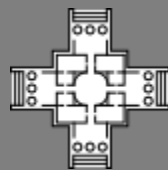28. November 2013

A Benchmark Engineering Methodology to Measure the Overhead of Application-Level Monitoring

Symposium on Software Performance

Kieker Joint Kieker/ Palladio Days

Jan Waller and Wilhelm Hasselbring
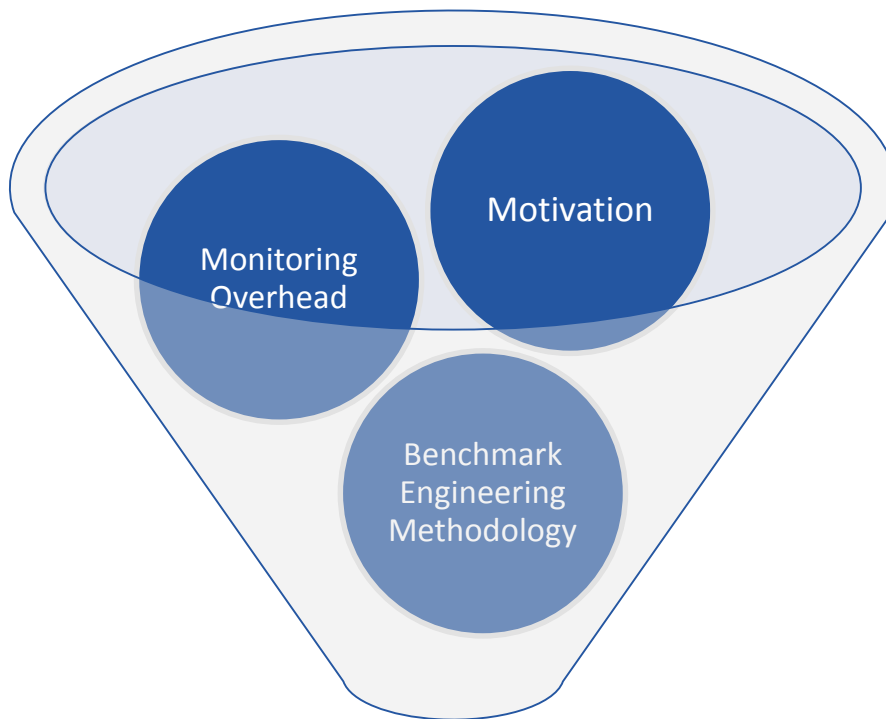Software Engineering Group, Kiel University, Germany

Motivation

Monitoring

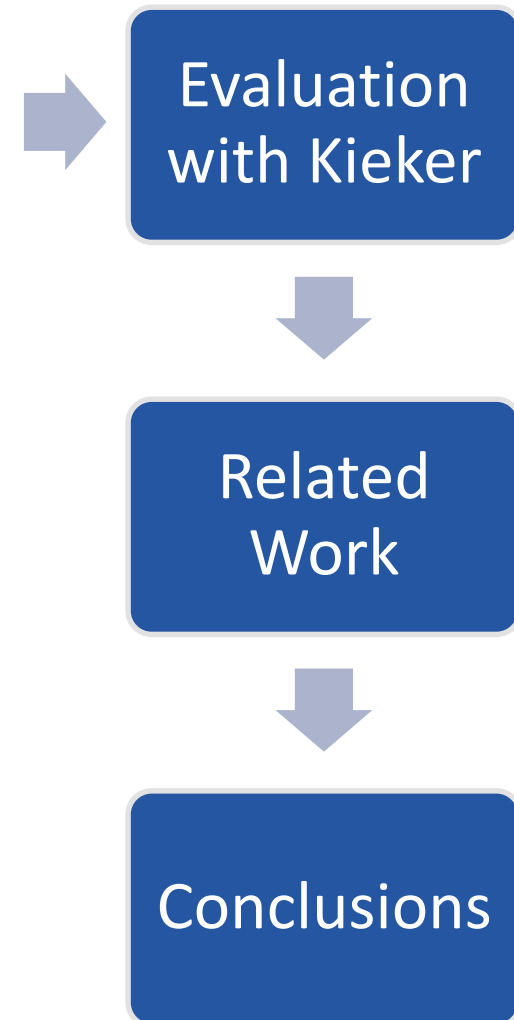- Insight
- Overhead vs. Details
- How to find exact cost?

Benchmarks

- What is a good benchmark?
- How to create a benchmark?

