

# Developer tools for porting & tuning parallel applications

2015-12-04 | **Brian J. N. Wylie**  
Jülich Supercomputing Centre

b.wylie @ fz-juelich.de

## Overview

### JLESC Collaborative Project (start May 2015)

- JSC: Brian Wylie, Christian Feld – *Scalasca/Score-P/CUBE*
- RIKEN AICS: Miwako Tsuji, Hitoshi Murai – *XcalableMP*
- BSC: Judit Gimenez – *Paraver/Extrac/Dimemas, OmpSs*

### Goal

- Integration and improvement of respective developer tools for porting and tuning parallel applications on large-scale computer systems
- Joint training activities applying tools

## Tools training

XcalableMP tutorial at JSC (1 Dec 2016)

Scalasca & Paraver tools training offered through VI-HPS

- Virtual Institute – High-Productivity Supercomputing [[vi-hps.org](http://vi-hps.org)]
- focus on parallel performance, correctness & debugging tools
- VI-HPS Tuning Workshops several times each year
  - *3-5 days for application developers to get introduced to tools and receive assistance applying tools to their own codes*
  - *RIKEN AICS hosting VI-HPS-TW20 (24-26 Feb 2016) for users of K computer and related Fujitsu FX10/100 systems*
  - *additional workshops planned in Germany & France*

