

Analisis Metode Alokasi Fragmen Pada Sistem Basis Data Terdistribusi

Miftahul Jannah¹⁾ Baby Lolita²⁾ Hustinawati³⁾

^{1) 2)} Jurusan Sistem Informasi, Universitas Gunadarma

³⁾ Program Doktor Teknologi Informasi, Universitas Gunadarma

Jl. Margonda Raya 100, Pondok Cina, Depok, Jawa Barat

Email : ^{1) 2) 3)} {miftah, b_lolita, [hustina](mailto:hustina@staff.gunadarma.ac.id)}@staff.gunadarma.ac.id

ABSTRAK

Perkembangan teknologi basis data tidak lepas dari perkembangan teknologi jaringan komunikasi sehingga melahirkan sistem basis data terdistribusi (*Distributed Database Systems/DDBs*). Pada DDBs setiap situs dapat berpartisipasi dalam pengeksekusian transaksi-transaksi yang mengakses data pada satu situs atau beberapa situs. Hal utama yang menjadi permasalahan dalam DDBs adalah bagaimana meminimalkan biaya transfer data dan jumlah query selama pemrosesan aplikasi. Banyak peneliti yang mengembangkan metode alokasi fragmen untuk memecahkan masalah tersebut, antara lain metode clustering yang mengelompokkan situs ke cluster dan meminimalkan biaya komunikasi antar situs dan meningkatkan kinerja sistem, metode NNA menyelidiki efek dari parameter yang berbeda pada faktor delay rata-rata untuk menerima waktu respon fragmen dan rata-rata waktu yang dihabiskan untuk memindahkan data dari satu node ke fragmen data, metode heuristik yang meminimalkan ukuran data yang ditransfer antar node jaringan dan mengembangkan sebuah algoritma untuk menghasilkan rencana pengalihan fragmen ketika mengevaluasi sebuah query. Paper ini akan menganalisa tiga metode alokasi fragmen tersebut. Hasil analisa menunjukkan bahwa metode cluster lebih baik dari metode NNA dan metode heuristik dalam meminimalkan biaya transaksi komunikasi dengan mendistribusikan fragmen basis data ke DDBs, meningkatkan ketersediaan dan integritas data dengan mengalokasikan beberapa salinan basis data fragmen yang sama, dan meminimalkan total waktu respon transaksi.

Kata kunci : basis data terdistribusi, alokasi fragmen, clustering, heuristik, NNA

1. Pendahuluan

Perkembangan dalam basis data dan teknologi jaringan dalam dua dekade terakhir menyebabkan kemajuan dalam sistem basis data terdistribusi (*Distributed Database Systems/DDBs*). Sebuah DDBs adalah kumpulan situs yang dihubungkan oleh jaringan komunikasi, dimana tiap-tiap situs dapat berpartisipasi dalam pengeksekusian transaksi-transaksi yang mengakses data pada satu situs atau beberapa situs. Perhatian utama dari DDBs adalah merancang alokasi

fragmen untuk meningkatkan kinerja aplikasi pengolahan dalam DDBs. Alokasi fragmen adalah

proses menempatkan fragmen di situs DDBs untuk meminimalkan biaya transfer data dan jumlah pesan selama proses aplikasi. Beberapa metode telah dikembangkan untuk teknik alokasi fragmen dalam DDBs seperti metode *cluster* yang dikembangkan oleh Ismail O. Hababeh, Nicholas Bowring dan Muthu Ramachandran, Algoritma *Near Neighborhood Allocation (NNA)* oleh Reza Basseda, dan Algoritma *Heuristik* oleh Leon Tãmbulea, dan Manuela Horvat-Petrescu. Paper ini akan menganalisa tiga metode alokasi fragmen tersebut.

2. Metode Alokasi Fragmen

2.1. Metode Clustering

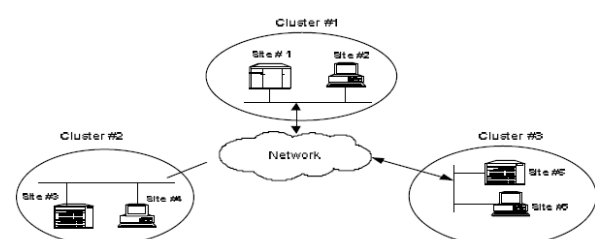
Metode alokasi fragmen dengan cara *clustering*/pengelompokan ini dikembangkan oleh Ismail O. Hababeh, Nicholas Bowring dan Muthu Ramachandran. Ismail dkk melakukan pengelompokan situs ke dalam *cluster* sesuai dengan biaya komunikasi, menentukan apakah suatu kumpulan data situs yang ditugaskan ke suatu *cluster* dianggap sebagai cara cepat untuk menentukan alokasi data untuk satu kumpulan situs, bukan dari situs ke situs.

