CUBE EVERYWHERE

Improving Accessibility of Cube Framework for Performance Analysis

November 6, 2025 | Pavel Saviankou, Anke Visser | Research Center Jülich

Structure of the Talk

1. Introduction: What is Cube?
2. Challenge of Access to Cube for the Performance Analysis
3. Solutions: Old and new ways
4. Outlook and Future Work

What is CubeGUI?

CubeGUI is Performance Explorer

- Cube is a powerful graphical tool designed for the **interactive exploration** of performance data.
- It maps performance metrics across **Call Trees, System Topology**, and **Software Metrics**.

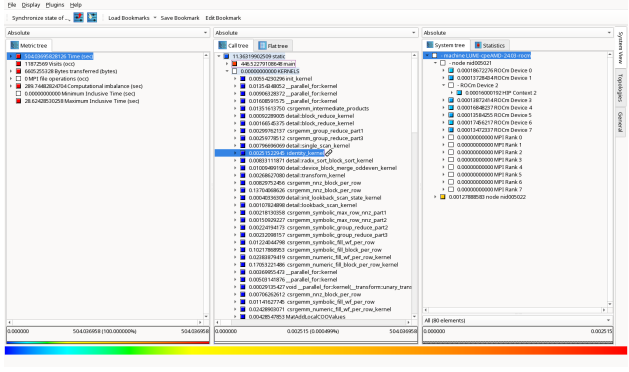

Figure: Multi-Dimensional Viewer

What is CubeGUI?

Serves multiple tools

- **Score-P**: Direct display of performance data.
- **Scalasca**: Display of performance patterns, detected by Scalasca.
- **TAU**: `Tau2Cube` allows to browse the data from TAU.
- **Callgrind***: In future version ($\geq$ 4.10), Cube will display results of Callgrind.

What is CubeGUI?

Detailed Data Exploration

- Displays every value of every metric and every callpath.
- Detailed exploration of data in trees.

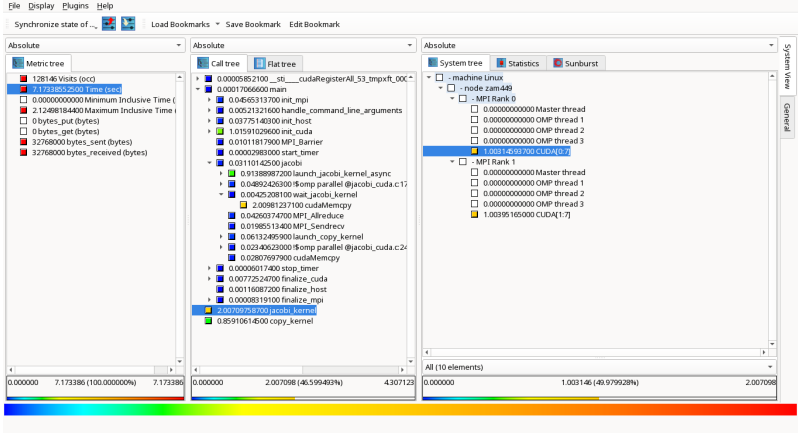

Figure: Granular Access to Metrics and Call Paths

What is CubeGUI?

High-level analysis: POP Analysis

- Focuses on **POP Efficiencies** instead of detailed values for every metric.
- Provided by plugin **"POP Advisor"** and by **"cube_pop_metrics"** tool.

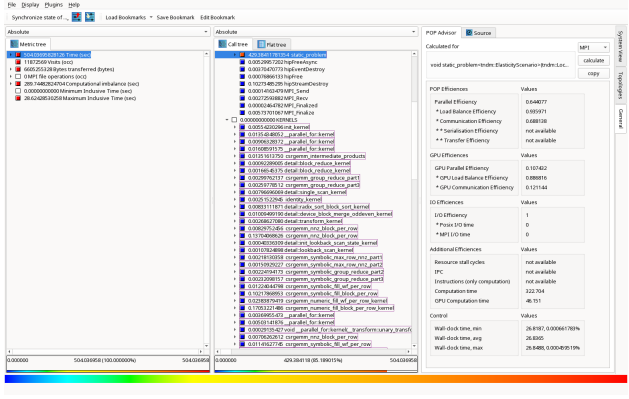

Figure: POP Advisor Plugin View

JÜLICH Forschungszentrum

Challenge of Access to Cube for Performance Analysis (I)

Cube → Data: Bring Cube to the Data

- The simplest solution: Install CubeGUI on the HPC system with the measurement data and use X-Forwarding to display it on your screen.
- Many, but not all HPC systems provide installation via package managers (`Spack`, `EasyBuild`).
- Alternatively the system admin installs the packages manually.
- The analyst may need to install it on the target system themselves.

The Problem

Too often, it is left to the user/analyst to install CubeLib/CubeGUI themselves.
Issue: Missing or non-working `Qt` due to driver issues, library conflicts, or lack of graphical libraries on the HPC system.