Outline

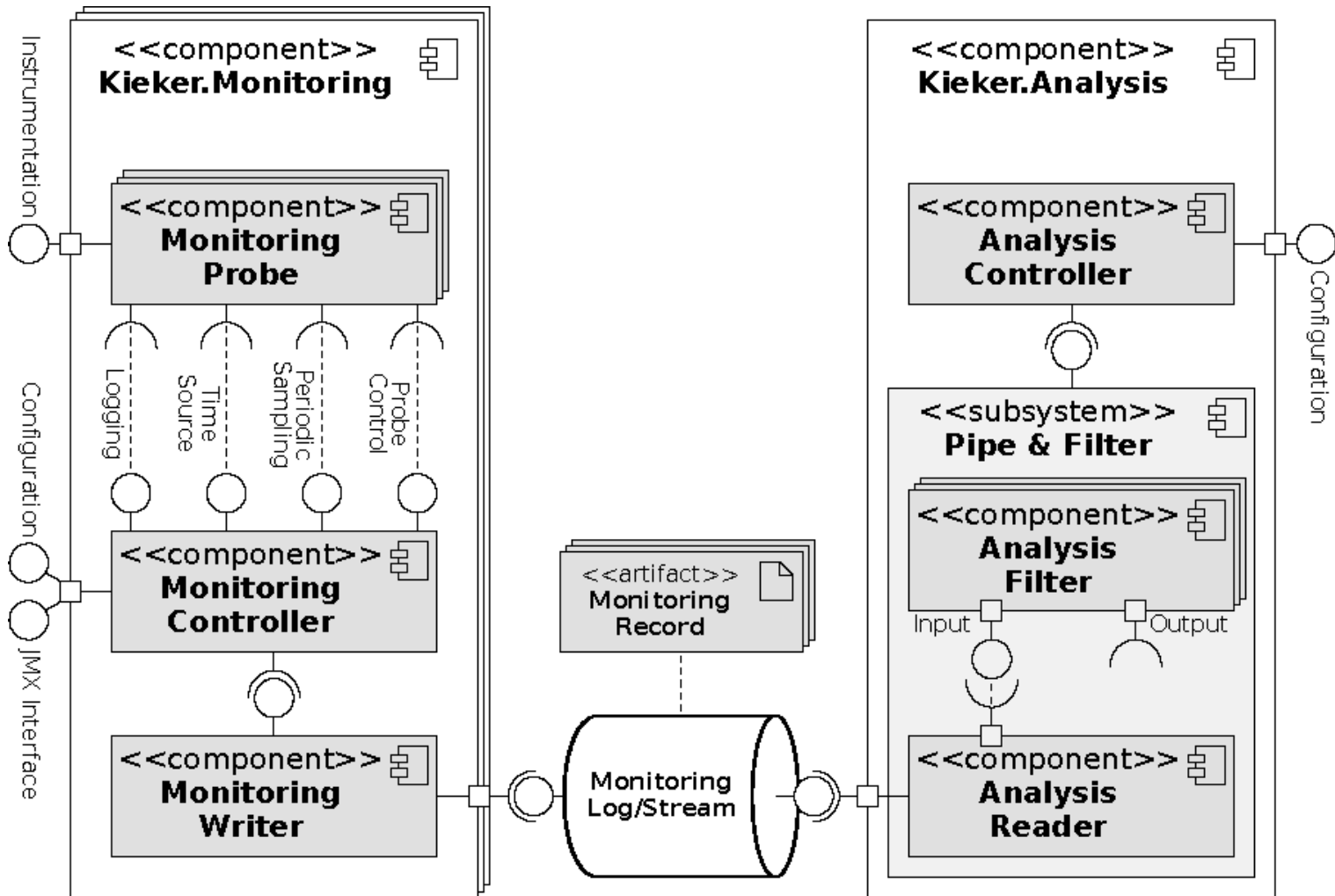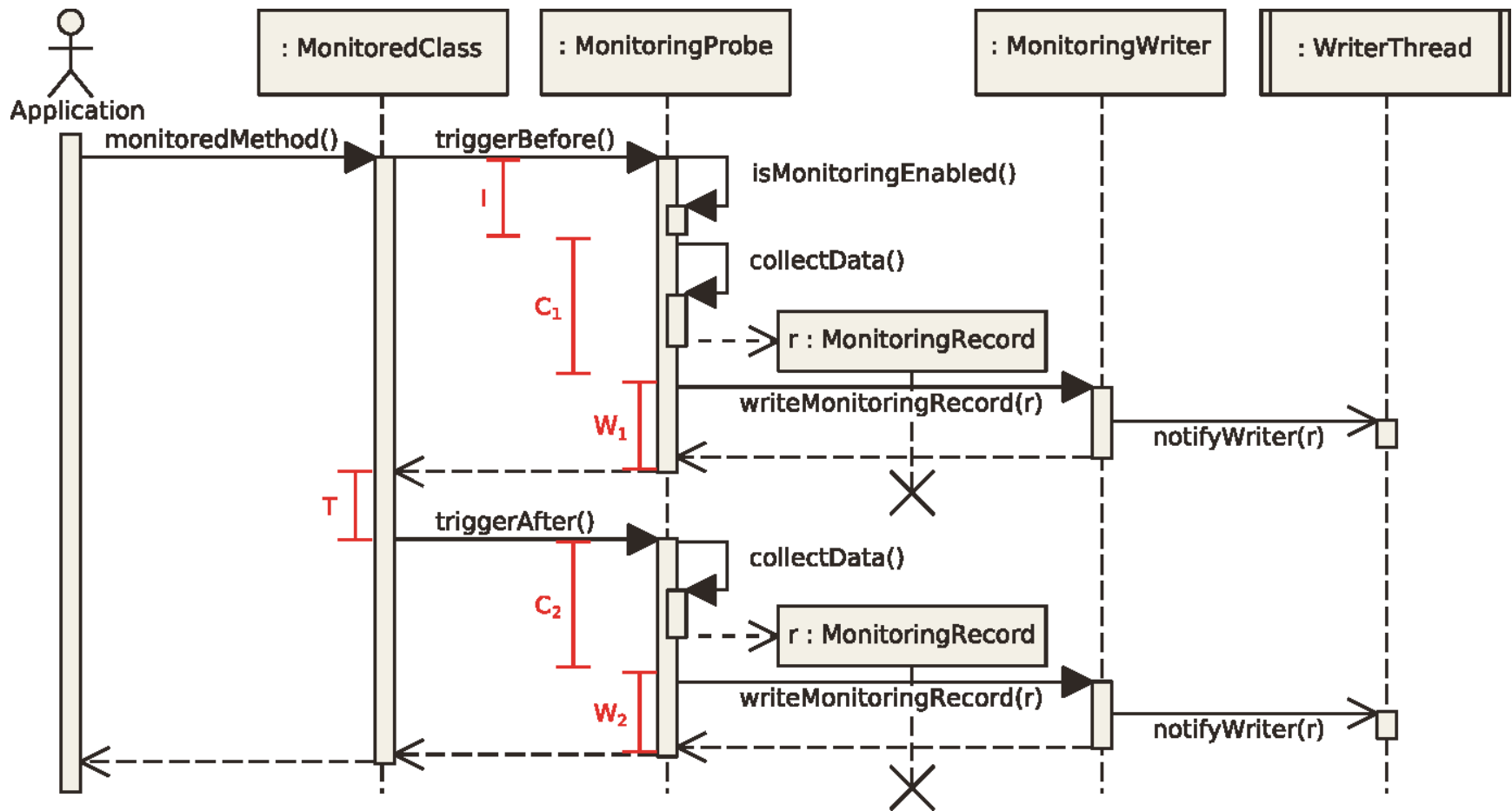Motivation

