

Activité Répartition de Charge

HAProxy

La répartition de charge permet d'utiliser tous les serveurs disponibles pour un service donné, de manière à optimiser l'utilisation du matériel. Les requêtes des clients sont réparties à tour de rôle sur chaque serveur...

1. Rappel : ACTIF/ACTIF vs ACTIF/PASSIF

1.1. Cluster actif/passif :

Le serveur passif est inutilisé pendant que le serveur principal est actif. La plupart du temps cela entraîne un gaspillage financier et de ressources. En effet, le serveur secondaire ne prend le relais qu'uniquement lorsque le serveur principal tombe en panne alors que le reste du temps il demeure passif.

1. Comment appelle-t-on ce mode de haute disponibilité ?

1.2. Cluster actif/actif :

Plusieurs serveurs sont actifs en même temps pour répondre à toutes les demandes des clients. Il faut alors mettre en place plusieurs serveurs proposant le même service. Ensuite il faut également faire en sorte que les requêtes des clients soient réparties équitablement sur chacun des serveurs proposant le service (répartition de charge). Plusieurs solutions de répartitions sont possibles et implémentées en fonction d'algorithmes définis en ce sens.

2. Comment appelle-t-on ce mode de haute disponibilité ?

2. Solution de mise en œuvre proposée : HAProxy

Nous allons dans ce TP nous concentrer sur la deuxième approche en la mettant en œuvre en suivant plusieurs étapes.

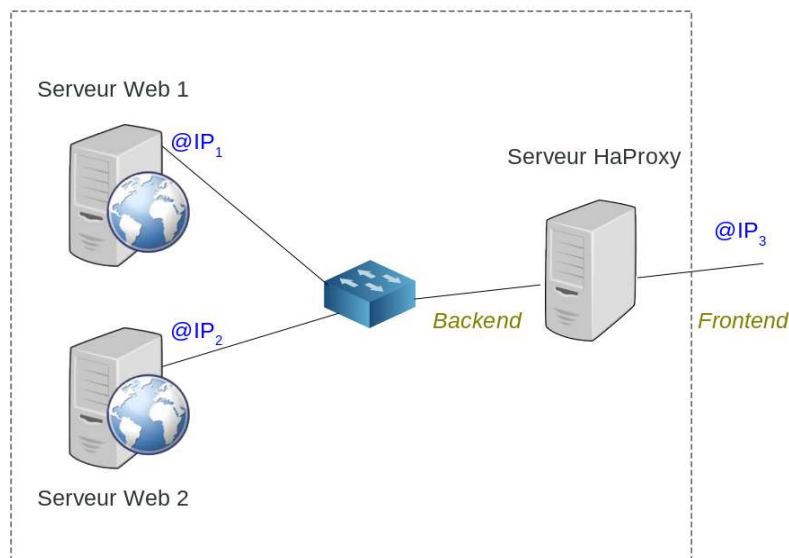
2.1. Configuration 1 : HAPROXY en réseau unique

HAProxy¹ est une solution libre, fiable et très performante de répartition de charge de niveau 4 (TCP) et 7 (HTTP)

Pour commencer donc, nous proposons d'étudier la répartition de charge au travers une infrastructure réseau simple comprenant deux serveurs Web.

Le service HAProxy va se charger de répartir les requêtes à tour de rôle sur ces deux serveurs web.

La mise en œuvre de cette première étape consistera donc à établir une infrastructure réseau qui respectera le schéma réseau donné ci-dessous :



Matériels requis : trois machines virtuelles Debian

(on choisira des machines avec la dernière version de Debian stable)

Travail à faire :

1. Créer trois VMs sous VirtualBox avec une distribution de GNU/Linux Debian
2. Configurer les cartes réseaux des trois VM en mode pont de Virtualbox
3. Configurer les interfaces réseau des VMs dans le même réseau en utilisant vos IP fixes
4. Renommer les machines
 - 1^{re} machine : **webserver1** dans le fichier /etc/hostname
 - 2^e machine : **webserver2** dans le fichier /etc/hostname
 - 3^e machine : **votreNom-haprox** dans le fichier /etc/hostname
5. Installer ou relancer Apache sur les deux premiers serveurs Debian (apt install ??)
6. Ajouter/modifier les pages d'accueil de chacun des serveurs Web (fichier index.html dans

1 Lien vers la documentation officielle : <https://www.haproxy.org/#docs>

/var/www/public_html² par défaut), en mettant en titre le nom du serveur hébergeant la page (de manière à les différencier durant la phase de test)

