ANALISIS PERFORMANSI KOMPUTASI PARALEL DENGAN GRAPHICAL PROCESSING UNIT

Arief Nur Andono[1], Tri Brotoharsono[2], Adiwijaya[3]

[1]Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Perkembangan teknologi mengantarkan pada sistem yang dapat bekerja secara adaptif atau dapat menyesuaikan dengan kondisi yang terjadi. Salah satu metode adaptif yang telah dikembangkan adalah Jaringan Syaraf Tiruan (JST). Dengan kelebihan algoritma JST, terdapat pula kelemahan yaitu berupa sulitnya menemukan parameter yang dapat memberikan hasil yang optimal seperti mengetahui jumlah Hidden Neuron yang memberikan MSE paling kecil.

Penelitian ini yaitu algoritma paralel dengan GPU, diharapkan dapat menyelesaikan permasalahan ini yaitu dengan cara melakukan pencarian nilainilai parameter JST secara paralel sehingga pencarian parameter yang lebih baik dapat dilakukan dengan lebih cepat serta konsumsi resource komputer yang lebih kecil.

Dari desain komputasi yang dilakukan menggunakan desain Non-uniform Memory Access didapatkan hasilnya sangat signifikan bahwa komputasi dengan GPU dapat melakukan komputasi 800 kali lebih cepat dibandingkan CPU bahkan lebih pada komputasi 150 JST. Nilai peningkatan ini jauh lebih besar dibanding jumlah CUDA core itu sendiri dimana GPU yang digunakan pada TA ini hanya memiliki 128 core akan tetapi penggunaan CUDA dengan desain ini tidak menurunkan penggunaan resource komputer dan bahkan menambah penggunaan memori karena untuk keperluan penyediaan data.

Kata Kunci : JST, MSE, Hidden Neuron, GPU, CUDA, Non-Uniform Memory Access, CUDA core.

Abstract

Technology development running to systems that could adapt or could change itself from some trigger from condition. One of adaptive method that has been developed for years is Neural Network (NN). NN is not only has so many advantage but also some disadvantages. One of them is there is no exact way to find out how many hidden neuron that should be used to get minimum Mean Square Error (MSE).

This research, Parallel Algorithm using GPU, trying to solve this problem that this algorithm could find out the parameters using parallel so the computation could be faster with lower resource.

Using Non-Uniform Memory Access design in computation show the result is so significant that GPU computation could compute 800 times faster than CPU on 150 NN computation. This speedup above the number of GPU core that using 128 core although it has disadvantages that the computation still using the same processor utilization and even higher memory resource to store complete data.

Keywords : NN, MSE, Hidden Neuron, GPU, CUDA, Non-Uniform Memory Access, CUDA core.

1. Pendahuluan

1.1 Latar belakang

Perkembangan teknologi mengantarkan pada sistem yang dapat bekerja secara adaptif atau dapat menyesuaikan dengan kondisi yang terjadi. Salah satu metode adaptif yang telah dikembangkan adalah Jaringan Syaraf Tiruan (JST). JST telah banyak digunakan untuk sistem yang dapat belajar sendiri dengan data yang telah ada seperti kebutuhan sistem prediksi harga, prediksi jurusan yang diminati dalam kasus pemilihan jurusan, dll.

Akan tetap penggunaan JST bukan tanpa kekurangan, salah satunya adalah sistem JST memiliki kemungkinan parameter yang tidak pasti dan sangat beragam. Hal ini berdampak pada pencarian solusi terbaik harus dilakukan dengan metode *trial and error* untuk setiap kemungkinan solusi. Hal ini tentu saja sangat sulit untuk dilakukan dan membutuhkan waktu pengerjaan yang lama untuk mendapatkan hasil terbaik untuk setiap kemungkinan solusi.[5]

Untuk menyelesaikan masalah tersebut maka dikembangkan metode penyelesaian dengan sistem paralel. Komputasi paralel telah banyak diimplementasikan seperti membangun *High Performance Computing* (HPC) *cluster* yaitu menggunakan banyak komputer dengan membagi tugas ke komputer-komputer yang dinyatakan sebagai *compute node* atau *slave node*, kemudian hasil komputasi tersebut dikelola kembali oleh *master node*. Akan tetapi peningkatan kemampuan komputasi paralel tidak sebanding dengan peningkatan jumlah komputer yang digunakan sehingga biaya yang diinvestasikan tidak sebanding dengan hasil yang didapat. Hal lain untuk mengatasi hal ini adalah dengan penggunaan *multicore general processor* (CPU), namun perkembangan *processor multicore* ini pun masih terbatas. Ide lain untuk melakukan komputasi paralel adalah dengan menggunakan *Graphical Processing Unit* (GPU) yang secara tradisional hanya melakukan komputasi grafis kini diarahkan agar dapat melakukan komputasi sebagaimana general processor.

