

Prácticas de Laboratorio

Desarrollo de un foro de mensajes

Fase 2: Backend distribuido

Fase 2.2: grupo de servidores

Servicios y Aplicaciones Distribuidas

Introducción

En la etapa anterior, hicimos que el servidor de datos "dmserver.js" publicara sus modificaciones. Por su parte el servidor web "forum.js" se subscribía a esas publicaciones.

En esta etapa, continuamos con el modelo publicador/suscriptor para lograr que diferentes servidores de datos propaguen sus cambios a los demás servidores. Para ello, será necesario es hacer que los servidores de datos "dmserver.js" se subscriban a las publicaciones que efectúan los demás servidores.

Llegados a este punto, contaremos con un conjunto de servidores de datos, un conjunto de servidores web y un conjunto de navegadores. Los cambios que inicien los navegadores web, deberán propagarse al conjunto de servidores de datos, y éstos, a su vez, propagarán los cambios a sus respectivos servidores web, que, a su vez, propagarán los cambios a los navegadores que se encuentren conectados.

Instalación del software

Para esta etapa no es necesarios software adicional.

Trabajo a realizar

1. Añadir un último argumento a "dmserver.js" que contendrá una lista separada por comas con las url de los diferentes servidores de datos. Si el argumento no existe, significará que no hay más servidores de datos. Si este último argumento tiene una única url, significará que tenemos dos servidores de datos, y si tiene más de una url, estarán separadas por comas. p.ej.:

`'tcp://127.0.0.1:20000,tcp://127.0.0.1:20001'`

2. El servidor de datos "dmserver" se suscribirá a todos los demás servidores de datos, utilizando por ejemplo la etiqueta "checkpoint" para suscribirse a estos cambios. `sock.subscribe ("checkpoint");`
3. Los servidores de datos, cuando reciban cambios que provengan de su socket "rep", los propagarán a los demás servidores usando la etiqueta "checkpoint": `pub.send (["checkpoint", data])` además de continuar enviando los cambios a sus servidores web, por ejemplo usando la etiqueta "webserver": `pub.send(["webserver", data])`
4. Cuando los servidores de datos reciban un mensaje de tipo "checkpoint", lo publicarán a sus servidores web mediante un mensaje de tipo "webserver", pero lógicamente no deberán retransmitirlo de nuevo a los demás servidores de datos.

Para esta sesión de laboratorio únicamente se pide realizar las propagaciones de datos, para los mensajes publicados a los temas del foro.

Recursos y bibliografía

- <http://zguide.zeromq.org/page:all>
- Apuntes del Seminario 6 : zeroMQ