Stockage de masse

Organisation & outils

Capitoul
13 décembre 2018

Bruno Buisson - LEGOS - OMP

Le contexte du Legos

Science & outils

- Océanographie (hauturière et côtière)
- Hydrologie (fleuves et grands lacs)
- Biogéochimie
- ….
- Données « in-situ »
- Analyses de laboratoire

} négligeables

- Données satellites
- Modèles numériques

} dimensionnant

→ De (très) gros volumes de données

Le contexte du Legos
Les chiffres

- ~ 120 personnes (~200 comptes informatiques)
- ~ 1 Po de données nets (1.3 Po bruts)
  - 825 To (NL-SAS, Raid 6)
    Données Scientifiques
  - 20 To (SAS / Cache SSD, Raid 5+1)
    Homedirs Unix + Profils Windows + Bureautique + Messagerie + Catalogue TiNa
  - 25 To (SAS / Cache SSD, VSAN)
    Stockage dédié virtualisation
  - 130 To (SATA / NL-SAS, Raid 6)
    Sauvegarde sur disques
- 2 ASR

Comment gérer ces volumes sans personnel dédié, ni très spécialisé ?

Le contexte du Legos

Organisation

Equipes de Recherche & « Projets »

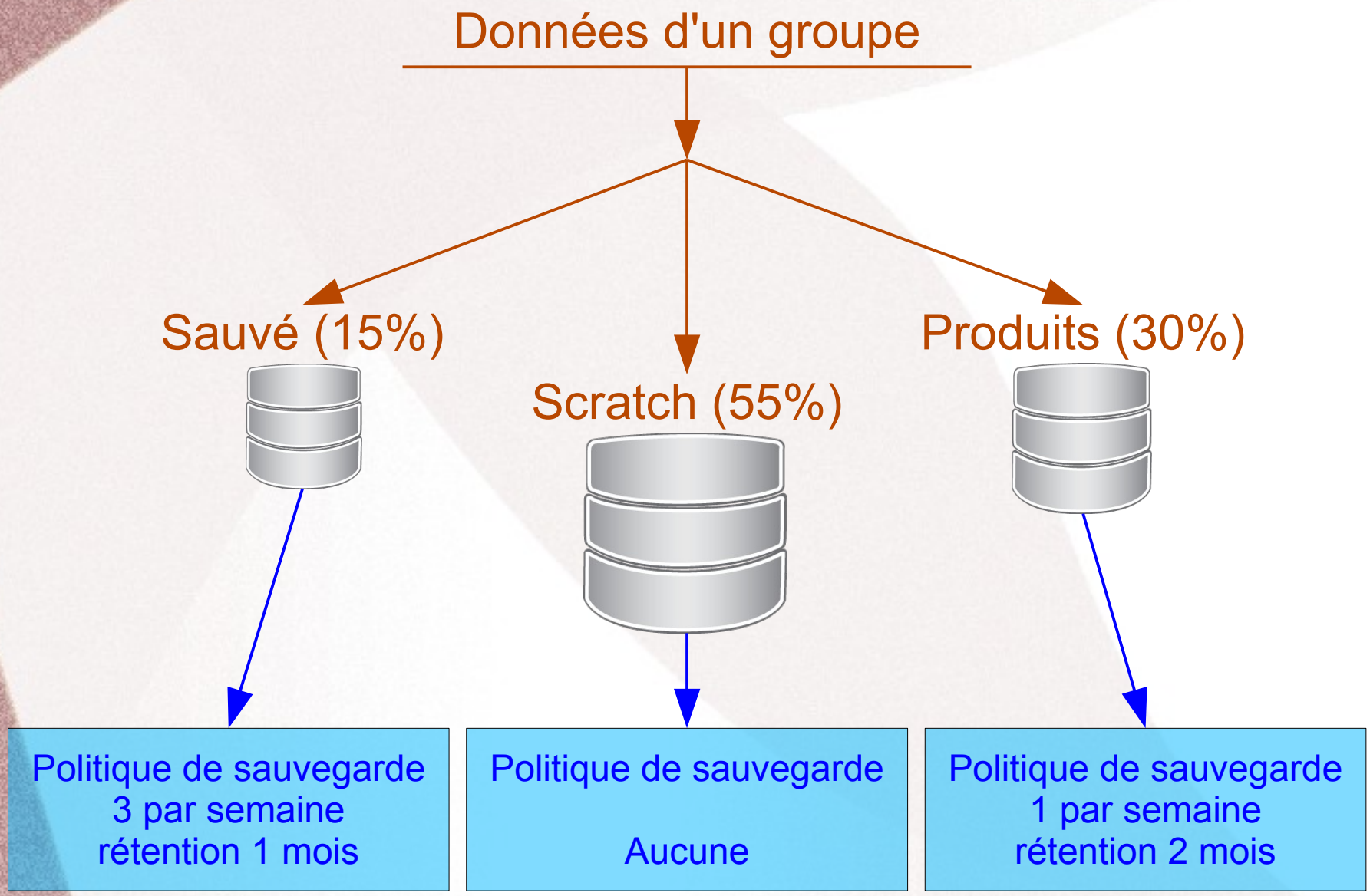
Les équipes de Recherche structurent le laboratoire : tout personnel scientifique appartient à (au moins) une équipe : 6 équipes (+ 2 services communs)

