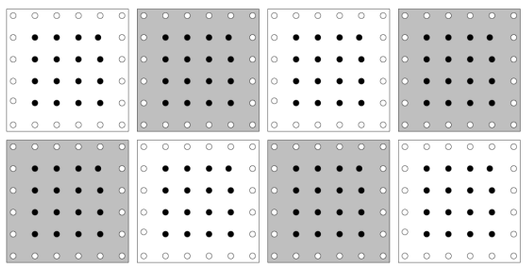


x100

Using Domain Decomposition to strong scale past 100 GPUs

Michael Clark

Harvard University

Ron Babich, Rich Brower, Balint Joo,
Steve Gottlieb and Guochun Shi

Outline

- Motivation
- Domain Decomposition
- Results
- Conclusions and Outlook

Motivation

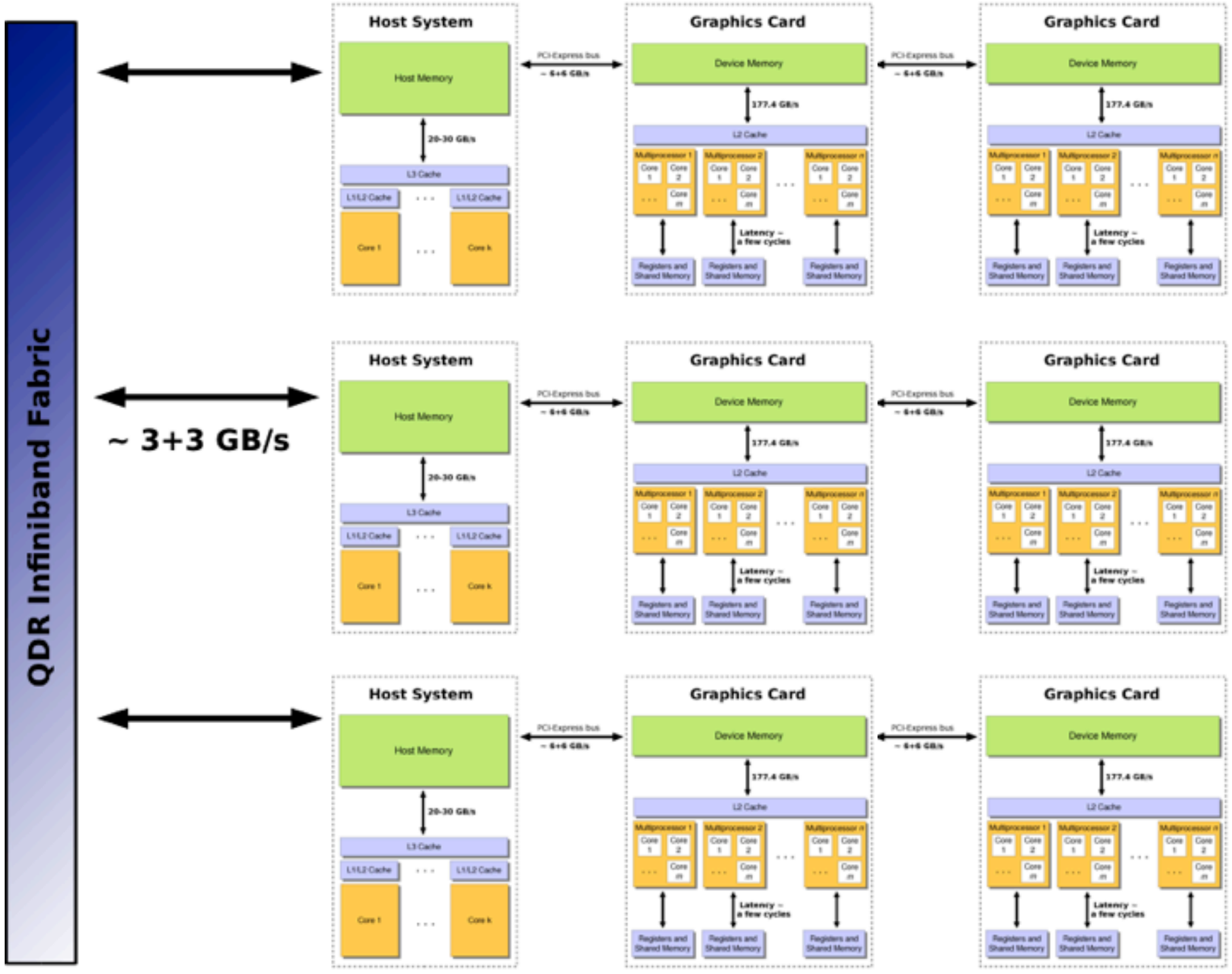
- Small scale generation and analysis calculations very efficient on small clusters of GPUs
  - Capacity computing
  - E.g., Bonati *et al*, Alexandru *et al*, etc.
- Large scale generation takes months running at O(10) Tflops
  - Capability computing
  - Current computations use BG / Cray etc.
- Can GPU clusters make headway in this regime?

(Clusters dedicated to lattice QCD at Fermilab and Jefferson Lab)

"Intrepid" - Argonne Leadership Computing Facility

"Jaguar" - Oak Ridge Leadership Computing Facility

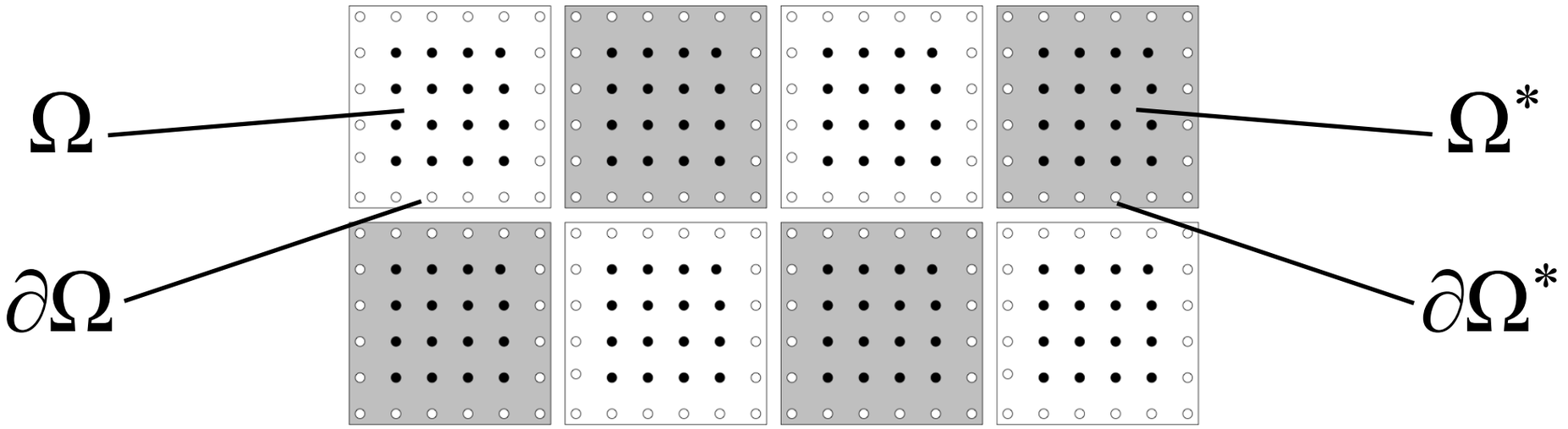
Memory Hierarchy



Domain Decomposition

- Only way to get to O(1000) GPUs is with communication reducing algorithms
- Schwarz Methods aka Domain Decomposition
    - Partition the lattice into domains
    - Solve each of the domains independently
    - Use domain solutions to correct full system
- Typically used a preconditioner
- Previous work in LQCD
    - Lüscher: SAP (multiplicative Schwarz) on CPU clusters
    - Osaki and Ishikawa: additive Schwarz for GPU scaling up to 8 GPUs
    - Nakamura *et al*: SAP on QPACE, ~12 Tflops at 256 Cells

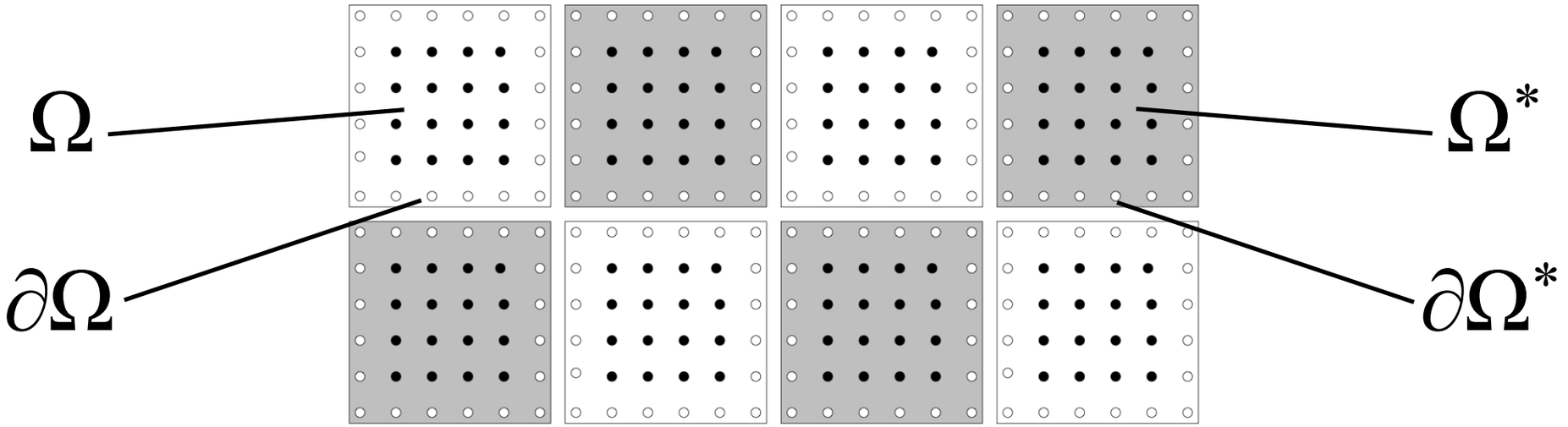
Multiplicative Schwarz



- Update domains sequentially and use most recent solutions Block Gauss Seidel
- Coloring algorithms used to parallelize, **but this means either**
  - Multiple colors per processor
  - Idling processors
- Preconditioner given by

$$K = D_{\Omega}^{-1} + D_{\Omega^*}^{-1} + D_{\Omega^*}^{-1} D_{\partial\Omega^*} D_{\Omega}^{-1}$$

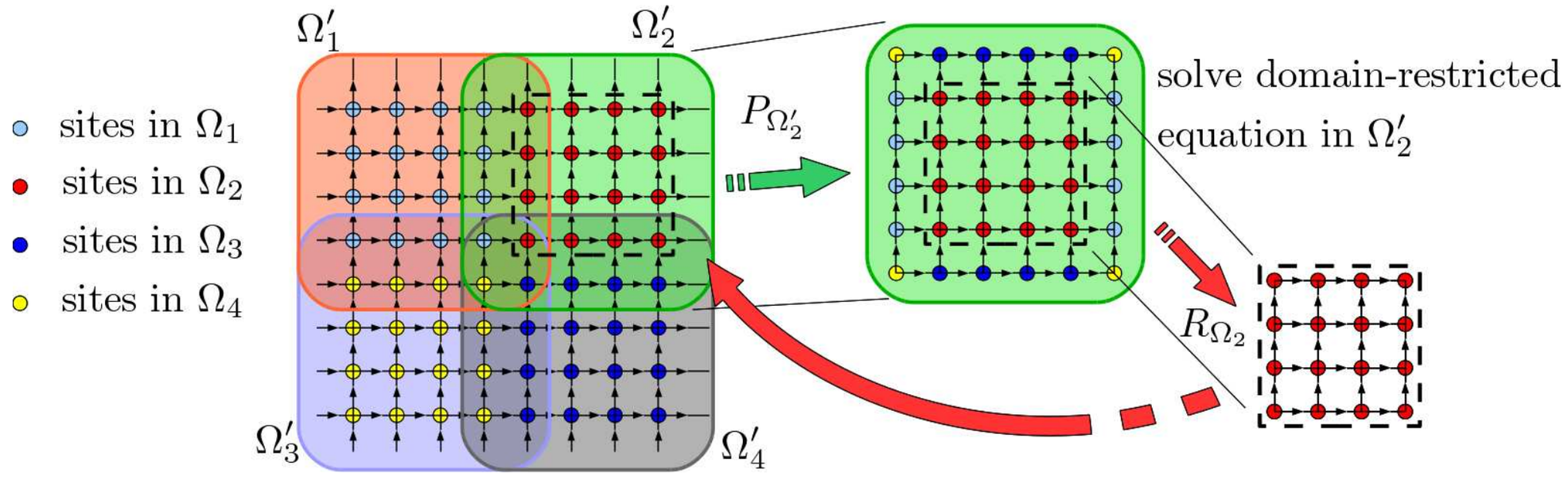
Additive Schwarz



- Update domains simultaneously Block Jacobi
- Expect higher iteration count than multiplicative
- Algorithm is trivially parallel
  - Better suited to GPU architecture
- Preconditioner given by

$$K = D_{\Omega}^{-1} + D_{\Omega^*}^{-1}$$

Restricted Additive Schwarz (Cai and Sarkis)



