Schneider
Rodriguez Garcia
Laprevotte
Lutz

Projet Tutoré

Systèmes de fichiers distribués

2013/2014

Sommaire

Introduction

La mission de notre projet se divise en deux. La première étape est d'installation deux systèmes de fichiers distribués : Ceph et Lustre. La deuxième est la mise en place de tests de performances afin de comparer leurs performances. Un système de fichiers, ou filesystem (fs) en anglais, est une couche indispensable pour qu'un système d'exploitation puisse fonctionner. Cette couche répond à une problématique : Comment sauvegarder les données sur un support donné, autrement dit, comment organiser l'information dans la mémoire ? Il en existe un certain nombre, chacun dispose de ses avantages et de ses inconvénients. A noter que les plus vieux filesystem sont obsolètes comme le FAT12 ou l'ext1car ils ne correspondent plus vraiment aux besoins actuels.

Ceph est un filesystem open-source, développé à partir d'un algorithme de nouvelle génération appelé CRUSH. Il est massivement scalable et est capable de fonctionner sur un parc de machines très diverses. Son fonctionnement repose sur 3 types de serveurs : les OSDs (Object Storage Deamon), qui sont les serveurs de stockage. Les données seront donc enregistrées sur ces nœuds. Après les OSDs, il est primordiale d'installer un Monitor. Celui-ci est 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 soucis quelque part. Le troisième type de nœud est le MDS (MetaData Server). Celui-ci détient toutes les informations permettant de trouver les données demandées. Attention il n'est pas encore recommandé de l'utiliser en production : celui-ci est encore en phase de développement.

Lustres est aussi un système de fichier open-source créé par Peter Braam en 1999, distribué sous la licence GPLv2. Il est utilisé par 6 des 10 supercalculateurs les plus puissant du monde. Il garantit d'excellentes performances tout en profitant d'une très haute disponibilité. Il peut être introduit dans une un réseau avec des dizaines de milliers de postes clients et des centaines de serveurs.

L'informatique évolue constamment et se complexifie. Ainsi, Il est désormais possible de partager des fichiers via le réseau internet. Cette technologie nécessite donc un type de filesystem bien particulier. C'est ce qu'on appel un système de fichiers distribué. Ceph et Lustre font partie de cette catégorie.

Il existe d'autres systèmes de fichiers équivalent comme NFS, Network File System. Créé par Sun Microsystem en 1984, ce système de fichier permet le partage de données entre machines Unix. GlusterFS est un autre filesystem assez répandu. Il est capable de gérer des petabytes de données et il fonctionne sur des relations client/serveur. Il est distribué sous licence GPLv3. MooseFS est aussi sous GPLv3. Il est disponible sous Linux, FreeBSD, OpenSolaris et MacOSX. Il respecte la norme POSIX et il est composé de 3 types de serveurs différents comme Ceph. Voici un tableau de comparaison des systèmes de fichiers selon différents critères :

	Gluster	Moose	Ceph	NFS
Facilité de mise en place	++	+	+	++
Fiabilité	++	++	-	++
Sécurité, disponibilité des données	+	++	++	- -
Évolutivité	+	++	++	- -
Économe en taille disque	++	-	-	++

Tableau issue d'un document du Loria.

Choix des machines

Nous avons choisit d'utiliser une configuration minimale pour tester Ceph. Comme dit précédemment, Ceph a besoin de 3 types de nœuds différents. Nous avons donc choisit 3 machines pour 4 noeuds : 2 OSDs (un unique OSD n'aurait pas eut d'intérêt) sur 2 machines différentes. Le moniteur a été installé sur une des deux machines précédentes. Ce procédé ne pose pas de problème surtout que le monitoring ne demande pas beaucoup de ressources. Il reste le métadonnée. Celui-ci a été installé sur la 3ième machine. Nous avons choisit de réserver notre ordinateur le plus puissant au MDS. En effet celui-ci peut s'avérer très gourmand dans le cas d'une forte activité. Nous avons aussi fait ce choix en raison de praticité.

Installation de ceph

Pour la 1ére installation de ceph que nous avons suivi un tutoriel où l'on déploie ceph sur 3 nodes:
avec 2 monitors et 2 osd sur 2 monitor et 1 mds .

