REVIEW TENTANG VIRTUALISASI

Rusydi Umar
Program Studi Teknik Informatika, Fakultas Teknologi Industri
Universitas Ahmad Dahlan Yogyakarta
email : rusydi_umar@rocketmail.com

Abstrak

Virtualisasi adalah cara untuk membuat komputer fisik bertindak bahwa seolah-olah komputer tersebut menjadi dua atau lebih komputer logika, dimana masing-masing komputer logika (nonfisik) mempunyai arsitektur dasar yang sama dengan komputer fisik. Virtualsasi digunakan untuk meningkatkan tingkat utilisasi dari komputer, karena sebagaimana kita ketahui, hampir semua komputer dalam keadaan nganggur (idle). Penggunaan kapasitas cpu berada dibawah 10% bahkan pada komputer server, kecuali pada cpu intensive applications. Paper ini akan membahas tentang mesin virtual (virtual machine), cluster, dan virtual cluster.

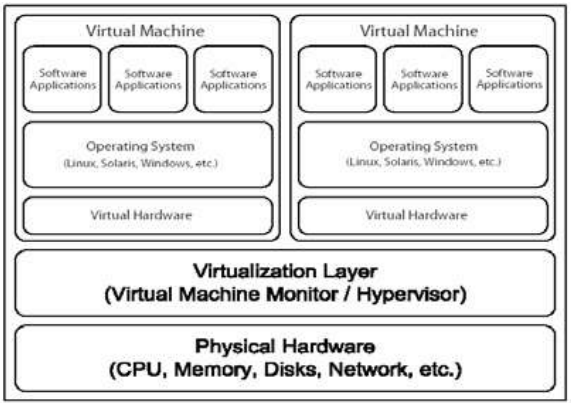
Kata Kunci : Cluster, Virtual Cluster, Virtual Mesin

1. PENDAHULUAN

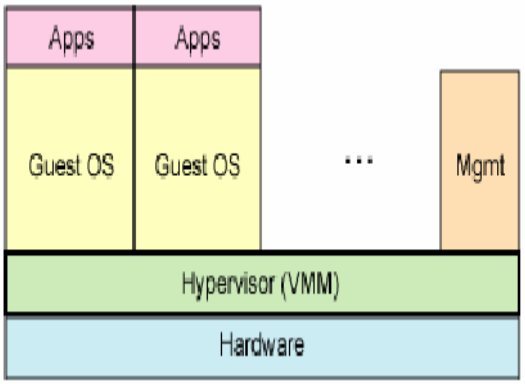
Teknologi virtualisasi adalah teknologi untuk membuat komputer fisik bertindak seolah-olah komputer tersebut adalah dua komputer nonfisik (komputer virtual) atau lebih. Masing-masing komputer nonfisik tersebut memiliki arsitektur dasar yang sama dengan komputer fisiknya. Ada berbagai cara untuk melakukan hal ini, tiap cara mempunyai kelebihan dan kekurangan masing-masing. Untuk membuat komputer fisik menjadi dua atau lebih komputer virtual, karakteristik perangkat kerasnya harus dikonstruksi kembali melalui perangkat lunak. Hal ini dapat dilakukan dengan lapisan perangkat lunak yang disebut abstraksi. Perangkat lunak abstraksi digunakan di banyak sistem perangkat lunak, termasuk di dalam keluarga sistem operasi windows. Windows *Hardware Abstraction Layer* (HAL) adalah sebuah contoh yang bagus dari sebuah abstraksi. Windows HAL menyediakan cara yang umum untuk semua *driver* guna berhubungan dengan perangkat keras yaitu dalam format yang umum. Hal ini membuat pekerjaan untuk membuat perangkat lunak dan *driver* menjadi lebih mudah karena *developer* tidak perlu menulis perangkat lunak khusus untuk tiap merek atau tipe dari komputer yang akan menjalankan perangkat lunak mereka. Abstraksi, dimana berhubungan dengan virtualisasi, adalah representasi dari sekumpulan perangkat keras umum yang keselurahnnya digerakkan oleh perangkat keras. Pada dasarnya abstraksi adalah perangkat yang bertindak sebagai perangkat keras. Teknologi virtualisasi mengijinkan instalasi dari sistem operasi pada perangkat keras yang sebenarnya tidak ada.

