

RANCANG BANGUN *CLUSTER* PC BERBASIS LINUX UNTUK KOMPUTASI PARALEL MASALAH FISIKA

Menik Ariani dan Arsali
Jurusan Fisika FMIPA Universitas Sriwijaya

ABSTRAK

Disajikan rancang bangun komputer paralel dari cluster sejumlah PC low-end. Cluster terdiri dari sekelompok interkoneksi workstation atau PC stand-alone. Paket software PVM (Parallel Virtual Machine) menggabungkan sekelompok komputer dengan bermacam arsitektur dan jaringan, sehingga membentuk sebuah mesin paralel semu yang dapat diperlakukan seperti sebuah mesin tunggal berkinerja tinggi. Pemrosesan paralel dengan cluster PC akan mempersingkat waktu komputasi atau simulasi sistem dengan banyak variabel dan timestep, yang melibatkan operasi aljabar linear pada matriks berukuran sangat besar. Kinerja komunikasinya ditunjukkan oleh hasil pengukuran bandwidth dan latensi. Kinerja komputasinya ditunjukkan melalui implementasi PVM pada operasi floating point, serta penyelesaian masalah fisika yang dalam hal ini ditunjukkan melalui penyelesaian persoalan distribusi panas dan potensial listrik. Cluster PC yang dibangun mampu memanfaatkan bandwidth sampai 88% dari bandwidth yang disediakan jaringan. Berdasarkan implementasi komputasi paralel pada sistem, cluster yang terdiri dari empat buah PC memiliki kinerja komunikasi dan komputasi cukup baik.

Kata kunci : Rancang bangun, cluster Linux PC

ABSTRACT

The design and constructing Linux PC clusters from of cheap PCs are presented. A cluster was defined as a collection of interconnected stand-alone PCs or workstations. PVM is the software package allowing us to combine a number of computers on different architectures, which connected over a network into a parallel virtual machine was can be deemed as a high performance single parallel machine. Parallel processing on PC cluster will reduce time of computation or simulation of the problems with many variables and timesteps. It involved linear algebra operation with large scales matrix. The performance of communication was indicated by measurement of bandwidth and latency. The performance of computation was indicated by PVM implementation on floating point operation and solution of physics problems as heat distribution and electric potential

The PC cluster was capable to use the bandwidth up to 88% bandwidth of the networks. Based on parallel computing implementation on the system, the cluster that constructed by four PCs possessed quite good performance in communication and computation.

Key Word : The desingn and Constructing ; Linux Pc, cluster.

1. PENDAHULUAN

1.1. Latar Belakang

Kemajuan di bidang teknologi mikroprosesor telah memungkinkan sebuah PC (*Personal Computer*) dapat menjadi salah satu *node* (bagian) dari sebuah superkomputer. Konsep meng-*cluster* PC telah menawarkan kemampuan pemrosesan paralel yang setara dengan kemampuan superkomputer, dengan bermacam keuntungan seperti: memanfaatkan PC yang sudah ketinggalan jaman oleh perkembangan prosesor tanpa harus membuangnya menjadi limbah, kemudahan dalam mengubah konfigurasi jaringan serta mengeksploitasi kekuatan komputasi dari masing-masing PC.

Salah satu piranti lunak pemrograman yang dirancang untuk memprediksi dan mengatasi masalah hubungan interkoneksi pada jaringan adalah PVM (*Parallel Virtual Machine*) yang berbasis *open source operating system*. PVM memungkinkan sejumlah komputer heterogen berbasis Linux dan terhubung jaringan Ethernet, dapat direalisasikan menjadi sebuah sistem paralel tunggal.

Sebagian besar komputer (PC) yang terdapat di Laboratorium Komputasi Jurusan Fisika FMIPA Universitas Sriwijaya, tidak dipakai selama berjam-jam dalam sehari, terutama pada malam hari. Berbagai PC tersebut, terutama pada waktu-waktu kosong (*idle time*), dapat difungsikan sebagai *node-node* dari suatu sistem paralel, yang didedikasikan untuk menyelesaikan masalah komputasi fisika. Dengan membagi masalah menjadi beberapa bagian, yang masing-masing dikerjakan oleh tiap PC (*node*), anggota *cluster*, secara serempak maka diharapkan akan mampu mereduksi banyak waktu komputasi dalam penyelesaian masalah komputasi yang rumit dan/atau besar.

