d'espace.

3.4 snapshots LVM

Les snapshots sont implémentés nativement pour les partitions LVM. On pourra donc s'en servir pour effectuer des sauvegardes à chaud, très pratique pour les serveurs qui nécessitent une disponibilité permanente, par exemple.

Après avoir mis en place un volume logique dans un groupe de volume, il suffira de lancer la commande :

```
lvcreate -s -n mon\_snapshot -L 20G /dev/monvg/monlv
```

Ainsi, un autre volume qui contiendra la sauvegarde sera créé dans le même groupe de volumes que le volume sauvegardé. Il faudra donc au départ laisser un espace considérable dans le groupe le volume concerné. De plus, il faudra indiquer la taille utilisée pour le snapshot. En fait, un snapshot peut prendre de la place, car il doit stocker toutes les modifications qui sont faites sur les données depuis la réalisation du dernier snapshot. Cette taille doit être choisie avec soin, il ne faut surtout pas prendre une valeur trop faible. Il est toujours possible de vérifier le taux de remplissage du snapshot avec la commande `lvs`.

Pour utiliser ou restaurer une sauvegarde faite à partir d'un snapshot LVM, il suffira juste de le monter et il sera alors immédiatement disponible.

A noter que la réalisation de snapshots n'est pas une solution à elle seule pour la mise en place d'une stratégie de sauvegarde. En pratique, on utilisera un snapshot pour effectuer une image d'un volume logique à chaud, ce qui sera fait instantanément sans gêner le bon fonctionnement du serveur. Ensuite, on utilisera le volume logique créé et contenant l'image disque comme la source pour effectuer une réelle sauvegarde à l'aide d'un des outils vus précédemment.

La réalisation d'instantanés de volumes logiques à l'aide des outils implémentés nativement à snapshot seront des outils pouvant être utilisés dans la réalisation de sauvegardes et permettant leur optimisation.

3.5 backup-manager

backup-manager est petit logiciel, plutôt simple d'utilisation, permettant lui aussi d'effectuer des sauvegardes. Un des principaux avantages par rapport à ses concurrents, est qu'en plus de supporter les différents types de sauvegarde standards (sauvegardes complètes et incrémentales), il implémente

des techniques d'optimisation. On pourra par exemple effectuer des sauvegardes sous la forme d'une archive compressée (tar.gz), ce qui permettra d'économiser de l'espace.

Le paramétrage est relativement simple, bien qu'offrant un nombre important d'options. Il supporte toutes les opérations utilisées pour la sauvegarde, à l'exception de la restauration. C'est compréhensible du fait que les sauvegardes sont stockées sous forme d'archive et un outil standard de manipulation d'archives sera suffisant, comme tar par exemple.

L'automatisation et la planification des sauvegardes ne sont pas implémentées. Toutefois, il suffira de configurer cron pour avoir ces fonctionnalités. Il permet de graver les sauvegardes directement sur CD/DVD ou de les exporter sur des machines distantes.

Dans le cadre d'une stratégie de sauvegarde simple, pour un particulier par exemple, ce logiciel reste un bon outil. Cela sera un peu plus compliqué lorsqu'il s'agira de sauvegarder différents postes clients, ou de grosses structures avec une stratégie élaborée. Les sauvegardes sont stockées sans réellement être organisées : on a donc tendance à être dépassé par le nombre de fichiers et de retrouver les bons fichiers lorsqu'il s'agit d'effectuer une restauration à un instant donné.

3.6 Areca Backup

3.6.1 Caractéristiques

- Archives de compression en format Zip et Zip64.
- Archives de cryptage (AES128 et algorithmes de chiffrement AES 256).
- Stockage sur disque dur local, lecteur réseau, clé USB, FTP, FTPS (implicite et explicite avec SSL / TLS) ou un serveur SFTP.
- Filtres de fichier : par extension, répertoire, expression régulière, taille, date, statut, avec AND / OR /NOT logique.
- Sauvegarde Incrémentielle, différentielle et sauvegarde complète.
- Archives et fusions : Permet fusionner les archives contigus dans une archive unique pour économiser l'espace de stockage.
- Sauvegarde à partir d'une date donnée : Permet de récupérer des archives à partir d'une date spécifique.
- Mécanisme de transaction : tous les processus critiques, telles que les sauvegardes ou fusions sont transactionnels (commit rollback). Cela garantit l'intégrité des sauvegardes.
- Rapports de sauvegarde : Areca génère des rapports de sauvegarde qui peuvent être envoyés par email.
- Programmation de scripts de sauvegarde : Areca peut lancer des scripts shell après la sauvegarde.
- Les permissions des fichiers, liens symboliques et les pipes nommés peuvent être stockés et récupérés (Linux uniquement).
- Les tâches : Une tâche de sauvegarde est appelée «target» dans la terminologie de Areca. Il définit les fichiers qui seront stockés (sources), où ils seront stockés (destination) la façon dont ils seront stockés (si elles seront compressées, cryptées, etc.). Les «targets» peuvent être organisés en «targets groups».
- Areca utilise la taille du fichier et date de dernière modification pour détecter les fichiers modifiés, si un de ces attributs est modifié (quelle que soit sa valeur), le fichier est marqué comme modifié. Cela permet une détection rapide des fichiers modifiés.

