ISCSI et GFS
Quelques réflexions sur les systèmes de fichiers très disponibles

Philippe DEPOUILLY[1]

[1] Labo. Bordelais d'Analyse et Géométrie, CNRS-Université de Bordeaux I

Le Contexte

Parlons de DAS, NAS, NFS, CIFS, SCSI, RAID...

Disque dur local

- Un ordinateur, une carte mère, un controleur, une nappe, un disque dur, une table de partition, un système de fichiers, un système d'exploitation
- Des protocoles IDE, SATA, SCSI...
- Un contexte très protégé : tout est à l'intérieur d'un boitier ou au pire au bout de un mètre de cable
- Utilisation locale des données
- Si on utilise un disque externe : **DAS (Direct Access Storage)**

NFS/CIFS

- Partager des données sur un réseau
- D'un ordinateur (serveur avec ses disques locaux) vers des ordinateurs (clients)
- Protocole orienté Un vers Un : un client à la fois modifie un fichier
- NFS permet de gérer les verrous mais les accès multiples sont mal gérés, problèmes de cohérence de cache : sur des intervalles courts, le fichier n'est plus le même pour chaque client, se corrige avec "actimeo=x", mais fait chuter les performances
- CIFS (smb) est réservé au monde Windows comme NFS est réservé au monde Unix
- Un et un seul serveur met à disposition Un et un seul système de fichiers (le sien)
- Un serveur mettant à disposition son système de fichier est appelé un **NAS (Network Acces Network)**

DAS/NAS/RAID

- **Le DAS** ne répond pas vraiment à nécessité de rationnalisation du travail d'AS&R
- **Le NAS** si, c'est une première réponse à la disponibilité des systèmes de fichiers : **accès multiples et distants**
- **Le RAID** (Redundant Array of Independant Disks) est un système automatique de duplication des données au moment de l'écriture
- RAID matériel/logiciel est une autre réponse à la disponibilité : **protection/sécurisation**

DAS/NAS/RAID

- RAID 0 : entrelacement (stripping), on associe des disques visibles sous un seul disque et on éclate les écritures sur chaque disques : performances accrues
- **RAID 1** : mirroir, on utilise un disque mirroir pour un disque de données : chaque écriture est double, chaque lecteur est unique
- RAID 0+1 : mirroir + entrelacement
- RAID 3 : N disques +1 pour une parité : un disque peut tomber en panne, la parité permet de retrouver l'information
- **RAID 5** : N+1 disques : la parité est entrelacée sur tous les disques
- RAID 6 : N+2 disques : deux disques peuvent tomber en panne... (peu convaincant)
- **JBOD (Just a Bunch Of Disks)** : des disques sans RAID/accumulation de disques

DAS/NAS/RAID

- Le NAS + RAID permet de rendre disponible des données sécurisées depuis un serveur vers des clients
- La réponse est satisfaisante pour le cas ”général” (ajoutons une sauvegarde des données)
- En revanche :
  - Problèmes de performances : NFS est réputé lent (NFSv4 apporte juste un mieux)
  - Ne répond pas à l'effet de mode de la haute disponibilité par la redondance totale (absence de **SPOF : Single Point Of Failure**)
  - Ne répond pas à la problématiques des accès concurrents fréquents (gestion de verrous : Locks)

Plus de disponibilité

Maintenant, parlons de SAN, FC, ISCSI, SPOF, TOE, InfiniBand...

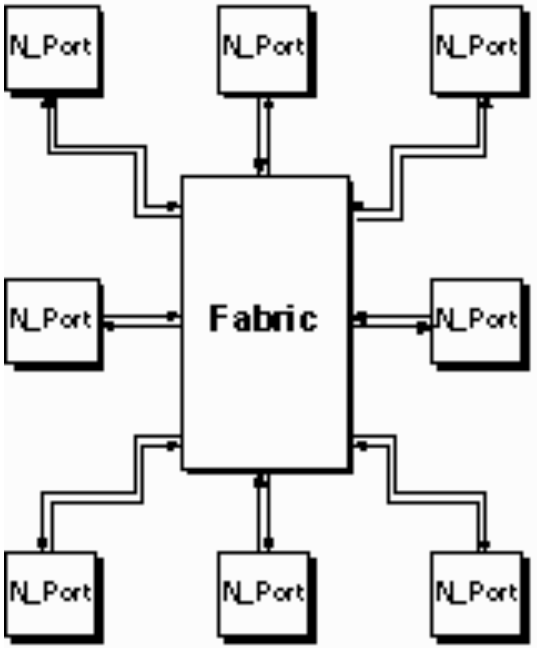
SAN

- Précédements, les disques sont directement branchés aux serveurs qui distribuent les données.
- L'accès aux données depuis les clients passent par trois intermédiaires : le réseau, le serveur et les disques.
- Un élément de la chaîne en panne, les données sont perdues.
- Le protocole SCSI (Small Computer System Interface) permet de contrôler des supports de stockage (disques durs, périphériques à bandes).
- La contrainte est que l'acheminement reste matériellement très controlé (un cable court entre le disque et le contrôleur qui permet d'acheminer les blocs de données parallèlement, même principe qu'un port parallèle).
- Un cable SCSI va de 1m à 20m (grand maximum).

