

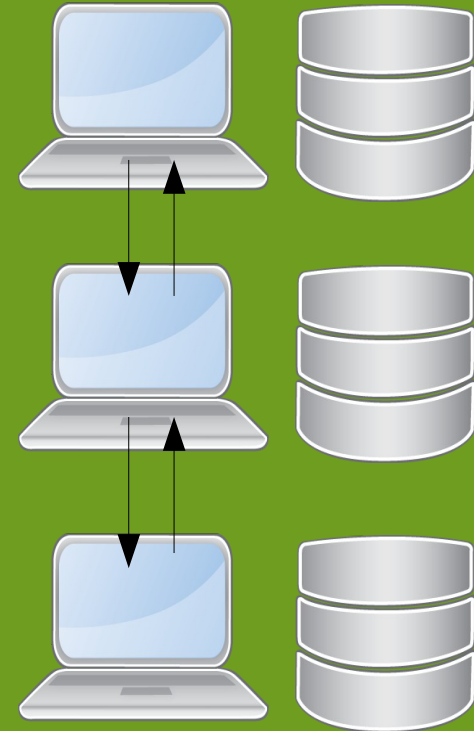
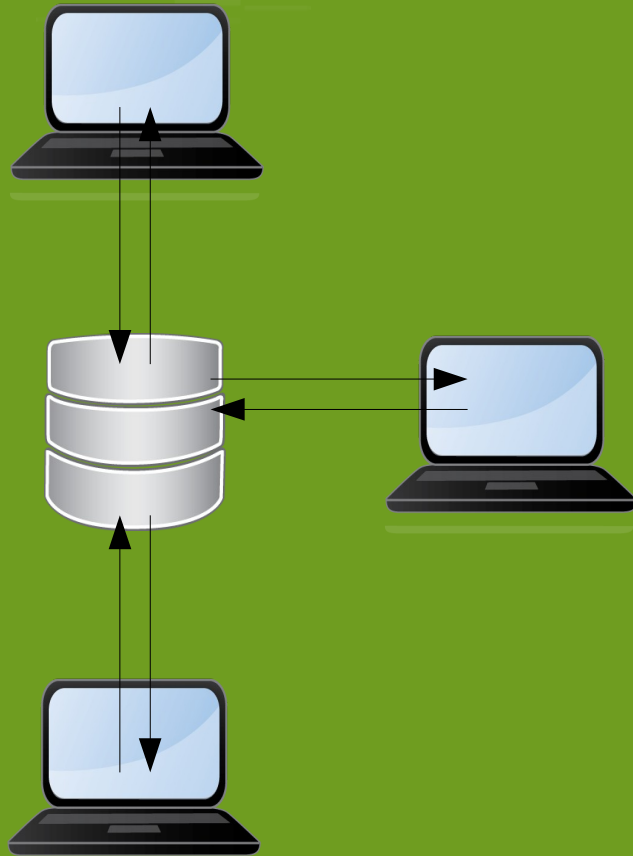