allinea

 JÜLICH  
FORSCHUNGSZENTRUM

RWTHAACHEN  
UNIVERSITY

 TECHNISCHE  
UNIVERSITÄT  
DRESDEN

UNIVERSITÉ DE  
VERSAILLES  
ST-QUENTIN-EN-YVELINES

 Universität Stuttgart

 BSC  
Barcelona  
Supercomputing  
Center  
Centre Nacional de Supercomputació

 Lawrence Livermore  
National Laboratory

 TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

 TUM  
Technische  
Universität  
München

 UNIVERSITY OF OREGON

 THE UNIVERSITY OF  
TENNESSEE

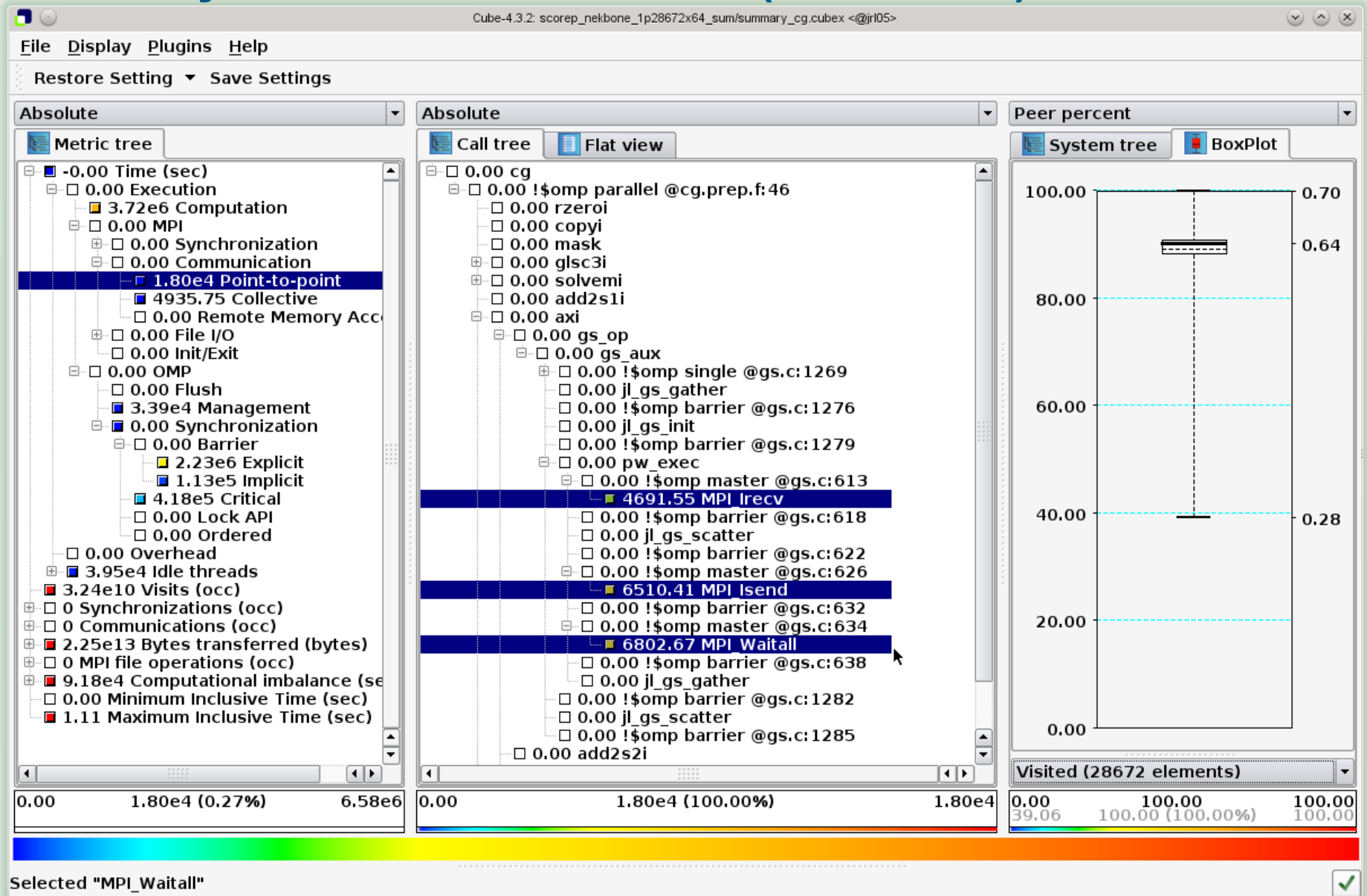
# Scalasca



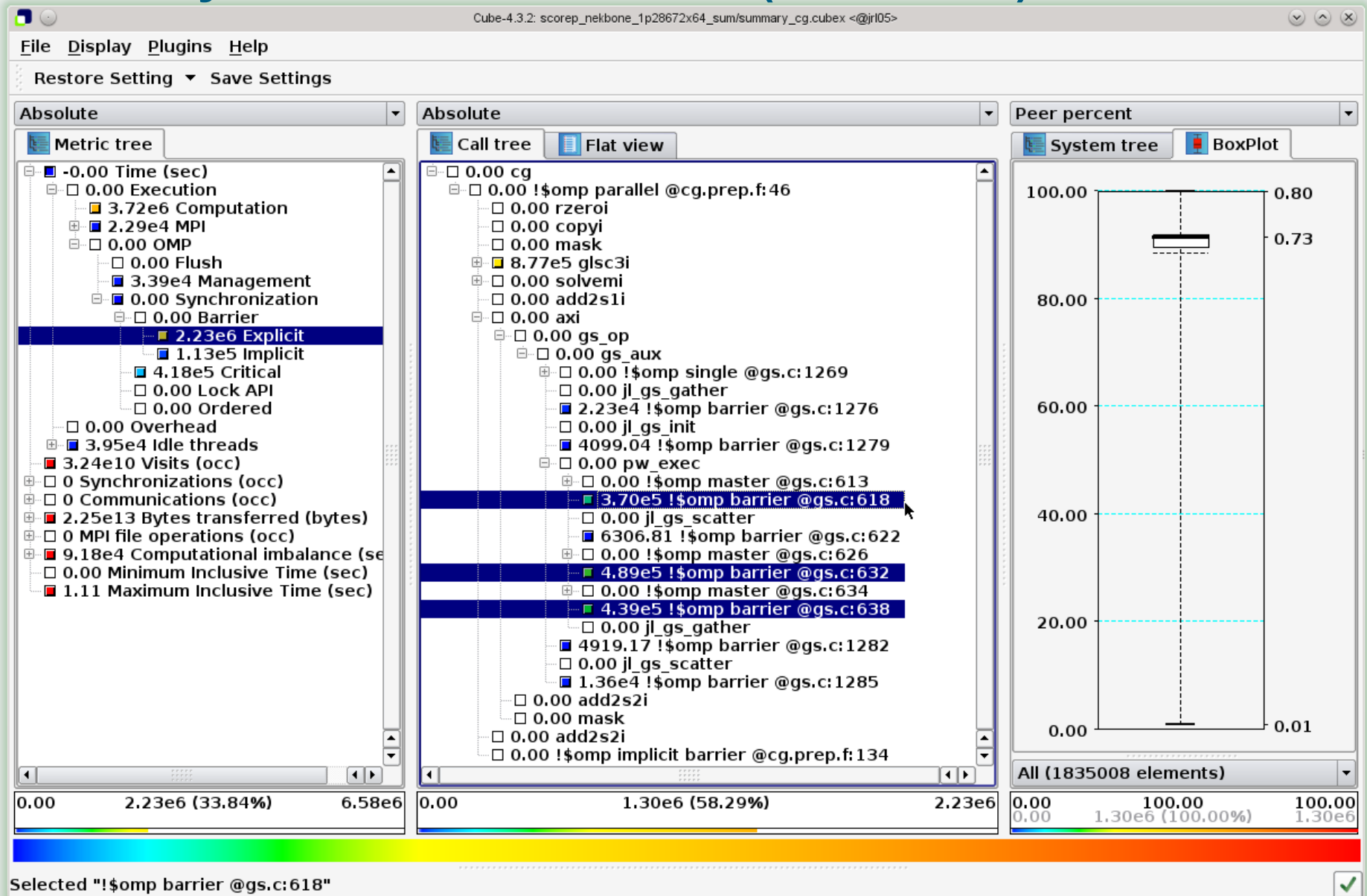
Developed to support scalable performance analysis of large-scale parallel applications

