Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Assessing measurement and analysis performance and scalability of Scalasca 2.0

August 27, 2013 | Ilya Zhukov and Brian J.N. Wylie

Outline

- Motivation
- What is Scalasca?
- Scalasca1 vs Scalasca2
- Measurements
  - What?
  - Where?
  - Results
- Conclusion

Motivation

- Scalasca is a scalable performance-analysis toolkit for parallel codes
  - Exists since 1998
  - Has its own instrumentation and measurement infrastructure
- Other parallel performance tools (Vampir, TAU, Periscope) have different instrumentation or/and measurement systems
  - There are converters → not convenient
- Integration is needed
  - Score-P (Scalable Performance Measurement Infrastructure for Parallel Codes)
    - Collaboration of tools' developers
    - Uniform instrumentation and run-time management
    - Uniform trace format (OTF2)
    - Uniform profile format (CUBE4)
- Scalasca supports Score-P since 2013 (Score-P 1.2)
  - What is benefit? Overhead?

scalasca

Scalable performance-analysis toolkit for parallel codes

- Specifically targeting large-scale applications running with 100,000s of processes or 1,000,000 threads

Integrated performance-analysis process

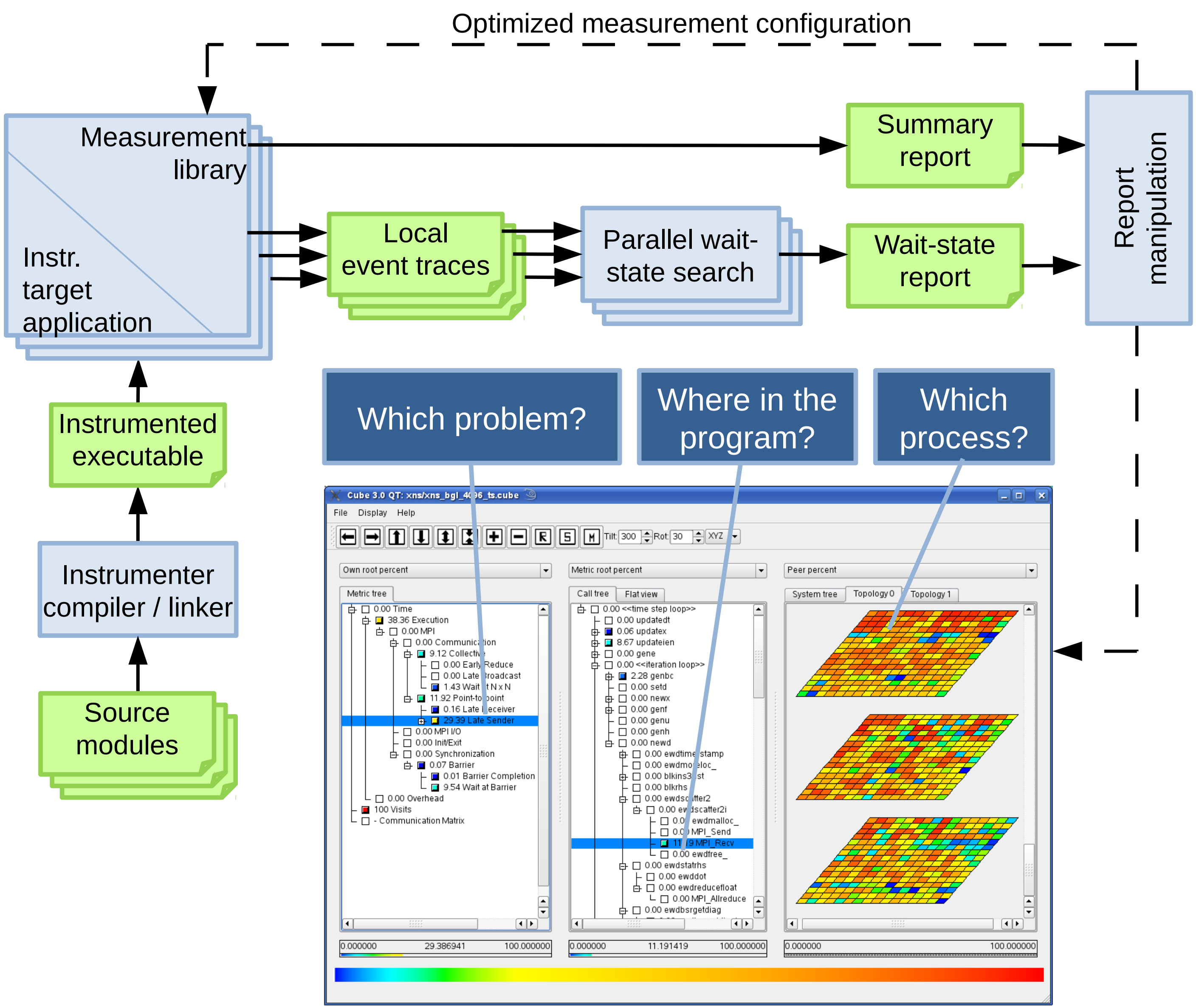
- Performance overview via call-path profiles
- In-depth study of application behavior via event tracing
  - *Automatic trace analysis identifying wait states*

Supports MPI 2.2 and basic OpenMP

License: new BSD

Website: http://www.scalasca.org

Scalasca architecture



Scalasca1 vs. Scalasca2

	Scalasca1	Scalasca2
Instrumentation & measurement system	EPIK	Score-P
Command line switches	different	
Manual instrumentation API	different	
Environmental variables	different	
Memory buffers	separate for each thread	memory pool on each process
Trace format	EPILOG	OTF2
IO	supports SIONlib	partially supports SIONlib
Report format	CUBE3	CUBE4

