

TUGAS
MATAKULIAH : KOMPUTASI BERBASIS JARINGAN



DOSEN PENGAMPU
Ibnu Febry Kurniawan, S.Kom., M.Sc.
Pradini Puspitaningayu, S.T., M.T., Ph.D.

DISUSUN OLEH:
Windy Chikita Cornia Putri (24051905004)

PROGRAM STUDI S2 INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SURABAYA
TAHUN AJARAN GASAL 2024/2025

broker.py

A. Latar Belakang

Pada sistem berbasis client-server, distribusi beban adalah salah satu cara untuk meningkatkan efisiensi dan kinerja layanan, terutama saat beban kerja harus dibagi ke beberapa server. Dalam studi kasus ini, kami mengimplementasikan sebuah broker yang berfungsi sebagai penghubung antara klien dan server pekerja (worker server), dengan menggunakan teknik Round Robin dan Load Balancing.

B. Deskripsi Sistem

- File broker.py berfungsi sebagai broker server yang mendistribusikan permintaan klien ke tiga server pekerja melalui dua pendekatan:
- Round Robin: Permintaan didistribusikan secara bergantian (sequential) ke setiap pekerja, memastikan distribusi merata tanpa memandang beban tiap server.
- Load Balancing (Minimal Beban): Permintaan dikirim ke pekerja dengan beban terkecil (jumlah permintaan tersimpan), sehingga pekerja yang kurang sibuk lebih sering menerima permintaan baru.

C. Alur Program

- Pendataan Server Pekerja: Daftar pekerja beserta port disimpan dalam variabel WORKERS.
- Distribusi Permintaan: Fungsi distribute_request menentukan metode distribusi berdasarkan allocation_type yang dikirim klien.
- Round Robin memperbarui pekerja berikutnya dengan round_robin_counter.
- Load Balancing memilih pekerja dengan jumlah permintaan terendah di requests_served.
- Pengiriman dan Penerimaan Data: Broker mengirim permintaan (application_id) ke pekerja yang dipilih dan menunggu respons. Setelah menerima respons dari pekerja, broker mengirimnya kembali ke klien.
- Server Broker: Fungsi broker_server menerima koneksi klien, mendistribusikan permintaan dengan distribute_request, dan mengembalikan respons ke klien.

worker.py

A. Latar Belakang

Dalam sistem terdistribusi, server pekerja (worker server) memainkan peran penting untuk menangani berbagai jenis tugas yang didistribusikan oleh broker. Tugas-tugas ini dapat memiliki tingkat kompleksitas yang berbeda, sehingga memerlukan waktu pemrosesan yang bervariasi.

B. Deskripsi Sistem

- File worker.py berfungsi sebagai server pekerja yang menerima tugas dari broker dan memprosesnya berdasarkan jenis tugas. Terdapat dua tipe tugas utama:
- Long Task: Tugas dengan tingkat kompleksitas tinggi yang disimulasikan membutuhkan waktu pemrosesan lebih lama.
- Short Task: Tugas dengan kompleksitas lebih rendah yang memerlukan waktu pemrosesan lebih singkat.

C. Alur Program

- Fungsi handle_request: Fungsi ini menerima request_type dan memprosesnya sesuai jenis tugas.
- Long Task: Simulasi pemrosesan yang kompleks, membutuhkan waktu 2 detik (time.sleep(2)).
- Short Task: Simulasi pemrosesan yang lebih cepat, membutuhkan waktu 1 detik (time.sleep(1)).
- Server Pekerja (worker_server):
- Menggunakan library socket, server pekerja dibuka pada port yang diberikan.
- Server mendengarkan permintaan dari broker dan menerima koneksi klien.
- Berdasarkan jenis permintaan yang diterima (Long atau Short), server pekerja memproses tugas dan mengirimkan respons kembali ke broker.

client.py

A. Latar Belakang

Program ini merupakan bagian dari aplikasi yang berkomunikasi dengan server menggunakan soket TCP untuk mengirimkan permintaan alokasi berdasarkan ID aplikasi dan metode alokasi tertentu. Sistem ini dapat digunakan dalam berbagai konteks, seperti manajemen sumber daya dalam aplikasi distribusi atau sistem beban kerja.

B. Deskripsi Sistem

Sistem ini terdiri dari:

- Client: Kode client.py yang bertindak sebagai pengirim permintaan.
- Server: (yang tidak ditampilkan dalam kode ini) seharusnya menerima permintaan dari client, memprosesnya berdasarkan `application_id` dan `allocation_type`, dan kemudian mengembalikan respons yang relevan.

C. Alur Program

- Import Library: Program dimulai dengan mengimpor modul socket, yang memungkinkan komunikasi melalui soket.
- Fungsi `send_request`: Fungsi ini menerima dua parameter: `application_id` dan `allocation_type`. Fungsi ini bertanggung jawab untuk:
 - Membuat soket TCP menggunakan `socket.socket()`.
 - Menghubungkan soket ke server yang berjalan pada localhost dengan port 8000.
 - Mengirimkan pesan yang berisi `application_id` dan `allocation_type`, dipisahkan oleh koma, setelah mengubahnya menjadi format byte menggunakan `.encode('utf-8')`.
 - Menerima respons dari server yang di-decode kembali ke format string dan mencetaknya.
- Pengujian Fungsi: Pada bagian `if __name__ == "__main__":`, fungsi `send_request` dipanggil tiga kali dengan kombinasi ID aplikasi dan metode alokasi yang berbeda untuk menguji sistem. Ini memberikan contoh penggunaan:
 - Mengirim permintaan dengan ID "Long" dan metode alokasi "round_robin".
 - Mengirim permintaan dengan ID "Short" dan metode alokasi "load_balancing".
 - Mengirim permintaan lagi dengan ID "Long" tetapi menggunakan metode alokasi "load_balancing".