Virtualisasi adalah konsep yang mengijinkan komputer untuk dibagi dalam beberapa lingkungan pada saat yang sama. Lingkungan ini dapat saling berhubungan atau bahkan tanpa saling berhubungan sama sekali. Sebuah lingkungan mungkin bisa jadi sadar atau tidak bahwa lingkungan tersebut berjalan di lingkungan virtual. Lingkungan biasa disebut dengan mesin virtual (Virtual Machine (VM)). VM akan

selalu menjadi rumah bagi sistem operasi (misalnya Linux, Windows, etc). Instalasi sistem operasi ini biasa disebut sebagai sistem operasi tamu (Guest operating systems). Instruksi untuk VM biasanya diewatkan langsung ke perangkat keras sehingga memungkinkan lingkungan untuk beroperasi lebih cepat dan efisien dari sistem emulasi, meskipun instruksi yang lebih komplek harus diperangkap dan diterjemahkan untuk menjamin kompatibilitas dan abstraksi dengan perangkat keras fisik. Didalam komputer yang memjadi tuan rumah virtualisasi, komputer tersebut mempunyai sekumpulan perangkat keras tempat dimana sistem operasi diinstal (linux, Windows, atau yang lain). Sistem operasi mempunyai *platform* virtual terinstal dalam satu atau lebih mesin virtual yang bertindak sebagai perangkat keras yang terpisah dan mempunyai kemampuan untuk diinstal sistem operasi padanya, seperi terlihat pada Gambar 1[1].



Gambar 1. Arsitektur system Virtualisasi



Gambar 2. Full Virtualization

2. SEJARAH VIRTUALISASI

Diawal tahun 1960 an, IBM mengenalkan *Time Sharing*, yang mana merupakan pendorong utama dibidang virtualisasi. Pada tahun 1964 IBM mengenalkan IBM System/360, yang menyediakan kemampuan virtualisasi yang terbatas dan di buat oleh seorang legendaris yaitu Gene Amdhal. Kemudian, pada tahun yang sama CP-40 dirilis dan menggunakan kata mesin virtual dan memori virtual.

Pada tahun 1998, VMware ditemukan oleh Diane Greene dan suaminya Dr. Mendel Rosenblum bersama dengan mahasiswanya dari Universitas Stanford dan teman kerjanya dari Berkeley. Pada Oktober 1998, penemu ini mematenkan penemuannya tentang teknik virtualisasi, berdasarkan penelitian yang dilaksanakan di Universitas Stanford. Paten tersebut dikabulkan pada tanggal 28 Mei 2002. Rencana virtualisasi Microsoft di fokuskan kepada *rehosting* aplikasi, konsolidasi *server*, dan otomatisasi pembangunan perangkat lunak dan lingkungan pengujian. Mereka merilis produk virtualisasi mereka pertama kali yaitu Microsoft Virtual PC 2004 [1]

3. EMULASI DAN SIMULASI

Emulasi adalah konsep yang mengijinkan sebuah lingkungan untuk berakting sebagai sebuah lingkungan yang lain. Ini dapat digambarkan sebagai impersonasi yang canggih. Lingkungan adalah *platform* untuk eksekusi, sistem operasi atau arsitektur perangkat keras. Instruksi di terjemahkan dari lingkungan eksekusi ke instruksi yang sebenarnya, yaitu yang dipahami oleh lingkungan yang sebenarnya. Emulasi digunakan

untuk menjalankan tiruan lingkungan, pengembangan sistem operasi dan pengujian perangkat lunak. Lingkungan yang diemulasi menyandang beban performa yang besar karena tambahan beban yaitu pernerjemahan instruksi, bila dibandingkan dengan teknik virtualisasi.

Simulasi adalah konsep dimana lingkungan meniru lingkungan yang lain. Peniruan ini menerima input yang sudah didefinisikan sebelumnya dan menyediakan respon yang sudah didefinisikan sebelumnya. Hal ini merupakan cara termudah dan konsep paling sederhana untuk diimplementasikan. Lingkungan adalah *platform* tempat eksekusi, sistem operasi dan arsitektur perangkat keras. Simulator digunakan secara berbeda dibandingkan dengan emulasi atau virtualisasi. Meraka pada dasarnya digunakan di dalam perangkat keras, desain dan prototipe mikrocip. Dengan melakukan hal ini pengujian dapat dilakukan pada perangkat keras dan mikrocip yang akan dibuat. Hal ini mengurangi biaya dan resiko yang berhubungan dengan kesalahan yang dibuat pada perangkat keras dan chip sebelum meraka di buat secara masal. [1].

