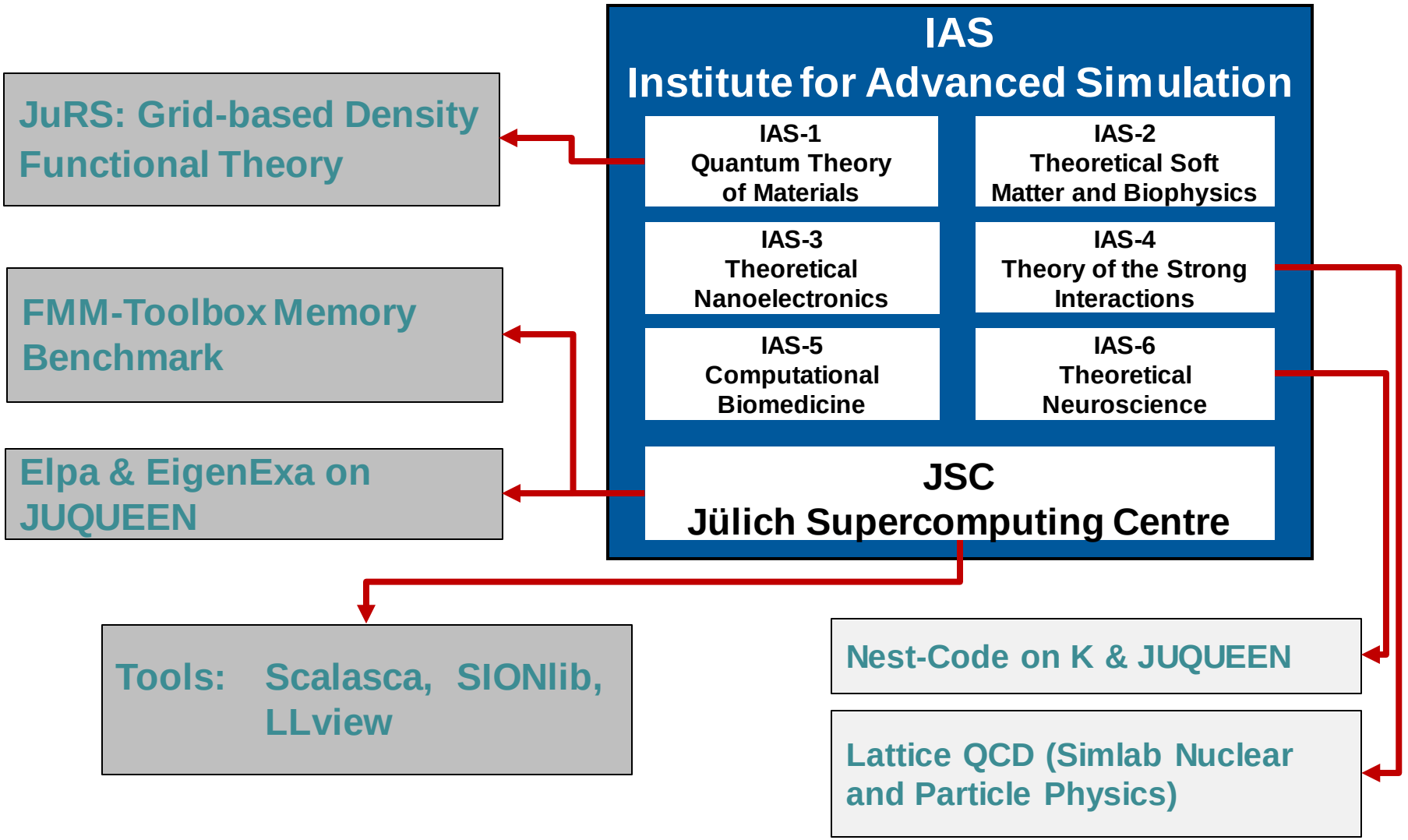
JÜLICH
FORSCHUNGSZENTRUM

# Status of FZJ-IAS activities on the K computer

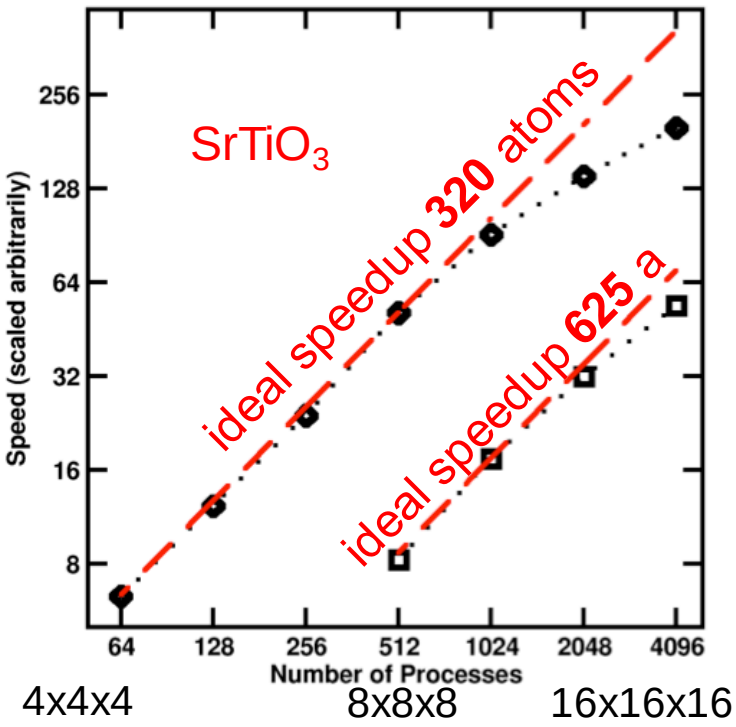
Wolfgang Frings
W.Frings@fz-juelich.de
Jülich Supercomputing Centre

Riken, Japan, 8. September 2014

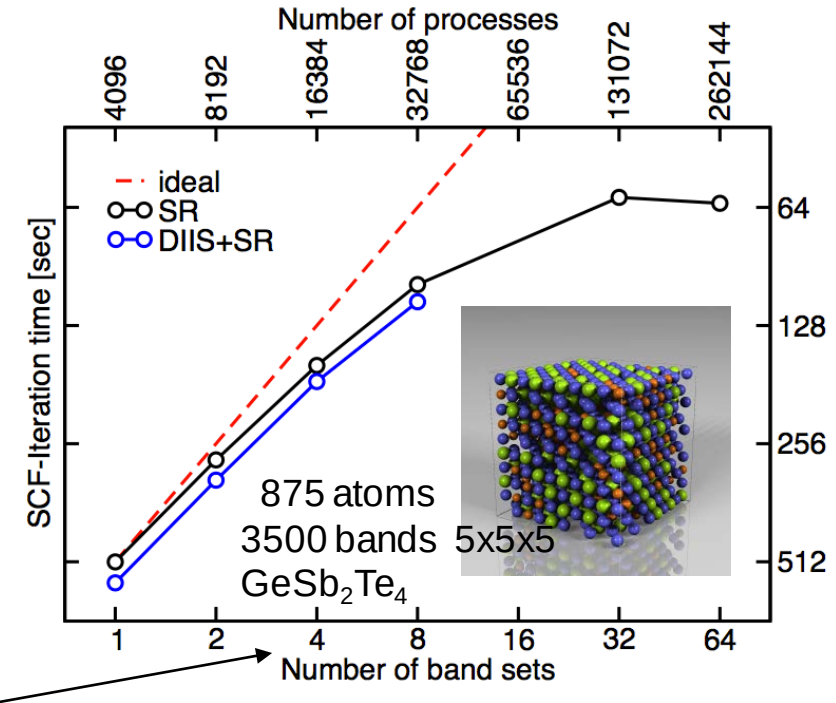
# FZJ-IAS & Activities on K and JUQUEEN



# Grid-based Density Functional Theory




+ Band Parallelization





→ Atoms with PAW accuracy, random alloy
10x10x10 $GeSb_2Te_4$ **7000 atoms**
(Projector Augmented Wave method)
running on 131,072 cores

