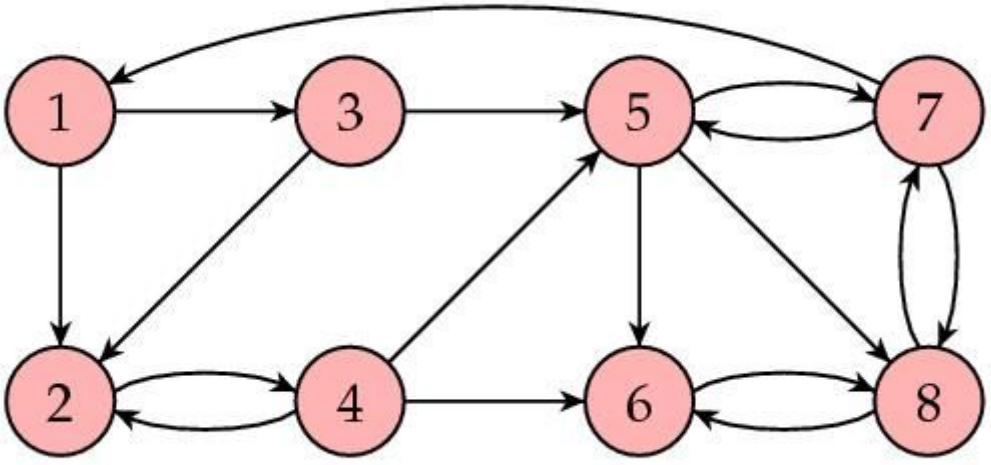
Philippe WU
Dino GURNARI

Projet
Calcul efficace du PageRank



Programmation Impérative
Projet réalisé en Ada

Philippe WU
Dino GURNARI

RÉSUMÉ

Ce rapport contient :

- Une introduction qui présente le sujet traité
- Le plan du rapport
- Une architecture des modules utilisés
- Une explication des principaux choix réalisés
- Une présentation des principaux algorithmes
- Un tableau des durées d'exécution moyenne des programmes
- Un bilan des difficultés rencontrées et les solutions adoptées
- Un bilan technique sur l'état d'avancement du projet et les perspectives d'amélioration
- Un bilan personnel

Philippe WU
Dino GURNARI

INTRODUCTION

En 1998, Brin & Page ont proposé de trier les pages Internet à partir de leur popularité. Leur tri, appelé Pagerank, repose sur deux critères : le référencement par les autres pages et la qualité de ces dernières. Ainsi, une page est d'autant plus populaire qu'elle est référencée par un grand nombre de pages. Pagerank prend également en compte la qualité des pages qui la référencent. Plus une page en référence d'autres, moins elle a de valeurs.

Le projet porte sur la réalisation de cet algorithme de PageRank.
L'objectif est de calculer tous les poids des pages web d'un réseau donné et de les trier par ordre décroissant.

