KONSEP DASAR PROXY

Proxy dapat dipahami sebagai pihak ketiga yang berdiri ditengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara, sedemikian sehingga pihak pertama dan pihak kedua tidak secara langsung berhubungan, akan tetapi masing-masing berhubungan dengan perantara, yaitu Proxy.

Sebuah analogi; bila seorang mahasiswa meminjam buku di perpustakaan, kadang si mahasiswa tidak diperbolehkan langsung mencari dan mengambil sendiri buku yang kita inginkan dari rak, tetapi kita meminta buku tersebut kepada petugas, tentu saja dengan memberikan nomor atau kode bukunya, dan kemudian petugas tersebut yang akan mencarikan dan mengambilkan bukunya. Dalam kasus diatas, petugas perpustakaan tersebut telah bertindak sebagai perantara atau Proxy. Petugas tersebut juga bisa memastikan dan menjaga misalnya, agar mahasiswa hanya bisa meminjam buku untuk mahasiswa, dosen boleh meminjam buku semua buku, atau masyarakat umum hanya boleh meminjam buku tertentu.

Mungkin proses tersebut menjadi lebih lama dibandingkan bila kita langsung mencari dan mengambil sendiri buku yang kita inginkan. Namun bila saja setiap kali petugas mencari dan mengambil buku untuk seseorang, si petugas juga membuat beberapa salinan dari buku tersebut sebelum memberikan bukunya kepada orang yang meminta, dan menyimpannya di atas meja pelayanan, maka bila ada orang lain yang meminta buku tertentu, sangat besar kemungkinan buku yang diminta sudah tersedia salinannya diatas meja, dan si petugas tinggal memberikannya langsung. Hasilnya adalah layanan yang lebih cepat dan sekaligus keamanan yang baik.

Analogi diatas menjelaskan konsep dan fungsi dasar dari suatu proxy dalam komunikasi jaringan komputer dan internet. Proxy server mempunyai 3 fungsi utama, yaitu,

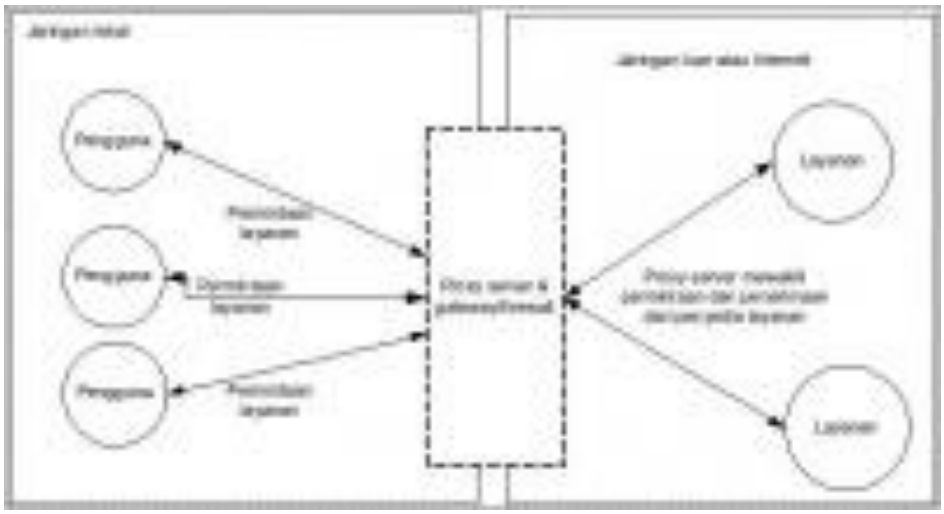
Connection Sharing
Filtering
Caching

Masing masing fungsi akan dijelaskan lebih lanjut dibawah.

Proxy dalam pengertiannya sebagai perantara, bekerja dalam berbagai jenis protokol komunikasi jaringan dan dapat berada pada level-level yang berbeda pada hirarki layer protokol komunikasi jaringan. Suatu perantara dapat saja bekerja pada layer Data-Link, layer Network dan layer Transport, maupun layer Aplikasi dalam hirarki layer komunikasi jaringan menurut OSI. Namun pengertian proxy server sebagian besar adalah untuk menunjuk suatu server yang bekerja sebagai proxy pada layer Aplikasi, meskipun juga akan dibahas mengenai proxy pada level sirkuit.

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu gateway, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. Gateway ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut akan sulit dilakukan bial tidak ada garis batas yang jelas jaringan lokal dan internet. Gateway juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya. Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh gateway secara bersama-sama (connection sharing). Dalam hal ini, gateway adalah juga sebagai proxy server, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet.

Diagram berikut menggambarkan posisi dan fungsi dari proxy server, diantara pengguna dan penyedia layanan:



PROXY, GATEWAY DAN FIREWALL

Proxy server juga biasanya menjadi satu dengan firewall server, meskipun keduanya bekerja pada layer yang berbeda. Firewall atau packet filtering yang digunakan untuk melindungi jaringan lokal dari serangan atau gangguan yang berasal dari jaringan internet bekerja pada layer network, sedangkan proxy server bekerja pada layer aplikasi. Firewall biasanya diletakkan pada router-router, untuk sehingga bisa melakukan filtering atas paket yang lewat dari dan ke jaringan-jaringan yang dihubungkan.

Karena firewall melakukan filtering berdasarkan suatu daftar aturan dan pengaturan akses tertentu, maka lebih mudah mengatur dan mengendalikan trafik dari sumber-sumber yang tidak dipercaya. Firewall juga melakukan filtering berdasarkan jenis protokol yang

digunakan (TCP,UDP,ICMP) dan port TCP atau port UDP yang digunakan oleh suatu layanan (semisal telnet atau FTP). Sehingga firewall melakukan kendali dengan metode boleh lewat atau tidak boleh lewat, sesuai dengan daftar aturan dan pengaturan akses yang dibuat. Bila suatu layanan tertentu atau alamat tertentu merupakan layanan atau alamat yang terpercaya, maka dapat diatur pada firewall agar paket dari sumber terpercaya diperbolehkan lewat.

