

BAB III RANCANGAN EKSPERIMEN

3.1 Skenario Eksperimen

Untuk menguji hipotesis awal, perlu dicari tahu parameter untuk mendapatkan kriteria beban kerja ringan dan beban kerja berat. Kemudian untuk melengkapi data perbandingan, dicari juga kriteria beban kerja antara kriteria ringan dan berat, yaitu sedang, dan parameter untuk mendapatkannya. Kombinasi kriteria beban kerja ini menghasilkan tiga variasi eksperimen, yaitu

- Beban kerja ringan.
- Beban kerja sedang.
- Beban kerja berat.

Dari tiga layanan terpilih, seperti disebut dalam 2.3, eksperimen dilakukan pada masing-masing kriteria beban kerja. Eksperimen dilakukan terhadap kombinasi jumlah layanan yang dijalankan pada server dengan kondisi satu jenis layanan dijalankan satu saja. Lalu kombinasi dilakukan terhadap satu layanan, dua layanan sekaligus, dan tiga layanan sekaligus. Kombinasi jumlah layanan ini menghasilkan tujuh variasi eksperimen, yaitu:

- Satu layanan HTTP.
- Satu layanan SMTP.
- Satu layanan CIFS.
- Layanan HTTP dan layanan SMTP.

- Layanan HTTP dan layanan CIFS.
- Layanan SMTP dan layanan CIFS.
- Layanan HTTP, layanan SMTP, dan layanan CIFS.

Kemudian, karena tujuan penelitian ini adalah membandingkan kinerja implementasi layanan yang diimplementasikan melalui server native dengan server virtual, maka ada dua variasi eksperimen dari sisi implementasi ini, yaitu:

- Implementasi server native
- Implementasi server virtual

Total variasi eksperimen adalah 42 variasi, berdasarkan kombinasi variasi eksperimen yang disebutkan di atas.

Implementasi virtual akan memisahkan masing-masing layanan ke dalam satu server virtual yang saling terpisah, dan dijalankan pada host yang sama pada saat bersamaan.

3.2 Pemilihan software untuk server

3.2.1 Linux Sebagai OS untuk Host, Guest, dan klien penguji

Distribusi GNU/Linux Fedora Core 7 (FC7) dipilih untuk eksperimen ini karena memiliki koleksi software yang cukup lengkap dan cocok untuk target uji coba eksperimen. FC7 memiliki pilihan software virtualization Xen, QEMU, dan KVM. Xen adalah virtualization dengan tipe paravirtualization, sedangkan QEMU [BEL05] dan KVM [QUM08] adalah virtualization dengan tipe full virtualization. Untuk server HTTP FC7 menyediakan pilihan Apache, software untuk server HTTP yang paling

populer di dunia, dan thttpd, sebuah server HTTP yang ringan, ringkas, dan handal. Untuk server SMTP FC7 menyediakan pilihan Sendmail dan Postfix. Untuk layanan CIFS FC7 menyediakan Samba, implementasi server CIFS yang paling populer di lingkungan UNIX.

3.2.2 Xen

Xen merupakan salah satu pemimpin virtualization untuk arsitektur x86. Dengan tipe paravirtualization, Xen diklaim mempunyai kinerja yang lebih baik dibanding dengan virtualization lain di Linux. Bahkan menurut [BAR03], kinerja virtual mesin Xen hampir sama dengan kinerja server native.

Pengamatan sekilas pada saat instalasi virtual mesin menggunakan berbagai virtualization yang tersedia di FC7 menunjukkan bahwa kinerja Xen lebih baik dibanding virtualization lain (QEMU dan KVM). Berangkat dari situ, maka Xen dipilih sebagai virtualization untuk uji coba.

3.2.3 Apache

Apache merupakan implementasi HTTP server yang memimpin di pasaran dunia. Menurut survei Netcraft [NET08], Apache merupakan web server yang paling banyak ditemukan melayani situs-situs Internet, yaitu sebanyak 49.12% dari total web server yang disurvei. Kepopuleran Apache dipicu oleh lisensinya yang Open Source, dan biasanya dipaketkan dengan distribusi berbagai sistem operasi terkemuka seperti Linux dan keluarga BSD. Selain itu banyak vendor software yang membungkus Apache dengan aplikasi mereka, menambah banyak pemakai Apache di dunia.

