Université de La Rochelle

Rapport de projet — Groupe n° 5

Dossier d'Architecture Technique

Auteurs :
Joris BERTHELOT
Laurent LE MOINE

Superviseurs :
Philippe HARRAND
Bertrand VACHON

Master ICONE 2011-2012

Table des matières

Introduction . . . 3
Postes de travail . . . 3
Code source . . . 3

I Infrastructure logicielle 4
0.1 Installation des paquets . . . 5
0.1.1 Configuration du proxy . . . 5
0.1.2 Script d'installation . . . 5
0.2 Réplication bas niveau . . . 6
0.2.1 Installation . . . 6
0.2.2 Configuration . . . 7
0.3 Déploiement des services . . . 9
0.3.1 Apache . . . 9
0.3.1.1 Installation . . . 9
0.3.1.2 Configuration . . . 9
0.3.2 MySQL . . . 10
0.3.2.1 Installation . . . 10
0.3.2.2 Configuration . . . 10
0.3.3 DNS . . . 11
0.3.4 LDAP . . . 14
0.4 Mise en place de Pacemaker et Corosync . . . 16
0.4.1 Présentation . . . 16
0.4.2 Installation . . . 17
0.4.3 Configuration . . . 17
0.4.3.1 Corosync . . . 17
0.4.3.2 Pacemaker . . . 18
0.4.4 Utilisation . . . 19

II Application Web 21
0.5 Présentation . . . 22
0.6 Code source . . . 22
0.6.1 LDAP . . . 22
0.6.1.1 Instance . . . 22
0.6.1.2 Récupération de toute les personnes . . . 23
0.6.1.3 Ajout d'une personne . . . 23
0.6.2 MySQL . . . 23

0.6.2.1 Instance . . . 24
0.6.2.2 Récupération de tous les produits . . . 24
0.6.2.3 Ajout d'un produit . . . 24

III Conclusion **25**
0.7 Difficultés rencontrés . . . 26
0.8 Retours sur échec . . . 26
0.9 Retours personnels . . . 27
0.9.1 Joris BERTHELOT . . . 27
0.9.2 Laurent LE MOINE . . . 27

IV Annexes **28**
.1 Bind . . . 29
.2 Pacemaker . . . 31
.3 Application Web . . . 32
.4 Bibliographie . . . 34
.4.1 LDAP . . . 34
.4.2 MySQL . . . 34
.4.3 Bind . . . 34
.4.4 Pacemaker, DRBD . . . 34

Introduction

Dans le cadre de notre formation Master Ingéniérie Informatique et de son Unité d'Enseignement Architecture : Conception et Gestion, nous avons réalisé un projet d'architecture réseau HA (High Availability) en 3 jours seulement.

Ce projet nous a permis de mettre en exergue nos connaissances récemment acquises lors des cours respectifs de la même UE mais aussi de reprendre et appliquer les concepts vus en TP la semaine auparavant.

Assigné comme table n° 5, nous avons utilisé les postes suivants :

Joris Berthelot (sera la machine « JB » dans le reste du rapport)

- Adresse IP : 10.192.10.23
- Host : mamba13

Laurent Le Moine (sera la machine « LLM » dans le reste du rapport)

- Adresse IP : 10.192.10.24
- Host : mamba14

Etant donné que ce projet fut réalisé en équipe, le code source des différents scripts et fichiers de configuration sont disponible sur Google Code + Subversion. Ainsi, vous pouvez à tout moment récupérer notre travail (ainsi que le code LaTeX du document) comme ceci :

```
svn export http://ulr-acg.googlecode.com/svn/trunk/ ulr-acg-src
```

Première partie

Infrastructure logicielle

Avant toute chose, vous devez savoir que l'ensemble des opérations décrites dans cette section sont à réaliser avec l'utilisateur `root`. Si vous lancez les scripts livrés avec le rapport sans être root, vous aurez droit à un gentil message d'erreur.

Nous avons aussi par ailleurs vidé et désactivé les tables de pare-feu afin de laisser toutes nos applications travailler sans avoir de gêne dans un premier temps :

```
# Vide les iptables
/sbin/chkconfig --del iptables
# Desactive le firewall
service iptables stop
# Active la synchronisation du temps sur le reseau
service ntpd start
```

0.1 Installation des paquets

Avant de commencer à configurer et déployer les services, nous aurons besoin d'installer un certain nombre de paquets afin de pouvoir parvenir à nos fins. Aussi divers que variés, nous avons scripté cette installation afin de faciliter la tâche.

0.1.1 Configuration du proxy

Il faudra auparavant configurer manuellement les paramètres du proxy si besoin afin de ne pas rendre le script d'installation inopérationnel. Pour se faire, veillez à bien changer les paramètres dans les Serveur mandataires (Système > Configuration > Serveurs mandataires) ainsi que rajouter les bons paramètres à Yum (proxy) :

```
# Configuration de Yum
vim /etc/yum.conf
```

0.1.2 Script d'installation

Pour configurer et installer les paquets manquants sur la machine, vous devez dans un premier temps savoir sur quelle interface réseau vous êtes relié au réseau extérieur (utilisez la commande `ifconfig`). Les scripts d'installation ont été développés seulement pour les machines « JB » et « LLM ».

