

TutoJres18 : GlusterFS

De la “paillasse”
au “génie des procédés”

“La simplicité est la sophistication extrême...”
Léonard de Vinci

Emmanuel Quémener

L'articulation PSMN/CBP à l'ENS-Lyon (et à la FLMSN)

Centre Blaise Pascal

- « Maison de la simulation »
- Hôtel à projets & conférences
- Hôtel à formations
- Un centre d'essais



Dryden Flight Research Center EC87 0182-14 Photographed 1987
X-29

Pôle Scientifique de Modélisation Numérique

- Méso-centre Equip@Meso
 - 6.4k cœurs, 1.5PB stockage
- Une usine de production



Stockage distribué & Centre Blaise Pascal

Déjà une longue histoire...

- Petit retour en arrière :
 - Idée lancinante depuis plus une dizaine d'années :
 - Proposition à l'ENS de Cachan en 2005
 - Faisabilité opérationnelle à l'époque difficile
 - Enquête besoins de stockage « recherche » de l'ENS-Lyon 2010T1
 - Conclusion : très onéreux de tout mettre en iSCSI propriétaire
 - Exploration d'autres solutions plus internalisées
 - Plus d'informations : JRES 2011, besoins de stockage
 - Étude via Distonet (Distonet pour Distributed Storage Network)
 - Exploration sur Cluster de AoE, iSCSI, CephFS, XtremFS et GlusterFS

DiStoNet : résultats encourageants

Mais quelle solution *Hic et Nunc* ?

- CephFS : pas encore au point...
- XtremFS : peu maintenu
- AoE : simple mais peu secure
- ISCSI : élaboré mais maintenance difficile...
- GlusterFS : en attendant d'avoir mieux...

GlusterFS : du KISS à l'état pur !

Au moins pour l'installation...

- KISS : *Keep It Simple Stupid*
- Simple à installer :
 - Côté serveur :
 - Pas de patch noyau profond (imposant la distribution GNU/Linux)
 - Pas de module à installer (imposant souvent au moins la version de noyau)
 - Pas de module à compiler (imposant une gestion au changement de noyau)
 - Juste un paquet à installer : `apt-get install glusterfs-server` suffit !
 - Côté client :
 - Pareil que pour le serveur
 - Juste un paquet à installer : `apt-get install glusterfs-client`

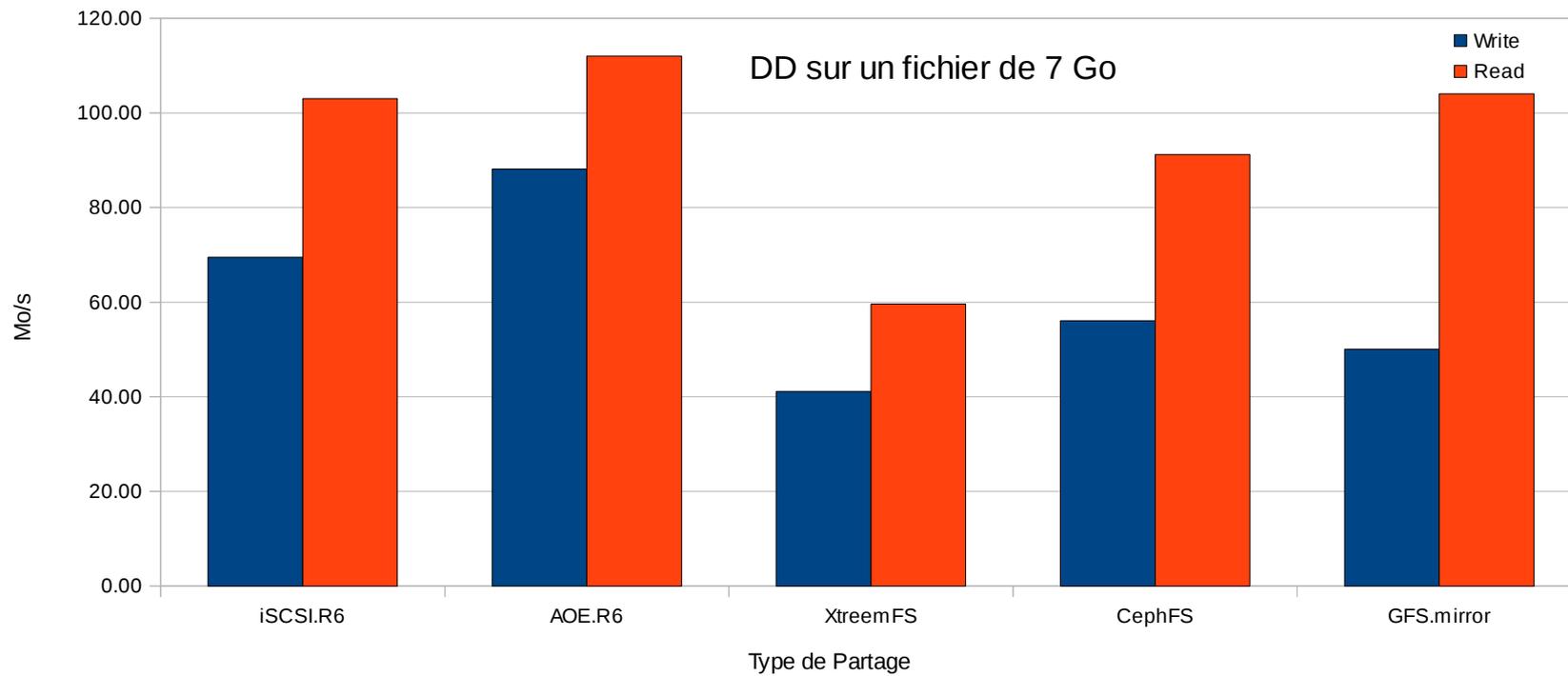
De l'installation à la configuration

Un parcours du combattant ?

