LMD mokslo darbai, III tomas, 414–418

Teorinis LU faktorizacijos lygiagrečiojo algoritmo efektyvumo modelis

Raimondas ČIEGIS (MII, VGTU), Vadimas STARIKOVIČIUS (VU,VGTU)
el. paštas: rc@fm.vtu.lt

1. Uždavinio formulavimas

Lygiagrečiųjų algoritmų efektyvumas priklauso nuo kelių faktorių: aritmetinių operacijų tolygaus paskirstymo tarp procesorių, sinchronizacijos kaštų minimizavimo, duomenų persiuntimo tarp procesorių minimizavimo. Į sinchronizacijos kaštus įtraukiame ir algoritmo dalies, kurią galime vykdyti tik nuosekliai, aštus. Teorinis tokio algoritmo modelis turi įvertinti visus faktorius. Toks įrankis vėliau leidžia prognozuoti algoritmo vykdymo laiką ir parinkti optimalų procesorių skaičių. Darbe [1] buvo sudarytas LU faktorizacijos lygiagrečiojo algoritmo teorinis modelis, tačiau jame buvo padaryta prielaida, kad duomenų persiuntimo kaštai yra maži lyginant su skaičiavimo ir sinchronizavimo laiku, todėl jie buvo neįtraukti į teorinį modelį. Tokia prielaida dažnai sutinkama ir kituose darbuose, analizuojančiuose lygiagrečiuosius algoritmus. LU faktorizacijos algoritmo atveju skaičiavimo kaštai yra $O(N^3/p)$ eilės dydis, o duomenų perdavimo kaštai – $O(pN^2)$ eilės dydis, čia N yra matricos eilučių arba stulpelių skaičius, p – procesorių skaičius. Tačiau šie kaštai esminiai priklauso ne tik nuo dydžio eilės, bet ir nuo konstantų reikmių, įeinančių į šių išraiškų apibrėžimus, reikšmių. Todėl duomenų persiuntimo kaštai gali būti svarbūs net ir tada, kai $N \geqslant 2500$, o $p \leqslant 5$, jei lygiagrečiojo kompiuterio procesorių skaičiavimo greitis yra daug didesnis už duomenų persiuntimo greičius. Tokia situacija yra tipiška, kai nagrinėjame virtualiuosius lygiagrečius kompiuterius, sudarytus iš kompiuterių klasterio, sujungto lokaliuoju tinklu.

Šiame darbe nagrinėjamas lygiagrečiojo LU faktorizacijos algoritmo teorinis modelis, kuriame įvertinami duomenų persiuntimo kaštai. Naudojantis šiuo įrankiu sprendžiami optimalaus procesorių skaičiaus parinkimo ir duomenų paskirstymo tarp procesorių uždaviniai.

Sudarydami lygiagretųjį LU faktorizacijos algoritmą, naudosime blokinį algoritmą [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

$$A = LU, \tag{1.1}$$

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

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

2. LU faktorizacijos lygiagretusis algoritmas

Tarkime, kad p procesorių sudaro vienmatį vektorių. Dvimačio procesorių masyvo atvejis buvo nagrinėjamas [1] darbe. Matricos A stulpeliai-blokai yra paskirstomi tarp p procesorių. Tegul $map(i)$ apibrėžia i-ajam procesoriui priklausančių stulpelių numerius.

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

1. i-tasis procesorius, kuriame saugomas k-tasis matricos stulpelis, t.y. $map(i) = k$, apskaičiuoja bloko A_{kk} išskaidymą

$$A_{kk} = L_{kk}U_{kk}, \tag{2.1}$$

ir po to apskaičiuoja matricos L k-ojo stulpelio blokus

$$L_{ik} = A_{ik}U_{kk}^{-1}, \quad i = k+1, \dots, M, \tag{2.2}$$

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

2. i-tasis procesorius nusiunčia kitiems procesoriams, kurie dar nebaigė darbo, matricos L k-ąjį stulpelį L_{ik}, $i = k, \dots, M$.

3. Visi procesoriai lygiagrečiai modifikuoja matricą A_{ij}, $i = k, \dots, M$, $j = k+1, \dots, M$

$$\tilde{A}_{k,k+1} = \tilde{A}_{k,k+1} - \begin{pmatrix} L_{k,k} \\ L_{k+1,k} \\ \dots \\ L_{M,k} \end{pmatrix} (U_{k,k+1} \dots U_{kM}), \tag{2.3}$$

čia pažymėjome $\tilde{A}_{k,k+1} = A(kr+1 : N, (k+1)r+1 : N)$. Šiame žingsnyje atliekama pagrindinė skaičiavimų dalis.

Sudarysime lygiagrečiojo LU faktorizacijos algoritmo teorinį skaičiavimo modelį. Papildomai apibrėžiame tris konstantas α, β, γ, nuo kurių priklauso šis modelis Čia α ir β apibrėžia duomenų persiuntimo tarp dviejų procesorių kaštus

$$T_{send} = \alpha + \beta n,$$

α yra pasirengimo perduoti duomenis laikas, β – vieno skaičiaus persiuntimo kaštai, γ – vienas tipinės operacijos *daugyba+sudėtis* atlikimo greitis. Teoriniame modelyje įvertinami algoritmo 1 kiekvieno žingsnio vykdymo kaštai.

