

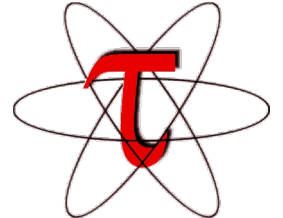
ParaTools Pro for E4S™: An HPC-AI ecosystem for science

14th Annual MVAPICH Users Group Meeting

11:15am – noon ET, Monday, August 17th, 2026
The Ohio State University, Columbus, OH



HPSF
HIGH PERFORMANCE
SOFTWARE FOUNDATION



Sameer Shende
Research Professor and Director,
Performance Research Laboratory, OACISS, University of Oregon
President and Director, ParaTools, Inc.
sameer@cs.uoregon.edu

https://oaciss.uoregon.edu/e4s/talks/E4S_MUG26.pdf



UNIVERSITY
OF OREGON

ParaTools

**Your software environment
should not be your first
research project**

Spend less time configuring
More time discovering

Faster onboarding

Faster discovery

E4S: An HPC-AI Software Ecosystem for Science

Your software environment should not be your first research project

- Complex HPC software stacks can take days – or weeks – to configure correctly
- E4S provides a Spack-based curated, tested, and interoperable scientific software ecosystem
- Prebuilt containers and facility-ready installations accelerate on-boarding and researchers spend more time advancing science

Spend less time configuring, more time discovering

- Start quickly with ready-to-use, reproducible scientific software environments across laptops, clusters, clouds, leadership systems
- Use tested combinations of libraries, tools, compilers, and programming models optimized for CPUs and GPUs
- Focus researcher effort on simulation, analysis, AI, and scientific insight

Faster onboarding, faster discovery

- Use prebuilt containers, binaries, and Spack environments to get started quickly
- Reduce platform-specific setup and troubleshooting; deploy complete environments without building every dependency
- Help new team members become productive sooner and move rapidly from installation to simulation, analysis and discovery

E4S: one of the world's largest collection of open-source GPU portable scientific libraries and tools

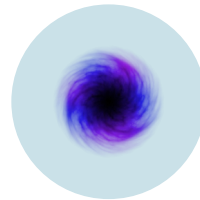
- More than 100 open-source libraries, tools, applications for scientific computing
- Performance-portable programming models enable one code base across diverse systems
- Broad support for NVIDIA, AMD, and Intel GPU architectures

E4S: Better quality, documentation, testing, integration, delivery, building & use

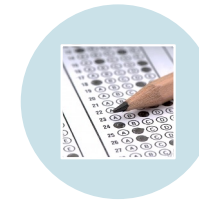
Delivering HPC software to facilities, vendors, agencies, industry, international partners in a brand-new way



Quality Commitment
Community policies, improvement



DocPortal
Single portal to all E4S product info



Portfolio testing
Especially leadership platforms



Curated collection
The end of dependency hell



Semi Annual releases
Release 26.06 – June



Build caches
10X build time improvement



Turnkey stack
A new user experience



<https://e4s.io>



E4S Strategy Group
US agencies, industry, international

Motivation

- As our software gets more complex, it is getting harder and harder to install and operate our HPC-AI workflows on GPUs correctly in an integrated and interoperable software stack on commercial cloud platforms!
- What makes for a productive HPC-AI environment on the cloud platforms?

E4S: An HPC-AI Software Ecosystem for Science

<https://e4s.io>



E4S 26.06: What's New?



- E4S is a curated Spack based distribution of 125+ HPC-AI packages.
- Compared with 25.11, E4S 26.06 adds a much richer agentic AI layer, deeper LLM inference optimization, and more workflow tooling around fine-tuning, serving, and domain-specific AI
 - Sglang, MCP, langchain, langgraph, a2a, flash_attr
 - Coding agent pi, NVIDIA NemoClaw, OpenClaw, NVIDIA NeMo, NVIDIA BioNeMo, Megatron, transformers, cupy
 - Ray, peft, triton
 - API: openai, google, anthropic
 - Chemistry: rdkit
- Support for updated Rocky Linux 9.8 with Hopper and Blackwell, Ubuntu 24.04 LTS.
- Spack 1.1.1 integration [<https://github.com/spack/spack/releases/tag/v1.1.1>].
- HPC Applications include: CP2K, DeaII, FFTX, GROMACS, LAMMPS, Nek5000, Nekbone, NWChem, OpenFOAM, WarpX, WRF, Quantum Espresso, and Xyce with GPU support where available.
- CUDA upgraded to 13.2 (aarch64, x86_64), ROCm upgraded to 7.2.3, oneAPI upgraded to 2026.0.
- Adaptive Computing's Heidi web-based platform for multi-user, multi-node, ParaTools Pro for E4S™ with Ubuntu 26.04 LTS and Rocky Linux 9.7 cloud images on AWS, Microsoft Azure, Google Cloud, OCI, and IBM Cloud.



- <https://adaptivecomputing.com/> and <https://paratoolspro.com>

E4S: Extreme-scale Scientific Software Stack



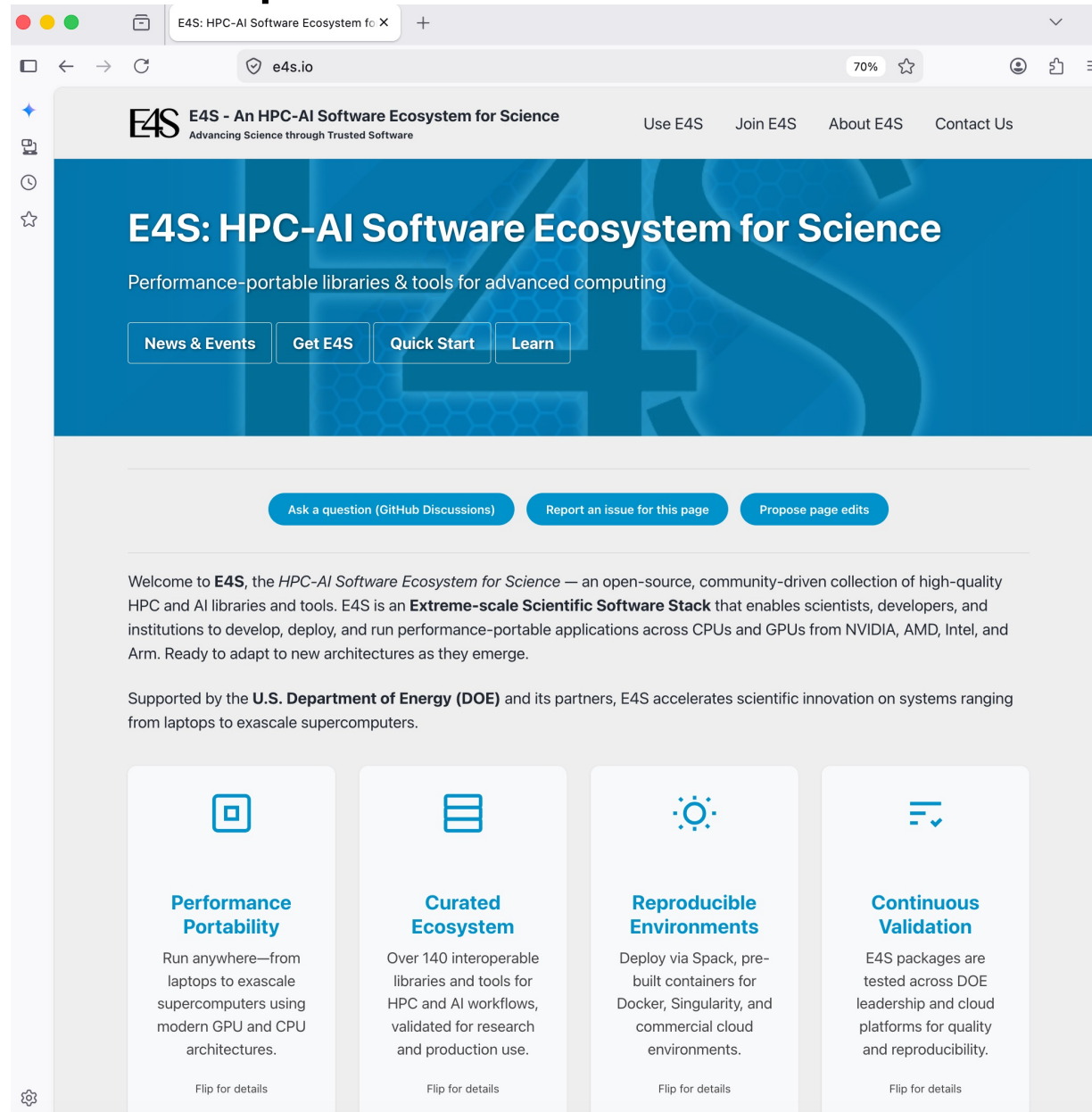
<https://e4s.io>

About E4S

- E4S is an **ecosystem for science** and a community effort to provide open-source software packages for developing, deploying and running scientific applications on HPC platforms.
- E4S has built a comprehensive, coherent software stack that enables application developers to productively develop highly parallel applications that effectively target diverse exascale architectures.
- E4S provides a curated, Spack based software distribution of 125+ HPC (TAU, Trilinos, PETSc, OpenFOAM, Gromacs, Nek5000, LAMMPS), EDA (e.g., Xyce), and AI/ML packages (e.g., Google ADK, NVIDIA NeMo™, NVIDIA BioNeMo™, Vllm, HuggingFace CLI, TensorFlow, PyTorch, OpenCV, TorchBraid, Scikit-Learn, Pandas, JAX, LBANN optimized for GPUs where available).
- Base images and full featured containers (with GPU support) and DOE LLVM containers.
- Commercial support for E4S through ParaTools, Inc. for installation, maintaining an issue tracker, and ECP AD engagement.
- E4S for clouds: Adaptive Computing's Heidi with ParaTools Pro for E4S™ image for **AWS, GCP, IBM Cloud, Azure, OCI**.
- With E4S Spack binary build caches, E4S supports both bare-metal and containerized deployment for GPU based platforms.
 - x86_64, ppc64le (IBM Power 10), aarch64 (ARM64) with support for CPUs and GPUs from NVIDIA, AMD, and Intel
 - Container images on DockerHub and E4S website of pre-built binaries of ECP ST products.
- e4s-chain-spack.sh to chain two Spack instances allows us to install new packages in home directory and use other tools.
- e4s-cl container launch tool allows binary distribution of applications by swapping MPI in the containerized app w/ system MPI.
- e4s-alc is an à la carte tool to customize container images by adding system and Spack packages to an existing image.
- E4S 26.06 released on June 22, 2026: https://e4s.io/talks/E4S_26.06.pdf



Updated E4S website <https://e4s.io>



Chatbot integration in E4S Website

SS Sameer Shende
Free

Upgrade

Chats

Profiling tools in E4S

E4S libraries for PDE solvers

Example chat: Ask anything

Software Ecosystem for Scienc...

GPTs

Explore

Projects

Atlas

Library

Search chats

New chat

Software Ecosystem for Science Guide 5.1

Get Plus

Software Ecosystem for Science Guide

By Michael Heroux

Expert on scientific libraries and tools in the E4S (Ecosystem for Science) ecosystem.

Which E4S libraries should I use for parallel PDE...

How do I install and configure PETSc with Spack?

Can you explain how Kokkos interacts with Trilinos?

What profiling tools in E4S help with performance...

+ Ask anything

Microphone icon



E4S Bot trained on E4S content on OpenAI's ChatGPT



New chat

Search chats

Library

Atlas

Projects

GPTs

Explore

Software Ecosystem for Scienc...

Chats

Profiling tools in E4S

E4S libraries for PDE solvers

Example chat: Ask anything

Software Ecosystem for Science Guide 5.1

Get Plus

Share

What profiling tools in E4S help with performance optimization?

E4S provides a **comprehensive suite of profiling and performance analysis tools** that help scientists and developers understand and optimize performance on heterogeneous HPC systems. According to the *E4S Performance Tools Report*, the key profiling tools are:

1. TAU (Tuning and Analysis Utilities)

- Role:** The most versatile profiler/tracer in E4S; known as the “*Swiss Army knife*” of HPC performance tools.
- Instrumentation methods:**
 - `tau_exec` runtime interception — no recompilation required.
 - Source instrumentation (PDT)** — automatically inserts timers into C/C++/Fortran code.
 - Compiler-based hooks** — uses compiler flags like `-finstrument-functions`.
- Capabilities:**
 - Profiling (summary of where time is spent).
 - Tracing (timeline of events across processes and GPUs).
 - GPU kernel timing (via CUDA/HIP/oneAPI).
 - Memory, communication, and I/O profiling.
- Integration:** Works with PAPI for hardware counters and Score-P for trace collection.

+ Ask anything

🗣️

⏸️

New version of GPT available - Continue chatting to use the old version, or start a [new chat](#) for the latest version.

E4S Documentation

About E4S - E4S - An HPC-AI Softw X

e4s.io/about/

90%

E4S

E4S - An HPC-AI Software Ecosystem for Science

Advancing Science through Trusted Software

Use E4S

Join E4S

About E4S

Contact Us

ABOUT E4S

Overview

E4S Introduction

E4S Learning

First-Time Users

E4S Product Families

E4S Containers

E4S FAQs

AI and the E4S Website

News & Events

About E4S

Ask a question (GitHub Discussions)

Report an issue for this page

Propose page edits

E4S provides a curated collection of scientific libraries and tools supporting scientific discovery through advanced computing technologies. The primary value E4S provides is bringing together a large collection of independently developed libraries and tools and enhancing the overall enterprise of developing and using them in aggregate. This portion of the E4S website focuses on topics that help understand what E4S is, how to contribute to it, and how it can be used.

Topic	Description
Overview - About E4S	High-level summary of the E4S project, mission, scope, and target audiences.
E4S Introduction	Short primer on E4S goals, components, and how it supports HPC-AI scientific workflows.
E4S First-Time Users	Quickstart guidance for new users: getting started, basic concepts, and common workflows.
E4S Product Families	Catalog and explanation of E4S product groups (programming systems, libraries, tools, etc.).
E4S FAQs	Frequently asked questions covering usage, licensing, support, and common troubleshooting.
E4S News & Events	Announcements, release notes, and upcoming events related to E4S.

© 2025 E4S Project, a Series of LF Projects, LLC. For website terms of use, trademark policy and other project policies please see lfprojects.org. Supported by the U.S. Department of Energy Office of Advanced Scientific Computing Research.



E4S: Bare-metal installation, Containers, and Cloud images

Get E4S - E4S - An HPC-AI Software Ecosystem for Science

e4s.io/get/

90%

E4S

E4S - An HPC-AI Software Ecosystem for Science

Advancing Science through Trusted Software

Use E4SJoin E4SAbout E4SContact Us

GET E4S

Overview

E4S Spack Caches

E4S Container Download

ParaTools Pro for E4S™

Get E4S

Ask a question (GitHub Discussions)

Report an issue for this page

Propose page edits

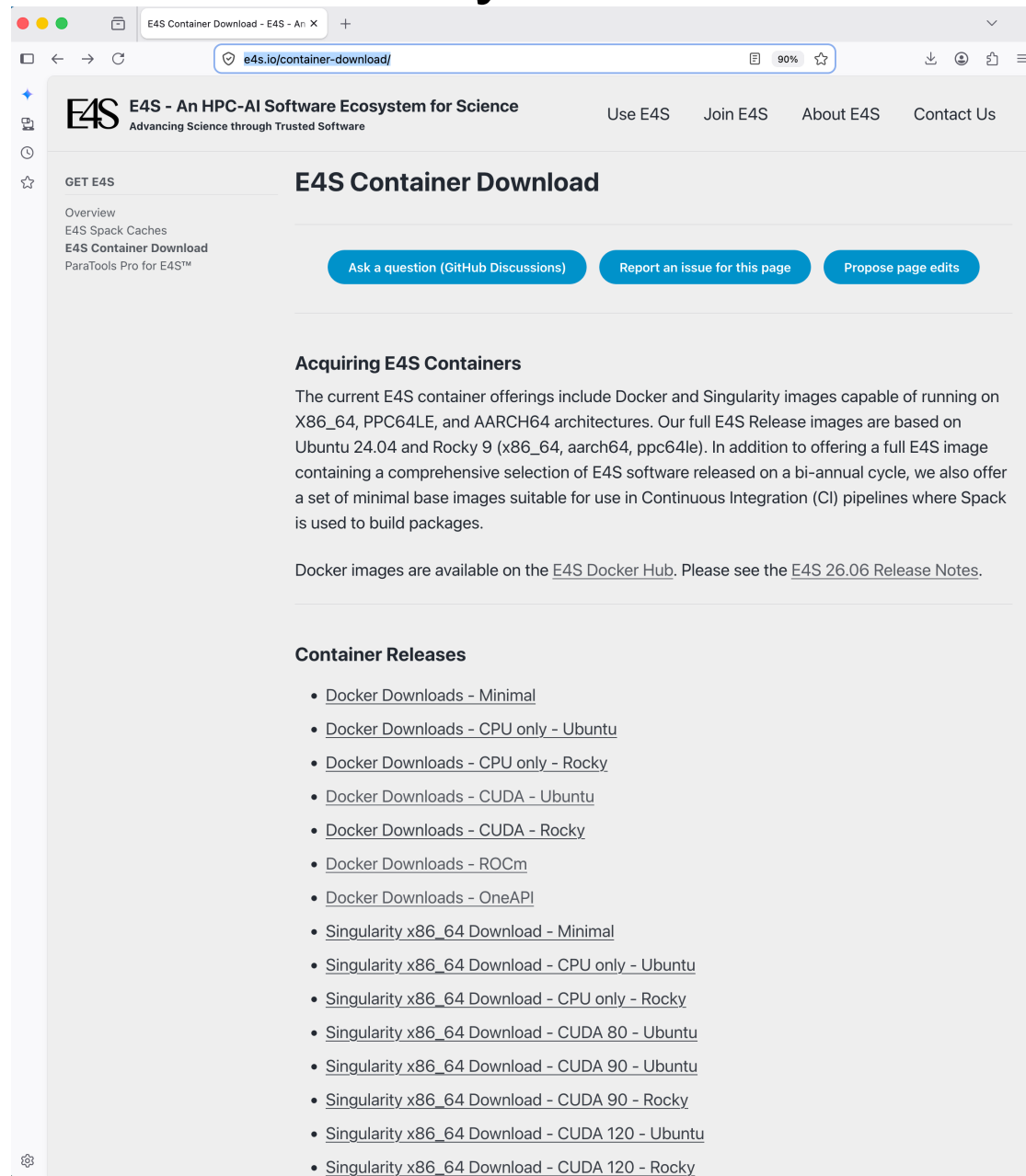
E4S offers multiple methods for obtaining its collection of HPC and AI software packages. Below is an overview of these deployment options. Additional documentation is available from the side menu.

Get Option	Description
Spack Build Cache	Pre-built binaries provided via Spack build caches enable faster installations without local compilation. More information available on the E4S Buildcache Page .
Containers	E4S offers containerized versions of its software stack, compatible with Docker, Singularity, Shifter, and CharlieCloud. See the Container Download Page for available containers. See side menu for additional container information.
Commercial Cloud Options	E4S containers are available via ParaTools Pro for E4S™ , including AWS Marketplace . E4S can be deployed on Amazon Web Services (AWS) for scalable cloud-based HPC/AI workloads and Google Cloud Platform (GCP) with access to high-performance VMs and storage.

© 2025 E4S Project, a Series of LF Projects, LLC. For website terms of use, trademark policy and other project policies please see [lfprojects.org](#). Supported by the U.S. Department of Energy Office of Advanced Scientific Computing Research.



Download E4S Containers: Rocky Linux 9.7 and Ubuntu 24.04 LTS



The screenshot shows the 'E4S Container Download' page on the website e4s.io. The page header includes the E4S logo and navigation links: 'Use E4S', 'Join E4S', 'About E4S', and 'Contact Us'. The main heading is 'E4S Container Download'. Below this, there are three buttons: 'Ask a question (GitHub Discussions)', 'Report an issue for this page', and 'Propose page edits'. The page content is divided into two main sections: 'Acquiring E4S Containers' and 'Container Releases'. The 'Acquiring E4S Containers' section explains that the current E4S container offerings include Docker and Singularity images capable of running on X86_64, PPC64LE, and AARCH64 architectures. It mentions that the full E4S Release images are based on Ubuntu 24.04 and Rocky 9 (x86_64, aarch64, ppc64le). It also offers a set of minimal base images suitable for use in Continuous Integration (CI) pipelines where Spack is used to build packages. The 'Container Releases' section lists various download links for Docker and Singularity images, categorized by architecture (x86_64) and OS (Ubuntu, Rocky). The links include 'Minimal', 'CPU only', 'CUDA', and 'ROCm' for both Docker and Singularity, as well as 'OneAPI' for Docker.

E4S - An HPC-AI Software Ecosystem for Science
Advancing Science through Trusted Software

Use E4S Join E4S About E4S Contact Us

E4S Container Download

GET E4S

Overview
E4S Spack Caches
E4S Container Download
ParaTools Pro for E4S™

Ask a question (GitHub Discussions) Report an issue for this page Propose page edits

Acquiring E4S Containers

The current E4S container offerings include Docker and Singularity images capable of running on X86_64, PPC64LE, and AARCH64 architectures. Our full E4S Release images are based on Ubuntu 24.04 and Rocky 9 (x86_64, aarch64, ppc64le). In addition to offering a full E4S image containing a comprehensive selection of E4S software released on a bi-annual cycle, we also offer a set of minimal base images suitable for use in Continuous Integration (CI) pipelines where Spack is used to build packages.

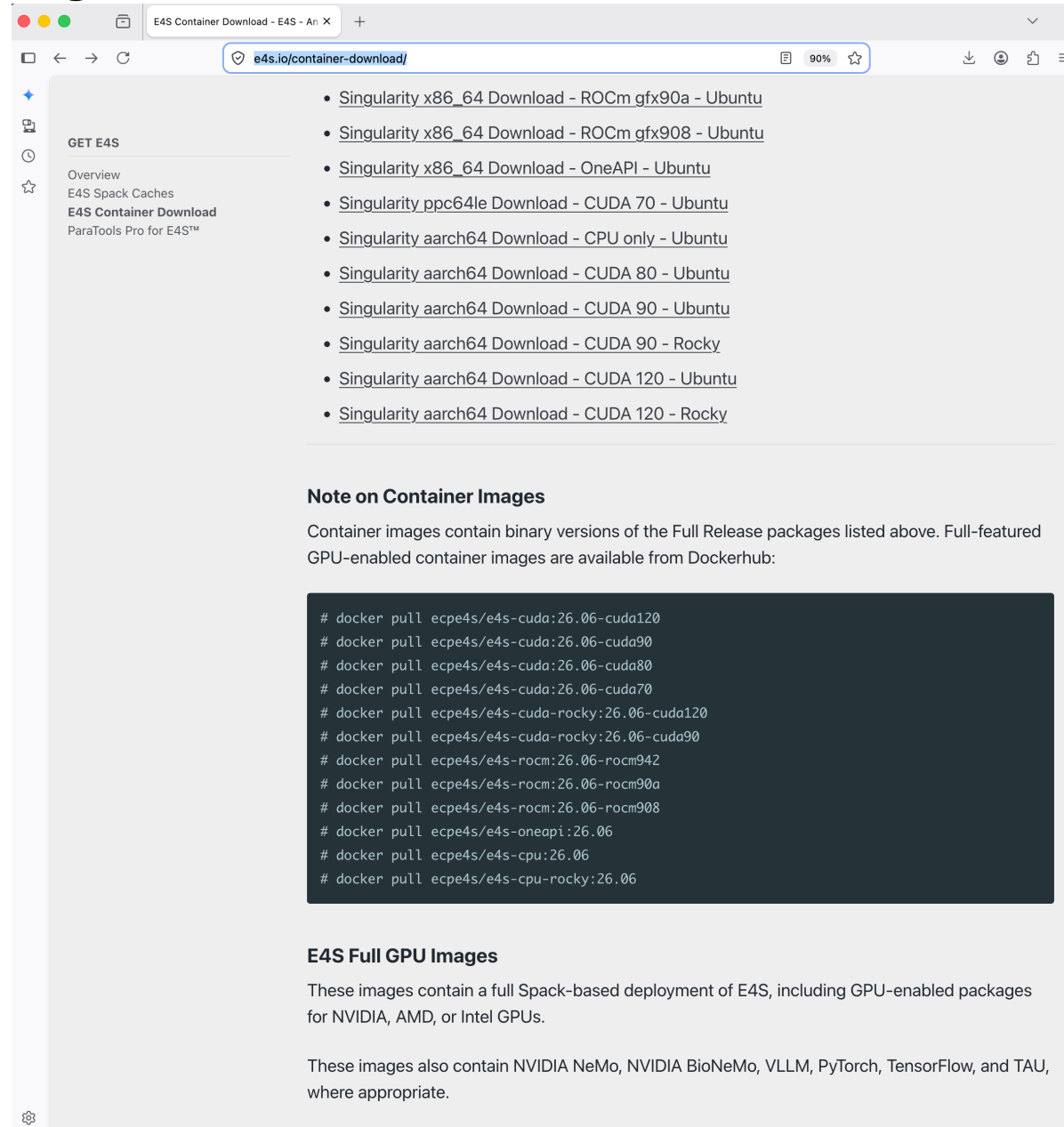
Docker images are available on the [E4S Docker Hub](#). Please see the [E4S 26.06 Release Notes](#).

Container Releases

- [Docker Downloads - Minimal](#)
- [Docker Downloads - CPU only - Ubuntu](#)
- [Docker Downloads - CPU only - Rocky](#)
- [Docker Downloads - CUDA - Ubuntu](#)
- [Docker Downloads - CUDA - Rocky](#)
- [Docker Downloads - ROCm](#)
- [Docker Downloads - OneAPI](#)
- [Singularity x86_64 Download - Minimal](#)
- [Singularity x86_64 Download - CPU only - Ubuntu](#)
- [Singularity x86_64 Download - CPU only - Rocky](#)
- [Singularity x86_64 Download - CUDA 80 - Ubuntu](#)
- [Singularity x86_64 Download - CUDA 90 - Ubuntu](#)
- [Singularity x86_64 Download - CUDA 90 - Rocky](#)
- [Singularity x86_64 Download - CUDA 120 - Ubuntu](#)
- [Singularity x86_64 Download - CUDA 120 - Rocky](#)

- Docker and Singularity
- ARM64 (aarch64), x86_64, and ppc64le
- Support for GPUS:
 - NVIDIA
 - AMD
 - Intel
- GPU Runtimes:
 - CUDA 13.2
 - ROCm 7.2.3
 - oneAPI 2026.0
- Languages:
 - C/C++/Fortran
 - Python
 - Rust
 - Julia
 - Chapel ...
- OSes:
 - Rocky Linux 9.7
 - Ubuntu 24.04 LTS

E4S container images available on DockerHub



The screenshot shows a web browser window with the URL `e4s.io/container-download/`. The page has a sidebar on the left with a navigation menu containing: GET E4S, Overview, E4S Spack Caches, E4S Container Download (which is highlighted), and ParaTools Pro for E4S™. The main content area features a list of ten download links for Singularity containers, each with a specific architecture, GPU configuration, and operating system. Below this list is a section titled "Note on Container Images" which explains that the container images contain binary versions of the Full Release packages and that GPU-enabled images are available from Dockerhub. A dark-themed code block follows, displaying ten Docker pull commands for various E4S container images. At the bottom, there is a section titled "E4S Full GPU Images" which describes these images as containing a full Spack-based deployment of E4S with GPU support for NVIDIA, AMD, and Intel GPUs, and lists additional software like NVIDIA NeMo, VLLM, PyTorch, TensorFlow, and TAU.

GET E4S

- Overview
- E4S Spack Caches
- E4S Container Download**
- ParaTools Pro for E4S™

- [Singularity x86_64 Download - ROCm gfx90a - Ubuntu](#)
- [Singularity x86_64 Download - ROCm gfx908 - Ubuntu](#)
- [Singularity x86_64 Download - OneAPI - Ubuntu](#)
- [Singularity ppc64le Download - CUDA 70 - Ubuntu](#)
- [Singularity aarch64 Download - CPU only - Ubuntu](#)
- [Singularity aarch64 Download - CUDA 80 - Ubuntu](#)
- [Singularity aarch64 Download - CUDA 90 - Ubuntu](#)
- [Singularity aarch64 Download - CUDA 90 - Rocky](#)
- [Singularity aarch64 Download - CUDA 120 - Ubuntu](#)
- [Singularity aarch64 Download - CUDA 120 - Rocky](#)

Note on Container Images

Container images contain binary versions of the Full Release packages listed above. Full-featured GPU-enabled container images are available from Dockerhub:

```
# docker pull ecpe4s/e4s-cuda:26.06-cuda120
# docker pull ecpe4s/e4s-cuda:26.06-cuda90
# docker pull ecpe4s/e4s-cuda:26.06-cuda80
# docker pull ecpe4s/e4s-cuda:26.06-cuda70
# docker pull ecpe4s/e4s-cuda-rocky:26.06-cuda120
# docker pull ecpe4s/e4s-cuda-rocky:26.06-cuda90
# docker pull ecpe4s/e4s-rocm:26.06-rocm942
# docker pull ecpe4s/e4s-rocm:26.06-rocm90a
# docker pull ecpe4s/e4s-rocm:26.06-rocm908
# docker pull ecpe4s/e4s-oneapi:26.06
# docker pull ecpe4s/e4s-cpu:26.06
# docker pull ecpe4s/e4s-cpu-rocky:26.06
```

E4S Full GPU Images

These images contain a full Spack-based deployment of E4S, including GPU-enabled packages for NVIDIA, AMD, or Intel GPUs.