- Initial domain residuals given by the restriction of full system residual
- Impose Dirichlet boundary conditions at domain boundaries
- Solve all domain systems simultaneously
- Full system correction given by the sum of the domain solutions

figure taken from Osaki and Ishikawa

Implementing the Preconditioner

- Non-overlapping domains only
- Domain volume = local GPU volume
    - Solver iteration count varies with N_{GPU}
- Very easy to implement
    - Switch off inter-GPU communication for restricted Dirac operator
- 10 steps of Minimum residual (MR) used to solve for preconditioner
- Preconditioner is a gross approximation
    - Only need 16-bit precision

Mixed Precision GCR-DD solver

Apply Preconditioner: reduced precision

(Re)Start

$r_0 = b - Mx$

$\hat{r}_0 = r_0$

$\hat{x}_0 = 0$

$k = 0$

Generate Subspace

$\hat{p}_k = \hat{K}^{-1}\hat{r}_k$

$\hat{z}_k = \hat{M}\hat{p}_k$

Reduced Precision M v

$\beta_{i,k} = (\hat{z}_i, \hat{z}_k)$

Orthogonalize $\hat{z}$-s

$\gamma_k = ||\hat{z}_k||$

normalize $\hat{z}_k$

$\alpha_k = (\hat{z}_k, \hat{r}_k)$

$\hat{r}_{k+1} = \hat{r}_k - \alpha_k \hat{z}_k$

$k = k + 1$

repeat for all k or until residuum drops enough or convergence

Update Solution

Solve for χ_l l=k,k-1,...,0:

$$\gamma_l \chi_l \sum_{i=l+1}^{k} \beta_{l,i}\chi_l = \alpha_l$$

Compute correction for x:

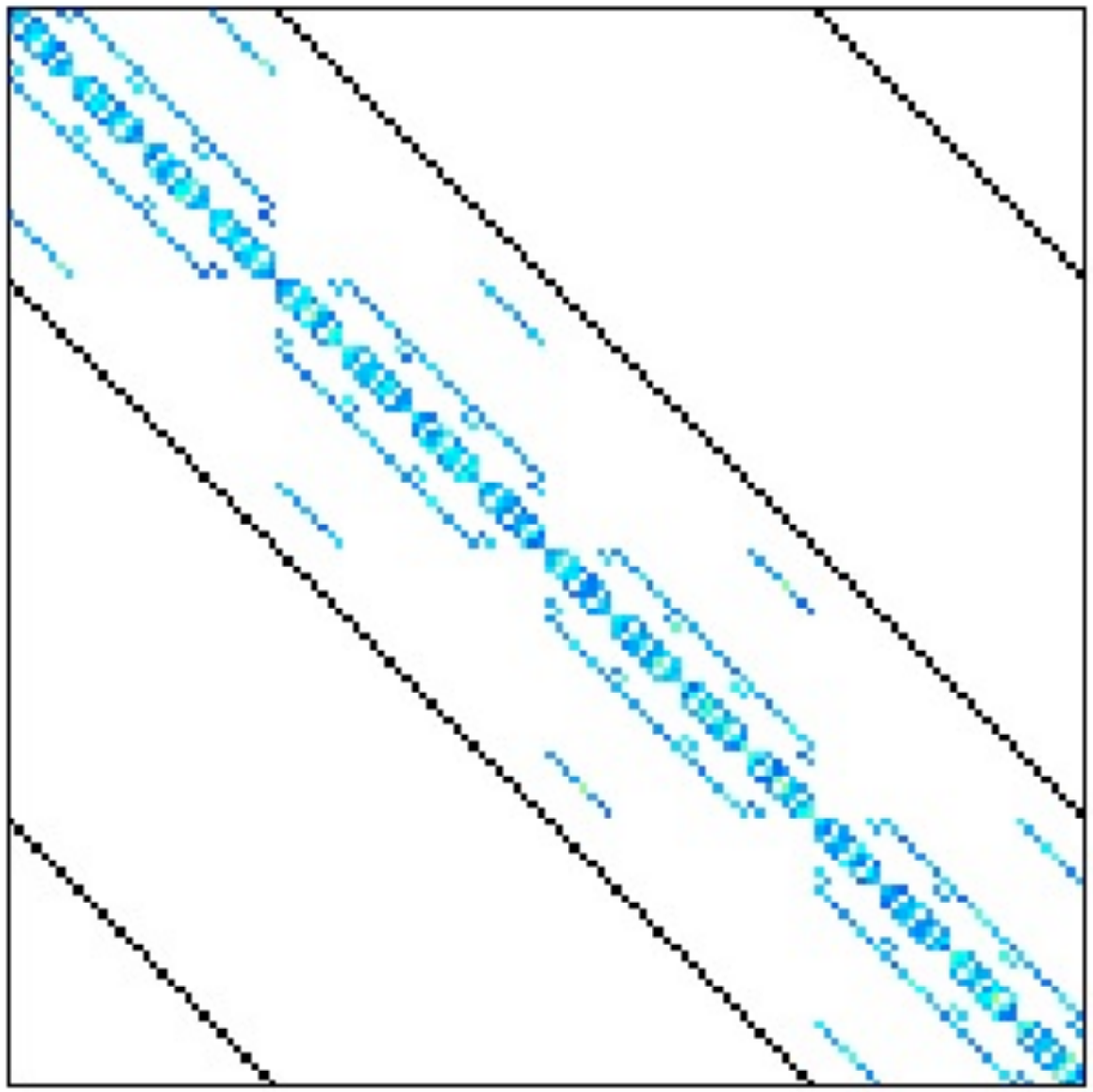
$$\hat{x} = \sum_{i=0}^{k-1} \chi_i p_i$$

$x = x + \hat{x}$

Full precision restart if not converged

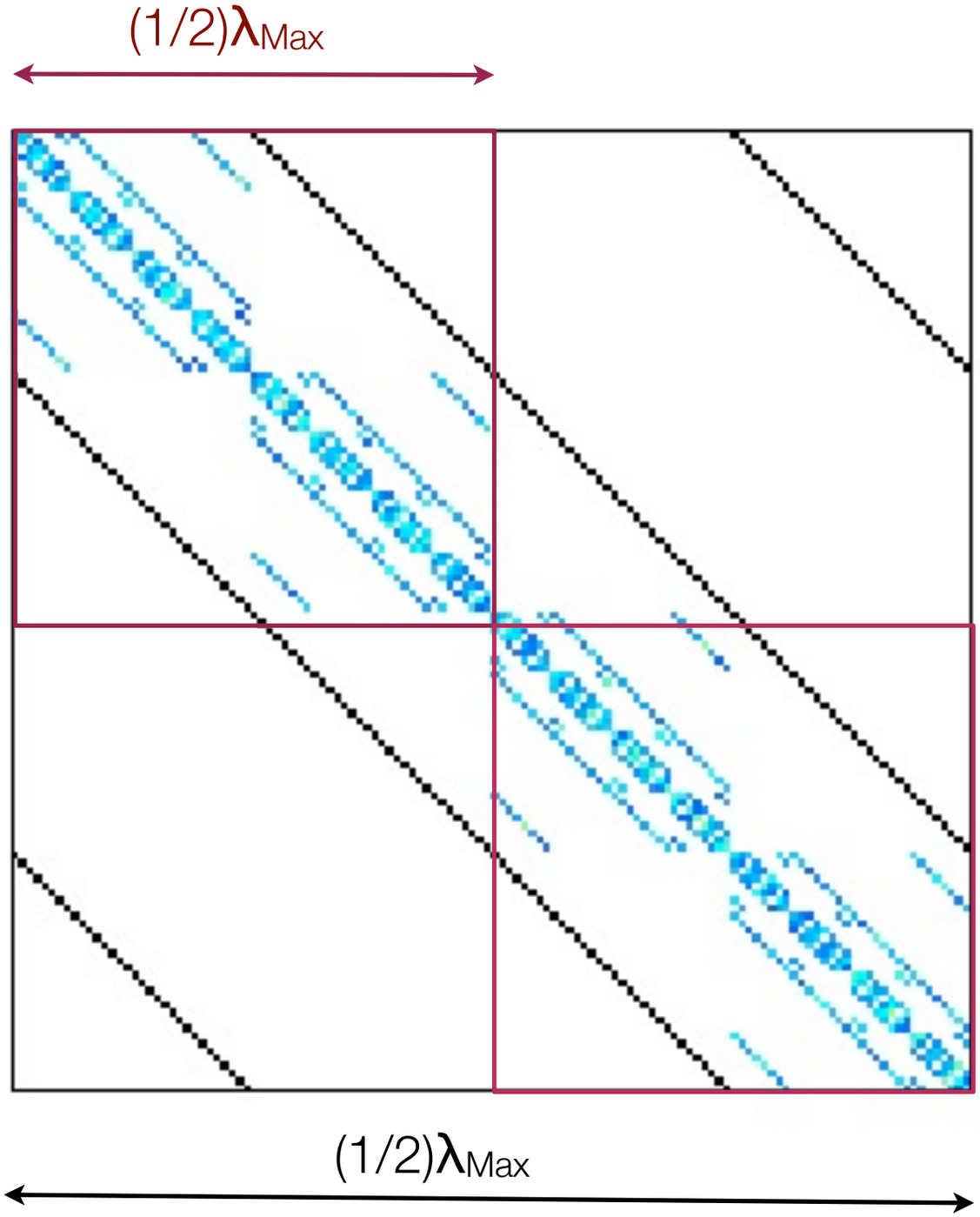
Quantities with ^ are in reduced precision

Size Matters



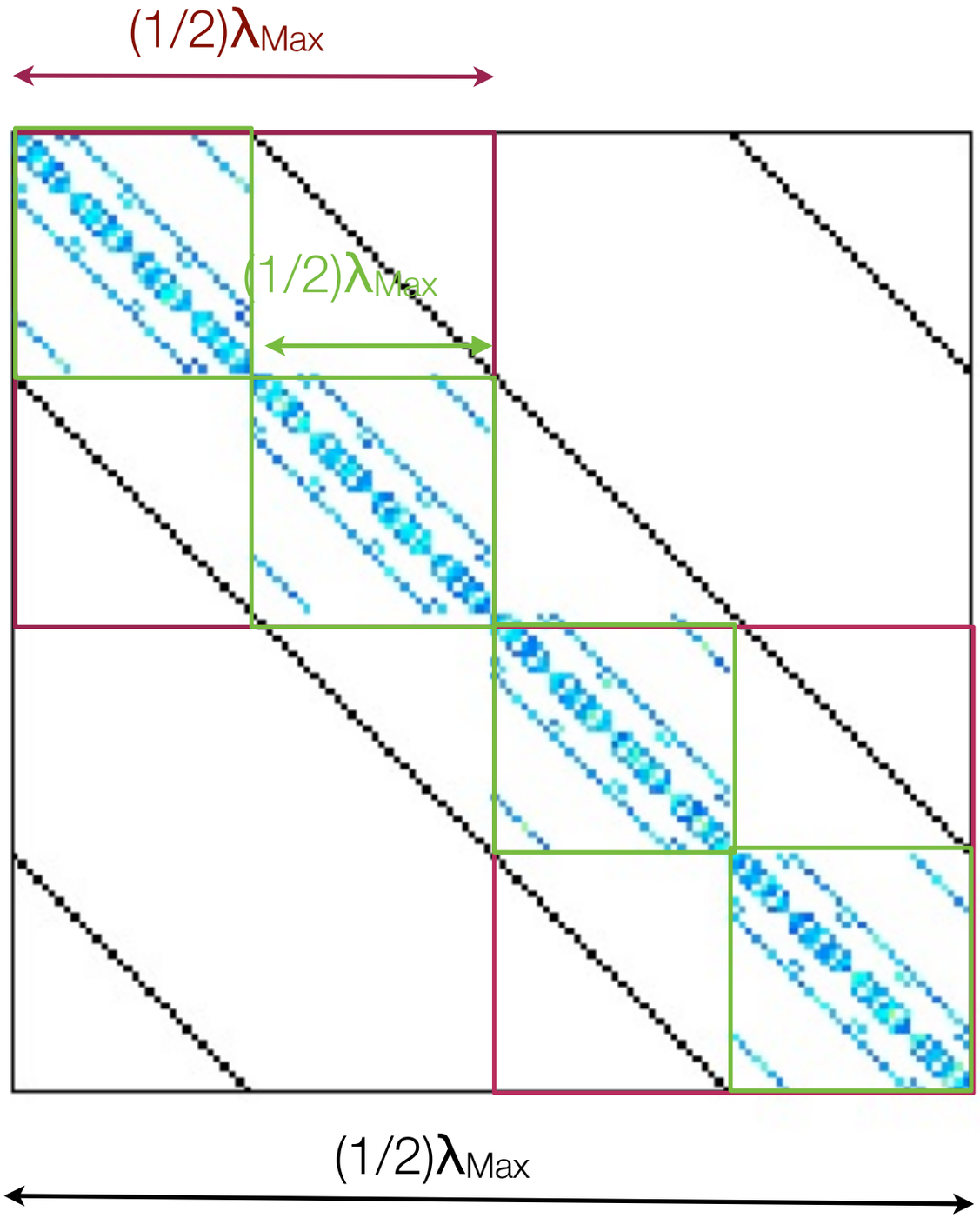
- No comms between domains
  - Block Diagonal Preconditioner
- Blocks impose λ cutoff
- Finer Blocks
  - lose structure in operator
  - lose long wavelength/low energy modes
- Heuristically (& from Lüscher)
  - keep wavelengths of ~ $O(\Lambda_{QCD}^{-1})$
  - Λ_{QCD}^{-1} ~ 1fm
  - 32^3x256 Aniso: (a_s=0.125fm, a_t=0.035fm)
    - Our case: 8^3x32 blocks are ideal
  - Iso: 1fm ~ 8-10 sites (a=0.11fm)
- Min. blocksize has scaling implications

