

Komputasi Paralel Integral Definit Rangkap Tiga dengan Metode Monte Carlo di Cluster Beowulf (Juli, 2013)

Muhammad Zulhaj Aliyansyah, Mahfudz Shidiq, Ir., MT., Muhammad Aswin, Ir., MT.
Teknik Elektro Universitas Brawijaya
jendral.hxr@gmail.com

ABSTRACT

Parallel computing is concurrent execution of arithmetic/logic operations on multiple processors. Beowulf cluster implements parallel computing system using consumer-grade hardware and utilizing free/open-source software.

Monte Carlo Integration is pleasingly parallel in which communications between nodes are minimal thus can be implemented efficiently on Beowulf cluster.

Accuracy and speed of Monte Carlo triple integration is compared against block integration, in which Monte Carlo method has greater accuracy in trade of slower speed.

Speed of Monte Carlo Integration grows linearly to concurrent process count as process count doesn't exceed available cores count in the cluster.

Energy efficiency in the cluster is constant within homogeneous cluster and drops as inferior-performing nodes are added to the cluster.

Keywords—Beowulf cluster, Monte Carlo method, numerical integration, parallel computing, triple integral.

ABSTRAK

Komputasi paralel adalah proses pengoperasian aritmatik atau logik terhadap bilangan-bilangan yang sejenis secara bersamaan pada prosesor majemuk. *Cluster Beowulf* mengimplementasikan sistem komputasi paralel dengan perangkat komputer *consumer-grade* dengan perangkat lunak berlisensi *free/open-source*.

Komputasi integral dengan metode Monte Carlo bersifat *pleasingly parallel* sehingga dapat dengan sankil diimplementasikan pada *cluster Beowulf*.

Ketelitian dan kecepatan komputasi integral rangkap tiga dengan metode Monte Carlo dibandingkan dengan metode balok, di mana metode Monte Carlo memiliki ketelitian yang lebih baik namun dengan kecepatan yang lebih lambat.

Kecepatan komputasi paralel integral rangkap tiga metode Monte Carlo berbanding lurus dengan cacah *proses* yang berjalan paralel iika cacah proses tidak melebihi cacah *core*.

Kesanggikan penggunaan energi sistem komputasi paralel cenderung konstan pada *cluster* dengan spesifikasi *node* yang homogen dan menurun saat terdapat *node* dengan spesifikasi yang inferior.

Kata Kunci—cluster Beowulf, integral numerik, integral rangkap tiga, komputasi paralel, metode Monte Carlo.

I. PENDAHULUAN

A. Integral Rangkap Tiga Metode Monte Carlo dan Balok

Deskripsi formal integral rangkap tiga dengan batas integrasi a, b, c, d, e, f dan variabel integrasi x, y, z adalah:

$$M = \int_a^b \int_c^d \int_e^f m(x, y, z) dx dy dz$$

dengan batas integrasi a, b, c, d, e, f dapat merupakan konstanta atau fungsi dari variabel yang bukan variabel integrasi yang dibatasinya.

Metode Monte Carlo adalah metode integrasi yang menggunakan bilangan acak untuk menentukan titik sampel dalam suatu integral. Hasil integral adalah rerata perkalian fungsi integran titik sampel dengan volume *cubicle* yang dibentuk oleh batas-batas integrasinya.

Pendekatan integral rangkap tiga dengan metode Metode Monte Carlo varian rerata sampel didefinisikan dalam deskripsi formal sebagai:

$$M \sim \frac{1}{N} \sum_{i=0}^{N-1} m(x_i, y_i, z_i) [(f_i - e_i)(d_i - c_i)(b_i - a_i)]$$

dengan

$$x_{i+1} = \text{rand}[a_i, b_i]$$

$$y_{i+1} = \text{rand}[c_i, d_i]$$

$$z_{i+1} = \text{rand}[e_i, f_i]$$

Deskripsi formal integral rangkap tiga metode balok dengan titik tinjau di tengah *cubicle* didefinisikan sebagai:

$$M \sim \sum_{i=0, j=0, k=0}^{N-1, O-1, P-1} m(x_i, y_j, z_k) \frac{(b_{jk} - a_{jk})}{N} \frac{(d_{ik} - c_{ik})}{O} \frac{(f_{ij} - e_{ij})}{P}$$

dengan

$$x_i = a_{jk} + (i + 0.5) \frac{(b_{jk} - a_{jk})}{N}$$

$$y_j = c_{ik} + (j + 0.5) \frac{(d_{ik} - c_{ik})}{O}$$

$$z_k = e_{ij} + (k + 0.5) \frac{(f_{ij} - e_{ij})}{P}$$

Integral rangkap tiga metode Monte Carlo akan dibandingkan dengan dengan metode balok yang memiliki kompleksitas algoritma yang sama dengan nilai $O(n)$. Parameter yang dibandingkan adalah ketelitian dan kecepatan.

