Accepted manuscript of: M. Jakubowski and P. Sypek, "Electromagnetic Simulations with 3D FEM and Intel Optane Persistent Memory," 2022 24th International Microwave and Radar Conference (MIKON), Gdansk, Poland, 2022, pp. 1-5, doi: 10.23919/MIKON54314.2022.9924749.

Electromagnetic Simulations with 3D FEM and Intel Optane Persistent Memory

Maciej Jakubowski*, Piotr Sypek†
Faculty of Electronics, Telecommunications and Informatics
Gdansk University of Technology
Gdansk, Poland
*maciej.jakubowski@pm.me, †piotr.sypek@pg.edu.pl

***Abstract*—Intel Optane persistent memory has the potential to induce a change in how high-performance calculations requiring a large system memory capacity are conducted. This article presents what this change may look like in the case of factorization of large sparse matrices describing electromagnetic problems arising in the 3D FEM analysis of passive high-frequency components. In numerical tests, the Intel oneAPI MKL PARDISO was used to solve relatively large electromagnetic problems defined using the finite element method.**

***Index Terms*—Intel Optane PMem, finite element method, sparse matrix factorization, Intel PARDISO**

I. Introduction

The frequency-domain finite element method is a very popular and efficient tool for analyzing electromagnetic problems [1]. This type of analysis involves the computationally intensive solution of a linear matrix equation ($\mathbf{A}x = b$). In the basic analysis, the number of solutions to this equation needed to evaluate the transfer function is equal to the number of discretization points of the sweep in the frequency domain. This number may be lower if the model order reduction method is used [2]–[4]. However, in both scenarios, the solution of the matrix equation must be determined multiple times, which is the most time-consuming part of the electromagnetic simulation. Additionally, when the simulated structure is electrically large or high simulation accuracy (leading to high mesh density) is required, the size of the matrix $\mathbf{A}$ becomes very large, and the resources needed in the solution phase grow.

Usually, the solution of large and sparse systems of linear equations can be determined using iterative methods, which are dedicated to solving this category of problems. It is known, however, that achieving a satisfactory convergence of iterative methods when applied to the solution of electromagnetic problems is challenging [5], [6]. Additionally, if model order reduction is applied, the solution phase is time-consuming, as each frequency point and right hand side vector is processed sequentially. In a situation where determining a large number of solutions to this type of matrix equation is required, it may be more effective to use direct methods based on matrix factorization [3]. However, factorization of a large sparse matrix requires a huge amount of memory. As a result, the amount of memory required to factorize a large sparse matrix severely limits the size of the problem that can be solved on a single workstation.

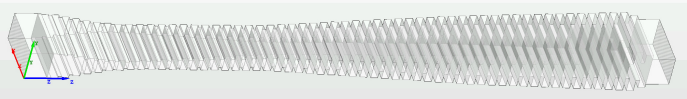

Fig. 1. Visualization of analyzed high-power waveguide low-pass filter [10]

The memory requirement for sparse matrix factorization strongly depends not only on the sparsity pattern and number of non–zero elements but also on the quality of its implementation. One of the most efficient and publicly available implementations of sparse matrix factorization is the Intel oneAPI MKL PARDISO [7]. This software package also partially alleviates memory resource constraints by allowing some data to be stored on a hard drive or SSD while factorizing a large sparse matrix (out-of-core Intel PARDISO). This technique allows for factorization of larger matrices, but the computational performance is much lower compared to the standard in-core Intel PARDISO.

For many years, there were only two options for extending the memory resources required for computation: either with slow, cheap disks or with fast, expensive DRAM. Recently, an alternative to these solutions, i.e. Intel Optane PMem, has appeared on the market. Intel Optane PMem is a block of memory connected to the processor via a DIMM socket, which enables fast data transfers and short access times between the processor and PMem, and can be used for efficient computation [8], [9]. For this reason, it is possible to use PMem to build efficient workstations or servers with high-capacity system memory. The article presents an evaluation of the computational efficiency of such a system during the factorization of sparse matrices representing electromagnetic problems. In electromagnetic simulators, such calculations are the core of their implementation.

