

Sistemas Distribuidos

Monica Scheduler

Laura Victoria Riera Pérez
Marié del Valle Reyes

Cuarto año. Ciencias de la Computación.
Facultad de Matemática y Computación, Universidad de La Habana, Cuba

12 de julio de 2023

REPOSITORIO DEL PROYECTO

<https://github.com/computer-science-crows/monica-scheduler>

Implementación de una Agenda Electrónica como Sistema Distribuido con DHT Kademlia

I. INTRODUCCIÓN

El tiempo es un recurso invaluable y su gestión eficiente es esencial para la productividad y el bienestar personal. Una estrategia comúnmente utilizada para la gestión del tiempo es el uso de una agenda. Sin embargo, en muchas ocasiones, es necesario coordinar dicha agenda con otras personas para llevar a cabo actividades conjuntas. Este proceso implica la identificación de horarios compartidos y la detección de intervalos de tiempo libres. Además, estas planificaciones pueden verse alteradas por eventos imprevistos que requieren asistencia, lo que conlleva la necesidad de modificar la agenda nuevamente.

Para abordar estos desafíos, este proyecto propone la creación de una agenda electrónica distribuida como herramienta de gestión del tiempo para eventos personales o grupales. El sistema se diseñó e implementó como un sistema distribuido, utilizando la tecnología de contenedores Docker para simular múltiples servidores y la Tabla Hash Distribuida (DHT) de Kademlia para la gestión de datos.

II. ARQUITECTURA DEL SISTEMA

El sistema se compone de dos partes fundamentales: la lógica del negocio y el uso del DHT Kademlia.

I. Lógica del Negocio

Este proyecto consiste en crear una agenda distribuida como herramienta de gestión del tiempo para eventos personales o grupales. Los requisitos clave para este sistema son:

- **Arquitectura Distribuida:** El sistema debe ser diseñado e implementado como un sistema distribuido. Esto significa que el sistema debe ser capaz de funcionar en múltiples máquinas mientras se presenta a los usuarios como un sistema coherente único.

- **Mecanismo de Autenticación e Identificación:** El sistema debe ser capaz de autenticar e identificar a cada usuario. Esto asegura que solo los usuarios autorizados puedan acceder al sistema y realizar ciertas acciones.
- **Formación Dinámica de Grupos:** El sistema debe permitir la creación dinámica de grupos, ya sean jerárquicos o no jerárquicos. Esto significa que los usuarios deben poder crear y gestionar grupos dentro del sistema de manera flexible.
- **Citas Grupales:** El sistema debe admitir la creación de citas grupales. Si se utiliza un grupo jerárquico, una cita creada por un superior debe aparecer automáticamente en las agendas de todos los miembros del grupo. Para grupos no jerárquicos, todos los miembros deben aceptar la cita para que se confirme.
- **Actualizaciones Automáticas de la Agenda:** Cuando se crea, modifica o elimina una cita, los cambios deben reflejarse automáticamente en las agendas de los usuarios relevantes.
- **Identificación de Conflictos:** El sistema debe ser capaz de identificar conflictos en las agendas locales. Por ejemplo, si un usuario tiene programadas dos citas al mismo tiempo, el sistema debe señalar esto como un conflicto.

II. Uso del DHT Kademlia

El DHT Kademlia se utiliza para la gestión de datos en el sistema. Kademlia es un protocolo de red para sistemas distribuidos peer-to-peer que utiliza una tabla hash distribuida para el almacenamiento y recuperación de datos.

El proceso de ingreso de un nodo a la red Kademlia se realiza mediante un mecanismo de broadcast. Cuando un nuevo nodo se quiere unir a la red, envía un mensaje de broadcast a todos los nodos existentes. Si hay algún servidor escuchando este devolverá su ip, lo cual permitirá al nodo poder conectarse a la red. Los nodos existentes actualizan sus tablas de enrutamiento para incluir al nuevo nodo.

III. IMPLEMENTACIÓN CON DOCKER

Para simular distintos servidores en un sistema distribuido, utilizamos Docker. Cada contenedor Docker puede considerarse como un nodo en el sistema distribuido, capaz de ejecutar su propio sistema operativo, tener su propia configuración de red y ejecutar las aplicaciones como si estuviera en una máquina separada. Esto permite simular un sistema distribuido en una sola máquina física, lo que facilita el desarrollo y las pruebas del sistema.

IV. PRUEBAS DEL SISTEMA

Para la presentación y prueba del sistema, se desarrolló una interfaz de línea de comandos (CLI). Los usuarios pueden interactuar con el sistema a través de la CLI, lo que permite una interacción flexible y eficiente con el sistema.

V. CONCLUSIONES Y TRABAJO FUTURO

La implementación de una agenda electrónica como sistema distribuido presenta numerosos desafíos, pero también ofrece muchas ventajas. La capacidad de gestionar el tiempo de manera eficiente y coordinar actividades con otros usuarios es crucial en el mundo moderno. Con la arquitectura distribuida, el mecanismo de autenticación e identificación, la formación dinámica de

grupos, la gestión de citas grupales, las actualizaciones automáticas de la agenda, la identificación de conflictos y la visualización de la agenda, nuestro sistema proporciona una solución robusta y flexible para la gestión del tiempo.

REFERENCIAS