

MEJORAMIENTO DE LA LATENCIA DE LA RED MEDIANTE EL CAMBIO DE TAMAÑO DE BÚFER PARA APLICACIONES FTP UTILIZANDO EL MODELO CLIENTE/SERVIDOR SEGÚN EL TAMAÑO PROMEDIO DE LOS ARCHIVOS A SER TRANSMITIDOS

Juan de Dios Murillo Morera

e-mail: jmurillo@una.ac.cr

Santiago Caamaño Polini

e-mail: scaamano@costarricense.cr

RESUMEN

En el entorno de la informática debemos establecer claramente la relación que existe entre el hardware y el software. Por medio del sistema operativo, podemos comunicarnos con el hardware del sistema y desarrollar aplicaciones. Cuando el usuario programa cierta aplicación en su PC, en la máquina se realizan múltiples procesos y muchas interrupciones o llamadas al sistema se ejecutan para poner a correr la aplicación. A través de mucho tiempo, establecer un mecanismo para optimizar la manera de compartir la información entre los usuarios, de máquina a máquina, ha sido de mucho interés. Cuando se habla de tecnología no se pueden delimitar las situaciones. Hoy, existen múltiples arquitecturas de red para establecer, compartir y almacenar la información, aunque se intenta delimitarlo al modelo cliente/servidor.

En esta investigación se analizarán ciertas características del modelo cliente/Servidor que lo convierten en una excelente opción para generar ganancias en empresas, tanto en tiempo, costos, seguridad y desempeño de todo el sistema y cómo se puede mejorar para obtener un mejor uso de ancho de banda.

ABSTRACT

In the environment of the computer science we must establish clearly the relation that exists between) the hardware and the software. By means of the operating system we can communicate with the hardware of the system and develop applications. When the user programs a certain application in his/her PC, in the machine there multiple processes have to run and many interruptions or calls to the system

execute to make the application to work. Across a lot of time, it has been of great interest to establish a mechanism to optimize the way of sharing the information among the users, from machine to machine. When one speaks about technology the situations cannot be delimited, nowadays multiple architectures of network exist to establish, to share and to store the information though this investigation will try to only treat the client/server model.

In this investigation, certain characteristics of the model analyzed client / server will be analyzed. That they turn it into an excellent option to generate earnings in companies, in time, costs, safety and performance of the whole system and how it is possible to improve it to obtain a better use of bandwidth.

Palabras Clave: latencia, transferencia, archivo modelo cliente/servidor.

Keywords: improving network latency buffer file transfer client/server model

INTRODUCCIÓN

Siempre ha existido la necesidad de efectuar un procesamiento de datos, almacenar información y compartirla. Actualmente, empresas e instituciones luchan por obtener mejores resultados en cuanto a costos, eficiencia, tiempo y calidad, lo cual se traduce en la necesidad de que su personal sea más productivo, que se reduzcan los costos y gastos de operación, al mismo tiempo que se generan productos y servicios más rápidamente y con mejor calidad.

En este contexto, es necesario establecer una infraestructura de procesamiento de información, que cuente con los elementos requeridos para proveer información adecuada, exacta y oportuna en la toma de decisiones y para proporcionar un mejor servicio a los clientes, o sea a toda la red en curso.

Se cuenta con terminales orientados a comandos y mainframes o mini procesadores en los años 60, en los años 80 aparecen las computadoras y redes de área local que cuentan con aplicaciones interactivas, ya para los años 90 se mezclan los *mainframes* junto con los PC constituyendo el modelo cliente servidor tradicional.

Evolución de la arquitectura cliente servidor: La era de la computadora central: Siempre ha existido la necesidad de controlar la información, ya sea de lugares o terminales remotas que se comunicaban con una computadora central, hasta procesos o interconexiones independientes. Se puede señalar distintas eras que poco a poco han ido evolucionado, dejando en claro que hasta hoy no se puede hablar de desarrollo completo, pues es constante el surgimiento de nuevas tecnologías y nuevas arquitecturas.

En la era de la computadora central, como su nombre lo indica, se da cuando las terminales remotas se conectaban a una computadora central. Esta máquina prestaba los servicios a un grupo particular de usuarios.