- available under open-source license from [www.scalasca.org](http://www.scalasca.org)
- offers flexible runtime summarization/profiling and event tracing
- based on Score-P instrumentation & measurement infrastructure and CUBE analysis report utilities & explorer GUI
- MPI + OpenMP, recently extended to support a variety of other threading paradigms (Pthreads, OpenCL, Qt, etc.), CUDA, etc.
- e.g., JUQUEEN: Nekbone 28,672 MPI x 64 OMP = 1.8 M threads

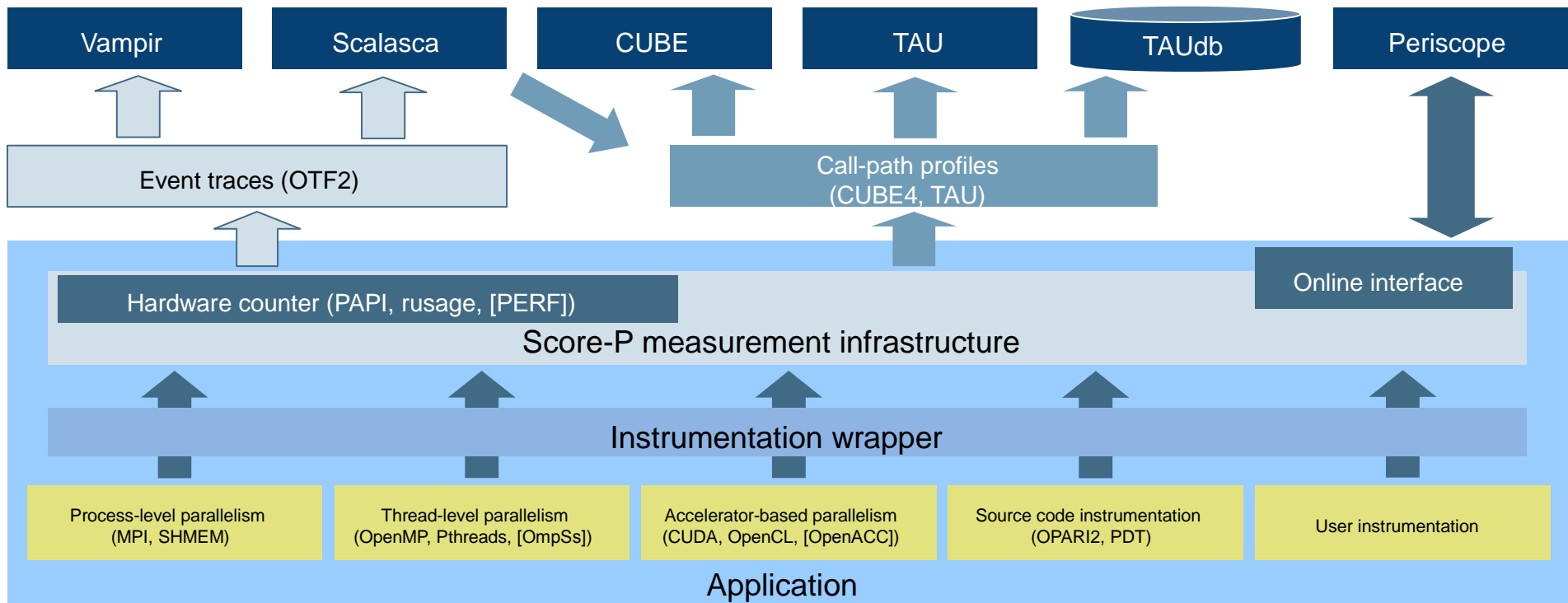
# Scalability to over 1.8M threads (MPI+OMP)



