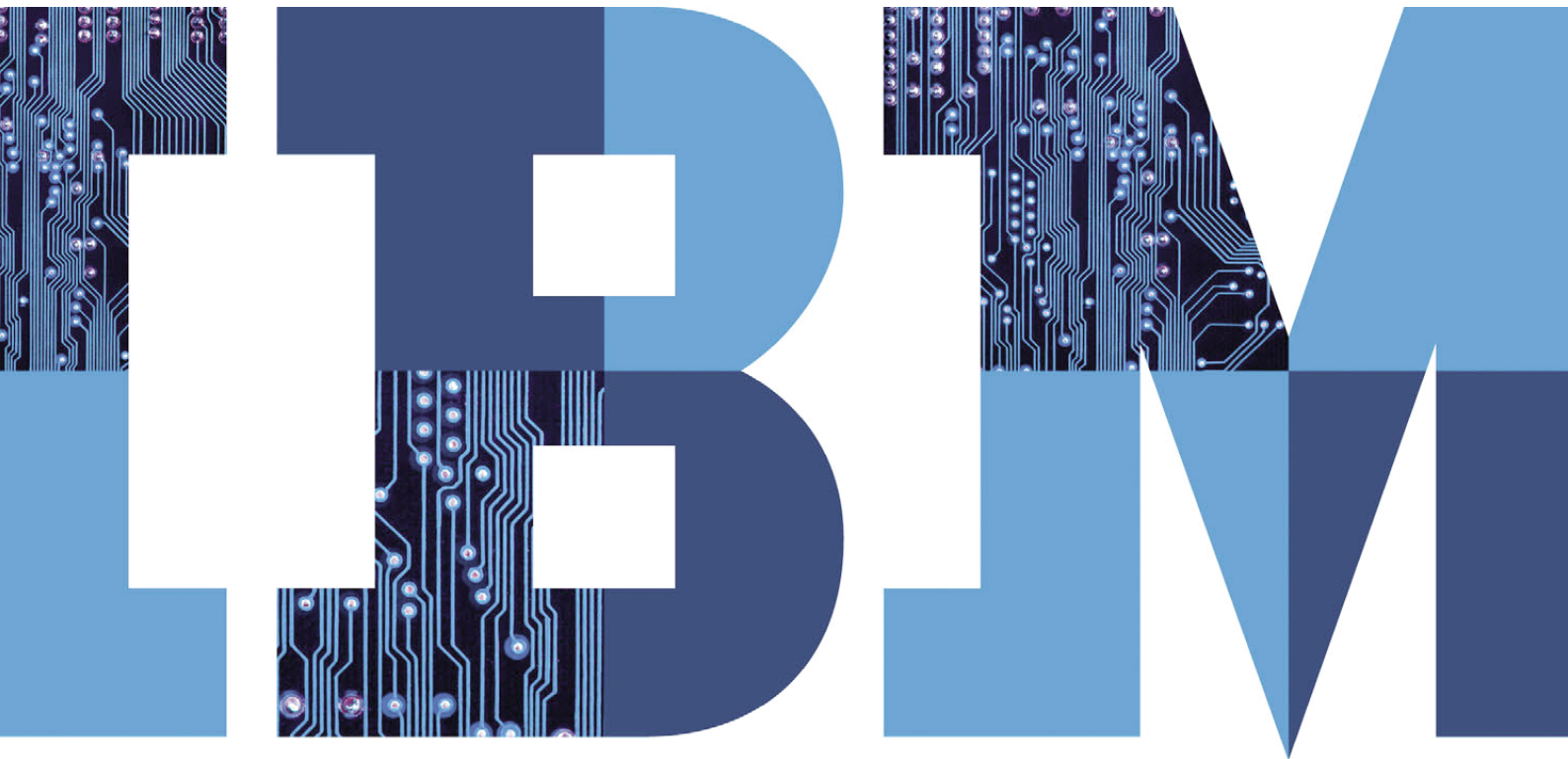


Des techniques de développement multicoeur de pointe pour la nouvelle génération de systèmes électroniques



Introduction

La miniaturisation reste le moteur de l'industrie électronique. Les dimensions des facteurs de forme des appareils et de leurs composants de base se réduisent au fur et à mesure de l'amélioration des capacités et des performances.

