

Analisis Kinerja Sistem Komputasi Grid Menggunakan Perangkat Lunak Globus Toolkit Dan MPICH-G2

M. Iqbal¹, R. A. Gani², S. Ahdan³, M. Bakri⁴, Wajiran⁵

^{1,2} Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Universitas Lampung
Jl. Prof dr Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Rajabasa, Bandar Lampung

^{3,4,5} Fakultas Teknik dan Ilmu Komputer (FTIK), Universitas Teknokrat Indonesia
Jl. H.ZA Pagaram, No 9-11, Labuhanratu, Bandar Lampung

Email : muhammadiqbal@fmipa.unila.ac.id¹, rusli.gani88@gmail.com²,
syaifulahdan@teknokrat.ac.id³, muhammadbakri@teknokrat.ac.id⁴, wajiran@gmail.com⁵

Abstract

The grid computation was developed by using fedora 14 operating system and globus toolkit software, NTP and certificate authority to built a grid system as well as MPICH-G2 as a reference to run parallel program. It was examined with matrix multiplication and was compared with single computation system based on processing time. This research employed 2 units of personal computer which earned a faster processing time than the single system, if only the program is divided into three process/jobs or more. If there was only 1 or 2 jobs, the process will not run faster than single computation. It can be concluded that the grid system will perform better if there are more nodes and core processor in each node and also balanced with the amount of the process/job.

Keywords: Grid computing, Globus toolkit, MPICH-G2, parallel computing

Abstrak

Perkembangan kecepatan prosesor dan jalur komunikasi jaringan komputer memicu dikembangkannya sistem komputasi paralel yang kompleks yang disebut juga sistem komputasi grid. Komputasi grid ini menggunakan sistem operasi Fedora 14 dan perangkat lunak Globus Toolkit, NTP dan Certificate Authority untuk membangun sistem grid serta MPICH-G2 sebagai pustaka untuk menjalankan program paralel. Diuji menggunakan perkalian matriks dan dibandingkan dengan sistem komputasi tunggal berdasarkan lama waktu pemrosesan. Penelitian menggunakan 2 komputer yang menghasilkan waktu pemrosesan yang lebih cepat dari sistem tunggal dengan syarat program dibagi menjadi 3 proses/job atau lebih, karena dengan hanya 1 atau 2 job maka prosesnya tidak akan lebih cepat dari komputasi tunggal. Dapat disimpulkan bahwa sistem grid akan lebih baik jika memiliki banyak node dan core prosesor pada tiap node dan diimbangi dengan banyaknya proses/job

Kata kunci: Grid computing, Globus toolkit, MPICH-G2, parallel computing

1. Pendahuluan

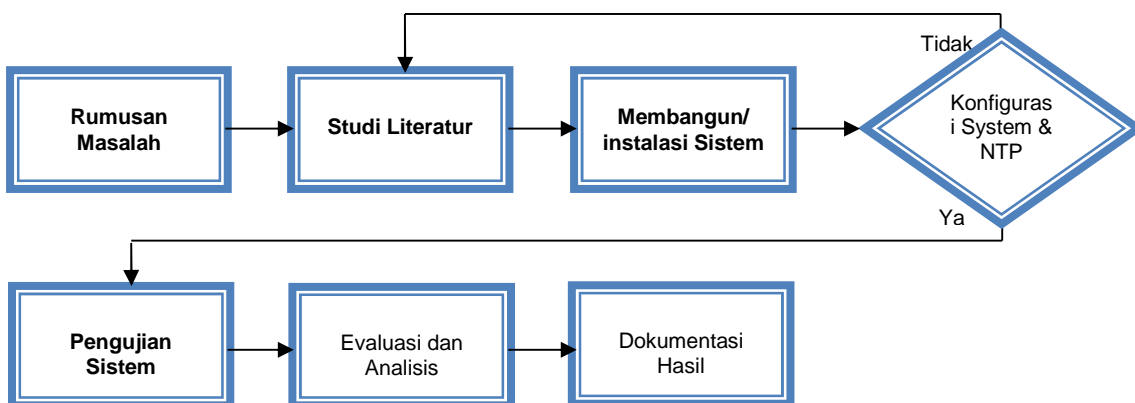
Perkembangan kecepatan prosesor berkembang sesuai dengan hukum Moore, dimana dalam 18 bulan sekali kecepatan prosesor akan meningkat 2 kali lipat. Perkembangan kecepatan prosesor tersebut diikuti perkembangan bandwidth jaringan komputer yang juga semakin besar. Semakin berkembangnya jalur komunikasi ini membuka peluang untuk menggabungkan kekuatan komputasi dari sumber-sumber komputasi yang terpisah dan

terhubung melalui jaringan komputer. Gabungan dari banyak sumber daya komputasi tersebut akan digunakan untuk menjalankan aplikasi yang berat dan kompleks agar proses yang dilakukan oleh aplikasi tersebut berjalan lebih efektif dan efisien (Masruro, 2007).

Dalam mengembangkan sistem komputasi grid, saat ini telah disediakan suatu perangkat lunak yang mampu menerapkan komputasi grid tersebut dan salah satunya adalah perangkat lunak open source Globus Toolkit. Dengan menggunakan Globus Toolkit ini kita dapat menerapkan sistem grid seperti membagi sumberdaya CPU dan juga menggabungkannya, pengiriman data dengan cepat dan aman baik di dalam sistem itu sendiri (dalam satu cluster) maupun antar cluster yang terpisah dan terhubung melalui jaringan komputer. Setelah itu sistem akan dianalisis kinerjanya, apakah sudah sesuai dengan hipotesa awal bahwa sistem grid memiliki keunggulan pada kecepatan pemrosesan dibandingkan dengan sistem komputasi tunggal (Ferreira, dkk., 2003 dan Foster, 2006).

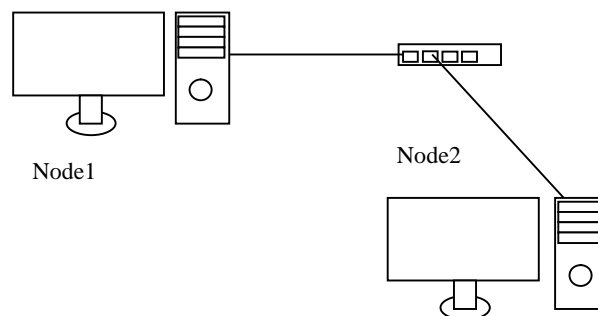
2. Metodologi Penelitian

Tahapan penelitian ini dapat dijelaskan dalam ilustrasi pada gambar 1. Dimana ada beberapa tahapan dimulai dari rumusan masalah dan studi literatur, kemudian membangun rancangan sistem grid, dan memberikan analisa dan konfigurasi sistem dan NTP pada rancangan sistem tersebut. Akhirnya mendokumentasikan hasil evaluasi dari konfigurasi tersebut.



Gambar 1. Metode penelitian

Dalam penelitian ini dibangun suatu sistem komputasi grid yang terdiri dari 2 personal komputer dan terhubung melalui jaringan lokal. Gambar 2 akan menjelaskan bagaimana arsitektur dari sistem grid yang akan dibangun



Gambar 2. Rancangan sistem grid

Pada gambar 2 akan dirancang sebuah sistem grid satu cluster yang memiliki spesifikasi perangkat keras yang sama. Dalam perancangannya, tiap-tiap komputer akan diinstall beberapa perangkat lunak yang digunakan untuk membangun sistem komputasi grid. Sebagai perinciannya dijelaskan sebagai berikut:

- a) Komputer 1 akan diinstall JDK, Apache Ant, NTP (Network Time Protokol) client, GT4.0.8 (Globus Toolkit bundle), SimpleCA dan MPICH-G2 sebagai API untuk menjalankan pemrosesan paralel.
- b) Komputer 2 akan diinstall JDK, Apache Ant, GT4.0.8 (Globus Toolkit bundle), NTP (Network Time Protokol) server, SimpleCA dan MPICH-G2 sebagai API untuk menjalankan pemrosesan paralel.

3. Hasil Dan Pembahasan

Komputasi grid yang dibangun menggunakan perangkat lunak Globus Toolkit dan menggunakan MPICH sebagai pustaka dan alat komunikasi untuk menjalankan program aplikasi secara paralel (Carpenter, 2002) (Imam Rusadi, F. 2006). Dalam penelitian ini, program paralel yang dijalankan diatas sistem grid akan dibandingkan dan dianalisis performanya dengan sistem komputasi tunggal. Sistem ini juga minimal memiliki 2 (dua) komputer tunggal agar tujuan grid minimal (1 cluster) dapat terpenuhi sehingga sistem dapat dikatakan sistem grid. Sistem yang dibuat memiliki sebuah komputer yang akan dijadikan mesin eksekusi yang nantinya akan diberikan hak kepada pengguna yang memang diberikan hak akses untuk mengeksekusi program paralel diatas layanan gridl. Satu mesin dijadikan sebagai server grid dan mesin aplikasi sedangkan mesin lain digunakan untuk server grid maupun server Network Time Protocol (NTP). Tiap mesin harus memiliki sistem waktu yang sama persis dengan menjalankan aplikasi Network Time Protocol (NTP) (Mills, 2003).

Pada tahap analisis kinerja sistem grid yang telah dibangun dan telah diintegrasikan dengan MPICH-G2 agar dapat menjalankan aplikasi paralel melalui layanan GLOBUS TOOLKIT 4 (globusrun) dibandingkan dengan aplikasi yang dijalankan dengan kompilasi dan dijalankan secara normal (non paralel) menggunakan gcc. Komputasi grid yang dibangun kemudian dilakukan pengujian. Pengujian dilakukan untuk mengetahui apakah fungsi-fungsi yang telah direncanakan bekerja dengan baik atau tidak (Prasetyawan et al., 2018). Aplikasi yang akan dijadikan perbandingan adalah aplikasi perkalian matriks menggunakan bahasa pemrograman C++. penggunaan bahasa pemrograman C++ disini dikarenakan dukungan MPICH-G2 dan Globus Toolkit 4 terhadap bahasa C, C++ dan fortran yang sudah penuh.

Pada tabel 1 memperlihatkan hasil dari running program perkalian matriks secara paralel menggunakan layanan Globus Toolkit 4. Dengan beberapa opsi yaitu dengan jumlah proses dari mulai 1 proses hingga 4 proses dimana proses disini merupakan bagian dari program matriks yang dipecah menjadi beberapa job dan akan dikerjakan oleh sistem(sampai 4 proses/job dikarenakan pada kasus ini prosesor yang digunakan adalah core2duo yang berarti dalam 1 komputer memiliki 2 fisik prosesor) dan opsi ordo matriks 500x500 hingga 3000x3000.

Tabel 1 Pengujian sistem grid menggunakan program matriks

No	Ordo matriks (nxn)	Waktu eksekusi program (detik)				
		Komputasi tunggal	Komputasi grid (np = jumlah proses/job)			
			np 1	np 2	np 3	np 4
1	500	1.357	4.523	1.525	1.187	1.108
2	1000	17.582	35.297	19.398	13.601	10.542
3	1500	67.361	104.275	82.157	53.169	39.410

4	2000	143.568	257.905	155.543	107.234	84.265
5	2500	311.768	431.841	328.265	227.645	170.059
6	3000	573.054	785.998	586.718	434.846	264.523

Pada tabel 1 terlihat bahwa pada pengujian sistem grid dibandingkan dengan sistem komputasi tunggal menggunakan program perkalian matriks terlihat bahwa dalam kasus sistem grid menggunakan satu proses akan mendapatkan hasil yang selalu lebih lama dibandingkan dengan sistem tunggal, sedangkan penggunaan dua proses akan mendapatkan hasil yang relatif mirip/hampir sama dengan sistem tunggal, hal ini dikarenakan sistem tunggal memang memiliki dua core prosesor. Lama waktu proses dengan tiga dan empat proses baru memperlihatkan hasil eksekusi yang lebih cepat dibandingkan dengan sistem tunggal.

