Sauvegardes dédupliquées avec BorgBackup : retour d'expérience

Maurice Libes
Institut OSU Pytheas
campus de luminy – Bât. Oceanomed
13009 Marseille

Didier Mallarino
Institut OSU Pytheas
Université de Toulon, Bât. X
Avenue de l'Université
83130 La Garde

Résumé

Le volume des données scientifiques stockées dans les laboratoires de recherche augmente régulièrement. Des volumes de plusieurs dizaines à quelques centaines de téraoctets de données sont devenus fréquents et les solutions traditionnelles commencent à souffrir de durées de sauvegarde trop importantes et peuvent être sujettes à des interruptions ou erreurs qui imposent de recommencer si le logiciel ne gère pas les reprises.

Ces fortes volumétries induisent plusieurs problèmes inhérents à la sauvegarde des données :

- *durées de sauvegarde de plus en plus longues,*
- *sollicitation du réseau accrue,*
- *politique de sauvegarde revue à la baisse par manque d'espace de stockage*

Dans ce contexte il est intéressant de chercher des solutions logicielles qui pourraient réduire le volume des données sauvegardées et donc la durée de ces sauvegardes

Nous présentons dans cet article la solution de sauvegarde dédupliquée "BorgBackup", logiciel libre en python, qui présente des fonctionnalités très intéressantes: déduplication, compression, chiffrement, reprise après interruption sur sauvegarde incomplète.

Cette solution de sauvegarde "BorgBackup" présente un niveau de maturité satisfaisant et donne tous les signes d'un bon candidat à mettre en exploitation : stabilité, facilité d'utilisation, fonctionnalités modernes

Après avoir rappelé en quoi consiste la déduplication, nous détaillerons les processus d'installation sur les clients et les serveurs, ainsi que le mode d'utilisation et les fonctionnalités courantes. Nous donnerons également quelques valeurs de performances en compression de volume, et durées de sauvegardes.

Mots-clefs

BorgBackup, sauvegardes, déduplication, compression, chiffrement

1 Introduction – contexte

Les solutions traditionnelles de sauvegarde basées sur les logiciels libres (BackupPC [6], bacula, rdiff-backup...) mettent souvent en œuvre diverses commandes de transfert de fichiers (rsync, scp, smb, ftp, tar), dont l'efficacité est variable selon le volume à sauvegarder. Elles sont basées sur des techniques éprouvées qui restent malgré tout sur une logique classique de sauvegardes totales puis incrémentielles.

Plusieurs problèmes inhérents à la sauvegarde des données reste omniprésents:

- les volumes à sauvegarder imposent de quasiment doubler le stockage sur des baies séparées afin d'y stocker les sauvegardes elles mêmes.
- les durées de sauvegarde sont de plus en plus longues et deviennent difficiles à gérer. Des sauvegardes plusieurs jours, voire semaines, sont courantes. De telles durées augmentent ainsi le risque d'interruptions en cours de sauvegarde qui imposent de tout recommencer si le logiciel ne gère pas finement les reprises.
- la politique de sauvegarde (profondeur de la sauvegarde[1]), nombre de sauvegardes totales et incrémentales) est souvent revue à la baisse par manque d'espace de stockage.
- il est nécessaire de s'intéresser à la nature et contenus des jeux de données qu'on sauvegarde pour adapter une politique (fréquence, profondeur) en accord avec les utilisateurs.

C'est dans ce contexte qu'il est intéressant de chercher des solutions logicielles qui pourraient réduire le volume des données sauvegardé, et diminuer significativement les durées de ces sauvegardes. En ce sens les technologies qui mettent en œuvre des processus de « déduplication » sont de bonnes candidates car elles permettent de réduire ces deux facteurs, ainsi que la bande passante utilisée entre le client et le serveur de sauvegarde.

Certaines technologies commerciales ont déjà implémenté ces technologies de déduplication (Live Navigator, Netapp). Du coté des logiciels libres, rares sont les solutions qui mettent en œuvre de la déduplication au niveau des suites de blocs de fichiers. Des logiciels comme BackupPC [6] ou UrBackup [5] font de la déduplication mais uniquement au niveau des fichiers entiers : si ces logiciels rencontrent des fichiers identiques ils ne les sauvegardent qu'une seule fois. En revanche dans ce type de déduplication, si deux fichiers ne diffèrent que d'un seul octet, ils seront chacun sauvegardés entièrement.

