.

How and why migrate from a centralized storage solution to a distributed architecture

Eric.mourgaya@arkea.com

Crédit Mutuel ARKEA

Plan

- Problématique
- Pourquoi choisir une solution distribuée opensource ?
- Les freins à la migration
- En quoi ceph répond à la problématique ?
- Quelques mots sur notre infrastructure
- Les services que nous offrons aux autres équipes
- La Migration des données
- Le futur

La Problématique

- Changement de baies de stockage tous les 3 ans
  - Temps de migration des données important
  - Indisponibilités et risque de corruption de la données
  - Finalement tout cela est moins fiable qu'on ne le pense
  - Sous estimation du besoin même sur 3 ans

- Prix du stockage
  - Un vrai casse tête
  - Classification de la donnée

- Les contraintes sur la nouvelle solution de stockage
  - La qos doit être au moins équivalente
  - Capitalisation sur nos infrastructures
    - Utilisation des disques des serveurs existants (en fait non !)
    - Augmentation de la disponibilité

Ceph une solution de stockage distribuée

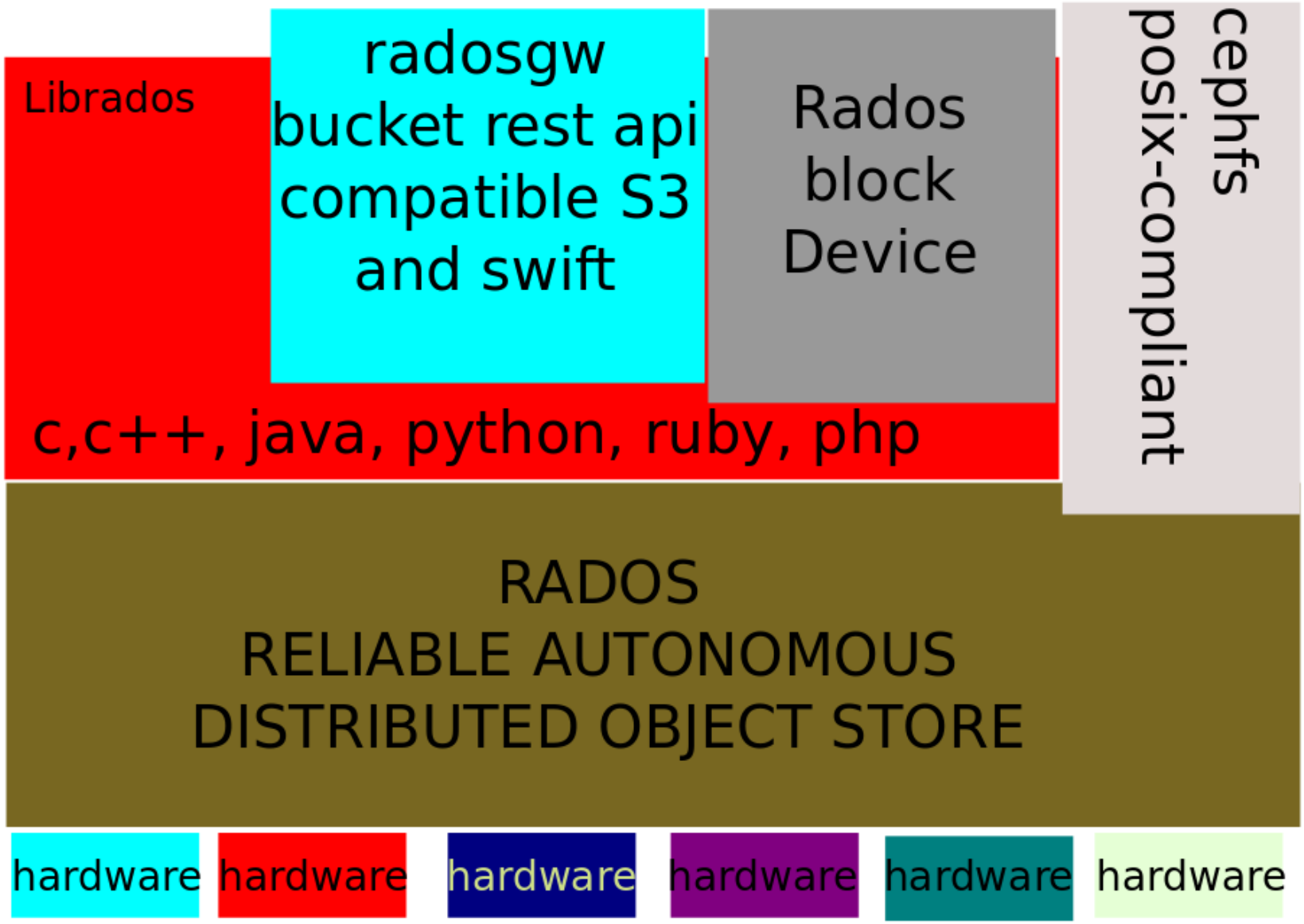
- Suite logique de la notion de clusters
- Mutualisation des ressources (cpu/disque/ram)
- Augmentation facile de la capacité de stockage
- Adéquation avec les autres solutions telles que :
  - Hadoop
  - Openstack
- Opensource : Rentre dans les standards du groupe Arkea
- Distribué : Tout comme l'opensource, devient un standard Arkea
  - Tolérence aux pannes
  - Passage à l'echelle