En la era de las computadoras dedicadas, cada computadora prestaba un servicio al usuario, surge con la aparición de computadoras pequeñas y su facilidad de manejo. En la era de la conexión libre aparecen las computadoras de escritorio donde el usuario puede trabajar tanto aplicaciones de cálculos como en el área de la presentación, estas necesitan de las computadoras de servicio que mediante un *software* pueden emular terminales.

En la era del cómputo a través de redes: Así como lo indica una red, todos los usuarios se encuentran comunicados y pueden intercambiar la información y accederla desde cada PC, porque la información se encuentra almacenada en varias computadoras.

En la era de la arquitectura cliente servidor: En esta arquitectura, la computadora de cada uno de los usuarios, llamada cliente, produce una solicitud de información a cualquiera de las computadoras que almacenan la información, conocidas como servidores, estas dan respuesta a las solicitudes, el cliente las interpreta y las despliega en pantalla.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Debido a la distribución del sistema, cada cliente puede obtener la información cuando lo requiera además de diferentes fuentes y puede manipularla de acuerdo con sus necesidades.

El modelo Cliente/Servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones públicas o privadas y, consecuentemente, desempeña un papel importante en este proceso de evolución.

Sin embargo, es necesario señalar que con la flexibilidad de la arquitectura del modelo, no siempre se escogen las mejores configuraciones para que trabaje de la forma más óptima posible, porque este documento pretende al menos dotar de cierta información que ayude a escoger una mejor configuración en cuanto al tamaño e los búferes para determinados tamaños de archivos según pruebas efectuadas con el menor contacto humano posible para que los resultados tengan una mayor validez.

OBJETIVOS

Objetivo principal

Investigar y analizar la arquitectura del modelo cliente servidor con el enfoque de primero ver sus ventajas y desventajas, para luego cambiar factores en la aplicación del modelo, que permitan elevar la eficiencia de un programa específico.

Objetivos específicos

Investigar en Internet características del modelo cliente servidor que lo hacen ser un modelo con múltiples ventajas de uso con el fin de que el lector inexperto entienda bien el trasfondo de este.

Realizar una aplicación en lenguaje C en el entorno del sistema operativo UNIX que permita transferir archivos con la menor intervención del usuario posible.

Efectuar pruebas con la aplicación previamente confeccionada modificando parámetros de tamaño de búfer tamaño de archivos transmitidos y para luego comparar los resultados y determinar la mejor configuración.

MARCO TEÓRICO

Conceptos

“Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos. Permite integrar una amplia gama de productos y servicios informáticos de manera que pueden ser utilizados eficazmente dentro de la organización” (Entorno Cliente-Servidor, 2009).

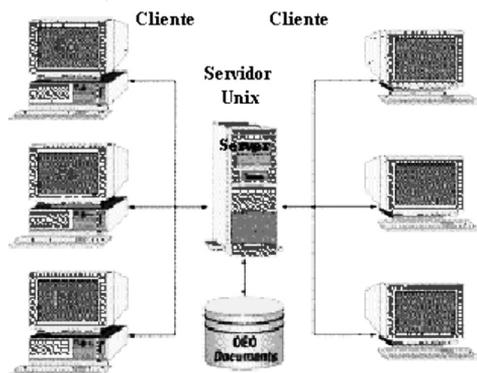


Figura 1. Ejemplo implementación modelo cliente/servidor.

El esquema cliente-servidor “es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se

divide entre dos o más procesos que cooperan entre sí “(Entorno Cliente-Servidor, 2009). Usualmente, la mayoría del trabajo pesado se hace en el proceso llamado servidor y el (los) proceso(s) cliente(s) solo se ocupa de la interacción con el usuario (aunque esto puede variar). Los principales componentes del esquema cliente-servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones.

Es necesario contar con una infraestructura de comunicaciones con propiedades de direccionamiento y transporte para que ambos factores: cliente y el servidor puedan establecer comunicación.

La mayoría de los sistemas cliente/servidor actuales se basan en redes locales y, por lo tanto, utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red requiere habilidad en la administración, transparencia, confiabilidad y desempeño.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplos servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Como ejemplos de servidores pueden citarse servidores de ventanas como X-windows, servidores de archivos como NFS, servidores para el manejo de bases de datos, como los servidores de SQL, servidores de diseño y manufactura asistidos por computador, entre otros.

