Load balancing

트래픽의 증가와 이에 따른 인프라 확장 방식

- 인터넷의 발달은 데이터의 통신을 매우 활발하게 만들었고, 이는 **트래픽의 폭발적인 증가**로 이어졌다.

  → 그 결과, **하나의 서버로는 많은 양의 트래픽을 감당하기 힘들어졌다.**

- 폭발적인 트래픽의 양을 소화하기 위해 두 가지 방안이 세상에 모습을 드러냈다.

Scale Up

- **서버 자체의 성능을 높이는 것을 의미**한다.
  - 하드웨어적 예로, 성능이나 용량 증강을 목적으로 서버에 디스크를 추가하거나, CPU나 메모리를 업데이트 시키는 것을 이야기한다.
- 한 서버의 성능을 높이는 것이기 때문에 **수직 스케일링(Vertical scaling)** 이라고도 부른다.

- **기술의 한계**가 있으므로, **단순히 서버 하나의 성능을 늘리는 것만으로는 많은 트래픽을 소화하기 힘들다.**

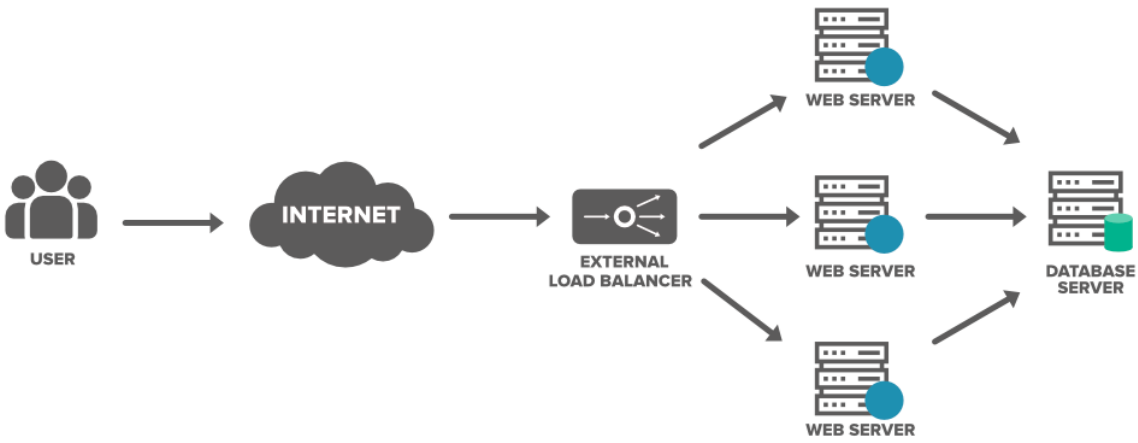
Scale Out

- **비슷한 사양의 서버를 추가로 연결하는 것을 의미한다.**
- **처리할 수 있는 데이터 용량을 늘리고, 기존 서버의 부하를 분담함으로서 성능 향상의 효과**를 기대할 수 있다.
- 서버를 추가로 확장하기 때문에 **수평 스케일링(Horizontal scaling)** 이라고도 부른다.
- **결국 분산 시스템을 구축하겠다는 의미**로 받아들일 수 있고, 여기서 대두되는 개념이 **로드 밸런싱**이다.

로드 밸런싱

- 단순히 다수 서버를 구축해 운영한다고 해서 **모든 클라이언트의 요청에 일관성 있게 응답할 수 있을까?**
  - **압도적인 트래픽을 여러 대의 서버로 분산**해줌으로서 **한 곳의 서버에 모든 트래픽이 몰리는 것을 방지**해야 한다.

    → 이럴 때 사용할 수 있는 기술이 바로 **로드 밸런싱**이다.



