Une infrastructure mutualisée et recyclée avec Ceph

Bruno Buisson & Philippe Saby
15 novembre 2017

Contexte scientifique et technique

L'Observatoire Midi-Pyrénées

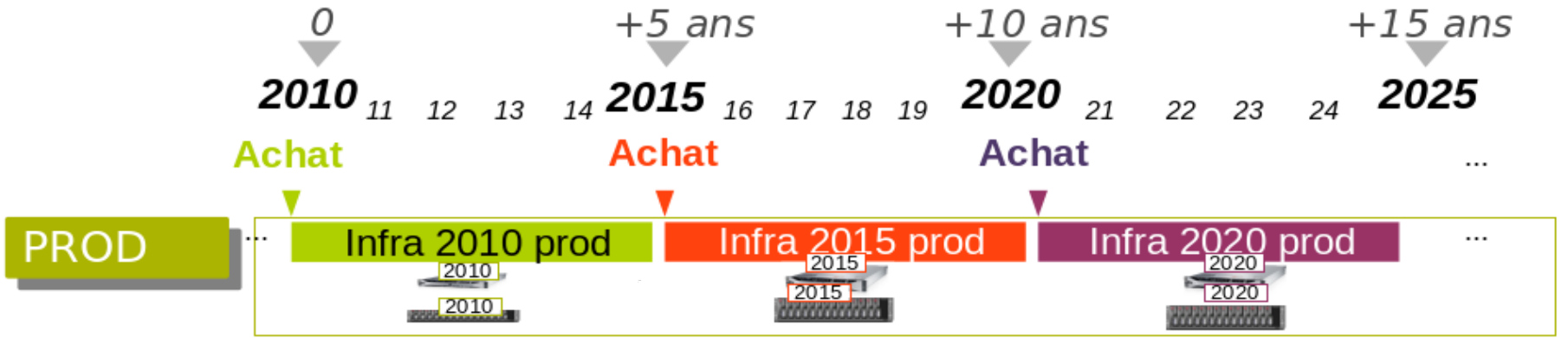
- Observatoire des Sciences de l'Univers (OSU CNRS/INSU, 6 UMR & 1 UMS)
- La donnée scientifique
  - Observations (In situ, satellites, télescopes, etc.) et modèles numériques
  - Volume de ~**2Po** - disques internes et baies de stockage DAS / NAS / SAN
  - **Sécuriser et pérenniser** (stockage secondaire, ~400To)
- ***Software Defined Storage*** (SDS)
  - Le « silo » de stockage à ± haute valeur ajoutée devient une infrastructure globale, pilotée par le logiciel
  - Vmware VSAN, Openstack swift, Scality, DataCore, Lustre, GPFS, Ceph, etc.

La valeur ajoutée n'est plus dans le matériel,
mais dans l'infrastructure globale interopérable

