

**RADIO OBSERVATORIO DE JICAMARCA**

---

**INSTITUTO GEOFÍSICO DEL PERÚ**

**IMPLEMENTACIÓN DE UN SISTEMA  
REDUNDANTE EN EL SERVIDOR JRO-RT**

**PHILLYPS JERGGER BRAVO OJEDA  
TECNOLOGÍA DE LA INFORMACIÓN**

**MARZO 2010**

## **RESUMEN**

En el presente reporte técnico se muestra la implementación de un sistema redundante en el servidor JRO-RT, uno de los principales servidores del Radio Observatorio de Jicamarca (ROJ) porque es el encargado de manejar el tráfico de datos entre la red local e Internet (Firewall).

Ahora se cuenta con un sistema redundante automatizado en el servidor JRO-RT que permite tener una alta disponibilidad en la red local, reduciendo el tiempo a milisegundos en restaurar automáticamente el servidor JRO-RT ante cualquier falla de software y/o hardware. Los usuarios no perciben la existencia de esta solución, el funcionamiento es transparente y el administrador es notificado del evento a través de un correo electrónico.

## ÍNDICE

<b>1</b>	<b>INTRODUCCIÓN</b> .....	<b>4</b>
<b>2</b>	<b>DESARROLLO</b> .....	<b>4</b>
2.1	Componentes de Ultra Monkey .....	5
2.1.1	Heartbeat .....	5
2.1.2	Ldirectord .....	5
2.1.3	MON (Service Monitoring Daemon) .....	5
2.2	Implementación del Sistema Redundante .....	5
2.2.1	Instalación de UltraMonkey .....	5
2.2.2	Configuración de Heartbeat .....	5
2.2.3	Activación .....	7
2.2.4	Caída .....	7
<b>3</b>	<b>RESULTADOS</b> .....	<b>7</b>
<b>4</b>	<b>CONCLUSIONES</b> .....	<b>7</b>
<b>5</b>	<b>RECOMENDACIONES</b> .....	<b>7</b>

# IMPLEMENTACIÓN DE UN SISTEMA REDUNDANTE EN EL SERVIDOR JRO-RT

## 1 INTRODUCCIÓN

Los sistemas tolerantes a fallas o sistemas redundantes se encargan de realizar el mismo proceso en más de una estación, si por algún motivo alguna dejara de funcionar o colapsara, inmediatamente otro tendría que ocupar su lugar y realizar las tareas del anterior. Esta tecnología es conocida como *cluster* de alta disponibilidad.

En la red local del ROJ existen muchos componentes necesarios para que esta funcione; cuantos más componentes intervengan aumentan las probabilidades que algo falle. Estos problemas pueden ocurrir en el propio servidor: fallos de fuentes de alimentación, tarjetas de red, etc. (que son componentes internos), y en la infraestructura necesaria para que el servidor pueda funcionar: componentes de conectividad, acceso a Internet, sistema eléctrico (componentes externos). Esta solución afrontará con éxito cualquier falla que presente el servidor JRO-RT relacionado con los componentes internos.

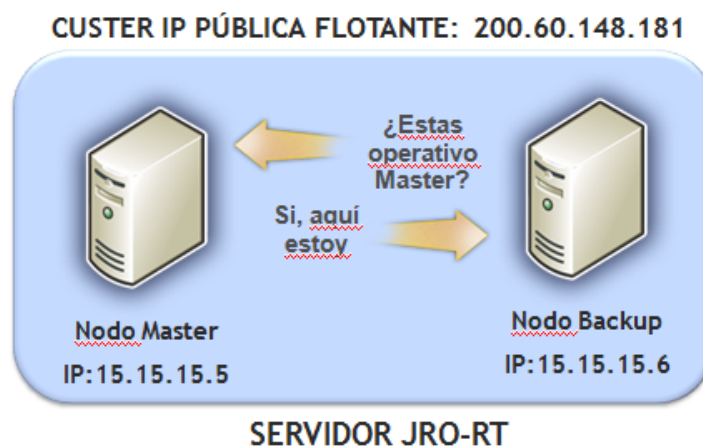
Ante la falta del suministro eléctrico ya se cuenta con un sistema redundante que consta de un generador eléctrico y varios UPS que garantiza el funcionamiento continuo de los servidores y las computadoras críticas del ROJ.

Por ello se viene trabajando en los sistemas redundantes de los componentes de conectividad y acceso a Internet que serán detallados en el próximo reporte trimestral.

## 2 DESARROLLO

El *cluster* de alta capacidad, que se encuentra instalado en el ROJ, está conformado por dos servidores —llamado también nodos— el cuál es el encargado de satisfacer todas las peticiones de servicios y/o conectividad solicitadas desde la red local a Internet o viceversa.

El trabajo que desempeña uno de los nodos —el nodo backup— consiste en el continuo monitoreo del correcto funcionamiento del otro nodo —nodo master— para sustituir sus funciones. En caso de presentar alguna falla levantará el IP público flotante, pasando en este momento a ser el nodo activo. Si el nodo master se vuelve a recuperar, el nodo backup pasará a ser nuevamente el nodo pasivo.



**Figura 1** Componentes del cluster en el servidor JRO-RT.

Actualmente existen diversas soluciones de cluster de alta disponibilidad, una es Ultra Monkey, el cual fue implementado en el ROJ para brindar un sistema redundante al servidor JRO-RT.

## 2.1 Componentes de Ultra Monkey

Ultra Monkey es un proyecto que integra distintas herramientas de Software Libre para conseguir alta disponibilidad en redes de área local, a continuación se mencionará los principales componentes.