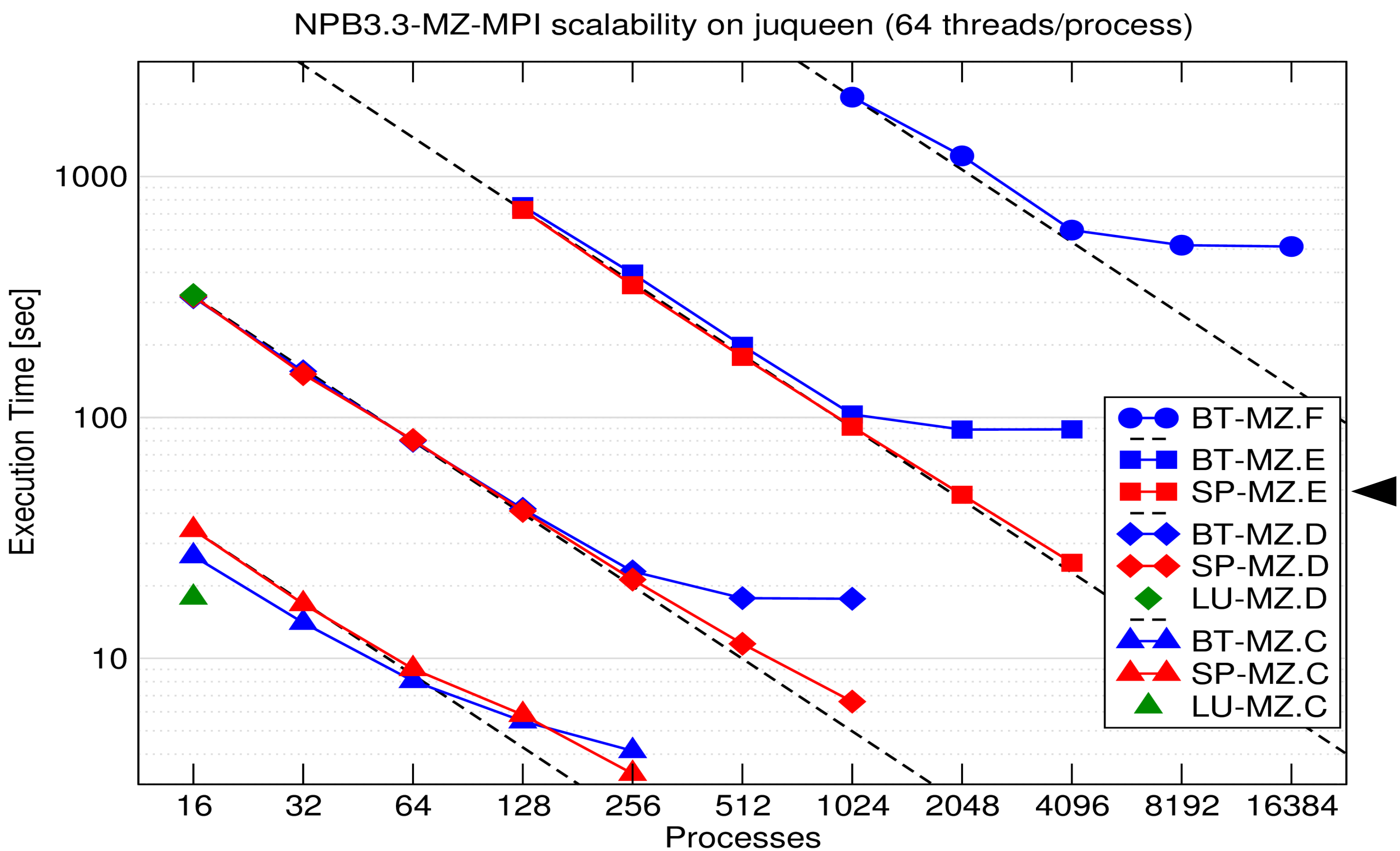
NAS parallel benchmark

- NPB3.3-MZ-MPI is a hybrid (MPI+OpenMP) Fortran77 parallel CFD application with various problem sizes ('classes')
- Solves a discretized version of unsteady, compressible Navier-Stokes equations in three spatial dimensions
- Has three 'multizone' variants
  - BT-MZ
    - ***B**lock **t**ri-diagonal solver*
    - *Unequal sizes of zones*
    - *Up to 16384 processes*
  - LU-MZ
    - ***L**ower-**u**pper Gauss-Seidel solver*
    - *Fixed number of equal-sized zones*
    - *Up to 16 processes*
  - SP-MZ
    - ***S**calar **p**enta-diagonal solver*
    - *Equal sizes of zones*
    - *Up to 16384 processes*

