Rapport de Projet sur les systèmes de fichiers distribués : Ceph et Lustre

Y. Laprévotte, R. S. M. Rodriguez-Garcia, J. Schneider, J. Lutz

24 mars 2014

Table des matières

1 Introduction

2 Les systèmes de fichiers distribués

3 Ceph

3.1 Présentation

Ceph est un système de fichier open-source, développé à partir d'un algorithme de nouvelle génération appelé CRUSH [1]. Il est évolutif (scalable) et est capable de fonctionner sur un parc de machines très diverses, on appelle ces parc de machines un cluster Ceph. Son fonctionnement repose sur 3 types de serveurs :
- les OSDs [2], qui sont les serveurs de stockage. Les données seront donc enregistrées sur ces nœuds. Il est préférable d'avoir une bonne quantitée d'espace disque, car les OSDs journalisent les opérations d'écritures, ce qui prend de la place de stockage. Il faut au minimum 2 OSD pour que le cluster Ceph soit opérationnel.
- Les Monitors (Ceph Monitor), ceux-ci sont en place pour surveiller que tout fonctionne correctement. Lorsque l'on dispose d'un réseau conséquent en nœuds, il est important de savoir très rapidement quand il y a un dysfonctionnement quelque part. Il est pertinent d'avoir plusieurs Monitors dans le cluster Ceph pour permettre de repérer plus rapidement les erreurs et avoir un bon niveau de tolérance aux pannes. Les monitors sont des daemons relativement léger et ne nécessitent pas une grande quantité de mémoire ou d'espace disque.
- Le troisième type de nœud est le MDS (MetaData Server). Celui-ci détient toutes les informations permettant de trouver les données demandées et leurs attributs. Ces informations sont très souvent consultées, elle sont dont stockées en mémoires pour en amélioré l'accès. Il faut donc une grande quantité de RAM sur les ordinateur qui hébergent les MDS.
Attention il n'est pas encore recommandé de l'utiliser en production : celui-ci est encore en phase de développement.

1. Controled Replication Under Scalable Hashing
2. Object Storage Deamon

3.2 Architecture du cluster

Nous avons choisit d'installer une configuration nous permettant de tester les performances de Ceph. Comme dit précédemment, Ceph a besoin de 3 types de nœuds différents. Nous avons choisit d'utiliser 6 machines pour 9 noeuds, ces machines seront utilisé sous Debian :

-La première machine sera l'Admin node, elle contiendra aussi le premier monitor et le premier OSD :
nom machine : golem
ip : 192.168.1.51
- La deuxième machine contiendra le deuxième Monitor et le second OSD :
nom machine : rondoudou
ip : 192.168.1.29
- La troisième machine contiendra le troisième Monitor et le troisième OSD :
nom machine : behemot
ip : 192.168.1.56
- La quatrième machine contiendra le premier MDS :
nom machine : carapuce
ip : 192.168.1.43
- La cinquième machine contiendra le deuxième MDS :
nom machine : smogogo
ip : 192.168.1.2
- La sixième machine contiendra le troisième MDS :
nom machine : porygon
ip : 192.168.1.48

Nous avons également dû installer les clients qui utilise ce cluster nous les avons installé sur nos machines personnelles.

3.3 Installation

Ceph a été crée pour fonctionner sur du matériel de base, pas besoin d'avoir de grosses configurations ou de matériels spécifiques pour faire tourner ceph, ce qui rend la construction et le maintien d'un cluster de données de l'ordre du pétaoctets facilement et économiquement réalisable. Pour l'installation de Ceph, il faut d'abord configurer le matériel que nous allons utiliser.

Recommandation Matérielles :

CPU : MDS : Les serveurs de métadonnése redistribuent dynamiquement leurs metadonnées, ce qui utilise une grande quantité de puissance CPU. Les MDS doivent posséder un assez bon processeur (quad-core ou mieux). Osd : Les serveurs de données ceph font tourner le service RADOS [3], calculent le placement des datas avec CRUSH, repliquent les données, et maintiennent des copies de la carte (map ?) du cluster. Les serveurs de données ont un besoin raisonnables de puissance CPU. Monitor : les monitor n'ont pas besoin de puissance CPU, ils

3. Reliable Autonomic Distributed Object Store

ne font que maintenir une copie de la carte du cluster pour repérer les erreurs dans le cluster.

