Sistem Operasi Terdistribusi

Wahyu Wijanarko

wahyu@wahyu.com
http://wahyu.com

PENDAHULUAN

Latar Belakang

Perkembangan pesat teknologi informasi menyebabkan bertambahnya permintaan suatu sistem, baik berupa perangkat keras maupun perangkat lunak yang dapat digunakan dengan baik dan cepat.

Permintaan yang terus bertambah ini tidak sebanding dengan kemampuan perangkat keras yang ada. Salah satu cara untuk mengatasi hal itu dibuat pengembangan di sisi perangkat lunak dengan membuat suatu sistem virtual di mana beberapa perangkat keras atau komputer dihubungkan dalam jaringan dan diatur oleh sebuah sistem operasi yang mengatur seluruh proses yang ada pada setiap komputer tersebut sehingga memungkinkan proses berjalan dengan cepat. Sistem operasi yang mengatur proses ini sering disebut sebagai sistem operasi terdistribusi (*distributed operating system*) .

Sistem operasi terdistribusi ini sekarang menjadi trend, terutama untuk *research* yang kadang membutuhkan CPU yang sangat cepat untuk melakukan perhitungan yang sangat kompleks. Dalam makalah ini akan dibahas mengenai sistem operasi terdistribusi, terutama untuk mengetahui apa dan bagaimana cara sistem ini bekerja.

Apakah yang Dimaksud Dengan Sistem Operasi Terdistribusi?

Sistem operasi terdistribusi adalah salah satu implementasi dari sistem terdistribusi, di mana sekumpulan komputer dan prosesor yang heterogen terhubung dalam suatu jaringan. Koleksi-koleksi dari objek-objek ini secara tertutup bekerja secara bersama-sama untuk melakukan suatu tugas atau pekerjaan tertentu. Tujuan utamanya adalah untuk memberikan hasil secara lebih, terutama dalam:

- *file system*
- *name space*
- waktu pengolahan
- keamanan
- akses ke seluruh *resources*, seperti prosesor, memori, penyimpanan sekunder, dan perangkat keras.

Sistem Operasi Terdistribusi Versus Sistem Operasi Jaringan

Suatu sistem operasi terdistribusi yang sejati adalah yang berjalan pada beberapa buah mesin, yang tidak melakukan *sharing* memori, tetapi terlihat bagi user sebagai satu buah komputer *single*. Pengguna tidak perlu memikirkan keberadaan perangkat keras yang ada, seperti prosesor. Contoh dari sistem seperti ini adalah Amoeba.

Sistem operasi terdistribusi berbeda dengan sistem operasi jaringan. Untuk dapat membedakannya, sistem operasi jaringan memiliki ciri-ciri sebagai berikut:

a. Tiap komputer memiliki sistem operasi sendiri
b. Tiap personal komputer memiliki sistem file sendiri, di mana data-data disimpan
c. Sistem operasi tiap komputer dapat berbeda-beda atau heterogen
d. Pengguna harus memikirkan keberadaan komputer lain yang terhubung, dan harus mengakses, biasanya menggunakan remote login (telnet)
e. File system dapat digunakan dengan dukungan NFS

Contoh dari sistem ini adalah UNIX dan LINUX Server.

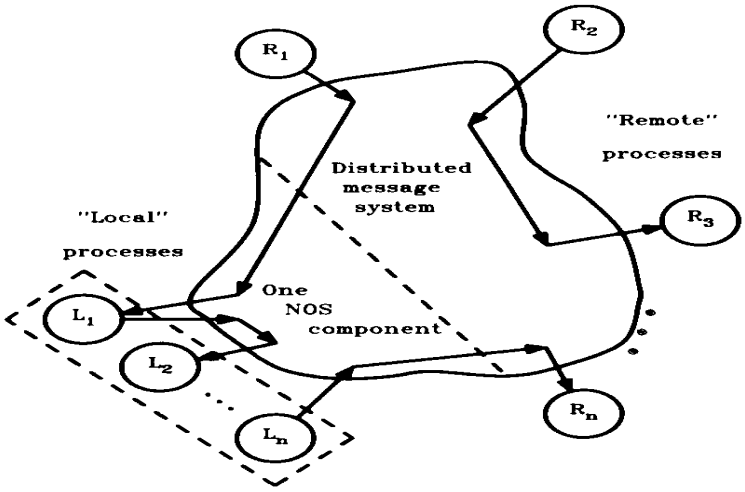


Fig. 2b. Logical interconnection.

