Étude et mise en œuvre d'une solution de stockage distribué

Sébastien Thiaux

CONSERVATOIRE NATIONAL DES ARTS & METIERS
Centre Régional associé de Rennes

Mémoire présenté en vue
d'obtenir le diplôme d'ingénieur **C.N.A.M.**
en informatique

Sébastien Thiaux

Étude et mise en œuvre d'une solution de stockage distribué

soutenu le 23 juin 2014

JURY

PRESIDENT :	Professeur POLLET
MEMBRES :	M. DUGUE
	M. BLAISOT
	M. LE GUERN

Remerciements

Au terme de ce travail, je tiens à remercier tous ceux qui ont contribué à la réalisation de ce projet. En particulier, j'adresse mes remerciements :

A **Sébastien Blaisot** et **Christophe Le Guern** pour m'avoir permis de réaliser ce mémoire chez PagesJaunes et avoir soutenu ma démarche de formation au CNAM depuis 2009,

A **Yoann Dugué** pour sa disponibilité et ses nombreux conseils qui ont conduit à la réalisation et à l'amélioration de ce mémoire,

Aux membres l'**équipe projet** pour leur enthousiasme et leur implication, et particulièrement à **Jean-Maxime Leblanc** qui m'a été d'une aide précieuse,

A mes collègues de l'équipe systèmes qui ont repris mes travaux pendant 6 mois et m'ont permis de me consacrer à ce projet dans de bonnes conditions,

A mes amis et ma famille qui m'ont encouragé.

Mes derniers remerciements vont à ma compagne Audrey qui m'a aidé, soutenu et surtout supporté.

Résumé

L'explosion de la volumétrie des données dans les systèmes informatiques impose de repenser les modèles de stockage traditionnels vers une approche maximisant la flexibilité et l'extensibilité.

L'amélioration des performances du matériel informatique standard permet aujourd'hui d'envisager de nouvelles alternatives. Ce mémoire porte sur l'étude des nouvelles technologies de stockage dites « logicielles » et leurs adaptations dans le contexte d'hébergement d'applications Internet de PagesJaunes.

Après avoir mené une étude comparative sur trois produits et sélectionné le plus adapté, ce mémoire se penche sur l'architecture de la solution retenue et son intégration dans le système d'information de PagesJaunes.

Des tests approfondis du système ont démontré d'excellentes qualités en termes de fiabilité et de performance. Les résultats obtenus permettent d'envisager un déploiement et une utilisation à grande échelle qui permettra à PagesJaunes de répondre aux nouveaux défis de l'hébergement d'applications web.

Mots clés : stockage - stockage distribué - stockage logiciel - SDS - logiciel open source - Cloud

Abstract

The explosive growth in the volume of data in computer systems requires rethinking traditional models of storage for an approach that maximizes flexibility and extensibility.

Recent performance improvements of standard hardware allow us to consider new alternatives. These alternatives focus on new storage technologies called "software defined storage" and the adaptation of this in the context of hosting Internet applications at PagesJaunes.

After conducting a comparative study on three products and selecting the most suitable, this paper focuses on the architecture of the solution and its integration into the information system of PagesJaunes.

Extensive tests of the chosen system have revealed excellent qualities in terms of reliability and performance. These impressive results could lead to widespread deployment and use that will allow PagesJaunes to rise up to future challenges of web hosting applications.

Key words: storage - distributed storage - software defined storage - SDS - open source software - Cloud

Table des matières

1. Introduction

1.1. Le contexte

Ce mémoire a été réalisé de décembre 2013 à juin 2014, dans le cadre de l'obtention du diplôme Ingénieur CNAM Informatique option Architecture et Ingénierie des Systèmes et des Logiciels (AISL). Il est pour moi l'occasion de conclure un cursus CNAM, initié il y a quatre ans, en travaillant sur un projet d'infrastructure informatique innovant et fortement structurant pour mon entreprise, PagesJaunes.

1.2. Présentation du sujet

Les acteurs du domaine Internet tel que PagesJaunes font face aujourd'hui à de profonds bouleversements dans l'organisation de leurs systèmes informatiques. Afin de rester compétitifs, dans un marché toujours plus concurrentiel, ils doivent adapter leurs infrastructures à de nouvelles contraintes de flexibilité et d'agilité.

L'avènement du Cloud Computing a été un premier changement majeur dans l'organisation des systèmes d'information. L'abstraction des ressources matérielles rendue possible par la virtualisation a permis de gagner en souplesse et déployer de nouveaux services en très peu de temps.

Cependant, le stockage a lui toujours été considéré comme une composante périphérique de l'infrastructure informatique. Pourtant, devant l'explosion de la volumétrie des données, il semble nécessaire de repenser les systèmes de stockage et de proposer de nouvelles technologies qui permettraient aux entreprises d'affronter les défis d'optimisation qui s'imposent.

1.3. L’objectif

L'objectif du projet est donc d'identifier, d'étudier et de mettre en œuvre une technologie de stockage pouvant accompagner l'entreprise PagesJaunes dans sa stratégie numérique. Cette stratégie étant axée sur la mise à disposition de nouveaux contenus et de nouveaux services aux internautes, la solution de stockage doit donc pouvoir s'adapter aux fortes variations de volumétrie tout en restant flexible et peu onéreuse.

1.4. Présentation du plan

Ce mémoire s'articule autour de sept grandes parties, représentatives de la démarche adoptée pendant la durée du projet.

Après une première partie consacrée à la présentation de l'entreprise et de l'activité d'hébergeur assurée par mon service, j'aborderai dans un deuxième volet une analyse de l'existant et les différents modes d'utilisation du stockage actuellement utilisés dans le cadre de l'hébergement d'applications Internet.

Dans une troisième partie, j'expliquerai ma démarche de recueil du besoin pour ce nouveau projet et j'identifierai les principaux cas d'utilisation envisagés.

La quatrième partie sera consacrée à l'étude globale dont l'objectif est d'identifier les technologies répondant aux besoins dans le respect des contraintes imposées. Cette partie se conclura par le choix d'une solution qui sera étudiée en détail dans le cinquième volet.

Dans la cinquième partie, j'expliquerai la technologie utilisée et les architectures à mettre en place afin de répondre aux cas d'utilisation, puis je clôturerai ce point par l'élaboration d'un plan de qualification.

La sixième partie sera consacrée à la réalisation, c'est-à-dire à la qualification de la solution et à son industrialisation.

Enfin, le septième et dernier volet sera consacré à la mise en œuvre de la solution et traitera du déploiement, de la recette et des formations.

1.5. Gestion de projet

1.5.1. L'organisation du projet

Pour piloter ce projet, j'ai utilisé une démarche basée sur le principe des jalons. J'ai découpé le projet en quatre grandes parties, représentatives des grandes étapes du projet. Chaque partie est séparée par un jalon qui permet de valider le passage à l'étape suivante.

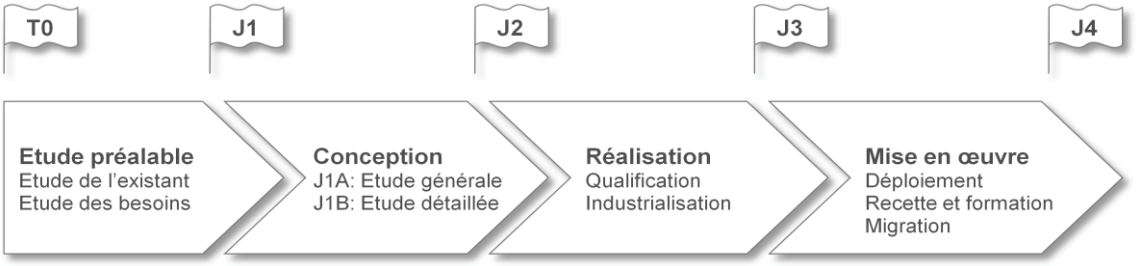

Figure 1 : Les jalons du projet

1.5.2. Mes rôles dans le projet

Tout d'abord, ce projet aura été pour moi l'opportunité d'endosser un rôle de chef de projet sur un projet de moyenne ampleur mais à fort potentiel stratégique. En tant que chef de projet j'ai donc assuré les responsabilités suivantes :

- Planifier les lots de travail
- Animer l'équipe projet : organiser les réunions, rédiger les comptes rendus
- Piloter et suivre les actions
- Maîtriser les délais et les coûts
- Gérer le matériel serveur : contacter les fournisseurs, demander les devis, commander le matériel
- Gérer les risques
- Améliorer la démarche qualité

Ensuite, en tant qu'ingénieur système j'ai assuré le pilotage technique des grandes étapes du projet, de l'analyse de l'existant à la mise en œuvre finale. Pour cela, j'ai mis au service du projet mon expérience et mes compétences dans les actions suivantes :

- Identifier les technologies et réaliser les choix
- Concevoir les architectures
- Elaborer la stratégie de tests, concevoir les tests et analyser des résultats
- Réaliser les optimisations logicielles et matérielles
- Rédiger les documentations et procédures

1.5.3. L'équipe projet

Système :
L'équipe systèmes est intervenue sur toutes les étapes du projet. J'ai été assisté dans les différentes étapes par Jean-Maxime Leblanc, ingénieur systèmes.

- Sébastien Thiaux (Chef de projet, Architecte et Ingénieur systèmes)
- Jean-Maxime Leblanc (Ingénieur systèmes)

Réseau :
L'équipe réseau a été impliquée dès le début du projet mais est intervenue techniquement de manière ponctuelle. Elle a été consultée principalement pour des problématiques d'architecture réseau, de sélection du matériel réseau et de performances réseau.

- Nicolas Raux (Architecte réseau)
- Vincent Libé (Architecte réseau)

Exploitation :
L'équipe d'exploitation a également été impliquée dès le début du projet et est intervenue sur les problématiques de supervision, métrologie, exploitation, installation du matériel et déploiement applicatif.

- Maëlig Herviault (Exploitant niveau 2)
- Peter Hart (Exploitant niveau 2)

Organigramme :

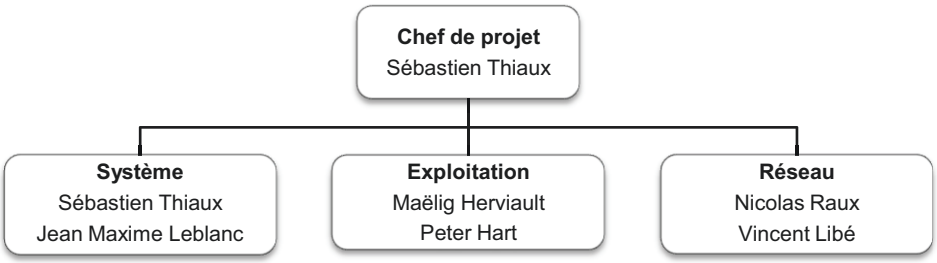

Figure 2 : Organigramme projet

1.5.4. Le planning

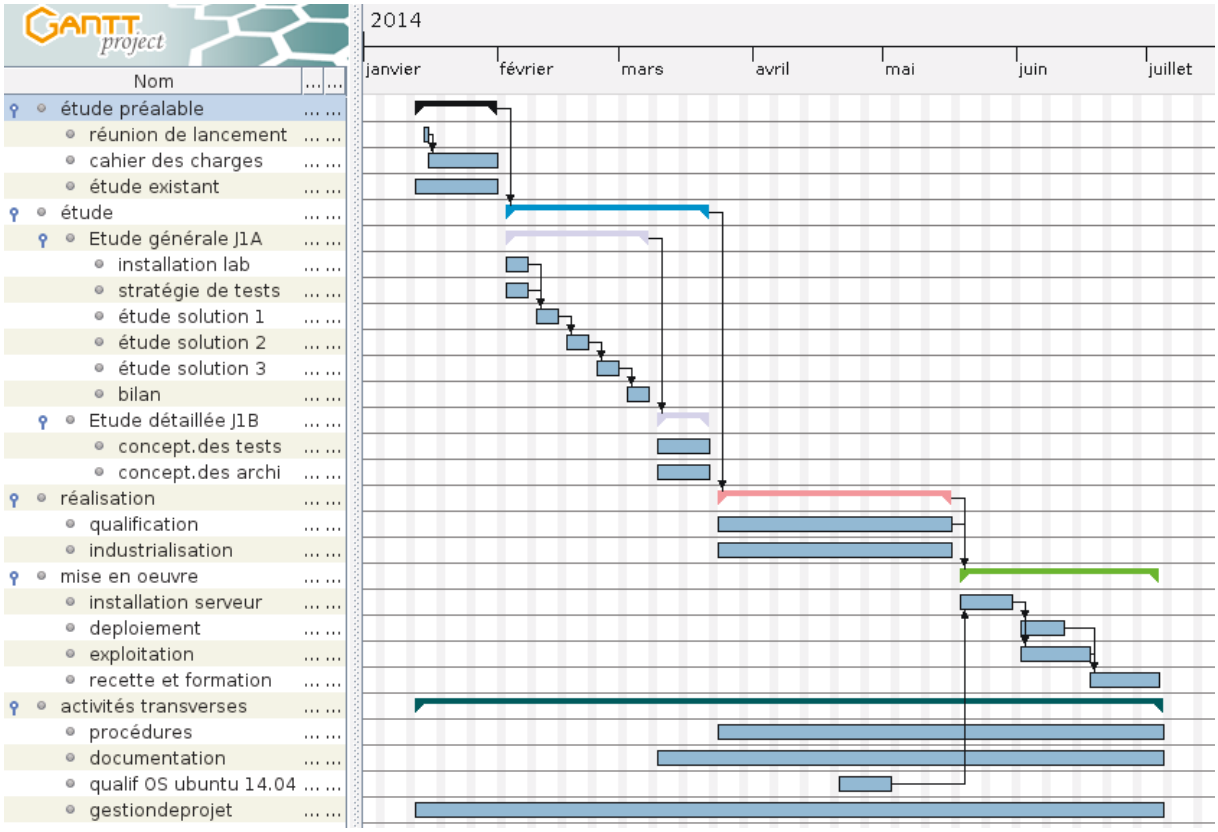

Figure 3 : Le planning du projet

2. Présentation de l'entreprise

2.1. L'organisation de l'entreprise

2.1.1. Le groupe Solocal

Activité :

Le groupe Solocal, anciennement groupe PagesJaunes, est un groupe français dont le cœur de métier est la recherche d'informations, la mise en relation et la publicité. Le groupe se positionne comme un interlocuteur de proximité pour les professionnels souhaitant communiquer localement. Son offre est constituée d'une gamme de produits et de services à destination du grand public et des professionnels.

Le modèle économique du groupe repose sur celui des medias : proposer des contenus de qualité générant de l'audience, monétiser cette audience, globale ou par segments, auprès des professionnels.

Au travers de ses filiales, Solocal est présent dans 3 métiers complémentaires :

- Editeur de contenus et services
- Média et conseil
- Régie publicitaire

Quelques chiffres[1] :

- Solocal : N°1 de la communication locale en France
- 998.9 millions d'euros de chiffre d'affaires en 2013
- 650 000 annonceurs
- Plus de 1.8 milliards de visites sur les sites du Groupe en 2013

[1] Sources : www.solocalgroup.com

L'organisation du groupe :

Le groupe Solocal est une holding cotée (SBF120, compartiment B), majoritairement détenue par la société Médiannuaire Holding. Ses activités opérationnelles sont gérées par des filiales implantées dans 4 pays : France, Espagne, Luxembourg et Autriche.

- PagesJaunes
- QDQ Media
- Mappy
- PJMS
- PagesJaunes Outre-Mer
- Eurodirectory
- Horizon Media
- Yelster Digital
- A vendre A louer
- Fine Media
- ClicRDV
- Leadformance
- Chronoresto

Figure 4 : Filiales du Groupe au 06 juin 2013

2.1.2. PagesJaunes S.A.

PagesJaunes S.A., principale filiale du groupe Solocal est le leader français de la publicité et de l'information locale sur Internet, mobile et annuaire imprimé.

PagesJaunes est aussi en France un des leaders des services de renseignements par téléphone avec le 118008, des petites annonces en ligne avec annoncesjaunes.fr et avendrealouer.fr et le premier créateur de sites Internet avec 120 000 sites dédiés aux professionnels.

Le site pagesjaunes.fr est le fer de lance de PagesJaunes. Il réunit des services de recherches (coordonnées, annuaire inversé, informations géolocalisées etc.), de représentations

géographiques (plans, itinéraires, 3D, vues aériennes, etc.) et d'informations pratiques (météo, trafic routier, webcams, réservation de spectacles, événements culturels, hôtels, etc.)

Quelques chiffres[2] :

- 981.5 millions de visites sur Internet fixe en 2013
- 342.5 millions de visites sur mobile en 2013
- Près de 8 Français sur 10 utilisent au moins un des médias PagesJaunes
- 112 000 sites Internet de professionnels créés par PagesJaunes
- 4 millions de professionnels référencés sur pagesjaunes.fr
- 3950 collaborateurs

2.1.3. Le pôle Business Solutions

Le pôle Business Solutions a pour fonction de développer les applications exposées aux clients (portails B2B, boutiques), de développer et maintenir les applications utilisées par les collaborateurs (outils commerciaux...), mais aussi de fournir des services d'hébergement d'infrastructure et infogérance d'applications développées par l'entreprise (Business Solutions, Pôle Media, ...).

Le pôle Business Solution est composé de 5 directions :

- **Sites :** développent de l'offre sites Internet pour professionnels du Groupe.
- **Adnet :** développement des applications exposées aux clients.
- **BSS (Business Solutions Strategy)**: roadmap, communication, budget.
- **BSA (Business Solutions Applications) :** développement des applications utilisées par les collaborateurs du groupe.
- **BSO (Business Solutions Opération) :** hébergement d'infrastructure, infogérance des applications, réseau, téléphonie et poste de travail.

[2] Sources : http://solocalgroup.com

2.1.4. La direction Business Solution Opération

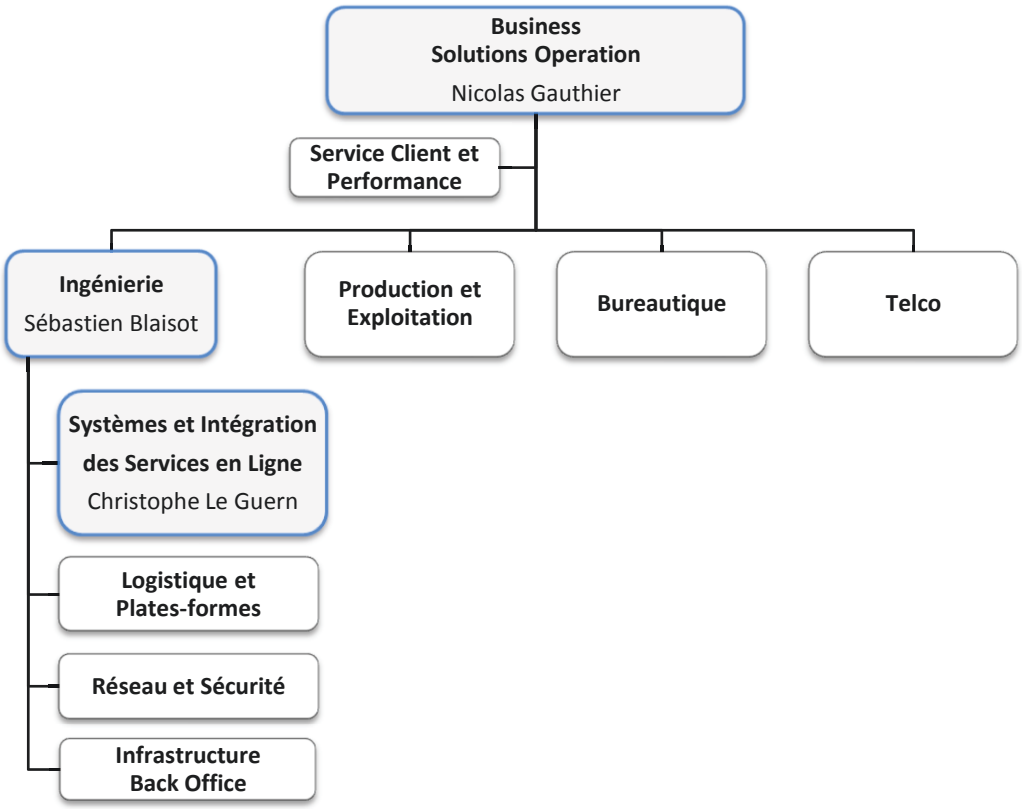

Figure 5 : Organigramme Business Solutions Operation

La **direction Business Solutions Operation** est responsable de la mise à disposition pour PagesJaunes des infrastructures techniques et informatiques. Ces ressources sont de types infrastructure d'hébergement, poste de travail à destination du personnel de PagesJaunes et notamment de la force commerciale, téléphonie, etc.

Environ la moitié des ressources du pôle Business Solutions Opération est basée à Rennes et gère plus spécifiquement l'hébergement des applications dans les datacenters PagesJaunes de Rennes. En particulier, le **département Ingénierie**, dispose de fortes compétences dans tous les domaines relatifs à l'hébergement : gestion de datacenters, réseaux, systèmes, conception d'architectures web, etc.

J'exerce mon activité d'ingénieur systèmes dans l'**équipe Systèmes et Intégration des Services en Ligne**. C'est également dans cette équipe que j'ai réalisé ce mémoire de fin

d'études. L'équipe **Systèmes et Intégration des Services en Ligne** assure la gestion technique ainsi que l'expertise technique sur les systèmes et middlewares pour les différentes entités. Elle architecture et administre les équipements de stockage (SAN, NAS) et les services transverses. Elle préconise les bonnes pratiques et propose des solutions d'architecture pour répondre aux problématiques des différentes équipes. Enfin elle participe à la création des architectures et à l'intégration des projets d'hébergement du groupe Solocal.

2.2. Présentation de l'offre d'hébergement

La direction Solutions Opération propose une offre d'hébergement informatique à destination des principales maîtrises d'œuvre du groupe. Les deux principaux clients de cette offre sont actuellement :

- Le pôle Business Solutions pour l'hébergement des applications internes, des applications clients et des sites Internet pour professionnels.
- Le pôle Média, entité de PagesJaunes, chargé de développer le portail pagesjaunes.fr.

2.2.1. Objectifs

L'offre d'hébergement de BSO s'articule autour des principaux objectifs suivants :

- Un hébergement sécurisé à forte valeur ajoutée.
- Une disponibilité 24h/24, 7j/7. En 2013, le taux de disponibilité sur le portail pagesjaunes.fr était de 99.98 %.
- Tolérance aux fortes charges.
- Une expertise dans les domaines des technologies Internet.

2.2.2. Les moyens techniques

Avec deux datacenters répartis sur des zones géographiques différentes, BSO apporte toutes les garanties pour offrir un service de haute qualité. Les deux datacenters proposent un total de 450m² d'espace d'hébergement, adoptant les dernières normes en matière d'énergie, de froid et de sécurité.

Les deux datacenters sont reliés entre eux par deux liens 10Gb/s et disposent chacun d'une connexion Internet de 10Gb/s.

2.2.3. Les offres de services

Suivant la structure et la nature des projets, BSO propose une palette de 5 offres allant de l'hébergement sec d'équipements, que l'équipe projet exploite elle même, aux plateformes entièrement infogérées par BSO.

- **Offre 0** : des espaces énergisés et refroidis, rackés dans les datacenters PagesJaunes.
- **Offre 1** : des serveurs connectés au réseau, hébergés dans les datacenters PagesJaunes.
- **Offre 2** : des serveurs clés en main avec les principaux logiciels installés.
- **Offre 3 et 4** : des serveurs hébergés avec options de supervision, de sauvegarde et d'administration.
- **Offre 5** : des plateformes entièrement architecturées, intégrées et administrées par BSO.

2.2.4. Les processus

Le processus principal d'hébergement de BSO comporte diverses activités opérationnelles, de support, de pilotage et de mesures qui s'articulent tout au long de la durée de vie d'une application, de l'avant projet jusqu'à la phase d'exploitation en production.

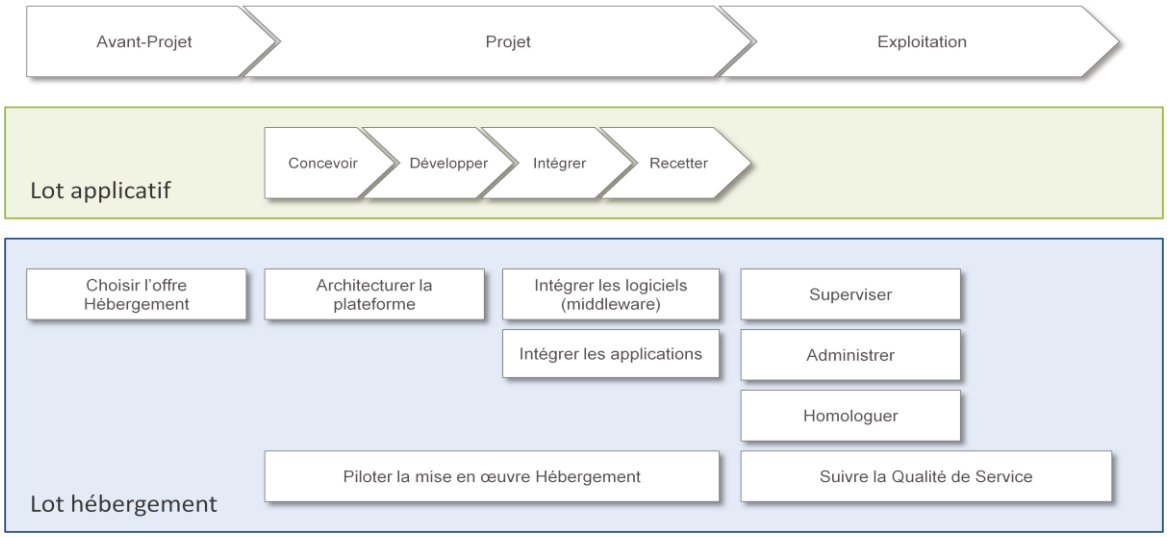

Figure 6 : Le processus hébergement

3. L’existant

3.1. La virtualisation de serveurs chez PagesJaunes

3.1.1. La virtualisation

Parmi les 3000 serveurs hébergés aujourd'hui dans les datacenters de PagesJaunes à Rennes, environ 2000 sont des serveurs dit « virtuels ». La virtualisation consiste à faire fonctionner plusieurs systèmes d'exploitation, éventuellement hétérogènes, de façon étanche sur une même machine physique.