Monitoring Overhead

Benchmark Engineering Methodology

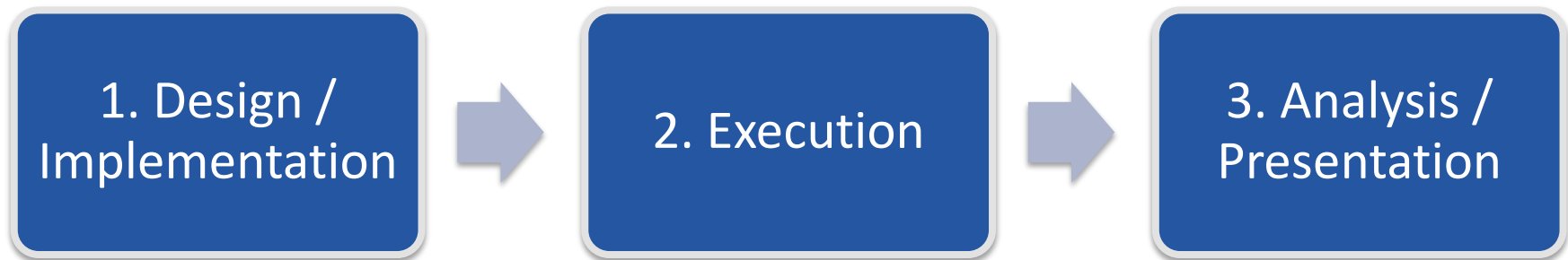MooBench (**Mo**nitoring **o**verhead **Bench**mark)

Evaluation with Kieker

Related Work

Conclusions

Kieker Monitoring Framework [vHWH12]

Monitoring Overhead [WH12, vHRH+09]

Benchmark Engineering Methodology

1. Design / Implementation	→	2. Execution	→	3. Analysis / Presentation

Benchmark Engineering [Sac11]

- Benchmark development methodology

- Should also include execution and analysis

➢ Split into three phases

➢ Provide requirements for each phase

Benchmark Engineering Methodology

| 1. Design / Implementation | → | 2. Execution | → | 3. Analysis / Presentation |

1. **Representative**
2. **Repeatable**
3. **Robust**
4. Fair

5. Simple
6. Scalable
7. Comprehensive
8. Portable

Benchmark Engineering Methodology

1. Design / Implementation	→	**2. Execution**	→	3. Analysis / Presentation
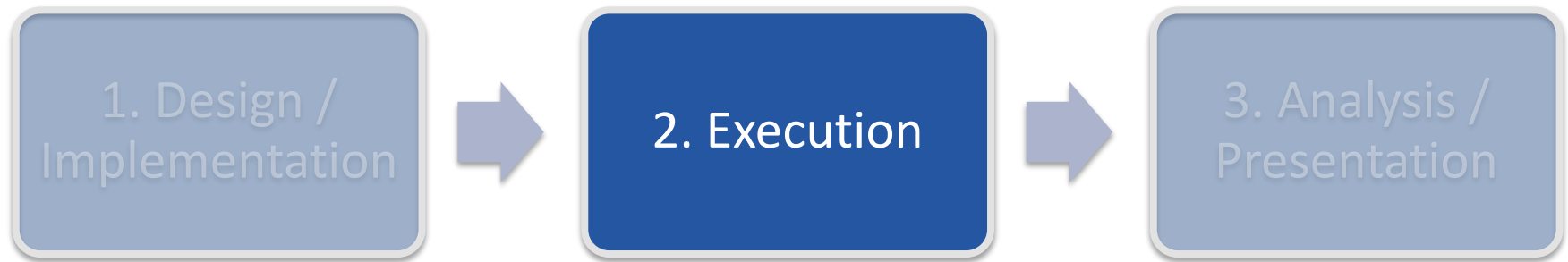
9. Robust Execution

10. Repeated Executions

11. Warm-up / Steady State

12. Idle Environment

Benchmark Engineering Methodology

| 1. Design / Implementation	→	2. Execution	→	3. Analysis / Presentation

13. Statistical Analysis

14. Reporting

15. Validation

MooBench (**Mo**nitoring **o**verhead **Bench**mark)

- Measures the three causes of overhead

- *Monitored Application*
  - very basic; single class; single method; fixed timing