Salah satu yang mendorong solusi penggunaan GPU adalah tidak memungkinkannya penambahan CPU dalam komputer tanpa mengubah komponen lain terutama *motherboard* komputer dan cara ini memerlukan biaya tinggi karena menggunakan *motherboard* khusus. Oleh sebab itu pemanfaatan GPU dilakukan karena dapat langsung digunakan melalui *interface PCIexpress* atau melalui *port external* tanpa modifikasi *hardware* secara keseluruhan. Hal lain mendorong penggunaan GPU adalah lebih cepatnya perkembangan GPU dibanding CPU itu sendiri yaitu perkembangan peningkatan jumlah *core* di GPU jauh lebih cepat dibanding CPU sedangkan perkembangan *clock rate* CPU saat ini tidak mengalami peningkatan yang signifikan sehingga perkembangan CPU juga mengarah pada *multicore processor*.

1.2 Perumusan masalah

Perumusan masalah dari tugas akhir ini adalah :

1. Bagaimana pengaruh GPU pada komputasi terhadap hasil perhitungan *Floating-Point Operation per Second* (FLOPS) dibandingkan tanpa menggunakan GPU?
2. Bagaimana pengaruh penggunaan GPU terhadap waktu penyelesaian (*response time*) masalah komputasi pada Jaringan Syaraf Tiruan?
3. Bagaimana pengaruh penggunaan GPU terhadap utilisasi CPU saat melakukan proses komputasi?
4. Bagaimana pengaruh penggunaan GPU terhadap utilisasi memori

1.3 Ruang Lingkup dan Batasan Masalah

Ruang lingkup dan batasan masalah dalam menyelesaikan kasus ini antara lain :

1. Pembatasan penggunaan sebuah komputer dengan sebuah GPU.
2. Penggunaan GPU Nvidia GTS 250 1GB *memory*
3. Penggunaan bahasa pemrograman hanya menggunakan bahasa python.
4. Hanya menggunakan sistem operasi linux.
5. Implementasi pada studi kasus jaringan syaraf tiruan *multi layer perceptron* dengan data harga emas dunia dalam dollar dengan menerapkan algoritma back propagation dengan 1 output.

1.4 Tujuan

Tujuan penulisan tugas akhir ini adalah :

1. Mengukur pengaruh penggunaan GPU pada HPc terhadap hasil perhitungan Floating-point Operation per Second (FLOPS).
2. Mendapatkan hasil response time komputasi paralel dengan GPU maupun tidak untuk diketahui besarnya pengaruh komputasi GPU pada HPC.
3. Memonitor dan menganalisis besar pengaruh penambahan GPU terhadap utilisasi CPU.
4. Memonitor dan menganalisis pengaruh penggunaan GPU terhadap utilisasi memori utama.

1.5 Metodologi penyelesaian masalah

Metode yang digunakan dalam tugas akhir ini antara lain:

1. Identifikasi Masalah.
   Memahami masalah yang akan diangkat, mempelajari bidang pengkajian serta menentukan cakupan masalah yang akan diselesaikan.
2. Studi Literatur
   Memperdalam dan memahami studi pustaka mengenai :
   a) Algoritma paralel
   b) Algoritma Jaringan Syaraf Tiruan
   c) Komputasi paralel dengan GPU
   d) Monitoring utilisasi CPU dan Memori
3. Perancangan Sistem

Membuat skenario perancangan pengujian sistem, antara lain:

a) Merancang algoritma serial dengan CPU yang mengerjakan beberapa JST dalam eksekusi.
b) Merancang Arsitektur paralel computing dengan GPU.
c) Merancang skenario pengujian sistem.
d) Merancang skenario monitoring sistem.

