Distributed filesystem experiments at the High Performance Computing Center of Strasbourg

Michel RINGENBACH
mir@unistra.fr
Direction Informatique
19/01/2016

UNIVERSITÉ DE STRASBOURG

Sujet

- Ce que n'est pas cette présentation
  - Un comparatif exhaustif entre solutions
  - Un cours théorique (voir Internet...)
- But de cette présentation
  - Voir des cas concrets et appliqués
  - Un historique, une évolution guidée
  - Des outils pour bencher, des résultats
  - Le tout à la frontière du centralisé/distribué
  - GPFS, BeeGFS, FluidFS, RozoFS

Plan

- HPC in Strasbourg University
- « My simulation is slow »
- Cascade effect
- Conclusion

High Performance Computing Center

This talk focuses on some extra experiments of the HPC Center of the University of Strasbourg (Unistra)

- Unistra is one of the major universities in France:
  - 48 000 students
  - 4800 employees
  - 3 Nobel prizes since 1987
  - major research institute in many scientific domains
  - ...some of them need HPC
- HPC Center (http://hpc.unistra.fr) serves the whole Alsace Region

High Performance Computing Center : Fundings

The HPC Center of the Unistra is funded by:

- Unistra: hosts the engineers responsible for the HPC Center
- The research labs fundings: until 2013, 100% of compute servers had been bought by the labs
  Labs are located not only in Strasbourg, but in all the Alsace region (too many logos to show)
- The French national initiative *Investissements d'Avenir*, *via* a national project: Equip@Meso
- French government, Alsace Region and Strasbourg Eurométropole

HPC in Strasbourg: Flops, Bytes....

- Around 350 servers, 5500 cores
- 500 TB of GPFS Storage (on departure)
- 576 TB of RozoFS storage (being installed)
- Many TB of BeeGFS
- 60 GPUs, from Tesla
  M2050 to K80
- 223 Tflops
- More than 250 active users
- More than 150
  softwaremodules

A team composed of 5 people:

- Operating all the HPC facilities (datacenter, clusters)
- Supporting more than 50 HPC scientific software by:
  - Defining a standard set of tools we strongly support: Intel compilers + in-house built OpenMPI, Cuda
  - In most cases, building/linking the scientific apps against these standard tools
  - **Writing and optimizing code**
- Doing all the training
- Promoting HPC for SMEs

Plan

- HPC in Strasbourg University
- « My simulation is slow »
- Cascade effect
- Conclusion

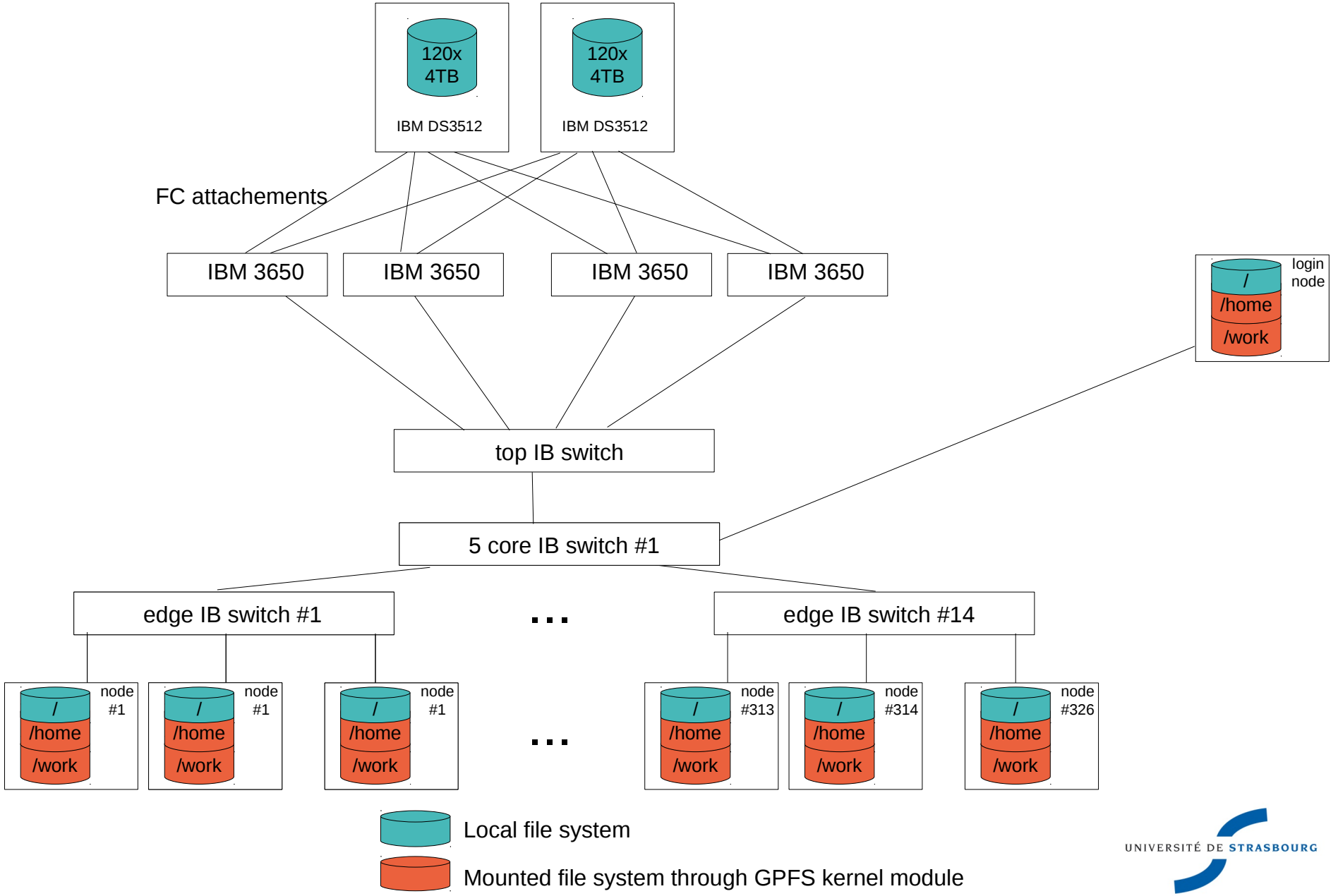
« My simulation is slow »

