

Middleware

30221 - Sistemas Distribuidos

Rafael Tolosana Calasanç

Dpto. Informática e Ing. de Sistemas

Lectura Recomendada: capítulo introductorio:

- G. Colouris, J. Dollimore, T. Kindberg and G. Blair.
Distributed systems: Concepts and Design. 5th Edition.
Addison-Wesley. May, 2011. ISBN: 978-0132143011.

Motivación

Motivación

The Procedure Call Model can be defined “as a way of making the networked environment seem completely familiar to application developers, rather than exposing the network directly to them and thus presenting them with a development model so different than they be scared away from writing distributed programs.”

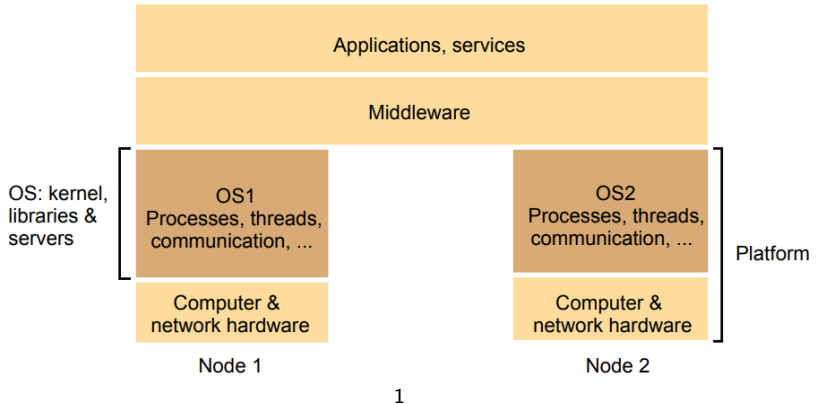
– James E. White , 1976

Motivación

Definición: [Colouris]

- La arquitectura software de un sistema es su estructura en términos de los componentes que la *conforman* y sus *relaciones* entre sí.
- Una arquitectura software es una *abstracción* de un sistema computacional, una abstracción sobre el *código fuente*.
 - Nos muestra aspectos del sistema que nos ayudan a comprenderlo
 - Las arquitecturas nos sirven para trabajar en equipo, comunicación y para mejorar el desarrollo del sistema

Motivación



¹Instructor's Guide for Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design
Edn. 5© Pearson Education 2012

Motivación

Formas de Comunicación entre Componentes

- Comunicación Directa
- Comunicación Indirecta (mediada)
 - Linda, publish-subscribe, colas de mensajes, etc.

Protocolos de Comunicación

- Un protocolo de comunicación es un conjunto de reglas y formatos que se utilizan para la codificación y envío de un mensaje entre procesos.
- UDP y TCP son protocolos de comunicación

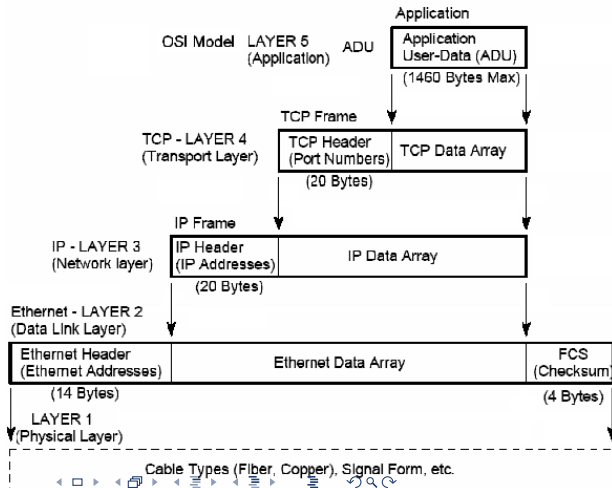
Protocolos de Interacción

- Un protocolo de interacción es una secuencia de mensajes intercambiados entre dos o más procesos para la realización de una tarea determinada
- Request - Reply es un protocolo de interacción

Motivación

Formato de los Mensajes / Datos

CONSTRUCTION OF A TCP/IP-ETHERNET DATA PACKET



Motivación

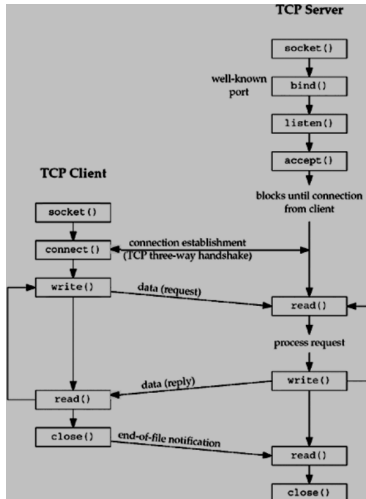
¿Cómo se codifican los datos?

