Experience Using MILC and QUDA on Various GPU Systems

Steven Gottlieb
Indiana University

Efficient Simulations on GPU Hardware
ETH Zürich
October 24-27, 2022

Volume Scaling Studies

- I have been doing volume weak scaling studies for probably three decades
  - QCD codes tend to be memory bandwidth bound, so the performance of a processor depends on how big the cache is and the local volume assigned to the processor.
  - With a small local volume, the cache will be useful and performance may be very good.
  - However, a small local volume makes great demands on the communication system since the ratio of local work to volume of data communicated fall like $1/L$, where L is the linear size of system.
  - In the early days, there was sometimes a sweet spot in L with maximum performance.
  - These days, we usually see performance increases with L, and we are looking for the smallest value of L for which we find the performance acceptable.

Gauge Configuration Generation

- ✦We will concentrate on gauge configuration generation with staggered quarks.
- ✦We use the MILC code with QUDA.
  - For NVIDIA GPUs, we use the CUDA backend; for AMD the more recent HIP backend. This is controlled by the OFFLOAD symbol.
- ✦The code is su3_rhmd_hisq.
- ✦In production running, the bulk of the time is spent on the multishift conjugate gradient (CG) solver.
- ✦The fermion force is the second most important routine.
  - Unfortunately, the flop counter in QUDA is not working for this routine, so code reports time per call, not a flop rate.
- ✦Disadvantage of volume scaling studies is not having equilibrated input configuration for each case.
  - Solver takes more iterations with equilibrated input config. Compensate by plotting maximum performance, which may still be an underestimate.

Many Choices

- ✦With MILC and QUDA, there are many, many choices:
  - generic precision of the MILC code (single, double)
  - QUDA also offers different precisions
  - MILC input file can request different precisions for different operations
  - QUDA solver offers multiple precisions within the solve: HALF_MIXED, MAX_MIXED
    - compile time option in MILC: WANT_MIXED_PRECISION_GPU = 0, 1, 2
- QUDA offers various message passing enhancements
  - Peer to Peer communications on node
  - gauge field compression schemes
  - GPU direct
  - these are controlled by run time environment variables
- QUDA does auto tuning so we do two runs:
  - first run tunes and creates tune cache; second uses values in tune cache

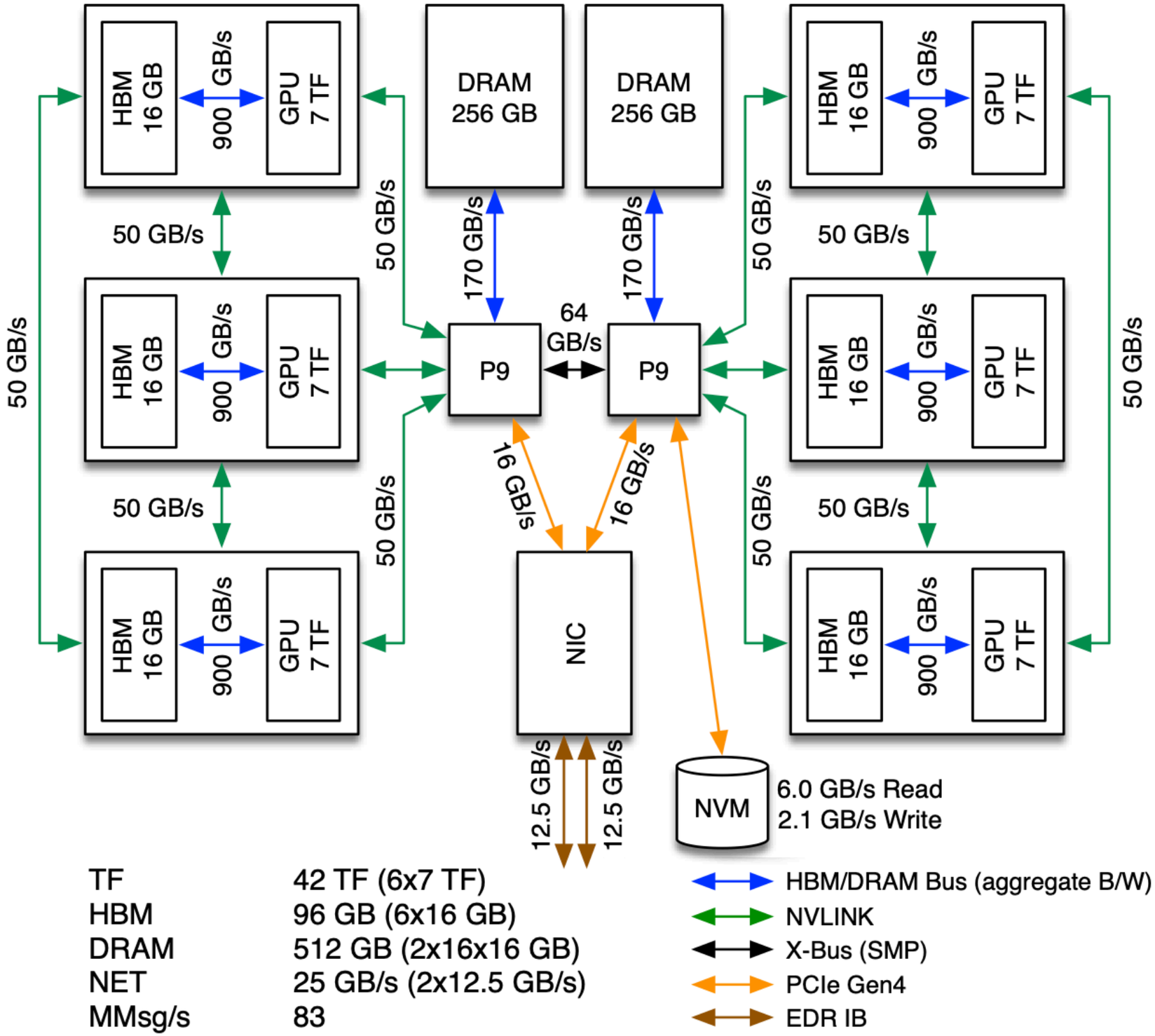
A Few Words About Plots

- ✦ We tend to show performance in way that emphasizes efficiency
  - i.e., we show flops/second per GPU or per node
  - if the result is constant as number of node changes, that is good
    - obviously want a high value
  - if we are showing time per site in local volume, flat is also good
    - but we want a small value here
- ✦ For the multi-GPU (mpi-task) runs, communicate in more dimensions until we get to 16 tasks
  - expect performance to drop until 16 tasks
  - want performance to be constant for > 16 tasks
  - most communication is point to point, except for global sums in CG solver

Platforms

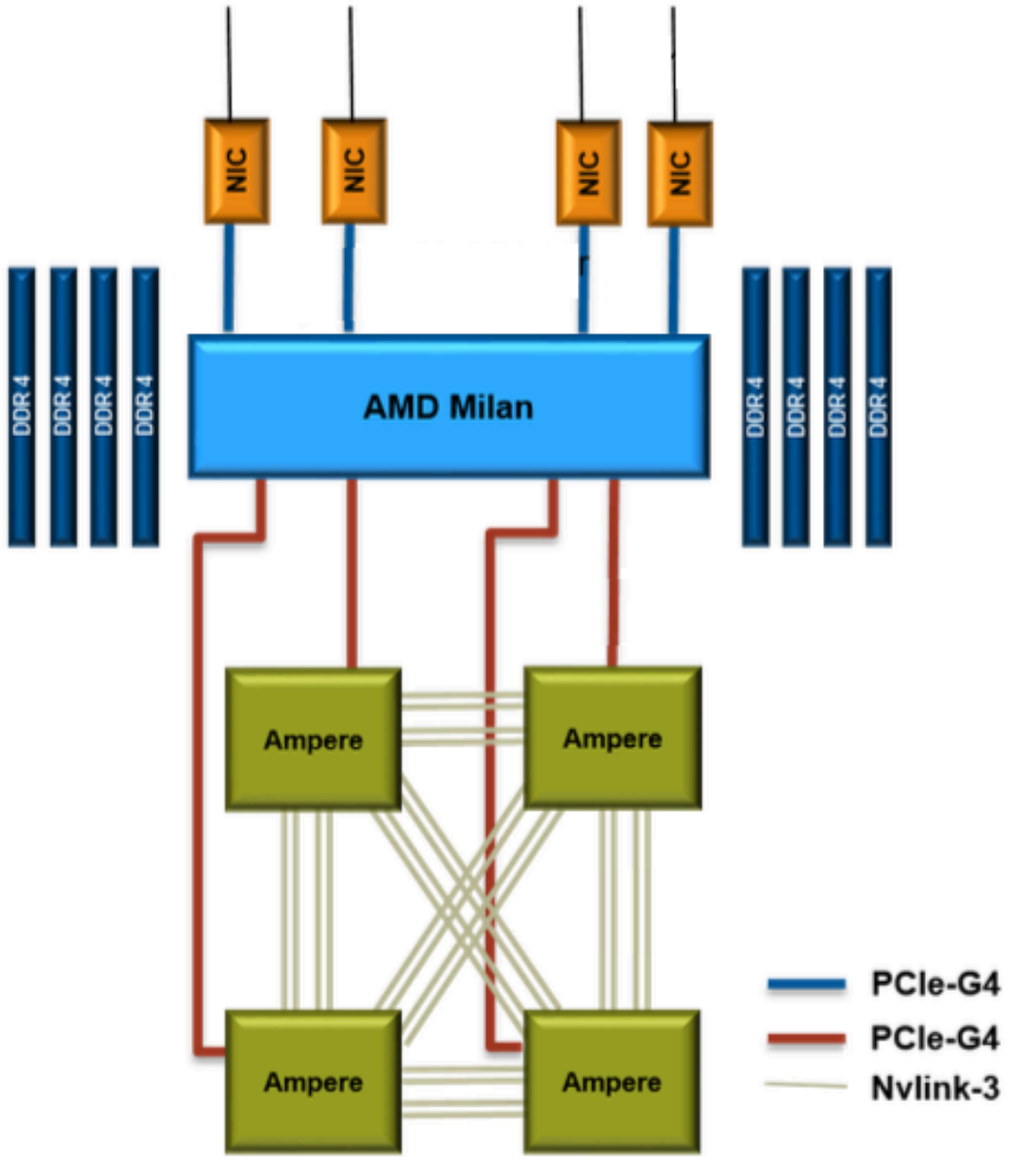
- ✦OLCF Summit: 6 NVIDIA Tesla V100-SXM2 GPUs per node with 16 GB high-bandwidth-memory per GPU with dual rail EDR Infiniband (23 GB/s)
- ✦NERSC Perlmutter: 4 A100-SXM4-40 GB GPUs per node
  - initially Slingshot 10 (2X100 Gb/s); now Slingshot 11 (4X200 Gb/s)
- ✦OLCF Crusher: 4 AMD MI250X GPUs per node and 4 HPE Slingshot 200 Gbps NICS (total injection BW is 100 GB/s)
  - However, each AMD MI250X has two Graphics Compute Dies (GCDs), so we treat it as a system with 8 GPUs per node, each with 64 GB of high-bandwidth-memory
- ✦Big Red 200: like a mini-version of Perlmutter
- ✦ALCF DGX-A100: 8 GPUs with NVLink per system
  - I always had problems running on multiple systems and never had the time to get the problem solved.

