

성능테스트 계획서

개요

- 목적
 - 대용량 트래픽 상황에 예외 상황을 통해 미리 부하를 줌으로써 테스트해보고 방지하며 안정된 서비스를 운영하기 위해서
 - community-server API 성능테스트
- 테스트 전략 & 일정(plan)
 - 성능테스트 도구 : Jmeter
- 기준값/목표
 - 게시글 10만개 일때 검색 API가 TPS 1000이상
- 결과 확인 사항
 - Jmeter 모니터링 지표를 바탕으로 현재 문제 체크
 - 처리량(Throughput, Transaction Per Second를 의미) : 단위 시간 당 대상 시스템에 의해서 처리되는 요청 건수(bytes/sec)
 - 평균 응답시간(Average) : 클라이언트가 서비스를 요청하고 응답을 받기까지 걸리는평균적인 소요시간(초)
 - 최대 응답시간(max) : 클라이언트가 서비스를 요청하고 응답을 받기까지 걸리는최대 소요시간(초)
 - Fail Rate(Error) : 전체 요청 건수 대비 오류 발생 건수(%)
 - 메모리 사용량 : 부하 상황에서 서버에서 IT 서비스를 제공할 때의 메모리사용량(%)
 - CPU 사용량 : 부하 상황에서 서버에서 IT 서비스를 제공할 때의 CPU사용량(%)

Summary Report

Name:

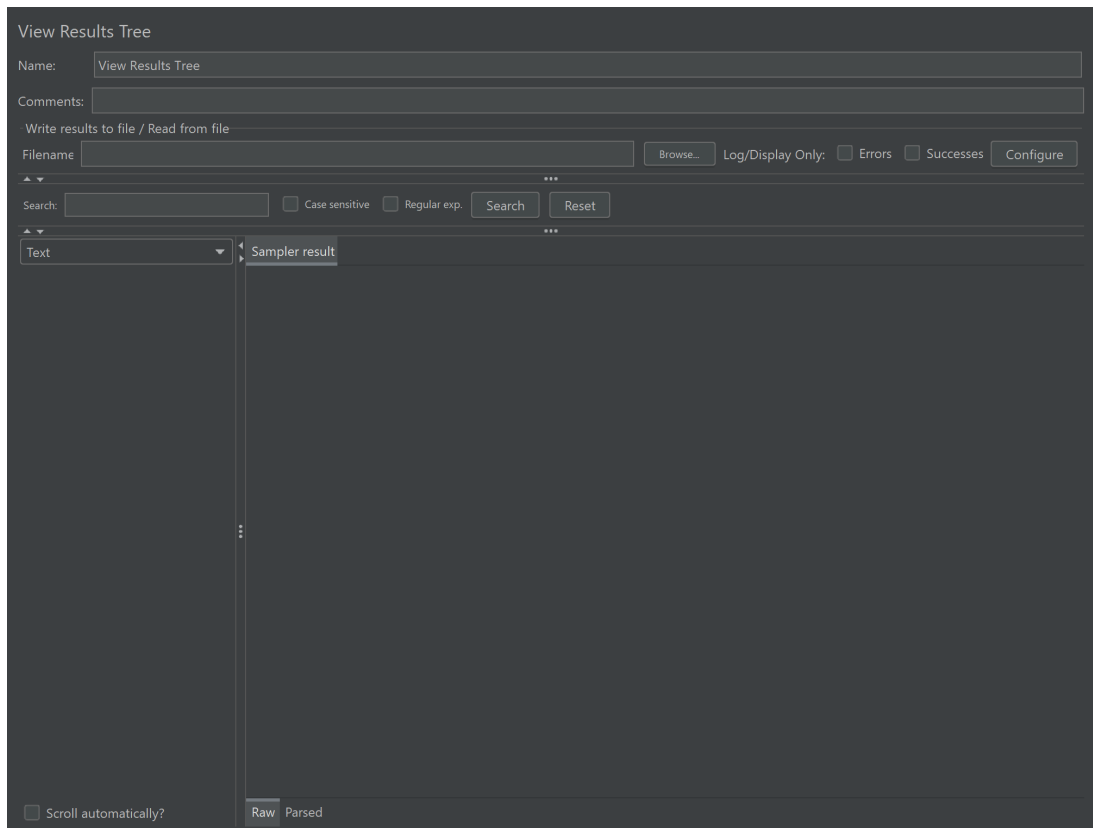
Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
TOTAL	0	0	#N/A	#N/A	0.00	0.00%	.0/hour	0.00	0.00	.0
<div> <input type="checkbox"/> Include group name in label: <input type="button" value="Save Table Data"/> <input checked="" type="checkbox"/> Save Table Header </div>										

- 리팩토링 부분 확인
 - 쿼리 튜닝(mysql의 explain로 확인)
 - 인덱스 추가
- 예외상황 체크
 - 예외 상황을 체크 하기 위해 jmeter에서 제공하는 결과 보고서의 리스너를 추가



테스트 데이터 정보

- 테스트 DB에 10만개의 게시물 추가

```
{
  "categoryNumber" : "${__RandomString(1,12, )}",
  "postName" : "${__RandomString(1,AaBb, )}",
  "content" : null,
  "createTime" : null
}
```

- 두개의 검색 조건을 두어 랜덤 검색

테스트 시나리오

- STRESS 테스트(일정 주기적(패턴)으로 부하를 줘서 프로그램 정상 가동 여부 테스트)
 - 동시 사용자 수 조정, 요청 패턴(서비스 복잡도에 따른 HTTP HEADER, BODY 용량)
 - 시나리오
 - BODY의 용량(검색 조건) 100명의 동시 사용자가 초당 100번을 호출하여 분당(10분) 사용자를 100씩 늘려 서버의 지표를 확인

- BODY의 용량(검색 조건) 50명의 동시 사용자가 초당 50번을 호출하여 분당 (10분) 사용자를 50씩 늘려 서버의 지표를 확인
 - BODY의 용량(검색 조건) 500명의 동시 사용자가 초당 50번을 호출하여 분당 (10분) 사용자를 50씩 줄여 서버의 지표를 확인
- 로컬에서의 테스트이기 때문에 한계가 있습니다.
- INDURANCEE 테스트
 - 장시간 동안 시스템이 얼마나 안정적으로 동작할 수 있는지를 확인
 - 시나리오
 - 100명의 동시 사용자가 초당 100번을 호출하였을때 10분동안 서버의 지표 CPU, RAM, DISK, TPS 가 이상이 있는지 테스트