Sangmin, Lee www.tuning-java.com

PERFORMANCE & JAVA TUNING

About me

- 자바 성능을 결정짓는 코딩 습관과 튜닝 이야기 집필
- Java Language Specification Third edition 공동 번역

- 하는 일
  - Performance test
  - Java application tuning

Agenda

- Performance (40 min)
- Java Tuning (1 H)
- Java Tuning Report example review (20 min)

Performance ???

- Time ...
- TPS ...(Transaction Per Second)

Time

- 웹에서의 시간은 ???
  - 응답시간과 대기시간
    = Response time and Think time

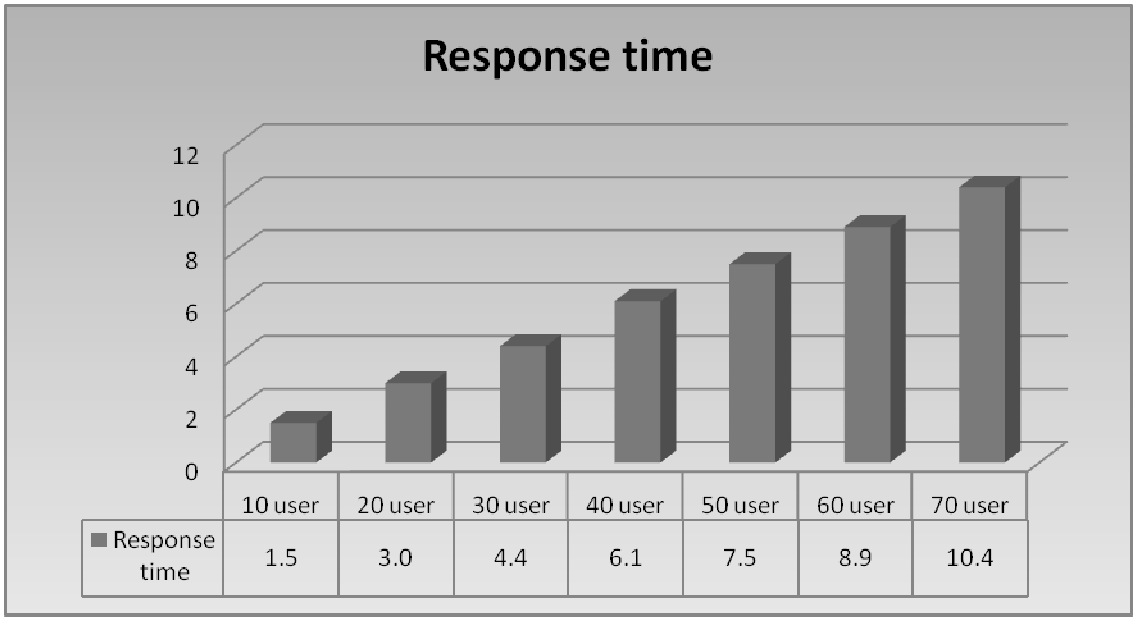
  - Think time
    - 두 Request time 사이의 시간