1.2. Perkembangan Cluster PC

Kumpulan PC paralel yang pertama dibangun pada tahun 1994 oleh Goddard Space Center, NASA untuk memenuhi kebutuhan Badan Antariksa AS tersebut. *Cluster* ini dihasilkan dengan cara menghubungkan 16 buah PC (Intel 80486DX4) berbasis Linux melalui jaringan Ethernet 10 Mbps. Kemampuannya mencapai 1Gigaflops (satu milyar operasi *floating-*

point perdetik) (Zaritski, 2002). Suatu kemampuan yang sanggup menandingi kecepatan Cray X-MP/4 (http://en.wikipedia.org/wiki/Cray_X-MP), sebuah superkomputer yang dikembangkan pada tahun 1983-1984. Beberapa penelitian lain yang relevan yaitu:

1. *Hyglac* di *Caltech Institute* mengintegrasikan 16 prosesor Pentium Pro, menghasilkan kecepatan 1,2 Gigaflops, dengan biaya sekitar \$50.000 (Hargrove,2001).
2. *Stone SouperComputer* (Oak Ridge National Laboratory, th 1997) dirancang untuk aplikasi *finite-element groundwater simulation, multivariate geographic clustering* dan *continental vegetation modeling* (Hoffman dan Hargrove, 2000).

2. DASAR TEORI

2.1. Konsep Pemrosesan Paralel

Pemrosesan paralel adalah metoda penyelesaian sebuah masalah tunggal dengan cara membaginya menjadi beberapa bagian. Tiap bagian diproses secara serempak oleh sekelompok pekerja (dalam hal ini prosesor). Jumlah serta kerjasama antar prosesor akan mempengaruhi waktu penyelesaian masalah.

2.2. Klasifikasi Komputer Paralel

Klasifikasi arsitektur komputer *sekuensial* dan *paralel* diperkenalkan pada tahun 1966 oleh Michael J. Flynn, yaitu (Quinn, 1994): .

1. *SISD (Single Instruction Stream Single Data Stream)*
Komputer yang terdiri dari prosesor tunggal dan sebuah unit pengendali yang mengendalikan seluruh unit fungsional.
2. *SIMD (Single Instruction Stream Multiple Data Stream)*
Komputer yang memiliki lebih dari satu unit pemroses dengan sebuah unit pengendali. Seluruh unit pemroses menerima dan menjalankan sebuah instruksi yang berasal dari unit pengendali. Operasi dilakukan terhadap himpunan data yang berasal dari aliran data yang berbeda.
3. *MISD (Multiple Instruction Stream Single Data Stream)*
Komputer yang memiliki sejumlah unit pemroses, yang masing-masing menerima dan mengoperasikan instruksi yang berbeda terhadap aliran data yang sama. Keluaran dari satu pemroses menjadi masukan bagi pemroses berikutnya.

4. MIMD (*Multiple Instruction Stream Multiple Data Stream*)

Arsitektur MIMD terdiri dari beberapa prosesor independen yang memiliki unit pengendali sendiri. Tiap prosesor dapat bekerjasama menyelesaikan masalah tunggal.

2.3. Kinerja Sistem Paralel

Beberapa hal yang mempengaruhi kinerja sistem paralel adalah *kinerja prosesor* (kecepatan prosesor, akses RAM, bus dll), kinerja jaringan (*bandwidth*, latensi), dan kinerja komputasi (*speedup*, rasio C/C) (Tannenbaum, 1998).

