In search of performance versatility

This talk: https://jedbrown.org/files/20150624-Versatility.pdf

Jed Brown jed@jedbrown.org (ANL and CU Boulder)

PADAL, Berkeley, 2015-06-24

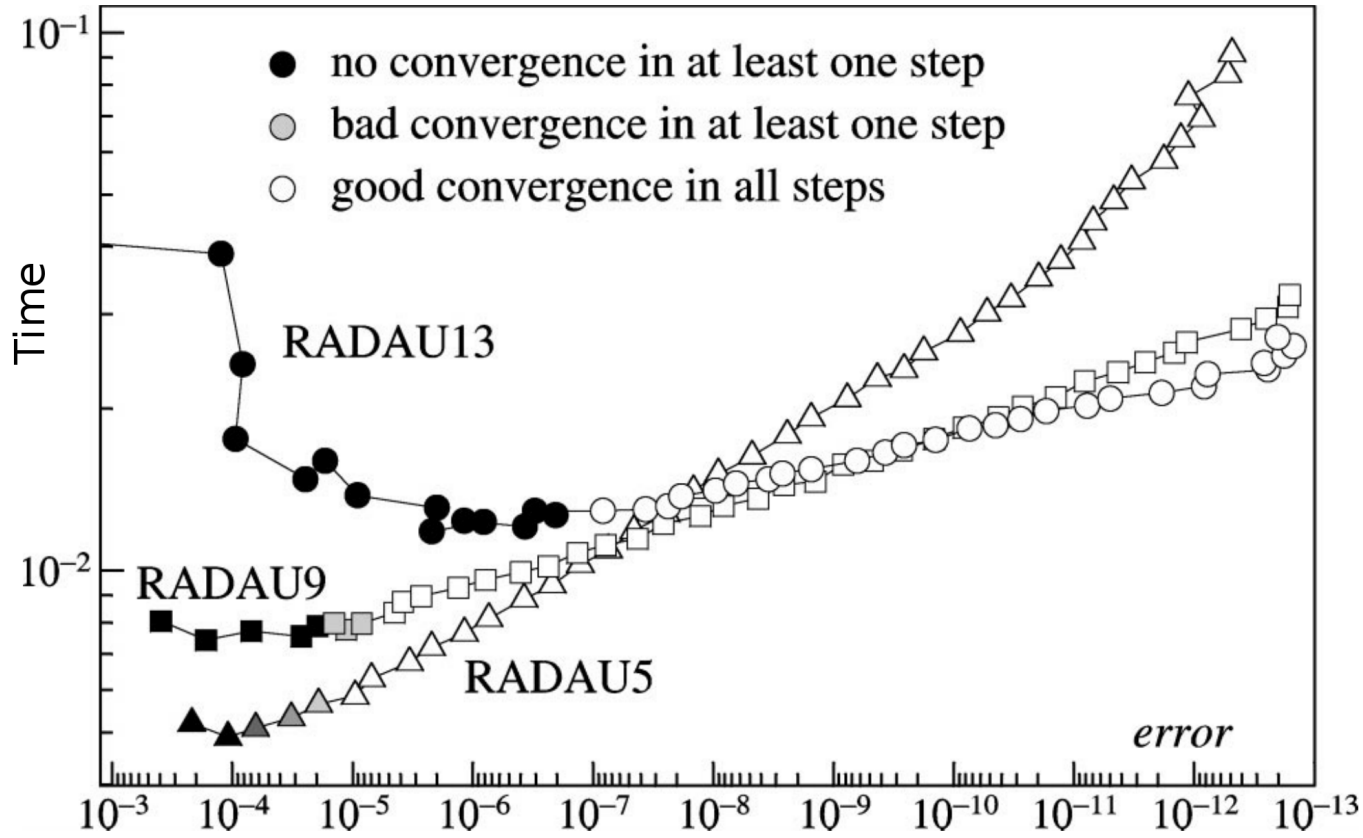
What is performance?

Dimensions

- Model complexity
- Accuracy
- Time
  - per problem instance
  - for the first instance
  - compute time versus human time
- Cost
  - incremental cost
  - subsidized?

