Dossier Appel D'Offre LC

Tàzio Gennuso & Pierre Prevelle

11 mai 2018

Analyse de la réponse technique

Comparons la solution technique apportée par la société XtremCompute et le cahier des charges.

Pour la partition A le cahier des charges spécifiait au moins 10 000 cœurs avec au moins 4Go de mémoire par cœur et la solution apporte 11 520 cœurs avec 4Go de mémoire par coeurs, ce qui correspond aux attentes. Pour la partition B, la contrainte du cahier des charges était 5 000 cœur avec 8Go de mémoire par cœur, la solution pour la partition B comprend 4800 cœurs avec 8Go de mémoire chacun, ce qui est un peu inférieur à la demande. Et pour la partition C, la contrainte était au moins 500 cœurs pour exécuter des codes CUDA, la solution propose 480 cœurs permettant d'exécuter des code CUDA grâce a des cartes Nvidia P100 Pascal. Au niveau des partitions les contraintes on était globalement respecté malgré un nombre de coeurs un peu inférieur à celui demandé.

Le nombre de nœud de login dans la solution ne suffit pas à la demande du cahier des charges puisque l'université voulait suffisamment de nœud pour que les populations soient séparées, soit 12 nœuds de login, or XtremeCompute n'en propose que 10. Pour l'offre logiciel en CentOS, rien n'est précisé dans la solution. Quand au débit d'I/O, le cahier des charges stipule que le débit doit être de 100Gb/s, et avec la solution au global pour les nœuds d'I/O le débit est de 160Gb/s donc le critère est bien respecté.

Architecture réseau du cluster

Il y a trois réseaux physique dans le cluster, un réseaux ethernet 1Gb/s, un réseau ethernet 10Gb/s et un réseau Infiniband EDR.

Il y a quatre réseaux logiques, le réseaux d'interconnect du cluster permettant la communication entre les nœuds (celui en Infiniband EDR peut le porter), le réseaux d'I/O qui permettra les échanges avec le serveur lustre, le réseau d'administration qui permet la gestion des différents services et matériel qui peut être en ethernet et le réseaux d'utilisation (celui qui communique avec l'extérieur et qui permet de communiquer avec l'extérieur) en ethernet .

Pour l'adressage, pour être en accord avec l'adressage déjà présent (selon l'annexe) on peut prendre les adresse de type 172.12…

13 switchs Ethernet 48 ports sont nécessaire à l'infrastructure ethernet du cluster car en tout nous devenons connecté 576 nœuds et 26 switchs.

Nommage des nœuds

On peut nommer les nœuds en fonction de leur partitions ou de si il sont des nœuds de service:

- nodeA[0000-0359]
- nodeB[0000-0149]
- nodeC[000-029]
- nodeS[000-035]

Ainsi on sait à quelles partitions appartiennent les nœuds et on peut ajouter des nouveau nœuds à ces nodesets si jamais on agrandit le cluster (nodeset -f).

Architecture NTP

Pour l'architecture NTP, nous utiliserons les serveurs de l'université comme référence, ils fonctionneront en mode client-serveur avec un nœud de service qui fera office de strate 2 (comme les serveurs sont de la strate 1 car connecté à des GPS), et celui-ci communiquera en broadcast avec les autres nœuds, on perd en précision mais cela évite le grands nombres de requêtes engendrés si tous les nœuds de doivent interroger ce nœud.

config du nœud de service: [path = /etc/ntp.conf]

```
driftfile /var/lib/ntp/drift

restrict default nomodify notrap nopeer noquery

restrict 127.0.0.1
restrict ::1

broadcast 127.0.0.1 autokey        # broadcast server

includefile /etc/ntp/crypto/pw

keys /etc/ntp/keys

server chronix1.univ8.fr
fudge chronix1.univ8.fr stratum 0
```

Config des autre nœuds: [path = /etc/ntp.conf]

```
driftfile /var/lib/ntp/drift

restrict default nomodify notrap nopeer noquery

restrict 127.0.0.1
restrict ::1

broadcastclient

includefile /etc/ntp/crypto/pw
```

```
keys /etc/ntp/keys
```

Architecture SYSLOG

On place un serveur rsyslog sur un nœud de chacun des 20 groupes. Ce serveur récupère les logs de tous les nœuds du groupe, et les renvoie au serveur rsyslog central, sur un des nœuds maîtres.

Architecture DNS

On place un serveur DNS par partition, qui seront tous en forward vers celui de la partition S, qui lui-même forward sur le serveur DNS de l'université. Cela n'est peut-être pas la méthode la plus efficace en terme de performance et de disponibilité, mais elle est plus résiliente, et ne requiert pas de ressources supplémentaires.

