

The problems of printing in “Server-Based Computing” environments

Luis Bengochea Martínez¹

¹ Departamento de Ciencias de la Computación.
Escuela Técnica Superior de Ingeniería Informática
Universidad de Alcalá.
28871-Alcalá de Henares (Madrid)
luis.bengochea@uah.es

Abstract. Server-Based Computing (SBC), lets you to run native applications installed in remote servers and access them from thin devices such as a personal computer, a PocketPC, a Smartphone or a PDA. The connection TCP/IP with the server could be made through public switched telephone networks, mobile data services GPRS or UMTS, Ethernet, wireless networking, or VPN (virtual private networking). This technology permits the employees of a business to utilize the corporate applications since any location. Nevertheless, one of the problems that presents is the printing of the documents generated by said applications, in the first place by the need to install, in the servers, many printer drivers and also by the great bandwidth required by the printing data flow. In this article, an overview of the technology SBC is offered, showing the specific printing problems in these environments and proposing a solution based on the use of PDF files to replace all the printed exits produced by the applications executed in this way.

Keywords. Ubiquitous office. Server-based computing. Thin client. Free software for ubiquitous computing.

La problemática de la impresión en entornos “Server-Based Computing”

Luis Bengochea Martínez¹

¹ Departamento de Ciencias de la Computación.
Escuela Técnica Superior de Ingeniería Informática
Universidad de Alcalá.
28871-Alcalá de Henares (Madrid)
luis.bengochea@uah.es

Resumen. La computación basada en servidores o “*Server-Based Computing*” (SBC), permite ejecutar aplicaciones nativas instaladas en servidores remotos accediendo a ellas desde un dispositivo que puede ser tanto un ordenador personal, cuanto un PocketPC, un Smartphone o una PDA. La conexión TCP/IP con el servidor puede hacerse mediante la red telefónica fija o móvil GPRS o UMTS, Ethernet, red inalámbrica o VPN. Esta tecnología permite a los empleados de una empresa utilizar las aplicaciones corporativas desde cualquier ubicación. Sin embargo, uno de los problemas que plantea es la impresión de los documentos generados por dichas aplicaciones, en primer lugar por la necesidad de disponer en los servidores de numerosos controladores de impresión y también por el gran ancho de banda que ocupa el flujo de datos de impresión. En este artículo se ofrece una visión panorámica de la tecnología SBC, de los problemas de impresión específicos en estos entornos y se propone una solución basada en el uso de ficheros PDF para sustituir a todas las salidas impresas producidas por las aplicaciones ejecutadas de esta forma.

Términos clave. Oficina ubicua. Computación basada en servidores. Cliente ligero. Herramientas gratuitas para computación ubicua.

1 Introducción

Antes de la irrupción de los ordenadores personales, todas las aplicaciones informáticas se utilizaban desde un terminal físico (VT220, IBM3270, etc.) conectado al ordenador central de la empresa de forma local o desde una ubicación remota mediante el uso de un modem y una línea telefónica.

Cuando se generalizó el uso de ordenadores personales y comenzaron a desarrollarse aplicaciones para ellos, con unos costes de desarrollo mucho menores que los que tenían las aplicaciones en los grandes sistemas o “*mainframes*”, se hizo popular el paradigma cliente-servidor en el que las aplicaciones se dividen en varias capas, la

primera de las cuales, con los niveles de presentación y validación local de datos, se ejecuta en un ordenador personal - “*cliente*” -, y las demás capas de la aplicación lo hacen en el ordenador central - “*servidor*” -, donde a su vez pueden existir varias capas de aplicación, hablando entonces de “*cliente-servidor-servidor*”, etc. Un único programa servidor puede atender a numerosos programas cliente, lo que permitió durante varios años, que las empresas distribuyeran entre sus clientes programas para llevar a cabo diferentes tareas (ver el saldo de una cuenta, realizar un pedido, etc.).

Los grandes problemas que plantea este paradigma, son:

1. La necesidad de distribuir la pieza “cliente” de la aplicación, tanto la primera vez que se vaya a instalar en los ordenadores remotos, como cada vez que se lleven a cabo modificaciones y mejoras.
2. Los diferentes sistemas operativos de los equipos “clientes”, requieren desarrollar programas específicos para cada uno de ellos. Esto complica, además, el proceso de distribución y mantenimiento.
3. Un mismo cliente, puede verse obligado a instalar en su ordenador numerosas aplicaciones que le envían las empresas con las que trabaja, a veces para llevar a cabo tareas idénticas, pero con interfaces y terminología diferentes.
4. La poca experiencia de los usuarios en la instalación y uso de los programas “cliente”, junto con la resolución de las incidencias que se producen, obligan a las empresas a disponer de costosos centros de atención y soporte.

