

Diseño e implementación de un prototipo multipuesto de trabajo para laboratorios de computación en instituciones educativas

Autor: Licenciado Nicolás Giqueaux

Profesión: Profesor de la Universidad Adventista del Plata.

Fecha: Marzo 2011

email: nicolasgiqueaux@doc.uap.edu.ar

Resumen

Debido al elevado costo de la implementación de un laboratorio de cómputos para las instituciones educativas, se propuso confeccionar un prototipo que permita construir con un CPU 4 puestos de trabajo, en lugar de disponer de un CPU por cada uno. El proceso consistió en los siguientes pasos: desarrollar la simulación por software del proyector para prescindir de la adquisición de uno. Medir el desempeño del prototipo para contrastar con a un laboratorio tradicional. Calcular la economía en costos de adquisición, consumo energético y mantenimiento. Corroborar el funcionamiento de este tipo de laboratorio evaluando la performance del prototipo.

Palabras clave: tecnologías aplicadas a la educación - laboratorio de informática – multitareas – multipuesto - multiseat.

Introducción

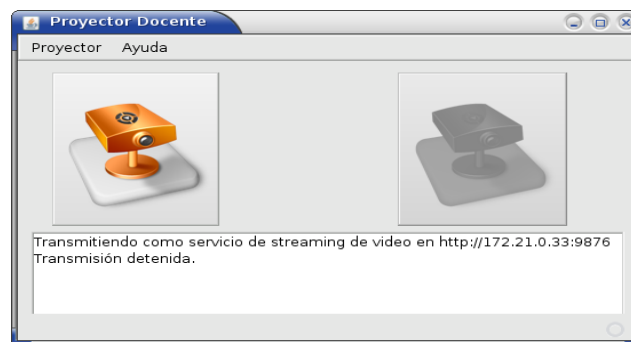
La composición de laboratorios de computación en instituciones educativas cuenta, en promedio, entre 20 y 25 puestos de trabajo. La exigencia de los mismos depende de los objetivos académicos que se dicten. La demanda de recursos de hardware en laboratorios de educación primaria o secundaria no supera las de una oficina administrativa estándar. No así en los laboratorios informáticos de las universidades, especialmente aquellas que tienen carreras de ingeniería y más aún si cuentan con carreras informáticas. Nuestro estudio se concentró en este último y más exigente contexto.

En la actualidad existen puestos de trabajo “pesados” donde se requiere un CPU por usuario; estos son potentes, porque el alumno dispone de todos los recursos del computador de forma exclusiva, pero a la vez costosos. En contraste se diseñaron puestos de trabajo *ligeros*, también conocidos como *livianos*, *delgados* o *thin client* (*Thin Client*, 2010), que carecen de discos rígidos y utilizan una mínima cantidad de memoria. No es necesario tener un CPU por usuario, sólo uno central que administra todos los *thin client*. Cada puesto de trabajo con esta arquitectura es menos potente y costoso pero tiene la característica que todos los procesos, del total de alumnos que trabajan en conjunto, comparten recursos de hardware y software de la central. Esto conlleva el riesgo de quedarse sin recursos de hardware disponibles con la consecuencia de ralentizar al sistema. Un ejemplo de implementación de un *thin client* son las javastation de Sun Microsystem (Dubinski, 2001).

Nuestro estudio buscó un punto intermedio que pretende idealmente acercarse a la potencia de un puesto *pesado* al costo de uno *ligero*. En este sentido se encuentran varias estrategias de implementación. Algunas basadas en software libre, como por ejemplo la desarrollada por el departamento de informática de la Universidad Federal Do Paraná, Curitiba, Brasil, (*Multiseat*, 2011) allí se definió al multipuesto como una computadora personal la cual soporta múltiples usuarios locales en simultáneo. Para ello sólo se necesita un solo CPU y tantos monitores, teclados y ratones como usuarios tenga, con la limitación de la cantidad de puertos USB disponibles. También el equipo de desarrollo de Fedora (una de las más importantes distribuciones GNU/Linux (*DistroWatch.com: Put the fun back computing. Use Linux, BSD*, 2010) está desarrollando e integrando esta funcionalidad para sus productos (*Features/Multiseat*, 2011). Bob Smith (s.f.) documentó su experiencia en la construcción de un sistema Linux de 6-cabezas 6-usuarios y se puede disponer de otras fuentes informales

