

ANALISA PERFORMA SOKET CONCURRENT FORK SERVER DAN CONCURRENT PRE-FORK SERVER PADA IPv4 DAN IPv6

Joni Nurdianto¹, Lutfi Ali Muharom², Eko Fajar Yanuarsa³
Jurusan Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Jember
Email : joni.nurdianto@gmail.com¹

ABSTRAK

Client-Server memiliki beberapa alternatif desain, terutama Concurrent Server. Pemilihan desain server yang benar membuat efisiensi dalam penggunaan waktu dan pengendalian proses. Sebuah server memiliki kontrol proses lebih dari klien sebagai server harus merespon multi-permintaan dan multi-processing di waktu yang sama dari platform Client yang berbeda seperti IPv4 atau IPv6. Penelitian ini menganalisis kinerja IPv4 dan IPv6 pada Desain Concurrent Fork Server dan Concurrent Pre-Fork Server. Percobaan untuk menganalisis CPU time termasuk kernel time untuk setiap server dilakukan pada soket TCP menggunakan beberapa teknik, termasuk menugaskan 20 dengan koneksi 20-100 koneksi berturut-turut untuk setiap klien pada setiap tes untuk setiap server pada jaringan IPv4 dan IPv6.

Kata kunci : IPv4, IPv6, Concurrent Fork Server, Concurrent Pre-Fork Server

I. PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya teknologi pada zaman sekarang ini, terutama teknologi komunikasi berbasis *Internet Protocol* (IP). Sehingga memungkinkan banyaknya penggunaan *Internet Protocol Version 4* (IPv4), yaitu teknologi yang digunakan saat ini akan mengalami masalah pada pengalaman yang semakin terbatas. Karena keterbatasan jumlah pengalaman pada IPv4 ini, sehingga dibutuhkan satu standar baru pada *routing protocol* yang mampu mengakomodasi jumlah pengalaman yang lebih banyak. Berangkat dari masalah ini, *Internet Protocol Version 6* (IPv6) kemudian dikembangkan dan dijadikan standar baru yang kelak akan

mengakomodasi pengalaman yang lebih banyak.

IPv4 dan IPv6 memiliki struktur yang berbeda, misalnya format *header*. Beberapa bidang Format header IPv4 tidak lagi tersedia atau digantikan dalam header IPv6, seperti 6-bit *DSCP field* dan 2-bit *ECN field* menggantikan *historical 8-bit traffic class field*, panjang 16 bit *payload* tidak disertakan di IPv6, dll. Hal ini bertujuan untuk meningkatkan kecepatan data *forwarding* dan mengurangi *delay*.

Selain masalah pada IPv4 dan IPv6, desain server juga penting dalam mengelola transfer data antara beberapa *dedicated server* dan beberapa *client* terutama dalam kasus *sending* dan *requesting queries* dari beberapa *client* dan pada saat yang sama *client* meminta beberapa tanggapan juga. *Client-Server*

memiliki beberapa alternatif desain terutama *Concurrent Server*, dalam penggunaan waktu dan pengendalian proses *server* tidak efisien karena dihasilkan dari memilih desain *server* yang tidak benar. Sebuah *server* memiliki control proses lebih dari *client*, sebagai *server* harus menanggapi multi-queri dan multi-prosesing diwaktu yang sama dari platform *client* yang berbeda seperti IPv4 dan IPv6.

1.1 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan yang akan diangkat dalam Tugas Akhir ini yaitu

- a. Bagaimana membandingkan kinerja dari 2 desain *Concurrent server* yaitu *Concurrent Fork Server* dan *Concurrent Pre-Fork Server* terhadap IPv4 dan IPv6.
- b. Bagaimana menganalisa *CPU time* dan *Kernel time* dari masing-masing *server*.

1.2 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

- a. Menetapkan 25 *client* dengan koneksi 20-100 koneksi *consecutive* untuk setiap *client* untuk setiap tes untuk setiap *server*.
- b. Menetapkan 25 *client* dengan 10 koneksi dan 1000 *byte* untuk setiap *child*.