La aparición de Internet y de las tecnologías asociadas de computación, basadas en el uso de servidores de aplicaciones Web a los que se accede desde navegadores universales como Internet Explorer o Netscape, supuso el declive de la tecnología cliente-servidor, ya que venían a dar solución en gran medida a los problemas expuestos anteriormente.

La transformación en Web de las aplicaciones, aparece así como la solución ideal para implementar una oficina ubicua, en la que los empleados – y también los clientes de la empresa, - pueden hacer uso de las aplicaciones corporativas sin importar dónde se encuentren en cada momento - en la oficina, en su casa, en un aeropuerto, - ni qué tipo de terminal estén utilizando – un ordenador de sobremesa, un portátil, una PDA, etc. Así mismo, la aparición de Java y todas las tecnologías asociadas a este lenguaje, ha permitido la rápida transformación de las aplicaciones existentes, propiciando así el amplio despliegue de esta forma de trabajo.

Sin embargo, todavía queda una gran cantidad de aplicaciones usadas en las empresas que, por razones diversas, no pueden ser rediseñadas para aprovechar este nuevo paradigma, como son:

- Aplicaciones heredadas, sobre las que no se dispone del conocimiento o la tecnología necesarios para rediseñarlas
- Aplicaciones adquiridas a fabricantes de software, cuya licencia obliga a usarlas tal como están diseñadas
- Aplicaciones desarrolladas por organismos públicos (como ayudas comunitarias, liquidaciones de impuestos, etc.), de uso obligatorio por las empresas, que deben usarse tal como están concebidas.

- Aplicaciones que debido a su propia naturaleza, no admiten un diseño que permita su ejecución en un servidor Web.

La mayor parte de estas aplicaciones corporativas, con excepción de alguna de las aplicaciones heredadas o las que se ejecutan en un “mainframe”, son aplicaciones con interfaz gráfica de usuario, por lo que no pueden ser accedidas desde un simple emulador de terminal de texto. Además podemos encontrarnos con aplicaciones que se ejecutan bajo sistemas operativos diferentes y en entornos distintos. La solución a este problema, consiste en ejecutar las aplicaciones en su entorno natural y transportar, mediante un protocolo de comunicaciones apropiado, hasta el dispositivo desde el que accede el usuario, las imágenes de pantalla producidas por la aplicación, y llevando hasta el ordenador donde ésta se ejecuta, las pulsaciones en el teclado o los movimientos de ratón que produce el usuario. Esta tecnología, que requiere que los servidores de aplicaciones tengan capacidades multiusuario, se denomina computación basada en servidor o “*server-based computing*” SBC en su terminología anglosajona.

Se trata pues de una solución híbrida entre el uso de aplicaciones web y las tradicionales aplicaciones cliente-servidor, que evita instalar y mantener piezas de software en los dispositivos remotos de los usuarios y facilita un rápido despliegue y un mantenimiento eficaz de las aplicaciones, debido a su carácter centralizado [1].

2 La computación basada en servidores

2.1 Los antecedentes

Como un antecedente del SBC podríamos considerar el sistema X-Windows desarrollado en el MIT en los años 80 (ver [2]). Inicialmente, se desarrolló para ser utilizado desde pequeñas estaciones de trabajo que tenían que acceder a aplicaciones que corrían sobre grandes sistemas Unix. También fue el estándar para acceso remoto a los sistemas operativos OSF y Linux. Al comienzo de esa década se construyeron terminales hardware conocidos como terminales X para ese fin, aunque la popularidad posterior del uso de ordenadores de sobremesa con sistemas Windows, hizo que se sustituyesen por programas de emulación.

El problema de X-Windows es que utiliza un complejo protocolo, con una funcionalidad muy rica, pero que exige un considerable ancho de banda, lo que le hace útil sólo cuando el X-terminal y el servidor de aplicaciones están conectados en red local o en una WAN de alta velocidad.