Características del modelo

El cliente y el servidor se pueden compactar como un solo elemento o pueden trabajar de manera independiente, donde sus tareas y actividades las realizan cada uno por aparte, por lo tanto, se puede hablar de una sola plataforma o varias plataformas.

Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los

servidores, ya sean por actualización o por reemplazo tecnológico. Se realizan de una manera transparente para el usuario final.

El modelo por lo tanto consta de clientes que son, en la arquitectura C/S, los que piden la solicitud para ser atendidos. Sus características son iniciar solicitudes o peticiones; tener un papel activo en la comunicación (dispositivo **maestro** o **amo**). Esperar y recibir las respuestas del servidor. Por lo general, puede conectarse a varios servidores a la vez. Normalmente, interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Además, el servidor es el que recibe la petición. Al iniciarse, esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo **esclavo**). Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente. Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado). No es frecuente que interactúen directamente con los usuarios finales.

De una manera general, el esquema de procedimiento del modelo sería el siguiente: El cliente envía una petición al servidor por medio de su dirección IP, que identifica al segmento de red junto con su puerto, que está reservado para un servicio en particular que se ejecuta en el servidor. El servidor recibe la solicitud y da respuesta con la dirección IP del equipo cliente y su puerto. En la figura 2 se muestra la ilustración.

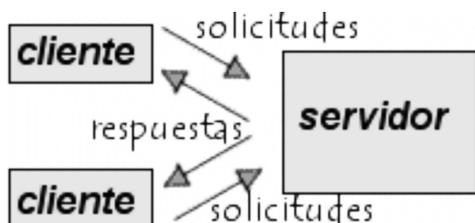


Figura 2. Arquitectura cliente/servidor.

Ventajas del cliente/servidor

El esquema cliente/servidor posibilita la integración entre diferentes sistemas y la compartición de información, permitiendo, por ejemplo, que las máquinas ya existentes puedan ser utilizadas, pero utilizando interfaces más llamativas o fáciles al usuario. De esta manera podemos, por ejemplo, integrar PCs con sistemas medianos y grandes, sin que todas las máquinas tengan que utilizar el mismo sistema.

Al favorecer el uso de interfaces gráficas interactivas, los mecanismos construidos con este esquema tienen una interacción más intuitiva con el usuario. Si se utilizan interfaces gráficas para interactuar con el usuario, el esquema cliente/servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.

Una ventaja adicional del uso del esquema cliente/servidor consiste en que en el desarrollo de aplicaciones se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC).

Su arquitectura de forma modular facilita la incorporación de nuevas tecnologías y el desarrollo de la infraestructura computacional trae como ventaja el término escalabilidad de las soluciones, donde plataformas ya existentes son reutilizadas para generar nuevas plataformas con mayores aplicaciones o habilidades para que el usuario pueda utilizarlas cada vez más apegadas a sus necesidades.

Inconvenientes del modelo

Este modelo adapta distintos elementos de *hardware* y de *software* distribuidos por diferentes proveedores, lo cual dificulta el diagnóstico de fallas, por lo tanto una desventaja es el mantenimiento de los sistemas. Además de lo anterior, se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.

La seguridad de un esquema cliente/servidor es otra preocupación importante. En este caso, los mecanismos son distintos que en el caso de los sistemas centralizados. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor. También, se puede recurrir a otras técnicas como el ciframiento.

Con el establecimiento de estándares, aparecieron los sistemas abiertos. “Un sistema abierto es un medio en el cual se pueden intercambiar componentes de *software* y *hardware*”. Esto permite que el usuario pueda elegir de acuerdo con sus necesidades los productos, motivando a los proveedores a una múltiple competencia, donde deben ofrecer mejores servicios para obtener cada día más clientes.

Un sistema abierto cuenta con las siguientes propiedades: interoperabilidad: Al existir interfaces bien establecidas el gasto en interconexión e integración se minimiza, pues diferentes proveedores pueden compartir información.

