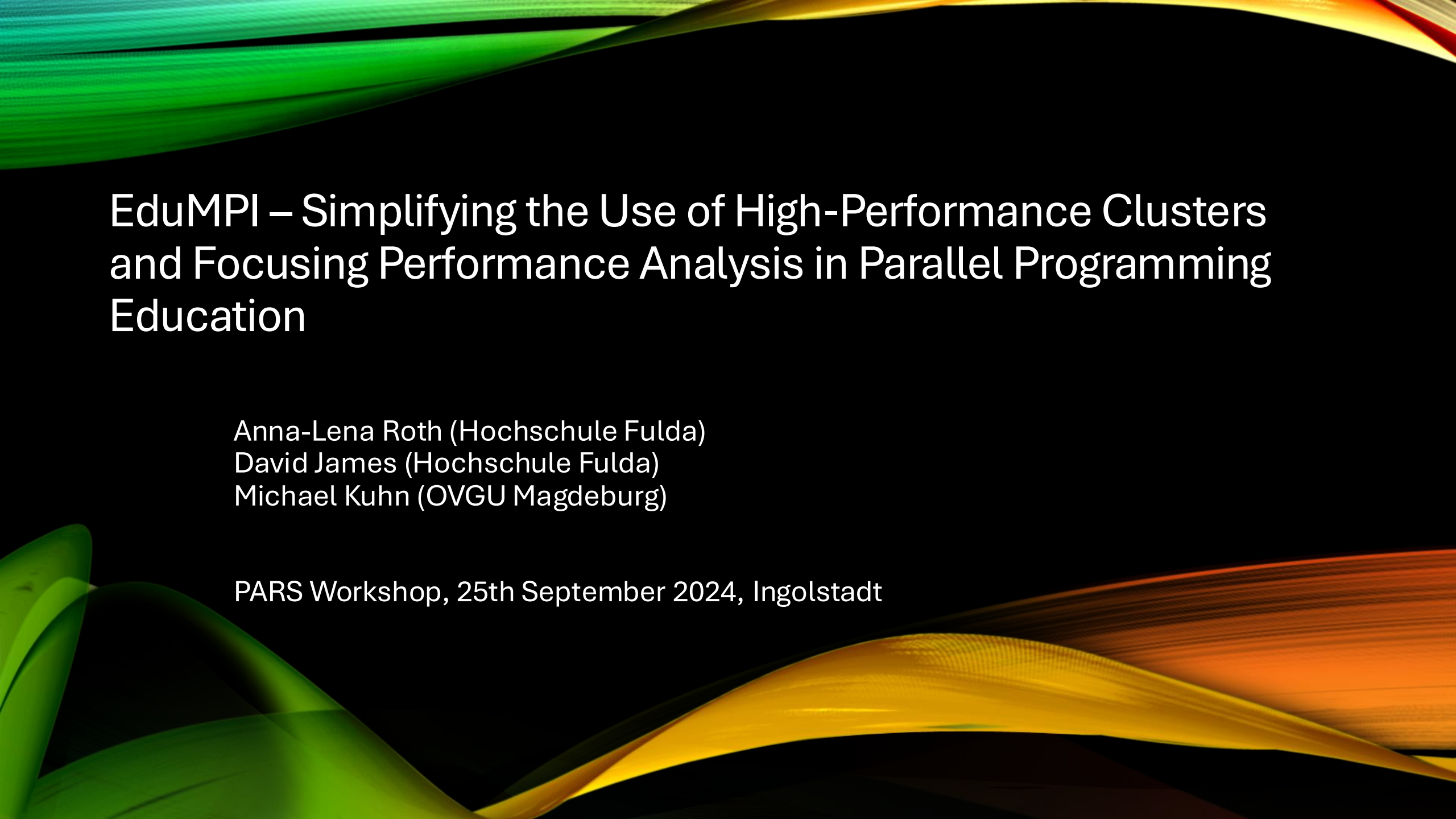




EduMPI – Simplifying the Use of High-Performance Clusters and Focusing Performance Analysis in Parallel Programming Education

Anna-Lena Roth (Hochschule Fulda)
David James (Hochschule Fulda)
Michael Kuhn (OVGU Magdeburg)