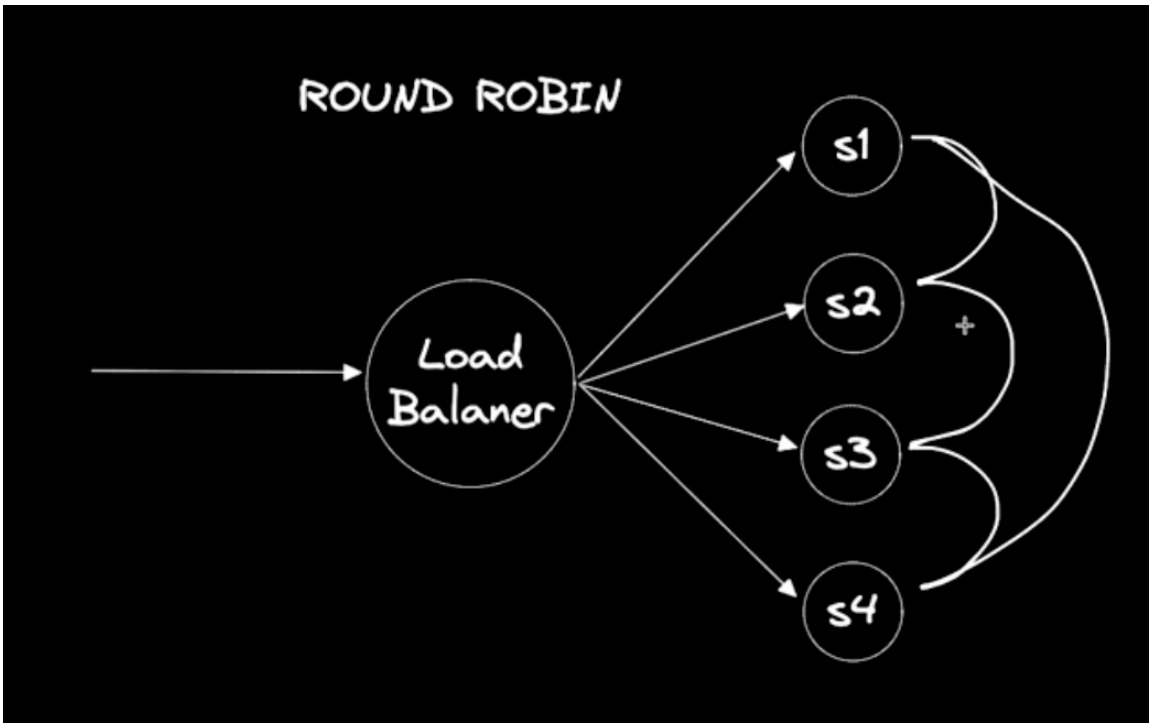
- 그리고 **로드 밸런싱을 구현하기 위해 사용하는 모듈**이 있는데, 이를 **로드 밸런서(Load balencer)**라고 한다.

- 로드 밸런서는 **서버에 가해지는 부하를 분산해주는 장치**로, 클라이언트와 서버풀 사이에 위치한다.
- 한 대의 서버에 부하가 집중되지 않도록 트래픽을 관리해 **각각의 서버가 최적의 성능을 내도록 지원**한다.

로드 밸런싱 알고리즘

- **그렇다면, 로드 밸런스 자체를 어떻게 해야할까?**
  - 요청을 특정 서버에 분배하는 로드 밸런싱 알고리즘에는 여러가지가 있다.

- **라운드 로빈 방식(Round Robin method)**



  - 여러 대의 서버가 있다고 가정했을 때, **들어오는 요청을 순차적으로 하나씩 할당하는 것**을 의미한다.
  - **로드밸러닝 대상 서버의 성능이 동일하고, 처리 시간이 짧은 애플리케이션일 때 주로 사용**한다.
  - **모든 서버가 트래픽을 균등하게 할당받을 수 있어** 심플하면서 좋은 알고리즘이라고 할 수 있다.
  - 일반적으로 널리 쓰인다.

- **랜덤 선택 방식(Random select method)**
  - 어떤 요청이 들어왔을 때 **로드 밸런서가 특정 서버를 랜덤으로 지정해 해당 서버에 요청을 할당하는 방식**이다.
  - 운이 나쁘면 한 서버에 지속적으로 트래픽이 할당될 수 있으나, 흔한 케이스는 아니다.
  - 구현이 쉽고 간단하다.

