LMD mokslo darbai, 384–389
1998, Vilnius

LU faktorizacijos lygiagretusis algoritmas

R. Čiegis (MII, VU), V. Starikovičius (VU)

Nagrinėsime LU faktorizacijos lygiagrečiuosius algoritmus, skirtus lygiagretiesiems kompiuteriams su paskirstyta atmintimi. Skaitiniai eksperimentai atlikti su virtualiu lygiagrečiuoju kompiuteriu, naudojant paketą PVM [1].

1. LU faktorizacijos blokinis algoritmas

Sudarant lygiagrečiuosius tiesinės algebros algoritmus plačiai naudojami blokiniai algoritmai [2]. Sakykime, kad A yra $N \times N$ matrica, kurią padalijame į $r \times r$ dimensijos blokus. Viso gauname $M = N/r$ blokinių stulpelių ir eilučių:

$$A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1M} \\ A_{21} & A_{22} & \dots & A_{2M} \\ \dots & \dots & \dots & \dots \\ A_{M1} & A_{M2} & \dots & A_{MM} \end{pmatrix}.$$

Nagrinėsime matricos A išskaidymą

$$A = LU,$$

čia L yra apatinė trikampė matrica su vienetine įstrižaine, o U yra viršutinė trikampė matrica

$$L = \begin{pmatrix} L_{11} & 0 & \dots & 0 \\ L_{21} & L_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ L_{M1} & L_{M2} & \dots & L_{MM} \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1M} \\ 0 & U_{22} & \dots & U_{2M} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & U_{MM} \end{pmatrix}.$$

LU faktorizacijos algoritmas

Tarkime, kad jau atlikome $k-1$ algoritmo žingsnį. Pateiksime k-ojo žingsnio formules:

1. Vienu neblokiniu algoritmu išskaidome matricą A_{kk}

$$A_{kk} = L_{kk}U_{kk}. \tag{1}$$

2. Apskaičiuojame matricos L k-jo stulpelio blokus

$$L_{ik} = A_{ik}U_{kk}^{-1}, \quad i = k+1, \dots, M, \tag{2}$$

šiame žingsnyje sprendžiame tiesinių lygčių sistemas su viršutine trikampe matrica U_{kk}.

3. Apskaičiuojame matricos U k-osios eilutės blokus
$$U_{ki} = L_{kk}^{-1} A_{ki}, \quad i = k+1, \dots, M, \tag{3}$$
šiame žingsnyje sprendžiame tiesinių lygčių sistemas su apatine trikampe matrica L_{kk}.

4. Modifikuojame matricą $A(kr+1:N, kr+1:N)$
$$\tilde{A}_{k+1,k+1} = \tilde{A}_{k+1,k+1} - \begin{pmatrix} L_{k+1,k} \\ L_{k+2,k} \\ \dots \\ L_{M,k} \end{pmatrix} \Big(U_{k,k+1} \dots U_{k,M} \Big).$$

Didžioji skaičiavimų dalis yra atliekama 4 žingsnyje. Bendras algoritmo sudėtingumas yra $\frac{N(N^2-1)}{3}$ daugybos veiksmų ir tiek pat sudėties veiksmų.

2. Lygiagretusis LU faktorizacijos algoritmas

Matrica A yra paskirstoma tarp p procesorių. Tarkime, kad procesoriai sudaro dvimatį masyvą ir $p = PQ$, t.y. P procesorių yra kiekviename stulpelyje ir Q procesorių kiekvienoje eilutėje. Jei $P = 1$ tai gauname vienmatį procesorių vektorių.

Nagrinėsime ciklinį matricos A stulpelių-blokų ir eilučių-blokų paskirstymo algoritmą, tada visoje uždavinio sprendimo eigoje gauname tolygų darbo paskirstymą tarp procesorių. Ciklinis paskirstymo algoritmas gerai žinomas, kai procesoriai yra homogeniški. Tada M objektų paskirstome tarp P procesorių pagal formulę
$$m \longrightarrow (i, j) = \Big(m \bmod P, \lfloor \frac{m}{P} \rfloor \Big),$$
čia j yra procesoriaus numeris, o $\lfloor \frac{m}{P} \rfloor$ – objekto lokalus numeris i-ajame procesoriuje. Matricos A paskirstymą gauname kaip stulpelių ir eilučių ciklinio paskirstymo sandaugą.

Šiame darbe pateiksime ciklinio paskirstymo algoritmą, kai turime heterogeninį lygiagretųjį kompiuterį. Pažymėkime procesorių greičius $w_0, w_1, \dots, w_{p-1}$. Tada ciklinio paskirstymo algoritmas yra toks:

do $j = 0, P-1$
  $M_j = 0$
  $t_j = \frac{1}{w_j}$
end do
do $i = M, 1, -1$
  $t_k = \min\limits_{0 \le j \le p-1} \{t_j\}$
  $Z_i = k$
  $M_k = M_k + 1$ (i-asis objektas skirtas k-jam procesoriui)
  $t_k = \frac{M_k + 1}{w_k}$
end do

Vektoriaus komponentėje Z_i saugome informaciją apie i-ojo objekto šeimininką. Heterogeninio lygiagrečiojo kompiuterio atveju visada imsime $P = 1, Q = p$ procesorių topologiją.