(*MultiSeat Configuration/Xephyr*, 2011). Existe también este concepto de multipuesto para plataforma Microsoft en implementaciones privativas tales como BetWin de la empresa ThinSoft (*ThinSoft*), ASTER de Ibik-Soft (*Ibik-Soft*) y SofXPand (*SoftXPand*).

Cabe destacar que el siguiente proyecto incorpora el desarrollo de una funcionalidad que le permite al docente proyectar el contenido de su propio monitor a los monitores de los alumnos. Dicho software, que permite el efecto mencionado, está en una versión beta funcional descargable aquí <http://www.inis.uap.edu.ar/ProyectorDocente.7z>. Diseñado para ejecutarse en Linux, Mac y Windows (la versión beta solo corre en Linux). Abajo se puede observar una captura de pantalla de su principal interfaz. Su uso es bien sencillo: se enciende y se apaga la transmisión de la pantalla del profesor y los alumnos se conectarán por medio de VLC (<http://www.videolan.org/vlc/>). Esto permite omitir la adquisición de un proyector dado que los alumnos verán, en sus propios monitores, lo que ve el docente en el suyo. Todos ven lo mismo.



Desarrollo del prototipo

El prototipo consiste en un hardware del mercado, nada fuera de lo común, buscando que sea lo más económico posible, que soporte 4 placas de vídeo y que tenga al menos 6 USB. Éste incluye los siguientes componentes:

- 1 Mother MSI 790FX-GD70
- 4 Placas de vídeo nvidia gforce 9300
- 1 HD de 160GB
- 2 GB RAM DDR3
- 1 microprocesador AMD Athlon X2
- 1 Gabinete con fuente de 450w
- 4 teclados(1 ps/2 y 3 usb)
- 4 ratones (1 ps/2 y 3 usb)
- 4 monitores

Como se puede apreciar, salvo el *motherboard*, no es hardware de alta prestaciones y por lo tanto económico. El costo total fue de \$5331.12 (a marzo de 2010).

La idea es simple: se arma un PC base, luego se le calzan las 4 placas de vídeo. La BIOS sólo considera una placa de vídeo como la principal conjuntamente con el teclado y ratón ps/2. A cada placa de vídeo se le conecta un monitor. Se enchufan los 3 teclados y ratones USB al CPU. Cada conjunto de monitor, teclado y ratón será un puesto de trabajo.



Para el software utilizado se optó por utilizar como sistema operativo de plataforma o anfitrión GNU/Linux Debian versión Lenny (<http://www.debian.org>) por las siguientes razones:

- Es software libre, lo cual produce ahorro en licencias y cuenta con un excelente sistema de repositorios de software.
- Dicha distribución es una de las más robustas (el conocido Ubuntu [<http://www.ubuntu.com/>] está basado en esta distribución).
- Incluye el administrador gráfico Xorg (<http://www.x.org>), el cual tiene un considerable número de parámetros de configuración indispensables para nuestro objetivo.
- Fácil instalación de GDM (<http://projects.gnome.org/gdm/>), quien administrará la sesión y escritorio de cada usuario de forma independiente. También puede usarse XDM o KDM.

Un aspecto interesante es la utilización de máquinas virtuales dentro del prototipo. En nuestro caso hemos utilizado VirtualBox (<http://www.virtualbox.org/>) que virtualizará Windows XP (u otro S.O.). De esta forma se logra que todos o algunos puestos de trabajo, los deseados, operen con Windows XP como sistemas operativos invitados, mientras que el sistema operativo anfitrión, Debian y agregados, gestiona todo lo referente a las terminales o puestos.



