



Condensé technique

L'architecture de pilote unifiée de NVIDIA

Une solution racée pour
réduire les frais de propriété
globaux



L'architecture de pilote unifiée NVIDIA : une union matériel-logiciels efficace

L'architecture de pilote unifiée (UDA) de NVIDIA® est la base des très primés pilotes graphiques des processeurs graphiques (GPU) du constructeur et des processeurs de ses chipsets nForce. La spécificité de sa conception et les efforts constamment déployés par l'entreprise pour en assurer la stabilité logicielle et la performance font de l'UDA la seule architecture de pilote du marché à offrir une compatibilité parfaite avec les produits anciens et futurs : un même pilote prend en charge toutes les versions (précédentes et à venir) du processeur correspondant. Lorsque NVIDIA sort une nouvelle version d'un processeur, le pilote NVIDIA existant en simplifie l'intégration, ou celle d'un nouvel ordinateur, dans les environnements comportant de nombreux ordinateurs de bureau et systèmes d'exploitation différents. Ne se contentant pas de ne pas mettre en danger les performances des processeurs graphiques ou des chipsets anciens et nouveaux, l'architecture de pilote unifiée de NVIDIA réduit aussi les coûts liés à la mise à jour des pilotes.

Le succès de l'architecture de pilote unifiée de NVIDIA est tel que les OEM et les revendeurs de systèmes du monde entier la considèrent comme un élément fondamental au moment de choisir processeurs graphiques et jeux de puces. Avec l'UDA, les utilisateurs finals apprécient la simplicité des mises à jour au moment de la sortie de nouveaux pilotes. Par ailleurs, les excellents résultats obtenus dans le domaine de la compatibilité se traduisent par une réduction significative des frais de propriété globaux et, donc, par des économies qui peuvent être répercutées sur la clientèle de l'entreprise, quelle que soit sa taille. De nombreuses caractéristiques de l'UDA contribuent à la réduction des coûts :

- ❑ **La diminution de la durée et du coût des interventions de maintenance :** en ayant une unique version des pilotes UDA, vous n'avez qu'une image logicielle à gérer, configurer et installer pour les déploiements d'ordinateurs à grande échelle.
- ❑ **La protection de l'investissement :** NVIDIA ajoute sans cesse de nouvelles fonctionnalités à ses pilotes et en améliore les performances. Les anciens processeurs peuvent ainsi bénéficier de ces améliorations et tourner plus rapidement sans aucun coût supplémentaire et sans compliquer la vie de l'utilisateur final ou du personnel de maintenance.
- ❑ **La réduction des conflits matériel/pilote :** chaque pilote est rigoureusement testé avec tous les produits précédents et chaque nouveau produit est testé avec les anciens pilotes. Le résultat est une meilleure stabilité de la solution pendant toute sa durée de vie.
- ❑ **Une évolutivité accrue :** la mise à jour des produits et l'ajout de nouveau matériel sont considérablement facilités. Les responsables informatiques peuvent

ajouter à leurs configurations une gamme complète de produits NVIDIA, qui utilisent tous les mêmes pilotes.

- **Une prise en charge multi-plate-forme supérieure :** puisque 90% du code du pilote NVIDIA peut être partagé par différents systèmes d'exploitation, NVIDIA peut offrir une assistance polyvalente, haute performance et d'une stabilité à toute épreuve pour toutes les unités centrales (UC) et tous les systèmes d'exploitation.

Ce document présente les architectures de pilote traditionnelles, explique quels sont les obstacles techniques et commerciaux auxquels celles-ci se heurtent et comment la solution UDA de NVIDIA relève les défis et élimine ces obstacles.

Les défis/obstacles

Pour intégrer le matériel au système d'exploitation (SE), les pilotes doivent être écrits en adoptant une structure en couches. La couche d'abstraction du matériel (HAL) renferme le code qui contrôle directement le matériel. Les dépendances avec le système d'exploitation constituent le « back-end » du pilote, souvent qualifié de partie « commune » puisque tous les pilotes de matériel nécessitent tout ou partie de ce code principal (voir figure 1).