PARS Workshop, 25th September 2024, Ingolstadt

PARALLEL PROGRAMMING EDUCATION

- Parallel programming and HPC are becoming increasingly important for developers in science as well as in industry.
- **Computer Science Curricula** (2013): parallel programming should become a core part of the computer science curriculum
- Examples:
 - **Scientific simulations:** Climate models, molecular dynamics or astronomical calculations
 - **Medicine and genetics:** Analysis of genetic data, simulation of protein structures
 - **Engineering:** Fluid dynamics, collision tests, material simulations
 - **Artificial intelligence** and machine learning

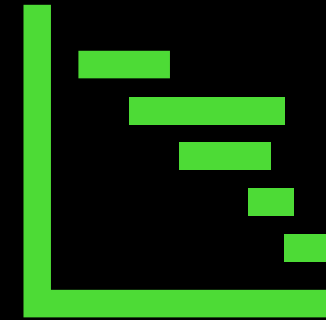


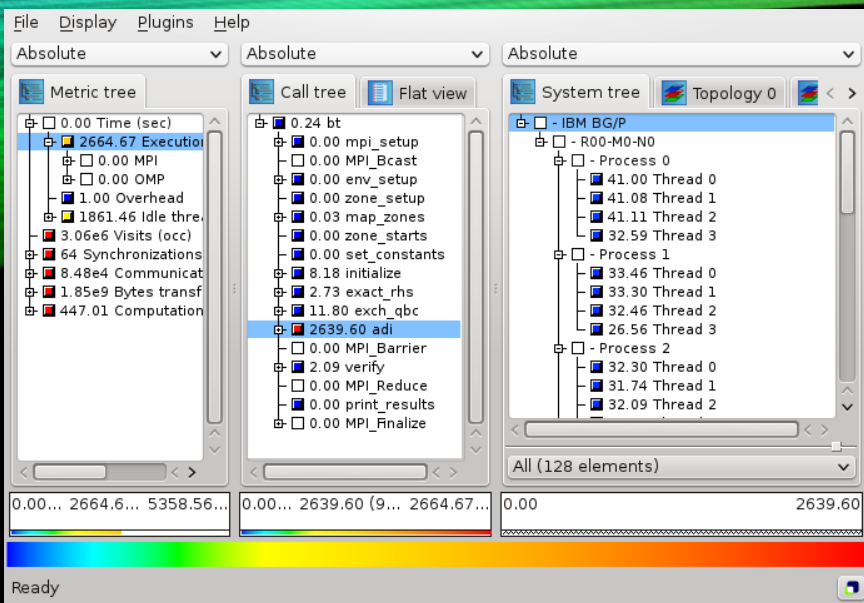
PARALLEL PROGRAMMING EDUCATION PROBLEMS

- Lack of access to hardware
 - expensive
 - administrative effort
 - Complexity of the topic
 - cluster architecture, network topologies
 - storage management
 - resource scheduling (Slurm)
 - programming models (OpenMP, MPI)
- This often leads to concepts of parallel programming only being practiced on individual computers and laptops.
- Lack of practical experience
 - Little understanding of the benefits of HPC
 - Complexity can only be conveyed theoretically

PERFORMANCE ANALYSIS

- Optimization of computing power
 - Efficient use of resources
 - Load distribution
- Elimination of bottlenecks
- Scalability
- Optimization of programming and understanding of program behavior
- Shortening of execution times
- Reduction of costs
- Debugging
- Comparison of systems and architectures