LU faktorizacijos lygiagretusis algoritmas

1. Procesorius (i, j), kuriame saugomas matricos blokas A_{kk}, apskaičiuoja šio bloko išskaidymą (1). Po to jis nusiunčia matricą U_{kk} j-ojo stulpelio procesoriaus, o matricą L_{kk} i-osios eilutės procesoriaus.
2. Procesoriai $(i, j), i = 0, 1, \dots, P-1$ apskaičiuoja matricos k-ojo stulpelio blokus L_{ik} (2) ir nusiunčia juos kitiems i-osios eilutės procesoriams.
3. Procesoriai $(i, j), i = 0, 1, \dots, Q-1$ apskaičiuoja matricos k-ojo eilutės blokus U_{ik} (3) ir nusiunčia juos kitiems j-osios stulpelio procesoriams.
4. Visi procesoriai lygiagrečiai modifikuoja matricą $A_{k+1,k+1}$ (4).

Pažymėkime T_k laiką, per kurį realizuojame LU faktorizacijos algoritmą su k procesorių, tada

$$S_p = \frac{T_1}{T_p}$$

yra lygiagrečiojo algoritmo pagreitėjimo koeficientas.

TEOREMA 1. *Jei turime homogeninį lygiagretųjį kompiuterį su p procesorių, tai LU faktorizacijos lygiagrečiojo algoritmo su cikliniu duomenų paskirstymu pagreitėjimas*

$$S_p \longrightarrow p, \quad \text{kai} \quad N \to \infty. \tag{4}$$

Įrodymas. Kadangi LU faktorizacijos algoritmo aritmetinių veiksmų skaičius yra $O(N^3)$ eilės dydis, o duomenų persiuntimo laikas yra $O\Big(p\big(\alpha+\beta N\big)N\Big)$ eilės dydis, tai pakankamai dideliems N duomenų perdavimo laikas sudaro kiek norimą mažą bendro skaičiavimo laiko dalį. Atlikę nesudėtingus kombinatorikos skaičiavimus, gauname tokį LU faktorizacijos algoritmo realizacijos laiką:

$$T_1 = \frac{N(N^2-1)}{3}.$$

Įvertinsime šio algoritmo realizacijos laiką, kai skaičiavimuose naudojame p procesorių. Nagrinėsime procesorių topologijos atvejį, kai $Q \le P$. Kiti atvejai tiriami analogiškai. Iš lygiagrečiojo LU faktorizacijos algoritmo formulavimo turime, kad

$$T_p = \sum_{k=1}^{M} T_p(k),$$

čia $T_p(k)$ yra k-ojo algoritmo žingsnio įvykdymo laikas. Jis susideda iš keturių dalių. Pirmosios dalies laikas yra $r(r^2-1)/3$, antrosios ir trečiosios dalių laikas yra $O(Mr^3/P)$. Ciklinio paskirstymo atveju matricos $\tilde{A}_{k+1,k+1}$ modifikavimo laikas yra

$$T_{\mathrm{mod}}(k) = \Big(\frac{M}{P} - \Big\lfloor\frac{k}{P}\Big\rfloor\Big)\Big(\frac{M}{Q} - \Big\lfloor\frac{k}{Q}\Big\rfloor\Big)r^3.$$

Tada gauname tokį T_p įvertį

$$\begin{aligned} T_p &= M\frac{r(r^2-1)}{3} + 2(M-1)O\Big(\frac{M}{P}r^3\Big) + \sum_{k=1}^{M-1}\Big(\frac{M}{P} - \Big\lfloor\frac{k}{P}\Big\rfloor\Big)\Big(\frac{M}{Q} - \Big\lfloor\frac{k}{Q}\Big\rfloor\Big)r^3 \\ &\quad + \Big(\frac{M^3}{3PQ} + \frac{M^2}{4}\frac{Q+P-4}{PQ} + \frac{M(3P-Q)}{12p}\Big)r^3 + O(M^2r^3) \\ &= \frac{N^3}{3p} + O(N^2r). \end{aligned}$$

Todėl, kai p ir r yra fiksuoti, imdami $N \to \infty$ įrodome (1) lygybę. □

Heterogeninio lygiagrečiojo kompiuterio atveju pagreitėjimo analizę atlikome panaudodami teorinį skaičiavimo modelį, kuriame įvertinome tik skaičiavimo laiką ir neatsižvelgėme į duomenų siuntimo laiką. Šio modelio algoritmas yra toks:

$$\begin{aligned} & T = 0 \\ & \text{do } k = 0, M-1 \\ & \quad j = S_k \\ & \quad \text{do } \ i = 0, r-1 \\ & \qquad T = T + \frac{\big((M-k)r-i-1\big)}{w_j} \\ & \quad \text{end do} \\ & \quad M_j = M_j - 1 \\ & \quad T = T + \max_{0\leqslant i\leqslant p-1} \frac{\big((M-k-0.5)M_i r^3\big)}{w_i} \\ & \text{end do} \end{aligned}$$

