



CNRS - DSI

## Retour d'expérience du CNRS sur Scality

Instituts  
thématiques



**Inserm**

Institut national  
de la santé et de la recherche médicale



# Plan de la présentation

- 1 Contexte global
- 2 Choix de la solution de stockage
- 3 Scality plus en détails
- 4 Scality après quelques mois d'utilisation
- 5 Annexes



# Plan de la présentation

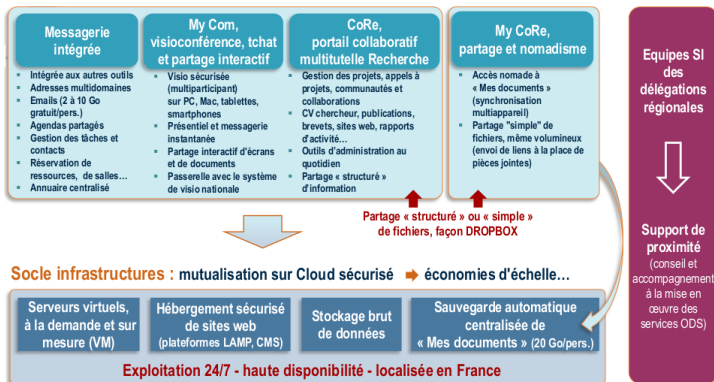
- 1 Contexte global
- 2 Choix de la solution de stockage
- 3 Scality plus en détails
- 4 Scality après quelques mois d'utilisation
- 5 Annexes

# Vue globale des offres de service

**Objectif : simplifier et sécuriser le quotidien des unités \***

- proposer des outils adaptés aux métiers de la recherche et à leurs contraintes de sécurité
- permettre aux unités de se décharger au besoin de certaines tâches « techniques »

**Ecosystème utilisateurs : intégration native des outils ➡ simplicité d'utilisation. Support local**



\* Les laboratoires, entités administratives et directions fonctionnelles peuvent bénéficier de ces services

Plus d'informations sur ces services sur <http://ods.cnrs.fr>

David Rousse | CNRS - DSI | Mai 2016

# Genèse du service My CoRe<sup>(1/2)</sup>

P. 5

## Deux besoins métier identifiés<sup>(via des enquêtes utilisateurs)</sup>

- Solution de synchronisation et de partage de fichiers, alternative aux solutions de "type Dropbox"
- Solution de sauvergarde des postes de travail

## Une seule solution choisie

- ownCloud (en version communautaire) car il a la meilleure couverture fonctionnelle par rapport aux besoins métier, est bien accueilli par les utilisateurs et est déjà utilisé dans d'autres entités
- Service déployé dans le centre serveur CNRS de l'IN2P3, afin de maîtriser les modalités d'hébergement et de disposer de la solution locale de sauvegarde du centre de calcul
- Service exploité par un prestataire de la DSI du CNRS, Atos<sup>(ex-Bull)</sup>, afin de couvrir la plage de service la plus large possible et faute de pouvoir le faire en interne par manque de ressources

