

Introducción a los sistemas distribuidos

Las computadoras eran grandes y caras. ~ Las minicomputadoras costaban por lo general cientos de miles de dólares cada una.

A partir de la mitad de la década de 1980, dos avances tecnológicos comenzaron a cambiar esta situación.

1. Desarrollo de poderosos microprocesadores. 16, 32 e incluso 64 bits
2. Invención de redes de área local de alta velocidad (**LAN**)

Redes de área local (LANs)

Permiten conectar docenas e incluso cientos de máquinas dentro de un edificio de tal forma que se pueden transferir pequeñas cantidades de información entre ellas en un "milisegundo".

Las cantidades mayores de datos se pueden desplazar entre las máquinas a razón de 10 a 100 millones de bits/segundo o más.

Redes de área amplia (WANs)

Permiten que millones de máquinas en toda la Tierra se conecten con velocidad que varían de 64 Kbps a GB/segundo para ciertas redes experimentales avanzadas.

Note

El resultado neto de estas tecnologías es que hoy en día es fácil reunir sistemas de cómputo compuestos por un gran número de CPU conectados mediante una red de alta velocidad. Reciben el nombre de **sistemas distribuidos**.

¿Qué es un sistema distribuido?

Un sistema distribuido es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora.

Aspectos de la definición:

- **Hardware:** Las máquinas son autónomas.
- **Software:** Los usuarios piensan que el sistema es como una única computadora.

Red de estaciones de trabajo en un departamento de una universidad/compañía, fábrica de robots, banco con cientos de sucursales por todo el mundo

Ventajas con respecto a los centralizados

- Tienen una proporción precio/desempeño mucho mejor que al de un sistema centralizado. es mejor tener varios CPU en el mismo sistema que uno sólo
- Permite la descentralización, tanto para los trabajadores como para el sistema en sí.
- Mayor confiabilidad. Al distribuir la carga de trabajo en muchas máquinas, **la falla de un circuito descompondrá una máquina y el resto seguirá intacto.** usado en el control de reactores nucleares y de aviación

- Permite un desarrollo gradual conforme surjan las necesidades, añadiendo procesadores al sistema. **ESCALABILIDAD**
- Los usuarios comparten información de manera conveniente.

Desventajas

- Desconocimiento sobre el software distribuido específico **tipos de sistemas operativos, lenguajes de programación y aplicaciones**. *¿Cuánto deben saber los usuarios sobre la distribución? ¿Qué tanto debe hacer el sistema y que tanto los usuarios?*
- La dependencia del sistema de la red y la pérdida o saturación de esta puede negar ventajas.
- Las personas pueden tener un acceso que no deberían a datos sensibles.

Formas de organizar N computadoras

Elemento	Sistema operativo de red	Sistema operativo distribuido	Sistema operativo de multiprocesador
¿Se ve como un uniprocador virtual?	No	Sí	Sí
¿Todos tienen que ejecutar el mismo sistema operativo?	No	Sí	Sí
¿Cuántas copias del sistema operativo existen?	N	N	1
¿Cómo se logra la comunicación?	Archivos compartidos	Mensajes	Memoria compartida
¿Se requiere un acuerdo en los protocolos de la red?	Sí	Sí	No
¿Existe una cola de ejecución?	No	No	Sí
¿Existe una semántica bien definida para los archivos compartidos?	Por lo general no	Sí	Sí

Figura 1-12. Comparación de tres formas distintas de organizar n CPU.

Aspectos del diseño de los sistemas distribuidos

Transparencia

Forma de lograr la imagen de un sistema.

Se conoce a un sistema como **transparente** cuando hace creer que la colección de máquinas es un sistema de tiempo compartido de un procesador.

Se puede lograr en dos niveles distintos:

- Ocultar la distribución a los usuarios.

- Hacer que el sistema sea transparente para los programas. diseñar interfaz de llamadas al sistema para que no se vea la presencia de varios procesadores

Tipo	Significado
Transparencia de localización	Los usuarios no pueden indicar la localización de los recursos
Transparencia de migración	Los recursos se pueden mover a voluntad sin cambiar sus nombres
Transparencia de réplica	Los usuarios no pueden indicar el número de copias existentes
Transparencia de concurrencia	Varios usuarios pueden compartir recursos de manera automática
Transparencia de paralelismo	Las actividades pueden ocurrir en paralelo sin el conocimiento de los usuarios

Figura 1-13. Distintos tipos de transparencia en un sistema distribuido.