These images also contain NVIDIA NeMo, NVIDIA BioNeMo, VLLM, PyTorch, TensorFlow, and TAU, where appropriate.

E4S: CPU Rocky Linux 9.8 x86_64 Singularity image

```
Singularity> spack find -x
-- linux-rocky9-x86_64_v3 / %c,cxx,fortran=gcc@13.3.1 -----
adios@1.13.1      cp2k@2026.1      libceed@0.12.0      quantum-espresso@7.5  tau@2.35.2
adios2@2.12.1     darshan-runtime@3.4.7 openmpi@5.0.10      rempi@1.1.0          trilinos@16.2.0
alquimia@1.1.0    darshan-util@3.4.7  papi@7.2.0          scr@3.1.0            variorum@0.8.0
amrex@26.05       datatransferkit@3.1.1 parallel-netcdf@1.14.1 slate@2025.05.28      wps@4.5
ascent@0.9.5      globalarrays@5.8.2  petsc@3.25.1        slepc@3.25.1         xyce@7.10.0
axom@0.14.0       gptune@5.0.0        phist@1.12.1        strumpack@8.0.0      zfp@1.0.1
butterflypack@3.2.0 h5bench@1.4         plumed@2.10.0       sundials@7.7.0
conduit@0.9.6     heffte@2.4.1        precise@3.4.0       superlu-dist@9.2.1

-- linux-rocky9-x86_64_v3 / %c,cxx=gcc@13.3.1 -----
boost@1.90.0      ginkgo@1.11.0      legion@26.03.0      omega-h@11.0.0-scorec sz@2.1.12.5
bricks@2023.08.25 glvis@4.4           libcatayst@2.1.0    openfoam@2512        sz3@3.2.0
cabana@0.7.0      gmp@6.3.0          libpressio@0.99.4   openpmd-api@0.17.0   turbine@1.3.0
caliper@2.14.0    gotcha@1.0.8        libunwind@1.8.3     papyrus@1.0.2        umap@2.1.1
chai@2025.12.0    gromacs@2026.1      mercury@2.4.1       pdt@3.25.2           umpire@2025.12.0
chapel@2.8.0      hdf5-vol-cache@v1.1 metall@0.30          pruners-ninja@1.0.2  upcxx@2025.10.0
dyninst@13.0.1    hdf5-vol-log@1.4.0  mgard@1.6.0         pumi@2.2.9           veloc@1.7
faodel@1.2108.1   hpctoolkit@2026.0.1 mpiutils@0.12        qthreads@1.18        vtk-m@2.3.0
fftx@1.2.0        lammps@20250722.3   nccmp@1.9.1.0       raja@2025.12.2       warpx@26.04
gasnet@2025.8.0   lbann@0.104         nco@5.3.4           swig@4.0.2-fortran

-- linux-rocky9-x86_64_v3 / %c,fortran=gcc@13.3.1 -----
fpm@0.13.0        libquo@1.4          nekbone@17.0         netlib-scalapack@2.2.3 py-libensemble@1.5.0 wannier90@3.1.0
hdf5@1.14.6       nek5000@19.0        netcdf-fortran@4.6.2 plasma@24.8.7         superlu@7.0.1      wrf@4.7.1

-- linux-rocky9-x86_64_v3 / %c=gcc@13.3.1 -----
aml@0.2.1         charliecloud@0.44    hdf5-vol-async@1.7   libnrm@0.1.0         py-h5py@3.16.0
argobots@1.2      flux-core@0.81.0     hypra@3.1.0          parsec@4.0.2411      py-petsc4py@3.25.1

-- linux-rocky9-x86_64_v3 / %c,cxx,fortran=gcc@13.3.1 -----
tasmanian@8.1

-- linux-rocky9-x86_64_v3 / %c,cxx=gcc@13.3.1 -----
arborx@2.1        hpx@1.11.0          kokkos-kernels@5.1.1 loki@0.1.7           mpark-variant@1.4.0
flit@2.1.0        kokkos@5.1.1        laghos@3.1           mfem@4.9.0

-- linux-rocky9-x86_64_v3 / no compilers -----
e4s-alc@1.0.3     exaworks@0.1.0      nrm@0.1.0           py-jupyterhub@1.4.1
e4s-cl@1.0.9      mpich@4.3.2         py-cinemasci@1.7.9  stc@0.9.0
=> 127 installed packages
```

E4S: CUDA Rocky Linux 9.8 x86_64 Singularity image

```
Singularity> spack find -x
-- linux-rocky9-x86_64_v3 / %c,cxx,fortran=gcc@13.3.1 -----
adios2@2.12.1  cp2k@2025.2      magma@2.10.0  slate@2025.05.28  sundials@7.7.0      trilinos@16.2.0
amrex@26.05    heffte@2.4.1    papi@7.2.0   slepc@3.25.1     superlu-dist@9.2.1  zfp@1.0.1
axom@0.14.0    libceed@0.12.0  petsc@3.25.1  strumpack@8.0.0   tau@2.35.2

-- linux-rocky9-x86_64_v3 / %c,cxx=gcc@13.3.1 -----
bricks@2023.08.25  chapel@2.8.0    gromacs@2026.1    legion@26.03.0      raja@2025.12.2
cabana@0.7.0       cusz@0.14.0    hpctoolkit@2026.0.1  libpressio@0.99.4    umpire@2025.12.0
caliper@2.14.0     fftx@1.2.0     hypre@3.1.0       mgard@1.6.0         upcxx@2025.10.0
chai@2025.12.0     ginkgo@1.11.0  lammers@20250722.3  omega-h@11.0.0-scorec  vtk-m@2.3.0

-- linux-rocky9-x86_64_v3 / %c=gcc@13.3.1 -----
flux-core@0.81.0  parsec@4.0.2411

-- linux-rocky9-x86_64_v3 / %cxx,fortran=gcc@13.3.1 -----
tasmanian@8.1

-- linux-rocky9-x86_64_v3 / %cxx=gcc@13.3.1 -----
arborx@2.1  hpx@1.11.0  kokkos@5.1.1  kokkos-kernels@5.1.1  mfem@4.9.0

-- linux-rocky9-x86_64_v3 / no compilers -----
mpich@4.3.2
==> 46 installed packages
Singularity> █
```

E4S: Expanded, curated collection of Python based tools for AI

```
Singularity> python
Python 3.12.12 (main, Mar  2 2026, 16:39:05) [GCC 13.3.1 20240611 (Red Hat 13.3.1-2)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> import transformers
>>> import vllm
>>> import jax
>>> import cupy
>>> import triton
>>> import flash_attn
>>> import transformer_engine
>>> import accelerate
>>> import peft
>>> import sglang
>>> import ray
>>> import nemo_run
>>> import langchain
>>> import langgraph
>>> import bionemo
>>> import a2a
>>> import mcp
>>> import pandas
>>> import mlflow
>>> import openai
>>> import anthropic
>>> import google
>>> import huggingface_hub
>>> import tokenizers
>>> import mpi4py
>>> import s3fs
>>> import jupyter
>>> import marimo
>>> import tritonclient

>>> import ray
>>> import megatron
>>> import nemo_run
>>> import datasets
>>> import gradio
>>> import geopandas
>>> import polars
>>> import numpy
>>> import scipy
>>> import seaborn
>>> import leptonai
>>> import lightning
>>> import tensorboard
>>> import outlines
```

E4S: Intel oneAPI 2026.0 Singularity image supports Intel GPUs

```
$ singularity run --nv e4s-oneapi-x86_64-26-06-1781565945.sif
Singularity> spack find -x
-- linux-ubuntu24.04-x86_64_v3 / %c,cxx,fortran=gcc@13.3.0 -----
adios2@2.12.1  gptune@5.0.0  openmpi@5.0.10  scr@3.1.0

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx,fortran=oneapi@2026.0.0 -
adios@1.13.1  axom@0.14.0      darshan-util@3.4.7  heffte@2.4.1      phist@1.12.1      strumpack@8.0.0      tau@2.35.2      zfp@1.0.1
amrex@26.05   butterflypack@3.2.0  datatransferkit@3.1.1  libceed@0.12.0    plumed@2.10.0     sundials@7.7.0      trilinos@16.2.0
amrex@26.05   conduit@0.9.6       globalarrays@5.8.2    papi@7.2.0        rempi@1.1.0       superlu-dist@9.2.1  unifyfs@2.0
ascent@0.9.5  darshan-runtime@3.4.7  h5bench@1.4          parallel-netcdf@1.14.1  slate@2025.05.28  tau@2.35.2          variorum@0.8.0

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx=gcc@13.3.0 %fortran=oneapi@2026.0.0
quantum-espresso@7.5

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx=gcc@13.3.0 -----
openfoam@2512  openpmd-api@0.17.0  swig@4.0.2-fortran

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx=oneapi@2026.0.0 -----
bolt@2.0      cabana@0.7.0      gasnet@2025.8.0  gromacs@2026.1  legion@26.03.0  metall@0.30      omega-h@11.0.0-scorec  pumi@2.2.9      sz3@3.2.0      upcxx@2025.10.0
boost@1.90.0  caliper@2.14.0   ginkgo@1.11.0   hdf5-vol-log@1.4.0  libcatapult@2.1.0  mgard@1.6.0     papyrus@1.0.2      qthreads@1.18  turbine@1.3.0  upcxx@2025.10.0
bricks@2023.08.25  chai@2025.12.0  gmp@6.3.0      hpctoolkit@2026.0.1  libunwind@1.8.3  mpi@1.0.12      pdt@3.25.2      raja@2025.12.2  umap@2.1.1     veloc@1.7
cabana@0.7.0  faodel@1.2108.1  gotcha@1.0.8   hpctoolkit@2026.0.1  mercury@2.4.1    nccmp@1.9.1.0   pruners-ninja@1.0.2  sz@2.1.12.5    umpire@2025.12.0  vtk-m@2.3.0

-- linux-ubuntu24.04-x86_64_v3 / %c,fortran=gcc@13.3.0 -----
py-libensemble@1.5.0

-- linux-ubuntu24.04-x86_64_v3 / %c,fortran=oneapi@2026.0.0 ----
hdf5@1.14.6  libquo@1.4  nekbone@17.0  netcdf-fortran@4.6.2  netlib-scalapack@2.2.3  superlu@7.0.1  wannier90@3.1.0

-- linux-ubuntu24.04-x86_64_v3 / %c=gcc@13.3.0 -----
charliecloud@0.44  parsec@4.0.2411

-- linux-ubuntu24.04-x86_64_v3 / %c=oneapi@2026.0.0 -----
aml@0.2.1  aml@0.2.1  argobots@1.2  flux-core@0.81.0  hdf5-vol-async@1.7  hypre@3.1.0  libnm@0.1.0

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx,fortran=oneapi@2026.0.0 ---
fortrilinos@2.3.0  tasmanian@8.1

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx=gcc@13.3.0 -----
hpx@1.11.0

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx=oneapi@2026.0.0 -----
arborx@2.1  arborx@2.1  flit@2.1.0  kokkos@5.1.1  kokkos@5.1.1  kokkos-kernels@5.1.1  laghos@3.1  loki@0.1.7  mfem@4.9.0  mpark-variant@1.4.0

-- linux-ubuntu24.04-x86_64_v3 / no compilers -----
e4s-alc@1.0.3  e4s-cl@1.0.9  exaworks@0.1.0  intel-oneapi-mpi@2021.18.0  nrm@0.1.0  py-jupyterhub@1.4.1  stc@0.9.0
=> 114 installed packages
Singularity> █
```

E4S: AMD ROCm 7.2.3 Singularity image

```
Singularity> ls /opt/rocm-7.2.3/bin/hipcc
/opt/rocm-7.2.3/bin/hipcc
Singularity> spack find -x +rocm
-- linux-ubuntu24.04-x86_64_v3 / %c,cxx,fortran=gcc@13.3.0 -----
amrex@26.05    libceed@0.12.0    papi@7.2.0    slate@2025.05.28    sundials@7.7.0    tau@2.35.2
heffte@2.4.1   magma@2.10.0    petsc@3.25.1  slepc@3.25.1    superlu-dist@9.2.1  trilinos@16.2.0

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx=gcc@13.3.0 -----
caliper@2.14.0  gasnet@2025.8.0    hypre@3.1.0    raja@2025.12.2    upcxx@2025.10.0
chai@2025.12.0  hpctoolkit@2026.0.1  legion@26.03.0  umpire@2025.12.0  vtk-m@2.3.0

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx,fortran=gcc@13.3.0 -----
tasmanian@8.1

-- linux-ubuntu24.04-x86_64_v3 / %c,cxx=gcc@13.3.0 -----
kokkos@5.1.1  mfem@4.9.0
==> 25 installed packages
```


E4S: CPU aarch64 Singularity image (macOS M-series, NVIDIA Grace)

```
Singularity> spack find -x
-- linux-ubuntu24.04-aarch64 / %c,cxx,fortran=gcc@13.3.0 -----
adios@1.13.1      axom@0.14.0      darshan-util@3.4.7  heffte@2.4.1      paraview@6.1.0    quantum-espresso@7.5  strumpack@8.0.0    wps@4.5
adios2@2.12.1    butterflypack@3.2.0  datatransferkit@3.1.1  libceed@0.12.0    petsc@3.25.1     rempi@1.1.0          sundials@7.7.0    zfp@1.0.1
alquimia@1.1.0   conduit@0.9.6      globalarrays@5.8.2   openmpi@5.0.10    phist@1.12.1     scr@3.1.0            superlu-dist@9.2.1
amrex@26.05      cp2k@2026.1        gptune@5.0.0         papi@7.2.0        plumed@2.10.0    slate@2025.05.28      tau@2.35.2
ascent@0.9.5     darshan-runtime@3.4.7  h5bench@1.4         parallel-netcdf@1.14.1  precice@3.4.0    slepc@3.25.1         trilinos@16.2.0

-- linux-ubuntu24.04-aarch64 / %c,cxx=gcc@13.3.0 -----
boost@1.90.0      faodel@1.2108.1    gotcha@1.0.8        lbann@0.104        metall@0.30       openfoam@2512         qthreads@1.18     umap@2.1.1
cabana@0.7.0      fftx@1.2.0          gromacs@2026.1       legion@26.03.0     mgard@1.6.0       openpmd-api@0.17.0    raja@2025.12.2    umpire@2025.12.0
caliper@2.14.0    gasnet@2025.8.0     hdf5-vol-cache@v1.1  libcatylist@2.1.0  mpifileutils@0.12  papyrus@1.0.2         swig@4.0.2-fortran  upcxx@2025.10.0
chai@2025.12.0    ginkgo@1.11.0       hdf5-vol-log@1.4.0   libpressio@0.99.4  nccmp@1.9.1.0     pdt@3.25.2           sz@2.1.12.5        veloc@1.7
chapel@2.8.0      glvis@4.4           hpctoolkit@2026.0.1  libunwind@1.7.2    nco@5.3.4          pruners-ninja@1.0.2  sz3@3.2.0          vtk-m@2.3.0
dyninst@13.0.1    gmp@6.3.0          lammps@20250722.3    mercury@2.4.1      omega-h@11.0.0-scorec  pumi@2.2.9          turbine@1.3.0      warpx@26.04

-- linux-ubuntu24.04-aarch64 / %c,fortran=gcc@13.3.0 -----
fpm@0.13.0        libquo@1.4          nekbone@17.0         netlib-scalapack@2.2.3  py-libensemble@1.5.0  wannier90@3.1.0
hdf5@1.14.6       nek5000@19.0        netcdf-fortran@4.6.2  plasma@24.8.7         superlu@7.0.1         wrf@4.7.1

-- linux-ubuntu24.04-aarch64 / %c=gcc@13.3.0 -----
aml@0.2.1         charliecloud@0.44    hdf5-vol-async@1.7   libnrm@0.1.0         py-h5py@3.16.0
argobots@1.2      flux-core@0.81.0     hypre@3.1.0          parsec@4.0.2411      py-petsc4py@3.25.1

-- linux-ubuntu24.04-aarch64 / %c,cxx,fortran=gcc@13.3.0 -----
tasmanian@8.1

-- linux-ubuntu24.04-aarch64 / %c,cxx=gcc@13.3.0 -----
arborx@2.1        flit@2.1.0          hpx@1.11.0           kokkos@5.1.1         kokkos-kernels@5.1.1  laghos@3.1           loki@0.1.7         mfem@4.9.0         mpark-variant@1.4.0

-- linux-ubuntu24.04-aarch64 / no compilers -----
e4s-alc@1.0.3     e4s-cl@1.0.9        exaworks@0.1.0       mpich@4.3.2          nrm@0.1.0            py-cinemasci@1.7.9  py-jupyterhub@1.4.1  stc@0.9.0
==> 125 installed packages
```

E4S: CUDA aarch64 Ubuntu Singularity image

```
Singularity> spack find -x
-- linux-ubuntu24.04-aarch64 / %c,cxx,fortran=gcc@13.3.0 -----
adios2@2.12.1  axom@0.14.0  heffte@2.4.1  magma@2.10.0  petsc@3.25.1  slepc@3.25.1  sundials@7.7.0  tau@2.35.2  zfp@1.0.1
amrex@26.05  cp2k@2025.2  libceed@0.12.0  paraview@6.1.0  slate@2025.05.28  strumpack@8.0.0  superlu-dist@9.2.1  trilinos@16.2.0

-- linux-ubuntu24.04-aarch64 / %c,cxx=gcc@13.3.0 -----
cabana@0.7.0  cusz@0.14.0  gromacs@2026.1  lammps@20250722.3  mgard@1.6.0  umpire@2025.12.0
caliper@2.14.0  fftx@1.2.0  hpctoolkit@2026.0.1  legion@26.03.0  omega-h@11.0.0-scorec  upcxx@2025.10.0
chai@2025.12.0  ginkgo@1.11.0  hypre@3.1.0  libpressio@0.99.4  raja@2025.12.2  vtk-m@2.3.0

-- linux-ubuntu24.04-aarch64 / %c=gcc@13.3.0 -----
flux-core@0.81.0  parsec@4.0.2411

-- linux-ubuntu24.04-aarch64 / %cxx,fortran=gcc@13.3.0 -----
tasmanian@8.1

-- linux-ubuntu24.04-aarch64 / %cxx=gcc@13.3.0 -----
arborx@2.1  hpx@1.11.0  kokkos@5.1.1  kokkos-kernels@5.1.1  mfem@4.9.0

-- linux-ubuntu24.04-aarch64 / no compilers -----
mpich@4.3.2
==> 44 installed packages
```

E4S: CPU aarch64 Rocky Linux 9.7 Singularity image

```
Singularity> spack find -x
-- linux-rocky9-aarch64 / %c,cxx,fortran=gcc@13.3.1 -----
adios@1.13.1      butterflypack@3.2.0    globalarrays@5.8.2    papi@7.2.0          quantum-espresso@7.5    sundials@7.7.0
adios2@2.12.1     conduit@0.9.6          gptune@5.0.0         parallel-netcdf@1.14.1 rempi@1.1.0            superlu-dist@9.2.1
alquimia@1.1.0    cp2k@2026.1           h5bench@1.4          petsc@3.25.1        scr@3.1.0              tau@2.35.2
amrex@26.05       darshan-runtime@3.4.7  heffte@2.4.1         phist@1.12.1        slate@2025.05.28       trilinos@16.2.0
ascent@0.9.5     darshan-util@3.4.7    libceed@0.12.0       plumed@2.10.0       slepc@3.25.1           wps@4.5
axom@0.14.0       datatransferkit@3.1.1 openmpi@5.0.10       precice@3.4.0       strumpack@8.0.0        zfp@1.0.1

-- linux-rocky9-aarch64 / %c,cxx=gcc@13.3.1 -----
boost@1.90.0      faodel@1.2108.1        gotcha@1.0.8          lbann@0.104          metall@0.30            openfoam@2512      qthreads@1.18      umap@2.1.1
cabana@0.7.0      fftx@1.2.0             gromacs@2026.1        legion@26.03.0       mgard@1.6.0            openpmd-api@0.17.0 raja@2025.12.2     umpire@2025.12.0
caliper@2.14.0    gasnet@2025.8.0        hdf5-vol-cache@v1.1  libcatalyst@2.1.0    mpiutils@0.12          papyrus@1.0.2     swig@4.0.2-fortran upcxx@2025.10.0
chai@2025.12.0    ginkgo@1.11.0          hdf5-vol-log@1.4.0   libpressio@0.99.4    nccmp@1.9.1.0          pdt@3.25.2        sz@2.1.12.5       veloc@1.7
chapel@2.8.0      glvis@4.4              hpctoolkit@2026.0.1  libunwind@1.7.2     nco@5.3.4              pruners-ninja@1.0.2 sz3@3.2.0          vtk-m@2.3.0
dyninst@13.0.1    gmp@6.3.0              lammps@20250722.3    mercury@2.4.1       omega-h@11.0.0-scorec  pumi@2.2.9        turbine@1.3.0     warpx@26.04

-- linux-rocky9-aarch64 / %c,fortran=gcc@13.3.1 -----
fpm@0.13.0       libquo@1.4             nekbone@17.0          netlib-scalapack@2.2.3 py-libensemble@1.5.0  wannier90@3.1.0
hdf5@1.14.6     nek5000@19.0           netcdf-fortran@4.6.2  plasma@24.8.7        superlu@7.0.1          wrf@4.7.1

-- linux-rocky9-aarch64 / %c=gcc@13.3.1 -----
aml@0.2.1        charliecloud@0.44      hdf5-vol-async@1.7   libnrm@0.1.0         py-h5py@3.16.0
argobots@1.2     flux-core@0.81.0       hypre@3.1.0          parsec@4.0.2411     py-petsc4py@3.25.1

-- linux-rocky9-aarch64 / %c,cxx,fortran=gcc@13.3.1 -----
tasmanian@8.1

-- linux-rocky9-aarch64 / %c,cxx=gcc@13.3.1 -----
arborx@2.1       flit@2.1.0             hpx@1.11.0           kokkos@5.1.1         kokkos-kernels@5.1.1  laghos@3.1         loki@0.1.7         mfem@4.9.0         mpark-variant@1.4.0

-- linux-rocky9-aarch64 / no compilers -----
e4s-alc@1.0.3    e4s-cl@1.0.9           exaworks@0.1.0       mpich@4.3.2          nrm@0.1.0             py-cinemasci@1.7.9 py-jupyterhub@1.4.1 stc@0.9.0
==> 124 installed packages
```

E4S: CUDA aarch64 Rocky Linux 9.7 Singularity image

```
Singularity> spack find -x
-- linux-rocky9-aarch64 / %c,cxx,fortran=gcc@13.3.1 -----
adios2@2.12.1  axom@0.14.0  heffte@2.4.1    magma@2.10.0  slate@2025.05.28  strumpack@8.0.0  superlu-dist@9.2.1  trilinos@16.2.0
amrex@26.05    cp2k@2025.2  libceed@0.12.0  petsc@3.25.1  slepc@3.25.1     sundials@7.7.0  tau@2.35.2         zfp@1.0.1

-- linux-rocky9-aarch64 / %c,cxx=gcc@13.3.1 -----
cabana@0.7.0   cusz@0.14.0   gromacs@2026.1   lammps@20250722.3  mgard@1.6.0       umpire@2025.12.0
caliper@2.14.0  fftx@1.2.0    hpctoolkit@2026.0.1  legion@26.03.0    omega-h@11.0.0-scorec  upcxx@2025.10.0
chai@2025.12.0  ginkgo@1.11.0  hypre@3.1.0      libpressio@0.99.4  raja@2025.12.2     vtk-m@2.3.0

-- linux-rocky9-aarch64 / %c=gcc@13.3.1 -----
flux-core@0.81.0  parsec@4.0.2411

-- linux-rocky9-aarch64 / %cxx,fortran=gcc@13.3.1 -----
tasmanian@8.1

-- linux-rocky9-aarch64 / %cxx=gcc@13.3.1 -----
arborx@2.1  hpx@1.11.0  kokkos@5.1.1  kokkos-kernels@5.1.1  mfem@4.9.0

-- linux-rocky9-aarch64 / no compilers -----
mpich@4.3.2
==> 43 installed packages
```

E4S: CPU IBM ppc64le Ubuntu Singularity image

```
Singularity> spack find -x
-- linux-ubuntu20.04-ppc64le / %c,cxx,fortran=gcc@9.4.0 -----
adios@1.13.1      axom@0.14.0      datatransferkit@3.1.1  nwchem@7.3.1      phist@1.12.1      slate@2025.05.28      tau@2.35.2
adios2@2.12.1     butterflypack@3.2.0  globalarrays@5.8.2    openmpi@5.0.10     plumed@2.10.0     slepc@3.25.1          trilinos@16.2.0
alquimia@1.1.0    conduit@0.9.6      h5bench@1.4           papi@7.2.0         quantum-espresso@7.5  strumpack@8.0.0       unifyfs@2.0
amrex@25.05       darshan-runtime@3.4.7  heffte@2.4.1          parallel-netcdf@1.14.1  rempi@1.1.0        sundials@7.7.0        wps@4.5
ascent@0.9.5      darshan-util@3.4.7    libceed@0.12.0        petsc@3.25.1       scr@3.1.0          superlu-dist@9.2.1    zfp@1.0.1

-- linux-ubuntu20.04-ppc64le / %c,cxx=gcc@9.4.0 -----
bolt@2.0          faodel@1.2108.1     hdf5-vol-cache@v1.1   libpressio@0.99.4   nco@5.3.4          pumi@2.2.9            umap@2.1.1
boost@1.90.0      fftx@1.2.0          hdf5-vol-log@1.4.0    libunwind@1.7.2     omega-h@11.0.0-scorec  qthreads@1.18         umpire@2025.12.0
bricks@2023.08.25  gasnet@2025.8.0     hpctoolkit@2024.01.1  mercury@2.4.1       openfoam@2512         raja@2025.12.2        upcxx@2025.10.0
cabana@0.7.0      ginkgo@1.11.0       lammps@20250722.3     metall@0.30         openpmd-api@0.17.0    swig@4.0.2-fortran    veloc@1.7
caliper@2.14.0    glvis@4.4           lbann@0.104           mgard@1.6.0         papyrus@1.0.2         sz@2.1.12.5           vtk-m@2.3.0
chai@2025.12.0    gmp@6.3.0           legion@26.03.0        mpifileutils@0.12   pdt@3.25.2           sz3@3.2.0             warpx@25.04
dyninst@13.0.1    gotcha@1.0.8        libcatalyst@2.1.0     nccmp@1.9.1.0       pruners-ninja@1.0.2   turbine@1.3.0

-- linux-ubuntu20.04-ppc64le / %c,fortran=gcc@9.4.0 -----
fpm@0.13.0        libquo@1.4          nekbone@17.0          netlib-scalapack@2.2.3  py-libensemble@1.5.0  wannier90@3.1.0
hdf5@1.14.6       nek5000@19.0        netcdf-fortran@4.6.2  plasma@24.8.7         superlu@7.0.1         wrf@4.7.1

-- linux-ubuntu20.04-ppc64le / %c=gcc@9.4.0 -----
aml@0.2.1         charliecloud@0.44    hdf5-vol-async@1.7    libnrm@0.1.0         py-h5py@3.16.0
argobots@1.2      flux-core@0.81.0     hypr@3.1.0           parsec@4.0.2411      py-petsc4py@3.25.1

-- linux-ubuntu20.04-ppc64le / %c,cxx,fortran=gcc@9.4.0 -----
fortrilinos@2.3.0  tasmanian@8.1

-- linux-ubuntu20.04-ppc64le / %c,cxx=gcc@9.4.0 -----
arborx@2.1        flit@2.1.0          hpx@1.11.0            kokkos@4.7.04        kokkos-kernels@4.7.04  laghos@3.1            loki@0.1.7          mfem@4.9.0          mpark-variant@1.4.0

-- linux-ubuntu20.04-ppc64le / no compilers -----
e4s-alc@1.0.3     e4s-cl@1.0.9        exaworks@0.1.0        mpich@4.3.2          nrm@0.1.0             py-jupyterhub@1.4.1  stc@0.9.0

==> 123 installed packages
```


