SPASS-meter - Measuring Diverse Software Attributes in an Integrated Manner

Holger Eichelberger
University of Hildesheim, Institute of Computer Science
Marienburger Platz 22, D-31141 Hildesheim, Germany
{eichelberger}@sse.uni-hildesheim.de
SPASS-meter - Measuring Diverse Software Attributes in an Integrated Manner

**SPASS-meter**

- **Motivation**
  - Assess quality requirements (in SPL)
  - Basis for self-adaptation in DSPL
- **Basic workflow**

---

Performance Engineer  \[\rightarrow\] SPASS-meter  \[\rightarrow\] Performance Engineer / (Adaptive) System

Specify Monitoring Scope  \[\rightarrow\] Instrumentation  \[\rightarrow\] Data collection  \[\rightarrow\] Data analysis / interpretation

Data aggregation

online
SPASS-meter - Measuring Diverse Software Attributes in an Integrated Manner

SPASS-meter

- Java program resource consumption
  - Execution time (CPU usage)
  - Response time (selected methods)
  - Memory consumption
  - File transfer
  - Network transfer
- Related system resource consumption
- Monitoring of user-defined program units e.g. components or services (online analysis)
- Direct vs. indirect monitoring
Further SPASS-meter Features

• Konfiguration
  – Annotations
  – External files

• Instrumentation modus
  – Dynamic
  – Static
  – Mixed

• Supports Java programs and Android Apps (Example DSPL ± OSGi)

• Optional:
  – Remote monitoring
  – JMX integration
  – OW2 Wildcat integration
Initial Overhead Evaluation

• Based on SPECjvm08
• Execution time overhead
  – Direct resources < 2,8%
  – Naive indirect monitoring < 11%
• Memory overhead < 1,4%

Experimental dynamic indirect monitoring < 3%
Summary and Future Work

• Summary
  – SPASS-meter:
    ▪ User-defined logical grouping (online analysis)
    ▪ Direct and indirect monitoring
    ▪ Rich set of features
  – Initial overhead evaluation

• Future Work
  – Application to service platforms
  – Non-Java Programs
Thank you for your interest.
Questions?

Holger Eichelberger
eichelberger@sse.uni-hildesheim.de