

ANALISIS MATEMATIKA PERBANDINGAN METODE *DIVIDE & BROADCAST* DAN METODE *DIVIDE & PARTIAL BROADCAST*

Pasrun Adam¹ dan Edi Cahyono²
^{1,2} Jurusan Matematika FMIPA Universitas Haluoleo
Kampus Bumi Tridharma Anduonohu
KENDARI 93232
Email : edi_cahyono@innov-center.org

Abstract. In this article we discuss the comparison of Divide & Broadcast Method (DBM) and Divide & Partial Broadcast Method. Both are data processing methods. From mathematical point of view the methods are to seek a relation of two sets of a large amount of data, too large for one processor. The comparison is based on three basic computations of the matching of two sets i.e. multiplication, addition and logarithm. The comparison is to obtain the efficiency of the methods. Previous researches have shown that based on multiplication and addition DPBM is more efficient than DBM, but the result based on logarithm is still unknown. In this article we report our result on the comparison of DBM and DPBM based on logarithm computation.

Key words: Divide & (Partial) Broadcast Method, data processing, relation of two sets

1. PENDAHULUAN

Artikel ini merupakan sebagian dari hasil salah satu riset payung yang dikembangkan di Jurusan Matematika FMIPA Universitas Haluoleo dengan topik “Pengembangan Perangkat Lunak Berbasis Matematika.” Riset payung ini telah dimulai sejak tahun 2006 dan melibatkan kerja sama dengan Dr. David Taniar, *senior lecturer* di *Monash University* serta Dr. J. Wenny Rahayu, *Assoc. Professor* pada *La Trobe University*, keduanya di Australia. Pada saat ini fokus riset payung ini masih pada pengembangan metode pemrosesan data pada sistem basis data (*database*).

Sistem database yang berkemampuan tinggi secara normal dapat diimplementasikan dalam suatu multi-prosesor (DeWitt & Gray, 1992). Dalam sistem ini, keberadaan algoritma yang seimbang dalam penggabungan data merupakan hal yang penting untuk mencapai hasil yang optimal dalam pemrosesan data. Algoritma penggabungan ini dapat diproses dalam dua tahap; i) pemisahan (partisi) data, yaitu membagi data ke dalam beberapa prosesor, ii)

penggabungan; yakni suatu operasi penggabungan (*join operation*) ke dalam setiap prosesor.

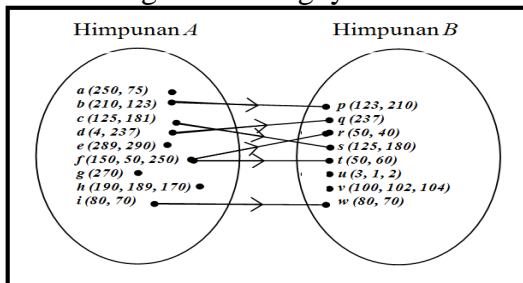
Metode yang umum digunakan dan telah dikenal lama adalah Metode *Divide & Broadcast* (MDB) (Leung & Ghogomu, 1993). Metode ini bekerja seperti berikut. Diberikan dua himpunan data A dan B dengan N prosesor. Pemrosesan data dilakukan dengan cara membagi hanya satu himpunan, katakan hanya B yang dibagi menjadi N subhimpunan. Setiap prosesor memproses himpunan A dan satu subhimpunan B . MDB dianggap tidak efisien karena himpunan A diproses pada setiap prosesor.

Untuk mengatasi hal ini, Taniar & Rahayu (1998) memperkenalkan Metode *Divide & Partial Broadcast* (MDPB) untuk mengurangi pemrosesan himpunan A pada setiap prosesor. Dalam MDPB kedua himpunan A dan B dibagi menjadi N subhimpunan, hanya sebuah prosesor yang memproses keseluruhan himpunan A . Analisis kuantitatif/matematis efisiensi kedua metode baru dibahas pada Cahyono dkk (2006) dan lebih detail pada Arman & Cahyono (2007). Artikel ini melaporkan

hasil riset tentang efisiensi kedua metode yang didasarkan pada perhitungan logaritmik yang belum dibahas pada ketiga literatur di atas. Perhitungan logaritmik sebagai dasar perhitungan kinerja pemrosesan data pada sebuah prosesor telah dibahas antara lain pada Matias & Vishkin (1991) serta Han (2001). Artikel ini menggunakan dasar perhitungan logaritmik untuk mempelajari efisiensi kinerja pemrosesan data pada lebih dari satu prosesor.

