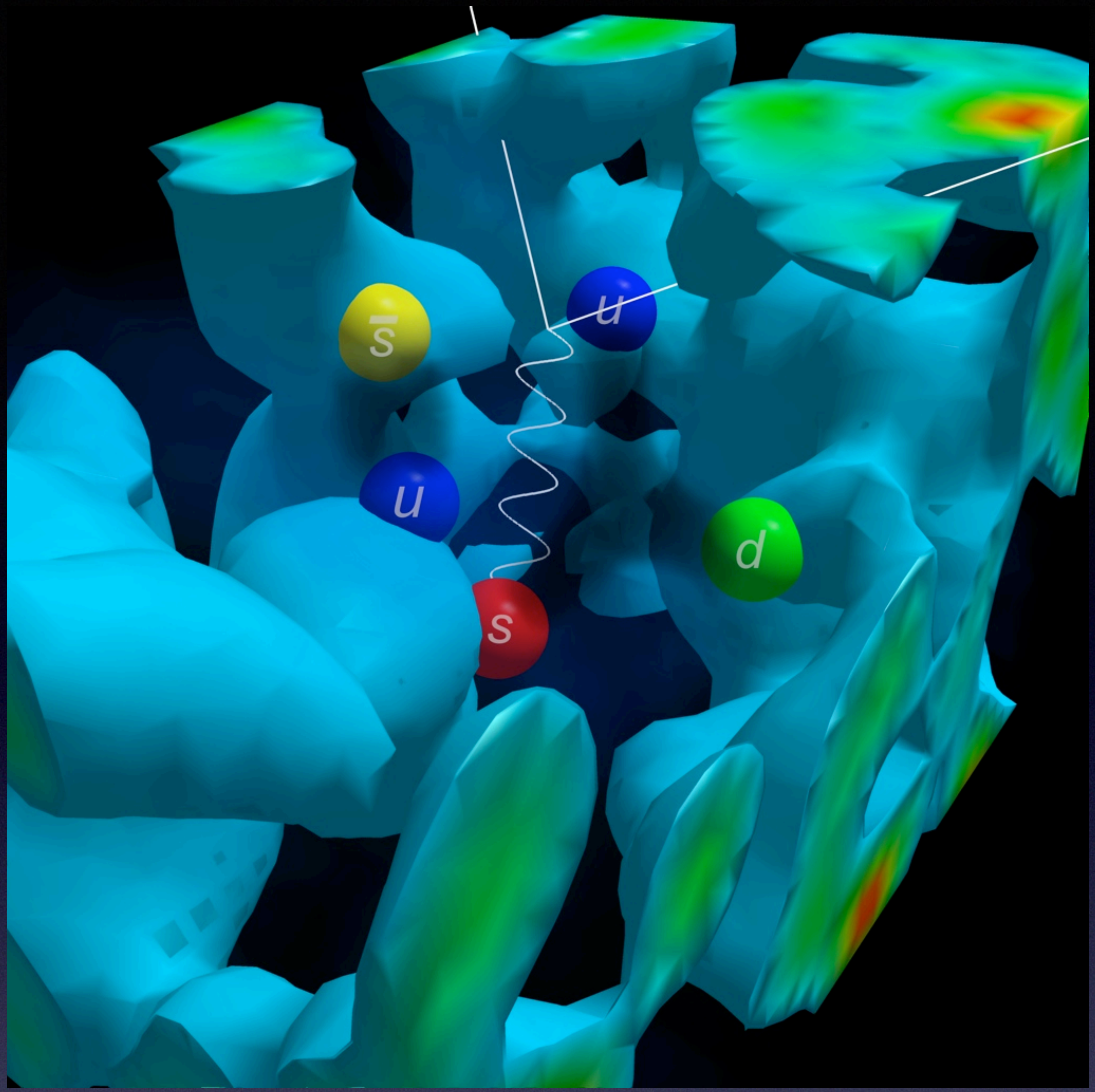
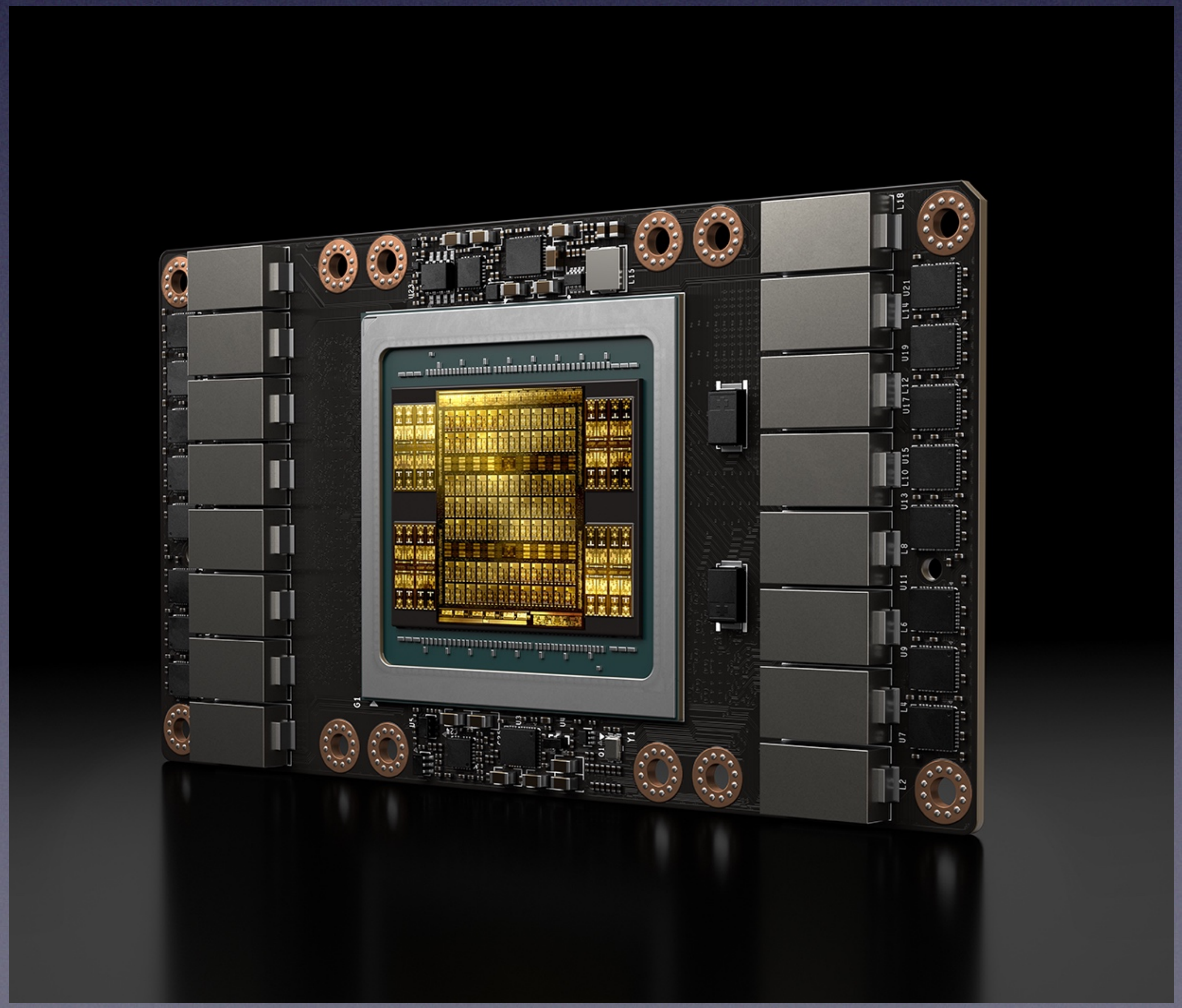


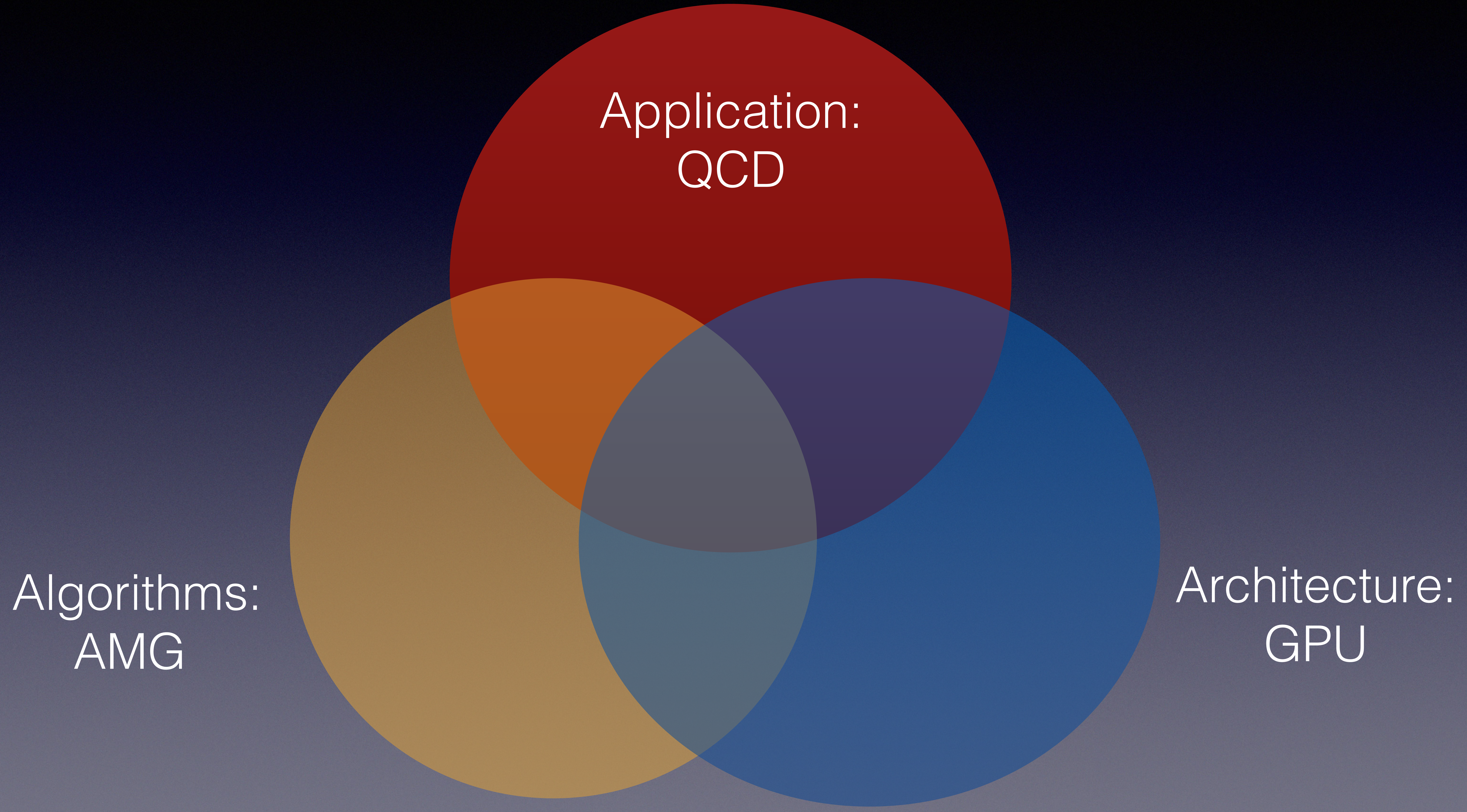
Past and Future of QCD on GPUs

Richard C. Brower, Boston University
with many slides from Kate Clark, NVIDIA



High Performance Computing in High Energy Physics
CCNU Wahan China, Sept 19, 2018

Optimize the Intersection



Question to address

- How do we put Quantum Field on the computer?
- How to Maximize Flops/$, bandwidth/$ at Min Energy?
- How to implement fastest algorithms: Dirac Sovlers, Sympletic Integrators, etc ?

Standard Lattice QCD Formulation

$$\text{Path Integral} = \int \mathcal{D}^2 A(x)\psi(x)\; e^{-\frac{i}{g^2}\int d^3x dt[\; F^2_{\mu\nu} + \overline{\psi}\gamma_\mu(\partial_\mu - iA_\mu + m)\psi\;]}$$

1. Complex time for probability $it \to x_4$

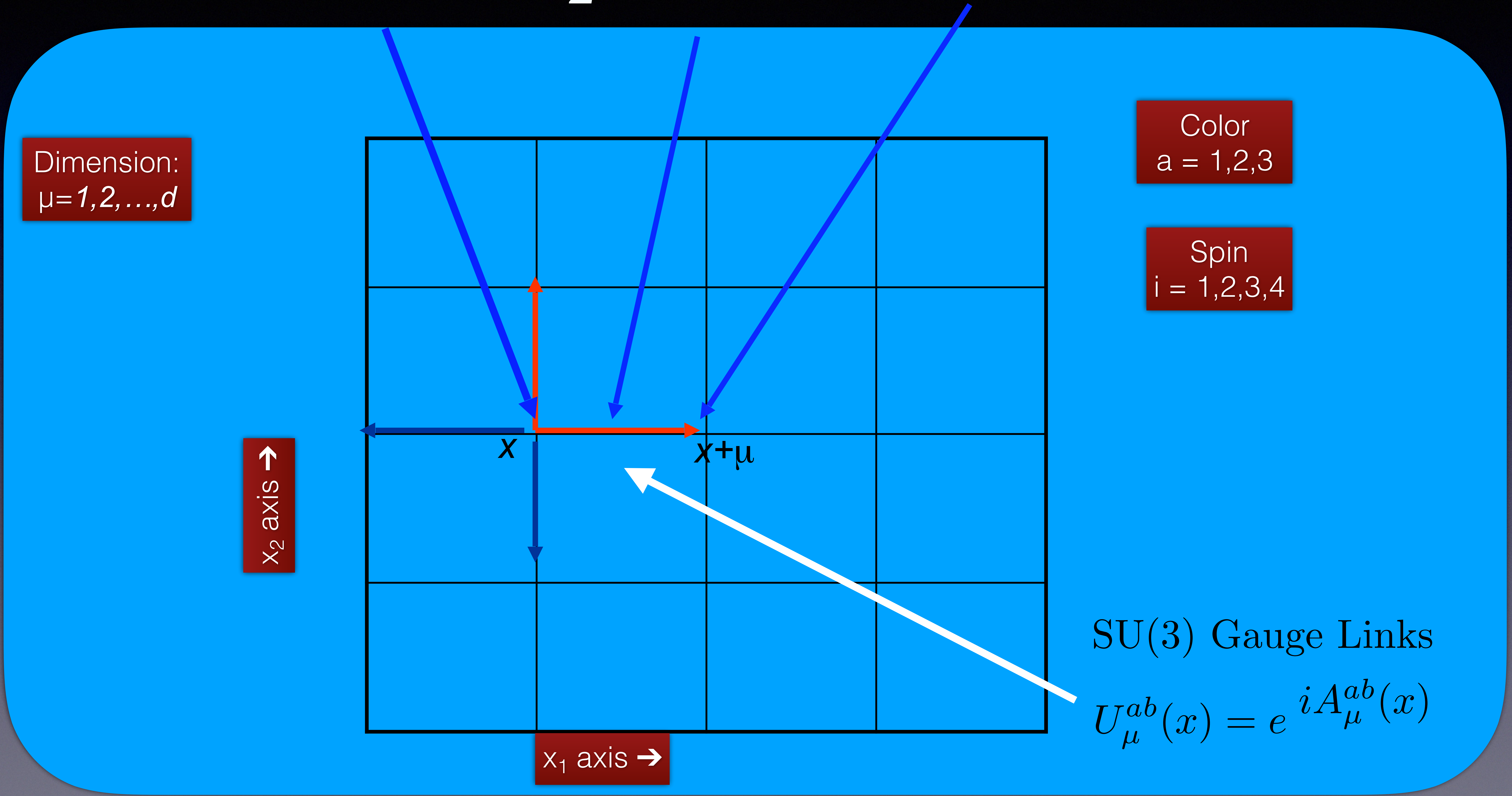
2. Lattice Finite Differences $(\partial_\mu - iA_\mu)\psi(x) \to (\psi_{x+\hat{\mu}} - e^{iaA_\mu}\psi_x)/a$

3. Fermionic integral $\int d\psi d\overline{\psi}\; e^{-\overline{\psi}_x D_{xy}(A)\psi_y} \to Det[D]$

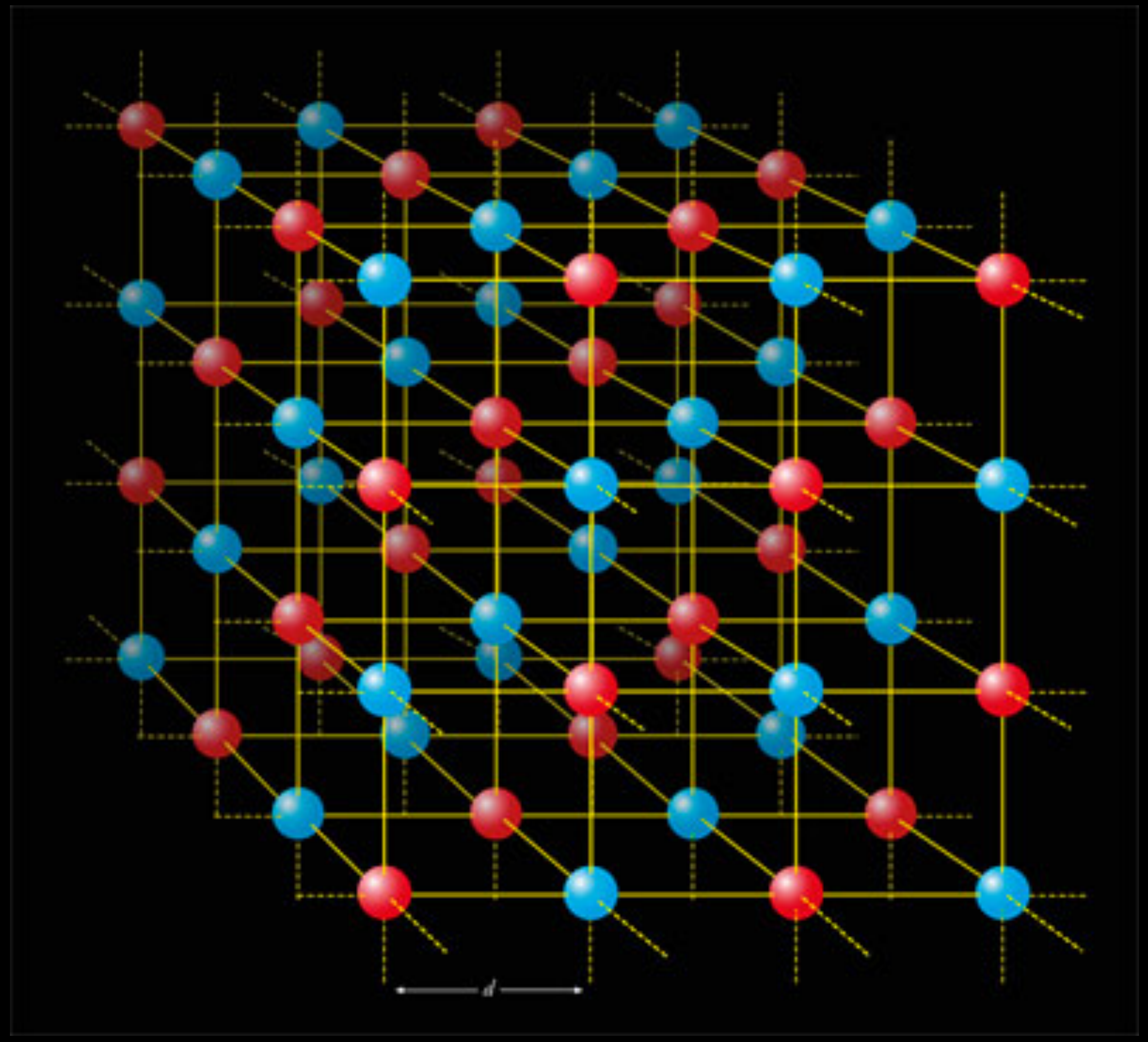
4. Bosonic (pseudo- Fermions) $Det[D] \to \int d\phi d\overline{\phi}\; e^{-\overline{\phi}_x [1/D]_{xy}\phi_y}$

Lattice Dirac

$$\overline{\psi}_{ia}(x)\,\frac{1-\gamma_\mu^{ij}}{2}\,U_\mu^{ab}(x)\,\psi_{jb}(x+\hat{\mu})$$



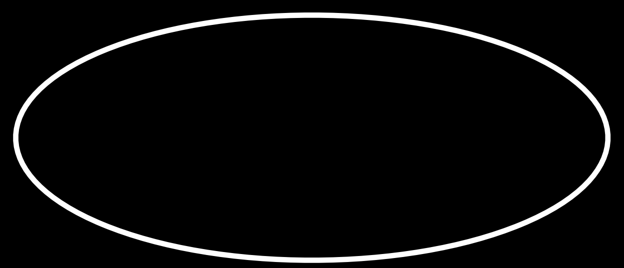
Quarks Propagation on Hypercubic Lattice*



- **Dominate Linear Algebra :**
  Matrix solver for Dirac operator.

