

- CLASSIFIED -

Tugas Besar

IF3230 Sistem Paralel dan Terdistribusi



SPECIAL TRAINING

MISSION “Setan Gundul”

Waktu Mulai Pengerjaan : Senin, 28 April 2014
Batas Waktu Pengumpulan I : Sabtu, 3 Mei 2014 pukul 23:24:25
Batas Waktu Pengumpulan II : Sabtu, 10 Mei 2014 pukul 23:24:25
Tempat Upload : <http://oddyseus.if.itb.ac.id/>
Penamaan File untuk Upload : TX-KY-GZZ.zip

PS: Deskripsi sengaja dibuat singkat, padat, dan tidak jelas. Problem? Tanyakan.

Latar Belakang:

Selamat telah menyelesaikan training. Setelah ini dengan kelompok yang sama dengan sebelumnya, kalian akan ditugaskan dalam sebuah misi percobaan. Daftar kelompok dapat dilihat pada <http://adf.ly/dQHZA>. Nomor ID yang kalian terima adalah nomor unik yang akan menjadi identitas kelompok kalian selama misi ini. Nomor ini akan digunakan untuk meng-*upload* data sesuai *deliverable* ke alamat <http://oddyseus.if.itb.ac.id/> untuk misi IF3230 Sistem Paralel dan Terdistribusi.

Kementerian Pembangunan Daerah Tertinggal Republik Indonesia melaporkan adanya malfungsi pada sistem mereka (insiden 27 Februari 2014). Ditengarai sistem telah dikompromisasi oleh peretas asal Russia dan menyebabkan lumpuhnya sistem selama dua hari. Berdasarkan hasil audit yang dilakukan oleh tim audit Third-I, diputuskan bahwa akan dikembangkan sebuah basis data terdistribusi menggantikan sistem basis data terpusat. Sistem ini nantinya akan di bawah pengawasan Third-I sebagai bagian dari sistem *Integrated Cyber Corridor*.

Meski tak termasuk kritikal, misi ini tetap bersifat rahasia. Tidak diperkenankan untuk menyebarluaskan informasi ini ke luar institusi Third-I.

Petunjuk:

Penamaan file mengikuti format TX-KY-GZZ.zip atau TX-KY-GZZ.rar dengan huruf X digantikan dengan nomor tahap (dijelaskan kemudian), Y diganti dengan nomor kelas, huruf ZZ digantikan dengan ID (nomor kelompok). Untuk nomor ID yang hanya terdiri dari satu digit, tambahkan awalan 0 (nol). Cukup salah satu anggota kelompok yang mengunggah file.

File yang dikumpulkan:

- *Source code* yang diberi komentar jelas (salah satu bagian dari penilaian)
- *Binary file* hasil kompilasi source code (format ELF64 atau jar)
- Dokumentasi program berisi minimum penjelasan tentang struktur data dan strategi yang digunakan untuk alokasi token ke server

Pengembangan akan dibagi menjadi dua bagian besar. Harap perhatikan detail yang diberikan.

Harap perhatikan waktu pengumpulan. Keterlambatan pengumpulan akan mengurangi nilai.

Tujuan:

Mengembangkan basisdata tersebar sederhana bersifat key-value.

Batasan:

1. Pilihan bahasa yang dapat digunakan adalah C/C++ dan Java
2. Boleh menggunakan *library* pendukung (tuliskan dalam dokumentasi). Library yang digunakan dibatasi hanya pada *network library*. Disarankan tidak menggunakan MPI.
3. Untuk pengguna C++, diperbolehkan menggunakan STL. Pengguna Java juga dapat menggunakan Collection.

Persiapan:

Segala percobaan akan dilakukan di mesin-mesin virtual. Ada empat mesin yang dipergunakan dalam misi kali ini:

1. Arctic - 167.205.33.194
2. Deux - 167.205.33.196
3. Tremors - 167.205.33.197
4. Delta - 167.205.33.220

Alamat mesin akan diberitahu kemudian. Mesin-mesin ini dapat diakses dari jaringan dalam, atau menggunakan fasilitas VPN.

PS: Deskripsi sengaja dibuat singkat, padat, dan tidak jelas. Problem? Tanyakan.

Untuk login gunakan username berupa “Evangelist_<kelas>_<ID>” tanpa tanda kutip, dengan <kelas> adalah bilangan 1 untuk ganjil dan 2 untuk genap, dan <ID> adalah ID nomor ID kelompok. Password yang digunakan adalah “tugasbesar” tanpa tanda kutip.