2. PERUMUSAN MASALAH

Pemrosesan data yang dibahas pada artikel ini, dari sudut pandang matematika adalah mencari relasi dari dua himpunan data, katakan A dan B . Misalkan banyaknya data pada himpunan A adalah r data, sedangkan pada B adalah s . Setiap data mempunyai sifat-sifat tertentu. Relasi didasarkan pada kesamaan sifat-sifat data tersebut. Gambar 1 mengilustrasikan himpunan A dan B serta relasi di antara keduanya, sifat-sifat data dituliskan dalam tanda kurung di belakangnya.

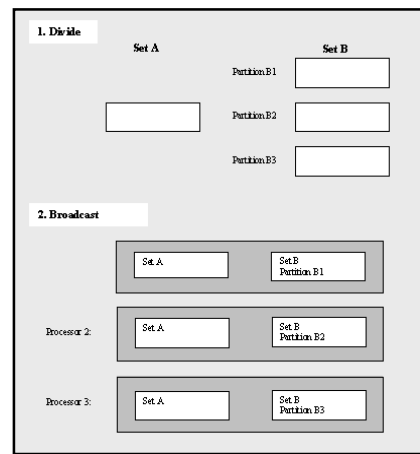


Gambar 1. Diagram Venn relasi antara himpunan A dan B , ilustrasi contoh pada Taniar & Rahayu (1998).

Mencari relasi pada contoh tersebut sangatlah mudah, bisa dengan pensil dan tangan. Namun demikian, bila banyaknya data pada kedua himpunan sangat besar bahkan terlalu besar untuk sebuah prosesor, maka diperlukan metode yang efisien. Misalkan digunakan N buah prosesor, efisiensi metode didasarkan pada dua hal, yaitu (1) *Elapsed time of the whole process*, waktu proses terlama yang diperlukan prosesor-prosesor yang terlibat, (2) *Total cost*, total waktu proses yang diperlukan oleh semua prosesor.

3. MDB dan MDPB

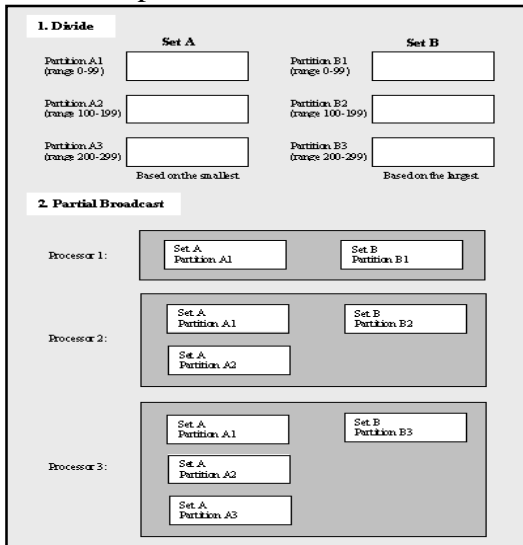
Untuk ilustrasi MDB dan MDPB berikut kita akan menggunakan tiga buah prosesor seperti contoh pada Taniar & Rahayu (1998). Pada MDB himpunan B dibagi menjadi 3 subhimpunan, katakan B_1, B_2 dan B_3 . Banyaknya data pada ketiga subhimpunan ini harus seimbang atau 'sama' tanpa memperhatikan sifat data agar metode bekerja efisien. Prosesor ke- n memproses himpunan A dan B_n , dengan $n = 1, 2, 3$. Hal ini diilustrasikan pada Gambar 2.



Gambar 2. Distribusi data pada MDB.

Berbeda dengan MDB, MDPB membagi kedua himpunan dengan memperhatikan sifat data. Himpunan A dibagi berdasarkan sifat minimum masing-masing data menjadi subhimpunan A_1, A_2, A_3 . Sebagai contoh sifat minimum data $a(250, 75)$ adalah 75, sedangkan $c(125, 181)$ adalah 125. Sedangkan himpunan B dibagi berdasarkan sifat maksimum masing-masing data menjadi subhimpunan B_1, B_2 dan B_3 . Sifat maksimum $p(123, 210)$ adalah 210, sedangkan $r(50, 40)$ adalah 50. Subhimpunan A_1 memuat data dengan sifat terkecil dari 0 sampai dengan 100, A_2 dari 101 sampai dengan 200 dan A_3 dari 201 sampai dengan 300. Sedangkan subhimpunan B_1 memuat data dengan sifat terbesar dari 0 sampai dengan 100, B_2 dari 101 sampai dengan 200, dan B_3 dari 201 sampai dengan 300.

