Advanced Parallel Debugging with TotalView

Hands On Session

ARGONNE
ATPESC2023
EXTREME-SCALE COMPUTING

Bill Burns | August 2023

Senior Director of Software Engineering and Product Manager

Agenda

- Compiling Your Program for Debugging
- Interactive Debugging with TotalView on ANL’s Systems
- TotalView Labs
- Questions and Answers

TotalView

Compiling Your Program for Debugging

Compile for Debugging

Debugging Compiler Flags

- The “-g” flag is required to turn on generation of debug information required by the debugger to find variables and source.
- Low level of optimization is recommended, -O0 or -O1, for the best debugging experience
  - Successful debugging of highly optimized programs, -O2 or -O3, depends on quality of compiler generated debug information

MPI Example

- mpicc **-g** -w mpi_array.c -o mpi_array
- mpicxx **-g** -w mpi_array.cpp -o mpi_array
- mpifort **-g** -w mpi_array.f -o mpi_array

CUDA Example

- When compiling an NVIDIA CUDA program for debugging, it is necessary to pass the -g -G options to the nvcc compiler driver. These options disable most compiler optimization and include symbolic debugging information in the driver executable file, making it possible to debug the application.
  - % /usr/local/bin/nvcc **-g** **-G** -c tx_cuda_matmul.cu -o tx_cuda_matmul.o
  - % /usr/local/bin/nvcc **-g** **-G** -Xlinker=-R/usr/local/cuda/lib64 tx_cuda_matmul.o -o tx_cuda_matmul

Interactive Debugging with TotalView on ANL’s Systems

TotalView is available on Theta, ThetaGPU, and Polaris

- Installed at:
  - theta: `/soft/debuggers/totalview-2023-02-15/toolworks/totalview.2023.2.15/bin/totalview`
  - thetagpu: `/soft/thetagpu/debuggers/totalview-2023-02-15/toolworks/totalview.2023.2.15/bin/totalview`
  - polaris: `/soft/debuggers/totalview-2023-02-15/toolworks/totalview.2023.2.15/bin/totalview`
- Set up to use TotalView (no module for latest version set up):
  - theta: `source /soft/debuggers/totalview-2023-02-15/useTVonTheta.sh`
  - thetagpu: `source /soft/thetagpu/debuggers/totalview-2023-02-15/useTVonTheta.sh`
  - polaris: `source /soft/debuggers/totalview-2023-02-15/useTVonPolaris.sh`
- Make sure module xalt is unloaded (it should be unloaded automatically when TotalView “use” script is run)
  - `module unload xalt`

Debugging with TotalView through VNC Remote Desktop

Set up remote desktop using VNC

- VNC provides an efficient remote graphical desktop comprised of a display server and a viewer
- On Theta login node start vncserver
  - vncserver -geometry 1800x970 :**25**
  - Creates VNC server with display port 25
- On local machine open SSH tunnel to login node
  - Use the display port for the VNC server
  - ssh -L 59**25**:*thetalogin4*:59**25** bburns@theta.alcf.anl.gov
- On local machine start vncviewer
  - vncviewer localhost:**25**

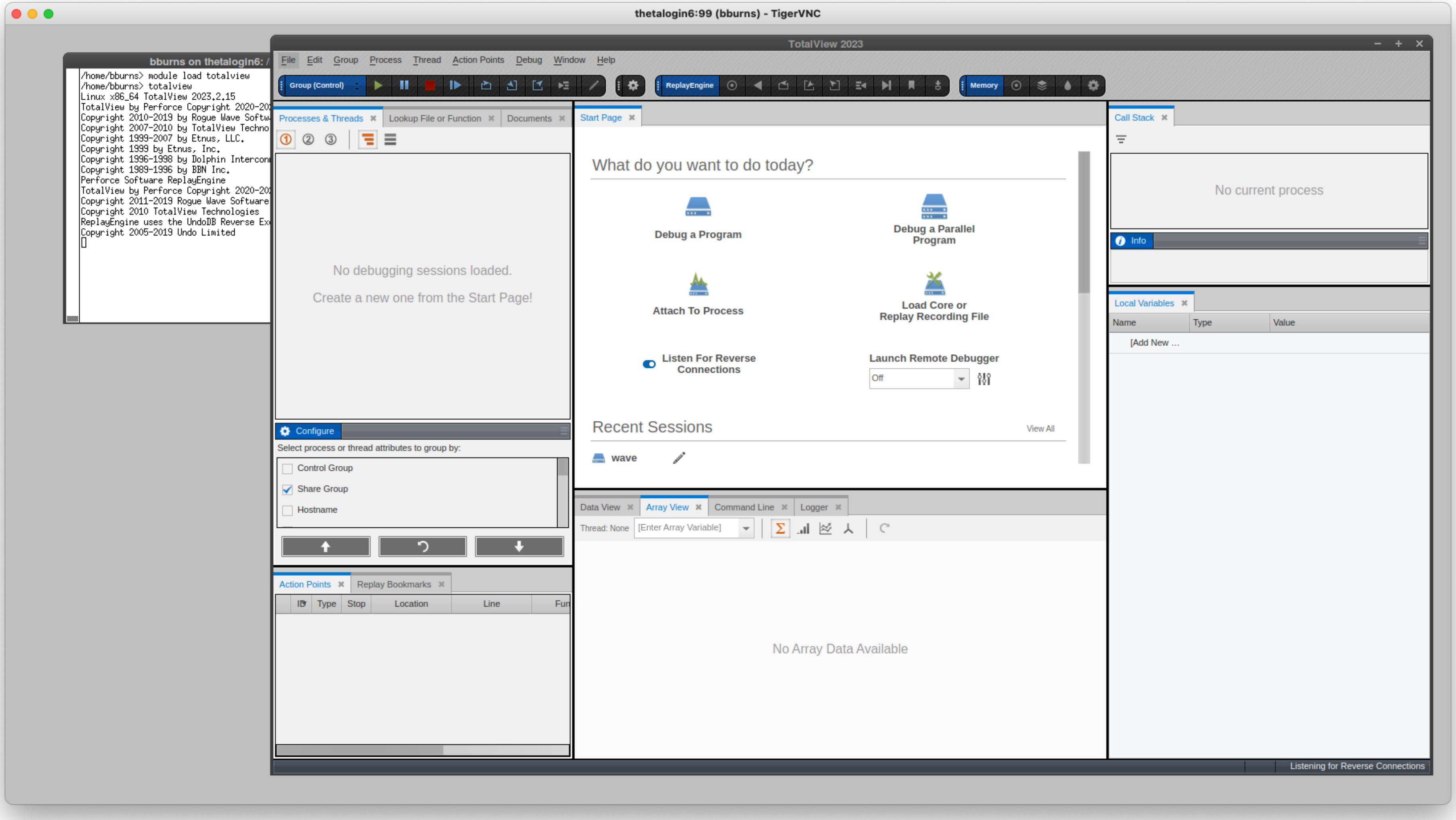
Configuring remote desktop window manager