4. TIPE DARI VIRTUALISASI

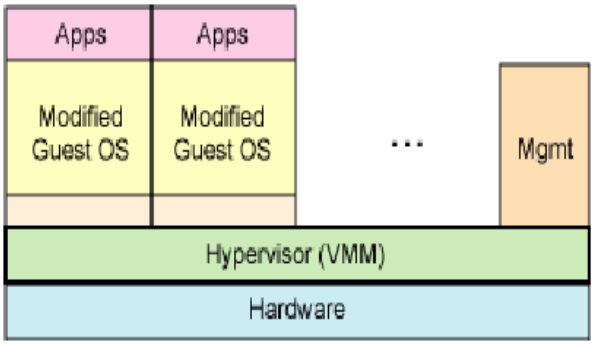
4.1. *Full Virtualization* (Virtualisasi Penuh)

Virtualisasi penuh yang biasa disebut dengan *native virtualization*. Model ini menggunakan mesin virtual yang menjembatani antara sistem operasi tamu dan perangkat keras aslinya, lihat Gambar 2. Menjembatani adalah kunci dari teknik ini, karena VM menjembatani antara sistem operasi tamu dengan perangkat keras. Beberapa instruksi yang dilindungi harus diperangkap dan ditangani dalam *hypervisor* karena perangkat keras dibawahnya tidak dimiliki oleh sistem operasi, tetapi diakses melalui *hypervisor*. [2]

Virtualisasi penuh lebih cepat daripada emulasi perangkat keras, tapi performanya lebih lambat dari perangkat keras langsung, karena jembatan/mediasi *hypervisor*. Keuntungan terbesar dari virtualisasi penuh adalah sistem operasi dapat dijalankan tanpa dimodifikasi. Batasan yang dimiliki hanyalah bahwa sistem operasi harus mendukung perangkat keras dibawahnya. Beberapa perangkat keras menimbulkan masalah pada metode virtualisasi penuh ini. Sebagai contoh beberapa instruksi yang sensitif yang perlu ditangani oleh VM tidak terperangkap. Oleh karena itu, *hypervisor* harus memindai mode pemerangkap khusus secara dinamis untuk menangani hal ini. *VMware ESX Server* dan *Microsoft Virtual Server* menggunakan tipe virtualisasi penuh. *Overhead* lebih penting pada tipe ini karena membutuhkan penerjemahan, pemerangkapan, dan emulasi dari beberapa instruksi. Intel® and AMD™ mulai menyediakan perangkat keras yang mendukung virtualisasi melalui Intel VT and AMD-V™, sehingga mengurangi *overhead*[2].

4.1. *Paravirtualization*

Paravirtualization adalah teknik lain yang popular yang mempunyai kemiripan dengan virtualisasi penuh. Metode ini menggunakan *hypervisor* untuk berbagi akses ke perangkat keras dibawahnya, tapi menyatukan kode yang sadar virtualisasi ke sistem operasi (lihat Gambar 3)[2]. Pendekatan ini menghilangkan kebutuhan rekompilasi atau pemerangkapan, karena sistem operasi berkooperasi dengan proses virtualisasi.



Gambar 3. Paravirtualization

Seperti disebutkan diatas, paravirtualization mengharuskan sistem operasi tamu dimodifikasi. Ini adalah kerugian bagi sistem operasi yang tidak siap dengan perubahan/modifikasi. Xen juga mengambil keuntungan dari *paravirtualization*. *Paravirtualization* menawarkan performa yang mendekati sistem yang tidak di virtualisasi. Mirip dengan virtualisasi penuh, beberapa sistem operasi dapat didukung secara bersamaan[2].

5. KEUNTUNGAN DARI VIRTUALISASI

Keuntungan operasional dan finansial dari virtualisasi adalah kunci dari komputasi enterprise dan lingkungan pengembangan perangkat lunak dapat disediakan oleh Virtualisasi. Berikut ini adalah keuntungan utama dari virtualisasi. [3]

5.1. Menggunakan perangkat keras yang ada dengan lebih baik

Beberapa dekade teakhir, prosesor telah berubah dari 8 bit ke 16 bit ke 32 bit dan sekarang ke 64 bit. Tiap perkembangan ini akan menaikkan ukuran prosesor dan meningkat jumlah memori dan ukuran media penyimpan yang dapat diakses oleh prosesor. Begitu juga dengan kecepatan dan kepadatan prosesor terus bertambah, dimana saat ini prosesor dapat dengan mudah melampaui 2 GHz, dan mempunyai banyak inti prosesor di tiap cipnya.

Menjalankan beberapa mesin virtual pada *server* yang telah ada akan mengefisienkan penggunaaan *server* tersebut. Multi prosesor atau *multi-core system* bahkan dapat menjalankan mesin virtual yang berbeda pada inti prosesor yang berbeda, hal ini mengambil keuntungan dari tiap inti prosesor.

