

Monday, December 28, 2009

### Ekstensi Storage dengan iSCSI

iSCSI telah beberapa lama hadir, sebagai solusi ekstensi storage lanjutan dari SCSI dan Fibre Channel. Meski saat ini juga telah hadir solusi lain: FCoE (Fibre Channel over Ethernet) dan AoE (ATA over Ethernet), tapi iSCSI masih secara default digunakan sebagai protokol pada mesin-mesin storage server untuk mendistribusikan block storage device yang disediakan ke seluruh jaringan. iSCSI mudah diterapkan dan masih menyediakan fitur-fitur yang memanfaatkan lapisan diatas TCP/IP-nya untuk mendukung prosedur keamanan interkoneksinya. ATA over Ethernet bresifat non-routable, sedang FCoE (sebagai emulasi FC di jaringan ethernet) belum luas digunakan meski sudah diterima secara resmi sebagai kernel modul sejak Linux kernel 2.6.29. iSCSI masih belum akan ditinggalkan dalam waktu dekat, karena kepraktisan dan fitur-fiturnya sebagai protokol solusi ruang penyimpanan jaringan.

#### Mengapa iSCSI

iSCSI adalah Internet SCSI yaitu protokol pertukaran data jaringan sebagai solusi atas interkoneksi SCSI dan Fibre Channel. Jika interkoneksi SCSI memiliki keterbatasan jangkauan koneksi dan kompleksitas manajemen blok penyimpanan, maka Fibre Channel adalah hal yang benar-benar baru. Fibre Channel perlu pelatihan dan pengalaman yang berbeda dari sistem administrator kebanyakan. Selain itu, anggaran yang dibutuhkan untuk Fibre Channel juga tak bisa di satukan dengan perencanaan pengadaan mesin biasa. Pada dasarnya hal-hal khusus mengenai Fibre Channel-lah yang membuat Total Biaya Kepemilikan FC menjadi tak mudah buat organisasi bisnis kebanyakan dan kampus.

iSCSI hadir sebagai solusi atas keterbatasan SCSI dan mahalnya FC, tapi FC dan iSCSI sama-sama menghapuskan keharusan penggunaan SCSI Controller, menggantikannya dengan Host Based Controller (HBA) dan Network Interface Card (NIC - LAN Card). iSCSI dapat memanfaatkan sumberdaya jaringan yang telah ada, dan tidak terlalu diperlukan pelatihan dan pengalaman tinggi khusus bagi administrator sistem umumnya untuk dapat menangani dan mempersiapkannya. Hampir semua mesin-mesin solusi penyimpanan, sudah dilengkapi dengan kemampuan iSCSI. Jadi begitu mesin storage server tersebut ada, konfigurasi untuk terhubung ke jaringan, maka semua mesin lain yang berada di jaringan tersebut langsung bisa memanfaatkannya. iSCSI membangun lapisan protokolnya diatas TCP/IP. Tentu selain ada ongkos lapisan tersebut, banyak fitur lain yang bisa diadakan karena itu. Fitur-fitur tersebut meliputi pembatasan akses dari tingkat pengalamatan IP sampai pengguna (user). Transfer rate FC dari 2 Gbps, diperkirakan bisa sampai 4 Gbps dalam waktu dekat. Sedang iSCSI sampai 2Gbps saja. Sementara itu, dengan mengurangi ongkos TCP/IP ternyata AoE dilaporkan Coraid diatas keduanya. Pengalaman saya pribadi, pada jaringan 100Mbps biasa dan Core Switch GigE, rata-rata saya peroleh transfer rate hitungan saya, iSCSI separuh kurang dari kecepatan transfer AoE.

Pada virtualisasi, solusi iSCSI ini sangat berguna, karena baik mesin-mesin virtual (guest OS) dapat direduksi sampai hanya kebutuhan dasar prosesnya, sementara tempat penyimpanan data sepenuhnya diletakkan di media penyimpanan jaringan berbasis iSCSI. Ini akan sangat memudahkan sistem administrator memelihara sistemnya, dimana pemisahan sistem proses dan data dipusatkan pada mesin-mesin yang berbeda yang dapat diatur konfigurasi agar dapat diperoleh optimasi kinerja dengan spesifikasi khusus berdasar fungsinya.