- **Gauge Evolution:**

In the semi-implicit Hamiltonian evolution in Monte Carlo time.



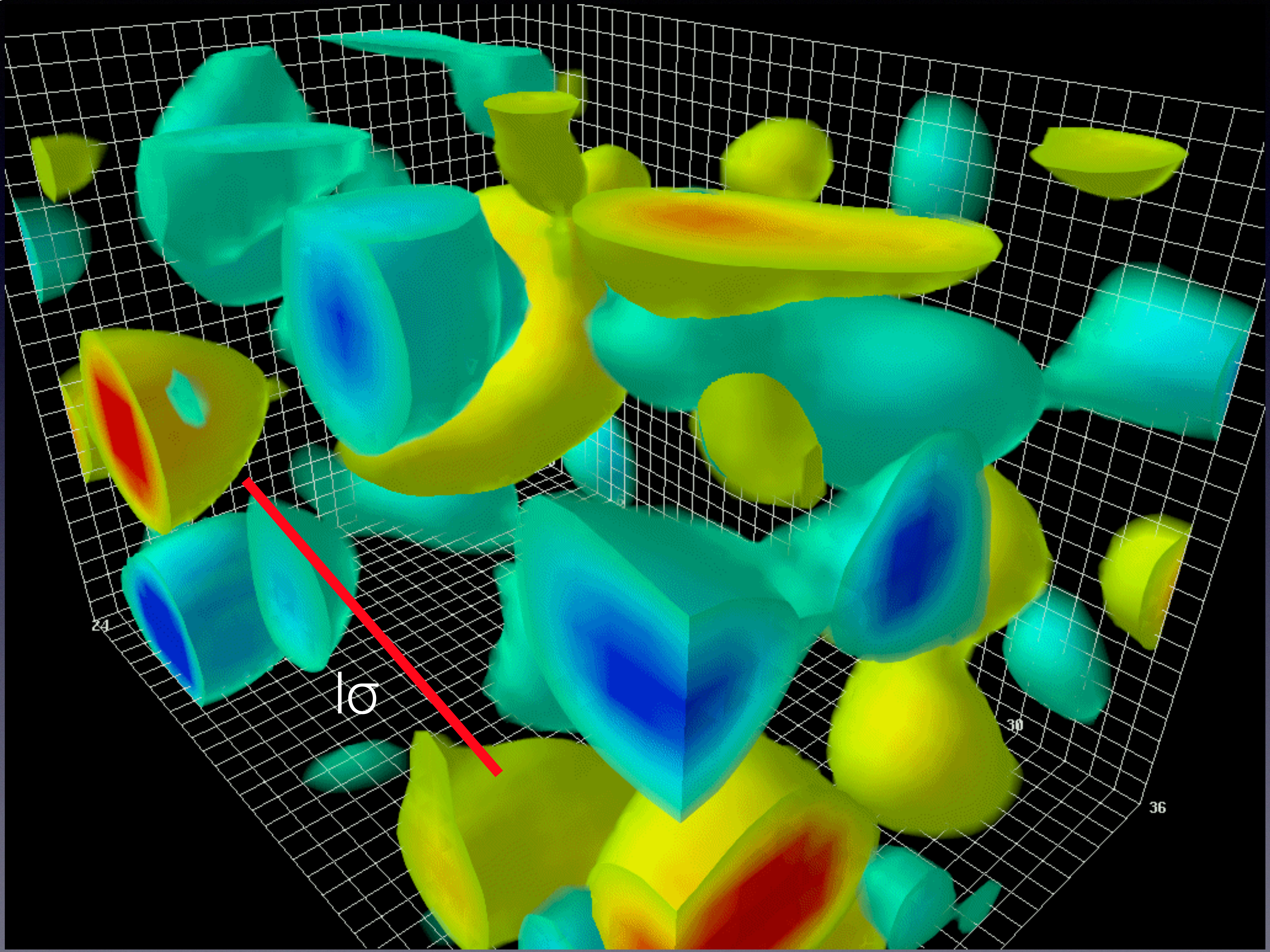
- **Analysis:**

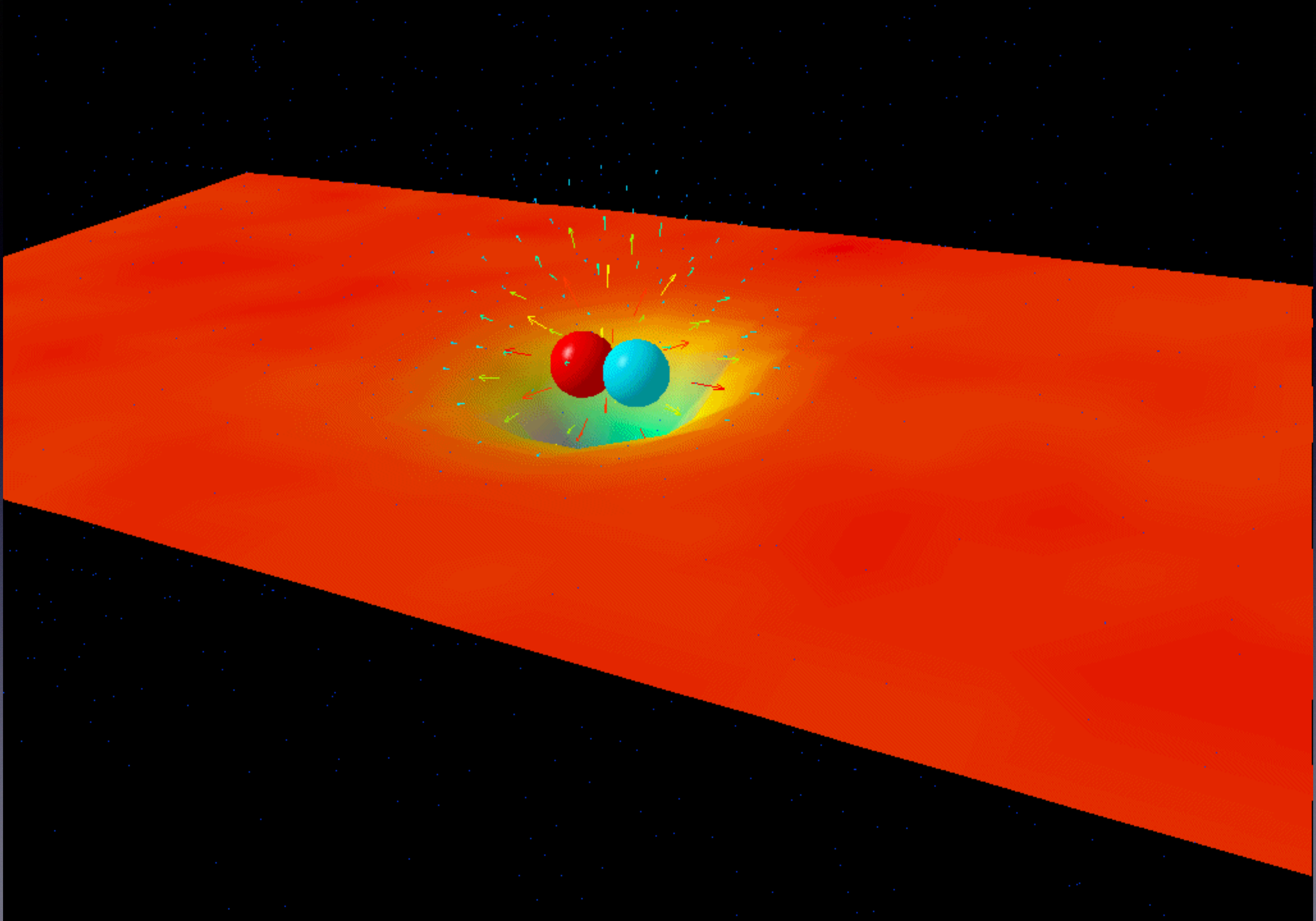
* Others: SUSY(Catteral et al), Random Lattices(Christ et al), Smiplicial Sphere (Brower et al)

Byte/flop in Dirac Solver is main bottleneck

- Bandwidth to memory rather than raw floating point throughput.
- Wilson Dirac/DW operator (single prec) : 1440 bytes per 1320 flops.
- Clover-improved Wilson : 1728 bytes per 1824 flops.
- Asqtad and HISQ :1560 bytes per 1146 flops.
- Other operations although less common are often much worse.
- Network bandwidth is dependent on sub-volume to surface area.

Complex Gloun “Plasma”,Instantons, Topological Zero Modes & Confinement length lσ



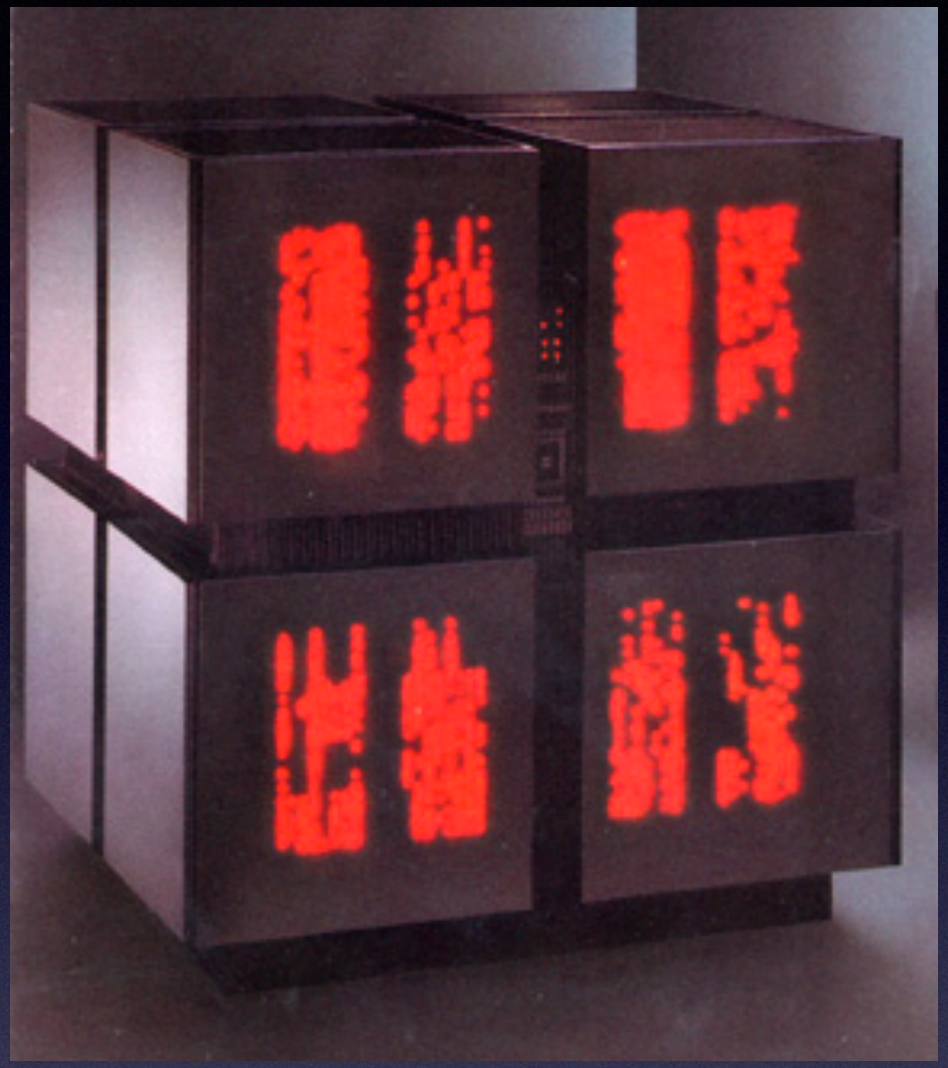


Plan of Presentation

- Brief History
    - Hardware
    - Software
    - Algorithms
- Preparing for the Exascale
- A quantum leap into the Future ?

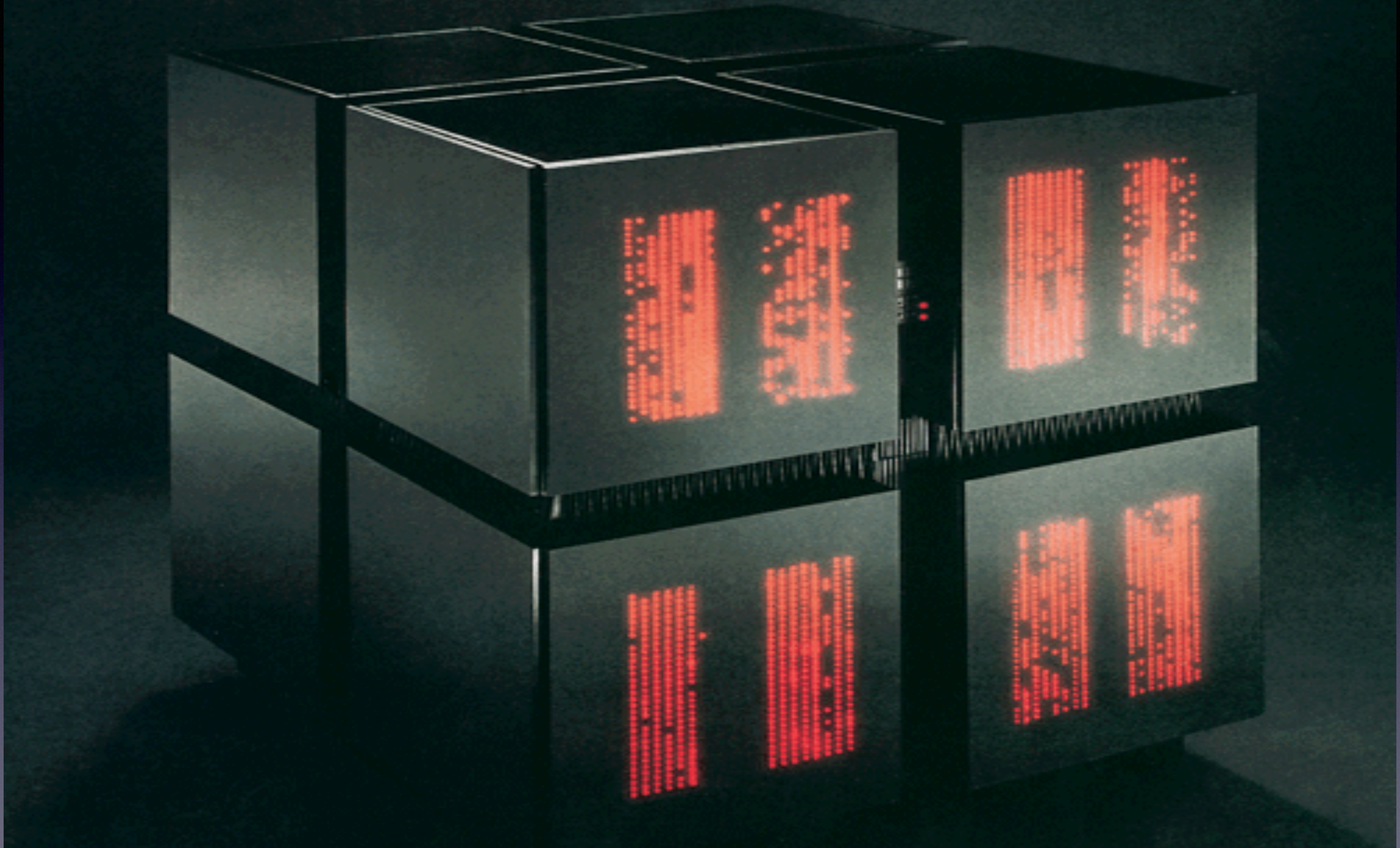
HARDWARE

Historical Perspective: 1980 First "Commercial" QCD machine



http://www.mission-base.com/tamiko/cm/cm-tshirt.html

CM-2 — Single Instruction Multiple Data (SIMD) Machine



In late 1980's Thinking Machines Corporation the 64K 1 bit processor CM-2 with performance in excess of 2500 MIPs, and floating point above 2.5Gflops

