TOTALVIEW

(a graphical debugger)

www.roguewave.com

TUesday Rabbit Diversion
Apr. 15 2014; Apr. 22 2014

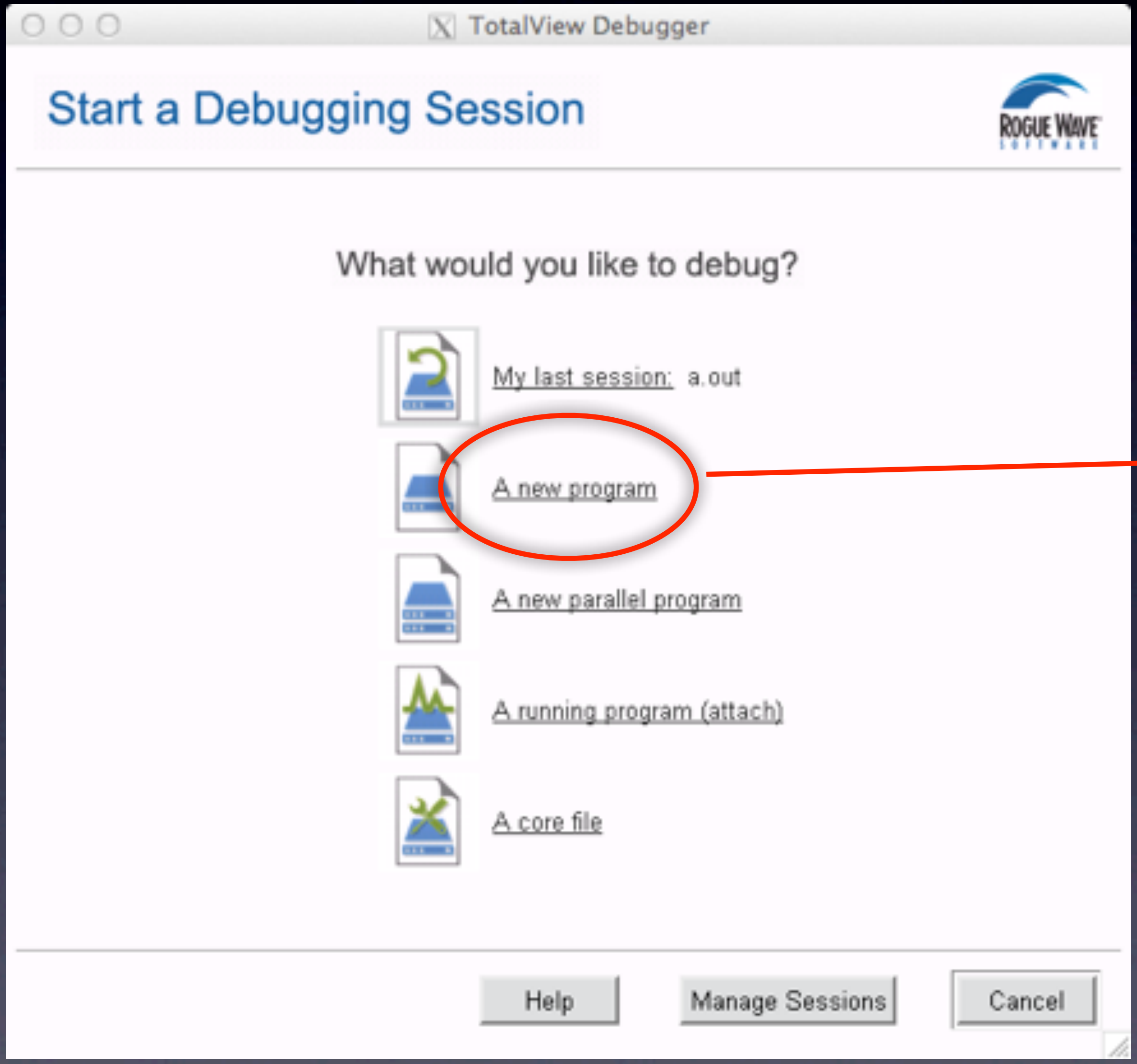
“...a GUI-based source code defect analysis tool that gives you unprecedented control over processes and thread execution and visibility into program state and variables.”

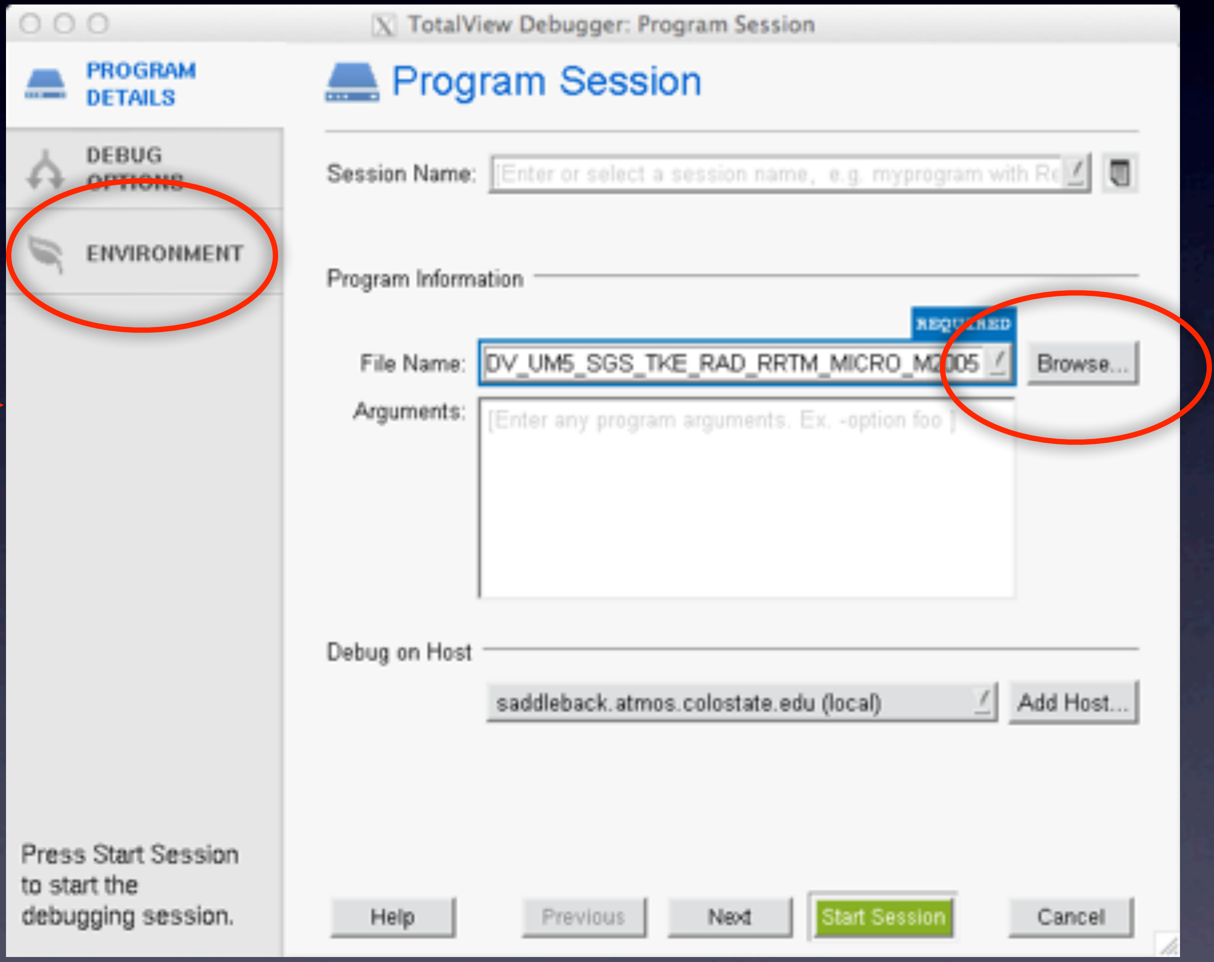
Launching Totalview - serial job

- Requires licensing - see Kelley for details
- path - /usr/local/toolworks/totalview/bin/
- make sure $DISPLAY is set to your machine
- compile your code with -c -g options (on saddleback include -O0)
- from terminal window : **tv8 <*executable*>** or **totalview <*executable*>**

Launching Totalview - serial job

- without specifying executable
- browse for your executable
- control stdin and stdout from ENVIRONMENT (default stdout is terminal window)