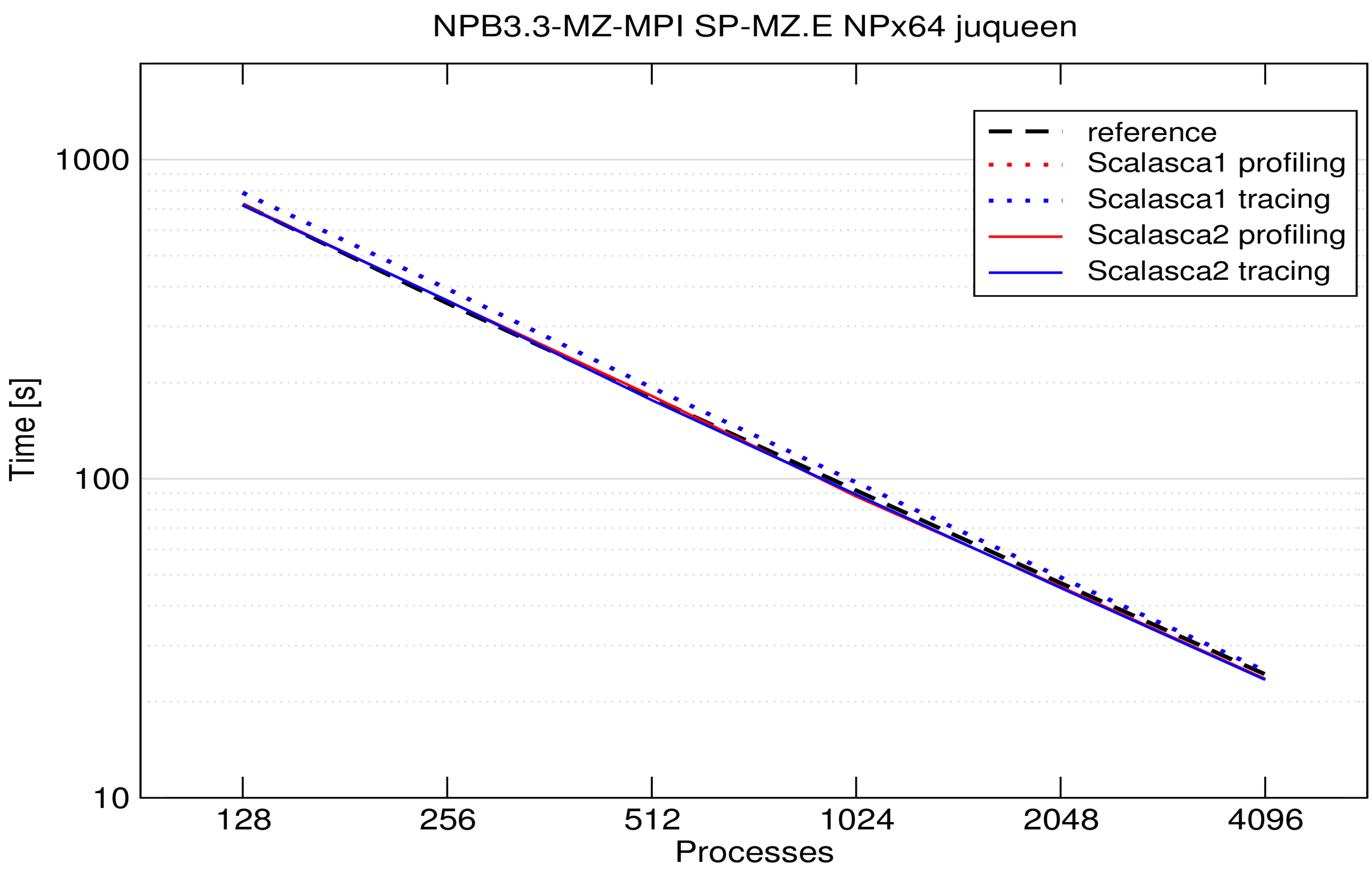
Experimental Setup

- Platform
  - JUQUEEN
    - *28 racks IBM Blue Gene/Q system*
    - *28,672 nodes*
      - 1 processor = 16 cores (4-way hardware threading)
      - 16 GB memory per node
- Software
  - IBM XL compiler
  - Blue Gene/Q MPI implementation
- Configuration
  - 1 task per node with 64 threads

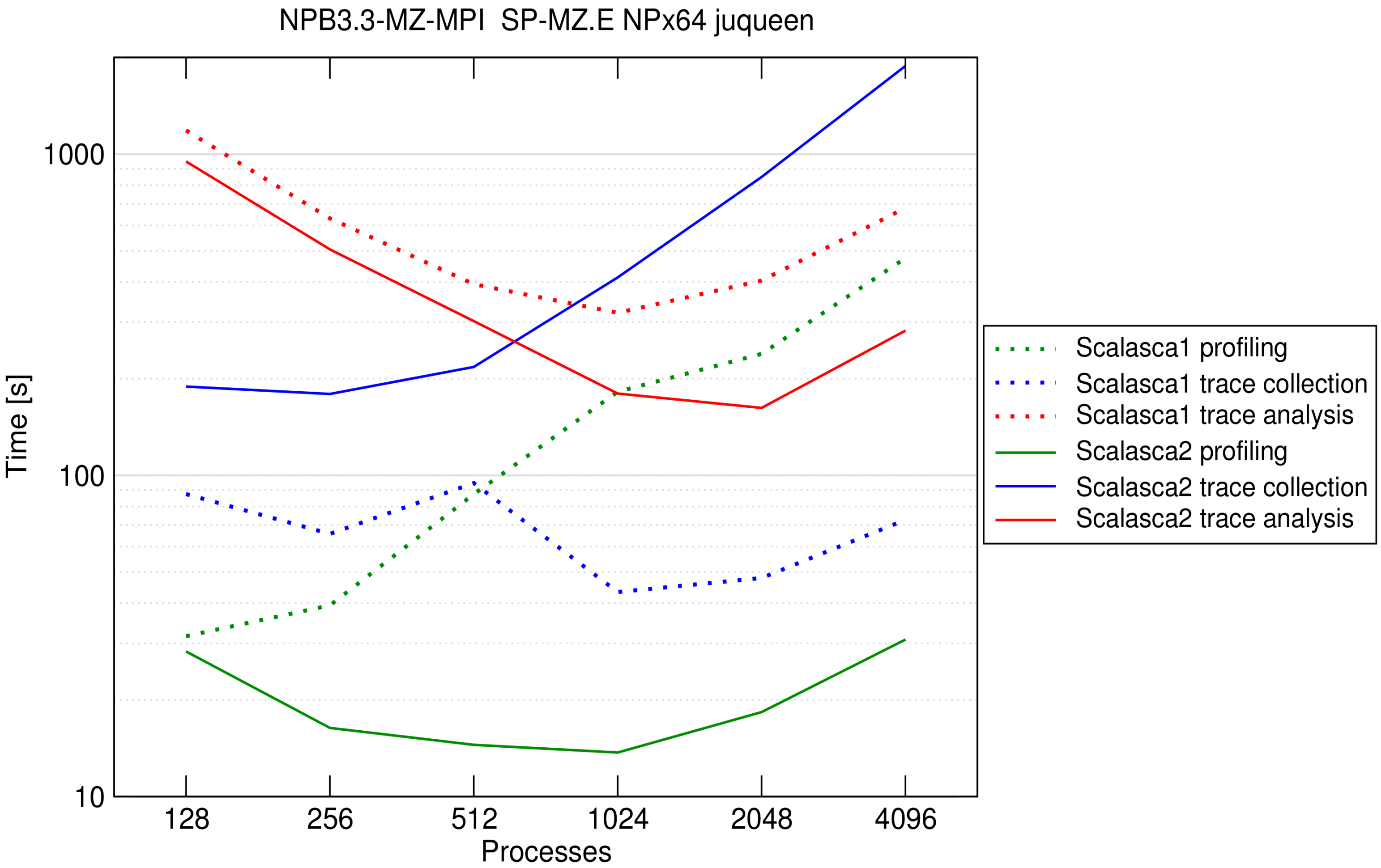
Scalability of uninstrumented application