Next 30 years of Message Passing: QCDOC/Intel-Clusters, BlueGene, …

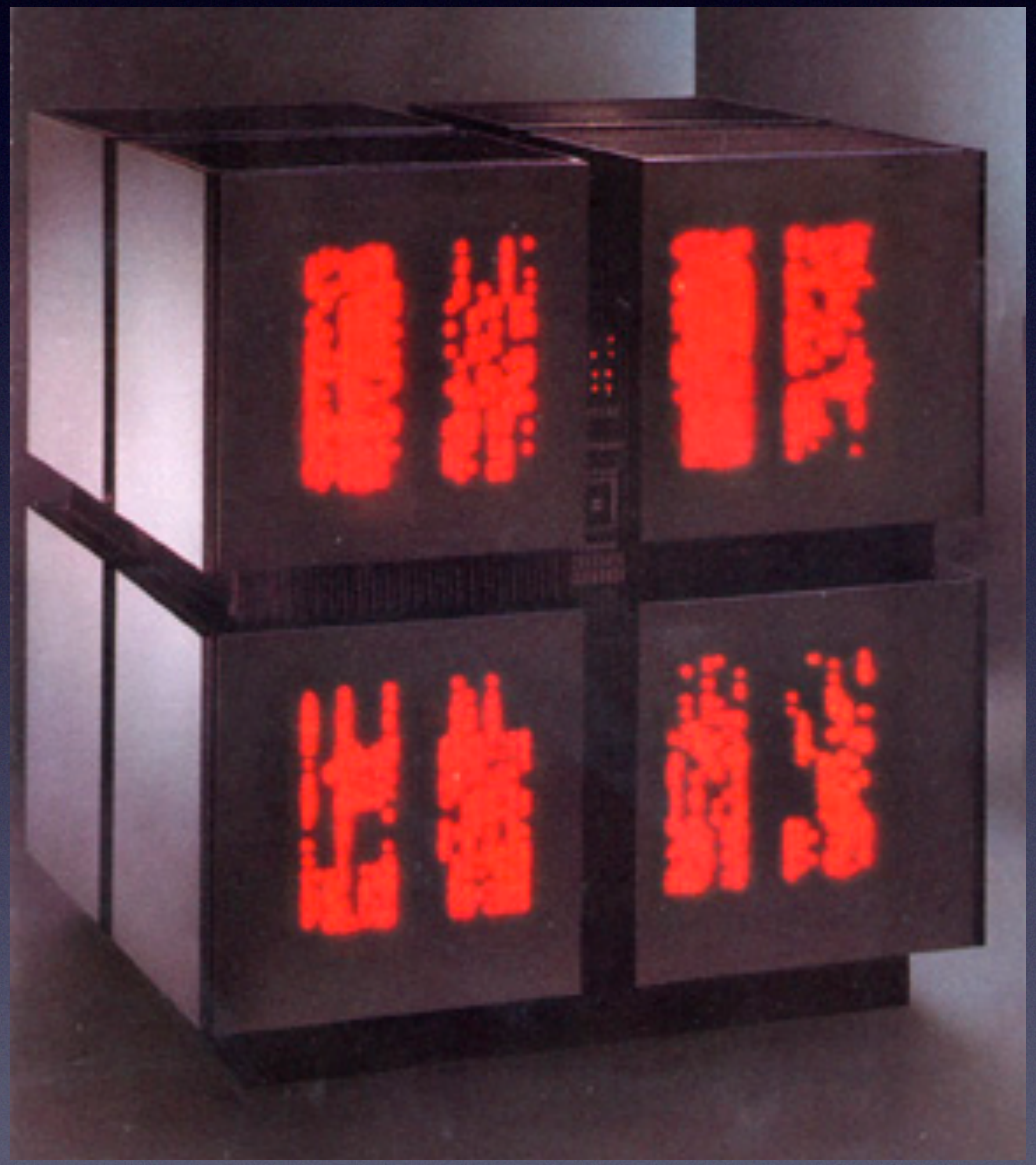
BNL

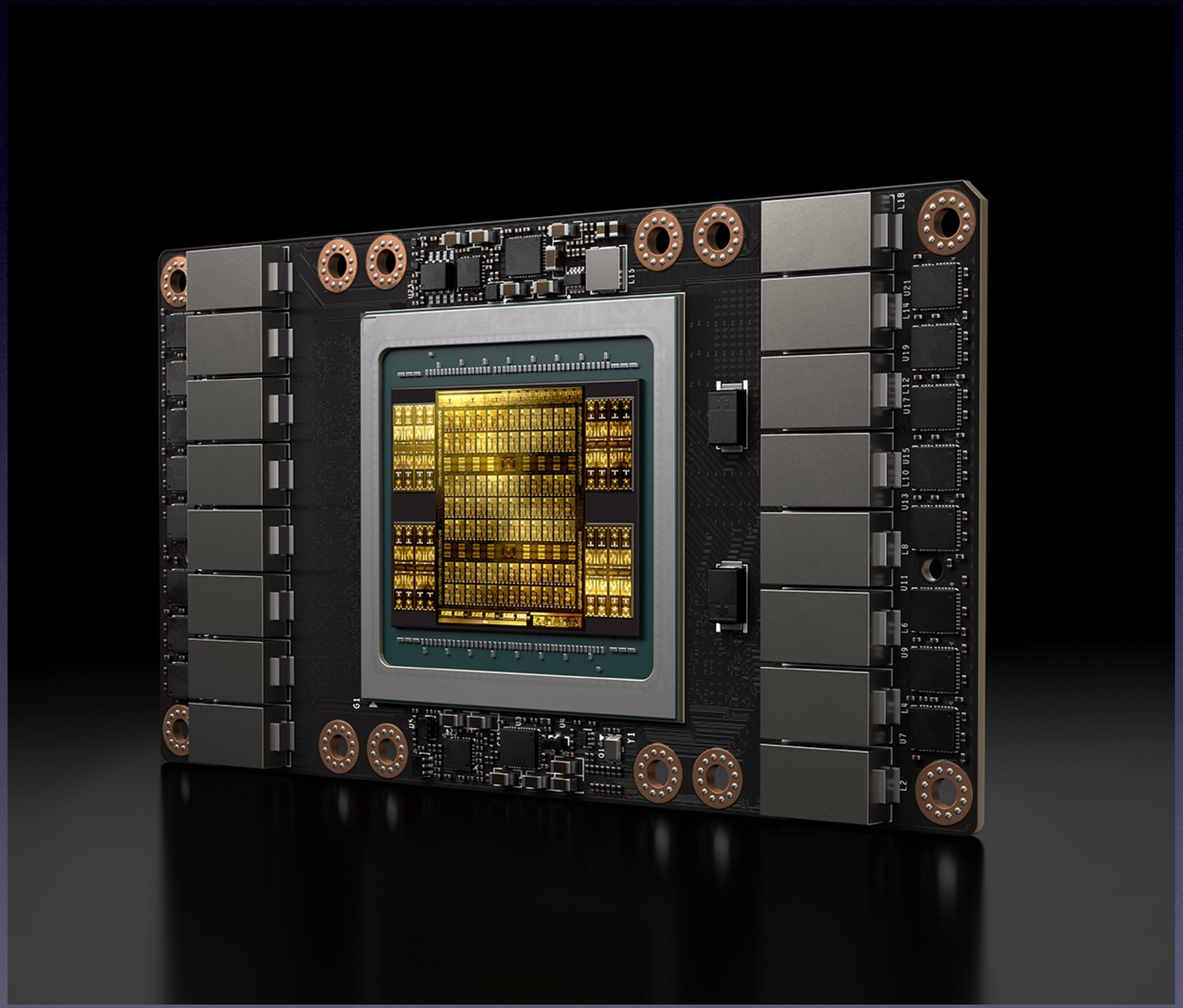
Intel Clusters

IBM BlueGene

Disruptive many-core Architectures

Now the Nvidia Volta chip





64k K bit serial PE.

0.002 Teraflops

5000 x 32 bit PE = 160K bits

20.0 Teraflops on single chip!

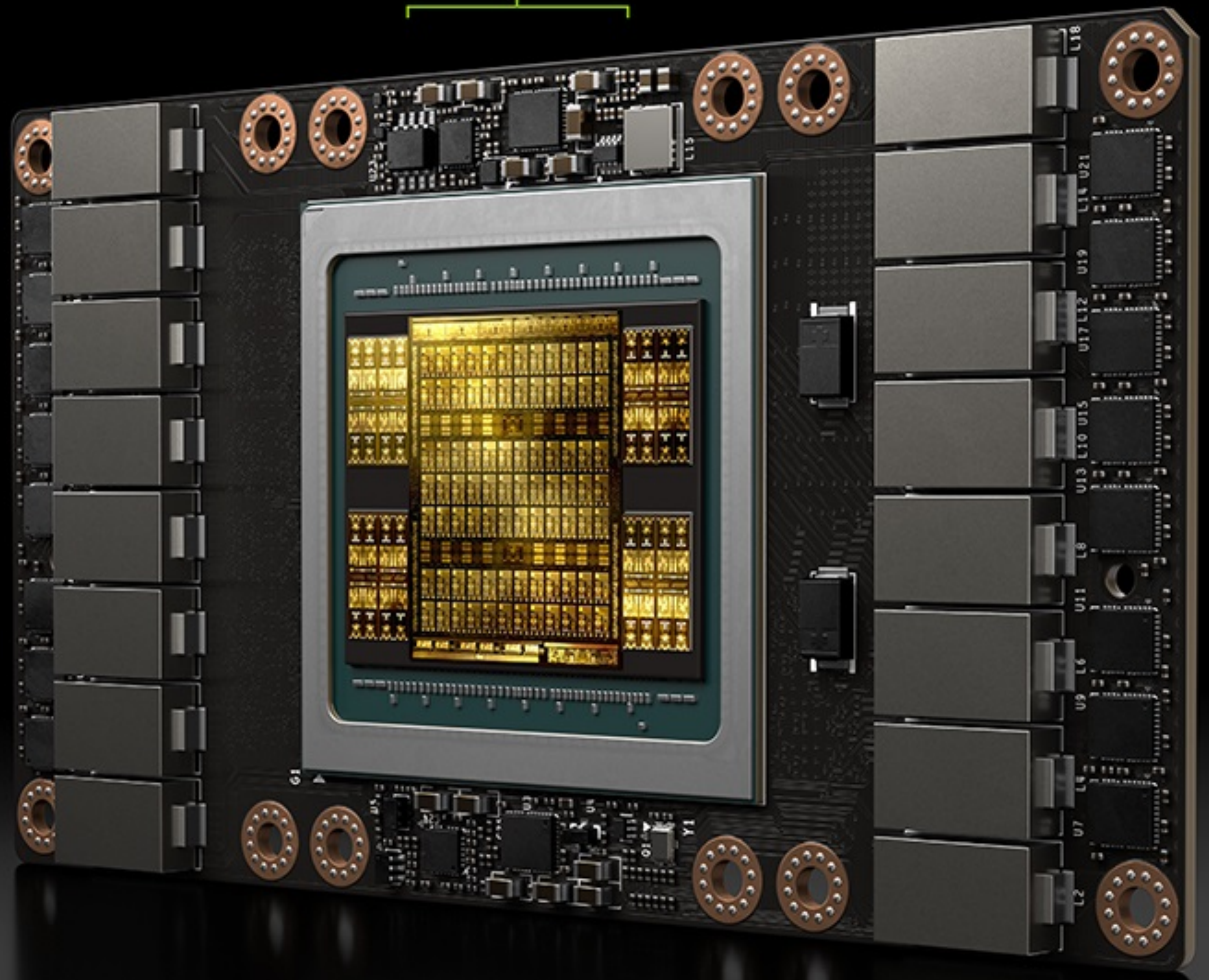
NVIDIA TESLA V100

World's First Fused HPC and AI Processor

HIGH-SPEED NVIDIA NVLINK™ TO GPUs, IBM POWER
300GB/s NVLink

NVIDA VOLTA™ TENSOR CORE GPU
640 TENSOR CORES
125 TFLOPS Tensor Ops

5120 NVIDIA CUDA® CORES
15.7 TFLOPS FP32
7.8 TFLOPS FP64

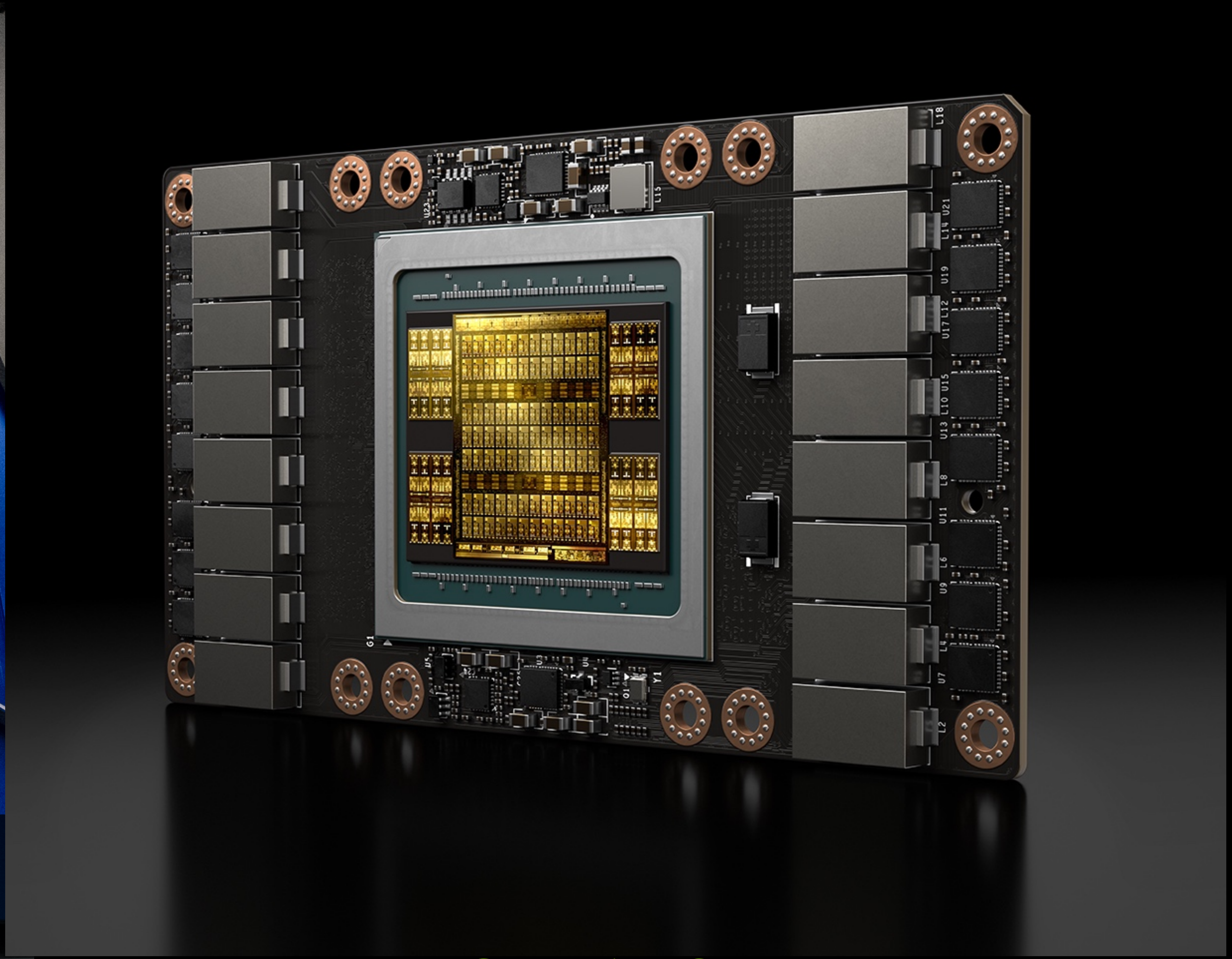


MEMORY
32GB/16GB HBM2

TENSOR CORE GPU | 21 BILLION TRANSISTORS | 125 TFLOPS | REVOLUTIONARY HPC AND AI PERFORMANCE

NVIDIA POWERS WORLD'S FASTEST SUPERCOMPUTERS

Summit Becomes First System To Scale The 100 Petaflops Milestone



122 PF
HPC

3 EF
AI

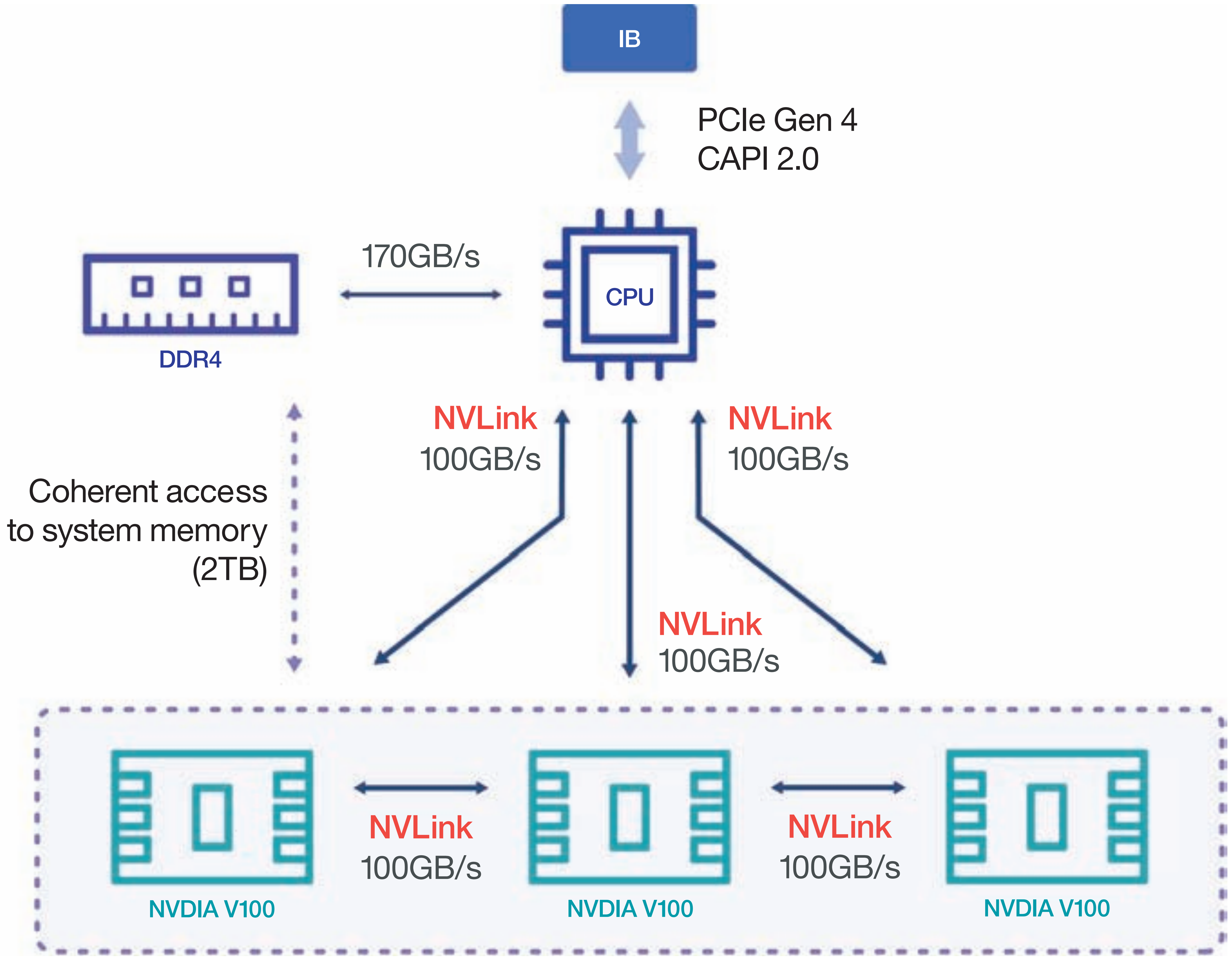
27,648
Volta Tensor Core GPUs

IBM AC 922: SUMMIT NODE

4/6 V100 GPUs

NVLink to GPU and P9

2 EDR IB



SOFTWARE

The pioneers of GPU computing in lattice QCD

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Computer Physics Communications 177 (2007) 631–639

Computer Physics Communications

www.elsevier.com/locate/cpc

Lattice QCD as a video game

Győző I. Egri [a], Zoltán Fodor [a,b,c,*], Christian Hoelbling [b], Sándor D. Katz [a,b], Dániel Nógrádi [b], Kálmán K. Szabó [b]

[a] *Institute for Theoretical Physics, Eötvös University, Budapest, Hungary*
[b] *Department of Physics, University of Wuppertal, Germany*
[c] *Department of Physics, University of California, San Diego, USA*

Received 2 February 2007; received in revised form 29 May 2007; accepted 7 June 2007

Available online 15 June 2007

Abstract

The speed, bandwidth and cost characteristics of today's PC graphics cards make them an attractive target as general purpose computational platforms. High performance can be achieved also for lattice simulations but the actual implementation can be cumbersome. This paper outlines the architecture and programming model of modern graphics cards for the lattice practitioner with the goal of exploiting these chips for Monte Carlo simulations. Sample code is also given.