Size Matters



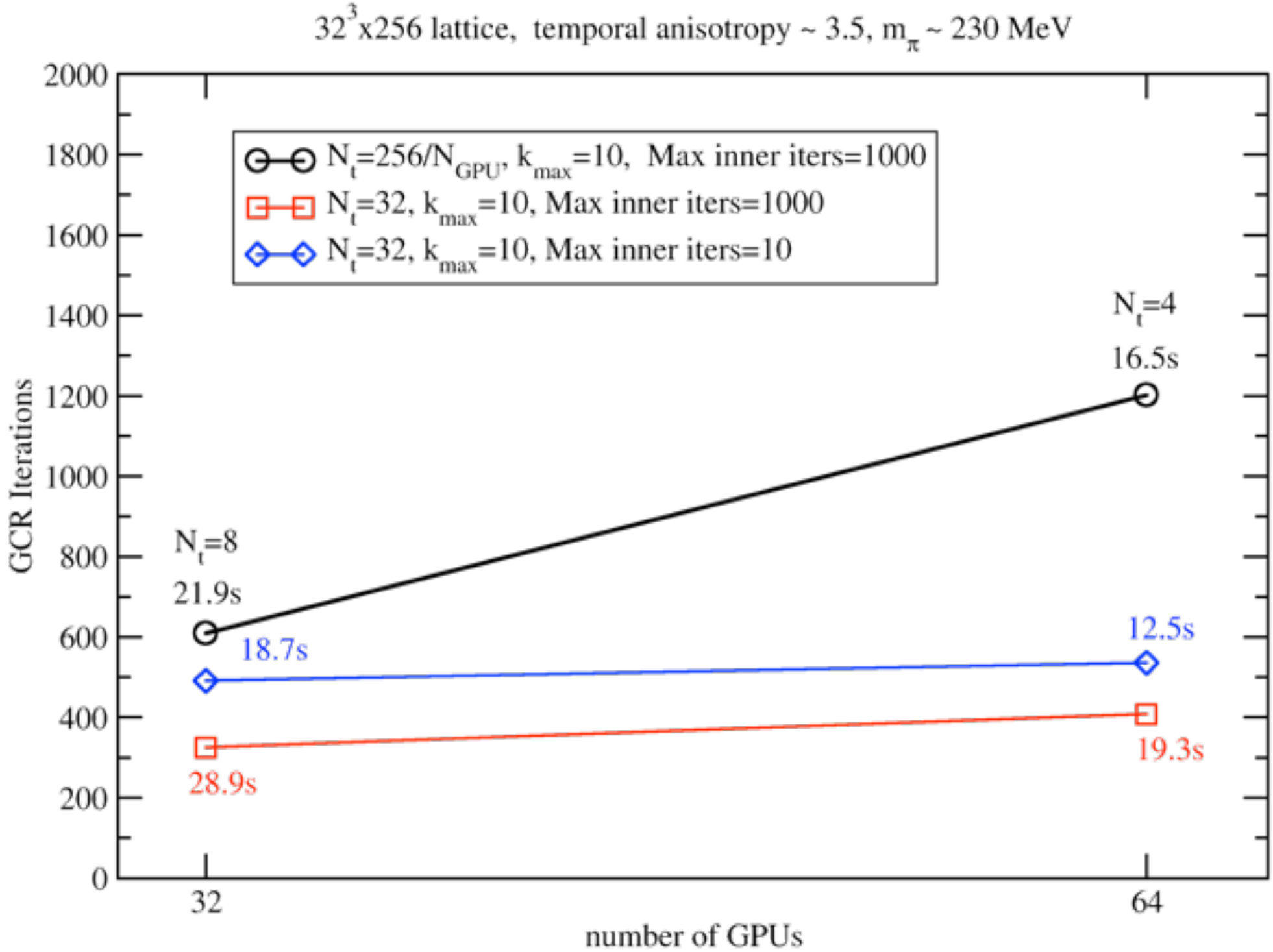
- No comms between domains
  - Block Diagonal Preconditioner
- Blocks impose λ cutoff
- Finer Blocks
  - lose structure in operator
  - lose long wavelength/low energy modes
- Heuristically (& from Lüscher)
  - keep wavelengths of ~ $O(\Lambda_{QCD}^{-1})$
  - Λ_{QCD}^{-1} ~ 1fm
  - 32^3x256 Aniso: (a_s=0.125fm, a_t=0.035fm)
    - Our case: 8^3x32 blocks are ideal
  - Iso: 1fm ~ 8-10 sites (a=0.11fm)
- Min. blocksize has scaling implications

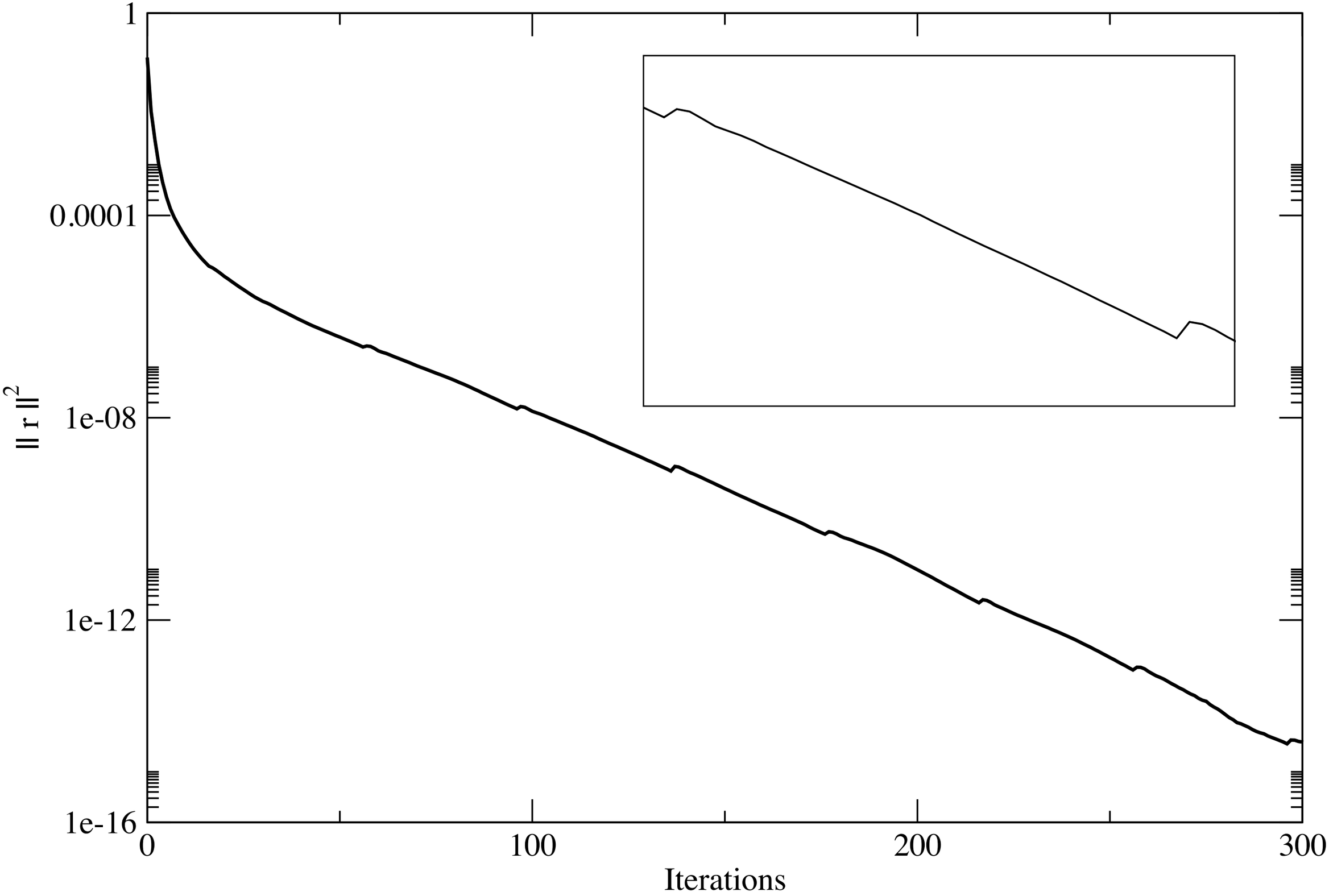
Size Matters

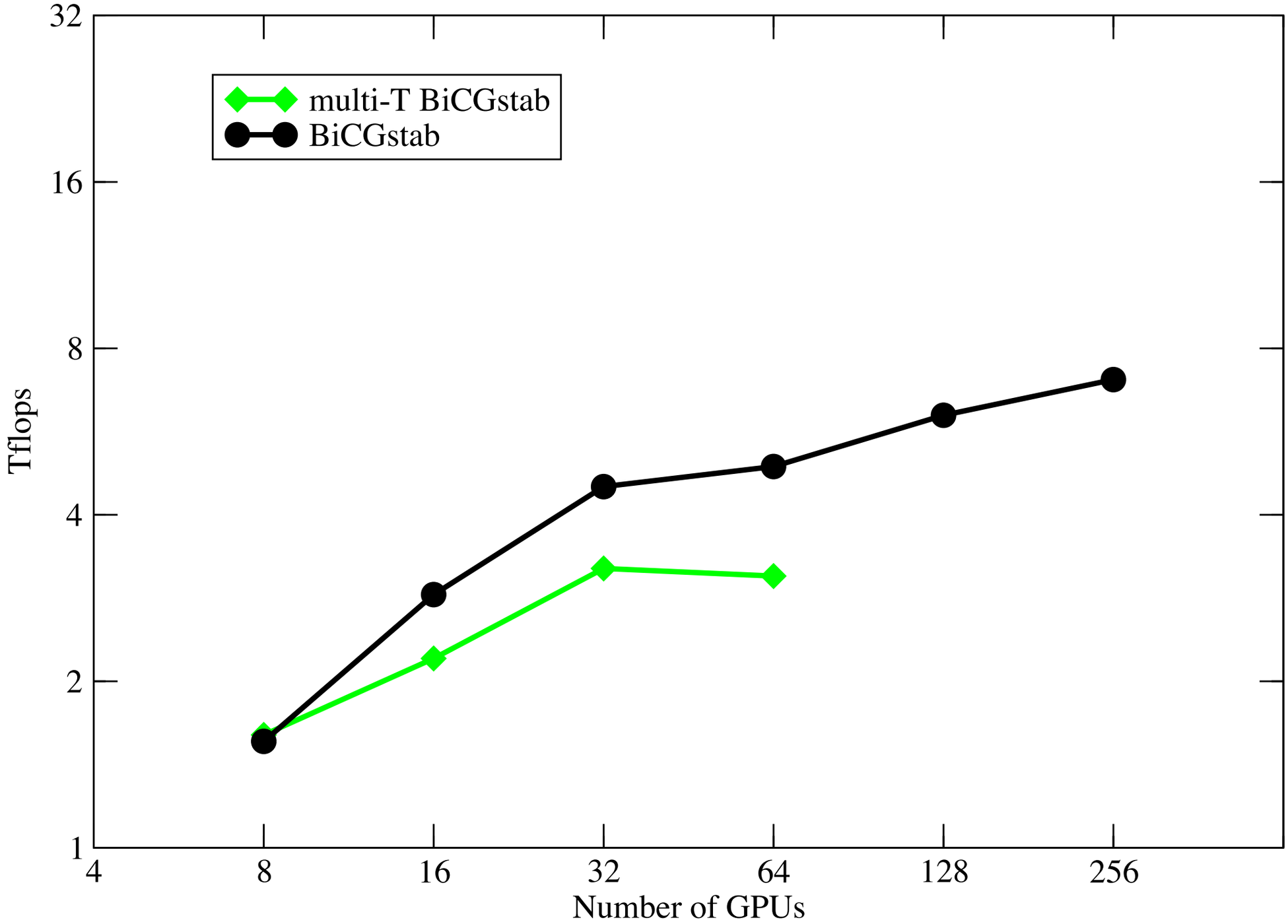


- No comms between domains
  - Block Diagonal Preconditioner
- Blocks impose λ cutoff
- Finer Blocks
  - lose structure in operator
  - lose long wavelength/low energy modes
- Heuristically (& from Lüscher)
  - keep wavelengths of ~ $O(\Lambda_{QCD}^{-1})$
  - Λ_{QCD}^{-1} ~ 1fm
  - 32^3x256 Aniso: (a_s=0.125fm, a_t=0.035fm)
    - Our case: 8^3x32 blocks are ideal
  - Iso: 1fm ~ 8-10 sites (a=0.11fm)
- Min. blocksize has scaling implications

