

Enxeñaría de Infraestruturas Informáticas

Grao en Enxeñaría Informática



UNIVERSIDADE DA CORUÑA

PRÁCTICAS SOBRE CLUSTERS DE ALTA DISPOÑIBILIDADE E BALANCEO DE CARGA

Xoán C. Pardo
xoan.pardo@udc.gal

PRACTICA 3

Balanceo de carga e medida do rendemento

Obxectivo

Nesta práctica vamos realizar dúas actividades diferentes:

1. Na primeira parte vamos configurar un balanceador de carga entre varias VMs, é dicir, vamos simular un cluster de balanceo de carga.
2. Na segunda parte vamos realizar unha análise do rendemento do cluster anterior usando un *benchmark*.

Xustificación da práctica

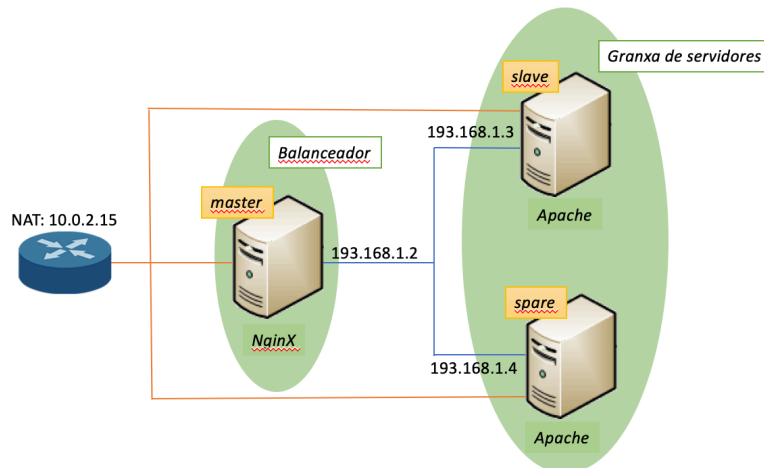
Haberá que entregar unha memoria en formato PDF na que se xustifique a realización de cada parte desta práctica. Os detalles sobre o que hai que incluír na memoria danse ao final de cada parte. A entrega da memoria será a través dunha tarefa no Moodle. A data límite de entrega indicárase no seu momento.

Proxecto Vagrant da práctica 3

Utiliza o proxecto Vagrant que podes descargar ou clonar con git desde aquí: https://github.com/gei-eii/practica_3.git. Unha vez descargado, e antes de usalo, terás que editar o Vagrantfile e o script bootstrap.sh para **cambiarlles os nomes ás VM e usar a nomenclatura obrigatoria** indicada no enunciado da práctica 0.

PRIMEIRA PARTE: Configuración dun balanceador de carga

Nesta primeira parte vamos configurar un balanceador de carga entre varias VMs coa intención de simular un cluster de balanceo de carga. Usaremos un cluster con 3 VMs conectadas entre si, no que unha funcionará como balanceador e as outras dúas como unha granxa de servidores web. Usaremos Nginx (<https://www.nginx.com>) como software do balanceador e Apache para os servidores web. A configuración é a que se mostra na seguinte figura:



Os pasos a seguir son os seguintes:

Configuración de Nginx

Para instalar e configurar Nginx segue os pasos seguintes:

- Instala Nginx na VM que vai facer o papel de balanceador de carga¹. Os pasos a seguir son os seguintes:
 - Importa a clave do repositorio software

```
lb$ sudo wget http://nginx.org/keys/nginx_signing.key
lb$ sudo apt-key add nginx_signing.key
```

- Engade o repositorio de Nginx ao final do ficheiro `/etc/apt/sources.list`

```
# Repositorios de Nginx para Ubuntu
deb https://nginx.org/packages/ubuntu/ bionic nginx
deb-src https://nginx.org/packages/ubuntu/ bionic nginx
```

- Instala NginX

```
lb$ sudo apt-get install nginx
```

A configuración de Nginx realízase mediante arquivos situados no directorio `/etc/nginx/conf.d/`. Por defecto os arquivos presentes nese directorio son importados dentro dun contexto `http` desde o ficheiro principal de configuración de Nginx (`/etc/nginx/nginx.conf`).