CUDA BREAK THROUGH AT NVIDIA

CUDA PROGRAMMING

```
void saxpy_serial(int n, float a, float *x, float *y)
{
    for (int i = 0; i < n; ++i)
        y[i] = a*x[i] + y[i];
}
// Invoke serial SAXPY kernel
saxpy_serial(n, 2.0, x, y);
```

Standard C Code

```
__global__ void saxpy_parallel(int n, float a, float *x, float *y)
{
    int i = blockIdx.x*blockDim.x + threadIdx.x;
    if (i < n)  y[i] = a*x[i] + y[i];
}
// Invoke parallel SAXPY kernel with 256 threads/block
int nblocks = (n + 255) / 256;
saxpy_parallel<<<nblocks, 256>>>(n, 2.0, x, y);
```

Parallel C Code

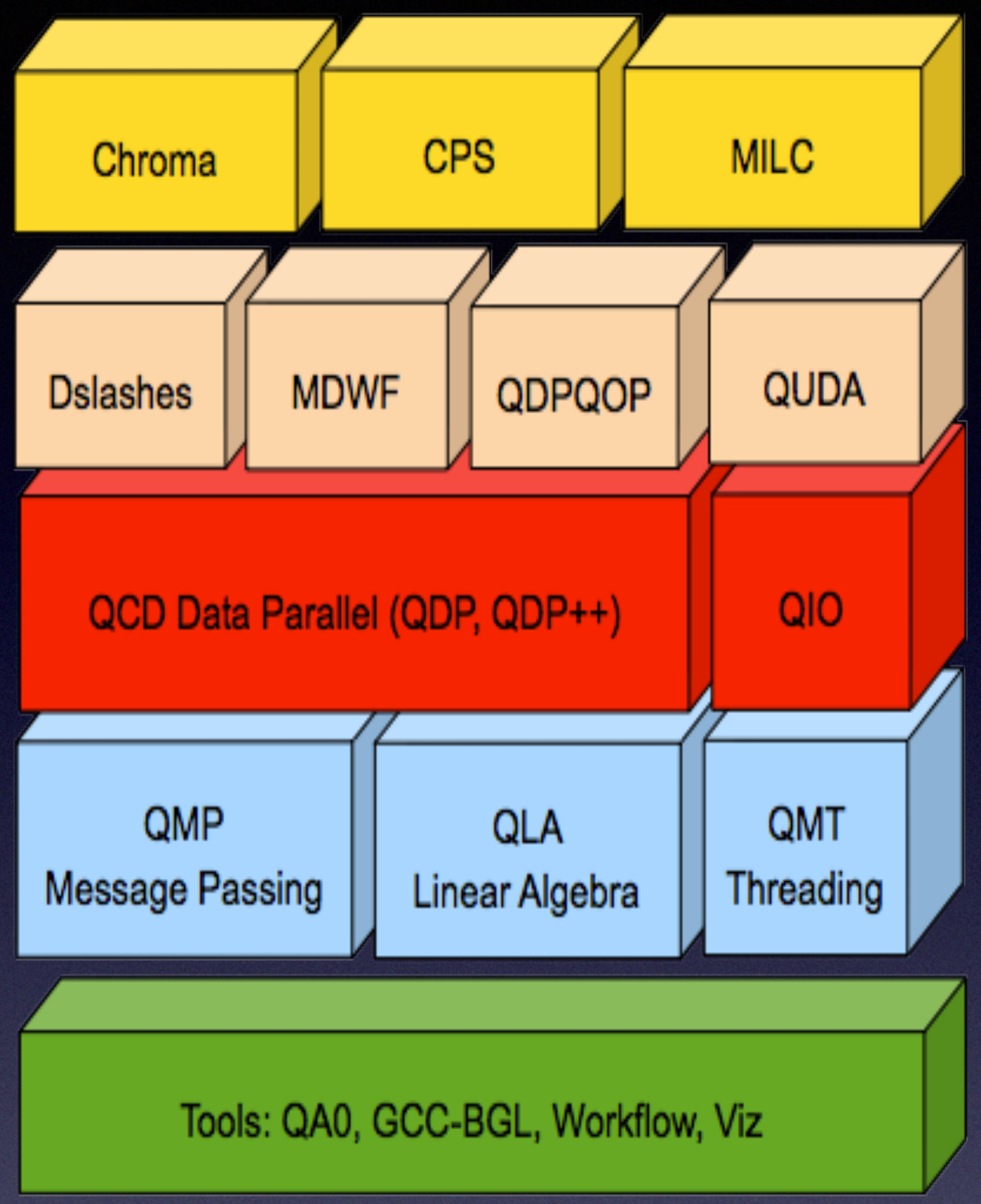
After 10 years

Enter QUDA

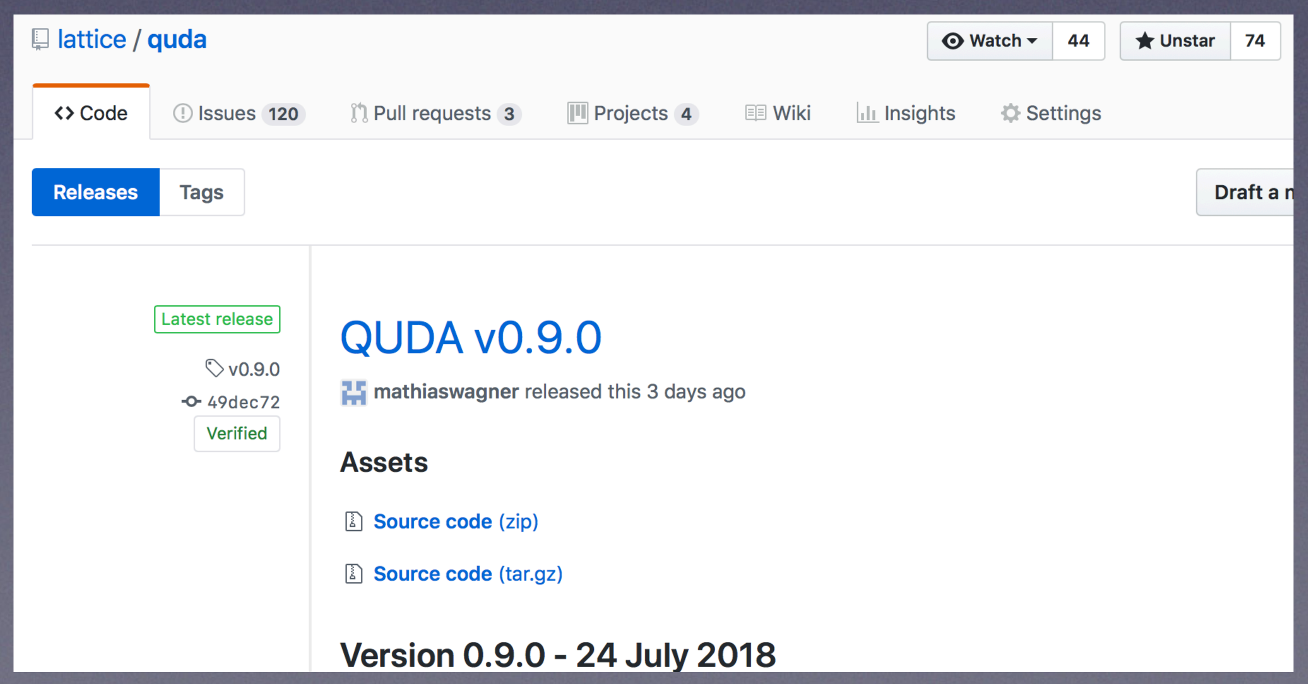
- "QCD on CUDA" – http://lattice.github.com/quda
- Effort started at Boston University in 2008, now in wide use as the GPU backend for BQCD, Chroma, CPS, MILC, TIFR, etc.
- Provides:
  - Various solvers for all major fermonic discretizations, with multi-GPU support
  - Additional performance-critical routines needed for gauge-field generation
- Maximize performance / Minimize time to science
  - Exploit physical symmetries to minimize memory traffic
  - Mixed-precision methods
  - Autotuning for high performance on all CUDA-capable architectures
  - Domain-decomposed (Schwarz) preconditioners for strong scaling
  - Eigenvector solvers (Lanczos and EigCG) new!
  - Multigrid solvers for optimal convergence new!

QUDA Community over the 10 years

- Ron Babich (NVIDIA)
- Simone Bacchio (Cyprus)
- Michael Baldhauf (Regensburg)
- Kip Barros (LANL)
- Rich Brower (Boston University)
- Nuno Cardoso (NCSA)
- Kate Clark (NVIDIA)
- Michael Cheng (Boston University)
- Carleton DeTar (Utah University)
- Justin Foley (Utah -> NIH)
- Joel Giedt (Rensselaer Polytechnic Institute)
- Arjun Gambhir (William and Mary)
- Steve Gottlieb (Indiana University)
- Kyriakos Hadjiyiannakou (Cyprus)
- Dean Howarth (BU)
- Bálint Joó (Jlab)
- Hyung-Jin Kim (BNL -> Samsung)
- Bartek Kostrzewa (Bonn)
- Claudio Rebbi (Boston University)
- Hauke Sandmeyer (Bielefeld)
- Guochun Shi (NCSA -> Google)
- Mario Schröck (INFN)
- Alexei Strelchenko (FNAL)
- Jiqun Tu (Columbia)
- Alejandro Vaquero (Utah University)
- Mathias Wagner (NVIDIA)
- Evan Weinberg (NVIDIA)
- Frank Winter (Jlab)

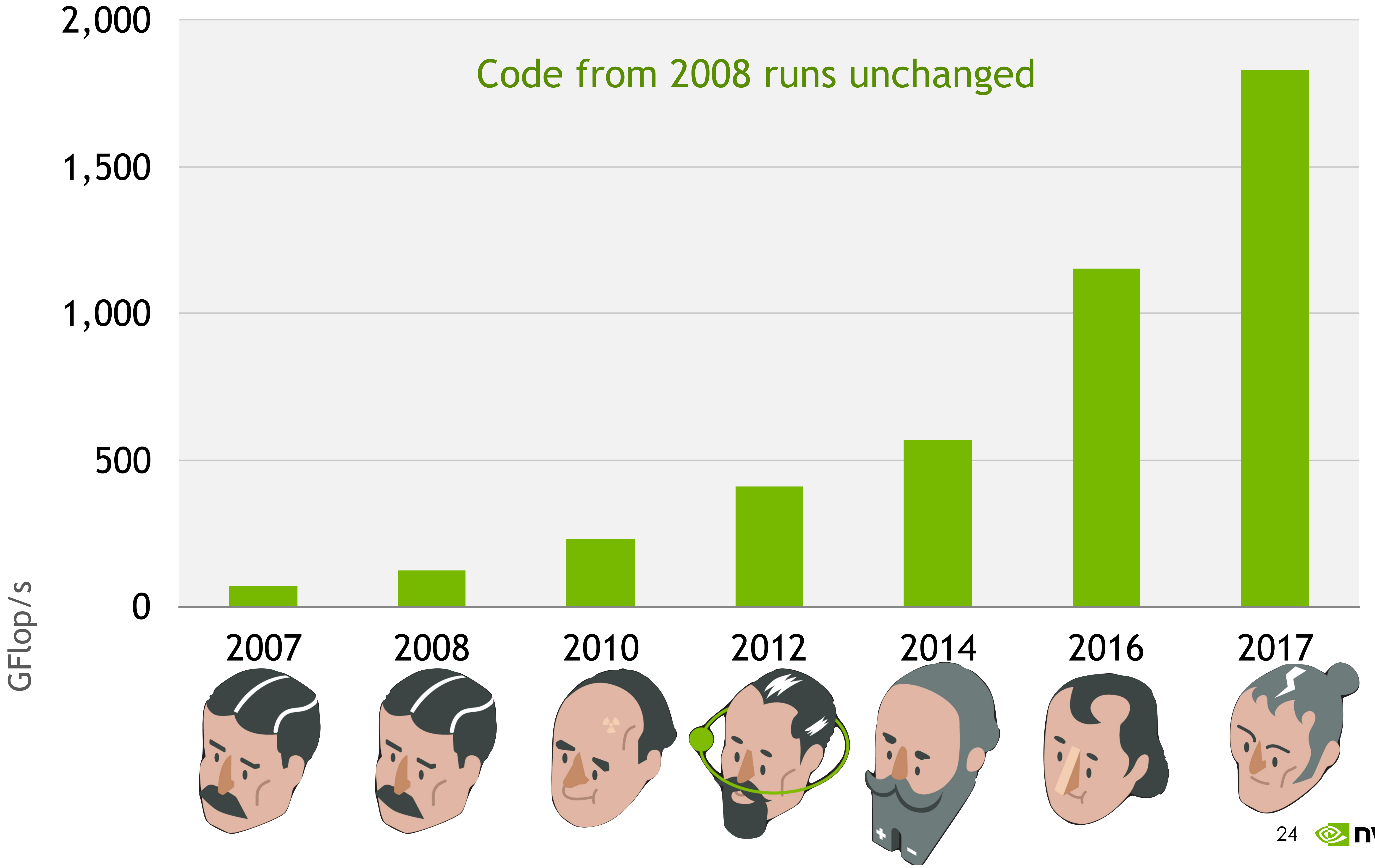


http://lattice.github.com/quda, BSD license



RECOMPILE AND RUN

Autotuning provides performance portability



Data "compression" to reduce memory traffic

- Reconstruct SU(3) matrices with 18 reals from 12 or 8 the fly, e.g.,

- $$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix}$$

$$\mathbf{c} = (\mathbf{a} \times \mathbf{b})^*$$

$$SU(2) : U = a_0\sigma_0 + \vec{a} \cdot \sigma$$

$$a_0^2 + \vec{a} \cdot \vec{a} = 1 \implies S_3 \text{ sphere}$$

- Better still 8 parameters manifold on S3 x S5

- Choose a 4th gamma.

$$P^{\pm 4} = \begin{pmatrix} 1 & 0 & \pm 1 & 0 \\ 0 & 1 & 0 & \pm 1 \\ \pm 1 & 0 & 1 & 0 \\ 0 & \pm 1 & 0 & 1 \end{pmatrix} \rightarrow \left\{ \begin{array}{l} P^{+4} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ P^{-4} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \end{array} \right\}$$

- Fix to the temporal gauge (t-direction U link to the identity).

- Load 24 Dirac Fields as large Floating point plus 23 16-bit integers

Mixed precision with reliable updates

- New idea is to apply this approach to mixed precision. (Clark et al., arXiv:0911.3191)

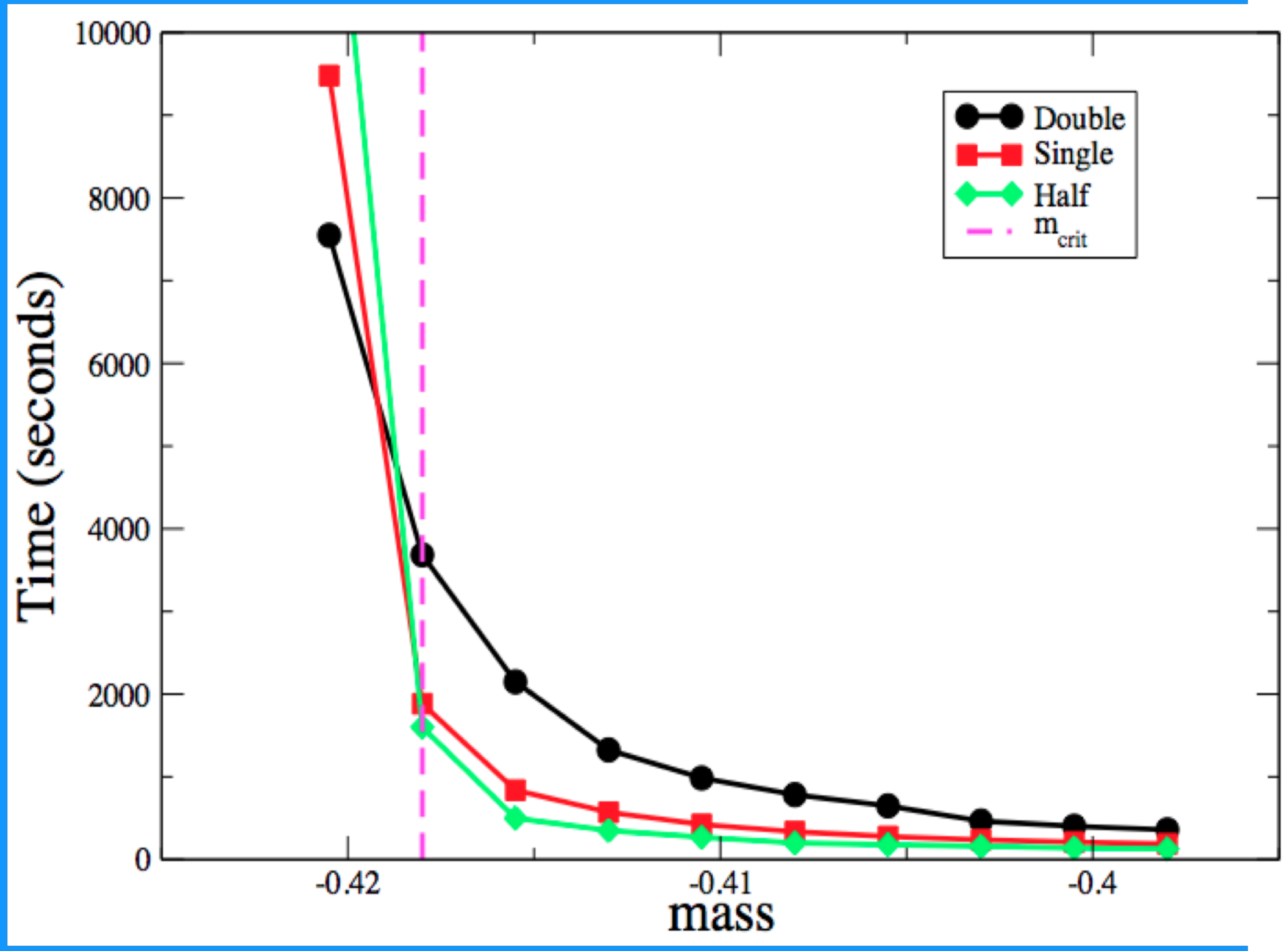
$r_0 = b - Ax_0;$
$\hat{r}_0 = r;$
$\hat{x}_0 = 0;$
$k = 0;$
while $||\hat{r}_k|| > \epsilon$ **do**
  Low precision solver iteration: $\hat{r}_k \to \hat{r}_{k+1}, \hat{x}_k \to \hat{x}_{k+1};$
  if $||\hat{r}_{k+1}|| < \delta M(\hat{r})$ **then**
    $x_{l+1} = x_l + \hat{x}_{k+1};$
    $r_{l+1} = b - Ax_{l+1}\;;$
    $\hat{x}_{k+1} = 0;$
    $\hat{r}_{k+1} = r;$
    $l = l + 1;$
  end
  $k = k + 1\;;$
end

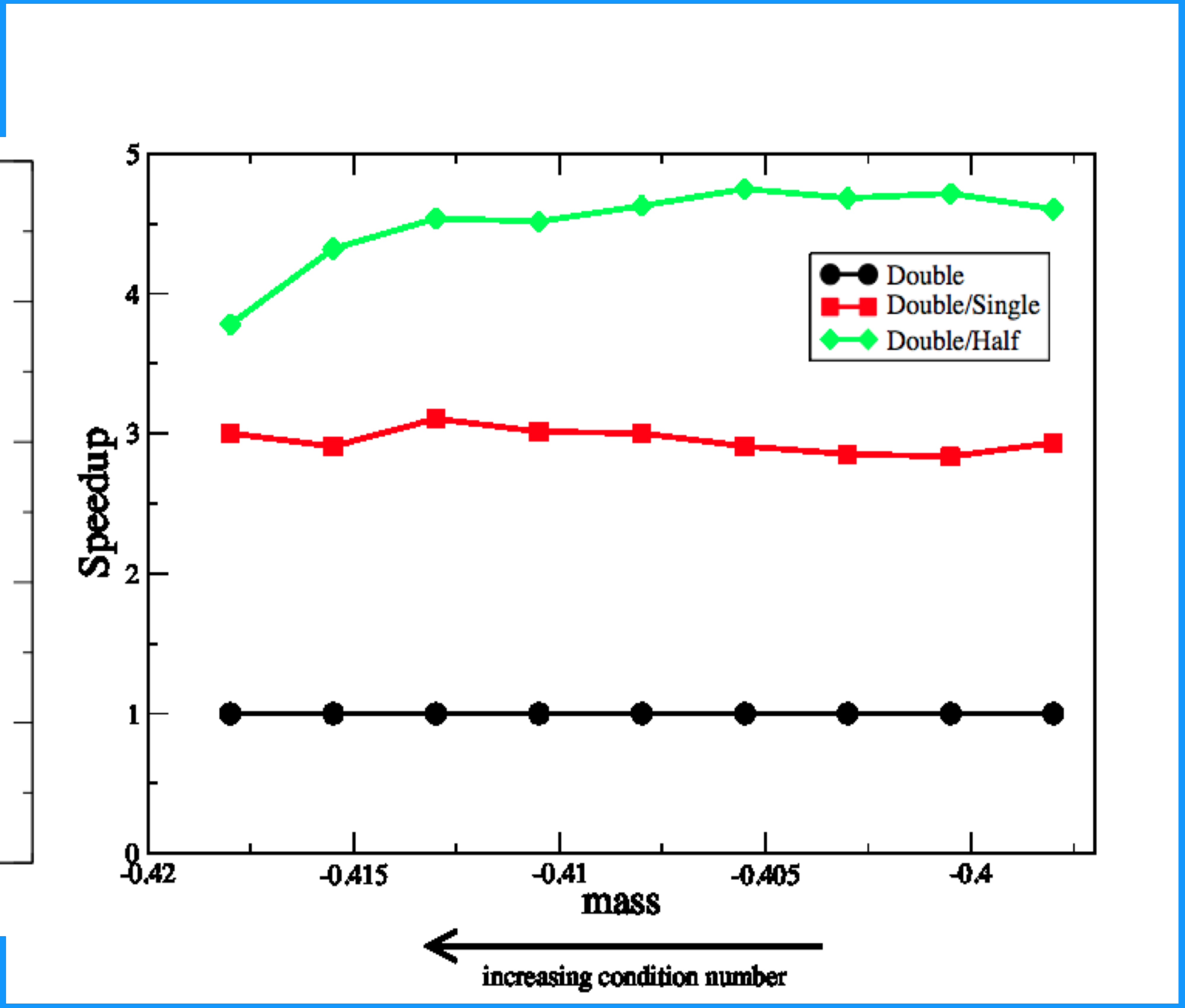
- $\hat{}$ denotes reduced precision.
- δ is a parameter determining the frequency of updates.
- $M(\hat{r})$ denotes the maximum iterated residual since the last update.

- Reliable updates seems to win handily at light quark masses (and is no worse than iterative refinement at heavy masses).