Selanjutnya untuk tujuan operasi penggabungan data, subhimpunan A_1 dan B_1 ditempatkan pada prosesor 1. Prosesor 2 ditempati oleh subhimpunan A_1, A_2 , dan B_2 . Sedangkan prosesor 3 harus melakukan operasi penggabungan subhimpunan A_1, A_2, A_3 dan B_3 . Hal ini diilustrasikan pada Gambar 3. Perhatikan bahwa pembagian kedalam beberapa prosesor ini menjamin diperolehnya semua relasi antar data pada kedua himpunan A dan B .



Gambar 3. Distribusi data dalam prosesor pada MDPB.

4. PERHITUNGAN EFISIENSI

Efisiensi metode didasarkan pada dua hal, yaitu (1) *Elapsed time of the whole process* (ET), waktu proses terlama yang diperlukan prosesor-prosesor yang terlibat, (2) *Total cost* (TC), total waktu proses yang diperlukan oleh semua prosesor. Pada ilmu komputer waktu proses (WP) prosesor merupakan fungsi terhadap banyaknya data pada masing-masing himpunan yang akan diproses, dan terdapat tiga jenis perhitungan. Misalkan kita perhatikan prosesor n yang harus memproses subhimpunan A dengan $r^{(n)}$ data dan subhimpunan B dengan $s^{(n)}$ data. Ketiga jenis perhitungan tersebut adalah:

- Perkalian
WP prosesor
 $n : f(r^{(n)}, s^{(n)}) = r^{(n)} \cdot s^{(n)}$
- Penjumlahan

WP prosesor

$$n : f(r^{(n)}, s^{(n)}) = r^{(n)} + s^{(n)}$$

- Logaritmik

WP prosesor

$$n : f(r^{(n)}, s^{(n)}) = r^{(n)} \ln r^{(n)} + s^{(n)} \ln s^{(n)}$$

Misalkan digunakan N buah prosesor, perhatikan bahwa

$$ET = \max\{f(r^{(n)}, s^{(n)}) | n=1, 2, \dots, N\} \quad (1)$$

dan

$$TC = \sum_{n=1}^N f(r^{(n)}, s^{(n)}) \quad (2)$$

Menurut Arman & Cahyono (2007) perbandingan MDB dan MDPB didasarkan perhitungan perkalian dan penjumlahan adalah seperti yang disajikan pada Tabel 1 berikut:

Jenis Perhitungan	TC	ET
Perkalian	Lebih efisien dari DB	Lebih efisien dari DB
Penjumlahan	Tidak lebih efisien dari DB	Lebih efisien dari DB

Tabel 1. Perbandingan metode DB dengan DPB

5. EFISIENSI BERDASARKAN PERHITUNGAN LOGARITMIK

Misalkan terdapat N prosesor. Himpunan A dibagi menjadi N subhimpunan yaitu A_1, \dots, A_N berdasarkan sifat minimumnya. Sedangkan himpunan B dibagi menjadi N subhimpunan yaitu B_1, \dots, B_N berdasarkan sifat maksimumnya. Catat bahwa A_1, \dots, A_N saling asing, demikian juga B_1, \dots, B_N . Katakan banyaknya data pada subhimpunan A_1, \dots, A_N berturut-turut adalah r_1, \dots, r_N dan banyaknya data pada subhimpunan B_1, \dots, B_N berturut-turut adalah s_1, \dots, s_N . Kita tuliskan banyaknya data pada himpunan A

$$\sum_{n=1}^N r_n = r \quad (3)$$

banyaknya data pada himpunan B

$$\sum_{n=1}^N s_n = s \quad (4)$$

Pada MDB banyak data dari himpunan A dan dari himpunan B yang dimasukkan ke prosesor n berturut-turut adalah $r^{(n)} = r$ dan $s^{(n)} = s_n$. Sedangkan pada MDPB banyak data dari himpunan A dan dari himpunan B yang dimasukkan ke prosesor n berturut-turut adalah $r^{(n)} = \sum_{i=1}^n r_i$ dan $s^{(n)} = s_n$.