Tabel 1. Membuat *cluster* situs

Cluster#	Jumlah Situs					
	S1	S2	S3	S4	S5	S6
C1	1	1	0	0	0	0
C2	0	0	1	1	0	0
C3	0	0	0	0	1	1

Tabel 2. Biaya komunikasi antara *cluster* DDBs

Cluster#	C1	C2	C3
C1	1	7.5	9.25
C2	7.5	1	8.25
C3	9.25	8.25	0.5



Gambar 1. Clustering enam situs melalui jaringan

Tabel 1 menunjukkan pengelompokan situs dalam *cluster*, untuk situs S_i dan S_j yang berada di *cluster* yang sama diberikan angka 1 dan untuk situs S_i dan S_j yang dikelompokkan di *cluster* yang berbeda diberikan angka 0. Tabel 2 menunjukkan dua situs (S_i, S_j) dikelompokkan dalam satu kelompok jika biaya komunikasi antar situs tersebut kurang dari atau sama dengan range biaya komunikasi (*Communication Cost Range/CCR*). Jumlah unit komunikasi yang diperbolehkan untuk perbedaan maksimum biaya komunikasi antara situs yang akan dikelompokkan dalam satu *cluster* ditentukan oleh jaringan DDBs. Pada gambar 1 memperlihatkan distribusi dari 6 situs melalui jaringan setelah situs dikempokkan berdasarkan biaya komunikasi [1].

Berdasarkan alokasi data dan biaya komunikasi dapat diketahui secara tepat apakah sebuah fragmen akan dialokasikan atau dihilangkan dari *cluster*.

Fragmen yang dialokasikan untuk semua *cluster* memiliki aplikasi yang menggunakan fragmen, dan nilai keputusan (D) alokasi fragmen untuk *cluster* dihitung sebagai nilai logis perbedaan antara biaya yang tidak mengalokasikan fragmen ke *cluster* dan biaya alokasi fragmen ke *cluster*. Jika biaya yang tidak mengalokasikan fragmen ke *cluster* (fragmen ditangani secara jarak jauh) lebih besar dari atau sama dengan biaya alokasi fragmen ke *cluster* maka nilai keputusan adalah benar dan fragmen dialokasikan ke *cluster*, dan jika biaya yang tidak mengalokasikan fragmen ke *cluster* kurang dari biaya alokasi fragmen ke *cluster* maka nilai keputusan adalah salah dan fragmen dibatalkan dari *cluster*.

Biaya alokasi fragmen F_i ke *cluster* C_j dihitung sebagai berikut :

- Biaya rata-rata pengambilan lokal di *cluster* C_j kali jumlah rata-rata frekuensi pengambilan dihasilkan dari transaksi T_k ke fragmen F_i di *cluster* C_j .
- Biaya rata-rata pembaruan lokal di *cluster* C_j kali jumlah rata-rata frekuensi pembaruan dihasilkan dari transaksi T_k ke fragmen F_i di *cluster* C_j .
- Biaya ruang yang ditempati oleh fragmen F_i di *cluster* C_j kali ukuran fragmen F_i (dalam byte).
- Remote pembaruan dikirim dari *cluster* lain C_x , biaya rata-rata lokal pembaruan di *cluster* C_j kali jumlah rata-rata frekuensi pembaruan dihasilkan transaksi T_k ke fragmen F_i untuk setiap *cluster* lain yang ada.
- Remote komunikasi dari *cluster* lain C_x , rasio pembaruan (Unit Pembaruan /Unit Komunikasi) kali jumlah rata-rata frekuensi pembaruan dihasilkan transaksi T_k ke fragmen F_i di *cluster* C_j kali biaya rata-rata komunikasi antara *cluster* lain dari yang ada .

