# Perbandingan Kinerja Cluster OpenMosix dengan Disk dan Tanpa Disk

Dewi Frestyani Maya Putri
Singgih Jatmiko
A Benny Mutiara Q N

***Abstrak***

*Efisiensi penggunaan perangkat keras dalam cluster komputer adalah dengan cara menggunakan perangkat keras yang seminimal mungkin tanpa harus mengorbankan kinerja sistem itu sendiri. Pada penelitian ini dilakukan pengujian penggunaan hardware minimal pada sistem cluster komputer openmosix dengan menggunakan program aplikasi untuk merender citra. Pengujian dilakukan pada sistem yang menggunakan disk dan tanpa menggunakan disk dan membandingkan kinerja keduanya. Ternyata setelah dilakukan pengujian ternyata dapat dikatakan bahwa tidak terdapat perbedaan kinerja antara sistem yang menggunakan disk dan yang tidak menggunakan disk. Oleh karena itu untuk menghemat penggunaan hardware, sistem diskless dapat diterapkan tanpa harus mengurangi kinerja sistem.*

*Kata kunci : Diskless, Clustering, OpenMosix, POV-ray, Rendering*

## 1. Pendahuluan

*Cluster* yaitu penyatuan sekelompok data yang mempunyai korelasi atau karakteristik sejenis, berarti kita menyatukan PC-PC atau server (atau apapun *device* yang ber-OS) ke dalam satu kesatuan komputasi. Cluster pada dasarnya juga menggunakan jaringan, tapi yang menjadi ciri khususnya adalah bagaimana konfigurasi ini digunakan untuk menyelesaikan problem.

Ide awal clustering adalah tantangan menciptakan kecepatan komputasi setara mainframe, namun dengan biaya yang rendah. Kita mungkin sedikit banyak tahu, bahwa harga mainframe, misal Cray, luar biasa mahalnya. Di era 70-an dan 80-an dimana mainframe (IBM AS/400 masuk kategori mainframe) cukup terkenal, hanya perusahaan-perusahaan bermodal besar yang sanggup membelinya.

Karakteristik utama *clustering*, yaitu bagaimana suatu *task* ditangani secara *bersama-sama*. Kata *bersama-sama* di sini yang membedakan cluster dengan mekanisme jaringan biasa yang menjalankan program client server. Memang seringkali rancu dengan *Distributed Computing*, tapi mungkin bisa diperjelas lagi bahwa clustering difokuskan pada mekanisme dari OS dan hardware untuk menyatukan sumber daya setiap node (bisa CPU, memori, disk, dsb). Dua tipe Cluster yang dominan yaitu :

1. *High Perfomance Computing (HPC).* Secara umum, tipe cluster HPC ditujukan pada bagaimana suatu proses komputasi diakselerasi, dengan demikian *task* bisa diselesaikan lebih cepat contoh clustering jenis ini adalah openMosix.
2. *High Availability (HA).* Secara umum, tipe cluster ini ditujukan agar program yang dijalankan di atasnya bisa terus berjalan, sekalipun salah satu node hang atau down. Contoh yang paling mudah adalah *web server* Apache yang diset dengan suatu redirector, sehingga jika salah satu server down, server lain bisa mengambil alih.

OpenMosix merupakan tools untuk kernel yang termasuk jenis keluarga unix, seperti linux. Terdiri dari algoritma-algoritma penggunaan sumber daya bersama yang dapat disesuaikan. Open mosix memperbolehkan multiple Uniprocessors (UP) dan Simetric Multiprocessors (SMP) menjalankan kernel yang sama untuk bekerja dalam pekerjaan yang mirip. Algoritma-algoritma penggunaan sumberdaya bersama dirancang untuk merespon langsung bermacam-macam pemakaian sumberdaya pada setiap node.

Jaringan cluster bisa terdiri dari sejumlah besar komputer, oleh karena itu, penghematan penggunaan hardware tentunya akan dapat menekan biaya hardware tanpa harus mengorbankan kinerjanya. Pada

penelitian ini akan diuji pengurangan hardware untuk tiap worktation yaitu tanpa harddisk. Untuk melihat kemampuan openMosix dalam menyelesaikan sebuah aplikasi perender citra, maka penulis mencoba melihat efektifitas POV-Ray (software perender) sebagai aplikasi saat berjalan di cluster OpenMosix dan kemudian membandingkan dengan cluster diskless openMosix yang memiliki keunggulan efisiensi sumberdaya tanpa harddisk.