- Terms relevant to scientist/engineer
- Compute meaningful quantities – needed to make a decision or obtain a result of scientific value—not one iteration/time step
- No flop/s, number of elements/time steps

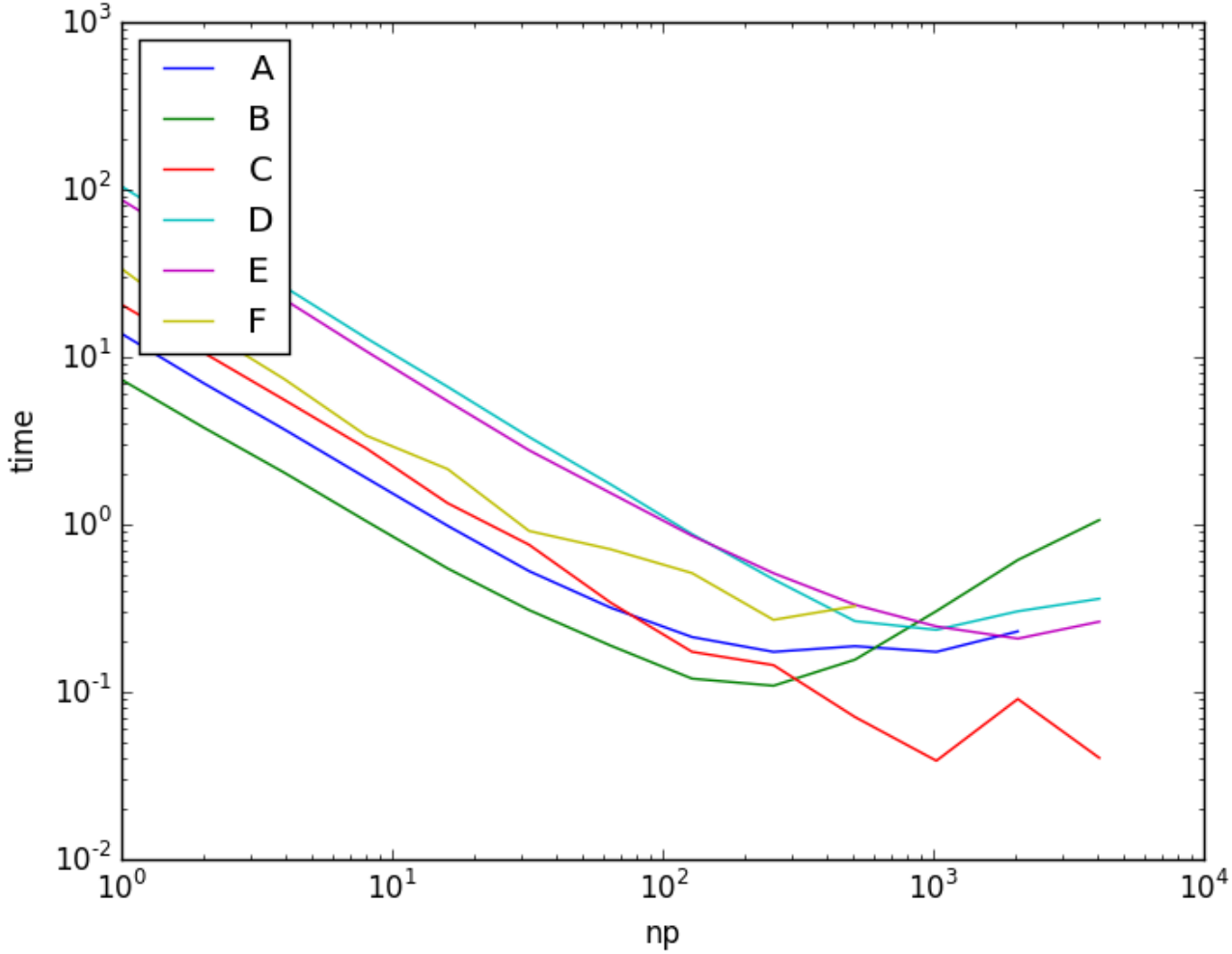
Work-precision diagram: *de rigueur* in ODE community



[Hairer and Wanner (1999)]