- c. Menetapkan 25 *client* dan 100 koneksi untuk setiap *server*, dalam membandingkan kinerja server pada IPv4 dan IPv6.
- d. Menggunakan Sistem Operasi berbasis Linux Ubuntu
- e. Tidak membahas pengalamatan IP address secara spesifik

1.3 Tujuan Penulisan

Tujuan penulisan tugas akhir ini adalah mengetahui performa Soket *Concurrent Fork Server* dan *Concurrent Pre-Fork Server* pada IPv4 dan IPv6

1.4 Manfaat Penulisan

Dari hasil penelitian ini diperoleh manfaat yaitu mengetahui performa *Concurrent Fork Server* dan *Concurrent Pre-Fork Server* pada IPv4 dan IPv6

II. Tinjauan Pustaka

2.1 *Concurrent Fork Server*

Concurrent server berarti *server* yang mempunyai kemampuan melayani beberapa *client* pada suatu waktu. *Concurrent-fork server* menggunakan fungsi *fork* dalam menangani banyak *client*. *Server* menciptakan 1 *child* untuk setiap *client* sehingga memiliki 1 *client* proses. Ketika *server receive* dan *accept* sambungan *client*, maka *server* akan membuat *child* (*copy-an* dari *server*) dan memungkinkan *child* menangani *client*. *fork* berfungsi

membuat proses baru yang terpisah untuk setiap *client* tunggal. *Fork* berfungsi membuat proses baru yang terpisah dari setiap *client* tunggal, dan membuat proses menjadi *parent* dan *child*. *Child* yang mengacu pada proses baru memiliki *frame* atau model yang sama (mengacu pada proses induk). Ketika *server* menerima dan menerima koneksi klien, *fork* tersebut akan mengcopy dirinya sendiri dan memungkinkan *child* menangani klien.

2.2 Concurrent Pre-Fork Server

Memiliki ide serupa dengan *concurrent-fork server*. Perbedaannya pada cara memberikan *child* untuk setiap *client*. Dalam *concurrent-fork server* sebuah proses *child* dibuat setelah server permintaan *client* mencapai server. Sedangkan *Pre-Fork Server*, ada proses kontrol tunggal yang bertanggung jawab untuk memulai *child proses*, karena *child proses* dibuat sebelum permintaan *client* untuk menghubungkan server. Dengan menggunakan tipe ini *client* tidak perlu menunggu untuk dilayani oleh *child proses* karena *child proses* akan siap sebelum *client* bisa terhubung ke server.

2.3 IPv4

Alamat IP versi 4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang

menggunakan protokol IP versi 4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalami hingga 4 miliar host komputer atau lebih tepatnya 4.294.967.296 host di seluruh dunia, jumlah host tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4 (karena terdapat 4 oktet) sehingga nilai maksimal dari alamat IP versi 4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai host yang dapat ditampung adalah $256 \times 256 \times 256 \times 256 = 4.294.967.296$ host.

2.4 IPv6

Setiap komputer yang terhubung ke internet setidaknya harus memiliki sebuah IP *address* pada setiap *interface*-nya dan IP *address* sendiri harus unik karena tidak boleh ada komputer / server / perangkat *network* lainnya menggunakan IP *address* yang sama di internet.

Dalam perkembangannya dunia internet sampai dengan saat ini berkembang cukup drastis sehingga untuk pemakaian IPv4 dirasa kurang memfasilitasi kebutuhan berinternet masyarakat dunia. Untuk itulah IETF (*Internet Engineering Task Force*) mendesain alamat IP baru yang disebut IPv6 (*internet Protocol Version 6*). Protocol IPv6 dikembangkan untuk menangani masalah banyaknya alamat yang tidak tertangani oleh IPv4, karena dunia internet tiap tahun berkembang sehingga perlu diganti dengan *Protocol* IPv6. IPv6 memiliki panjang total 128-bit, dan secara

teoritis dapat mengalami hingga $2^{128}=3,4 \times 10^{38}$ host komputer di seluruh dunia.

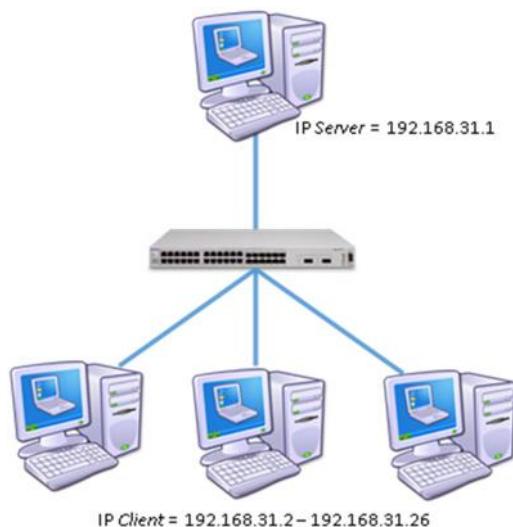
2.5 Konsep Penelitian

Berikut gambaran dari rancangan penelitian



III. HASIL DAN PEMBAHASAN

3.1 Konfigurasi IPv4



Pada konfigurasi ini komputer *Server* diberi alamat IP 192.168.31.1/24 dan untuk

komputer *Client* yang pertama diberi alamat IP 192.168.31.2 sampai komputer yang ke 25 di beri IP 192.168.31.26 secara berurutan.

