



Informe técnico

Arquitectura UDA de NVIDIA

Una forma inteligente de reducir
los costes de propiedad



Arquitectura UDA de NVIDIA: una feliz unión entre hardware y software

La arquitectura UDA (Unified Driver Architecture) de NVIDIA® es la base sobre la que se construyen los prestigiosos controladores gráficos que acompañan a las unidades de procesamiento de gráficos (GPU) y los procesadores de plataforma nForce de la compañía. Un diseño exclusivo y la firme apuesta por la estabilidad y el rendimiento del software han erigido a los controladores UDA en la única solución capaz de garantizar la compatibilidad a lo largo de todas las generaciones de productos, ya que un solo controlador soporta todas las versiones existentes del hardware asociado. Cuando aparece una nueva versión de un procesador NVIDIA, el controlador UDA simplifica la integración del nuevo hardware en entornos donde conviven diversos tipos de PC y sistemas operativos. Además, UDA no pone en peligro el rendimiento de los sistemas gráficos existentes o futuros, sino que evita la complejidad normalmente asociada a la actualización de controladores.

El éxito de esta arquitectura sigue siendo uno de los principales factores que citan los fabricantes e integradores de sistemas de todo el mundo como motivo para elegir los sistemas gráficos NVIDIA para sus productos. UDA permite disfrutar de la máxima flexibilidad y simplicidad en los procesos de actualización cuando aparecen nuevos controladores. El excelente historial de compatibilidad que han demostrado estos controladores con todas las versiones de productos se traduce en una notable reducción del coste total de propiedad que beneficia tanto a las grandes compañías como a las pequeñas empresas. Son muchas las características de UDA que contribuyen a esta reducción de los costes:

- ❑ **Disminución del tiempo y los costes de mantenimiento:** una sola versión de los controladores NVIDIA basta para instalar, configurar y administrar una imagen del software válida para todos los sistemas cuando se realizan implantaciones a gran escala.
- ❑ **Protección de la inversión:** NVIDIA dedica constantes esfuerzos a mejorar las funciones y el rendimiento de los controladores. Gracias a ello, los procesadores antiguos pueden aprovechar las mejoras incorporadas y ejecutar las aplicaciones con más rapidez sin costes adicionales y sin crear nuevos problemas a los usuarios o los técnicos de mantenimiento.
- ❑ **Menos problemas de incompatibilidad del hardware y los controladores:** cada controlador se somete a rigurosas pruebas de compatibilidad con todos los productos anteriores y los productos nuevos se prueban incluso con los controladores antiguos. Estas pruebas dan como resultado un funcionamiento más estable durante la vida del producto.
- ❑ **Más escalabilidad:** la actualización y adición de nuevos productos de hardware se simplifica. Los responsables de TI pueden incorporar toda una gama de productos NVIDIA a sus configuraciones utilizando el mismo controlador.

- **Completo soporte multiplataforma:** el 90% del código del controlador puede ser compartido por distintos sistemas operativos, lo que significa que NVIDIA puede proporcionar soporte completo y estable para todas las CPU y todos los sistemas operativos.

En este documento se explica la arquitectura de los controladores tradicionales y los inconvenientes que acarrearán para las empresas. A continuación se presenta la solución aportada por NVIDIA para resolver tales inconvenientes: la arquitectura UDA.

El problema

Para poder integrar el hardware y el sistema operativo (SO), los controladores se suelen escribir utilizando un modelo de capas en el que la capa de abstracción (denominada HAL en inglés) contiene el código que controla directamente el hardware. Los vínculos con el SO constituyen la parte básica del controlador, conocida como la sección “común”, ya que los controladores de todos los dispositivos de hardware necesitan, en mayor o menor medida, este código básico (véase la Figura 1).

