Arquitecturas Distribuidas

TEMA 1. Introducción a las arquitecturas distribuidas

Tema 1. ARQUITECTURAS DISTRIBUIDAS: CONCEPTOS BÁSICOS

1. ¿Qué es un sistema distribuido?
2. Servicios
3. Arquitectura
4. Definición de AD
5. Modelos de sistema
6. Retos de diseño
7. Motivación
8. Comunicación entre procesos
    1. Interfaz Socket
    2. Llamada a procedimiento remoto (LPR)
9. Resumen de definiciones

¿Qué es un sistema distribuido (S.D)?

- Coulouris: “Un sistemas cuyos componentes se comunican y coordinan sus acciones sólo mediante el paso de mensajes”.
- Una aplicación que se comunica con otros procesos en una red para coordinar sus acciones de manera que en conjunto realicen una o varias tareas relacionadas

Algunas definiciones básicas

- Recurso: cualquier entidad hardware o software. Por ejemplo: un ordenador, un sistema de almacenamiento de archivos, una impresora, un archivo, un canal de comunicaciones, un servicio, etc.
- Nodo: término genérico usado para representar cualquier dispositivo.
- Proceso: instancia de un programa en ejecución
- Cliente: un consumidor de información.
- Servidor: un proveedor de información.

¿Para qué necesitamos un S.D?

- El objetivo fundamental es **compartir recursos**
- Objetivo secundario: conectar usuarios remotos a recursos remotos **de manera abierta y "escalable"**
  - Abierta: no hay restricciones en el tipo de recursos y componentes que forman el sistema. Los componentes está abiertos a interacciones continuas con otros componentes.
  - "Escalable": el sistema puede acomodar fácilmente incrementos en el número de usuarios y recursos Ejemplo: Internet

¿Compartir?

- Problema: “compartir” no es inmediato, exige comunicación entre componentes, y “entidades” que gestionen los recursos.
- Las “entidades” encapsulan un recurso y regulan su uso.
- Las “entidades” poseen distintas interfaces, modos de funcionamiento y requieren el establecimiento de “políticas” comunes: nombramiento, forma de acceso, etc.

Servicios

- ¿Qué es un “servicio”?
  - Coulouris: “parte diferente de un sistema de computadores que gestiona una colección de recursos relacionados y presenta su funcionalidad a usuarios y aplicaciones”
  - Trabajo o capacidad que se ofrece a los clientes
- ¿Cómo se ofrece? ¿Cómo sabe el cliente lo que se le ofrece?
- Los servicios se ofrecen mediante una **interfaz**: conjunto bien definido de operaciones ofertadas
  - Permite que usuario y proveedor evolucionen independientemente siempre que no cambie la interfaz

Arquitectura

- Arquitectura: abstracción de un sistema, en el que se representa su estructura, sus propiedades externas "visibles" y la relación entre componentes
- No se detalla la implementación de los componentes, sí las relaciones entre ellos.
  - Papeles funcionales: ¿Qué tarea desempeñan?
  - Patrones de comunicación entre ellos

Definición de AD

- El termino “Arquitecturas Distribuidas” (AD) hace referencia a “*una serie de procedimientos, políticas y requerimientos aplicados a la construcción de un sistema distribuido*”, con objeto de:
  - Unificar y simplificar el diseño, facilitando así la construcción y mantenimiento del mismo, y estandarizando su desarrollo (metodologías).
  - Reducir costes.
  - Reutilizar componentes.

Modelo cliente/servidor

- Modelo fundamental
- Servidor: programa en ejecución (**proceso**) que acepta peticiones de otros programas que se están ejecutando
  - Pasivo: recibe peticiones
  - Suele estar continuamente en ejecución
- Cliente: programa (proceso) que solicita una petición (invoca una operación) a un servidor
  - Activo: inicia peticiones.
  - Está en ejecución sólo el tiempo que lo está la aplicación que lo inicia.
  - Cliente y servidor pueden estar en la **misma o diferentes máquinas**
- Cliente/Servidor, por tanto hace referencia al rol desempeñado en una solicitud. Un proceso puede ser cliente y servidor en diferentes momentos.
- Los servidores gestionan los recursos y ofrecen **servicios**

Modelo cliente/servidor

- Problema) Existen servicios que requieren una cooperación mayor que la que se presenta en este modelo.
  - Sol) Modelos basados en objetos remotos
  - Sol) Redes de pares P2P (*peer to peer*). Sigue utilizando el modelo cliente/servidor, solo que ahora en los componentes no se distingue el rol que toma cada uno: todos pueden actuar como servidores y/o clientes

Resumen modelos arquitectónicos

- Modelo cliente/servidor: el modelo fundamental y más utilizado
  - Distinción clara de las funciones de cada componente
  - Servicios pueden ser proporcionados por diferentes servidores
- Procesos de "igual a igual" (pares, P2P)
  - Todos los procesos realizan tareas semejantes
  - Clientes y servidores simultáneamente
- Código móvil
  - Código se transfiere al cliente
  - Ejemplos: Applets, javascript
- Agentes móviles
  - Proceso en ejecución se transfiere completamente a otra máquina
- Clientes ligeros

Características de un SD

- Separación funcional: las funciones se reparten entre diferentes entidades
- Distribución inherente: evidentemente, los recursos se comparten de manera remota, además las tareas se realizan sin que el usuario sea consciente de qué recursos se utilizan ni dónde están localizados
- Heterogeneidad: diversidad de dispositivos, aplicaciones, sistemas operativos, lenguajes de programación, etc.