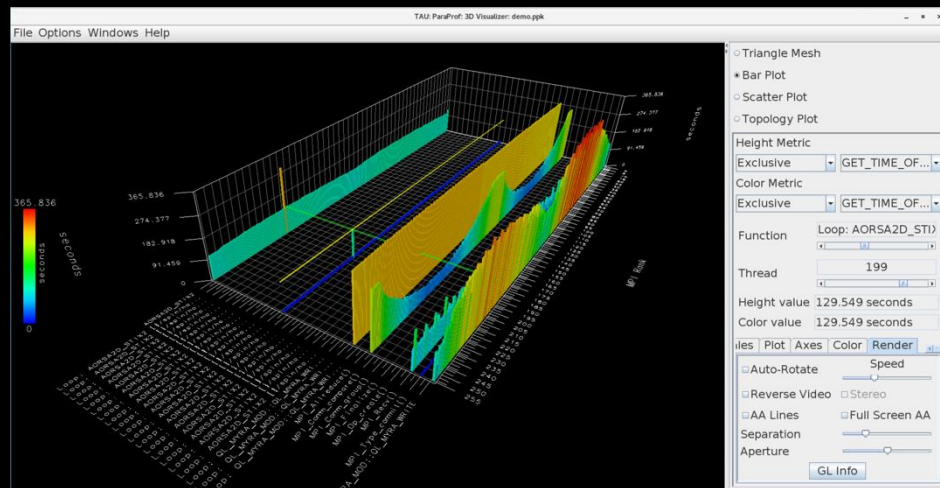




<https://apps.fz-juelich.de/scalasca/releases/cube/4.3/docs/manual/userguide.html>

PERFORMANCE ANALYSIS PROFESSIONAL TOOLS

- Usage usually via instrumentation and measurement
- Complex operability and displays
- Typically: Post-mortem analysis, after termination of the program.
- Very important to be able to write efficient programs.



<https://pop-coe.eu/blog/tool-time-observing-application-performance-using-the-tau-performance-system%C2%AE>

PROFESSIONAL PERFORMANCE ANALYSIS

STUDENTS FEEDBACK

Isn't it enough if my program works without errors? Why do I need performance analysis?

Performance analysis is too difficult! I don't understand what to look for!

Performance analysis is unimportant. It should be a bonus point topic at best.



PARALLEL PROGRAMMING EDUCATION

EduMPI

Login to HPC@HS-Fulda

Remote Hostname or IP-Address:

Authentication (FDAL authentication key):
<fd number>

EduMPI installation path (standard: /opt/edumpi)
<installation path>

Connect to HPC Cluster

Timescale Database Connection

Hostname: <Enter host address>

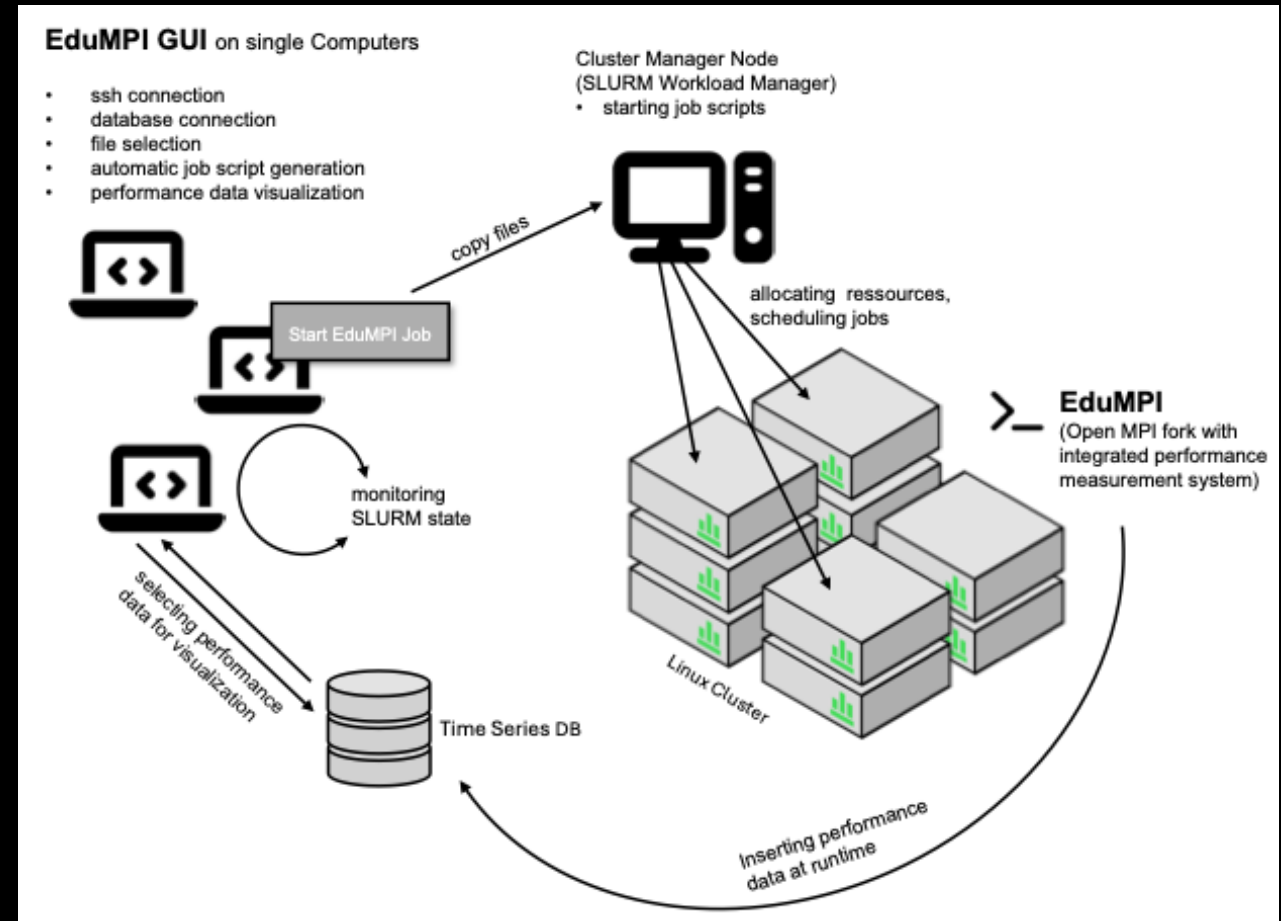
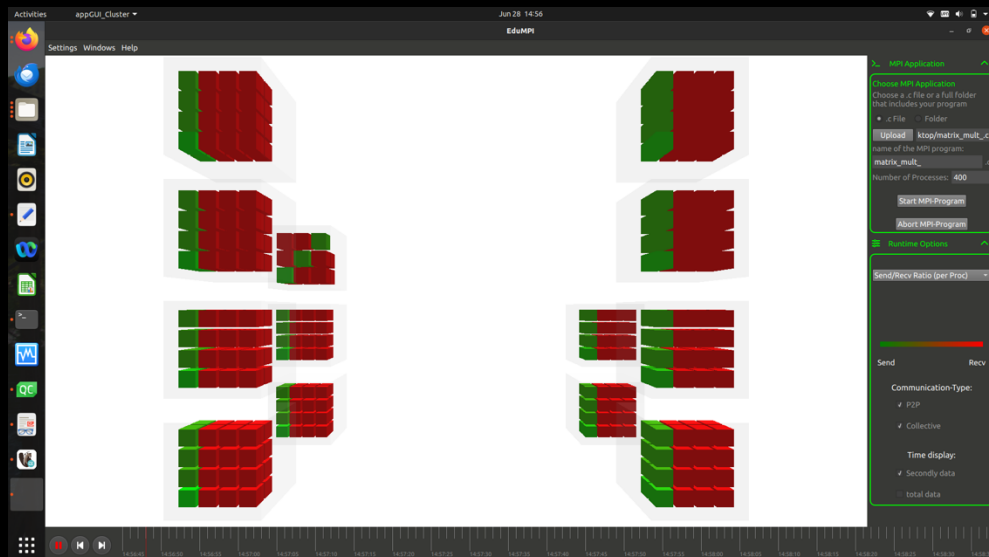
Port: <Enter port>

