2.7 Higher-Dimensional Meshes of Trees

Παράλληλοι Αλγόριθμοι

Θανάσης Κουτσώνας

μΠλ∀

25 Μαΐου 2006

Περίγραμμα

Meshes-of-Trees r-διαστάσεων.

- *Ορισμοί*
- *Ιδιότητες*

Ο γράφος Shuffle-Tree.

- *Κατασκευή του γράφου*
- *Ένα παράδειγμα: Πολλαπλασιασμός πινάκων*

Meshes-of-Trees r-διαστάσεων

Το $N \times N \times \cdots \times N$ *mesh of trees* r-διαστάσεων δημιουργείται προσθέτοντας δέντρα πάνω από τους κόμβους ενός r-διάστατου πλέγματος πλευράς N.

Σε κάθε κόμβο-φύλλο αντιστοιχεί ένα r-διάνυσμα $(i_1, i_2, \ldots, i_r)$, όπου $0 \leq i_1, \ldots, i_r \leq N - 1$.

Δύο κόμβοι-φύλλα συμμετέχουν στο ίδιο δέντρο αν τα διανύσματά τους διαφέρουν ακριβώς σε ένα στοιχείο. Η θέση του στοιχείου υποδηλώνει την διάσταση του κοινού τους δέντρου.

Ιδιότητες

Ένα mesh of trees r-διαστάσεων έχει:

- $|V| = (r+1)N^r - rN^{r-1}$ κόμβους
- $|E| = 2rN^r - 2rN^{r-1}$ ακμές
- rN^{r-1} πλήρη δέντρα με N φύλλα
- διάμετρος: $2r \log N = \Theta(\log |V|)$
- bisection width: $N^{r-1} = \Theta(|V|/rN)$

Μια πιο έξυπνη λύση

Το mesh of trees r-διαστάσεων έχει πολλούς επιπρόσθετους κόμβους, ενώ επιπλέον κάθε φύλλο έχει βαθμό r.

Όμως στον γράφο Shuffle-Tree, αφαιρώντας όλα τα δέντρα παρά μόνο μίας διάστασης και προσθέτοντας ακμές στα φύλλα ανάλογες του πλήθους τους, έχουμε μια δομή ίδιας υπολογιστικής δύναμης, με μόνο N^{r-1} δέντρα και βαθμό φύλλων το πολύ 3.

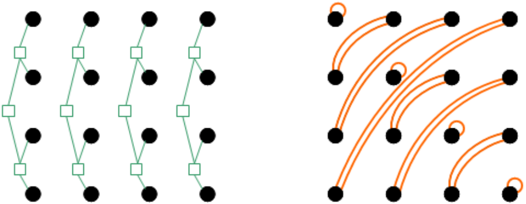
Κατασκευή του γράφου Shuffle-Tree

Ξεκινάμε πάλι από το r-διάστατο πλέγμα $N \times N \times \cdots \times N$:

$$\{(i_1, i_2, \ldots, i_r) \mid 0 \leq i_1, \ldots, i_r \leq N-1\}$$

- Προσθέτουμε ένα πλήρες δέντρο πάνω από τα φύλλα: $\{(i_1, \ldots, i_r) \mid 0 \leq i_r \leq N-1\}$ για $0 \leq i_1, \ldots, i_{r-1} \leq N-1$.
- Προσθέτουμε ακμές μεταξύ δύο φύλλων, όταν το διανύσμα θέσης το ενός είναι κυκλική μετατόπιση του άλλου. π.χ. το φύλλο $(i_1, i_2, \ldots, i_{r-1}, i_r)$ ενώνεται με τα $(i_r, i_1, \ldots, i_{r-1})$ και $(i_2, \ldots, i_r, i_1)$

Shuffle-Tree για $r = 2$ και $N = 4$



Ιδιότητες

Ο γράφος Shuffle-Tree (r, N) έχει:

- $|V| = 2N^r - N^{r-1}$ κόμβους
- $|E| = \Theta(N^r)$ ακμές
- N^{r-1} πλήρη δέντρα, δηλαδή το $1/r$ από ότι πριν.

Επιπλέον, τώρα κάθε κόμβος έχει βαθμό 2 ή 3.

Ένα απλό παράδειγμα

Υπολόγισε το $s_i = \sum_{j=0}^{N-1} a_{ij}$, για $0 \leq i < N$.

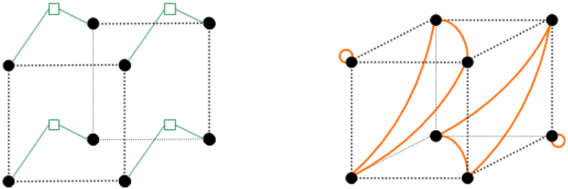
Σε πλήρες mesh of trees r-διαστάσεων θα χρειαζόταν $\log N$ βήματα.

Στο shuffle-tree χρειάζεται $\log N + 1$ βήματα:

- Σε ένα βήμα ανταλλάσω τα a_{ij} με τα a_{ji},
- Σε $\log N$ βήματα το άθροισμα s_i βρίσκεται στην ρίζα του δέντρου της i-οστής στήλης.

Πολλαπλασιασμός πινάκων

Για τον πολλαπλασιασμό των $N \times N$ πινάκων $A = (a_{ij})$ και $B = (b_{ij})$, είδαμε ότι με mesh of trees $N \times N \times N$ χρειάζονται $2 \log N + 1$ βήματα. Με το Shuffle-Tree $(3, N)$;



Shuffle-Tree για $r = 3$ και $N = 2$

Πολλαπλασιασμός πινάκων

Με το Shuffle-Tree $(3, N)$ χρειάζονται μόνο $2\log N + 3$:

- Στο 1ο βήμα εισάγουμε το $B = (b_{ij})$ και στο 2ο βήμα το $A = (a_{ij})$ στη ρίζα του i, j δέντρου για $i, j \in [0, N-1]$.
- Στο βήμα $\log N + 2$ όλα τα φύλλα στέλνουν το περιεχόμενό τους 'προς τα δεξιά'.
- Εκτελούνται οι πράξεις και τα γινόμενα περνούν δεξιά.
- Μετά από $\log N$ βήματα τα αποτελέσματα c_{ik} βρίσκονται στις ρίζες των k, i δέντρων.

Είναι όμως πράγματι ίδιας υπολογιστικής ικανότητας;

Είδαμε πως αν και η δομή αυτή έχει $1/r$ δέντρα σε σχέση με πριν, υπολογίζει τα περισσότερα προβλήματα σε μόνο κάποιο μικρό σταθερό αριθμό βημάτων παραπάνω.

Π.χ. για τον πολλαπλασιασμό πινάκων, με το ένα τρίτο των δέντρων χρησιμοποίησε μόνο δύο παραπάνω βήματα!

Αν όμως θέλουμε να χρησιμοποιήσουμε την δομή για υπολογισμούς με *pipelining*, θα πρέπει να επιβραδύνουμε την είσοδο των δεδομένων κατά ένα παράγοντα r.