

TAREA 4

ALDAIR BERNAL BETANCUR

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS

SISTEMAS DISTRIBUIDOS

PEREIRA

2018

¿Qué está mal con los sockets?

Los sockets son una parte fundamental en las redes cliente-servidor, sirven para proveer un mecanismo para establecer una conexión con otro programa

Secuencia de las operaciones

1. El cliente hace llamado a un procedimiento local
2. Un mensaje es enviado por el cliente hacia el sistema remoto
3. Este mensaje es transferido por el kernel al sistema remoto mediante unos protocolos
4. El server ejecuta la llamada de procesamiento en el actual para después ser enviada pasando los argumentos recibidos por el cliente

El cliente continúa con la ejecución del código.

Son dos los principales beneficios de RPC: el programador ahora puede usar el procedimiento la semántica de llamadas y la escritura de aplicaciones distribuidas porque RPC oculta todo el código de red en funciones de código auxiliar.

Usando como referencia el modelo OSI, RPC es una etiqueta de presentación de servicio.

¿Cómo se pasan estos parámetros?

Pasar por valor es simple (simplemente copiar el valor en el mensaje de red). Pasar por referencia es de más complejidad y poco recomendado.

¿Cómo representamos los datos?

En un sistema local no se tienen problemas de incompatibilidad con los datos, Con RPC una máquina remota puede tener diferente ordenación de bytes, diferentes tamaños y diferentes representación de punto flotante.

¿Qué protocolo de transporte podemos usar?

Algunas implementaciones permiten sólo una para ser utilizado (por ejemplo, TCP). La mayoría de las implementaciones RPC apoyan varios y permiten al usuario elegir.

¿Qué pasa cuando las cosas van mal?

Ahora hay más oportunidades para los errores, un servidor puede generar un error, puede haber problemas en la red, el servidor puede bloquearse, o el cliente puede desaparecer mientras el servidor está ejecutando o corriendo el código, La transparencia de las llamadas a procedimiento remoto se rompe aquí.

Programación con llamadas a procesamientos remotos

Los lenguajes más populares hoy (c, c++, java, scheme...) no tienen ningún concepto de procedimientos remotos y por lo tanto son incapaces de generar las funciones necesarias de código auxiliar.

Ventajas de RPC

Usted no tiene que preocuparse de conseguir una única dirección de transporte, El servidor puede unirse a cualquier puerto y registrar el puerto con su servidor de nombres RPC. El cliente se contactara con este servidor para solicitar el número de puerto que corresponde al programa que necesita. El sistema es independiente del transporte.

Primera generación de llamadas a procesamiento remoto

Implementación Sun's RPC

Sun microsystem fue uno de los primeras sistemas en proveer librerías RPC y un servidor. Sun proporciona un compilador que toma la definición de una interfaz de procedimiento remoto y genera las funciones de cliente y del servidor de código auxiliar.

Segunda generación de llamadas a procesamiento remoto: soporte de objetos

Como lenguajes orientados a objetos comenzaron a ganar popularidad en la década de 1980. El mecanismo RPC aún funcionó pero no soporto técnicas de programación orientada a objetos de una manera automatizada, transparente.

Tercera generación RPCs, web services y XML

XML: es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.

SOAP: es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Microsoft.Net: es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.