Size Matters



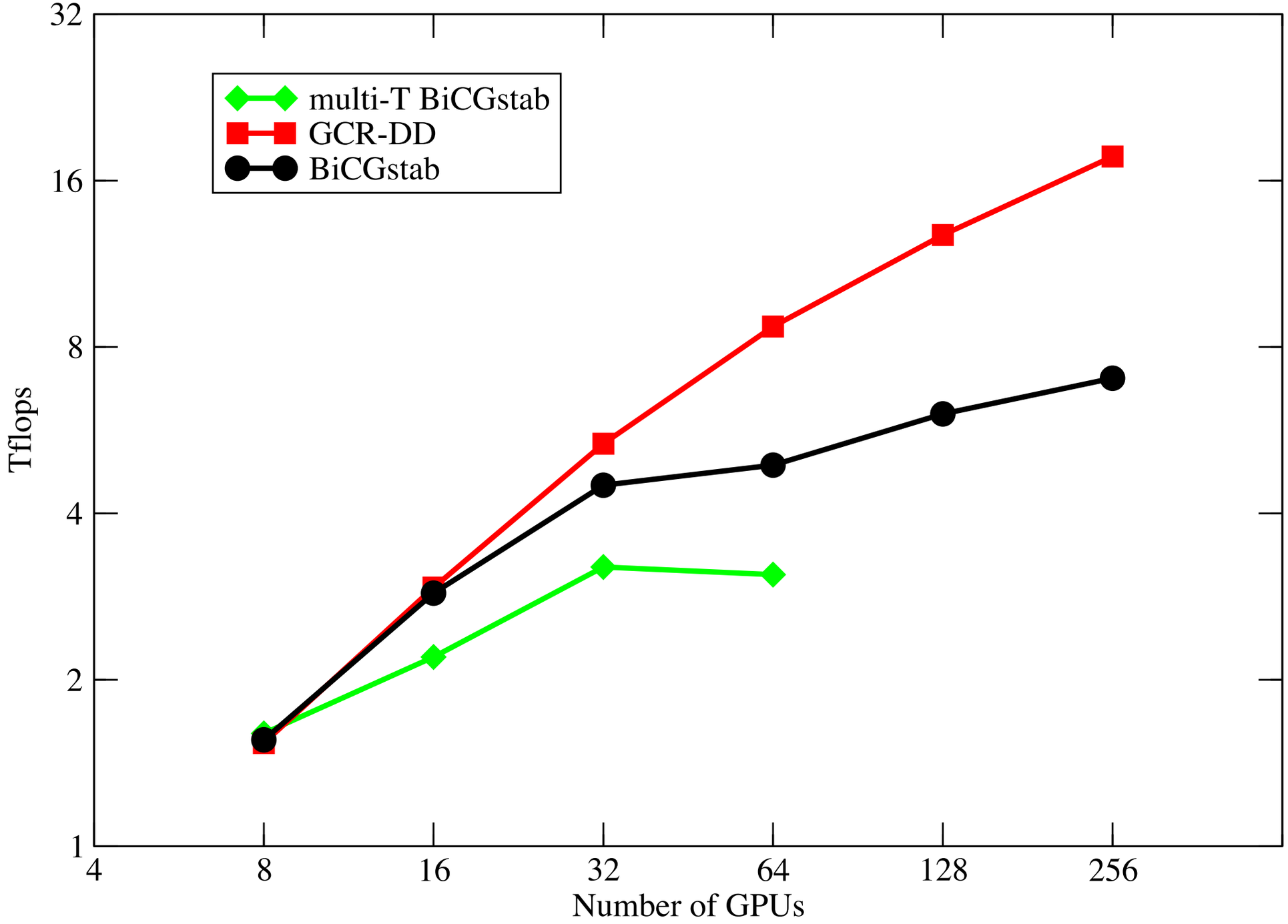

1
0.0001
1e-08
1e-12
1e-16
0
100
200
300
Iterations
|| r ||^2



Wilson clover

$32^3 x 256$

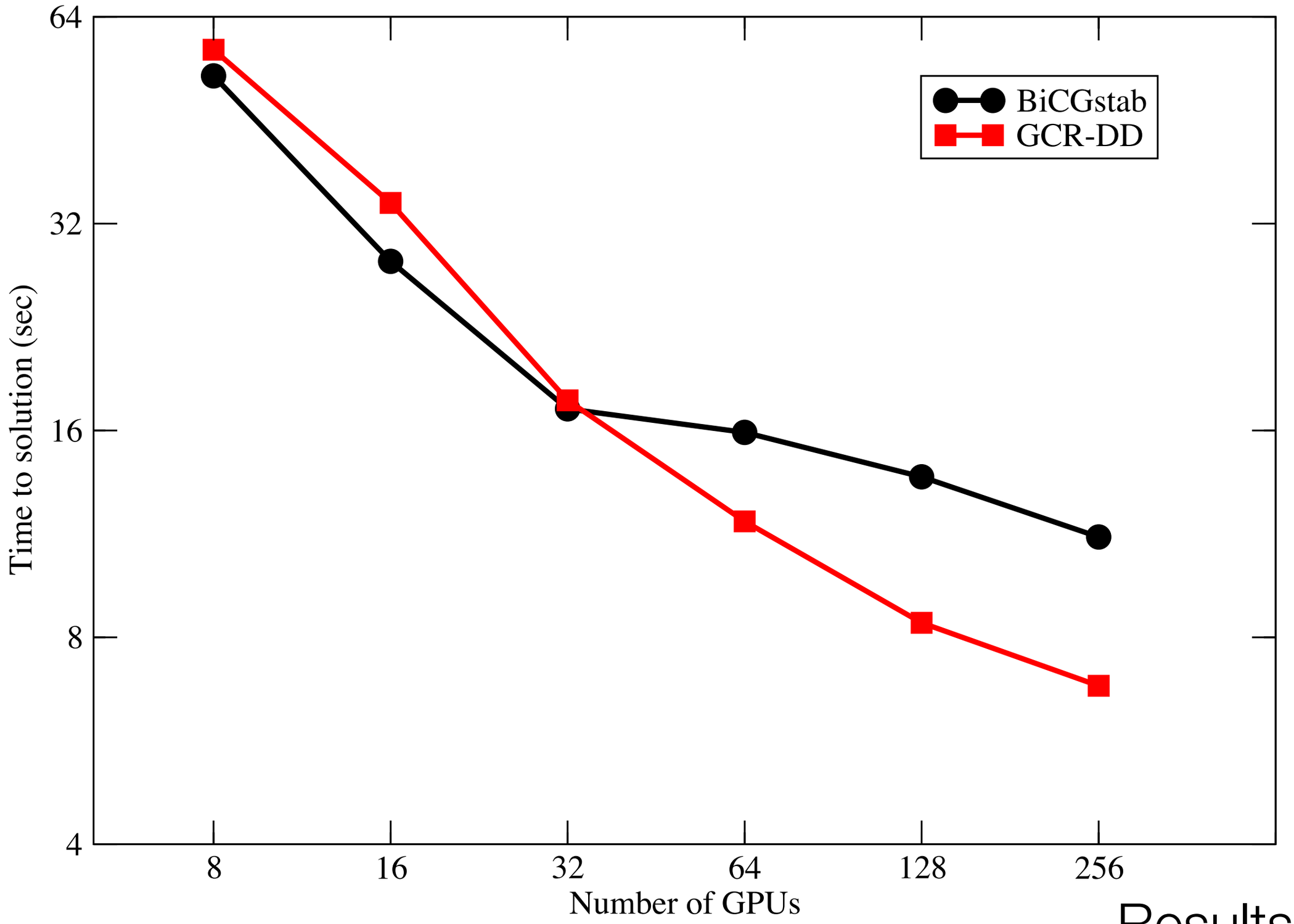
Results on Edge @ LLNL



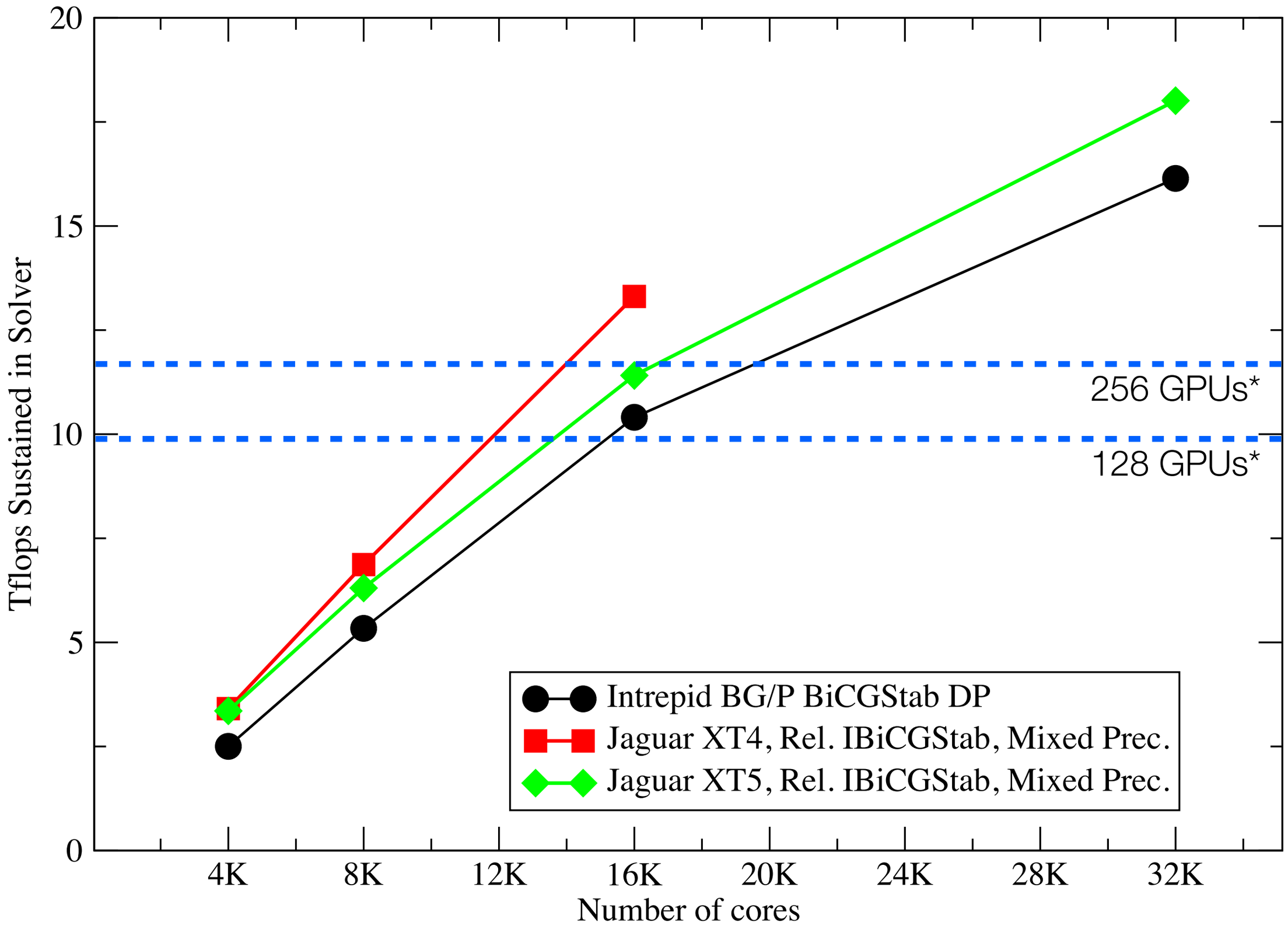
Wilson clover

$32^3 x 256$

Results on Edge @ LLNL



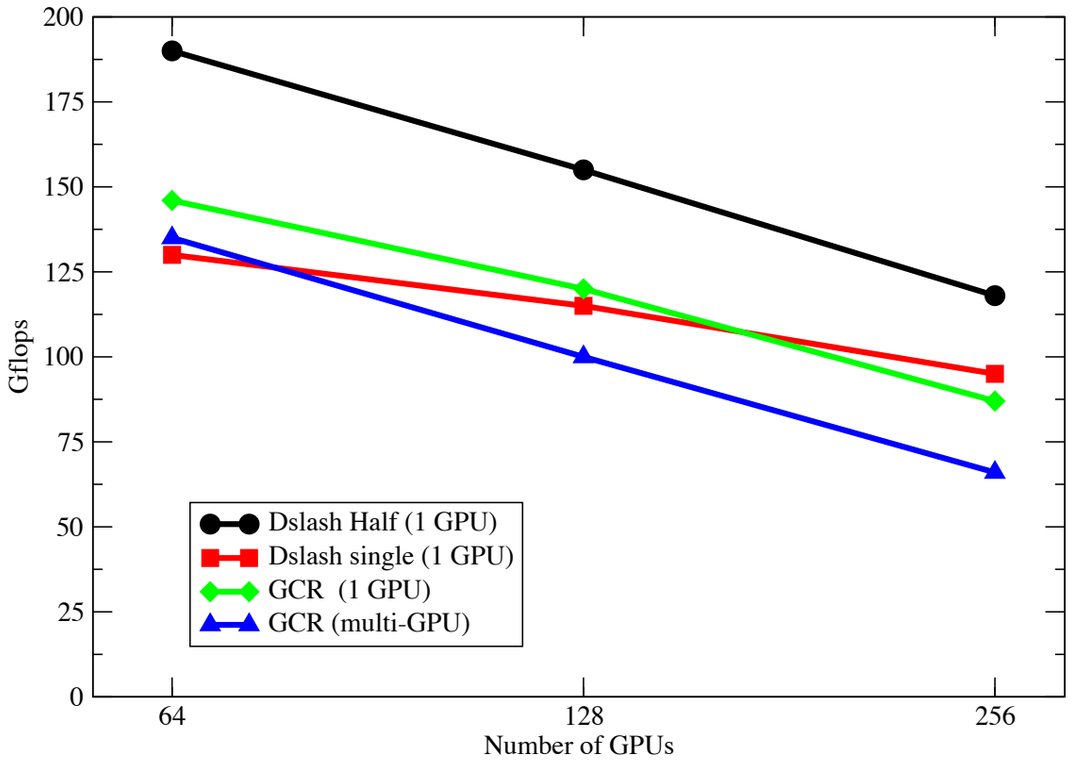
Results on Edge @ LLNL



*GPU Tflops scaled according solver iterations

Please sir, can I have some more?

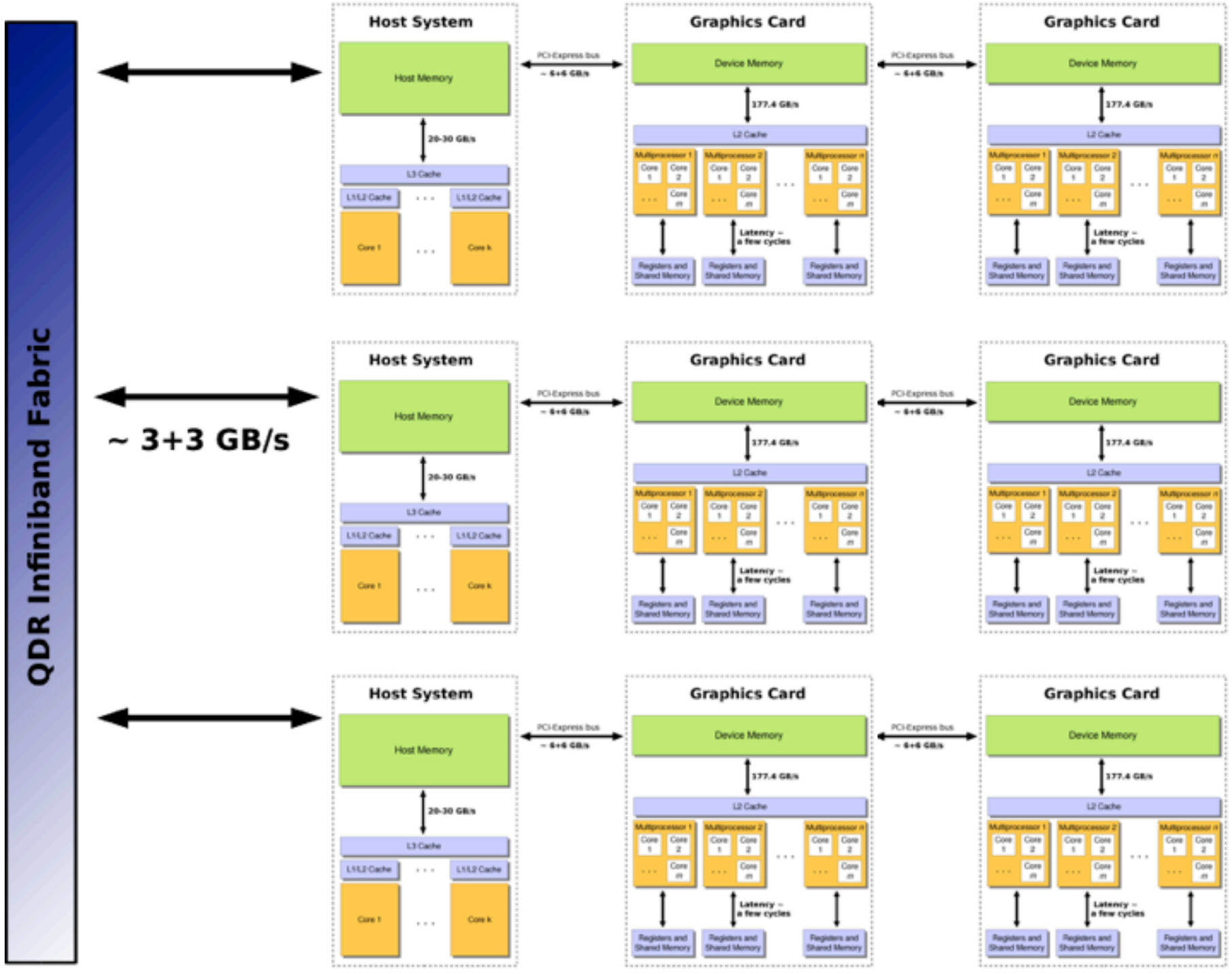
- 17.2 Tflop on 256 GPUs = 69 Gflops/GPU
  - using all the precision tricks
  - local problem is small
