

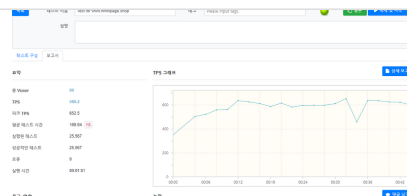
Ngrinder 로 부하테스트 해보기

≡ 작성자	현호
👤 FE/BE	BE
🕒 생성 일시	@2023년 3월 30일 오후 1:21

[Spring] 백엔드 부하 테스트 해보기 - nGrinder(with aws, jmeter)

부하테스트 툴을 알아보면 nGrinder 와 jmeter 이 2가지가 주를 이루어 나온다. jMeter JMeter 는 아파치에서 만든 java 기반의 오픈소스로 오래된 테스트 툴중 하나입니다. 꽤 오래되었고, 그만큼 안정적이라고 합니다. 많은 기능이 존재하고 확장성이 용이하지만 GUI가 이뻐보이지 않았습니

다. 장점
<https://thalals.tistory.com/288>



로커스트와의 차이

1. 언어: Locust는 Python으로 작성되었으며, nGrinder는 Java로 작성되었습니다.
2. 사용자 경험: Locust는 사용하기 쉽고 직관적인 UI를 제공하지 않습니다. 대신 사용자는 Python 스크립트를 작성하여 시나리오를 정의하고 실행해야 합니다. 반면 nGrinder는 GUI를 제공하여 사용자가 쉽게 성능 테스트를 설정하고 실행할 수 있습니다.
3. 분산 시스템: Locust는 분산 시스템을 지원하지만, 사용자는 스크립트를 직접 수정하여 이를 설정해야 합니다. 반면 nGrinder는 분산 시스템을 설정하기 위한 쉬운 UI를 제공합니다.
4. 대상 시스템: Locust는 웹 서버에 대한 성능 테스트에 초점을 맞추고 있습니다. 반면 nGrinder는 웹 서버 뿐만 아니라 데이터베이스 및 애플리케이션 서버 등 다양한 유형의 시스템에 대한 성능 테스트를 지원합니다.
5. 지원하는 프로토콜: Locust는 HTTP 프로토콜만 지원합니다. 반면 nGrinder는 HTTP 뿐만 아니라 JDBC, JMS, TCP 등 다양한 프로토콜을 지원합니다.

Performance Test

Basic Configuration

Agent	<input type="text" value="1"/> Max: 1	Vuser: 750	
Vuser per agent	<input type="text" value="750"/> Max: 3000	Processes <input type="text" value="10"/>	Threads <input type="text" value="75"/>
Script	<input type="text" value="svn"/> <input type="text" value="GetTestName.groovy"/>	<input type="button" value="R 7"/>	
Script Resources	<div></div>		
Target Host	<div></div> <input type="button" value="+Add"/>		

<input checked="" type="radio"/> Duration	<input type="text" value="0"/> : <input type="text" value="01"/> : <input type="text" value="00"/> HH:MM:SS
<input type="radio"/> Run Count	<input type="text" value="0"/> Max: 10000

테스트 설정

- Vuser per agent (가상사용자 수) : process * thread값
 - 사용자 수를 정하는 기준은 , 백엔드 요청시에 에러가 발생하기 시작하는 수로 결정
- Duration (테스트 실행 시간) : 가상 사용자의 수를 1분으로 설정 (동시에 요청하는 가상 사용자가 많으므로 길게 설정할 필요 x)

테스트에 사용한 스크립트

api 선정 기준 : 사용자가 주로 느끼는 것은 get요청으로 조회하는 시간이다.

이를 기준으로 용량이 많이 들어간 api를 선정하였다.

-

```

import static net.grinder.script.Grinder.grinder
import static org.junit.Assert.*
import static org.hamcrest.Matchers.*
import net.grinder.script.GTest
import net.grinder.script.Grinder
import net.grinder.scriptengine.groovy.junit.GrinderRunner
import net.grinder.scriptengine.groovy.junit.annotation.BeforeProcess
import net.grinder.scriptengine.groovy.junit.annotation.BeforeThread
// import static net.grinder.util.GrinderUtils.* // You can use this if you're using nGrinder after 3.2.3
import org.junit.Before
import org.junit.BeforeClass
import org.junit.Test
import org.junit.runner.RunWith

import org.ngrinder.http.HTTPRequest
import org.ngrinder.http.HTTPRequestControl
import org.ngrinder.http.HTTPResponse
import org.ngrinder.http.cookie.Cookie
import org.ngrinder.http.cookie.CookieManager

/**
 * A simple example using the HTTP plugin that shows the retrieval of a single page via HTTP.
 *
 * This script is automatically generated by ngrinder.
 *
 * @author admin
 */
@RunWith(GrinderRunner)
class TestRunner {

    public static GTest test
    public static HTTPRequest request
    public static Map<String, String> headers = ["email":"ac@naver.com"]
    public static Map<String, Object> params = [:]
    public static List<Cookie> cookies = []

    @BeforeProcess
    public static void beforeProcess() {
        HTTPRequestControl.setConnectionTimeout(300000)
        test = new GTest(1, "j8b105.p.ssafy.io")
        request = new HTTPRequest()

        // Set header data
        headers.put("email", "ac@naver.com")
        grinder.logger.info("before process.")
    }

    @BeforeThread
    public void beforeThread() {
        test.record(this, "test")
        grinder.statistics.delayReports = true
        grinder.logger.info("before thread.")
    }

    @Before
    public void before() {
        request.setHeaders(headers)
        CookieManager.addCookies(cookies)
        grinder.logger.info("before. init headers and cookies")
    }

    @Test
    public void test() {
        HTTPResponse response = request.GET("https://j8b105.p.ssafy.io/api/user", params)

        if (response.statusCode == 301 || response.statusCode == 302) {
            grinder.logger.warn("Warning. The response may not be correct. The response code was {}.", response.statusCode)
        } else {
            assertThat(response.statusCode, is(200))
        }
    }
}

```

