

Projet tutoré

Outils de gestion centralisée de machines virtuelles

Bocca Augustin,Lamoureux Mathieu,Michaux Sebastien,Tournois Julien

vendredi 17 février

Préambule :

Table des matières

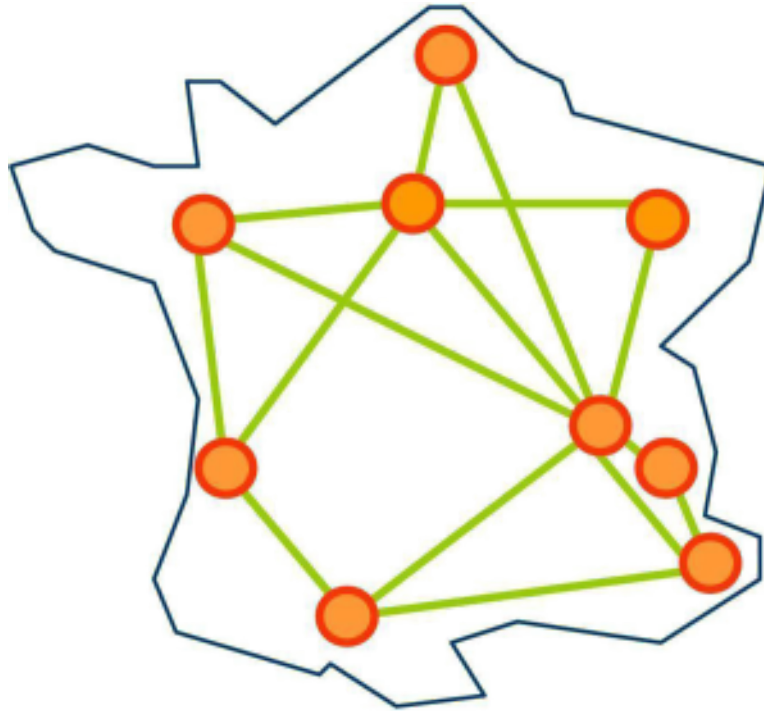
1	Introduction	3
1.1	Présentation de Grid5000	3
1.1.1	Infrastructure des sites	4
1.1.2	Réseau	6
1.1.3	Environnement logiciel	6
1.2	Présentation du projet	7
1.3	Introduction à la virtualisation	7
2	Ganeti	8
2.1	Introduction	8
2.2	Installation	8
2.3	Utilisation de ganeti	9
3	Scripts	11
3.1	réservation	11
3.2	déploiement	12
3.3	configuration de base	13
3.4	installation de ganeti	15

Chapitre 1

Introduction

1.1 Présentation de Grid5000

Aujourd'hui, grâce à Internet, il est possible d'interconnecter des machines du monde entier pour traiter et stocker des masses de données. Cette collection hétérogène et distribuée de ressources de stockage et de calcul a donné naissance à un nouveau concept : les grilles informatiques. L'idée de mutualiser les ressources informatiques vient de plusieurs facteurs, évolution de la recherche en parallélisme qui, après avoir étudié les machines homogènes, s'est attaquée aux environnements hétérogènes puis distribués ; besoins croissants des applications qui nécessitent l'utilisation toujours plus importante de moyens informatiques forcément répartis. La notion de grille peut avoir plusieurs sens suivant le contexte : grappes de grappes, environnements de type GridRPC (appel de procédure à distance sur une grille), réseaux pair-à-pair, systèmes de calcul sur Internet, etc... Il s'agit d'une manière générale de systèmes dynamiques, hétérogènes et distribués à large échelle. Un grand nombre de problématiques de recherche sont soulevées par les grilles informatiques. Elles touchent plusieurs domaines de l'informatique : algorithmique, programmation, intergiciels, applications, réseaux. L'objectif de GRID'5000 est de construire un instrument pour réaliser des expériences en informatique dans le domaine des systèmes distribués à grande échelle (GRID). Cette plate-forme, ouverte depuis 2006 aux chercheurs de la communauté grille, regroupe un certain nombre de sites répartis sur le territoire national. Chaque site héberge une ou plusieurs grappes de processeurs. Ces grappes sont alors interconnectées via une infrastructure réseau dédiée à 10 Gb/s fournie par RENATER. À ce jour, GRID'5000 est composé de 9 sites : Lille, Rennes, Orsay, Nancy, Bordeaux, Lyon, Grenoble, Toulouse et Nice. Début 2007, GRID'5000 regroupait plus de 2500 processeurs et près de 3500 cœurs.

