

Performances de la programmation multi-thread

Laurent Philippe

`laurent.philippe@lifc.univ-fcomte.fr`

Contexte

La notion de thread a fait son apparition dans les systèmes d'exploitation avec les premiers ordinateurs à mémoire partagée. L'objectif était d'utiliser de manière efficace l'ensemble des processeurs dans le cadre d'une même application. Le coût de ces machines à mémoire partagée est longtemps resté élevé ce qui a limité leur utilisation à des contextes nécessitant une optimisation des temps de calcul, comme la programmation de simulations scientifiques. A l'inverse, la programmation multi-thread a connu un large essor, même sur les ordinateurs mono-processeur, en raison de la simplification qu'elle apportait à la programmation interactive ou multi-tâches. Cette notion a même été intégrée dans le cœur même des processeurs avec les technologies d'hyper-threading. Il existe maintenant de nombreuses bibliothèques de programmation multi-thread, utilisables dans différents langages et proposant un support au développement. Les dernières évolutions des architectures de processeurs multi-cœurs et la baisse du coût des machines à mémoire partagée, maintenant largement utilisées pour les serveurs, font qu'il est maintenant difficile de passer à côté de ce type de programmation pour des applications destinées au service.

Sujet

Le but de ce projet est de faire le bilan des technologies de programmation multi-thread et des outils de synchronisation qui les accompagnent puis de faire une étude de performances sur différentes plates-formes dans le but de comprendre la portée de la notion de thread.