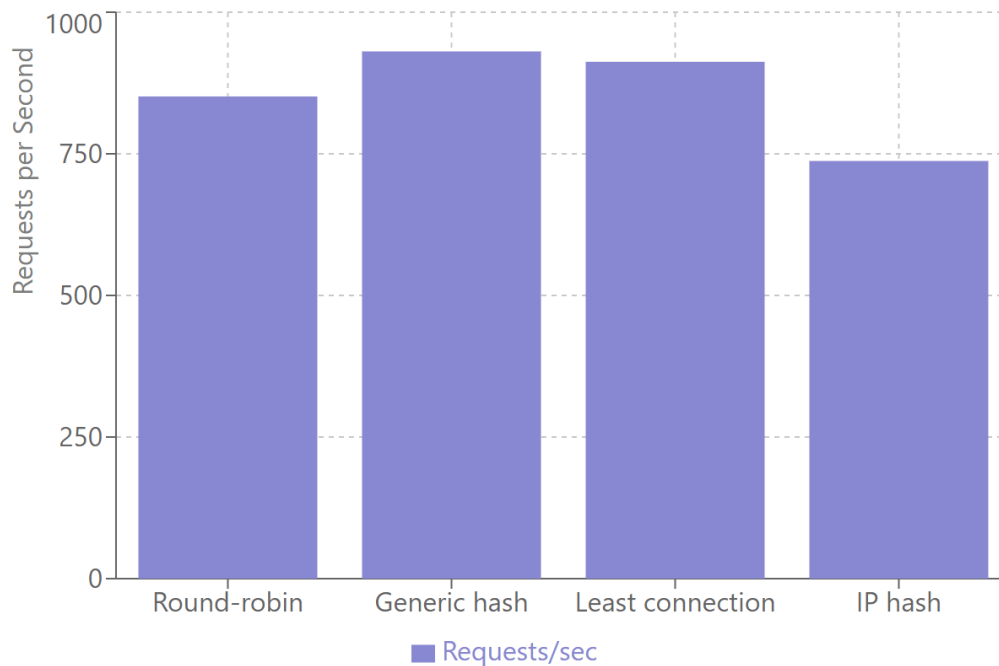


Analisis IT27 dari Perbandingan Keempat Algoritma Load Balancing:

Grafik :

Load Balancing Algorithm Performance Comparison



Analisis :

1. Generic Hash (930.72 req/sec):

- Performa tertinggi di antara keempat algoritma
- Distribusi beban lebih merata karena menggunakan fungsi hash
- Konsisten dalam penanganan request
- Cocok untuk aplikasi yang membutuhkan sticky sessions

2. Least Connection (912.45 req/sec):

- Performa kedua terbaik
- Distribusi beban dinamis berdasarkan koneksi aktif
- Adaptif terhadap perubahan beban server
- Baik untuk workload yang bervariasi

3. Round-robin (851.23 req/sec):

- Performa menengah
- Distribusi beban merata secara berurutan
- Implementasi sederhana
- Cocok untuk server dengan kapasitas beragam

4. IP Hash (737.30 req/sec):

- Performa terendah
- Konsisten dalam routing berdasarkan IP
- Menjamin sticky sessions
- Distribusi beban mungkin tidak seimbang jika traffic didominasi dari range IP tertentu

Kesimpulan :