Skema Network Operating System

(http://www.webstart.com/jed/papers/Managing-Domains/Figure-2b.gif)

MANFAAT DAN KEUNGGULAN SISTEM OPERASI TERDISTRIBUSI

Sistem operasi terdistribusi memiliki manfaat dalam banyak sistem dan dunia komputasi yang luas. Manfaat-manfaat ini termasuk dalan *sharing resource*, waktu komputasi, reliabilitas, dan komunikasi (Silverschatz Galvin, 1998 hal. 17).

Shared Resource

Walaupun perangkat sekarang sudah memiliki kemampuan yang cepat dalam proses-proses komputasi, atau misal dalam mengakses data, tetapi pengguna masih saja menginginkan sistem berjalan dengan lebih cepat. Apabila hardware terbatas, kecepatan yang diinginkan user dapat diatasi dengan menggabung perangkat yang ada dengan sistem DOS (*Distributed Operating System*).

Manfaat Komputasi

Salah satu keunggulan sistem operasi terdistribusi ini adalah bahwa komputasi berjalan dalam keadaan pararel. Proses komputasi ini dipecah dalam banyak titik (*nodes*), yang mungkin berupa komputer pribadi, prosesor tersendiri, dan kemungkinan perangkat prosesor-prosesor yang lain. Sistem operasi terdistribusi ini bekerja baik dalam memecah komputasi ini dan baik pula dalam mengambil kembali

hasil komputasi dari titik-titik *cluster* untuk ditampilkan hasilnya.

Reliabilitas

Fitur unik yang dimiliki oleh DOS ini adalah reliabilitas. Berdasarkan *design* dan implementasi dari *design* sistem ini, maka hilangnya suatu node tidak akan berdampak terhadap integritas system. Hal ini berbeda dengan komputer personal, apabila ada salah satu *hardware* yang mengalami kerusakan, maka system akan berjalan tidak seimbang, bahkan sistem bisa tidak dapat berjalan atau mati.

Dalam sistem operasi terdistribusi tadi sebenarnya cara kerjanya mirip dengan personal computer, tetapi bedanya apabila ada *node* yang *mati*, maka akan terjadi proses *halt* terhadap *node* tersebut dan proses komputasi dapat dialihkan. Hal ini akan membuat sistem DOS selalu memiliki reliabilitas yang tinggi.

Komunikasi

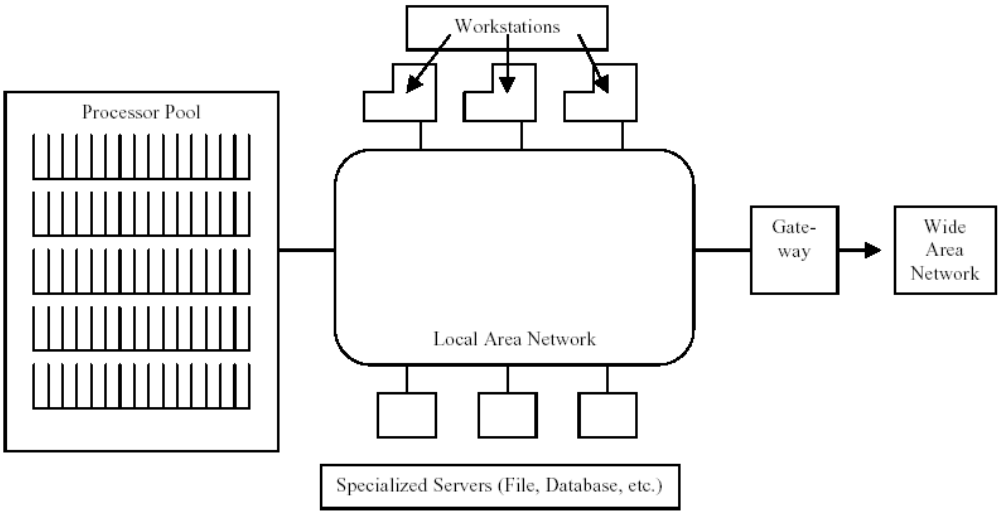
Sistem operasi terdistribusi biasanya berjalan dalam jaringan, dan biasanya melayani koneksi jaringan. Sistem ini biasanya digunakan user untuk proses *networking*. User dapat saling bertukar data, atau saling berkomunikasi antar titik baik secara LAN maupun WAN.