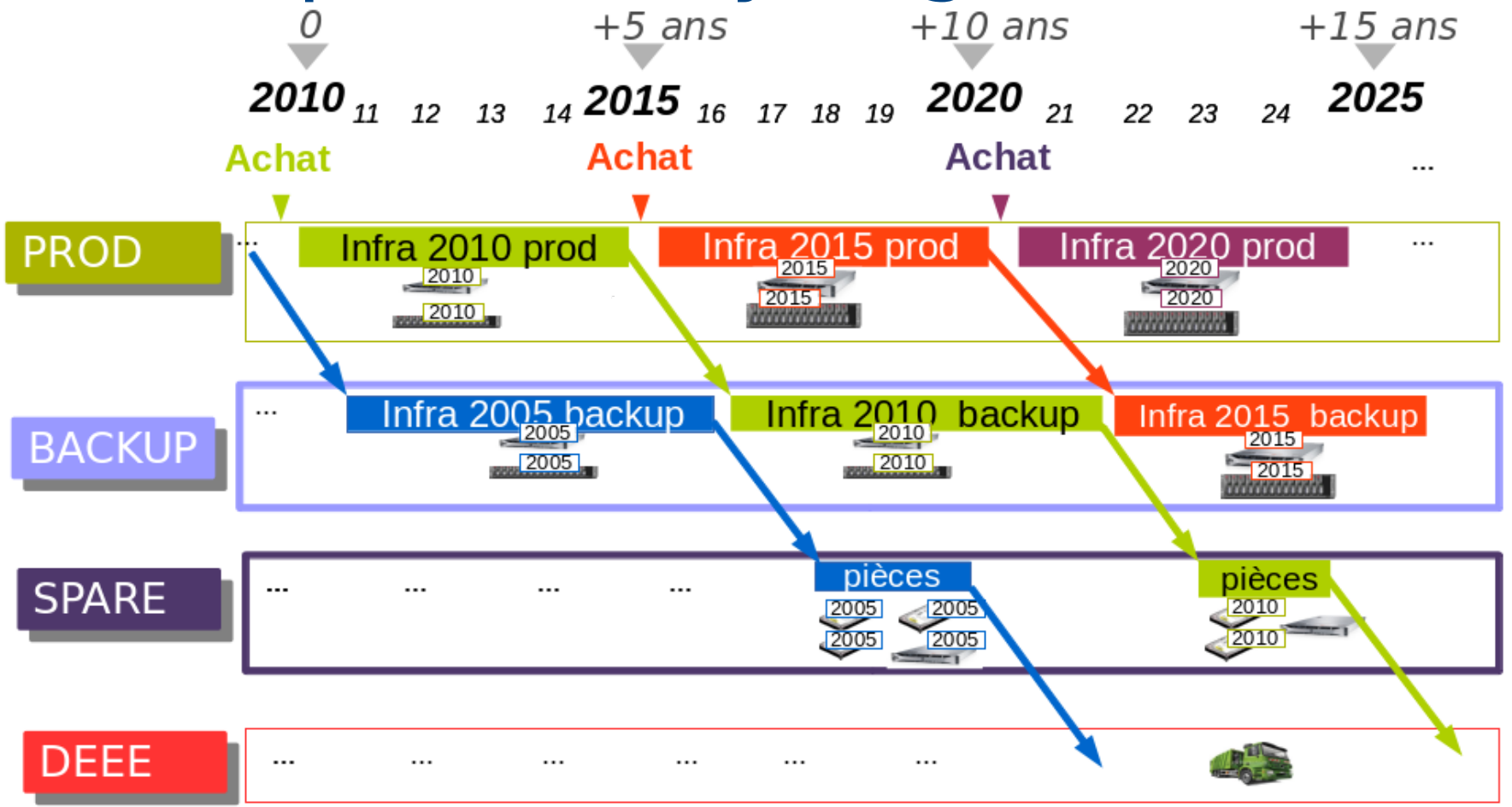
Le projet d'archivage

- Utiliser du matériel recyclé et potentiellement obsolète
  - Serveurs
  - Stockage (disques internes et baies)
- Mutualiser les infrastructures
- Répartir géographiquement les données
  - Trois d*atacenters* sur deux sites géographiques éloignés et indépendants
- CEPH pour le « *Software Defined Storage* »
  - Sécurité des données assurée par la résilience du cluster Ceph
  - Indépendance totale vis-à-vis du matériel
  - Volume quasi illimité (stockage objet)