Biaya tidak mengalokasikan fragmen F_i ke *Cluster* C_j dihitung sebagai berikut:

- Biaya rata-rata lokal pengambilan lokal di *cluster* C_j kali jumlah rata-rata frekuensi pengambilan dihasilkan oleh transaksi T_k ke fragmen F_i di *cluster* C_j .
- pengambilan jarak jauh dari *cluster* lain C_x , rasio pengambilan (unit pengambilan / unit komunikasi) kali jumlah rata-rata frekuensi pengambilan yang dihasilkan dari transaksi T_k ke fragmen F_i di *cluster* C_j untuk setiap *cluster* kali biaya rata-rata komunikasi antara *cluster*.

Nilai keputusan mengalokasikan fragmen F_i ke *cluster* C_i merupakan sebuah nilai yang logis dan dihitung sebagai berikut :

$$D(T_k, F_i, C_j) = (CN(T_k, F_i, C_j) \geq CA(T_k, F_i, C_j)) \dots \dots \dots (1)$$

Tabel 3. Fragmen dan frekuensi pengambilan dan pembaruan dalam *cluster* dari masing-masing situs

Fragmen #	Cluster #	Situs #	Frekuensi pengambilan	Frekuensi pembaruan	
F1	C1	S1	80	10	
		S2	60	26	
	C2	S3	60	16	
		S4	0	0	
	C3	S5	35	5	
		S6	25	5	
F2	C2	S3	20	4	
		S4	20	6	
	C3	S5	5	30	
		S6	10	20	
	F3	C1	S1	0	20
			S2	0	10
C2		S3	30	0	
		S4	0	0	
C3		S5	40	30	
		S6	30	10	
F4	C1	S1	10	20	
		S2	10	20	
	C2	S3	65	12	
		S4	5	12	
F5	C1	S1	70	20	
		S2	6	10	
	C2	S3	20	10	
		S4	20	10	
	C3	S5	35	10	
		S6	45	20	
F6	C1	S1	0	10	
		S2	0	0	
	C3	S5	25	5	
		S6	5	5	
F7	C2	S3	25	5	
		S4	35	10	
	C3	S5	10	0	
		S6	30	0	
F8	C1	S1	10	20	
		S2	80	20	
	C2	S3	20	0	
		S4	60	10	
	C4	S5	0	20	
		S6	20	0	

Tabel 4. Biaya ruang, pengambilan, dan pembaruan

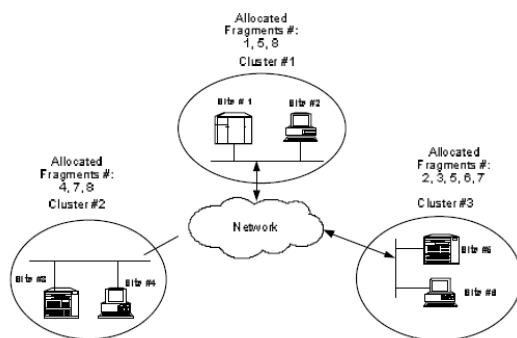
Cluster	situs	Biaya	Biaya	Biaya
---------	-------	-------	-------	-------

#	#	ruang	pengambilan	pembaruan
C1	S1	0.004	0.15	0.25
	S2	0.006	0.25	0.35
C2	S3	0.005	0.15	0.25
	S4	0.007	0.17	0.27
C3	S5	0.003	0.13	0.23
	S6	0.005	0.15	0.25

Tabel 5. Alokasi dan membatalkan alokasi fragmen di semua fragmen cluster

Fragmen #	Cluster #	Biaya alokasi	Biaya tidak alokasi	Nilai keputusan	Status alokasi
F1	C1	59.45	177.24	1	Alokasi
	C2	74.83	74.76	0	Batal
	C3	85.5	74.16	0	Batal
F2	C2	74.26	49.84	0	Batal
	C3	30.01	135.96	1	Alokasi
F3	C1	60.32	0	0	Batal
	C2	103.23	37.38	0	Batal
	C3	54.72	86.52	1	Alokasi
F4	C1	47.13	25.32	0	Batal
	C2	68.73	87.22	1	Alokasi
F5	C1	86.56	96.21	1	Alokasi
	C2	92.66	49.84	0	Batal
	C3	86.80	98.88	1	Alokasi
F6	C1	15.46	0	0	Batal
	C2	18.31	37.08	1	Alokasi
F7	C2	7.41	74.76	1	Alokasi
	C3	34.71	37.08	1	Alokasi
F8	C1	59.22	113.94	1	Alokasi
	C2	95.63	99.68	1	Alokasi
	C3	80.12	24.72	0	Batal

Tabel 3 menunjukkan fragmen dan jumlah frekuensi pengambilan dan pembaruan dari setiap cluster dan situs masing-masing. Biaya ruang, pengambilan, dan update ditunjukkan pada tabel 4. Tabel 5 menunjukkan alokasi dan membatalkan alokasi fragmen berdasarkan biaya alokasi fragmen, biaya fragmen yang tidak dialokasikan dan nilai keputusan.



