Bacula, une solution de sauvegarde par le réseau

Matthieu Guionnet
LIX Laboratoire d'Informatique de l'École Polytechnique (UMR 7161)
route de Saclay
91120 Palaiseau

Résumé

« Il arrive la nuit et absorbe l'essence vitale de vos ordinateurs » tel est le slogan du projet Bacula, contraction de backup et Dracula. Bacula est une solution libre de sauvegarde par le réseau sur le modèle client/serveur. Le logiciel est développé depuis 2000 et dispose d'un grand nombre de fonctions parmi lesquelles le multi-volume et le multi-support. Il fonctionne sur tout système respectant le standard POSIX et avec un client spécifique sur Windows qui utilise le Volume Shadow Snapshot (VSS) propre à ce système. Il met en œuvre un nombre assez important de démons affectés à des tâches précises et des mécanismes de gestion à adopter qui n'en facilitent pas toujours une approche rapide.

Mots clefs

sauvegarde, réseau, logiciel libre, gpl, client/serveur, unix, linux, windows, disques durs, bandes, LTO, auto-changeurs

1 Introduction

1.1 Présentation

La mise en œuvre d'une solution de sauvegarde n'est jamais simple. Si celle-ci doit, de plus, se faire au travers des réseaux de machines et de systèmes d'exploitation nombreux et variés, les solutions deviennent alors plus rares. Parmi le peu de solutions libres disponibles, il existe Bacula. Celle-ci n'est pas suffisamment connue malgré ses nombreuses qualités. Cet article tentera donc de corriger cela en essayant de vous en faciliter l'accès, au moins pour l'essayer, au mieux pour l'adopter.

Bacula 8 se présente comme une solution de sauvegarde orientée « entreprise », soutenant la comparaison avec les logiciels de sauvegarde propriétaires reconnus.

Les caractéristiques présentées à la suite de l'historique du projet sont limitées en nombre pour cet article. Il convient de regarder l'imposante documentation fournie pour en découvrir de nombreuses autres.

Les 720 pages, dont 121 à destination des développeurs, peuvent peut-être représenter un blocage à l'évaluation de Bacula pour un projet de sauvegarde centralisée. C'est pourquoi, dans cet article, une approche générale du fonctionnement de Bacula sera présentée. L'installation et la configuration des principaux composants de Bacula seront traitées. Une section sera également consacrée aux supports de sauvegarde.

1.2 Histoire du projet

Le projet a été commencé en janvier 2000 par Kern Sibbald.

Le projet est transféré sur sourceforge 8 et sort pour l'occasion la version 1.16, le 14 avril 2002. Un peu plus de 4 années après, le 26 juin 2006, la version 1.38 devient disponible. Cette version marque une étape dans le but de favoriser l'adoption de Bacula que ce soit par les utilisateurs ou bien par les développeurs, en utilisant les outils courants des autres projets comme le format *tar* compressé avec *gzip* ou encore les utilitaires GNU, *configure* et *make*.

Le projet passe en novembre de cette même année sous la protection légale de la Free Software Foundation Europe 8.

Le développement s'accélère donc et, moins d'un an après, en janvier 2007, une réécriture du projet est disponible, ce sera la version 2.0.

À partir de la série 2, le chiffre suivant celui indiquant la série renseigne sur la stabilité du projet. Si ce chiffre est pair, c'est que la version est considérée comme stable. Sinon c'est une version de développement.

Les versions 2.2 et 2.4 sont sorties respectivement en septembre 2007 et en juin 2008. Chacune a apporté son lot d'améliorations.

La version de développement 2.5 avait 208 000 lignes modifiées par rapport à la version 2.4, soit quasiment le nombre de lignes du projet. Cette différence dans les sources a amené les mainteneurs du projet à passer à la série 3. Celle-ci a commencé le 9 avril 2009 avec la version 3.0.0. Depuis, des versions mineures sont sorties pour corriger des bogues. La dernière disponible au moment de la rédaction de cet article est la 3.0.3, sortie le 18 octobre 2009. L'article se concentrera sur cette dernière série.