II. CSF Benchmark

In this paper, to analyze the performance of sparse matrix factorization for different sizes of electromagnetic problems and with different hardware configurations, the results of tests performed on a single example of an electromagnetic simulation are shown. For the example, the analysis of the electromagnetic properties of the compact high-power low-pass filter shown in Fig. 1 was selected. The filter was designed

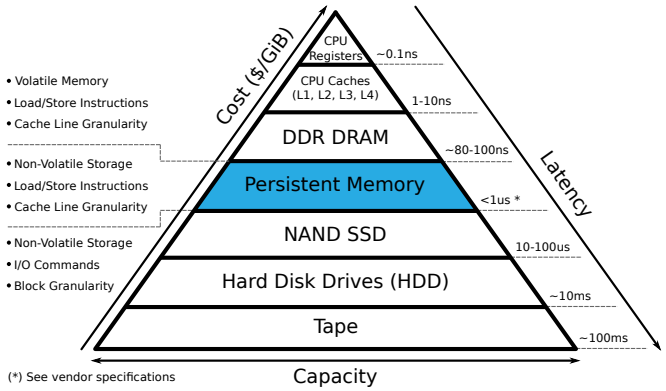

Fig. 2. Memory/storage hierarchy pyramid with estimated latencies [12]

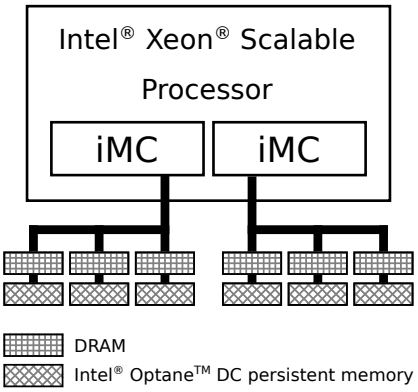

Fig. 3. 2–2–2 topology with six DDR4 DIMMs and six PMMs [16]

for Ka-band satellite applications and is described in detail in [11]. The matrices arising in the simulation of this structure were generated using InventSim [10], which is a commercial electromagnetic field simulator based on 3D FEM. These matrices were then used to construct a linear problem that was then solved with the Intel PARDISO direct solver. The Intel PARDISO performance measurement for solving the problems described in Tab. I is referred to as the CSF (Custom Sparse Factorization) benchmark in this article.

III. Intel Optane PMem

Intel Optane persistent memory (PMem) bridges the gap between DRAM and block storage, as shown graphically in the memory hierarchy pyramid in Fig. 2, [12]. Its high performance was confirmed for advanced applications, e.g. in [8], [9], [13]. Additionally, PMem is not only a fast but also a cost-effective solution for building a system with high memory capacity, and "typical Optane-based systems provide 25% more memory for 20–30% cost reduction" [14]. However, Intel Optane PMem performance is not equal to DRAM performance, e.g. read/write latencies can be four times higher than in the case of DRAM [15].

In the next part of this article, the performance of a system with a large amount of non-volatile memory will be tested with the CSF benchmark and Intel Optane PMem operating in Memory Mode. In Memory Mode, the Intel Optane PMem is configured as system memory and the DRAM as cache. Intel Optane PMem also can operate in App Direct Mode, where the DRAM and PMem are configured as separate pools of memory, but this mode will not be tested in this article.

The topology of connecting the DRAM and PMem to the processor can be flexibly formed [16]. For example, the 2–2–2 topology is shown in Fig. 3. In the 2–2–2 notation, each number represents a single memory channel and its value represents the number of modules connected to that channel, for example, a connection to the same memory channel of one DRAM module and one Intel Optane PMem module is denoted by 2.

