GPUによる汎用計算を目指して
～木関数の簡単な並列アルゴリズム～

Alexander Bowe　小池 敦　定兼 邦彦

問題

・根付き木に定義される関数 f を計算する

・各節点 v は重み $w(v)$ を持つ

・v での関数の値は再帰的に定義される

$$f(v;\otimes,\oplus)=\begin{cases} w(v) & v\text{は葉} \\ w(v)\otimes\left(\bigoplus_{u\in S(v)} f(u;\otimes,\oplus)\right) & \text{それ以外}\end{cases}$$

$S(v)$ は v の子節点の集合

$\otimes,\oplus$は2項演算子

・根 r での値 $f(r;\otimes,\oplus)$ を求める

応用

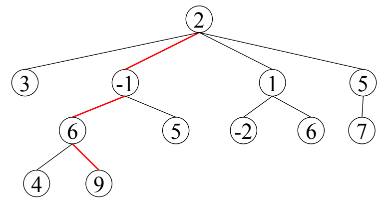
・$f(r;+,\max)$ は最大パス問題の解を与える

$$f(v)=w(v)+\max_{u\in S(v)} f(u)$$

・根から葉までのパスの重み

＝そのパス上の節点の重みの和

最大パス問題＝最大重みのパスを求める



結果

・n: 節点, p プロセッサ EREW-PRAM

・$O(n/p+\log^2 p)$ 時間

$\otimes$ は結合的かつ $\oplus$ に対して左分配的

$a\otimes(b\oplus c)=(a\otimes b)\oplus(a\otimes c)$

$\oplus$ は結合的

・$O(n/p+\log p)$ 時間

上の条件に加えて

$\otimes$ は逆元を持つ，つまり $\otimes$ は群

・節点の重みが k ビット，f の値が d ビットのとき

木は $2n(k+1)$ ビット

作業領域は $O(p(d+\log n))$ ビット

データ構造

2つの配列で表現

$P[1..2n]$: 木構造を表現する括弧

$W[1..2n]$: 節点の重みを表現する配列

節点 v に対応する括弧の位置を s,t とすると

$W[s]=w(v),\ W[t]=(w(v))^{-1}$

P (( ) ((( ) ( )) ( )) (( ) ( )) (( )))

W 2 3 -3 -1 6 4 -4 9 -9 -6 5 -5 1 1 -2 2 6 -6 -1 5 7 -7 -5 -2

木関数の効率的並列計算アルゴリズム

P, Wの部分配列に対し，7つ組みを定義する

(L, M, B, E, R, A, F)

$D[i]=\sum_{j=1}^{i} g(P[j])$　各節点の深さ

$M=\min_{0\le i\le m} D[i]$　根の深さ

$L=D[m]$　最後の節点の深さ

$B=g(P[1])$　最初の括弧

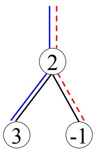
$E=g(P[m])$　最後の括弧

$R=\bigotimes_{i=1}^{m} W[i]$　オイラーパスの重み

$A=\bigotimes_{i=h+1}^{m} W[i]$　根から最右葉のパスの重み

(h は $D[i]$ の最小値の添字)

P		(	(	)	(	
D	0	1	2	1	2	
W		2	3	-3	-1	



(2,0,1,1,1,1,5)

2+3

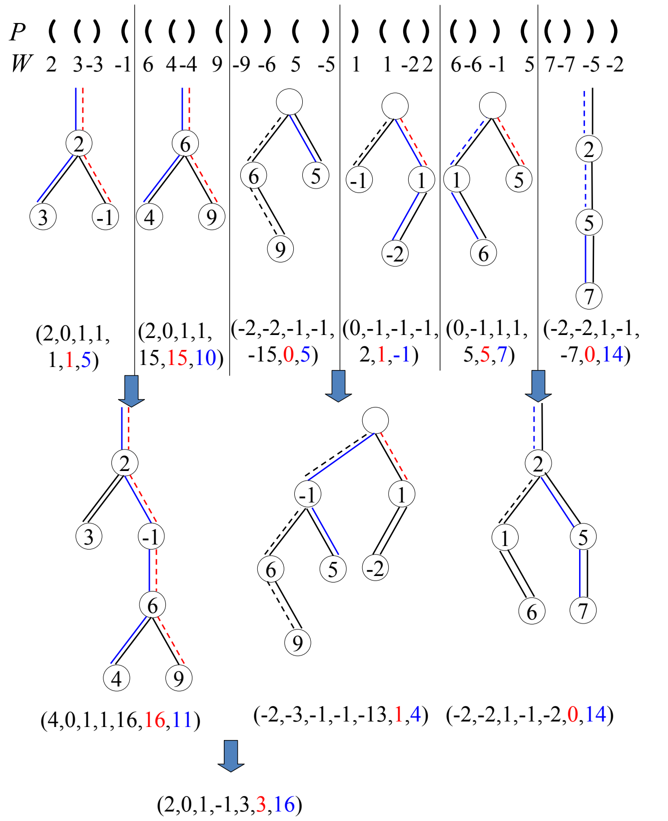
2+(-1)

2+3+(-3)+(-1)

隣り合う2つの7つ組みは次のように併合できる

$$L=L_1+L_2$$
$$M=\min\{M_1, L_1+M_2\}$$
$$B=\begin{cases}B_2 & \text{if } B_1=0\\ B_1 & \text{otherwise}\end{cases}$$
$$E=\begin{cases}E_1 & \text{if } E_2=0\\ E_2 & \text{otherwise}\end{cases}$$
$$R=R_1\otimes R_2$$
$$A=\begin{cases}A_1\otimes R_2 & \text{if } M=M_1\\ A_2 & \text{otherwise}\end{cases}$$
$$F_1'=\begin{cases}F_1\oplus A_1 & \text{if } E_1=1 \text{ and } B_2=-1\\ F_1 & \text{otherwise}\end{cases}$$
$$H=A_1\otimes R_2\otimes(A_2)^{-1}$$
$$F=\begin{cases}F_1'\oplus(H\otimes F_2) & \text{if } M=M_1\\ \left((H)^{-1}\otimes F_1'\right)\oplus F_2 & \text{otherwise}\end{cases}$$

$$\begin{array}{lll} (& \to & (1,\ 0,\ 1,\ 1,\ w(v),\ w(v),\ I_\oplus)\\ ) & \to & (-1,\ -1,\ -1,\ -1,\ (w(v))^{-1},\ I_\otimes,\ I_\oplus)\\ \varepsilon & \to & (0,\ 0,\ 0,\ 0,\ I_\otimes,\ I_\otimes,\ I_\oplus)\end{array}$$



NII　連絡先：定兼 邦彦（Kunihiko Sadakane）／ 国立情報学研究所 情報学プリンシプル研究系 准教授