Le serveur hôte, qui va accueillir les serveurs invités, est couramment appelé hyperviseur. Les serveurs invités sont souvent dénommées « serveurs virtuels », « machines virtuelles » ou par l'abréviation VM (abréviation anglaise de Virtual Machine).

L'hyperviseur accueille une couche logicielle, qui vient s’intercaler entre le matériel et les systèmes d’exploitation des machines virtuelles, et dont le rôle consiste à exposer aux machines hébergées un matériel virtuel (processeur, mémoire, contrôleurs d’entrée/sortie, etc.).

La technique de la virtualisation offre de nombreux atouts :

- Réduction des coûts d'hébergement, par la diminution du nombre de machines physiques (électricité, climatisation, espace, câblage, maintenance matérielle)
- Rapidité de déploiement, car l'étape d'installation physique d'un serveur (installation, câblage) n'est pas nécessaire pour un serveur virtuel
- Flexibilité, en augmentant ou diminuant dynamiquement les ressources allouées aux machines virtuelles au gré des besoins.
- Augmentation de la sécurité, par cloisonnement en séparant les différentes applications hébergées sur des machines virtuelles différentes.

3.1.2. La technologie de virtualisation au sein de PagesJaunes

La virtualisation est utilisée chez PagesJaunes depuis 2008 et fait massivement usage de technologies open-source sur des systèmes d'exploitation Linux. Entre 2008 et 2011, les hyperviseurs de PagesJaunes utilisaient la technologie XEN, intégrée à la distribution RHEL de RedHat. Depuis 2011, l'essentiel des hyperviseurs utilise la technologie de virtualisation KVM sur des serveurs utilisant la distribution Ubuntu.

3.1.3. Les modes d'utilisation du stockage dans le cadre de la virtualisation

PagesJaunes utilise la virtualisation, dans son implémentation la plus simple, qui consiste à utiliser directement les ressources matérielles de l'hyperviseur. Ces ressources sont : de la puissance de calcul au travers du partage de temps CPU, de la mémoire RAM, des entrées sorties réseau, des entrées/sorties disque pour le stockage.

Un exemple typique d'hyperviseur chez PagesJaunes utilise, au niveau du stockage, huit disques de 300Go chacun. Ces disques utilisent l'interface SAS 2.0 (remplaçante d'SCSI) et ont une vitesse de rotation de 15000 tours par minute. L'ensemble de ces disques est agrégé dans une même grappe RAID (de l'anglais Redundant Array of Independent (or inexpensive) Disks), à l'aide d'un contrôleur dédié en mode RAID 5 qui permet d'améliorer la tolérance aux pannes et les performances. Afin d'améliorer encore les performances, le contrôleur RAID intègre un cache utilisé pour les écritures. Ce cache permet de gommer la latence inhérente à l'utilisation de disques durs à plateaux, qui sont par nature des périphériques relativement lents.

Une implémentation plus complexe de la virtualisation consiste à utiliser des baies de stockage SAN (de l'anglais Storage Area Network) pour le stockage. Les baies de stockages sont des matériels concentrant de nombreux disques durs dont les ressources de stockage peuvent être mises à disposition de serveurs distants, à travers un réseau le plus souvent basé sur des fibres optiques. Cette solution, bien que performante, est relativement peu utilisée chez PagesJaunes pour l'hébergement Internet, principalement pour des raisons de coûts. En effet, l'utilisation de baies de stockages implique un coût important en termes d'acquisition de matériels, de licences, de support et d'exploitation.

3.1.4. Quelques chiffres

Comme dans beaucoup d'entreprises du domaine informatique, PagesJaunes a profité de l'extraordinaire opportunité que constitue la virtualisation pour optimiser les ressources informatiques. Aujourd'hui, l'essentiel des serveurs installés est virtuel et la tendance pour 2014 est confirmée à la hausse. Un objectif important de BSO est de stabiliser le nombre de machines physiques en 2014.

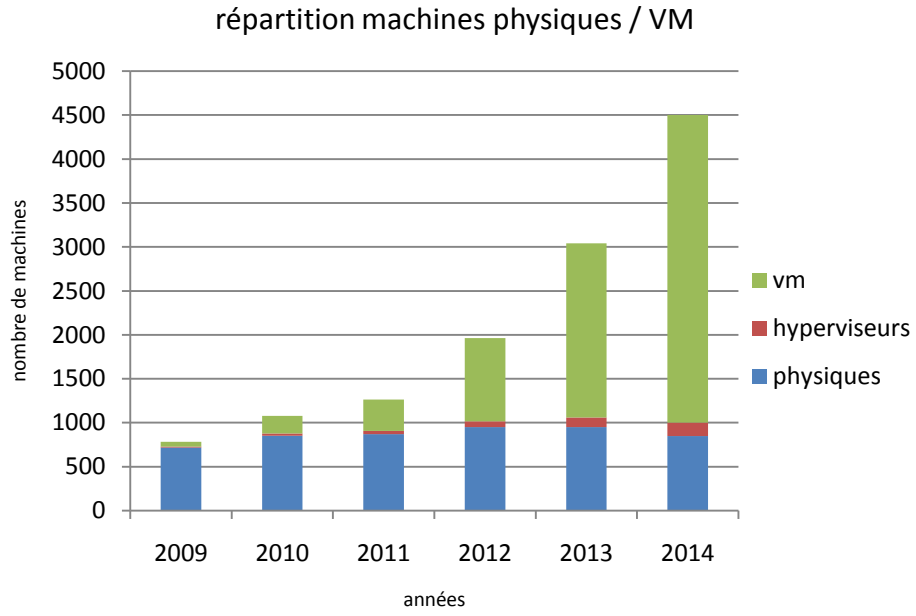

Figure 7 : Répartition machines physiques, machines virtuelles avec estimations pour 2014

La recherche d'optimisation des coûts ainsi que l'amélioration croissante des performances des serveurs ont permis de concentrer toujours plus de machines virtuelles par hyperviseur. Dans le cas de PagesJaunes, compte tenu de notre technique de stockage local, ceci représente un risque fort en cas de perte de machine car une machine physique concentrait en 2013 vingt machines virtuelles. Ce risque est néanmoins fortement atténué par la construction d'architectures web distribuées ne s'appuyant pas sur un seul serveur physique mais, au contraire, profitant des ressources de plusieurs machines virtuelles, réparties sur plusieurs hyperviseurs.

Au niveau de la volumétrie, plus de 100To de stockage étaient consacrés aux hyperviseurs en 2013. En 2014, nous avons choisi de faire évoluer nos offres de virtualisation afin de limiter

le recours aux machines physiques, en débridant la volumétrie des machines virtuelles pour la faire passer de 100Go maximum à 200Go maximum. La demande de stockage liée à la virtualisation devrait donc croître de manière très importante.

Globalement, les datacenters de PagesJaunes accueillent environ 1.3Po (péta octets) de capacité de stockage, soit 1300To.

3.2. La gestion du stockage des machines virtuelles

3.2.1. Attribution des ressources

L'installation des serveurs virtuels est réalisée par l'équipe d'Exploitation, à la demande des équipes d'Intégration. Le déploiement d'un serveur virtuel est réalisé grâce à des scripts d'installation développés par l'équipe Systèmes. Les scripts d'installation sont chargés d'allouer un espace disque aux serveurs virtuels, de configurer le nombre de processeurs virtuels et la quantité de RAM, puis d'installer le système d'exploitation invité. Enfin, a lieu une étape de post-installation qui comprend diverses tâches comme la configuration du réseau, la configuration des accès ...

Avant de lancer ce script de déploiement, il est nécessaire d'identifier un hyperviseur disposant des ressources disponibles. Les paramètres de RAM, de CPU et d'espace disque sont particulièrement importants (au niveau réseau, les ressources sont souvent suffisantes grâce aux interfaces 1Gb). Afin d'identifier un hyperviseur susceptible d'héberger la nouvelle machine virtuelle, l'exploitant consulte l'inventaire des hyperviseurs et tente d'identifier un potentiel candidat.

Cette opération fastidieuse met en exergue différentes problématiques :

- Faire coïncider les besoins en ressources CPU libres, ressources de RAM libres et espace disque libre n'est pas toujours une tâche facile.
- La grande variété de machines virtuelles en termes d'utilisation implique une forte disparité d'utilisations des hyperviseurs. Certains étant, par exemple, surchargés en utilisation RAM par rapport à l'utilisation disque, ou inversement, il n'y a plus d'espace disque de disponible alors qu'il reste beaucoup de RAM.

- Conséquence de ce deuxième problème, les ressources inutilisées en termes d'espace disque, de CPU et de RAM sont des ressources perdues et représentent donc un coût.

3.2.2. Gestion des performances

Au-delà de l'aspect purement quantitatif évoqué dans le point précédent, les ressources doivent être gérées avec soin au niveau des performances. Ceci est particulièrement important au niveau du stockage. En effet, tous les serveurs virtuels partagent une même ressource disque au sein d'un hyperviseur. Une machine virtuelle en surconsommation d'entrées/sorties disque (autrement appelé IO de l'anglais Input/Output) peut donc affecter les performances de toutes les autres machines virtuelles du même hyperviseur.

Cet aspect performance est particulièrement difficile à surveiller. Il faut tout d'abord disposer d'outils pour obtenir les bonnes mesures en temps réel de l'activité des disques. Ensuite, il faut être capable de déterminer si un usage disque est normal, anormal, impactant pour les autres serveurs ou pas.

Afin d'avoir une estimation de la consommation d'entrées/sorties disque, un agent est installé sur chaque serveur. Cet agent, nommé « collectd[3] », recueille diverses informations sur le serveur telles que les informations CPU, RAM, utilisation disque, mais aussi débit disque, latence disque... Ces données sont ensuite envoyées à un serveur central qui les collecte et les enregistre dans des bases RRD. De ces bases sont générés des graphiques qui permettent de déterminer en quasi temps réel (l'intervalle entre chaque mesure est d'une minute) la consommation d'IO disque de chaque serveur et l'évolution de cette consommation au cours du temps.

L'équipe d'exploitation a la possibilité de consulter ces graphiques et de prévenir le service compétent en cas d'anomalie. Ce système a l'avantage d'exister, mais les résultats sont délicats à interpréter et surtout la surveillance et l'administration de la performance des disques sont une vraie épreuve de micro-management dans un environnement comportant plusieurs milliers de serveurs.

[3] http://collectd.org

3.2.3. Gestion des pannes matérielles

Les pannes matérielles peuvent se manifester à plusieurs niveaux sur un hyperviseur. Parmi les pannes les plus couramment rencontrées sur nos hyperviseurs, on peut citer :

- carte mère : nécessite un démontage
- mémoire : nécessite un démontage
- carte réseau : nécessite un démontage
- alimentation : changement à chaud
- disque dur : changement à chaud

Comme montré dans cette liste, certains équipements peuvent être remplacés « à chaud », c'est-à-dire sans avoir besoin d'éteindre électriquement l'hyperviseur. Dans d'autre cas, une coupure électrique sur le serveur est nécessaire, ce qui peut engendrer une interruption de service des services hébergés s'ils ne sont pas redondés.

Il est important de noter que dans le cas des pannes de disques durs, étant donné que ceux-ci sont montés dans des grappes RAID 5, la panne d'un disque est tolérée. A contrario, une panne de plusieurs disques entraine non seulement une interruption du service, mais aussi une perte irrémédiable des données, ce qui constitue un risque majeur.

Dans ces conditions, et compte tenu du nombre de machines virtuelles hébergées sur chaque hyperviseur, il est particulièrement important de disposer d'un système de détection de panne afin d'éviter de rendre indisponibles plusieurs dizaines de machines virtuelles.

Cette détection est actuellement assurée par un système de supervision appelé « Nagios[4] ». Pour superviser les pannes de disque, un agent est installé sur chaque hyperviseur et remonte une alerte en cas d'incident sur la carte RAID.

3.2.4. Migration à chaud des machines virtuelles

Lorsqu'une panne est détectée ou lorsqu'un problème de performance affecte le fonctionnement global d'un hyperviseur, il peut être nécessaire de migrer tout ou une partie des machines virtuelles sur un autre hyperviseur.

[4] http://www.nagios.org/

Afin de réaliser cette opération, l'exploitant peut réaliser une opération de migration « à chaud ». La migration à chaud permet, au contraire de l'opération de migration « à froid », de migrer une ou plusieurs machines virtuelles d'un hyperviseur à un autre de manière dynamique, sans avoir besoin de réaliser une coupure de service, ce qui rend l'opération totalement transparente.

La migration de machine virtuelle à chaud comprend les étapes successives suivantes :

1) réservation des ressources sur l'hyperviseur de destination
2) copie itérative des blocs de données disque
3) copie itérative des pages mémoires
4) copies des registres processeur et de l'état des périphériques
5) activation de la machine virtuelle sur l'hôte de destination
6) mise à jour des informations réseau

La migration à chaud existe depuis longtemps sur les systèmes de virtualisation utilisant un stockage centralisé de type SAN. Dans ce cas, la migration des données disque n'est pas nécessaire (étape numéro 2).

La fonctionnalité de migration à chaud avec déplacement des blocs disques (LiveBlocMigration) d'un hyperviseur à l'autre a été intégrée au système KVM en 2011 et est donc utilisée par PagesJaunes depuis cette date.

Le gain en flexibilité est évident, mais cette solution souffre d'un inconvénient majeur : le temps de migration. En effet, dans le cas de la migration par bloc, il est nécessaire de copier l'intégralité des données disque d'un hyperviseur sur un second hyperviseur en passant par un lien réseau d'un débit de 1Gb/s. Cette opération peut prendre plus de 4 heures 30 sur un hyperviseur de 2To.

3.3. Les autres modes de stockage chez PagesJaunes

3.3.1. Le stockage NAS

En dehors du stockage propre aux machines virtuelles, l'hébergement d'applications Internet nécessite souvent de partager des données entre plusieurs serveurs ou de déporter le stockage sur un système externe. Le stockage NAS (de l'anglais Network Attached Storage) fournit à un groupe de machines consommatrices, un espace de stockage sécurisé et accessible à travers un réseau IP.

Différents protocoles permettent aux NAS de communiquer avec les machines clientes. Le protocole NFS (Network File System) permet de partager des données principalement entre systèmes UNIX et est donc massivement utilisé chez PagesJaunes, en raison de la forte prédominance de machines sous système d'exploitation Linux.

Le protocole CIFS (Common Internet File System), permettant le partage de fichiers entre clients Windows est lui aussi utilisé, mais de manière plus anecdotique.

Le stockage NAS est actuellement assuré chez PagesJaunes par deux baies de stockage de marque NetApp acquises en 2008, avec des licences NAS et SAN.

Ces baies, en fin de vie, ne sont quasiment plus utilisées qu'en tant que baies NAS afin de réaliser des partages NFS. La capacité totale de ces baies et leur pourcentage d'utilisation ont fortement variés au cours du temps et des demandes. Aujourd'hui, l'espace de stockage utilisé est de seulement 31To pour une capacité totale de 68To.

Le coût de maintenance de ces baies au regard de leur utilisation se justifie de moins en moins, d'autant plus que l'espace utilisé en tant que partages NFS tant à se stabiliser au profit de l'utilisation plus courante de stockage objet.

3.3.2. Le stockage objet

Contrairement aux systèmes de stockage classiques comme les NAS, les systèmes de stockage objet organisent leur structure de fichiers dans un espace d'adressage plat. L'organisation des fichiers se fait à travers un système de métadonnées associées à chaque fichier qui permet de réaliser des opérations complexes comme des recherches.

L'accès aux systèmes de stockage objet se fait au travers d'une API de type REST qui s'appuie sur des commandes HTTP. Les commandes sont de type GET, PUT et DELETE pour lire, écrire et effacer des objets.

Le stockage lui-même s'effectue sur un ensemble de nœuds distribués qui assurent le stockage et la protection des données (chaque objet est écrit plusieurs fois sur des nœuds différents, ce qui permet d'assurer que les objets seront toujours accessibles même en cas de défaillance d'un nœud).

Le stockage objet a été introduit chez PagesJaunes en 2012, au cours d'un projet dans lequel j'ai été activement impliqué. L'objectif était de faire face à deux problématiques :

- Le stockage NAS sur les baies NetApp n'offrait pas des performances suffisantes dans certains cas de figure, par exemple sous de très fortes charges avec des arborescences de fichiers complexes sollicitant de manière excessive le cache des baies.
- La volumétrie des données était déjà en forte croissance et nécessitait donc de mettre en place un système capable de s'adapter rapidement à la demande.

Ce genre de système présente néanmoins deux inconvénients :

- Un développement spécifique au niveau des clients est nécessaire pour utiliser l'espace de stockage. Le développement doit s'appuyer sur l'API REST.
- L'espace de stockage ne propose pas, par nature, de système de fichiers à la manière d'un NAS ou des blocs de données à la manière d'un SAN. Cela implique qu'il n'est pas possible de l'utiliser pour accueillir des machines virtuelles par exemple.

Le système de stockage objet existant chez PagesJaunes s'appuie sur la solution open source Openstack Swift[5]. Après un démarrage plutôt timide en termes d'utilisation, le système

[5] http://www.openstack.org/software/openstack-storage/

commence à être activement intégré aux nouveaux projets depuis 2013, notamment pour l'hébergement de fichiers statiques sur les sites à fort trafic.

4. Les besoins

4.1. Recueil des besoins

4.1.1. Les parties prenantes

Les besoins étant souvent différents en fonction des personnes qui les émettent, il est important de comprendre quelles sont les principales parties prenantes dans ce projet :

Le donneur d'ordre :
S'agissant d'un composant d'infrastructure, le nouveau système de stockage fera parti des composants d'hébergement proposés par l'équipe d'Ingénierie à destination des clients internes de Solocal. Le rôle du donneur d'ordre est donc assuré par Sébastien Blaisot, responsable de l'équipe Ingénierie.

L'équipe projet :
L'équipe projet, dans laquelle j'exerce la fonction de chef de projet, est responsable de l'étude et la mise en œuvre de la solution. Elle dispose de ressources dans les différents corps de métiers nécessaires à la bonne exécution de ces taches.

L'équipe Systèmes et Intégration :
L'équipe Systèmes et Intégration intervient en tant que support niveau 3 sur les composants d'infrastructure, notamment sur les systèmes de stockage SAN, NAS et Objets. Elle intervient également pour définir les architectures techniques sur certains projets de Solocal et peut donc être amenée à proposer l'utilisation de la nouvelle plateforme de stockage.

L'équipe Intégration Pôle Media et l'équipe de développement :
Ces équipes sont directement en charge du développement et de l'intégration des composants du portail PagesJaunes. En cela, ils sont de très gros consommateurs de serveurs et d'espace de stockage pour tous les usages : développement, intégration continue, recette, tests en charge etc.

L'équipe Exploitation :
L'équipe d'Exploitation assure les tâches d'administration courante et de supervision sur les plateformes. Elle est garante du bon fonctionnement et de la disponibilité des composants d'infrastructure et des services hébergés en production.

4.1.2. Méthode

Le recueil des besoins consiste à identifier, affiner et prioriser les besoins et attentes formulés par les utilisateurs. Les besoins ont été recueillis au cours de réunions et d'entretiens que j'ai menés auprès des différentes parties prenantes durant la phase d'étude préalable du projet.

L'objectif de ces réunions était d'étudier, avec chaque interlocuteur, la vision perçue du stockage actuel et d'identifier les comportements, usages et habitudes vis-à-vis de ce stockage. De ces entretiens sont ressorties de nombreuses doléances, très différentes suivant les groupes d'utilisateurs.

Après analyse des besoins exprimés, l'équipe projet a formalisé les demandes sous formes d'exigences et les a classifiées et priorisées. Dans ce projet, bon nombre d'exigences sont des exigences dites « non-fonctionnelles ». Afin de les classifier correctement, j'ai utilisé le système FURPS développé par Robert Grady chez Hewlett-Packard.

- **Functionality** (Fonctionnalité))
- **Usability** (Utilisabilité)
- **Reliability** (Fiabilité)
- **Performance** (Performance/Efficience)
- **Supportability** (Maintenabilité)

4.1.3. Périmètre opérationnel

Après concertation avec le donneur d'ordre, il a été acté que le périmètre d'utilisation de la nouvelle plateforme sera limité à l'hébergement de services orientés Internet. Toutes les applications de type back-office sont exclues du périmètre opérationnel de la solution.

4.2. Objectifs stratégiques

Ce projet vise à proposer une nouvelle plateforme de stockage s'articulant autour de deux objectifs principaux :

Adaptabilité et réactivité :
Dans un contexte concurrentiel tendu, il est primordial d'innover et de lancer le plus rapidement possible ses offres sur le marché. Le système de stockage doit permettre une mise à disposition rapide et « à la demande » de ressources de stockage.

Coût :
Comme dans la plupart des services informatiques, les budgets ne croissent pas au même rythme que l'évolution des demandes. Le stockage de données fait pourtant partie des équipements les plus coûteux des services informatiques. Ces coûts sont caractérisés par des coûts matériels, de licences, de maintenance et de formation. Il est donc nécessaire d'apporter par l'innovation une solution permettant de limiter le budget alloué au stockage.

4.3. Description des exigences

4.3.1. Exigences fonctionnelles

Faciliter la migration à chaud des VM :
Les opérations de maintenance sur les hyperviseurs sont actuellement longues et difficiles à réaliser, car l'intégralité des données disque de machines virtuelles doit transiter sur le réseau pour être recopiée sur un nouvel hyperviseur. Il est donc nécessaire d'identifier un système de stockage permettant de partager les données entre les hyperviseurs de manière à réaliser des opérations de migration à chaud rapides et efficaces.

Permettre la création de snapshots :
Les processus de développement, d'intégration logicielle et de mise en production peuvent être grandement facilités par l'utilisation de snapshots. Les snapshots sont des copies instantanées de données permettant un retour arrière très rapide en cas de problème. Les snapshots permettent également de créer un système de versions entre chaque modification. Ils peuvent également faciliter les sauvegardes.

Permettre de créer des volumes de très grande taille :

Un problème fréquemment rencontré par certaines équipes est la difficulté d'obtenir un espace de stockage de très grande taille. Si le serveur est virtuel, nous limitons la capacité disque à 200Go afin de limiter la fragmentation des hyperviseurs. Pour tout espace supérieur, un serveur physique est recommandé. Le nouveau système doit permettre l'utilisation de volumes ou de machines virtuelles de plusieurs To sans que cela ne pose de problème d'administration.

4.3.2. Exigences non-fonctionnelles

Extensibilité :

Afin de répondre de la manière la plus réactive possible aux besoins métiers, le système doit pouvoir être étendu à volonté en termes de capacité et cela jusqu'à des volumes très importants de plusieurs centaines de To. L'ajout (ou la suppression) de capacité doit se dérouler sans interruption de service.

Fiabilité :

Le système ne doit pas souffrir de problème de fiabilité pouvant engendrer des pertes ou corruptions de données. Pour cela, le système doit être capable de détecter les pannes, de vérifier la cohérence et l'intégrité des données, mais aussi de mettre en œuvre des mécanismes de réplication de données permettant de pallier les problèmes de fiabilité du matériel.

Facilité d'administration et d'utilisation :

Le système doit permettre une gestion centralisée du stockage, être facile à administrer et à utiliser. Pour cela, il doit disposer d'une interface d'administration permettant de vérifier son état. Il doit également disposer d'une API permettant aux administrateurs d'interagir facilement avec ses principales fonctionnalités à partir de scripts ou logiciels. Le système doit être utilisable par ces interfaces par un administrateur après une formation de quelques heures.

Performance :

Le système doit être approprié à l'utilisation induite par l'hébergement d'applications Internet fortement sollicitées. Le système doit également permettre à l'administrateur d'adapter ses performances à deux niveaux :

- Si la quantité d'applications hébergées augmente, le système doit permettre d'accroitre les performances globales en proportion.

- Si certaines applications nécessitent plus de performances que d'autres, le système doit permettre de les placer sur un espace dédié à la performance. A contrario, les applications ne nécessitant pas de performance doivent être placées sur des espaces à moindre performance.

Le système doit disposer d'indicateurs permettant de surveiller les performances et la volumétrie en temps réel.

Disponibilité :
Le système doit être disponible à 99,99%. Afin de respecter ce taux, le système doit être tolérant aux pannes et ne pas disposer de point de faiblesse unique (SPOF de l'anglais Single Point Of Failure). Le système doit également permettre de réaliser les mises à jours logicielles et matérielles de manière transparente.

Sécurité des données :
Le système doit éventuellement permettre une reprise d'activité sur un deuxième datacenter en cas d'incident majeur, dans le cadre de la mise en œuvre d'un PRA (Plan de Reprise d'Activité).

4.4. Description des cas d'utilisation

4.4.1. L'accès en mode bloc pour la virtualisation

Afin d'étendre les fonctionnalités du système de virtualisation de PagesJaunes et de pallier ces limites, il est nécessaire de quitter le mode de stockage dit « en silo », où chaque ressource de stockage appartient uniquement à un seul hyperviseur, pour adopter un système où les ressources de stockage peuvent être partagées.

Ce mode de ressources partagées va permettre une plus grande flexibilité dans l'attribution des ressources, une gestion centralisée de l'administration du stockage et une meilleure prise en compte des besoins fonctionnels (migration à chaud, snapshots, clones, etc.).

Habituellement, ce besoin est satisfait par l'utilisation d'un SAN ou d'un système permettant de présenter les données aux hyperviseurs sous forme de bloc. L'architecture peut être représentée sous la forme suivante :

- Les nœuds de stockage contiennent les données brutes et permettent, par une intelligence de stockage, d'assurer la sécurité de ces données et d'apporter les fonctionnalités étendues.
- Les nœuds clients (hyperviseurs), grâce à un connecteur spécifique, présentent des volumes de données distants aux machines virtuelles en leur donnant l'illusion que ce stockage est local.
- Le réseau de données est responsable de la liaison entre les hyperviseurs et les nœuds de stockage. Il peut s'agir d'un réseau fibré (fibre optique), ou d'un réseau cuivré. Pour des raisons de coût (achat de cartes additionnelles pour les serveurs et de switchs fibre pour les interconnexions), nous privilégierons une solution cuivrée.