Aplikasi sistem operasi terdistribusi berbasis Linux.

(http://www.tu-chemnitz.de/mbv/TechnThDyn/mpf/Athlon_Cluster/cluster1.jpg)

HARDWARE SISTEM OPERASI TERDISTRIBUSI

Sistem operasi terdistribusi, yang saat ini akan dibahas sebagai titik tolak adalah Amoeba, yang saat ini banyak digunakan sebagai salah satu implementasi dari sistem operasi terdistribusi itu sendiri. Sistem Amoeba ini tumbuh dari bawah hingga akhirnya tumbuh menjadi sistem operasi terdistribusi.



Design Sistem Operasi Amoeba

(http://people.msoe.edu/~sebern/courses/cs384/papers9899/fedke.pdf)

Sistem operasi terdistribusi pada umumnya memerlukan *hardware* secara spesifik. Komponen utama dalam sistem ini adalah : *workstation*, LAN, *gateway*, dan *processor pool*, seperti yang diilustrasikan pada gambar di atas.

Workstation atau komputer personal mengeksekusi proses yang memerlukan interaksi dari user seperti *text editor* atau *manager* berbasis *window*. Server khusus memiliki fungsi untuk melakukan tugas yang spesifik. Server ini mengambil alih proses yang memerlukan I/O yang khusus dari larikan *disk*. *Gateway* berfungsi untuk mengambil alih tugas untuk terhubung ke jaringan WAN.

Procesor pool mengambil alih semua proses yang lain. Tiap unit ini biasanya terdiri dari prosesor, memori lokal, dan koneksi jaringan. Tiap prosesor mengerjakan satu buah proses sampai prosesor yang tidak digunakan habis. Untuk selanjutnya proses yang lain berada dalam antrian menunggu proses yang lain selesai. Inilah keunggulan sistem operasi terdistribusi dalam hal reliabilitas. Apabila ada satu unit pemroses yang mati, maka proses yang dialokasikan harus di *restart*, tetapi integritas sistem tidak akan terganggu, apabila proses deteksi berjalan dengan baik. Desain sistem ini memungkinkan untuk 10

sampai 100 prosesor.

Spesifikasi perangkat keras yang harus disediakan pada tiap *cluster* minimalnya adalah :

File server: 16 MB RAM, 300MB HD, Ethernet card.

Workstation: 8 MB RAM, monitor, keyboard, mouse

Pool processor: 4 MB RAM, 3.5” floppy drive

ARSITEKTUR SOFTWARE

Sistem operasi terdistribusi sejati memiliki arsiitektur *software* yang unik. Arsitektur software ini dikarakterkan dalam objek di dalam hubungan antara klien dan server. Proses-proses yang terjadi di klien menggunakan *remote procedure* yang memanggil dan mengirimkan *request* ke server untuk memproses data atau objek yang dibawa. Tiap objek yang dibawa memiliki karakteristik yang disebut sebagai kapabilitas.

Kapabilitas ini besarnya adalah 128 bits. 48 bits pertama menunjukkan servis mana yang memiliki objek tersebut. 24 bits berikutnya adalah nomor dari objek. 8 bits berikutnya menampilkan operasi yang diijinkan terhadap objek yang bersangkutan. Dan 48 bits terakhir merupakan “*check field*” yang merupakan *field* yang telah terenkripsi agar tidak dapat dimodifikasi oleh proses yang lain.

Operasi diselesaikan oleh RPC (*remote procedure calls*) yang dibuat oleh klien di dalam proses yang kecil dan ringan. Proses dengan tipe seperti ini memiliki bidang alamat sendiri, dan bisa saja memiliki satu atau lebih hubungan. Hubungan ini ketika berjalan memiliki program counter dan *stack* sendiri, tetapi dapat saling berbagi kode dan data antara hubungan lain di dalam proses. Ada 3 macam basis panggilan sistem yang dapat digunakan dalam proses yang dimiliki user, yaitu **do_operation, get_request,** dan **send_reply**.