Bandwidth adalah ukuran kecepatan transfer data, khusus digunakan untuk mengukur kemampuan komunikasi dari multikomputer dan multiprosesor. *Bandwidth* biasanya dinyatakan dalam megabyte per second (MBps) atau megabit per second (Mbps). Latensi berkaitan dengan waktu tetap yang terpakai saat melayani pengiriman/penerimaan sebuah pesan data. *Speedup* adalah rasio dari waktu eksekusi program serial terbaik terhadap waktu eksekusi program saat diimplementasikan pada sebuah pemroses paralel (Lewis, 1993). Rasio C/C (*computation-to-communication*)

merupakan perbandingan antara waktu yang digunakan untuk kalkulasi (komputasi) dengan waktu yang dibutuhkan untuk komunikasi (pengirim/penerimaan pesan). Rasio ini bergantung pada kecepatan relatif dari prosesor dan media komunikasi yang digunakan.

2.4. Sistem Pemrograman Paralel dengan *Parallel Virtual Machine (PVM)*

Sistem operasi yang dipakai sebagai *platform* adalah Linux, yang mampu bekerja secara *multitasking* dan *multiuser*. *Multitasking* artinya dapat menjalankan beberapa aplikasi secara bersamaan. *Multiuser* artinya mendukung penggunaan aplikasi atau komputer untuk melayani beberapa user sekaligus. Linux merupakan tiruan (clone) dari UNIX yang dirancang untuk dijalankan pada PC, sehingga memiliki kompatibilitas tinggi terhadap UNIX. Salah satu kelebihan Linux adalah sifatnya yang *free-software*, artinya dapat dimiliki dan disebarluaskan tanpa membayar lisensi pada pembuatnya. Paket *software PVM (Parallel Virtual Machine)* melaksanakan pembuatan/penghapusan *task*, pertukaran data antar *task* serta sinkronisasi *task* (Geist, et al., 1994).

Aplikasi pemrograman paralel yang dilakukan dengan PVM yaitu model *master-slave* atau *host-node* [Geist, et al., 1994]. Program kontrol dinamakan *master (host)*. Gunanya antara lain untuk membuat proses PVIM yang baru, menginisialisasi, dan mengatur beban kerja *slave (node)*. Program *slave (node)* tugasnya melakukan komputasi jenis tertentu.

2.5. Komputasi Paralel pada Masalah Distribusi Panas

Persamaan distribusi panas untuk bahan berdimensi satu dapat dituliskan sebagai (Poulidakos, 1994):

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} ; \quad \alpha = \frac{k}{c_p \rho} \quad (2.1)$$

Variabel T menyatakan temperatur bahan pada posisi (x,t) , dimana $0 \leq x \leq L$. Tetapan difusivitas α adalah ukuran laju difusi panas pada bahan, dengan k = konduktivitas panas, ρ = massa jenis bahan dan c_p = panas jenis bahan. Syarat awal dan syarat batas yang diberikan dianggap berjenis *Dirichlet* yaitu:

$$T(0,t) = T_a ; \quad T(L,t) = T_b ; \quad T(x,0) = T_0(x) , \\ 0 \leq x \leq L$$

Persamaan (2.1) dapat diselesaikan dengan metoda beda hingga eksplisit FTCS

(*forward in time central in space*), yaitu (Poulidakos, 1994):

$$T_i^{P+1} = T_i^P + \frac{\alpha \Delta t}{\Delta x^2} (T_{i-1}^P - 2T_i^P + T_{i+1}^P) \quad (2.2)$$

Indeks i menyatakan *node*, sedangkan indeks P menyatakan *timestep*. Jika $\Delta t = t_i - t_{i-1}$ dan $\Delta x = x_i - x_{i-1}$ maka waktu komputasi total = $\Delta t \cdot \text{timestep}$ dan $x = \Delta x \cdot \text{nodes}$, dimana $i = 0$ ($\text{nodes}-1$). Syarat kestabilan yang harus dipenuhi adalah $\frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2}$ (Powell, 2002).