B. Komputasi Paralel

Komputasi adalah proses yang dijalankan prosesor dalam sebuah rentang satuan waktu yang meliputi:

- 1) operasi baca: menerima masukan berupa data dengan ukuran tertentu,
- 2) operasi penghitungan: melakukan operasi aritmatik atau logik terhadap masukan,
- 3) operasi tulis: mengembalikan hasil penghitungan dengan ukuran tertentu.

Komputasi paralel adalah eksekusi proses pengoperasian aritmatik atau logik yang sejenis secara bersamaan dalam proses.^[1] Sebuah masalah komputasi yang kompleks dibagi menjadi bagian-bagian yang lebih kecil, sederhana, dan dijalankan di beberapa prosesor secara bersamaan sehingga keseluruhan proses dapat diselesaikan dengan lebih cepat.

Secara teori peningkatan kecepatan komputasi paralel dijelaskan dengan hukum Amdahl:

$$S = \frac{1}{\frac{1-\alpha}{P} + \alpha}$$

dengan

S = peningkatan kecepatan (perkalian)

P = cacah prosesor

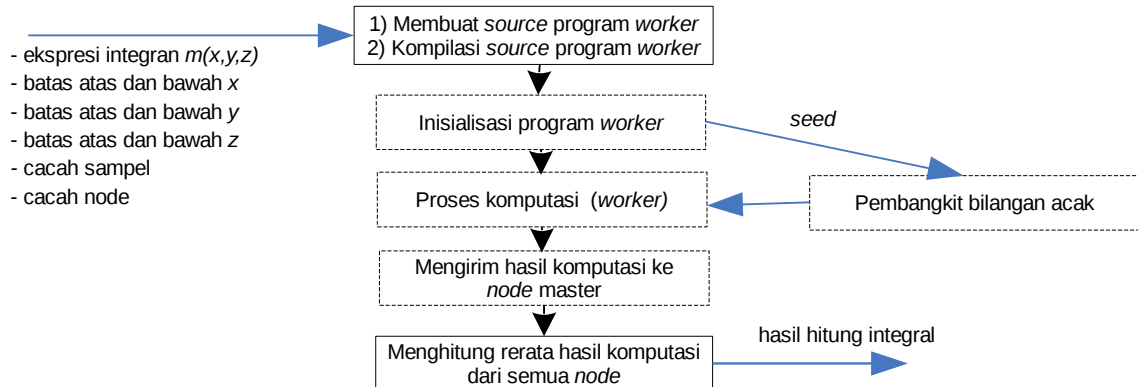
α = rasio bagian rutin program yang tidak dapat diparalelisasi terhadap keseluruhan rutin program

Peningkatan kecepatan komputasi paralel secara teori dibatasi kendala bahwa sebuah rutin tidak dapat dipecah menjadi unit yang lebih kecil dan harus dijalankan secara sekuensial di satu prosesor. Secara praktek peningkatan kecepatan komputasi paralel dibatasi performa perangkat keras, penjadwalan proses sistem operasi, konfigurasi lingkungan komputasi dan overhead rutin program karena komunikasi dan sinkronisasi antar proses.

C. Cluster Beowulf

Cluster Beowulf adalah kumpulan komputer yang menggunakan perangkat keras *consumer-grade* dengan harga yang murah (contoh: komponen komputer bekas pakai), menggunakan perangkat lunak gratis dengan lisensi yang bersifat FOSS (*free open source software*) untuk membangun keseluruhan sistem, dan mengakomodasi keperluan komputasi paralel.

Cluster Beowulf memiliki keuntungan karena struktur dan metode implementasinya, antara lain: skalabilitas, konvergensi, fleksibilitas konfigurasi, *failover*, kemudahan, kecepatan, dan harga implementasi yang murah dibandingkan sistem cluster konvensional.