Apache dikembangkan sebagai proyek yang didukung secara organisasional, legal, dan keuangan oleh Apache Software Foundation. Proyek Apache tidak hanya HTTP server, tapi banyak proyek lain yang masih berhubungan teknis dengan HTTP server, termasuk Jakarta dan Tomcat yang sering dipakai vendor software untuk dibungkus bersama aplikasi buatan mereka.

Apache merupakan default Web server yang aktif dalam instalasi FC7, karena itu eksperimen ini menguji Apache sebagai HTTP server. Apache tersedia dalam lisensi Apache License, versi 2.0.

3.2.4 Sendmail

Sendmail merupakan salah satu pelopor, mungkin malah yang pertama, software yang mengimplementasikan protokol SMTP. Sendmail masih digunakan di banyak tempat dan masih dikembangkan dengan sponsor Sendmail, Inc.

Sendmail tersedia dalam FC7, dan merupakan default SMTP yang aktif dalam instalasi FC7, walau FC7 menyediakan alternatifnya yaitu Postfix. Karena itu eksperimen ini menguji Sendmail sebagai SMTP server.

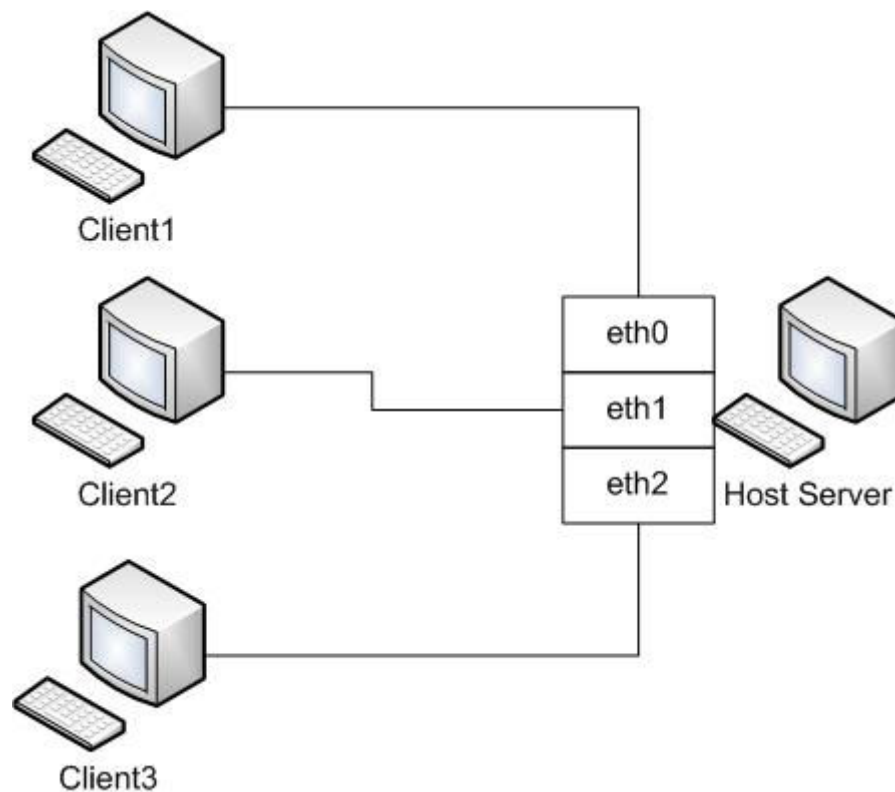
3.2.5 Samba

Samba adalah aplikasi Open Source yang menyediakan layanan file dan print untuk bermacam klien SMB/CIFS, termasuk berbagai versi keluarga Microsoft Windows. Samba telah ada sejak tahun 1992, dan merupakan salah satu pelopor dalam implementasi CIFS versi Open Source. Samba tersedia dalam lisensi GNU General Public License. Samba disertakan dalam FC7, dan merupakan standar de

facto dalam implementasi CIFS di Linux. Karena itu eksperimen ini menguji Samba sebagai layanan file-sharing.

