ISSN : 2302-3805

KARAKTERISTIK KINERJA ALGORITMA *RECURSIVE DECOUPLING* PADA SISTEM MULTIPROSESOR BERBASIS PVM

Tri Prabawa

Program Studi Teknik Informatika, STMIK AKAKOM Yogyakarta
Jl. Raya Janti 143, Yogyakarta 55198

E-mail : tprabawa@akakom.ac.id

Abstrak

Tulisan ini membahas penyelesaian sistem Au = d, dengan matriks koefisien A tridiagonal, dengan metode recursive decoupling pada sistem multiprosesor. Sistem persamaan linier tersebut diperoleh dari hasil diskritisasi persoalan yang berbentuk persamaan differensial parsial. Ide dasar metode pemisahan rekursif adalah menurunkan baris-baris independen dengan berdasarkan pada strategi rank-one updating dan proses partisi berulang pada sistem matriks sehingga didapat bentuk matriks diagonal blok yang masing-masing blok berukuran 2x2.

Pemecahan masalah pada sistem komputasi paralel adalah dengan cara melakukan dekomposisi persoalan secara algoritmik atau geometrik, sehingga dapat diidentifikasi karakteristik paralelisasinya. Karakteristik kinerja algoritma paralel dapat dilihat dari pengukuran waktu eksekusi, rasio komputasi dan komunikasi, speed-up, dan tingkat efisiensi. Untuk mengetahui karakteristik ini, maka algoritma recursive decoupling diimplementasikan pada sistem parallel virtual machine (PVM). PVM adalah sebuah perangkat lunak yang digunakan untuk pembuatan jaringan komputer paralel. Perangkat lunak ini didesain sedemikian rupa sehingga mengizinkan sebuah jaringan komputer heterogen, yang terdiri atas beberapa mesin dengan sistem operasi Windows atau Unix sehingga dapat digunakan sebagai sebuah model prosesor paralel tunggal yang terdistribusi.

Dari hasil uji coba terlihat bahwa terjadi kenaikan percepatan seiring dengan bertambahnya jumlah prosesor yang dipakai. Percepatan berkisar antara 1,61 (2 prosesor) sampai dengan 5,90 (8 prosesor). Namun sebaliknya, dengan bertambahnya jumlah prosesor yang dipakai terjadi penurunan efisiensi. Tingkat efisiensi mencapai 80,43% (2 prosesor) dan terendah 36,25% (8 prosesor).

Kata kunci : sistem persamaan linier tridiagonal, recursive decoupling, speed-up, efisiensi, dan PVM.

1. Pendahuluan

Dalam beberapa aplikasi, penggunaan model matematika menjadi amat populer, karena teknik ini banyak dipakai dalam pemodelan dari pelbagai persoalan nyata. Dengan model matematika bentuk persoalannya menjadi jelas dan sederhana, serta metode dan analisisnya lebih dapat dipertanggung jawabkan. Banyak pemodelan dari suatu fenomena fisik; seperti mekanika fluida, penjalaran panas dan lain sebagainya; biasanya memberikan berbagai macam bentuk persamaan differensial parsial (PDP). Pada umumnya, kecuali dalam hal yang amat sederhana, penyelesaian secara analitik dari suatu PDP sulit diperoleh, sehingga perlu dicari solusi numeriknya sebagai alternatif jawaban.

Meskipun solusi numerik memerlukan banyak perhitungan, namun pada perkembangannya metode numerik telah memberikan hasil yang berarti, terutama setelah didukung dengan pemakaian perangkat komputer digital. Kebutuhan akan kecepatan penyelesaian masalah menjadi amat penting terutama untuk persoalan yang cukup besar dan kompleks, serta informasinya segera diperlukan. Penyelesaian yang diinginkan dapat dikerjakan secara cepat, dengan kontribusi komputer, dan jika memungkinkan diproses secara paralel.