## 2. Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah melakukan percobaan dengan membandingkan kinerja jaringan cluster dengan dan tanpa hardisk. Kemudian melakukan analisa berdasarkan hasil uji dan studi literatur yang ada.

### 2.1 Jaringan Komputer Diskless

**a. Konsep**

Pengertian diskless adalah mengijinkan client yang tidak dilengkapi dengan media penyimpanan seperti harddisk, disket, CDROM dan sebagainya untuk dapat mengaktifkan sistem operasi. Proses diskless akan membantu komputer client untuk dapat mengaktifkan sistem operasi tersebut dengan mengekseskusi file kernel di sisi komputer client. Setelah proses diskless selesai, dilanjutkan dengan akses melalui jaringan untuk mengeksekusi X-Server di sisi komputer client, sehingga komputer client dapat mengakses aplikasi diskless.

Proses tersebut memungkinkan kmputer lama seperti komputer 486 yang mempunyai RAM 8 MB menggunakan diskless dapat menjalankan kernel dan mengeksekusi X-Server. Setelah proses eksekusi berhasil, maka proses dialihkan ke XDM pada komputer client dengan konfigurasi yang tinggi. Proses yang telah diarahkan tersebut seolah-olah berjalan di komputer client dengan kecepatan yang tinggi. Sebenarnya, proses tersebut terjadi di server sedangkan outputnya di client.

Booting melalui jaringan merupakan konsep lama, ide dasarnya adalah komputer client dengan kode booting seperti BOOTP (*boot protocol*) atau DHCP (*Dynamic Host Cvonfiguration Protocol*) dalam memory non-volatile (ROM) chips mendapatkan sistem file root server dalam suatu jaringan ketika komputer client tidak dilengkapi dengan media penyimpanan, misalnya harddisk. Ada beberapa hal yang harus dipenuhi komputer-komputer yang akan melakukan pertukaran data yang cukup kompleks, yaitu :

- Kartu jaringan
- IP address
- Image Kernel
- File system

Untuk mengenali komputer-komputer dalam jaringan tersebut satu dengan yang lainnya, terdapat informasi yang unik. Informasi unik tersebut didapat dari kartu jaringan client. Nomor unik tersebut terdiri atas 48 bit yang terdiri atas 6 blok bilangan hexa yang dipisahkan dengan tanda titik dua. Pada masing-masing blok terdiri atas dua digit, misalnya 00:a0:24:2e:ba:be. Nomor unik tersebut disebut sebagai hardware address atau MAC. Untuk mencapai bentuk diskless komputer client dianggap tidak mempunyai harddisk. Dengan demikian untuk mendapatkan file sistem server, komputer menggunakan nomor unik (MAC) ini. Protocol yang digunakan adalah BOOTP dan DHCP. Dengan demikian komputer client harus terdaftar dalam suatu database. Ketika proses protocol tersebut dijalankan untuk mendapatkan IP address dan informasi lainnya, komputer client harus men-download kernel yang terletak di server. TFTP (*Trivial File Transfer Protoco*l) adalah protokol yang digunakan untuk men-download kernel.

Ketika kernel berhasil di-download, kernel kemudian melakukan inisialisasi perangkat keras komputer client yang dimiliki. Akhirnya, komputer client membutuhkan file sistem root. Untuk itu protokol NFS (*Network File System*) diperlukan. Dengan NFS komputer client dapat menjalankan sistem server melalui jaringan. Sebenarnya, proses tersebut berjalan di server namun outputnya di komputer client.

Secara sederhana, komputer client hanya menjalankan sistem operasi yang telah di-download dengan bantuan protocol TFTP sedangkan file sistem server tetap di server namun outputnya di client.