Challenge of Access to Cube for Performance Analysis (II)

Data → Cube: Bring Data to the Cube

- If Cube cannot be installed on the measurement system, data must be copied to the user's local system.
- Initial measurements are often "raw" and unfiltered, resulting in large `.cubex` / `.otf2` files, making copying challenging.

Remote Access as an implicit copy of the data

Mounting the remote file system is often possible.

Challenge of Access to Cube for Performance Analysis (III)

Client System is different from the HPC System

- Analysis system might be vastly different than the HPC System (e.g., Windows/MacOS client vs. HPC Linux).
- **Incompatible Endianness:** Correcting endianness on the fly can severely impact the performance of CubeGUI and its tools.
- **Tool Availability:** While Cube is often available on Mac/Win, other complementary tools (such as Vampir or HPCToolkit) may not be, forcing the analyst to use multiple different systems.

Challenge of Access to Cube for Performance Analysis (IV)

Performance Analyst is not a System Administrator

- Performance Analysis should focus on the *performance*, not on the software configuration and maintenance.
- Analysts are often not experts in software stacks or system administration because they are highly specialised.

Solutions for these Challenges (I)

Cube Everywhere

- **No need to copy data**: Use X-Forward, SSH mount, or a **pure client/server architecture**.
- **Many client are supported**: Support for various client platforms: *Unix*, Win, Mac, (**Web, Android, iOS**).
- **No admin rights needed**: Install via **TarBall** with Prefix in `$HOME`.
- **No installation needed**: Provide **AppImage** (*Unix*), **.dmg** (*Mac*), and similar binaries for seamless execution.

Solutions for these Challenges (II)

Client-Server Setup: Comparison

The client-server setup is now the most usable solution for the access challenges.

Traditional X-Forwarding

1. Open terminal with `ssh -X user@server.example.com`
2. Type `<path to cube>/cube`
3. Wait until GUI appears
4. Click ”Open” and browse data.

Client-Server Tunneling

1. Open terminal with `ssh -L 3300:localhost:3300 user@server.example.com`
2. Type `<path to server>/cube_server`
3. In a separate window, start the local CubeGUI client.
4. Click ”Open URL” feature and browse data.

Solutions for these Challenges (III)

Client-Server Advantage

- **Minimal HPC Requirements**: `cube_server` requires only C++11 and a few system libraries. (*Easy to fulfill on HPC*).
- **Flexible GUI**: `CubeGUI` requires `Qt` (and additional libraries), which is *easier to fulfill on a workstation/laptop*.

Current Limitation and Future Work

Limitation: Not all plugins use the server-side setup yet (only `SourceCodeViewer` and `POP Advisor`).
Future: `cube_server` will have **MPI implementation** and run on the backend → Even more performant.

Solutions for these Challenges (IV)

More Clients for Client-Server

- Currently: Unix, Mac and Win clients.
- **Android client** is in testing: ***Available from*** $\geq$ 4.10.
- iOS is on the horizon.
- And more...

Figure: Cube in Play Store

Solutions for these Challenges (V)

Download but not Install

- **Problem**: Analyst cannot install Cube even on their own corporate laptop.
- **Solution (Linux/Mac)**: We provide **self-containing*** AppImage for Unix, and .dmg for MacOS.
- **Solution (Windows)**: We provide a setup.exe which installs in the user-specific directory. However, the initial execution often requires administrator rights, which is not perfect yet.

Solutions for these Challenges (VI)

Browser is always available

Ultimate Solution: Just open Cube in Web browser!

- New supported platform: **WebAssembly client**.
- Integration into JupyterLab
- Available from version 4.10 onwards.

Advantages for users

- Web Browser is usually available - no additional Installation required
- Authentication process and connection security is done by JupyterLab
- Full access to remote file system and to other resources
- Responsive GUI
- Cube can recover from temporary network failures

Solutions for these Challenges (VII)

JupyterLab on HPC

- Available on all HPC systems in JSC: JUWELS, JURECA, JUSUF, JUPITER(upcoming).
- Simplifies access for new users especially in workshops
  - Register for acess to HPC System
  - Login to JupyterLab
  - Access to CubeGUI and and to command line tools
  - Access to tutorial data

PROBLEM SOLVED Simple access to Cube $\rightarrow$ User registration + 1 click = Performance Analysis

Solutions for these Challenges (VIII)

JupyterLab for User: Start Screen

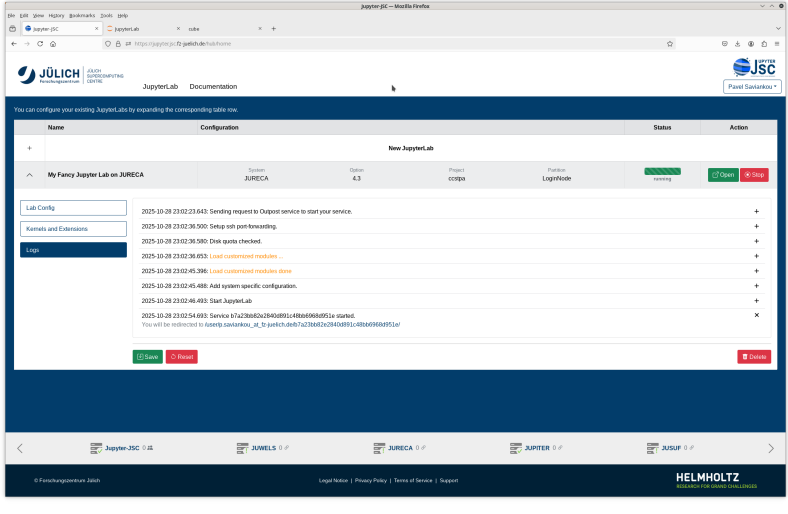

Figure: Login to JupyterLab

Solutions for these Challenges (IX)