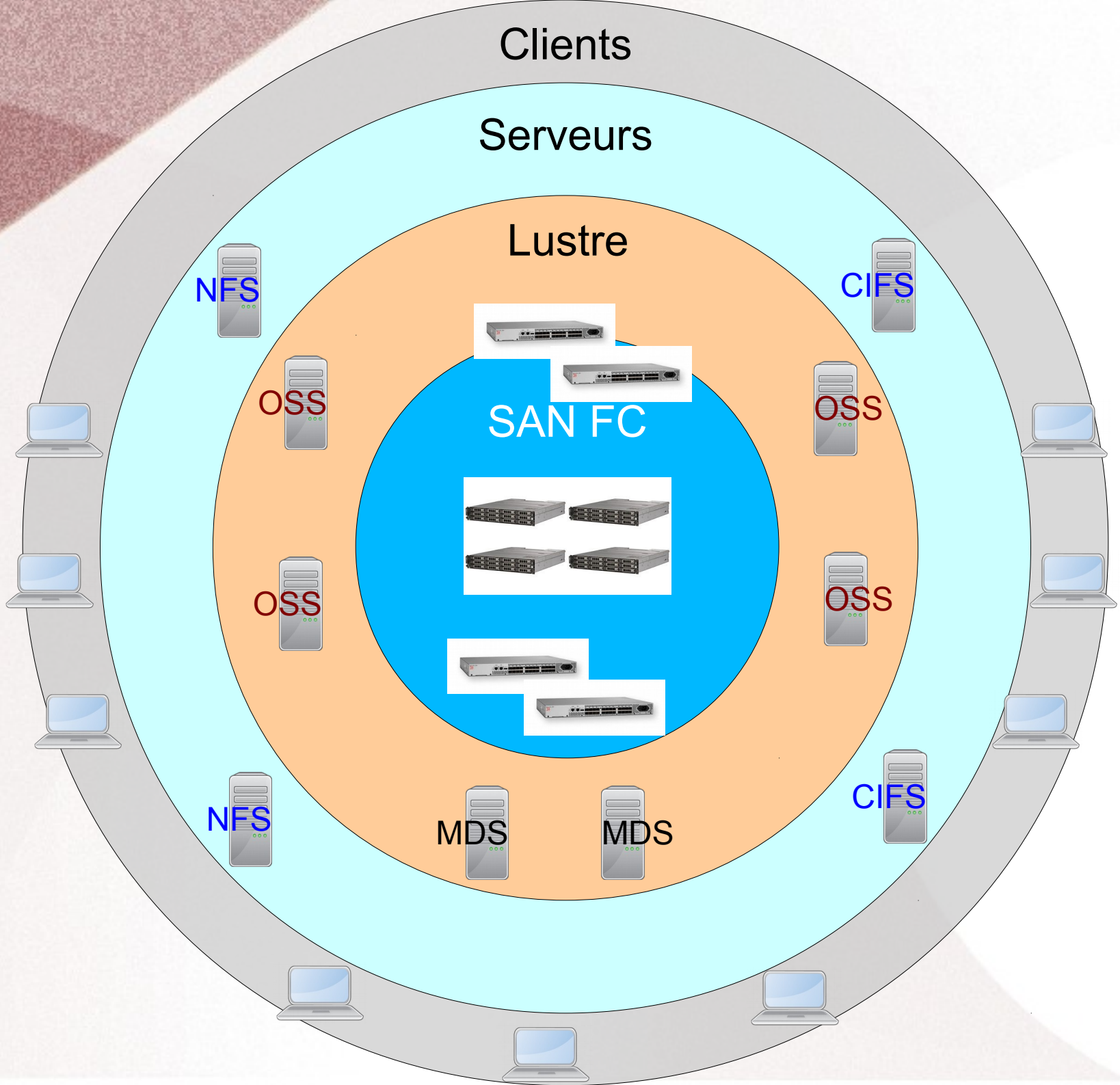
Un projet n'est qu'un regroupement de personnes désirant partager des outils, des données, des savoir-faire, etc. : 35 projets