5.2. Mengurangi Harga Perangkat Keras

Dengan virtualisasi tidak diperlukan lagi untuk membeli perangkat keras baru bila diinginkan untuk menambah server atau layanan. Menambah server atau layanan akan menambah kepuasan pelanggan/pengguna.

5.3. Mengurangi Infrastruktur IT

Setiap server fisik menggunakan listrik tertentu, ruang tertentu dan sistem pendinginan tertentu. Dengan mesin virtual maka kebutuhan listrik, kebutuhan ruang dan sistem pendingin tetap.

5.3. Menyederhanakan Sistem Administrasi

Menjalankan beberapa mesin virtual pada sebuah mesin fisik, mempengaruhi kesehatan dari sistem tersebut dan membutuhkan sebuah infrastruktur perangkat lunak untuk migrasi atau cloning bila terjadi kesalahan perangkat keras.

5.3. Meningkatkan Uptime Dan Mempercepat Failure Recovery

Portabilitas dari mesin virtual akan mebuatnya mudah untuk dipindahkan dari *server* yang lain jika ada kesalahan perangkat keras. Mesin virtual Xen dapat dipindahkan tanpa menggangu performanya, proses migrasinya transparan bila dilihat oleh pengguna, dan proses yang menggunakannya.

5.4. Menyederhanakan Ekspansi Kapasitas

Mesin virtual dapat dipindahkan dari satu mesin fisik ke mesin fisik yang lain sehingga bisa mengambil keuntungan dari peningkatan perangkat keras, seperti CPU yang lebh kuat, tambahan inti CPU, tambahan memori, tambahan dan kartu jaringan dan lain-lain.

5.5. Lebih sederhana dalam dukungan perangkat lunak asli

Dengan menjalankan sistem operasi dalam partisi logika, pengguna dapat meningkatkan ke sistem operasi yang baru, tanpa kehilangan kemampuan untuk mengoperasikan perangkat lunak aslinya.

5.6. Menyederhanakan Pengembangan System-Level

Solusi tradisional untuk testing *kernel* dan *driver* adalah dengan me-*reboot kernel* tersebut, tapi dengan adanya mesin virtual, maka *reboot* menjadi lebih cepat dari pada *reboot* pada mesin fisik. Pendekatan ini juga penting untuk *low-level debugging* jika memakai mesin virtual karena lingkungan pengembangan, sistem pengembangan dan mesin virtual semuanya dapat berada dalam satu *desktop* pada waktu yang sama.

5.7. Menyederhanakan Instalasi dan Deployment Sistem

Dengan menggunakan sistem pengembangan yang tidak tergantung dengan perangkat keras tertentu, maka akan sangat mudah untuk migrasi sistem tersebut ke tempat yang lebih cepat, lebih kuat, lebih baik perangkat input outputnya dan seterusnya.

5.8. Menyederhanakan sistem dan Testing Aplikasi

Instalasi produk perangkat lunak yang tidak membutuhkan perangkat keras khusus dan mengujinya pada banyak sistem operasi yang berbeda akan menjadi mudah dengan adanya virtualisasi *server*.

6. KERUGIAN VIRTUALISASI

Banyak alasan untuk menggunakan virtualisasi dalam lingkungan komputasi. Walau demikian tetap saja masih ada kekurangan dari adanya teknologi virtualisasi yaitu [3].

6.1. Satu Titik Kesalahan

Kelemahan dari virtualisasi *server* adalah menaikkan kemungkinan kesalahan pada sebuah perangkat keras yang ditempati oleh beberapa *server* virtual. Jika banyak *server* dan layanan yang berhubungan dengannya berjalan pada mesin masing-masing, maka kesalahan pada satu mesin akan berakibat hanya pada satu *server* saja. Berbeda dengan beberapa *server* yang menempati sebuah mesin, kesalahan pada mesin akan mengakibatkan kesalahan pada seluruh server virtual.