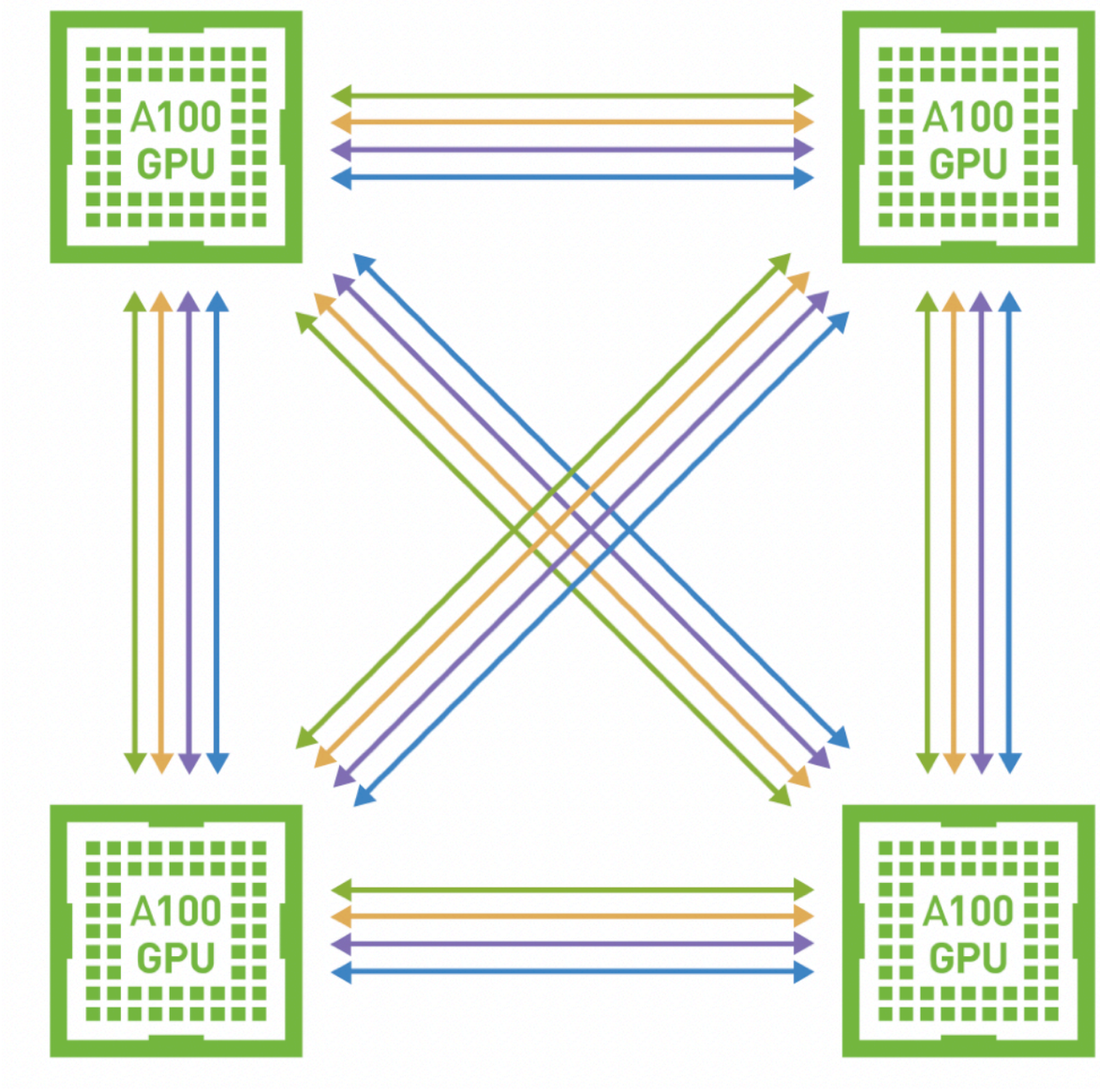
Summit



https://docs.olcf.ornl.gov/systems/summit_user_guide.html#summit-nodes

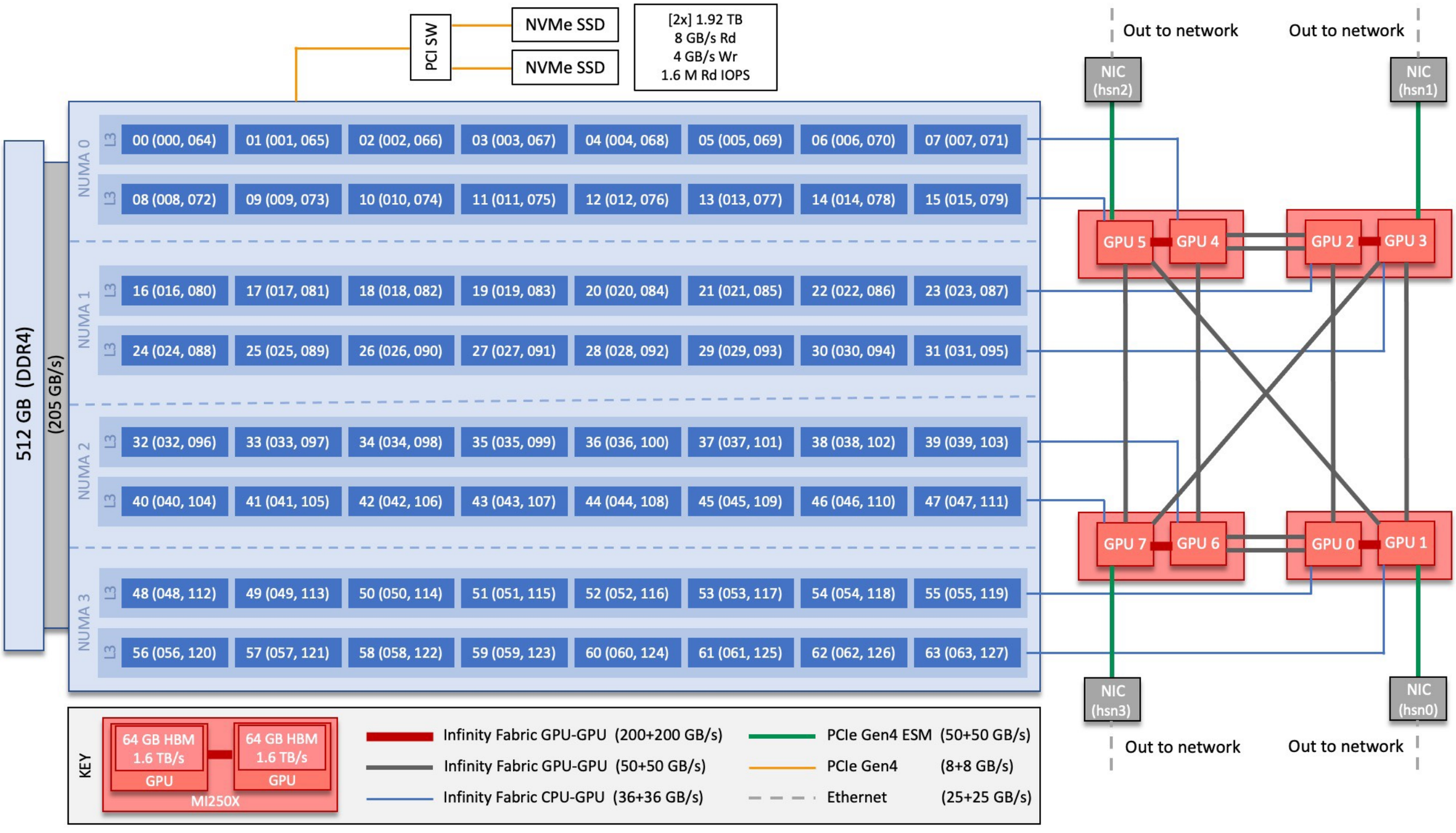
Perlmutter





- https://docs.nersc.gov/systems/perlmutter/architecture/

Crusher



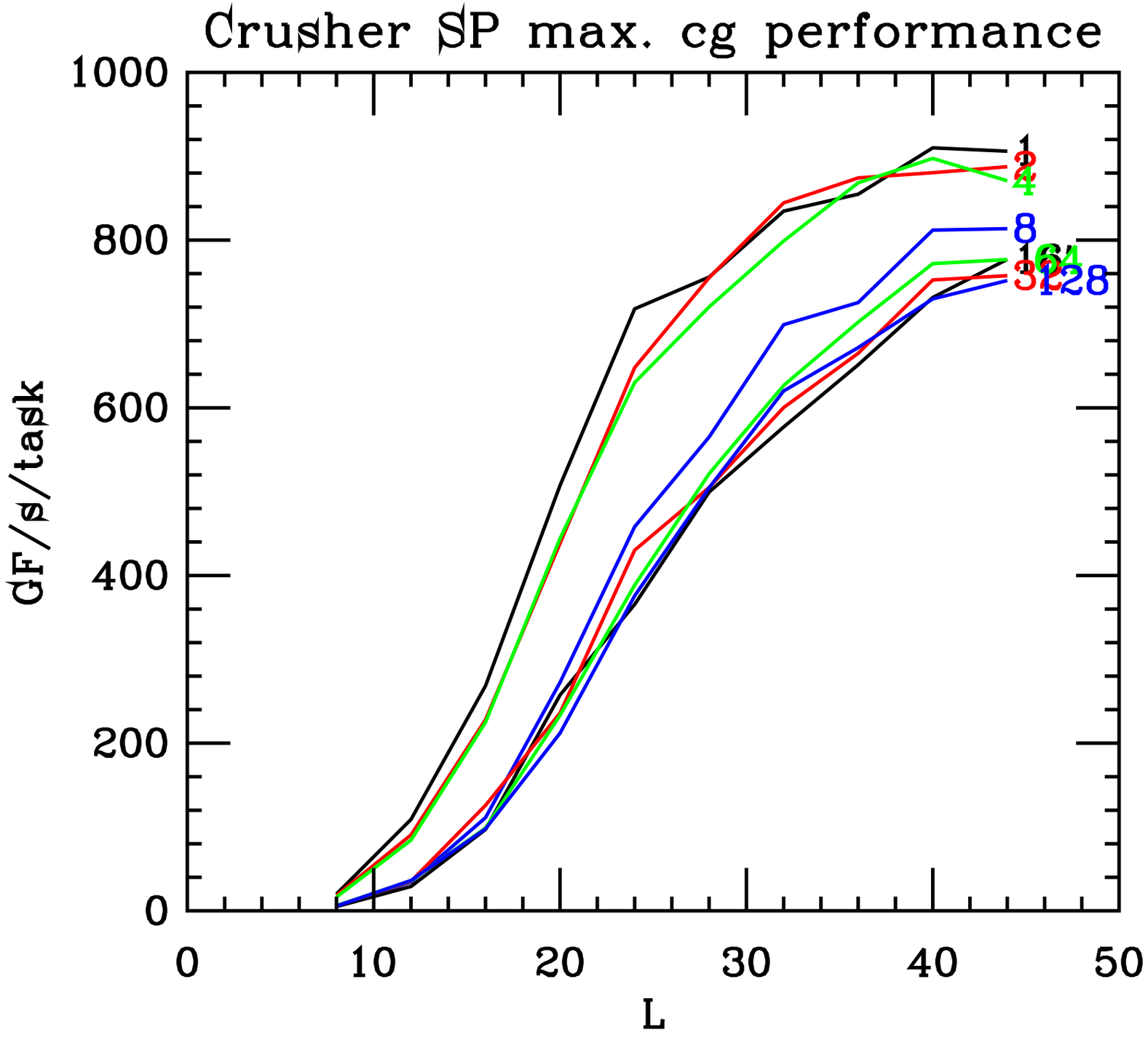
https://docs.olcf.ornl.gov/systems/crusher_quick_start_guide.html#system-overview

Crusher Results

- On Crusher have recently run volume study and some benchmarks with equilibrated configurations
