TDDC78: Lab Report

Lab 4: Totalview & ITAC

Name	PIN	Email
Alexander Yngve	930320-6651	aleyn573@student.liu.se
Pål Kastman	851212-7575	palka285@student.liu.se

Contents

1   Totalview

Most of the time we write our code in Emacs, so we never use built-in debuggers as can be found in for some IDE:s, such as Eclipse. The only other debugger that we have tried out in previous courses before this one, is GDB. The most noticable difference between these two is of course that GDB is run from a terminal while Totalview is a graphical tool.

After fiddling with Totalview for a while we managed to get our source code visible in the debugger and setting breakpoints and running the program. The code contained no bugs that we were aware of, so we generated one by switching the operands of a subtracion as can be seen in listing 1 and 2.

Listing 1 – Original code.

```
threshold_filter(data, xsize, yends[rank]-ystarts[rank],
    colmax, world_size);
```

Listing 2 – Code with a bug in it.

```
threshold_filter(data, xsize, ystarts[rank]-yends[rank],
    colmax, world_size);
```

The result of the bug is that the filter is not doing anything, which makes it logical to start debugging at the call to the filter. In figure 1 a breakpoint has been set at the call to the filter. In figure 2 the debugger has entered the filter function where we can see that the argument ysize has a negative value causing the filter to never enter the its for-loop. The bug has been found!
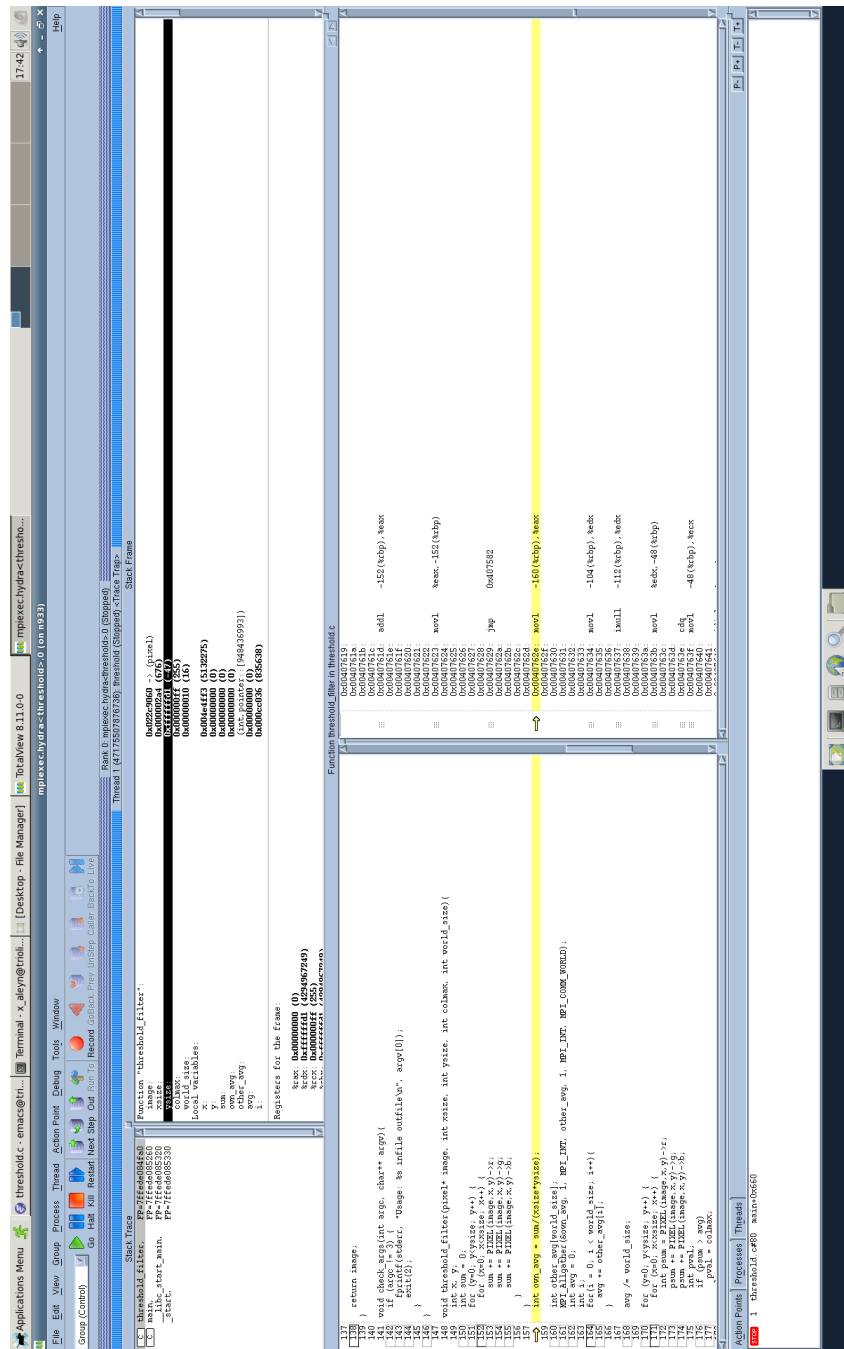
Alexander Yngve
Pål Kastman

Figure 1 – Setting a breakpoint at the call to the filtering function.