Packet filtering pada firewall mempunyai keunggulan yaitu kecapatan yang lebih dan tidak memerlukan konfigurasi tertentu pada pengguna-pengguna yang terhubung. Namun di sisi lain dapat menimbulkan kesulitan, karena akan sangat sulit bila kita harus membuat satu daftar aturan yang banyak dan kompleks. Disamping itu, yang bisa dilakukan firewall hanya memperbolehkan atau tidak memperbolehkan suatu paket lewat berdasarkan pada alamat IP sumber atau alamat IP tujuan yang ada pada paket tersebut. Penyerang bisa melakukan memalsukan alamat IP pada paket (spoofing) emnggunakan alamat IP tertentu yang terpercaya, dan firewall akan melewatkannya. Penyerang juga dapat melakukan penyadapan paket (sniffing) dengan relatif mudah untuk mengetahui struktur alamat IP pada header paket yang lewat di jaringan.

Dalam analogi perpustakaan diatas, filtering pada firewall serupa dengan petugas perpustakaan menimpan daftar mahasiswa dan dosen yang terpercaya, dan mereka boleh langsung mengambil sendiri buku yang diinginkan dari rak. Ini bisa menghasilkan proses sirkulasi buku yang lebih cepat, namun memerlukan penanganan khusus atas daftar yang diperbolehkan tersebut. Ini juga beresiko bila ada seseorang yang menggunkan identitas palsu, sehingga seolah-olah dia adalah salah satu dari yang ada dalam daftar yang diperbolehkan.

Proxy server menggunakan cara yang berbeda. Proxy server memotong hubungan langsung antara pengguna dan layanan yang diakases (atau antara mahasiswa dan buku-buku perpustakaan dalam analogi diatas). Ini dilakukan pertama-tama dengan mengubah alamat IP, membuat pemetaan dari alamat IP jaringan lokal ke suatu alamat IP proxy, yang digunakan untuk jaringan luar atau internet. Karena hanya lamat IP proxy tersebut yang akan diketahui secara umum di internet (jaringan yang tidak terpercaya), maka pemalsuan tidak bisa dilakukan.

PENDEKATAN LAYER OSI

Karena proxy bekerja pada layer aplikasi, proxy server dapat berjalan pada banyak aplikasi antara lain HTTP Proxy atau Web Proxy untuk protokol HTTP atau Web, FTP Proxy, SMTP Proxy/POP Proxy untuk email, NNTP proxy untuk Newsgroup, RealAudio/RealVideo Proxy untuk multimedia streaming, IRC proxy untuk Internet Relay Chat (IRC), dan lain-lain. Masing-masing hanya akan menerima,meneruskan atau melakukan filter atas paket yang dihasilkan oleh layanan yang bersesuaian.

Proxy aplikasi spesifik memiliki pilihan konfigurasi yang sangat banyak. Sebagai contoh, Web Proxy dapat dikonfigurasi untuk menolak akses ke situs web tertentu pada waktu-waktu tertentu. Demikian juga proxy yang lain, misalnya dapat dikonfigurasi untuk hanya memperbolehkan download FTP dan tidak memperbolehkan upload FTP, hanya memperbolehkan pengguna tertentu yang bisa memainkan file-file RealAudio, mencegah akses ke email server sebelum tanggal tertentu, dan masih banyak lagi.

Proxy server juga sangat baik dalam hal kemampuan menyimpan catatan (logging) dari trafik jaringan, dan dapat digunakan untuk memastikan bahwa koneksi untuk jenis trafik tertentu harus selalu tersedia. Sebagai contoh, sebuah kantor mempunyai koneksi terus menerus ke Internet untuk keperluan akses Web menggunakan satu koneksi Dial-up. Proxy server dapat dikonfigurasi untuk membuka satu lagi koneksi Dial-up kedua bila ada pengguna yang melakukan download melalui FTP pada koneksi Dial-up pertama dalam waktu lama.

Sebagaimana biasa, kelemahan dari konfigurasi yang sangat fleksibel dan banyak pilihan adalah timbulnya kompleksitas. Aplikasi pada sisi pengguna seperti Web Browser atau RealAudio Player harus ikut dikonfigurasi untuk bisa mengetahui adanya proxy server dan bisa menggunakan layanannya. Bila suatu layanan baru dibuat di internet yang berjalan pada layer aplikasi, dengan menggunakan protokol baru dan port yang baru, maka harus dibuat juga proxy yang spesifik dan bersesuaian dengan layanan tersebut. Proses penambahan pengguna dan pendefinisian aturan akses pada suatu proxy juga bisa sangat rumit.

Sebagai perantara antara pengguna dan server-server di internet, proxy server bekerja dengan cara menerima permintaan layanan dari user, dan kemudian sebagai gantinya proxy server akan mewakili permintaan pengguna, ke server-server di internet yang dimaksudkan. Dengan demikian, sebenarnya proxy server hanya meneruskan permintaan pengguna ke server yang dimaksud, akan tetapi disini identitas peminta sudah berganti, bukan lagi pengguna asal, tetapi proxy server tersebut. Server-server di internet hanya akan mengeahui identitas proxy server tersebut, sebagai yang meminta, tetapi tidak akan tahu peminta sebenarnya (yaitu pengguna asalnya) karena permintaan yang sampai kepada server-server di internet bukan lagi dari pengguna asal, tetapi dari proxy server.

Bagi penggguna sendiri, proses yang terjadi pada proxy server diatas juga tidak kelihatan (transparan). Pengguna melakukan permintaan atas layanan-layanan di internet langsung kepada server-server layanan di internet. Penguna hanya mengetahui keberadaan atau alamat dari proxy server, yang diperlukan untuk melakukan konfigurasi pada sisis pengguna untuk dapat menggunakan layanan dari proxy server tersebut.

CACHING

Fungsi dasar yang ketiga dan sangat penting dari suatu proxy server adalah caching. Proxy server memiliki mekanisme penyimpanan obyek-obyek yang sudah pernah diminta

dari server-server di internet, biasa disebut caching. Karena itu, proxy server yang juga melakukan proses caching juga biasa disebut cache server.

