

perf4j: Performance Logging and Performance Statistics for Java

Michael.Kuperberg@kit.edu

April 28th, 2010



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft



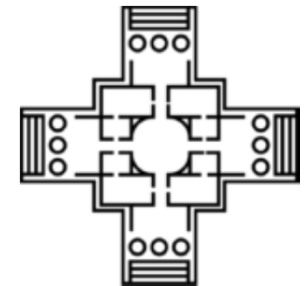
Universität Karlsruhe (TH)
Research University • founded 1825

Java Users Group Karlsruhe, Lightning talks



JUG Karlsruhe

jug-ka.de | twitter.com/@jugka



perf4j: Starting Up

- <http://perf4j.codehaus.org/> [P4J] own description:
"Perf4J is to System.currentTimeMillis() as log4j is to System.out.println()"
 - Includes a tool for parsing logs, generating aggregated statistics and performance graphs
 - Integration with logging frameworks, e.g. log4j, java.util.Logging, SLF4J, Apache Commons Logging
 - Custom logging appenders, etc.
- Support for AOP (@Profiled), servlets (graphs on the web); integration for Grails, Seam available

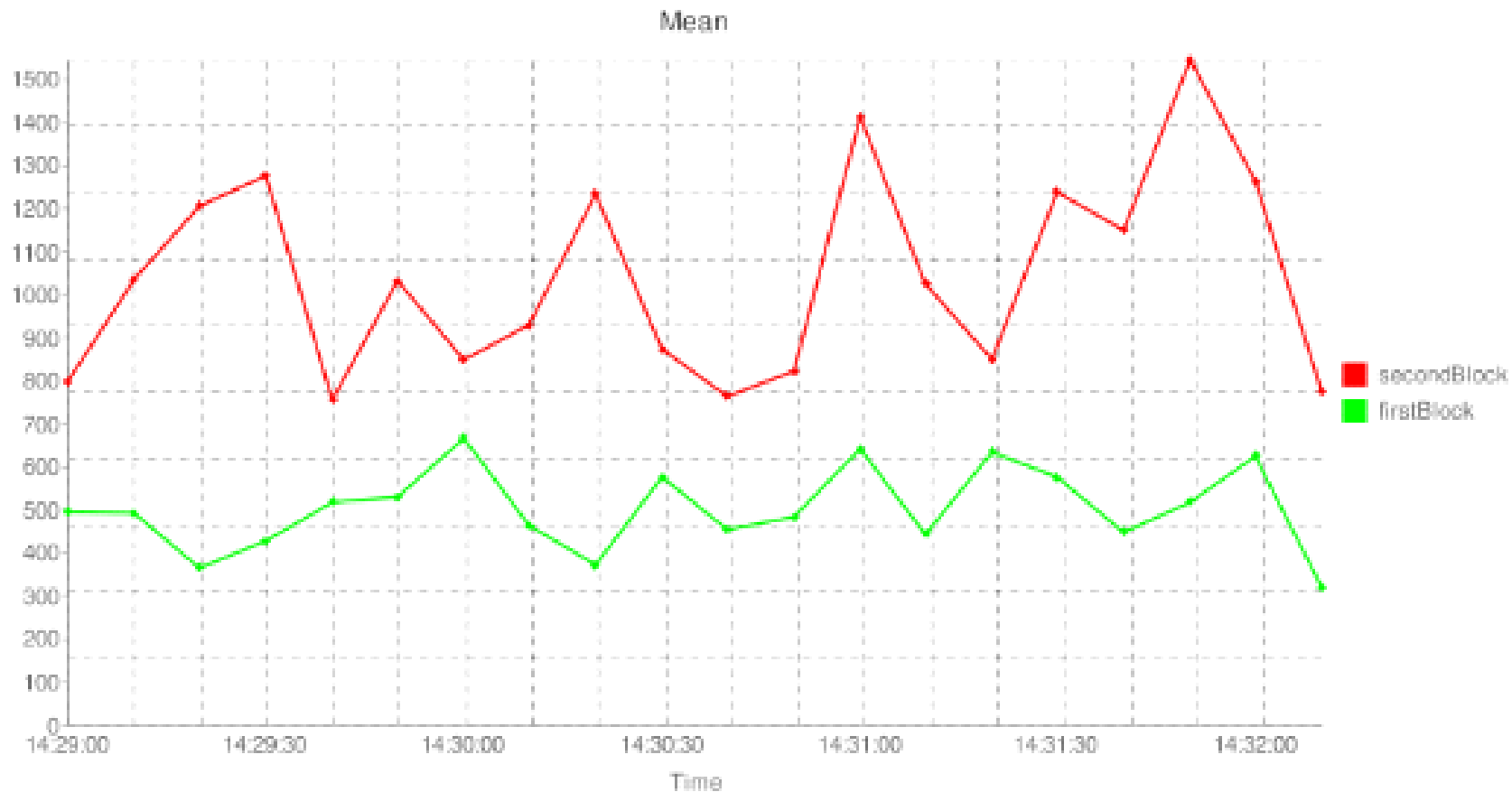
Getting and installing perf4j

- Apache2 licence; current perf4j version is 0.9.13
- Maven repository (but also works "by hand"):
<http://repository.codehaus.org/org/perf4j/perf4j/>
- Two ways to use:
 - direct use of StopWatches in source code
 - @Profiled AOP annotation in source code
- (Demo 1)