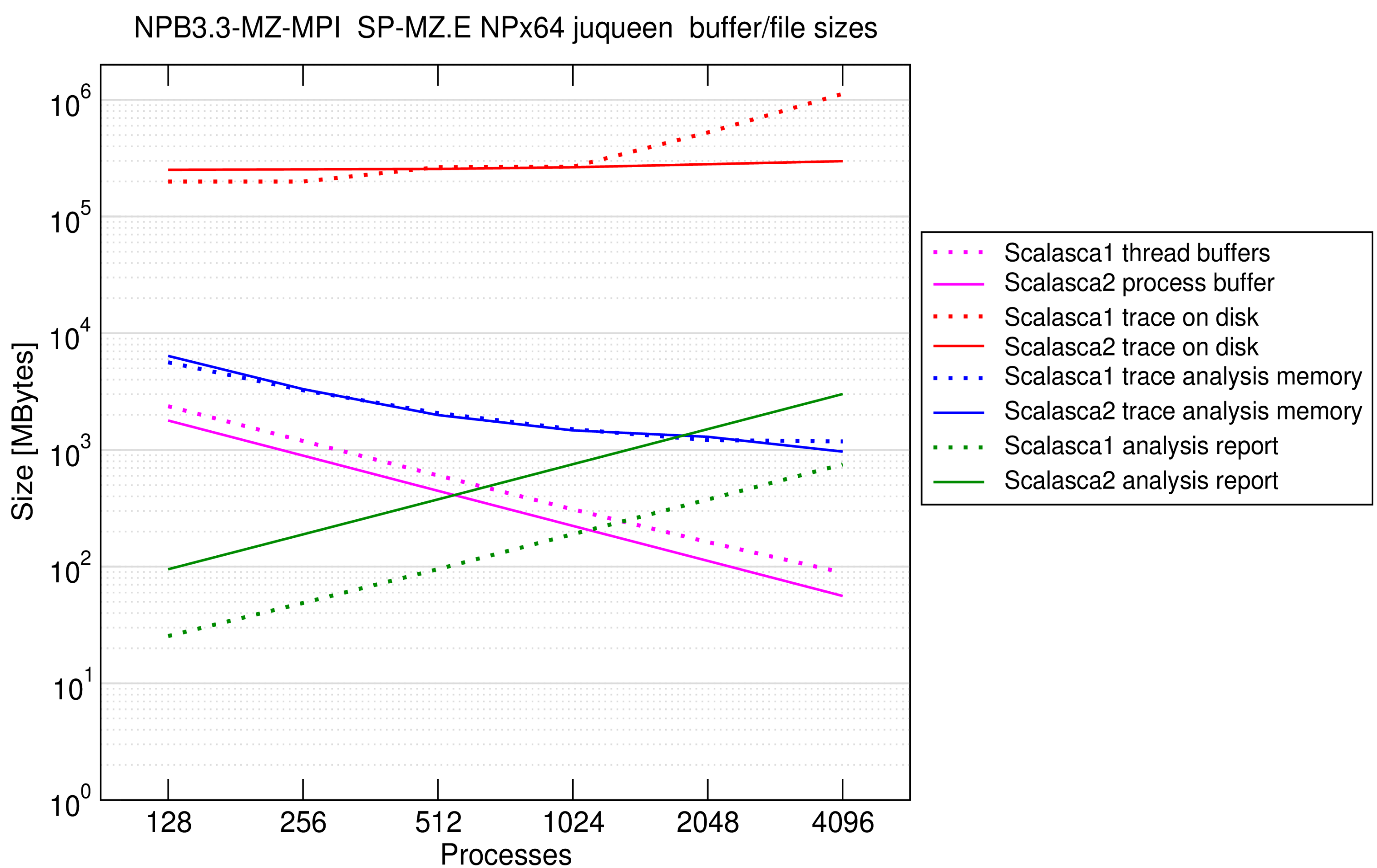
Scalability of instrumented application kernel