Les appareils électroniques deviennent aussi des éléments d'un écosystème fortement interconnecté du fait des liens de communication qui viennent le compléter. Les constructeurs de matériel électronique adoptent des processeurs multicoeur, un système de traitement composé de deux ou plusieurs cœurs indépendants,¹ pour faire profiter de ces progrès à leurs produits de nouvelle génération.

Avec l'expansion continue des processeurs multicoeur dans les matériels électroniques - PC, téléphones portables, jeux électroniques, équipement de réseau, systèmes de régulation industrielle et maintenant l'équipement médical – les exigences du développement logiciel pour ces systèmes sont de première importance dans l'industrie de l'électronique. Les avantages des architectures multicoeur sont nombreux : meilleure performance, une moindre consommation d'énergie, des coûts réduits et une plus grande flexibilité, mais ne sont véritablement dégagés qu'à la condition de développer le logiciel adapté. De nombreux développeurs de logiciel dans l'industrie de l'électronique actuelle manquent de compétences pour élaborer un logiciel optimisé pour le multicoeur. La complexité des architectures multicoeur peut aussi avoir une croissance exponentielle lorsque le nombre de cœurs passe de deux à quatre jusqu'à trente-deux cœurs ; autrement dit, les moyens traditionnels de développement ne peuvent plus évoluer. La seule façon de traiter la complexité consiste à automatiser le système et adopter un langage multi-niveaux qui permet d'automatiser le développement du logiciel sur différents noyaux. Le délai de commercialisation étant un facteur de différenciation critique dans l'industrie de l'électronique, tenir les délais limite de mise sur le marché passe par l'utilisation d'un code source logiciel existant dans l'organisation. Il s'agit toutefois, dans la plupart des cas, d'un code en langage source mono-cœur, non optimisé pour une technologie multicoeur. Ce Livre blanc IBM® Rational® traite des méthodes de développement de pointe des logiciels qui permettent à vos

équipes développement de tirer parti des nouvelles technologies multicoeur, ainsi que d'un système automatisé permettant de réutiliser le logiciel à noyau unique pour des systèmes multicoeur, ce qui vous permet à la fois d'être à la pointe de votre secteur et de réaliser des économies de temps et d'argent de façon drastique par réutilisation des systèmes existants.

Les perspectives offertes par le multicoeur

La tendance est à l'amélioration des performances et aux économies d'énergie, à la réduction des coûts sur les plates-formes informatiques utilisées pour fabriquer des produits électroniques de pointe. Il a été de tradition de pousser les processeurs pour améliorer les performances en augmentant le nombre de transistors sur une microplaquette et la fréquence de l'horloge, même en remettant en question la loi de Moore.² La tendance ne s'est pas inversée, mais pour faire un véritable bond en avant en termes de gestion des performances et de puissance, il faut innover dans le domaine des technologies multicoeur.

Les systèmes multicoeur sont économes en énergie et fonctionnent généralement à des fréquences d'horloge plus lentes, ce qui permet d'améliorer nettement la durée de vie des batteries et de réduire la production de chaleur. Les températures de fonctionnement des refroidisseurs signifient que les entreprises peuvent faire appel à des groupes refroidisseurs sans ventilateurs, plus silencieux, à meilleur rendement énergétique. Il est possible d'adopter des appareils dont les facteurs de formes sont plus petits, ce qui réduit le nombre de processeurs distincts en faveur d'un nombre réduit de processeurs multicoeur. Les environnements multicoeur offrent aussi la possibilité de disposer d'un véritable système de fonctionnement multitâche et d'améliorer très nettement les performances des applications par rapport à des environnements monocœur, qui peuvent atteindre très vite le seuil d'utilisation d'une unité centrale. Une véritable mise en parallèle est maintenant possible lorsqu'on gère une charge croissante d'applications à forte puissance. L'équilibrage des charges et la séparation de la fonctionnalité du système avec les technologies multicoeur permettent en outre d'améliorer la robustesse et la sécurité du système.

