Le projet DiStoNet : le stockage distribué du cluster au poste de travail

Emmanuel Quémener
Centre Blaise Pascal - Laboratoire de l'Informatique du Parallélisme
ENS-Lyon - 15 parvis Descartes - BP 7000 69342 Lyon Cedex 07 - FRANCE

Loïs Taulelle
Pôle Scientifique de Modélisation Numérique
ENS-Lyon - 15 parvis Descartes - BP 7000 69342 Lyon Cedex 07 - FRANCE

Résumé

Dans nos établissements, nous sommes confrontés à un dilemme : d'un côté, pour les données informatiques, les stockages, traitements, sauvegardes et archivages sont exposés à une explosion de leurs volumes. De l'autre, la chasse est lancée à une réduction des coûts informatiques. Victorieusement, ces dernières années, la virtualisation a rationalisé l'usage des matériels centraux pour offrir plus de services avec le même budget matériel. Néanmoins, il demeure un territoire où les équipements restent sous-utilisés : les postes de travail. Comment, dans un souci d'efficacité, agréger ces éléments disséminés sur un site ? Tel est l'objectif du projet DiStoNet. L'ambition est à la hauteur du gain espéré : le coût du disque dur dans une infrastructure de stockage dédié (SAN) ne représente que 10% du coût total de la solution.

De fait, tout projet nécessite un cadre : l'approche intégrateur ou COTS (des briques logicielles existantes et éprouvées) et les outils ouverts (les composants choisis Open Source). L'objectif étant de bénéficier de l'espace de stockage inoccupé des postes de travail, la progression ira de l'environnement le plus maîtrisé (nœud de cluster) au plus inattendu (espace disque sur équipement itinérant), en passant par la cible essentielle : les postes libre-service et les stations de travail.

Cinq approches seront présentées : deux appartenant au « mode bloc » et trois appartenant au « mode fichier ». Chacune ayant fait l'objet d'une implémentation sur des équipements au CBP à l'ENS-Lyon, leurs facilités de déploiement et d'administration, la sécurité associée aux données (suivant ses 4 piliers) seront développées sur la base de ce retour d'expérience de plusieurs mois.

Mots clefs

Stockage distribué, iSCSI, AoE, GlusterFS, XtreemFS, CephFS, Global File System

1 Contexte : besoins croissants & ressources inexploitées

Le stockage, intégrant sa sauvegarde et son archivage devient, pour nos institutions, le principal défi informatique de ces prochaines années : la recherche, l'enseignement et la diffusion des savoirs voient leurs besoins exploser. Techniquement, face à des capacités de disques durs de nos postes de travail toujours en inflation, les infrastructures centralisées se bâtissent autour d'équipements matériels et logiciels spécifiques, avec des ratios de prix compris entre un et deux ordres de grandeur supérieurs pour des volumes identiques.

La vision que nous nous proposons de développer se construit autour d'une simple constatation : les ressources des postes de travail sont sous utilisées. En effet, les messageries électronique ou instantanée, les agendas, les espaces de partages sont, très généralement, déportés sur des serveurs. L'usage du poste de travail se résume ainsi en l'utilisation de clients d'applications distantes. Si nous nous interrogeons sur son usage, le constat est accablant : le ratio entre le temps réel d'utilisation d'une machine (en considérant un temps de travail réglementaire) et le temps pendant laquelle une machine « allumée » en permanenceest d'environ 20%. Puisque le maître mot de ces dernières années est la rationalisation, pourquoi ne pas envisager d'exploiter le mieux possible ces ressources inutilisées, lorsqu'elles ne le sont pas par leurs principaux utilisateurs ?

De plus, la nature des espaces de stockage est diversifiée : des éléments de grappe de calcul disposant de disques largement sur-dimensionnés (ne portant que le système et un espace temporaire), des postes de salles de travaux pratiques, des postes de travail disséminés dans les entités et, enfin, des postes restant sous la responsabilité des utilisateurs. C'est uniquement sur les espaces de stockage disque que nous nous focaliserons et, sur leur usage distant, sous la forme d'une DIStribution du STOckage eN réseau ETendu (ou DIStributed STOrage NETwork).