La solution de sauvegarde "BorgBackup" [1] que nous présentons travaille au niveau des suites de blocs constitutives des fichiers, et seuls les nouveaux fragments modifiés de ces fichiers sont sauvegardés.

2 Rappel sur la déduplication en gestion de données

De nombreux documentations expliquent ce qu'est la déduplication, la définition de Wikipedia [3] est claire : « *en informatique la déduplication est une technique de stockage de données, consistant à factoriser des séquences de données identiques afin d'économiser l'espace utilisé* ».

1. Nombre de sauvegardes retenues

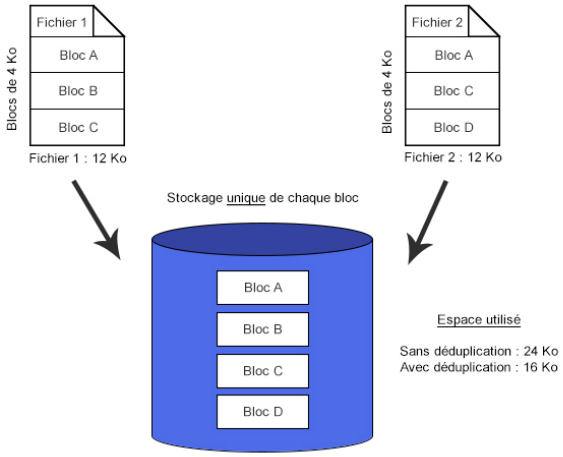

Figure 1 - schéma d'une sauvegarde dédupliquée de segments de fichier [7]

La déduplication est donc une une technologie qui opère non pas au niveau des fichiers comme certaines solutions traditionnelles, mais au niveaux des suites de blocs (parfois appelés « tronçons » ou « fragments » ou encore « chunks » dans le vocabulaire de Borg) constitutifs des fichiers. Ainsi, chaque fichier analysé est découpé en une multitude de tronçons de taille fixe. La déduplication consiste, donc tout d'abord à identifier les blocs par un calcul d'empreinte cryptographique (qui produit une valeur numérique unique appelé « hash ») puis à factoriser les suites de blocs redondantes en enlevant les données « dupliquées » (d'où le terme de déduplication). Sur des suites de plusieurs milliers de fichiers, cette mutualisation réduit significativement le nombre d'octets sauvegardés dans un dépôt.

Lors de la déduplication avec BorgBackup tous les fragments stockés dans un même dépôt sont pris en compte sans se préoccuper de leur origine (ils peuvent venir de différents fichiers et différents pc).

3 Fonctionnalités principales de BorgBackup

BorgBackup fonctionne en mode « client-serveur ». Il faut installer BorgBackup coté « client » sur chaque machine à sauvegarder, ainsi que sur une machine « serveur » dont le rôle sera de les stocker dans un dépôt de sauvegarde (nécessité d'avoir un important volume de stockage du coté du serveur Borg)

Brièvement voici les principales étapes d'une sauvegarde. Une grande partie du travail de sauvegarde se fait du coté du client à sauvegarder :

- ***Coté « client » :***
- Borg passe en revue chaque fichier à sauvegarder et teste si un fichier est nouveau, ou a été modifié (taille, *mtime*, ou l'*inode*)
- Si le fichier est retenu, borg le découpe en différents fragments (si le fichier est de la taille minimale, configurable, requise pour la fragmentation). La taille des fragments est définie par une variable que l'on peut modifier en ligne de commande. Elle est par défaut de 2^HASH_MASK_BITS (2^21=2Mb). [9]. Ce qui signifie que en dessous de cette valeur de 2Mb, le fichier ne sera pas fragmenté, et on aura de ce fait la même déduplication au niveau du fichier entier, que celle qui est effectuée par exemple par BackupPC.
- Borg teste si les fragments du fichier en cours de traitement existent déjà dans un « cache » local du coté client.
- Si le fragment rencontré est nouveau, alors, il est compressé (optionnellement), puis chiffré avant d'être envoyé par le réseau vers le serveur Borg.

– ***Coté serveur :***
– le serveur reçoit les fragments de fichiers compressés et chiffrés que lui envoient les clients, et les stocke (clé/valeur) dans un dépôt de sauvegarde.
– il calcule également une checksum crc32 pour chaque entrée stockée pour pouvoir tester des problèmes éventuels de corruption de données

Les fonctionnalités principales de BorgBackup sont les suivantes [1] :

i) Déduplication de l'espace de stockage : Chaque fichier à sauvegarder est fragmenté en un certain nombre de morceaux (« *chunks* »), et seuls les nouveaux fragments sont ajoutés au dépôt.