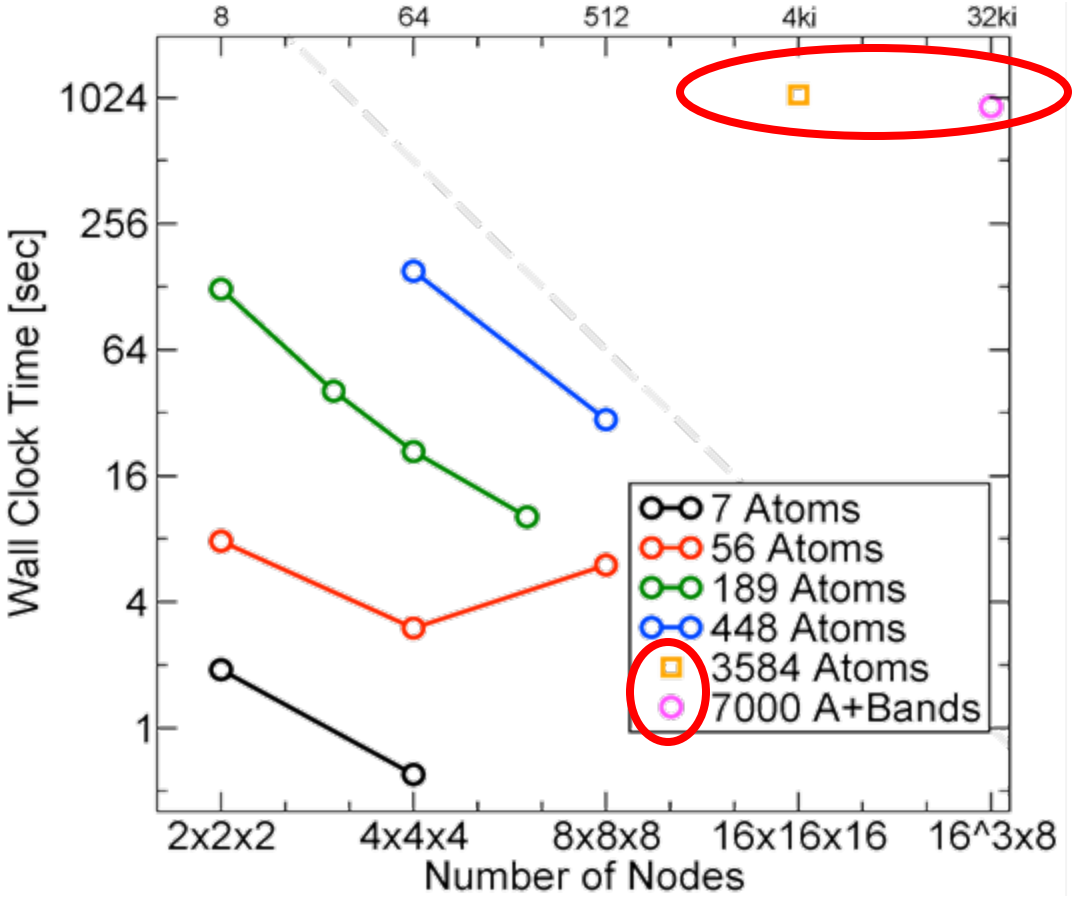
Contact: **Paul Baumeister, JSC, p.baumeister@fz-juelich.de**

# juRS on the K-Computer

- No problems porting the Fortran95 juRS-sources:

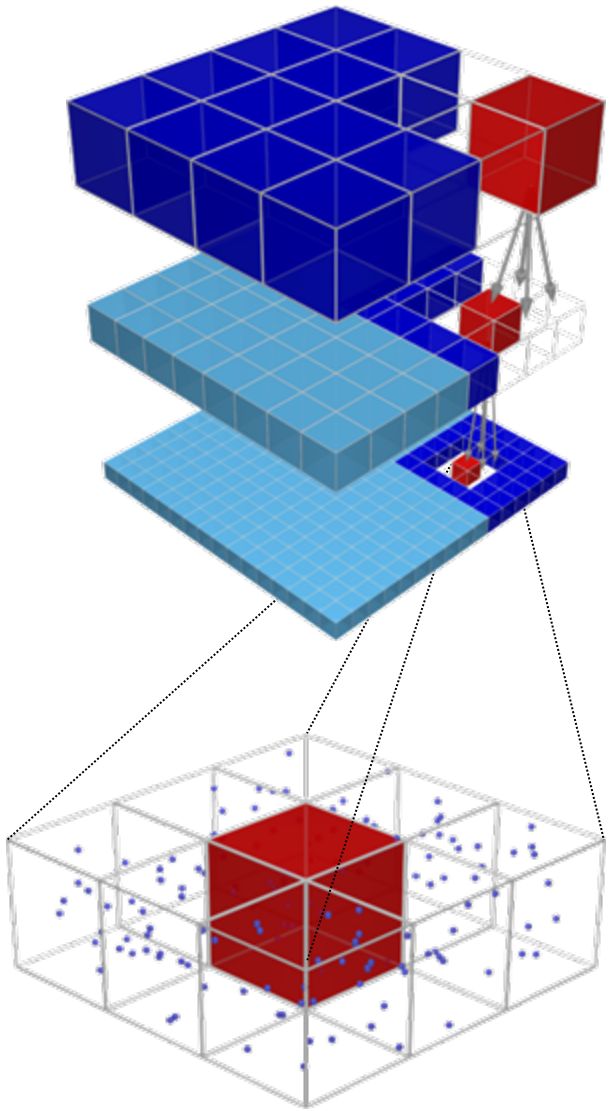
```
mpifrtpx -Ad -Kopenmp -Kfast -Koptmsg=2 -Kilfunc -Kfp_contract
```

- linking with `-SSL2 -SCALAPACK -Kopenmp`



# FMM Toolbox on K: Overview

- Fast Multipole Method (**FMM**):
  - Computing Coulomb (Long-range)-interaction in O(N) instead of classical pairwise O(N^2)
  - Used for molecular dynamics simulations e.g. Gromacs MD Code
- DFG-funded project *GROMEX*
  - → extend Gromacs to allow FMM as backend Coulomb solver
  - → extend FMM to allow for dynamic protonation within Gromacs
  - → Redesign expensive compute kernels in C++
  - → Develop performance portable FMM implementation for MD simulations
- First steps:
  - Test basic features of todays different HPC platforms (incl. K computer)



