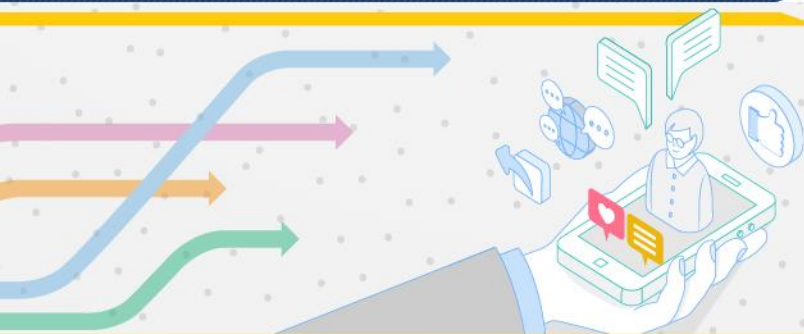


애플리케이션 테스트 수행 part 2



성능 테스트 실행

학습내용

- 성능 테스트 실행
- 성능 테스트 실행 결과 분석

학습목표

- 애플리케이션 성능 테스트를 실행할 수 있다.
- 애플리케이션 성능 테스트 결과를 분석할 수 있다.

성능 테스트 실행

1 성능 테스트 케이스 수행 예시

① 테스트 목표



상황을 고려한 Workload Model이 적용된
테스트로 38.23 TPS를 목표로 함

② 목표치

사용자 수	10,000	호출 간격	44.4
TPS	> 38.23	응답시간	Read < 1Sec.

③ 측정항목

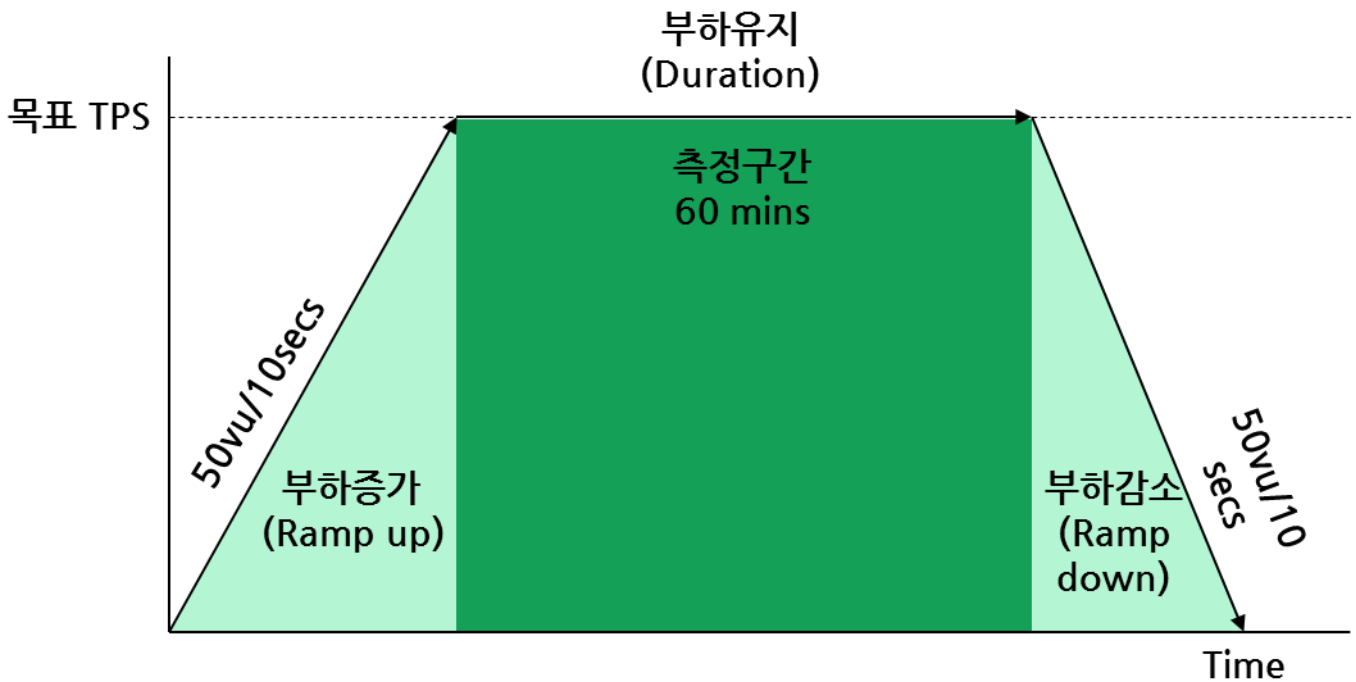


거래 성공 비율, TPS, 응답시간, 시스템 사용률

성능 테스트 실행

1 성능 테스트 케이스 수행 예시

4 시나리오



Ramp-UP Model

(100 Users/3초) * 51초(51초 후에 1,000 Users 유입, 측정 시작)