Gambar 2. Abstraksi Program Sistem Komputasi Paralel Integral

Komputasi dengan cluster Beowulf melibatkan 4 hal dalam pertimbangan sistem:^[2]

1. sistem perangkat keras,
2. manajemen sumber daya komputasi,
3. pustaka pemrograman paralel,
4. algoritma paralel.

II. PERANCANGAN DAN IMPLEMENTASI

Perangkat yang digunakan untuk implementasi sistem komputasi paralel integral definit rangkap tiga metode Monte Carlo diuraikan sebagai berikut:

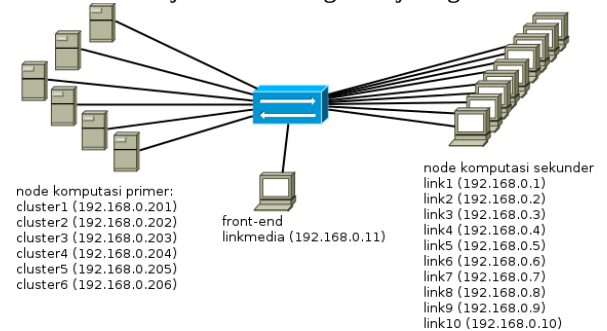
□□ Perangkat keras:

- 10 unit komputer personal HP Compaq CQ3138L: Intel Core 2 Duo E7500, DDR3 2 GB,
- 6 unit komputer personal: Intel i3 550/540, DDR3 2 GB
- 1 unit komputer personal: Intel Dual Core E5400, DDR2 2 GB,
- switch DES 1024R dan perkabelan UTP Cat 5e,
- energimeter Wanf D02A

□□ Perangkat lunak:

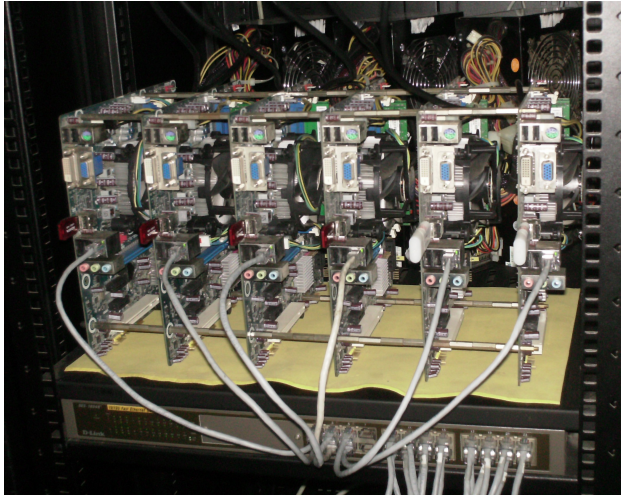
- GNU/Linux Salix64: kernel 3.2.29, glibc 2.15, gcc 4.7.1,
- GNU/Linux WattOS 12.04.1 LTS: kernel 3.2.0, glibc 2.15, gcc 4.6.5,
- Open MPI 1.7.

Gambar 1 menjelaskan konfigurasi jaringan cluster:



Gambar 1. Konfigurasi Jaringan Lokal Sistem Komputasi Paralel

Gambar 2 menjelaskan abstraksi program sistem komputasi paralel integral definit rangkap tiga metode Monte Carlo. Gambar 3 memperlihatkan unit node komputasi primer.



Gambar 3. Node Komputasi Primer Sistem Komputasi Paralel

Sistem komputasi paralel integral definit rangkap tiga menggunakan NFS untuk menyebarkan kunci publik SSH dan *executable binary* dari program komputasi integral definit rangkap tiga metode Monte Carlo dan balok.

Masukan program diberikan melalui argumen dari *stdin* dan keluaran ditampilkan melalui *stdout*. Pembangkit bilangan acak yang digunakan adalah pembangkit bilangan acak pustaka standar sistem operasi Linux.

Kode program dan skrip *shell* dapat lihat di alamat terlampir (setelah daftar pustaka).