→ Ce sont tous des groupes Unix

Sauvegarde

Politiques de stockage




Clients
Serveurs
Lustre
NFS
CIFS
OSS
OSS
SAN FC
OSS
OSS
NFS
MDS
MDS
CIFS

Stockage primaire
Lustre

- FS distribué, orienté HPC (mais pas que, la preuve)
  - Multi FS (20 au Legos), même si on croit souvent que 1 cluster Lustre = 1 FS : c'est faux !
- Lnet : Réseau Ethernet / Infiniband, … La plus faible latence, le mieux
  - Dédier un réseau (vlan ok) au trafic Lustre
- Stockage interne aux serveurs ou externe (DAS, SAN)
  - Backend de stockage ldiskfs (historique) ou ZFS (moderne)
  - (quasi) illimité en taille (512 PB si ldiskfs, 8EB si ZFS)
- Scalabilité OK si la CPU augmente en même temps que le stockage
- Prise en compte des quota, des ACLs et des attributs étendus
- Module HSM copytool (posix, S3, Rados, …)

Stockage primaire
Lustre Metadata

- Cohérence globale d'un FS assurée par les Metadata Servers (MDS)
  - Stockage sur des Metadata Targets (MDT)
- Nécessitent de bonnes (grosses) perfs en I/O (SAS → SSD si 10GE)
- Une partition (très) spéciale : Management Target (MGT), indispensable pour démarrer le cluster.
  - Montage sur le Management Server (MGS)
- Les MGT et MDT sont non perdables, sinon perte totale d'un FS, voire du cluster entier
  - Fonctionnement par paire actif/passif
  - Contrôle qu'un MDT n'est monté que par un seul MDS (Multi Mount Protection, MMP)
  - La « vraie » HA n'est assurée qu'avec un outil tiers (pacemaker,…)

Le stockage MGT et MDT doit être (plus que) redondé (Raid 6+1, idéalement sur 2 baies distinctes)

