

Implémentation du PageRank avec MPI

Ayaz BADOURALY

École Centrale Paris

13 avril 2016

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

Cadre historique : 1998, invention du *pagerank*



Sergey BRIN



Lawrence PAGE

Problématique

- En pratique : jeux de données énormes
- Solution : librairie MPI

Problématique

- En pratique : jeux de données énormes
- Solution : librairie MPI

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

Modélisation de la navigation du surfeur

- A : le surfeur clique sur un lien externe de sa page courante
- D : le surfeur arrive sur une page sans lien externe
- E : le surfeur change de page au hasard

Modélisation de la navigation du surfeur

- A : le surfeur clique sur un lien externe de sa page courante
- D : le surfeur arrive sur une page sans lien externe
- E : le surfeur change de page au hasard

Modélisation de la navigation du surfeur

- A : le surfeur clique sur un lien externe de sa page courante
- D : le surfeur arrive sur une page sans lien externe
- E : le surfeur change de page au hasard

Modélisation du web

$$M = dA + dD + (1 - d)E$$

où d est le *damping factor*

Définition du pagerank

$$Mp = p$$

Algorithme du pagerank

$$\left\{ \begin{array}{lcl} p^0 & = & \begin{pmatrix} \frac{1}{\dim(M)} \\ \vdots \\ \frac{1}{\dim(M)} \end{pmatrix} \\ \forall n \in \mathbb{N} : p^{n+1} & = & Mp^n \end{array} \right.$$

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation**
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation**
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

Les vecteurs

Vecteur mathématique

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

Vecteur informatique

```
{ dim      = 6  
  vector   = [1; 2; 3; 4; 5; 6]
```

Les matrices COO

Matrice mathématique

$$\begin{pmatrix} 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 6 & 7 \\ 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 9 & 0 \end{pmatrix}$$

Matrice informatique

$$\begin{cases} \text{nnz} &= 9 \\ \text{rows} &= [0; 0; 0; 1; 1; 2; 2; 3; 4] \\ \text{cols} &= [0; 1; 2; 3; 4; 3; 4; 4; 3] \\ \text{values} &= [1; 2; 3; 4; 5; 6; 7; 8; 9] \end{cases}$$