Algoritmas 2. *Lygiagrečiojo algoritmo vykdymo laikas*

$T_i = 0,\ i = 1, 2, \ldots, p$
$M_i = |map(i)|$
do $k = 1, M$
  $map(master) = k$
  $T_{master} = T_{master} + \frac{r(r^2-1)}{3}\gamma$
  $T_{master} = T_{master} + (M - K)\frac{r^2(r-1)}{2}\gamma$
  $M_{master} = M_{master} - 1$
  do $j = 1, p$
    if $((j \neq master)\&\&(M_j > 0))$
      $T_{master} = T_{master} + \alpha + \left((M-k)r^2 + \frac{r(r-1)}{2}\right)\beta$
      $T_j = T_{master}$
    end if
  end do
  do $j = 1, p$
    if $(M_j > 0)$
    $T_j = T_j + M_j\left(\frac{r^2(r-1)}{2} + (M-K)r^3\right)\gamma$
  end do
end do
$map(master) = M$
$T = T_{master}$

Pastaba 1. Šiame modelyje padarėme prielaidą, kad *broadcast* tipo duomenų persiuntimas realizuotas, panaudojant sinchronizuotą ciklinį duomenų persiuntimą. Vienas iš sudėtingiausių tokio teorinio modelio sudarymo etapų yra duomenų persiuntimo modelio identifikavimas, kai naudojamos specializuotos *broadcast* funkcijos realizacijos. Daugeliu atvejų standartas nenurodo, kaip šios funkcijos turi būti realizuotos, o apibrėžia tik tokios funkcijos rezultatą (žr. pvz. PVM arba MPI standartus).

3. Skaičiavimo eksperimento rezultatai

Pradžioje nagrinėsime rezultatus, gautus virtualiuoju lygiagrečiuoju kompiuteriu, sudarytu iš RS6000 darbo stočių, sujungtų lokaliuoju tinklu. Tada pagalbiniame skaičiavimo eksperimente buvo rastos tokios parametrų α, β, γ skaitinės reikšmės

$$\alpha = 1 \cdot 10^{-3},\ \beta = 8 \cdot 10^{-6},\ \gamma = 4 \cdot 10^{-8}.$$

Tuo atveju, kai neatsižvelgiame į duomenų perdavimo kaštus (t.y. $\alpha = 0, \beta = 0$), optimalus duomenų paskirstymas tarp procesorių yra *ciklinis* [1]

$$\mathrm{map}(i) = \{j : j \bmod p = i - 1\}. \tag{3.1}$$

Turėdami teorinį LU faktorizacijos lygiagrečiojo algoritmo sudėtingumo modelį, galime spręsti duomenų paskirstymo tarp procesorių uždavinį ir atsižvelgti į duomenų persiuntimo kaštus. Pastebėsime, kad skirtingų atvaizdavimų $map(i)$ yra $O(p^M)$, todėl tikslus tokio uždavinio sprendimas yra labai sudėtinga problema. Mes susiaurinsime leistinų paskirstymų aibę ir nagrinėsime rekursyvinį paieškos algoritmą.

Pažymėkime $T(p, M, r, map)$ laiką, per kurį išsprendžiame LU faktorizacijos uždavinį, kai stulpeliai paskirstyti tarp p procesorių atvaizdavimu $map[i]$, $i = 1, 2, \ldots, p$.

Algoritmas 3. *Matricos paskirstymas tarp procesorių.*

$M_{local} = M,\ T_{opt} = \infty$
do $i = p, 2, -1$
  do $j = 1, M_{local}$
    stulpelius nuo $\max(M - M_{local} + 1, M - i * j + 1)$ iki M
    paskirstome tarp i procesorių blokiniu būdu po j blokų
    likusius stulpelius nuo $M - M_{local} + 1$ iki $M - i * j$
    paskirstome ciklišKai
    if $(T(p, M, r, map) < T_{opt})$
      $T_{opt} = T(p, M, r, map)$
      $map_{opt} = map$
      $M_{new} = M - \max(M - M_{local} + 1, M - i * j + 1) + 1$
    end if
  end do
  $M_{local} = M_{new}$
end do

Lentelėje 1 yra pateikti prognozuojami T_{mod} ir realūs T_{real} LU faktorizacijos laikai, kai $M = 100$, $r = 30$ ir naudojame ciklinį duomenų paskirstymą, gautą algoritmu 3. Iš šių rezultatų matome, kad teorinis modelis teisingai prognozuoja, kad ciklinio duomenų paskirstymo atveju optimalus procesorių skaičius yra $p = 3$. Panaudodami algoritmą 3, gauname geresnį duomenų paskirstymą ir imdami $p = 4$ sumažiname skaičiavimo laiką.

1 lentelė.
RS6000 klasteris – virtualusis lygiagretusis kompiuteris

	ciklinis		optimalus	
p	T_{mod}	T_{real}	T_{mod}	T_{real}
1	234	235	234	345
2	155	157	152,9	154,1
3	152	153	128,7	128,7
4	170	166	121,7	118,4
5	194	194	121,7	118,4

Po to skaičiavimai buvo pakartoti SP2 superkompiuteriu, kurio koeficientai α, β, γ yra tokie

$$\alpha = 2 \cdot 10^{-4},\ \beta = 3,5 \cdot 10^{-7},\ \gamma = 1,54 \cdot 10^{-8}.$$

Šiuo atveju ir algoritmu 3 gautieji duomenų paskirstymai buvo labai artimi cikliniam duomenų paskirstymui.

Literatūra

[1] R. Čiegis, V Starikovičius, LU faktorizacijai lygiagretusis algoritmas, *LMD konf. darbai*,**3**, 121–126 (1998).
[2] J. Choi, J. Dongarra, D. Walker, The design of a parallel dense linear algebra software library, *Numerical Algorithms*, **10**, 379–399 (1995).

A tool for the analysis of the efficiency of a parallel LU factorization algorithm

R. Čiegis, V. Starikovičius

This paper considers issues on the design of a tool for the analysis of a parallel LU factorization algorithm. The costs of data communication are included into the theoretical model. Problems of defining the optimal number of processors and efficient data distributions among processors are investigated. Results of computational experiments are given.