Request & Response	Think time	Request & Response

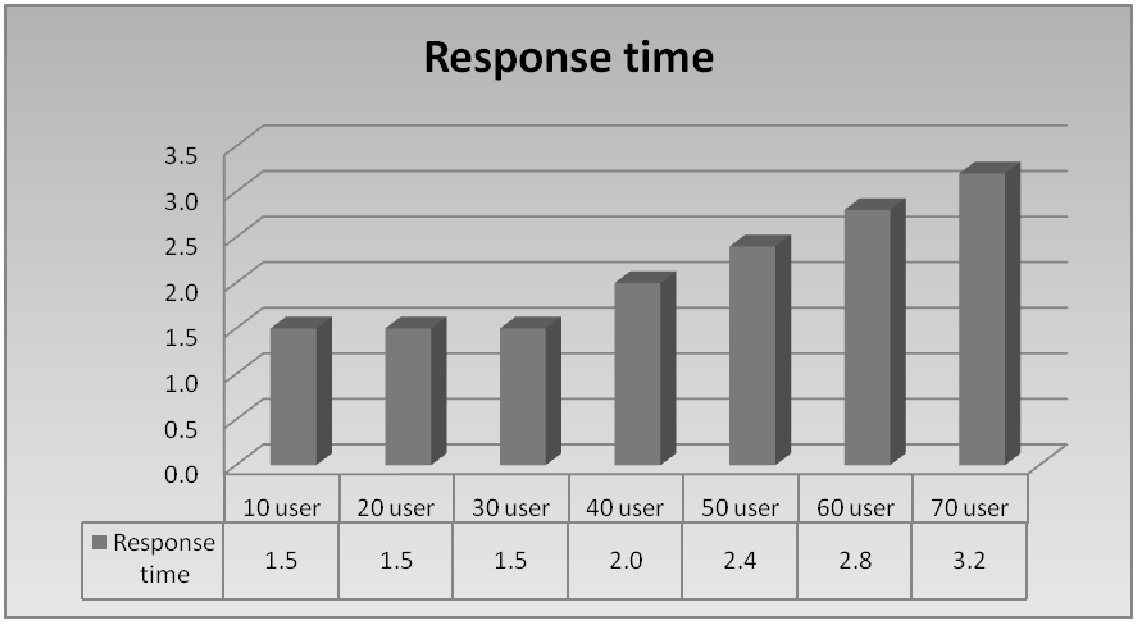
Response Time

- Response time is divided by…
  - Network connection
  - Send request data
  - Wait time
  - Receive response data
  - Network close