4
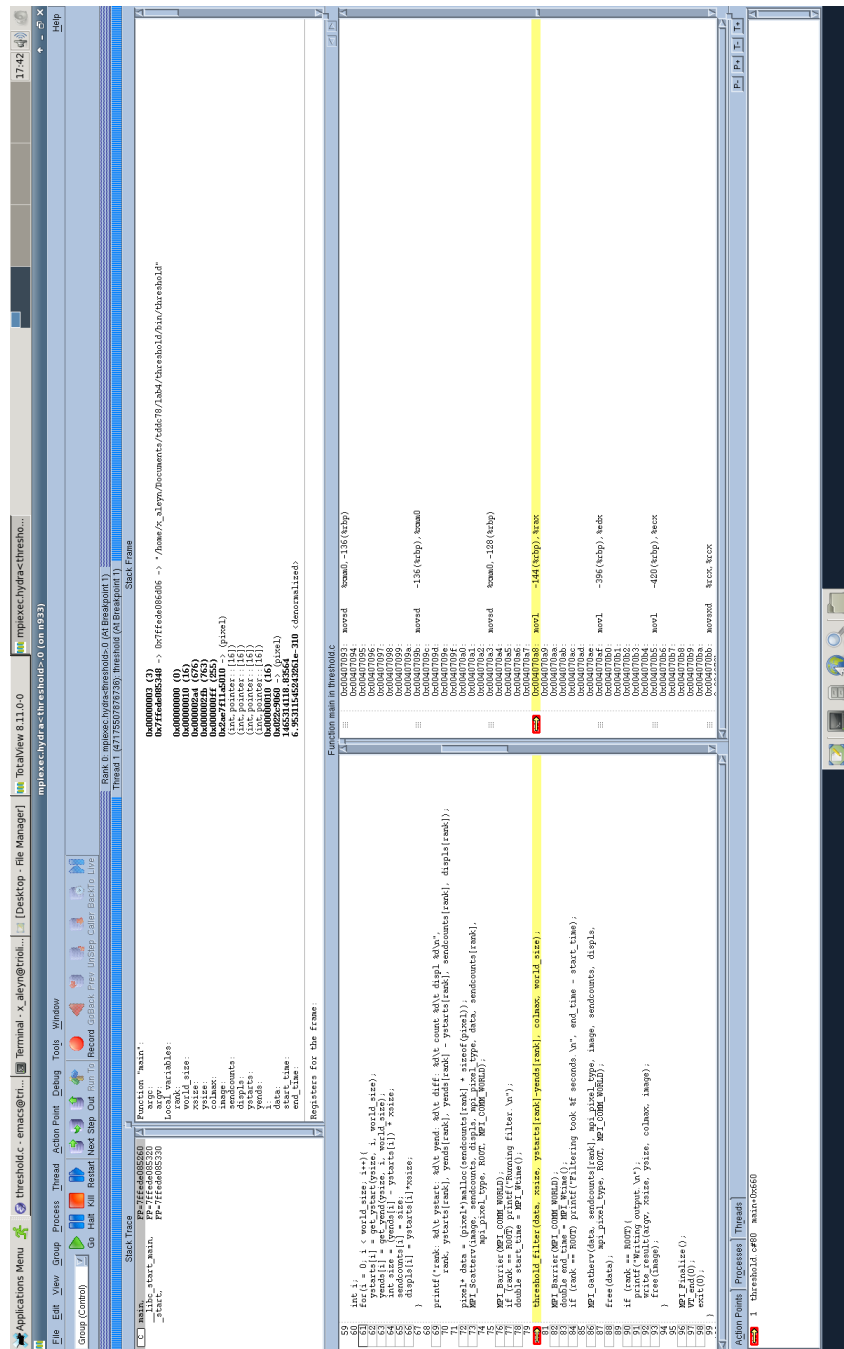
Figure 2 – A faulty value found in the arguments to the filter.

2   ITAC

After a bit of trouble with compiling and linking our program with ITAC functions we finally got it to work. ITAC was quite straight forward to use and with ease we could see which functions in our program consumed the most time. This can be seen in figure 3. As can be seen no single function seem to be to time consuming. And the one that takes the most time is MPI_Scatterv which can't be optimized further.

Figure 3 – Intel ITAC.

3   Discussion

Since using debuggers is quite a rare thing for us, Totalview felt very tricky to use. This is to be expected because Totalview is an advanced debugger, more advanced than GDB since it is specialized for multithreaded programs. However, when writing large parallel programs, we think it might be worth learning to use a debugger like Totalview, since a lot of bugs in parallel programs are very confusing and much harder to find than in serial programs.

ITAC on the other hand, was quite easy to use but a bit difficult to install and compile. Once you figured it out though it was a breeze. If you need to speed up your program and dont know where to start, it is definitely worth learning ITAC.