

INFORMATIQUE EMBARQUÉE

La virtualisation devient une réalité industrielle

▼ Largement répandue pour les systèmes d'information d'entreprise et de plus en plus courante dans les télécoms, la virtualisation fait son apparition dans l'industrie. Cette technologie consiste à faire cohabiter plusieurs systèmes d'exploitation sur une plate-forme matérielle unique. Elle favorise la réutilisation du code existant et la création d'architectures dimensionnées au plus juste, et s'accorde parfaitement avec les technologies de processeurs multicœur.

Le concept de virtualisation regroupe un ensemble de techniques visant à améliorer les performances globales d'un système informatique. Le principe de base est de faire croire à un système d'exploitation qu'il s'exécute sur une plate-forme matérielle réelle, alors qu'il tourne sur une plate-forme virtuelle. Pour ce faire, un "hyperviseur" est inséré entre la plate-forme matérielle et le (ou les) système(s) d'exploitation. L'hyperviseur est une couche logicielle qui se charge de dupliquer le matériel en autant de machines virtuelles que nécessaire. « Ce qui n'est pas une mince affaire à concevoir, indique Bruno Rouchouse, directeur technique chez Wind River. Il s'agit en effet d'ajouter un nouvel ordonnanceur au système, et cela sans que les ordonnanceurs des différents systèmes d'exploitation ne s'en aperçoivent. » Grâce à cette technologie, il devient

possible de faire cohabiter plusieurs systèmes d'exploitation (ou OS, pour Operating Systems) sur un même matériel. Chacun d'entre eux s'exécute comme s'il était le seul à disposer des ressources hardware. Selon les besoins, il est possible d'installer au-dessus de l'hyperviseur un OS du commerce (générique ou temps réel), ou une application unique installée nativement sur le matériel. Celle-ci peut être programmée en langage C (on l'appelle alors *bare metal*), mais aussi en Java (on rajoute une machine virtuelle supplémentaire) ou en Ada (langage robuste largement répandu dans l'aéronautique).

Déjà incontournable dans les télécoms

Si la virtualisation est une technique relativement complexe, elle n'en est pas moins très ancienne. Les premières applications remontent en effet aux débuts de l'informatique, puisque le groupe IBM a conçu les premières machines virtualisées dans les années 60. Depuis lors, elles se sont très largement répandues dans le monde des systèmes d'information d'entreprises (ou IT, pour Information Technologies) et deviennent progressivement incontournables dans les télécoms. A tel point qu'aujourd'hui la plupart des serveurs sont proposés avec des solutions de virtualisation pour la mise en parallèle de tâches (des processus de recherche de données pour un serveur d'entreprise, par exemple, ou des communications pour un serveur télécoms). On optimise l'utilisation du parc matériel en multipliant les tâches réalisées sur un seul serveur.

