

Saturday, December 26, 2009

### Mengubah Ukuran VDI VirtualBox

VDI (VeeDeeEye) adalah VirtualBox image yang sudah kompatibel dengan format OVF (Open Virtual Desktop). Rilis terakhir saat ini adalah VirtualBox 3.12 yang tersedia di platform OSX, Windows dan Linux. Seperti jika kita menggunakan mesin fisik dengan hardisk (bare metal machine), alokasi ruang untuk VDI haruslah direncanakan dengan seksama terutama jika VirtualBox dipilih sebagai virtualisasi untuk beberapa service di mesin server. Meskipun saat ini NAS/SAN dapat dibangun sendiri dengan anggaran rendah, dengan ketersediaan protokol-protokol penyimpanan yang dapat dijalankan dari jaringan: FCoE (Fiber Channel over Ethernet), iSCSI dan AoE (ATA over Ethernet), ruang hardisk di VDI tak bisa sembarang dialokasikan. Tapi jika alokasi kita kurang tepat, masih selalu ada jalan untuk mengaturnya kembali, tanpa harus instalasi ulang sistem operasi guest dari awal.

#### Tambah VDI

VDI adalah image yang berlaku seolah sebagai harddisk. Jika kebutuhan penambahan ruang penyimpanan begitu mendesak, tetapi host/induk berlaku sebagai mesin server yang tidak boleh ada downtime, maka kita memanfaatkan protokol-protokol seperti iSCSI, FCoE, atau AoE. Dimana ruang data yang sesungguhnya bisa berada dimana saja di dalam jaringan yang tersedia dari NAS/SAN Server. Jika akan ditambahkan secara fisik dari alokasi ruang data induk-nya, maka guest VirtualBox harus mati terlebih dahulu.

Misalnya awalnya secara sembarang kita tentukan saja VDI Image yang akan kita siapkan sebagai tempat instalasi sistem operasi guest Windows XP adalah 8 GB. Ketika segalanya telah siap, tiba-tiba karena satu dan lain hal piranti lunak yang perlu diinstalasikan membutuhkan ruang lebih besar: 8 GB jadi amat kurang. Maka yang dapat kita lakukan (saya menggunakan VirtualBox 3.12, versi terakhir saat ini).

Buat VDI image baru, dari File > Virtual Media Manager > Actions > New. Saya memilih menggunakan mode dynamically expanding storage. Dengan begitu misal kita tambahkan 10 GB, maka alokasi yang disiapkan memang 10 GB sebagai batas maksimum penggunaan ruang secara fisik. Bisa lebih kecil jika memang data yang mengisi image tersebut belum sampai 10 GB.

Setelah selesai tambahkan vdi baru tersebut pada storage guest. Jika vdi yang pertama sebagai primary master, maka vdi kedua dapat ditambahkan sebagai primary slave atau secondary slave. Simpan konfigurasi baru, lalu jalankan VirtualBox guest tersebut. Jika guest adalah Windows XP, maka klik-kanan menu/icon My Computer > Manage > Disk Management. Klik disk baru, partisi dan format sebagai harddisk ntfs yang baru.

#### Kloning ke VDI Baru

Pada dasarnya proses dan langkah-langkahnya hampir sama seperti menambah VDI. Hanya saja, VDI baru ini akan digunakan untuk menyalin seluruh isi VDI yang pertama. Penyalinannya menggunakan piranti bantu lain: saya gunakan clonezilla. Clonezilla adalah sistem operasi komplit dengan aplikasinya kloning harddisk-nya, yang bekerja seperti Norton Ghost. Bedanya Clonezilla berbasis Linux dan Opensource, dan dari sisi kinerja, jauh lebih sederhana dan cepat dibanding Norton Ghost. Menggunakan Norton Ghost 8 saya perlu proses yang lebih berbelit dan waktu lebih dari 30 menit untuk kloning VDI atau harddisk sebesar 160 GB, maka menggunakan Clonezilla saya hanya perlu waktu kurang dari 15 menit di dalam VirtualBox untuk menyalin isi VDI pertama ke VDI kedua (yang sudah dibuat lebih besar ukurannya).