Komputer paralel adalah suatu perangkat komputer yang mempunyai sejumlah alat pemroses (disebut prosesor) yang saling bekerja sama dalam suatu koordinasi program kendali [1]. Adanya arsitektur seperti ini memungkinkan suatu masalah diselesaikan secara paralel. Dalam menggunakan arsitektur komputer yang demikian maka kecepatan algoritma sangat ditentukan oleh jumlah prosesor yang dipakai serta pola hubungan interkoneksi antara prosesor yang satu dengan yang lain.

2. Dasar Teori

Diberikan suatu sistem persamaan linier dari sistem tridiagonal Au = d, dengan A matriks tridiagonal dan dapat ditulis sebagai berikut:

$$\begin{pmatrix} b1\ c1 & & & & & \\ a2\ b2\ c2 & & 0 & & & \\ & a3\ b3\ c3 & & & & \\ & & a4\ b4\ c4 & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & 0 & & & bn-1 & cn-1 \\ & & & & an & bn \end{pmatrix} \begin{pmatrix} u1 \\ u2 \\ \\ \cdot \\ \\ \cdot \\ un-1 \\ un \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ \\ \cdot \\ \\ \cdot \\ dn-1 \\ dn \end{pmatrix} \qquad (1)$$

Persoalannya adalah mencari vektor u yang memenuhi sistem persamaan Au = d, yang banyak dijumpai sebagai hasil diskritisasi suatu PDP dengan memakai metode beda hingga atau elemen hingga dengan kondisi syarat batas tertentu.

Solusi analitis sistem persamaan tersebut ada dan dapat ditulis $u = A^{-1} d$, jika matriks A nonsingular [9]. Sistem persamaan (1) dapat diselesaikan baik secara langsung maupun iteratif. Penyelesaian secara langsung memerlukan $\theta(n^2)$ *flops* sedangkan iteratif $\theta(n)$ / langkah dan menjadi mahal jika terjadi konvergensi yang lamban [7]. Nampak bahwa jika untuk n besar, kedua cara ini memerlukan waktu komputasi yang besar dan kurang efisien.

Metode *Recursive Decoupling* Untuk Sistem Tridiagonal

Metode pemisahan rekursif berdasarkan pada strategi perubahan rank-satu (*rank-one updating*) dan proses partisi berulang pada sistem matriks sehingga didapat bentuk matriks diagonal blok yang masing-masing blok berukuran 2x2. Metode ini akan dipakai untuk menyelesaikan bentuk matriks tridiagonal sembarang, dan untuk sederhananya dipilih matriks dengan ukuran $n = 2^k$ dengan k = bulat positif.

Proses Partisi

Pandang sistem persamaan (1) yang selanjutnya dinyatakan sebagai :

$$\begin{pmatrix} b1 & c1 & & & & \\ a2 & b2 & c2 & & 0 & \\ & a3 & b3 & c3 & & \\ & & a4 & b4 & c4 & \\ & & & \cdot & \cdot & \\ & & & \cdot & \cdot & \\ 0 & & & & bn-1 & cn-1 \\ & & & & an & bn \end{pmatrix} \begin{pmatrix} u1 \\ u2 \\ \cdot \\ \cdot \\ un-1 \\ un \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ \cdot \\ \cdot \\ dn-1 \\ dn \end{pmatrix} \qquad (2)$$

Dengan $|bi| \geq |ai| + |ci|$ untuk setiap i = 1, 2, 3, ...n, syarat ini menjamin proses reduksinya berjalan stabil. Proses partisi dilakukan terhadap matriks koefisien A sedemikian sehingga diperoleh bentuk blok matriks berukuran 2x2 sebagai berikut

$$\begin{pmatrix} e1 & c1 & & & & \\ a2 & e2 & & & 0 & \\ & & e3 & c3 & & \\ & & a4 & e4 & & \\ 0 & & & & en-1 & cn-1 \\ & & & & an & en \end{pmatrix} + \sum_{j=1}^{m-1} \begin{pmatrix} x1 \\ x2 \\ \cdot \\ \cdot \\ xn-1 \\ xn \end{pmatrix}^{j} x \begin{pmatrix} y1 \\ y2 \\ \cdot \\ \cdot \\ yn-1 \\ yn \end{pmatrix}^{(j)T} \qquad (3)$$