Gambar 2. Alokasi Fragmen di cluster

Alokasi fragmen di cluster pada gambar 2 terlihat fragmen 1,5, dan 8 dialokasikan ke cluster #1. Fragmen 4,7, dan 8 dialokasikan ke cluster #2. Fragmen 2,3,5,6, dan 7 dialokasikan ke cluster #3.

2.2. Algoritma NNA

Algoritma NNA (*Near Neighborhood Allocation*) oleh Reza Basseda pada dasarnya merupakan variasi dari algoritma optimal. Pada algoritma ini, semua fragmen pada awalnya didistribusikan selama node sesuai dengan metode statis, setiap node j menjalankan algoritma yang optimal untuk setiap fragmen seperti berikut ini :

- Untuk setiap fragmen lokal yang tersimpan, menginisialisasi untuk mengakses baris counter ke nol. ($S_{ij} = 0$ adalah indeks fragmen ϵ_i dan ϵ_i node)
- Proses permintaan akses untuk fragmen disimpan.
- Meningkatkan akses sesuai counter dari node untuk mengakses fragmen yang disimpan. (Kalau node (x) , akses fragmen i , set enam = enam +1)
- Jika node mengakses pemilik saat ini, lanjutkan ke langkah 2. (akses lokal adalah akses remote)
- Jika counter node jauh lebih besar daripada counter dari pemilik node saat ini, transfer kepemilikan fragmen bersama-sama dengan akses array counter ke remote node. (bermigrasi fragmen jika node x mengakses fragmen i dan enam > S_{ij} , mengirim fragmen i ke node (x))
- Ulangi dari langkah 2.

Masalah dari algoritma ini adalah bahwa jika frekuensi perubahan pola akses untuk setiap fragmen tinggi maka akan menghabiskan banyak waktu untuk mentransfer fragmen ke node yang berbeda. Jadi, waktu respon dan penundaan akan ditingkatkan. Pada algoritma NNA, persyaratan untuk memindahkan sebuah fragmen yang diperoleh seperti pada algoritma optimal. Namun, tujuan data bergerak berbeda. Algoritma NNA mempertimbangkan topologi jaringan dan routing untuk menentukan tujuan. Dengan menggunakan Algoritma ini untuk menghindari terlalu sering memindahkan data karena fragmen akhirnya akan ditempatkan di suatu node yang memiliki rata-rata biaya akses node yang menggunakannya. Jadi, penundaan gerakan akan berkurang dan waktu respon juga akan ditingkatkan. [2]

2.3. Algoritma Heuristik

Algoritma *heuristik* dikembangkan oleh Leon Tãmbulea, Manuela Horvat-Petrescu. Algoritma *heuristik* meminimalkan ukuran data yang ditransfer antar node jaringan dan mengembangkan sebuah algoritma untuk menghasilkan rencana pentransferan fragmen saat mengevaluasi sebuah *query*. Rencana pentransferan ini diperoleh dengan menggunakan rencana evaluasi query dan distribusi fragmen di dalam node jaringan. [3]

Dua strategi evaluasi yang dapat dipertimbangkan adalah sebagai berikut:

- Evaluasi dilakukan pada node S , maka setiap fragmen yang diperlukan dipindahkan dari

node dimana yang disimpan ke dalam node S .
 Ukuran seluruh transfer adalah:

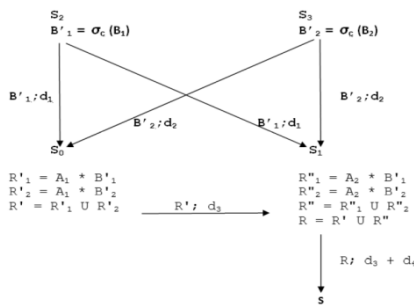
$$\dim(A_1) + \dim(A_2) + \dim(B_1) + \dim(B_2) \dots \dots (2)$$

