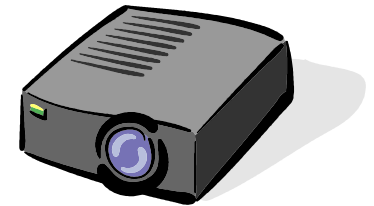


Présentation S.e.D.Re.

Les architectures 64 bits compatibles x86
Révolution ? Évolution ? Régression ?

Olivier DEMENGEON



Plan

- **Introduction**
 - Mémoire virtuelle (aperçu)
- L'architecture 32 bits x86
- L'architecture 64 bits x86
- “Quel est le bon choix ?”

Introduction

- La technologie 32 bits est présente dans les PC depuis les processeurs 386 (1985)
- Intel a développé un premier processeur 64 bits : l'Itanium (2001)
- AMD a développé des processeurs 64 bits compatible x86 : l'Athlon 64 et l'Opteron (2004)
- Intel sort des Pentium 4 et Xeon, comprenant les extensions 64 bits des processeurs AMD (2005)

Introduction

- L'Itanium et l'Itanium 2 ont connus de nombreux problèmes :
 - Incompatibilité x86
 - Ajout d'une émulation x86, en parti logicielle, d'où des problèmes de performances
 - Vitesse de l'exécution du code très très fortement liée au compilateur
 - Difficulté de monter en fréquences
 - Mauvais rapport prix / performance

Introduction

- Seront exclus de la présentation les Itanium et Itanium 2 :
 - Positionnés sur un segment très réduit du marché : les serveurs très haut de gamme
 - Incompatibles nativement avec les applications x86
- La présentation sera axée sur les processeurs compatible avec l'architecture x86 afin de pouvoir continuer à bénéficier des applications déjà existantes

Introduction

- Une présentation technique rapide sera effectuée afin de :
 - présenter les principaux problèmes auxquels nous pouvons être confrontés lors du passage d'une architecture 32 bits à une architecture 64 bits
 - fournir quelques conseils lors d'un choix de processeur en fonction des besoins exprimés

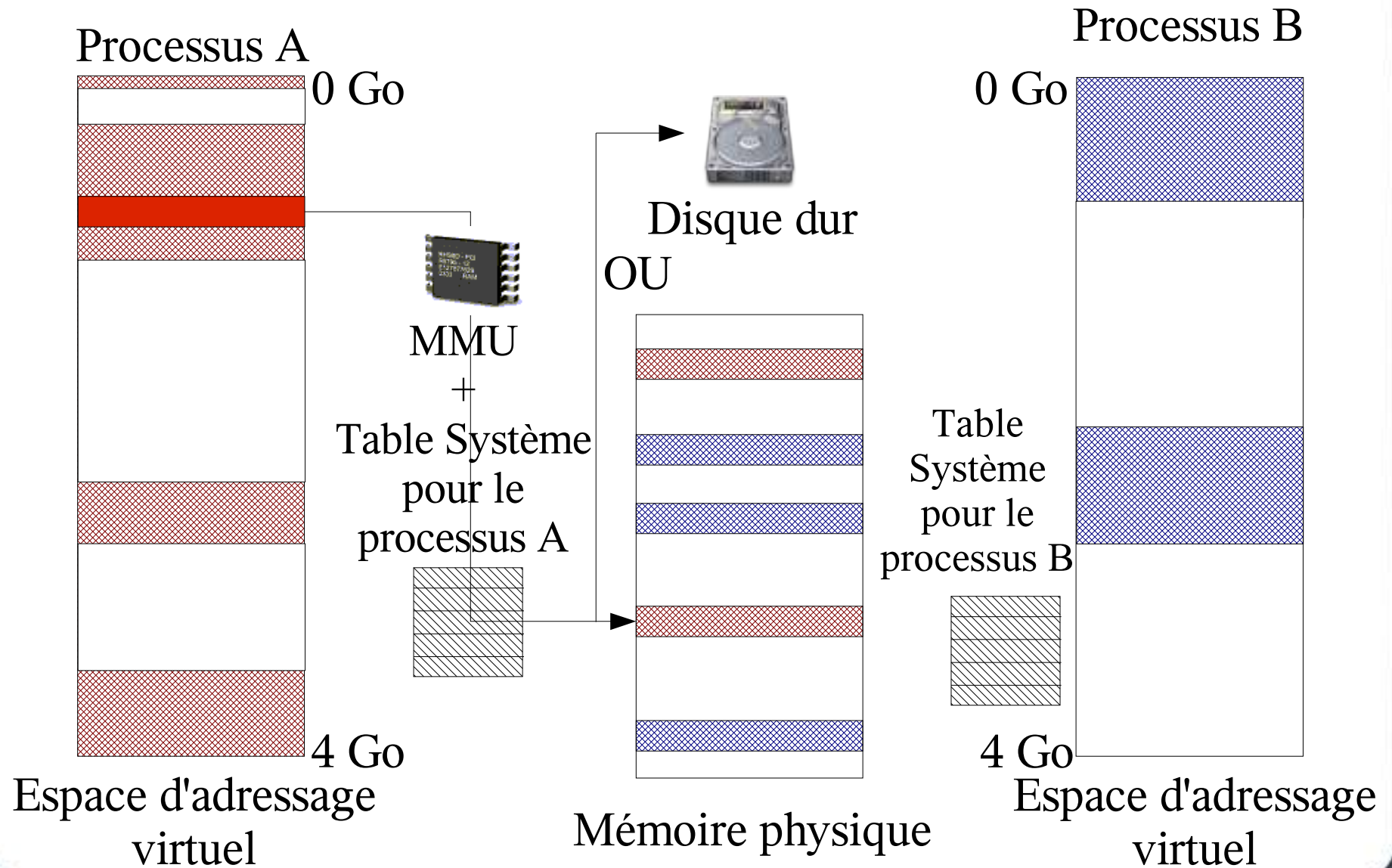
Mémoire virtuelle

- Les processeurs x86 intègrent la notion de mémoire virtuelle depuis le 286
- Elle est gérée conjointement par le processeur et par le système d'exploitation
- Les processus utilisent des adresses virtuelles qui sont traduites par le processeur (MMU : **M**emory **M**anagement **U**nit) avec l'aide du système d'exploitation :
 - en adresses physiques
 - en exceptions si le contenu de l'adresse virtuelle ne se trouve pas en mémoire physique

Mémoire virtuelle

- La traduction d'adresse virtuelle se fait à l'aide de tables gérées par le système :
 - Correspondance adresse virtuelle / adresse réelle ou swap
- Cela permet :
 - Une gestion facilitée de la mémoire pour les applications et une gestion simplifiée de la mémoire pour le système :
 - les applications ne s'occupent pas de la gestion de l'emplacement physique de son espace mémoire
 - D'isoler des processus en leur donnant à chacun un espace d'adressage virtuel différent

Mémoire virtuelle



Plan

- L'architecture 32 bits x86
 - Caractéristiques matérielles
 - Gestion de la mémoire
 - Présentation
 - Windows XP Professionnel
 - Linux
 - Extension de la mémoire physique supportée
 - PAE et périphériques 32 bits

Caractéristiques matérielles

- Adresses mémoires codées sur 32 bits
 - Espace d'adressage physique et virtuel maximum de 4 Go
- Taille des opérandes par défaut : 32 bits
- Registres :
 - 1 registre « compteur ordinal » sur 32 bits
 - 8 registres généraux de 32 / 16 / 8 bits
 - 6 registres de segment 16 bits
 - 8 registres XMMS 128 bits (dépend du model du processeur)
 - Problème lié à l'architecture x86 : faible nombre de registres

