CENTRE
RÉGIONAL DE
RESSOURCES
INFORMATIQUES

UNIVERSITÉ
Clermont
Auvergne
& associés

Mesocentre Clermont Auvergne

Antoine MAHUL, Université Blaise Pascal
antoine.mahul@clermont-universite.fr

CRRI

Centre Régional de Ressources Informatiques

- Lieu de mutualisation de ressources informatiques
- Point de présence RENATER en Auvergne
- Opérateur du réseau métropolitain CRATERE (17 établissements)
- Gestion du réseau universitaire
- Opérateur du datacenter universitaire (200 m²)
- Appui au SI universitaire
- Appui au mésocentre

Mésocentre

- Un mésocentre est :
  - un ensemble de moyens humains
  - de ressources matérielles et logicielles
  - au services de la communautés scientifique
  - fédérateur (EPST, Universités, Industriels) en région
  - doté de sources de financement propres
- Pour fournir un environnement scientifique et technique propice au calcul haute performance
- Faisant l’objet d’évaluations scientifiques régulières
- Pour s'inscrire dans une démarche nationale

Gouvernance du mésocentre

- Une gouvernance pour permettre une utilisation raisonnée des ressources du CRRI :
  - Moyens mutualisés de calcul pour la production
  - Services autour du calcul scientifique
  - Stockage de données pour la recherche
- Deux comités :
  - Orientation
  - Utilisateurs
- Objectif premier : permettre un accès simple à des moyens plus importants

Ressources du mésocentre

- Ressources de calcul pour la Grille EGI
  - En production sur la grille depuis 2006
  - 192 coeurs de calcul, 48 To de stockage
- Cluster de calcul (HPC1)
  - En production depuis 2009
  - 120 coeurs de calcul
- Cluster de calcul (HPC2)
  - En production depuis 2015
  - > 700 coeurs de calcul, 10 To de stockage
- Stockage distribué en mode Objet
  - Production printemps 2016
  - 520 To bruts
- Architecture SMP (multicoeurs, grande quantité de RAM)
- Cloud IaaS

Cluster de calcul HPC2

- Seconde génération du cluster de calcul local
- Mise en production au **printemps 2015**
- Nœuds de calcul :

Noeuds	Qtté	CPU	RAM	HT	Cœurs	Inter	Financement
hpcnode[01-04]	4	2 x Sandy Bridge	64 Go	Non	16	GbE	CNRS / LMGE
hpcnode[05-08]	4	2 x Ivy Bridge	96 Go	Oui	32	GbE	CPER 2011
hpcnode[09-24]	16	2 x Ivy Bridge	128 Go	Oui	32	GbE	CPER 2014
hpcnode[25-28]	4	2 x Ivy Bridge	256 Go	Oui	32	GbE	CPER 2014
hpcnode[29-36]	8	2 x Ivy Bridge	128 Go	Oui	32	QDR	CPER 2014

- Installation (OS/Applications) homogène sur tous les nœuds

HPC2 : infra système

- Job manager : SLURM
  - Allocation de ressources (CPU, RAM, GPU)
  - Gestion de l'architecture des processeurs (socket / core / threads)
  - Confinement et placement des processus
  - Fair Queueing : plus on utilise, moins on est prioritaire
- Déploiement des noeuds et des applis :
  - Cobbler + Puppet
  - Modules d’environnement

Modules d'environnements

- Mécanisme pour gérer dynamiquement les variables d'environnement
  (`PATH, LD_LIBRARY_PATH, MANPATH…`)
- Permet de gérer:
  - Plusieurs versions d'une application
  - Les versions par défaut
  - Les dépendances entre applications
  - Les chaînes de compilation (compilateur / MPI…)
- Via la commande `module` (fonction shell)

Organisation des modules

Organisation hiérarchique pour maitriser la chaîne de compilation

- Modules « core » : compilateurs et outils de base

  gcc, intel, java, cmake, binutils…

- Modules « apps » : codes sans dépendances à un compilateur (codes précompilés…)

  blast, bioperl…

- Modules « compiler » : codes liés à une version de compilateur (gcc, intel, java)

