Travail sur Work Stealing avec Fred, Denis, Nicolas

September 1, 2018

Abstract

Dans ce document, on va détailler les différents travaux sur le work stealing (ce qu'on a, ce qu'on a fait, ce qu'il faut finir, et ce qu'on va faire) le but de ce document est de clarifier les choses et de savoir qu'est-ce qu'on peut faire avec les données qu'on a et ce qu'on peut faire.

1 Work Stealing avec Communication

Le défi de l'algorithme work stealing sur les plates-formes distribuées est de prendre en considération le temps de communication pour transfert les tâches après un vol réussit entre deux processeurs (voleur et victime).

Coût de communication en tant que latence: le cas où on considère la latence permet de donner le même coût pour toutes les communications (pour envoyer une demande de travail, pour envoyer une réponse vide ou envoyer un ensemble de tâches).

Coût de communication en tant que Band passante: Dans le cas où on considère la Band passante, le coût communication entre deux processeurs s'influence par deux facteurs : 1-) la quantité de données transférées entre les deux processeurs, 2-) la charge qui existe dans le canal de communication qui lie les deux processeurs concernés.

Le cas de la band passante pose la question suivante : quelle est la topologie de communication qu'on va utiliser? : (un seul canal de communication entre tous les processeurs ? un canal entre chaque deux processeurs? ou quoi ?)

2 Notre simulateur

Pour les simulations, j'ai développé avec Fred un simulateur paramétrable, qui peut exécutier un ensemble de taches sur un ensemble de machines suivants différents algorithmes. Je vais détailler dans cette partie les fonctionnalités et les limites de notre simulateur :

les tâches: le simulateur peut prendre en entrée une quantité de travail W, cette quantité peut être gérée comme un ensemble de tâches indépendantes. Le simulateur peut aussi prendre un seuille avec W pour créer un arbre de tâches (les feuilles sont des tâches réelles dont la taille égale à peu pès au seuille et les nodes sont des taches de taille 0).

Le simulateur peut aussi prendre comme entrée un fichier JSON qui contient des arbres réels avec toutes les informations sur les tâches et les précédences.

les processeurs: Le simulateur permet de créer un ensemble de processeurs dans deux topologies:

1. Un seul cluster de p processeurs ou la communication égale la latence donner par l'utilisateur.
2. Deux clusters avec le même nombre de processeurs ou la communication à l'intérieur et à l'extérieur sont configurables.

les communication: Le simulateur ne prend en compte pour le moment que la latence dans les Communication, dans les deux version un et deux clusters. **la version qui prend en compte la band passante neccesit une discussion pour definir l'architecture et les differents régles**

l'algorithme WS: Le simulateur utilise le work stealing classique et plusieurs paramètres sont configurables :

1. REMOTE STEAL PROBABILITY
2. LOCAL GRANULARITY
3. REMOTE GRANULARITY
4. TASK THRESHOLD
5. SIMULTANEOUSLY STEAL

3 les premier travaux : sur 1 cluster

4 en cours : exp sur 2 cluster

5 les idées :