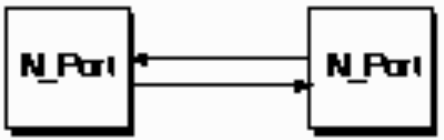
SAN

- Une réponse est de dissocier disques et serveurs : le serveur accède aux disques via un réseau :
- **Le Fibre Channel** et plus récement le **ISCSI**, ce sont des disques **SAN (Storage Area Network)**
- Acheminer SCSI sur un réseau pour profiter des mécanismes de typologie de réseau (étoile, bus, commutation, etc.)

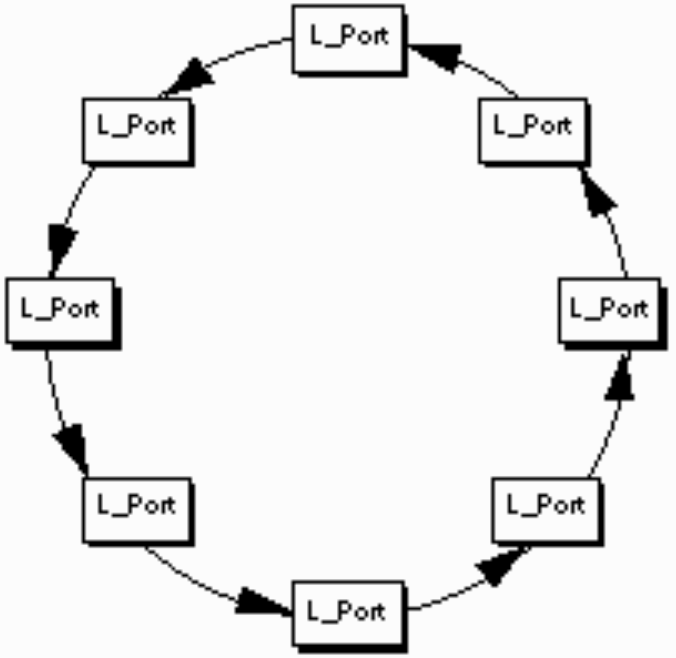
Fibre Channel



Fabric

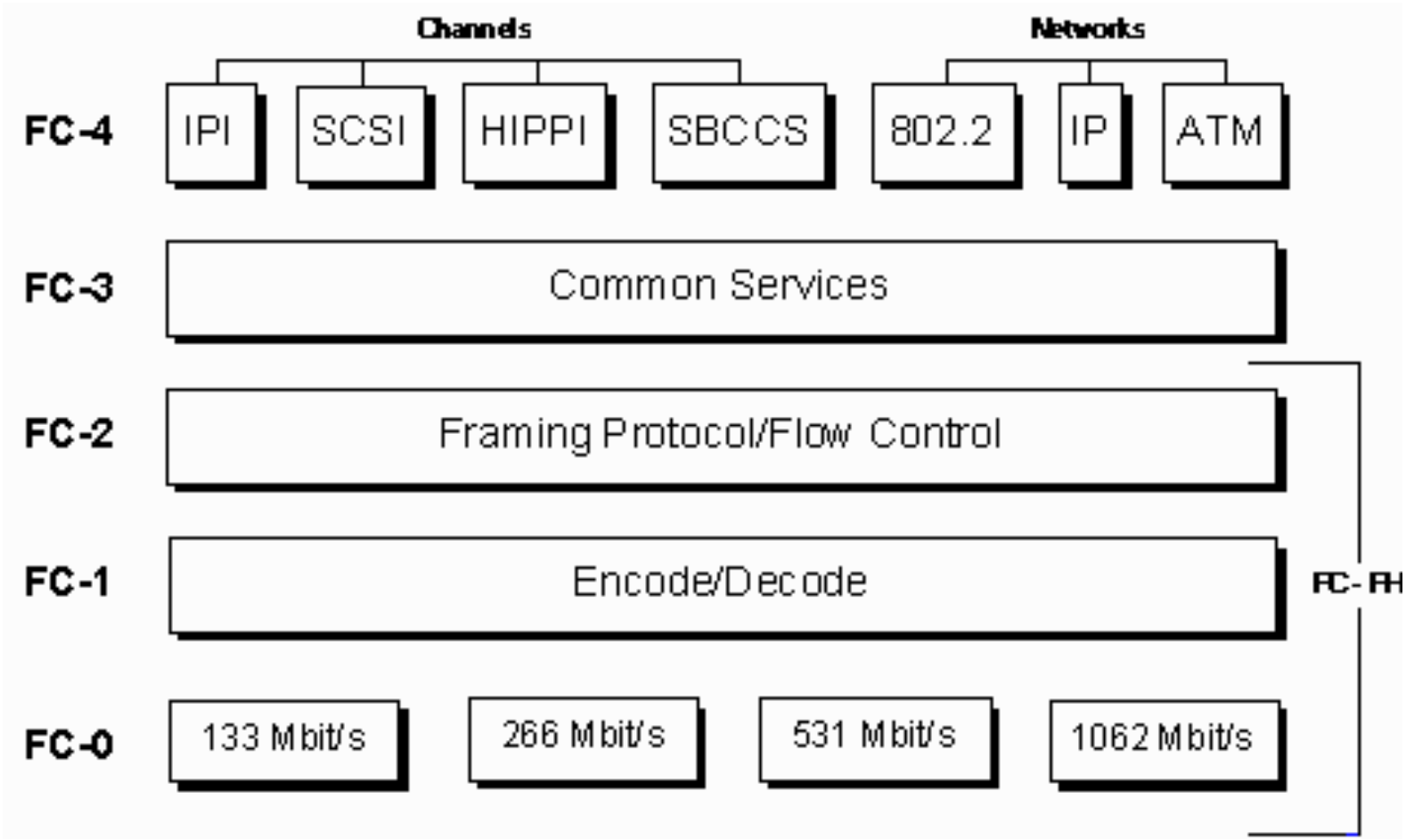


Point_to_Point



Loop

Fibre Channel



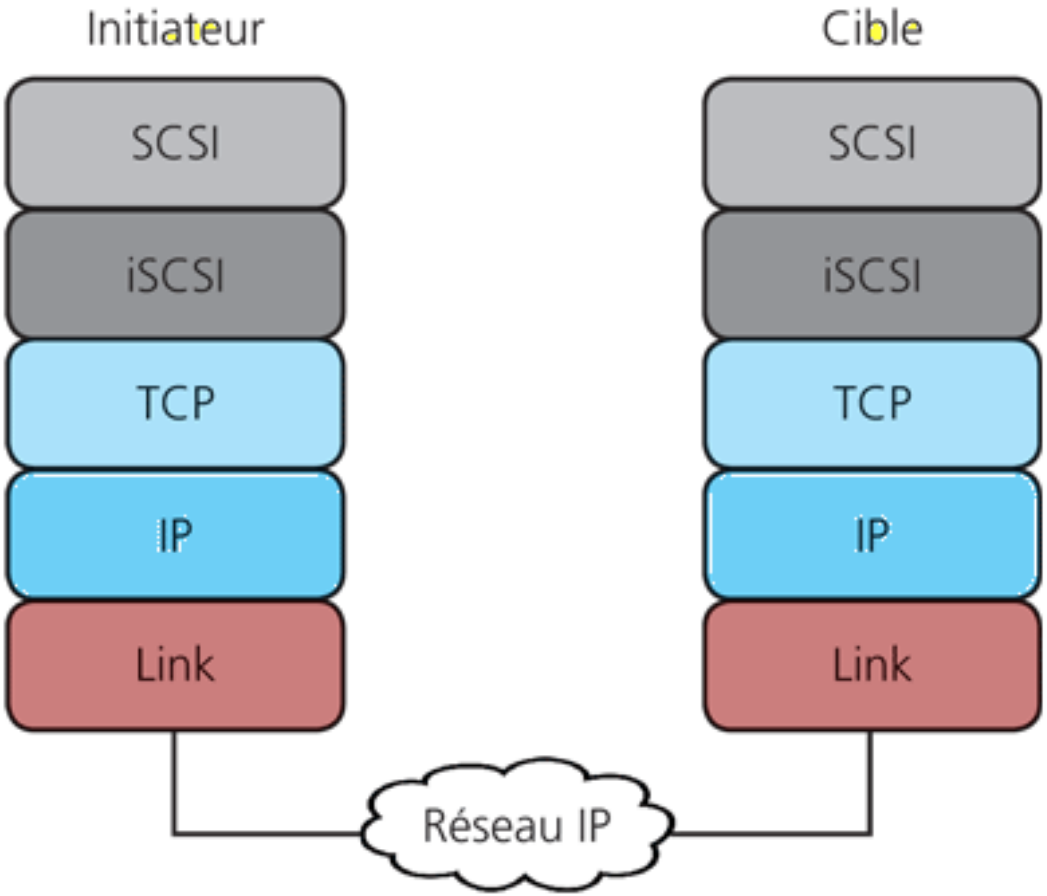
Fibre Channel

- FC est performant (1, 2 et 4 Gb/s)
- FC est robuste
- FC offre les fonctionnalités de haute disponibilité (sans SPOF)
- Nécessite du matériel spécifique (cartes controleurs HBA, commutateurs, drivers)
- FC est honéreux

ISCSI

- ISCSI est une réponse récente, le SAN du pauvre
- ISCSI encaspule des requêtes SCSI dans des paquets TCP, sur des trames Ethernet...
- S'appuie sur le matériel existant (tous les serveurs ont au moins une carte et sont connectés sur un réseau Ethernet)
- Les Baies ISCSI troquent le connecteur SCSI contre un connecteur Ethernet Gb/s