Flexibilidad

La mejor manera de evitar los problemas es mantener abiertas las opciones.

Existe una discusión entre si cada máquina debería ser con un núcleo monolítico o micronúcleo.

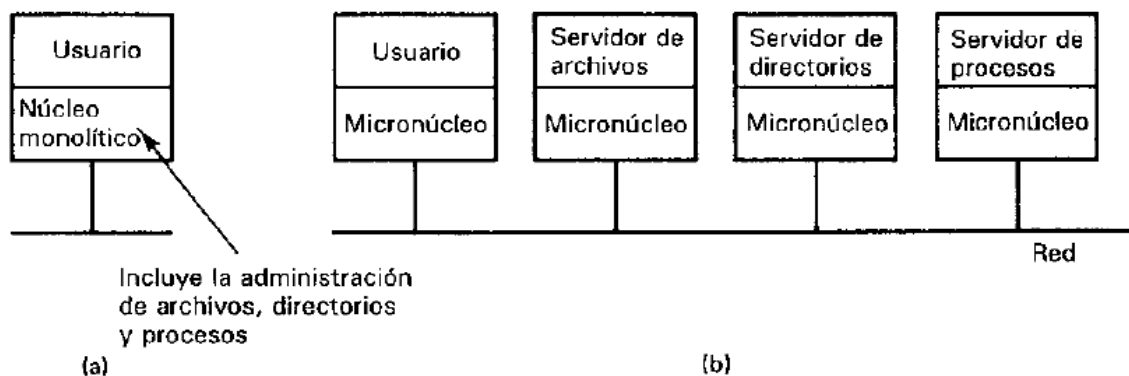


Figura 1-14. (a) Núcleo monolítico. (b) Micronúcleo.

El micronúcleo o microkernel es más flexible, ya que casi no hace nada. Proporciona cuatro servicios mínimos:

1. Un mecanismo de comunicación entre procesos.
2. Cierta administración de la memoria.
3. Una cantidad limitada de planificación y administración de procesos de bajo nivel.
4. Entrada/salida de bajo nivel.

La única ventaja potencial del núcleo monolítico es el rendimiento.

Confiabilidad

Si una máquina falla, alguna otra máquina se va a encargar del trabajo.

Los datos confiados no deben perderse ni mezclarse.

Los archivos y otros recursos deben ser protegidos contra el uso no autorizado.

Disponibilidad

Fracción de tiempo en que se puede utilizar el sistema.

Se puede mejorar mediante:

- Diseño que no exija el funcionamiento simultáneo de un número sustancial de componentes críticos.
- Redundancia. Duplicación de piezas clave de hardware y software.

Tolerancia de fallas

¿Qué pasa si un servidor falla y vuelve a arrancar de manera súbita?

Los sistemas distribuidos se pueden diseñar de forma que escondan las fallas y las oculten de los usuarios.

Es posible construirlo de forma que los usuarios no noten la pérdida de uno o dos servidores.

El truco está en arreglar esta cooperación de modo que no añada un costo sustancial al sistema cuando anda normal.

Desempeño

La comunicación, factor esencial de un sistema distribuido es algo lenta por lo general. Con frecuencia, hay que minimizar el envío de mensajes para optimizar el **desempeño**.

Warning

La dificultad de esta estrategia es que la mejor forma de mejorar el desempeño es tener muchas actividades en ejecución paralela en distintos procesadores. **Requiere el envío de muchos mensajes.**

Prestar atención al **tamaño de grano** de todos los cálculos:

Paralelismo de grano fino: Trabajos que implican gran número de pequeños cálculos y que interactúan en gran medida con otros.

Paralelismo de grano grueso: Trabajos que implican grandes cálculos y bajas tasas de interacción.

Escalabilidad

La mayor parte de los sistemas distribuidos están diseñados para trabajar con unos cuantos cientos de CPU.

Permite que a la arquitectura actual se le pueda adicionar más poder de cómputo.

Concepto	Ejemplo
Componentes centralizados	Un solo servidor de correo para todos los usuarios
Tablas centralizadas	Un directorio telefónico en línea
Algoritmos centralizados	Realización de un ruteo con base en la información completa

Figura 1-15. Cuellos de botella potenciales que los diseñadores deben intentar evitar en los sistemas distribuidos de gran tamaño.