- *Benchmark Driver*
  - initializes; executes; collects; records

- *Designed/implemented*, *executed*, and *analyzed/presented* according to our benchmark engineering methodology
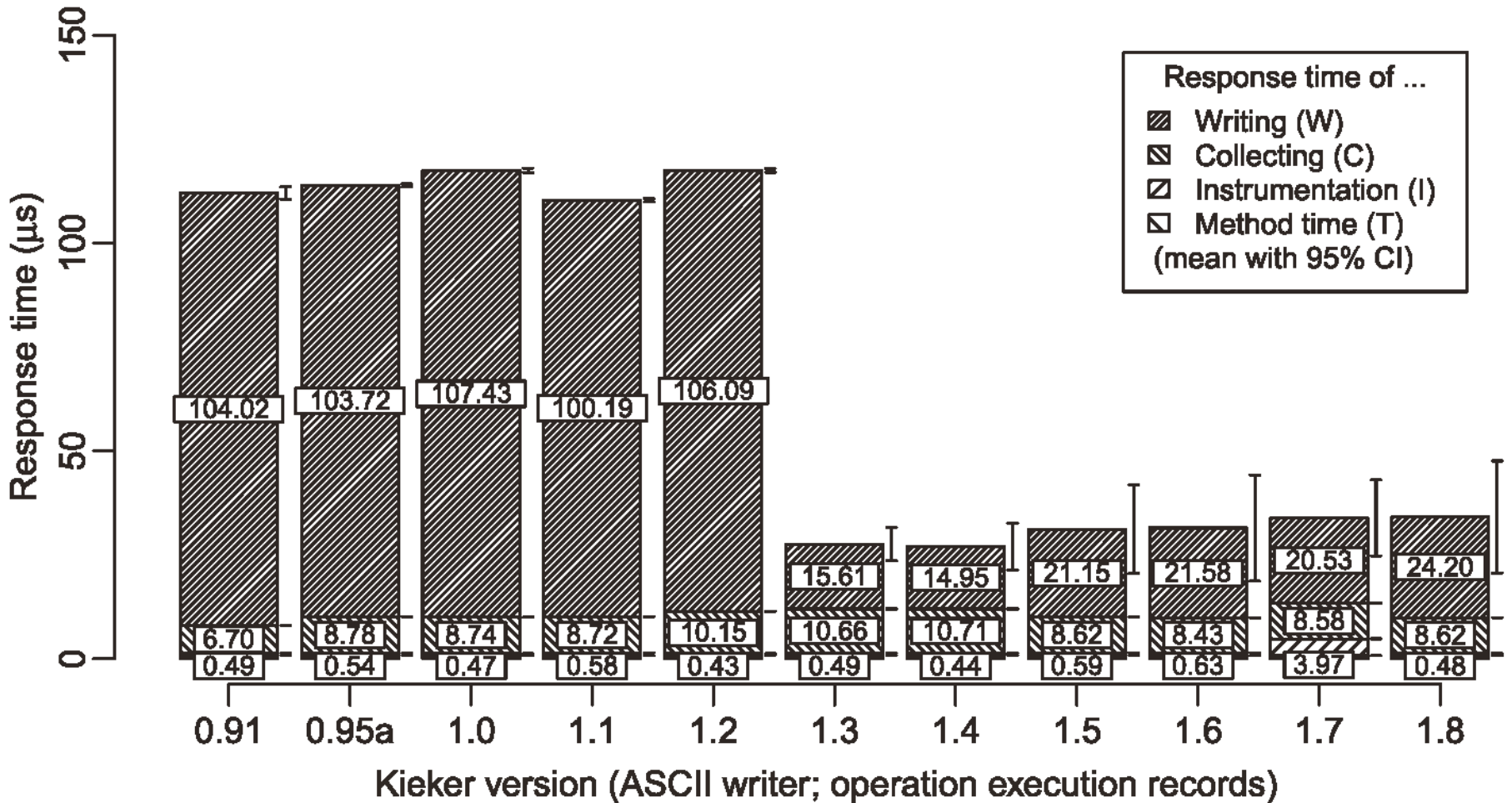
