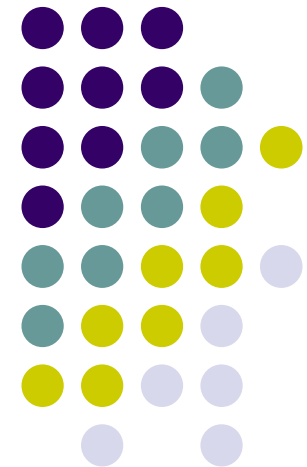


Heartbeat DRBD

Application aux vservers

29 mai 2008





Pourquoi vserver?

- Isoler les différents services
- Optimiser la mutualisation
- Utilise peu de ressources
- Performances
- Restauration rapide
- Migration facile



Idées de départ

- Pourquoi se contenter de ne clustériser que des services
- Un vserver est une arborescence de fichiers
- Mises à jour des services facilitées
- Regrouper les vservers sur la partition en raid1 ip
- Vserver + drbd = performance
- Fiabiliser la partie hardware



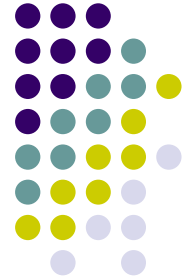
Problèmes contraintes ^{1/2}

- Besoin d'adapter heartbeat
 - Les vservers doivent être considérés comme un service
 - Comment faire du monitoring sur un vserver
 - Surveiller un service sur un vserver
- Attention aux services sur les vservers
 - Penser aux privilèges noyau des services (capabilities)
 - Services à problèmes : <http://linux-vserver.org>
 - Pas de services exotiques -> sécurité de l'ensemble du cluster



Problèmes contraintes 2/2

- Synchroniser les fichiers de configurations des vservers (non présents sur drbd)
- Gérer lors d'une bascule l'envoi de paquet arp
- Gérer le démarrage de Mon



Mise en œuvre - hardware

- Matériel utilisé à la délégation
 - 2 serveurs identiques
 - Bi-xeon dual core 2 Ghz
 - 4 disques 146 Go SAS hot swap
 - Raid 5
 - 4 Go de Ram
 - 2 ports gigabit
 - Alimentations redondantes
 - 1 serveur avec lecteur de sauvegarde AIT3



Mise en œuvre - software

- Serveurs hôtes
 - Debian etch

- Vservers
 - Debian etch
 - Debian lenny



Outils utilisés

- Hearbeat 1.2.5-3 (etch)
- Module DRBD 0.7.21-4
- Mon 0.99.2-9 (etch)
- Csync2 1.33-2 (etch)



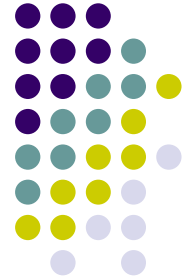
Configuration Csync2 ^{1/2}

- Clé partagée : csync2 -k filename
- Lancé par la crontab
- Fichier de config : /etc/csync2.cfg

```
#  
group dr  
{  
    host apollo12 apollo13;  
  
    key /etc/csync2.key_dr;  
  
    include /etc/vservers;  
    include /var/cache/vservers;  
    include /var/lib/vservers;  
    include /var/run/vservers.rev;
```



Configuration Csync2 ^{2/2}



```
action
{
    logfile "/var/log/csync2_action.log";
    do-local;
}

backup-directory /var/backups/csync2;
backup-generations 3;

auto younger;
}
noss! * *;
```



Configuration heartbeat



- /etc/heartbeat/haresources
 - Fichiers de configurations identiques sur le maître et l'esclave

```
apollo12 drbddisk::clusterdr Filesystem::/dev/drbd0::/srv::ext3 cluster  
MailTo::supervision.ssi@cnr-s-mrs.fr
```



Compilation DRBD

- Installer le package « module-assistant » sous debian
- Choisir le module à compiler, debian s'occupe du reste



Configuration DRBD

- /etc/drbd.conf

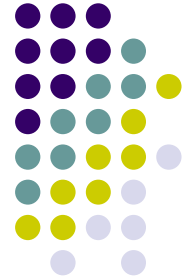
```
resource clusterdr {
  protocol    C;
  incon-degr-cmd "/sbin/halt -f";
  startup { wfc-timeout 240; degr-wfc-timeout 240; }
  disk { on-io-error detach; }
  syncer {
    rate      700000K;
    al-extents 257;
  }

  net {
    sndbuf-size 2048K;
    max-buffers 32768;
  }

  on apollo12 {
    device    /dev/drbd0;
    disk      /dev/sda11;
    address   192.168.1.1:7791;
    meta-disk internal;
  }

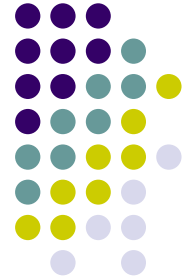
  on apollo13 {
    device    /dev/drbd0;
    disk      /dev/sda11;
    address   192.168.1.2:7791;
    meta-disk internal;
  }
}
```





Le script « cluster » 1/5

- Script présent dans /etc/init.d/
- Script permettant de lancer ou arrêter l'ensemble des vservers
- Possibilité de connaître l'états des vservers
- Lance automatiquement un script « arp » lors du démarrage de chaque vserver
- Lance ou arrête Mon dans certains cas