L’Acceptation de ce type de solutions

- Les équipes techniques
  - Administration plus difficile pour les équipes de gestion du stockage
- Souvent Projet difficile à prioriser
- Technologie encore jeune
- Changement dans les mentalités des décideurs
- Adaptation des développements aux protocoles tels que S3, rbd etc.
- Le budget, pour économiser il faut dépenser !
- Bref un vrai changement

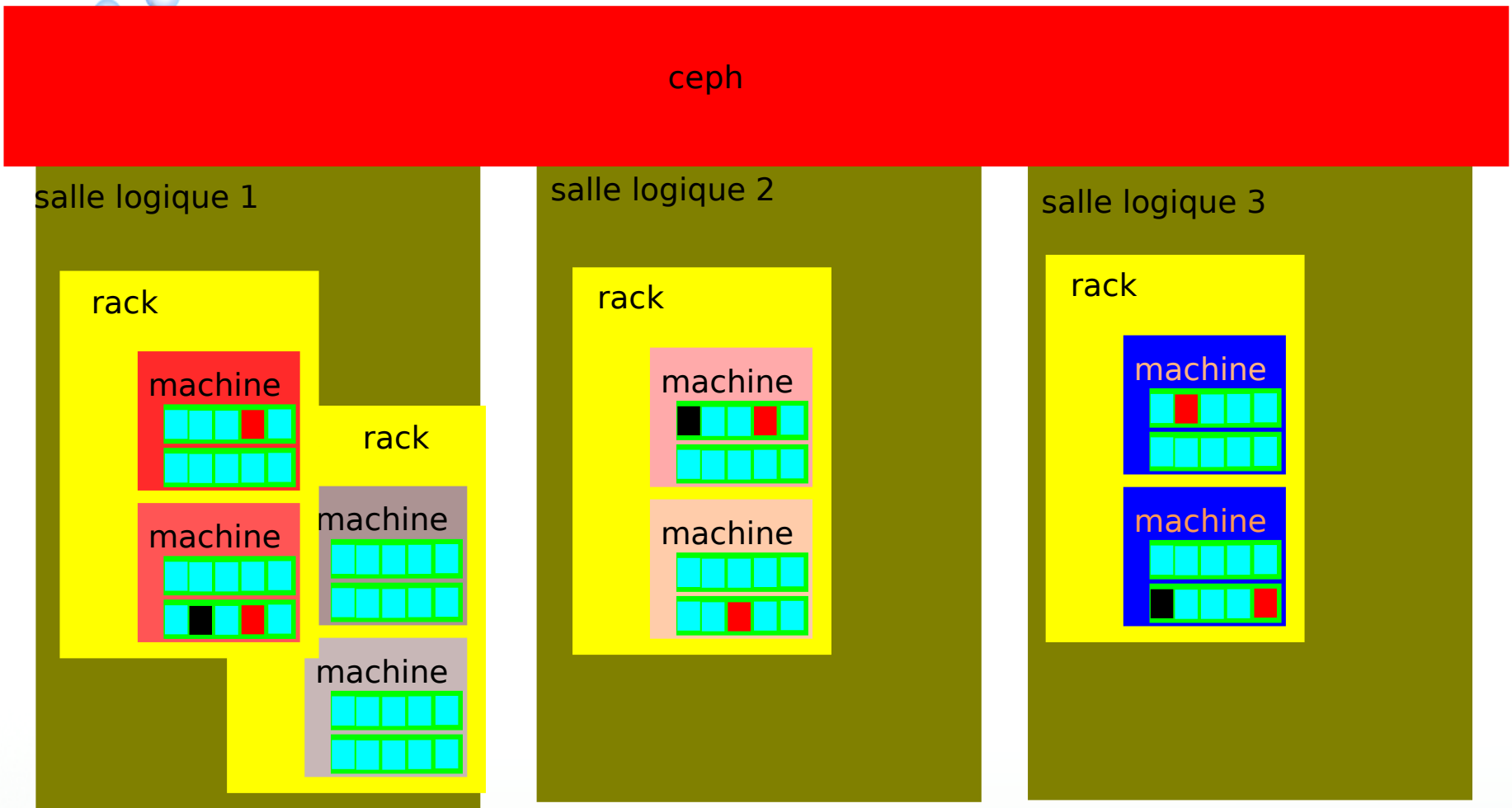
Levée des freins

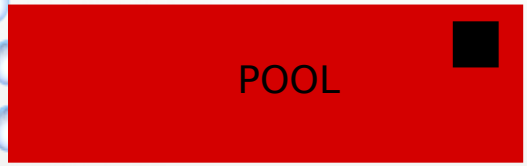
- Formation des équipes de gestion du stockage
- Collaboration avec les équipes d'Orange labs pour la création d'une interface d'administration : Inkscope
- Création d'Api et support aux équipes de développement
- Mise en place d'un POC et stockage de données non critiques pendant plus d'un an
- Adéquation avec la demande coté développeur


Librados
radosgw
bucket rest api
compatible S3
and swift
Rados
block
Device
cephfs
posix-compliant
c,c++, java, python, ruby, php
RADOS
RELIABLE AUTONOMOUS
DISTRIBUTED OBJECT STORE
hardware
hardware
hardware
hardware
hardware
hardware

LA CRUSHMAP

localisation de la données par calcul





replicated or erasure coded

une copie par disque, par machine, par rack ou par salle

Donc finalement, nous avons

- Des moniteurs : Pour la gestion de la cohérence
- Des OSD : Pour le stockage des données
- Des pg : Correspond à un découpage logique des disques (sorte de physical volume unitaire)
- Des pools : regroupement de pg pour avoir une notion de volume groupe
- Des image rbd : stocké au sein d'un pool notion équivalente aux luns
- Des Fonctionnalités de réplication, snapshot, auto-réparation, de type de stockage, etc.
- Des protocoles d'accès multiples

CEPH chez Arkea

- Pour la production un cluster sur deux salles
- Réplication triple des données (je sais c'est pas top quand on a deux salles!)
- Gestion des confs par chef
- Inkscope pour l'administration au quotidien