Example: Warm-up vs. Steady State
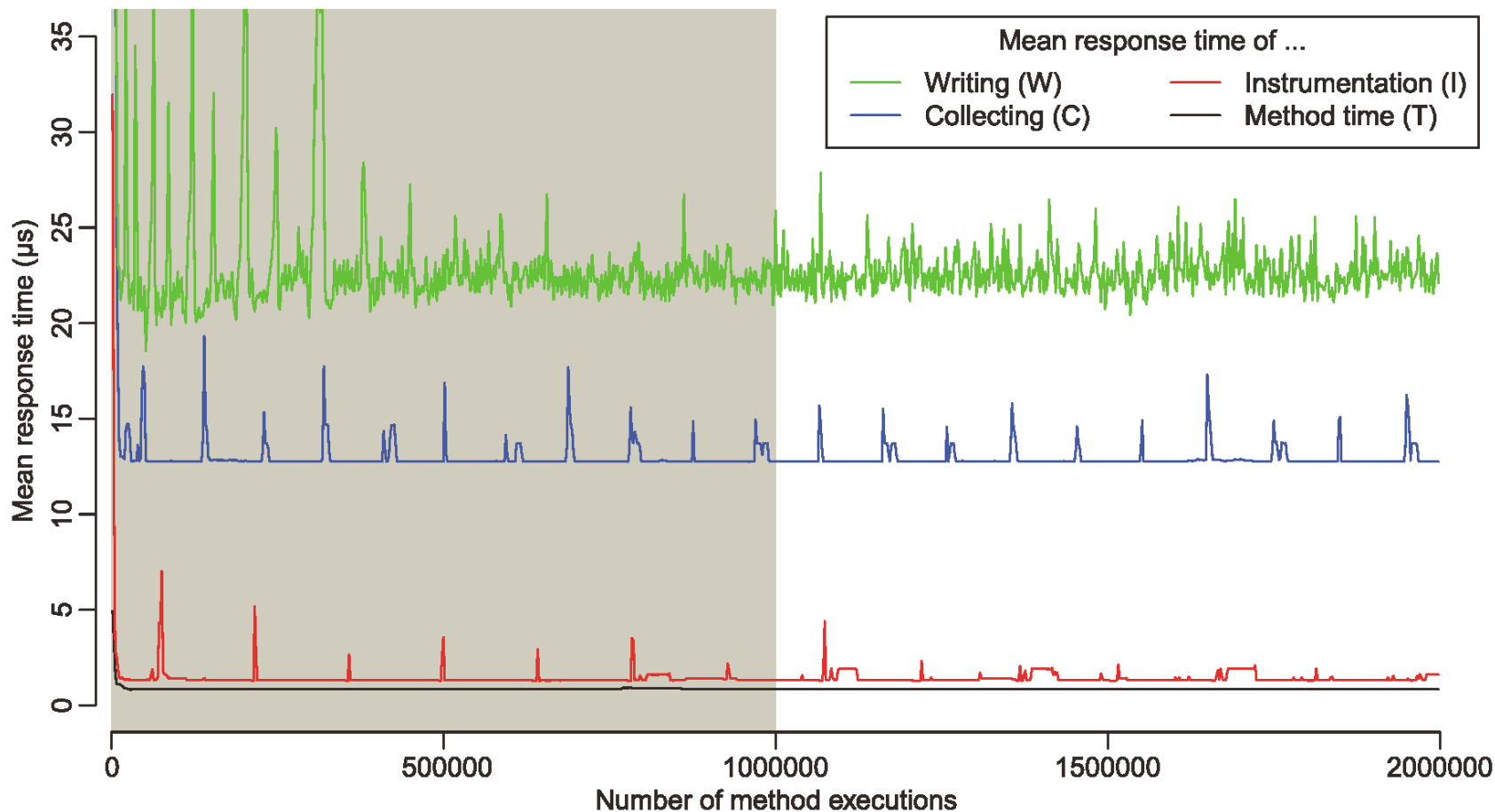
Kieker: Small Moments in History
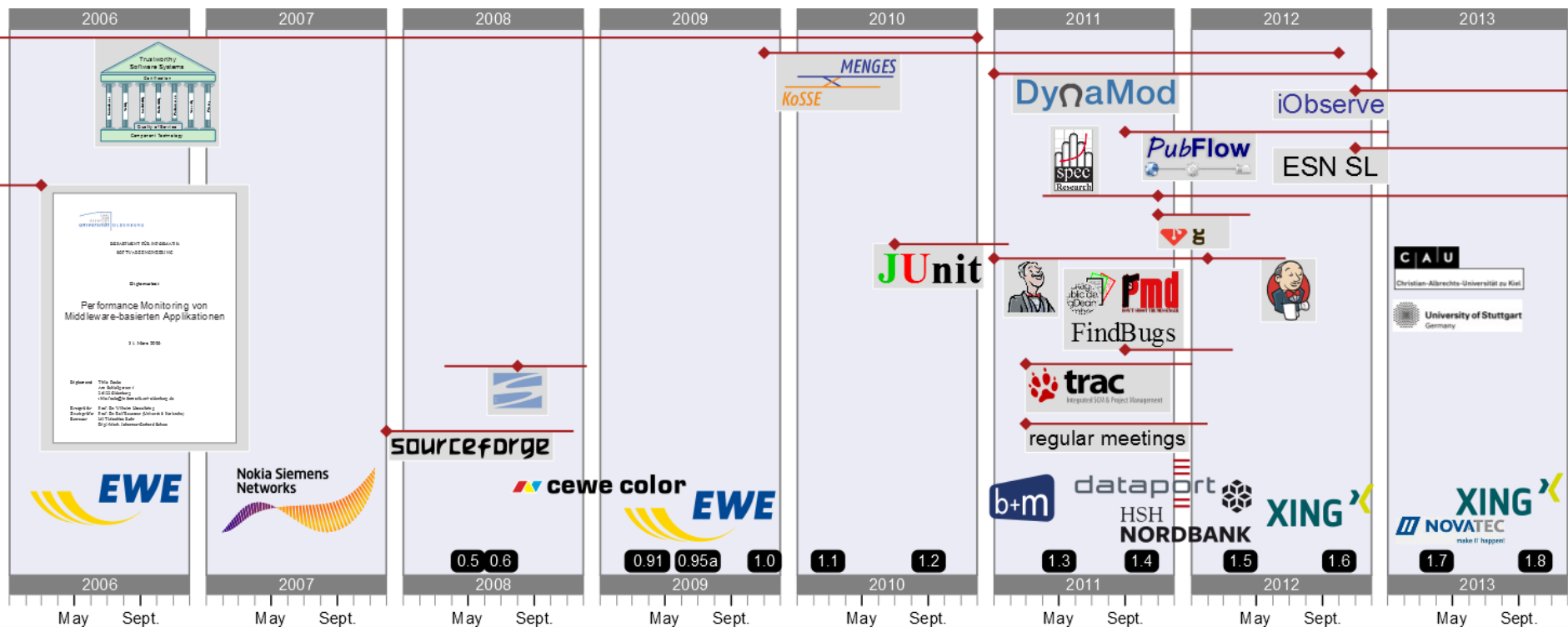
Performance comparison with MooBench