7. Installer le paquet haproxy uniquement sur le serveur n°3 (apt-get install haproxy)
8. Effectuer un backup du fichier de configuration du service avant de le modifier (cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bck)
9. Puis éditer ce fichier de configuration comme ci-dessous :

```
global      # on laisse les paramètres par défaut
defaults   # on laisse les paramètres par défaut
# ajouter en fin de fichier les lignes suivantes
# et éventuellement commenter les configurations déjà présentes
frontend webpublic
    bind @IP3:80 #(cf. schema) à remplacer par adresse IP (publique) du
serveur HAProxy
    default_backend serveurs_caches # section couvrant les serveurs répartis
    option forwardfor # autoriser la redirection http vers les serveurs web
    backend serveurs_caches # section couvrant les serveurs répartis
        balance roundrobin
    server webserver1 @IP1:80 check # webserver1 est le hostname de mon serveur
web 1
    server webserver2 @IP2:80 check # webserver2 est le hostname de mon serveur
web 2
```

Quelques compléments sur cette configuration :

- frontend et backend sont des sections standards obligatoires
 - frontend définit comment les clients ont accès au service web
 - backend définit la composition du cluster et le comportement des nœuds
- webpublic et serveur_caches sont des noms inventés par l'administrateur
- la ligne server se décompose ainsi :
 - server *nom_du_serveur_apache* @*IP_du_serveur_apache*:*port_d_ecoute* check
- balance roundrobin signifie que la répartition de charge se fait à tour de rôle. On pourrait aussi indiquer un autre mode de répartition par exemple pour chaque nœud, le pourcentage de requêtes qu'il doit traiter.

10. Créer un fichier haproxytest.txt sur chaque serveur dans /var/www/public_html.

HaProxy vérifiera l'existence de ce fichier sur chaque serveur toutes les deux secondes.

Si le fichier est absent d'un serveur, HaProxy le considérera comme inactif.

11. Lancer le service haproxy :

```
#systemctl start haproxy.service (par exemple)
```

12. Vérifier le bon fonctionnement du service :

```
#systemctl status haproxy.service (par exemple)
```

Pour obtenir des détails sur les erreurs de configuration et les corriger :

```
journalctl -xe | grep haproxy | more
```

13. Tester la mise en œuvre effective du service de répartition de charge HAProxy

Pour cela vous aller ouvrir un navigateur Internet sur votre machine hôte et ouvrir une page Web en effectuant une requête HTTP sur l'IP correspondant au répartiteur HAProxy (@IP3).

Rafraîchissez la page régulièrement et videz parfois le cache du navigateur.

3. Que pouvez-vous remarquer ?

² À adapter en fonction du chemin où se trouve le fichier index.html par défaut à l'installation du serveur Web Apache

2.2. Consultation des journaux et des statistiques

Il est possible de superviser le bon fonctionnement de la répartition de charges des serveurs grâce à l'interface web de HAProxy.

La mise en place de ces fichiers de journalisation (logs) passe par une modification du fichier de configuration en y ajoutant des directives liées au mot-clé « stats ». Il existe deux stratégies différentes.

2.2.1. Insertion des directives dans la section Frontend

```
stats enable           # autoriser la diffusion des statistiques
stats auth admin:adminpassword # authentification pour accéder aux
statistiques
stats refresh 60s      # délai de rafraîchissement
stats uri /haproxy?stats # URI permettant l'accès au rapport des
statistiques
```

2.2.2. Création d'une section spécifique à la journalisation

```
listen webstats      # déclaration et dénomination de la section
bind @IP3:8080       # adresse IP et port des statistiques
stats enable
stats uri /          # l'URI = @IP3 :8080/ (autre possibilité que précédemment)
stats auth admin:adminpassword
stats refresh 30s
```

Travail à faire :

1. Après avoir choisi l'une ou l'autre des deux méthodes précédentes, procéder à la mise en œuvre de la journalisation du processus de répartition de charge.

Remarques :

Il sera nécessaire de redémarrer le service HaProxy une fois le fichier de configuration modifié afin de permettre à celui-ci de prendre en compte ces modifications.

2. Une fois la mise en œuvre de la journalisation effective, consulter ces nouvelles informations via le navigateur. Pour cela, il faut saisir une URL particulière en fonction de l'option choisie :
 - `http://@IP3?stats`
 - `http://@IP3:8080`
3. Effectuer une copie d'écran des résultats obtenus ici

2.3. Traitement des sessions par ajout de cookies

Avec cette première mise en œuvre, si un utilisateur recharge la page régulièrement, il sera redirigé d'un serveur à l'autre quasi systématiquement. Ce fonctionnement n'est pas adapté à toutes les situations, en effet il arrive qu'il soit nécessaire de garder une connexion active vers un serveur (par exemple pour traiter des données en PHP). Pour résoudre ce problème, nous pouvons envoyer un cookie à l'utilisateur. Selon le cookie que le navigateur client stockera sur son poste, il dialoguera uniquement avec le serveur associé au cookie et pourra alors maintenir des échanges durables avec celui-ci, notamment pour la pérennité des sessions PHP.