2.6. Komputasi Paralel pada Masalah Potensial Listrik

Potensial $\phi(x,y)$ akibat adanya rapat muatan $\rho(x,y)$ pada bidang persegi panjang di daerah $0 \leq x \leq L_x$ dan $0 \leq y \leq L_y$ dapat diselesaikan melalui persamaan (Farlow, 1982):

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -\rho(x,y) \quad (2.3)$$

Syarat awal dan syarat batas yang diberikan dianggap berjenis *Dirichlet* yaitu nilai ϕ sepanjang tepi persegi panjang diketahui sebesar:

$$\phi(0,y) = \phi_0, \phi(L_x,y) = \phi_0, \phi(x,0) = \phi_0, \phi(x,L_y) = \phi_0 ;$$

Ungkapan beda hingga (*finite difference*) dari persamaan (2.3) adalah:

$$\frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h_x^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{h_y^2} = -\rho_{i,j} \quad (2.4)$$

Untuk memenuhi syarat batas, di batas persegi panjang akan terbentuk persamaan:

$$-4\phi_{1,j} + \phi_{2,j} + \phi_{1,j-1} + \phi_{1,j+1} = -h^2 \rho_{1,j} - \phi_{0,j} \quad (2.5)$$

$$-4\phi_{N_x-1,j} + \phi_{N_x-2,j} + \phi_{N_x-1,j-1} + \phi_{N_x-1,j+1} = -h^2 \rho_{N_x-1,j} - \phi_{N_x,j} \quad (2.6)$$

$$-4\phi_{i,1} + \phi_{i+1,1} + \phi_{i-1,1} + \phi_{i,2} = -h^2 \rho_{i,1} - \phi_{i,0} \quad (2.7)$$

$$-4\phi_{i,N_y-1} + \phi_{i+1,N_y-1} + \phi_{i-1,N_y-1} + \phi_{i,N_y-2} = -h^2 \rho_{i,N_y-1} - \phi_{i,N_y} \quad (2.8)$$

Persamaan-persamaan dapat disederhanakan ke bentuk operasi matriks yang selanjutnya dapat diproses secara paralel.

3. IMPLEMENTASI PVM PADA KOMPUTASI FISIKA

3.1. Instalasi Sistem PVM

Dalam penelitian ini *cluster* dibangun oleh 4 buah PC (Intel Pentium IV 2,0 GHz, RAM 128 MB, HD 6 GB), terdiri dari 1 *node master* dan 3 *node slave*, yang terhubung melalui jaringan Ethernet. Sistem operasi yang dipakai sebagai *platform* adalah Linux, lengkap dengan kompilator bahasa

pemrograman C/C++, Fortran versi GNU dari *Free Software Foundation*, paket plot data (DISLIN, GNUPlot) dan pembuat dokumen. Pengesetan dilakukan dengan memodifikasi file *.bashrc* milik *user* pada setiap mesin anggota *virtual machine*. Untuk mempersiapkan mesin-mesin yang akan membentuk *virtual machine*, maka pada file */etc/host.equiv* di setiap mesin diisikan nama-nama *host*.

3.2. Algoritma untuk pengukuran Komunikasi Sistem

Algoritma pengukuran kecepatan pengemasan data, *bandwith* dan *latensi* jaringan dirancang melalui batasan dan aturan:

- *Master* membangkitkan suatu data berukuran mulai dari 10 - 10 Mbyte
- Tiap data dikirimkan ke *node (slave)* yang ada serta diulangi sebanyak 20 kali.
- *Slave* menerima data dari *master* yang selanjutnya dikembalikan lagi ke *master*.
- Menghitung rata-rata waktu pengemasan, *bandwith*, dan *latensi*

3.3. Algoritma Paralel pada Model Difusi Panas

Program paralel *master-slave* untuk masalah difusi panas dirancang berdasarkan algoritma sebagai berikut:

1. *Master* membagi panjang kawat menjadi sejumlah *node*, kemudian *men-spawn* 5 buah *slave* yang akan dikerjakan oleh 4 buah PC (*penguin1*, *penguin2*, *penguin3* dan *penguin4*).
2. Pada setiap *timestep*, proses *slave* melakukan pertukaran informasi nilai batas dengan *node* tetangga, kemudian mengirimkan nilai temperatur baru yang diperoleh (dengan metoda beda hingga eksplisit) kepada *master*.
3. *Master* mengumpulkan data nilai T yang dikirim oleh semua proses *slave*.
4. *Master* menghentikan proses jika seluruh *slave* telah selesai melakukan komputasi.