Scalability of instrumented application overhead



Scalability buffer and trace sizes



Scalability summary

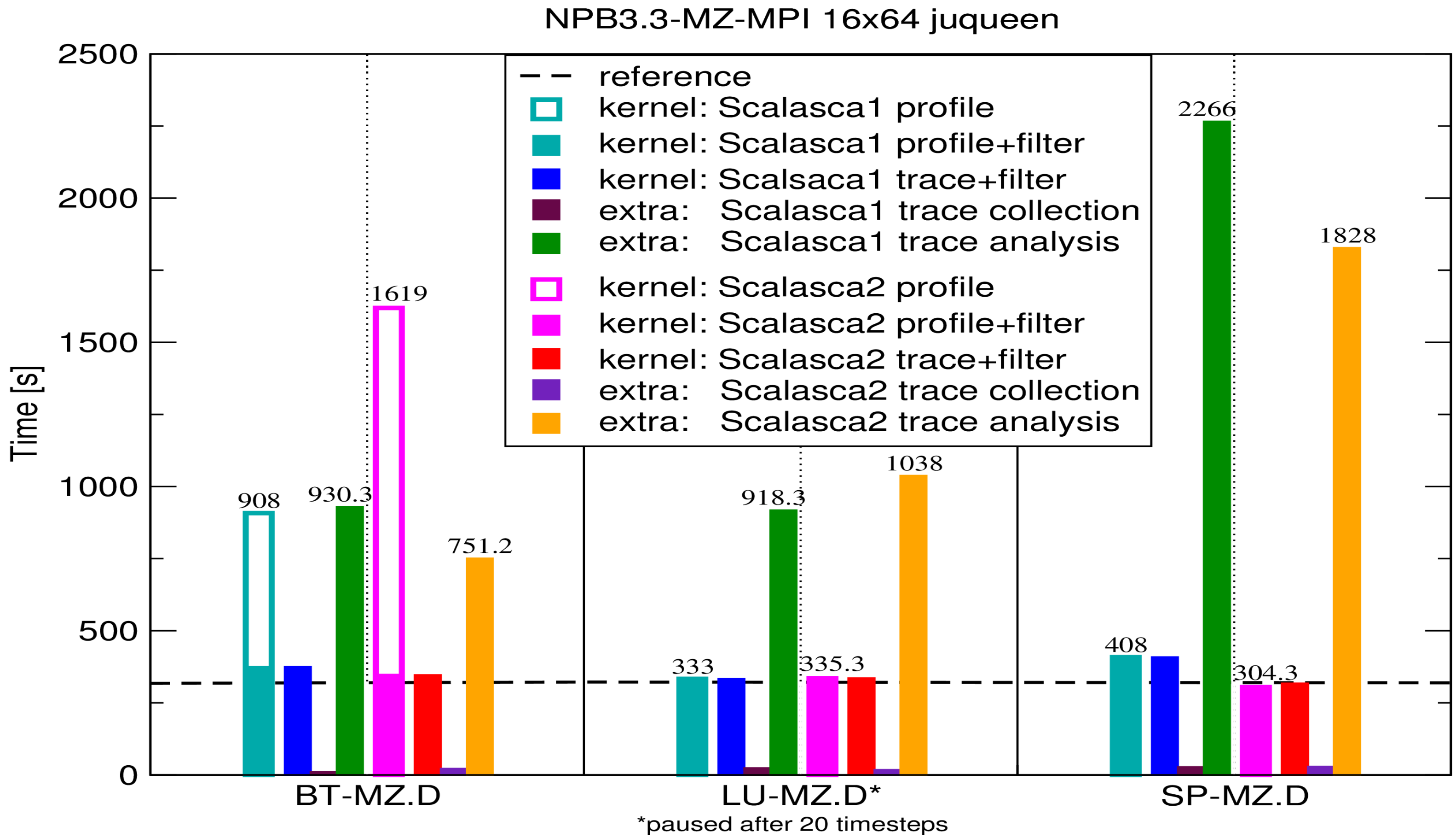
Profiling

- Measurement dilation (Scalasca1 11-28%, Scalasca2 5-7%)
- Profile collation in Scalasca2 is faster (less metrics and metadata)

Tracing

- Trace writing is slower in Scalasca2 (currently no SIONlib support for hybrid applications, new OTF2 format)
- Analysis report collating and writing is faster in Scalasca2
- Timestamp correction is considerably faster in Scalasca2
- Trace replay analysis is slightly faster in Scalasca2
- Memory and buffer sizes
  - Trace collection buffer uses space more efficiently for Scalasca2
  - Trace analysis requires similar amount of memory