Response time graph-1



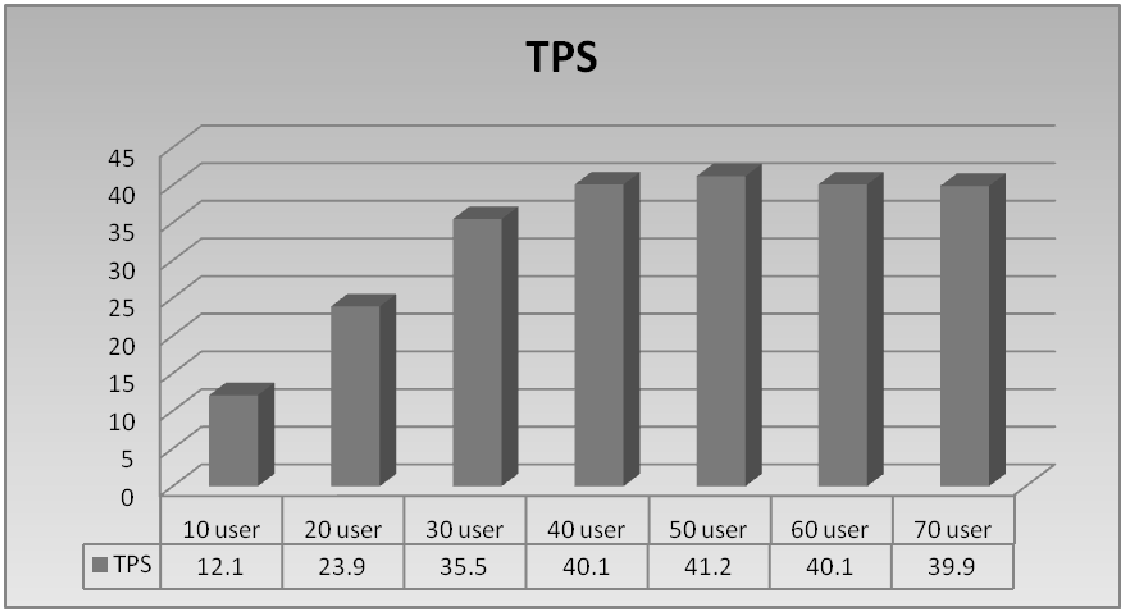
Response time graph-2



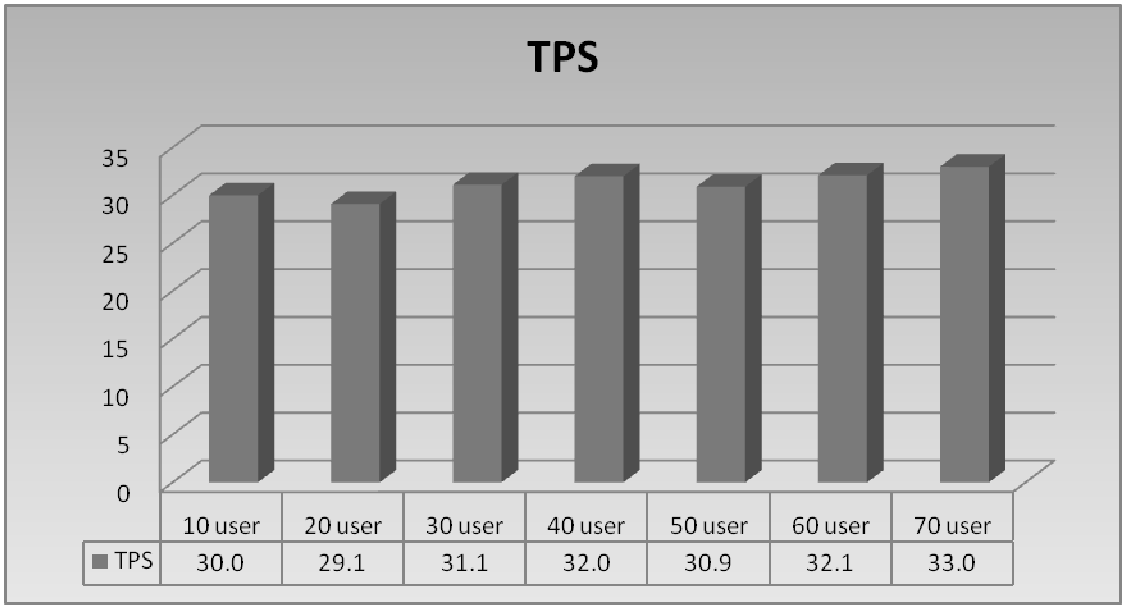
TPS

- Transaction Per Second means...
- 1 TPS = 60 TPM = 3,600 TPH
- Our system's TPS is 20 TPS means ~~~
  Our system can serve 72,000 transactions per hour.

TPS graph - 1


TPS
45
40
35
30
25
20
15
10
5
0
10 user
20 user
30 user
40 user
50 user
60 user
70 user
TPS
12.1
23.9
35.5
40.1
41.2
40.1
39.9

TPS graph - 2


TPS
35
30
25
20
15
10
5
0
10 user
20 user
30 user
40 user
50 user
60 user
70 user
TPS
30.0
29.1
31.1
32.0
30.9
32.1
33.0

Response time vs TPS

- Which is better to show system capability ?

Performance test tools

- There are a lot of performance tools in the world
  - Load Runner
  - Performance suite enterprise
  - SilkPerformer
  - Web LOAD
  - JMeter
  - MS Web application stress tool
  - …

Process of performance test

Agenda

- Performance (40 min)
- Java Tuning (1 H)
- Java Tuning Report example review (20 min)

20 80's law

- 20 80의 법칙
  - 상위 20%의 application이 80%의 사용량과 리소스를 점유함.

- 5 95의 법칙
  - 상위 5%의 application이 95%의 사용량과 리소스를 점유함.
    ➔ 대부분의 사이트를 분석해 본 결과 이 법칙이 더 정확함.

Web 에서 성능에 영향을 주는 요소는 ?

1. Web server
2. WAS
3. DB Server
4. File Server
5. Legacy Server
6. Network
7. Nothing
8. I don’t know

Why ???

- Because ...

Response Time (think again)

- Response time is divided by…
  - Network connection
  - Send request data
  - Wait time
  - Receive response data
  - Network close

N/W connect	Send request	Server time	Receive response	N/W close

Server time

- Server time is divided by …
  (in a point of WAS view)
  - CPU 시간과 대기 시간

    = WAS 에서 잡아먹는 시간 +딴데서 잡아먹는 시간

    = Thread time + Wait time

Thread time

- Thread time consumes WAS's CPU

- Can we reduce these time ?

Wait time

- Wait time is divided by…
  - Network time
  - DB time
  - IO time
  - Other system’s response time

- Can we reduce these time ?

What should I do ?

