I - S - 2021 - RRF - Programación Paralela y Concurrente - 002

Proyecto01-Avance03: Aplicación distribuida (opcional)

En el avance 1 se construyó un *servidor web* concurrente que podía atender una cantidad arbitraria de clientes de forma simultánea, los cuales solicitan obtener sumas de Goldbach calculadas por una *aplicación web* serial, lo cual no es óptimo.

En el avance 2 se rediseñó la solución para hacer un uso más balanceado de las CPUs paralelizando la aplicación web. De esta forma los `HttpConnectionHandlers` eran hilos livianos de concurrencia de tareas que esperaban solicitudes pero no las atendían, sino que las trasladaban a la aplicación web, la cual contaba con hilos de paralelismo de datos (calculadoras) que se encargaban de encontrar las sumas de Goldbach. Además se usó una unidad de descomposición más fina que los sockets para mejorar el balance de carga entre los hilos de paralelismo de datos. Esta solución logra hacer un uso eficiente de los recursos de una máquina, pero desaprovecha otras que podrían estar disponibles en una organización o un clúster.

En el avance 3 el objetivo es distribuir la aplicación de Goldbach para que sea escalable, logre aprovechar varias máquinas que se tengan disponibles en una organización, y reduzca sustancialmente los tiempos de respuesta. De esta forma, si el servidor web recibe una carga sustancial de trabajo, puede responder en menor tiempo y disminuir la probabilidad de que los clientes se impacienten y se desconecten dejando trabajo residual en el

servidor.

En este avance se deben distribuir las calculadoras de Goldbach en la aplicación web. Es decir, una parte de la aplicación web corre junto con el servidor web en el mismo proceso. Las calculadoras de Goldbach corren en procesos separados, sean en la misma o distintas máquinas.

Las calculadoras deben iniciarse primero y correr como servidores TCP (véase la clase `TcpServer`) o si se prefiere, como otros servidores web (o más apropiadamente "servicios web").

El proceso con el servidor web lee de un archivo de configuración la ubicación de las calculadoras, indicadas como direcciones IP y puertos. El servidor recibe las solicitudes HTTP, las descompone, y distribuye el trabajo entre las calculadoras mediante un mapeo eficiente que balancee la carga.

El servidor web recibe de las calculadoras las sumas de Goldbach encontradas, por lo tanto ocurre también el proceso inverso: el proceso con el servidor web también debe tener un servidor TCP y las calculadoras ser clientes TCP (o su correspondientes web si se usan "servicios web").

Se debe:

1. [15%] Plantear un diseño concurrente de los requerimientos anteriores.
2. [30%] Implementar el paso de mensajes eficiente con las clases de la capa de red provista (`TcpServer`, `TcpClient`, `HttpServer`).
3. [20%] Balancear la carga entre las calculadoras (descomposición y mapeo).
4. [15%] Detener las calculadoras cuando se detiene el servidor web (con `Ctrl+C` o el comando `kill`).
5. [20%] Documentación y estructura de proyecto. No genera diagnósticos de Valgrind, Sanitizers, Linter, Doxygen.

El avance 3 es de carácter opcional con un peso de 25% del proyecto01, y su fecha límite de entrega es viernes 23 de julio sin posibilidad de extensión. Se entrega en una carpeta propia para el avance en el repositorio (ej. `Proy01a03` o `webserver_distr`). Todos los miembros del grupo deben demostrar un nivel similar de participación en el desarrollo de la solución.