Spread Introduction

Spread consists of two parts, an executable daemon which is executed on each machine which is part of the heavyweight 'machine' group, and a library which provides a programming interface for writing group multicast applications and which is linked into the application.

The daemon, called "spread", should be run as a non-priveledged user (we created a 'spread' user) and only needs permissions to create a socket in /tmp and read its config file which should be in the same directory as the executable. By default the daemon binds to and runs off the non-priveledged port 4803 unless the config file indicates otherwise.

Each daemon can be started independently and if it does not find any other members of its defined configuration currently active it runs as a machine group of 1 machine. When other daemons are started they will join this machine group. The machines which are running a daemon with a common config file form a 'machine group' (in contrast to a 'process group').

The daemons handle multicasting mesages between each other,providing ordering and delivery guarantees, detecting and handling failures of nodes or links, and managing all applications which are connected to each daem**on.**

1) Installation:

Download the spread toolkit from http://www.spread.org/download.html

(I have tested against many versions and only 4.0.0 seems to be working)

Or copy from Assignment2_Example

Dependencies:

1. c compiler (“sudo apt-get install gcc” – to install gcc in ubuntu)
2. **hard lesson:** (gcc 8 is working. but anything above gcc 9 is not working. so if newer linux os is used then the compiler should be downgraded)

steps:

1. extract the compressed file and go to the extracted directory
2. Run "./configure"
3. Run “make”
4. Run “make install” (with root permissions)
5. The target execution file is <extracted_directory>/daemon/spread

2) spread config file:

the spread daemon takes the configuration file.

The config file (spread.conf):

```
Spread_Segment  <broadcast address>: <port> (eg:- 127.0.0.255,5000)
{
<servername>            <machine address>  (eg:- spread_server 127.0.0.1)
}
```

3) running spread daemon:

```
spread –l y -n <servername> –c spread.conf
```

4) Setup spread server in IFI machine (Optional):

a) Starting server:

Copy the already-built spreadtoolkit to IFI machine.

To start the server, the config file should be provided. The machine address should be corresponding to the ip address of the machine. The broadcast address should be the corresponding broadcast address. (ifconfig can be used to obtain both the addresses).

accessing IFI machine:

```
ssh -YC <username>@login.ifi.uio.no
```

For example, spread config file:

129.240.65.60 is the ip address of the machine and 129.240.65.255 is the corresponding broadcast address. Use unique host name since different students can be allocated same ifi machine while accessing it through ssh. Try different custom ports such as 8000 if 4803 doesn't work.

```
Spread_Segment  129.240.65.255:4803
{
spread_server 129.240.65.60
}
```

then run the spread daemon from the IFI server machine by using the command specified in step 3

b) accessing spread server running at ifi:

If the ifi machine is used to access the server, then the ip address and port (129.240.65.60):(4803) can be used in java connect function.

If the outside machine is used then ssh tunnel should be setup and the spread server can be accessed using the tunneled ip address and port.

setting up ssh tunnel:

```
ssh -L <local ip>:<local port>:<remote ip>:<remote port> <username>:<remote machine hostname>
```

eg:

```
ssh -L 127.0.0.1:8011:129.240.65.60:4803 truongl@login.ifi.uio.no
```

5) Java library:

1. go to spread package root directory -> java
2. compile the source files. easiest option is to build the jar file and then use the jar file in the project. use the command 'ant' to build jar file. Or you can copy from Assignment2_Example. Note: you need some ways to implement it as a dependency

```
ant
```

the alternative option is to copy the spread class files into the project source directory.

6) Spread library:

It consists of 12 classes. The imporant ones are:

1. SpreadConnection – represents a connection to the spread daemon
2. SpreadGroup – represents the spread group
3. SpreadMessage – represents message that is either sent or received

7) spread connection:

```
SpreadConnection connection = new SpreadConnection();

connection.connect(<ip address>, <port number>, <connection name>, <priority>,

<group membership>);
```

parameters:

- **ip address –** ip address of the host that runs the spread server
- **port number –** port number of the spread server
- **connection name -** unique connection name. unique per client
- **priority** – boolean value to determine whether it's a priority connection or not. it doesn't have any effect.
- **group membership** - denotes whether the group membership messages are received or not. set it to true.

8) spread group:

```
SpreadGroup group = new SpreadGroup();

group.join(connection, <group name>);
```

parameters

- group name – name of the group

9) spread message:

```
SpreadMessage message = new SpreadMessage();
message.addGroup(group);
message.setFifo();
 message.setObject(<message data>);
connection.multicast(message);
```

10) Listener:

group members can receive message using the listener. Spread provides two types of interfaces: BasicListener and AdvancedListener. To use the listener, the interface should be implemented and the corresponding class object should be added with the connection.

```
Listener listener = new Listener();
connection.add(listener);
```

References:

1. http://www.spread.org/docs/guide/users_guide.pdf
2. http://www.spread.org/docs/javadocs/java.html