- Everything you have to do is…

Find Bottleneck !!!!!

How to find bottleneck ? -1

- Well… the best way is…

Use tools !!!

- But after find point, you must tune application.

How to find bottleneck ? -2

- If you don't have tool…

Use
System.currentTimeMillis();
Or
System.nanoTime();

How to find bottleneck ? -3

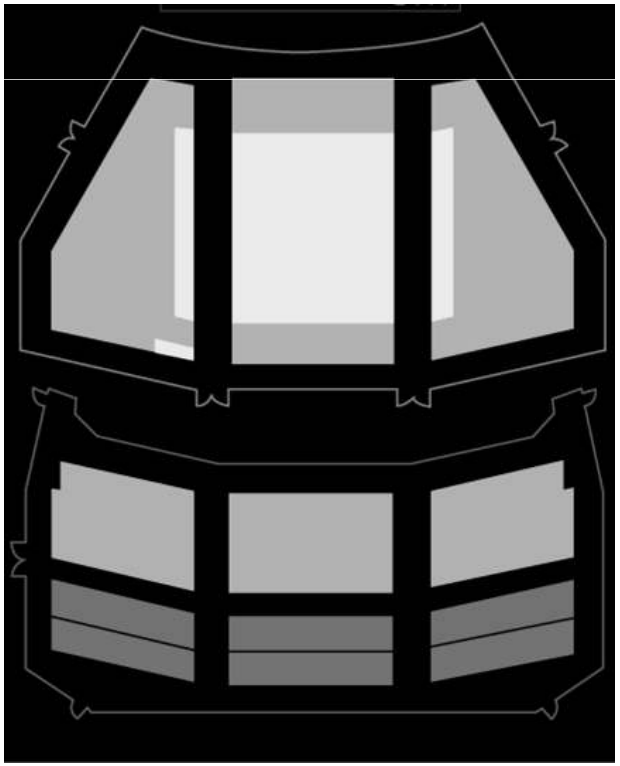
- If you don't have tool…
  - Analyze access log
    - 웹 로그에는 기본적으로 응답시간이 찍히지 않음.
    - %D (마이크로 초) or %T(초) 를 access log 포맷에 추가

How to approach ?