- Window manager and VNC desktop start is controlled through ~/.vnc/xstartup
- Might need to configure
- Only icewm is installed on Theta in /usr/bin
- Example xstartup

```
#!/bin/sh

[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
if [ -x /usr/bin/icewm ]; then
  /usr/bin/icewm &
else
  echo "No window manager found. You should install a window
manager to get properly working VNC session."
fi
```

TotalView VNC Session / Theta



thetalogin6:99 (bburns) - TigerVNC

bburns on thetalogin6: /

```
/home/bburns> module load totalview
/home/bburns> totalview
Linux x86_64 TotalView 2023.2.15
TotalView by Perforce Copyright 2020-20
Copyright 2010-2019 by Rogue Wave Softw
Copyright 2007-2010 by TotalView Techno
Copyright 1999-2007 by Etnus, LLC.
Copyright 1999 by Etnus, Inc.
Copyright 1996-1998 by Dolphin Intercon
Copyright 1989-1996 by BBN Inc.
Perforce Software ReplayEngine
TotalView by Perforce Copyright 2020-20
Copyright 2011-2019 Rogue Wave Software
Copyright 2010 TotalView Technologies
ReplayEngine uses the UndoDB Reverse Ex
Copyright 2005-2019 Undo Limited
▯
```

TotalView 2023

No debugging sessions loaded.

Create a new one from the Start Page!

Select process or thread attributes to group by:

□ Control Group

☑ Share Group

□ Hostname

What do you want to do today?

Launch Remote Debugger: Off

Recent Sessions

wave

No current process

Thread: None [Enter Array Variable]

No Array Data Available

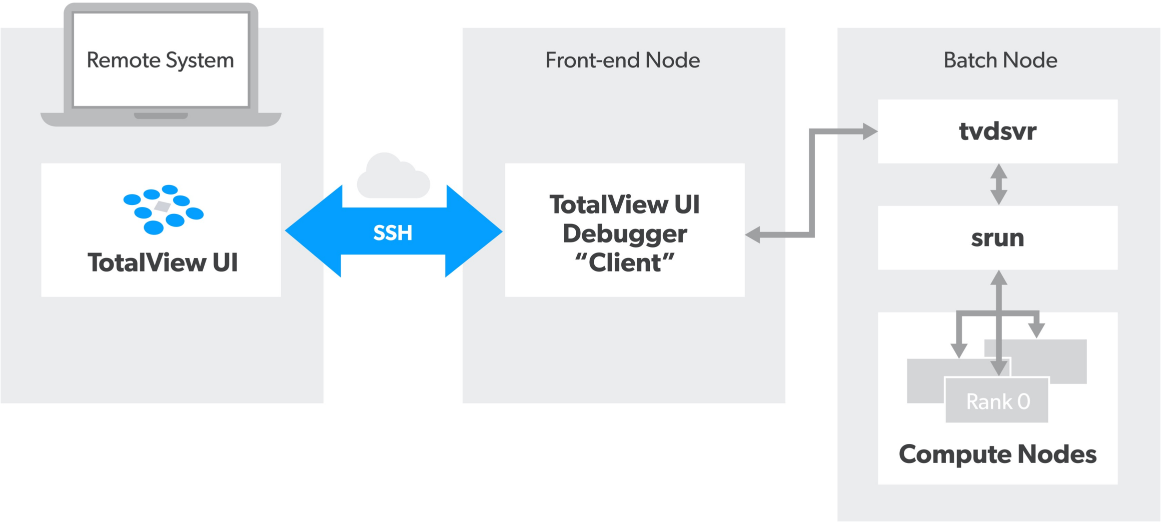
Listening for Reverse Connections

Using TotalView Remote UI on a Laptop

Debugging with TotalView From a Laptop to Theta and ThetaGPU

Install TotalView on your laptop

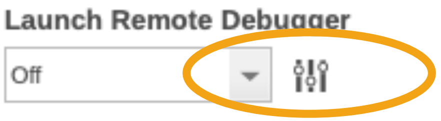
- Download and install TotalView on your Linux x86 or Mac x86 laptop
  - Copy from /soft/debuggers/totalview-2023-02-15
    - totalview-2023.2.15.linux_x86-64-installer.run
    - totalview-2023.2.15-darwin-x86-installer.dmg
  - Download from https://totalview.io/success/downloads
- Run installer
  - Do not select the *Install new license* option on *Setup License Key* step
  - TotalView will start ok without a license