3.4. Algoritma Paralel pada Masalah Potensial Listrik

Masalah yang ditinjau adalah potensial dalam ruang 2-dimensi $\phi(x, y)$ akibat adanya rapat muatan $\rho(x, y)$ pada bidang

segi empat dengan $0 \leq x \leq L_x$ dan $0 \leq y \leq L_y$. Persamaan (2.5) sampai (2.8).

dapat disederhanakan ke dalam bentuk operasi matriks jika diperkenalkan matriks kolom v dengan panjang $N_x-1 \times N_y-1$ yang berisi gabungan semua baris dan matriks $\phi_{i,j}$

oleh kaitan $v_k = \phi_{i,j}$ dengan $k = (i-1)(N_y - 1) + j$ serta kebalikannya

$$\text{yaitu } i = \text{int}\left(\frac{k-1}{N_y-1}\right) + 1, \quad j = [(k-1) \bmod (N_y-1)] + 1.$$

Dengan indeks k maka perangkat persamaan menjadi

$$-4v_1 + 1v_2 + 1v_{N_y} = -h^2 \rho_1 - \phi_{0,1} - \phi_{1,0} \quad ;$$

$$1v_1 - 4v_2 + 1v_3 + 1v_{N_y+1} = -h^2 \rho_2 - \phi_{0,2} \dots$$

$$v_{k-N_y+1} + v_{k+N_y-1} - 4v_k + 1v_{k+N_y} + v_{k+N_y+1} = -h^2 \rho_k$$

Bentuk tersebut adalah bentuk matriks $Av = B$ dengan A matriks pentadiagonal. Paralelisasi dilakukan dengan cara pembagian baris matriks A oleh *master* ke sejumlah *slave* (PC anggota *cluster*). Analogi dengan matriks $Ax = b$, maka penyelesaiannya: (contohnya dengan 6 variabel, dan PC anggota cluster PE1 dan PE2)

$$\begin{matrix}
 x_1^m = \frac{b_1 - a_{12}x_2^m - \dots - a_{16}x_6^m}{a_{11}} \\
 x_2^m = \frac{b_2 - a_{21}x_1^m - \dots - a_{26}x_6^m}{a_{22}} \\
 \vdots \\
 x_6^m = \frac{b_6 - a_{61}x_1^m - \dots - a_{66}x_6^m}{a_{66}}
 \end{matrix}
 \quad
 \begin{matrix}
 A \\
 \begin{pmatrix}
 a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\
 a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\
 a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\
 a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\
 a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\
 a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66}
 \end{pmatrix}
 \end{matrix}
 \quad
 \begin{matrix}
 X \\
 \begin{pmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6
 \end{pmatrix}
 \end{matrix}
 \quad
 =
 \quad
 \begin{matrix}
 b \\
 \begin{pmatrix}
 b_1 \\
 b_2 \\
 b_3 \\
 b_4 \\
 b_5 \\
 b_6
 \end{pmatrix}
 \end{matrix}
 \quad
 \begin{matrix}
 PE_1 \\
 \\
 \\
 \\
 PE_2
 \end{matrix}$$

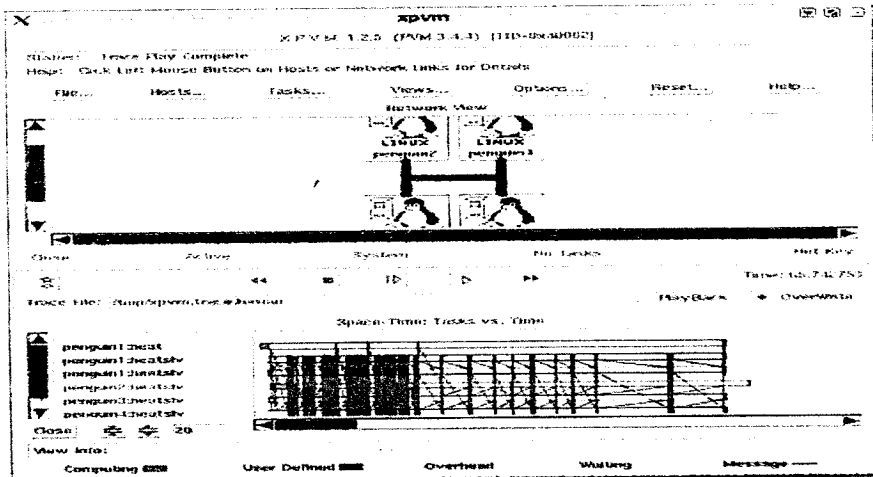
- Partisi baris A dan B ke sejumlah prosesor.
- Setiap kali iterasi, nilai x terbaru disimpan pada tiap prosesor.

