로드밸런싱(Load Balancing)

작성일시	@2023년 3월 23일 오후 9:35
강의 번호	CS
유형	
자료	
복습	☐
학습 소스 출처 1	https://www.youtube.com/watch?v=9_6COPOMZvI
학습 소스 출처 2	
날짜	

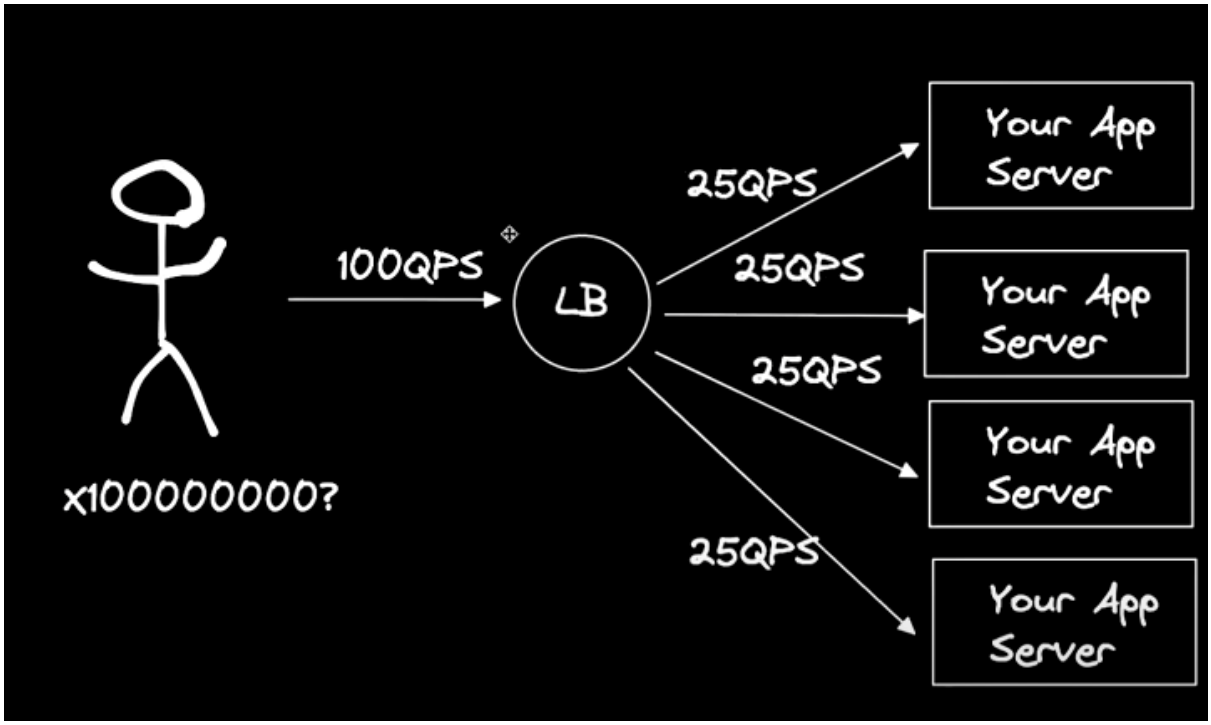
늘어나는 트래픽

웹사이트에 트래픽이 늘어났을때 처리하는 방법에는 두가지가 있다.