  R, fftw, openblas, python, numpy, openmpi…

  => Il faut charger au préalable le module du compilateur

- Modules « mpi » : codes liés à une version de compilateur et de la bibliothèque MPI:

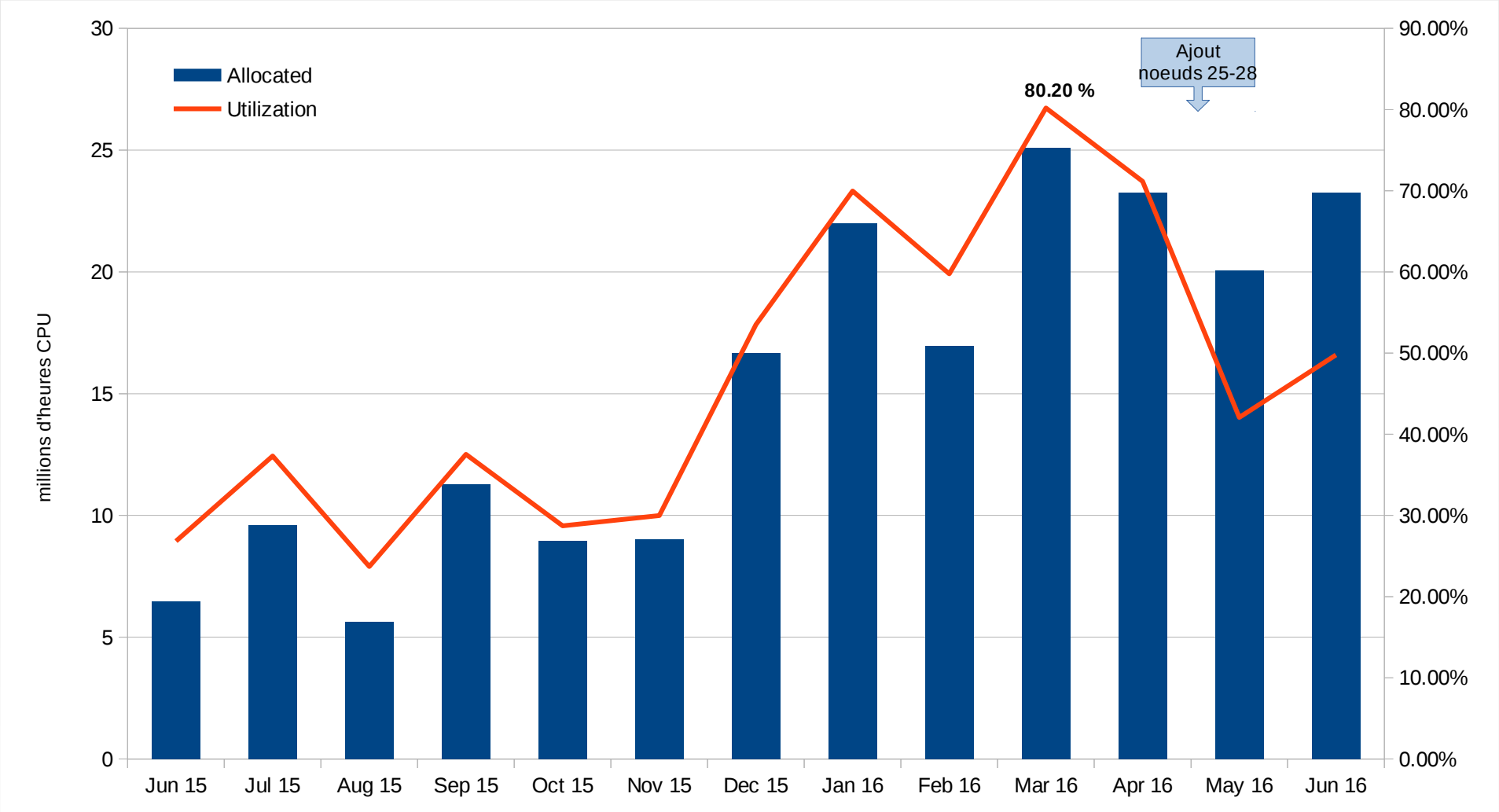
  abinit,parmetis,ray, scalapack…

  => Il faut charger au préalable les modules du compilateur et de mpi

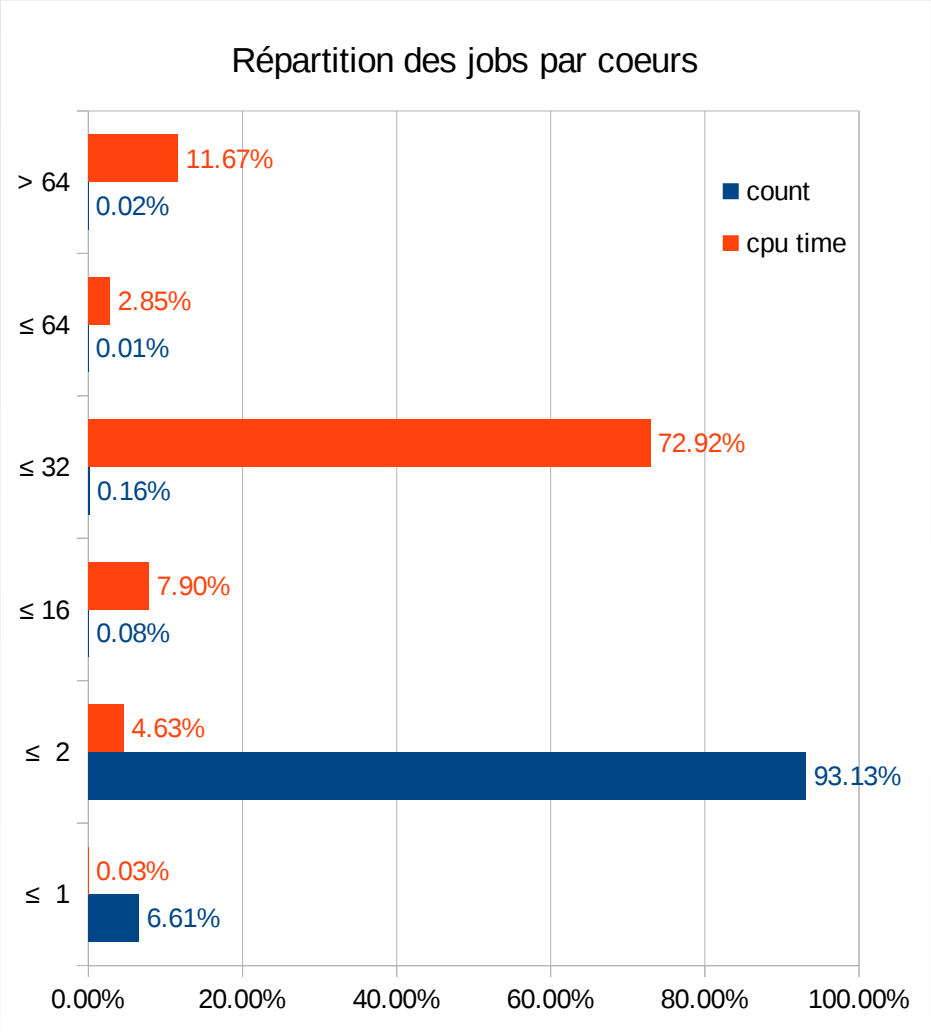
HPC2 : utilisations