Réduire les coûts, mais pas seulement

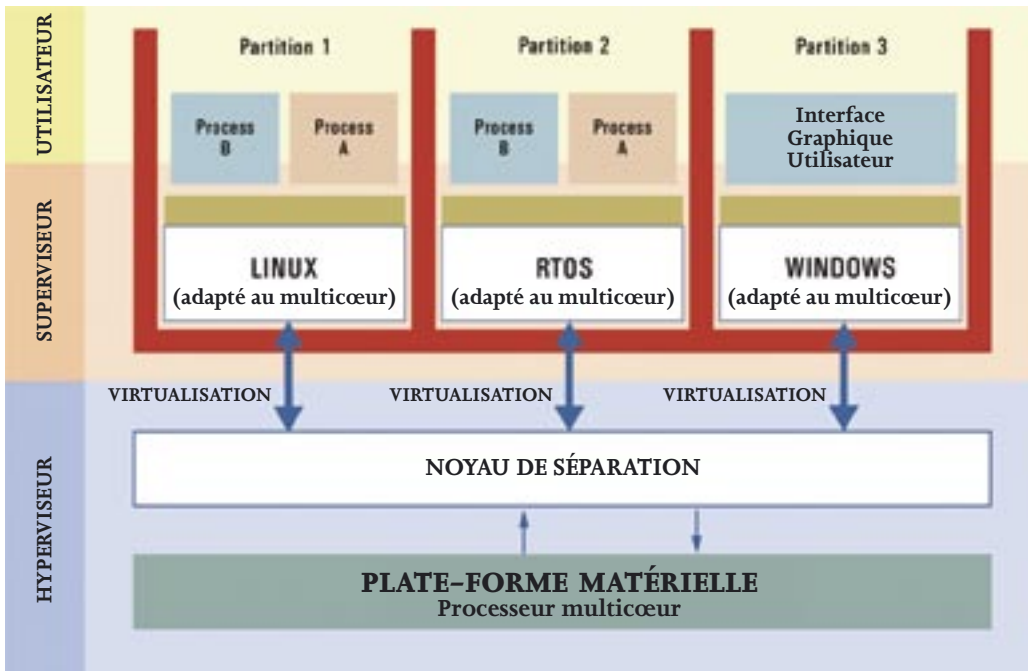
L'objectif principal de la virtualisation est la réduction des coûts matériels. En exécutant sur une même plate-forme des applications qui tournaient auparavant sur des machines différentes, on diminue d'autant le coût global de l'installation. Mais ce n'est pas tout, car la virtualisation est aussi un bon moyen d'assurer la pérennité des applications. En effet, la virtualisation est absolument transparente pour la partie applicative. Un programme qui s'exécute sur un OS pourra continuer à s'exécuter sur le Guest OS (voir encadré) sans aucune modification. De plus, dans les gros systèmes tels que ceux utilisés dans les télécoms, il est fréquent d'avoir à intervenir à chaud sur des serveurs. Impensable en effet d'effectuer des tests réseau, des téléchargements de mises à jour, voire des redémarrages d'application en arrêtant le système. Grâce à la virtualisation, une modification sur une application n'a en théorie aucune influence sur le fonctionnement des autres, même si elles sont installées sur le même serveur. Jacques Brygier, directeur marketing chez Sysgo, va plus loin : « L'une des attentes essentielles de la virtualisation concerne la sûreté de fonctionnement. Une application industrielle doit bien entendu être exempte de bugs, mais si jamais il arrive qu'elle plante, elle ne doit surtout pas entraîner les autres avec elle. Or c'est l'une des garanties que peut apporter une virtualisation bien conçue. » Un autre intérêt est l'optimisation de l'utilisation des proces- →

L'essentiel

- ▶ La virtualisation consiste à faire tourner plusieurs OS sur une seule plate-forme matérielle.
- ▶ La technologie est déjà largement exploitée pour les data centers et autres serveurs d'entreprise.
- ▶ Les éditeurs ont adapté la virtualisation aux exigences de l'informatique embarquée.
- ▶ Les premières applications industrielles font leur apparition. Les marchés visés sont l'aéronautique, les transports ou le contrôle industriel.



Les systèmes MicroTCA sont à la fois peu encombrants et très performants... Des caractéristiques idéales pour la virtualisation des applications d'automatismes.



LynuxWorks

Plus que la seule multiplication des OS, la virtualisation vise surtout la spécialisation des OS : chacun d'entre eux prend en charge la partie de l'application pour laquelle il est le plus adapté. Une des utilisations les plus classiques consiste à faire cohabiter une application sous Linux (stockage de données, par exemple), une application temps réel (pour le contrôle d'une machine) et une application sous Windows (pour la partie affichage et l'interface utilisateur).

→ seurs. Grâce aux outils fournis avec les solutions de virtualisation, les développeurs peuvent étudier au plus près l'activité du processeur. Ils évaluent sa charge, mais ils organisent aussi le traitement des tâches en provenance des différents OS. Ces outils permettent de déterminer le nombre de machines qu'il est possible de faire tourner sur un processeur. Car on ne peut pas multiplier à l'infini les machines virtuelles (le processeur ne pourrait plus suivre).