    - Low occupancy?
    - Driver overheads?
    - Communications?
- 40^3x256, 2x bigger V
  - Expect 21 Tflops if comms bound
  - Obtain 24 Tflops
- Performance bound by small local volume
  - Solve multiple systems simultaneously
  - Optimize for small volumes?



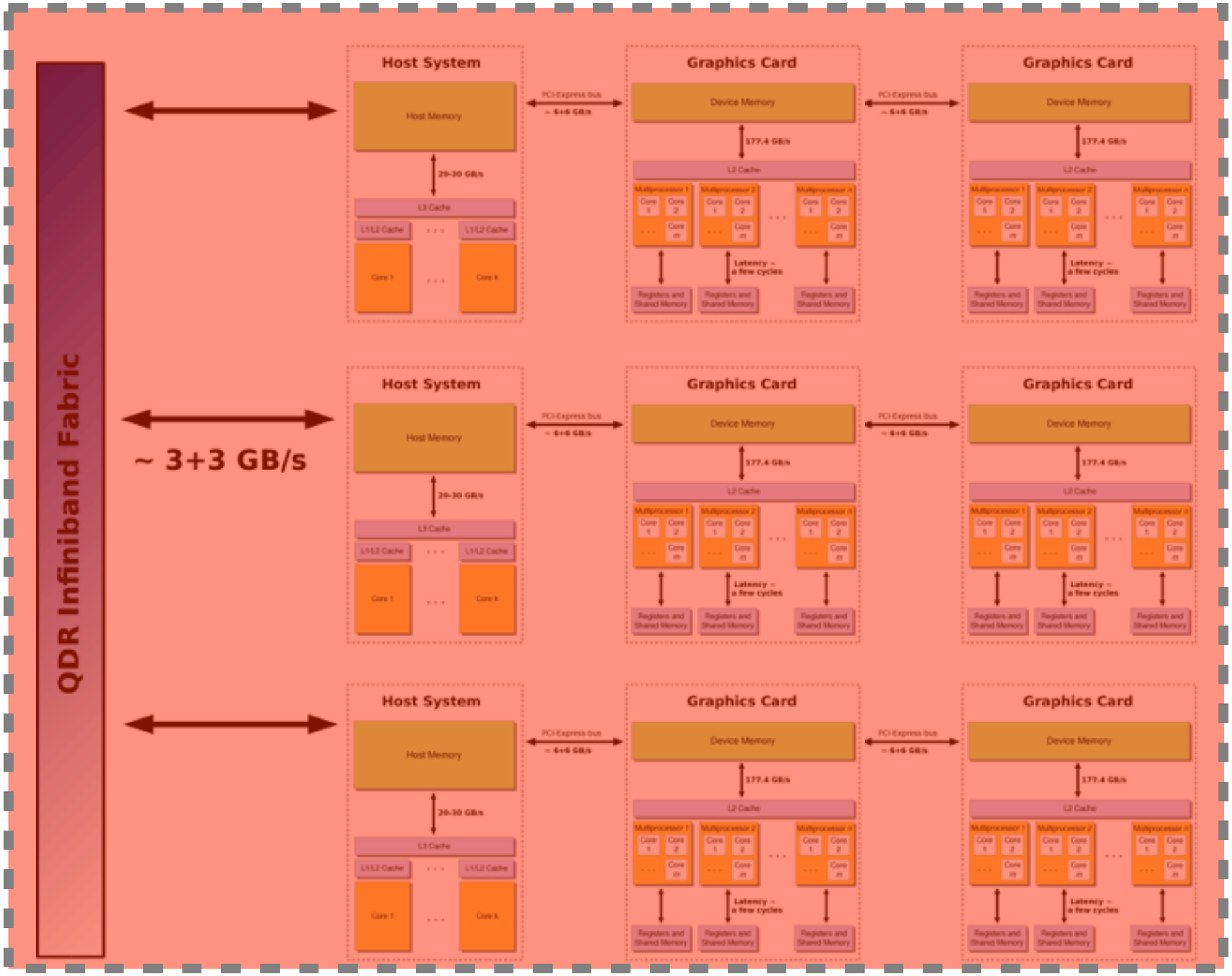
Refining the Algorithm

- RAS preconditioner = block Jacobi
  - Expect acceleration from over-relaxed preconditioner
- Preconditioner only requires $\tau > 0.1$
  - Use 8-bit precision?
- No incorporation of GPUDirect yet
  - Modest improvement expected since not totally comms bound
- Multi-level domain decomposition