Dengan menggunakan Clonezilla, kita perlu menjadikan CD atau file iso Clonezilla sebagai VirtualBox guest. Lalu pada setting > storage, tambahkan VDI pertama (sumber) dan VDI kedua (tujuan, yang berukuran lebih besar). Jalankan VirtualBox guest Clonezilla tersebut. Ikuti langkah-langkahnya: copy disk to local\_disk. Kita akan dituntun menentukan disk sumber (VDI pertama, disk tujuan (VDI kedua) dan jalankan proses penyalinan.

Setelah selesai, kembali ke VirtualBox guest WindowsXP, ubah setting konfigurasi storage, dengan menjadi VDI kedua (yang lebih besar) sebagai primary master dan hapus VDI pertama (yang lebih kecil). Jalankan VirtualBox guest tersebut dengan konfigurasi baru.

#### Jalan Lain

Sebagaimana telah disebut diatas, kedua cara ini membutuhkan downtime beberapa saat. Pada kondisi kritikal dimana

## **Blog Export: Meta Soliloquy Blog, <http://meta.wacana.net/>**

VirtualBox guest tidak dapat dimatikan, penambahan kapasitas harddisk tidak melalui penambahan atau mengubah ukuran VDI, tetapi melalui protokol-protokol storage area network: iSCSI, FCoE dan AoE. Secara pribadi pilihan saya saat ini adalah AoE (ATA over Ethernet), karena praktis, sederhana dan cepat menyiapkannya.

Posted by Meta Nurwidyanto in Virtualization at 07:06

Sunday, December 20, 2009

### Windows CAL Habis? Masih Ada Ubuntu (Linux)

Jika kita menggunakan Windows Server (200,2003 atau 2008) kita pasti akan berurusan dengan Client Access License (CAL). Biasanya pembelian Windows Server disertai paket CAL. Untuk Windows 2003 CAL bisa diatur berdasarkan perangkat keras (device) atau pengguna (user). CAL ini mengatur berapa jumlah user yang masih bisa mengakses server dalam waktu bersamaan, misal menggunakannya sebagai fileserver. Jadi pada dasarnya jumlah user sebanding dengan jumlah CAL yang dilisensikan. Jika di sebuah organisasi membutuhkan fileserver atau sekedar filesharing masih menggunakan Windows pasti saya sarankan pindah saja ke Samba, apalagi jika user-nya terus bertambah besar. Samba adalah fileserver opensource yang bisa dijalankan dari Linux/BSD/OpenSolaris, Samba membebaskan kita dari pembatasan dengan CAL. Tapi sering masalahnya tak semudah itu. Umumnya hal yang membuat kita tak bisa beranjak dari Windows, adalah legacy application yang sangat terikat ke sistem operasi Windows. Beberapa aplikasi tertentu yang sudah dibangun di platform dotnet dan umumnya lebih ke server-side, mengakses physical path fileserver. Ada gangguan jika kita paksa. Pada dasarnya jika jumlah CAL melebihi jumlah user yang mengakses, gangguan dari Windows bisa mulai dari pop-up (di server) notifikasi kekurangan CAL tersebut hingga user benar-benar tidak bisa mengakses fileserver. Tapi ada beberapa cara mengatasinya, secara legal. Beberapa? Ya memang ada lebih dari satu cara.

#### SSHFS

Sebelum menggunakan cara-cara alternatif, sebaiknya kita periksa dulu, apakah benar tidak ada aplikasi besar yang bisa dimigrasikan dalam waktu cepat ke platform lingkungan pengembangan non-windows. Tapi jika memang demikian situasinya, kita bisa mulai dari SSH. Windows pada dasarnya tidak menyediakan SSH service, setidaknya yang tersedia bebas dan bisa benar-benar diandalkan menangani user dan akses ke fileserver sebagaimana di Linux/Unix. Kita akan menggunakan SSH service di Windows.