Abordons maintenant le stockage dans nos établissements et la nature des accès distants aux données. Généralement, deux modes différencient l'accès distant à nos données : le « mode fichier » et le « mode bloc ». Si les partages distants forment plutôt des arborescences, les infrastructures de stockage dédiées (SAN) ont généralisé l'accès en « mode bloc » : le volume distant dispose de toutes les caractéristiques d'un volume interne sous forme de disque.

Comme environnement à notre étude, nous avons choisi de nous placer dans un cadre exclusivement Open Source, de manière à ne pas quitter une addiction (aux technologies de SAN propriétaires) pour tomber dans un autre (aux technologies de stockage réparti toutes aussi propriétaires). Nous suivrons donc ce paradigme : tout ce qui a été expérimenté ici est reproductible sans coûts de licences logicielles par quiconque. Pour limiter les opérations de compilation et d'installation, nous avons utilisé un système d'exploitation doté d'une vaste bibliothèque de composants et reconnue pour ses critères de « qualité » : la Debian Linux.

Fort de ces spécifications et de ce cadre applicatif, nous avons effectué une étude en partant des environnements les plus maîtrisés (réseau privé de calculateurs scientifiques) jusqu'aux environnements où, au final, l'administrateur de l'espace de stockage dispose de peu de contrôle.

2 Un florilège d'approches, mais seulement cinq étudiées

Parmi les protocoles de « mode bloc » que nous avons étudié figurent le iSCSI et le AoE (*ATA over Ethernet*). Nous n'avons pas évalué DRDB[1], existant depuis plusieurs années dans le domaine de la haute disponibilité sous Linux : ce dernier permet, en effet, une duplication synchrone de disque à travers le réseau. Nous avons préféré étudier les implémentations logicielles et Open Source de deux des protocoles utilisés dans les SAN : le iSCSI utilisant le protocole IP pour ses transferts et le AoE, beaucoup plus bas niveau, utilisant Ethernet.

Parmi les protocoles de « mode fichier » que nous avons étudiés figurent GlusterFS, XtreemFS et CephFS. Nous avons exclu de cette étude d'autres systèmes tels que Lustre, GFS, OCFS2 (permettant plutôt d'accéder à un même volume de machines différentes) ainsi que les solutions propriétaires comme GPFS.

GlusterFS[6] est un projet OpenSource encore méconnu. Il existe trois modes de fonctionnement : avec agrégation (équivalent du JBOD), avec réplication (équivalent de RAID1) et avec accès parallèle (équivalent de RAID0). GlusterFS propose également l'équivalent d'un

« LiveCD » permettant de créer, à partir de rien, une infrastructure SAN rapidement. Nous verrons à quel prix...

XTreemFS[7] est issu de la recherche sur les grilles et constitue une part d'un projet plus vaste, XtreemOS. Comme GlusterFS, son installation est aisée pour une utilisation rapide. Il intègre depuis peu des mécanismes de réplication.

CephFS[8] est apparu récemment dans l'archive standard du noyau Linux (à partir du 2.6.34) mais demeure pour l'heure expérimental. Suggéré comme fonctionnant efficacement avec BtrFS (le successeur de Ext2/Ext3/Ext4), intégrant des mécanismes de duplication et une sécurité accrue, il constitue certainement une solution d'avenir (si la masse critique de ses utilisateurs lui assure sa pérennité).

3 Mise en œuvre et administration

3.1 « Mode bloc » : le disque distant ou un composant parmi d'autres

Les protocoles AoE et iSCSI sont assez simples à mettre en œuvre dans notre contexte OpenSource : les composants sont présents sous forme de paquets dans de nombreuses distributions (ici la Debian version Squeeze).

Côté serveur ou « cible », l'AoE est mieux intégré que l'iSCSI (qui demande cependant une compilation de son module).