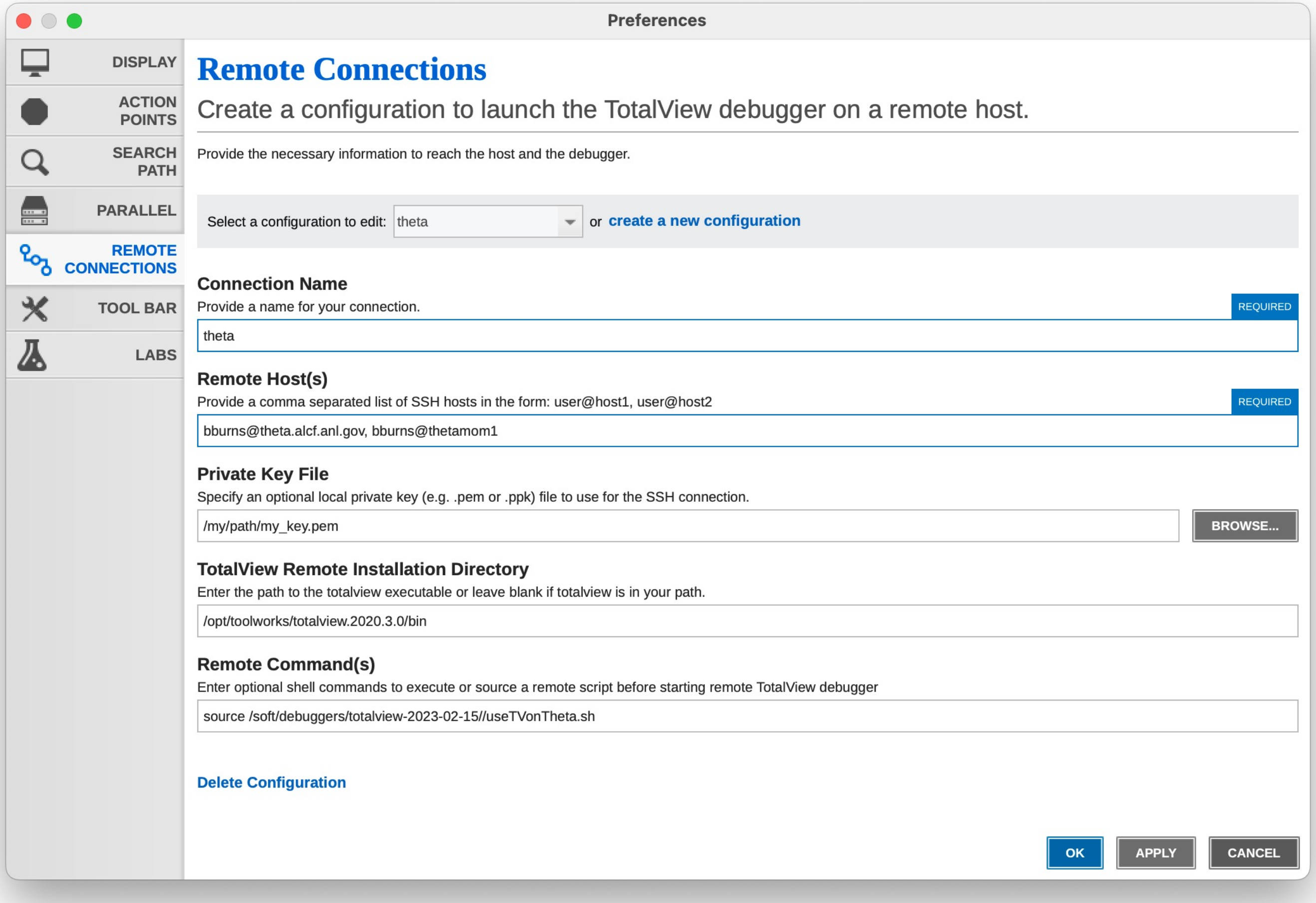
Configure TotalView Remote UI Theta Connection

Create remote host configuration

- Start TotalView from the **terminal**
  - Mac: `/Applications/Perforce/totalview.2023.2.15/TotalView.app/Contents/bin/totalview`
- Click *Create remote host configuration* button from Launch Remote Debugger control on Start Page



- Provide the following information:
  - Connection Name : **theta**
  - Remote Host(s) : **<username>@theta.alcf.anl.gov, thetamom1**
  - Remote Command(s) : **source /soft/debuggers/totalview-2023-02-15/useTVonTheta.sh**
- Click OK
- Select **theta** from the Launch Remote Debugger drop down
- Provide authentication through prompt on terminal



Remote Connections

Create a configuration to launch the TotalView debugger on a remote host.

Provide the necessary information to reach the host and the debugger.