- Côté serveur : 4 étapes
 - Définition des « voisins » (ceux qui participent au stockage)
 - Définition du volume en précisant :
 - Son nom : (pas trop tarabiscoté quand même)
 - Son type : équivalent RAID0 ou RAID1, ou RAID10
 - Son type de réseau : TCP/IP ou RDMA (InfiniBand)
 - Ses « participants » : chaque voisin et le point de montage
 - Activation du volume (et c'est tout)...
- Côté client :
 - Montage du volume (via client Fuse, facteur limitant)

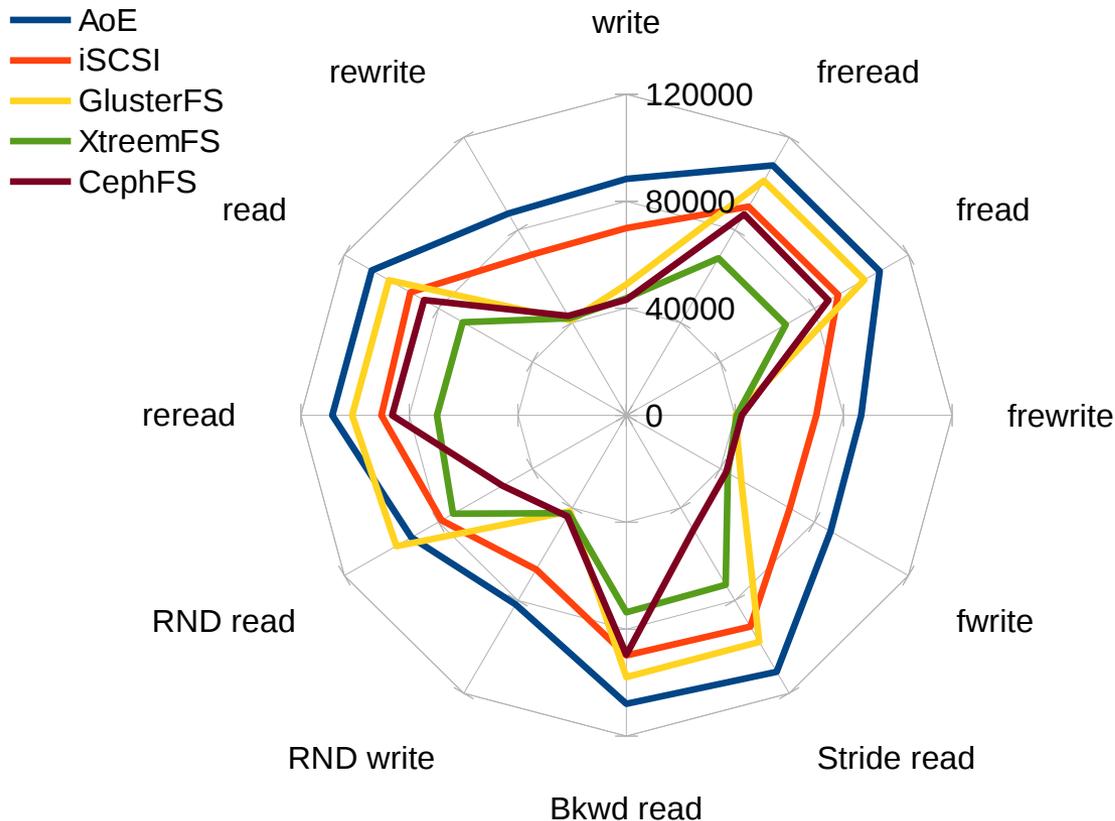
Des comparatifs en 2011 GlusterFS vs les autres...

Un dd sur des blocs de 8 machines en GigE, W/R



Des comparatifs en 2011 GlusterFS vs les autres...

Un IOzone3 sur des blocs de 8 machines en GigE



Étude pour le PSMN en 2013

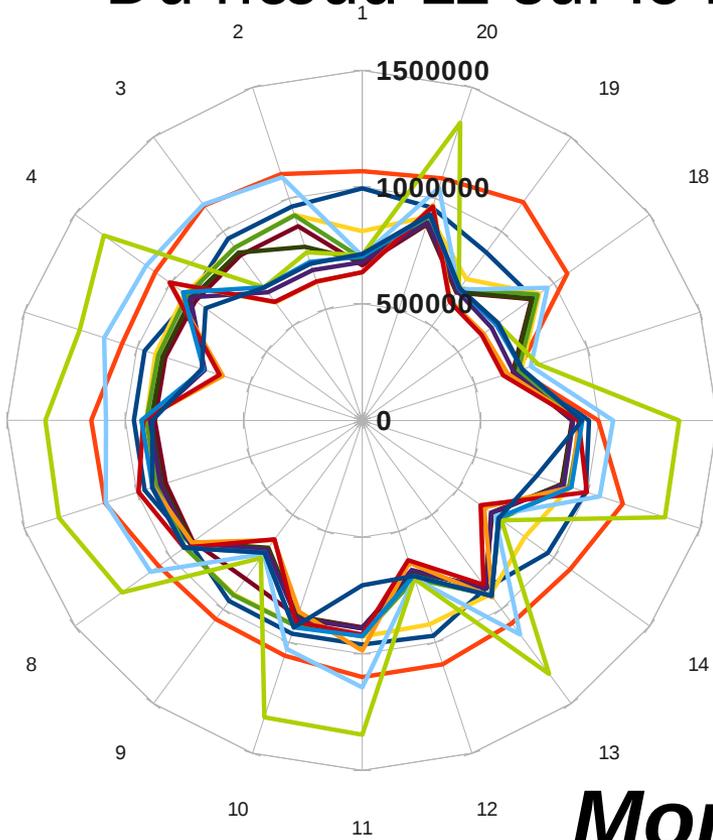
Là où la reproductibilité s'invite...