Session Name: [Enter or select a session name, e.g. myprogram with Re

Program Information

REQUIRED

File Name: DV_UM5_SGS_TKE_RAD_RRTM_MICRO_M2005

Arguments: [Enter any program arguments. Ex. -option foo]

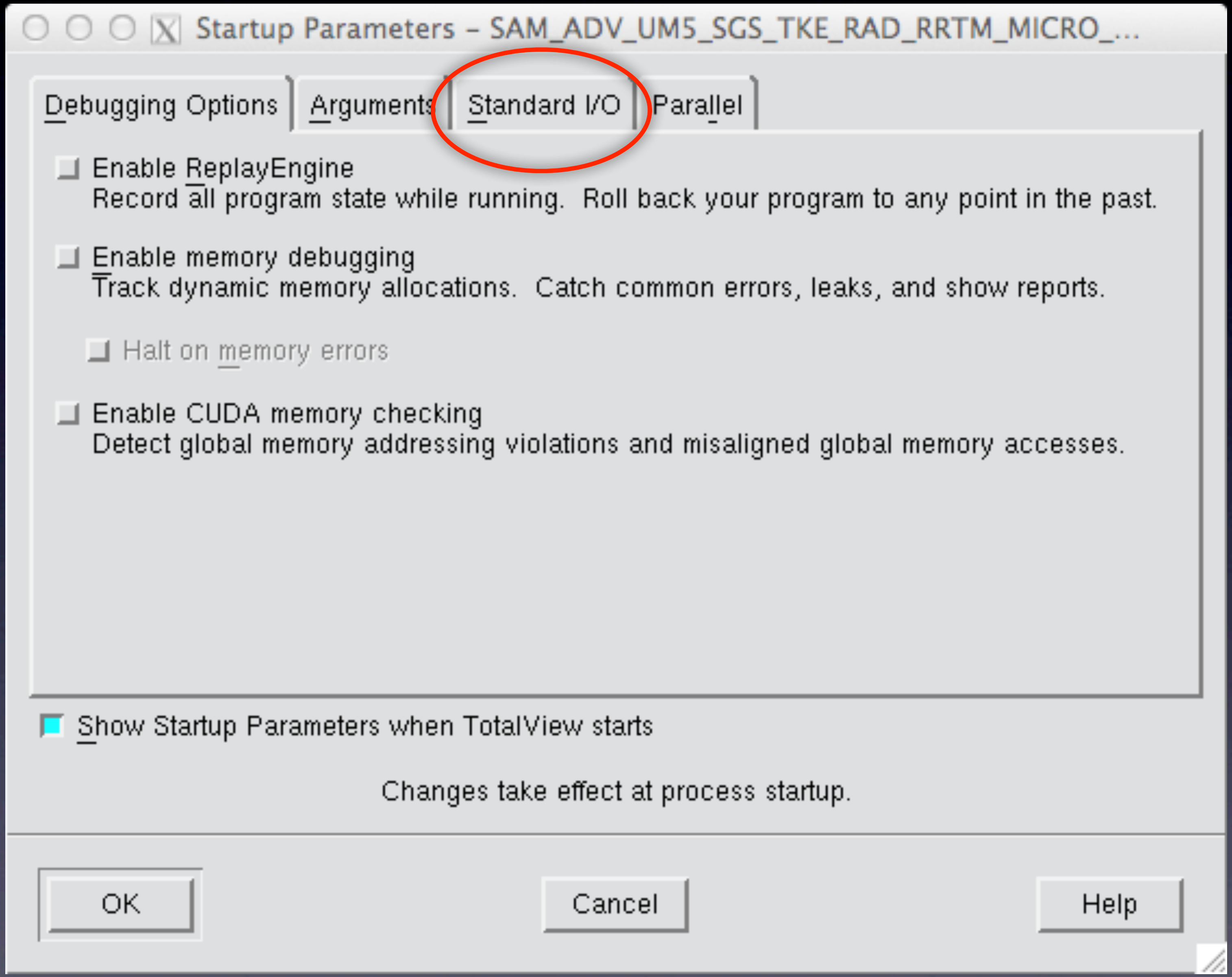
Debug on Host

saddleback.atmos.colostate.edu (local)

Press Start Session to start the debugging session.

Launching Totalview - serial job

- executable specified at launch
- get an window to set ENVIRONMENT



Startup Parameters – SAM_ADV_UM5_SGS_TKE_RAD_RRTM_MICRO_...

☐ Enable ReplayEngine
Record all program state while running. Roll back your program to any point in the past.

☐ Enable memory debugging
Track dynamic memory allocations. Catch common errors, leaks, and show reports.

☐ Halt on memory errors

☐ Enable CUDA memory checking
Detect global memory addressing violations and misaligned global memory accesses.

☑ Show Startup Parameters when TotalView starts

Changes take effect at process startup.