- Once upon a time...
  - We were challenged by users of the Relion application http://www2.mrc-lmb.cam.ac.uk/relion/index.php/Main_Page, used for reconstruction of 2D or 3D classes in cryo-electron microscopy
  - The execution times on the computing centers were much more longer than in the lab computers
  - Strategically not acceptable for the HPC Center
- At first we went wrong:
  - Compared speed-up against standard experiments of the author of the code (Sjors Scheres)
  - Profiled the application on users data-sets: I/O problems
  - Several hundreds of small files in real Datasets

« My simulation is slow »

- GPFS if responsible: 1GB/s or more but totally flooded when small files
- Bought in early 2012, first Relion alert : mid 2014
- Other big challenging applications arrived since

GPFS setup


120x
4TB
IBM DS3512
120x
4TB
IBM DS3512
FC attachements
IBM 3650
IBM 3650
IBM 3650
IBM 3650
login
node
/
/home
/work
top IB switch
5 core IB switch #1
edge IB switch #1
edge IB switch #14
node
#1
node
#313
node
#314
node
#326
Local file system
Mounted file system through GPFS kernel module

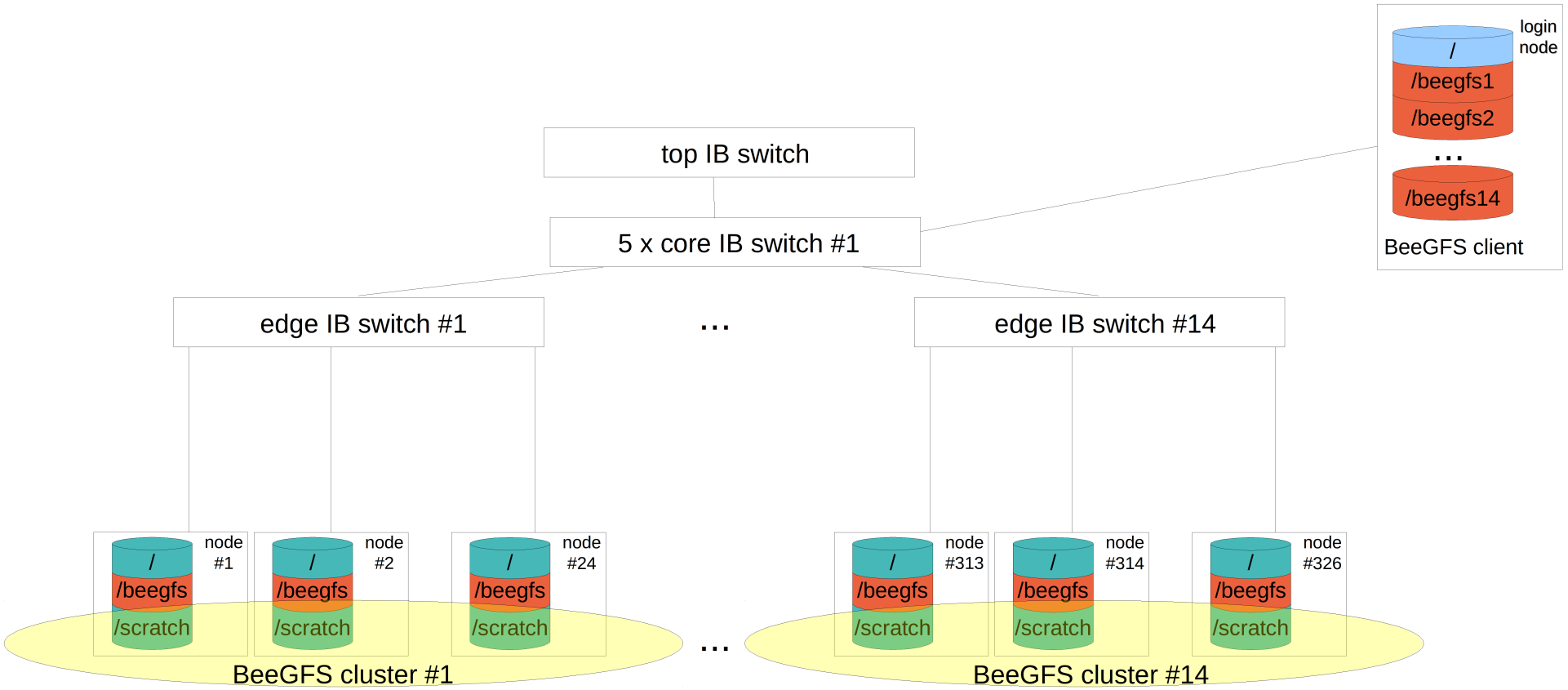
GPFS setup

- Can we scale-up the configuraton ?
- Our GPFS setup in not easy to extent
  - A lot of links
  - Have to partitionning disk space if add more GPFS serveurs
- New 4 TB disks not officially supported !
  - We had to test ourself
  - Materials up to date !

« My simulation is slow »

- In the meantime, we were trying BeeGFS to answer the question « what can we do with those nearly unused local disks on the compute nodes ? »
- BeeGFS is easy to deploy
- Can use nodes local disk as is
- Works on infiniband
- True SDS

BeeGFS setup



« My simulation is slow »

- Very simplified BeeGFS setup:
  - Uses system disk (or even /dev/shm on some test-cases)
  - Ext3 : Should be better with XFS (+ eventually RAID)
    - But was deployed during exploitation. Works fast.
  - Data lays on a specific directory, is visible outside BeeGFS
  - 1 Meta-data server per (max) 24 nodes
  - Volumes named after the IB switch they belong
- Which usage for this data ?
  - Temporary (scratch) data of jobs
  - No backup
  - Warning !!!!

