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

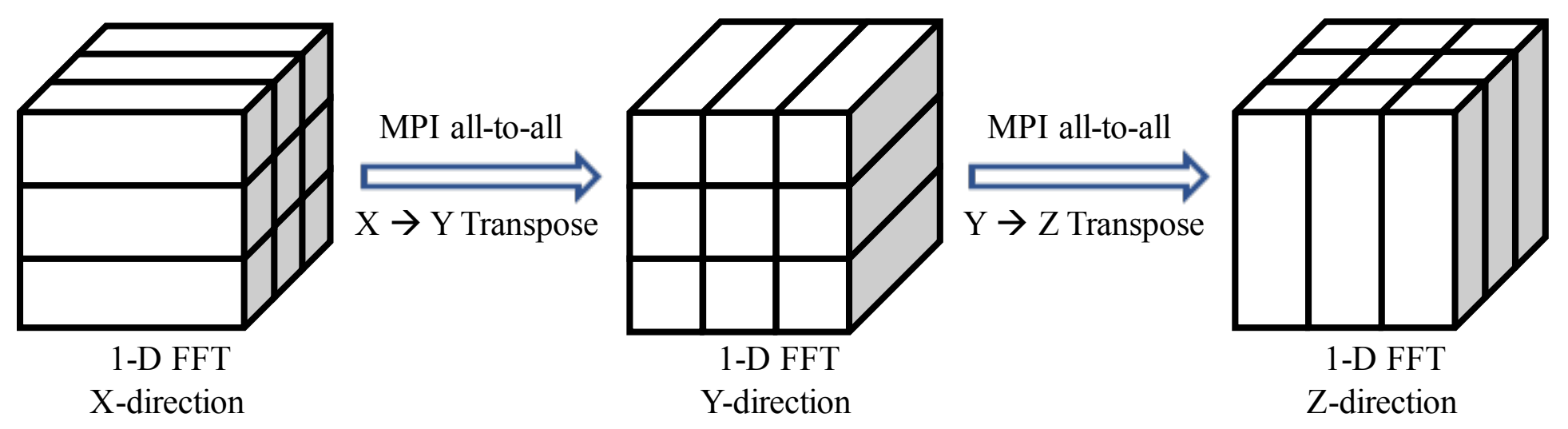
Huda Ibeid, Luke Olson, and William Gropp
University of Illinois at Urbana-Champaign