1,000 Users 유입 완료 후 2분 동안 부하 발생, 성능 측정

Ramp-Down Model

(300 Users/초) * 30초(30초 후에 시험 종료)

Think Time Model

호출 간격(1.4초) - Real 응답 시간

성능 테스트 실행

1 성능 테스트 케이스 수행 예시

5 사전 확인 사항

테스트 시작 시간	10:00(AM)	테스트 종료 시간	14:00(PM)
스크립트 수행(N번)	50	부하 발생기 상태 확인	
데이터베이스 상태 확인		시나리오 확인	Load_1, Load_2
Configuration 확인		테스트 데이터 Import	Model_01

2 ReadyAPI를 이용한 성능 테스트 수행

The screenshot displays the ReadyAPI 2.4.0 interface. On the left, a list of test cases is visible, including 'Stack List Size' and 'MCode'. The main window shows the '작업 관리자' (Job Manager) tab, which provides a dashboard for monitoring system resources during the test. The dashboard includes a CPU usage graph (92% at 1.86GHz), memory usage (4.9/7.9GB at 62%), disk usage (0(C: D:) at 7%), and network status. A table on the right lists system specifications: CPU (Intel(R) Pentium(R) 3805U @ 1.90GHz), Memory (16GB), Storage (2335GB), and Network (79321Mbps). The interface also includes tabs for Dashboard, Projects, SoapUI, Secure, LoadUI, and ServiceV.

