

JuViz: Browser-based interactive Visualization for HPC

Pairing Cloud and HPC for Visualization with JupyterLab and ParaView trame

Jonathan Windgassen¹, Jens Henrik Göbbert¹, Sébastien Jourdain²

1: Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH 2: Kitware, Inc.

Introduction

JuViz is a novel approach to browser-based interactive visualization. It bridges the cloud-based development environment JupyterLab^[1] with ParaView.^[2] With JuViz, users can access flexible visualization apps in the browser. It was originally developed for the cloud platform JupyterJSC at the Jülich Supercomputing Centre. JuViz allows adding customized workflows in form of visualization apps, which can be specialized for certain communities or visualization tasks, tailored to specific needs and requirements. It enables the easily accessible, intuitive and flexible visualization of large data, an essential aspect to find wide adoption within the scientific community and allows easy integration into existing visualization workflows.

Key Features

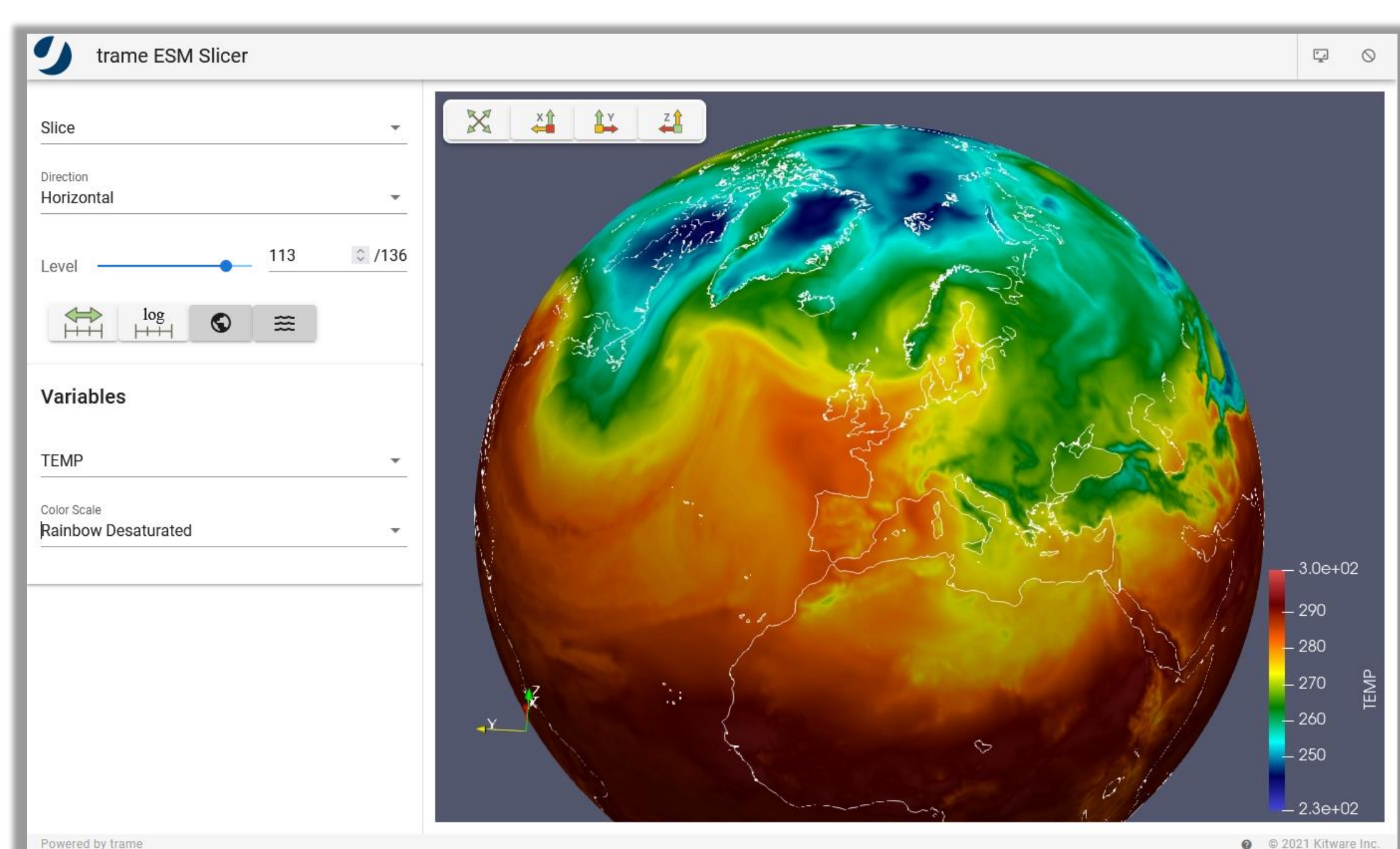
- Designed for HPC
- Integrated into JupyterLab via custom JupyterLab extension
- Fast development and integration of visualization apps
- Development in Python
- Apps can be easily shared with other users on the cluster
- Connection with ParaView servers for **Remote Rendering**
- **In-situ Visualization** with ParaView Catalyst^[3] / AdiosCatalyst
- Compatibility with ParaView and other visualization frameworks
- Platform and architecture agnostic

Step 1: Design your trame App

trame is a framework for creating interactive web applications, developed by Kitware. It can be used with VTK/ParaView, but also supports other popular libraries like matplotlib, code, markdown, etc. Applications are not limited to visualization and analysis, but can also be used for machine learning, dataset exploration, etc.