Pour AoE, les composants intégrés dans les AoeTools[2] sont vblade et sa version persistante vblade-persist. La mise à disposition d'un disque SDA sur l'interface réseau eth1 avec les identifiants majeur et mineur 0 et 1 se réalise par la simple commande :

```
vbladed 0 1 eth1 /dev/sda
```

Pour iSCSI, l'implémentation iSCSI utilisée est iSCSItarget[3] : elle nécessite la compilation d'un module. Voici la mise à disposition du disque SDA suivant le « nom » iSCSI (IQN) iqn-2011-11.toulouse.jres:22.23.24.25 :

```
IncomingUser discoveryname Discover4Jres2011
Target iqn.2011-11.toulouse.jres:22.23.24.25
        IncomingUser inputname Input4Jres2011
        Lun 0 Path=/dev/sda,BlockSize=4096,Type=fileio
```

Côté client, rien n'est à faire, tout existe déjà autant dans le noyau que dans les outils de contrôle fournis sous forme de *packages*.

Pour AoE, la découverte et la disponibilité des disques est immédiate. Alors que la commande aoe-discover lance la découverte des périphériques AoE, la commande aoe-stat permet de les visualiser. Tous apparaissent sous forme de périphériques dans /dev/etherd/eX.Y avec X et Y représentant les majeur/mineur. Reste ensuite à les exploiter comme des disques locaux.

Pour iSCSI, les outils classiquement proposés sont issus du projet Open-ISCSI[4]. La commande « fourre-tout » est iscsiadm, laquelle est aussi bien utilisée pour la découverte (iscsiadm -m discovery -t st -p serveur) que pour le montage (iscsiadm -m node -l -T IQN). Par défaut, iSCSI crée un contrôleur IET SCSI : chaque disque rajouté se trouve donc « noyé » au milieu des autres disques locaux ou distants. Il est alors nécessaire d'explorer les variables systèmes pour savoir quel IQN est associé à telle machine distante : l'administration n'en est que plus compliquée !

Comme l'AoE et le iSCSI sont du « mode bloc », chaque disque apparaît comme s'il était local : il est alors possible d'agréger ces volumes en utilisant les mêmes composants que pour les

disques locaux : tous les modes de RAID via mdadm, toutes les opérations LVM (Logical Volume Manager) via lvm2 et le chiffrement via dm-crypt sont exploitables, de même que tous les systèmes de fichiers existant sur la machine.

A noter que le système de fichiers présenté comme le plus complet et le plus abouti, remplaçant à la fois les couches RAID et LVM, ZFS, dispose maintenant d'une implémentation native sous Linux[5]. La toute dernière version (la 0.6.0-rc6), déployée également, semble démontrer de réelles capacités dans la gestion de nombreux disques.

3.2 « Mode fichier » : un environnement complet à déployer

Le prérequis de GlusterFS pourrait se résumer à la couche « Fuse ». La mise en œuvre de GlusterFS est très simple : l'installation du même logiciel (sous forme de paquets) sur les nœuds et l'agrégateur, le lancement d'une commande sur l'agrégateur pour interroger les nœuds en précisant le nom de la machine et le point de partage, la création du volume sur l'agrégateur (avec la précision du type « normal », « miroir » ou « parallèle »), le démarrage du volume créé, la création du point de montage et le montage. Les commandes d'administration sont complètes et permettent de visualiser rapidement la composition et l'état des éléments formant le ou les volumes.

La mise en œuvre de XtreemFS est un peu plus compliquée : cela commence par l'installation de Java, indispensable (à noter que la version OpenJDK est fonctionnelle !) et de Fuse. Le mécanisme est comparable à GlusterFS sauf que la configuration des nœuds est à faire de manière atomique, chacun ayant la sienne : ainsi est détaillé le point de montage des données et un identifiant unique pour chaque OSD (Object Storage Data). Pour l'agrégateur, deux services sont indispensables : le DIR (Directory Service), le MRC (pour les méta-données et la définition de la réplication). Il est alors possible, en se connectant sur l'interface Web de gestion, de connaître l'état de ces trois services OSD, DIR et MRC. La création des volumes est simplissime : juste la précision du serveur et du volume à créer. Une commande de montage spécifique permet ensuite de disposer de l'arborescence.