Performance Comparison

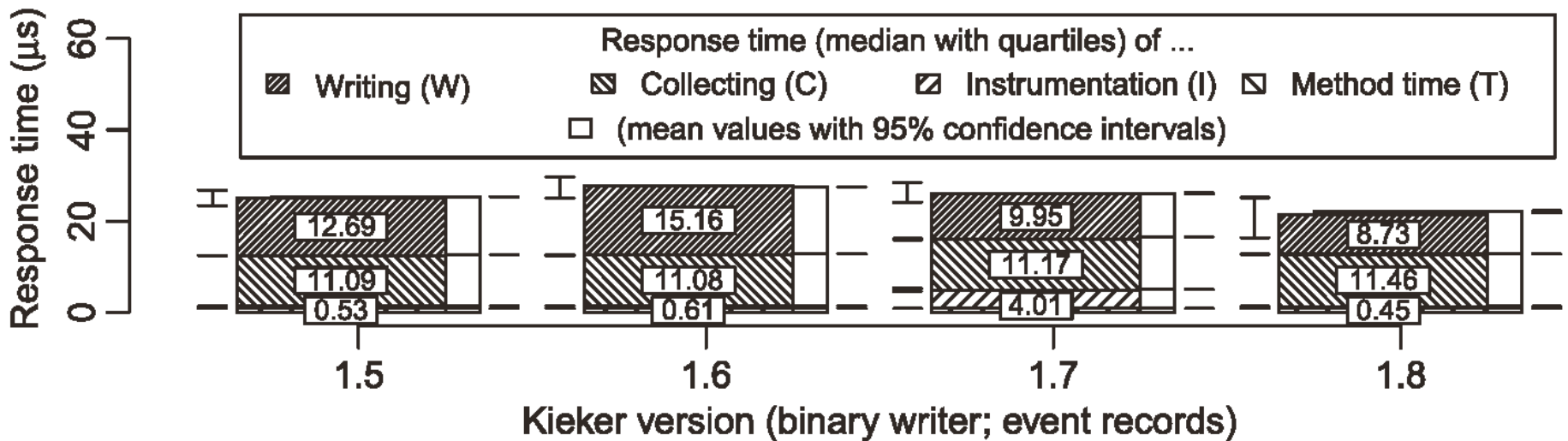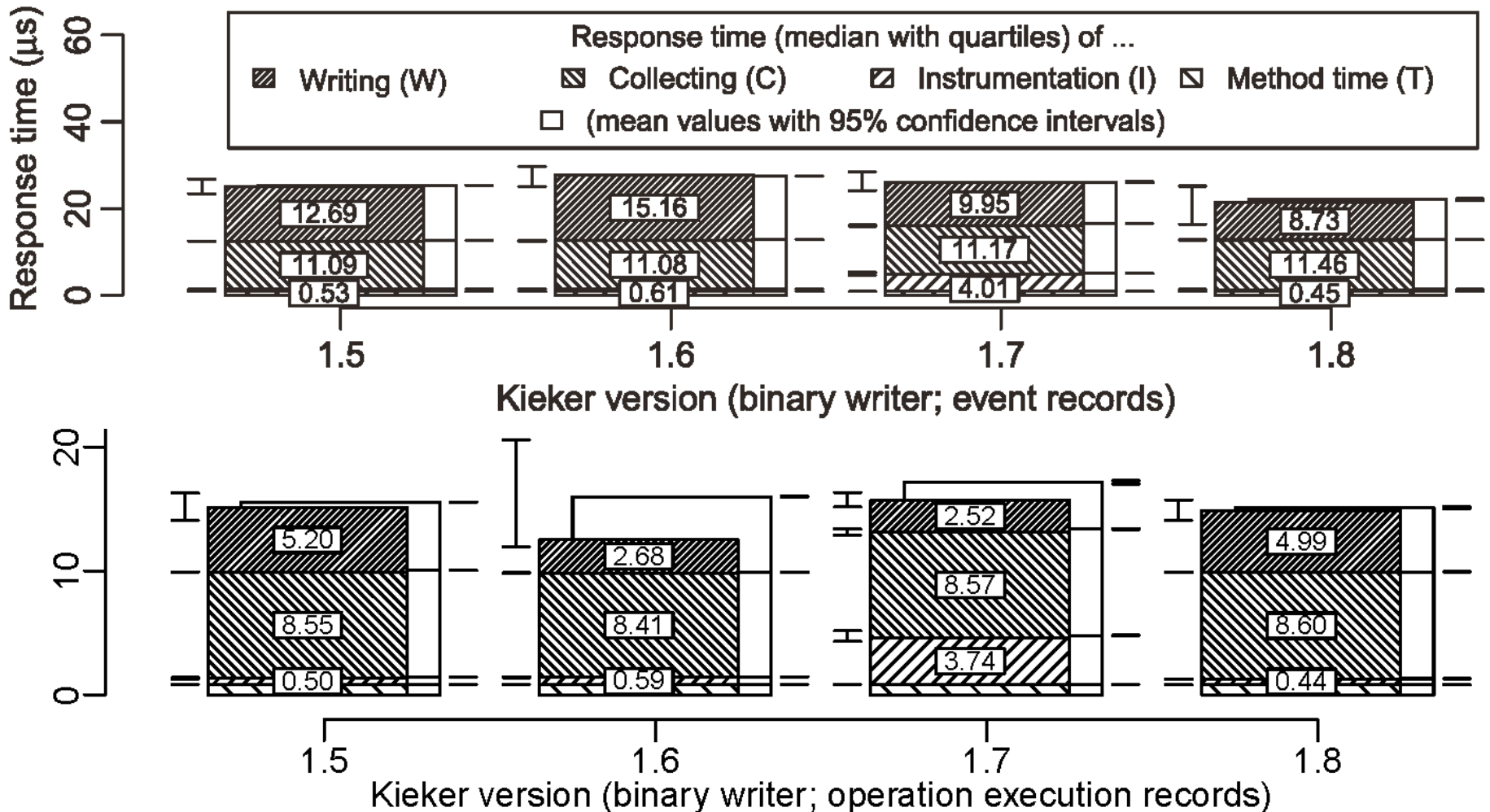Kieker version (ASCII writer; operation execution records)

Performance Comparison (cont.)

Performance Comparison (cont.)

Performance Comparison (cont.)
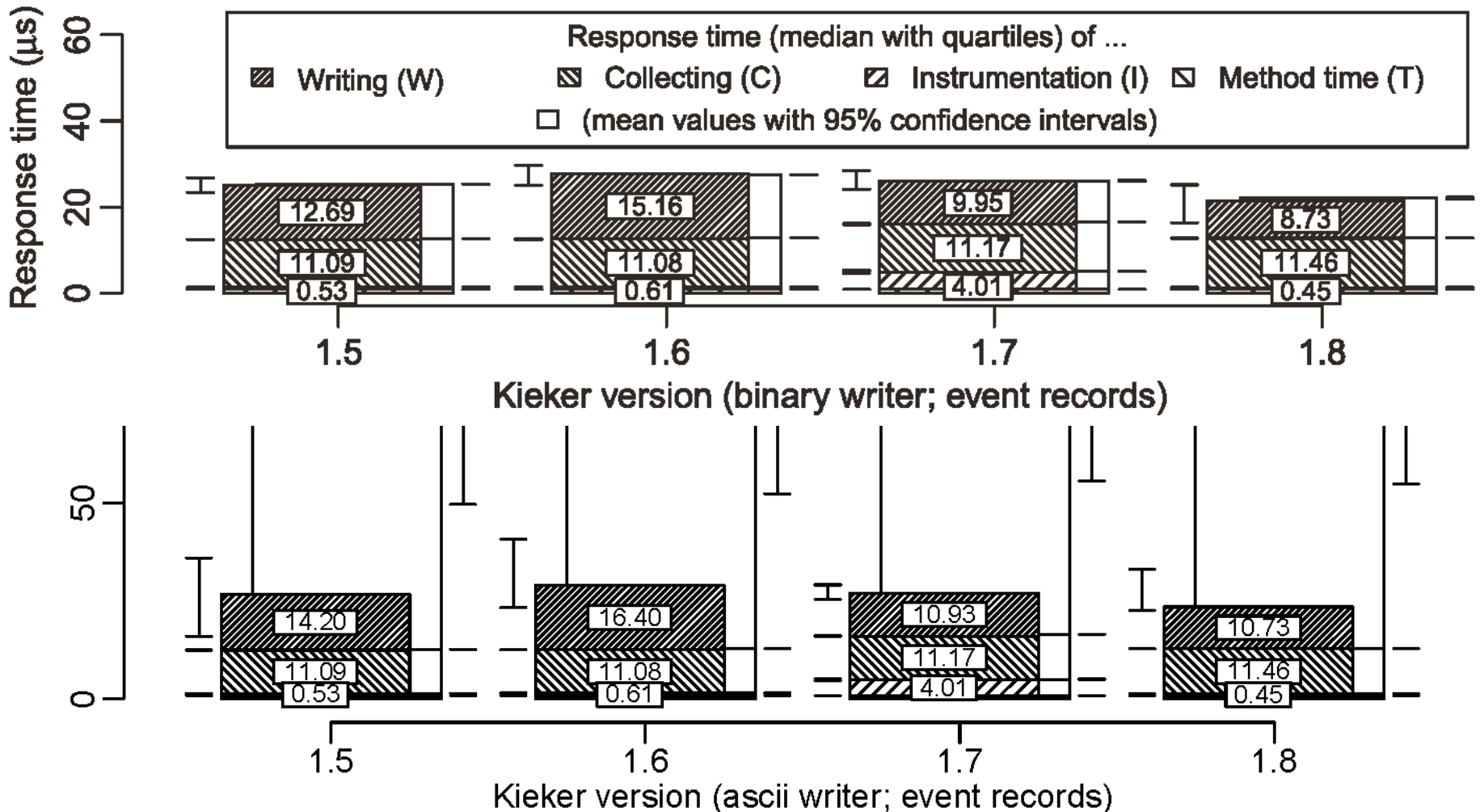
Performance Comparison (AMD)