Además, es el terminal X el que ejecuta un pesado servidor X, que necesita considerables recursos de computación, lo que imposibilita que se utilice un cliente ligero, como puede ser una PDA, que es precisamente una de las premisas de la oficina ubicua. También plantea problemas el modelo de seguridad de X-terminal, ya que el tráfico puede ser analizado y extraídas contraseñas de acceso a recursos críticos de la empresa, por lo que algunos administradores de sistemas prohíben su uso fuera de la zona militarizada de la red.

Otra aproximación al SBC lo constituye el sistema VNC (Virtual Network Computing) [3] desarrollado en el “*Oracle Research Laboratory*” y que está formado por una pieza de software, el “*VNC server*”, que se instala en el servidor – que puede ser un simple equipo de sobremesa con sistema operativo Windows o Linux – y otra pieza, el “*VNC client*”, que se instala en el equipo cliente y que al estar basada en Java, permite que haya versiones para todo tipo de clientes ligeros. La tecnología subyacente, es un protocolo sencillo que actúa a nivel de buffers de tramas y por tanto susceptible de ser utilizado con cualquier sistema operativo y con cualquier tipo de transporte, como TCP/IP (o incluso con USB).

VNC, que tiene además la ventaja de que se distribuye bajo licencia GNU, se utiliza principalmente para implementar soluciones de oficina ubicua en las que el empleado de la empresa puede acceder a su ordenador de sobremesa, situado en la sede corporativa, desde cualquier lugar del mundo donde pueda conectar su equipo ligero. Podría hacerlo incluso desde un cibercafé o desde una consola pública de acceso a internet de las que hay en los aeropuertos. Una vez tomado el control de su ordenador, accedería a las aplicaciones como si estuviese sentado en su puesto de trabajo.

La aparición del sistema operativo Windows NT Server de Microsoft, tuvo un impacto muy significativo en la computación comercial. Numerosas aplicaciones, como sistemas de bases de datos, sistemas de correo electrónico, servidores web y otros, migraron desde los grandes sistemas Unix o VMS a los nuevos servidores con NT. Una pequeña compañía, Citrix, tuvo la idea de aprovechar la capacidad multiusuario de NT, para hacer que el servidor, además de proporcionar servicios a las aplicaciones de red, pudiese abrir varias sesiones de usuario en las que al otro lado se encuentra un usuario con un ordenador en cuya pantalla se visualiza la salida gráfica producida en el servidor que a su vez recibe las pulsaciones de teclado y ratón. El sistema, denominado “*Winframe*” (dando a entender que convertía al servidor en un mainframe de windows) incluía, además, un protocolo optimizado para el transporte de la información de pantalla, teclado y ratón que fluye entre el servidor y el cliente. Se trata del protocolo ICA (“*Independent Computing Architecture*”).

Ha habido otras aproximaciones diferentes al SBC, como el lanzamiento por Sun, en Septiembre de 1999, de un nuevo dispositivo cliente ultra ligero, el “*Sun Ray Enterprise Appliance*”, sin CPU ni memoria. Su misión única consiste en enviar los eventos de teclado y ratón al servidor y recibir de éste los pixels para representar imágenes en la pantalla. Sin embargo, carece de un protocolo de comunicación optimizado, lo que obliga a que su conexión con el servidor tenga que hacerse obligatoriamente en una red Ethernet. Dispositivos de este tipo, pueden ser útiles en entornos en los que se busca que el puesto de trabajo no sea un PC, pero no se pueden considerar como clientes ligeros para implementar una oficina ubicua.

2.1 El SBC en la actualidad

Cuando Microsoft comprendió el alcance que podía tener la solución de SBC que Citrix había desarrollado para su sistema operativo NT Server, adquirió una parte de la tecnología de Citrix usada en Winframe, para desarrollar su propia solución de

SBC, que denominó “Terminal Server”, con un protocolo de comunicaciones llamado RDP.

Actualmente coexisten ambas soluciones, aunque se trata en ambos casos de soluciones propietarias sujetas a licencias de uso con costes. “Terminal Server” de Microsoft, proporciona la capacidad multiusuario de un servidor Windows al que se puede acceder desde ordenadores cliente con sistemas Windows mediante protocolo RDP y que permite a las aplicaciones ejecutadas en el servidor, utilizar los dispositivos locales, como discos e impresoras, del equipo cliente. Para usar “Terminal Server” es necesario adquirir una licencia CAL para cada uno de los clientes ligeros que utilizan el SBC, excepto para los que tengan sistema operativo Windows 2000 o posterior.