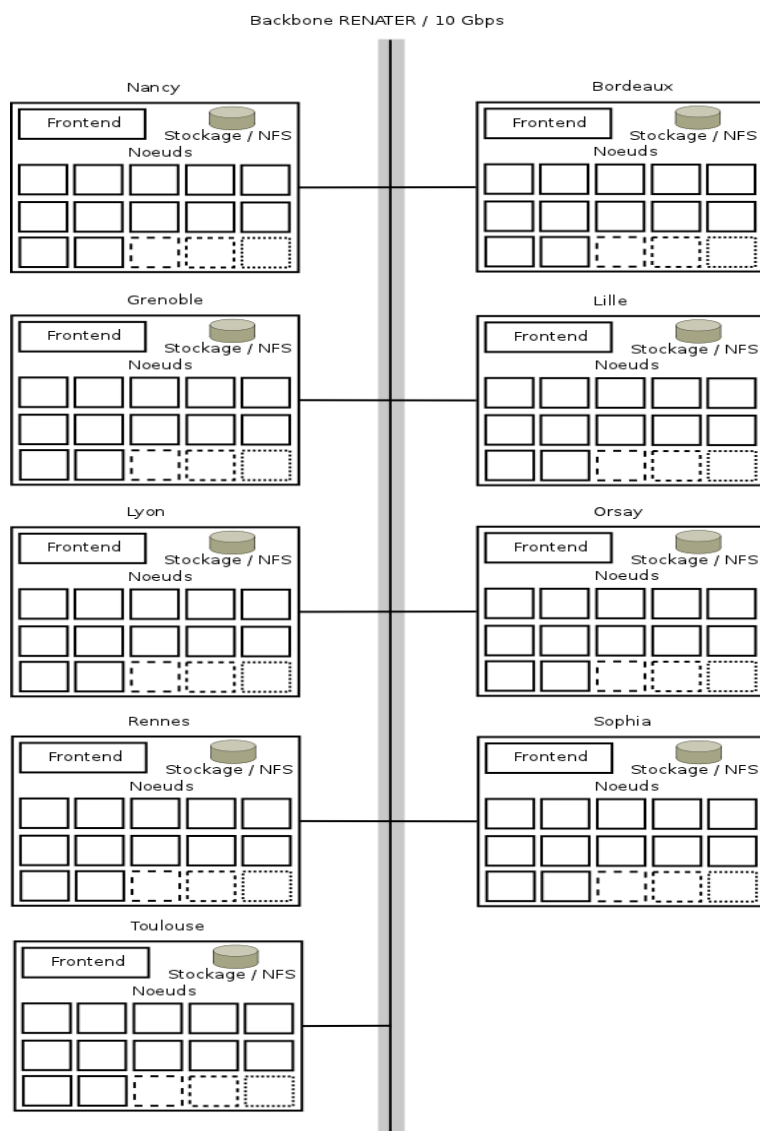


Architecture Grid5000

1.1.1 Infrastructure des sites

Chaque site héberge :

- un frontend, serveur permettant d'accéder aux clusters disponibles ,
- un serveur de données, pour centraliser les données utilisateurs ,
- plusieurs clusters, c'est-à-dire des grappes de machines homogènes, appelées noeuds (nodes).



L'utilisateur de Grid 5000 accède à chaque site par son frontend en utilisant le protocole SSH.

Commande :

```
1 ssh jdoe@access.grid5000.fr
```

Sur tous les serveurs du site, un répertoire home, local à chaque site, est monté avec NFS 2. A partir du frontend, il est possible d'accéder aux machines des clusters en exécutant des réservations à l'aide de la commande :

```
1 oarsub
```

Grâce à notre tuteur, M. Lucas Nussbaum nous avons pu visiter la salle serveurs du site de Nancy située au Loria, ainsi qu'une présentation de la plate-forme (matériel utilisé, connexions réseau, administration).