Luego de armar el hardware e instalar el software se procede a la configuración del mismo. Una de las primeras modificaciones es ampliar en el kernel (<http://www.kernel.org/>) la cantidad de memoria virtual que utiliza en la paginación interna (*Paginación*, 2011). De no hacer esto el *kernel* no podrá manejar la cantidad de placas de vídeo instaladas. Esto se determina cuando se carga el *kernel* en el momento de arranque del computador, en nuestro prototipo quien se encarga de levantar el kernel es el gestor de arranque GRUB (<http://www.gnu.org/software/grub/>). Se indica que levante el kernel con 256M agregando el siguiente parámetro: `vmalloc=256M`. Para ello modificar el archivo de configuración ubicado normalmente en `/boot/grub/menu.lst`.

Lo siguiente será configurar el administrador de gráficos Xorg (X.Org X Server 1.4.2), editar su archivo de configuración ubicado normalmente en `/etc/X11/xorg.conf` para declarar los cuatro puestos de trabajo que tendrá que administrar. Esta declaración consiste en asociar, entre otros parámetros, la placa de vídeo con el teclado y ratón que definirán a cada puesto. Por ejemplo:

```
# xorg.conf (X.Org X Window System server configuration file)
Section "ServerFlags"
    Option "AutoAddDevices" "False"
EndSection

#SEAT 0
Section "InputDevice"
    Identifier "Keyboard0"
    Driver "evdev"
    Option "XkbModel" "evdev"
    Option "XkbLayout" "es"
    Option "XkbRules" "xorg"
    Option "Device" "/dev/input/event0"
EndSection

Section "InputDevice"
    Identifier "Mouse0"
    Driver "mouse"
    Option "Protocol" "IMPS/2"
    Option "Device" "/dev/input/mouse3"
    Option "ZAxisMapping" "4 5"
    Option "Emulate3Buttons" "yes"
EndSection

Section "Device"
```

```
        Identifier    "Device0"
        Driver        "nvidia"
        BusID         "PCI:1:0:0"
        Option        "AllowGLXWithComposite" "true"
        Option        "RenderAccel" "true"
EndSection
```

```
Section "Monitor"
    Identifier    "Monitor0"
    VendorName    "Unknown"
    ModelName     "Samsung SyncMaster"
    HorizSync     30.0 - 55.0
    VertRefresh   50.0 - 120.0
EndSection
```

```
Section "Screen"
    Identifier    "Screen0"
    Monitor       "Monitor0"
    Device        "Device0"
    Option        "TwinView" "0"
    Option        "metamodes" "1024x768_60 +0+0"
    SubSection    "Display"
        Depth     16
    EndSubSection
    Option        "AddARGBGLXVisuals" "true"
EndSection
```

```
Section "Extensions"
    Option        "Composite" "Enable"
EndSection
```

```
Section "ServerLayout"
    Identifier    "Seat0"
    Screen        "Screen0"
    InputDevice  "Mouse0" "CorePointer"
    InputDevice  "Keyboard0" "CoreKeyboard"
EndSection
```

```
#SEAT 1
Section "InputDevice"
    Identifier    "Keyboard1"
    Driver        "evdev"
    Option        "XkbOptions" "compose:rwin"
    Option        "XkbModel" "pc104"
    Option        "XkbLayout" "es"
    Option        "Device" "/dev/input/event1"
    Option        "XkbRules" "xorg"
    Option        "XkbModel" "evdev"
EndSection
```

```
Section "InputDevice"
    Identifier    "Mouse1"
    Driver        "mouse"
    Option        "Protocol" "auto"
    Option        "Device" "/dev/input/mouse1"
    Option        "ZAxisMapping" "4 5"
    Option        "Emulate3Buttons" "yes"
```

```

EndSection

Section "Device"
    Identifier "Device1"
    Driver "nvidia"
    BusID "PCI:2:0:0"
EndSection

Section "Monitor"
    Identifier "Monitor1"
EndSection
Section "Screen"
    Identifier "Screen1"
    Monitor "Monitor1"
    Device "Device1"
    DefaultDepth 24
    SubSection "Display"
        Modes "1024x768 800x600"
    EndSubSection
EndSection

Section "ServerLayout"
    Identifier "Seat1"
    Screen "Screen1"
    InputDevice "Mouse1" "CorePointer"
    InputDevice "Keyboard1" "CoreKeyboard"
EndSection

#SEAT 2
Section "InputDevice"
    Identifier "Keyboard2"
    Driver "evdev"
    Option "XkbOptions" "compose:rwin"
    Option "XkbModel" "pc104"
    Option "XkbLayout" "es"
    Option "Device" "/dev/input/event4"
    Option "XkbRules" "xorg"
    Option "XkbModel" "evdev"
EndSection

Section "InputDevice"
    Identifier "Mouse2"
    Driver "mouse"
    Option "Protocol" "auto"
    Option "Device" "/dev/input/mouse0"
    Option "ZAxisMapping" "4 5"
    Option "Emulate3Buttons" "yes"
EndSection

Section "Device"
    Identifier "Device2"
    Driver "nvidia"
    BusID "PCI:7:0:0"
EndSection

Section "Monitor"
    Identifier "Monitor2"