#### NAS/SAN Hemat

Membandingkan iSCSI dengan solusi NAS lainnya seperti SAMBA/CIFS, NFS atau SSHFS sepertinya kurang tepat meski sama-sama diatas TCP/IP. iSCSI beroperasi menyediakan block storage melalui jaringan, seolah sebagai perangkat harddisk lokal yang didedikasikan pada satu mesin tertentu untuk tiap blok daripada Fileserver yang dapat digunakan bersama. Bayangkan iSCSI adalah sebuah controller SCSI dari sebuah mesin ke perangkat SCSI tertentu. Jadi tiap controller hanya akan terhubung ke mesin tertentu atau harddis/block storage tertentu saja.

Pada perkembangannya saat ini, Storage Server menjadi lebih terjangkau karena sistem operasi masa kini dengan aplikasi tertentu bisa difungsikan secara khusus menjadi NAS sekaligus SAN (hybrid). Dengan demikian biaya pengadaan Storage Server bisa lebih ditekan lagi. Jadi tidak hanya Storage Server saja yang bisa menyediakan interkoneksi iSCSI, tetapi mesin biasa bisa didesain dengan menggunakan bahkan bukan hardisk SCSI dengan konfigurasi piranti lunak RAID, menjadi solusi handal NAS/SAN.

Beberapa pengukuran (benchmarking), menunjukkan bahwa solusi iSCSI menggunakan konsumsi mesin (CPU dan RAM) lebih rendah dibandingkan NFS atau SAMBA. Ini bisa dimengerti karena iSCSI, alokasi dan fungsi kontrol dan

manajemen file dari penyedia block storage (iSCSI target) telah dipindahkan ke mesin pengguna block storage (iSCSI initiator). Saya sudah menganggap ini adalah bentuk lain dari distribusi pengelolaan data (sistem proses tersebar). Sementara NFS/SAMBA sudah menerima cluster file server (fully cluster aware), sehingga setiap perubahan dari pengguna yang terhubung akan diterima juga perubahannya oleh pengguna lain yang terhubung, sehingga ongkos bagi proses itu juga cukup tinggi. Jadi meski mirip, implementasi iSCSI berbeda dengan NFS/SAMBA.

Saat ini sudah banyak OS Appliance, atau sistem operasi yang didesain untuk difungsikan NAS/SAN secara khusus. Solusinya dari yang paling sederhana untuk SOHO/Home Network: FreeNAS (berbasis BSD), OpenFiler (berbasis Linux) sampai yang diarahkan untuk melayani kelas yang lebih besar seperti Nexenta (berbasis OpenSolaris). Meski begitu kita pun bisa membangun sendiri NAS/SAN hemat dengan distribusi Linux seperti Debian/Ubuntu/Fedora secara cepat dan mudah. Jika memerlukan GUI yang kaya dan berbasis web seperti di FreeNAS, OpenFiler atau Nexenta kita dapat memanfaatkan Webmin yang sudah menyediakan plugin iSCSI untuk mengatur iSCSI target.

### iSCSI Target di Ubuntu

Ubuntu yang kita perlukan untuk menjadi iSCSI target, bisa dimulai instalasinya sebagai JeOS (just enough operating system). Artinya adalah Ubuntu server dengan hanya sistem operasi dasar saja, sisanya akan kita tambahkan berdasar keperluan.