La déduplication fonctionne sur tous les morceaux de fichiers présents dans un dépôt. Il n'y a aucun fragment de fichier en double dans ces sauvegardes : tous les fragments dans le même dépôt sont référencés, peu importe s'ils proviennent de différentes machines, de différents fichiers, de sauvegardes précédentes, à partir de la même sauvegarde ou même à partir du même fichier unique.

L'intérêt de la déduplication au niveau des fragments de fichier, est qu'elle s'applique quels que soient les noms des fichiers et répertoires : si des fichiers sont copiés ou renommés, ils ne seront pas sauvegardés 2 fois. Si un gros fichier change de quelques octets, seuls les quelques fragments modifiés seront sauvegardés.

ii) Chiffrement des données sauvegardées : Par défaut, les sauvegardes de Borg sont sécurisées par chiffrement AES 256-bit. Chaque fragment est chiffré du coté client, avant d'être envoyé au serveur Borg. L'intégrité des données et l'authenticité sont vérifiées en utilisant un algorithme HMAC-SHA256.

Le chiffrement de Borg protège la confidentialité des données sauvegardées contre une éventuelle attaque du serveur de stockage. Chaque dépôt de sauvegarde possède sa propre clé de chiffrement. On ne peut relire un dépôt de sauvegarde sans avoir fourni la clé de chiffrement, via une « passphrase », spécifiques à chaque dépôt de sauvegarde.

iii) Compression des données : La compression est optionnelle dans BorgBackup, mais fortement recommandée. Toutes les données sauvegardées peuvent être compressées au choix par un des trois algorithme de compression : lz4 (très rapide, mais faible compression), zlib (rapidité et compression moyenne) ou lzma (rapidité plus faible, mais compression élevée). La compression des fragments de fichiers se fait du coté client avant l'envoi au serveur.

iv) Vitesse : Les sauvegardes effectuées avec Borg sont très rapides. Borg gère la détection rapide des fichiers non modifiés, ainsi que la mise en cache locale des fichiers et des index des nouveaux fragments. Du coté client la rapidité de la sauvegarde repose donc sur l'existence d'un cache local indexé qui permet de détecter rapidement si un fragment de fichier est déjà présent ou pas. Un fragment déjà présent en cache n'est pas envoyé au serveur Borg, d'où une forte économie de bande passante réseau et de volumétrie de stockage. En outre, pour améliorer les performances, le code critique de BorgBackup (fragmentation, compression et chiffrement) est implémenté en C/Cython.

v) Reprise sur erreur : Il n'y a pas à proprement parler de processus de reprise sur erreur, car si une sauvegarde est interrompue avant son achèvement complet, les fragments de fichiers de la sauvegarde partielle sont de toute façon stockés dans le dépôt de sauvegarde, et seront intégralement réutilisés par la prochaine sauvegarde qui durera d'autant moins longtemps. On peut dire que chaque fragment de fichier stocké dans le dépôt sert à réduire d'autant la prochaine sauvegarde.

vi) Un outil de contrôle et de corrections des dépôts de fragments est disponible (borg check) pour vérifier la cohérence des informations stockées dans les dépôts.

vii) Sauvegardes distantes : Borg peut stocker des données sur un serveur distant accessible via Ssh. Des gains importants de performance peuvent être atteints avec ssh par rapport à un système de fichiers réseau (sshfs, nfs, …).

viii) Sauvegardes accessibles comme un simple système de fichiers UNIX : Pour la restauration des données, les archives de sauvegarde peuvent être « montées » comme un simple système de fichiers dans l'espace utilisateur. Les sauvegardes sont alors accessible par les commandes traditionnelles de gestion de

fichiers « *cd* » « *ls* » « *cp* » . La restauration se faisant alors par une simple copie de fichier.

ix) « *last but not least* » Borg-Backup est un logiciel libre (Open Source Software) sous licence BSD.

4 Installation de BorgBackup

4.1 sur les clients et les serveurs

L'installation de BorgBackup est simple à réaliser. On installe le même code du coté client ou serveur. Sur les distributions récentes de Linux (Debian8, Ubuntu16, Mint18, etc) des paquets « *borgbackup* » sont disponibles :

```
$ apt-get install borgbackup borgbackup-doc
```

Si le paquetage « borgbackup » n'existe pas, il faut faire une installation à partir des sources par pip3 de python3. BorgBackup fonctionne avec *python3* seulement.