Características que distinguen a los algoritmos descentralizados de los centralizados

1. Ninguna máquina tiene la información completa acerca del estado del sistema.
2. Las máquinas toman decisiones sólo con base en la información local.
3. La falla de una máquina no arruina el algoritmo.
4. No existe una hipótesis implícita de la existencia de un reloj global.

Redes

Arquitectura de comunicaciones

Software que da soporte a un grupo de computadores en red. Proporciona soporte para aplicaciones distribuidas `email`, `transf. de ficheros` y `acceso a terminales remotos`.

Cada computador tiene su propio sistema operativo y es posible tener una mezcla **siempre y cuando todas las máquinas soporten la misma arquitectura de comunicaciones**.

La más utilizada es el conjunto de protocolos TCP/IP

Sistema operativo de red

Red de máquinas. `estaciones de trabajo de un solo usuario` y `una o más máquinas servidoras`

Proporcionan servicios de red o aplicaciones. `almacenamiento de ficheros` y `gestión de impresión`

Cada computador tiene su propio sistema operativo.

El sistema operativo de red es un añadido al sistema operativo local. `permite a las máquinas interactuar con los servidores`

El usuario conoce la existencia de múltiples computadores.

Normalmente se usa para dar soporte a aplicaciones de red

Sistema operativo distribuido

Sistema operativo común compartido por una red de computadores.

Parece un sistema operativo normal centralizado para los usuarios, pero les proporciona acceso transparente a los recursos de diversas máquinas.

Puede depender de una **arquitectura de comunicaciones** para las funciones básicas de comunicación, pero normalmente incorpora un conjunto de funciones de comunicación más sencillas y con mayor eficiencia.

La necesidad de una arquitectura de protocolos

Tareas típicas a realizar

1. El sistema emisor debe activar el enlace directo de comunicación de datos o debe informar a la red de comunicaciones de la identidad del sistema destinatario deseado.
2. El sistema emisor debe verificar que el sistema de destino está preparado para recibir datos.
3. La aplicación de transferencia de ficheros del sistema origen debe verificar que el programa de gestión de ficheros del sistema destino está preparado para aceptar y almacenar el fichero de ese usuario en particular.
4. Si los formatos de los ficheros o las representaciones de datos en los sistemas son incompatibles, uno de los dos sistemas deberá ejecutar una función de traducción de formato.

El intercambio de información entre computadoras con finalidad cooperativa se conoce como **comunicación de computadoras**.

Cuando uno o más computadores se interconectan a través de una red de comunicación, el conjunto de computadores se denomina **red de computadores**.

Protocolo

Se utiliza para comunicar entidades de diferentes sistemas.

Conjunto de reglas que gobiernan el intercambio de datos entre dos entidades.

TFTP, TCP/IP, UDP, IPV6

Elementos principales

Sintaxis: Formatos de datos y niveles de señales.

Semántica: Información de control para realizar coordinación y gestión de errores.

Temporización: Ajuste de velocidades y secuenciamiento.

Capas TCP/IP

CAPA APLICACIÓN
CAPA DE TRANSPORTE o <i>máquina a máquina</i>
CAPA DE INTERNET
CAPA DE ACCESO RED

CAPA APLICACIÓN
CAPA FÍSICA

Capa aplicación: Contiene la lógica necesaria para soportar las aplicaciones de usuario.

Capa de transporte: Asegura que los datos lleguen a la aplicación destino y que los reciba en el orden que fueron enviados.

Protocolo de internet: El **protocolo de internet (IP)** se utiliza en esta capa **capa de acceso red** para proporcionar la función de encaminamiento a través de múltiples redes.

Capa de acceso red: Se preocupa del intercambio de datos entre un sistema final **servidor, estación de trabajo** y la red a la que está unido, es decir, enviar datos a través de una red para dos sistemas finales unidos a la misma.

Capa física: Cubre la interfaz física entre un dispositivo de transmisión de datos y un medio de transmisión o red. Se preocupa de las características del medio de transmisión, la naturaleza de las señales, la tasa de datos, etc.

Aplicaciones estandarizadas para su uso con TCP: SMTP, FTP, TELNET

Socket

Permite la comunicación entre un proceso cliente y un proceso servidor y puede ser orientado a conexión o no orientado a conexión.

