

Un escritorio OpenGL acelerado con Xgl y Compiz

# MÁS ALLÁ DE X11

Un miembro del equipo de desarrollo del X11 para Suse revela lo que hay que saber acerca de Xgl. **POR MATTHIAS HOPF**

Los fanáticos de Mac eran escépticos cuando Apple presentó el interfaz gráfico Quartz Extreme [1], que aceleraba los efectos del escritorio utilizando el hardware 3D. Windows Vista de Microsoft, con su tecnología Aero, intenta cubrir esta diferencia con respecto a los Macs. En el mundo de Linux, Xgl [2] proporciona ahora una tecnología comparable e incluso más avanzada que soporta efectos similares.

Xgl es un servidor X de David Revemann que utiliza OpenGL para implementar la salida gráfica. Cuando un programa le pide a Xgl que dibuje una línea, éste pasa los vértices al subsistema OpenGL, que se encarga de enviar los comandos correspondientes al hardware gráfico. A pesar de este énfasis con OpenGL, Xgl también posee el protocolo que las aplicaciones actuales utilizan para comunicarse con el Servidor X, eliminando la necesidad de reescribir las aplicaciones.

## Fondo

Tras un largo período sin evolución en el desarrollo del servidor X, los programadores han integrado características nuevas en los últimos años para proporcionar las bases para la nueva tecnología Xgl. Los kits de herramientas modernos como Qt y

GTK ya utilizan muchas de estas características sin que los usuarios se den cuenta de ello. Dos extensiones principales del protocolo, Render y Composite, juegan un papel importante para Xgl y para el gestor de composición Compiz.

La extensión Render añade nuevas primitivas básicas para mostrar imágenes y polígonos, junto con un nuevo sistema de tipos para mejorar la visualización de las fuentes. Esto refleja el hecho de que los comandos gráficos, denominados *core requests*, ya no tendrán la demanda de los kits de herramientas como Qt y GTK. Todas las primitivas pueden ser enlazadas ahora en el framebuffer usando los operadores Porter-Duff [3], soportando el renderizado de superficies semitransparentes (alpha blending) y fuentes suavizadas (pixel coverage). Muchas aplicaciones modernas hacen bastante uso del suavizado de fuentes.

Hasta ahora, el sistema de ventanas utilizado en las X ha soportado el solapamiento de las ventanas, pero no proporcionaba la capacidad de dibujar en las zonas de las ventanas invisibles y mostrar este contenido. Para conseguirlo, todas las ventanas tienen primero que ser dibujadas en una zona invisible del framebuffer, antes de que se junten las ventanas (composición) en el framebuffer visible. Esto

es lo que hace exactamente la extensión del servidor X Composite.

Un proceso externo se encarga de la tarea de combinar todas las ventanas para proporcionar una vista global de una forma similar a como lo hace el gestor de ventanas. Puede utilizar la extensión Render para dibujar múltiples ventanas semitransparentes y superimpuestas. La composición y la gestión de las ventanas tienen que trabajar codo con codo, en el futuro es de esperar ver más gestores de ventanas con composición con la capacidad de mezclar ambos procesos.

Otro componente importante del servidor X, que necesita desesperadamente ser revisado, es la arquitectura de aceleración hardware, que es la responsable de la representación hardware eficiente de los comandos gráficos. La arquitectura XAA previa está construida en torno a las peticiones del núcleo y por ello es bastante difícil de extender. Dicha arquitectura ha sobrevivido más allá de sus capacidades y necesita ser reemplazada. Las alternativas más prometedoras son EXA y OpenGL.

EXA es una solución rígida y fácil de implementar, pero OpenGL posee la ventaja de ser un interfaz de programación ampliamente soportado por los controladores actuales. No hay necesidad de que el servidor X con-

trole el hardware. En el futuro, habrá solamente un único interfaz hardware para los gráficos, en vez de dos interfaces separados como son XAA/EXA y OpenGL.