Le script « cluster » 2/5

- /etc/init.d/cluster

```
#
DRBDADMIN="/sbin/drbdsetup"
DRBD_MOUNT="/dev/drbd0"
INIT_CMD="/etc/init.d/util-vserver"
MON="/etc/init.d/mon"
MOUNT="/bin/mount"
UMOUNT="/bin/umount"
SHARED_MOUNTPOINT="/srv"
VSERVER_CMD="/usr/sbin/vserver"
VSERVERS=( hebdo hubble atlas ldap relay etats )
VSERVER_FAIL=0
i=0
```



Le script « cluster » 3/5

```
case "$1" in
start)
    $MON stop
    while [ ${VSERVERS[i]} ]
    do
        echo -n "Starting cluster:"
        /etc/vservers/${VSERVERS[i]}.sh pre-start
        $VSERVER_CMD ${VSERVERS[i]} start
        echo "."
        ((i += 1))
    done
    $MON start
;;
```


Le script « cluster » 4/5



```
stop)
  $MON stop
  while [ ${VSERVERS[i]} ]
  do
    echo -n "Stopping cluster:"
    $VSERVER_CMD ${VSERVERS[i]} stop
    echo "."
    ((i += 1))
  done
;;
```

Le script « cluster » 5/5



```
status)
  while [ ${VSERVERS[i]} ]
  do
    $VSERVER_CMD ${VSERVERS[i]} status >/dev/null
    status=$?
    if [ $status -eq 0 ]; then
      log_success_msg "${VSERVERS[i]} vserver is running."
    else
      log_failure_msg "${VSERVERS[i]} vserver is stopped."
    fi
    ((i += 1))
  done
  ;;
*)
  echo "Usage: /etc/init.d/cluster {start|stop|status}"
  exit 1
  ;;
esac

exit 0
```



Le script « arp »

- Un script pour chaque vserver, ex :

```
case $1 in
pre-start)
    /usr/lib/heartbeat/send_arp -i 200 -r 5 -p /var/lib/heartbeat/rsctmp/send_arp/send_arp-
10.10.10.x eth0 10.10.10.x auto 10.10.10.x ffffffff
    ;;
post-start)
    ;;
pre-stop)
    ;;
post-stop)
    ;;
*)
    echo $0 pre-start
    echo $0 pre-stop
    echo $0 post-start
    echo $0 post-stop
    ;;
esac
```



Installation du/des vservers

- Le volume DRBD est monté dans /srv
- `newvserver --vsroot /srv/vservers/ --hostname kma-et --domain cnrs-mrs.fr --ip 10.10.10.x/24 --dist etch --mirror http://ftp.fr.debian.org/debian/ --interface eth0`



Mon

- `/etc/mon/mon.cf`
- Monitoring des services
- Lancement de commandes sur événement
- Parfait pour surveiller les services sur les vservers
- Sous debian
 - Configuration : `/etc/mon/`
 - Ex de config : `/usr/share/doc/mon/examples/mon.cf`
 - Scripts : `/usr/lib/mon/`
- Penser à désactiver le démarrage automatique de Mon :
 - `update-rc.d -f mon remove`

Mon configuration



```
cfbasedir = /etc/mon
alertdir  = /usr/lib/mon/alert.d
mondir    = /usr/lib/mon/mon.d
startupalerts_on_reset = yes
maxprocs  = 20
histlength = 100
randstart = 10s
startupalerts_on_reset = yes
```

```
logdir = /var/log/mon/
dtlogging = yes
dtlogfile = dtlog
```

```
hostgroup vserver_hubble hubble.dr12.cnrs.fr
```

```
watch vserver_hubble
  service http
    interval 30s
    monitor http.monitor
    period wd {Mon-Fri} hr {2am-11pm}
      alert ha-up-down.alert -S "Service http on hubble down !" supervision@dr12.cnrs.fr
      upalert mail.alert -S "Service http on hubble up !" supervision@dr12.cnrs.fr
```



Quelques pièges

- Mises à jours ou restart des services : arrêter mon
- Mise à jour du noyau debian
 - Commencer par le nœud esclave
 - Après le reboot compiler DRBD
 - Heartbeat stop sur le maître
 - Débrancher lien gigabit et série
 - Mettre à jour le maître
- Mon ne doit pas être lancé avant les vservers



Conclusions

- Solution de « presque haute dispo »
- Stable pour le moment
- Faire attention aux mises à jours
 - Sur les hôtes (module drbd)
 - Sur les vservers (mon)
- Faire attention à la séquence de lancement heartbeat/mon



Références

- DRBD : <http://www.drbd.org>
- Heartbeat :
 - <http://www.linux-ha.org/Heartbeat>
 - <http://2005.jres.org/paper/91.pdf> (v 0.7)
- Mon : <http://www.debianhelp.co.uk/mon.htm>
- Vservers :
 - <http://linux-vserver.org>
 - http://fr.wikibooks.org/wiki/Linux_Vserver