Concept : le recyclage



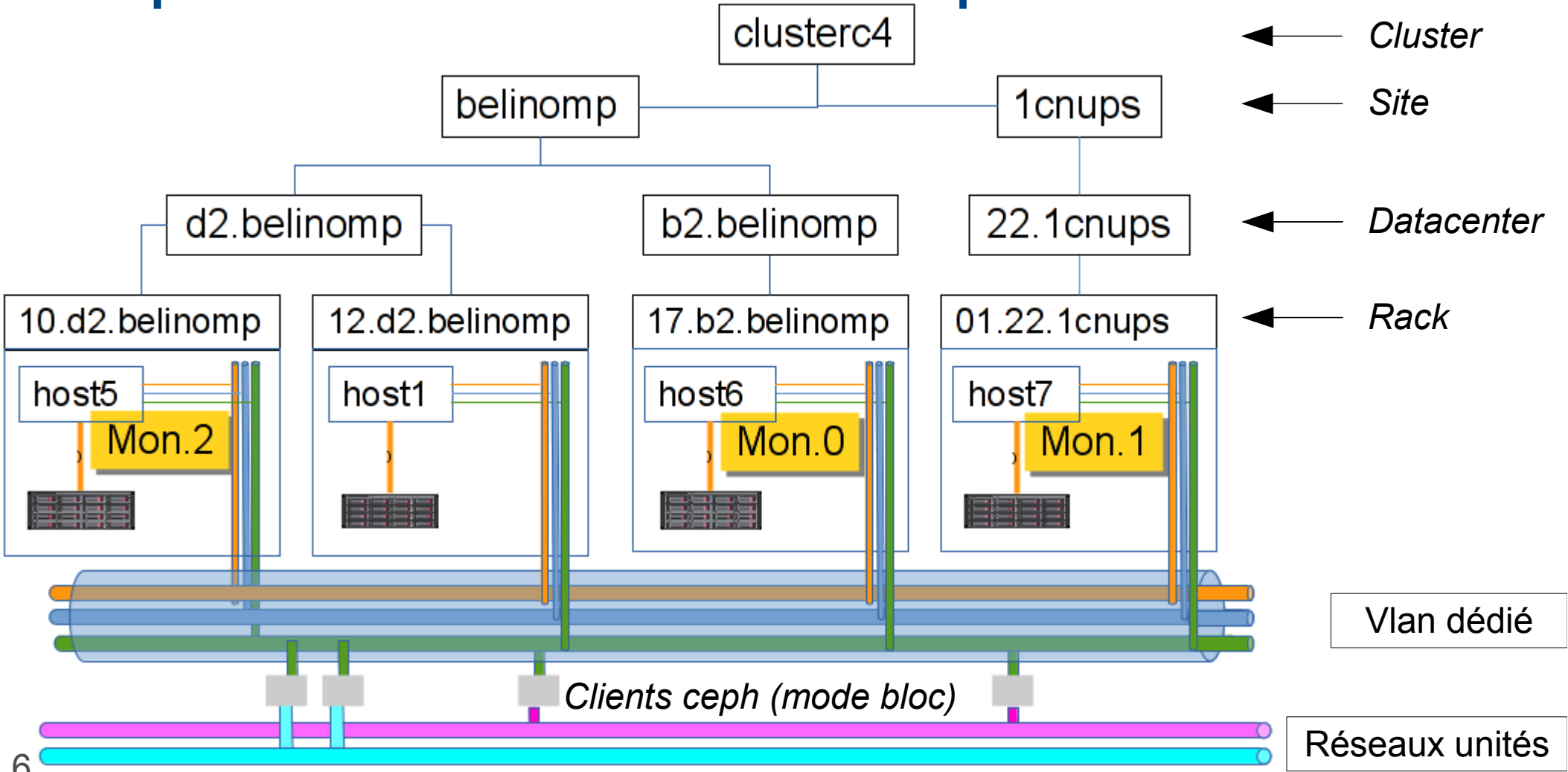
Concept : le recyclage



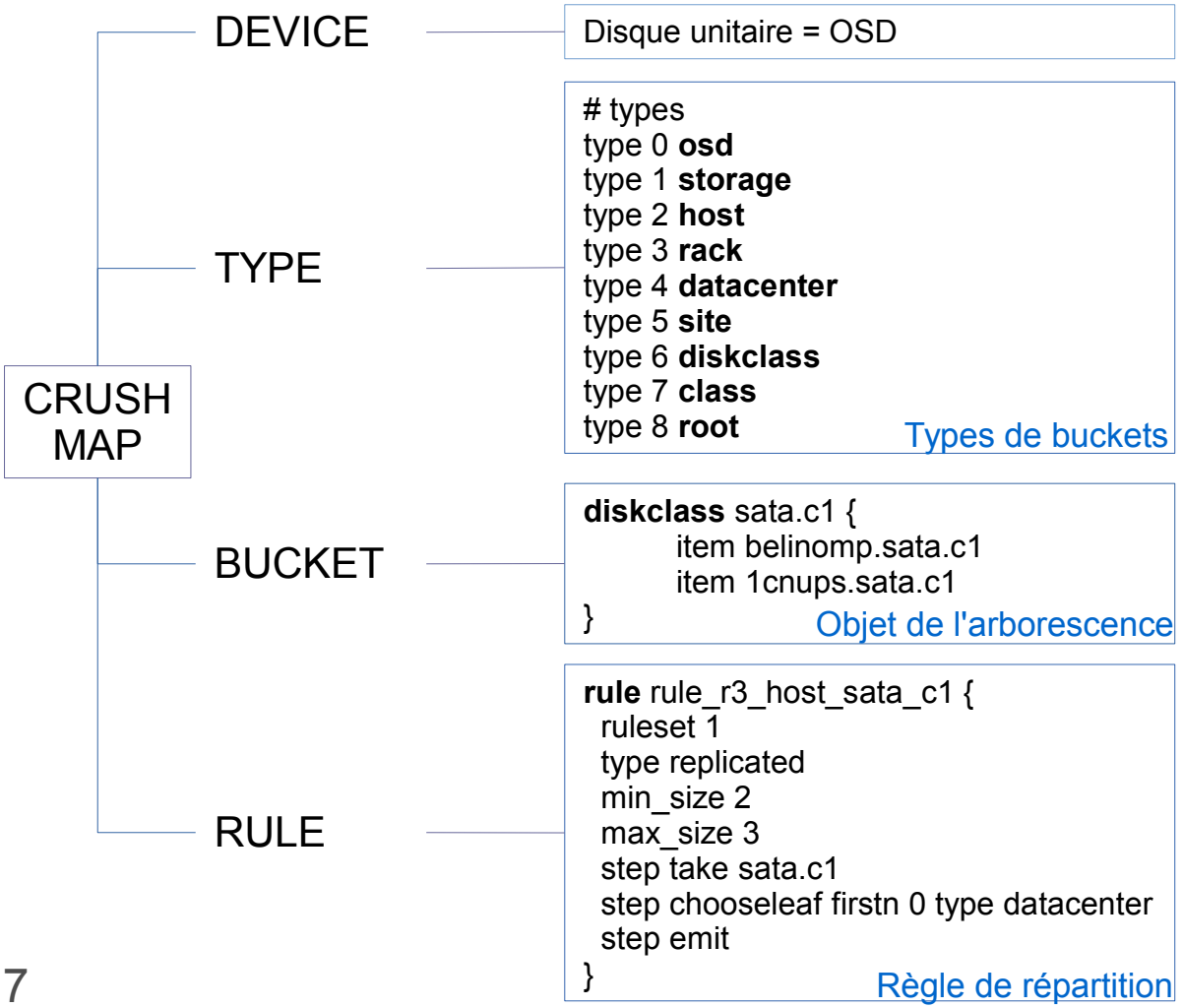
Ceph : Proof of Concept

- Serveurs sous Debian Jessie (actuellement 7 nœuds Dell 2950 → R710)
- Stockage avec disques (SCSI, SATA, NL-SAS, SAS) internes, baies DAS
- Répartition géographique sur trois *datacenters*
- Ceph version stable (Kraken)
- Vlan dédié (flux internes au cluster, administration)
- Objectifs principaux
  - Valider le déploiement de Ceph sur les matériels
  - Assurer la répartition géographique des données (*crushmap* Ceph)
  - Fournir à chaque unité participante des points d'accès au cluster Ceph, à travers différents types d'interface (LUN, machines passerelles, protocole « *cloud* », etc.)
  - Éviter la centralisation et faire en sorte que chaque unité puisse être à la fois « consommatrice » et/ou « productrice » de service

Ceph : l'architecture répartie



Ceph : la *crushmap* (1/2)



CRUSH :

Controlled Replication Under Scalable Hashing

Ceph : la *crushmap* (2/2)

CRUSH MAP

- DEVICE — Disque unitaire = OSD
- TYPE

```
# types
type 0 osd
type 1 storage
type 2 host
type 3 rack
type 4 datacenter
type 5 site
type 6 diskclass
type 7 class
type 8 root
```

Types de buckets

- BUCKET

```
diskclass sata.c1 {
        item belinomp.sata.c1
        item 1cnups.sata.c1
}
```

Objet de l'arborescence