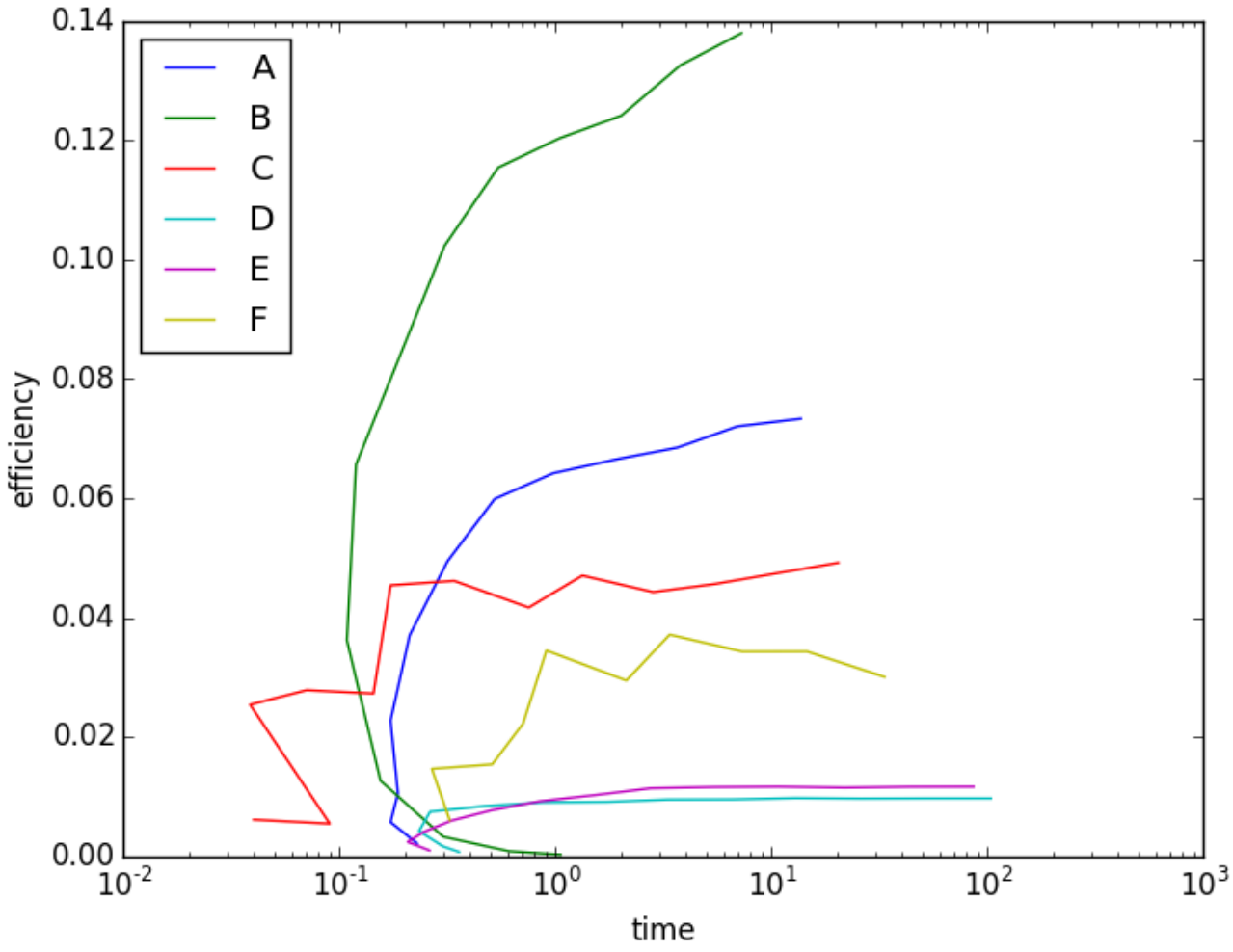
- Tests discretization, adaptivity, algebraic solvers, implementation
- No reference to number of time steps, flop/s, etc.
- Useful performance results inform *decisions* about *tradeoffs*.

Strong Scaling: efficiency-time tradeoff



- Good: shows absolute time
- Bad: log-log plot makes it difficult to discern efficiency
  - Stunt 3: `http://blogs.fau.de/hager/archives/5835`
- Bad: plot depends on problem size