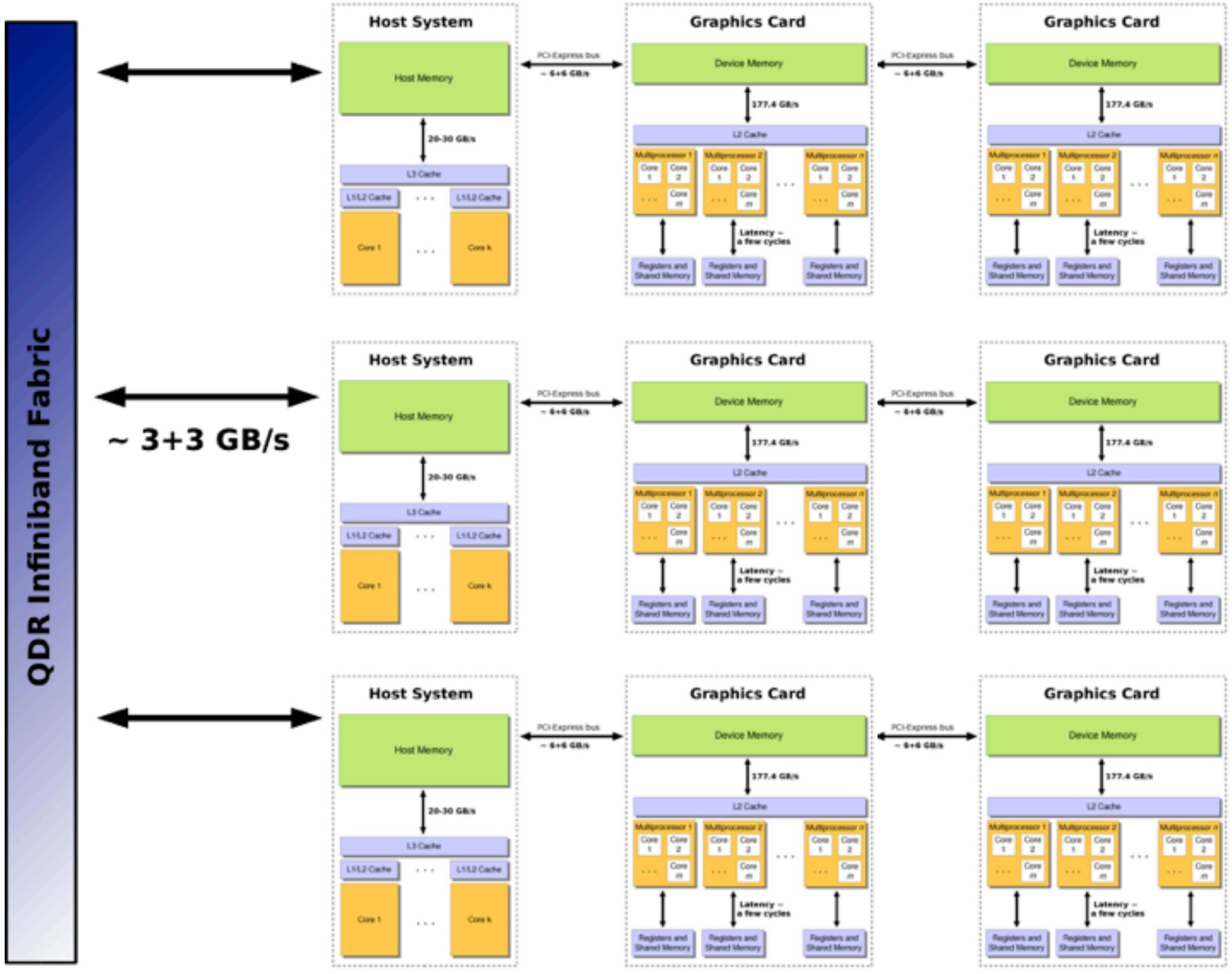
Multi-level domain decomposition



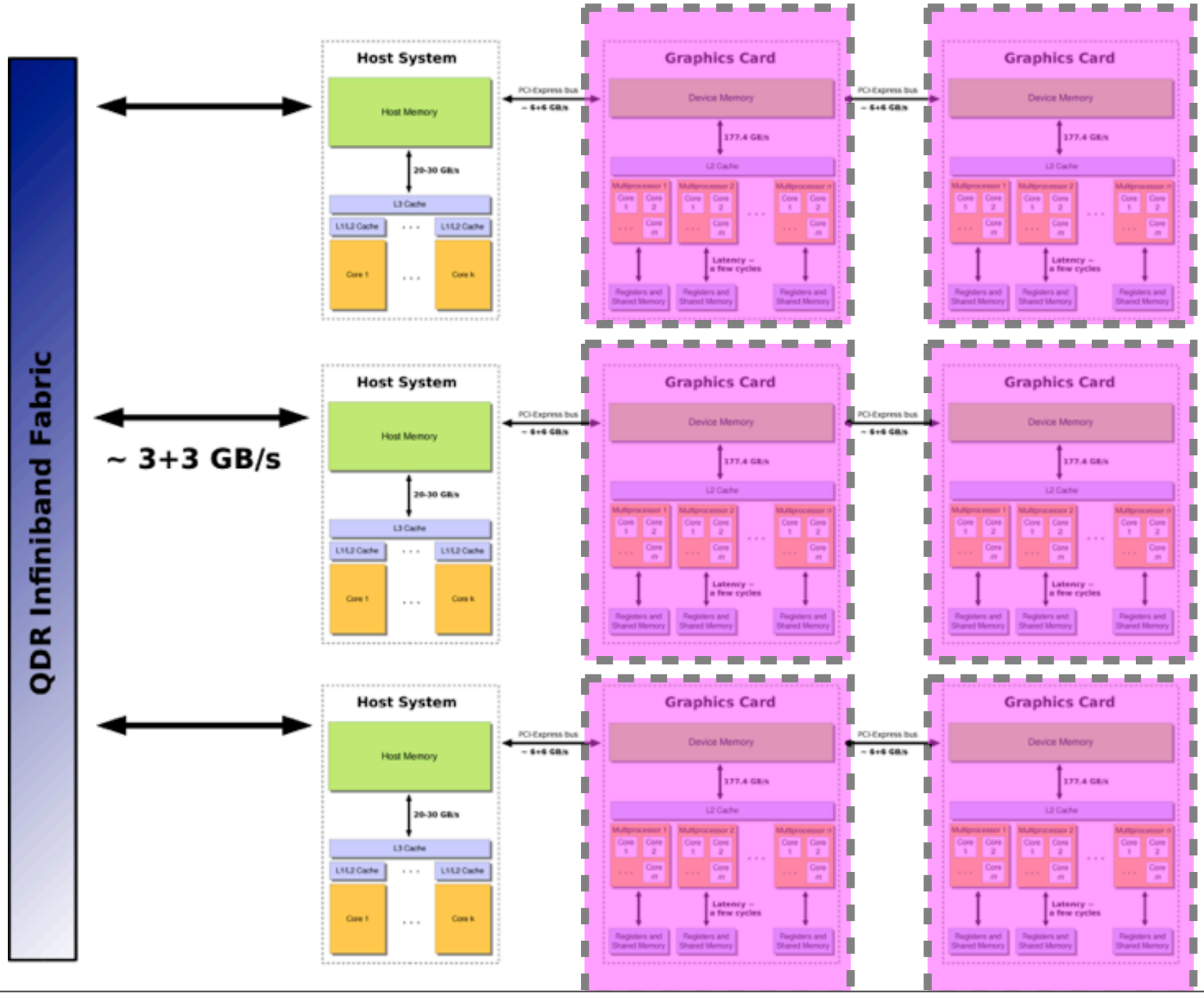
Multi-level domain decomposition



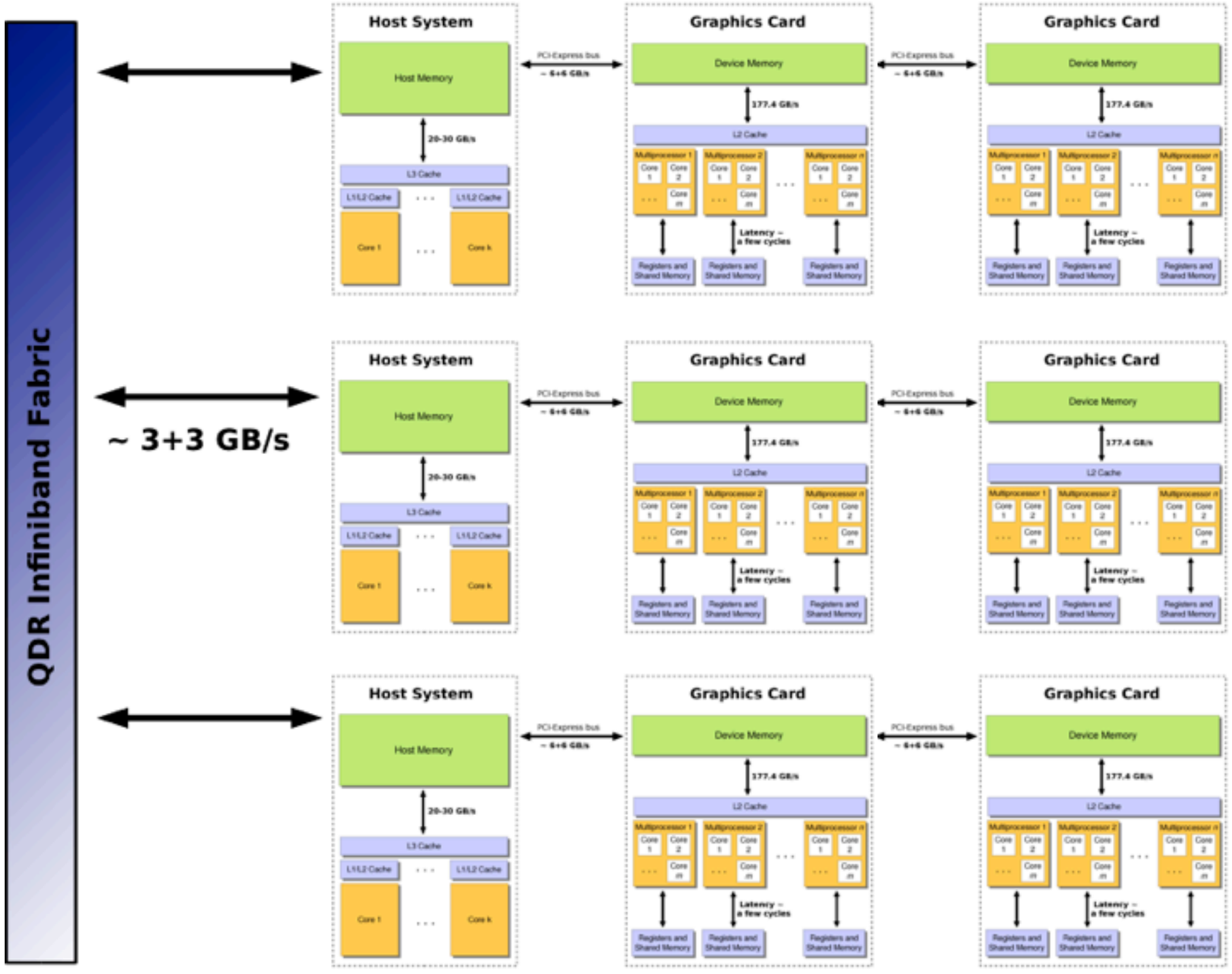
Multi-level domain decomposition



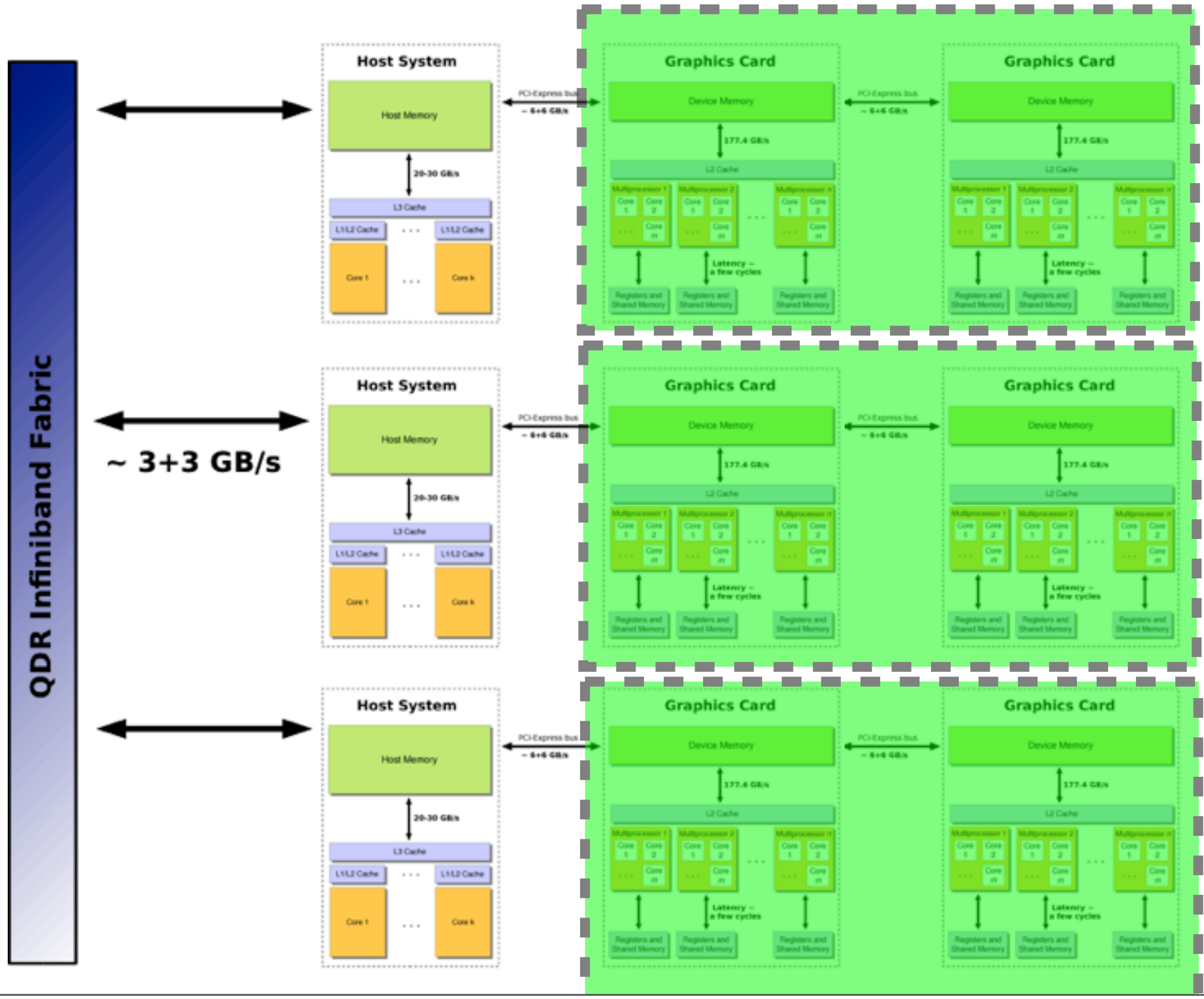
Multi-level domain decomposition



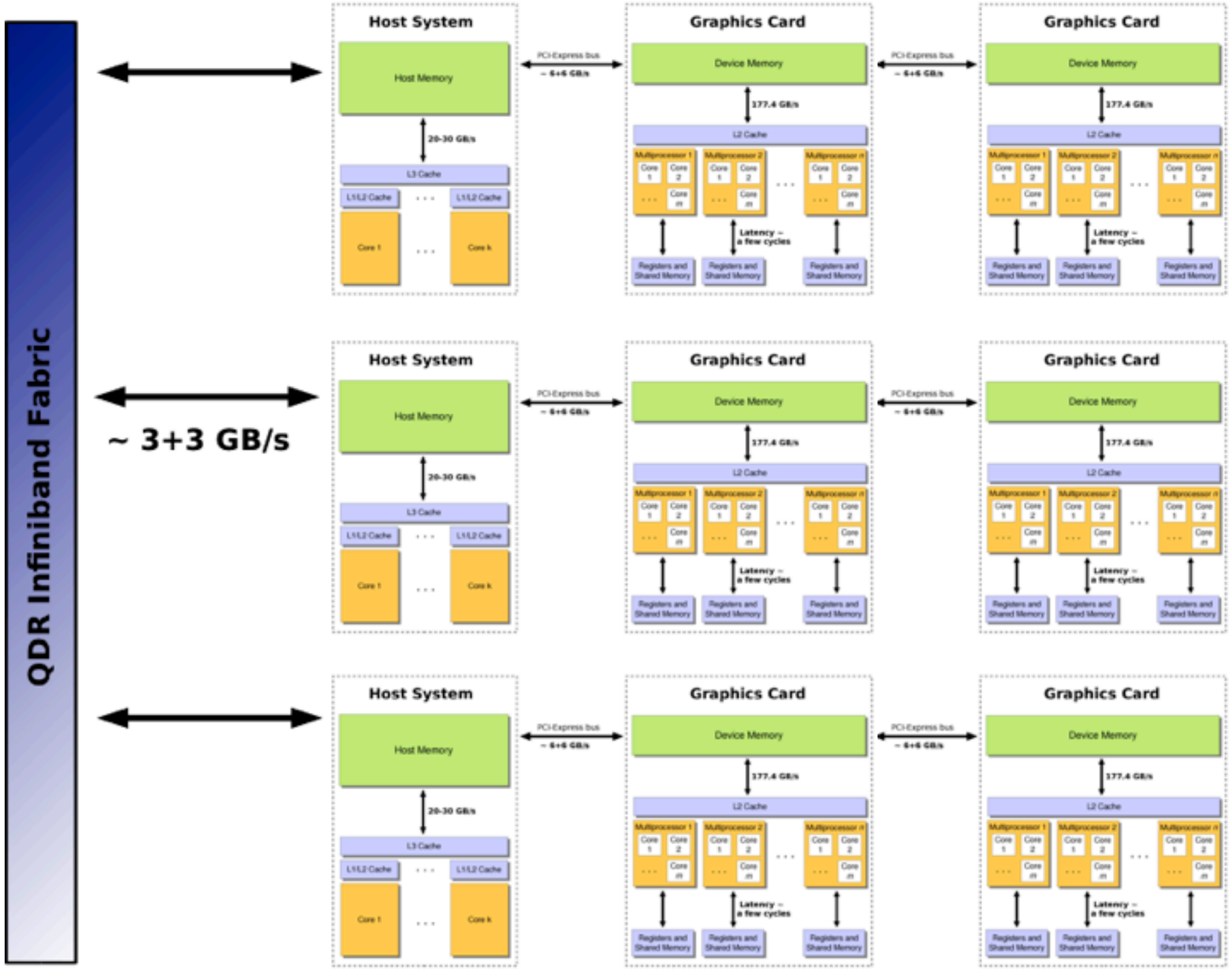
Multi-level domain decomposition



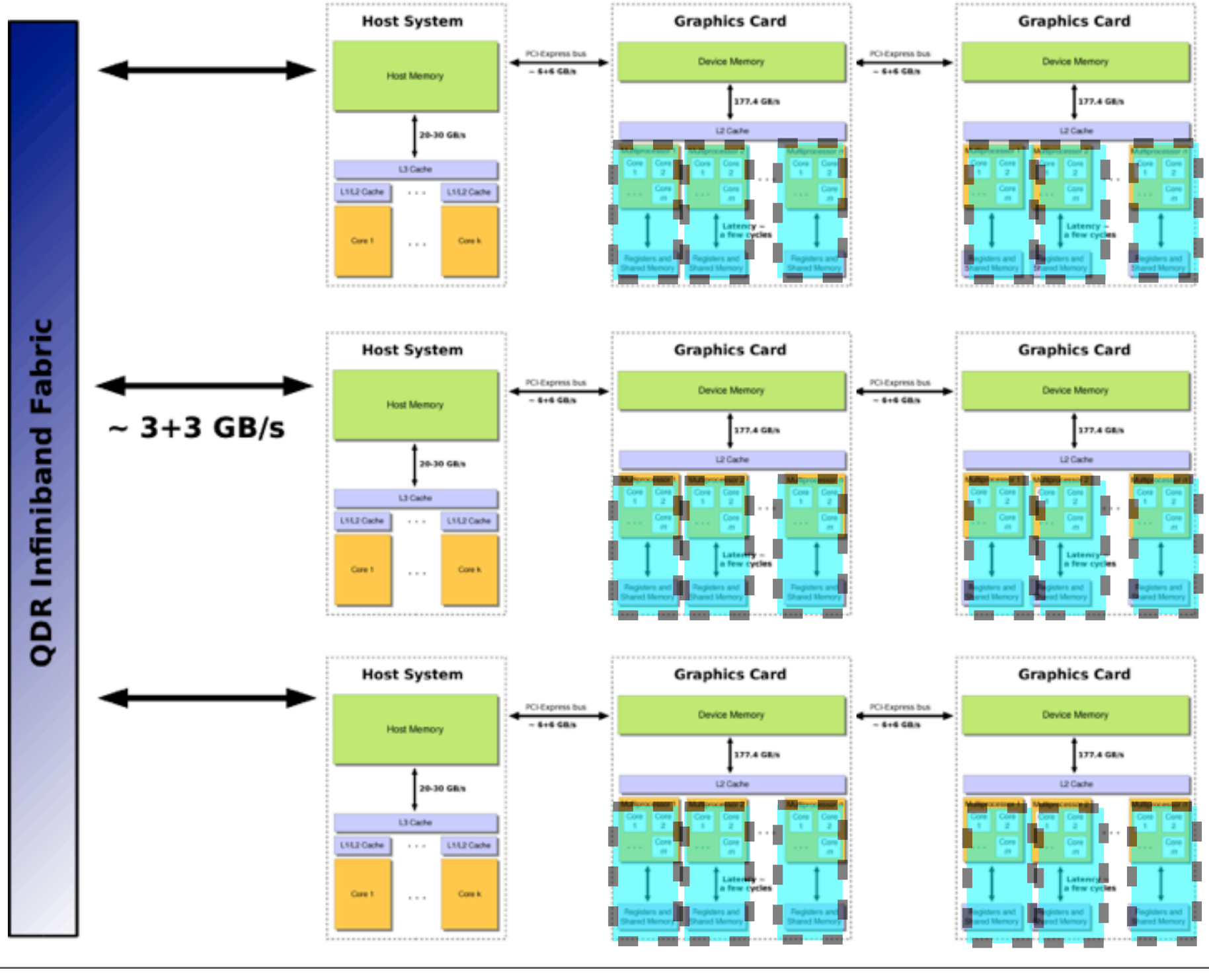
Multi-level domain decomposition



