Load test for Web services

Hyunchan, Park

http://oslab.jbnu.ac.kr

Division of Computer Science and Engineering

Jeonbuk National University

개인 과제 #3

- 제출: 하나의 ppt에 각 캡처 파일을 넣은 후, PDF 파일로 변환해 LMS “과제 2” 제출
  - 파일 이름: 학번.pdf
  - Page #1: 제목, 학번, 이름
  - Page #2~#5: 아래 캡처 화면 하나씩
- 리눅스 인스턴스 생성 후, HTTP로 접속한 첫 화면
  - Page #2
- Wordpress 구성 후, 관리자 화면
  - Page #3
- AB 부하 테스트 결과
  - Page #4~#5
- 기한: 10/12 (월) 23:59
  - 지각 감점: 5%p / 12H
  - 1주 이후 제출 차단
- 과제 수행 후, 인스턴스 중지!

AB 부하 테스트

- Instance types
  - t2.micro, t2.medium 각 1대에 대해 테스트
  - 2개의 인스턴스가 동작 중인 EC2 콘솔 화면: page #4
  - 각 인스턴스에 대한 테스트 결과: page #5
- 각 성능 결과 캡처
  - 2개의 JPG (page #4, page #5)

참고 자료

- AWS 기반 웹 및 애플리케이션 서버 부하 테스트: A to Z
  - https://aws.amazon.com/ko/blogs/korea/how-to-loading-test-based-on-aws/

- “AWS와 부하 테스트의 절묘한 만남”, 김무현,AWS 솔루션즈아키텍트

부하 테스트

- 웹서비스의 품질을 파악하기 위한 테스트
  - 현재 서비스 구성의 제한(limit)을 찾기 위함
  - 원하는 부하를 수용할 수 있게끔 구성되었는지 확인하기 위함
  - 병목 지점을 찾고 병목 현상을 제거하기 위함

- 우리가 만든 웹서비스의 성능을 테스트하기 위해 사용
  - 얼마나 많은 요청을 처리할 수 있을까? (현재의 HA 구성에서)
  - 많은 요청 수에 대해 자동으로 처리 성능을 조절할 수 있을까?
    - 차후 Auto Scaling 의 동작 및 성능 확인에 사용

부하 테스트와 스트레스 테스트

부하 테스트 (Load Test)[1]

소프트웨어 시스템에 요청을 보내서 응답을 측정하는 절차

- to determine a system's behavior under both normal and anticipated peak load conditions
- to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation

스트레스 테스트 (Stress Test)[2]

시스템의 안정성을 결정하기 위해서 수행되는 의도적인 심한 테스트. 일반적인 운영 용량을 넘은 테스트를 수행하여 결과를 관찰함

- to determine breaking points or safe usage limits
- to confirm mathematical model is accurate enough in predicting breaking points or safe usage limits
- to confirm intended specifications are being met
- to determine modes of failure (how exactly a system fails)
- to test stable operation of a part or system outside standard usage

[1] 출처: https://en.wikipedia.org/wiki/Load_testing
[2] 출처: https://en.wikipedia.org/wiki/Stress_testing

Apache AB

참고자료:

https://httpd.apache.org/docs/2.4/programs/ab.html

http://blog.hkwon.me/ab-apache-http-server-benchmarking-tool/

AB: Introduction

- 아파치 웹서버 성능검사 도구
  - Apache HTTP Server 의 간단한 성능 벤치마킹 도구
    - 정적 컨텐츠, REST API 등
  - 아파치 서버 패키지에 포함
  - 특히 아파치가 현재 초당 몇개의 요청을 서비스하는지 알려줌
- 유의할 점
  - AB는 서버의 응답에 걸리는 시간만 측정
    - Server-side centric benchmark
  - 사용자가 실제로 느끼는 체감 성능은?
    - HTML translation, image file loading, and etc.
    - Request 간의 delay를 줄 수 없기 때문에, 실제 요청 패턴과 차이가 있음

