Predictive Modeling of Loop Tiling for Deep Memory Hierarchies in the Context of Performance Portability

Al. Levchenko

SPbPU Supercomputer Center, St. Petersburg, Russia

At the pre-exascale stage of supercomputing development, the advent of systems with petabytes of globally addressable memory is predicted [1]. Unsatisfactory spatial and temporal data locality will inevitably result in loss of performance portability for systems with deep memory hierarchies. Parameterized loop tiling, an important method to partition the iteration space, can be used for improved memory utilization in hardware caches as well as software-managed global access memories, thereby enabling performance portability [2]. A major research constraint is the high cost of searching the optimal parameters of tiling transformations in real HPC environment due to the lack of proper performance models, while a bunch of polyhedral infrastructures already exist at the near-production level.

The contributions of this paper include extensions to polyhedral-related models such as prediction formula for execution time of tiled loop and iteration schedule, model for determining the tile size, which is most tunable parameter [3], and architecture-specific model for DMA operations alongside revised cache-based cost model. The presented models require minimal information about the dependencies of the algorithm and can be used, in particular, to improve the existing algorithms of model-driven automatic tile size selection to find the best tile layout for a loop nest [4]. This paper focuses on using parameterized tiling to exploit the target system with globally addressable memory in two ways: (1) at the level of multi-machine macronode with 12Tb of RAM, and (2) by combining the memories of macronode set into a global space under hybrid MPI+PGAS model. It is argued that the resulting model can be used to target global shared memory macronode with ccNUMA architecture, which can be considered for portable tiled code generation as a prototype of basic node of target system with hybrid PGAS-based global memory. Current model features (Table 1) include tiling effect predictions for shared memory parallelism when applied to the most dominating kernels like Gauss-Seidel in the first instance, where the memory reference pattern cannot be statically determined.

Table 1. Model features considered piecewise or in whole

Submodels	Considerations
Communication-avoiding operations	Transfer time and latency
Execution time modeling	Intra- and inter-tile
Computations (Gauss-Seidel)	Cache-efficiency (tiled vs. traditional)
Performance impact of architecture	Considered for target ccNUMA

Future work will include research on how to produce tiled code for most important computational kernels to verify presented models empirically. Existing ways include development of pass pipeline extensions of polyhedral compilation frameworks (e.g., Polly system for LLVM) [5]. The main work should be done for HPC proxy applications like LULESH, which require irregular control flow. Optimal tiled code generation confirmed by the formally virefied model for surrogate kernels will mean achieving performance portability for real applications.

References

1. L. Eisymont. Hybrid strategy development of the supercomputer components // Open Systems. DBMS. 2017. Vol. 25. No 2. P. 8–11. URL: https://www.osp.ru/os/2017/02/13052216

2. T. Wijesinghe, K. Senevirathne, C. Siriwardhana, W. Visitha, S. Jayasena, T. Rusira, M. Hall. Parameterized Diamond Tiling for parallelizing stencil computations // Moratuwa Engineering Research Conference (MERCon). Moratuwa, Sri Lanka. May 29-31, 2017. P. 99–104. URL: http://dx.doi.org/10.1109/MERCon.2017.7980464

3. M. Takayanagi and T. Suzuki. Construction of Performance Model of Tile CAQR and Performance Result of the Implementation // 2017 IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC). September 18-20, 2017. Seoul, South Korea. P. 151–157. URL: http://dx.doi.org/10.1109/MCSoC.2017.18

4. A. Bhattacharyya, G. Kwasniewski, T. Hoefler. Using compiler techniques to improve automatic performance modeling // 2015 International Conference on Parallel Architecture and Compilation Techniques (PACT). San Francisco, CA, USA. October 18-21, 2015. P. 468–479. URL: http://dx.doi.org/10.1109/PACT.2015.39

5. L. Bagnères, O. Zinenko, S. Huot, C. Bastoul. Opening Polyhedral Compiler's Black Box // Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO). Barcelona, Spain. March 12-18, 2016. P. 128–138. URL: http://doi.acm.org/10.1145/2854038.2854048