[If3230]

Laporan Eksplorasi OpenMPI

oleh :

Daniar Heri Kurniawan / 13512064

Jacquiline Ibrahim / 13512074

Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2015

Pembahasan

a. Algoritma perkalian matriks paralel
Algoritma yang digunakan adalah algoritma pembagian per baris. Sehimgga tiap proses minimal akan mendapat satu baris. Setelah pembagian baris pada tiap proses, proses master akan mengumpulkan hasil dari proses slave nya dalam satu variabel.

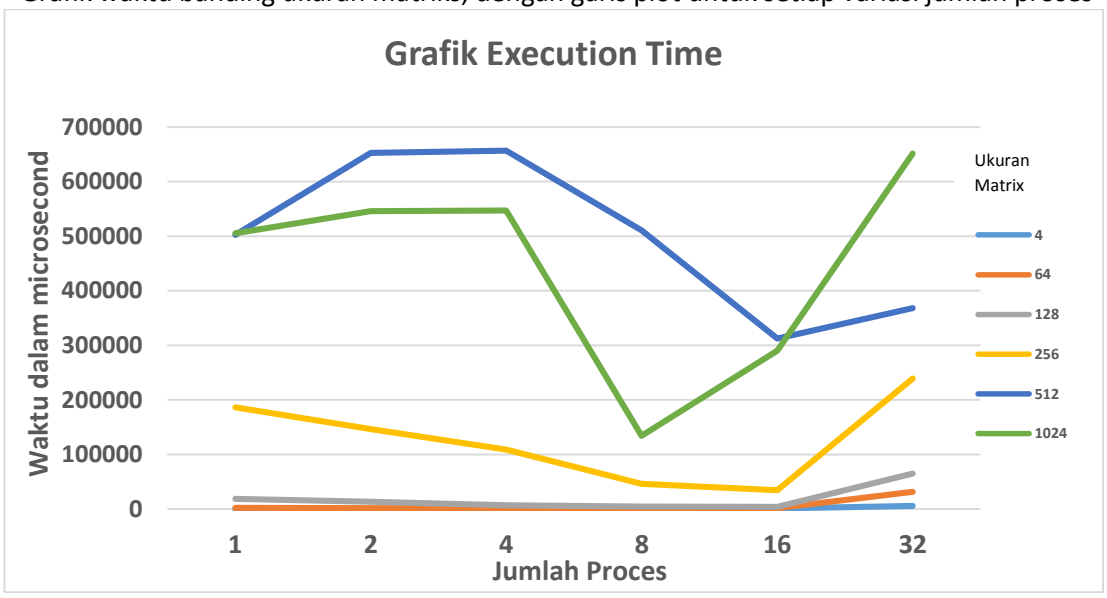
```
/*Pembagian baris untuk tiap proses*/
from = taskid * N / numtasks;
to = (taskid + 1) * N / numtasks;

MPI_Barrier(MPI_COMM_WORLD);

MPI_Bcast(B, N * N, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Scatter (A, N * N / numtasks, MPI_INT, A[from], N * N / numtasks, MPI_INT, 0,
MPI_COMM_WORLD);

/*Perhitungan pada masing-masing proses*/
for (i = from; i < to; i++){
        for (j = 0; j < N; j++) {
                C[i][j] = 0;
                for (k = 0; k < N; k++){
                        C[i][j] += A[i][k] * B[k][j];
                }
        }
}

/*Proses master mengumpulkan dari para slavenya*/
MPI_Gather(C[from], N * N / numtasks, MPI_INT, C, N * N / numtasks, MPI_INT, 0,
MPI_COMM_WORLD);

/*Sebelum terkumpul semua, proses tidak bisa melewati blok ini*/
MPI_Barrier(MPI_COMM_WORLD);
```

b. Matriks yang digunakan pada percobaan 3a, hasil perkaliannya dan apakah program benar atau tidak.
Program dapat menghitung matrix dengan benar.