Dengan $e_1 = b_1$, $e_{2j-1} = b_{2j-1} - c_{2j-2}$, j = 2, 3, ...m
$e_n = b_n$, $e_{2j} = b_{2j} - a_{2j+1}$, j =1, 2, ...m-1, dan m = n/2
Sedangkan vektor $x^{(j)}$ dan $y^{(j)}$merupakan vektor dengan elemen tidak nol hanya pada elemen ke 2j dan 2j+1 yang didefinisikan sebagai

$$x_k = \begin{cases} 1, untuk\ k = 2j, 2j+1 \\ 0,\ untuk\ yang\ lain \end{cases} \qquad (4)$$

$$y_k = \begin{cases} a2j+1,\ untuk\ k = 2j \\ c2j, untuk\ k = 2j+1 \\ 0,\ untuk\ yang\ lain \end{cases} \qquad (5)$$

Atau dapat ditulis sebagai
$x(j) = (0, 0, ..., 1, 1, ...0)^T$ (6)
$y(j) = (0, 0, ...,a_{2j+1}, c_{2j}, ...0)^T$ (7)
Persamaan (3) secara sederhana dapat ditulis menjadi
$A = J + x^{(j)} y^{(j)T}$
Dengan J berbentuk

$$J = \begin{pmatrix} J1 & & & & \\ & J2 & & & \\ & & J3 & & \\ & & & Jm & \\ & & & & Jm \end{pmatrix} \qquad (8)$$

Sedangkan J1, J2, ...Jm terdiri dari submatrik 2x2 berbentuk

$\begin{pmatrix} e2j-1 & c2y-1 \\ a2j & e2j \end{pmatrix}$ dengan j = 1, 2, ...m dan m = n/2

Ide dasar partisi ini mengacu pada metode inversi Sherman-Morrison yang bertujuan untuk mereduksi kompleksitas komputasi A^{-1} secara langsung. Menurut Golub [7] metode inversi Sherman-Morrison memberikan formula perhitungan invers matrik A berbentuk $A = (J + uv^T)$ dengan A elemen R^{nxn} , Sedangkan u dan v vektor berukuran nx1, Invers matrik A didefinisikan sebagai

$$A^{-1} = (J + uv^T)^{-1} = J^{-1} - (J^{-1} u\ v\ J^{-1}) / (1 + v^T J^1 u) \qquad (9)$$

Dengan memakai persamaan tsb, maka invers matrik $A = J + xy^T$ adalah

$$A^{-1} = (J + xy^T) = J^{-1} - \alpha (J^{-1} x) (y^T J^1) \qquad (10)$$

Dengan $\alpha = 1/ (1 + y^T J^{-1} x)$
Kemudian dengan menghitung invers A, solusi persamaan (2) adalah

$$\begin{aligned} U &= A^{-1} d = (J + xy^T)^{-1} d \\ &= \{J^{-1} - \alpha (J^{-1} x) (y^T J^1)\} d \\ &= J^{-1} d - \alpha\ J^{-1} x\ y^T J^1 d \\ &= J^{-1} d - \alpha\ (J^{-1} x)\ y^T (J^{-1} d) \qquad (11) \end{aligned}$$

dengan $\alpha = 1/ (1 + y^T J^{-1} x)$

Dengan menghitung nilai-nilai (11), yaitu $J^{-1}x$, J^{-1} d dan α diperoleh solusi dar sistem persamaan (2).

Proses Pemisahan Rekursif.