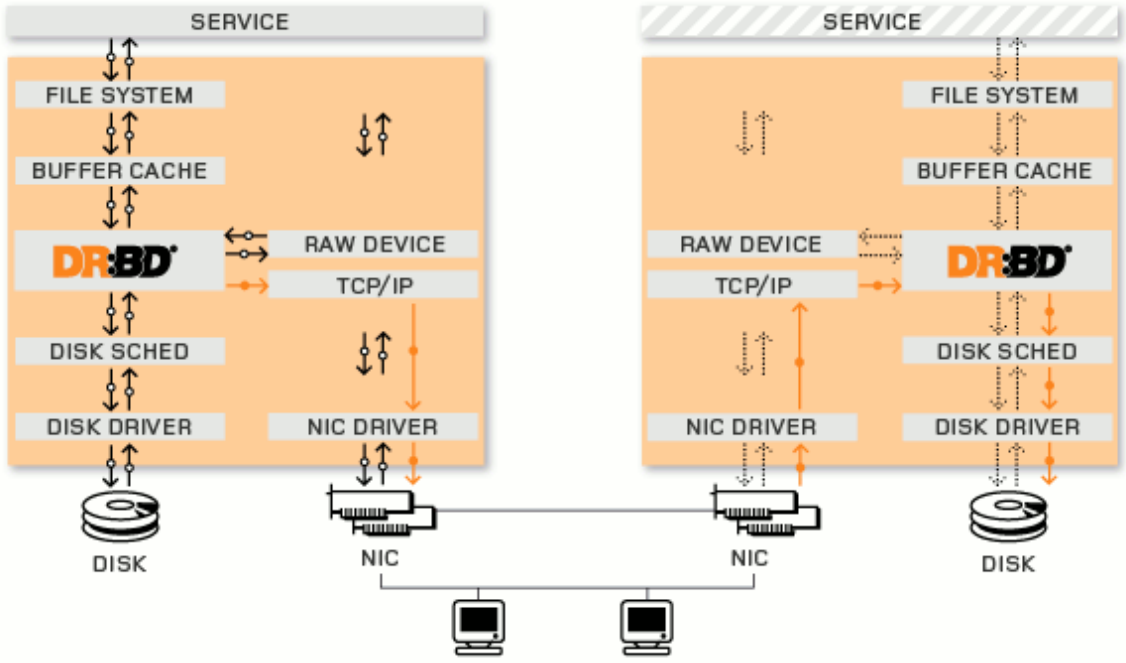
Pour faire évoluer ces scripts sur d'autres machines, il faudra modifier la valeur de certaines variables en début de script.

```
# Preparation des scripts
cd scripts
chmod u+x *
# Lancement du script d'installation
./setup.sh
```

0.2 Réplication bas niveau

La réplication bas niveau permet de créer non pas une redondance applicative mais directement sur le support des données applicatives (système de fichier). L'intêret à cela est d'éviter de configurer chaque service pour sa propre réplication (si existant) et d'aller droit au but en répliquant directement le volume sur lequel repose les données.

Voici un petit schéma (issu du site de DRBD) afin d'imager le concept :



0.2.1 Installation

Pour la réplication bas niveau (système de fichiers), nous avons utilisé DRBD : un logiciel permettant de faire de la réplication de données au sein d'une architecture en grappe. DRBD est assez complexe et fastidieux à mettre en place car il demande quelques notions assez poussées sur les volumes, leur synchronisation, la reprise sur failover, la notion de maître et escalve, etc.

Ce logiciel ne fonctionnait autrefois qu'en mode maître/esclave mais depuis les dernières versions, on peut partir sur une configuration maître/maître afin que les données soient bien synchronisées de manière bidirectionnelle.

```
# Installation de drbd
yum install -y drbd drbd-pacemaker drbd-udev
```

0.2.2 Configuration

Avant de configurer DRBD, il faut choisir où seront stockées nos données et comment s'y prendre. Dans une configuration idéale, il aurait fallu stocker nos données sur des partitions logiques cryptées et répliquées avec en RAID 15 mais nous n'avons pas eu le temps de nous soucier de cela donc nous avons vu simple : une partition logique dans un groupe de volumes.

Pour se faire, nous avons utilisé le disque `/dev/sdb` vide par défaut et nous y avons créé une partition de type Linux LVM :

```
fdisk /dev/sdb
> d
> <ENTER>
> n
> p
> <ENTER>
> <ENTER>
> <ENTER>
> t
> 8E
> w
# Creation d'un volume physique a partir de /dev/sdb1
pvcreate /dev/sdb1
# Creation d'un goupe de volumes
vgcreate ulr-acg /dev/sdb1
# Creation d'un volume dans le groupe de volumes
lvcreate -n ulr-data -L 1G ulr-acg
```

Maintenant que notre volume de données est prêt, nous décidons qu'il sera monté dans `/var/cluster` et que toutes les données applicatives seront dedans.

La configuration de DRBD peu s'avérer très simple mais permet un certain degré de complexité en fonction des architectures. La syntaxe est claire et s'apparente à celle du serveur de nom.
Voici la configuration que nous avons utilisé :

```
global {
    # Enables statistics usage
    usage-count yes;
}
common {
    # Maner to sync data, it's flagged as completed when both disks has written
    protocol C;
}
resource ulr-data {
    # Stores data meta-data in the volume
    meta-disk internal;
    # DRDB device
    device  /dev/drbd1;
    # Volume to work on
    disk    /dev/mapper/ulr--acg-ulr--data;
    syncer {
        # Hash method to check data integrity
        verify-alg sha1;
        # Network sync speed
        rate 100M;
    }
    # Allows both machine are primary
    net {
        allow-two-primaries;
    }
    # JB's machine
    on mamba13 {
        address     10.0.0.23:7789;
    }
    # LLM's machine
    on mamba14 {
        address     10.0.0.24:7789;
    }
}
```

DRBD implique qu'un lien réseau doit être établi en supplément des liens existants. Il est important de comprendre que DRBD utilise un réseau qui lui est propre afin d'y transférer les données.

Une fois configuré, nous pouvons lancer DRBD :

```
/etc/init.d/drbd start
```

Une fois le service DRBD démarré, nous devons initialiser le disque afin que le système de réplication soit opérationnel :

```
# Creation du disque a partir du volume
drbdadm --force create-md ulr-data
# Activation du volume au sein de DRBD
drbdadm up ulr-data
# Declaration de notre disque comme maitre
drbdadm -- --overwrite-data-of-peer primary ulr-data
```

Pour vérifier que DRBD fonctionne bien, il suffit de voir son état en faisant la commande sui-

vante :

```
cat /proc/drbd
```

Si tout va bien, on peut créer le système de fichier sur le disque maître et le monter dans `/var/cluster` :

```
# Creation d'un systeme de fichier de type EXT4
mkfs.ext4 /dev/drbd1
# Montage du systeme de fichier
mount /dev/drbd1 /var/cluster
```

Voici comment créer un volume logique contenant un système de fichier de type EXT4 réparti sur plusieurs machines avec DRBD.

0.3 Déploiement des services

Dans cette partie, il est important de comprendre que lors de la mise en place d'une architecture en grappe avec des noeuds répliqués, il faut toujours un noeud de référence, surtout dans une architecture active/passive comme celle que nous allons mettre en place. Suivant la machine sur laquelle vous allez lancer les scripts, il faudra ou non déployer les données.

0.3.1 Apache

0.3.1.1 Installation

Pour installer Apache, il vous faudra très simplement lancer son script d'installation : `scripts/apache.sh`. Ce script va essayer de stopper Apache, vérifier l'intégrité de son fichier de configuration et si il ne concorde pas avec le notre, il va le remplacer. Ensuite, selon si vous êtes le noeud référence de la grappe ou non, il va vous proposer de déployer les données.

