

Prácticas de Laboratorio

Desarrollo de un foro de mensajes

Fase 2: Backend distribuido

Fase 2.3: consistencia

Servicios y Aplicaciones
Distribuidas

Introducción

En las etapas anteriores hemos construido un grupo de servidores de datos replicados, que se mantienen mutuamente actualizados mediante zeroMQ. En esta etapa realizaremos varias pruebas de ejecución del sistema, y trataremos de observar los modelos de consistencia que éste garantiza.

Para realizar las pruebas inyectaremos pequeños retrasos en algún nodo, simulando que su ejecución es lenta, en algunos casos. Para ello contaremos con la función `retardo(n)`, que usaremos en algunas partes del código para retrasar la ejecución cierto número de milisegundos:

```
function retardo(n) {  
  time = new Date().getTime();  
  time2 = time + n;  
  while (time < time2) {  
    time = new Date().getTime();  
  }  
}
```

Instalación del software

Para esta etapa no es necesario software adicional.

Trabajo a realizar

El trabajo lo dividimos en dos partes. Prueba convencional del software y pruebas de consistencia.

Prueba convencional del software

Probar la corrección del software. Para ello lanzaremos 3 servidores de datos "dmserver". Estarán conectados entre ellos mediante zeroMQ. Lanzaremos a su vez 3 servidores web "forum". Cada uno de ellos conectado a un servidor de datos diferente. Lanzaremos 3 navegadores web. Cada uno de ellos conectado a un servidor web diferente. Posteando mensajes en cada navegador debemos observar que los mensajes se actualizan en los diferentes navegadores.

Pruebas de consistencia

Utilizando terminales adicionales, lanzaremos clientes "dmclient", contra cada uno de los servidores de datos.

Prueba 1: consistencia FIFO.

Posteamos dos mensajes de forma consecutiva sobre el mismo "dmserver" y debemos observar cómo los dos mensajes llegan en el mismo orden a los navegadores web conectados.

Repetimos el proceso añadiendo un "retardo" en el "dmserver", justo antes de que éste publique sus cambios al servidor web.

Prueba 2: consistencia causal.

Supongamos los 3 "dmserver" llamados S1, S2 y S3. Posteamos un mensaje sobre S1. A continuación desde otro terminal, consultamos la lista de mensajes en el servidor S2, y acto seguido posteamos un nuevo mensaje sobre S2. Observamos qué mensajes aparecen en los navegadores.

Diseñamos una prueba para forzar el fallo en la consistencia causal, mediante "retardos" hemos de lograr observar que el navegador conectado a S3, muestre el mensaje enviado a S2, antes que el mensaje inicial.

Prueba 3: consistencia secuencial.

Diseñar una prueba que muestre si nuestro sistema cumple o no la consistencia secuencial.

Recursos y bibliografía

- Apuntes del Seminario 8 y 9: consistencia