BeeGFS setup

- A same node can be a :
  - management server
  - meta-data server
  - storage server
  - client
- We have standard and experimental setups :

IB#	NODES	#NODES	MGMT	META	STORAGE	CLIENTS
11	hpc-n[409-424]	16	409	409-410	411-424	all
12	hpc-n[430-460]	24	430	all	all	all
14	hpc-n[464-483]	20	480	480-483	480-483	464-479

BeeGFS setup


```
#! /bin/bash

mode=xdsh # mode=xdsh|postscript

nodename=$(hostname)

mgmtnode=switch_conf2mgmt_node($nodename) # Management node (must be part of the BeeGFS
    cluster)

SW=switch_conf2switch($nodename) # Management node (must be part of the BeeGFS cluster)

mountpoint=/scratch-beegfs # Mount point on clients (will be created at package deployment
    time)

ln -s $mountpoint $mountpoint-ib-edge${SW} # For coherency with multiple mounting on login
    node

mkdir /fhgfs

if test "$nodename" == "$mgmtnode"; then

   services="$services fhgfs-mgmtd"

   services="$services fhgfs-admon"

fi

services="$services fhgfs-storage"

services="$services fhgfs-meta"

services="$services fhgfs-client"

services="$services fhgfs-helperd"
```

BeeGFS setup


```
cd /etc/yum.repos.d # Install the repo and the packages

wget http://www.fhgfs.com/release/latest-stable/dists/fhgfs-rhel6.repo

yum -y install fhgfs-mgmtd fhgfs-meta fhgfs-storage fhgfs-admon install fhgfs-client
   fhgfs-helperd fhgfs-utils

# Must activate infiniband

sed -i 's/buildArgs=-j8/buildArgs=-j8 FHGFS_OPENTK_IBVERBS=1/' /etc/fhgfs/fhgfs-client-
   autobuild.conf

/etc/init.d/fhgfs-client rebuild

# Customization of configuration files

sed -i "s?storeMgmtdDirectory\s*=?storeMgmtdDirectory=/scratch/fhgfs/fhgfs_mgmtd?"
   /etc/fhgfs/fhgfs-mgmtd.conf

sed -i "s?storeMetaDirectory\s*=?storeMetaDirectory=/scratch/fhgfs/fhgfs_meta?"
   /etc/fhgfs/fhgfs-meta.conf

sed -i "s/sysMgmtdHost\s*=/sysMgmtdHost=$mgmtnode/" /etc/fhgfs/fhgfs-meta.conf

sed -i "s?storeStorageDirectory\s*=?
   storeStorageDirectory=/scratch/fhgfs/fhgfs_storage?" /etc/fhgfs/fhgfs-storage.conf

sed -i "s/sysMgmtdHost\s*=/sysMgmtdHost=$mgmtnode/" /etc/fhgfs/fhgfs-storage.conf

sed -i "s/sysMgmtdHost\s*=/sysMgmtdHost=$mgmtnode/" /etc/fhgfs/fhgfs-admon.conf

sed -i "s/sysMgmtdHost\s*=/sysMgmtdHost=$mgmtnode/" /etc/fhgfs/fhgfs-client.conf
```

BeeGFS setup


```
# Mounting point

echo "$mountpoint /etc/fhgfs/fhgfs-client.conf" > /etc/fhgfs/fhgfs-mounts.conf

# Activating services, depending on the node role

for service in fhgfs-mgmtd fhgfs-meta fhgfs-admon fhgfs-storage fhgfs-helperd fhgfs-
   client; do

   chkconfig $service off

done

for service in $services; do

   chkconfig $service on

done

# When not used at node installation time (postscript), we have to start the services,
   otherwise they are started after post installation boot

if test "$mode" == "postscript"; then exit; fi

for service in $services; do

   # Some sleep to ensure that mgmt, meta and storage servers are up, before launching
   clients.

   if test "$service" == "fhgfs-helperd"; then

      sleep 60

   fi

   service $service start
```

« My simulation is slow »

- Performances : GPFS / BeeGFS
  - BeeGFS: Maximum bandwidth (dd, large files): 1GB/s
  - GPFS: 1GB/s or more but totally flooded when small files
- How to use this scratch space?
  - Users have to deal with 2 namespaces: ***/home***, several ***/scratch-XYZ*** ← Named after the IB switch
  - Data staging mandatory (cp, **parallel cp**, ...)
  - Need to know where data is

« My simulation is slow »

- Users point of view
  - BeeGFS is great !
  - On the Relion code, speed up x 4
- What can users do with a 4x speedup ?
  - Run more simulations
  - Get more results
  - In this case, this lead to a publication in Nature

« My simulation is slow »

- Administrator point of view
  - Easy to deploy
  - *Sort of* fault-tolrant: generally when loosing a compute node part of a BeeGFS array, write can continue, 4/24 chances to loose datas
  - Free Storage : 0 more U needed and lots of TB !
  - BeeGFS is the scratch solution we promote and deploy
  - Filesystem space = Job Node space
  - Makes node maintenance more difficult
  - Disposable high performance storage !