Enfin, la virtualisation va de pair avec l'émergence d'une autre tendance du secteur l'informatique : la disponibilité de processeurs multicœur. Conçus pour pallier les problèmes de dissipation de chaleur qu'impliquait la multiplication des transistors, ces processeurs participent tout comme la virtualisation à l'augmentation du nombre d'applications exécutables simultanément. Bien entendu, il est tout à fait possible d'utiliser un processeur double cœur sans hyperviseur. Les technologies SMP (Symmetric Multiprocessing) et AMP (Asymmetric Multiprocessing) ont été conçues pour les architectures multiprocesseur, et s'appliquent aussi au multicœur. En SMP, un ordonnanceur se charge d'envoyer chaque instruction vers l'un ou l'autre des cœurs selon leur disponibilité. Avec l'AMP, en revanche, les applications sont hiérarchisées pour s'exécuter en priorité sur l'un ou l'autre des cœurs. Cette technique consiste le plus souvent à exécuter simultanément un OS temps réel et un OS généraliste. Il faut pour cela réserver l'un des

cœurs à l'OS temps réel, et que les instructions en provenance de ce dernier soient prioritaires. Reste toutefois qu'un système AMP, aussi performant soit-il, sera toujours plus sensible aux plantages car la mémoire est partagée. Si l'un des deux OS plante, tout le système s'arrête. Le principe de l'AMP ne garantit pas l'étanchéité entre les OS. « C'est pour cette raison que la virtualisation est l'un des moyens les plus élégants d'utiliser les processeurs multicœur, avance Didier Irlande, responsable marketing Europe chez VirtualLogix. Une même application peut tourner N fois sur N cœurs sans qu'il y ait besoin de retoucher au code. »

Des télécoms vers l'informatique embarquée

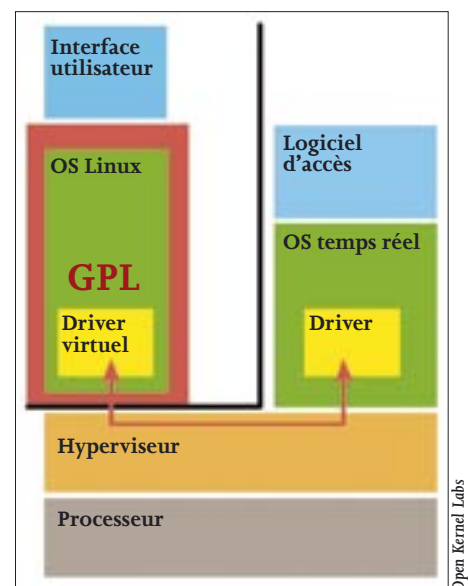
Progressivement, les solutions de virtualisation pour serveurs s'adaptent à l'électronique embarquée. Il faut dire que ce secteur fait depuis longtemps face à une augmentation du nombre de calculateurs et de logiciels embarqués. « On distingue deux tendances récurrentes dans l'informatique embarquée, résume Jacques Brygier, directeur marketing chez Sysgo. Soit on cherche à disposer de plus de fonctions et de plus de performances dans le même format, soit on vise à réduire le format tout en gardant le même niveau de performances. Dans les deux cas, la virtualisation représente la solution tout indiquée. »

L'industrie aéronautique est la première à avoir identifié l'utilité de systèmes embarqués répartis. Afin de réduire le nombre de calculateurs, les constructeurs ont créé les systèmes IMA, pour Integrated Modular Avionics,

dans les années 1990. Il s'agit d'un standard visant à garantir la sécurité et la sûreté de fonctionnement des applications, tout en séparant des fonctions selon leur degré de criticité (depuis les commandes de vol, très critiques, jusqu'aux fonctions d'infotainment, peu critiques). L'IMA est basée sur le concept du multipartitionnement, un premier pas vers la virtualisation pour l'embarqué. Aujourd'hui, les nouveaux téléphones portables et autres équipements électroniques grand public intègrent des solutions de virtualisation légères, optimisées pour les capacités modestes des calculateurs. Jacques Brygier (Sysgo) indique que « les clients industriels, très exigeants sur les aspects de pérennité, accusent toujours un retard de quelques années par rapport au secteur grand public concernant l'adoption des nouvelles technologies. Les premiers projets importants dans le contrôle industriel et dans les transports n'ont réellement démarré pour nous qu'en 2007. »