Strong Scaling: efficiency-time tradeoff



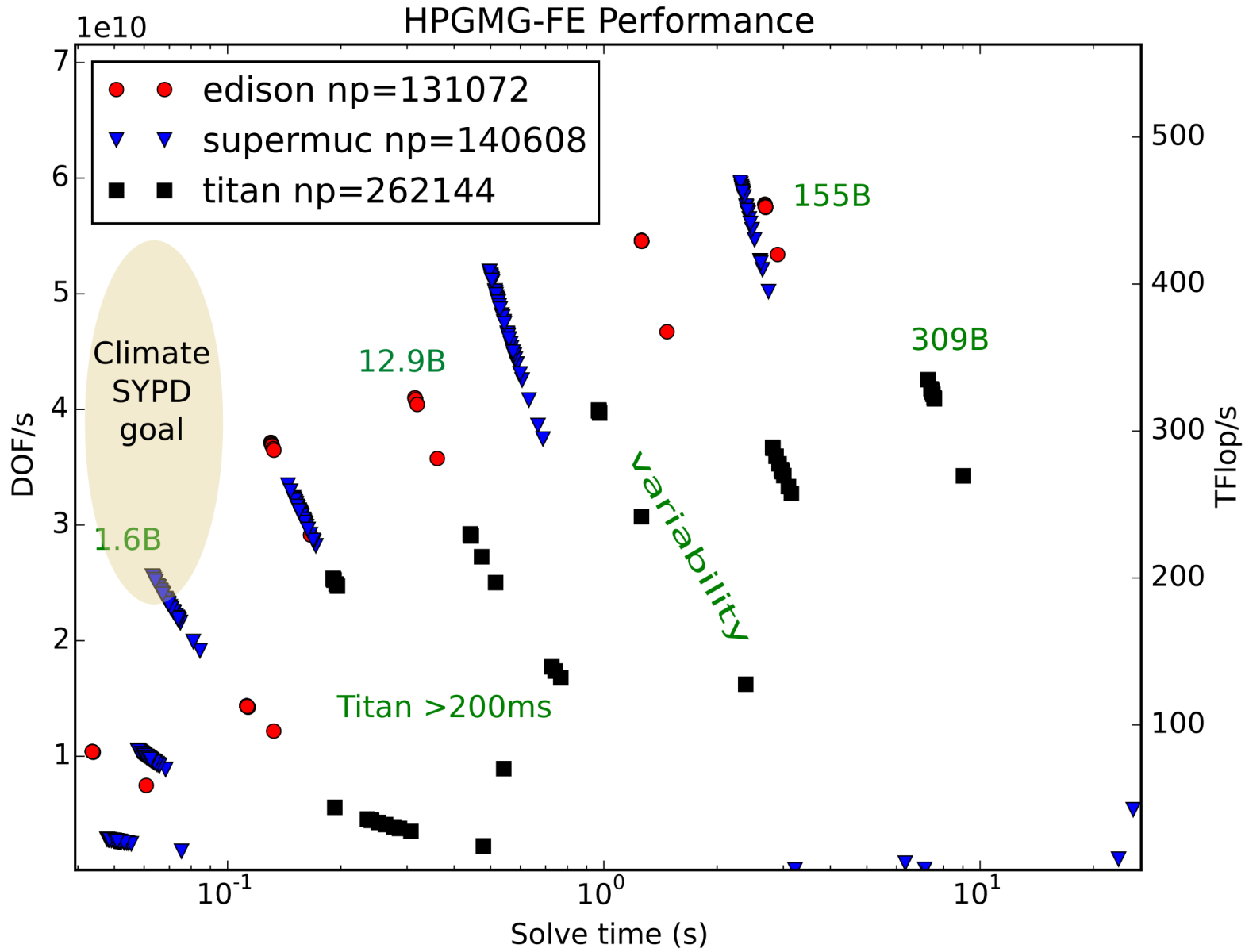
- Good: absolute time, absolute efficiency (like DOF/s/cost)
- Good: independent of problem size for perfect weak scaling
- Bad: hard to see machine size (but less important)