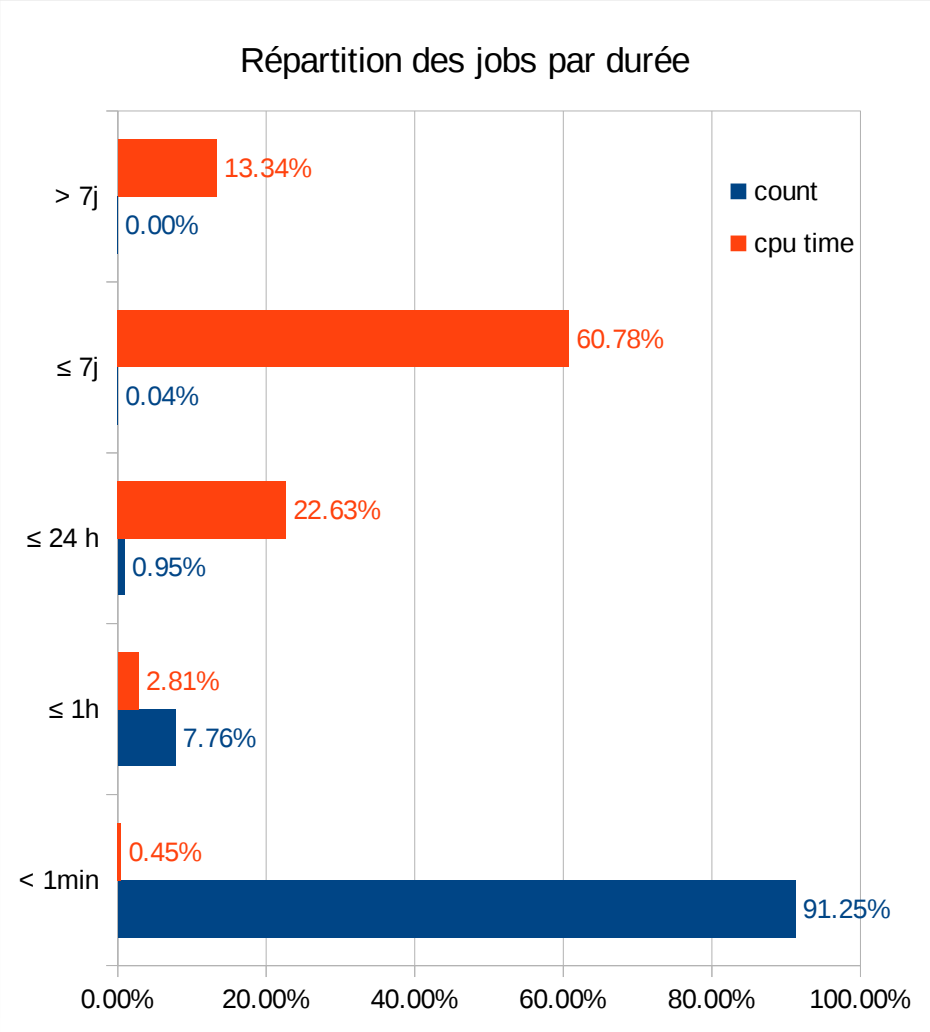
- Utilisateurs
  - 80 utilisateurs enregistrés, 19 laboratoires
  - 40 utilisateurs actifs (au moins 1 job en 2016)
- Jobs (depuis janvier 2015)
  - 2 millions de jobs soumis
  - 193 millions d'heures CPU allouées
- Applications préinstallées
  - 52 modules d'environnements
  - GCC, ICC, OpenMPI, Python, Numpy, Scipy, R, Bioconductor, Perl, Bioperl...

HPC2 : Occupation



HPC2 : Répartition des jobs





Nature des jobs

- Codes parallèles avec MPI
  - Sciences des matérieux (Abinit, CASINO)
  - Quelques codes bioinformatique (Ray, PhyML...)
- Codes multithreads
  - Chimie numérique (Gaussian, codes python...)
  - Workflows bioinformatique
  - Geant4/Gate (Physique des particules)
  - Matlab (Sciences de la Terre, Neurosciences)

Stockage objet CEPH

- CPER 2014 :