Configuration du réseau

Nous avons modifié le fichier /etc/hosts sur toutes nos nodes, pour attribuer des nom à chaque machine avec leurs adresses IP

fichier:

```
#Ceph cluster

192.168.1.51        golem
192.168.1.29        rondoudou
192.168.1.43        carapuce
```

Nous avons vérifié la connexion entre les nodes en faisant des pings sur chaque machine:

Configuration du système

Synchronisation de l'heure sur toutes les nodes avec NTP server :

ntp est un protocole permettant de distribuer l'heure sur un réseau informatique.

Commandes:

```
$sudo apt-get  install ntp
$sudo service ntp restart
```

Tous les nodes ont été mise à jour

```
$sudo apt-get update
```

et lsb a été installé sur tous les nodes

```
$sudo apt-get install lsb
```

lsb permet de standardiser la structures internes des systèmes d'exploitation sur gnu/Linux .

Un utilisateur ceph a été créer sur chaques nodes

```
$sudo adduser -d /home/ceph -m ceph
$sudo passwd ceph
```

on lui a donnée les privilége root, dans le fichier /etc/sudoers nous avons ajouté:

ceph ALL = (root) NOPASSWD:ALL

Configuration ssh

$ssh-keygen

$ssh-copy-id ceph@rondoudou

$ssh-copy-id ceph@carapuce

et la modification de ~/.shh/config, pour avoir une connexion automatique avec l'utilisateur ceph sur chaque node

fichier config

```
Host golem
        User ceph
Host rondoudou
        User ceph
Host carapuce
        User ceph
```

Installation de ceph

Installation de ceph sur le premier serveur(golem)

wget -q -O- 'http://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc' | sudo apt-key add -

echo deb http://ceph.com/debian-dumpling/$(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list

sudo apt-get update

Installation du monitor initial

installation de ceph-deploy sur golem

$apt-get install ceph-deploy

création de l'espace de travail

$mkdir cluster

$cd cluster

A partir de maintenant la plupart des commandes vont être lancé à partir du cluster (golem), il faut une bonne configuration du réseau sinon l'installation du système de fichier échouera,

Déclaration du premier node

$ceph-deploy new golem

Installation du premier node

```
$ceph-deploy install  golem
```

On a dû utiliser l'option –no-adjust-repos pour ne pas avoir de problème avec le proxy, et il manquait la clé release.asc, nous l'avons ajouté directement sur la machine avec la commande

```
$sudo apt-key add release.asc
```

Création du monitor initial

```
$ceph-deploy mon create golem
```

il y a maintenant de nouveau fichier dans le cluster que nous avons créé

- ceph.conf: contient la configuration de base de notre cluster
- ceph.log: le journal d'erreur du cluster
- ceph.mon.keyring: la clé du monitor initial

Nous avons regroupé les clés, elle vont être utilisé par toutes les nodes du cluster pour s'authentifier:

```
$ceph-deploy gatherskeys golem
```

On a modifié /etc/ceph/ceph.conf, pour mettre à jour notre nouvelle configuration.

Fichier ceph.conf

```
[global]
fsid = 10c95f01-2dd2-4863-affa-60c4eafcd8d2
mon_initial_members = golem
mon_host = 192.168.1.51
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd_journal_size = 1024
filestore_xattr_use_omap = true

[mon.golem]
        host = golem
        mon addr = 192.168.1.51:6789
```

On a ensuite redémarré le service ceph, pour voir si la nouvelle configuration à bien été mise en service.

Installation du premier osd

Pour installer un osd sur une node, il faut une partition en xfs.

On regarde d'abord la partition qu'on vas utiliser

```
$ceph-deploy disk list golem
```

on a ensuite «zap» la partition :

```
$ceph-deploy disk zap golem:sda3
```

Ensuite pour commencer l'installation, on a préparé l'installation

```
$ceph-deploy osd prepare golem:sda3
```

Et on l'a activé:

```
$ceph-deploy osd activate golem:sda3
```

Nous avons modifié le fichier /etc/ceph/ceph.conf, pour que l'osd soit utilisé par le système de fichier:

```
[global]
fsid = 10c95f01-2dd2-4863-affa-60c4eafcd8d2
mon_initial_members = golem
mon_host = 192.168.1.51
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd_journal_size = 1024
filestore_xattr_use_omap = true

[mon.golem]
        host = golem
        mon addr = 192.168.1.51:6789
[osd.0]
        host = golem
        addr = 192.168.1.51:6789
```

Et on a redémarré le service ceph pour voir si tout marche correctement:

```
$service ceph restart
```

ceph status montre que le monitor initial et le premier osd sont actifs.

