Implementasi Komputasi Paralel Dengan Compute Unified Device Architecture (CUDA) Untuk Perhitungan Simple Linear Regression

Danang Haryo Sulaksono[1], Enggar Alfianto[2], Siti Agustini[3]
Institut Teknologi Adhi Tama Surabaya[1,2,3]
e-mail: danang_h_s@itats.ac.id

ABSTRACT

Nvidia CUDA began with Nvidia's research on GPGPU (Versatile Computing in the Graphics Processing Unit). From this research emerged CUDA technology for a parallel place that has been implemented in GPUs made by Nvidia today so that it allows GPUs (supported by CUDA technology) to become open architecture such as CPUs. In this research, This study uses the NVIDIA GTX 1050 ti with 4 GB GPU memory specifications with 768 core units. The scheme of the parallel computing method that will be designed using computing with CUDA will be implemented in linear regression for the calculation of Wave Energy Converter data. In the implementation process, several processes will be started from the data input, process and output processes. The results of the process consist of information on the execution time, which will be made a comparison of each CUDA method. In this study, computational benchmarks with published GPUs process linear regression calculations. The duration of computing will use milliseconds (ms), and the test will be carried out ten times as much by entering data starting from 10,000 to 100,000. Set the data for computing using Wave Energy Converter data. Linear regression computing time with data between 10,000 to 100,000 shows very fast computing time, between values between 341 ms and 833 ms. This proves that calculations with CUDA are very efficient for calculations with large processes.
Keyword: *Automosi, contoh, embedded system, kendaraan air, sistem dinamis.*

ABSTRAK

Nvidia CUDA berawal dari riset Nvidia untuk GPGPU (*General-Purpose Computing on Graphics Processing Unit*). Dari riset tersebut muncul teknologi CUDA untuk suatu pemrosesan paralel yang sudah diimplementasikan dalam GPU buatan Nvidia saat ini, sehingga memunginkan GPU (didukung oleh teknologi CUDA) menjadi aristektur terbuka seperti CPU. Dalam penelitian ini, kami memanfaatkan GPU sebagai media kalkulasi. Data yang digunakan dalam kalkulasi adalah data *Wave Energy Converters* yang berjumlah ribuan. Penelitian ini menggunakan NVIDIA GTX 1050 ti dengan spesifikasi memori GPU 4 GB dengan core 768 unit. Skema dari metode komputasi paralel yang akan dirancang menggunakan komputasi dengan CUDA akan diimplementasikan pada regresi linier untuk perhitungan data *Wave Energy Converters*. Dalam proses implementasi ini akan terdapat beberapa proses dimulai dari proses input data, proses dan output. Adapun output dari proses tersebut berupa informasi *excecutiontime* yang akan dijadikan pembanding dari tiap metode CUDA. Pada penelitian ini benchmark komputasi dengan CUDA meliputi lamanya GPU memproses perhitungan regresi linier. Lamanya komputasi akan menggunakan satuan *milissecond (ms)* dan pengujian akan dilakukan sebanyak sepuluh kali secara bertahap dengan memasukkan data dimulai dari 10.000 sampai dengan 100.000. Set data untuk komputasi menggunakan data Wave Energy Converters. Waktu komputasi regresi linier dengan data antara 10.000 sampai dengan 100.000 menunjukkan waktu komputasi yang sangat cepat, antara nilai antara 341 ms sampai dengan 833 ms. Hal ini membuktikan bahwa komputasi dengan CUDA sangat efisien untuk perhitungan dengan proses yang besar.
Kata kunci: Automosi, contoh, *embedded system*, kendaraan air, sistem dinamis.