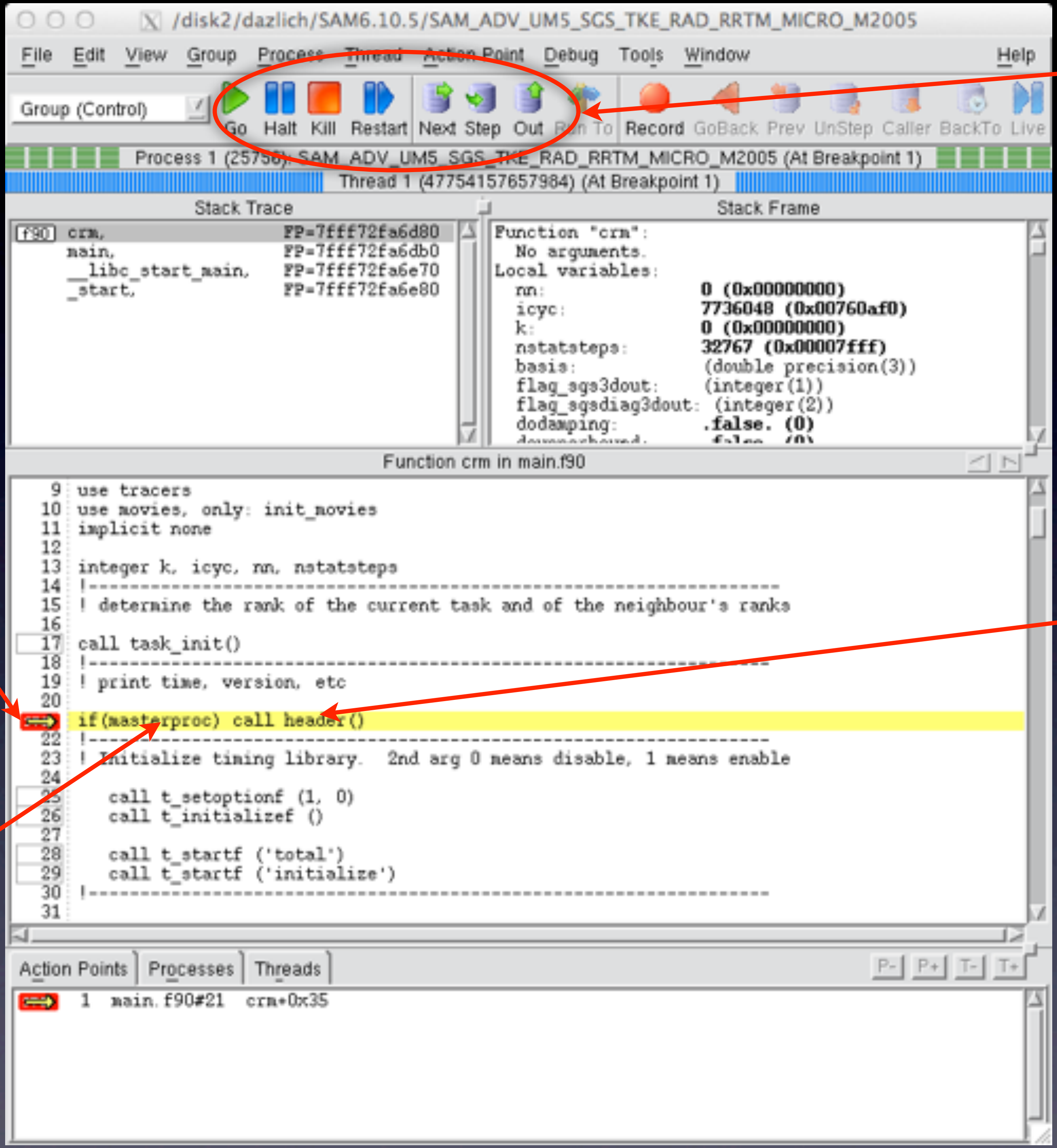
Totalview window

buttons to control execution

click to set a breakpoint

double-click to dive into subroutines

hover (scalar) or double-click (array) to see values


/disk2/dazlich/SAM6.10.5/SAM_ADV_UM5_SGS_TKE_RAD_RRTM_MICRO_M2005
File Edit View Group Process Thread Action Point Debug Tools Window Help
Group (Control)
Go Halt Kill Restart Next Step Out Run To Record GoBack Prev UnStep Caller BackTo Live
Stack Trace
Stack Frame
Action Points Processes Threads
P- P+ T- T+

```
Process 1 (25756): SAM_ADV_UM5_SGS_TKE_RAD_RRTM_MICRO_M2005 (At Breakpoint 1)
Thread 1 (47754157657984) (At Breakpoint 1)
```

```
f90  crm,                    FP=7fff72fa6d80
     main,                   FP=7fff72fa6db0
     __libc_start_main,      FP=7fff72fa6e70
     _start,                 FP=7fff72fa6e80
```

```
Function "crm":
  No arguments.
Local variables:
  nn:                 0 (0x00000000)
  icyc:               7736048 (0x00760af0)
  k:                  0 (0x00000000)
  nstatsteps:         32767 (0x00007fff)
  basis:              (double precision(3))
  flag_sgs3dout:      (integer(1))
  flag_sgsdiag3dout:  (integer(2))
  dodamping:          .false. (0)
```

Function crm in main.f90

```
 9  use tracers
10  use movies, only: init_movies
11  implicit none
12
13  integer k, icyc, nn, nstatsteps
14  !----------------------------------------------------------------------
15  ! determine the rank of the current task and of the neighbour's ranks
16
17  call task_init()
18  !----------------------------------------------------------------------
19  ! print time, version, etc
20
    if(masterproc) call header()
22  !----------------------------------------------------------------------
23  ! Initialize timing library.  2nd arg 0 means disable, 1 means enable
24
25     call t_setoptionf (1, 0)
26     call t_initializef ()
27
28     call t_startf ('total')
29     call t_startf ('initialize')
30  !----------------------------------------------------------------------
31
```

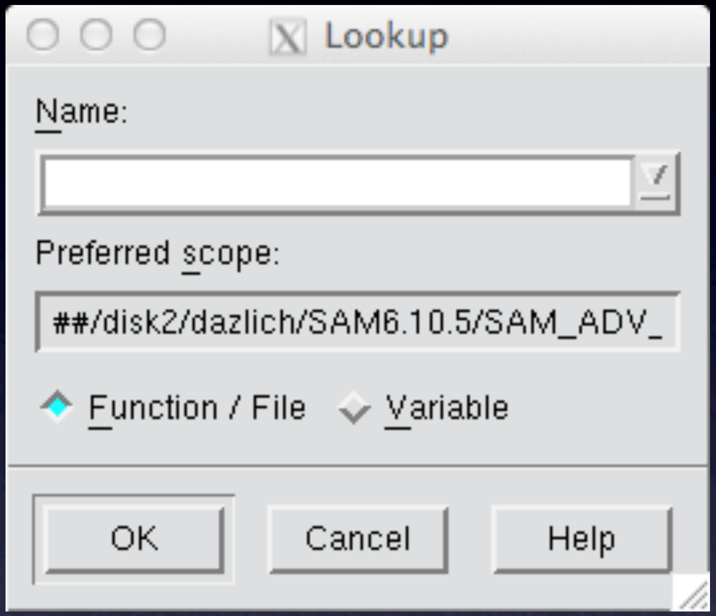
```
1   main.f90#21   crm+0x35
```

Execution control

- GO - run until execution terminated or a breakpoint is encountered
- HALT - stop execution immediately
- KILL - kill the program
- RESTART - start over and GO from the beginning
- NEXT - advance one line of code in current subroutine
- STEP - advance to next executable line of code - will step into a called subroutine
- OUT - return to the calling subroutine

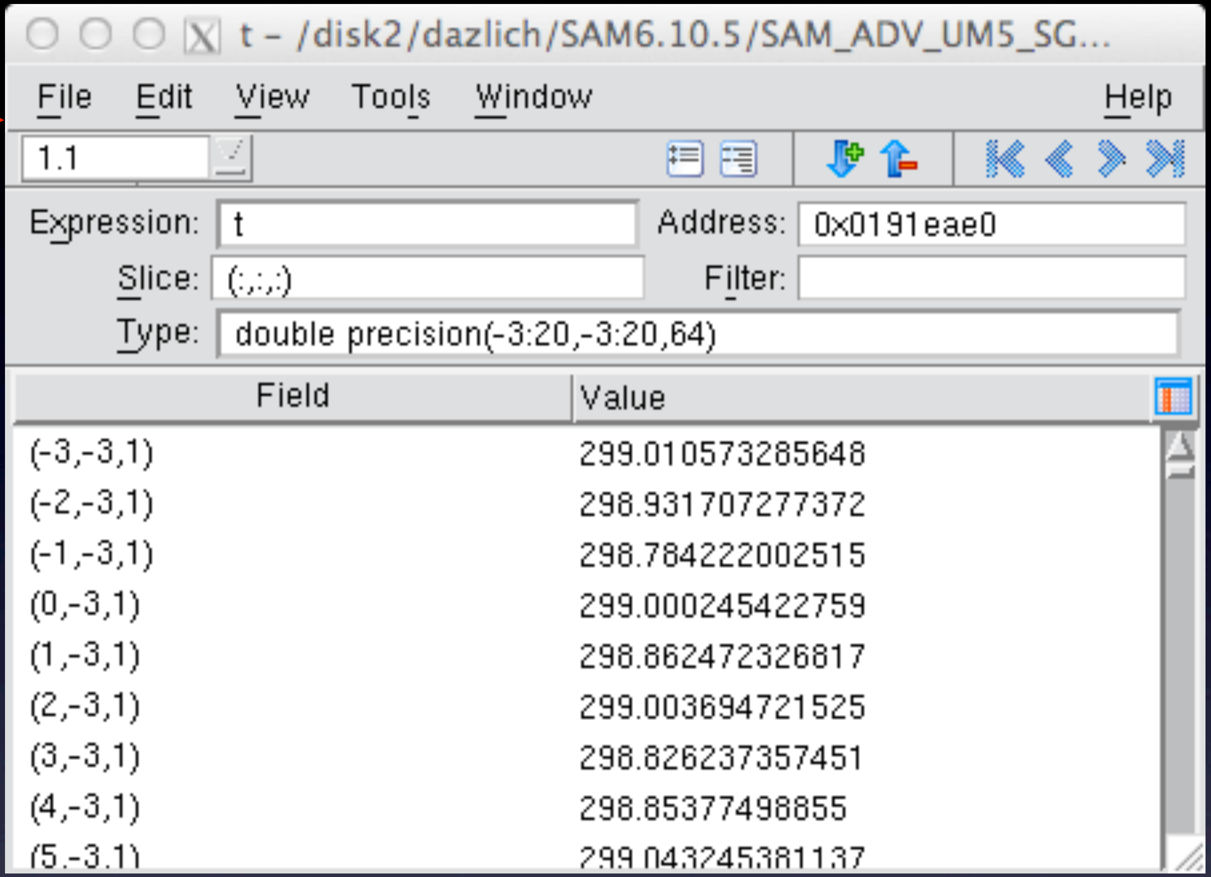
Navigating

- Double-click on a subroutine, scroll and repeat....
- OR File > Open Source



Examining Data

- Hover over a scalar ...OR...
- Double-click a variable
- Can select a slice of the array to view
- File > Save Pane will save values to disk
- Tools > Statistics gives summary of values
- Window > Duplicate copies the pane and lets you manipulate it independently of the original pane
- Main window: Tools > Fortran Modules lets you examine contents of a module - only way to get parameter values
- Execution must be halted to examine data