AB: usage and options

Usage: ab [options] [http[s]://]hostname[:port]/path

옵션	설명
-n	성능을 검사하기위해 보내는 요청수. 기본값으로 요청을 한번만 보내기때문에 일반적인 성능검사 결과를 얻을 수 없다.
-c	동시에 요청하는 요청수. 기본적으로 한번에 한 요청만을 보낸다.
-g	측정한 모든 값을 'gnuplot' 혹은 TSV (Tab separate values, 탭으로 구분한 값) 파일에 기록한다. 라벨은 output 파일의 첫번째 라인을 참고한다.
-t	성능을 검사하는 최대 초단위 시간. 내부적으로 -n 50000을 가정한다. 정해진 시간동안 서버 성능을 검사할때 사용한다. 기본적으로 시간제한 없이 검사한다.
-v	출력 수준을 지정한다. 4 이상이면 헤더에 대한 정보를, 3 이상이면 (404, 202, 등) 응답코드를, 2 이상이면 경고(warning)와 정보(info)를 출력한다.
-A	프록시를 통해 BASIC Authentication 정보를 제공한다. :로 구분한 사용자명과 암호를 **base64 인코딩**하여 전송한다.
-X	proxy[:port] 프록시 서버를 사용하여 요청한다.

AB: Example

- ab -c 50 -t 10
  - 서버가 10초동안 동시에 50개의 요청을 지속적으로 처리
  - 즉, 서버에 50개의 요청 부하가 지속적으로 몰려있는 상황
    - 50명의 사용자가 아니라, n명의 사용자가 무작위로 요청을 보내는데,
    - 서버 입장에서 볼 때, 언제나 50개의 요청이 대기 큐에 쌓여있는 것
  - 성능이 좋을수록, 총 처리된 요청 개수가 많아질 것.

- ab -n 500 -c 10
  - 서버가 10개의 동시 요청을 총 500개 처리하는 시나리오
  - 대기큐에 10개의 요청이 항상 쌓여있는데, 총 요청 수는 500개
  - 성능이 좋을수록, 총 수행 시간이 짧아질 것.

AB: Result (-c 500 -t 10)

```
Concurrency Level:      500
Time taken for tests:   10.001 seconds
Complete requests:      2447
Failed requests:        0
Total transferred:      34507594 bytes
HTML transferred:       33479854 bytes
Requests per second:    244.67 [#/sec] (mean)
Time per request:       2043.566 [ms] (mean)
Time per request:       4.087 [ms] (mean, across all concurrent requests)
Transfer rate:          3369.47 [Kbytes/sec] received
```

* 아래 네모 표시
① 1초당 처리한 요청 수
② 클라이언트에서 각 요청이 응답을 받기까지 걸린 평균 시간
③ 서버에서 각 요청을 처리하는데 걸린 시간
④ 전송 속도

AB: Result

```
Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        1   10  89.8      1     999
Processing:    81 1807 512.9   1899    2740
Waiting:       71 1807 512.9   1898    2740
Total:         81 1817 520.6   1905    3137

Percentage of the requests served within a certain time (ms)
  50%   1905
  66%   2022
  75%   2103
  80%   2175
  90%   2333
  95%   2495
  98%   2660
  99%   2726
 100%   3137 (longest request)
```

AB: Result Analysis

- Failed requests

```
Failed requests:        497
   (Connect: 0, Receive: 0, Length: 497, Exceptions: 0)
```