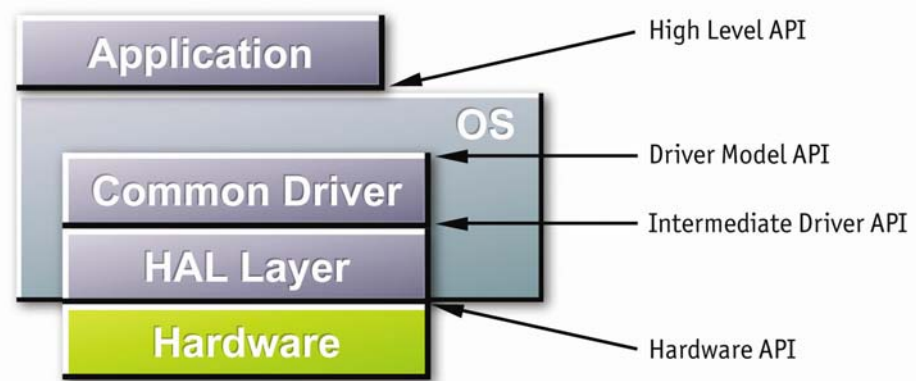


Figure 1. Pile du modèle de pilote courant

Les problèmes techniques

L'emploi de cette pile de pilote courante, bien que simple, a vu se multiplier les obstacles à mesure qu'augmentait la complexité des systèmes et des environnements informatiques. Trois aspects de l'environnement d'exploitation influent sur les pilotes :

- ❑ **Les versions des systèmes d'exploitation** par exemple les différents systèmes d'exploitation de Microsoft® Windows® (Windows NT®, Windows 2000, Windows XP, Windows 9x), Linux et Apple® Mac OS X.
- ❑ **Les interfaces de programmation d'applications (API)**, y compris GDI, VPE, WDM, OpenGL® et Microsoft DirectX®.
- ❑ **Le matériel** pris en charge par la vaste gamme des solutions graphiques et chipsets de NVIDIA : passées, présentes et futures.

Un seul pilote monolithique finirait par atteindre une taille démesurée pour des performances très médiocres s'il devait prendre en charge toutes les solutions offertes dans chacun de ces domaines. En conséquence, l'architecture de pilote traditionnelle a débouché sur la multiplication des versions, chaque version de pilote prenant en charge un sous-ensemble de systèmes d'exploitation, d'API et de matériel.

L'application de l'architecture et des techniques de programmation de pilotes traditionnelles présente de nombreux inconvénients :

- ❑ **Des délais de commercialisation accrus**
Tous les pilotes doivent être revus pour prendre en charge les nouvelles versions des systèmes d'exploitation, API et solutions matérielles. La multitude de combinaisons possibles se traduit par des procédures de test et de certification longues.
- ❑ **Des coûts d'assistance supérieurs**
Les différentes versions de pilotes pour les différents SE, API et solutions matérielles se traduisent par une augmentation du temps consacré à la maintenance des équipements informatiques afin d'assurer la stabilité des combinaisons matériel/logiciels.
- ❑ **Des performances moindres**
Il devient pratiquement impossible d'ajuster toutes les variantes d'exécution avec un si grand nombre de pilotes. Par ailleurs, l'utilisation d'un seul et énorme pilote ralentit encore les performances.
- ❑ **Une évolutivité réduite**
Une approche conventionnelle en matière de programmation ne permet pas d'évoluer correctement à mesure que les logiciels et le matériel se sophistiquent. La complexité d'un pilote monolithique augmente de manière exponentielle en anéantissant toute possibilité de maintenir la qualité et les performances.

Les aspects commerciaux

L'analyse commerciale de l'architecture de pilote traditionnelle a mis en évidence d'autres inconvénients :

- ❑ **Le coût élevé du développement des logiciels**
Une entreprise qui utilise des pilotes traditionnels doit disposer d'un nombre élevé d'ingénieurs système pour la gestion et le débogage de tous les pilotes. Ces coûts se répercutent sur les clients.
- ❑ **La lourdeur du processus de distribution**
La gestion de la qualité est complexe et laborieuse. Pire encore, tous les tests doivent être répliqués à chaque échelon de la chaîne logistique : chez le vendeur du pilote, les OEM, les vendeurs de systèmes et dans les grosses entreprises qui certifient chaque équipement avant son déploiement.
- ❑ **Des phases de test et de certification lentes et onéreuses**
La certification doit être effectuée pour chaque combinaison pilote-processeur et pour chaque système intégrant le pilote. Les gros pilotes monolithiques doivent généralement être soumis à des tests plus nombreux et les procédures de certification peuvent devenir très longues.

La solution de NVIDIA

Les techniciens de NVIDIA ont cherché à mettre au point une architecture de pilote, qui minimise les frais de propriété globaux pour les processeurs NVIDIA, optimise les performances et assure une stabilité et une compatibilité sans égales sur toute la gamme de ses GPU et chipsets. La concrétisation de leurs efforts est l'architecture de pilote unifiée de NVIDIA (UDA), une solution qui atteint tous ces objectifs et place NVIDIA à la pointe de l'innovation technologique (voir figure 2).