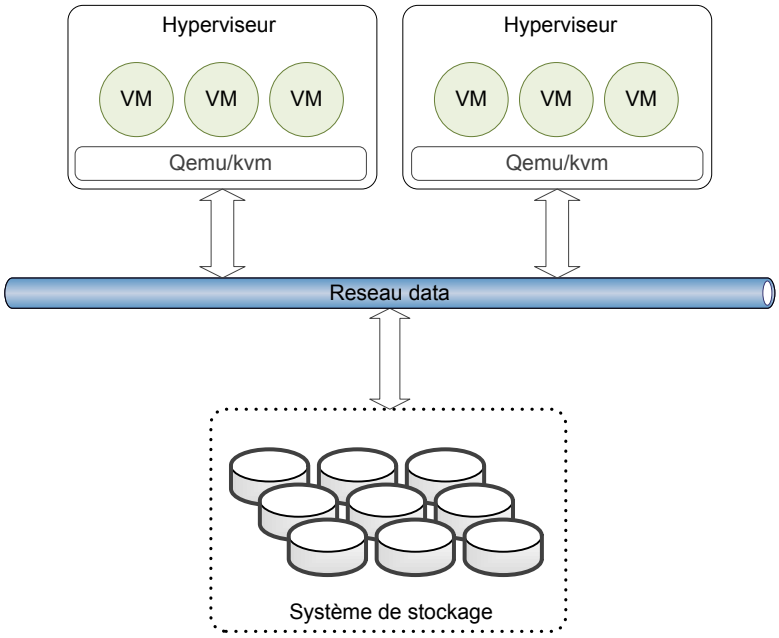

Figure 8 : Stockage en mode bloc pour la virtualisation

4.4.2. L'accès en mode bloc pour les solutions de Cloud Computing

La virtualisation a rendu de nombreux services en étendant considérablement le nombre de serveurs hébergés grâce à une optimisation des ressources matérielles. Cependant, cette évolution majeure ne suffit plus aujourd'hui à satisfaire les besoins de flexibilité de nos utilisateurs (développeurs, intégrateurs).

La tendance actuelle dans le monde de l'hébergement est de proposer des solutions permettant une mise à disposition instantanée de ressources, « à la demande ». Cette tendance se traduit

par l'utilisation de services Cloud, qui rend possible à un utilisateur de déployer une ressource serveur en faisant une abstraction complète de l'infrastructure sur laquelle cette ressource repose. Dans les faits, ces solutions de Cloud Computing font un usage intensif des techniques de virtualisation, mais en y ajoutant une couche de gestion des ressources et une couche d'administration basée sur des API.

Afin de satisfaire ce besoin de flexibilité, un nouveau projet est mené depuis le mois de février au sein de l'équipe Systèmes, visant à doter PagesJaunes d'une solution de Cloud Computing sur le mode IAAS (Infrastructure As A Service).

Ce projet, mené en parallèle du projet de stockage, a pour premier objectif de fournir un Cloud privé de petite taille aux équipes d'intégration et de développement du pôle Média. Par la suite, s'il est satisfaisant, ce système pourrait être étendu et éventuellement remplacer les systèmes de virtualisation existants.

La solution de stockage doit donc s'intégrer parfaitement avec la solution Cloud choisie par l'équipe projet Cloud. L'architecture fonctionnelle peut être décrite de la manière suivante : un utilisateur doit pouvoir déployer une machine virtuelle en interagissant avec une API sur un contrôleur Cloud. Ce contrôleur est chargé de déployer une machine virtuelle sur un nœud de calcul disponible, de réserver un espace de stockage et de mettre à disposition cet espace sur la machine virtuelle, en mode bloc, sur les mêmes principes que pour le point précédent.

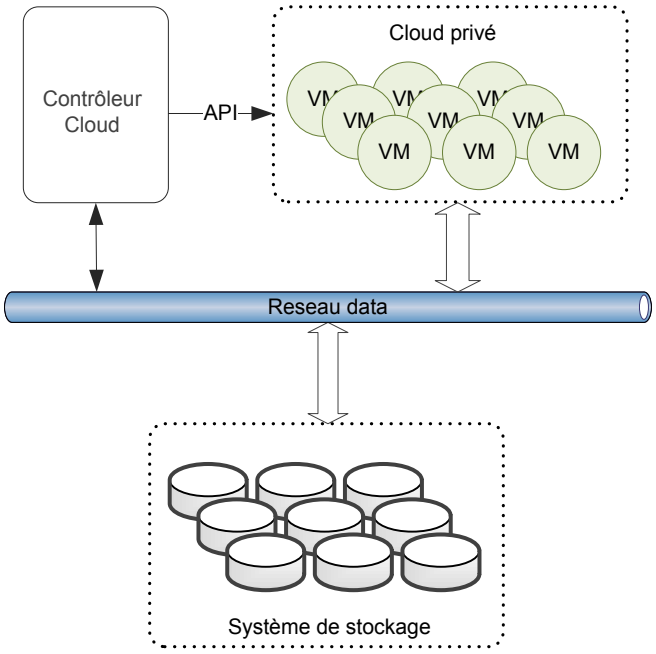

Figure 9 : Stockage en mode bloc pour le Cloud Computing

4.4.3. L'accès en mode fichier

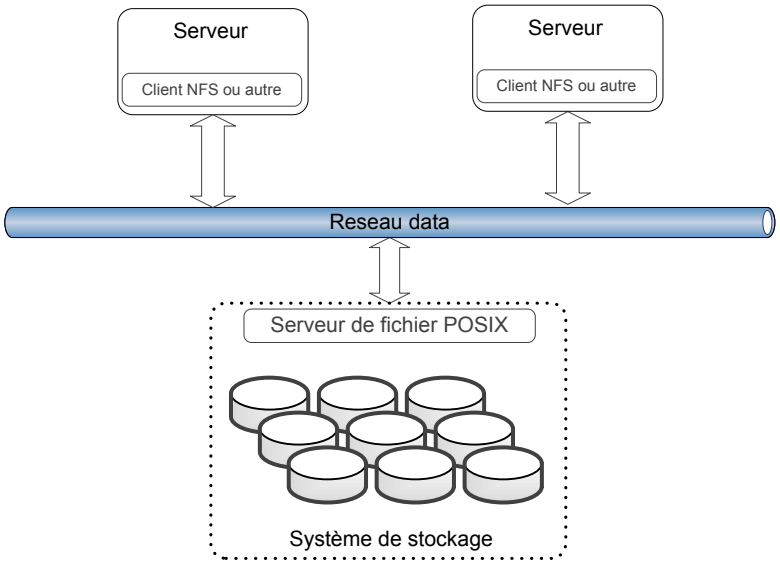

Figure 10 : Stockage en mode fichier

Le dernier mode d'utilisation prévu pour le nouveau système de stockage est le mode « serveur de fichiers ». Ce mode consiste à proposer à un ensemble de serveurs client un espace de stockage étendu et sécurisé, sous forme de système de fichiers distant accessible par un réseau IP. Ce système de fichiers peut permettre des accès concurrents à plusieurs serveurs clients.

Typiquement, il s'agit de proposer une alternative économique aux baies de stockage NAS NetApp de PagesJaunes. Pour cela, le système de stockage doit proposer des volumes de stockage de plusieurs Téra octets aux serveurs clients, en utilisant à minima le protocole NFS V3 et, si possible, le protocole CIFS en sus.

5. L’étude générale

5.1. Principes généraux

5.1.1. Le stockage distribué

Un système de stockage distribué fournit un service de stockage sur des objets pouvant fonctionner au niveau bloc, au niveau fichier ou de manière hybride. Ce système de stockage utilise des ressources de stockage fédérées, dispersées sur un réseau, mais étant perçues par les clients comme un espace de stockage local. Un système de stockage distribué dispose des propriétés suivantes :

Transparence : les utilisateurs accèdent au stockage du système et l'utilisent à la manière d'un stockage local. Les pannes inhérentes à la nature distribuée du système ne doivent pas être perçues par les utilisateurs et, en général, toute la complexité du système ne doit pas être visible des clients.

Tolérance aux pannes : Le système ne doit pas être menacé par une indisponibilité d'une partie du système. Les fautes peuvent être liées au réseau, aux serveurs, à la corruption de données ou à des accès concurrents très importants provoqués par de multiples clients.

Elasticité : Le système doit pouvoir être étendu en termes de capacité, de manière dynamique, jusqu'à plusieurs centaines de nœuds. Par opposition aux architectures de stockage traditionnelles, qui sont limitées à un nombre fini de contrôleurs (par exemple 2), les systèmes distribués fournissent de multiples contrôleurs permettant d'étendre la capacité de calcul et de limiter les goulots d'étranglement.

5.1.2. Le stockage logiciel

Le stockage logiciel, autrement appelé SDS pour Software Defined Storage, est un concept récent qui s'inscrit dans la lignée des concepts de datacenters définis par logiciel (SDDC) et

de réseaux définis par logiciel (SDN pour Software Defined Networking). Ces concepts ont été initiés par les géants du web, tels que Google ou Amazon, et visent à mettre en avant la couche logicielle faisant fonctionner l'ensemble des composants de l'infrastructure par rapport à la vision traditionnelle qui ne considère que la partie hardware (matériel).

Ces nouveaux concepts logiciels ont pu émerger grâce aux extraordinaires progrès réalisés dans le domaine matériel, principalement au niveau des processeurs. Les coûteux ASIC, réalisant des tâches ultra spécialisées dans des architectures matérielles propriétaires, tendent à être concurrencés par des processeurs de type Intel basés sur des architectures ouvertes de type x86.

Les nouvelles possibilités offertes par l'utilisation de matériel standard ont notamment ouvert la voie à un bouleversement dans le domaine du stockage en encourageant les développements de logiciels orientés « stockage ». De nombreux acteurs, dont les acteurs historiques du stockage, ont saisi cette opportunité pour proposer de nouvelles solutions où le stockage est entièrement géré par du logiciel sur la base d'architectures matérielles standards. Le logiciel est en charge de la distribution et de la réplication des données sur les nœuds de stockage, de la vérification de l'intégrité de celles-ci, de la fourniture de l'interface entre les blocs de données et les clients. Il fournit aussi des fonctions avancées telles que la déduplication de données ou la création d'instantanés (snapshot), de clones, etc.

Au niveau des services informatiques, les avantages du stockage logiciel (SDS) sont nombreux :

- Utilisation de serveurs standards x86 permettant de diminuer fortement les coûts.
- Plus de liens exclusifs avec un fournisseur de matériel.
- Utilisation d'API ouvertes (exemple amazon S3) permettant de libérer les développements.
- Unification du stockage rendu possible par la séparation de la couche hardware de stockage par une couche de stockage logique.

5.1.3. Le théorème CAP

Avant de poursuivre, il est important de prendre en compte une spécificité importante des systèmes distribués. En 2000, le chercheur en Informatique Eric Brewer a établi un théorème

connu sous le nom de théorème CAP, démontrant qu'il n'était pas possible pour un système distribué de fournir les trois propriétés suivantes simultanément, mais seulement deux.

- Consistency (Consistance) : tous les nœuds servent les mêmes données au même moment.
- Availability (Disponibilité) : toutes les requêtes doivent recevoir une réponse.
- Partition tolerence (Résistance au morcellement) : aucune panne moins importante qu'une coupure totale du réseau ne doit empêcher le système de répondre correctement.

Figure 11 : Illustration du théorème CAP

L'exemple ci-dessous illustre un exemple de système distribué. On peut observer que si les clients peuvent écrire sur les deux nœuds, en cas d'apparition d'une partition, le système peut se trouver dans un état où les données écrites sur le premier nœud sont différentes de celles écrites sur le second, ce qui constitue un problème de consistance des données.

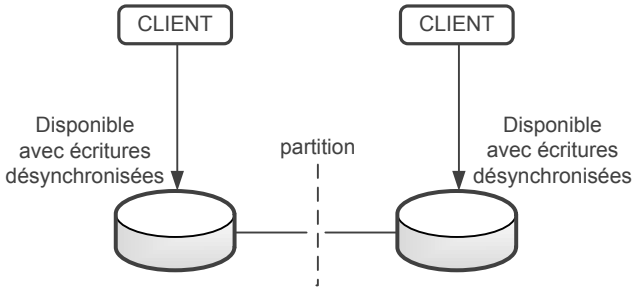

Figure 12 : Exemple d'un système AP

Un système distribué étant par nature sujet à des risques de partition, une interprétation plus moderne du théorème tend à devenir plus courante : *En cas de partition réseau, le système*

peut choisir entre la consistance ou la disponibilité totale du système. Cela amène également à reconsidérer les catégories de systèmes : en réalité il existe, d'une part, des systèmes globalement C+P/C+A gérant simultanément les propriétés de consistance et de partitionnement et, d'autre part, des systèmes A+P.

Certains systèmes privilégieront donc la disponibilité au détriment de la consistance (par exemple le système de base de données Cassandra ou le système de stockage Openstack Swift). Ces systèmes sont qualifiés d' « éventuellement-consistant ».

D'autres systèmes privilégieront la consistance au détriment de la disponibilité d'une partie du système. Ceci peut être réalisé en utilisant un mécanisme de quorum afin de déterminer quelle partition peut continuer à fonctionner et quelle partition doit être inaccessible.

Nous pouvons dès à présent constater que les exigences de « fiabilité » (consistance) et de disponibilité demandées dans l'expression des besoins devront être étudiées avec attention.

5.2. Les solutions potentielles

Dans le monde en pleine ébullition du stockage logiciel, il existe quatre grandes catégories de solutions :

1. Les produits commerciaux « traditionnels » ne comprenant pas de logiciel open source (EMC VNX et VMAX).
2. Les produits issus des dernières innovations, qui se basent sur des logiciels open source, mais ajoutent une part significative de logiciels propriétaires (EMC Isilon, Nexenta, Nexsan, Nutanix, Scale Computing, Parallels Cloud Server).
3. Les produits dérivés de solution open source proposés avec un support (Cloudera et Hortonworks (Hadoop), Inktank (Ceph), Red Hat (GlusterFS, OpenStack), SwiftStack (OpenStack Swift)).
4. Les produits open source que tout un chacun peut télécharger et utiliser en assurant lui-même le support à l'aide des informations fournies par une communauté d'utilisateurs et de développeurs (Ceph, Hadoop Distributed File System, Sheepdog, GlusterFS, OpenStack Object Storage/Swift et OpenStack Bloc Storage).

Le choix d'un type de solution parmi ces quatre catégories peut se faire en prenant en compte deux critères : **le coût** total des solutions et **les compétences** nécessaires à la mise en place et à l'administration de ces solutions. La catégorie 1 propose des solutions entièrement intégrées avec un support logiciel obligatoire opéré par le concepteur ou un partenaire certifié. Les prestations sont souvent proposées à un tarif très important. A l'inverse, la catégorie 4 propose une solution logicielle, qui sur le papier, est gratuite, mais nécessitera de fortes compétences internes pour mettre en œuvre la solution et garantir un niveau de qualité important une fois en production.

PagesJaunes dispose d'une forte expertise dans les technologies open source, en particulier au niveau de la partie hébergement Internet. La quasi-totalité des 3000 serveurs utilise des systèmes d'exploitation open source. Les équipes d'hébergement ont mis en place et administrent, depuis 2012, une solution de stockage objet open source (Openstack Swift) et disposent donc d'une expérience significative dans ce domaine.

Afin de respecter un des objectifs principaux de ce projet, le coût, j'ai limité les recherches de solutions aux catégories 3 et 4. Une première sélection parmi les technologies proposées m'a permis de limiter l'étude à trois produits : Sheepdog, GlusterFS et Ceph car seuls ces trois solutions de stockage disposent des fonctionnalités pouvant répondre aux cas d'utilisation exprimés par le client. En effet, il existe de nombreux systèmes de fichiers distribués, mais seuls ces trois produits permettent d'exposer des périphériques de type bloc aux clients.

Sheepdog[6] est une solution de stockage distribué open source développée par les laboratoires NTT (Nippon Telegraph and Telephone Corporation).

GlusterFS[7] est une solution de stockage basée sur un système de fichiers distribués. Initialement développé par la société Gluster, le produit est, depuis 2011, développé et maintenu par Red Hat suite au rachat de Gluster. GlusterFS est proposé sous forme de logiciel libre et fait office d'incubateur pour les nouvelles fonctionnalités qui sont ensuite incorporées au produit commercial de Red Hat nommé « Red Hat Storage » (RHS).

[6] http://sheepdog.github.io/sheepdog/
[7] http://www.gluster.org/

Ceph[8] est une solution de stockage distribué open source initialement créée par Sage Weil dans le cadre de son doctorat en 2006. Aujourd'hui, la solution est majoritairement développée au sein de la société Inktank, créée par Sage Weil et basée aux Etats-Unis. Inktank propose des prestations de conseil, des formations et du support sur la solution Ceph, directement ou par l'intermédiaire de partenaires (en France : Canonical et eNovance). Le 30 avril 2014, Redhat annonce l'acquisition de la société Inktank, ce qui permettra à Ceph de profiter de la renommée et de la puissance commerciale de Redhat.

Afin de choisir la meilleure solution dans le contexte d'utilisation de PagesJaunes, j'ai choisi d'utiliser une méthode d'évaluation de logiciel open source. Le point suivant détaille la méthode, l'utilisation que j'en ai faite et le choix final.

5.3. Méthodologie

5.3.1. La méthode QSOS

La méthode de Qualification et de Sélection de logiciels Open Source[9] (QSOS) est une méthode d'évaluation de logiciels libres initiée par Atos Origin. Cette méthode prend en compte les aspects fonctionnels et techniques ainsi qu'une analyse des risques spécifiques des logiciels libres.

QSOS repose sur un processus itératif en quatre étapes :

- **Définir** les données de référentiel (types de licences, types de communautés, grilles de couverture fonctionnelle par domaine...)
- **Évaluer** les logiciels selon deux axes principaux : couverture fonctionnelle et maturité du projet
- **Qualifier** le contexte spécifique de l'entreprise, en effectuant une pondération des critères précédents
- **Sélectionner** et comparer les logiciels répondant aux besoins.

[8] http://ceph.com/
[9] http://www.qsos.org/

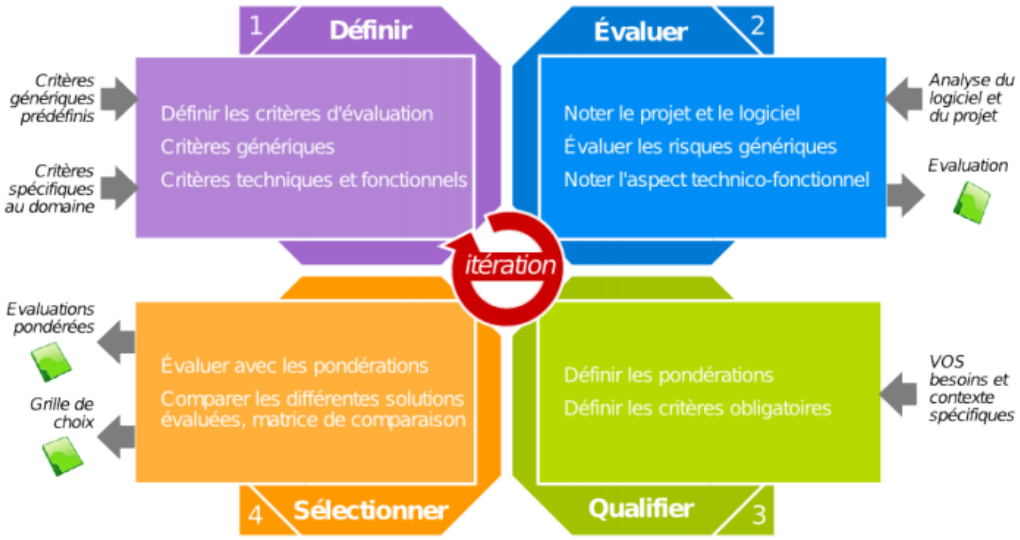

Figure 13 : Processus général de QSOS

5.3.2. Définir

L'objectif de cette étape est de définir les différents critères qui seront réutilisés dans les étapes suivantes du processus général. Doivent être décrits pour chaque projet :

- La maturité
- La couverture fonctionnelle
- Le type de licence

Les **critères de maturité** sont imposés par la méthode QSOS. La figure suivante détaille la hiérarchie des différents axes d'analyse.

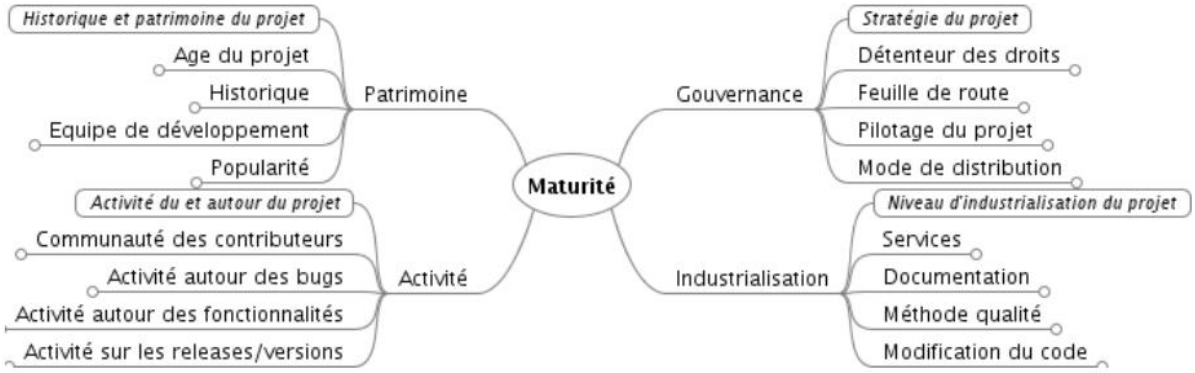

Figure 14 : Les critères de maturité de la méthode QSOS

La **couverture fonctionnelle**, elle, est entièrement à définir pour chaque logiciel. La difficulté, ici, est de choisir des critères suffisamment génériques pour qu'ils puissent être reportés d'une solution à l'autre. J'ai choisi d'utiliser les critères de plus haut niveau suivants :

- Les interfaces de stockage (bloc, fichier, Cloud, objet)
- Les fonctions avancées (snapshot, clones, géo-réplication, etc.)
- L'architecture (architecture CAP, modèle de stockage, modèle de sécurité des données, gestion des partitions, etc.)

Il existe de nombreux **types de licences** dans le monde de l'open source. Il est important, lors de l'utilisation d'un logiciel open source, de bien identifier sa licence et les avantages ou contraintes qui y sont attachés. QSOS propose une catégorisation des types de licences selon les axes suivants :

- propriétarisation : le code dérivé peut-il être rendu propriétaire ou doit-il rester libre ?
- persistance (contamination) : l'utilisation du code du logiciel, à partir d'un autre module, se traduit-il ou non par la nécessité de placer ce module sous la même licence ?

Projet	Licence	Propriétarisation	Persistance
Sheepdog	GPLV2	Non	Oui
GlusterFS	LGPL	Non	Partielle
Ceph	LGPL	Non	Partielle

Tableau 1 : Les types de licences

Une licence inadaptée peut conduire à l'élimination d'un produit si celle-ci rend son utilisation difficile ou impossible, dans le respect du cadre juridique de ladite licence. Dans notre cas d'utilisation, les licences GPLV2 et LGPL ne posent aucune contrainte spécifique pour le déroulement du projet et l'usage du stockage pour PagesJaunes.

5.3.3. Evaluer

Cette étape consiste à noter les logiciels selon les critères définis dans l'étape précédente. Cette étape implique une analyse approfondie des fonctionnalités de chaque logiciel et nécessite donc beaucoup de temps pour être correctement réalisée. Chaque critère est noté de 0 à 2 suivant la règle de notation suivante :

Note	Description
0	Fonctionnalité non couverte
1	Fonctionnalité partiellement couverte
2	Fonctionnalité totalement couverte

Tableau 2 : Les critères de notation QSOS

Les versions évaluées pour chaque logiciel sont les suivantes :

- Sheepdog V0.8.1
- GlusterFS V3.4.2
- Ceph V0.72

5.3.4. Qualifier

Dans cette étape, il s'agit de replacer l'évaluation dans le contexte d'utilisation. Pour cela, chaque critère peut être pondéré avec une note comprise entre 0 et 3.

Les pondérations doivent être appliquées sur les critères de maturité et de couverture fonctionnelle du projet en appliquant les règles suivantes :

Note	Description maturité	Description fonctionnalités
3	Critère critique	Fonctionnalité requise
1	Critère pertinent	Fonctionnalité optionnelle
0	Critère non pertinent	Fonctionnalité non requise

Tableau 3 : Les pondérations sur la maturité et les critères fonctionnels

Les tableaux suivants illustrent les résultats des étapes de définition, évaluation et qualification pour les critères de maturité et les critères fonctionnels selon les 3 types de sous-critères retenus : interfaces de stockage, fonction avancées et architecture.