Configurer un espace CephFS a été, de loin, la plus difficile parmi les trois solutions exposées. L'installation initiale se réalise simplement par l'intégration d'archives externes aux arbres standards de logiciels. Contrairement à ce qui est mentionné dans la documentation (à savoir la très forte intrication entre BtrFS et CephFS), il n'a été possible de créer un partage CephFS que sur les vieux systèmes de fichiers, Ext4 en l'occurence. Pour la configuration, il est nécessaire d'installer les paquets CephFS et de configurer le « maître » avec toute la configuration initiale de la grappe de stockage : doivent être précisés le moniteur (MON), le serveur de méta-données (MDS) et les unités de stockage (OSD). On propage ensuite la configuration sur tous les éléments du cluster. Approche générique : il est possible de grouper les configurations associées à chaque service sous forme de « tronc commun » avec des identifiants liés aux nœuds. Le montage du volume agrégé est alors accessible : pour le monter, Fuse peut-être utilisé. Sur un noyau récent (supérieur à 2.6.34), il est normalement possible d'utiliser un module noyau directement, mais les tests ont montré des instabilités sur le module associé à un noyau 3.0.6 : la solution Fuse a donc été privilégiée pour les tests.

4 Sécurité

4.1 Approche « mode bloc » : une sécurité liée au réseau et à l'OS

Côté disponibilité, iSCSI et AoE reposent sur les couches bas niveau du système d'exploitation.

Côté intégrité, c'est sur le mécanisme de la couche d'agrégation qu'il repose : les mécanismes RAID via MDADM ou via ZFS on Linux offrent des contrôles intéressants. Quant à la crainte d'une modification « à la volée » sur le réseau des trames AoE ou paquets iSCSI, elle est certes

techniquement possible, plus facilement sur iSCSI que AoE : à quel public s'adresse-t-il ? Nous verrons qu'il existe des solutions à base de tunnel niveau 2 mais au détriment de la performance.

Côté confidentialité, deux aspects sont à analyser : le stockage interne à la cible et l'accès au stockage de la cible. Pour le premier, la sécurité associée aux données, en aval, est gérée au niveau du système de fichiers. Si une sécurité supérieure est exigible, la couche de chiffrement peut soit être introduite avant de supporter le système de fichiers (via dm-crypt), soit après (via encfs) : attention cependant au fait qu'un volume dm-crypt ne porte pas d'UUID avant son ouverture ! Quant au second aspect, sur la partie stockage local et sur le réseau, des mécanismes de filtrage (par adresse MAC sur AoE et par adresse IP sur iSCSI) sont possibles. Le protocole iSCSI intègre en plus de ses TCP Wrappers un mécanisme d'authentification Chap.

4.2 Approche « mode fichier » : des implémentations très diverses

Côté disponibilité, ces trois protocoles autorisent un mode de duplication permettant la disponibilité des données en cas de rupture d'un ou de plusieurs éléments. Il est même possible de savoir sur quel nœud se trouve tel élément stocké. XtreemFS offre également la possibilité de « blinder » la duplication finement, de toute l'infrastructure jusqu'au document.

Côté intégrité, les stratégies sont radicalement différentes. GlusterFS s'appuie sur le système de fichier hôte de chaque nœud. Si l'infrastructure vient à « planter », les données restent dans une arborescence, sur le point de montage de chaque nœud : l'intégrité repose donc sur le système de fichiers hôte. XtreemFS intègre des mécanismes internes de contrôle d'intégrité avec des sommes de hachage, paramétrable. Pour CephFS, la réponse semble être approxmiative si on en croit les forums. Attention, la documentation de CephFS est en refonte complète à cette heure (préférez le Wiki plutôt que l'arborescence officielle de documentation!.