- Objectif :
 - Évaluation de GlusterFS comme /scratch de haute performance
- Plate-forme d'expérimentation : 20 nœuds + infrastructure
 - 20 nœuds Sandy Bridge 2x8 cœurs avec 64 GB de RAM
 - Un système SIDUS Debian Wheezy
 - Interconnexion InfiniBand FDR 56 Gb/s
 - Pas de latence disque : RamDisk BRD/Ext2 et TMPFS de 60 GB
 - 10 paires GlusterFS : 1 serveur sur RamDisk, 1 client
 - Usage de IOZone3 : 13 tests de lecture/écriture
 - 20 expériences pour un échantillon statistique représentatif

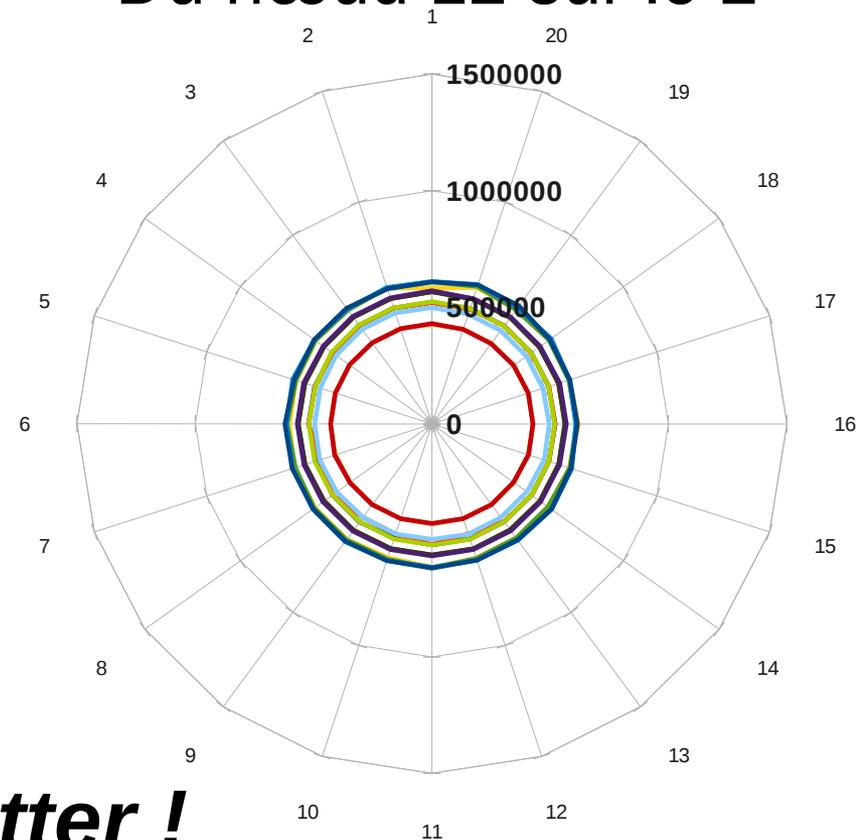
Étude pour le PSMN en 2013

Des comportements « étranges »

Du nœud 11 sur le 1



Du nœud 12 sur le 2



- Write
- Rewrite
- Read
- Reread
- Rnd read
- Rnd write
- Bkwd read
- Record rewrite
- Stride read
- Fwrite
- Frewrite
- Fread
- Freread

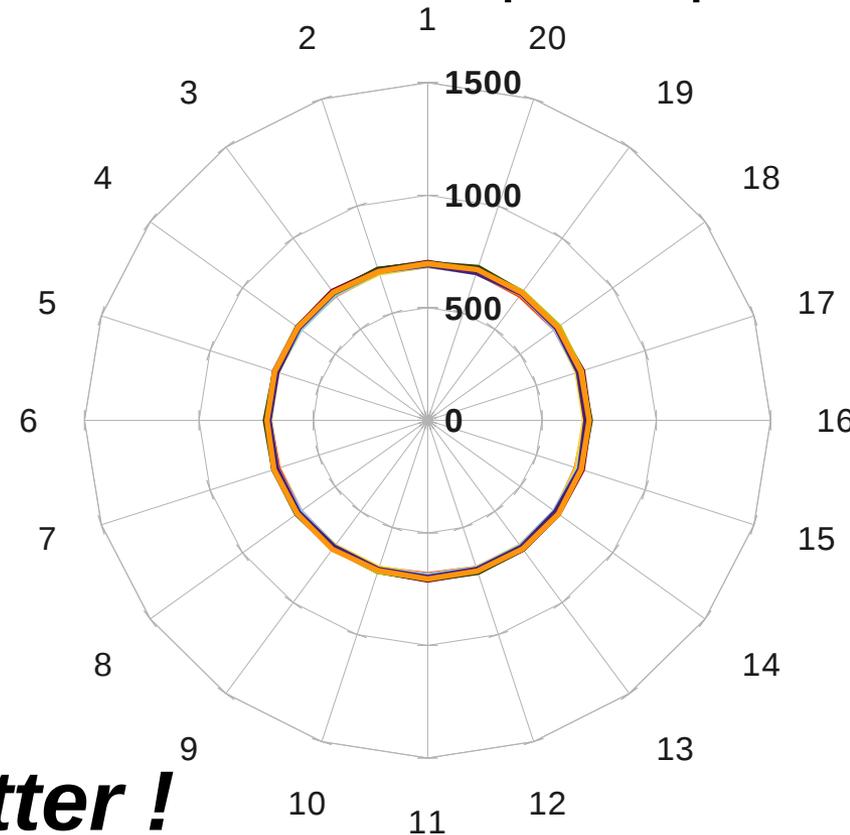
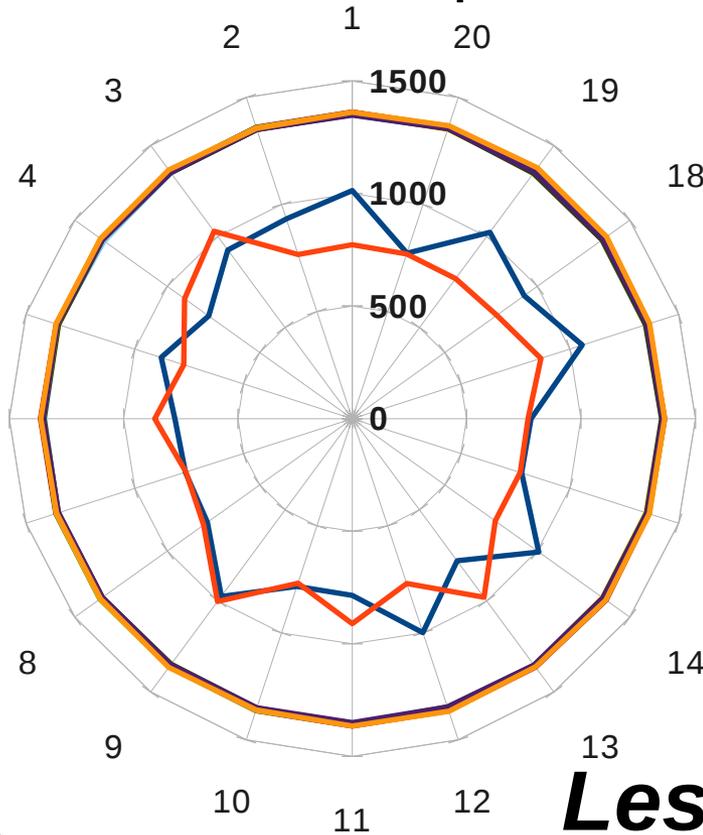
More is Better !