- Crea no directorio anterior un arquivo chamado `xcpm1920-lb.conf` co seguinte contido:

```
upstream xcpm1920-webfarm {
    server xcpm1920-slave weight=2;
    server xcpm1920-spare;
}

server {
    server_name xcpm1920-lb;
    listen 80;
    root /var/www;
    error_log /var/log/nginx/xcpm1920-lb.error.log;

    location / {
        proxy_pass http://xcpm1920-webfarm;
    }
}
```

Esta configuración especifica o seguinte²:

- Bloque `server` → crea un *servidor virtual* denominado `xcpm1920-lb` que xestiona o tráfico recibido no porto 80 (HTTP). O directorio raíz (o que toman como base as directivas `location`) será `/var/www` e o *log* de erros o indicado coa directiva `error_log`.

¹ Cando sexa necesario indicar a VM que fai o papel de balanceador usaremos o *prompt* `lb$` nos exemplos.

² Podes consultar o manual de Nginx en <https://docs.nginx.com/nginx/>

- Bloque `upstream` → define o grupo de servidores entre os que se repartirá o tráfico. Como algoritmo de balanceo de carga vaise utilizar un *round-robin* (algoritmo por defecto) ponderado. O primeiro servidor recibirá o dobre de carga (directiva `weight=2`) que o segundo.
- Bloque `location` → configura a onde se encamiña cada solicitude HTTP en base á ruta indicada na URI da solicitude. Neste caso calquera solicitude que teña como URI `http://xcpm1920-lb` será encamiñada (directiva `proxy_pass`) ao grupo de servidores definidos na directiva `upstream` anterior.
- Reinicia o proceso Nginx para cargar a nova configuración.

```
lb$ sudo nginx -s reload
```

Comprobación da configuración de NginX

- Comproba que cando accedemos cun navegador web (p.e. `links`) á dirección `http://xcpm1920-lb`, por cada dous desvíos ao primeiro servidor prodúcese un ao segundo. Ten en conta o seguinte para realizar a comprobación:
 - O servidor Apache ten que estar levantado nos dous servidores da granxa.
 - Para poder diferenciarlos entre si, cada servidor da granxa ten que servir unha páxina web diferente.
 - Como non temos DNS, hai que configurar no balanceador o `/etc/hosts` para que o nome do *servidor virtual* se resolva á IP do balanceador.

Configuracións adicionais

- Configura o balanceador de maneira que funcione da maneira seguinte³:
 - A URI `http://xcpm1920-lb` reparte as solicitudes entre o *slave* e o *spare*, de cada 5 solicitudes 3 irán para o *slave* e 2 para o *spare*.
 - A URI `http://xcpm1920-lb/slave` mostra a páxina do *slave*.
 - A URI `http://xcpm1920-lb/spare` mostra a páxina do *spare*.
 - A URI `http://xcpm1920-rr` reparte as solicitudes alternativamente (*round-robin*) entre o *slave* e o *spare*.
- Comproba que o funcionamento do balanceador é o solicitado.

³ A información sobre a configuración das localizacións e algoritmos de balanceo podes consultala aquí <https://docs.nginx.com/nginx/admin-guide/web-server/web-server/#locations> e aquí <https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/#method>

Xustificación da primeira parte

Inclúe na memoria de xustificación desta práctica as capturas de pantalla que mostren a execución (comando e resultado) da seguinte secuencia de comandos:

```
#Contido do(s) ficheiro(s) de configuración nginx
lb$ cat /etc/nginx/conf.d/*.conf

#Funcionamento do balanceador

# 5 accesos a -lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb

# 5 accesos máis a -lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb
lb$ curl http://xcpm1920-lb

# acceso a -lb/slave e -lb/spare
lb$ curl http://xcpm1920-lb/slave
lb$ curl http://xcpm1920-lb/spare

# 6 accesos a -rr
lb$ curl http://xcpm1920-rr
lb$ curl http://xcpm1920-rr
lb$ curl http://xcpm1920-rr
lb$ curl http://xcpm1920-rr
lb$ curl http://xcpm1920-rr
lb$ curl http://xcpm1920-rr
```

Explica BREVEMENTE o comportamento que observaches no cluster ao executar a secuencia de comandos.