Gestion de la mémoire

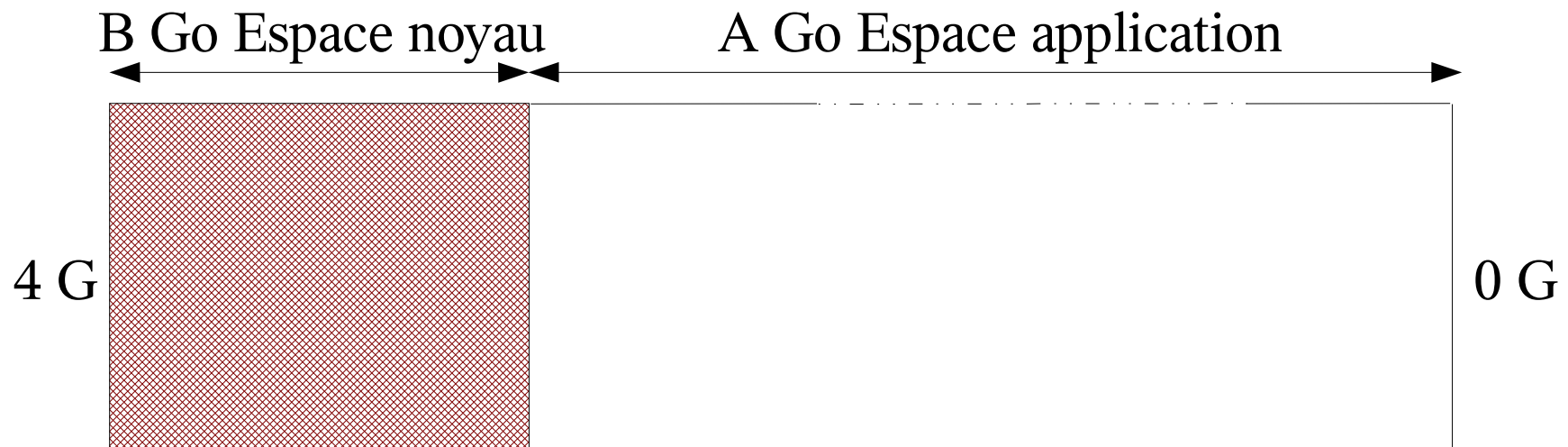
- Un processus a un espace d'adressage virtuel propre de 4 Go, isolé des autres processus
- L'espace d'adressage virtuel d'un processus fait parti de son contexte
- Pour passer d'un processus à un autre, il faut effectuer un changement de contexte
 - Les opérations de changement de contexte ont un coût en termes de cycles CPU

Gestion de la mémoire

- Les applications et le noyau des OS sont isolés pour des raisons de stabilité et de sécurité
- Les commutations entre une application et l'OS sont généralement assez nombreuses (spécifiques à chaque programme)
 - Pour des raisons de performance le noyau n'est pas isolé des applications par des espaces d'adressage différents
 - Généralement l'espace d'adressage du noyau de l'OS est inclus dans l'espace d'adressage de chaque processus

Gestion de la mémoire

- L'inclusion de l'espace d'adressage du noyau dans celui des processus :
 - Accélère les appels systèmes
 - Diminue la mémoire disponible pour l'applications
 - $4 \text{ Go} = A \text{ Go pour l'application} + B \text{ Go pour le système}$



Gestion de la mémoire

- L'isolement du noyau de l'OS et des applications est effectué par un mécanisme de droits s'appliquant aux différentes zones de la mémoire
- La gestion de la mémoire diffère selon les OS
- Les OS abordés par la suite seront Windows XP Professionnel et Linux 2.6

Gestion de la mémoire Windows XP Pro.

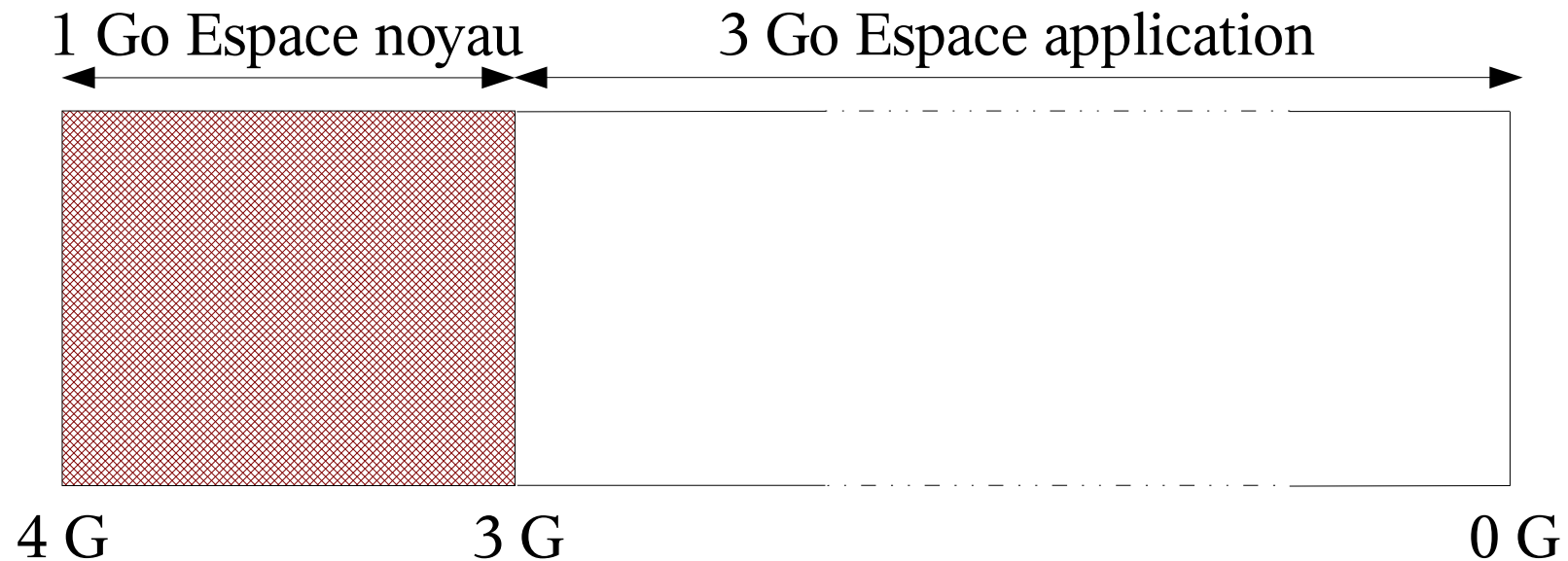
- Gère un espace d'adressage virtuel de 4 Go pour chaque processus, suivant les modèles :
 - 2 Go pour le processus / 2 Go pour l'OS, par défaut
 - 3 Go pour le processus / 1 Go pour l'OS, microsoft recommande l'utilisation de ce mode uniquement pour des tests sous XP Professionnel
- Peut gérer jusque 4 Go de mémoire physique

Gestion de la mémoire linux 2.6

- Gère un espace d'adressage virtuel de 4 Go pour chaque processus, suivant le modèle :
 - 3 Go pour le processus / 1 Go pour l'OS par défaut
 - Ce découpage est modifiable dans le code source
- Peut gérer jusque 64 Go de mémoire physique
 - La taille maximum de la mémoire physique gérée est fixée à la compilation lors du choix de modèle mémoire : 1 Go / 4 Go / 64 Go