Kern Sibbald, toujours mainteneur et coordinateur du projet, a fondé, fin 2008, la société bacula system SA 8. Elle apporte un support au projet mais en optant pour une numérotation différente. La dernière version est la 2.6.

Pour la petite histoire, Kern Sibbald est co-fondateur de la société AutoDesk en 1989, société particulièrement connue pour son logiciel propriétaire AutoCAD. Depuis 1992, fin de son activité dans cette société, Kern a été essentiellement actif dans le monde du logiciel libre, une bien belle conversion.

1.3 Principales caractéristiques

Bacula est un ensemble de programmes sous licence libre GNU Public Licence, permettant à un administrateur système de contrôler la sauvegarde, la restauration et la vérification de données informatiques par le réseau.

Les clients peuvent être de types variés. Des pré- et post- scripts aux travaux de sauvegardes, voire de restauration, permettent une flexibilité importante et ouvrent des choix limités uniquement par l'imagination des administrateurs systèmes. Le premier exemple est bien entendu la création d'instantanés des systèmes de fichiers clients le permettant, comme Btrfs pour les clients sous Linux. Les clients Windows profitent directement de la fonction d'instantanés offerte par VSS.

Le nombre de clients peut être très important et ces derniers sont potentiellement répartis sur de larges réseaux.

Les systèmes d´exploitation pouvant faire tourner la partie serveur du projet sont beaucoup moins nombreux. Linux est celui préféré pour cette tâche et le reste de l'article sera basé sur ce choix.

Bacula est le seul projet libre de sauvegarde à baser son catalogue sur une base de données relationnelle de type SQL. Cet élément est donc le seul à n'être pas écrit par les développeurs du projet. Le choix de la base est laissé à la discrétion de l'utilisateur lors de l'installation.

Les supports de sauvegardes sont variés. Les disques durs et bandes magnétiques sont bien évidement supportés, avec, en option, des bibliothèques robotisées pour ces dernières. Mais il est aussi possible d'utiliser des CD ou des DVD.

Les sauvegardes pourront ensuite être migrées d'un support à un autre. Toutes les informations permettant de retrouver les données souhaitées se trouvent ainsi stockées dans le catalogue.

2 Architecture et Configuration rapide

2.1 L'architecture de Bacula

Comme entrevu précédemment, le projet Bacula se base sur le principe de client-serveur. Les tâches sont séparées. Une tâche correspondant à un démon et toutes les communications s'effectuant sur le réseau en TCP/IP, il est facile de répartir les tâches sur des machines distinctes.

La Figure 1 présente l'architecture d'une installation de Bacula réduite à un seul client. Elle montre aussi les interactions des éléments entre eux.

La pièce maîtresse du système est, comme son nom l'indique, le directeur (bacula-dir). Son nom précise aussi son unicité. C'est à partir de celui-ci que les travaux de sauvegarde sont configurés et lancés. Il contacte les clients, File Daemon (FD), pour qu'ils envoient leurs données sur les volumes du Pool dirigé par le service chargé des supports de sauvegarde, Storage Daemon (SD). Il enregistre tous ces événements dans le catalogue.

Le catalogue est une base de données relationnelle SQL. Le directeur est le seul à y avoir accès. Il n'y a qu'un catalogue et le choix de la base SQL est laissé à l'administrateur entre les bases MySQL, PosgreSQL ou SQLite. Cette dernière n'est pas une base SQL mais une librairie écrite en C qui stocke les données dans un fichier local. Elle est très pratique, car rapide à mettre en œuvre et ne nécessite pas d'administration. Mais elle est déconseillée pour la mise en production et doit être limitée aux tests.

Le catalogue est la seule partie non écrite par les développeurs de Bacula. Le choix de la base devrait augmenter, car depuis la version 3 de Bacula, ce dernier peut se reposer, pour les connexions SQL au catalogue, sur la librairie libdbi. Celle-ci se présente comme une interface SQL permettant de cacher les bases réellement utilisées et elle a comme objectif de multiplier les bases supportées dans le futur.

