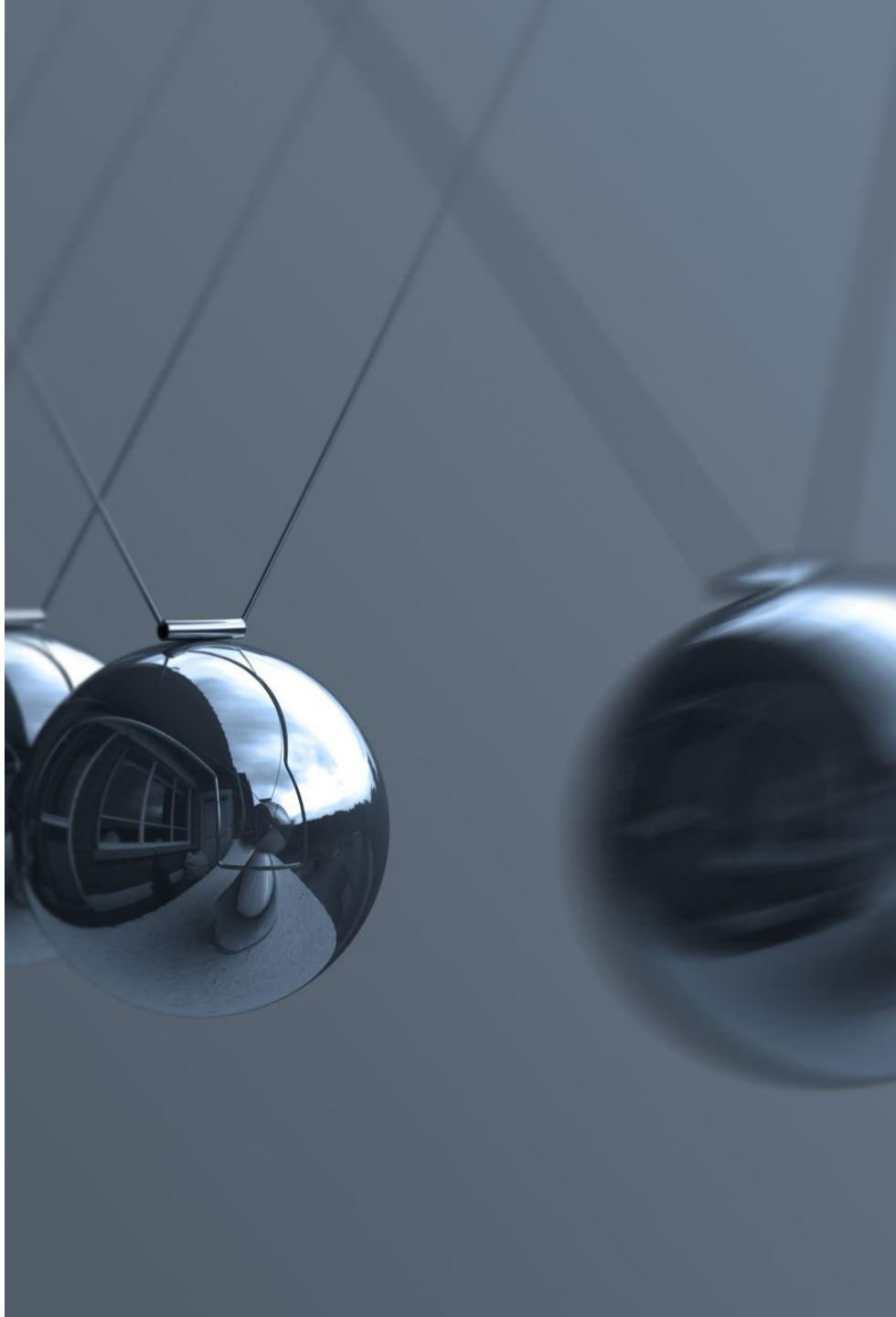
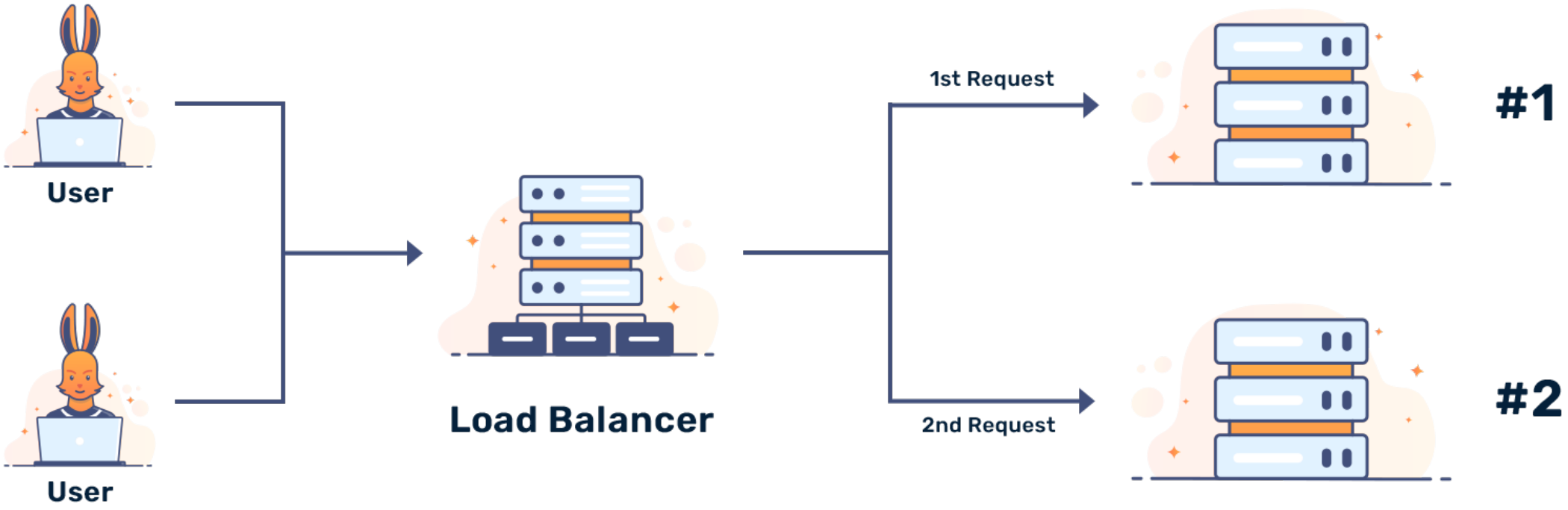
Computation-Centric 서비스를 위한 Load Balancing 알고리즘 탐색