Une offre récente

Aujourd'hui, tous les principaux éditeurs proposent des solutions de virtualisation spécialement adaptées au secteur industriel. Pour ce faire, ils ont adapté leurs offres. Premier impératif : installer un hyperviseur ne doit pas ralentir le système, sous peine de ne plus pouvoir utiliser les OS temps réel. La réponse est apportée par la paravirtualisation. « Il s'agit d'une modification des couches basses de l'OS, indique Andy Ramianrasoa, responsable marketing chez Trango. Cela évite que l'hyperviseur ne ralentisse les échanges entre les OS et le matériel. Tout le monde parle la même langue, la langue de l'hyperviseur. On considère cette méthode comme un moyen de rendre les OS coopératifs. » La



Open Kernel Labs

Les licences GPL des OS libres ont la particularité de s'étendre à tous les drivers. En exécutant cet OS dans un environnement virtualisé, on empêche la propagation de la GPL.

Le principe de la virtualisation

Virtualiser un système consiste à ajouter une couche logicielle entre le matériel et les systèmes d'exploitation. Cette couche logicielle, appelée "hyperviseur", est chargée de faire cohabiter un certain nombre de machines virtuelles. Les trois caractéristiques essentielles d'un hyperviseur: avoir un contrôle absolu sur toutes les ressources matérielles, fournir aux Guest OS (*Operating Systems* "invités", c'est-à-dire qui tournent sur des machines virtuelles) un environnement exactement identique à celui de la machine réelle, et exécuter les différentes applications en dégradant le moins possible les performances. « *L'hyperviseur propose une vue virtuelle des ressources matérielles. C'est en quelque sorte un OS pour les OS* », explique Stéphane Deruelle, responsable Europe du Sud chez Wind River.

Concrètement, un hyperviseur se compose d'un micronoyau, d'un system software et d'un ordonnanceur. Le micronoyau contient le "cœur" de l'hyperviseur. Il est radicalement différent selon les éditeurs, selon qu'ils favorisent la performance, la sécurité des données ou la sûreté de fonctionnement de l'application. Le system software, quant à lui, prend en charge les fonctions "évoluées" de l'hyperviseur. Il connaît le matériel et tous les OS qui y sont installés. Il effectue donc les éventuels échanges de données entre les OS, gère les partitions ainsi que les droits des utilisateurs, etc. Enfin, l'ordonnanceur est en quelque sorte le chef d'orchestre du système. Il choisit l'ordre selon lequel sont exécutés les OS, en fonction de leur priorité, de leur criticité ou de la charge du processeur. Ces trois éléments forment une couche d'abstraction: les machines virtuelles mises à disposition par l'hyperviseur offrent toutes la même interface d'accès au matériel. Cela implique que les OS installés au-dessus de cette couche d'abstraction doivent eux aussi parler le même langage, ce qui

n'est pas le cas. Une solution consiste à intégrer dans l'hyperviseur les outils pour traduire les instructions à la volée. C'est ce que l'on appelle la virtualisation "native". Mais l'hyperviseur doit être développé à chaque changement d'OS, et les opérations de codage et de décodage d'instructions ralentissent considérablement le système. Aussi, si la performance du système est au cœur des préoccupations, on utilisera plutôt le principe de paravirtualisation. Cette technique consiste à adapter les couches basses de l'OS afin que toutes les instructions qui sont transférées à l'hyperviseur puissent être traitées directement par ce dernier. Cela a pour conséquence de ne pas dégrader les performances par rapport à un système non virtualisé, mais impose tout de même une modification de l'OS. Bien sûr, modifier les couches basses d'un OS peut paraître complexe à la plupart des utilisateurs, mais l'avantage est qu'un éditeur peut proposer des OS paravirtualisés adaptés à son hyperviseur. Il doit développer une gamme de drivers de communication (un par OS), mais n'est plus obligé de concevoir un micronoyau différent pour chaque architecture (l'hyperviseur reste le même quel que soit l'OS). La paravirtualisation n'est cependant pas le seul moyen d'assurer des performances temps réel. Des mécanismes matériels intégrés à certains processeurs permettent de conserver le comportement temps réel des OS invités. Intel a rajouté ces mécanismes à ces derniers produits (processeurs Core 2 Duo, ainsi que les gammes Xeon et Itanium). Sur les systèmes équipés de cette technologie Intel VT (pour *Virtualization Technology*), plus besoin de modifier l'OS. Le composant se charge de reproduire l'environnement d'exécution pour lequel il a été conçu, et donc l'OS peut conserver toutes ses couches basses. On ne parle plus alors de paravirtualisation, mais de "Hardware Assisted Virtualization".