Center for Exascale Simulation of Plasma-Coupled Combustion

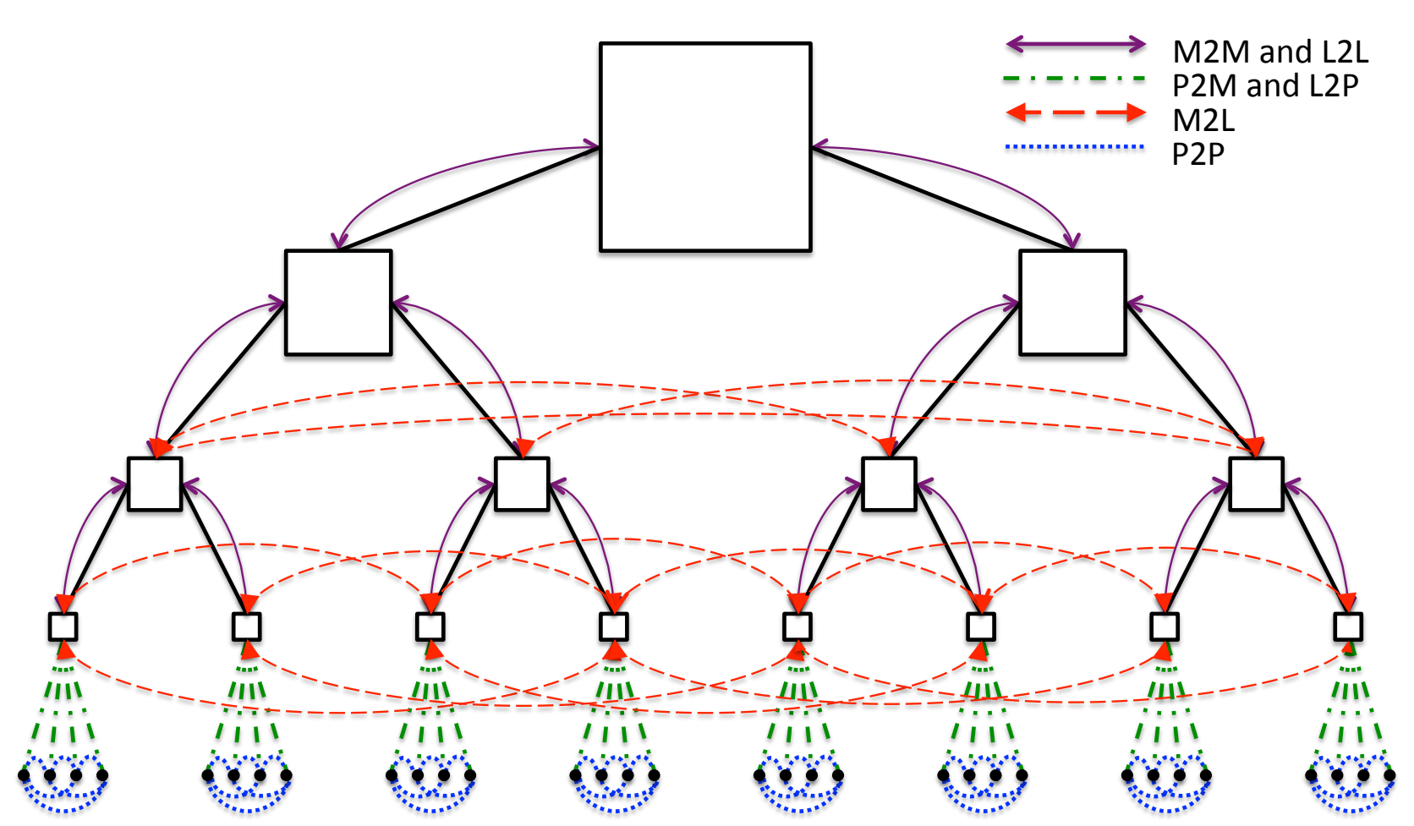
Abstract

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 [1] 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.