Côté confidentialité, XtreemFS et CephFS utilisent des clés permettant de fixer la sécurité d'accès aux volumes. Leur configuration pour XtreemFS est cependant assez compliquée comparée à CephFS (SSL sous Java et ses keyrings). GlusterFS lui, ne dispose d'aucun dispositif spécifique pour contrôler l'accès au volume de montage : l'accent a tant été porté sur la facilité de mise en œuvre qu'il s'est réalisé au détriment de la sécurité d'accès aux données.

5 Rapide comparatif

Le tableau suivant offre un rapide aperçu de ce que nous pouvons attendre de ces approches.

Mode	**Mode « bloc »**		**Mode « fichier »**		
Type	**AoE**	**iSCSI**	**GlusterFS**	**XtreemFS**	**CephFS**
Socle	Néant !		Ni BtrFS Ni NilFS		Ext3/4
Client	Tous !		POSIX ACL		
Réplication	MDADM, ZFS, BtrFS		« Replicated »	Totale ou locale	Locale
Snapshot	LVM, ZFS, BtrFS, NilFS		Non	Non	Oui
Robustesse	MDADM, ZFS		Duplication de serveurs de Méta-données		
Disponibilité	Forte !		Bonne	Expérimentale	Expérimentale
Intégrité	Non	SCSI !	Intégrée	Intégrée	Intégrée
Confidentialité	Dmcrypt		Aucune !	SSL entre nœuds	Clé entre nœuds

6 Performances et empreintes système

A l'heure actuelle, ces cinq solutions AoE, iSCSI, GlusterFS, XtreemFS, CephFS, sont déployées sur les ressources du Centre Blaise Pascal, sous forme de volumes.

Afin de procéder à une véritable comparaison, une plate-forme expérimentale a été construite, basée sur les nœuds de cluster de Sun v22z (assez anciens) du Centre Blaise Pascal : 5 ensembles de 8 machines ont donc été formés pour déployer les solutions. Pour chacune des plate-formes, il a été choisi :

- l'utilisation d'un disque dédié de 136 Go en SCA sur chaque nœud ;
- une sécurisation des données par réplication : RAID6 pour iSCSI et AoE, mode réplication pour GlusterFS, XtreemFS (apparu dans la version 1.3) et CephFS ;
- l'usage du Gigabit Ethernet comme réseau d'interconnexion ;
- un unique agrégateur de volumes (une Sun x2100) disposant de 4 Go de mémoire.

Les tests ont été réalisés à l'aide de deux outils : la commande « dd » appliquée sur des blocs de 2 Mo sur l'équivalent en volume de dix cédéroms (en écriture puis en lecture) puis l'outil de test iozone3[9] sur un volume de taille identique (soit 7 Go).

Pour les volumes AoE et iSCSI agrégés dans un volume RAID6 (donc une double parité répartie sur deux disques), des tests basiques, figure6, (à base de dd en lecture et en écriture) puis invasifs (à base de iozone3) ont montré une efficacité de 70 à 88 Mo/s en écriture et de 103 à 112 Mo/s en lecture (pour AoE et iSCSI respectivement), comparable à celle d'un montage iSCSI sur une infrastructure dédiée. Sur le même test, les approches GlusterFS, CephFS et XtreemFS présentent des performances assez différentes : elles proposent typiquement des débits entre 41 et 56 Mo/s en écriture (la palme pour CephFS) et entre 60 et 104 Mo/s en lecture (le GlusterFS en mode Raid à la première place) pour une écriture et une lecture brutales sur de gros fichiers.

