ANF UST4HPC 2023

Stockage distribue

Beegfs

21-06-2023 JN BOUVIER

jean-noel.bouvier@univ-grenoble-alpes.fr

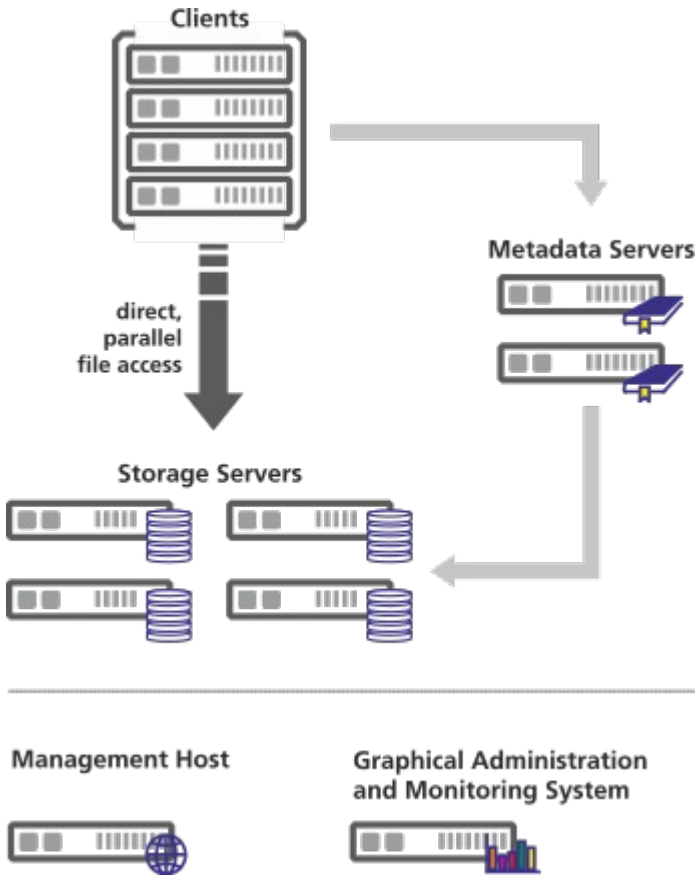
besoins et contraintes

- **HPC : stockage volumineux**
  - augmentation des volumes de donnees de recherche
  - mutualisation des moyens de stockage
- **HPC : stockage performant**
  - accès aux données : nombreux, longs, intensifs
  - NFS : facile à mettre en oeuvre mais peu performant
- ***évolutivité***
  - infrastructure hétérogène : financements multiples au fil de l’eau
- **facilité d'administration**
  - best effort

beegfs

- **c'est quoi beegfs ?**
  - www.beegfs.io
  - ex **Fraunhofer** FS
  - développé et optimisé pour le **HPC**
  - utilisé par de nombreux centres du **TOP500** computers
  - FS **distribué** : **metadata** et **data**
  - TCP/IP et/ou RDMA (InfiniBand, Omni-Path, RoCE)
  - **Linux**
  - **pas** de spécificités matérielles
  - différents FS sous-jacents supportés : EXT4, XFS, ZFS
  - quotas (tracking / enforcement)
  - data striping

