

# 자기주도 온라인 프로젝트 I

Day 1.

## JMeter를 활용한 성능 측정

김성준 컨설턴트



Day 3

# CONTENTS

Samsung Software Academy For Youth



## Jmeter를 활용한 성능 측정

김성준 Project consultant

- STA테스팅 컨설팅 수석 컨설턴트
- NHN Japan(현 LINE Japan) QA팀
- CJ 올리브네트웍스 CJWorld 팀
- 삼성SDS 그룹통합 PJT추진팀



# 성능 테스트란...



**성능 테스트란 특수한 상황(예:수강신청 기간 등)에서 정상적으로 시스템의 작동을 보증하는지 확인하기 위한 테스트이며 테스트 목적은 크게 3가지가 있습니다.**

- 1. 목표 성능 도달 여부 확인**
- 2. 한계 성능 측정**
- 3. 부하 중에서의 기능 수행 체크**

# 주요 용어

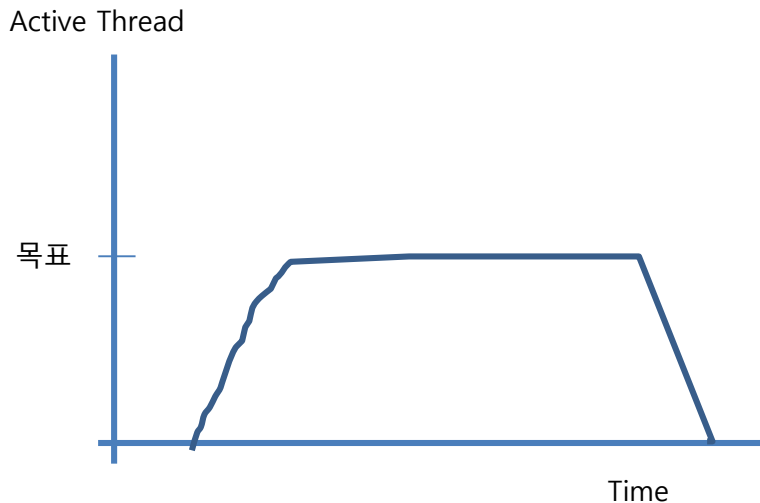


트랜잭션	TPS	응답 시간
화면 조작을 통한 Request 발생 ~ Response를 받기까지의 동작을 의미함	1초에 처리할 수 있는 트랜잭션 수를 의미이며 성능 테스트의 중요한 지표.	Request ~ 최종 화면 표시까지 걸리는 시간이며 영향을 주는 변수들이 많아 상위 90% 값을 최대 값으로 삼는다.

# 성능 테스트 종류



## 1. 부하 테스트(Load Test)



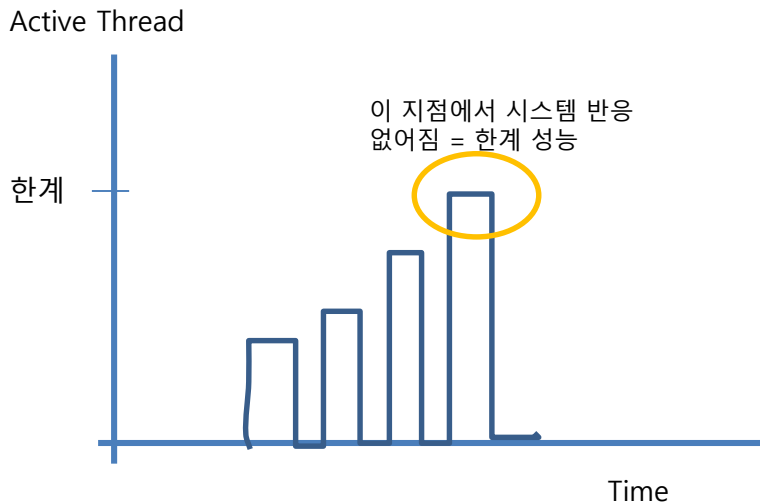
목표 성능 도달 여부를 측정하기  
위함.

