A Perspective on the SPPEXA Collaboration from Japan

Takayuki Aoki

A national research project was running in Japan from 2010 to 2017 named “Development of System Software Technologies for Post-Peta Scheme High Performance Computing” (so called Post-Peta CREST) . It was supported by JST (Japan Science and Technology Agency), which is the Japanese counterpart to DFG (“Deutsche Forschungsgemeinschaft”). The Post-Peta CREST project was similar to the first funding phase of SPPEXA in the sense that it had a primarily national scope. Then, the Post-Peta CREST project opened up to international collaboration, and some projects were extended for two more years, where they formed collaborative research groups with SPPEXA phase-II projects. Projects with contributions from Japan are *ExaFSA*, *ExaStencils*, *EXAMAG*, *ESSEX-II*, *EXASOLVERS*, *AIMES*, and *MYX*, with more than 10 researchers in the second phase of SPPEXA. To highlight the success of the Japanese collaboration with SPPEXA, we have a brief look at two working groups.

ppOpen-HPC and ESSEX-II

As a part of Post-Peta CREST projects between 2011 and 2015, a group at the University of Tokyo developed ppOpen-HPC, which is an open source infrastructure for the development and execution of optimized and reliable simulation code on post-peta-scale (pp) parallel computers based on many-core architectures. The framework covers various types of procedures for scientific computations in various types of computational models, such as FEM, FDM, FVM, BEM, and DEM.

T. Aoki (✉)

Tokyo Institute of Technology, Tokyo, Japan

e-mail: taoki@gsic.titech.ac.jp

© The Author(s) 2020

H.-J. Bungartz et al. (eds.), *Software for Exascale Computing - SPPEXA 2016–2019*, Lecture Notes in Computational Science and Engineering 136, https://doi.org/10.1007/978-3-030-47956-5_3

Automatic tuning (AT) technology enables automatic generation of optimized libraries and applications under various types of environments. The most updated version of ppOpen-HPC was released as open source software, which is available at <https://github.com/Post-Peta-Crest/ppOpenHPC>.

In 2016, the team of ppOpen-HPC joined the SPPEXA phase-II project ESSEX-II including members from the University of Erlangen-Nuremberg, which is funded by JST-CREST and SPPEXA under Japan (JST)-Germany (DFG) collaboration until 2018. ESSEX-II developed pK-Open-HPC (extended version of ppOpen-HPC, a framework for exa-feasible applications), such as preconditioned iterative solvers for quantum science.

Sparse coefficient matrices derived from applications in quantum science have generally relatively very small diagonal components, and they are generally ill-conditioned. Therefore, it is difficult to apply preconditioned iterative methods developed in ppOpen-HPC directly to such applications. The ESSEX-II team developed a regularization method for robustness based on blocking and diagonal shifting, which provide efficient and robust convergence of ill-conditioned problems in quantum science. Preconditioning methods with the regularization method are implemented in GHOST/PHIST libraries for solving matrices, which integrates all linear solvers and related methods developed in ESSEX/ESSEX-II projects. Moreover, they proposed a new method for global parallel reordering, which provides robust and efficient convergence of parallel iterative solvers with ILU-based preconditioning for very ill-conditioned problems. The developed method kept iteration number constant in strong scaling cases up to $O(104)$ MPI processes for very ill-conditioned problems. This is the first method for global parallel reordering.

In the ESSEX-II project, CRAFT (A library for application-level Checkpoint/Restart and Automatic Fault Tolerance) has been developed for fault resilience on exascale systems by checkpointing. ESSEX-II integrated the dynamic load balancing function and CRAFT, and developed a prototype of a fault-resilient framework for parallel FEM applications. Parallel FEM codes can continue computations by this framework, when some of the computing nodes fail. This framework does not need spare nodes for fault resilience. This idea can be extended to various types of procedures for dynamic scheduling on exascale systems.