Se lo puede considerar como *un punto final entre una comunicación*.

Cuando se comunican el socket del cliente con el socket del servidor, los dos computadores pueden intercambiar datos.

Está formado por la concatenación de **un valor de puerto y una dirección IP**. Es único en internet.

Se utiliza para definir una **interfaz de programación de aplicaciones (API)** **interfaz genérica de comunicaciones para escribir programas que usan TCP y UDP**.

Computación cliente/servidor

Interfaz de programación de aplicaciones (API): Conjunto de funciones y programas que permiten a los clientes y servidores comunicarse.

Cliente: Elemento de la red que solicita información. Puede interrogar a una base de datos o solicitar información a un servidor. **PC o estación de trabajo**

Middleware: Conjunto de controladores, API, y software adicional que mejoran la conectividad entre una aplicación cliente y un servidor.

Base de datos relacional: Base de datos en la que el acceso a la información está restringido por la selección de filas que satisfacen todos los criterios de búsqueda.

Servidor: Computador que almacena la información para los clientes de la red. **estación de trabajo potente, minicomputador, mainframe**

Lenguaje estructurado de consultas (SQL): Lenguaje desarrollado por IBM y estandarizado por ANSI que permite acceder, crear, actualizar e interrogar bases de datos relacionales.

Características

- Es imperativo que los usuarios tengan aplicaciones de fácil manejo en sus sistemas.
- A pesar de que las aplicaciones están dispersas, se realiza un esfuerzo por centralizar las bases de datos corporativas, y muchas funciones de utilidad y de gestión de redes.
- Existe un compromiso de mantener un sistema abierto y modular por parte de las organizaciones de usuarios y vendedores. Más libertad de selección de productos para el usuario.
- La red es fundamental. Alta prioridad de la gestión y seguridad de la red.

Important

La característica fundamental de una arquitectura cliente/servidor es **la distribución de las tareas de la aplicación entre el cliente y el servidor.**

Estación de trabajo cliente

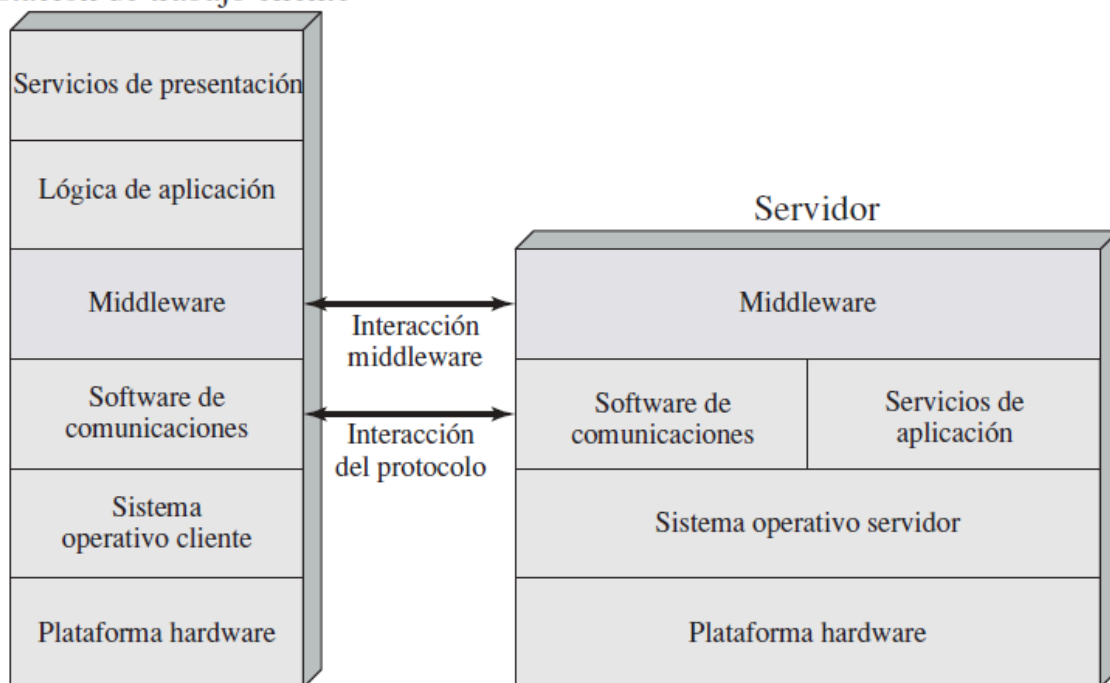


Figura 14.8. El papel del middleware en la arquitectura cliente/servidor.