5 Utilisation et fonctionnalités de Borg-Backup

Pour montrer la simplicité d'utilisation, nous donnons quelques commandes principales de BorgBackup dans cet article. On se référera à la documentation de BorgBackup [4] pour voir la totalité des commandes.

5.1 Sauvegardes

5.1.1 Initialisation d'un dépôt de sauvegarde « coté serveur »

Pour faire la sauvegarde entre un PC client et un serveur borg distant il faut commencer par créer un dépôt (« *repository* ») de sauvegarde sur le serveur. On peut choisir de créer un dépôt commun pour sauvegarder plusieurs PC, ou bien un dépôt de sauvegarde par poste ou par volume de données à sauvegarder. Ce choix impacte la durée ou les volumes à sauvegarder.

La mise en commun de plusieurs machines dans un même dépôt va améliorer l'espace utilisé par la mutualisation des fragments communs (notamment si on sauvegarde des données similaires). Mais cela va entraîner la nécessaire synchronisation des caches entre le serveur et les postes clients, ce qui va augmenter le temps de sauvegarde.

On crée pour cela un utilisateur sur le serveur distant pour effectuer les sauvegardes (user « borg » par exemple) et on fait un échange de clé ssh entre la machine cliente et le serveur (copie de la clé publique de l'utilisateur qui va lancer la sauvegarde (« dupont », « borg » ou « root ») dans le fichier *.ssh/authorized_keys* de l'utilisateur « borg » sur le serveur).

- Le client initialise un dépôt de sauvegarde distant sur le serveur via ssh, par la syntaxe classique
  - user@nom_du_serveur:/chemin/du/dépôt/::nom_archive
  - exemple : ***borg@monserveur.monlabo.fr:/mnt/nas-borg/sauve-pcml::pcml-2016-12-31***

où /mnt/nas-borg sera une baie de disques montée sur le serveur Borg (en attachement direct ou via NFS).

La commande suivante crée un premier dépôt de sauvegarde :

```
$ borg init borg@monserveur.monlabo.fr:/mnt/nas-borg/sauve-pcml
Enter new passphrase:
```

Lors de l'initialisation, une clé de chiffrement est générée, et la passphrase qui est demandée coté client lors de la création d'un dépôt (borg init) va servir à protéger la clé de chiffrement des sauvegardes. Cette passphrase sera demandée lors de tout accès au dépôt de sauvegarde. Pour automatiser les lancements des sauvegardes par script, on peut la mettre dans la variable d'environnement *BORG_PASSPHRASE*

5.1.2 Sauvegarder un répertoire

Le dépôt ayant été créé, on peut commencer à créer des « archives » de sauvegarde dans ce dépôt :

```
$ borg create -v --stats borg@monserveur.monlabo.fr:/mnt/nas-borg/sauve-pcml::pcml2017-07-13 /home/libes/
```

A la fin de la sauvegarde BorgBackup affiche des statistiques intéressantes pour chaque sauvegarde : notamment la durée de la sauvegarde, le volume total à sauvegarder, le volume sauvegardé après déduplication et compression

```
Archive name: pcml2017-07-13
Time (start): Thu, 2017-07-13 12:44:18
Time (end):  Thu, 2017-07-13 14:37:40
Duration: 1 hours 53 minutes 22.05 seconds
Number of files: 199511
------------------------------------------------------------------------------
                       Original size      Compressed size    Deduplicated
size
This archive:        159.62 GB            159.63 GB            122.42 GB
All archives:        159.62 GB            159.63 GB            122.42 GB

                       Unique chunks         Total chunks

Chunk index:          200063               256783
```

Nous détaillerons ces chiffres un peu plus bas.

5.1.3 Lister les sauvegardes

– On peut bien entendu lister le contenu d'un dépôt de sauvegarde

```
$ borg list --info   borg@monserveur.monlabo.fr:/mnt/nas-borg/sauve-pcml

Enter passphrase for key ssh://borg@139.124.xx.yyy/mnt/provigo-borg/borg-pcml:
pcml-2017-08-30                      Wed, 2017-08-30 02:30:20
...
pcml-2017-09-01                      Fri, 2017-09-01 02:30:16
pcml-2017-09-02                      Sat, 2017-09-02 02:30:18
```