## Mirando Más a Fondo

Xgl no acelera la ejecución de los programas OpenGL. Por el contrario, sólo es posible el renderizado indirecto por razones técnicas al tiempo de escribir este artículo. Dicho de otro modo, los comandos OpenGL son manejados por Xgl a través del protocolo GLX antes de ser pasados al hardware gráfico. El renderizado indirecto es mucho más lento que el renderizado directo para los programas que necesitan generar grandes cantidades de polígonos (juegos) o texturas (vídeo).

Por ahora, Xgl no puede acceder de forma nativa al hardware; para ello se vale de un sistema que inicializa el framebuffer y proporciona la interfaz OpenGL. Actualmente se usa el popular servidor X de Xorg; o lo que es lo mismo, Xgl abre una ventana que cubre la pantalla entera en el servidor Xorg. A continuación, las aplicaciones X se podrán conectar a Xgl, mientras que el servidor X estándar tendrá sólo que tratar con Xgl a lo largo de toda la sesión.

Los comandos X11 que el servidor tiene que tratar pueden ser bastante complejos; por este motivo existe un nivel de abstracción para encapsular las sentencias OpenGL. Este nivel, en forma de librería Glitz, es básicamente el acelerador OpenGL de la librería Cairo, un sistema para operaciones gráficas que funciona independientemente de la resolución.

Cuando un gestor de composición entra, se añade más complejidad al canal gráfico. Como se muestra en la Figura 1, el servidor X primero redirecciona todas las salidas de las ventanas a una zona no visible del framebuffer. Una zona de memoria de este tipo se crea por un pBuffer o FBO (Frame Buffer Object). Todos los comandos X11 de una aplicación se redirigen a este espacio de memoria y OpenGL los renderiza. Este proceso ocurre de forma separada para cada programa.

Luego, el gestor de composición dibuja el contenido de la ventana

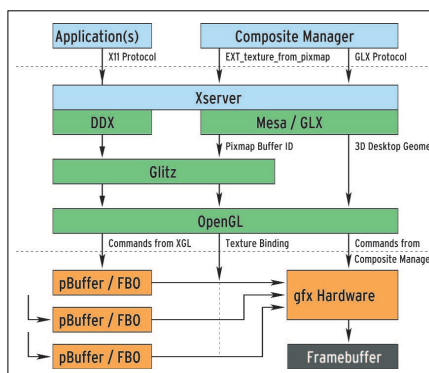


Figura 1: Array de píxeles de un gestor de composición bajo Xgl.

como texturas en los objetos OpenGL. Los objetos son normalmente rectángulos, pero pueden ser más complejos, como objetos tridimensionales para las transiciones.

Así pues, Xgl no es el responsable por sí mismo de los efectos que se comentan tanto últimamente, sin embargo, permite a los programadores crear gestores de composición que puedan hacer uso de los comandos OpenGL para mostrar las ventanas. Esta opción no está abierta para el servidor X normal, ya que OpenGL no está enlazado con el sistema de ventanas subyacente; es decir, no puede acceder al contenido de la ventana dibujada usando los comandos X11.

Como Xgl utiliza OpenGL internamente, se puede hacer accesible el contenido de las ventanas a un gestor de composición externo utilizando la extensión GLX\_EXT\_texture\_from\_pixmap. Esta extensión no está proporcionada por el controlador OpenGL, sino por Xgl. X.org incluye esta extensión desde la implementación de AIGLX, pero aún carece de soporte para algunas características.

Como se mencionó anteriormente, el gestor de composición utiliza el renderizado indirecto para dibujar el escritorio; es decir, todos los comandos OpenGL se envían a Xgl utilizando el protocolo GLX antes de ser pasados al hardware gráfico. Esta es la única forma para que otro proceso utilice las texturas en el rango de direcciones de Xgl. Este problema también afecta a todas las aplicaciones OpenGL, ya que tienen que dibujar en una zona invisible del framebuffer, que también tiene que estar



Figura 2: Compiz cambia a otro escritorio virtual.

en el rango de direcciones del servidor X. Consúltense [4].