ISSN : 2302-3805

Proses ini bertujuan mengubah vektor u dan d sedemikian hingga elemen-elemennya menjadi berpasangan (*couple*) yang dapat ditulis

$$u = \begin{pmatrix} \left.\begin{matrix} u1 \\ u2 \end{matrix}\right\} \\ \left.\begin{matrix} u3 \\ u4 \end{matrix}\right\} \\ . \\ . \\ \left.\begin{matrix} un-1 \\ un \end{matrix}\right\} \end{pmatrix} = \begin{pmatrix} u1 \\ u2 \\ . \\ . \\ dm \end{pmatrix} \qquad (12)$$

$$\text{dan } d = \begin{pmatrix} \left.\begin{matrix} d1 \\ d2 \end{matrix}\right\} \\ \left.\begin{matrix} d3 \\ d4 \end{matrix}\right\} \\ . \\ . \\ \left.\begin{matrix} dn-1 \\ dn \end{matrix}\right\} \end{pmatrix} = \begin{pmatrix} d1 \\ d2 \\ . \\ . \\ dm \end{pmatrix} \qquad (13)$$

dari persamaan (11) solusi Au = d dapat dinyatakan

$$\begin{aligned} u &= (J + \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1} d \\ &= (J^{-1} - \alpha J^{-1} \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1}) d \\ &= (Jd - \alpha J^{-1} \sum_{j=1}^{m-1} x^{(j)} y^{(j)T})^{-1} d \end{aligned} \qquad (14)$$

dengan

$$\alpha = 1/(1 + \sum_{j=1}^{m-1} y^{(j)T}) J^{-1} \sum_{j=1}^{m-1} x^{(j)})$$

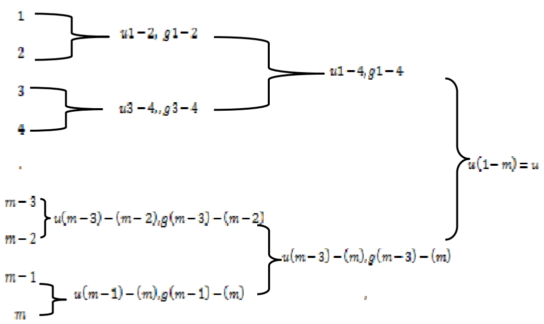
Dengan menyatakan J^{-1} d' = u' dan J^{-1} $x^{(j)} = g^{(j)}$ maka bentuk (14) dapat disederhanakan menjadi

$$u = (u' - \alpha g^{(j)} y^{(j)T} u') = (1 - \alpha g^{(J)} (y^{(j)T}) u' \quad \ldots(15)$$

dengan $\alpha = 1/(1 + y^{(j)T} g^{(j)})$ dan j = 1, 2, ….m-1

persamaan (15) merupakan bentuk formula perubahan rank satu, maka tahapan solusi untuk persamaan diatas adalah

(a) Mencari solusi vektor u' dari persamaan u' = J^{-1} d
(b) Mencari solusi vektor $g^{(j)}$ dari persamaan $g^{(j)} = J^{-1} x^{(j)}$
(c) melakukan perubahan rank satu (15)



3. Pemecahan Masalah Secara Paralel

Dari persoalan diatas karena matriks koefisien memiliki struktur yang spesifik, maka dimungkinkan pemakaian metode recursive decoupling yang penyelesaiannya dapat dikerjakan secara paralel. Komputer paralel adalah suatu perangkat komputer yang mempunyai sejumlah alat pemroses (disebut prosesor) yang saling bekerja sama dalam suatu koordinasi program kendali [1]. Model komputer yang sering dipertimbangkan sebagai sistem multiprosesor didefinisikan sebagai SIMD (*single instruction multiple data*) dan MIMD (*multiple instruction multiple data*).

Dalam menggunakan arsitektur komputer yang demikian maka kecepatan algoritma sangat ditentukan oleh jumlah prosesor yang dipakai serta pola hubungan interkoneksi antara prosesor yang satu dengan yang lain. Mengingat bahwa komputasi pada sistem multi prosesor akan lebih cepat dibanding dari sistem komputer biasa (yang disebabkan adanya tambahan prosesor), maka perlu didefinisikan suatu besaran yang merupakan ukuran peningkatan kecepatan yang sebenarnya. Besaran ini antara lain adalah efisiensi dan peningkatan kecepatan (*speed-up*) dari sistem multiprosesor, yang didefinisikan sebagai berikut:

$$E_{(efisisensi)} = T(1) / p\, T(p)$$

$$S_{(speed\text{-}up)} = T(1) / T(p)$$

dimana *T(1)* = waktu yang diperlukan untuk menyelesaikan masalah dengan *1* prosesor, *p* = banyak prosesor yang dipakai, dan *T(p)* = waktu yang diperlukan untuk menyelesaikan masalah dengan *p* prosesor.

