

# HP-SEE TotalView Debugger

[www.hp-see.eu](http://www.hp-see.eu)

Josip Jakić  
Scientific Computing Laboratory  
Institute of Physics Belgrade  
[josipjagic@ipb.ac.rs](mailto:josipjagic@ipb.ac.rs)



# HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

# Agenda



**HP-SEE**

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Introduction
- ❑ Getting started with TotalView
- ❑ Primary windows
- ❑ Basic functions
- ❑ Further functions
- ❑ Debugging parallel programs
- ❑ Topics not covered
- ❑ References and more information

# Introduction



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ TotalView is a sophisticated software debugger product from Rogue Wave Software, Inc.
- ❑ Used for debugging and analyzing both serial and parallel programs
- ❑ Designed for use with complex, multi-process and/or multi-threaded applications
- ❑ Supported on most HPC platforms
- ❑ Provides both a GUI and command line interface
- ❑ Includes memory debugging features
- ❑ Supports the usual HPC application languages:
  - ❑ C/C++
  - ❑ Fortran77/90
  - ❑ Assembler

# Getting started with TotalView (1/2)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ **-g** flag enables generation of symbolic debug information for most compilers
- ❑ Programs compiled without the **-g** option are allowed to be debugged, however, only the assembler code can be viewed
- ❑ Programs should be compiled without optimization flags
- ❑ Parallel programs may require additional compiler flags
- ❑ A variety of ways to start the program
  - ❑ totalview (invokes New Program dialog box)
  - ❑ totalview filename
  - ❑ totalview filename corefile
  - ❑ totalview filename -a args
  - ❑ totalview filename -remote hostname [:portnumber]

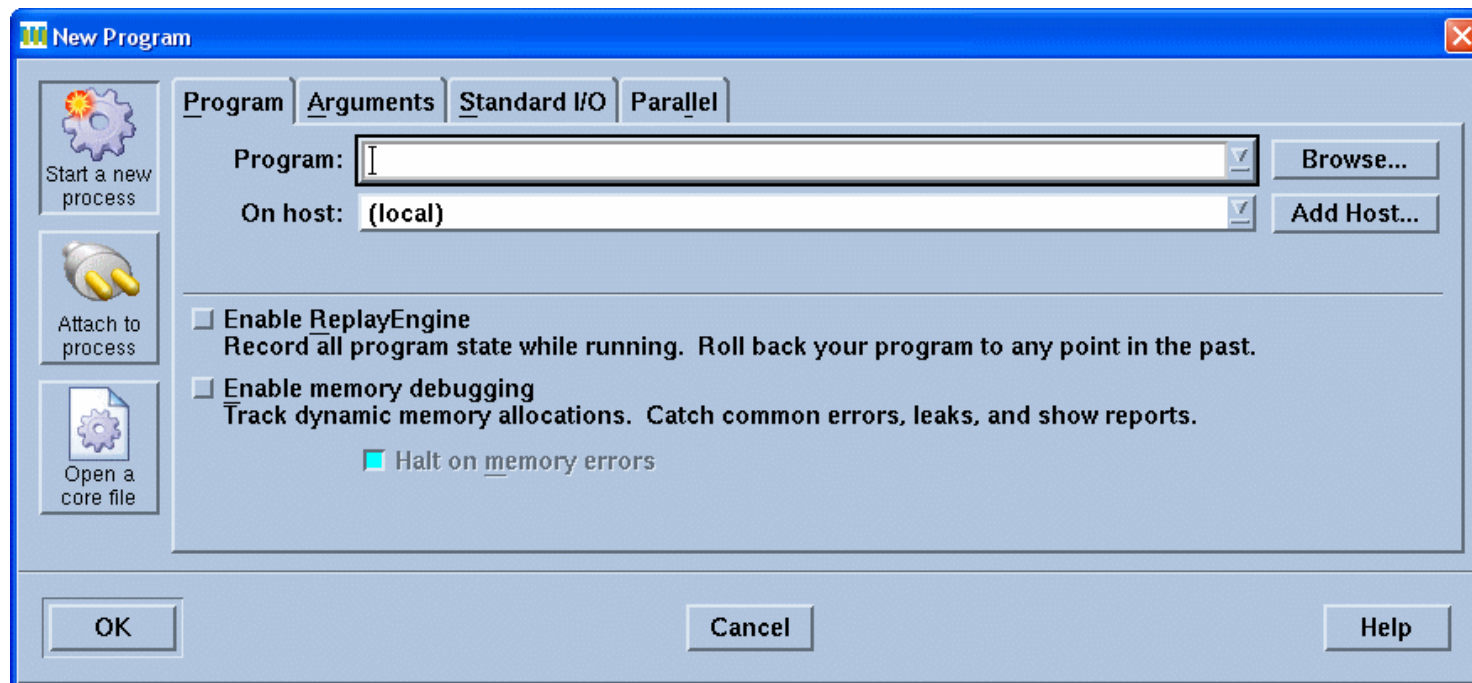
# Getting started with TotalView (2/2)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- New Program dialog box



