
REPORT



HW01

Load balancer

과목명	분산처리	담당교수	남재현 교수님
학 번	32180879	전 공	컴퓨터공학과
		이 름	김원섭

목차

1. 문제 정의	3
2. 아키텍처 구조	3
3. 검증	5
1. 실험 환경 설정	5
2. 컴파일 & 실행	5
4. 성능 테스트	7

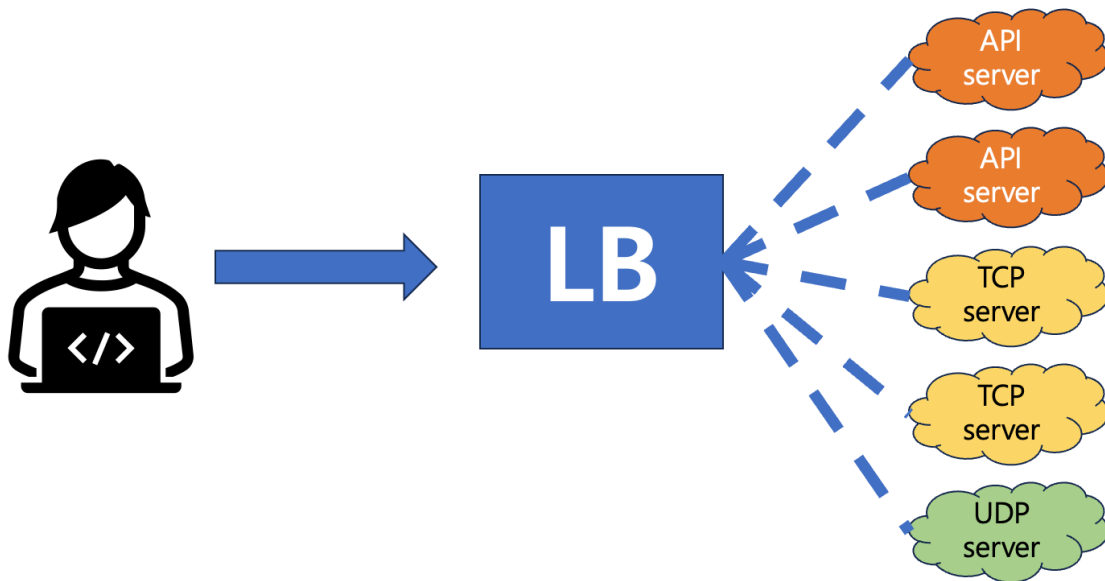
1. 문제 정의

이번 과제에서의 큰 틀은 Load Balancing을 하는 서버를 만드는 것입니다.

Load Balancing이란 서버가 처리해야 할 업무나 요청을 여러 서버로 나누어 처리하는 것을 의미합니다. 이는 하나의 서버로 요청들이 집중되지 않게 관리하여 서버가 최적의 퍼포먼스를 유지할 수 있도록 하는 것을 목표로 합니다.

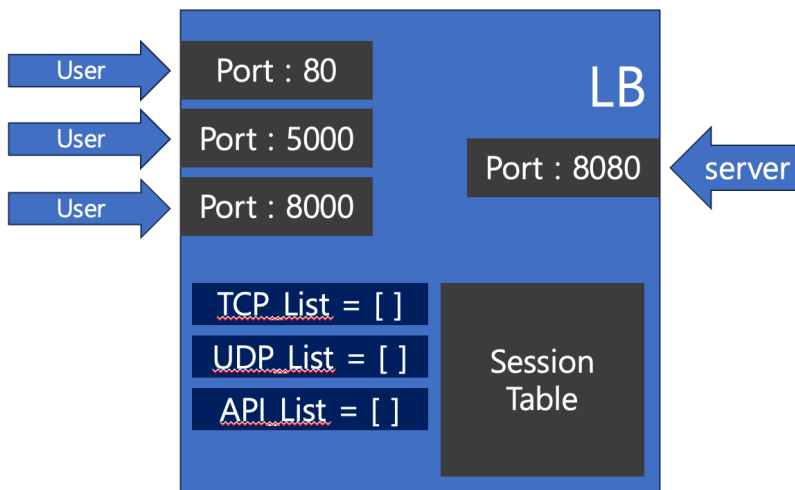
2. 아키텍처 구조

전체적인 구조를 그림으로 나타내면 다음과 같습니다.