III. PENGUJIAN DAN ANALISIS

Terdapat 4 persamaan integral sebagai soal uji, 4 persamaan tersebut adalah:^[3]

Soal Uji 1

$$M = \int_{-3; -\sqrt{9-x^2}; -\sqrt{9-x^2-x^2}}^{3; \sqrt{9-x^2}; \sqrt{9-x^2-x^2}} (x^2+y^2+z^2)^{1.5} dydzdx$$

Soal Uji 2

$$M = \int_{0;0;0}^{3; \sqrt{9-x^2}; 2} \sqrt{x^2+y^2} dzdydx$$

Soal Uji 3

$$M = \int_{0;0;0}^{2; \sqrt{4-x^2}; \sqrt{4-x^2-y^2}} z \sqrt{4-x^2-y^2} dzdydx$$

Soal Uji 4

$$M = \int_{1;3;0}^{2; x; \sqrt{3}y} \frac{y}{y^2+x^2} dzdydx$$

Ketelitian komputasi integral definit metode Monte Carlo akan dibandingkan dengan metode balok dan penghitungan secara analitis dengan mengubah cacah sampel pengujian. Metode yang lebih teliti memiliki galat yang lebih kecil.

Ketepatan komputasi integral definit metode Monte Carlo diukur dengan menghitung nilai varians dari beberapa pengujian. Ketepatan akan semakin tinggi jika nilai varians semakin kecil.

Waktu proses komputasi integral definit metode Monte Carlo dan metode balok diukur untuk memastikan skalabilitas kompleksitas algoritma dengan variabel cacah sampel yang berubah.

Cacah *proses* digunakan sebagai variabel dalam pengukuran peningkatan kecepatan dan kesanggulan kecepatan komputasi.

Cacah *node* digunakan sebagai variabel dalam pengukuran peningkatan kecepatan, kesanggulan kecepatan komputasi dan penggunaan energi.

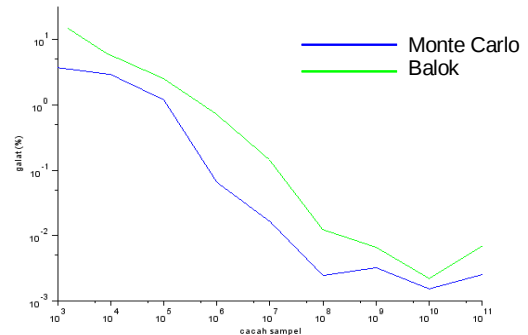
A. Pengujian Pengaruh Cacah Sampel

Cacah sampel yang digunakan dalam pecobaan pengaruh cacah sampel dalam komputasi paralel integral definit rangkap tiga metode Monte Carlo adalah deret logaritmik dengan basis 10: 1e3, 1e4, 1e5, 1e6, 1e7, 1e8, 1e9, 1e10, dan 1e11.

Cacah sampel yang digunakan dalam pecobaan pengaruh sampel dalam komputasi paralel integral definit rangkap tiga dengan metode balok: 1536, 8232, 98304, 943296, 9725376, 98304000, 994121664, 9963862464, dan 99972228696.

Pengujian pengaruh cacah sampel dijalankan di *node* komputasi primer dengan cacah *proses* 24. Pengujian pengaruh cacah sampel dijalankan dengan menjalankan skrip *shell* *sample.sh* dan *sample-blok.sh* dari *front-end*.

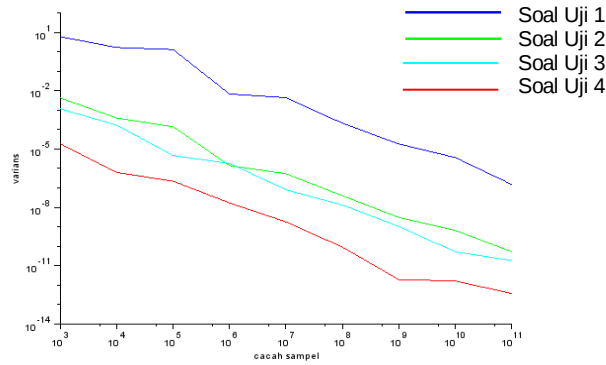
Gambar 4 memperlihatkan pengaruh cacah sampel terhadap nilai galat salah satu soal uji (Soal Uji 1).



Gambar 4. Grafik Galat terhadap Cacah Sampel Soal Uji 1

Nilai galat cenderung semakin kecil dengan meningkatnya cacah sampel pada keempat soal uji. Hal ini membuktikan bahwa ketelitian komputasi paralel integral definit rangkap tiga metode Monte Carlo dan balok berbanding lurus dengan cacah sampel.