Šio teorinio modelio rezultatų analizė leidžia suformuluoti tokį teiginį: *jei greičiausiam procesoriui tenka nemažiau kaip 10 stulpelių-blokų, tai lygiagrečiojo algoritmo efektyvumas yra nemažesnis už 85 % heterogeninio lygiagrečiojo kompiuterio galingumo.*

Lentelėje 1 yra pateikti rezultatai, gauti panaudojant šį teorinį modelį, kai $p = 3$, o procesorių greitaeigiškumas yra

$$w_0 = 1, \quad w_1 = 0.25, \quad w_2 = 0.05.$$

Tada suminis lygiagrečiojo kompiuterio greitaeigiškumas yra $W_3 = 1.3$. Čia M_i pažymėjome stulpelių–blokų skaičių ciklinio paskirstymo algoritme tekusį i-jam procesoriui, E_3 yra algoritmo efektyvumas

$$E_3 = \frac{T_0}{T_3 W_3}.$$

Lentelė 1. Homogeninio ciklinio paskirstymo algoritmo efektyvumo analizė.

M	M_0	M_1	M_2	E_3
_4	4	0	0	0.77
10	8	2	0	0.81
15	12	3	0	0.86
30	24	5	1	0.90

Lentelė 2. Homogeninis lygiagretusis kompiuteris, $N = 3200$.

p	P	Q	T_p	S_p
_1	1	1	158.0	1
4	1	4	43.3	3.65
4	2	2	42.0	3.76
6	1	6	30.6	5.16
6	2	3	28.4	5.56
16	1	16	18.4	8.58
16	4	4	12.1	13.06

Lentelė 3. Heterogeninis lygiagretusis kompiuteris, $p = 3$.

$w_0 : w_1 : w_2$	W_3	T_1	T_3	S_{w_3}	E_{w_3}
_1:1:1	3	277	103	2.69	0.896
1:1:0.76	2.76	212	92	2.30	0.835
1:0.95:0.72	2.67	201	90	2.24	0.837

3. Skaičiavimo eksperimento rezultatai

Pradžioje nagrinėsime rezultatus, gautus homogeniniu lygiagrečiuoju superkompiuteriu SP2 Danijos UNI-C skaičiavimo centre. Lentelėje 2 pateikti LU faktorizacijos algoritmo realizacijos laikai, kai $N = 3200$, $r = 40$.

Pastebėsime, kad procesorių topologijos 1×16 atveju vienam procesoriui teko tik po 5 stulpelius-blokus, todėl darbo paskirstymas tarp procesorių skaičiavimų eigoje nebuvo tolygus. Naudodami procesorių topologiją 4×4 uždavinį išsprendžiame 1.5 karto greičiau. Sprendžiant didesnį uždavinį su tokiu pačiu procesorių skaičiumi priklausomybė nuo procesorių topologijos mažėja. Pavyzdžiui, imdami $N = 6400$, $r = 405$ gavome, kad $T_{1\times 16}/T_{4\times 4} = 1.25$.

Dabar pateiksime rezultatus, gautus heterogeniniu virtualiuoju lygiagrečiuoju kompiuteriu, sudarytu iš 3 skirtingų kompiuterių. LU faktorizacijos uždavinį, kai $N = 1200$, $r = 30$, pirmasis kompiuteris išsprendė per 201 sekundę, antrasis – per 212 sekundžių, trečiasis – per 277 sekundes. Lentelėje 3 yra pateikti skaičiavimo laikai bei pagreitėjimo ir efektyvumo koeficientų reikšmės, čia w_i yra kompiuterio greitaeigiškumo koeficientas, naudotas cikliniame duomenų paskirstymo algoritme, W_3 yra suminis kompiuterio greitaeigiškumas.

Lygiagretieji algoritmai yra ypač efektyvūs, kai naudodami daugiau procesorių sprendžiame vis didesnį uždavinį. Pastebėsime, kad šiuo atveju laiką T_1 galime įvertinti tik ekstrapoliuodami, nes vieno procesoriaus atmintyje negalime patalpinti visą matricą. Procesorių heterogeniškumo svorius ω_i imsime proporcingus procesoriaus turimos atminties dydžiui.

LITERATŪRA

[1] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM: Parallel Virtual Machine*, MIT Press, Cambridge, Massachusetts, London, 1993.

[2] J. Choi, J. Dongarra, and D. Walker, The design of a parallel dense linear algebra software library, *Numerical Algorithms*, **10** (1995), 379–399.

A parallel algorithm of LU factorization

R. Čiegis, V. Starikovičius

This paper discusses issues in the design of LU factorization parallel algorithm on distributed memory parallel computers. A new cyclic distribution method is proposed for heterogeneous parallel computers, which include virtual parallel computers. The efficiency of the algorithm is investigated and results of computational experiments are given.