

QUDA-Accelerated Batched Solvers for LQCD Workflows

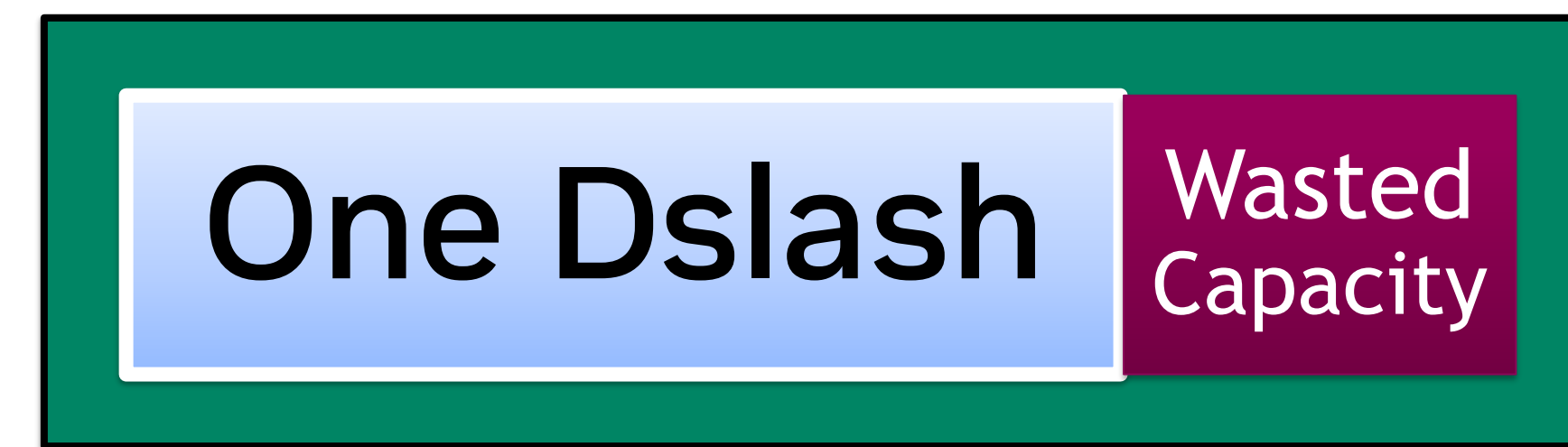
Kate Clark¹, Balint Joo¹, Jiqun Tu¹, Mathias Wagner¹, Evan Weinberg¹
 1. NVIDIA Corporation

What's Up?

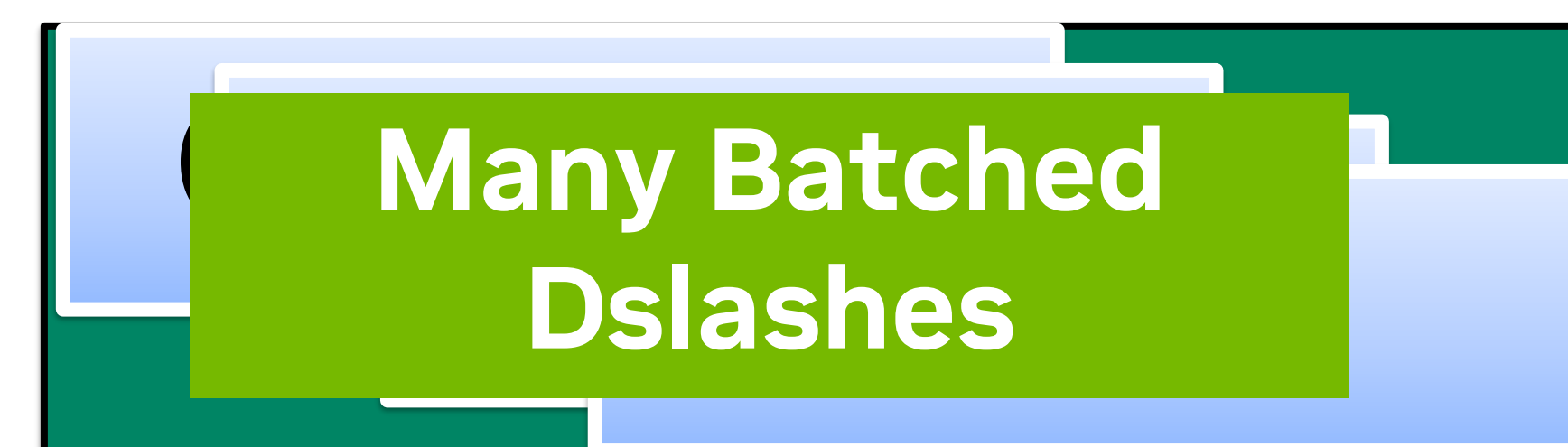
- State of the art: hundreds or thousands of propagators per gauge configuration
- Batch methods promote reuse and expose additional parallelism
- Traditional solvers, deflation methods, and multigrid methods
- QUDA can now provide drop-in multiplicative improvements in time-to-science
- Hybrid split grid/batched solves available in Chroma, prototyped in MILC

