

Utilización de arquitectura mixta con clientes delgados en el ámbito educativo

Ing. Cuenca, Fernando; Sanchez Balzaretto, Ignacio J.; Yapura, Gustavo
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

"El avance del software hace que las aplicaciones cada vez requieran equipos con más capacidad. Los laboratorios, sobre todo en el ámbito de la educación pública, no siempre pueden seguir el ritmo debido a los altos costos que implica el recambio tecnológico. Integrando varias tecnologías como booteo remoto y clientes delgados en entornos mixtos donde hay servidores linux y windows, se aborda el problema de no poder aplicaciones nuevas en equipos viejos que ya no pueden utilizarlas dando una solución que permite estirar la vida útil de los equipos y mejorar el servicio a las cátedras."

Palabras Clave

Laboratorio, infraestructura, clientes delgados, booteo remoto, ambiente educativo, X11, RDP, VNC, NFS, SMB.

Introducción

Uno de los principales problemas que se presentan en los gabinetes y laboratorios de computación es la velocidad con la que el equipamiento entra en obsolescencia por el avance del software requerido por parte de las cátedras o cursos. Tradicionalmente se salva este obstáculo renovando periódicamente el parque informático, pero esto posee grandes costos y un impacto ambiental pocas veces tenido en cuenta.

Partiendo de esa problemática, y ante la imposibilidad de afrontar la renovación constante, se planteó la posibilidad de adaptar y combinar diversas tecnologías como la de booteo remoto y clientes delgados para poder dar una solución que permita extender la vida útil de los equipos. Para esto se propuso la utilización de los equipos que ya no pueden ejecutar las aplicaciones requeridas, como meras estaciones bobas también conocidas como clientes delgados.

El booteo remoto es hacer que un equipo cargue su sistema operativo y aplicaciones directamente desde la red sin que intervenga

el almacenamiento propio de la máquina. Para lograr eso generalmente se agrega una eprom en la placa de red que tiene el programa para realizar la transferencia inicial a través de la red.

Los clientes delgado (thin client) son equipos que solo realizan la representación por pantalla, mientras que los programas son ejecutados en uno o más servidores. Estos equipos tienen el software necesario para la representación y algunos más avanzados tienen dispositivos de IO como disquetes, usb, puertos serie y paralelos. Existen varios protocolos de comunicaciones para estos dispositivos pero los más utilizados son X11, VNC y RDP.

Debido a los requerimientos de software se planteó un arquitectura mixta con servidores de aplicaciones Linux y Windows tratando de minimizar los inconvenientes que puede traer a los usuarios y aprovechando para brindar servicios extras.

Elementos del Trabajo y metodología

Cuando se comenzó el trabajo en el laboratorio de sistemas tenía un parque informático con recursos limitados. Las estaciones de trabajo no poseían disco rígido y como sistema operativo se usaba windows 3.11 booteando desde la red desde un servidor novel netware. Las primeras pruebas se realizaron montando un servidor windows NT4 terminal server edition y utilizando el paquete de citrix metaframe. Esto permitió el uso de aplicaciones que antes no podían ser ejecutadas en las estaciones locales. Los altos costos de licencias y compatibilidad con el hardware de las estaciones hicieron que se descartara esta opción y se buscara otra alternativa.

Durante la evaluación de las otras alternativas se decidió probar con clientes delgados utilizando GNU/Linux como base para los mismos ya que es software libre, por lo que permite el acceso al código fuente y la posibilidad de modificarlo para ser adaptarlo a las necesidades del proyecto.

El kernel de linux es altamente configurable y entre las opciones que trae se encuentra la posibilidad de usar el sistema de archivos raíz (root filesystem) via un protocolo de red llamado NFS (network filesystem), también puede ser configurado para que obtenga la configuración de la red vía los protocolos rarp, bootp o dhcp y la posibilidad de tener incluido en forma estática dentro del kernel el driver de la placa de red.