Select a configuration to edit: theta or create a new configuration

Connection Name
Provide a name for your connection.
theta

Remote Host(s)
Provide a comma separated list of SSH hosts in the form: user@host1, user@host2
bburns@theta.alcf.anl.gov, bburns@thetamom1

Private Key File
Specify an optional local private key (e.g. .pem or .ppk) file to use for the SSH connection.
/my/path/my_key.pem

TotalView Remote Installation Directory
Enter the path to the totalview executable or leave blank if totalview is in your path.
/opt/toolworks/totalview.2020.3.0/bin

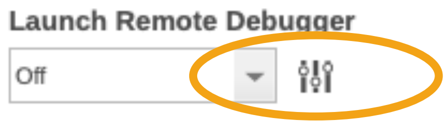
Remote Command(s)
Enter optional shell commands to execute or source a remote script before starting remote TotalView debugger
source /soft/debuggers/totalview-2023-02-15//useTVonTheta.sh

Delete Configuration

Configure TotalView Remote UI ThetaGPU Connection

Create remote host configuration

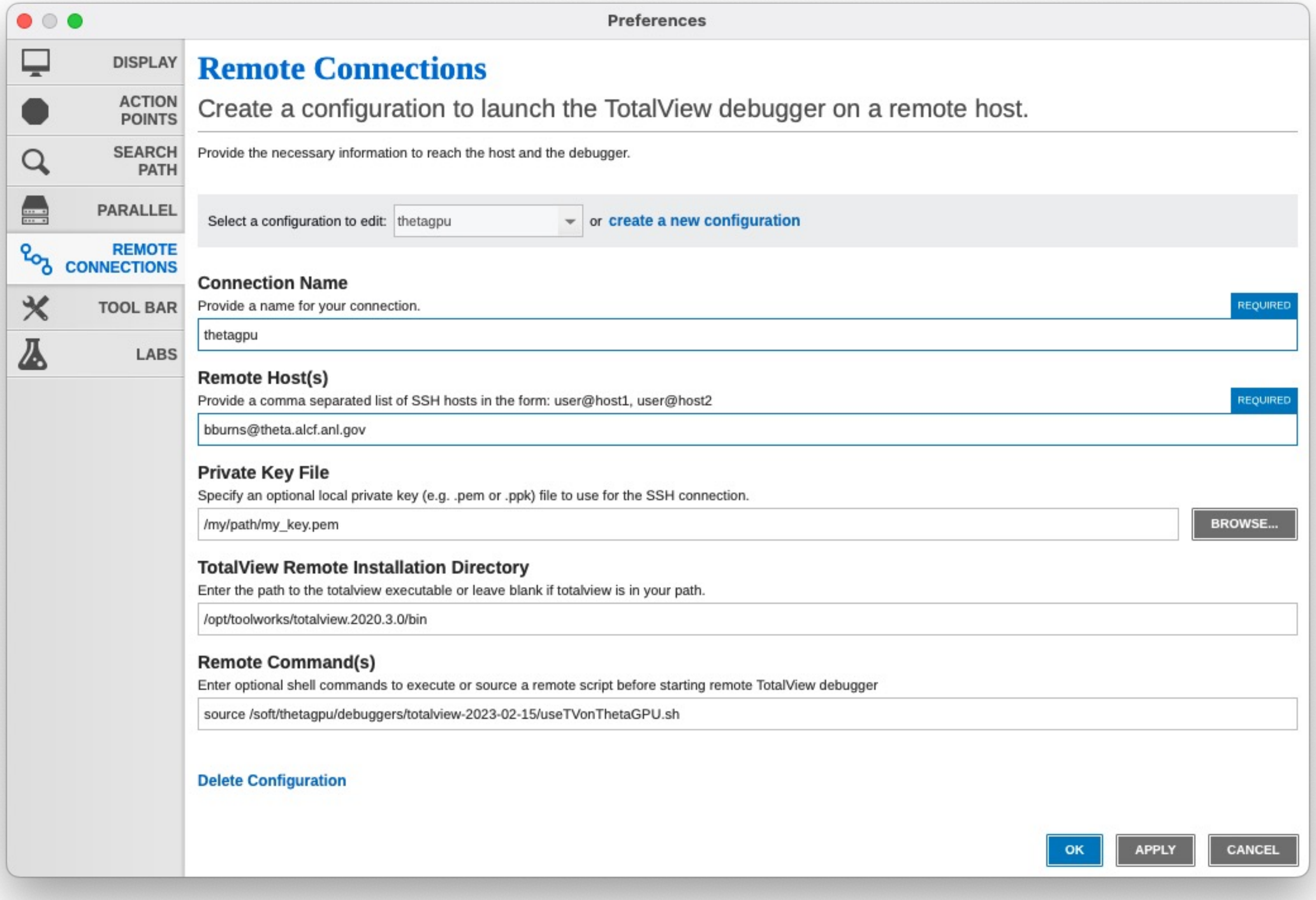
- Start TotalView from the **terminal**
  - Mac: `/Applications/Perforce/totalview.2023.2.15/TotalView.app/Contents/bin/totalview`
- Click *Create remote host configuration* button from Launch Remote Debugger control on Start Page



Launch Remote Debugger

Off

- Provide the following information:
  - Connection Name : **thetagpu**
  - Remote Host(s) : **<username>@theta.alcf.anl.gov**
  - Remote Command(s) : **source /soft/thetagpu/debuggers/totalview-2023-02-15/useTVonThetaGPU.sh**
- Click OK
- Select **thetagpu** from the Launch Remote Debugger drop down
- Provide authentication through prompt on terminal



Remote Connections

Create a configuration to launch the TotalView debugger on a remote host.

Provide the necessary information to reach the host and the debugger.

