

Cluster MySQL con réplica circular

MySQL Cluster Replication with Circular

*Yusdiel Rodríguez Amoros¹, Marcos Bagarotti Marin¹, Alecnys Perera Lumpuis¹,
¹Centro Nacional de Sanidad Agropecuaria. Carretera de Jamaica y Autopista Nacional,
San José de las Lajas, Mayabeque, Cuba. correo: yrodriguez@censa.edu.cu,
mbagarotti@censa.edu.cu, alecnys@censa.edu.cu

Resumen

El Centro Nacional de Sanidad Agropecuaria cuenta una red de datos, de más de 326 usuarios, los que contribuyen a la investigación en el tema de salud animal y sanidad vegetal, utilizando las tecnologías de la informática y las comunicaciones. El departamento de Informática, desarrolla software de alto valor agregado para el trabajo de la institución y durante este proceso, la elección del gestor de Base de Datos es un aspecto importante a tener en cuenta para el rendimiento de una aplicación informática. Desplegar soluciones informáticas en un entorno de producción, utilizando un único servidor de Base de Datos no es recomendable, porque se encuentra expuesto a fallas en el servicio debido a cortes eléctricos, problemas de hardware, mantenimiento, entre otros. La utilización de la tecnología de clúster en los servicios de redes a través de herramientas de software libre como son: Pacemaker, Corosync, OpenAIS y MySQL permite incrementar la disponibilidad del servicio. Este trabajo aborda la aplicación de esta tecnología en la institución, lo que ha permitido alcanzar una alta disponibilidad de los servicios de base de datos para las aplicaciones informáticas.

Abstract

The National Center for Animal and Plant Health (CENSA) has a data network of over 326 users, contributing to research in the field of animal and plant health, using information technologies and communications. The IT department develops a high value-added software for the work of the institution, and during the process, choosing the database manager is an important factor to take into account the performance of a computer application aspect. Deploying solutions in a production environment, using a single database server, is not recommended, because it is exposed to service failures due to power outages, hardware problems, maintenance, among others. The use of technology cluster network services through free software tools such as: Pacemaker, Corosync, MySQL and OpenAIS allows increasing service availability. This paper approaches the

application of this technology in the institution, which has helped to achieve a high availability of services for database applications.

Introducción

El término clúster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardware comunes y que se comportan como si fuesen una única computadora. Hoy en día desempeñan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno[1].

La tecnología clúster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

En un clúster de alta disponibilidad de aplicación si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios que han fallado en cualquiera de las otras. Cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la original. Esta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdida de datos, y evita molestias a los usuarios, que no tienen por qué notar que se ha producido un problema[2].

Desarrollo

Para la implementación y desarrollo de un clúster de base de datos MySQL, se utilizaron dos servidores profesionales Hewlett-Packard (HP), con microprocesador Intel Xeon y 16GB de memoria RAM y la distribución de Linux openSUSE versión 12,1. A continuación se explica brevemente las herramientas utilizadas para realizar el trabajo. MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Este es muy utilizado en aplicaciones web como MediaWiki, Drupal o phpBB, en plataformas (Linux/Windows, Apache, MySQL, PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con este gestor. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones[3].

Pacemaker es un administrador de clúster. Este permite obtener el máximo rendimiento de los servicios, haciendo uso de los mensajes en la infraestructura del clúster, permite realizar cualquier tipo de redundancia incluyendo Activo/Activo Activo/Pasivo, etc. Pacemaker esta compuesto por tres partes fundamentales, el núcleo del clúster que permite los mensajes y la pertenencia al clúster (OpenAIS), un conjunto de script para iniciar, detener y monitorear los recursos y un cerebro que procesa y reacciona a los eventos del clúster[4].

Para lograr un clúster completamente funcional con este tipo de características es necesario instalar los software Pacemaker y Corosync en las dos computadoras que se quieren clústerizar como se muestra a continuación:

```
#zypper in corosync pacemaker
Obteniendo los datos del repositorio...
Leyendo los paquetes instalados...
mysql no se encontró como nombre de paquete, probando en prestaciones.
Resolviendo dependencias...
Los siguientes paquetes NUEVOS van a ser instalados:
  clúster-gluu corosync crmsh fence-agents libcorosync4 libdlm libdlm3 libesmtp libestr0
libfreebl3 libglue2 libgthread-2_0-0 libnet1 libpacemaker3 libqb0 libsensors4
  libsnmp30 libsoftokn3 libXss1 mariadb mozilla-nss mozilla-nss-certs openhpi Open-
PMI pacemaker pacemaker-cli perl-TimeDate psmisc pssh python-dateutil python-pex-
pect
  python-pssh python-pycurl python-tk resource-agents rsyslog snmp-mibs syslog-ser-
vice tcl tk xhost
Los siguientes paquetes recomendados fueron seleccionados automáticamente:
  crmsh fence-agents libdlm
Se sugiere el siguiente paquete, pero no será instalado:
  graphviz
41 nuevos paquetes a instalar.
Tamaño total a descargar: 18,2 MiB. Después de la operación, se utilizarán 105,6 MiB
adicionales.
¿Desea continuar? [s/n/? mostrar todas las opciones] (s):s
```