Example *trame* app: ESM-Slicer

- Visualize and analyze atmospheric data
- Slice the dataset for the different variables in 2D and 3D
- ~700 Lines of Code → fast development



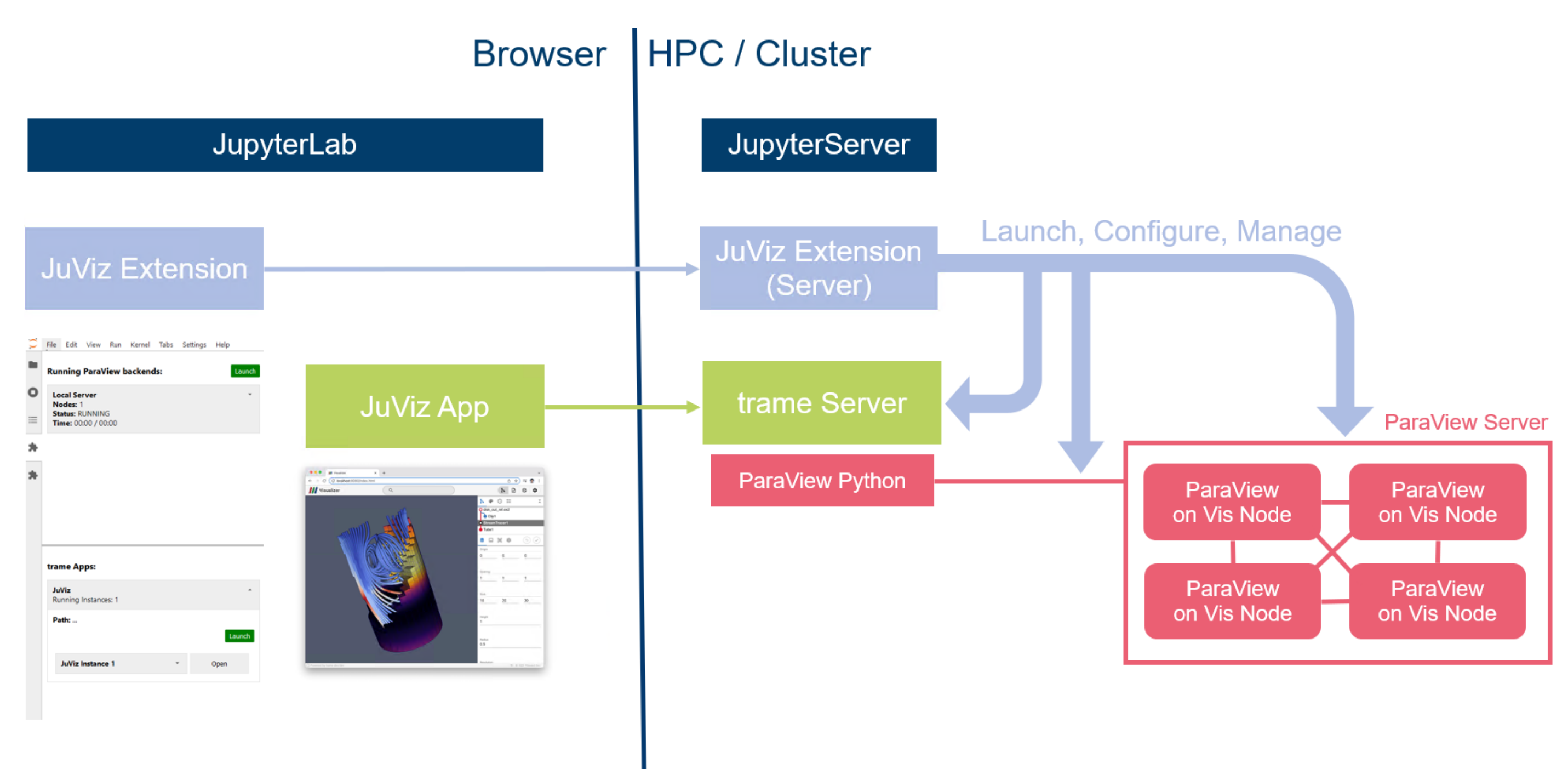
JuViz: https://github.com/jwindgassen/jupyter_viz_extension