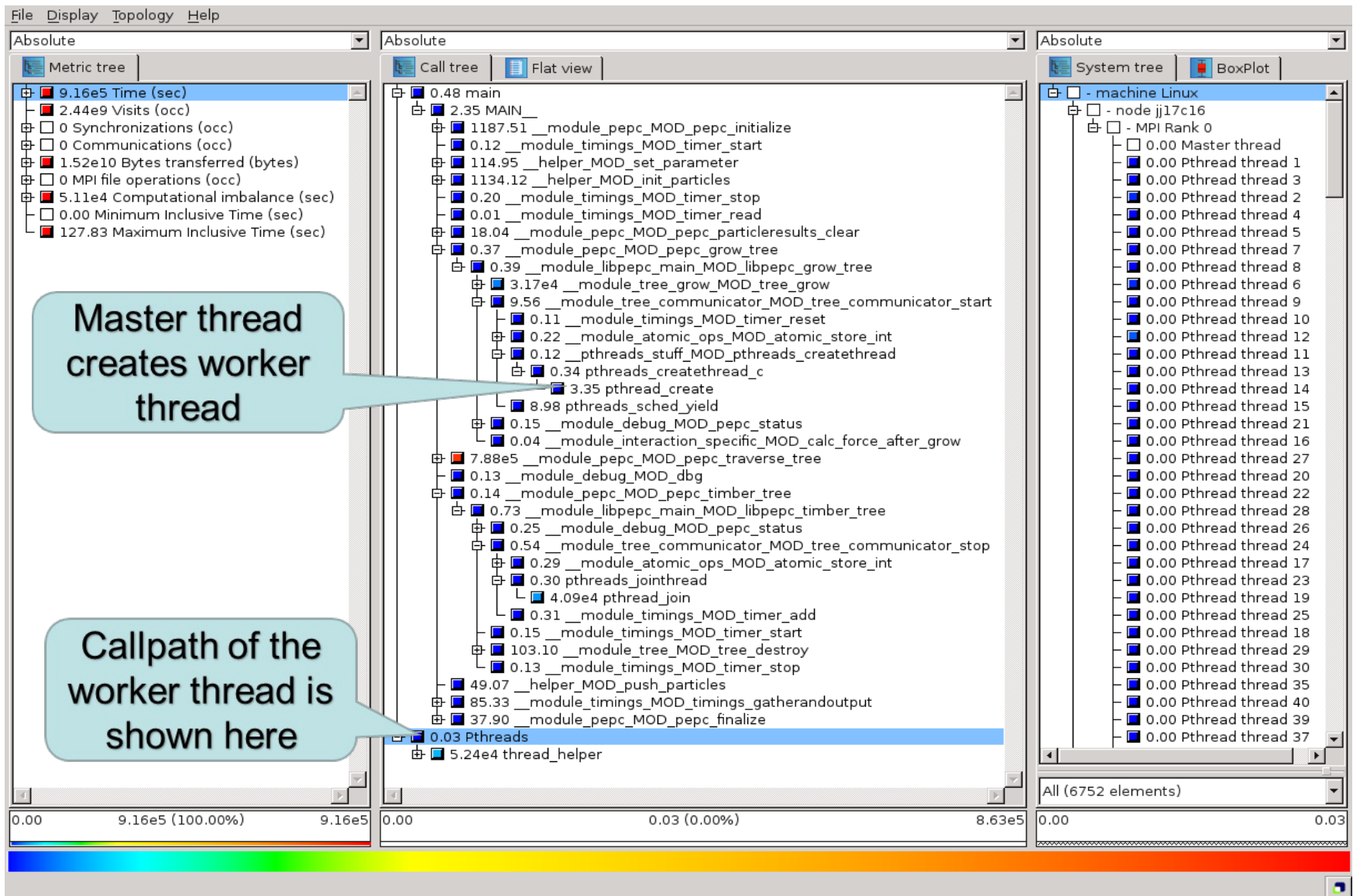
# Scalability to over 1.8M threads (MPI+OMP)