Vertical scale up : 서버 자체의 퍼포먼스를 늘리는 방법

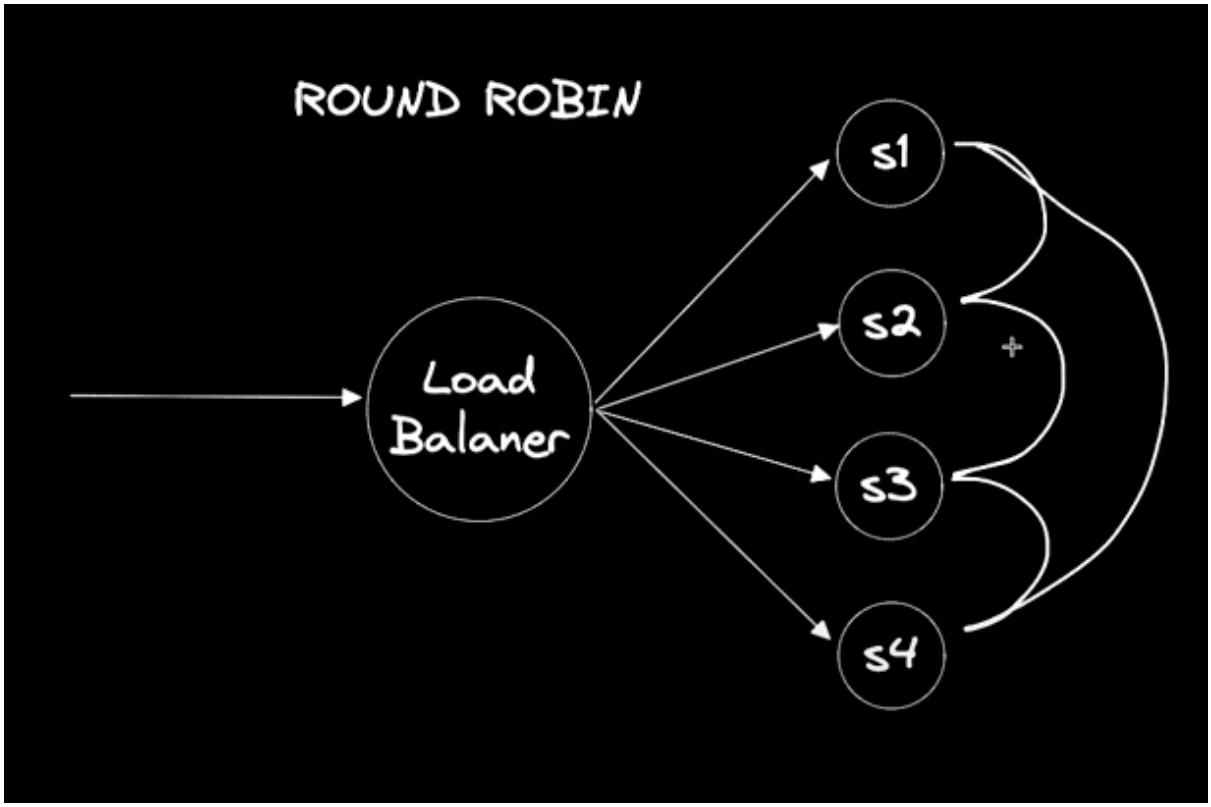
→ 물리적이나 비용적으로 한계가 존재

Horizental scale out : 서버를 분산시스템을 구축하는 방법



로드밸런싱 구현방법

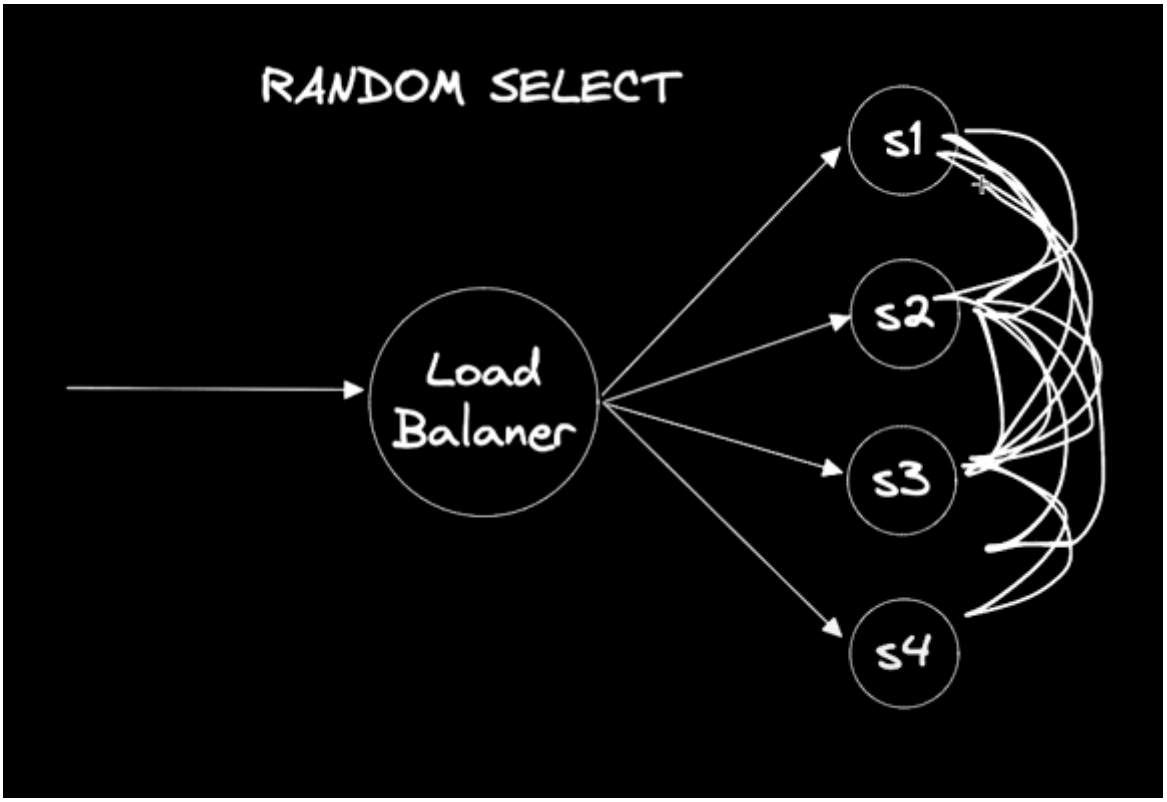
1. 라운드 로빈 방법