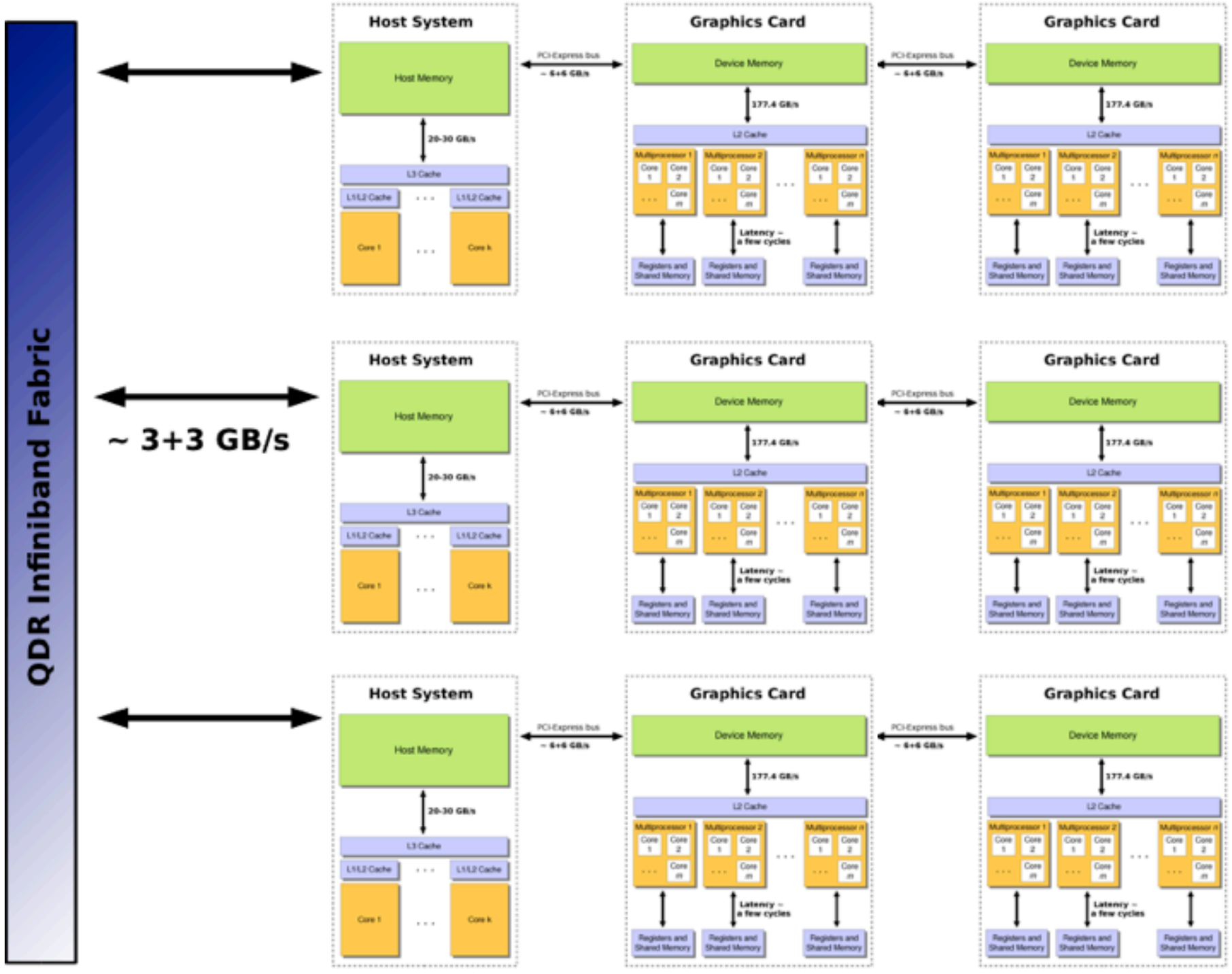
Multi-level domain decomposition



Multi-level domain decomposition



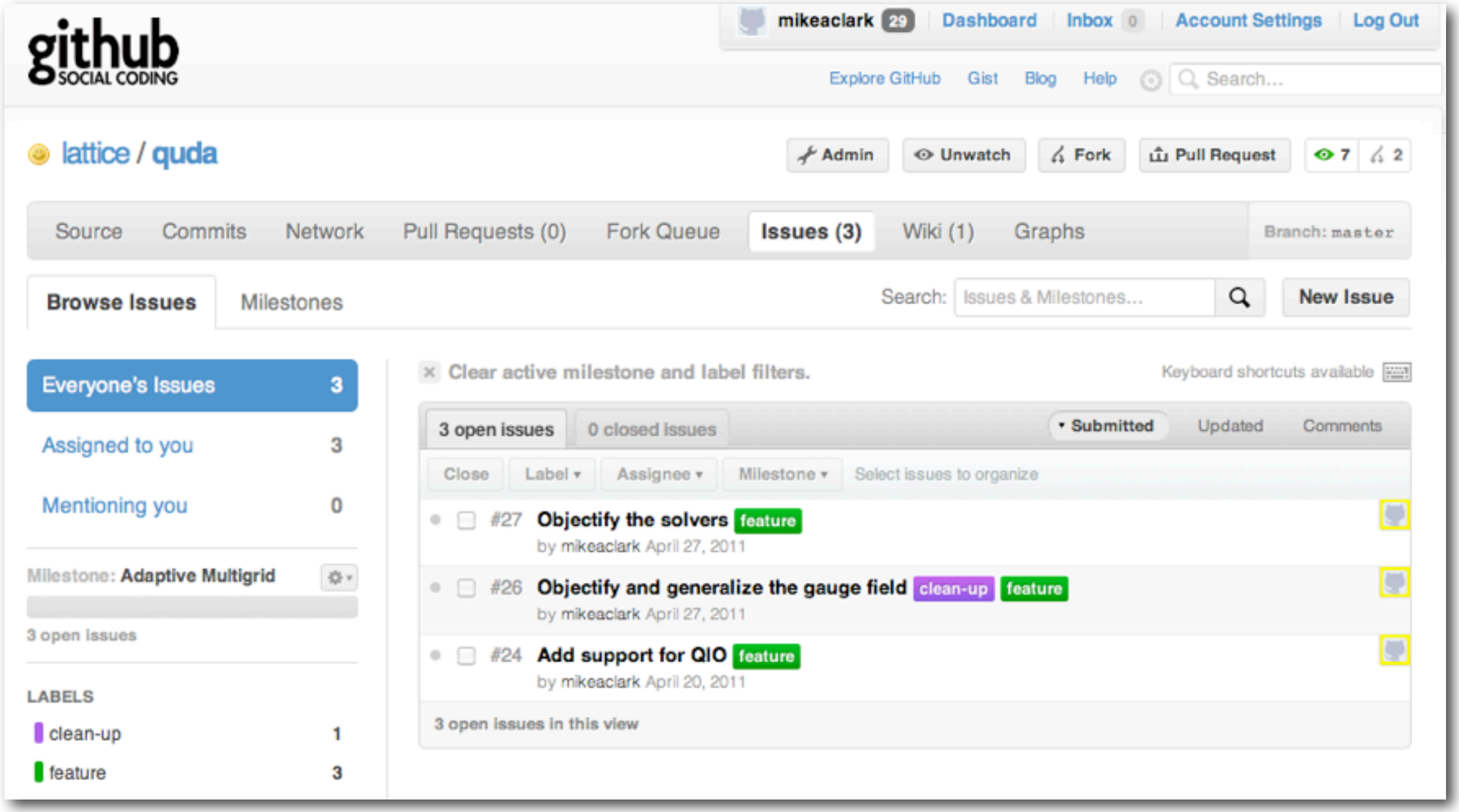
Multi-level domain decomposition



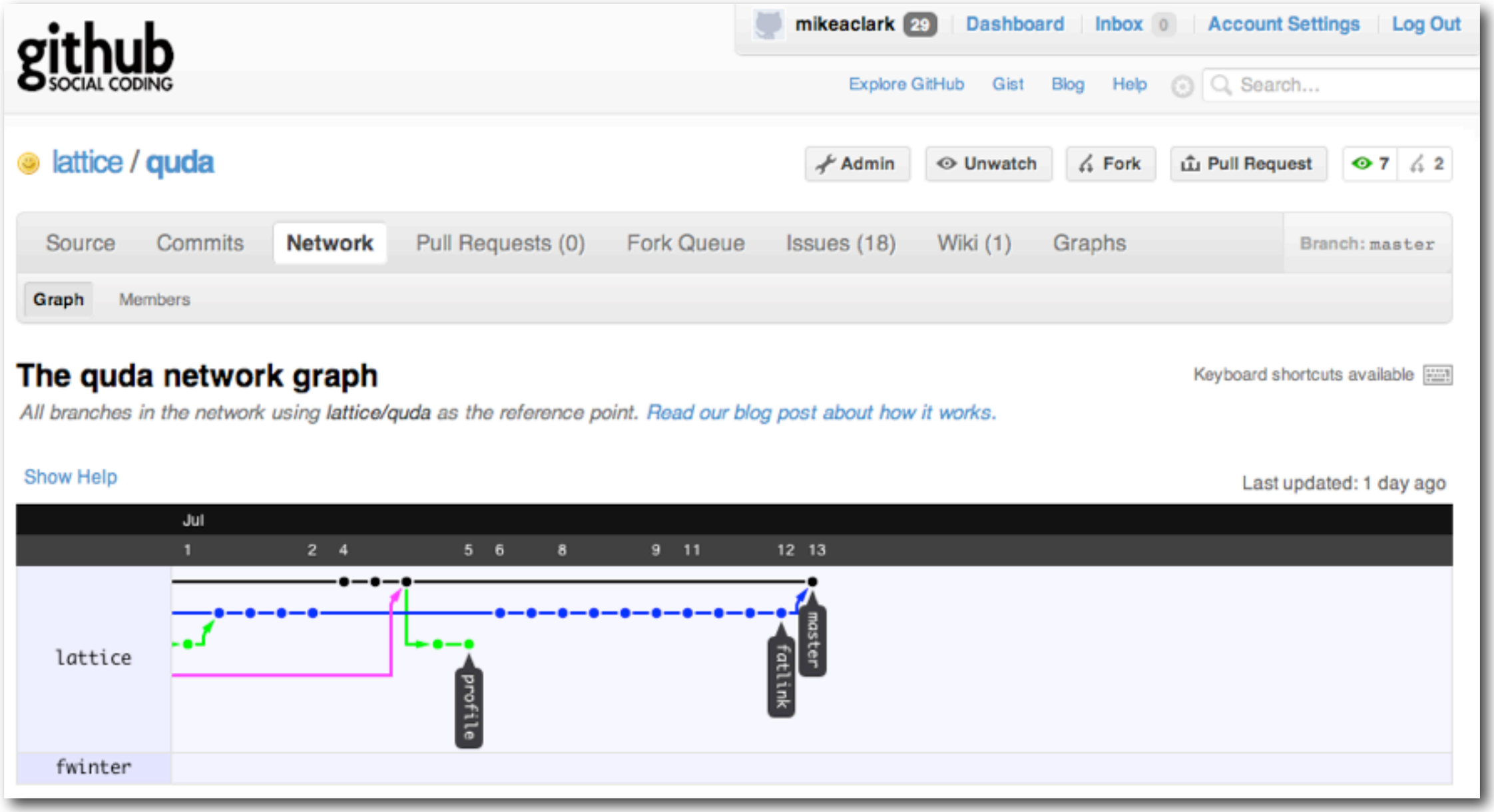
Conclusions and Outlook

- First demonstration of QCD GPU computation at the 10+ Tflops scale
  - To appear in **Proceedings of SC11**, “Scaling lattice QCD past 100 GPUs”
- Communication reduction algorithms critical for strong scaling