Ada beberapa SSH server yang bisa digunakan di Windows: OpenSSH, WinSSHD atau FreeSSHD. Sebenarnya yang benar-benar bebas digunakan untuk enterprise sekalipun hanya OpenSSH dan saat ini paket OpenSSH untuk Windows yang masih aktif dikembangkan adalah CopSSH. CopSSH menggunakan pustaka Cygwin, yaitu implementasi POSIX diatas Windows, tiruan lingkungan UNIX di Windows. Berikut ini adalah langkah-langkah dengan menggunakan CopSSH di Windows XP.

instalasikan CopSSH  
aktivasi user (windows) di CopSSH  
mencoba koneksi ke CopSSH dari mesin lain

Pada kasus ini, saya mencobanya dengan tetap menyediakan Samba menggunakan mesin lain yaitu Ubuntu Linux dengan Samba di dalamnya. Dari mesin Ubuntu Linux ini akan dihubungkan dengan SSHFS mengakses directory Windows dan akan mendistribusikannya ke user dengan Samba. SSHFS adalah pemetaan akses ke filesystem melalui SSH (mount by SSH). SSHFS dapat menggunakan FUSE (Filesystem in User Space), jadi koneksi ke SSH Server (tujuan) tidak menggunakan root (sumber). Ada 2 macam cara koneksi SSHFS, manual menggunakan user/password atau menggunakan SSH Key. Menggunakan SSHFS dengan user/password sama saja dengan koneksi biasa dengan ssh ditambah mount folder tujuan ke folder lokal:

```
$mkdir /home/meta/remote  
$sshfs meta@192.168.1.111:/home/meta/ /home/meta/remote
```

Langkah ini berarti folder /home/meta di mesin 192.168.1.111 akan kita mount ke folder /home/meta/remote di mesin lokal kita dengan menggunakan user meta.

Koneksi dari Ubuntu Linux ke Windows lewat SSHFS hanya memerlukan satu user saja. Jadi dalam pembelian paket dasar Windows 2003 SBS, misalnya sudah termasuk 20 CAL, maka masih akan tersisa 19 CAL lagi. CAL yang tersisa dapat dipergunakan jika benar-benar diperlukan saja. Pada koneksi user ke Samba di Ubuntu Linux, tidak diperlukan lisensi user seperti CAL. Pengaturan hak akses user terhadap directory diatur oleh Samba.

#### Virtualisasi

Cara berikutnya sebenarnya hampir sama, tetapi yang digunakan adalah Virtualisasi. Dalam kasus ini akan digunakan

VirtualBox (alternatif lain yang tidak murah adalah VMWare). Menggunakan virtualisasi, posisi host-guest bisa dibolak-balik. Bisa menggunakan Windows sebagai host OS dan Linux sebagai guest OS, atau sebaliknya Linux sebagai host OS dan Window sebagai guest OS. Pada kasus saya, legacy application membuat saya memutuskan lebih baik Windows sebagai host dan Linux sebagai guest.

VirtualBox sangat mudah diinstalasikan. Yang kita perlukan hanya download VirtualBox 3.1 dari sini. Setelah instalasi VirtualBox, maka siapkan iso dari Ubuntu. VirtualBox memungkinkan instalasi sitem operasi guest dari CD/DVD atau dari file iso-nya. Saya sendiri pilih menggunakan iso file Ubuntu Server 9.10, karena sudah punya iso-nya dan tidak perlu membakarnya dulu ke CD.

Pada setting Network, pilih bridge: yang berarti host dan guest berada sejajar dalam satu network. Misal host ber-ip: 192.168.1.23 maka guest dapat kita berikan ip:192.168.1.24. Di Ubuntu (sebagai guest OS), setup ip address interface tersebut pada `/etc/network/interfaces`.

Pada model ini, Ubuntu akan membaca physical path folder dari induknya (host OS). Pada setting VirtualBox kita perlu menambahkan share folder host yang dapat dibaca guest nantinya.