3.6.2 Modes de stockage

Areca peut gérer de multiples modes de stockage :

- Standard : C'est le mode par défaut, une nouvelle archive sera créée pour chaque sauvegarde. Tous les fichiers nouveaux ou modifiés depuis la dernière sauvegarde seront stockés dans cette archive.

- Delta : Une nouvelle archive sera créée pour chaque sauvegarde. Tous les *parties* modifiées de fichiers depuis la dernière sauvegarde seront stockés dans cette archive. Ce mode est particulièrement utile pour la manipulation des fichiers volumineux. (Areca utilise un algorithme qui est similaire à rsync pour détecter et gérer les parties modifiées des fichiers).
- Image : Une archive unique sera créée et mise à jour à chaque sauvegarde.

3.6.3 types de sauvegarde

Areca peut gérer les types de sauvegardes suivants :

- Sauvegarde complète : Lors d'une sauvegarde complète est effectuée, tous les fichiers sont stockés dans une archive (même s'ils ont été modifiés ou non).
- Sauvegarde incrémentale : Quand une sauvegarde incrémentale est réalisée, seuls les fichiers qui ont été modifiés depuis la dernière sauvegarde sont stockés dans l'archive.
- La sauvegarde différentielle : Lors d'une sauvegarde différentielle est effectuée, seuls les fichiers qui ont été modifiés depuis la dernière sauvegarde complète sont stockés dans l'archive.

3.7 Sauvegarde en ligne

Nous n'avons pas eu l'occasion de réaliser des tests sur les sauvegardes en ligne, étant donné qu'ils ne sont pas gratuits. Même si cela peut être utile dans certains cas, notamment lorsque le nombre de données à sauvegarder n'est pas trop important, on comprend bien les limites de ces techniques. Le facteur limitant est le débit montant qui en général beaucoup moins élevé que le débit descendant. De plus, même si un chiffrement de données est possible, cela implique d'avoir une totale confiance dans le prestataire, et surtout dans sa sécurité et sa disponibilité.

Chapitre 4

Tests de logiciels

Afin de tester les différents outils de sauvegarde, nous avons procédé à la mise en place de tests, ce qui permettra de comparer les logiciels au niveau des performances. Pour cela il a fallu réfléchir à différents scénarios de sauvegardes que l'on pourrait rencontrer dans un contexte professionnel.

4.1 Serveur mail

Les mails sont généralement des données importantes au sein d'une entreprise. En effet, ils constituent, dans beaucoup de situations, la majorité des échanges avec l'extérieur et une partie des échanges internes. En plus d'être vitales pour l'entreprise au quotidien, ces données peuvent servir d'historique, ce qui peut parfois être intéressant. Etant donné l'importance de ces mails, sauvegarder ces données semble donc cohérent et justifié. L'objectif sera donc de tester différents outils pour la sauvegarde et de les comparer.

4.1.1 Scénario

Données

Il a été décidé de générer une quantité importante de données afin que cela soit représentatif d'une situation plausible. 20 G0 de mails seront d'abord générés.

Utilisateurs

La encore, nous avons essayé d'être le plus réaliste possible. Il a été décidé de générer 100 utilisateurs, qui seront séparés en 3 groupes :

- 10 utilisateurs pour le premier service
- 30 utilisateurs pour le second service
- 60 utilisateurs pour le troisième service

De plus, chaque utilisateur fait partie d'une « équipe » de 3 à 4 membres.

Répartition des mails

Ici encore nous avons essayé d'être réalistes dans les échanges de mails entre utilisateurs. Ainsi, nous avons fixé les rapports suivant :

- 5% des mails sont envoyés à tous les utilisateurs
- 10% des mails sont envoyés à tout le service
- 25% des mails sont envoyés à tous les membres de l'équipe.
- 60% des mails sont propres à chaque utilisateur.