0.3.1.2 Configuration

La configuration d'Apache est très légèrement spécifique dû à notre application Web qui utilise Silex mais sinon rien de bien particulier ormis la configuration du service Apache `server-status` qui doit être disponible pour Pacemaker.

Il faudra donc bien décommenter le bout de code suivant à la fin du fichier :

```
1 <Location /server-status>
2     SetHandler server-status
3     Order deny,allow
4     Deny from all
5     # Only from localhost where Pacemaker runs
6     Allow from 127.0.0.1
7 </Location>
```

Enfin, spécifier le `DocumentRoot` à `/var/cluster/www/org/tp/g1b5/web` et y ajouter les règles de réécritures suivantes afin de faire fonctionner notre application :

```
<IfModule mod_rewrite.c>
    Options -MultiViews
    RewriteEngine On
    # All requests which are not a file
    RewriteCond %{REQUEST_FILENAME} !-f
    # Except this request
    RewriteCond %{REQUEST_URI} !=/server-status
    RewriteRule ^ index.php [L]
</IfModule>
```

Une fois configuré, nous pouvons lancer Apache :

```
/etc/init.d/httpd start
```

0.3.2 MySQL

0.3.2.1 Installation

L'installation de MySQL se fait très facilement grâce au script d'installation : `scripts/mysql.sh`. Le script arrête le serveur MySQL, charge le fichier de configuration (`confs/mysql/my.cnf`) à la place de l'ancien, démarre ensuite le service mysqld, puis crée les utilisateurs et peuple la base si nous sommes sur le premier noeud de la grappe.

0.3.2.2 Configuration

Le fichier de configuration de MySQL est court, mais il est important dans notre cas de bien préciser le chemin du répertoire ou seront stockées les bases, à savoir dans notre cluster, dans le répertoire `/var/cluster/mysql`. De même, il est aussi important de préciser que l'adresse utilisée par le serveur est celle de l'adresse de la grappe et non celle de la machine : `10.192.10.50`.

```
[mysqld]
# On definit le repertoire ou seront stockees les donnees sur le cluster
datadir=/var/cluster/mysql
# On definit l'adresse IP utilisee par le serveur MySQL
bind-address=10.192.10.50
# Chemin du fichier de socket pour les connexions locales
socket=/var/lib/mysql/mysql.sock
user=mysql
# On desactive les liens symboliques pour ameliorer la securite du serveur
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Une fois configuré, nous pouvons lancer MySQL :

```
/etc/init.d/mysqld start
```

Nous devons aussi définir le mot de passe de l'utilisateur "root", qui est le même sur les deux machines. Cela ce fait simplement grâce à la commande :

```
mysqladmin password <mot_de_passe>
```

Il faut ensuite créer un utilisateur "tpuser", et lui allouer des droits de sélection sur toutes les machines du réseau `10.192.10.0`, afin de permettre l'accès à la base "projet_hd" depuis une machine de ce réseau.

```
CREATE USER 'tpuser'
IDENTIFIED BY 'tpuser';

GRANT ALL PRIVILEGES
ON *.* TO 'root'@'%';

CREATE DATABASE projet_hd;

GRANT SELECT PRIVILEGES
ON projet_hd.* TP 'tpuser'@'10.192.10.%';
```

Il faut aussi créer puis initialiser la table "product", qui contient les données sur notre stock. Elle contient les champs "id", "name", "price" et "quantity".

```
CONNECT projet_hd;

CREATE TABLE product (
    id SMALLINT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    price SMALLINT NOT NULL,
    quantity SMALLINT NOT NULL,
    PRIMARY KEY (id)
);

INSERT INTO product(name, price, quantity) VALUES(
    ('A Games of Thrones', 11, 9),
    ('A Clash of Kings', 11, 2),
    ('A Storm of Swords', 11, 12),
    ('A Feast for Crows', 11, 19),
    ('A Dance with a Dragon', 11, 3)
);
```

0.3.3 DNS

Pour commencer, il faut installer le paquet "bind". Comme pour les autres services, nous avons fait un script qui configure le serveur bind. Il copie tout simplement les fichiers de configuration que nous avons écrit, puis lance le service `named`.

Nous avons choisi une configuration maître/esclave classique plutôt que d'utiliser DRBD et Pacemaker, car il n'y a pas de données à répliquer comme avec l'annuaire LDAP ou MySQL, il y a seulement les fichiers de zone.
De même, pour les adresses des serveurs DNS, nous avons utilisé l'adresse des machines plutôt que celle de la grappe : le maître est donc la machine LLM, ayant l'adresse `10.192.10.24`, et l'esclave la machine JB, `10.192.10.23`.

Le fichier de configuration de bind est simple, et nous avons seulement rajouté la partie concernant les fichiers de zone propre à nos besoins ; ils contiennent l'adresse des serveurs DNS est celle des machines correspondantes, mais l'adresse de "g1b5.tp.org" et celle des différents services est celle de la grappe (`10.192.10.50`).

Nous n'avons pas pu donner d'adresses IPv6 aux différents services car Corosync ne gère pas la cohabitation IPv4/IPv6. Les services ont donc seulement une adresse IPv4, seul les serveurs DNS ont une adresse IPv6.

Le fichier de configuration de Bind :

```
options {
    listen-on port 53 { ANY; };
    listen-on-v6 port 53 { ANY; };
    directory     "/var/named";
    dump-file     "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query     { ANY; };
    recursion yes;

    dnssec-enable yes;
    dnssec-validation yes;
    dnssec-lookaside auto;

    version "Try again! ;)";

    bindkeys-file "/etc/named.iscdlv.key";

    managed-keys-directory "/var/named/dynamic";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
    type hint;
    file "named.ca";
};

zone "g1b5.tp.org" IN {
    type master;
    file "/etc/named/g1b5.tp.org.zone";
};


zone "0.0.0.0.9.c.0.0.0.0.0.0.0.0.d.f.ip6.arpa" {
    type master;
    file "/etc/named/rv6.g1b5.tp.org.zone";
};

zone "10.192.10.in-addr.arpa" {
    type master;
    file "/etc/named/rv4.g1b5.tp.org.zone";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

Les autres fichiers de configuration (zones et zones inverses) sont disponibles dans l'annexe .1.