성능 테스트 실행 결과 분석

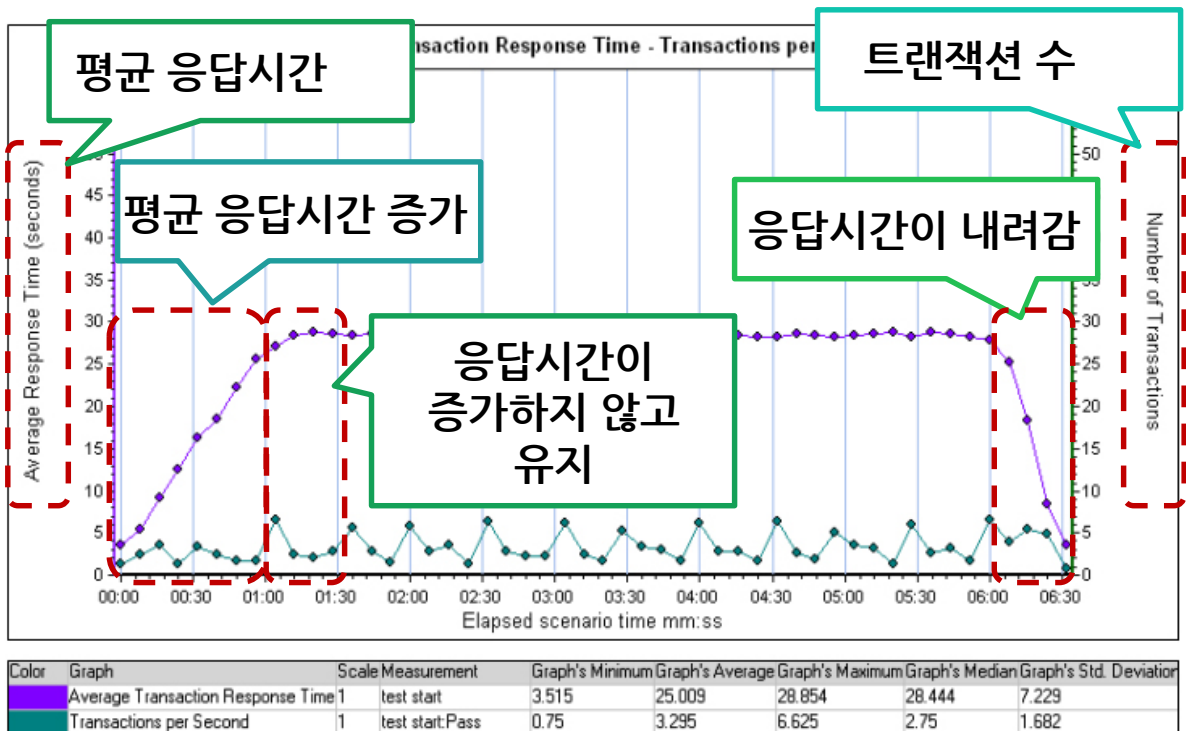
1 LoadRunner

1 성능 테스트 실행 결과 수집

- ▶ 성능 테스트 대상 서버 CPU, 메모리, 디스크 사용량 수집
- ▶ TPS 변화량, 응답시간 수집

2 성능 테스트 실행 결과 분석

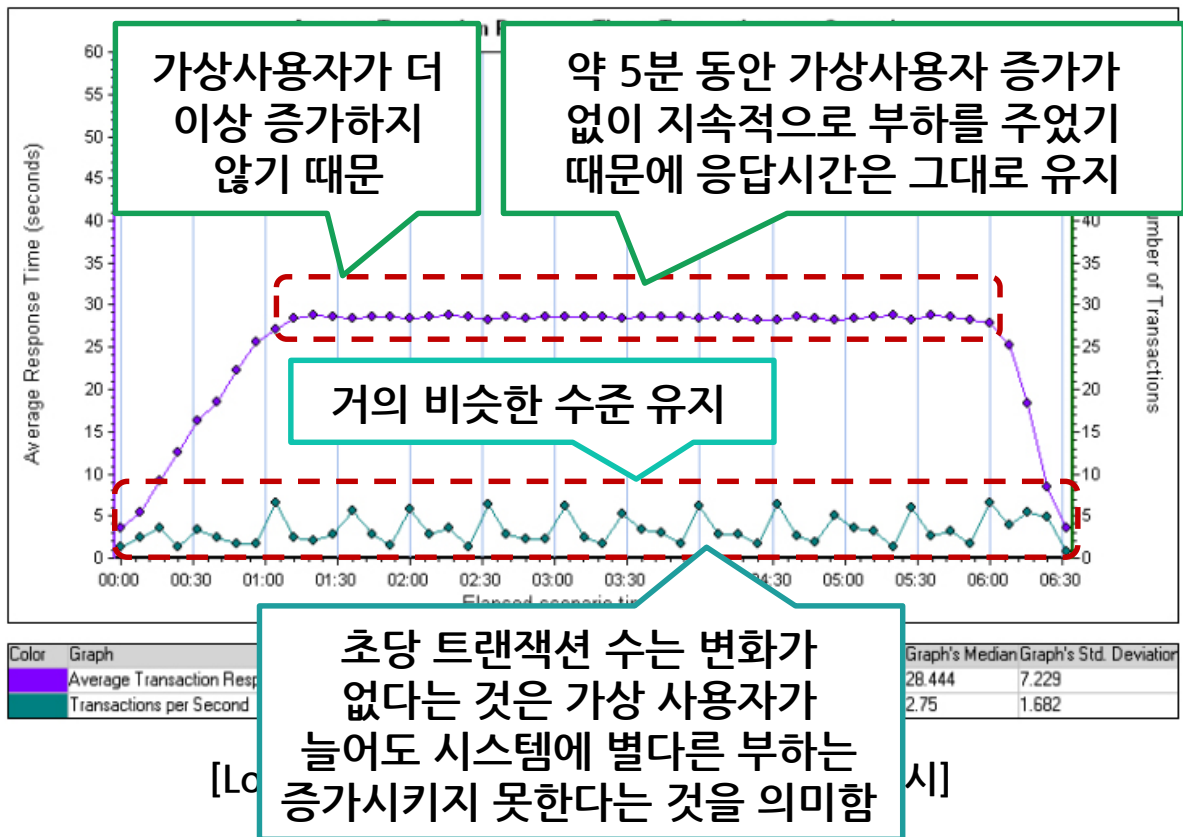
- ▶ 목표 TPS 도달여부 분석



[LoadRunner를 활용한 성능 테스트 결과 예시]

성능 테스트 실행 결과 분석

1 LoadRunner



성능 테스트 실행 결과 분석

1 LoadRunner

CPU, 메모리, 디스크 확인

- 성능 테스트 대상 서버의 **CPU, 메모리, 디스크** 사용량 확인

응답시간, TPS 확인

- 성능 테스트 대상 서버의 **TPS, 응답시간** 변화 분석



부하를 주입하였을 때에 서버의 **CPU, 메모리, 디스크, 응답시간, TPS**가 목표 수치에 도달하였는지 확인이 중요

2 J-Meter

1 오픈 소스 도구

2 TPS와 응답시간 관계를 잘 보여줌

3 해당 업무 단위 성능 테스트 분석

- ▶ 성능 테스트의 목표는 구체적으로 주어짐
- ▶ 업무별로 주어지며, TPS 단위 사용

성능 테스트 실행 결과 분석

2 J-Meter

오픈
소스도구

주로 간단한 프로그램에 대한 부하
테스트 수행

TPS와
응답시간

TPS와 응답시간 간의 관계를 그래프로
보여줌

오픈 소스 도구로서 주로 소규모 프로그램 성능 테스트에 활용

성능 테스트 수행자는 업무별로 몇 TPS가 목표로 설정되었는지 확인

성능 테스트 실행 결과 분석

2 J-Meter



오픈 소스 도구인 J-Meter를 활용한 성능 테스트 수행 결과 예시



[J-Meter를 활용한 성능 테스트 사례]