Beberapa kelebihan dalam menerapkan jaringan diskless adalah sebagai berikut :

a. Biaya, linux adalah program yang open source, kita dapat membeli dengan harga yang jauh lebih murah serta komputer model lama dapat digunakan dengan hasil yang sangat memuaskan. Serta dapat meminimalisasi perangkat seperti CDROM dan floppy.
b. Pemeliharaan, jauh lebih mudah dibandingkan membangun jarigan biasa. Disini cukup dilakukan pada satu komputer yaitu komputer server.
c. Performance, jika pembaca menggunakan koputer server Pentium III, maka komputer client seolah-olah akan menjalankan sistem dengan komputer Pentium III.
d. Backup, Semua backup data terjadi di server.
e. Keamanan, kita dapat mengatur administrasi login yaitu tidak boleh ada dua user login ke server.
f. Upgrading hardware dan program menjadi lebih mudah.

Beberapa kelebihan dalam menerapkan jaringan diskless adalah sebagai berikut :

a. Butuh waktu yang cukup lama untuk mempelajari diskless menggunakan Linux bagi mereka yang belum pernah atau baru mengenal Linux.
b. Kinerja client akan menurun apabila jika ada penambahan komputer client.
c. Butuh resource server yang cukup handal.

**b. Pengoperasian Jaringan Diskless**

Berikut ini dijelaskan teori-teori operasi yang dilakukan komputer client pada saat boot sampai mendapatkan sistem Linux dari server dalam jaringan diskless :

1. Saat komputer terminal dinyalakan, komputer client akan mencari “kernel” pada disket boot atau eprom pada lan card (jika anda memasang boot eprom pada LAN Card anda), kemudian melakukan proses booting.
2. Kode booting yang tersimpan dalam “kernel” tersebut segera mencari DHCP server ke jaringan local untuk mendapatkan alamat IP.
3. DHCP Server pada server LTSP menanggapi request tersebut dan segera memberikan alamat IP kepada komputer terminal yang request tersebut.
4. DHCP Server membaca konfigurasi di /etc/dhcpd.conf, dan mencocokan MAC Address dari komputer terminal yang request, kemudian melakukan proses:
    - Memberikan alamat IP komputer terminal tersebut (‘ip=’)
    - Netmask setting dari jaringan local (‘nm=’)
    - Direktori dari file booting (‘hd=’)
    - Nama dari kernel yang dikirim (‘bf=’)
5. Kode booting pada komputer workstation akan menerima informasi dari DHCP server dan mengkonfigurasi tcp/ip interface dari lan card (Ethernet) sesuai dengan parameter yang telah diberikan.
6. Kode booting kemudian mengirimkan permintaan tftp ke server untuk mulai mengambil kernel dari server.
7. Setelah kernel diambil oleh workstation, kode booting kemudian mulai menjalankan kode kernel yang diambil dari server melalui tftp tersebut.
8. Kernel kemudian dijalankan untuk melakukan inisialisasi system dan mengaktifkan driver dari perangkat keras yang terpasang pada komputer terminal.
9. Kernel akan memberikan semua permintaan pengiriman DHCP pada jaringan. Kode booting tidak memberikan informasi pada kernel, tetapi kernel meminta informasi pada dirinya sendiri.
10. Server akan memberikan tanggapan dengan mengirimkan paket informasi lainnya. Informasi yang dibutuhkan kernel untuk dapat melanjutkan proses.
    Bagian dari informasi yang diberikan adalah :