Stockage primaire
Lustre Data

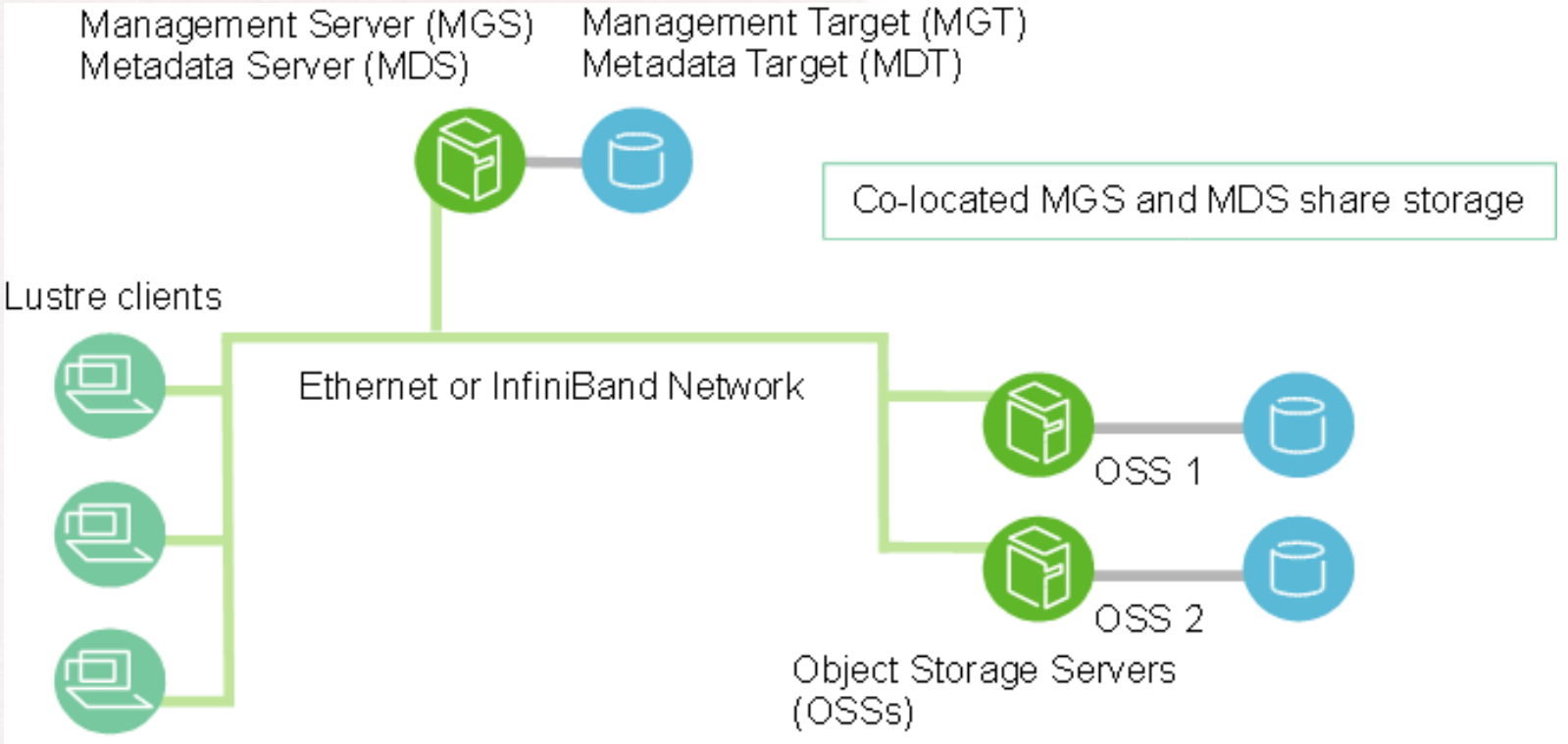
- Les données sont gérées par les Object Storage Servers (OSS)
  - Stockage sur des Object Storage Target (OST)
  - Les OSS travaillent par paire (actif/passif)
  - Contrôle qu'un OST n'est monté que par un seul OSS (MMP)
  - La « vraie » HA n'est assurée qu'avec un outil tiers (pacemaker,…)
- Les OST peuvent être répartis sur plusieurs OSS d'un cluster. La répartition de charge est naturelle, ainsi que le passage à l'échelle par augmentation des OSS
- Le transfert des données d'un OST (ancien) vers un autre (nouveau) se fait à chaud
- L'ajout (resp. suppression) d'un OST à un FS se fait à chaud