E4S: CUDA IBM ppc64le Ubuntu Singularity image

```
Singularity> spack find -x
-- linux-ubuntu20.04-ppc64le / %c,cxx,fortran=gcc@9.4.0 -----
adios2@2.12.1 heffte@2.4.1 magma@2.10.0 slepc@3.25.1 sundials@7.7.0 tau@2.35.2
amrex@25.05 libceed@0.12.0 petsc@3.25.1 strumpack@8.0.0 superlu-dist@9.2.1 zfp@1.0.1

-- linux-ubuntu20.04-ppc64le / %c,cxx=gcc@9.4.0 -----
bricks@2023.08.25 caliper@2.14.0 fftx@1.2.0 hpctoolkit@2024.01.1 lammmps@20250722.3 omega-h@11.0.0-scorec upcxx@2025.10.0
cabana@0.7.0 chai@2025.12.0 ginkgo@1.11.0 hypre@3.1.0 mgard@1.6.0 umpire@2025.12.0 vtk-m@2.3.0

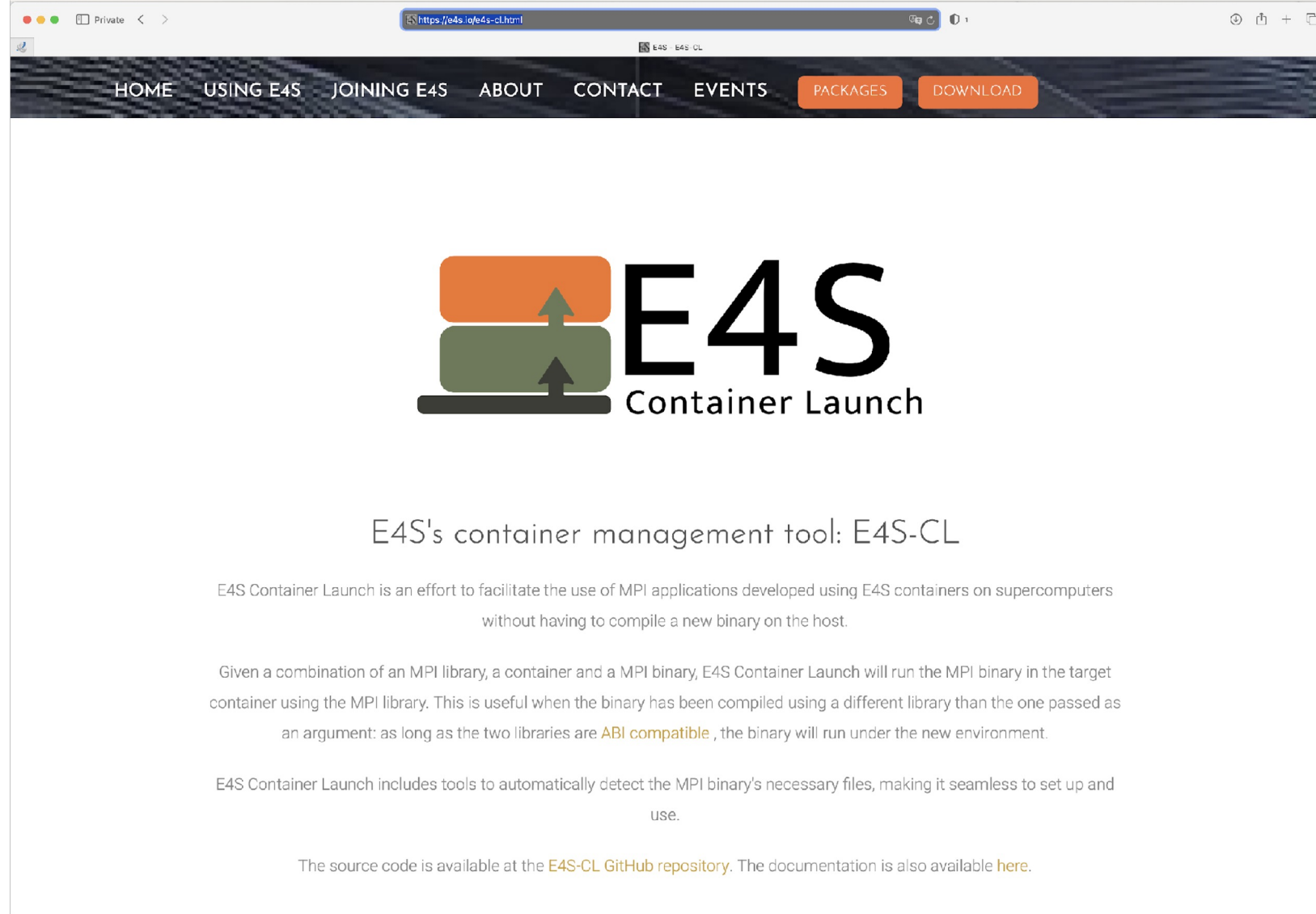
-- linux-ubuntu20.04-ppc64le / %c=gcc@9.4.0 -----
flux-core@0.81.0 parsec@4.0.2411

-- linux-ubuntu20.04-ppc64le / %cxx,fortran=gcc@9.4.0 -----
tasmanian@8.1

-- linux-ubuntu20.04-ppc64le / %cxx=gcc@9.4.0 -----
hpx@1.11.0 kokkos@4.7.04 kokkos-kernels@4.7.04 mfem@4.9.0

-- linux-ubuntu20.04-ppc64le / no compilers -----
mpich@4.3.2
==> 34 installed packages
Singularity> █
```


E4S Tools: e4s-cl: Container Launch tool for MPI applications



- Distribute your MPI application as a binary with an E4S image
- While deploying on a system substitute the embedded containerized MPI in application with the system/vendor MPI
- Use inter-node network interfaces efficiently for near native performance!

e4s-cl: A tool to simplify the launch of MPI jobs in E4S containers

- E4S containers support replacement of MPI libraries using MPICH ABI compatibility layer and Wi4MPI [CEA] for OpenMPI replacement.
- Applications binaries built using E4S can be launched with Singularity using MPI library substitution for efficient inter-node communications.
- e4s-cl is a new tool that simplifies the launch and MPI replacement.

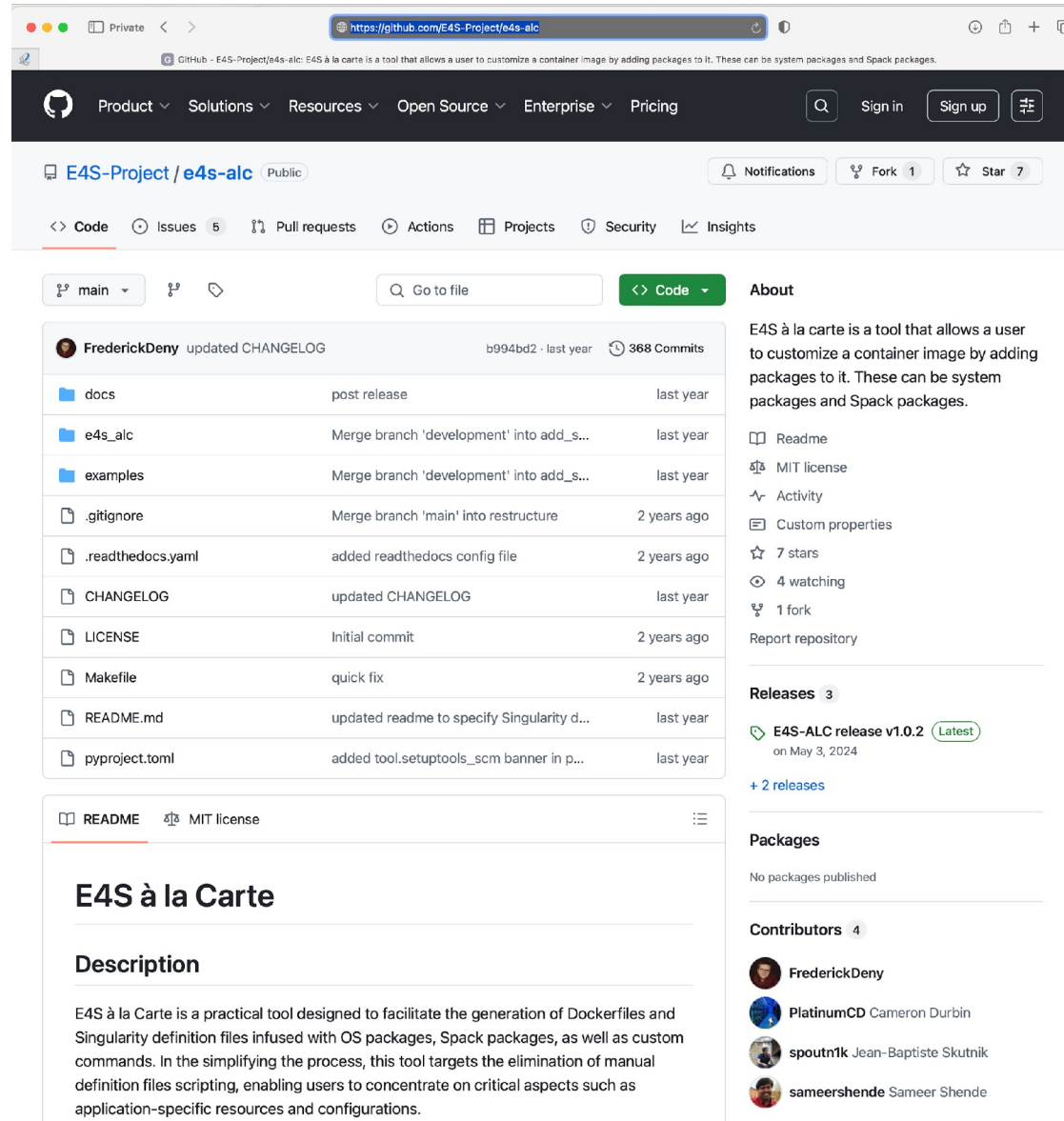
- e4s-cl init --backend [singularity|shifter|docker] --image <file> --source <startup_cmds.sh>
 - e4s-cl mpirun -np <N> <command>

- Usage:

```
% e4s-cl init --backend singularity --image ~/images/e4s-gpu-x86.sif --source ~/source.sh
% cat ~/source.sh
  . /spack/share/spack/setup-env.sh
  spack load trilinos+cuda cuda_arch=90
% e4s-cl mpirun -np 4 ./a.out
```



E4S Tools: E4S à la carte or e4s-alc: Customize container images



The screenshot displays the GitHub repository page for `E4S-Project/e4s-alc`. The repository is public and has 7 stars and 1 fork. The main branch is `main`. The repository structure includes:

- `docs`: post release (last year)
- `e4s_alc`: Merge branch 'development' into add_s... (last year)
- `examples`: Merge branch 'development' into add_s... (last year)
- `.gitignore`: Merge branch 'main' into restructure (2 years ago)
- `.readthedocs.yaml`: added readthedocs config file (2 years ago)
- `CHANGELOG`: updated CHANGELOG (last year)
- `LICENSE`: Initial commit (2 years ago)
- `Makefile`: quick fix (2 years ago)
- `README.md`: updated readme to specify Singularity d... (last year)
- `pyproject.toml`: added tool.setuptools_scm banner in p... (last year)

The `README` section is titled **E4S à la Carte** and includes a description:

Description

E4S à la Carte is a practical tool designed to facilitate the generation of Dockerfiles and Singularity definition files infused with OS packages, Spack packages, as well as custom commands. In the simplifying the process, this tool targets the elimination of manual definition files scripting, enabling users to concentrate on critical aspects such as application-specific resources and configurations.

The right sidebar shows the repository's activity, including 7 stars, 4 watching, 1 fork, and 3 releases. The latest release is **E4S-ALC release v1.0.2** (Latest) on May 3, 2024. There are 2 contributors listed: FrederickDeny, PlatinumCD, spoutn1k, and sameershende.

- Add new system packages
- Add new Spack packages
- Add new tarballs
- Customize the container image
- Start with a base image
- Add packages
- Create a new container image!

E4S Tools: e4s-chain-spack.sh to customize software stack

```
sameer@mothra:~$ ls ~/images
e4s-cuda80-x86_64-25.06.sif
sameer@mothra:~$ singularity run --nv ~/images/e4s-cuda80-x86_64-25.06.sif
Singularity> /etc/e4s/e4s-chain-spack.sh ~/spack
Cloning into '/home/sameer/spack'...
remote: Enumerating objects: 686113, done.
remote: Counting objects: 100% (976/976), done.
remote: Compressing objects: 100% (463/463), done.
remote: Total 686113 (delta 772), reused 518 (delta 510), pack-reused 685137 (from 3)
Receiving objects: 100% (686113/686113), 230.82 MiB | 37.06 MiB/s, done.
Resolving deltas: 100% (326280/326280), done.
```

Specify location of downstream
Spack installation directory

```
-----
Configuration SUCCESS!
```

```
Downstream: /home/sameer/spack
Upstream: /spack
```

```
To use the downstream Spack instance, run the following command in your shell:
. /home/sameer/spack/share/spack/setup-env.sh
-----
```

Source downstream Spack's
setup-env.sh

```
Singularity> . /home/sameer/spack/share/spack/setup-env.sh
```

```
Singularity> spack find valgrind
```

```
==> Error: No package matches the query: valgrind
```

```
Singularity> spack install valgrind
```

```
[+] /usr/local/mpich/install/mpich (external mpich-4.2.3-47excoypwhfmx57rfs6reouvnninugcf)
```

```
[+] /usr (external glibc-2.35-a7drdl4tlx4bu3mzhor75pskvd3pdot6)
```

```
[+] /spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/gcc-runtime-11.4.0-f63c77kavzjtpmnhucd2oyfaxagwjzla
```

```
[+] /spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/boost-1.86.0-6qkv24gbidwxhllgah6jrkym5ev2cng5
```

```
[+] /spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/gmake-4.4.1-qp5blvcyuzgzhqsrp2ew6ggq2nlos34b2
```

```
==> Installing valgrind-3.23.0-feuxx36lsqp7quzmhmo4opbiadwpsars [6/6]
```

```
==> No binary for valgrind-3.23.0-feuxx36lsqp7quzmhmo4opbiadwpsars found: installing from source
```

```
==> Fetching https://mirror.spack.io/_source-cache/archive/c5/c5c34a3380457b9b75606df890102e7df2c702b9420c2ebef9540f8b5d56264d.tar.bz2
```

```
==> Ran patch() for valgrind
```

```
==> valgrind: Executing phase: 'autoreconf'
```

```
==> valgrind: Executing phase: 'configure'
```

```
==> valgrind: Executing phase: 'build'
```

```
==> valgrind: Executing phase: 'install'
```

```
==> valgrind: Successfully installed valgrind-3.23.0-feuxx36lsqp7quzmhmo4opbiadwpsars
```

```
Stage: 3.78s. Autoreconf: 0.01s. Configure: 48.56s. Build: 37.71s. Install: 2.97s. Post-install: 0.60s. Total: 1m 33.97s
```

```
[+] /home/sameer/spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/valgrind-3.23.0-feuxx36lsqp7quzmhmo4opbiadwpsars
```

```
Singularity> spack load valgrind
```

```
Singularity> which valgrind
```

```
/home/sameer/spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/valgrind-3.23.0-feuxx36lsqp7quzmhmo4opbiadwpsars/bin/valgrind
```

Install a new Spack package
in downstream Spack directory

Load new package (valgrind)
using spack load

E4S Tools: e4s-chain-spack.sh to customize software stack

```
Singularity> which valgrind
/home/sameer/spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/valgrind-3.23.0-feuwx36lsqp7quzmhm04opbiadwpsars/bin/valgrind
Singularity> valgrind --help
usage: valgrind [options] prog-and-args

tool-selection option, with default in [ ]:
  --tool=<name>          use the Valgrind tool named <name> [memcheck]
                        available tools are:
                        memcheck cachegrind callgrind helgrind drd
                        massif dhat lackey none exp-bbv

basic user options for all Valgrind tools, with defaults in [ ]:
  -h --help              show this message
  --help-debug           show this message, plus debugging options
  --help-dyn-options     show the dynamically changeable options
  --version              show version
  -q --quiet             run silently; only print error msgs
  -v --verbose           be more verbose -- show misc extra info
  --trace-children=no|yes Valgrind-ise child processes (follow execve)? [no]
  --trace-children-skip=patt1,patt2,... specifies a list of executables
                        that --trace-children=yes should not trace into
  --trace-children-skip-by-arg=patt1,patt2,... same as --trace-children-skip=
                        but check the argv[] entries for children, rather
                        than the exe name, to make a follow/no-follow decision
  --child-silent-after-fork=no|yes omit child output between fork & exec? [no]
  --vgdb=no|yes|full     activate gdbserver? [yes]
                        full is slower but provides precise watchpoint/step
  --vgdb-error=<number>  invoke gdbserver after <number> errors [999999999]
                        to get started quickly, use --vgdb-error=0
                        and follow the on-screen directions
  --vgdb-stop-at=event1,event2,... invoke gdbserver for given events [none]
                        where event is one of:
                        startup exit abexit valgrindabexit all none
  --track-fds=no|yes|all track open file descriptors? [no]
                        all includes reporting stdin, stdout and stderr
  --time-stamp=no|yes    add timestamps to log messages? [no]
  --log-fd=<number>      log messages to file descriptor [2=stderr]
  --log-file=<file>      log messages to <file>
  --log-socket=ipaddr:port log messages to socket ipaddr:port
  --enable-debuginfod=no|yes query debuginfod servers for missing
                        debuginfo [yes]

user options for Valgrind tools that report errors:
  --xml=yes              emit error output in XML (some tools only)
  --xml-fd=<number>      XML output to file descriptor
  --xml-file=<file>      XML output to <file>
  --xml-socket=ipaddr:port XML output to socket ipaddr:port
  --xml-user-comment=STR copy STR verbatim into XML output
  --demangle=no|yes      automatically demangle C++ names? [yes]
  --num-callers=<number> show <number> callers in stack traces [12]
  --error-limit=no|yes   stop showing new errors if too many? [yes]
  --exit-on-first-error=no|yes exit code on the first error found? [no]
  --error-exitcode=<number> exit code to return if errors found [0=disable]
  --error-markers=<begin>,<end> add lines with begin/end markers before/after
                        each error output in plain text mode [none]
  --show-error-list=no|yes|all show detected errors list and
                        suppression counts at exit [no].
                        all means to also print suppressed errors.
  -s                    same as --show-error-list=yes
```

Downstream Spack's package is loaded in your environment

e4s-chain-spack.sh helps customize the software stack using upstream /spack (read-only in the container) for package dependencies while installing a new package in the downstream Spack in your writable home directory.

ParaTools Pro for E4S™

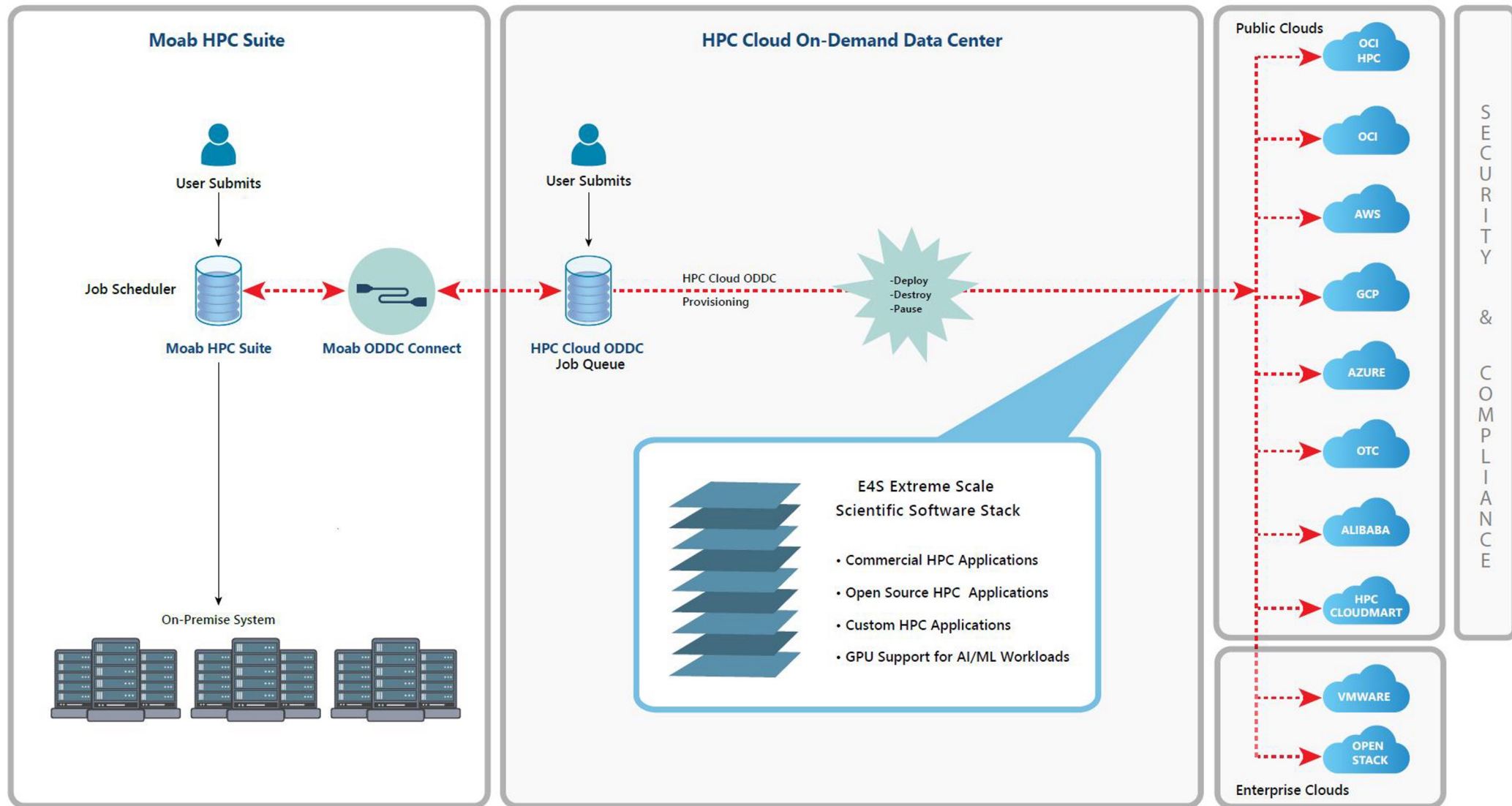
<https://paratoolspro.com>



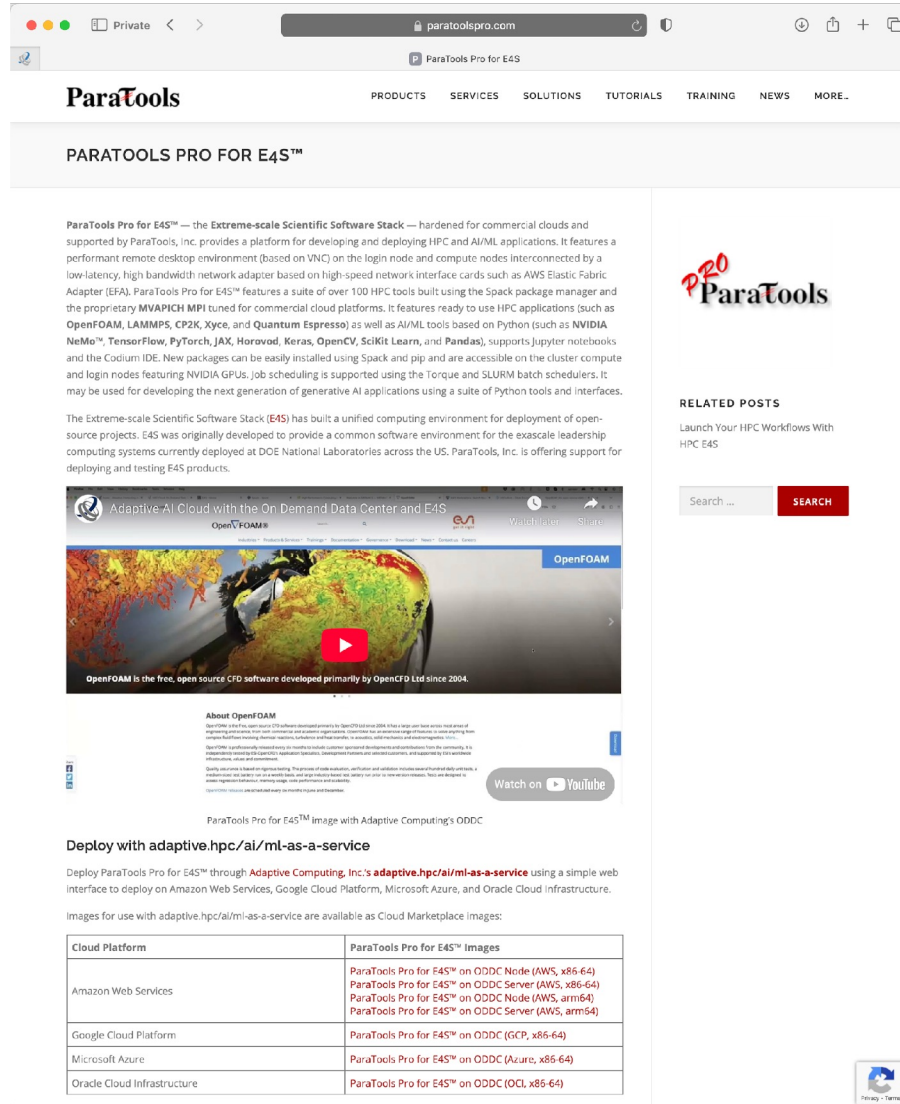
Key considerations for cloud-based deployment for E4S

- MPI - the core inter-node communication library has several implementations
 - Intel MPI, MVAPICH2-X, OpenMPI
 - Interfacing MPI with the job scheduling package (MOAB, Torque, SLURM)
- Cloud providers have different inter-node network adapters:
 - Elastic Fabric Adapter (EFA) on AWS
 - Infiniband on Azure
 - Mellanox Connect-X 5 Ethernet (ROCE) on Oracle Cloud Infrastructure (OCI)
 - IPU on Google Cloud (GCP)
 - VNIC on IBM Cloud
- Intra-node communication with XPMEM (driver and kernel module support is critical)
- GPU Direct Async (GDR) support for communication between GPUs in MVPICH-Plus release
- ParaTools, Inc. building E4S optimized with MVAPICH-Plus for AWS, OCI, GCP, and Azure
- Using Adaptive Computing, Inc.'s Heidi/ODDC to launch E4S jobs on multiple cloud providers!

Adaptive Computing's Heidi/ODDC with ParaTools Pro for E4S™



E4S on Commercial Cloud Platforms: ParaTools Pro for E4S™



- ParaTools Pro for E4S™* images in vendor marketplaces support:
 - AWS
 - Azure
 - Google Cloud (GCP)
 - Oracle Cloud Infrastructure (OCI)
- Supports SLURM and Torque for scheduling jobs on multi-node GPU accelerated nodes
- Shared GPU accelerated login node with a VNC based remote desktop
- Adaptive Computing's Heidi/ODDC
- AWS PCS and PC (x86, ARM64)
- Azure Cyclecloud
- Google GCluster

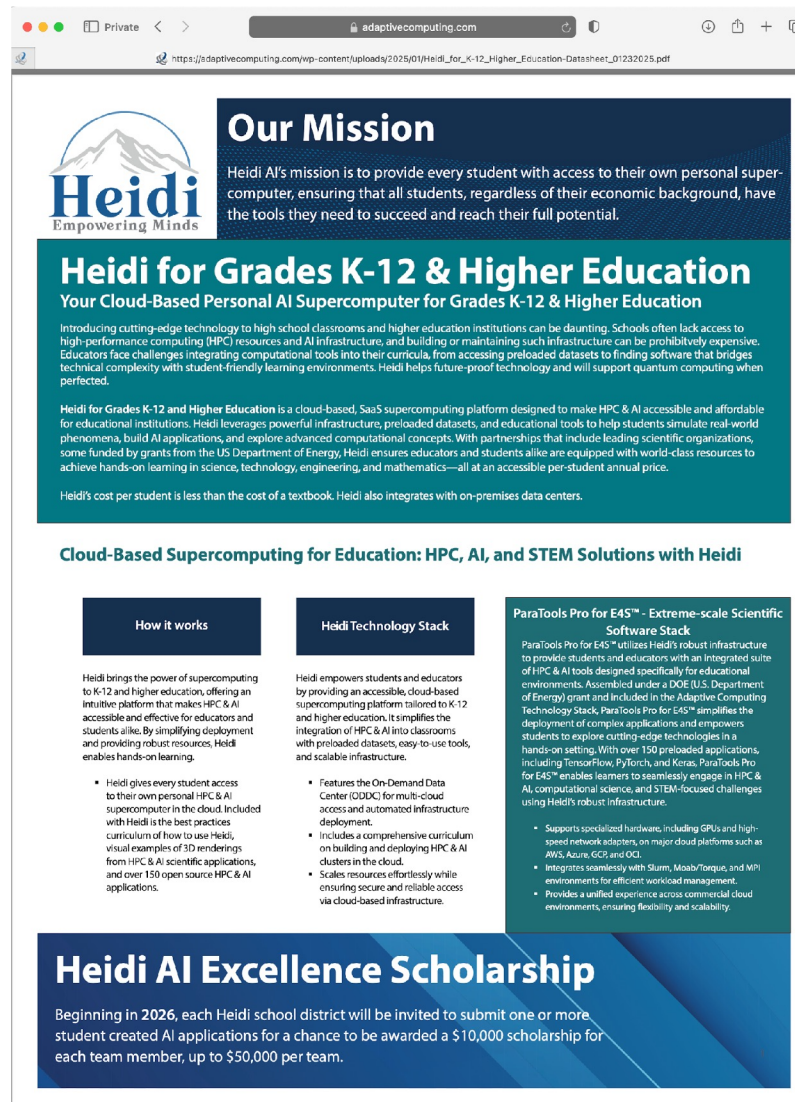
* Acknowledgment:
Supported by
DOE SBIR Phase I and II
DE-SC0022502



<https://paratoolspro.com> and <https://www.energy.gov/technologytransitions/sbirsttr>



E4S on Adaptive Computing's Heidi AI/On Demand Data Center (ODDC)



Heidi
Empowering Minds

Our Mission

Heidi AI's mission is to provide every student with access to their own personal super-computer, ensuring that all students, regardless of their economic background, have the tools they need to succeed and reach their full potential.