```

EndSection

Section "Screen"

Identifier "Screen2"
Monitor "Monitor2"
Device "Device2"
DefaultDepth 24
SubSection "Display"
Modes "1024x768 800x600"
EndSubSection

EndSection

Section "ServerLayout"

Identifier "Seat2"
Screen "Screen2"
InputDevice "Mouse2" "CorePointer"
InputDevice "Keyboard2" "CoreKeyboard"

EndSection

#SEAT 3

Section "InputDevice"

Identifier "Keyboard3"
Driver "evdev"
Option "XkbOptions" "compose:rwin"
Option "XkbModel" "pc104"
Option "XkbLayout" "es"
Option "Device" "/dev/input/event8"
Option "XkbRules" "xorg"
Option "XkbModel" "evdev"

EndSection

Section "InputDevice"

Identifier "Mouse3"
Driver "mouse"
Option "Protocol" "auto"
Option "Device" "/dev/input/mouse2"
Option "ZAxisMapping" "4 5"
Option "Emulate3Buttons" "yes"

EndSection

Section "Device"

Identifier "Device3"
Driver "nvidia"
BusID "PCI:8:0:0"

EndSection

Section "Monitor"

Identifier "Monitor3"

EndSection

Section "Screen"

Identifier "Screen3"
Monitor "Monitor3"
Device "Device3"
DefaultDepth 24
SubSection "Display"
Modes "1024x768 800x600"
EndSubSection

EndSection

Section "ServerLayout"

Identifier "Seat3"
Screen "Screen3"
InputDevice "Mouse3" "CorePointer"
InputDevice "Keyboard3" "CoreKeyboard"

EndSection

Los puntos importantes en este archivo a personalizar, para cada puesto, son las direcciones físicas para las placas de vídeo, teclados y ratones. Las direcciones de las placas de vídeo se obtienen ejecutando el comando mostrado abajo como administrador en una consola, la primera columna indica la dirección. En nuestro prototipo:

```
multiseat:~# lspci | grep VGA  
01:00.0 VGA compatible controller: nVidia Corporation Device 06e1 (rev a1)  
02:00.0 VGA compatible controller: nVidia Corporation Device 06e1 (rev a1)  
07:00.0 VGA compatible controller: nVidia Corporation Device 06e1 (rev a1)  
08:00.0 VGA compatible controller: nVidia Corporation Device 06e1 (rev a1)
```