Untuk menangani hal ini maka ada langkah dan rencana khusus, yang diantaranya adalah:

1. *Set up* perangkat keras cadangan seperti *network interface cards* pada sistem *host*, dan satukan seluruhnya sehingga kesalahan pada sebuah kartu akan terlihat oleh mesin virtual.
2. Beli dan pelihara perangkat keras duplikat untuk sistem fisik yang ditempati oleh mesin virtual yang penting.
3. Replikasikan mesin virtual yang ditempati oleh layanan yang amat penting ke seluruh perangkat keras, sehingga kesalahan pada perangkat keras yang satu dapat diback up perangkat keras yang lain.
4. Jalankan perangkat lunak sistem monitoring terpusat untuk memberi tahu bila ada kesalahan perangkata keras atau perangkat lunak sebelum kesalahan itu menjadi kritis. Hal ini memberi kesempatan untuk memindahkan layanan yang terpengaruh ke mesin fisik yang lain

6.2. Kepadatan Saluran Jaringan

Sebagian besar sistem virtualisasi penuh menggunakan kartu jaringan virtual, *subnet* virtual dan menjembatani paket untuk dipetakan ke perangkat keras fisik. Jika *host* fisik hanya menyediakan sebuah kartu jaringan, dan menjalankan beberapa mesin virtual yang mengeksekusi *network intensive task*, maka permintaan layanan ke sebuah kartu jaringan menjadi sangat padat. Untuk menangani hal ini, adalah dengan menginstal beberapa kartu jaringan pada *host* fisik dan menghubungkan masing-masing kartu ke mesin virtual. Sayangnya dengan menerapkan hal ini akan menambah kesulitan dalam memigrasi sebuah mesin virtual dari mesin fisik yang satu ke mesin fisik yang lain.

6.3. Menaikkan Kompleksitas Jaringan dan Waktu *Debug*

Penggunaan *virtual network interfaces* dan mesin virtual membuat manajemen jaringan menjadi lebih rumit, sehingga desain dan perancangan infrastruktur mesin virtual harus sangat cermat dan memakan banyak waktu.

6.4. Menaikkan kompleksitas Administrasi

Sebelumnya telah disebutkan tentang menyederhanakan sistem administrasi. Hal ini tidak berlaku bila menggunakan sistem manajemen terdistribusi yang tidak mengenal mesin virtual. Untuk itu berhati-hatilah saat membeli virtual mesin, yaitu virtual mesin yang dapat berkomunikasi dengan sistem manajemen terdistribusi.

7. KOMPUTER CLUSTER

Sebagian besar komputer saat ini bekerja dengan 32 bit pada waktu yang sama, bahkan ada yang bekerja dengan operasi 64 bit, yang mirip dengan menaikkan lebar

jalur tanpa hambartan. Metode lain untuk menaikkan performa adalah dengan menaikkan *clock speed*, yang mirip dengan menaikkan batas kecepatan maksimum. Jadi komputer modern sangat mirip dengan jalan bebas hambatan yang sangat lebar dan batas kecepatan maksimum yang sangat tinngi [5].

Walaupun demikian tetap saja ada batas performa yang dapat dicapai hanya dengan menambah *clock speed* atau lebar *bus*. Sebagai pengganti sebuah komputer, kenapa tidak menggunakan banyak komputer untuk memecahkan masalah?

Dalam bentuk yang paling sederhana, *cluster* adalah dua atau lebih komputer yang bekerja sama untuk menyelesaikan masalah. Jangan rancu dengan model komputasi *client-server*, dimana sebuah aplikasi mungkin secara logika satu atau dua klien meminta layanan ke satu atau lebih *server*. Ide dibalik *cluster* adalah menggabungkan kekuatan komputasi dari simpul yang terlibat untuk memberikan skalabilitas yang lebih tinggi, lebih komplek, atau membuat cadangan untuk memastikan ketersediaan kekuatan komputasi. Jadi tidak hanya sesederhana dimana klien meminta ke satu atau lebih *server*, cluster menggunakan beberapa mesin untuk menyediakan lingkungan komputasi yang lebih kuat melalui *single system image* [6].

Cluster dapat dibagi menjadi dua kategori: *High Availability* (HA) dan *High-Performance Computing* (HPC). *Cluster* HA menyediakan layanan yang sangat dapat diandalkan sedangkan *Cluster* HPC menyediakan kekuatan komputasi yang lebih dari sebuah komputer.

7.1. *High Availability* (HA) *clusters*

Pada HA *cluster*, pada umumnya terdapat dua atau lebih mesin yang tahan banting yang menduplikasikan fungsi-funsinya ke masing-masing mesin. Dua skim yang pada umumnya dipakai untuk mendapatkan hal tersebut. Pada skim pertama, satu mesin secara diam-diam memantau mesin yang lain, dan menunggu untuk mengambil alih bila ada kesalahan. Pada skim yang kedua, kedua mesin dalam keadaan aktif. Dalam lingkungan ini harus dipatuhi bahwa beban jangan sampai melebihi 50 persen pada setiap mesin, jika tidak maka akan timbul masalah kapasitas bila ada mesin yang rusak. Kedua simpul ini biasanya meiliki *disk drive array* bersama.