Plan

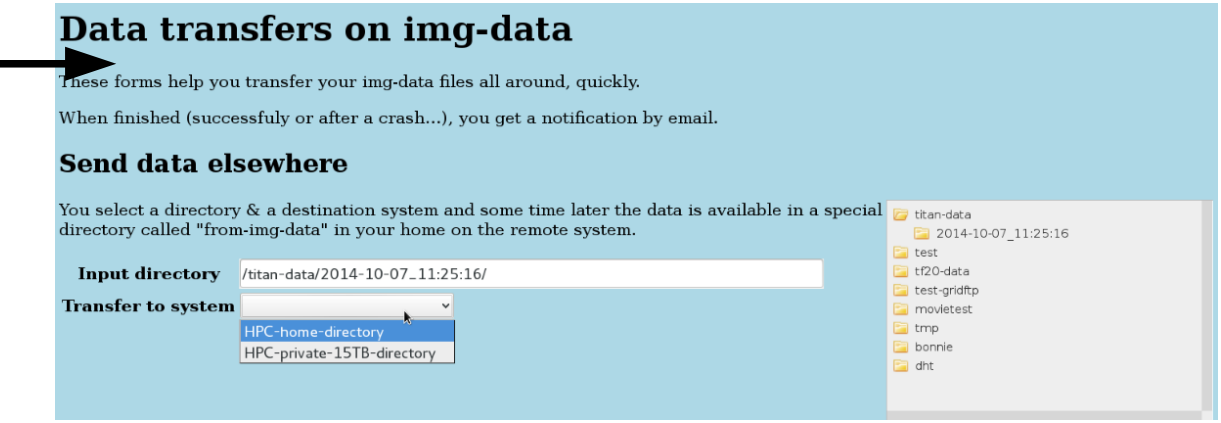
- HPC in Strasbourg University
- « My simulation is slow »
- Cascade effect
- Conclusion

Cascade effect

Now the users are able to deal with big data, how to transfer this data from the laboratory to the HPC ?

- We added a 10Gb metropolitan Vlan'ned network link: 5 kms between sites
- Transfer protocol: Grid-FTP with ssh authentication
- 700 MB/s point-to-point

Web interface developed at lab



Microscope
Hi-Res images

Cascade effect

Funny little things we never thought about before

- HPC Cluster = Nodes + IB
- Hold-on, what did I need Infiniband for, anyway?
  - IB is used for MPI
  - IB is used for GPFS
- For MPI, IB stay on the switch : since 2013, no job is allowed to spread on more than 1 switch
  - IB islands
- It's the same for BeeGFS
- The overall IB network blocking factor 1:2
  - The « inter-switch » IB network is only used for GPFS

Cascade effect

- Should we really keep all these useless IB links ?
  - We can probably lower the blocking factor (1:3,...) ✓
  - Would lead to bigger IB islands → bigger MPI jobs ✓
- Given that GPFS is not that performant for **home directories** (lots of small files sometimes), we have to replace it by something else
- By the way, do we really need a parallel filesystem for the home directories ?
- Why not use Gb Ethernet for file access ?

Cascade effect

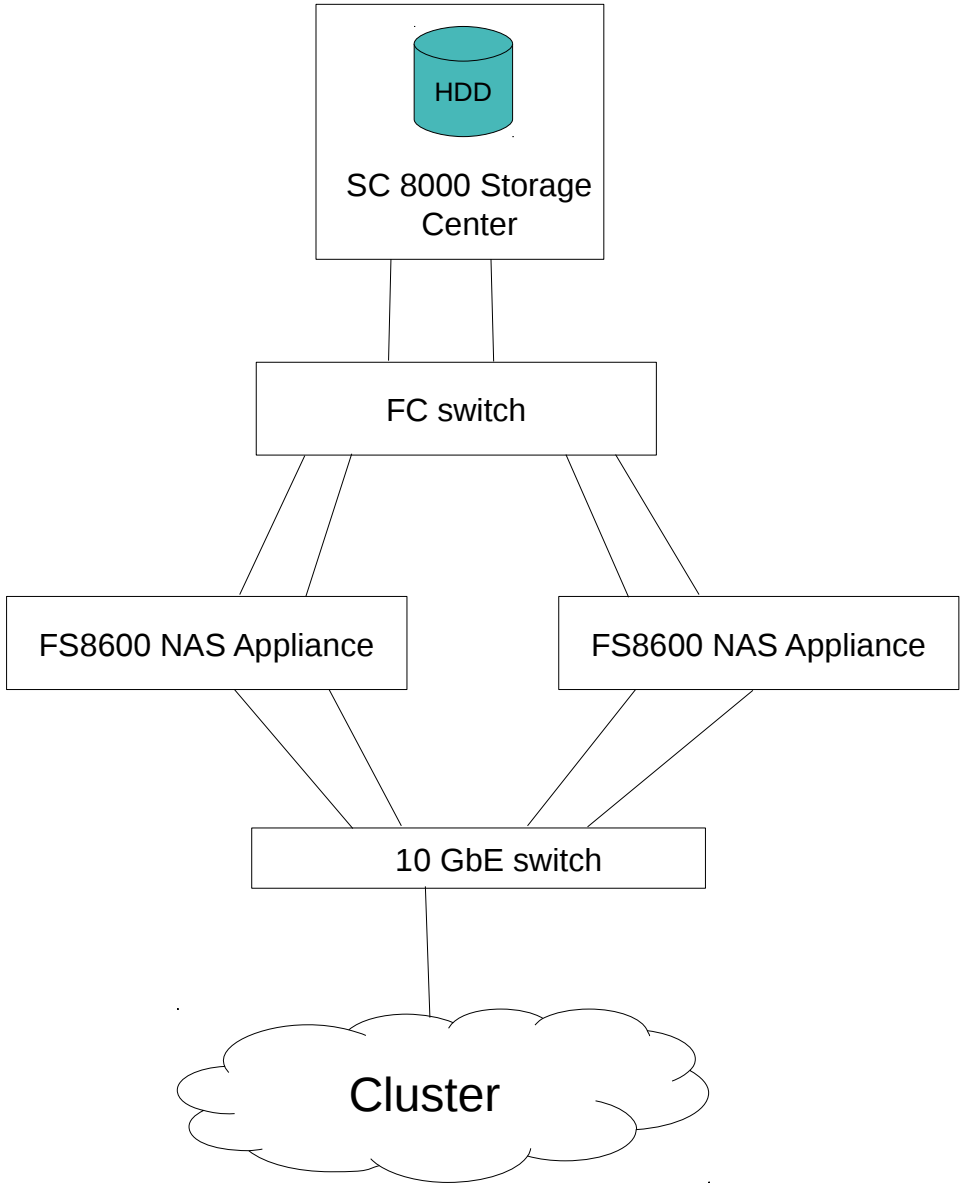
- **Statement: we want to build upon capacitive drives (7200 RPM, >= 4 TB)**
- We tried on-site several filesystems (Thanks to Dell and Rozo Systems)
- We tried :
  - Dell Compellent : pseudo-parallel NAS (up to 4 NAS heads) delivering CIFS, NFS, …
  - RozoFS: SDS, based on standard hardware, NFS and *native* mode via fuse
- We compared it with GPFS and BeeGFS using identical benchmarks