Étude pour le PSMN en 2013

Et une variabilité reproductible !

Pour les 10 couples avant

Pour les 10 couples après



- 11v1
- 12v2
- 13v3
- 14v4
- 15v5
- 16v6
- 17v7
- 18v8
- 19v9
- 20v10

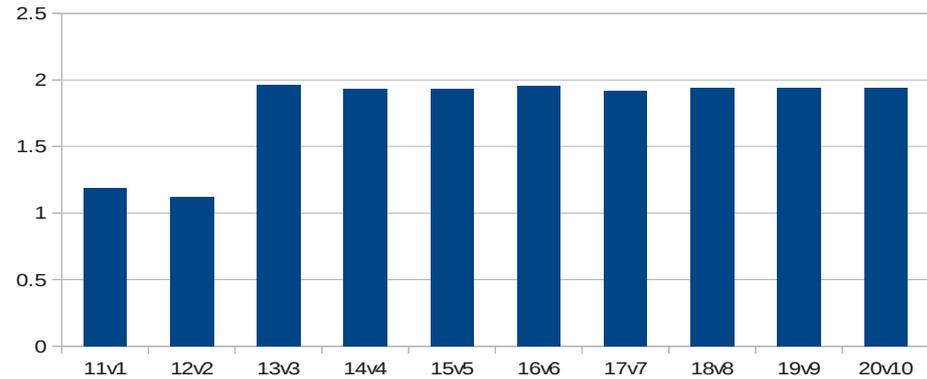
Less is Better !

Étude pour le PSMN en 2013

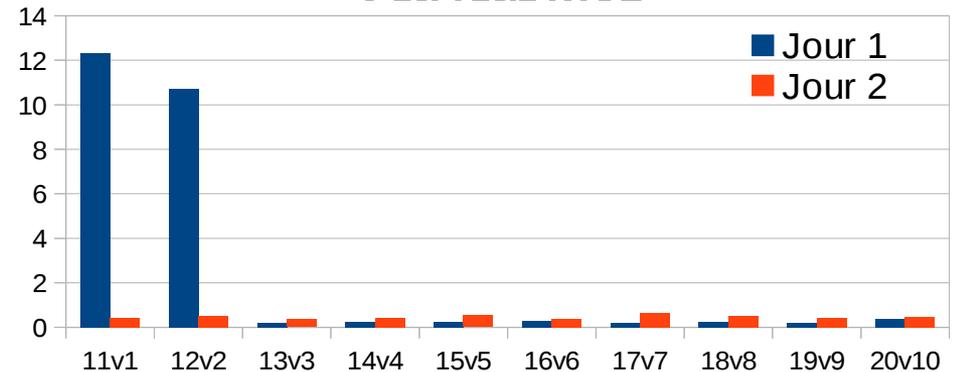
GlusterFS : au-delà du stockage !

- Deux questions :
 - Comment multiplier par 2 la vitesse ?
 - Comment diviser de 20 à 30 sa variabilité ?
- La réponse :
 - Optimiser le réseau ? Non
 - Optimiser les noyaux des OS ? Non
 - Changer les paramètres du BIOS ? OUI !!!
 - BIOS des nœuds 1 & 2 en Max Performance
 - BIOS des nœuds 3 à 20 par défaut
- Solution : uniformiser les BIOS
 - en Maximum Performance !

