

# Redesigning The Absorbing Boundary Algorithm For Asynchronous High Performance Acoustic Wave Propagation

Rached Abdelkhalak

*Extreme Computing Research Center (ECRC)*  
*King Abdullah University of Science and Technology*  
Thuwal, Saudi Arabia  
rached.abdelkhalak@kaust.edu.sa

Vincent Etienne

*EXPEC Advanced Research Center*  
*Saudi Aramco*  
Dhahran, Saudi Arabia  
vincent.etienne@aramco.com

Kadir Akbudak

*Extreme Computing Research Center (ECRC)*  
*King Abdullah University of Science and Technology*  
Thuwal, Saudi Arabia  
kadir.akbudak@kaust.edu.sa

Thierry Tonellot

*EXPEC Advanced Research Center*  
*Saudi Aramco*  
Dhahran, Saudi Arabia  
thierrylaurent.tonellot@aramco.com

## I. INTRODUCTION

Extreme thread concurrency for parallel performance and data locality for traffic reduction associated with dynamic core frequencies, portend the end of the uniform bulk synchronous programming model. It is paramount to design new algorithms, programming models, and amenable software tools in order to relax the synchrony of existing algorithms and their implementations. Many standard means of expressing algorithms, including loops and function calls, introduce artifactual ordering and synchronization from which future software must be freed.

The industrial project codenamed **ExaWave** addresses these challenges for seismic modeling applications by introducing: Fine-grained computations to exploit the thread concurrency and maximize strong scalability on a single shared-memory node; asynchronous execution to cope with the dynamic core frequencies and bus bandwidth saturation while mitigating the load imbalance overhead; data motion reduction techniques for the stencil time integration. Beside the traditional spatial blocking (SB) technique for stencil computations [1], we integrate the Multicore-optimized Wavefront Diamond (MWD) tiling approach [2], [3], which takes into account the three aforementioned ingredients, into the **ExaWave** framework.

However, the fine-grained asynchronous handling of the Convolutional Perfectly Matched Layer (CPML) [4] for absorbing boundary conditions turns out to be a challenging open research problem, due to severe inherent data dependencies constraints, which impedes MWD performance impact.

We combine techniques of loop fusion to reduce memory traffic and sliding windows to cut down the engendered extra flops, in order to consolidate CPML integration with the overall asynchronous MWD technique.

## II. STENCIL COMPUTATIONS

### A. Spatial Blocking (SB)

Stencil computations require proceeding in time. A straightforward algorithm is to proceed in time one by one. In other words, synchronization is performed after each time step.

One mainly-used optimization while processing the spatial points is to process the points block-wise to increase locality. In this work, we utilize blocking in  $X$  and  $Y$  dimensions.  $Z$  dimension is the fastest index and left for vectorization.

### B. Multicore-optimized Wavefront Diamond (MWD) Tiling

The MWD approach [2], [3] contains blocking both in time and space as shown in Fig. 1a. In this figure, diamond-shaped blocks are utilized for blocking in time and  $Y$  in order to satisfy the dependencies across several time steps.

## III. IMPLEMENTATION OF CPML

In this work, we solve 2<sup>nd</sup> order formulation of 3D acoustic wave equation and use discretization with 8<sup>th</sup> order in space and 2<sup>nd</sup> order in time. We implement boundary conditions based on the CPML formulation by Komatitsch and Martin [4] in the MWD framework. Figure 1b shows the data dependencies while calculating boundary conditions for the points in the  $Z$  dimension. In this figure, a red point represents a boundary condition computation and the 8 points between two dashed lines are the values involved in this computation. As also seen in this figure, pressure  $P_z$  is computed depending on 8 values in  $A_z$  (since we are considering 8th order finite difference in space) and each value in  $A_z$  is computed according to pressure component stored in  $p$ .

The two-level dependency in the CPML implementation and the dependencies incurred by blocking in time as mentioned in Section II-B necessitate novel algorithms to calculate boundary

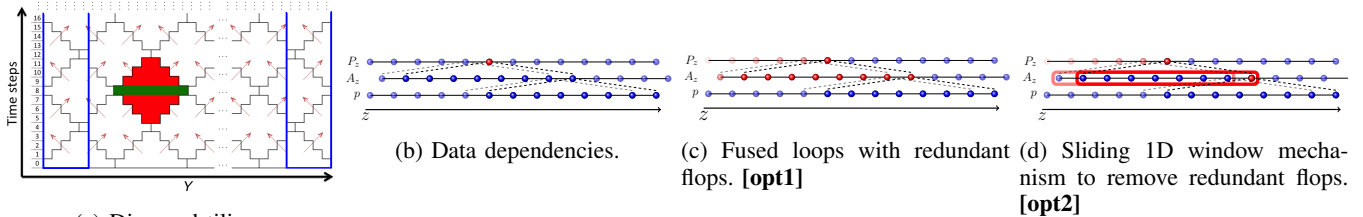
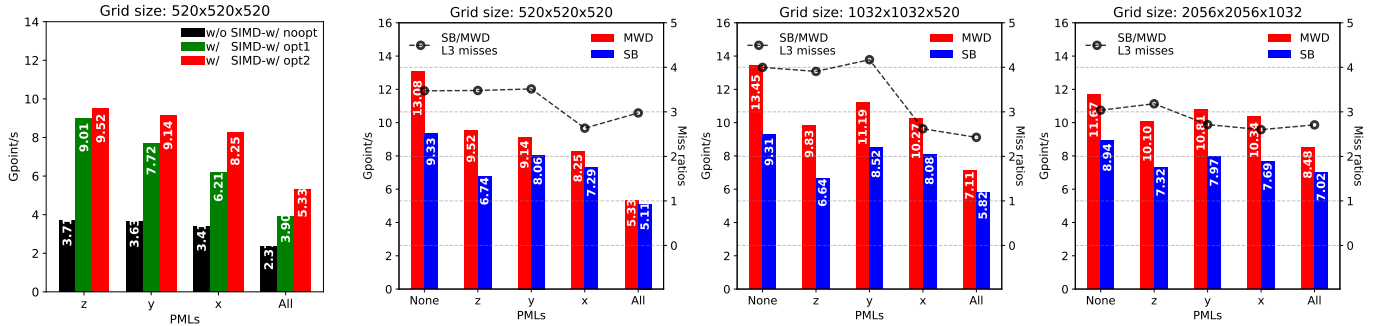