Database: <Enter database name>

User: <Enter user name>

Password: <password>

confirm

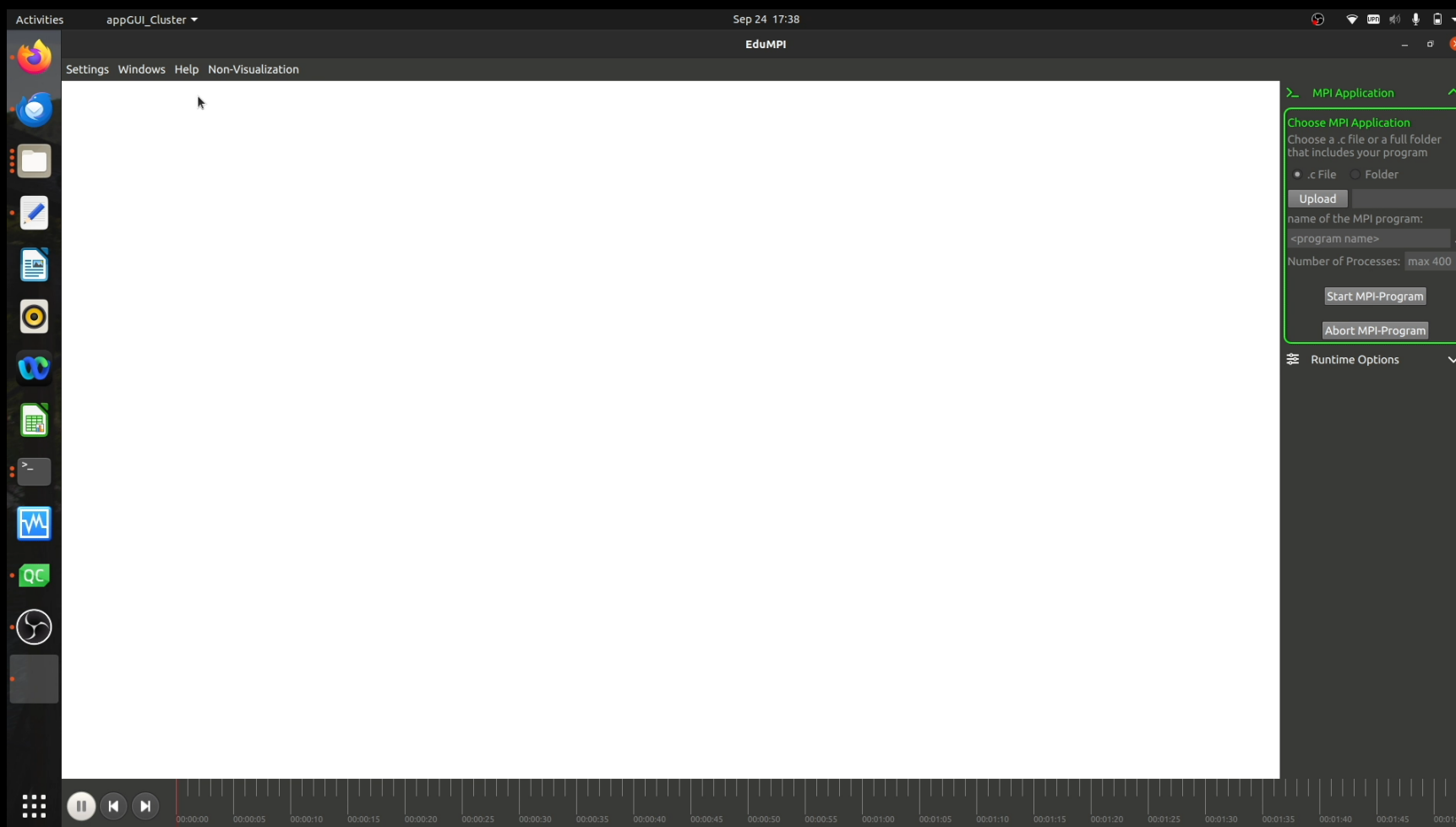


OPEN MPI FORK WITH ANALYSIS FUNCTIONALITY

- **Open MPI, version 5.0** is used (free, open-source)
- For each MPI function called, performance data is automatically measured and collected via a “struct” programmed into the code.
- As soon as a function is completed, the collected data is written to a Time Series database where it is automatically pre-aggregated.
- Special data (e.g. for internal collective communication) can be retrieved through the code-internal work.
- Students simply use EduMPI instead of OpenMPI via the corresponding GUI
 - instrumentation or measurement is not necessary
 - understanding SLURM is not necessary