Stockage primaire
Lustre Pools

- Les OST peuvent être regroupés en Pools, ce qui permet de définir des sous-ensembles étanches à l'intérieur d'un même FS. S'il est plein, I/O error ! Mais on peut continuer d'écrire dans les autres Pools
- L'attribution d'un Pool est faite par répertoire
- Le stripping permet d'augmenter les performances
  - Les OST deviennent moins indépendants
  - Le stripping est défini par défaut, par répertoire ou par fichier
- Suivant le type de stockage physique, faire très attention à la manière dont celui-ci gère les locks (typiquement au niveau d'une LUN pour un SAN). Cela peut avoir un gros impact en performances
  - Privilégier des LUNS relativement petites

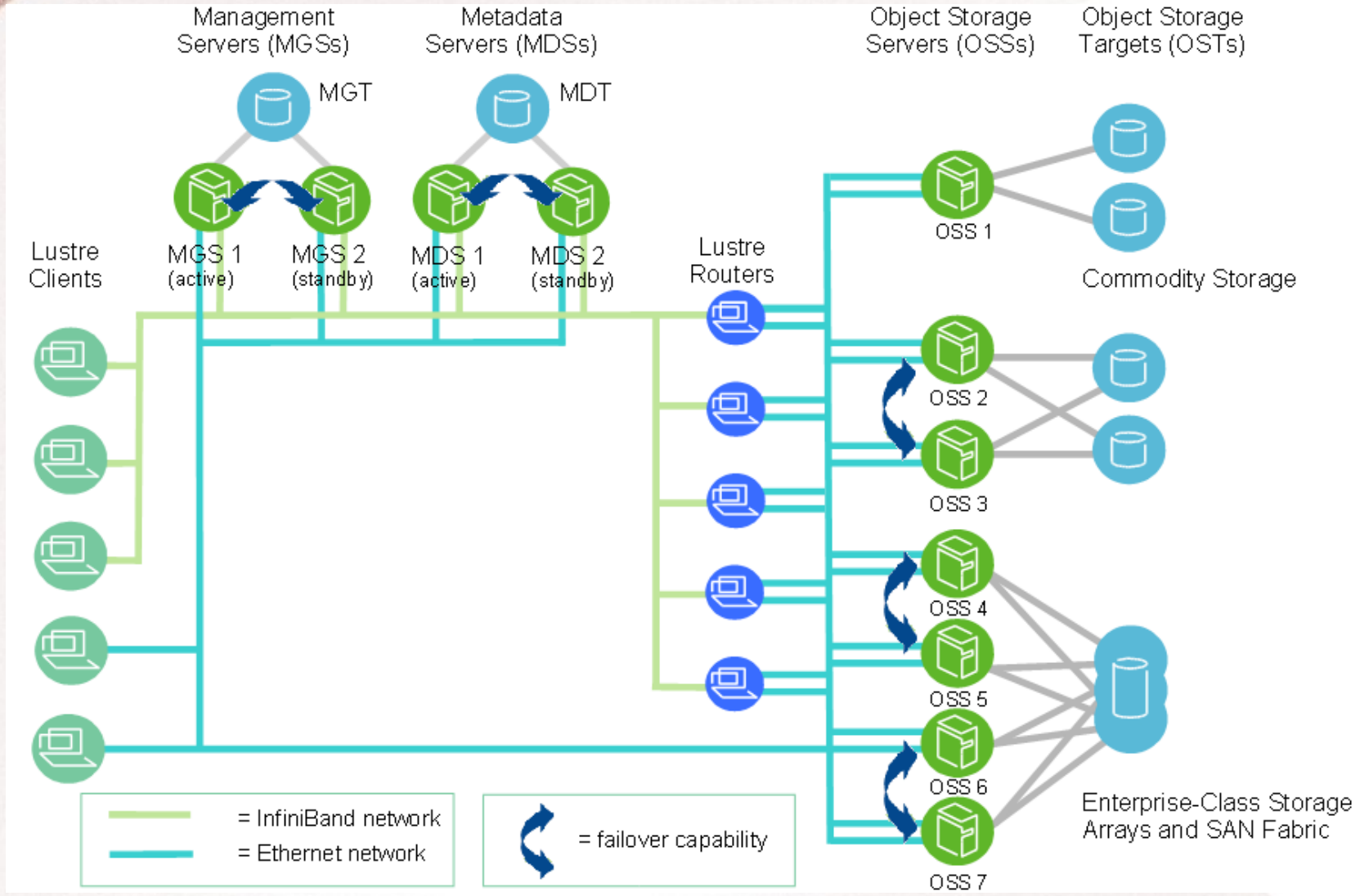
Stockage primaire

Architecture simple Lustre



Stockage primaire

Architecture ultime Lustre



Stockage primaire
Installer les machines Lustre

- Les MDS et les OSS sont dédiées à cette tâche
  - Noyau modifié, basé sur des versions bien précises
  - La matrice de compatibilité doit être étudiée avec soin
- Une machine Linux devient cliente Lustre par le chargement d'un module noyau
  - Versions du noyau à respecter strictement
- Conséquences : mises à jour noyau quasi-interdites
- Les compatibilités de versions entre les serveurs et les clients sont aussi très strictes et peuvent être problématiques