La taille de la base de données peut être maîtrisée en déclarant les durées de rétention d'une partie des données qui y sont stockées (File Retention, Job Retention et Volume Retention).

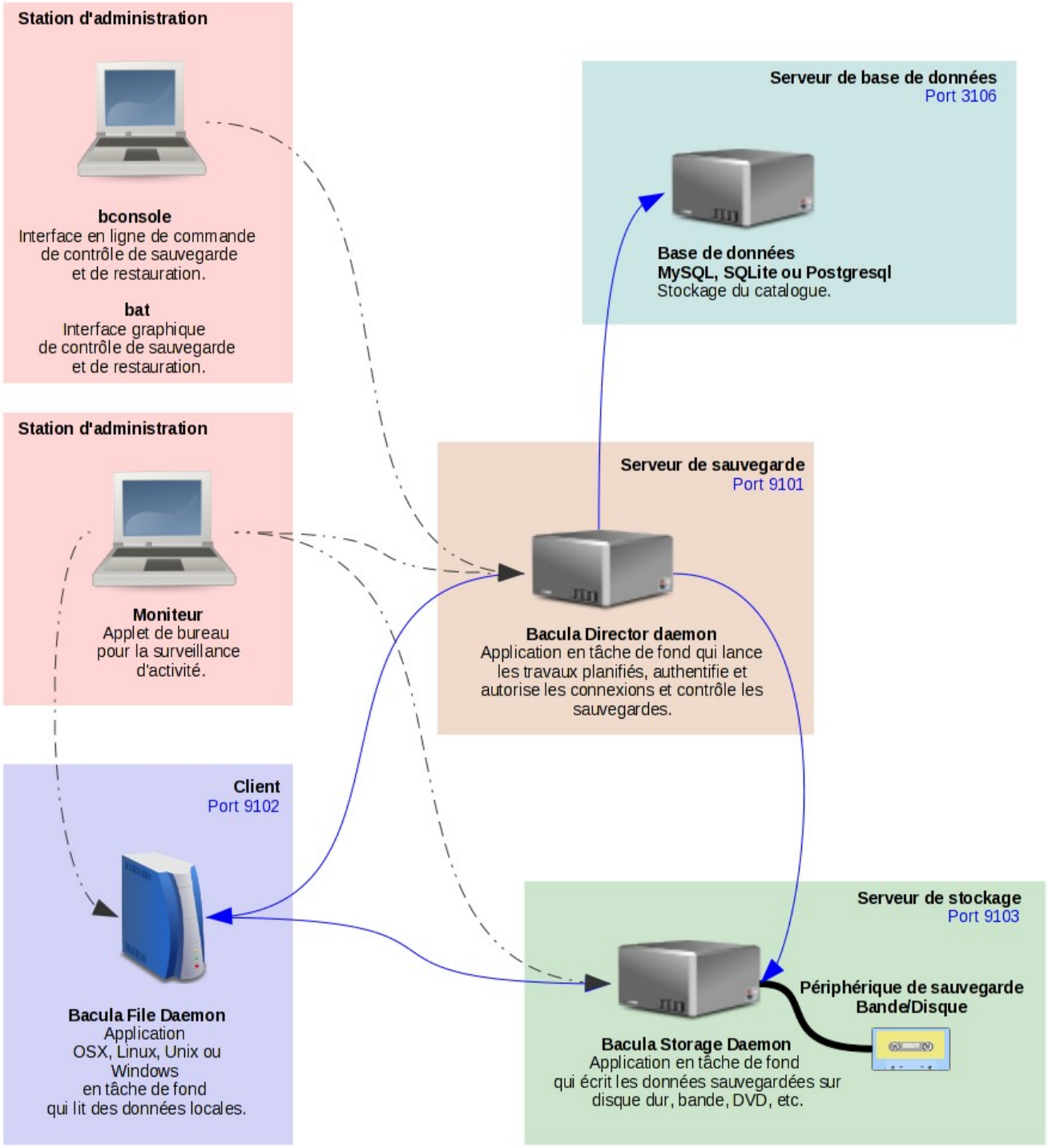

Figure 1 - Interactions des applications dans Bacula.