0.3.4 LDAP

Il faut tout d'abord installer le paquet "openldap", puis lancer le script d'installation, qui remplace les fichiers de configuration par défaut par les notre.
Pour configurer OpenLDAP, nous avons utilisé l'ancienne méthode, avec le fichier `/etc/openldap/slapd.conf`, plutôt que la nouvelle méthode qui est très peu documentée. Il faut bien penser à supprimer le répertoire `/etc/openldap/slapd.d/` car sinon OpenLDAP ne prend pas en compte le fichier `slapd.conf`.

Le fichier `/etc/openldap/ldap.conf` contient seulement le suffixe de notre base :

```
1 #/etc/openldap/ldap.conf
2 # LDAP Defaults
3 #
4
5 # See ldap.conf(5) for details
6 # This file should be world readable but not world writable.
7
8 BASE    dc=g1b5,dc=tp,dc=org
9 URI     ldap://localhost
```

Le fichier `/etc/openldap/slapd.conf` est quant à lui plus fourni :

```
#/etc/openldap/slapd.conf
include   /etc/openldap/schema/core.schema
include   /etc/openldap/schema/cosine.schema
include   /etc/openldap/schema/inetorgperson.schema
include   /etc/openldap/schema/nis.schema

pidfile     /var/run/openldap/slapd.pid
argsfile    /var/run/openldap/slapd.args

# On definit une base de donnees bdb, ayant pour suffixe g1b5.tp.org, et comme root Ma
# Le mot de passe du root est hache en md5
database  bdb
suffix  "dc=g1b5,dc=tp,dc=org"
rootdn  "cn=Manager,dc=g1b5,dc=tp,dc=org"
rootpw  {MD5}nsoz4g836Njg9I6ijrlC9w==

# On definit le chemin du repertoire qui contiendra les bases sur le cluster
directory /var/cluster/ldap

# Les differents indices a maintenir dans la base de donnees
index default eq
index objectClass eq
index cn,name,surname,givenname eq,sub

# Definitions des access list, seul le manager peut ecrire.
access to attrs=userPassword
    by self write
    by anonymous auth
    by dn="cn=Manager,dc=g1b5,dc=tp,dc=org" write
    by * none

access to *
    by self write
    by dn="cn=Manager,dc=g1b5,dc=tp,dc=org" write
    by * read
```

Ensuite le script peuple la base sur demande à l'aide d'un fichier `.ldif`. Nous stockons dans la base nos employés, qui sont actuellement au nombre de 2 (vos dévoués) :

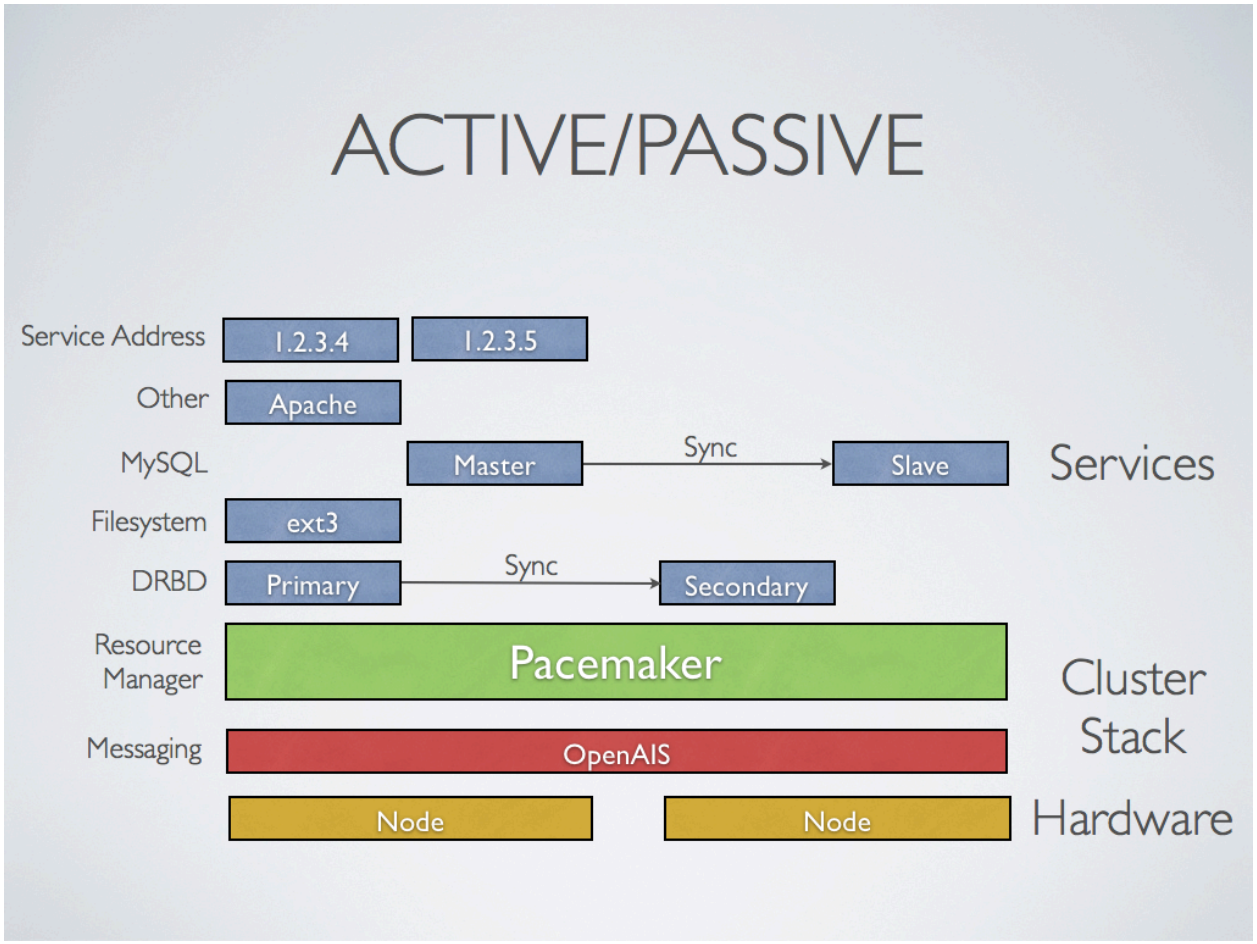
```
dn: ou=employee,dc=g1b5,dc=tp,dc=org
ou: employee
objectClass: organizationalUnit
description: Employee of G1B5 Co LTD

dn: cn=Berthelot,ou=employee,dc=g1b5,dc=tp,dc=org
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: organizationalPerson
cn: Berthelot
sn: Berthelot
givenName: Joris

dn: cn=Le_Moine,ou=employee,dc=g1b5,dc=tp,dc=org
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: organizationalPerson
cn: Le_Moine
sn: Le_Moine
givenName: Laurent
```

0.4 Mise en place de Pacemaker et Corosync

Dans une architecture HA, le système doit pouvoir constamment suivre son état et celui de son entourage afin de décider si il doit basculer (= "failover") ou non vers un noeud de secours. Pour cela, avec Fedora, nous avons choisi d'utiliser le service Pacemaker couplé à Corosync car ils nous ont semblé très complets et très professionnels.