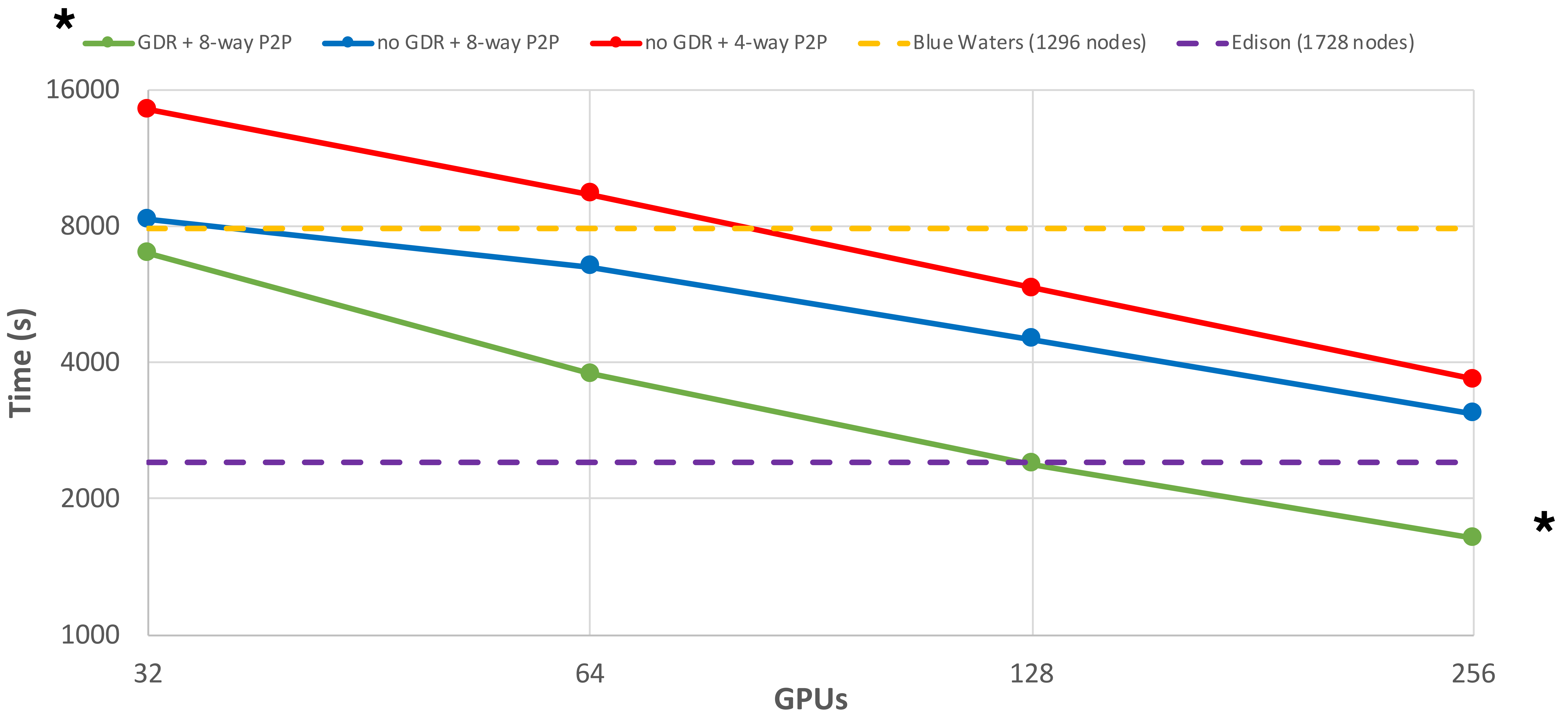
Mixed precision with reliable updates

- Using a mixed-precision solver incorporating "reliable updates" (Clark et al., arXiv:0911.3191) with half precision greatly reduces time-to-solution while maintaining double precision accuracy.





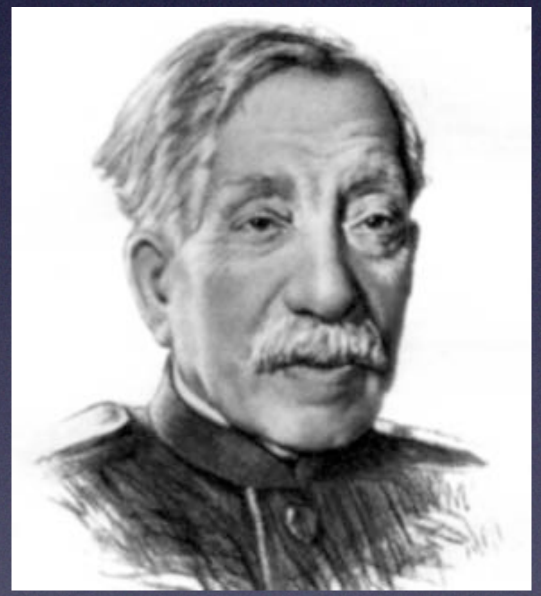
MILC Large APEX benchmark



*** GDR: GPU Direct RDMA**

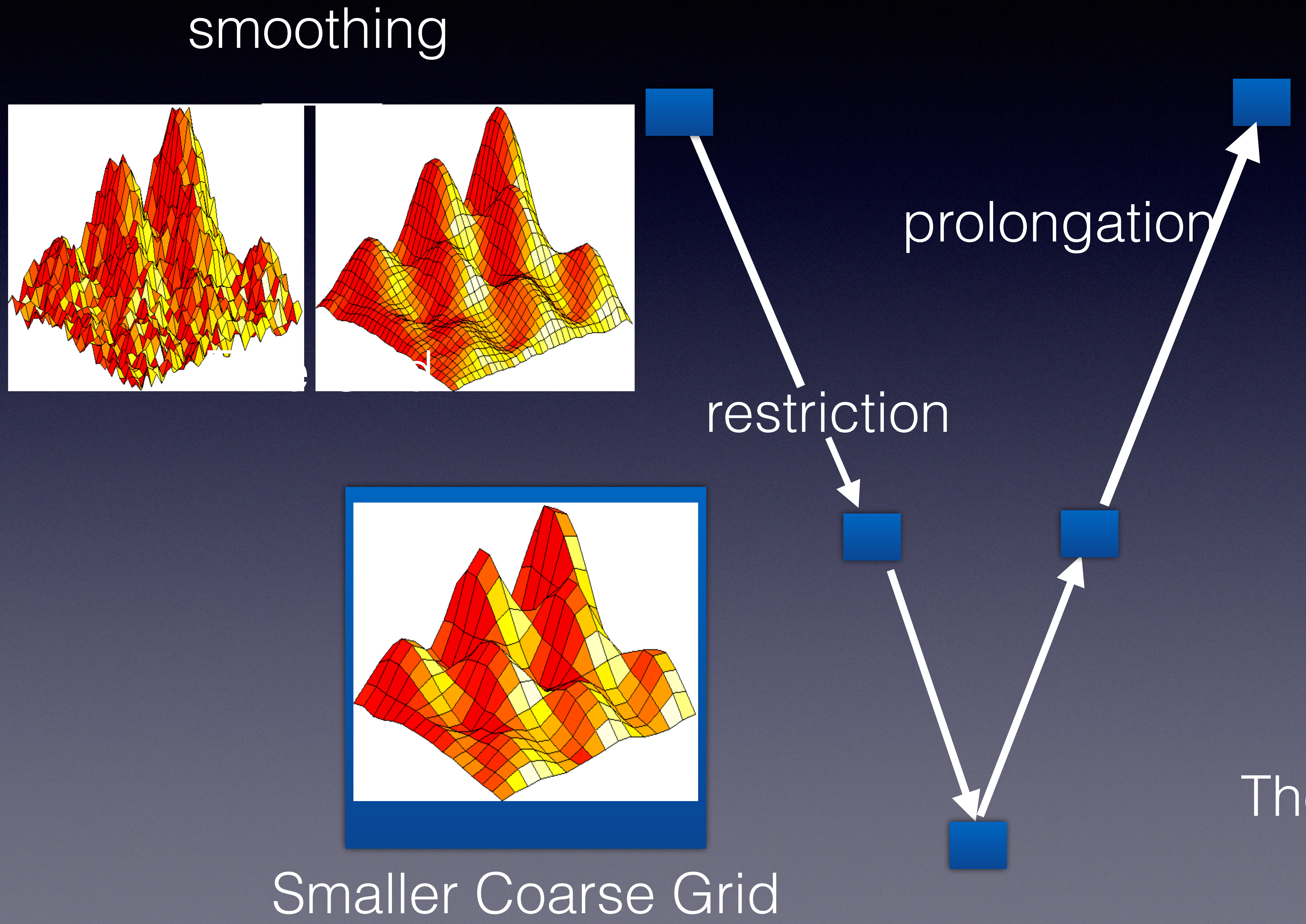
Algorithms: Must use the best

The most efficient ***algorithms*** exploit multiple scales



MULTIGRID
Renormalization Group
Domain Decomposition

Adaptive Smooth Aggregation Algebraic Multigrid



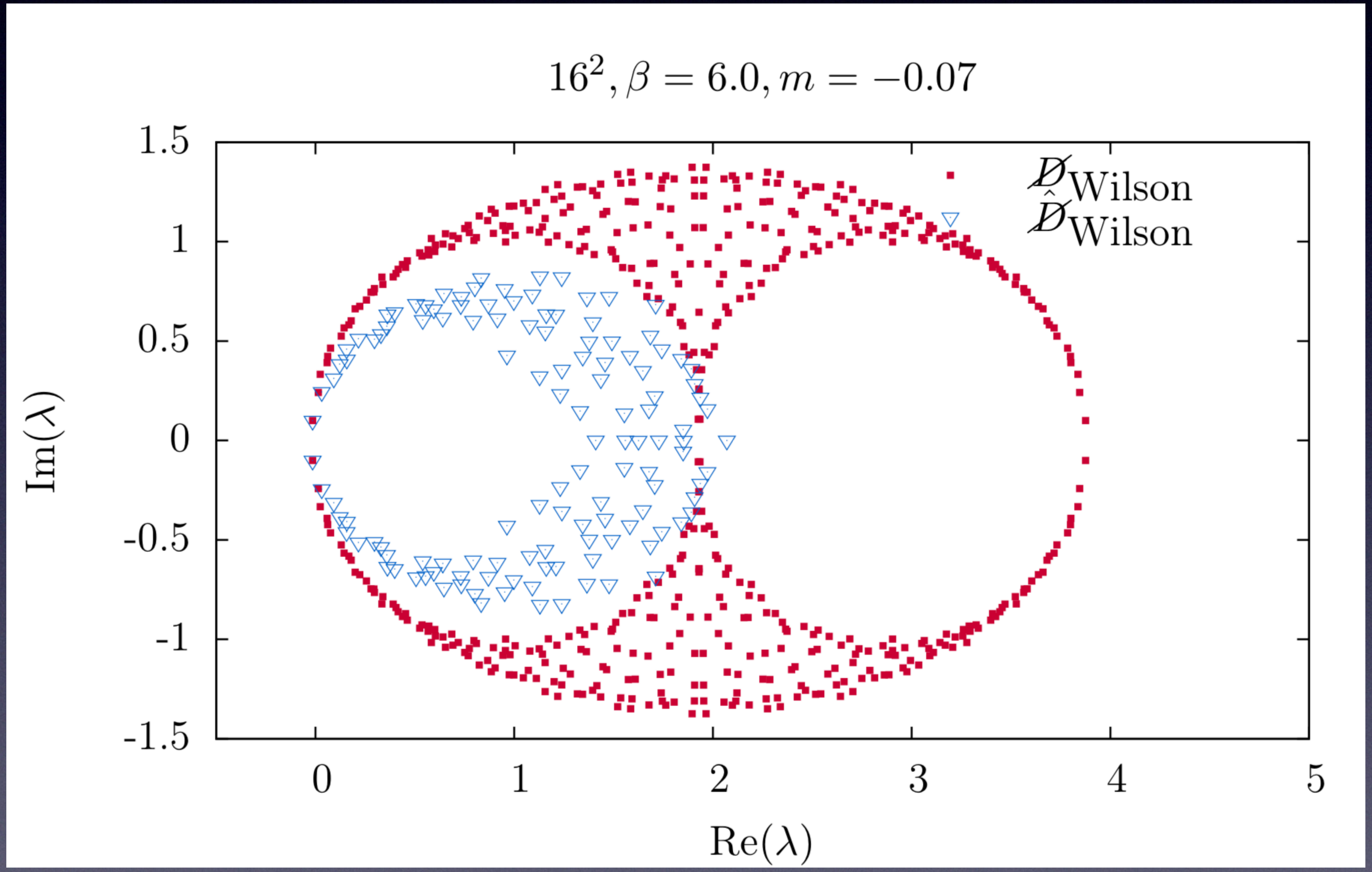
Slow convergence of Dirac solver is due to small eigenvalues

Spilt the vector space into near null space $\mathcal{S}$ *and the complement* $\mathcal{S}_\perp$

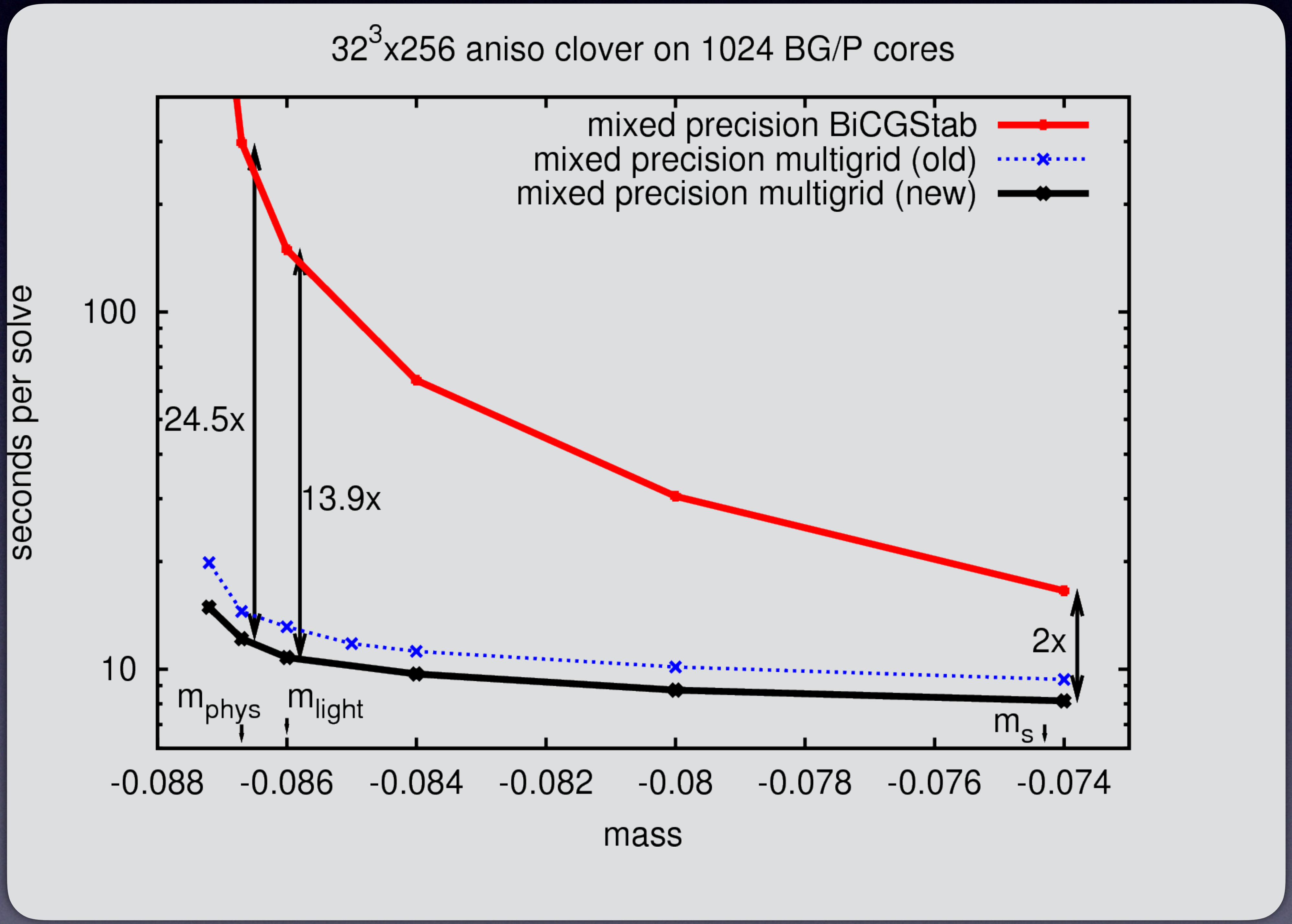
The Multigrid V-cycle

Wilson Dirac Multigrid: 2010

Preservation of Spectrum



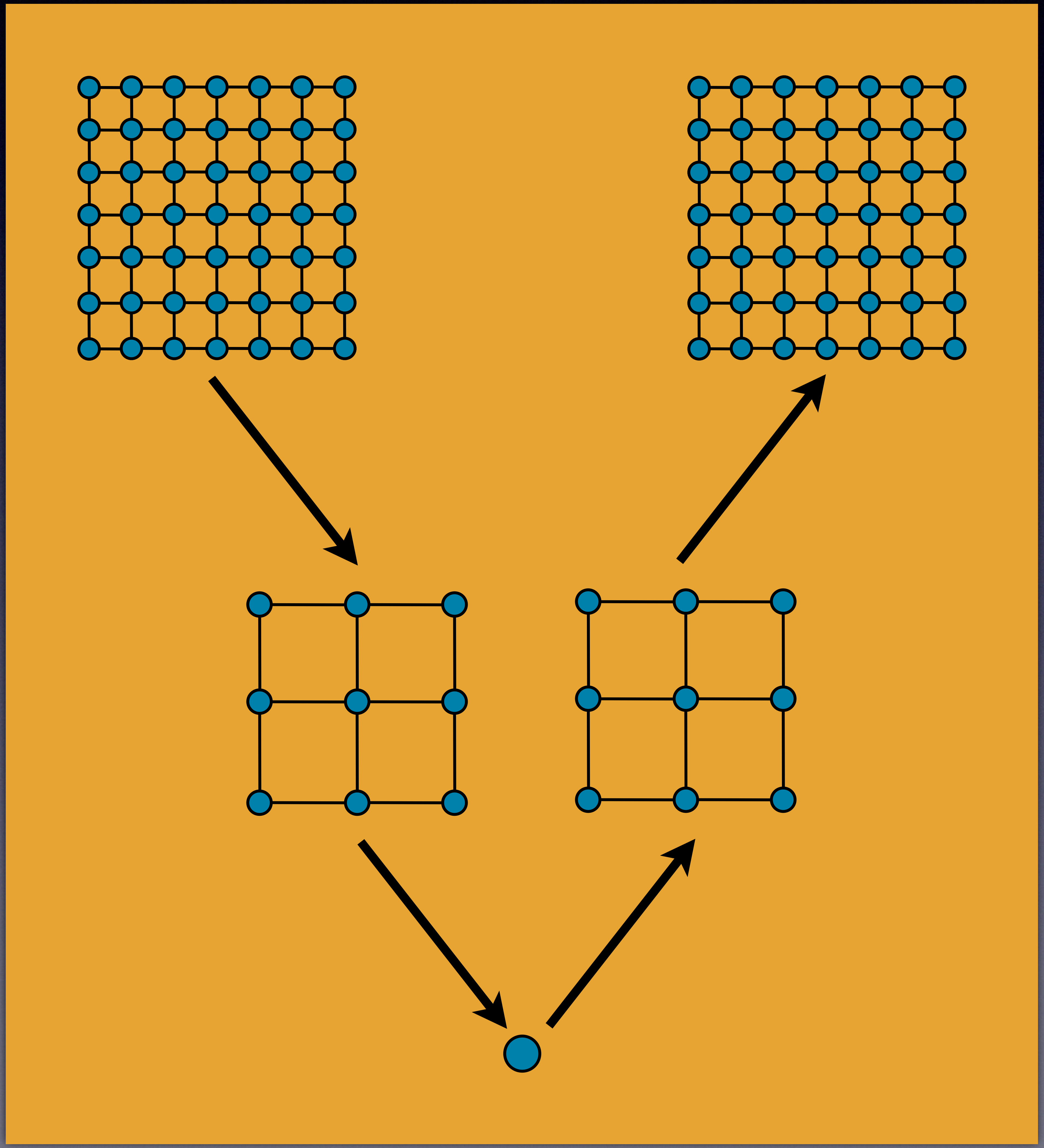
Speed in depend of Quark Mass



"Adaptive multigrid algorithm for the lattice Wilson-Dirac operator" R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. Manteuffel, S. McCormick, J. C. Osborn, and C. Rebbi, PRL. (2010).

