# *Human-Oriented Discussions on Performance Analysis*

Scalable Tools Workshop ‘23

Connor Scully-Allison

https://bit.ly/stw-humans

# Participants

- Connor Scully-Allison (The University of Utah)
- Kate Isaacs (The University of Utah)
- Wileam Phan (Rice University)
- Sameer Shende (University of Oregon)
- Jan Laukemann (I'm University of Erlangen-Nuremberg / NHR@FAU)
- Shadmaan Hye (University of Utah)
- David Boehme (LLNL)
- William Jalby (UVSQ/UPSaclay)
- Stephanie Brink (LLNL)
- Kathleen Shoga (LLNL)
- Rahat Zaman (University of Utah)
- David Montoya (Trenza)
- Mahesh Rajan (Trenza)

**https://bit.ly/stw-humans**

# Working Group Goals

- What groups of users exist in the performance tools space?
- Who are they?
  - What do we know about them now?
  - What do we still need to know?
  - How do they differ from us?
- What can we do to understand them better?

**https://bit.ly/stw-humans**

# Exercise: Who is your user?

- Code developers are the people we expect to use the tools. Code runners are the people who generate the configurations and performance problems. There isn't always a good channel for the configurations/data to pass back to the developers.
- Code developers are encouraged to add features over performance.
- Code developer teams need a performance champion — or tool developer champions need to come in and set them up and create internal experts.
- People don't want to work with performance tools unless they absolutely need them but then there's a learning curve. Currently community looking into encouraging them to look at performance continuously.
- There are probably 5-6 kinds of users. Which should we prioritize.
- Users in private sectors can be very secretive about source code access and could benefit from tools that perform analysis without any annotation
- Not all users are code developers. Many are domain experts and need to relate performance data back to the domain execution