Il y a au moins un service de gestion du stockage (SD) dans un ensemble défini de sauvegarde. Il peut avoir plusieurs périphériques de stockage configurés. Il gère, à la demande du directeur (Dir) et sur un des supports que celui-ci a choisi, l'écriture et la lecture des données venant d'un client.

Le ou les clients (FD) attendent les ordres du directeur. Celui-ci mettra en place, pour l'exécution d'un travail (Job), la communication entre un client et un SD. Le travail sera le plus souvent une sauvegarde, une lecture de données du client et l'écriture de celles-ci par le SD sur un support. Mais un travail peut aussi correspondre à une restauration de données et, dans ce cas-là, les rôles pour la lecture et pour l'écriture seront inversés. Les restaurations peuvent se faire sur n'importe lequel des clients connus du directeur.

Dans chacun des cas, l'ensemble des paramètres formant le travail est à la discrétion du directeur. Parmi ces paramètres, il y a le calendrier d'exécution du travail, ainsi que le niveau de sauvegarde.

Les niveaux de sauvegarde sont au nombre de 3 : la sauvegarde totale ou Full, la sauvegarde incrémentale ou Incremental et la sauvegarde différentielle ou Differential.

- la sauvegarde totale correspond à la sauvegarde de l'ensemble des données ;
- la sauvegarde incrémentale consiste à sauver les données modifiées ou nouvellement créées depuis la dernière sauvegarde quel que fut son niveau ;
- la sauvegarde différentielle, reprend le principe de la sauvegarde incrémentale sauf que les différences sont obtenues avec la dernière sauvegarde totale effectuée.

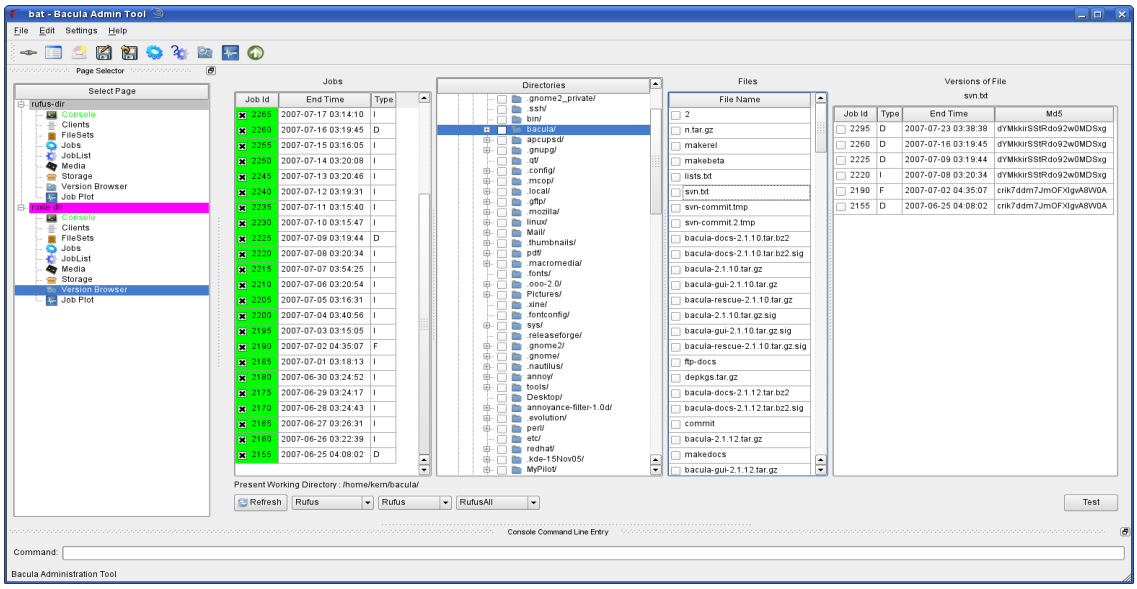

Figure 2 - Capture d'écran de Bat.

La console sert à contrôler le directeur. Elle peut être de type texte ou bien graphique. Il existe aussi des projets avec pour objectif une interface de type « web », mais les 2 premiers types sont les plus faciles et donc plus courants. La version graphique qui est actuellement la plus développée est Bat (Bacula Admin Tool), Figure 2.