Contact: **Andreas Beckmann, Ivo Kabadshow, JSC, i.kabadshow@fz-juelich.de**

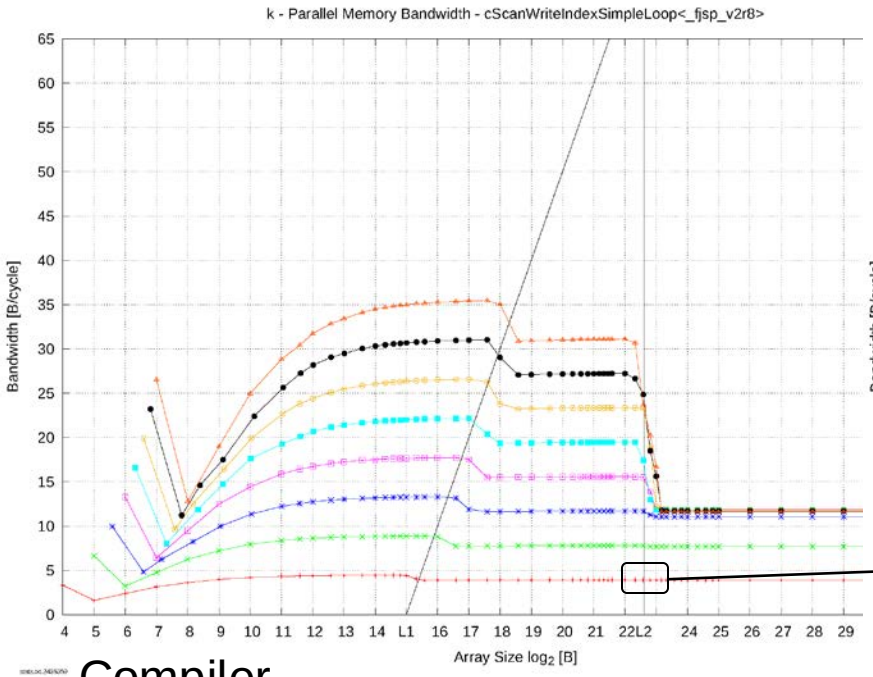
# FMM Toolbox on K: Memory Bandwidth

FMM is compute bound / memory bandwidth bound

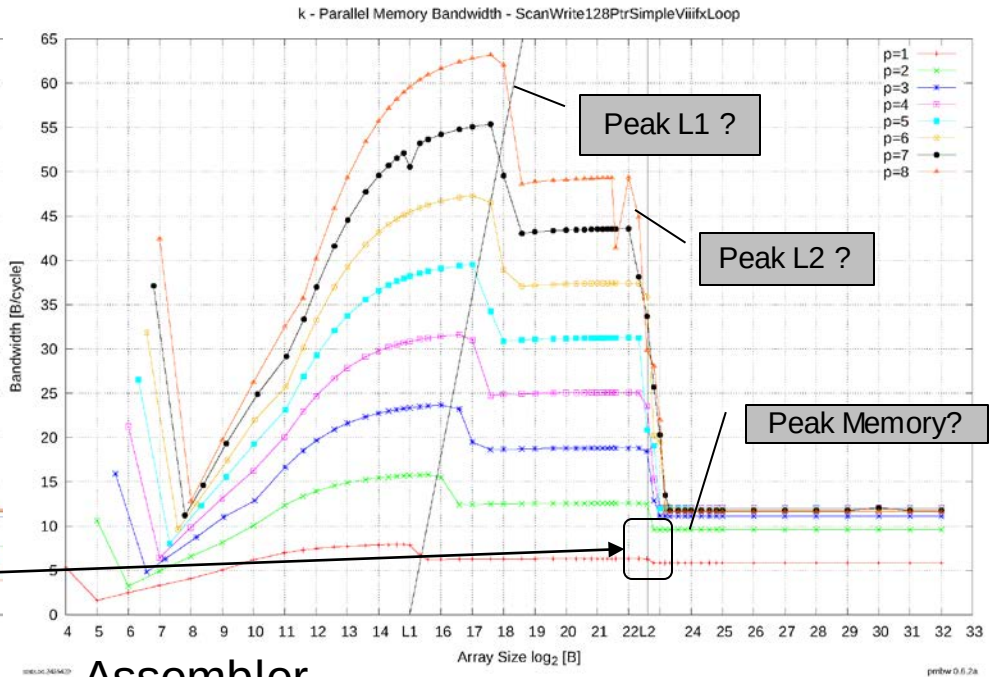
→ What is the maximum write bandwidth achievable on a single node?

Benchmark **pmbw** (parallel memory bandwidth)

- 128bit floating point stores (double x 2); linear access per thread
- separate memory per thread; pure write, no computation! single node tests only

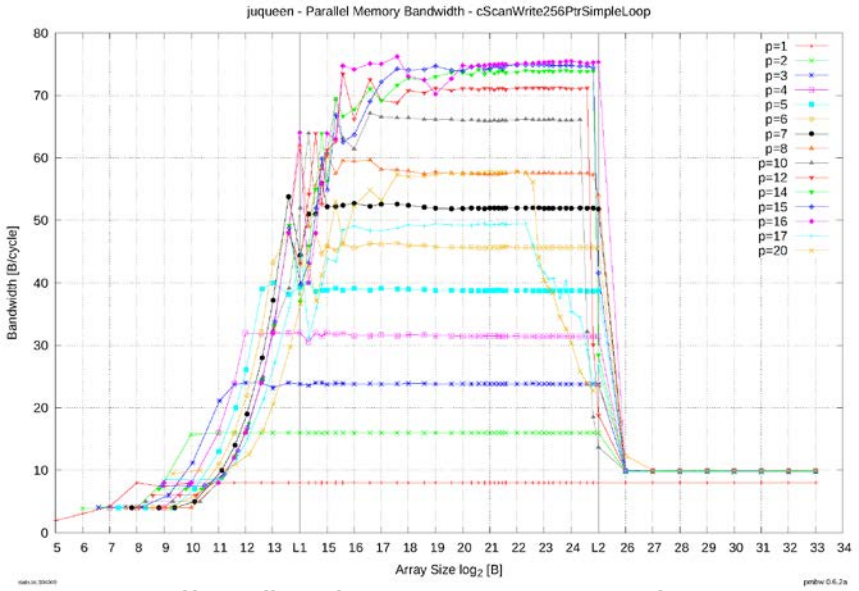



Compiler

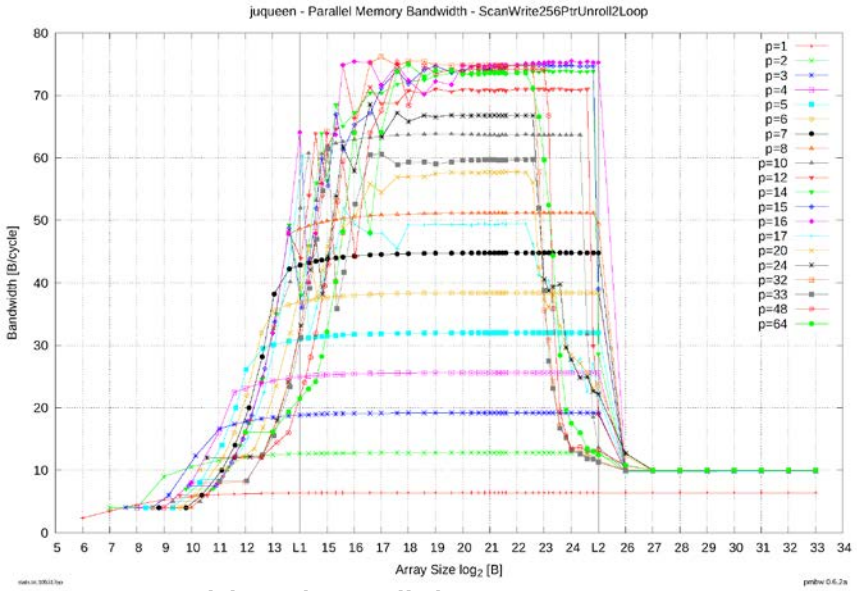



Assembler

# Benchmark pmbw on Juqueen




Compiler (bgclang, LLVM+QPX)




Assembler (Unroll2)

# Test of Elpa and EigenExa on JUQUEEN

- **ELPA** (EigensoLver for Petaflop Architectures)

  BMBF funded project (RZG Garching, Uni Wuppertal, …)

  - Fortran 95, add-on to ScaLAPACK, OpenMP
  - ELPA 1: one step reduction and back transformation
  - ELPA 2: two step reduction to banded form, back transf. on BlueGene/Q

  → ELPA-2 especially developed for part of the eigenspectrum

- **EigenExa Library**

  Toshiyuki Imamura, RIKEN Advanced Institute for Computational Science

  - Fortran 95, OpenMP for reduction and back transformation available
  - eigen s: one step reduction and back transformation
  - eigen sx: two step reduction to pentadiagonal form

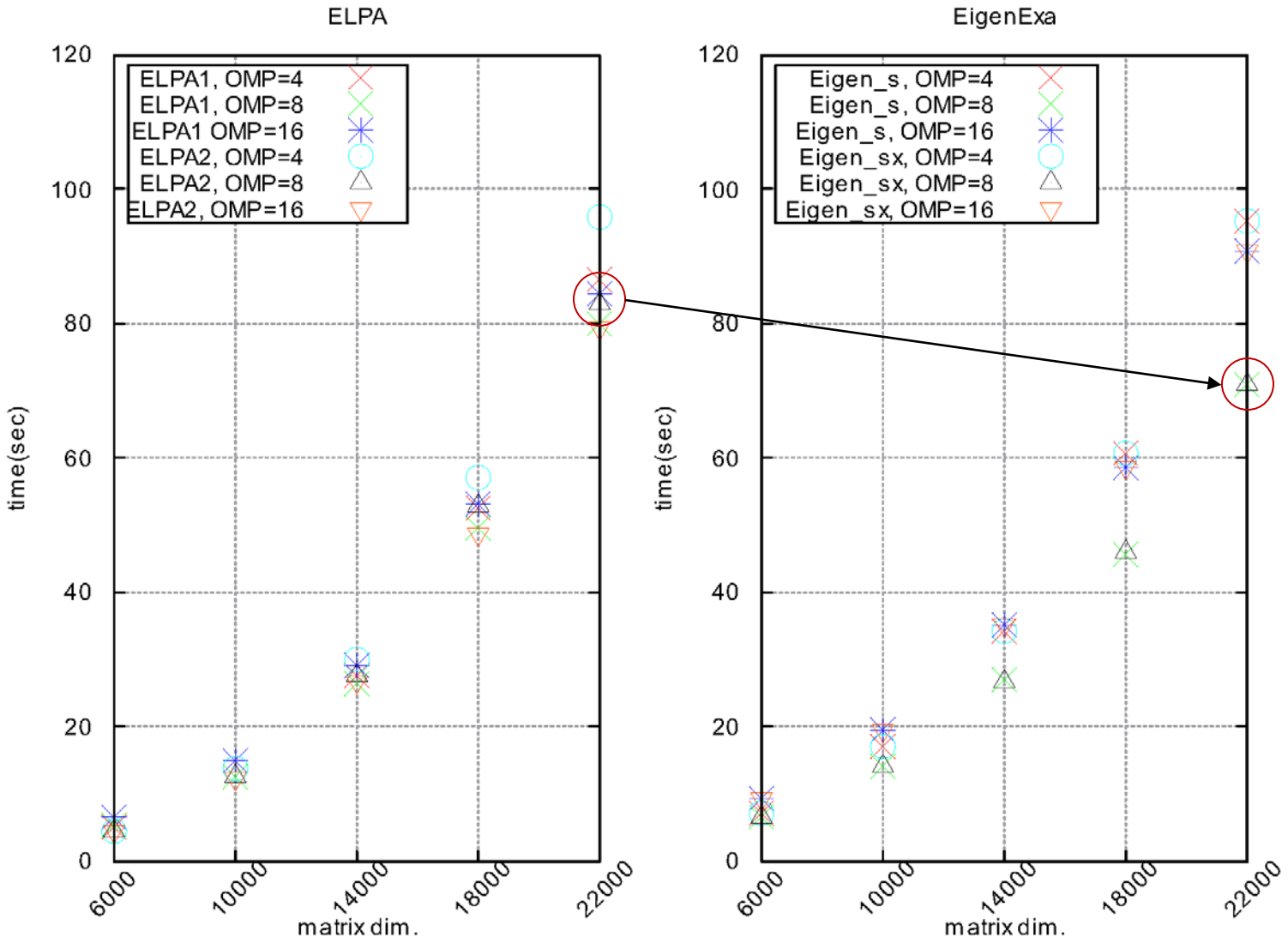
  → EigenExa developed to compute full eigenspectrum