Suatu sistem paralel dapat digambarkan sebagai teknik pemrosesan secara simultan pada subproses yang independen. Menurut Hwang dan Briggs, pemrosesan paralel didefinisikan sebagai bentuk pemrosesan yang efisien dengan menitikberatkan pada ekploitasi kejadian-kejadian yang bersamaan [8]. Tujuan utamanya adalah mereduksi waktu proses yang dibutuhkan atau untuk penyelesaian masalah yang sebelumnya dipandang terlalu besar [6]. Akhirnya

ISSN : 2302-3805

timbul pemikiran bagaimana membuat algoritma sistem paralel.

Untuk melihat proses-proses yang dapat dikerjakan secara simultan, langkah pertama adalah melakukan proses dekomposisi persoalan sehingga diperoleh bagian-bagian yang independen atau mencari letak paralelisme dari suatu permasalahan. Setelah melakukan dekomposisi terhadap permasalahan tersebut, maka diperoleh beberapa submasalah yang lebih sederhana dan dapat diselesaikan secara paralel.

Ada 2 macam cara melakukan dekomposisi masalah, yaitu : (i) dekomposisi secara algoritmis dan (ii) dekomposisi secara geometrik [2]. Dekomposisi secara algoritmis adalah dekomposisi algoritma sekuensial yang ada atas beberapa blok instruksi, dimana tiap blok instruksi akan dikerjakan oleh prosesor yang berbeda. Strategi ini biasanya cenderung melihat alur pemecahan masalah yang dihadapi. Sedangkan dekomposisi secara geometrik adalah mendekomposisi masalah yang ada menjadi beberapa submasalah dengan uturan tertentu dimana tiap submasalah bisa dipecahkan secara paralel dan independen. Strategi ini biasanya cenderung melihat struktur data dari persoalan yang dihadapi.

Untuk memecahkan permasalahan sistem tersebut, digunakan strategi dekomposisi secara geometrik, karena lebih banyak melihat pada masalah struktur data dari persoalan yang dihadapi, yaitu bagaimana mendekomposisi data atas beberapa kelompok data yang akan diproses oleh prosesor yang berbeda.

Selain itu untuk mendapatkan model algoritma paralel dapat ditempuh dengan cara memodifikasi algoritma sekuensial, sehingga akan diperoleh bentuk algoritma paralel. Sedangkan untuk memperoleh gambaran kinerjanya, algoritma paralel perlu dibandingkan dengan algoritma sekuensial.

4. Hasil Dan Analisis

Secara kualitatif komputasi pada metode *recursive decoupling* tidak cenderung *fine grain,* yaitu rasio komputasi dan komunikasi yang kecil. Berdasarkan hasil uji coba pada mesin paralel berbasis PVM, pengukuran waktu hasil eksekusi dan karakteristik kinerja algoritma recursive decoupling dapat disajikan sebagai berikut.

Tabel 1 Presentasi Pengukuran Waktu Eksekusi Algoritma *Recursive Decoupling*

NO	Ukuran Data	Waktu eksekusi			
		P=1	P=2	P=4	P=8
1	512	280	168	114	96
2	1024	712	379	249	191
3	2048	1817	938	575	421
4	4096	4639	2440	1421	989
5	8192	13886	6694	3766	2502
6	16384	33107	16554	9094	5821
7	32768	78281	39141	21180	13257

Tabel 2 Presentasi Perhitungan *Speed-up* Algoritma *Recursive Decoupling*

NO	Ukuran Data	*Speed-up* (Sp = T1/Tp)			
		P=1	P=2	P=4	P=8
1	512	1	1,61	2,44	2,90
2	1024	1	1,88	2,86	3,73
3	2048	1	1,94	3,16	4,32
4	4096	1	1,98	3,41	4,89
5	8192	1	2,00	3,55	5,35
6	16384	1	2,00	3,64	5,69
7	32768	1	2,00	3,70	5,90