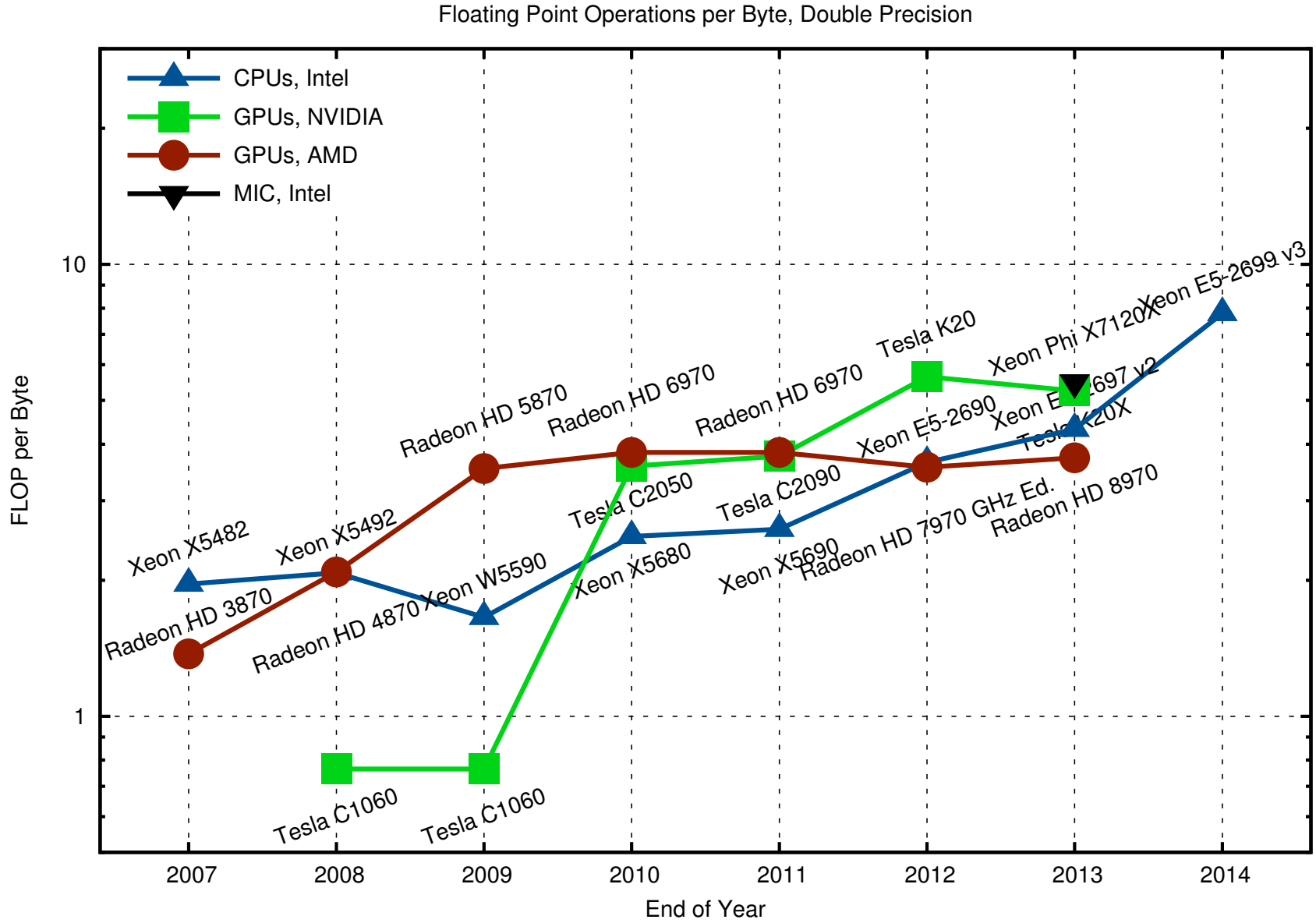
Exascale Science & Engineering Demands

- Model fidelity: resolution, multi-scale, coupling
  - Transient simulation is not weak scaling: $\Delta t \sim \Delta x$
- Analysis using a sequence of forward simulations
  - Inversion, data assimilation, optimization
  - Quantify uncertainty, risk-aware decisions
- Increasing relevance $\Longrightarrow$ external requirements on time
  - Policy: 5 SYPD to inform IPCC
  - Weather, manufacturing, field studies, disaster response
- "weak scaling" [. . .] will increasingly give way to "strong scaling"

  [The International Exascale Software Project Roadmap, 2011]
- ACME @ 25 km scaling saturates at $< 10\%$ of Titan (CPU) or Mira
  - Cannot decrease Δx: SYPD would be too slow to calibrate
  - "results" would be meaningless for 50-100y predictions, a "stunt run"
- **ACME v1 goal of 5 SYPD is pure strong scaling.**
  - Likely faster on Edison (2013) than any DOE machine –2020
  - Many non-climate applications in same position.