Features of perf4j

- Parser supports CSV output, graph creation
- log4j appender (for java.util.logging: announced)
- Support for EJBs
- Performance statistics available dynamically
 - e.g. for monitoring, continuously running systems
- Performance stats can be exposed using JMX
- (Demo 2)

Example Graph (taken from [P4J])



Tips and Tricks

■ The semantics of `start()` and `stop()` are tricky...

- "Note that a `StopWatch` is reusable. That is, you can call `start()` and `stop()` in succession and the `getElapsedTime()` method will refer to the time since the most recent `start()` call."
- **Stop only samples the `StopWatch` – doesn't terminate it!**

Shortcomings of perf4j

- StopWatches do not support "pause", "continue"
 - "start" is actually a "restart" (reset) – diff to "lap"?
 - Semantics of lap(): "sets [the] startTime property to the current time and resets the elapsedTime property, [...] any existing tag and message are not changed" [P4J]
- Metrics such as medians are missing
- perf4j's graphs are PNGs, data is going to Google
- Uses `currentTimeMillis()` as timer method
 - Accuracy may be insufficient
 - Timer method choice is currently fixed

Conclusions

- **perf4j helps with performance monitoring**
 - "logging for performance" with appenders etc.
 - cleaner code when using AOP
 - automated data processing and analysis

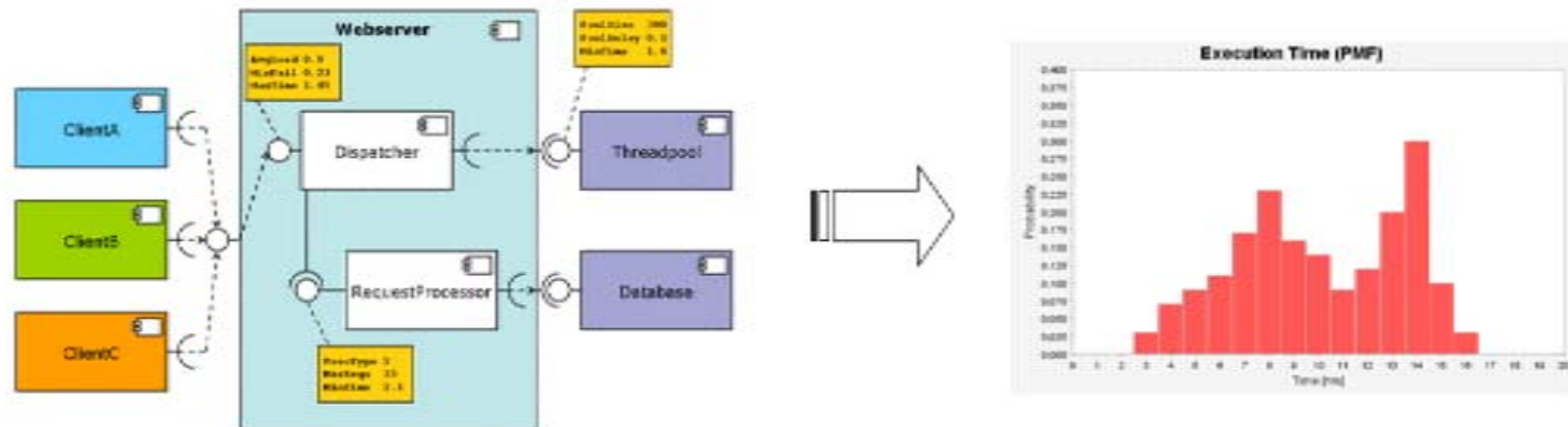
- **Alternatives exist, e.g.**
 - JAMon (Java Application Monitor) <http://jamonapi.sf.net>
 - Java Simon - Simple Monitoring API
<http://code.google.com/p/javasimon/>

- **Future work: quantify perf4j overhead 😊**

The Larger Context:

<http://www.palladio-approach.net>

■ Model-based Architecture Performance Evaluation



- Research on software architectures (KIT and FZI): benchmarking, reverse engineering, reliability, etc.
- Development using Eclipse: GEF, GMF, GEF etc.
- Student theses (master, bachelor) and jobs available