FFT, FMM, and Multigrid on the Road to Exascale Performance Challenges and Opportunities

Extended Abstract

Huda Ibeid
University of Illinois at Urbana-Champaign
Urbana, Illinois
hibeid@illinois.edu

Luke Olson
University of Illinois at Urbana-Champaign
Urbana, Illinois
lukeo@illinois.edu

William Gropp
University of Illinois at Urbana-Champaign
Urbana, Illinois
wgropp@illinois.edu

ABSTRACT

FFT, FMM, and multigrid methods are widely used fast and highly scalable solvers for elliptic PDEs. However, emerging systems are introducing challenges in comparison to current petascale computers. The International Exascale Software Project Roadmap [6] identifies several constraints on the design of exascale software. Challenges include massive concurrency, energy efficiency, resilience management, exploiting the high performance of heterogeneous systems, and utilizing the deeper and more complex memory hierarchy expected at exascale. In this study, we perform model-based comparison of the FFT, FMM, and multigrid methods in the context of these constrains and use the performance models to offer predictions about the methods performance on possible exascale system configurations, based on current technology trends.

1 CONCURRENCY

To gain some insight into the solvers performance on the massively concurrent systems expected at exascale, we derive analytical performance models that include computation and both intra- and inter-node communication costs. The intra-node communication along with the computation cost account for the single node performance which is a critical building-block in scalable parallel programs, whereas the inter-node term reflects the impact of network communication on the scalability.

Assuming arithmetic and memory operations can be overlapped, the total execution time T_{exe} is given by

$$T_{exe} \approx \max(T_{comp}, T_{mem}), \tag{1}$$

where T_{comp} is the computation time and T_{mem} is the time spent transferring data in a two-level memory hierarchy between the main memory and a cache with size Z and cache-line length L in elements. The inter-node communication cost is modeled using the (α, β) model where a message send cost can be represented as

$$T_{net} = m\alpha + n\beta, \tag{2}$$

where α represents communication latency, β is the send time per element over the network, m is the number of messages sent, and n is the total message size

The Roofline model can be used to assess the quality of attained floating-point performance (GFLOP/s) by combining machine peak performance, machine bandwidth, and arithmetic intensity. Figure 1 shows a roofline model of an AMD Istanbul processor along with the arithmetic intensity of various phases of FFT, FMM, and multigrid.

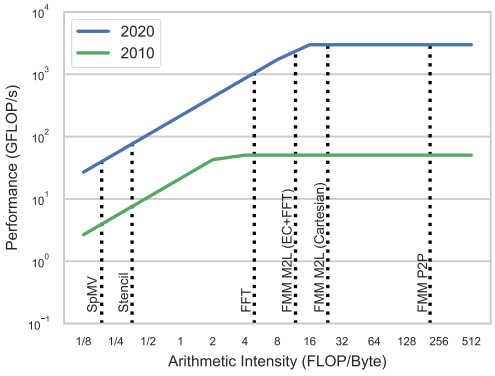


Figure 1: Roofline model of AMD Istanbul and computation intensity of the FFT, FMM, and multigrid methods.

Comparing the three methods shows that the FMM computations have very high arithmetic intensity due to its matrix-free nature. On the other hand, SpMV and stencil operations, which are the basic building blocks of the classic algebraic and geometric multigrid methods, respectively, have very low arithmetic intensities. The 3-D FFT has an intermediate arithmetic intensity that grows slowly with the problem size. Figure 1 also shows the Roofline model of a possible future CPU-based exascale system. The machine characteristics of the exascale system are based on extrapolating historical technology trends [10]. We observe that although FMM is compute-bound on contemporary systems, FMM could become memory-bound at exascale.

2 ENERGY

Power is a major hurdle on the road to exascale. Achieving an exaFLOP will be limited to no more than 20 MW of power when an exascale system could use as much as 100 MW if existing petascale design methodologies are used. This imposes design constraints on both the hardware and software to improve the overall efficiency.

To compare power and energy efficiency of the FFT, FMM, and multigrid methods, we use the energy Roofline model introduced in [4]. The energy Roofline model defines energy cost (Joules) as

$$E \equiv E_{flop} + E_{mem} + E_0(T_{exe}), \tag{3}$$

where E_{flop} is the total energy of computation, E_{mem} is the total energy of memory operations, and E_0 is the total constant energy.