Tabel 3 Presentasi Perhitungan Efisiensi Algoritma *Recursive decoupling*

NO	Ukuran Data	Efisiensi (Sp/p x 100%)			
		P=1	P=2	P=4	P=8
1	512	100	80,43	61,00	36,25
2	1024	100	94,40	71,50	46,63
3	2048	100	98,17	79,00	54,00
4	4096	100	99,64	85,25	61,13
5	8192	100	100	88,75	66,88
6	16384	100	100	91,00	71,13
7	32768	100	100	92,50	73,75

Dari tabel 1 dan tabel 2, terlihat bahwa terjadi kenaikan percepatan seiring dengan bertambahnya jumlah prosesor yang dipakai. Percepatan berkisar antara 1,61 (2 prosesor) sampai dengan 5,90 (8 prosesor).

Namun sebaliknya, dari tabel 2 dan tabel 3, terlihat bahwa dengan bertambahnya jumlah prosesor yang dipakai terjadi penurunan efisiensi. Tingkat efisiensi mencapai 80,43% (2 prosesor) dan terendah 36,25% (8 prosesor).

5. Kesimpulan

Berdasarkan uraian di atas dapat disimpulkan beberapa hal berikut:

(1) Dengan menggunakan sistem multiprosesor waktu penyelesaian sistem persamaan linier **Au = d** diatas dapat dipercepat.
(2) Pada metode *recursive decoupling* karena faktor granularitas, efek *fork* dan *join*, dan proses sinkronisasi menyebabkan waktu komputasinya menurun (banyaknya komputasi tiap level berikutnya menurun).
(3) Kinerja algoritama *recursive decoupling* mencapai percepatan tertinggi pada 8 prosesor (4,22) dan efisiensi terendahnya terjadi pada 8 prosesor (35,58%).

Daftar Pustaka

 ISSN : 2302-3805

[1] Akl, Selim G. 1989. *The Design and Analysis of Parallel Algorithms*, Prentice Hall International Inc.
[2] Askew, C.R., Carpenter, D.B., Chalker, J.T., Hey, A.J.G., Moore, M., Nicole, D.A, and Pritchard, D.J., 1988. *Monte Carlo Simulation on transputer arrays*. Parallel Computing 6, pp 247-258.
[3] Berstsekas and Tsitsiklis, 1989, *Parallel and Distributed Computation, Numerical Methods*, Prentice Hall New Jersey.
[4] Evans, DJ., 1990, *A Recursive Decoupling Method for Solving Tridiagonal Linier Systems*, International Journal Computer Mathematics.
[5] Evans, DJ., 1992, *Design of Parallel Numerical Algorithms*, Elsevier Science Publisher.
[6] Freman and Phillips, 1992, *Parallel Numerical Algorithms*, Prentice Hall, London
[7] Golub and Van Loan, 1989, *Matrix Computation*, Second Edition, The John Hopkins University Press
[8] Hwang, Kai and Briggs, FA., 1984. *Computer Architecture and Parallel Pocressing*. McGraw-Hill. Book Company
[9] Mitchell and Griffiths, 1989, *The Finite Difference Method in partial Differetial Equations*, John Wiley & Sons
[10] Tanembaum, 2002, *Structured Computer Organization*, Prentice Hall International Inc.
[11] Tri Prabawa, 2013, *Analisis Kinerja Algoritma Reduksi Siklis untuk Sistem Persamaan Linier dengan Matriks Tridiagonal berbasis PVM*. Proceeding Seminar Nasional Riset Teknologi Informasi STMIK Akakom Yogyakarta.

Biodata Penulis

Tri Prabawa, menyelesaikan studi S1 bidang Matematika, Universitas Gadjah Mada (1986) dan S2 bidang Ilmu Komputer, Universitas Indonesia (1993). Dosen Prodi Teknik Informatika STMIK Akakom Yogyakarta (1994 – sekarang)

ISSN : 2302-3805