- Numerous options for various means of selecting a program

# Primary Windows (1/7)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## □ Root Window

ID	Rank	Host	Status	Description
1		<local>	B	srunk (5 active threads)
2	0	hera15.llnl.gov	B	srunk<scatter>.0 (1 active threads)
3	1	hera15.llnl.gov	T	srunk<scatter>.1 (1 active threads)
4	2	hera15.llnl.gov	T	srunk<scatter>.2 (1 active threads)
5	3	hera15.llnl.gov	T	srunk<scatter>.3 (1 active threads)
5.1	3	hera15.llnl.gov	T	in_dl_debug_state
6	4	hera15.llnl.gov	T	srunk<scatter>.4 (1 active threads)
7	5	hera15.llnl.gov	T	srunk<scatter>.5 (1 active threads)
8	6	hera15.llnl.gov	T	srunk<scatter>.6 (1 active threads)

- Appears when the TotalView GUI is started
- Overview of all processes and threads, showing assigned ID, MPI rank, host, status and brief description/name for each

# Primary Windows (2/7)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- Root Window
  - Process and Thread State Codes

State Code	Description
B	Stopped at a breakpoint
E	Stopped because of an error
H	In a Hold state
K	Thread is executing within the kernel
M	Mixed - some threads in a process are running and some not
R	Running
T	Thread is stopped
W	At a watchpoint

# Primary Windows (3/7)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## □ Process Window

srun-scatter>.0

File Edit View Group Process Thread Action Point Debug Tools Window Help

Group (Control) Go Halt Kill Restart Next Step Out Run To Prev UnStep Caller BackT

Rank 0: srun-scatter>.0 (At Breakpoint 2)  
Thread 1 (46912531031056): scatter (At Breakpoint 2)

**Stack Trace**

Function	FP
C main	FP=7fffffffde50
_libc_start_main	FP=7fffffffdd10
_start	FP=7fffffffdd20

**Stack Frame**

Function "main":  
argc: 0x00000001 (1)  
argv: 0x7fffffffdd38 -> 0x7fffff

Local variables:  
numtasks: 0x00000004 (4)  
rank: 0x00000000 (0)  
sendcount: 0x00000004 (4)  
recvcnt: 0x00000004 (4)  
source: 0x00000001 (1)  
sendbuf: (float[4][4])  
recvbuf: (float[4])

**Function main in mpi\_scatter.c**

```
20 {13.0, 14.0, 15.0, 16.0} };  
21 float recvbuf[SIZE];  
22  
23 MPI_Init(&argc, &argv);  
24 MPI_Comm_rank(MPI_COMM_WORLD, &rank);  
25 MPI_Comm_size(MPI_COMM_WORLD, &numtasks);  
26  
27 if (numtasks == SIZE) {  
28     source = 1;  
29     sendcount = SIZE;  
30     recvcnt = SIZE;  
31     MPI_Scatter(sendbuf, sendcount, MPI_FLOAT, recvbuf, recvcnt,  
32               MPI_FLOAT, source, MPI_COMM_WORLD);  
33  
34     printf("rank= %d Results: %f %f %f %f\n", rank, recvbuf[0],  
35          recvbuf[1], recvbuf[2], recvbuf[3]);  
36 }  
37 else  
38     printf("Must specify %d processors. Terminating.\n", SIZE);  
39
```

**Action Points** Processes Threads P- P+ T- T+

2 mpi\_scatter.c#34 main+0xc7...