Mekanisme caching akan menyimpan obyek-obyek yang merupakan hasil permintaan dari dari para pengguna, yang didapat dari internet. Karena proxy server bertindak sebagai perantara, maka proxy server mendapatkan obyek-obyek tersebut lebih dahulu dari sumbernya untuk ekmudian diteruskan kepada peminta yang sesungguhnya. Dalam proses tersebut, proxy server juga sekaligus menyimpan obyek-obyek tersebut untuk dirinya sendiri dalam ruang disk yang disediakan (cache).

Dengan demikian, bila suatu saat ada pengguna yang meminta suatu layanan ke internet yang mengandung obyek-obyek yang sama dengan yang sudah pernah diminta sebelumnya, yaitu yang sudah ada dalam cache, maka proxy server akan dapat langsung memberikan obyek dari cache yang diminta kepada pengguna, tanpa harus meminta ulang ke server aslinya di internet. Bila permintaan tersebut tidak dapat ditemukan dalam cache di proxy server, baru kemudian proxy server meneruskan atau memintakannya ke server aslinya di internet.

Proses caching ini juga tidak kelihatan bagi pengguna (transparan), karena bagi pengguna tidak tampak siapa sebenarnya yang yang memberikan obyek yang dimintanya, apakah proxy server yang mengambil dari cache-nya atau server asli di internet. Dari sisi pengguna, semua akan nampak sebagai balasan langsung dari internet.

Salah satu proxy yang paling banyak dibahas dan digunakan secara luas adalah HTTP proxy atau Web proxy. HTTP proxy server merupakan proxy yang berdiri diantara alokasi web pengguna misalnya web browser dan web server atau HTTP server.

Ketika pengguna membuka browser dan mengetikkan URL, maka content yang diminta URL tersebut dinamakan “Internet Object”. Pertama dia akan bertanya terlebih dahulu ke sebuah DNS (Domain Name Server). DNS akan mencari IP Address dari URL tersebut dalam databasenya dan memberi jawaban kepada browser tersebut kembali. Setelah browser mendapatkan IP Address, maka ia akan membuka hubungan ke port http web server tujuan. Web server akan mendengarkan adanya permintaan dari browser lalu memberikan content yang diminta tersebut. Seteleh browser menerima content maka hubungan dengan web server bias diputus. Content lalu ditampilkan dan disimpan didalam hardisk.

Content yang disimpan didalam hardisk biasanya disebut cache object yang natinya akan digunakan jika pengguna kembali mengunjungi site yang sama, misalnya dengan mengklik tombol back atau melihat history. Dalam kunjungan berikutnya, browser akan memeriksa validasi content yang disimpannya, validasi ini dilakukan dengan membandingkan header content yang ada pada cache object dengan yang ada pada web server, jika contect belum expired (kadaluwarsa) maka contect tadi akan ditampilkan kembali ke browser.

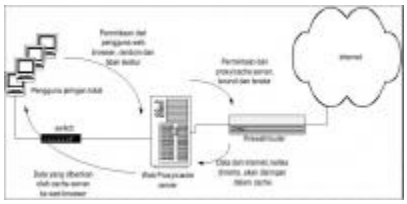
Cache object yang disimpan dalam hardisk local ini hanya bias dipakai oleh pengguna sendirian, tidak bias dibagi dengan pengguna yang lainnya, lain hal jika content tersebut disimpan pada sebuah server, dimana semua computer terhubung dengan server tersebut, maka cache object tersebut bias dipakai bersama-sama, server tersebutlah yang nantinya akan dinamakan cache server atau proxy server.

Cache server diletakkan pada titik diantara klien dan web server . Pada contoh diatas klien akan meminta content dari suatu web server ke cache server, tidak langsung ke web server tujuan. Cache server inilah yang bertanggung jawab untuk mendownload content yang diminta dan memberikannya pada klien. Content tadi disimpan pada hardsik local cache server. Lain waktu, ada klien yang meminta content yang sama, maka cache server tidak perlu mengambil langsung dari server tujuan tapi tinggal memberikan content yang sudah ada. Disinilah letak optimasi cache server tersebut.

Ada dua jenis metode caching, yaitu pasif dan aktif. Seperti telah kita ketahui, object yang disimpan bisa saja mencapai expired, untuk memeriksanya dilakukan validasi. Jika validasi ini dilakukan setelah ada permintaan dari klien, metode ini disebut pasif. Pada caching aktif, cache server mengamati object dan pola perubahannya. Misalkan pada sebuah object didapati setiap harinya berubah setiap jam 12 siang dan pengguna biasanya membacanya jam 14, maka cache server tanpa diminta klien akan memperbaharui object tersebut antara jam 12 dan 14 siang, dengan cara update otomatis ini waktu yang dibutuhkan pengguna untuk mendapatkan object yang fresh akan semakin sedikit.

Pada kondisi tertentu, kapasitas penyimpanan akan terkuras habis oleh object. Namun cache server mempunyai beberapa metode penghapusan untuk menjaga kapasitas tetap terjaga, sesuai dengan kopnfigurasi yang telah ditetapkan. Penghapusan ini didasarkan pada umur dan kepopuleran, semakin tua umur object akan tinggi prioritasnya untuk dihapus. Dan juga untuk object yang tidak popular akan lebih cepat dihapus juga.

Diagram berikut menggambarkan proses dan mekanisme caching :



HIRARKI CACHE

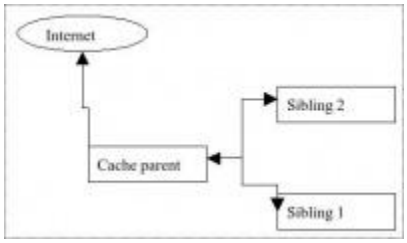
Antara cache server bias terjalin saling kerja sama. Protokol “kerja sama” ini bernama Internet Cache Protocol (ICP). Dengan ICP, sistem cache bias mempunyai hirarki. Hirarki dibentuk oleh dua jenis hubungan, yaitu parent dan sibling.