Dell Compellent / FluidFS

- Dell Compellent / FluidFS
  - Primary access on SSDs, tiering during night :

    not adapted for HPC
- We wanted to test a full SATA 7.2K configuration
- An interest : SC280 = 84 x 7.2K HDD / 5U (4Po/42U)

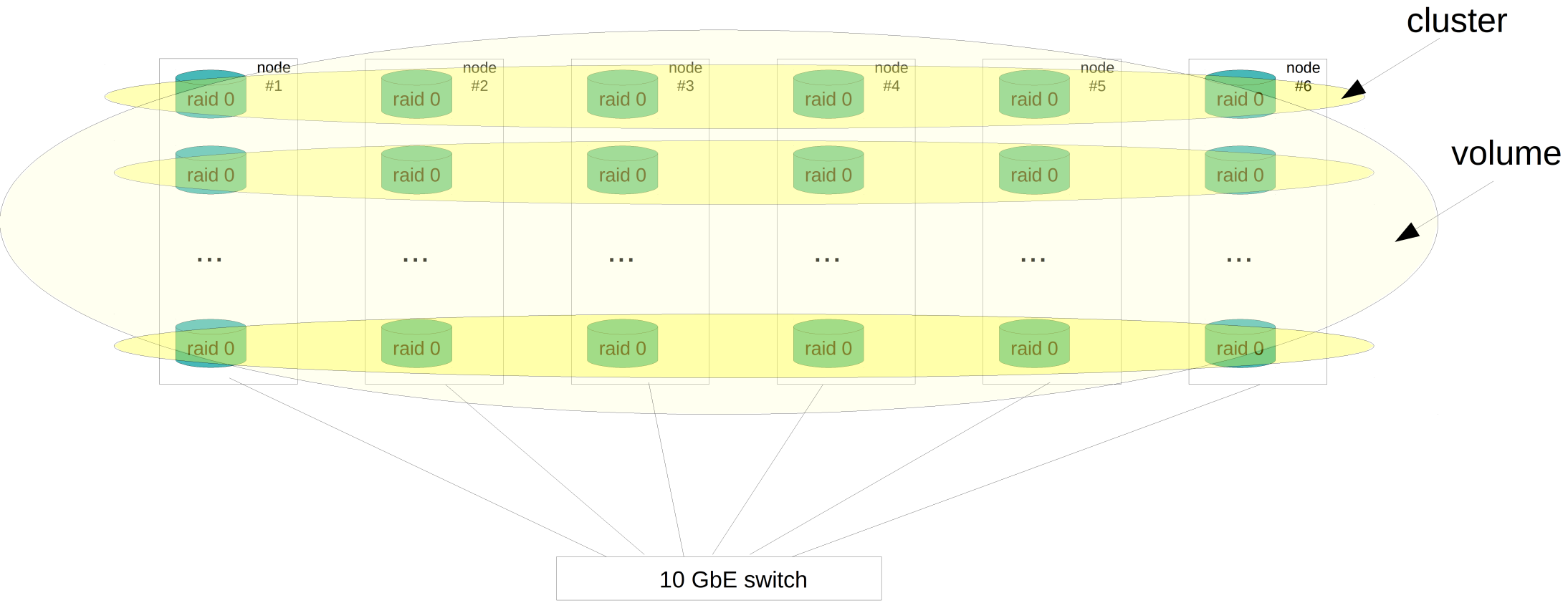
Dell Compellent / FluidFS



RozoFS

- RozoFS : un autre SDS
- Notre setup : 4 "projections"par fichier (code à effacement par transformées de Mojette), réparties sur 6 serveurs
  - Coût disque = 1,5x
  - Perte possible de 2 serveurs pour 1 fichier
    - Reconstruction sur un autre serveur en cas de perte
- Pas de RAID. Gros RAID très risqués, car les défaillances se produisent souvent en série
- En théorie, N-4 perte de serveurs possible
  - si place suffisante sur 4 serveurs
  - si les reconstructions ont le temps de se faire

RozoFS setup



Procédure de test

- On attaque avec un grand nombre de noeuds :
  - 128 noeuds sont facilement mobilisables, car non mutualisés (Equipex)
  - Jobs en cours suspendus
  - noeuds 1 GbE, mais saturation des liens vers les POC au bout d'un nombre suffisant de noeuds
- E/S brutes en volume : dd
- E/S aléatoires : fio (couteau suisse)
- Différentes mesures ("ls -lR" = point de souffrance des utilisateurs)
- Benchs figés au bout d'un moment, pour faciliter la comparaison

Procédure de test

Exemple : fio exécuté sur un cluster de 16 noeuds BeeGFS

```
DEST=scratch-beegfs # dellfs|workdir|scratch-beegfs|rozofs : /$DEST/fio/ doit exister
 ND=101
 SCRIPT=/$DEST/fio/fio.script; SIZE=4176 # 4176 Mo cumulés par noeud (R/W)
 for N in 1 2 3 4 5 6 7 8 9 10 16; do
    ((NF=$ND+$N-1));
    ((SIZET=$SIZE*$N))
    date +"%s">date.out;
    xdsh hpc-n[$ND-$NF] -f $N "mkdir /$DEST/fio/\$(hostname); cd /$DEST/fio/\$(hostname);
   fio $SCRIPT; mv /$DEST/fio/\$(hostname) /$DEST/fio/\$(hostname).`date +'%Y%m%d-%H%M
   %S'`";
    # Mettre à jour le commentaire dans le printf :
    cat date.out | awk '{t='`date +"%s"`'-$1; printf "%s;8;9000;%s;%s;%s;%s;%s;# 16
   storage/client (409-424) sur edge11\n" , "'$DEST'" , '$N' ,  t , '$SIZE' ,'$SIZET'/t ,
   "'$VARIANTE'" }' >> fio.csv;
 done
```