3.2.6 Topologi Jaringan

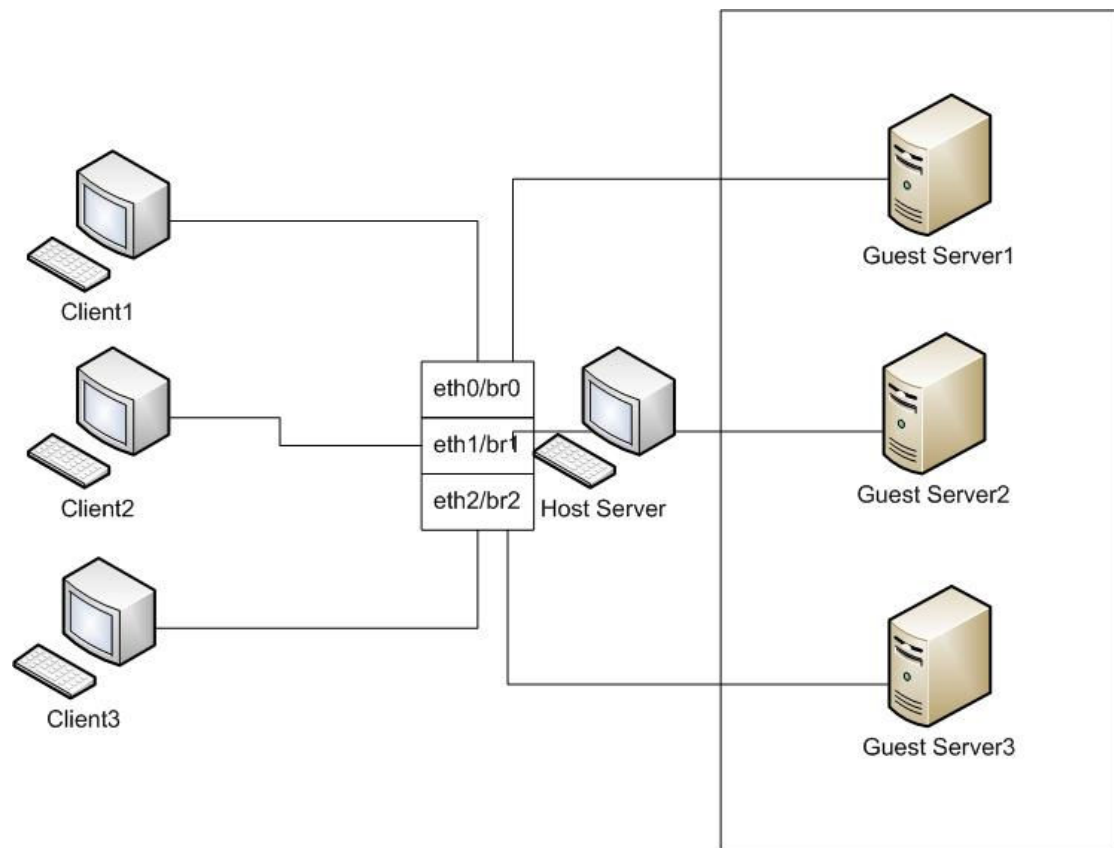
Gambar 6 memperlihatkan topologi jaringan yang digunakan untuk menghubungkan masing-masing PC klien ke server. Client1 menggunakan alamat IP 192.168.160.43 dengan network mask 255.255.255.248 dan alamat broadcast 192.168.160.47 pada network interface eth0. Client2 menggunakan alamat IP 192.168.161.43 dengan netmask 255.255.255.248 dan alamat broadcast 192.168.161.47 pada network interface eth0. Client3 menggunakan alamat IP 192.168.162.43 dengan network mask 255.255.255.248 dan alamat broadcast 192.168.162.47 pada network interface eth0.



Gambar 6 Topologi jaringan eksperimen dengan native server

Sementara itu host server memiliki tiga kartu jaringan yang diberi label eth0, eth1, dan eth2. eth0 menggunakan alamat IP 192.168.160.41 dengan network mask 255.255.255.248 dan alamat broadcast 192.168.160.47. eth1 menggunakan alamat IP 192.168.161.41 dengan network mask 255.255.255.248 dan alamat broadcast 192.168.161.47. eth2 menggunakan alamat IP 192.168.162.41 dengan network mask 255.255.255.248 dan alamat broadcast 192.168.162.47. Client1 terhubung secara langsung ke eth0 pada host server, Client2 terhubung langsung ke eth1 pada host server, dan Client3 terhubung langsung ke eth2 pada host server.