Select a configuration to edit: thetagpu or create a new configuration

Connection Name
Provide a name for your connection.
thetagpu

Remote Host(s)
Provide a comma separated list of SSH hosts in the form: user@host1, user@host2
bburns@theta.alcf.anl.gov

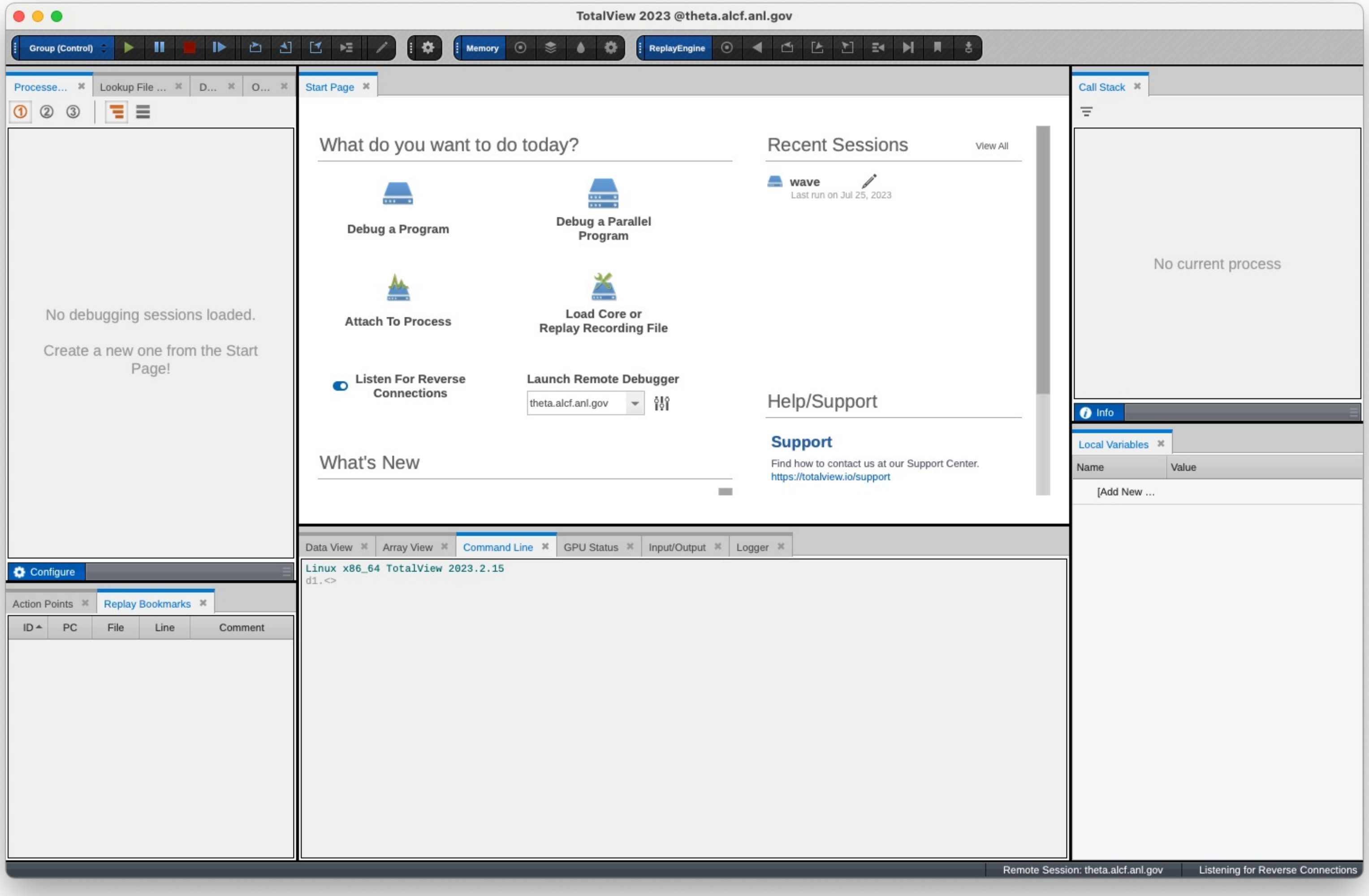
Private Key File
Specify an optional local private key (e.g. .pem or .ppk) file to use for the SSH connection.
/my/path/my_key.pem

TotalView Remote Installation Directory
Enter the path to the totalview executable or leave blank if totalview is in your path.
/opt/toolworks/totalview.2020.3.0/bin

Remote Command(s)
Enter optional shell commands to execute or source a remote script before starting remote TotalView debugger
source /soft/thetagpu/debuggers/totalview-2023-02-15/useTVonThetaGPU.sh

Delete Configuration

TotalView Remote UI



No debugging sessions loaded.

Create a new one from the Start Page!

What do you want to do today?