Por su parte, Metaframe de Citrix, que, en el caso de servidores Windows, requiere que esté previamente instalado “Terminal Server”, proporciona varios servicios añadidos, como son:

- Permite clientes ligeros no Windows, incluyendo dispositivos Java.
- Se puede usar con servidores UNIX y Linux para proporcionar acceso a aplicaciones desarrolladas en este sistema operativo.
- Desde el servidor se puede acceder a numerosos dispositivos del equipo cliente, como discos, impresoras, puertos serie, USB o tarjeta de audio.
- Proporciona una gestión centralizada de “granjas” formadas por cientos de servidores de aplicaciones, con balanceo de carga.
- El protocolo ICA optimizado para pequeños anchos de banda que permite el uso de comunicaciones de cualquier tipo, desde redes locales a móviles GPRS o 3G.
- Utiliza cifrado RSA (“*Rivest-Shamir-Adleman*”) para proteger la información que se transmite entre el servidor y el cliente, lo que en determinadas aplicaciones puede constituir un requisito imprescindible para su utilización fuera de la empresa.

Existen otras soluciones, también propietarias, como Tarantella (antes Santa Cruz Operation) que está más orientados a proporcionar acceso a servidores Unix y Linux, aunque admite también servidores Windows con “Terminal Server” o “Metaframe” instalados.

3 El reto de la movilidad

Muchas empresas consideran la movilidad de sus empleados como una ventaja competitiva. Hay estudios [1] que muestran que en la actualidad, más de un 10% de los empleados utilizan las aplicaciones corporativas de forma remota durante más de la mitad de su tiempo de trabajo. Por tanto, el reto de los departamentos informáticos consiste en proporcionar buenos sistemas de movilidad, minimizando los problemas de mantenimiento y garantizando la seguridad.

Las tecnologías SBC actuales permiten, en el lado del cliente, minimizar el mantenimiento de los equipos al prescindir de aplicaciones locales, protección antivirus, etc. En los servidores, la instalación y mantenimiento de las aplicaciones se lleva a

cabo de forma centralizada y el acceso a las mismas se hace de forma idéntica por los empleados que se encuentran en el edificio o los que están viajando [4].

Aunque se utilice Internet para acceder a los servidores de la empresa, se puede establecer una VPN antes de comenzar a usar una aplicación, y utilizar los mismos mecanismos de logon y las mismas contraseñas que con el equipo de sobremesa. Los sistemas SBC que cifran la comunicación, proporcionan una confidencialidad añadida. Además, centralizando también las aplicaciones ofimáticas, se evita el riesgo para la seguridad de la empresa que supone la pérdida o robo de un ordenador portátil.

La tecnología SBC aparece así como una solución universal a la movilidad, ya que permite el uso de clientes ligeros y comunicaciones móviles con ancho de banda reducido, y convierte al servidor de terminales en el punto de entrada a todas las aplicaciones de la empresa, ya sean aplicaciones web, aplicaciones cliente-servidor clásicas o aplicaciones específicas sobre sistemas Unix o mainframe, como se muestra en la figura siguiente:

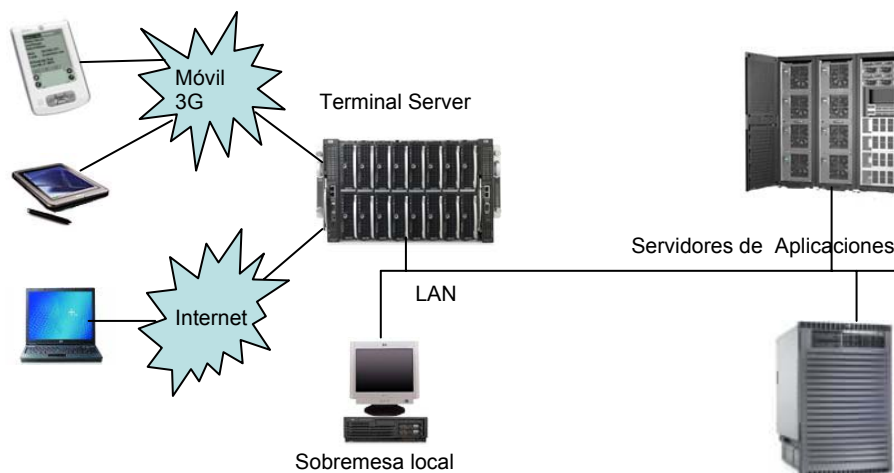


Fig. 1. El servidor de terminales puede actuar como un nivel intermedio entre todas las aplicaciones corporativas, incluidas las basadas en Web, y los dispositivos clientes ligeros.