La démarche adoptée pour l'UDA de NVIDIA repose sur le principe que le logiciel des pilotes NVIDIA se doit de privilégier la mise en œuvre des fonctionnalités des API au lieu de s'attacher aux différences du matériel. Toutes les opérations de contrôle du matériel sont gérées au moyen du modèle de programmation orientée objet basée sur les classes. Les fonctionnalités plus sensibles aux performances (les fonctions de traitement graphique et audio) exploitent une couche d'abstraction mise en œuvre dans le matériel qui réside sur les chips NVIDIA. Cette solution minimise la surcharge de code associée aux pilotes traditionnels. Cette combinaison matériel/logiciel est la clé d'un équilibre performances-qualité exceptionnel, impossible à atteindre avec les solutions reposant uniquement sur le logiciel.

En sus, ne se contentant pas d'afficher d'excellentes performances, les pilotes de NVIDIA assurent une compatibilité sans faille avec les produits existants et à venir. Grâce à des techniques de programmation basées sur des classes et à un niveau élevé d'abstraction qui protègent le logiciel du matériel de bas niveau, un même pilote peut prendre en charge toutes les versions du matériel auquel il est associé.

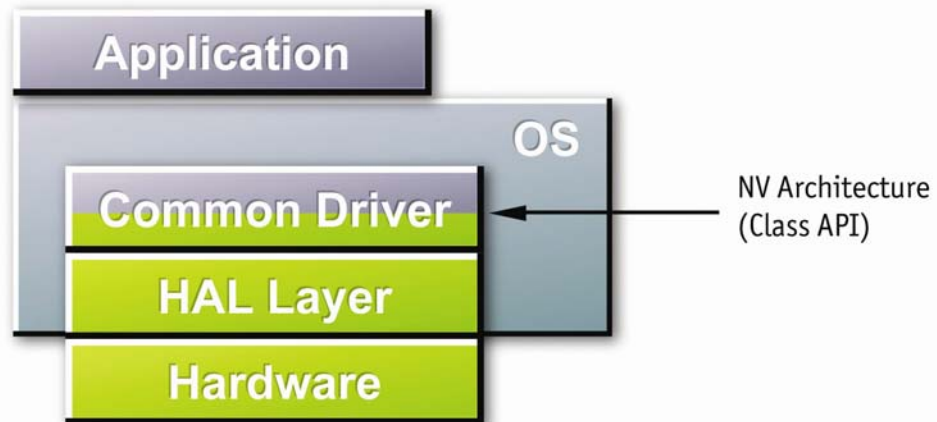


Figure 2. Pile du modèle du pilote NVIDIA pour les GPU et les fonctionnalités sensibles aux performances

La programmation orientée objet

L'UDA de NVIDIA comprend une abstraction orientée objet de la fonctionnalité matérielle, autrement dit un modèle basé sur des classes. Les *objets* et les *classes* fournissent des mécanismes qui permettent de contrôler efficacement l'accès aux fonctionnalités et aux informations contextuelles. En abstrayant la fonctionnalité du moteur, l'architecture adoptée par NVIDIA en conserve l'indépendance tout en réduisant la charge de travail de l'UC, de la mémoire et du bus pour les fonctions système et graphiques.

Note: Une **classe** est une définition abstraite d'une fonction matérielle donnée (à implémenter sous forme de mappage de registre). Un **objet** est une instance d'une classe (un mappage de registre spécifique) et contient, par conséquent, un contexte particulier. Une **méthode** est un registre en écriture seule à l'intérieur d'un objet.

La structure de classes et les *méthodes* de l'UDA imitent les modèles des API courantes et leur usage. Par exemple, le pilote des GPU NVIDIA fournit des classes conçues pour s'adapter à la perfection aux dernières interfaces de programmation de Microsoft Windows.

Le gestionnaire de ressources

Au sein de la couche matérielle de l'UDA, le gestionnaire de ressources de NVIDIA constitue une solution intelligente pour la reconnaissance et la gestion des classes et des objets (voir figure 3). Le logiciel pilote NVIDIA fournit une bibliothèque au niveau du noyau pour la reconnaissance et la gestion des classes et des objets. Cette innovation NVIDIA permet de gérer les états et les contextes en allouant toutes les ressources architecturales nécessaires à chaque pilote client. L'on remarquera par ailleurs que dans ce schéma de gestion contextuelle, les différents clients ne se rendent pas compte que d'autres utilisateurs utilisent aussi le chip.