SEGUNDA PARTE: Medida do rendemento da granxa de servidores

Nesta segunda parte vamos comprobar o rendemento da granxa de servidores configurada na primeira parte. Para medir as prestacións da granxa usaremos o *benchmark* *ab* para o servidor Apache (<https://httpd.apache.org/docs/2.4/programs/ab.html>). Os pasos son os seguintes:

- Instala o *benchmark* no balanceador.

```
lb$ sudo apt-get install apache2-utils
```

- Realiza 10000 solicitudes de 100 en 100 (simultáneas).

```
lb$ ab -k -c 100 -n 10000 http://xcpm1920-lb/
```

- Consulta a documentación⁴ do *benchmark* para comprender a saída que proporciona e o significado dos parámetros que se lle pasan.
- Fai máis probas usando tamén as outras URIs configuradas no balanceador na primeira parte da práctica e probando diferentes valores para os parámetros *-c* e *-n*. Fíxate nas diferencias/similitudes nos resultados e asegúrate de entendelos.

Xustificación da segunda parte

De todas as métricas que proporciona o *benchmark* *ab*, vamos fixarnos nas seguintes:

- O tempo que tarda en se executar o test.
- O número de solicitudes por segundo que procesa.
- A velocidade de transferencia.
- O tempo de conexión total (mínimo, media e máximo).
- Porcentaxe de solicitudes procesadas (50%, 75%, 90%, 95%, 98%, 100%).

Vamos realizar medidas destas métricas na URI `http://xcpm1920-lb/` escollendo un valor para *n* (o número de solicitudes) e usando 3 valores diferentes para o parámetro *c* (número de solicitudes concurrentes): 1, 10 e 100.

Inclúe na memoria de xustificación desta práctica o seguinte:

- Unha táboa como a seguinte cos resultados de medir as métricas que nos interesan para cada combinación dos valores de *n* e *c* (en total 3 táboas). Haberá que repetir a execución do *benchmark* un mínimo de 10 veces para cada combinación, para que os resultados sexan estatisticamente significativos. Calcula para cada métrica os valores de media e desviación estándar do conxunto de execucións e engádeo ao final da táboa.

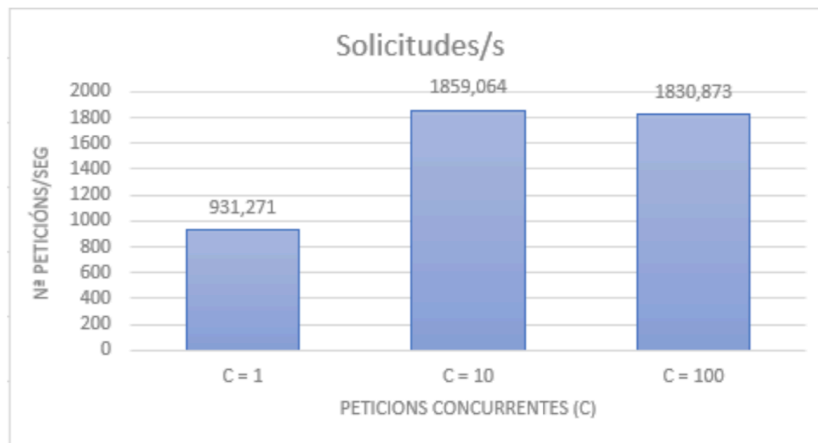
⁴ Referencia recomendada: https://www.tutorialspoint.com/apache_bench/apache_bench_quick_guide.htm

Parámetros: n = <valor de n>; c = <1 10 100>												
Execución	Tempo total	solicitudes/s	velocidade de transferencia	Tempo de conexión total			Porcentaxe de solicitudes					
				min	med	max	50	75	90	95	98	100
1												
...												
10												
Media												
Desviación												

- Unha gráfica cos valores medios de cada métrica nas 3 táboas (5 gráficas en total, 1 por métrica). Explica os resultados.

Exemplos de gráficas

Solicitudes por segundo (valores medios)



Porcentaxe de solicitudes (valores medios)