라운드 로빈 방법은 서버들에게 균등하게 트래픽을 분배하는 방법입니다. 로드밸런서가 도착한 요청을 순서대로 서버에게 전달하고, 다음 요청은 다음 서버에게 전달합니다. 이 과정을 반복하여 모든 서버가 공평하게 일을 처리하도록 합니다.

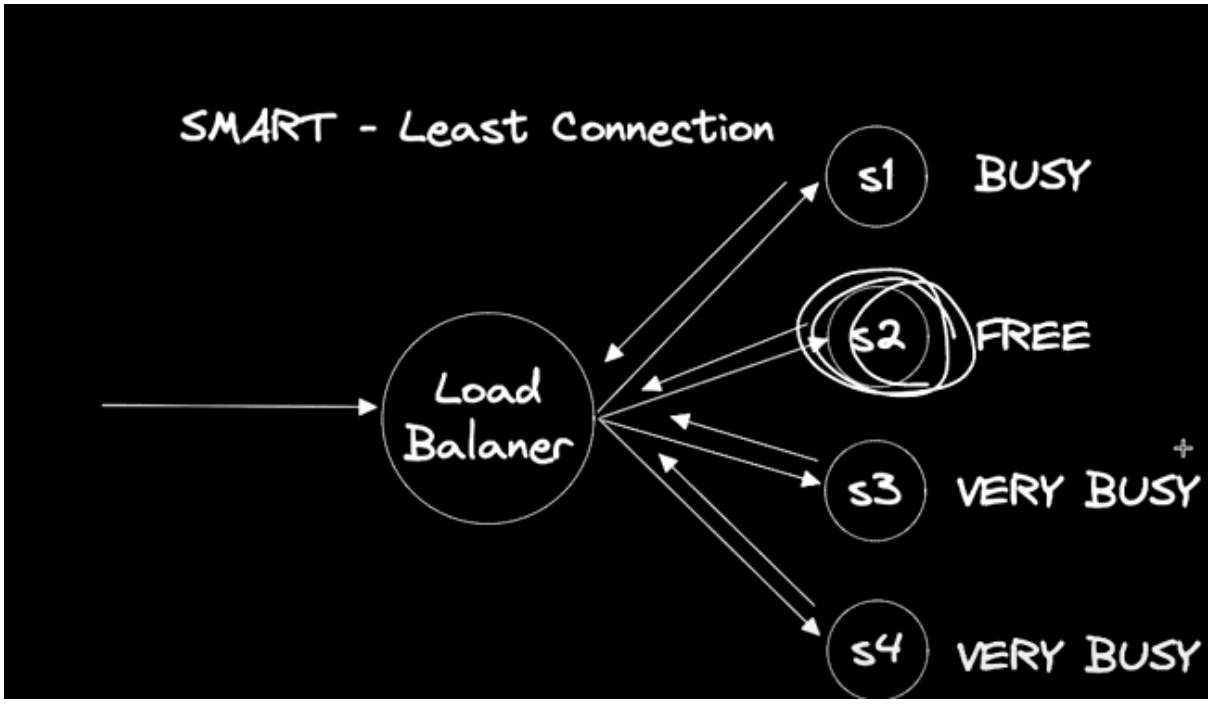
이 방법은 구현하기 쉽고, 서버간의 부하를 균등하게 분배할 수 있습니다. 하지만 서버의 퍼포먼스가 차이가 나면 공평하지 않은 분배가 될 수 있습니다.