# Primary Windows (4/7)



**HP-SEE**

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## ❑ Process Window

- ❑ For multi-process/multi-threaded programs, every process and every thread may have its own Process Window if desired
- ❑ Comprised of:
  - ❑ Pull-down menus
  - ❑ Execution control buttons
  - ❑ Navigation control buttons
  - ❑ Process and thread status bars
  - ❑ 4 "Panels"

# Primary Windows (5/7)



**HP-SEE**

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Process Window
  - ❑ Stack Trace Pane
    - ❑ Shows the call stack of routines the current executable is running
  - ❑ Stack Frame Pane
    - ❑ Displays the local variables, registers and function parameters for the selected executable.
  - ❑ Source Pane
    - ❑ Displays source for the currently selected program or function with program counter, line numbers and any associated action points

# Primary Windows (6/7)



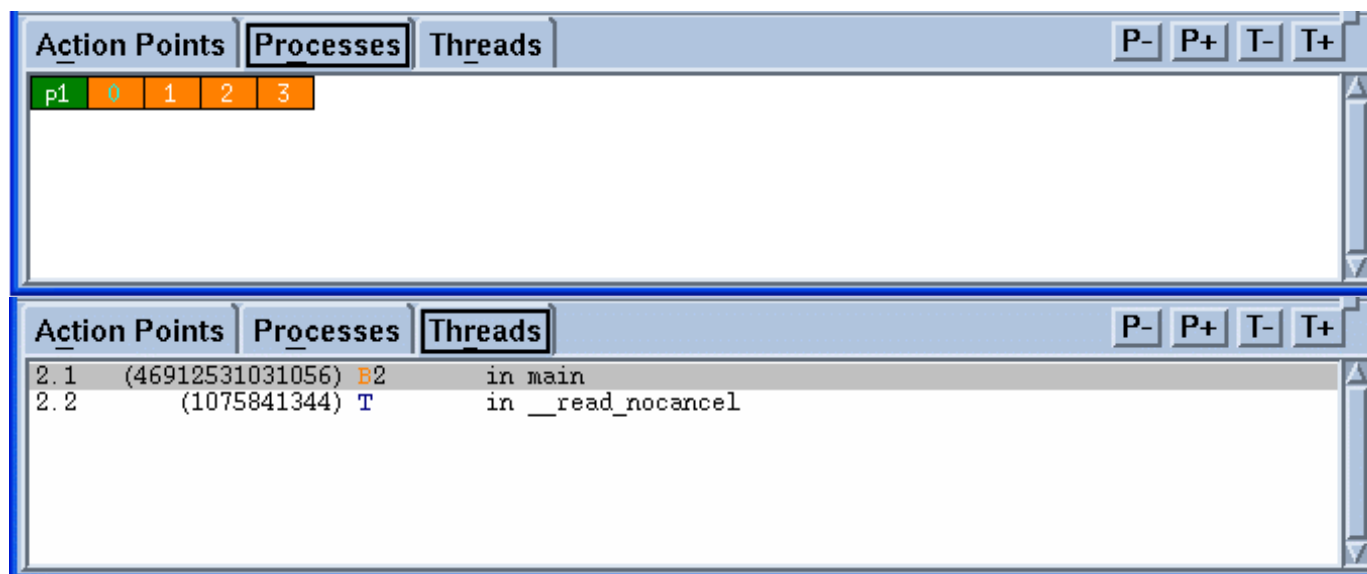
HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## □ Process Window

### □ Action Points, Processes, Threads Pane

- A multi-function pane. By default, it shows any action points that have been set
- May also select Processes to show attached processes or Threads to show associated threads



# Primary Windows (7/7)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## Variable Window

Field	Type	Value
a	double *	0x00602010 -> 1.0000000000000000e+00
b	double *	0x00602ca0 -> 1.0000000000000000e+00
sum	double	4.0000000000000000e+02
veclen	int	0x00000064 (100)

- Displays detailed information about selected program variables
- Permits editing, diving, filtering and sorting of variable data