– et obtenir les informations et statistiques de chaque archive donnant les tailles originales à sauvegarder (358.26Gb) , la taille résultante compressée (347.97) et la taille résultante dédupliquée qui a été stockée (132.31MB) dans cette nouvelle archive.

```
$ borg info borg@monserveur.monlabo.fr:/mnt/nas-borg/sauve-
pcml::pcml-2017-09-03
Name: pcml-2017-09-03
Hostname: pcml
Username: libes
Time (start): Sun, 2017-09-03 02:30:19
Time (end):   Sun, 2017-09-03 02:33:32
Command line: /usr/bin/borg create -v --stats --compression
zlib,3 borg@monserveur.monlabo.fr:/mnt/nas-borg/sauve-pcml::pcml-
2017-09-03 /home/libes /opt --exclude /home/*/.cache
Number of files: 520776

          Original size      Compressed size    Deduplicated size
This archive : 358.26 GB       347.97 GB            132.31 MB
All archives:    3.58 TB         3.47 TB            207.34 GB

                       Unique chunks         Total chunks
Chunk index:                  369070              6496746
```

Le but de cet article n'est pas de faire le tour exhaustif de toutes les commandes que vous trouverez dans la documentation officielle de BorgBackup [4]. Le jeu de commande de Borg est assez complet pour manipuler et gérer correctement les sauvegardes.

On peut bien entendu faire un ensemble complet d'actions sur les sauvegardes : les renommer *(borg rename),* détruire *(borg delete)* des archives et des dépôts, élaborer une politique de sauvegarde *(borg prune –keep-daily dd –keep-weekly ww –keep-monthly mm)* en conservant les archives une période donnée, etc.

5.2 Restauration

Afin de montrer la simplicité de la restauration des données, nous donnons les deux manières que propose BorgBackup pour restituer les données sauvegardées :

1. soit en « montant » *(borg mount)* la sauvegarde dans un système de fichiers UNIX local du PC client

```
$ borg mount borg@monserveur.monlabo.fr:/mnt/nas-borg/sauve-
pcml::pcml-2017-09-07  /mnt/restore-borg/

$ ls /mnt/restore-borg/home/libes/Documents/ -dl

où :  /mnt/restore-borg est un point de montage local sur lequel
sera montée toute la sauvegarde demandée
```

2. soit en extrayant *(borg extract)* directement les répertoires et fichiers depuis le dépôt du serveur

```
$ borg extract -v --list  borg@monserveur.monlabo.fr:/mnt/nas-
borg/sauve-pcml::pcml-2017-09-07  home/libes/Documents/CLUSTER


ll ./home/libes/Documents/
total 12
drwxr-xr-x 9 libes libes 4096 juil. 13 12:22 CLUSTER/
```

6 Quelques mesures de performances

Nous donnons ci dessous quelques statistiques de volumétrie et de durée des sauvegardes et les comparons, autant que faire se peut, à un autre logiciel comme BackupPC. Dans tous les exemples que nous donnons, par défaut, BorgBackup chiffre les fragments sauvegardés en AES256.

6.1 Déduplication et compression

La déduplication effectuée au niveau des fragments de fichiers apporte une économie notoire du volume sauvegardé. Celle ci dépend bien évidemment du jeu de données et sera d'autant plus intéressant qu'il y aura des gros fichiers supérieurs à la taille minimale de fragmentation, et comportant statistiquement des séquences identiques. Si on rajoute de la compression sur les fragments de fichier on améliore encore plus le volume sauvegardé.

Dans cet exemple ci dessous sur un PC banal d'informaticien :-), sur une première sauvegarde de 159Go , non compressés, on aboutit à 122Go sauvegardé en dédupliqués… soit un gain de 23 % environ par la seule déduplication des données (1 % de fichiers supérieurs à 2Mb)

```
Time (start): Thu, 2017-07-13 12:44:18
Time (end):   Thu, 2017-07-13 14:37:40
Duration: 1 hours 53 minutes 22.05 seconds
Number of files: 199511
------------------------------------------------------------
              Original size      Compressed size    Deduplicated
size
This archive:        159.62 GB       159.63 GB      122.42 GB
All archives:        159.62 GB       159.63 GB      122.42 GB

                       Unique chunks         Total chunks
Chunk index:                  200063               256783
```

- Sur le même jeu de données, si on rajoute une compression moyenne (zlib,3) on améliore le volume sauvegardé, pour arriver à 109G sauvegardé, (31 % de gain), mais au détriment de la durée de sauvegarde qui augmente de 24 minutes

```
Duration: 2 hours 17 minutes 55.86 seconds
Number of files: 199508
------------------------------------------------------------------
               Original size      Compressed size    Deduplicated
size
This archive:     159.62 GB       142.30 GB           109.32 GB
All archives:     159.62 GB       142.30 GB           109.32 GB
```