0.4.1 Présentation

Pour notre projet, nous avons choisi de mettre en place une architechure en grappe avec un système actif/passif. Ci-dessous un schéma de l'architechure :



Les « node » correspondent à nos machines (JB et LLM), la couche de messagerie OpenAIS correspondra au service Corosync qui est un projet dérivé d'OpenAIS. Pour les couches supérieures à Pacemaker, nous les avons vu précédemment.

Corosync est un moteur de clustering mettant en oeuvre une messagerie de service sur réseau en utilisant une adresse et un port multicast.

0.4.2 Installation

```
# Installation de pacemaker et de corosync
yum install -y pacemaker corosync
```

0.4.3 Configuration

0.4.3.1 Corosync

Avant toute chose, voici les adresses IP nécessaires :

Adresse IP multicast : 239.0.0.1

Port multicast : 6800

Réseau de la grappe : 10.192.10.0

Adresse IP de la grappe : 10.192.10.50

On va commencer par la configuration de Corosync qui est simpliste. Pour cela, on prends le fichier de configuration par défaut et on va simplement y changer les adresses IP comme ceci :

```
CONF="/etc/corosync/corosync.conf"
cp -f $CONF.example $CONF
sed -i.bak "s/.*mcastaddr:.*/mcastaddr:\ 239.0.0.1/g" $CONF
sed -i.bak "s/.*mcastport:.*/mcastport:\ 6800/g" $CONF
sed -i.bak "s/.*bindnetaddr:.*/bindnetaddr:\ 10.192.10.0/g" $CONF
```

Ensuite, on va signaler à Corosync de charger le plugin Pacemaker afin qu'il puisse "travailler" avec :

```
cat << EOT > /etc/corosync/service.d/pcmk
service {
        # Load the Pacemaker Cluster Resource Manager
        name: pacemaker
        ver:  1
}
EOT
```

Une fois configuré, nous pouvons lancer Corosync :

```
/etc/init.d/corosync start
```

0.4.3.2 Pacemaker

Pacemaker offre une invite de commande propre à lui-même (comme les switchs Cisco) que nous pouvons lancer en tapant `crm`. Il est intéressant de savoir qu'en interne, la configuration de Pacemaker est formattée en XML car suite à quelques erreurs de notre part, nous avons pu constater des erreurs de validation d'entrée contre des Schémas XML.

Afin de vous épargner la configuration à la main de Pacemaker, nous pouvons exporter et injecter directement une configuration donnée :

```
# Exporter une configuration
crm configure show > /abs/path/conf.latest
# Exporter au format XML
crm configure show xml > /abs/path/conf.latest.xml
# Injecter une configuration
crm configure load replace /../ulr-acg/confs/pcmk/conf.work
```

Mais **avant cela**, vous devez démarrer Pacemaker et vider sa configuration actuelle (si existante) :

```
/etc/init.d/pacemaker start
cibsadmin -E --force
```

Voici donc notre fichier de configuration pour Pacemaker (la version initiale qui marchait correctement) :

```
primitive ClusterIP ocf:heartbeat:IPaddr2 params ip="10.192.10.50" \
    cidr_netmask="24" op monitor interval="5s"
primitive FS ocf:heartbeat:Filesystem params \
    device="/dev/drbd/by-res/ulr-data" \
    directory="/var/cluster" fstype="ext4"
primitive UlrData ocf:linbit:drbd params drbd_resource="ulr-data" \
    op monitor interval="30s"
primitive WebApp ocf:heartbeat:apache params \
    configfile="/etc/httpd/conf/httpd.conf" op monitor interval="10s"
primitive MySQL lsb:mysqld
primitive LDAP lsb:slapd
ms UlrDataClone UlrData meta master-max="1" master-node-max="1" \
    clone-max="2" clone-node-max="1" notify="true"
colocation WebApp-with-FS inf: WebApp FS
colocation FS-with-UlrDataClone inf: FS UlrDataClone:Master
colocation WebApp-with-ClusterIP inf: WebApp ClusterIP
colocation MySQL-with-ClusterIP inf: MySQL ClusterIP
colocation LDAP-with-ClusterIP inf: LDAP ClusterIP
order MySQL-after-FS inf: FS MySQL
order LDAP-after-FS inf: FS LDAP
order WebApp-after-FS inf: FS WebApp
order WebApp-after-ClusterIP inf: ClusterIP WebApp
order FS-after-UlrDataClone inf: UlrDataClone:promote FS:start
property $id="cib-bootstrap-options" \
    dc-version="1.1.6-1.fc14-b379478e0a66af52708f56d0302f50b6f13322bd" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
```

Dans ce fichier, on peut y entrevoir des déclarations de ressources comme une IP "flottante" (`10.192.10.50`) sur laquelle le monde extérieur va se connecter aux services que contient la grappe, des ressources de battement de coeur pour un système de fichier, les serveurs Apache, MySQL, LDAP et enfin la ressource de réplication bas niveau DRBD.

Mais ce n'est que la déclaration des ressources, ensuite viennnent les dépendances de "colocation" qui obligent deux ressources données à toujours être de paire et enfin les ordres de démarrage des ressources en fonction des dépendances. Par exemple, Apache ne peut pas démarrer avant que le système de fichier soit opérationnel.

Les autres fichiers de configuration (celui qui nous a valu la présentation catastrophique et l'actif/actif) sont disponibles en annexe n° .2.

0.4.4 Utilisation

Pour ~~admirer~~ surveiller le bon fonctionnement de Pacemaker, il y a la commande `crm_mon`.

