Scalasca GCS Project Report

## Name/affiliation of principal investigator

Dr.-Ing. Bernd Mohr, JSC, Forschungszentrum Jülich GmbH

## Name(s)/affiliation(s) of project team members

Christian Feld, JSC, Forschungszentrum Jülich GmbH  
Dr. Markus Geimer, JSC, Forschungszentrum Jülich GmbH  
Michael Knobloch, JSC, Forschungszentrum Jülich GmbH  
Dr. Pavel Saviankou, JSC, Forschungszentrum Jülich GmbH  
Marc Schlütter, JSC, Forschungszentrum Jülich GmbH  
Anke Visser, JSC, Forschungszentrum Jülich GmbH  
Dr. Brian Wylie, JSC, Forschungszentrum Jülich GmbH  
Ilya Zhukov, JSC, Forschungszentrum Jülich GmbH

## Project name + Project ID

Scalable Performance Analysis of Large-Scale Parallel Applications (SCALASCA)

## Which HPC system served as computing platform?

JUWELS: Cluster & Booster & GPU modules

## Abstract

Jülich Supercomputing Centre (JSC) has a long tradition to develop tools for the performance instrumentation, measurement and analysis of HPC applications. The Score-P instrumentation and measurement infrastructure is a tool suite for profiling and event tracing of HPC applications. Scalasca integrates runtime summaries with in-depth studies of concurrent behaviour via event tracing and automated trace analysis. A distinctive feature of Scalasca is its ability to identify and quantify the cost of wait states even for very large processor counts. Using Score-P for instrumentation and measurement, Scalasca helps developers of parallel applications to significantly improve the efficiency and scalability of their codes. Benefits are two-fold, delivering more timely results of scientific computations and more efficiently utilising valuable computing resources.

## Introduction

Jülich Supercomputing Centre of Forschungszentrum Jülich GmbH has a long tradition to develop software packages for the performance instrumentation, measurement and analysis of HPC applications. The Scalasca and Score-P performance toolsets have been specifically designed to analyze parallel application execution behaviour on large-scale systems, but which are also well suited for small- and medium-scale HPC platforms. Scalasca (<https://scalasca.org>) integrates runtime summaries with in-depth studies of concurrent behaviour via event tracing and automated trace analysis. A distinctive feature of Scalasca is its ability to identify and quantify the cost of wait states even for very large processor counts. The Score-P instrumentation and measurement infrastructure (<https://score-p.org>) is a highly scalable tool suite for profiling, event tracing, and online analysis of HPC applications. Using Score-P for instrumentation and measurement of applications parallelized with MPI and possibly combined with OpenMP or POSIX threads or kernel offload to GPUs, Scalasca helps developers of parallel applications to significantly improve the efficiency and scalability of their codes. Benefits are two-fold, delivering more timely results of scientific computations and more efficiently utilising valuable computing resources.

Assigned compute time was used to develop and test new features and further improve scalability of our tools. This project allocation for JUWELS Cluster/Booster/GPUs was also used to complement project allocations dedicated to other activities:

1. the former JSC Cross-sectional Team “Performance Analysis”, to support performance measurement and analysis of applications of JSC users,
2. the EU H2020 Performance Optimisation and Productivity (POP) Centre of Excellence, for its performance assessment services.