Recent Sessions

Help/Support

Support

Find how to contact us at our Support Center.
https://totalview.io/support

What's New

No current process

```
Linux x86_64 TotalView 2023.2.15
d1.<>
```

Easy Interactive Debugging with TotalView Reverse Connections

TotalView Reverse Connections for Cluster Environments

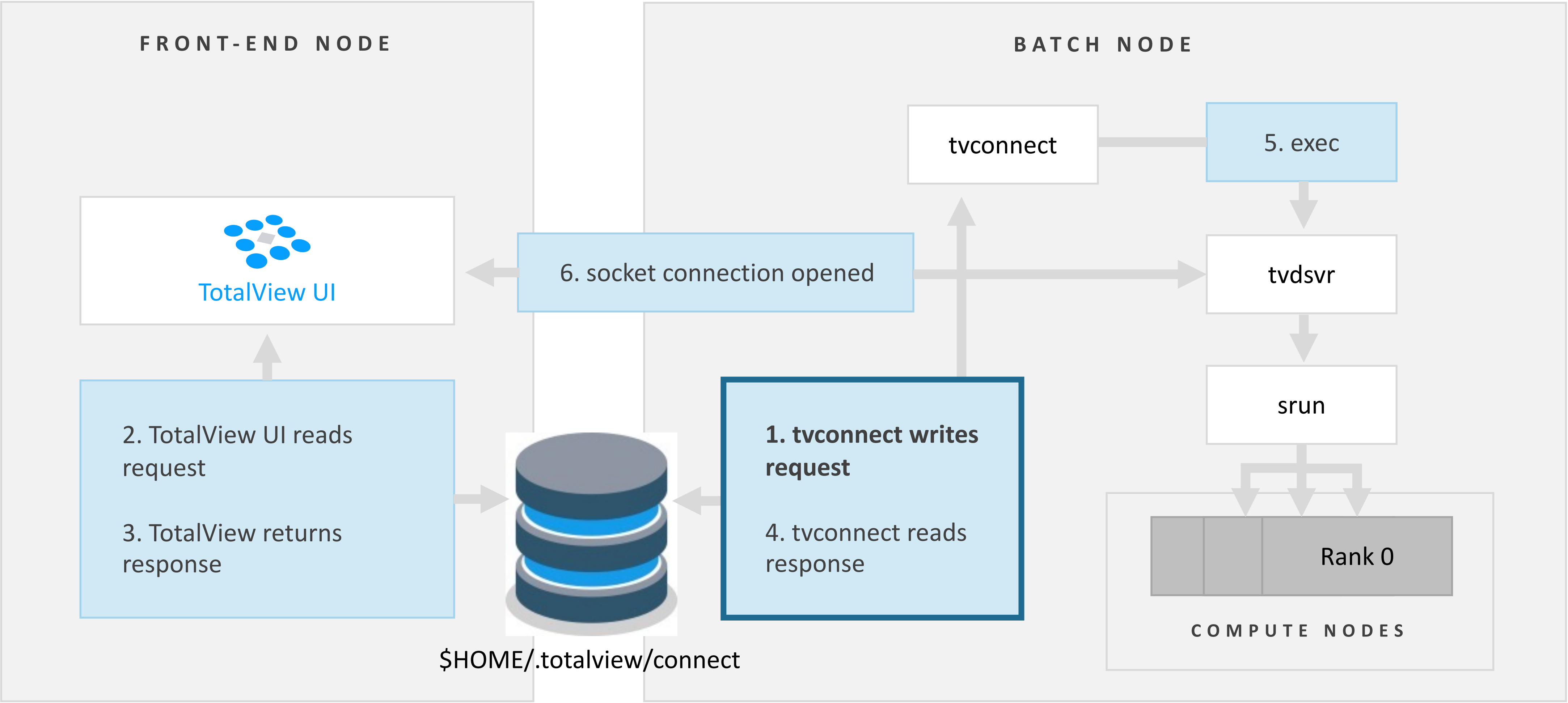
The Problem:

- Establishing an interactive debugging session in a cluster environment can be difficult
  - Timing issues when submitting through a job manager and when the job runs
  - The organization of modern HPC systems often makes it difficult to deploy tools such as TotalView
  - The compute nodes in a cluster may not have access to any X libraries or X forwarding
  - Launching a GUI on a compute node may not be possible

The Solution:

- Disconnect starting debugger UI from the backend job launch and debug session acquisition
- TotalView Reverse Connect workflow enables developers to start the TotalView UI on a front-end node and, when a job is run in the cluster, a remote TotalView reverse connect agent connects it back to the waiting UI
- Simply start TotalView on login node and then prefix parallel or application launch with `tvconnect`

Reverse Connection Flow Using TotalView's tvconnect



FRONT-END NODE

BATCH NODE

TotalView UI

2. TotalView UI reads request

3. TotalView returns response

$HOME/.totalview/connect

1. tvconnect writes request

4. tvconnect reads response

6. socket connection opened

tvconnect

5. exec

tvdsvr

srun

Rank 0

COMPUTE NODES

Debugging with TotalView on Theta using Reverse Connections

1. Log into theta.alcf.anl.gov

2. Request an allocation for interactive debugging

   Example: `qsub -A <allocation> -n 2 -q debug-flat-quad -I -t 60 --attrs=filesystems=home:eagle`

   Setup for TotalView use on thetamom# node

   ```
   source /soft/debuggers/totalview-2023-02-15/useTVonTheta.sh
   ```

3. Open new term, ssh **-X** to any thetamom# Node, setup and start TotalView

   Open new xterm

   ```
   ssh -X thetamom1
   source /soft/debuggers/totalview-2023-02-15/useTVonTheta.sh
   totalview
   ```

4. On original qsub thetamom node run parallel job under tvconnect

   ```
   cd ~/Demos/theta/LLNLMPIExamples
   tvconnect aprun -n 4 ./mpi_array
   ```

Debugging with TotalView on ThetaGPU using Reverse Connections

1. Log into theta.alcf.anl.gov

2. Log into ThetaGPU Login Node

   `ssh thetagpusn1` (or `thetagpusn2`)

3. Request an allocation for interactive debugging

   Example: `qsub -A <allocation> -n 1 -q single-gpu -I -t 60 --attrs=filesystems=home:eagle`

   Setup for TotalView use on thetagpu## node

   ```
   source /soft/thetagpu/debuggers/totalview-2023-02-15/useTVonThetaGPU.sh
   ```

4. Open new thetalogin# xterm, setup and start TotalView

   Open new xterm on thetalogin#

   ```
   source /soft/thetagpu/debuggers/totalview-2023-02-15/useTVonThetaGPU.sh
   totalview
   ```

5. On original qsub thetagpu## node run CUDA parallel job under tvconnect

   ```
   cd ~/Demos/thetagpu/Perlmutter_Training_Jan2022/CUDA/Ex-4
   tvconnect mpirun -n 4 ./vec_add
   ```