ISCSI



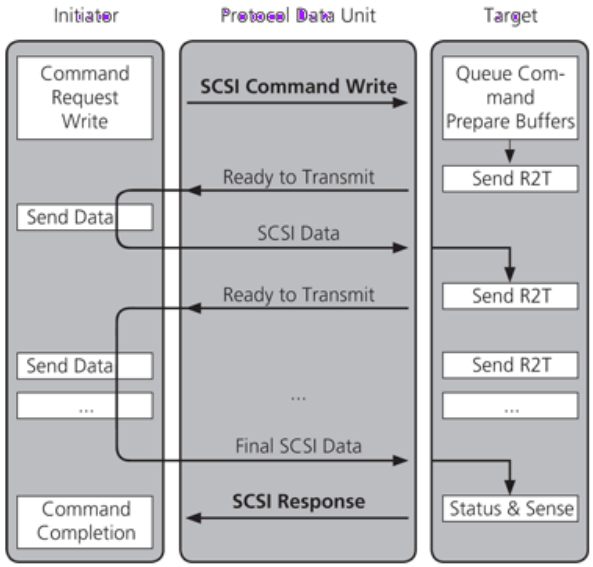
Les objectifs de ISCSI

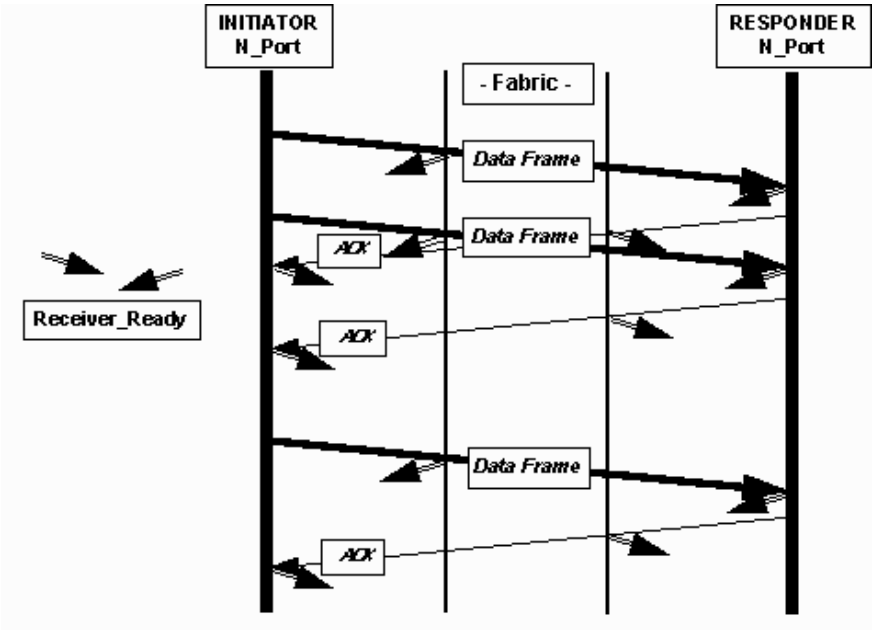
- S'appuyer sur l'infrastructure existante (l'adressage IP et les commutateurs existants)
- Ne pas interférer avec le trafic existant (ne pas saturer un réseau)
- Utiliser les mécanismes TCP de garantie de trafic pour garantir le protocole SCSI
- Assurer la gestion SCSI sur un LAN (Approche plus radicale de DRBD)
- S'appuie sur des réseau Gb/s (pas 100Mb/s)
- Même sur du WAN (routeurs, etc.) !
- Une réponse très prometteuse !
- Mais...

Les objectifs de ISCSI

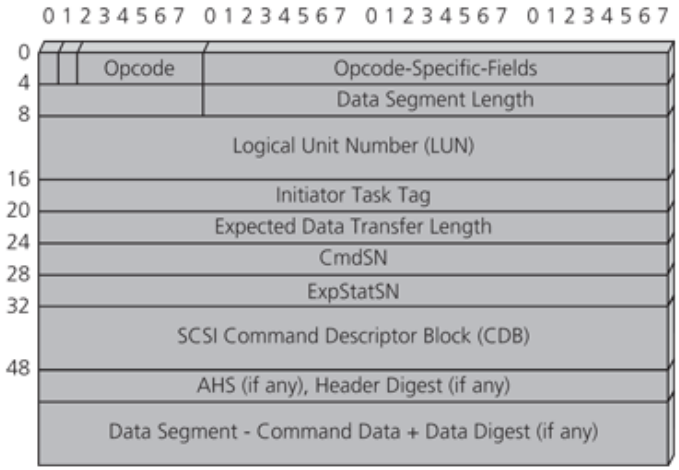
- Un LAN, ou pire un WAN, n'est pas suffisament sûr pour assurer ce trafic
- C'est coûteux au niveau du protocole (beaucoup de traitements d'erreurs)
- C'est coûteux au niveau de la CPU : un accès à un pilote SCSI coûte environ 5.000 cycles de CPU, contre 50.000 cycles au moins pour un empilement TCP/IP.
- TCP/IP : Le transfert de 1 bit réclame 1 Hz de fréquence du processeur, donc 1Gb nécessite 1GHz de processeur
- Le débit ne peut pas être garanti comme du SCSI ou du FC, car le réseau n'est pas toujours dédié
- ISCSI est un protocol encaspulé très complexe car l'information acheminée est très sensible