Maturité :

Définir	Evaluer			Qualifier
Critères de maturité	**Sheepdog**	**GlusterFS**	**Ceph**	**Pondération**
maturité	**0,98**	**1,75**	**2**	
Patrimoine : Historique et patrimoine du projet	0,83	1,50	2	3
Age du projet	2	2	2	1
Historique	2	2	2	1
Equipe de développement	1	2	2	1
Popularité	0	1	2	3
Activité : Activité du et autour du projet	1,00	2	2	3
Communauté des contributeurs	1	2	2	3
Activité autour des bugs	1	2	2	3
Activité autour des fonctionnalités	1	2	2	3
Activité sur les releases/versions	1	2	2	3
Gouvernance : Stratégie du projet	1,25	2	2	3
Détenteur des droits	2	2	2	1
Feuille de route	1	2	2	1
Pilotage du projet	1	2	2	1
Mode de distribution	1	2	2	1
Industrialisation : Niveau d'industrialisation du projet	0,83	1,50	2	3
Offres de services (Support, Formation)	0	2	2	1
Documentation	1	1	2	3
Méthode qualité : Processus et méthode qualité	1	2	2	1
Modification du code	1	2	2	1

Tableau 4 : Les critères de maturité

Critères fonctionnels :

Définir	Evaluer			Qualifier
Critères interfaces de stockage	**Sheepdog**	**GlusterFS**	**Ceph**	**Pondération**
mode bloc	**-**	**0,57**	**2**	
compatibilité native	-	1	2	3
ubuntu	-	1	2	1
red-hat	-	1	2	1
compatibilité virtualisation	-	0,5	2	3
kvm ubuntu	-	0	2	3
kvm red-hat	-	2	2	1
xen red-hat	-	2	2	0
mode cloud	**-**	**1**	**2**	
compatibilité openstack	-	1	2	3
compatibilité cloudstack	-	1	2	0
mode fichier	**-**	**2**	**1**	
système de fichier compatible POSIX	-	2	1	3
mode objet	**-**			
compatible openstack swift	-	2	2	0
compatible amazon S3	-	0	2	0

Tableau 5 : Les critères fonctionnels d'interfaces

Définir	Evaluer			Qualifier
Critères fonctions avancées	Sheepdog	GlusterFS	Ceph	Pondération
fonctions avancées	-	0,57	1	
snapshot	-	0	2	3
clones	-	0	2	3
thin-provisionning	-	1	1	3
déduplication	-	0	0	1
géo-réplication	-	2	1	3
erasure-coding	-	1	2	1
IHM	-	1	1	3
api de management	-	1	1	3

Tableau 6 : Les critères pour les fonctions avancées

Définir	Evaluer			Qualifier
Critères architecturaux	Sheepdog	GlusterFS	Ceph	Pondération
architecture	-	0,8	1,6	
distribution dynamique des objets	-	1	2	3
caching	-			3
auto-réparation	-	1	2	3
gestion du quorum	-	1	2	3
load balancing	-	1	2	3

Tableau 7 : Les critères architecturaux

5.3.5. Sélectionner

L'objectif de cette étape consiste à sélectionner le logiciel correspondant le mieux aux besoins, en fonction des notations appliquées lors de l'étape précédente.

Pour cela, il est possible d'utiliser, par exemple, une représentation graphique de type « radar » afin d'obtenir une comparaison visuelle.

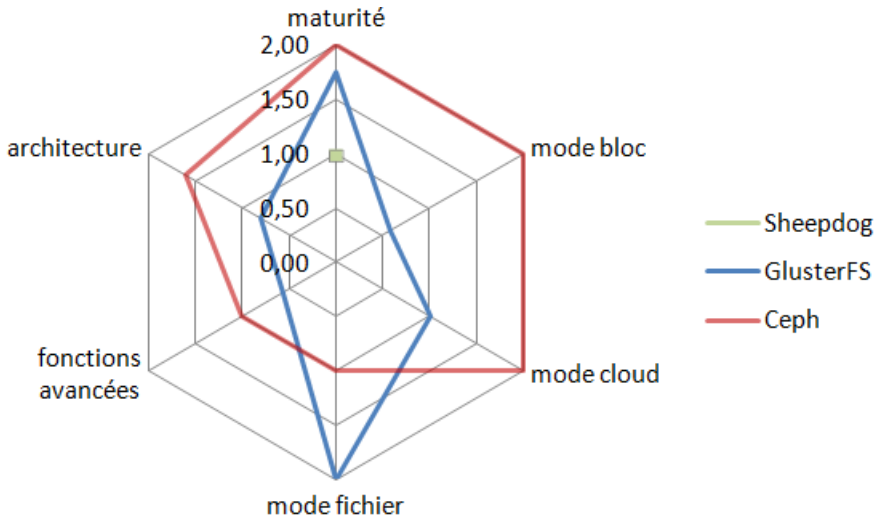

Figure 15 : Radar QSOS

5.1. Bilan de l'étude générale

Sheepdog :

Sheepdog propose des fonctionnalités intéressantes au niveau de la gestion des volumes (snapshot, clones, thin-provisioning). En termes de maturité, le projet souffre globalement d'un manque de visibilité et d'une faible adoption. Ceci m'a amené à éliminer rapidement cette solution de l'étude.

GlusterFS :

GlusterFS propose une solution robuste et simple. GlusterFS est la seule solution à proposer un système de géo-réplication permettant de réaliser des copies asynchrones de volumes sur un cluster géographiquement distant.

En revanche GlusterFS ne dispose pas encore de certaines fonctionnalités avancées telles que la possibilité de réaliser des snapshots ou clones de volumes.

Au niveau du stockage en mode bloc, j'ai noté que la librairie libgfapi, qui permet un accès natif aux périphériques blocs de GlusterFS, n'était pas compilée dans la version de KVM installée sur nos hyperviseurs[10] (ubuntu 12.04). La solution n'est utilisable, dans notre environnement, qu'à la condition de recompiler QEMU afin d'y inclure cette librairie, et ceci au détriment du support logiciel Ubuntu assuré par la société Canonical. Une autre solution est d'utiliser GlusterFS en mode FUSE (Filesystem in UserSpacE en français : « système de fichiers en espace utilisateur ») sur les hyperviseurs, au prix de performances médiocres.

Le projet peut être considéré comme très mature au niveau de sa qualité. Par contre, suite à une réunion de présentation réalisée à ma demande par Red Hat, il semble que Red Hat Storage soit assez peu déployé en Europe pour le moment. La solution GlusterFS communautaire, elle, est utilisée mais sur des clusters de relativement petite taille.

En conclusion la solution GlusterFS ne répond que partiellement à notre besoin et ne peut donc être retenue.

[10] https://bugs.launchpad.net/cloud-archive/+bug/1246924

Ceph :

Ceph propose une solution innovante sur de nombreux points et riche en fonctionnalités. Dans l'ensemble Ceph correspond parfaitement aux besoins. L'utilisation de Ceph en mode bloc est prévue nativement dans le noyau Linux et ne nécessite donc pas d'installation de modules complémentaires.

Parmi les points négatifs, je note tout de même que le système de fichiers CephFS est toujours en cours de développement et n'est pas prêt pour la production. Ceci rend la fonction « stockage en mode fichier » plus difficile à mettre en œuvre. Autre point négatif, Ceph ne propose pas de géo-réplication asynchrone (le caractère asynchrone étant important pour cette fonctionnalité, comme nous le verrons plus tard). Ces problèmes peuvent, cependant, être contournés par la mise en place d'architectures spécifiques.

Au niveau des critères de maturité, le projet peut être considéré comme très mature. Le projet est particulièrement actif, bien documenté et de très bonne qualité.

J'ai eu l'occasion de participer, avec trois autres membres de l'équipe projet le 11 février 2014, à une rencontre d'utilisateurs Ceph à Rennes (Ceph Breizh Meetup). Cette rencontre m'a permis d'échanger avec un panel de personnes utilisant Ceph dans un contexte de production sur des clusters à forte capacité. Orange, Crédit Mutuel Arkéa et l'Université de Nantes étaient par exemple représentés. En conclusion de cette rencontre, j'ai pu me rendre compte que Ceph était utilisé par des acteurs d'envergure, y compris localement, et qu'un fort esprit communautaire animait cette solution, ce qui est un point extrêmement positif.

A l'issue de cette étape d'étude générale, une comparaison visuelle des solutions à partir du graphique radar permet d'affirmer que la solution Ceph est la plus adaptée. Ceph est, par conséquent, la solution retenue. Le prochain chapitre traite de l'étude détaillée des concepts de Ceph, de la définition des architectures et de la conception du plan de test.

6. L'étude détaillée

6.1. L'architecture de stockage unifiée

Ceph est un système de stockage distribué particulièrement novateur et polyvalent qui tend à devenir une référence dans les domaines du stockage de masse et des architectures Cloud. Comme déjà évoqué, Ceph propose une capacité de stockage extensible en utilisant du matériel standard tel que des serveurs x86.

L'architecture de Ceph repose sur deux couches :

La couche fournisseur de stockage : Elle est basée sur un magasin de stockage appelé **RADOS** (Reliable, Autonomic, Distributed Object Store). RADOS est responsable de la distribution des données sur les nœuds du cluster et de leur réplication, afin de garantir la tolérance aux pannes. RADOS est livré avec une bibliothèque d'API (librados) compatible avec C, C++, Java, Python, Ruby et PHP. Cette bibliothèque sert d'interface entre le magasin d'objets RADOS et les interfaces de stockage de la couche supérieure. RADOS s'appuie directement sur le système de fichiers existant des serveurs (ext4 ou xfs par exemple) et ne nécessite pas l'utilisation de RAID.

La couche interface de stockage : Ceph comporte trois interfaces de stockage permettant trois usages différents :

- **RADOSGW** (Rados GateWay) fournit une interface de type REST, permettant au client d'interagir avec le stockage via le protocole HTTP et une API compatible Amazon S3. Ce type de service est couramment appelé « stockage objet » et correspond au même type de solution déjà implémenté chez PagesJaunes : OpenStack Swift.
- **RBD** (Rados Block Device) fournit un accès en mode bloc au magasin RADOS, ce qui le rend notamment utilisable à des fins de virtualisation. C'est cette interface qui est étudiée dans le cadre des besoins de PagesJaunes.

- **CephFS** (Ceph FileSystem) fournit un système de fichiers distribués compatible POSIX s'appuyant sur RADOS. Ce système de fichiers est toujours en cours de développement et n'est donc pas conseillé pour un usage en production.

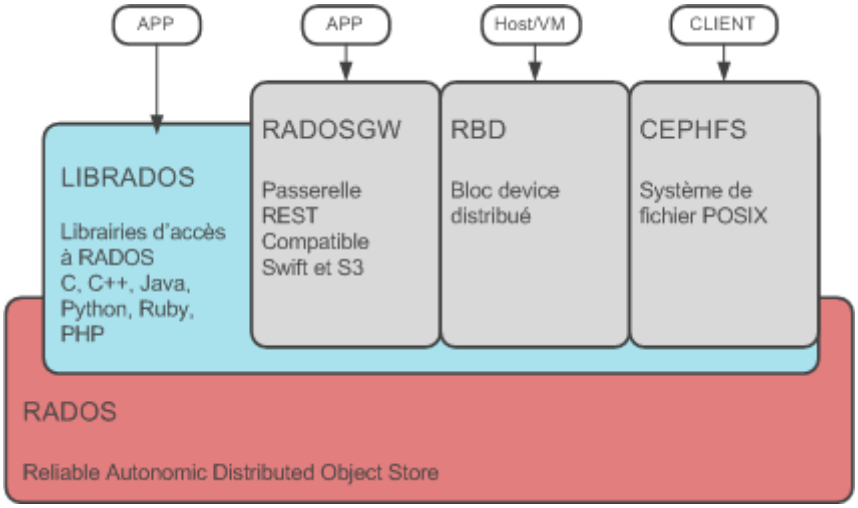

Figure 16 : Les couches de Ceph

6.2. Les composants

RADOS, en dehors de l'utilisation de CephFS, repose sur deux composants majeurs : Les OSD (Object Storage Daemon) et les moniteurs (MON).

6.2.1. Les OSD Object Storage Daemon

Les OSD (Object Storage Deamon) sont des processus chargés de la gestion des données et de l'intelligence de stockage. Ils stockent chaque objet sous forme binaire dans un espace d'adressage plat (pas de hiérarchisation par répertoire). Un OSD correspond à un processus chargé de la gestion d'un disque physique. Un nœud de stockage possède donc autant d'OSD que de disques mis à la disposition du cluster Ceph. Chaque OSD dispose d'un espace mémoire et de ressources CPU mis à disposition par le nœud de stockage hôte. Sur un cluster de grande taille, la somme des ressources allouées aux OSD peut être extrêmement importante (plusieurs centaines de processeurs et quelques To de mémoire RAM), par contraste avec les systèmes de stockage traditionnels limités physiquement en puissance de calcul.

6.2.2. Les moniteurs

Les moniteurs sont des processus Ceph chargés de réaliser l'interface entre les clients et le magasin de stockage RADOS en maintenant une carte globale de l'état du cluster. Dans cette carte, on peut notamment trouver l'OSDmap, carte des OSD, et la MONmap, carte des moniteurs.

Les clients contactent les moniteurs afin de connaître la liste des emplacements susceptibles d'accueillir leurs données. En cela, les moniteurs sont des processus particulièrement critiques qu'il faut absolument redonder.

Afin de garantir l'intégrité des cartes gérées par les moniteurs, ceux-ci implémentent un mécanisme imposant une consistance stricte des cartes. Ceph utilise le protocole PAXOS permettant de résoudre le consensus dans un réseau de nœuds faillibles. Le consensus nécessite d'obtenir une majorité de moniteurs actifs (par exemple 2 sur 3).

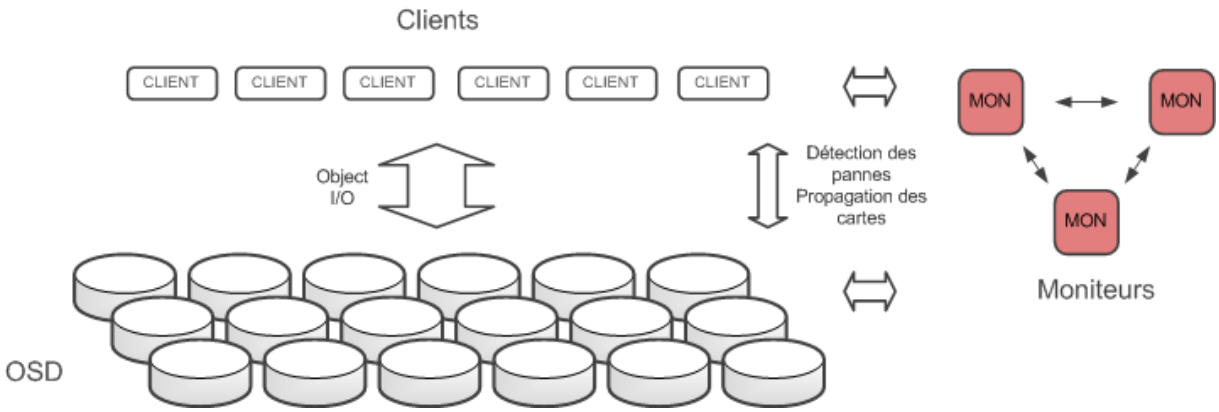

Figure 17 : Les deux composants principaux : OSD et MON

6.3. La gestion dynamique du cluster

6.3.1. L'algorithme de placement

Ceph utilise un algorithme de gestion responsable du placement des données nommé CRUSH (Controlled Replication Under Scalable Hashing).

CRUSH permet à Ceph de se passer de serveurs de métadonnées qui constituent souvent pour les systèmes de stockage distribués, un point de contention important et un point de faiblesse

unique (SPOF Single Point of Failure). Les serveurs de métadonnées ont pour rôle de tenir à jour une topologie détaillée et complète des données stockées. Lorsqu'un accès à une donnée est requis par un client, celui-ci contacte au préalable les serveurs de métadonnées qui vont lui indiquer le nœud de stockage à utiliser ainsi que l'emplacement de la donnée.

Ceph propose, grâce à l'algorithme CRUSH, une approche différente où l'emplacement des données est calculé par un algorithme de placement pseudo-aléatoire, et donc déterministe. CRUSH est utilisé par l'ensemble des composants de Ceph, clients compris, et permet de définir l'emplacement des données en fonction de la carte du cluster RADOS. Pour rappel, cette carte, très légère, est maintenue par les moniteurs. Lorsqu'un accès à une donnée est requis par un client, celui-ci contacte un des serveurs moniteurs afin de vérifier qu'il dispose bien de la dernière version de la carte du cluster, puis, en fonction de celle-ci, calcule grâce à CRUSH l'emplacement des objets sur le cluster. Ceph permet donc un accès direct entre les clients et les nœuds de stockage et ne comporte pas de point d'entrée unique.

Ceph permet, de plus, une distribution de la charge particulièrement effective en découpant les fichiers en « objets » de 4Mo. Les objets sont ensuite placés dans des groupes de placement (placement group PG). Les groupes de placement sont affectés à des OSD différents grâce à CRUSH. Lors de l'accès à un fichier, de multiples OSD sont sollicités ce qui permet de répartir la charge sur le cluster.

6.3.2. La gestion de la typologie du cluster

CRUSH permet de gérer la typologie du cluster de manière extrêmement fine en proposant un système hiérarchique de placement des OSD suivant les niveaux suivants :

- datacenter (centre de donnée)
- room (salle)
- row (travée)
- rack (armoire)
- host (serveur)
- device (disque)

Ce système hiérarchique permet de configurer des zones de disponibilité, c'est-à-dire des zones distinctes conçues pour ne pas être affectées en cas de panne sur une zone voisine. Il

faut prendre en compte que le stockage logiciel, fonctionnant sur du matériel standard et faillible, considère que la panne matérielle est un événement normal, susceptible d'intervenir fréquemment et devant donc être géré de manière appropriée.

Afin de prendre en compte ces cas de panne, il est par exemple possible de configurer CRUSH par des règles de placement (placement rules), pour que les données soient répliquées sur des serveurs différents.

En cas de panne sur un serveur, les OSD et les MON remarquent quasi-instantanément la perte de disques du cluster et engagent un processus d'autoréparation. Le cluster amorce une reconstruction de l'ensemble des PG manquants sur les autres serveurs, en respectant les règles de placement définies dans CRUSH, jusqu'à atteindre le nombre de réplicas de données requis.

Il faut noter que la reconstruction de certains clusters distribués peut engendrer une charge extrême, provoquée par une réorganisation totale de l'organisation des données (replication storm : tempête de réplication). L'algorithme CRUSH a été conçu pour limiter au maximum les déplacements de données lors des pannes tout en garantissant des performances acceptables.

6.4. L’architecture du cluster

6.4.1. Dimensionnement

Architecturer un cluster Ceph repose avant tout sur la bonne définition des besoins en termes de performance et de volumétrie.

Concernant la volumétrie, j'ai choisi de débuter la construction du cluster sur la base d'un volume relativement modeste d'environ 10To utiles. Cette capacité permettra de stocker plusieurs dizaines de machines virtuelles dans un premier temps, puis la capacité pourra être étendue au besoin.

Compte tenu de l'usage qu'il en sera fait, il est important de savoir si le cluster doit être optimisé pour favoriser la performance ou la capacité. Cette question est étroitement liée à la nature des données qui sont accueillies. Dans notre cas de figure, nous sommes en présence

d'une situation mixte puisque certaines données, telles que les données gérées en mode fichier (NAS), peuvent être qualifiées de « données froides » qui sont peu sujettes aux changements et nécessitent seulement une capacité de stockage importante. D'autres données, telles que les images des machines virtuelles, peuvent être qualifiées de « données chaudes » nécessitant des temps d'accès faibles et donc des performances maximales.

Le tableau suivant illustre deux exemples de configuration pouvant convenir au dimensionnement des serveurs de stockage suivant les cas de figure :

Serveur	Type performance	Type capacité
type de châssis	serveur type 1U	serveur type 2U ou 4U
processeur	8 cœurs	4 à 8 cœurs
disques	sas 300Go 15000tr/min	sata 1 à 2To 7500tr/min
journaux	journaux ceph sur ssd	journaux ceph sur sata
réseau	2 ports ethernet 10Gb/s	2 ports ethernet 10Gb/s
RAM	2Go RAM par disque	2Go RAM par disque

Tableau 8 : Exemples de configurations matérielles

Afin de ne prendre aucun risque sur les performances, j'ai choisi de me baser sur la configuration « performance » permettant de répondre avant tout aux besoins des machines virtuelles.

La première configuration, orientée performance, correspond pour beaucoup aux serveurs que nous utilisons actuellement en tant qu'hyperviseurs. Afin de ne pas investir dans un nouveau type de serveurs et de requalifier toute notre chaine d'installation pour un nouveau modèle, j'ai choisi d'utiliser le même modèle de serveurs, en adaptant seulement la configuration disque entre les serveurs typés « puissance » et les serveurs typés « capacité ».

Serveur	Type performance	Type capacité
type de châssis	serveur type 1U	serveur type 1U
processeur	Intel Xeon 2*6 cœurs	Intel Xeon 2*6 cœurs
RAM	16Go	16Go
disques système	2 x sas 300Go 15000tr/min RAID1	2 x sas 300Go 15000tr/min RAID1
disques données	**5 x sas 300Go 15000tr/min RAID0**	**5 x sata 1To 7500tr/min RAID0**
journaux	1 x sas 64Go SSD	1 x sas 64Go SSD
réseau	4 x ethernet 1Gb/s	4 x ethernet 1Gb/s
RAM	2Go RAM par disque	2Go RAM par disque

Tableau 9 : La configuration matérielle choisie

Dans cette configuration, il faut noter les deux points suivants :

1) L'usage de disques SSD est fortement recommandé pour les journaux de Ceph dans le but de garantir un bon niveau de performance. En effet, afin de garantir un acquittement des écritures sur chaque nœud, Ceph utilise un mécanisme de journal faisant transiter les écritures sur un espace temporaire avant de libérer (flush) les écritures sur le disque de l'OSD de destination. Cet acquittement devant être le plus rapide possible, l'usage de disques SSD est recommandé. Dans le contexte de production actuel, les SSD ne sont pas utilisés, nous avons donc peu de recul quant à leur niveau de performance et leur fiabilité.

2) Il est recommandé d'utiliser un réseau dédié 10Gb afin de garantir une bande passante suffisante entre les nœuds du cluster et les clients. Les switchs réseau dont nous disposons ne sont actuellement pourvus que de ports 1Gb. Après étude, il s'avère que le passage en 10Gb représente un investissement conséquent qu'il semble risqué de prendre pour le moment. Afin de pallier la limitation de bande passante en 1Gb, j'ai choisi de :

- Ne pas utiliser des serveurs disposant de trop d'OSD. En effet, plus le nombre de disques est important, plus la bande passante nécessaire pour en tirer pleinement parti augmente. Mieux vaut donc multiplier les serveurs que le nombre de disques par serveur.
- Configurer le réseau grâce à des agrégats de lien 1Gb. Cette technique vise à regrouper plusieurs liens réseau en un seul, de manière à augmenter la bande passante totale en configurant conjointement le switch et le serveur.

Finalement, la solution reposera sur :

- 9 serveurs disposant chacun de 5 disques de 300Go dédiés aux OSD. Ces serveurs sont orientés vers la « performance ».
- 3 serveurs disposant chacun de 5 disques de 1To dédiés aux OSD. Ces serveurs sont orientés vers la « capacité ».

Serveur	Type performance	Type capacité
disques par serveur	5 x sas 300Go RAID0	5 x sata 1To RAID0
nombre de serveurs	9	3
capacité	13,5To	15To

Tableau 10 : Capacité brute du cluster

La capacité brute du cluster est donc de 28.5To sans prendre en compte les facteurs de réplication, c'est-à-dire de 10 à 20To utiles suivant la technique de protection des données utilisée (ex : réplication ou « erasure coding »).

Concernant les serveurs moniteurs, ils ne nécessitent pas une configuration puissante, car leur rôle principal n'est que de maintenir une carte du cluster. Trois machines virtuelles feront office de serveurs moniteurs pour le cluster, ce qui permet d'éviter de consommer inutilement des machines physiques.

6.4.2. Définition de l'architecture physique

Au contraire des architectures de stockage actuellement utilisées, nous souhaitons pouvoir bénéficier d'une architecture de stockage hautement disponible et tolérante aux pannes. Ceph est un cluster distribué hautement disponible et permettant un haut degré de configuration avec notamment la définition de zones de disponibilité. Pour rappel, une zone de disponibilité est un espace physique fonctionnant de manière autonome et isolé, de manière à ce qu'une panne provenant d'une zone de disponibilité voisine ne puisse en affecter le fonctionnement.

Il est donc primordial de définir correctement les zones de disponibilité du cluster au regard de l'environnement d'hébergement.

1ère solution : La première idée de répartition venant à l'esprit serait de réaliser un cluster global utilisant les deux datacenters puisqu'ils sont, au premier abord, parfaitement indépendants. Cependant, ce n'est pas entièrement vrai puisqu'ils sont reliés entre eux par des fibres 10Gb connectées par deux concentrateurs réseau sans redondance. En cas de panne sur l'un de ces équipements ou de mise à jour, le cluster se trouverait dans une situation de partitionnement majeure. Ceph fonctionnant sur la base de l'établissement d'un quorum, le stockage d'un des datacenters s'en trouverait inaccessible.

2ème solution : Un autre choix possible consiste à réaliser deux clusters, un par datacenter, et d'exploiter au mieux les caractéristiques de chacun. Cette solution a été retenue et est détaillée ci dessous :

Dans notre datacenter principal, trois salles serveurs peuvent être utilisées comme zones de disponibilité. Ces trois salles sont, en effet, indépendantes sur la sécurité incendie et la sécurité des accès. Elles sont partiellement indépendantes concernant leur alimentation

électrique avec l'usage de tableaux électriques différentiés. Elles sont aussi partiellement indépendantes au niveau de la climatisation.

Composants	Indépendance entre Datacenter 1 et 2	Indépendance entre salles 1, 2 et 3 du Datacenter 1
Alimentation électrique	totale	partielle
Climatisation	totale	partielle
Extinction incendie	totale	totale
Sécurité physique	totale	totale
Interconnexion réseau	faible	configurable

Tableau 11 : Degrés d'indépendance des zones

Un serveur moniteur est installé dans chaque salle. En cas d'indisponibilité d'une salle, les deux restantes continuent de communiquer et les membres du cluster continuent d'obtenir un quorum. Le cluster reste donc fonctionnel.

Sur le deuxième datacenter, un cluster Ceph totalement indépendant et n'exploitant qu'une seule salle est installé. Dans un premier temps, pour des raisons de budget, ce cluster n'est composé que de trois serveurs orientés « capacité », qui permettront de satisfaire les besoins de stockage de masse en « mode fichiers » et les snapshots de volumes. Dans un second temps, ce cluster sera étendu avec des nœuds de type « performance », permettant de gérer les machines virtuelles.