Apapun sistem operasi guest, jika akan membaca share folder fisik dari induk, kita akan memerlukan VirtualBox Guest Addition. Instalasikan Virtualbox guest addition lebih dulu. Di Ubuntu untuk instalasi Virtualbox guest addition, kita memerlukan paket untuk kompilasi (compiler).

```
#apt-get install build-essential
```

Hasil instalasi Virtualbox guest addition ini adalah filesystem vboxsf. Jadi kita bisa mount share folder induk (host) di Ubuntu dari root.

```
#mkdir /mnt/drive_d  
#mount -t vboxsf DRIVE_D /mnt/drive_d
```

Ter-mount sudah filesystem induk (apapun filesystem-nya akan dikenali sebagai vboxsf). Jika kita masukkan ke dalam `/etc/fstab`, maka:

```
#echo 'DRIVE_D /mnt/drive_d vboxsf auto,rw 0 1' >> /etc/fstab
```

Masih ada satu soal lagi yang tersisa jika kita menjalankan Virtualbox dari Windows adalah bagaimana Virtualbox dapat dijalankan sebagai service di Windows, sehingga tiap kalai Windows reboot juga akan langsung menjalankan Virtualbox sebagai service di Windows. Dari googling, setidaknya saya temukan 3 cara. Dan saya memilih yang paling aman dan bersih untuk menjalankannya. Kita dapat mengikuti prosedur instalasinya disini.

Model yang akan digunakan adalah berbagi file di Windows tanpa perlu Windows user.

Selanjutnya physical path folder yang berasal dari share di Windows, dapat kita bagi ke semua user Samba. Saya tak akan membahas konfigurasi Samba (dengan LDAP atau tidak), Samba sebagai fileserver tetap dapat berfungsi dan melayani dengan baik: tanpa CAL.

### Kesimpulan

Dua cara diatas cukup mengatasi masalah saya menghemat Windows CAL, rasanya secara legal. Jika untuk kebutuhan ini spesifikasi perangkat keras atau komputer yang akan kita gunakan masih agak terbatas, misal RAM 1GB dan PC Pentium biasa, maka saya pilih cara pertama, dengan SSHFS saja. Tapi pada kasus saya, kebetulan ada Dual Core dengan RAM 4 GB, virtualisasi adalah cara yang saya pilih, karena performa lebih stabil dan meyakinkan. Ketika saya uji, perhitungan saya tidak terlalu meleset, ada peningkatan sekian puluh persen efisiensi dengan virtualisasi dibanding dengan SSHFS saja.

Monday, March 2, 2009

### VirtualBox: Desktop atau Headless?

Setelah mencoba beberapa virtualisasi: VMWare, Xen, Qemu (KVM), VirtualBox dan coLinux, akhirnya sampai juga pilihan saya ke VirtualBox. Setelah lebih dari sebulan dengan Kubuntu 8.10 Intrepid Ixex saya tak mengalami masalah apapun, Kubuntu menjadi desktop utama kerja saya. Maka sistem operasi lain, akan saya letakkan di lingkungan virtual saja. Maka jadilah WindowsXP, Windows7 Beta, gOS dan lainnya minggir ke virtualisasi karena keberadaannya hanya sampai pengujian aplikasi-aplikasi yang berjalan di atasnya saja. Sampai saat ini VirtualBox berjalan seperti yang saya inginkan di desktop saya, semua Virtual OS berjalan sempurna sebagaimana adanya, sampai tiba saatnya ke mesin-mesin server.