Des PC et des stations de travail aux smartphones et portables haut de gamme, la demande des utilisateurs pour des performances, une durée de vie des batteries en constante amélioration et des fonctions avancées a conduit les fabricants à créer et livrer des produits faisant appel à des technologies multicoeur. La prochaine génération d'appareils électroniques personnels et de produits plus intelligents fera appel à des environnements multicoeur pour offrir une connectivité, une réactivité, une facilité d'utilisation et des applications à forte productivité encore améliorées à une clientèle de plus en plus férue de nouvelles technologies. Le logiciel sera l'élément vital pour tenir les promesses du multicoeur, et la réussite ou l'échec repose visiblement sur l'équipe du logiciel embarqué.

Les défis à relever avec le multicoeur

Les promesses offertes par les technologies multicoeur sont synonymes de coûts et de risque importants pour les équipes développement actuelles. Les environnements multicoeur complexifient encore plus la conception et la fourniture de produits et les applications à forte concentration de logiciels qui les pilotent. Il faut faire des choix concernant la distribution des ressources et des fonctionnalités, affectant l'architecture des produits. Pour les concepteurs de systèmes, les décisions sur le nombre réel de processeurs, de noyaux requis, les caractéristiques du système d'exploitation (symétrique ou asymétrique) et l'intergiciel sont plus difficiles à prendre. Côté logiciel, les concepteurs doivent tenir compte des nouveaux problèmes relatifs au découpage des applications, à la communication entre noyaux et entre tâches, et l'évolutivité du modèle dans un système multicoeur. Les fonctions du(des) système(s) d'exploitation et l'intergiciel potentiels utilisés rendent encore plus complexe la conception du logiciel dans son ensemble.

Les entreprises qui passent à une technologie multicoeur vont devoir faire évoluer l'ensemble des compétences de leurs équipes de conception de logiciels. La mise en parallèle doit devenir la norme, il faut modifier la méthode de construction de leurs applications pour tirer pleinement parti de l'architecture parallèle disponible dans un environnement multicoeur. Les critères relatifs à une plus large utilisation de nombreuses tâches, des systèmes de

communication inter-tâches et l'affectation des tâches centrales deviennent des facteurs de réussite essentiels pour le développement de l'applicatif. Il importe de plus en plus de réutiliser toutes les tâches et tous les composants des systèmes à forte puissance logicielle.

Il faut faire appel à des outils de refactorisation qui réorganisent les applications existantes pour les faire tourner sur la nouvelle architecture multicoeur afin de tirer parti des améliorations des performances de base de l'architecture parallèle et, surtout, afin d'assurer leur bon fonctionnement. Ce qui a tourné sans problème et en toute sécurité sur un système monocoeur, même avec un système d'exploitation multitâches, pourrait s'avérer défectueux, voire dangereux dans l'architecture parallèle proposé par un système multicoeur.

Le débogage et les tests s'avèrent encore plus vitaux lors du développement d'un logiciel vers un système multicoeur. Les techniques traditionnelles de débogage au niveau des codes sources et d'essais en phase finale ne peuvent évoluer dans la plupart des cas vers des environnements multicoeur. De nouvelles techniques et de nouveaux environnements de tests s'avèrent nécessaires pour assurer le bon fonctionnement du logiciel et le niveau de fonctionnalité requis.