Architecture LDAP

Pour le service LDAP, nous utiliserons le système de maîtres/esclaves pour doubler les serveurs LDAP pour protéger des éventuels pics de charges, ainsi nous mettrons en place dans chaque partition un serveur esclave (pour le nœuds de calcul et de service (A, B, C et S) ainsi qu'un serveur LDAP dédiés uniquement à l'administration). Ces serveurs esclaves communiqueront avec des serveurs maîtres, les serveurs LDAP de l'université. Nous pourrions aussi utiliser une solution de load balancer (comme keepalived (soft) ou un modèle F5 (hard) par exemple) pour répartir la charge entre les deux serveurs.

Config client:

```
Provider: /etc/nsswitch.conf
 Passwd:
compat ldap
 Group:
compat ldap
 Shadow:
compat
 /etc/nscd.conf
 /etc/nslcd.conf
 /etc/openldap/ldap.conf
 BASE dc=ccc,dc=cdc,dc=fr
 URI
ldap://amnesixl1.univ8.fr
 #SIZELIMIT
12
 #TIMELIMIT
15
 #DEREF
never
```

```
TLS_CACERTDIR /etc/openldap/certs
```

Config slave:

```
Configuration classique
# Replication
syncrepl rid=123
provider=ldap://amnesixl1.univ8.fr
type=refreshOnly
interval=00:00:02:00
retry="60 +"
searchbase="dc=ccc,dc=cdc,dc=fr"
filter="(objectClass=*)"
scope=sub
```

Configuration Slurm

Pour mettre en place une comptabilité des ressources consommées ainsi qu'une politique d'allocation des ressources en fonction des différentes populations, il faut mettre en place des QOS dans slurm.

Configuration de la surveillance

Avec un outil de métrologie comme Shinken, on peut surveiller des points cruciaux pour le calculateur, tels la charge cpu (qui doit toujours être supérieure à 95%), l'espace disque des machines, le réseau, le nombre d'utilisateurs, ou la température du matériel.
On utilisera les plugins suivants :

- check_by_ssh
- check_disk
- check_dns
- check_load
- check_ntp_peer
- check_ntp_time
- check_ping
- check_procs
- check_sensors
- check_ssh
- check_swap
- check_tcp
- check_udp
- check_users

Configuration Puppet

On utilisera un des nœuds de service comme serveur puppet, en utilisant Hiera et un ENC on facilitera l'exécution des mise a jour puisque les classe seront dans Hiera et non plus dans les manifests, et l'ENC permettra de faire des mise a jour en fonction des utilisations des noeuds (grâce au partition et à la répartition des noeuds) mais aussi il permettra de coupler puppet avec un le serveur mettant déjà les mise a jour de l'université, le serveur cotomatix.

Comme différentes classes pour les noeuds on pourra proposer une classe admin, une classe service, une classe login et une classe calcul qui correspondent respectivement au noeuds d'admin, aux différents noeuds de service externe, au noeuds de login et finalement aux noeuds de calcul qui proposent des fonctionnalité différentes et ont besoin de mise à jour différentes.

Organisation des nœuds logins et routeur

Pour la répartition des noeuds de login, nous savons que nous avons que 10 noeuds disponible sur les 12 demandés, en regardant la répartition on voit que les popula[06-09] et popula[10-11] n'ont besoin que d'un noeuds par population et que les popula[00-03] ont besoin d'un noeuds pour deux populations, on peut donc considérer que nous pouvons retirer des noeuds de login aux popula[04-05] pour qu'elles n'est que deux noeuds de login et non pas quatre. Quand au placement de ceux ci sur le cluster, étant données qu'il est judicieux de mettre les noeuds d'I/O reparti sur les deux switch pour essayer de répartir la charge sur le réseau et d'ainsi mettre 8 noeuds sur un switch et 8 noeuds sur l'autre (car deux switchs sont dédiés aux noeuds de services), on peut donc essayer de répartir les noeuds de login aussi pour les même raison en essayant de séparer les populations car les noeuds de login n'ont pas nécessairement besoin d'êtres proches les uns des autres pour être performants (contrairement à ceux de calcul) pour répartir la charge sur le réseaux lorsque toutes une même population se connecte.

Installation

On commencera d'abord par installer la salle machine (faire des travaux si besoin) puis on installe les racks et tous les réseaux (électriques, hydraulique puis ceux de connexions). Puis on procède à l'installation logicielle des master nodes, on configure le réseaux de management puis on installe les noeuds de services (installation logicielle puis configuration du réseaux). Après on met en place les outils d'installation des noeuds de calcul, on déploie les noeuds de services, ensuite on déploie les noeuds de calculs puis on les configure.

Une fois l'installation matériel et logicielle terminée on entre dans une phase de validation de performances, progressivement d'un simple noeuds à l'ensemble de la machine. Après cela on procède à plusieurs test : des tests de performance, de robustesse et de bon fonctionnement. Puis avant de le mettre en phase de production on le configure pour l'insérer

dans la structure de l'université, on vérifie régulièrement le bon fonctionnement, des premiers utilisateurs test utilise la machine et on regarde les performances, puis on peut éventuellement, essayer l'ensemble de la machine grâce à un "Grands Challenges" et enfin on le fait rentrer en phase de production.