4. KINERJA KOMUNIKASI DAN KOMPUTASI CLUSTER PC

4.1. Kinerja Komunikasi Jaringan

Latensi 41.02 μ s berkaitan dengan waktu tetap yang terpakai (terbuang ke dalam perangkat keras jaringan), untuk setiap kali pengiriman data. Pengiriman data dalam

paket-paket kecil ternyata memakan waktu yang sangat besar akibat adanya dominasi faktor latensi. Untuk pengiriman data sebesar 10^6 Byte, sistem yang dibangun mampu memanfaatkan 88 % dari bandwidth yang disediakan Ethernet.



Gambar 4.1. Tampilan XPVM program paralel distribusi panas

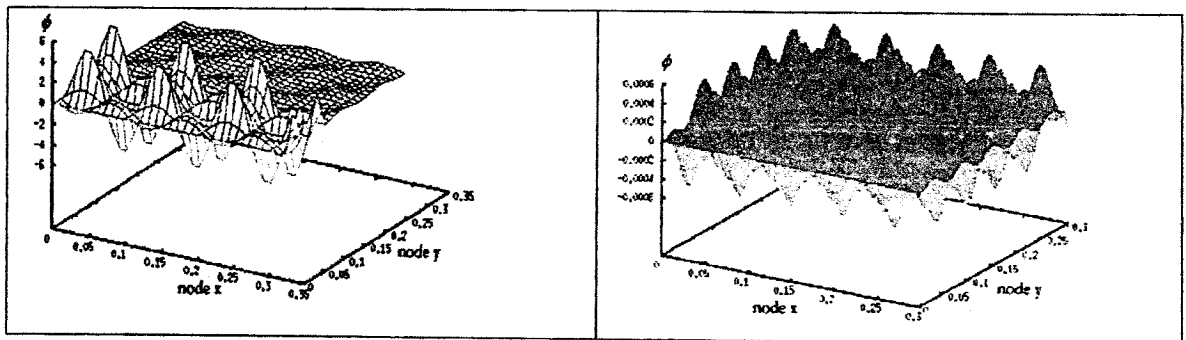
4.2. Komputasi Paralel Masalah Distribusi Panas Bahan 1-D

Gambar (4.1) mengilustrasikan proses eksekusi dengan 4 keadaan Δt . Pada satu keadaan, di setiap *timestep*, *slave-slave* (penguin1, penguin2, penguin3, penguin4, saling bertukar informasi nilai T di setiap

batas *node*, meng-*update* sesuai pendekatan *explicit finite difference*, lalu mengirim hasilnya ke *master* (penguin1)

4.3. Komputasi Paralel Masalah Potensial Listrik

Distribusi potensial listrik pada plat bujur sangkar digambarkan sebagai berikut:



Gambar 4.2. Distribusi potensial listrik pada plat bujur sangkar

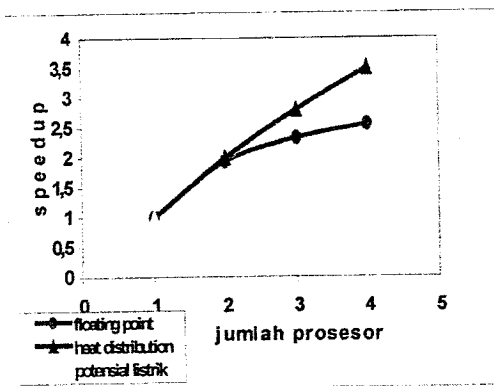
Gambar 4.2 mengilustrasikan distribusi potensial pada plat bujur sangkar yang berukuran sama, namun dengan pembagian interval x (node x) dan interval y (node y) yang berbeda. Distribusi nilai ϕ pada plat berukuran 50×50 diperoleh dari hasil perkalian matriks $Av=b$ yang menghasilkan 50×50 nilai ϕ . Demikian juga dengan plat yang berukuran 300×300 . Interval nilai ϕ