En ce qui concerne la gestion des noeuds ou des ressources, il faut utiliser la commande `crm`. Voici quelques exemples :

```
# Verifier le statut d'une ressource
crm resource statut Apache
# Redemarrer une resource
crm resource restart MySQL
# Deplacer une ressource sur un autre noeud
crm ressource move FS mamba14
# Voir le statut d'un noeud
crm node status mamba13
# Mettre un noeud en standby (simulation du failover)
crm node standby mamb14
# Et le rendre disponible a nouveau
crm node online mamba14
```

Cette console est vraiment trés puissante et offre une souplesse réelle digne d'un outil professionnel. Toutes les actions d'administration sont scriptables soit directement en Bash soit en injectant des fragments XML aux bons endroits via d'autres commandes dédiées.

Deuxième partie

Application Web

0.5 Présentation

L'application Web demandée devait permettre de d'afficher et d'insérer des données dans l'annuaire LDAP et dans la base de données MySQL.

Développée en 3h seulement, l'application repose sur PHP 5.3 et utilise des frameworks récents comme Silex (Symfony 2), Zend Framework 2 et enfin le framework CSS bootstrap de Twitter. Encore une fois, un maximum d'outils facile à mettre en oeuvre et puissants en quelques heures seulement.

Une capture d'écran de notre application est présente en annexe n° .3.

0.6 Code source

Dans cette partie du rapport, nous allons vous montrer comment utiliser les outils choisis pour arriver à nos fins.

0.6.1 LDAP

Pour la partie LDAP, nous avons utilisé la brique LDAP du Zend Framework 2.

0.6.1.1 Instance

```php
<?php
use Zend\Ldap\Ldap;
use Zend\Ldap\Exception as LdapException;
// ...
$app['ldap'] = $app->share(function() {
    try {
        $ldap = new Ldap(array(
            'host'              => 'ldap://localhost',
            'username'          => 'cn=manager,dc=g1b5,dc=tp,dc=org',
            'password'          => 'pa$$wd',
            'bindRequiresDn'    => true,
            'accountDomainName' => 'g1b5.tp.org',
            'baseDn'            => 'ou=employee,dc=g1b5,dc=tp,dc=org',
        ));
        return $ldap->bind();
    } catch (LdapException $e) {
        var_dump($e->getMessage());
    }
});
```

0.6.1.2 Récupération de toute les personnes

```php
<?php
use Zend\Ldap\Ldap;
use Zend\Ldap\Exception as LdapException;
// ...
$app->get('/ldap.html', function(Silex\Application $app) {
    $filter = Filter::equals('objectClass', 'person');
    $results = $app['ldap']->search($filter, 'ou=employee,dc=g1b5,dc=tp,dc=org',
    Ldap::SEARCH_SCOPE_SUB);
    return $app['twig']->render('ldap.twig', array(
        'dir'       => true,
        'entries'   => $results
    ));
});
```

0.6.1.3 Ajout d'une personne

```php
<?php
use Zend\Ldap\Ldap;
use Zend\Ldap\Exception as LdapException;
// ...
$app->post('/ldap.html', function(Silex\Application $app) {
    $name = $app->escape($app['request']->get('name'));
    $entry = array();
    Attribute::setAttribute($entry, 'cn', $name);
    Attribute::setAttribute($entry, 'sn', $name);
    Attribute::setAttribute($entry, 'objectClass', 'inetOrgPerson');
    $app['ldap']->add('cn=' . $name . ' ,ou=employee,dc=g1b5,dc=tp,dc=org', $entry);
    $filter = Filter::equals('objectClass', 'person');
    $results = $app['ldap']->search($filter, 'ou=employee,dc=g1b5,dc=tp,dc=org',
    Ldap::SEARCH_SCOPE_SUB);
    return $app['twig']->render('ldap.twig', array(
        'dir'       => true,
        'entries'   => $results
    ));
});
```

0.6.2 MySQL

Concernant MySQL, nous n'avons pas utilisé d'ORM comme Doctrine mais nous avons simplement utilisé PDO.

0.6.2.1 Instance

```php
<?php
$app['mysql'] = $app->share(function() {
   try {
       $pdo_options[\PDO::ATTR_ERRMODE] = \PDO::ERRMODE_EXCEPTION;
       return new \PDO('mysql:host=localhost;dbname=projet_hd', 'root', 'pa$$wd',
       $pdo_options);
   } catch (\PDOException $e) {
       var_dump($e->getMessage());
   }
});
```

0.6.2.2 Récupération de tous les produits

```php
<?php
$app->get('/mysql.html', function(Silex\Application $app) {
    $response = $app['mysql']->query('SELECT * FROM `product`')
                ->fetchAll(\PDO::FETCH_ASSOC);
    return $app['twig']->render('mysql.twig', array(
        'db'        => true,
        'entries'   => $response
    ));
});
```

0.6.2.3 Ajout d'un produit

```php
<?php
$app->post('/mysql.html', function(Silex\Application $app) {
    $req = $app['mysql']->prepare('INSERT INTO product(name, price, quantity)
        VALUES(:name, :price, :quantity)');
    $req->execute(array(
        'name'      => $app->escape($app['request']->get('name')),
        'price'     => $app->escape($app['request']->get('price')),
        'quantity'  => $app->escape($app['request']->get('qte'))
    ));
    $response = $app['mysql']->query('SELECT * FROM `product`')
                ->fetchAll(\PDO::FETCH_ASSOC);
    return $app['twig']->render('mysql.twig', array(
        'db'        => true,
        'entries'   => $response
    ));
});
```

Troisième partie

Conclusion

0.7 Difficultés rencontrés

Durant ce projet, la contrainte majeure était de réaliser une telle architechure en 3 jours seulement sans avoir aucune expérience dans le domaine. Nous avons souhaité nous orienter vers des choix fonctionnels et matures plutôt qu'universitaires et souvent peu digne du monde professionnel.

Le premier facteur de difficulté aura été l'abscence d'indications concernant le choix des technologies car nous avons réellement été soumis à un environnement pragmatique où nous jouons le rôle des responsables SI vers qui on vient se conseiller suite à une problématique bien présente. Le choix et la décision des technologies est une chose mais ce qui est le plus chronophage est de faire de la veille-expresse sur les produits et les solutions existantes sur le marché.

Une fois les choix réalisés, il a fallu se documenter pour savoir implémenter au mieux possible l'infrastructure logicielle. Nous ne vous cacherons pas que nous avons lu 90% de documentation en anglais durant ces 3 jours. De l'anglais pas toujours facile à comprendre de part ses termes techniques mais surtout de part la notion nouvelle que nous découvrions sur le tas.