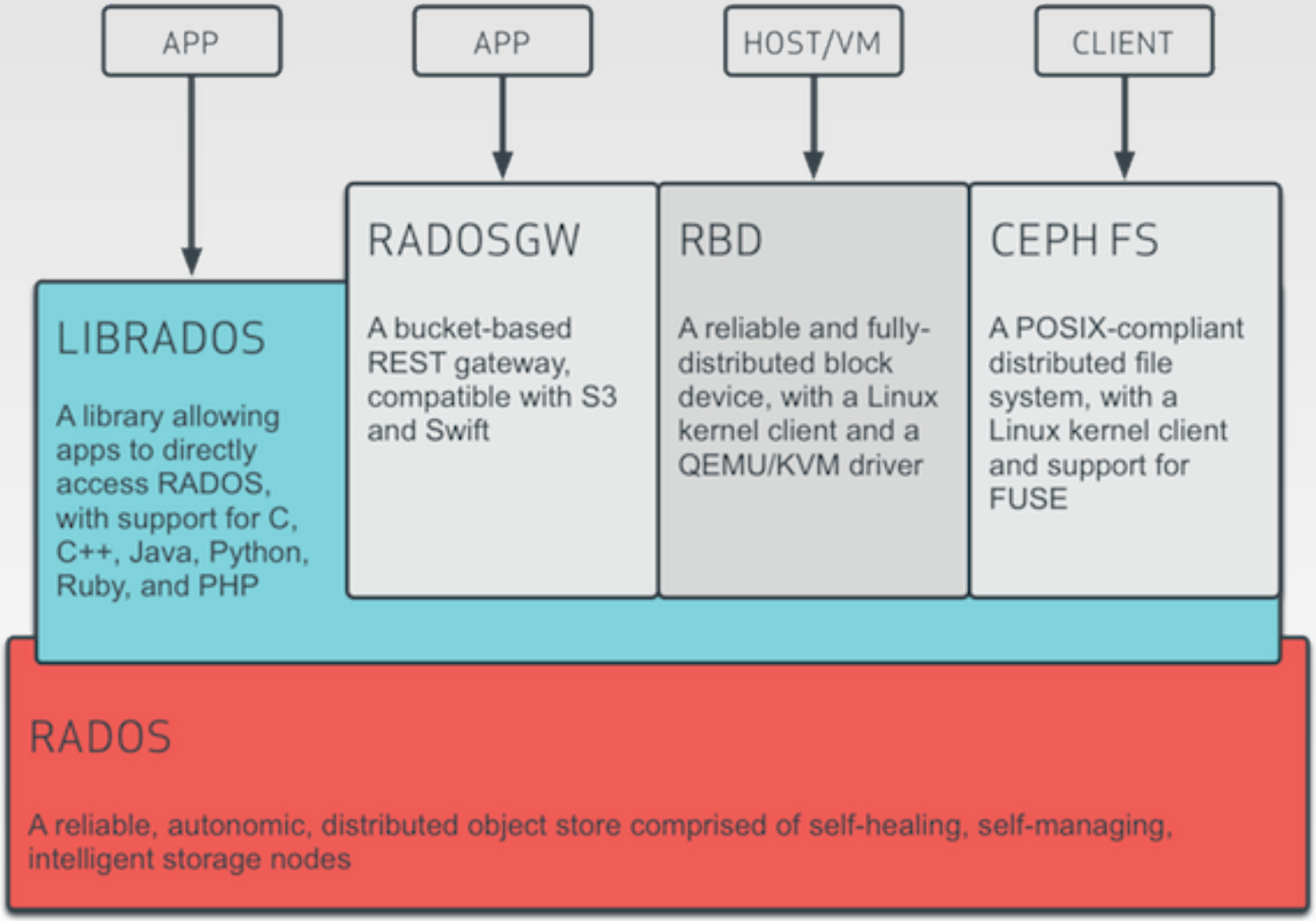
  “acquisition d’outils mutualisés pour le stockage, l’analyse de grandes masses de données, la modélisation et le calcul intensif”

- Projet de stockage capacitif
  - Stockage de grands volumes de données (>100 To)
  - Allocation d'espace de stockage à la demande
  - Extensibilité à moindre coût
  - Maîtrise des frais de fonctionnements
- Première brique d'un cloud IaaS

CEPH

- Plateforme de stockage distribué
  - Stockage en mode objet
  - Architecture *Scale-out* adaptée au passage à l'échelle
  - S'appuie sur du matériel ordinaire (*commodity hardware*)
  - Sans SPOF
- Logiciel Libre
  - Technologie issue de l'université de Californie (Santa Cruz)
  - Thèse de Sage A. Weil (2007)
  - Développement et support commercial assuré par Inktank
  - Inktank racheté par Redhat le 30 avril 2014