  - ROCM 5.1 is current version of AMD software used
  - Can now use P2P and direct communication between GPUs and NIC. In fact, on Crusher NICS are directly connected to GPU.

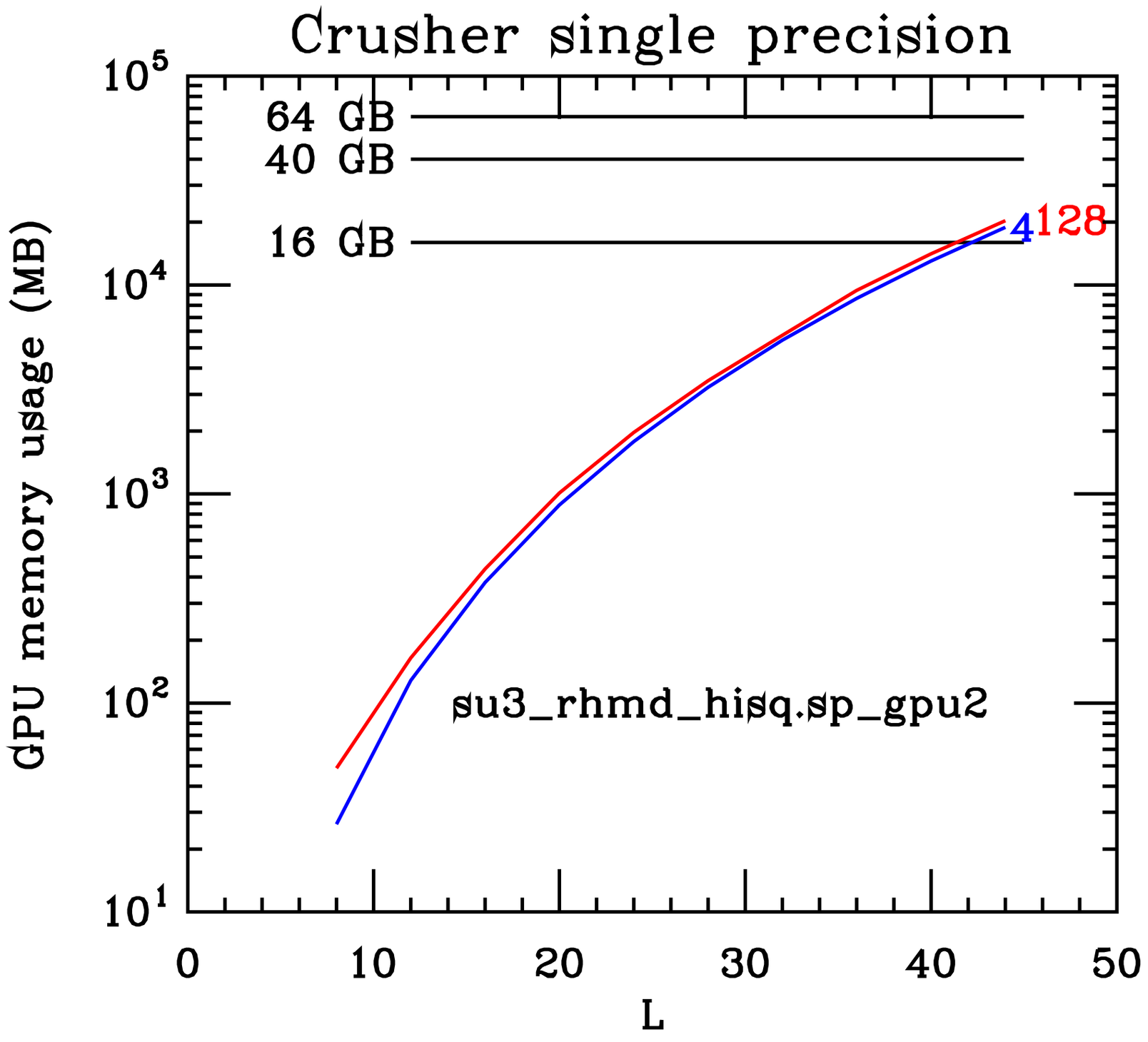
Crusher Multishift CG

- There are L^4 grid points per GCD (mpi task)
- We only show maximum performance for multishift light quark solve in each run.
- With small L, not enough work on GPU, so $L \geq 32$ desirable
- Performance fairly stable from 16-128 tasks
- approx 800 GF/GCD