In this article, the performance of sparse matrix factorization was measured for three test variants:

- DRAM-only, where the Intel Optane PMem and SSD are not used in the calculations,
- MM, where the Intel Optane PMem was set to Memory Mode and acted as system memory,
- OOC, where the SSD was used to store temporary data during out-of-core Intel PARDISO calculations.

The details of these test configurations are as follows:

A. DRAM-only test

In this test variant, the system memory consisted only of DRAM. This benchmark is performed to define a reference system performance level when the Intel Optane PMem and SSD are not used in the calculations.

B. MM test

In this variant of the test, the Persistent Memory Modules (PMMs) were configured in Memory Mode and constituted system memory, with the DRAM acting as the last-level cache (LLC). This memory configuration does not require changes to the operating system or the applications being executed. This important feature allowed us to measure the performance of Intel PARDISO operating in in-core mode on data stored in system memory comprised of PMMs.

C. OOC test

In this test variant, if the memory requirement of in-core Intel PARDISO exceeds the size of the DRAM, out-of-core Intel PARDISO is used for sparse matrix factorization. For smaller problems, the computation is performed using in-core Intel PARDISO which stores all data in DRAM.

During this test, temporary data generated by out-of-core Intel PARDISO was stored on the SSD. Out-of-core Intel PARDISO "can solve very large problems by holding the matrix factors in files on the disk, which requires a reduced amount of main memory compared to IC" (in-core Intel PARDISO) [7]. It is important to note that not all data managed by out-of-core Intel PARDISO can be transferred to the disk, so the DRAM capacity can still limit the maximum size of the matrix for which it is possible to calculate the sparse matrix factorization.

TABLE I
MAIN PROPERTIES OF SPARSE MATRICES

Test number	Number of rows	Number of non-zeros	Non-zeros per row
1	100,254	1,541,325	15.37
2	150,082	2,407,039	16.04
3	230,204	3,515,046	15.27
4	350,060	5,631,346	16.06
5	562,704	10,050,956	17.86
6	920,980	16,513,754	17.93
7	1,515,634	29,233,635	19.29
8	2,490,592	51,014,880	20.48
9	3,299,680	66,899,280	20.27
10	4,105,922	82,954,035	20.20
11	5,547,038	111,990,629	20.19
12	7,346,496	144,255,036	19.64
13	11,219,216	239,266,296	21.33
14	14,001,618	295,678,299	21.12
15	18,257,014	392,264,345	21.49
16	21,946,996	470,771,798	21.45
17	26,768,864	568,199,704	21.23
18	32,353,620	693,404,462	21.43
19	42,918,578	909,810,911	21.20
20	50,292,162	1,040,894,587	20.69
21	91,789,118	2,010,041,081	21.90

TABLE II
SERVER CONFIGURATION

DRAM capacity	384 GB
Intel Optane PMem capacity	1536 GB
DRAM	12x Samsung 32GB DDR4 2Rx4
PMem	12x 128GB Intel Optane PMem 100 Series
CPUs	2x Intel Xeon Platinum 8280L
Hyper-Threading	Enabled
Number of CPU cores	112
Motherboard	Lenovo SR650
OS	Debian 11.3
Linux kernel version	5.17.8
Intel oneMKL version	2022.1.0
Compiler	icpc 2021.6.0 20220226

IV. IBM CLOUD

Numerical tests were carried out on an IBM Cloud Bare Metal server, made available in the Dallas 13 location. The server details are listed in Tab. II. The server memory modules were connected in the recommended 2–2–2 topology, which achieves the highest capacity and bandwidth of the system memory [16]. As a result, the described server had 12 DRAM and 12 PMem modules (two processors times two memory controllers per processor times three memory channels per memory controller). The memory topology of this server is presented in Fig. 3. During the MM tests, the entire PMem capacity was configured in the Memory Mode. The disk space used for the out-of-core Intel PARDISO calculations was built from Micron 5200 MAX TCG-E 1.92 TB SSDs in a RAID 10 configuration. Its capacity was 5363 GB.