Une description détaillée du site de Nancy est disponible sur le site de Grid 5000.

1.1.2 Réseau

Les sites et toutes les machines qu'ils comprennent sont interconnectés par RENATER 3 en 10Gbits/s. De plus, chaque site peut disposer de plusieurs réseaux locaux 4 :

- réseau en ethernet, 1 Gb/s
- réseaux hautes performances (Infiniband 20 Gb/s ou 10 Gb/s, et Myrinet 20 Gb/s)

1.1.3 Environnement logiciel

Tous les serveurs de Grid 5000 fonctionnent sous Debian GNU/Linux. À partir du frontend, l'utilisateur peut réserver des machines en utilisant la suite de logiciels OAR dédiée à la gestion de ressources de clusters, et déployer ses propres images de systèmes à l'aide des outils kadeploy. Il y a deux types de réservation :

- par défaut, pour des besoins de calcul avec OpenMPI;
- pour le déploiement d'environnements (deploy).

1.2 Présentation du projet

Objectif Mettre en place, évaluer et comparer différents outils permettant de gérer de manière centralisée et automatisée une infrastructure basée sur des machines virtuelles : Ganeti, Open-XenManager, virt-manager, Archipel...

1.3 Introduction à la virtualisation

Chapitre 2

Ganeti

2.1 Introduction

Ganeti est un outil de gestion de machines virtuelles se basant sur les technologies de virtualisation existantes comme XEN et KVM.

Ganeti nécessite un logiciel de virtualisation pré-installé sur les serveurs afin de pouvoir fonctionner. Une fois installé, l'outil prendra en charge la partie gestion des instances virtuelles (Xen DomU), par exemple, la gestion de création de disque, l'installation du système d'exploitation (en coopération avec les scripts d'installation du système d'exploitation spécifique), et le démarrage, l'arrêt, le basculement entre les systèmes physiques. Il a été conçu pour faciliter la gestion de cluster de serveurs virtuels et de fournir une récupération rapide et simple.

2.2 Installation

```
1 apt-get install ganeti2 ganeti-htools ganeti-instance-  
    debootstrap
```

Puis dans `/etc/host` il faut ajouter l'adresse IP du node avec le nom de l'host ainsi que l'adresse du cluster :

```
1 10.144.64.1 cluster1  
  172.16.65.56 griffon -56.nancy.grid5000.fr
```

*Lorsque l'on a plusieurs nodes il faut une autre adresse
Ip au cluster car elle doit etre accessible à tous les nodes*

Ensuite, dans /boot/ creer des liens symboliques :

```
ln -s vmlinuz-2.6.32-5-xen-amd64 vmlinuz-2.6.xenU
ln -s initrd.img-2.6.32-5-xen-amd64 initrd.img-2.6.xenU
```

Il est obligatoire de creer un LVM d'au moins 20Go sur tous les noeuds car ce volume permettra de stocker par exemple les systèmes de fichiers , si vous voulez utiliser toutes les fonctionnalités ganeti :

```
umount /dev/sda5
pvcreate /dev/sda5
vgcreate xenvg /dev/sda5
```

Dans /etc/network/interface remplacer le paragraphe de eth0 par celle du brige xen-br0 :

```
1 auto xen-br0
  iface xen-br0 inet static
3 address 172.16.65.56 #address du node
  netmask 255.255.240.0
5 network 172.16.64.0
  broadcast 172.16.79.255
7 gateway 10.144.64.254
  bridge_ports eth0
9 bridge_stp off
  bridge_fd 0 option{ Utilisation }
```

Pourquoi un nom d'hôte ? Bien que la plupart des distributions utilisent uniquement le nom court dans le fichier / etc / hostname, ganeti utilise le nom complet. La raison en est que 'le nom d'hôte-fqdn' appelle la bibliothèque de résolution. Deplus ganeti peut être utilisé entre autres pour accueillir les serveurs DNS.

