

Projet Programmation Parallèle

1 Contexte

On s'intéresse à la résolution du problème des pigeons en logique. Il s'agit d'un problème d'affectation qui s'annonce ainsi :

Etant donné n pigeons et q pigeonniers, chaque pigeon devant trouver un pigeonnier et un pigeonnier ne pouvant accueillir qu'au maximum un seul pigeon, restera-t-il des pigeons sans pigeonnier ?

Ce problème est très simple à résoudre puisqu'il suffit que $n \leq q$ pour qu'il ait une solution et que toute permutation d'une solution est également une solution. Dans le cas contraire ($n > q$) le problème n'admet pas de solution, on dit qu'il est insatisfiable.

Pour le résoudre on peut le modéliser en calcul propositionnel sous forme de contraintes (également appelées opérateurs) de cardinalité (α, β, V) qui signifie qu'au minimum α et au plus β variables de V sont vraies.

Le problème des pigeons s'exprime alors par les ensembles de contraintes suivantes :

- contraintes sur les pigeons (C1) : un pigeon est dans un et un seul pigeonnier
- contraintes sur les pigeonniers (C2) : un pigeonnier accueille au plus un pigeon

Si on modélise le problème par une matrice de variables booléennes $X(n, q)$, telles que si x_i^j est vraie cela signifie que le pigeon i est dans le pigeonnier j . On obtient alors :

- un pigeon est dans un et un seul pigeonnier (C1) :

$$n \text{ contraintes } (1, 1) \quad \left\{ \begin{array}{l} (1, 1, x_1^1, \dots, x_1^q) \\ \dots \\ (1, 1, x_n^1, \dots, x_n^q) \end{array} \right.$$

- un pigeonnier accueille au plus un pigeon (C2) :

$$q \text{ contraintes } (0, 1) \quad \left\{ \begin{array}{l} (0, 1, x_1^1, \dots, x_n^1) \\ \dots \\ (0, 1, x_1^q, \dots, x_n^q) \end{array} \right.$$

2 Travail demandé

- écrire un premier programme `solver.cpp` qui permet de résoudre le problème de manière non parallèle,

- écrire un second programme `psolver.cpp` qui permet de résoudre le problème de manière parallèle soit en utilisant MPI, soit en utilisant OpenMP,
- Il est de votre ressort de déterminer quelle(s) partie(s) vous désirez paralléliser. Les programmes devront prendre en paramètres le nombre de pigeons et le nombre de pigeonnières.

3 Organisation et remise du travail

- vous pouvez vous constituer par groupe de projet de 4 étudiants au maximum,
- vous enverrez votre travail sous forme d'une archive `.tgz` par email à mon adresse électronique,
- le titre de l'email sera **projet parallèle** et vous joindrez l'archive et indiquerez les noms des participants du groupe,
- la date de remise est le 24 décembre à minuit,
- vous créerez un makefile pour automatiser la compilation des sources
- vous créerez un shell `perf.sh` qui permet de lancer et mesurer le temps d'exécution d'une instance satisfiable puis d'une instance insatisfiable