Parent :cache server yang wajib mencarikan content yang diminta oleh klien

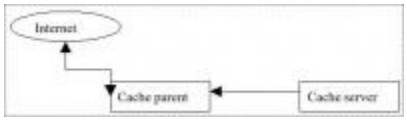
Sibling :cache server yang wajib memberikan content yang diminta jika memang tersedia. Jika tidak, sibling tidak wajib untuk mencarikannya

Dari dua hubungannya ini, sistem cache bias didesain secara bertingkat. Misalkan dalam mendesain sebuah ISP atau network kampus, anda bias mempunyai lebih dari satu cache server yang saling sibling satu dengan yang lainnya. Skenario lainnya misalkan antara cache kantor pusat dan kantor cabang, dimana kantor pusat terletak di gateway internet. Parent kantor pusat selain digunakan network lokalnya, juga dibebani trafik yang berasal dari cache server milik kantor cabang.

Untuk bermacam-macam desain cache dapat dilihat dari skema gambar berikut :



Pada gambar diatas jelas bahwa antara cache sibling yang satu dengan yang lainnya saling bertukar object, dan jika tidak ada maka cache sibling akan meminta content ke cache server, dan cache server wajib untuk memberikannya, dalam kondisi yang sesungguhnya hubungan cache sibling bias lebih dari satu.



Hubungan jenis ini bersifat ketergantungan penuh, cache child (cache server) mau tidak mau harus meminta kepada parent, dan parent pun berkewajiban untuk memenuhi permintaan child tanpa kecuali, pada kondisi ada atau tidaknya object yang diminta di dalam hardsiknya. Dan bagi child, bila parent tidak bias memenuhi permintaan, maka cache child akan memberikan pesan error pada browser klien bahwa URL maupun content yang diminta tidak dapat diambil.

TRANSPARENT PROXY

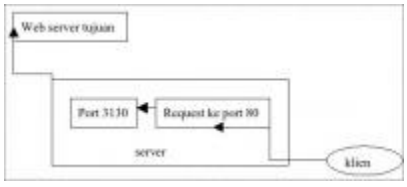
Salah satu kompleksitas dari proxy pada level aplikasi adalah bahwa pada sisi pengguna harus dilakukan konfigurasi yang spesifik untuk suatu proxy tertentu agar bisa

menggunakan layanan dari suatu proxy server. Bila diinginkan agar pengguna tidak harus melakukan konfigurasi khusus, kita bisa mengkonfigurasi proxy/cache server agar berjalan secara benar-benar transparan terhadap pengguna (transparent proxy). Biasanya cara ini memerlukan bantuan dan konfigurasi aplikasi firewall (yang bekerja pada layer network) untuk bisa membuat transparent proxy yang bekerja pada layer aplikasi.

Transparent proxy dapat berguna untuk "memaksa pengguna" menggunakan proxy/cache server, karena pengguna benar-benar tidak mengetahui tentang keberadaan proxy ini, dan apapun konfigurasi pada sisi pengguna, selama proxy server ini berada pada jalur jaringan yang pasti dilalui oleh pengguna untuk menuju ke internet, maka pengguna pasti dengan sendirinya akan "menggunakan" proxy/cache ini.

Cara membuat transparent proxy adalah dengan membelokkan arah (redirecting) dari paket-paket untuk suatu aplikasi tertentu, dengan menggunakan satu atau lebih aturan pada firewall/router. Hal ini bisa dilakukan karena setiap aplikasi berbasis TCP akan menggunakan salah satu port yang tersedia, dan firewall dapat diatur agar membelokkan paket yang menuju ke port layanan tertentu, ke arah port dari proxy yang bersesuaian.

Sebagai contoh, pada saat klient membuka hubungan HTTP (port 80) dengan suatu web server, firewall pada router yang menerima segera mengenali bahwa ada paket data yang berasal dari klien dengan nomor port 80. Disini kita juga mempunyai satu HTTP proxy server yang berjalan pada port 3130. Maka pada firewall router kita buat satu aturan yang menyatakan bahwa setiap paket yang datang dari jaringan lokal menuju ke port 80 harus dibelokkan ke arah alamat HTTP proxy server port 3130. Akibatnya, semua permintaan web dari pengguna akan masuk dan diwakili oleh HTTP proxy server diatas.



Jadi secara umum keuntungan dari metode transparent proxy itu sendiri adalah :

Kemudahan administrasi jaringan, dengan artian browser yang digunakan klien tidak harus dikonfigurasi secara khusus yang menyatakan bahwa mereka menggunakan fasilitas proxy yang bersangkutan.
Sentralisasi kontrol, dengan artian, pergantian metode bypass proxy maupun penggunaan proxy oleh klien dapat dilakukan secara terpusat.

SQUID WEB PROXY/CACHE

Salah satu contoh aplikasi proxy/cache server adalah Squid. Squid dikenal sebagai aplikasi proxy dan cache server yang handal. Pada pihak klien bekerja apliaksi browser yang meminta request http pada port 80. Browser ini setelah dikonfigurasi akan meminta content, yang selanjutnya disebut object, kepada cache server, dengan nomor port yang telah disesuaikan dengan milik server, nomor yang dipakai bukan port 80 melainkan port 8080 3130 (kebanyakan cache server menggunakan port itu sebagai standarnya).

Pada saat browser mengirimkan header permintaan, sinyal http request dikirimkan ke server. Header tersebut diterima squid dan dibaca. Dari hasil pembacaan, squid akan memparsing URL yang dibutuhkan, lali URL ini dicocokkan dengan database cache yang ada.

Database ini berupa kumpulan metadata (semacam header) dari object yang sudah ada didalam hardisk. Jika ada, object akan dikirimkan ke klien dan tercatat dalam logging bahwa klien telah mendapatkan object yang diminta. Dalam log kejadian tersebut akan dicatat sebagai TCP_HIT. Sebaliknya, jika object yang diminta ternyata tidak ada, squid akan mencarinya dari peer atau langsung ke server tujuan. Setelah mendapatkan objectnya, squid akan menyimpan object tersebut ke dalam hardisk. Selama dalam proses download object ini dinamakan "object in transit" yang sementara akan menghuni ruang memori. Dalam masa download tadi, object mulai dikirimkan ke klien dan setelah selesai, kejadian ini tercatat dalam log sebagai TCP_MISS.