Lógicamente, cuando se utilizan dispositivos muy ligeros, como puede ser una PDA, con una pantalla de pequeño tamaño y una capacidad de proceso muy limitada, el número de aplicaciones corporativas que son susceptibles de ser utilizadas se reduce. Por una parte, la propia naturaleza de la aplicación puede imposibilitar su uso en este tipo de dispositivos y por otra, la propia percepción que tienen los usuarios de dichos dispositivos, si además está conectado a través de una red celular de baja velocidad y alta latencia, puede desaconsejar su uso en aplicaciones que no estén diseñadas específicamente para ellos. En este sentido, se han llevado a cabo estudios [5] para medir la relación que existe entre el grado de satisfacción de los usuarios móviles dotados de dispositivos ligeros y las métricas típicas de una red de comunicacio-

nes, como ancho de banda y latencia, con el objetivo de determinar dónde se encuentra el límite que permita hacer uso de dispositivos móviles muy ligeros.

4 El problema de la impresión en entornos SBC

Aunque prácticamente todos los sistemas comerciales de SBC proporcionan la capacidad de “mapear” desde el servidor las impresoras locales del equipo cliente, - en el caso de que tenga, - la impresión constituye uno de los puntos débiles de las soluciones a la movilidad proporcionadas por el SBC.

Los problemas surgen, fundamentalmente, por la naturaleza de los procesos de impresión implementados en los distintos sistemas operativos, diseñados bajo la suposición de que las impresoras físicas están conectadas directamente al ordenador donde se está ejecutando la aplicación o, en todo caso, a un servidor de impresoras con el que se dispone de una comunicación con gran ancho de banda.

4.1 Descripción básica del proceso de impresión

Aunque la descripción que se hace a continuación, tanto del proceso de impresión, como de la solución que se propone, corresponden a los sistemas operativos basados en Windows [6], son también de aplicación, con pocas variantes, para otros sistemas operativos diferentes.

Cuando una aplicación debe producir alguna salida impresa, lo primero que hará es llamar a la interfaz de dispositivo gráfico (GDI) que es la responsable de crear la salida visible. Para ello, deberá hacer uso del controlador específico asociado a la impresora elegida para imprimir. Con la información de documento procedente de la aplicación, la interfaz GDI y el controlador se intercambian datos para procesar el trabajo de impresión en el lenguaje de la impresora y, a continuación, lo transfieren a la cola de impresión.

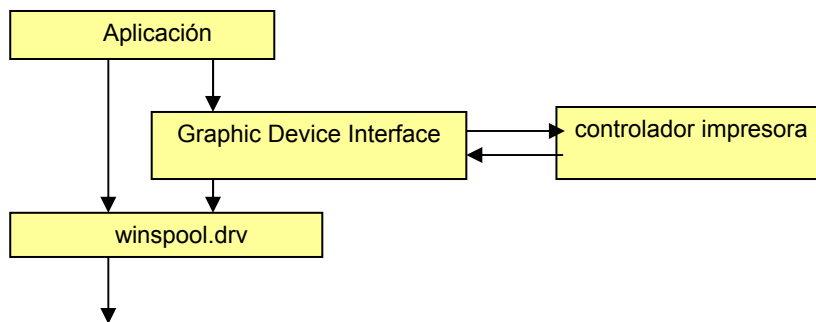


Fig. 2. La primera parte de un proceso de impresión consiste en enviar al “spool” del sistema un fichero con las instrucciones para que la impresora de destino produzca la salida impresa deseada.

El tipo de datos generado por el trabajo de impresión puede ser RAW (al natural) que es directamente interpretable por la impresora, o EMF (meta-fichero mejorado), que permite reducir un poco el tamaño del archivo y mejorar el tráfico entre el spool del sistema donde se produce la salida impresa y el servidor de impresión donde está realmente conectada la impresora, aunque sólo en el caso de que ambos usen Windows. También puede ser ficheros de impresión en lenguaje Postscript (PS) en el caso de que la impresora admita dicho lenguaje.

