

Programación Distribuida y Tiempo Real
Facultad de Informática
Universidad Nacional de La Plata

Práctica 3

1) Utilizando como base el programa ejemplo¹ de gRPC:

a.- Mostrar experimentos donde se produzcan errores de conectividad del lado del cliente y del lado del servidor. Si es necesario realice cambios mínimos para, por ejemplo, incluir `sleep()` o `exit()`, de forma tal que no se reciban comunicaciones o no haya receptor para las comunicaciones.

2) Describir y analizar los tipos de API que tiene gRPC. Desarrolle una conclusión acerca de cuál es la mejor opción para los siguientes escenarios:

- Un sistema de pub/sub
- Un sistema de archivos FTP
- Un sistema de chat

3) Analizar la transparencia de gRPC en cuanto al manejo de parámetros de los procedimientos remotos. Considerar lo que sucede en el caso de los valores de retorno. Puede aprovechar el ejemplo provistos.

4) Con la finalidad de contar con una versión muy restringida de un sistema de archivos remoto, en el cual se puedan llevar a cabo las operaciones enunciadas informalmente como

- leer: dado un nombre de archivo, una posición y una cantidad de bytes a leer, retorna 1) los bytes efectivamente leídos desde la posición pedida y la cantidad pedida en caso de ser posible, y 2) la cantidad de bytes que efectivamente se retornan leídos.
- escribir: dado un nombre de archivo, una cantidad de bytes determinada, y un buffer a partir del cual están los datos, se escriben los datos en el archivo dado. Si el archivo existe, los datos se agregan al final, si el archivo no existe,

¹ <https://github.com/pdytr/pdytr-grpc-demo>

se crea y se le escriben los datos. En todos los casos se retorna la cantidad de bytes escritos.

a.- Defina e implemente con gRPC un servidor. Documente todas las decisiones tomadas.

b.- Investigue si es posible que varias invocaciones remotas estén ejecutándose concurrentemente y si esto es apropiado o no para el servidor de archivos del ejercicio anterior. En caso de que no sea apropiado, analice si es posible proveer una solución (enunciar/describir una solución, no es necesario implementarla).

Nota: diseñe un experimento con el que se pueda demostrar fehacientemente que dos o más invocaciones remotas se ejecutan concurrentemente o no.

5) Timeouts en gRPC:

a.- Desarrollar un experimento que muestre el timeout definido para las llamadas gRPC y el promedio de tiempo de una llamada gRPC.

b.- Reducir el timeout de las llamadas gRPC a un 10% menos del promedio encontrado anteriormente. Mostrar y explicar el resultado para 10 llamadas.

c.- Desarrollar un cliente/servidor gRPC de forma tal que siempre se supere el tiempo de timeout. Una forma sencilla puede utilizar el tiempo de timeout como parámetro del procedimiento remoto, donde se lo utiliza del lado del servidor en una llamada a `sleep()`, por ejemplo.

Entrega de la práctica (individual o en grupos de dos alumnos como máximo):

- Se debe entregar un único informe detallando lo realizado para cada ejercicio. Debe tener un formato bien definido identificando materia, trabajo práctico y autor/es. Se debe entregar en formato electrónico con tipo de archivo .pdf, en tamaño de hoja A4.
- Para cada programa modificado o generado para resolver los ejercicios, debe explicarse el cambio o la implementación realizada. Si bien el programa fuente puede estar comentado, el cambio o la implementación realizada debe explicarse en el texto del informe (no es aceptable “ver código fuente” en el informe).
- Se debe entregar en formato electrónico tanto el informe como todo el código fuente usado/desarrollado.