#### Virtual Desktop

PC Desktop saya adalah mesin standar yang tak terlalu berlebihan: Core2Duo RAM 2GB, HD 120GB, NVidia GeForce 256 MB. Itu adalah modal awal saya. Setelah Xen dan Qemu, yang paling akhir karena penasaran saya coba memang VirtualBox. Pada Kubuntu 8.10 awalnya adalah VirtualBox OSE (OpenSource Edition). Saya adalah pemula di dunia virtualisasi. Maka ketika pertukaran data dari Host ke Guest menggunakan SFTP/SCP tentu menjadi tidak lucu, atau saya harus instalasi Samba/NFS jadi lebih tidak lucu lagi. Saya tidak butuh Samba untuk ini, yang saya butuhkan ternyata adalah VirtualBox Guest Addition. Tapi ini ternyata tak cukup mudah buat saya dengan VirtualBox OSE. Saya pergi ke situs VirtualBox dan mengambil versi lain VirtualBox dari sana. Ingat, VirtualBox tetap OpenSource. Beberapa driver memang milik pihak ketiga, tapi binary dan VirtualBox tetap bebas kita gunakan. Tidak sepeser pun perlu kita bayarkan ke Sun (Sun membeli VirtualBox dari Innotek beberapa tahun lalu, untuk memperkuat portofolio virtualisasinya). Mungkin karena itu pula VMWare lantas membebaskan ESXi dan VMWare Server, karena meskipun VMWare tetap top penyedia Virtualisasi, tapi Xen, KVM dan Sun VirtualBox akan menjadi lain di mata para pengembang Virtual Appliance karena ketersediannya sebagai Opensource. Belum pula Microsoft masuk kompetisi virtual dengan VirtualPC Hyper-V. Setelah saya menginstalasikan VirtualBox-2.1.2 dengan beberapa langkah mudah:

```
sudo echo "deb http://download.virtualbox.org/virtualbox/debian intrepid non-free" >> /etc/apt/sources.list
sudo apt-get update
sudo apt-get install virtualbox-2.1
```

Guest Addition, Bridge Network, Pertukaran data Host-Guest tak jadi soal lagi. Adept Kubuntu membuat upgrade VirtualBox-2.1.2 ke VirtualBox-2.1.4 lebih mudah lagi. Setelah instalasi, akan ada vboxsetup dimana kita membutuhkan kernel-source dan compiler program di Kubuntu. Kernel Source akan langsung diambil VirtualBox ketika kita menginstalasinya, tapi barangkali compiler tertinggal untuk diambil, mudahnya kita tinggal:

```
sudo apt-get install build-essential
```

maka semua yang kita butuhkan untuk kompilasi program, dalam hal ini modifikasi kernel generic untuk virtualisasi VirtualBox menjadi transparan. Dan jika kita update kernel, maka perlu juga mengambil kernel-source-nya karena vboxsetup perlu dijalankan lagi agar VirtualBox dapat berjalan lagi dikompilasi untuk berjalan di kernel hasil update terbaru.

Pemetaan perangkat keras dari Host ke Guest tak jadi soal lagi. Pada kasus saya, gOS dan WindowsXP hanya saya alokasikan image sebesar 10 GB. Meski masih tersedia banyak ruang di Kubuntu saya dari 120 GB, tapi saya lebih memilih mencari cara lain tanpa harus membuat image yang terlalu besar. Untuk penggunaan bersama hardisk dan ruang di Home saya, hanya Guest Addition yang dibutuhkan. Instalasi standar VirtualBox-2.1 sudah termasuk Guest Addition, maka saya cuma perlu buka /home/meta di Host untuk share sebagai 'home' read-write ke Guest.

Di WindowsXP guest, saya hanya perlu jalankan dari prompt:  
net use \\vboxsrv\home x:  
Artinya 'home' di host akan terpetakan sebagai drive x di Guest.

Di gOS guest, saya hanya perlu jalankan dari prompt:  
mount -t vboxsf home /mnt/vboxhost  
Artinya 'home' di host akan terpetakan ke /mnt/vboxhost yang sudah anda siapkan lebih dahulu directory-nya.

Sound terpetakan sempurna, kita dapat memilih memetakannya dari Host dengan ALSA atau OSS. Asalkan ALSA atau OSS sudah terinstalasi baik, maka Guest akan dapat menggunakannya secara langsung. Jadi meskipun sedang bekerja di Guest, bukan berarti tidak bisa sambil memainkan mp3player atau efek bunyi di Desktop virtual Guest.

### VirtualBox Headless

Instalasi VirtualBox di desktop seperti Kubuntu 8.10 Intrepid Ibex memang mudah. Desktop Kubuntu sebagaimana desktop berbasis KDE lainnya: PCLinuxOS, Mandriva, OpenSUSE, dll saya kira juga tidak akan ada kesulitan yang berarti. Tapi bagaimana dengan server? Mesin-mesin server yang saya gunakan adalah Ubuntu 8.04 LTS, yang karena beberapa pertimbangan, saya putuskan upgrade ke Ubuntu 8.10 Server meskipun tidak punya titel LTS (Long Term Support).