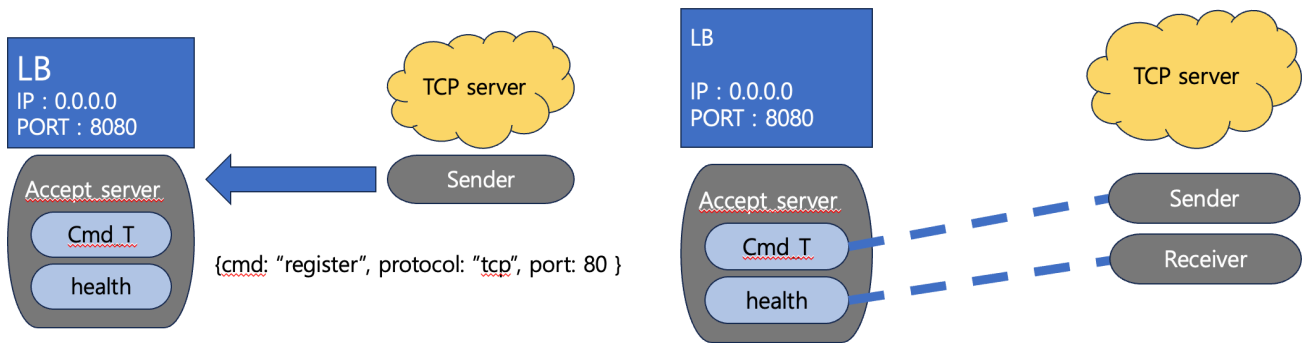
사용자의 요청을 LB(Load Balancer)가 받고 이를 라운드로빈 기법을 적용하여 백서버에 전달합니다. LB에 대해서 설명을 한다면 4개의 포트를 열고 통신 합니다. 이는 tcp, udp, api를 사용하는 유저를 받기 위한 포트 3개 백서버를 연결,관리를 위한 포트 1개 입니다. 그리고 백서버의 정보를 저장하기 위한 리스트와 유저와 백서버의 릴레이

중의 세션을 저장하기 위한
세션 테이블



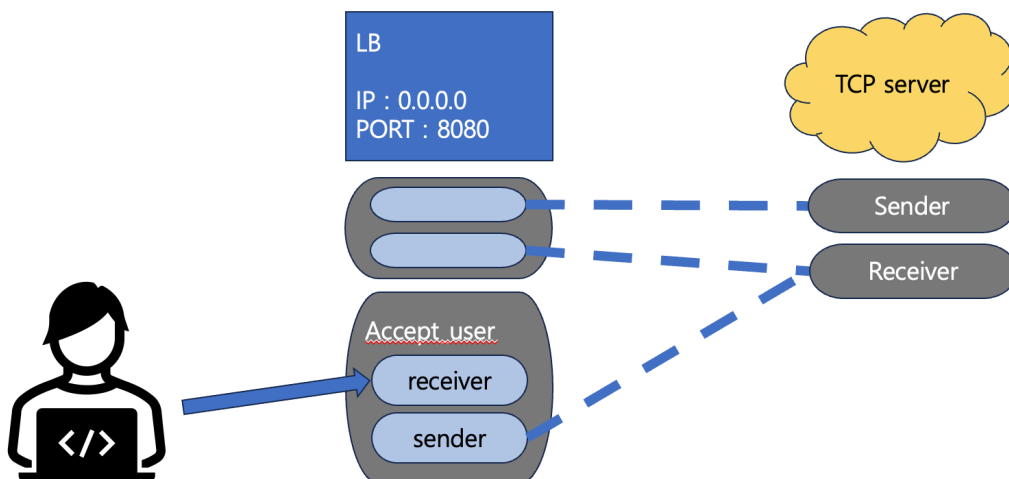
예를 들어서 그림으로 나타내면 다음과 같습니다.

1. 처음으로 백서버가 LB에게 커맨드를 JSON의 형태로 전송을 하면 LB가 처리를 한 후 연결이 되고 리스트에 저장됩니다.

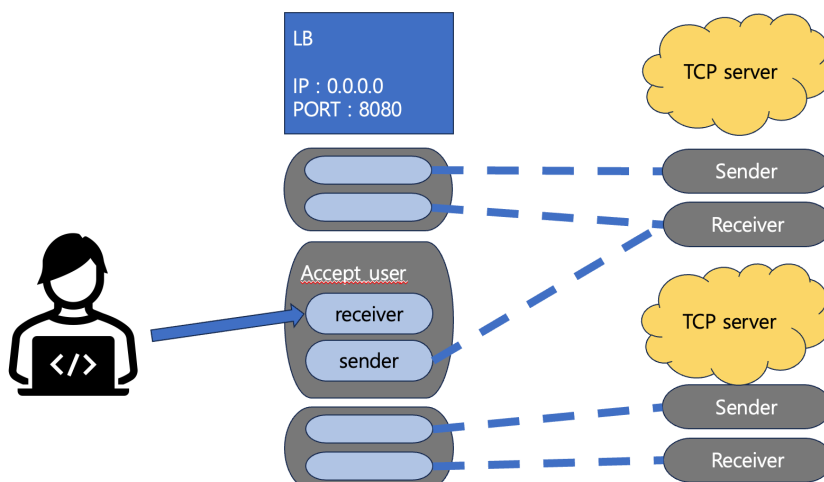


2. 다음으로 유저가 LB에 연결이 된다면

유저한테 받은 데이터를 백서버에게 넘겨 줍니다.