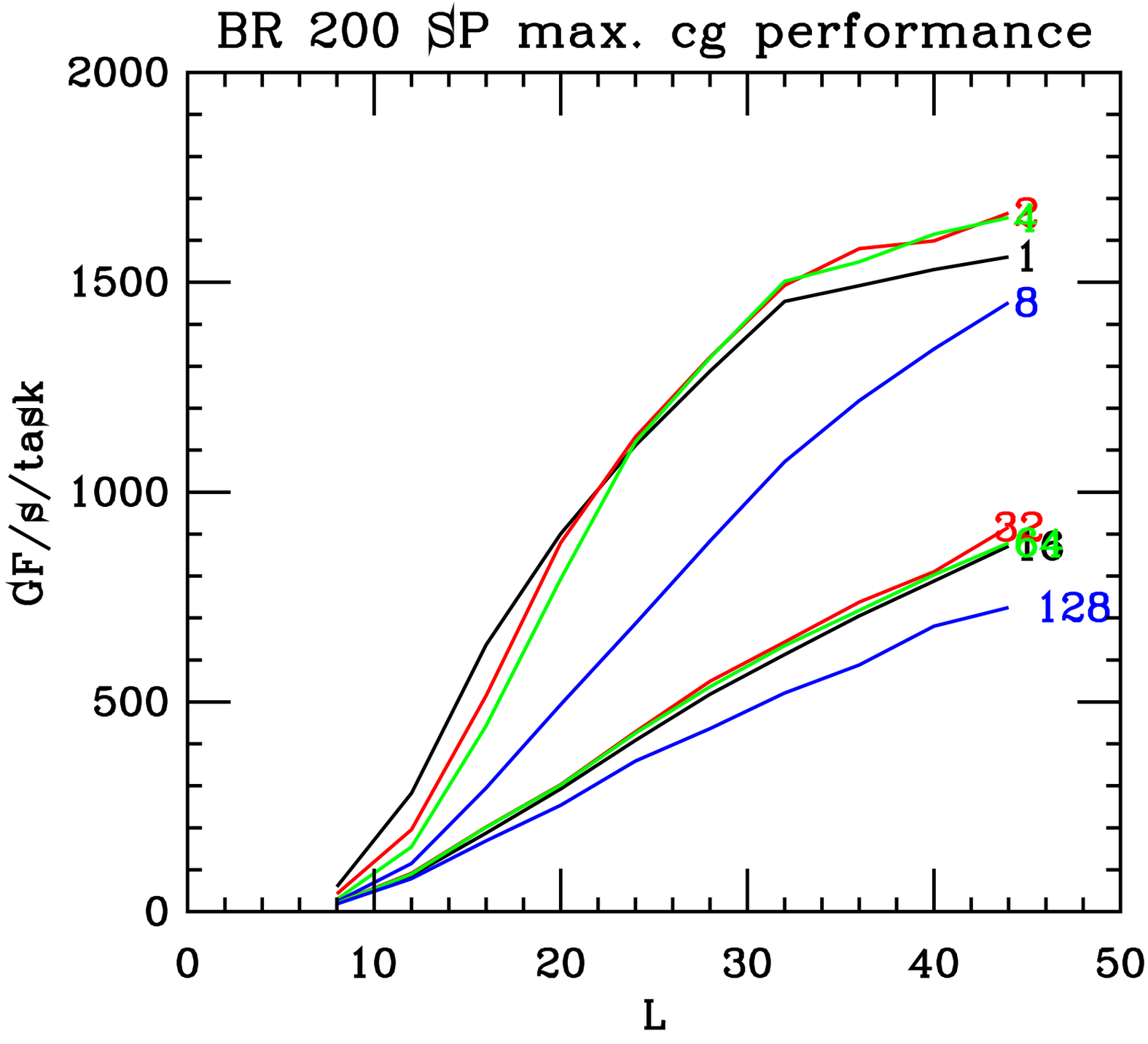
Memory Footprint

- QUDA reports how much device memory is used.
- Horizontal lines correspond to maximum HBM on V100, A100, and MI250X.
- For the latter there are two GCDs per GPU.
- Tuning gets expensive for larger volumes.



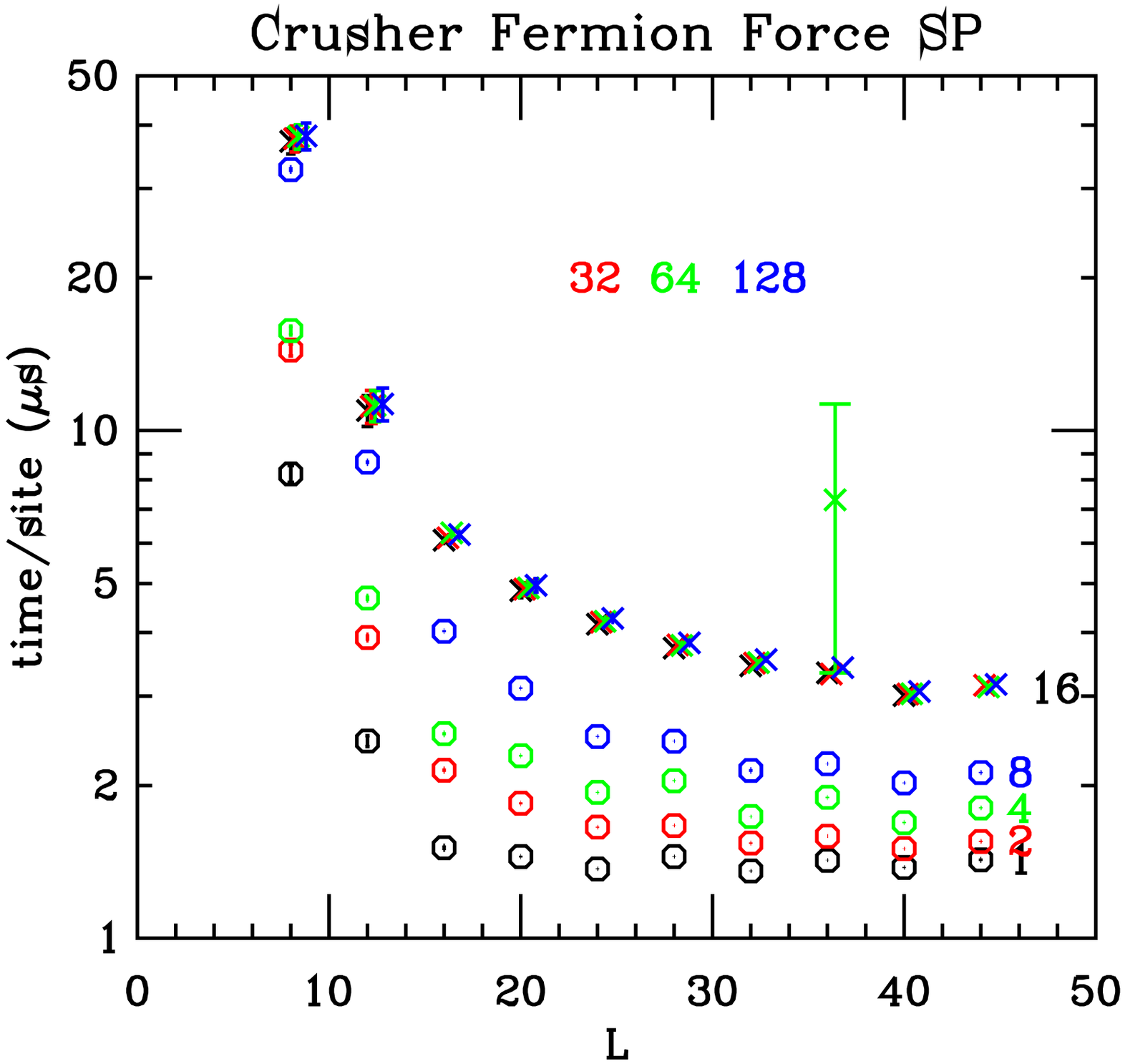
Big Red 200 Multishift CG

- For small GPU count better performance than Crusher; however, for 16 or more nodes, it appears that network is limiting.
- Probably worth doing some longer runs with higher iteration count.