Tidak ada batasan khusus pada pengaturan topologi jaringan ini. Hanya saja karena masing-masing komputer juga dipakai untuk kebutuhan lain, maka pemberian alamat IP dimulai dari alamat IP untuk host server. Host server sudah memiliki satu alamat IP sebelum digunakan untuk eksperimen, yaitu 192.168.160.41 pada eth0. Selanjutnya, karena setiap pengujian dalam eksperimen memerlukan argumen alamat IP, maka dipilih pengalamatan yang sekiranya mudah untuk diingat dalam menentukan argumen yang dibutuhkan software benchmark. Byte ketiga pada alamat IP merupakan representasi nomor klient: 160, 161, dan 162 (0 untuk client1, 1 untuk client2, dan 2 untuk client3).



Gambar 7 Topologi jaringan dengan virtual server

Gambar 7 memperlihatkan topologi jaringan untuk eksperimen dengan virtual server. Konfigurasi jaringan pada client1, client2, dan client3 sama dengan konfigurasi pada topologi jaringan untuk native server, demikian juga konfigurasi jaringan pada host server. Untuk setiap guest server (virtual server), masing-masing menggunakan konfigurasi jaringan yang bisa diakses oleh satu klien saja, untuk memisahkan akses dari klien lainnya. Karena itu setiap server memiliki alamat network sendiri-sendiri, sesuai dengan kliennya.

Guest Server1 menggunakan alamat IP 192.168.160.42 dengan network mask 255.255.255.248 serta alamat broadcast 192.168.160.47 pada network interface eth0. Guest Server2 menggunakan alamat IP 192.168.161.42 dengan network mask 255.255.255.248 serta alamat broadcast 192.168.161.47 pada network interface eth0. Guest Server3 menggunakan alamat IP 192.168.162.42 dengan network mask

255.255.255.248 serta alamat broadcast 192.168.162.47 pada network interface eth0. Network interface eth0 pada Guest Server1 dihubungkan dengan mode BRIDGE pada network interface eth0 host server. Demikian pula network interface eth0 pada Guest Server2 dihubungkan dengan mode BRIDGE pada network interface eth1 host server, dan network interface eth0 pada Guest Server3 dihubungkan dengan mode BRIDGE pada network interface eth2 host server. Pengkaitan dengan mode BRIDGE ini dilakukan agar seluruh klien dapat mengakses pasangan virtual server masing-masing.

3.3 Software Benchmark

Banyak penelitian di lembaga lain yang menguji kinerja layanan-layanan yang diujikan dalam penelitian ini. Sebagian di antaranya menggunakan alat benchmark yang disediakan oleh SPEC (Standard Performance Evaluation Corporation, <http://www.spec.org/>), sebuah perusahaan non-profit yang dibentuk untuk membuat, memelihara, dan memajukan sekumpulan alat benchmark terstandarisasi untuk teknologi informasi, tapi dengan harga yang cukup mahal. Karena itu pencarian untuk alat benchmark diarahkan kepada alat lain yang tersedia bebas, cukup umum dipakai, dan kalau bisa tersedia dengan lisensi Open Source. Arahan ini dianggap cukup, karena penelitian akan mencari nilai independent variable dan dependent variable tanpa harus membandingkan nilainya terhadap penelitian lain. Pilihan jatuh kepada httpperf untuk menguji HTTP server, Postal untuk menguji SMTP server, dan dbench untuk menguji layanan file-sharing.

3.3.1 httpperf

httpperf [MOS98] adalah software untuk mengukur kinerja suatu web server. httpperf menyediakan berbagai fitur yang fleksibel untuk membuat bermacam beban kerja untuk mengukur kinerja web server. httpperf berfokus pada menyediakan alat yang berkinerja baik dan handal untuk memfasilitasi pengujian kinerja web server pada tingkatan mikro dan makro. httpperf dibuat oleh David Mosberger dari HP Research Labs untuk menguji kinerja suatu web server dengan simulasi berbagai kondisi yang menyerupai keadaan sebenarnya di dunia nyata. Menimbang karakteristik httpperf yang memiliki fitur untuk mensimulasikan berbagai beban kerja, maka httpperf dianggap cocok untuk eksperimen ini. httpperf menghasilkan nilai kinerja layanan dalam satuan jumlah request per detik yang dilayani web server.

3.3.2 Postal

Postal adalah sebuah program yang digunakan untuk mengetahui kinerja suatu SMTP server. Postal bekerja dengan cara mengirimkan sejumlah data acak sebagai pesan dengan alamat pengirim dan penerima yang harus sudah didefinisikan sebelumnya. Postal menghasilkan nilai layanan dalam komponen rata-rata jumlah pesan yang diterima dan diproses oleh sebuah SMTP server dalam satu menit.