4.1.1 Performa *CPU Time* dan *Kernel Concurrent Fork Server* dan *Concurrent Pre-Fork Server* pada IPv4

Pengujian performa *CPU Time* dan *Kernel Time* dilakukan dengan bantuan tool aplikasi NMON yang dipasang di komputer *Server*. Pengujian dilakukan dengan cara mengaktifkan tool NMON agar mencatat semua proses selama 1 menit setelah itu komputer *Client* melakukan request koneksi ke komputer *Server*, dengan melakukan 20, 40, 60, 80, 100 koneksi dan 1000 byte setiap *Client*

a. *CPU time*

CPU Time adalah jumlah waktu yang dibutuhkan di dalam central processing unit (*CPU*) yang digunakan untuk pengolahan instruksi dari program komputer atau sistem operasi. Penggunaan *CPU* diperoleh dari hasil capture NMON di bawah ini adalah tabel dari 5 proses ujicoba.

80	7.012,816	6.387,633
100	10.894,123	9.025,483

Tabel CPU time pada Ipv4

koneksi	<i>Concurrent Fork Server</i>	<i>Concurrent Pre-Fork Server</i>
20	12.597 %	6.116 %
40	21.004 %	7.093 %
60	28.621 %	7.590 %
80	35.397 %	8.857 %
100	42.446 %	10.081 %

Dari ujicoba yang dilakukan bisa dilihat perbandingan penggunaan *CPU* dimana *Concurrent Fork Server* lebih banyak menggunakan *CPU* (42.446 %) dengan 100 koneksi per *Client* dibandingkan *Concurrent Pre-Fork Server* yang hanya menggunakan *CPU* (10.081 %) pada jaringan IPv4.

b. Kernel

Kernel adalah suatu perangkat lunak yang menjadi bagian utama dari sebuah sistem operasi. Tugasnya melayani bermacam program aplikasi untuk mengakses perangkat keras komputer secara aman.

Kernel Time adalah jumlah waktu yang dibutuhkan dalam menjalankan beberapa proses yang dibutuhkan oleh aplikasi

Tabel Kernel pada IPv4

konek si	<i>Concurrent Fork Server</i>	<i>Concurrent Pre-Fork Server</i>
20	2.442,238	2.445,881
40	4.242,874	3.986,419
60	5.597,362	5.687,326

Dari ujicoba yang dilakukan bisa dilihat perbandingan penggunaan *Kernel* dimana *Concurrent Fork Server* lebih banyak menggunakan *Kernel* (10.894,123 proses) dengan 100 koneksi per *Client* dibandingkan *Concurrent Pre-Fork Server* yang hanya menggunakan *Kernel* (9.025,483proses) pada jaringan IPv4.

4.1.2 Performa CPU Time dan Kernel Concurrent Fork Server dan Concurrent Pre-Fork Server pada IPv6

Pengujian performa *CPU Time* dan *Kernel Time* dilakukan dengan bantuan tool aplikasi NMON yang dipasang di komputer *Server*. Pengujian dilakukan dengan cara mengaktifkan tool NMON agar mencatat semua proses selama 1 menit setelah itu komputer *Client* melakukan request koneksi ke komputer *Server*, dengan melakukan koneksi 20, 40, 60, 80, 100 koneksi dan 1000 byte setiap *Client*.

a. CPU time

CPU Time adalah jumlah waktu yang dibutuhkan di dalam central processing

unit (*CPU*) yang digunakan untuk pengolahan instruksi dari program komputer atau sistem operasi. Penggunaan *CPU* diperoleh dari hasil capture NMON di bawah ini adalah tabel dari 5 proses ujicoba.

Tabel *CPU time* pada IPv6

koneksi	<i>Concurrent Fork Server</i>	<i>Concurrent Pre-Fork Server</i>
20	9.357 %	2.767 %
40	23.532 %	5.525 %
60	35.822 %	7.263 %
80	53.140 %	9.120 %
100	66.235 %	9.668 %

Dari ujicoba yang dilakukan bisa dilihat perbandingan penggunaan *CPU* dimana *Concurrent Fork Server* lebih banyak menggunakan *CPU* (66,235 %) dengan 100 koneksi per *Client* dibandingkan *Concurrent Pre-Fork Server* yang hanya menggunakan *CPU* (9.668 %) pada jaringan IPv6.

b. Kernel

Kernel adalah suatu perangkat lunak yang menjadi bagian utama dari sebuah sistem operasi. Tugasnya melayani bermacam program aplikasi untuk mengakses perangkat keras komputer secara aman.