  - Final report is larger in Scalasca2 (no compression)

NPB3.3-MZ-MPI class D



	16x64	16x64	16x64
	BT-MZ	LU-MZ	SP-MZ
S1 default thread max_tbc, [MB]	6,438	1,603	119
S1 revised thread max_tbc, [MB]	31	86	
S2 default process max_tbc, [MB]	182,249	21,692	4,991
S2 revised process max_tbc, [MB]	1,441	1,988	

Benchmark comparison summary

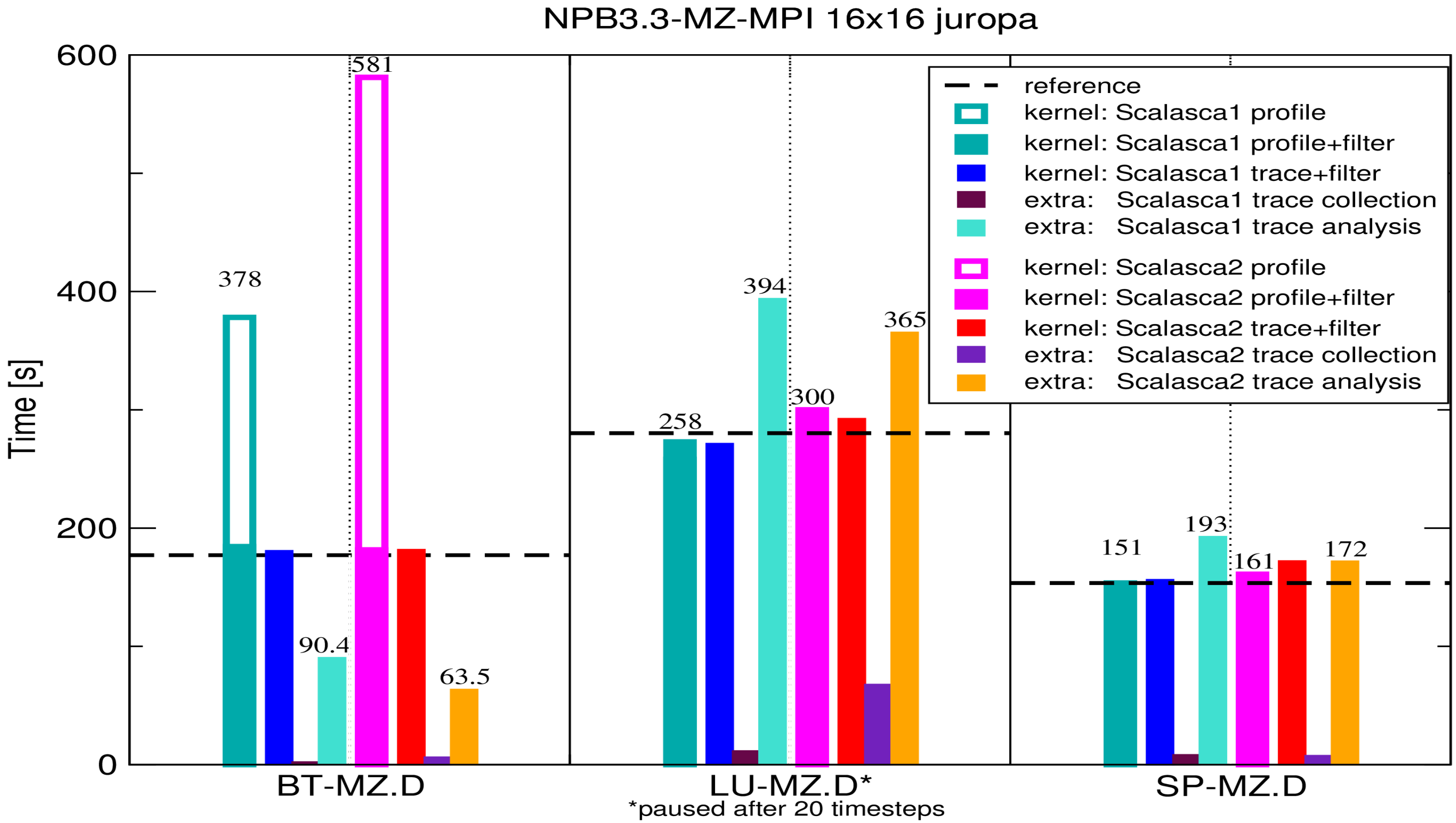
- Filtering is necessary for several very frequently called small routines in BT-MZ
  - Reduces measurement dilation and trace size
- Trace analysis of LU-MZ is only possible for smaller number of iterations
  - Pause measurement to record only 20 out of 300 iterations
  - Important for analysis to retain OpenMP flush operations
- Measurement and analysis times are comparable for Scalasca1 and Scalasca2