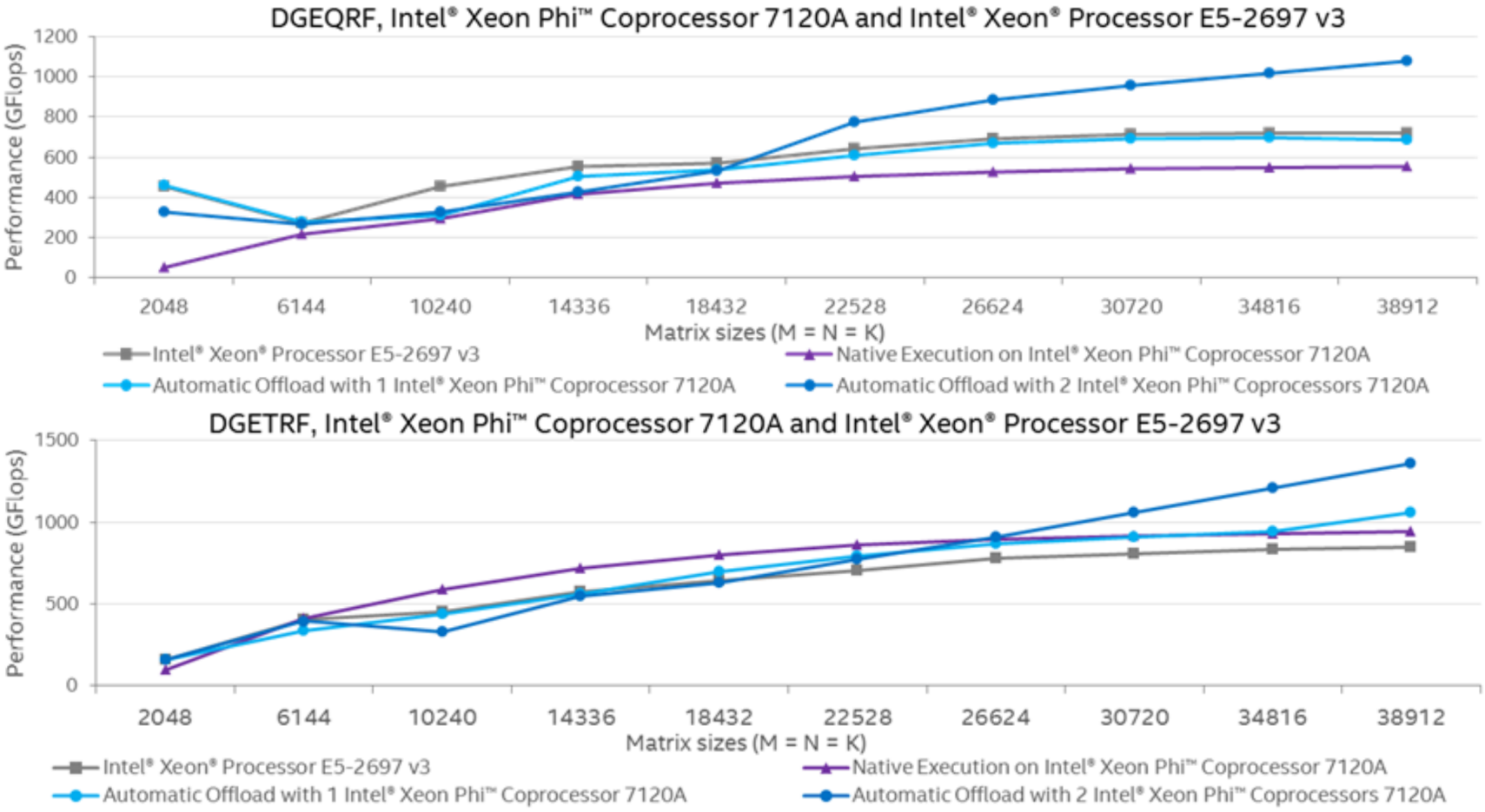
HPGMG-FE on Edison, SuperMUC, Titan




Floating Point Operations per Byte, Double Precision
CPUs, Intel
GPUs, NVIDIA
GPUs, AMD
MIC, Intel
FLOP per Byte
10
1
2007
2008
2009
2010
2011
2012
2013
2014
End of Year
Xeon X5482
Xeon X5492
Xeon W5590
Xeon X5680
Xeon X5690
Xeon E5-2690
Xeon E5-2699 v3
Radeon HD 3870
Radeon HD 4870
Radeon HD 5870
Radeon HD 6970
Radeon HD 6970
Radeon HD 7970 GHz Ed.
Radeon HD 8970
Tesla C1060
Tesla C1060
Tesla C2050
Tesla C2090
Tesla K20
Xeon Phi X7120

[c/o Karl Rupp]

Arithmetic intensity is not enough



- QR and LU factorization have same complexity.
- Stable QR factorization involves more synchronization.
- Synchronization is much more expensive on Xeon Phi.

How much parallelism out of how much cache?