User
User
Load Balancer
1st Request
2nd Request
#1
#2

Round Robin

Weighted Round Robin

Least Connection

Least Response Time

=> Custom Load Balancer

Custom Load Balancer

- 1. Asynchronous Non-Blocking I/O
- 2. 가중치 값이 낮을 수록 우선순위 – PriorityQueue
- 2. Initialization : 동일한 Computation 작업 => '응답시간'
- 3. Service (동적 가중치 변경):
  - Connection 수 고려
  - 일정 횟수만큼 응답시간 저장 => 평균 응답 시간 계산
  - 주기적으로 ping 요청 => 응답시간 확인
  - 주기적으로 가중치 왜곡 보정 => 초기 값으로 Reset

∴ **요청이 많고 다량의 Computation을 요하는 서비스**
=> 성능, 확장성, 가용성, 신뢰성 ↑

spring Eureka

HOME LAST 1000 SINCE STARTUP

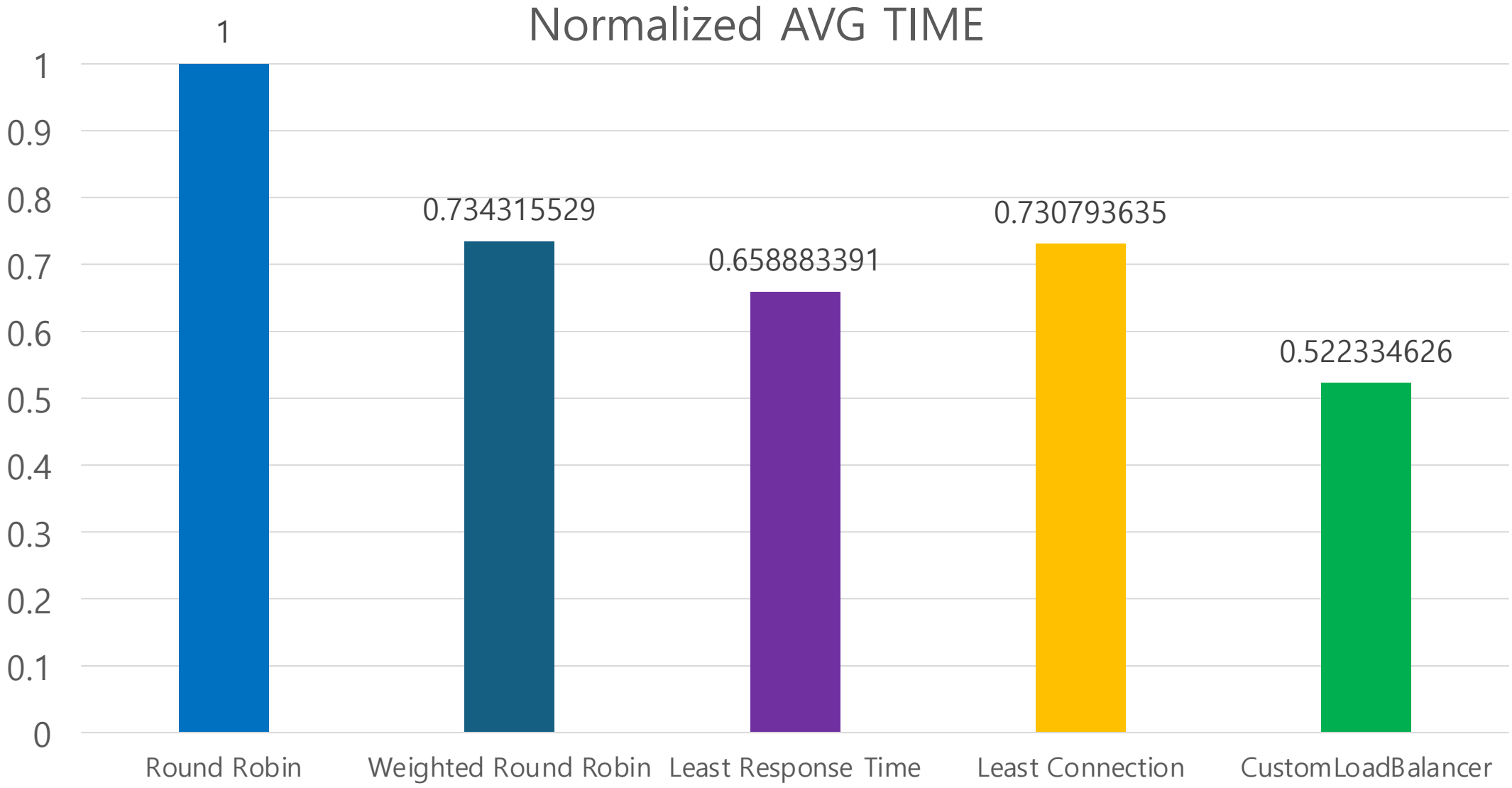
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DSPROJECT-SERVER	n/a (1)	(1)	UP (1) - [redacted].253:dsproject-server:8000
DSPROJECT-SERVICE	n/a (3)	(3)	UP (3) - [redacted]114:DSPROJECT-SERVICE:3000 , [redacted]118:DSPROJECT-SERVICE:3000 [redacted].183:DSPROJECT-SERVICE:3000

	Service Server (DSPROJECT-SERVICE)	RAM
1	Intel(R) Xeon(R) Gold 6430 @ 2.10GHz – 64 Cores	256GB
2	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz – 32 Cores	80GB
3	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz – 24 Cores	128GB

	Registry Server (Eureka Server)	RAM
1	(VM)Ampere Altra Neoverse-N1 @ 3GHz – 3vCPU (ARM)	18GB

	Load Balancer & Test (DSPROJECT-SERVER)	RAM
1	Apple M1 Pro @ 3.2GHz – (6+2) Cores (ARM)	16GB


Normalized AVG TIME
1
0.734315529
0.658883391
0.730793635
0.522334626
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Round Robin
Weighted Round Robin
Least Response Time
Least Connection
CustomLoadBalancer

Thank you