La información de los teclados se elabora de los archivos que contenga la carpeta `/dev/input/eventN`, donde cada N es el número de evento que genera cada dispositivo. Algunos de estos archivos son para los eventos generados por los teclados, otros corresponden a otros dispositivos. Se infiere cuáles corresponden explorando el contenido del archivo `/proc/bus/input/devices`. Dentro de este se busca las diferentes entradas de información de los dispositivos detectados por el kernel. Especialmente interesa el evento asociado al hardware, línea de *Handlers*, del grupo de información de los teclados. En nuestro prototipo se observan los 4 teclados así:

```
I: Bus=0011 Vendor=0001 Product=0001 Version=ab83  
N: Name="AT Translated Set 2 keyboard"  
P: Phys=isa0060/serio0/input0  
S: Sysfs=/class/input/input0  
U: Uniq=  
H: Handlers=kbd event0  
B: EV=120013  
B: KEY=4 2000000 3803078 f800d001 feffffdf ffefffff ffffffff fffffffe  
B: MSC=10  
B: LED=7
```

```
I: Bus=0003 Vendor=0749 Product=1000 Version=0110  
N: Name="Ever Electronics Corp. Usb KeyBoard"  
P: Phys=usb-0000:00:12.1-1/input0  
S: Sysfs=/class/input/input1  
U: Uniq=  
H: Handlers=kbd event1  
B: EV=120013  
B: KEY=10000 7 ff800000 7ff febefdf f3cfffff ffffffff fffffffe  
B: MSC=10  
B: LED=7
```

```
I: Bus=0003 Vendor=0749 Product=1000 Version=0110  
N: Name="Ever Electronics Corp. Usb KeyBoard"  
P: Phys=usb-0000:00:12.1-3/input0  
S: Sysfs=/class/input/input4  
U: Uniq=
```

```
H: Handlers=kbd event4
B: EV=120013
B: KEY=10000 7 ff800000 7ff febeffdf f3cfffff ffffffff fffffffe
B: MSC=10
B: LED=7
```

```
I: Bus=0003 Vendor=0749 Product=1000 Version=0110
N: Name="Ever Electronics Corp. Usb KeyBoard"
P: Phys=usb-0000:00:13.0-2/input0
S: Sysfs=/class/input/input8
U: Uniq=
H: Handlers=kbd event8
B: EV=120013
B: KEY=10000 7 ff800000 7ff febeffdf f3cfffff ffffffff fffffffe
B: MSC=10
B: LED=7
```

La información de los ratones se consigue de los archivos que contenga la carpeta `/dev/input/mouseN`. Se explora cuáles son detectados desde una consola:

```
multiseat:~# ls /dev/input/mouse*
/dev/input/mouse0 /dev/input/mouse1 /dev/input/mouse2 /dev/input/mouse3
```

Como resultado se asocia la información por puesto de trabajo para formular un cuadro que se utiliza como fuente de información para la configuración de Xorg.

Puesto	Placa de vídeo	Teclado(/dev/input/)	Mouse(/dev/input/)
Seat0	01:00:00	Event0	mouse3
Seat1	02:00:00	Event1	mouse1
Seat2	07:00:00	Event4	mouse0
Seat3	08:00:00	Event8	mouse2

Finalmente se configura el gestor de escritorio GDM para que administre los 4 puestos de manera independiente entre sí. Para ello se debe editar su archivo de configuración ubicado en `/etc/gdm/gdm.conf` asociando cada servidor gráfico con una sesión de usuario que brindará un escritorio de trabajo. Siguiendo con el análisis de nuestro caso veremos:

```
# GDM Custom Configuration file.
# This file is the appropriate place for specifying your customizations to the
# GDM configuration. If you run gdmsetup, it will automatically edit this
# file for you and will cause the daemon and any running GDM GUI programs to
# automatically update with the new configuration. Not all configuration
# options are supported by gdmsetup, so to modify some values it may be
# necessary to modify this file directly by hand.
[daemon]
[security]
[xdmcp]
[gui]
[greeter]
GraphicalTheme=debian-moreblue
[chooser]
[debug]

[servers]
0=Standard0
```

