Paralelni algoritam za rješavanje trodijagonalnog linearnog sustava LU faktorizacijom

Filip Nikšić
fniksic@gmail.com

PMF—Matematički odjel

Paralelni algoritmi 1
30. svibnja 2008.

Pregled

Problem

Sučeni smo sa sustavom $Tx = y$:

$$\begin{bmatrix} a_1 & b_2 & & & & \\ c_2 & a_2 & b_3 & & & \\ & c_3 & a_3 & b_4 & & \\ & & c_4 & a_4 & \ddots & \\ & & & \ddots & \ddots & b_n \\ & & & & c_n & a_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

Rješenje

Rješavanju pristupamo na klasičan način i rješavamo ga u tri faze:

1. Faktoriziramo matricu sustava $T = LU$; $Tx = y$ postaje ekvivalentno s $L(Ux) = y$
2. Riješimo jednostavan sustav $Lz = y$
3. Riješimo jednostavan sustav $Ux = z$

LU faktorizacija (1)

$$\begin{bmatrix} 1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ & & l_n & 1 \end{bmatrix} \begin{bmatrix} d_1 & e_2 & & \\ & d_2 & \ddots & \\ & & \ddots & e_n \\ & & & d_n \end{bmatrix} = \begin{bmatrix} a_1 & b_2 & & \\ c_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_n \\ & & c_n & a_n \end{bmatrix}$$

Pažljivim raspisivanjem dobivamo:

- $e_i = b_i, \;\; i \geqslant 2$
- $d_1 = a_1$
- $d_i = a_i - c_i b_i / d_{i-1}, \;\; i \geqslant 2$
- $l_i = c_i / d_{i-1}, \;\; i \geqslant 2$

LU faktorizacija (2)

- Zapišimo $d_i = p_i / q_i$. Tad imamo:

$$\frac{p_i}{q_i} = \frac{a_i p_{i-1} - c_i b_i q_{i-1}}{p_{i-1}}$$

- Uz dodatnu definiciju $b_1 = 0$, $c_1 = 0$ imamo:

$$\begin{aligned} \begin{bmatrix} p_i \\ q_i \end{bmatrix} &= \begin{bmatrix} a_i & -c_i b_i \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ q_{i-1} \end{bmatrix} \\ &= M_i \begin{bmatrix} p_{i-1} \\ q_{i-1} \end{bmatrix} = M_i \ldots M_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = N_i \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{aligned}$$

LU faktorizacija (3)

- Razdijelimo ulazne podatke na p procesora. Ako je $n = qp + r$, prvih r procesora dobije $q + 1$ komponenti vektora a, b i c (i y), ostali q komponenti.
- Svaki procesor lokalno računa nepotpune parcijalne produkte N_i' od podataka koje ima na raspolaganju.
- Paralelnim prefiksom procesori upotpune svoje parcijalne produkte N_i.
- Svaki procesor lokalno računa svojih $q + 1$, odnosno q komponenti d_i i l_i.

Rješavanje $Lz = y$ (1)

$$\begin{bmatrix} 1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ & & l_n & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

Kao i prije, pažljivim raspisivanjem dobijemo:

- $z_0 = y_0$
- $z_i = y_i - l_{i+1} z_{i-1}, \;\; i \geqslant 1$

Rješavanje $Lz = y$ (2)

- Zapisujemo $z_i = z_i/1$ i imamo:

$$\frac{z_i}{1} = \frac{y_i - l_{i+1} z_{i-1}}{1}$$

- Pretvaramo u matrično množenje, uz definiciju $l_1 = 0$:

$$\begin{aligned} \begin{bmatrix} z_i \\ 1 \end{bmatrix} &= \begin{bmatrix} -l_{i+1} & y_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_{i-1} \\ 1 \end{bmatrix} \\ &= P_i \begin{bmatrix} z_{i-1} \\ 1 \end{bmatrix} = P_i \dots P_0 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = Q_i \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned}$$

Rješavanje $Lz = y$ (3)

- Svaki procesor iz prethodne faze ima svojih $q + 1$, odnosno q komponenti l_i i y_i. Lokalno računa nepotpune parcijalne produkte Q_i'.
- Paralelnim prefiksom upotpune se parcijalni produkti Q_i.
- Svaki procesor lokalno računa (pročita iz matrica) svoje komponente z_i.

