

Speeding up Science Through Parametric Optimization on HPC Clusters

Wednesday, 22 March 2023 14:20 (20 minutes)

Science is constantly encountering parametric optimization problems whose computer-aided solutions require enormous resources. At the same time, there is a trend towards developing increasingly powerful computer clusters. Geneva is currently one of the best available frameworks for distributed optimization of large-scale problems with highly nonlinear quality surfaces. It is a great tool to be used in wide-area networks such as Grids and Clouds. However, it is not user-friendly for scheduling on high-performance computing clusters and supercomputers. Another issue is that it only provides a framework for parallelizing workloads on the population level of optimization algorithms, but does not support distributed parallelization of the cost function itself. For this reason, a new software component for network communication – called MPI-Consumer – has been developed.

In Geneva's system architecture, the server node runs the optimization algorithms and distributes candidate solutions to clients. The clients evaluate the candidate solutions based on a user-defined cost function and then send the result back to the server.

When scaling to high-dimensional problems and hundreds or even thousands of nodes, the server's performance is a fundamental challenge, because the speed of answering client requests has a direct impact on the clients' CPU efficiency. We tackle this challenge by making immense use of multithreading on the server. Furthermore, we use asynchronous client requests to hide server response times behind computing times.

Additionally, as the number of compute nodes and the runtime of cluster jobs increase, fault tolerance becomes increasingly important due to the growing probability of errors. Typical MPI programs, however, do not offer fault tolerance, such that client failures or connection issues might result in a crash of the entire system. To address this issue, we have used MPI in a client-server model using asynchronous operations and timeouts to improve fault tolerance.

In some use cases, such as certain hadron physics applications, the cost function itself requires another level of distributed parallelization because its computation requires enormous amounts of CPU time or memory, which go beyond the resources available on single cluster nodes. Using the MPI-Consumer with Geneva, this is no longer a complex or tedious task. The MPI-Consumer provides access to pre-configured subgroups of client nodes, which can be used by domain experts to intuitively parallelize their cost function.

Extensive quantitative evaluation with up to 1000 nodes shows that the MPI-Consumer scales perfectly on HPC clusters and vastly improves Geneva's user experience for high-performance computing. The MPI-Consumer even outperforms some WAN consumers developed earlier for Geneva and, therefore, can be used as a model for the improvement of Geneva as a whole.

The MPI-Consumer has been integrated into the Geneva optimization library and is now available to users [1]. Also, independent of Geneva's parametric optimization functionality, the MPI-Consumer can be used as part of a generic networking library as a scalable implementation of a fault-tolerant client-server model for high-performance computing clusters. Geneva is currently used by scientists at GSI in Darmstadt for fundamental research in hadron physics on the Virgo HPC cluster.

(1) Berlich, R; Gabriel, S; Garcia, Geneva Source Code <https://github.com/gemfony/geneva>.

Primary author: WESSNER, Jonas (GSI Helmholtz Center for Heavy Ion Research)

Co-authors: Prof. LUTZ, Matthias F.M. (GSI Helmholtz Center for Heavy Ion Research); Dr SCHWARZ, Kilian (Hochschule Darmstadt University of Applied Sciences); Dr BERLICH, Rüdiger (Gemfony scientific)

Presenter: WESSNER, Jonas (GSI Helmholtz Center for Heavy Ion Research)

Session Classification: Converging Infrastructure Clouds, Virtualisation & HPC

Track Classification: Track 9: Converging High Performance Computing Infrastructures: Supercomputers, clouds, accelerators