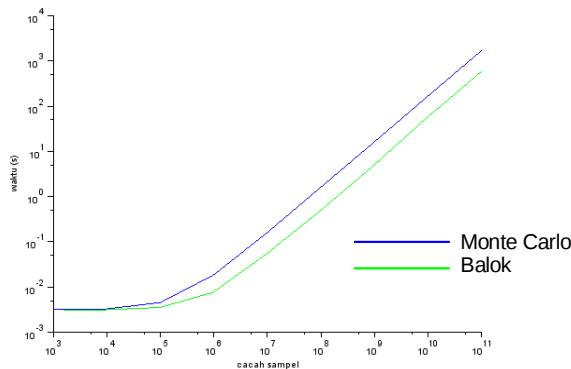
Metode Monte Carlo memiliki ketelitian yang lebih baik daripada metode balok karena memiliki nilai galat yang lebih kecil. Gambar 5 memperlihatkan pengaruh cacah sampel terhadap nilai varians keempat soal uji.



Gambar 5. Grafik Nilai Varians terhadap Cacah Sampel

Nilai varians komputasi paralel integral definit rangkap tiga metode Monte Carlo menurun cenderung linier seiring peningkatan cacah sampel pada semua soal uji. Hal ini membuktikan bahwa ketepatan komputasi paralel integral definit rangkap tiga metode Monte Carlo berbanding lurus dengan cacah sampel.

Gambar 6 memperlihatkan pengaruh cacah sampel terhadap waktu proses komputasi integral definit rangkap tiga Metode Monte Carlo dan Balok. Salah satu soal uji (Soal Uji 4)



Gambar 6. Grafik Waktu Komputasi terhadap Cacah Sampel Soal Uji 4

Waktu komputasi meningkat secara linier seiring pertambahan cacah sampel setelah setelah waktu komputasi melebihi ~ 10 ms. Hal ini membuktikan bahwa algoritma komputasi paralel integral definit rangkap tiga metode Monte Carlo dan balok memiliki kompleksitas algoritma yang sama dengan nilai $O(n)$.

Komputasi paralel integral definit rangkap tiga metode balok lebih cepat dibanding metode Monte Carlo dengan perbandingan antara 0,8109 hingga 5,0437 kali.

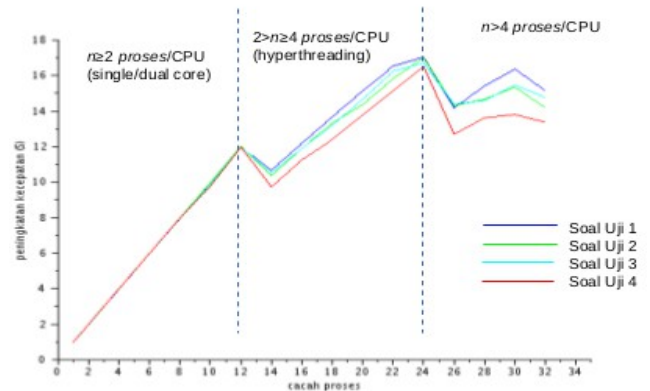
B. Pengujian Pengaruh Cacah Proses

Cacah sampel yang digunakan dalam percobaan pengaruh cacah proses dalam komputasi paralel integral definit rangkap tiga metode Monte Carlo adalah $1e9$.

Parameter yang diukur adalah waktu proses komputasi dengan resolusi 1 ms. Dari waktu komputasi dapat dihitung S (*speedup*) yang merupakan perbandingan waktu komputasi dengan waktu komputasi dengan cacah proses tunggal dan kesangkilan komputasi yang merupakan perbandingan S dan cacah proses.

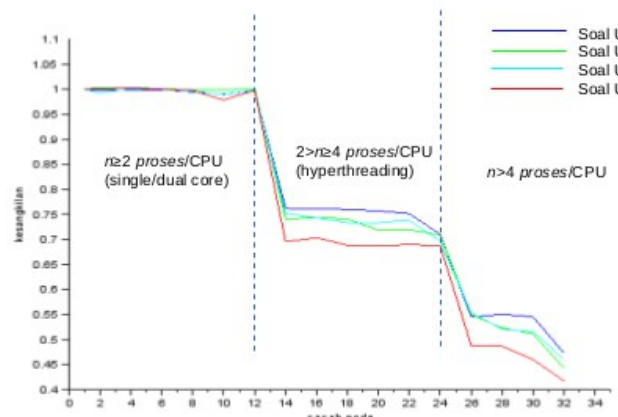
Pengujian pengaruh cacah proses dijalankan di *node* komputasi primer. Pengujian pengaruh cacah sampel dijalankan dengan menjalankan skrip *shell* numprocs.sh dari *front-end*.