Debugging with TotalView on Polaris using Reverse Connections

1. Log into polaris.alcf.anl.gov

   ```
   ssh -X polaris.alcf.anl.gov
   ```

2. Request an allocation for interactive debugging

   Example: `qsub -X -I -A ATPESC2023 -q debug -l select=1:system=polaris -l walltime=60:00 -l filesystems=home:eagle`

   Setup for TotalView use on polaris node

   ```
   source /soft/debuggers/totalview-2023-02-15/useTVonPolaris.sh
   ```

3. Run job launch under TotalView, must specify the "-nocuda" flag

   ```
   cd ~/Demos/polaris/LLNLMPIExamples
   totalview -nocuda aprun -n 4 ./mpi_array
   ```

Batch Script Submission on Theta with Reverse Connect

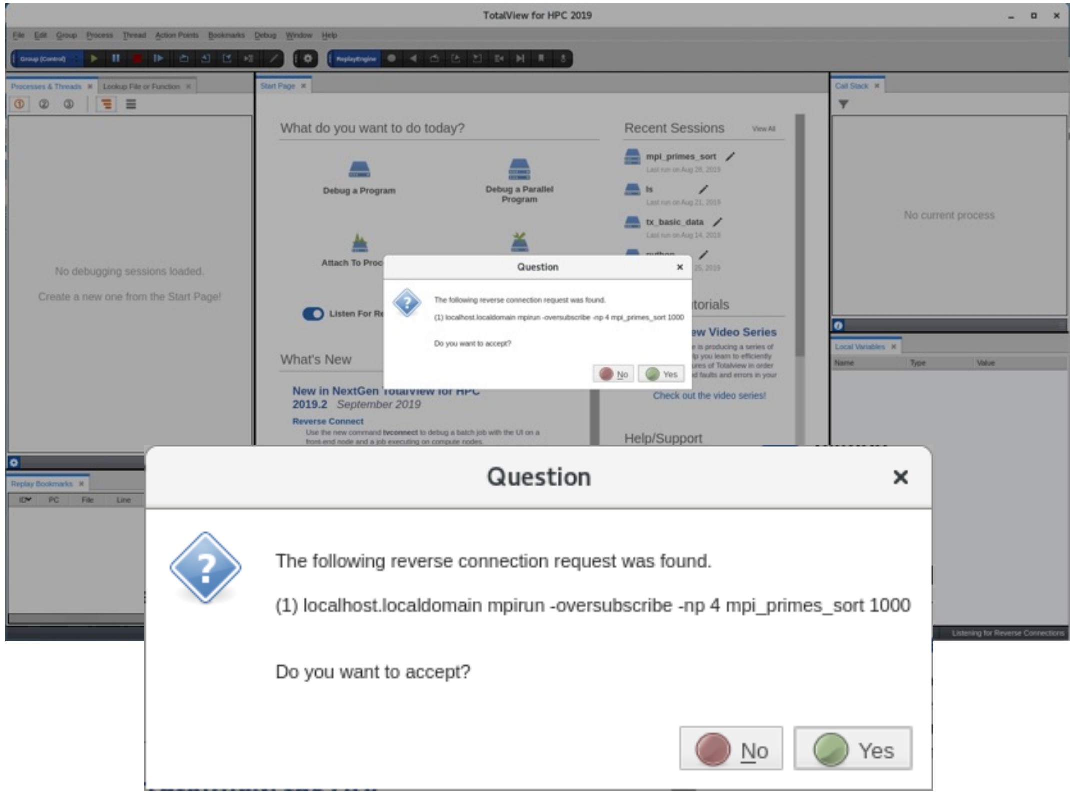
- Start a debugging session using TotalView Reverse Connect.
- Reverse Connect enables the debugger to be submitted to a cluster and connected to the GUI once run.
- Start TotalView UI on thetamom1 node and listen for reverse connections

  ```
  ssh -X thetamom1
  ```

- Very easy to utilize, simply prefix job launch or application start with “tvconnect” command.

```
#!/bin/bash
#COBALT -n 1
…
#COBALT –q debug-flat-quad
# Setup environment
source /soft/debuggers/totalview-2023-02-15/useTVonTheta.sh

tvconnect aprun -n ${NTOTRANKS} -N ${NRANKS_PER_NODE} \
   -d ${NDEPTH} -j ${NTHREADS_PER_CORE} -cc depth ./mpi_array
```



Question

The following reverse connection request was found.

(1) localhost.localdomain mpirun -oversubscribe -np 4 mpi_primes_sort 1000

Do you want to accept?

TotalView Hands on Labs

TotalView Hands on Labs

- Remotely connect to machine and enable Reverse Connection
- Copy /grand/ATPESC2023/EXAMPLES/track-6-toolsTotalView/ATRESC2023-TV-labs.tar.gz
- Programs are in labs/programs/
- Labs:
  - Lab 1 Debugger Basic
  - Lab 2 Viewing, Examining, Watching and Editing Data
  - Optional Lab 3 Examining and Controlling a Parallel Application
    - Using remote connect (tvconnect)
    - qsub -q training tvconnect.job
    - Modify and submit tvconnect.job on your machine

UI Navigation and Process Control

Demo

- TotalView UI demo (wave)

Control Program Execution with Action Points

Demo

- TotalView evaluation point demo (Combined)

Parallel Debugging Demo

Demo

- TotalView UI demo (mpi_array_broken or Qt Threads example)

Examining and Editing Data

Demo

- TotalView UI demo (wave)

Memory Debugging

Demo

- TotalView memory debugging demo

TotalView Reverse Debugging

Demo

- TotalView ReplayEngine Demo

TotalView Power Tip

- When debugging an MPI application, set a breakpoint after MPI_Init and then turn on reverse debugging.