성능 목표치는 요구사항에서  
주어지며 이를 도달하지 못하면  
도달을 위한 개선작업 必

# 성능 테스트 종류



## 2. 스파이크 테스트(Spike Test)



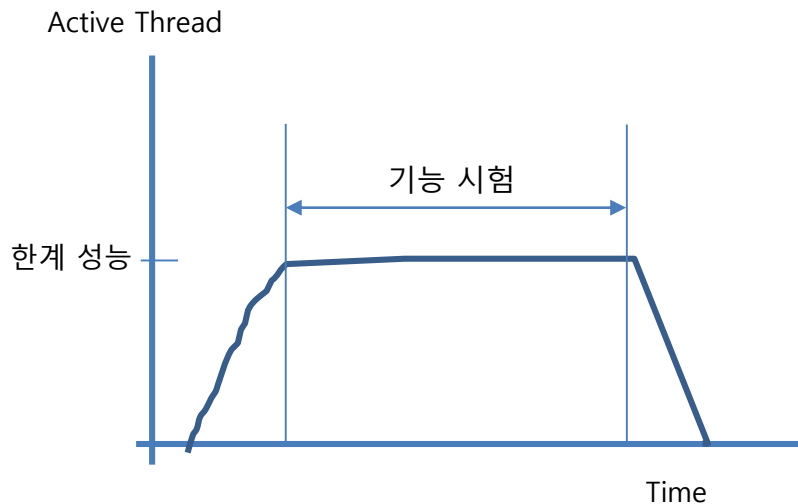
스파이크 테스트 다음과 같은 목적이 있음.

1. 일시에 부하가 몰릴 때  
시스템이 버틸 수 있는지 체크
2. 한계 성능 체크

# 성능 테스트 종류

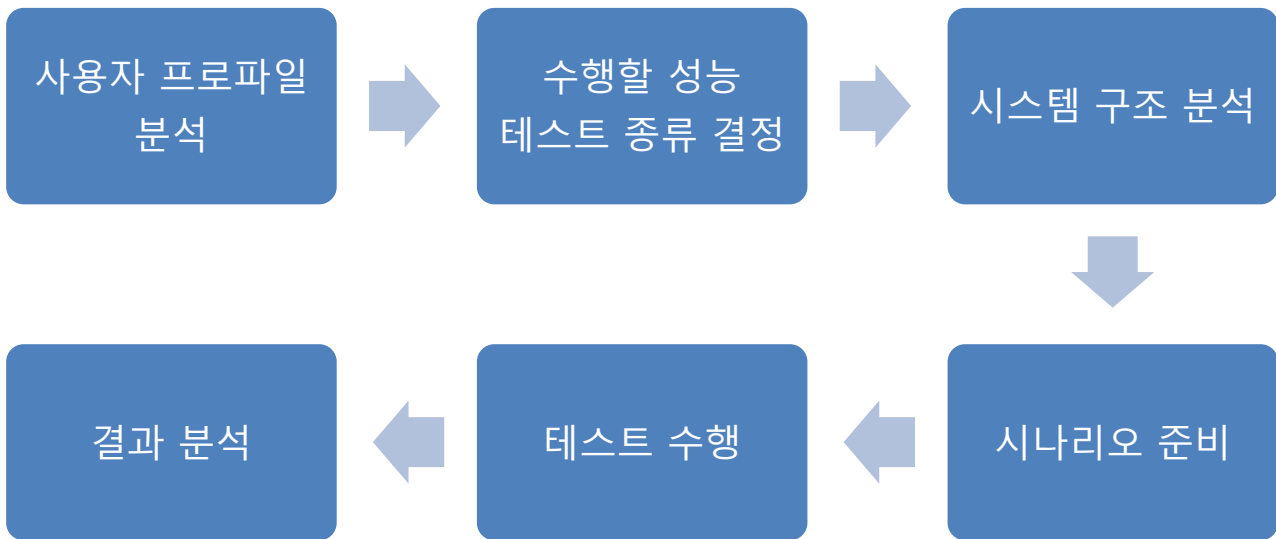


## 3. 신뢰성 테스트



한계 부하가 걸린 상태에서  
시스템의 모든 기능들이 어느  
정도 정상 운용되는지 테스트  
하는 것이며 이 결과는 SW품질  
특성 중 '신뢰성'을 나타내는  
지표로 활용됨

# 성능 테스트 수행 흐름





# 성능 테스트 도구



상용	오픈 소스
Load Runner (Micro Focus) Load UI Pro (SmartBear) ...	Jmeter nGrinder Gatling Locust ...