Bagian yang pertama mengirimkan pesan ke server, setelah proses memblok sampai server mengirimkan balasan. Server menggunakan panggilan sistem ke dua untuk mengindikasikan bahwa server akan menerima pesan pada *port* tertentu. Server juga menggunakan panggilan sistem ke tiga untuk mengirimkan kembali informasi ke proses yang dipanggil.

Dengan dibangun dari perintah sistem yang primitif, maka sistem ini menjadi antarmuka untuk program aplikasi. Hal ini diselesaikan oleh tingkat dari pengarahan yang mengijinkan pengguna untuk berfikir terhadap struktur ini sebagai objek dan operasi-operasi terhadap objek ini.

Berhubungan dengan objek-objek adalah *class*. Kelas dapat berisi kelas yang lain dan juga hierarki

secara alami. Pewarisan membuat antarmuka objek untuk implementasi manipulasi objek seperti menghapus, membaca, menulis, dan sebagainya.

MANAJEMEN SISTEM

Manajemen Berkas

Dalam sistem operasi terdistribusi ini sistem berkas dipetakan dengan baik dengan berorientasi pada objek yang ada dan kapabilitasnya. Hal ini akan menjadi berkesan abstrak, terutama untuk kelas pengguna. Ada tingkatan yang lebih ekstra dalam pemetaan berkas yang ada, mulai dari simbol, pengurutan nama *path*, dan kapabilitasnya. Melalui sistem ini objek lokal tidak ada bedanya dengan objek publik.

Dalam sistem ini ada semacan tingkatan akses yang sebenarnya mirip UNIX. Setiap user dan group memiliki hak akses yang berbeda-beda pada setiap berkas atau folder yang ada pada sistem operasi terdistribusi.

Dalam implementasi sistem Amoeba, terutama di negeri Belanda, hak akses yang dimiliki pengguna terbatas pada hak baca file, tulis/membuat file, dan hapus file. Dengan hal ini, maka keamanan server dapat terjaga.

Pelayanan terhadap direktori yang ada dibuat sangat ketat dalam hal keamanan. Bahkan dibuat semacan kode acak yang akan menyandikan file tersebut sehingga tidak mudah dibaca oleh siapapun. Kode penyandinya akan digunakan lagi oleh sistem untuk mengembalikan file seperti semula kepada user. Kode ini hanya akan diberikan kepada pemilik file tersebut. Jadi ketika user mengakses file/berkas yang bersangkutan, maka kode penyandi akan dibuat oleh sistem, agar pemilik file dapat membacanya.

Pelayanan direktori ini juga bertanggungjawab dalam hal *backup* sistem. Hal ini akan menyebabkan file selalu berada dalam keadaan yang aman, dan lebih kebal tehadap gangguan yang terjadi di dalam sistem, karena pelayanan direktori ini menyimpan *cache* dari file atau direktori yang berada pada sistem.

Manajemen Proses

Dalam sistem operasi terdistribusi yang sejati, tiap proses berada pada alamat segmen-segmen virtual. Proses-proses ini dapat memiliki lebih dari satu hubungan. Kaitan-kaitan ini dialokasikan ke prosesor-prosesor sampai semua prosesor habis digunakan. Hasil dari manajemen proses seperti ini menghasilkan utilisasi yang lebih baik, di mana tidak perlu *switch* apabila harus ada proses yang berat, karena satu proses dialokasikan ke satu prosesor. Sedangkan untuk proses yang tidak kebagian tempat,

maka akan masuk ke antrian. Kaitan-kaitan proses ini menggunakan *semaphore* untuk menunjukkan aktifitasnya (Tanenbaum hal. 1).

Masing- masing proses memiliki kontrol sendiri pada spasi alamatnya. Masing-masing proses dapat menambah atau menghapus segmen dari spasi alamat virtualnya melalui operasi pemetaan. Objek seperti *file* yang berisi kapabilitas, dan yang membaca adalah *kernel*, dan apabila proses diijinkan, maka ia dapat memetakan atau menghapus pemetaan segmen pada alamat virtualnya.

Untuk membangun sebuah proses, maka pendekripsi proses mengirimkannya ke kernel. Hal ini diketahui sebagai pengiriman *request* untuk proses. Sebuah deskriptor proses dapat berisi deskriptor host, kapabilitas proses, penanganan kapabilitas, dan juga jumlah segmen.