The Challenge of Multigrid on GPU

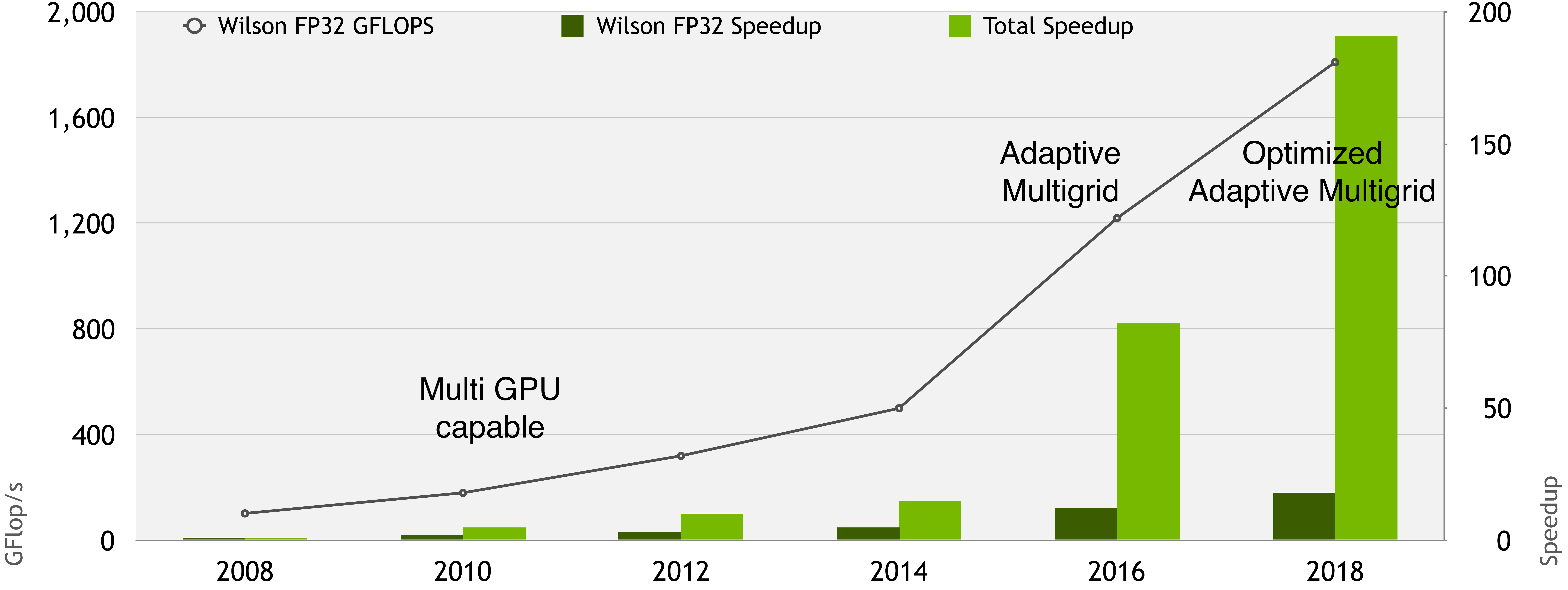
Find Grain Parallelism below on thread per lattice site



- GPU requirements very different from CPU
  - Each thread is slow, but O(10,000) threads per GPU
- Fine grids run very efficiently
  - High parallel throughput problem
- Coarse grids are worst possible scenario
  - More cores than degrees of freedom
  - Increasingly serial and latency bound
  - Little's law (bytes = bandwidth * latency)
  - Amdahl's law limiter
- Multigrid exposes many of the problems expected at the Exascale

NODE PERFORMANCE OVER TIME

Multiplicative speedup through software and hardware



Time to solution is measured time to solution for solving the Wilson operator against a random source on a 24x24x24x64 lattice, β=5.5, M_π= 416 MeV. One node is defined to be 3 GPUs

CHROMA HMC MULTIGRID

HMC typically dominated by solving the Dirac equation, but

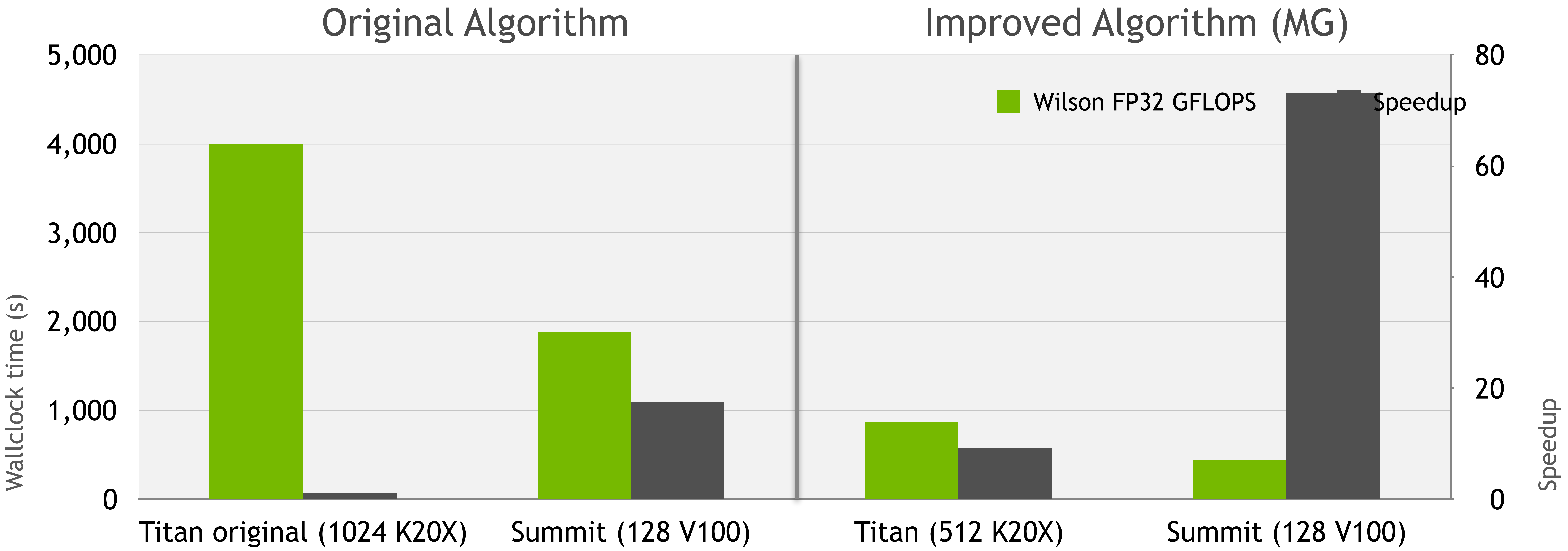
- Few solves per linear system
- Can be bound by heavy solves (c.f. Hasenbusch mass preconditioning)

Multigrid setup must run at speed of light

- Reuse and evolve multigrid setup where possible
- Use the same null space for all masses (setup run on lightest mass)
- Evolve null space vectors as the gauge field evolves (Lüscher 2007)
- Update null space when the preconditioner degrades too much on lightest mass

MULTI-GRID ON SUMMIT

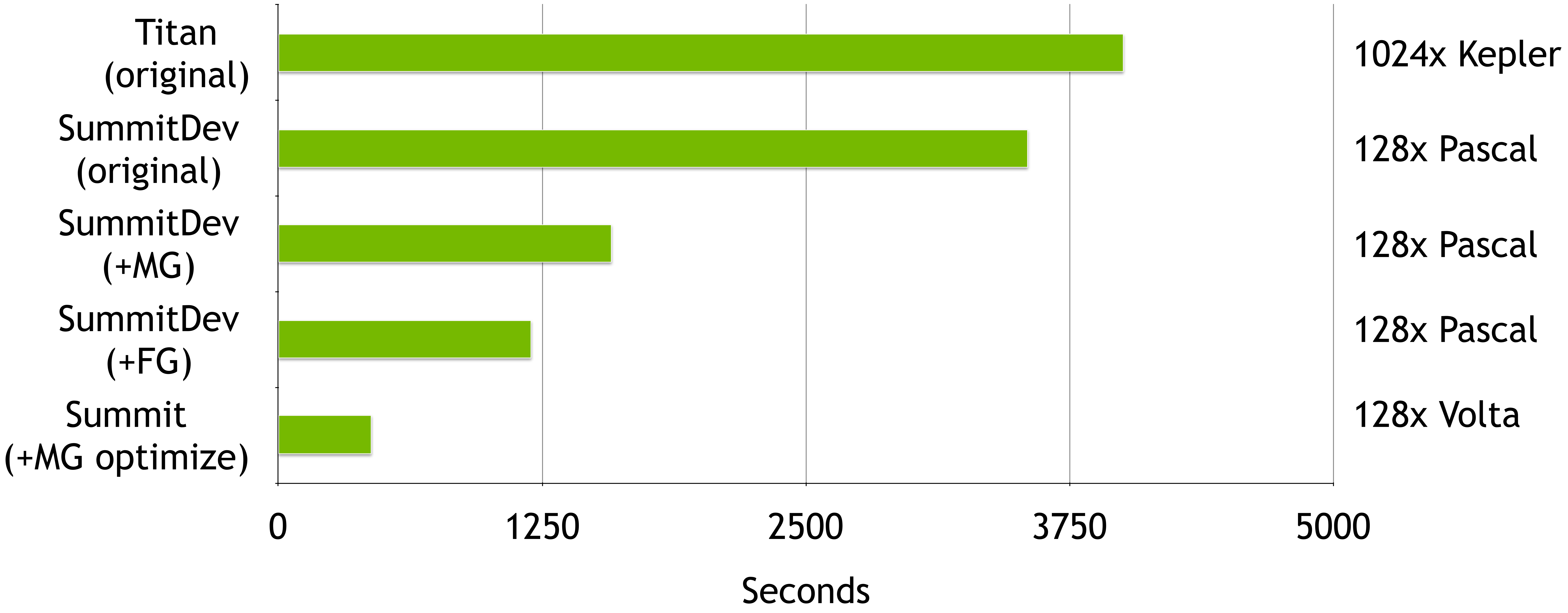
Full Chroma Hybrid Monte Carlo



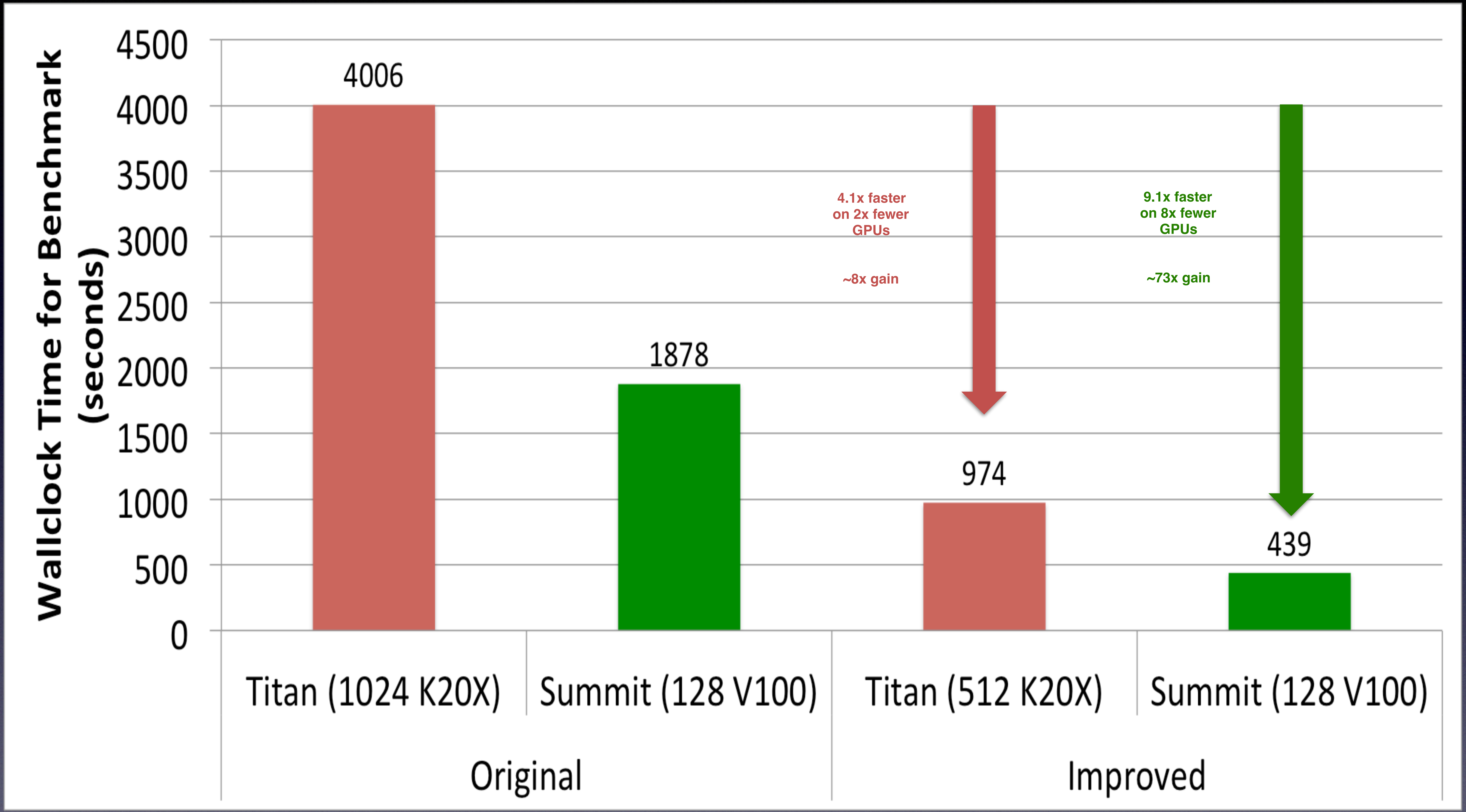
Data from B. Joo (Jefferson Lab). Chroma w/ QDP-JIT (F. Winter, Jefferson Lab) and QUDA.

B. Joo gratefully acknowledges funding through the US DOE SciDAC program (DE-AC05-06OR23177)

HMC SPEEDUP PROGRESSION



100 X reduction in GPU-hours to generate gauge configurations



- Remarkable achievement in Wilson-Clover gauge configuration generation
- Requires multiple sparse-matrix solves during molecular dynamics evolution
- Incorporated new Wilson-Clover adaptive multigrid solver & force gradient integrator
- Retuned molecular dynamics to accommodate

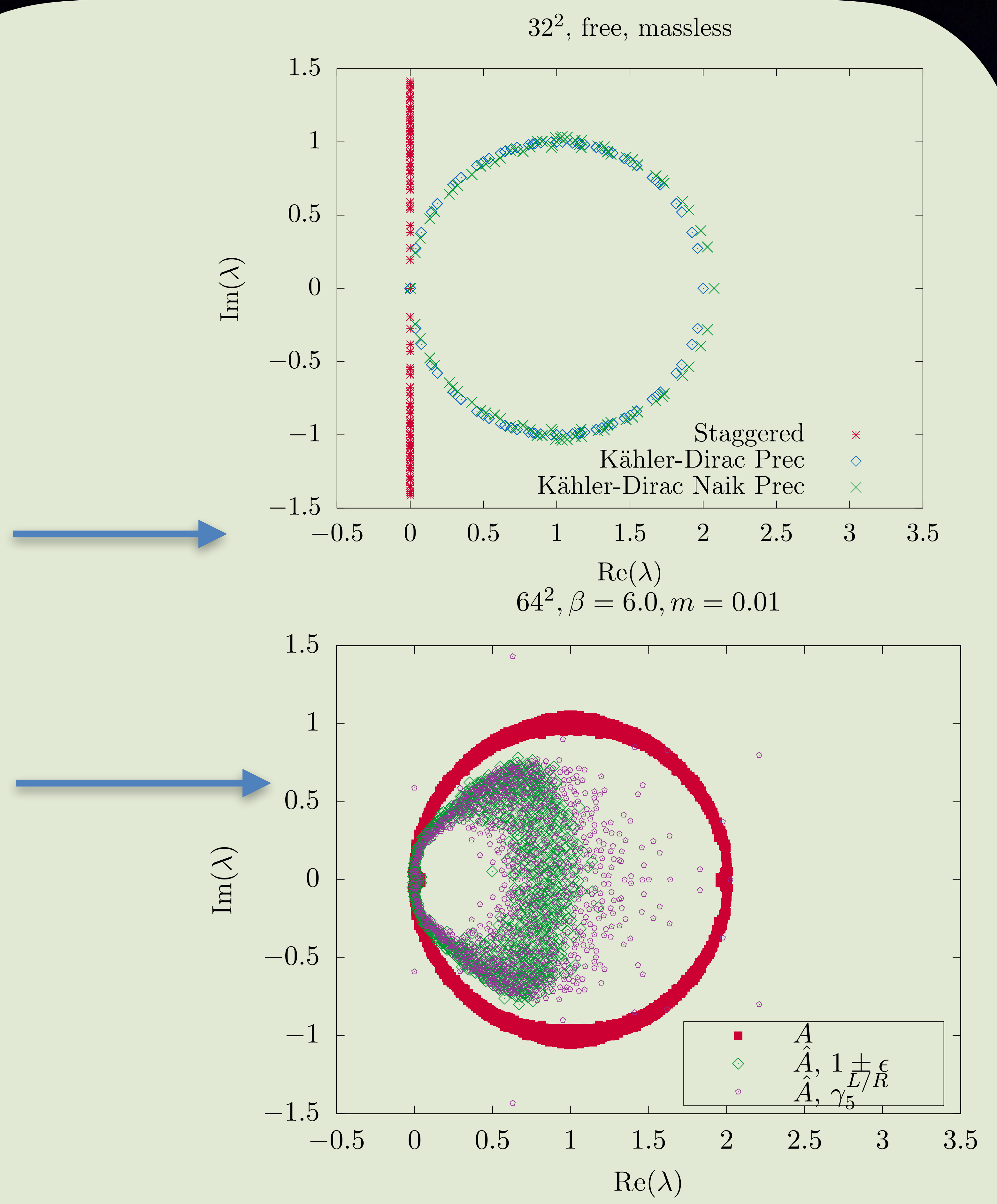
Kahler Dirac and Stagger MG

- An algorithmic breakthrough for Staggered MG
- promises to reduce significantly the cost of calculations with this quark action

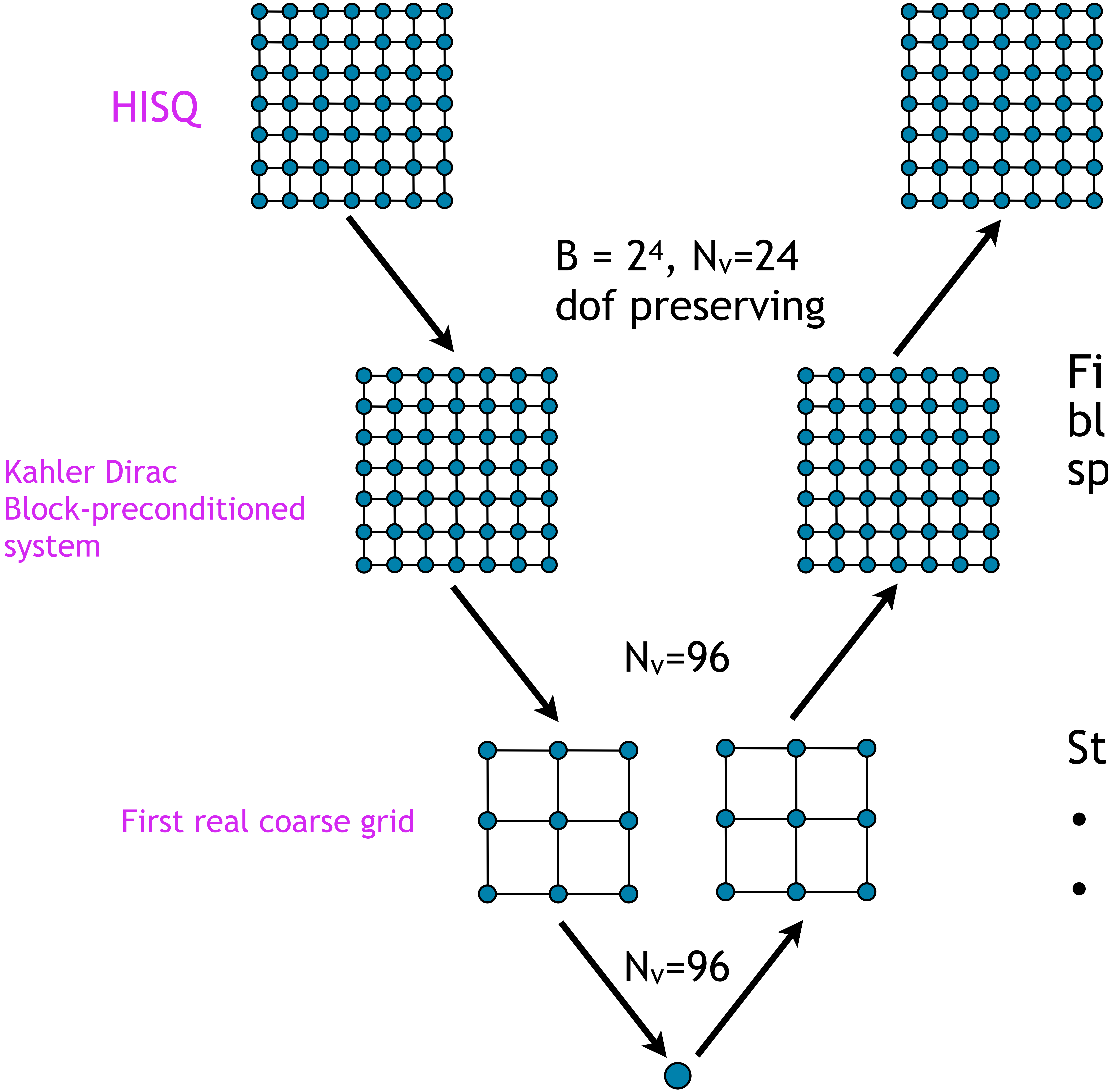
  - Step I: Breakthrough exploits the Kähler-Dirac (spin/taste) block to map spectrum to a complex circle.