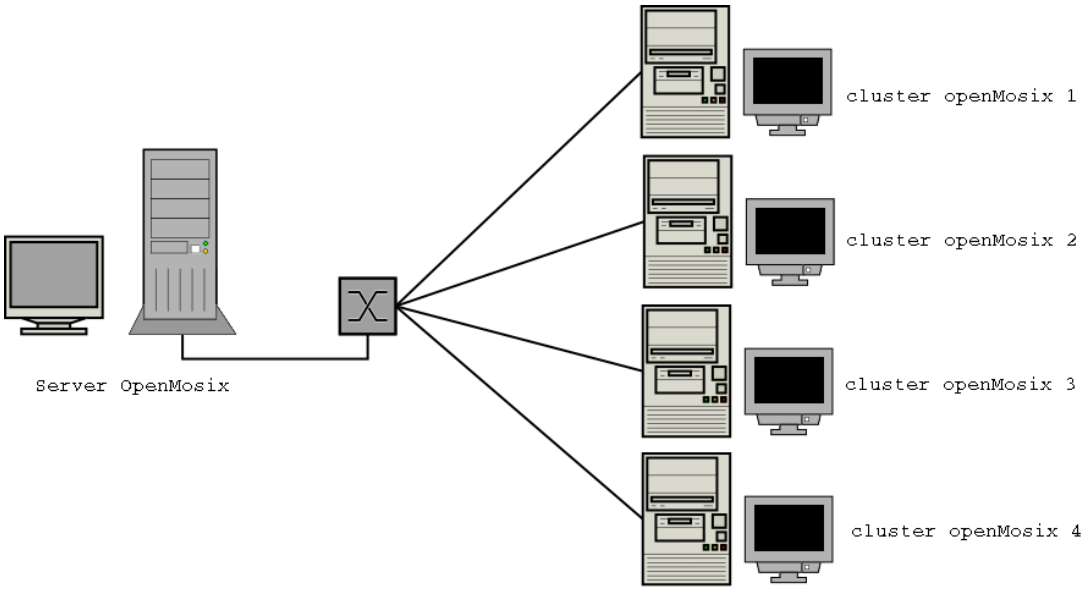
- Alamat IP dikirimkan pada komputer terminal (‘ip=’)
- Netmask setting dari jaringan local (‘nm=’)
- Mengkaitkan direktori root melalui nfs (‘rp=’)
- Gateway (‘gw=’)
- DNS Server (‘ds=’)

Hostname komputer terminal (nama hostname dimasukan pada bagian pertama dalam bootptab), dan root filesystem akan di-mount melalui NFS dengan mode read only (hanya dapat dibaca). Hal ini untuk mencegah terjadinya modifikasi terhadap root file-system yang dipakai secara bersama oleh banyak komputer terminal.

11. Kemudian dari kernel akan membaca init.
12. Init akan membaca file /etc/inttab dan memulai setting up environment.
13. salah satu baris penting dalam file inittab adalah perintah rc.local, skrip rc.local tersebut akan dijalankan di komputer terminal (bagian dari ‘sysinit’).
14. Script rc.local kemudian mengaktifkan ramdisk sebesar 4 MB untuk semua kebutuhan menulis dan memodifikasi setiap saat.
15. Ramdisk ini kemudian dikaitkan dalam directory /tmp. Tujuannya adalah untuk memberikan ruang untuk menulis bagi beberapa program yang dijalankan dan beberapa simbolik links juga akan dibuat. Sebagai contoh, jika komputer terminal berjalan, komputer terminal akan mencoba untuk memodifikasi permissions dalam /dev/tty0 dari bagian device. Jika bagian device ada dalam directory /dev, permissions tidak bisa memodifikasi karena system file root adalah hanya bisa dibaca (read only). Jadi, dibutuhkan simbolik links untuk semua file dan membuat actual files /nodes dalam directory /tmp (berisi file sementara dan dapat ditulisi).
16. Kemudian, proses dilanjutkan dengan me-mount system file /proc (sytem file semu yang dapat ditulis diatas memori). Direktory /proc ini berisi file-file informasi system.
17. Mengaktifkan konfigurasi loopback network interface.
18. Beberapa directory dan file kemudian akan terbentuk dalam directory /tmp yang akan digunakan oleh beberapa program yang dijalankan pada saat system berjalan. Direktory-direktory tersebut antara lain adalah :
    - /tmp/compiled
    - /tmp/var
    - /tmp/var/run
    - /tmp/var/log
    - /tmp/var/lock
    - /tmp/var/lock/subsys
19. File /etc/XF86Config akan terbentuk dengan cara membaca konfigurasi LTSP di file /tftpboot/lts/ltsroot/etc/lts.conf. Didalam file konfigurasi tersebut terdapat informasi tentang tipe mouse, dan X parameter beserta kombinasi lain.
20. Kemudian, skrip /tmp/start_ws akan terbentuk. Skrip ini akan menjalankan X server pada komputer terminal. Ketika skrip ini menjalankan, ia akan mengirim XDMCP query ke XDM server, pengiriman ini sesuai dengan konfigurasi di /tftpboot/lts/ltsroot/etc/lts.conf.
21. File /tmp/syslog.conf akan terbentuk. File ini akan memberikan informasi konfigurasi yang dibutuhkan oleh syslogd, sesuai konfigurasi di lts.conf. /tmp/syslog.conf ini kemudian akan di simbolik link-kan ke /etc/syslog.conf
22. Kemudian, syslogd berjalan sesuai dengan konfigurasi /etc/syslog.conf yang di simbolik link-kan ke /tmp/syslog.conf tersebut.
23. Kendali kemudian dikembalikan pada init. Init akan melihat baris initdefault, yang menentukan system akan dijalankan pada run level mana.
    Ltscore-3.01 menjalankan init 2.
    Runlevel 2 akan menyebabkan system menjalankan skrip set_runlevel, skrip ini membaca lts.conf dan menentukan workstation itu akan berjalan pada init berapa. Standard run levels untuk LTSP adalah 3,4 dan 5.
    3- runlevel ini akan menjalankan shell, sangat berguna untuk melakukan debugging.
    4- runlevel ini akan menjalankan satu atau lebih sesi telnet dalam mode teks.