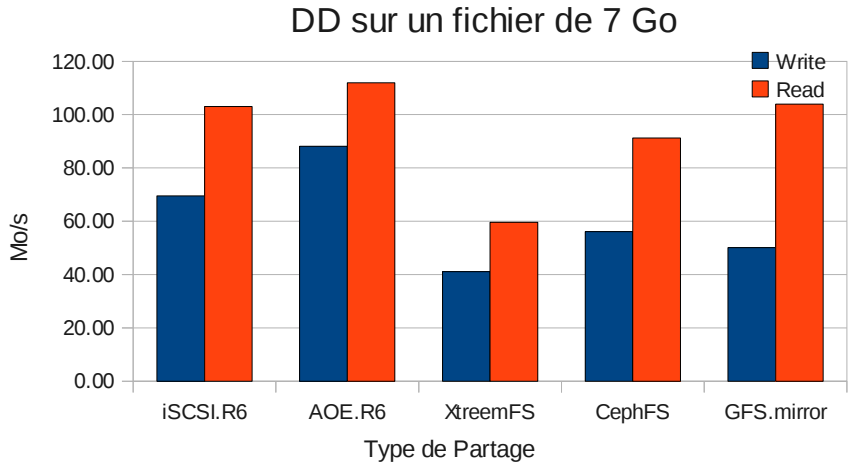

Illustration 1: Performances en écriture/lecture de gros fichier

Les tests IOZone (figure7) offrent un éclairage complémentaire sur les performances en lecture écriture dans des situations multiples. A noter que le test « Record Rewrite » a été exclu parce qu'il présentait des performances presque 10 fois supérieures en « mode bloc » qu'en « mode fichier » : cela écrasait complètement le graphique...

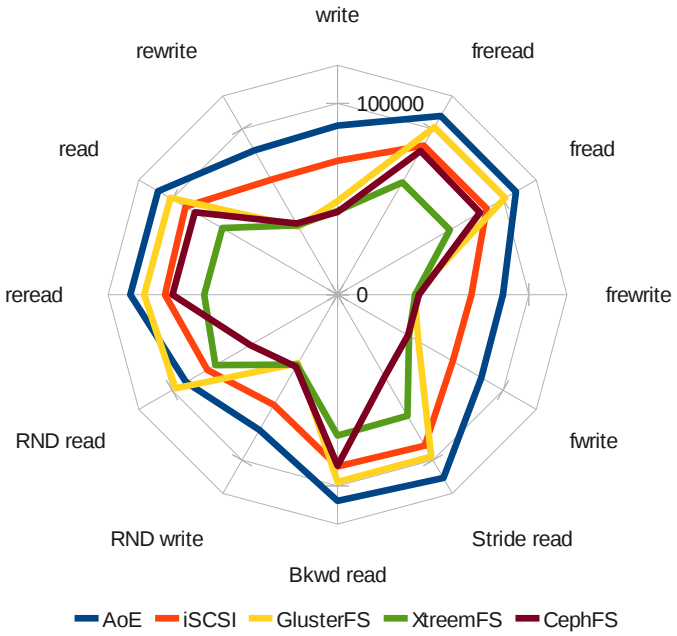

Illustration 2: Médiane de 11 lancements pour chaque approche, en Ko/s.

Quelles conclusions tirer de ces tests ? Les protocoles de « mode bloc » sont dans l'ensemble plus performants que leurs homologues, AoE, dans tous les cas, est le plus performant. En plus de sa simplicité d'installation, GlusterFS reste le plus efficace des « mode fichier ». CephFS talonne le iSCSI sur les tests de lecture. XtreemFS est le moins performant.

Pour l'heure, ces volumes sont utilisés comme espaces de sauvegarde et d'archivage avec des techniques à la « Rsnapshot » (basées sur des copies liées pour dédupliquer les données). Très invasives sur les systèmes de fichiers, leur usage sur plusieurs mois devraient juger, darwinisme oblige, de leur caractère opérationnel.

Dans le cadre de l'expérimentation, les mêmes implémentations ont été déployées, en plus, sur les postes de la salle libre-service du CBP, dans un environnement moins contraint.