Contact: **I. Gutheil, JSC, i.gutheil@fz-juelich.de**

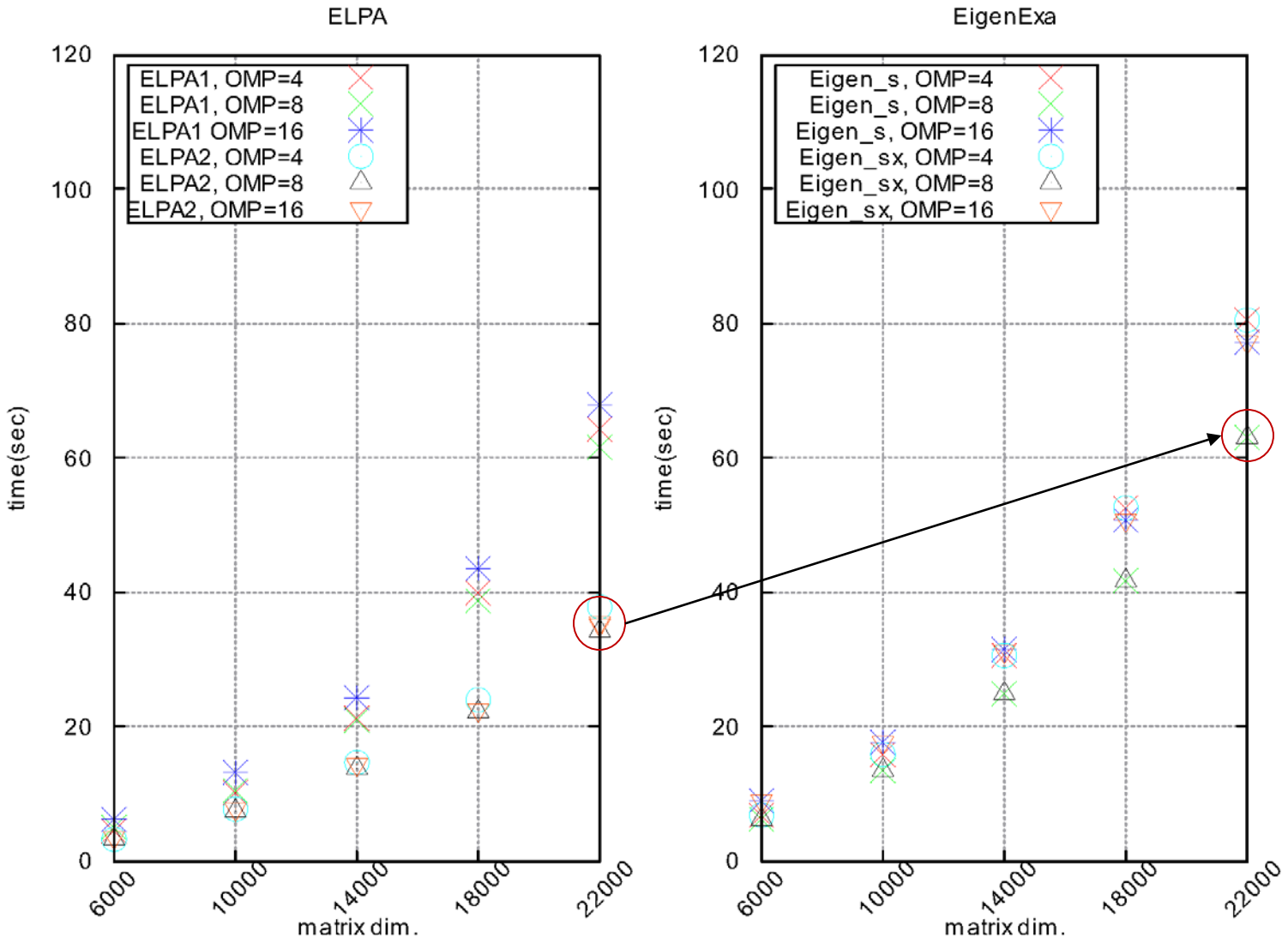
# Test Environment on JUQUEEN

- **Partition**: 1 node card, 512 cores, hybrid with 4 and 8 MPI processes per compute card
- **Matrix/Blocking**: N=6000 to 22000 by steps of 4000
  - ELPA: block size nb = 16 (ELPA 2 bandwidth)
  - EigenExa: nb = 48 (reduction), nb = 128 (back transformation), no accuracy improvement for eigenvalues
- **Test program ELPA**:
  - Generates diagonal matrix with random eigenvalues
  - Full matrix constructed via random Householder transformation
  - Deviation from given eigenvalues, residual, orthogonality tested
- **Test program EigenExa**:
  - Test program delivered with EigenExa, very similar to ELPA test program for pure MPI and hybrid version
  - Full eigenspectrum and 5 % of the eigenspectrum

# JUQUEEN Node card, 4 MPI processes per compute card, full eigenspectrum



# JUQUEEN Node card, 4 MPI processes per compute card, 5 % of the eigenspectrum



# scalasca on Fujitsu FX10 & K

JÜLICH FORSCHUNGSZENTRUM

German Research School for Simulation Sciences

**Sc**alable performance **a**nalysis of **la**rge-**sc**ale parallel **a**pplications

- portable toolset for scalable performance measurement & analysis of MPI, OpenMP & hybrid OpenMP+MPI parallel applications
- *supporting most popular HPC computer systems*
  *Cray XT/XE/XK, IBM BlueGene L/P/Q, IBM SP & blade clusters, Fujitsu FX10 & K computer, NEC SX, SGI Altix, Linux cluster®, ...*
- Integrated instrumentation, measurement & analysis toolset
  - → Customizable automatic/manual instrumentation
  - → Runtime summarization (*aka* profiling)
  - → Automatic event trace analysis
- Availability, Sources, documentation & publications:
  - 3-clause New BSD open-source license
    http://www.scalasca.org, scalasca@fz-juelich.de

Contact: **Brian Wylie, JSC, b.wylie@fz-juelich.de**

# Scalasca Status on K

- **Scalasca-1.4**
  - ported to K computer & Fujitsu FX10 by Tomotake Nakamura (released March 2013)
  - included PAPI hardware counters and 1/2/3D FJMPI topology
  - 6D-Tofu network topology added March 2014

  `/opt/aics/UNITE/packages/scalasca/1.4.4`
- **Scalasca-2.1**
  - based on Score-P 1.3 (released August 2014)
  - newly-developed instrumentation & measurement infrastructure
  - now includes support for FX10 & K computer
    - *no topologies, problem with OpenMP/C++*