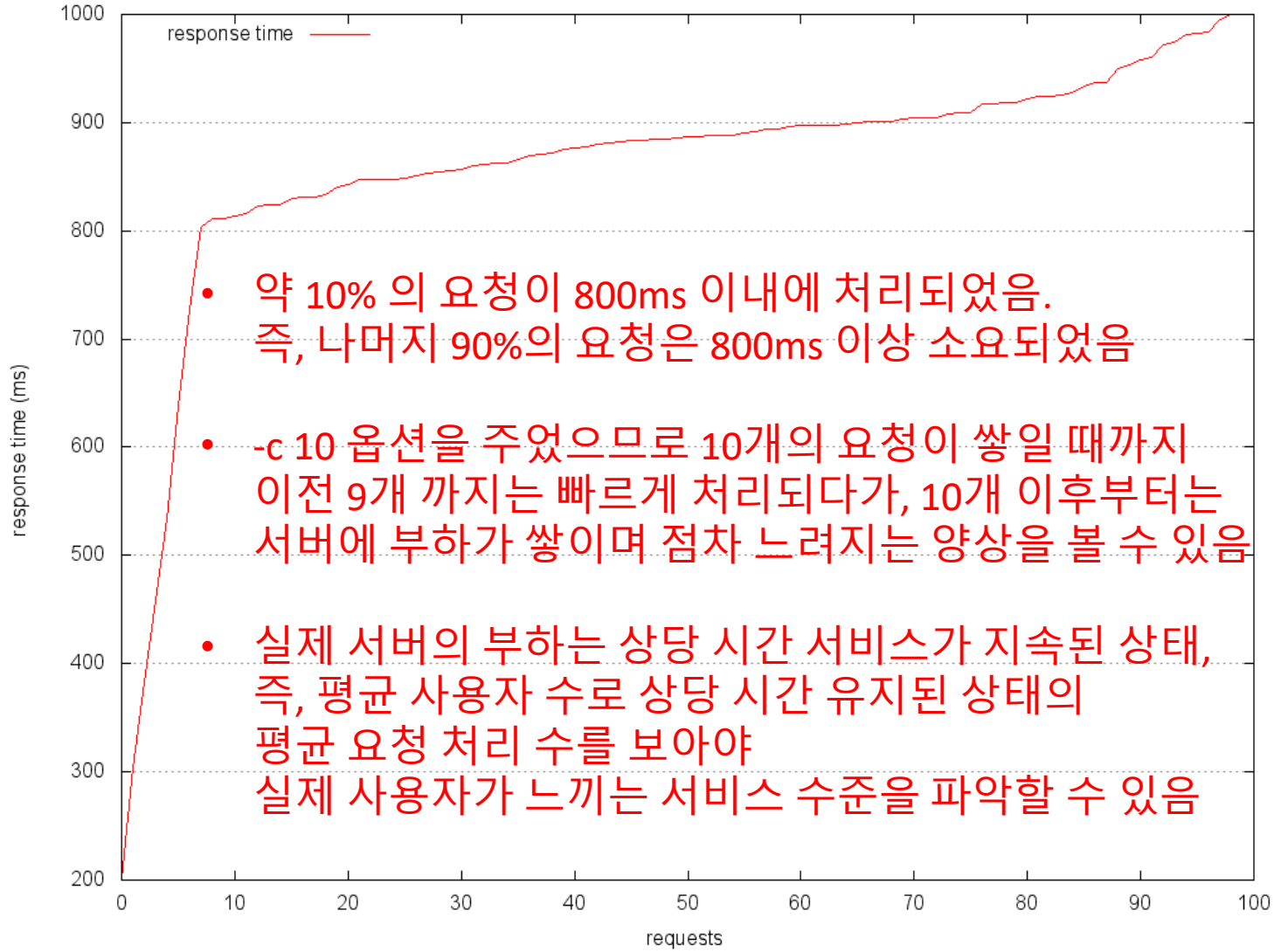
  - Fail 이 있을 경우, reliability 의 훼손
    - 즉, 서버가 제공할 수 있는 범위를 벗어남
      - 벤치마킹 중간에 정지될 수 있음
    - Length 제외: 첫 요청에 대한 응답과 다른 길이의 응답이 오는 경우를 측정함. 동적인 컨텐츠의 경우, 지속적으로 발생할 수 있음

- Response time
  - 요청 처리 시간의 표준 편차가 너무 크거나,
  - 요청 처리 시간 백분위에서 tail 이 길게 형성이 되거나,
  - 요청 처리 시간 자체가 너무 긴 경우,
  - 서비스의 품질이 사용자 요구사항에 크게 미달할 수 있음

Long tail of Response time

$ ab -n 100 -c 10 -g result.plot http://www.google.com/index.html

참고 사이트



- 약 10% 의 요청이 800ms 이내에 처리되었음.
  즉, 나머지 90%의 요청은 800ms 이상 소요되었음
- -c 10 옵션을 주었으므로 10개의 요청이 쌓일 때까지 이전 9개 까지는 빠르게 처리되다가, 10개 이후부터는 서버에 부하가 쌓이며 점차 느려지는 양상을 볼 수 있음
- 실제 서버의 부하는 상당 시간 서비스가 지속된 상태, 즉, 평균 사용자 수로 상당 시간 유지된 상태의 평균 요청 처리 수를 보아야 실제 사용자가 느끼는 서비스 수준을 파악할 수 있음