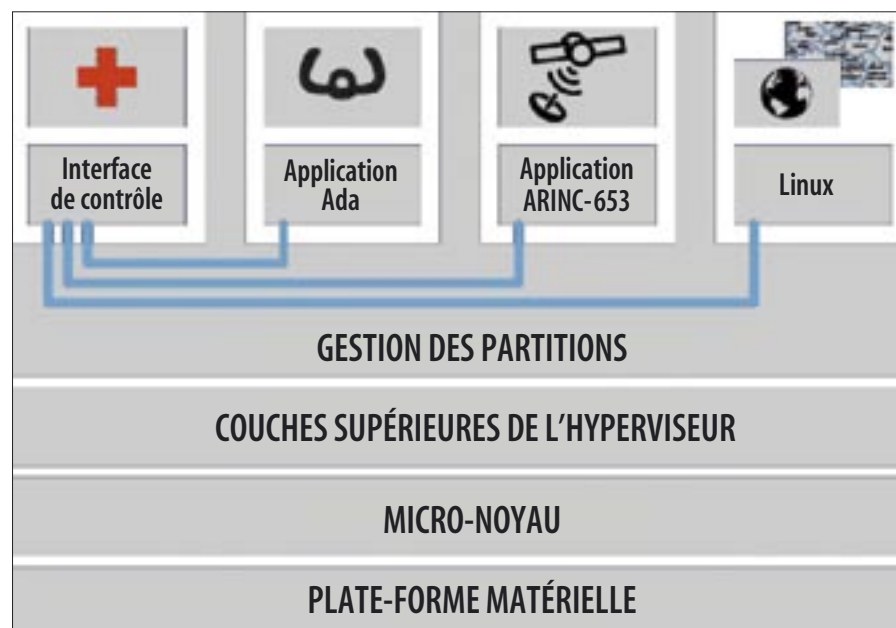
paravirtualisation impose d'utiliser un "Guest OS" (OS invité) spécialement adapté à l'hyperviseur. Cela oblige à changer les BSP (Board Support Package) de chaque système d'exploitation par un BSP conçu pour cet hyperviseur. « Les éditeurs proposent déjà des versions paravirtualisées pour les principaux OS du commerce, commente Stuart Fischer, directeur marketing produit pour LynxSecure chez LynuxWorks. Ils les développent sur mesure si le client utilise un OS spécifique pour son application. »

Comme dans le monde IT, les solutions de virtualisation embarquée peuvent apporter des solutions fiables pour le traitement du partitionnement. Mais l'embarqué présente d'autres contraintes. La première est bien entendu l'espace mémoire restreint, quelques kilo-octets, pour les OS et les applications. Les hyperviseurs de plusieurs méga-octets des serveurs télécoms ne peuvent évidemment pas être utilisés.

Les années 2000 ont vu l'apparition sur le marché de véritables hyperviseurs embarqués, optimisés pour ne pas dépasser quelques milliers de ligne de code. Ils sont proposés par l'essentiel des éditeurs de solutions

embarquées. Parmi eux: Green Hills, Lynux Works, Open Kernel Labs, Sysgo, Trango, VirtualLogix ou Wind River. Bien que différentes dans le détail, les offres de ces éditeurs satisfont toutes aux besoins de l'informatique em-

barquée (performances, partitionnement offrant différents niveaux de fiabilité, et empreinte mémoire réduite). Mais ce ne sont pas les seules caractéristiques des hyperviseurs embarqués. →



Une des possibilités offertes par la virtualisation consiste à réserver une des partitions pour le contrôle de l'état du système. Depuis celle-ci, l'utilisateur pourra effectuer des diagnostics sur les autres partitions, les redémarrer, charger des applications, voire changer les niveaux de priorité.