  `/opt/aics/UNITE/packages/scalasca/2.1` (1st release candidate)
  - Used for **14th VI-HPS Tuning Workshop** (RIKEN AICS, 25-27 Mar 2014)

# Scalasca: Analysis of ABySS-P execution on K computer

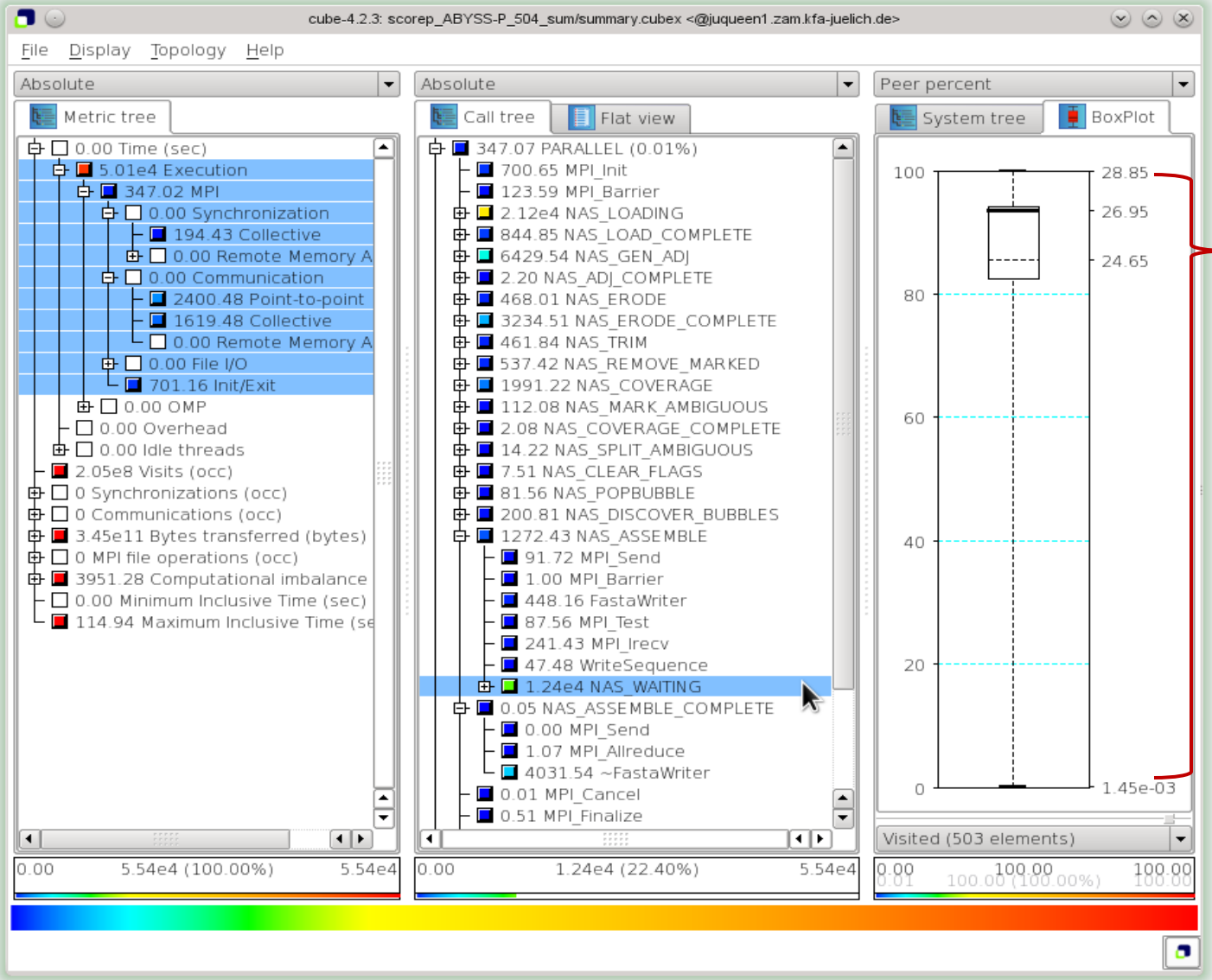
Parallel genome sequence assembly using MPI (C++)

- Performance analysis on K computer by Itaru Kitayama (RIKEN)
  - *Limited scalability to around 768 compute nodes (1ppn)*

Status

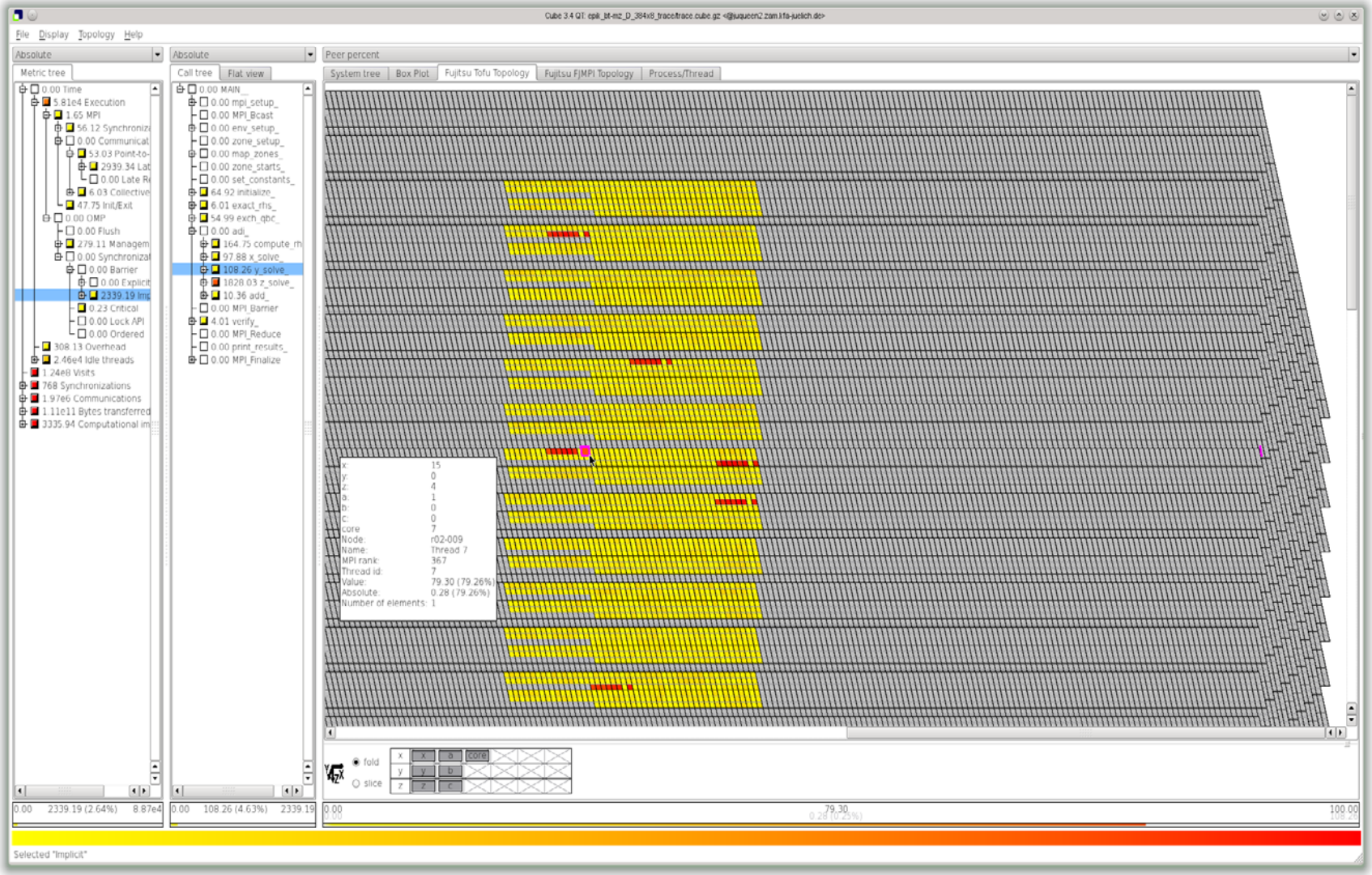
- Manually annotated execution stages and file I/O operations
  - *provide context for MPI communication & synchronization*
- Summary experiments identify inefficient phases and causes
  - *NAS_LOADING read & distribute input data*
  - *NAS_POP/DISCOVER_BUBBLES serialized file writing*
  - *NAS_ASSEMBLE write sequence data load imbalance*
    - busy wait idling in NAS_WAITING

