

Total View

TotalView is a GUI-based source code multi-process, multi-thread debugger.

License and Limitations for Anselm Users

On Anselm users can debug OpenMP or MPI code that runs up to 64 parallel processes. These limitation means that:

1 user can debug up 64 processes, or 32 users can debug 2 processes, etc.

Debugging of GPU accelerated codes is also supported.

You can check the status of the licenses here:

```
cat /apps/user/licenses/totalview_features_state.txt
```

```
# totalview
# -----
# FEATURE                                TOTAL    USED    AVAIL
# -----
TotalView_Team                          64       0      64
Replay                                  64       0      64
CUDA                                    64       0      64
```

Compiling Code to run with TotalView

Modules

Load all necessary modules to compile the code. For example:

```
module load intel
```

```
module load impi ... or ... module load openmpi/X.X.X-icc
```

Load the TotalView module:

```
module load totalview/8.12
```

Compile the code:

```
mpicc -g -O0 -o test_debug test.c
```

```
mpif90 -g -O0 -o test_debug test.f
```

Compiler flags

Before debugging, you need to compile your code with theses flags:

`-g**` : Generates extra debugging information usable by GDB. `-g3**` includes even more debugging information. This option is available for GNU and INTEL C/C++ and Fortran compilers.

`-O0**` : Suppress all optimizations.**

Starting a Job with TotalView

Be sure to log in with an X window forwarding enabled. This could mean using the `-X` in the ssh:

```
ssh -X username@anselm.it4i.cz
```

Other options is to access login node using VNC. Please see the detailed information on how to use graphic user interface on Anselm here.

From the login node an interactive session with X windows forwarding (`-X` option) can be started by following command:

```
qsub -I -X -A NONE-0-0 -q qexp -lselect=1:ncpus=16:mpiprocs=16,walltime=01:00:00
```

Then launch the debugger with the totalview command followed by the name of the executable to debug.

Debugging a serial code

To debug a serial code use:

```
totalview test_debug
```

Debugging a parallel code - option 1

To debug a parallel code compiled with **>OpenMPI** you need to setup your TotalView environment:

Please note:** To be able to run parallel debugging procedure from the command line without stopping the debugger in the mpiexec source code you have to add the following function to your `~/tvdrc` file:

```
proc mpi_auto_run_starter {loaded_id} {
    set starter_programs {mpirun mpiexec orterun}
    set executable_name [TV::symbol get $loaded_id full_pathname]
    set file_component [file tail $executable_name]
```

```

        if {[lsearch -exact $starter_programs $file_component] != -1} {
            puts "*****"
            puts "Automatically starting $file_component"
            puts "*****"
            dgo
        }
    }
}

```

Append this function to TotalView's image load callbacks so that
TotalView run this program automatically.

```
dlappend TV::image_load_callbacks mpi_auto_run_starter
```

The source code of this function can be also found in

```
/apps/mpi/openmpi/intel/1.6.5/etc/openmpi-totalview.tcl
```

You can also add only following line to you ~/.tvdr file instead of the entire function:

```
source /apps/mpi/openmpi/intel/1.6.5/etc/openmpi-totalview.tcl**
```

You need to do this step only once.

Now you can run the parallel debugger using:

```
mpirun -tv -n 5 ./test_debug
```

When following dialog appears click on “Yes”

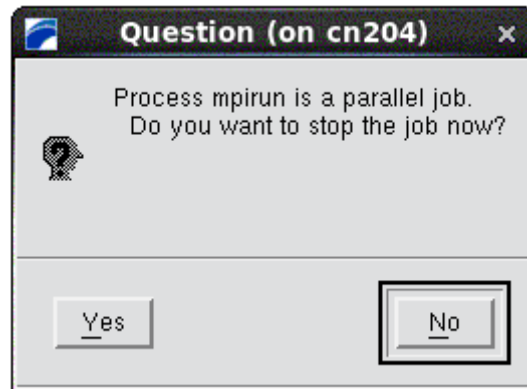


Figure 1:

At this point the main TotalView GUI window will appear and you can insert the breakpoints and start debugging:

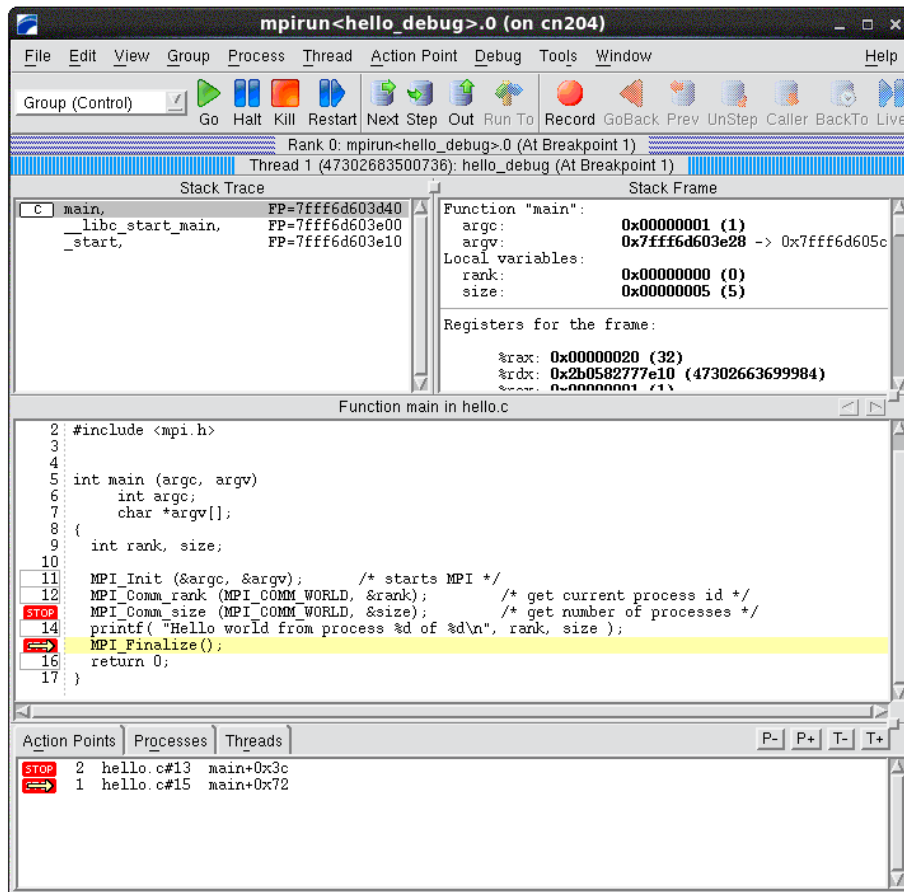


Figure 2:

Debugging a parallel code - option 2

Other option to start new parallel debugging session from a command line is to let TotalView to execute mpirun by itself. In this case user has to specify a MPI implementation used to compile the source code.

The following example shows how to start debugging session with Intel MPI:

```
module load intel/13.5.192 impi/4.1.1.036 totalview/8/13
```

```
totalview -mpi "Intel MPI-Hydra" -np 8 ./hello_debug impi
```

After running previous command you will see the same window as shown in the screenshot above.

More information regarding the command line parameters of the TotalView can be found TotalView Reference Guide, Chapter 7: TotalView Command Syntax.

Documentation

[1] The TotalView documentation web page is a good resource for learning more about some of the advanced TotalView features.