## Genèse du service My CoRe<sup>(2/2)</sup>

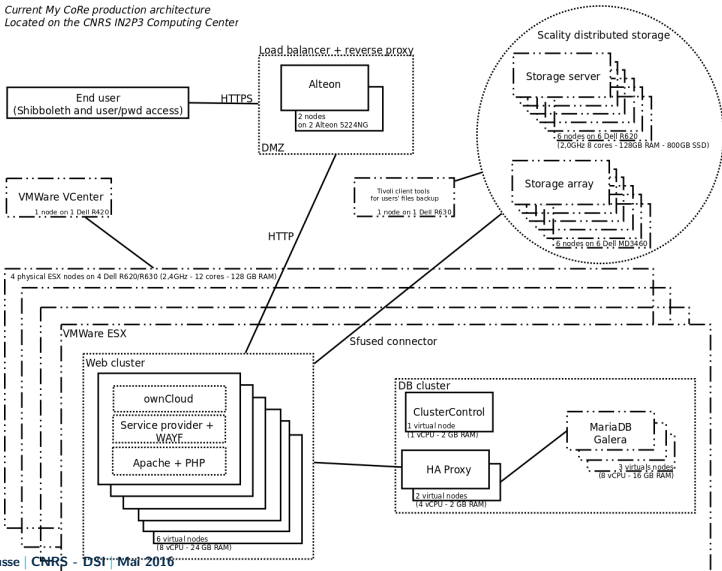
### Macro-étapes du projet

- ☐ Janvier à septembre 2013 : étude préalable
- ☐ Octobre 2013 à mai 2014 : cadrage fonctionnel et technique, dont le **choix de Scality en mai 2014**
- ☐ Juin à décembre 2014 : mise en œuvre
- ☐ Janvier à septembre 2015 : service pilote sur les unités de trois délégations régionales du CNRS
- ☐ Depuis octobre 2015 : service productif ouvert à tous les agents travaillant dans des unités CNRS<sup>(20 Go par agent)</sup>

# Architecture actuelle du service My CoRe

P. 7 / 26

Current My CoRe production architecture  
Located on the CNRS IN2P3 Computing Center





# Plan de la présentation

- 1 Contexte global
- 2 Choix de la solution de stockage
- 3 Scality plus en détails
- 4 Scality après quelques mois d'utilisation
- 5 Annexes



# Besoin au niveau de la couche de stockage

## Backend de stockage pour My CoRe avec

- ☐ une forte résilience, car pas de "sauvegarde",
- ☐ qui soit unique pour toute la volumétrie,
- ☐ un accès "file system" stable,
- ☐ un coût au Go faible,
- ☐ la possibilité d'être multi sites.

# Étapes du choix de Scality<sup>(1/2)</sup>

## Délai et manpower faibles pour faire ce choix

- Tour d'horizon sur "papier" des solutions (HDFS, CephFS, GlusterFS, iRODS)
- Définition d'une short list
  - ▶ Dell Compellent : le plus économique mais volumétrie limitée à 2 Po et résilience limitée
  - ▶ EMC Isilon : le plug and play mais matériel spécifique
  - ▶ Scality : le plus souple
- Étude de Scality en détail : maquettag<sup>(voir en annexe)</sup> et récupérations de divers retours d'expérience sur la solution
- **Choix final de Scality en mai 2014**

# Étapes du choix de Scality<sup>(2/2)</sup>

P. 11 / 26

Critère	Scality	Dell Com- pellent	EMC <sup>2</sup> Isilon
Architecture	Solution distribuée adaptée aux stratégies cloud	Solution industrielle centralisée	Solution industrielle distribuée
Maturité	Solution récente mais disposant de bons ReX	Excellente	Excellente
Administrabilité	Exploitation complexe mais outillage suffisant	Très industrialisée	Solution la plus simple à exploiter
Coûts	Plus cher à la cible (dans le strict contexte projet)	Moins cher à la cible	Coût à la cible important
Evolutivité	Forte (multi projets et multi sites)	Faible	Bonne (mais en dessous de Scality)
Qualité de service attendue	Pas de différenciation nette	Pas de différenciation nette	Pas de différenciation nette
Conclusion	Solution la plus souhaitable mais la plus onéreuse	Solution qui répond aux besoins projet de stockage, sans permettre d'évolution	Solution facile d'utilisation mais trop liée à un matériel donné (par opposition à Scality)

Détail dans le §3.2 de [https://github.com/CNRS-DSI-Dev/mycore\\_press/blob/master/JRES2015/JRES-20151208-article.pdf](https://github.com/CNRS-DSI-Dev/mycore_press/blob/master/JRES2015/JRES-20151208-article.pdf)



# Plan de la présentation

- 1 Contexte global
- 2 Choix de la solution de stockage
- 3 Scality plus en détails**
- 4 Scality après quelques mois d'utilisation
- 5 Annexes