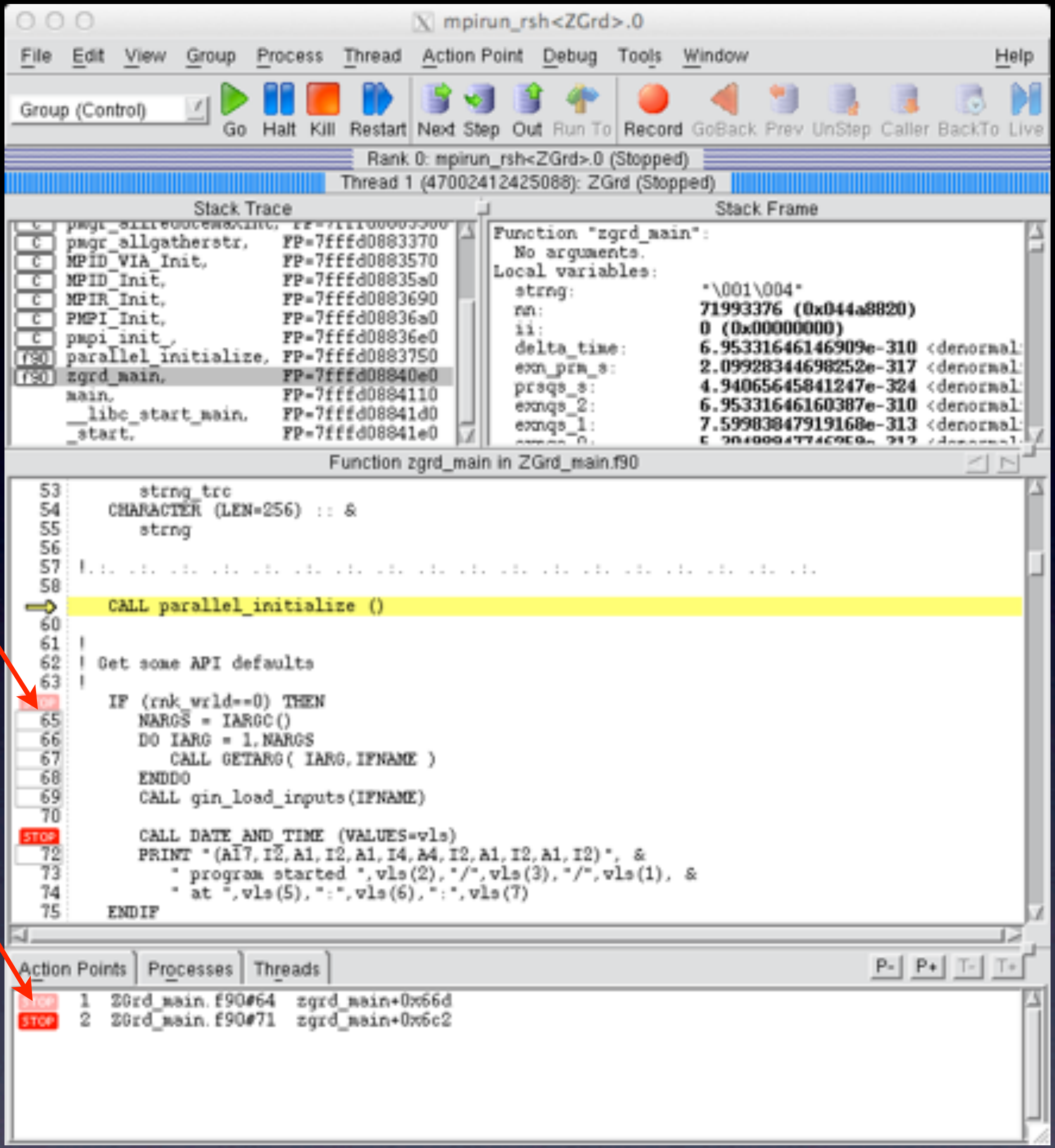


Expression: t Address: 0x0191eae0
Slice: (:,:,:) Filter:
Type: double precision(-3:20,-3:20,64)

Field	Value
(-3,-3,1)	299.010573285648
(-2,-3,1)	298.931707277372
(-1,-3,1)	298.784222002515
(0,-3,1)	299.000245422759
(1,-3,1)	298.862472326817
(2,-3,1)	299.003694721525
(3,-3,1)	298.826237357451
(4,-3,1)	298.85377498855
(5,-3,1)	299.043245381137

Breakpoints

- let the user define where execution will halt so data can be examined
- Set/remove by clicking on line number in left column
- can disable/enable by clicking on the list in the lower frame.
- double-clicking on line in lower frame takes the source frame there.


mpirun_rsh<ZGrd>.0
File Edit View Group Process Thread Action Point Debug Tools Window Help
Group (Control)
Go Halt Kill Restart Next Step Out Run To Record GoBack Prev UnStep Caller BackTo Live
Stack Trace
Stack Frame
Action Points Processes Threads
P- P+ T- T+

Rank 0: mpirun_rsh<ZGrd>.0 (Stopped)
Thread 1 (47002412425088): ZGrd (Stopped)

```
C     pmgr_allgatherstr,       FP=7fffd0883370
C     MPID_VIA_Init,           FP=7fffd0883570
C     MPID_Init,               FP=7fffd08835a0
C     MPIR_Init,               FP=7fffd0883690
C     PMPI_Init,               FP=7fffd08836a0
C     pmpi_init_,              FP=7fffd08836e0
f90   parallel_initialize,     FP=7fffd0883750
f90   zgrd_main,               FP=7fffd08840e0
      main,                    FP=7fffd0884110
      __libc_start_main,       FP=7fffd08841d0
      _start,                  FP=7fffd08841e0
```

```
Function "zgrd_main":
  No arguments.
Local variables:
  strng:               "\001\004"
  nn:                  71993376 (0x044a8820)
  ii:                  0 (0x00000000)
  delta_time:          6.95331646146909e-310 <denormal
  exn_prm_s:           2.09928344698252e-317 <denormal
  prsqs_s:             4.94065645841247e-324 <denormal
  exnqs_2:             6.95331646160387e-310 <denormal
  exnqs_1:             7.59983847919168e-313 <denormal
```

Function zgrd_main in ZGrd_main.f90

```
         strng_trc
      CHARACTER (LEN=256) :: &
         strng

 !.:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:.

      CALL parallel_initialize ()

 !
 ! Get some API defaults
 !
      IF (rnk_wrld==0) THEN
         NARGS = IARGC()
         DO IARG = 1,NARGS
            CALL GETARG( IARG,IFNAME )
         ENDDO
         CALL gin_load_inputs(IFNAME)

         CALL DATE_AND_TIME (VALUES=vls)
         PRINT "(A17,I2,A1,I2,A1,I4,A4,I2,A1,I2,A1,I2)", &
            " program started ",vls(2),"/",vls(3),"/",vls(1), &
            " at ",vls(5),":",vls(6),":",vls(7)
      ENDIF
```

```
STOP  1  ZGrd_main.f90#64  zgrd_main+0x66d
STOP  2  ZGrd_main.f90#71  zgrd_main+0x6c2
```

Evaluation Points

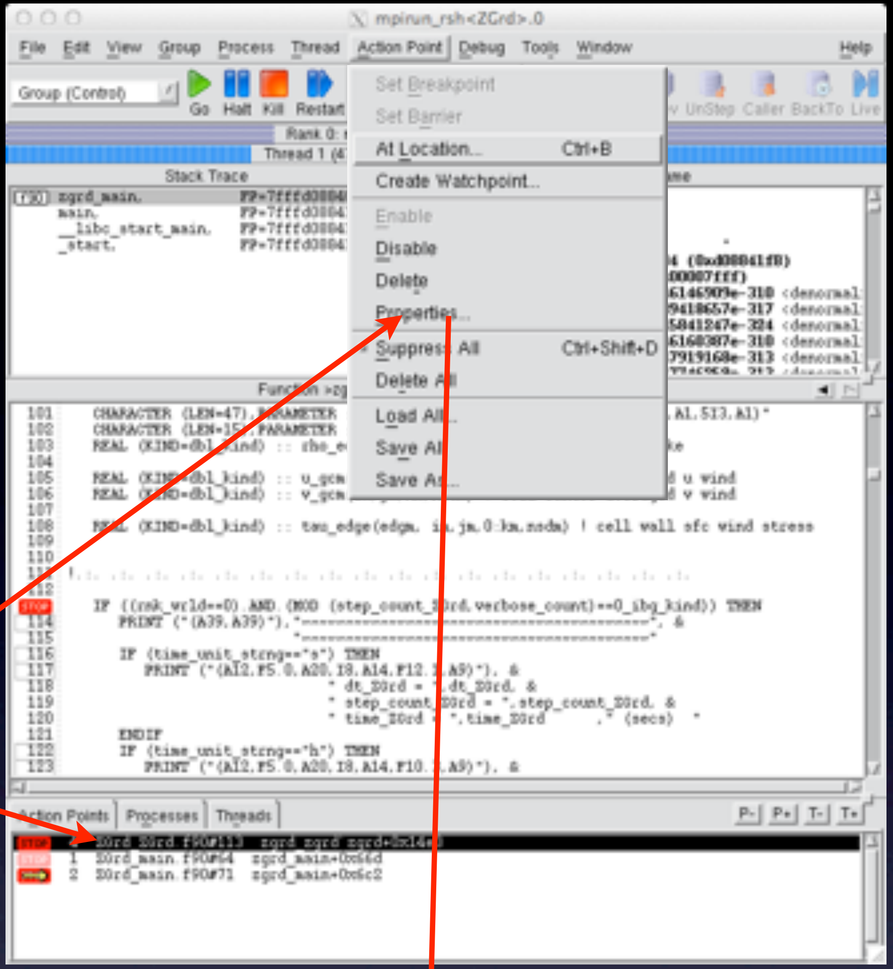
- Execution will halt every time a breakpoint is encountered - can be a nuisance in a do loop.