2. 랜덤 셀렉트 방법



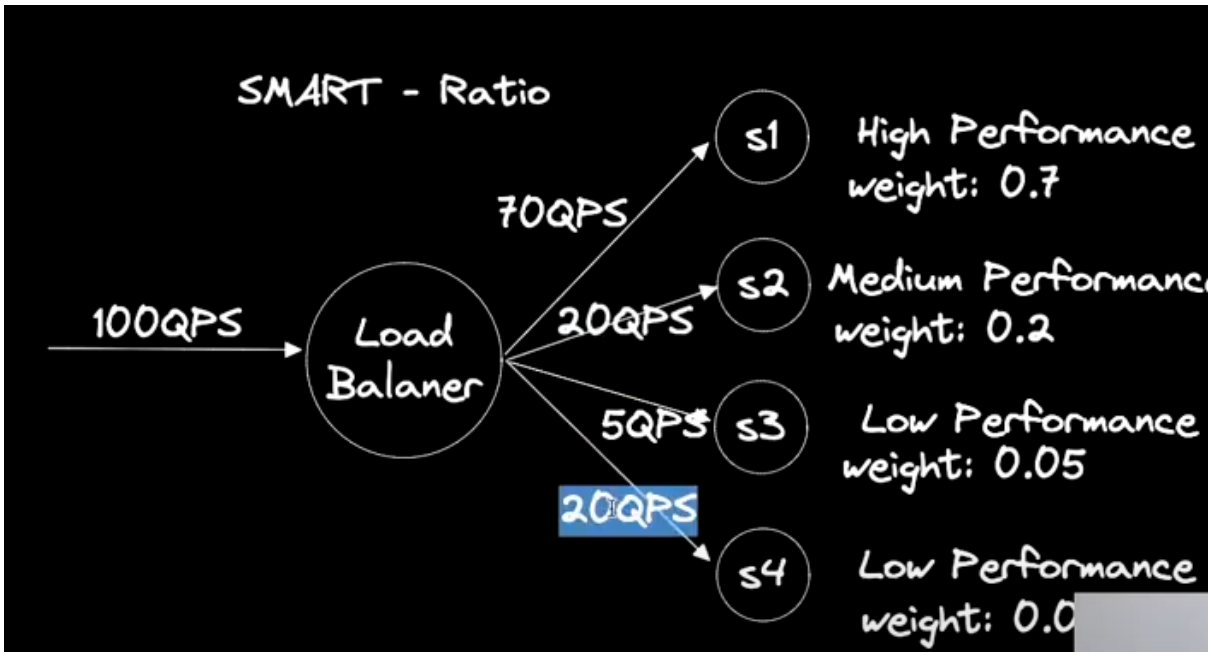
랜덤 셀렉트 방법은 이름 그대로 요청이 들어올 때마다 서버 중 랜덤으로 하나를 선택해 요청을 전달하는 방식입니다. 이 방법은 간단하게 구현할 수 있고, 각 서버에게 균등하게 부하를 분산시킬 수 있습니다. 하지만 랜덤으로 선택하기 때문에 일부 서버에게 불균형한 부하가 생길 수 있습니다.