# Éléments clés

## Le RING

- ☐ Stockage logiciel d'objets
- ☐ Capacité illimitée (scale-out)
- ☐ Stockage mutualisé
- ☐ ARC

## Les points forts

- ☐ Compatible tout serveur x86
- ☐ Ratio brut/utile d'environ 1.6
- ☐ Très hautement disponible
- ☐ Pas de RAID matériel

# Rôles et architecture

## 3 types de serveurs

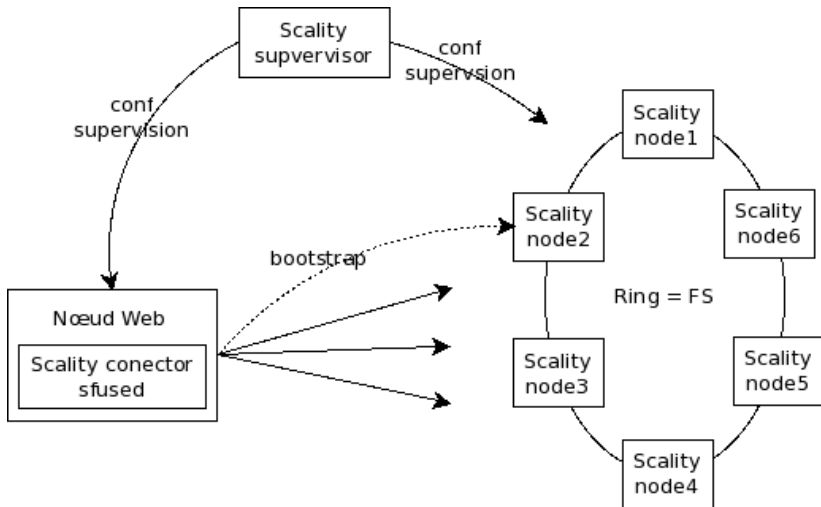
- ❑ Scality-node : Nœud de stockage
- ❑ Scality-supervisor : Manager pour la conf. et le monitoring
- ❑ Scality-connector : Accès client sur le stockage

## Architectures possibles

- ❑ Au minimum 6 serveurs, pas de maximum
- ❑ Un ou plusieurs RING
- ❑ Géo-distribution ou réplication

On peut mixer toutes ces possibilités :-)

## Schéma de principe



# Solution Scality sur My CoRe

## Configuration Scality actuelle

- ☐ 300 To de stockage brut (240 To réel), mais à ce jour 7 To utilisés (800000 répertoires, 8 millions de fichiers)
- ☐ 2 RING Scality, 1 de data, 1 de metadata (celui de metadata est sur des disques SSD)
- ☐ Pas de stockage des disques de VM sur Scality (pas adapté)
- ☐ ARC(9,3) au lieu de ARC(4,2) pour optimiser les coûts
- ☐ 12 nœuds (au sens Scality) par serveur physique, avec 6 serveurs physiques auxquels sont adossés 6 baies de stockage
- ☐ Connecteur Scality en mode fichier (SFUSED) avec 8 Go de RAM sur les VM clientes
- ☐ 2 interfaces réseaux 10GBs sur chaque serveur de stockage, configurées en actif/passif, sur un réseau dédié

## Evolutions possibles de la configuration autour de Scality

- ☐ Passage du mode fichier (SFUSED) au mode objet depuis les VM clientes ownCloud si les améliorations du connecteurs SFUSED annoncées début 2016 ne sont pas suffisantes
- ☐ Chiffrement du système de fichiers via Scality



# Scality après 16 mois d'utilisation

## Les points positifs ...

- ☐ Solution stable et effectivement résiliente (2 incidents majeurs sans perte de données)
- ☐ Support Scality compétent et réactif

## ... et les points moins positifs !

- ☐ Limitations du connecteur SFUSED (mauvaise performance des accès au delà de 2 millions de fichiers lors de certaines opérations, type listing, et corruption de fichiers lors d'accès concurrents en écriture sur un même objet)
- ☐ Mode start-up de Scality pas toujours adapté à des contraintes d'exploitation dans un cadre infogéré
- ☐ Toujours pas open source :-)



P. 18 / 26

## Questions et réponses

DS<sup>My</sup>  
CORE

Merçi de votre attention !