Deskriptor host berisi proses ini memiliki jenis apa, dan dapat berjalan di mana. Isinya adalah baris instruksi, kebutuhan memori, kelas mesin, informasi, dan sebagainya. Kernel harus memiliki deskriptor host yang sama untuk melanjutkan proses.

Kapabilitas proses adalah memiliki tingkatan lebih tinggi dari proses, yang mengatur apa yang dapat dilakukan oleh proses, atau proses ini hanya dapat dilakukan oleh siapa. Pengatur kapabilitas mirip dengan hal ini, tetapi hanya bekerja untuk proses yang tidak normal.

Alamat proses terenkapsulasi di dalam peta memori internal. Peta ini meiliki entri untuk setiap segmen dari alamat untuk proses yang potensial. Entri berisi alamat virtual, panjang segmen, pemetaan segmen, dan kapabilitas dari objek yang mengetahui dari mana objek tersebut diinisialisasi.

Ada juga kaitan pemetaan yang mendeskripsikan atribut yang lain, termasuk di antaranya mendefinisikan inisial keadaan dari kaitan, status prosesor, program counter, stack pointer, stack base, nilai register, dan keadaan sistem pemanggil. Hal ini mengijinkan deskriptor untuk digunakan di proses.

Proses memiliki dua macam keadaan, yaitu proses sedang berjalan atau sedang *stunned. Stunned* terjadi bila proses masih ada, tetapi tidak melakukan eksekusi apapun, atau sedang dalam proses debug. Pada keadaan ini kernel memberitahu komunikator (kernel yang lain) adanya proses yang dalam keadaan *stunned.* Kernel yang lain tersebut berusaha berkomunikasi dengan proses itu sampai proses di-*kill* atau proses tersebut berjalan kembali. *Debugging* dan migrasi pada proses ini selesai setelah adanya *stunning.*

JENIS SISTEM OPERASI TERDISTRIBUSI

Ada berbagai macam sistem operasi terdistribusi yang saat ini beredar dan banyak digunakan.

Keanekaragaman sistem ini dikarenakan semakin banyaknya sistem yang bersifat *opensource* sehingga banyak yang membangun OS sendiri sesuai dengan kebutuhan masing-masing, yang merupakan pengembangan dari OS *opensource* yang sudah ada.

Beberapa contoh dari sistem operasi terdistribusi ini diantaranya :

- Amoeba (Vrije Universiteit)

  Amoeba adalah sistem berbasis mikro-kernel yang tangguh yang menjadikan banyak *workstation* personal menjadi satu sistem terdistribusi secara transparan. Sistem ini sudah banyak digunakan di kalangan akademik, industri, dan pemerintah selama sekitar 5 tahun.

- Angel (City University of London)

  Angel didesain sebagai sistem operasi terdistribusi yang pararel, walaupun sekarang ditargetkan untuk PC dengan jaringan berkecepatan tinggi. Model komputasi ini memiliki manfaal ganda, yaitu memiliki biaya awal yang cukup murah dan juga biaya *incremental* yang rendah. Dengan memproses titik-titik di jaringan sebagai mesin *single* yang bersifat *shared memory*, menggunakan teknik *distributed virtual shared memory* (DVSM), sistem ini ditujukan baik bagi yang ingin meningkatkan performa dan menyediakan sistem yang portabel dan memiliki kegunaan yang tinggi pada setiap platform aplikasi.

- Chorus (Sun Microsystems)

  CHORUS merupakan keluarga dari sistem operasi berbasis mikro-kernel untuk mengatasi kebutuhan komputasi terdistribusi tingkat tinggi di dalam bidang telekomunikasi, internetworking, sistem tambahan, *realtime*, sistem UNIX, *supercomputing*, dan kegunaan yang tinggi. Multiserver CHORUS/MiX merupakan implementasi dari UNIX yang memberi kebebasan untuk secara dinamis mengintegrasikan bagian-bagian dari fungsi standar di UNIX dan juga service dan aplikasi-aplikasi di dalamnya.

- GLUnix (University of California, Berkeley)

  Sampai saat ini, *workstation* dengan modem tidak memberikan hasil yang baik untuk membuat eksekusi suatu sistem operasi terdistribusi dalam lingkungan yang *shared* dengan aplikasi yang berurutan. Hasil dari penelitian ini adalah untuk menempatkan *resource* untuk performa yang lebih baik baik untuk aplikasi pararel maupun yang seri/berurutan. Untuk merealisasikan hal ini, maka sistem operasi harus menjadwalkan pencabangan dari program pararel, mengidentifikasi *idle*

resource di jaringan, mengijinkan migrasi proses untuk mendukung keseimbangan *loading*, dan menghasilkan tumpuan untuk antar proses komunikasi.