# STOCKAGE CENTRALISE

Formation cluster INRA - CNRS janvier 2016

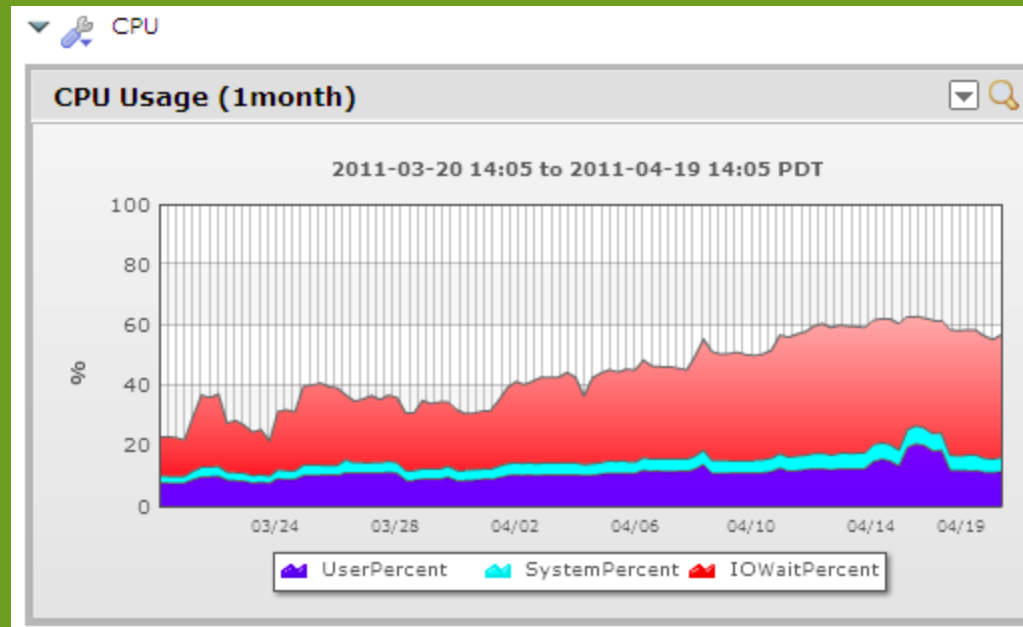


# STOCKAGE CENTRALISE / DISTRIBUE



Centralisation = SPOF

# PROBLEMATIQUE



# Architecture

## Comment choisir son stockage

❖ Où placer les données ? Que faut-il distribuer, que faut il « localiser » ?

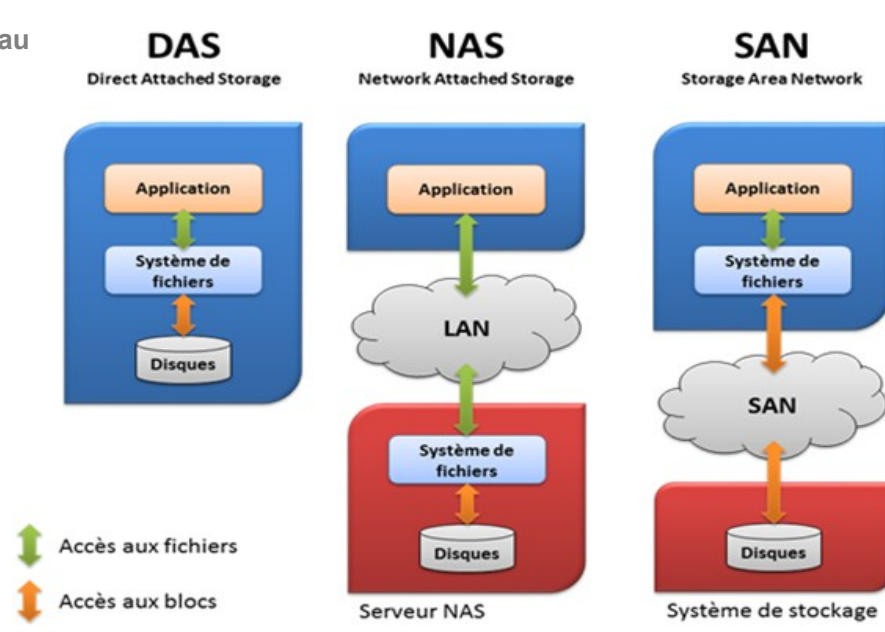
❖ Type de donnée (stockage, travail) ? Taille des fichiers ? Droits ?

❖ Disques locaux : près des processeurs mais peu d'axes

❖ Disques NAS : Importance de la liaison réseau

❖ Disques SAN : couteux

❖ Séparer le réseau de donnée et calcul



# Architectures

## Données localisées

- ❖ Fichiers de petite tailles
- ❖ Spool, config, etc = données du nœud, qui ne bougent pas
- ❖ Disque performant (SSD, RAID 0, SAS)
- ❖ Performances dépendent :
  - ❖ Caractéristiques physique du disque
  - ❖ Filesystem
- ❖ Désavantages :
  - ❖ Débug moins facile
  - ❖ Administration plus compliquée
  - ❖ Beaucoup de copie de fichier