4. Implementasi dan pengumpulan data.
   Melakukan implementasi berdasarkan hasil perancangan yang telah diskenariokan dan melakukan pengambilan data antara lain :
   a) Nilai FLOPS dari sistem komputasi paralel dengan dan tanpa GPU.
   b) Nilai response time pemrosesan dari sistem komputasi paralel dengan dan tanpa GPU.
   c) Nilai utilitas CPU pada sistem komputasi dengan dan tanpa GPU.
   d) Nilai utilitas memori pada sistem komputasi dengan tanpa GPU.
5. Analisis dan Kesimpulan
   Analisis data dan pengambilan kesimpulan dari data hasil penelitian yang telah didapatkan.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan, dapat dihasilkan beberapa kesimpulan antara lain :

1. Response time penggunaan GPU jauh lebih baik dibanding tidak menggunakan GPU.
2. Peningkatan performa komputasi GPU pada kasus ini sangat signifikan dan dapat mencapai melebihi jumlah total CUDA core yang ada pada GPU dengan desain NUMA antar block.
3. Perulangan pemanggilan kernel CUDA tidak mempengaruhi performansi komputasi.
4. Penggunaan GPU yang memanggil kernel CUDA dengan metode *synchronous* tidak menurunkan utilitas CPU.
5. Desain program untuk komputasi CUDA paralel menggunakan lebih banyak memori komputer dibanding program serial.
6. Penggunaan GPU dengan lebih banyak block akan menurunkan fungsi utama GPU sebagai display *frame per second* dalam rendering video.

5.2 Saran

Berdasarkan hasil analisis terhadap komputasi menggunakan GPU menunjukkan metode pemrograman dengan GPU ini masih dapat dikembangkan sehingga dapat menghasilkan kemampuan komputasi yang lebih baiklagi. Adapun saran-saran yang dapat diajukan adalah:

1. Penggunaan studi kasus atau data yang memiliki keterikatan yang lebih rendah sehingga dapat lebih paralel seperti pada kasus pemrosesan video dimana gambar *frame* pada video dapat dipartisi.
2. Penggunaan *map-reduce* dalam desain paralel untuk memanfaatkan thread CUDA agar jumlah kernel dapat ditingkatkan serta mengurangi penggunaan percabangan atau perulangan dalam komputasi.
3. Penggunaan *asynchronous* dalam memanggil kernel CUDA diharapkan dapat menurunkan utilisasi CPU.

Referensi

[1] *Jason sanders* and *Edward Kandrot*. CUDA by Example, an Introduction General-Purpose GPU Programming. Addison-Wesley. Juli 2010

[2] *David B. Kirk* and *Wen-mei W. Hwu*. Programming Massively Parallel Processors: A Hands-on Approach. Morgan-Kaufmann. 2009.

[3] *James Balfour*. CUDA Threads and Atomics, Report of Nividia Research. 2011

[4] Laplante, Phillip A.Real-Time Systems Design and Analysis.Wiley-Interscience. 2004.

[5] Suyanto. Artificial Intelligence. Penerbit Informatika. Juni 2007.

[6] http://en.wikipedia.org/wiki/FLOPS. Diakses pada 5 Februari 2011

[7] http://mathema.tician.de/software/pycuda. Diakses pada 16 April 2011

[8] V. Kumar, A. Grama, A. Gupta, and G. Karypis, Introduction to Parallel Computing, 2nd edn, Harlow, UK, Pearson Education 2003.

[9] http://en.wikipedia.org/wiki/CUDA. Diakses pada 24 Juni 2011.

[10] *Almasi, G.S.* and *A. Gottlieb*. Highly Parallel Computing . Benjamin-Cummings publishers, Redwood City, CA.1989.

[11] S.V. Adve et al. Parallel Computing Research at Illinois : The UPCRC Agenda. November 2008.

[12] Barney, Blaise. "Introduction to Parallel Computing". Lawrence Livermore National Laboratory. Retrieved 2007-11-09.

[13] Hennessy, John L. and David A. Patterson. Computer Architecture: A Quantitative Approach. 3rd edition, Morgan Kaufmann, p. 43. ISBN 1-55860-724-2.2002.

[14] http://www.behardware.com/articles/659-5/nvidia-cuda-preview.html. Diakses 27 Juli 2011.

[15] http://http.developer.nvidia.com/GPUGems3/gpugems3_ch31.html. Diakses pada 27 Juli 2011.