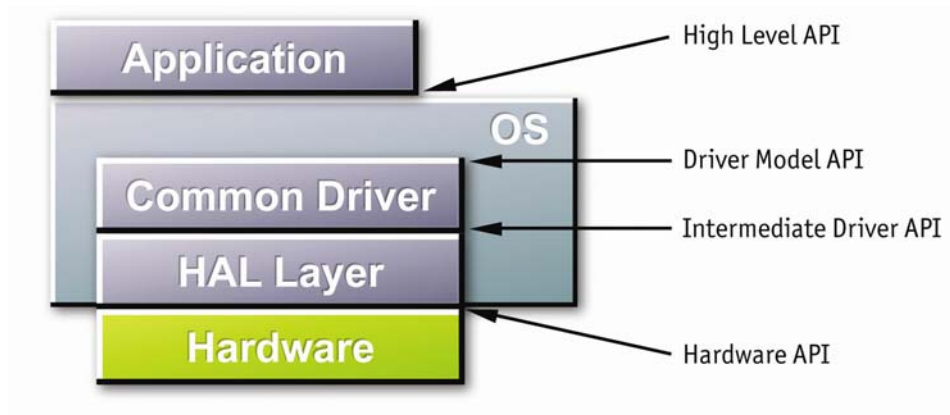


Figura 1. Modelo de controladores tradicional

Problemas técnicos

El tradicional modelo de capas, aunque sencillo, ha ido generando muchos problemas a medida que los sistemas y los entornos crecían en complejidad. Los controladores se ven afectados por tres aspectos de los sistemas de información:

- ❑ **Las versiones de los sistemas operativos**, por ejemplo, los diferentes entornos Windows® de Microsoft® (Windows NT®, Windows 2000, Windows XP, Windows 9x), Linux y Apple® Mac OS X.
- ❑ **Las API de programación**, entre los que se incluyen GDI, VPE, WDM, OpenGL® y Microsoft DirectX®.
- ❑ **Los dispositivos de hardware** soportados por las numerosas y variadas soluciones gráficas de NVIDIA: pasadas, presentes y futuras.

Un solo controlador monolítico crecería de forma desproporcionada y con muy bajo rendimiento si tuviese que soportar todas las opciones disponibles en cada una de estas áreas. Como consecuencia, la arquitectura tradicional ha dado lugar a numerosas versiones de controladores, cada una de ellas dedicada a soportar grupos específicos de SO, API y productos de hardware.

El uso de la arquitectura de controladores y las técnicas de programación tradicionales genera numerosos inconvenientes de tipo técnico:

- ❑ **Periodos más largos de entrada en el mercado**
Todos los controladores deben optimizarse para poder soportar una nueva versión del SO, el API o la solución de hardware. La multiplicación de las combinaciones se traduce en ciclos de pruebas y certificación más largos.
- ❑ **Más costes de soporte**
Las distintas versiones de controladores para los diferentes SO, API y productos de hardware incrementan el tiempo de mantenimiento necesario para administrar combinaciones de hardware y software que resulten estables.
- ❑ **Menos rendimiento**
Se vuelve prácticamente imposible ajustar todas las variantes de ejecución cuando intervienen tantos controladores, pero utilizar un solo controlador de gran tamaño degrada aún más el rendimiento.
- ❑ **Escasa escalabilidad**
Un enfoque de programación convencional no ofrece la escalabilidad necesaria porque el hardware y el software cada vez son más complejos. Un solo controlador aumentaría exponencialmente la complejidad y no sería posible mantener la calidad y el rendimiento.

Problemas empresariales

Un análisis de la arquitectura tradicional desde el punto de vista empresarial plantea otra serie de inconvenientes:

- ❑ **Costes de desarrollo de software**
El uso del modelo tradicional exige la intervención de un elevado número de técnicos para programar y corregir los diferentes controladores. Al final, estos costes repercuten en los clientes.
- ❑ **Complejidad en la fase de publicación de la versión**
La fase de control de calidad es larga y compleja. Además, para empeorar las cosas, los esfuerzos de control de calidad se duplican en cada parte de la cadena de suministro: el proveedor del controlador, los fabricantes o integradores de sistemas y las grandes empresas que homologan sus equipos antes de implantarlos.
- ❑ **Procesos de certificación lentos y costosos**
Es preciso llevar a cabo el proceso de certificación por cada combinación de controlador-hardware, y por cada sistema que integra el controlador. Los controladores únicos de gran tamaño suelen llevar asociados numerosos procesos de comprobación y ciclos de certificación más largos.

La solución de NVIDIA

Los desarrolladores de NVIDIA se marcaron como objetivo diseñar e implementar una arquitectura de controladores que minimizase el coste total de propiedad de sus procesadores, maximizase el rendimiento y proporcionase total estabilidad y compatibilidad en toda la gama de soluciones gráficas de la compañía. El resultado fue la arquitectura UDA (Unified Driver Architecture), que responde a todos esos objetivos y se coloca como el principal logro de estas características en el sector (véase la Figura 2).