만약 이때 서버가 추가가 된다면 이 세션을 유지하는 세션 테이블을 저장하고 있기 때문에 릴레이 중에는 라운드 로빈이 돌아가지 않습니다.



3. 검증

1. 실험 환경 설정

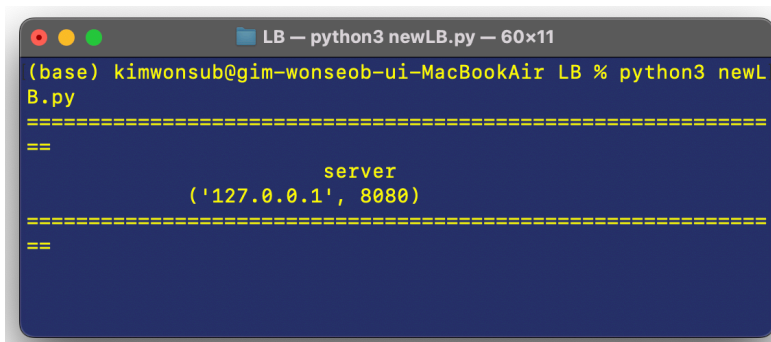
이번 과제에서는 리눅스 환경에서의 실험을 진행합니다. 따라서 제일 먼저 여러 가지 환경 설정이 필요합니다. 우선 파이썬을 설치 하여야 합니다. 만약 우분투 20.04 LTS 버전이라면 파이썬이 설치되어 있기 때문에 안하셔도 됩니다. 파이썬 설치 `sudo apt-get install python3.6` 명령어를 입력 하시면 됩니다. 이렇게 설치를 하면 간단히 `python3` 명령어를 이용하여 파이썬을 실행 하실수 있습니다.

그리고 로드밸런서를 실행하기 위해 기기의 포트에 관한 방화벽을 설정해 주셔야합니다. 방화벽은 `ufw` 명령어를 통해서 관리할수 있는데 일단 `ufw enable` 명령어를 입력하여 방화벽을 활성화 해주시기 바랍니다. 만약 root계정이 아니라면 `sudo`를 붙여 주십시오. 그 다음 `ufw allow 포트/프로토콜` 으로 지정한 서비스나 포트를 허용하여 주시기 바랍니다. 예를 들어 로드밸런서가 8080/TCP를 사용하기 때문에 `ufw 8080/tcp` 를 입력하면 됩니다. 나머지 유저를 받는 포트들을 허용하여 주시면 됩니다. 그리고 서버의 ip는 고정 아이피를 사용하거나 하나의 컴퓨터에서 사용하시면 127.0.0.1을 입력 하시면됩니다.

2. 컴파일 & 실행

이번 과제에서의 코드는 파이썬으로 작성되었기 때문에 따로 컴파일이 필요 없습니다. 따라서 파이썬만 설치가 되어있다면 `python3`라는 명령어를 통해 실행 하시면 됩니다. 만약 가상 환경을 사용하시려면 `python3 -m venv`를 사용하여 가상환경을 만들고 `source` 명령어로 실행하시면 됩니다. LB.py라는 파일이라면 `python3 LB.py`를 터미널에 입력하시면 실행됩니다.

3. 작동 확인



```
LB — python3 newLB.py — 60x11
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python3 newLB.py
=====
==
                        server
                ('127.0.0.1', 8080)
=====
==
```

LB 첫 실행 화면

```
LB — python3 newLB.py — 60x11
B.py
=====
server
('127.0.0.1', 8080)
=====
server connect success : 127.0.0.1 55386
15시 27분 45초 server ('127.0.0.1', 55386) : hello
15시 27분 50초 server ('127.0.0.1', 55386) : hello
15시 27분 55초 server ('127.0.0.1', 55386) : hello
=====

LB — python3 newTCP.py — 60x11
=====
server connect success : 127.0.0.1 55386
15시 27분 45초 server ('127.0.0.1', 55386) : hello
15시 27분 50초 server ('127.0.0.1', 55386) : hello
15시 27분 55초 server ('127.0.0.1', 55386) : hello
{'cmd': 'register', 'protocol': 'tcp', 'port': 9991}
connecting server
connected server
register successful : 127.0.0.1,55386,tcp,9991
=====

LB — python3 newTCP.py — 60x10
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python3 newTCP.py
send cmd(1:register, 2: unregister, q: quit):
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):1

LB — python3 newTCP.py — 60x10
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):1
LoadBalancer : successful
send cmd(1:register, 2: unregister, q: quit):send cmd(1:register, 2: unregister, q: quit):
```

터미널은 LB / 아래 터미널은 TCP 서버

Hello는 Health check의 입력으로 서버가 살아있는지를 확인합니다.