## Scientific work accomplished and results obtained

* New versions of the performance tools Scalasca (V2.6), Cube (V4.6) and Score-P (V7.0 & 7.1) were released, including the following enhancements:
  + Support for recording OpenCL 2.1/2.2 functions.
  + Support for recording Kokkos events.
  + Support for MPI-3 Improbe, Imrecv, Mprobe & Mrecv events.
  + Distinction of GPUs as separate devices in multi-GPU environments.
  + Added compile-time USR event filtering support for Intel compilers.
  + Improved management of instrumenting with Nvidia compiler and flushing of CUDA event buffer at measurement finalization.
  + Improved thread-safety of CUDA event adapter.
  + Improved handling of POSIX (vectorized) I/O operations.
  + Extended profile scoring utility with additional sorting modes and ability to generate an initial filter file.
  + Fix handling of OpenMP tasks outside of parallel regions.
  + Fix handling of MPI Alltoallw & MPI Ialltoallw when using MPI IN PLACE.
  + Added preset mode for multi-run measurements.
  + Added support for restricting trace collection to multiple filesystems.
  + Extend trace analysis to “orphaned threads” (from Pthreads, C++ threads, etc).
  + Improved consistency of wait-state and root-cause analyses.
  + Additional metric hierarchies created for CUDA, OpenCL & OpenACC.
  + GUI Advisor plugin extended to support hybrid MPI+OpenMP metrics.
  + Added task view to call-tree panel of GUI.
  + Extended information display of all tree attributes.
  + Extended functionality of metric editor.
  + Preliminary CUBE GUI client–server implementation.
* Application engagement
  + Large-scale performance analysis of multi-GPU nodes with SPEChpc 2021 benchmark codes using OpenACC and OpenMP target offload.
  + Large-scale performance assessment of multi-GPU nodes with new GPU-enabled versions of Nek5000 CFD application using OpenACC and OpenMP target offload combined with CUDA.
  + Preparatory porting, testing and performance analysis of other multi-GPU benchmarks and applications.

## Realization of the project

Our group organised several performance analysis and tuning workshops, and assisted with porting and scaling workshops using JUWELS Booster, all held virtually during the reporting period (2021-2022) due to pandemic restrictions. Separate project allocations (including those of the participants) could be used for these activities. The Scalasca project allocation was expected to be used for follow-up assessments and investigation of scaling issues (particularly with JUWELS Booster), however, due to both the immaturity of the applications and their very limited scalability at the time this was not necessary. Consequently, relatively little use of the project allocation was required.

The Nek5000 CFD code from KTH was ported and first measured during the JUWELS-Booster Porting Workshop. The code uses OpenACC and OpenMP offloading combined with CUDA, and these versions were compared and analysed to identify poor MPI scaling performance with 512 GPUs. On-going work is examining new versions using the Hypre preconditioner. Preparatory porting was done of the Hypersonic Task-based Research (HTR) Solver from TUM for hypersonic aerothermodynamics, which is written in Regent and using Legion with OpenMP for kernel offload to multi-GPU nodes, both of JUWELS GPU and JUWELS-Booster.

A noise-resilient performance modeling approach for applications running on heterogeneous systems using accelerator devices was prepared on JUWELS-Booster and JUWELS GPU with a deep neural network (DNN) trained on the CIFAR-100 image recognition dataset.

From a collaboration with HZDR/ORNL/U.Delaware, the new suite of SPEChpc 2021 benchmark codes and additional applications using MPI and either OpenACC or OpenMP target offload to multi-GPU nodes were executed and analysed on JUWELS-Booster and compared to other HPC systems with Nvidia V100 and AMD MI100 GPUs [SPEChpc2021].

## Publications with the appropriate acknowledgement

**[SPEChpc2021]** Holger Brunst, Sunita Chandrasekaran, Florina M. Ciorba, Nick Hagerty, Robert Henschel, Guido Juckeland, Junjie Li, Veronica G. Melesse Vergara, SandraWienke, and Miguel Zavala, “First experiences in performance benchmarking with the new SPEChpc 2021 suites”, in: Procceedings of the 2022 IEEE/ACM 22nd International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2022. <https://doi.org/10.1109/CCGrid54584.2022.00077>

## Additional references

* Holger Brunst, Sunita Chandrasekaran, Florina M. Ciorba, Nick Hagerty, Robert Henschel, Guido Juckeland, Junjie Li, Veronica G. Melesse Vergara, Sandra Wienke, and Miguel Zavala, “First experiences in performance benchmarking with the new SPEChpc 2021 suites”, 2022. <https://doi.org/10.48550/arXiv.2203.06751>
* Holger Brunst, Sunita Chandrasekaran, Florina M. Ciorba, Nick Hagerty, Robert Henschel, Guido Juckeland, Junjie Li, Veronica G. Melesse Vergara, Sandra Wienke, and Miguel Zavala, “First Experiences in Performance Benchmarking with the New SPEChpc 2021 Suites - Measurement Data”, Zenodo, Oct. 2021. <https://zenodo.org/record/5753669>
* Marcus Ritter, Alexander Geiß, JohannesWehrstein, Alexandru Calotoiu, Thorsten Reimann, Torsten Hoefler, and Felix Wolf, Noise-Resilient Empirical Performance Modeling with Deep Neural Networks, in: Proc. of the 35th IEEE International Parallel and Distributed Processing Symposium (IPDPS), Portland, Oregon, USA, pp. 23–34, IEEE. May 2021. <https://doi.org/10.1109/IPDPS49936.2021.00012>