Middleware

Conjunto de interfaces de programación y protocolos estándares entre la aplicación y el software de comunicaciones y el sistema operativo.

Tiene como propósito básico permitir que una aplicación o usuario en el cliente acceda a una variedad de servicios en el servidor sin preocuparse de las diferencias entre servidores.

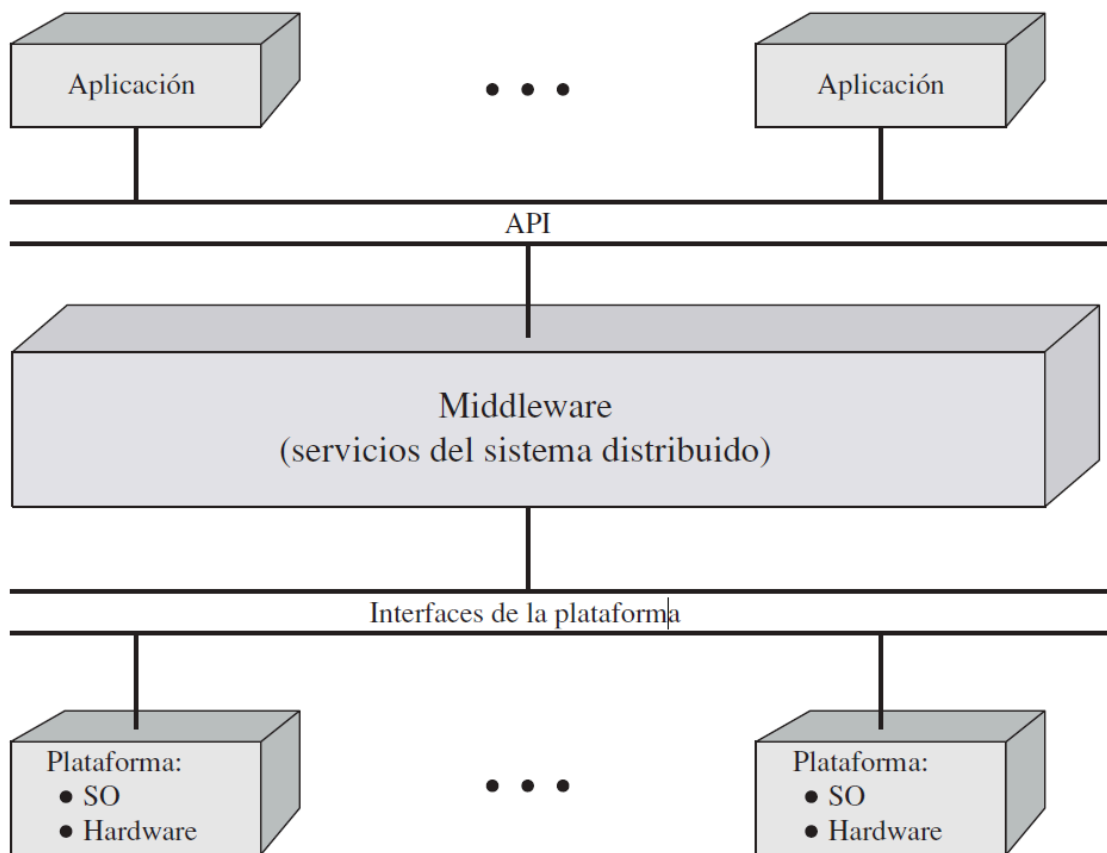


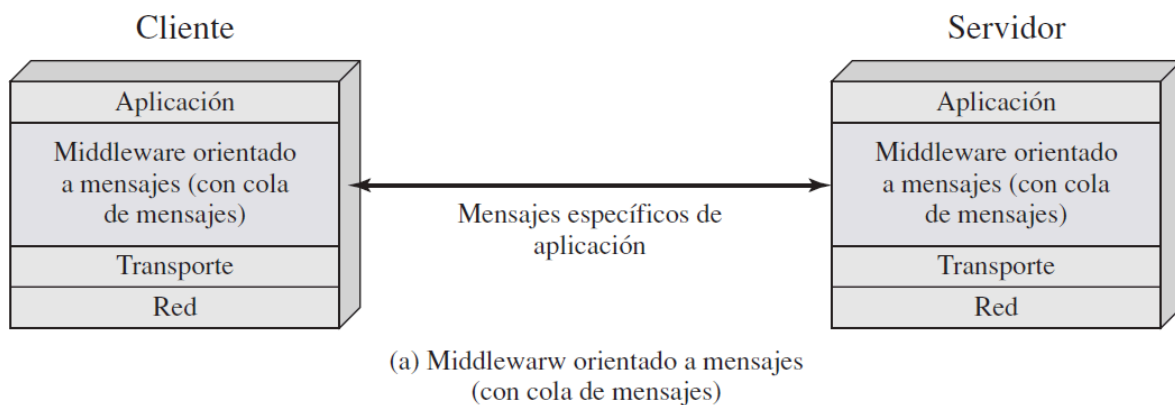
Figura 14.9. Visión lógica del middleware.

Suelen estar basados en uno o dos mecanismos:

- Paso de mensajes
- Llamadas a procedimiento remoto

Paso de mensajes distribuido

Las computadoras **no comparten la memoria principal**; cada una es un sistema aislado.



Send() & Receive()

Fiable VS. No fiable

Fiable

Garantiza la entrega si es posible. No es necesario informar al proceso emisor que el mensaje fue enviado.

Usan un protocolo de transporte fiable y realizan comprobaciones de errores, acuse de recibo (*ack*), retransmisión y reordenamiento.

TCP

No fiable

Envía el mensaje por la red sin informar el éxito o fracaso.

Reduce la complejidad y la sobrecarga de procesamiento y comunicación del servicio de paso de mensajes.

UDP

Bloqueante VS No bloqueante

Bloqueante

Suspende el proceso hasta que el mensaje ha sido transmitido o hasta que recibe un acuse de recibo.

No bloqueante

No se suspende el proceso como resultado de realizar un `send` o `receive`.

Proporciona un uso eficiente y flexible de los servicios de paso de mensajes por parte de los procesos.

Tiene como **desventaja** la dificultad de chequear y depurar programas que utilicen estas primitivas.

Llamadas a procedimiento remoto

Permite a los programas en diferentes máquinas interactuar a través del uso de llamadas a procedimiento.

Se puede ver como un refinamiento del paso de mensajes **fiable y bloqueante**.

Ventajas

1. Son una abstracción ampliamente aceptada, utilizada y entendida.
2. Permite especificar las interfaces remotas como un conjunto de operaciones con nombre y tipo de dato.
3. Puede generar el código de comunicación para una aplicación de forma automática.
4. Los desarrolladores pueden escribir módulos cliente y servidor que se pueden mover entre computadoras y sistemas operativos con pocas modificaciones.