PENDAHULUAN

Percepatan waktu komputasi menggunakan GPU-CUDA pada perhitungan data *Wave Energy Converters* dengan pola regresi. Pada era digital, pemrosesan data dengan cepat sangat diperlukan. Karena saat ini data apapun bersifat digital, sehingga ide mempercepat waktu komputasi menjadi isu sentral dalam dunia komputasi, informatika dan teknologi informasi. Dalam pengolahan data, percepatan waktu komputasi mengandalkan dua cara. Yang pertama adalah memanfaatkan algoritma. Cara ini biasanya digunakan dalam metode analaisis data dengan persamaan yang kompleks. Misalkan dalam dunia fisika untuk mencari nilai energi dengan metode Density Functional Theory. Cara kedua adalah dengan memanfaatkan multi core computation. Cara ini dapat menghasilkan nilai komputasi paling cepat, dikarenakan proses perhitungan yang digunakan adalah paralel, sehingga percepatannya

memenuhi hukum Amdahl dan hukum Gustafson [1]. Istilah dalam komputasi paralel untuk mempercepat perhitungan adalah speed-up. Potensial speed-up memenuhi Hukum Amdahl dengan persamaan sebagai berikut [1].

$$S = \frac{1}{1} - p + (\frac{p}{s})$$

Dimana S adalah potensial speedup, p adalah persentase waktu eksekusi dan s adalah speedup laten dari hasil eksekusi yang telah diparalelkan. Sedangakan hukum yang diutarakan oleh Gustafson dituliskan dalam persamaan matematis

$$S = 1 - p + sp$$

Hukum yang diutarakan gustafson ini lebih realistik, karena Gustafson menganggap bahwa potensial paralel berbanding lurus secara linier terhadap jumlah prosesor. Pernyataan Gustafson berbeda dengan Amhdahl yang mengatakan potensial paralel yang memiliki potensial terbatas dengan nilai maksimum 90% dari nilai speedup oleh jumlah prosesor [1].

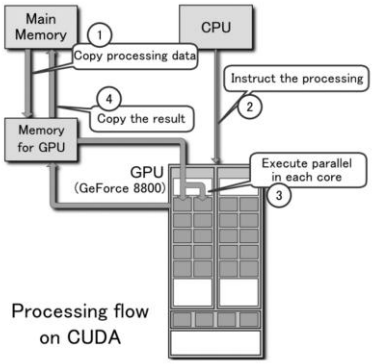
Dalam proses paralel, dikenal dua media yang sering digunakan, pertama adalah media core dalam prosesor, kedua adalah core dalam GpU. Kedua media memiliki kelebihan dan kekurangan masing-masing. Core dalam prosesor dapat dengan mudah digunakan untuk proses paralel, terutama yang membutuhkan komputasi continue antar algoritma. Karena core dalam processor dapat berkomunikasi secara langsung antar core, tanpa harus melibatkan memori lain. Sedangkan GPU memiliki kelebihan jumlah core ribuan, namun satu sama lain tidak dapat berkomunikasi secara langsung. Sehingga tidak dapat digunakan untuk proses kalkulasi continue.

Dalam penelitian ini, kami memanfaatkan GPU sebagai media kalkulasi. Data yang digunakan dalam kalkulasi adalah data *Wave Energy Converters* yang berjumlah ribuan. Kami memilih GPU karena data tersebut dapat dilakukan kalkulasi regresi linier tanpa harus melibatkan kalkulasi continue. Sehingga kalkulasi menggunakan GPU adalah pilihan yang tepat.

TINJAUAN PUSTAKA

CUDA

Nvidia CUDA berawal dari riset Nvidia untuk GPGPU (*General-Purpose Computing on Graphics Processing Unit*). Dari riset tersebut muncul teknologi CUDA untuk suatu pemrosesan paralel yang sudah diimplementasikan dalam GPU buatan Nvidia saat ini, sehingga memunginkan GPU (didukung oleh teknologi CUDA) menjadi aristektur terbuka seperti CPU. Kelebihan pemrosesan di GPU adalah GPU memiliki arsitektur banyak inti yang bekerja secara paralel dan setiap inti menjalankan ribuan thread secara simultan sehingga GPU computing menyediakan sebuah efisensi waktu yang besar dari sebuah pemrosesan atau kalkulasi dibandingkan dengan CPU computing.