3.3.3 dbench

dbench [TRI07] adalah program untuk menguji kinerja suatu filesystem yang dimuat pada sistem operasi berjalan. dbench menghasilkan beban filesystem dan

menghitung kinerja filesystem yang diuji dalam bentuk throughput dengan satuan megabyte per detik (MBps). `dbench` bekerja dengan cara memanggil bermacam fungsi IO terhadap filesystem yang dimuat, tanpa menggunakan fungsi jaringan. `dbench` mensimulasikan apa yang dikerjakan oleh `Netbench`, sebuah alat benchmark proprietary dari Ziff-Davis, tanpa membutuhkan kehadiran fisik sejumlah komputer klien yang bertindak sebagai klien penguji. `dbench` hanya membutuhkan satu komputer saja untuk bertindak sebagai klien, dan mensimulasikan sejumlah komputer klien dalam bentuk proses yang berbeda.

`dbench` digunakan dalam eksperimen ini dengan cara memuat suatu filesystem yang di-share oleh CIFS server ke dalam suatu direktori di sistem operasi lokal pada komputer klien. Selanjutnya seluruh operasi `dbench` akan diterjemahkan ke dalam fungsi-fungsi yang diberikan layanan CIFS.

3.3.4 sysstat

`sysstat` adalah software untuk mencatat berbagai informasi statistik sistem yang berjalan pada sistem operasi Linux. Dalam penelitian ini `sysstat` digunakan untuk mencatat utilisasi CPU pada server.

Program `sar` dari `sysstat` adalah program yang dipakai untuk mencatat utilisasi CPU selama proses pengujian berlangsung. Keluaran dari `sar` adalah persentase CPU dalam keadaan idle, dan untuk memperhitungkan pemakaian CPU maka rumus yang dipakai adalah

$$\text{Rata-rata utilisasi CPU} = 100 - \text{rata-rata CPU idle}$$

Contoh keluaran `sar` dapat dilihat pada Lampiran D.

3.4 Rancangan Eksperimen

Eksperimen dirancang untuk mengetahui nilai kinerja masing-masing layanan dengan berbagai parameter. Seluruh komputer (satu server dan tiga klien) dibuatkan partisi khusus untuk instalasi Linux. Kemudian seluruh komputer diinstal dengan FC7. Khusus untuk komputer server virtualization, dilengkapi dengan paket kernel yang dimodifikasi Xen agar virtualization Xen bisa dijalankan.

Untuk memudahkan eksperimen dan menyeragamkan hasilnya, dibuat beberapa script yang disebar ke masing-masing PC klien, yang spesifikasinya dibuat spesifik untuk masing-masing layanan.

3.4.1 Konfigurasi server untuk layanan (virtual dan native)

Untuk setiap server yang akan menjalankan layanan yang akan diuji, beberapa konfigurasi diatur sedemikian rupa untuk memudahkan uji coba. Perubahan dari default konfigurasi yang disediakan distribusi Linux (FC7, Mandriva, OpenSUSE) adalah:

- **Matikan SELinux (Security Enhanced Linux).** SELinux merupakan suatu framework untuk meningkatkan keamanan Linux. Bila aktif, SELinux akan menambah overhead setiap layanan yang dijalankan server, bahkan bisa jadi menggagalkan uji layanan. Bahkan bila diset dalam mode pasif (hanya memberi peringatan), overhead yang terjadi masih cukup besar. Karena itu SELinux diset dalam mode disabled (mati total).

- **Hapus semua rule iptables.** Iptables merupakan suatu fitur dari kernel Linux yang bisa memberikan fungsi firewall. Penggunaan firewall melalui iptables menambah overhead layanan yang diakses lewat jaringan. Bahkan jika layanan firewall dimatikan, masih ada beberapa rule di dalam iptables yang menambah overhead dan menyebabkan uji coba terhadap SMTP server tidak berjalan. Karena itu semua rule dalam iptables dihapus.
- **Hapus semua e-mail yang datang.** Untuk uji coba terhadap SMTP server, dibuatkan satu user account khusus sebagai alamat e-mail yang valid sebagai penerima pesan dalam SMTP. Kemudian, karena alat benchmark akan mengirim sejumlah besar pesan yang dibuat secara acak untuk disampaikan ke alamat penerima e-mail, maka dibuat sebuah script khusus di dalam user account penerima e-mail untuk seketika menghapus seluruh e-mail yang datang, dengan tujuan mencegah penuhnya filesystem, karena bisa menyebabkan server *crash*.
- **Meniadakan ketergantungan terhadap DNS.** Hal ini diperlukan karena ketergantungan protokol SMTP terhadap layanan DNS. Mengingat query DNS hanya akan menambah overhead, sementara setup DNS untuk lingkungan uji coba akan menambah waktu padahal tidak diperlukan, maka diputuskan untuk meniadakan ketergantungan terhadap DNS dalam layanan SMTP, HTTP, maupun CIFS. File `/etc/resolv.conf` diedit agar seluruh query DNS merujuk kepada DNS server yang tidak ada, yaitu dirinya sendiri tapi tanpa menjalankan satupun DNS server. Cara ini paling efisien karena konfigurasi ini