TABLE III
TOTAL PEAK MEMORY CONSUMPTION AND RELATIVE COMPUTATION TIME

Test number	DRAM-only test [MB]	OOC test [MB]	MM test [MB]	$\frac{t_{\mathrm{OOC}}}{t_{\mathrm{MM}}}$
1	381	381	381	0.95
2	648	648	647	0.97
3	905	905	905	1.00
4	1,662	1,662	1,661	1.00
5	3,016	3,016	3,022	0.92
6	5,125	5,125	5,090	0.97
7	7,690	7,690	7,697	0.93
8	18,174	18,174	18,082	0.87
9	24,734	24,734	24,653	0.94
10	28,859	28,859	28,800	0.93
11	44,794	44,794	44,688	0.96
12	57,836	57,836	57,891	0.95
13	102,433	102,433	102,628	0.98
14	135,179	135,179	135,222	0.94
15	183,456	183,456	183,451	0.92
16	253,527	253,527	252,267	1.09
17	316,687	316,687	316,772	1.22
18	OOM	358,653	412,317	3.35
19	OOM	358,735	582,275	3.61
20	OOM	358,793	662,122	3.78
21	OOM	OOM	1,442,983	–

V. NUMERICAL RESULTS

The numerical tests were performed using the CSF benchmark for the set of electromagnetic problems listed in Tab. I. All sparse matrices used in the tests were real and symmetric, and were generated using the same InventSim project where only the mesh properties were changed for subsequent problems to generate matrices of different sizes. The sparsity of the test matrices (the number of non-zero elements divided by the number of rows) increased from 15 to 22 with their size.

The total peak memory consumed by Intel PARDISO during factorization is presented in Tab. III. The differences in memory consumption for the 1–17 tests performed with different memory configurations may be due to different solutions to the graph partitioning problem, which is NP-hard, and whose approximate solution is obtained using heuristics [17]. Factorization for 18–21 problems could not be computed for the DRAM-only tests as more system memory than the size of the DRAM was required. In these cases, an "out of memory" system error occurred, denoted as OOM in this article.

In the case of the OOC tests, the solutions were determined in the case of problems from 1 to 20. However, during the OOC test for problem 21, there was an OOM error. This is due to a feature of Intel's PARDISO OOC implementation that only allows some internal data to be allocated to a block device. This is a significant limitation of the out-of-core computation used to determine the problem solution. The size of the DRAM memory used to solve problems 18–20 in the OOC tests resulted from the memory limit specified by the value of the environment variable MKL_PARDISO_OOC_MAX_CORE_SIZE.

