

Universidad Simón Bolívar
Redes de Computadores 1
Juan García 05-38207
Jessica Perez 05-38724
Enero – Marzo 2009

Manual de Uso - Proyecto 3

Nombre de las funciones:

'c_tiempo' : aplicación Cliente RPC

's_tiempo' : aplicación Servidor RPC

Descripción:

'c_tiempo' y 's_tiempo' son usados para sincronizar los relojes de una maquina cliente, con el de una maquina servidor. Para ello usa el Algoritmo de Marzullo, el cual consiste en que el cliente pregunta varias veces la hora al servidor y encuentra un promedio entre los intervalos para hallar con mayor exactitud la el tiempo deseado.

'c_tiempo' se aplica en el cliente y es estrictamente necesario especificar la dirección del servidor, el número de veces que el cliente preguntará al servidor y una opción que permite actualizar la hora de la maquina local. Opcionalmente se le puede pedir que sólo muestre la hora, sin llegar a modificarla. Adicionalmente, la aplicación cliente permite realizar concurrentemente peticiones a varios servidores, dispuestos en un archivo de texto plano, en una misma corrida del programa.

's_tiempo' se corre en el servidor, el cual posee una medida de calidad, que afecta los intervalos que Marzullo utilizara en su calculo. De manera que, dependiendo de la calidad, los intervalos que se obtengan de un servidor dado, aparecerán en la lista final de intervalos, tantas veces repetidos, como sea el numero de calidad. Esto, para que al calcular la hora final, aquellos servidores con alto nivel de calidad, tengan relevancia en el calculo, sobre aquellos con baja calidad.

El protocolo que usaremos en esta entrega sera RPC (Remote Procedure Call). Este protocolo le permite al cliente hacer llamadas a funciones de forma remota, es decir, funciones que poseen su cuerpo de código en otra maquina no local. Para realizar la comunicación mediante RPC usaremos la herramienta Rpcgen.

Sintaxis del Cliente 'c tiempo':

Para ejecutar el programa cliente, se debe ingresar en la línea de comandos, la siguiente instrucción:

'c tiempo -f <archivo_texto> -v <número de preguntas> [-n]'

donde:

-f <nombre_archivo> Indica el archivo de texto en donde se encuentran listados, las direcciones IP o los nombres de los servidores a los cuales se quieren hacer peticiones.

-v <número de preguntas> Indica el número de veces que el cliente preguntará la hora al servidor.

-n Indica que la hora obtenida debe mostrarse por la salida estándar, en lugar de tratar de establecerla como hora del sistema.

Sintaxis del Servidor 's tiempo' :

Para ejecutar el programa servidor, debe colocarse en la línea de comandos, la siguiente instrucción:

's tiempo -c <calidad>'

donde:

-c <calidad> Indica la calidad del servidor que vamos a ejecutar.

“NOTA”: En ambos casos la opción '-t' y '-u' son mutuamente excluyentes, y la opción '-n' es opcional. Además el orden de entrada de los parámetros no afecta el resultado de la función.

Detalles de Implementación:

Anteriormente nuestra aplicación Cliente / Servidor utilizaba los protocolos UDP y TCP para realizar la comunicación, pero en esta nueva entrega mudamos el paradigma y utilizamos RPC. Para ello nos valimos de la herramienta Rpcgen de los sistemas Unix, que al darle un esquema de implementación de las funciones que queremos correr remotamente, el te genera todas las herramientas necesarias para realizar la conexión, dejando ocultas todas las operaciones de conexión y transmisión de datos.

Principalmente las responsabilidades de nosotros los programadores, fue de implementar las funciones que procesan los tiempos que se piden al servidor, generar la concurrencia necesaria en el cliente para que pueda realizar peticiones a varios

servidores e implementar las funciones que el servidor utilizaría para entregarnos las peticiones.

En general la implementación fue parecida a la previa, con la excepción de la herramienta Rpcgen. Primero que nada se crea el archivo 'tiempo.x', el cual contiene la estructura de las funciones que queremos correr remotamente. Al ejecutar 'rpcgen tiempo.x', se generan una serie de archivos que permiten establecer la comunicación entre las maquinas. Nuevamente recordamos que esto es invisible para el programador. Luego procedimos a implementar el cuerpo de las funciones remotas en el archivo 's_fun_tiempo.c', el cual contiene solo dichas funciones. Luego, siguiendo modelos que genera rpcgen, se desarrollo el archivo servidor 's_tiempo.c', el cual se ejecuta en la maquina servidor, y permite recibir el parámetro de calidad por consola, pasándolo a través de archivos a la función encargada de enviar la calidad.

Para finalizar implementamos el archivo cliente, el cual realiza las llamadas a las funciones mediante código especial generado por rpcgen, pero que mediante manejo de estructuras y la función de Marzullo, pudimos salvar dichas peticiones y obtener una hora para setear en nuestra maquina cliente.

En general fue una implementación mas organizada, debido a que la sección de comunicación y establecimiento de sockets fue invisible para nosotros, y lo que se implemento fue como aprovechar el protocolo RPC para realizar las mismas actividades que se hacían con TCP o UDP.

Diagrama de secuencia RPC:

