Konfigurasi Jaringan Bertingkat Pada Cluster Paralel Orange Pi

Ari Badia Chrstian[1], Dr. Ir. M. Aswin, M.T.[2], Waru Djuriatno, S.T., M.T.[3]
[1]Mahasiswa Teknik Elektro Univ. Brawijaya, [2,3] Dosen Teknik Elekro Univ. Brawijaya
Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya
Jalan MT. Haryono 167, Malang 65145, Indonesia
E-mail: arihwmaster@gmail.com

ABSTRAK

Orange Pi merupakan salah satu PC namun berukuran jauh lebih kecil dan mengonsumsi daya yang jauh lebih rendah. Untuk meningkatkan performa komputasi paralel, akan digunakan komputer paralel pada Orange Pi. Paralelisasi Orange Pi dapat didesain dengan menggunakan sistem paralel bertingkat (Sub-Cluster). Penggunaan konfigurasi jaringan bertingkat diharapkan dapat menambah Bandwith pada saat pemrosesan paralel, sehingga kecepatan komputasi secara keseluruhan dapat meningkat. Pada penelitian ini, digunakan 17 buah Orange Pi yang dibagi dalam 4 sub-cluster. Tiap sub-cluster terdiri atas 4 buah Orange Pi. Setiap sub-cluster terhubung pada router yang berbeda. Pengujian dilakukan sebanyak 2 kali, pada sistem paralel yang sama dan sistem paralel bertingkat. Pengujian kecepatan dilakukan dengan menjalankan program perkalian matriks di kedua sistem paralel. Hasil dari penelitian ini menunjukkan bahwa peningkatan kecepatan didapat pada kasus pemrosesan paralel di pengujian dengan kondisi awal yang sama dengan jumlah proses 17, 51 dan 68. Peningkatan juga didapat pada pengujian dengan semua ukuran program dengan jumlah proses 51 dan 68.
Kata kunci: *paralel, sub-cluster, jaringan sama,ukuran, jumlah proses, MPI.*

ABSTRACT

Orange Pi is a kind of PC, but much smaller in size and power consumption is much lower. To improve the performance of parallel computing, parallel computer will be used with Orange Pi. Orange Pi parallelization can be designed using a parallel system of multilevel (Sub-Cluster). The use of multilevel network configuration is expected to add bandwidth at the time of parallel processing, so that the overall computing speed can be increased. In this study, we used 17 pieces Orange Pi and divided into four sub-clusters. Each sub-cluster consisting of four pieces of Orange Pi. Each sub-cluster connected to a different router. Testing was done 2 times, on the same parallel system and system of multilevel parallel. Testing is done by running matrix multiplication program in two parallel systems. The results of this study indicate that increased computing performance obtained in the case of parallel processing with the same inital cindition with number of process 17, 51, and 68. Increased computing performance also obtained in all program size with number of process 51 and 68.
***Key word**: parallel, sub-cluster, same network, size, number of process, MPI*

I. PENDAHULUAN

Dunia teknologi berkembang dengan sangat pesat. Perkembangan ini didukung dengan kemajuan dunia digital yang sangat pesat. Seiring dengan perkembangan teknologi yang semakin pesat, kebutuhan akan komputasi yang tinggi juga semakin meningkat. Dengan kecepatan komputasi yang tinggi, maka dapat diperoleh waktu proses komputasi yang lebih rendah pula, sehingga kita dapat menyelesaikan suatu permasalahan komputasi dengan waktu yang lebih singkat. Namun saat ini, kecepatan komputasi yang tinggi biasanya di iringi dengan harga perangkat mikroprosesor yang tinggi pula. Ini lah yang menyebabkan timbul ide untuk menggunakan beberapa prosesor dengan kecepatan rendah secara bersama-sama untuk mendapat hasil komputasi yang tinggi. Penggunaan prosesor secara bersama-sama untuk menyelesaikan suatu masalah tertentu ini disebut dengan komputasi paralel.

Platform komputer yang digunakan, komputer paralel (parallel computer), bisa berupa komputer dengan beberapa prosesor mapupun beberapa komputer yang terhubung dengan cara tertentu. Pendekatan tersebut seharusnya mampu meningkatkan kemampuan komputer secara signifikan. Maksudnya adalah p prosesor/komputer mampu menghasilkan hingga p kali kecepatan komputer dengan satu prosesor/komputer. Berapa pun kecepatan prosesor/komputer itu dengan harapan problem yang ad dapat diselesaikan dengan waktu 1/p. Tentu saja, situasi tersebut merupakan situasi ideal yang jarang sekali terjadi. Seringkali, problem tidak dapat dipecah menjadi bagian-bagian kecil. Selain itu, diperlukan interaksi antara masing-masing bagian, baik untuk transfer data atau sinkronisasi. Namun seberapa jauh pengingkatan kecepatan dapat dicapai bergantung pada masalah dan peran parallelisme. Satu hal yang menyebabkan kemampuan komputer paralel menjadi tak terbatas adalah peningkatan kecepatan eksekusi suatu prosesor secara berkelanjutan akan meningkatkan pula kecepatan komputer paralel. Dengan demikian, akan selalu ada problem dengan tantangan besar yang tak bisa diselesaikan dalam waktu yang memadai menggunakan komputer saat ini.

Komputasi paralel tidak hanya dapat digunakan dengan perangkat keras yang biasa digunakan pada komputer personal kita (motherboard, prosesor, ram, harddisk), tapi juga dapat menggunakan open-source single board computer, seperti Orange Pi. Orange Pi merupakan sebuah mini PC berbasis open-source yang sudah memiliki spesifikasi yang cukup untuk menjalankan fungsinya sebagai komputer. Orange Pi sendiri memiliki ukuran yang lebih ringkas jika dibandingkan dengan komputer pada umumnya dan lebih hemat energi.

Pada penggunaan komputer paralel, biasanya digunakan sistem Beowulf Cluster, dimana kumpulan komputer umum yang identik yang digunakan secara bersama dalam satu jaringan local yang sama dengan library dan program yang sudah terinstal sehingga setiap

komputer dapat saling berkomunikasi. Hasilnya adalah komputasi paralel dengan performansi yan tinggi dari perangkat komputer yang dapat didapat secara mudah di toko-toko komputer.

Bandwidth adalah lebar saluran data yang dilewati secara bersama-sama oleh data-data yang di transfer. *Bandwidth* paling banyak digunakan sebagai ukuran kecepatan aliran data. Di dalam jaringan komputer, Bandwidth sering digunakan sebagai suatu sinonim untuk data transfer rate yaitu jumlah data yang dapat dibawa dari sebuah titik ke titik lain dalam jangka waktu tertentu (pada umumnya dalam detik). Jenis Bandwidth ini biasanya diukur dalam bps (bits per second). Adakalanya juga dinyatakan dalam Bps (bytes per second). Secara umum, koneksi dengan Bandwidth yang besar/tinggi memungkinkan pengiriman informasi yang besar seperti pengiriman gambar/images dalam video presentation.

Besarnya saluran atau Bandwidth akan berdampak pada kecepatan transmisi. Data dalam jumlah besar akan menempuh saluran yang memiliki Bandwidth kecil lebih lama dibandingkan melewati saluran yang memiliki Bandwidth yang besar. Kecepatan transmisi tersebut sangat dibutuhkan untuk aplikasi Komputer yang memerlukan jaringan terutama aplikasi real-time, seperti videoconferencing. Penggunaan Bandwidth untuk LAN bergantung pada tipe alat atau medium yang digunakan, umumnya semakin tinggi Bandwidth yang ditawarkan oleh sebuah alat atau medium, semakin tinggi pula nilai jualnya.

Istilah traffic dapat didefinisikan sebagai banyaknya informasi yang melewati suatu channel komunikasi (medium komunikasi). Semakin banyak informasi yang dikirm, maka semakin tinggi pula traffic dan dibutuhkan bandwith yang lebih besar.

Pada komputer paralel Cluster Beowulf, salah satu faktor yang menurunkan kecepatan komputasi paralel adalah traffic yang tinggi. Ketika rutin program Cluster Beowulf dijalankan, maka dibutuhkan komunikasi antar node dan master. Pada Sistem Cluster Beowulf, semua node dan master berjalan pada satu jaringan yang sama, yang berarti menggunakan bandwith yang sama. Penggunaan bandwith secara bersama-sama inilah yang menyebabkan komputasi Cluster Beowulf menjadi terhambat.

Pada penelitian ini dibuat batasan masalah yaitu daya yang terpakai tidak dihitung, proses komputasi paralel dijalankan menggunakan OpenMPI, dan pengujian dilakukan dengan menggunakan program perkalian matriks.

Pada konfigurasi jaringan bertingkat atau sub-cluster pada skripsi ini, node tidak berjalan pada jaringan yang sama. Setiap 4 node memiliki sub-cluster sendiri, dan artinya setiap sub-cluster memiliki bandwith sendiri. Sehingga bandwith yang tersedia untuk komunikasi node-master saat rutin program komputasi paralel dijalanan menjadi lebih tinggi. Diharapkan dengan penggunan konfigurasi jaringan bertingkat pada Cluster Beowulf, bandwith yang lebih besar dapat mewadahi semua traffic yang terjadi saat rutin program dijalankan, dan akhirnya diharapkan meningkatkan performa dari komputasi paralel.

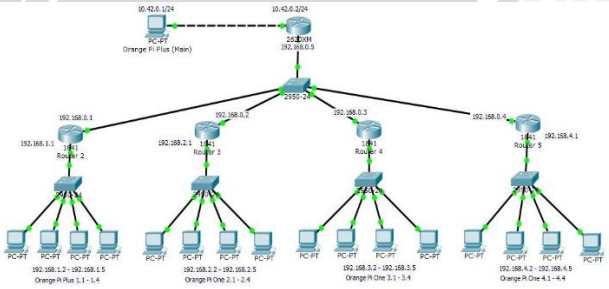
Program pengujian menggunakan algoritma “manager/worker” atau pendekatan “task parallelism”. Idenya adalah bahwa pekerjaan yang perlu dilakukan dapat dibagi oleh "manager" menjadi potongan-potongan terpisah dan potongan dapat diberikan ke masing-masing proses "worker". Dengan demikian manager mengeksekusi algoritma berbeda dari para worker, tetapi semua worker mengeksekusi algoritma yang sama. Sebagian besar implementasi MPI memungkinkan proses MPI yang akan menjalankan program yang berbeda (file executable), tetapi lebih mudah (dan dalam beberapa kasus diperlukan) untuk menggabungkan kode manager dan worker ke satu program dengan struktur seperti berikut.

Kadang-kadang pekerjaan dapat merata dibagi menjadi persis sebanyak worker yang ada, tetapi pendekatan yang lebih fleksibel adalah membuat manager untuk menjaga sekumpulan unit pekerjaan yang akan dilakukan lebih besar dibanding dengan jumlah worker, dan menetapkan kerja baru secara dinamis untuk worker setelah tugas mereka telah diselesaikan dan mengirim hasilnya kembali ke manager. Pendekatan ini, disebut penjadwalan diri (self-scheduling), bekerja dengan baik dengan adanya tugas dari berbagai ukuran dan / atau worker dari kecepatan yang bervariasi.

Teknik ini digambarkan dengan program paralel untuk mengalikan matriks dengan vektor. Program ini bukan cara yang sangat baik untuk melaksanakan operasi ini, tetapi ini menggambarkan pendekatan komputasi paralel dan cukup sederhana untuk ditampilkan secara keseluruhan. Program ini mengalikan matriks persegi dengan vektor b dan menyimpan hasilnya dalam c. Satuan kerja adalah titik produk individual dari baris dengan vektor b. Dengan demikian pada kode manager, dimulai dengan menginisialisasi. Manager kemudian mengirimkan unit awal kerja, satu baris untuk setiap worker. Tag MPI digunakan pada setiap pesan tersebut untuk mengkodekan nomor baris yang dikirim. Sejak nomor baris mulai dari 0 tapi ingin dipesan 0 sebagai tag dengan arti khusus dari "tidak ada pekerjaan lagi yang harus dilakukan", tag ditetapkan ke posisi baris +1. Ketika salah satu worker mengirim kembali produk titik, disimpan di tempat yang tepat di c dan mengirim baris lain kepada worker itu untuk kembali dikerjakan. Setelah semua baris telah ditetapkan, worker yang menyelesaikan tugas dikirim pesan "tidak ada pekerjaan lagi", ditunjukkan dengan pesan dengan tag 0.

II. PERANCANGAN DAN PEMBUATAN ALAT

Konfigurasi Sub-cluster sistem keseluruhan ditunjukkan dalam Gambar 1.



Gambar 1 Konfigurasi Sub-Cluster

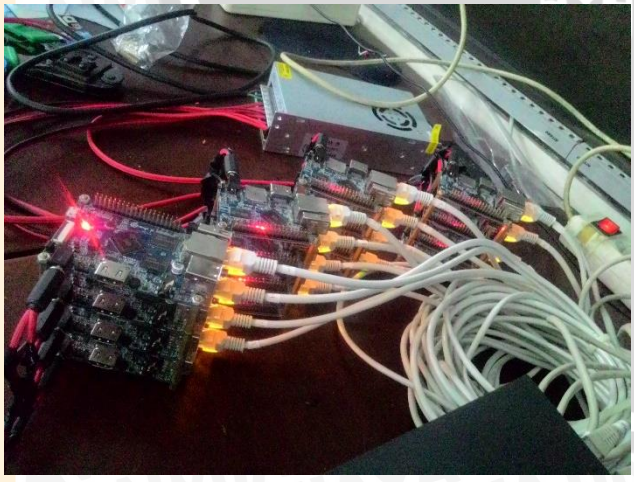
Tahapan perancangan sistem sub-cluster adalah sebagai berikut:

1. Merangkai peralatan yang digunakan.
2. Melakukan instalali sistem Operasi Lubuntu 14.04 pada setiap Orange Pi.
3. Melakukan konfigurasi IP dan Routing pada setiap router yang digunakan.

4. Memasang semua paket perangkat lunak yang digunakan (SSH, MPI)
5. Melakukan konfigurai hosts dan hostname pada setiap perangkat.
6. Pengujian PING, apakah perangkat sudah saling terhubung atau tidak.

Gambar 2 Perangkat Setelah dirakit

Gambar 3 Orange Pi Di nyalakan

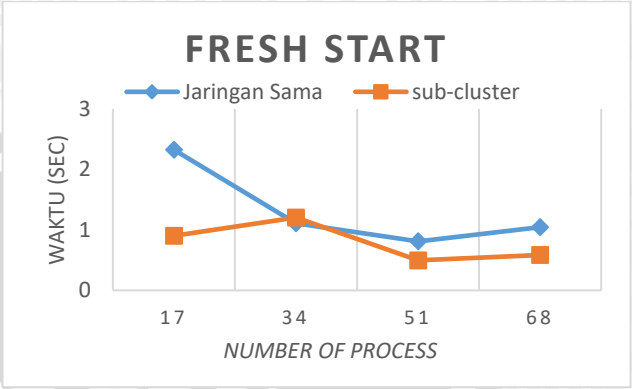


Gambar 4 Sub Cluster Siap Digunakan

Setelah perangkat telah selesai di konfigurasi, maka program MPI yang digunakan sudah siap untuk dijalankan. Binary program hasil compiling di gandakan ke semua perangkat Orange Pi yang ada. Program di jalankan melalui Main. Pengujian dilakukan dengan menajalankan program perkalian matriks. Progam dijalankan dengan 4 jumlah proses yang berbeda, yaitu 17, 34, 51, dan 68 proses. Dan juga di uji dengan berbagi ukuran yang berbeda.

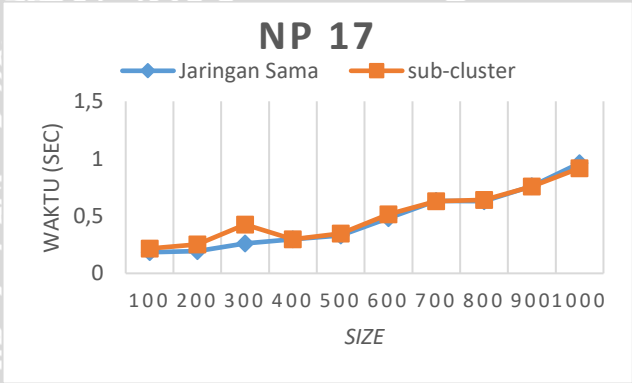
III. HASIL PENGUJIAN

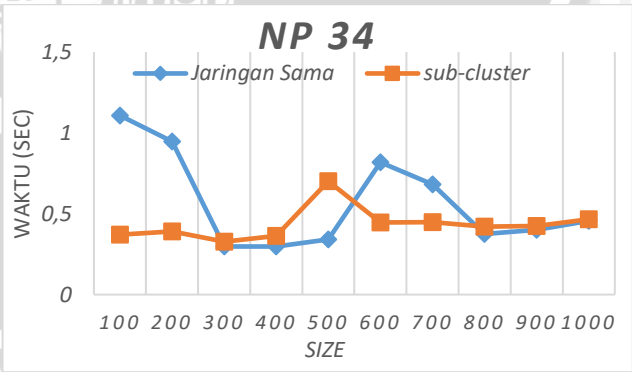
3.1 Program dijalankan dengan kondisi awal yang sama, yaitu saat perangkat baru dinyalakan (fresh start). Ini untuk menghindari adanya program yang tersimpan di memori.

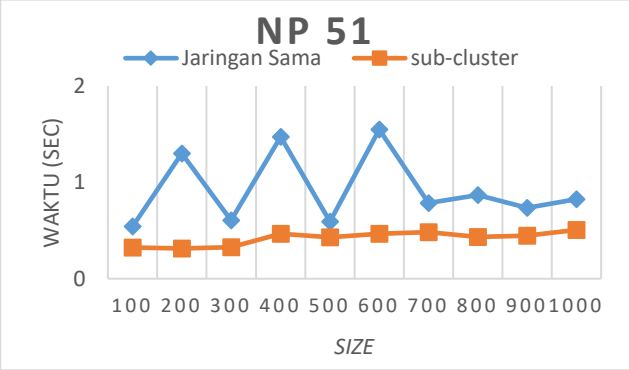


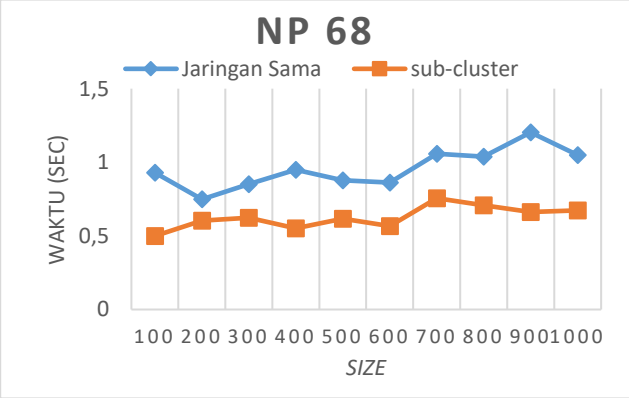
Pada pengujian ini, jaringan sub-cluster lebih cepat dalam mengeksekusi program dengan menggunakan *number of process* 17, 51 dan 68.

3.2 Program dijalankan dengan ukuran dan jumlah proses yang berbeda.









Data hasil pengujian jaringan sama

	waktu (sec)			
size	np 17	np 34	np 51	np 68
100	0,1842	1,1075	0,5420	0,9304
200	0,1934	0,9475	1,3002	0,7484
300	0,2601	0,2980	0,6046	0,8511
400	0,2978	0,2973	1,4725	0,9496
500	0,3301	0,3406	0,5945	0,8765
600	0,4813	0,8180	1,5488	0,8616
700	0,6331	0,6822	0,7848	1,0588
800	0,6277	0,3744	0,8675	1,0385
900	0,7594	0,3999	0,7366	1,2032
1000	0,9607	0,4554	0,8248	1,0496

Data hasil pengujian jaringan bertingkat (sub-cluster)

	Waktu (sec)			
size	np 17	np 34	np 51	np 68
100	0,2165	0,3707	0,3244	0,4994
200	0,2513	0,3901	0,3142	0,6042
300	0,4244	0,3273	0,3265	0,6231
400	0,2969	0,3622	0,4681	0,5525
500	0,3482	0,7010	0,4314	0,6158
600	0,5149	0,4455	0,4679	0,5662
700	0,6281	0,4471	0,4837	0,7561
800	0,6402	0,4200	0,4348	0,7086
900	0,7568	0,4242	0,4456	0,6632
1000	0,9171	0,4661	0,5072	0,6725

Pada pengujian ini, dapat dilihat bahwa hasilnya cukup bervariasi. Peningkatan kecepatan didapat pada kasus pemrosesan paralel dengan jumlah proses 34 dengan ukuran program 100, 200, 600, 700, dan pada semua ukuran program dengan jumlah proses 51 dan 68. Sedangkan pada pengujian dengan jumlah proses 17 wkatu yang dibutuhkan untuk menjalankan program hampir sama. Hal ini menunjukkan bahwa penignkatan kecepatan komputasi paralel sangat bergantung pada banyak variabel. Jumlah prosesor, kecepatan prosesor, jumlah proses yang digunakan, bandwith total pada jaringan, design jaringan, memori yang digunakan, permasalahan yang akan diselesaikan dan juga bagaimana program ditulis merupakan beberapa faktor diantaranya.

IV. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan perancangan metode sub-cluster dan pengujian dengan membandingkan dengan metode jaringan biasa, maka dapat disimpulkan

1. Metode sub-cluster merupakan salah satu metode yang bisa digunakan untuk merancang suatu komputer paralel.
2. Metode sub-cluster dapat meningkatkan kecepatan komputasi dalam menjalankan program perkalian matriks, meskipun tidak di semua jenis pengujian terlihat adanya peningkatan kecepatan.

4.2 Saran

1. Metode sub-cluster di pakai dan di uji dengan menggunakan permasalahan dan penulisan program lain yang sesuai.
2. Penggunaan NFS untuk mempermudah menjalankan program.
3. Penggunaan *batch* saat menjalankan program agar memudahkan dalam pengambilan data.

Daftar Pustaka

[1] Agus Kurniawan. 2010. *Pemrograman Paralel dengan MPI dan C*. Yogyakarta: Penerbit ANDI.

[2] Brown, Robert G.. 2004. *Engineering a Beowulf-style Compute Cluster*. New York City.

[3] Barry Wilkinson & Michael Allen. 2010. *Parallel Programing*. Yogyakarta: Penerbit ANDI.

[4] Georga S. Almasi & Allan Gottlieb. 1994. *Highly Parallel Computing*. California: The Benjamin/Cummings Publishing Company, Inc.

[5] I Putu Agus Eka Pratama. 2014. *Handbook Jaringan Komputer*. Bandung: Penerbit Informatika.

[6] William Gropp & Ewing Lusk. 2003. *Beowulf Cluster Computing with Linux*. Massachusetts: MIT Press.