

Développement d'une couche de cache de calculs pour la bibliothèque `igraph`

Rushed Kanawati
A³, LIPN
USPC/UP13 - CNRS UMR 7030
rushed.kanawati@lipn.univ-paris13.fr

Description du sujet

`igraph` est une bibliothèque *open source* de manipulation et d'analyse de graphes. Cette bibliothèque peut être utilisée dans des programmes écrits en Python ou en R. La bibliothèque est disponible sur : `igraph.sourceforge.net` Elle est très utilisée pour l'analyse de grands graphes et offre un nombre important de primitives d'analyse des réseaux complexes (ex. Calcul de différentes centralités, du coefficient de clustering, différentes implémentations des algorithmes reconnus pour la détection de communautés dans les réseaux complexes).

Or, la plupart des réseaux complexes ont des tailles très importants en nombre de nœuds. Pensez par exemple aux réseaux décrivant des interactions sur des sites de média sociaux (ex. Facebook) ou de microblogging (ex. twitter). Ceci rends beaucoup de calculs usuels sur ces graphes très couteaux en termes de temps de calculs. Une idée simple pour améliorer les performances est de fournir une *couche de cache de calculs*. **Cette couche doit permettre de sauvegarder les calculs déjà faits pour un graphe de sorte à ne pas refaire le même calcul plus d'une fois.**

Par exemple, si pour un graph G on a déjà calculé le diamètre ¹, l'appel pour calculer les longueurs de plus courts chemins dans le graph va retourner immédiatement les résultats puisque le calcul du diamètre s'appuie sur le calcul des plus courts chemins et que ces résultats seront sauvegardées dans

¹Le diamètre d'un graphe est la plus grande distance possible qui puisse exister entre deux de ses sommets.

la couche de cache. De même, si on calcul la centralité d'intermédiarité² pour un graph G , le calcul de plus courts chemins entre n'importe quel couple de nœuds sera immédiat puisque le résultat sera déjà dans le cache.

Outre l'accélération des calculs, la disponibilité d'une couche de cache permet de simplifier la programmation. Par exemple, elle va permettre de pouvoir désigner les mesures sur un graphe par les méthodes de calculs de ces mesures au lieu de sauvegarder les résultats dans des variables intermédiaires afin d'éviter de les recalculer comme c'est couramment fait aujourd'hui. La clarté et la lisibilité du code seront ainsi renforcées.

Travail demandé

- 1 Analyser les dépendance en terms de calculs entres les méthodes de bases offertes par les classes de représentation de graphs : les classes `Graph` et `GraphBase`.
- 2 Proposer un schéma efficace pour implementer une couche de cache de calculs.
- 3 Réaliser une étude d'évaluation des performances de la couche cache proposée.
- 4 Extension du principe de cache de calculs aux graphs dérivés d'un graphe (ex. sous-graphes, composantes connexes)
- 5 Fournir une couche de persistance de la couche de cache de calculs.

Intérêt du projet

- 1 Enrichir l'ensemble de solutions open source d'analyse de grands graphes.
- 2 Apprentissage/revision des algorithmes de base de manipulation et de mesures sur les graphes
- 3 Apprentissage de l'analyse des graphes complexes et des média sociaux.

²Le pourcentage de plus courts chemins dans le réseau qui passent par le nœud