2. Evaluasi dilakukan secara terdistribusi , untuk kasus ini q permintaan dapat diubah sebagai berikut:

$$\begin{aligned} q &= A * \sigma_c(B) = (A_1 \cup A_2) * \sigma_c(B_1 \cup B_2) = \\ &= [A_1 * \sigma_c(B_1 \cup B_2)] \cup [A_2 * \sigma_c(B_1 \cup B_2)] = \\ &= [A_1 * (\sigma_c(B_1) \cup \sigma_c(B_2))] \cup [A_2 * (\sigma_c(B_1) \cup \sigma_c(B_2))] \dots \dots (3) \end{aligned}$$

Permintaan q dapat dievaluasi dengan menggunakan grafik pada gambar 3, dimana mempertimbangkan bahwa hasil evaluasi $\sigma_c(B_i)$ akan memiliki ukuran / dimensi yang lebih kecil dari $\dim(B_i)$, $i = 1, 2$.

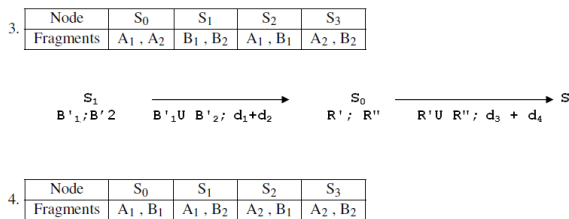
Perhatikan dengan $B'_i = \sigma_c(B_i)$, $i = 1, 2$ hasil seleksi dari S_2 dan S_3 dan node d_3 $\dim(B'_1)$, $d_4 \dim(B'_2)$.



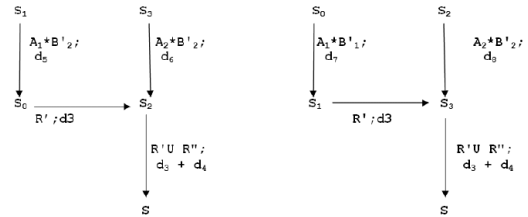
Gambar 3. Node grafik mewakili simpul jaringan

Dalam node dari grafik pada gambar 3 (sebuah node dari grafik mewakili simpul jaringan) disajikan operator (unary dan biner) yang dievaluasi dan pada link yang ditunjukkan transfer data yang dibutuhkan dalam proses evaluasi. Total biaya transfer akan menjadi $2(d_1 + d_2 + d_3) + d_4$.

Dalam sebuah basis data terdistribusi fragmen dapat direplikasi fragmen (fragmen dapat disimpan di lebih dari satu node).



Gambar 4. Rencana distribusi dua fragmen yang sesuai



Gambar 5. Node dari grafik yang mewakili simpul jaringan

Gambar 4 dan gambar 5 menunjukkan rencana distribusi fragmen menggunakan node dari grafik yang mewakili simpul jaringan.

Algoritma Redistribusi Fragmen berisi empat fase:

Fase pertama :

Untuk setiap node S_i , $0 \leq i < m$ initializations berikut akan dilakukan:

- (a) $s_i = 0$; (ukuran / dimensi yang digunakan oleh node S_i)
- (b) $L_i = F$; (set fragmen disimpan dalam node S_i)
- (c) Untuk setiap fragmen F_j , $0 \leq j < n$: $r_j = 0$; (jumlah replika fragmen F_j)

Fase kedua :

Nilai-nilai dari matriks V yang akan diambil dalam urutan menurun. A tupel $t = (F_i, F_j, v_{i,j})$ yang berhubungan dengan nilai $V_{i,j}$ dari matriks V merupakan keuntungan yang diperoleh untuk total transfer ukuran / dimensi jika fragmen F_i dan F_j disimpan di node yang sama. Untuk ini tupel t menentukan node menawarkan keuntungan terbaik dalam kasus ini bahwa hal tersebut fragmen akan ditambahkan di sana. Untuk menemukan node ini, algoritma akan menghitung k keuntungan c_k diperoleh jika fragmen yang melekat pada node S_k , dan akan terus melacak dari indikator. Nilai dari indikator adalah kepada bagian fragmen (F_i, F_j) relatif L_k , untuk $0 \leq k < m$.