Dependiendo de si se usa o no servidor de impresión, el spooler de impresión, que es un conjunto de DLLs y controladores que reciben, procesan, planifican y distribuyen los trabajos de impresión, envía el trabajo al proveedor de impresión local o al proveedor de impresión remota (quien lo transferirá al spool del servidor de impresión que tenga la impresora). En el caso de una impresora conectada al cliente ligero y mapeada en el servidor de la aplicación, el archivo es enviado al spool del equipo cliente donde, será procesado por el SO como si se tratase de una impresión local.

Una vez situado el fichero en el spool del sistema que tiene la impresora, es enviado al monitor de impresión (si se utiliza una impresora con comunicación bidireccional, un monitor de lenguaje supervisa la comunicación de dos direcciones entre el remitente y la impresora) quien lo transfiere al puerto (LPT, COM, TCP/IP port, etc.), cuyo monitor a su vez, lo envía a la impresora de destino. Finalmente, la impresora recibe el trabajo, convierte cada página al formato que corresponda y lo imprime.

4.2 Impresión en entornos SBC

De lo expuesto en el apartado anterior, puede observarse que los dos problemas fundamentales que, en relación con la impresión, se plantean a la hora de diseñar una solución de oficina ubicua basada en SBC, son los siguientes:

- Lo servidores de aplicaciones necesitan tener instalados los controladores de todas las impresoras que estén conectadas a los dispositivos clientes. Cada vez que se adquiriera una impresora local para cualquier portátil o laptop de un empleado, habrá que instalar sus controladores en los servidores SBC de la empresa.
- Como los archivos RAW, EMF o PS destinados a las impresoras no están concebidos para ser transmitidos a través de una red de comunicaciones, pueden llegar a tener un tamaño desproporcionado que impida que sea recibido en el equipo cliente en un tiempo razonable. (Ver Tabla 1).

Tabla 1. Tamaño original de un fichero original de prueba y de los ficheros de impresión que se generan según el tipo de impresora y la resolución utilizada:

Formato del fichero	Tamaño
Original (word)	4.052.480 bytes
Fichero RAW en PCL a 600dpi	34.828.288 bytes

Fichero RAW en PCL a 300dpi	15.265.792 bytes
Fichero PS a 600 dpi	9.654.261 bytes
Fichero PS a 300 dpi	9.631.344 bytes
Fichero PDF	2.052.096 bytes

El primer problema puede paliarse con una política centralizada de empresa para adquirir únicamente impresoras que hayan sido previamente homologadas por los administradores del departamento informático que, antes, habrán probado e instalado los controladores de dichas impresoras en los servidores SBC. Sin embargo, la rapidez con que aparecen nuevos modelos de impresoras en el mercado, que van incorporando mejoras y características diferentes, junto con la disparidad de dispositivos móviles a los que se pueden conectar, hace poco realista esta solución.

En las soluciones propietarias de SBC, se proporcionan controladores universales (normalmente PCL) que se pueden utilizar para casi cualquier clase de impresora, pero con el inconveniente de ser excesivamente rígidos.

Para el segundo problema expuesto, la única solución posible pasa por no transmitir archivos RAW de impresión, sino establecer un formato final para todos los impresos, como puede ser PDF (“Portable Document Format”) y transmitir los archivos de impresión en dicho formato para ser impresos localmente en el equipo cliente. También esta solución permite que los empleados con dispositivos móviles sin capacidad de impresión, puedan guardar en disco sus salidas impresas para ser procesadas posteriormente, las manden a la impresora de una delegación de la empresa cercana a su ubicación, las envíen por e-mail a otro empleado o al destinatario del papel, etc.

5 Una solución basada en software libre

Para hacer que cualquier aplicación que se ejecuta en un servidor produzca sus salidas impresas en formato PDF, es necesario definir una impresora lógica (no conectada a ninguna impresora real) que produzca un archivo en formato PDF cada vez que se inicie un trabajo de impresión. Aunque existen soluciones propietarias, como Acrobat Distiller de Adobe, la solución que proponemos se basa en la utilización de dos herramientas de uso libre:

- **RedMon** (“Redirect Monitor”) que nos permite definir puertos cuyo monitor, en lugar de enviar el fichero de impresión a un dispositivo físico, lo redirige como entrada estándar (“stdin”) al programa que le indiquemos. Se distribuye bajo licencia de libre uso, y está disponible en <http://www.cs.wisc.edu/~ghost/redmon/>.
- **Ghostscript**, un intérprete del lenguaje PostScript que permite generar una salida en formato PDF. Se distribuye bajo licencia GNU y está disponible en la dirección <http://www.cs.wisc.edu/~ghost/>.

