

Parallel Task Force : le parallélisme, pourquoi, comment ?

Vincent C. LAFAGE

¹D2I, Institut de Physique Nucléaire Université d'Orsay







mardi 14 octobre 2014





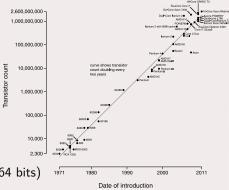
Microprocessor Transistor Counts 1971-2011 & Moore's Law

Loi de Moore depuis 1971 :

le nombre de transistors sur un circuit intégré double tous les deux ans

$$1971 \Rightarrow 10 \mu \text{m}, 2012 \Rightarrow 22 \text{nm}$$

- + de registres
- + de mémoire cache
- + d'instructions processeurs
- + de taille de bus (de 4 bits à 64 bits)
- + de gestion de la mémoire (MMU)
- + d'unités de calcul (un, plusieurs FPU, ALU vectorielle...)
- + d'étages de pipeline (superscalaires cf Pentium ca 1993)





... doublement de la fréquence tous les 2 ans, 2 ans et demi? Vrai depuis 1980 Plus depuis ≈ 2004 !!! Pentium 4 : 3 GHz... on attend entre 16 et 32 fois plus, > 50 à 100 GHz!!!

- $\mathcal{P} = C.V^2.f$
- 70–85 W \Rightarrow 2,5 kW!!!
- ⇒ Trop d'énergie à dissiper!
- ⇒ plusieurs processeurs sur un même circuit : plusieurs cœurs



- ⇒ pour continuer à bénéficier de la puissance croissante de la technologie, pas d'autre solutions que la programmation parallèle : coordonner le travail des différents cœurs
- ⇒ alors qu'on nous a toujours enseigné l'aspect essentiellement séquentiel de la programmation!



nouvelles technologies : part croissante d'une énergie rare & chère. Consommation CC-IN2P3 ; 1,6 MW \to 1,2 MW budget électrique total 0,864 M€

- évolution vers un meilleur ratio W/MIPS ou W/MFLOPS : Intel XScale, 600 MHz, 0,5 W
 5 fois plus lent, 80 fois plus économe!
- diminuer la fréquence, vu qu'il y aura plus de coeurs



Comment paralléliser?

- c'est un paradigme de programmation (concurrence) complémentaire des approches classiques : procédural, modulaire, générique, objet.
- ... et qui s'appuie sur différentes technologies matérielles également à différentes échelles
 - pipeline (DSP)
 - vectorisation (SIMD)
 - parallèlisation par processus (lourds) assez indépendants
 - parallèlisation par processus légers (threads = fils) partageant l'espace d'un même processus : économie de mémoire
 - distribution (parallèlisation par processus tournant sur des machines différentes)











S2I :: groupe Parallèle

Christophe DIARRA Grille, MPI, OpenMP

lvana $\operatorname{HRIVNACOVA}$ C++, ROOT / Geant4 multithread

Luz GUEVARA OpenMP, GPU

Vincent LAFAGE Fortran, optimisation calcul numérique



Success story

- Code de capture d'électrons
 9 kSLOC de Fortran, 5 kSLOC de C(++)
- Accélération avant //ⁿ
 - ... facteur 12 (stockage plutôt que recalcul)
 - ... facteur >50 au total
 - utiliser les bibliothèques standard (nearbyint)
 - Spherical Bessel Benchmark (15 codes, 9 algos)
 - vectorisation des boucles...
- amélioration de la précision



Méthode

Ces résultats sont le fruit d'une méthode :

- ⇒ on ne rentre pas dans 15 000 lignes de code comme dans un moulin : mise du code sous controle de version syn
 - analyse statique ftncheck, cppchecker
 - branches mortes (procédures, variables)
 - * métriques sloccount
- analyse dynamique "profiling"
 - * identification des goulets d'étranglements
 - * optimisation de fond de boucles
- chasse aux problèmes de mémoire
- typographie, indentation, documentation
- pêche aux mauvaises pratiques numériques
 - D. GOLDBERG, What every computer scientist should know about floating point arithmetic
 - * constantification des constantes : extraction des constantes en dur, uniformisation : combien de valeurs de π distinctes?
 - * accelération (Horner, Richardson, stockage intermédiaire...)



gprof

valgrind

doxygen



Conclusion

- + beaucoup de perspectives
 - pour les anciens codes...
 - . . .et pour les nouveaux
- + beaucoup de solutions
- pas si évident
- ⇒ offre de conseil
 - ! d'abord, optimiser

Sans programmation parallèle, aucun homme n'est jamais assez fort pour ce calcul

lafage@ipno.in2p3.fr

2: 53138