beegfs

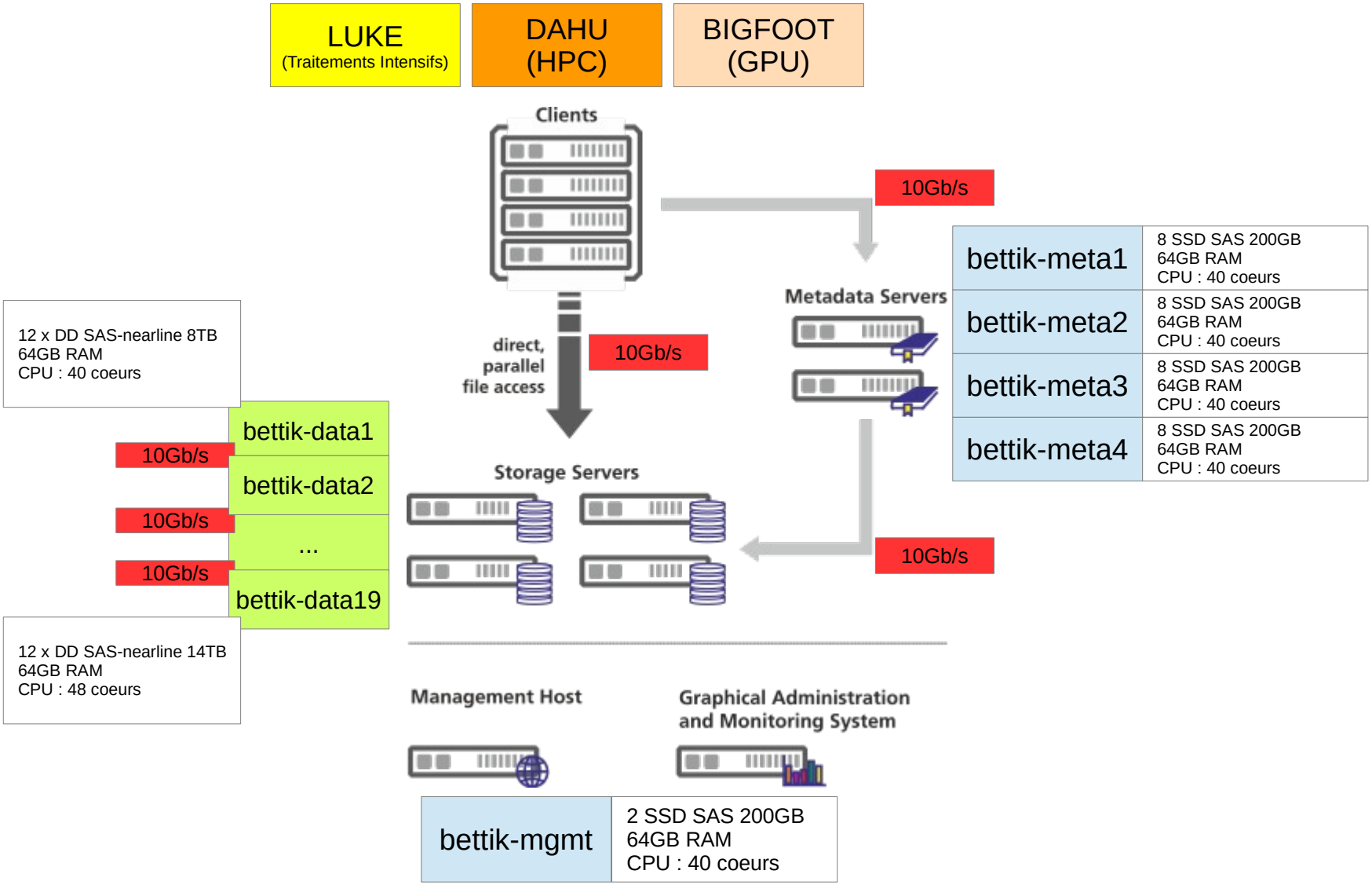

Clients
direct,
parallel
file access
Metadata Servers
Storage Servers
Management Host
Graphical Administration
and Monitoring System

beegfs @ CIMENT UGA

- **historique du projet**
  - équipe de ISTerre, 4 noeuds, 20To
  - performances espérées : 1GB/s par noeud de stockage
  - raccordement au réseau 10Gb/s

- **aujourd'hui**
  - 4 metadata servers | 16 instances (~8kE)
  - 20 storage servers (~8kE)
  - Infra globale ~ 200kE
  - 10Gb/s ethernet TCP/IP
  - 1.7Po | nombreux utilisateurs | 3 clusters | 330 millions de fichiers
  - 3 personnes | 10%

beegfs @ CIMENT UGA



beegfs @ CIMENT UGA

- **Installation|configuration|optimisation**
  - https://doc.beegfs.io/
  - support et packages : RHEL 8 and 9 | SLES 15 | Debian 10 and 11 | Ubuntu 18.04, 20.04 and 22.04
  - Management|Monitoring server
    - 2 disks SSD 500Go RAID1
  - Metadata servers (x4)
    - 4 x 2 disks SSD 500Go RAID1
    - EXT4 (mkfs.ext4 **-i 1024 -I 512** /dev/sdX)
    - EXT4 (noatime,nodiratime,nobarrier)
    - Beegfs tuneNumWorkers = 128
    - demo bettik-meta1 : megasasctl|df -h|df -i|htop
  - Storage servers (x20)
    - X disks SAS 8-14To RAID6
    - XFS (mkfs.xfs -d **su=<taille-chunk> sw=<nb-disks-RAID-sauf-parity> -isize=512** /dev/sdX)
    - XFS (noatime,nodiratime,logbufs=8,logbsize=256k,largeio,inode64,swalloc,allocsize=131072k)
    - Beegfs tuneNumWorkers = <nb-disks>
    - demo bettik-data1 : megasasctl|df -h|df -i|htop