### 2.1.1 Heartbeat

Funciona enviando periódicamente un paquete al otro nodo, que si no llegara, indicaría que un nodo no está disponible, por lo tanto se sabe que el nodo ha caído y se toman las medidas necesarias.

### 2.1.2 Ldirectord

Está encargado de llevar la administración de las tablas de *forwarding* del *kernel*, el cual será arrancado y detenido cuando sea necesario por Heartbeat.

### 2.1.3 MON (Service Monitoring Daemon)

Mon es un software para la monitorización del sistema, el cuál permite definir una serie de alarmas y acciones a ejecutar cuando un servicio y/o nodo deja de funcionar. Las dos principales acciones que va realizar son: asignar la IP pública flotante y enviar un correo electrónico al administrador.

## 2.2 Implementación del Sistema Redundante

La redundancia la realizaremos a los PROXY, y para ello se asume que en ambos servidores ya se encuentra instalado SQUID.

### 2.2.1 Instalación de UltraMonkey

En ambos nodos se debe realizar estos procesos:

- Descargar e instalar en el orden respectivo: pils, stonith, ipvsadm, ldirectord que las podremos encontrar en <http://rpmfind.net/linux/>.

Estas librerías incorporan en la conectividad el soporte de manejar las conexiones entrantes y salientes, también proveen el soporte de alta disponibilidad al Kernel.

- Luego descargar e instalar de <http://dag.wieers.com/> el repositorio heartbeat de acuerdo a la arquitectura del servidor, para luego finalizar con un yum install heartbeat.

### 2.2.2 Configuración de Heartbeat

#### 2.2.2.1 Configuración de los nombres de los nodos

Asigne nombres a los nodos (archivo "network").

Nombre del archivo: etc/sysconfig/network

HOSTNAME=master.jro.igp.gob.pe

Luego hay que generar la resolución del nombre a IP (archivo "hosts").

Nombre del archivo: /etc/hosts

127.0.0.1 localhost.localdomain localhost

```
15.15.15.5    master.jro.igp.gob.pe    master
15.15.15.6    master.jro.igp.gob.pe    backup
```

### 2.2.2.2 Configuración del archivo de configuración del heartbeat

Configuración standard en todos los nodos. No existe diferencia entre master y backup, el primero en activarse será el master.

```
Nombre del archivo: /etc/ha.d/ha.cf
#Log Local
logfile /var/log/ha-log
logfacility loca10
#Tiempo expresado en segundos para la revisión
keepalive 2
#Tiempo expresado en segundos en declarar que esta caído
deadtime 10
#Tarjeta de red a enviar mensajes
bcast eth0
udpport 694
#Declaración de los nodos que intervienen
node master.jro.igp.gob.pe
node backup.jro.igp.gob.pe
```

### 2.2.2.3 Configuración de las llaves

La llave permite la autorización de aceptar la consulta y a responder, todos los nodos debe de contar con la misma llave.

```
Nombre del archivo: /etc/ha.d/authkeys
# Autenticacion ssh2
auth 1
# Llave a compartir
1 sha1 Roj2010IT
Configurar permisos:
chmod 600 /etc/ha.d/authkeys
```

### 2.2.2.4 Configuración de los recursos

Personalizando los recursos a asignarse:

Nombre del archivo: /etc/ha.d/haresources

```
backup.jro.igp.gob.pe
```

```
15.15.15.6/24/eth0
```

```
Mailto::jro-redes@jro.igp.gob.pe::Master Activo
```

```
master.jro.igp.gob.pe
```

```
15.15.15.5/24/eth0
```

```
Mailto::jro-redes@jro.igp.gob.pe::Caida Master
```

### 2.2.3 Activación

Es importante la sincronización entre los participantes del cluster, para evitar conflictos de las asignaciones de los recursos (IP flotante).

Activación permanente:

```
chkconfig --level 35 heartbeat on  
service heartbeat start
```

### 2.2.4 Caída

Para simular una caída observamos el IP flotante, que lo tiene el nodo master. A partir de ahora tomaremos la IP flotante como el IP de acceso al servicio del PROXY.

Activamos el monitoreo en el backup:

```
tail -f /var/log/ha-log
```

Apague el master:

```
halt
```

Verifique la asignación de la IP Flotante en backup

```
ifconfig eth0
```

Verifique la recepción del correo con la alerta y comprobar el acceso a las redes privadas y públicas

## 3 RESULTADOS

- Reducción de tiempo a milisegundos en restaurar automáticamente el servidor JRO-RT ante cualquier falla de software y hardware.

## 4 CONCLUSIONES

Con la implementación de un sistema redundante en el servidor JRO-RT permite:

- Reducción de tiempo a milisegundos en restarurar automáticamente el servidor JRO-RT ante cualquier falla de software y hardware, los usuarios no perciben la existencia de esta solución, el funcionamiento es transparente y el administrador es notificado con un correo del evento.
- Contar con todas las herramientas ya instaladas para poder aplicar balance de carga y contar con un plan de escalabilidad, por si necesitamos más capacidad, sólo bastará añadir más nodos.
- Facilitar el mantenimiento preventivo, reparaciones y actualizaciones del servidor JRO-RT en un horario de labor normal sin obstruir las labores de los usuarios que van hacer uso de la red local.

## 5 RECOMENDACIONES

- Ampliar la capacidad de los generadores eléctricos que es el principal sistema redundante eléctrico que contamos.
- Instalar e implementar un sistema redundante de los componentes de conectividad y de conexión a Internet.