Hubungan antar cache atau nantinya disebut peer itu sendiri ada dua jenis, yaitu parent dan sibling. Sibling kedudukannya saling sejajar dengan sibling lainnya, sedangkan parent adalah berada diatas sibling, dua jenis peer ini yang selanjutnya akan bergandengan membentuk jaringan hirarki cache

ICP sebagai protokol cache berperan dalam menanyakan ketersediaan object dalam cache. Dalam sebuah jaringan sebuah cache yang mempunyai sibling, akan mencoba mencari yang dibutuhkan ke peer sibling lainnya, bukan kepada parent, cache akan mengirimkan sinyal icp kepada sibling dan sibling membalasnya dengan informasi ketersediaan ada atau tidak. Bila ada, cache akan mencatatkan ICP_HIT dalam lognya. Setelah kepastian object bias diambil dari sibling, lalu cache akan mengirimkan sinyal http ke sibling untuk mengambil object yang dimaksud. Dan setelah mendapatkannya, cache akan mencatat log SIBLING_HIT.

Jika ternyata sibling tidak menyediakan object yang dicari, cache akan memintanya kepada parent. Sebagai parent, ia wajib mencarikan object yang diminta tersebut walaupun ia sendiri tidak memilikinya (TCP_MISS). Setelah object didapatkan dari server origin, object akan dikirimkan ke cache child tadi, setelah mendapatkannya cache child akan mencatatnya sebagai PARENT_HIT.

Konfigurasi, penggunaan dan metode Squid

Konfigurasi-konfigurasi mendasar squid antara lain :

http_port nomor port. Ini akan menunjukkan nomor port yang akan dipakai untuk menjalankan squid. Nomor port ini akan dipakai untuk berhubungan dengan klien dan peer.
icp_port nomor port. Ini akan menunjukkan nomor port yang akan dipakai untuk menjalankan squid. Nomor port ini akan dipakai untuk berhubungan dengan klien dan peer.
cache_peer nama_peer tipe_peer nomor_port_http nomor_port_icp option. Sintask dari cache peer ini digunakan untuk berhubungan dengan peer lain, dan peer lain yang dikoneksikan ini tipenya bergantung dari tipe peer yang telah dideklarasikan ini, bias bertipe sibling maupun bertipe parent,dan port yang digunakan untuk hubungan ICP maupun HTTP juga dideklarasikan disini, sedangakan untuk parameter option disini ada bermacam-macam salah satunya adalah default yang berarti dia adalah satu-satunya parent yang harus dihubungi (jika bertipe parent) dan proxy-only yang berarti bahwa object yang dipata dari peer tersebut tidak perlu disimpan dalam hardisk local.
Dead_peer_timeout jumlah_detik seconds. Masing-masing peer yang telah didefinisikan sebelumnya mempunyai waktu timeout sebesar yang ditentukan dalam konfigurasi ini, Jika peer tidak menjawab kiriman sinyal ICP dalam batas waktu yang telah ditentukan, peer akan dianggap tidak akan dapat dijangkau, dan cache server tidak akan mengambil object dari server yang bersangkutan dalam interval waktu tertentu.
Hierarcy_stoplist pola1 pola2 Sintaks ini digunakan untuk menyatakan apa yang harus tidak diminta dari peer, melainkan harus langsung dari web server origin, jika pola1 dan pola 2 adalah parameter cgi-bin, ?, dan lain-lain maka jika ada request URL yang mengandung karakter tersebut maka akan diambilkan langsung ke server origin.
Cache_mem jumlah_memori (dalam bytes) Sintaks ini akan menentukan batas atas jumlah memori yang digunakan untuk menyimpan antara lain : intransit object yaitu object yang dalam masa transisi antara waktu cache mendownload sampai object disampaikan ke klien, dan hot object, yaitu object yang sering diakses.
Cache_swap_low/high jumlah (dalam persen) Squid akan menghapus object yang ada didalam hardisknya jika media tersebut mulai penuh. Ukuran penuh ini yang diset pada cache_swap_low dan cache_swap_high. Bila batas swap_low telah tercapai maka squid mulai menghapus dan jika batas swap_high tercapai maka squid akan semakin sering menghapus.
Cache_dir jenis_file_sistem direktori kapasitas_cache dir_1 jumlah dir_2 Sintaks ini akan menjelaskan direktori cache yang dipakai, pertama adalah jenis file sistemnya, lalu didirektori mana cache tersebut akan disimpan, selanjutnya ukuran cache tersebut dalam MegaBytes lalu jumlah direktori level 1 dan direktori level 2 yang akan digunakan squid untuk menyimpan objectnya.

ACL (Access Control List)

Selanjutnya konfigurasi-konfigurasi lanjutan squid, selain sebagai cache server, squid yang memang bertindak sebagai "parent" untuk meminta object dari kliennya dapat juga dikonfigurasi untuk pengaturan hak akses lebih lanjut, untuk pertama kali yang dibicarakan adalah ACL (access control list), ACL sendiri terdiri dari beberapa tipe antara lain :

Src - IP Address asal yang digunakan klien
Dst - IP Address tujuan yang diminta klien
Myip - IP Address local dimana klien terhubung
srcdomain - Nama domain asal klien
dstdomain - Nama domain tujuan klien
srcdom_regex- Pencarian pola secara string dari nama domain asal klien
dstdom_regex - Pencarian pola secara string dari nama domain tujuan klien
Time - Waktu dinyatakan dalam hari dan jam
Proto - Protokol transfer (http, ftp, gopher)
Method - Metode permintaan http (get, post, connect)

Berikutnya adalah control list yang akan digunakan untuk mengatur control dari ACL, control list tersebut antara lain :

http_access - memperbolehkan acess http
icp_access - memperbolehkan peer untuk mengirimkan icp untuk menquery object
miss_access - memperbolehkan klien meminta object yang belum ada (miss) didalam cache
no_cache - object yang diminta klien tidak perlu disimpan ke hardisk
always_direct - permintaan yang ditangani langsung ke server origin
never direct - permintaan yang ditangani secara tidak langsung ke server origin.