Heidi for Grades K-12 & Higher Education

Your Cloud-Based Personal AI Supercomputer for Grades K-12 & Higher Education

Introducing cutting-edge technology to high school classrooms and higher education institutions can be daunting. Schools often lack access to high-performance computing (HPC) resources and AI infrastructure, and building or maintaining such infrastructure can be prohibitively expensive. Educators face challenges integrating computational tools into their curricula, from accessing preloaded datasets to finding software that bridges technical complexity with student-friendly learning environments. Heidi helps future-proof technology and will support quantum computing when perfected.

Heidi for Grades K-12 and Higher Education is a cloud-based, SaaS supercomputing platform designed to make HPC & AI accessible and affordable for educational institutions. Heidi leverages powerful infrastructure, preloaded datasets, and educational tools to help students simulate real-world phenomena, build AI applications, and explore advanced computational concepts. With partnerships that include leading scientific organizations, some funded by grants from the US Department of Energy, Heidi ensures educators and students alike are equipped with world-class resources to achieve hands-on learning in science, technology, engineering, and mathematics—all at an accessible per-student annual price.

Heidi's cost per student is less than the cost of a textbook. Heidi also integrates with on-premises data centers.

Cloud-Based Supercomputing for Education: HPC, AI, and STEM Solutions with Heidi

How it works	Heidi Technology Stack	ParaTools Pro for E4S™ - Extreme-scale Scientific Software Stack
Heidi brings the power of supercomputing to K-12 and higher education, offering an intuitive platform that makes HPC & AI accessible and effective for educators and students alike. By simplifying deployment and providing robust resources, Heidi enables hands-on learning. Heidi gives every student access to their own personal HPC & AI supercomputer in the cloud. Included with Heidi is the best practices curriculum of how to use Heidi, visual examples of 3D renderings from HPC & AI scientific applications, and over 150 open source HPC & AI applications.	Heidi empowers students and educators by providing an accessible, cloud-based supercomputing platform tailored to K-12 and higher education. It simplifies the integration of HPC & AI into classrooms with preloaded datasets, easy-to-use tools, and scalable infrastructure. - Features the On-Demand Data Center (ODDC) for multi-cloud access and automated infrastructure deployment. - Includes a comprehensive curriculum on building and deploying HPC & AI clusters in the cloud. - Scales resources effortlessly while ensuring secure and reliable access via cloud-based infrastructure.	ParaTools Pro for E4S™ utilizes Heidi's robust infrastructure to provide students and educators with an integrated suite of HPC & AI tools designed specifically for educational environments. Assembled under a DOE (U.S. Department of Energy) grant and included in the Adaptive Computing Technology Stack, ParaTools Pro for E4S™ simplifies the deployment of complex applications and empowers students to explore cutting-edge technologies in a hands-on setting. With over 150 preloaded applications, including TensorFlow, PyTorch, and Keras, ParaTools Pro for E4S™ enables learners to seamlessly engage in HPC & AI, computational science, and STEM-focused challenges using Heidi's robust infrastructure. - Supports specialized hardware, including GPUs and high-speed network adapters, on major cloud platforms such as AWS, Azure, GCP, and OCI. - Integrates seamlessly with Slurm, Moab/Torque, and MPI environments for efficient workload management. - Provides a unified experience across commercial cloud environments, ensuring flexibility and scalability.

Heidi AI Excellence Scholarship

Beginning in 2026, each Heidi school district will be invited to submit one or more student created AI applications for a chance to be awarded a \$10,000 scholarship for each team member, up to \$50,000 per team.

- ParaTools Pro for E4S™ images in commercial cloud marketplaces launched using Heidi
- Supports Torque for scheduling jobs on multi-node GPU accelerated nodes
- Shared GPU accelerated login node with a VNC based remote desktop

ParaTools Pro for E4S™ on Commercial Clouds: AWS Marketplace

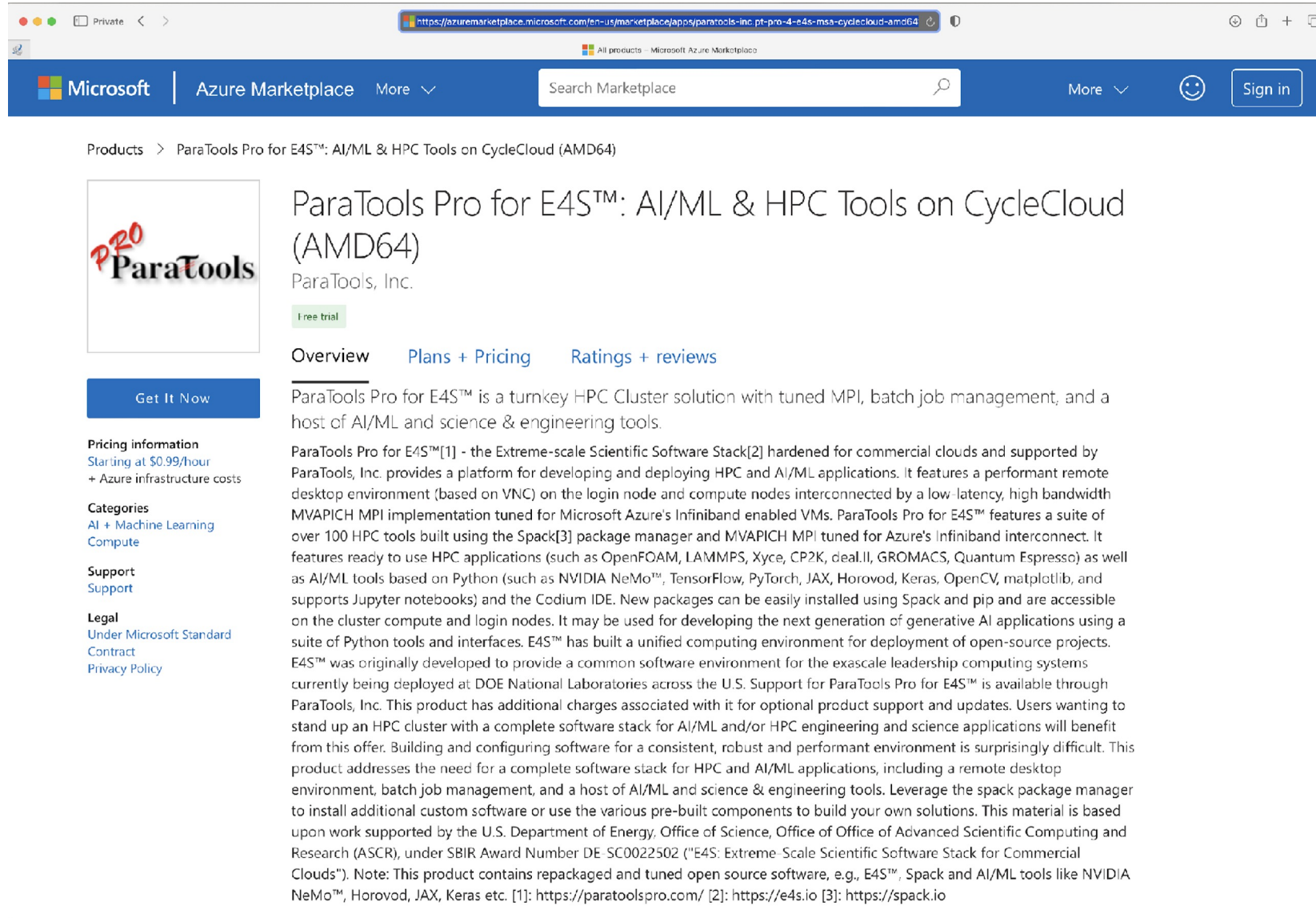
The screenshot shows the AWS Marketplace search results for 'ParaTools Pro for E4S'. The search bar at the top contains the text 'ParaTools Pro for E4S'. The results are displayed in a list format, showing four offerings. Each offering includes a 'Free Trial' badge and a description of the software stack and its capabilities. The offerings are:

- ParaTools Pro for E4S™: AI/ML & HPC Tools on ParallelCluster (arm64)**
By ParaTools Inc. | Ver v2025.03.05.1047-pcluster-5.12.0-e4s-24.11-arm64
Starting from \$0.99 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on DCV) on the login...
- ParaTools Pro for E4S™: AI/ML & HPC Tools on ParallelCluster (x86)**
By ParaTools Inc. | Ver v2025.03.24.1709-pcluster-3.12.0-e4s-24.11-amd64
Starting from \$0.99 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on DCV) on the login...
- ParaTools Pro for E4S™: AI/ML & HPC Tools on AWS PCS (x86)**
By ParaTools Inc. | Ver v2025.03.18.1119-pcs-1.1.1-e4s-24.11-amd64
Starting from \$0.10 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on DCV) on the login...
- ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC Node (x86)**
By ParaTools Inc. | Ver v17423266-odd-cv1.0-e4s-24.11-node-amd64
Starting from \$0.99 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login...

ParaTools Pro for E4S™ on AWS:

- AWS Trainium and Inferentia custom AI hardware with NeuronX SDK
- AWS PCS and PC on x86_64 and aarch64 nodes
- NVIDIA GPUs
- SLURM (PCS and PC) and Torque (ODDC node/server)
- Also on AWS Marketplace in GovCloud (US East & West)
- Elastic Fabric Adapter (EFA)
- MVAPICH MPI
- [X-ScaleSolutions, LLC and The Ohio State University]

ParaTools Pro for E4S™ on Commercial Clouds: Azure Marketplace



Products > ParaTools Pro for E4S™: AI/ML & HPC Tools on CycleCloud (AMD64)

ParaTools Pro for E4S™: AI/ML & HPC Tools on CycleCloud (AMD64)
ParaTools, Inc.
Free trial

[Get It Now](#)

Pricing information
Starting at \$0.99/hour
+ Azure infrastructure costs

Categories
AI + Machine Learning
Compute

Support
[Support](#)

Legal
[Under Microsoft Standard Contract](#)
[Privacy Policy](#)

Overview [Plans + Pricing](#) [Ratings + reviews](#)

ParaTools Pro for E4S™ is a turnkey HPC Cluster solution with tuned MPI, batch job management, and a host of AI/ML and science & engineering tools.

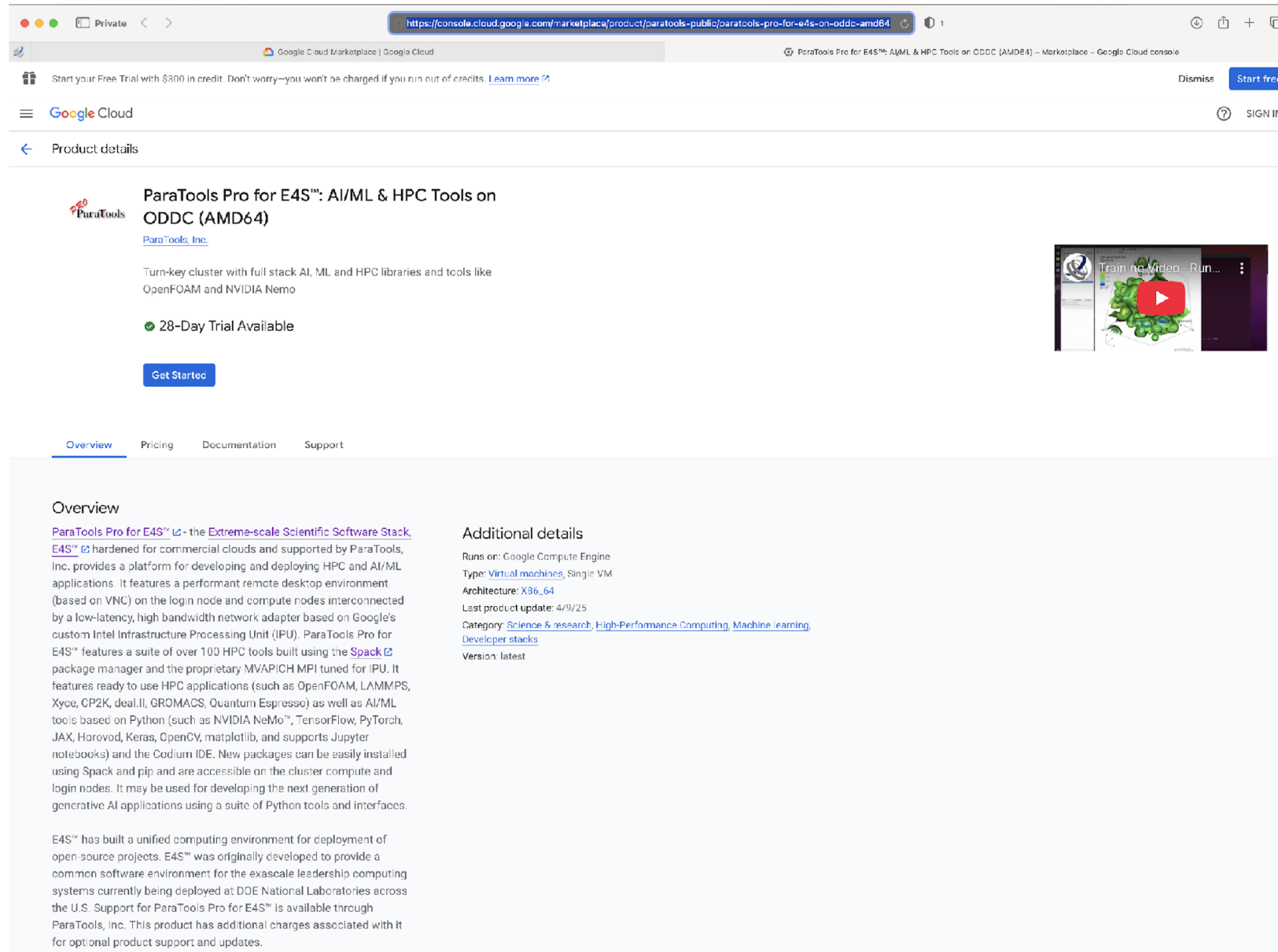
ParaTools Pro for E4S™[1] - the Extreme-scale Scientific Software Stack[2] hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login node and compute nodes interconnected by a low-latency, high bandwidth MVAPICH MPI implementation tuned for Microsoft Azure's Infiniband enabled VMs. ParaTools Pro for E4S™ features a suite of over 100 HPC tools built using the Spack[3] package manager and MVAPICH MPI tuned for Azure's Infiniband interconnect. It features ready to use HPC applications (such as OpenFOAM, LAMMPS, Xyce, CP2K, deal.II, GROMACS, Quantum Espresso) as well as AI/ML tools based on Python (such as NVIDIA NeMo™, TensorFlow, PyTorch, JAX, Horovod, Keras, OpenCV, matplotlib, and supports Jupyter notebooks) and the Codium IDE. New packages can be easily installed using Spack and pip and are accessible on the cluster compute and login nodes. It may be used for developing the next generation of generative AI applications using a suite of Python tools and interfaces. E4S™ has built a unified computing environment for deployment of open-source projects. E4S™ was originally developed to provide a common software environment for the exascale leadership computing systems currently being deployed at DOE National Laboratories across the U.S. Support for ParaTools Pro for E4S™ is available through ParaTools, Inc. This product has additional charges associated with it for optional product support and updates. Users wanting to stand up an HPC cluster with a complete software stack for AI/ML and/or HPC engineering and science applications will benefit from this offer. Building and configuring software for a consistent, robust and performant environment is surprisingly difficult. This product addresses the need for a complete software stack for HPC and AI/ML applications, including a remote desktop environment, batch job management, and a host of AI/ML and science & engineering tools. Leverage the spack package manager to install additional custom software or use the various pre-built components to build your own solutions. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Office of Advanced Scientific Computing and Research (ASCR), under SBIR Award Number DE-SC0022502 ("E4S: Extreme-Scale Scientific Software Stack for Commercial Clouds"). Note: This product contains repackaged and tuned open source software, e.g., E4S™, Spack and AI/ML tools like NVIDIA NeMo™, Horovod, JAX, Keras etc. [1]: <https://paratoolspro.com/> [2]: <https://e4s.io> [3]: <https://spack.io>

ParaTools Pro for E4S™

on Azure Marketplace supports

- SLURM (Azure CycleCloud) and Torque schedulers (Adaptive Computing's ODDC)
- Support for Infiniband Network adapter

ParaTools Pro for E4S™ on Google Cloud Marketplace



Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud

Product details

ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC (AMD64)

[ParaTools, Inc.](#)

Turn-key cluster with full stack AI, ML and HPC libraries and tools like OpenFOAM and NVIDIA Nemo

28-Day Trial Available

[Get Started](#)

[Overview](#) [Pricing](#) [Documentation](#) [Support](#)

Overview

[ParaTools Pro for E4S™](#) - the [Extreme-scale Scientific Software Stack](#), [E4S™](#) hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login node and compute nodes interconnected by a low-latency, high bandwidth network adapter based on Google's custom Intel Infrastructure Processing Unit (IPU). ParaTools Pro for E4S™ features a suite of over 100 HPC tools built using the [Spack](#) package manager and the proprietary MVAPICH MPI tuned for IPU. It features ready to use HPC applications (such as OpenFOAM, LAMMPS, Xyce, CP2K, deal.II, GROMACS, Quantum Espresso) as well as AI/ML tools based on Python (such as NVIDIA NeMo™, TensorFlow, PyTorch, JAX, Horovod, Keras, OpenCV, matplotlib, and supports Jupyter notebooks) and the Codium IDE. New packages can be easily installed using Spack and pip and are accessible on the cluster compute and login nodes. It may be used for developing the next generation of generative AI applications using a suite of Python tools and interfaces.

E4S™ has built a unified computing environment for deployment of open-source projects. E4S™ was originally developed to provide a common software environment for the exascale leadership computing systems currently being deployed at DOE National Laboratories across the U.S. Support for ParaTools Pro for E4S™ is available through ParaTools, Inc. This product has additional charges associated with it for optional product support and updates.

Additional details

Runs on: Google Compute Engine
Type: [Virtual machines](#), Single VM
Architecture: [X86_64](#)
Last product update: 4/9/25
Category: [Science & research](#), [High-Performance Computing](#), [Machine learning](#)
[Developer stacks](#)
Version: latest

ParaTools Pro for E4S™

Google Cloud Marketplace supports

- SLURM (GCluster) and Torque schedulers (Adaptive Computing's ODDC)
- Support for Google IPU network adapters

ParaTools Pro for E4S™ on Oracle Cloud Marketplace

The screenshot displays the Oracle Cloud Marketplace page for 'ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC (AMD64)'. The page includes a header with the Oracle logo and navigation links. The main content area features the app's logo, a brief description, a 'Get App' button, and a 'Self-Paced Training' link. Below this, there are tabs for 'Overview', 'Ratings (0)', and 'Provider'. The 'Overview' tab is selected, showing a detailed description of the app's features and capabilities. The description highlights that ParaTools Pro for E4S™ is a hardened Extreme-scale Scientific Software Stack using HPC Cloud ODDC, with over 100 HPC and AI/ML tools, and cluster configuration built using the Spack package manager and MVAPICH MPI tuned for OCI's network interconnect. It is a platform for developing AI/ML applications. The overview also mentions that ParaTools Pro for E4S™ provides a platform for developing and deploying HPC and AI/ML applications, featuring a performant, tuned MVAPICH2 MPI communication library to leverage OCI's low-latency, high bandwidth RDMA over Converged Ethernet (RoCE) network. ParaTools Pro for E4S™ features a suite of over 100 HPC tools built using the Spack package manager and the proprietary MVAPICH MPI tuned for RoCE. It features a performant remote desktop environment for GUI tools based on VNC, ready to use HPC applications (such as OpenFOAM, World Research and Forecasting Model-WRF, LAMMPS, Xyce, CP2K, deal.II, GROMACS, Quantum Espresso) as well as AI/ML tools based on Python (such as NVIDIA NeMo™, TensorFlow, PyTorch, JAX, Horovod, Keras, OpenCV, matplotlib and supports Jupyter notebooks) and the Codium IDE. It includes over 100 HPC libraries and tools such as PETSc, TAU, Imrinos, HDF5, HPCtoolkit, PA-M, NetCDF, ADIOS2, and Hypra. New packages can be easily installed using Spack and pip and are accessible on the cluster compute and login nodes. It may be used for developing the next generation of generative AI applications using a suite of Python tools and interfaces. The overview also states that E4S™ has built a unified computing environment for deployment of open-source projects. E4S™ was originally developed to provide a common software environment for the exascale leadership computing systems currently being deployed at DOE National Laboratories across the U.S. Support for ParaTools Pro for E4S™ is available through ParaTools, Inc. This product has additional charges associated with it for optional product support and updates. A note at the bottom states that this material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Office of Advanced Scientific Computing and Research (ASCR), under SBIR Award Number DE-SC0022502 ("E4S: Extreme-Scale Scientific Software Stack for Commercial Clouds"). A final note mentions that the product contains repackaged and tuned open source software (e.g., E4S™, Spack and AI/ML tools like Horovod, JAX, Keras etc.) which is configured and linked against a tuned MVAPICH MPI implementation specifically targeting OCI's RoCE network layer.

ParaTools Pro for E4S™
for Oracle Cloud Infrastructure
(OCI) Marketplace
supports Torque (ODDC) and
RDMA over Converged Ethernet
(RoCE) network adapters and
GPUs on login and compute nodes

ParaTools Pro for E4S™ on IBM Cloud Marketplace

IBM Cloud

ParaTools Pro for E4S (TM) - AI/ML HPC Tools on Heidi

ParaTools, Inc. • Third Party • Date of last update: 04/16/2026 • Docs • Get help • Readme file • View details • Change notices

Virtual private cloud (x86)

Select a delivery method

Server Image

Select product version

Product version

25.11.17715353-RL-unified

Version last updated: 04/16/2026

Summary

ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened

Summary

ParaTools Pro for E4S (TM) - AI/ML HPC Tools on Heidi

Deployment target: Virtual private cloud (x86)

Delivery method: Server Image

Already have an account? [Log in](#)

Sign up to create

ParaTools Pro for E4S™
for IBM Cloud Marketplace

Using ParaTools Pro for E4S™ image on AWS with Heidi AI/ODDC

Login to:

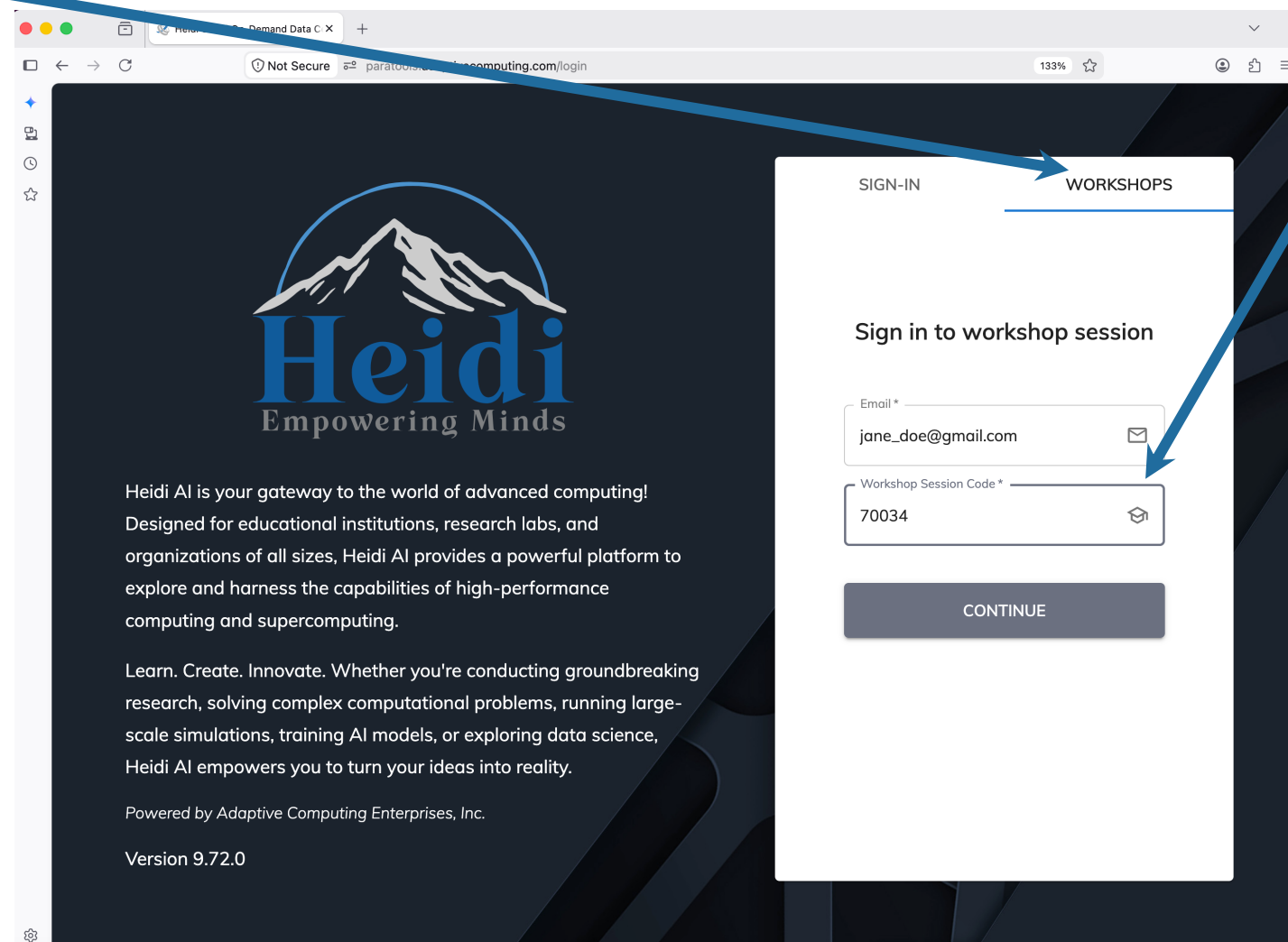
<https://paratools.adaptivecomputing.com>

with the credentials. Firefox private window recommended.

Click on Workshops tab and use code:70034

Connect to <https://paratools.adaptivecomputing.com>

- Use Workshops tab and enter email, session code **70034**



The screenshot shows a web browser window displaying the Heidi Empowering Minds login page. The page has a dark blue background with the Heidi logo (a mountain peak) and the text "Heidi Empowering Minds". Below the logo, there is a paragraph of text: "Heidi AI is your gateway to the world of advanced computing! Designed for educational institutions, research labs, and organizations of all sizes, Heidi AI provides a powerful platform to explore and harness the capabilities of high-performance computing and supercomputing." followed by another paragraph: "Learn. Create. Innovate. Whether you're conducting groundbreaking research, solving complex computational problems, running large-scale simulations, training AI models, or exploring data science, Heidi AI empowers you to turn your ideas into reality." and finally "Powered by Adaptive Computing Enterprises, Inc." and "Version 9.72.0". On the right side of the page, there is a white modal box with two tabs: "SIGN-IN" and "WORKSHOPS". The "WORKSHOPS" tab is selected. Inside the modal box, there is a heading "Sign in to workshop session" and two input fields: "Email *" with the value "jane_doe@gmail.com" and "Workshop Session Code *" with the value "70034". Below the input fields is a "CONTINUE" button. Two blue arrows point from the text in the list above to the "WORKSHOPS" tab and the "Workshop Session Code" input field.

Heidi Empowering Minds

Heidi AI is your gateway to the world of advanced computing!
Designed for educational institutions, research labs, and organizations of all sizes, Heidi AI provides a powerful platform to explore and harness the capabilities of high-performance computing and supercomputing.

Learn. Create. Innovate. Whether you're conducting groundbreaking research, solving complex computational problems, running large-scale simulations, training AI models, or exploring data science, Heidi AI empowers you to turn your ideas into reality.