- **최소 연결 방식(Least connection method)**
  - **서버가 로드 밸런서에게 얼마만큼의 트래픽을 감당할 수 있고, 얼마만큼 연결하고 있는지 역으로 정보를 알려주는 방식**이다.
  - **로드 밸런스는 제공받은 정보를 기반으로 트래픽을 어떤 식으로 할당할지 결정**한다.
  - 서버, 로드 밸런서 간 실시간 통신이 이뤄지므로 복잡한 방법이다.
  - **동적으로 변하는 요청에 대해 부하를 분산시킬 수 있어 효과적**이다.

- **가중 라운드 로빈 방식(Weighted round robin method)**
  - **분산 시스템을 구축할 때 모든 서버의 성능이 동일한 것은 현실적으로 쉽지 않다.**
  - 이에 맞춰 **미리 서버들의 성능을 계산해 가중치를 지정하고, 가중치에 따라 트래픽을 서버에 할당하는 방식**이다.
    - 로드 밸런서가 성능 차이를 고려하지 않고 많은 트래픽을 몰아주면, 한 서버가 다운될 수 있다.
    - 이에 따라 연쇄적으로 다른 서버도 다운되는 것을 방지하는 것이 가중 라운드 로빈 방식이다.

로드 밸런싱을 구축하는 방법

- **소프트웨어적 방식**
  - **로직을 구현해 트래픽을 분산시키는 방법**이다.
  - 로직만 구현하면 되므로 **저렴한 비용으로 구축이 가능**하다.

    - 로드 밸런서 자체를 Scale Out 해야 하는 상황에서도 쉽게 구축이 가능하다.
    - 예로 Apache, HAProxy 가 있다.

- **하드웨어적 방식**
    - **물리적으로 서버를 묶는 것을 의미**한다.
    - 서버는 보통 데이터 센터에 위치하고, 이 **데이터 센터에서 서버를 물리적으로 묶는 것**이다.
    - 물리적 방식으로 서버를 묶는 것이기 때문에 굉장히 안정적이고, 데이터 센터에 아무나 접근할 수 없기 때문에 보안적으로 안전하다.
    - 하지만, 물리적 방법으로 서버를 직접 묶는 것이기 때문에 **비용이 비싸다.**
    - 대표적으로 L4 스위치, L7 스위치가 있다.

로드 밸런서가 다운됐을 때는?

- **대용량 트래픽을 분산시켜주는 모듈은 로드 밸런서**이기 때문에, **로드 밸런서가 다운되면 서비스도 함께 다운된다.**
    - 특정 포인트에서 에러가 발생했을 때 전체 시스템이 다운되는 경우를 **Single Point Of Failure (SPOF)** 라고 부른다.
    - **로드 밸런서를 한 서버에만 구축해 서비스를 제공**하게 되면, **아무리 좋은 방식으로 분산 시스템을 구축하더라도 로드 밸런서가 다운되면 의미가 없다.**

- **이에 따라 로드 밸런서 자체도 Scale Out 을 할 필요가 있다.**
    - 하나의 예시로 **마스터, 슬레이브 방식**을 들 수 있다.
        - **마스터가 항상 트래픽을 처리하되 여벌로 한 두 개정도 로드 밸런서를 가지고 있는 방법**을 의미한다.
        - **마스터가 예상치 못한 이유로 다운됐을 때, 슬레이브를 마스터로 임명해 서비스를 계속 이어나갈 수 있도록 조치할 수 있다.**

References

- https://www.youtube.com/watch?v=9_6COPOMZvI
- https://m.post.naver.com/viewer/postView.naver?volumeNo=27046347&memberNo=2521903
- https://butter-shower.tistory.com/109
- https://tecoble.techcourse.co.kr/post/2021-10-12-scale-up-scale-out/
- https://www.geeksforgeeks.org/load-balancing-approach-in-distributed-system/