- Serialización / Marshalling

Codificación de los Datos

Codificación de los Datos

TCP / IP



Codificación de los Datos

Operaciones TCP / IP

- `read(data byte[])` // bloqueante
- `write(data byte[])` // *no* bloqueante

Codificación de los Datos

Operaciones TCP / IP

- `read(data byte[])` // bloqueante
- `write(data byte[])` // *no* bloqueante

Retos

- `Byte[]` vs. Estructuras de Datos
 - Demasiados detalles de bajo nivel
- Heterogeneidad: v.gr. de las máquinas

Codificación de los Datos

Aproximaciones para la Codificación de los Datos

- Automática
- Manual (intervención humana)

Codificación de los Datos

Aproximaciones para la Codificación de los Datos

- Automática
- Manual (intervención humana)

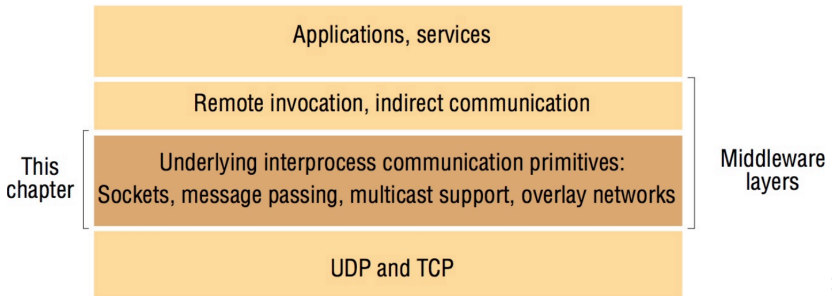
Aproximaciones para la Codificación de los Datos

- Acuerdo entre emisor receptor
- Metadatos incluidos en los mensajes
- Datos en texto → XML / JSON

Middleware

Middleware

Arquitectura SSDD



2

²Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn.
4©Pearson Education 2005

Invocación Remota

Origen del Middleware: RPC

- Consiste en invocar un procedimiento remoto como si fuera local
- El servidor publica una *interfaz*
- Se genera código automático para serializar los datos
- El cliente invoca al procedimiento como si fuera local

Invocación Remota

Origen del Middleware: RPC

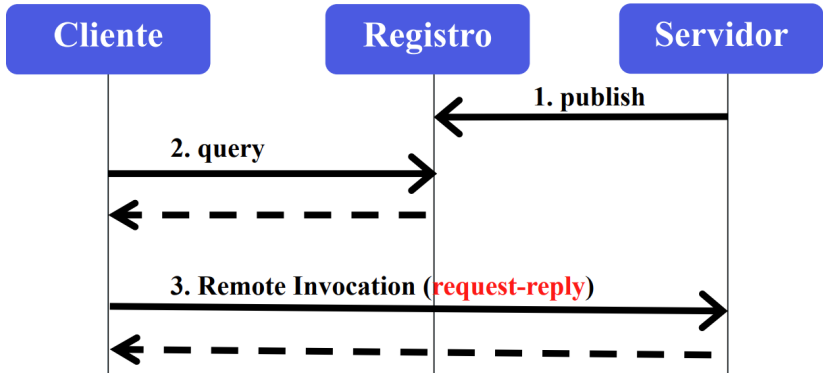
- Consiste en invocar un procedimiento remoto como si fuera local
- El servidor publica una *interfaz*
- Se genera código automático para serializar los datos
- El cliente invoca al procedimiento como si fuera local

Un poco de historia

- Idea original de James E. White, 1976
- En 1984, Bruce J. Nelson y Andre D. Birrell (Xerox) presentaron RPC para implementar SSDD.
- De RPC se derivarían en las décadas siguientes, diferentes modelos: Common Object Request Broker Architecture, Java Remote Method Invocation, Sun RPC, RPC-based Web Services, **GoRPC**, etc.