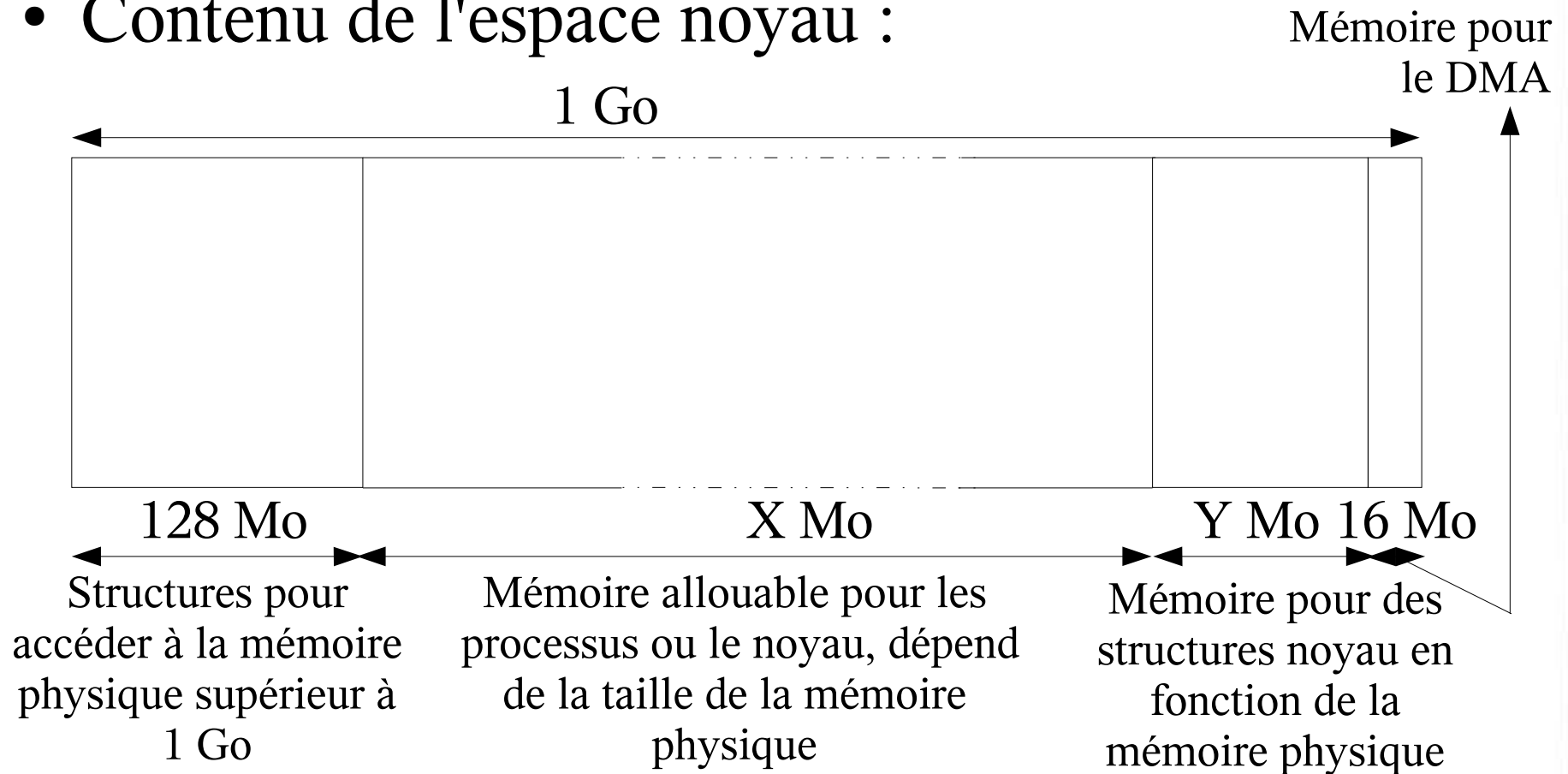
Gestion de la mémoire linux 2.6

- Gestion de l'espace d'adressage des processus :



Gestion de la mémoire linux 2.6

- Contenu de l'espace noyau :



Gestion de la mémoire linux 2.6

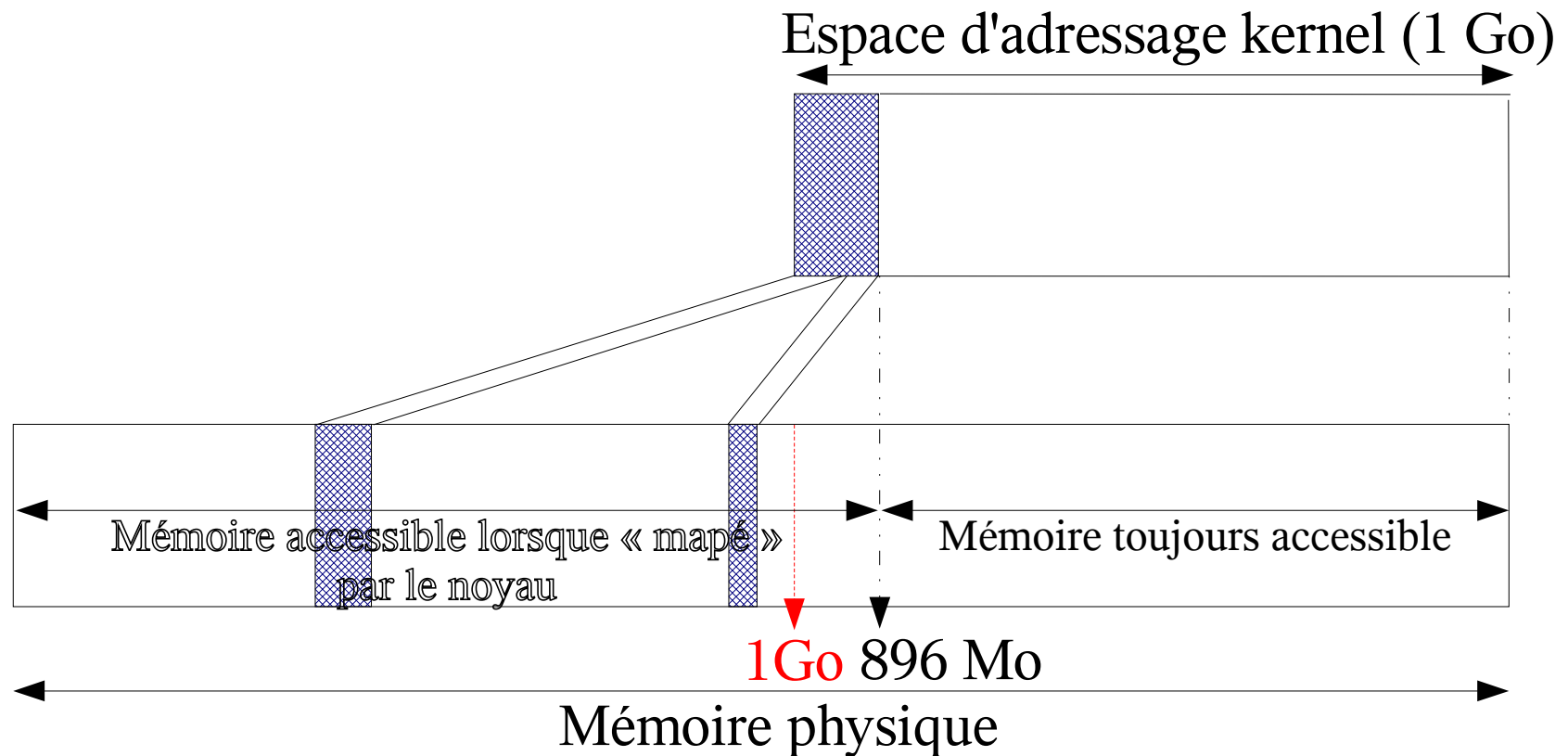
- Gestion de la mémoire physique :
 - L'espace noyau contenu dans chaque processus est “mapé” sur le premier giga-octet de mémoire physique
 - Le Noyau peut manipuler uniquement la mémoire physique présente au sein de son espace d'adressage
 - Lorsque le noyau est configuré avec le modèle mémoire 1 Go, seul 896 Mo de mémoire sont disponibles

Gestion de la mémoire linux 2.6

- Les modèles 4 Go et 64 Go permettent de gérer plus de mémoire physique :
 - Les 128 Mo de la fin de la zone de l'espace d'adressage noyau sont utilisés pour “maper” des zones de mémoires physique supérieur à 896 Mo
 - Ce mécanisme de pagination à un impact sur les performances
 - Certains anciens périphériques ne peuvent pas y accéder
 - Certaines données du noyau ne peuvent pas y être placées

