

Sistemas Distribuidos

Grado en Ingeniería Informática

T2: Comunicación indirecta (1ª parte)

Departamento de Ingeniería Informática
Universidad de Cádiz



Curso 2017 – 2018

Indice

1 Introducción

2 Comunicación en grupo



Sección 1 | Introducción



Introducción

- La comunicación indirecta se lleva a cabo a través de un intermediario.
- No existe acoplamiento directo entre el emisor y uno o más receptores.
- Algunas técnicas de comunicación indirecta:

Comunicación en grupo Comunicación mediante una abstracción de grupo en la que el emisor no es consciente de la identidad de los receptores.

Sistemas publicador-suscriptor Una familia de enfoques que se caracterizan por enviar eventos a múltiples receptores a través de un intermediario.

Sistemas de cola de mensajes Los mensajes se envían a una cola, extraídos más tarde por los receptores.

Comunicación directa

Recordatorio

- Acoplamiento directo entre un emisor y un receptor.
- Rigidez ante posibles cambios.
- Ejemplo de una interacción cliente-servidor:
 - Gran dificultad para sustituir un servidor por otro con funcionalidades similares.
 - Si el servidor falla, afectará al cliente (que tratará el fallo).



Comunicación indirecta (I)

Definición

Comunicación entre entidades en un sistema distribuido a través de un intermediario sin acoplamiento directo entre emisor y receptores.

R. Needham, M. Wilkes y D. Wheeler

Todos los problemas en informática se pueden resolver mediante otro nivel de indirección.

Desacoplamiento de espacio El emisor no sabe (ni necesita saber) la identidad de los receptores, y viceversa. Por tanto, los participantes (emisores y receptores) pueden ser reemplazados, actualizados, replicados o migrados.

Desacoplamiento de tiempo El emisor y los receptores pueden tener tiempos de vida independientes. No es necesario que emisor y receptores coexistan al mismo tiempo.

Comunicación indirecta (II)

Ventajas

- La comunicación indirecta es muy utilizada en sistemas distribuidos donde el cambio es anticipado.
- Ejemplo: entornos móviles donde los usuarios deben conectarse y desconectarse rápidamente de la red global.
- Estos sistemas deben ser gestionados para ofrecer servicios más fiables.
- Este tipo de comunicación también se utiliza para el envío de eventos en sistemas distribuidos, donde los receptores son desconocidos.
- La infraestructura de Google utiliza la comunicación indirecta.

Comunicación indirecta (III)

Inconvenientes

- Sobrecarga de rendimiento introducida al añadir el nivel de indirección.
- *No existe ningún problema de rendimiento que no pueda resolverse eliminando el nivel de indirección (J. Gray).*
- Estos sistemas son más difíciles de gestionar con precisión.

Comunicación indirecta (IV)

Acoplamiento de espacio y tiempo en sistemas distribuidos

	Tiempo acoplado	Tiempo desacoplado
Espacio acoplado	Comunicación dirigida hacia 1 o varios receptores. Receptores deben existir en ese momento. Ej.: paso de mensajes.	Comunicación dirigida hacia 1 o varios receptores. Emisores y receptores pueden tener tiempos de vida diferentes.
Espacio desacoplado	Emisor no necesita saber la identidad de receptores. Receptores deben existir en ese momento. Ej.: IP <i>multicast</i> .	Emisor no necesita saber la identidad de receptores. Emisores y receptores pueden tener tiempos de vida diferentes. Ej.: comunicación indirecta .

Comunicación indirecta (V)

Comunicación asíncrona vs desacoplamiento de tiempo

Comunicación asíncrona

- Un emisor envía un mensaje y entonces continúa (sin bloquear).
- No es necesario que el emisor se encuentre con el receptor al mismo tiempo.

Desacoplamiento de tiempo

- Añade una dimensión extra: el emisor y los receptores pueden existir en momentos diferentes.
- Ejemplo: el receptor podría no existir en el instante en que se inicia la comunicación.

Sección 2 | Comunicación en grupo



Introducción

- La comunicación en grupo es un ejemplo de paradigma de comunicación indirecta; fundamental en los sistemas distribuidos.
- Proporciona un servicio en el que un mensaje se envía a un grupo y entonces se entrega a todos los miembros de ese grupo.
- El emisor no es consciente de las identidades de los receptores.
- Representa una abstracción de la comunicación *multicast*.
- Áreas de aplicación:
 - Difusión fiable de información a un gran número de clientes.
 - Apoyo para aplicaciones colaborativas, en las que deben enviarse eventos a múltiples usuarios, preservando una vista común de usuario (ej.: juegos multiusuario).
 - Apoyo a un rango de estrategias tolerantes a fallos (actualización consistente de datos replicados o implementación de servidores replicados).
 - Apoyo para la monitorización y gestión de sistemas (ej.: estrategias de balanceo de carga).

El modelo de programación (I)

- Los procesos deben unirse o dejar un grupo.
- Los procesos pueden enviar un mensaje a este grupo, que será propagado a todos los miembros con garantías de fiabilidad y orden.
- La comunicación en grupo implementa una comunicación *multicast*.

Broadcast Comunicación a todos los procesos del sistema.

Unicast Comunicación a un único proceso.

- Un proceso utiliza sólo una operación *multicast* para enviar un mensaje a cada grupo de procesos (en Java esta operación es `aGroup.send(aMessage)`):
 - Utilización eficiente del ancho de banda.
 - Se minimiza el tiempo total para entregar el mensaje a todos los destinatarios.
 - Garantía de entrega a todos los destinatarios.
 - Fiabilidad y orden de los mensajes.