Exemples de matrices creuses

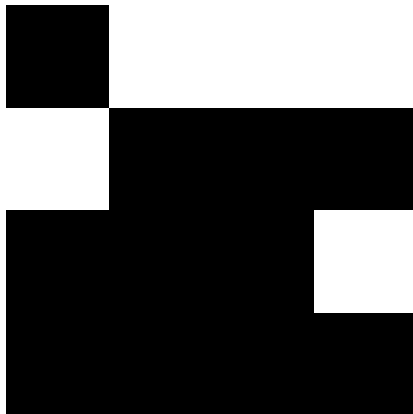


Figure: Matrice d'adjacence du graphe interstices_1

Exemples de matrices creuses

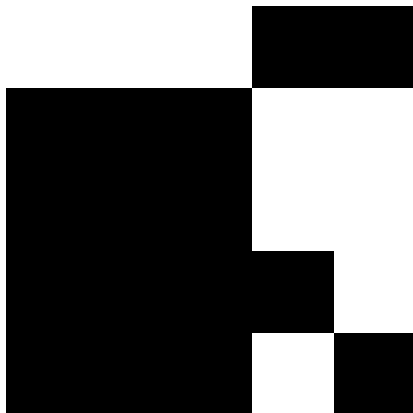


Figure: Matrice d'adjacence du graphe interstices_2

Exemples de matrices creuses

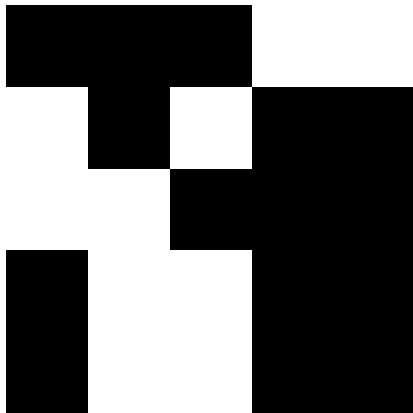


Figure: Matrice d'adjacence du graphe `interstices_3`

Exemples de matrices creuses

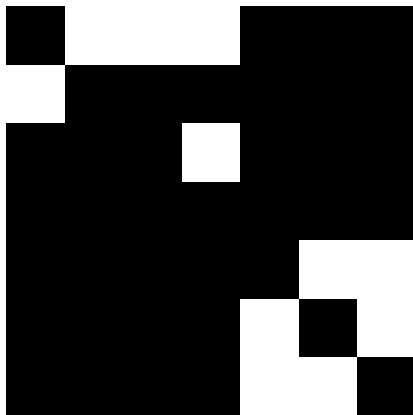


Figure: Matrice d'adjacence du graphe interstices_4

Exemples de matrices creuses

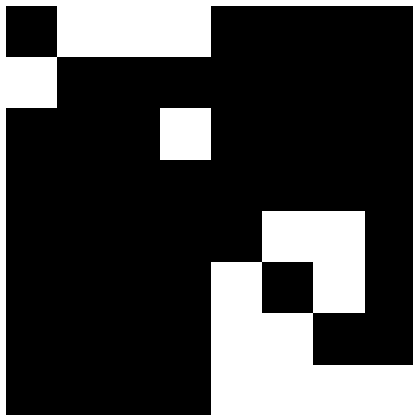


Figure: Matrice d'adjacence du graphe interstices_5

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation**
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

Chargement de la matrice en mémoire

Processus maître

- lecture du graphe écrit dans un fichier
- création de la matrice transposée d'adjacence
- matrice rendue stochastique en colonnes
- calcul d'une répartition équitable entre les processus esclaves
- MPI_Send

Processus esclaves

- MPI_Recv
- multiplication par le *damping factor*

Chargement de la matrice en mémoire

Processus maître

- lecture du graphe écrit dans un fichier
- création de la matrice transposée d'adjacence
- matrice rendue stochastique en colonnes
- calcul d'une répartition équitable entre les processus esclaves
- MPI_Send

Processus esclaves

- MPI_Recv
- multiplication par le *damping factor*

Chargement de la matrice en mémoire

Processus maître

- lecture du graphe écrit dans un fichier
- création de la matrice transposée d'adjacence
- matrice rendue stochastique en colonnes
- calcul d'une répartition équitable entre les processus esclaves
- MPI_Send

Processus esclaves

- MPI_Recv
- multiplication par le *damping factor*

Chargement de la matrice en mémoire

Processus maître

- lecture du graphe écrit dans un fichier
- création de la matrice transposée d'adjacence
- matrice rendue stochastique en colonnes
- calcul d'une répartition équitable entre les processus esclaves
- MPI_Send

Processus esclaves

- MPI_Recv
- multiplication par le *damping factor*

Chargement de la matrice en mémoire

Processus maître

- lecture du graphe écrit dans un fichier
- création de la matrice transposée d'adjacence
- matrice rendue stochastique en colonnes
- calcul d'une répartition équitable entre les processus esclaves
- MPI_Send

Processus esclaves

- MPI_Recv
- multiplication par le *damping factor*

Chargement de la matrice en mémoire

Processus maître

- lecture du graphe écrit dans un fichier
- création de la matrice transposée d'adjacence
- matrice rendue stochastique en colonnes
- calcul d'une répartition équitable entre les processus esclaves
- MPI_Send

Processus esclaves

- MPI_Recv
- multiplication par le *damping factor*

Calcul du pagerank

Processus maître

- MPI_Allreduce
- ajout du terme complémentaire du *damping factor* à next_pagerank
- calcul de la distance entre pagerank et next_pagerank
- MPI_Bcast
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée

Processus esclaves

- multiplication de pagerank par la sous-matrice
- MPI_Allreduce
- ajout du terme complémentaire du *damping factor* à next_pagerank
- MPI_Bcast
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée
- copie de next_pagerank dans pagerank

Calcul du pagerank

Processus maître

- **MPI_Allreduce**
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- calcul de la distance entre `pagerank` et `next_pagerank`
- **MPI_Bcast**
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée

Processus esclaves

- multiplication de `pagerank` par la sous-matrice
- **MPI_Allreduce**
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- **MPI_Bcast**
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée
- copie de `next_pagerank` dans `pagerank`

Calcul du pagerank

Processus maître

- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- calcul de la distance entre `pagerank` et `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée

Processus esclaves

- multiplication de `pagerank` par la sous-matrice
- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée
- copie de `next_pagerank` dans `pagerank`

Calcul du pagerank

Processus maître

- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- calcul de la distance entre `pagerank` et `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée

Processus esclaves

- multiplication de `pagerank` par la sous-matrice
- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée
- copie de `next_pagerank` dans `pagerank`

Calcul du pagerank

Processus maître

- MPI_Allreduce
- ajout du terme complémentaire du *damping factor* à next_pagerank
- calcul de la distance entre pagerank et next_pagerank
- MPI_Bcast
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée

Processus esclaves

- multiplication de pagerank par la sous-matrice
- MPI_Allreduce
- ajout du terme complémentaire du *damping factor* à next_pagerank
- MPI_Bcast
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée
- copie de next_pagerank dans pagerank

Calcul du pagerank

Processus maître

- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- calcul de la distance entre `pagerank` et `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée

Processus esclaves

- multiplication de `pagerank` par la sous-matrice
- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée
- copie de `next_pagerank` dans `pagerank`

Calcul du pagerank

Processus maître

- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- calcul de la distance entre `pagerank` et `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée

Processus esclaves

- multiplication de `pagerank` par la sous-matrice
- `MPI_Allreduce`
- ajout du terme complémentaire du *damping factor* à `next_pagerank`
- `MPI_Bcast`
- prise de décision d'itérer une nouvelle fois le calcul ou non en fonction de la précision souhaitée
- copie de `next_pagerank` dans `pagerank`

Finalisation

Processus maître

- écriture des résultats dans un fichier

Processus esclaves

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

PageRank

- [1] *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Sergey Brin and Lawrence Page, 1998,
infolab.stanford.edu/~backrub/google.html
- [2] *Using your laptop to compute PageRank for millions of webpages*, Michael Nielsen, 2008, michaelnielsen.org/blog/using-your-laptop-to-compute-pagerank-for-millions-of-webpages/
- [3] *Le théorème de Perron-Frobenius, les chaînes de Markov et un célèbre moteur de recherche*, Bachir Bekka, 2007,
agreg-maths.univ-rennes1.fr/documentation/docs/Perron-Frobenius.pdf
- [4] *La formule de PageRank sur des exemples*,
interstices.info/encart.jsp?id=c_21839&encart=0&size=780,580

Divers

- [5] *A description on how to use and modify libpng*, Glenn Randers-Pehrson, 2016,
www.libpng.org/pub/png/libpng-manual.txt
- [6] *CUDA Toolkit Documentation*,
docs.nvidia.com/cuda/cusparse/

Plan

- 1 Introduction
- 2 Algorithme du pagerank
- 3 Implémentation
 - Les structures de données
 - Le programme
- 4 Bibliographie
- 5 Conclusion

Travail réalisé

- Cahier des charges rempli
- Code segmenté en librairies réutilisables

Points d'amélioration

- Utilisation des matrices au format CSR
- Meilleure répartition des données entre les processus
- Analyse des temps d'exécution

Remerciements

- À Frédéric MAGOULES pour ses cours et ses conseils ;
- À VIA Centrale Réseaux pour m'avoir fourni un cluster de test