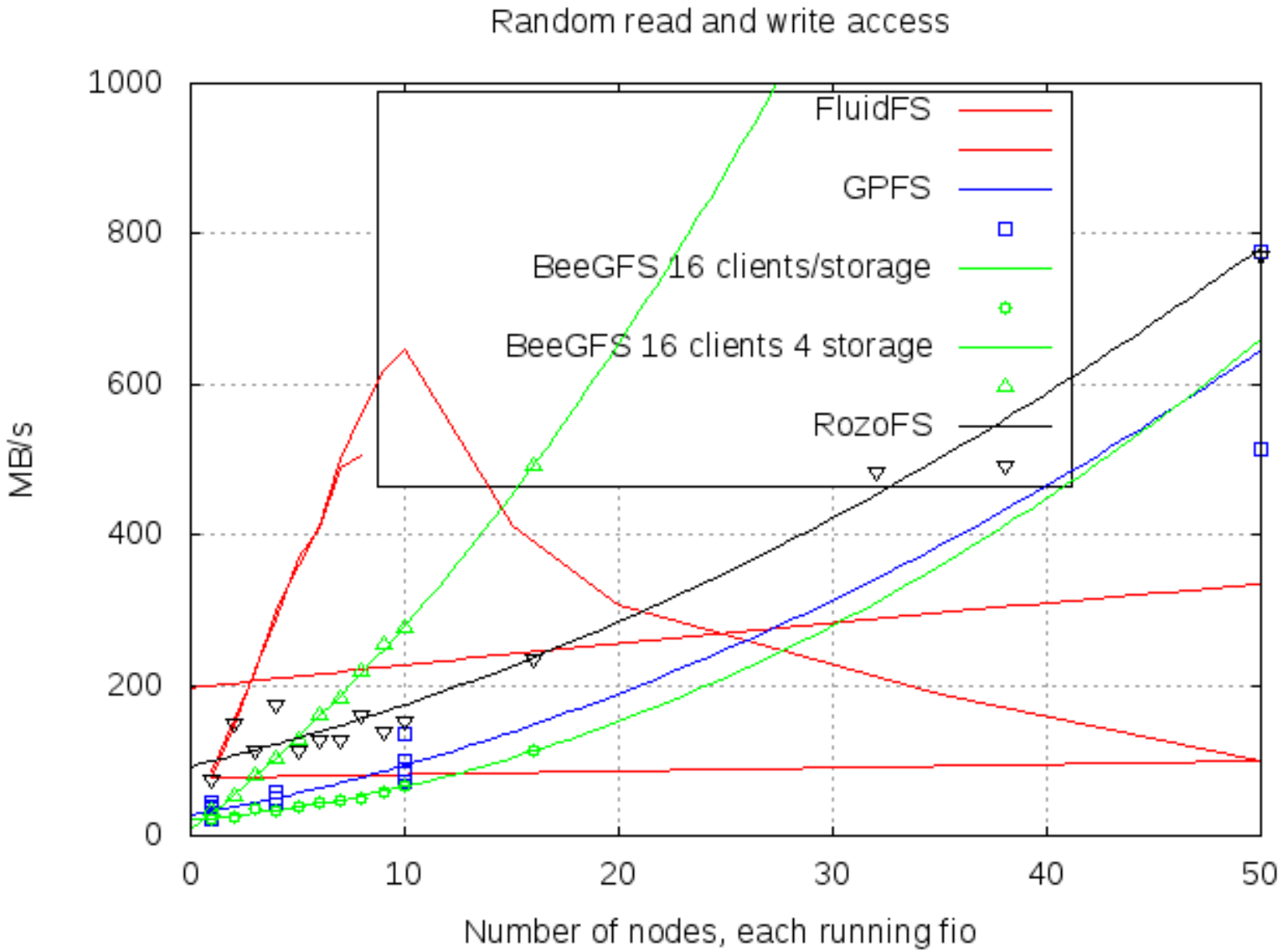
Procédure de test

- Script fio

```
[global]
 name=bench
 direct=1
 [job1]
 rw=randread
 size=10m
 bs=4k
 [job2]
 rw=randwrite
 size=10m
 bs=4k
 [job3]
 rw=randread
 size=10m
 ...
```

