

## Looking for Job Security? Consider becoming an HPC Performance Tool Developer

Bernd Mohr, Jülich Supercomputing Centre Institute for Advanced Simulation



When I started to work on performance tools for parallel computers 25 years ago, I wasn't sure how long I would be able to work in this interesting and exciting area of high performance computing. Performance was always in the center of HPC, so anyone helping application developers to optimize and tune their codes were in high demand. But parallel computing and HPC became "mainstream" with the new millennium: Cluster systems made HPC affordable also for smaller universities and enterprises, and with the Message Passing Interface (MPI) there was a portable and well-recognized standard for programming these systems. I imagined that over time, parallel programming would become easy and everyone would be able to write portable, efficient and high-performance codes. Boy was I wrong!

What happened? In order to satisfy the ever increasing hunger for even more performance, computer architects came up with ever more complex designs: we went from simple clusters of single-core nodes, first to SMP, then to NUMA multi-core nodes. And as these homogeneous designs were not complex enough, one or more accelerators in the form of GPUs, DSPs, FPGAs, special processors like the IBM Cell, or Intel Xeon Phi were added to each node. Simple 2D-mesh interconnects were replaced by multi-dimensional (5D or 6D) tori or various forms of complicated trees (e.g., Dragonfly). Memory in various forms and caches are spread out over all levels of the hardware architecture and are shared more or less between various processing elements.

To get the desired high performance out of these beasts, an HPC application developer had to distribute the necessary calculations over all the processing cores available in the system, so that they are all busy all the time and the data needed for the calculations are in the right places in the cache and memory hierarchy at the

right time. Ideally, in addition, an HPC program is written in a way that it adapts to the specific configuration of a specific supercomputer it runs on, so that it always achieves high performance independent from the system used. Sounds simple, right?

To their rescue came computer scientists who provided new parallel programming models to ease the burden of the application developer to parallelize and tune their codes (cough, cough). For inter-node communication and synchronization, there is not only MPI but also PGAS languages (e.g., Co-array Fortran or UPC) or libraries (e.g., GPI or OpenSHMEM). Intra-node shared-memory programming is supported by a myriad of thread models (POSIX, Windows, Java, ACE, Boost, QT and whatever threads), tasking models (OmpSs, Cilk++, MTAPI, etc.) or combinations of threads and tasks (OpenMP or TBB). Finally, to fully exploit accelerators, there is CUDA, OpenCL, OpenACC, or HMPP, just to name a few. Did I forget something? For sure.

So to harness the full power of the cluster systems, programmers just use the right programming model or even a combination of them, so they easily can orchestrate the work and data distribution for their code. Not a big deal, or is it?

So, in reality programming HPC systems is still very complex and if you manage to create a correctly working program, it does not automatically mean you get a very efficiently working program. Luckily, there are still some groups out there dedicated to helping these “poor” application developers. They want to make their life easy by developing and providing techniques, procedures and software to analyze the performance of the parallel codes and to locate performance bottlenecks in them and their causes.

Have I kindled your interest that you want to learn more about it? Luckily, there is a whole session (“[Performance Measurement Tools](#) [1]”) on this topic at the upcoming International Supercomputing Conference in Leipzig. The following ISC tutorials - “Practical Hybrid Parallel Application Performance Engineering”, “Node-Level Performance Engineering” and “I/O Performance Optimizations on Large-Scale HPC Systems” will also address this topic.

See you soon!

*This blog was originally published on [ISC Events](#) [2] and is reproduced here with permission.*

**Source URL (retrieved on 02/01/2015 - 8:20am):**

<http://www.scientificcomputing.com/blogs/2014/04/looking-job-security-consider-becoming-hpc-performance-tool-developer>

**Links:**

[1] [http://www.isc-events.com/isc14\\_ap/sessiondetails.htm?t=session&o=37&a=select&ra=personendetails](http://www.isc-events.com/isc14_ap/sessiondetails.htm?t=session&o=37&a=select&ra=personendetails)

[2] <http://www.isc-events.com/isc14/blogs.html>