## Annexe 1 : liens utiles

### URLs en relation avec My CoRe

- ReX Scality pour le CNES = [https://github.com/CNRS-DSI-Dev/mycore\\_press/blob/master/CNES-CNRS-Scality-20140619.pdf](https://github.com/CNRS-DSI-Dev/mycore_press/blob/master/CNES-CNRS-Scality-20140619.pdf)
- Presse My CoRe au JRES 2015 = [https://github.com/CNRS-DSI-Dev/mycore\\_press/tree/master/JRES2015](https://github.com/CNRS-DSI-Dev/mycore_press/tree/master/JRES2015)
- Pourquoi Scality ? Paragraphe 3.2 = [https://github.com/CNRS-DSI-Dev/mycore\\_press/blob/master/JRES2015/JRES-20151208-article.pdf](https://github.com/CNRS-DSI-Dev/mycore_press/blob/master/JRES2015/JRES-20151208-article.pdf)
- Autres présentations et informations = [https://github.com/CNRS-DSI-Dev/mycore\\_press](https://github.com/CNRS-DSI-Dev/mycore_press)



## Annexe 2 : maquettage avec le support Scality

P. 20 / 26

### Réalisation d'une maquette avant de valider le choix ...

- ☐ Voir le logiciel en action
- ☐ Tester le support Scality

### ... la plus minimale possible

- ☐ Deux VM pour les nœuds de stockage
- ☐ Une VM pour le supervisor
- ☐ Deux VM clientes LAMP

## Annexe 3 : classe de Service de Scality

### Le fichier sfused.conf

- Node de démarrage
- Nombre de copies distribuées
  - ▶ Par type de fichiers
  - ▶ Par expressions régulières
  - ▶ Par connector (donc par application)
- Niveau de mise en cache

```
{
  "general": {
    "mountpoint": "/owncore-ring",
    "file_cos": 2,
    "cat_cos": 4,
    "cat_page_cos": 2,
    "dir_cos": 4,
    "dir_page_cos": 2,
    "rootfs_cos": 4,
    [...]
  },
  "ring_driver:0" : {
    "type": "chord",
    "bootstraplist": "192.168.55.21:4244,192.168.55.22:4244",
    [...]
  },
  "cache:0" : {
    "ring_driver": 0,
    "size": 2000000000,
    [...]
  },
}
```



## Annexe 4 : commandes système liées à Scality

P. 22 / 26

### Intégré au système

- ☐ Dépôt RPM utilisable avec YUM
- ☐ Gestion des démons standards
- ☐ Le RPM RingSH permet d'utiliser le supervisor en CLI !
- ☐ Scripts de déploiement Salt (Puppet like)

### Exemple : Initialiser un nœud de stockage

```
# yum install scality-node -y  
# scality-node-config --prefix /mnt/disk --disks 2 --nodes 6 --ip  
<ip_scality-node>--supervisor-ip <ip_scality-supervisor>
```



## Annexe 5 : le Supervisor de Scality


### Couple WebUI et sagentd

- ☐ Ajouter/enlever des nodes à chaud
- ☐ Etendre le RING à chaud
- ☐ Remonter les incidents matériels
- ☐ Tableau de bord
- ☐ Métrologie