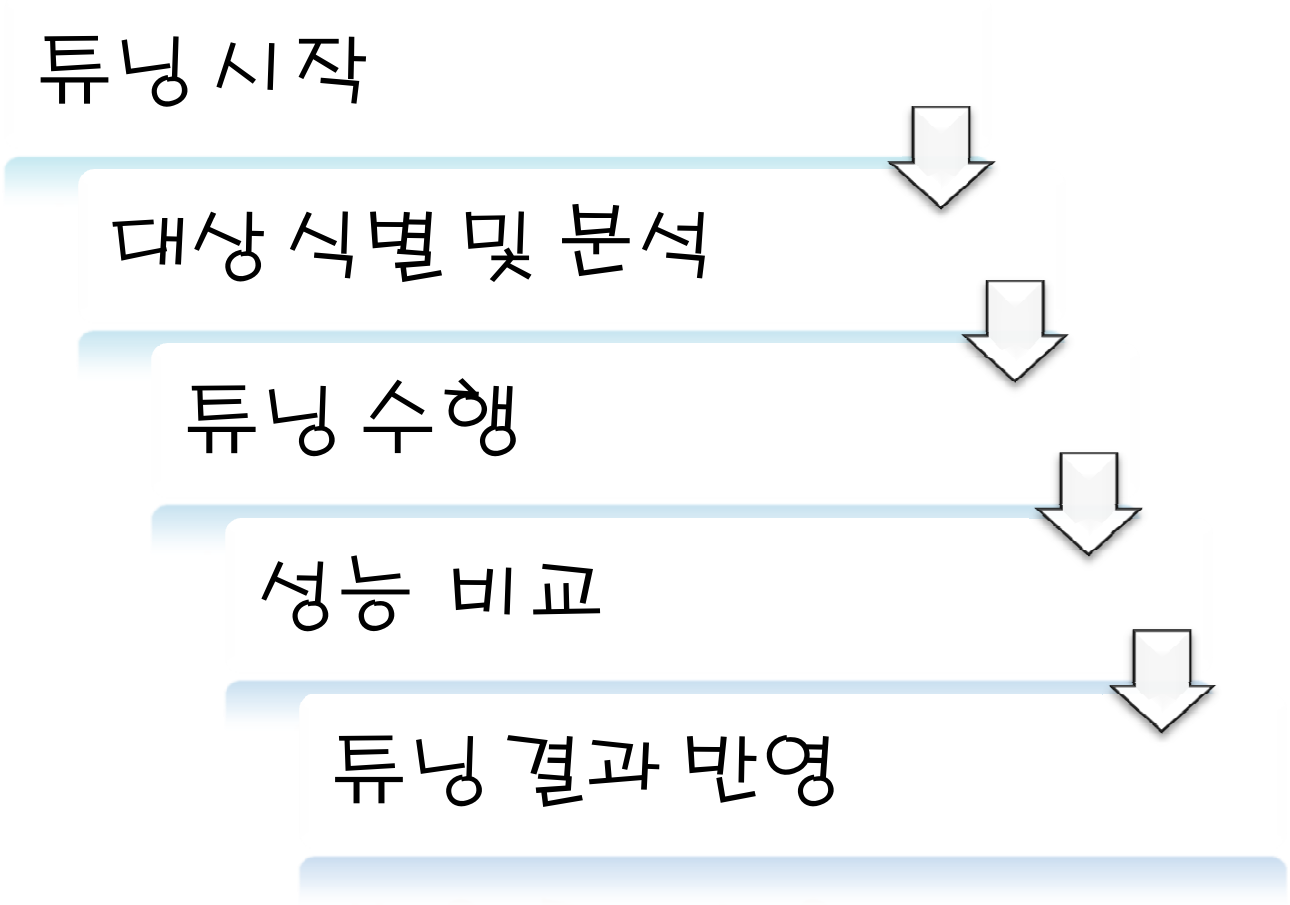
- The most important thing is …

Performance tuning tools

APM vs Profiling tool



Performance tuning process



Java Tuning 의 대상들

Pattern

XML

I/O

String

JDBC

GC Setting

Log

Case study

- Application is toooooo slow or doesn’t response
- Server dies every day

Application is too slow or doesn't response

```
public ReasonVO getSlowReason(Object problem) {
    if(problem instanceof Environment) {
            return checkEnvironment(problem);
    } else if(problem instanceof  WASSetting) {
            return checkWASSetting(problem);
    } else if (problem instanceof  Program) {
            return new ReasonVO("Too much reason", …);
    }
}
```

Application is too slow or doesn't response

```
public ReasonVO checkEnvironment(Object problem) {
    if(problem instanceof DB) {
            return checkDB(problem);
    } else if (problem instanceof Network) {
            return checkNetwork(problem);
    }  else if(problem instanceof Storage) {
            return checkStorage(problem);
    } else {
            return checkExtraEnvironment(problem);
    }
}
```

Application is too slow or doesn't response

```
public ReasonVO checkWASSetting(Object problem) {
    if(problem instanceof  ThreadNumber) {
            return checkThreadNumber(problem);
    } else if (problem instanceof  DBConnectionPool) {
            return checkDBConnectionPool(problem);
    }  else if(problem instanceof WebServer) {
            return checkWebServer(problem);
    } else {
            return checkExtraSetting(problem);
    }
}
```

Application is too slow or doesn't response

- How to prevent.
  - Monitor with monitoring tool. (Best)
  - Monitor with WAS Console
    - Check thread usage
    - Check memory
    - Check DB Connection pools
  - Monitor with JMX
    - Build your own JMX Codes.

Server dies everyday.

```
public ReasonVO getDieReason(Object problem) {
    if(problem instanceof MemoryProblem) {
            return analysisMemory(problem);
    }  else if(problem instanceof TooMuchUser) {
            return checkServerSettingOrExpandServer(problem);
    } else {
            return new ReasonVO("Too much reason",…);
    }
}
```

Agenda

- Performance (40 min)
- Java Tuning (1 H)
- Java Tuning Report example review (20 min)
  file link

Q n A

Thank you