- Grâce à la déduplication si on copie des répertoires entiers, ou si on renomme des fichiers identiques, on ne modifie quasiment pas le volume sauvegardé par Borg. En effet, quels que soient les noms de fichiers, l'algorithme de déduplication fait que les fragments de fichiers existants sont déjà dans le dépôt. Dans l'exemple ci dessous on a rajouté 63G (159G+63G=222G) de plusieurs répertoires et fichiers identiques dans le même jeu de données. On voit que par déduplication ces 63Go de données identiques se sont traduits par une sauvegarde de 15MB de données, en 11 minutes. La majeure partie des fragments de fichiers en double existant déjà dans le dépôt.

```
Time (start): Thu, 2017-07-13 20:06:47
Time (end):   Thu, 2017-07-13 20:18:27
Duration: 11 minutes 39.12 seconds
Number of files: 201688
------------------------------------------------------------------
               Original size      Compressed size    Deduplicated
size
This archive:     222.08 GB       222.09 GB           15.72 MB
All archives:     381.70 G        381.72 GB           122.43 GB
```

Cette économie de volume sauvegardé par déduplication est extrêmement variable d'un jeu de données à l'autre selon la nature des données. Nous ne pouvons pas donner de pourcentage d'économie de sauvegarde constant.

6.2 les durées de sauvegardes

La première sauvegarde effectuée par BorgBackup présente une durée "normalement" longue puisqu'il faut remplir le dépôt de sauvegarde une première fois avec de nouveaux fragments, constituer les index pour associer les fragments avec les fichiers, puis compresser, et chiffrer les fragments avant envoi. Mais dès la 2eme sauvegarde, le bénéfice de la mise en commun des fragments de fichiers et leur référencement dans un cache local indexé, est extrêmement intéressant et réduit fortement le volume et la durée de sauvegarde.

BorgBackup procure un nouveau paradigme de la sauvegarde : en effet

- Le travail le plus important est effectué sur le client Borg. La première sauvegarde consiste à remplir le dépôt avec l'ensemble des fragments uniques existants dans le volume initial ;
- Par la suite seuls les nouveaux fragments non présents dans le dépôt sont sauvegardés après une recherche rapide du hash de ces fragments dans un cache local sur le pc client ;

Après « n » sauvegardes, dans un dépôt bien rempli, sur un PC dont le jeu de données évolue lentement, la majeure partie des fragments sont présents dans le dépôt, et donc la durée de sauvegarde des nouveaux

fragments se résume en grande partie à une recherche indexée dans un cache local au PC client. L'envoi de quelques fragments nouveaux au serveur devient ridiculement faible !

Voici un exemple des évolutions des sauvegardes sur un volume de données scientifiques d'un PC d'un chercheur. Après la première sauvegarde pour remplir le dépôt, les ajouts dans les sauvegardes suivantes deviennent très courts :

Vol. a sauver	**Nb fichiers**	**Durée**	**Taille dédupliquée**	**Taille de l'archive complète (15 jours de rétention)**	**Chunks unique/total**
347.38 GB (***1ère***)	768993	***4h 14 min***	103.49 GB	103.49 GB	695181 / 864806
347.39 GB	768990	***4 min***	41.74 MB	103.53 GB	695314 / 1729578
347.39 GB	769012	***5 min***	69.93 MB	103.60 GB	696096 /2594379
414.48 GB (*)	1139525	***47 min***	82.32 MB	103.68 GB	697737 / 3844301
433.09 GB (**)	1271531	***57 min***	93 MB	103.78 GB	699607 / 5230865
433.12 GB	1271544	***6 min***	106.77 MB	103.88 GB	701330 / 6617412
433.17 GB	1271558	***6,5 min***	43.83 MB	104.08 GB	705471/ 10433870
433.96 GB (***)	1291864	***1h 12min***	39.71 GB	143.79 GB	1228119 / 11842271
433.24 GB	1271566	***7 min***	54.16 MB	143.76 GB	1227523 / 10978934

(*) copie de 68 G de fichiers déjà existants : on voit que la duplication de fichiers déjà existants (+le travail du jour du chercheur) se traduit par une sauvegarde de 82M de nouveaux fragments en 47 min

(**) copie de 86 G de fichiers déjà existants renommés : dans ce cas nous avons en plus renommé les fichiers dupliqués sans changer le contenu : ce qui se traduit par 93M sauvegardé en 57 min

(***) modification des 86G de fichiers : traduits par 39G de nouveaux fragment dédupliqués en 1h12

- Nombre de fichiers de taille supérieure à 2 Mb: 1% (sur lesquels la fragmentation s'appliquera)

6.3 Comparaison par rapport à des sauvegardes en rsync avec BackupPC

Nous avons comparé les volumes et durées des sauvegardes effectuées par Borg avec celles obtenues avec BackupPC en rsync, sur des jeux de données identiques ou suffisamment proches.