CEPH : principes



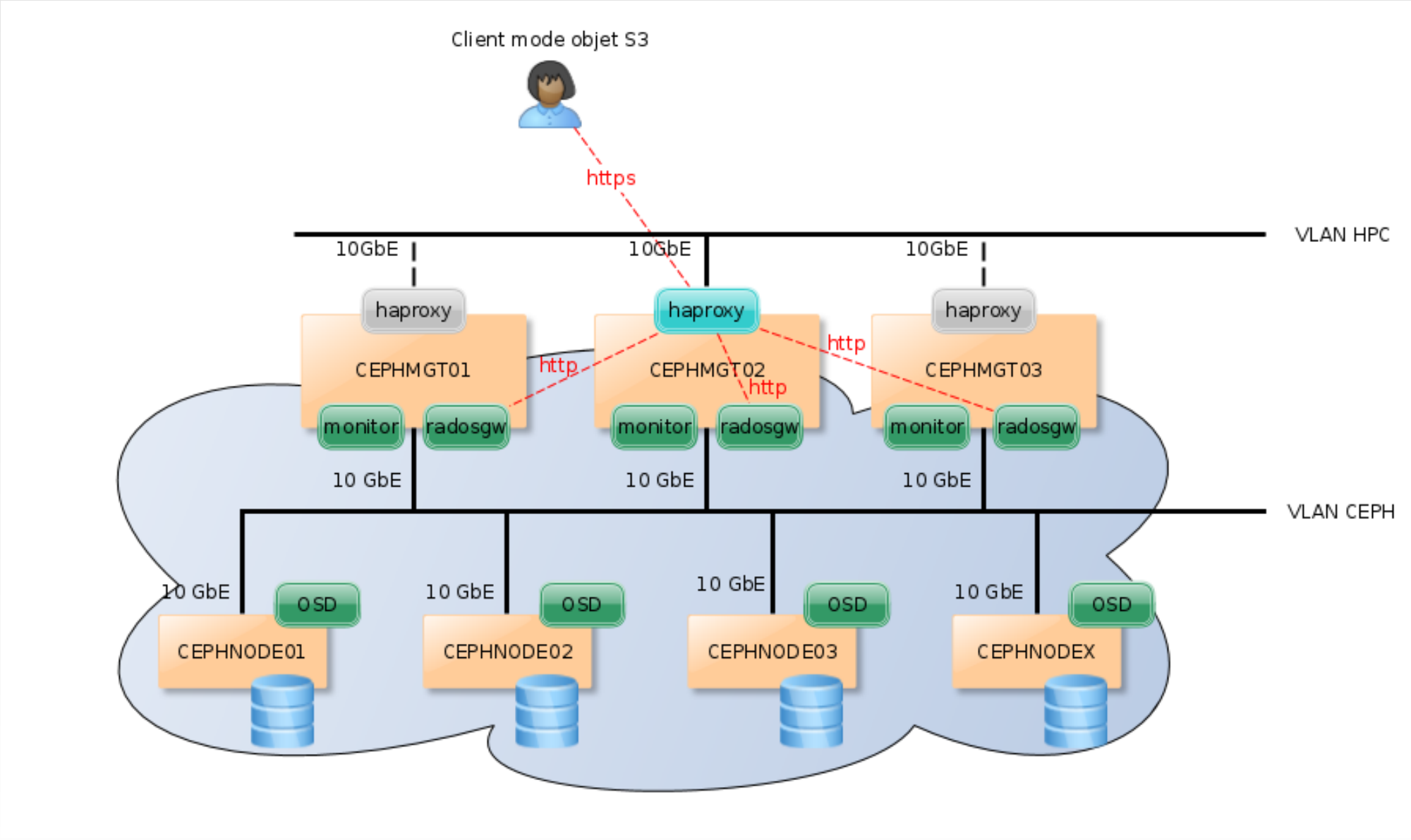
CEPH : fonctionalités

- Cluster de stockage RADOS
  - Réplication des données chaudes (surcout : x2, x3...)
  - Correction d'erreur pour les données froides (x1.5)
  - Equilibrage automatique
  - Algo de placement (CRUSH) assez évolué et modulable
  - Intégration possible avec Openstack
- Client Ceph : interface vers les utilisateurs
  - RADOSGW : API compatible Swift et S3
  - RBD : mode bloc (intégré au kernel Linux)
  - CephFS : mode fichier (stable depuis peu)

