

컴퓨터네트워크

실습 1 및 숙제 1 웹 서버 성능 테스트

이영석

October 12, 2018 (Friday)

1 개요

많은 사용자가 서버에 몰릴때 서버에는 당연히 과부하가 발생한다. 이런 상황에서도 서버는 모든 클라이언트에게 정상적으로 서비스를 제공할 수 있어야 한다. 서비스가 느려질 경우 사용자의 서비스 만족도가 떨어지고 이는 기업의 수익에도 영향을 줄 수 있다. 유명 웹 서비스는 서버가 얼마나 많은 클라이언트를 허용 가능한지 파악하기 위해 성능 테스트를 진행한다. Apache jmeter, HTTPUnit 등의 오픈 소스가 성능 테스트 기능을 제공한다. 이번 주 실습에서는 라즈베리파이를 설치하여 물리적인 서버를 설치한다. 실제 서버에서 직접 구현한 HTTP Server를 실행시킨다. 성능 테스트 프로세스를 구현하여 HTTP Server의 성능을 측정한다.

2 실습

2.1 Python BeautifulSoup

```
from bs4 import BeautifulSoup

def parseHTML(html_source):
    with open("test_50kb.html", "r") as fp:
        soup = BeautifulSoup(fp)

    img_list = soup.find_all('img')
    return img_list
```

2.2 Python Visualization

```
import matplotlib.pyplot as plt

value1 = [82,76,24,40,67,62,75]
value2=[62,5,91,25,36,32,96,95,3,90,95,32,27,55]
value3=[23,89,12,78,72,31,25,52]

box_plot_data=[value1,value2,value3]

plt.boxplot(box_plot_data)
plt.show()
```

3 숙제: 웹 서버 성능 테스트

3.1 싱글 클라이언트 - 성능 측정 순서

1. 프로세스 시작 타임스탬프(startTime) 기록
2. 클라이언트 소켓을 생성하여 서버와 연결.
3. 서버에 샘플 웹 페이지 html 요청 전송, 응답 수신
4. html의 `img` 태그를 파싱하여 html이 참조하는 다른 웹 자원의 경로를 파악
5. 참조 웹 자원을 연속적으로 요청, 응답 수신
6. 모든 웹 자원의 요청,응답이 완료되면 다운로드 종료 타임스탬프(endTime) 기록
7. 다운로드 시간 = $\text{startTime} - \text{endTime}$
8. 다운로드 시간 기록

3.2 멀티 클라이언트 - 성능 측정

1. Python Process Pool을 활용하여 프로세스 N개를 생성한다.
2. 각각의 클라이언트는 싱글 클라이언트 성능 측정을 실행한다.
3. N이 10, 20, 30일때 각 클라이언트에서 기록된 다운로드 시간을 시각화한다.

3.3 주의사항

- 마감날짜: 2018년 10월 18일(목) 18:00 (딜레이 1일당 20% 감점, 5일 지난 후 0점)
- 제출방법: 충남대학교 사이버캠퍼스 교과목 숙제 게시판
- 제출할 것: 숙제리포트 pdf 파일, 파이썬 코드 파일, 결과 캡처 파일, 시각화 이미지
- 담당조교: 민지원 (minjiwon@cs-cnu.org , 공대 5호관 633호)

References