Dalam menjalankan program paralel diatas yang menggunakan layanan MPICH-G2 maka kerja dari proses tersebut adalah sebagai berikut:

1. Sebelum program perkalian matriks dijalankan, maka hal pertama yang harus dilakukan adalah menjalankan perintah `grid-proxy-init` untuk mendaftarkan sertifikasi CA pada mesin eksekusi ke mesin administrator CA dengan masa berlaku sertifikat yaitu 12 jam.
2. Setelah sertifikat terdaftar pada Admin CA maka lakukan perintah `mpirun` untuk mengeksekusi program paralel matriks pada mesin grid, dimana tiap-tiap mesin grid harus memiliki file program tersebut. Ketika `mpirun` dijalankan, MDS (Monitoring dan Discovery Service) bekerja agar pengguna dapat menggunakan sumber daya milik grid, mesin-mesin mana yang dapat melakukan eksekusi terutama prosesor dan memori.
3. Proses di MPICH-G2 setelah perintah `mpirun` dilaksanakan selanjutnya MPICH-G2 membuat file Resource Specification Language (RSL) yang berisikan hasil eksekusi.
4. MPICH-G2 selanjutnya memanggil pustaka yang terdapat di Globus yaitu DUROC (Dynamic Update Request Online Coallocator) dan DUROC menerima informasi dari kode RSL yang telah dibuat sebelumnya. DUROC kemudian menjadwalkan dan menjalankan aplikasi perkalian matriks ke semua sumber daya yang telah ditentukan.
5. Pustaka DUROC menggunakan layanan GRAM (Grid Resource Allocation dan Management) yaitu salah satu komponen dari Globus yang digunakan untuk memulai dan mengatur sub job ke setiap sumber daya. Untuk setiap sumber daya DUROC menghasilkan request GRAM ke GRAM Server. Dengan kondisi autentifikasi pengguna telah didapatkan, autentifikasi pengguna diproses di local scheduler di masing-masing sumber daya, Subjob di semua sumber daya terlibat dalam mengeksekusi bagian program perkalian matriks MPI yang dikendalikan oleh DUROC dan dengan bantuan GRAM Server, Hasil akhirnya DUROC dan pustaka MPICH-G2 menggabungkan semua subjob menjadi satu job MPI.

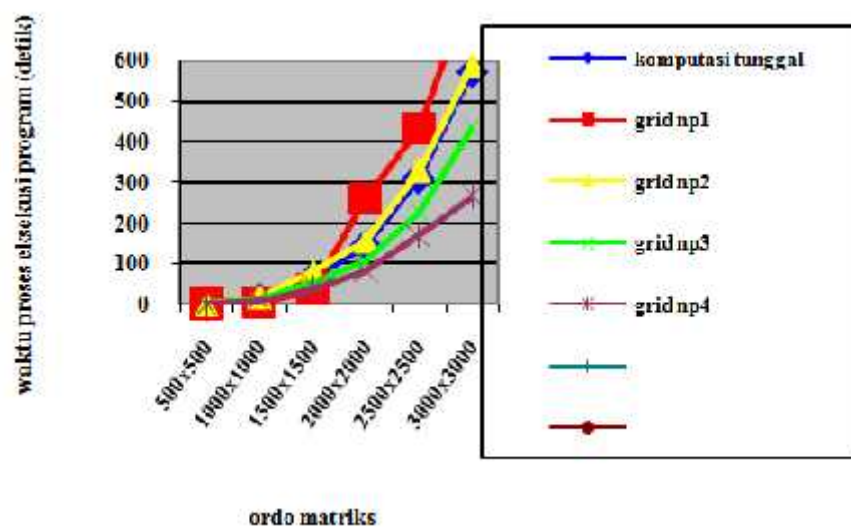
MPICH-G2 dapat memilih metode komunikasi terbaik diantara satu atau beberapa dua proses. Misalkan jika dua proses terletak di sumber daya yang sama, MPICH-G2 menggunakan persediaan sumber daya lokal (jika tersedia) untuk menjalankan komunikasi diantara mesin dan apabila kedua proses berada di sumber daya yang berbeda, MPICH-G2 menggunakan Globus Communication (Globus I/O) dan Globus Data Conversion (Globus DC) untuk komunikasi intermachine (TCP) (Hayes, 2009) (OpenMPI).