3. Least Connection 방식



Least Connection 방식은 현재 가장 적은 연결 상태를 가진 서버에 우선적으로 요청을 전달하는 방식입니다. 로드밸런서는 서버의 연결 상태를 모니터링하고, 요청이 들어올 때마다 연결 상태가 가장 적은 서버에게 요청을 전달합니다. 이 방법은 효율적으로 부하를 분산시킬 수 있으며, 서버의 퍼포먼스 차이에도 민감하지 않습니다. 하지만 서버의 연결 상태를 모니터링해야 하기 때문에 구현이 복잡할 수 있습니다.

4. Ratio 방식



로드밸런서는 서버의 트래픽 분배 비율을 설정하는 Ratio 방식도 있습니다.

Ratio 방식은 서버의 퍼포먼스나 연결 상태에 따라 트래픽 분배 비율을 조정할 수 있어, 다른 방식과 함께 사용하면 더욱 효율적인 로드밸런싱을 구현할 수 있습니다.

로드밸런싱의 구현방식

1. 소프트웨어적인 방식

로드밸런서를 소프트웨어적으로 구성하는 방법 중 하나로 HAProxy가 있습니다. HAProxy는 고성능 TCP/HTTP 로드밸런서로, 대규모 서버 클러스터에서 사용할 수 있습니다. 높은 가용성과 부하분산을 제공하며, 다양한 로드밸런싱 알고리즘을 지원합니다. 또한, SSL 엔드투엔드 암호화와 ACL(액세스 제어 목록) 기능 등 다양한 보안 기능을 제공합니다.

로드밸런서의 한 종류인 Reverse proxy 방식은 클라이언트와 서버 사이에 중간 단계로 Reverse proxy 서버를 두는 방식입니다. 클라이언트는 Reverse proxy 서버에 요청을 보내고, Reverse proxy 서버는 이를 적절한 서버로 전달합니다. 이 방식은 보안성이 높고, 캐싱을 통한 성능 향상 등 다양한 장점이 있습니다. 대표적으로 Nginx와 Apache가 사용되며, 소프트웨어적인 방식으로 구현됩니다.

2. 하드웨어적인 방식

로드밸런서를 하드웨어적으로 구성하는 방법으로, 특별한 하드웨어 장비를 사용하여 로드밸런싱을 구현합니다. 이 방법은 대규모 트래픽 처리와 빠른 응답 시간을 제공할 수 있습니다. 또한 물리적으로 존재하므로 물리적인 안정성을 가질 수 있다.

하지만 비용이 매우 높고, 유지보수에 많은 비용이 들어가기 때문에 중소규모 서비스에서는 사용되지 않는 경우가 많습니다.

하나의 서버라도 다운된다면?

SPOF(Single Point of Failure) 상황은 로드밸런싱에서 매우 중요한 문제입니다. 만약 로드밸런서 자체가 다운되거나 로드밸런서와 연결된 서버 중 하나가 다운된다면, 서비스 전체가 중단될 가능성이 큽니다.

이러한 상황을 방지하기 위해서는 로드밸런서와 서버들을 여러 대로 구성하여, 하나가 다운되더라도 다른 서버들이 부하를 분산하도록 구성하는 것이 중요합니다. 이를 통해 SPOF 상황을 최소화할 수 있습니다.

그러나 여러 대의 서버를 구성하는 것만으로는 충분하지 않습니다. 로드밸런서와 서버들은 서로 다른 물리적인 장소에 위치하도록 구성하고, 서버들 간의 독립성을 보장하는 것이 좋습니다. 또한, 백업 로드밸런서를 구성하여 만약 주 로드밸런서가 다운된 경우에도 서비스가 지속될 수 있도록 하는 것이 좋습니다.

마지막으로, SPOF 상황을 예방하기 위해서는 주기적인 모니터링과 이상 징후를 조기에 파악하고 대처하는 것이 매우 중요합니다. 로드밸런서와 서버들의 상태를 모니터링하고, 이상 징후를 조기에 파악하여 대처할 수 있도록 구성하는 것이 필요합니다.

따라서, 로드밸런싱을 구성할 때는 SPOF 상황을 고려하여 여러 대의 서버를 구성하고, 백업 로드밸런서를 구성하며, 주기적인 모니터링과 대처 방안을 마련하는 것이 중요합니다.