Atau, daripada mempunyai kedua simpul yang mengakses *array* yang sama, dapat dibuat dua buah *array* yang terpisah yang secara konstan saling mereplika untuk menangani toleransi kesalahan. Dalam subsistem ini diperlukan jaminan integritas data. Jika ada masalah, satu sistem harus dapat mengantisipasi mesin yang lain, sehingga menjamin integritas [5].

7.2. *High-Performance Computing*

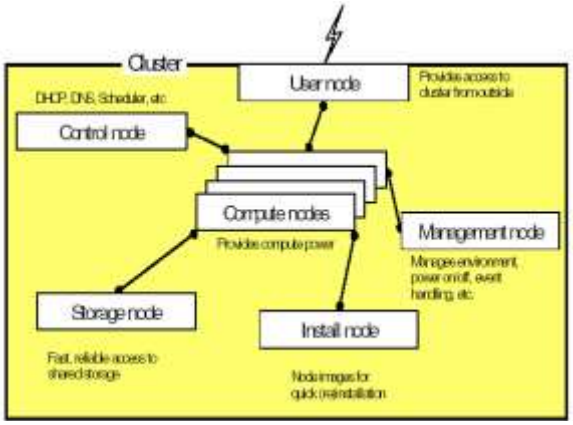
High-Performance Computing (HPC) fokus pada pengembangan superkomputer, algoritma proses paralel, dan perangkat lunak yang berhubungan dengannya. Harganya murah dan dapat diimplementasikan di sektor kumputasi paralel terdistribusi dimana komputasi dibutuhkan untuk menyelesaikan masalah besar yang saintifik (misalnya, desain produk awal, studi lingkungan, dan penelitian), menyimpan dan memproses data dalam jumlah besar (misalnya, data mining, penelitian geofisika, mesin pencari di internet, dan pengolahan citra).

7.3. Fungsi Logika dari sebuah simpul

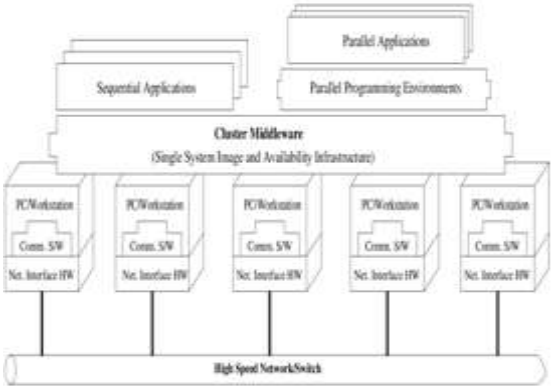
Cluster adalah dua atau lebih komputer bekerja bersama sebagai sebuah mesin untuk menyediakan layanan. Walaupun dari luar terlihat seperti hanya satu buah sistem, sistem didalamnya sebenarnya agak rumit. Gambar 4 memperlihatkan fungsi logika yang dapat diemban oleh simpul dalam *cluster*.

8. Arsitektur Komputer Cluster

Cluster adalah dua atau lebih komputer (simpul) yang berhubungan. Simpul-simpul dapat diletakkan dalam sebuah lemari atau terpisah dan dihubungkan dengan LAN. LAN *cluster* dapat terlihat sebagai sebuah sistem oleh pengguna dan aplikasi. Sitem seperti ini dapat mengambil keuntungan yaitu layanan yang cepat dan handal dengan harga yang murah. Arsitektur dari sebuah *cluster* dapat dilihat pada Gambar 5[7].



Gambar 4. Struktur Logika dari Cluster



Gambar 5. *Cluster Computer Architecture*

Simpul dari *cluster* dapat bekerja secara kolektif sebagai sumber daya komputasi yang terintegrasi. *Cluster middleware* bertanggung jawab untuk menawarkan ilusi sebagai sebuah sistem (*single system image*). Lingkungan pemrograman dapat menawarkan alat yang portabel, efisien, dan mudah digunakan untuk pengembangan aplikasi. Alat itu meliputi *message passing libraries, debuggers*, dan *profilers*. Perlu diingat bahwa *cluster* dapat digunakan untuk menjalankan program sekuensial maupun program paralel.

9. *CLUSTER* VIRTUAL

Sudah menjadi kebutuhan umum bahwa setiap produksi *server* mempunyai *server* untuk pengembangan dan pengujian, sehingga dapat dibuat percobaan tanpa takut untuk merusak hal-hal yang penting, tapi biasanya hal ini tidak mungkin dilakukan pada *cluster*. Solusinya adalah dengan membuat *cluster* virtual.