# Architectures

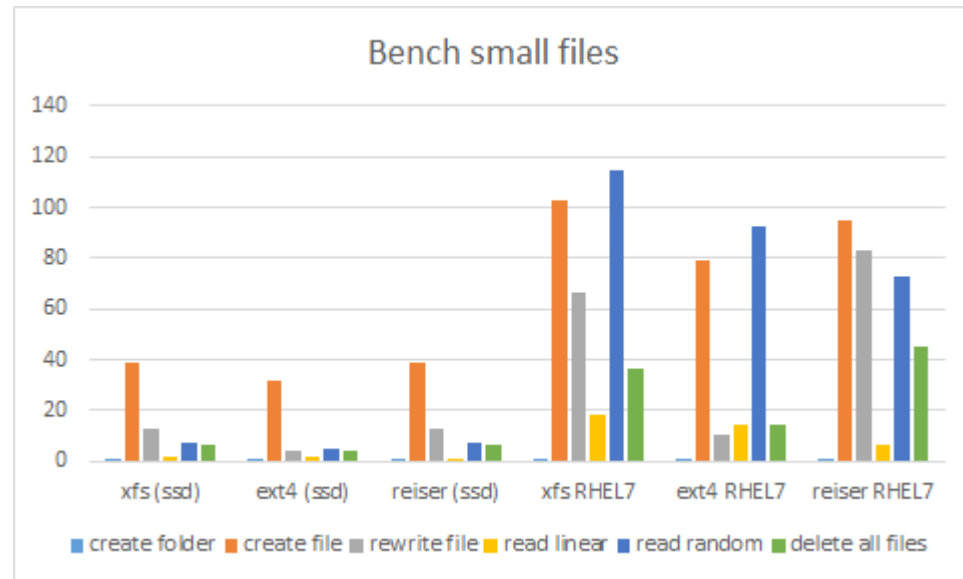
## Filesystems locaux

- ❖ Ext4 : le plus courant, par défaut dans RHEL 6
- ❖ ZFS : facile d'utilisation, stable, plein de fonctionnalités, sans limites de taille mais performances liées à l'archi
- ❖ XFS : performant mais pas très bon pour petits fichiers. Par défaut dans RHEL 7
- ❖ ReiserFS : stable, performant pour petits fichiers

# Architectures

## Benchmarks

- ❖ Create 300 000 files (512B to 1536B) with data from /dev/urandom
- ❖ Rewrite 30 000 random files and change the size
- ❖ Read 30 000 sequential files
- ❖ Read 30 000 random files



source <https://github.com/sysmsso/sysadmin/blob/master/bench/bench.py>

# Architectures

## NFS classique

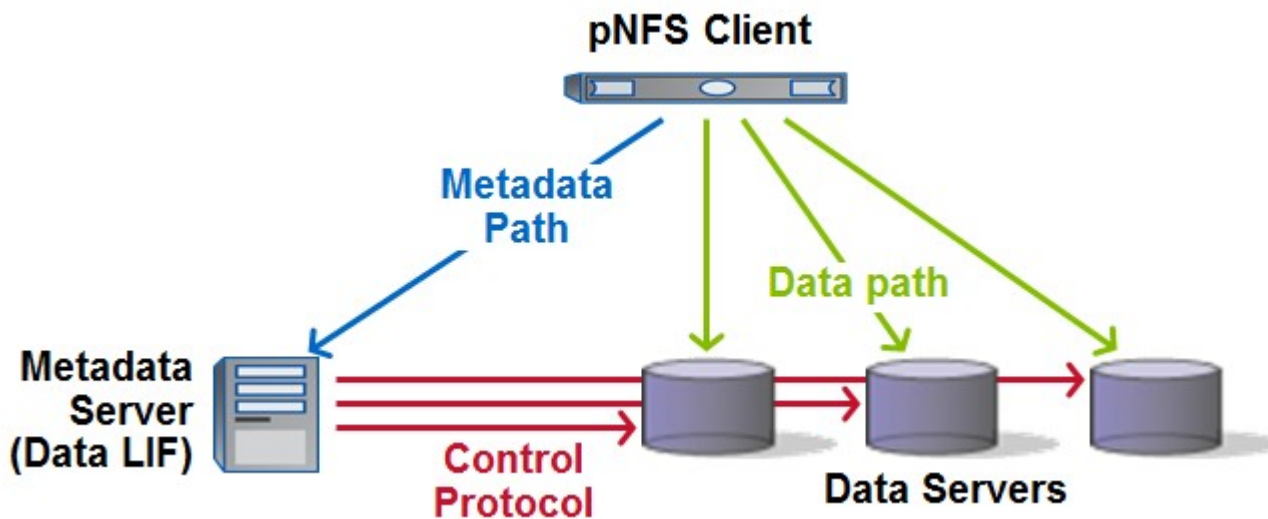
- ❖ NFS avec une baie NAS
- ❖ Un nœud « arbitre » qui localise et assemble les données pour le client
  - ❖ Augmente latence
  - ❖ Débits faibles
- ❖ Conseillé pour logiciels, configs, stockage
- ❖ Délicat pour I/O intensif
- ❖ Gestion des écritures concurrentes pas toujours facile