Sygo

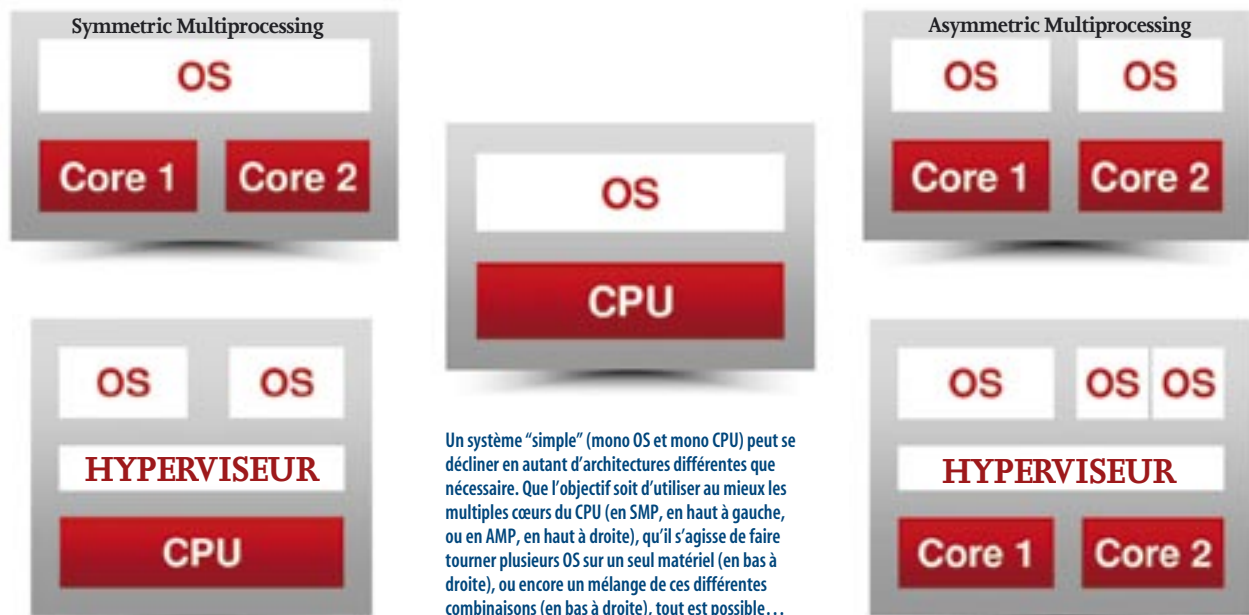
De nombreuses possibilités d'architectures

La virtualisation est un concept large, qui englobe un grand nombre de techniques (paravirtualisation, émulation, *hardware assisted virtualization*, etc.) pour s'adapter à tous les besoins. Elle n'admet aucune limite théorique en ce qui concerne le nombre d'OS et leur mode de fonctionnement (OS générique, OS temps réel ou application *bare metal*). La seule contrainte est la puissance de calcul disponible.

Avec l'arrivée des processeurs multicœur, et notamment les gammes d'Intel qui intègrent des capacités de virtualisation natives, le nombre d'architectures possibles devient quasiment infini. Chaque OS dispose de sa propre machine virtuelle, mais

celle-ci peut s'exécuter sur un ou plusieurs cœurs. Un cœur peut aussi être réservé entièrement à un OS, dans le cas de contraintes temps réel fortes.

Par ailleurs, si la virtualisation "duplique" le matériel en machines virtuelles, cela signifie qu'elles doivent forcément être toutes identiques. L'utilisateur garde un contrôle total sur ces machines virtuelles. Si un OS temps réel effectue un contrôle critique sur un process, le port chargé de l'acquisition des données peut tout à fait être dédié à cet OS (les autres n'y auront pas accès et ne le voient même pas). Cela évite que les OS génèrent des interruptions en même temps lorsque les données arrivent.