# Basic Functions (1/4)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Viewing Source Code
  - ❑ Source, Assembler or Both
  - ❑ To toggle between the different display modes:
    - ❑ Process Window > View Menu > Source As
- ❑ Displaying Function / File Source Code
  - ❑ Finding and displaying the source code:
    - ❑ Process Window > View Menu > Lookup Function
- ❑ Setting a Breakpoint
  - ❑ Most basic of TotalView's action points used to control a program's execution
  - ❑ Halts execution at a desired line before executing the line
  - ❑ "Boxed" lines are eligible for breakpoints

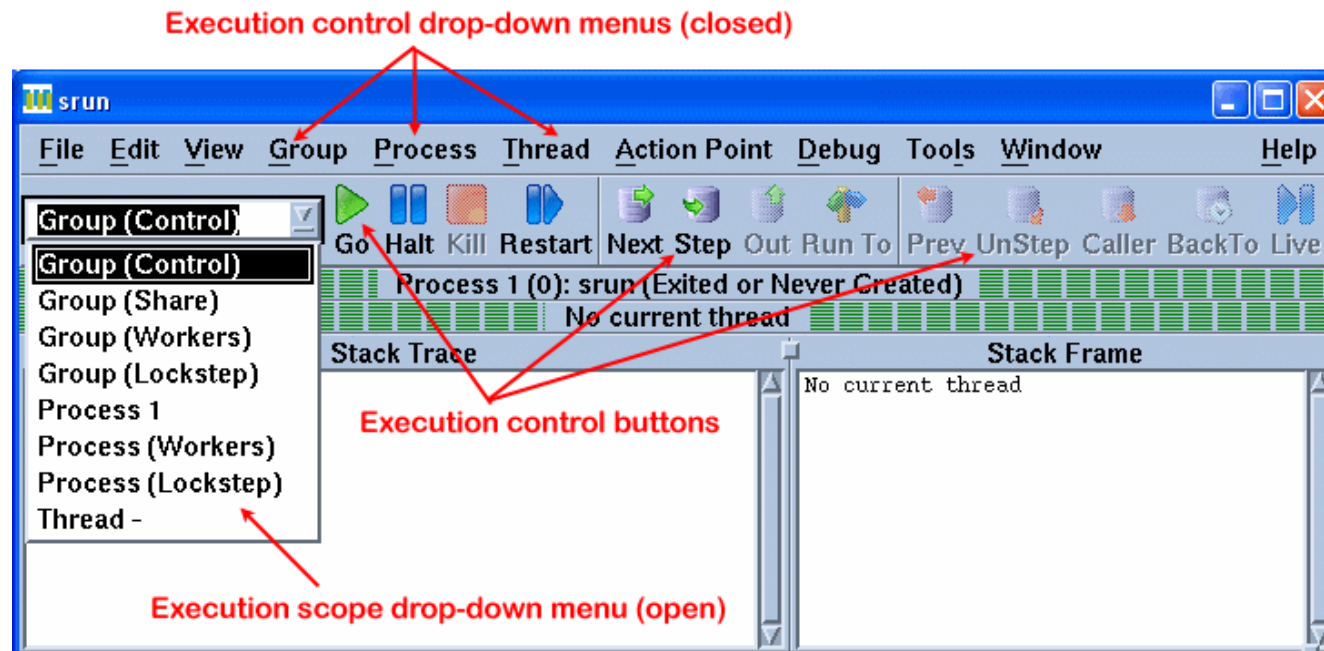
# Basic Functions (2/4)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- Controlling the execution of a program within TotalView involves two decisions:
  - Selecting the appropriate command
  - Deciding upon the scope of the chosen command



# Basic Functions (3/4)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Group, Process, Thread Command Scopes
  - ❑ For serial programs, execution scope is not an issue because there is only one execution stream
  - ❑ For parallel programs, execution scope is critical - you need to know which processes and/or threads your execution command will effect
- ❑ Viewing and Modifying Data
  - ❑ TotalView allows you to view variables, registers, areas of memory and machine instructions
  - ❑ Leaving a Variable Window open allows you to perform runtime monitoring of variables (updated each time program is stopped)
  - ❑ You can edit variables from within the Variable Window
  - ❑ The modified variable has effect when the program resumes execution

# Basic Functions (4/4)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## □ Arrays

□ For array data, TotalView provides several additional features:

- Displaying array slices
- Data filtering
- Data Sorting
- Array statistics

## □ Array Viewer

- To view a multi-dimensional array in "spreadsheet" format:
  - Variable Window > Tools Menu > Array Viewer



# Further Functions(1/5)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Viewing a Core File
  - ❑ TotalView can be used to examine the core file from a crashed job and examining the state (variables, stack, etc.) of the program when it crashed
    - ❑ Check your shell's limit settings, use either the limit (csh/tcsh) or ulimit -a (sh/ksh/bash) command and override if necessary
- ❑ Code fragments
  - ❑ Code fragments can include a mixture of:C, Fortran or Assembler language
  - ❑ TotalView built-in variables (\$tid, \$pid, \$systid ... )
  - ❑ TotalView built-in statements (\$stop, \$hold, \$stopall ...)
  - ❑ Code fragments can be entered by two methods:
    - ❑ Evaluate Window
    - ❑ Evaluation Point

# Further Functions(2/5)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ TotalView supports four different types of action points:
  - ❑ Breakpoint
  - ❑ Process Barrier Point
  - ❑ Evaluation Point
    - ❑ causes a code fragment to execute when reached
  - ❑ Watchpoint
    - ❑ Monitors when the value stored in memory is modified and either stop execution or evaluates
- ❑ Managing action points
  - ❑ Deleting Action Points
    - ❑ **Delete All**
  - ❑ Disabling / Enabling Action Points
    - ❑ **Suppress All**
  - ❑ Saving / Loading Action Points

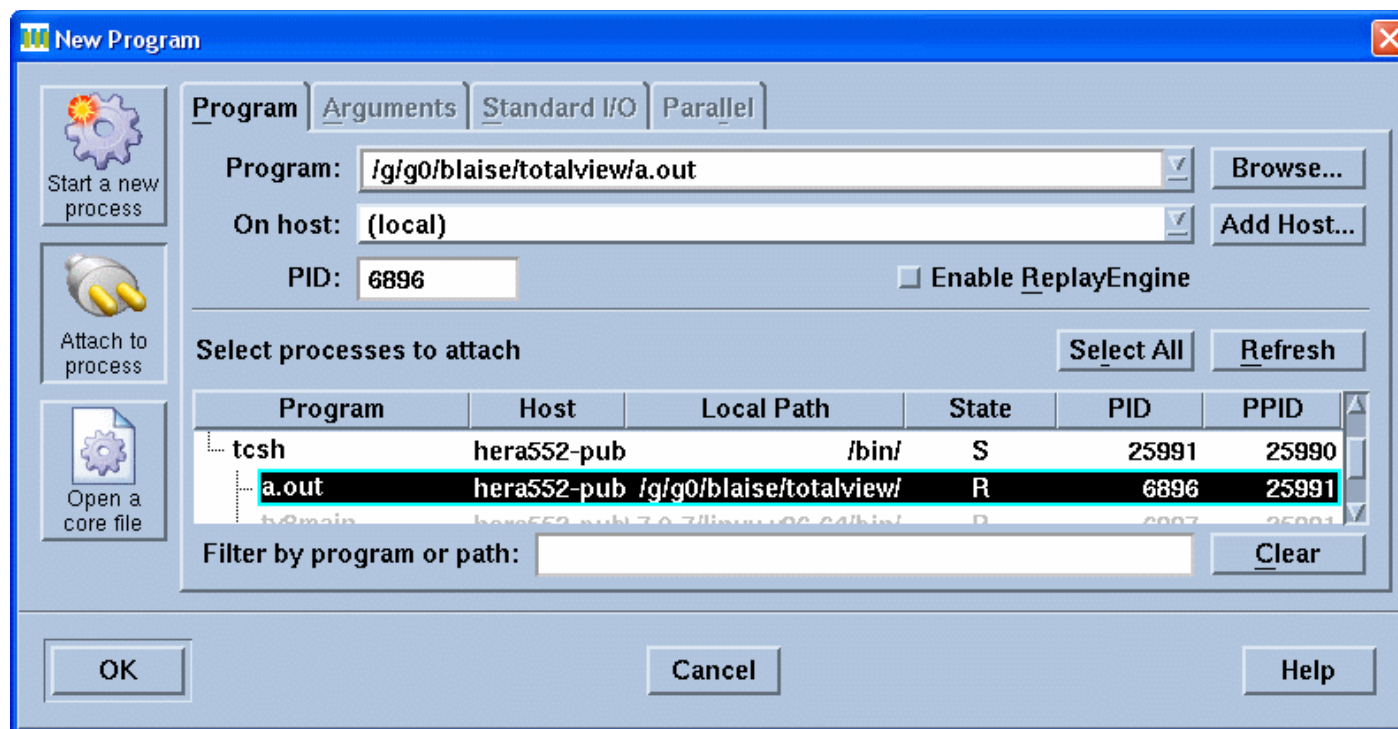
# Further Functions(3/5)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- Attaching / Detaching Processes
  - In the New Program Dialog Box, select the Attach to process button