7 Retour au poste de travail

Il existe des solutions permettant de fédérer des ressources distantes pour l'offrir comme service global. Elles se déploient sans difficulté : leur intégration dans des environnements maîtrisés peut être largement automatisée. Reste le point initial : comment réaliser pareille opération sur une machine « non maîtrisée » ?La solution expérimentée consiste à mettre à disposition, en fonction de la destination (locale ou distante), une machine virtuelle sous VirtualBox, proposant ce service. En effet, VirtualBox a été choisie pour son implémentation Open Source et son universalité sur les plates-formes. La machine est construite de manière identique à celles des architectures déjà utilisées : le système est soit distant (PXE puis NFSROOT), soit local. Toute ou une partie de la ressource disque déclarée est alors exploitée et rejoint l'architecture globale : ainsi, la ressource partagée n'est que le disque virtuel mis à disposition. Tout mécanisme de service et éléments de sécurité sont intégrés à la machine virtuelle, alors ressource de la fédération de stockage.

Des postes déployés pour un projet particulier, disposant d'un environnement scientifique complet, forment déjà le démonstrateur de cette approche en gestation. L'utilisateur verra tout son intérêt à conserver la machine active simplement parce que c'est elle qui lui propose l'accès au service (au moins de manière privilégiée) !

Quant aux mécanismes de sécurité, notamment liés à l'interconnexion réseau entre les nœuds et l'agrégateur, le choix s'est porté sur l'utilisation d'un réseau privé interne au réseau de stockage. Des solutions à base de stunnel ou de tunnel SSH pouvaient être envisagées mais elles manquaient d'universalité : avec l'usage d'OpenVPN au niveau 2, il est possible, sur ce pool de machine, de se retrouver dans un réseau local complètement privatif, condition raisonnable de l'exploitation de certains outils, notamment GlusterFS.

8 Conclusion

Pratiquer le stockage distribué dans son propre établissement ne tient plus, désormais, de l'expérience de pensée : des solutions présentées permettent déjà de pouvoir lier efficacement les volumes disques inutilisés de serveurs, voire de postes de travail bien maîtrisés. La stabilité de ces solutions tient essentiellement à la disponibilité du réseau et des ressources systèmes. Si les consommations de processeur et de mémoire restent limitées (presque anecdotiques), le réseau se retrouve au centre de toutes les attentions : comme dans un cerveau, les neurones ne sont rien sans une bonne interconnexion les reliant.

Dans notre étude, nous avons étudié finalement le pire cas de figure : un espace de stockage permanent, accessible tout le temps. La difficulté s'en trouve réduite si nous ne cherchons pas à accéder à ces volumes en permanence mais seulement pendant des opérations de sauvegarde ou d'archivage. Dans un contexte d'économie d'énergie, éteindre les disques inutilisés et les réactiver à discrétion pour les transformer en « bande à accès directe haute performance » est une solution que nous comptons aussi proposer, pour du stockage pérenne cette fois.

Ainsi, qu'il s'agisse d'éléments de calcul (les nœuds de cluster), des postes de travail de salle d'enseignement, ces solutions fonctionnent sous forme de démonstrateurs. D'autres démonstrateurs sur les postes de travail « non maîtrisés » sont en cours de mise au point. Seul l'usage, l'accumulation des problèmes et leur règlement permettront, dans le futur, de juger si à cette étape succédera une phase de mise en production.

9 Bibliographie

[1] Page officielle de DRDB http://www.drbd.org/

[2] Page officielle des outils AoE http://aoetools.sourceforge.net/

[3] Page officielle de iSCSItarget http://iscsitarget.sourceforge.net/

[4] Page officielle de OpenISCSI http://www.open-iscsi.org/

[5] Page officielle de ZFS on Linux http://zfsonlinux.org/

[6] Page officielle du projet GlusterFS http://www.gluster.org/

[7] Page officielle du projet XTreemFS http://www.xtreemfs.org/

[8] Page officielle du projet CephFS http://ceph.newdream.net/

[9] Page officielle de IOZone http://www.iozone.org/