PLAN DU RAPPORT

I. L'Architecture du projet

II. Les choix réalisés

III. Les principaux algorithmes

IV. La performance des programmes

V. Bilan technique

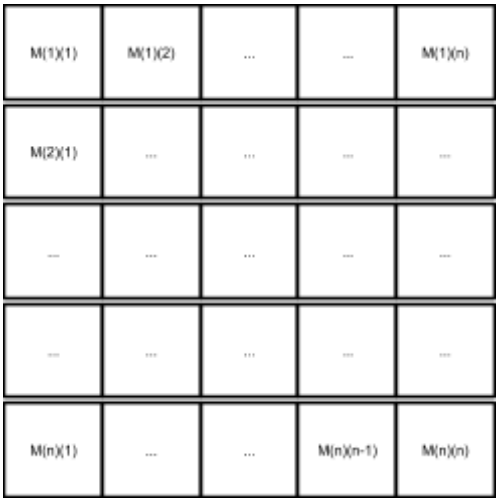
VI. Bilan personnel

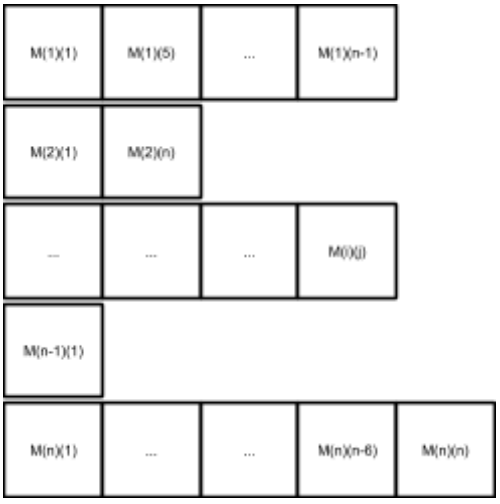
Philippe WU
Dino GURNARI

ARCHITECTURE DU PROJET

Choix de l'implémentation des matrices :

- Pour l'implémentation naïve (à gauche) nous avons choisi de réaliser des listes de listes telles qu'une matrice M soit en fait une liste de lignes. Ce choix a été effectué pour faciliter le calcul du produit vecteur matrice VM utile au calcul du pagerank.
- Pour l'implémentation creuse (à droite) nous avons choisi de réaliser une liste de LCA telle que la liste principale contient toutes les lignes sous forme de LCA et les LCA uniquement les valeurs non nulles ainsi que l'indice de la colonne. De plus si une ligne est totalement vide nous somme censé mettre 1/N à tous les coordonnées, pour un gain de mémoire nous ne mettons pas ces valeurs mais les prenons en compte lors du calcul du poids si une ligne est entièrement vide (c'est-à-dire M(i) = Null).