# Jmeter 소개



← → ↻ 🔒 https://www.dotcom-tools.com/blog/best-open-source-load-testing-tools/ 📄 ☆ ⚙

**dotcom-tools**  
by Dotcom-Monitor

Pro Solutions Free Tools About Dotcom-Monitor Blog [LOGIN](#) [FREE TRIAL](#)

Type and hit enter ... 🔍

**Web Performance Tools Reviewed**

Application Monitoring Tools

**Quick Links**

[Start Monitoring!](#)

[Dotcom-Monitor Home](#)

[Knowledge Base](#)

[Video Tutorials](#)

**Blog Categories**

[Design](#)

[Load Testing](#)

[Performance Testing](#)

[Tech News](#)

[Website Speed](#)

**AUG 12**



## The 7 Best Open Source Load Testing Tools of 2019

All open source load testing tools don't have the same functionality and some will better suit to your needs than others.

# 프로젝트 수행 안내



간단한 구조로 진행합니다.

- Jmeter Server-Client 같은 구조를 사용하지 않습니다.

가급적 Jmeter 버전은 5.x로 진행합니다.

- Ultimate Thread Group이용

성능 측정은 AWS 에 배포 한 공통 과제를 대상으로 합니다..

- npm run serve 등 개발 툴로 띄워놓은 것은 No.
- 본인 PC에서 nginx등으로 띄운 것은 OK

# 주의 사항



- 성능 테스트 대상은 공통 과제시 수행한 결과물을 대상으로 하며, 웹 및 임베디드는 아래 기능을 대상으로 수행합니다.

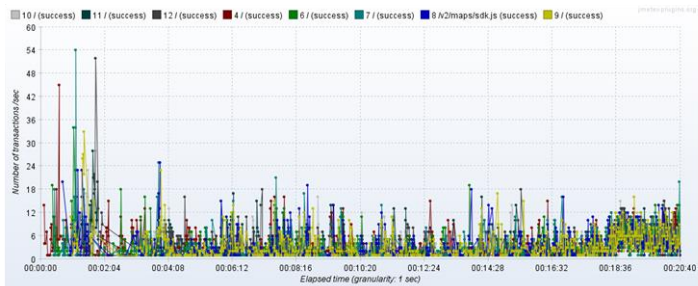
웹 과제	임베디드 과제
<ul style="list-style-type: none"><li>로그인 페이지를 대상</li><li>만일 로그인 페이지가 없다면 사용자가 많이 접할 것 같은 다른 페이지 선정</li></ul>	<ul style="list-style-type: none"><li>백엔드 REST서버 중 많이 사용될 API URL을 대상</li></ul>
<ul style="list-style-type: none"><li>테스트용 ID/PW 필요</li></ul>	

- 이 과제는 개인과제입니다. 개인이 각각 AWS에 부하를 보내는 경우 조원(같이 수행했던 친구들)들이 동시에 발생시키면 AWS에 엄청난 부하가 걸려버립니다. 사전 작업은 상관없지만 테스트 수행은 조원들과 협의를 하여 서로 다른 시간대에 수행을 하셔야 합니다.

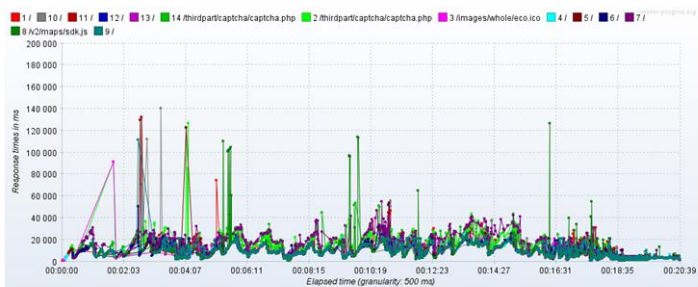
# SAMPLE(1)