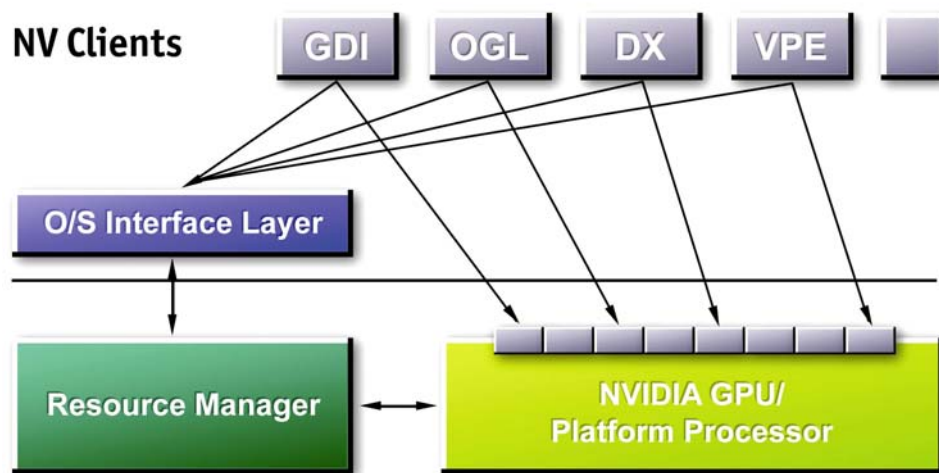


Figure 3. Modèle d'architecture de pilote montrant le gestionnaire de ressources

L'allocation et la gestion des *objets* (ressources matérielles) de l'architecture offre plusieurs fonctionnalités intéressantes pour la gestion de la compatibilité. Comme décrit plus haut, le gestionnaire de ressources établit un canal entre le matériel et le pilote client de sorte à minimiser le code de pilote nécessaire pour l'identification des ressources matérielles.

En reconnaissant des classes désignées, le gestionnaire de ressources peut gérer les cas d'exception. Les différences matérielles restent ainsi invisibles pour l'utilisateur et peuvent être gérées de manière à assurer la compatibilité du pilote sur toutes les plates-formes. Le gestionnaire de ressources permet également l'utilisation d'interfaces de classe existantes ce qui assure la compatibilité avec les versions suivantes du matériel. L'interfaçage avec tout le matériel se fait en utilisant ces conventions standard, ce qui signifie que le code existant n'a pas besoin d'être modifié pour la prise en charge de nouveaux produits.

Les autres fonctions assurées par le gestionnaire de ressources sont les suivantes :

- ❑ **Gestion des erreurs :**
La reconnaissance des classes par le gestionnaire de ressources constitue un excellent mécanisme de débogage.
- ❑ **Portions d'architecture dépendantes du système d'exploitation**
Le gestionnaire de ressources peut également gérer les différences existants entre divers SE au niveau de la gestion de la mémoire, du contrôle des processus/threads, du *plug-n-play*, de la gestion de la consommation, etc.

Les avantages de l'UDA

L'UDA de NVIDIA a été testée sur plusieurs générations de technologies NVIDIA et de nombreuses versions des produits NVIDIA qui remontent jusqu'à 1998, année du lancement du RIVA128™. Les OEM et les clients de NVIDIA confirment qu'ils tirent de nombreux avantages de l'UDA :

- ❑ **Un meilleur design, davantage de fonctionnalités**
En réduisant le nombre d'ingénieurs système chargés du contrôle et de la gestion des nouvelles versions, davantage de ressources humaines peuvent être affectées à l'optimisation des performances et à l'ajout de nouvelles fonctionnalités. L'UDA de NVIDIA permet également de changer plus librement le matériel de bas niveau sans aucune conséquence pour les OEM ou les utilisateurs finals.
- ❑ **Des performances qui riment avec excellence**
Intégrer des fonctionnalités de pilote sans le matériel assure des performances exceptionnelles et accélère toutes les applications.
- ❑ **Stabilité et fiabilité**
L'UDA de NVIDIA a passé avec succès le test du temps et contribue à l'excellente réputation du matériel et des logiciels de l'entreprise en matière de stabilité et de fiabilité.
- ❑ **Des délais de commercialisation extrêmement courts**
Les produits de NVIDIA font leur apparition sur le marché en un temps record : moins de 100 jours s'écoulent entre les premiers croquis et la mise en production de la puce, contre plus de six mois dans le reste du secteur. Les OEM accèdent ainsi plus rapidement aux nouvelles technologies. De plus, l'UDA permet

d'installer et de gérer les nouvelles solutions avec une extrême facilité puisqu'un seul et même pilote prend en charge aussi bien les versions précédentes que les nouvelles.