trame: <https://kitware.github.io/trame/>

[1]: <https://jupyterlab.readthedocs.io/en/latest/>

[2]: <http://www.paraview.org>

[3]: <https://www.paraview.org/hpc-insitu/>

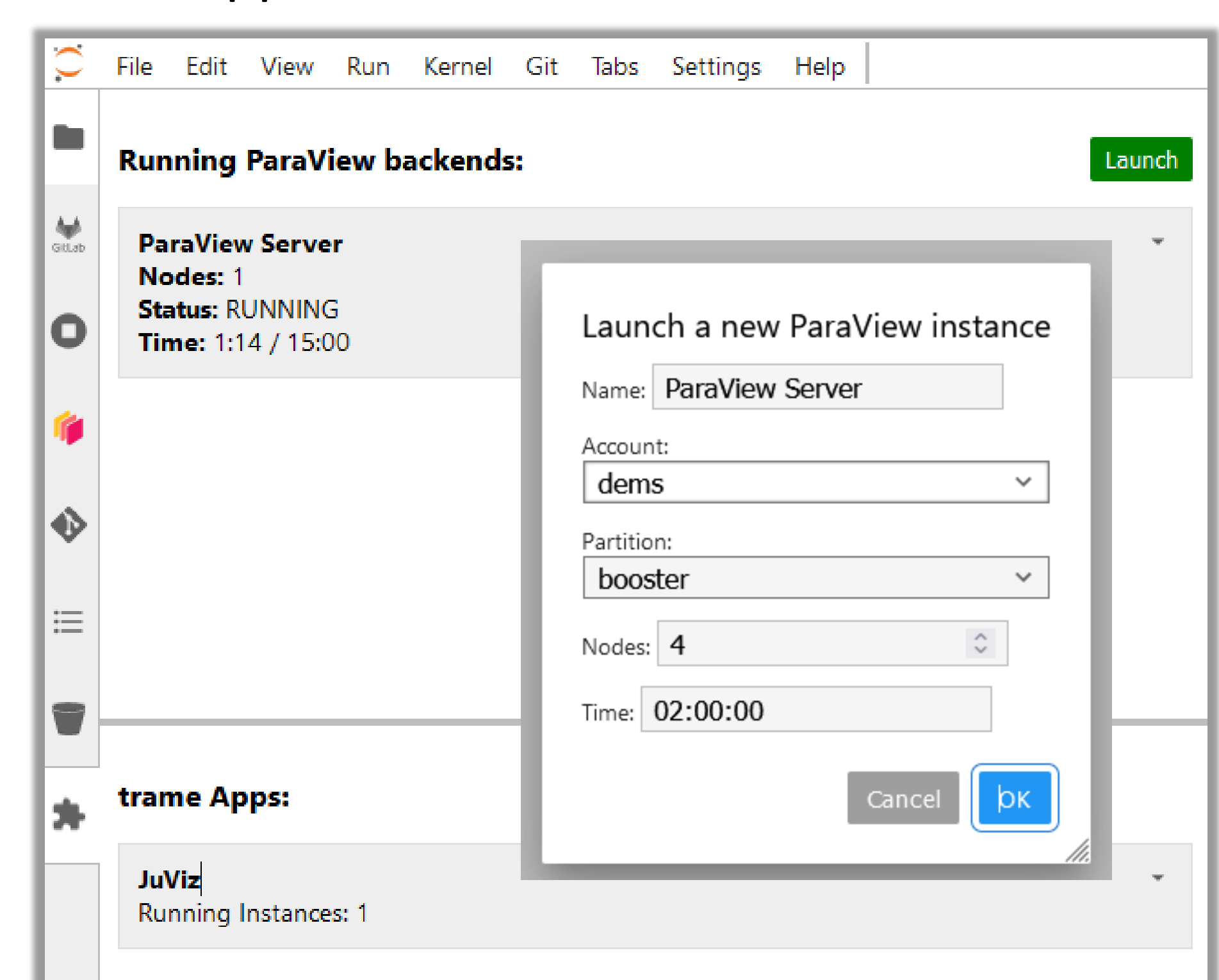


Step 2: Integrate into JuViz

- Upload *trame* app to the cluster and add it to JuViz
- Add config files: launch options, environment configuration, and loading ParaView, Python, and *trame* dependencies

Step 3: Launch App and Connect to Simulation

- Launch a ParaView Server via the JuViz UI
- Connect the ParaView Server to a running simulation on the cluster with **AdiosCatalyst**
- Launch app and connect to the ParaView server
- Open and use app in the browser



Acknowledgement

This work was carried out as part of the InHPC-DE project, funded by the Federal Ministry of Education and Research (BMBF) of Germany. We extend our sincere gratitude to the BMBF for their generous support, which made this research possible.



Contact: j.windgassen@fz-juelich.de | Website: www.fz-juelich.de/ias/jsc

Mitglied der Helmholtz-Gemeinschaft