Gambar 7 menunjukkan pengaruh cacah proses terhadap peningkatan kecepatan komputasi integral definit rangkap tiga metode Monte Carlo keempat soal uji.



Gambar 7. Grafik Peningkatan Kecepatan terhadap Cacah Proses

Gambar 8 memperlihatkan pengaruh cacah proses terhadap kesangkilan komputasi integral definit rangkap tiga metode Monte Carlo pada keempat soal uji.



Gambar 8. Grafik Kesangkilan terhadap Cacah Proses

Peningkatan kecepatan (S) meningkat secara ideal dengan cacah proses antara 1 dan 12 di mana tiap unit prosesor dalam 6 unit *node* komputasi primer beroperasi secara *single core* atau *dual core*.

Terdapat penurunan nilai S saat prosesor mengaktifkan *hyperthreading* (cacah proses 14), namun meningkat dengan gradien yang lebih kecil hingga cacah proses 24. Terjadi penurunan nilai S saat cacah proses melebihi cacah *core* (24) dalam *node* komputasi primer.

Nilai peningkatan kecepatan (S) maksimal $\sim 16-17$ kali dengan cacah proses 24 yang sesuai dengan cacah *core* dalam *node* komputasi primer. Hal ini masih jauh dari nilai peningkatan kecepatan secara teoritis (~ 24).

Kesangkilan komputasi paralel integral definit rangkap tiga metode Monte Carlo ideal saat cacah proses ≤ 12 , menurun secara berundak setelah cacah proses 12 dan 24 yang merupakan kelipatan cacah *core* *node* komputasi primer.

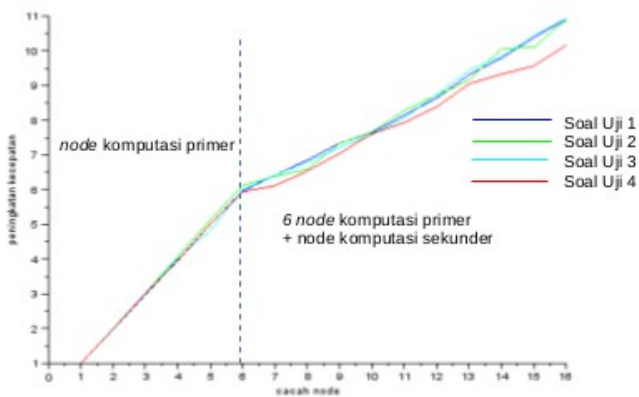
C. Pengujian Pengaruh Cacah Node

Peningkatan kecepatan dihitung dengan perbandingan waktu proses komputasi dengan cacah *node* majemuk terhadap waktu pengujian dengan cacah *node* tunggal. Kesangkalan komputasi adalah perbandingan peningkatan kecepatan (S) pada pengujian dengan cacah node majemuk terhadap cacah *node*.

Konsumsi energi dihitung dengan perkalian waktu komputasi dan daya tipikal tiap *node* komputasi. Kesangkalan penggunaan energi adalah perbandingan penggunaan energi dengan cacah *node* majemuk terhadap penggunaan energi pada pengujian dengan cacah *node* tunggal.

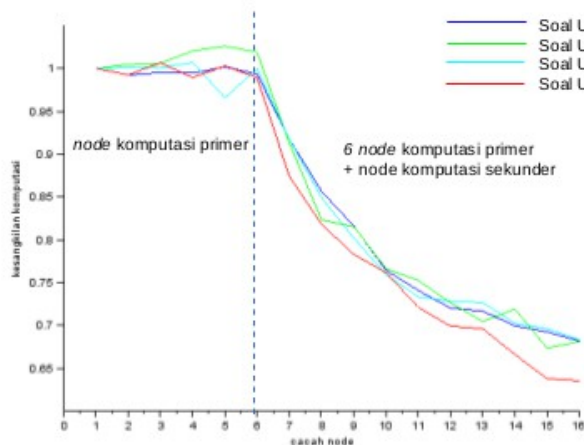
Pengujian pengaruh cacah *node* dijalankan di *node* komputasi primer dan sekunder dengan menjalankan skrip *shell nodes.sh* dari *front-end* dengan cacah sampel $1e9$.

Gambar 9 memperlihatkan pengaruh cacah *node* terhadap peningkatan kecepatan komputasi integral definit rangkap tiga metode Monte Carlo pada keempat soal uji..



