Keeping up with Growing Machine Sizes: Challenges and Opportunities for Scaling Tools

Martin Schulz
schulzm@llnl.gov

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
PO Box 808, L-560, Livermore, CA 94551, USA

Current high-end HPC systems are already scaling well beyond 10,000 processors and Blue Gene/L already has over 200,000 cores. Combined with the recent push towards multi- and many-core chips, machines are expected to continue to scale even further reaching over a million cores in the next few years. While this promises unprecedented compute power and opens the door for new scientific discoveries through advanced simulation, it comes at the price of increased complexity in both hardware and software.

To deal with this complexity, users will require programming environments that scale with the machine. This includes performance analysis and debugging tools, which must be capable of collecting, analyzing, and presenting data from all cores in a system. To satisfy these requirements we cannot simply scale existing tool solutions that work on few hundred nodes; instead we require a set of new techniques that are explicitly designed and optimized for scale.

In this talk I will provide an overview of the challenges tools face when having to scale to current and future high-end machines. I will discuss open research questions as well as highlight existing solutions. In particular, I will focus on three approaches that we are currently pursuing in the Department of Energy's (DOE) Advanced Simulation and Computing (ASC) program: the use of hierarchical communication and online analysis in the *Stack Trace Analysis Tool (STAT)* [1]; the use of online compression techniques to store trace data in the *ScalaTrace* framework [3, 2]; and the need for application specific rapid tool prototyping, as supported by the P^N *MPI infrastructure* [4].

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48 (UCRL-CONF-404661).

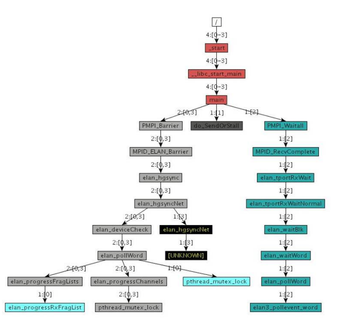

Fig. 1. STAT stack prefix tree.

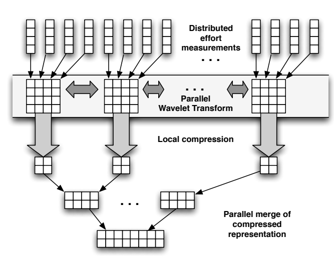

Fig. 2. Parallel Wavelet Compression.

STAT is a lightweight debugging tool that identifies process equivalence classes. It groups processes that exhibit similar behavior by sampling stack traces from each task of the parallel application over time and merging them into a call graph prefix tree (Figure 1) using a scalable and hierarchical on-line analysis. The resulting call graph prefix tree intuitively represents the application's behavior classes over space and time, and allows users to focus their debugging efforts to few enough nodes for efficient use of conventional debuggers.

ScalaTrace is an MPI tracing framework that extracts the underlying communication pattern from MPI applications and then uses this information to store the corresponding traces compactly. Using this approach results in traces that are orders of magnitude smaller than with conventional trace formats, often even constant size independent of the number of nodes and tasks. We are currently enhancing this appraoch with data from additional sources, such as phase-based timing information, and are investigating scalable ways for compressing this data, e.g., adaptive histograms or distributed wavelet compression (Figure 2).

P^{N}MPI extends the PMPI profiling interface to support multiple concurrent PMPI-based tools by allowing users to assemble tool stacks dynamically. In addition, P^{N}MPI enables a transparent virtualization of MPI execution environments as well as allows modules to switch between tool stacks at runtime. The latter provides the ability to restrict the application of existing, unmodified tools to a dynamic subset of MPI calls or even call sites. Using these features, P^{N}MPI enables users to assemble, to specialize, and to apply new MPI tools quickly.

These are just a few components that contribute to scalable tool solutions. Several important challenges remain and need to be addressed by the HPC tool community, e.g., scalable tool launch, data transport and storage, large scale process control, or efficient instrumentation. No single tool infrastructure or development group will be able to cover all of these aspects alone. We therefore require a community-wide effort towards a scalable, widely ported, and truly interoperable set of tool components.

References

1. D. C. Arnold, D. H. Ahn, B. R. de Supinski, G. L. Lee, B. P. Miller, and M. Schulz. Stack trace analysis for large scale debugging. In *International Parallel and Distributed Processing Symposium*, 2007.
2. B. R. de Supinski, R. Fowler, T. Gamblin, F. Mueller, P. Ratn, and M. Schulz. An open infrastructure for scalable, reconfigurable analysis. In *To appear in the Proceedings of the International Workshop on Scalable Tools for High-End Computing (STHEC)*, 2008.
3. M. Noeth, F. Mueller, M. Schulz, and B. R. de Supinski. Scalable compression and replay of communication traces in massively parallel environments. In *International Parallel and Distributed Processing Symposium*, Apr. 2007.
4. M. Schulz and B. R. de Supinski. P^{N}MPI tools a whole lot greater than the sum of their parts. In *Supercomputing 2007 (SC'07)*, 2007.