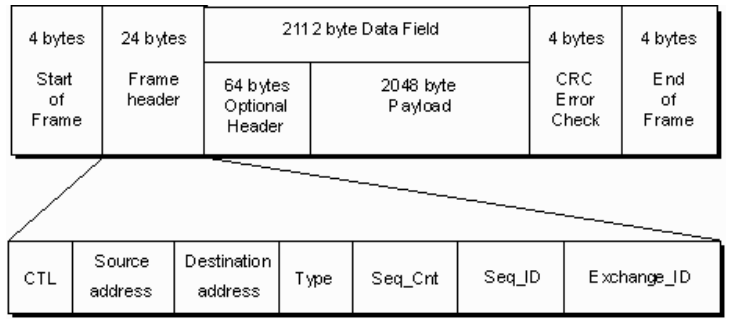
Exemples de transactions ISCSI et FC





PDU ISCSI / Trame FC





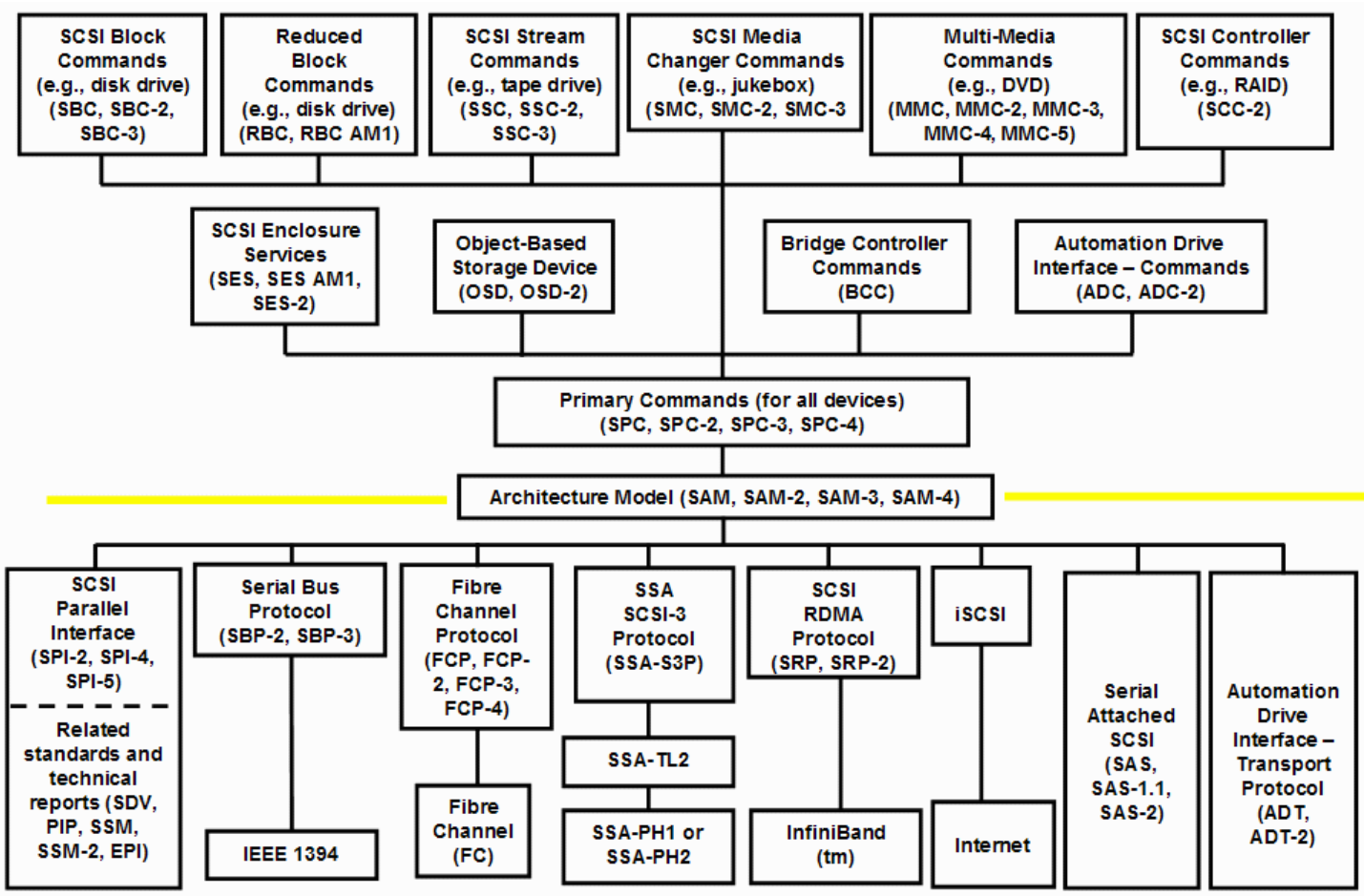
Les chiffres ISCSI

- depuis un serveur PIII bipro 1,3GHz, 2go de RAM, carte réseau intégrée e1000
- connecté à la baie directement via un cable RJ45 (sans commutation)
- sur la baie SATA/ISCSI : système de fichier reiserfs de 1TB en RAID 5
- copie de 500MB entre 10 et 12 secondes (même test sur notre baie IDE/SCSI : 15 secondes)
- jusqu'à 30% d'une CPU en trafic soutenu (stress I/O petits et gros fichiers)
- ISCSI sur 1 Gb/s ne rivalise actuellement pas avec du FC 1 ou 2 Gb/s
- ISCSI ne permet pas de raccorder directement des robots de bandes