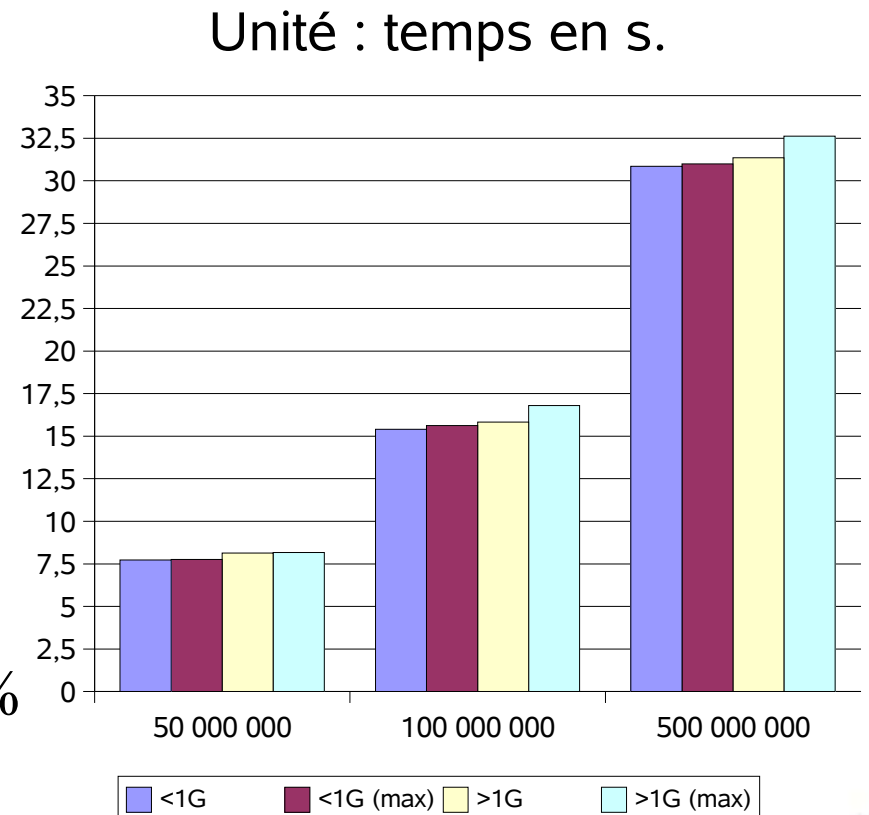
Gestion de la mémoire linux 2.6

- Accès à la mémoire au dessus de 1 Go :



Gestion de la mémoire linux 2.6

- Evaluation des performances lors de l'utilisation de la mémoire physique $>$ à 1 Go :
 - Opteron 250 (2,4 Ghz)
avec 6 Go de RAM ECC Registered
 - Ecriture aléatoire de données dans un tableau de 120000000 éléments de 4 octets (480 Mo)
 - Faible dégradation des performances d'environ 1% à 2%



Gestion de la mémoire linux 2.6 alternative

- Il existe un **patch noyau** permettant d'implémenter une autre méthode de gestion de la mémoire physique (solution retenue par la Fedora Core pour les machines ayant plus de 4 Go de RAM)
- Séparation de l'espace d'adressage des processus et du noyau, afin de fournir un espace d'adressage de 4 Go au noyau
 - Le noyau ayant un espace d'adressage de 4 Go, il peut alors gérer 4 Go de mémoire physique
 - Les processus disposent entièrement de leurs 4 Go d'espace d'adressage

Gestion de la mémoire linux 2.6 alternative

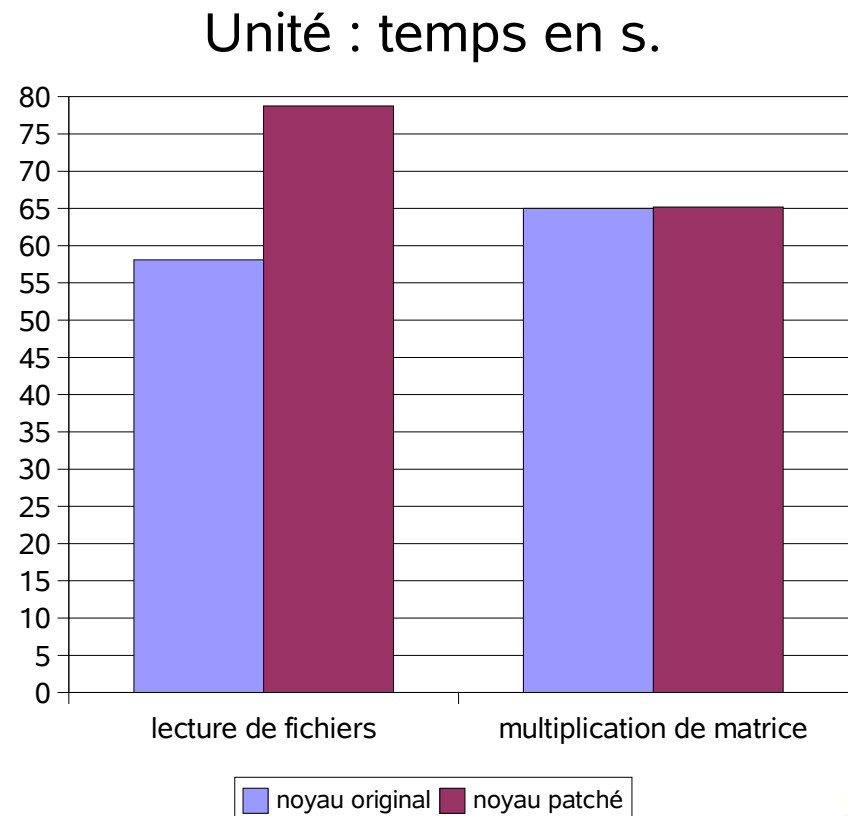
- Lorsqu'il y a plus de 4 Go de mémoire physique :
 - Le noyau pour accéder à la mémoire physique au dessus de 4 Go doit “mapper” une partie de cette mémoire dans son espace d'adressage
 - L'accès à la mémoire physique supérieure à 4 Go est plus lente à cause du coût de ce mécanisme de pagination
- Chaque appel système entraînant un changement de contexte => il y a une dégradation des performances

Gestion de la mémoire linux 2.6 alternative

- Impact de la séparation des espaces d'adressage des applications et du noyau :
 - Tests sur un Opteron 250 (2,4 Ghz)
 - Une application utilisant beaucoup d'appels systèmes : lecture du 1er octet de 10 000 000 de fichiers (boucle open, read, close)
 - Une application utilisant très peu d'appels systèmes : 10 multiplications de matrices d'entiers

Gestion de la mémoire linux 2.6 alternative

- Impact de la séparation des espaces d'adressage des applications et du noyau :
 - Les applications utilisant de nombreux appels systèmes sont pénalisées : dans les cas extrêmes une baisse de performance de 35% peut être constatée



Extension de la mémoire physique supportée

- La quantité de mémoire maximale gérée est apparue insuffisante
 - Ajout de l'extension **Physical Addressing Extension (PAE)**
- PAE supporté par la plupart des processeurs depuis le i686
 - Extension de l'adressage mémoire sur 36 bits
 - Le processeur peut gérer jusqu'à 64 Go

Extension de la mémoire physique supportée

- Les limites de PAE :
 - La mémoire rendue accessible par la PAE n'est utilisable directement que par le système d'exploitation
 - La taille de l'espace d'adressage des processus est toujours limitée à 4 Go
 - Utilisé pour conserver en mémoire les processus
 - L'utilisation de la PAE entraîne un coût au niveau des performances, le processeur ne peut accéder simultanément qu'à 4 Go (gestion du “mapping”)

PAE et périphériques 32 bits

- Les périphériques 32 bits ne peuvent adresser directement que les 4 premiers Go de la mémoire physique :
 - L'OS doit fournir des buffers intermédiaires dans la mémoire physique que peut adresser le périphérique

Plan

- L'architecture 64 bits x86
 - Matériels
 - Caractéristiques matérielles
 - Intel
 - AMD
 - Gestion de la mémoire
 - Windows XP 64 Professionnel
 - Linux 2.6
 - Compatibilité 32 bits x86
 - Performances
 - 32 bits vs 64 bits (OS 64)
 - 32 bits vs 32/64 bits vs 64 bits

Matériels

- Nous allons regarder les processeurs assurant une compatibilité avec l'existant :
 - Athlon (FX) 64 et Optéron
 - Pentium 4 et Xeon avec extension 64 bits (EM64T)
- La compatibilité 32 bits x86 :
 - Avantages : pouvoir utiliser l'existant
 - Utilisation de bios 32 bits
 - Utilisation d'OS et d'applications existantes
 - Inconvénient : hérite des problèmes des anciennes architectures
 - Par ex : mauvaises performances de l'unité de calcul flottant

