

# Virtualización y Sistemas Operativos Avanzados

## Trabajo Práctico N°3

Año 2019

## SPREAD TOOLKIT

### Formato de entrega

Se deberá enviar un email a [constanza.iquaglia@gmail.com](mailto:constanza.iquaglia@gmail.com) con el asunto “VSOA TP3” y un archivo adjunto .tar.gz que contenga:

- Documentación:
  - Nombre, apellido y correo electrónico de cada integrante.
  - Respuestas a las consignas planteadas. Incluir el código fuente con las aclaraciones que se consideren necesarias y las instrucciones para compilar y ejecutar las aplicaciones.
- Código fuente: Entregar dentro de una carpeta src el código fuente desarrollado.
- Makefile: Archivo de reglas para *make* que incluya los comandos necesarios para poder correr *make* y *make clean*. Los objetos se deberán crear en la carpeta obj.

### Consigna

Implementar una aplicación distribuida que permita conectar e intercambiar mensajes entre diferentes nodos y obtener información sobre el rendimiento del sistema.

La aplicación deberá desarrollarse utilizando Spread Toolkit. Esta herramienta permite que los nodos del sistema se unan a un grupo e intercambien mensajes.

En primer lugar, se deberá definir un número máximo de nodos MAX\_NODES. Al iniciar, los diferentes nodos se irán uniendo un único grupo. Cuando haya MAX\_NODES nodos en el grupo, éstos deberán comenzar a intercambiar una cantidad definida de mensajes entre ellos.

El proceso de envío y recepción de mensajes es el siguiente:

1. Cada nodo envía un mensaje de MSG\_LEN bytes por multicast.
2. Luego de enviar el mensaje, cada nodo espera a recibir mensajes del resto, es decir MAX\_NODES mensajes.
3. Se repiten pasos 1 y 2 hasta enviar MSG\_COUNT mensajes.

El proceso termina cuando todos hayan enviado y recibido todos los mensajes.

Además cada nodo deberá medir el tiempo desde que comienza hasta que termina el intercambio de mensajes y el throughput del sistema (cantidad de bytes enviados en total/tiempo).

Se deberá ir variando la configuración del sistema para observar su comportamiento. El informe final deberá presentar los resultados de las siguientes pruebas:

1. Medir el throughput del sistema para diferentes tamaños de mensajes (manteniendo la cantidad de nodos y tamaño de mensajes).
2. Evaluar el tiempo total de intercambio de mensajes para diferentes cantidades de nodos, dejando los demás parámetros fijos.
3. Sin variar los parámetros MSG\_LEN, MSG\_COUNT y MAX\_NODES, comparar el comportamiento del sistema utilizando orden causal y orden total.

Además presentar los datos que considere pertinentes de cada nodo (características físicas, OS utilizado, etc) y las conclusiones obtenidas de las mediciones realizadas.

### **Formato de la aplicación**

`spread_bench <NODE_ID> <MAX_NODES> <MSG_COUNT> <MSG_LEN>`

- **NODE\_ID:** Identificador del nodo.
- **MAX\_NODES:** Cantidad máxima de nodos.
- **MSG\_COUNT:** Cantidad de mensajes que los nodos enviarán por multicast, cuando el grupo llegue a MAX\_NODES integrantes.
- **MSG\_LEN:** Longitud en bytes del mensaje a enviar.

## Guía práctica Spread Toolkit

### Instalación

1. Descargar el código fuente de Spread Toolkit desde la página oficial <http://www.spread.org/download.html>.
2. Descomprimir el .tar.gz que se descarga.
3. Dentro de la carpeta descomprimida correr:  

```
$ ./configure  
$ make  
$ sudo make install
```

### Configuración

Dentro de la carpeta docs de spread hay un archivo de configuración de ejemplo (sample.spread.conf). Copiarlo dentro de la carpeta raíz de spread.

```
$ mv docs/sample.spread.conf spread.conf
```

Editar el archivo de configuración para agregar las IPs de los nodos que formen parte del sistema distribuido.

```
Spread_Segment 192.168.1.255:3333 {  
    machine1 192.168.1.20  
    machine2 192.168.1.21  
}
```

La IP de Spread\_Segment (en el ejemplo 192.168.1.255) es la dirección de broadcast de la red que contiene a los nodos. El puerto (3333) es el puerto en el que se levantará el servicio de Spread. Si no se especifica puerto utiliza el 4803.

El archivo de configuración debe replicarse en todos los nodos.

### Iniciar el demonio de Spread

```
$ spread [-l y/n] [-n proc.name] [-c config.file]
```

- -l especifica si mostrar o no el log. Por default no se muestra.
- -n especifica el nombre de identificación del proceso.
- -c 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 configuración, simplemente se puede correr:

```
$ spread
```

Para probar si el servicio está funcionando correctamente, se puede utilizar el programa spuser dentro de la carpeta examples. Spuser inicia una interfaz de usuario que permite utilizar las funciones de grupo y multicast por línea de comando.

```
$ ./spuser
```

## Funciones API C

Spread Toolkit provee varias APIs para diferentes lenguajes: C, C++, Java y Python. Para el desarrollo de este trabajo se utilizará C.

Para poder utilizar la API se deberá incluir la librería sp.h.

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

## 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);
```

## 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. El campo service\_type define el servicio de mensaje requerido:

- UNRELIABLE\_MESS
- RELIABLE\_MESS
- FIFO\_MESS
- CAUSAL\_MESS
- AGREED\_MESS
- SAFE\_MESS

```
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);
```

## Compilación en C

Para compilar el programa se le debe pasar la librería de Spread al compilador:

```
gcc spread_bench.c -o spread_bench -lsread
```

## **Bibliografía**

- <http://www.spread.org>
- Documentación: <http://www.spread.org/docspread.html>
- Guía de usuario de Spread: [http://www.spread.org/docs/guide/users\\_guide.pdf](http://www.spread.org/docs/guide/users_guide.pdf)
- Archivos de configuración y readme provistos por Spread.