Notre usage de ISCSI

- Depuis juillet 2005 : 1,5To plein utilisé pour du post-traitement
- Module noyau open-iscsi.org opérationnel depuis août 2005
- Les performances sont équivalentes à une baie IDE/SCSI
- La gestion est beaucoup plus souple qu'un controleur SCSI (le driver est en grande partie dans l'espace utilisateur : connecter/déconnecter un disque est très simple)
- Prévision de passer l'ensemble des baies en ISCSI (6 à 9 To) en 2006
- Les nouveaux serveurs commencent à intégrer les carte Ethernet **TOE** (TCP/IP Offload Engine)
- Le 10 Gb/s Ethernet arrive...
- Remaque : il est tout à fait possible d'éloigner des Baies en utilisant les drivers ISCSI (Host et Target)

SCSI/FC/ISCSI/Infiniband et même FireWire (IEEE1394)...



Et au niveau du système de fichiers ?

Maintenant, parlons de HA, RHCS, verrous POSIX, DLM, GULM, GFS, Lustre...

Mais jusqu'où irons-nous ?

- Nous avons parlé de SAN et RAID : dissocier les disques des serveurs, sécuriser et fournir des mécanismes de redondance au niveau des disques
- Nous sommes capable de dissocier les serveurs des ordinateurs : la virtualisation avec VMware, Xen, etc.
- Le disque système devient un fichier quelconque, il peut donc démarrer sur n'importe que la matériel et se duplique à volonté
- Maintenant nous souhaitons que les serveurs de fichiers soient multiples et redondants : c'est le dernier maillon faible
- Avec la virtualisation des serveurs, on virtualise les sytèmes de fichiers : plusieurs serveurs offrent le même système de fichiers, on n'identifie plus un seul serveur pour accéder aux données, mais un nuage de serveurs

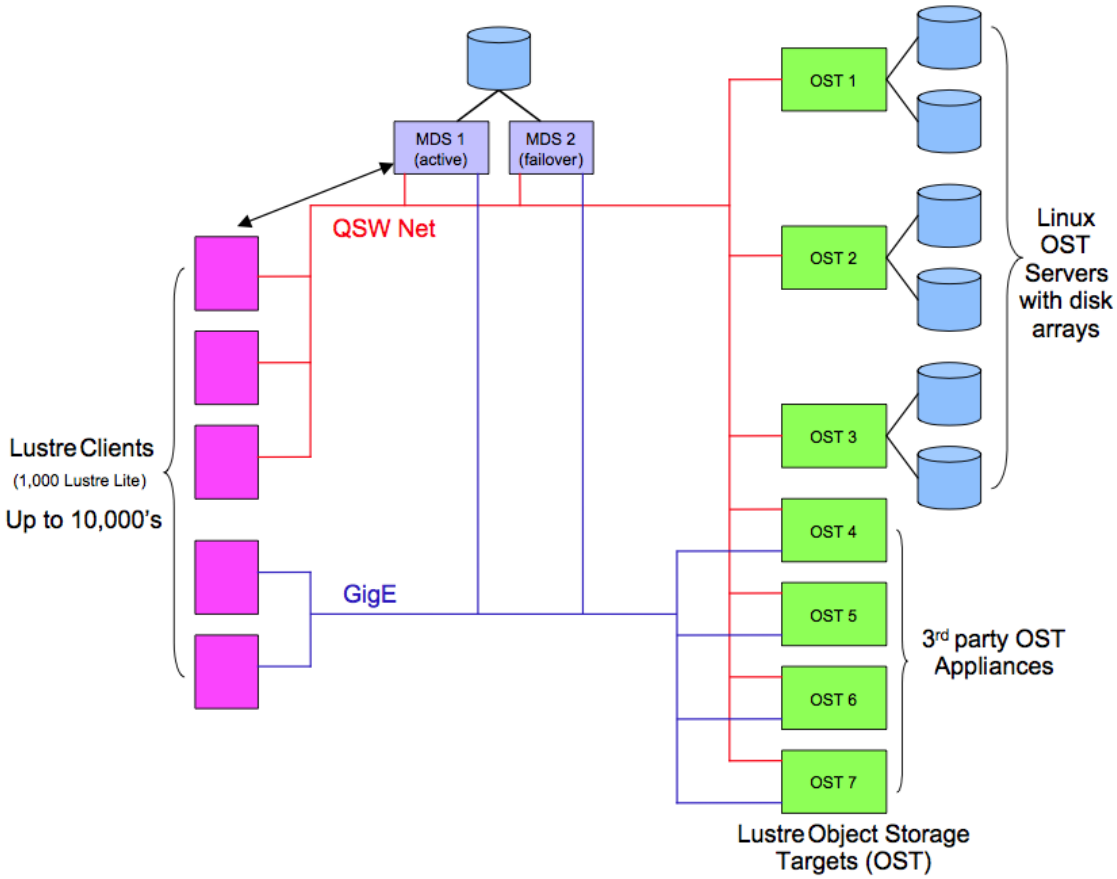
Pourquoi ?