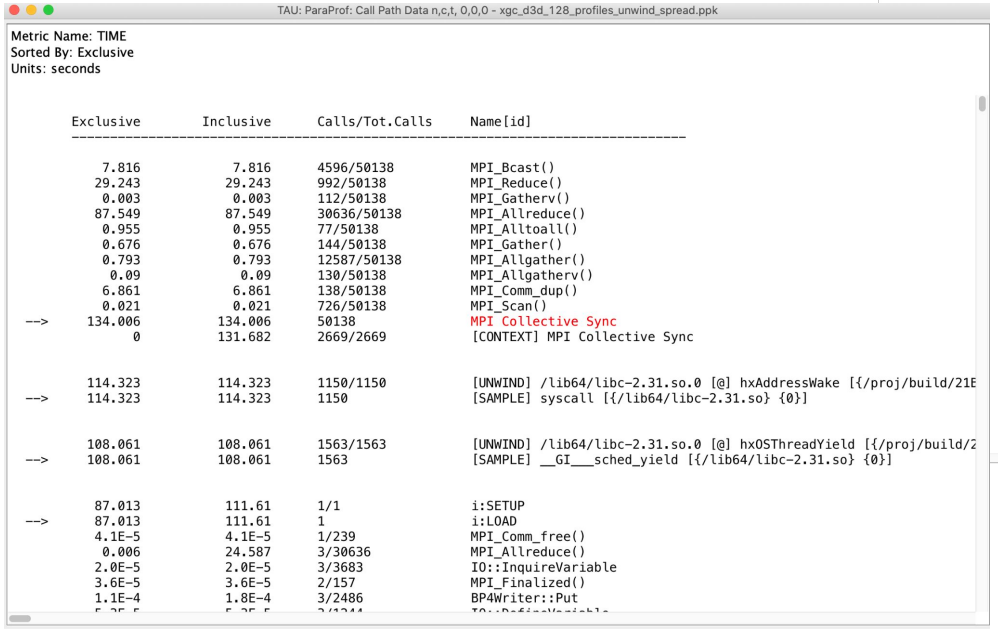
**https://bit.ly/stw-humans**

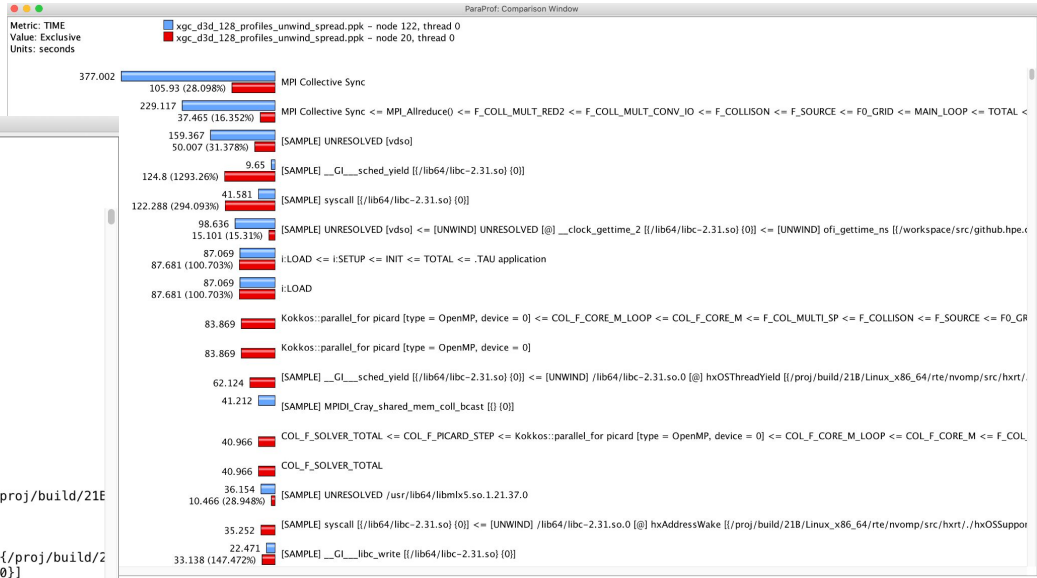
# Exercise: How do you differ?

- Tool developers have comparatively structured workflows for performance analysis compared to their users
  - With significant foundation understanding of where performance issues commonly lie
- Tool developers are frustrated by the performance and usability of visualization in tools
  - “Zooming on a trace should be like google maps”
- Legacy data formats hold us back from performant GUI development