Accélération



Variabilité



Implémentation GlusterFS/PSMN

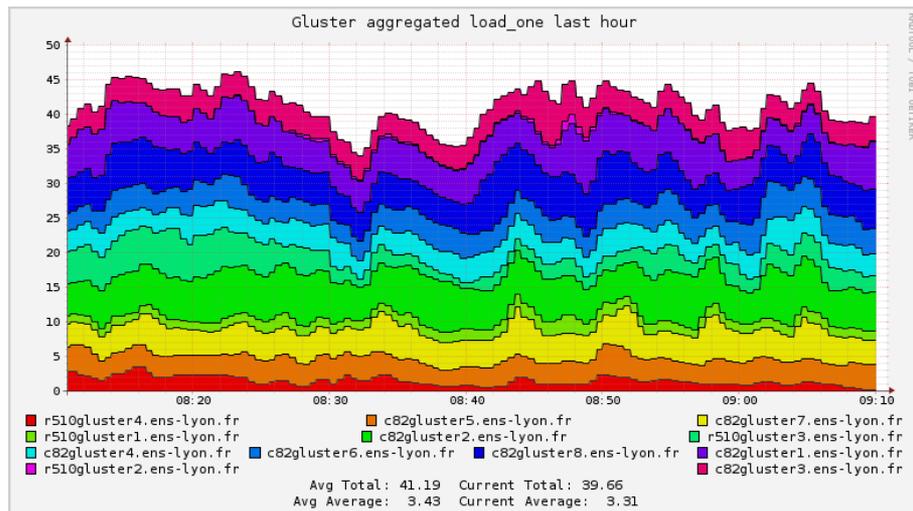
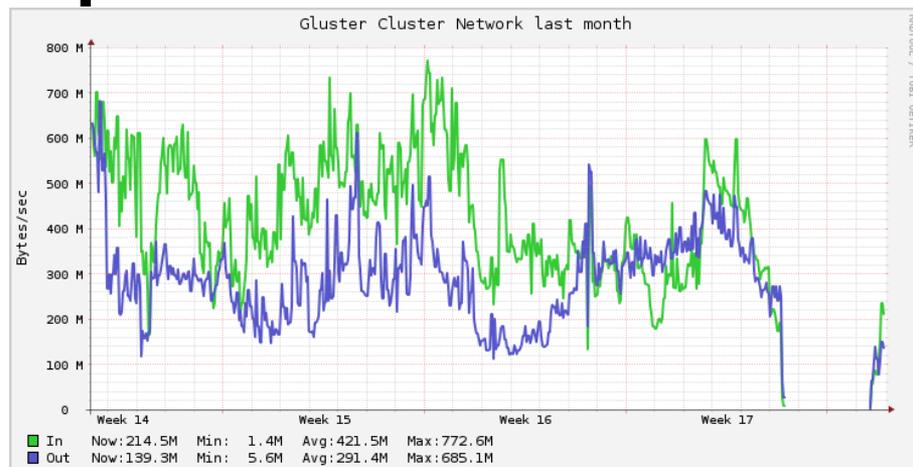
- Objectif :
 - Utiliser tous les disques disponibles pour du stockage distribué
 - Exploiter l'Infiniband FDR 56 Gb/s
- Implémentation :
 - Tous les disques : ZFSonLinux RAIDz1 pour le stockage de données
 - Système distant avec persistance : démarrage iSCSI
 - Création d'une matrice et clonage sous ZFSonLinux (snapshot & clone)
- Configuration : 8 nœuds c8220XD pour 250 nœuds
 - Stockage : 15 disques durs de 1TB (5 internes + 12 externes)
 - Puissance : E5-2670 16 cœurs à 2.6GHz & 64GB de RAM

GlusterFS au PSMN

La conf' & la perf' : pas mal, non ?

```
(root) e5-2670comp1.psmn - Konsole
File Edit View Bookmarks Settings Help
root@c82gluster4:~# gluster volume info

Volume Name: scratch
Type: Distributed-Replicate
Status: Started
Number of Bricks: 4 x 2 = 8
Transport-type: tcp
Bricks:
Brick1: 10.50.82.1:/media/toshare
Brick2: 10.50.82.2:/media/toshare
Brick3: 10.50.82.3:/media/toshare
Brick4: 10.50.82.4:/media/toshare
Brick5: 10.50.82.5:/media/toshare
Brick6: 10.50.82.6:/media/toshare
Brick7: 10.50.82.7:/media/toshare
Brick8: 10.50.82.8:/media/toshare
Options Reconfigured:
performance.cache-size: 2048MB
performance.io-thread-count: 32
nfs.disable: off
performance.flush-behind: on
performance.write-behind-window-size: 1024MB
performance.cache-refresh-timeout: 1
root@c82gluster4:~#
```



GlusterFS pour la biologie ou le cauchemar se « *repeat** »

- Faire tourner la suite Repeat(Masker|Modeler)
 - Processus de (3|6) « produits » très hétérogènes
 - A petite échelle (portable ou station de travail) : OK
 - A plus grande échelle, passage au PSMN : KO
 - Plantage du serveur NFS (avec un Loïs « zorgieusement désappointé »)
- Exploration du « comportement » au CBP
 - Lancement sur station de travail, disque SSD
 - Lancement sur nœud de cluster, espace GlusterFS (un serveur)
 - Lancement sur serveur rapide, espace Ramdisk

GlusterFS pour la biologie

Lorsque les I/O font leur loi...

2013	Bases	Sequences	Files	User	Elapsed	Input/Output
Pelodiscus	2202483752	19904	309531	620394	394995	2073771400
Tetraodon	358618246	27	403873	1040280	453677	1684136336
Killifish	1230898532	6012	376817	773775	531665	2531555536
2015						
Amphiprion	870234352	1081012	1193669	1244091	1036729	530305664
Roussette	4311024850	3502619	743440	1662646	1906300	271617672

En regardant plus finement avec « /usr/bin/time »