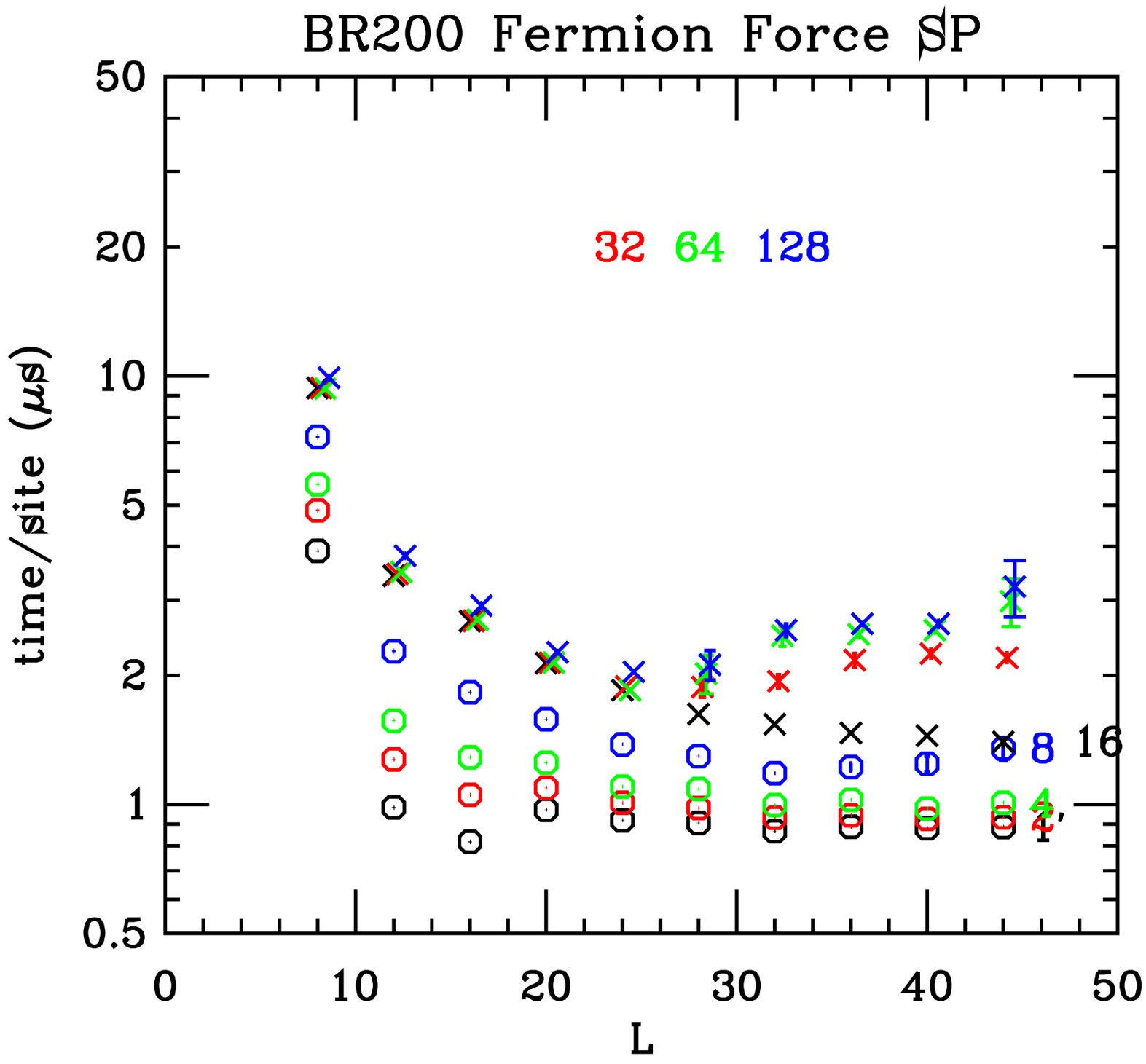
Crusher Fermion Force Time

- y-axis is time/site so small is better
- cycle through black, red, green, blue twice.
- For $L = 8$, performance is poor because there is not enough work, when communicating it gets even worse.
- For 16 or more mpi tasks performance mainly depends on local volume.



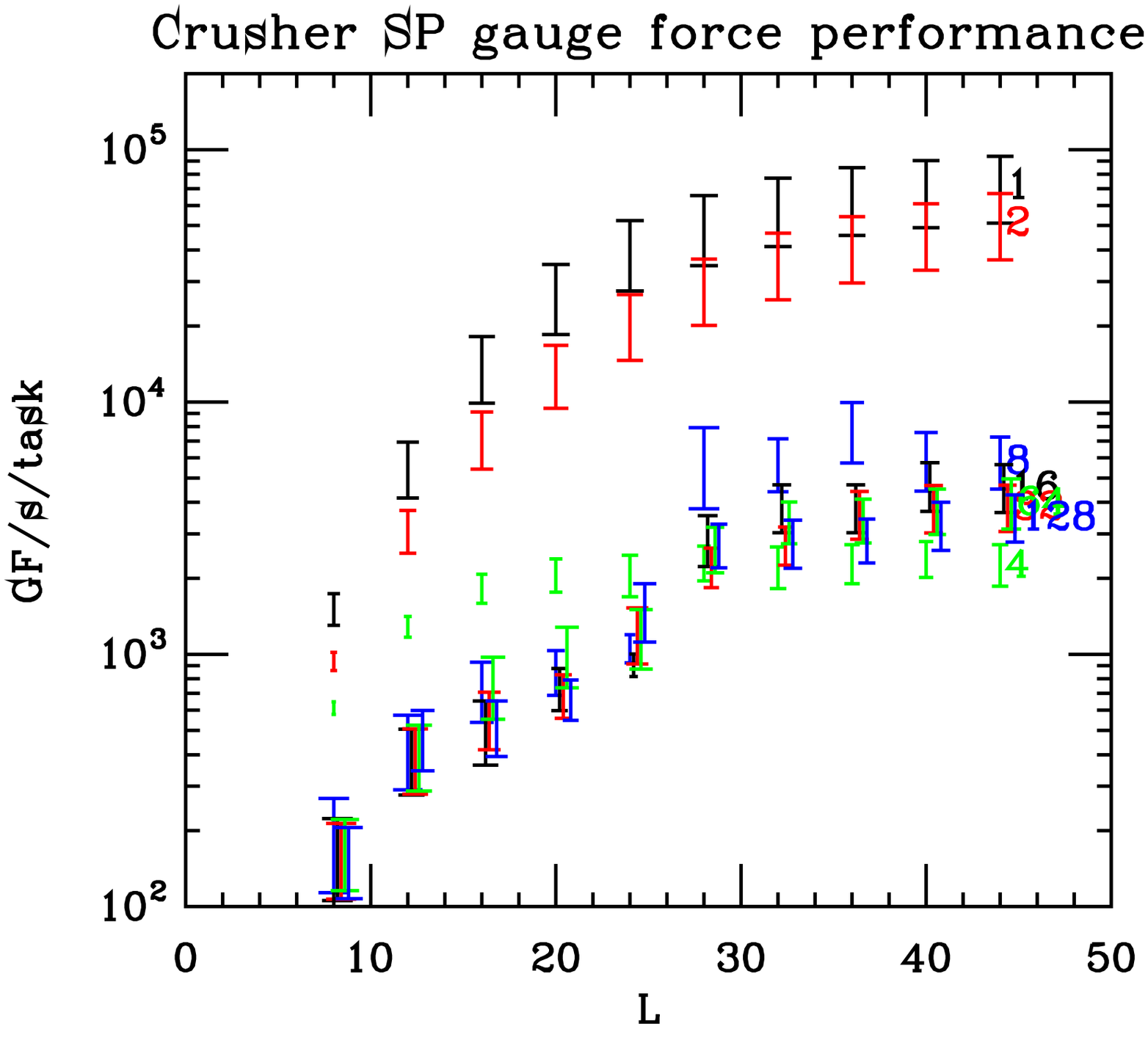
Big Red 200 Fermion Force Time

- Performance is better here than for Crusher on single node; however, for larger node count and larger values of L Crusher wins.
- Possibly due to higher network performance on Crusher.



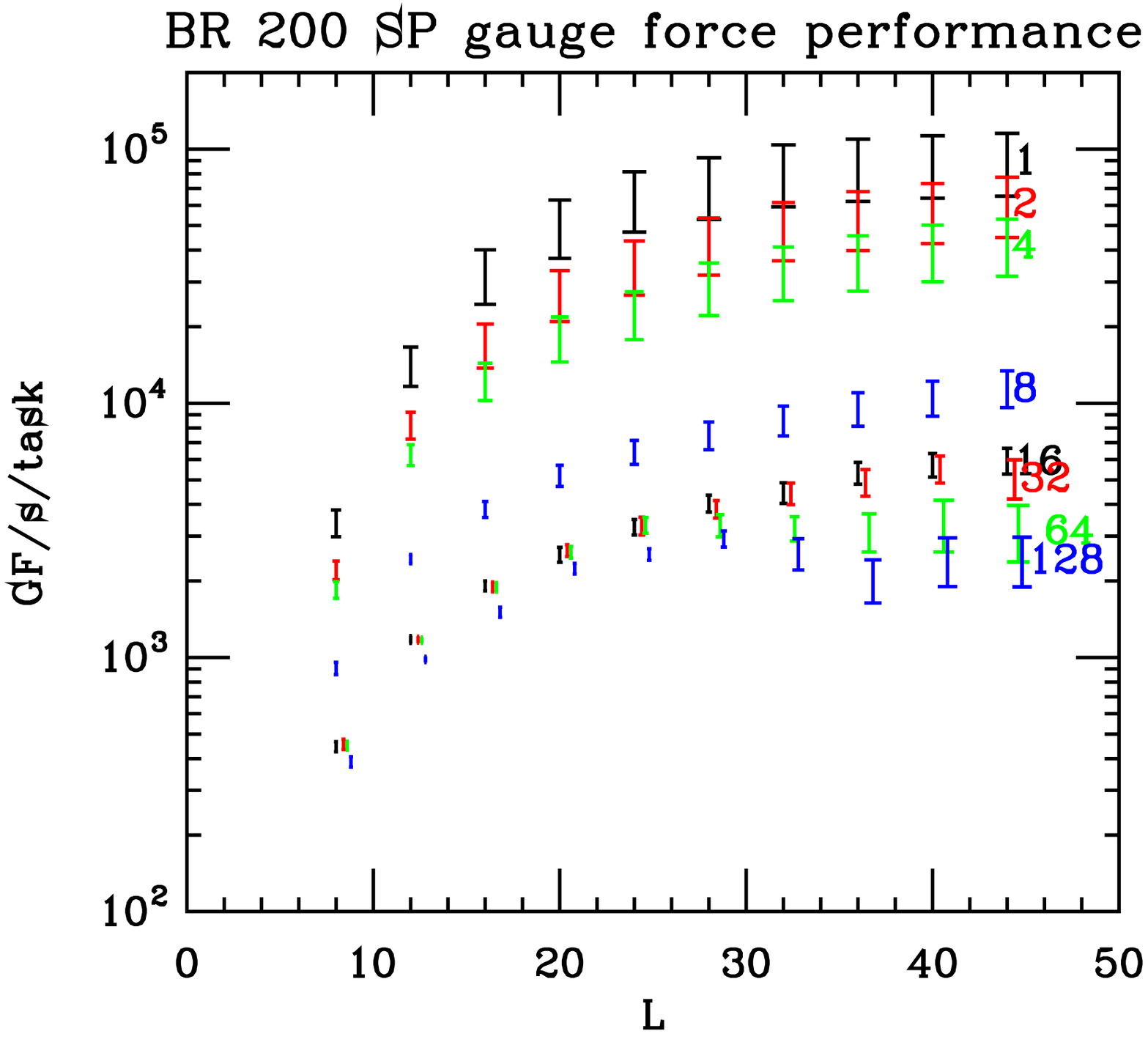
Crusher Gauge Force

- Gauge force has been optimized to avoid communication so it has a higher performance than fermion force.
- It only takes a small fraction of the time in a typical run.
- The performance is more variable during run, possibly due to reloading the gauge field. Need to investigate.