- RHEL (CentOS) sont privilégiées. Les versions serveurs et clients existent pour beaucoup de releases.
- Sles et Ubuntu ont moins de versions disponibles, souvent seules les modules clients sont disponibles
- Le portage sous Debian a été abandonné en 2014 et jamais repris depuis

Stockage primaire
Configuration réseau Lustre

```
# Le module noyau lnet MGS/MDS
MDS # cat >> /etc/modprobe.d/lustre.conf
options lnet networks=tcp1(eth1.101),tcp11(eth2.101),tcp111(eth2.101)
^D

# Le module noyau lnet OSS
OSS # cat >> /etc/modprobe.d/lustre.conf
options lnet networks=tcp1(eth1.101),tcp11(eth2.101),tcp111(eth2.101)
^D

# Le module noyau lnet Client
CLN1 # cat >> /etc/modprobe.d/lustre.conf
options lnet networks=tcp1(eth1.101),tcp11(eth1.101),tcp111(eth1.101)
^D
```

tcp1, tcp11, tcp111 sont des Network Identifier (NID) de lnet. Ils définissent un réseau Lustre. Combinés à l'interface réseau d'une machine, ils identifient chaque point de communication Lustre.

Stockage primaire
Créer le MGT Lustre

```
# Le MGT du cluster
MGS # mkfs.lustre    --mgs
                     --servicenode=mgs-master@tcp1
                     --servicenode=mgs-slave@tcp1
                     --mkfsoptions='-v -m 0' --verbose
                     /dev/vg_mgt/mgt
MGS #

MGS # cat >> /etc/fstab
/dev/vg_mgt/mgt      /lustre/mgt     lustre   defaults,_netdev       0 0
^D
MGS #

MGS # mount /lustre/mgt
MGS #
```