- RULE

```
rule rule_r3_host_sata_c1 {
  ruleset 1
  type replicated
  min_size 2
  max_size 3
  step take sata.c1
  step chooseleaf firstn 0 type datacenter
  step emit
}
```

Règle de répartition

Arborescence Ceph

```
root clusterc4
    class c1
        diskclass sata.c1
            site belinomp.sata.c1
                datacenter d2.belinomp.sata.c1
                    rack 12.d2.belinomp.sata.c1
                        host host1.sata.c1
                            osd.69          up
                            osd.70          up
                            ...
                    rack 10.d2.belinomp.sata.c1
                        host host5.sata.c1
                            osd.40          up
                            osd.41          up
                            osd.42          up
                            ...
                datacenter b2.belinomp.sata.c1
                    rack 17.b2.belinomp.sata.c1
                        host host6.sata.c1
                            osd.10          up
                            osd.11          up
                            osd.12          up
                           ...
```

Ceph : intégration / suppression

Intégration

Installer selon choix géographique

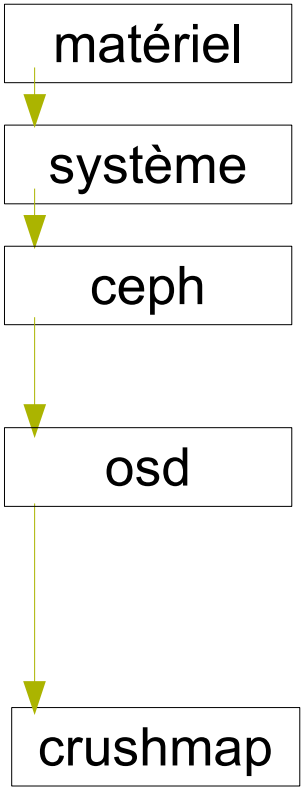
Installer le système (preseed + scripts)

Déployer ceph
ceph-deploy install --release ...