- Tant qu'à virtualiser autant aller jusqu'au bout...
- Amélioration des performances
- Robustesse : **HA (Haute Disponibilité) Actif/Actif** : plusieurs serveurs offrent le même servce simultanément, **Actif/Passif (Failover)** : un serveur fourni le service, un autre est dispo pour remplacer en cas de panne
- Offrir ce que NFS ne peut pas proposer : des accès directs aux fichiers avec des verrous POSIX (les appels systèmes lockf et flock du client sont gérés directement par le système de fichiers sur le serveur au contraire de NFS où on passe par un démon (rpc.lockd))

Les deux prospectives : GFS et RHCS de redhat.com et Lustre de clusterfs.com

- Il y a beaucoup de choix (PVFS, GPFS, OCFS, GoogleFS, etc.), on en regarde surtout deux :
- Lustre
  - Lustre est avant tout un système de fichiers pour de la Haute Performance (les premiers clusters HPC du top 500 utilisent Lustre)
  - Orienté objet, manipule des méta-donnée, approche OO du système de fichiers
  - Version Lustre Light et Lustre commerciale
- GFS (Global File System)
  - GFS : projet de Sistina racheté par RedHat
  - Intégré avec RedHat Cluster Suite (RHCS)
  - Plus orienté cluster HA
  - Maintenant disponible dans sa dernière version en GPL (de base dans Fedora, et packagé sur d'autres distributions)

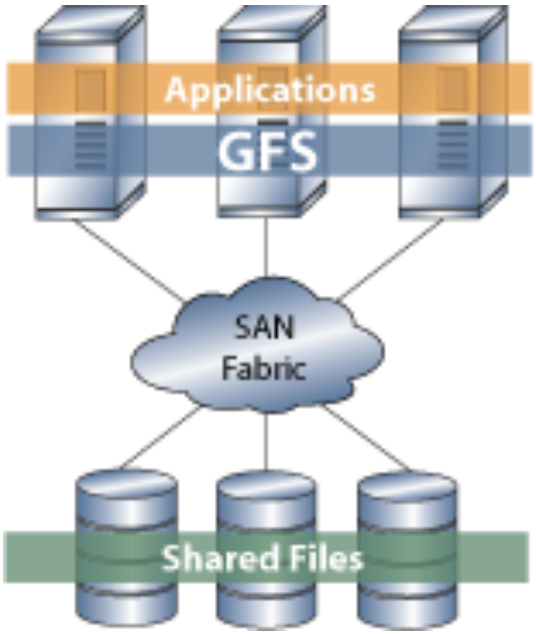
Lustre

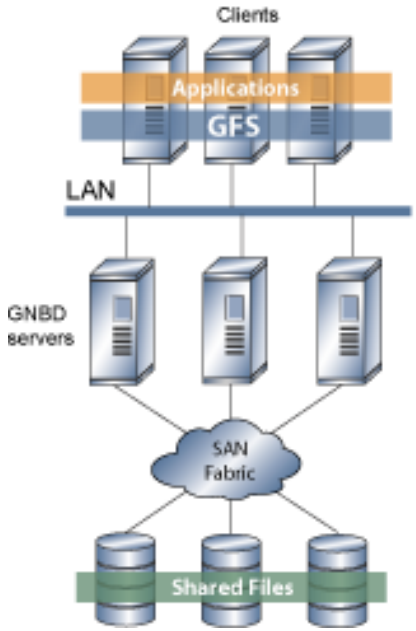


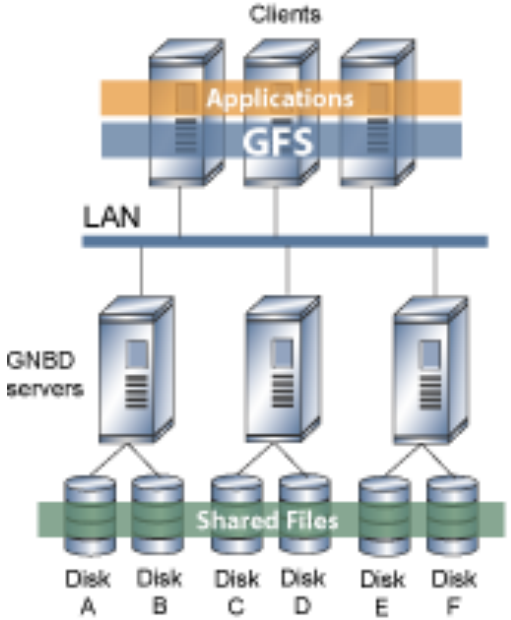
Lustre

- Lustre est orienté Cluster HPC, les noeuds servant pour le calcul offrent leurs disques
- Le système de fichier est distribué
- On recherche avant tout de la performance I/O
- Tente de respecter la norme POSIX avec les appels systèmes mmap et exec (dernière version : 1.46, mais ”hazardous” selon la FAQ de Lustre)
- Permet de déployer des systèmes de fichiers de plusieurs centaines de TB !
- Permet plusieurs milliers de clients simultanés
- La gestion des Locks est gérée par des serveurs dédiés redondants (un peu comme NFS, Lustre ne gère pas directement lockf et flock car un fichier peut être dispersé sur plusieurs systèmes)
- Basé sur ext3 (donc permettra bientôt le redimensionnement d’un système de fichiers)