Saturation

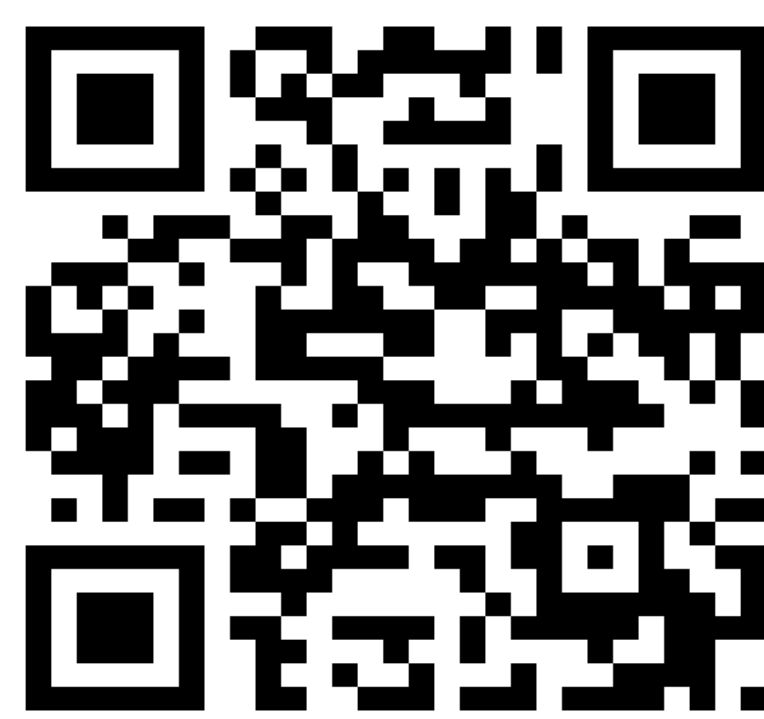
- Devices are getting wider and require more parallelism to fill the device
- Strong-scaled regime; lower levels in multi-grid solves
- Batched solvers produce embarrassingly parallel work
- Resources can be better utilized