Powered by Adaptive Computing Enterprises, Inc.
Version 9.72.0

SIGN-IN WORKSHOPS

Sign in to workshop session

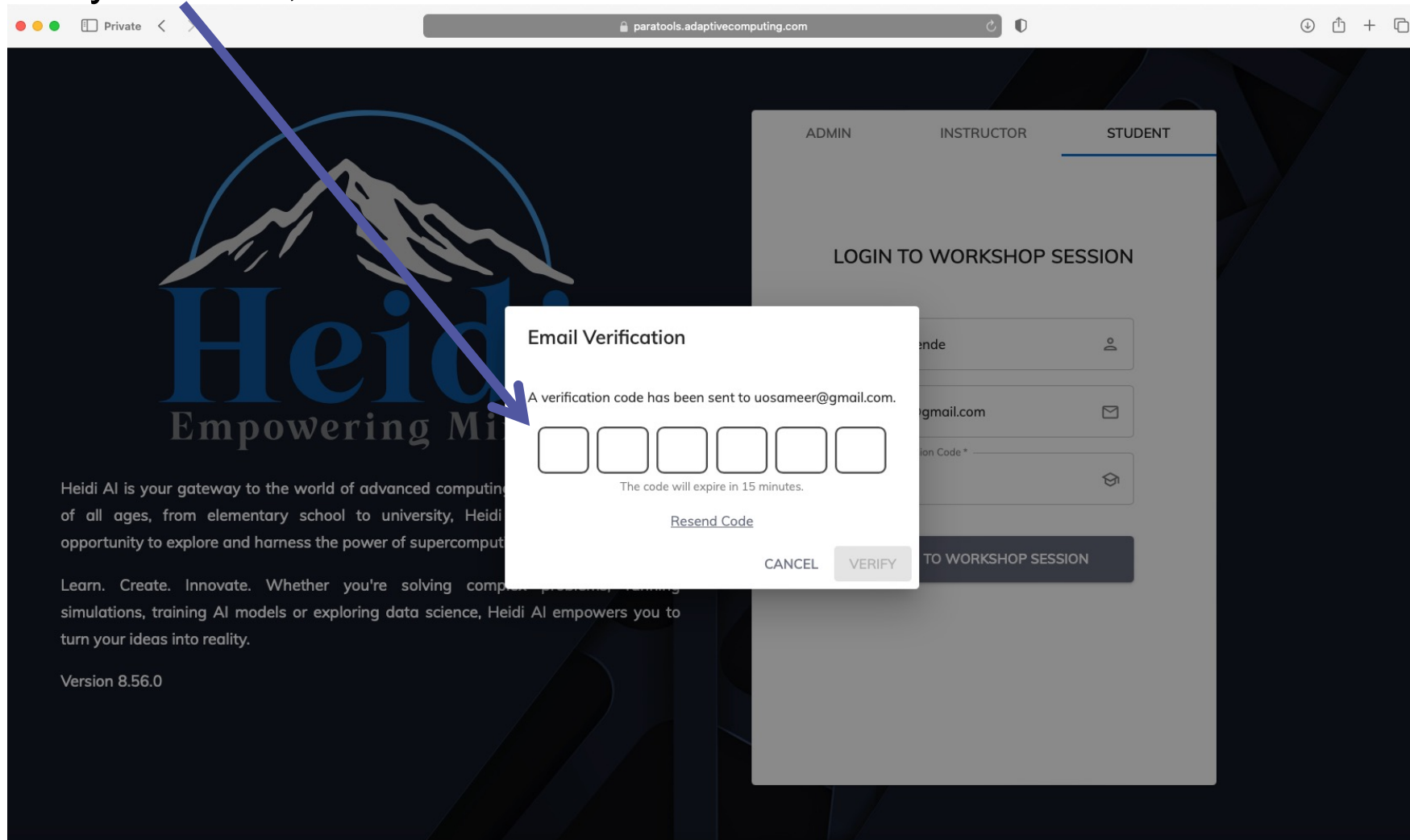
Email *
jane_doe@gmail.com

Workshop Session Code *
70034

CONTINUE

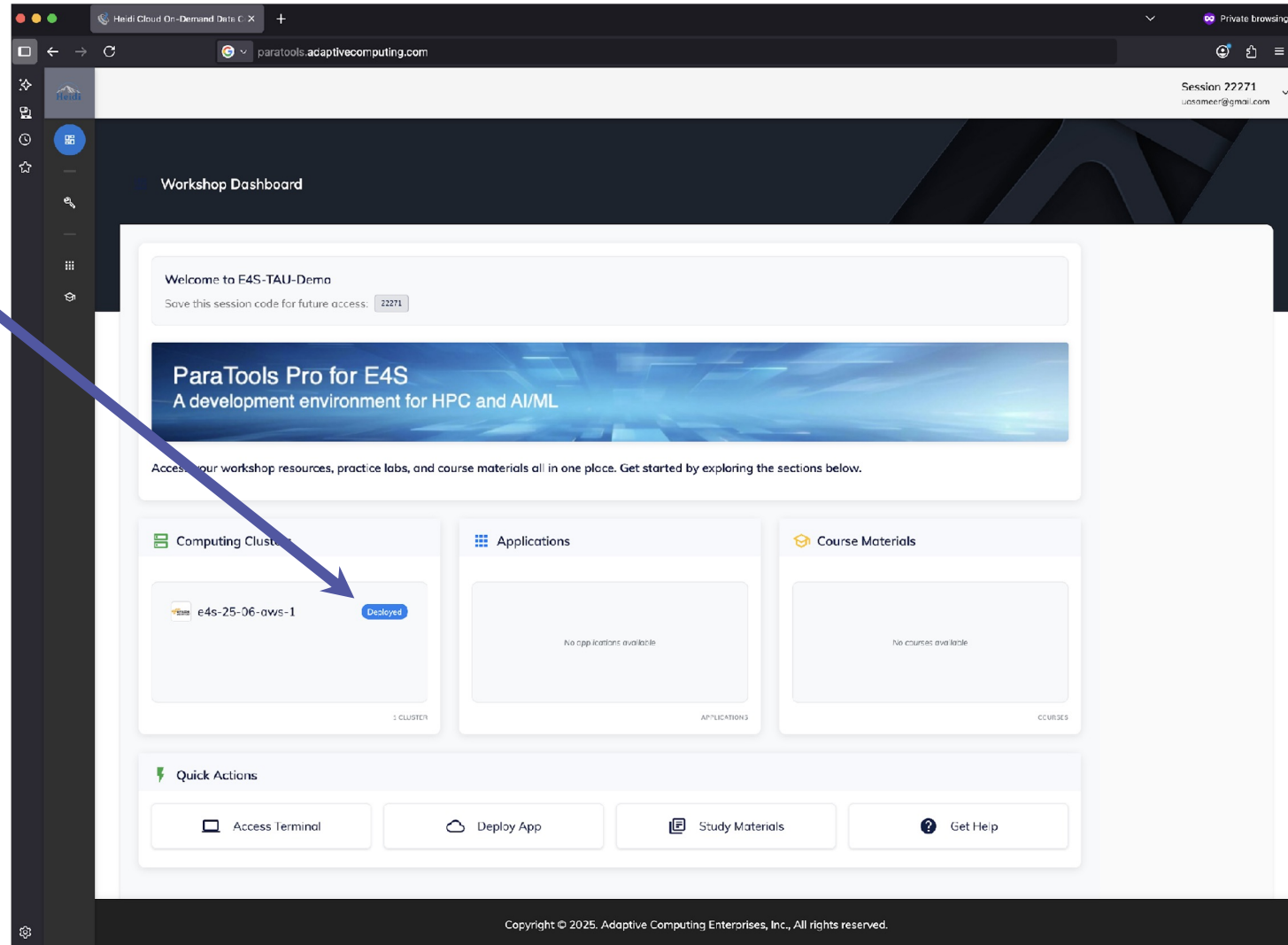
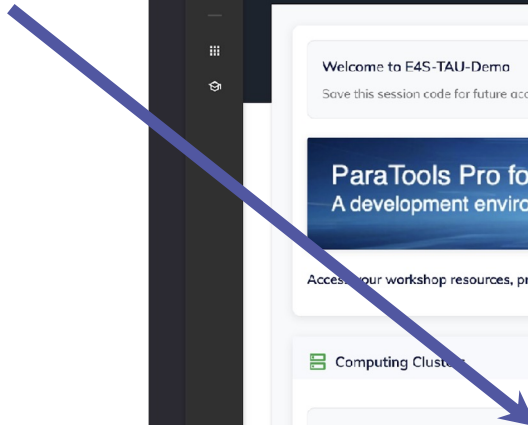
Connect to <https://paratools.adaptivecomputing.com>

- Check your email, enter verification code.



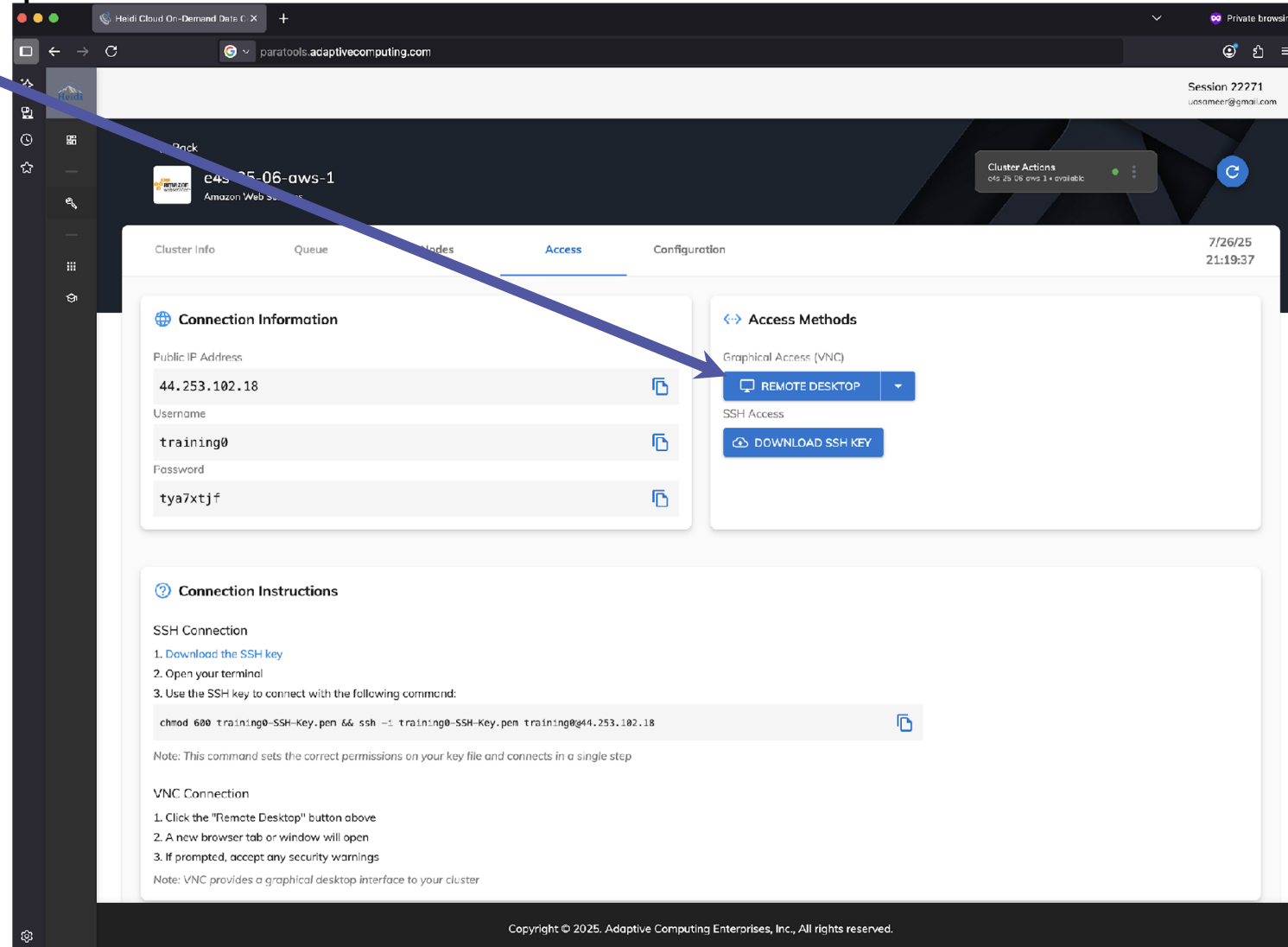
Connect to <https://paratools.adaptivecomputing.com>

- Click here



Connect to https://paratools.adaptivecomputing.com

- Click Remote Desktop



The screenshot shows the web interface of paratools.adaptivecomputing.com. A blue arrow points from the 'Click Remote Desktop' instruction to the 'REMOTE DESKTOP' button in the 'Access Methods' section. The interface includes a sidebar with navigation icons, a top navigation bar with tabs for Cluster Info, Queue, Nodes, Access (selected), and Configuration. The main content area displays connection information (Public IP Address: 44.253.102.18, Username: training0, Password: tya7xtjf) and access methods (Graphical Access (VNC) with a 'REMOTE DESKTOP' button, and SSH Access with a 'DOWNLOAD SSH KEY' button). Below this, there are 'Connection Instructions' for both SSH and VNC connections.

Connection Information

Public IP Address: 44.253.102.18

Username: training0

Password: tya7xtjf

Access Methods

Graphical Access (VNC)

REMOTE DESKTOP

SSH Access

DOWNLOAD SSH KEY

Connection Instructions

SSH Connection

- Download the SSH key
- Open your terminal
- Use the SSH key to connect with the following command:

```
chmod 600 training0-SSH-Key.pem && ssh -i training0-SSH-Key.pem training0@44.253.102.18
```

Note: This command sets the correct permissions on your key file and connects in a single step

VNC Connection

- Click the "Remote Desktop" button above
- A new browser tab or window will open
- If prompted, accept any security warnings

Note: VNC provides a graphical desktop interface to your cluster

Copyright © 2025. Adaptive Computing Enterprises, Inc., All rights reserved.

Connect to <https://paratools.adaptivecomputing.com>

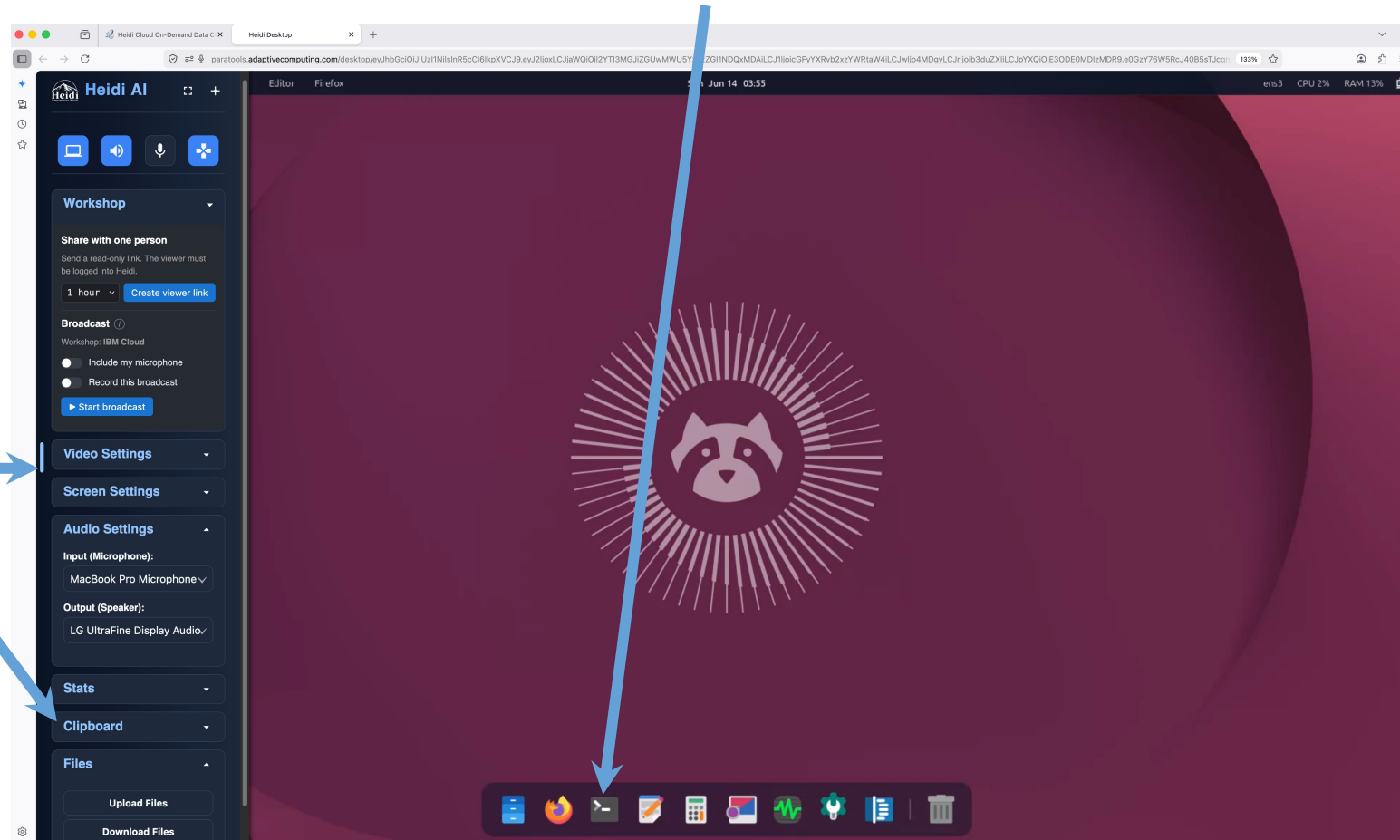
- You may have to enable pop-up windows and accept

The screenshot displays the web interface of paratools.adaptivecomputing.com. A blue arrow points to the address bar, which contains the URL paratools.adaptivecomputing.com. The interface features a dark sidebar on the left with navigation options: Dashboard, COMPUTE / IDE, Heidi Lab, Cluster Manager, CONTENT LIBRARY, Applications, and Courses. The main content area shows the 'Access' tab for a cluster named 'e4s-25-06-aws' on Amazon Web Services. The 'Access' tab includes sections for 'Connection Information' (Public IP Address: 54.70.253.146, Username: training0, Password: 7y19fqvw), 'Access Methods' (Graphical Access (VNC) with 'OPEN VNC VIEWER' button, SSH Key Authentication with 'DOWNLOAD SSH KEY' button, and VPN Access), and 'Connection Instructions' (SSH Connection steps: 1. Download the SSH key, 2. Open your terminal). The top right corner shows 'Session 77946' and 'uosameer@gmail.com'. The bottom of the page has a copyright notice: 'Copyright © 2025. Adaptive Computing Enterprises, Inc., All rights reserved.'

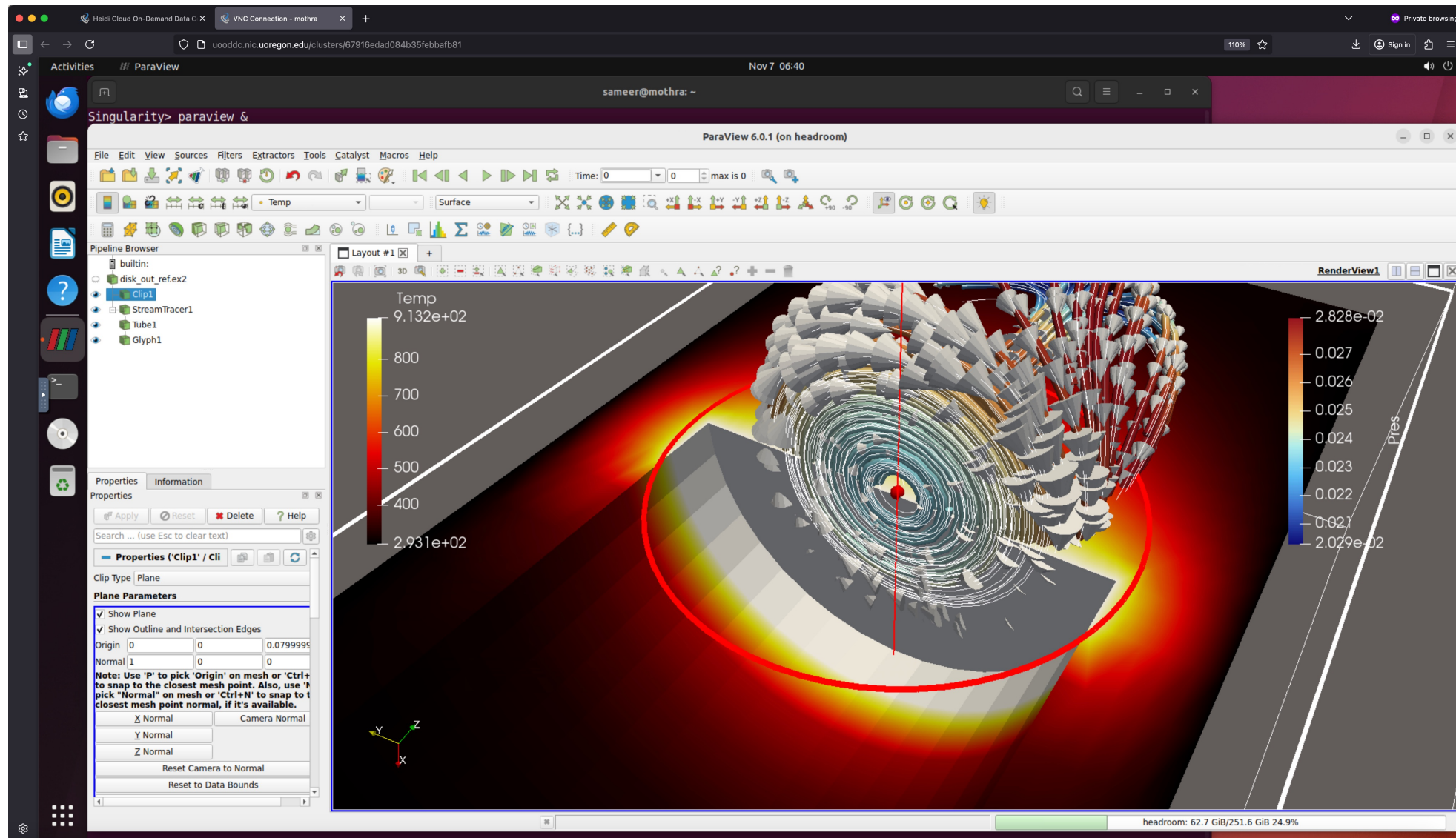
Connect to Students tab with code 70034 at <https://paratools.adaptivecomputing.com>

- You should see this racoon. Click on Terminal.

To copy text from other windows click on the this button to access the clipboard



E4S: Visualization Tools: ParaView



E4S: Marimo Reactive Notebooks

The screenshot shows a VNC connection to a Singularity environment. The terminal window displays the following commands and output:

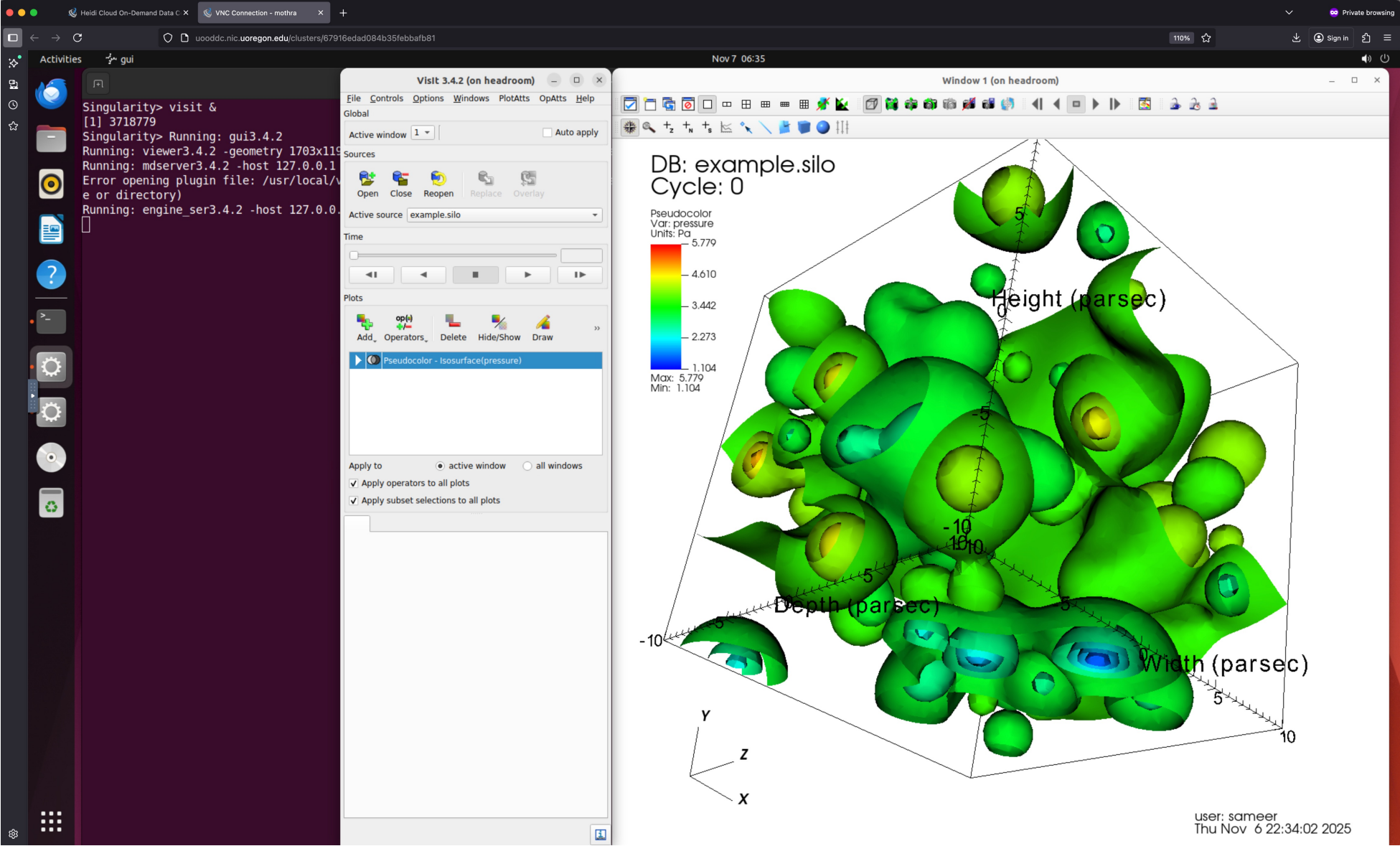
```
Singularity> sycl-ls
[level_zero:gpu][level_zero:0] Intel(R) oneAPI Unified Runtime over Level-Zero, Intel(R) Data Center GPU Max 1100 12.60.7 [1.6.35096+9]
[opencl:cpu][opencl:0] Intel(R) OpenCL, Intel(R) Xeon(R) Silver 4410T OpenCL 3.0 (Build 0) [2025.20.8.0.06 160000]
[opencl:gpu][opencl:1] Intel(R) OpenCL Graphics, Intel(R) Data Center GPU Max 1100 OpenCL 3.0 NEO [25.35.35096.9]
Singularity> which adk
/opt/python/pkgs/python-3.12.11/bin/adk
Singularity> which marimo
/opt/python/pkgs/python-3.12.11/bin/marimo
Singularity> marimo new
```

Below the terminal, a web browser window titled "Untitled Notebook — Mozilla Firefox (on headroom)" displays a Marimo notebook. The notebook code is as follows:

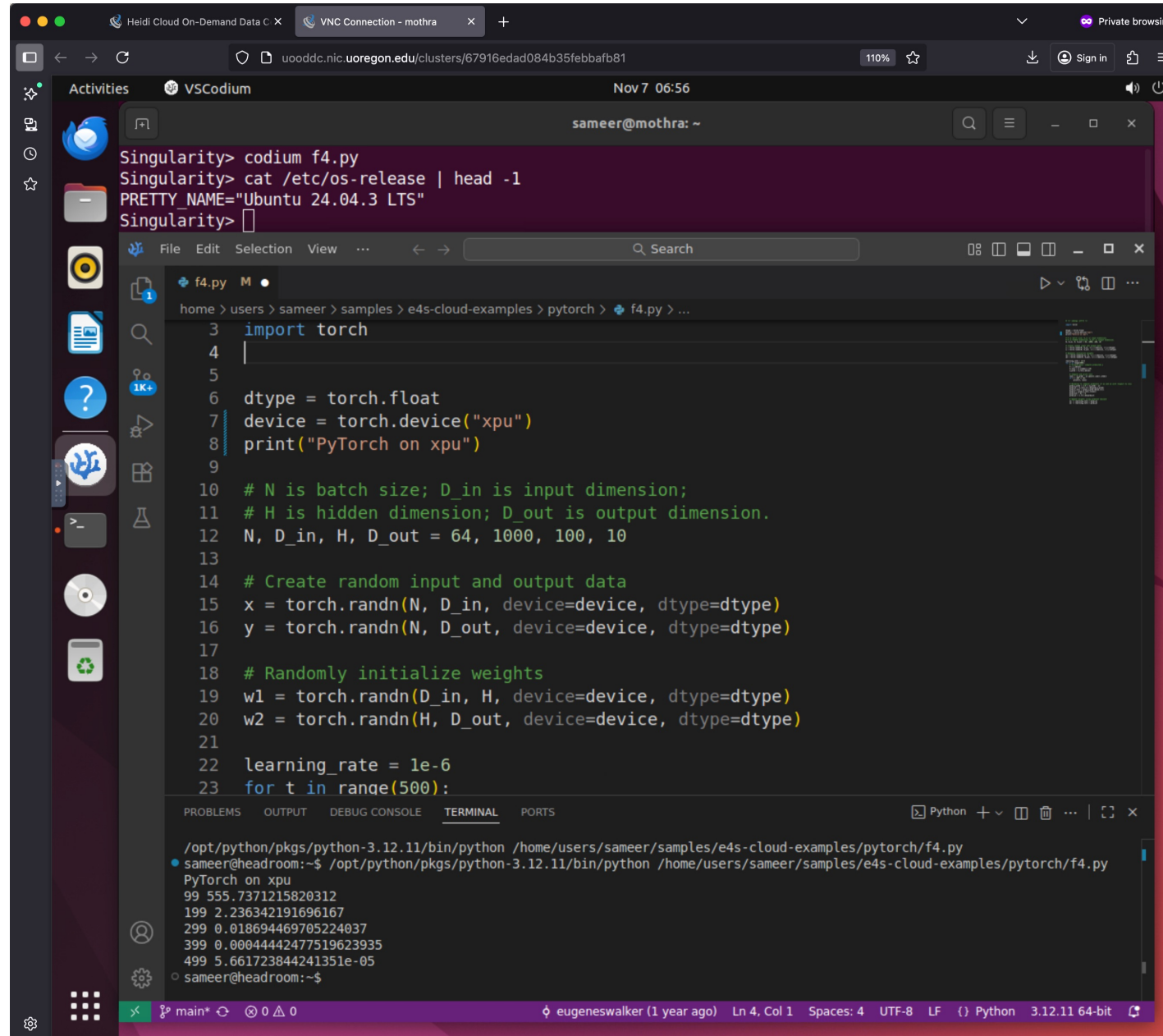
```
1 import marimo as mo
2 import plotly.graph_objects as go
3 import numpy as np
4
5 # Generate 3D surface data
6 x = np.linspace(-5, 5, 100)
7 y = np.linspace(-5, 5, 100)
8 x, y = np.meshgrid(x, y)
9 z = np.sin(np.sqrt(x**2 + y**2))
10
11 # Create a Plotly 3D surface plot
12 fig = go.Figure(
13     data=[go.Surface(z=z, x=x, y=y, colorscale='Viridis')]
14 )
15 fig.update_layout(title="3D Surface Example")
16
17 # Display the 3D plot in Marimo
18 mo.ui.plotly(fig)
```

The notebook output shows a 3D surface plot titled "3D Surface Example". The plot is a surface plot with a color scale ranging from 0 to 1, using the Viridis color map. The plot is displayed in a 3D view with axes labeled x, y, and z.