# Scalasca: Analysis of ABySS-P




load imbalance in NAS_ASSEMBLE

# Scalasca: BT-MZ.D on K, 384x8 *y_solve* OMP barrier imbalance



# Scalasca2 / Score-P development plans (FX10/K)

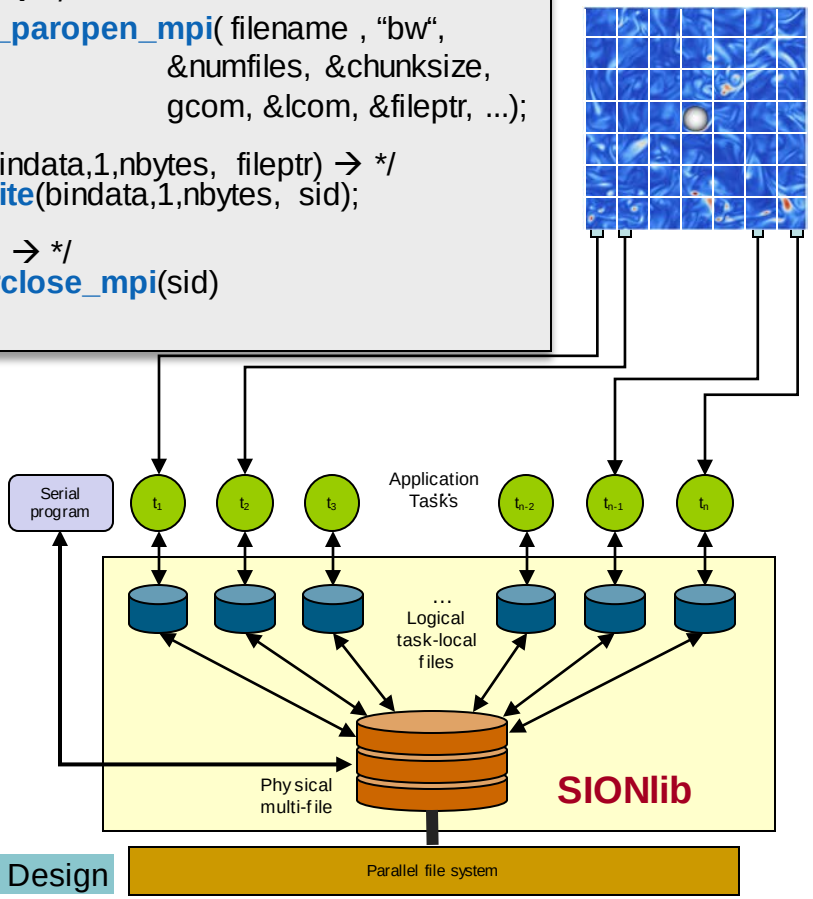
- Use of SIONlib for handling trace files
  - to avoid bottlenecks with multiple files per thread
- Record machine and application topologies
  - FJMPI, 6D-Tofu, etc. (as with Scalasca-1)
- Incorporate metrics from Tofu network counters?
- POSIX thread support?

# SIONlib: Overview

- Scalable Massively Parallel I/O to Task-Local Files
- Extension of I/O-API (ANSI C or POSIX)
- C, C++ and Fortran bindings, implementation language C
- Parallel Interfaces: MPI, OpenMP, Hybrid, Generic
- Current versions: 1.5.1
- Scalability on JUQUEEN: > 100 GB/s, 1.8 mio tasks
- Open source license: http://www.fz-juelich.de/jsc/sionlib

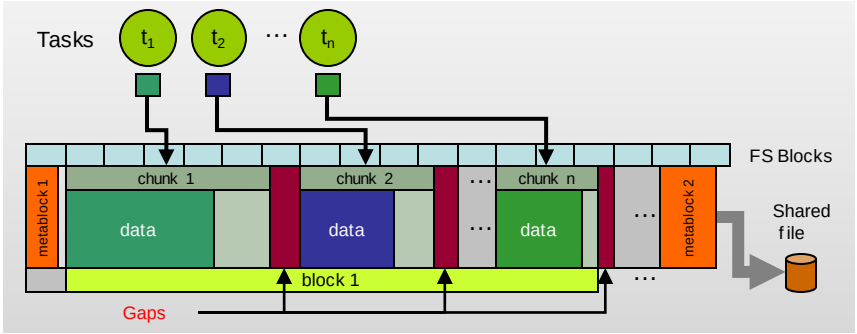
→ SIONlib-API

```
/* fopen() → */
sid=sion_paropen_mpi( filename , "bw",
                      &numfiles, &chunksize,
                      gcom, &lcom, &fileptr, ...);

/* fwrite(bindata,1,nbytes, fileptr) → */
sion_fwrite(bindata,1,nbytes, sid);

/* fclose() → */
sion_parclose_mpi(sid)
```




→ Design

→File-Format



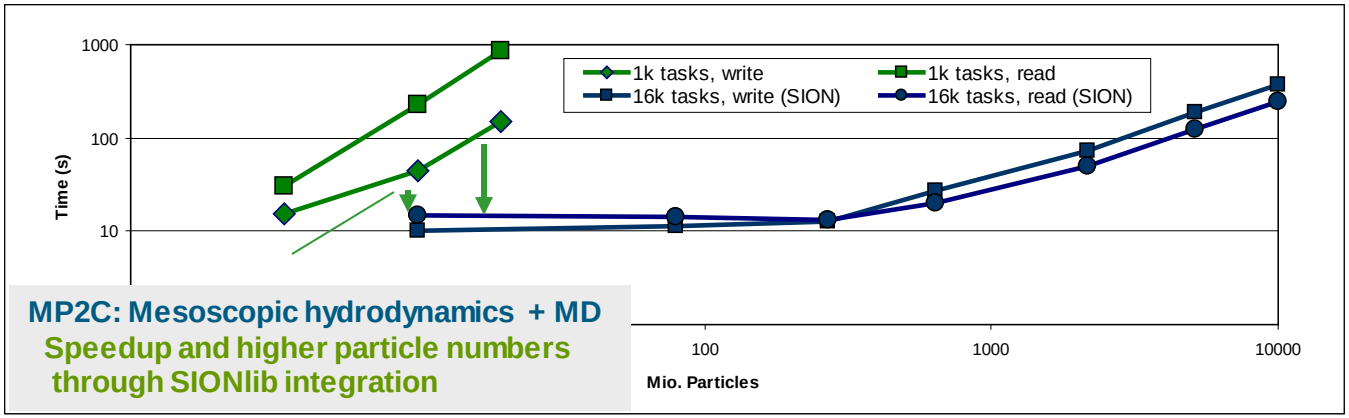

Contact: **Wolfgang Frings, JSC**
**w.frings@fz-juelich.de**

# SIONlib: Applications & Use Cases

- **Applications → Checkpointing, Restart Files**

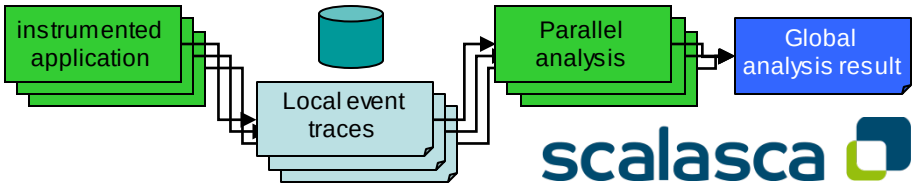
**DUNE-ISTL** (Multigrid solver, Univ. Heidelberg) **ITM** (Fusion-community),
**LBM** (Fluid flow/mass transport, Univ. Marburg), **PSC** (particle-in-cell code),
**OSIRIS** (Fully-explicit particle-in-cell code), **PEPC** (Pretty Efficient Parallel C. Solver)
**Profasi**: (Protein folding and aggr. simulator) **NEST** (Human Brain Simulation)