Performance Comparison (long methodtime)

Replication & Validation

- All results available online

  – raw results and generated diagrams

- MooBench as open-source software

- Prepared experiments for all Kieker versions

- Detailed description of experiments in paper

- Further results and downloads:

  http://kieker-monitoring.net/overhead-evaluation/

Related Work

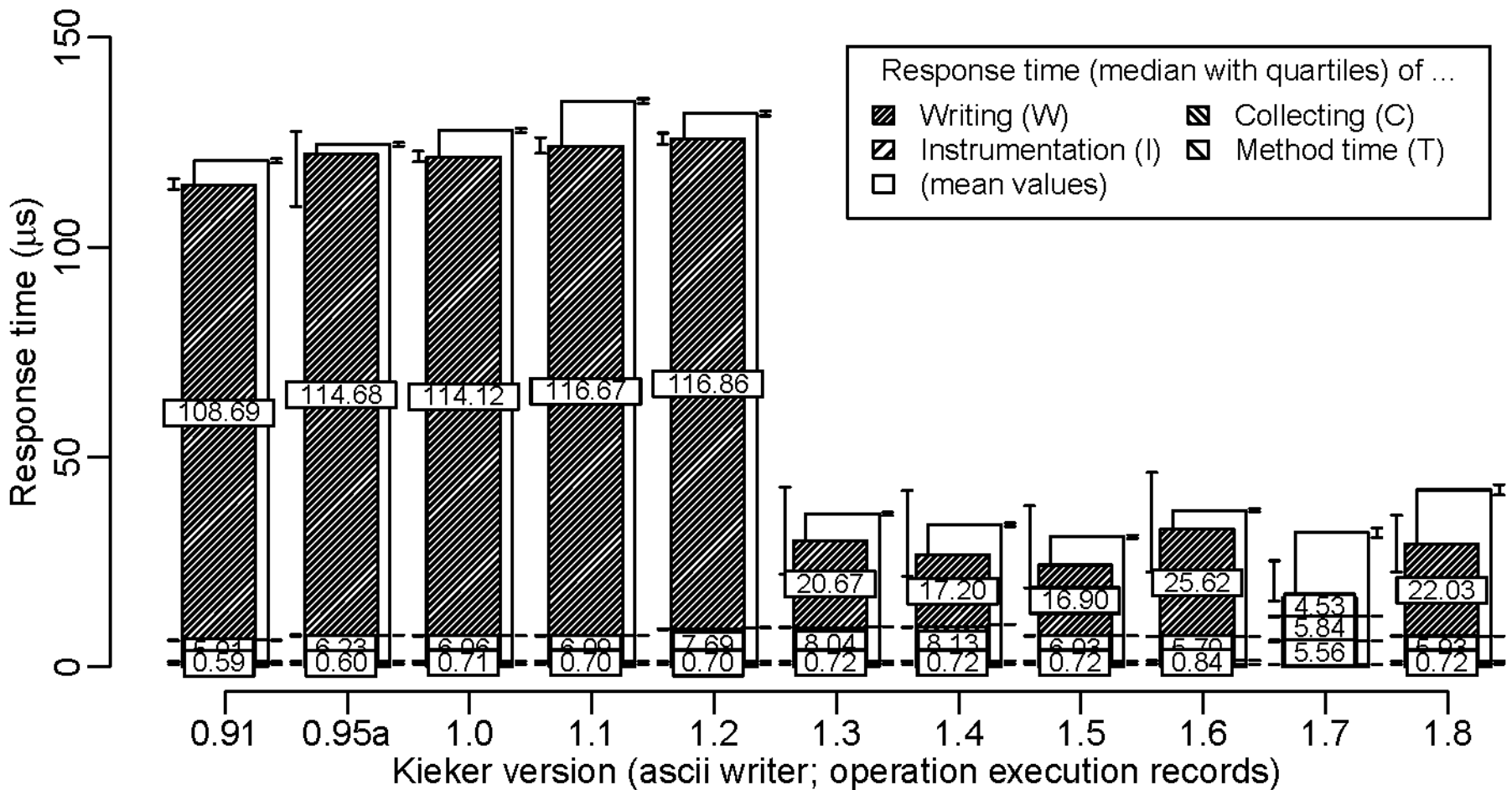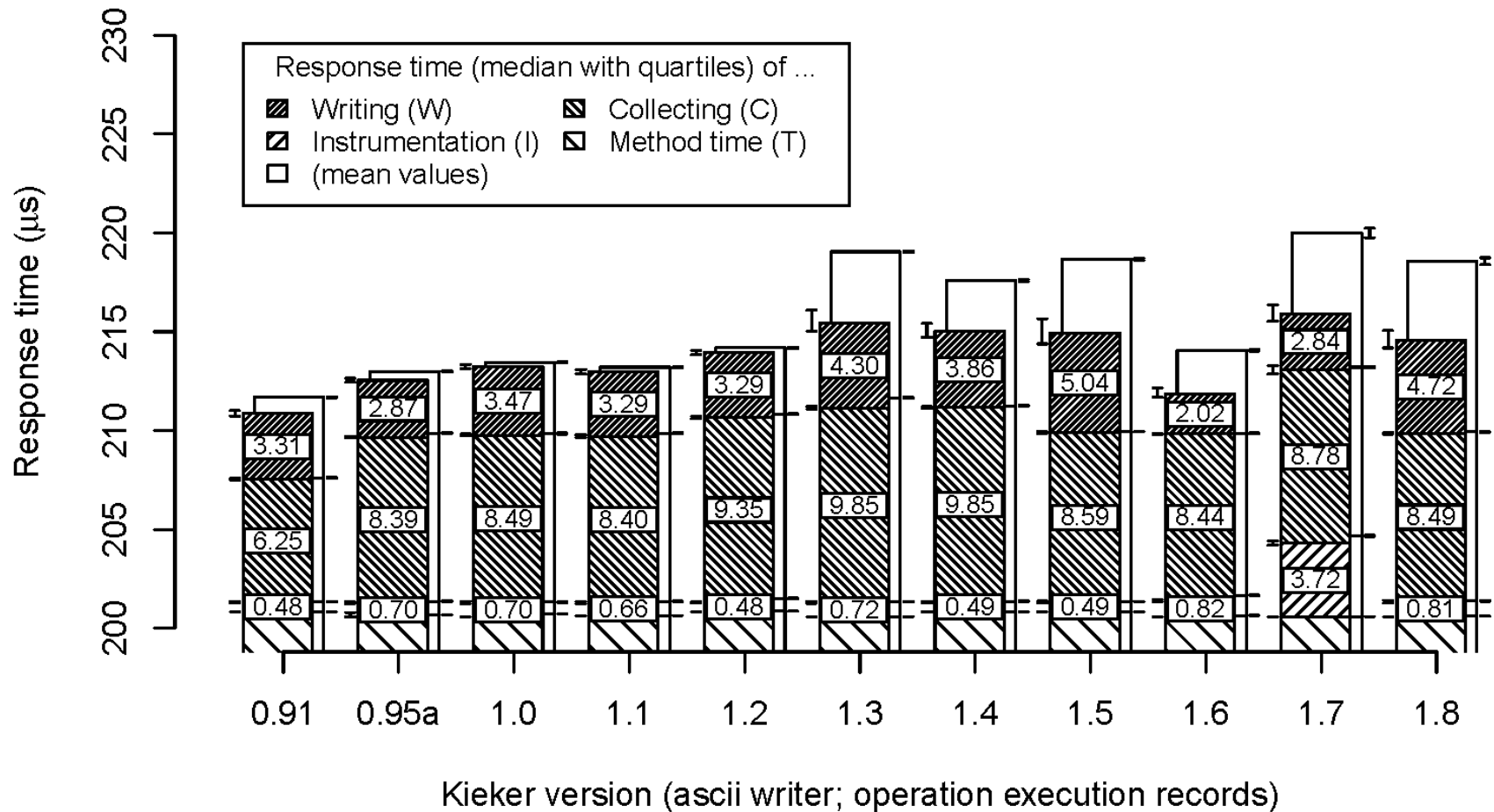- Benchmark engineering
  - lack of [Hin88, Pri89, Sac11, FAS+12, VMSK12]
  - requirements [Gra93, Hup09, Sac11]