Seluruh kegiatan akan dilakukan di mesin tersebut.

Tahap 1 – Model Client Server Tunggal

Deadline: 3 Mei 2014

Pada tahap 1 ini, hanya dibutuhkan dua program: *client* dan *server*. Keduanya merupakan aplikasi berformat binary ELF64 (64-bit) untuk C++ dan jar untuk Java. Setiap tim harus menjamin program yang dihasilkan dapat berjalan pada mesin uji.

Spesifikasi yang diberikan adalah sebagai berikut:

1. Basis data dapat menyimpan sejumlah tabel berisi sejumlah pasangan key-value. Implementasi fisik dibebaskan (misal: menyimpan data dalam sebagai *file*, menyimpan data di memori).
2. Key dan value bertipe string. Jika data berupa bilangan, ubah sebagai string.
3. Setiap data yang disimpan pada tabel harus dilabeli dengan timestamp secara otomatis. Operasi update data pada sebuah tabel yang memiliki key yang sama tidak akan menghapus data lama, namun akan menambahkan data dengan timestamp yang lebih baru.
4. Client dapat terhubung ke server database dengan menggunakan *command line*, dan dapat menerima salah satu dari perintah berikut:
 - a. Membuat tabel: **create table <nama table>;**
 - b. Menyisipkan row baru: **insert <nama table> <key> <value>;**
 - c. Menampilkan isi tabel: **display <nama table>;** Perintah ini akan menampilkan isi tabel berupa pasangan key-value dan *timestamp*. Jika terdapat beberapa data dengan key sama, tampilkan data dengan *timestamp* terakhir.
5. Server selalu menyimpan data sebagai triplet <key, value, timestamp>

Pilihlah struktur data yang tepat agar server selalu dapat dengan cepat mencari data dengan timestamp terakhir berdasarkan nama tabel, key.

Tak ada batasan format dalam penjelasan struktur data. Usahakan penjelasan singkat dan mudah dipahami.

Tahap 2 – Model Kluster Server

Deadline: 10 Mei 2014

Server yang ada pada tahap ini adalah pengembangan dari *server* di tahap satu. Bisa berupa file program yang sama, maupun berbeda. *Server* yang dikembangkan harus dapat dijalankan menggunakan n komputer. Strategi yang dipergunakan adalah sebagai berikut:

Setiap server akan memiliki nilai unik yang disebut sebagai token (panjang 32 bit). Token ini digunakan untuk menentukan data apa yang akan disimpan pada server tersebut (berdasarkan nilai hash key data tersebut). Kalian dapat menggunakan berbagai strategi untuk mengalokasikan token ke server. Sebagai contoh, digunakan strategi berikut (tidak harus diikuti):

1. Saat hanya ada 1 server, maka server tersebut diberi token bernilai 0, dan bertanggung jawab menyimpan data dengan range hash key dari $0 - (2^{31}-1)$ (semua data)
2. Saat ada server baru bergabung (total 2) maka rentang nilai akan dibagi 2: server pertama memiliki token 0 dan menyimpan nilai data dengan range hash key dari $0 - (2^{30}-1)$, sementara server kedua memiliki token bernilai 2^{31} dan menyimpan data dengan range hash key dari $2^{30} - (2^{31}-1)$.
3. Saat ada server baru lagi bergabung (total 3), maka rentang salah satu server akan dibagi 2. Misal jika yang dibagi adalah server pertama, maka token untuk server pertama tetap 0, dengan data range menjadi dari $0 - (2^{29}-1)$, server yang baru (server ketiga) mendapat token 2^{29} , dengan data range dari $2^{29} - (2^{30}-1)$, dan server kedua tetap dengan token 2^{30} dengan range data $2^{30} - (2^{31}-1)$.
4. Demikian seterusnya. Setiap ada server baru bergabung, maka range nilai dari salah satu server yang ada akan dibagi 2, dan diberikan ke server baru.

Rancanglah mekanisme yang mengatur protokol penambahan server ke dalam cluster, termasuk jika ada data yang harus dimigrasi dari server lama ke server baru. Update informasi range data masing-masing server pada setiap server.

Rancanglah mekanisme yang memungkinkan client dapat menghubungi server mana pun untuk menulis atau membaca data dengan key apapun. Saat sebuah server menerima request dari client, server harus menentukan server mana yang seharusnya bertanggung jawab untuk mengelola data tersebut. Jika request itu bukan ditujukan untuk dirinya, *server* kemudian meneruskan request ke *server* yang seharusnya alih-alih segera memproses.