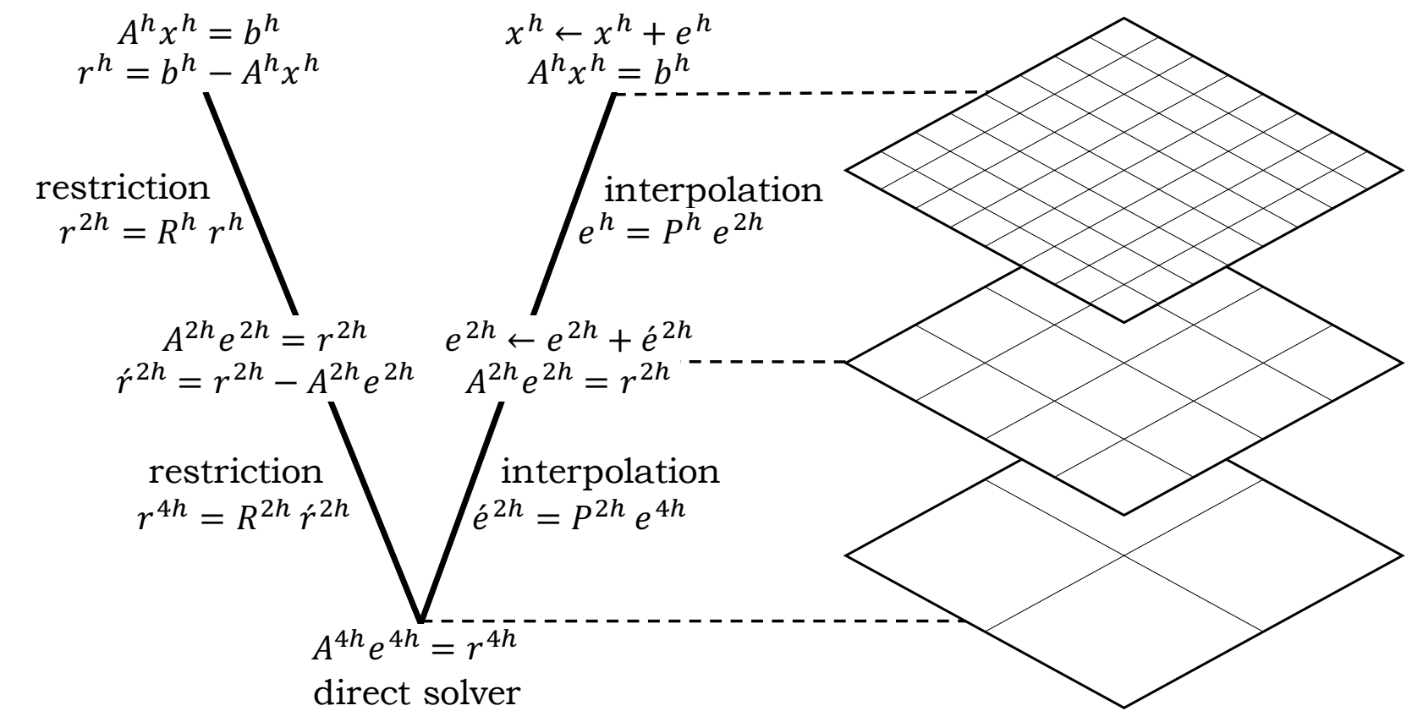
Fast Fourier Transform



Fast Multipole Method



Multigrid



References

[1] Jack Dongarra et al. The international exascale software project roadmap. In *IJHPCA*, 2011.
[2] Jee Choi et al. A roofline model of energy. In *IPDPS*, 2013.
[3] Li Yu et al. Quantitatively modeling application resilience with the data vulnerability factor. In *SC '14*, 2014.

Processor Architecture Projections