maks dan ϕ min pada ukuran 50×50 yaitu $[-6,6]$.

Semakin besar ukuran matriks, interval nilai ϕ maksimum dan ϕ minimum semakin kecil, mendekati nilai syarat batasnya yaitu 0. Interval ϕ maksimum dan ϕ minimum pada ukuran 300×300 berada pada nilai $[-6.10^{-4}, 6.10^{-4}]$.

4.4. Speedup

Speedup yang diperoleh berdasarkan hasil komputasi paralel pada cluster PC dengan *hostname penguin1, penguin2, penguin3, penguin4*



Gbr.4.3. Speedup komputasi paralel

speedup yang dihasilkan berdasarkan running program paralel operasi *floating point*, distribusi panas dan potensial listrik. Pada operasi floating point penjumlahan, kenaikan speedup sangat baik ($S_4 = 3,5$) sedangkan pada distribusi panas sudah menunjukkan kenaikan yang berarti ($S_4 = 2,6$). Hal ini disebabkan pemakaian metode *explicit finite difference* yang mengakibatkan tingginya frekuensi pertukaran (komunikasi) data antar slave (PC). Hasil program paralel potensial listrik belum menunjukkan kenaikan yang

berarti ($S_4 = 1,2$). Hal ini disebabkan besarnya waktu yang diperlukan untuk inialisasi data (dalam hal ini menentukan elemen matriks) oleh master.

5. KESIMPULAN

1. Telah berhasil dilakukan rancang bangun sistem pemrosesan paralel dari sejumlah PC berbasis Linux yang terhubung jaringan Ethernet, termasuk implementasinya untuk komputasi masalah distribusi panas dan potensial listrik.
2. Speedup pada masalah distribusi panas sudah menunjukkan kenaikan yang berarti, namun tidak pada masalah potensial listrik belum karena tingginya waktu komunikasi antar prosesor.

6. DAFTAR PUSTAKA

- Farlow, S.J., 1982. *Partial Differential and Equation for Scientists and Engineers*. John Wiley & Sons.
- Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V., 1994, *PVM: Parallel Virtual Machine -*

- A Users' Guide and Tutorial for Networking Parallel Computing*, The MIT Press Cambridge, Massachusetts, London, England.
- Hargrove, W. W., 2001, *The Do-It-Yourself Supercomputer*, Scientific American Vol. 265
- Hoffman, F. M., Hargrove, W. W., 2000, *High Performance Computing An Introduction to Parallel Programming With Beowulf*, Open Source Developers Journal, Winter 2000.
- Lewis, T. G., 1993, *Foundations of Parallel Programming: A Machine-Independent Approach*, IEEE Computer Society Press, California, USA, hal. 1 - 15.
- Poulikakos, D., 1994, *Conduction Heat Transfer*, Prentice-Hall International, Inc.,
- Powell, A., 2002, *Finite Difference Solution of the Heat Equation*, Massachusetts Institute Technology, Cambridge, http://web.mit.edu/22.00j/www/lec22_00_16.pdf, 3 Maret 2003
- Quinn, M.J., 1994, *Parallel Computing: Theory and Practice*, McGraw-Hill, Inc.,
- Tannenbaum, A.S., 1998. *Structured Computer Organization*. Prentice Hall
- Zaritski, R. M. , 2002, *Building an Inexpensive, Powerful Parallel Machine and Using It for Numerical Simulations*, Linux Journal, June 15 2002
http://en.wikipedia.org/wiki/Cray_X-MP, diakses 2 Agustus 2005