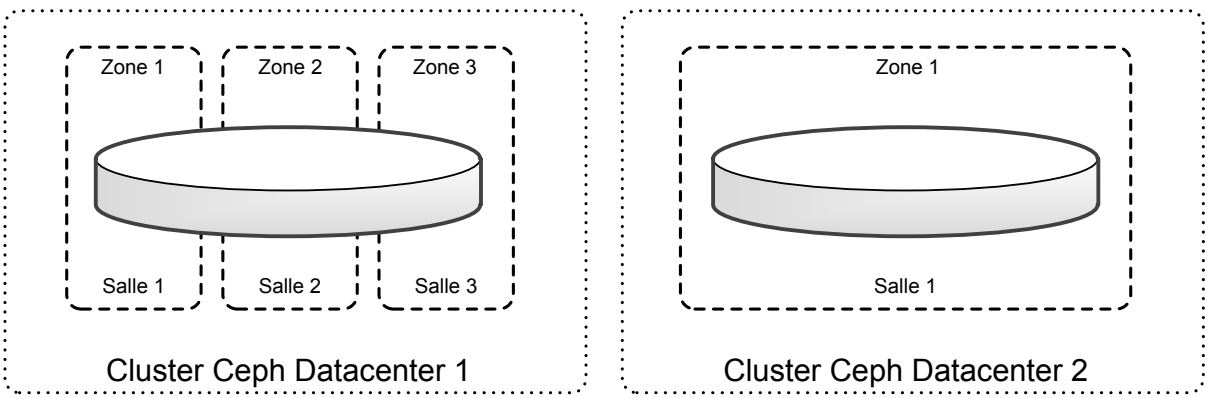

Figure 18 : Répartition des zones des clusters

6.4.3. Définition de l'architecture cluster

La gestion de la topologie du cluster se fait en éditant la carte principale du cluster, appelée **CrushMap**. Cette carte est utilisée par l'algorithme CRUSH, en conjonction avec des règles de gestion, dans le but de répartir les données selon les souhaits de l'administrateur.

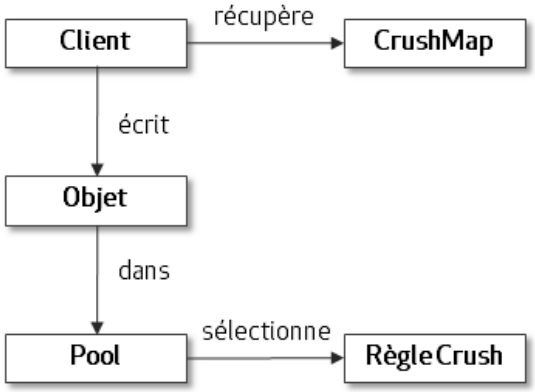

Figure 19 : Le mécanisme de placement

Configuration de la topologie :

La CrushMap définit plusieurs notions importantes :

- La liste des périphériques de stockage (OSD), avec leur poids respectif en fonction de leur capacité (le cluster gère des équipements qui peuvent être hétérogènes).
- L'arborescence de ces périphériques dans le cluster, notamment au niveau des zones de disponibilité (les salles serveurs).
- Au moins une racine de stockage. Dans notre cas, nous définissons deux racines en fonction de l'usage (disques SAS pour la performance, disques SATA pour la capacité).

Une fois éditée, la CrushMap doit être compilée puis injectée dans le cluster. Elle est ensuite propagée aux serveurs et aux clients par l'intermédiaire des moniteurs. La figure suivante représente une visualisation symbolique et abrégée de la CrushMap, telle qu'elle pourrait apparaitre en questionnant le cluster du datacenter 1 avec la commande « ceph osd tree ».

```
pool sas
      room salle_1_sas
            host ceph01
                  osd.1 à osd.5
            host ceph02
                  osd.6 à osd.10
            host ceph03l
                  osd.11 à osd.15
      room salle_2_sas
            host ceph04
                  osd.16 à osd.20
            host ceph05
                  osd.21 à osd.25
            host ceph06
                  osd.26 à osd.30
      room salle_3_sas
            host ceph07
                  osd.31 à osd.35
            host ceph08
                  osd.36 à osd.40
            host ceph09
                  osd.41 à osd.45
pool sata
      room salle_1_sata
            host ceph10
                  osd.46 à osd.50
      room salle_2_sata
            host ceph11
                  osd.51 à osd.55
      room salle_3_sata
            host ceph12
                  osd.56 à osd.60
```

Figure 20 : La typologie abrégée du cluster principal

Détail :

- pool sas et pool sata sont les racines de stockage orientées respectivement « performance » et « capacité ».
- room salle_1, salle_2 et salle_3 sont les salles serveurs.
- host ceph01 à ceph12 sont les nœuds de stockage Ceph.
- osd.1 à osd.60 sont les périphériques de stockage.

Configuration des règles

Une fois que la typologie est définie, il est nécessaire de configurer les règles de placements des objets au sein de cette configuration.

Une règle définie :

- Le nombre de réplicas de chaque objet.
- La base de réplication (réplicas propagés entre serveurs, entre racks ou entre salles...).
- Le type de protection de données : par réplication ou par un mécanisme similaire à du RAID 5 réseau, nommé « erasure coding[11] » et permettant des gains d'espace disque important. Ce mécanisme est à privilégier pour les données archivées.
- Eventuellement des règles complexes : placer le premier réplica sur le pool SAS et les autres sur le pool SATA, etc.

Dans notre cas, deux règles sont définies sur le même modèle, une pour le pool SAS et l'autre pour le pool SATA :

- Le nombre de réplicas de chaque objet est fixé à 2 ou 3 suivant le degré de protection souhaité.
- La base de réplication : Les réplicas sont paramétrés pour être écrits dans des salles différentes dans le cas du datacenter 1, et sur des serveurs différents dans le cas du datacenter 2.

6.5. La définition de l'architecture réseau

6.5.1. La définition de l'architecture réseau physique

Pour garantir une infrastructure réseau haute disponibilité et extensible aux besoins de Ceph, nous avons étudié divers scénarios présentés ci-après :

[11] Option erasure coding disponible dans la version 0.8 de Ceph sortie en mai 2014

1ère solution : le réseau en étoile

Le premier scénario, qui n'a pas été retenu, propose d'intégrer la solution Ceph en utilisant les règles générales d'intégration réseau de PagesJaunes, qui consistent à relier chaque switch à un concentrateur réseau appelé « cœur de réseau ».

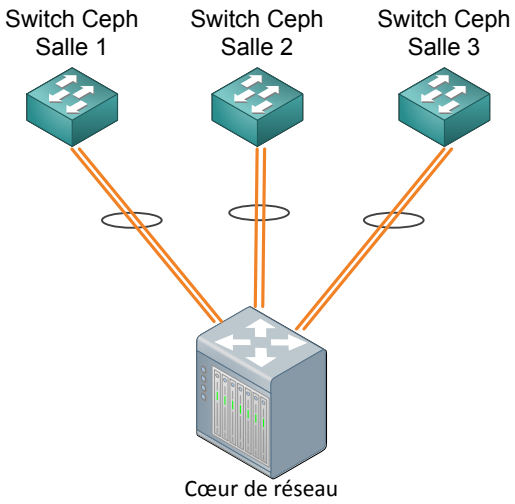

Figure 21 : Réseau en étoile

Cette solution est la plus simple à mettre en œuvre, mais elle a néanmoins l'inconvénient principal de ne pas être tolérante aux pannes. En effet, en cas de panne du cœur de réseau, ou de mise à jour nécessitant un redémarrage de celui-ci, l'ensemble du cluster serait impacté. Le cœur de réseau représente un point de faiblesse unique, autrement appelé SPOF (Single Point Of Failure).

Avantages	Inconvénients
Simplicité de mise en œuvre	Le cœur de réseau est un SPOF majeur
Gestion de l'extensibilité facilitée	Pas de switching au plus court

Tableau 12 : Avantages/inconvénients d'un réseau en étoile

2ème solution : le réseau en anneau :

Une deuxième solution, qui a été retenue, est de réaliser un anneau constitué de trois switchs où chaque switch est situé dans une des trois salles serveurs du datacenter 1. Cette conception en anneau permet d'assurer la connectivité entre deux switchs même en cas de défaillance d'un des switchs (quel qu'il soit).

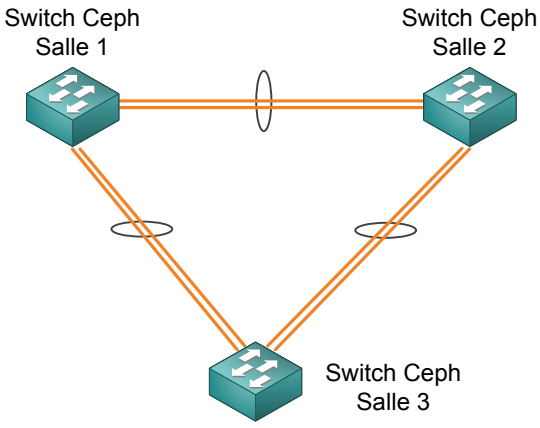

Figure 22 : Réseau en anneau

Cette solution est plus complexe à mettre en œuvre que la précédente, mais répond parfaitement aux principes des zones de disponibilité totalement indépendantes et permet donc d'avoir une isolation réseau forte entre les salles. Cette solution permet, de plus, de réaliser les opérations de commutation réseau entre deux switchs sans rebond vers un switch intermédiaire, ce qui permet d'obtenir une latence optimale.

Avantages	Inconvénients
Pas de SPOF	Industrialisation réseau spécifique
Switching au plus court	Gestion de l'extensibilité non-triviale

Tableau 13 : Avantages/inconvénients d'un réseau en anneau

6.5.2. Le choix du protocole de switching

La conception en anneau implique la présence d'une boucle réseau qu'il faut maîtriser. Pour cela, plusieurs protocoles peuvent être utilisés :

- Spanning Tree (STP) ou Rapid Spanning Tree (RSTP).
- EAPS (Ethernet Automatic Protection Switching, protocole propriétaire Extreme Networks[12]) : ressemblant à STP, mais fonctionnant uniquement en anneau, avec un temps de convergence beaucoup plus rapide que STP ou RSTP.
- MLAG (Multi-Chassis Link Aggregation, protocole propriétaire Extreme Networks).

[12] http://www.extremenetworks.com/

Chaque protocole possède ses avantages et inconvénients. Nous ferons un focus sur les technologies ci-dessus, exceptées STP et RSTP qui sont moins performantes que l'équivalent Extreme Networks : EAPS.

1ère solution : EAPS

La première solution, non retenue, utilise le protocole EAPS. Dans une boucle EAPS, on définit un switch maître et des switchs de transit. Chaque switch possède un port primaire et un port secondaire.

Le switch maître bloque son port secondaire afin d'éviter la boucle. Lorsqu'un switch détecte une défaillance sur un lien, il avertit le switch maître afin que celui-ci active son port secondaire.

Cette solution est simple mais ne permet pas d'utiliser tous les liens simultanément, et donc de profiter d'une bande passante suffisante.

Avantages	Inconvénients
Pas de SPOF	Pas d'utilisation de tous les liens : et capacité de switching limitée à 2 x 2 liens
Résistance à une double panne (crash d'un LAG) entre 2 switches	Pas de switching au plus court

Tableau 14 : Avantages/inconvénients du protocole EAPS

2ème solution : MLAG

La deuxième solution, qui a été retenue, utilise le protocole MLAG d'Extreme Networks.

Un LAG (Link Aggregation Group) est un groupement de liens Ethernet entre deux équipements. Le LAG permet d'augmenter, d'une part, la bande passante et/ou d'augmenter le niveau de sécurité (redondance) selon l'implémentation qui en est faite (partage de charge ou master/standby).

Le MLAG est l'extension de la technologie LAG à plusieurs équipements. Un MLAG est constitué de « pairs » (pour l'extrémité redondée) raccordés entre eux par un lien ISC (Inter

Switch Control). Ce lien permet aux « pairs » de mettre à jour leurs tables de commutation afin d'éviter les boucles.

Avantages	Inconvénients
Tous les liens sont actifs	En cas de panne du double lien ISC, la communication de switch 1 vers switch 2 n'est plus opérationnelle
Switching au plus court chemin	Il faut que les équipements sur l'ISC soient identiques et en version identique.
Pas de SPOF	

Tableau 15 : Avantages/inconvénients du protocole MLAG

6.5.3. La gestion de l'extensibilité

1ère solution : le réseau maillé

Le principe de ce réseau est de reconstruire de nouveaux anneaux à chaque ajout de switchs dans une salle serveur. Ceci revient à réaliser un réseau maillé.

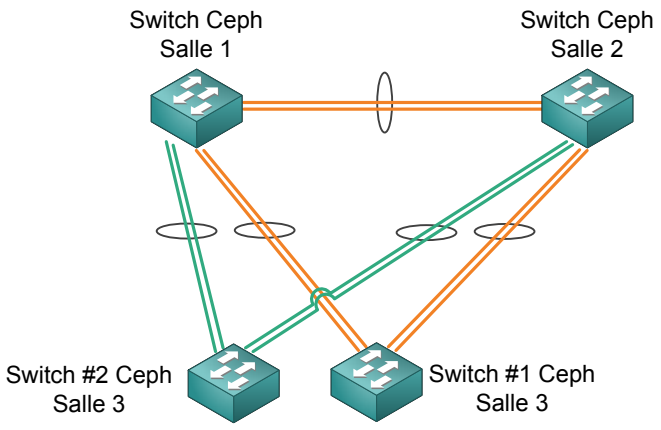

Figure 23 : Extension par ajout d'un anneau

L'avantage de cette solution est de proposer de multiples routes entre les équipements, donc de répartir la charge et de limiter les impacts des pannes sur ces équipements par répartition. Les principaux désavantages sont les problèmes de câblage et l'extension limitée de la solution au nombre de ports disponibles par switch. Cette solution n'a pas été retenue.

Avantages	Inconvénients
Perte limitée en cas de panne d'un switch Pas de spof	Problème logistique important : Passage de nombreuses fibres entre salles
Switching au plus court	Extensibilité limitée : saturation du nombre de ports

Tableau 16 : Avantages/inconvénients d'un réseau maillé

2ère solution : le réseau backbone

Cette solution revient à considérer les switchs 1, 2 et 3 comme des concentrateurs de switchs dans les salles. Les trois switchs principaux constituent un backbone. L'extension de switch peut se faire, sur ces switchs principaux, en constituant une nouvelle étoile autour des switch du backbone ou en chainant les switchs en « stack ».

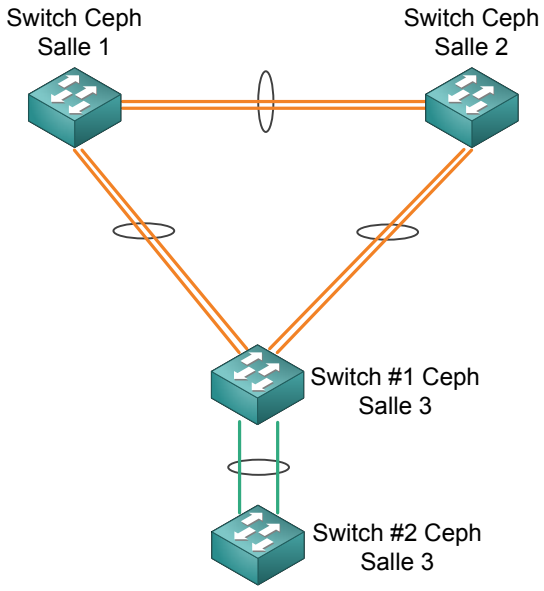

Figure 24 : Extension par ajout de switchs aux nœuds du backbone

Cette solution, qui a été retenue, a pour principal avantage de rester simple. L'extension peut se faire directement en chainant de nouvelles unités aux switchs de l'anneau en utilisant des « stacks », une technologie d'interconnexion de switchs à 10Gb/s, si ceux-ci sont séparés de moins de 10 mètres.

Cette solution présente deux désavantages principaux :

- La bande passante dans le backbone doit être très importante pour gérer le trafic inter-salle. Nous envisageons, dans un premier temps, d'utiliser des agrégats de liens 1Gb/s, puis de faire évoluer les switchs avec des modules spécifiques 10Gb/s en cas de besoin.
- Chaque extrémité du backbone constitue un SPOF. Cependant, ce cas de figure est entièrement pris en compte dans le mécanisme de protection du cluster par zone de disponibilité. Cet inconvénient n'est donc pas critique.

Avantages	Inconvénients
Simplicité de mise en œuvre	Pas de Switching au plus court
Bande passante importante disponible entre switchs d'une même salle (10Gb)	Bande passante importante nécessaire sur les switchs backbone
	Le switch #1 est un SPOF

Tableau 17 : Avantages/inconvénients d'un réseau backbone

6.5.4. L'architecture réseau logique

L'architecture réseau logique globale, intégrant la couche client et la couche cluster Ceph, est décomposée comme suit :

- ***n* réseaux d'accès** aux machines clientes consommatrices du stockage. Ces machines peuvent être des hyperviseurs ou toutes autres machines souhaitant accéder au stockage distribué. L'ensemble de ces réseaux se trouve en dehors de l'architecture Ceph.
- **1 réseau data**. Ce réseau permet l'interconnexion entre les machines clientes et le cluster Ceph. Il est propagé par VLAN dans l'architecture réseau physique décrite précédemment.
- **1 réseau de réplication**. Ce réseau est dédié aux opérations d'écritures et aux processus de réplication entre OSD des différents nœuds Ceph, et n'est donc pas accessible par les clients. Il est également propagé par VLAN dans l'architecture physique décrite précédemment.
- **1 réseau d'administration**, physiquement séparé, permettant d'administrer les machines.

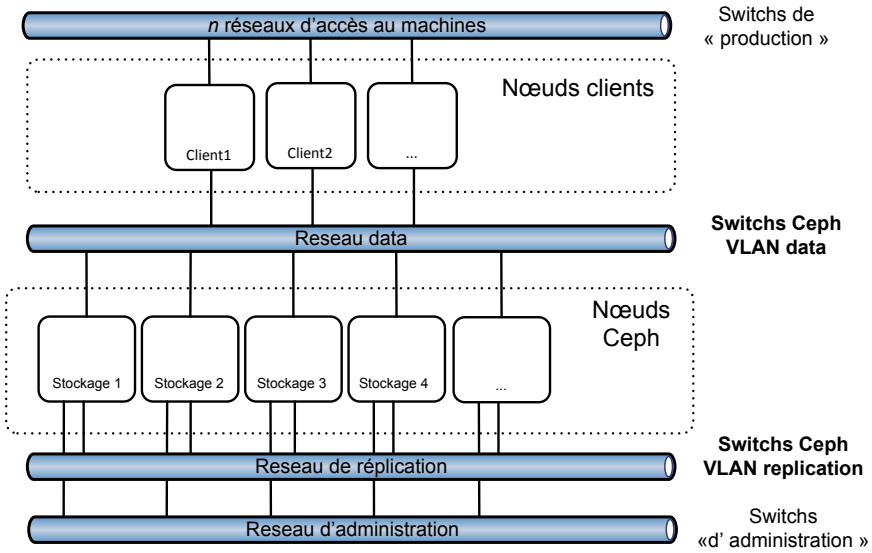

Figure 25 : L'architecture réseau logique

6.6. Le schéma d'implantation générale

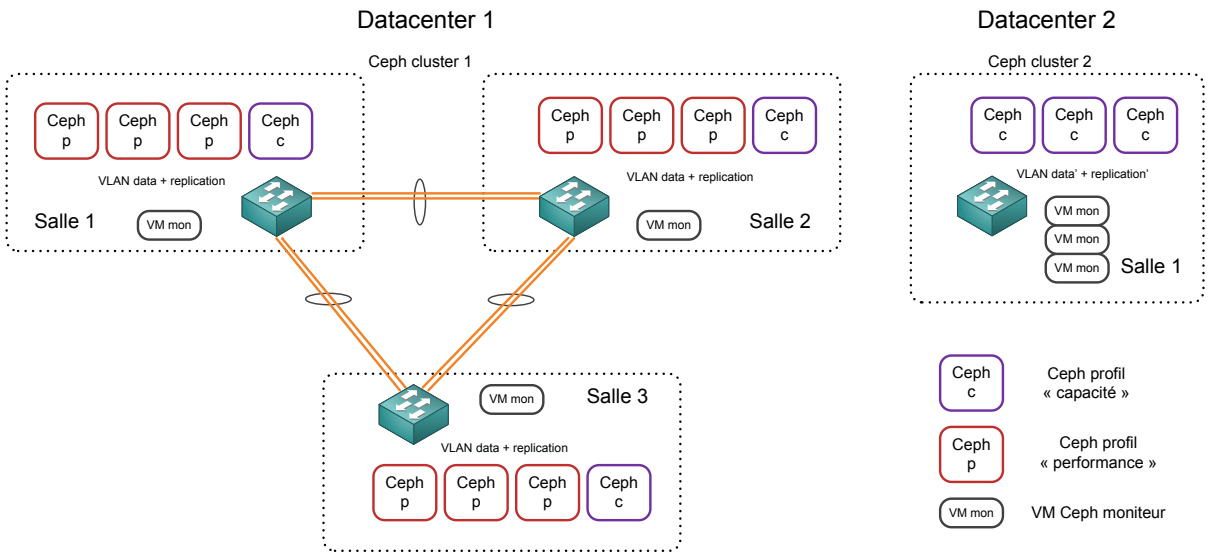

Figure 26 : Le schéma d'implantation générale

6.7. Les architectures des autres composants

Ce chapitre a pour objectif d'expliquer les architectures des composants utilisant la solution Ceph, afin de répondre aux différents besoins exprimés dans les cas d'utilisation.

6.7.1. Stockage pour les machines virtuelles

L'externalisation du stockage des machines virtuelles sur Ceph n'impose pas de changement majeur vis-à-vis de l'infrastructure actuellement utilisée. Dans un premier temps, chaque hyperviseur doit simplement être connecté au réseau de stockage (réseau data) grâce à un ou deux ports Ethernet.

La deuxième étape consiste à échanger une clé d'authentification entre Ceph et les hyperviseurs. Il est primordial que cette clé soit commune à tous les hyperviseurs afin que la fonctionnalité de migration à chaud puisse fonctionner.

Enfin, l'installation de la librairie RBD sur les hyperviseurs ainsi que la reconfiguration des fichiers de définition des machines virtuelles (en XML) utilisés par libvirt avec la liste des serveurs moniteurs Ceph, permettent de connecter une machine virtuelle au cluster Ceph.

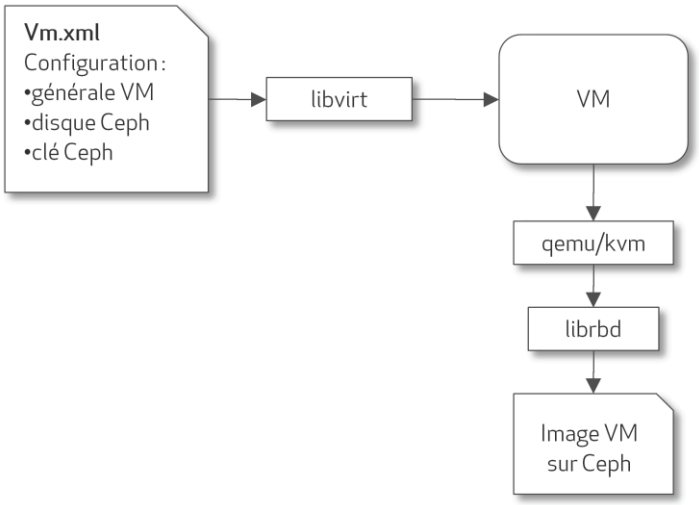

Figure 27 : Architecture KVM Ceph

6.7.2. Stockage Cloud

L'équipe projet « Cloud » a étudié deux solutions afin de répondre à la demande de création d'un Cloud privé IAAS :

- Openstack: Projet open source porté par la société éponyme Openstack.
- Cloudstack : Projet open source porté par la fondation Apache.

Le projet Openstack a été sélectionné, d'une part, pour la grande qualité de son implémentation et, d'autre part, parce que le composant Openstack Swift était déjà utilisé au sein de PagesJaunes en tant que stockage objet.

Openstack est architecturé autour des composants suivants :

- Le composant **Object Storage (nom de code « Swift »)** est la solution de stockage objet qui permet le stockage de fichiers.
- Le composant **Image (nom de code « Glance »)** est le service de catalogue d'images de machines virtuelles.
- Le composant **Compute (nom de code « Nova »)** fournit les machines virtuelles à la demande. Il s'agit d'un service spécifique d'Openstack utilisant un hyperviseur comme support de fonctionnement. Typiquement, le système de virtualisation KVM est utilisé.
- Le composant **Dashboard (nom de code « Horizon »)** est l'interface web proposée aux clients permettant de gérer les machines virtuelles.
- Le composant **Indentity (nom de code « Keystone »)** fournit les services d'identification et d'autorisation pour tous les composants d'OpenStack.
- Le composant **Network (nom de code « Quantum »)** fournit un service de réseau à la demande de type SDN (Software Defined Networking) aux machines virtuelles.
- Enfin, le composant **Block Storage (nom de code « Cinder »)** fournit le service de stockage bloc de la solution en accueillant les images des machines virtuelles et leurs snapshots.

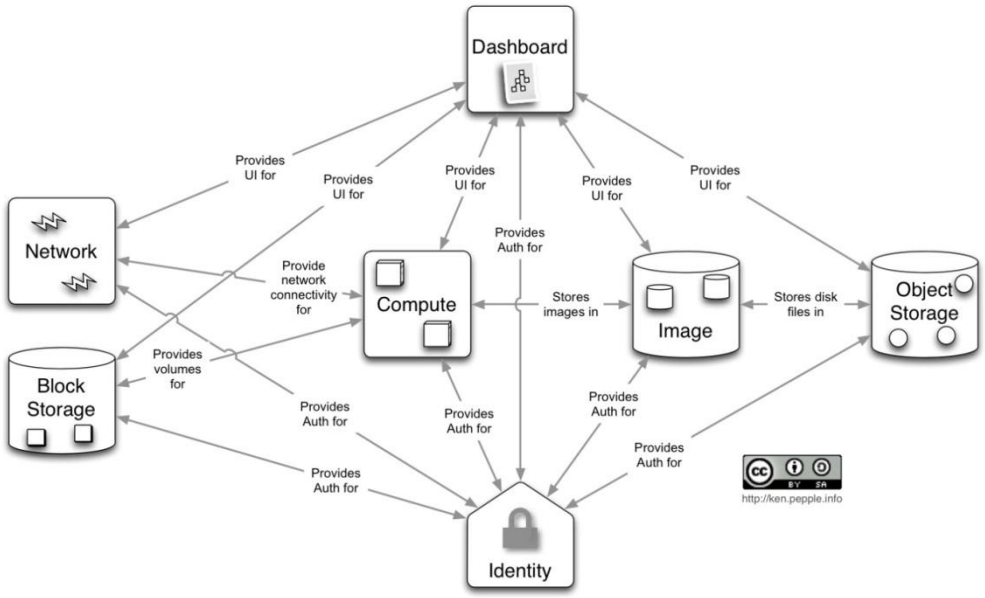

Figure 28 : Interactions entres les composants d'OpenStack.
Source http://ken.pepple.info

L'architecture Cloud, prévue conjointement entre l'équipe projet « stockage distribué » et l'équipe « IAAS », prévoit d'utiliser Cinder (stockage objet) et Glance (catalogue d'images) avec Ceph comme composant de stockage sous-jacent. L'ensemble des composants, à l'exception des composants Compute, est agrégé au sein d'une même machine qui fait office de contrôleur Cloud.