  - Flops are cheaper than bytes
- GPU clusters **can** replace capability super computers
- Huge algorithmic parameter space to explore
  - Overlapping blocks, boundary conditions, multi-level
- DD preconditioner ⇒ smoother for αMG

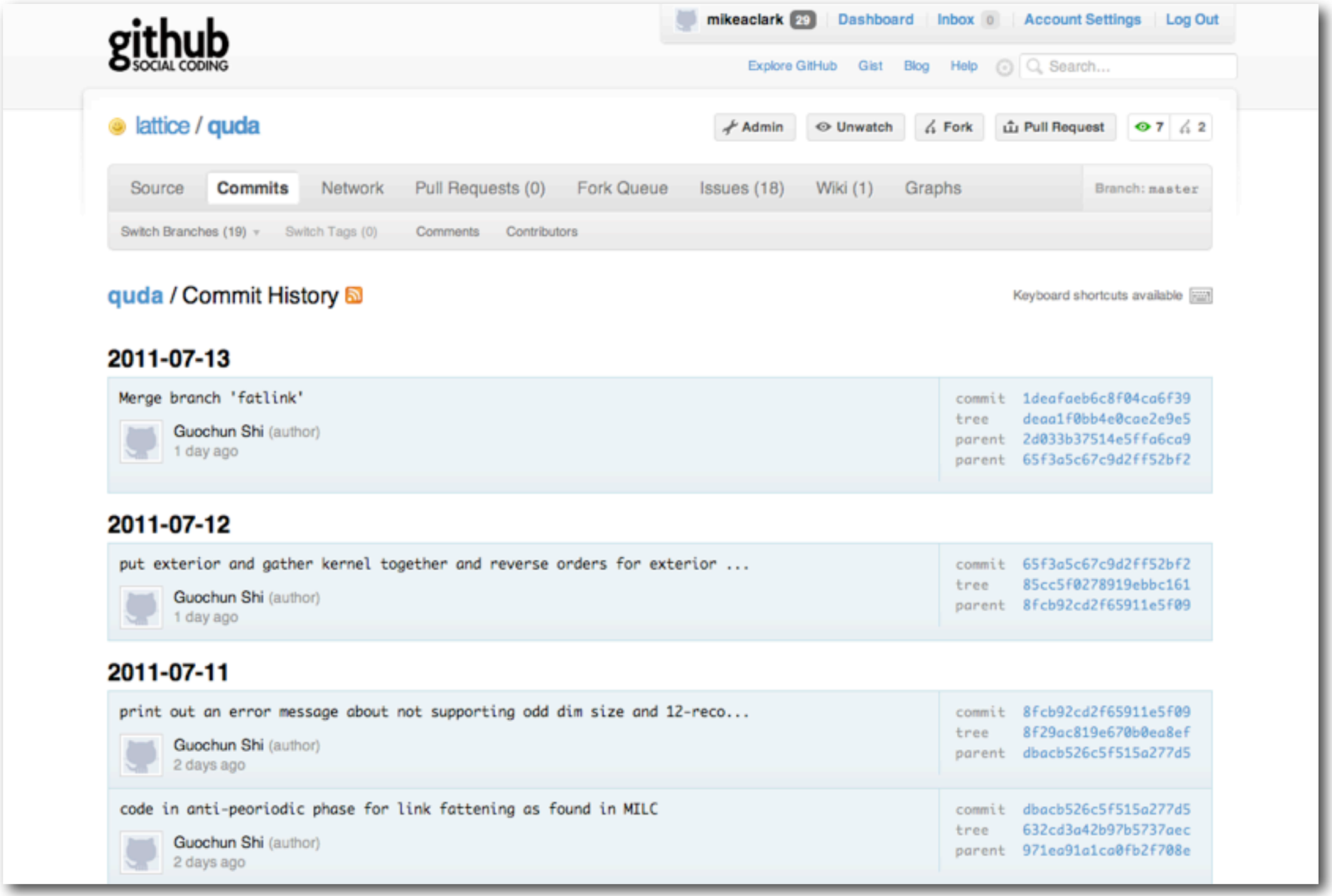
https://github.com/lattice/quda



https://github.com/lattice/quda



https://github.com/lattice/quda



https://github.com/lattice/quda

Come join the fun...