Virtualisasi di mesin-mesin Server sebenarnya masih saya anggap dalam tahap pengujian. Kebutuhan di tempat kerja saya, adalah menjalankan aplikasi-aplikasi berbasis Windows yang berat di grafis dan belum ada substitusinya di Linux seperti AutoCAD dan Adobe Illustrator. Grafis generatornya adalah AutoCAD yang sudah dioptimasi dengan VBA. Lalu hasilnya dipercantik dengan Adobe Illustrator, terutama untuk diekspor ke SVG dan PDF. Memang sudah ada GIMP, Inkscape dan XARA, tapi sampai saat ini masih belum bisa memenuhi otomatisasi proses yang sudah berjalan dengan AutoCAD dan Adobe Illustrator.

Maka paling tidak dengan kekuatan mesin Server Xeon QuadCore RAM 8GB, saya bisa membuat dua atau tiga Virtual OS di Server. Setidaknya proses yang berjalan tidak harus dilakukan di desktop. Otomasi proses generator grafis dapat dilakukan di Server dan ditengok lagi oleh admin jika sudah diperoleh hasilnya. Desktop adalah desktop, kekuatan mesinnya adalah sebagaimana mesin desktop saya, dan yang lebih mengganggu, ketika proses dijalankan, terutama di AutoCAD dengan VBA-nya, yang empunya desktop harus berhenti bekerja, menunggu hasil keluar. Belum lagi aplikasi-aplikasi desktop yang umumnya jadi berjibun: Mp3Player, Yahoo Messenger, Thunderbird Email, Mozilla Firefox, Antivirus, Desktop Search Engine dst. Sumberdaya mesin desktop akan banyak terkuras untuk aplikasi-aplikasi yang bisa jadi tidak diperlukan, bisa jadi memang tidak bisa ditinggalkan karena satu dan lain hal.

Pada instalasi VirtualBox memang tidak ada masalah yang berarti di server, Proses instalasi berlangsung cepat, terutama, karena saya salin isi /var/cache/apt/archives di desktop Kubuntu 8.10 yang sudah terinstalasi VirtualBox ke mesin Server. Jadi tidak semua pustaka-pustaka terkait perlu diambil dari mirror Ubuntu. Masalah muncul ketika harus menjalankannya. Bagaimana?

Di Kubuntu, antar muka VirtualBox sangat membantu, mulai dari membuat image, mengatur setting: Network, CD/DVD/ISO, BaseMemory, VRAM dll sampai menjalankannya. Tapi bagaimana melakukan semua itu di server? VirtualBox sebenarnya juga memiliki blockquote>i>command line untuk melakukan semua yang bisa dilakukan dengan antarmuka. Baiklah, saya memilih cara mudah, untuk membuat image, setting sampai instalasi mesin Virtual-nya, saya melakukannya di desktop saya. Setelah semua siap, maka tinggal kirim VirtualBox Image VDI (VeeDeeEyes) ke mesin Server, dan menjalankannya. Tapi menjalankan dengan VBoxManage akan membuka antarmuka mesin virtual VirtualBox, dan itu di Linux memerlukan X Server terinstalasi. Bah, saya tidak merasa butuh X di server. Sebab Virtualisasi lain seperti Xen juga tak memerlukannya.

VirtualBox ternyata juga sudah menyediakan VBoxHeadless, yang dapat dijalankan dengan:

```
VBoxHeadless -s [nama_mesin_virtual] -v 4000 -a localhost  
dari desktop Kubuntu/Linux kita bisa menggunakan perintah:  
rdesktop -a 24 [ip_address_vboxserver]:4000
```

Itu berarti VirtualBox dijalankan dengan command line, dapat dibuka dan dihubungi dengan VRDP (Virtual Remote Desktop) pada port 4000. Mudah saja, ketika instalasi awal WindowsXP atau OS apapun, kita buka koneksi RDP hingga desktop hasilnya dapat dikontrol dari network melalui port 4000. Dan itulah yang saya lakukan. Setelah dijalankan dengan VBoxHeadless, maka mesin Virtual tersebut dapat digunakan siapapun yang diotorisasi dari PC Desktop manapun di jaringan melalui protocol RDP.