Gambar 1. Alur Proses pada CUDA

Gambar 1 adalah proses dari CUDA, GPU memory akan mengambil data dari main memory CPU dan CPU mengirim instruksi yang akan diproses oleh GPU, setelah itu GPU akan memproses dengan *Core* paralelnya. Setelah output keluar maka hasil akan dikembalikan kembali ke main memory. Pemorgraman CUDA seperti membuat program C biasa. Saat kompilasi, *syntax – syntax* C biasa akan diproses oleh compiler C, sedangkan *syntax* dengan keyword CUDA akan diproses oleh *compiler* CUDA (nvcc) CUDA memungkinkan *programmer* untuk mengeksekusi program pada GPU. Program – program yang akan implementai akselerasi CUDA adalah programprogram yang dapat dipecah menjadi banyak eksekusi paralel. Semakin banyak CUDA *Core* maka akan semakin cepat performa komputasi paralel.

Regresi Linier

Regresi digunakan untuk melihat bentuk hubungan antar variabel melalui suatu persamaan. Terdapat tiga jenis regresi yang digunakan sesuai dengan tujuan analisis yaitu Regresi Linier Sederhana, Regresi Linier Berganda, dan Regresi non Linear. Hubungannya bisa berupa hubungan sebab akibat selain itu juga dapat mengukur seberapa besar suatu variabel mempengaruhi variabel lain dan dapat digunakan untuk melakukan peramalan nilai suatu variabel berdasarkan variabel lain. Regresi linier sederhana adalah suatu metode yang digunakan untuk melihat bentuk hubungan antar satu variabel independent (bebas) dan mempunyai hubungan garis lurus dengan variabel dependennya (terikat). Sebuah variabel hasil observasi yang diperoleh sangat mungkin dipengaruhi oleh variabel lainnya, misalkan tinggi badan dan berat badan seseorang. Untuk suatu tinggi tertentu ada besaran berat badan yang mempengaruhi, demikian juga sebaliknya.

Namun kenyataanya hubungan antar variabel bebas dan variabel terikat jarang sekali sesederhana itu. Biasanya banyak faktor atau dalam hal ini kita sebut banyak variabel bebas yang menentukan atau dapat mempengaruhi variabel terikat. Untuk kasus demikian maka akan diselesaikan dengan Regresi linier Berganda.Model Persamaan Regresi Linear Sederhana adalah seperti berikut ini :

$$Y = a + bX$$

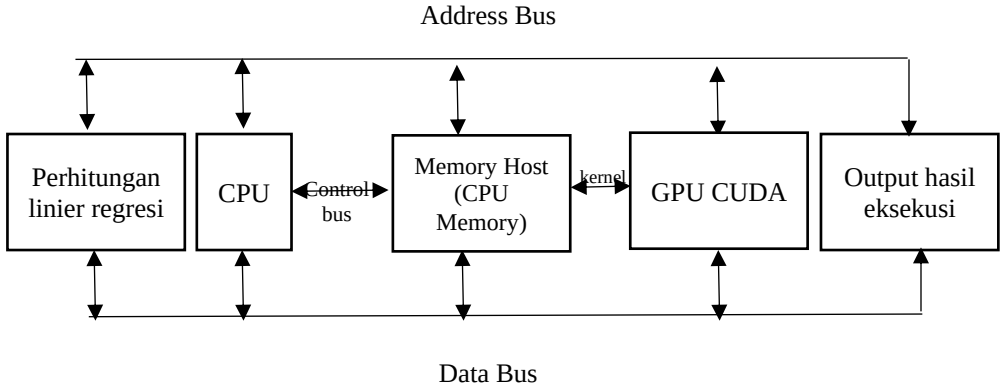
Dimana :
Y = Variabel Response atau Variabel Akibat (Dependent)
X = Variabel Predictor atau Variabel Faktor Penyebab (Independent)
a = konstanta
b = koefisien regresi (kemiringan); besaran Response yang ditimbulkan oleh Predictor.

Nilai-nilai a dan b dapat dihitung dengan menggunakan Rumus dibawah ini :

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

METODE

Penelitian ini menggunakan NVIDIA GTX 1050 ti dengan spesifikasi memori GPU 4 GB dengan core 768 unit. Skema dari metode komputasi paralel yang akan dirancang menggunakan komputasi dengan CUDA akan diimplementasikan pada regresi linier untuk perhitungan data *Wave Energy Converters*. Dalam proses implementasi ini akan terdapat beberapa proses dimulai dari proses input data, proses dan output. Adapun output dari proses tersebut berupa informasi *excecutiontime* yang akan dijadikan pembanding dari tiap metode CUDA.