Un MGT par cluster Lustre, quelque soit le nombre de FS gérés

Stockage primaire
Créer un MDT Lustre

```
# Le premier MDT d'un FS « my_fs »
MDS1 # mkfs.lustre   --mdt --fsname=my_fs --index=0
                     --mgsnode=mgs-master@tcp1
                     --mgsnode=mgs-slave@tcp1
                     --servicenode=mds-master@tcp11
                     --servicenode=mds-slave@tcp11
                     --mkfsoptions='-v -m 0' --verbose
                     /dev/vg_mdt/my_fs_mdt0
MDS1 #

MDS1 # cat >> /etc/fstab
/dev/vg_mdt/my_fs_mdt0 /lustre/my_fs/mdt0 lustre defaults,_netdev,acl 0 0
^D
MDS1 #

MDS1 # mount /lustre/my_fs/mdt0
MDS1 #
```

→ Plusieurs MDT par FS sont possibles, un au minimum

Stockage primaire
Créer un OST Lustre

```
# Le premier OST de « my_fs »
OSS1 # mkfs.lustre   --ost --fsname=my_fs --index=0
                     --mgsnode=mgs-master@tcp1
                     --mgsnode=mgs-slave@tcp1
                     --servicenode=oss-master@tcp111
                     --servicenode=oss-slave@tcp111
                     --mkfsoptions='-v -m 0' --verbose
                     /dev/vg_ost/my_fs_ost0
OSS1 #

OSS1 # cat >> /etc/fstab
/dev/vg_ost/my_fs_ost0      /lustre/my_fs/ost0     lustre   defaults,_netdev 0 0
^D
OSS1 #

OSS1 # mount /lustre/my_fs/ost0
OSS1 #
```

Plusieurs OST par FS sont possibles, un au minimum

Stockage primaire
Montage d'un FS sur un client

Le montage de « my_fs » sur un client

```
CLN1 # cat >> /etc/fstab
mgs-master@tcp1:mgs-slave@tcp1:/my_fs          /mnt/my_fs     lustre
defaults,_netdev,localflock    0 0
^D
CLN1 #

CLN1 # mount /mnt/my_fs
CLN1 #

CLN1 # df -h -t lustre
mgs-master@tcp1:mgs-slave@tcp1:/my_fs
11T     2.0T    9.0T    19%    /mnt/my_fs
CLN1 #
```

Stockage primaire
Utilisation d'un FS par un client

- Un client Lustre peut utiliser les FS pour son usage propre (exemples : agent TiNa, service FTP ou HTTP, ...)
- Il peut aussi les exporter en NFS ou CIFS (Samba)
- Les exports NFS se font de manière standard, mais attention au FSID. Il est dit qu'il est indispensable de le fixer lors de l'export NFS, avec « d'anciennes » versions du client Lustre. Nous avons eu des soucis lors de ré-export (exportfs -a)
- L'usage de pNFS ou NFS ganesha peut améliorer les performances et le passage à l'échelle (nombre de clients)
- Samba propose aujourd'hui la notion de CTDB (Clustered TDB), ce qui facilite la mise en place d'un véritable cluster CIFS en mode actif/actif