Instances: 1 and 2, Command: ab -c 100 -t 10

```
Concurrency Level:      100
Time taken for tests:   10.079 seconds
Complete requests:      439
Failed requests:        0
Total transferred:      6190778 bytes
HTML transferred:       6006398 bytes
Requests per second:    43.56 [#/sec] (mean)
Time per request:       2295.934 [ms] (mean)
Time per request:       22.959 [ms] (mean, ac
Transfer rate:          599.82 [Kbytes/sec] r

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.6      0       9
Processing:    76 2005 503.7   2162    2498
Waiting:       74 2005 503.7   2162    2498
Total:         76 2005 503.4   2163    2499

Percentage of the requests served within a ce
  50%   2163
  66%   2192
  75%   2238
  80%   2285
  90%   2323
  95%   2352
  98%   2396
  99%   2406
 100%   2499 (longest request)
```

```
Concurrency Level:      100
Time taken for tests:   10.008 seconds
Complete requests:      955
Failed requests:        0
Total transferred:      13467410 bytes
HTML transferred:       13066310 bytes
Requests per second:    95.42 [#/sec] (mean)
Time per request:       1047.982 [ms] (mean)
Time per request:       10.480 [ms] (mean, acr
Transfer rate:          1314.10 [Kbytes/sec] r

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        1    1   0.5      1      16
Processing:    33  926 748.0    690    2196
Waiting:       33  926 748.0    690    2196
Total:         34  927 748.0    691    2197

Percentage of the requests served within a cer
  50%    690
  66%   1446
  75%   1650
  80%   1759
  90%   2032
  95%   2112
  98%   2160
  99%   2182
 100%   2197 (longest request)
```

Instances: 4 and 8, Command: ab -c 100 -t 10

```
Concurrency Level:      100
Time taken for tests:   10.005 seconds
Complete requests:      1718
Failed requests:        0
Total transferred:      24227236 bytes
HTML transferred:       23505676 bytes
Requests per second:    171.72 [#/sec] (mean)
Time per request:       582.355 [ms] (mean)
Time per request:       5.824 [ms] (mean, acr
Transfer rate:          2364.79 [Kbytes/sec] r

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.3      0       4
Processing:    33  545 467.8    399    1473
Waiting:       33  545 467.8    399    1473
Total:         33  545 467.7    399    1473

Percentage of the requests served within a cer
  50%    399
  66%    800
  75%    929
  80%   1055
  90%   1259
  95%   1321
  98%   1389
  99%   1420
 100%   1473 (longest request)
```

```
Concurrency Level:      100
Time taken for tests:   10.004 seconds
Complete requests:      2557
Failed requests:        0
Total transferred:      36058814 bytes
HTML transferred:       34984874 bytes
Requests per second:    255.61 [#/sec] (mean)
Time per request:       391.224 [ms] (mean)
Time per request:       3.912 [ms] (mean, acr
Transfer rate:          3520.10 [Kbytes/sec] r

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        1    1   0.3      1      10
Processing:    38  378 227.9    316     832
Waiting:       36  378 227.9    316     832
Total:         39  379 227.9    317     833

Percentage of the requests served within a cer
  50%    317
  66%    557
  75%    606
  80%    621
  90%    671
  95%    737
  98%    785
  99%    801
 100%    833 (longest request)
```

t3.large = 8 x t3.micro

t3.2xlarge = 32 x t3.micro

Micro를 필요에 따라 여러 개 사용하는게 좋을까?
아니면, 좀더 높은 사양의 인스턴스를 1개만 사용하는게 좋을까?

