LA MEMOIRE CACHE

Concept de base

La mémoire cache, appelée encore peu mémoire tampon est devenue en quelques années le compagnon indispensable des microprocesseurs. C'est même devenue, à tort, le principale critère de comparaison entre les processeurs.
Cependant il n'existe pas une mémoire cache mais plusieurs qui se différencient par leur architecture, leur taille ou encore leur vitesse de fonctionnement.

La mémoire cache a pour rôle l'accélération des communications entre un processeur et un composant servant à stocker les données (RAM, disque dur..).

En effet, pour calculer et exécuter ses instructions, un microprocesseur a besoin d'informations. Celles ci sont situées dans une unité de stockage. Or les microprocesseurs sont si performants qu'aucune unité de stockage n'est capable de fournir autant d'informations que le microprocesseur peut en traiter. La mémoire cache a pour fonction de palier à cette insuffisance.

Lorsque le microprocesseur a besoin d'une donnée , il regarde si elle est disponible dans la mémoire cache, si ce n'est pas le cas, il va la chercher dans l'unité de stockage et en même temps la dépose dans la mémoire cache. Ainsi la prochaine fois qu'il aura besoin de cette information, il y accédera directement par la mémoire cache et donc plus rapidement.

Attention, toute unité de stockage peut servir de mémoire cache, il suffit juste qu'elle soit plus rapide que l'unité de stockage principale. Voici 3 exemples :

Microprocesseur ---------- mémoire cache ----------RAM
Microprocesseur ---------- mémoire cache ----------Disque Dur
Microprocesseur ---------- mémoire cache ----------Internet

Dans le premier exemple, le mémoire cache se matérialise sous la forme d'une mémoire SRAM, dans le second exemple, une RAM traditionnelle fera l'affaire, dans le troisième cas, le disque dur fera office de mémoire cache à travers un fichier de pagination.

Les différents types de niveaux de cache

Une mémoire cache est une unité de stockage plus petite mais plus rapide qui s'interpose entre le microprocesseur et l'unité de stockage. Rien ne nous empêche de

répéter cette opération et d'ajouter une autre mémoire cache, même une troisième, etc……

On peut associer actuellement jusqu'à trois niveaux de cache (L1, L2, L3) entre un microprocesseur et la mémoire RAM.

Cache de Niveau 1 ou L1

Ce cache a été le premier à être mis en œuvre et a plusieurs particularités. D'une part, il est intégré au microprocesseur et sa taille est essentiellement fonction de l'architecture du microprocesseur. AMD a toujours privilégié des caches L1 de grande taille, 64ko pour le k6 et 128 ko pour les premiers ATHLON, tandis qu'INTEL privilégie des caches L1 de petite taille, 32 ko pour les PENTIUM 2/3. D'autre part, c'est un cache scindé en deux parties de taille égale. L'une stocke les instructions des programmes, l'autre les données des programmes. Les autres caches ne font pas cette distinction.

Cache de niveau 2 ou L2

Cette mémoire cache sert d'intermédiaire entre le cache L1 et la mémoire RAM. Il ne différencie pas données et programmes, il est moins rapide que le cache L1, mais sa taille est plus importante de 256 Ko à 2 Mo, voire plus pour les années à venir.

Ce cache a subit de nombreuses évolutions, et se retrouve aujourd'hui intégré dans le microprocesseur. Toutefois, s'il est intégré, il n'est pas imbriqué comme le cache L1. Cela veut dire que changer la taille du cache L1 implique souvent une modification de l'architecture du processeur, ce n'est pas le cas du cache L2.

Cache de niveau 3 ou L3

Si ce type de cache est courant sur des machines haut de gamme (Sun, HP, IBM, ALPHA,…), dans le monde des PC, il n'a existé qu'un seul exemple de microprocesseur utilisant une mémoire cache de niveau 3. Il s'agit du k6-3 d'AMD. Jusqu'à présent, ce type de cache a toujours été composé de mémoire SRAM, et implanté sur la carte mère. Sa taille varie de 1 Mo à 8 Mo.

L'organisation de la mémoire cache

Cache exclusif et inclusif

Cela désigne la manière dont vont coopérer les caches entre eux, en particulier les caches L1 et L2.

Le cache inclusif est la méthode la plus ancienne et la plus courante. Elle était utilisée sur tous les microprocesseurs jusqu'à l'arrivée des Duron et des Athlon Thunderbirds d'AMD. Lorsqu'une donnée part de la RAM vers la CPU, elle passe par le cache L2 puis par le cache L1. Une donnée peut donc être stockée, au même instant, à la fois dans le cache L1 et dans le cache L2. Il peut y avoir une redondance d'informations.

Le cache exclusif est apparu est les Duron et les Athlon Thunderbirds. Les caches L1 et L2 vont fonctionner comme s'il y'avait qu'un seul cache. Une donnée ne peut être au même instant dans les deux caches. Cependant, l'accès est plus long que dans le cadre des méthodes de cache inclusif. Cela est d'autant plus vrai que le cache est grand.