E4S: VisIt



E4S: VS Codium Integrated Development Environment



The screenshot displays the VS Codium IDE interface. The top terminal shows the following commands and output:

```
Singularity> codium f4.py
Singularity> cat /etc/os-release | head -1
PRETTY_NAME="Ubuntu 24.04.3 LTS"
Singularity>
```

The code editor shows the file `f4.py` with the following Python code:

```
3 import torch
4
5
6 dtype = torch.float
7 device = torch.device("xpu")
8 print("PyTorch on xpu")
9
10 # N is batch size; D_in is input dimension;
11 # H is hidden dimension; D_out is output dimension.
12 N, D_in, H, D_out = 64, 1000, 100, 10
13
14 # Create random input and output data
15 x = torch.randn(N, D_in, device=device, dtype=dtype)
16 y = torch.randn(N, D_out, device=device, dtype=dtype)
17
18 # Randomly initialize weights
19 w1 = torch.randn(D_in, H, device=device, dtype=dtype)
20 w2 = torch.randn(H, D_out, device=device, dtype=dtype)
21
22 learning_rate = 1e-6
23 for t in range(500):
```

The bottom terminal shows the execution output:

```
/opt/python/pkgs/python-3.12.11/bin/python /home/users/sameer/samples/e4s-cloud-examples/pytorch/f4.py
sameer@headroom:~$ /opt/python/pkgs/python-3.12.11/bin/python /home/users/sameer/samples/e4s-cloud-examples/pytorch/f4.py
PyTorch on xpu
99 555.7371215820312
199 2.236342191696167
299 0.018694469705224037
399 0.00044442477519623935
499 5.661723844241351e-05
sameer@headroom:~$
```

ParaTools Pro for E4S™: NVIDIA BioNeMo™ on IBM Cloud

paratoolsadmin@e4s-ibm: ~/examples/bionemo\$ codium

Zero-shot prediction of BRCA1 variant effects with Evo 2

Deploy this tutorial on brev.dev: [Deploy Now](#)

Note - this notebook is a reproduction of The Arc Institute's same-titled notebook [here](#), using the BioNeMo 2 implementation of Evo2.

Evo2 is a foundation AI model trained on 9.3 trillion DNA base pairs, predicting variant effects without prior task-specific training.

Without being explicitly trained on BRCA1 variants, we show Evo 2's ability to generalize across all life forms.

The human *BRCA1* gene encodes for a protein that repairs damaged DNA ([Moynahan et al., 1999](#)). Certain variants of this gene have been associated with an increased risk of breast and ovarian cancers ([Miki et al., 1994](#)). Using Evo 2, we can predict whether a particular single nucleotide variant (SNV) of the *BRCA1* gene is likely to be harmful to the protein's function, and thus potentially increase the risk of cancer for the patient with the genetic variant.

```
%capture
import os

# Runs a subset of the model layers to test that the notebook runs in CI, but the output will be incorrect.
FAST_CI_MODE: bool = os.environ.get("FAST_CI_MODE", False)
```

```
import glob
import gzip
import json
import math
import os
from pathlib import Path

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import torch
from Bio import SeqIO
from sklearn.metrics import auc, roc_auc_score, roc_curve
```

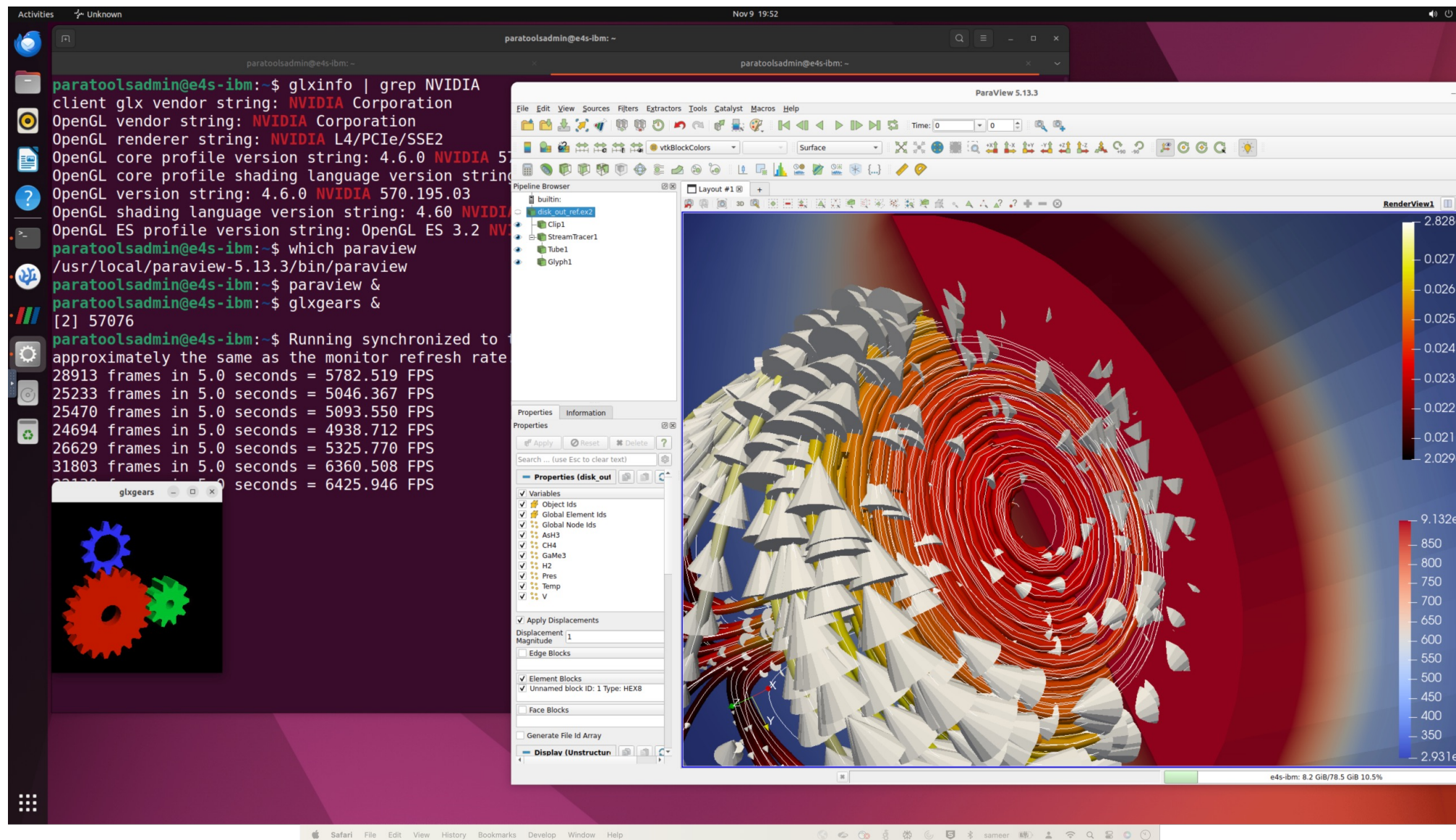
We start by loading a dataset from [Findlay et al. \(2018\)](#), which contains experimentally measured function scores of 3,893 *BRCA1* SNVs. These function scores reflect the extent by which the genetic variant has disrupted the protein's function, with lower scores indicating greater disruption. In this dataset, the SNVs are classified into three categories based on their function scores: **LOF** (loss-of-function), **INT** (intermediate), and **FUNC** (functional). We start by reading in this dataset.

To keep the notebook streamlined, we've abstracted much of the preprocessing logic into accompanying scripts located in [brca1_utils](#). The full notebook can be viewed [here](#).

```
def download_data(data_dir="brca1", commit_hash="3819474bee6c24938016614411f1fa025e542bbe"):
    """Download required data files if they don't exist locally.

    Parameters:
    -----
    data_dir : str
        Directory to store downloaded files
    commit_hash : str
        GitHub commit hash for data version
    """
```

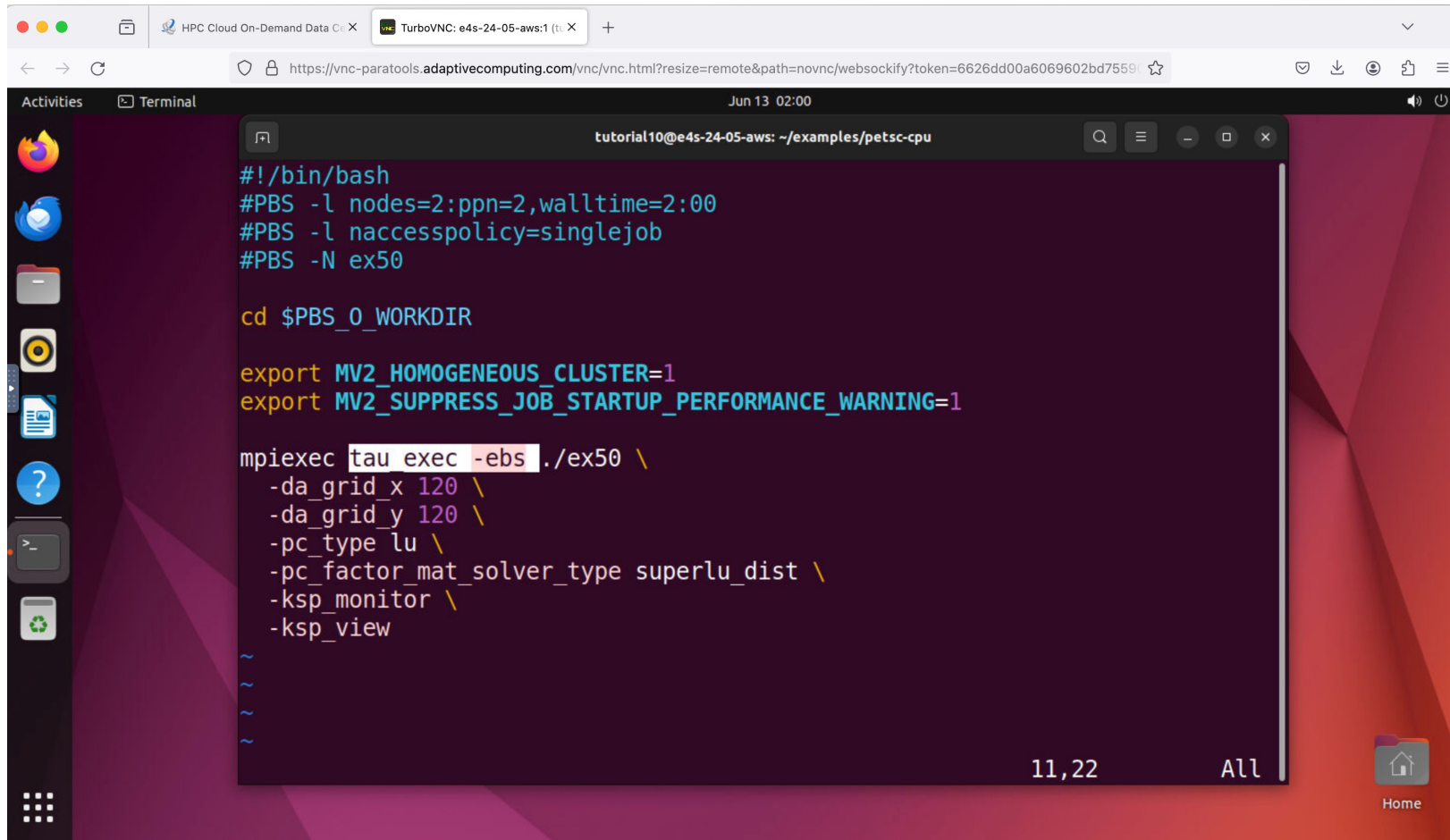

ParaTools Pro for E4S™: HPC-AI Software Ecosystem on Clouds



TAU Exercise #1:

Instrumenting PETSc application using TAU's Perfstubs interface

Launching the binary using tau_exec -ebs



The screenshot shows a terminal window titled 'tutorial10@e4s-24-05-aws: ~/examples/petsc-cpu'. The terminal output is as follows:

```
#!/bin/bash
#PBS -l nodes=2:ppn=2,walltime=2:00
#PBS -l naccesspolicy=singlejob
#PBS -N ex50

cd $PBS_O_WORKDIR

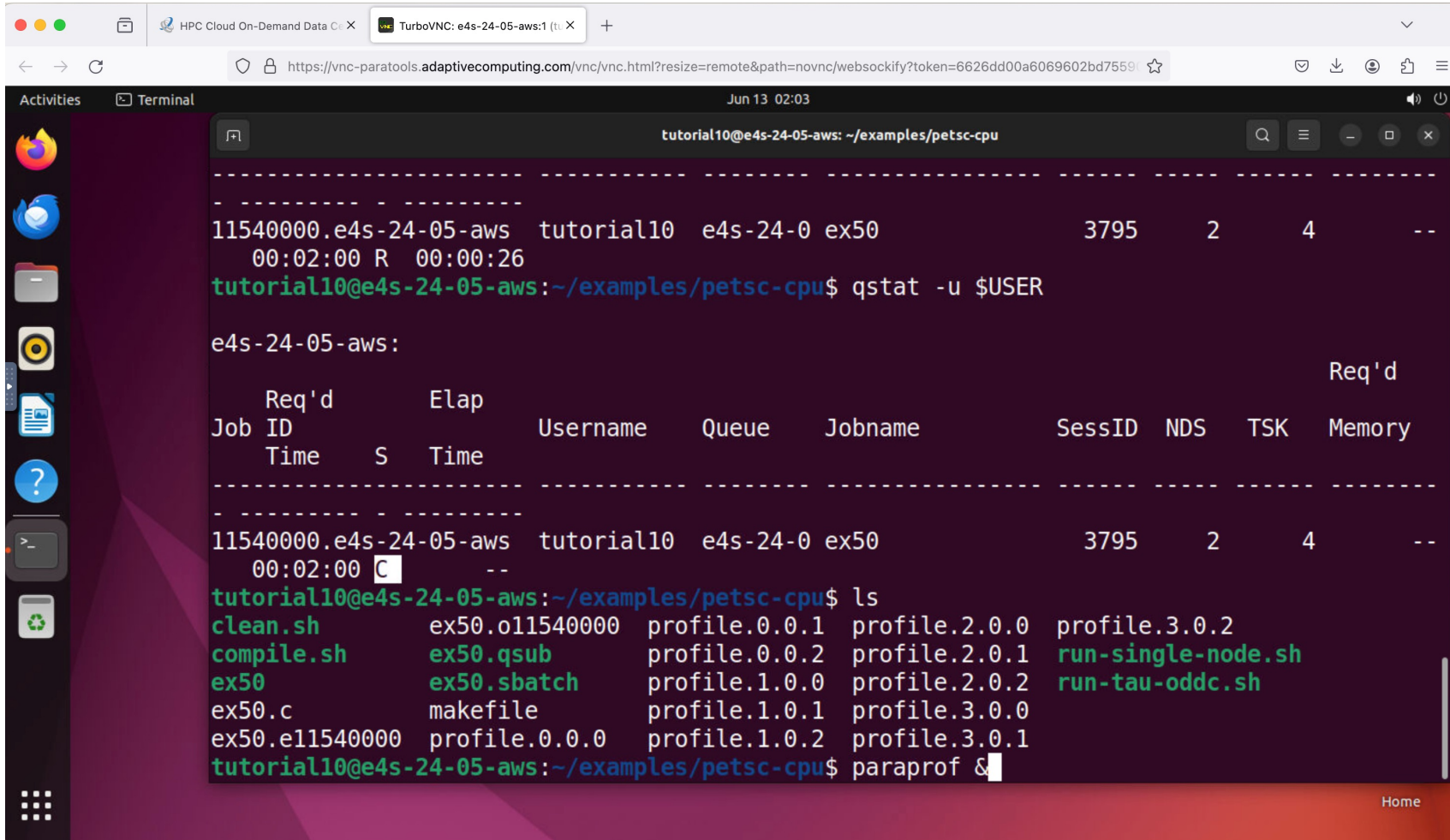
export MV2_HOMOGENEOUS_CLUSTER=1
export MV2_SUPPRESS_JOB_STARTUP_PERFORMANCE_WARNING=1

mpiexec tau exec -ebs ./ex50 \
  -da_grid_x 120 \
  -da_grid_y 120 \
  -pc_type lu \
  -pc_factor_mat_solver_type superlu_dist \
  -ksp_monitor \
  -ksp_view
```

The terminal window also shows a status bar at the bottom with '11,22' and 'All'.

```
cd ~/examples/petsc-cpu
cat tau.sbatch
./compile.sh
sbatch tau.sbatch
pprof -a
paraprof &
```

TAU's ParaProf Profile Browser: Source Code Browser

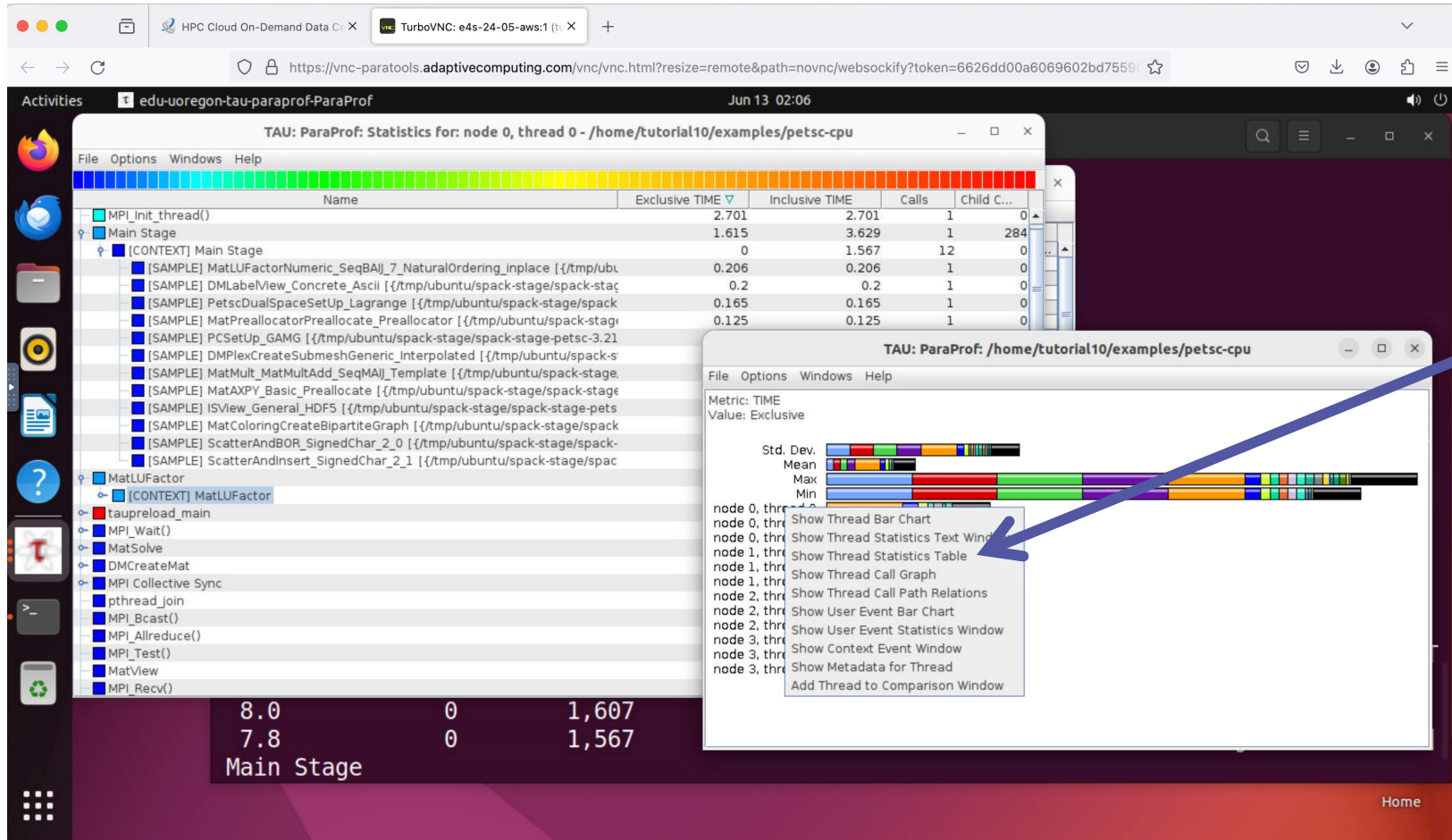


```
tutorial10@e4s-24-05-aws: ~/examples/petsc-cpu
-----
11540000.e4s-24-05-aws tutorial10 e4s-24-0 ex50 3795 2 4 --
00:02:00 R 00:00:26
tutorial10@e4s-24-05-aws:~/examples/petsc-cpu$ qstat -u $USER
e4s-24-05-aws:
Req'd Elap
Job ID Time S Time Username Queue Jobname SessID NDS TSK Req'd Memory
-----
11540000.e4s-24-05-aws tutorial10 e4s-24-0 ex50 3795 2 4 --
00:02:00 C --
tutorial10@e4s-24-05-aws:~/examples/petsc-cpu$ ls
clean.sh ex50.o11540000 profile.0.0.1 profile.2.0.0 profile.3.0.2
compile.sh ex50.qsub profile.0.0.2 profile.2.0.1 run-single-node.sh
ex50 ex50.sbatch profile.1.0.0 profile.2.0.2 run-tau-oddc.sh
ex50.c makefile profile.1.0.1 profile.3.0.0
ex50.e11540000 profile.0.0.0 profile.1.0.2 profile.3.0.1
tutorial10@e4s-24-05-aws:~/examples/petsc-cpu$ paraprof &
```

sbatch tau.sbatch
queue -u \$USER

After it completes
ls
paraprof &

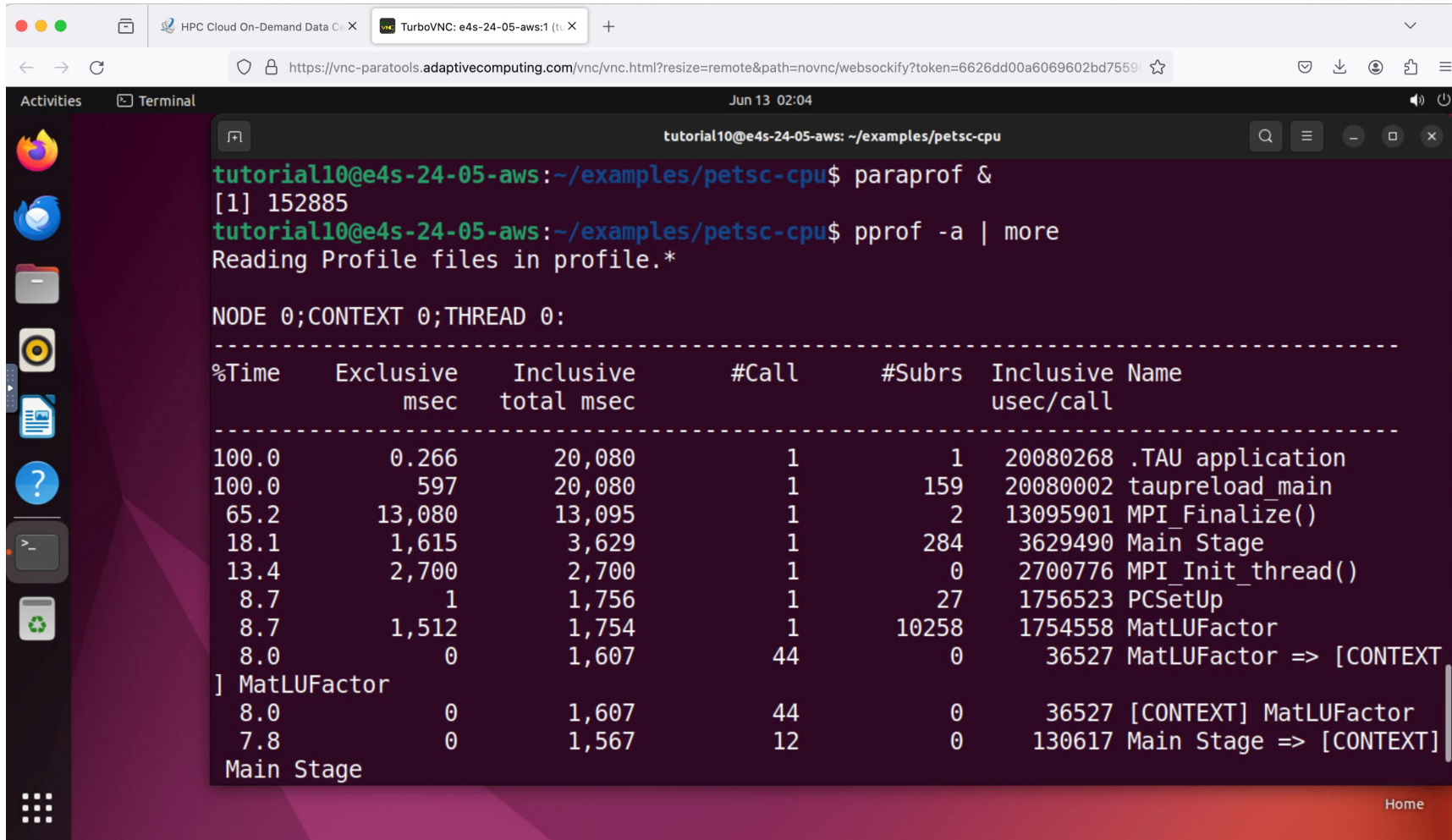
TAU's paraprof browser with PETSc performance profile



paraprof

Choose
Show thread statistics table by
right clicking on
node 0, thread 0.

Using pprof: TAU's text based profile browser



```
tutorial10@e4s-24-05-aws: ~/examples/petsc-cpu
tutorial10@e4s-24-05-aws:~/examples/petsc-cpu$ paraprof &
[1] 152885
tutorial10@e4s-24-05-aws:~/examples/petsc-cpu$ pprof -a | more
Reading Profile files in profile.*

NODE 0;CONTEXT 0;THREAD 0:
-----
%Time   Exclusive   Inclusive   #Call   #Subrs   Inclusive Name
      msec    total msec                usec/call
-----
100.0    0.266      20,080      1        1   20080268 .TAU application
100.0      597      20,080      1       159   20080002 taupreload_main
65.2   13,080    13,095      1        2   13095901 MPI_Finalize()
18.1    1,615     3,629      1       284   3629490 Main Stage
13.4    2,700     2,700      1        0   2700776 MPI_Init_thread()
8.7        1     1,756      1        27   1756523 PCSetUp
8.7    1,512     1,754      1     10258   1754558 MatLUFactor
8.0        0     1,607     44        0    36527 MatLUFactor => [CONTEXT
] MatLUFactor
8.0        0     1,607     44        0    36527 [CONTEXT] MatLUFactor
7.8        0     1,567     12        0   130617 Main Stage => [CONTEXT]
Main Stage
```

pprof -a | more

Here we see PETSc timers translated into TAU timers using the Perfstubs library.