- You can be selective about when you will halt at a break point by converting it to an evaluation point.

- First, create the breakpoint. Then select it in the lower frame.

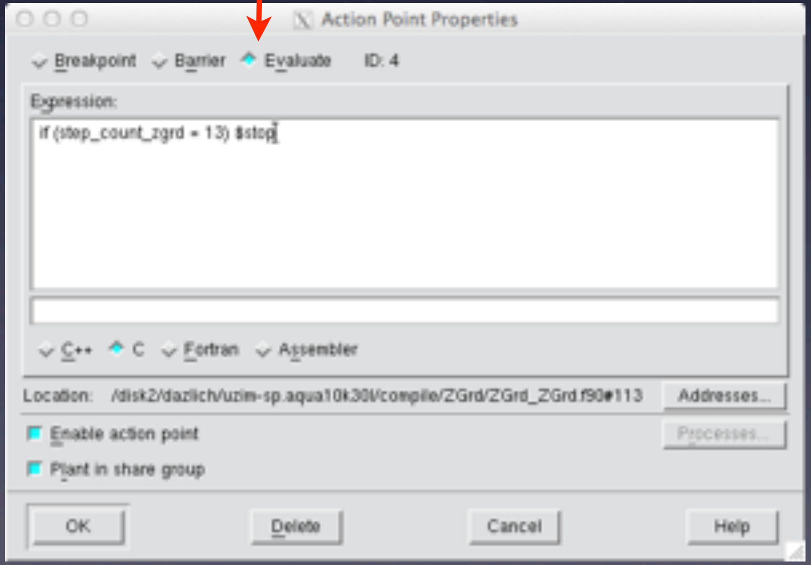
- Choose 'Properties' in the 'Action Point' menu

- Click 'Evaluate', enter the expression, then OK. Syntax is fortran except for the $stop.



```
      IF ((rnk_wrld==0) .AND. (MOD (step_count_ZGrd,verbose_count)==0_ibg_kind)) THEN
         PRINT ("(A39,A39)"),"=======================================", &
                             "======================================="
         IF (time_unit_strng=="s") THEN
            PRINT ("(A12,F5.0,A20,I8,A14,F12.1,A9)"), &
                   " dt_ZGrd = ",dt_ZGrd, &
                   " step_count_ZGrd = ",step_count_ZGrd, &
                   " time_ZGrd = ",time_ZGrd ," (secs)  "
         ENDIF
         IF (time_unit_strng=="h") THEN
            PRINT ("(A12,F5.0,A20,I8,A14,F10.1,A9)"), &
```

```
4  ZGrd ZGrd.f90#113   zgrd zgrd zgrd+0x14e
1  ZGrd_main.f90#64    zgrd_main+0x66d
2  ZGrd_main.f90#71    zgrd_main+0x6c2
```



ID: 4

Expression:

```
if (step_count_zgrd = 13) $stop
```

C++ [✓ C] Fortran Assembler

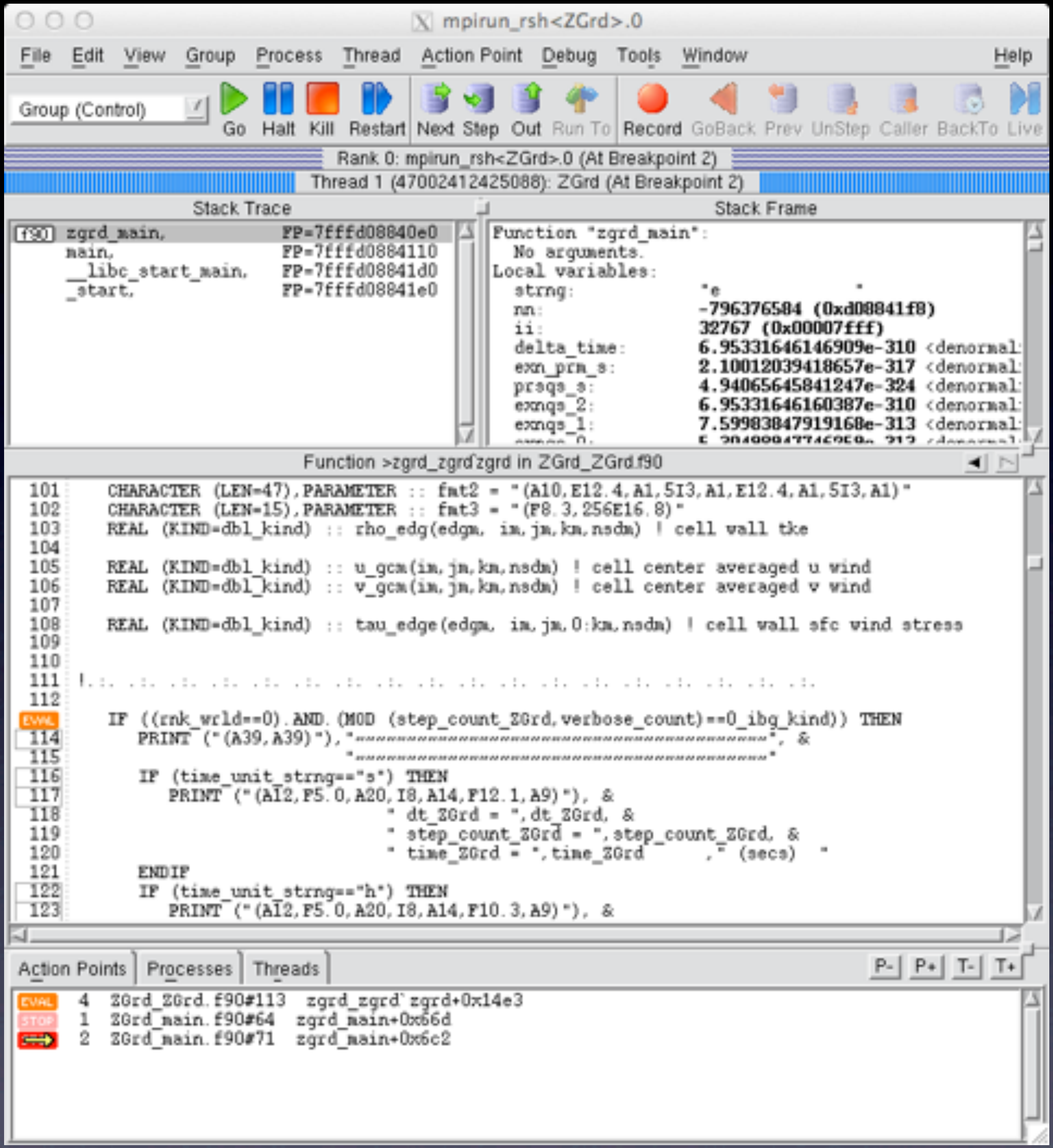
Location: /disk2/dazlich/uzim-sp.aqua10k30l/compile/ZGrd/ZGrd_ZGrd.f90#113

[✓] Enable action point

[✓] Plant in share group

Evaluation Points (2)

- The evaluation point is distinct from the breakpoint by color (orange) and label (eval)


mpirun_rsh<ZGrd>.0
File Edit View Group Process Thread Action Point Debug Tools Window Help
Group (Control)
Go Halt Kill Restart Next Step Out Run To Record GoBack Prev UnStep Caller BackTo Live
Stack Trace
Stack Frame
Action Points
Processes
Threads
P- P+ T- T+