Utilizando como base la distribución de GNU/Linux RedHat 5.2 se procedió a compilar el kernel para que tuviera lo mínimo e indispensable pero que incluyera las tres partes que hacían que el mismo pudiera trabajar como cliente delgado. Luego de varios intentos de configuración se consiguió un kernel que funcionaba según las expectativas.

Se prepara en un equipo la infraestructura de servidor que incluía un servidor NFS y un servidor bootp. Para las primeras pruebas se exporto el sistema de archivos del servidor para ser utilizado en forma remota.

De estas pruebas se observo que los scripts de inicio no estaban preparados para levantar una estación teniendo el sistema de archivos vía NFS y que al iniciarse la estación esta necesita poder escribir ciertos directorios y que estos tienen que ser únicos por cada máquina.

Luego de que se corrigieran los scripts de inicio y que la primera máquina estuviera funcionando se procedió a relizar una implementación pero se seleccionó tanto para los servidores como para las estaciones la distribución RedHat 6.0 ya que poseía mejoras en el soft y mayor soporte de hardware.

Para la infraestructura fueron evaluadas las alternativas de concentrar todos los servicios en un equipo o distribuirlo entre varios. La primera opción tiene la ventaja que reduce el trabajo de mantener multiples equipos pero implica utilizar hardware más potente en el servidor. Utilizar varios equipos tiene la ventaja que es más fácil escalar utilizando hardware barato que si fuera un solo equipo. Como los recursos eran limitados se opto por la segunda opción y reutilizar servidores existentes.

Se armo a un de los servidores para que preste el servicio de archivo, incluyendo esto ultimo los root filesystem de las estaciones remotas y autenticación de usuarios de la red. Para el sistema de archivos vía de red se utilizó NFS. Se exportaron los recursos donde se almacenan los directorios de los usuarios y los directorios que forman la estación remota. Para la autenticación se utilizó (NIS).

El segundo servidor se destino para servidor de aplicaciones. Para proveer el login gráfico se habilito XDMCP (X Display Manager Control Protocol).

Para la elección del entorno gráfico se probaron kde, gnome y wmaker evaluandose el consume de memoria, tiempo de CPU y usabilidad concluyendo con la elección de KDE porque era el que mejor se adaptaba a lo que los alumnos estaban acostumbrados, aunque consumía más recursos que wmaker.

Para el caso particular del laboratorio se respeto el diseño de red anterior por lo que se armaron 2 equipos más para que actúen como routers, servidores de DNS, servidor de bootp y firewall.

Cada estación completa necesita unos 400MB, como el espacio en el servidor de archivos es limitado y no podimos tener la cantidad de copias necesarias para bootear todas las maquina, se comenzó a optimizar el espacio que necesitaba la estaciones. Para lo cual se busco lo mínimo que hacia falta para que la estación arrancara y el resto se

reemplazó por enlaces simbólicos hacia recursos compartidos. En los primeros intentos se logro bajar el consumo de espacio por estación a menos de 25MB y luego eso se lo llevo a apenas 6MB por estación.

Para el mantenimiento de las estaciones y de los usuarios se programaron varios scripts para automatizar el proceso lo que redujo fuertemente el tiempo de administración y también permitió programar tareas de mantenimiento nocturno para que al día siguiente la estructura de las estaciones y alumnos estuvieran sin problemas.

Como todos los usuarios tenían que trabajar sobre la misma máquina en realidad (el servidor de aplicaciones) no se podía usar un usuario único, por lo que a cada máquina se le asocio un usuario genérico propio. Así e usuario para la máquina 1 era alumno1.

Para facilitar las cosas se decidió que los usuarios no tuvieran que ingresar una clava para ingresar. Para evitar que los usuarios usaran otra cuenta que no fuera la que le correspondía a su máquina se implementó la política de limites de concurrencia simultanea en el servidor. Eso quiere decir que no puede haber dos personas usando una sola cuenta desde 2 máquinas distintas.

