SpaceFox

Zeste de savoir

Le microprocesseur, ce monstre de puissance qui passe son temps à attendre

21 décembre 2018

Table des matières

1. Introduction

Avez-vous déjà remarqué à quel point le microprocesseur de votre ordinateur est un composant *extrêmement* puissant, et à quel point le moindre accès aux données est une horreur de lenteur de son point de vue ?

Pour essayer de se représenter tout ça, on va imaginer que *vous êtes* un cœur de microprocesseur, ralenti d'un facteur un milliard.

Or donc, vous êtes un cœur d'un microprocesseur moderne . Vous êtes avec d'autres collègues dans un *open space* ; votre boulot – et vous n'avez pas le choix – c'est d'exécuter des instructions, c'est-à-dire pour vous de taper au clavier, une touche par impulsion d'horloge[1]. Un processeur moderne peut être cadencé à 4 GHz, ce qui fait pour vous 4 frappes par seconde, ce qui est déjà rapide .

Et voici que vous avez besoin d'une information que vous n'avez pas. Zut, flûte, raalgamaziel, il vous faut la procurer.

Est-ce qu'elle est dans le cache L1 ? Vous prenez un peu plus d'une seconde *(un peu plus d'1 ns ou 4 instructions)* pour lire ce post-it sur votre bureau.

Non. Est-ce qu'elle est dans le cache L2 ? Vous prenez environ cinq secondes *(environ 5 ns ou 20 instructions)* pour parcourir la feuille à côté de votre clavier.

Toujours pas. Est-ce que par hasard elle serait dans le cache L3 ? Vous devez vous lever et prendre entre quinze et vingt secondes *(15 à 25 ns soit 60 à 100 instructions)* pour consulter ce schéma accroché à un mur.

Ça commence à puer cette histoire. Où diable peut se trouver cette information indispensable ?

Est-ce un résultat de calcul que doit vous donner un collègue *(donc un autre cœur)* ? Selon l'organisation de votre bureau *(la topologie du processeur)* et l'humeur dudit collègue, vous aurez la réponse dans entre quarante secondes et deux minutes vingt *(40 à 140 ns soit 160 à 560 instructions)*.

C'est aussi l'ordre de grandeur du temps qu'il va vous falloir patienter pour récupérer une information planquée dans les banques de mémoire du bureau d'à côté : une grosse minute à plus d'une minute et demie *(70 à 100 ns, soit 280 à 400 instructions)*. C'est déjà très long, même par rapport à votre schéma sur le mur du bureau.

Et si jamais vous n'avez *toujours pas* votre information, là c'est le drame. Parce que toute autre forme de stockage va être *vraiment* très lente à réagir par rapport à tout ce qu'on vient de voir.

Si vos données sont sur un SSD au format M2 , cas le plus favorable, vous en avez pour une petite journée (au moins 5h30) de recherche à la bibliothèque *(plus de 0.02 ms, soit plus de 20 000 ns… et donc 80 000 instructions[2])*.

Un SSD au format SATA est encore plus long, c'est comme si vos données étaient livrées par un coursier rapide : ça arrive vite, mais il faut quand même compter plus d'une journée et d'une nuit entières… *(à la louche 100 s, 400 000 instructions[3])*.

Et si votre information n'est toujours pas là… c'est la catastrophe.

S'il faut la faire venir depuis un disque dur mécanique ou par Internet depuis une connexion fibre, non seulement vous pouvez partir en vacances, mais en plus vous n'aurez jamais assez de congés payés pour patienter jusqu'à ce que votre donnée soit arrivée, puisqu'il vous faudra environ deux mois entre la demande et l'arrivée de ce dont vous avez besoin ! *(un ordre de grandeur de 5 ms… soit 5 000 000 ns et 20 000 000 instructions[4] !)*

Un chiffre que l'on peut tripler si la connexion Internet est une liaison cuivre ; la présence d'un WiFi dans le circuit permet probablement de concevoir intégralement un enfant.

(Tant qu'on est dans les comparaisons bizarres, si vous exécutez le code d'un jeu vidéo, vous avez un budget de 16 666 666 secondes soit plus de six mois pour le calcul d'une image pour avoir un rendu fluide. C'est énorme… tant que les données sont en mémoire, cf ci-dessus).

Mais la véritable situation catastrophique dans laquelle vous allez être véritablement bloqué sans rien faire, c'est si par malheur votre code attend la réaction d'un humain : s'il est surpris, il va mettre au moins une seconde réelle à réagir, soit un milliard de secondes pour vous, ce qui équivaut à… près de 32 ans. Vous affichez la popup en débutant votre premier boulot, l'utilisateur clique quand vous commencez à penser à votre retraite. S'il est réactif.

2. Conclusion

Et voilà qui remets à notre échelle des unités très variées (ms, ns, parfois s) que l'on utilise peu et donc appréhende mal surtout si elles sont mélangées ! J'espère ne pas m'être trop planté dans mes calculs, et que cette petite analogie filée vous aura permis de comprendre :

— La puissance réelle et affolante des microprocesseurs modernes ;
— Pourquoi on a plein de niveaux de cache, des *pipelines* et autres techniques pour que le processeur serve pendant les attentes, et pourquoi les SSD sont aussi agréables par rapport à des disques mécaniques ;

1. En réalité, un microprocesseur n'exécute *pas* une instruction par impulsion d'horloge, pour des tas de raisons assez compliquées. Mais pour les besoins de la démonstration, on va faire comme si.
2. C'est énorme : 80 000 instructions, si représentées par 80 000 frappes au clavier, c'est approximativement la taille de deux des nouvelles de cette page .
3. Votre vous-processeur aurait pu recopier *Alice au Pays des Merveilles*.
4. Avec toujours la même équivalence et un standard de 5 caractères par mot pour l'anglais, ça laisse le temps de copier plus de deux fois l'intégrale du Trône de Fer .

2. Conclusion

— Pourquoi c'est scandaleux que sur du matériel moderne on trouve des logiciels qui ne font rien de particulièrement complexe mais qui utilisent quand même tout le processeur.