Pelaksanaan program diawasi oleh DUROC dengan bantuan setiap server GRAM di setiap sumber daya. Saat aplikasi matriks mulai dijalankan, sumber daya di hold, sampai aplikasi selesai. GRAM memberitahukan ke DUROC setiap perubahan state. Semua perhitungan di setiap subjob ditahan oleh DUROC dan dilepaskan ketika semua subjob telah mulai mengeksekusi. Sama halnya, ketika pengguna meminta pemberhentian job, DUROC

akan meneruskan ke GRAM dan GRAM memproses mengirimkan pesan untuk segera dihentikan semua subjob termasuk proses yang sedang berlangsung.

Dilihat dari tabel 1 maka dapat diberikan beberapa analisa diantaranya:

1. Pada perkalian matriks dengan ordo 500 nxn dengan elemen 500, maka perkalian menggunakan sistem grid akan lebih lambat dibandingkan dengan sistem tunggal. Dengan melihat bagaimana sistem grid mengeksekusi program paralel MPI, yaitu dengan membagi sebuah aplikasi menjadi beberapa job, dan job akan dibagi lagi menjadi beberapa subjob dan akan dikirim ke sumberdaya yang ada melalui globus-io (melalui protokol TCP pada jaringan) kemudian setelah subjob selesai dieksekusi, kemudian akan dikembalikan lagi dan akan disatukan menjadi sebuah kesatuan job, maka terlihat bahwa lambatnya proses pengerjaan sistem grid terletak pada proses komunikasi, pengiriman, penerimaan dan penyatuan subjob. Jadi pengerjaan subjob yang semestinya lebih cepat dari proses tunggal dikarenakan telah dibagi-bagi, terlihat malah lebih lambat oleh karena proses komunikasi yang memakan cukup banyak waktu dan tidak mampu diimbangi dengan kecepatan proses eksekusi subjob.
2. Penggunaan hanya 1 jumlah proses pada eksekusi aplikasi MPI sangat tidak efisien, dikarenakan proses komunikasi grid yang memakan waktu cukup lama dan tidak seimbang dengan sumber daya yang digunakan (hanya 1 core prosesor) sehingga hasil untuk tiap eksekusi aplikasi menggunakan 1 proses akan memakan waktu yang lebih lama daripada menjalankan aplikasi sejenis menggunakan sistem tunggal dengan compiler C/C++ biasa.
3. Peningkatan kecepatan proses berbanding lurus dengan jumlah proses yang digunakan dan juga jumlah prosesor yang ada. Terlihat pada tabel 1 pada jumlah proses 2 sedikit lebih lambat dari pemrosesan tunggal, disebabkan pengiriman tiap baris matriks ke proses yang memakan cukup waktu.
4. Penggunaan dua proses terlihat hampir sama waktu eksekusinya dengan program matriks non paralel, hal ini disebabkan prosesor yang digunakan pada sistem merupakan prosesor yang memiliki core ganda. Gambar 3 memperlihatkan perbandingan kecepatan pemrosesan perkalian matriks antara sistem grid globus toolkit dan sistem komputasi tunggal.