1=Standard1
2=Standard2
3=Standard3

```
[server-Standard0]
name=Standard0
command=/usr/X11R6/bin/X -nolisten tcp -novtswitch -sharevts -layout Seat0 vt5
handled=true
flexible=false
```

```
[server-Standard1]
name=Standard1
command=/usr/X11R6/bin/X -nolisten tcp -novtswitch -sharevts -layout Seat1 vt6
handled=true
flexible=false
```

```
[server-Standard2]
name=Standard2
command=/usr/X11R6/bin/X -nolisten tcp -novtswitch -sharevts -layout Seat2 vt7
handled=true
flexible=false
```

```
[server-Standard3]
name=Standard3
command=/usr/X11R6/bin/X -nolisten tcp -novtswitch -sharevts -layout Seat3 vt8
handled=true
flexible=false
```

Costo de adquisición

Para calcular la diferencia de costo entre armar cuatro puestos utilizando un CPU por puesto (modo tradicional) y un CPU para cuatro puestos (modo multipuesto o *multiseat*), se debe considerar el costo unitario (a marzo de 2010) de los componentes a utilizar, y a partir de estos costos se suma lo necesario para construir cada modelo teórico:

Mother MSI 790fx-gd70	\$1.229,00
Placa vídeo nvidia Gforce 9300	\$232,00
HD Seagate 160GB	\$248,00
Micro AMD Athlon X2 245	\$376,00
2 GB DDR3	\$368,00
Gabinete c/teclado y mouse 450 w	\$213,00
Teclado Genius USB	\$36,00
Mouse Genius USB	\$23,00
Monitor Samsung SyncMaster 943	\$861,00

La suma de lo necesario para un puesto de trabajo tradicional es de \$3.527, esto multiplicado por 4 es \$14.108. La suma de los costos de componentes necesarios para un multipuesto es \$7.008. Esto significa un ahorro de casi un 50% en cuestiones de costos de adquisición de hardware, sin considerar los costos de licencias ahorradas gracias a la implementación de software libre.

La simulación del proyector también acarrea un ahorro importante (adquisición del equipo y mantenimiento de lámpara) al prescindir del mismo. Incluso el alumno dispone del beneficio adjunto de poder grabar vídeo de la presentación de la clase.

Ahorro energético

Se observa un 70% menos de potencia instalada que un laboratorio tradicional. Lo cual significa un ahorro no sólo en la cantidad y/o dimensión de los estabilizadores de tensión o UPS sino que también será verificado reflejado en el consumo eléctrico. Este cálculo se realiza teóricamente a partir de la información del fabricante de hardware: el promedio de potencia eléctrica instalada de los CPU es de 450w y el de monitores LCD es de 30w. Por lo tanto la potencia total instalada para 4 puestos con un CPU por cada uno es de 1920w. Comparando con el total de potencia instalada para 4 puestos con un solo CPU, que es de 570w, se destaca un ahorro energético considerable.

Mantenimiento

El mantenimiento es más simple al verse reducido la cantidad de CPU. Una de las tareas típicas de mantenimiento de los laboratorios informáticos educativos es el clonado de los puestos o instalación de software. Cada vez que se realiza esto sobre el prototipo quedarán listo cuatro puestos de trabajo, por lo que la tarea lleva menos tiempo. Además, al trabajar con Debian conectado a los repositorios de software (*commonalities*), por su solidez y escalabilidad no es necesario tener que clonar cada año los puestos y es muy práctico instalar y sacar programas. En la práctica, al utilizar Debian, los virus informáticos no tienen injerencia generando un ahorro de tiempo y esfuerzos en materia de mantenimiento.