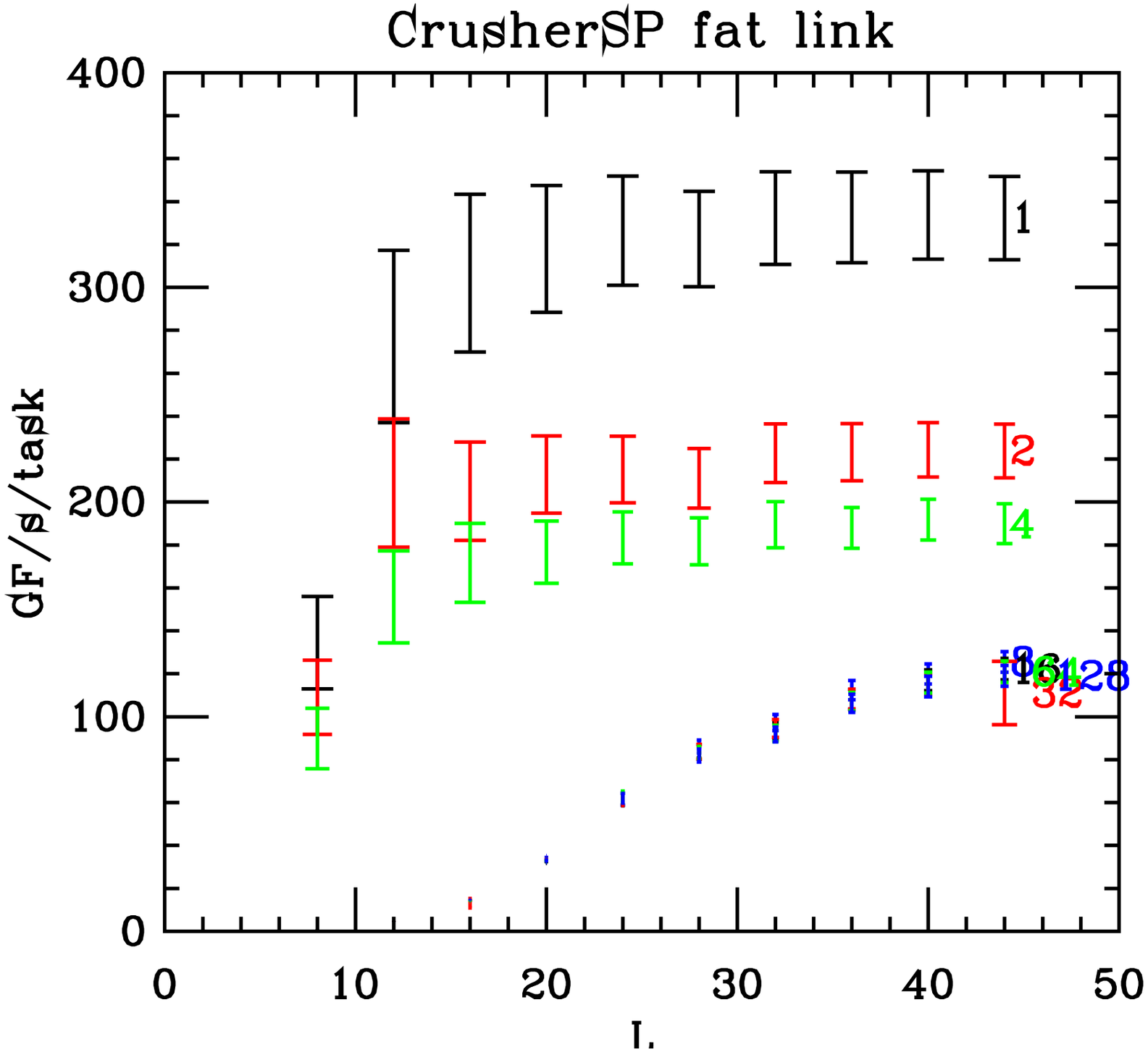
Big Red 200 Gauge Force

- The performance on BR 200 seems even more variable than on Crusher.
- Comparative benchmarks can be difficult as different cases run on different hardware.
- Times vary more than one would like on production jobs.
- See histogram of Summit run times.



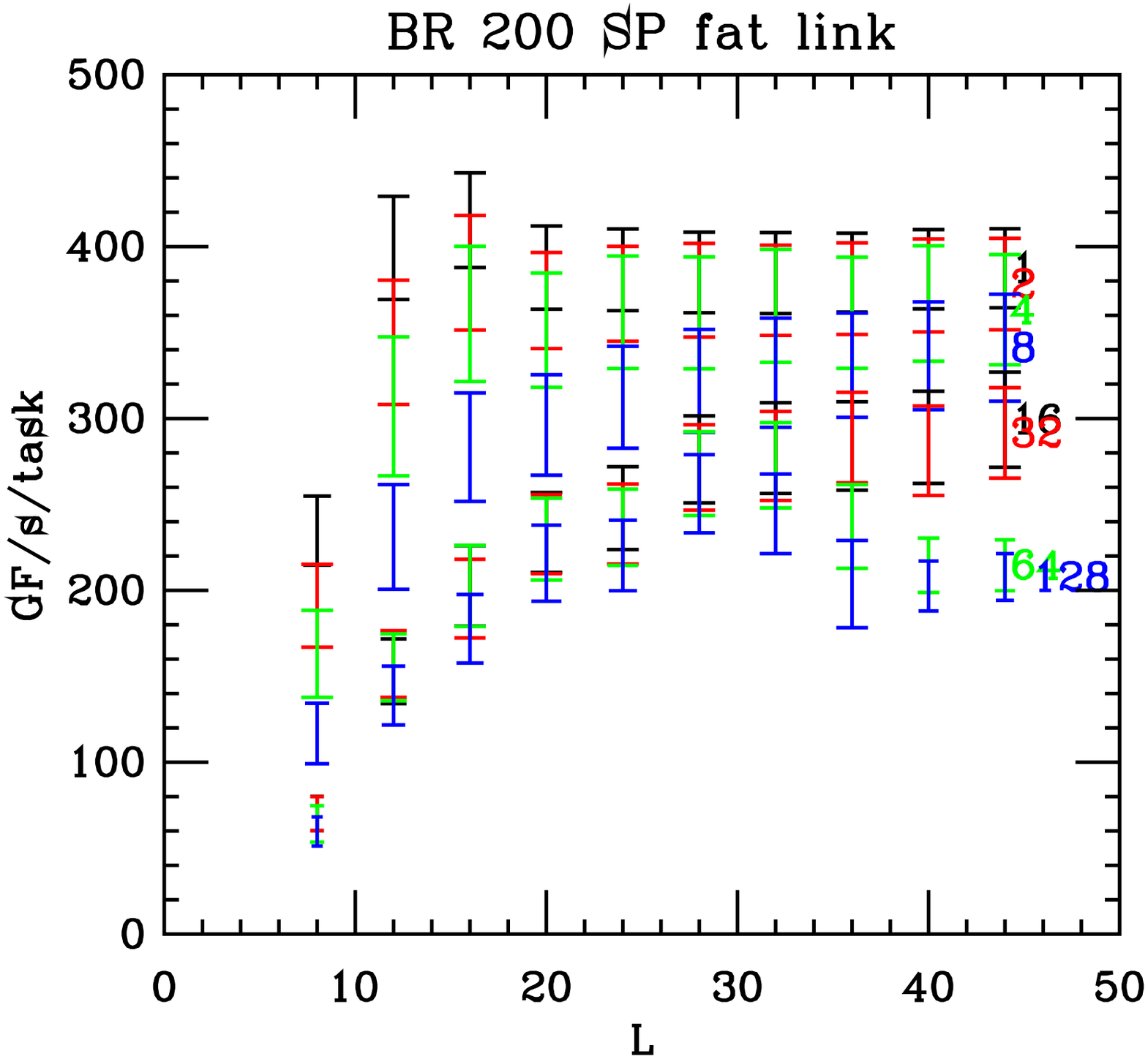
Crusher Link Fattening

- As before, $L \geq 32$ is desirable.
- Communication is frequent here.
- No new wrinkles.