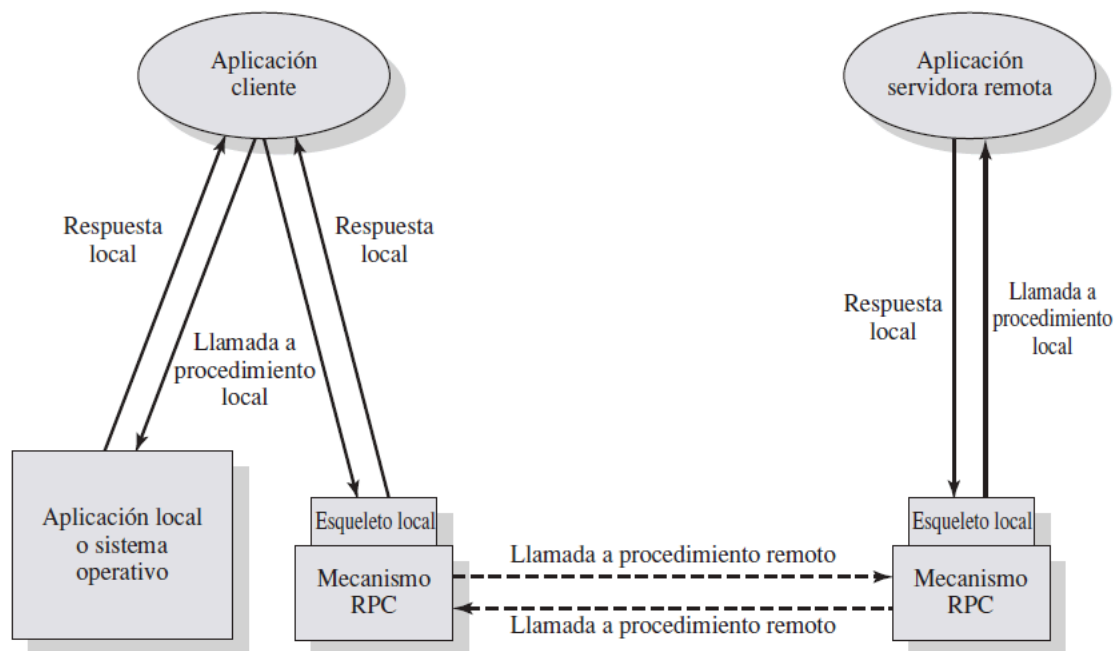
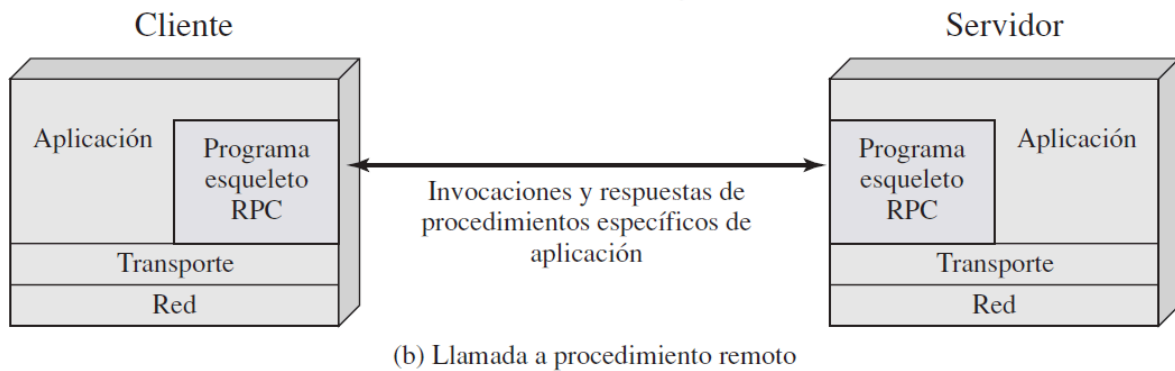


Figura 14.12. Mecanismo de llamadas a procedimiento remoto.



(b) Llamada a procedimiento remoto

Síncronos VS. Asíncronos

Síncronas

El proceso llamante espera hasta que el proceso llamado devuelva el valor.

Fácil de entender y programar. Predecible.

Asíncronas

No bloquean al llamante. Las respuestas se pueden recibir como y cuando sean requeridas, permitiendo al cliente ejecutar en paralelo con el servidor.

Cluster

Sistemas que proporcionan un alto rendimiento y alta disponibilidad. Atractivos para aplicaciones de servidor.

Grupo de computadoras completas e interconectadas que trabajan juntas como un recurso de computación unificado y pueden crear la ilusión de ser *una única máquina*.

Cada computadora de un *Cluster* se denomina *nodo*.

Beneficios

Escalabilidad absoluta

Es posible crear un gran *cluster* que supere la potencia de la mayor de las máquinas.

Puede tener decenas o centenas de máquinas, cada una con un multiprocesador.

Escalabilidad incremental

Se configura de tal manera que sea posible añadir nuevos sistemas en pequeños incrementos.

Alta disponibilidad

El fallo de uno de los *nodos* no significa pérdida de servicio.

Relación precio/prestaciones

Es posible hacer un *cluster* con igual o mayor poder computacional que una única máquina mayor, con menor coste.

Configuración del cluster

Tabla 14.2. Métodos de *cluster*: beneficios y limitaciones.