BECOMES



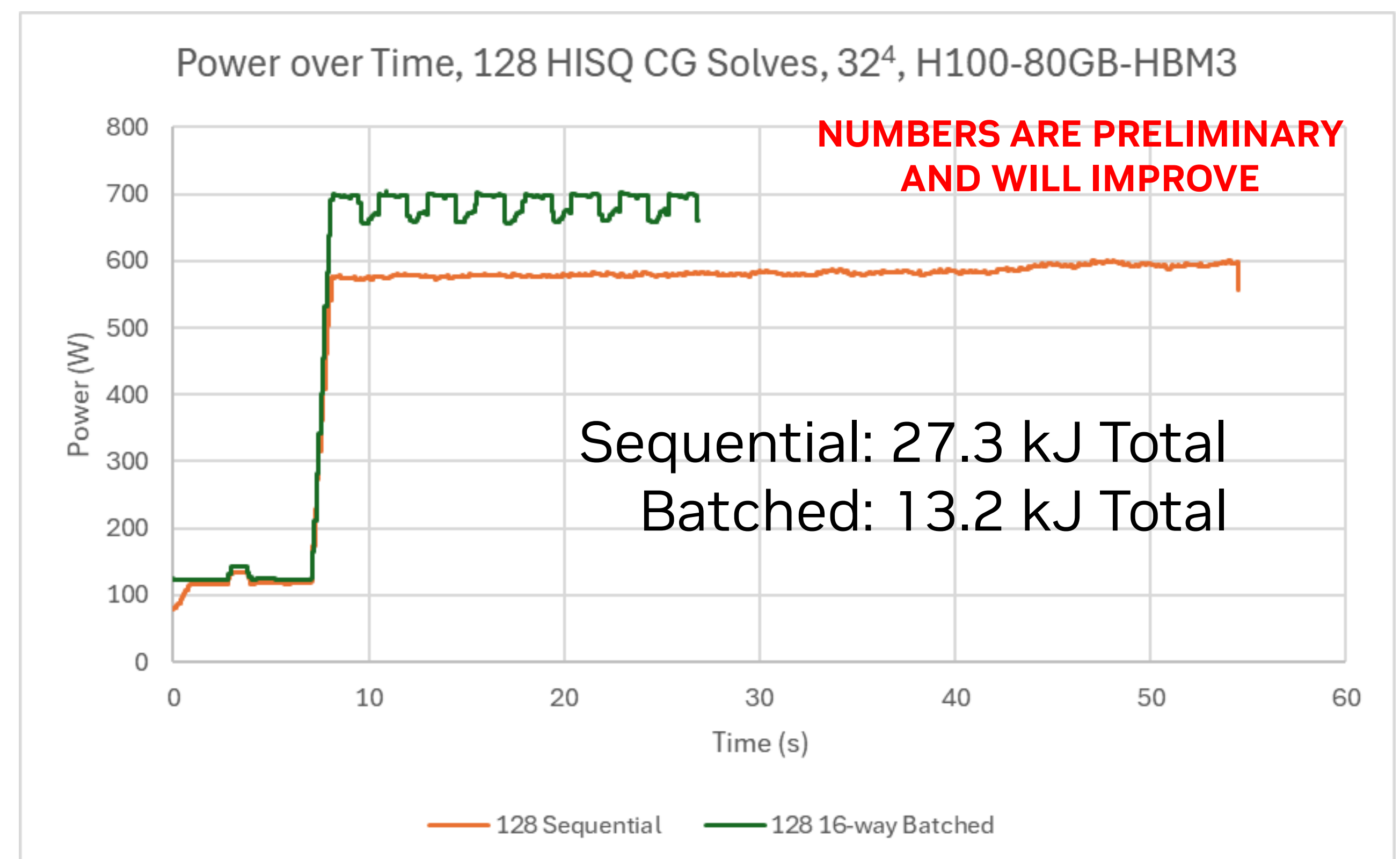
QUDA – Not Just CUDA Anymore



- <https://github.com/lattice/quda> --- Open source, BSD
- Usable in BQCD, Chroma, CPS, MILC, TIFR, and more
- Backends for CUDA, HIP, SYCL (in review)
- Solvers for major fermionic discretizations plus pure gauge algorithms and more
- Mixed-precision methods before they were cool
- Autotune and maximize performance
- **Batched solvers, deflation, multi-grid acceleration**
- NVSHMEM for improving strong scaling
- A performant algorithmic playground for exascale++

Sustainability

- Moving electrons takes energy (Physics 1)
- Batch operations enable data reuse---electrons don't need to move as far.
- Batched solves require less energy.



Composability

- Split grid: distribute propagator calculations over processor sub-domains to improve throughput
- Batch solvers: group propagator calculations to improve throughput
- Why not both?
- Optimal splitting vs batching can be a function of device vs network performance

24 sequential solves
 64 nodes
 invertQuda

BECOMES

6 batch solves
 16 nodes

6 batch solves
 16 nodes

6 batch solves
 16 nodes

6 batch solves
 16 nodes

invertMultiSrcQuda

HISQ Multigrid

- Fine level: $24^3 \times 48$; Intermediate level: $6^3 \times 8$ $N_c=64$, Coarsest level $2^3 \times 4$ $N_c=96$
- The bumps are level 1 setup, level 2 setup, solves
- Setup: 12.6 sec, 4.58 Tflops to 5.46 sec, 10.6 Tflops
- Solve throughput: 0.24 sec/solve to 0.15 sec/solve
- Saves power **and** time

