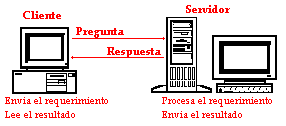
**Sistemas Cliente/Servidor**

El Cliente-Servidor es un sistema distribuido entre múltiples Procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan. La Tecnología Cliente/Servidor, es un modelo que implica productos y servicios enmarcados en el uso de la Tecnología de punta, y que permite la distribución de la información en forma ágil y eficaz a las diversas áreas de una organización (empresa o institución pública o privada), así como también fuera de ella.

Las funciones de servidor a menudo requieren alguna gestión de recursos, en que un servidor sincroniza y gestiona el acceso al recurso, y responde a las peticiones de cliente con datos o información de estado. Los programas de cliente normalmente manejan las interacciones del usuario y con frecuencia solicitan datos o inician alguna modificación de datos en nombre de un usuario.

Por ejemplo, un cliente puede facilitar un formulario en que el usuario (una persona que trabaja en un terminal de entrada de datos, por ejemplo) puede especificar pedidos de un producto. El cliente envía esta información de pedido al servidor, que comprueba la base de datos del producto y realiza las tareas necesarias para la facturación y el envío. Por lo general, varios clientes pueden utilizar un mismo servidor. Por ejemplo, decenas o cientos de clientes pueden interactuar con unos cuantos servidores que controlan el acceso a las bases de datos.



Los procesos clientes son más sencillos que los procesos de los servidores, los primeros no requieren de privilegios de sistemas para funcionar, en cambio los procesos servidores sí.

Los usuarios cuando quieren acceder a un servicio de red, ejecutan un software cliente. El diseño de los servidores debe ser muy cuidadoso, debe incluir código para la manipulación de:

* autenticación: verificar la identidad del cliente.
* seguridad de datos: para que estos no puedan ser accedidos inapropiadamente.
* privacidad: garantizar que la información privada de un usuario, no sea accedida por alguien no autorizado.
* protección: asegurar que las aplicaciones no monopolicen los recursos del sistema.
* autorización: verificar si el cliente tiene acceso al servicio proporcionado por el servidor.

La mayoría de las comunicaciones punto-a-punto en las redes (incluida Internet), están basadas en el modelo Cliente/Servidor. Desde el punto de vista Internet/Intranet, tendríamos:

* Un servidor es un ordenador remoto -- en algún lugar de la red -- que proporciona información según petición.
* Un cliente funciona en su ordenador local, se comunica con el servidor remoto, y pide a éste información.
* El servidor envía la información solicitada.

Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

**Sockets**

Es un "enchufe", es decir, una **conexión** con otro ordenador que nos permitirá **intercambiar** datos. Por así decirlo, cuando quiere conectarse a una página web, el navegador, que es un programa, crea un socket, que contendrá información acerca del servidor, información suficiente para poder realizar una asociación, y de esta manera poder ejercer una conexión. La definición y utilización de los sockets es clara: Conectar dos ordenadores para que puedan comunicarse entre ellos de forma remota o local.

**¿Cómo funcionan?**

Deben existir dos nodos: **Cliente** y **Servidor**. Para que ambos se comuniquen deben de enviarse información entre ellos para determinar con quien van a hablar. Un ejemplo práctico es el correo (no electrónico): Si quiere enviar una carta debe saber el destino donde va a enviar la carta, cuál es la ciudad donde vive el destinatario, su dirección, nombre, etc... Acto seguido, el remitente (cliente) al enviar la carta (mensaje) a una administrador de correo (servidor), leerá los datos y los enviará a su destino (otro cliente). Este es el caso más particular de intercambio de datos que existe. También podemos enviar datos directamente al servidor, y que el servidor nos conteste.

**Tipos**

-TCP: Están preparados para conexiones de tres pasos, en ingles se denomina three way handshake, ya que el cliente para conectarse, le pide permiso al servidor, el servidor acepta o declina (supongamos que acepta), y a partir de ahí comienza el envío y la recepción de datos. Se denomina así porque es un "apretón de manos" en tres pasos.

-UDP: No soportan conexión en tres pasos. Lo que hace el cliente es enviar el paquete con los datos al servidor sin ningún control. Un ejemplo de esta ineficiencia es Skype, el cual utiliza paquetes UDP, y si se pierde uno por el camino no lo podemos recuperar o no llegará al destinatario. Por ese motivo no se recomienda utilizar UDP excepto en casos concretos.

**Sistema operativo de red**

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otros ordenadores a través de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades. El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desee añadir.

El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware. Los Sistemas Operativos de red mas ampliamente usados son: Novell Netware, Personal Netware, LAN Manager, Windows NT Server, UNIX, LANtastic.

**Sistemas distribuidos**

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos ( impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso ) en una sola máquina virtual que el usuario acceda en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual. Todo lo anterior es el marco teórico de lo que se desearía tener como sistema operativo distribuido, pero en la realidad no se ha conseguido crear uno del todo, por la complejidad que suponen: distribuir los procesos en las varias unidades de procesamiento, reintegrar sub-resultados, resolver problemas de concurrencia y paralelismo, recuperarse de fallos de algunos recursos distribuidos y consolidar la protección y seguridad entre los diferentes componentes del sistema y los usuarios.