Cette architecture implique une configuration appropriée des composants Glance et Cinder. Il est également nécessaire de paramétrer le composant Nova (service de virtualisation Compute) des hyperviseurs Cloud avec le module RBD, le fichier de configuration des moniteurs et une clé d'authentification Ceph.

6.7.3. Stockage en mode serveur de fichiers

Le stockage en mode serveur de fichiers vise à fournir les mêmes fonctionnalités que les baies NetApp dans la configuration NAS. Le système de stockage doit donc proposer aux clients un système de fichiers utilisant un protocole réseau gérant les fichiers.

Comme nous l'avons déjà évoqué, l'implémentation du système de fichiers distribué de Ceph, nommé CephFS, n'est pas prête pour la production. Il est cependant possible d'utiliser

l'espace de Ceph comme support de stockage en mode fichier, en créant une passerelle entre les clients et le cluster. Cette passerelle est chargée d'utiliser le protocole souhaité, tel que NFS, FTP, ou CIFS.

Un simple serveur disposant d'un attachement sur un volume RBD (en mode bloc) de grande capacité peut facilement faire office de passerelle, en installant le logiciel de serveur de fichiers adéquat. Dans notre cas nous choisissons d'utiliser NFS comme première passerelle, il convient donc d'installer le paquet logiciel « nfs-server ».

Afin de garantir les mêmes fonctionnalités que nos baies de stockage NAS NetApp, la passerelle NFS doit être elle-même de « haute disponibilité ». Dans le but de garantir une disponibilité maximale et de se prémunir d'une panne matérielle, j'ai prévu de réaliser une passerelle montée en cluster autour de deux machines physiques.

Le cluster NFS utilise un gestionnaire de cluster open source, nommé « pacemaker[13] », chargé de gérer des ressources qui, dans notre cas, sont des montages RBD vers Ceph. Les deux machines du cluster communiquent entre elles l'état de santé de chaque membre par un lien nommé « heartbeat ». Ainsi, le cluster est capable d'effectuer les actions nécessaires au rétablissement immédiat du service en cas de défaillance d'un des membres.

Nous configurons pacemaker pour la répartition de charge et la haute disponibilité :

- répartition de charge : Les deux serveurs fonctionnent en même temps, en partage de charge, sur des ressources RBD différentes.
- haute disponibilité : En cas de défaillance d'un des nœuds, les ressources gérées par ce nœud sont automatiquement prises en charge par le nœud restant. Sur le serveur sain, pacemaker va donc monter la ressource RBD, reconfigurer les exports NFS puis basculer l'IP virtuelle du nœud défaillant sur le nœud sain, de manière à ce que les clients pointent vers le nouvel emplacement.

Cette architecture permet d'obtenir une solution similaire en termes de fonctionnalités et de protection des données à nos baies NetApp.

[13] http://clusterlabs.org/

Le schéma suivant illustre l'architecture de la solution.

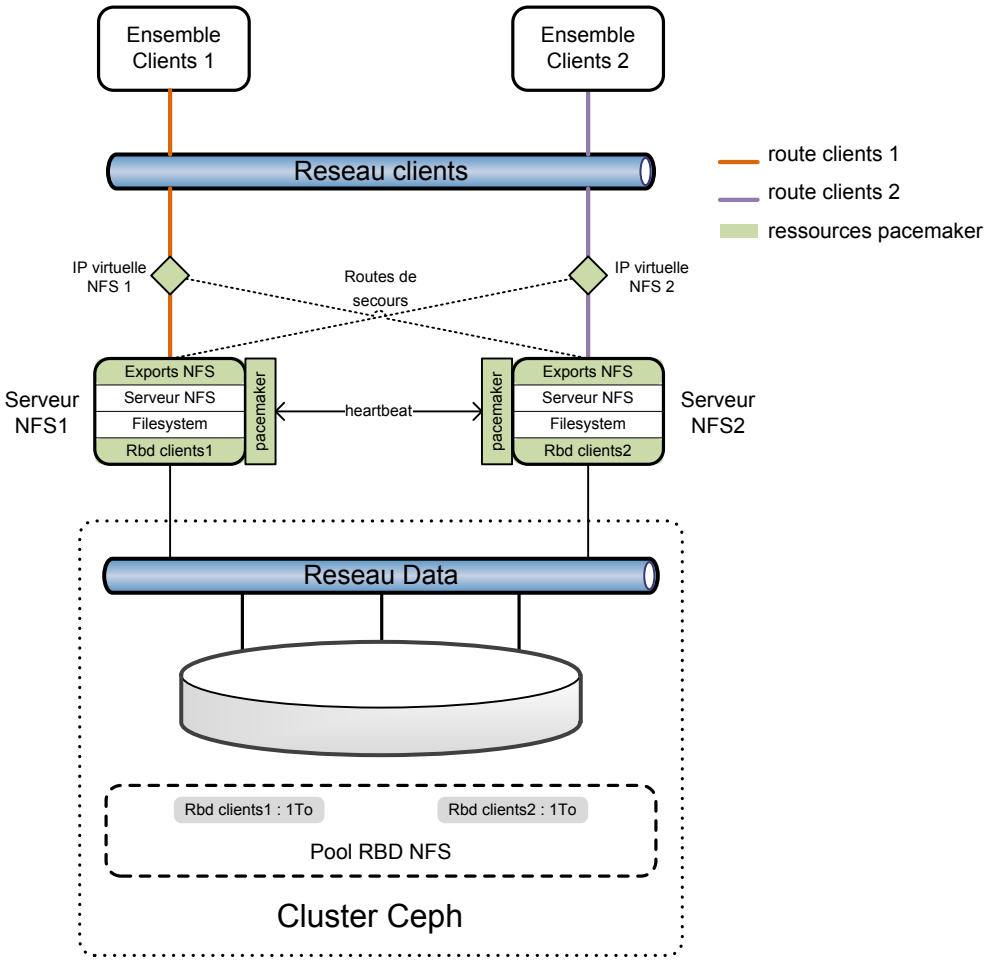

Figure 29 : Architecture d'un cluster NFS basé sur Ceph et géré par pacemaker

6.8. La gestion de la sécurité

6.8.1. La sécurité des accès

Ceph gère la sécurité des accès grâce à l'implémentation d'un protocole d'authentification et d'autorisation nommé CephX. Ce protocole, proche du protocole Kerberos, repose sur l'attribution, par les serveurs moniteurs, d'un ticket de session aux clients réclamant un accès à une ressource du cluster. Chaque client doit disposer d'une clé secrète afin d'obtenir un ticket. Cette clé sert également au chiffrement de la communication durant la phase d'authentification.

6.8.2. Le mécanisme de réplication sur site distant

La sécurité des données est un facteur primordial dans tout système d'information. Or il n'est pas de meilleur moyen de protéger les données contre les désastres, qu'ils soient physiques (ex : incendie) ou d'origine humaine (par exemple une erreur de manipulation d'un opérateur), que de répliquer ces données dans un endroit géographique séparé, physiquement et logiquement indépendant.

Comme déjà évoqué, Ceph n'intègre pas de système de géo-réplication permettant de déporter les données sur un autre site. Une configuration d'un cluster Ceph utilisant deux datacenters permettrait d'avoir un exemplaire des données sur les deux sites, mais au prix des deux contraintes suivantes :

- Une disponibilité non garantie (établissement du quorum impossible sur un datacenter en cas de coupure de l'inter-lien reliant les deux sites).
- Une latence plus importante sur le cluster : toutes les communications étant synchrones, l'écriture sur le site distant doit être acquittée avant que le client ne reçoit l'acquittement final pour sa transaction.
- Un risque de perte de données sur les deux datacenters en cas d'erreur humaine.

Afin de répondre à cette problématique de réplication distante, j'ai choisi de réaliser des synchronisations de données croisées et asynchrones entre les clusters des datacenters 1 et 2, grâce à la fonctionnalité de création d'instantanés (snapshots) intégrée à Ceph. Ceph propose, en effet, un mécanisme capable de figer l'état d'un volume RBD dans le temps et d'en exporter le contenu sous forme de fichiers sur un serveur distant. Cette fonctionnalité est complétée par une fonction d'export différentiel capable d'exporter uniquement les blocs modifiés entre deux snapshots. Ainsi, il est possible de réaliser des instantanés très fréquemment tout en n'utilisant qu'une bande passante et un temps de traitement minimal.

Cette configuration implique :

- la création de deux clusters Ceph (un sur chaque site)
- l'installation de serveurs passerelles initiant une synchronisation des volumes RBD croisée entre les deux sites.

- la réalisation d'un script de synchronisation permettant de réaliser des réplications croisées entre les clusters.

Le schéma suivant illustre l'architecture de la solution.

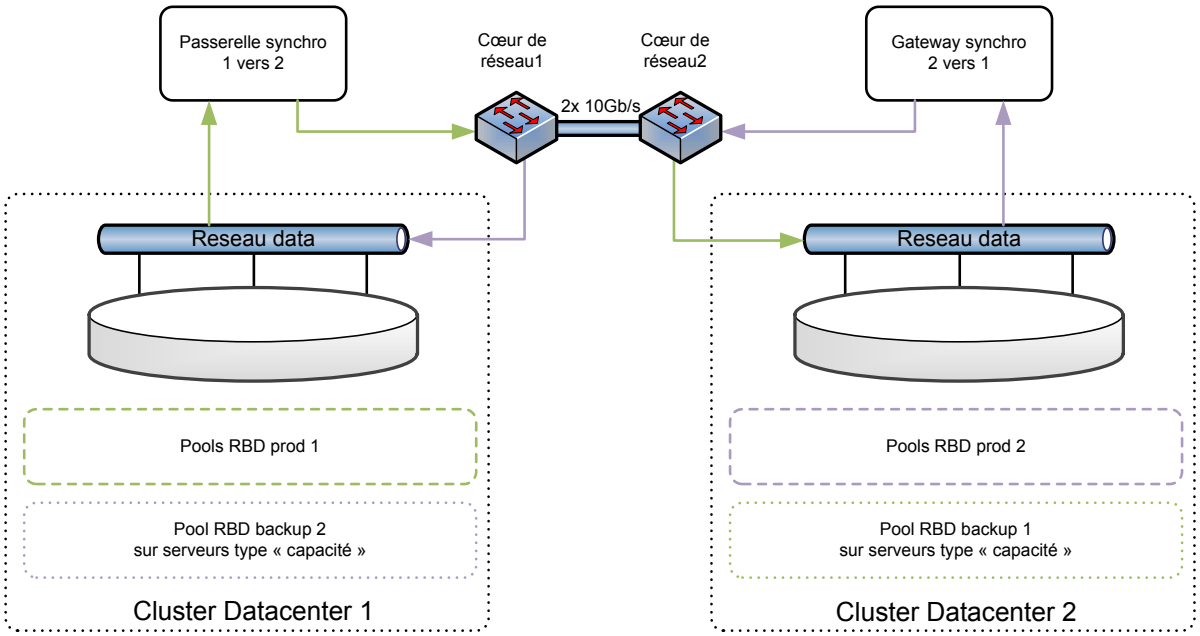

Figure 30 : Architecture du système de géo-réplication

6.9. La gestion des performances

La solution Ceph propose des volumes de stockage reposant sur une architecture physique dont les capacités sont partagées entre de multiples clients. La garantie d'un niveau de performance élevé peut être assurée grâce à l'utilisation de caches, tant au niveau client qu'au niveau cluster, ainsi que par l'utilisation de mécanismes de gestion de la qualité de service.

6.9.1. Les mécanismes de cache niveau client

Le cache sur disque « flashcache » : Pour garantir un niveau de performance maximal sur des périphériques blocs, il est possible d'utiliser des logiciels permettant de tirer parti d'un périphérique local rapide de type SSD. Flashcache (projet opensource développé par

Facebook) permet la création de périphériques « cache » sur SSD de petite capacité, servant de mémoire tampon à un volume plus important mais plus lent.

Ce cache peut être configuré pour les écritures (appelé mode WriteBack dans tous les systèmes de cache). Chaque écriture d'un client est donc acquittée très rapidement sur le périphérique SSD cache local. Ensuite, le cache purge les données sur les OSD Ceph lorsque les données du cache sont modifiées (dirty pages) ou lorsqu'un seuil de remplissage maximal du cache est atteint.

Le cache peut également être configuré uniquement pour les lectures (mode WriteThrought). Dans ce mode, chaque lecture effectuée sur les OSD est cachée au niveau du client sur le disque SSD et est vidée du cache, en fonction de son utilisation, à l'aide d'un algorithme LRU (Least Resently Used). Dans ce mode, les écritures ne passent pas par le cache et sont directement adressées à Ceph.

Le cache mémoire « rbdcache » : L'implémentation de la bibliothèque de gestion RBD (librbd) au niveau des clients ne permet pas de tirer avantage du cache natif de Linux (page-cache). Pour cette raison les développeurs de Ceph ont implémenté un cache spécifique (rbdcache), en mémoire RAM, pouvant être activé par les clients.

6.9.2. Les mécanismes de cache niveau cluster

Le cache RAID : Un cache de 1Go est présent sur chaque contrôleur RAID installé dans les serveurs du cluster Ceph. Ce cache est partagé pour l'ensemble des 8 disques présents dans les serveurs. Par défaut, il est configuré en mode WriteBack, c'est-à-dire qu'il cache les écritures.

Le cache Linux « page-cache » : Chaque serveur du cluster Ceph profite également du cache natif Linux. Tout l'espace RAM libre, non utilisé par le système d'exploitation et par les processus OSD notamment, est disponible pour servir de cache.

Le « cache tiering » Ceph : Ceph implémente depuis sa toute dernière version (V0.80 nom de code Firefly sortie le 7 mai 2014) un mécanisme permettant de monter un pool de stockage en sur-couche d'un autre pool. Cette fonctionnalité, appelée « cache tiering », permet de connecter les clients sur un pool rapide de type SSD, qui lui-même pointe vers un pool plus lent, mais de forte capacité.

6.9.3. La gestion de la QoS

Ceph proposant des espaces physiques de stockage mutualisés entre plusieurs clients, il est primordial de disposer d'un moyen de gérer la qualité de service des ressources (QoS de l'anglais Quality of Service).

KVM supporte nativement la limitation d'accès aux ressources matérielles, notamment au niveau des accès disque. L'implémentation de cette fonctionnalité dans libvirt permet de contrôler finement, par client, la bande passante en lecture, écriture, ou le nombre d'opérations par seconde autorisées.

La solution Openstack, basée sur KVM, intègre également cette fonctionnalité.

Par conséquent, il est possible de contrôler les performances et de créer des niveaux de services différents, que ce soit pour les machines virtuelles classiques ou pour les services de virtualisation Cloud.

6.10. Le plan de qualification

6.10.1. Les objectifs

Le contexte central du stockage distribué dans l'infrastructure d'hébergement réclame une grande attention dans la définition des objectifs principaux du plan de qualification. L'étape de qualification sera décomposée en trois grands domaines de tests :

- Le **domaine fonctionnel** : les tests de validation du respect des besoins fonctionnels et des cas d'utilisation.
- Le **domaine de la fiabilité** : les tests de fiabilité visent à tester les mécanismes de protection des données et à vérifier que le système reste bien disponible, même en cas d'apparition d'une panne sur l'un des composants. La supervision du système est également vérifiée et validée pour chaque cas de test.
- Le **domaine des performances** : les tests d'étude et de validation des performances permettant de déterminer si le système est apte à l'hébergement d'applications dans le contexte de PagesJaunes.

Les ressources matérielles :

Afin de réaliser la qualification, nous disposons d'un « laboratoire système » qui est une baie dédiée à la réalisation de tests matériels ou logiciels. Dans cette baie, j'ai réservé les équipements suivants :

- 6 serveurs de stockage Ceph configurés en type « performance ».
- 2 hyperviseurs (KVM ubuntu) servant de clients Ceph, installés avec 7 disques RAID 5 plus 1 disque SSD séparé dédié aux tests Flashcache.
- 20 adresses IP provisionnées afin d'installer des machines virtuelles.
- 1 switch manageable 48 ports Gb de janvier à mai 2014, puis 3 switchs manageables 48 ports Gb à partir du mois de mai 2014, afin de tester les différentes topologies évoquées précédemment.
- Autres ressources : 1 serveur DNS, 1 serveur NTP, 1 serveur de supervision Nagios, 1 serveur de métrologie Collectd.

Ressources logicielles de pilotage :

La gestion du plan de test est basée sur les outils de documentation et de gestion de ticket familier de l'équipe :

- Outil de pilotage et de gestion des demandes associées aux tests : JIRA[14].
- Outil de documentation et de construction du cahier de test : Confluence.

Architecture réseau :

L'architecture réseau est prévue pour se rapprocher le plus possible d'une situation de production. L'architecture prévoie donc la création de trois réseaux distincts (VLAN) sur le switch, ainsi que la configuration d'agrégats de port à l'aide du protocole LACP (Link Aggregation Control Protocol).

- 1 réseau accès aux services des machines faisant également office de réseau d'administration.
- 1 réseau data (chaque serveur est configuré avec un agrégat de deux liens (LACP) utilisant un algorithme de distribution basé sur les couches 3 et 4 du modèle OSI.
- 1 réseau de réplication sur les serveurs Ceph.

[14] http://www.atlassian.com

Architecture de la plateforme de test :

L'architecture de la plateforme de test est conçue en deux couches. La première couche est la couche cliente, consommatrice de stockage, basée sur les deux hyperviseurs. Ces hyperviseurs peuvent exploiter le stockage directement par l'intermédiaire de montages RBD ou par des machines virtuelles. Un des serveurs sert également de serveur de déploiement et d'administration du stockage.

La seconde couche est la couche de stockage basée sur Ceph. Cette couche est reliée à la couche cliente par un réseau data. Chaque nœud dispose d'un accès au réseau de réplication. Tous les nœuds disposent de processus OSD, mais seuls les trois premiers nœuds accueillent un processus moniteur.

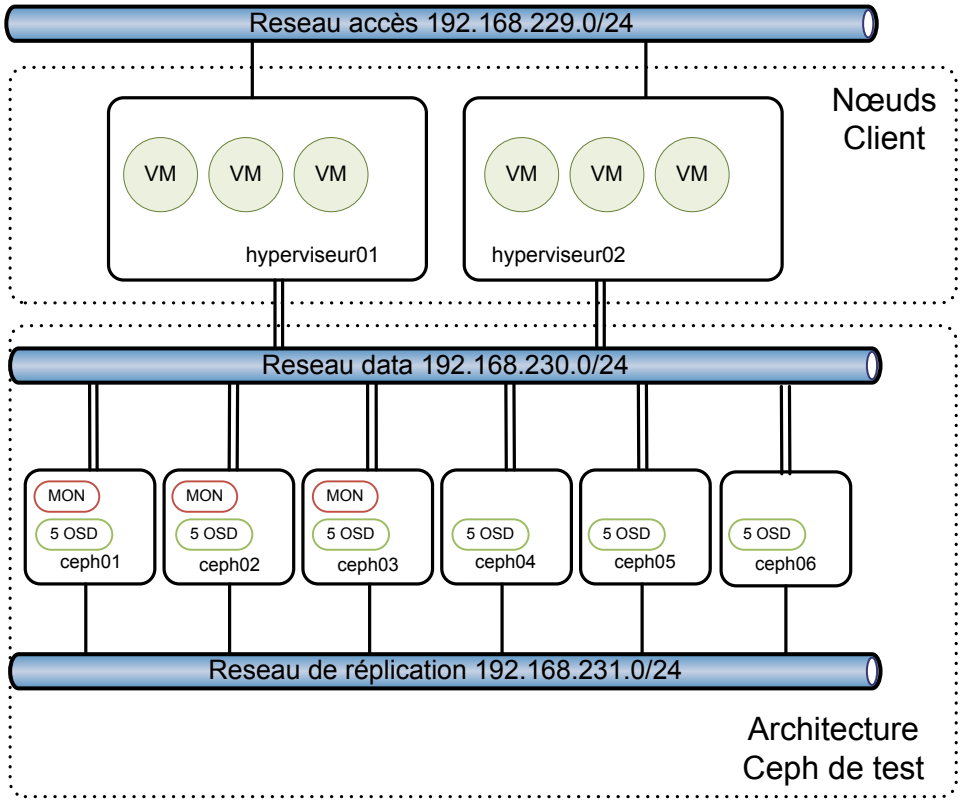

Figure 31 : L'architecture de la plateforme de test

6.10.2. La stratégie de test

Tests fonctionnels et de fiabilité :

La rédaction des cas de tests est réalisée par la personne en charge de l'étude et la réalisation du sous-système testé, nommée **responsable**. Tous les tests sont consignés dans un cahier de tests dans la base documentaire Confluence.

Un **valideur** corrige ou complète les cas de tests et fait approuver les modifications auprès du responsable du sous-système.

Finalement les tests sont exécutés par une tierce personne, le **testeur**, ou par le valideur suivant les disponibilités.

Tests de performance :

Les tests de performance sont séparés en trois parties. Les premiers tests sont consacrés à vérifier les performances intrinsèques de la solution et de ses composants, en soumettant le système à différentes charges. La deuxième phase de test est consacrée à valider la montée en charge du système, en soumettant celui-ci à une charge croissante d'utilisateurs. Enfin, les derniers tests consisteront à valider les performances d'une application dont le stockage constitue un élément critique de fonctionnement.

Les outils utilisés dans les tests de performance sont les suivants :

- Générateur de trafic disque : fio
- Outil de benchmark base de données : pgbench
- Outils d'analyse : collectd, collectl, top, iptraf, vmstat

6.10.3. La gestion des anomalies

Les anomalies mises en évidence lors de l'étape de test sont enregistrées dans l'outil de ticket JIRA pour analyse et sont classées en trois niveaux de gravité suivant leur impact.

Anomalie	Description
Bloquante	Interdit l'utilisation d'une fonction indispensable ou d'un cas d'utilisation
Moyenne	Interdit ou limite l'utilisation d'une fonction non indispensable
Basse	N'affecte pas l'utilisation d'une fonctionnalité

Tableau 18 : Les niveaux de gravité des anomalies

Les anomalies sont ensuite qualifiées puis traitées de deux manières différentes :

- Par une demande d'évolution s'il est nécessaire d'intervenir sur le périmètre fonctionnel ou technique. Suivant le caractère et la complexité de l'évolution demandée, celle-ci peut être acceptée ou rejetée. En cas de rejet, la fonctionnalité peut être abandonnée ou limitée.
- Par une demande de correction en cas d'une non-conformité dans l'étape de réalisation.

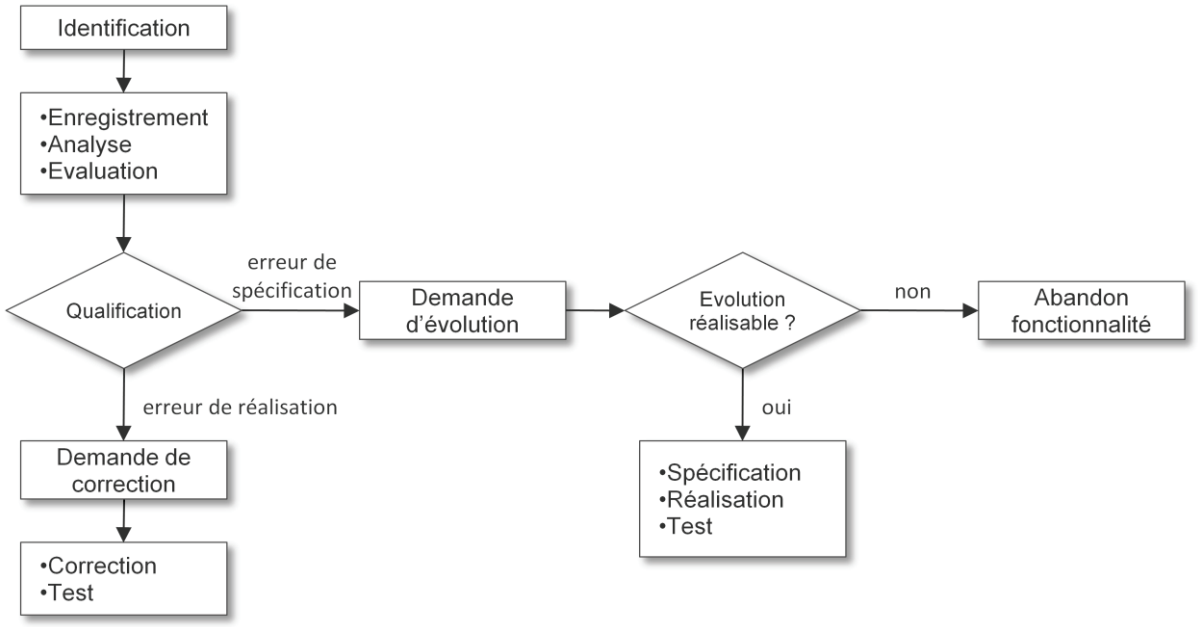

Figure 32 : Le processus de gestion des anomalies

6.10.4. Exemple de fiche de test

La figure suivante illustre un modèle de fiche de test telle qu'enregistrée dans le cahier de tests dans l'outil Confluence.

Identifiant	**Identifiant unique du test**		
Titre	Titre du test		
Objectif	Objectif du test		
Priorité	Priorité du test de 1 à 3		
Scénario :			
Pré-requis	Ensemble des pré-requis nécessaire au testeur pour lancer le cas de test		
Initialisation	Jeu de données, ou dépendance nécessaire au lancement du test		
Tests	**Données d'entrée**	**résultat**	**ok?**
	action 1	résultat de l'action 1	validation action 1
	action 2	résultat de l'action 2	validation action 2
	action 3	résultat de l'action 3	validation action 3
Résultat	Conclusion du test		
Réalisé	le --/--/-- par trigramme du testeur		
Crushmap	Version de la crushmap utilisé		
Statut	Statut du test : non-réalisé, succès, erreur		

Figure 33 : Exemple de fiche de test

Chaque test fait référence à une CrushMap précise stockée dans un outil de gestion de version. Ceci permet de déterminer les conditions exactes du test dans une certaine typologie de cluster.