Rank 0: mpirun_rsh<ZGrd>.0 (At Breakpoint 2)

Thread 1 (47002412425088): ZGrd (At Breakpoint 2)

```
f90 zgrd_main,             FP=7fffd08840e0
    main,                  FP=7fffd0884110
    __libc_start_main,     FP=7fffd08841d0
    _start,                FP=7fffd08841e0
```

```
Function "zgrd_main":
  No arguments.
Local variables:
  strng:              "e                 "
  nn:                 -796376584 (0xd08841f8)
  ii:                 32767 (0x00007fff)
  delta_time:         6.95331646146909e-310 <denormal
  exn_prm_s:          2.10012039418657e-317 <denormal
  prsqs_s:            4.94065645841247e-324 <denormal
  exnqs_2:            6.95331646160387e-310 <denormal
  exnqs_1:            7.59983847919168e-313 <denormal
```

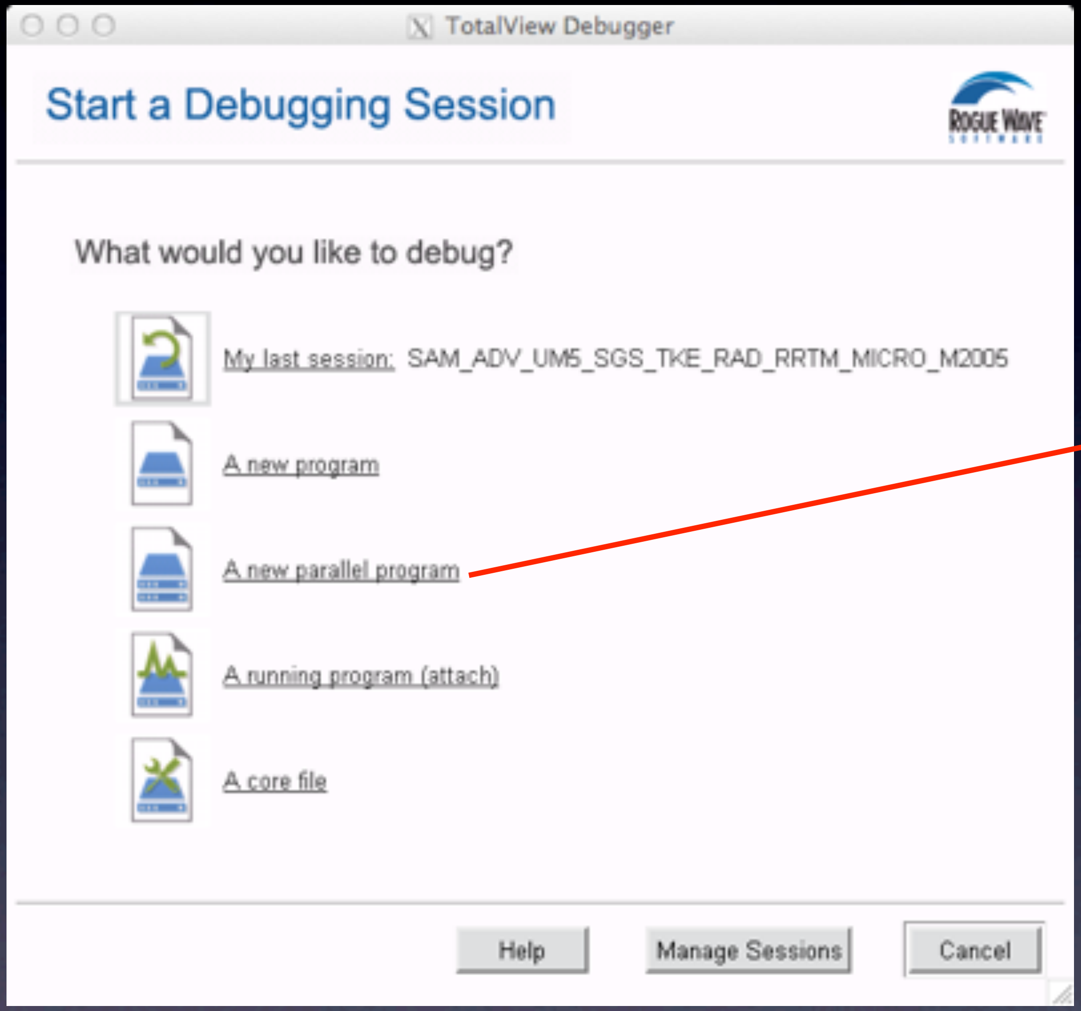
Function >zgrd_zgrd`zgrd in ZGrd_ZGrd.f90

```
   CHARACTER (LEN=47),PARAMETER :: fmt2 = "(A10,E12.4,A1,5I3,A1,E12.4,A1,5I3,A1)"
   CHARACTER (LEN=15),PARAMETER :: fmt3 = "(F8.3,256E16.8)"
   REAL (KIND=dbl_kind) :: rho_edg(edgm, im,jm,km,nsdm) ! cell wall tke

   REAL (KIND=dbl_kind) :: u_gcm(im,jm,km,nsdm) ! cell center averaged u wind
   REAL (KIND=dbl_kind) :: v_gcm(im,jm,km,nsdm) ! cell center averaged v wind

   REAL (KIND=dbl_kind) :: tau_edge(edgm, im,jm,0:km,nsdm) ! cell wall sfc wind stress


!.:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:.

   IF ((rnk_wrld==0).AND.(MOD (step_count_ZGrd,verbose_count)==0_ibg_kind)) THEN
      PRINT ("(A39,A39)"),"=======================================", &
                          "======================================="
      IF (time_unit_strng=="s") THEN
         PRINT ("(A12,F5.0,A20,I8,A14,F12.1,A9)"), &
                              " dt_ZGrd = ",dt_ZGrd, &
                              " step_count_ZGrd = ",step_count_ZGrd, &
                              " time_ZGrd = ",time_ZGrd       ," (secs)  "
      ENDIF
      IF (time_unit_strng=="h") THEN
         PRINT ("(A12,F5.0,A20,I8,A14,F10.3,A9)"), &
```

```
EVAL  4  ZGrd_ZGrd.f90#113  zgrd_zgrd`zgrd+0x14e3
STOP  1  ZGrd_main.f90#64   zgrd_main+0x66d
      2  ZGrd_main.f90#71   zgrd_main+0x6c2
```

Parallel Startup

- terminal command line : `totalview`



TotalView Debugger

Start a Debugging Session

What would you like to debug?