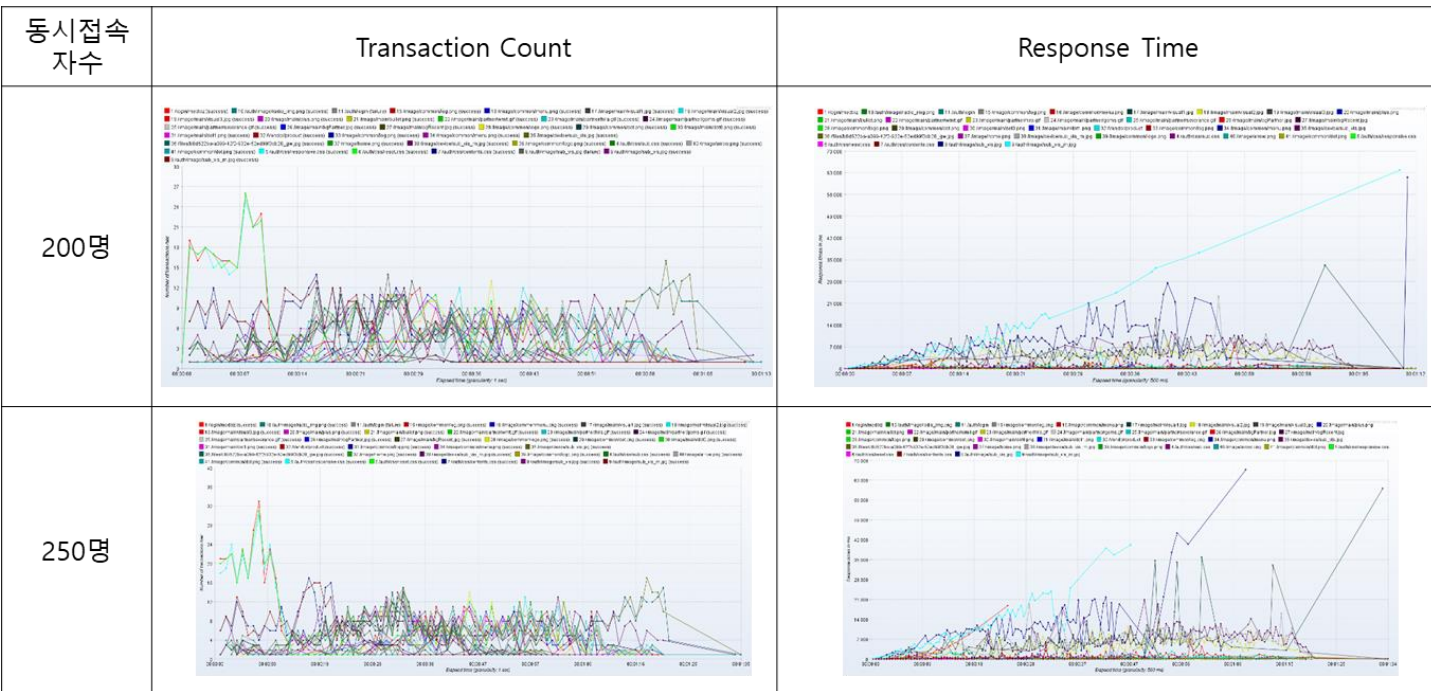
- Transaction은 초반에 다소 올라갔지만, 이후 안정적인 수치를 보이고 있음
- Hang 등이 발생하지 않았음



- Response Time은 스트레스 테스트로 높은 평균 수치를 기록하나 스트레스 환경이므로 용인할 만한 부분임
- 전반적으로 20sec안쪽으로 측정되며, 안정적인 결과를 보여주고 있음



# SAMPLE(2)



# 생각해 봅시다



1. 90초간 100유저를 발생시켰는데 성능 테스트가 실패한 경우

- 응답시간이 끝없이 증가한 경우 또는
- Transaction이 0 가까이 떨어지는 경우

이를 개선하기 위해 어떤 작업을 해야 할까요

2. (테스트에 실패하지 않았을 경우)우리 시스템의 한계 성능은 얼마까지 나올까요?

- 최대 사용자는?
- TPS는?

# Q & A