Las políticas de límites también se utilizaron para acotar el consumo de recursos que podía hacer un usuario del servidor. Los recurso que se limitaron fueron la cantidad de procesos simultáneos, la cantidad de memoria utilizada y el tiempo máximo de ejecución de cada proceso.

Con esas restricciones no se alteraba en uso normal del sistema ya que los valores estaban contemplado para eso. Ante un problema con un programa o una intención del usuario de producir un daño a través de una denegación de servicio el servidor seguiría andando sin perjudicar al resto de los usuarios.

Otra parte importante de la puesta en marcha era asegurar una forma de que siempre un

usuario correspondiera a una máquina. La solución de este problema se realizó mediante un script que se ponía en el proceso de validación del usuario en el programa gdm que es el que provee el login en el entorno gráfico.

Durante el periodo de puesta en marcha surgieron varios problemas más, entre ellos el acceso a la disquera que estaba en la estación desde el servidor. La primera solución consistía en un programa propio que mandaba comando a la disquetera y hacia la transferencia de archivos vía NFS., pero este método resultó ser poco confiable.

La solución a esto fue utilizar floppyd que es parte de las herramientas de mtools que permite acceder remotamente a un dispositivo tipo fat y utiliza además el sistema de autenticación de X11 por lo que se evitan problemas de seguridad

Originalmente las eproms de booteo que estaban en las placas de red no servía para cargar el kernel de linux vía red por lo que al principio se booteaba con disquete. Mientras tanto se busco una solución que fue utilizar el software creado por el proyecto ethetboot y se pudieron armar un cargador que entrara en las eproms viejas y que permitiera cargar así automáticamente el kernel sin depender de algún otro medio.

Para la integración con los servidores windows se opto por utilizar un cliente para el protocolo RDP llamado rdesktop que se conectara directamente a los servidores windows. Como el usuario ya había sido validado se modificó el cliente para que utilizando información del usuario generar una clave para validarse contra el servidor y evitar que el usuario tengo que autenticarse dos veces. También se incremento la seguridad al evitar que dos usuarios entren con la misma cuenta al servidor.

Los resultados no resultaron lo esperado debido al gran consumo de ancho de banda que se presentaba entre el servidor linux y las estaciones cuando se estaba ejecutando el cliente RDP, mientras que entres los

servidores el ancho de banda consumido era mucho menos.

Se detecto que el problema era porque se estaba ejecutando el rdesktop desde el servidor linux y este transmitía los cambios en la pantallas hacia las estaciones como mapas de bits. Para solucionar esto se creo un juego de llaves publicas/privadas y se ejecuto vía ssh el cliente en las estaciones locales. Esto redujo fuertemente el ancho de banda y disminuyo la carga en el servidor de aplicaciones linux.

Para la integración de archivos entre los dos entornos se utilizó el protocolo SMB. En el servidor de archivos se instaló samba y se crearon cuentas para los usuarios con la clave que eran generaba automáticamente para conectarse a los servidores windows.

Aprovechando la estructura del laboratorio se decidió utilizar VNC como herramienta de asistencia a las cátedras para que los usuarios puedan ver lo que hace el profesor. Para esto se instaló un software servidor de VNC en los servidores windows y en los clientes se utilizó lo mismo que para el rdesktop y se lo ejecutaba localmente desde las estaciones.

Superadas las quince conexiones el servidor windows se tornaba demasiado lento y perdía toda utilidad. Se probó lo mismo pero desde el servidor linux y no se presentaba el inconveniente, incluso con más de sesenta conexiones. Basándose en esto ultimo se propuso como solución modificar los scripts de inicio de la sesión de VNC en linux para que en vez de mostrar un escritorio se conecte a los servidores windows utilizando rdesktop.

Resultados

Con la puesta en marcha y mantenimiento del laboratorio utilizando una infraestructura mixta se pudo demostrar que es viable este tipo de infraestructura en ambientes educativos. Pero también se encontró que la implementación y puesta en marcha no está libre de problemas.

