**SSDD**

Práctica 1

Pablo Orduna Lagarma y Daniel Naval Alcalá

24 de septiembre de 2018

RESUMEN

En esta memoria se van a describir las soluciones implementadas a los escenarios propuestos; para ello se hará uso de un diagrama de secuencias acompañado de un texto explicativo para complementar la información de cada solución.

Se demostrará de forma breve y esquemática que las soluciones planteadas tienen sentido y pueden ser aplicadas a un entorno real.

A continuación, se comentarán las conclusiones sobre el desarrollo de la práctica y finalmente se describirán las fuentes bibliográficas consultadas para el desarrollo de esta.

INTRODUCCIÓN

A lo largo del desarrollo de esta práctica se han implementado varios tipos de arquitecturas de sistemas distribuidos para resolver el problema matemático del cálculo de números perfectos en un rango determinado de manera distribuida para conseguir una calidad de servicio adecuada en diferentes escenarios. Para ello se proponen diferentes tipos de clientes y situaciones que debería afrontar el servidor manteniendo en todo momento respuestas a todas las peticiones que solicita el cliente en el menor tiempo posible.

Estas situaciones las afrontaremos con las arquitecturas vistas en clase.

IMPLEMENTACIÓN

ESCENARIO 1

El cliente realiza 1 petición cada 2 segundos

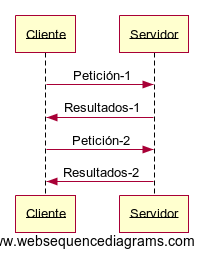


Ilustración 1: Diagrama de secuencias del primer escenario

El servidor utilizado en este escenario se encuentra ya proporcionado como material de la práctica, se trata de un modelo cliente-servidor que actúa de forma secuencial y no permite la paralelización de trabajos.

ESCENARIO 2

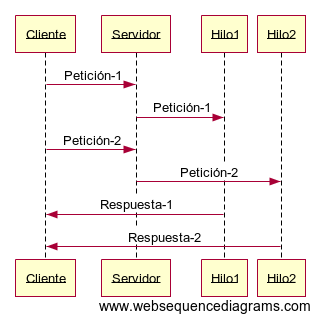
El cliente realiza más de 1 petición cada 2 segundos, dependiendo del sistema

Ilustración 2: Diagrama de secuencias del segundo escenario

En este caso el servidor se ha modificado ligeramente a partir de la versión del escenario para permitir la paralelización de trabajos, de esta manera es posible tratar cada petición en un hilo independiente del servidor posibilitando responder a una mayor cantidad de peticiones.

ESCENARIO 3

El cliente realiza más de 1 petición cada 2 segundos, dependiendo del sistema

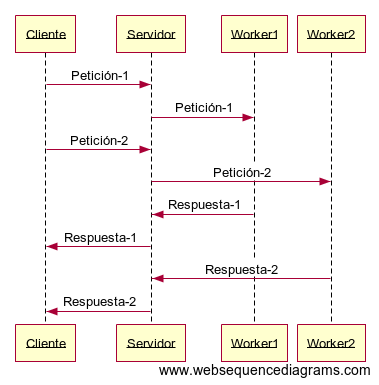


Ilustración 3: Diagrama de secuencias del tercer escenario

Aunque para este escenario funciona correctamente el servidor planteado para el Escenario 2 debido a que una máquina lo suficientemente potente puede hacerse cargo de todas las peticiones con facilidad, se ha visto oportuno implementar un modelo Máster-Worker ya que el numero de peticiones es mucho más elevada que en el escenario anterior.

La implementación se basa en el hecho de que el Máster puede recibir dos tipos de mensajes:

* Mensajes del cliente con peticiones:

Cuando esto ocurra, se enviará un mensaje a un Worker que se encuentre disponible para que realice la tarea solicitada.

* Mensajes de los Workers con resultados:

En esta ocasión se añadirá el WorkerX a la lista de Workers disponibles y se mandarán los resultados al cliente.

Siempre y cuando no haya Workers en la lista de disponibles, se recibirán únicamente mensajes de los Workers, de manera que siempre llegue una petición habrá al menos siempre un Worker disponible.

ESCENARIO 4

El cliente realiza más de 1 petición cada 2 segundos, dependiendo del sistema y además los recursos computacionales tienen cierto grado de indeterminismoImagen que contiene mapa, texto

Descripción generada con confianza muy alta

*Ilustración 4:Diagrama de secuencias del cuarto y último escenario*

Este último escenario cuenta con un servidor que es una ligera modificación del anterior. Esta modificación se basa en que por cada petición se envía la tarea a tres Workers que concurrirán para conseguir realizar el trabajo propuesto, de manera que se devuelve al cliente el resultado del Worker que finalice la tarea el primero ignorando los resultados del resto de Workers que procesaban esa petición.

VALIDACIÓN Y CONCLUSIÓN

La solución planteada para todos los escenarios propuestos se ha comprobado empíricamente ejecutando los programas desarrollados en diferentes máquinas dependiendo del escenario.

El escenario 1 y 2 han funcionado correctamente sin violación de QoS; sin embargo, en la solución para los escenarios 3 y 4 se obtienen tiempos lo suficientemente elevados para que no sea posible garantizar la QoS. Por ello concluimos que en una infraestructura con un gran número de Workers divididos en diferentes máquinas funcionarían correctamente debido a la gran diferencia en la potencia de cómputo.

BIBLIOGRAFÍA

* <https://elixir-lang.org>
* <https://hexdocs.pm>
* <https://stackoverflow.com>
* [https://elixirforum.com](https://elixirforum.com/)