La difficulté suivante fut liée aux machines elles-même : impossible de prévoir que tout va fonctionner du premier coup, le fait que les machines aient besoin d'un serveur mandataire pour accèder au Web rajoute quelques contraintes sur les configurations mais surtout le manque de maîtrise complet sur les distributions Fedora nous aura fait perdre un peu de temps non négligeable.

Malgré tout cela, nous sommes parvenus automatiser de manière relativement complète l'installation, la configuration et le déploiement de notre projet à l'aide de Subversion et de scripts Bash.

0.8 Retours sur échec

Après 3 jours de recherche, de développement et de tests acharnés sur les différentes briques de notre architechure, nous avons échoué lors de la présentation de notre projet. En voici les raisons majeures :

1. Manque d'expérience avec les technologies utilisées
2. Manque d'organisation et de rigueur dans les procédures
3. Manque de temps (implique du stress)
4. Analyses sur tests défaillants pas assez complètes

D'une certaine manière, tous ces facteurs sont liés car ils s'entre-croisent et s'encouragent et nous le savions mais pas c'est pas toujours évident d'avoir le temps de prendre le recul nécessaire afin de mieux repartir.

Après des heures intensives de travail, nous avions réussi à faire tourner notre architechure active/passive en simulant des failovers et des failback avec succès mais sans tester tous les cas possibles et surtout les cas d'évaluation. Suite à ce succès précipité, nous avons voulu optimiser la configuration de Pacemaker afin de la simplifier mais nous nous sommes rendu compte après analyse que cette nouvelle configuration ne répondait plus au contraintes d'ordre de basculement des services lors du failover.

Dans le stress et la précipitation, nous avons voulu essayer une architechure active/active car nous pensions que le type d'architechure active/passive ne répondait pas à la demande de l'évaluation (analyse d'échec pas assez poussée) mais après quelques heures d'installation et de configuration, cette nouvelle monture n'a pas été un succès à cause d'un soucis de synchronisation de volume avec DRBD bien que la nouvelle configuration active/active de Pacemaker était déjà prête.

0.9 Retours personnels

0.9.1 Joris BERTHELOT

Bien que le projet ne fut pas un succès au moment de l'évaluation, nous pouvons tout de même avancer que notre architechure a bel et bien tourné et surtout sous les yeux d'un de nos évaluateur quelques heures auparavant. C'est donc tout de même avec une certaine déception que nous vous remettons ce dossier sachant que son contenu n'a pas pu fonctionner comme nous l'aurions souhaité au moment idéal.

Très personnellement, j'ai éprouvé un réel plaisir à m'activer intensément pour la bonne réussite du projet et même si nous n'avons pas réussi, j'ai appris énormément de choses sur un domaine qui m'attirait depuis quelques temps déjà. En PME ou en auto-entreprise, il est vraiment peu probable de devoir réaliser de genre d'architechure en grappe donc je suis très agréablement surpris d'avoir abordé cela en cycle universitaire.

0.9.2 Laurent LE MOINE

J'ai trouvé ce projet intéressant, et ancré dans les problématiques propres aux entreprises, ce qui change par rapport aux projets habituels.

L'installation des différents services ne nous ont pas posé de problèmes, à part LDAP qui ne prenait pas en compte le fichier de configuration et utilisait la nouvelle méthode, mais une fois ceci compris nous n'avons pas eu d'autres problèmes (autres services maîtrisés).

Les gros problèmes sont bien sur venus de DRBD et Pacemaker. Nous étions arrivés à une configuration relativement fonctionnelle et efficace, bien qu'elle n'était pas parfaite. Le passage à la configuration actif/actif a été une erreur.

Avec le recul, nous aurions d'abord dû préparer une réplication basique, mais fonctionnelle, puis ensuite s'occuper de mettre en place DRBD et PaceMaker, plus efficaces et plus professionnels. Nous sommes partis sur le postulat que dans une entreprise il vaut mieux mettre en place ce genre de systèmes, plutôt que la réplication basique de MySQL et de LDAP. Ces systèmes ont l'avantage d'être autonomes.

Joris BERTHELOT
joris@berthelot.tel
&
Laurent LE MOINE
laurent.le.moine17@gmail.com

Quatrième partie

Annexes

.1 Bind

Fichier de configuration du serveur esclave :

```
options {
    listen-on port 53 { ANY; };
    listen-on-v6 port 53 { ANY; };
    directory     "/var/named";
    dump-file     "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query     { ANY; };
    recursion yes;

    dnssec-enable yes;
    dnssec-validation yes;
    dnssec-lookaside auto;

    version "Try again! ;)";

    bindkeys-file "/etc/named.iscdlv.key";

    managed-keys-directory "/var/named/dynamic";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
    type hint;
    file "named.ca";
};

zone "g1b5.tp.org" IN {
    type slave;
    file "g1b5.tp.org.zone";
    masters { 10.0.0.24; };
};


zone "0.0.0.0.9.c.0.0.0.0.0.0.0.0.d.f.ip6.arpa" {
    type slave;
    file "rv6.g1b5.tp.org.zone";
    masters { 10.0.0.24; };
};

zone "10.192.10.in-addr.arpa" {
    type slave;
    file "rv4.g1b5.tp.org.zone";
    masters { 10.0.0.24; };
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

Fichier de configuration de zone :

```
$ttl 3600
@        IN    SOA        ns.g1b5.tp.org. admin.g1b5.tp.org. (
                      2011102001
                      10800
                      3600
                      604800
                      38400 )
;
@        IN    NS        ns.g1b5.tp.org.
@        IN    NS        ns2.g1b5.tp.org.
;
@        IN     A        10.192.10.50
;
ns        IN     A        10.192.10.24
ns        IN     AAAA    fd00:0:c9::501
;
ns2        IN     A        10.192.10.23
ns2        IN     AAAA    fd00:0:ca:4::502
;
www        IN     A        10.192.10.50
;
sql        IN     A        10.192.10.50
;
ldap    IN     A        10.192.10.50
```

Fichier de configuration de zone inverse IPv4 :

```
; 10.192.10.0/24
$TTL 3600
$ORIGIN 10.192.10.in-addr.arpa.
@    IN SOA @ admin.g1b5.tp.org. (
        2011102002    ; Serial number (YYYYMMdd)
        10800        ; Refresh time
        3600        ; Retry time
        604800        ; Expire time
        38400        ; Default TTL (bind 8 ignores this, bind 9 needs it)
)

; Name server entries
    IN    NS        ns.g1b5.tp.org.
    IN    NS        ns2.g1b5.tp.org.
; IPv4 PTR entries

; Subnet #1
$ORIGIN 10.192.10.in-addr.arpa.