Jika Networking Host-Guest memilih Host Interface, berarti RDP bisa langsung ke IP Address Guest. Tapi jika NAT, maka IP address Guest seolah berada di belakang IP address Guest sehingga RDP dari PC Desktop lain di jaringan diarahkan ke IP address Host.

### Sehat dan Hemat dengan Virtualisasi

Saya rasa saat ini, virtualisasi dapat dijadikan alternatif yang sehat dan hemat untuk segerombolan mesin-mesin Rackmount Server. Sebab barangkali jika kita membutuhkan lebih dari 10 mesin, Blade layak dipikirkan, tapi jika kita hanya punya anggaran untuk 4 atau 5 Rackmount kelas menengah, bagaimanapun tetap lebih cost efisien biaya

dibandingkan Blade. Hitungan penghematan sumberdaya Blade tidak masuk sama sekali jika hanya ada anggaran seperti demikian.

Virtualisasi adalah salah satu alternatif paling masuk akal. Pilihan virtualisasinya sudah banyak, dari yang free, opensource sampai berbayar. VirtualBox sangat menarik karena juga tersedia di banyak macam OS: Linux, Windows, Solaris/OpenSolaris, dan OSX. Jadi ada keleluasaan soal pindah platform, dan image VDI pun bisa menjadi seolah media portable. Bahkan saat ini sudah ada beberapa tempat yang menyediakan VDI siap pakai yang free dan berbayar untuk kebutuhan khusus. VMWare menyediakan ESXi dan VMWare Player sebagai freeware, Xen dan KVM tersedia sebagai opensource di platform \*nix, sedang VirtualPC Hyper-V terkunci di Windows yang tentu saja karena Microsoft, sementara VirtualBox lebih leluasa platform ketersediannya dan tetap opensource. Maka saat ini, dibandingkan dengan VMWare, Xen atau KVM saya pilih VirtualBox, terutama dari sisi portabilitas.

Posted by Meta Nurwidyanto in Virtualization at 08:27

Monday, December 29, 2008

### Virtualisasi Portabel coLinux

coLinux adalah virtualisasi untuk menjalankan Linux di atas Windows. coLinux membuat Windows 2000 atau Windows XP dapat bekerja bersama Linux dalam satu mesin secara transparan. coLinux membuat Windows dan Linux bekerja beriringan, berbagi perangkat, port dan services.

coLinux membuat kita tak perlu berpikir panjang jika memerlukan virtualisasi dengan Windows sebagai host. Umumnya hal ini terjadi untuk beberapa service yang membutuhkan sumberdaya yang relatif lebih hemat jika dikerjakan di Linux misalnya: DNS, MailServer, Webproxy, SMS Gateway dll.

coLinux Debian: Virtual Debian di atas Windows

Instalasi coLinux bisa anda dapatkan di sini. Selanjutnya yang ada perlukan adalah mengunduh image-image yang diperlukan. Karena saya menginginkan Debian, maka saya mengunduh image Debian dan swap image ukuran 768 MB. Swap image ini bisa anda abaikan jika anda merasa RAM anda cukup besar dan anda tidak terlalu

memerlukan partisi swap. Selanjutnya kita lakukan langkah-langkah berikut:

- Instalasi binary coLinux yang kita unduh, letakkan di root directory misal: C:\coLinux
- Ekstrak image debian pada C:\coLinux sebagai debian4.vfs
- Ekstrak image swap ke folder yang sama
- Edit file konfigurasi example.conf sesuai kebutuhan