Sebagai contoh diberikan sintaks konfigurasi ACL seperti dibawah ini :

```
#bagian ACL
ACL     localnet src 192.168.100.0/24
ACL     localkomp 127.0.0.1/255.255.255.255
ACL     isp dst 202.59.206.65/30
ACL     allsrc src 0.0.0.0/0.0.0.0
ACL     alldst dst 0.0.0.0/0
ACL     other src 10.10.11.11/32
ACL     domainku srcdomain .jatara.net
#bagian control list
http_access deny other
http_access allow localnet
http_access allow lokalkomp
http_access allow domainku
http_access deny allsrc
always_direct allow isp
```

```
always_direct deny alldst
```

Pada konsep sintaks konfigurasi squid adalah bahwa sesuatu yang telah dieksekusi pada baris yang lebih atas maka dia tidak dieksekusi lagi dibaris yang paling bawah, walaupun dalam parameter ACL yang dibawah tersebut dia juga termasuk, untuk lebih jelasnya, jika ada IP Address 192.168.100.0/24 maka IP Address yang berkisar dari 192.168.100.1 – 192.168.100.254 (ACL localnet) telah diijinkan untuk mengakses http yang ditunjukkan oleh http_access allow localnet, dan dibawahnya ada ACL allsrc yang itu adalah mencakup semua daftar IP Address dan ACL itu tidak diperbolehkan mengakses http, yaitu http_access_deny allsrc, tapi karena pada ACL localnet dia telah dieksekusi untuk sebagai IP Address yang boleh mengakses, maka walaupun dibaris bwahnya di dieksekusi lagi, itu tidak akan berpengaruh,hal-hal seperti itu digunakan untuk seorang administrator cache server untuk melakukan pengontrolan agar tidak akan terlalu detail melakukan pengaturan jika baris atas dan bawah sama-sama saling mempengaruhi.

Peering

Kembali membicarakan tentang konfigurasi peering. Maka di squid option atau parameter-parameter untuk pengaturan squid banyak sekali variasinya antara lain terdapat dalam contoh dibawah ini :

```
Cache_peer ugm.ac.id sibling 8080 3130 proxy-only
Cache_peer itb.ac.id parent  3128 3130 no-digest round-robin
Cache_peer ui.ac.id  parent  3128 3139 weight=2 no-digest
```

Untuk pengaturan diatas, tipe peer baik sibling maupun parent, nomor port untuk hubungan icp maupun http telah dijelaskan pada bab sebelumnya, disini akan dibahas tentang option yang ada yaitu proxy-only, round-robin, dan no-digest.

Pada bagian sibling cache peer itu didefinisikan sebagai proxy-only yang berarti seluruh object yang didapatkan dari sibling tidak akan disimpan ke dalam hardsik, begitu object selesai didownload maka object tersebut akan langsung diserahkan kepada klien dan object akan dihapus dari memori, option selanjutnya adalah weight, option weight adalah digunakan untuk pengaturan prioritas yang semakin tinggi nilainya maka dia adalah cache parent yang akan dihubungi terlebih dahulu, option round-robin berfungsi untuk memutar giliran parent mana yang akan diminta mencarikan object, pada kasus ini jika ada terdapat banyak parent yang tidak diberi option weight untuk prioritas maka option round-robin digunakan untuk menggilir cache yang akan dihubungi secara bergantian.

Sedang option no-digest adalah merupakan salah satu alternative squid berbicara dengan peer. Cache digest menggunakan cara mengumpulkan header masing-masing object yang telah disimpan kedalam sebuah file. File ini yang nantinya akan diforward atau didownload oleh peer dengan menggunakan protokol http. Header ini dikumpulkan dalam versi terkompres dengan rasio tinggi.

Dengan memperoleh cache-digest dari peer, squid memperoleh kejelasan status ada tidaknya object yang diminta, tanpa perlu bertanya dulu sebelumnya lewat protokol ICP, Jelas dari sini squid dapat mengoptimisasi banwitdh, terutama jika peer terletak dalam jarak logika hoop yang cukup jauh. Cache digest itu sendiri degenerate secara berkala dan besarnya tergantung dari jumlah setiap object, masing-masing object tersebut disimpan dalam header sebanyak 10 bits.

Object Cache

Pengaturan object sebuah cache server merupakan salah satu hal yang perlu diperhatikan disini. Telah diketahui sebelumnya bahwa object disimpan pada dua level cache_dir yang besar levelnya didefinisikan pada konfigurasi utama squid. Object itu sendiri berisikan content URL yang diminta klien dan disimpan dalam bentuk file binary, masing-masing object mempunyai metadata yang sebagian dari isinya disimpan didalam memori untuk memudahkan melacak dimana letak object dan apa isi dari object tersebut. Banyak sifat-sifat yang perlu diamati untuk optimasi squid ini, antara lain :

Umur object Umur obect merupakan sebuah ukuran waktu yang dihabiskan sebuah object untuk tinggal didalam hardisk cache. Umur object dibatasi oleh beberapa factor, yaitu :

metode penghapusan object object dihapus bisa melalui beberap algoritma penghapusan :

Logistic Regression : yaitu menghapus object dengan kemungkinan logistic regression terkecil. Kemungkinan logistic regression bisa diartikan sebagai besarnya kemungkinan object tersebut akan diakses diwaktu yang akan dating.
Least Recently Used : yaitu metode penghapusan object berdasarkan waktu kapan object tersebut terakhir diakses. Semakin lama (besar) waktunya, kemungkin dihapus juga akan semakin besar.
Least Frequently Used : Metode penghapusan object yang paling jarang diakses.
First In First Out : Penghapusan yang merunut metode berdasarkan waktu masuk ke dalam cache_dir, yaitu object yang paling awal masuk, berarti itu adalah object yang akan dihapus terlebih dahulu.
Random : Menghapus object secara random.

Kapasitas hardisk cache

Semakin besar kapasitas cache, berarti semakin lama umur object tersebut bisa disimpan, jika pemakaian hardisk sudah mendekati batas atas (cache_swap_high) penghapusan akan semakin sering dilakukan.

Memori

Memori dipakai squid dalam banyak hal. Salah satu contoh pemakaiannya adalah untuk disimpannya object yang popular, lazimnya disebut hot object. Jumlah hot object yang disimpan dalam memori bisa diatur dengan option cache_mem pada squid.conf

Sebenarnya yang paling memakan memori adalah metadata object, karena kebanyakan object sendiri sebenarnya disimpan dalam direktori cache_dir hardsik local. Semakin banyak kapasitas cache_dir, semakin banyak pula metadata dan semakin membebani pemakaian memori. Pada kebanyakan kasus untuk setiap 1.000.000 jumlah object, rata-rata dibutuhkan sebesar 72 MB memori untuk keseluruhan object dan 1,25 MB untuk metadata. Jumlah object ini bisa didapatkan dari besar cache_dir dibagi dengan jumlah rata-rata kapasitas object, biasanya setiap object bernilai 13 KB.