2.3 Utilisation de ganeti

Initialiser le cluster avec ganeti :

```
gnt-cluster init --no-drbd-storage cluster1
```

ajouter le node :

```
1 gnt-node add griffon-56.nancy.grid5000.fr
```

Il est ensuite possible de verifier la liste des nodes avec :

```
1 gnt-node list
```

Qui devrait renvoyer quelque chose comme cela :

```
Node DTotal DFree MTotal MNode MFree Pinst Sinst  
griffon-56.nancy.grid5000.fr 283.2G 283.2G
```

Chapitre 3

Scripts

3.1 réservation

```
1 puts "_____"
2 puts "Souhaitez vous reserver des noeuds?(y/n) "
3 loop do
4   test = gets.chomp
5   if test.eql?("n")
6     puts "#####"
7     puts "#sortie du programme#"
8     puts "#####"
9     break;
10  end
11  if test.eql?("y")
12    #Reservation de machines
13    puts "_____"
14    puts "Script de reservation"
15    puts "Choisir un nombre de noeud:"
16    noeuds = gets.chomp.to_i
17    puts "Choisir un temps de reservation(HH:MM:SS):"
18    temps = gets
19    puts "\nVous avez reserve #{noeuds} noeuds"
20    puts "pour une duree de #{temps}"
21    puts "_____"
22    exec "oarsub -I -t deploy -n'virtu' -l"
23    slash_22=1+nodes==#{noeuds},walltime=#{temps}"
24    break;
25  end
26 end
```

3.2 déploiement

```

1 #Deployer une image cree
  puts "Voulez vous deployer une image?(y/n) "
3 loop do
  test = gets.chomp
5  if test.eql?("n")
    puts "#####"
7    puts "#sortie du programme#"
    puts "#####"
9    break;
  end
11  if test.eql?("y")
    #choix de l'image
13    puts "
      "
      puts "image disponibles:"
15    puts `ls /home/$USER/image | grep .env`
    puts "
      "
17    puts "Choix de la distribution(tout saisir):"
    debian = gets.chomp
19    puts "
      "
      exec "kadeploy3 -f $OAR_FILE_NODES -a #{debian} -k
21 $HOME/.ssh/id_rsa.pub"
      break;
23  end
end
25 #Deploiement de l environnement sur les noeuds reserves
  puts "Voulez vous deployer un environnement?(y/n) "
27 loop do
  test = gets.chomp
29  if test.eql?("n")
    puts "#####"
31    puts "#sortie du programme#"
    puts "#####"
33    break;
  end
35  if test.eql?("y")
    #choix de la version a deployer
37    puts "
      "
      puts "distributions disponibles:"
39    puts `kaenv3 -l | cut -d - -f1 | uniq | tail -n +3`
    puts "
      "
41    puts "Choix de la distribution:"

```

```

43     debian = gets.chomp
    puts "
        "
    puts "version de la distribution:"
45     puts `kaenv3 -l | grep #{debian}| cut -d ' ' -f1`
    debian = debian+"-x64-"
47     puts "
        "
    puts "Choix de la version de la distribution a
        deployee
49     (sans #{debian}):"
    version = gets.chomp
51     version = debian+version
    puts version
53     exec "kadeploy3 -e #{version} -f $OAR_FILE_NODES -k
    $HOME/.ssh/id_rsa.pub"
55     break;
    end
57 end

```

3.3 configuration de base

```

1  ##
   #####
   #-----Configuration minimale
   #-----#
3  #-----#
   #
5  #recuperation des noeuds reserves
   cat $OAR_FILE_NODES | uniq > $HOME/script_base/
   list_nodes
7  list_nodes="$HOME/script_base/list_nodes"
9  echo "-----"
   echo "Liste des machines reservee:"
11 cat $list_nodes
   echo "-----"
13 echo "Copie des clees SSH vers toutes les machines."
   for node in $(cat $list_nodes)
15 do
       scp $HOME/.ssh/id_rsa* root@$node:~/.ssh/
17 done
   echo "-----"
19 #mise a jour des noyaux