Berikut ini adalah contoh file konfigurasi yang saya buat sesuai kebutuhan dan kondisi PC saya.

```
kernel=vmlinux
cobd0="debian4.vfs"
cobd1="swap_768Mb"
root=/dev/cobd0
ro
initrd=initrd.gz
mem=256
ipaddress 10.0.2.15 broadcast 10.0.2.255 netmask 255.255.255.0
gateway 10.0.2.2 nameserver 10.0.2.3
eth0=slirp,,tcp:22:22/tcp:80:80
eth1=tuntap
ttys0=COM10,"BAUD=230400 PARITY=n DATA=8 STOP=1 dtr=on rts=on"
```

#### Penjelasan Konfigurasi

Konfigurasi diatas membuat kita bisa langsung menjalan coLinux Debian dengan ip address sebagaimana tersebut diatas, dimana port 22 dan 80 induk (Windows) akan diteruskan ke tamu (Debian). Mem adalah alokasi RAM yang diperuntukkan coLinux dalam mega bytes. ttys0 adalah pemetaan serial port di coLinux guest (Debian) dimana COM10 adalah port serial coLinux host-nya (Windows) Jika sudah siap, jalankan coLinux dengan:

```
colinux-daemon.exe @example.conf
```

Untuk mendaftarkan coLinux menjadi service di Windows maka jalankan:

```
colinux-daemon.exe @example.conf --install-service coLinux
```

menjalankan coLinux yang sudah menjadi Windows service:

```
net start coLinux
```

Pada coLinux yang sudah menjadi Windows service, untuk mengaksesnya dengan coLinux-console-nt.com atau coLinux-console-ftk.com.

### Portabilitas

Apa yang portabel dari coLinux? coLinux setelah instalasi pada dasarnya hanya membutuhkan ruang hardisk yang kecil 6-7Mb. Kita bisa membawa program sumber coLinux (4 MB) dengan coLinux daemon setelah instalasinya bersama dengan image yang dibutuhkan. Total kebutuhan ruang kira-kira 1,2 GB. Tergantung dari image apa yang anda akan gunakan. Jika kita menggunakan harddisk portabel seukuran 160 GB, maka bisa beberapa macam image yang kita bawa sesuai kebutuhan, dan dengan harddisk portabel sebesar itu masih bisa pula kita tambahkan lagi LiveUSB full Kubuntu persistent.

Jika kita sudah mengkonfigurasi mengkonfigurasi image Debian tersebut dengan benar dan sudah pula menginstalasikan beberapa program lain di dalamnya seperti: Apache, MySQL, Python, PHP atau aplikasi berbasis web terinstalasi dalam Apache, maka kita dapat menjalankannya di tempat lain yang kita konfigurasi dengan coLinux.

Dimana pun yang paling menjemukan kita lakukan, adalah pindah mesin. Pindah mesin tersebut bisa terjadi karena, kita bekerja di rumah, lalu kita bawa ke tempat lain (kampus, kantor, komputer orang dsb). Saat ini mobilitas orang begitu tinggi, tetapi kita tidak ingin bekerja di lingkungan orang (komputer orang) dimana segala yang kita butuhkan tidak ada, seperti aplikasi tertentu, database tertentu, bahasa pemrograman tertentu atau sistem operasi tertentu.

Masih perlu diakui pula, Windows masih dominan sebagai sistem operasi utama desktop di sekitar kita, entah itu berlisensi atau bajakan. LiveCD/LiveDVD atau bahkan LiveUSB adalah ide baik, pendekatan yang menarik karena membuat kita bisa membawa Linux kemana saja dengan mudah, bahkan tanpa instalasi.

Tapi konsekuensi tanpa instalasi adalah non volatile, artinya konfigurasi, pekerjaan, instalasi tidak dapat kita lakukan karena kita tidak dapat menyimpannya, semua harus disimpan di tempat lain (USB Flash atau Harddisk). Pada kasus LiveUSB masih mungkin dilakukan jika kita memilih model LiveUSB persistent. Maka virtualisasi adalah pilihan yang beralasan. coLinux membuat virtualisasi menjadi mudah, sederhana dan kompak.

Adopsi coLinux dalam bentuk transparan lengkap dengan Xwindow sehingga berjalan seolah sebagai perangkat lunak asli Windows adalah Ulteo.