RAM : La ram est surtout utilisée par les serveurs de métadonnées et les monitors, car ils doivent être capable de parcourir leur données rapidement, ils doivent donc avoir une bonne quantité de RAM. Il est recommandé d'avoir 1 GB de RAM par daemon. Les Osds n'utilisent pas beaucoup de RAM pour leurs fonctionnements normal, mais ils utilisent plus de RAM si l'on lance une récupération de données. Les disques de stockage doivent également être choisis en fonction des temps d'accès aux données (préférence pour du disque

3.4 Test de disponibilités

3.5 Test de performances

4 Annexe

4.1 How to install ceph

Partitionnement :
En premier, on a partitionné les machines sur lesquels un OSD sera installé, nous avons créer de nouvelle partition en xfs pour stocker les données du cluster avec le logiciel Gparted. Sur chaque machines nous avons fait une partition de environs 10 Go.
Configuration du Réseau :
Sur chaque machine nous avons modifié le fichier /etc/hosts ajoutant l'alias et l'adresse IP de chaque machine ainsi elles peuvent facilement se connecter entre elles en indiquant ses alias.
Fichier /etc/hosts :
#Ceph cluster
192.168.1.51 golem
192.168.1.29 rondoudou
192.168.1.56 behemot
192.168.1.43 carapuce
192.168.1.2 smogogo
192.168.1.48 porygon
Création d'utilisateur ceph
Ceph nécessite un utilisateur spécial pour la configuration et l'administration du cluster à partir de la machine d'administration, nous avons crée l'utilisateur ceph avec les droits d'administrateur du système.
sudo useradd -d /home/ceph -m ceph
fichier /etc/sudoers :
ceph ALL = (root) NOPASSWD :ALL
Configuration ssh Pour effectuer la gestion du cluster, les machines doivent se communiquer entre elles avec des tunnels ssh, avec l'utilisateur ceph il faut générer les clés publiques pour s'identifier avec les autres machines.
su ceph

ssh-keygen
Copier les clés sur tous les autres postes.
ssh-copy-id ceph@nomdemachine
Modifier le fichier /.ssh/config pour se connecter par les tunnels ssh avec l'utilisateur ceph par défaut.
Fichier config :
Host golem
User ceph
Host rondoudou
User ceph
Host behemot
User ceph
Host carapuce
User ceph
Host smogogo
User ceph
Host porygon
User ceph
Synchronisation de l'heure des machines avec NTP :
Afin de prévenir un décalage d'horloge entre les nodes du cluster nous avons installé le serveur NTP sur golem qui nous permet de synchroniser l'horloge de toute les nodes :

ceph@golem : sudo apt-get install ntp

ceph@golem : sudo /etc/init.d/ntp restart

Nous avons également installé lsb sur toute les machines, il permet de standardiser la structure interne des systèmes d'exploitation basés sur GNU/Linux :
ceph@golem : sudo apt-get install lsb
Sous Debian, l'installation de Ceph est simple car les développeurs mettent régulièrement les paquets à disposition mais les sources le sont autant. La première étape consiste à rajouter les dépôts de Ceph pour apt-get. Pour ce faire, rajoutez les deux lignes suivantes à la fin de votre fichier '/etc/apt/sources-list' :
deb http ://ceph.net/debian/ wheezy main
deb-src http ://ceph.net/debian/ wheezy main
Seconde étape, mettre a jour apt-get pour la prise en compte de ces nouveaux dépôts :

[ceph@golem]sudo apt-get update

Enfin, nous avons utilisez apt-get pour installer les paquets :

[ceph@golem] apt-get install ceph ceph-deploy

On a ensuite créer un répertoire de travail, il est utilisé par l'Admin Node, nous avons créer ce répertoire dans le dossier courant de l'utilisateur ceph :

[ceph@golem] mkdir cluster-cheese

[ceph@golem] cd cluster-cheese

A partir de maintenant toute les commandes utilisé pour créer le système de fichier distribué devrons être lancer dans ce dossier, sinon il est possible que l'utilisation d'une de ces commandes créer un deuxième cluster et des problèmes surviennent.

Création cluster et installation des monitors :

Nous allons commencer à installer les différents nodes de notre cluster ceph avec la commande :

ceph-deploy new golem rondoudou behemot

cette commande permet de déclarer les nodes du cluster.

Ensuite il y a l'installation de ceph sur ces nodes : Pour lancer cette commande nous avons dû rajouter l'option –no-adjust-repos qui permet de passer les toutes les modifications du proxy sur le paquet et ira directement à l'installation du paquet :

ceph-deploy install –no-adjust-repos rondoudou behemot

(ceph est déja installé sur golem car c'est aussi notre Admin Node.)