Caractéristiques matérielles

- Adresse mémoire virtuelle et physique sur 64 bits maximum
 - Espace d'adressage de 16 Eo maximum
- Les processeurs actuels peuvent gérer jusque :
 - 256 To d'espace adressage virtuel
 - 1 To de mémoire physique
- La quantité de mémoire gérable dépend de l'implémentation du processeur :
 - Réduire la taille des divers caches processeurs (réduction de coût et de complexité)

Caractéristiques matérielles

- Taille des opérandes par défaut : 32 bits
- Modifications des registres :
 - Extension des registres 32 bits sur 64 bits
 - Ajoute de nouveaux registres afin de réduire les problèmes causé par leur nombre limité
 - 8 registres généraux
 - 8 registres XMMS
- Les processeurs supportent au minimum SSE/SSE2
 - En mode 64 bits, les compilateurs n'utilisent plus la FPU mais plutôt le SSE/SSE2

Caractéristiques matérielles INTEL

- Spécificités des processeurs INTEL 64 :
 - Les processeurs Intel intégrant la technologie EM64T sont très proches de leurs équivalant 32 bits (Pentium 4 ou XEON)
 - Support de Hyper-Threading, SSE3
 - Les fonctionnalités supportées par l'EM64T sont légèrement différentes de celles de l'AMD64, car elles ont été développées à partir d'un « draft » de l'architecture AMD64

Caractéristiques matérielles AMD

- Spécificités des processeurs AMD 64 :
 - Intégration du contrôleur mémoire
 - Modifier le type de mémoire revient à changer de processeur
 - Le type de mémoire utilisable est limité par le processeur
 - Les Opterons nécessitent de la mémoire Registered ECC
 - Intégration de contrôleurs HyperTransport
 - Bus de communication du processeur avec le contrôleur mémoire et les différents I/O
 - En SMP, chaque processeur est relié directement à une partie de la mémoire physique (architecture de type NUMA)
- Améliorations du contrôleur mémoire et ajout de SSE3 dans les derniers processeurs AMD

Gestion de la mémoire Windows XP 64 Professionnel

- Gère un espace d'adressage virtuel de 16 To pour chaque processus :
 - 8 To pour le processus / 8 To pour l'OS
- Peut gérer un maximum de 16 To de mémoire physique

Gestion de la mémoire linux 2.6 (64 bits)

- Gère un espace d'adressage virtuel de 256 To pour chaque processus :
 - 128 To pour le processus / 128 To pour l'OS
- Peut gérer un maximum de 128 To de mémoire physique

Compatibilité processeurs 32 bits

- Les processeurs ont 2 modes principaux de fonctionnement :
 - Mode hérité : fonctionne comme un processeur 32 bits actuel, pour les OS 32 bits
 - Mode long : fonctionne en mode 64 bits, pour les OS 64 bits, 2 sous-modes coexistent
 - Mode 64 bits : fait fonctionner les applications 64 bits
 - Mode Compatible : les applications 32/16 bits s'exécutent
- Les applications 32 bits ont pratiquement les mêmes restrictions que sur les architectures 32 bits x86

Compatibilité logiciel 32 bits

- Les OS 64 bits doivent implanter le passage entre le mode 64 bits et le mode compatible pour pouvoir exécuter les applications 16/32 bits
- Les OS 64 bits ne peuvent pas partager l'espace d'adressage mémoire des applications 32 bits
 - Les applications 32 bits disposent de l'intégralité de leur espace d'adressage, soit l'intégralité des 4 Go pour le processus

Compatibilité logiciel 32 bits

- Il est normalement impossible pour une application d'utiliser des bibliothèques 32 et 64 bits
 - Présenté comme une caractéristique pour isoler les nouvelles applications 64 bits des applications existantes

Incompatibilité logiciel 32 bits

- Certains OS 64 bits (ex: windows) ne sauvegardent plus les registres FPU / MMX / 3DNow lors des changements de contexte d'applications
 - Les programmes compilés en 64 bits utilisant ces registres doivent faire attention
- Les OS définissent certains types de données de manière différente :
 - Le type « long » est représenté sur 4 / 8 octets sous Windows / Linux

OS 64 bits et périphérique 32 bits

- Les périphériques 32 bits ne peuvent adresser directement que les 4 premiers Go de la mémoire physique
- Sur une plate-forme Intel :
 - L'OS doit fournir des buffers intermédiaires dans les 4 premiers Go de la mémoire physique
- Les processeurs AMD incluent une IOMMU :
 - Les périphériques manipulent des adresses virtuelles.
 - Elles sont traduites en adresses réelles par la IOMMU avec l'aide de tables gérées par le système
 - Les adresses virtuelles peuvent être “mapées” sur n'importe quelle zone de la mémoire physique

Performances

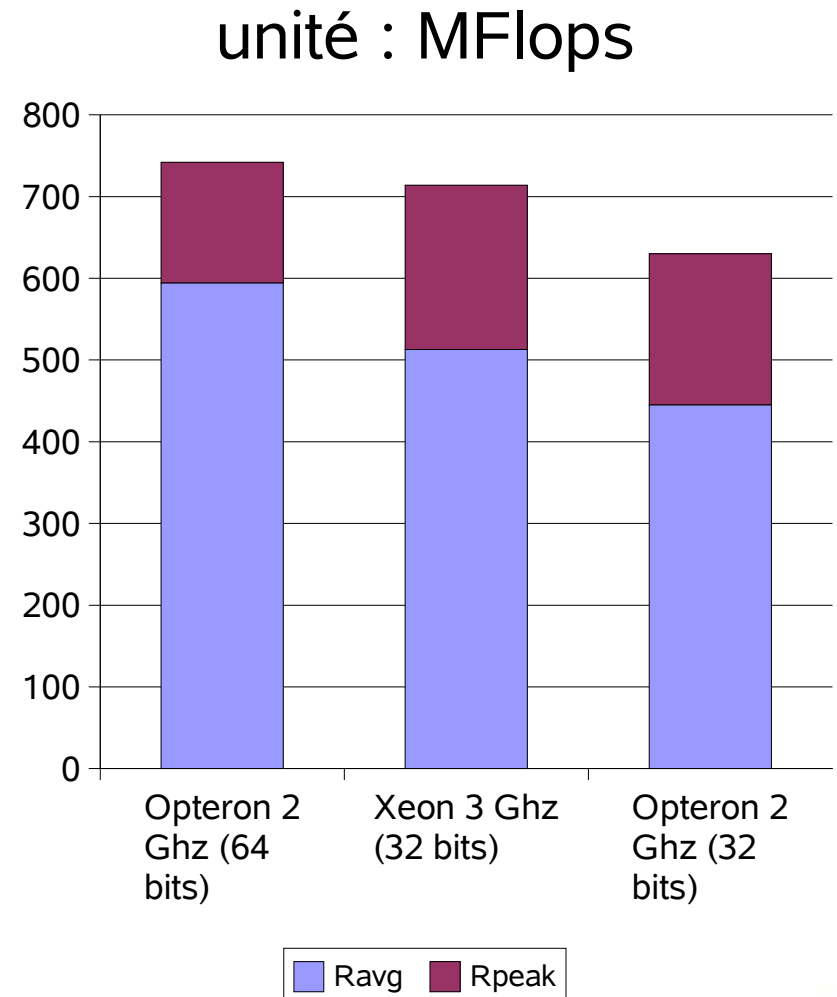
- Plate-forme de test :
 - Matériels :
 - Xeon 3 Ghz, 2 * 512 Mo DDR 400
 - Opteron 146 (2 Ghz), 2 * 512 Mo DDR 400 ECC Registered (ECC désactivé)
 - Logiciels :
 - Windows XP Professional x64 Edition RC1
 - Fedora Core 3 x86-64 avec gcc 3.4
 - Certains tests n'ont pas pu être effectués sur Xeon en 64 bits suite à un bug dans cette version de gcc, normalement corrigé dans gcc 4.0