beegfs @ CIMENT UGA

- **Installation|configuration|optimisation**
  - installation 1 serveur meta avec 2 instances
  - installation 3 serveurs data
- **Mises a jour**
  - # apt-get dist-upgrade

beegfs @ CIMENT UGA

- **Commandes utiles**
  - état du cluster

    # beegfs-check-servers
  - utilisation des disks et des FileSystems

    # beegfs-df
  - Accès aux logs

    # less /var/log/beegfs-*.log
  - beegfs-ctl

    # beegfs-ctl --help

    # beegfs-ctl --listnodes --notype=management|metadata|storage|client

beegfs @ CIMENT UGA

- **Commandes utiles**
  - stats servers

    # beegfs-ctl --serverstats --perserver --interval=2
  - stats clients

    # beegfs-ctl --clientstats --nodetype=metadata --interval=2

    # beegfs-ctl --clientstats --nodetype=storage --interval=2
  - stats users

    # beegfs-ctl --userstats --names --nodetype=metadata --interval=2

    # beegfs-ctl --userstats --names --nodetype=storage –interval=2
  - interface web

    http://[monit]:8000

beegfs @ CIMENT UGA

- **Commandes utiles**
  - migration fichiers de dataX

    # beegfs-ctl --migrate --targetid=X /{repertoire}
  - tests de performance de storage

    # beegfs-ctl --storagebench --alltargets --**write** --blocksize=1M --size=2G --threads=5

    # beegfs-ctl --storagebench --alltargets --**read** --blocksize=1M -- size=2G --threads=5

    # beegfs-ctl --storagebench –alltargets --**status --**verbose

    # beegfs-ctl --storagebench --alltargets --**cleanup**

beegfs @ CIMENT UGA

- **Data striping : 10GB**
  - **10GB : 1 file**

    $ dd if=/dev/zero of=/bettik/bouvijea/10GB.img bs=256k count=40000
  - **chunk size**

    # beegfs-ctl --setpattern --chunksize=1m --numtargets=4 /bettik/bouvijea/10G-1m

    # beegfs-ctl --setpattern --chunksize=512k --numtargets=4 /bettik/bouvijea/10G-512k

    # beegfs-ctl --setpattern --chunksize=256k --numtargets=4 /bettik/bouvijea/10G-256k
  - **comparaisons**

    $ dd if=/dev/zero of=/bettik/bouvijea/10G-1m/10GB.img bs=256k count=40000

    $ dd if=/dev/zero of=/bettik/bouvijea/10G-512k/10GB.img bs=256k count=40000

    $ dd if=/dev/zero of=/bettik/bouvijea/10G-256k/10GB.img bs=256k count=40000

beegfs @ CIMENT UGA

- **Data striping : 10GB**
  - **10GB : 1000 files**
  - **chunk size**

    # beegfs-ctl --setpattern --chunksize=1m --numtargets=4 /bettik/bouvijea/10G-1m-1000

    # beegfs-ctl --setpattern --chunksize=512k --numtargets=4 /bettik/bouvijea/10G-512k-1000

    # beegfs-ctl --setpattern --chunksize=256k --numtargets=4 /bettik/bouvijea/10G-256k-1000
  - **comparaisons**

    $ time for i in `seq 1 1000`; do dd if=/dev/zero of=/bettik/bouvijea/10G-1m-1000/test$i.img bs=256k count=40 oflag=direct > /dev/null 2>&1; done

    $ time for i in `seq 1 1000`; do dd if=/dev/zero of=/bettik/bouvijea/10G-512k-1000/test$i.img bs=256k count=40 oflag=direct > /dev/null 2>&1; done

    $ time for i in `seq 1 1000`; do dd if=/dev/zero of=/bettik/bouvijea/10G-256k-1000/test$i.img bs=256k count=40 oflag=direct > /dev/null 2>&1; done

Questions ?