No modification to the source, build system, or the binary!

TAU Exercise #2:

**Instrumenting PETSc application using TAU's
Perfstubs interface and generating callpath profiles**

Generating callpath profiles

```
Edit tau.sbatch
# add
export TAU_CALLPATH=1
export TAU_CALLPATH_DEPTH=100

export TAU_PROFILE_FORMAT=merged
mpirun ...

% sbatch tau.sbatch
% paraprof tauprofile.xml
```

TAU Exercise #3:

CUDA instrumentation using CUPTI (CUDA Profiling Tools Interface)

PETSc CUDA: What it was doing

```
% cd ~/examples/petsc-cuda
% ./compile.sh
% cat tau.sbatch
# spack load tau+mpi+cuda

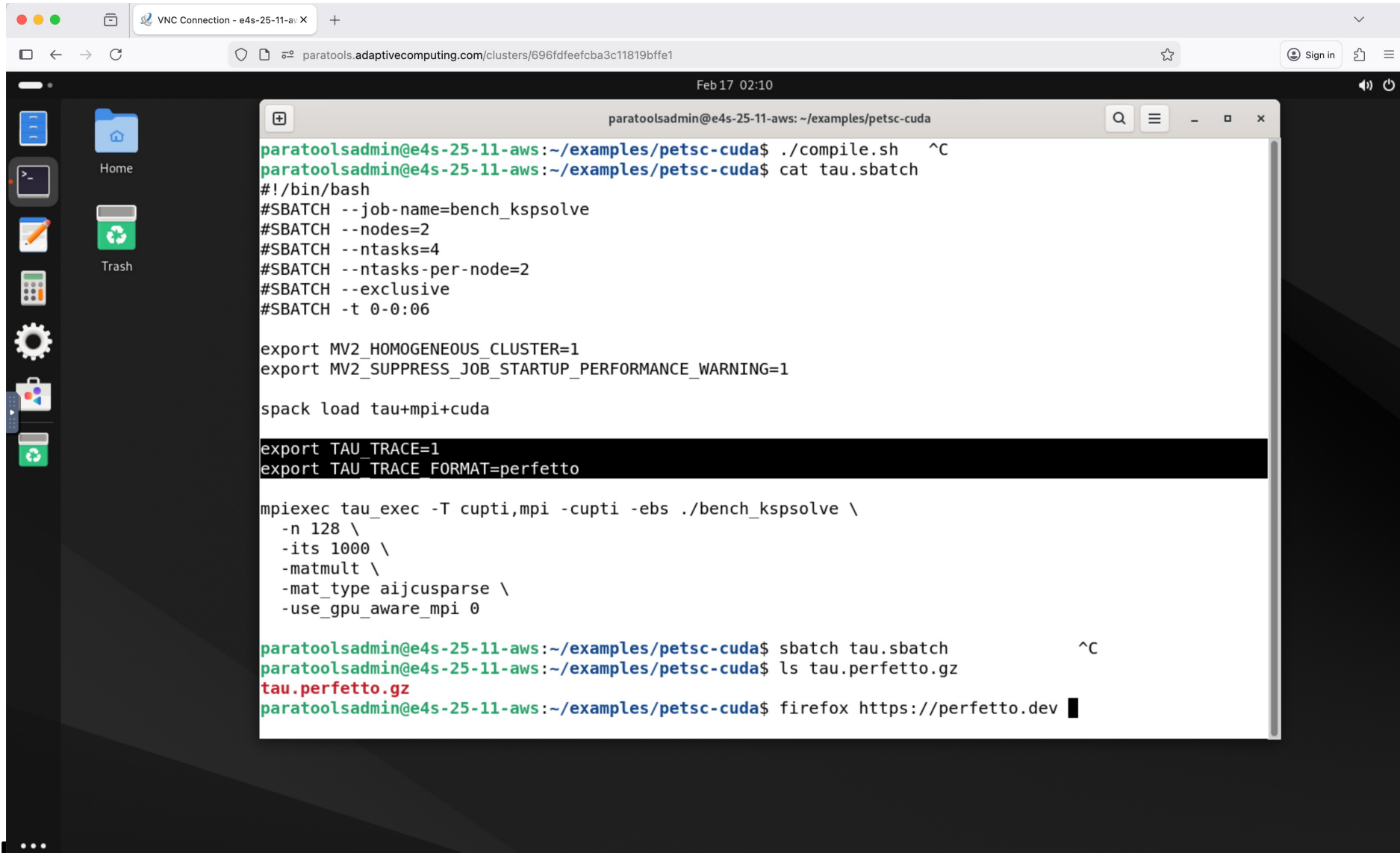
mpiexec tau_exec -T cupti,mpi -cupti -ebs ./ksp...
% sbatch tau.sbatch

% paraprof &
```


TAU Exercise #4:

Generating CUDA traces for Perfetto.dev using CUPTI (CUDA Profiling Tools Interface)

PETSc CUDA with Perfetto Traces generated with TAU



The screenshot shows a VNC connection to a terminal window. The terminal prompt is `paratoolsadmin@e4s-25-11-aws: ~/examples/petsc-cuda`. The user runs `./compile.sh` and `cat tau.sbatch`. The `tau.sbatch` file contains the following content:

```
#!/bin/bash
#SBATCH --job-name=bench_kpsolve
#SBATCH --nodes=2
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=2
#SBATCH --exclusive
#SBATCH -t 0-0:06

export MV2_HOMOGENEOUS_CLUSTER=1
export MV2_SUPPRESS_JOB_STARTUP_PERFORMANCE_WARNING=1

spack load tau+mpi+cuda

export TAU_TRACE=1
export TAU_TRACE_FORMAT=perfetto

mpiexec tau_exec -T cupti,mpi -cupti -ebs ./bench_kpsolve \
  -n 128 \
  -its 1000 \
  -matmult \
  -mat_type aijcusparse \
  -use_gpu_aware_mpi 0
```

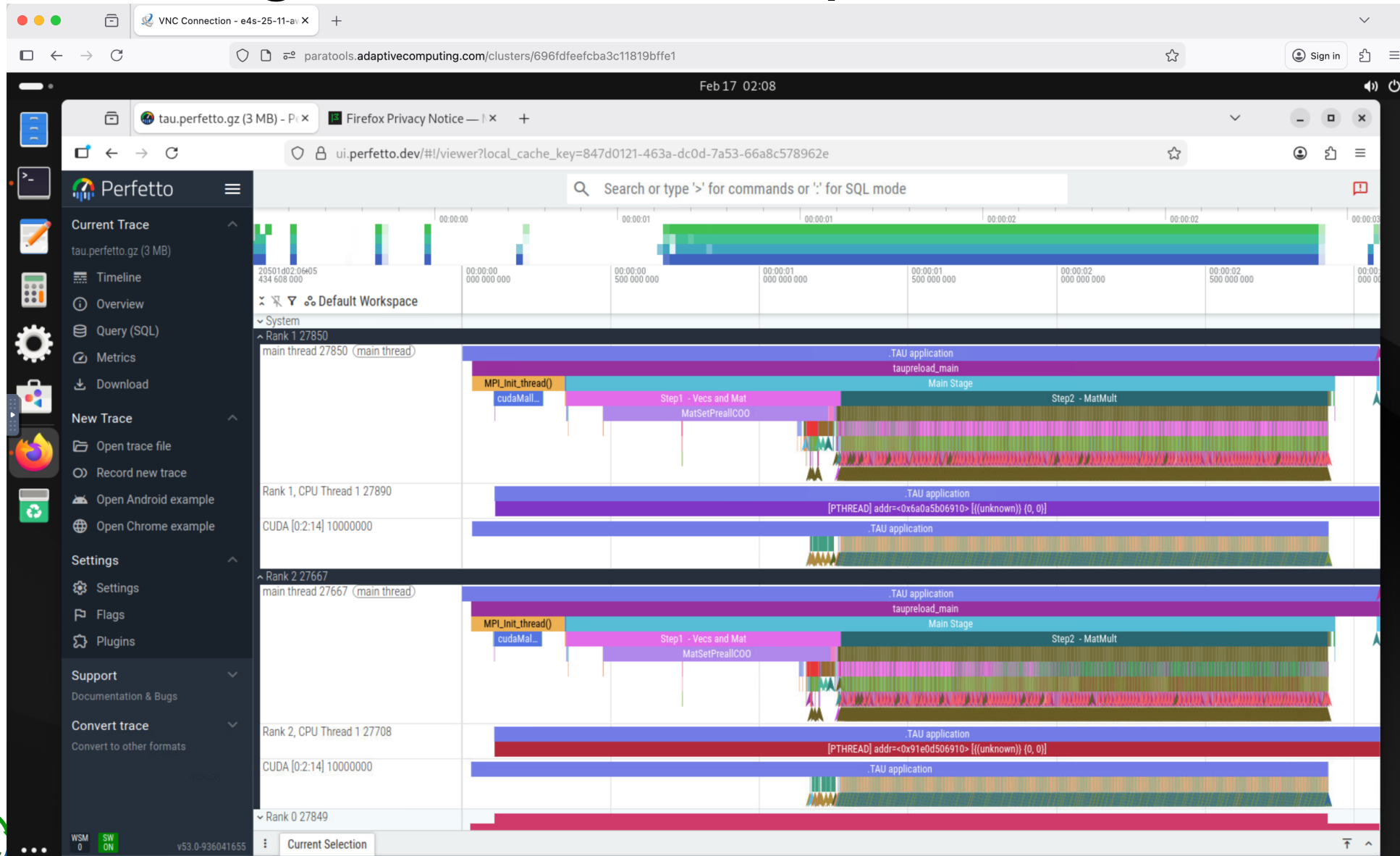
The user then runs `sbatch tau.sbatch`, `ls tau.perfetto.gz`, and `firefox https://perfetto.dev`.

Add
`export TAU_TRACE=1`
`export TAU_TRACE_FORMAT=`
`perfetto`

Generating Traces

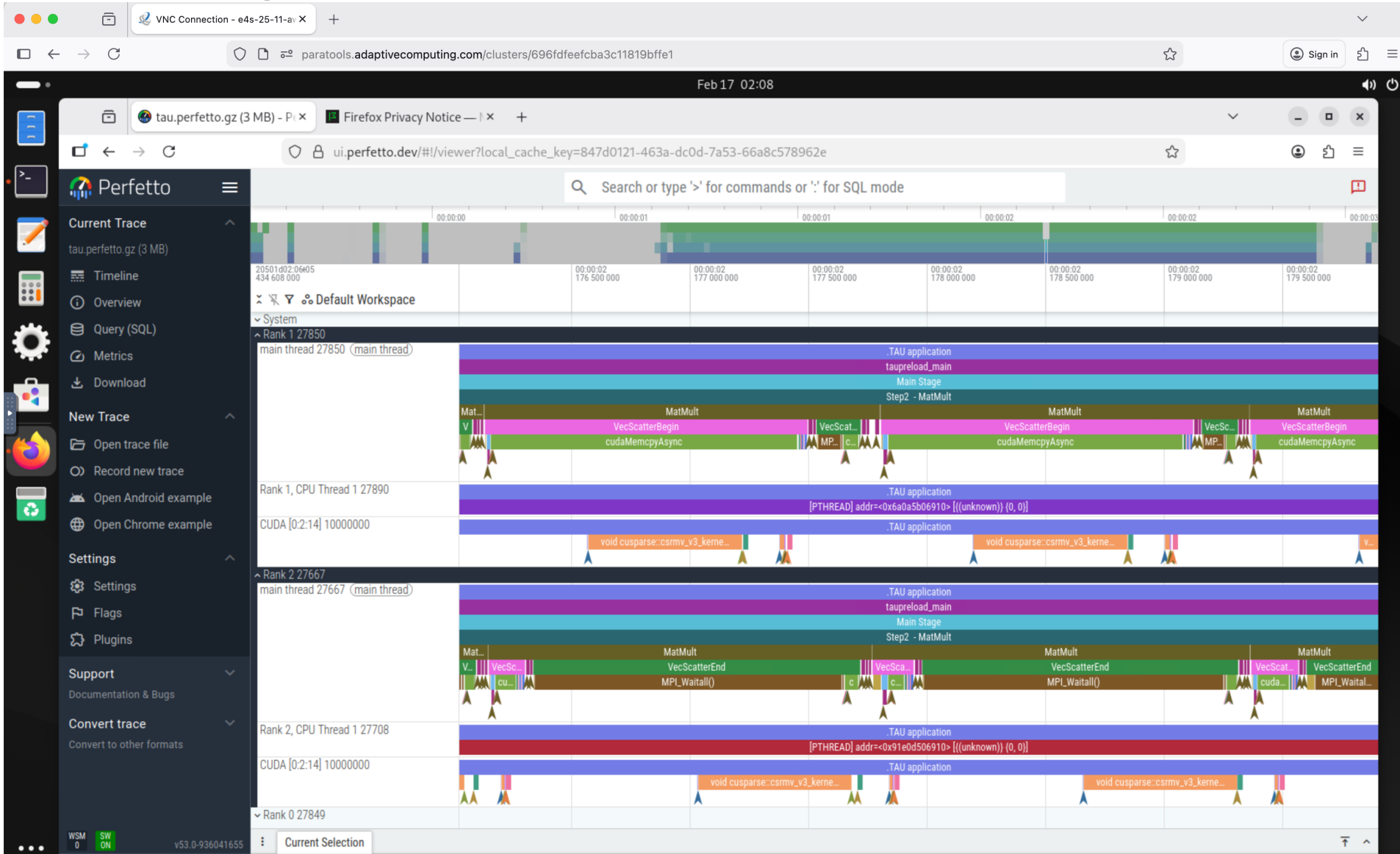
```
% sbatch tau.sbatch  
% firefox https://perfetto.dev &  
Click -> Trace Viewer -> Open the UI ->  
Open Trace File -> navigate to tau.perfetto.gz  
Open the four trace rows (one for each rank)  
Use wasd keys to widen/shrink/left/right  
scroll
```

Visualizing Traces with https://Perfetto.dev



wasd
W = widen
S = Shrink
A = Left
D = Right

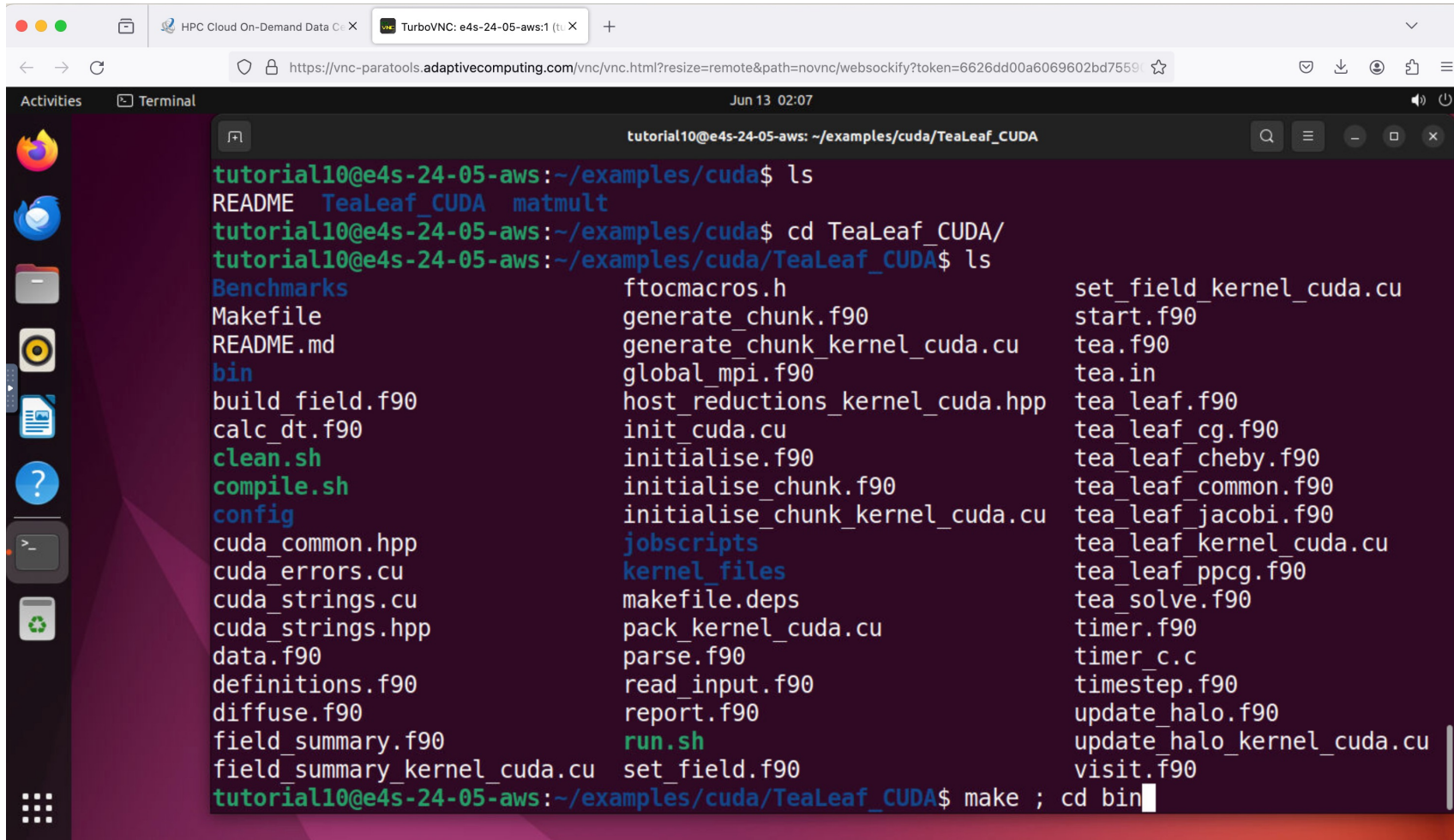
Visualizing Traces with https://Perfetto.dev



TAU Exercise #5:

TeaLeaf application

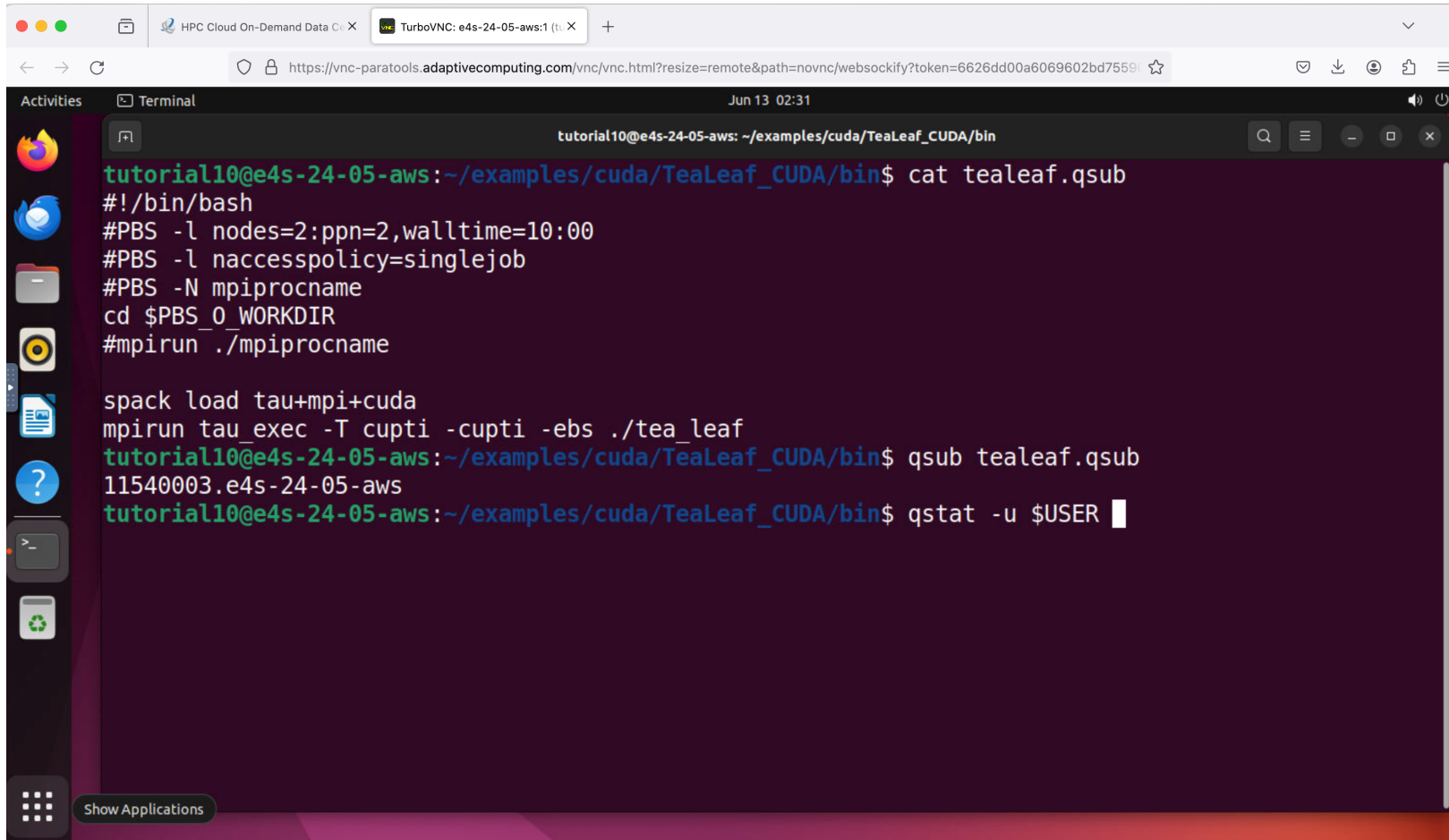
Compiling TeaLeaf_CUDA application



```
tutorial10@e4s-24-05-aws: ~/examples/cuda/TeaLeaf_CUDA
tutorial10@e4s-24-05-aws:~/examples/cuda$ ls
README  TeaLeaf_CUDA  matmult
tutorial10@e4s-24-05-aws:~/examples/cuda$ cd TeaLeaf_CUDA/
tutorial10@e4s-24-05-aws:~/examples/cuda/TeaLeaf_CUDA$ ls
Benchmarks      ftocmacros.h      set_field_kernel_cuda.cu
Makefile         generate_chunk.f90  start.f90
README.md        generate_chunk_kernel_cuda.cu  tea.f90
bin              global_mpi.f90      tea.in
build_field.f90  host_reductions_kernel_cuda.hpp  tea_leaf.f90
calc_dt.f90      init_cuda.cu        tea_leaf_cg.f90
clean.sh         initialise.f90       tea_leaf_cheby.f90
compile.sh        initialise_chunk.f90  tea_leaf_common.f90
config           initialise_chunk_kernel_cuda.cu  tea_leaf_jacobi.f90
cuda_common.hpp  jobscripts          tea_leaf_kernel_cuda.cu
cuda_errors.cu   kernel_files         tea_leaf_ppcg.f90
cuda_strings.cu  makefile.deps        tea_solve.f90
cuda_strings.hpp  pack_kernel_cuda.cu  timer.f90
data.f90         parse.f90            timer_c.c
definitions.f90  read_input.f90       timestep.f90
diffuse.f90      report.f90           update_halo.f90
field_summary.f90 run.sh              update_halo_kernel_cuda.cu
field_summary_kernel_cuda.cu  set_field.f90      visit.f90
tutorial10@e4s-24-05-aws:~/examples/cuda/TeaLeaf_CUDA$ make ; cd bin
```

```
cd ~/examples/cuda
cd TeaLeaf_CUDA
ls
make
cd bin
```

Submit the CUDA job to run on four MPI ranks



The screenshot shows a terminal window titled 'tutorial10@e4s-24-05-aws: ~/examples/cuda/TeaLeaf_CUDA/bin'. The user has executed the following commands:

```
tutorial10@e4s-24-05-aws:~/examples/cuda/TeaLeaf_CUDA/bin$ cat tealeaf.qsub
#!/bin/bash
#PBS -l nodes=2:ppn=2,walltime=10:00
#PBS -l naccesspolicy=singlejob
#PBS -N mpirocname
cd $PBS_O_WORKDIR
#mpirun ./mpiprocname

spack load tau+mpi+cuda
mpirun tau_exec -T cupti -cupti -ebs ./tea_leaf
tutorial10@e4s-24-05-aws:~/examples/cuda/TeaLeaf_CUDA/bin$ qsub tealeaf.qsub
11540003.e4s-24-05-aws
tutorial10@e4s-24-05-aws:~/examples/cuda/TeaLeaf_CUDA/bin$ qstat -u $USER
```

sbatch tau.sbatch

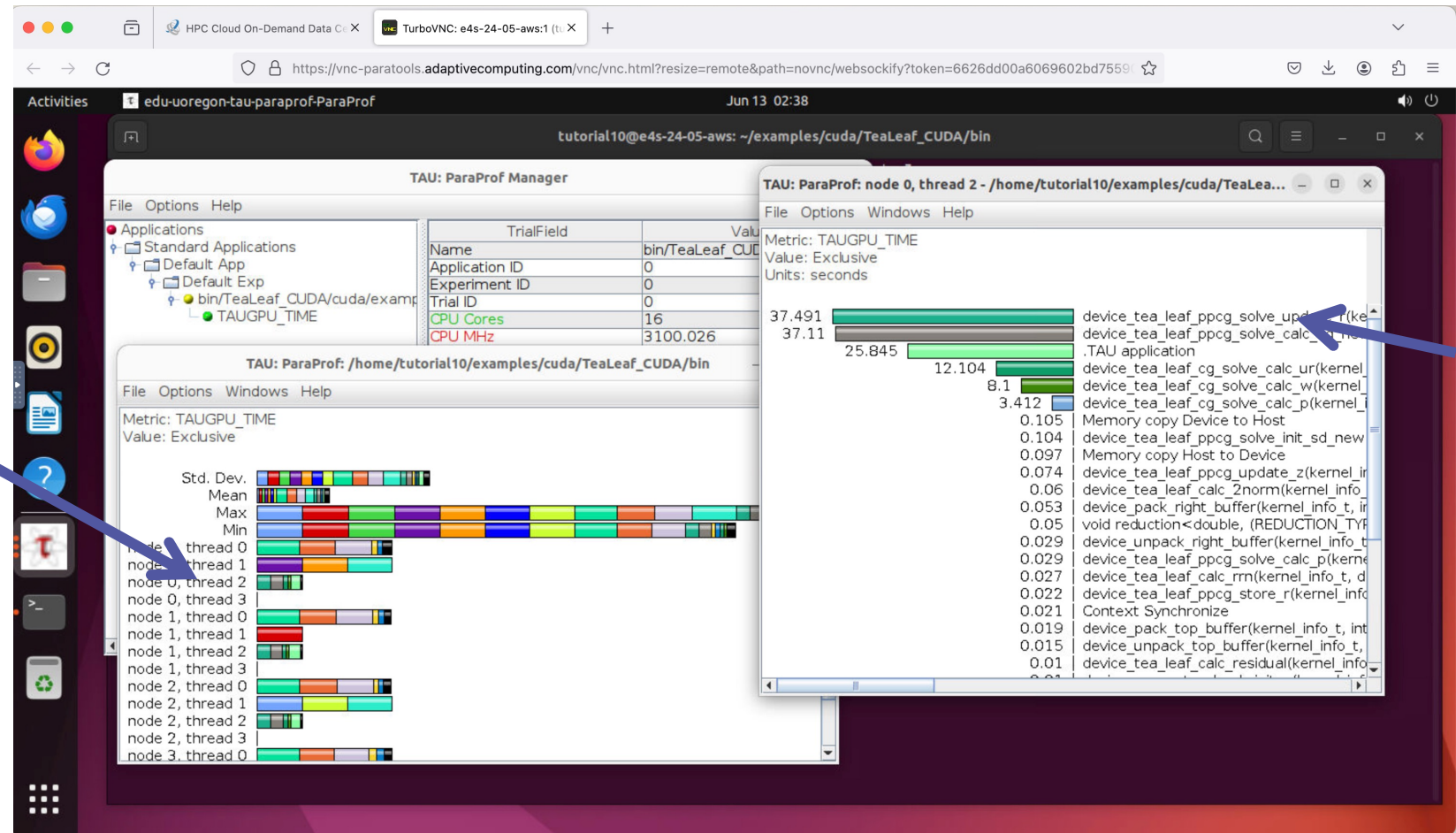
queue -u \$USER

After it runs

paraprof &

TAU's paraprof shows the time spent in individual CUDA kernels

Left click node 0, thread 2



CUDA kernels

TAU Exercise #6:

paraprof 3D display

TAU paraprof

The screenshot shows a terminal window with the following commands and output:

```
tutorial10@e4s-24-05-aws: ~/examples/tau$ ls
demo.ppk  fetch.sh
tutorial10@e4s-24-05-aws: ~/examples/tau$ paraprof demo.ppk &
[1] 309831
tutorial10@e4s-24-05-aws: ~/examples/tau$
```

Below the terminal, the TAU ParaProf Manager window is open, displaying a table of trial fields for the demo.ppk application:

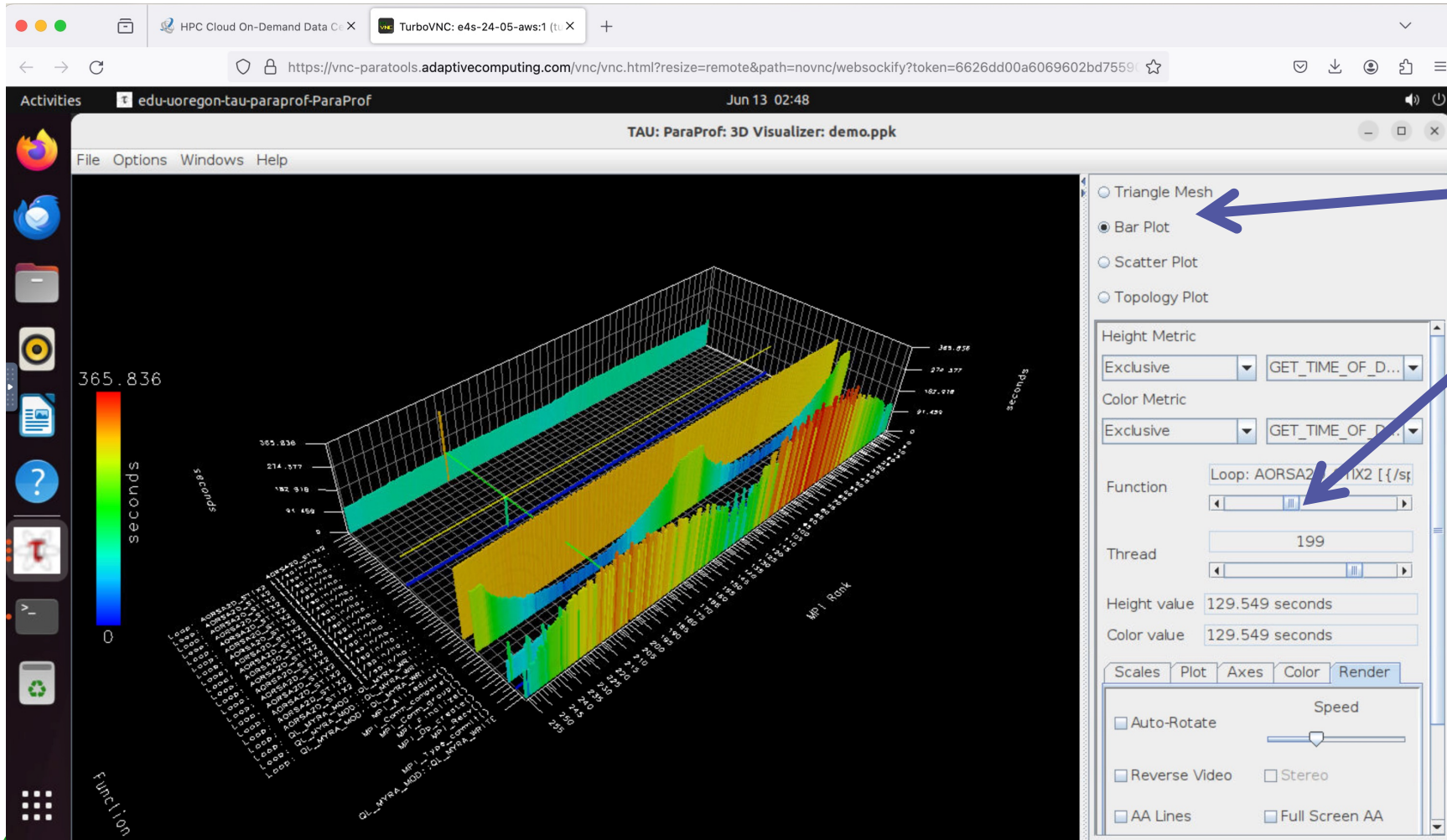
TrialField	
Name	demo.ppk
Application ID	0
Experiment ID	0
Trial ID	0
File Type Index	0
File Type Name	ParaProf Pa

Overlaid on the manager window is a 'TAU: ParaProf: demo.ppk' window showing a 3D visualization of the application's performance. A blue arrow points to the '3D Visualization' option in the 'Windows' menu of this window.

cd ~/examples/tau
paraprof demo.ppk &

Choose 3D Visualization

TAU paraprof 3D visualization

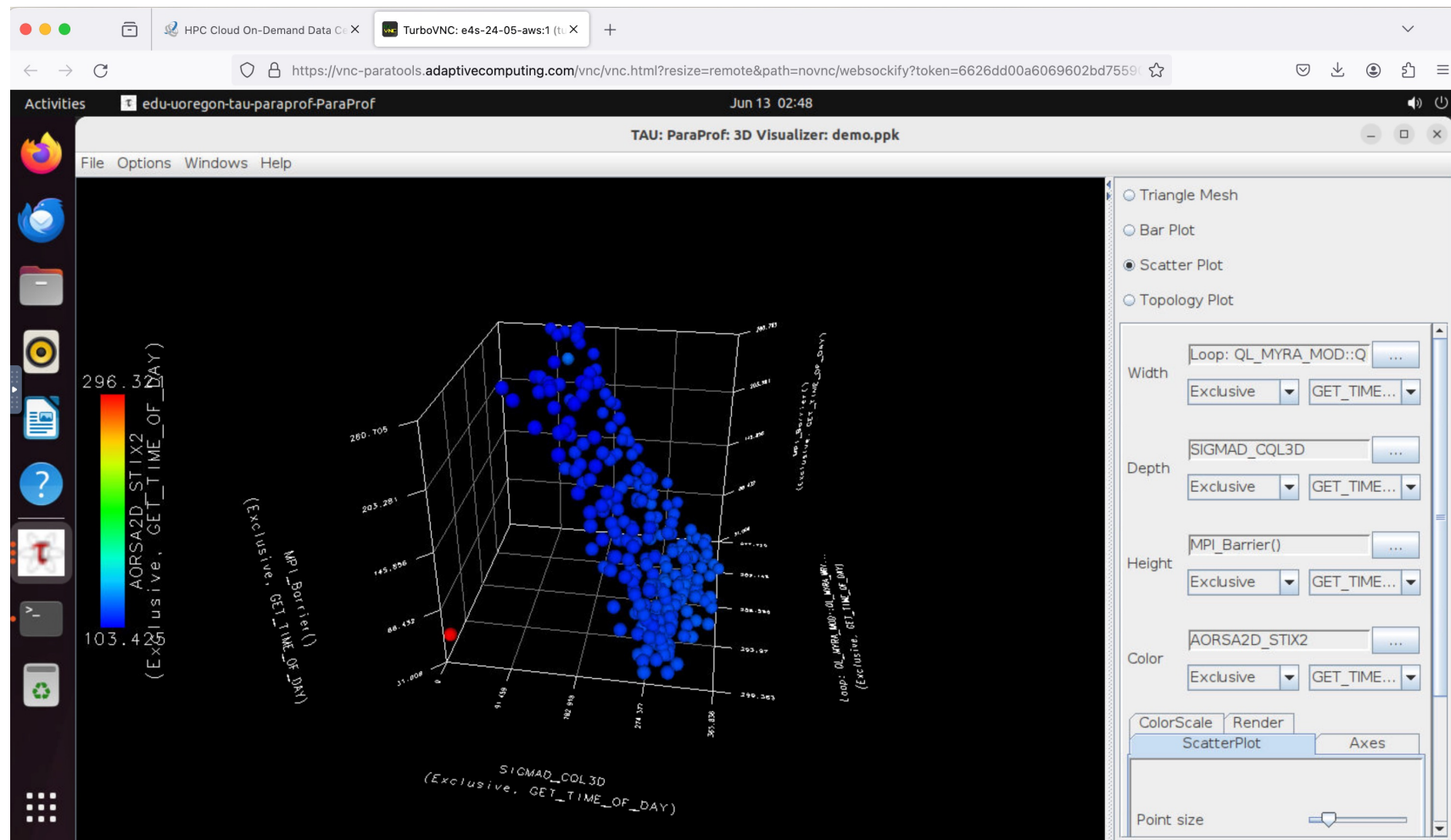


Choose Bar Plot and move
Function and Thread
Sliders

First mouse button to rotate
Second mouse button to
translate (left to right)
Scroll wheel (or +/- keys) to
zoom in.

Try Scatter plot next

TAU paraprof: 3D Scatter Plot



TAU Exercise #7:

Julia instrumentation

TAU paraprof



TAU Instrumentation
API Bindings

LLVM Plugin

Distributed.jl
Instrumentation

CUPTI
rocp
oneAPI IO

Manual
Instrumentation

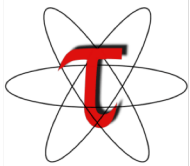
Automatic
Instrumentation

GPU Kernel
Instrumentation

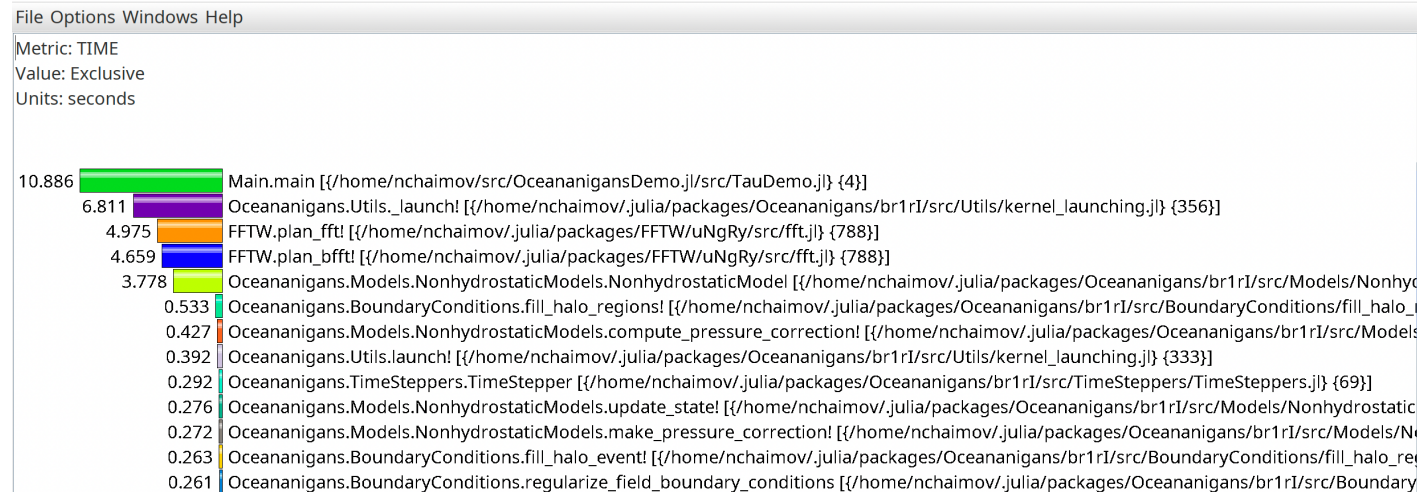
Distributed
Library
Instrumentation

`cd ~/examples/tau/julia`

Launch all four examples
with `tau_julia`
`paraprof`



TAU: ParaProf: node 0, thread 0 - OceananigansCPU.ppk



E4S: An HPC-AI Software Ecosystem for Science!

The screenshot displays a terminal window with the following commands and output:

```
Singularity> spack find +cuda
-- linux-ubuntu22.04-x86_64_v3 / gcc@11.4.0 -----
adios2@2.10.2      camp@2024.07.0  flux-core@0.67.0  kokkos@4.5.01    libceed@0.12.0  petsc@3.22.4    tasmanian@8.1     zfp@1.0.0
amrex@25.03       camp@2024.07.0  ginkgo@1.9.0      kokkos@4.6.01    magma@2.9.0     raja@2024.07.0  tau@2.34.1
arborx@1.5        chai@2024.07.0  gromacs@2024.4     kokkos@4.6.01    mfem@4.7.0      raja@2024.07.0  trilinos@16.1.0
axom@0.10.1       chapel@2.4.0    heffte@2.4.1      kokkos@4.6.01    mgard@2023-12-09 slate@2024.10.29 umpire@2024.07.0
blaspp@2024.10.26 cp2k@2025.1     hpctoolkit@2024.01.1 kokkos-kernels@4.6.01 nvcomp@2.2.0    slepc@3.22.2     umpire@2024.07.0
bricks@2023.08.25 cusz@0.14.0     hpx@1.10.0        lammmps@20240829.1 papi@7.1.0      strumpack@8.0.0  umpire@2024.07.0
cabana@0.7.0      fftx@1.2.0     hwloc@2.11.1      lapackpp@2024.10.26 parsec@3.0.2209 sundials@7.2.1   upcxx@2023.9.0
caliper@2.12.1    flecsi@2.3.2   hypre@2.32.0      legion@24.12.0   petsc@3.22.4    superlu-dist@9.1.0 vtk-m@2.2.0
==> 57 installed packages
Singularity> paraprof demo.ppk &
```

The VS Code editor shows a Python script named `foo.py` with the following content:

```
26 # Untar and convert .sph to .wav (using sox)
27 tar = tarfile.open(an4_path)
28 tar.extractall(path=data_dir)
29
30 print("Converting .sph to .wav...")
31 sph_list = glob.glob(data_dir + '/an4/**/*.sph', recursive=True)
32 for sph_path in sph_list:
33     wav_path = sph_path[:-4] + '.wav'
34     cmd = ["sox", sph_path, wav_path]
35     subprocess.run(cmd)
36 print("Finished conversion.\n*****")
37
38 # NeMo's "core" package
39 import nemo
40 # NeMo's ASR collection - this collections contains complete ASR models and
41 # building blocks (modules) for ASR
42 import nemo.collections.asr as nemo_asr
43
44 # This line will download pre-trained QuartzNet15x5 model from NVIDIA's NGC cloud
45 quartznet = nemo_asr.models.EncDecCTCModel.from_pretrained(model_name="QuartzNet15x5")
46
47 files = [os.path.join(data_dir, 'an4/wav/an4_clstk/mgah/cen2-mgah-b.wav')]
48 for fname, transcription in zip(files, quartznet.transcribe(audio=files)):
49     print(f"Audio in {fname} was recognized as: {transcription}")
50
```

The 3D visualization shows a surface plot of the script's output, with a color bar indicating values from 0 to 365.836 seconds. The plot is titled "TAU: ParaProf: 3D Visualizer: demo.ppk (on illyad)". Below the plot, a table shows the minimum values for each node:

Node	Min
node 0	0.000
node 1	0.000
node 2	0.000
node 3	0.000

E4S CI using GPU Clusters on Frank@UO



Category: Servers - OACISS System: X

oaciss.uoregon.edu/frank

Category: Discussion

Read View source View history Search OACISS Systems Wiki

Category: Servers

This is a list of all OACISS servers in the Frank cluster at UO. You may also select a section header to view the Wiki-generated category index for systems of that type.



Some relevant pages:

- [Introduction](#) and about -> click me
- The [NetworkInfrastructure](#) page describes the host naming (dns) conventions, as well as documenting the physical setup and connections within the OACISS racks in the machine room. All OACISS systems automatically search .nic.uoregon.edu for DNS, so only the short hostname is needed for ssh internally.
- The [Service storage](#) describes available storage for users of OACISS systems. OACISS currently has a total of just under 350TB of online storage available.
- The new [HowtoMPI](#) page describes various tested-working MPI setups and the steps

Click on the server links to access more information about individual machines. Note that only the three machines designated as login gateways (orthus, sphinx, cerberus) are accessible by machines outside of nic.uoregon.edu.

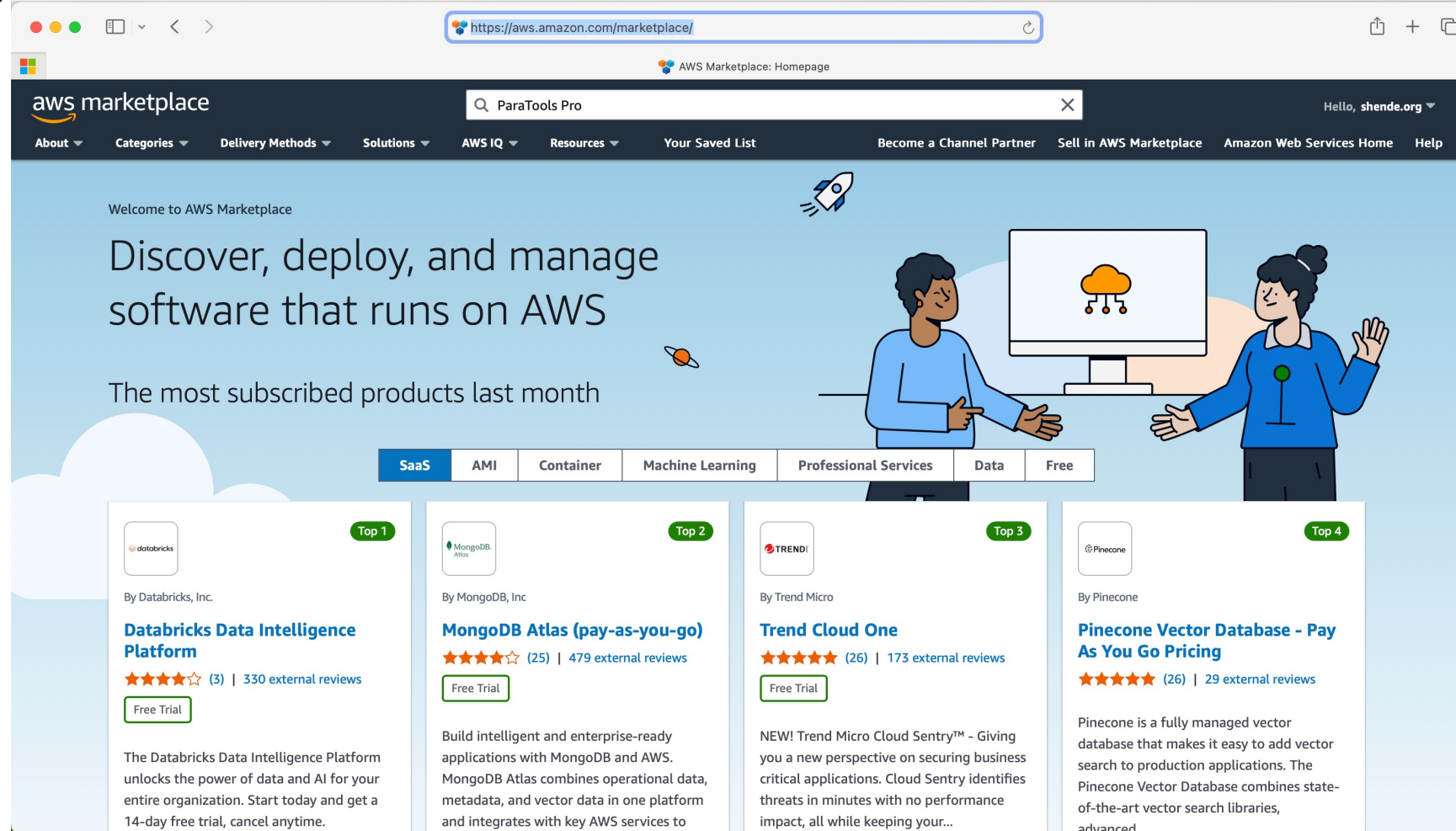
IMPORTANT: Cerberus is in the process of being decommissioned and replaced with a new and much more capable system, Sphinx. Sphinx is the new secondary login node.

Nodes in Computing Center datacenter						
Name	Description	OS	Model	Processors	Local Network	Physical location
Compute: Orthus	Primary login gateway	RHEL-8.8	VM	6 x Cascade Lake	10GbE	
Compute: Stravinsky	Xeon 6gen system + NVIDIA RTX 6000 Pro Blackwell	Ubuntu 24	CG480-S5063	2 x 6767P Granite Rapids	10GbE	R8a.U29
Compute: Odyssey	AMD quad MI300A system	RHEL 8.10	SM AS-4145-GT	4 x MI300a, 24cpu	10GbE	R81.U27
Compute: Pinwheel	2xMI210 Debian 12 system	Debian 12	SM AS-2024US-TRT	2 x Epyc Milan 7413	10GbE	R81.U32
Compute: Headroom	Intel Data Center Max 1100 (Pontie Vecchio)	Ubuntu 22	Supermicro	2 x Xeon 4410T @ 3GHz	10GbE	R82.U8
Compute: Hopper1/2	NVIDIA Grace-Hopper GH200	RHEL 9.4	Quanta S74G-2U	Grace (72c)+GH200 Hopper GPU @ 3.4GHz	10GbE + Connect X-7 200 Gbps (pt2pt w/hopper1/2.eth)	R81.U33-36
Compute: Grace1/2	NVIDIA Grace-Grace	Ubuntu 22.04 LTS	Supermicro AFS-221GL-NR01 (x2)	Grace-Grace (144c) Superchip @ 3.4GHz	10GbE + EDR 100 Gbps (pt2pt w/grace1/2)	R81.U4-8
Compute: Roberta	HPE CPE Epyc Genoa	RHEL 8.10	DL385 gen 11	2 x Epyc 9654 Genoa 96c @ 2.4GHz	10GbE + NDR 400 Gbps (pt2pt w/gary)	R82.U6
Compute: Gary	HPE CPE Epyc Genoa	RHEL 8.10	DL345 gen 11	Epyc 9124 Genoa 16c @ 3GHz	10GbE + NDR 400 Gbps (pt2pt w/roberta))	R82.U4
Compute: Picard	Atipa Sapphire Rapids + 2 A2000	RHEL 8.10	R283-S91	2 x 32c Xeon 6430 @ 3.4GHz	100GbE	R82.U6
Compute: Athena	4x100 nlink box	RHEL 8.10	Gigabyte RS292	2 x Epyc Milan 7763 5	10GbE	R85.U39
Compute: Ilyad	AMD + 1 H100 (80GB) + 2 MI210	RHEL 8.10	Preproduction SuperMicro	2 x 64c Epyc Milan @ 2.8GHz	100GbE + 200G IB	R85.U22
Compute: Gilgamesh	AMD + 2 MI210	RHEL 8.10	Preproduction SuperMicro	2 x 24c Epyc Milan 7413 @ 2.6GHz	100GbE + 200G IB	R85.U26
Compute: Jupiter	Quad Cooper lake + Intel DG1	Ubuntu 22.04.4	Supermicro Sys-240	4 x 24c Xeon Gold 6438 @ 2.3GHz	100GbE + EDR	R86.U10
Compute: Saturn	Quad Ice lake + A100 (80GB) + 2 A100 (40GB)	Ubuntu 22.04.4	Gigabyte RS292-4S1	4 x 26c Xeon Platinum 8367HC @ 3.2GHz	100GbE + EDR	R86.U10
Compute: Omnia	2 x 8352Y CPU + AMD MI210 + Intel DG1 + 2TB Optane-200	Ubuntu 22.04	Supermicro SYS-220U-TNR	2 x 36 core 8352Y	40GbE	R84.U1

<https://oaciss.uoregon.edu/frank>

Using your own cloud credentials with Adaptive Computing's Heidi

STEP 1: Subscribe to ParaTools Pro for F4S image on AWS Marketplace



Go to <https://aws.amazon.com/marketplace>
Login, then search for ParaTools Pro

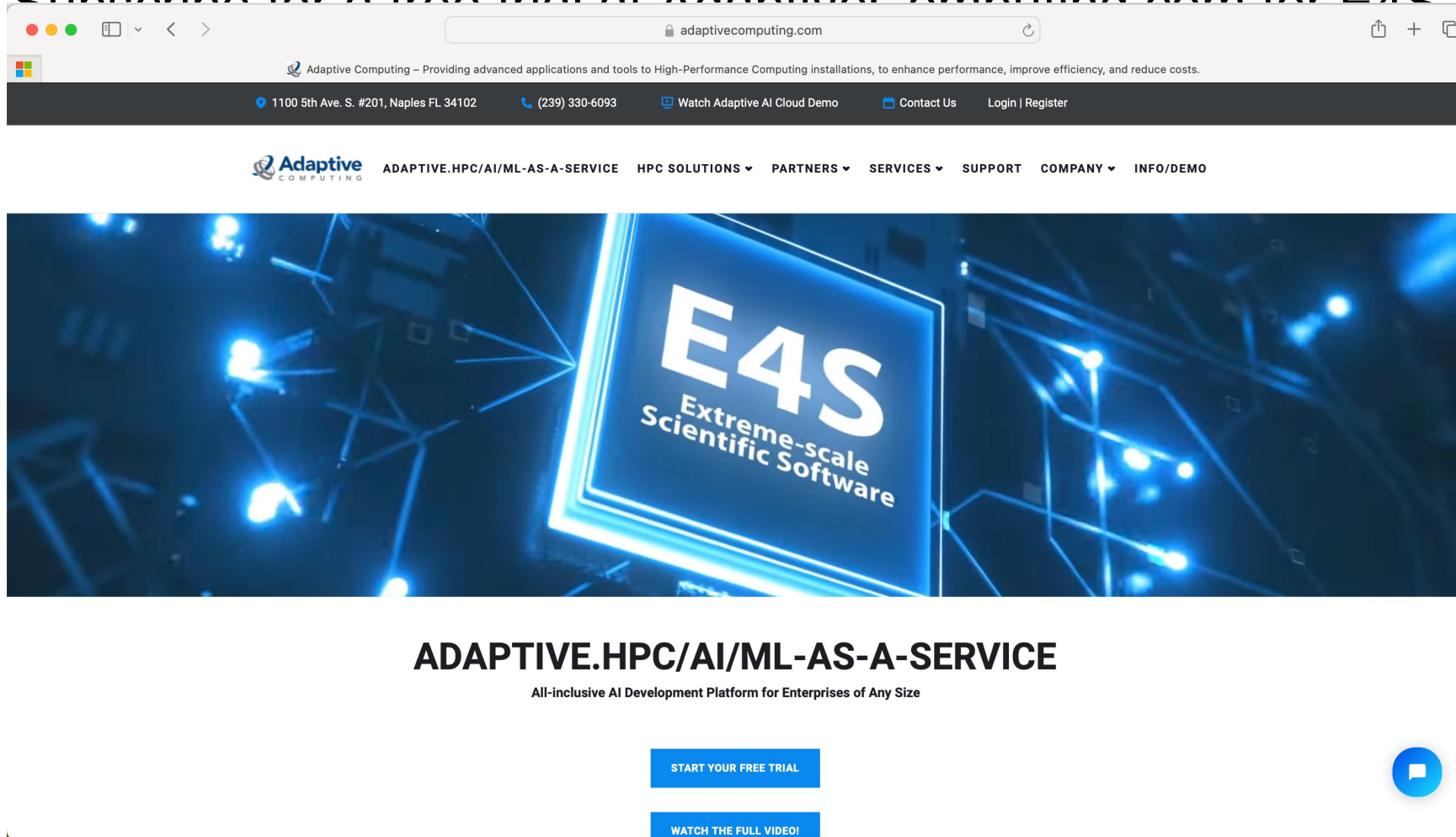
STEP 1: Subscribe to ParaTools Pro for E4S image on AWS Marketplace

Free trial for
31 days!

Click on ODDC Server (x86) click subscribe

Click on ODDC Node (x86) click subscribe

STEP 2: Subscribe for a free trial at AdaptiveComputing.com for E4S



Go to <https://adaptivecomputing.com>
Click on "Start your free trial"

STEP 2: Subscribe for a free trial on Adaptive Computing for E4S

adaptivecomputing.com

Request Trial - Adaptive Computing

1100 5th Ave. S. #201, Naples FL 34102 (239) 330-6093 Watch Adaptive AI Cloud Demo Contact Us Login | Register

Adaptive COMPUTING ADAPTIVE.HPC/AI/ML-AS-A-SERVICE HPC SOLUTIONS PARTNERS SERVICES SUPPORT COMPANY INFO/DEMO

REQUEST TRIAL

You Are Here: [HOME](#) / [REQUEST TRIAL](#)

Please fill out the form below to request a trial. A team member will contact you within 24 hours. Thank you for your interest in Adaptive Computing!

Full Name (required)

Email (required)

Phone Number (required)

Company Name (required)

Job Title (optional)

Please sign me up for the following trial(s)

☐ adaptive.ai-as-a-service ☐ ODDC (HPC On-Demand Data Center) ☒ E4S

I would like to:

☐ Request a sales call ☐ Get more information

Your Message (optional)

What else should we know?

Fill out the form and check the E4S box

Conclusions

- Using E4S containers saves time
- Customize full featured or base containers using e4s-alc
- Substitute MPI in the container with your system MPI using e4s-cl
- Run on commercial cloud platforms

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Next-Generation Scientific Software Technologies program, under contract numbers DE-AC02-06CH11357, DE-AC05-00OR22725, DOE SBIR DE-SC0022502, Sandia National Laboratories Contract PO 2542488, and NSF OAC award 2311831.



U.S. DEPARTMENT OF
ENERGY

Office of
Science



- <https://science.osti.gov/ascr>
- <https://pesoproject.org>
- <https://ascr-step.org>
- <https://hpsf.io>
- <https://www.energy.gov/technologytransitions/sbirsttr>