Konfigurasi awal yang tidak akan saya bahas, adalah saran saya untuk mengkonfigurasi mesin iSCSI target dengan RAID. Jika mesin yang ada tidak menyediakan RAID Controller, kita bisa menggunakan piranti lunak RAID, dengan setidaknya menyediakan dua harddisk ATA/SATA berkapasitas sama (misal 160 GB) yang dikonfigurasi setidaknya sebagai RAID 1 (mirroring). Tahapan instalasinya adalah:

```
Instalasi Ubuntu server
Konfigurasi ke jaringan
Hubungkan ke internet (diperlukan untuk mengambil paket-paket iSCSI
```

Setelah Ubuntu JeOS RAID1 siap, maka kita dapat mulai mengkonfigurasi iSCSI target dengan:

```
#apt-get update && apt-get install iSCSItarget
```

Pada iSCSI, sebuah block storage yang akan kita sediakan untuk seluruh isi jaringan disebut Logical Unit (LUN). Setiap LUN bisa berupa harddisk (device) atau sebuah image (raw data) yang dialokasikan sebagai harddisk (virtual device). Siapkan image yang akan kita gunakan sebagai LUN:

```
#dd if=/dev/zero of=/mnt/md2/server2F.img bs=1G count=75
#dd if=/dev/zero of=/mnt/md2/server2G.img bs=1G count=75
```

Artinya telah menyiapkan image untuk disediakan sebagai iSCSI target, masing-masing berukuran 75GB. Image-image tersebut nantinya akan berfungsi seolah harddisk baru pada iSCSI initiator. Jadi dalam keadaan belum dipartisi dan diformat. Partisi dan formatnya dapat dilakukan dari mesin iSCSI target atau iSCSI initiator, bisa dilakukan di mesin iSCSI target jika tidak sedang penuh bebannya.

```
#losetup -f /mnt/md2/server2F.img
#losetup -f /mnt/md2/server2G.img
```

Hasilnya dapat kita lihat dengan:

```
#losetup -a
```

Kira-kira akan kita dapatkan:

```
/dev/loop0: [0902]:40966 (/mnt/md2/server2F.img)
/dev/loop1: [0902]:40968 (/mnt/md2/server2G.img)
```

Harddisk image tersebut akan kita partisi dan format:

```
#cfdisk /dev/loop0
#cfdisk /dev/loop1
```

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

```
#mkfs.ext3 /dev/loop0
#mkfs.ext3 /dev/loop1
```

Jika telah selesai jagan lupa lepaskan image tersebut dengan:

```
#losetup -d /dev/loop0
#losetup -d /dev/loop1
```

SCSI Image telah siap untuk didistribusikan sebagai iSCSI target, maka kita perlu menyunting konfigurasinya di Ubuntu di `/etc/ietd.conf`. Berikut ini adalah contoh konfigurasinya:

```
Target iqn.2009-11.com.papillon:storage.lun0
  #IncomingUser meta metapassword
  #OutgoingUser meta metapassword
Lun 0 Path=/mnt/md2/server2F.img,Type=fileio
Alias LUN0
```

```
Target iqn.2009-11.com.papillon:storage.lun1
  #IncomingUser meta metapassword
  #OutgoingUser meta metapassword
Lun 1 Path=/mnt/md2/server2G.img,Type=fileio
Alias LUN1
```

Jika menggunakan autentikasi user/password anda dapat membuka marking baris deklarasi IncomingUser dan OutgoingUser. Saya lebih memilih membatasi pembukaan port pada mesin tertentu, jadi otomatis mesin lain tak akan perlu mendapatkan service ini tersedia. Sebagai catatan, iSCSI target secara default akan menggunakan port 3260, jadi kita perlu membuka port tersebut di firewall, jika di mesin iSCSI target sudah terpasang firewall. Jalankan service iSCSI target di Ubuntu:

```
#/etc/init.d/iSCSI target
```

### iSCSI Initiator di Ubuntu

Mesin lain yang akan memanfaatkan iSCSI target menggunakan iSCSI initiator. Meski iSCSI target ini bisa dimuat (mount) oleh lebih dari satu mesin, tapi akan membuat data terkorupsi. Oleh karena itu, administrator harus melakukan pembatasan agar setiap blok penyimpanan hanya akan dihubungkan ke satu mesin saja. Pembatasan lain selain alamat ip juga autentikasi, dimana hanya user tertentu (sistem administrator mesin lain) yang boleh menggunakannya.