Le contrôle par ligne de commande dispose de la console la plus stable avec le programme bconsole. De plus, celui-ci est aussi le plus facile à compiler.

Un ou plusieurs moniteurs peuvent être installés pour surveiller, depuis un bureau Gnome, KDE ou tout autre environnement freedesktop, l'activité des services souhaités. L'article n'ira pas plus loin dans la présentation de cette fonctionnalité.

2.2 Installation(s) et configuration

Les sources et binaires du projet sont hébergés sur la plateforme Sourceforge.

L'installation depuis les fichiers sources au format *tar.gz* est plutôt aisée. Le projet base sa construction sur les outils connus *autoconf* et *automake*. Les options de compilation sont listées avec la commande ./configure –-help lancée dans le répertoire racine du projet après sa décompression.

Suivant les options choisies, il faudra plus ou moins de bibliothèques pour satisfaire la compilation du projet.

Pour ces raisons, il est conseillé d'utiliser directement les binaires d'installation disponibles sur le site pour les clients sous Windows.

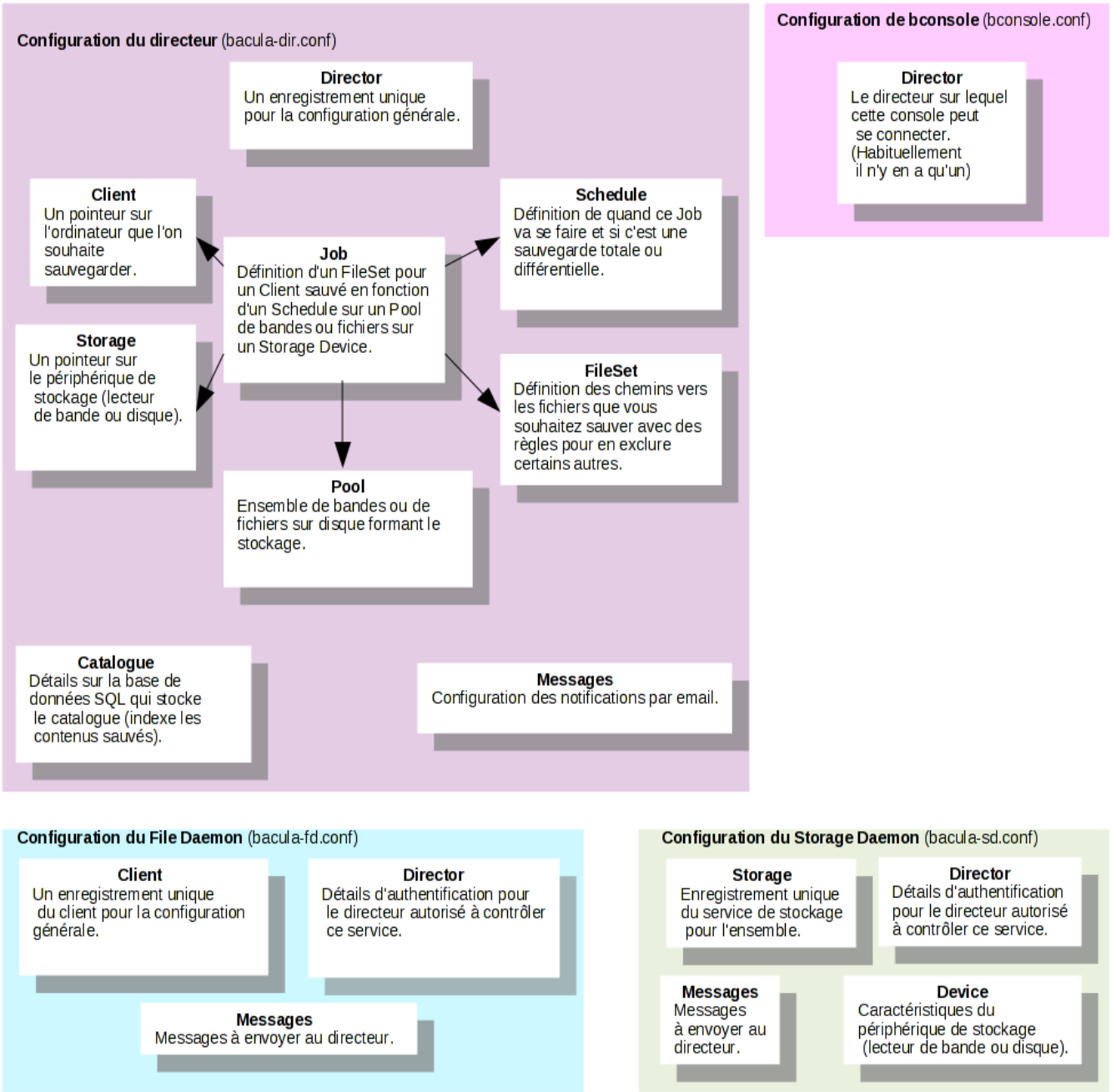

Figure 3 - Définitions des objets dans les configurations de Bacula.

Certaines distributions Linux, comme Debian, proposent leurs paquets (packages) prêts à être installés. Sur Redhat et CentOS, l'utilisation du dépôt EPEL permet d'obtenir une version packagée de Bacula.

Mais attention, les versions sont souvent vieilles et ne permettent pas de profiter des dernières avancées. La version de Bacula sur Debian Lenny est la 2.4, tandis que sur le dépôt EPEL, la version disponible est la 2.0, qui est encore plus ancienne.

Ces versions peuvent servir éventuellement à installer les clients pour ces distributions. En effet, le directeur et le démon de stockage en version 3 acceptent aussi les clients en 2.x.

Des binaires, *deb* pour Debian et *rpm* pour Redhat et les autres distributions les acceptant, sont disponibles pour installer les programmes dans une version récente.

