Leveraging Stencil Computation Performance with Temporal Blocking using Large Cache Capacity on AMD EPYC™ 7003 Processors with AMD 3D V-Cache™ Technology

Long Qu, Hatem Ltaief, and David Keyes
Extreme Computing Research Center
King Abdullah University of Science and Technology
Thuwal, Saudi Arabia
{long.qu, hatem.ltaief, david.keyes}@kaust.edu.sa

***Abstract*—In structured grid finite-difference, finite-volume, and finite-element discretizations of partial differential equation conservation laws, regular stencil computations constitute the main core kernel in many temporally explicit approaches for such problems. For various blocking dimensions, the Spatial Blocking (SB) approach enables data reuse within multiple cache levels. Introduced in GIRIH, the Multi-core Wavefront Diamond blocking (MWD) method optimizes practically relevant stencil algorithms by combining the concepts of diamond tiling and multi-core aware wavefront temporal blocking, leading to significant increase in data reuse and locality. We evaluate the performance of MWD on a variety of recent multi-core architectures. Among all of them, the new AMD EPYC™ 7003 processors with AMD 3D V-Cache™ Technology, codenamed Milan-X, provide an unprecedented Last Level Cache (LLC) capacity. We show that the Milan-X hardware design is ideal for the MWD method, and can achieve significant performance gain over its predecessor Rome and still better than Milan, which has the same cores but less LLC. To our knowledge, this is the first time MWD performance is studied and reported on AMD Milan-X architecture.**

***Index Terms*—stencil computation, temporal blocking, multi-core wavefront diamond tiling, high performance computing.**

I. Introduction

In structured grid finite-difference, finite-volume, and finite-element discretizations of partial differential equation conservation laws, regular stencil computations constitute the main core kernel in many temporally explicit approaches for such problems. For various blocking dimensions, the Spatial Blocking (SB) approach enables data reuse within multiple cache levels. Moreover, SB-based stencil computations are easy to deploy on a variety of architectures and applications thanks to an efficient data layout. This makes SB-based approaches standing out for their code simplicity, flexibility, and sustainability. However, the straightforward generalization of SB to manycore architectures, with each core owning an exclusive share of cache, may leave performance on the table. We investigate the performance of Temporal Blocking (TB) using the Wavefront Diamond blocking (MWD) technique for stencil computations on recent x86 architectures. MWD decouples I/O operations from main memory in favor of streaming most of data from the Last Level Cache (LLC). We study MWD performance on AMD Milan-X architecture that provisions a much larger L3 cache than other existing x86 architectures. Initially deployed ahead of its time with architectures exhibiting low LLC capacity, we now demonstrate performance superiority of MWD on Milan-X against various architectures.

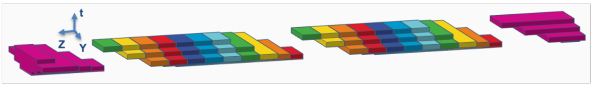

Fig. 1. A 3D view of multi-threaded computation in MWD blocking.

II. The design of MWD Blocking

The Temporal Blocking (TB) method improves data locality further by adding another level of blocking along the time dimension via a diamond tiling mechanism. The key idea of Multicore Wavefront Diamond blocking (MWD) method [2], [3], as implemented in GIRIH [1], is to reuse a freshly computed lattice within a thread group as many times as possible before evicting it from a cache shared by multiple threads. In particular, we employ different optimization strategies for each dimension. We use SIMD auto-vectorization for the Z dimension, which is the innermost dimension. We divided the diamond into separate pieces for thread parallelism in the middle dimension, or the Y dimension. We apply wavefront parallelism to increase in-cache reuse inside thread groups for the outermost dimension (i.e., the X dimension). Fig. 1 shows a 3D view of MWD blocking with multiple threads shown in different colors sharing the computation of a diamond.

By combining the concepts of diamond tiling and multi-core aware wavefront temporal blocking, MWD leads to significant increase in data reuse. The Last Level Cache (LLC) is shared among cores to reduce memory accesses across successive

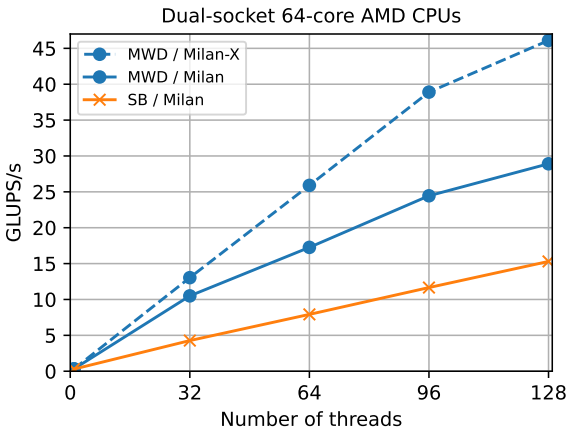

Fig. 2. Results of $1024 \times 1024 \times 512$ domain size with 1000 time steps.

iterations. A recent study [4] shows the impact of MWD on the performance of seismic applications and highlights its performance superiority versus SB.

