LSINF1252 - Factorisation de nombres

MONNOYER Charles et PARIS Antoine

19 avril 2015

1 Architecture globale

Pour structurer une application qui réalise des calculs, il est courant d'utiliser des producteurs/consommateurs[1]. Dans notre cas, les producteurs seraient chargé d'extraire les nombres à factoriser des fichiers passés en ligne de commande et de les placer dans le buffer. Les consommateurs seraint quant à eux chargé de factoriser les nombres contenus dans le buffer et de sauvegarder le résulat dans une structure de données adéquates.

2 Threads utilisés

3 Mécanismes de synchronisation

4 Principale structures de données

5 Algorithme de factorisation

L'algorithme que nous avons décidé d'implémenter pour ce projet est un algorithme à but général (c'est à dire dont le temps d'éxécution dépend de la taille du nombre à factoriser, et non de la taille de ces facteurs premiers). Il s'agit du *Shanks's square forms factorization algorithm (SQUFOF)*. Nous avons choisis cet algorithme car il possède une bonne complexité temporelle ($\sqrt[4]{n}$, où n est le nombre à factorisé) tout en étant facile à implémenter. Deux contraintes importantes sont cependant à noter, cet algorithme ne fonctionne pas si son entrée n est un carré parfait ou un nombre premier. Cependant, cela ne posera pas problème en pratique. En effet, dans le cas où n est un carré parfait, il suffit de donner $\sqrt{n}$ en entrée à l'algorithme [1] Dans le cas où n est un nombre premier, l'algorithme est inutile et il peut simplement retourner n.

Références

[1] O. Bonaventure G. Detal C. Paasch. *SINF1252 : Systèmes informatiques*. EPL, 2014.

1. Les facteurs premiers de n sont identiques aux facteurs premiers de $\sqrt{n}$.