Big Red 200 Link Fattening

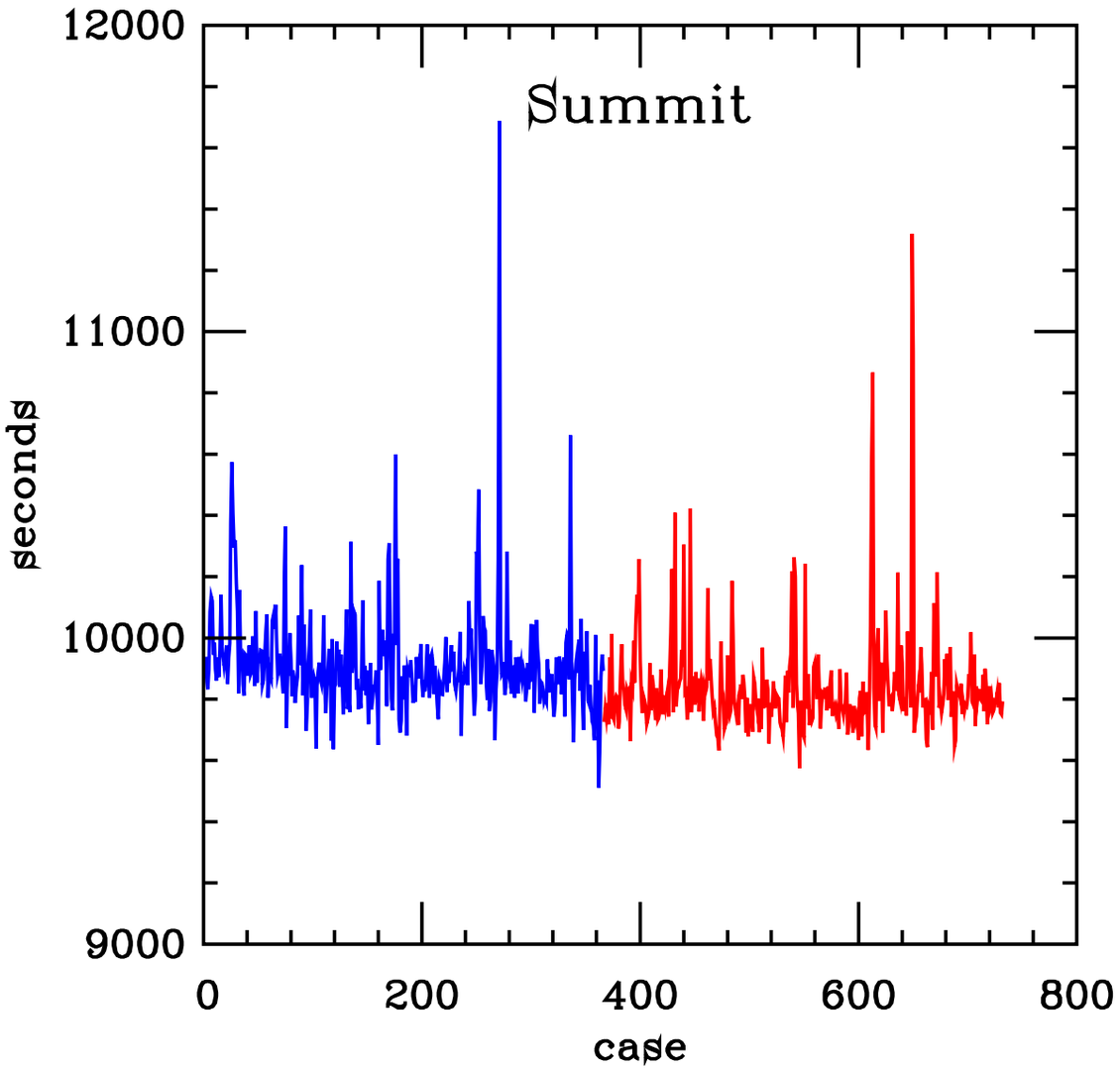
- Performance here is better than what is seen on Crusher.
- Not clear why this is the case.
- Performance may be more variable than on Crusher.
- Once again, longer runs might be useful.

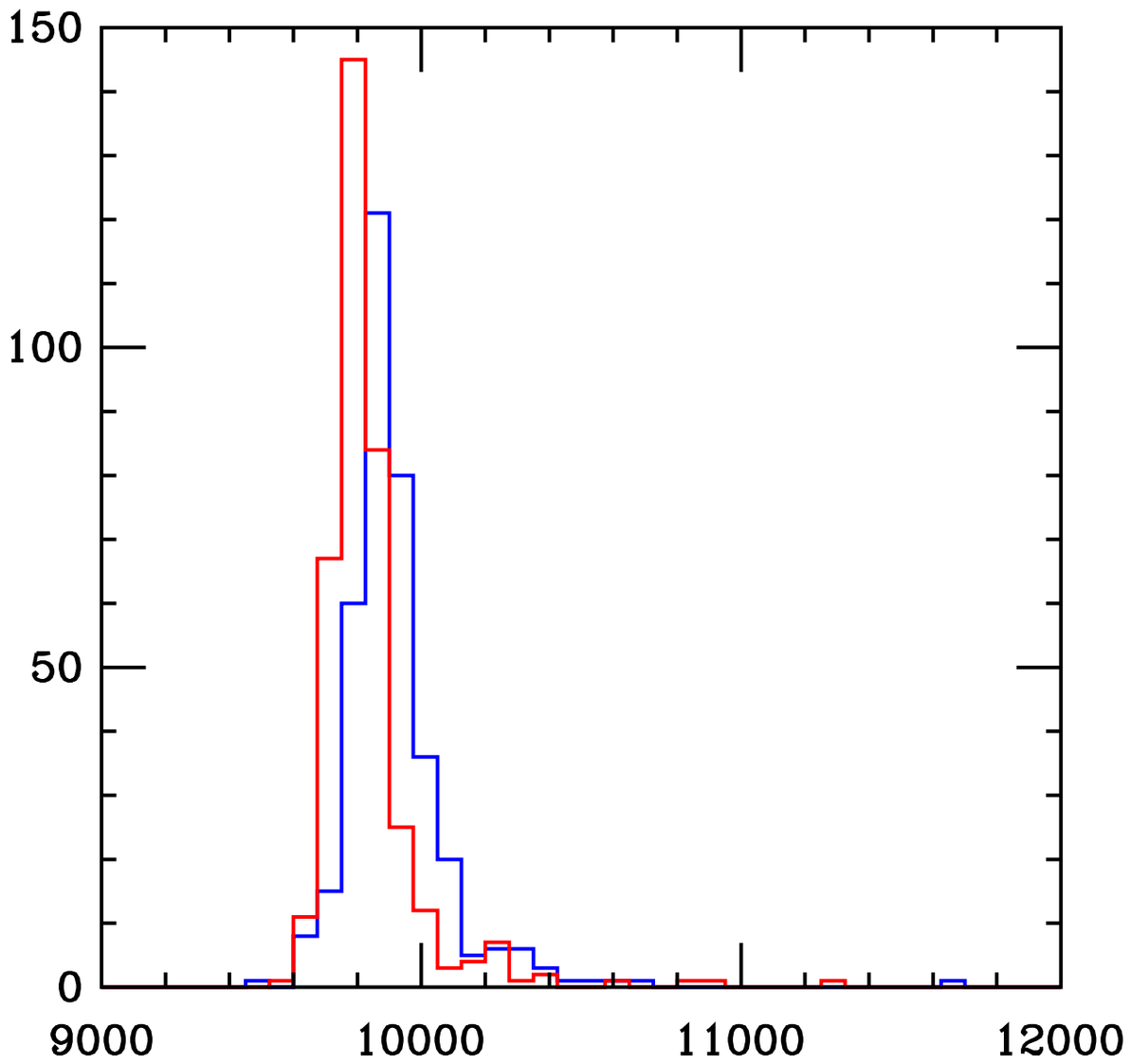


Summit

- I thought I had volume study benchmarks for Summit, but when I went looking for them, I only found a single volume of 32^4 for 1 node to 1024 nodes.
  - These runs were done in Spring, 2018
  - I used a launcher script for numa control
  - Did not use compression, P2P is enabled
  - It would not be wise to look at these results
- From Dec., 2020 have some spectrum benchmarks
  - $64^3 \times 96$ on 8 nodes, i.e., 48 GPUs with $64^2 \times 16 \times 8$ on GPU
    - single precision, single mass solve about 375 GF/GPU
  - $96^3 \times 192$ on 72 nodes, i.e., 432 GPUs with $48 \times 16^2 \times 32$
    - double precision, single mass solve about 235 GF/GPU

