VIRTUALIZACIÓN Y SISTEMAS OPERATIVOS AVANZADOS

GUÍA 3. SPREAD TOOLKIT Y BROADCAST TOLERANTE A FALLOS

OBJETIVO

- Implementar una aplicación distribuida que permita asociar nodos a grupos en los cuales se distribuya información.
- Familiarizarse con el funcionamiento de Spread Toolkit.
- Estudiar como Spread Toolkit proporciona un servicio resistente/tolerante a fallos.

PREPARACIÓN CONTENEDORES

Para este trabajo práctico, se continuará utilizando la máquina virtual, con el sistema Ubuntu 22.04 (y que hacía las veces de sistema host), utilizada en los trabajos anteriores.

En este caso, se utilizarán tres contenedores LXC nuevos y separados dentro de la máquina virtual. Cada uno representará un nodo participante en una comunicación distribuida.

Para la creación y configuración de los nodos, en este caso se cuenta con un script creado para automatizar las siguientes tareas:

- Creación de contenedores.
- Creación y configuración del usuario a usar dentro de cada uno.
- Configuración de acceso por SSH.
- Establecimiento de variables de entorno para el uso sencillo y correcto de la librería Spread Toolkit.
- Creaciones de links para archivos de la librería.
- Traspaso de archivos desde la maquina host tanto de archivos de Spread Toolkit, así como de los ejemplos a utilizar, a cada uno de los contendores.

El script en cuestión se llama initContainers.sh y se puede encontrar en la carpeta TP3 dentro del archivo TP3.zip entregado para llevar a cabo la clase de laboratorio. El script crea de a un contenedor a la vez, para lo cual se debe pasar como parámetro el nombre del contenedor a crear. Para ejecutar, hacer lo siguiente:

```
chmod 777 initContainers.sh

./initContainers.sh <nombre-contenedor>
```

NOTA: Asegurarse que, junto al script, se encuentre la carpeta SPREAD (también ubicada dentro de TP3/).

INICIO Y USO DE CONTENEDORES

Para iniciar los contenedores LXC, usar el siguiente comando (si es que no están iniciados):

```
lxc start ubuntuC-node1
```

lxc start ubuntuC-node2

lxc start ubuntuC-node3

Si no es posible (o no se quiere) por algún motivo conectar por SSH a los contenedores, se puede ejecutar lo siguiente dependiendo el contenedor al cual se quiere conectar:

lxc exec ubuntuCClient -- su --login ubuntuadm

lxc exec ubuntuCServer -- su --login ubuntuadm

Una vez dentro de los contenedores, se podrá encontrar la ubicación /home/ubuntuadm/SPREAD. Esta carpeta no debe ser cambiada de lugar.

USO DE SPREAD TOOLKIT

Introducción

Spread Toolkit consta de dos partes, un demonio/daemon ejecutable que se ejecuta en cada maquina y una biblioteca que proporciona una interfaz de programación para escribir aplicaciones de “multicast” en grupo y que está vinculada a la aplicación.

El demonio, llamado "spread", debe ejecutarse como un usuario sin privilegios y leer su archivo de configuración, que puede estar en el mismo directorio que el ejecutable.

Cada demonio se puede iniciar de forma independiente y, si no encuentra ningún otro miembro (según lo que indica la configuración usada) se ejecuta como un grupo de máquinas de 1 máquina. Cuando se inicien otros demonios, se unirán a este grupo de máquinas. Las máquinas que ejecutan un demonio con un archivo de configuración común forman un “grupo de máquinas”.

Los demonios manejan mensajes de multicast entre ellos, brindando garantías de orden y entrega, detectando y manejando fallas de nodos o enlaces, y administrando todas las aplicaciones que están conectadas a cada demonio.

Spread Toolkit ofrece librerías para varios lenguajes como C, C++, Java y Python, y para diferentes arquitecturas. Para el desarrollo de este trabajo se utilizará C en Linux.

Instalación y configuración

A través del script mencionado previamente (initContainers.sh), ya se hizo disponible Spread Toolkit en los contenedores que serán utilizados. Todos los archivos relacionados al mismo se encuentran en la ubicación /home/ubuntuadm/SPREAD de cada contenedor.