  - Step II: Coarse level projection preserves low modes for multi-level preconditioning.

- Complete elimination of ill-condition approaching zero mass (Brower, Clark, Strelchenko & Weinberg, arXiv:1801.07823)



HISQ MG ALGORITHM



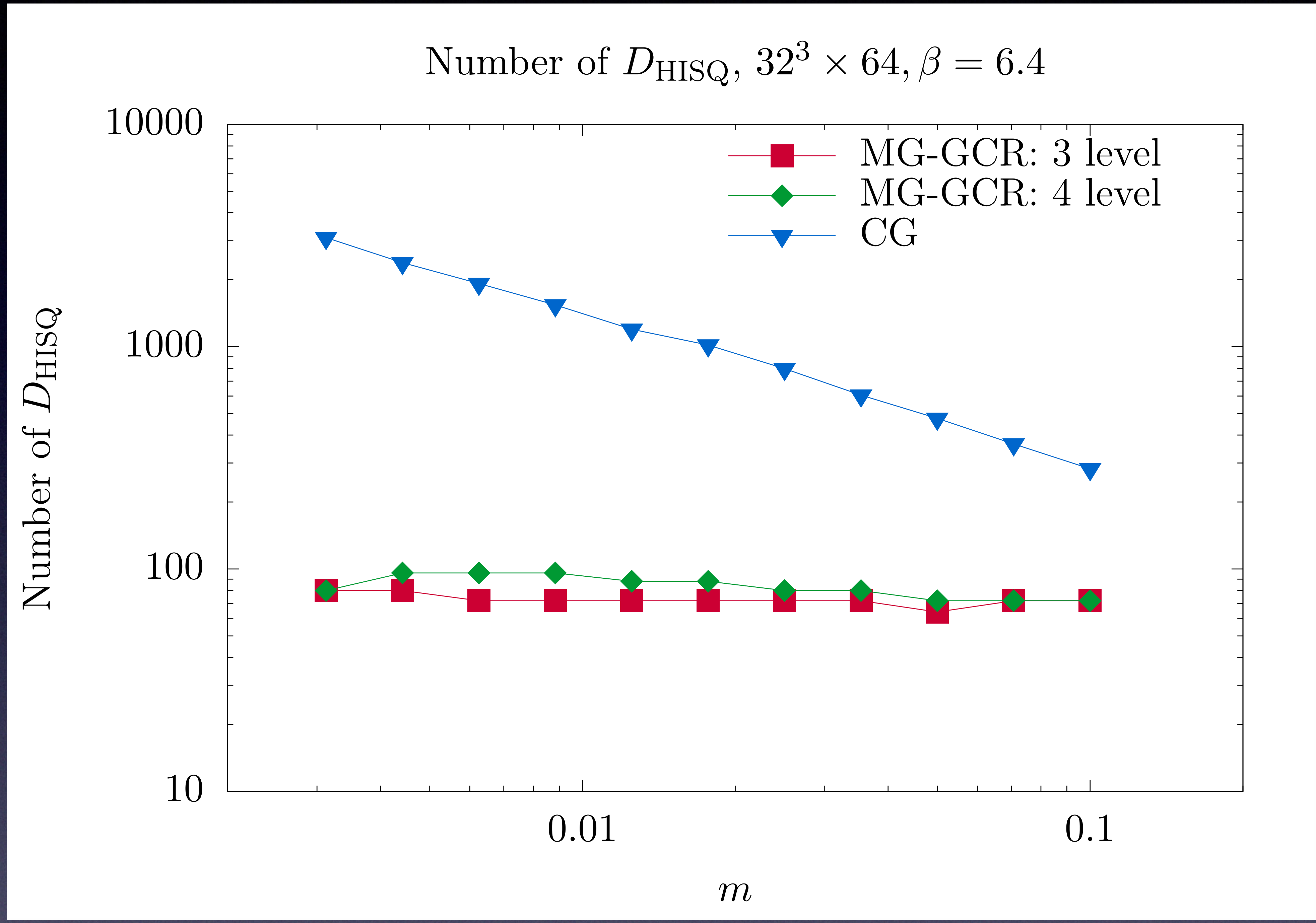
First Step is Kahler-Dirac transformation block-preconditioned to prepare spectrum for MG coursing.

Staggered has 4-fold degeneracy

- Need 4x null space vectors (N_v=24 -> 96)
- Much more memory intensive

4d QCD GPU Staggered MG in 2019

- 2-d staggered multigrid algorithm generalizes to 4-d with HISQ fermions in QCD
  - Again removal of mass dependence from the fine grid and block pre-conditioner.
  - No need to include Naik contribution when coarsening.



- On going optimization to achieve multi-GPU high performance on Summit.
  - More robust adaptive setup to deal with large null space required
  - Better approach to bottom solver (deflation, direct solve, etc.)

Exascale API and Portability

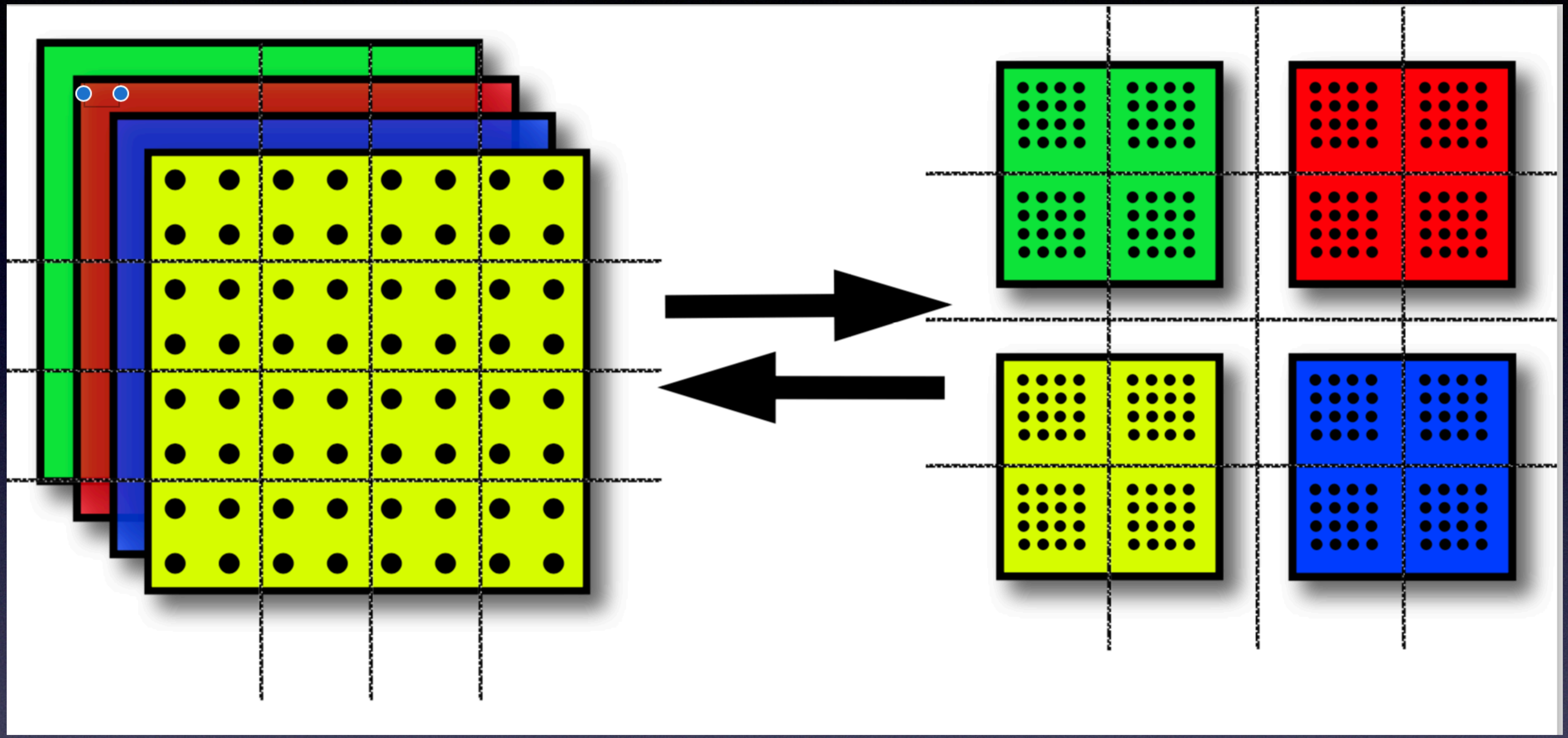
National Nuclear Security Administration

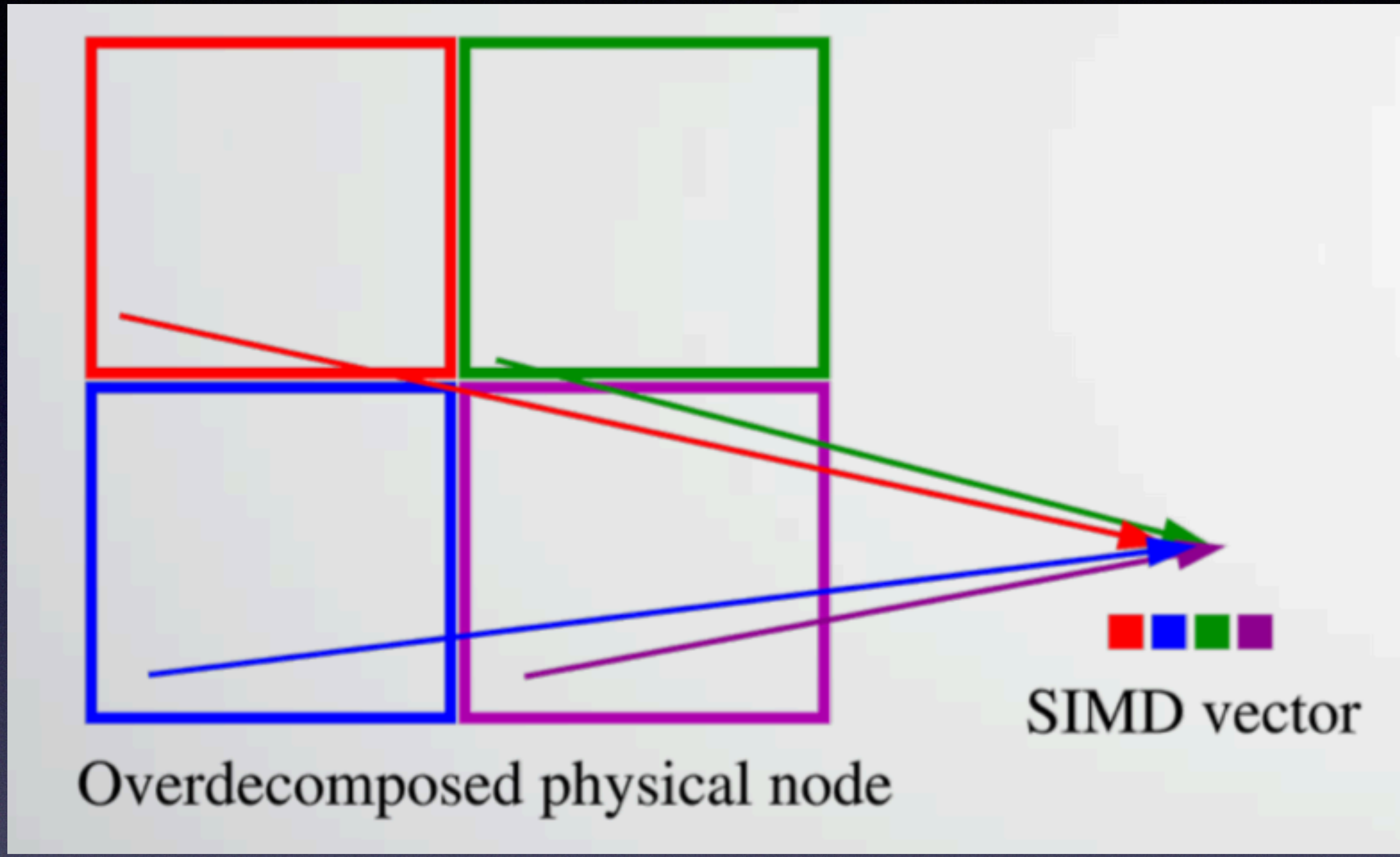
U.S. DEPARTMENT OF ENERGY | Office of Science

EXASCALE COMPUTING PROJECT

- USQCD Exascale Computing Project (ECP) is developing a unified framework for QCD software: Goal is 50x or better improvement.
- Project tasks: Critical Slowing in HMC (Christ), Software API (DeTar), Tensor Contractions (Edwards) and Solvers (Brower)
- Nuclear Physics focus: Chroma (lead developer Edwards and Joo) GPU implementation with JIT/QUDA
- HEP Physics focus: GRID is new C++ 14 (lead developer Peter Boyle) with direct interface to GPUs

Grid: A next generation data parallel C++ QCD library





Split Grid Strategy in GRID software: https://github.com/paboyle/Grid

Advance is C++ on GPUs is giving increasing portability of Grid software on Intel and GPU architectures extending GPU to the entire software stack.
Two way exchange between QUDA and C++ advances

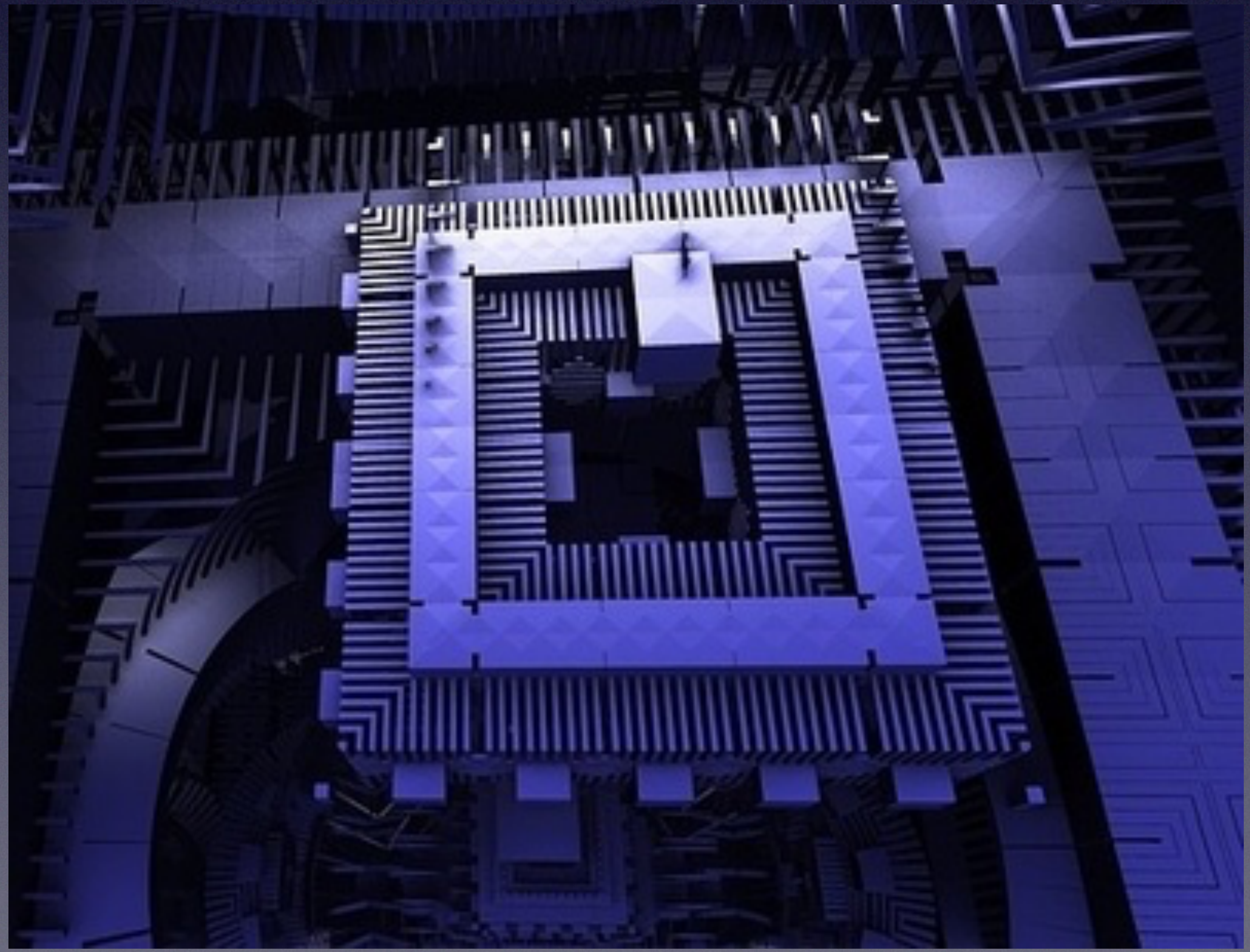
Top 10of TOP 500 (June 2018) https://www.top500.org/lists/top500/

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory	2,282,544	122,300.0	187,659.3	8,806
2	**Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China**	10,649,600	93,014.6	125,435.9	15,371
3	**Sierra - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/NNSA/LLNL United States**	1,572,480	71,610.0	119,193.6	
4	**Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China**	4,981,760	61,444.5	100,678.7	18,482
5	**AI Bridging Cloud Infrastructure (ABCI) - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan**	391,680	19,880.0	32,576.6	1,649
6	**Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland**	361,760	19,590.0	25,326.3	2,272
7	**Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States**	560,640	17,590.0	27,112.5	8,209
8	**Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States**	1,572,864	17,173.2	20,132.7	7,890
9	**Trinity - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/NNSA/LANL/SNL United States**	979,968	14,137.3	43,902.6	3,844
10	**Cori - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/SC/LBNL/NERSC United States**	622,336	14,014.7	27,880.7	3,939

Sign Problem: Quantum Leap to the Real Solution?

- Chemical Potential (Karsch) in Euclidean time hasa complex phase?
- Parton in real time (Xiang Dong Ji), dynamic, scattering , jets production?
- Quantum Path Integral is real time (Feynman). Use quantum computer?