❑ **Des frais de propriété globaux réduits**

Les gestionnaires d'équipements informatiques peuvent générer une unique image logicielle avec une seule version du pilote NVIDIA et la déployer sur des milliers d'ordinateurs incorporant différents processeurs NVIDIA. Ils ne doivent ainsi homologuer, gérer et maintenir qu'une seule architecture logicielle pour de nombreuses configurations matérielles.

Conclusion

L'UDA de NVIDIA a établi une nouvelle norme en matière d'architecture de pilote en assurant des performances très élevées, une compatibilité hors pair et des frais de propriété globaux fortement réduits. L'UDA est bien plus qu'une simple trouvaille technologique, ses avantages profitent à tous ceux qui choisissent la technologie NVIDIA :

- ❑ **Les entreprises** gagnent temps et argent grâce à la simplification des procédures de gestion et d'évolution de leur parc informatique.
- ❑ **Les gestionnaires d'équipements informatiques** peuvent gérer différents produits matériels avec les mêmes pilotes, adopter de nouveaux ordinateurs sans changer de pilotes, réduire les délais de mise en œuvre des nouveaux systèmes et assurer l'assistance pour des configurations flexibles de plates-formes matérielles.
- ❑ **L'utilisateur final** bénéficie d'un accès rapide aux nouveaux produits et peut utiliser ses vieux jeux et applications sur un ordinateur neuf.
- ❑ **Les développeurs de logiciels** peuvent écrire une seule version d'une application qui sera optimisée pour toutes les solutions NVIDIA.
- ❑ **Les vendeurs de systèmes** peuvent offrir à leurs clients la possibilité de générer une seule image logicielle prenant en charge une ligne complète de matériel, ce qui simplifie l'introduction de nouveaux ordinateurs au sein d'une structure existante, allège considérablement les cycles de certification et de qualification et se traduit par une réduction des frais de propriété globaux.

Les solutions de NVIDIA continueront à miser sur l'UDA afin de toujours offrir des pilotes logiciels hautes performance, rapides, fiables et de haute qualité. L'UDA élimine les obstacles qui entravaient les architectures de pilote traditionnelles d'hier et permet d'introduire rapidement des innovations matérielles sans conséquence sur la compatibilité avec les systèmes et les applications existantes. Aucun autre fabricant de processeurs graphiques et de chipsets n'est en mesure de proposer l'utilisation d'une seule et même image logicielle pour toute la gamme de produits d'une entreprise.

Avis

TOUTES LES SPÉCIFICATIONS DE CONCEPTION, CARTES DE RÉFÉRENCE, FICHIERS, PLANS, DIAGNOSTICS, LISTES ET AUTRES DOCUMENTS DE NVIDIA (LE « MATÉRIEL ») SONT FOURNIS « EN L'ÉTAT ». NVIDIA N'ACCORDE AUCUNE GARANTIE, QU'ELLE SOIT EXPLICITE, IMPLICITE, LÉGALE OU AUTRE, SUR LE MATÉRIEL FOURNI, ET DÉCLINE EXPRESSÉMENT TOUTE RESPONSABILITÉ IMPLICITE EN CE QUI CONCERNE LA NON CONTREFAÇON, LA VALEUR MARCHANDE ET L'ADÉQUATION À UN USAGE PARTICULIER D'UN TEL MATÉRIEL.

Les informations fournies sont jugées précises et fiables. Toutefois, NVIDIA Corporation ne peut être tenue pour responsable des conséquences de l'utilisation de ces informations ou d'une violation quelconque de brevets ou autres droits de tiers pouvant résulter de leur utilisation. Aucune licence n'est octroyée implicitement ou de quelque autre manière sous quelque brevet ou droits de brevet de NVIDIA Corporation. Les informations techniques contenus dans ce document peuvent être modifiées sans préavis. Ce document annule et remplace toute autre information fournie antérieurement. Les produits de NVIDIA Corporation ne bénéficient pas d'une autorisation d'utilisation en tant que composants critiques dans des dispositifs ou systèmes essentiels sans l'accord préalable écrit de NVIDIA Corporation.

NVIDIA et le logo NVIDIA sont des marques déposées et RIVA128 est une marque commerciale de NVIDIA Corporation.

D'autres noms de sociétés et de produits peuvent être des marques commerciales ou déposées des entreprises auxquelles ils sont associés.

Copyright NVIDIA Corporation 2003



NVIDIA.
NVIDIA Corporation

www.nvidia.fr