Calculations with PMem in Memory Mode are not limited

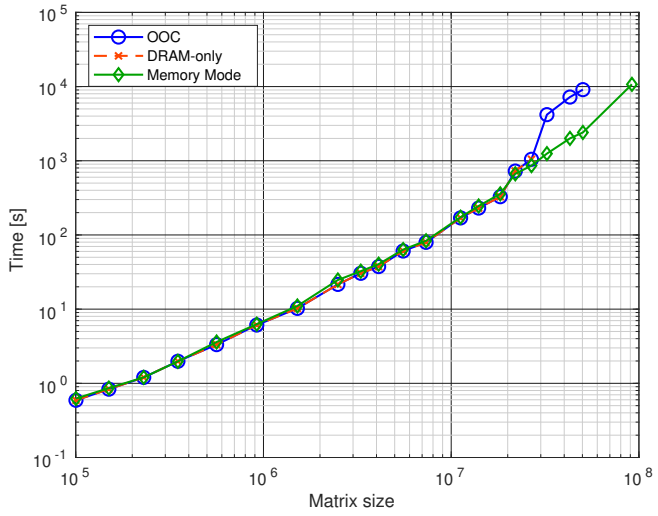

Fig. 4. Time of Intel PARDISO factorization

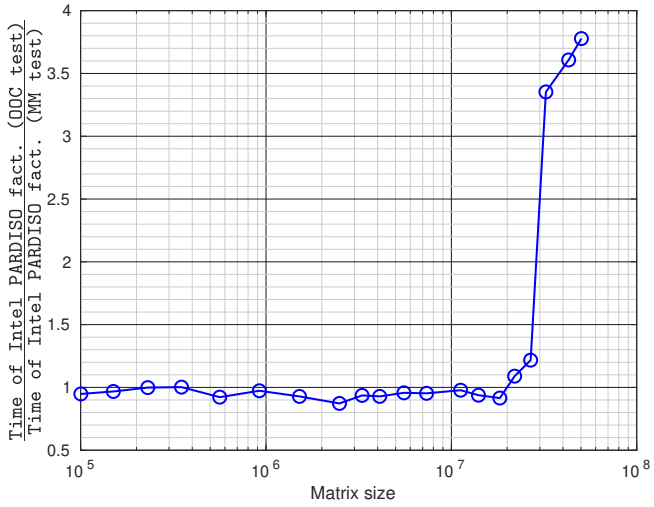

Fig. 6. Comparison of the MM and OOC test results

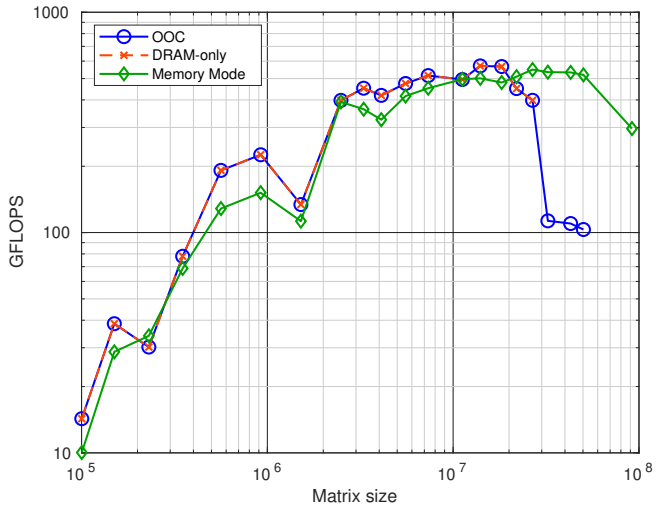

Fig. 5. Intel PARDISO numerical factorization performance

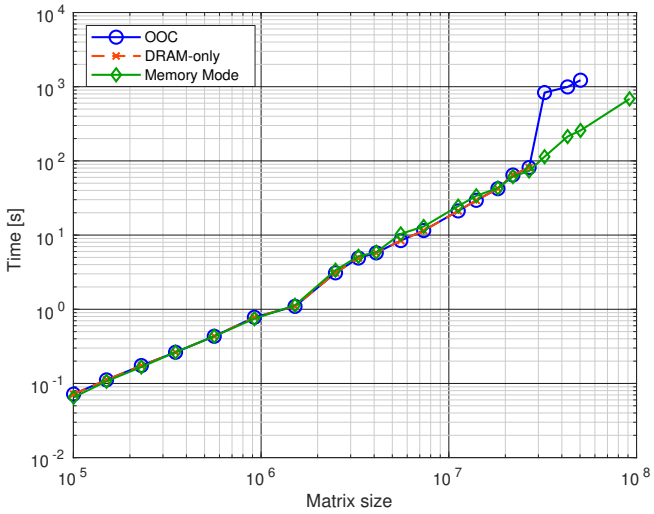

Fig. 7. Execution time of solving the matrix equation with two RHS vectors

by the size of DRAM memory, as is the case in the DRAM-only tests and indirectly in the OOC tests. In the tests with the Intel Optane PMem, it was possible to determine the factorization of the sparse matrices even when 1.5 TB of system memory was required.