# Architectures

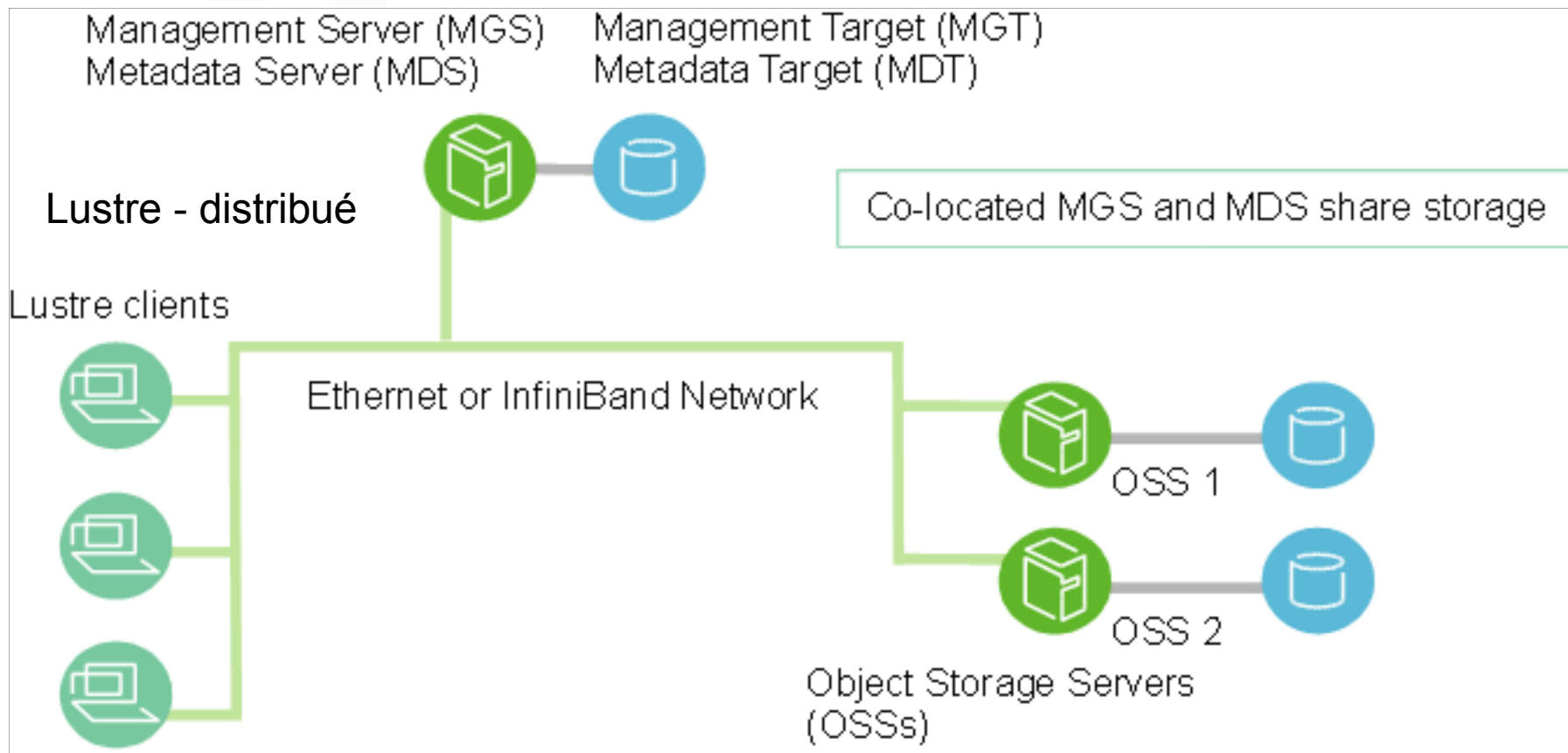
## Passer au distribué

- ❖ Contourner les limitations de NFS avec pNFS
- ❖ Plus de nœud « arbitre », déport au niveau des clients et du serveur de metadata
- ❖ La RFC 5661 définit ainsi les accès en mode fichiers → NetApp
- ❖ La RFC 5663 en mode bloc → EMC
- ❖ La RFC 5664 en mode objet → Panasas
- ❖ Parallélisme : séparation des requêtes de métadonnées de celles portant sur les données elles-mêmes