Dentro de la carpeta SPREAD hay un archivo de configuración de ejemplo llamado spread.conf. Su contenido es el siguiente:

```
Spread_Segment 10.62.245.255 {
        ubuntuC-node1 10.62.245.177
        ubuntuC-node2 10.62.245.197
        ubuntuC-node3 10.62.245.219
}
```

En este archivo se encuentra información del segmento spread en el cual se ubicarán todos los nodos, que a su vez puede formar grupos entre sí. Se indica tanto la IP de cada uno de los nodos que forman parte del sistema distribuido, como la dirección de broadcast de la red que contiene a los nodos. Si no se especifica puerto en la configuración, se utiliza por defecto el 4803. Este archivo de configuración debe estar replicado en todos los nodos.
Para replicar la configuración a los nodos, se adjunta un script llamado updateSpreadConf.sh que copia el archivo spread.conf a cada contenedor (a la carpeta /home/ubuntuadm/SPREAD/). El archivo spread.conf que se va a copiar, debe encontrarse junto al script, y para ejecutar este, se debe hacer lo siguiente:

```
./updateSpreadConf.sh <nombre-contenedor>
```

Iniciar demonio Spread

Para iniciar el demonio ejecutar lo siguiente desde la carpeta /home/ubuntuadm/SPREAD:

```
./spread
```

Las opciones que pueden usarse son:

[-l y/n] : especifica si mostrar o no el log. Por default no se muestra.
[-n <proc name>] : especifica el nombre de identificación del proceso.
[-c <file name>] : toma la ruta de configuración de spread. Por default busca ./spread.conf. Si no encuentra un archivo de configuración lo busca en /etc/spread.conf.

Dentro del directorio donde se encuentra el archivo de configuraci´on, simplemente se puede correr:

```
./spread
```

API C

Para poder utilizar la API se debería incluir la librería sp.h:

```
#include <sp.h>
```

Funciones Connect y disconnect

Estas funciones sirven para la conexión con el demonio de Spread.

```
int SP connect(const char * spread_name, const char * private_name, int priority, int
group_membership, mailbox * mbox, char * private_group);

int SP disconnect(mailbox mbox);
```

Funciones Join y Leave

Utilizadas para unirse o abandonar un grupo.

```
int SP join(mailbox mbox , const char * group);

int SP leave(mailbox mbox, const char * group);
```

Multicast y read

Utilizadas para envío y recepción de mensajes dentro de un grupo.

```
int SP_multicast(mailbox mbox, service service_type, const char * group, int16 mess_type, int
mess_len, const char * mess);
```

*int SP_receive(mailbox mbox, service * service_type, char sender[MAX_GROUP_NAME], int max_groups, int * num_groups, char groups[][MAX_GROUP_NAME], int16 * mess_type, int * endian_mismatch, int max mess_len, char * mess);*

El campo service_type define el servicio de mensaje requerido:

UNRELIABLE_MESS
RELIABLE_MESS
FIFO_MESS
CAUSAL_MESS
AGREED_MESS
SAFE_MESS

Más funciones

Para más información sobre las funciones presentadas y sobre funciones adicionales, así como también mayor detalle sobre los parámetros, tipos de mensajes y su significado, dirigirse a http://www.spread.org/docs/spread_docs_4/docs/c_api.html

COMPILACIÓN DE ARCHIVOS

Para compilar los archivos fuente que utilicen Spread Toolkit, se deberá primero instalar gcc en los contenedores. Para hacerlo, ejecutar:

```
sudo apt-get install gcc
```

Una vez hecho esto, se puede compilar los archivos dentro de los contendores simplemente ejecutando:

```
gcc -o <nombre-salida> <archivo.c> -lspread
```

La opción -lspread es necesaria para que se utilice la librería.

ACTIVIDADES

1. Crear un archivo spread.conf valido teniendo en cuenta las IPs asignadas a los contenedores. Ver esto con el comando lxc list.
2. Iniciar el demonio spread en todos los nodos.
3. Ejecutar el programa spuser dentro de la carpeta. Con este programa será posible hacer pruebas de las funcionalidades que provee spread (manejo de grupos, comunicación entre miembros de un grupo, algunos tipos de mensajes, etc.).
4. Analizar el contenido del programa simple_user.c de la carpeta examples. Luego, compilar y correr el mismo.
5. Ocasionar eventos dentro del grupo de nodos, tales como la desconexión de la red de un nodo, la caída de un proceso miembro y la caída de spread o la caída completa del nodo. Una vez que sucedan estos eventos, analizar que mensajes son producidos y que representan en Spread Toolkit.

MAS INFORMACIÓN SOBRE SPREAD TOOLKIT

http://www.spread.org

http://www.spread.org/docs/guide/users_guide.pdf

http://www.spread.org/docs/spread_docs_4/docs/c_api.html