SAMSUNG SW ACADEMY FOR YOUTH

對外秘

프로젝트 명세서

공통 PJT 결과에 대한

성능 테스트 수행

對外秘

목차

1. 프로젝트 개요

예를 들어 여러분들이 수강 신청 웹 사이트를 만들었다고 가정해 봅시다. 이 사이트는 수강신청 기간에 최소한 수천 명 이상의 접속을 견딜 수 있어야 합니다. 이런 경우 여러분들은 웹 사이트를 구축한 후에 성능 테스트를 통해 이런 상황을 견딜 수 있다는 것을 보장해야 할 것입니다.

여러분들이 공통 PJT 기간중에 구축했던 과제물에 대해 성능 테스트를 해 보고 만일 원하는 성능이 나오지 않는다면 어떻게 해야 할 지에 대해 생각해 보도록 합시다.

참고로 성능 테스트의 종류 및 목표 부하를 기술해 보겠습니다.

필요 상황에 따라 성능 테스트의 종류는 보통 아래와 같이 분류됩니다.

1. 목표 성능 도달 여부 확인 (Load testing)
2. 한계 성능 측정 (Spike Testing)
3. 한계 초과 부하 중에서도 기능 완전성 체크 (Stress Testing):어느 정도 부하가 걸린 상태에서 기능들이 얼마나 정상적으로 작동하는가를 체크

도메인별 일반적인 요구 성능 수준은 아래와 같습니다.

NO	도메인	최대 가용 접속자(서버당)	예상 동시 접속자(서버당)
1	게임서버(베틀그라운드 등)	200,000 유저	10,000 유저
2	커뮤니티(페이스북 등)	20,000 유저	1,000 유저
3	일반 웹 사이트	4,000 유저	200 유저

2. 과제

여러분들이 공통 기간에 완성한 과제에 대해 성능 테스트를 수행하고 결과를 기록합니다.

성능테스트에 사용될 도구는 JMeter 최신버전(현 시점에서 5.2.1) 입니다.
(https://jmeter.apache.org/download_jmeter.cgi)

상용 및 오픈소스로 나온 도구들이 많이 있으며, 오픈소스 계열에서는 JMeter 가 항상 TOP7 중 상위에 랭크되고 있을 정도로 많이 쓰이고 있습니다. (참고 URL : https://www.dotcom-tools.com/blog/best-open-source-load-testing-tools/)

그 외에 최근에 많이 쓰이는 도구로는 Google 진영에서 밀고 있는 로커스트(https://locust.io/) 라는 도구도 있습니다. Python 스크립트 지원 및 풍부한 가상 유저수 지원이 장점이기는 하지만 , 스크립트 레코딩을 지원을 하지 않는다는 단점이 있습니다. (참고 URL : https://medium.com/@giljae/jmeter-vs-locust-무엇을-써야-할까-1d1e0769c08))

JMeter 로 진행할 작업은 다음과 같습니다.

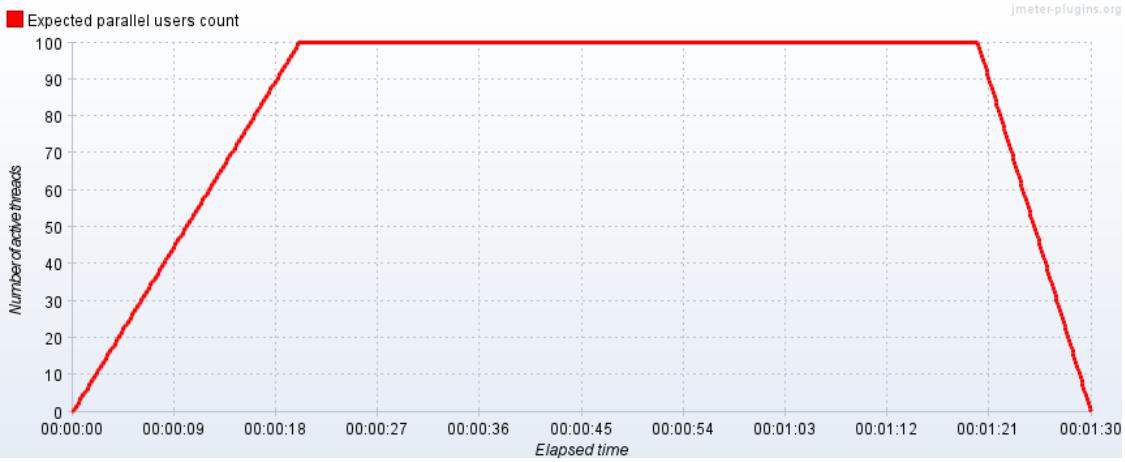
[공통 - 웹 과제 수행 팀의 경우]

1. 측정 시나리오 설정
   - 보통 사용자들이 제일 많이 접하는 로그인 기능을 대상으로 설정합니다.
   - 로그인을 위해서 테스트 ID/PW 를 준비해야 합니다.
2. 100 개의 쓰레드(여기서 1 쓰레드=1 사용자)로 StartupTime:20sec, Hold Load:60sec, Shutdown:10sec 로 맞춘 뒤 (하단 이미지 참조) 아래 항목을 측정
   - TPS(Transaction Per Second) : 시스템의 전반적인 성능 지표
   - 응답시간

對外秘

[공통 - 임베디드 과제 수행 팀의 경우]

1. 측정 시나리오 설정
   - 백앤드 REST API 서버의 성능 측정을 수행함
   - 가장 많이 사용될 것 같은 REST API 의 URL 을 선정.
2. 100 개의 쓰레드(여기서 1 쓰레드=1 사용자)로 StartupTime:20sec, Hold Load:60sec, Shutdown:10sec 로 맞춘 뒤 (하단 이미지 참조) 아래 항목을 측정
   - TPS(Transaction Per Second) : 시스템의 전반적인 성능 지표
   - 응답시간



[주의사항]

공통 프로젝트로 웹 과제를 한 경우는 AWS 에 배포한 결과물을 대상으로 시나리오 레코딩 후에 레코딩된 스크립트로 테스트를 수행합니다.

임베디드 과제를 한 경우에는 현재 임베디드 키트가 여러분에게 없기 때문에 백엔드(AWS 에 배포한 장고, 스프링의 REST API 서버 등)을 대상으로 합니다. 이 경우는 단일 API URL 만 테스트 합니다.

이 과제는 개인과제입니다. **개인이 각각 AWS 에 부하를 보내는 경우 조원(같이 수행했던 친구들)들이 동시에 발생시키면 AWS 에 엄청난 부하가 걸려버립니다. 스크립트 까지는 동시에 작업이 가능하지만 테스트 수행은 조원들과 협의를 하여 서로 다른 시간대에 수행을 하셔야**

합니다.

3. 심화 과제

- 만일 90 초동안 100 유저를 버티지 못했다면 어디를 어떻게 개선해야 이를 해결할 수 있을 지 생각해 봅시다.

- 90 초동안 100 유저 테스트에 성공했다면, 유저 수를 조정해 가면서 대상 시스템의 한계 성능(TPS, 한계 유저수)를 찾아 봅시다.

4. 산출물 제출

산출물은 아래 내용이 들어간 자유 양식(md 파일 등)으로 제출 바랍니다.

1. 측정 결과(그래프는 이미지 캡쳐)
    - TPS 그래프
    - 응답속도 그래프
    - 테스트 성공/실패 판단 여부
2. 심화 과제(선택입니다) : *.md 파일에 기록
    - 1 이 성공인 경우 : 한계 성능(TPS, 한계 유저수)
    - 1 이 실패인 경우 : 개선 방안