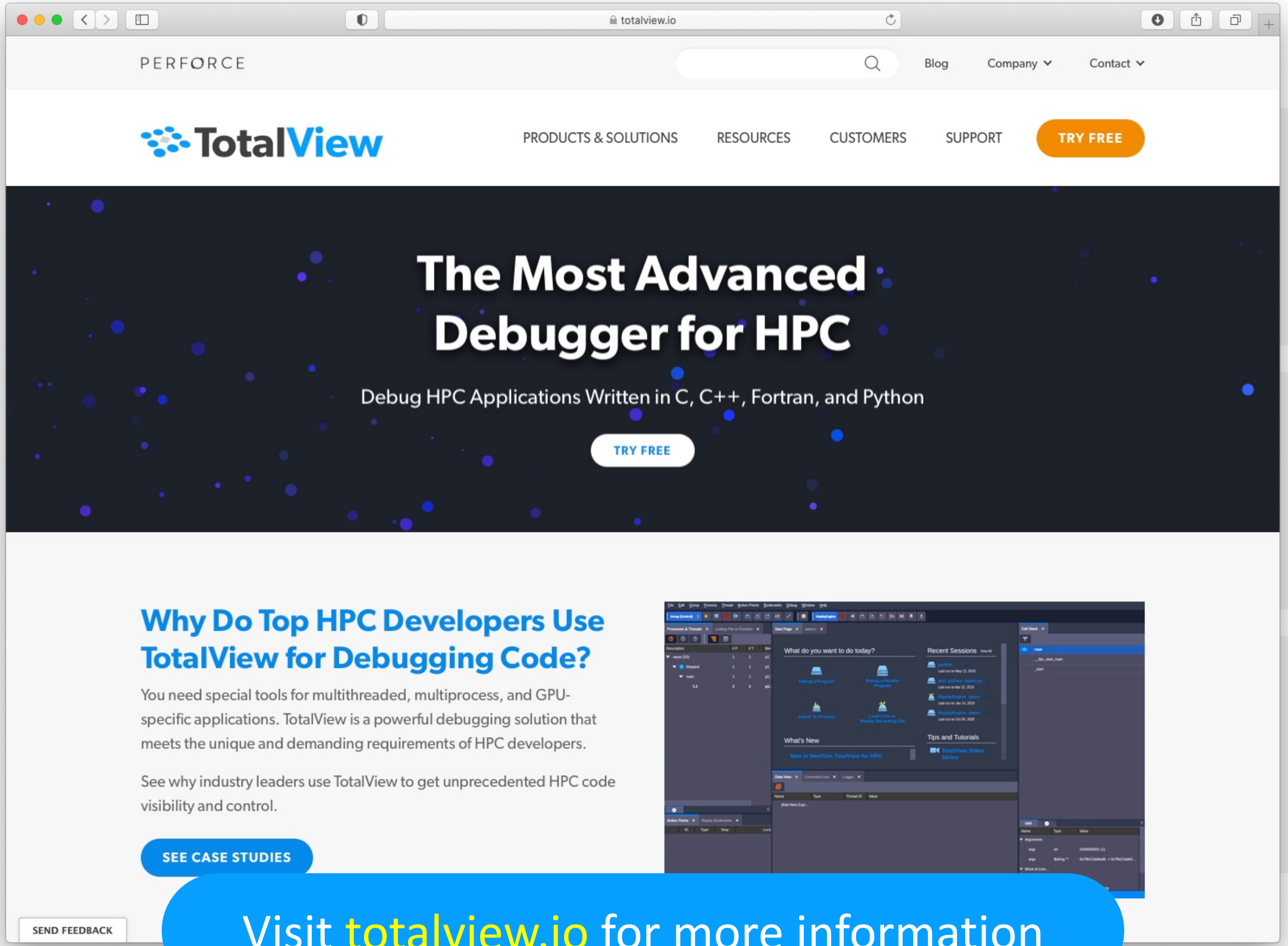
GPU Debugging with TotalView

NVIDIA GPU Debugging Demo

- Demo TotalView GPU debugging on ThetaGPU

Questions and Answers

Resources and Documentation



The Most Advanced Debugger for HPC

Debug HPC Applications Written in C, C++, Fortran, and Python

Why Do Top HPC Developers Use TotalView for Debugging Code?

You need special tools for multithreaded, multiprocess, and GPU-specific applications. TotalView is a powerful debugging solution that meets the unique and demanding requirements of HPC developers.

See why industry leaders use TotalView to get unprecedented HPC code visibility and control.

Visit totalview.io for more information

DOCUMENTATION

help.totalview.io

VIDEO TUTORIALS

totalview.io/support/video-tutorials

BLOG

totalview.io/blog

TotalView Debugging Feature References

Getting Started with TotalView

- https://totalview.io/video-tutorials/getting-started-totalview

How to Use Remote User Interface Debugging

- https://totalview.io/video-tutorials/how-use-remote-user-interface-debugging

Controlling Execution with Evaluation Points

- https://totalview.io/video-tutorials/controlling-execution-evaluation-points

Reverse Debugging

- https://totalview.io/video-tutorials/reverse-debugging

Debugging Python and C++ Mixed Language Applications

- https://totalview.io/video-tutorials/debugging-python-and-c-mixed-language-applications

Debugging the Toughest Challenges with NVIDIA and AMD GPUs

- https://totalview.io/resources/debugging-toughest-challenges-nvidia-and-amd-gpus

PERFORCE

Thank You

Contact Info

Bill Burns

bburns@perforce.com