Stockage primaire
Cluster NFS avec Lustre

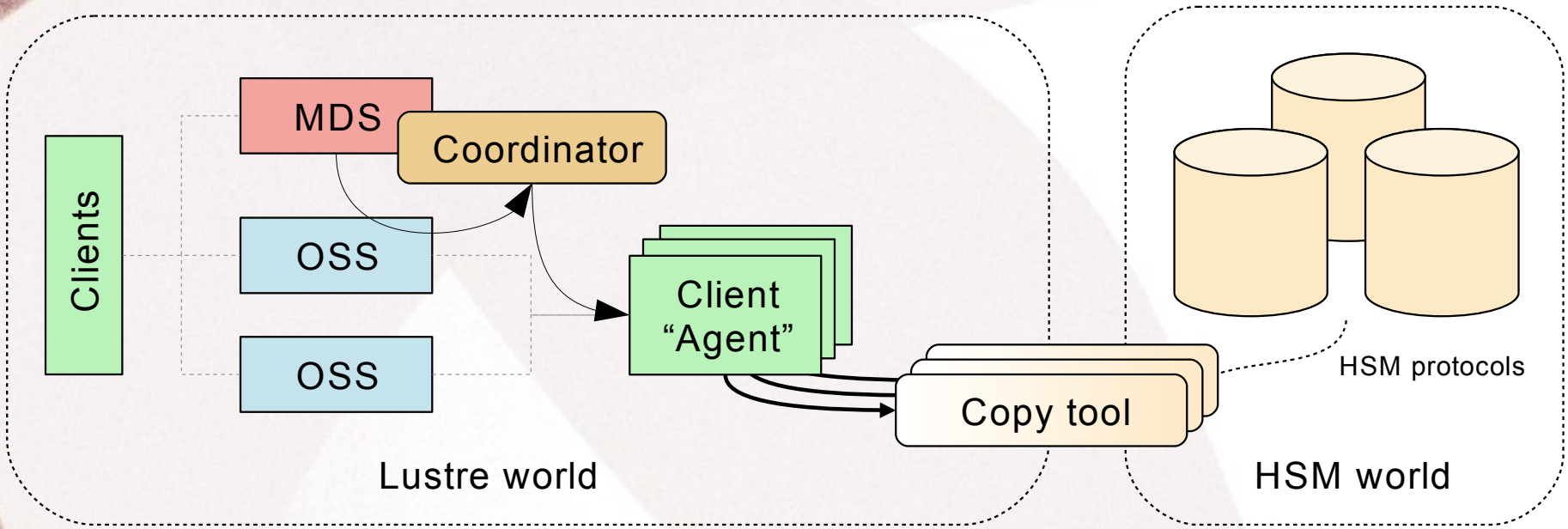
Sur une machine cliente d'un ensemble de serveurs NFS+Lustre, on peut utiliser des points de montages de ce type (extrait d'une table d'automount) :

```
# FS my_fs, avec 3 pools

my_fs  /scratch     nas1(1),nas2(3),nas3(5):/export/&/scratch  \
       /produits    nas2(1),nas3(3),nas1(5):/export/&/produits\
       /sauve       nas3(1),nas1(3),nas2(5):/export/&/sauve
```

Stockage hiérarchique

Le HSM à la sauce Lustre



- Fonctionnalité standard de Lustre pour un HSM Posix, mais peut nécessiter des add-ons pour d'autres protocoles (S3 ou Rados Ceph)
- La décision de déplacer vers le HSM nécessite un « policy engine ». Robinhood est très souvent cité avec Lustre

Stockage hiérarchique

Ceph Tier 2 de Lustre ?

- Ceph est par nature un bon candidat pour assurer le stockage d'un système d'archivage
- Dans le cas d'un archivage économique de type HSM, il peut être couplé avec Lustre avec un module copytoolRados
- A travers sa crushmap, Ceph permet de définir les dépendances entre des éléments d'un stockage
  - Disk /dev/sda ∈ Serv1 ∈ Rack12 ∈ DataCenter3
  - Serv1 « Dépend » Elec1 « Dépend » Onduleur1
- On déterminer le nombre de copies des données et le type de risque que l'on veut éviter et Ceph répartit automatiquement les données à travers le cluster
- Les éléments d'un cluster Ceph n'ont pas besoin d'être très sûrs, puisque c'est le cluster qui assure intrinsèquement la fiabilité du stockage global
- Ceph accepte un niveau de dégradation (disque ou serveur HS par exemple) temporaire et permet de réparer avant que Ceph déclenche une réorganisation des données pour corriger la perte de ressources

Références

http://wiki.lustre.org

http://wiki.lustre.org/images/6/64/LustreArchitecture-v4.pdf

https://sourceforge.net/apps/trac/robinhood/wiki/Doc

https://github.com/ComputeCanada/lustre-obj-copytool

https://ceph.com

https://www.jres.org/fr/presentation?id=145

Stockage secondaire

Atempo TiNa + HSS