Nous allons mettre en place cette fonctionnalité.

Travail à faire :

1. Ajouter et modifier les directives suivantes dans la section backend :

```
cookie SRV insert indirect nocache           # ajouter
server webserver1 @IP2:80 check cookie webser1      # modifier
server webserver2 @IP2:80 check cookie webser2      # modifier
```

Dans cet exemple, le serveur HAProxy va intercepter le cookie de session PHP et il ajouter/supprimer la référence du serveur de backend.

Plus d'info dans la [documentation officielle HAProxy](#).

Signification des mots clés :

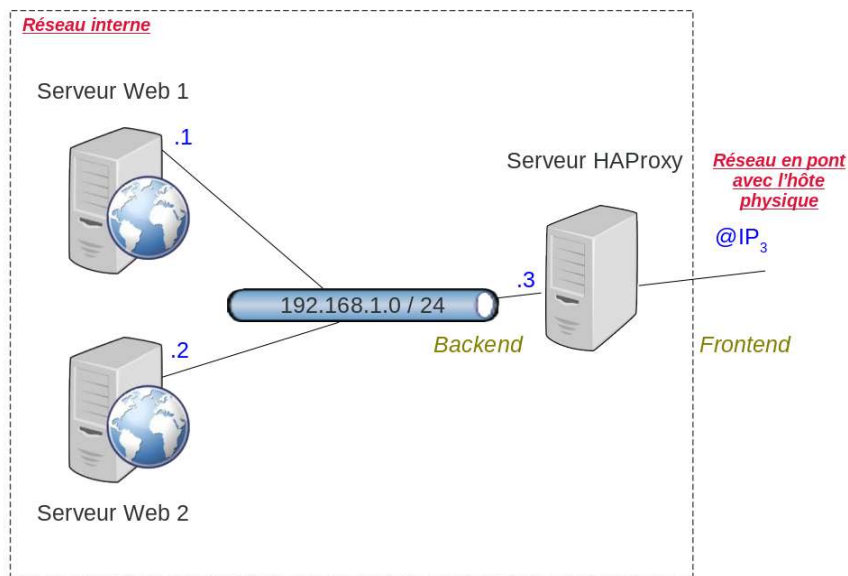
- insert et indirect font que le cookie se trouve entre le client et le load balancer sans être jamais transmis au serveur qui n'en a pas connaissance et ne pourrait pas le comprendre.
 - nocache permet de s'assurer qu'un cache en amont ne le stockera pas.
2. Tester le service une fois le fichier de configuration modifié comme demandé.
 4. **Que constatez-vous ?**

2.4. Configuration 2 : HAProxy avec deux réseaux locaux

Vous devez mettre œuvre la configuration expliquée ci-après.

Travail à faire :

1. Éteindre les trois serveurs pour les configurer en mode réseau interne de Virtualbox
2. Configurer leurs interfaces réseau en mode interne selon un nouveau schéma d'adressage
3. Sur la 3^e machine, configurer une carte réseau en mode pont et une autre dans le même réseau interne que les deux VMs serveurs Web.



Le serveur HAProxy doit faire office de « routeur » entre le réseau des serveurs web et le réseau d'exploitation utilisé par les clients (salariés de l'organisation, étudiants, utilisateurs externes via une DMZ, etc).

Les serveurs web ne doivent pas être joignables depuis le réseau local d'exploitation et le serveur HAProxy doit avoir deux cartes réseau (l'ip de l'interface vers sio.lan correspond à l'adresse IP fixe de HAProxy dans le réseau SIO).

4. Activer le routage sur HAProxy en mode temporaire (désactivé au prochain démarrage) :
`#echo 1 > /proc/sys/net/ipv4/ip_forward`
5. Donner la procédure permettant de garder la fonction de routage active sur le serveur HAProxy même en cas de redémarrage de cette machine
6. Effectuer l'ensemble des modifications de configuration nécessaires à la mise en œuvre de cette seconde configuration et recopier dans ce compte rendu les fichiers de configurations obtenus pour que cette solution soit fonctionnelle. Ajouter également des copies d'écran pour démontrer l'efficacité de cette mise en œuvre.
7. Bonus : effectuer l'ajout d'une section permettant la répartition de charge sur des services de type mysql

```
frontend mysql-in
  bind *:3306
  default_backend serveurs_mysql
backend serveurs_mysql
  balance roundrobin
  server webserver1 @private_IP1:3306 check
  server webserver2 @private_IP2:3306 check
```