

Implementation of a remote-host debugger for parallel programs and evaluation of visual forms of representation

- ▶ Author: Nicolas With
- ▶ Type: Master's Thesis
- ▶ Date: 2023-05-09
- ▶ Reviewers: Jun.-Prof. Dr. Michael Kuhn, Jun.-Prof. Dr. Christian Lessig
- ▶ Supervisors: Michael Blesele

Parallel distributed computing has become increasingly important over the years as the demand for computing power has grown beyond the capabilities of traditional single-core applications. Developing parallel programs for these systems is much more complex than developing traditional programs, because it introduces a whole new set of problems. Debugging distributed parallel programs is even more complex because it is not just one program, perhaps containing multiple threads, but many instances of the same program, started on multiple nodes. Many individual developers and small working groups do not have resources to buy a license for a proprietary debugger. Therefore, this work aims to provide a free and open source parallel debugger for distributed systems called Parallel GDB. Parallel GDB is a fully-working source-level debugger with a GUI to support direct interaction with the source code. A startup dialog with exportable configuration files simplifies starting the debugger. The target program is automatically launched on the selected debug platform, even via SSH. Breakpoints can be set directly in the source code. A dialog to select the processes for which they should be active is provided. There is also an option to stop all selected processes if this breakpoint is hit by any of the previously selected processes. Source files containing processes are automatically opened in a notebook. If a source file cannot be found on the computer Parallel GDB is running on, the user can open another file instead. Also, any source file can be opened in advance, for example to set a breakpoint in it. Parallel GDB also supports following a specific process, jumping and scrolling in the source files to the current location of the selected process. This debugger is built completely from open-source libraries and thus is itself completely open-source and free. The evaluation shows that Parallel GDB itself would have an almost constant launch time regardless of the number of processes, but is limited by the launcher and the debug platform used. The need for serialization by the launcher and synchronization for resource allocation on the cluster results in a linear launch time in reality. The analysis of a specific use case shows that the implemented design achieves the desired performance. Both commercial debuggers compared run on the remote cluster using X-forwarding, so the GUI is really slow. Since Parallel GDB runs on the host machine, it does not suffer from this performance problem. However, it also shows that this work needs to be extended, at least with a call stack analyzer, to reach its full potential.