Kernel Time adalah jumlah waktu yang dibutuhkan dalam

menjalankan beberapa proses yang dibutuhkan oleh aplikasi

Tabel Kernel pada IPv6

koneksi	<i>Concurrent Fork Server</i>	<i>Concurrent Pre-Fork Server</i>
20	1.704,520	998,902
40	4.536,770	1.265,352
60	7.233,632	1.539,623
80	10.894,123	1.781,873
100	13.339.745	5.635,232

Dari ujicoba yang dilakukan bisa dilihat perbandingan penggunaan *Kernel* dimana *Concurrent Fork Server* lebih banyak menggunakan *Kernel* (13.339.745 proses) dengan 100 koneksi per *Client* dibandingkan *Concurrent Pre-Fork Server* yang hanya menggunakan *Kernel* (5.635,232 proses) pada jaringan IPv6.

IV. PENUTUP

Dari hasil ujicoba didapatkan hasil sebagai berikut :

1. Untuk *Concurrent Fork Server* dimana proses yang dibangun untuk menangani setiap koneksi masuk setelah diminta, hasil penelitian menunjukkan bahwa butuh 42.446% per detik penggunaan *CPU* dalam menangani 100 koneksi per client untuk koneksi IPv4 sementara 66.235 % per detik untuk koneksi IPv6. Sedangkan penggunaan kernel pada IPv4 membutuhkan 10.894,123 proses per detik untuk 100 koneksi per client

sementara pada IPv6 membutuhkan 13.339.745 proses per detik.

2. Untuk Concurrent Pre-Fork Server dimana child server sudah siap sebelum permintaan koneksi diminta, butuh 10.081 % per detik penggunaan CPU dalam menangani 100 koneksi per client untuk koneksi IPv4 sementara 9.668% per detik untuk koneksi IPv6. Sedangkan penggunaan kernel pada IPv4 membutuhkan 9.025,483 proses per detik untuk 100 koneksi per client sementara pada IPv6 membutuhkan 5.635,232 proses per detik.
3. Berdasarkan hasil penelitian ini menunjukkan bahwa kinerja Concurrent Fork Server lebih berat dibandingkan Concurrent Pre-Fork Server pada jaringan IPv4 maupun IPv6

DAFTAR PUSTAKA

- [1] Anonim. *BAB III METODOLOGI PENELITIAN*. Di Download pada 17 April 2014. <http://www.digilib.its.ac.id/public/ITS-Undergraduate-5121-4201100038-bab3.pdf>.
- [2] Grox.net. *IPv6 calculator*. Diakses 24 Oktober 2014. <http://grox.net/utills/ipv6.php>.
- [3] Indowebsiana.Com. *Apa itu Teknologi IPv6 dan Mengapa ini Sangat Penting..*?. Diakses 24 Oktober 2014. http://www.indowebsia.com/shownews.php?news_id=74.
- [4] Nmonvisualizer.github.io. NMONvisualizer diakses 23 desember 2014

<http://nmonvisualizer.github.io/nmonvisualizer/index.html>

- [5] John, J., Minoli, D. Amoss.(2007). *Handbook of IPv4 to IPv6 Transition*. Auerbach Publications
- [6] Rikih Gunawan. Pemrograman Socket dengan Python diakses 27 oktober 2014. http://www.unej.ac.id/files/pdf2/rikih-socket_python.pdf
- [7] Syafrizal, Melwin. *TCP/IP*. Di Download pada 26 Oktober 2014. <http://journal.amikom.ac.id/index.php/KIDA/article/view/4481/2175>
- [8] Teddy Mantoro, Media A. Ayu, Amir Borovac and Aqqiela Z. Z. Zay Department of Computer Science, KICT International Islamic University Malaysia, Kuala Lumpur, Malaysia, teddy@ieee.org
- [9] Ti.unmuhjember.ac.id. Panduan Tugas Akhir 2014 di download pada 24 Oktober 2014. http://ti.unmuhjember.ac.id/wp-content/uploads/2014/12/Panduan_Proposal_20141.pdf
- [10] Widiyansah, AA.(2013). *Perancangan Jaringan Laboratorium Komputer Universitas Muhammadiyah Jember Menggunakan Internet Protocol Version 6 (IPv6)*.(Skripsi). Jurusan Teknik Informatika Universitas Muhammadiyah Jember.