Voici deux exemples de comparaison sur quelques jours de sauvegarde entre BackupPC et BorgBackup.

i) Un dépôt de données scientifiques sur un NAS #940G

Première sauvegarde de Borg	**Durée**	**Nombre de fichiers**	**Original size**	**Compressed size**	**Deduplicated size**

Archive name: netapp-gis-2017-03-20	959 min	264228	936.49 GB	551.80 GB	434.99 GB

Date	***Volume à sauve-garder***	***Nombre fichiers***	***BackupPC***			***BorgBackup***	
			Durée	**Fichiers déjà existants**	**Fichiers nouveaux : Vol. sauvé.**	**Durée**	**Vol. dédupliqué. sauvé**
2017-09-13	950G	266868	839 min (Full)	949 G	331 M	7mn	123 M
2017-09-14	954 G	266912	13 min (Incr)	543.3 M	84 M	6 min	65 M
2017-09-15	954.2	266923	41 min (Incr)	643.4 M	156 M	6 min	165 M
2017-09-16	954,22	266927	28 min (Incr)	822 M	23	5 min	51 M
2017-09-17	954,22	266927	370 min (Full)	950 G	0,2	4 min	31 M
2017-09-18	954,22	266927	*22 min (Incr)*	0 M	0	5 min	29 M
2017-09-19	954.21	266935	*35 min (Incr)*	5 M	278 M	5 min	118 M
2017-09-20	954.31	266988	10 min (Incr)	287 M	150 M	5 min	81 M

Sur ce premier tableau, on peut donc constater la bonne tenue en volumétrie de BackupPC par rapport à BorgBackup du fait que la majeure partie de la « déduplication » effectuée se passe au niveau fichier car seul 1 % des fichiers dépassent 2M.

En revanche, on peut constater un gain très important en terme de durée de sauvegarde notamment face aux sauvegardes complètes que réalise BackupPC, puisque BorgBackup se contente de détecter et transférer quelques nouveaux fragments en quelques minutes. Cela se traduit également au niveau de la bande passante et de la charge du serveur par une quantité de données manipulées beaucoup plus faible.

ii) Comparaison sur une machine d'un chercheur

Date	***Volume à sauve-garder***	***Nb fic***	***BackupPC***			***BorgBackup***	
			Durée	**Fichiers déjà existants**	**Fichiers nouveaux : Vol. sauvé.**	**Durée**	**Vol dédupliqué**
10/09/207	347 G	768993	9h00	319 G	0	4h 14 min	103 G
14/09/2017 (*)	414 G	1139525	3h20 (Incr)	63 G	0,1 M	47 min	82 M

15/09/2017 (**)	433 G	1271531	5h00 (Incr)	81 G	17 M		57 min	93 M
17/09/2017	433G	1271544	5h20 (Full)	410	0 M		5 min 47	60M
18/09/2017	433 G	1271544	1h18 (Incr)	1,9 M	1,3 M		5 min 58	93 M
19/09/2017	433 G	1271566	18 min	2,7 M	2,5 M		6 min 30	104 M
20/09/2017 (***)	434 G	1291864	10 h	137 M	93 G		1h12	39,7 G

- Sur les 2 premiers cas (*, **) ou on a rajouté un lot de fichiers identiques, on voit que BackupPC est capable de détecter des fichiers identiques déjà existants, renommés ou pas, et ne sauvegarde qu'une faible fraction de cet ajout. Cela s'effectue cependant avec des durées de détection très importantes par rapport à Borg.
- En revanche sur le 3ème cas (***), nous avons modifié 86G de fichiers, on voit que BackupPC est obligé de sauvegarder la totalité de ces nouvelles données en 10 heures, alors que BorgBackup par son mécanisme de déduplication, ne sauvegarde que 39G, en 1h12. On aurait encore amélioré ce ratio en faveur de Borg si la proportion de fichiers supérieurs à 2M (taille de fragmentation) étaient plus élevée.