Mengingat pentingnya ketersediaan memori, penting untuk melihat sebagus apa aplikasi pengalokasian memori yang ada pada sistem operasi yang sedang bekerja. Secara default pada sistem operasi sudah tersedia rutin program untuk alokasi memori atau malloc (memory allocation). Namun pada beban yang sangat besar dan tanpa diimbangi penambahan memori yang memadai, malloc akan mencapai batas atas performansi dan kemudian mencapai status ketidakstabilan, dan squid akan menuliskan banyak pesan error pada log, misalnya seperti : “xmalloc : Unable to allocate 4096 bytes!”.

Jika ini terjadi, langkah yang dapat dilakukan adalah melakukan penambahan memori, dan langkah kedua jika ingin lebih stabil adalah menginstall library untuk rutin program malloc yang lebih baru.

PROXY SERVER LAYER NETWORK

Salah satu contoh proxy yang bekerja pada layer jaringan adalah aplikasi firewall yang menjalankan Network Address Translation (NAT). NAT selalu digunakan pada router atau gateway yang menjalankan aplikasi firewall. NAT digunakan untuk mengubah alamat IP paket TCP/IP, biasanya dari alamat IP jaringan lokal ke alamat IP publik, yang dapat dikenali di internet.

Pada suatu jaringan lokal (local Area Network), setiap komputer didalamnya menggunakan alamat IP lokal, yaitu alamat IP yang sudah disediakan untuk keperluan jaringan lokal, dan tidak akan dikenali atau diterima oleh router-router di Internet. Ketika komputer-komputer pada jaringan lokal tersebut memerlukan untuk mengakses layanan di internet, paket-paket IP yang berasal dari jaringan lokal harus diganti alamat sumbernya dengan satu alamat IP publik yang bisa diterima di internet. Disinilah proses NAT dilakukan oleh aplikasi firewall di Gateway, sehingga suatu server di internet yang

menerima permintaan dari jaringan lokal akan mengenali paket datang menggunakan alamat IP gateway, yang biasanya mempunyai satu atau lebih alamat IP publik.

Pada proses NAT ini, aplikasi firewall di gateway menyimpan satu daftar atau tabel translasi alamat berikut catatan sesi koneksi TCP/IP dari komputer-komputer lokal yang menggunakannya, sehingga proses pembaliknya bisa dilakukan, yaitu ketika paket jawaban dari internet datang, gateway dapat mengetahui tujuan sebenarnya dari paket ini, melakukan proses pembaliknya (de-NAT) dan kemudian menyampaikan paket tersebut ke komputer lokal tujuan yang sebenarnya.

PROXY SERVER PADA LEVEL SIRKUIT

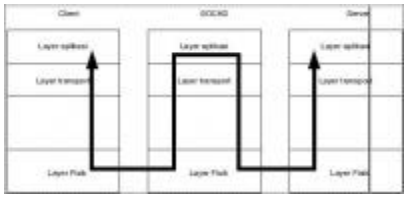
Proxy server yang bekerja pada level sirkuit dibuat untuk menyederhanakan keadaan. Proxy ini tidak bekerja pada layer aplikasi, akan tetapi bekerja sebagai "sambungan" antara layer aplikasi dan layer transport, melakukan pemantauan terhadap sesi-sesi TCP antara pengguna dan penyedia layanan atau sebaliknya. Proxy ini juga masih bertindak sebagai perantara, namun juga membangun suatu sirkuit virtual diantara layer aplikasi dan layer transport.

Dengan proxy level sirkuit, aplikasi klien pada pengguna tidak perlu dikonfigurasi untuk setiap jenis aplikasi. Sebagai contoh, dengan menggunakan Microsoft Proxy Server, sekali saja diperlukan untuk menginstall WinSock Proxy pada komputer pengguna, setelah itu aplikasi-apliakasi seperrti Windows Media Player, IRC atau telnet dapat langsung menggunakannya seperti bila terhubung langsung lke internet.

Kelemahan dari proxy level sirkuit adalah tidak bisa memeriksa isi dari paket yang dikirimkan atau diterima oleh aplikasi-aplikasi yang menggunakannya. Kelemahan ini dicoba diatasi menggunakan teknologi yang disebut SOCKS. SOCKS adalah proxy level sirkuit yang dapat digunakan untuk semua aplikasi (generik proxy) yang berbasis TCP/IP, dikembangkan sekitar tahun 1990 oleh Internet Engineering Task Force (IETF) dan sudah mencapai versi 5 (RFC 1928). SOCKS menyediakan standar yang independen dari platform yang digunakan untuk mengakses proxy level sirkuit. Salah satu kemampuan penting SOCKS versi 5 adalah tambahan proses autentikasi dan password, serta memberikan layanan proxy terhadap layanan berbasis UDP, dengan pertama-tama melakukan koneksi TCP, den kemudian menggunakannya untuk relay bagi data UDP.

SOCKS terdiri dari dua komponen, yaitu SOCKS server dan SOCKS klien. SOCKS server diimplementasikan pada leyer aplikasi, sedangkan SOCKS klien diimplementasikan diantara layer aplikasi dan layer transport. Kegunaan pokoknya adalah untuk bisa menyelenggarakan koneksi dari satu host pada satu sisi dari SOCKS server dengan host lain pada sisi yang lain dari SOCKS server, tanpa kedua host harus terhubung langsung dalam konteks TCP/IP.

Diagram berikut menggambarkan posisi SOCKS:



Ketika satu aplikasi klien ingin terhubung dengan server aplikasi, pertama-tama dia menghubungi SOCKS proxy server. Proxy inilah yang akan melakukan relay data dan menghubungkan klien dengan server. Bagi si klien, SOCKS proxy server adalah server, dan bagi server, SOCKS proxy server adalah klien. SOCKS proxy melakukan 3 tahap proses yaitu membuat permintaan koneksi, membuat sirkuit proxy-nya, dan melakukan relay data. SOCKS versi 5 menambah satu prosedur yaitu proses autentikasi pada setiap langkah diatas.