Performances

- En théorie, principalement :
 - L'augmentation du nombre de registres généraux permet de diminuer le nombre d'accès aux caches ou à la mémoire. INTEL et AMD devrait obtenir des résultats équivalents
 - En mode 64 bits, les opérations sur les flottants sont effectuées par l'unité SSE / SSE2 et non plus par l'unité x87. Normalement avantage à INTEL qui avait des processeur 32 bits plus performants en SSE / SSE2 que AMD

Performances 32 bits vs 64 bits (OS 64) linux

- HPLinpack 1.0a
 - Calculs de matrices (BLAS/...)
 - L'Opteron en mode 64 s'en sort mieux : ceci peut s'expliquer par un nombre plus élevé de registre (diminution du nombre d'accès cache et mémoire)

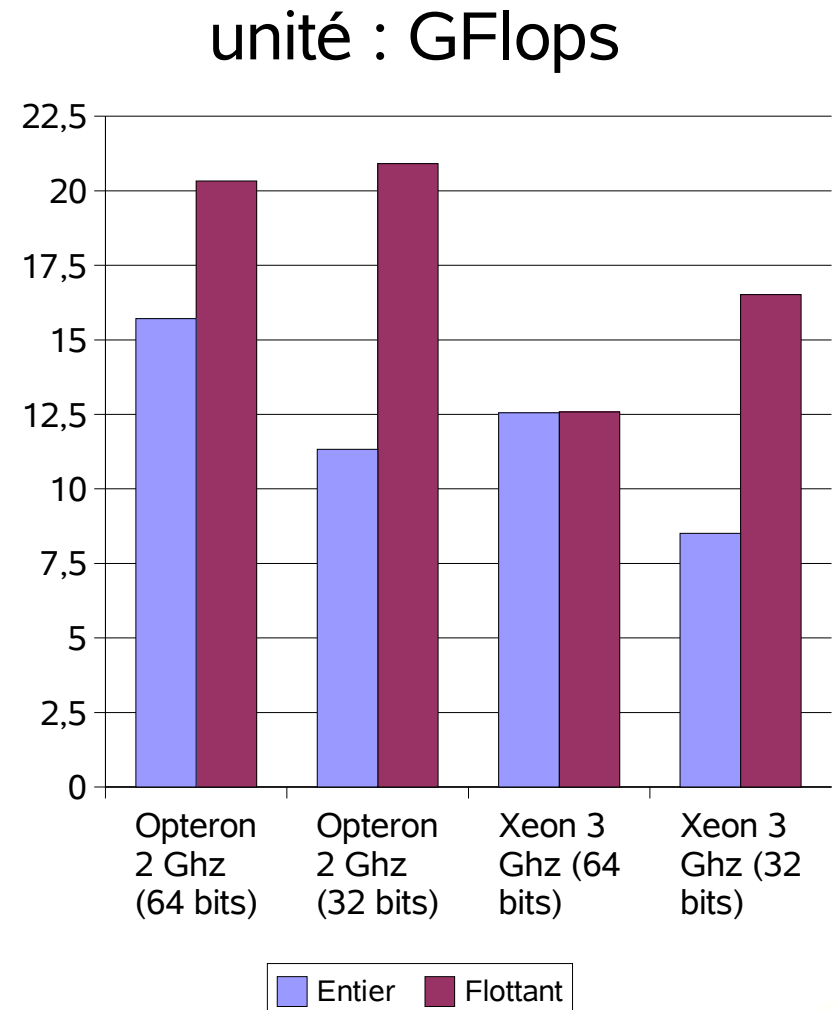


Performances 32 bits vs 64 bits (OS 64) linux

- HPLinpack 1.0a
 - L'application (32 bits) sur le Xeon est plus performante que sur l'Optéron, notamment grâce à l'utilisation du jeu d'instruction SSE3

Performances 32 bits vs 64 bits (OS 64) linux

- nBench 2.2.1
 - Ancien Benchmark de CPU et FPU
 - Le gain sur les calculs sur des entiers par les applications 64 bits s'explique généralement par l'augmentation du nombre de registres globaux



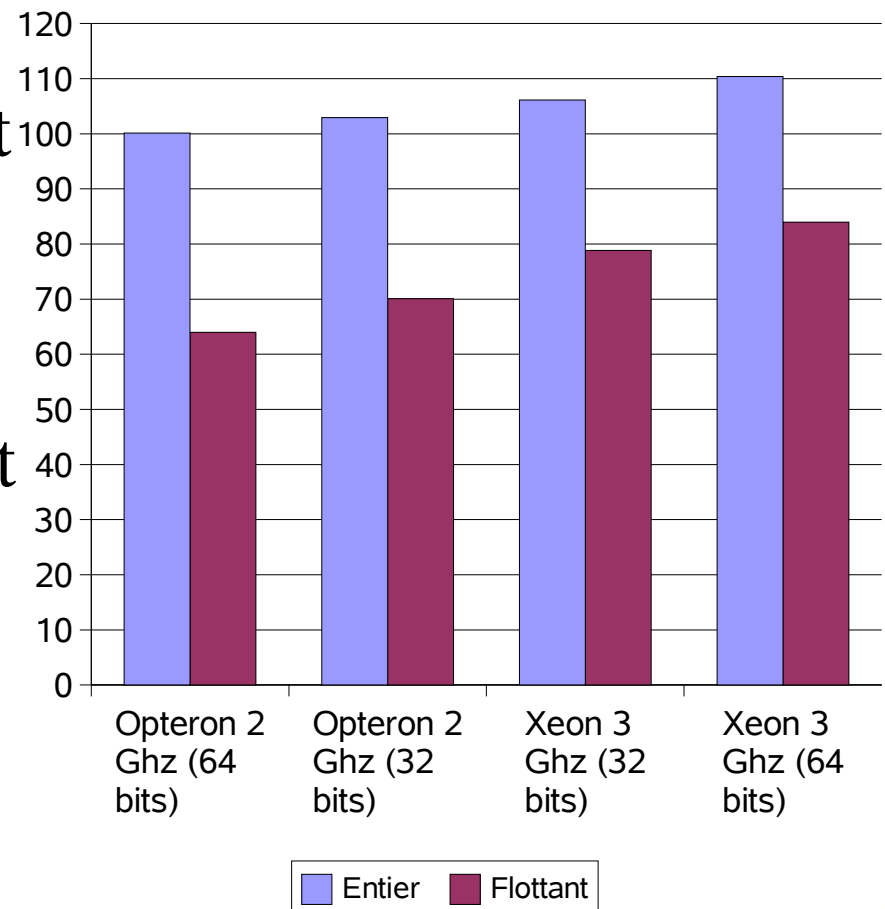
Performances 32 bits vs 64 bits (OS 64) linux

- nBench 2.2.1
 - L'application 64 bits est un peu en retrait sur le calcul flottant : le code original utilise la FPU x87 et le compilateur a encore quelques lacunes quant à la transformation de ce dernier en SSE / SSE2
 - La contre-performance sur les calculs sur des flottants pour les processeurs Xeon provient du compilateur qui a des problèmes avec le passage des registres FPU vers les registres SSE

Performances 32 bits vs 64 bits (OS 64) linux

- SS Bench 1.06b
 - Benchmark récent de CPU et FPU : le code est adapté aux processeurs 32 bits récents
 - La contre-performance, en 64 bits, du Xeon vient du compilateur qui est meilleur avec l'AMD64 que l'EM64T

unité : temps en s.