Los avances tecnológicos en las redes de área local y la creación de microprocesadores de 32 y 64 bits lograron que ordenadores más o menos baratos tuvieran el suficiente poder en forma autónoma para desafiar en cierto grado a los mainframes, y a la vez se dio la posibilidad de intercomunicarlos, sugiriendo la oportunidad de partir procesos muy pesados en cálculo en unidades más pequeñas y distribuirlas en los varios microprocesadores para luego reunir los sub-resultados, creando así una máquina virtual en la red que exceda en poder a un mainframe. El sistema integrador de los microprocesadores que hacen ver a las varias memorias, procesadores, y todos los demás recursos como una sola entidad en forma transparente, se le llama sistema operativo distribuído. Las razones para crear o adoptar sistemas distribuidos se dan por dos razones principales: por necesidad (debido a que los problemas a resolver son inherentemente distribuidos) o porque se desea tener más seguridad y disponibilidad de recursos.

En el primer caso tenemos, por ejemplo, el control de los cajeros automáticos. Ahí no es posible ni eficiente mantener un control centralizado, es más, no existe capacidad de cómputo y de entrada/salida para dar servicio a los millones de operaciones por minuto. En el segundo caso, supóngase que se tienen en una gran empresa varios grupos de trabajo; cada uno necesita almacenar grandes cantidades de información en el disco duro con una alta seguridad y disponibilidad. La solución puede ser que para cada grupo de trabajo se asigne una partición del disco duro en servidores diferentes, de manera que si uno de los servidores falla, no se deje dar el servicio a todos, sino sólo a unos cuantos y, más aún, se podría tener un sistema con discos en espejo (mirror) a través de la red, de manera que si un servidor se cae, el servidor en espejo continúa trabajando y el usuario no se da cuenta de estos fallos, es decir, obtiene acceso a recursos en forma transparente.

Los sistemas distribuidos deben de ser muy seguros, ya que si un componente del sistema falla, otro componente debe de ser capaz de reemplazarlo. Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, etc.

Son sistemas cuyos componentes hardware y software, que están en computadoras conectadas en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo. Se establece la comunicación mediante un protocolo preestablecido.

**Características de los sistemas distribuidos**

* Concurrencia.- Esta característica de los sistemas distribuidos permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que interactúan en la red.
* Carencia de reloj global.- Las coordinaciones para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no tienen una temporización general, está más bien distribuida en los componentes.
* Fallos independientes de los componentes.- Cada componente del sistema pudiera fallar de manera independientemente, y los demás continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continua trabajando.

**Ventajas de los sistemas distribuidos**

Con respecto a Sistemas Centralizados:

* Una de las ventajas de los sistemas distribuidos es la economía, pues es mucho más barato, añadir servidores y clientes cuando se requiere aumentar la potencia de procesamiento.
* El trabajo en conjunto. Por ejemplo: en una fábrica de ensamblado, los robots tienen sus CPUs diferentes y realizan acciones en conjunto, dirigidos por un sistema distribuido.
* Tienen una mayor confiabilidad. Al estar distribuida la carga de trabajo en muchas máquinas la falla de una de ellas no afecta a las demás, el sistema sobrevive como un todo.
* Capacidad de crecimiento incremental. Se puede añadir procesadores al sistema incrementando su potencia en forma gradual según sus necesidades.

Con respecto a PCs Independientes:

* Se pueden compartir recursos, como programas y periféricos, muy costosos. Ejemplo: Impresora Láser, dispositivos de almacenamiento masivo, etc.

1. Al compartir recursos, satisfacen las necesidades de muchos usuarios a la vez. Ejemplo: Sistemas de reservas de aerolíneas.
2. Se logra una mejor comunicación entre las personas. Ejemplo: el correo electrónico.

* Tienen mayor flexibilidad, la carga de trabajo se puede distribuir entre diferentes computadoras.

**Desventajas de los sistemas distribuidos**

El principal problema es el software, el diseño, implantación y uso del software distribuido, pues presenta numerosos inconvenientes. Los principales interrogantes son los siguientes:

* ¿Qué tipo de S. O., lenguaje de programación y aplicaciones son adecuados para estos sistemas?
* ¿Cuánto deben saber los usuarios de la distribución?
* ¿Qué tanto debe hacer el sistema y qué tanto deben hacer los usuarios?

Un aspecto primordial en este tipo de sistemas tiene que ver con las redes de comunicación. Por ejemplo: -Pérdida de mensajes, saturación en el tráfico, etc. Otro problema que puede surgir al compartir datos es la seguridad de los mismos. En general se considera que las ventajas superan a las desventajas, si estas últimas se administran seriamente.

FUENTES:

* <https://www.ecured.cu/Cliente-Servidor>
* <http://nereida.deioc.ull.es/~cleon/doctorado/tic02/cs.html>
* Andrew S. Tanenbaum, SISTEMAS OPERATIVOS MODERNOS