tidak membutuhkan timeout dalam proses penerimaan e-mail oleh SMTP server.

- **Kurangi overhead CPU.** Untuk mengurangi overhead CPU masing-masing server, native maupun virtual, dan host, semua layanan dijalankan dengan OS berjalan tanpa menjalankan GUI (X-Windows), yaitu dengan cara masuk ke mode init 3. Selain itu, layanan lain yang sekiranya tidak diperlukan dalam suatu uji kinerja dinon-aktifkan.

3.4.2 Konfigurasi lingkungan hardware untuk server virtual dan native

Host server memiliki total RAM sebesar 1536MB (1,5GB). Nantinya akan ada maksimum tiga server virtual yang berjalan berbarengan di atas host server. Karena target layanan adalah layanan HTTP, SMTP, dan CIFS yang tidak membutuhkan banyak memori, maka masing-masing mesin virtual diset agar menggunakan RAM sebesar 384MB. Kemudian, agar perbandingan menjadi adil, pada saat boot untuk uji kinerja server native dibatasi agar RAM yang dipakai hanya sebesar 384MB saja.

Host server memiliki prosesor Hyperthread, yang memungkinkan seolah-olah mesinnya memiliki dua prosesor. Kemudian kernel yang terpasang di host OS memiliki fitur SMP (Symmetric Multi Processor) yang memungkinkan host memanfaatkan dua prosesor itu bekerja secara simetrik. Masing-masing mesin virtual diset agar hanya memiliki satu prosesor virtual.

Untuk memungkinkan server virtual bisa diakses langsung oleh klien yang terpisah secara fisik, server virtual harus bisa mendapat akses untuk berkomunikasi lewat divais jaringan yang terpasang secara fisik pada host. Untuk itu, konfigurasi

masing-masing NIC di host diset agar menjadi jembatan bagi virtual NIC pada masing-masing server virtual.

3.4.3 Koneksi klien dengan server

Eksperimen ini menyediakan tiga komputer yang berfungsi sebagai klien. Setiap komputer klien akan menguji satu jenis layanan saja. Dalam kasus pengujian terhadap implementasi layanan pada server native, satu klien akan menguji satu layanan melalui satu hubungan fisik jaringan. Dalam kasus pengujian terhadap implementasi layanan pada server virtual, satu klien akan menguji satu layanan yang diletakkan dalam satu server virtual, juga melalui satu hubungan fisik jaringan. Karena masing-masing komputer klien memiliki hubungan fisik tersendiri ke server, maka setiap klien memiliki akses penuh terhadap bandwidth yang dimiliki, yaitu sebesar 100 Mbps.

3.4.4 Pencarian parameter untuk membedakan beban kerja

Dalam bagian 3.1 disebutkan bahwa eksperimen akan dilakukan terhadap masing-masing layanan dengan variasi pada tingkat beban kerja. Tingkat beban kerja yang ditentukan ada tiga, yaitu beban kerja ringan, beban kerja sedang, dan beban kerja berat. Kriteria umumnya juga sudah disebutkan. Untuk mencapai kriteria itu, maka perlu dicari spesifikasi teknis yang dapat memenuhinya. Untuk itu sejumlah kecil percobaan dilakukan terhadap masing-masing layanan yang dijalankan pada lingkungan server native dan masing-masing software benchmark.