범용 – 현재 세대

t3.nano	2	변수	0.5GiB	EBS 전용	시간당 0.0065 USD
t3.micro	2	변수	1GiB	EBS 전용	시간당 0.013 USD
t3.small	2	변수	2GiB	EBS 전용	시간당 0.026 USD
t3.medium	2	변수	4GiB	EBS 전용	시간당 0.052 USD
t3.large	2	변수	8GiB	EBS 전용	시간당 0.104 USD
t3.xlarge	4	변수	16GiB	EBS 전용	시간당 0.208 USD
t3.2xlarge	8	변수	32GiB	EBS 전용	시간당 0.416 USD

Large and 2xLarge, Command: -c 100 -t 10

```
Concurrency Level:      100
Time taken for tests:   10.024 seconds
Complete requests:      835
Failed requests:        0
Total transferred:      10918460 bytes
HTML transferred:       10599490 bytes
Requests per second:    83.30 [#/sec] (mean)
Time per request:       1200.453 [ms] (mean)
Time per request:       12.005 [ms] (mean, ac
Transfer rate:          1063.73 [Kbytes/sec]

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.5      0       6
Processing:    37 1115 194.5   1149    1550
Waiting:       33 1115 194.5   1149    1550
Total:         37 1116 194.1   1149    1550

Percentage of the requests served within a ce
  50%   1149
  66%   1180
  75%   1203
  80%   1218
  90%   1257
  95%   1313
  98%   1377
  99%   1423
 100%   1550 (longest request)
```

```
Concurrency Level:      100
Time taken for tests:   10.001 seconds
Complete requests:      2525
Failed requests:        93
   (Connect: 0, Receive: 0, Length: 93,
Non-2xx responses:      93
Total transferred:      32083180 bytes
HTML transferred:       31122536 bytes
Requests per second:    252.48 [#/sec] (
Time per request:       396.072 [ms] (me
Time per request:       3.961 [ms] (mean
Transfer rate:          3132.87 [Kbytes/

Connection Times (ms)
              min  mean[+/-sd] median
Connect:        0    0   0.4      0
Processing:    40  388  78.8    395    1
Waiting:       38  388  78.8    395    1
Total:         41  388  78.8    395    1

Percentage of the requests served within
  50%    395
  66%    408
  75%    420
  80%    429
  90%    445
  95%    465
  98%    516
  99%    525
 100%   1143 (longest request)
```

Large and 2xLarge, Command: -c 1000 -t 10

```
Concurrency Level:      1000
Time taken for tests:   10.015 seconds
Complete requests:      783
Failed requests:        0
Total transferred:      10238508 bytes
HTML transferred:       9939402 bytes
Requests per second:    78.18 [#/sec] (
Time per request:       12790.834 [ms]
Time per request:       12.791 [ms] (me
Transfer rate:          998.33 [Kbytes/

Connection Times (ms)
              min  mean[+/-sd] median
Connect:        0   12   2.3     12
Processing:   126 5152 2768.3   5180
Waiting:      105 5152 2768.4   5180
Total:        126 5164 2768.9   5192

Percentage of the requests served withi
  50%   5191
  66%   6701
  75%   7550
  80%   7994
  90%   9039
  95%   9461
  98%   9718
  99%   9836
 100%   9910 (longest request)
```

```
Concurrency Level:      1000
Time taken for tests:   10.001 seconds
Complete requests:      4688
Failed requests:        2306
   (Connect: 0, Receive: 0, Length: 2306, Exc
Non-2xx responses:      2306
Total transferred:      38148048 bytes
HTML transferred:       36454084 bytes
Requests per second:    468.76 [#/sec] (mean)
Time per request:       2133.282 [ms] (mean)
Time per request:       2.133 [ms] (mean, acr
Transfer rate:          3725.09 [Kbytes/sec]

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    4   7.0      0      22
Processing:    62 1708 1116.1   1676    5881
Waiting:       29 1708 1116.1   1676    5881
Total:         62 1712 1113.4   1677    5901

Percentage of the requests served within a ce
  50%   1677
  66%   2028
  75%   2410
  80%   2672
  90%   3219
  95%   3453
  98%   4360
  99%   5319
 100%   5901 (longest request)
```

과제 수행 후, 인스턴스 “중지”

- 앞으로 이 인스턴스를 기반으로 계속 실습 진행
- Name tag 를 지정하여 구분해둘 것
- 실습을 수행하지 않을 때는 반드시 “중지” 하여 Credit 절약
  - 10GB 스토리지를 한 달 간 사용하면? 약 $1