5.1 Definición de una impresora virtual PDF

Una vez instalados los dos productos antes citados, podemos definir una nueva impresora virtual en cada uno de los servidores SBC, que será la que usen los clientes ligeros del sistema. Para ello seguiremos estos pasos:

- Crear un nuevo puerto (RPTn:) de tipo “redirected port” al que irá conectada la nueva impresora.
- Elegir, de entre todas las impresoras postscript que nos ofrece el SO (porque dispone de los correspondientes controladores), el modelo que mejor se ajuste a las necesidades de impresión, como permitir o no el uso de colores, mayor o menor resolución, etc.
- Hacer la instalación como si dispusiésemos físicamente de una impresora de ese modelo. Definir los parámetros de impresión que se usarán por defecto (tipo de papel, resolución, color, etc.).
- Abrir las propiedades de la impresora y modificar la configuración del puerto RTPn: al que está conectada, usando los siguientes parámetros:

Redirigir el puerto al programa: `C:\gs\gs8.50\bin\gswin32c.exe` (o el lugar donde esté instalado Ghostscript).

Argumentos para el programa: `-q -sDEVICE#pdfwrite -dNOPAUSE -dBATCH -dSAFER -dCompatibilityLevel#1.2 -sOutputFile=%1" -_` (donde “-_” indica que la entrada de datos será por stdín y “%1” que el nombre del fichero de salida será el que el usuario decida cada vez que imprima algo).

Output: Elegiremos la opción “*Prompt for filename*” para obtener el valor con el que sustituir “%1”.

Cuando un usuario remoto abra una sesión en un servidor que tenga definida una impresora PDF de este tipo, podrá usarla como una impresora normal y utilizar los ficheros PDF, así producidos, en la forma que considere más conveniente de acuerdo con el tipo de dispositivo que esté usando y del ancho de banda que tenga la red de comunicaciones a la que está conectado.

6 Conclusiones

La tecnología SBC permite que una empresa establezca una forma única de acceso a todas sus aplicaciones corporativas, incluyendo las de productividad personal, como la ofimática, para hacer que cualquier dispositivo ligero para el que exista un cliente de SBC, pueda ser utilizado desde cualquier lugar y usando cualquier red de comunicaciones.

En el caso de que las aplicaciones produzcan cualquier tipo de salida impresa, se pueden definir impresoras virtuales PDF para ser utilizadas por los clientes ligeros que carecen de impresora y para reducir el tamaño de los ficheros de impresión que se transmiten desde el servidor al cliente, permitiendo además otro tipo de usos para

esos documentos, como enviarlos por correo o guardarlos en disco para su tratamiento posterior.

Referencias

1. Volchkov, A.: "Server-based computing opportunities". IT Professional Volume 4, Issue 2, March-April 2002 Page(s):18 – 23.
2. Revett, M.; Boyd, I.; Stephens, C.: "Network computing: a tutorial review". Electronics & Communication Engineering Journal. Volume 13, Issue 1, Feb 2001 Page(s):5 – 15.
3. Richardson, T.; Stafford-Fraser, Q.; Wood, K.R.; Hopper, A.: "Virtual network computing". Internet Computing, IEEE. Volume 2, Issue 1, Jan.-Feb. 1998 Page(s):33 – 38.
4. McIntosh,D.; Born,R.: "Server-based computing: back to the future solves enterprise desktop problems". Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology, 2000. ICMIT 2000. Volume 2, 12-15 Nov. 2000 Pages:732 - 737.
5. Zhimei Jiang; Mason,H.; Byoung Jo Kim; Shankaranarayanan,N.K.; Henry,P.: "A subjective survey of user experience for data applications for future cellular wireless networks". Symposium on Applications and the Internet, 2001. Proceedings. 2001 8-12 Jan. 2001 Pages:167 - 175.
6. Microsoft Corp.: "Windows 2000 Server Deployment Planning Guide". Capitulo 16, "Deploying Terminal Services". <http://www.microsoft.com/windows2000/techinfo/reskit/dpg>.