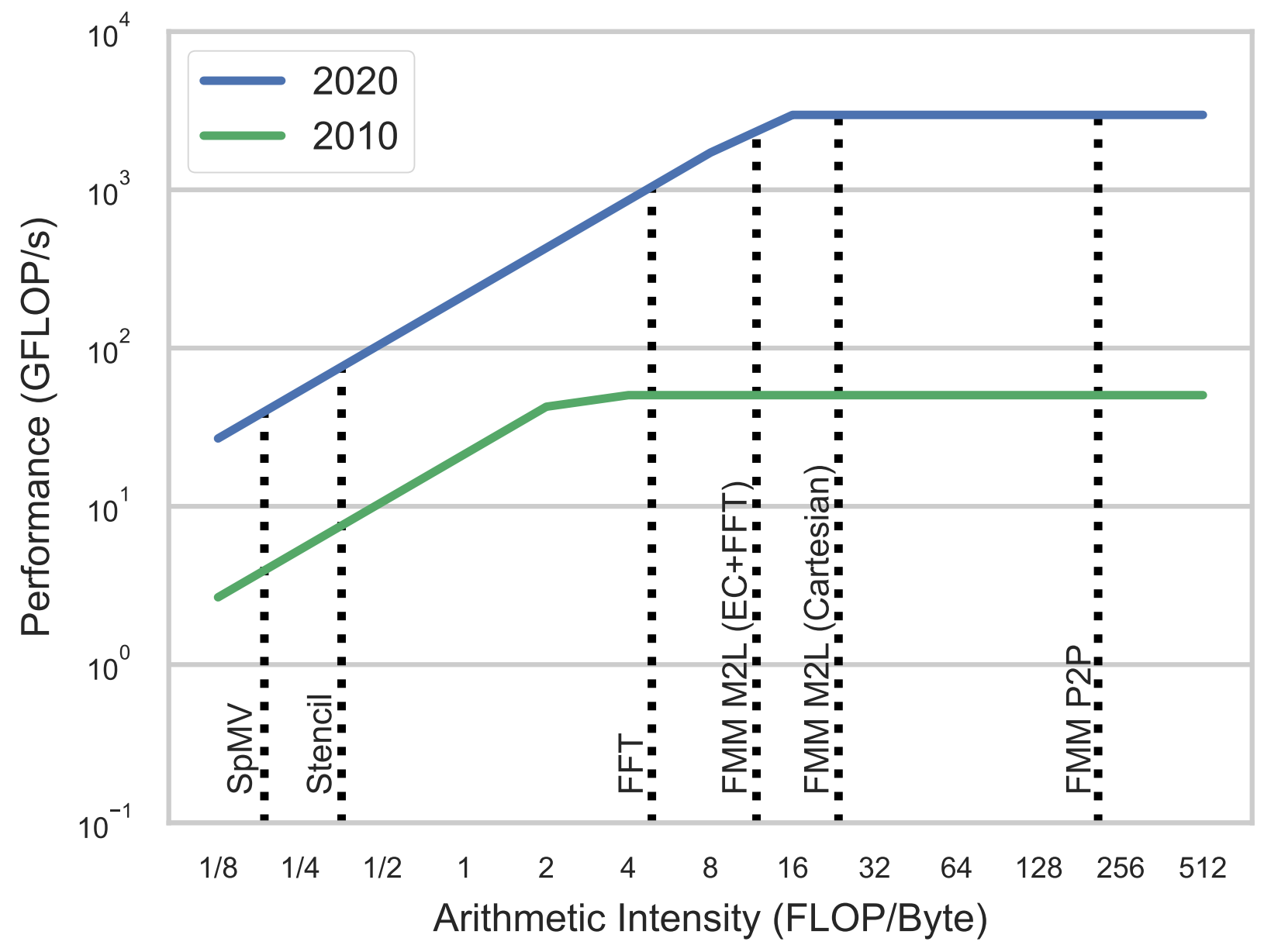
Parameter		2010 Value	10-year increase	2020 Value
Peak	$1/t_{cpu}$ $1/t_{gpu}$	50.4 GF/s 515 GF/s	59.0×	3 TP/s 30 TP/s
Cores	ρ_{cpu} ρ_{gpu}	6 448	40.7×	134 18k
Memory bandwidth	$1/\beta_{cpu}$ $1/\beta_{gpu}$	21.3 GB/s 144 GB/s	9.7×	206 GB/s 1.4 TB/s
Fast memory	Z_{cpu} Z_{gpu}	6 MB 2.7 MB	32.0×	192 MB 86.4 MB
Line size	L_{cpu} L_{gpu}	64 B 128 B	2.0×	128 B 256 B
Link bandwidth	$1/\beta_{link}$	10 GB/s	21.8×	218 GB/s