- Un cluster secours sur un troisième site (replication asynchrone des données)
- Un cluster {(pour la) /(de)} recette (Je sais c'est pas top non plus)
- Des protocoles d'accès multiples
  - S3
  - rbd
  - Cephfs
  - Front nfs actif /passif (possibilité actif/actif avec nfs ganesha)
  - Librados
  - Iscsi (mauvaise expérience)

La migration de données vers ceph

- Par simple copy pour le rbd
- Depuis Jewel utilisation de la gateway nfs permettant un accès aux stores s3 via nfs
- Modification des applicatifs pour l'utilisation de
  - Librados
  - S3
- Ajout d'une interface réseau d'accès aux données par machine cliente (accès direct aux osd)

Les cas d'usages

- Stockage de données peu critique
  - Sauvegarde des poste de travail
  - Sauvegarde des bases de données de recette
  - Sauvegarde de la messagerie
- Backend de stockage images openstack
- Utilisation direct de librados au niveau applicatif
- Stockage S3 pour les batches et les applicatifs
- Stockage de données pour seafile

Le futur

- Consolidation de la réplication des données sur le troisième site
- Tests de PRA en réel
- Identification des données éligibles à une stockage ceph
- Ajout d’une arborescence de stockage orienté performance
- Consolidation des mises à jour de CEPH

Comment CEPH répond à la problématique

- Plus de migration de données
- Archivage et stockage de données
- Scalable
- Flexibilité sur les protocoles d’accès à la données
- Intéropérabilité avec
  - Les applications
  - Hadoop
  - Openstack

Conclusion

- Les technologies distribués ont fait leur preuves
- CEPH un couteau suisse
- CEPH respecte aux contraintes applicatives (performance/coût/protocole)
- Recherche de contributeurs pour Inkscope
- Cependant tout n’est pas parfait
  - Nécessite plus que des compétences d’admin de stockage traditionnel
  - Support interne pour les équipes de développement
  - Procédure de montée de version à Sécuriser