Que vous choisissiez les sources ou les binaires, il vous faudra décider de la base de données relationnelle avec laquelle le directeur stockera et gèrera son catalogue. SQLite est la solution la plus simple. Même si elle n'est pas conseillée pour la mise en production, elle est bien adaptée à l'évaluation.

Certaines options sont difficiles à compiler sur certaines distributions et donc à installer. Parmi celles-ci, il y a le programme Bat qui nécessite, dans sa dernière version, une librairie Qt plus récente que celle fournie dans la distribution CentOS. Même si l'on peut très bien se passer de Bat et se contenter de bconsole, avoir une interface graphique peut rassurer certaines personnes. Une solution rapide est d'utiliser une autre distribution pour y installer uniquement le service manquant. Ainsi un ordinateur sous Fedora pourra fournir Bat, pour se connecter à un directeur (Dir) lancé sur une autre machine sous CentOS.

Après l'installation, les fichiers de configuration des services seront présents sous Linux dans le répertoire /etc ou /etc/Bacula.

La Figure 3 présente les objets présents dans les fichiers de configuration. Les flèches indiquent les dépendances nécessaires à la définition d'un travail (Job).

La bonne déclaration des objets dans ces fichiers n'est pas suffisante pour avoir un système de sauvegarde fonctionnel. Il sera nécessaire d'utiliser une console pour labelliser des supports de stockage et les associer à un Pool. La relation d'une sauvegarde (Job) avec un support passe ainsi par un Pool.

2.3 Les supports

Bacula prend en charge plusieurs supports et les associe entre eux dans des ensembles (Pool). Ceux-ci ont pour principale fonction de permettre de sauver des données qui dépassent le volume d'un support.

Parmi eux, se trouvent les très en vogue disques durs qui profitent de la forte baisse des prix et de l'augmentation de leur capacité. Leur rapidité élevée et leur temps d'accès faible en font un support populaire. Les constructeurs vont jusqu'à proposer des baies de disques durs émulant un lecteur de bande associé à une bibliothèque robotisée.