Portabilidad: Permite a un sistema instalado en un medio ser instalado en otro minimizando el costo de la migración.

Integración: Permite compartir e intercambiar información mostrando consistencia de comportamiento y presentación.

Los sistemas abiertos son la plataforma adecuada para desarrollo de aplicaciones distribuidas, porque se pueden combinar las ventajas de diferentes máquinas y sistemas operacionales. Para implementar el intercambio de información, el modelo de comunicación más popular es el modelo cliente-servidor, el cual permite que el usuario invoque servicios de forma transparente.

METODOLOGÍA

Con el fin de poder cumplir con los objetivos prácticos, se decide utilizar una mezcla entre código en lenguaje C y las utilidades de los sistemas UNIX, mejor conocidos como *scripts*. De esta manera, se identificaron cuatro módulos principales para realizar las pruebas:

1. Programa de creación de archivos de prueba de un tamaño específico.
2. Servidor de la aplicación.
3. Cliente de la aplicación.
4. *Script* de procesamiento de datos de las pruebas.

Una vez identificados los módulos, es hora de explicar el flujo del trabajo que se seguirá para realizar las pruebas con el menor contacto con el usuario posible.

Primeramente, se utilizará el *script* como maestro de todo el flujo de trabajo. De manera tal que este primero accionará al programa que crea los archivos de prueba para que haga un tamaño específico de archivo y seguidamente, se llamará 10 veces al servidor y al cliente y este recogerá y procesará el tiempo que tarda un búfer en transferirse de un lugar a otro y el tiempo de ejecución del servidor.

El tiempo en ambos casos es calculado mediante el uso del comando de la fecha del sistema con una precisión de horas a nanosegundos. Así, en el caso de la transferencia del búfer, se obtiene la fecha justo antes de que el servidor envíe y luego es dada justo después de que el cliente lo recibe para luego calcular la diferencia. En el caso de la ejecución del servidor, la fecha se obtiene justo antes de empezar a ejecutar el servidor y se obtiene nuevamente después para luego calcular la diferencia de tiempo. Por último, con todos estos datos, el *script* obtendrá los promedios de los tiempos de ejecución y transferencia de búferes, para un tamaño de búfer y archivo específicos.

Después de la parte de procesamiento, se usará el tiempo promedio que tardó un búfer en trasladarse del servidor al cliente para un tamaño de archivo con el fin de obtener la tasa de transmisión (bytes/segundos) para los búferes. Posteriormente, se hará lo mismo pero con el tamaño del archivo contra el tiempo de ejecución del servidor.

ANÁLISIS DE RESULTADOS

A continuación, se presentarán los datos obtenidos mediante el uso del *script*. Sin embargo, es necesario decir que no se pudo probar con tamaños mayores a 10 Megabytes

debido a que las operaciones con punto flotante con scripts son tan lentas y se llegaba a manipular tantos datos, especialmente para los búferes más pequeños (llegaba a tardar más de 10 minutos procesar la información de una sola prueba), que sería necesario tener más tiempo para poder probar con tamaños mayores de archivos.

Pero, los datos obtenidos aún así ofrecen resultados interesantes los cuales deben ser analizados para poder cumplir lo mejor posible con los objetivos de la investigación.

Tabla 1. Tasa de transferencia (velocidad de transmisión) de datos (bytes/s) según el tamaño del archivo y el tiempo de ejecución del servidor.

		Tamaño del archivo (bytes)				
		1K	200K	500K	1M	10M
Tamaño Del Búfer (Bytes)	500	51,14K	115,60K	119,24K	115,21M	116,36M
	800	66,67K	184,58K	194,70K	185,69M	188,21M
	1000	36,66K	238,07K	219,56K	233,99M	229,81M
	9.77K	78,14K	1,89M	2,07M	2,27G	2,01G
	97.66K	79,02K	8,92M	11,50M	15,63G	16,99G
	488.28K	74,18K	9,05M	13,94M	22,76G	44,26G
	1M	44,84K	8,85M	13,92M	24,77G	39,44G
2M	70,73K	8,94M	13,58M	21,85G	53,27G	

Tabla 2. Tasa de transferencia (velocidad de transmisión) de datos (bytes/s) según el tamaño del búfer y el tiempo promedio que tarda en ser transportado.