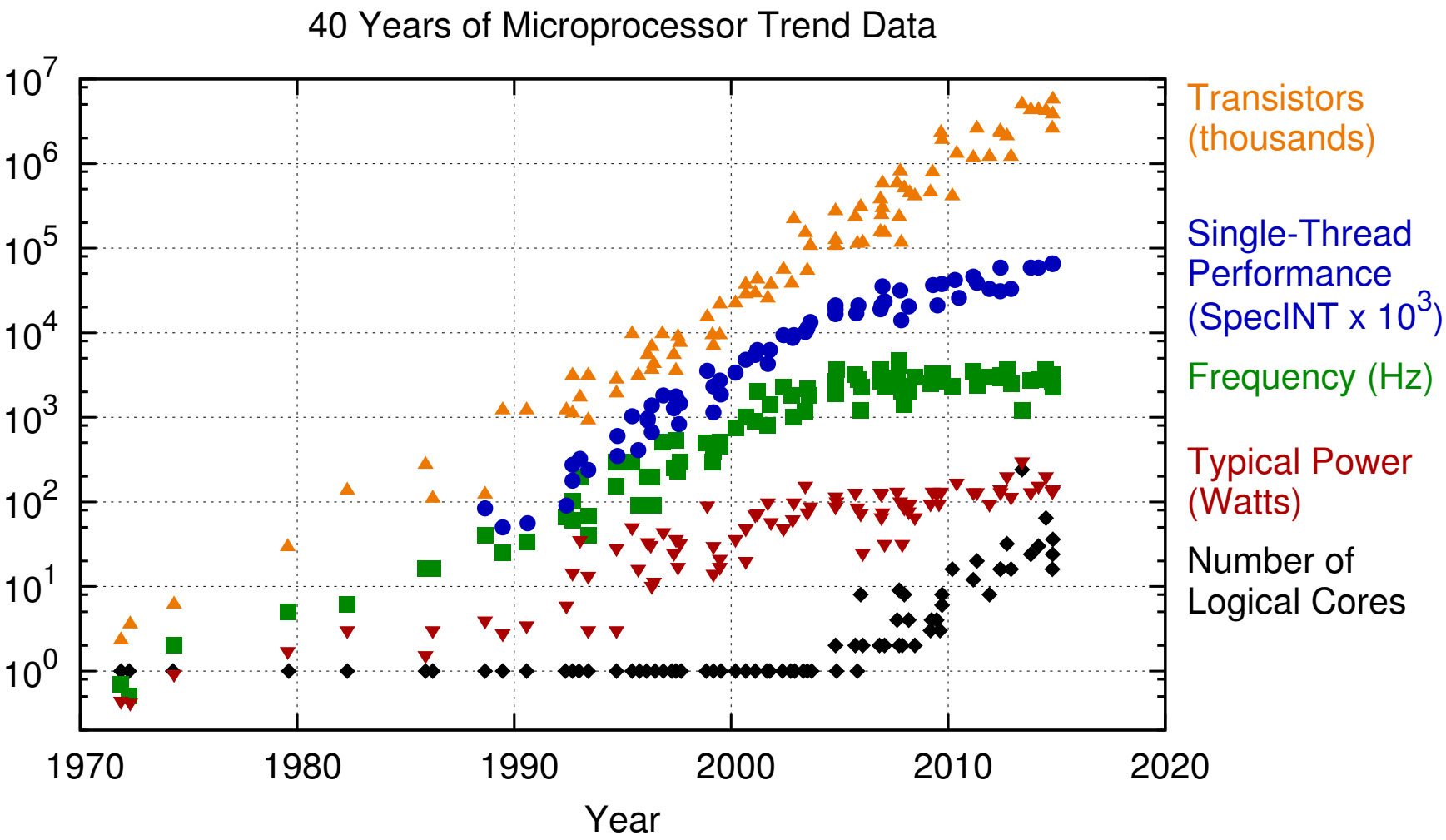
Processor	v width	threads	F/inst	latency	L1D	L1D/#par
Nehalem	2	1	2	5	32 KiB	1638 B
Sandy Bridge	4	2	2	5	32 KiB	819 B
Haswell	4	2	4	5	32 KiB	410 B
BG/P	2	1	2	6	32 KiB	1365 B
BG/Q	4	4	2	6	32 KiB	682 B
KNC	8	4	4	5	32 KiB	205 B
Tesla K20	32	*	2	10	64 KiB	102 B

- Most “fast” algorithms do about $O(N)$ flops on N data
- xGEMM does $O(N^{3/2})$ flops on N data
- Exploitable parallelism limited by cache and register load/store
- L2/L3 performance highly variable between architectures

Vectorization versus memory locality

- Each vector lane and pipelined instruction need their own operands
- Can we extract parallelism from smaller working set?
  - Sometimes, but more cross-lane and pipeline dependencies
  - More complicated/creative code, harder for compiler
- Good implementations strike a brittle balance (e.g., Knepley, Rupp, Terrel; HPGMG-FE)
- Applications change discretization order, number of fields, etc.
  - CFD: 5-15 fields
  - Tracers in atmospheric physics: 100 species
  - Adaptive chemistry for combustion: 10-10000 species
  - Crystal growth for mesoscale materials: 10-10000 fields
- AoS or SoA?
  - Choices not robust to struct size
  - AoS good for prefetch and cache reuse
  - Can pack into SoA when necessary