Fase ketiga :

Jika fragmen F_i tidak digunakan dalam operator biner maka diperoleh dari Q pada periode waktu T digunakan untuk menganalisis dan mengevaluasi permintaan dari Q , maka hasil pada langkah akan menjadi $r_i = 0$, dan itu berarti bahwa fragmen F_i tidak disimpan dalam node jaringan. Dalam hal kasus sebuah node dari jaringan dengan cukup ruang untuk menyimpan fragmen F_i tidak menghapus fragmen lain akan dipilih.

Fase keempat :

R_i mengatur node mana yang berisi fragmen F_i disimpan dihitung untuk setiap fragmen F_i . (R_i adalah himpunan dari replika F_i) Pada akhir algoritma adalah menentukan nilai untuk setiap node L_k for each S_k , $0 \leq k < n$. Virtual node dapat dialokasikan node yang menggunakan informasi yang diperoleh dari sekumpulan permintaan Q . Node virtual dapat

dialokasikan ke node yang sesuai dengan node evaluasi yang diperlukan.

Algoritma untuk menghasilkan rencana $P(op)$ transfer fragmen ketika mengevaluasi query :

a. Jika "op" adalah operator unary, setelah form $op(x)$ maka:

- Jika x adalah fragmen disimpan dalam basis data terdistribusi lalu $P(op) = f\{(S, op, s); o | S \in R(X)\}$, dimana $R(X)$ merupakan sekumpulan node dimana fragmen x disimpan.
- Jika x adalah fragmen diperoleh dari evaluasi sebelumnya operator op' maka:

$$P(op) = \{(plan(p), (result(p), op, result(p)); dim(p)) | p \in P(op)\} \dots (4)$$

b. Jika "op" sebuah operator binary memiliki bentuk $op(x,y)$ maka:

- Jika x dan y adalah fragmen disimpan dalam basis data terdistribusi lalu:

$$P(op) = \{(S_1, op, S_2); 1 \mid \text{where } S_1 \in R(X) - R(Y), S_2 \in R(Y)\} \cup \{(S_1, op, S_2); 1 \mid \text{where } S_1 \in R(Y) - R(X), S_2 \in R(X)\} \cup \{(S_1, op, S_2); 0 \mid \text{where } S_1 \in R(X) \cap R(Y)\} \dots (5)$$

Evaluasi dari operator "op" dapat dibuat dalam node dimana fragmen x disimpan dan fragmen y dialihkan, atau dalam suatu node dimana fragmen y disimpan dan fragmen S dialihkan.

- Jika x adalah fragmen yang disimpan didalam basis data terdistribusi dan y adalah fragmen yang berisi evaluasi sebelumnya dari operator "op" maka :

$$P(op) = P_1(op) \cup P_2(op) \cup P_3(op) \text{ where: } P_1(op) = \{(plan(p), (result(p), op, S); dim(p)+1) \mid \text{where } p \in P(op'), S \in R(X) - \{result(p)\}\} \dots (6)$$

$$P_2(op) = \{(plan(p), (S, op, result(p)); dim(p)+1) \mid \text{where } p \in P(op'), S \in R(X) - \{result(p)\}\} \dots (7)$$

$$P_3(op) = \{(plan(p), (result(p), op, result(p)); dim(p)) \mid \text{where } p \in P(op'), \text{if } result(p) \in R(X)\} \dots (8)$$

Evaluasi operator "op" dapat dibuat dalam sebuah node dimana fragmen x disimpan dengan mengalihkan hasil dari node, dimana "op" dapat dievaluasi. di node dimana op' dievaluasi dengan mentransfer fragmen x. node dimana op' dapat dievaluasi dan dimana fragmen x disimpan.

- Jika x adalah fragmen yang diperoleh dari evaluasi sebelumnya dari operator op' dan y adalah fragmen yang disimpan dalam basis data terdistribusi, rencananya adalah membangun seperti pada b_2 .
- Jika x dan y adalah hasil evaluasi operator op_1 dan op_2 maka:

$$P(op) = P_1(op) \cup P_2(op) \cup P_3(op) \text{ where: } P_1(op) = \{(plan(p_1), plan(p_2), (result(p_1), op, result(p_2)); dim(p_1) + dim(p_2) + 1) \mid \text{where } p_1 \in P(op_1), p_2 \in P(op_2), result(p_1) \neq result(p_2)\} \dots (9)$$

$$P_2(op) = \cup\{(plan(p_1), plan(p_2), (result(p_2), op, result(p_1)); dim(p_1) + dim(p_2) + 1) \mid \text{where } p_1 \in P(op_1), p_2 \in P(op_2), result(p_1) \neq result(p_2)\} \dots (10)$$

$$P_3(op) = \cup\{(plan(p_1), plan(p_2), (result(p_1), op, result(p_1)); dim(p_1) + dim(p_2)) \mid \text{where } p_1 \in P(op_1), p_2 \in P(op_2), result(p_1) = result(p_2)\} \dots (11)$$