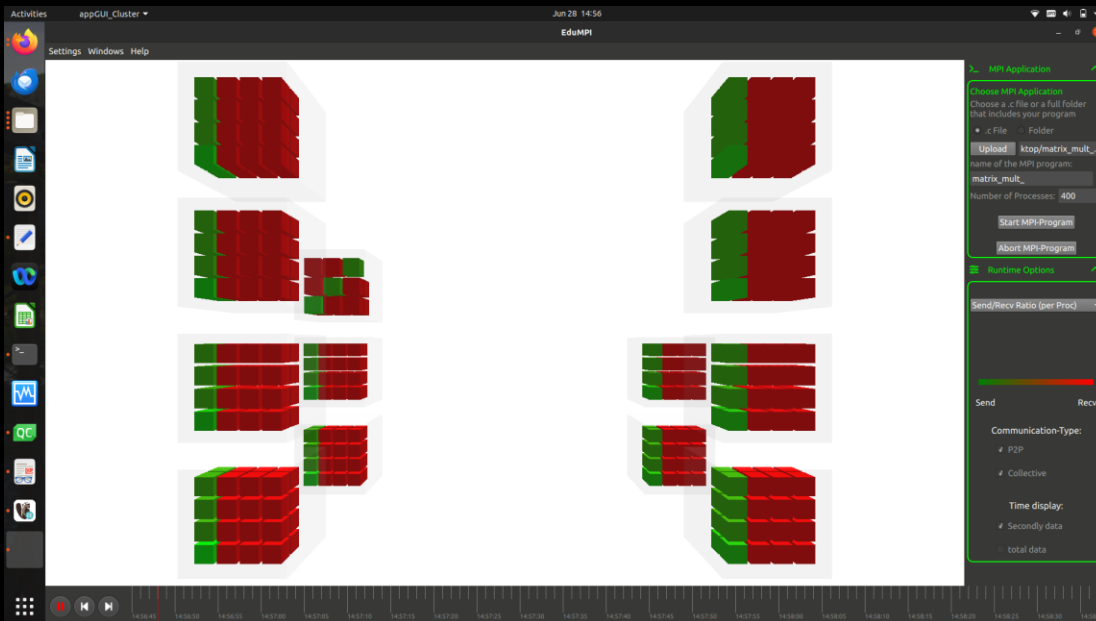
EduMPI

EduMPI GUI – DEMO VIDEO

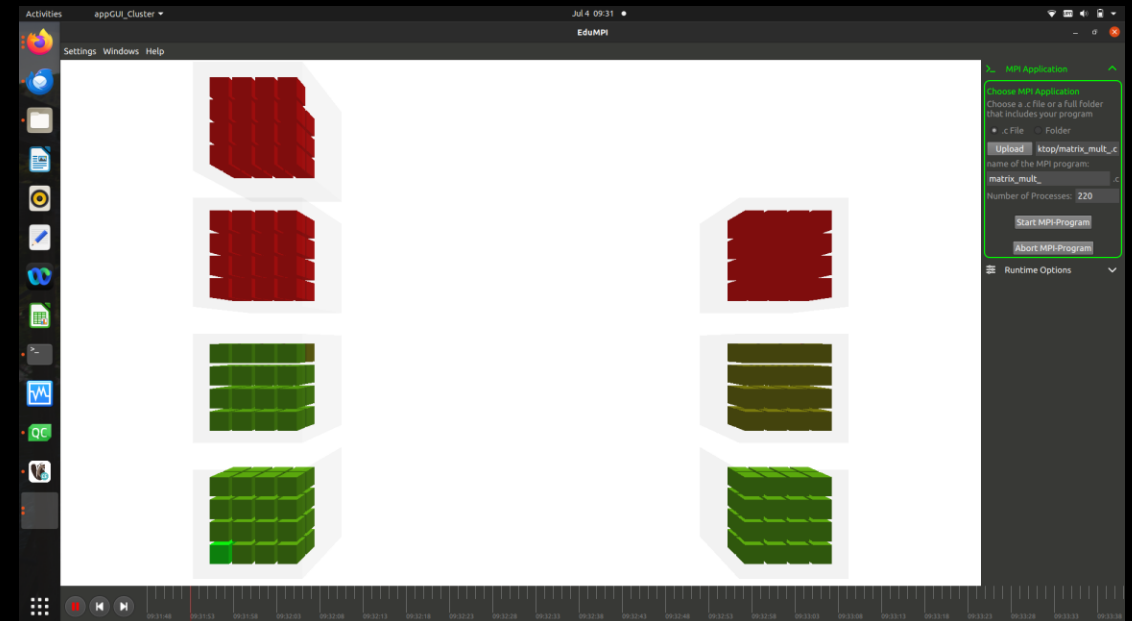


SPECIAL FEATURES - INTERNAL MPI COMMUNICATION (ALGORITHM)

- Same Broadcast, using 400 vs. 220 Ranks



Algorithm: Scatter_Allgather



Algorithm: Binomial Tree

EduMPI FUTURE WORK

- Expansion of functionalities
 - Loading previous EduMPI runs from the database for re-analysis
 - Filter according to communicators used
 - Analysis dashboard with summarized performance data after termination
- Evaluation of the user experience and improvement of parallel programming education
 - Usability tests, including rapid usability tests and comprehensive usability tests under laboratory conditions
 - Regular implementation of user surveys