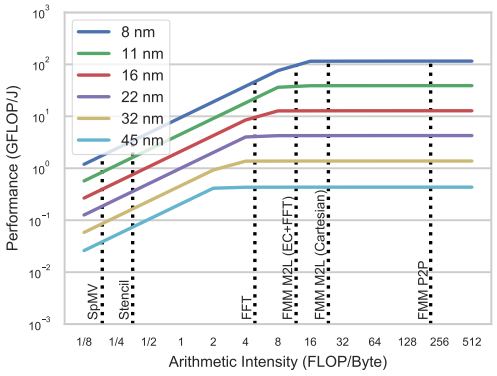
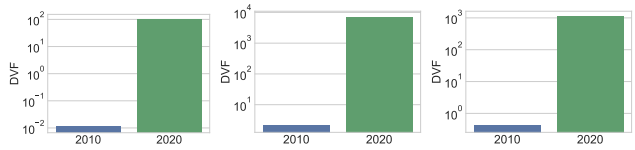


Figure 2: Energy Roofline model of various technology nodes.



(a) FFT **(b) FMM** **(c) Multigrid**

Figure 3: Data vulnerability factors.

To estimate the energy efficiency on future multicore chips, we use the transistor scaling projection model presented in [7]. Figure 2 shows how the energy efficiency is predicted to improve over time as transistor sizes decrease. Nevertheless, the per-transistor energy efficiency improvements have slowed dramatically in comparison to the historic rates. Microarchitecture innovations are needed to achieve the power budget imposed for exascale machines.

3 RESILIENCY

The exponential increase in core counts expected at exascale will lead to increases in the number of routers, switches, interconnects, and memory systems. Consequently, resilience is a major challenge for HPC applications on future exascale systems.

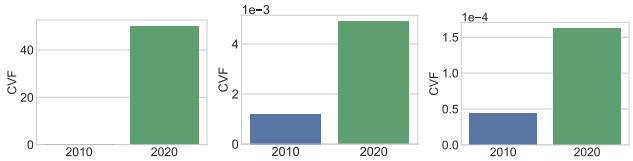
To asses resiliency, we quantify the vulnerability of the data structures and communication schemes. For the data structure, we use the data vulnerability factor (DVF) introduced in [11]. The DVF for a specific data structure (DVF_d) is defined as

$$DVF_d = FIT \times T_{exe} \times S_d \times N_{ha}, \tag{4}$$

where FIT is the failure in time (failures per billion hours per Mbit), S_d is the size of the data structure, and N_{ha} is the number of accesses to the hardware (main memory in this study).

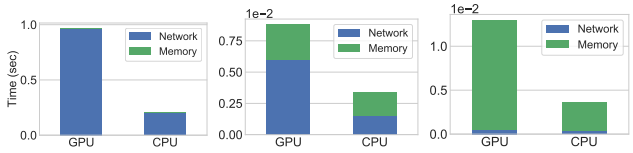
Figure 3 shows the DVF of the FFT, FMM, and multigrid methods. The results show that algorithms with random memory access pattern, such as the FMM, have higher DVF (more sensitive to memory errors) in comparison to algorithms with template-based memory access patterns, such as FFT and multigrid.

For communication, we introduce the communication vulnerability factor (CVF) which reflects the communication scheme as



(a) FFT **(b) FMM** **(c) Multigrid**

Figure 4: Communication vulnerability factors.



(a) FFT **(b) FMM** **(c) Multigrid**

Figure 5: Exascale projections of the FFT, FMM, and multigrid methods on GPU-based and CPU-based systems.

well as the network characteristics. The CVF for a specific kernel (CVF_k) is defined as

$$CVF_k = m \times T_{net} \times R_n, \tag{5}$$

where R_n is the network resilience.

Figure 4 shows that the all-to-all communication scheme of FFT makes it more sensitive to network failures in comparison to the hierarchical methods, FMM and multigrid. The hierarchical nature of FMM and multigrid reduces the $O(\sqrt{P})$ communication complexity of FFT to $O(\log P)$.