5- runlevel ini untuk mode grafik. X window akan dijalankan. Setelah Xserver dijalankan di workstation (melalui skrip /tmp/start_ws), kemudian ia mengirimkan XDMCP query ke server. Pengiriman query ini akan memunculkan layar login ke server. Untuk ini, anda perlu menjalankan display manager, seperti XDM, GDM atau KDM pada sisi server. Jika user login, proses akan dijalankan pada server, tapi display output dari proses tersebut akan ditampilkan pada workstation atau komputer terminal.

## 3. Arsitektur Jaringan Cluster OpenMosix




gambar Arsitektur Jaringan Cluster OpenMosix

Arsitektur ini digunakan baik untuk yang dengan disk maupun tanpa disk. Pada jaringan cluster dengan harddisk, jaringan cluster openMosix difokuskan pada mekanisme dari OS dan hardware untuk menyatukan sumber daya setiap node (bisa CPU, memori, disk, dsb), dimana suatu task ditangani secara bersama-sama. Kata bersama-sama disini yang membedakan cluster dengan mekanisme jaringan biasa yang menjalankan program client server.

Pada jaringan cluster tanpa hardisk, saat komputer terminal dinyalakan komputer akan mencari “kernel” pada disket boot. Client 1,2,3,4 melakukan proses booting dengan disket boot. Kode booting yang tersimpan dalam “kernel” pada masing-nasing client segera mencari dhcp server ke jaringan local untuk mendapatkan alamat IP. DHCP Server pada server LTSP menanggapi request tersebut dan segera memberikan alamat IP kepada komputer yang request. Server akan memberikan tanggapan dengan mengirimkan paket informasi lainnya. Informasi yang dibutuhkan kernel untuk dapat melanjutkan proses selanjutnya.

Spesifikasi komputer Jaringan Cluster OpenMosix :

**Server :**

- Processor AMD Duron 1300 MHz On Board Motherboard ECS K7SOM+
- Memori DDR 256 MB
- Hard Disk 40 GB 5400 RPM
- LAN Card 10/100 MBPS Edimax
- Switch 100 MBPS
- VGA On board 32 MB share memory
- Sistem Operasi Linux RedHat 9.0, kernel 2.4.21-openMosix

**Client :**

- Processor AMD Duron 1300 MHz On Board Motherboard ECS K7SOM+
- Memori DDR 128 MB
- Hard Disk 40 GB 5400 RPM
- LAN Card 10/100 MBPS Edimax
- VGA Onboard 32 MB Share memory
- Sistem Operasi Linux RedHat 9.0, kernel 2.4.21-openMosix

## 4. Pengujian Cluster Open mosix dan OpenMosix diskless dengan Povray

Pengujian cluster dengan menggunakan Aplikasi rendering yaitu povray untuk melihat efektifitas cluster diskless dalam merender gambar. Adapun data hasil pengujian terhadap kedua bentuk jaringan cluster tersebut adalah :