Parameter beban kerja layanan HTTP

Perbedaan parameter untuk menguji variasi beban kerja layanan HTTP terletak pada rentang waktu antara satu request dengan request lainnya. Spesifikasi untuk mendapatkan perbedaan beban kerja ini adalah:

- Beban kerja ringan: alat benchmark mensimulasikan klien HTTP mengirim 1000 request, masing-masing diikuti dengan 100 child request, dengan rentang waktu antara 0.1 - 0.2 detik. Spesifikasi ini mencatat tingkat utilisasi CPU sebesar 4% pada server HTTP.
- Beban kerja sedang: alat benchmark mensimulasikan klien HTTP mengirim 1000 request, masing-masing diikuti dengan 100 child request, dengan rentang waktu antara 0.01 - 0.02 detik. Spesifikasi ini mencatat tingkat utilisasi CPU sebesar 42% pada server HTTP.
- Beban kerja berat: alat benchmark mensimulasikan klien HTTP mengirim 1000 request, masing-masing diikuti dengan 100 child request, dengan rentang waktu antara 0.001 - 0.002 detik. Spesifikasi ini mencatat tingkat utilisasi CPU sebesar 78% pada server HTTP.

Parameter beban kerja layanan SMTP

Perbedaan parameter untuk menguji variasi beban kerja layanan SMTP terletak pada jumlah thread yang mensimulasikan jumlah klien yang mengakses server SMTP. Spesifikasi untuk perbedaan parameter ini adalah:

- Beban kerja ringan: alat benchmark mensimulasikan klien SMTP dengan cara membuat 50 thread yang masing-masing koneksi SMTP ke server, masing-masing mengirim 5 pesan dengan ukuran 5kb. Spesifikasi ini menghasilkan tingkat utilisasi CPU sebesar 13%.

- Beban kerja sedang: alat benchmark mensimulasikan klien SMTP dengan cara membuat 150 thread yang masing-masing koneksi SMTP ke server, masing-masing mengirim 5 pesan dengan ukuran 5kb. Spesifikasi ini menghasilkan tingkat utilisasi CPU sebesar 40%.
- Beban kerja berat: alat benchmark mensimulasikan klien SMTP dengan cara membuat 50 thread yang masing-masing koneksi SMTP ke server, masing-masing mengirim 5 pesan dengan ukuran 5kb. Spesifikasi ini menghasilkan tingkat utilisasi CPU sebesar 70%.

Parameter beban kerja layanan CIFS

Khusus untuk layanan CIFS, mengingat proses yang terlibat dalam layanan ini umumnya membutuhkan transfer data/file dalam ukuran yang besar, maka tingkat utilisasi CPU kemungkinan tidak akan bisa setinggi layanan lain karena jumlah transfer dibatasi oleh bandwidth jaringan. Batasan ini juga membatasi jumlah proses yang dapat dilayani server CIFS dalam satu waktu, sehingga utilisasi CPU untuk layanan CIFS tidak bisa setinggi tingkat utilisasi layanan HTTP atau layanan SMTP pada beban kerja berat.

Karena software benchmark yang dipakai menghasilkan data dalam bentuk throughput (MBps), kriteria beban kerja berat dicari berdasarkan jumlah proses/klien minimum yang menghasilkan throughput terbesar. Menimbang bahwa bandwidth jaringan yang dipakai adalah 100 Mbps, maka perkiraan throughput terbesar berada di kisaran nilai sekitar 10 MBps. Dari percobaan awal didapat informasi bahwa nilai throughput tersebut didapat pada jumlah proses/klien sembilan atau lebih besar. Hanya saja pada eksperimen utama, jika jumlah proses lebih dari 10 pada kondisi

layanan yang diuji pada server lebih dari satu di saat bersamaan, akan menyebabkan proses benchmarking layanan CIFS gagal.

Perbedaan parameter untuk menguji variasi beban kerja layanan CIFS terletak pada jumlah thread yang mensimulasikan jumlah klien yang mengakses server SMTP. Untuk layanan ini didapat parameter sebagai berikut:

- Beban kerja ringan: alat benchmark mensimulasikan klien mensimulasikan proses berbagai transaksi Input Output (IO) dari satu klien. Spesifikasi ini menghasilkan tingkat utilisasi CPU sebesar 9%.
- Beban kerja sedang: alat benchmark mensimulasikan klien mensimulasikan proses berbagai transaksi Input Output (IO) dari empat klien. Spesifikasi ini menghasilkan tingkat utilisasi CPU sebesar 17%.
- Beban kerja berat: alat benchmark mensimulasikan klien mensimulasikan proses berbagai transaksi Input Output (IO) dari sepuluh klien. Spesifikasi ini menghasilkan tingkat utilisasi CPU sebesar 25%.