Gambar 3 Perbandingan kecepatan pemrosesan antara komputasi grid dan tunggal

Terlihat dari grafik 1 bahwa kecepatan pemrosesan paralel menggunakan layanan grid dan dengan satu proses akan sangat lambat, bahkan jika dibandingkan dengan kecepatan pemrosesan pada komputasi tunggal, sepertinya penggunaan komputasi grid dan algoritma pemrograman paralel benar-benar harus memanfaatkan sumber daya yang banyak (lebih dari satu proses) untuk memperlihatkan kemampuannya, karena pada pemrosesan komputasi paralel menggunakan grid akan memanfaatkan proses kirim dan ambil sehingga waktu komunikasi juga patut dipertimbangkan. Disini stabilitas jaringan juga sangat menentukan, sehingga hasil yang ideal akan sangat sulit didapatkan

4. Kesimpulan

Setelah dilakukan penelitian tentang kinerja sistem komputasi grid menggunakan globus toolkit dan MPICH-G2, didapatkan beberapa kesimpulan sebagai berikut

1. Penggunaan jumlah komputer yang tergabung dalam sistem grid akan berdampak sangat besar dalam performa sistem grid tersebut juga penggunaan komputer yang memiliki prosesor dengan core lebih dari satu.
2. Menjalankan aplikasi paralel menggunakan sistem berbasis grid akan meningkatkan keamanan data, dengan menggunakan layanan SSH dan penggunaan autentikasi menggunakan CA (Certificate Authority) serta pembatasan hak akses dan eksekusi grid dan aplikasi paralel. Diharapkan dengan adanya beberapa sistem keamanan ini, pemrosesan paralel yang menggunakan layanan jaringan akan lebih aman.
3. Kecepatan pemrosesan menggunakan sistem berbasis MPICH dan diatas sistem grid akan sangat dipengaruhi oleh performansi jaringan, jumlah proses dan kalkulasi proses yang dilakukan, jumlah prosesor dan node. Semakin stabil jaringan, banyak jumlah proses yang diimbangi dengan jumlah prosesor dan node maka kecepatan pemrosesan system grid akan meningkat cukup signifikan bila dibandingkan dengan sistem komputasi tunggal / konvensional.

Jika melihat dari hasil penelitian bahwa peningkatan kecepatan grid tidak lebih dari 100% maka hal ini tidak bisa dikatakan efisien bila dihubungkan dengan penggunaan sumber daya terutama penggunaan komputer dan juga sumber daya listrik. Potensi grid akan dapat lebih digunakan jika komputer/node yang tersedia memadai/cukup banyak dan menjalankan aplikasi yang tidak sanggup dijalankan oleh sistem komputasi tunggal.

Referensi

- Masruro, A., 2007. *Komputasi Grid Sebagai Jawaban Keterbatasan Sumber Daya Komputasi*. Jurnal DASI, vol 8, no 4. STMIK AMIKOM Yogyakarta.
- Ferreira, L., dkk. 2003. *Introduction to Grid Computing with Globus, 2nd ed*. IBM, New York.
- Foster, I., 2006. *A Globus Primer*. Early Draft.
- Carpenter, B., 2002. *The Development of Data-Parallel Programming*. NPAC at Syracuse University, NewYork.
- Imam Rusadi, F. 2006. *Evaluasi Lingkungan Pemrograman Paralel Berbasis Message Passing Interface dalam Infrastruktur Komputasi Grid di Universitas Indonesia*. FASILKOM Universitas Indonesia.
- Mills, D.L., *Internet time synchronization: the Network Time Protocol*. IEEE Trans. Communications 39, 1991, 1482-1493.
- Hayes T, 2009. *An Efficient Open MPI Transport System for Virtual Workers Nodes*. Trinity College Dublin.
- OpenMPI Home page: http://www.open-mpi.org/papers/trinity-btl-2009/xenmpi_report.pdf, Tanggal akses : 9 Februari 2011.
- Prasetyawan, P., Ferdianto, Y., Ahdan, S. & Trisnawati, F., 2018. Pengendali Lengan Robot Dengan Mikrokontroler Arduino Berbasis Smartphone. Jurnal Teknik Elektro ITP, 7(2).