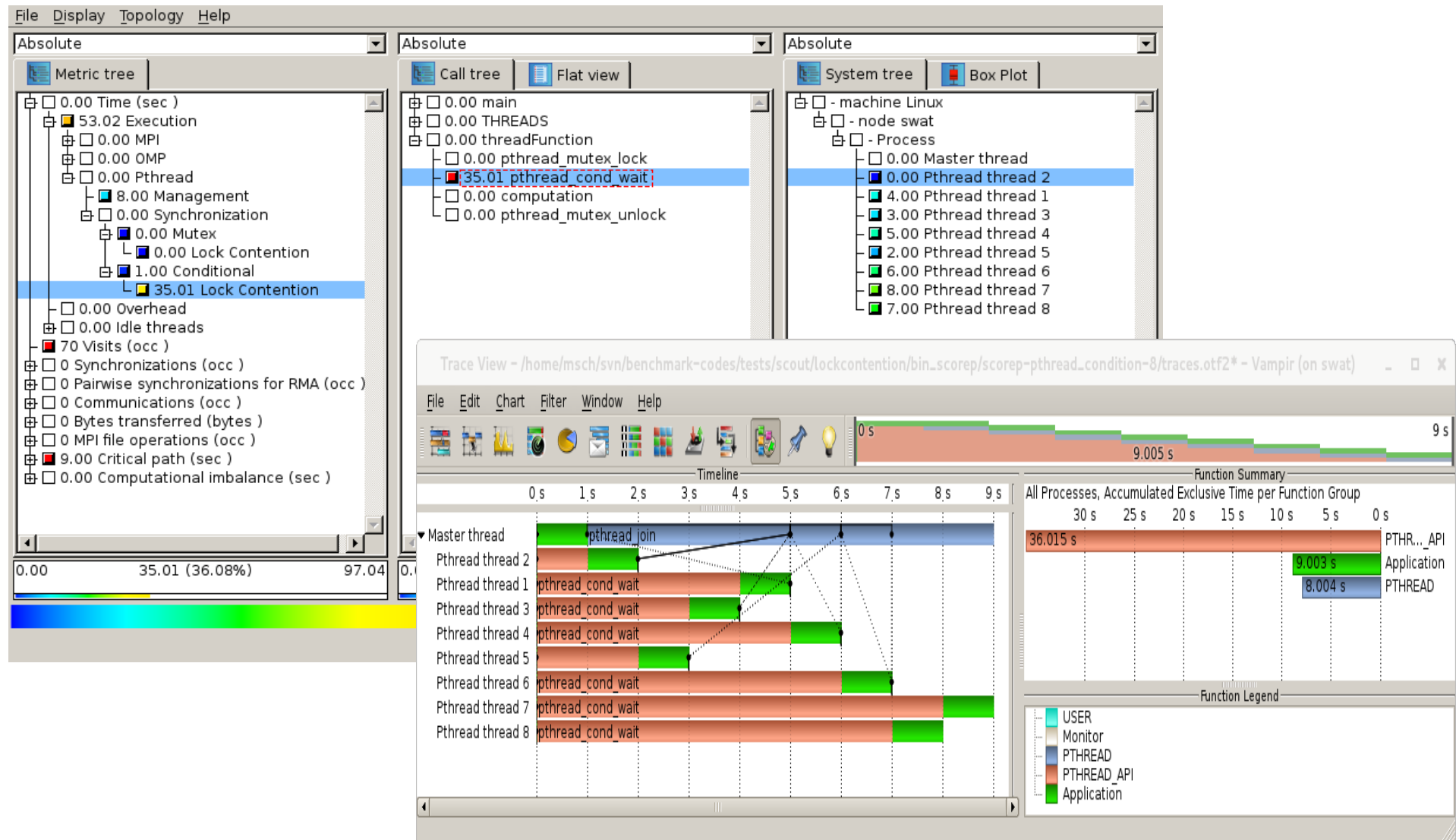
# Score-P architecture overview



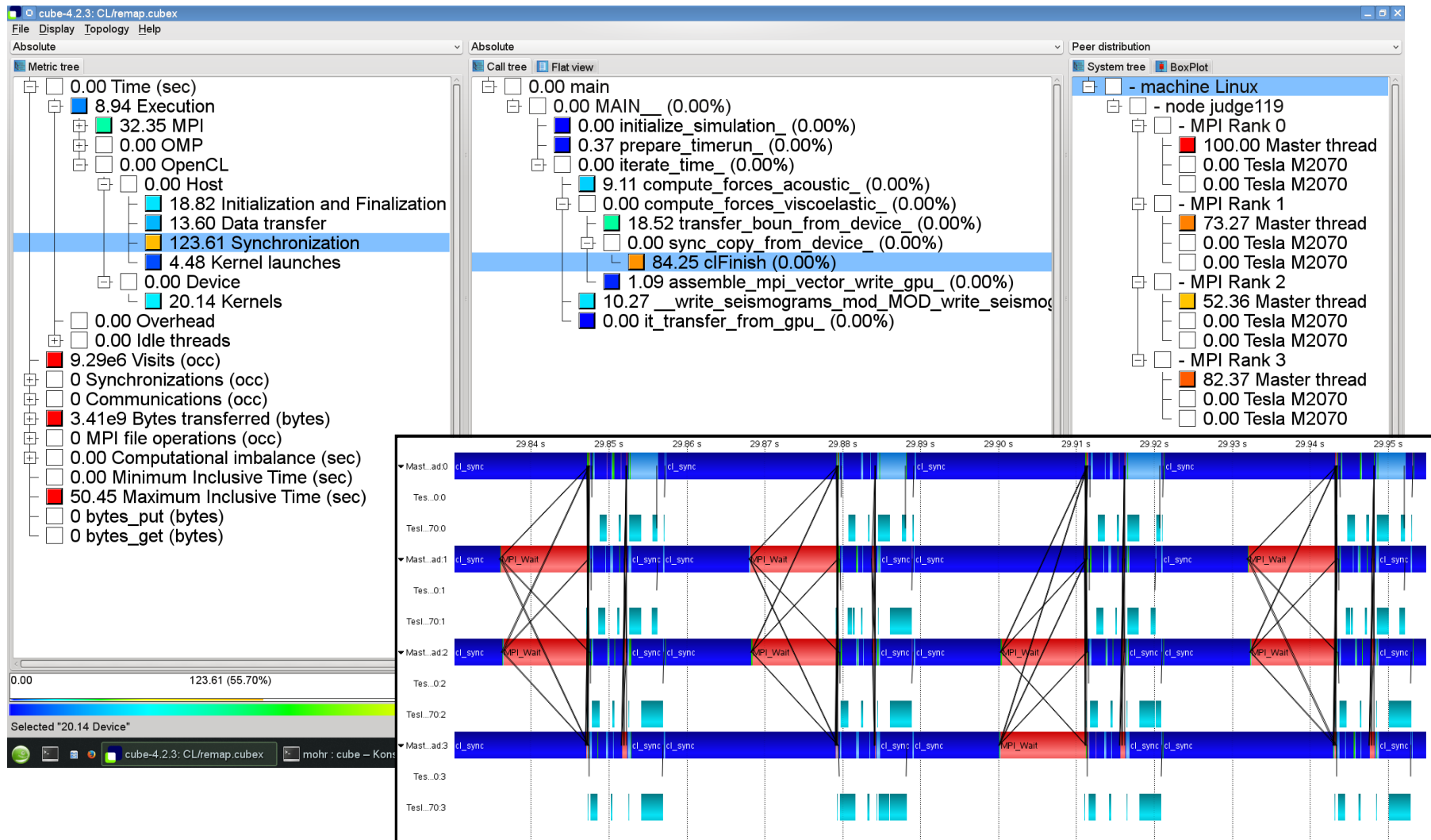
# PEPC 32 MPI ranks with 13 pthreads per rank on Juropa