주로 간단한 프로그램에 대한 부하 테스트를 수행하는데 활용

가상사용자 증가에 따른 TPS와 응답시간 간의 관계를 그래프로 보여줌

가상 사용자를 증가시킴에 따라 TPS와 응답시간의 변화를 시간에 따라서 보여줌

성능 테스트 실행 결과 분석

2 J-Meter

예 오더(A)의 목표 TPS는 4.56 TPS

오더(A)
애플리케이션에
부하를 줌

초당 4.56개
트랜잭션 수행

목표 성능
달성

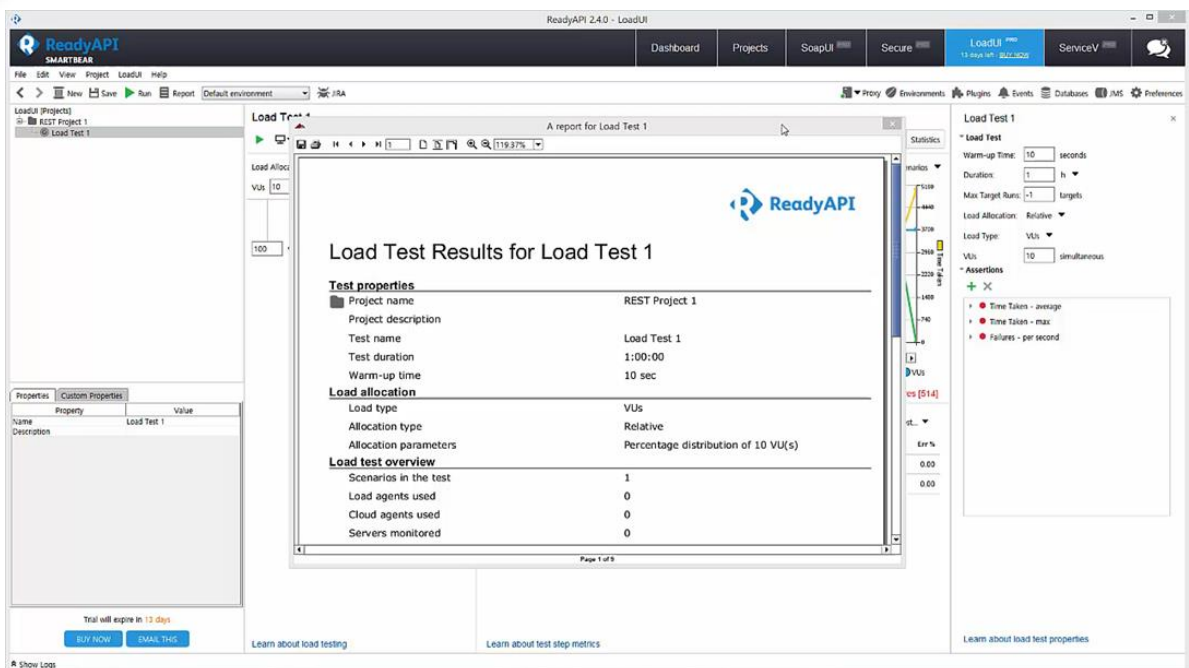


성능 테스트
수행자

애플리케이션 성능 테스트 수행시 목표 TPS 기록

테스트 결과를 보면서 목표 달성 여부 확인

3 RealAPI



1. 성능 테스트 실행

- 성능 테스트 수행자는 애플리케이션 성능 테스트 계획서에 따라 성능 테스트를 수행함
- 성능 테스트 수행 중에 서버의 CPU, 메모리, 디스크 사용률을 확인함
- 성능 테스트 수행 중에 응답시간, TPS, 가상 사용자 수를 확인함
- TPS(Transaction Per Second) : 초당 트랜잭션의 개수

2. 성능 테스트 실행 결과 분석

- 성능 테스트 수행자는 성능 테스트 도구에서 나온 성능 테스트 결과 나온 보고서를 확인함
- LoadRunner : 성능 테스트 대상 업무에 대해 명확하게 확인, 입력값과 출력값도 같이 확인
- 성능 테스트 수행자는 업무별로 몇 TPS가 목표로 설정이 되었는지 확인함