Dalam *Xen virtual machine monitor*, dapat dibuat beberapa mesin virtual, yang semuanya berjalan secara bersamaan dalam sebuah komputer, install sistem operasi yang berbeda, atau hanya konfigurasinya saja yang berbeda, dan menghubungkannya lewat kartu jaringan virtual. Xen adalah alat yang sangat bagus untuk membuat Beowulf virtual *cluster*. Sangat berguna saat digunakan untuk mengajar atau belajar tentang *cluster* atau pengujian perangkat lunak/fitur baru tanpa takut merusak cluster yang sudah ada [8].

9.1. Tipe dari *Virtual Cluster*

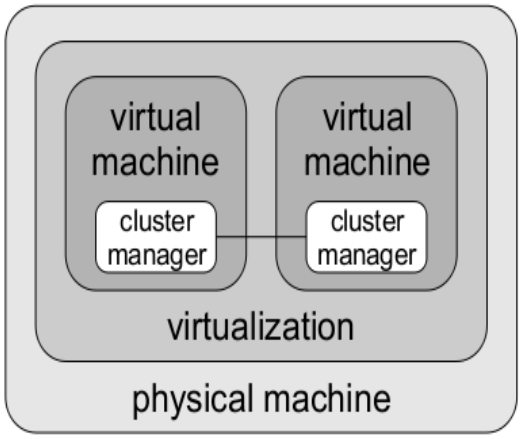
Dengan mengkombinasikan virtualisasi dan HA *clustering* sangat mungkin untuk mengambil keuntungan dari naiknya tingkat kemudahan manajemen dan menghemat konsolidasi *server* melalui virtualisasi tanpa mengurangi *uptime* dari *server* kritis [9].

9.1.1. Virtual/virtual, sebuah mesin fisik (*single physical machine*)

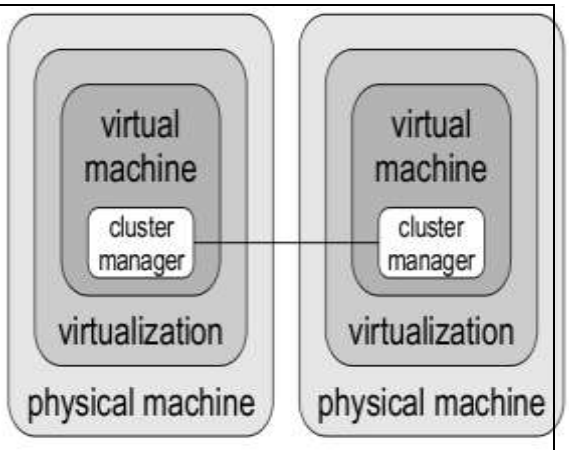
Tipe dari *cluster* virtual ini berguna untuk lingkugan pengujian tetapi tidak melindungi dari kesalahan perangkat keras dan melindungi dari *kernel panics* (*hardware* atau *paravirtualization*).

9.1.2. Virtual/virtual, *multiple physical machines*

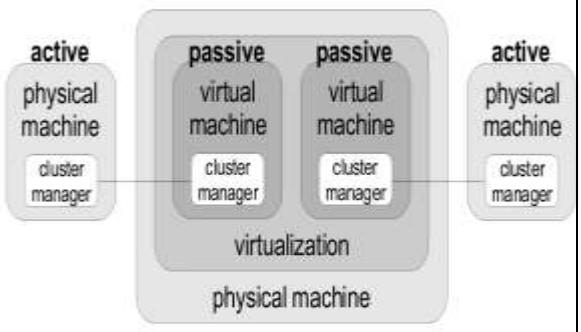
Memungkinkan untuk menjalankan aplikasi yang dalam keadaan normal tidak dapat berjalan berkali-kali dalam sebuah instan Sistem Operasi. Manajemen *cluster* menjadi lebih mudah dengan adanya virtualisasi (prosedur *backup* dan *recovery*) dan melindungi dari kesalahan perangkat keras.