Con el modelo UDA, el software de controladores de NVIDIA se centra en implementar las funciones de las API en lugar de identificar las diferencias del hardware. Todas las operaciones de control del hardware se manejan a través del modelo de programación orientada a objetos basada en clases. Las funciones más sensibles al rendimiento (el procesamiento de gráficos y audio) se apoyan en una capa de abstracción HAL implementada en el hardware que reside en los chips NVIDIA. Esta implementación de funciones en el hardware reduce el exceso de código asociado a los controladores tradicionales. La combinación de hardware/software equilibra el rendimiento y la calidad de forma insuperable y ofrece unos resultados que no pueden obtenerse con una solución basada exclusivamente en el software.

Todos los controladores NVIDIA proporcionan un rendimiento excepcional y, al mismo tiempo, total compatibilidad con todas las generaciones de productos: pasadas, presentes y futuras. Un solo controlador soporta todas las versiones del hardware asociado utilizando técnicas de programación basadas en clases y un alto nivel de abstracción que independiza el software del hardware de nivel básico.

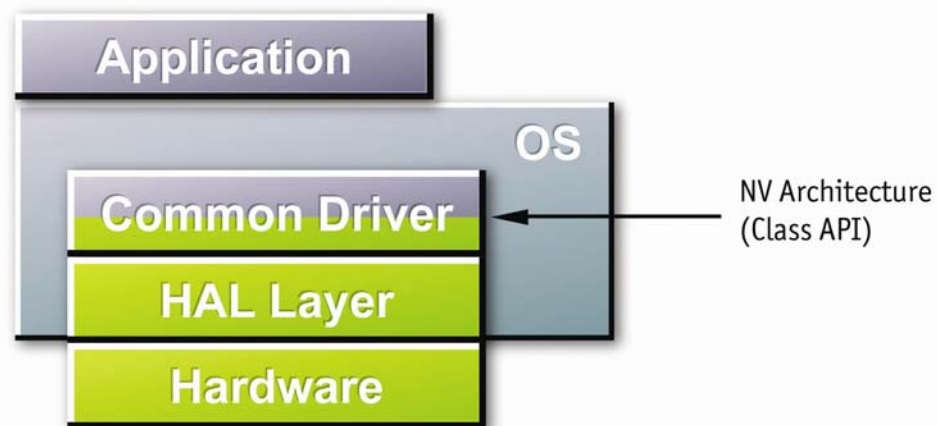


Figura 2. Modelo de controladores de NVIDIA para GPU y funciones sensibles al rendimiento

Programación orientada a objetos

El modelo UDA incluye una abstracción orientada a objetos de la funcionalidad del hardware, también denominada modelo basado en clases. *Objetos* y *clases* proporcionan mecanismos para controlar con eficacia el acceso a la funcionalidad y la información de contextos. Al abstraer la funcionalidad del motor, la arquitectura de NVIDIA mantiene la independencia del motor al tiempo que reduce la carga de trabajo de la CPU, la memoria y el bus durante el procesamiento de las funciones gráficas y del sistema.

Nota: Una **clase** es la definición abstracta de una determinada función de hardware (que se implementa como mapa de registros). Un **objeto** es una instancia de una clase (una correspondencia específica del mapa de registros) y, por tanto, contiene un determinado contexto. Un **método** es un registro de sólo escritura dentro de un objeto.

La estructura de clases y los *métodos* de UDA reproducen los modelos y el uso de las API existentes. Por ejemplo, el controlador de GPU proporciona clases diseñadas para coincidir exactamente con las últimas interfaces de programación de Windows.

Administrador de recursos

Dentro de la capa de hardware de UDA, el Administrador de recursos (Resource Manager) ofrece una forma inteligente de reconocer y gestionar las clases y los objetos (véase la Figura 3). El software de controladores de NVIDIA proporciona una librería en el nivel del kernel que se encarga de reconocer y administrar clases y objetos. La novedosa solución de NVIDIA se ocupa de gestionar los estados y contextos asignando todos los recursos de la arquitectura según se van necesitando para cada controlador cliente. Tenga presente que, en este modelo de administración de contextos, los diferentes clientes no saben que hay otros usuarios utilizando el chip en ese momento.