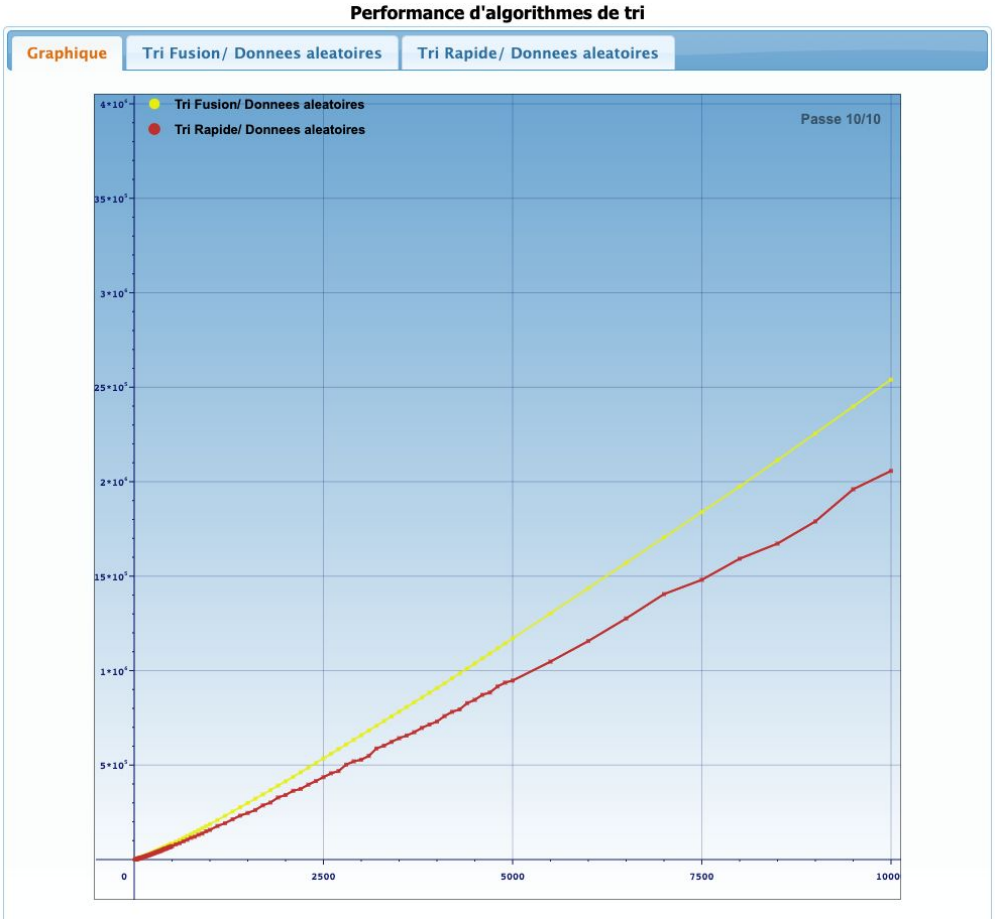


Philippe WU
Dino GURNARI

PRINCIPAUX CHOIX RÉALISÉS

Choix de l'algo de tri :

Pour la réalisation du tri des pages web, nous avons implémenté un algorithme de tri. Parmi les différents algorithmes dont nous disposons en notre connaissance à savoir : le tri par sélection, tri par insertion, tri fusion, tri rapide, nous avons décidé d'implémenter l'algorithme du tri rapide.
Pour des tableaux de grande taille, les algorithmes de tri rapide/fusion sont plus performants que les algorithmes de tri par sélection/insertion. En effet, la taille des réseaux peut atteindre des tailles de l'ordre de 10^5.
Nous avons donc cherché à comparer l'algorithme de tri rapide et tri fusion à travers une simulation pour déterminer notre choix final. (cf figure)



source : http://lwh.free.fr/pages/algo/tri/comparaison_tri.html

Philippe WU
Dino GURNARI

PRÉSENTATION DES PRINCIPAUX ALGORITHMES

Calcul de G : Pour former G nous devons tout d'abord lire le fichier Réseau et ligne par ligne ajouter 1 dans une matrice vide lorsqu'un lien entre deux pages existe. Une fois cela fait nous normalisons les lignes et dans le cas des matrices naïves nous remplaçons les lignes vides par des lignes de 1/N (N étant le nombre total de pages). Enfin nous appliquons la formule suivante pour obtenir G :

$$G = \alpha \cdot S + \frac{(1-\alpha)}{N}\mathbf{e}\mathbf{e}^T \quad \alpha \in [0,1]$$

La distinction est faite entre matrices creuses et naïves de la même manière que précédemment avec les lignes vides.

Calcul des Poids : Le calcul des poids est plutôt simple, en effet c'est simplement un calcul matriciel itéré un certain nombre de fois. Dans le cas des matrices naïves il suffit d'utiliser la formule d'un produit de matrices et pour les matrices creuses il faut vérifier pour chaque ligne si : elle est vide auquel cas nous utilisons 1/N ou si elle ne l'est pas ne compter que les coefficient non-nuls de la matrice creuse.

Spécificité calcul en Matrice creuse : Pour limiter la taille des matrices creuses nous avons simplifié la création de G et le calcul du poids pour mélanger les deux étapes, en effet si une ligne de G est vide c'est comme si elle contenait des 1/N et pour une ligne non vide de S tous les éléments nuls de cette ligne correspondront à (1-α)/N dans G donc nous les omettons aussi mais lors du calcul du Poids nous les prenons en compte.

Philippe WU
Dino GURNARI

PERFORMANCE DES PROGRAMMES

Temps d'exécution sous windows à l'aide de la commande PowerShell
Measure-Command{}

Fichiers .net	Matrices Naïves	Matrices Creuses
exemple_sujet	33 ms	31 ms
Worm	85 ms	113 ms
brainlinks	STACK_OVERFLOW	STACK_OVERFLOW
Linux26	STACK_OVERFLOW	STACK_OVERFLOW

Conclusion : Pour les matrices naïves le fait d'obtenir un erreur STACK_OVERFLOW est logique lorsque l'on parle de matrice de taille 10000*10000. Cependant nous avons obtenus la même erreur pour les matrices creuses ce qui implique que notre implémentation creuse n'est pas optimale. Nous essayerons d'améliorer cela d'ici le rendu du code source et la présentation orale.