Gambar 2. Blok Diagram.

merupakan gambaran umum blok diagram dari alur perancangan, Input berupa, perhitungan regresi linier. Pertukaran data dan instruksi komputasi melalui address bar dan data bus, data input komputasi dapat diproses langsung di CPU sehingga keluar output hasil proses dan waktu eksekusi, atau dapat di paralelisasi melalui CUDA GPU kemudian akan keluar output hasil proses dan waktu eksekusi. Memory host mengirim data dan instruksi melalui kernel menuju GPU. Setiap kernel terhubung ke satu grid dalam GPU, tiap grid memiliki block core paralel untuk komputasi, dan tiap block core terdapat multi thread. Data dan instruksi yang diterima akan dikomputasi melalui block core paralel dari GPU.

HASIL DAN PEMBAHASAN

Benchmark Komputasi CUDA

Pada penelitian ini benchmark komputasi dengan CUDA meliputi lamanya GPU memproses perhitungan regresi linier. Lamanya komputasi akan menggunakan satuan *milissecond (ms)* dan pengujian akan dilakukan sebanyak sepuluh kali secara bertahap dengan memasukkan data dimulai dari 10.000 sampai dengan 100.000. Set data untuk komputasi menggunakan data Wave Energy Converters yang diambil dari UCI dataset dengan alamat https://archive.ics.uci.edu/ml/datasets/Wave+Energy+Converters

Tabel 1. Data hasil benchmark komputasi perhitungan regresi linier dengan CUDA

Percobaan ke-	Jumlah data	Hasil Komputasi (dalam ms)
1	10.000	341,748
2	20.000	388,834
3	30.000	492.533
4	40.000	523,674
5	50.000	588,039
6	60.000	637,032
7	70.000	677,834
8	80.000	740,790
9	90.000	783,728

10	100.000	833,462

KESIMPULAN

Waktu komputasi regresi linier dengan data antara 10.000 sampai dengan 100.000 menunjukkan waktu komputasi yang sangat cepat, antara nilai antara 341 ms sampai dengan 833 ms. Hal ini membuktikan bahwa komputasi dengan CUDA sangat efisien untuk perhitungan dengan proses yang besar.

UCAPAN TERIMA KASIH

Ucapan terima kasih diberikan kepada R3-VIORA yang berkerjasama dan mendukung sepenuhnya penelitian ini dengan memberikan waktu dan tempat usahanya sebagai bahan penelitian.

DAFTAR PUSTAKA

[1] J.B. Kulkarni, A. A. Sawan2, V. S. Inamdar, Database Processing by Linear Regression on GPU using CUDA, Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011), 2011.

[2] K. Bagus, A. T. Bharata, S. N. Akhmad, Analisis Perbandingan Komputasi GPU dengan CUDA dan Komputasi CPU Untuk *Image* dan *Video Processing*, Universitas Gadjah Mada Yogyakarta, Indonesia, 2015.

[3] A. C. P. Siregar dan D. H. Sulaksono, (2018), Komputasi Penentuan Kualitas pada Fiber Optik Berdasarkan Rugi Daya dengan Gaussian Naive Bayes Menggunakan Teknologi CUDA, Institut Teknologi Adhitama Surabaya, Indonesia, 2018.

[4] B. Blaise, Introduction To Parallel Computing, Lawrence livermore National laboratory, Retrieved 2007.

[5] T. Mahajan and S. Masih, Enhancing Blowfish File Encryption Algorithm through Parallel Computing on GPU. IEEE International Conference on Computer, Communication and Control. 2015.

[6] A. horga, " Genetic algorithms –CPU vs GPU implementation discussion Metaheuristics course report".

ISSN (print): 2686-0023
ISSN (online): 2685-6875

Halaman ini sengaja dikosongkan