4 HETEROGENEITY

As accelerators advance in both performance and energy efficiency, heterogeneous systems have became critical ingredient in the pursuit of exascale computing. One bottleneck to consider on heterogeneous systems is the PCIe bus. Due to the high compute capability of the GPU, the PCIe bus can have a significant impact on the performance. The PCIe transfer time for n elements is given by

$$T_{PCIe}(n) = n\beta_{PCIe}, \tag{6}$$

where β_{PCIe} is the I/O bus inverse bandwidth in seconds per element.

Figure 5 shows memory and network access costs of the FFT, FMM, and multigrid methods. We consider extrapolated GPU- and CPU-based systems that have the same peak performance. The results show that all methods spend less time in both forms of communication on the CPU-based system. However, this system requires over 10× as many processors as the GPU-based system. This cost could be prohibitive.

ACKNOWLEDGMENTS

This material is based in part upon work supported by the Department of Energy, National Nuclear Security Administration, under Award Number DE-NA0002374.

REFERENCES

[1] Lorena A Barba and Rio Yokota. 2013. How will the fast multipole method fare in the exascale era. *SIAM News* 46, 6 (2013), 1–3.

[2] Aparna Chandramowlishwaran, JeeWhan Choi, Kamesh Madduri, and Richard Vuduc. 2012. Brief Announcement: Towards a Communication Optimal Fast Multipole Method and Its Implications at Exascale. In *Proceedings of the Twenty-fourth Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '12)*. ACM, New York, NY, USA, 182–184. https://doi.org/10.1145/2312005.2312039

[3] Jee Choi, Aparna Chandramowlishwaran, Kamesh Madduri, and Richard Vuduc. 2014. A CPU: GPU Hybrid Implementation and Model-Driven Scheduling of the Fast Multipole Method. In *Proceedings of Workshop on General Purpose Processing Using GPUs (GPGPU-7)*. ACM, New York, NY, USA, Article 64, 8 pages. https://doi.org/10.1145/2576779.2576787

[4] Jee Whan Choi, Daniel Bedard, Robert Fowler, and Richard Vuduc. 2013. A roofline model of energy. In *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*. IEEE, 661–672.

[5] Kenneth Czechowski, Casey Battaglino, Chris McClanahan, Kartik Iyer, P.-K. Yeung, and Richard Vuduc. 2012. On the Communication Complexity of 3D FFTs and Its Implications for Exascale. In *Proceedings of the 26th ACM International Conference on Supercomputing (ICS '12)*. ACM, New York, NY, USA, 205–214. https://doi.org/10.1145/2304576.2304604

[6] Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, et al. 2011. The international exascale software project roadmap. *International Journal of High Performance Computing Applications* 25, 1 (2011), 3–60.

[7] Hadi Esmaeilzadeh, Emily Blem, Renée St. Amant, Karthikeyan Sankaralingam, and Doug Burger. 2013. Power Challenges May End the Multicore Era. *Commun. ACM* 56, 2 (Feb. 2013), 93–102. https://doi.org/10.1145/2408776.2408797

[8] H. Gahvari and W. Gropp. 2010. An introductory exascale feasibility study for FFTs and multigrid. In *2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS)*. 1–9. https://doi.org/10.1109/IPDPS.2010.5470417

[9] Huda Ibeid, Rio Yokota, and David Keyes. 2016. A performance model for the communication in fast multipole methods on high-performance computing platforms. *The International Journal of High Performance Computing Applications* 30, 4 (2016), 423–437. https://doi.org/10.1177/1094342016634819 arXiv:http://dx.doi.org/10.1177/1094342016634819

[10] Richard Vuduc and Kent Czechowski. 2011. What GPU Computing Means for High-End Systems. *IEEE Micro* 31, 4 (July 2011), 74–78. https://doi.org/10.1109/MM.2011.78

[11] Li Yu, Dong Li, Sparsh Mittal, and Jeffrey S. Vetter. 2014. Quantitatively Modeling Application Resilience with the Data Vulnerability Factor. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '14)*. IEEE Press, Piscataway, NJ, USA, 695–706. https://doi.org/10.1109/SC.2014.62