Universidad Rey Juan Carlos

TITULACIÓN EN INGENIERÍA DEL SOFTWARE

Curso Académico 2017/2018

Trabajo Fin de Grado

Comparativa de tecnologías de servidor para servicios basados en websocket

Autor : Michel Maes Bermejo

Tutor : Micael Gallego Carrillo

Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.

Índice general

Capítulo 1

Introducción y motivación

Hoy en día, un desarrollador de software tiene múltiples herramientas (entre lenguajes y librerías) para abordar cualquier proyecto que tenga entre manos.

Es una práctica común usar una tecnología concreta sobre la que sentimos predilección o las que creemos que pueden resolver mejor nuestro problema. En ocasiones, nos equivocamos en nuestra elección y descartamos opciones mucho más efectivas.

Este problema de desinformación puede abordarse mediante el estudio de las distintas tecnologías que proponen una solución al mismo, pero dado que el ámbito del desarrollo software es muy amplio, vamos a centrarnos en las tecnologías de servidor para servicios basados en WebSockets.

Estas tecnologías proporcionan una comunicación en tiempo real con clientes muy diversos (aplicaciones móviles, navegadores, otro servidores). Un ejemplo actual son los servicios de mensajería instantánea cómo WhatsApp o Telegram, cuyo crecimiento de usuarios se ha disparado en los últimos años. Hoy en día este tipo de aplicaciones tienen un impacto drástico en la vida diaria, siendo casi una herramienta imprescindible, por lo que prevenir una caída de servicio ante un alto número de clientes es fundamental.

La motivación de este proyecto surge de la necesidad de comprender mejor estas tecnologías y proporcionar argumentos sólidos que justifiquen el uso de una u otra, dependiendo de las necesidades de nuestro proyecto y de los recursos de los que dispongamos.

Para ello, tomaremos como punto de partida las tecnologías reactivas, que siguiendo el Manifiesto Reactivo[1] cuentan entre sus características:

- Tiempos de respuestas rápidos
- Tolerantes a fallos
- Adaptación a variaciones en la carga de trabajo
- Uso de mensajes asíncronos para la comunicación (no bloqueantes)

Para este proyecto, nos centraremos en Java, un lenguaje consolidado que cuenta con librerias y frameworks que nos ayudarán a abordar esta comparativa.

[1] `http://www.reactivemanifesto.org/`

Capítulo 2

Objetivos

El objetivo principal de este proyecto será realizar una comparativa entre distintas tecnologías que den solución a la comunicación en tiempo real mediante el uso de WebSockets. Dicha comparativa se realizará en base al rendimiento y el consumo de recursos de cada una de las tecnologías comparadas, ante diferentes niveles de carga y haciendo uso de un número variado de servidores.

Con este fin, se implementará un servidor de mensajería instantánea (que a partir de ahora denominaremos simplemente Chat) para cada tecnología y un cliente que se conectará a ese servidor simulando varios usuarios enviando mensajes que podrá medir el tiempo que tarda un mensaje desde que se envía hasta que se recibe.

Otro objetivo relevante del proyecto será su extensibilidad, de forma que cualquier desarrollador pueda implementar su aplicación de chat, sumarla a la comparativa y así contribuir al proyecto.

El proyecto base corresponde al realizado por el mismo autor de el proyecto que nos ocupa, en el que se realizó una comparativa entre las aplicaciones de Akka, Vert.x, SpringBoot y NodeJS, las cuales se compararón haciendo uso de una sola máquina. Este proyecto pretende ser una continuación y expansión del anterior, concretamente:

- Distribuir cada aplicación para que pueda ser lanzada en varias máquinas que formen un clúster.
- Actualizar las librerias a su última versión a fin de contar con las herramientas más recientes.

- Mejorar el cliente existente para que sea capaz de recoger métricas de una aplicación distribuida.

Las tecnologías que compararemos en este proyecto serán:

- Akka
- Vert.x
- SpringBoot + RabbitMQ

Capítulo 3

Tecnologías, Herramientas y Metodologías

3.1. Tecnologías

3.1.1. Websockets

RFC 6455[1] define WebSocket como un protocolo que proporciona un canal de comunicación bidireccional y full-dúplex sobre un único socket TCP. Aunque inicialmente estaba pensado para cualquier tipo de comunicaciones entre el navegador y el servidor web, puede usarse también para aplicaciones cliente/servidor.

Por otro lado, W3C se encarga de normalizar la API[2] de WebSocket. Define una interfaz para el navegador compuesta por 4 métodos que corresponden a manejadores o gestores (*handlers*) para cada evento.

[1] `https://tools.ietf.org/html/rfc6455`

[2] `https://www.w3.org/TR/2011/WD-websockets-20110929`

Podemos ver un ejemplo de estos manejadores en el código mostrado a continuación (Javascript en el navegador).

```
var socket = new WebSocket("ws://example.com:9000/chat");
// Send new text
socket.send("Some text");
socket.onmessage = function(event) {
    var data = JSON.parse(event.data);
    // Use data
});
socket.onopen  = function(e){ console.log("WS Opened")};
socket.onclose = function(e){ console.log("WS Closed")};
socket.onerror = function(e){ console.log(e)};
```

3.2. Herramientas

3.3. Metodologías

Capítulo 4

Descripción informática

4.1. Requisitos

4.2. Diseño e Implementación

4.3. Pruebas

Capítulo 5

Estudio comparativo

5.1. Latencia

5.2. Uso de CPU

5.3. Uso de memoria

5.4. Desarrollo

5.5. Conclusiones generales de la comparativa

Capítulo 6

Conclusiones del proyecto y trabajos futuros

Mis conclusiones

Apéndice A

Manual de usuario