Philippe WU
Dino GURNARI

BILAN TECHNIQUE

- Difficultés rencontrées et les solutions adoptées :

  Nous avons eu des difficultés notamment sur la mise en place du tri et des matrices creuses. Pour les matrices creuses nous avons eu du mal à déterminer comment les implémenter assez facilement pour pouvoir effectuer les diverses opérations nécessaires au calcul des poids mais aussi assez “creuses” pour ne pas prendre trop d’espace mémoire, c’est pour cela que nous avons décidé de partir sur des matrices composées de listes de LCA, pour avoir une base fixe ainsi que des sous-parties creuses.

  En ce qui concerne la mise en place du tri, nous avons été confrontés à plusieurs difficultés. Nous avons eu deux types d'erreurs lors de la compilation. La première concerne une mauvaise conception de la procédure “Echanger” qui prenait en argument des paramètres qui n’étaient en fait pas des variables. Ce problème a été résolu en remplaçant dans les paramètres les valeurs à échanger par leur indice.

  Ensuite la deuxième erreur concerne une erreur de type “constraint error”. Nous avons fait l’erreur d’implanter notre programme de tri dans le programme principal avant même de le tester. Du coup le programme principal se lançait sans faire le tri.
  Après avoir mis en place un programme de test et vérifié à la main l’algorithme de tri, nous avons réussi à trouver d’où venait le problème. Le programme était correct. Le problème venait des paramètres lors de l’appel du programme de tri.

- L’état d’avancement du projet et les perspectives d’amélioration :

  In fine, le projet fonctionne, il compile et s'exécute, il est possible de modifier le nombre d’itération et la valeur alpha grâce en appelant le programme dans la ligne de commande, ainsi que choisir quel réseau étudier. De plus, la lecture et l'écriture des fichiers fonctionnent parfaitement. L’implémentation Naïve fonctionne mais l’implémentation Creuses mériterait elle d’être peaufinée car bien qu'elle s’exécute, les résultats ne sont pas les bons dû surement à un problème dans le calcul des poids.

Philippe WU
Dino GURNARI

BILAN PERSONNEL

Les apports personnels tirés du projet :

Dino Gurnari :
Ce projet, au-delà de l'aspect presque "historique" qu'il représente avec l'origine des moteurs de recherches modernes, m'a permis d'apprendre à gérer un travail de groupe et les difficultés que cela représente. La répartition du travail et la réflexion commune nécessaire nous à permis de rendre un travail dont nous sommes fiers malgré le fait de n'avoir pas pu atteindre notre objectif final de tout faire fonctionner parfaitement. En effet il reste du travail d'optimisation à faire pour obtenir une meilleure implémentation mais je pense que le principal intérêt de ce genre de projet et de nous faire gagner en autonomie et en expérience pour à l'avenir pour entreprendre des projets bien plus important, en conséquent je ne retire que du positif de ce projet et j'ai hâte d'en commencer un nouveau pour mettre en application ce que j'ai appris ici.

Philippe Wu :
Malgré le peu de temps que j'ai pu consacrer au projet, je l'ai trouvé très enrichissant. Le sujet est intéressant. Je trouve que travailler en binôme sur un projet est la partie la plus enrichissante d'un point de vue pédagogique car cette situation se rapproche le plus de ce qu'on sera amené à faire dans le monde professionnel. S'investir sur quelque chose de concret est beaucoup plus intéressant que l'enseignement scolaire classique. On est amené à échanger nos réflexions afin d'aboutir à des solutions. Lorsqu'on est bloqué, on peut discuter avec notre binôme pour se débloquer. C'est aussi très satisfaisant personnellement de réussir à surmonter les obstacles auxquels on a été confrontés. Je regrette seulement de ne pas avoir pu m'investir pleinement du fait des autres impératifs scolaires.