Le système n'est pas stable ici, car il faut 2 osd pour que le système de fichier fonctionne correctement.

Installation du second monitor

On a ajouté le second monitor:

```
$ceph-deploy install rondoudou

$ceph-deploy mon create rondoudou
```

et on a modifié /etc/ceph/ceph.conf sur toute les nodes:

```
[global]
fsid = 10c95f01-2dd2-4863-affa-60c4eafcd8d2
mon_initial_members = golem
mon_host = 192.168.1.51
auth cluster required = cephx
```

```
auth service required = cephx
auth client required = cephx
osd_journal_size = 1024
filestore_xattr_use_omap = true

[mon.golem]
        host = golem
        mon addr = 192.168.1.51:6789

[mon.rondoudou]
        host = rondoudou
        mon addr = 192.168.1.29:6789

[osd.0]
        host = golem
        addr = 192.168.1.51:6789
```

Installation du second osd

On regarde d'abord la partition qu'on vas utiliser:

```
$ceph-deploy disk list rondoudou
```

on a ensuite «zap» la partition:

```
$ceph-deploy disk zap rondoudou:sda3
```

on a préparé en activé l'osd:

```
$ceph-deploy osd prepare rondoudou:sda3
$ceph-deploy osd activate rondoudou:sda3
```

on a modifié le fichier /etc/ceph/ceph.conf, pour que l'osd soit utilisé par le système de fichier:

```
[global]
fsid = 10c95f01-2dd2-4863-affa-60c4eafcd8d2
mon_initial_members = golem
mon_host = 192.168.1.51
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd_journal_size = 1024
filestore_xattr_use_omap = true

[mon.golem]
        host = golem
        mon addr = 192.168.1.51:6789

[mon.rondoudou]
        host = rondoudou
        mon addr = 192.168.1.29:6789

[osd.0]
```

```
        host = golem
        addr = 192.168.1.51

[osd.1]
        host = rondoudou
        addr = 192.168.1.29
```

On a ensuite redémarré le service ceph pour voir le nouveau monitor et le second osd:

$service ceph restart

Installation de MDS

on a ensuite créé le mds, le serveur de métadonnées:

```
$ceph-deploy mds create carapuce
```

mais il y a eu une erreur lors de la creation du mds:

```
[carapuce][WARNIN] No data was received after 300 seconds, disconnecting...
Traceback (most recent call last):
 File "/usr/bin/ceph-deploy", line 21, in <module>
 sys.exit(main())
 File "/usr/lib/python2.7/dist-packages/ceph_deploy/util/decorators.py", line 62, in newfunc
 return f(*a, **kw)
 File "/usr/lib/python2.7/dist-packages/ceph_deploy/cli.py", line 138, in main
 return args.func(args)
File "/usr/lib/python2.7/dist-packages/ceph_deploy/mds.py", line 169, in mds
 mds_create(args)
   File "/usr/lib/python2.7/dist-packages/ceph_deploy/mds.py", line 157, in mds_create
create_mds(distro.conn, name, args.cluster, distro.init)
   File "/usr/lib/python2.7/dist-packages/ceph_deploy/mds.py", line 55, in create_mds
os.path.join(keypath),
TypeError: 'NoneType' object is not iterable
```

Après cette erreur le cluster que nous avions installé probléme rencontré sur ceph après l'installation du mds -> suppression de ceph avec purge autoremove, suppression du cluster et des fichier de conf et réinstallation complète on a suivi le tutoriel du site officiel pour recommencer.