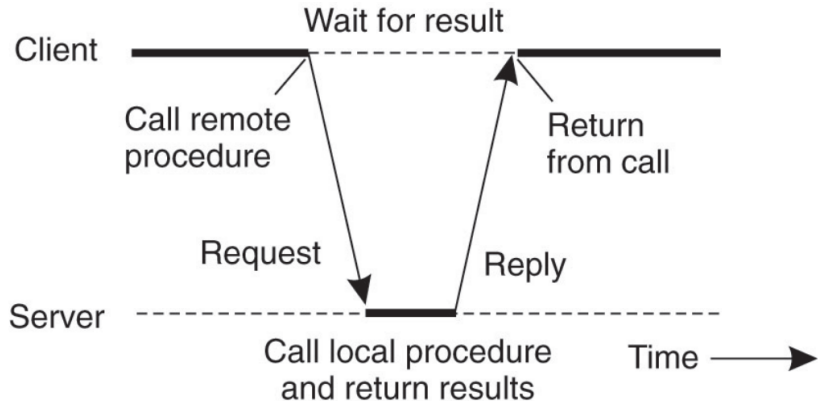
Invocación Remota

Modelo Interacción RPC



Invocación Remota

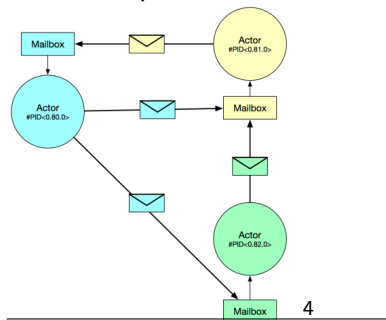
Modelo Interacción RPC



3

Un Actor es un proceso

- puede recibir y enviar mensajes de forma asíncrona
- solo tiene memoria local
- un proceso recibe los mensajes en un buzón
- cada proceso tiene un identificador único PID



⁴Charla donde se explica el Model Actor: https://www.youtube.com/watch?v=7erJ1DV_Tlo



Pearson Education 2005

Comunicación Indirecta

Modelo: Cola de Mensajes

- Emisor y receptor desacoplados temporal y espacialmente
- Comunicación 1 a 1

Comunicación Indirecta

Modelo: Cola de Mensajes

- Emisor y receptor desacoplados temporal y espacialmente
- Comunicación 1 a 1

Modelo: Publish – Subscribe

- Emisor y receptor desacoplados temporal y espacialmente
- Comunicación N a M

Comunicación Indirecta

Modelo: Cola de Mensajes

- Emisor y receptor desacoplados temporal y espacialmente
- Comunicación 1 a 1

Modelo: Publish – Subscribe

- Emisor y receptor desacoplados temporal y espacialmente
- Comunicación N a M

Implementación: El Broker de Mensajes

- Patrón / Componente
- Posibilita comunicación indirecta
- El broker tiene *wrappers*
- El broker puede transformar / adaptar mensajes entre procesos (*wrappers*)

Publish – Subscribe

Comunicación de Grupo

- ¿Cuál es el problema clásico del productor - consumidor?
- ¿Qué sucedería si tuviéramos N productores?
- ¿Qué sucedería si tuviéramos M consumidores?

Publish – Subscribe

Comunicación de Grupo

- Múltiples productores (fuentes de datos)
- Múltiples consumidores
- Los productores ofrecen fuentes de datos
- Los consumidores se registran a una fuente de datos
- Cada vez que el productor produce un dato este se envía a un grupo de consumidores

Publish – Subscribe



⁷Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4 © Pearson Education 2005

Ejemplo: Apache Kafka



8

Resumen

Resumen

Concepto de Middleware

- Abstracción sobre la comunicación
- Serialización / Marshalling automático de los datos
- Modelo de Comunicación / Sincronización

Middleware

30221 - Sistemas Distribuidos

Rafael Tolosana Calasanz

Dpto. Informática e Ing. de Sistemas