Les systèmes IBM Rational pour évoluer vers le multicoeur

Nous recommandons trois méthodes pour un développement efficace du système multicoeur :

- Etudier les options possibles pour évaluer les solutions alternatives ;
- Utiliser le logiciel existant pour accélérer le développement ;
- Automatiser la création du logiciel pour améliorer la qualité.

Etudier les options possibles pour évaluer les solutions alternatives

Réutiliser simplement un logiciel existant pour l'adapter à un processeur multicoeur peut en fait aboutir à le faire tourner à la même vitesse, voire plus lentement.³ Ceci est dû au fait que les noyaux actuels sont généralement plus lents, pris

individuellement, que celui du processeur d'origine. Cela se vérifie notamment par le fait que les données partagées peuvent ralentir la communication. Toute l'application pourrait même tourner à la vitesse d'un noyau unique. L'étude des différentes options possibles est essentielle pour évaluer des solutions alternatives et faire en sorte que l'on obtiendra les améliorations souhaitées en termes de performance, de vitesse et d'utilisation des ressources.

La modélisation de l'architecture logicielle de votre produit et la création de différentes fonctions afin qu'il puisse tourner sur plusieurs noyaux sont vitales pour le succès de sa mise en œuvre. Les tâches actuelles sont appliquées aux noyaux, auxquelles viennent s'ajouter des tâches supplémentaires, si nécessaire. La modélisation vous permet aussi de mieux appréhender la façon d'optimiser les communications entre les tâches et les opérations intégrées dans vos applications.

Une autre méthode clé consiste à simuler le modèle que vous avez créé afin de vérifier le bon fonctionnement de l'ensemble comme prévu. Sans cette simulation, il vous faut espérer que le logiciel fonctionnera.

Lors de l'étude des différentes options possibles, le modèle vous permet de documenter les raisons pour lesquelles vous avez pris certaines décisions. Il importe aussi de comparer votre modèle aux spécifications d'origine. Comprendre les aspects en temps réel des exigences vous permet de décider de la nature des tâches à mettre en correspondance avec chaque noyau. Par exemple, il est possible que vous n'avez pas besoin de prévoir l'interface utilisateur du logiciel sur les mêmes noyaux que les composants de contrôle en temps réel ou de contrôle critique.

Il importe de mener l'étude des différentes options possibles pour comprendre les fonctions de quel élément de votre logiciel doivent passer sur certains noyaux. A défaut, vous ne pourrez pas tout simplement évaluer les meilleures solutions alternatives et être certain du fonctionnement optimal de vos applications sur les noyaux.

Utiliser le logiciel actuel pour accélérer le développement

Votre produit actuel pourrait déjà très bien tourner avec le logiciel d'application actuel. Il tourne bien dans

l'environnement opérationnel à noyau unique actuel, et pourrait même tourner correctement dans le nouvel environnement multicœur. Il vous faut conserver sa valeur en le réutilisant autant que possible, et vous devez faire en sorte d'améliorer ses performances et le faire évoluer pour satisfaire aux exigences de son nouvel environnement.

Le développement pour un système multicœur s'appuyant sur une approche orientée-modèle offre d'importantes possibilités de réutilisation de votre logiciel actuel. De la simple visualisation de la structure et des relations du logiciel et de ses composants jusqu'à la refactorisation du code source pour optimiser le déploiement vers un environnement multicœur, à une intégration et une extension dans le cadre d'une conception et d'une architecture d'applications tout à fait nouvelles, l'ensemble d'outils approprié basé sur des modèles pourra nettement accélérer vos possibilités de réutiliser le logiciel qui a fait ses preuves dans votre entreprise.

La visualisation de votre logiciel d'applications actuel dégage des gains immédiats en vous permettant de mieux comprendre la façon dont sont structurés le logiciel et les interactions des composants. Elle vous permet d'identifier les critères optimaux et inférieurs à la norme de votre modèle et de générer la documentation du modèle actuel pour déterminer plus facilement la meilleure façon d'utiliser le code source dans le nouveau modèle, sans même devoir le changer. Ce code source visualisé peut être destiné au nouvel environnement multicœur et vous pourrez étudier les différentes options possibles pour décider du meilleur sort réservé à l'application, soit sa réutilisation telle quelle, soit sa refactorisation.