Retos de diseño de los sistemas distribuidos

- ¿Qué características debería proporcionar nuestro SD?
  - Transparencia
  - Soporte de la concurrencia
  - **Gestión de fallos**
  - Escalabilidad
  - Seguridad
  - **Apertura**: clave del éxito de Internet

Motivación de la asignatura

- Es necesario conocer herramientas, patrones básicos de trabajo, así como los problemas comunes de las de AD.
- Es necesario conocer el funcionamiento de los sistemas distribuidos más utilizados hoy en día

Programa

- Durante este curso:
  - Exploraremos la AD más exitosa y utilizada, y trabajaremos haciendo uso de ella: el WWW.
    - Lenguajes estructurados: HTML, SGML, XML
    - Web dinámica. La evolución de la Web. Web 2.0
    - BBDD relacionales
  - Estudiaremos otros tipos de sistemas distribuidos: LPR, CORBA, P2P .

Tema 1. ARQUITECTURAS DISTRIBUIDAS: CONCEPTOS BÁSICOS

1. ¿Qué es un sistema distribuido?
2. Servicios
3. Arquitectura
4. Definición de AD
5. Modelos de sistema
6. Retos de diseño
7. Motivación
8. **Comunicación entre procesos**
    1. Interfaz Socket
    2. Llamada a procedimiento remoto (LPR)
9. Resumen de definiciones

Comunicación entre procesos (I)

- Proceso: instancia de un programa en ejecución
- Hilo *(thread)*: tarea que se ejecuta concurrentemente (en paralelo) dentro un proceso
- Los hilos comparten memoria con el resto de hilos de un proceso
- Distintos procesos dentro de una misma máquina NO comparten memoria
- *¿Cómo comparten datos distintos procesos?*
  - *Intercambio de mensajes*
- Comunicación entre procesos: técnicas para el intercambio de información (mensajes) entre procesos (local o remotamente)

Comunicación entre procesos (II)

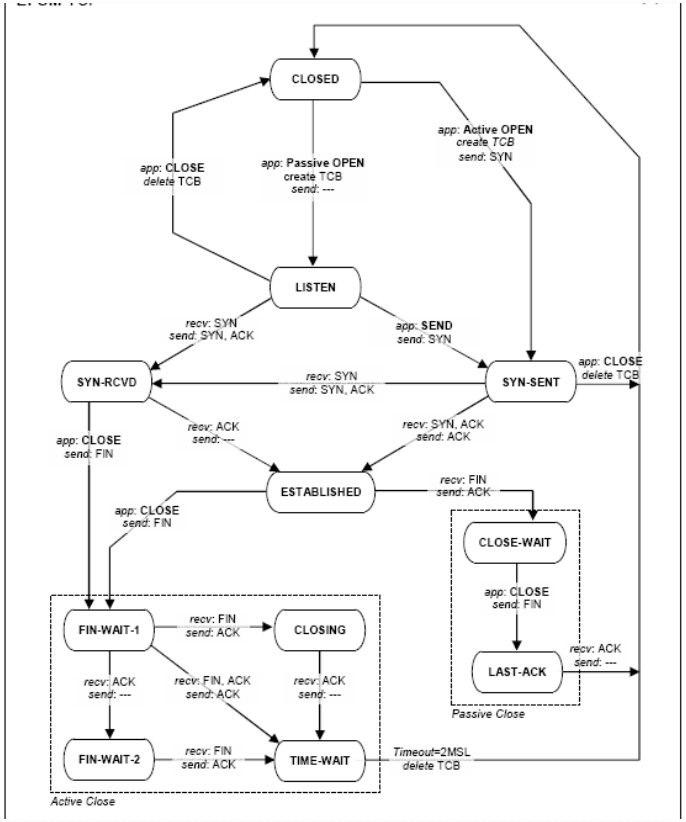
- Procesos pueden estar en la misma máquina (com. local) o en distintas máquinas (com. remota)
- Mecanismos para la comunicación entre procesos:
  - Archivos
  - Señales
  - Tuberías
  - Colas de mensajes
  - Sockets
  - Otros
- Operaciones básicas de paso de mensajes: *enviar* y *recibir*.
- Características de la comunicación entre procesos:
  - Comunicación síncrona o asíncrona.
  - Destinos de los mensajes (dirección IP, puerto)
  - Fiabilidad y ordenación

Interfaz *Socket*

- **Interfaz estándar** que abstrae las tareas de intercomunicación entre procesos
- *Socket()* crea un conector y devuelve un descriptor de la conexión. Acepta 3 parámetros:
  - Dominio: PF_INET, PF_INET6, PF_UNIX
  - Tipo: Stream (fiable OC), datagrama (no fiable, NOC), otros
  - Protocolo: por defecto TCP para stream y UDP para datagrama
- Operaciones:
  - Servidor: socket*(), bind(), listen(), accept(), write(), read(), close().*
  - Cliente: *socket(), connect(), write(), read(), close()*
  - Permiten el transporte de datos sobre canales fiables o no fiables
- Normalmente la implementación de las operaciones la proporciona el propio S.O.

Protocolo TCP

- Mantiene información de estado: ¿en qué paso estoy?, ¿qué puede ocurrir? ¿a qué estado voy?
- Se describe mediante una máquina de estados
- Proporciona un servicio de transporte de datos fiable y orientado a la conexión: requiere establecimiento y cierre de conexión



Llamada a procedimiento remoto (LPR)

- Extensión del modelo de programación para aplicarlo a programas distribuidos: invocación de operaciones en distintos procesos en la misma o distintas máquinas.
- Llamada a procedimiento remoto (*Remote Procedure Call*, RPC) = Invocación a un método remoto (*Remote Method Invocation, RMI),* cuando se utiliza POO.
  - Un objeto instanciado en un proceso invoca a un método (función) de un objeto instanciado en otro proceso (local o remoto).
  - La capa RMI es un software que permite realizar estas operaciones de manera *transparente* al programador => **Abstracción de las operaciones realizadas**