Composition des mails

Chaque mail aura une taille aléatoire. Toutefois, il devra peser au moins 100Ko et au plus 2Mo.

Sauvegardes

Une première sauvegarde complète portera sur les 20 Go de données générées. Après que les 20 Go de données aient été sauvegardées, nous générerons encore 1 Go de données selon le même procédé pour ensuite effectuer une sauvegarde différentielle. Enfin, une restauration sera réalisée.

4.1.2 Mise en place

Serveur mail

Un serveur mail postfix a été mis en place et configuré classiquement. Il est possible de gérer les mails des utilisateurs en mode mailbox ou maildir :

- en mode mailbox, tous les mails d'un utilisateurs sont stockés dans le même fichier au nom de l'utilisateur
- en mode maildir, chaque mail est un fichier séparé et l'ensemble des mails d'un utilisateurs sont stockés dans un répertoire au nom de l'utilisateur

Les tests se porteront sur les deux configurations pour voir si cela a un impact sur les sauvegardes.

Création d'utilisateurs

A l'aide de scripts bash, 100 utilisateurs ont été générés.

Services et équipes

La encore, ce sont des scripts bash qui ont permis d'alimenter le fichier `/etc/aliases` et de générer les fichiers adéquats. Un échantillon de ces fichiers est disponible en annexe.

Création des données

Pour générer les mails, nous avons opté pour l'utilisation de l'outil postal. A la base postal est un benchmark qui permet de tester un serveur mail en lui envoyant des mails selon des paramètres envoyés au logiciel. Ainsi, la commande utilisée est la suivante :

```
postal -M 100 -m 2000 -c 50 -r 1000 localhost list_users.txt
```

Avec :

- `-M` : taille minimale d'un mail (en Ko)
- `-m` : taille maximale d'un mail (en Ko)
- `-c` : nombre de mails par connexion
- `-r` : nombre de mails par minute
- `localhost` : serveur cible
- `list_user.txt` : la liste des utilisateurs pour envoyer les données

Pour ce qui est de la liste des utilisateurs, elle a été générée avec un script bash. Pour respecter les quotas précédemment, le groupe qui contient tous les utilisateurs apparaît une fois, les services 2 fois, les équipes 5 fois, etc... Il suffira de lancer la commande et de l'arrêter au moment où les 20 Go de données auront été générées.

4.2 Résultats

4.2.1 Sauvegarde complète

4.2.2 Sauvegarde incrémentale

4.2.3 Restauration

Logiciels	Taille initiale	Temps	Données transférée	Espace occupé
bacula	20 Go	20 min	20 Go	20 Go
rsnapshot	20 Go	22 min	10 Go	20 Go
git	20 Go	41 min	20 Go	20 Go (+2.9 Go)
backup-manager	20 Go	25 min	3.9 Go	3.9 Go
lvcreate	20 Go	1 sec	20 Go	20 Go

TABLE 4.1 – Sauvegarde complète sur un serveur de mail

Logiciels	Type BAL	Taille ajoutée	Données transférées	Temps	Espace occupé	Espace to
bacula	mailbox	+1 Go	21 Go	20 min	+21 Go	41 Go
bacula	maildir	+1 Go	1 Go	48 sec	+1 Go	21 Go
rsnapshot	mailbox	+1 Go	20 Go	20 min	+21 Go	41 Go
rsnapshot	maildir	+1 Go	1 Go	51 sec	+1 Go	21 Go
git	mailbox	+1 Go	21 Go	70 min	+22 Go (+3.1 Go)	22 Go (+6
git	maildir	+1 Go	1 Go	74 sec	+1 Go	21 Go
backup-manager	mailbox	+1 Go	4.1 Go	28 min	+4.1 Go	8 Go
backup-manager	maildir	+1 Go	54 Mo	1 min	+54 Mo	1.2 Go
lvcreate	mailbox	+1 Go	??	1 sec	+20 Go	20 Go
lvcreate	maildir	+1 Go	??	1 sec	+20 Go	20 Go

TABLE 4.2 – Sauvegarde incrémentale sur un serveur de mail

Logiciels	Temps	Simplicité	Configuration	Recouvrement partiel	Procédure
bacula	20 min	+	++	++	++
rsnapshot	20 min	++	-	+	++
git	11 min	+	-	-	+
backup-manager	10 min	+	-	-	++
lvcreate	20 min	++	-	++	++

TABLE 4.3 – Restauration sur un serveur de mail