Gambar 9. Grafik Peningkatan Kecepatan (S) terhadap Cacah *node*

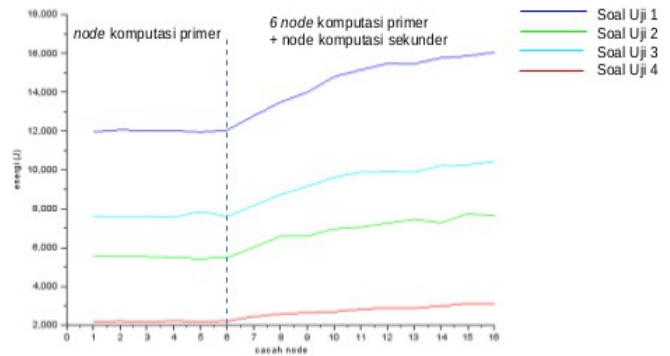
Gambar 10 memperlihatkan pengaruh cacah *node* terhadap kesangkalan komputasi integral definit rangkap tiga metode Monte Carlo keempat soal uji.



Gambar 10. Grafik Kesangkalan Komputasi terhadap Cacah *Node*

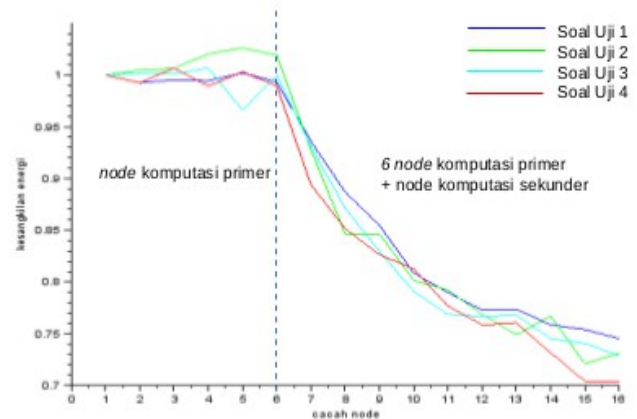
Peningkatan kecepatan (S) meningkat secara linier saat *node* pengujian menggunakan *node* komputasi primer (cacah *node* ≤ 6). Peningkatan kecepatan (S) meningkat secara linier dengan gradien yang lebih kecil saat *node* pengujian menurutsertakan *node* komputasi sekunder (cacah *node* > 6).

Gambar 11 menunjukkan pengaruh cacah *node* terhadap penggunaan energi proses komputasi integral definit rangkap tiga metode Monte Carlo pada keempat soal uji.



Gambar 11. Grafik Penggunaan Energi terhadap Cacah *Node*

Gambar 12 menunjukkan pengaruh cacah *node* terhadap kesangkalan penggunaan energi komputasi integral definit rangkap tiga metode Monte Carlo pada keempat soal uji.



Gambar 12. Grafik Kesangkalan Penggunaan Energi terhadap Cacah *Node*

Penggunaan energi komputasi integral definit rangkap tiga metode Monte Carlo cenderung konstan saat pengujian menggunakan *node* komputasi primer (cacah *node* > 6) kemudian meningkat cenderung linier saat pengujian menurutsertakan *node* komputasi sekunder. Hal ini disebabkan karena spesifikasi dan performa *node* komputasi sekunder yang inferior dibandingkan *node* komputasi primer.

Kesangkalan penggunaan energi komputasi integral definit rangkap tiga metode Monte Carlo cenderung konstan saat pengujian menggunakan *node* komputasi primer kemudian turun cenderung logaritmik saat menurutsertakan *node* komputasi sekunder.

A. Kesimpulan

Berdasarkan pengujian dan analisis sistem komputasi paralel integral definit rangkap tiga, dapat ditarik kesimpulan yang diuraikan dalam uraian berikut.