Rješavanje $Ux = z$ (1)

Slično kao i dosad, prvo pažljivo raspišemo sustav i dobijemo:

- $x_{n-1} = z_{n-1}/d_n$
- $x_i = z_i/d_{i+1} - e_{i+2}x_{i+1}/d_{i+1}, \;\; i < n-1$

Pretvaramo u matrično množenje, uz definiciju $e_{n+1} = 0$:

$$\begin{aligned}
\begin{bmatrix} x_i \\ 1 \end{bmatrix} &= \begin{bmatrix} -e_{i+2}/d_{i+1} & z_i/d_{i+1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i+1} \\ 1 \end{bmatrix} \\
&= R_i \begin{bmatrix} x_{i+1} \\ 1 \end{bmatrix} = R_i \ldots R_{n-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = S_i \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\end{aligned}$$

Rješavanje $Ux = z$ (2)

- Primijetimo da sad parcijalne produkte treba izračunati "unatrag".
- To postižemo tako da prenumeriramo procesore $j \mapsto p - j - 1$ i lokalno podatke $i \mapsto q - i$, odnosno $i \mapsto q - i - 1$.
- Zatim postupimo kao i u prethodnoj fazi. Izračunamo parcijalne produkte, lokalno svaki procesor izračuna svoje komponente rješenja x_i koje konačno skupimo u cjelovito rješenje.

Implementacija

- Program implementiramo u okviru *MPIApp* aplikacije kao modul *triLU*.
- Za međuprocesnu komunikaciju koristimo *MPI*.
- Za baratanje ulazom i izlazom koristimo se *HDF5* bibliotekom.
- Nulti (root) proces zadužen je za distribuciju podataka ostalim procesima. On svakom procesu šalje njegov komad vektora a, b, c i y.
- Kad računanje završi, root prima komade rješenja x, ali i vektora l i d od ostalih procesa.

Implementacija

- Program implementiramo u okviru *MPIApp* aplikacije kao modul *triLU*.
- Za međuprocesnu komunikaciju koristimo *MPI*.
- Za baratanje ulazom i izlazom koristimo se *HDF5* bibliotekom.
- Nulti (root) proces zadužen je za distribuciju podataka ostalim procesima. On svakom procesu šalje njegov komad vektora a, b, c i y.
- Kad računanje završi, root prima komade rješenja x, ali i vektora l i d od ostalih procesa.

Implementacijske poteškoće

1. Svakom procesu je za $Ux = z$ potrebna dodatna komponenta vektora b (prva komponenta koju ima proces sa za 1 većim indeksom).
2. Svakom procesu je za LU potrebna dodatna komponenta vektora d (zadnja komponenta koju ima proces sa za 1 manjim indeksom).
3. Već za relativno male sustave ($n > 1000$) množenjem 2×2 matrica dobijemo ili overflow ili underflow.

Vremenska složenost

- Složenost samog računskog dijela je $\mathcal{O}(n/p + \log p)$.
- Distribucija podataka na početku traje $(p-1)(\alpha + \beta \cdot 4n/p)$.
- Paralelni prefiksi ukupno traju $\mathcal{O}(\log p)$.
- Skupljanje podataka na kraju traje $(p-1)(\alpha + \beta \cdot 3n/p)$.
- Sve skupa:

$$T(p, n) = \mathcal{O}(n/p + \log p) + 2(p-1)\alpha + \beta \cdot 7n(p-1)/p$$

Zaključak

- Komunikacija je ozbiljno usko grlo u ovom rješenju.
- Valjalo bi razmotriti načine da se i sama distribucija podataka paralelizira.