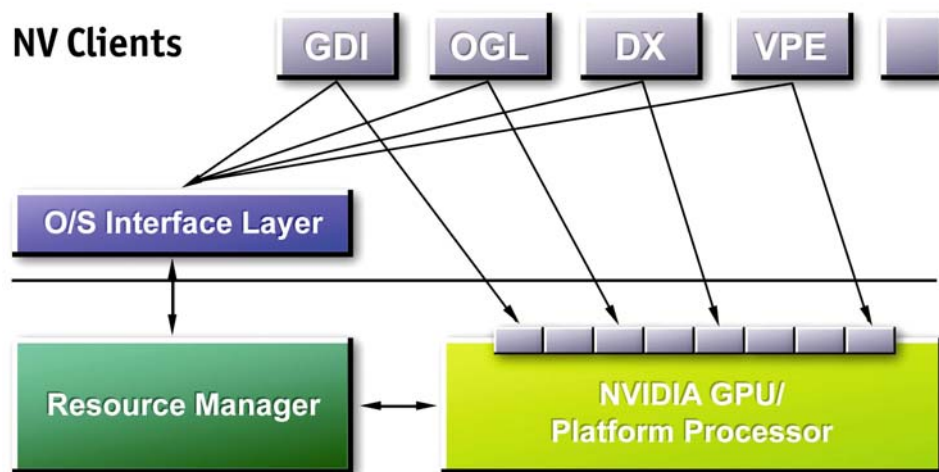


Figura 3. Arquitectura de controladores con el Administrador de recursos

La asignación y gestión de los *objetos* de la arquitectura (recursos de hardware) abre la posibilidad de manejar la compatibilidad de varias formas. Como se ha indicado anteriormente, el Administrador de recursos establece un canal entre el hardware y un controlador cliente de forma que elimina el exceso de controladores necesarios para manejar los recursos de hardware correspondientes.

Dado que reconoce las clases designadas, el Administrador puede manejar casos de excepción del hardware. De esta forma, las diferencias en las implementaciones del hardware se mantienen transparentes para el usuario y se gestionan de tal manera que posibilitan la compatibilidad de los controladores en todas las plataformas. Además, esta gestión de recursos permite utilizar interfaces de clases ya existentes para mantener la compatibilidad entre diferentes versiones del hardware. Los ingenieros de las aplicaciones establecen la interacción con todos los productos de hardware utilizando estas convenciones estándar, lo que significa que no es preciso cambiar el código para soportar los nuevos productos.

Otras funciones controladas por el Administrador de recursos incluyen:

❑ **Gestión de errores**

Las funciones de reconocimiento de clases constituyen un excelente mecanismo para implementar correcciones de errores en los controladores.

❑ **Partes de la arquitectura de dependen del SO**

El Administrador de recursos también puede utilizarse para manejar las diferencias existentes entre varios SO en la gestión de la memoria, el control de procesos/ subprocesos, las funciones *plug-n-play*, la gestión de la energía y otras funciones.

Ventajas de UDA:

La arquitectura UDA se ha venido utilizando durante varias generaciones de tecnologías y productos de NVIDIA desde que en 1998 apareciese el controlador NVIDIA RIVA128™. Los integradores de sistemas y usuarios que utilizan esta arquitectura afirman disfrutar de las siguientes ventajas:

❑ **Mejor diseño con más funciones integradas**

Dado que no son necesarios tantos ingenieros para controlar y gestionar las versiones de los controladores, se pueden dedicar más recursos humanos a la optimización e incorporación de nuevas funciones. Además, la solución UDA también concede más libertad para cambiar el hardware de bajo nivel sin que ello afecte a los fabricantes de sistemas y los usuarios.

❑ **Rendimiento excelente**

La integración de las funciones del controlador en el hardware garantiza el mejor rendimiento posible en todas las aplicaciones.

❑ **Estabilidad y fiabilidad**

UDA ha superado con éxito la prueba del tiempo y ha sido determinante para consolidar la reputación de NVIDIA como proveedor de productos de software y hardware caracterizados por su fiabilidad y estabilidad.

❑ **Plazos de llegada al mercado extremadamente cortos**

Las nuevas versiones de NVIDIA se desarrollan en tiempos récord: menos de 100 días desde el inicio del diseño a la fase de producción, plazo que se prolonga a más de seis meses en las demás empresas del sector. Los fabricantes de sistemas acceden rápidamente a la nueva tecnología y la instalación y soporte de nuevas versiones es sumamente sencilla, ya que un solo controlador soporta las versiones existentes y futuras.