- 1) Ketelitian komputasi integral definit rangkap tiga metode Monte Carlo dan metode balok berbanding lurus dengan cacah sampel dengan metode Monte Carlo memiliki ketelitian yang lebih baik dibanding metode balok.
- 2) Ketepatan komputasi integral definit rangkap tiga metode Monte Carlo berbanding lurus dengan peningkatan cacah sampel.
- 3) Waktu proses komputasi komputasi integral definit rangkap tiga metode Monte Carlo dan metode balok berbanding lurus dengan cacah sampel secara linier.
- 4) Peningkatan kecepatan komputasi integral definit rangkap tiga metode Monte Carlo meningkat seiring pertambahan cacah proses jika cacah proses tidak melebihi cacah core, sementara kesangkilannya menurun secara step saat cacah proses melebihi kelipatan cacah core dalam node komputasi primer.
- 5) Peningkatan kecepatan meningkat seiring pertambahan cacah node dalam komputasi integral definit rangkap tiga metode Monte Carlo. Kesangkilan komputasi integral definit rangkap tiga metode Monte Carlo konstan saat menggunakan node komputasi primer dan menurun saat menurutsertakan node komputasi sekunder.
- 6) Penggunaan energi proses komputasi integral definit rangkap tiga metode Monte Carlo cenderung konstan saat menggunakan node komputasi primer dan meningkat saat menurutsertakan node komputasi sekunder. Sementara kesangkilan penggunaan energi komputasi integral definit rangkap tiga metode Monte Carlo cenderung konstan saat menggunakan node komputasi primer dan menurun saat menggunakan node komputasi sekunder.

B. Saran

Saran untuk pengembangan lebih lanjut antara lain:

- penggunaan komputasi berbasis GPU,
- penggunaan sistem komputer dengan spesifikasi yang superior.
- penggunaan jumlah node komputasi paralel yang lebih banyak,
- penggunaan sistem komputer yang lebih hemat dalam penggunaan energi,
- komputasi paralel menggunakan perangkat mobile atau embedded,
- penggunaan pembangkit bilangan acak yang adaptif terhadap bentuk persamaan integral,
- penerapan skenario *load balancing* dan *failover*,
- studi kasus komputasi paralel integral dengan model yang lebih kompleks.

- [1] Reif, John., Sangutevar Rajasekaran. 2008. *Handbook of Parallel Computing: Models, Algorithms and Applications*. New York: Chapman & Hall/CRC Press.
- [2] Sterling, Thomas. 2002. *Beowulf Cluster Computing with Linux*. Cambridge: The MIT Press.
- [3] Purcell, Edwin J. Dale Valeberg, Steven E. Rigdon. 2007. *Calculus*. Upper Saddle River, NJ: Pearson Education Inc.
- [4] Rosenthal, Jeffrey S. 2000. *Parallel Computing and Monte Carlo Algorithms*. Toronto: University of Toronto. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.126&rep=rep1&type=pdf> (diakses 28 Maret 2013)
- [5] Dai, Rui-kai., Jian-xin Li, Chun-li Dont, dkk. 2009. A Parallel Monte Carlo Simulation on Cluster System for Particle Transport. <http://origin-www.computer.org/csdl/proceedings/aici/2009/3816/02/3816b031.pdf> (diakses 23 Maret 2013)
- [6] Wang, Ping. 2011. *Modified Monte Carlo for Triple Integral*. Qingdao: Qingdao Agricultural University. <http://origin-www.computer.org/csdl/proceedings/icm/2011/4522/01/4522a234.pdf> (diakses 28 Maret 2013)
- [7] Nakano, Aiichiro. 2013. *Monte Carlo Basics*. <http://cacs.usc.edu/education/phys516/01-1mcbasics.pdf> (diakses 14 April 2013)
- [8] Barrett, Brian W. 2012. *Open MPI Data Transfer*. Sandia National Laboratories. http://www.open-mpi.org/video/internals/Sandia_BrianBarrett-1up.pdf http://www.open-mpi.org/video/internals/Sandia_BrianBarrett-2up.pdf (diakses 12 Maret 2013)
- [9] Woodman, Lawrence. 2009. *Setting up a Beowulf Cluster Using Open MPI on Linux*. <http://techtinkering.com/2009/12/02/setting-up-a-beowulf-cluster-using-open-mpi-on-linux> (diakses 30 Oktober 2012)
- [10] Radajewski, Jacek., Douglas Eadline. 1998. *Beowulf HOWTO*. <http://www.ibiblio.org/pub/linux/docs/HOWTO/archive/Beowulf-HOWTO.html> (diakses 7 November 2012)

Zulhaj Aliyansyah, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juni 2013, Komputasi Paralel Integral Definit Rangkap Tiga dengan Metode Monte Carlo di Cluster Bewoulf.

Dosen Pembimbing: Mahfudz Shidiq, Ir., MS
Muhammad Aswin, Ir. MT.

Kode dan skrip implementasi dapat dilihat di GIT <https://github.com/jendralhxr/montice>