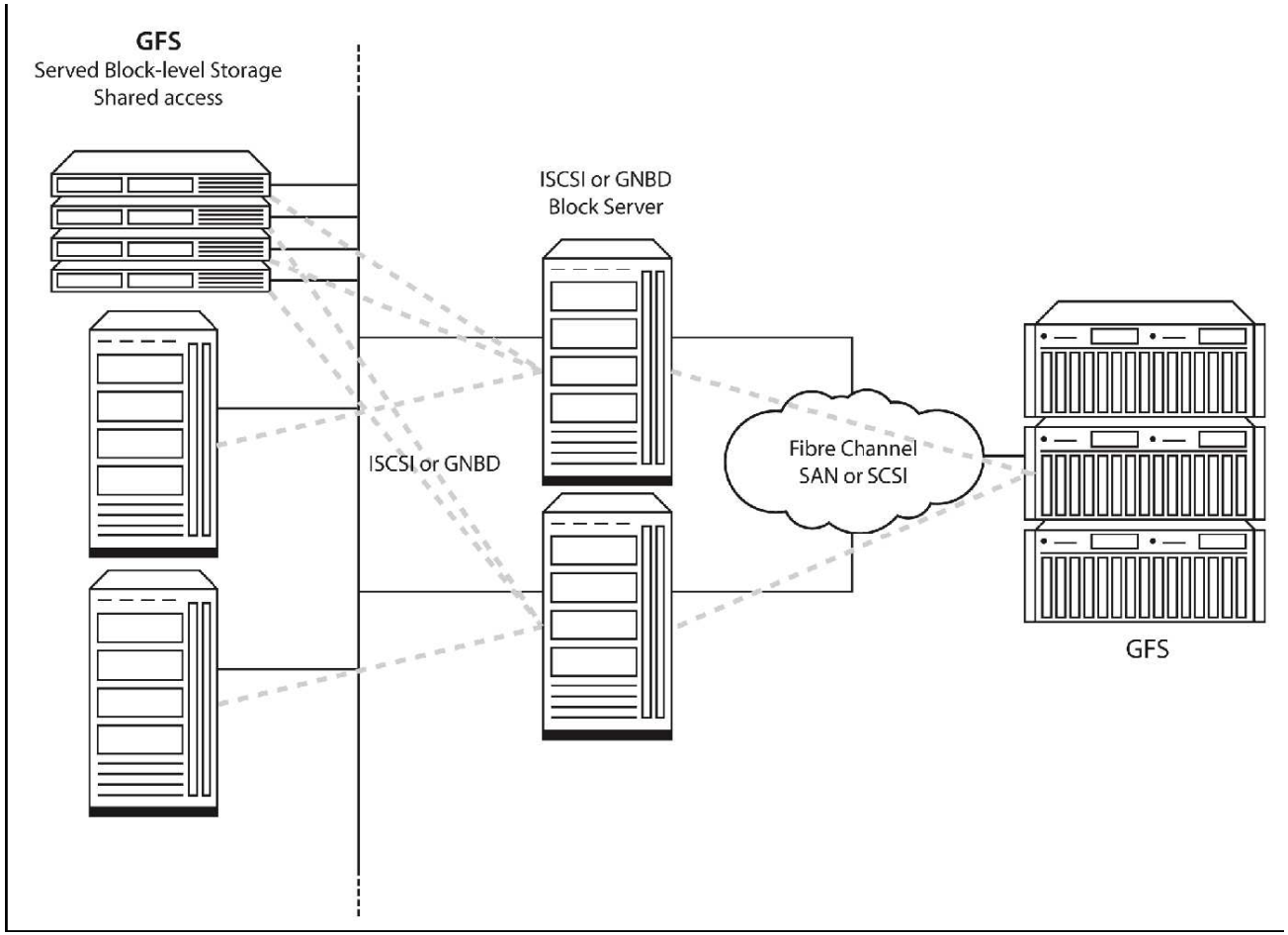
GFS







GFS



GFS

- Orienté Cluster HA et SAN
- Totalement respectueux de POSIX
- Et plutôt utile pour apporter la visibilité de SAN comme disques locaux
- Limité à 16 TB en 32 bits et 8 EB (ExaBytes) en 64 bits (sic !)
- Basé sur LVM2
- Technologie de snapshot (LVM)
- Limité à quelques centaines (256, 300 ?) clients
- Plutôt dédié aux systèmes de fichiers applicatifs (système, BD, etc.)
- Gestion de Locks via DLM (Distributed Lock Manager), GULM (Grand Unified Lock Manager) devient obsolète
- Disparition des serveurs dédié à la gestion de Locks
- ISCSI est préféré plutôt que GNBD

Où allons nous ?

- ISCSI semble être une norme interessante pour renforcer les espaces de stockages (disques durs) car il permet les chemins multiples (passage par des chemins réseaux redondants)
- GFS semble être une bonne approche pour un usage plus général d'un système de fichier qui peut être distribué mais surtout orienté SAN et clusterisé (sécurisé)
- Choix futurs plutôt vers ISCSI que FC ou SCSI
- Maquette GFS en cours pour le système de fichiers de l'IMB
- On en reparlera cette automne...