In the DRAM-only test, the sparse matrix numerical factorization time increased from 591 ms for the smallest problem (100,254) to 1047 s for the largest problem (26,768,864). This means that the computation time increased by 1771 times while the size of the problem increased by 267 times. A graphical presentation of the factorization time increase is shown in Fig. 4. The increase in calculation time as the size of the problem increases is reduced due to the better parallel performance of the Intel PARDISO for larger problems, as shown in Fig. 5. The significant increase in the computing performance of Intel PARDISO along with the increase in the problem size is probably due to the fact that a sufficiently large amount of data is required for the effective parallelization of calculations in this type of algorithm.

The Intel PARDISO factorization time presented in Fig. 4 is similar for the DRAM-only and MM tests. The similarity of the measured times is more accurately presented in the last column of Tab. III, which contains the result of dividing the factorization time measured for the OOC test (t_{OOC}) by the factorization time measured for the MM test (t_{MM}). The similar level of performance seen for both cases is a very positive outcome as it means that significantly increasing the system memory size by adding Intel Optane PMem to the server has a very limited effect on increasing the computation time. On the other hand, for the OOC test performed for larger problems, the computation time was significantly longer in comparison to the computation time for the MM tests. This is largely due to the much lower out-of-core numerical factorization performance which was around 100 GFLOPS for the OOC test for problems 18, 19 and 20, which is significantly lower than the more than 500 GFLOPS measured for the

DRAM-only and MM tests for larger problems. These results show that the way the data is accessed can have a large impact on the computing performance of Intel PARDISO.

In recent decades, adding a memory source other than DRAM (e.g. disk memory) to calculations has often required the design and implementation of extensive add-on solutions. These solutions also sometimes limited the functionality of the program, as in the case of Intel PARDISO operating in out-of-core mode (DRAM memory size can still limit the ability to perform calculations). With the introduction of Intel Optane PMem, this approach is firstly no longer required, and secondly, out-of-core computing performance may be significantly lower compared to computing performance with Intel Optane PMem. This advantage is clearly visible in Fig. 6 which shows the computation time for the OOC test compared to the computation time for the MM test. For problems 18, 19 and 20, the sparse matrix factorization time for the OOC tests was more than three times larger than for the MM test. This means that with the introduction of Intel Optane PMem, out-of-core computing may become even less popular for algorithms that require a significant amount of memory for computation.

In the next step, the time to solve the matrix equation with Intel PARDISO was measured, assuming that the LU factorization had been calculated earlier. For this series of tests, the right hand side (RHS) of the matrix equation contained two columns corresponding to the two ports of the analyzed microwave filter shown in Fig. 1. The time needed to solve the matrix equation using Intel PARDISO was also similar for the DRAM-only and MM tests (Fig. 7). These results confirm that PMem is an efficient high-capacity memory source while the sparse matrix factorization is computed.

VI. Conclusions

In this article, the high computing performance of a microwave filter simulation carried out using Intel Optane PMem was presented. The time of calculations made on a server equipped with Intel Optane PMem configured in memory mode was similar to the computing time performed on the system in which the system memory was comprised only of DRAM modules. Such high performance was achieved for a system that had four times the memory capacity (384 GB compared to 1536 GB), which is very beneficial for the efficient simulation of large electromagnetic problems by the finite element method. Therefore, it can be concluded that Intel Optane PMem allows for efficient execution of one of the more time-consuming and numerically complex stages of electromagnetic simulation based on the finite element method. Of course, more tests need to be carried out to confirm the high efficiency of calculations using Intel Optane PMem for all parts of the implementation of electromagnetic simulators, but the presented results bode well in this respect.

The performance measured for calculations run on a system containing Intel Optane PMem compared to the performance of out-of-core Intel PARDISO was significantly higher. This result shows that the old approach of using block storage as a source of additional large memory capacity may become obsolete in the future, as a simpler solution with Intel Optane PMem allows for much higher computing performance without additional software modification costs. Most likely, this trend will be more visible when Intel Optane technology is more mature, and therefore its price is lower.

References

[1] J.-M. Jin, *The Finite Element Method in Electromagnetics*. John Wiley & Sons, 2015.