Fig. 1: Data dependencies in diamond tiling and CPML implementation.



(a) Performance effect of loop fusion and sliding window techniques in MWD. (b) Performance comparison of MWD and SB for three different grid sizes and different configurations of PMLs.

Fig. 2: Experimental results.

conditions in an efficient way. In this work, we propose fusing the two-level boundary condition computations inside one level of on-demand computation. This fusion incurs doing redundant flops due to on-demand scheme as shown in Fig. 1c. For the red point in  $P_z$ , 8 red points in  $A_z$  are calculated depending on values in  $p$ . Hence the calculations of  $A_z$  are mostly redundant. This scheme is referred to as **opt1**.

We propose to use a sliding array mechanism for caching recently calculated values of  $A_z$  in order to remove the redundant computations of these values. Fig. 1d shows the sliding windows for previous and current points of  $P_z$ . As seen in the figure, only one point of  $A_z$  is calculated and other seven elements cached in the sliding window are reused. This scheme is referred to as **opt2**. For  $X$  and  $Y$  dimensions, a two-dimensional array, i.e., a sliding surface, is used instead of a one-dimensional (1D) array since vectorization along the fastest  $Z$  index is not possible via using a 1D array.

#### IV. RESULTS

We report the number of Giga point updates per second (Gpoint/s) performance of MWD and SB on a two-socket system with Intel Skylake processors Platinum 8176 (2.10 GHz, 38.5 MB L3 cache, 28 cores per chip, AVX512). The sustained bandwidth reported by STREAM benchmark is 182 GB/s. The system has 192 GB RAM and the codes are compiled with icc 18.0.1.

We use three different grid sizes:  $(520 \times 520 \times 520)$ ,  $(1032 \times 1032 \times 520)$ , and  $(2056 \times 2056 \times 1032)$  in the experiments. The width of CPML is set to 10. The percentages of all boundary points in 6 layers are 11.1%, 7.5%, and 3.8% for the above-mentioned grid sizes, respectively.

Fig. 2a shows the performance improvement obtained via loop fusion [**opt1**] and sliding windows [**opt2**] for MWD. The

baseline method is selected as PML implementation without vectorization (SIMD instructions) since vectorization cannot be enabled for all layers of PMLs without the proposed optimizations as mentioned in Section III. As seen in the figure, using both optimizations results approximately 3x speedup when all PMLs are enabled.

Fig. 2b shows the performance of MWD against SB in terms of Gpoint/s as bars and the ratio of L3 cache misses of SB to those of MWD as a curve. As seen in the figure, MWD achieves more than 20% speedup for middle and large size grids. As also seen in the figure, SB incurs up to four times L3 cache misses than MWD.

#### V. CONCLUSION

We propose loop fusion and sliding window techniques for implementing CPML boundary conditions in the MWD framework. Preserving 20% to 50% performance improvement over SB (for middle and large size grids) even if PMLs are enabled shows the success of the proposed techniques. As future work, we plan to investigate other formulations for boundary conditions which will require less memory bandwidth and computation in order to eliminate the performance degradation in smaller grid sizes.

#### REFERENCES

- [1] V. Etienne, T. Tonellot, P. Thierry, V. Berthoumieux, and C. Andreolli. Optimization of the Seismic Modeling with the FDTD Method. *SEG Annual Meeting, Expanded Abstracts*, pages 3536–3540, 2014.
- [2] T. Malas, G. Hager, H. Ltaief, H. Stengel, G. Wellein, and D. Keyes. Multicore Optimized Wavefront Diamond Blocking for Optimizing Stencil Updates. *SIAM Journal on Scientific Computing*, 37(4):439–464, 2015.
- [3] T. Malas, G. Hager, H. Ltaief, and D. E. Keyes. Multidimensional Intratile Parallelization for Memory-Starved Stencil Computations. *ACM Trans. Parallel Comput.*, 4(3):12:1–12:32, December 2017.
- [4] D. Komatitsch and R. Martin. An Unsplit Convolutional Perfectly Matched Layer Improved at Grazing Incidence for the Seismic Wave Equation. *GEOPHYSICS*, 72(5):SM155–SM167, 2007.