**https://bit.ly/stw-humans**

# TAU performance data explains why MPI collectives take as long as they do!

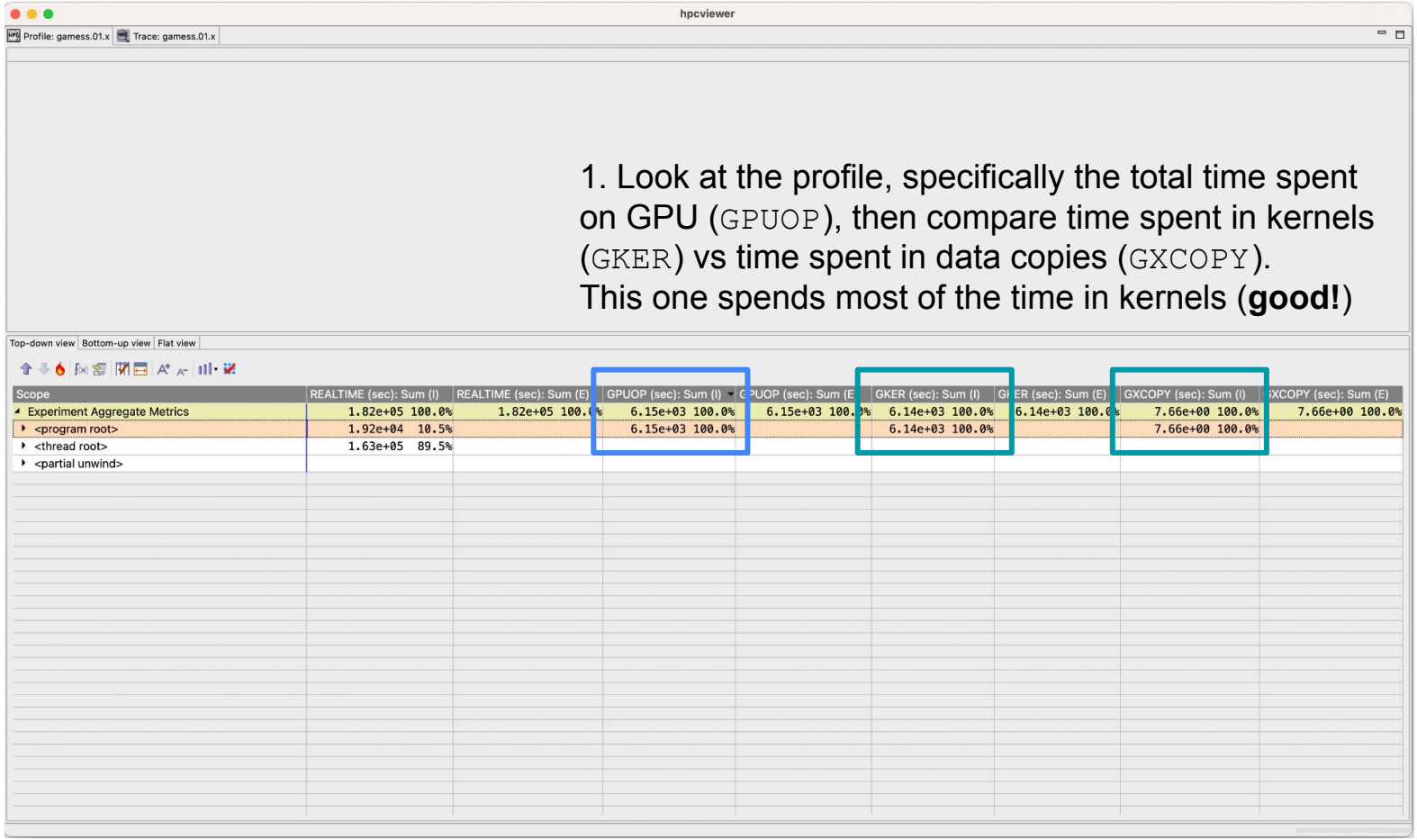



Time spent in MPI, MPI_Barrier inside the collective operation (MPI Collective Sync), load imbalance, time spent in Kokkos, in application functions, I/O, sampling, system calls.