En conclusion, on s'aperçoit que BackupPC procure des ratio de volumes sauvegardés intéressants et comparables avec ceux de BorgBackup. En effet, la compression de données qu'il met en œuvre (zlib,3 par défaut) est la même que celle que nous avons utilisée avec BorgBackup. Par ailleurs BackupPC effectue une déduplication au niveau de fichiers entiers, qui est suffisante dans de nombreux cas (petits fichiers, peu de modifications etc.) en terme de volumétrie. Dans nos tests nous avons en effet noté que sur des jeux de données (pc , volume scientifique…) la proportion de fichiers supérieurs à 2M est de l'ordre de 1 a 2%).

Même si les taux de réduction apportés par BackupPC sont comparables à ceux effectués par Borg (pour les tests que nous avons faits), la déduplication effectuée par BorgBackup au niveau des fragments de fichiers ne peut de toute façon qu'apporter une amélioration dans plusieurs cas de figure : le cas le plus favorable étant de gros fichiers qui changent peu).

Dans tous les tests que nous avons faits les résultats de volumétrie effectués par BorgBackup ont toujours été légèrement meilleurs que ceux issus de BackupPC

En revanche, pour ce qui concerne les durées de sauvegardes, le gain en faveur de Borg est flagrant : le paradigme de sauvegarde introduit par Borg procure :

- des durées de sauvegarde quotidiennes très nettement inférieures,
- une économie de bande passante réseau vers le serveur,
- une moindre sollicitation du serveur de sauvegarde qui peut fonctionner sur une machine virtuelle avec des ressources faibles en processeurs et mémoire.

7 Conclusions – Avantages et Inconvénients

7.1 Avantages

Après quelques mois d'utilisation, de divers tests et de mise en exploitation, voici en résumé les qualités que nous avons perçues dans l'utilisation de BorgBackup :

- simplicité et rapidité d'installation : la configuration par défaut est suffisante ;
- simplicité d'utilisation : un jeu de commandes facile à utiliser ;
- stabilité : pas de plantage ou résultat erroné. Une mise en exploitation sans problème ;
- bonne documentation et un forum actif ;
- logiciel libre, pas de format propriétaire dans la sauvegarde et une équipe de développement active ;
- Très bonnes performances:
  - forte réduction du volume sauvegardé par l'alliance de la déduplication et de la compression ;
  - forte réduction des durées de sauvegardes ;
  - économie de bande passante réseau : le serveur ne reçoit que les nouveaux fragments de fichiers ;
  - Faible consommation de ressources du serveur : BorgBackup procure un transfert de la majeure partie de l'intelligence de la sauvegarde du coté client, le serveur Borg est assez peu sollicité, et peut fonctionner sur une machine virtuelle modeste (2G de RAM). Alors qu'avec un serveur BackupPC dans notre contexte nous avons dû consacrer une machine physique dédiée plus puissante.
- Sécurisation des sauvegardes par chiffrement en AES256 ;
- trois mode de compression possible ;
- reprise sur erreur : les interruption des sauvegardes ne posent pas de problème, chaque sauvegarde partielle est réutilisée dans la suivante ;

7.2 Inconvénients

- Pas d'interface web pour gérer les différentes sauvegardes effectuées de manière centralisée. La solution est de surveiller chaque sauvegarde sur les PC clients à l'aide d'un script qui remonte l'état des sauvegardes à un serveur de surveillance comme Icinga, Nagios, Zabbix, Xymon
- ne fonctionne que sous Linux, FreeBSD, et MacOS X
- peut fonctionner sous Windows en installant cygwin (non testé).

Bibliographie

[1] Solution de sauvegarde dédupliquée BorgBackup
http://borgbackup.readthedocs.io/en/stable/index.html

[2] Installation de BorgBackup : https://borgbackup.readthedocs.io/en/stable/installation.html#source-install

[3] Définition de la déduplication https://fr.wikipedia.org/wiki/D%C3%A9duplication

[4] les commandes de borgbackup : https://borgbackup.readthedocs.io/en/stable/usage.html

[5] Solution de sauvegarde UrBackup https://www.urbackup.org/

[6] backuppc : http://backuppc.sourceforge.net/BackupPC-4.1.3.html

[7] Schémas de déduplication : http://linux.arcticdesign.fr/la-deduplication-vous-fait-gagner-de-la-place/deduplication-schema/

[8] https://doc.ubuntu-fr.org/borgbackup

[9] Taille des fragments: https://borgbackup.readthedocs.io/en/stable/internals.html#chunks