**MP2C**:




- **Tools/Projects**

**Scalasca:** Performance Analysis

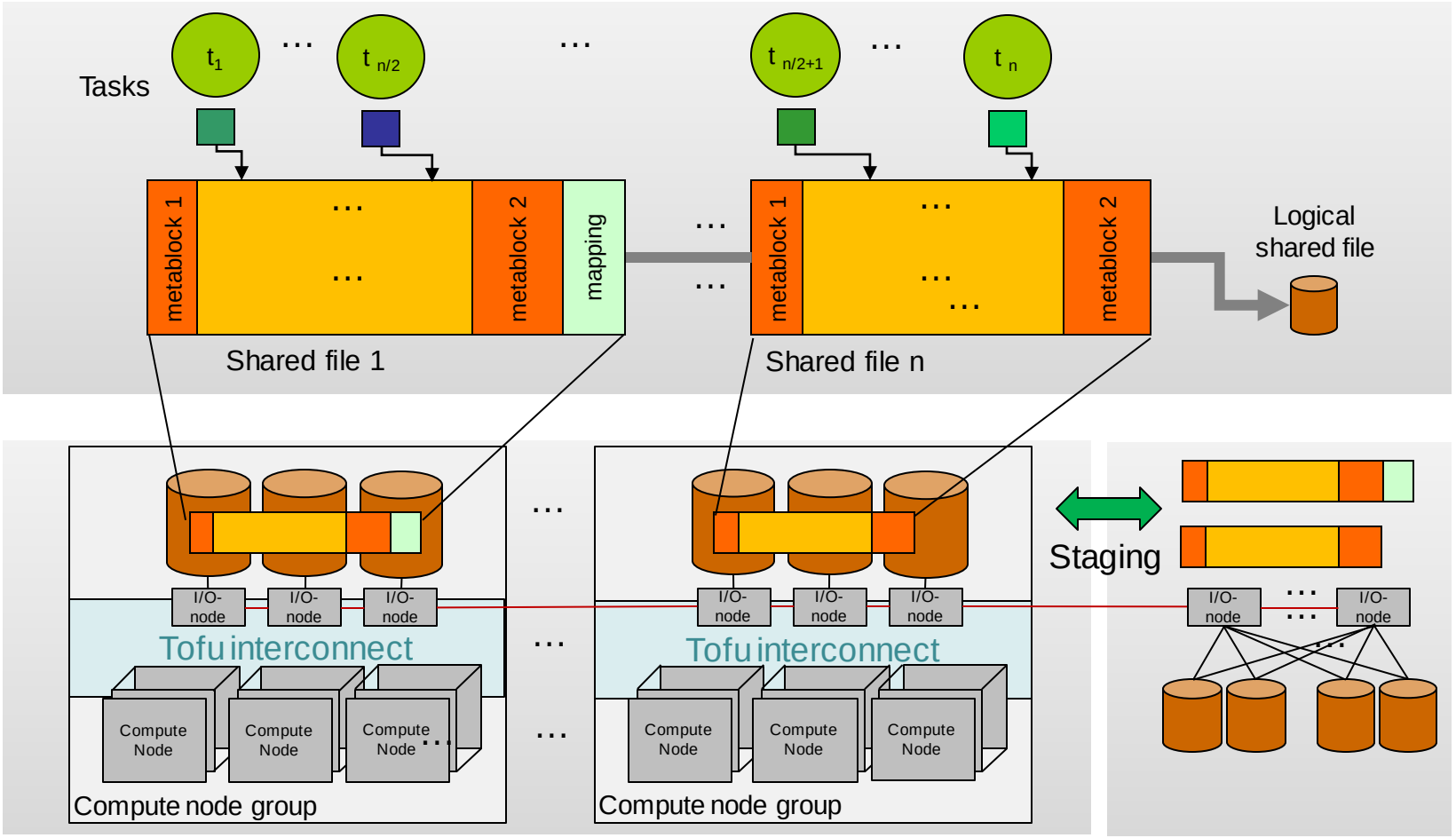



**Score-P:** Scalable Performance Measurement Infrastructure for Parallel Codes

**DEEP-ER:** Adaption to new platform and parallelization paradigm

# SIONlib: Strategy for K

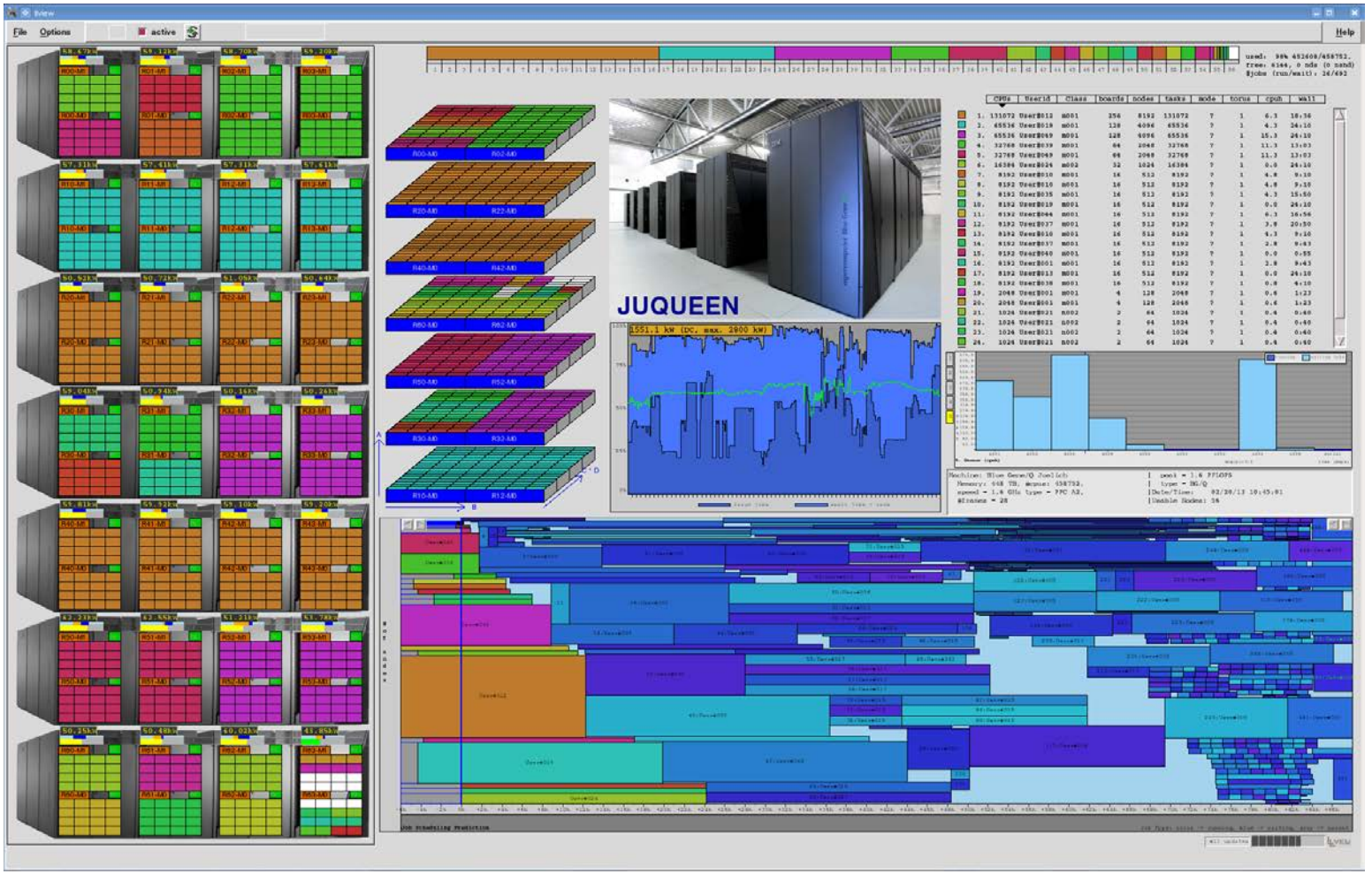
- Multiple physical files for SIONlib shared file