# Annexe 6 : le provisioning dans Scality

P. 24 / 26

 **SCALITY** Supervisor Administration

Logged in as root

Hardware | Logout

Local > owncore

Dashboard | Operation | Administration | Provisioning

**Status**  
State RUN  
Autojoin Off  
Online 12 Nodes  
0 RS2 Connectors  
Alerts 0

**Storage capacity**  
# Objects  
# Unique objects 1  
Average size 0.00 KB  
Avg size (unique) 0.00 KB  
  
Unique 0.00 % 0 GB  
Stored 0.00 % 0 GB  
Used 1.84 % 0 GB  
Available 98.16 % 20.00 GB  
Total 21.00 GB

**Welcome to the provisioning wizard, step one: choose the servers you want to configure.**

	Server	Zone	State	# nodes
<input checked="" type="checkbox"/>	192.168.55.21		RUN	<span>6</span>
<input checked="" type="checkbox"/>	192.168.55.22		RUN	<span>6</span>

Warning! We recommend having at least 6 servers.

Configure

[Manual provisioning]

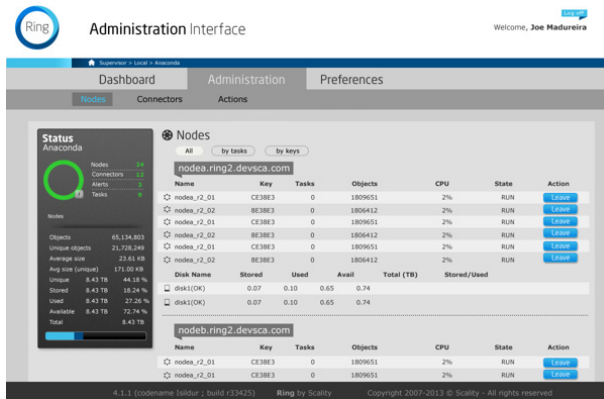
4.2.5 (codename Jago ; build r38965)

Copyright 2007-2013 © Scality - All rights reserved

Impression d'écran sur une version ancienne de Scality



# Annexe 7 : la webUI dans Scalify<sup>(1/2)</sup>



**Status**  
Anaconda

Nodes: 24  
Connectors: 0  
Alerts: 3  
Tasks: 8

Objects: 65,134,803  
Unique objects: 21,728,249  
Average size: 23.61 KB  
Avg size (unique): 171.00 KB  
Unique: 8.43 TB 44.16 %  
Stored: 8.43 TB 18.24 %  
Used: 8.43 TB 27.26 %  
Available: 8.43 TB 72.74 %  
Total: 8.43 TB

**Nodes**

All by tasks by keys

nodea.ring2.devscs.com

Name	Key	Tasks	Objects	CPU	State	Action
nodea_r2_01	CE3BE3	0	1809651	2%	RUN	<a href="#">Leave</a>
nodea_r2_02	BE3BE3	0	1806412	2%	RUN	<a href="#">Leave</a>
nodea_r2_01	CE3BE3	0	1809651	2%	RUN	<a href="#">Leave</a>
nodea_r2_02	BE3BE3	0	1806412	2%	RUN	<a href="#">Leave</a>
nodea_r2_01	CE3BE3	0	1809651	2%	RUN	<a href="#">Leave</a>
nodea_r2_02	BE3BE3	0	1806412	2%	RUN	<a href="#">Leave</a>

Disk Name	Stored	Used	Avail	Total (TB)	Stored/Used
disk1(OK)	0.07	0.10	0.65	0.74	
disk1(OK)	0.07	0.10	0.65	0.74	

nodeb.ring2.devscs.com

Name	Key	Tasks	Objects	CPU	State	Action
nodea_r2_01	CE3BE3	0	1809651	2%	RUN	<a href="#">Leave</a>
nodea_r2_01	CE3BE3	0	1809651	2%	RUN	<a href="#">Leave</a>

4.1.1 (codename Isidur ; build r33425) Ring by Scalify Copyright 2007-2013 © Scalify - All rights reserved

Impression d'écran sur une version ancienne de Scalify



*Impression d'écran sur une version ancienne de Scality*