Déployer les osd
ceph-deploy disk prepare ...
ceph-deploy disk activate ...

« Renommer » le nouvel hôte
***ceph osd crush rename-bucket ***
hostx hostx.sas.c1

« Positionner » le nouvel hôte
***ceph osd crush move hostx.sas.c1 ***
rack=12.d2.belinomp.sas.c1



Ceph : intégration / suppression

Intégration

Installer selon choix géographique

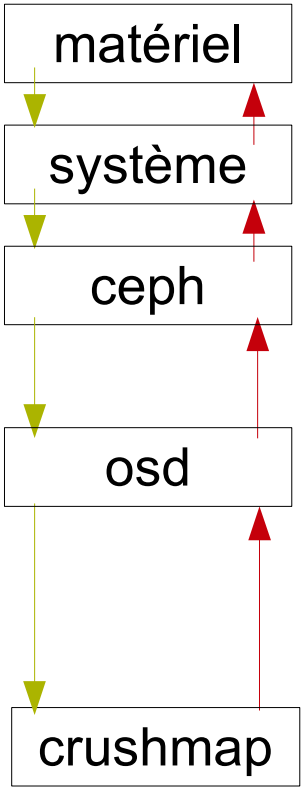
Installer le système (preseed + scripts)

Déployer ceph
ceph-deploy install --release ...

Déployer les osd
ceph-deploy disk prepare ...
ceph-deploy disk activate ...

« Renommer » le nouvel hôte
***ceph osd crush rename-bucket ***
hostx hostx.sas.c1

« Positionner » le nouvel hôte
***ceph osd crush move hostx.sas.c1 ***
rack=12.d2.belinomp.sas.c1



Suppression

Retirer matériel et mise en pièces

Arrêter le système

Supprimer les paquets ceph

Supprimer les osd
ceph osd rm {x ... n}

Arrêter les démons osd
systemctl stop ceph-osd@{x .. n}

Supprimer les osd puis l'hôte
ceph osd crush remove hostx.sas.c1
ceph osd crush remove osd.{x ... n}

Rendre les osd indisponibles
ceph osd out osd.{x ... n}

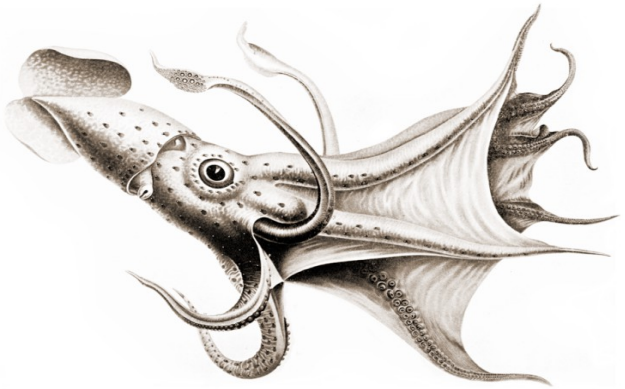
Conclusion et perspectives

- Le stockage objet sera l'une des solutions de demain
  - L'offre existe aujourd'hui (Scality, Datacore, Ceph, etc.)
- Ceph s'est avéré être un bon outil
  - Stockage distribué, sécurisé et pérenne
  - *Crushmap* permettant d'appliquer la « bonne » politique de stockage (répartition géographique en particulier)
  - Modèle adapté aux structures distribuées où chaque laboratoire participant peut être « producteur » et/ou « consommateur »
- Le modèle « matériel recyclé » est :
  - Peu coûteux financièrement
  - Adapté au cas d'usage de l'archivage

L'objectif initial est atteint :
créer un projet fédérateur autour du stockage pour l'archivage

- Questions ouvertes à plus long terme :
  - Le vieillissement du matériel
  - Le coût humain consacré à l'administration de la solution

Merci de votre attention



Une infrastructure mutualisée et recyclée avec Ceph

Ceph : le stockage

Object, Pool, Placement group, OSD