# Architectures

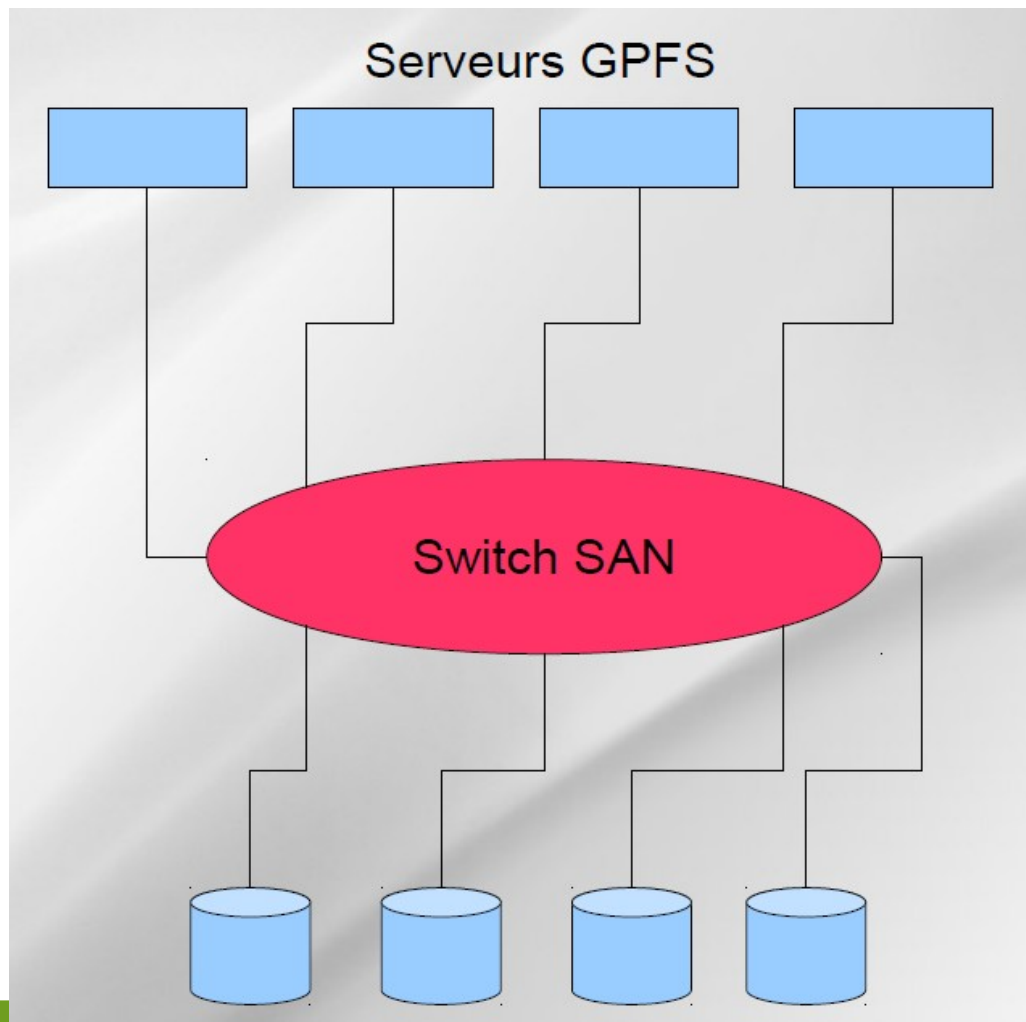
Distribué



# Architectures

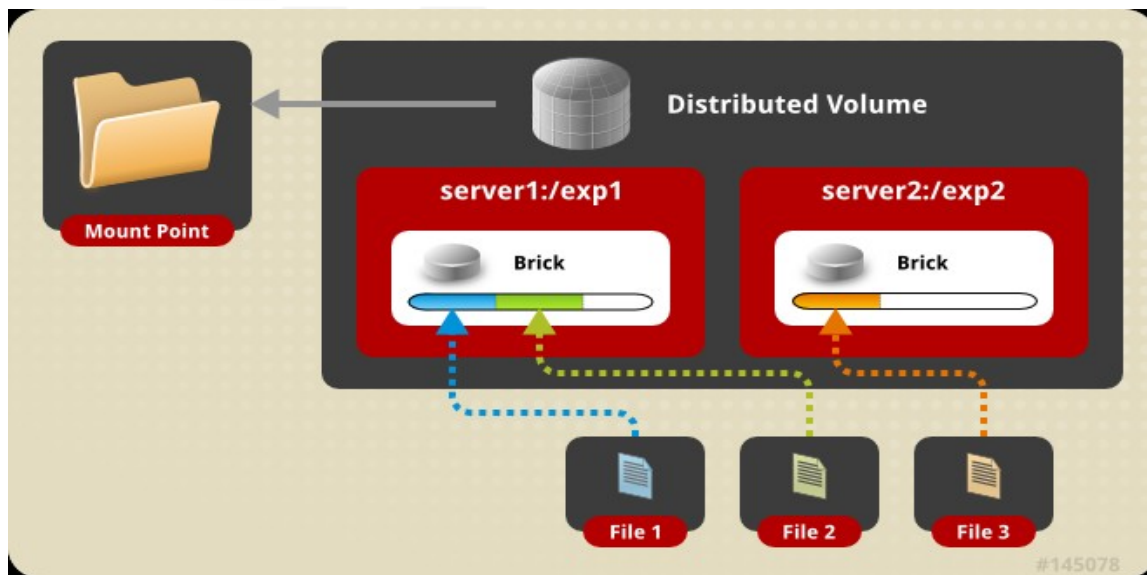
Distribué

GPFS