Concurrency

Assuming arithmetic and memory operations can be overlapped, the total execution time 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 memory access cost.



Inter-node communication cost is modeled using the baseline model

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

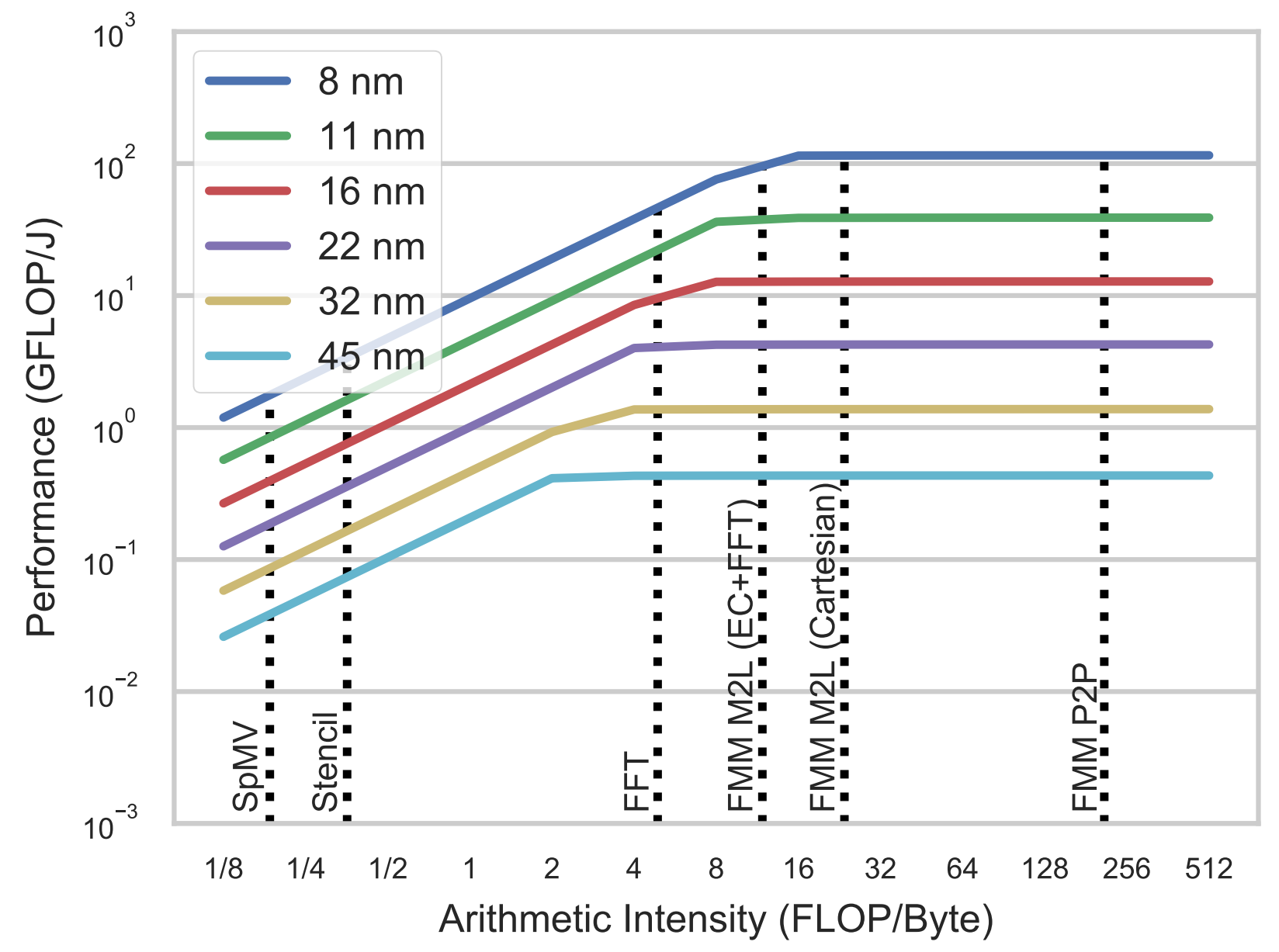
where α represents communication latency, β_{link} is the send time per element, m is the number of messages sent, and n is the total message size.