Les CD et DVD sont un autre choix. Mais ce sont des solutions peu utilisées car la volumétrie et le débit sont faibles. Leur intérêt se trouve souvent dans la prise en compte des problématiques liées à l'archivage légal, bien que dans ce cas, les bandes magnétiques LTO proposent aussi une solution avec la technologie WORM (Write Once Read Many).

Les bandes magnétiques, ou bandes, forment un ensemble qui appartient à l'histoire de la sauvegarde informatique. Elles sont synonymes de sauvegarde dans l'esprit de beaucoup d'informaticiens. Et bien évidement, Bacula les prend en charge. Les bandes sont en perte de vitesse par rapport aux disques durs. Elles ont souffert d'un nombre important de formats de lecteurs, cartouches et autres cassettes. Les prix sont longtemps restés élevés et la pérennité de ces formats est difficile à évaluer.

Mais les temps changent. Le format Linear Tape-Open (LTO), issu d'un consortium, a finalement pris le dessus sur les autres formats propriétaires. Les raisons sont notamment :

- les performances excellentes. La quatrième version, LTO-4, sortie en 2007, offre une capacité native de 800 Go, un débit de 120 Mo/s et l'ajout de chiffrement matériel (256-bit AES-GCM). La capacité peut doubler dans les cas favorables avec l'activation de la compression au niveau du lecteur ;
- les prix des lecteurs et des bibliothèques robotisées qui ont chuté. Le lot n°4 du marché établi par le Groupement Recherche[1] illustre ces chutes. Pour les cartouches, les écarts de prix peuvent être très importants. Il est facile de trouver des cartouches de marques connues parmi les moins onéreuses ;
- « l'écologie », élément important dans la reconquête des cœurs des administrateurs. Un ensemble de 48 bandes LTO-4, formant 38 To de stockage en natif dans une bibliothèque, ne consomme presque rien en électricité (le lecteur et le chariot de transfert). C'est à comparer à un système équivalent en capacité de stockage composé de disques durs.

[1]Groupement d'établissements publics de recherche français, des EPST et des universités principalement, afin de définir et de contracter des marchés de matériels informatiques.

Bien évidement, tout n'oppose pas les bandes magnétiques aux disques durs. Chacun a son utilité. Mais il est tout indiqué de commencer son système de sauvegarde sur des supports bandes magnétiques pour ensuite ajouter des disques durs et utiliser les 2 types dans des cas comme ceux présentés dans le chapitre suivant.

Malgré tout, le format LTO a des défauts. L'un des principaux, surtout depuis la version 4, est la fragilité des lecteurs. Ceux-ci sont prévus pour fonctionner à la vitesse de 120 Mo/s. Or, ce débit est difficile à obtenir des clients Bacula dans bien des cas. Pour compenser, le lecteur passera son temps à alterner fréquemment les états écriture et arrêt dans l'attente de données. Ce phénomène entraîne une usure prématurée du lecteur.

Heureusement, Bacula propose une directive (SpoolData=yes) à placer dans la définition des Job, c'est-à-dire sur le directeur. La section device définie sur le Storage Daemon (SD), sur lequel pointe le Job, devra alors contenir les champs MaximumSpoolSize, MaximumJobSpoolSize et SpoolDirectory. Ces champs permettent de mettre en place un cache dans un fichier sur un disque dur local du SD. Ainsi, le lecteur écrira-t-il les données depuis le cache local. Celui-ci compensera donc la faiblesse des débits généralement observée avec les clients (FD). La durée de vie du lecteur sera alors conservée et la fenêtre de sauvegarde réduite.

3 Possibilités et stratégies de sauvegardes

Après une installation rapide telle que celle présentée dans le chapitre précédent, il convient de voir les modifications et les ajouts éventuels qui seraient bénéfiques à votre propre cas.

Si la « confiance » dans le réseau sur lequel circuleront les échanges de sauvegarde est faible, un premier ajout possible, est d'augmenter la sécurisation des échanges avec la mise en place de TLS (Transport Layer Security). Les communications seront alors chiffrées, assurant ainsi leur confidentialité.

Une aide pour la configuration des pare-feux est disponible et il est envisageable, voire conseillé, de filtrer pour réduire les connexions à l'ensemble du système déjà mis en place.