JupyterLab for User: Available Apps

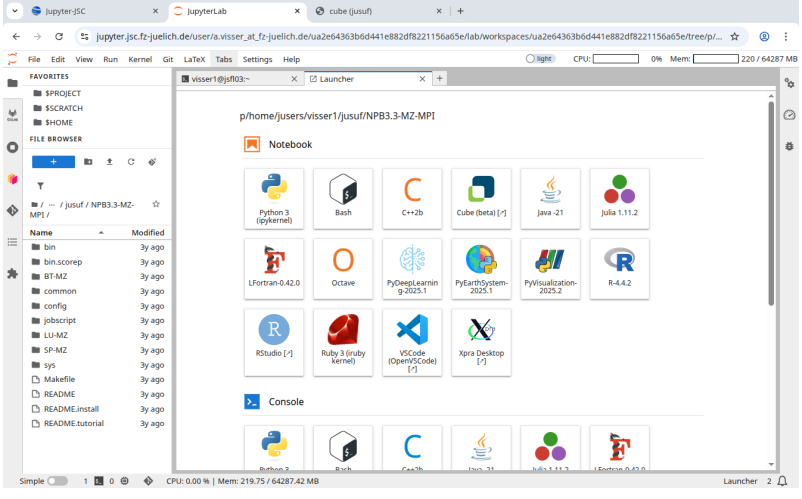

Figure: Access to Cube and Other Tools

JÜLICH Forschungszentrum

Solutions for these Challenges (X)

JupyterLab for User: Cube in Action

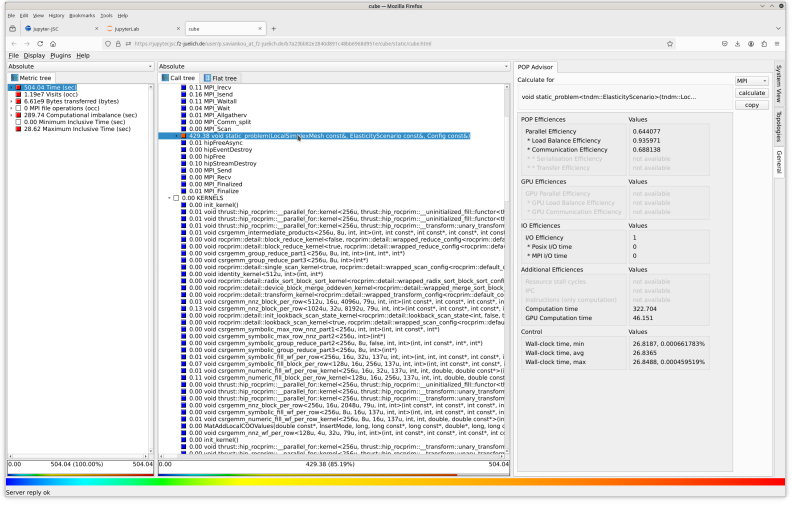

Figure: Cube running POP Advisor within the Browser

JÜLICH Forschungszentrum

Solutions for these Challenges (XI)

Jupyter for Admin: Setup Steps

As an administrator, setting up Cube for JupyterLab is simple:

1. Install CubeLib on the HPC system (e.g. using easybuild)
2. Install ”Cube web service” package (e.g. using pip)
3. Install ”Jupyter cube proxy” package (e.g. using pip)
4. Configure the JupyterLab extension.
5. Done!

Once these steps are completed, users can access Cube directly from JupyterLab without any additional installation steps on their part.

Outlook and Future Work

Improvements

Client - Server Improvements

- More plugins use the server-side calculations.
- **Session management on server**: Start analysis on desktop, proceed on laptop, finish on tablet.
- **Global ID** for measurements: remove the struggle of remembering where the data is located (e.g., `cube://localhost:3300/profile.cubex` hides the target system).

POP Advisor Improvements

- Reusing FOA of one analysis in another analysis in comfortable way.
- `cube_pop_metrics` + `POP Advisor` will reuse **Focus of Analysis (FOA)** for comfortable analysis.

Any Questions?

Thank You!

Contact:

P.Saviankou@fz-juelich.de
A.Visser@fz-juelich.de

Performance Optimisation and Productivity 3

A Centre of Excellence in HPC

Contact:

- https://www.pop-coe.eu
- pop@bsc.es
- @POP_HPC
- youtube.com/POPHPC

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101143931. The JU receives support from the European Union's Horizon Europe research and innovation programme and Spain, Germany, France, Portugal and the Czech Republic.