Collaborations in ESSEX-II project have been continuing in the JHPCN projects (“Numerical Library with High-Performance/Adaptive-Precision/High-Reliability” (starting in 2018), “[Innovative Multigrid Methods](#)” (starting in 2018)), and in “Innovative Methods for Scientific Computing in the Exascale Era by Integrations of (Simulation+Data+Learning)” funded by “Grant-in-Aid for Scientific Research (S) (KAKENHI S)” (2019-2023)

Xevolver and ExaFSA

The so-called Xevolver project is one of the Post-Peta CREST projects from 2011 to 2017. A group at the Tohoku University discussed how they could help in legacy code migration to future-generation extreme-scale computing systems that will be massively parallel and heterogeneous. Even today an HPC application code is likely optimized assuming a particular system configuration, and hence specialized only for its target system. In general, such an application is not performance-portable at all. As the HPC system architectures are now diverging and also getting more complicated in terms of accelerators, it will require more time and effort to migrate or re-optimize the code to another system in the future. To make matters worse, system-specific code optimizations are tightly interwoven with the computation and thereby degrade the code readability and maintainability, even though HPC applications need to evolve not only for achieving high performance, but also for advancing computational science. Therefore, in the project, our team has developed a code transformation framework, Xevolver, so that users can define their own code transformations and thus express system-specific code optimizations as code transformation rules. Since code transformation rules can be defined separately from application codes themselves, the Xevolver framework can contribute to separation of system-specific performance concerns from application codes, and hence prevent overcomplicating the codes.

In 2016, core members of the Xevolver research team joined the second phase of the ExaFSA project in order to demonstrate that the Xevolver approach is effective for optimizing real-world applications in practice. The Xevolver approach assumes that an HPC application is developed by a team work of at least two kinds of programmers. One is application developers and the other is performance engineers. Application developers are interested in simulation results rather than performance, while performance engineers are mainly focusing on sustained simulation performance. Therefore, Japanese researchers have worked as performance engineers using Xevolver by considering German research groups as application developers.

The ExaFSA project focused on engineering two solvers, FASTEST and Ateles, which have been developed in the ExaFSA project as primary building blocks of a practical coupled simulation. An incompressible flow solver, FASTEST, has a long history of development and was once optimized for classic vector machines. Thus, some of important kernels still have two versions, default version and its vector-optimized version. In the ExaFSA project, hence, they used the Xevolver framework to express the differences between the two versions, and demonstrated that the vector-optimized version can be generated by transforming the default version. That is, the Xevolver approach can express the system-specific code optimizations as code transformation rules, and thus even simplify the code while achieving high performance and portability. Ateles is based on Discontinuous Galerkin (DG) discretization method, and a part of the simulation framework, APES, was developed at the University of Siegen in Germany. Unlike FASTEST, Ateles is written using modern Fortran language features to hide the implementation

details. However, the kernel loops still need to be optimized in different ways for individual system architectures to achieve high performance. For example, some loop optimizations with compiler directives are mandatory for the NEC SX-ACE vector computing system to properly vectorize and thus efficiently execute the loops. In this project, Xevolver is used to apply the loop optimizations without major modifications of the original code. Accordingly, the ExaFSA project was a very good opportunity for us to demonstrate that the Xevolver approach can help an appropriate division of labor between application developers and performance engineers by achieving separation of concerns. This clarification of role-sharing will be very helpful for long-term application development especially in an upcoming extreme-scale computing era.

The Role of Japan in HPC Collaborations

The SPPEXA program was unique in the sense that it established sustainable connections in the field of HPC between France, Germany, and Japan. With the supercomputing infrastructure in Japan (and its upcoming flagship supercomputer [Fugaku](#)), the three countries are suitable partners for portability and methodology comparisons and, thus, synergistic research developments (such as within SPPEXA connection).

A new field of interest in Japan, Germany, and France is data science and its connection to HPC. SPPEXA participated in the tri-lateral workshop in Tokyo “[Convergence of HPC and Data Science for Future Extreme Scale Intelligent Applications](#)”, where we discussed new possible collaborations in the fields of HPC and Big Data. Looking back at SPPEXA, we see many success stories, and hope for a lot of continuing collaborations.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.