### a. Data OpenMosix

| Banyak Pengulangan | Waktu (detik) 1 komputer | Waktu (detik) 2 komputer | Waktu (detik) 3 komputer |
|---|---|---|---|
| 1 | 40 | 37 | 34 |
| 2 | 45 | 37 | 41 |
| 3 | 43 | 41 | 38 |
| 4 | 41 | 36 | 38 |
| 5 | 57 | 44 | 38 |
| 6 | 38 | 40 | 41 |
| 7 | 43 | 40 | 41 |
| Jumlah | 307 | 275 | 271 |
| Rata-rata | 43,857 | 39,285 | 38,714 |

### b. Data OpenMosix Diskless

| Banyak Pengulangan | Waktu (detik) 1 komputer | Waktu (detik) 2 komputer | Waktu (detik) 3 komputer |
|---|---|---|---|
| 1 | 41 | 37 | 34 |
| 2 | 45 | 39 | 43 |
| 3 | 43 | 42 | 39 |
| 4 | 41 | 38 | 38 |
| 5 | 57 | 44 | 39 |
| 6 | 38 | 40 | 43 |
| 7 | 43 | 42 | 41 |
| Jumlah | 308 | 282 | 277 |
| Rata-rata | 44 | 39,285 | 39,571 |

Dari data di atas, terlihat perbedaan waktu proses yang di lakukan oleh dua jenis cluster ini tidak terlalu jauh, oleh karena itu penggunaan diskless untuk cluster openMosix dengan fungsi sebagai pengolahan gambar efektif diterapkan meskipun yang terbaik adalah jaringan cluster openMosix yang bukan diskless.

## 5. Penutup

Dari hasil yang diperoleh selama pengujian cluster openMosix maka dapat disimpulkan bahwa mengambil kesimpulan bahwa:

1. Tidak terdapat perbedaan kecepatan proses yang signifikan antara jaringan dengan disk dan tanpa disk, sehinggan jaringan cluster tanpa disk dapat diimplementasikan tanpa harus mengorbankan kinerja jaringan cluster
2. Hal ini disebabkan karena disk hanya dibutuhkan pada saat booting dan bukan pada saat eksekusi.
3. Pengambilan proses dari server cluster openMosix dan openMosix Diskless terjadi secara otomatis tanpa aplikasi yang bersangkutan dirubah karena openMosix bekerja pada level kernel.

Penelitian ini sebenarnya masih jauh dari sempurna, tetapi sudah dapat dijadikan dasar yang kuat untuk menerapkan jaringan cluster tanpa hardisk. Untuk penelitian selanjutnya dapat dilakukan perbandingan kinerja dengan jumlah komputer yang lebih banyak dan proses yang lebih berat, misalnya simulasi. Untuk tahap lebih lanjut , program untuk booting sebaiknya dapat disimpan dalam ROM (bootROM) sehingga tidak diperlukan disk drive untuk floppy dan pada akhirnya dapat menghemat hardware yang digunakan

## 7. Daftar Pustaka

[1] Anonim, *Instalasi Cluster dengan openMosix*, URL : http://www.clustering.org
[2] Anonim, *Kompilasi dan Konfigurasi Kernel* , URL : http://www.kernel.org
[3] Anonim, *POV-ray source code*, URL: http://www.povray.org/povlegal.html
[4] Anonim, *The userland-tools of the openMosix-system*, URL: http://openmosix.sourceforge.net/
[5] Michael J. Quinn, *Parallel Computing : Theory and Practice Second Edition*, 1994.
[6] Mulyadi Santosa, *Clustering di Linux dengan OpenMosix* Email : A_Mulyadi@telkom.net
[7] Ted G. Lewis & Hesham El-rewini, *Introduction to Parallel Computing*, Prentice-Hall Internasional Editions, 1992.
[8] Rusmanto, *Buku Mini InfoLinux : Panduan Mudah Membuat Diskless System*, Dian Rakyat, Jakarta, 2003.
[9] Hasan B.Usman Lamatungga & R.Fadli U. L., *Tip dan Trik Jaringan Tanpa Hardisk*, PT Elex Media Komputindo, Jakarta, 2003.
[10] Dr. Richardus Eko Indrajit & Hasan B. Usman Lamatungga, *Buku Pintar Linux : Membangun Jaringan Diskless Berbasis Linux*, PT Elex Media Komputindo, Jakarta, 2003.