Comportamiento y pruebas

Para el estudio de performance se utilizó la herramienta "ATOP" (<http://www.atoptool.nl/>). Este software recopila información sobre los niveles de ocupación del microprocesador, memoria y acceso a disco, entre otros. Sólo se consideran estas tres variables, utilizando el hardware y software ya descritos, para saber cuán sobrecargado puede estar el CPU cuando los 4 usuarios le demanden tareas. Las pruebas se realizan en diferentes contextos, cada uno con su exigencia característica que se describe a continuación:

- Oficina: se estima que las actividades de una oficina administrativa típica es la de procesamiento de documentos, navegación en internet, correo electrónico, mensajería y algún aplicativo propio.
- Ciber: si bien en un ciber puede darse la exigencia de una oficina, ésta puede ser mayor en los momentos de juegos en red, por la complejidad y demanda de recursos de los mismos. Las experiencias aquí fueron realizadas con juegos.
- Departamento de Desarrollo de Software: además de las exigencias de oficina se debe sumar el consumo de recursos de los entornos de desarrollo, tales como Visual Studio 2010, Netbeans 6, Eclipse, Delphi 2010, etc. Especialmente en los momentos de compilación y ejecución.
- Laboratorio de Carrera informática: similar al punto anterior.
- Virtualizado: en este entorno se ejecutarán 4 Windows XP virtualizados, uno en cada puesto.

Cabe destacar que en las pruebas y mediciones realizadas en los diferentes contextos se considera la peor circunstancia: cuando los cuatro puestos tienen todos los programas abiertos y hacen todo a la vez. Esta demanda de recursos, salvo en los juegos, es poco

probable en simultáneo. Si la prueba es favorable en su peor momento hipotético, de seguro lo será en otros momentos de menor consumo de recursos.

Según el manual de ATOP, los recursos se consideran en ocupación crítica (sobrecarga) cuando: más del 90% del tiempo el microprocesador está ocupado, más del 80% del tiempo del disco es en accesos (lectura/escritura) y más de 90% de la capacidad de la memoria está en uso. Cuando estas variables se sobrecargan se empieza a experimentar lentitud general en la operatoria del sistema.

En la siguiente tabla se observa el desempeño en los diferentes contextos mientras se prueba el prototipo durante 1 hora:

Contexto	Uso promedio del microprocesador	Uso promedio del disco	Uso de la memoria
Oficina	3,00%	16,00%	50,00%
Ciber	93% usando http://torcs.sourceforge.net/ 65% usando http://openarena.ws/smfnews.php	1,00%	55,00%
Depto. Desarrollo y Lab. Informático (en compilación y ejecución con Netbeans 6)	75,00%	10,00%	98,00%
Virtualizado	10,00%	100,00%	98,00%

De esta tabla comparativa se deduce que en contextos de oficina administrativa estándar no hay gran exigencia de recursos. Para el caso de ciber, dependiendo del juego y su complejidad gráfica, podría llegar a haber una sobrecarga del sistema; en los casos estudiados, si bien se hizo un consumo importante del microprocesador en ningún momento se entorpeció el normal desempeño del mismo. En cuanto a departamentos de desarrollo y laboratorios de informática, la performance es buena a pesar de que el sistema trabaja casi al límite; se recomienda aumentar la memoria RAM en otros 2GB. La carencia del prototipo está en el entorno virtualizado con VirtualBox. Si bien se observa que el microprocesador está ocioso, la performance es baja dado que éste está esperando información del disco, el cual está sobrecargado en actividades de lectura/escritura. Esto es debido a que los sistemas operativos disponen de algoritmos que optimizan el acceso a disco, pero dichos algoritmos, del sistema operativo anfitrión, quedan *aislados* a causa de que el *disco rígido* del sistema operativo invitado no es más que un archivo para el sistema operativo anfitrión; en el archivo se concentra toda la actividad de escritura/lectura del sistema operativo invitado y esto produce un cuello de botella. Por otro lado, el rendimiento también se ve afectado por la cantidad de memoria utilizada por parte del entorno virtual.