- GUIDE

Guide (Grenoble Universities Integrated Distributed Environment) adalah sistem operasi terdistribusi yang berorientasi obyek untuk pempangunan dan operasi dari aplikasi terdistribusi pada PC atau server dengan jaringan yang tersambung LAN. Guide adalah hasil penggabungan Bull and the IMAG Research Institute (Universities of Grenoble), yang telah membangun Bull-IMAG joint Research Laboratory. Ini juga memiliki kaitan erat dengan COMANDOS Esprit Project (Construction and Management of Distributed Open Systems) dan BROADCAST Esprit Basic Research project.

- Hurricane

Sistem operasi Hurricane memiliki hierarki sebagai sistem operasi dengan *cluster* yang merupakan implementasi dari *Hector multiprocessor*. Peng-*cluster*-an mengatur resource pada sistem, menggunakan pasangan yang ketat antara *cluster*, dan kehilangan pasangan pada *cluster*. Prinsip sistem terdistribusi diaplikasikan dengan mendistribusikan dan mereplika servis pada sistem dan objek data untuk meningkatkan kelokalan, meningkatkan konkurensi, dan untuk mencegah sistem terpusat, sehingga membuat sistem berimbang.

- Mach (Carnegie Mellon University)

Mach adalah satu dari beberapa komunitas penelitian tentang sistem operasi. Sistem ini aslinya dimulai di CMU, dan Mach menjadi basis dari banyak sistem penelitian. Walaupun pekerjaan dengan Mach di CMU sudah lama tidak diterapkan, tetapi masih banyak kelompok-kelompok lain yang masih menggunakan Mach sebagai basis pada penelitiannya.

- Mach at OSF (OSF Research Institute)

OSF Research Institute masih menggunakan teknologi yang dimulai dari CMU dan menggunakan ini sebagai basis dari banyak penelitian, termasuk sistem operasi untuk mesin pararel , kernel berorientasi objek yang aman, dan penelitian-penelitian tentang sistem operasi yang lain.

- Maruti (University of Maryland) Group Members

Maruti adalah sistem operasi berbasis waktu, yang merupakan proyek di University of Maryland.

Dengan Maruti 3.0, kita memasuki fase baru pada proyek ini. Menurut mereka, mereka memiliki sistem operasi yang lebih nyaman untuk kalangan yang lebih luas.

- Masix (Blaise Pascal Institute MASI Laboratory)

Masix adalah sistem operasi terdistribusi yang berbasis pada mikro kernel dari Mach, yang saat ini di bawah pengembangan dari MASI Laboratory. Tujuan utama dari sistem ini adalah untuk secara simultan mengeksekusi banyak data aplikasi personal, yang berjalan baik baik di semua platform, baik Unix, DOS, OS/2 dan Win32.

- MOSIX (Hebrew University, Jerusalem, Israel)

Sebuah solusi untuk masalah saat ini menjadi ada untuk lingkungan multikomputer, yang disebut MOSIX. Mosix adalah pengembangan dari UNIX, yang mengijinkan user untuk menggunakan *resource* yang ada tanpa ada perubahan pada level aplikasi. Dengan penggunaan yang transparan, algoritma proses migrasi dinamis, MOSIX melayani servis jaringan, seperti NFS, TCP/IP, dari UNIX, untuk level proses, dengan menggunakan penyeimbangan *load* dan distribusi dinamis pada *cluster-cluster* yang homogen.

- Plan 9 (Bell Labs Computing Science Research Center)

Plan 9 adalah sistem operasi baru yang dibangun di Bell Labs. Ini adalah sebuah sistem yang terdistribusi. Pada kebanyakan konfigurasi, ini menggunakan tiga macam komponen : terminal yang ada pada meja pengguna, *server* file yang menyimpan data permanen, dan *server* CPU yang melayani CPU lainnya lebih cepat, authentikasi user, dan *network gateways*. Salah satu kesemuan yang menarik dari Plan 9 adakah pengiriman file yang esensial pada semua servis system.