Pada MDB karena pembagian himpunan B tidak bergantung sift-sifat data, maka agar diperoleh $ET_{MDB} = \max \{r \ln r + s_n \ln s_n \mid n = 1, 2, \dots, N\}$ terkecil haruslah

$$s_n = \frac{s}{N} \text{ untuk } n = 1, \dots, N \quad (5)$$

Dengan demikian ET MDB terkecil dicapai untuk kasus (5) yaitu

$$ET_{MDB_0} = \frac{s}{N} \ln \frac{s}{N}. \quad (6)$$

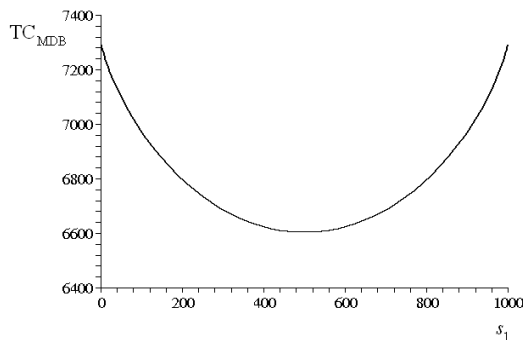
Di sisi lain TC MDB diberikan oleh

$$\begin{aligned} TC_{MDB} &= \sum_{n=1}^N (r \ln r + s_n \ln s_n) \\ &= Nr \ln r + \sum_{n=1}^N (s_n \ln s_n) \end{aligned} \quad (7)$$

Untuk kasus (5), maka (7) menjadi

$$TC_{MDB_0} = Nr \ln r + \sum_{n=1}^N \left(\frac{s}{N} \ln \frac{s}{N} \right) = Nr \ln r + s \ln \frac{s}{N}. \quad (8)$$

Gambar 4 menunjukkan TC MDB untuk $N = 2$ dengan $0 < s_1 < s = 1000$ dan $r = 50$. Perhatikan bahwa TC MDB terkecil juga dicapai pada kondisi (5), yaitu $s_1 = s_2 = 500$.



Gambar 4. TC_{MDB} untuk berbagai nilai s_1 dengan $s = 1000$.

Proposisi 5.1 TC_{MDB} (7) dengan kondisi (4) mencapai minimum apabila (5) dipenuhi. Nilai minimum TC_{MDB} ini diberikan oleh (8).

Sedangkan pada MDPB kita mempunyai ekspresi matematika berikut

$$ET_{MDPB} = \max \left\{ \left(\sum_{i=1}^n r_i \right) \ln \left(\sum_{i=1}^n r_i \right) + s_n \ln s_n \mid n = 1, 2, \dots, N \right\} \quad (9)$$

Dengan mengambil kondisi (5), maka diperoleh

$$\begin{aligned} \left(\sum_{i=1}^n r_i \right) \ln \left(\sum_{i=1}^n r_i \right) + s_n \ln s_n &= \left(\sum_{i=1}^n r_i \right) \ln \left(\sum_{i=1}^n r_i \right) + \frac{s}{N} \ln \frac{s}{N} \\ &< \left(\sum_{i=1}^{n+1} r_i \right) \ln \left(\sum_{i=1}^{n+1} r_i \right) + \frac{s}{N} \ln \frac{s}{N} \\ &= \left(\sum_{i=1}^{n+1} r_i \right) \ln \left(\sum_{i=1}^{n+1} r_i \right) + s_{n+1} \ln s_{n+1} \end{aligned}$$

untuk $n = 1, 2, \dots, N - 1$ (10)

Akibatnya, bila (5) dipenuhi

$$\begin{aligned} ET_{MDPB} &= \left(\sum_{i=1}^N r_i \right) \ln \left(\sum_{i=1}^N r_i \right) + \frac{s}{N} \ln \frac{s}{N} \quad (11) \\ &= r \ln r + \frac{s}{N} \ln \frac{s}{N} = ET_{MDB_0} \end{aligned}$$

Berdasarkan fakta (10) kita bisa memilih s_n sedemikian rupa sehingga diperoleh

$$\begin{aligned} \left(\sum_{i=1}^n r_i \right) \ln \left(\sum_{i=1}^n r_i \right) + s_n \ln s_n \\ = \left(\sum_{i=1}^{n+1} r_i \right) \ln \left(\sum_{i=1}^{n+1} r_i \right) + s_{n+1} \ln s_{n+1} \end{aligned}$$

untuk $n = 1, 2, \dots, N - 1$. (12)

Pemilihan ini akan berakibat tidak terpenuhinya (5), melainkan (12) mengakibatkan

$$s_n > s_{n+1}, \text{ untuk } n = 1, 2, \dots, N - 1 \quad (13)$$

Untuk kondisi (12) dan (13) dipenuhi, maka diperoleh

$$\begin{aligned} ET_{MDPB} &= r \ln r + s_N \ln s_N < r \ln r + \frac{s}{N} \ln \frac{s}{N} \quad (14) \\ &= ET_{MDB_0} \end{aligned}$$

Hal ini kita rangkum dalam Proposisi 5.2 berikut.