6.10.5. La gestion des risques

Formation et délai d'apprentissage sur les technologies

Le degré de technicité du projet est important et certaines technologies utilisées sont nouvelles ou peu maitrisées par une partie des membres de l'équipe (SSD, LACP, MLAG, Ceph, etc.). Une étape d'appropriation de ces technologies sera nécessaire et pourra engendrer un décalage dans le planning.

Changement de périmètre, extension des besoins fonctionnels

Au fur et à mesure de l'avancement dans l'étude du système de stockage, de nouveaux besoins peuvent apparaître. C'est notamment le cas avec le stockage en mode fichier et le stockage en mode Cloud qui n'étaient pas prévus au lancement du projet. L'ajout de nouvelles fonctionnalités dans le développement de Ceph peut également soulever de nouvelles opportunités d'usage. Ces changements sont traités au cas par cas, par exemple, en rejetant ou en reportant l'étude de ces nouvelles fonctionnalités si elles sont jugées non critiques, ou en impliquant de nouvelles personnes dans le projet dans le cas contraire.

Indisponibilité des ressources matérielles

Les tests réalisés réclament un matériel spécifique qui n'est pas forcément disponible (3 switchs, modules 10Gb) ou disponible en quantité insuffisante (disques de 1To). Dans ce cas de figure, les tests tenteront de simuler l'usage du matériel manquant ou d'extrapoler les résultats.

Indisponibilité des membres de l'équipe projet

Les membres de l'équipe projet consacrent une partie de leur temps de travail au projet, mais doivent également assumer leurs tâches quotidiennes et la réalisation de leurs autres projets. Il existe un risque important d'indisponibilité d'une partie de l'équipe suivant l'activité des autres projets. Dans ce cas, le projet de stockage pourrait prendre du retard.

7. La réalisation

7.1. La qualification

Après avoir mis en œuvre en plateforme de test, l'environnement Ceph, ainsi que les différents composants constitutifs de l'architecture, l'étape de qualification a pu débuter. Les trigrammes des différents intervenants sont détaillés ci-dessous.

- STH Sébastien Thiaux
- JML Jean-Maxime Leblanc
- MHE Maëlig Herviault
- NRA Nicolas Raux
- VLI Vincent Libé
- CLG Christophe Le Guern
- GKE Guillaume Kerivel

Pour chaque domaine de test (domaine des fonctionnalités, domaine de la fiabilité et domaine des performances), un tableau d'activité détaille les actions assignées à chaque participant. Chaque ligne des tableaux est une section de tests et a fait l'objet d'une étude approfondie, d'une rédaction de documentation et de multiples cas de tests. Les lignes grisées représentent les sections qui n'ont pas été testées à la date de rédaction de ce rapport.

7.1.1. Les fonctionnalités

Les sections de tests du domaine fonctionnel sont détaillées dans le tableau ci-dessous.

Domaine fonctionnel	responsable	valideur	testeur
Fonctionnalités Administration			
Fonctionnalités Administration et déploiement Cluster	JML	STH	MHE
Fonctionnalités Administration Sécurité	JML	STH	MHE
Fonctionnalités Administration Ressources	JML	STH	MHE
Fonctionnalités Cas d'utilisation			
Fonctionnalités Cas Virtualisation	JML	STH	MHE
Fonctionnalités Cas Openstack	CLG / GKE	CLG / GKE	CLG / GKE
Fonctionnalités Cas NFS	MHE	STH	JML
Fonctionnalités Avancées			
Fonctionnalités Avancées CrushMap	JML	STH	STH
Fonctionnalités Avancées Géo-réplication	STH	JML	JML
Fonctionnalités Avancées LiveMigration	STH	JML	MHE
Fonctionnalités Avancées Snapshot	STH	JML	JML
Fonctionnalités Avancées Flashcache	STH	JML	MHE
Fonctionnalités Avancées RBD-cache	STH	JML	JML
Fonctionnalités Avancées Tiering (V0.8)	-	-	-
Fonctionnalités Interfaces Ceph			
Fonctionnalités Interface RBD	STH	JML	MHE
Fonctionnalités Interface CephFS	-	-	-
Fonctionnalités Interface RadosGW	-	-	-

Tableau 19 : Les sections de tests du domaine fonctionnel

Les tests des fonctionnalités d'administration générale du cluster ont prouvé une excellente capacité d'extension de celui-ci tout en conservant une relative aisance d'administration. Seule la sécurité d'accès au réseau de donnés nous parait insuffisante, car elle ne permet pas un chiffrement des communications au sein du réseau data entre les différents clients, ce qui peut poser un problème de sécurité. Nous décidons donc de ne brancher sur le réseau de données que des serveurs dit « de confiance » gérés par BSO, en attendant de trouver une solution à ce problème.

Les différents cas d'utilisation (virtualisation, serveurs de fichier et Cloud Openstack) sont couverts et fonctionnellement aptes à une utilisation en production. Les fonctionnalités avancées, telles que la gestion des snapshots ou la migration à chaud, sont parfaitement opérationnelles. Les fonctionnalités de cache ont montré quelques limitations : en effet, l'utilisation d'un périphérique flashcache rend impossible la fonctionnalité de migration à chaud pour les machines virtuelles. Flashcache ne peut donc être utilisé que pour les

« attachements directs » de volume RBD sur les serveurs tels que le cas d'utilisation en serveur de fichiers.

Parmi les différentes interfaces de Ceph, seul RBD (Rados Bloc Device) a été testé, mais nous n'excluons pas de tester CephFS (stockage fichier) et RadosGW (stockage objet) dans un avenir proche.

7.1.2. La fiabilité

Les sections de tests du domaine de la fiabilité sont détaillées dans le tableau ci-dessous.

Domaine de la fiabilité	responsable	valideur	testeur
Fiabilité Cluster Ceph			
Fiabilité composants MON et OSD	STH	STH	STH/JML
Fiabilité disque	STH	JML	JML
Fiabilité rbd	STH	JML	JML
Fiabilité serveur	STH	JML	MHE
Fiabilité réseau	NRA / VLI	STH	STH/JML
Fiabilité Cas d'utilisation			
Fiabilité Cas NFS	MHE	STH	STH
Fiabilité Cas Openstack	CLG / GKE	CLG / GKE	CLG / GKE
Fiabilité Cas Virtualisation	JML	STH	MHE

Tableau 20 : Les sections de tests du domaine de la fiabilité

Les tests de fiabilité ont eu pour objectifs de vérifier les comportements du cluster et des architectures clientes dans des cas de fonctionnements anormaux. Tous les tests ont été menés sur un cluster actif, avec des opérations de lecture et d'écriture simultanées sur des machines virtuelles. Toutes les pannes générées ont également permis de valider le bon fonctionnement de la supervision du cluster.

Les tests de fiabilité ont passé en revue les différentes couches du cluster :

- Sur la couche logicielle, la fiabilité des composants de Ceph OSD et moniteurs a été testée en effectuant des opérations anormales ou en tuant des processus. Tous les mécanismes de vérification d'intégrité des données (scrub) ont été également vérifiés dans cette section.
- Sur la couche disque, nous avons simulé la perte d'un ou plusieurs disques, l'apparition d'erreurs sur ceux-ci, ou un remplissage critique.

- La couche RBD a été validée en vérifiant le comportement des clients lorsque ceux-ci sont privés de leur stockage distant, par exemple lors d'une coupure réseau.
- La couche serveur a permis de valider que le cluster continuait de fonctionner après la perte entière d'un serveur.
- Les tests de la couche réseau ont pour objectifs de valider le fonctionnement en anneau et de vérifier que le cluster supporte bien la perte entière d'une zone.

Après tous ces tests, qui n'ont eu pour seule limite que l'imagination de l'équipe projet, nous ne sommes pas parvenus à perdre ou à corrompre des données, ce qui témoigne de l'extrême fiabilité du cluster Ceph.

7.1.3. Les performances

Les sections de test du domaine des performances sont détaillées dans le tableau ci-dessous.

Domaine des performances	responsable	valideur	testeur
Tests de performance			
Tests performance native	STH	JML	STH/JML
Tests performance avec cache	STH	JML	STH/JML
Tests de performance optimisations	STH	JML	STH/JML
Tests de montée en charge	STH	JML	STH/JML
Tests de vieillissement	STH	JML	STH/JML

Tableau 21 : Les sections de tests du domaine des performances

Notions générales

IOPS (Opérations d'Entrée/Sortie par seconde, prononcé i-ops) : une mesure commune de performance utilisée pour comparer les systèmes de stockage, disque dur, SSD, baie SAN etc.

Latence : Délai entre une demande d'opération I/O et la réalisation effective de cette opération. La latence influence grandement le nombre d'IOPS pouvant être réalisées.

Débit : Mesure de la quantité de données pouvant être transférées dans un laps de temps précis, généralement une seconde.

Opérations séquentielles : Les opérations séquentielles réalisent des accès sur des blocs de données adjacents du système de stockage. Un cas typique d'opérations séquentielles est rencontré lors de transferts de fichiers.

Opérations aléatoires : A l'inverse des opérations séquentielles, les opérations aléatoires réalisent des accès sur des blocs de données non-contigus. Un cas typique d'opérations séquentielles est rencontré lors d'utilisation de base de données.

Les tests de performance

Objectif et conditions de tests :

L'objectif de ces tests est de vérifier les qualités du système de stockage Ceph par rapport à un système comparable SAN. Les tests utilisent l'outil « fio », lancé depuis le serveur client « hyperviseur01 » pour la partie Ceph, et depuis le serveur « hyperviseur02 », relié en fibre sur la baie NetApp, pour la partie SAN.

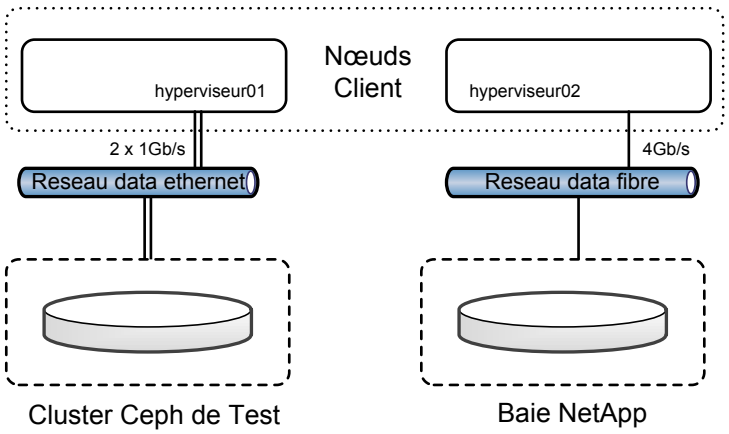

Figure 34 : Architecture tests de performance

Quatre types de tests sont lancés : Les tests séquentiels en lecture et en écriture, où est principalement mesuré le débit, et les tests aléatoires en lecture et en écriture, où sont mesurés le nombre d'opérations par seconde (IOPS) et la latence.

Le modèle de commande « fio » lancé est le suivant :

```
fio -filename=$filename -direct=1 -thread -rw=$rwpolicy -ioengine=$moteur -bs=$bs -
size=4G -numjobs=$num_theads -iodepth=16

filename :   chemin vers le périphérique Ceph
direct=1 :   ne pas bufferiser les IO
rw :         politique de génération : séquentielle lecture ou écriture, aléatoire,
lecture ou écriture
ioengine :   moteur de génération d'IO. Le moteur libaio, moteur asynchrone natif
de linux, est utilisé.
bs : taille des blocs (4ko op. aléatoires, 4Mo op. séquentielles)
size : taille totale des données échangées par test
numjobs : nombre de threads simultanés
iodepth : taille de la pile d'opérations pour le moteur asynchrone
```

Résultats des tests

Les accès séquentiels, en lecture comme en écriture, sont limités par la bande passante réseau disponible entre le serveur client et le cluster Ceph, c'est-à-dire à 200Mo/s. Ceci correspond aux résultats attendus.

L'analyse des accès aléatoires est bien plus intéressante : elle reflète avec plus d'exactitude le trafic qui pourra être rencontré sur le cluster Ceph lorsque plusieurs centaines de clients y seront connectés.

Le graphique suivant illustre le nombre d'IOPS pouvant être réalisé en écriture sur le cluster Ceph en fonction du nombre de nœuds et en fonction de la concurrence. Les accès sont dit « concurrents » lorsque plusieurs clients, ici caractérisés par des threads linux, accèdent de manière simultanée à une même ressource partagée.

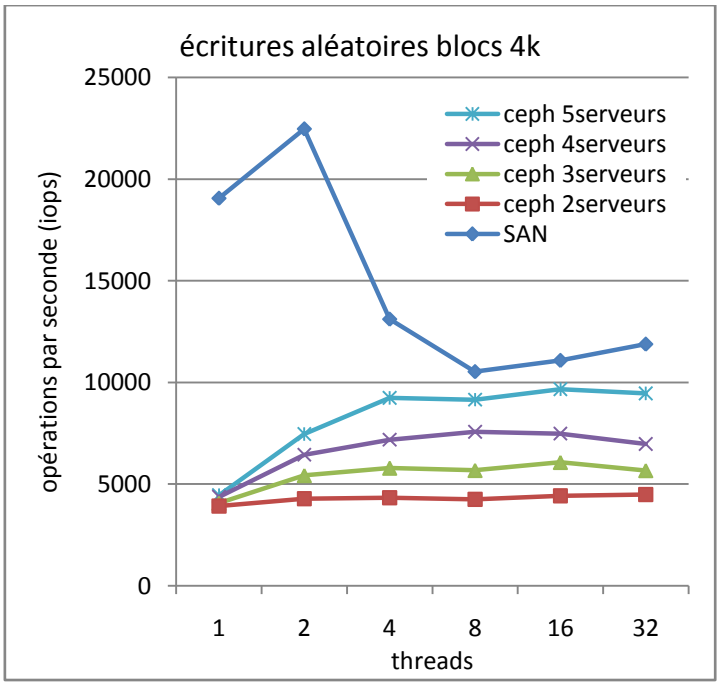

Figure 35 : IOPS en écritures aléatoires

Au sujet du SAN, on peut observer d'excellentes performances si celui-ci n'est pas soumis à une concurrence importante. Le cluster Ceph se contente de performances relativement modestes en faible concurrence, quel que soit le nombre de nœuds, mais prouve une excellente capacité à monter en charge et à traiter les accès parallèles lorsqu'on y rajoute des nœuds.

Le graphique suivant illustre la latence constatée pour le même test, en condition d'écritures aléatoires.

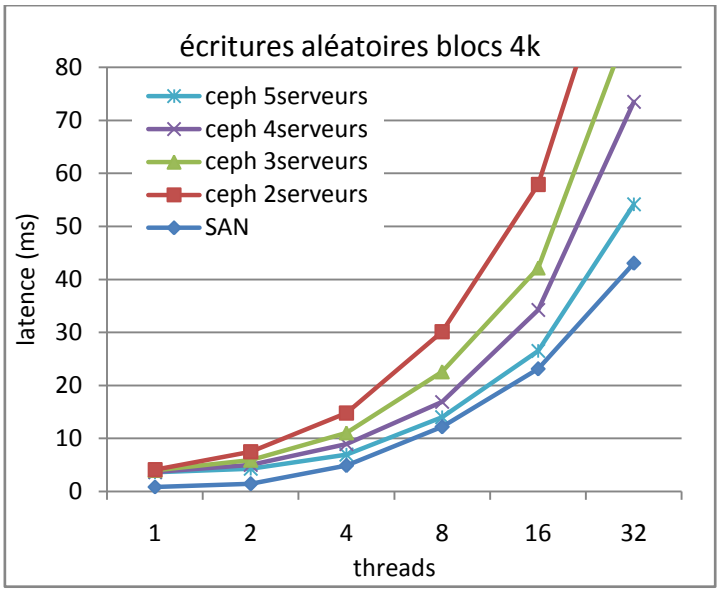

Figure 36 : Latence en écritures aléatoires

Le SAN profite d'une latence très basse de 0.83ms en faible concurrence là ou Ceph débute à 3.5ms, puis les résultats pour Ceph tendent à se rapprocher du SAN sous la concurrence en augmentant le nombre de nœuds.

L'écart initial constaté est provoqué par la latence réseau ainsi que par la nécessité pour Ceph de procéder à des opérations d'écriture multiples : d'abord sur l'OSD primaire pour le premier réplica, puis sur l'OSD secondaire pour le second réplica (dans le cas d'un facteur de réplication fixé à 2) avant d'acquitter l'opération auprès du client.

Influence des caches

Les tests précédents ont été réalisés en cherchant à minimiser autant que possible l'influence des différents caches pouvant masquer les performances réelles. Les tests en écriture, en particulier, ont été forcés pour demander un acquittement sur disque et non en mémoire « buffer » (direct=1). Pour les tests en lecture, le « page-cache » de Linux intervient rapidement au niveau des nœuds Ceph et évite ainsi les accès disque lorsque les données sont déjà en mémoire. Il a donc été nécessaire de purger tous les caches entre chaque test.

Concernant le cache coté client de type « flashcache » placé en amont d'un volume Ceph, les tests ont montré qu'il permet d'obtenir des performances de 20 000 IOPS en écritures aléatoires, constantes de 1 à 32 threads, de quoi satisfaire les applications les plus gourmandes. Par contre, à notre grande surprise, les disques SSD que nous avons utilisés sont bridés à 100Mo/s en écritures séquentielles et chutent en performance sur des accès séquentiels concurrents. Il s'agit pourtant de matériel type « entreprise ». Le problème n'a pas été constaté sur des SSD « grand public ».

Les tests ont également porté sur les paramètres d'optimisation système. En particulier deux facteurs d'optimisation ont été travaillés avec attention. Tout d'abord, la taille des trames réseaux (MTU) a été conservée à sa valeur par défaut (1500 octets), contrairement à certaines préconisations recommandant de l'augmenter à 9000 (le passage à 9000 dégradant les performances en accès aléatoires sur les blocs de 4ko). Le deuxième facteur étudié est l'algorithme de l'ordonnanceur d'entrées/sorties du noyau Linux que nous avons changé de « CFQ » à « deadline », ce dernier mode engendrant de nets gains de performance.

Le test de montée en charge

Objectif et conditions des tests :

Ce test vise à déterminer la charge maximale acceptable sur Ceph lorsque le système est utilisé par un nombre croissant de machines virtuelles soumises à une activité disque importante.

Un injecteur exécute des commandes SQL, grâce à un outil de benchmark PostgreSQL nommé « pgbench », sur un certain nombre de machines virtuelles utilisant Ceph comme base de stockage et ayant chacune une base de données PostgreSQL d'installée. Les transactions exécutées sont basées sur le modèle TPC-B du « Transaction Processing Performance Council[15] ». Dans ce modèle pgbench exécute cinq fois les commandes SELECT, UPDATE et INSERT par transaction.

Au contraire du test de performance réalisé précédemment ce test exécute en parallèle des opérations de lecture et d'écriture.

[15] http://www.tpc.org/tpcb/

La base de données initialisée sur chaque VM par pgbench est une base relativement simple de quatre tables dont l'une possède 10 millions de lignes, ce qui, dans les faits, suffit à engendrer un trafic disque important.

A titre de comparaison, le même test est lancé sur des machines virtuelles utilisant directement les disques de l'hyperviseur.

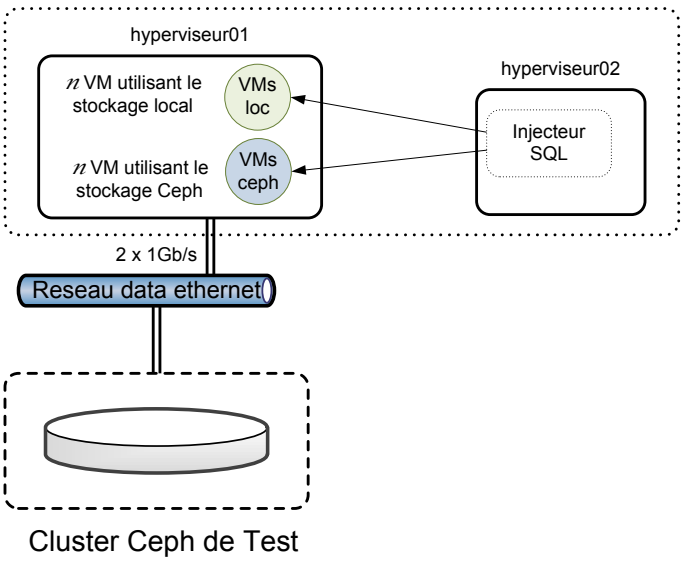

Figure 37 : Architecture tests de montée en charge

Les résultats de ces tests sont présentés dans le graphique ci-dessous :

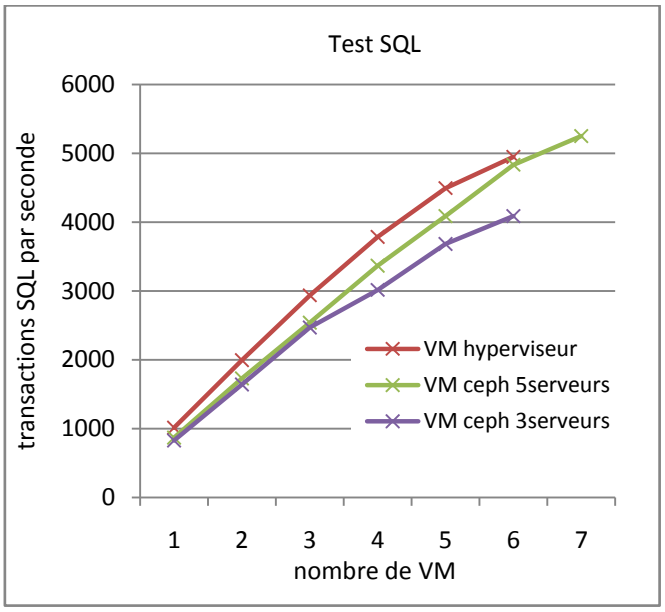

Figure 38 : Tests de montée en charge sur des VM postgresSQL

Résultats des tests

Les tests montrent avant tout une très bonne tenue des performances des machines virtuelles utilisant Ceph, bien qu'elles soient légèrement inférieures à celles utilisant les ressources disque locales de l'hyperviseur. Comme prévu, les résultats avec 5 nœuds Ceph sont meilleurs qu'avec 3. Le test n'est pas parvenu à montrer les limites réelles du système de stockage, en raison d'une saturation de l'activité CPU sur l'hyperviseur à partir de six machines virtuelles.

Le test de vieillissement

Objectif et conditions des tests :

L'objectif de ce test est de valider le fonctionnement et la tenue correcte des performances dans le temps pour une application réelle et fortement consommatrice de ressource disque. Pour cela, nous avons installé sur une machines virtuelle sur Ceph une solution de SIEM (Security Information Management System) de gestion et de corrélation d'évènements de sécurité. Cette machine sert de collecteur de données pour des équipements de sécurité externe (Firewall, logs DNS, logs IPS) et est donc soumise à des contraintes de charges importantes au niveau de la collecte de ces données. La gestion de la corrélation des événements entraine également des contraintes complexes de plus en plus fortes sur la base de données du SIEM au fur et à mesure que la base se remplit.

Résultats des tests

Cette machine a collecté les événements de sécurité d'une zone d'hébergement de PagesJaunes en continue pendant plusieurs semaines, avec des pointes à 700 000 événements par heure. Le SIEM s'est révélé toujours utilisable à l'issue du test, bien que quelques ralentissements non critiques apparaissent par rapport à la même machine installée sur un serveur physique.

Bilan des tests de performance

A la condition de disposer d'un réseau suffisamment dimensionné, les débits en accès séquentiel croissent quasi-linéairement par rapport au nombre d'OSD et peuvent donc atteindre plusieurs dizaines de Go/s sur la globalité d'un cluster. Les accès aléatoires sont certes moins performants que sur du SAN, mais la différence s'amenuise sous des charges

induisant une forte concurrence. Les différents caches coté client (rbdcache et flashcache) permettent, de plus, d'améliorer significativement les performances. Enfin, de nouvelles améliorations sont en cours dans le domaine des performances du côté du cluster. Le système de « cache tiering », inclu dans la version de Ceph sortie en mai 2014, n'a pas encore été testée mais parait prometteur.

7.2. L'industrialisation

Dans le contexte d'hébergement de PagesJaunes avec plusieurs milliers de serveurs, l'industrialisation des solutions est primordiale, et à plus forte raison lorsqu'il s'agit de systèmes critiques. Les objectifs d'une infrastructure industrialisée sont multiples :

En tout premier lieu, l'industrialisation permet de garantir **la disponibilité, la réactivité et l'efficacité** du système en permettant l'installation, la réinstallation ou la configuration rapide de composants. Ensuite, elle permet de garantir **une exploitabilité maximale** grâce à la mise en place d'outils d'automatisation et de procédures évitant les erreurs humaines. Enfin, le dernier objectif, qui est peut-être le plus important : l'industrialisation permet de garantir **une homogénéité des installations et configurations** entre les serveurs et une reproductibilité des opérations d'administration.

7.2.1. L'industrialisation des déploiements Ceph

L'installation d'un serveur Ceph se déroule suivant un processus adapté du processus d'installation général de nos serveurs physiques et utilise en grande partie les mêmes outils.

Dans un premier temps, le serveur Ceph entre dans une phase de préparation matérielle, allant de la mise à jour des différents firmwares matériels à la configuration RAID spécifique de Ceph déjà détaillée. Cette étape se termine par l'installation du système d'exploitation qui, dans notre cas, est Ubuntu Server (distribution Linux dérivée de Debian), grâce à des scripts d'installation automatique (Preseed). L'ensemble de ces étapes est réalisé en PXE (Preboot Execution Environment) en démarrant le serveur Ceph à installer sur un réseau d'installation spécialisé.

Dans un second temps, le serveur Ceph est configuré grâce à des scripts d'installation et de configuration développés spécifiquement dans le cadre du projet pour Ceph. Ces scripts, écrits en langage python, utilisent une librairie nommée « Fabric[16] » servant à faciliter les taches d'administrations sur un ou plusieurs serveurs, en utilisant le protocole de connexion à distance SSH (Secure Shell) pour propager de manière sécurisée des commandes.