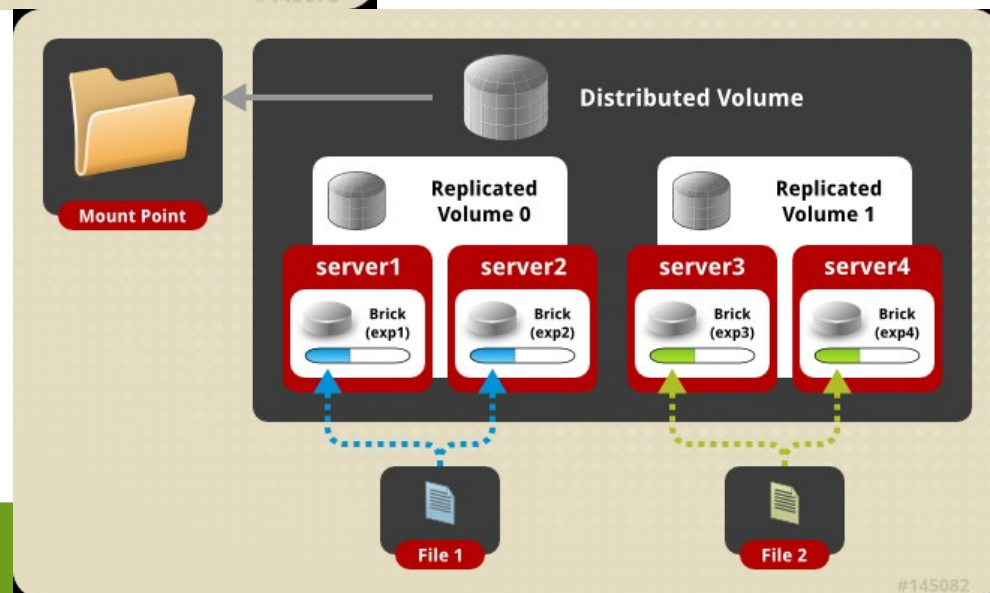


# Architectures

Distribu 



Glusterfs - distribu 





# RETOUR D'EXPÉRIENCE

## Cas concret



◆ PANASAS au CTIG

◆ GPFS a Toulouse