# Further Functions(4/5)

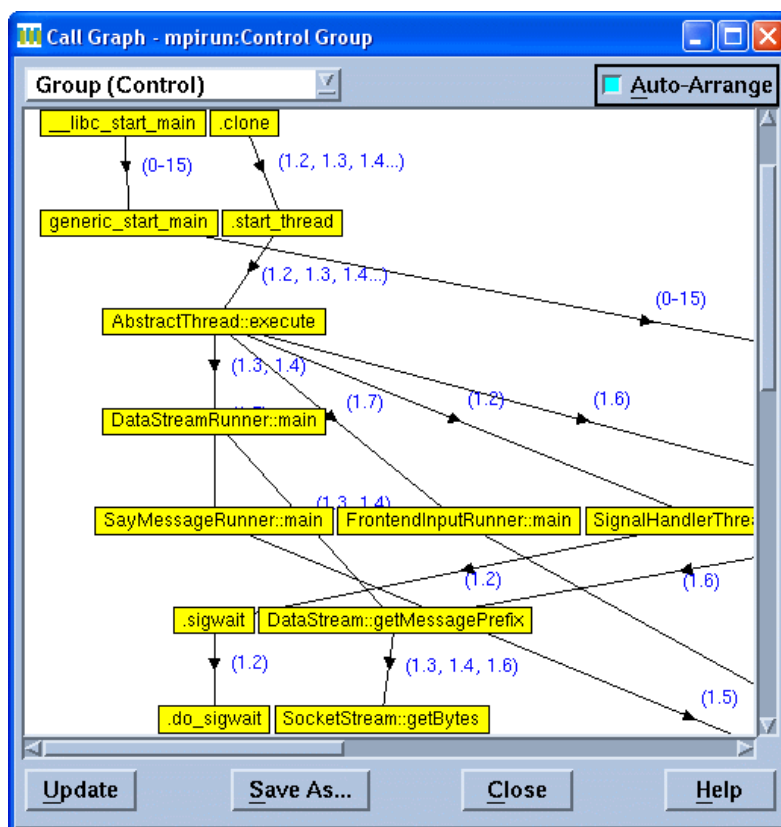


HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## ▣ Displaying Program's Call Graph

- ▣ Process Window > Tools Menu > Call Graph



# Further Functions(5/5)



**HP-SEE**

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Some other functions and settings
  - ❑ Setting Executable Command Arguments
  - ❑ Setting Source Code Search Paths
  - ❑ Setting stdin, stdout, and stderr
  - ❑ Setting Preferences
  - ❑ Signal Handling
  - ❑ Debugging Memory Problems
  - ❑ Visualizing Array Data
  - ❑ Command Line Interpreter (CLI)



- ❑ Demonstration on topics covered so far using simple serial code
  - ❑ Starting TotalView
  - ❑ Primary windows
  - ❑ Basic Functions
  - ❑ Setting Evaluation Points
  - ❑ Attach to a hung process
  - ❑ Debugg the hung process

# Debugging Parallel Programs (1/8)



**HP-SEE**

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

## ❑ Process/Thread Groups

### ❑ Types of P/T Groups:

#### ❑ Control Group:

- Contains all processes and threads created by the program across all processors

#### ❑ Share Group:

- Contains all of the processes and their threads, that are running the same executable

#### ❑ Workers Group:

- Contains all threads that are executing user code

#### ❑ Lockstep Group:

- Includes all threads in a Share Group that are at the same PC (program counter) address