Aplikasi yang menggunakan SOCKS versi 5 sejumlah mempunyai keunggulan yaitu :

Proxy generik yang tidak tergantung pada aplikasinya (application-independent proxy). SOCKS membuat dan mengatur channel komunikasi yang digunakan untuk semua aplikasi jaringan. Adanya aplikasi baru tidak memerlukan pengembangan tambahan. Proxy layer aplikasi harus membuat software proxy baru untuk setiap aplikasi baru, dan proxy layer network dengan inspeksi penuh harus membuat cara inspeksi protokol baru.
Akses yang transparan pada jaringan dengan banyak server proxy.
kemudahan autentikasi dan metode enkripsi. Hanya menggunakan satu protokol saja untuk pembangunan channel komunikasi semua pengguna dan aplikasi, dan proses autentikasinya. Kebanyakan protokol tuneling memisahkan proses autentikasi dan proses pembangunan channel komunikasi.
kemudahan membangun aplikasi jaringan tanpa harus membuat proxy-nya.
manajemen kebijakan yang sederhana atas keamanan jaringan.

Diagram berikut menggambarkan aliran kendali model aliran kendali SOCKSv5:

Apa itu Proxy? (part 2)

Rabu, 6 Februari 2008 17:07:14 - oleh : admin

Istilah Proxy sendiri banyak dikenal / digunakan terutama di dunia / kalangan diplomatik. Secara sederhana proxy adalah seseorang / lembaga yang bertindak sebagai perantara atau atas nama dari orang lain / lembaga / negara lain. Teknik ini dikenal dengan beberapa nama yang ada di pasaran, misalnya:

Internet Connection Sharing (ICS) – istilah ini digunakan oleh Microsoft pada Windows-nya.
Proxy Server – ini biasanya berupa software tambahan yang dipasang di komputer yang bertindak sebagai perantara.
Internet Sharing Server (ISS) – biasanya berupa hardware berdiri sendiri lengkap dengan modem, hub dan software proxy di dalamnya.
Network Address Translation (NAT) – istilah lain yang digunakan untuk software proxy server.
IP Masquerade – teknik yang digunakan di software NAT / Proxy server untuk melakukan proses proxy.

Mengapa teknik proxy menjadi penting untuk share akses Internet dari sebuah LAN secara bersama-sama? Sebagai gambaran umum, dalam sebuah jaringan komputer – termasuk Internet, semua komponen jaringan di identifikasi dengan sebuah nomor (di Internet dikenal sebagai alamat Internet Protokol, alamat IP, IP address). Mengapa digunakan nomor? Karena penggunaan nomor IP akan memudahkan proses route & penyampaian data – dibandingkan kalau menggunakan nama yang tidak ada aturannya. Kira-kira secara konsep mirip dengan pola yang dipakai di nomor telepon.
Nah sialnya, (1) nomor IP ini jumlah-nya terbatas dan (2) seringkali kita tidak menginginkan orang untuk mengetahui dari komputer mana / jaringan mana kita mengakses Internet agar tidak terbuka untuk serangan para cracker dari jaringan Internet yang sifatnya publik.
Berdasarkan dua (2) alasan utama di atas, maka dikembangkan konsep private network, jaringan private atau kemudian dikenal dengan IntraNet (sebagai lawan dari Internet).

Jaringan IntraNet ini yang kemudian menjadi basis bagi jaringan di kompleks perkantoran, pabrik, kampus, Warung Internet (WARNET) dsb. Secara teknologi tidak ada bedanya antara IntraNet & Internet, beda yang significant adalah alamat IP yang digunakan. Dalam kesepakatan Internet, sebuah Intanet (jaringan private) dapat menggunakan alamat IP dalam daerah 192.168.x.x atau 10.x.x.x. IP 192.168 & 10 sama sekali tidak digunakan oleh Internet karena memang dialokasikan untuk keperluan IntraNet saja.

Proses pengkaitan ke dua jenis jaringan yang berbeda ini dilakukan secara sederhana melalui sebuah komputer atau alat yang menjalankan software proxy di atas. Jadi pada komputer yang berfungsi sebagai perantara ini, selalu akan mempunyai dua (2) interface (antar muka), biasanya satu berupa modem untuk menyambung ke jaringan Internet, dan sebuah Ethernet card untuk menyambung ke jaringan IntraNet yang sifarnya private. Untuk menghubungkan ke dua jaringan yang berbeda ini, yaitu Internet & IntraNet, perlu dilakukan translasi alamat / IP address. Teknik proxy / Network Address Translation sendiri sebetulnya sederhana dengan menggunakan tabel delapan (8) kolom, yang berisi informasi:

Alamat IP workstation yang meminta hubungan.
Port aplikasi workstation yang meminta hubungan.
Alamat IP proxy server yang menerima permintaan proxy.
Port aplikasi proxy server yang menerima permintaan proxy.
Alamat IP proxy server yang meneruskan permintaan proxy
Port aplikasi proxy server yang meneruskan permintaan proxy.
Alamat IP server tujuan.
Port aplikasi server tujuan.

Dengan cara ini, paket dengan informasi pasangan alamat IP:port dari workstation user yang meminta servis pasangan alamat IP:port server tujuan bisa diganti agar server tujuan menyangka permintaan servis tersebut datangnya dari pasangan alamat IP:port proxy server yang meneruskan permintaan proxy. Server tujuan akan mengirimkan semua data yang diminta ke pasangan alamat IP:port proxy server yang meneruskan permintaan proxy – yang kemudian meneruskannya lagi ke pasangan alamat IP:port workstation pengguna yang menggunakan alamat IP 192.168.x.x.

Jika kita lihat secara sepintas, sebetulnya teknik proxy ini merupakan teknik paling sederhana dari sebuah firewall. Kenapa? Dengan teknik proxy, server tujuan tidak mengetahui bahwa alamat komputer yang meminta data tersebut sebetulnya berada di balik proxy server & menggunakan alamat IP private 192.168.x.x.

(source from : onno w. purbo)