Nous passons maintenant à l'installation des monitors, pour cela nous allons créer 1 monitors sur chacun des nodes présents (golem, rondoudou, behemot) avec la commande :

ceph-deploy mon create-initial

cette commande génère de nouveau fichier dans notre répertoire cluster-cheese :

[ceph@golem]ls

ceph.conf ceph.log ceph.mon.keyring

si on regarde ceph.conf :

fichier ceph.conf $[global] fsid = 10c95f01-2dd2-4863-affa-60c4eafcd8d2 mon_initial_members = golem, rondoudou, behemot mon_host = 192.168.1.51, 192.168.1.29, 192.168.1.56 auth cluster required = cephx auth service required = cephx auth client required = cephx osd_journal_size =$
1024

On vois que nos 3 monitors ont été ajouté dans la configuration du cluster.

Installation des Osds

Pour l'installation des OSD nous avons créé une partition de type xfs sur les machines golem, rondoudou et behemot, nous avons utilisé la commande :

ceph-deploy disk list <nomdemachine>

Qui permet de voir les partitions et leurs système de partitionnement sur les différents nodes.

Ensuite depuis golem on a formaté ces partitions avec les commandes :

ceph-deploy disk zap golem :sda3 ceph-deploy disk zap rondoudou :sda3 ceph-deploy disk zap behemot :sda6

Après avoir formaté les partitions nous avons préparé et activé les Osds :

ceph@golem : /cluster-cheese :ceph-deploy osd prepare golem :sda3

ceph@golem : /cluster-cheese :ceph-deploy osd activate golem :sda3

ceph@golem : /cluster-cheese :ceph-deploy osd prepare rondoudou :sda3

ceph@golem : /cluster-cheese :ceph-deploy osd activate rondoudou :sda3

ceph@golem : /cluster-cheese :ceph-deploy osd prepare behemot :sda3

ceph@golem : /cluster-cheese :ceph-deploy osd activate behemot :sda3

(Pour certaine de ces commandes nous avond du rajouter l'option –overwrite-conf, pour modifié la configuration de ceph sur les nodes, ex : ceph-deploy –overwrite-conf osd prepare golem :sda3)

Après avoir installé les Osds nous pouvons déjà regardé si ceph est installé correctement avec la commande :

ceph status

<image ceph status sans mds>

Installation MDS

Nous arrivons à la dernière partie de l'installation où nous allons installé les mds avec la commandes :

ceph-deploy mds create carapuce smogogo porygon

et nous voulons 2 mds d'actif, pour l'instant il y en a un d'actif de base on utilise la commande suivante :

$ceph mds set_max_mds 2$

Ensuite nous avons fait un ceph status pour voir si le système ceph était bien installé : <images console ceph status final>

Nous avons un système ceph fonctionnel, qui nous permet de réaliser des test de disponibilité et de performance. Avec la commande :

ceph osd lspools

Nous pouvons voir qu'il y a 3 "piscine" sur notre cluster ceph, nous allons utilisé la pool rbd.

Installation client Pour l'installation du client, on a d'abord installé ceph sur la machine-client à partir de golem (Admin Node) :

ceph-deploy install ceph-client

On a ensuite copier la configuration de notre cluster de golem vers le client avec la commande : ceph-deploy admin ceph-client

cette commande à copier le ceph.conf et le ceph.client.admin.keyring dans le dossier /etc/ceph sur le ceph-client.

Nous allons créer dans notre piscine rbd un nouveau bloc qui nous permettra de stocké des données, nous créons ici un bloc de 10 Go dans la piscine rbd

rbd create foo –size 10096 –pool rbd

Nous pouvons voir avec :

rbd ls rbd

et rbd –image foo -p rbd info

que notre bloc a bien été créer.

Et maintenant nous avons fait la commande :

rbd map foo -pool rbd

pour ajouter à la map de rbd le nouveau bloc

on peut le voir avec la commande rbd showmapped

on a ensuite mit un système de fichier sur le bloc : mkfs.ext4 -m0 /dev/rbd/rbd/-foo

et nous avons finalement monter le bloc sur le système client avec : $mkdir /mnt/rbd_foo mount /dev/rbd/rbd/f$

A partir d'ici nous pouvons écrite/lire des fichiers sur notre système de fichier distribué ceph à partir d'un client et réaliser les test de performances.