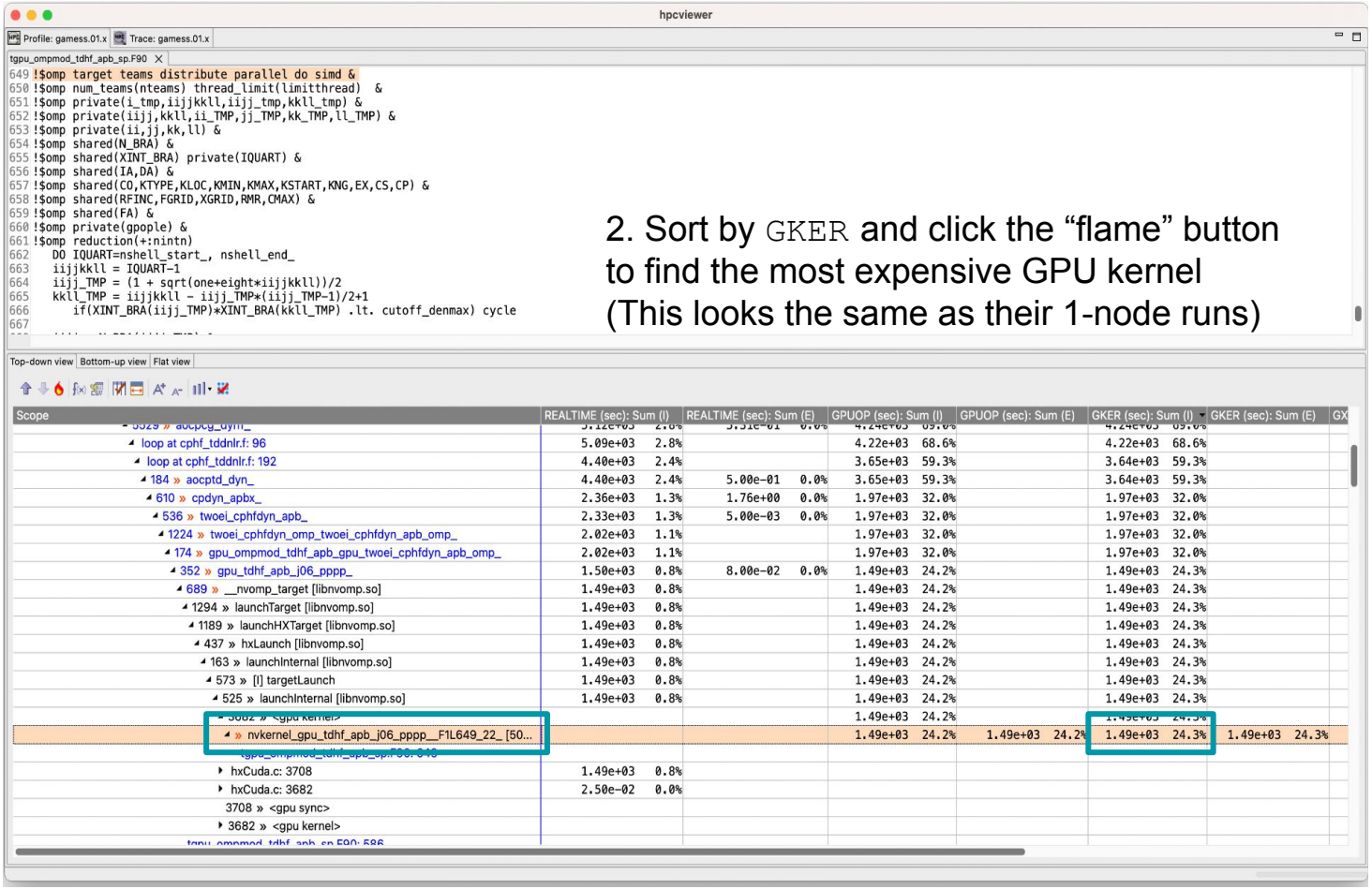
**https://bit.ly/stw-humans**

# HPCToolkit analysis of GAMESS 5-node run on Perlmutter




https://bit.ly/stw-humans

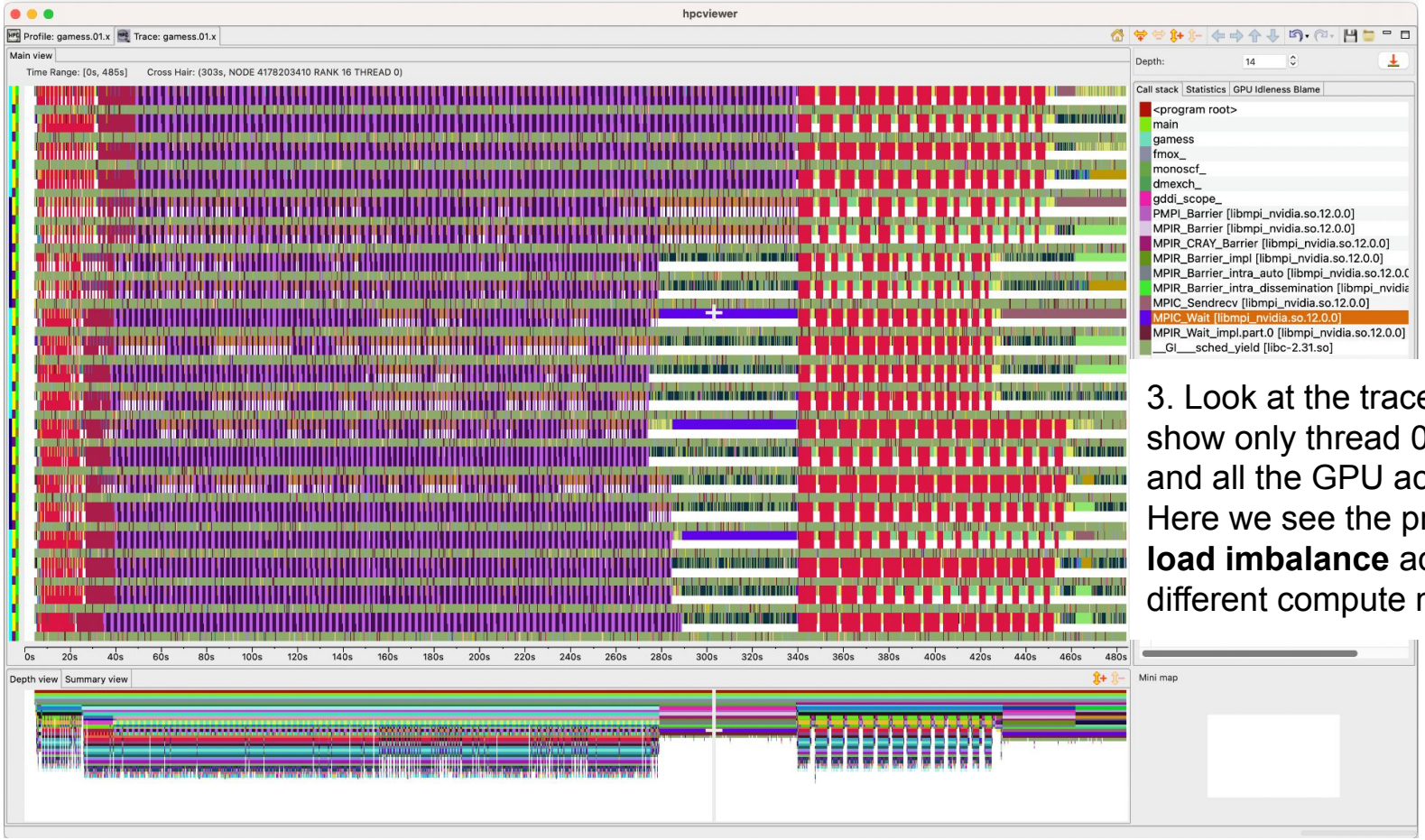
# HPCToolkit analysis of GAMESS 5-node run on Perlmutter




2. Sort by `GKER` and click the “flame” button to find the most expensive GPU kernel (This looks the same as their 1-node runs)

**https://bit.ly/stw-humans**

# HPCToolkit analysis of GAMESS 5-node run on Perlmutter




3. Look at the trace, filter to show only thread 0 on CPU and all the GPU activity. Here we see the problem is **load imbalance** across different compute nodes

**https://bit.ly/stw-humans**

# What else should we be asking? What do we want to know about them?

- How often is analysis being done?
- Granularity of what they are looking for/ their interest.
  - “See everything at absolutely no overhead”
- Where does a user lie on the interactive non interactive analysis spectrum
- Users are interested in seeing data related to abstractions that they are familiar with
  - Annotations that they have inserted into the code
  - “Kokkos parallel for picard”
  - They want to relate the performance to the domain application function

**https://bit.ly/stw-humans**

# What actionable steps can we take to understand our users better?

- Provide clear use cases/samples alongside API Documentation
  - **Quickstart for <class of user>**
    - This makes it transferable between different users on the same team
    - Provide a template github repo alongside documentation
    - Keep tutorials short
- Acquaint new users with the diversity of commonly used tools
  - There is few opportunities to try new tools and they are often driven by problems
  - **A reference document of some variety that guides tool developers**
    - **Provide pros and cons of the tools**
- We need to measure user workflows and integrate them better.
  - **HCI experts can help here**

**https://bit.ly/stw-humans**