Conclusiones

Si se extrapola esta experiencia de estrategia de construcción de laboratorios informáticos de entidades educativas a los otros contextos ya descritos, hay que considerar algunas salvedades:

- En oficinas: funcionaría con la limitación de la distancia de los cables, ya que la disposición física de los puestos de trabajo en oficinas no necesariamente están juntos. Esto puede subsanarse utilizando teclados y ratones inalámbricos y extensiones del cable de vídeo, pero aun así la cercanía de los puestos en la oficina es relativa a las distancias permitidas por los dispositivos inalámbricos y a lo que la norma establece como máxima longitud de cable de vídeo.

- En ciber: funcionaría con limitación del audio, indispensable para juegos o telefonía IP. En nuestro prototipo sólo se dispone de una placa de audio (la *onboard*), por lo tanto sólo un puesto de trabajo tiene audio. Esto podría subsanarse agregando mas placas de audio, pero el *mother* debería tener más ranuras o puertos USB (o agregar un HUB USB para ampliar los conectores si hicieran falta).
- En departamento de desarrollo de software: funcionaría con limitaciones similares a las de una oficina y además los entornos de desarrollo deberán correr sobre el sistema operativo anfitrión. Los entornos virtualizados no son recomendados dado que acceden mucho al disco en momentos de compilación y esto provoca un cuello de botella saturando la CPU. Se estima que esto último puede mejorarse utilizando un RAID 0 (<http://es.wikipedia.org/wiki/RAID>) o indicando al entorno virtual que utilice acceso directo al disco y que no simule disco en un archivo.
- En laboratorio de informática: funcionaría como el punto anterior sin las limitaciones de una oficina dado que los puestos pueden estar unos juntos a otros.

La elección del tipo de arquitectura a implementar en laboratorios de informática para instituciones educativas depende del estrés al cual será sometido y a la asignación presupuestaria disponible. Si el presupuesto no es problema y se requiere potencia computacional lo más común es utilizar una estrategia tradicional. En el caso invertido al anterior se utiliza los *thin client* y para situaciones intermedias se recomienda la propuesta aquí presentada.

Bibliografía

Smith, B. (s.f.). *Build a Six-Headed, Six-User Linux System*. Recuperado el 20 de febrero de 2011 desde: <http://www.linuxtoys.org/multiseat/multiseat.html>

DistroWatch.com: Put the fun back computing. Use Linux, BSD (2010). Recuperado el 20 de marzo de 2011 desde: <http://www.distrowatch.com>

[Features/Multiseat](#) (2011). Recuperado el 20 de marzo de 2011 desde: <http://fedoraproject.org/wiki/Features/Multiseat>

Ibik-Soft (s.f.). Recuperado el 20 de marzo de 2011 desde <http://international.ibik-soft.com>

Multiseat (2011). Recuperado el 20 de marzo de 2011, del sitio web del Departamento de Informática de la Universidad Federal Do Paraná: http://wiki.c3sl.ufpr.br/multiseat/index.php/Main_Page

MultiSeat Configuration/Xephyr (2011). Recuperado el 20 de marzo de 2011 desde http://en.wikibooks.org/wiki/Multiterminal_with_Xephyr

Dubinski, R. S. (2001). *What is a JavaStation?* Recuperado el 18 de febrero de 2010 desde: <http://tldp.org/HOWTO/JavaStation-HOWTO/whatischapter.html>

Thin Client (2010). En Enciclopedia Británica online. Recuperado el 18 de febrero de 2010 de <http://www.britannica.com/EBchecked/topic/1516602/thin-client>

ThinSoft (s.f.). Recuperado el 18 de febrero de 2010 de <http://www.thinsoft.com.mx/betwin.html>

SoftXPand (s.f.). Recuperado el 18 de febrero de 2010 de <http://www.miniframe.com/solutions/multi-workstations.html>

Paginación (s.f.). Recuperado el 25 de febrero de 2011 desde: <http://www.win.tue.nl/~aeb/linux/lk/lk-9.html>