Performance Variability





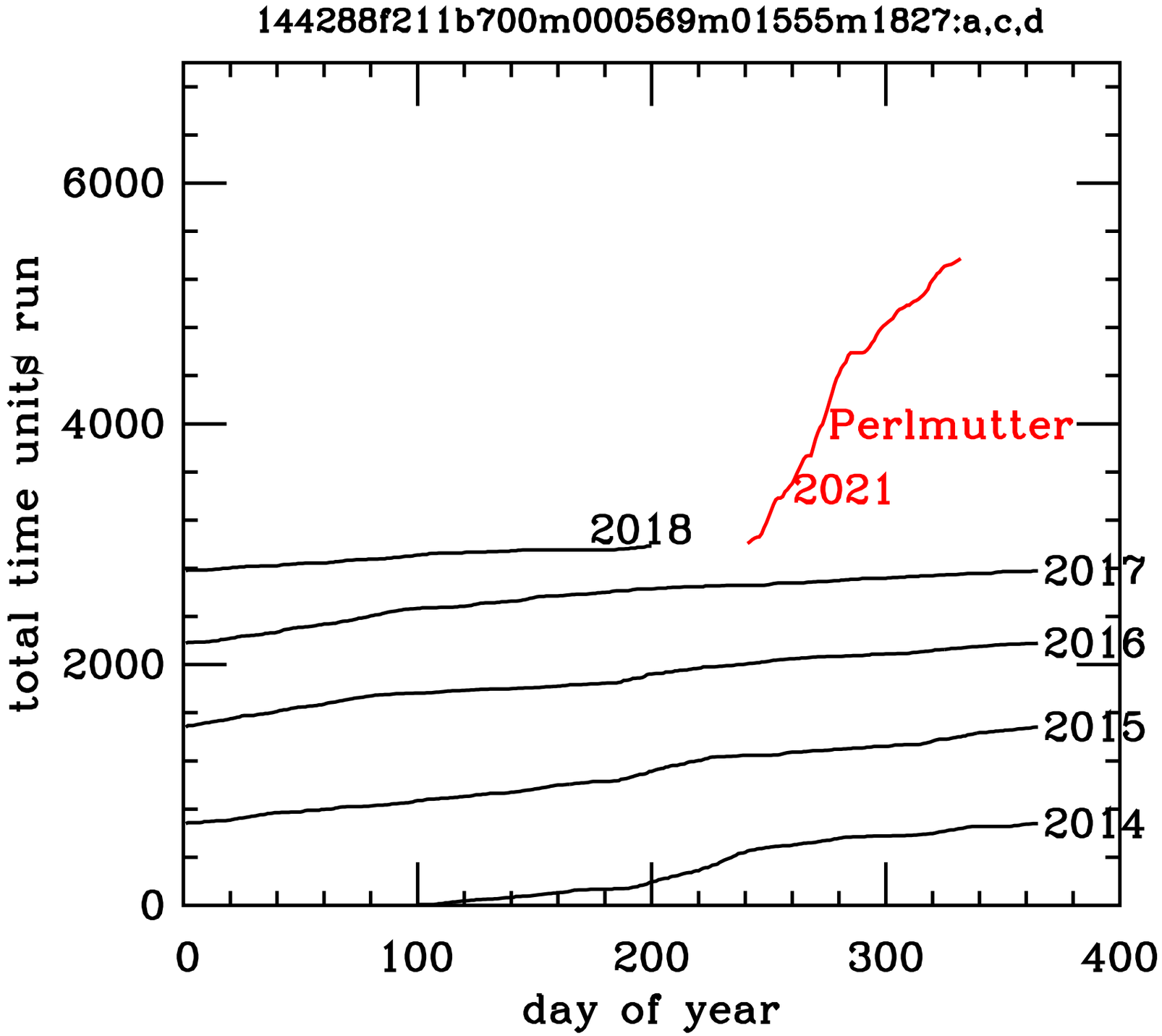
- These are from production runs on Summit average times are 9918(9) and 9836 (9). No idea why they vary so much. Shift red curve to left as both streams running at about the same time.
- Take all these benchmarks with a pile of salt!

Summit II

- Running gauge generation on Summit using 54 nodes = 324 GPUs on $96^3 \times 192$ lattice.
- Local volume $32^2 \times 16 \times 32$
- SP multishift CG with 12 masses runs at 200 GF/CPU
  - MILC compiled for mixed precision QUDA solver
- SP single mass solver (with large iteration count) runs about 260 GF/CPU
- Gauge force runs at either 640 GF or 1.0 TF per GPU
  - This probably depends on whether configuration need to be loaded, but really not sure. Must look at code.

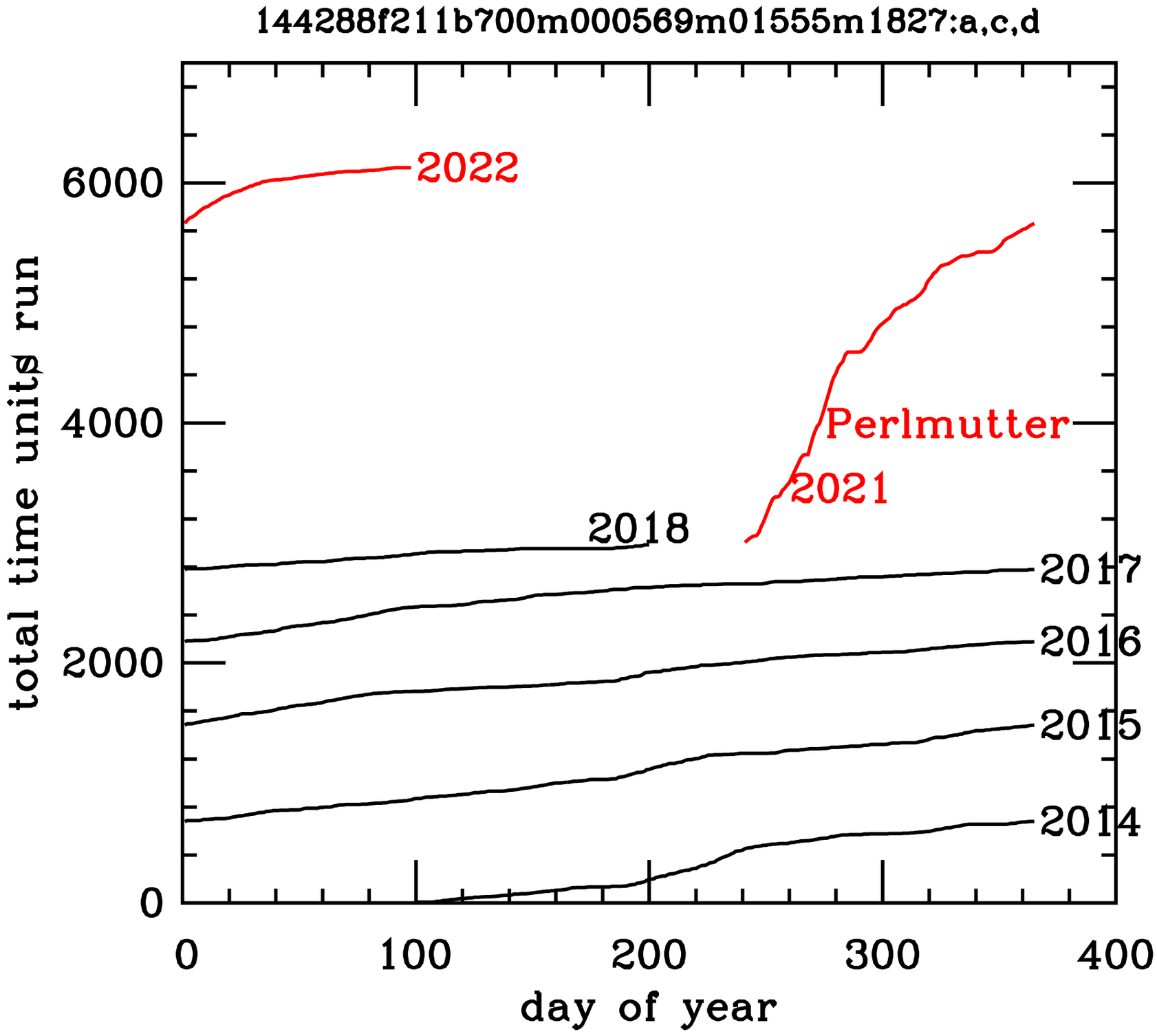
Power of Perlmutter

- Started generating configuration in 2014, by mid-2018 it was half done.
- Goal was 6000 time units.
- Late summer 2021 we were able to resume running during Perlmutter early science period.
- Note the remarkable change in slope due to power of Perlmutter.



Power of Perlmutter II

- Previous graph prepared early Dec., 2021.
- Slope decreased as Perlmutter became much busier.
- For lattice QCD, need both a fast computer and allocation to use it frequently.
- We created 500 new configurations.



Cross Platform Comparison

- Table compares times to run a trajectory of length 2 and save the configuration on four different computers

Computer	nodes or cores	MPI ranks	generate (hr)	save (hr)	total (hr)
Edison	18432 cores	36864	7.10	0.24	7.34
Cori	1024	65536	4.34	0.98	5.32
Blue Waters	1536(?)	49152	8.45	0.39	8.84
Perlmutter	128	512	1.46	0.07	1.53

Crusher with real input lattice

- $64^3 \times 96$: 4 nodes = 32 GCDs
  - local volume $64 \times 32 \times 16 \times 24$ (4.7 GB GPU mem/GCD)
  - multishift CG with 12 masses: 685 GF/GCD
  - single mass solve (approx. 5000 iters): 925 GF/GCD
  - gauge force (highly variable): up to 1.47 TF/GCD
  - 1,218 seconds for 100 step trajectory
- $64^3 \times 96$: 1 nodes = 8 GCDs
  - local volume $64^2 \times 32 \times 24$ (16.8 GB GPU memory/GCD)
  - multishift CG with 12 masses: 865 GF
  - single mass solve (approx. 5000 iters): 1.32 TF/GCD
  - gauge force (highly variable): up to 13.8 TF/GCD
  - 3,451 seconds for trajectory

Other volumes on Crusher

- I have some results on both smaller and larger volumes, up to $96^3 \times 192$
- A job is in the queue for $144^3 \times 288$ on 64 nodes.
- I did not have the time or energy to compile those results.

- Avoiding flying to Europe three months in a row was a good idea, but trying to participate remotely when talks start so early are still taking a toll.
- Thanks for recording the talks.

Conclusions

- I realized that I am not as organized as I thought I was.
- There are many options, and I have not always been consistent in what I have run at various times.
- In preparing this talk, I found a few issues that deserve more consideration. (Thanks for that!)
- GPUs using QUDA can perform very well as long as one does not make the local volume too small.
- We have time on Perlmutter for various projects and hope to get access to Frontier soon.
- These new systems have the capability to accelerate our progress. Probably, yours as well!