SPECint is increasing despite stagnant clock

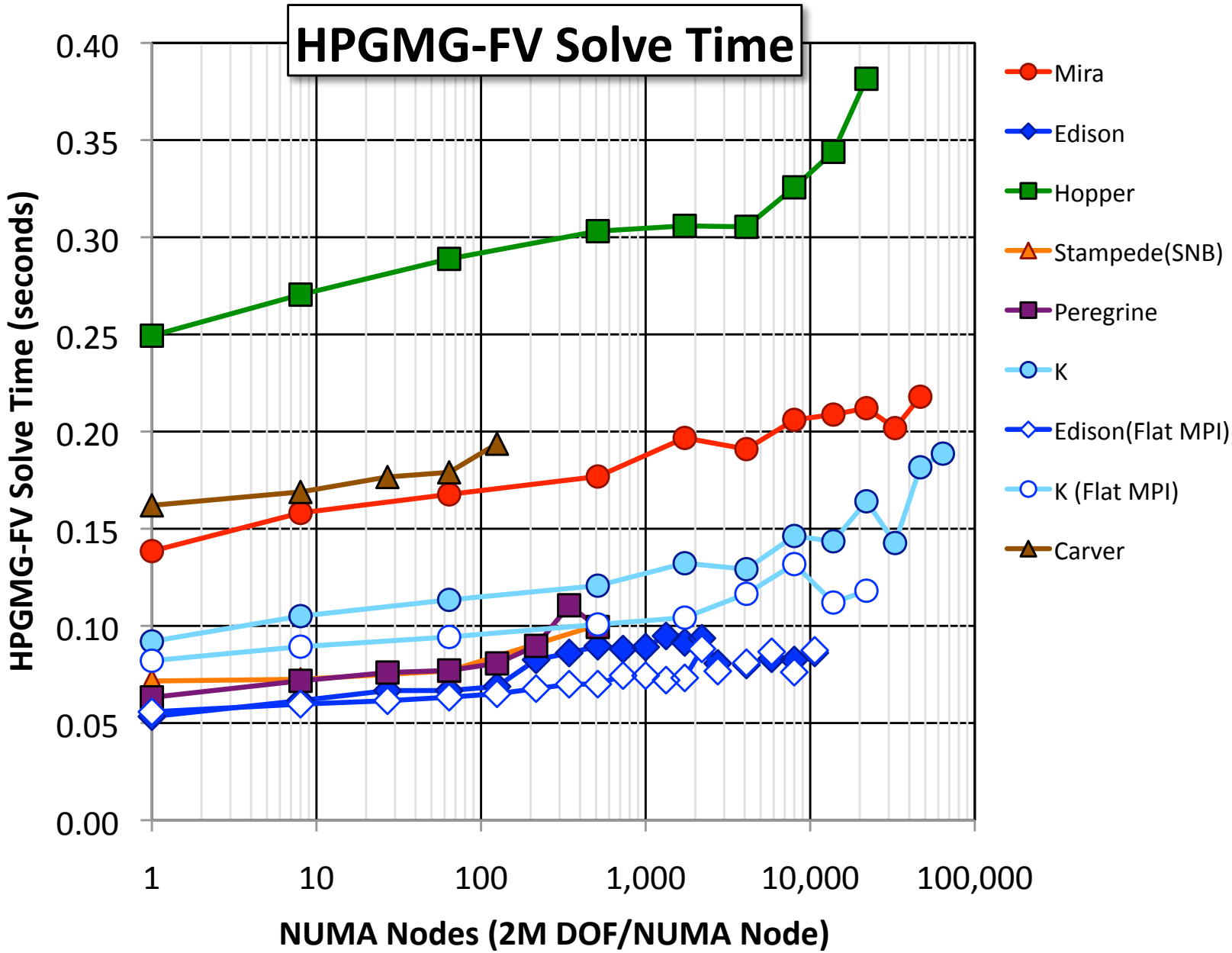


- Karl Rupp's update to figure by Horowitz et al.

Messaging from threaded code

- Off-node messages need to be packed and unpacked
- Many MPI+threads apps pack in serial – bottleneck
- Extra software synchronization required to pack in parallel
  - Formally $O(\log T)$ critical path, T threads/NIC context
  - Typical OpenMP uses barrier – oversynchronizes
- `MPI_THREAD_MULTIPLE` – atomics and $O(T)$ critical path
- Choose serial or parallel packing based on T and message sizes?
- Hardware NIC context/core now, maybe not in future
- What is lowest overhead approach to message coalescing?

HPGMG-FV: flat MPI vs MPI+OpenMP (Aug 2014)



Outlook

- Application scaling mode must be scientifically relevant
- Algorithmic barriers exist
  - Throughput architectures are not just “hard to program”
- Vectorization versus memory locality
- Over-decomposition adds overhead and lengthens critical path
- Versatile architectures are needed for model coupling and advanced analysis
  - Why will Cori have DRAM?
- Abstractions must be durable to changing scientific needs
- “Energy efficiency” is not if algorithms give up nontrivial constants
- What is the cost of performance variability?
  - Measure best performance, average, median, 10th percentile?
- The real world is messy!