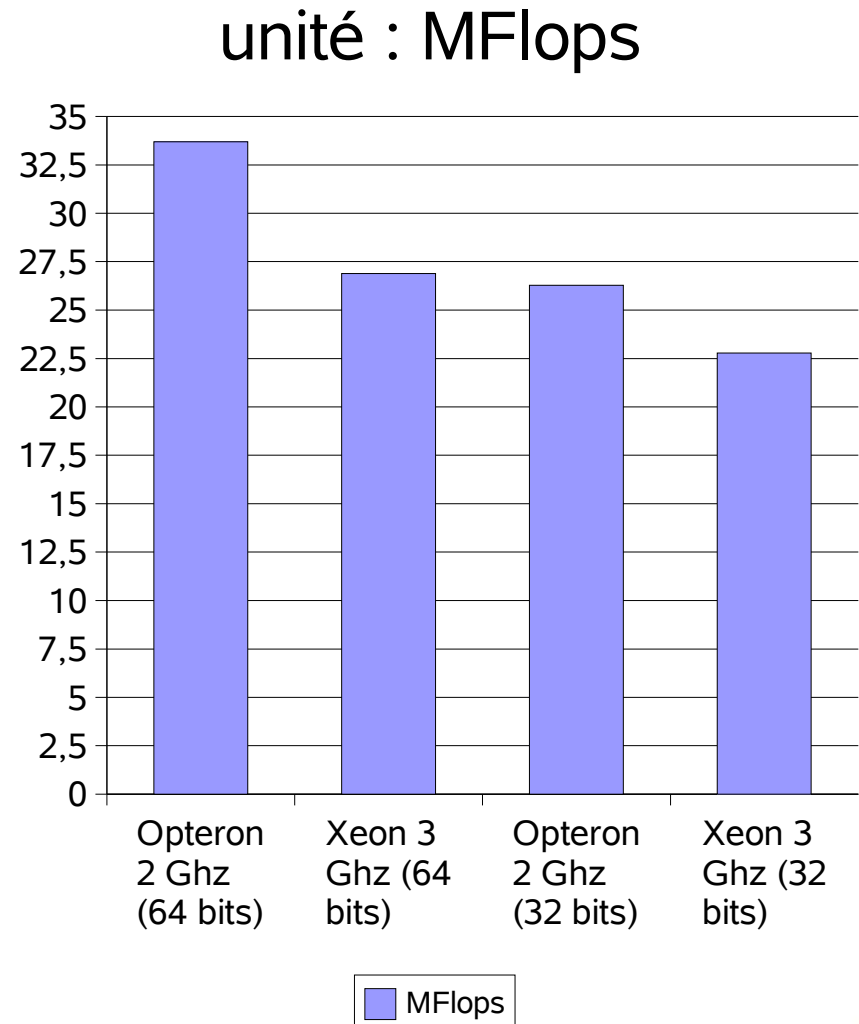


Performances 32 bits vs 64 bits (OS 64) linux

- SSBench 1.06b
 - Pour l'Opteron, l'application 64 bits a un léger gain de performance
 - Les algorithmes utilisés étant optimisés pour les architectures x86 32 bits, le compilateur n'arrive pas à exploiter pleinement les nouvelles ressources disponibles en mode 64 bits
 - Pour l'Opteron les performances sont identiques en mode 32 et 64 bits, le compilateur générant le code ne pouvant utiliser des registres supplémentaires

Performances 32 bits vs 64 bits (OS 64) windows

- Mandelbrot Benchmark
 - Calcul de fractal
 - Les modes 64 bits sont plus performants grâce à l'augmentation du nombre de registres SSE, l'algorithme s'y prêtant bien



Conclusion

Performances 32 bits vs 64 bits

- En pratique :
 - AMD semble plus performant qu'INTEL
 - Le mode 64 bits fait rarement ralentir les applications 32 bits sur la plate-forme AMD
 - Sur la plate-forme INTEL, les résultats sont plus mitigés
 - Ces différences semblent venir principalement du compilateur gcc, plus abouti en AMD64 qu'en EM64T puisque disponible depuis plus longtemps
 - Les résultats d'INTEL rappellent les premiers résultats obtenus par AMD avec du code compilé par les premières versions de gcc supportant l'AMD64

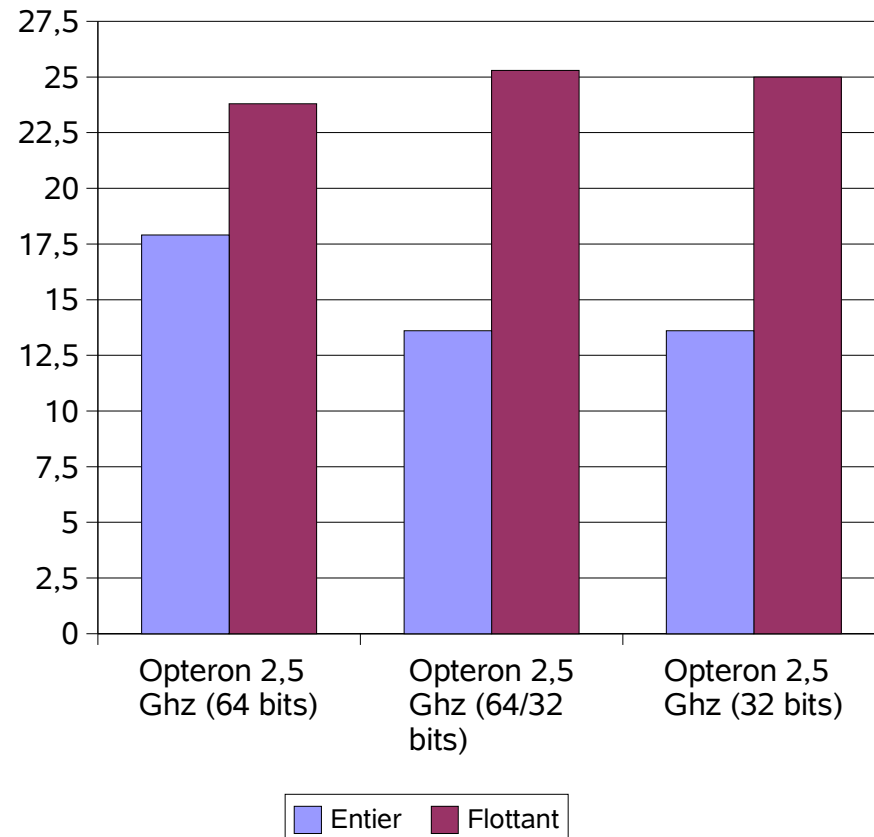
Performances des différents modes

- Plate-forme de test :
 - Matériel :
 - Opteron 250 (2,4 Ghz), 6Go ECC Registered
 - Logiciel :
 - Fedora Core 3 x86-64 avec gcc 3.4

Performances des différents modes

- nBench 2.2.1

unité : GFlops



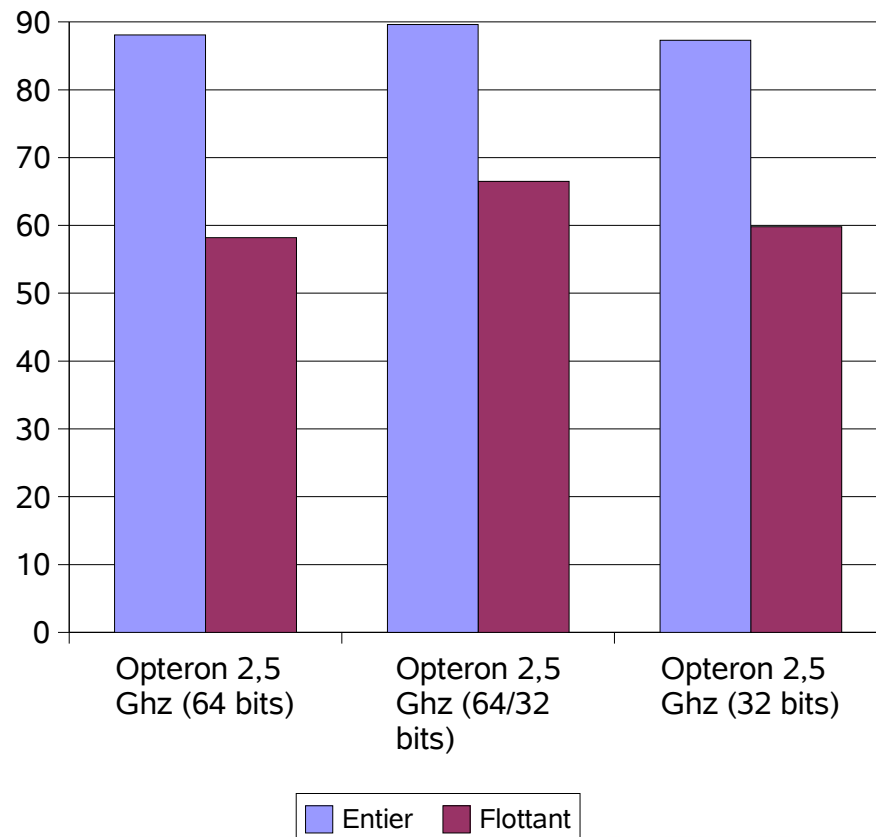
Performances des différents modes

- nBench 2.2.1
 - Les performances de l'application 32 bits sont à peu près comparable dans un OS 32 ou 64 bits
 - L'application 64 bits :
 - Plus rapide sur les calculs entiers car les algorithmes utilisés peuvent se servir des registres généraux supplémentaires
 - Un peu en retrait sur le calcul flottant : le code original utilise la FPU x87 et le compilateur a encore des lacunes quant à la transformation de ce dernier en SSE / SSE2

Performances des différents modes

- SS Bench 1.06b

unité : temps en s.