# Debugging Parallel Programs (2/8)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Debugging Threaded Codes
  - ❑ Finding Thread Information
    - ❑ Root Window
    - ❑ Process Window
  - ❑ Selecting a Thread
    - ❑ Thread Navigation Buttons
  - ❑ Execution Control for Threaded Programs
    - ❑ Three Scopes of Influence
    - ❑ Synchronous vs. Asynchronous
    - ❑ Thread-specific Breakpoints



# Debugging Parallel Programs

## (3/8)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Viewing and Modifying Thread Data
  - ❑ Laminated Variables
    - ❑ In a parallel program, the same variable will usually have multiple instances across threads and/or processes
    - ❑ Laminating a variable means to display all occurrences simultaneously in a Variable Window
    - ❑ Laminated variables can include scalars, arrays, structures and pointers
    - ❑ Variable Window > View Menu > Show Across > Thread

# Debugging Parallel Programs

## (4/8)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Debugging OpenMP Codes
  - ❑ Thread-based
  - ❑ Setting the number of threads
    - ❑ Default: usually equal to the number of cpus on the machine
    - ❑ OMP\_NUM\_THREADS environment variable at run time
    - ❑ OMP\_SET\_NUM\_THREADS routine within the source code
  - ❑ Code transformation
  - ❑ Master thread vs. Worker threads
  - ❑ Execution Control
    - ❑ You can not step into or out of a PARALLEL region
    - ❑ Set a breakpoint within the parallel region and allow the process to run to it
  - ❑ Manager Threads

# Debugging Parallel Programs

## (5/8)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

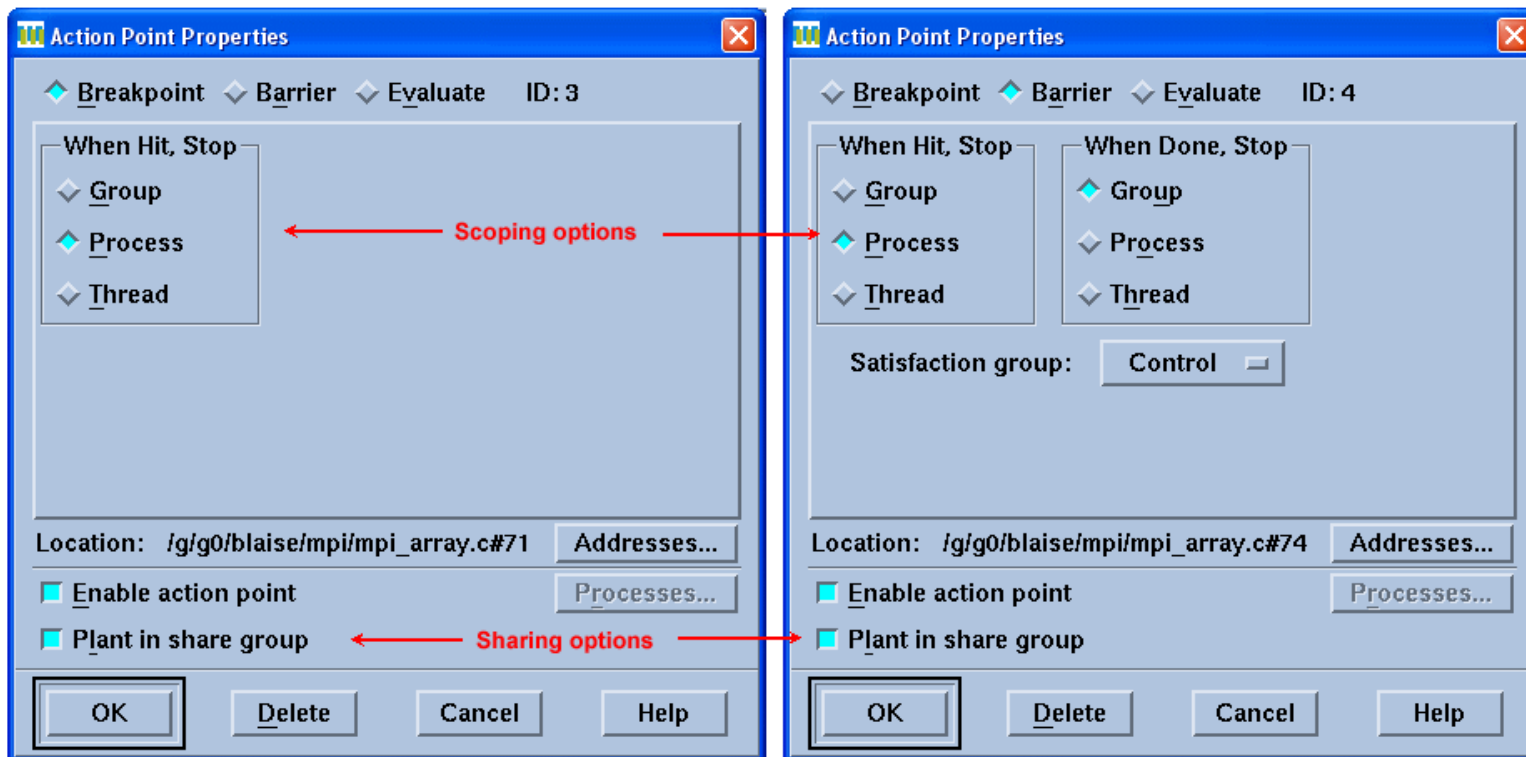
- ❑ Debugging MPI Codes
  - ❑ Multi-Process
  - ❑ MPI manager process
    - ❑ Typically, MPI programs run under a "manager" process, such as poe, srun, prun, mpirun, dmpirun, etc.
  - ❑ Automatic process acquisition
- ❑ MPI features similar to OpenMP
  - ❑ Selecting an MPI Process
    - ❑ Process Navigation Buttons
  - ❑ Controlling MPI Process Execution
    - ❑ MPI task execution can be controlled at the individual process level, or collectively as a "group"
- ❑ Starting and Stopping Processes
- ❑ Holding and Releasing Processes