Conclusion

- Functionality validation of Scalasca2 with hybrid MPI+OpenMP benchmark codes on JUQUEEN
  - Also on JUROPA Linux cluster (see paper for details)
- Performance and scalability are comparable to its predecessor Scalasca1
- To be improved:
  - SIONlib support
  - CUBE4 metric remapper
  - CUBE4 compression

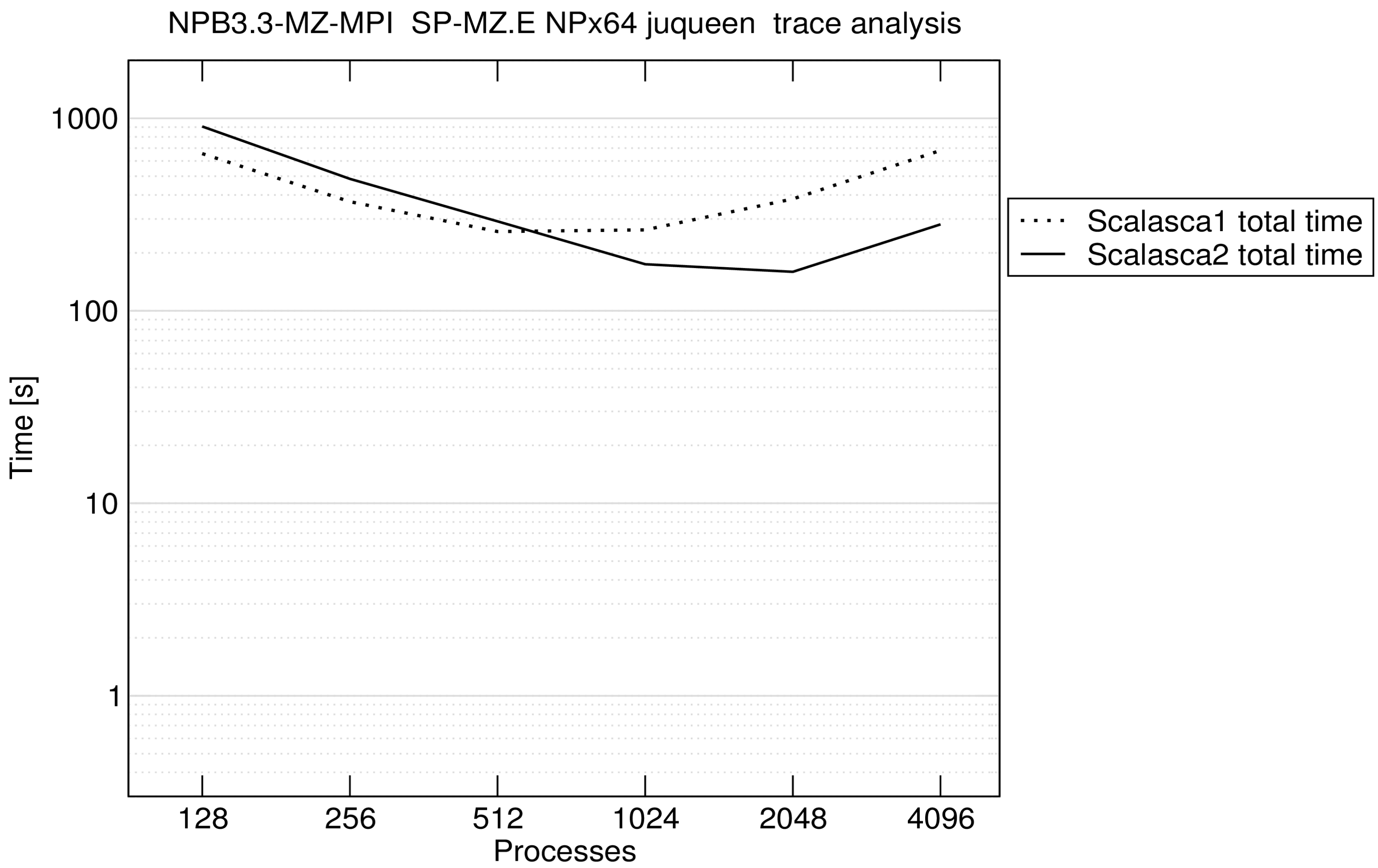
Appendix

NPB3.3-MZ-MPI class D

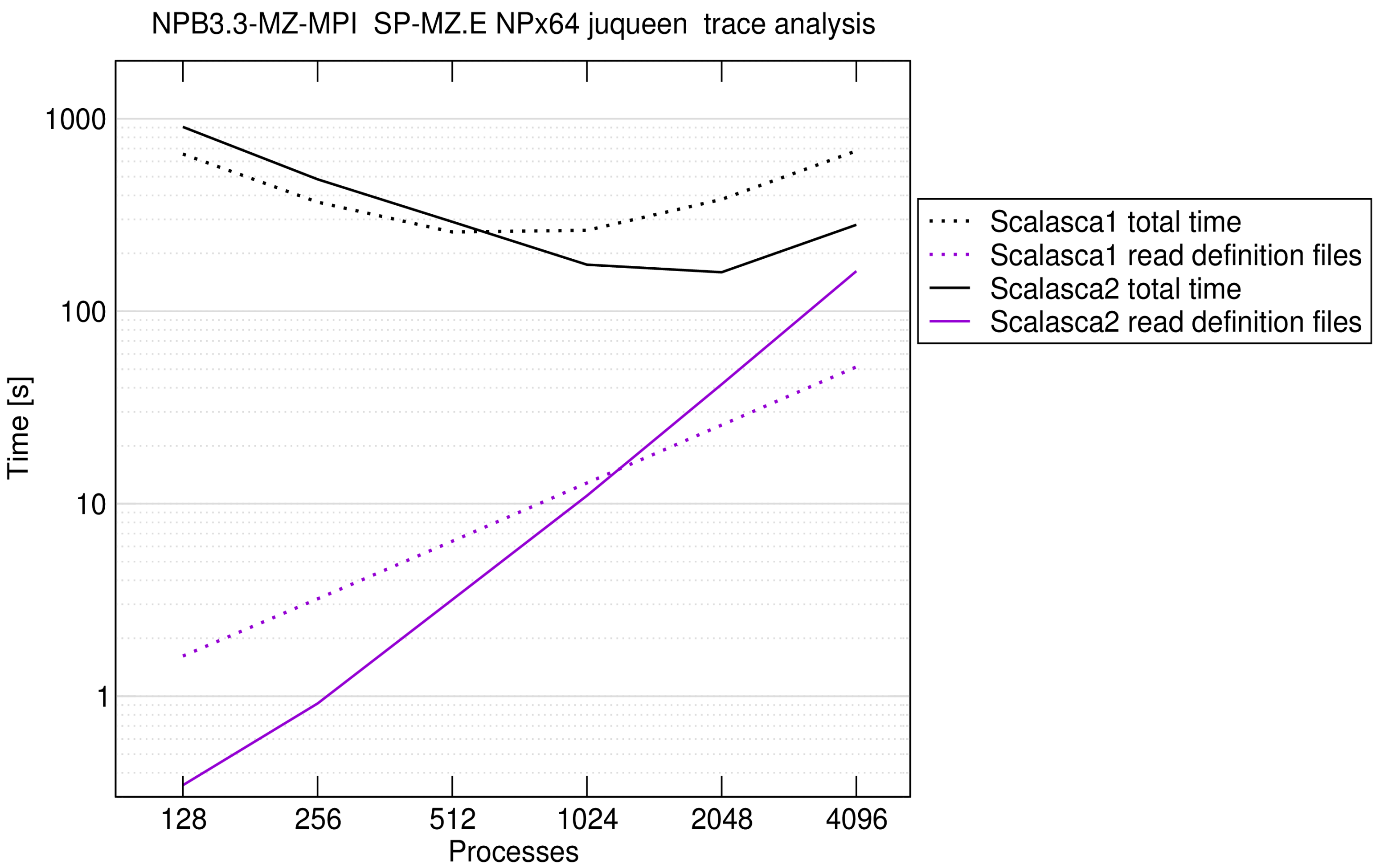


	16x64	16x64	16x64
	BT-MZ	LU-MZ	SP-MZ
S1 default thread max_tbc, [MB]	13,488	6,396	74
S1 revised thread max_tbc, [MB]	31	602	
S2 default process max_tbc, [MB]	181,174	21,230	2,321
S2 revised process max_tbc, [MB]	365	3,120	