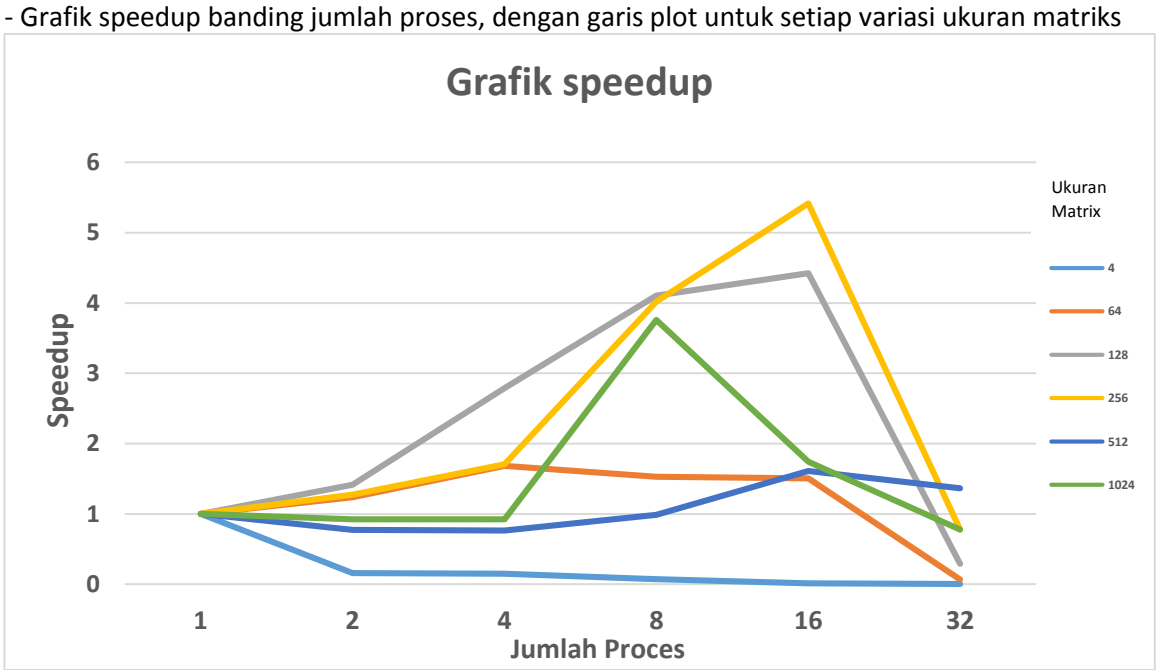
Matrix A :		Matrix B:		Hasil :
5 -4 -7 -5 7 9 3 -1 -6 4 -8 -5 0 1 9 -6	x	9 7 3 -3 0 2 9 3 6 -4 1 8 2 -4 -8 -3	=	-7 75 12 -68 79 59 113 33 -112 18 50 -19 42 -10 66 93

c. Grafik hasil percobaan 3b:

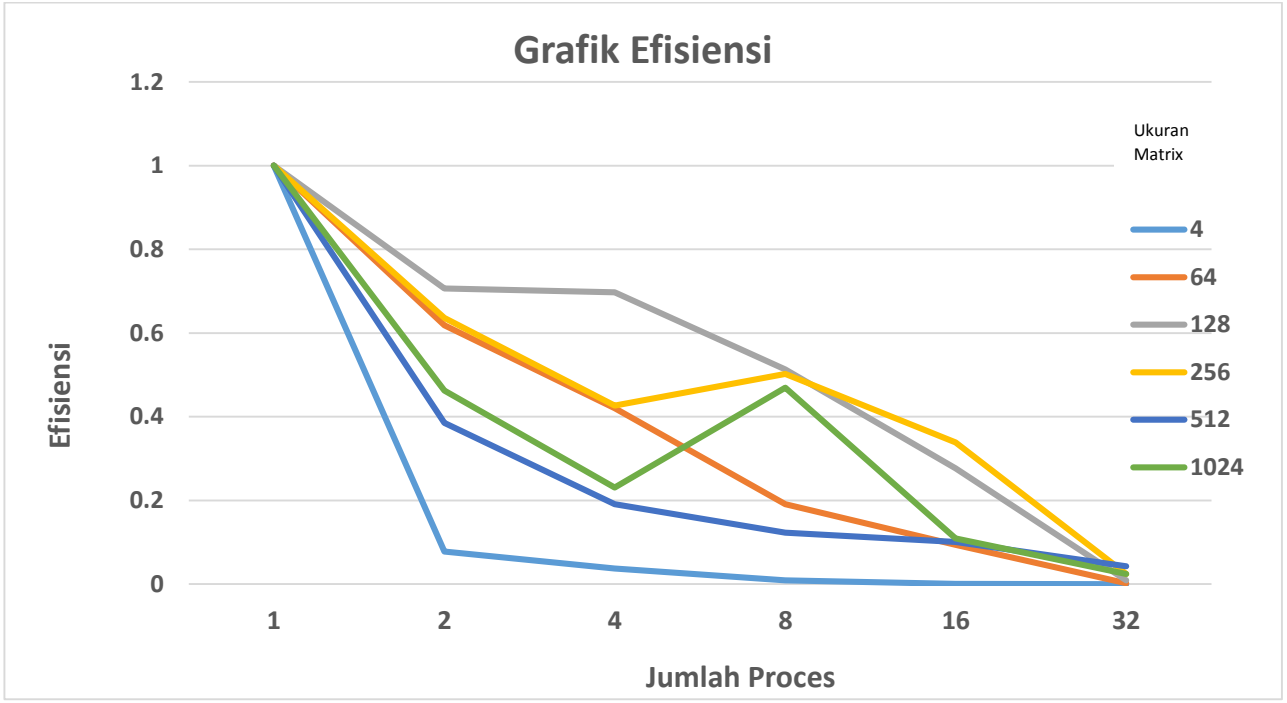
- Grafik waktu banding ukuran matriks, dengan garis plot untuk setiap variasi jumlah proses



- Grafik speedup banding jumlah proses, dengan garis plot untuk setiap variasi ukuran matriks



- Grafik efisiensi banding jumlah proses, dengan garis plot untuk setiap variasi ukuran matriks



d. Analisis hasil

Berdasarkan pengamatan running program yang telah kami buat. Menurut kami, program yang kami buat tidak scalable karena ketika ukuran matriksnya bertambah, efisiensinya tidak selalu meningkat. Namun jumlah proses yang paling optimal adalah 16 buah karena jika terlalu banyak akan ada proses yang tidak ditangani oleh processor secara langsung sehingga membutuhkan waktu switching yang lebih lama.