CEPH : infrastructure

- Infrastructure (CPER 2014)
  - 13 nœuds de 40 To (6 actuellement en production)
  - 3 nœuds « contrôleurs »
  - Interconnection 10GbE
- API S3
  - RadosGW déployé sur les 3 contrôleurs
  - Load balancing (avec haproxy)
  - Haute disponibilité (avec keepalived / VRRP)

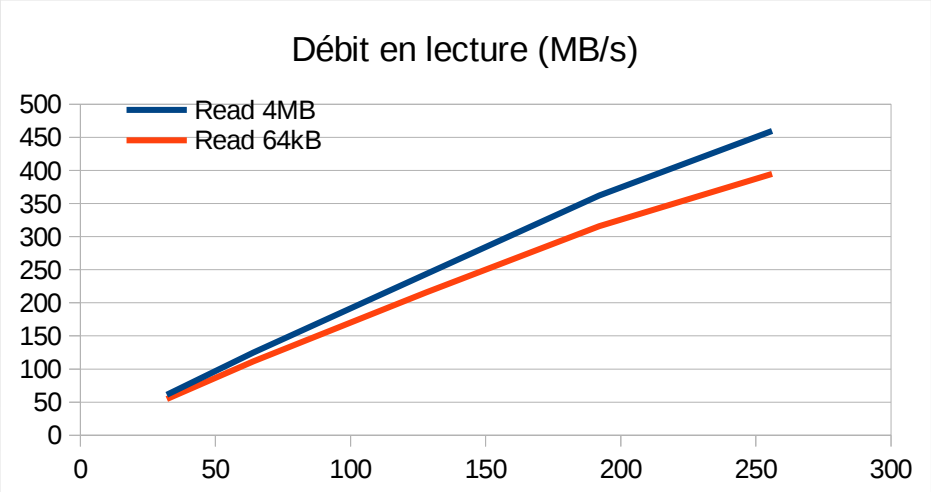
CEPH : architecture

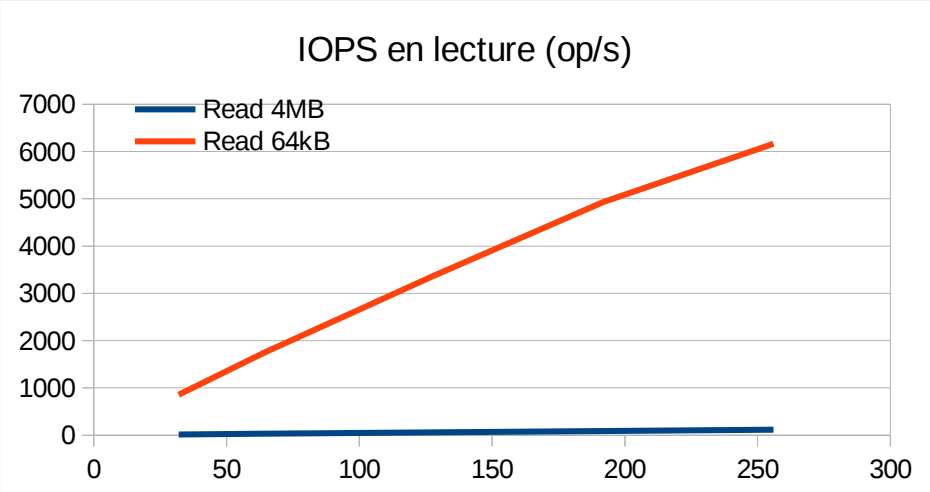


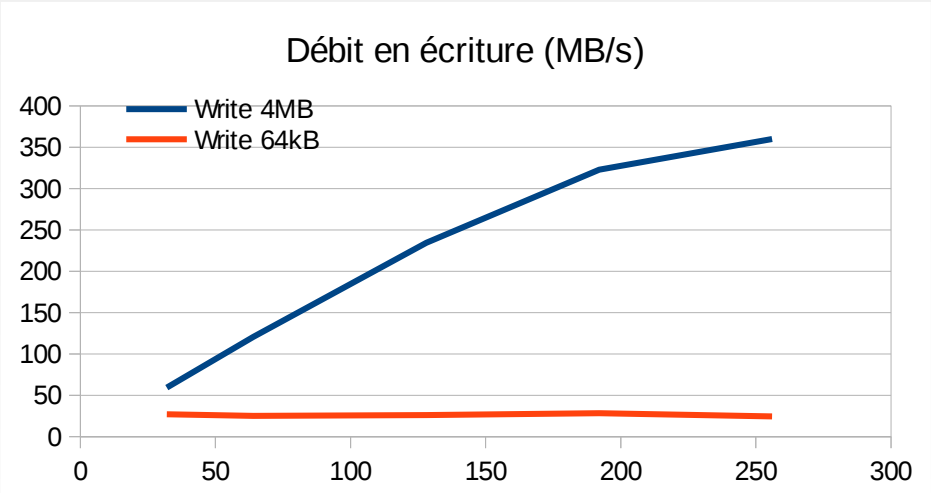
Benchmarks S3 (1)

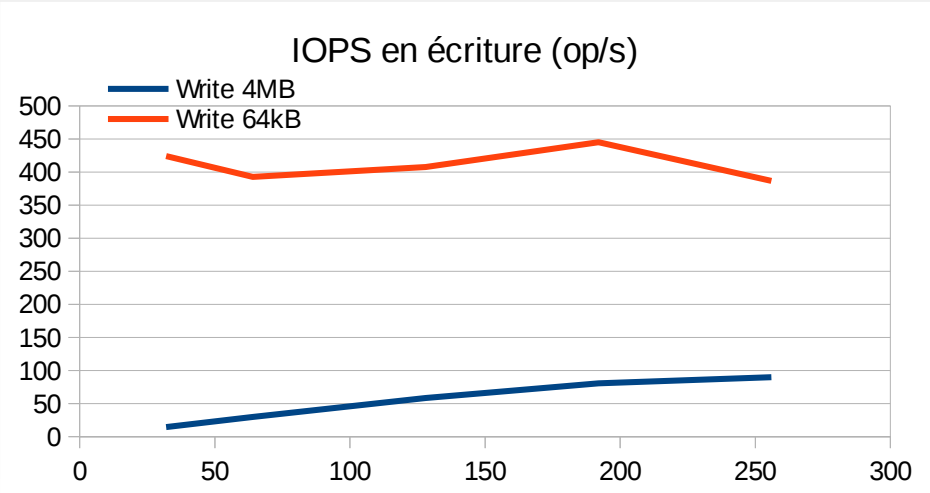
- Logiciel libre Intel COSBench
- Déploiements des clients sur le cluster HPC2 (32 clients par nœud, 1GbE)
- Scénario :
  - Ecriture : chaque client écrit des objets aléatoirement pendant 300s
  - Lecture : chaque client lit des objets aléatoirement pendant 300s
  - Tests avec des objets de 4MB et de 64kB
  - Tests de 32 à 256 clients (sur 1 à 8 nœuds)

Benchmarks S3 (2)









Défi Audace

- Projet CPER (2015-2020) autour du traitement de grands volumes de données
- Projet transverse aux projets CPER thématiques
- Equipements structurants
  - Plateforme Galatica au LIMOS
  - Plateforme à mémoire partagée pour le mésocentre
    ~ 320 coeurs physiques, 8To de RAM
  - Plateforme Openstack de production
  - Equipements datacenter et réseaux
- Financements de thèses / post-docs

Conclusion

- Mésocentre Clermont Auvergne
  - Ressources mutualisées par les labos, pour les labos
  - Dimension locale & visibilité nationale
  - Aspects financiers
    - Investissement labos ou projets CPER
    - Fonctionnement par les tutelles
- A l'échelle Auvergne-Rhône-Alpes :
  - 1 centre national, 4 mésocentres...
  - Coordination des infrastructures numériques de recherche ?
  - Recherche de synergies et de complémentarités ?
  - Fédération de ressources ?