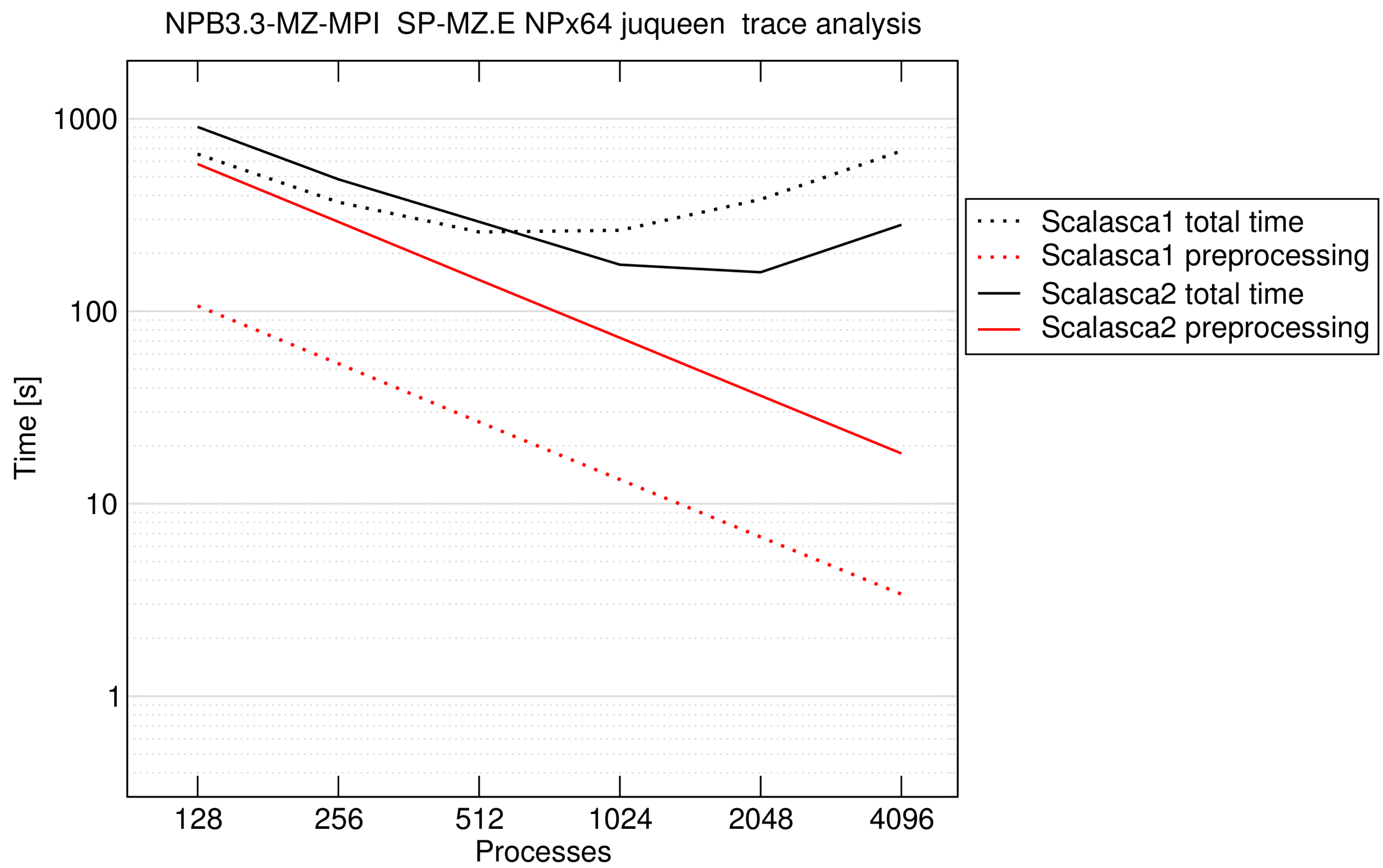
Trace analysis (1)



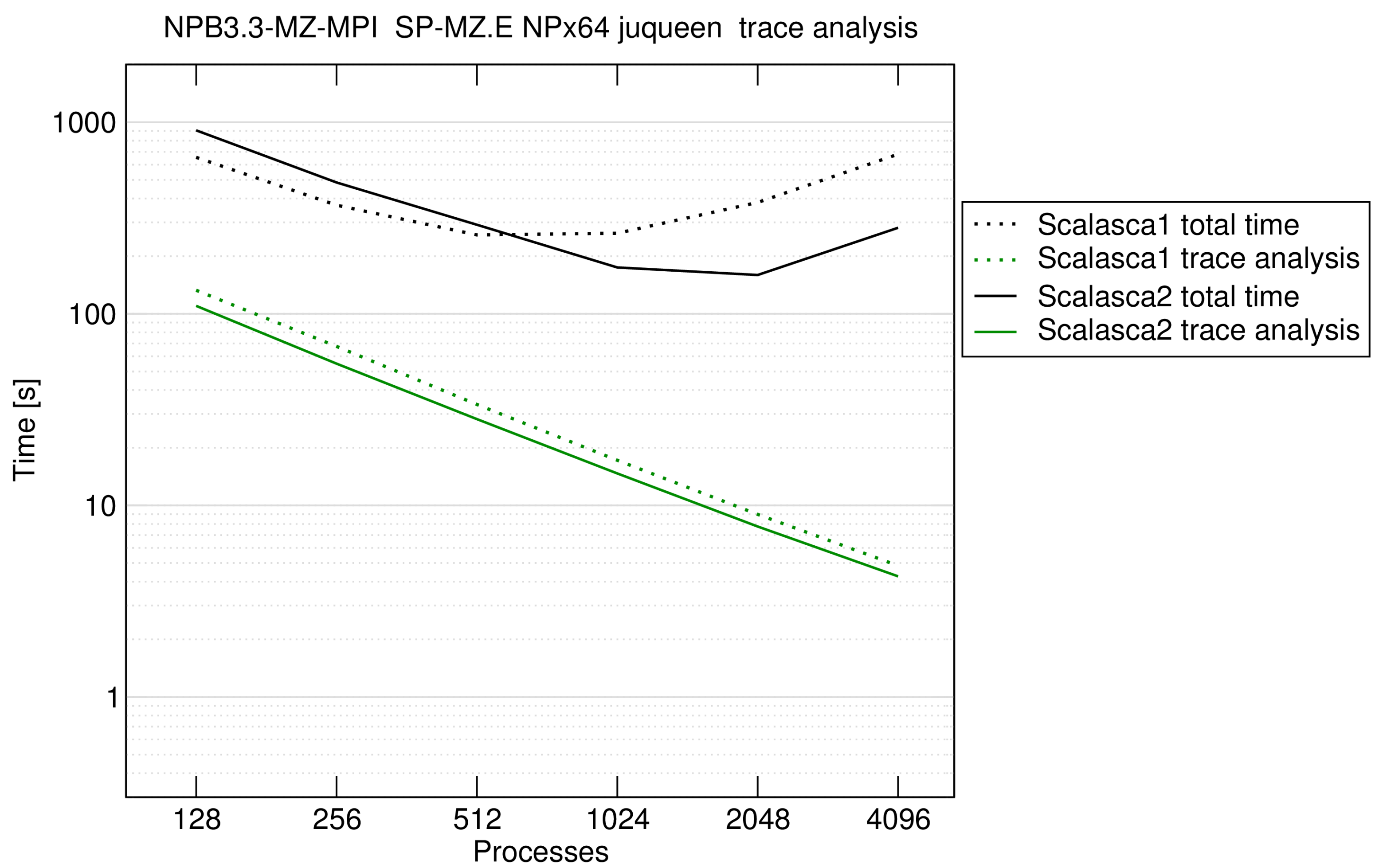
Trace analysis (2)



Trace analysis (4)



Trace analysis (5)



Trace analysis (6)