[2] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of Multiport Systems Using a Fast Implementation of the Vector Fitting Method," *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 6, pp. 383–385, 2008.

[3] A. Lamecki, L. Balewski, and M. Mrozowski, "An Efficient Framework for Fast Computer Aided Design of Microwave Circuits Based on the Higher-Order 3D Finite-Element Method," *Radioengineering*, vol. 23, no. 4, pp. 970–978, 2014.

[4] G. Fotyga, "A Model-Order Reduction Approach for Electromagnetic Problems With Nonaffine Frequency Dependence," *IEEE Access*, vol. 9, pp. 47 481–47 490, 2021.

[5] Y. Zhu and A. C. Cangellaris, *Multigrid Finite Element Methods for Electromagnetic Field Modeling*. John Wiley & Sons, 2006.

[6] A. Dziekonski, A. Lamecki, and M. Mrozowski, "GPU Acceleration of Multilevel Solvers for Analysis of Microwave Components With Finite Element Method," *IEEE Microwave and Wireless Components Letters*, vol. 21, no. 1, pp. 1–3, 2010.

[7] *Intel oneAPI Math Kernel Library. Developer Reference. 2022.0*. [Online]. Available: https://www.intel.com/content/www/us/en/develop/documentation/onemkl-developer-reference-c/top.html

[8] J. Izraelevitz, J. Yang, L. Zhang, J. Kim, X. Liu, A. Memaripour, Y. J. Soh, Z. Wang, Y. Xu, S. R. Dulloor *et al.*, "Basic Performance Measurements of the Intel Optane DC Persistent Memory Module," *arXiv preprint arXiv:1903.05714*, 2019.

[9] M. Weiland, H. Brunst, T. Quintino, N. Johnson, O. Iffrig, S. Smart, C. Herold, A. Bonanni, A. Jackson, and M. Parsons, "An Early Evaluation of Intel's Optane DC Persistent Memory Module and Its Impact on High-Performance Scientific Applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019, pp. 1–19.

[10] InventSim 3D FEM Electromagnetic Field Solver. [Online]. Available: http://inventsim.com

[11] F. Teberio, I. Arregui, A. Gomez-Torrent, E. Menargues, I. Arnedo, M. Chudzik, M. Zedler, F.-J. Görtz, R. Jost, T. Lopetegi *et al.*, "High-Power Waveguide Low-Pass Filter with All-Higher-Order Mode Suppression Over a Wide-Band for Ka-band Satellite Applications," *IEEE Microwave and Wireless Components Letters*, vol. 25, no. 8, pp. 511–513, 2015.

[12] S. Scargall, *Programming Persistent Memory: a Comprehensive Guide for Developers*. Springer Nature, 2020.

[13] O. Patil, L. Ionkov, J. Lee, F. Mueller, and M. Lang, "Performance Characterization of a DRAM-NVM Hybrid Memory Architecture for HPC Applications Using Intel Optane DC Persistent Memory Modules," in *Proceedings of the International Symposium on Memory Systems*, 2019, pp. 288–303.

[14] S. Ghosh, N. R. Tallent, M. Minutoli, M. Halappanavar, R. Peri, and A. Kalyanaraman, "Single-Node Partitioned-Memory for Huge Graph Analytics: Cost and Performance Trade-Offs," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14.

[15] T. Hirofuchi and R. Takano, "The Preliminary Evaluation of a Hypervisor-based Virtualization Mechanism for Intel Optane DC Persistent Memory Module," *arXiv preprint arXiv:1907.12014*, 2019.

[16] "Intel Optane DC Persistent Memory. Quick Start Guide. Revision 1.1," June 2020. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/memory-and-storage/data-center-persistent-mem/Intel-Optane-DC-Persistent-Memory-Quick-Start-Guide.pdf

[17] B. Hendrickson and T. G. Kolda, "Graph partitioning models for parallel computing," *Parallel Computing*, vol. 26, no. 12, pp. 1519–1534, 2000.