Les scripts ont pour fonction de configurer le réseau avec les agrégats de lien, d'installer la base logicielle de Ceph et de la configurer, puis de déployer l'ensemble des outils tels que la supervision, la métrologie etc.

Les scripts Fabric s'appuient sur un outil de gestion de configuration (SVN) permettant de versionner le code et les configurations.

Les scripts sont testés et validés chaque nuit grâce à une chaine d'intégration continue basée sur le logiciel Jenkins[17]. Ainsi, nous nous assurons de la reproductivité des installations et configurations.

Pour ces deux étapes, les logiciels (firmwares, système d'exploitation, Ceph, outils) sont installés depuis des miroirs logiciels locaux à PagesJaunes. Un miroir logiciel local est une copie d'un miroir logiciel distant « officiel » permettant de se prémunir d'éventuelles pannes sur ce dernier.

La figure suivante illustre le processus d'installation d'un serveur Ceph :

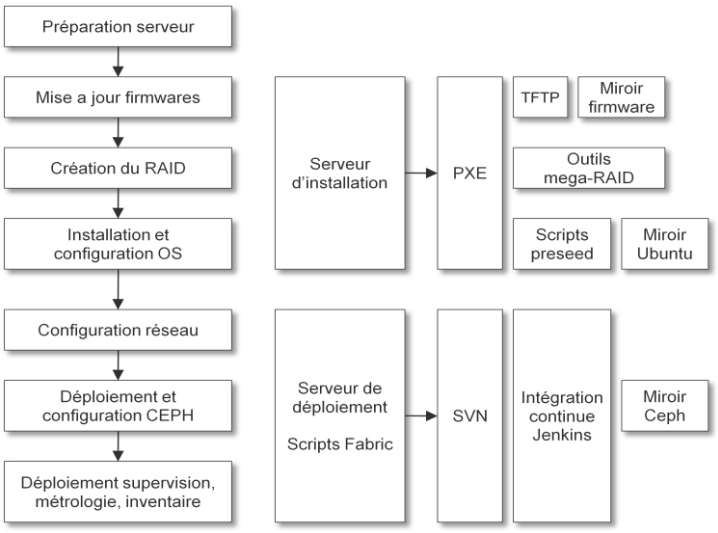

Figure 39 : Le processus d'installation d'un nœud Ceph et les outils associés

[16] http://www.fabfile.org/
[17] http://jenkins-ci.org/

7.2.2. La supervision

La supervision jouant un rôle crucial dans le système de stockage Ceph, ce fut, par conséquent, un des premiers sujets abordés. L'ensemble des tests exécutés sur la plateforme Ceph a ainsi été réalisé avec une supervision active, ce qui a permis de la rendre particulièrement fiable.

Les objectifs de celle-ci sont multiples :

- Vérifier en temps réel l'état de l'ensemble des composants du système Ceph : réseau, matériels, applicatifs…
- Etre immédiatement alertés de tout risque de dysfonctionnement.
- Identifier les problèmes avant que les utilisateurs ne soient affectés.
- Disposer des informations nécessaires à la correction rapide des incidents.

Le schéma suivant illustre le fonctionnement de la supervision entre les nœuds Ceph et le système de supervision « Nagios ».

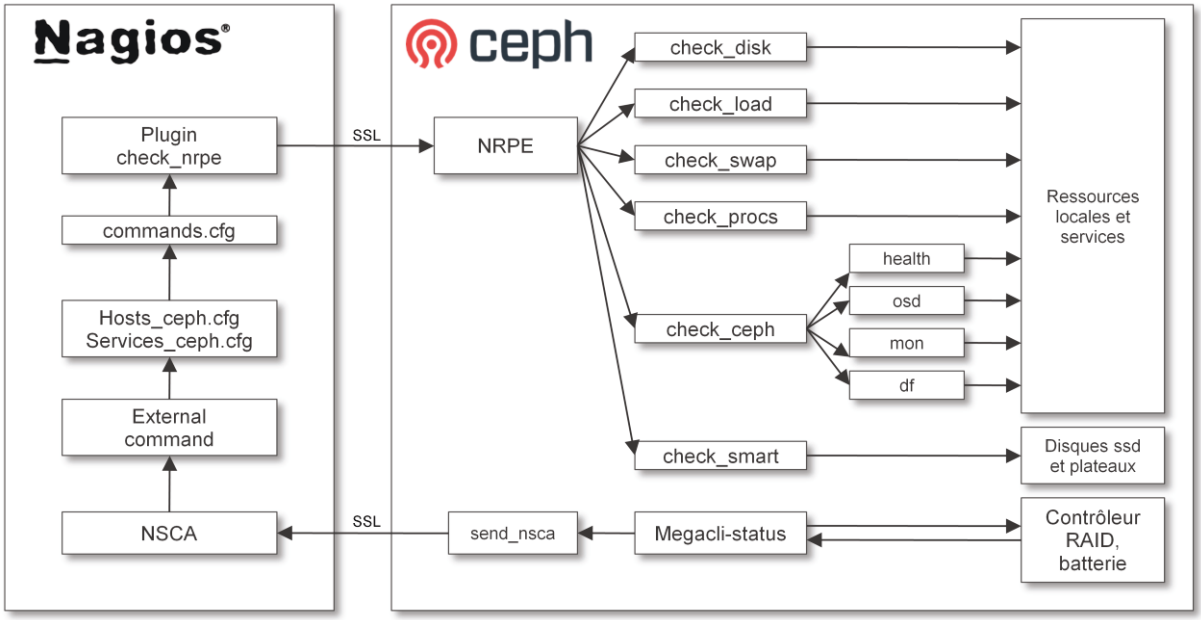

Figure 40 : La supervision des serveurs Ceph

Un agent NRPE (Nagios Remote Plugin Executor) est installé sur chaque serveur Ceph. Cet agent permet de transmettre des informations provenant de différents plugins au serveur de supervision Nagios à la demande de ce dernier.

Trois types de vérification sont réalisés par NRPE :

- **La supervision des ressources locales et des services** : Des alertes sont fixées sur les espaces disques, sur la charge ou sur la présence de processus critiques, etc.
- **La supervision spécifique à Ceph** : Un plugin a été créé par l'équipe projet et permet de vérifier l'état de santé global du cluster, ainsi que l'état de chaque OSD et de chaque moniteur.
- **La supervision des disques** : Afin de contrôler finement l'état de santé des disques, une technique de supervision spécifique, appelée SMART, a été étudiée et implémentée. La technologie SMART (Self-Monitoring, Analysis and Reporting Technology system) permet d'anticiper les pannes disques, de surveiller et d'informer de l'état de certains indicateurs de fiabilité comme la température, le nombre de secteurs réalloués, les erreurs de localisation des secteurs.

La supervision des switchs est assurée par un plugin nagios SNMP. Une carte de visualisation générale de l'activité réseau est générée à partir du module Nagios additionnel nommé « Nagvis[18] »

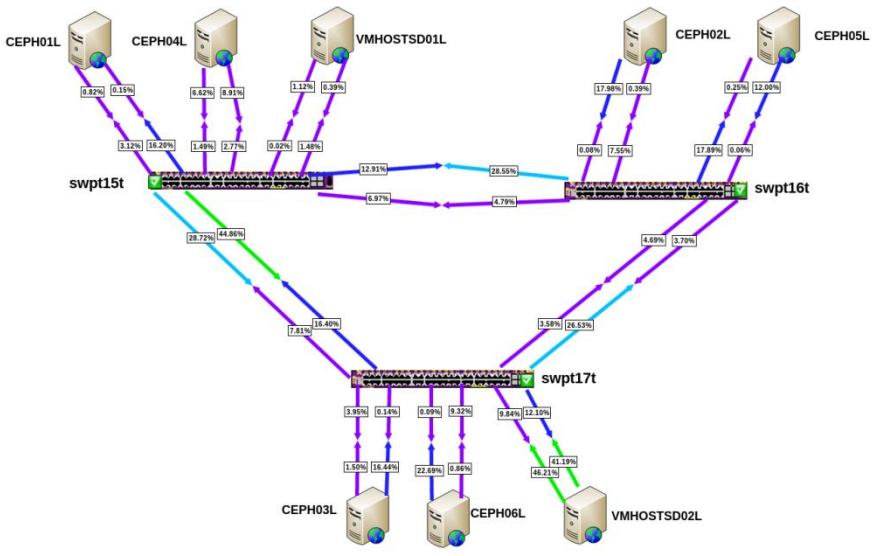

Figure 41 : Visualisation de l'activité réseau sur l'architecture de test avec trois switchs

[18] http://www.nagvis.org/

7.2.3. La métrologie

L'administration du cluster Ceph impose de disposer d'un certain nombre d'indicateurs représentatifs de l'activité du système et de l'évolution de cette activité dans le temps. Disposer d'un système de mesure précis et paramétrable permet de satisfaire les objectifs suivants :

- Mesurer la qualité de service rendue avec des indicateurs clés de performance.
- Anticiper les besoins d'évolution du système en termes de capacité et de puissance.
- Suivre le fonctionnement et analyser à posteriori le fonctionnement en cas d'incident.

La métrologie est assurée par un agent « collectd » installé sur chaque serveur. Les informations de consommation et d'activités mémoire, disque, processeur, sont remontées par cet agent jusqu'à un serveur central collectd qui les agrège sous forme graphique. Un plugin, créé en langage python pour Ceph, permet de remonter les informations du cluster, tels que le débit total, le nombre d'IOPS, le nombre d'erreurs sur les OSD, les erreurs sur les moniteurs etc. En tout, c'est près d'une centaine d'indicateurs pour chaque serveur qui est relevée chaque minute et envoyée au serveur central collectd. Un outil de visualisation graphique développé en interne permet de créer des tableaux de bord pour les indicateurs les plus pertinents afin de rendre les données facilement exploitables.

7.2.4. La plateforme d'intégration

En plus des deux clusters installés en production, un troisième cluster de 3 serveurs d'ancienne génération est installé afin de :

- Réaliser des tests.
- Valider les processus de mises à jour logicielles de Ceph et des composants périphériques (système d'exploitation, firmware, réseaux, etc.).
- Servir de plateforme de formation.

7.3. Etat d'avancement de la réalisation

Le planning fixé initialement a dérivé rapidement lors de l'étape de qualification, ce qui a engendré un retard d'environ un mois. Les trois raisons principales sont les suivantes :

- D'inévitables obstacles techniques sur les nouvelles technologies ont mis plus de temps à être résolus que prévu (problème de LACP, de Jumbo Frame, de SSD, de complexité générale de mise en œuvre de la plateforme de test, etc.).
- La disponibilité des membres de l'équipe était fortement dépendante de l'activité des autres projets. Certains domaines d'étude ont donc pu être décalés sans incidence sur les délais, mais d'autres ont fini par se trouver sur le chemin critique (l'étude de la boucle réseau par exemple).
- Le projet a subi une évolution importante des besoins : Le besoin initial du projet concernait uniquement le cas de « virtualisation » des serveurs. Les autres cas, « Cloud » et « Serveur de fichiers » avaient été considérés comme facultatifs. Ces ajouts de fonctionnalités m'ont conduit à répartir encore plus la charge sur les membres de l'équipe.

Au terme de cette étape de réalisation, la solution est prête pour un déploiement en production. Ce déploiement ne saurait, néanmoins, se faire sur une base de système d'exploitation (OS) non-pérenne. Or, notre système d'exploitation de prédilection, Ubuntu Server 12.04, est en fin de vie dans la version actuellement utilisée en production, et une étape de qualification importante de la nouvelle version (14.04) doit être réalisée par l'équipe système. Cette étape qui devait être réalisée fin avril 2014, a également pris du retard, retard qui se répercute sur le projet de stockage distribué.

Enfin, une nouvelle version majeure de Ceph, sortie mi-mai 2014 (V0.8), apporte de nouvelles fonctionnalités intéressantes (erasure coding - cache tiering) que nous souhaiterions pouvoir qualifier et qui, par conséquent, vont retarder l'étape de mise en œuvre de la solution en production.

Par conséquent, à la date de rédaction de ce rapport, la mise en œuvre de la solution n'a pas pu être réalisée. Néanmoins, à défaut d'être mise en œuvre, elle devra suivre les étapes décrites dans le chapitre suivant.

8. Mise en œuvre de la solution

8.1. Le déploiement

La phase de déploiement en production peut être lancée à l'issue de la phase de qualification.

La première étape, nécessaire avant de déployer la solution, concerne l'acquisition du matériel. La plateforme d'intégration Ceph fait usage de matériel existant, mais il est nécéssaire, pour les clusters des datacenters 1 et 2, de commander du matériel neuf.

Après avoir réceptionné et analysé les propositions commerciales de nos différents fournisseurs, la meilleure d'entre-elles est sélectionnée. Les 15 serveurs de la solution reviennent à environ 60 000 euros HT pour une capacité brute de 43.5To.

L'étape suivante consiste à réceptionner le matériel lors de la livraison puis à piloter les opérations d'installation dans les datacenters, conformément à la définition de l'architecture physique. Les opérations d'intégration physique des serveurs et de raccordement électrique sont réalisées par l'équipe « logistique et plateformes ». L'installation des switchs et le câblage réseau sont réalisés par l'équipe réseau.

Enfin, la dernière étape concerne le processus de déploiement des serveurs Ceph déjà décrit dans le chapitre 7.2 relatif à l'industrialisation. Le matériel des serveurs est configuré, les systèmes d'exploitation sont installés puis la solution Ceph et ses composants sont déployés grâce aux scripts de déploiement automatique « fabric ». Cette étape est réalisée par l'équipe d'exploitation avec l'assistance de l'équipe systèmes.

8.2. La recette

Le but de la recette est d'organiser la réception des livrables, du matériel, de la solution logicielle et des documentations afin de permettre au client de s'assurer de la conformité de la réalisation et de l'aptitude de la solution à fonctionner sur un environnement de production.

Dans le cadre du projet de stockage distribué, l'équipe projet tient lieu de « fournisseur », l'équipe d'exploitation et l'équipe systèmes tiennent lieu de « client ».

Une première phase de recette s'effectue sur la base :

- de scénarios de recette, basés sur les cas d'utilisation et définis par le client.
- de fiches de tests réalisées par le client et couvrant les aspects fonctionnels, la vérification des documentations, la conformité aux normes (déploiement, nommage, scripts, procédures, etc.).

Une deuxième phase de la recette concerne la vérification d'aptitude au bon fonctionnement (VABF). Les objectifs de la VABF sont de contrôler :

- la capacité de la solution à monter en charge et à respecter un niveau de performance suffisant.
- l'exploitabilité et la maintenabilité de la solution dans des conditions réelles de production.

En cas d'anomalie, le client transmet au fournisseur la liste des problèmes rencontrés, ainsi que le détail des actions permettant de reproduire les anomalies.

A l'issue de ces deux étapes, si aucune anomalie bloquante n'est détectée, un PV de recette permet de prononcer la réception et de lancer la migration en production.

8.3. Les formations

La première étape consiste à **identifier les différentes populations** à former afin d'adapter le contenu et les supports de formation. Trois grandes catégories d'utilisateurs apparaissent :

- **Les différentes MOE de PagesJaunes**. Elles sont consommatrices du stockage dans la mise en œuvre de leurs projets. Elles doivent obtenir un niveau de formation général, qui doit rester superficiel dans les concepts techniques de Ceph, mais suffisant pour en comprendre les particularités. Des sessions de présentation avec supports power-point sont adaptées à ces utilisateurs.
- **Les équipes d'exploitation N1 et N2**. Elles sont à la fois utilisatrices de la solution (installation des machines virtuelles, réalisation des mises en production, etc.) et exploitantes de celle-ci. Elles doivent obtenir un niveau de qualification suffisant pour administrer la solution, réaliser les premiers diagnostics d'incident et gérer l'extensibilité en ajoutant des nœuds. Pour ces utilisateurs, des ateliers de formation basés sur des exercices d'administration et de simulation de pannes sont particulièrement indiqués. L'environnement d'intégration peut servir d'environnement de formation.
- **Les équipes Systèmes et Réseau**. Elles sont en charge de l'expertise technique de la solution. Elles ont un rôle de conseil et d'architecte auprès des MOE et doivent donc comprendre les avantages et les limites de la solution afin d'en tirer le meilleur parti. Elles doivent avoir une parfaite connaissance technique des mécanismes internes de Ceph afin de résoudre les pannes les plus complexes ou de diagnostiquer les problèmes de performance. Elles réalisent les mises à jour logicielles de Ceph et suivent l'actualité technique du produit. Pour ces utilisateurs, des ateliers de formation peuvent être montés. Le cahier de tests est une source d'information importante et complète, pouvant servir de document de référence pour la formation.

La seconde étape consiste à **désigner les formateurs, élaborer les modules de formation et à préparer les supports**. Pour cette étape les formateurs seront Jean-Maxime Leblanc et moi-même.

La dernière étape consiste à **organiser les sessions de formation**. Cette étape regroupe les tâches suivantes :

- planifier les sessions de formation
- allouer les ressources nécessaires au déroulement des formations (salles, équipements, etc.)
- évaluer chaque module (avec une population-test)
- dispenser les formations à l'ensemble des populations

- évaluer au fur et à mesure la réussite des formations et améliorer celles-ci le cas échéant.

8.4. Le plan de migration

La migration des données des anciens systèmes vers le nouveau est l'une des étapes les plus critiques et nécessite donc d'être traitée comme un projet à part entière. La migration sur Ceph concernera deux des trois cas d'utilisation déjà décrits dans le chapitre 4.4. Le cas d'utilisation de type Cloud étant un nouveau service, aucune action particulière n'est à prévoir.

Cas	Existant	Cible
Virtualisation	Fichiers image de VM sur disque locaux	VM RBD sur Ceph
Cloud	-	RBD sur Ceph
Serveur de fichier	Fichiers sur NFS NetApp	Fichiers sur nouveau serveurs NFS

Tableau 22 : Tableau existant/cible

L'étape de migration étant particulièrement longue et complexe, elle gagnera à être encadrée par une méthode qualité simple mais reconnue : le modèle PDCA (Plan, Do, Check, Act). L'usage de cette méthode permet d'améliorer la qualité entre chaque itération selon un processus d'amélioration continue.

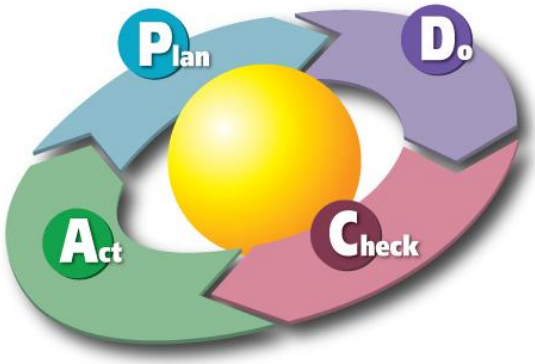

Figure 42 La roue de Deming illustrant le modèle PDCA

Planification (Plan) :

Il convient, dans un premier temps, de cerner avec précision les différents acteurs de ce projet et leurs rôles respectifs.

Dans un second temps, une liste des serveurs à migrer doit être établie. Il convient de migrer d'abord les services les moins critiques, de manière à laisser le système de stockage sous observation quelques mois. Cette mise en production partielle permet de valider le produit en conditions réelles (période de vérification de service régulier (VSR)).

La troisième étape est la préparation de la migration :

- Etablir un planning
- Etablir un plan de communication pour les clients.
- Etablir des fiches de migration par serveur et/ou application (analyse de risque, méthode de migration et de retour arrière, recette)

Réaliser (Do) :

Cette étape consiste à réaliser les actions prévues dans l'étape de planification avec en particulier la réalisation de migration de services sur Ceph.

Contrôler (Check) :

Cette étape a pour objectifs de contrôler les ressources mises en œuvre dans l'étape de réalisation et de vérifier que les résultats obtenus correspondent bien à ce qui a été prévu dans l'étape de planification.

Ajuster (Act) :

Enfin, la dernière étape consiste à ajuster les écarts, vérifier que les solutions mises en place sont efficaces, rechercher des points d'amélioration. Après chaque étape de migration, il est important de rassembler l'équipe et de capitaliser sur l'expérience vécue en partageant les problèmes rencontrés et les solutions trouvées. L'identification des causes de non performance et la mise à jour des processus et procédures permettent d'améliorer le processus général.

9. Conclusion

9.1. Bilan projet

L'objectif de ce projet était d'identifier, d'étudier et de mettre en œuvre une technologie de stockage adaptée aux fortes variations de volumétrie tout en restant flexible et peu onéreuse. Cet objectif a été pleinement rempli par l'utilisation d'une technologie innovante basée sur le stockage logiciel : Ceph. Afin de répondre aux différents besoins de PagesJaunes, cette solution a dû être intégrée dans différents types d'architectures en gardant toujours en mémoire les impératifs de fiabilité et de performance.

9.2. Perspectives

La fiabilité intrinsèque de la solution de stockage étudiée permet d'envisager une baisse généralisée des coûts matériels, tant au niveau du stockage qu'aux niveaux des serveurs utilisant ce stockage. En effet, puisque la redondance des données est gérée par logiciel, il n'est plus nécessaire de disposer de matériel serveur à la pointe de la fiabilité avec des contrats de maintenance coûteux. Une réflexion globale sur le matériel serveur sera engagée à partir du deuxième semestre 2014, avec pour objectif, à court terme, une réduction de coût d'environ 30%.

Par ailleurs, cette nouvelle solution de stockage s'inscrit dans une stratégie à plus long terme. En effet, les avancées technologiques réalisées sur les technologies du Cloud ouvrent de nouvelles perspectives d'innovations. Dans l'objectif de poursuivre sa transformation numérique, PagesJaunes s'inscrit dans cette dynamique de changements technologiques, dans laquelle, le stockage logiciel est l'un des piliers.

9.3. Bilan personnel

La réalisation de ce projet a été pour moi une expérience extrêmement enrichissante sous plusieurs aspects :

Tout d'abord, le contexte d'étude varié et les nombreuses technologies utilisées m'ont permis d'évoluer dans un environnement passionnant pendant toute la durée du projet. Les problèmes ont été nombreux et parfois complexes, mais des solutions ont toujours pu être trouvées.

Ensuite, cette expérience en tant que chef de projet a été l'occasion de mettre en œuvre beaucoup de techniques liées à la gestion de projet apprises au CNAM. Parmi ces techniques, on peut retrouver l'ingénierie des besoins, la qualification, la gestion des risques, la sécurité, la qualité, etc. J'ai pu compléter ces connaissances par des recherches personnelles et en consultant des ouvrages spécialisés dans la gestion projet.

Enfin, les aspects humains ont joué un rôle particulièrement important puisqu'il a fallu fédérer trois équipes (systèmes, réseau et exploitation), qui travaillent souvent séparément, autour d'un objectif commun. Cette expérience de management a été extrêmement positive et a abouti à une collaboration étroite et constructive entre équipes.

Pour conclure, la réalisation de ce mémoire a été pour moi une expérience technique et personnelle formidable qui met un terme à un cursus CNAM extrèmement enrichissant.

Liste des figures

Liste des tableaux

Références Bibliographiques

Comité Européen de Normalisation, 2007. *Best Practices for the Design and Development of Critical Information Systems*, 69 p.

CONSTANTINIDIS, Y., 2010. Expression *des besoins pour le SI : Guide d'élaboration du cahier des charges.* Editions Eyrolles, Paris, 240 p.

EMC Education Services, 2012. *Information Storage and Management: Storing, Managing, and Protecting Digital Information in Classic, Virtualized, and Cloud Environments, 2nd Edition,* John Wiley & Sons, 489 p.

ENGLENDER, O., FERNANDES, S., 2012. *Manager un projet informatique : Comment recueillir les besoins, identifier les risques, définir les coûts ?,* Editions Eyrolles, Paris, 360 p.

GILBERT, S., LYNCH, N., Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services, 13 p

WEIL, S., BRANT, S., MILLER, E., MALZAHN, C., 2006. *CRUSH: Controlled, Scalable, Decentralized Placement of* Replicated *Data*, University of California, Santa Cruz, 12 p*, http://ceph.com/papers/weil-crush-sc06.pdf*

WEIL, S., 2007. *Ceph : reliable, scalable, and high-performance distributed storage.* Thèse Doctor of philosophy in computer science, University of California, Santa Cruz, 221 p, *http://ceph.com/papers/weil-thesis.pdf*

Sites internet :

CEPH, *http://ceph.com/docs/master/*
GLUSTER, *http://www.gluster.org/*
HAN, S., *http://www.sebastien-han.fr/*
DACHARY, L. *http://dachary.org/*
SHEEPDOG, *http://sheepdog.github.io/sheepdog/*

Résumé

L'explosion de la volumétrie des données dans les systèmes informatiques impose de repenser les modèles de stockage traditionnels vers une approche maximisant la flexibilité et l'extensibilité.

L'amélioration des performances du matériel informatique standard permet aujourd'hui d'envisager de nouvelles alternatives. Ce mémoire porte sur l'étude des nouvelles technologies de stockage dites « logicielles » et leurs adaptations dans le contexte d'hébergement d'applications Internet de PagesJaunes.

Après avoir mené une étude comparative sur trois produits et sélectionné le plus adapté, ce mémoire se penche sur l'architecture de la solution retenue et son intégration dans le système d'information de PagesJaunes.

Des tests approfondis du système ont démontré d'excellentes qualités en termes de fiabilité et de performance. Les résultats obtenus permettent d'envisager un déploiement et une utilisation à grande échelle qui permettra à PagesJaunes de répondre aux nouveaux défis de l'hébergement d'applications web.

Mots clés : stockage - stockage distribué - stockage logiciel - SDS - logiciel open source - Cloud

Abstract

The explosive growth in the volume of data in computer systems requires rethinking traditional models of storage for an approach that maximizes flexibility and extensibility.

Recent performance improvements of standard hardware allow us to consider new alternatives. These alternatives focus on new storage technologies called "software defined storage" and the adaptation of this in the context of hosting Internet applications at PagesJaunes.

After conducting a comparative study on three products and selecting the most suitable, this paper focuses on the architecture of the solution and its integration into the information system of PagesJaunes.

Extensive tests of the chosen system have revealed excellent qualities in terms of reliability and performance. These impressive results could lead to widespread deployment and use that will allow PagesJaunes to rise up to future challenges of web hosting applications.

Key words: storage - distributed storage - software defined storage - SDS - open source software - Cloud