

Universidad EAFIT
ST0263: Tópicos Especiales en Telemática, 2024-2
Reto No 1

Título: Arquitectura P2P y Comunicación entre procesos mediante API REST, RPC y MOM

Realice el diseño e implementación de un sistema P2P donde cada nodo / proceso contiene uno o más microservicios que soportan un sistema de compartición de archivos distribuido y descentralizado.

La versión de este reto1 se requiere realizarla en un esquema de red P2P estructurada basada en Chord / DHT o similar o puede explorar otra alternativa de red no estructurada sin servidor con superpeer o pura.

Cada nodo, proceso o peer contiene módulos servidor (PServidor) definidos como varios microservicios y un módulo cliente (PCliente).

El punto de acceso al servicio debe ser cualquier otro peer de la red, es decir, si en un momento dado se tienen 4 peers (P1, P2, P3, P4), y el P2 quiere iniciar en el sistema, podría contactar al siguiente Peer de acuerdo con la implementación o tipo de red P2P implementada. Este sistema inicial P2P solo realizará el mantenimiento de la red P2P y localización de recursos, LA transferencia de archivos se realizará mediante servicios ECO o dummies send/receive.

Cada uno de los microservicios PServidor debe soportar concurrencia, es decir, permitir a más de un proceso remoto comunicarse simultáneamente. Para la comunicación RPC ud utilizará el middleware API REST, gRPC y MOM.

A nivel de lógica de negocio, implementará consultas acerca de los recursos (archivos) que tiene en cada nodo (peer) donde corre el microservicio. Se recomienda que a nivel de recursos sean los archivos que cada uno de los nodos tiene dado un directorio específico configurable al momento del Bootstrap del microservicio. Por ahora, solo se compartirá el índice o listado de los archivos que posee y su URI/URL, no se trata de transferencia ni de sincronización real de archivos, PERO el peer que tiene el recurso si debe implementar 2 servicios ECO o DUMMY para la descarga (download) y carga (upload) de archivos. Separar en diferentes microservicios las diferentes funcionalidades del peer.

Cada uno de los peer tendrá un archivo de configuración que leerá dinámicamente cuando suba el proceso (Bootstrap). En el archivo de configuración mínimo podría contener entre otros:

IP sobre la que hará listening (ej: 0.0.0.0)

Port sobre el que hará listening (depende del middleware)

Directorio sobre el que listará o buscará archivos.

URL de un peer/superpeer semilla de acuerdo al tipo de red p2p diseñada e implementada

Realice todas las adecuaciones o variantes que desee de acuerdo con sus intereses académicos o profesionales, o impleméntelo como dice el enunciado.

Entregables:

- El repositorio github individual de cada alumno con el formato: “username-st0263”
- El readme.md diligenciado con la plantilla de entregas. Ver:
 - <https://github.com/st0263eafit/st0263-242/blob/main/README-template.md>

Actividades:

1. Preguntas que tiene del enunciado (entendimiento del problema)
2. Defina la versión inicial de la arquitectura y tipo de red P2P, revísela con su compañeros y reciba retroalimentación del profesor.
3. Defina los servicios específicos que tendrá cada componente del sistema
4. Definir el mecanismo de localización basado en índice central o distribuido (ideal distribuido)
5. Defina las interacciones entre componentes, los tipos de comunicaciones y tipo de middleware específico que va a emplear (REST API, gRPC, MOM), debe emplear todos estos middlewares.
6. Defina un plan de desarrollo, desde victorias tempranas, hasta la finalización del proyecto.
7. Desarrollo y Pruebas en localhost o AWS
8. Despliegue los nodos como máquinas virtuales con docker en AWS Academy
9. Realice pruebas en AWS.
10. Realice la documentación
11. Entregue y sustente al profesor mediante un video creado por ud donde explique el proceso de diseño, desarrollo y ejecución (no más de 30 mins)

(Este enunciado se seguirá actualizando de acuerdo con aclaraciones y retroalimentación recibida, los cambios o adiciones serán resaltados)

Fecha de entrega:

31 agosto de 2024