Proposisi 5.2 Bila pembagian himpunan A dilakukan sedemikian rupa sehingga (12) dipenuhi, maka $ET_{MDPB} < ET_{MDB_0}$

Proposisi 5.2 menyatakan bahwa ET MDPB lebih baik daripada ET MDB apabila kondisi (12) dipenuhi. Sedangkan untuk TC kedua metode kita simpulkan dalam Proposisi 5.3.

Proposisi 5.3. *Bila pembagian himpunan A dilakukan sedemikian rupa sehingga (5) dipenuhi, maka $TC_{MDPB} < TC_{MDB_0}$*

Bukti:

$$\begin{aligned} TC_{MDPB} &= \sum_{n=1}^N \left(\left(\sum_{i=1}^n r_i \right) \left(\ln \sum_{i=1}^n r_i \right) + s_n \ln s_n \right) \\ &= \sum_{n=1}^N \left(\left(\sum_{i=1}^n r_i \right) \left(\ln \sum_{i=1}^n r_i \right) + \frac{s}{N} \ln \frac{s}{N} \right) \\ &< \sum_{n=1}^N \left(r \ln r + \frac{s}{N} \ln \frac{s}{N} \right) \\ &= Nr \ln r + s \ln \frac{s}{N} \\ &= TC_{MDB_0} \end{aligned}$$

Proposisi 5.3 menyatakan bahwa TC MDPB lebih baik daripada TC MDB bila kondisi (5) dipenuhi.

6. KESIMPULAN

Kita telah membahas efisiensi Metode Divide & Broadcast (MDB) dan Metode Divide & Partial Broadcast (MDPB) untuk pemrosesan data yang dilakukan pada multi prosesor. Efisiensi didasarkan pada (1) *Elapsed time of the whole process* (ET), waktu proses (WP) terlama yang diperlukan prosesor-prosesor yang terlibat, (2) *Total cost* (TC), total waktu proses yang diperlukan oleh semua prosesor. ET dan TC didasarkan pada perhitungan logaritmik. Diperoleh bahwa ET MDPB lebih bagus (lebih kecil) daripada ET MDB bila pembagian himpunan B membuat WP tiap prosesor seragam atau sama. Sedangkan TC MDPB lebih bagus (lebih kecil) daripada TC MDB bila pembagian himpunan B seragam atau 'sama'.

7. UCAPAN TERIMA KASIH

Riset yang dilakukan oleh Pasrun Adam dibiayai oleh hibah Riset Fundamental DIPA Universitas Haluoleo tahun 2010.

8. DAFTAR PUSTAKA

- [1]. Arman & Cahyono, E., (2007), *Pemodelan Matematika dan Analisis Pemrosesan Data Divide & Broadcast dan Divide & Partial Broadcast*. Jurnal Ilmiah Mat Stat, Universitas Bina Nusantara, Vol. 7 No. 2 Juli 2007:139-154.
- [2]. Arman & Saidi, L., (2006), *Pemodelan Matematika dan Analisa Metode Pemrosesan Data Divide & Partial Broadcast*, Laporan Penelitian Dosen Muda/BBI 2006, Dibiayai oleh Bagian Proyek Peningkatan Penelitian Pendidikan Tinggi, Departemen Pendidikan Nasional, dengan kontrak No. 034/SP3/PP/DP2M/II/2006.
- [3]. Cahyono, E., Arman, Taniar, D., and Rahayu, J.W., (2006), *Divide and Partial Broadcast Method: a Note from Mathematical Point of View*, Proceedings of the International Confrence on Mathematics and Natural Sciences, ITB, Bandung, hal 726 – 730.
- [4]. DeWitt, D & Gray, J., (1992), *Parallel Database Systems: The Future of High Performance Database Systems*, Communication of the ACM, vol 35, no 6, pp. 85-98.
- [5]. Han, Y. (2001) *Improved fast integer sorting in linear space*, Information and Computation, Vol. 170, Issue 1, hal 81 – 94.
- [6]. Leung, C.H.C & Ghogomu, H.T., (1993), *A High Performance Parallel Database Architecture*, Proceedings of the 7th ACM International Confrence on Supercomputing, hal. 377-386.
- [7]. Matias, Y. & Vishkin, U., (1991), *On parallel hashing and integer sorting*, Journal of Algorithm, Vol. 12, Issue 4, hal 573 – 606.
- [8]. Taniar, D. & Rahayu, J. W, (1998), *Divide and Partial Broadcast Method*

Pasrun Adam, Edi Cahyono (Analisis Matematika Perbandingan Metode Devide dan Broadcast dan Metode

for Parallel Collection Join Queries,
High-Performance Computing and
Networking, Lecture Notes in
Computer Science, vol. 1401, Springer
verlag, pp 937-939.