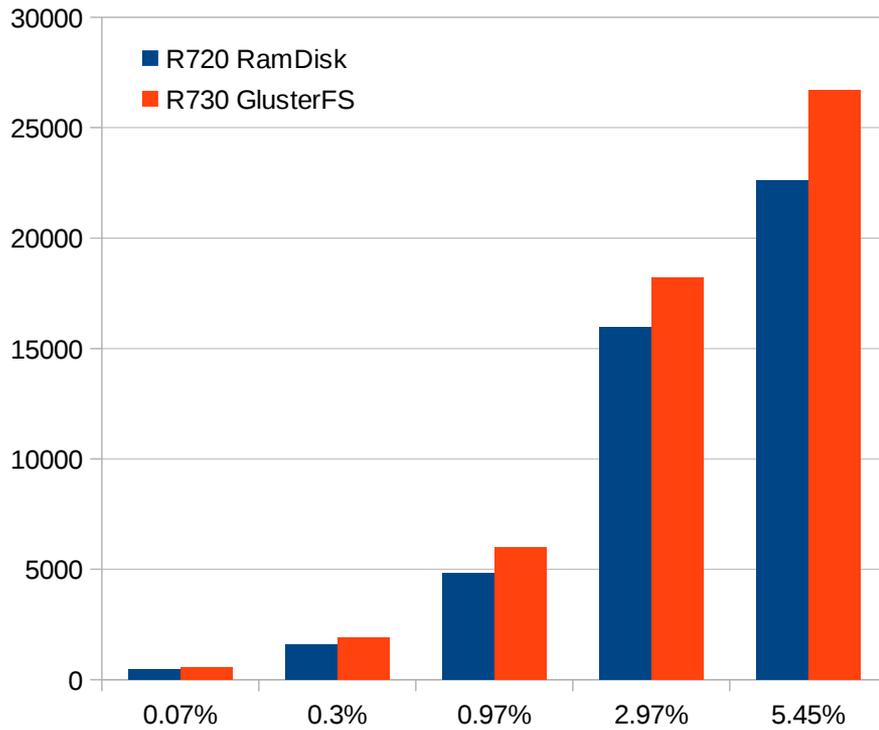
2013	User	System	Elapsed	Input/Output	Data Rate
Pelodiscus	620394	2209	394995	2073771400	5250
Tetraodon	1040280	1565	453677	1684136336	3712
Killifish	773775	2316	531665	2531555536	4762
2015					
Amphiprion	1244091	193623	1036729	530305664	512
Roussette	1662646	372336	1906300	271617672	142

Le Match : RamDisk vs GlusterFS

GlusterFS : crash, mais pourquoi ?

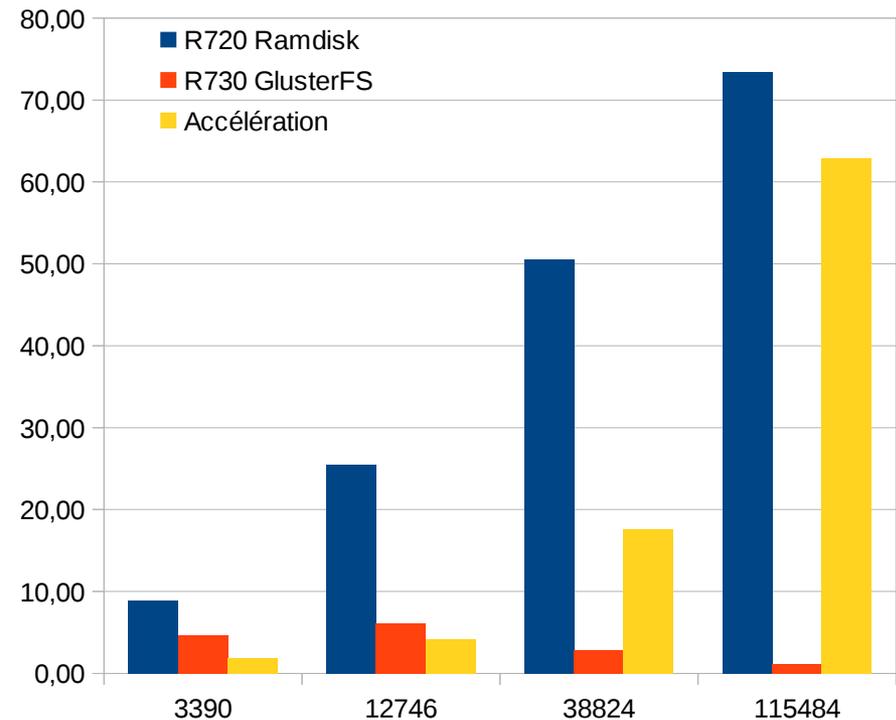
Progression « *Input Database Coverage* »

Less is Better !



Progression « *Family Refinement* »

Best is Better !



Quel enseignement de repeat* ?

GlusterFS : pas la panacée...

- La panacée n'existe pas :
 - en matière de stockage, de traitement, de « modes »
- Deux nécessités :
 - Exploration la plus fine possible sur ensemble « pertinent »
 - Diversité des installations pour explorer les différentes approches
- Mais quel facteur limitant pour GlusterFS ?
 - Exploration sur un serveur GlusterFS distribué...

Petit exemple récent

Création d'un pool de 4 pairs

- La configuration matérielle : 4 nœuds Sunfire x4150
 - 8 cœurs E5440 à 2.9GHz, 32 GB de RAM,
 - 8 disques NearLine SAS de 1TB en 2.5 pouces
 - Infrastructure InfiniBand DDR sur x4150 et QDR sur R410
- La configuration logicielle : SIDUS sur Debian Jessie
 - Solution « diskless » pour le système
 - Solution ZFSonLinux 0.6.5.6, RaidZ1 sur 8 disques de 1TB
- Nécessité d'une persistance de données...
 - 3 volumes ZFS pour 3 dossiers : GlusterData, GlusterLib, GlusterLog

D'abord, la « cuisine » ZFS

Création des espaces...