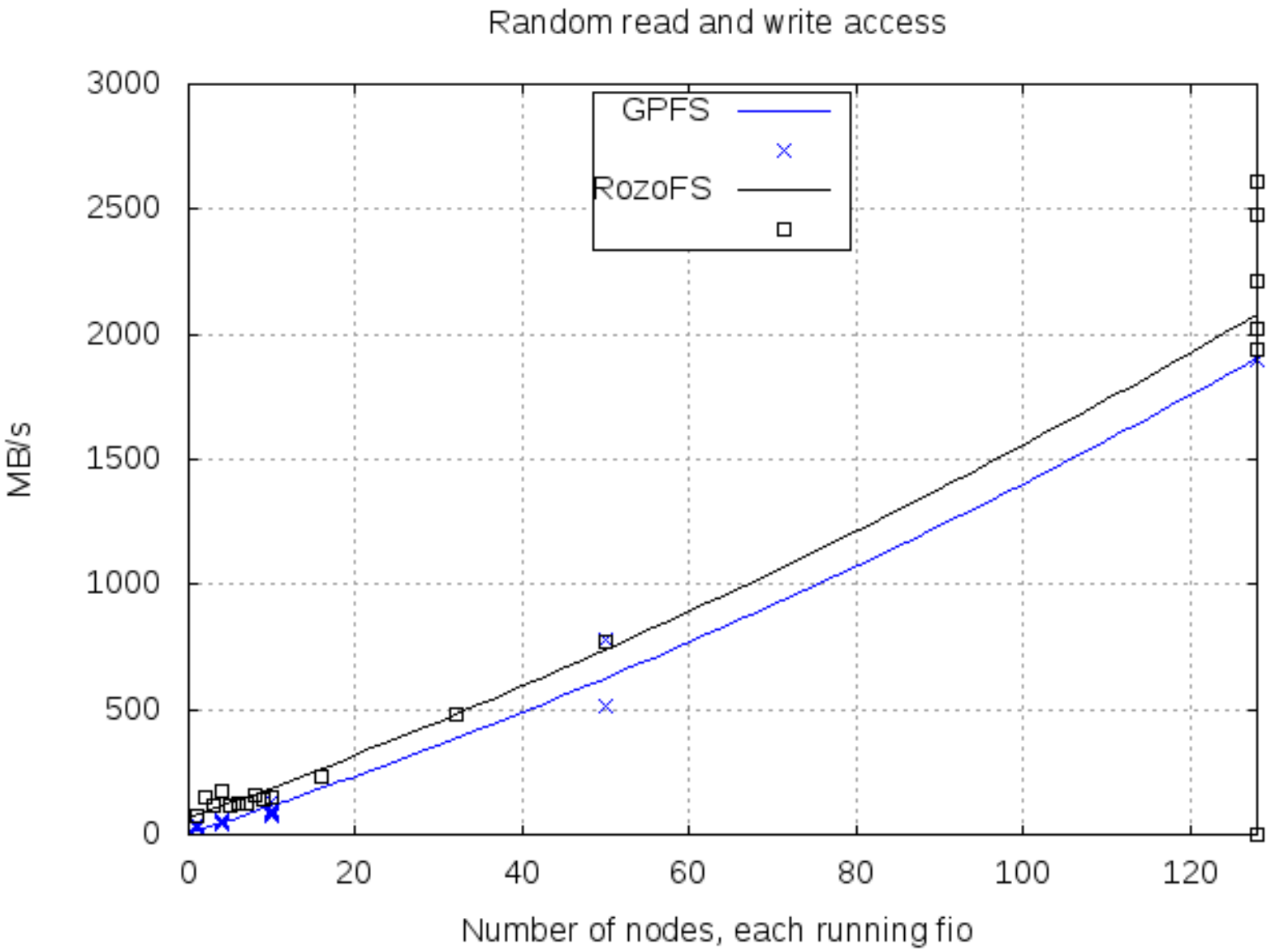
Results

- 1 to 50 nodes

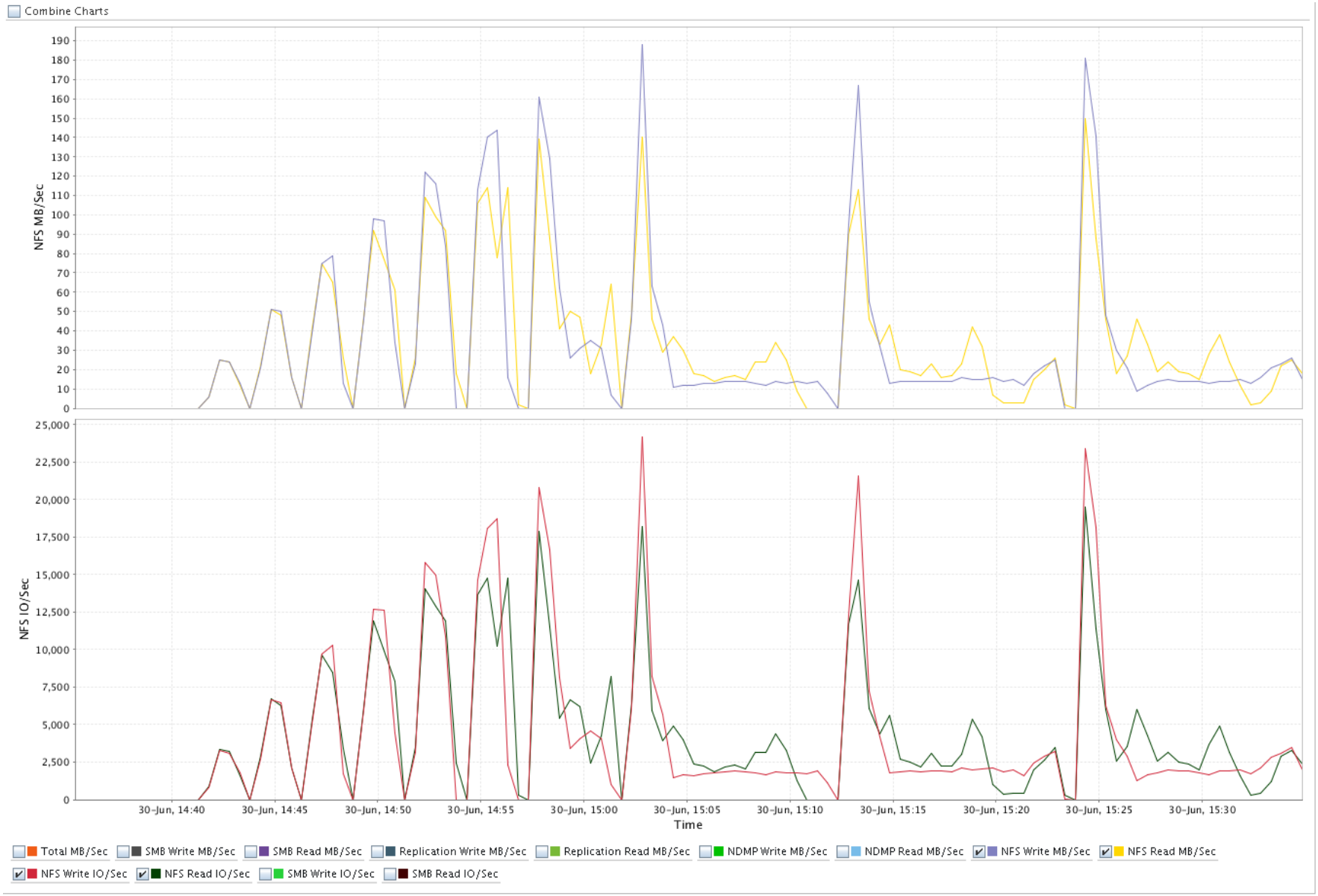


Results

- 1 to 128 nodes

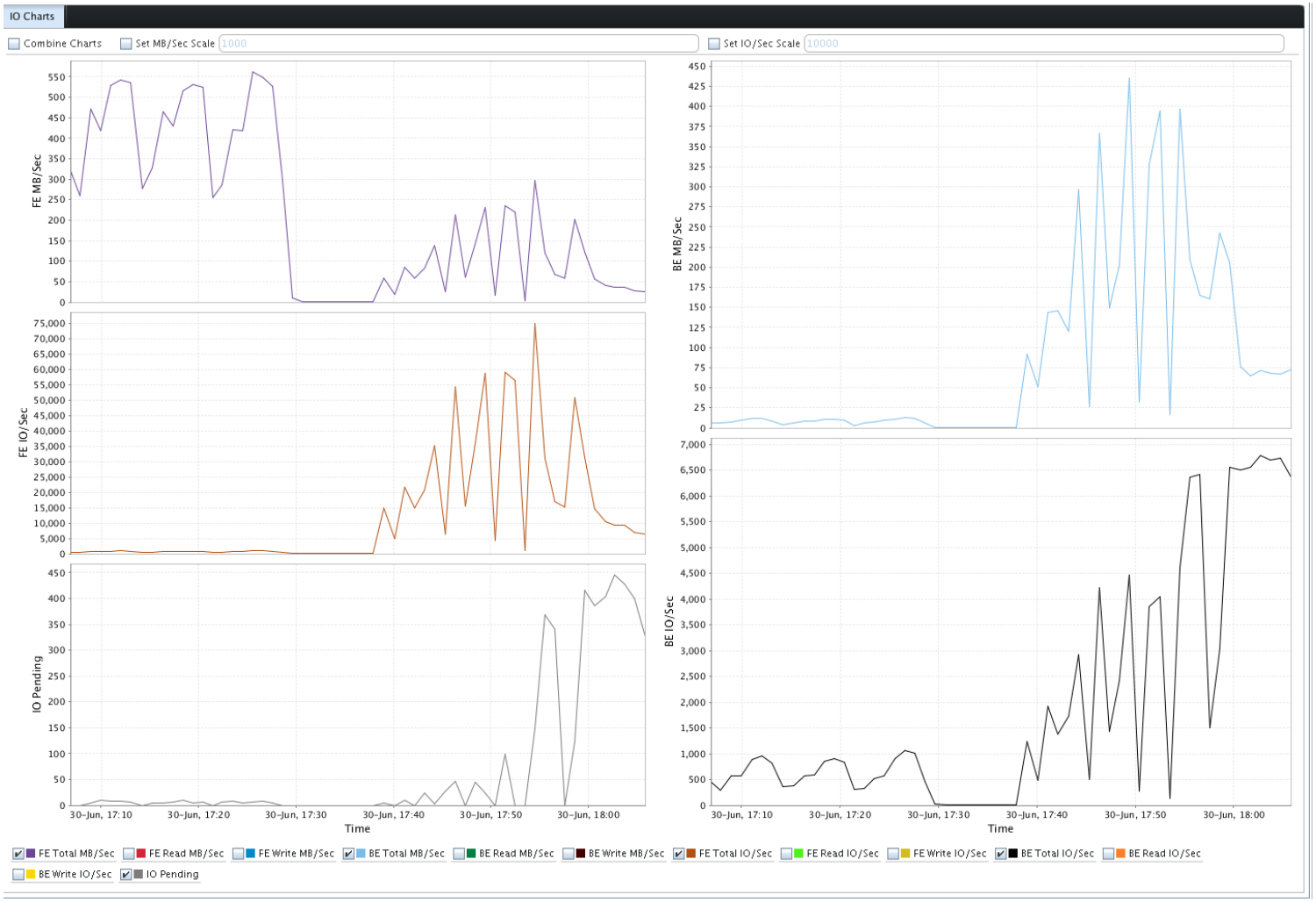


FluiFS / Compellent case



FluiFS / Compellent case

- Théorie : 80 IPOS / SATA 7200t (1840 avec 23 disques). A monitorer sur le back end du contrôleur (liaisons disques). BE sur graphiques.
- Résultat : le nombre d'IOPS en attente (Pending) augmente brusquement à partir de 7 noeuds, quand on dépasse les 4000 IOPS. Ce seuil correspond à 173 IOPS par disque.
- A partir de là, les perfs s'écroulent de 200 Mo/s à 75 Mo/s.



FluiFS / Compellent case

- Grande difficulté à diagnostiquer l'origine de la baisse des performances
  - Il a fallu trouver LE spécialiste chez Dell
- Matériels spécialisés, interconnectés par des switchs, travaillant chacun dans son coin
  - A l'opposé du concept de SDS

Result

- Dell Compellent: average good results, but scalability probably limited (size and performance). Not SDS....
- RozoFS: very good performance in NFS mode. Native mode led to data loss
- We choosed RozoFS:
  - Standard hardware
  - 9 I/O servers (Dell R730xd)
  - 10 GbE network backbone
  - 576 TO at the moment, up to 1.7 PO with 6TB disks

Plan

- HPC in Strasbourg University
- « My simulation is slow »
- Cascade effect
- Conclusion

Conclusion

- A single application infulenced the whole system
  - Regional computing centers are adaptive !
- *scratch* filesystems are the perfect sandbox
- Data needs to be close to the compute... during the compute !
- SDS, SSD and 7200 RPM disks are the keys to scale-up (capacity) and scale-out (performance) storage, at reasonable prices
- 10GbE and 40GbE is a game changer... makes parallel I/O possible on Ethernet
- We now use 2 SDS systems :

  BeeGFS and RozoFS