- Puma and relatives (Sandia National Laboratory)

Sistem operasi Puma menargetkan aplikasi dengan performa tinggi yang dipasangkan dengan arsitektur memory terdistribusi. Ini adalah turunan dari SUNMOS.

Sistem Operasi Tedistribusi Lainnya

Selain sistem operasi-sistem operasi di atas, masih banyak lagi sistem operasi terdistribusi yang dibangun, baik secara *opensource* maupun yang '*closed source*'. Sistem-sistem itu diantaranya adalah

- Alpha Kernel (Carnegie Mellon University)

- QNX
- Spring Real-Time Project (University of Massachsetts, Amherst)
- Spring System (Sun)
- Sprite (University of California, Berkeley)
- Sting
- Sumo (Lancaster University)
- Tao Operating System (Tao Systems)
- Tigger (Trinity College Dublin)
- TUNES

KESIMPULAN

Dalam sistem operasi terdistribusi terjadi proses yang lebih rumit dari sistem yang biasa, tetapi dapat menghasilkan suatu sistem dengan performa dan kemampuan yang lebih.

Dari uraian di atas telah banyak disinggung keunggulan-keunggulan dari sistem operasi terdistribusi. Tetapi di samping keunggulan-keunggulan yang ada sistem ini juga memiliki kelemahan yang banyak, diantaranya adalah perawatan tiap cluster yang sangat sulit, selain itu juga boros daya, karena harus menghidupkan banyak CPU, membutuhkan jaringan berkecepatan tinggi.

Kelemahan-kelemahan tersebut sebenarnya tidak seberapa jika dibandingkan dengan hasilnya. Misalnya saja search engine paling ramai seperti Google™, yang menggunakan teknologi ini, karena hardware yang paling canggih saat ini masih belum mencukupi untuk menangani jutaan request ke server Google tiap detiknya, sehingga mereka harus membuat sistem pararel yang mampu melayani keperluan tersebut. Selain itu dalam dunia research, juga diperlukan sistem ini, terutama untuk melakukan perhitungan-perhitungan yang tentu saja sangat rumit dan membutuhkan pemroses yang hebat dan cepat supaya dapat segera dicari hasilnya.

REFERENSI

Hariyanto, Bambang, Ir. Sistem Operasi, Bandung : Informatika, 1997

Tanenbaum, Andrew S. The Amoeba Distributed Operating System

Gottlieb, Allen. Distributed Operating System Lectures Notes Spring 1997-1998

Silverschatz Galvin, Operating System Concepts, Addison-Wesley 1998

http://java.icmc.sc.usp.br/os2_course/

http://www.cs.arizona.edu/people/bridges/os/distributed.html

http://www.cs.vu.nl/~ast/books/mos2/sample1.pdf

http://people.msoe.edu/~sebern/courses/cs384/papers9899/fedke.pdf

http://www.cs.panam.edu/~meng/Course/CS6334/Note/master/

http://allan.ultra.nyu.edu/gotllieb/os

ftp://ftp.cse.ucse.edu

http://www.tu-chemnitz.de/mbv/TechnThDyn/mpf/Athlon_Cluster/cluster1.jpg

http://www.webstart.com/jed/papers/Managing-Domains/Figure-2b.gif

http://www.cs.vu.nl/pub/amoeba/

http://www.mosix.org

http://www.openmosix.org

http://www.google.com

BIOGRAFI PENULIS

Wahyu Wijanarko. Lahir di Kulonprogo, 7 Januari 1984. Menamatkan SMTA di SMU N 1 Bantul pada tahun 2001. Saat ini menjadi mahasiswa di Jurusan Teknik Elektro Fakultas Teknik Universitas Gadjah Mada Yogyakarta, dengan konsentrasi Sistem Komputer dan Informatika.
Menjadi anggota Open Source Initiative UGM, dan juga ikut serta dalam pengembangan sistem komputer UGM di UPT Pusat Komputer UGM. Menjadi anggota di berbagai milis komputer Indonesia, dan juga beberapa forum webmaster.
Berpengalaman di dalam pengembangan berbagai sistem informasi berbasis web maupun GUI, dan desain database dengan menggunakan software opensource, serta administrasi server berbasis GNU/Linux.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:
URL: *http://wahyu.com/*
Email: *wahyu@wahyu.com*