- Files on K Local Storage

K Global Storage

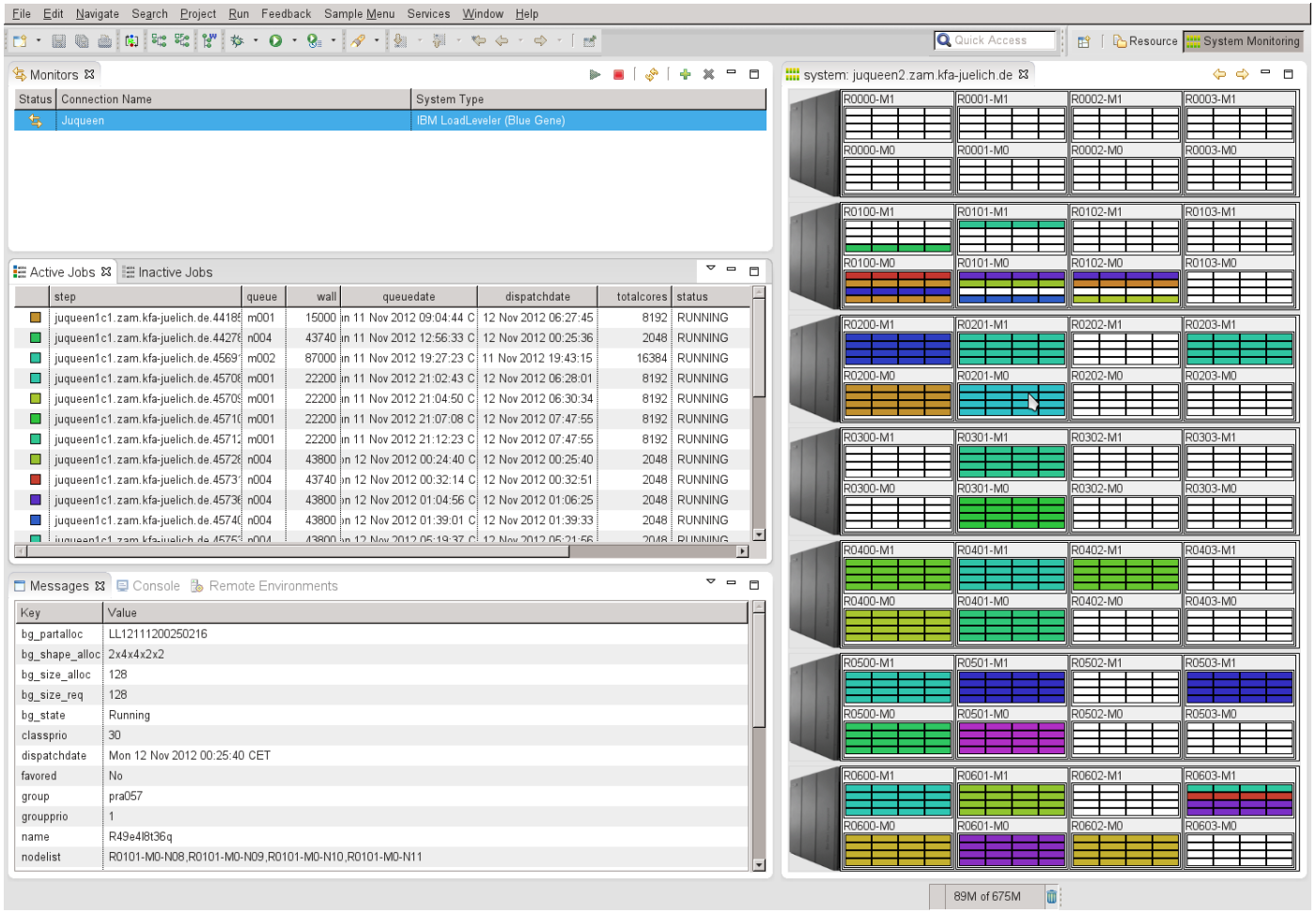
# Batch System Monitoring: LLview




Contact: llview.jsc@fz-juelich.de
WWW: http://www.fz-juelich.de/jsc/llview

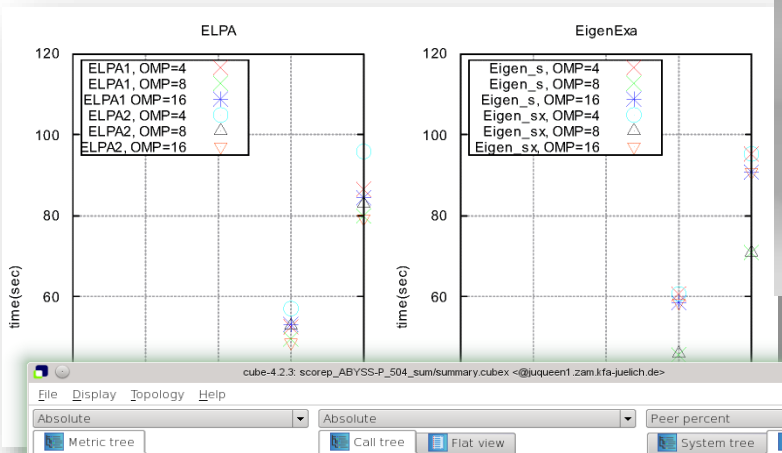
# Batch System Monitoring: Eclipse/PTP

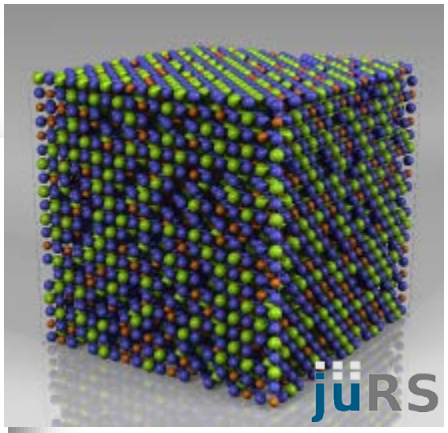
Main components of LLview in **PTP's monitoring perspective**

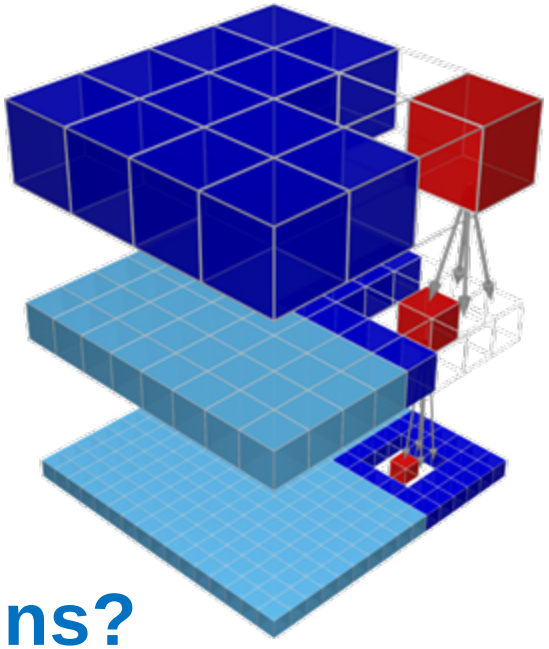


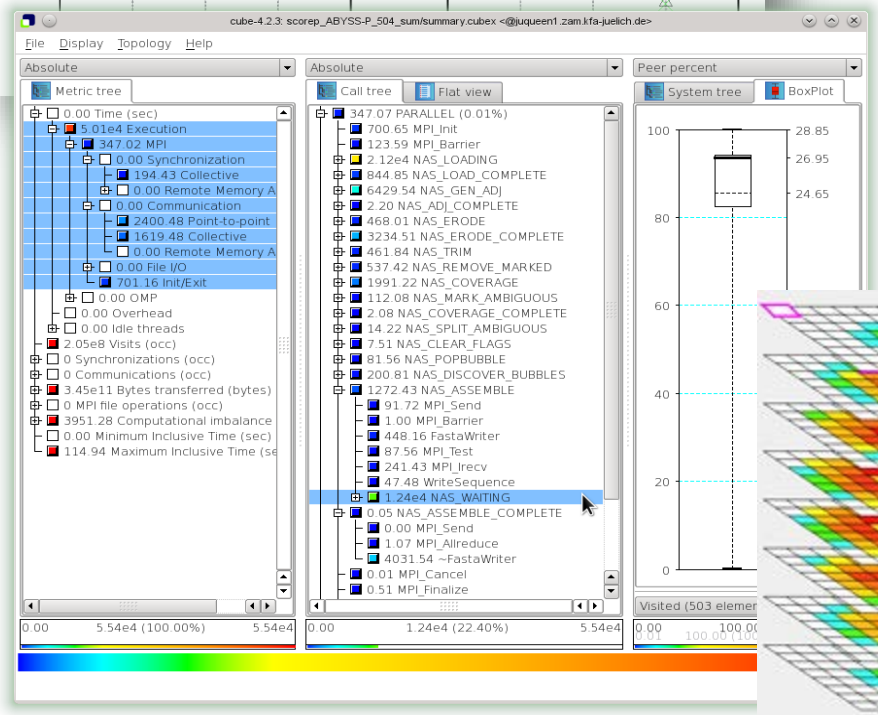

WWW: https://www.eclipse.org/ptp

# Thank You!






## Questions?