

Concurrency (Go) – Loadbalancing

Voor opdracht 6 heb ik in go een loadbalancer gemaakt die HTTP request verdeelt over meerdere servers. Ik heb 3 verschillende algoritmes geïmplementeerd, namelijk:

- Round Robin
- Weighted Round Robin
- Least Connections

In deze applicatie is het mogelijk om meerdere clients tegelijkertijd request te laten sturen naar een specifieke load balancer, die deze weer verdeelt over meerdere servers. Zo heeft elke client, server en load balancer draait in zijn eigen goroutine. Een goroutine zijn threads in GO, maar dan wegen ze niet zo zwaar en kan je er best veel mee opzetten. Clients sturen continu HTTP request met een wachttijd tussen de 0 en 1 seconden. De load balancer bepaalt dan op basis van een algoritme naar welke server deze request gaat. De server wacht vervolgens een aantal seconden en stuurt vervolgens een response terug. In dit verslag wordt zo ook de performance onderzocht. De client houdt bij hoelang het duurt minus de wachttijd.

Ik heb het volgende getest:

- Aantal clients: 50, 100 en 150 clients tegelijkertijd
- Aantal servers: 3, 5 en 8 servers, alleen heb ik ze niet allemaal in dit verslag verwerkt

Zo heb ik voor elke algoritme aangeroepen om te controleren wanneer een load balancer goed of slecht werkt. Zo draaiden ik bijvoorbeeld de commando ‘go run main.go -Clients=50 -servers=3 -algorithm==rr duration=60’ om vervolgens de resultaten te krijgen.

Resultaten

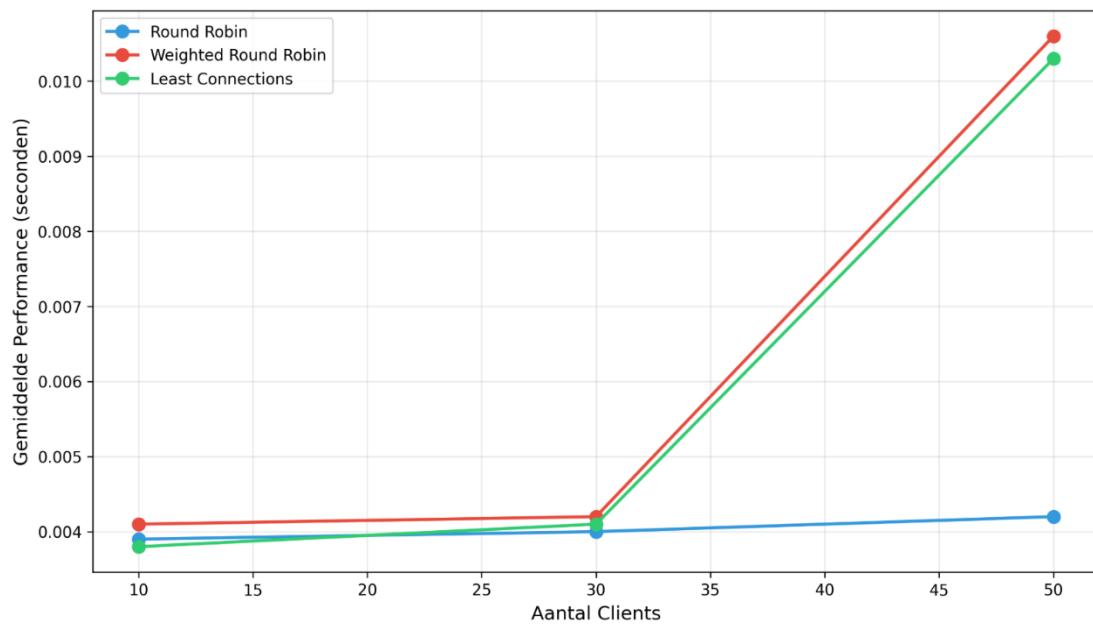
De eerst volgende resultaten zijn bij een lage load, namelijk 10 clients en 3 servers. Wat opvalt aan de resultaten hieronder is dat de performance bij een lage load bijna allemaal gelijk zijn

Algoritme	Performance	Throughput
Round Robin	0.0039 seconden	32 req/s
Weighted Round Robin	0.0041 seconden	31 req/s
Least Connections	0.0038 seconden	33 req/s

Hieronder zie je de testresultaten bij een hogere load, namelijk 50 clients en 3 servers. Bij deze hogere load met 50 clients is Round Robin sneller. Dit komt waarschijnlijk

doordat Round Robin het simpelste algoritme heeft vergleken met de Least Connections en Weighted Round Robin.

Algoritme	Performance	Throughput
Round Robin	0.0042 seconden	99 rreq/s
Weighted Round Robin	0.0106 seconden	98 req/s
Least Connections	0.0103 seconden	99 req/s



Uit deze kleine onderzoek is naar voren gekomen dat Round Robin bij een hogere load het snelste is. Bij een lage load maakt het allemaal niet zoveel uit.