El modelo de programación (II)

Grupos de procesos

- Grupos donde las entidades de comunicación son procesos:
 - Los mensajes se entregan a los procesos.
 - Los mensajes son normalmente vectores desestructurados de bytes.
- Ejemplo: *JGroups toolkit*.
- Más utilizados que los grupos de objetos.

El modelo de programación (III)

Grupos de objetos

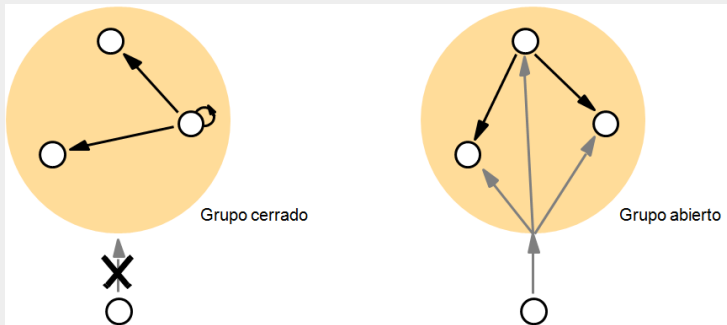
- Ofrecen un enfoque de alto nivel para la computación de grupos.
- Un grupo de objetos es una colección de objetos que procesan el mismo tipo de invocaciones concurrentemente.
- Los objetos del cliente no necesitan ser conscientes de la replicación.
- Invocan operaciones en un objeto local y único que actúa como un proxy para el grupo.
- El proxy usa un sistema de comunicación de grupo para enviar las invocaciones a los miembros de los grupos de objetos.
- Ejemplo: *Electra*.

El modelo de programación (IV)

Grupos cerrados y abiertos

Cerrado Únicamente los miembros del grupo pueden realizar *multicast* dentro del grupo. Un proceso se entrega a sí mismo cualquier mensaje que disemine al grupo.

Abierto Procesos externos al grupo pueden realizar *multicast*.



El modelo de programación (V)

Grupos solapados y no solapados

Solapados Las entidades (procesos u objetos) podrían ser miembros de distintos grupos.

No solapados Los miembros no se solapan, cada proceso pertenece a un único grupo.

Sistemas síncronos y asíncronos

Aspectos de implementación (I)

Fiabilidad en las comunicaciones

El *multicast* fiable tiene 3 propiedades:

- Integridad** El mensaje que se recibe es el mismo que el que se envió y se entrega correctamente una única vez.
- Validez** El mensaje que se envía será finalmente entregado.
- Acuerdo** Si el mensaje se entrega a un proceso, entonces será entregado a todos los procesos del grupo.

Aspectos de implementación (II)

Orden relativo de los mensajes entregados a múltiples destinos

Los servicios de comunicación en grupo ofrecen *multicast* ordenado, con algunas de estas propiedades de ordenación:

- FIFO** Si un proceso envía un mensaje antes que otro, se entregará en este orden a todos los procesos del grupo.
- Causal** Si un mensaje ocurre antes que otro mensaje (relación causal) también será así en la entrega de los mensajes asociados a todos los procesos.
- Total** Si un mensaje se entrega antes que otro en un proceso, entonces se utilizará el mismo orden en todos los procesos.

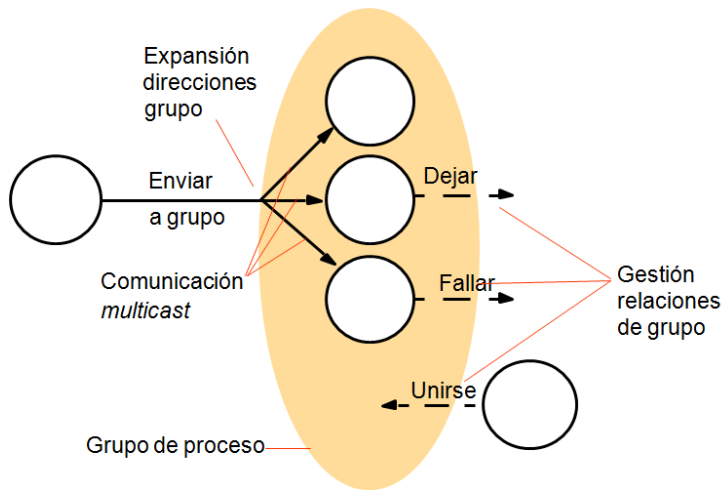
Aspectos de implementación (III)

Gestión de las relaciones de grupo

- Las entidades que participan en la comunicación pueden **unirse** o **dejar** el grupo, e incluso **fallar**.
- Un servicio de relación de grupo realiza 4 tareas:
 - **Ofrecer una interfaz para los cambios en las relaciones de grupo:** El servicio ofrece operaciones para crear y destruir grupos de procesos y añadir o eliminar un proceso de un grupo. Un proceso puede pertenecer a varios grupos (solapamiento de grupo).
 - **Detección de fallo:** El servicio monitoriza colisiones y miembros inalcanzables debido a fallos de comunicación. El detector marcará los procesos como *Suspected* o *Unsuspected*.
 - **Notificación de los cambios a los miembros del grupo:** El servicio notifica a los miembros cuando se añade un proceso o se excluye.
 - **Realizar una expansión de las direcciones de grupo:** Cuando un proceso disemina un mensaje proporciona el identificador de grupo (no una lista de procesos del grupo). El servicio obtendrá del identificador las relaciones de grupo para realizar la entrega.

Aspectos de implementación (IV)

Rol de gestión de las relaciones de grupo



Bibliografía



Coulouris, G.; Dollimore, J.; Kindberg, T.
Distributed Systems: Concepts and Design (5^a ed.)
Addison-Wesley, 2012.