Le calendrier (schedule) des sauvegardes est proposé par défaut à l'ensemble des travaux suivant un plan traditionnel :

- Full, le premier dimanche du mois ;
- Differential, tous les dimanches ;
- Incremental, tous les jours.

Ce calendrier convient dans la plupart des cas. Mais il peut s'avérer utile d'en choisir un autre, si par exemple le nombre de restaurations est relativement élevé. Dans ce cas, afin d'accélérer la récupération des données, il peut être judicieux d'effectuer les sauvegardes selon le plan suivant :

- Full, une fois par semaine, le dimanche par exemple ;
- Incremental, tous les jours.

Bacula donne la possibilité de lancer des scripts à effectuer avant et/ou après chaque travail. L'utilisation de ces scripts la plus souvent mise en avant est la sauvegarde cohérente d'une base de données 8. Par défaut, il y a une sauvegarde du catalogue de Bacula par lui même, en exportant la base SQL sous la forme d'un fichier texte sur le disque dur local, et une fois ce dernier enregistré sur un volume, il est supprimé.

Depuis la version 3 de Bacula, il existe Accurate, une nouvelle option disponible dans la définition d'un Job. Il suffit de rajouter une ligne Accurate=yes dans cette dernière pour la rendre active. Cette option est prise en compte pour les sauvegardes de niveau Differential et Incremental. Ainsi, Bacula ne se contentera plus seulement des attributs de date de création et de modification d'un fichier pour déterminer la différence avec la précédente sauvegarde. Une liste des précédents fichiers sauvés sera créée afin que le client (FD) et son directeur (Dir) puissent suivre les fichiers supprimés et ceux anciens qui pourraient avoir été rajoutés au système de fichiers surveillé. Ce travail supplémentaire nécessite plus de ressources processeur et mémoire sur les 2 services (SD et Dir). En contre partie, la restauration sera plus précise (accurate, en anglais).

Il conviendra aussi de juger de la pertinence des possibilités de migrations de sauvegardes déjà effectuées. Cette fonctionnalité permet de bouger les données d'un Pool à un autre.

Un exemple fréquent est d'avoir 2 Pools, le premier étant composé de disques durs et le second de bandes. On utilise le premier lors des sauvegardes régulières. Les temps d'accès des disques durs sont intéressants pour effectuer les restaurations qui ont lieu dans des temps proches de la sauvegarde (souvent inférieurs à la semaine). Des migrations pourront être configurées pour que

les données précédentes passent sur le deuxième Pool. Ce dernier permet d'avoir une durée de conservation plus longue des données en économisant de l'argent et de l'électricité.

Un autre exemple d'utilisation des migrations de travaux est la possibilité de les utiliser pour dupliquer les sauvegardes et les répartir géographiquement dans le but de réduire les pertes en cas de sinistre.

Bacula offre beaucoup d'autres possibilités, la documentation est heureusement présente pour vous faciliter l'exploration.

4 Perspectives

Le développement de Bacula a eu un rythme continu. Sa vitalité a toujours été importante malgré sa faible visibilité.

La création de l'entreprise Bacula Systems apporte un support par abonnement. Des stages sont maintenant disponibles auprès de celle-ci ou des partenaires régionaux comme la société Mandriva pour la France.

L'acquisition des compétences est ainsi facilitée comme cela existe déjà pour des logiciels propriétaires concurrents.

La plupart des développeurs prenant part à cette nouvelle société, celle-ci pourrait éventuellement ralentir le développement de la version communautaire du projet.

Mais l'histoire du projet, les dernières versions mineures disponibles (de la 3.0.0 à la 3.0.3) et surtout la trajectoire du principal mainteneur, semblent garantir un équilibre entre le projet libre et l'entreprise de support.

L'avenir se présente bien pour les deux.

Bibliographie

[1] http://www.bacula.org/fr

[2] http://sourceforge.com

[3] http://mailman.fsfeurope.org/pipermail/press-release/2006q4/000161.html

[4] http://www.baculasystems.com

[5] http://wiki.bacula.org/doku.php?id=application_specific_backups