Setelah rencana itu dihasilkan dalam kasus a_2, b_2, b_3, b_4 ada kesempatan operator dua rencana yang berbeda $p_1 \neq p_2$ namun dengan node evaluasi akhir yang sama $final(p_1) = final(p_2)$. Menggunakan transformasi ini operator akan memiliki maksimum satu rencana untuk node terakhir.

Dalam hal ini dipilih rencana dengan dimensi yang lebih kecil. Menggunakan transformasi ini operator akan memiliki maksimum satu rencana untuk node terakhir.

3. Hasil Analisis

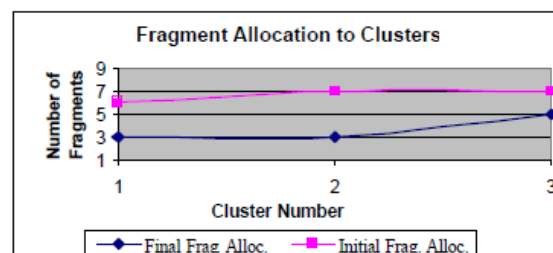
Pada metode *cluster* yang dilakukan oleh Ismail O. Hababeh dkk, pengelompokan situs ke dalam cluster meminimalkan biaya komunikasi antara situs dan meningkatkan kinerja sistem. Biaya rata-rata komunikasi antar cluster dan situs, dan rata-rata jumlah pengambilan dan pembaruan dipertimbangkan dalam perhitungan alokasi fragmen, karena waktu yang diperlukan untuk proses perhitungan rata-rata kurang dari waktu pemrosesan. Peningkatan kinerja sistem dapat ditingkatkan dengan menghilangkan fragmen berlebihan pada *cluster* basis data dan meningkatkan ketersediaan dan kehandalan data. Tabel 6 menunjukkan kinerja yang mengalokasikan fragmen ke DDBs *cluster* sebelum dan setelah menggunakan metode yang diusulkan.

Tabel 6. Kinerja evaluasi fragmen alokasi

Cluster#	Inisial alokasi fragmen	Final alokasi fragmen	Perbaikan %
C1	6	3	50 %
C2	7	3	57.14 %
C3	7	3	28.57 %

Sebelum menerapkan metode *cluster*, mengalokasikan fragmen untuk semua kelompok memiliki aplikasi yang meminta fragmen dan menghasilkan 20 alokasi, setelah menerapkan metode *cluster* diperoleh 11 alokasi, sehingga metode *cluster* memperbaiki kinerja sistem sebesar 45,00%.

Gambar 6 menunjukkan peningkatan kinerja sistem yang dicapai oleh pengalokasian dan pengelompokan pada metode *cluster*.



Gambar 6. Fragment alokasi ke *cluster*

Evaluasi dalam algoritma NNA oleh Reza Basseda mempertimbangkan dua faktor yaitu: delay rata-rata untuk menerima respon (waktu respon) untuk fragmen dan rata-rata waktu yang dihabiskan untuk memindahkan data dari satu node ke fragmen data. Algoritma ini akan menyelidiki efek dari parameter yang berbeda pada kedua faktor tersebut. Temuan dari eksperimen menunjukkan bahwa algoritma NNA melakukan yang lebih baik untuk ukuran fragmen query dan tingkat produksi yang lebih besar. Ambang batas untuk ukuran fragmen hampir 8.000 byte. Untuk jaringan yang lebih besar, dengan menggunakan algoritma NNA dapat mengurangi keterlambatan tanggapan atas fragmen.

Algoritma *heuristik* yang dilakukan oleh Leon, dkk menggunakan rencana permintaan eksekusi dan beberapa informasi yang dapat dihasilkan saat evaluasi relasi operator dari rencana eksekusi. Algoritma *heuristik* meminimalkan ukuran pengalihan data antar node jaringan saat query dievaluasi.