24    IN  PTR        ns.g1b5.tp.org.
23    IN  PTR     ns2.g1b5.tp.org.
50    IN  PTR     g1b5.tp.org.
50    IN  PTR     sql.g1b5.tp.org.
50    IN    PTR        ldap.g1b5.tp.org.
50    IN    PTR        www.g1b5.tp.org.
```

Fichier de configuration de zone inverse IPv6 :

```
; fd00:0:c9::/64
$ORIGIN 0.0.0.0.9.c.0.0.0.0.0.0.0.0.d.f.ip6.arpa
$TTL 3600
@    IN SOA @ admin.g1b5.tp.org. (
        2011102003    ; Serial number (YYYYMMdd)
        10800        ; Refresh time
        3600        ; Retry time
        604800        ; Expire time
        38400        ; Default TTL (bind 8 ignores this, bind 9 needs it)
)

; Name server entries
    IN    NS        ns.g1b5.tp.org.
    IN    NS        ns2.g1b5.tp.org.
; IPv6 PTR entries
501    IN  PTR        ns.g1b5.tp.org.
502    IN  PTR     ns2.g1b5.tp.org.
```

.2 Pacemaker

Fichier de configuration de Pacemaker qui a voulu notre chute :

```
primitive MySQL lsb:mysqld
primitive LDAP lsb:slapd
primitive ClusterIP ocf:heartbeat:IPaddr2 params ip="10.192.10.50" \
cidr_netmask="24" op monitor interval="5s"
primitive ClusterFS ocf:heartbeat:Filesystem params \
device="/dev/drbd/by-res/ulr-data" directory="/var/cluster" fstype="ext4"
primitive Volume ocf:linbit:drbd params drbd_resource="ulr-data" \
op monitor interval="30s"
primitive Apache ocf:heartbeat:apache params \
configfile="/etc/httpd/conf/httpd.conf" op monitor interval="10s"
ms VolumeClone Volume meta master-max="1" master-node-max="1" \
clone-max="2" clone-node-max="1" notify="true"
group Services ClusterFS ClusterIP Apache MySQL LDAP
colocation Services-on-VolumeClone inf: Services VolumeClone:Master
order Services-after-VolumeClone inf: VolumeClone Services:start
property $id="cib-bootstrap-options" \
    dc-version="1.1.6-1.fc14-b379478e0a66af52708f56d0302f50b6f13322bd" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    cluster-recheck-interval=5m \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100" \
    migration-threshold=2 \
    failure-timeout=30s
```

Fichier de configuration de Pacemaker pour une architechure active/active (non testé) :

```
primitive MySQL lsb:mysqld
primitive LDAP lsb:slapd
primitive ClusterIP ocf:heartbeat:IPaddr2 params ip="10.192.10.50" \
cidr_netmask="24" clusterip_hash="sourceip" op monitor interval="5s"
primitive ClusterFS ocf:heartbeat:Filesystem params \
device="/dev/drbd/by-res/ulr-data" directory="/var/cluster" fstype="gfs2"
primitive Volume ocf:linbit:drbd params drbd_resource="ulr-data" \
op monitor interval="30s"
primitive Apache ocf:heartbeat:apache params \
configfile="/etc/httpd/conf/httpd.conf" op monitor interval="10s"
ms VolumeClone Volume meta master-max="1" master-node-max="1" \
clone-max="2" clone-node-max="1" notify="true"
group Services ClusterFSClone ClusterIPClone ApacheClone MySQL LDAP
colocation Services-on-VolumeClone inf: Services VolumeClone:Master
order Services-after-VolumeClone inf: VolumeClone Services:start
clone ClusterIPClone ClusterIP meta globally-unique="true" \
clone-max="2" clone-node-max="2"
clone ApacheClone Apache
clone ClusterFSClone ClusterFS
property $id="cib-bootstrap-options" \
    dc-version="1.1.6-1.fc14-b379478e0a66af52708f56d0302f50b6f13322bd" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    cluster-recheck-interval=5m \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100" \
    migration-threshold=2 \
    failure-timeout=30s
```

.3 Application Web

Capture d'écran de la page d'accueil de notre application Web :

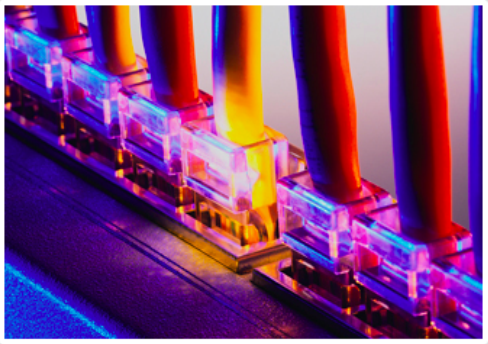
Home Annuaire Serveur de données

Bienvenue sur le site de G1B5 !

> The Domain Name Server (DNS) is the Achilles heel of the Web. The important thing is that it's managed responsibly.
> — Tim Berners-Lee

Bienvenue sur notre petit site de projet Master 2 ICONE ACG.
Ca se voit peut-être pas comme ça à première vue mais cette page sera jamais down, si si, je vous mets au défi de faire tomber notre serveur Apache plus de 2 min !

Vous êtes 127.0.0.1 connecté sur local.g1b5.



10.192.10.23 Joris Berthelot
10.192.10.24 Laurent Le Moine

Développé avec amour, Silex et bootstrap en 2h... et c'est valide HTML5 B-)

.4 Bibliographie

Ici sont répertorié tous les liens utiles pour l'avancement du projet.

.4.1 LDAP

http://www.openldap.org http://articles.mongueurs.net/magazines/linuxmag65.html

.4.2 MySQL

http://dev.mysql.com/doc/refman/5.5/en/index.html

.4.3 Bind

http://www.bind9.net/ http://www.redfoxcenter.org/serveur/bind.html

.4.4 Pacemaker, DRBD

http://www.clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/
http://www.drbd.org/users-guide/re-drbdconf.html
http://http://www.drbd.org/users-guide/re-drbdadm.html
http://en.wikipedia.org/wiki/Computer_cluster