Les performances de la mémoire cache s'apprécient en fonction de la quantité de mémoire RAM que le cache pourra gérer, (avec 256 ko, on ne peut pas gérer efficacement voir pas du tout 4 Go de RAM), la rapidité avec laquelle le processeur pourra accéder à ces données, le % de chance qu'a le microprocesseur de trouver l'information dans le cache. Plus ce pourcentage est élevé, plus le traitement est rapide.

Pour optimiser la gestion de la mémoire en rapport à la taille de la RAM, plusieurs techniques se sont développés : le *« direct mapped »*, le *« N-way set associative »*, et le *« fully associative »*. Chaque technique apporte ses avantages et ces inconvénients. Le *« Direct Mapped »* est une technique simple qui peut privilégié les accès CPU – RAM plutôt que CPU – cache. L*e « N-way set associative »* résout ce problème mais demande un temps de lecture du cache et donc de l'accès au donnée plus long. Quant au *« fully associative »*, il oblige la CPU à lire toutes les lignes du cache pour savoir l'information y figure. Ce qui entraîne des dégradations de performances si l'on augmente la taille du cache.

Pour pouvoir comparer les performances respectives de chaque méthode, on utilise deux informations : Le ratio de réussite, rapport entre le nombre total d'accès au cache sur le nombre d'accès ayant permis de trouver l'information dans le cache. On exprime la valeur en %. C'est le nombre de chance qu'a le microprocesseur de trouver l'information dans le cache.

Plus le % est élevé, moins le processeur fait appel à la RAM, et les programmes fonctionnent plus rapidement. On considère les ratios suivants pour les méthodes de d'organisation du cache : *« Direct mapped »*, 60 à 80%, *« N-way associative »*, 80 à 90% et *« fully associative »*, 90-95%.

Cet indice est à prendre en compte avec le temps de latence du microprocesseur qui désigne le temps moyen que le microprocesseur met pour consulter les lignes de cache. C'est le temps que le microprocesseur met pour savoir si l'information est ou n'est pas dans le cache. On considère que le temps de latence est largement défavorable au

« fully associative », adaptée à des caches de petite taille. Le « Direct mapped » apparaît comme le cas idéal, surtout avec de faibles quantités de mémoire RAM. Le *« N-way associative »* dégage le meilleur compromis au niveau des performances c'est pourquoi il est aujourd'hui le plus employé.

La gestion du cache

Le microprocesseur utilise des méthodes bien particulière pour lire et écrire dans la mémoire cache. Pour lire les données, il n'utilise qu'une seule technique, il demande l'information à la mémoire cache, si elle ne la contient pas, il lit la RAM. Pour l'écriture des données, il peut utiliser le *« Write-trough cache »*, toutes les écritures des données allant de la CPU à la mémoire se font aussi dans le cache et le *« Write –back cache »*, toute les écriture se font dans la mémoire cache. Les données ne sont écrites en mémoire RAM qu'au moment où celles-ci ne sont plus utilisées par le microprocesseur et deviennent inutiles à conserver dans la mémoire cache.

La technique de *« Write-through cache »* est à privilégier car elle est une méthode qui garantit l'intégrité des données. En cas de dysfonctionnement de votre système, elle peut permettre la récupération des données au redémarrage de la machine.

Pour conclure, la mémoire cache permet d'augmenter la vitesse moyenne de communication entre un processeur et des composants de stockage comme la RAM ou un disque dur. On peut utiliser plusieurs cache en cascade. Celui qui est le plus près de la CPU est le plus rapide et le plus petit (sauf pour le Duron). On parle de cache L1, L2 etL3. Quand on dispose de plusieurs niveaux de cache, on peut les faire fonctionner de façon indépendante (inclusif) soit de concert (exclusif), comme s'il s'agissait d'un seul cache. Il existe trois méthodes d'organisation des informations dans le cache : *«Direct mapped »*où chaque ligne de cache correspond à un bloc déterminé de la RAM ; le *« fully associative »* où chaque ligne de cache peut gérer n'importe quel bloc de la RAM ; et le *« N-way associative »*, compromis des autres méthodes où l'on regroupe n lignes de caches pour les affecter à un bloc déterminé. Le fonctionnement normal entre une mémoire cache et un composant de stockage (RAM, disque dur) implique que toute écriture qui se fait dans le cache se fasse aussi en mémoire, c'est la méthode appelée *« write – through »*. Pour l'améliorer, on utilise la méthode *« write-back »* (écriture différée). Cependant elle ne garantit pas la cohérence du cache, c'est à dire que les données contenues dans le cache doivent correspondrent à celles situées dans le composant de stockage (RAM, disque dur). Dans ce cas, on dit que l'intégrité des données est préservée.