Tugas Besar
IF3230 Sistem Paralel dan Terdistribusi
Semester 2 - 2016/2017
Load Balancer dengan Konsensus Raft

Deskripsi Tugas

Anda ditugaskan untuk mencari bilangan-bilangan prima. Karena anda memiliki beberapa komputer, disebut ***server***, maka anda berpikiran untuk membagi pencarian bilangan tersebut ke dalam *server-server* tersebut. Di dalam setiap *server*, terdapat program yang dapat mencari bilangan prima ke-N, disebut ***worker***. Agar masing-masing *server* menerima beban kerja yang seimbang, maka anda perlu membuat suatu ***load balancer***.

Load balancer tersebut terdiri atas beberapa ***node***. Setiap *node* dapat menerima *request* berupa angka prima yang ingin dicari dan melemparkan *request* tersebut ke *server* yang memiliki beban terendah, yaitu *server* dengan *CPU load* terendah. *Request* dikirimkan sebagai *HTTP request*. *Request* tersebut berisi suatu integer N yang menyatakan bilangan prima ke-N yang ingin dicari. *Response* dari *request* tersebut adalah bilangan prima yang dicari tersebut.

Setiap beberapa detik (silahkan ditentukan sendiri), *server* akan mengirimkan beban kerja ke **semua *node*** *load balancer* memanfaatkan sebuah program monitor yang disebut ***daemon***. Jika dalam *timeout* sekian *load balancer* tidak menerima informasi beban kerja dari salah satu *server*, maka *server* tersebut dianggap mati dan tidak dapat menerima *request*. *Load balancer* akan melakukan konsensus untuk mengubah state *server* tersebut menjadi mati. Jika *load balancer* menerima informasi dari *server*, maka *server* tersebut dianggap menyala dan dapat menerima *request*.

Untuk menjamin bahwa informasi beban kerja dalam setiap *node load balancer* adalah informasi terbaru, maka digunakan algoritma konsensus untuk melakukan *voting* informasi beban kerja mana yang terbaru untuk digunakan oleh seluruh *node*. Dalam kasus ini, digunakan **algoritma konsensus Raft** (lihat referensi di bawah). Saat *load balancer* menerima informasi beban dari *server*, *load balancer* akan melakukan konsensus untuk *update state* informasi beban kerja yang terbaru.

Saat *load balancer* menerima request dari client, *load balancer* akan melihat *state* informasi beban masing-masing *server* **berdasarkan konsensus** (tidak meminta informasi dahulu ke masing-masing *server*), memilih *server* dengan *CPU load* terendah, dan meneruskan request ke *worker* di *server* tersebut.

Perlu diingat agar kasus *node load balancer* mati ditangani dengan baik. Jika *leader* dari *load balancer* mati, maka *load balancer* harus memilih *leader* baru sesuai dengan metode *leader election* pada Raft. Selain itu, jika ada *follower load balancer* yang mati, konfigurasi Raft tidak berubah. Sebagai contoh, jika jumlah *node* 5, meskipun salah satu *node* mati, maka *majority* tetap 3.

Worker

Untuk tugas ini, kami telah menyediakan program pencarian bilangan prima sebagai *worker* untuk dijalankan di *server*. *Worker* ini berupa *script* Python bernama worker.py. Saat script ini dijalankan, script akan membuat sebuah *server* HTTP di localhost port 13337. Untuk mencari bilangan prima, lakukan *request* GET ke *server* tersebut. Misalkan, untuk mencari bilangan prima ke 3, kirim *request* GET ke localhost:13337/3. Server akan mengembalikan bilangan prima ke-3, yaitu 5.

Pengerjaan

Tugas dikerjakan dalam kelompok maksimal 3 orang (anggota kelompok tidak boleh dari kelas yang berbeda). Deadline untuk pengumpulan tugas ini adalah **28 April 2017, 23.59**. Deliverables dari tugas ini adalah ***source code*** dari ***node*** dan ***daemon*** *load balancer* yang kalian buat. Untuk pembuatan algoritma Raft, dipersilahkan mempelajari referensi, tetapi **tidak boleh menyalin langsung kode implementasi Raft yang sudah ada**. Karena **tidak ada laporan**, diharapkan agar *source code* kalian **diberi komentar** dengan baik. Mekanisme pengumpulan serta jadwal untuk demo tugas besar akan diumumkan kemudian melalui milis. Segala kecurangan akan mendapatkan konsekuensi. Bila ada pertanyaan yang kurang jelas mengenai tugas, dipersilahkan bertanya dan berdiskusi melalui milis. Selamat mengerjakan!

Referensi

https://raft.github.io/
http://thesecretlivesofdata.com/raft/