# Debugging Parallel Programs (6/8)



HP-SEE  
High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- Breakpoints and Barrier Points
  - Individual breakpoint and barrier point behavior can be customized via the Action Point Properties Dialog Box



# Debugging Parallel Programs (7/8)



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Displaying Message Queue State
  - ❑ Process Window > Tools Menu > Message Queue
  - ❑ The Message Queue Window
  - ❑ Types of Messages Displayed:
    - ❑ Pending receives - non-blocking and blocking.
    - ❑ Pending sends - non-blocking and blocking.
    - ❑ Unexpected messages - messages sent to this process which do not yet have a matching receive operation.
- ❑ Message Queue Graph
  - ❑ Process Window > Tools Menu > Message Queue Graph

# Debugging Parallel Programs

## (8/8)



**HP-SEE**

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ Debugging Hybrid Codes
  - ❑ Hybrid codes are programs that use more than one type of parallelism
  - ❑ Combines technics used in threaded, OpenMP and MPI debugging
- ❑ Attaching to a Running Batch Job
  - ❑ If you have a batch job that is already running, you can start TotalView on one of the cluster's login nodes and then attach to it



- ❑ OpenMP example
  - ❑ Specify number of threads
  - ❑ Set breakpoint inside parallel region
  - ❑ Display a variable's value across all threads
- ❑ MPI example
  - ❑ Start TotalView using mpirun and executable
  - ❑ Set a barrier point
  - ❑ Display variables across processes

# Topics not covered



**HP-SEE**

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ CLI
- ❑ Setting up remote debugging sessions
- ❑ Memory debugging
- ❑ Replay engine
- ❑ and more...



# References and More Information



HP-SEE

High-Performance Computing Infrastructure  
for South East Europe's Research Communities

- ❑ The most useful documentation and reference material is from TotalView's vendor site:  
<http://www.roquewave.com/>
- ❑ Online tutorial:  
[/https://computing.llnl.gov/tutorials/totalview/](https://computing.llnl.gov/tutorials/totalview/)

## TotalView on PARADOX

- ❑ Login to ui.ipb.ac.rs via ssh using -X flag
  - ❑ `$ ssh <username>@ui.ipb.ac.rs -X`
- ❑ TotalView is located at the following path:
  - ❑ `/opt/toolworks/totalview.8.9.0-2/bin/totalview`