Performances des différents modes

- SS Bench 1.06b
 - L'application 64 bits a un léger gain de performance
 - Les algorithmes utilisés étant optimisés pour les architectures x86 32 bits, le compilateur n'arrive pas à exploiter pleinement les nouvelles ressources disponibles en mode 64 bits
 - L'application 32 bits dans un environnement 64 bits est légèrement en retrait :
 - Ce qui peut être expliqué par le coût du basculement entre le mode 64 et le mode compatible

Conclusion

Performances des différents modes

- Ces tests ont été effectués pour évaluer les performances du mode 64 bits par rapport au vrai “mode” 32 bits :
 - Les performances sur les calculs d'entiers sont dans le pire des cas équivalentes
 - Les performances sur les calculs de flottants peuvent dans certains cas subir une faible dégradation, généralement dûe au compilateur

Conclusion

Performance

- En pratique :
 - Au vue des résultats : **importance d'un bon compilateur**
 - gcc travaille bien avec les processeurs AMD64
 - le compilateur Intel doit utiliser au mieux leurs processeurs
 - Le 64 bits est utile dans les cas suivants :
 - Un ou plusieurs processus nécessitant une grande quantité de mémoire (plus de 3 Go)
 - Des programmes utilisant des algorithmes pouvant tirer partie de l'ajout de registres

Plan

- Introduction
- L'architecture 32 bits x86
- L'architecture 64 bits x86
- **“Quel est le bon choix ?”**

Processeur 32 bits linux

- Modèle mémoire 1 Go
 - Avantage : pas de perte de performance induite par la gestion mémoire
 - Restriction : seulement 896 Mo de mémoire physique utilisables
- Modèle mémoire 4 Go
 - Avantage : 4 Go de mémoire physique utilisable
 - Restrictions :
 - Quelques baisses de performance lors de l'accès à la mémoire physique > 1Go (environ 1%)
 - Certains vieux périphériques 32 bits ne peuvent accéder à la mémoire physique > 2 Go

Processeur 32 bits linux

- Modèle mémoire 4 Go / dissociation noyau - application
 - Avantage supplémentaire :
 - Les processus peuvent utiliser l'intégralité de leur espace d'adressage
 - Restriction supplémentaire :
 - Perte de performance pour des applications qui utilisent de nombreux appels systèmes (dans le pire des cas environ 35%)

Processeur 32 bits linux

- Modèle mémoire 64 Go
 - Avantage : 64 Go de mémoire physique utilisable
 - Utiliser pour du swap mémoire (plus rapide que du swap disque)
 - Restrictions :
 - Les processus ont toujours un espace d'adressage de 4 Go (max. utilisable 3Go)
 - Quelques baisses de performance lors de l'accès à la mémoire physique > 1Go
 - Les périphériques 32 bits ne peuvent pas accéder à la mémoire physique > 4 Go : l'OS doit gérer des buffers de transition (dégradation supplémentaire de performance)

Processeur 32 bits linux

- Modèle mémoire 64 Go / dissociation noyau - application
 - Avantage supplémentaire :
 - Les processus peuvent utiliser totalement leur espace d'adressage de 4 Go
 - Restrictions supplémentaires :
 - Perte de performance pour des applications qui utilisent de nombreux appels systèmes (dans le pire des cas environ 35%)
 - Les processus ont toujours un espace d'adressage de 4 Go

Processeur 64 bits linux

- Généralités

- Avantages :

- Compatibilité avec les applications / OS 32 bits existantes
 - Support de SSE/SSE2
 - Extension de l'architecture x86 avec l'ajout de registres globaux et SSE (réduction des accès « mémoire »)
 - Extension de la mémoire physique gérée ainsi que de l'espace d'adressage des processus

- Restrictions :

- Une très faible perte de performance (jusque 2%) possible lors de l'exécution de code 32 bits sur un OS 64 bits
 - L'utilisation de FPU / MMX / 3DNow fortement déconseillée
 - Choisir un bon compilateur

Processeur 64 bits linux

- AMD 64
 - Avantages :
 - Utilisation de périphérique 32 bits sans perte de performance (IOMMU)
 - L'architecture est connue depuis quelques années, les compilateurs ont eu le temps de l'intégrer
 - Restrictions :
 - Pas de support des bibliothèques et applications optimisées pour l'hyper-threading
 - Pas de compilateurs dédiés, peu de bibliothèques optimisées

Processeur 64 bits linux

- EM64T
 - Avantages :
 - Support des bibliothèques et applications optimisées pour l'hyper-threading
 - Des compilateurs et bibliothèques optimisés existent
 - Restrictions :
 - Quelques pertes de performances avec les périphériques 32 bits, l'OS doit utiliser des buffers de transition
 - L'architecture est moins connue que l'AMD64, les compilateurs libres (type gcc) sont moins performants actuellement

Windows : Que choisir ?

- Pour des applications ayant besoin de moins de 2 Go chacune et au total utilisant moins de 4 Go de mémoire physique :
 - un processeur 32 bits est à conseiller
- Dans les autres cas :
 - Pour une utilisation intensive de périphériques, la plate-forme AMD64 est conseillée
 - Utiliser une plate-forme adaptée au mieux à son application (bibliothèques spécifiques Intel -> Intel EM64T, bibliothèques spécifiques AMD -> AMD64)

Linux : Que choisir ?

- Pour des applications gourmandes en mémoire (plus de 4 Go au total), ou une application ayant besoin de plus de 3Go de données :
 - utilisant intensivement les périphériques, une solution à partir d'AMD 64 bits est conseillée.
 - effectuant de nombreux calculs, **si le compilateur Intel / Microsoft peut être utilisé**, une solution à partir d'Intel EM64T est conseillée, surtout si il est possible d'utiliser des bibliothèques optimisées pour ces processeurs
 - dans le cas d'une application voulant utiliser plus de 4 Go de mémoire, il est **impératif** sur une architecture 64 bits de **recompiler l'application en 64 bits**

Linux : Que choisir ?

- Si de nombreux processus consommant au total beaucoup de mémoire (plus de 4 Go), mais ne se servant intensivement que d'une petite partie de la mémoire :
 - utilisant peu les périphériques, une solution 32 bits peut être envisagée avec un modèle mémoire 64 Go
 - utilisant intensivement les périphériques, une solution à partir d'AMD 64 bits peut être envisagée

Linux : Que choisir ?

- Pour une ou des applications nécessitant plus de 1 Go et moins de 4 Go de mémoire :
 - si les applications respectent fortement la théorie de la localité, une solution 32 bits doit être envisagée avec comme modèle mémoire 4 Go
- Pour des applications où la performance est un critère et où les algorithmes peuvent tirer parti des ajouts de registres, avec pour les flottants un bon code SSE/SSE2, une plate-forme 64 bits semble être un bon choix. **Le code devra être 64 bits**

Fin

Les architectures 64 bits compatibles x86 sont
une évolution des processeurs actuels x86