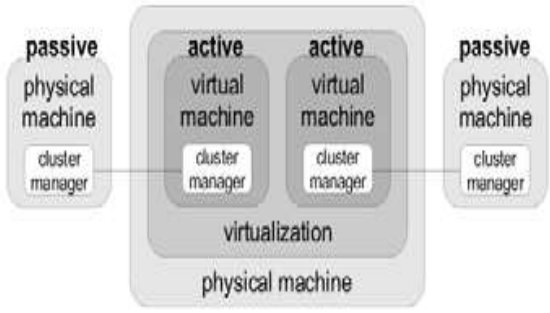
Gambar 6. *Virtual Cluster Single Physical Machine*



Gambar 7. *Virtual Cluster* dengan *virtual/virtual, multiple physical machines*



Gambar 8. *Virtual Cluster* dengan *physical machine* sebagai simpul pasif



Gambar 9. *Virtual Cluster* dengan *physical machine* sebagai simpul aktif

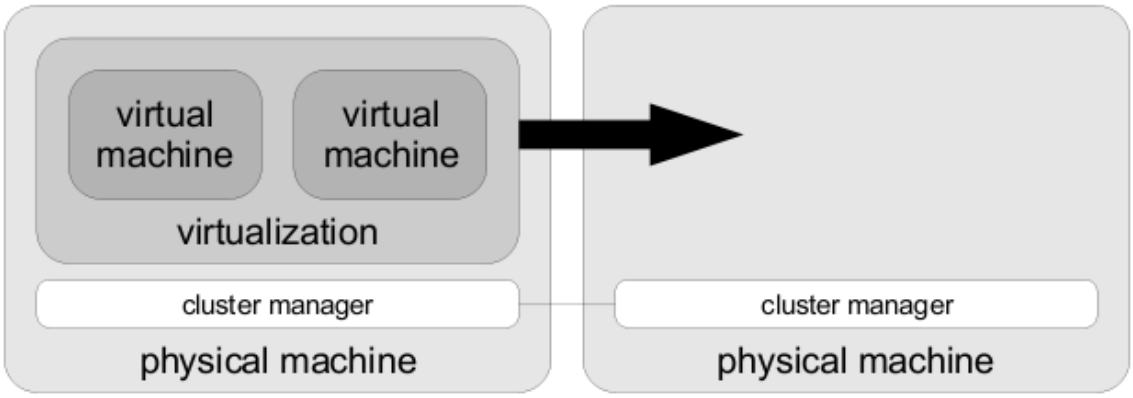
9.1.3. *Physical/virtual*

Memungkin untuk menghemat biaya karena hanya satu mesin fisik yang digunakan sebagai simpul pasif untuk beberapa *cluster*. Sangat berguna ketika *cluster* yang berbeda menjalankan sistem operasi yang berbeda.

9.1.4. *Virtual/physical*

Simpil pasif dapat dibuat lebih lemah, mirip dengan "*physical/virtual*", tetapi peran pasif dan aktif dipertukarkan. Simpul aktif harus dibuat lebih kuat.

9.1.5. Physical/physical with fail-/switchover of virtual machines



Gambar 10. *Virtual Cluster Physical/physical* dengan *fail-/switchover* dari *virtual machines*

Konfigurasi *cluster* yang sederhana dengan hanya satu buah HA. *Cluster* manajer berjalan secara langsung dalam mesin fisik. Tidak dibutuhkan perubahan yang *cluster* spesifik untuk aplikasi HA. Sangat bagus untuk virtualisasi sistem operasi (waktu *boot* dari mesin virtual).

DAFTAR PUSTAKA

[1] David Marshall, Wade A. Reynolds, and Dave McCrory, *Advance Server Virtualization,* Auerbach Publications, 2006

[2] Shannon Meier, Bill Virun, Joshua Blumert, and M. Tim Jones, *IBM Systems Virtualization: Servers, Storage, and Software*, IBM Corp. April 2008.

[3] William von Hagen, *Professional Xen® Virtualization*, Wiley Publishing, Inc., Indianapolis, Indiana, 2008

[4] Kenneth Majors and friends, *The Best Damn Server Virtualization*, Syngress Publishing, Inc., Elsevier, Inc., 2007

[5] Luis Ferreira and friends, *Linux HPC Cluster Installation*, IBM Corp. April 2001

[6] Egan Ford, Brad Elkin, Scott Denham, Benjamin Khoo, Matt Bohnsack, Chris Turcksin, and Luis Ferreira, *Building a Linux HPC Cluster with xCAT*, IBM Redbook, 2002

[7] Luis Moura e Silva and Rajkumar Buyya, *Parallel Programming Models and Paradigms*,

[8] Angel de Vicente, *Building A Virtual Cluster with Xen*, http://www.clustermonkey.net/, 2006

[9] Werner Fischer and Christoph Mitasch, *High availability clustering of virtual machines – possibilities and pitfalls,* Linuxtag 2006 Wiesbaden, Germany