```

```
21 taktuk -l root -f $list_nodes broadcast exec [ apt-get  
    update ]  
    taktuk -l root -f $list_nodes broadcast exec [ apt-get  
        dist-upgrade  
23     -q -y --force-yes ]  
    #changement des mdp root  
25 taktuk -l root -f $list_nodes broadcast exec [ 'echo -e  
        pttvirtu\npttvirtu" | passwd root ' ]
```

3.4 installation de ganeti

```

#!/bin/bash
2 #passage en wheezy
rm /etc/apt/sources.list
4 echo "## wheezy security" > /etc/apt/sources.list
echo "deb http://security.debian.org/ wheezy/updates
  main contrib non-free" >> /etc/apt/sources.list
6 echo "deb-src http://security.debian.org/ wheezy/
  updates main contrib non-free" >> /etc/apt/sources.
  list
echo " " >> /etc/apt/sources.list
8 echo "#wheezy" >> /etc/apt/sources.list
echo "deb http://ftp.fr.debian.org/debian/ wheezy main
  contrib non-free" >> /etc/apt/sources.list
10 echo "deb-src http://ftp.fr.debian.org/debian/ wheezy
  main contrib non-free" >> /etc/apt/sources.list
#Installation de ganeti
12 apt-get update
apt-get dist-upgrade -y --force-yes
14 apt-get install -y --force-yes ganeti2 ganeti-htools
  ganeti-instance-debootstrap
echo "Ajout du node dans /etc/hosts"
16 hostname='cat /etc/hostname'
#recuperation des variables
18 #ip du node
ifconfig eth0 > troll
20 ipnode='head -2 troll | tail -1 | cut -d':' -f2 | cut -
  d' ' -f1 '
echo $ipnode $hostname >> /etc/hosts
22 #ip du broadcast
ipbroadcast='head -2 troll | tail -1 | cut -d'B' -f2 |
  cut -d':' -f2 | cut -d' ' -f1 '
24 #ip du masque de sous reseau
ipmask='head -2 troll | tail -1 | cut -d'M' -f2 | cut -
  d':' -f2 | cut -d' ' -f1 '
26 #ip du reseau
ipnetwork='head -1 ipnetwork '
28 #ip de la passerelle
ipgateway='head -1 ipgateway '
30 #ajout de cluster1 dans dans /etc/hosts
echo "ajout de cluster1 dans /etc/hosts"
32 ipcluster='cat ipcluster '
echo $ipcluster "cluster1" >> /etc/hosts
34 #Dans /boot/ creer des liens symboliques :
ln -s /boot/vmlinuz-2.6.32-5-xen-amd64 /boot/vmlinuz
  -2.6.xenU
36 ln -s /boot/initrd.img-2.6.32-5-xen-amd64 /boot/initrd.
  img-2.6.xenU
#Pour le moment changera surement.
38 echo "creation du LVM"
umount /dev/sda5
40 pvcreate /dev/sda5

```



```
vgcreate xenvg /dev/sda5
42 #Creation du bridge xen-br0
echo " " >> /etc/network/interfaces
44 echo "auto xen-br0" >> /etc/network/interfaces
echo "iface xen-br0 inet static" >> /etc/network/
    interfaces
46 echo "address" $ipnode >> /etc/network/interfaces
echo "netmask " $ipmask >> /etc/network/interfaces
48 echo "network" $ipnetwork >> /etc/network/interfaces
echo "gateway" $ipgateway >> /etc/network/interfaces
50 echo "broadcast" $ipbroadcast >> /etc/network/interfaces
echo "bridge_ports eth0" >> /etc/network/interfaces
52 echo "bridge_stp off" >> /etc/network/interfaces
echo "bridge_fd 0" >> /etc/network/interfaces
54 #Suppression des ligne de eth0
sed -i '9d' /etc/network/interfaces
56 sed -i '9d' /etc/network/interfaces
#supression des fichier temporaires
58 rm troll ipcluster ipgateway ipnetwork
#initialisation du cluster
60 gnt-cluster init --no-drbd-storage cluster1
#ajouter le node
62 gnt-node add $hostname
#et verifier
64 gnt-node list
```