TCP서버 관리자는 1을 입력하여 register 합니다. 그리고 LB에서 저장에 성공한다면 successful이라는 내용을 입력으로 줍니다.

```
LB — python3 newLB.py — 60x7
15시 35분 22초 server ('127.0.0.1', 56338) : hello
15시 35분 27초 server ('127.0.0.1', 56338) : hello
15시 35분 32초 server ('127.0.0.1', 56338) : hello
user connect success : 127.0.0.1 56577
session table : 1
15시 35분 37초 server ('127.0.0.1', 56338) : hello

LB — python3 user_tcp.py — 60x10
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python3 user_tcp.py
LB:connected!
send msg(quit : q):
```

그리고 유저가 접속을 한다면 세션 테이블에 저장합니다.

```
LB — python3 newTCP.py — 60x9
send cmd(1:register, 2: unregister, q: quit):
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):
LoadBalancer : hello
send cmd(1:register, 2: unregister, q: quit):

LB — python3 newLB.py — 60x9
15시 41분 22초 server ('127.0.0.1', 57170) : hello
15시 41분 27초 server ('127.0.0.1', 57170) : hello
15시 41분 32초 server ('127.0.0.1', 57170) : hello
user connect success : 127.0.0.1 57483
session table : 1
user connect success : 127.0.0.1 57484
session table : 2
15시 41분 37초 server ('127.0.0.1', 57170) : hello

LB — python3 user_tcp.py — 60x9
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python3 user_tcp.py
LB:connected!
send msg(quit : q):

LB — python user_tcp.py — 60x9
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python user_tcp.py
LB:connected!
send msg(quit : q):
```

```
LB - python3 newTCP.py - 60x9
send cmd(1:register , 2: unregister , q: quit):
User : 1111
send cmd(1:register , 2: unregister , q: quit):
User : 22222
send cmd(1:register , 2: unregister , q: quit):
LoadBalancer : hello
send cmd(1:register , 2: unregister , q: quit):
LoadBalancer : hello
send cmd(1:register , 2: unregister , q: quit):

LB - python3 newLB.py - 60x9
to tcp ('127.0.0.1', 57174)
15시 43분 49초 server ('127.0.0.1', 57170) : hello
user : 1111
to tcp ('127.0.0.1', 57174)
user : 22222
to tcp ('127.0.0.1', 57174)
15시 43분 54초 server ('127.0.0.1', 57170) : hello
15시 43분 59초 server ('127.0.0.1', 57170) : hello

LB - python3 user_tcp.py - 60x9
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python3 user_
_tcp.py
LB:connected!
send msg(quit : q): hello world 111111111
send msg(quit : q): 111111111
send msg(quit : q): 1111
send msg(quit : q):

LB - python user_tcp.py - 60x9
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python user_
tcp.py
LB:connected!
send msg(quit : q): hello world 222222222
send msg(quit : q): 222222
send msg(quit : q): 22222
send msg(quit : q):

LB - python3 newLB.py - 60x7
15시 35분 57초 server ('127.0.0.1', 56338) : hello
15시 36분 02초 server ('127.0.0.1', 56338) : hello
15시 36분 07초 server ('127.0.0.1', 56338) : hello
user : hello world
to tcp ('127.0.0.1', 56341)
15시 36분 12초 server ('127.0.0.1', 56338) : hello

LB - python3 newTCP.py - 60x9
send cmd(1:register , 2: unregister , q: quit):
LoadBalancer : hello
send cmd(1:register , 2: unregister , q: quit):
LoadBalancer : hello
send cmd(1:register , 2: unregister , q: quit):
User : hello world
send cmd(1:register , 2: unregister , q: quit):
LoadBalancer : hello
send cmd(1:register , 2: unregister , q: quit):

LB - python3 user_tcp.py - 60x10
(base) kimwonsub@gim-wonseob-ui-MacBookAir LB % python3 user_
_tcp.py
LB:connected!
send msg(quit : q): hello world
send msg(quit : q):
```

4. 성능 테스트

이번 과제에서 API 서버 구현과 성능테스트를 하지 못했습니다.