# Lock contention (pthreads)



# MPI+OpenCL: SPECFEM3D\_GLOBE



# Tool integration prototypes

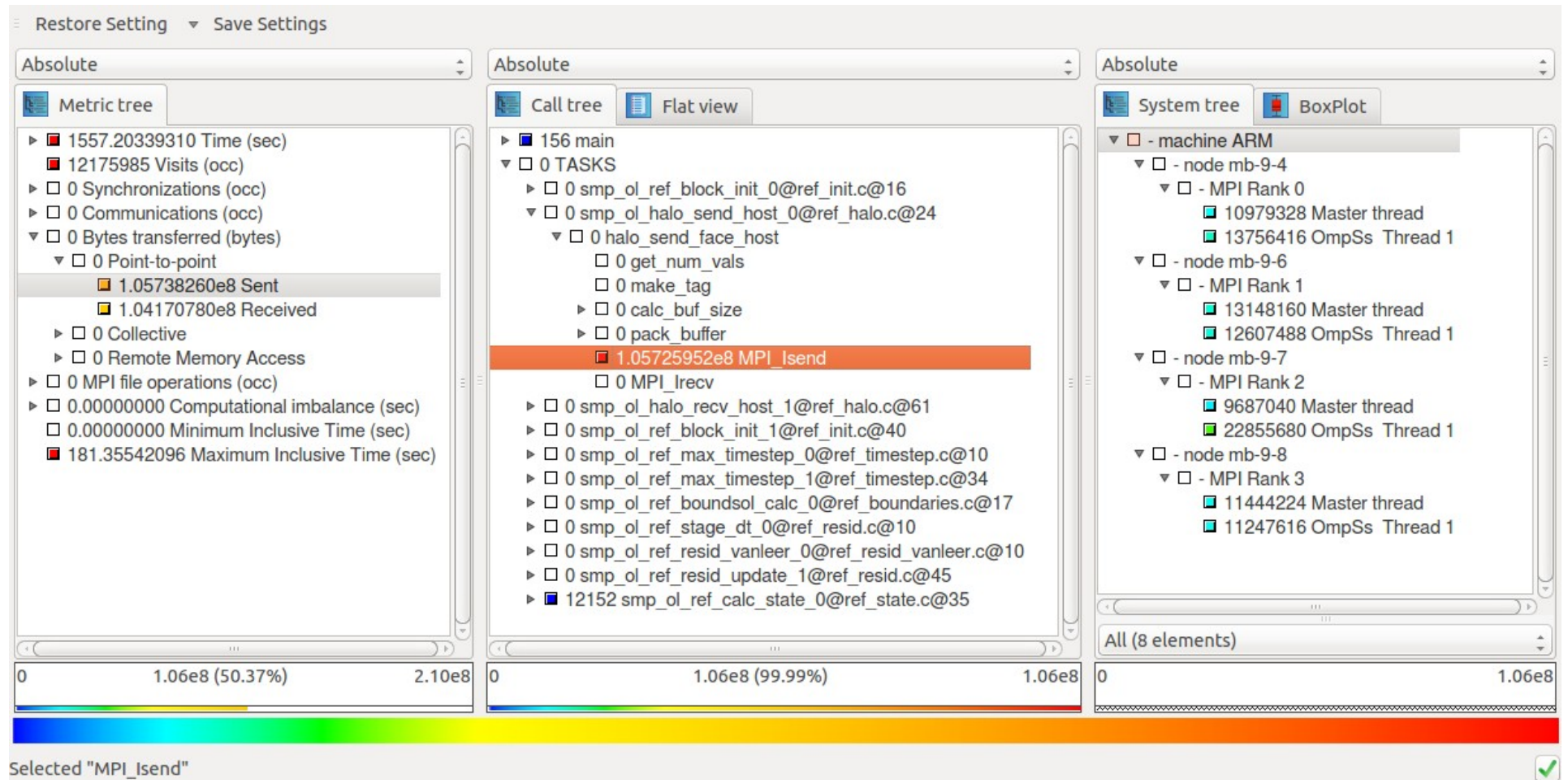
## Analysis

- Scalasca can direct Paraver or Vampir to show sections of event trace with severest instances of communication/synchronisation inefficiencies