Si vous estimez finalement que le logiciel actuel ne tourne pas aussi bien dans le nouvel environnement ou que son architecture est insuffisante pour l'environnement multicœur, la refactorisation est une solution valable. En important le code source comme éléments modèles, il est possible de le refactoriser et le régénérer pour la prise en charge d'une architecture d'application de meilleure qualité, adaptée à la technologie multicœur. La modélisation et la génération automatique du code source permettent une refactorisation peu coûteuse du code source existant pour la réécriture de l'application.

Enfin, vous pouvez utiliser les composants du logiciel actuel sur de nouveaux modèles et de nouvelles applications créés pour les produits destinés à un environnement multicoeur. Ici également, les capacités de visualisation, de refactorisation et de génération automatique intégrées à un outil performant de développement basé sur des modèles renforcent votre capacité de décision pour leur réutilisation. Il est possible d'écrire de nouvelles applications destinées aux produits, ce qui réduit la nécessité de reprendre la conception de zéro. Les composants réutilisés peuvent aussi se prévaloir d'une bonne réussite avec les produits antérieurs, leurs comportements sont connus et testés, ce qui limite les tâches de nouveaux tests et de nouveau débogage par rapport à leur performance dans le cadre de la nouvelle application.

Avec le passage à un nouveau matériel, la réutilisation d'un code source récupéré est toujours déterminante, car trop laborieuse pour tout réécrire. En outre, disposer d'un code source déjà testé est vital pour l'exécution d'un projet dans les délais impartis. La visualisation graphique du code source réutilisable vous aide à comprendre la façon de l'adapter à la fonctionnalité que vous ajoutez lors du passage à une nouvelle plate-forme multicoeur.

Automatiser la production d'un logiciel pour améliorer la qualité

Du fait d'un environnement fortement parallèle inhérent à l'environnement multicoeur, les communications entre tâches dans une application et entre applications revêtent une importance capitale dans la réussite du développement de ces applications. Les tâches doivent communiquer dans la plupart des situations avec différents systèmes lorsqu'elles passent du noyau unique de résidence à différents noyaux. Cette forte dépendance à des communications performantes augmente l'intérêt de disposer de bonnes interfaces entre les tâches et d'utiliser un environnement de modélisation permettant de générer automatiquement les systèmes supports.

La compréhension, la spécification et l'exploration complètes des solutions impliquant le passage à un matériel et des protocoles de communication différents nécessitent

l'utilisation de modèles et la génération automatisée des codes source. Ceci permet de visualiser le modèle et la duplication de ce modèle dans le code source généré. Lors du transfert d'une tâche d'un noyau à un autre, il est possible de régénérer la modification du modèle et de le transcrire dans un code spécifique au nouveau noyau, si nécessaire, même s'il utilise un système d'exploitation différent ou s'il doit utiliser un protocole différent pour communiquer à des tâches sur le noyau d'origine. Ceci nécessite généralement un codage manuel du changement, mais pour une application modélisée, ce changement s'opère normalement par simple affectation des fonctions à l'aide des capacités de génération automatique d'un outil de développement basé sur un modèle prenant en compte le reste de la conversion. A chaque phase du développement, la modélisation et la génération du code source sont d'une importance vitale pour aider les développeurs à tester et développer leurs applications multicoeur. Ceci se vérifie notamment sur les marchés sensibles aux délais, comme l'industrie de l'électronique.