Energy

To compare power and energy efficiency, we use the energy Roofline model [2] where energy cost (Joules) is defined as

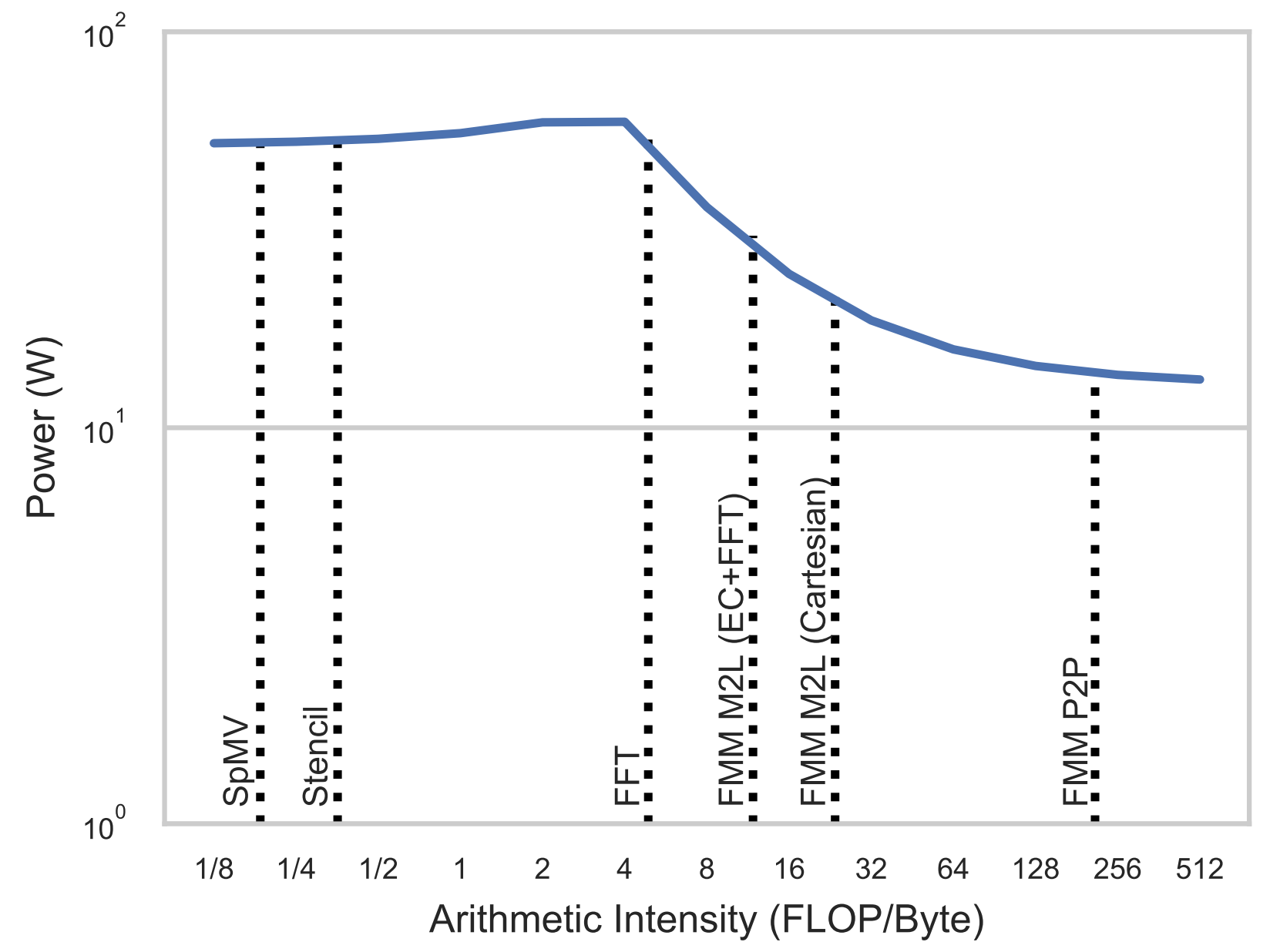
$$E \equiv E_{flop} + E_{mem} + E_0(T_{exe}), \tag{3}$$

