Profiling Scout Applications

Useful tools and methods we use for profiling our Scout applications

Judith Gull
Scout User Group Meeting – 27.10.2014
Why is it so slow?

- Business Logic on Client or Server
- External Webservices
- Database Queries
- Network Connection
Will the system still be usable with many concurrent users?

- Can we measure and tune before productive use
- How do we need to setup application servers/database
3 common ways we profile Scout applications at BSI

- Very simple profiling with TuningUtility
- Custom Scout Profiler on Service-Level
- Loadtests with Apache JMeter
Scout TuningUtility
TuningUtility
Simple Timing for Development

TuningUtility.startTimer();
try {
    SERVICES.getService(IPersonProcessService.class).load(formData);
} finally {
    TuningUtility.stopTimer("Load person formData");
}

#TUNING: load person formData took 125.671268ms
```java
try {
    for (int i = 0; i < 100; i++) {
        TuningUtility.startTimer();
        codeToMeasure();
        TuningUtility.stopTimer("repeatCode", false, true);
    }
} finally {
    TuningUtility.finishAll();
}

#TUNING: repeatCode[100] sum=1114.534945ms min=7.799033ms avg=11.145349ms median=10.510715ms max=29.425593ms
[without 1 smallest and 1 largest:…
```

TuningUtility – Repeated Calls

- **Try**: Begins the execution of the method.
- **For Loop**: Iterates through 100 times, starting from 0 to 99.
  - **Start Timer**: Begins timing the inner block.
  - **Code Measurement**: Measures the performance of the code block.
  - **Stop Timer**: Stops the timer and records the results.
- **Finally**: Ensures all resources are cleaned up after the try block finishes.
- **#TUNING**: Outputs statistics about the repeated calls.

**Statistics**:
- **Sum**: Total execution time of 1114.534945 milliseconds.
- **Minimum**: Execution time of 7.799033 milliseconds.
- **Average**: Average execution time of 11.145349 milliseconds.
- **Median**: Median execution time of 10.510715 milliseconds.
- **Maximum**: Execution time of 29.425593 milliseconds.
- **Without 1 smallest and 1 largest**: Additional statistics excluding the smallest and largest values.
Summary – TuningUtility

→ Very easy to use

→ Do not use in production!
Custom Scout Profiler
Why a custom profiler?

- General purpose measure every method -> complicated

- Must be connected to JVM (often remotely with JMX)

- Difficult to profile both client and server
Profiling on Service Level

Displays server and client durations in single table

Only shows service methods (relevant for business)

Measures durations in client and server
Works with different server/client time

-> much simpler than general purpose java profilers!
Diagram Export
Profiling in Production

- stores profiling data as files on the server
- possible to enable per user session or global
- possible to run in production
Implementation

- On profiling start: Register profiling proxy service for every service
- Proxy Captures time and delegates to “real” service
Additional Profiling Tasks

⇒ Add any additional profiling tasks where needed

Profiler jobProfiler =
Profiler.beginTask(getJobName(), "JOB");
try{
    execRun(jobRunDesc);
}finally{
    jobProfiler.endTask();
}
Summary - Custom Scout Profiler

- Easy to use with deployed applications
- Mostly sufficient to get an idea where the performance problems are
- Learnings: Measurements can be quite different (use multiple)
Load Testing with Apache JMeter
Load Testing Scout Applications

Prepare Tests for JMeter

Optimize

Run Load Tests

Analyze Results

Write Testcases, Plan Setup Infrastructure, multiple test clients

Multiple phases, Include manual Testers

JMeter Graphs and Tables
Prepare test client for JMeter

- Small Scout Extension for JMeterTests
  (creating session, formatting output)

- Prepare Testcases: Implement most common use cases
Configure Tests in JMeter

- Create Plans to simulate real users
- Configure random executions
- CSV Test Properties for multiple executions
Distribute Load Test Client

- Export special product file with Scout JMeter extension, project specific test cases, configuration
- Headless application (GUI rendering is not measured)
Simulation with increasing load over time
Summary – Load Tests

➤ Load tests are useful
  ➤ there is usually room for improvement
  ➤ problems are not always obvious
➤ Load tests are not free, require careful planning, significant amount of time to prepare
➤ Difficult to map “simulated users” to actual users
Questions