Di Ubuntu, instalasi iSCSI initiator dengan:

```
#apt-get install open-iSCSI open-iSCSI-utils
#/etc/init.d/open-iSCSI restart
```

Sekarang kita akan mencari iSCSI volume pada mesin iSCSI target, misal alamat ip-nya: 192.168.1.253

```
# iSCSIadm -m discovery -t sendtargets -p 192.168.1.253
```

Hasilnya misalnya adalah `iqn.2009-11.com.papillon:storage.lun0` dan `iqn.2009-11.com.papillon:storage.lun1`, maka kita dapat menyambungnya dengan:

```
# iscsiadm --mode node --targetname iqn.2009-11.com.papillon:storage.lun0 --portal 192.168.1.253:3260 --login
# iscsiadm --mode node --targetname iqn.2009-11.com.papillon:storage.lun1 --portal 192.168.1.253:3260
--login
```

Akhirnya lakukan restart ulang service ini:

```
#/etc/init.d/open-iscsi restart
```

Memeriksanya, dapat dilakukan dengan melihat log:

```
# tail -f /var/log/messages
```

atau

```
#dmesg
```

Kita akan dapatkan misalnya /dev/sdc1 dan /dev/sdd1. Pada Ubuntu, buat dulu sebuah folder untuk mounting harddisk-harddisk baru tersebut, misal:

```
#mkdir /mnt/lun0
#mkdir /mnt/lun1
#mount /dev/sdc1 /mnt/lun0
#mount /dev/sdd1 /mnt/lun1
```

Kita sudah akan mendapatkan seolah 2 tambahan harddisk baru dengan kapasitas 75GB melalui protokol iSCSI.

### Windows iSCSI Target & Initiator

Di Windows bisa dikatakan hampir tak ada iSCSI Target yang bebas. Microsoft tak menyiapkan Windows agar dapat diubah menjadi Storage Server berbasis iSCSI. Tapi pihak ketiga ada yang menyediakan piranti lunak iSCSI target dan initiator. Hanya iSCSI Initiator yang disediakan oleh Microsoft untuk diunduh bebas, dan sudah menjadi bagian integral dari Windows Vista, Windows 2008 dan Windows7. Instalasi dan konfigurasi iSCSI initiator di Windows dapat disimak di sini.

### iSCSI Lebih Jauh

SCSI sampai hari ini, masih merupakan protokol yang banyak digunakan pada penyediaan fasilitas penyimpanan data jaringan. Sebab, meskipun telah hadir AoE dan FCoE yang secara umum kinerjanya jauh lebih baik dari iSCSI, tapi implementasinya belum luas. Di platform Windows, bahkan belum saya temukan aplikasi AoE target yang bisa menyamai AoE Vblade di POSIX-Compliant OS seperti Linux, BSD ataupun OpenSolaris.

Tapi iSCSI termasuk protokol-protokol yang telah dapat menekan posisi FC dengan memindahkannya ke jaringan, sehingga banyak hal yang bisa dihemat, dari pelatihan sistem administrator sampai biaya pengadaan perangkat: controller, harddisk dan mesin (PC Server). iSCSI bisa diinstalasikan tanpa downtime, penambahan dan perawatan LUN juga dapat dilakukan tanpa downtime. Ini sangat berharga bagi sistem administrator dengan mesin-mesin layanan penuh dan Service Level yang ketat soal downtime. Di Linux maupun Windows semuanya tersedia murah, mudah dan cepat. Tinggal kita yang memutuskan, seberapa bagus kinerjanya dibanding protokol lain dan dijalankan di atas sistem operasi apa. Selamat mencoba!

Posted by Meta Nurwidyanto in Opensource at 11:30

Hi it's me, I am also visiting this website regularly, this web page is actually good and the people are really sharing good thoughts.  
Anonymous on Jan 1 2012, 10:23