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.

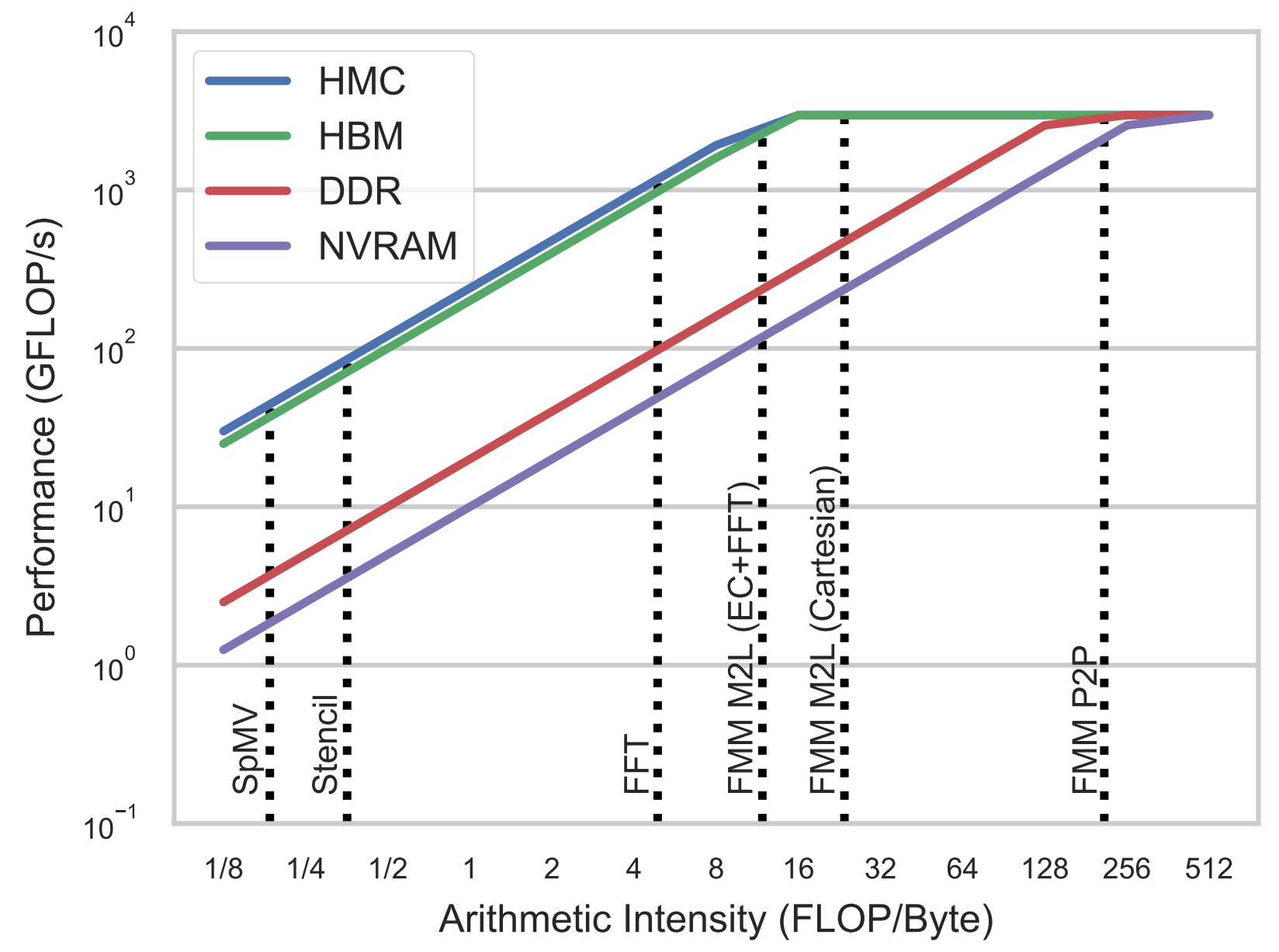


The above energy model can be used to estimate the average power consumed by an algorithm as