## Suitable Graphics

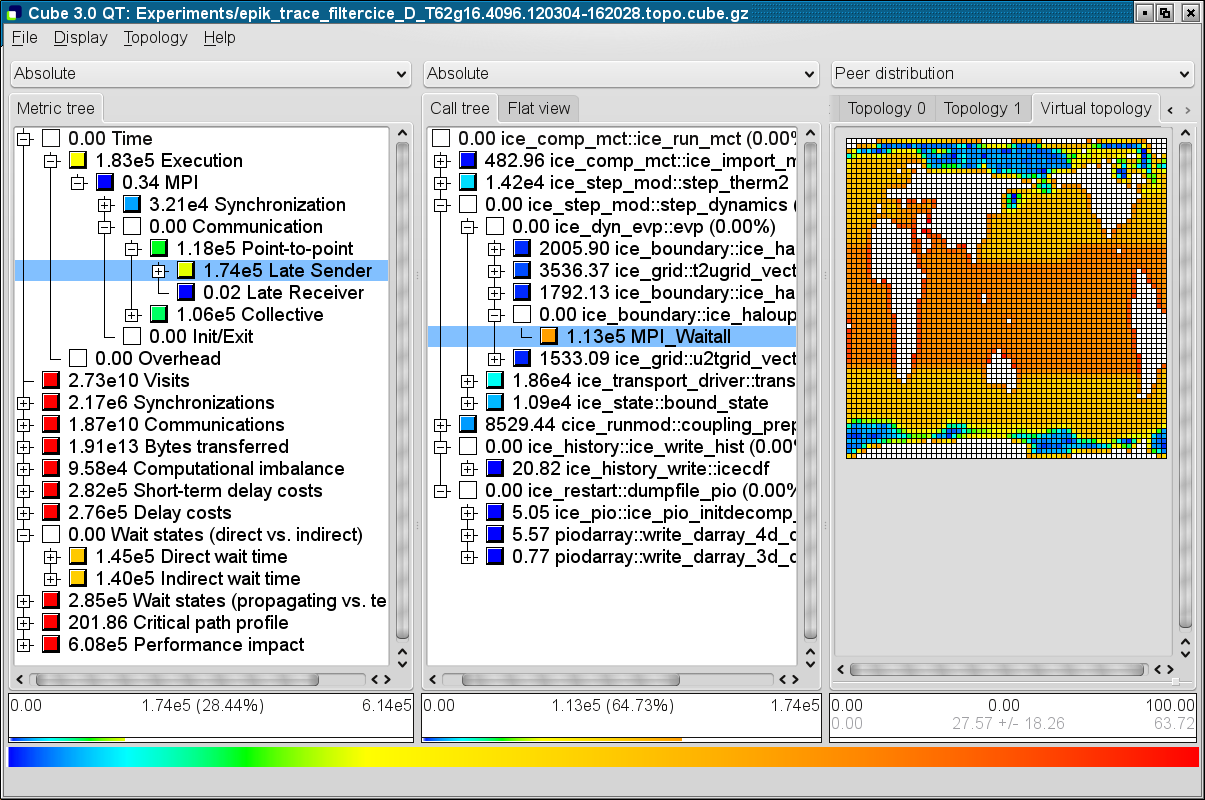


Figure 1: Typical result display from a Scalasca trace analysis. Left pane shows the metrics measured and calculated. Middle pane shows the distribution over the call tree of the application of the selected value in the metrics pane (blue bar). Right pane shows distribution of the selected call-path over the processes used for the measurement. In the example shown, this is done via a virtual MPI 2D Cartesian topology used by the application.