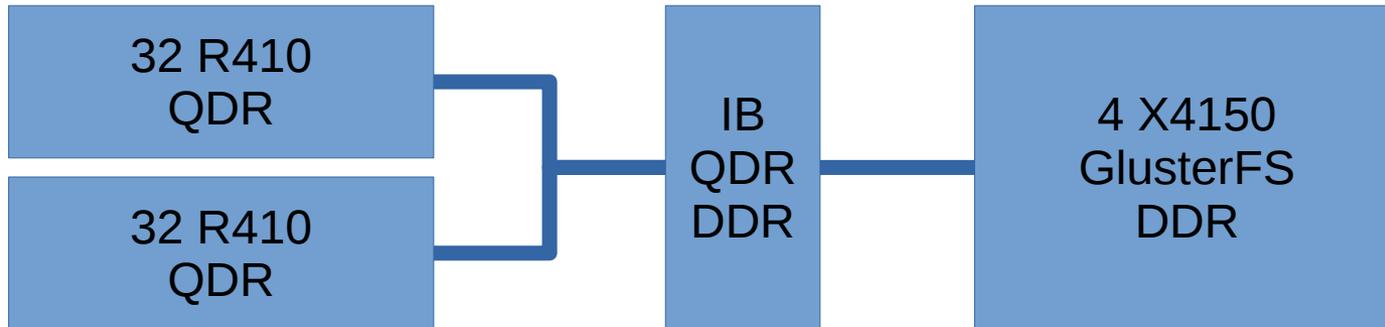
- Création du pool ZFS sur chaque nœud
 - `clush -w x41z[1-4] 'zpool create -o ashift=12 -f raid5 raidz $(ls /dev/disk/by-id/wwn* | tr "\n" " ")'`
- Tuning du ZFS
 - `clush -w x41z[1-4] "echo $(($(free | grep Mem | awk '{ print $2 }')/4*1024)) > /sys/module/zfs/parameters/zfs_arc_max"`
- Tuning des volumes
 - `clush -w x41z[1-4] zfs set atime=off raid5`
 - `clush -w x41z[1-4] zfs set compress=lz4 raid5`
- Création des volumes ZFS
 - `clush -w x41z[1-4] zfs create raid5/GlusterData`
 - `clush -w x41z[1-4] zfs create -o mountpoint=/var/lib/glusterd raid5/GlusterLib`
 - `clush -w x41z[1-4] zfs create -o mountpoint=/var/log/glusterfs raid5/GlusterLog`

Puis, la configuration GlusterFS

Association, création, démarrage...

- Activation du serveur GlusterFS sur chaque pair
 - `clush -w x41z[1-4] /etc/init.d/glusterfs-server start`
- Ajout de chaque pair à la liste
 - `ssh x41z1 gluster peer probe 10.11.12.52 ;`
 - `ssh x41z1 gluster peer probe 10.11.12.53 ;`
 - `ssh x41z1 gluster peer probe 10.11.12.54`
- Création de l'espace partagé de nom distonet, en RDMA & TCP
 - `ssh x41z1 gluster volume create distonet transport rdma,tcp 10.11.12.51:/raid5/GlusterData/ 10.11.12.52:/raid5/GlusterData/ 10.11.12.53:/raid5/GlusterData/ 10.11.12.54:/raid5/GlusterData/ force`
- Démarrage du volume
 - `ssh x41z1 gluster volume start distonet`
- Visualisation du volume
 - `ssh x41z1 gluster volume info`

Ensuite, le montage sur les nœuds



```
(root) lethe - Konsole <3>
File Edit View Bookmarks Settings Help
root@x41z1:~# gluster volume info
Volume Name: distonet
Type: Distribute
Volume ID: 651f79b0-4ad3-4739-8d8c-aa0de9bcfa56
Status: Started
Number of Bricks: 4
Transport-type: tcp,rdma
Bricks:
Brick1: 10.11.12.51:/raid5/GlusterData
Brick2: 10.11.12.52:/raid5/GlusterData
Brick3: 10.11.12.53:/raid5/GlusterData
Brick4: 10.11.12.54:/raid5/GlusterData
Options Reconfigured:
performance.cache-refresh-timeout: 1
performance.flush-behind: on
performance.io-cache: on
performance.write-behind-window-size: 1073741824
performance.cache-max-file-size: 4096KB
performance.cache-size: 4096KB
performance.io-thread-count: 64
nfs.disable: on
root@x41z1:~#
```

- Pour 64 nœuds Dell R410
 - Interconnexion Infiniband QDR
- Pour le montage sur 64 nœuds Dell R410
 - `clush -w r410node[1-64] 'mount.glusterfs 10.11.12.$((${echo $HOSTNAME} | sed -e "s/r410node//g")%4+51)):distonet /distonet'`

Enfin, l'expérimentation

Après la conf', la perf'

- Expériences : accès massif & simultané des 64 nœuds
 - Chaque nœud écrit dans son dossier ou dans le même, salve de 16
 - Opération d'écriture, de listage, de lecture et d'effacement
 - Nombre de fichiers de 512 à 65536 pour un volume total de 12GB

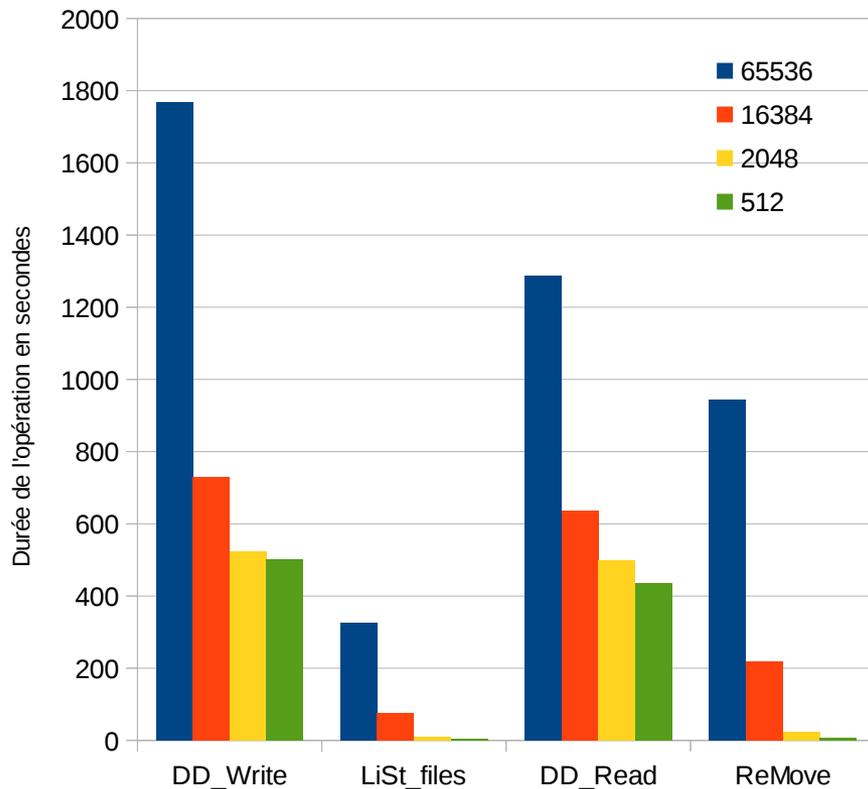
- Les lignes d'exécution, exemple pour 16384 :