## Measurement

- OmpSs runtime can generate events for Extrae/Score-P measurements of tasks
- XMP compiler can include instrumentation for Scalasca measurements of parallel regions

# Score-P MPI+OmpSs profile



# XcalableMP-Scalasca instrumentation

\*Masahiro NAKAO, Center for Computational Sciences, University of Tsukuba

XMP code

```
#pragma xmp loop profile  
for(i = 0; i < 100; i++)  
    array[i] = func(i);
```

```
#pragma xmp bcast (a)
```

xmpcc -profile

profile (specific) pragma-blocks

```
EPIK_USER_REG(r_loop#, "loop#");
```

```
EPIK_USER_START(r_loop#);
```

```
#pragma xmp loop profile  
for(i = 0; i < 100; i++)  
    array[i] = func(i);
```

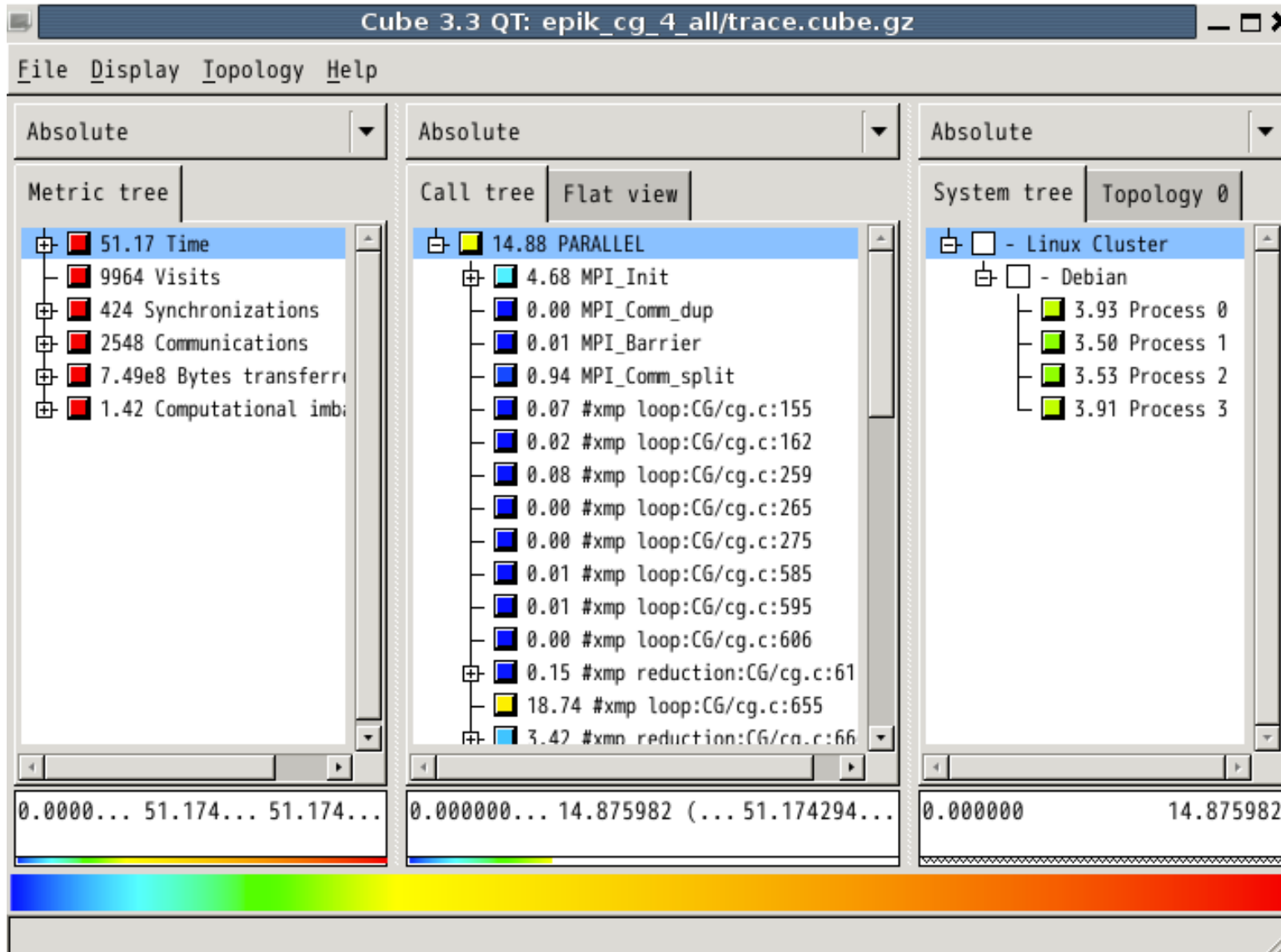
```
EPIK_USER_END(r_loop#);
```

```
EPIK_GEN_OFF();
```