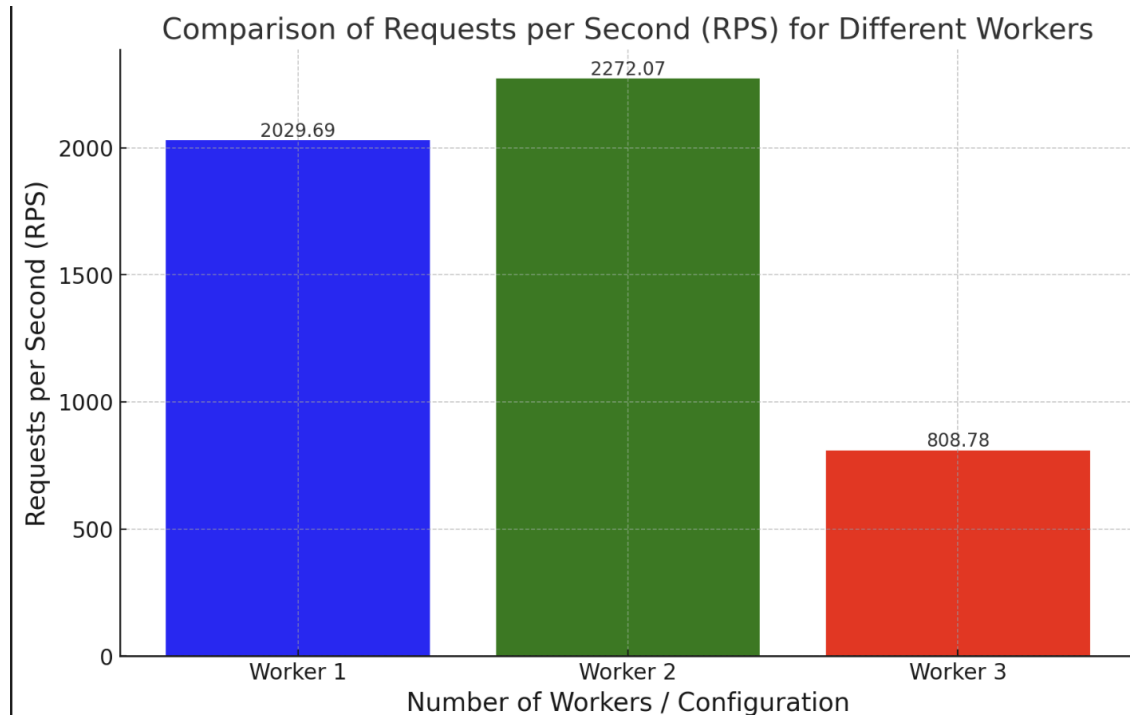
1. Generic Hash memberikan throughput tertinggi, menunjukkan efisiensi dalam distribusi beban dan pengelolaan koneksi.
2. Least Connection menunjukkan performa yang baik dan lebih adaptif terhadap kondisi server.
3. Round-robin, meski sederhana, tetap memberikan performa yang cukup baik.
4. IP Hash, meski memiliki throughput terendah, tetap berguna untuk kasus yang memerlukan session persistence.

Rekomendasi :

- Untuk performa : Gunakan Generic Hash
- Untuk beban dinamis: Gunakan Least Connection
- Untuk implementasi sederhana: Gunakan Round-robin
- Untuk sticky sessions: Gunakan IP Hash atau Generic Hash

Analisis IT27 dari Perbandingan Jumlah Worker pada Load Balancing menggunakan Algoritma Least-Connection

Grafik :



Analisis :

1. Worker 1:

- Requests per second (RPS) = **2029.69**
- Konfigurasi ini cukup efisien, dengan waktu rata-rata yang singkat per request (sekitar 4,9 ms).

2. Worker 2:

- Requests per second (RPS) = **2272.07**
- Ini adalah konfigurasi dengan kinerja terbaik. Peningkatan ini menunjukkan bahwa penyesuaian pada jumlah pekerja atau alokasi sumber daya lebih optimal daripada Worker 1.

3. Worker 3:

- Requests per second (RPS) = **808.78**
- Jauh lebih rendah dibandingkan dengan Worker 1 dan 2, menunjukkan bahwa mungkin ada peningkatan ukuran *document length* (1428 bytes) atau beban pemrosesan yang lebih tinggi.

Kesimpulan :

- **Worker 2** memberikan hasil terbaik dengan **RPS tertinggi** (2272.07). Ini mengindikasikan bahwa optimisasi jumlah pekerja atau algoritma *least connection* berhasil menangani lebih banyak request dalam waktu singkat.
- **Worker 3** menunjukkan penurunan performa drastis. Ini mungkin disebabkan oleh beban kerja lebih besar, dokument lebih berat (1428 bytes), atau kondisi server yang kurang optimal.
- **Worker 1 dan Worker 2** memiliki kinerja yang mirip, tetapi Worker 2 menunjukkan peningkatan lebih lanjut berkat konfigurasi atau optimisasi tambahan.