- `(/usr/bin/time clush -w r410node[1-64] 'seq -w 1 1 16384 | /usr/bin/time xargs -P 16 -l '{}' dd if=/dev/zero of=/distonet/benchs/${HOSTNAME}-'{}'.raw bs=2048 count=360 ') > dd_w_2048_OneFolder_$(date "+%Y%m%d-%H%M").log 2>&1`
- `(/usr/bin/time clush -w r410node[1-64] '/usr/bin/time ls -l /distonet/benchs/${HOSTNAME}-* | wc -l') > ls_$(date "+%Y%m%d-%H%M").log 2>&1`
- `(/usr/bin/time clush -w r410node[1-64] 'seq -w 1 1 16384 | /usr/bin/time xargs -P 16 -l '{}' dd of=/dev/null if=/distonet/benchs/${HOSTNAME}-'{}'.raw bs=16384 count=360 ') > dd_r_16384_OneFolder_$(date "+%Y%m%d-%H%M").log 2>&1`
- `(/usr/bin/time clush -w r410node[1-64] '/usr/bin/time rm -f /distonet/benchs/${HOSTNAME}*') > rm_$(date "+%Y%m%d-%H%M").log 2>&1`

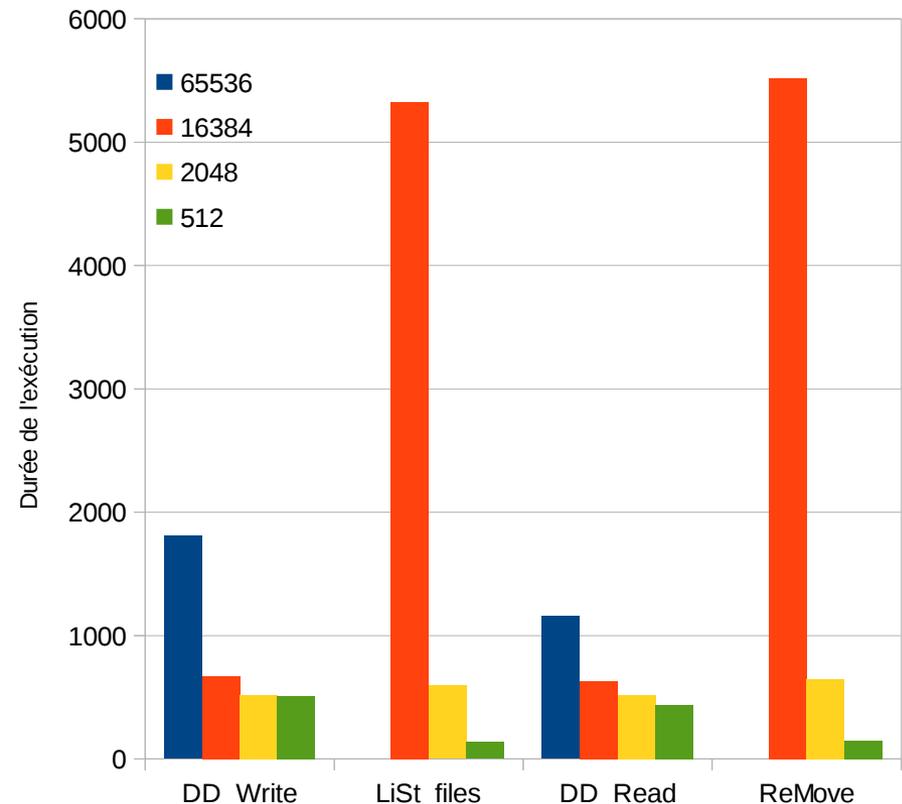
Des résultats instructifs...

Comment flinguer son partage ?

Chacun dans son dossier...



Tout dans le même !



Qualités & Défauts de GlusterFS

- Qualités :
 - Ses coûts « marginaux » :
 - Coût d'entrée : rien, ou presque...
 - Coût d'exploitation : pareil...
 - Coût de sortie : en s'arrangeant bien, le « socle » suffit...
 - Sa simplicité de mise à disposition & de partage
 - Son développement (et sa maintenance) par RedHat
- Défauts :
 - Sa gestion des quotas
 - Sa gestion des snapshots (OK avec du LVM, « manuelle » en ZFS)
 - Son absence de vrai « mode bloc » (mais TGT GlusterFS)
 - Sa sensibilité

En conclusion

- GlusterFS : **simple**, rapide, efficace
 - Mais pas universel (pas plus que les autres)
- A chaque « usage » son « espace »
 - L'usage : le mariage du traitement & des données
 - Réservation d'un type d'espace, même temporaire
 - « Mode bloc » devenant indispensable
- Ecosystème informatique en mutation
 - Avant, seulement les architectures de traitement (processeurs, réseau)
 - Même conclusion que Romaric David en introduction