Conclusion

Vu la tendance à commercialiser des produits innovants et différenciés dans des délais plus courts, l'industrie de l'électronique dans son ensemble adoptera de plus en plus le système informatique multicoeur afin de tirer les bénéfices d'une meilleure performance, d'une plus faible consommation d'énergie, d'une plus longue durée de vie des batteries, d'une réduction des coûts et d'une plus grande souplesse. Mais ces retombées bénéfiques impliquent obligatoirement une architecture du logiciel correspondant adaptée à la technologie multicoeur. On peut utiliser plusieurs méthodes de pointe pour simplifier et accélérer le développement logiciel pour le multicoeur. Le développement basé sur des modèles permet aux concepteurs de logiciels de mener des études sur les différentes options possibles de la technologie multicoeur, de visualiser et refactoriser le code source pour assurer une meilleure réutilisation, et générer automatiquement le code source en fonction de la configuration du matériel et des protocoles de communication utilisés.

Pour en savoir plus...

Sur les techniques de développement multicoeur, contactez votre représentant ou votre chargé d'affaires IBM, ou visitez le site : ibm.com/software/rational/info/multicoeur/

Les solutions de financement proposées par IBM Global Financing permettent également d'assurer une gestion efficace de la trésorerie, de se protéger de l'obsolescence technologique, d'améliorer le coût total de propriété et le retour sur investissement. Nos Services Mondiaux de Reconstitution de l'actif financier permettent également de traiter les problèmes liés à l'environnement grâce à de nouvelles solutions plus économes en énergie. Pour de plus amples informations sur IBM Global Financing, visitez le site : ibm.com/financing.



© Copyright IBM Corporation 2011

Compagnie IBM France
17 Avenue de l'Europe
92 275 Bois-Colombes Cedex

Imprimé en France
Mars 2011
Tous droits réservés.

IBM, le logo IBM, ibm.com, et Rational sont des marques commerciales ou des marques déposées d'International Business Machines Corporation, aux Etats-Unis, dans d'autres pays, ou aussi bien aux Etats-Unis que dans d'autres pays. Si ces termes et d'autres termes IBM de marques commerciales sont signalés à leur première apparition dans la présente publication par un symbole de marque déposée (® ou ™), ces symboles indiquent des marques déposées ou des marques commerciales relevant du droit coutumier des Etats-Unis, qui sont la propriété d'IBM à la date de publication du présent document. Il est possible que ces marques commerciales soient aussi des marques déposées ou relevant du droit coutumier dans d'autres pays. Une liste actuelle des marques commerciales IBM est disponible sur Internet à la page « Informations sur les droits de reproduction et les marques commerciales » sur le site ibm.com/legal/copytrade.shtml

D'autres noms de sociétés, de produits ou de services peuvent être des marques commerciales ou des marques de service de tiers.

Les renvois dans la présente publication à des produits ou des services IBM n'impliquent pas l'intention d'IBM de les rendre accessibles à tous les pays où opère IBM.

Les informations contenues dans cette publication sont données uniquement à titre indicatif. En dépit de tous les efforts mis en œuvre pour vérifier l'exhaustivité et l'exactitude des informations contenues dans cette publication, elles sont fournies « telles que d'origine » sans garantie d'une quelconque nature, expresse ou implicite. Ces informations sont également basées sur les programmes et la stratégie produits actuels d'IBM, susceptibles d'être modifiés par IBM sans préavis. IBM ne sera pas tenu responsable du préjudice causé par l'utilisation ou lié à l'utilisation de ce document ou de tout autre documentation. Aucun élément contenu dans la présente publication n'est destiné, ni n'a pour effet de créer des garanties ou des représentations de la part d'IBM (ou de ses fournisseurs ou donneurs de licence), ou d'altérer les termes et conditions du contrat de licence en vigueur régissant l'utilisation des logiciels IBM.

¹ http://en.wikipedia.org/wiki/Multicoeur_processor

² http://en.wikipedia.org/wiki/Moore's_law

³ <http://www.forbes.com/2009/11/23/google-microsoft-programming-technology-cio-network-multicoeur-hardware.html>



Recyclable, merci de recycler