Calcul du PageRank en Parallèle

Le but de cet exercice est d'implanter une version parallèle du fameux algorithme PageRank pour classer des pages web. L'algorithme s'exécute sur un graphe avec N nœuds dont chaque nœud correspond à une page web et chaque arête (i, j) represente un lien (hyperlink) entre deux pages web ayant les nœuds correspondants i et j. Ce graphe est representé par une matrice d'adjacence $\mathbf{A}$ tel que $\mathbf{A}_{ij} \neq 0$ si et seulement si l'arête (i, j) existe dans le graphe. Sur cette matrice, l'algorithme se déroule de manière iterative jusqu'à une convergence dans les valeurs de PageRank (stockées dans un vecteur $\mathbf{x}$) comme la suite:

Initialiser $\mathbf{x}_{old}$ tel que $\mathbf{x}_{old}(i) = 1/N$
for iter = 1 ... maxIters **do**
  for j = 1 ... N **do**
    $\mathbf{x}(j) = 0$
    for $\mathbf{A}_{ij} \neq 0$ **do**
      $\mathbf{x}(j) = \mathbf{x}(j) + \mathbf{x}_{old}(i) * (1/|\mathbf{A}_{i,:}|)$
  $\mathbf{x} = \mathbf{x}/||\mathbf{x}||$
  if $||\mathbf{x} - \mathbf{x}_{old}||/||\mathbf{x}|| \leq 1e-6$ **then**
    break;
  else
    $\mathbf{x}_{old} = \mathbf{x}$
return $\mathbf{x}$

Ici, $|\mathbf{A}_{i,:}|$ correspond au nombre d'eléments non-nul dans la ligne i de la matrice $\mathbf{A}$, $||\mathbf{x}|| = \sqrt{\sum \mathbf{x}_i^2}$ est la norme du vecteur $\mathbf{x}$ et **maxIters** est une constante pour borner le nombre d'itérations en cas d'une convergence lente.

On vous fournit un code squelette **pagerank.cpp** qui construit la matrice d'adjacance pour un graphe qui modélise des pages web. La matrice est construite dans un format particulier, appelé CSC (compressed sparse column), qui comprend pour chaque colonne j une liste d'indices des lignes appartenant à cette colonne j (c'est à dire, chaque elément i dans cette liste correspond à l'elément non-nul (i, j) dans la matrice d'adjacence et l'arête (i, j) dans le graphe correspondant). Tous les indices de ligne de toutes les colonnes sont stockés dans un tableau **rowIdx** de manière contigue, et les indices de ligne d'une colonne j se trouve dans l'intervalle [**colBegin[j]**, **colBegin[j + 1] - 1**] de ce tableau. Donc, la liste des lignes dans la colonne j est comme la suite:

rowIdx[colBegin[j]], rowIdx[colBegin[j] + 1], ..., rowIdx[colBegin[j + 1] - 1]

Vous pouvez également lire la fonction **printAdjMatrix()** afin de mieux comprendre le format.

On vous fournit la version séquentielle du calcul de PageRank avec cette structure de donnée dans la fonction **calculatePageRankSeq()**. La matrice d'adjacence et les vecteurs $\mathbf{x}$ et $\mathbf{x}_{old}$ seront stockés entièrement dans chaque processus; cependant, chaque processus se tachera de calculer N/P eléments consecutifs du vecteur $\mathbf{x}$ (c'est à dire, processus 0 s'occupera des premières N/P eléments, processus 1 calculera les N/P eléments qui suivent, etc.). Vous pouvez supposer que N sera toujours divisible par P, le nombre de processus. Vérifier bien votre code parallèle en affichant la norme du $\mathbf{x}$ à chaque itération et en la comparant avec la vérsion séquentielle.