Reinstallation

Configuration du Réseau

Après avoir échoué dans la première installation de Ceph, nous avons décidé de recommencer à 0, Il a été nécessaire d'effectuer la configuration des machines à nouveau:

- golem(monitor administration, osd)
- rondoudou(monitor, osd)
- carapuce(monitor, mds)

Sur chaque machine nous avons modifié le fichier /etc/hosts ajoutant l'alias et l'adresse IP de chaque machine ansi 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
192.168.1.43        carapuce
```

Creation 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 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 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 carapuce
        User ceph
Host rondoudou
        User ceph
```

Creation de cluster et installation des moniteurs

À partir de la machine d'administration (golem) nous avons installé ceph-deploy et créé le cluster avec les 3 moniteurs:

ceph@golem:~$sudo apt-get ceph-deploy

ceph@golem:~$ceph-deploy new golem rondoudou carapuce

À partir de golem nous avons installé ceph sur les autres postes:

ceph@golem:~$ceph-deploy install rondoudou carapuce

Nous avons créé un répertoire pour garder la configuration initial du cluster:

ceph@golem:~$mkdir cluster

ceph@golem:~$cd cluster

Pour créér les 3 moniteurs on a utilissé:

ceph@golem:~$ceph-deploy mon create-initial

Le fichier généré ceph.conf doit être modifié pour travailler avec les nouveaux moniteurs que nous avons installé:

fichier ceph.conf

```
[global]
fsid = 10c95f01-2dd2-4863-affa-60c4eafcd8d2
mon_initial_members = golem, rondoudou, carapuce
mon_host = 192.168.1.51, 192.168.1.29, 192.168.1.43
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd_journal_size = 1024
```

filestore_xattr_use_omap = true

[mon.golem]
host= golem
mon addr = 192.168.1.51

[mon.rondoudou]
host= rondoudou
mon addr = 192.168.1.29

[mon.carapuce]
host= carapuce
mon addr = 192.168.1.43

Installation des OSD

Pour l'installation des OSD nous avons créé une partition de type xfs sur la machine golem et la machine rondoudou, depuis golem on a formaté ces partitions (sda3) avec les commandes:

ceph@golem:~/cluster$ceph-deploy disk zap golem:sda3

ceph@golem:~/cluster$ceph-deploy disk zap rondoudou:sda3

Ensuite, nous avons preparé et activé les partitions:

ceph@golem:~/cluster$ceph-deploy osd prepare golem:sda3

ceph@golem:~/cluster$ceph-deploy osd activate golem:sda3

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

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

On a modifié le fichier ceph.conf pour travailler avec les nouveaux osd

Fichier ceph.conf

[global]
fsid = 10c95f01-2dd2-4863-affa-60c4eafcd8d2
mon_initial_members = golem, rondoudou, carapuce
mon_host = 192.168.1.51, 192.168.1.29, 192.168.1.43
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd_journal_size = 1024
filestore_xattr_use_omap = true

[mon.golem]
host= golem
mon addr = 192.168.1.51

```
[mon.rondoudou]
        host= rondoudou
        mon addr = 192.168.1.29

[mon.carapuce]
        host= carapuce
        mon addr = 192.168.1.43
[osd.0]
        host = golem
        addr = 192.168.1.51

[osd.1]
        host = rondoudou
        addr = 192.168.1.29
```

Instalaltion MDS

En raison de la procédure réalisée sur la première installation nous avons trouvé une erreur dans cette étape, mais gràce l'ajout de moniteurs depuis le début, on a réussi à installer le MDS dans notre cluster avec la commande suivante executée depuis golem:

```
ceph@golem:~/cluster$ceph-deploy mds create carapuce
```

Emulation sur machines virtuelles

Pour tester notre cluster sur un matériel différent, nous avons décidé de créer des machines virtuelles pour émuler un cluster et observer le performance.

L'environnement de test c'est le suivant

processeur Interl core i7(8 cores, deux pour chaque machine et deux pour la machine host)
Memoire vive 4Go
1 disque dur ssd de 120 Go

Conclusion

L'installation d'un système en fonctionnement n'a pas était facile. C'est pourquoi nous allons éviter de faire les mêmes erreurs avec Lustre, ce qui nous fera gagner beaucoup de temps. Il reste maintenant à finir avec Ceph et de renouveller l'expérience avec Lustre. Dans un futur très proche, on va donc procéder à toutes ces étapes :

- Test de Ceph avec des postes clients
- Tests de performances de Ceph
- Choisir nouveau type de cluster pour Lustre
- Rédaction manuel d'installation Ceph

Nous pensons effectuer certains de ces taches en parallèle afin d'optimiser au mieux le travail en équipe et d'être le plus efficace possible. Pour finir je dirai que nous sommes toujours présents à chaque séance et que tous les membres de l'équipe se sentent impliqués dans ce projet.