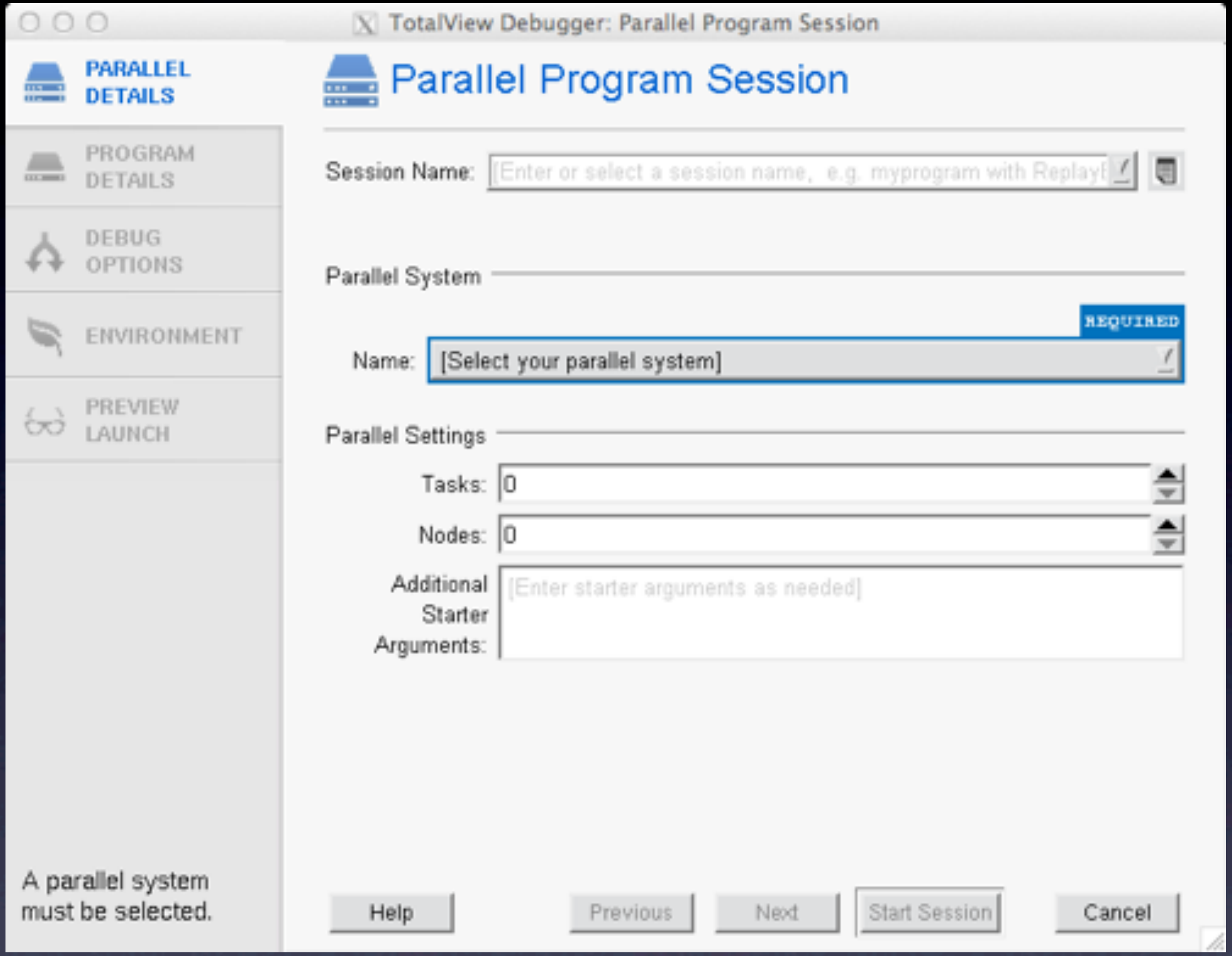
My last session: SAM_ADV_UM5_SGS_TKE_RAD_RRTM_MICRO_M2005

A new program

A new parallel program

A running program (attach)

A core file



TotalView Debugger: Parallel Program Session

Parallel Program Session

Session Name: [Enter or select a session name, e.g. myprogram with ReplayE]

Parallel System

Name: [Select your parallel system]

Parallel Settings

Tasks: 0

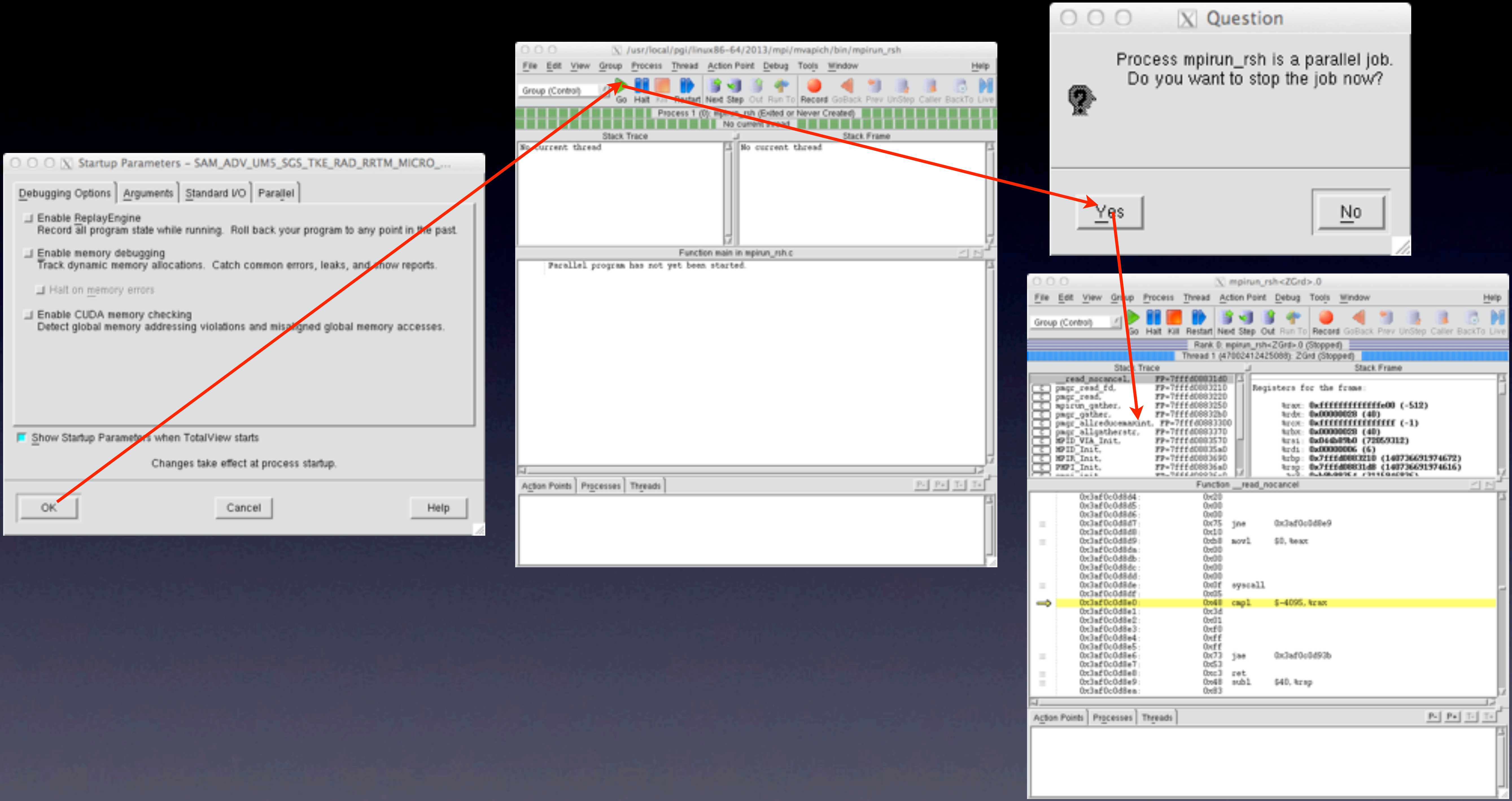
Nodes: 0

Additional Starter Arguments: [Enter starter arguments as needed]

A parallel system must be selected.

Parallel Startup (alternative)

- terminal command line : `mpirun -tv -np 4` *`executable`*



Startup Parameters – SAM_ADV_UM5_SGS_TKE_RAD_RRTM_MICRO_...

Debugging Options | Arguments | Standard I/O | Parallel

☐ Enable ReplayEngine
Record all program state while running. Roll back your program to any point in the past.

☐ Enable memory debugging
Track dynamic memory allocations. Catch common errors, leaks, and show reports.

☐ Halt on memory errors

☐ Enable CUDA memory checking
Detect global memory addressing violations and misaligned global memory accesses.

☑ Show Startup Parameters when TotalView starts

Changes take effect at process startup.

OK | Cancel | Help

/usr/local/pgi/linux86-64/2013/mpi/mvapich/bin/mpirun_rsh

Process 1 (0): mpirun_rsh (Exited or Never Created)
No current thread

```
No current thread
```

Function main in mpirun_rsh.c

```
Parallel program has not yet been started.
```

Question

Process mpirun_rsh is a parallel job.
Do you want to stop the job now?