$$Power \equiv \frac{E}{T_{exe}}. \tag{4}$$



Memory

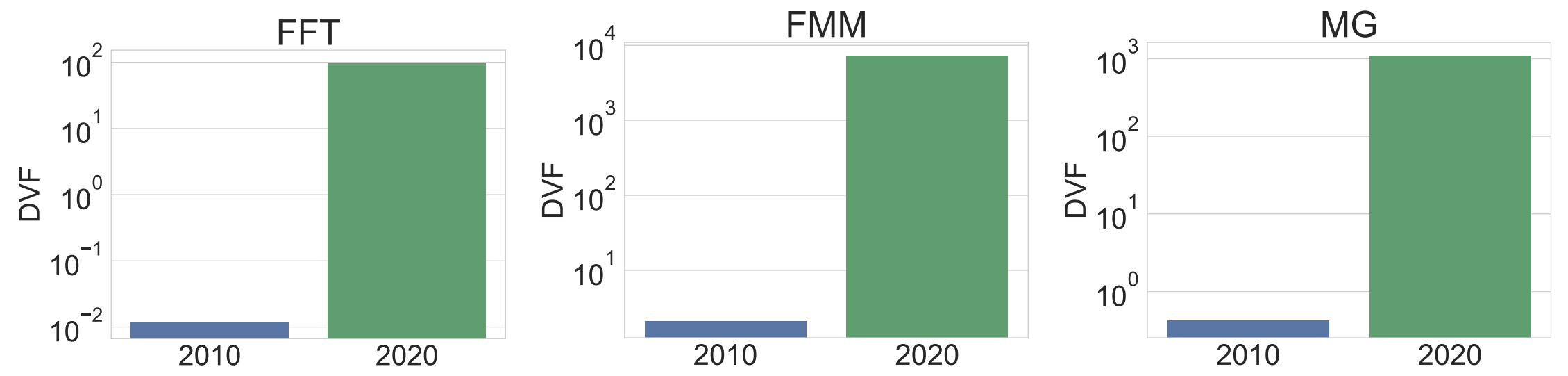


Resiliency

For the data structure, we use the data vulnerability factor (DVF) [3],

$$DVF_d = FIT \times T_{exe} \times S_d \times N_{ha}, \tag{5}$$

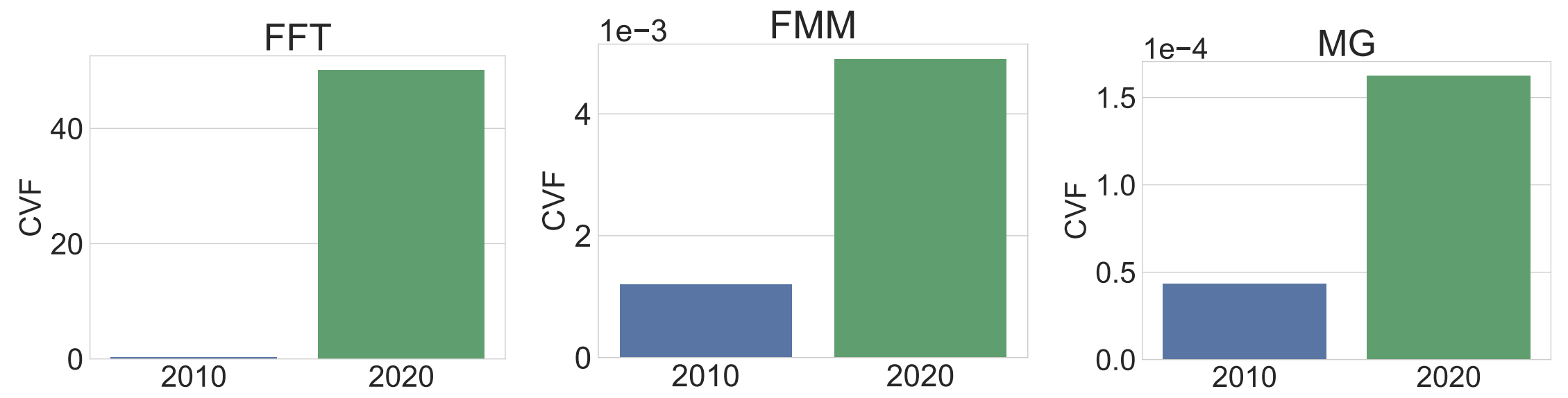
where FIT is the failure in time, S_d is the size of the data structure, and N_{ha} is the number of hardware accesses.



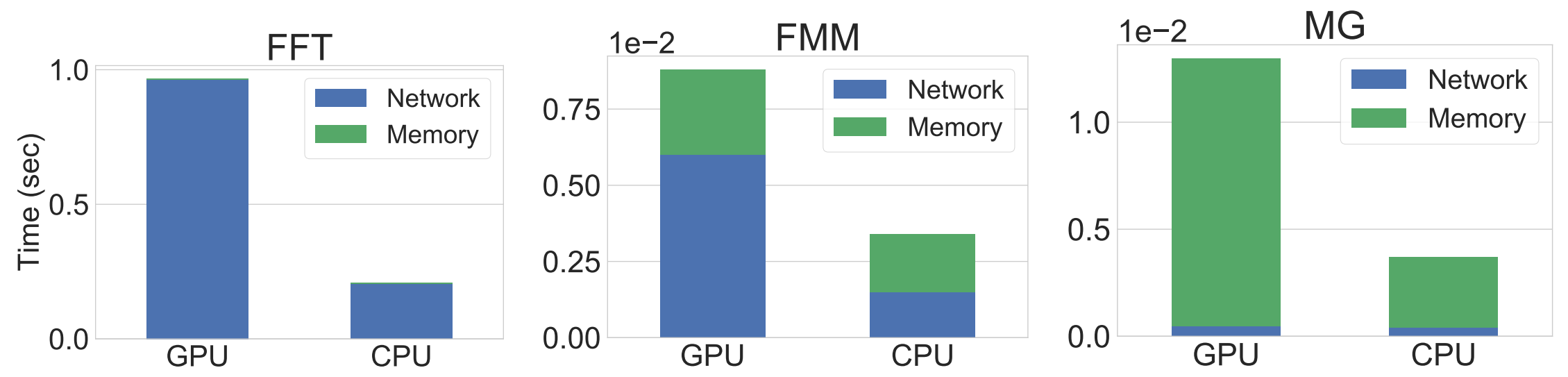
For communication, we introduce the communication vulnerability factor (CVF),

$$CVF_k = m \times T_{net} \times R_n, \tag{6}$$

where R_n is the network resilience.



Heterogeneity



Exascale Lessons

On the application side:

- Reduce memory requirements.
- Improve arithmetic intensity.
- Enable energy-efficient software.

On the architectural side:

- Increase memory bandwidth.
- Enable energy-efficient computers.

H. Ibeid, L. Olson, and W. Gropp, FFT, FMM, and Multigrid on the Road to Exascale: Performance Challenges and Opportunities, arXiv (2017).

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