→ Sécurité et portabilité : des besoins particuliers

De nombreuses applications nécessitent un environnement sécurisé, dans le sens où elles doivent résister aux attaques extérieures, d'origine accidentelle ou malveillante. Les hackers arrivent en effet aujourd'hui à descendre de plus en plus profondément à l'intérieur des OS (et des matériels sur lesquels ils sont installés). De manière générale, les systèmes sont donc de plus en plus vulnérables aux attaques. De plus, il est naturel qu'un système qui concentre plusieurs applications soit plus exposé aux menaces qu'un système monoapplication. Certains éditeurs ont donc intégré aux hyperviseurs embarqués des protections qui assurent une indépendance totale des machines virtuelles, de telle sorte qu'il est impossible d'accéder à un OS critique en pénétrant le système via l'OS générique. Mais il faut bien noter que le concept

de virtualisation ne suffit pas à lui seul à assurer la sécurité d'un système. Si le besoin en sécurité est important, la virtualisation doit s'accompagner d'une implémentation entièrement pensée dans ce sens.

La portabilité des applications était également une condition indispensable pour l'adoption de la virtualisation dans l'industrie. Des éditeurs tels que VirtualLogix proposent des hyperviseurs pour la plupart des architectures utilisées en informatique embarquée : Intel x86, Freescale ARM, mais aussi des DSP (ceux de Texas Instruments, notamment). A noter que Intel a fait office de précurseur dans ce domaine, en proposant une virtualisation matérielle sur ses processeurs récents. Il s'agit de la technologie VT (Virtualization Technology). L'avantage : plus besoin de paravirtualiser les OS pour conserver les aspects temps réel, car de nouvelles fonctions du processeur évitent le besoin de traduire certaines instructions privilégiées. En

revanche, une application développée avec la technologie VT interdit la portabilité sur une autre plate-forme que la x86 d'Intel.

Mais selon Stuart Fischer (LynuxWorks), la sauvegarde du code existant reste l'une des principales motivations pour le passage à une solution virtualisée. « Il est toujours délicat de changer de matériel informatique dans une usine, explique-t-il, car cela implique souvent l'adaptation des applications. Il faut recompiler le logiciel pour le porter sur une nouvelle architecture, ou le modifier pour qu'il s'adapte à un nouveau système d'exploitation. Dans tous les cas cela entraîne des coûts supplémentaires, et parfois on prend le risque qu'une application ne fonctionne plus du tout. » Les hyperviseurs du marché sont multiOS par définition. Si une application a été développée pour un OS en particulier, il suffit de porter cet OS sur l'hyperviseur (en le transformant en Guest OS si nécessaire). En aucun cas le code n'a besoin d'être retouché. Autre aspect intéressant : pour les OS embarqués

sur base Linux, la virtualisation est un moyen de limiter la propagation des licences libres. En effet, la multiplication des OS basés sur le cœur Linux dans l'embarqué peut poser des problèmes de propriété intellectuelle. Car si des composants d'un OS ou des drivers sont développés dans un environnement Open Source, ils deviennent eux aussi Open Source. Cela implique que le code source de ces logiciels entre dans le domaine public. « Avec des fonctions telles que la technologie Hypercell d'Open Kernel Labs, il est possible d'empêcher cette propagation, commente Rob McCammon, son directeur marketing produit. Le principe : on extrait des fonctions ou des applications qui tournaient sur l'OS libre de droits, puis on les porte sur une machine virtuelle séparée. Ils deviennent des composants autonomes qui tournent directement au-dessus de l'hyperviseur. Ainsi, plus de problèmes pour préserver le savoir-faire propre à l'entreprise. »