## Compiz

Los seres humanos están acostumbrados a comprender los escenarios tridimensionales. Por ello tiene sentido proyectar la GUI en un escritorio tridimensional, suponiendo que se mantienen al mínimo las interacciones con las representaciones de los programas tridimensionales. Las interacciones genuinas en 3D aún tienen algunos problemas técnicos y no son muy intuitivas.

La proyección de los píxeles bidimensionales en los objetos tridimensionales es una tarea típica de OpenGL. Al mismo tiempo, se consiguen efectos como la semitransparencia sin muchos problemas, ya que es uno de los trucos típicos que OpenGL tiene escondido bajo la manga. Por ello, la única razón por la cual no se realiza un gestor de composición de esta clase ha sido que Xgl se necesita para abrir el interfaz OpenGL para los gestores de composición.

Compiz es un gestor de composición diseñado específicamente para el entorno Xgl. En una sesión Compiz no se nota que se esté utilizando OpenGL para generar la salida. Las sombras y las decoraciones de las ventanas con semitransparencias son los únicos efectos que se tienen. Pero cuando se conmuta a otro escritorio virtual, la naturaleza tridimensional del escritorio se vuelve bastante obvia (Figura 2).

Compiz utiliza una arquitectura de plugins muy flexible, pero no del todo estable, para todos sus efectos. Un programa externo renderiza las decoraciones de las ventanas y se las entrega a Compiz. Esto hace que sea

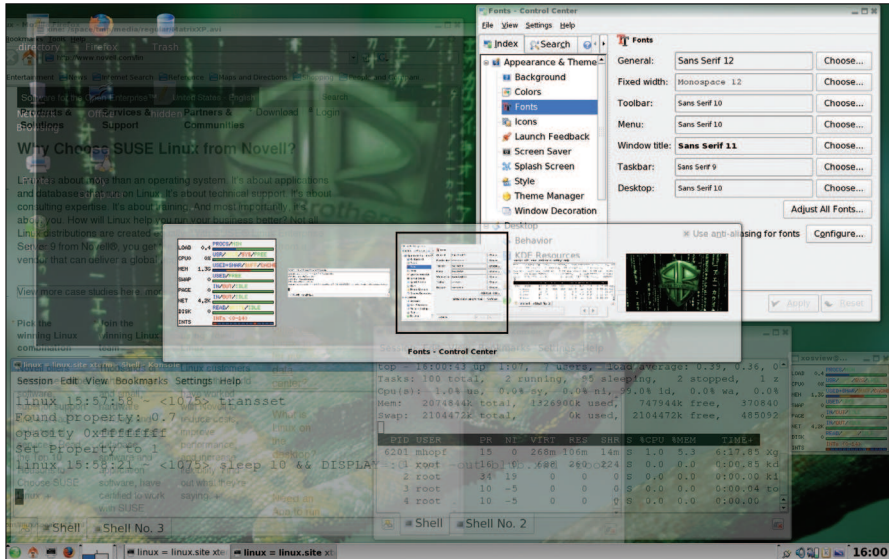


Figura 3: El plugin Switcher con thumbnails con contenido dinámico de aplicaciones.

fácil la integración de temas o de diferentes conjuntos de widgets. Actualmente, hay un programa para maquillar las ventanas para Gnome que también funciona con KDE.

Los plugins de Compiz están disponibles para las funcionalidades básicas del gestor de ventanas (decoración, movimiento, ubicación, cambio de tamaño), para funciones extendidas (cubo, escala, conmutación) y para los efectos (fundido, minimizado, rotación, wobbly, zoom). Como los plugins con funciones equivalentes son intercambiables, los usuarios pueden fácilmente modificar Compiz para ajustarse a sus propios gustos. Véase [6] para una lista más detallada de plugins, junto con las descripciones de las funciones e instrucciones de cómo utilizarlas.