- Benchmarks for monitoring
  - AppDynamics [App10]
    - KonaKart as macro-benchmark
    - comparison with and without monitoring
  - SpassMeter [ES12]
    - SPECjvm2008 as series of micro-benchmarks
    - also compares to Kieker

Conclusions

- Definition of monitoring overhead

- Benchmark engineering methodology

- MooBench (**Mo**nitoring **o**verhead **Bench**mark)

- Performance comparison of Kieker versions

http://kieker-monitoring.net/MooBench

http://kieker-monitoring.net

References

[App10] AppDynamics. AppDynamics Lite Performance Benchmark Report, 2010.

[ES12] Holger Eichelberger and Klaus Schmid. Erhebung von Produkt-Laufzeit-Metriken: Ein Vergleich mit dem SPASS-Meter-Werkzeug. In Proceedings of the DASMA Metrik Kongress, pages 171–180. Shaker Verlag, 2012. In German.

[FAS+12] Enno Folkerts, Alexander Alexandrov, Kai Sachs, Alexandru Iosup, Volker Markl, and Cafer Tosun. Benchmarking in the Cloud: What it Should, Can, and Cannot Be. In Proceedings of the 4th TPC Technology Conference on Performance Evaluation & Benchmarking, pages 173–188. Springer, 2012.

[Gra93] Jim Gray, editor. The Benchmark Handbook: For Database and Transaction Systems. Morgan Kaufmann, 2 edition, 1993.

[Hin88] David F. Hinnant. Accurate Unix Benchmarking: Art, Science, or Black Magic? IEEE Micro, 8(5):64–75, 1988.

[Hup09] Karl Huppler. The Art of Building a Good Benchmark. In First TPC Technology Conference on Performance Evaluation and Benchmarking, pages 18–30. Springer, 2009.

[Pri89] Walter J. Price. A Benchmark Tutorial. IEEE Micro, 9(5):28–43, 1989.

[Sac11] Kai Sachs. Performance Modeling and Benchmarking of Event-Based Systems. PhD thesis, TU Darmstadt, Germany, 2011.

[vHRH+09] André van Hoorn, Matthias Rohr, Wilhelm Hasselbring, Jan Waller, Jens Ehlers, Sören Frey, and Dennis Kieselhorst. Continuous Monitoring of Software Services: Design and Application of the Kieker Framework. Technical Report 0921, Department of Computer Science, Kiel University, Germany, 2009.

[vHWH12] André van Hoorn, Jan Waller, and Wilhelm Hasselbring. Kieker: A Framework for Application Performance Monitoring and Dynamic Software Analysis. In Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, pages 247–248. ACM, 2012.

[VMSK12] Marco Vieira, Henrique Madeira, Kai Sachs, and Samuel Kounev. Resilience Benchmarking. In Resilience Assessment and Evaluation of Computing Systems, pages 283–301. Springer, 2012.

[WH12] Jan Waller and Wilhelm Hasselbring. A Comparison of the Influence of Different Multi-Core Processors on the Runtime Overhead for Application-Level Monitoring. In Multicore Software Engineering, Performance, and Tools, pages 42–53. Springer, 2012.