III. MWD Blocking on AMD Milan-X Chip Architecture

The AMD Milan-X processors, which are based on the same Zen 3 cores as Milan, feature three times the L3 cache capacity. Each Complex Core Die (CCD) in Milan has 32MB of cache; Milan-X increases that to 96MB per CCD by stacking an additional 64MB of 3D cache. This results in a striking 768MB of L3 cache size with eight CCDs.

Such unprecedented capacity of the LLC fits well with the MWD design. A high number of iterations can be processed simultaneously inside the diamond before their evictions. Therefore, MWD enhances data reuse and locality across cores sharing the same LLC.

IV. Performance comparaison between Milan and Milan-X

We evaluate the performance of MWD and compare it to SB method on a variety of recent multi-core architectures including contemporary chips from Intel, NEC, Fujitsu and AMD. A plateau is observed for SB approach as the memory bandwidth is saturated, except for multiprocessors with HBM, i.e., NEC Aurora and Fujitsu A64FX, for which the performance of MWD-based stencil computation continues to scale. Among all of them, AMD multi-processors achieves the highest one-node performance for MWD, thanks to its largest on-chip LLC capacity. As shown in Fig. 2, on a dual-socket 64-core AMD Milan node, the MWD method reaches over 24 Gstencils per second, which is about $1.5X$ speedup over the classic SB method on the same system. By upgrading to AMD Milan-X with the same number of cores, we get an extra $1.5X$ speedup thanks to its L3 capacity enlargement. The Milan-X hardware design is ideal for the MWD method.

We further evaluate the MWD kernel performance using the roofline performance model analysis. As shown in Fig. 3

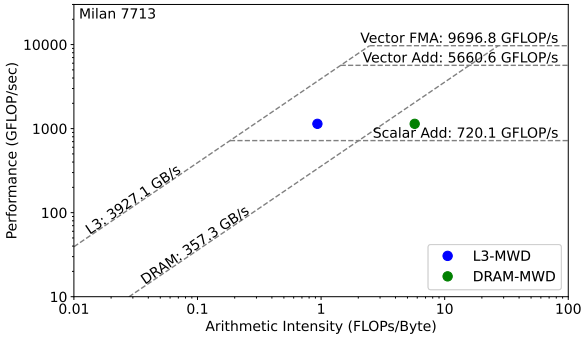

Fig. 3. Roofline Performance Model Analysis on AMD Milan 7713.

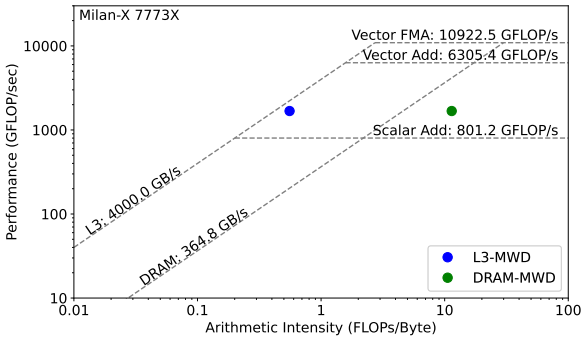

Fig. 4. Roofline Performance Model Analysis on AMD Milan 7753X.

and 4, the actual hardware specifications of AMD Milan and AMD Milan-X are quite similar, i.e., bandwidth, cores count and cores type, etc. The main difference is the larger L3 cache that further pushes away the MWD bandwidth obtained from L3 as opposed to main memory, which highlights how MWD decouples from main memory.

V. Summary and future work

We show the performance improvements of MWD over SB on large grid sizes using different multi-core architectures. MWD works best on hardware with a wide bandwidth gap between LLC and main memory. MWD achieves high performance in presence of large shared LLC and is agnostic to main memory technology (e.g., DDR and HBM). We plan to port MWD on AMD Instinct GPUs and to integrate it into the reverse time migration [4].

Acknowledgment

We would like to acknowledge Advanced Micro Devices, Inc. (AMD) for providing the remote access of the AMD Milan-X hardware.

References

[1] GIRIH can be found at https://github.com/ecrc/girih
[2] Malas T., Hager G., Ltaief H., Stengel H., Wellein G. and Keyes D. (2015) Multicore optimized wavefront diamond blocking for optimizing stencil updates. SIAM J. Sci. Comput. 37(4): 439–464.
[3] Malas T., Hager G., Ltaief H. and Keyes D. (2017) Multidimensional intratile parallelization for memory-starved stencil computations. ACM Trans. Par. Comput. 4(3): 12:1–12:32.

[4] Qu L., Abdelkhalek R., Said I., Ltaief H. and Keyes D. (2022) exploiting temporal data reuse and asynchrony in the reverse time migration, Int. J. High Perf. Comput. Appl., in press.