La aceptación por parte de las cátedras fue dispar porque si bien ahora se podía ejecutar mucho más software también había algunas limitaciones en el entorno por limitaciones de software o para asegurar la integridad del sistema.

Otro problema es que las aplicaciones que usan gráficos intensivos poseen un rendimiento muy bajo y eso hace que sea imposible que se usen.

Otro beneficio que trae aparejada esta tecnología es la facilidad de administración y de instalación de aplicaciones una vez que esta andando. Mediante el uso de scripts y archivos de configuración se pueden agregar en forma rápida nuevos clientes a la infraestructura y disponen de todo el software instalado.

En general las aplicaciones para windows son más difíciles de hacer funcionar en entornos multiusuarios como se presentan en los servidores terminal server pues las aplicaciones fueron pensadas para estar en entornos monousuario. Muchas requieren privilegios elevados innecesariamente y la falta de documentación de su instalación y requerimientos del sistemas entorpecen la adaptación a entornos multiusuarios.

Discusión

Las tecnologías aplicadas han evolucionado desde el momento de su utilización original a la actualidad. El desarrollo en este area sigue estando signado por la discusión sobre los costos y beneficios de la aplicación de estas tecnologías.

Sin embargo, si no hay recursos económicos para renovar el equipamiento, este tipo de uso es preferible al descarte.

Pero a medida que avanzó la tecnologías asociadas, las aplicaciones también evolucionaron y sus requerimientos de infraestructura son mayores. La brecha entre los servidores y las estaciones de trabajando ya casi no existe por lo que cada día hace más inviable el uso de clientes delgados para las aplicaciones cotidianas.

La aparición de web2.0 está rompiendo nuevamente esta barrera trasladando las aplicaciones al lado del servidor y utilizando los navegadores como clientes delgados pero que necesitan una cierta cantidad de recursos.

Conclusión

Usando las tecnologías antes mencionadas se pudo estirar la vida útil del equipamiento que había disponible y brindar mejores servicios.

Si bien no todas las cátedras pudieron sacar provecho de esto, muchas si pudieron utilizar el equipamiento.

No se puede satisfacer todas las necesidades con estas tecnologías, pero si gran parte de las mismas.

Agradecimientos

Al Ing. Fabian Gibellini, Ing. Hugo Levrino, Marcos Roldan por la ayuda prestada en distintas etapas del proyecto.

Al resto de los operadores de red que mantienen el sistema y lo mejoran.

Al los encargados de aulas y encargados de turno que constantemente asisten sugiriendo ideas y mejoras.

A los docentes y directivos del Dpto. Sistemas por el apoyo en el proyecto.

Referencias:

- [1] Linux XDMCP HOWTO - Thomas Chao y otros
- [2] NFS-Root mini-HOWTO - <http://www.tldp.org/HOWTO/mini/NFS-Root.html>
- [3] The Linux NIS(YP)/NYS/NIS+ HOWTO - Thorsten Kukuk
- [4] The Linux Bootdisk HOWTO - Tom Fawcett
- [5] Root over NFS - Another Approach - George Gousios
- [6] Linux Ethernet-Howto - Paul Gortmaker
- [7] SMB HOWTO - David Wood

Datos de Contacto:

Ignacio J. Sanchez Balzaretti. Laboratorio de Sistemas de la Universidad Tecnológica Nacional - Facultad Regional Córdoba

Maestro Marcelo Lopez, Esquina Cruz Roja Argentina- Ciudad Universitaria – Córdoba- CP 5016.- ignacio@bbs.fr.utn.edu.ar

Gustavo Yapura. Laboratorio de Sistemas de la Universidad Tecnológica Nacional - Facultad Regional Córdoba

Maestro Marcelo Lopez, Esquina Cruz Roja Argentina- Ciudad Universitaria – Córdoba- CP 5016.- gdit@bbs.fr.utn.edu.ar