		Tamaño del archivo (bytes)				
		1K	200K	500K	1M	10M
Tamaño Del Búfer (Bytes)	500	223,32K	232,61K	239,11K	242,03K	244,72K
	800	371,79K	375,32K	394,67K	390,32K	396,22K
	1000	230,16K	477,84K	438,93K	492,29K	481,97K
	9.77K	3,63M	4,24M	4,43M	4,83M	4,17M
	97.66K	39,07M	39,78M	73,04M	158,41M	65,32M
	488.28K	154,20M	205,14M	161,10M	58,54M	30,79M
	1M	181,47M	437,33M	433,41M	417,23M	19,98M
2M	887,93M	858,76M	870,98M	596,79M	41,30M	

Como se puede ver en las tablas, se ha dado los datos en bytes/segundo para hacer más fácil su interpretación.

Analizando los datos, se puede observar que con archivos pequeños, los diferentes tamaños de búferes tienen un desempeño general similar, aunque con búferes más grandes la velocidad de los datos es mayor en cuanto al envío de los bloques de datos entre el servidor y el cliente. También, se puede observar que la velocidad global es menor que la velocidad de transmisión de los búferes.

Además, se observa que conforme los archivos se van haciendo más grandes (1 mega byte o más), aunque los búferes más grandes aparentemente velocidad al transmitir los búferes, se aprecia que con respecto a búferes más pequeños, se empieza a ensanchar la diferencia entre las tasas favoreciendo a los búferes más grandes. Así, se puede apreciar que a nivel global, la velocidad de transmisión excede a la de la transmisión de los búferes.

Todo lo anterior indica que se produce una mejora en el desempeño de la aplicación con búferes grandes, especialmente con archivos mayores en tamaño aunque con archivos pequeños se muestra que los búferes presentan un desempeño similar.

Debe notarse que al revisar cada fila de las tablas, los búferes más pequeños, aunque a nivel global siempre sufren un incremento en la tasa de transmisión que los búferes grandes aunque a menor escala y presentan un comportamiento más estable al transmitir bloques de datos lo que facilita dimensionar el tráfico que se produce en cada envío.

CONCLUSIONES

Primeramente, en el aspecto teórico, el modelo se adapta a múltiples plataformas y redes así como sistemas operativos. Todo funciona al mismo tiempo. Por su capacidad de interoperabilidad, se implementan diferentes estándares como APPC, TCP/IP, OSI, NFS, DRDA corriendo sobre DOS, OS/2, Windows o PC UNIX, en Token Ring, Ethernet, FDDI o medio coaxial, estas entre otras. En cuanto al aspecto práctico, se puede concluir que los datos prueban que un mayor tamaño del búfer puede ser más beneficioso para aplicaciones que requieran transmitir grandes cantidades de información. Sin embargo, es necesario recordar que un búfer más grande implica un mayor uso de la memoria principal por lo que es importante recordar balancear el tamaño del búfer con el consumo de memoria para tener una aplicación óptima de cliente/servidor en el ambiente en que se desenvolverá.

Así, se concluye que los búferes grandes son más costosos de medir según el tipo de tráfico que se quiera transmitir, porque el tráfico de los archivos de prueba era muy estable; sin embargo se produjeron grandes variaciones con estos búferes, por lo que es aconsejable para futuros estudios de optimización que se utilice un método estadístico más allá del utilizado en esta investigación para obtener resultados más certeros y que además permitan explicar mejor el por qué de la mejora del rendimiento global sobre el rendimiento de transmisión de bloques de datos.

BIBLIOGRAFÍA

- Entorno Cliente-Servidor, recuperado el 12 de abril del 2009 de <http://es.kioskea.net/contents/cs/csintro.php3>
- Tecnología Cliente-Servidor, recuperado el 10 de abril del 2009 de <http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/defi.HTM>
- Cliente-Servidor, recuperado el 10 de abril del 2009 de <http://es.wikipedia.org/wiki/Cliente-servidor>
- Valle, José G. y James G. Gutiérrez. (2005), Arquitectura cliente-servidor recuperado el 10 de abril del 2009 de <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#algunos>