Los programas basados en OpenGL a menudo son criticados por ser simplemente vistosos y nada más. Sin embargo, esta tecnología realmente proporciona un soporte útil para el desarrollo en el campo de la accesibilidad, ayudando a los usuarios con discapacidades (usando un plugin de zoom, por ejemplo) y soporta el desarrollo de ayudas selectivas como Exposé (plugin de escalado), que se ha probado positivamente en el mundo Mac. Los thumbnails para las aplicaciones cuando los usuarios están seleccionando una herramienta (Figura 3). También son muy útiles,

especialmente si son capaces de mostrar la salida actual que está produciendo el programa.

Aunque Compiz se encuentra en el comienzo de su ciclo de vida de desarrollo, es actualmente bastante funcional. Puede que aun no estén disponibles características típicas de los gestores de ventanas actuales y que algunos manejadores de excepciones para tipos de programas específicos no están totalmente implementados, como las miniaturas de las ventanas para los paneles. Es decir, hasta que la integración con KDE no se haya completado, algunas funciones no estarán completadas.

## El Futuro de las X

¿Es Xgl el futuro del sistema de ventanas X? Las opiniones estaban divididas en la última conferencia de desarrolladores de X.org [7] en Santa Clara. Mientras que algunos lamentaban el estado de los controladores de código abierto y la incompatibilidad de los controladores propietarios y GPL, otros consideraban que OpenGL era una interfaz gráfica adecuada y comentaban que les gustaría ver el código completo de los controladores del servidor X. Un servidor X funcionando con la base de OpenGL podría soportar más fácilmente un protocolo futuro que eliminara las primitivas gráficas centrales para mejorar la comunicación entre los clientes y los servidores.

Una de las mayores debilidades de Xgl es su falta de soporte del hardware nativo. Por ahora, está forzado a depender del servidor Xorg como intermediario. Una rama experimental llamada Xegl es capaz de comunicarse directamente con el hardware gráfico, aunque está restringido a las tarjetas Radeon R100 y R200 y el estado actual del servidor es desconocido.

Todos los implicados parecen estar de acuerdo en que el futuro pertenece a los gestores de composición basados en OpenGL. Y Compiz en particular podría ser el gestor de ventanas del futuro gracias a su arquitectura flexible de plugins.

En el momento de escribir este artículo, Xgl y Compiz son bastante estables, pero están aún en una etapa temprana de desarrollo. Los paquetes están disponibles para openSuse [5] permitiéndoles a los usuarios la oportunidad de comprobar la nueva tecnología sin muchos problemas. La versión actual de Ubuntu, Dapper Drake, permite a los usuarios instalar Xgl y Compiz. El sistema es tan estable con algunos controladores que es adecuado para usarse en sistemas de producción. Sin embargo, como Xgl no acepta todas las extensiones OpenGL a sus clientes, muchas aplicaciones OpenGL y juegos aun generan errores en tiempo de ejecución. ■

## RECURSOS

- [1] Apple's Quartz Extreme: <http://www.apple.com/macintosh/features/quartzextreme>
- [2] Xgl Wiki en freedesktop.org: <http://www.freedesktop.org/wiki/Software/Xgl>
- [3] T. Porter & T. Duff, "Composición Digital de Imágenes, Gráficos por Ordenador" Volumen 18, Número 3, Julio de 1984, p. 253-259
- [4] Matthieu Herrb y Matthias Hopf, Nuevas Evoluciones del Sistema de X Window, OpenBSDCon '05: <http://www.openbsd.org/papers/eurobsd2005/herrb-hopf.pdf>
- [5] Xgl Wiki: <http://www.opensuse.org/Xgl>
- [6] Compiz Wiki: <http://www.opensuse.org/compiz>