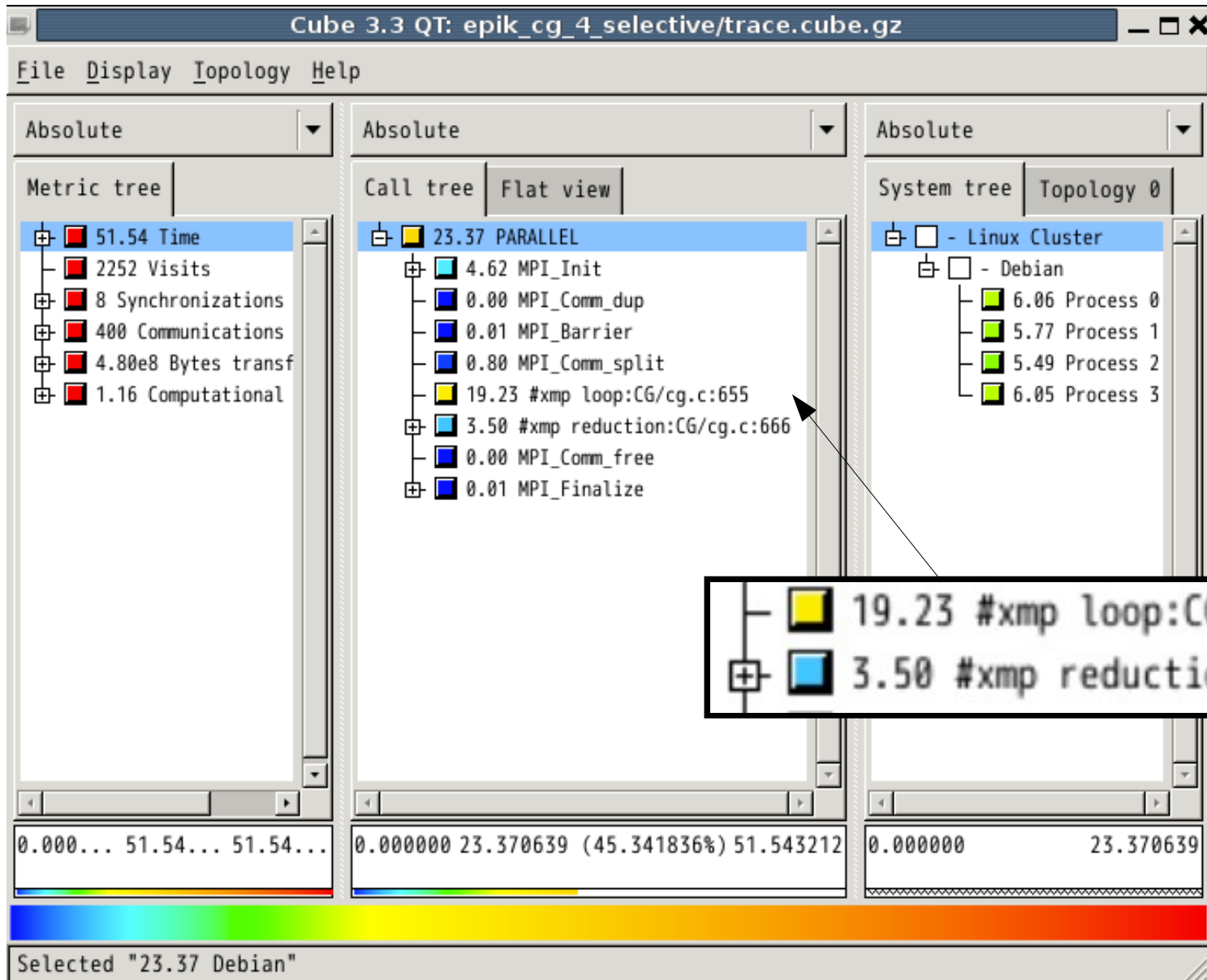
```
#pragma xmp bcast (a)
```

```
EPIK_GEN_ON();
```

## XMP -allprofile



## XMP -profile



## Score-P on-going developments

Support for MPI\_THREAD\_MULTIPLE

Support for MPI-3

Support for OMPT

Support for OpenACC

Support for additional architectures & platforms

Support for binary instrumentation

Support for I/O and memory events

Support for orphan (parentless) threads

Support for sampling and callstack walking

...