Método de <i>Cluster</i>	Descripción	Beneficios	Limitaciones
Pasivo en Espera	En caso de fallo en el servidor primario, un servidor secundario toma control.	Fácil de implementar.	Alto coste debido a que el servidor secundario no está disponible para procesar otras tareas.
Secundario Activo	El servidor secundario también se utiliza para procesamiento de tareas	Coste reducido porque el servidor secundario puede ser utilizado para procesamiento.	Creciente complejidad.
Diferentes Servidores	Cada servidor tiene sus propios discos. Los datos se copian continuamente del servidor primario al secundario.	Alta disponibilidad.	Alta sobrecarga de red y de servidor debido a las operaciones de copia.
Servidores Conectados a Discos	Los servidores están unidos a los mismos discos, pero cada servidor posee sus propios discos. Si un servidor falla el otro servidor toma control de sus discos.	Sobrecarga de red y de servidores reducida debido a la eliminación de las operaciones de copia.	Normalmente requiere tecnologías de replicación de discos o RAID para compensar el riesgo de fallo de disco.
Servidores Comparten Discos	Varios servidores comparten acceso a disco de forma simultánea.	Baja sobrecarga de red y de servidores. Reducido riesgo de periodos de inactividad causados por fallos de disco.	Requiere software de gestión de cerrojos. Normalmente se utiliza con tecnologías de replicación de discos o RAID.

Arquitectura de un cluster

Funciones o servicios deseables

- **Punto de entrada único:** Usuario se autentifica en el *cluster* y no en una computadora.
- **Jerarquía de ficheros única:** Los usuarios ven una sola jerarquía de directorios bajo el directorio raíz.
- **Punto de control único:** Hay un nodo por defecto encargado de gestionar y controlar el *cluster*.
- **Red virtual única:** Cualquier nodo puede acceder a otro punto del *cluster*.
- **Espacio de memoria único:** Permite a los programas compartir variables.
- **Sistema de control de trabajos único:** Un usuario puede enviar su trabajo sin especificar la computadora que lo ejecutará.
- **Interfaz de usuario único:** Interfaz gráfico común entre usuarios.
- **Espacio de E/S único:** Cualquier nodo puede acceder remotamente a un periférico de E/S o disco.
- **Espacio de procesos único:** Un proceso en cualquier nodo puede crear o comunicarse con otro proceso en un nodo remoto.
- **Puntos de control:** Salva el estado del proceso y los resultados intermedios para poder recuperarse luego de un fallo.
- **Migración de procesos:** Permite el balanceado de carga.

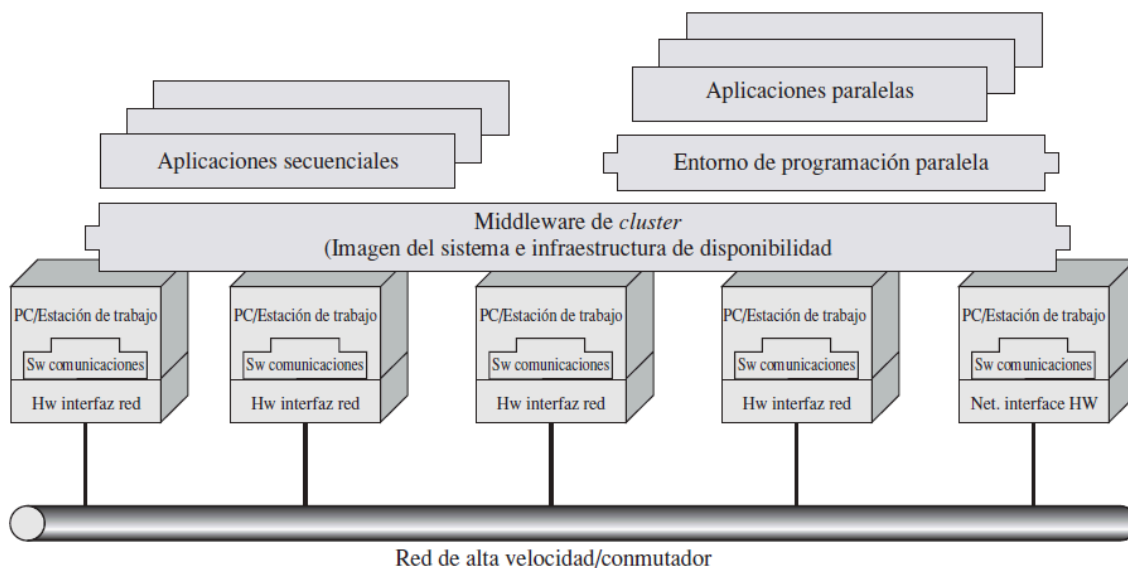


Figura 14.14. Arquitectura de computación *cluster* [BUY99a].