❑ **Menor coste total de propiedad**

Los responsables de TI pueden generar una sola imagen del software con una versión del controlador NVIDIA e implantarla en miles de sistemas con diferentes procesadores NVIDIA. Esto les permite homologar, administrar y realizar el mantenimiento de una sola arquitectura de software en múltiples configuraciones de hardware.

Conclusiones

La arquitectura UDA de NVIDIA ha establecido un nuevo modelo de diseño de controladores capaz de proporcionar rendimiento, compatibilidad y una importante reducción del coste total de propiedad. Pero UDA es más que un diseño inteligente. Sus ventajas tienen un impacto positivo en todos aquellos que eligen la tecnología NVIDIA.

- ❑ **Las empresas** se benefician de un notable ahorro de tiempo y costes de TI gracias a la simplificación de los procedimientos de actualización y soporte.
- ❑ **Los responsables de sistemas** pueden manejar distintos productos de hardware con los mismos controladores, migrar a nuevas plataformas sin cambiar los controladores, reducir los tiempos de implantación de nuevos sistemas y dar soporte a entornos de hardware heterogéneos con diversas configuraciones.
- ❑ **Los usuarios** pueden acceder rápidamente a nuevos productos y ejecutar los juegos y las aplicaciones existentes en las plataformas de última generación.
- ❑ **Los desarrolladores de software** pueden escribir una versión de una aplicación optimizada para todas las soluciones NVIDIA.
- ❑ **Los fabricantes de sistemas** ofrecen a sus clientes la posibilidad de generar una sola imagen de software que soporta toda una línea de hardware, lo que simplifica la incorporación de nuevas plataformas en la instalación existente, agiliza los ciclos de homologación y certificación, y reduce considerablemente el coste total de propiedad.

Las soluciones NVIDIA seguirán utilizando UDA como fórmula para proporcionar controladores de software de alta calidad, alto rendimiento y alta estabilidad. UDA resuelve las carencias asociadas a las arquitecturas de controladores tradicionales y permite introducir innovaciones en el hardware de forma rápida y sin perjudicar la compatibilidad con los sistemas y aplicaciones existentes. Ningún otro fabricante de soluciones gráficas es capaz de ofrecer una sola imagen de software para toda la gama de productos de una compañía.



Aviso legal

TODAS LAS ESPECIFICACIONES DE DISEÑO DE NVIDIA, PLACAS DE REFERENCIA, ARCHIVOS, DIBUJOS, DIAGNÓSTICOS, LISTAS Y OTROS DOCUMENTOS (DENOMINADOS CONJUNTAMENTE O POR SEPARADO "MATERIALES") SE ENTREGAN "TAL CUAL". NVIDIA NO OFRECE NINGUNA GARANTÍA EXPRESA, IMPLÍCITA, ESTATUTARIA O DE OTRA NATURALEZA CON RESPECTO A LOS MATERIALES Y RECHAZA EXPRESAMENTE CUALQUIER GARANTÍA IMPLÍCITA DE COMERCIABILIDAD, NO INFRACCIÓN O ADECUACIÓN A ALGÚN PROPÓSITO EN PARTICULAR.

NVIDIA Corporation considera que la información suministrada es exacta y fiable, pero no asume responsabilidad alguna por las posibles consecuencias o infracciones de derechos sobre patentes, u otros derechos de terceras partes, que pudieran derivarse de su uso. NVIDIA no otorga licencia alguna por implicación, ni de ningún otro modo, bajo ninguna patente o derecho de patente de NVIDIA Corporation. Las especificaciones mencionadas en esta publicación son susceptibles de cambios sin previo aviso. El contenido de este documento sustituye y prevalece sobre cualquier otra información anteriormente suministrada por NVIDIA. No se autoriza el uso de los productos de NVIDIA Corporation como componentes esenciales de dispositivos o sistemas de apoyo o sostenimiento de la vida sin el permiso previo y por escrito de NVIDIA Corporation.

NVIDIA y el logotipo de NVIDIA son marcas registradas y RIVA128 es una marca comercial de NVIDIA Corporation.

Los restantes nombres de empresas y productos pueden ser marcas comerciales y/o registradas de sus respectivos propietarios.

Copyright de NVIDIA Corporation 2003



NVIDIA.
NVIDIA Corporation

<http://es.nvidia.com>