Untuk mengevaluasi permintaan beberapa fragmen yang diperlukan, fragmen disimpan di node dari distribusi basis data. Beberapa perluasan hasil percobaan menunjukkan bahwa dalam transfer data bukanlah mempertimbangkan ukuran transfer data melainkan biaya transfer. Pada algoritma redistribusi digunakan ukuran maksimum/dimensi untuk setiap node yang unik, bukan maksimum ukuran untuk semua node. Dalam algoritma menemukan rencana transfer antara node (algoritma digunakan untuk mengevaluasi permintaan) biaya transfer, bukan jumlah pengalihan fragmen.

4. Kesimpulan

Algoritma NNA merupakan variasi yang sederhana dari algoritma optimal yang dapat digunakan dalam desain sederhana sistem basis data terdistribusi. Metode *heuristik* menggunakan evaluasi determinan operator relasional dari eksekusi query yang meminimalkan ukuran data yang ditransfer antara node jaringan saat *query* dievaluasi tetapi membutuhkan perhitungan yang lebih kompleks untuk menangkap lebih banyak data. Oleh karena itu, dapat digunakan dalam sistem jaringan yang tidak kompleks. Metode *cluster* dirancang untuk memenuhi persyaratan dari situs *cluster* dan menentukan alokasi fragmen sistem basis data terdistribusi, meminimalkan komunikasi biaya antara situs, dan meningkatkan kinerja dalam lingkungan sistem jaringan yang heterogen. Metode *cluster* dikembangkan untuk pengelompokan situs ke *cluster*, yang membantu dalam mengurangi biaya komunikasi antara situs selama proses alokasi fragmen dan meningkatkan ketersediaan dan kehandalan alokasi beberapa salinan fragmen yang sama pada situs. Metode *cluster* ini dapat diimplementasikan dalam lingkungan jaringan yang heterogen dan bahkan dengan parameter input yang sangat besar. Dalam melakukan analisa, yang menjadi bahan pertimbangan

adalah biaya komunikasi antara situs, alokasi fragmen yang optimal dan meminimalkan ukuran data yang ditransfer antara node jaringan saat *query* dievaluasi pada sistem basis data terdistribusi. Untuk lingkungan jaringan yang kecil, biaya komunikasi di Metode *cluster* dan metode heuristik lebih besar dari NNA, tetapi untuk lingkungan jaringan yang lebih besar Metode *cluster* dan *heuristik* bertindak lebih baik dari NNA. Alasannya adalah bahwa untuk lingkungan jaringan kecil biaya memindahkan data ke node tujuan rendah dan biaya gerakan tidak melebihi biaya akses. Dalam kasus lingkungan jaringan besar gerakan fragmen membutuhkan lebih banyak waktu dan juga meningkatkan lalu lintas jaringan. Tetapi untuk jaringan yang heterogen dan bahkan dengan parameter input yang sangat besar maka metode *cluster* lebih baik dari NNA dan *heuristik*. Alasannya adalah untuk perhitungan yang lebih kompleks dan untuk menangkap lebih banyak data jaringan yang heterogen dan parameter input yang sangat besar membutuhkan evaluasi determinan operator relasional dari eksekusi *query* yang lebih kompleks. Aspek lain dari algoritma NNA adalah bahwa fragmen yang digunakan oleh sebuah node atau simpul tetangga dapat dikelompokkan menggunakan pendekatan *cluster*, sehingga dapat menanggapi permintaan data yang lebih efektif. Dengan demikian dapat disimpulkan bahwa metode *cluster* lebih baik dari metode NNA dan *heuristik* dalam meminimalkan biaya transaksi komunikasi dengan mendistribusikan fragmen basis data ke DDBs situs, meningkatkan ketersediaan dan integritas data dengan mengalokasikan beberapa salinan basis data fragmen sama pada situs, dan meminimalkan total waktu respon transaksi.

Referensi

- [1] Ismail O. Hababeh, UGRU, Nicholas Bowring, Muthu Ramachandran, 2005, *A Method for Fragment Allocation Design in the Distributed Database Systems*, Faculty of Information and Engineering Systems - School of Computing Leeds Metropolitan University Leeds - LS6 3QS, U.K
- [2] Reza Basseda, 2006, *Fragment Allocation in Distributed Database Systems*, Faculty of Electrical and Computer Eng., School of Engineering, University of Tehran, Database Research Group
- [3] Leon Tâmbulea, Manuela Horvat-Petrescu, 2008, *Redistributing Fragments into a Distributed Database*, International Journal of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. III, No. 4, pp. 384-394