Pour finir, on peut se demander comment définir l'architecture d'un système virtualisé. « C'est souvent le client qui s'en charge, s'il dispose des compétences nécessaires en interne, indique Jacques Brygier (Sysgo). Car la plupart des hyperviseurs sont livrés avec des environnements de développement. Ces derniers facilitent beaucoup le travail des architectes logiciel. Selon leur degré d'expertise, ils ont à leur disposition des outils simples, à base de cases à cocher, ou des outils plus perfectionnés avec lesquels les interactions entre l'OS et le matériel sont entièrement configurables. » De manière générale, pour concevoir une application virtualisée, un concepteur d'architecture doit avoir une vision d'ensemble de son application. C'est seulement dans un second temps que peut être définie la compartimentation du système, en prenant en compte les exigences temps réel ainsi que les contraintes de mémoire, de puissance de calcul et de criticité. Rob McCammon (Open Kernel Labs) ajoute

que « contrairement au monde des systèmes d'informations, où l'on vise une séparation totale entre les OS, les applications embarquées nécessitent souvent un maintien des communications entre OS. » Cas typique : si un OS temps réel effectue du traitement de données à la volée, il peut avoir besoin d'envoyer régulièrement des données à l'OS générique à des fins d'archivage. Pour cela, l'essentiel des hyperviseurs embarqués laisse le choix entre un isolement des OS ou la création de communications sécurisées.

De nouveaux marchés

Grâce à ces nouvelles offres de virtualisation, les éditeurs élargissent leur champ de prospection au-delà des seuls marchés des technologies de l'information, télécoms et grand public. L'aéronautique et l'aérospatiale ont été les premiers à implémenter des modèles multipartitionnés, précurseurs de solutions de virtualisation plus sophistiquées. De la même manière, les industries ferroviaire et automobile pourront bientôt confier à des hyperviseurs la répartition des fonctions entre les différents calculateurs. Le contrôle industriel, les OEM et les fabricants de matériel médical figurent également parmi les principales applications visées, car « une des applications typiques reste la multiplication des fonctions d'un équipement. Avec un hyperviseur, il est possible d'utiliser une seule machine pour effectuer à la fois du contrôle temps réel et des fonctions d'affichage, commente Stéphane Deruelle, directeur Europe du Sud chez Wind River. Grâce à cette cohabitation entre un OS générique et un RTOS (OS temps réel), on optimise les coûts d'installation. Et on garantit aussi la pérennité des applications. Si un OS spécifique avait été développé pour une tâche spéciale, on pourra toujours porter cet OS dans un nouvel environnement virtualisé. »

Frédéric Parisot

Un aperçu de l'offre du marché

Chaque solution de virtualisation dispose de caractéristiques qui dépendent principalement de l'éditeur. Si l'aspect sûreté est primordial, il est préférable de se tourner vers les éditeurs qui s'en sont fait la spécialité. Parmi ceux-ci se trouvent Sysgo (PikeOS), Wind River (Wind River Hypervisor), Lynux Works (LynxSecure) et Green Hills Software (Integrity PC). Ces offres, développées par les spécialistes du logiciel embarqué, sont particulièrement adaptées pour la conception de systèmes certifiés (normes ARINC 653 ou DO-178B pour l'aéronautique, par exemple). Mais elles sont également applicables à toutes autres sortes d'équipements embarqués. A l'inverse, des éditeurs issus du monde des microprocesseurs sont plus spécialisés dans l'optimisation des ressources matérielles. C'est le cas du français Trango, racheté récemment par VMware pour adresser le marché des mobiles. Son hyperviseur vise une utilisation à 100 % du CPU. D'autres éditeurs, historiquement spécialisés dans les OS, proposent des offres relativement pluridisciplinaires. C'est le cas de VirtualLogix et de son hyperviseur VLX, qui s'adapte aussi bien aux applications réseau que télécoms ou embarqué (l'accent est mis sur la portabilité et sur l'utilisation d'OS sur base Linux et Windows) et qui offre une solution complète pour la haute disponibilité appelée vHA. La société Open Kernel Labs, enfin, met l'accent sur les performances. Grâce à un partenariat avec l'organisme de recherche NICTA, elle bénéficie de technologies avancées dans le domaine de la séparation des fonctions. Grâce à la fonction Hypercell, n'importe quelle application peut être séparée et isolée de l'OS sur lequel elle était installée, pour tourner directement au-dessus de l'hyperviseur OKL4. Ce système permet de créer des architectures flexibles et modulaires.

Elexis