Si termina la instalación satisfactoriamente, el siguiente paso es verificar la conexión entre las dos maquinas, utilizando el comando ping. Para configurar OpenAIS es necesario establecer un puerto, una dirección multicas y la dirección de dominio del clúster. En este trabajo se utilizó 226.94.1.1 como dirección multicast y 4000 como puerto del

servicio. Establecidos estos parámetros se puede iniciar el servicio de Corosync de la siguiente manera:

```
# rcopenais start
```

```
Starting OpenAIS/Corosync daemon (corosync): starting... OK
```

Pacemaker tiene una herramienta que proporciona una interfaz de línea de comandos llamada *crm*, que permite la administración y configuración del clústeres de alta disponibilidad a partir de cero[5].

```
node hera \
```

```
    attributes standby="off"
```

```
node zeus \
```

```
    attributes standby="off"
```

```
primitive ClusterIP ocf:heartbeat:IPaddr2 \
```

```
    params ip="172.16.0.200" cidr_netmask="32" \
```

```
    op monitor interval="30s"
```

```
primitive MySQL ocf:heartbeat:mysql \
```

```
    params                binary="/usr/bin/mysqld_safe"                user="mysql"
```

```
    pid="/var/run/mysql/mysqld.pid" socket="/var/run/mysql/mysql.sock" \
```

```
    op start interval="0" timeout="120" \
```

```
    op stop interval="0" timeout="120" \
```

```
    op promote interval="0" timeout="120" \
```

```
    op demote interval="0" timeout="120"
```

```
clone MysqlClone MySQL \
```

```
    meta target-role="Started"
```

```
location cli-prefer-ClusterIP ClusterIP \
```

```
    rule $id="cli-prefer-rule-ClusterIP" inf: #uname eq zeus
```

```
property $id="cib-bootstrap-options" \
```

```
    dc-version="1.1.6-9971ebba4494012a93c03b40a2c58ec0eb60f50c" \
```

```
    clúster-infraestructure="openais" \
```

```
    expected-quorum-votes="2" \
```

```
    stonith-enabled="false" \
```

```
    no-quorum-policy="ignore"
```

```
rsc_defaults $id="rsc-options" \
```

```
    resource-stickiness="100"
```

Esta configuración muestra el estado funcional, donde todos los nodos comparten una misma dirección IP, se puede apreciar en la directiva `primitive MySQL ocf:heartbeat:mysql` que el servicio de MySQL se encuentra clusterizado. En la siguiente captura también se muestra el estado del clúster

#crm_mon

Last updated: Thu Mar 13 15:40:16 2014

Last change: Thu Mar 13 09:10:39 2014 by root via crm_attribute on hera

Stack: openais

Current DC: hera - partition with quorum

Version: 1.1.6-9971ebba4494012a93c03b40a2c58ec0eb60f50c

2 Nodes configured, 2 expected votes

3 Resources configured.

=====
Online: [hera zeus]

ClusterIP (ocf::heartbeat:IPaddr2): Started zeus

Clone Set: MysqlClone [MySQL]

Started: [hera zeus]

El clúster garantiza la disponibilidad del servicio, pero la integridad y la disponibilidad de la información hay que garantizarla con otro tipo de herramientas. En este caso, para que la información se mantenga en la misma base de datos, se utilizó el método de *Réplica circular*, que permite tener la misma información en las dos bases de datos. La replicación funciona de la siguiente forma: el servidor esclavo se inicia con una copia exacta de los datos del servidor principal, a continuación se activa el registro binario en el principal y el esclavo se conecta a este periódicamente y comprueba los cambios efectuados en el registro binario desde la última vez que se conectó[6]. Para mas información de la *Réplica circular* se puede consultar el manual de referencia de MySQL5 en esta dirección <http://dev.mysql.com/doc/refman/5.1/en/>.

Conclusión

La tecnología clúster posee muchos beneficios para las aplicaciones de función crítica ya que permite un modelo de escalabilidad de alto impacto. En el centro de datos del CENSA se aplica esta tecnología obteniendo resultados satisfactorios. Las aplicaciones web que tienen una misión crítica para el trabajo realizado en la institución, han logrado una alta disponibilidad utilizando este tipo de tecnología



www.sociedadlainformacion.com

Edita:



Director: José Ángel Ruiz Felipe
Jefe de publicaciones: Antero Soria Luján

D.L.: AB 293-2001

ISSN: 1578-326x