Yes | No

mpirun_rsh<ZGrd>.0

Rank 0: mpirun_rsh<ZGrd>.0 (Stopped)
Thread 1 (47002412425088): ZGrd (Stopped)

```
__read_nocancel,       FP=7fffd00831d0
pmgr_read_fd,          FP=7fffd0083210
pmgr_read,             FP=7fffd0083220
mpirun_gather,         FP=7fffd0083250
pmgr_gather,           FP=7fffd00832b0
pmgr_allreducemaxint,  FP=7fffd0083300
pmgr_allgatherstr,     FP=7fffd0083370
MPID_VIA_Init,         FP=7fffd0083570
MPID_Init,             FP=7fffd00835a0
MPIR_Init,             FP=7fffd0083690
PMPI_Init,             FP=7fffd00836a0
```

```
Registers for the frame:

    %rax: 0xfffffffffffffe00 (-512)
    %rdx: 0x00000028 (40)
    %rcx: 0xffffffffffffffff (-1)
    %rbx: 0x00000028 (40)
    %rsi: 0x044b89b0 (72059312)
    %rdi: 0x00000006 (6)
    %rbp: 0x7fffd0083210 (140736691974672)
    %rsp: 0x7fffd00831d8 (140736691974616)
```

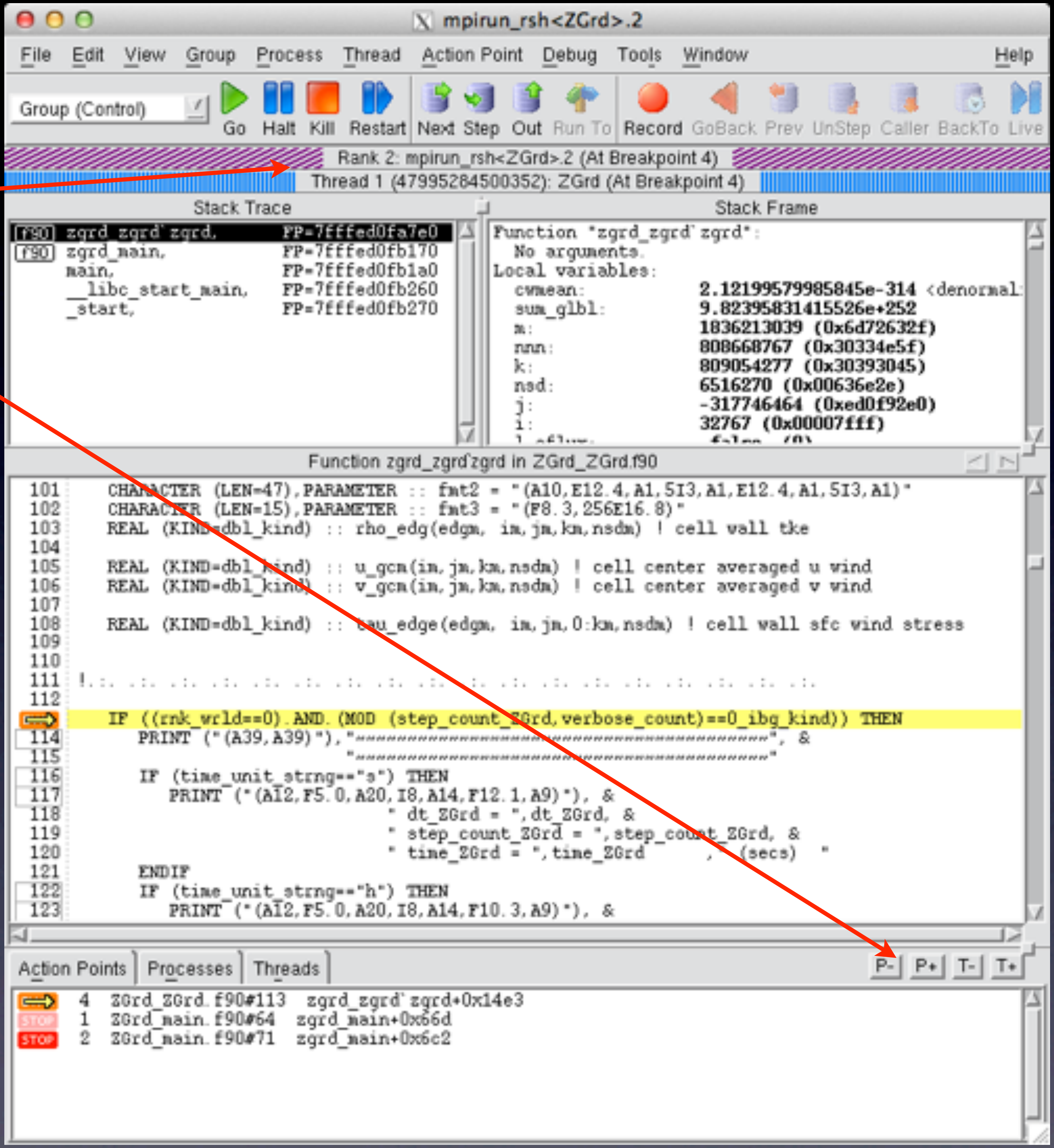
Function __read_nocancel

```
0x3af0c0d8d4:   0x20
0x3af0c0d8d5:   0x00
0x3af0c0d8d6:   0x00
0x3af0c0d8d7:   0x75  jne     0x3af0c0d8e9
0x3af0c0d8d8:   0x10
0x3af0c0d8d9:   0xb8  movl    $0, %eax
0x3af0c0d8da:   0x00
0x3af0c0d8db:   0x00
0x3af0c0d8dc:   0x00
0x3af0c0d8dd:   0x00
0x3af0c0d8de:   0x0f  syscall
0x3af0c0d8df:   0x05
0x3af0c0d8e0:   0x48  cmpl    $-4095, %rax
0x3af0c0d8e1:   0x3d
0x3af0c0d8e2:   0x01
0x3af0c0d8e3:   0xf0
0x3af0c0d8e4:   0xff
0x3af0c0d8e5:   0xff
0x3af0c0d8e6:   0x73  jae     0x3af0c0d93b
0x3af0c0d8e7:   0x53
0x3af0c0d8e8:   0xc3  ret
0x3af0c0d8e9:   0x48  subl    $40, %rsp
0x3af0c0d8ea:   0x83
```

File Edit View Group Process Thread Action Point Debug Tools Window Help
Group (Control)
Go Halt Kill Restart Next Step Out Run To Record GoBack Prev UnStep Caller BackTo Live
Stack Trace
Stack Frame
Action Points Processes Threads

Parallel Navigation

- Default process is rank 0
- Rank and thread number displayed across top
- Navigate with the P- and P+ buttons
- Must open a new data pane to display data when process is changed
- ‘Tools’ drop down menu has ‘Message Queue’ to examine pending MPI messages



Rank 2: mpirun_rsh<ZGrd>.2 (At Breakpoint 4)

Thread 1 (47995284500352): ZGrd (At Breakpoint 4)

```
f90  zgrd_zgrd`zgrd,        FP=7fffed0fa7e0
f90  zgrd_main,             FP=7fffed0fb170
     main,                  FP=7fffed0fb1a0
     __libc_start_main,     FP=7fffed0fb260
     _start,                FP=7fffed0fb270
```

```
Function "zgrd_zgrd`zgrd":
  No arguments.
Local variables:
  cwmean:          2.12199579985845e-314 <denormal
  sum_glbl:        9.82395831415526e+252
  m:               1836213039 (0x6d72632f)
  nnn:             808668767 (0x30334e5f)
  k:               809054277 (0x30393045)
  nsd:             6516270 (0x00636e2e)
  j:               -317746464 (0xed0f92e0)
  i:               32767 (0x00007fff)
```

Function zgrd_zgrd`zgrd in ZGrd_ZGrd.f90

```
    CHARACTER (LEN=47),PARAMETER :: fmt2 = "(A10,E12.4,A1,5I3,A1,E12.4,A1,5I3,A1)"
    CHARACTER (LEN=15),PARAMETER :: fmt3 = "(F8.3,256E16.8)"
    REAL (KIND=dbl_kind) :: rho_edg(edgm, im,jm,km,nsdm) ! cell wall tke

    REAL (KIND=dbl_kind) :: u_gcm(im,jm,km,nsdm) ! cell center averaged u wind
    REAL (KIND=dbl_kind) :: v_gcm(im,jm,km,nsdm) ! cell center averaged v wind

    REAL (KIND=dbl_kind) :: tau_edge(edgm, im,jm,0:km,nsdm) ! cell wall sfc wind stress


!.:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:. .:.

    IF ((rnk_wrld==0).AND.(MOD (step_count_ZGrd,verbose_count)==0_ibg_kind)) THEN
       PRINT ("(A39,A39)"),"=======================================", &
                           "======================================="
       IF (time_unit_strng=="s") THEN
          PRINT ("(A12,F5.0,A20,I8,A14,F12.1,A9)"), &
                                 " dt_ZGrd = ",dt_ZGrd, &
                                 " step_count_ZGrd = ",step_count_ZGrd, &
                                 " time_ZGrd = ",time_ZGrd      ," (secs)  "
       ENDIF
       IF (time_unit_strng=="h") THEN
          PRINT ("(A12,F5.0,A20,I8,A14,F10.3,A9)"), &
```

```
4  ZGrd_ZGrd.f90#113   zgrd_zgrd`zgrd+0x14e3
1  ZGrd_main.f90#64    zgrd_main+0x66d
2  ZGrd_main.f90#71    zgrd_main+0x6c2
```