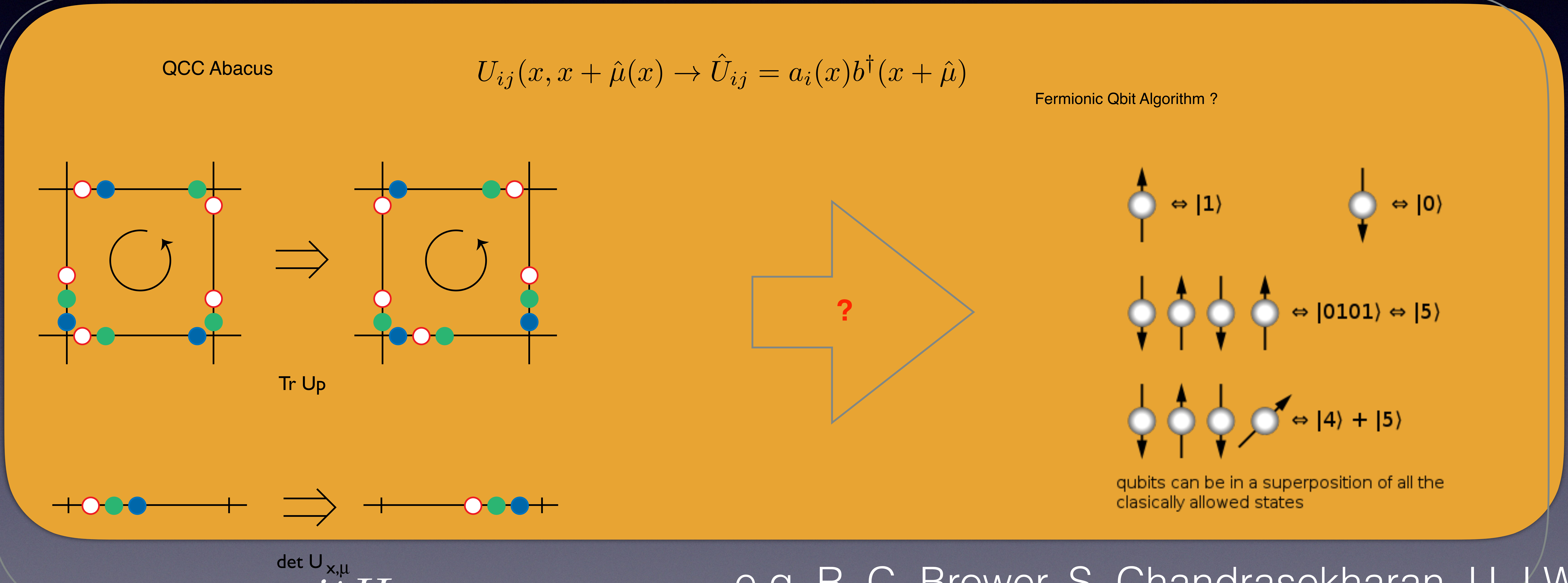
Path Integral Machine



US House Passes Bill to Invest More Than $1.2 Billion in Quantum Information Science

The Quantum accelerator?

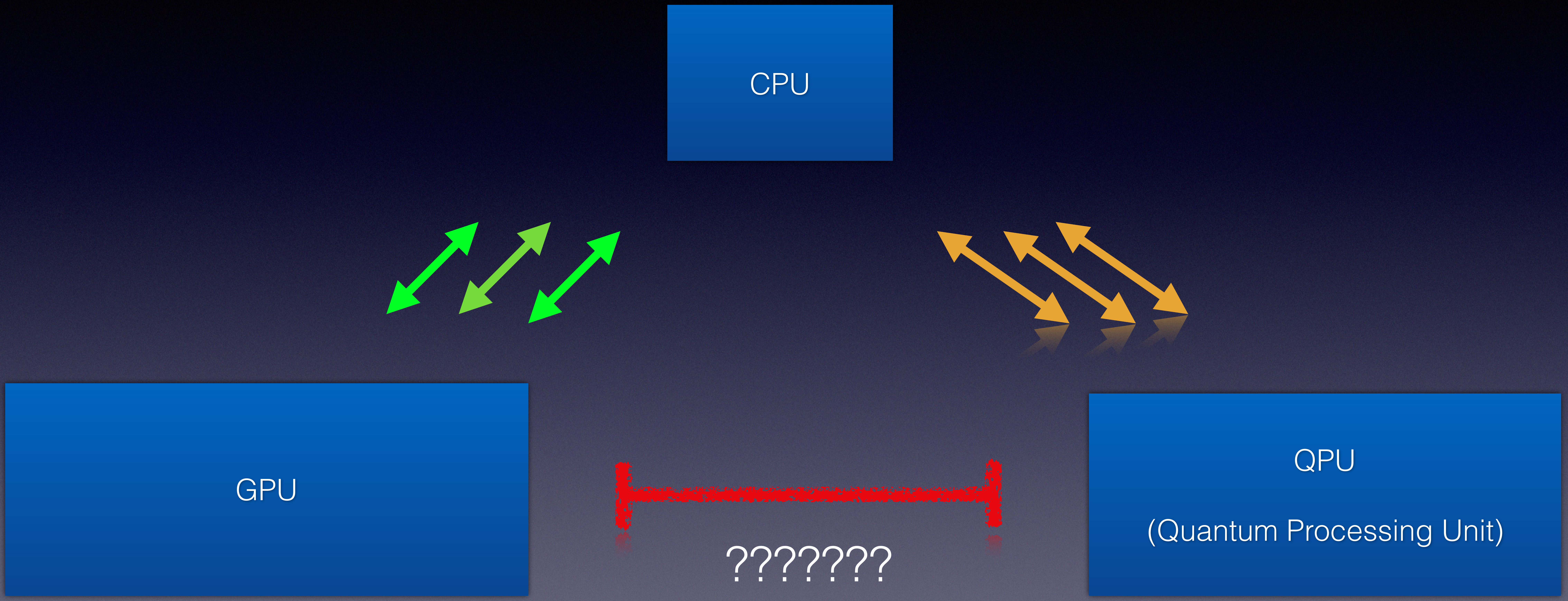
There is a totally Ferimonic (Qbit) QCD lattice Hamiltonian Appropriate for digital Quantum Computer!



$$|A(t), \psi(t)\rangle = e^{-itH}|A(0), \psi(0)\rangle$$

e.g. R. C. Brower, S. Chandrasekharan, U-J Wise , QCD as quantum link model, Phys. Rev D 60 (1999)

IN PRINCIPLE EXPONENTIAL BETTER SOL'N TO SIGN PROBLEM



Heterogenous Classical/Quantum Computer?

DON'T HOLD YOUR BREATH BUT THERE ARE LOT'S OF IDEAS AND $'S

Back to the Future: Exascale Challenges

- **Critical slowing down**
    - **With decreasing lattice spacing the cost of generating decorrelated gauge configuration grows rapidly**
- **Communication avoidance**
    - Internode communication may substantially reduce performance on exascale architectures. (Summit could double the network with $'s)
- **Complexity of nuclear matrix elements**
  - The problem complexity grows very rapidly with increasing atomic number.
- Portability to diverse architecture.
  - Exascale points to increasely heterogeneity & complex memory hierarchy. Software tools in C++, MPI, openACC etc should must assist portable code design

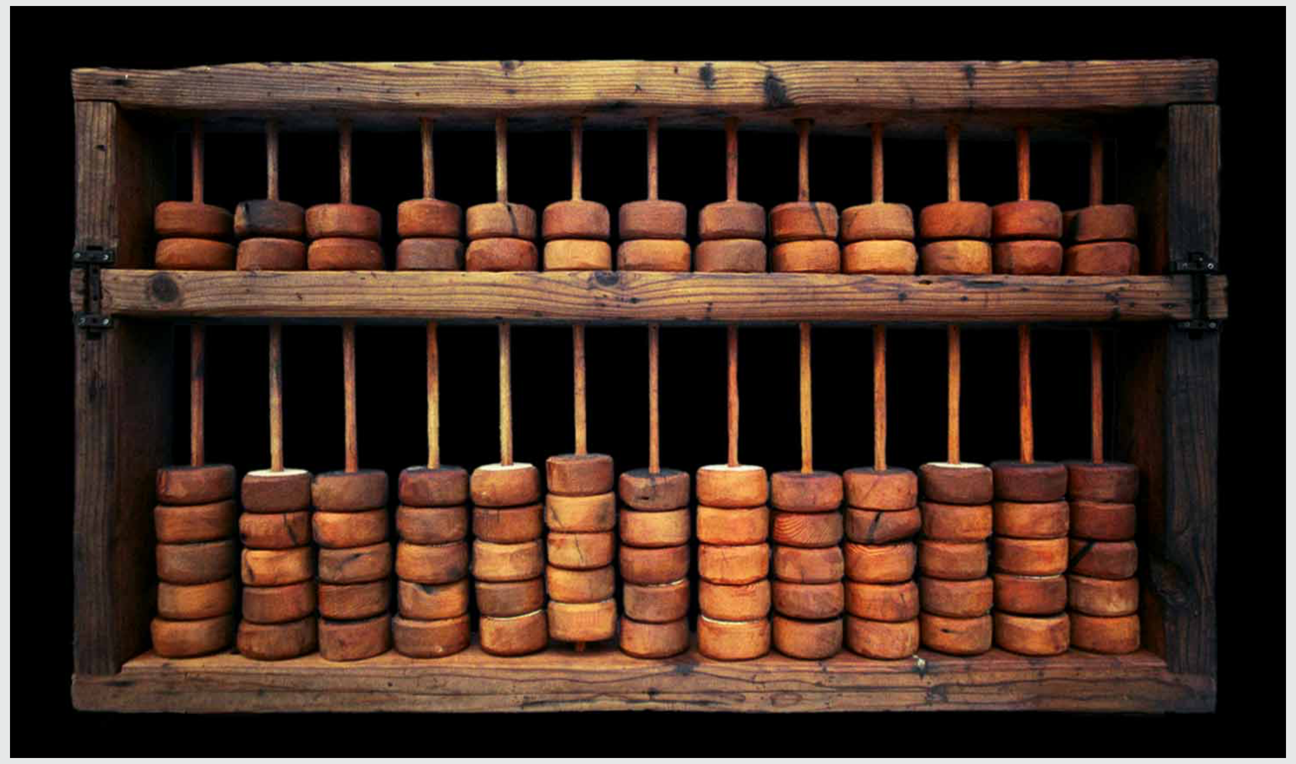
QUESTIONS ?

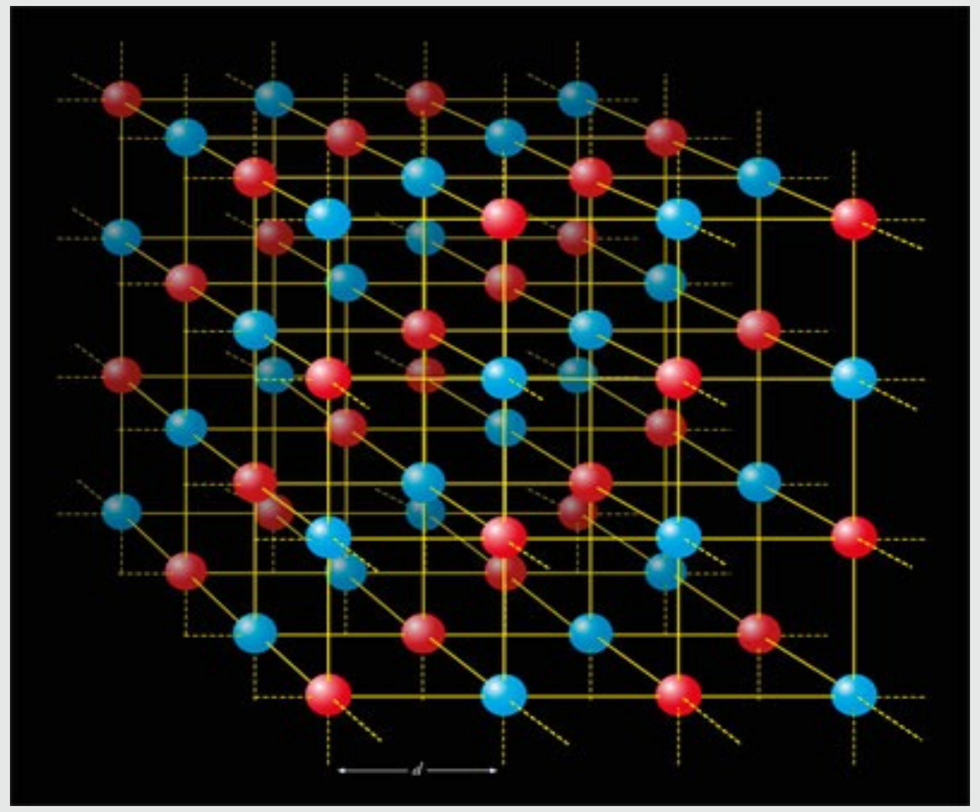
BACK UP SLIDES

THE QCD ABACUS

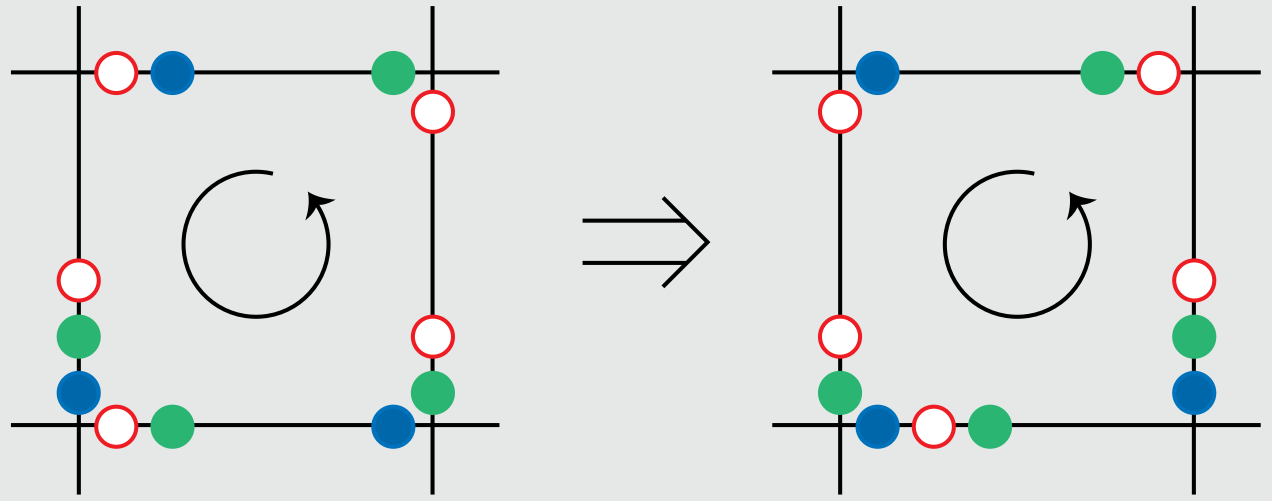
$$\hat{H} = \beta \sum_{x,\mu\neq\nu} \mathrm{Tr}[\hat{U}_{x,\mu}\hat{U}_{x+\hat{\mu},\nu}\hat{U}^{\dagger}_{x+\hat{\nu},\mu}\hat{U}^{\dagger}_{x,\nu}] + \sum_{x,\mu} [\mathrm{det}\hat{U}_{x,\mu} + \mathrm{det}\hat{U}^{\dagger}_{x,\mu}]$$

circa 2400 b.c Abacus

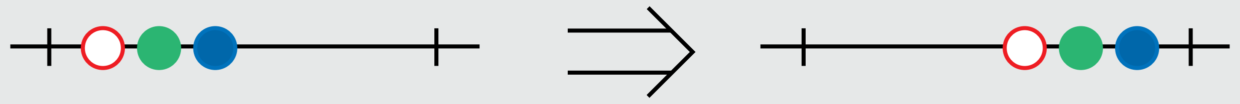




circa 20xx a.d.



Tr Up



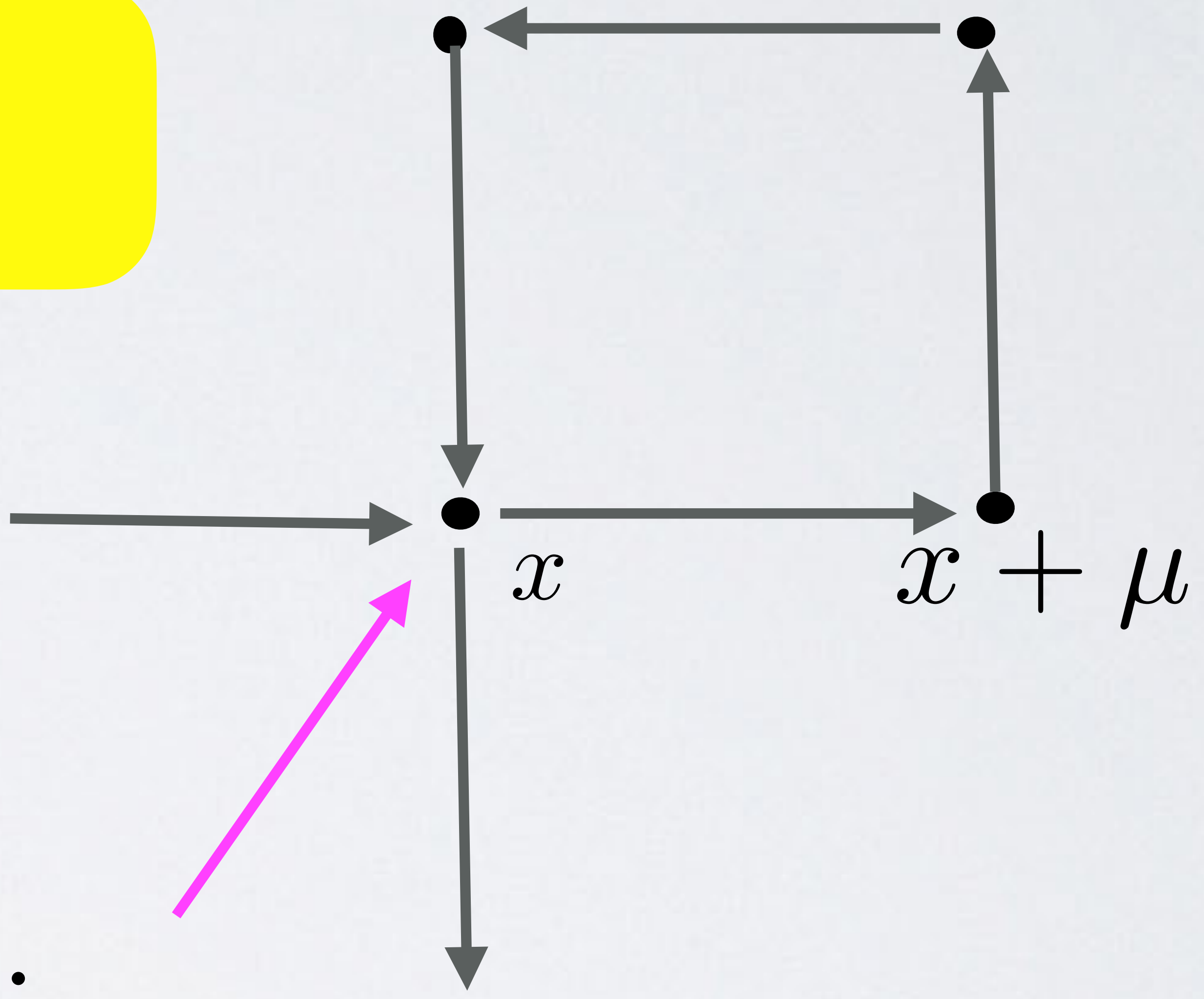
det $U_{x,\mu}$

OPERATOR QUANTUM GAUGE LINK

On each link (x,x + μ) introduce $2N_c$complex fermion

$a_{,}a_i^\dagger$ right(+) moving and $b_j, b_j^\dagger$ left(-) moving fluxon

LINK: $a_i \longrightarrow b_j^\dagger$



$$U_{ij}(x, x+\mu) \quad \to \quad \hat{U}_{ij} = a_i(x)\; b_j^\dagger(x+\mu)$$

$$\{a_i, a_j^\dagger\} = \delta_{ij} \qquad \{b_i, b_j^\dagger\} = \delta_{ij}$$

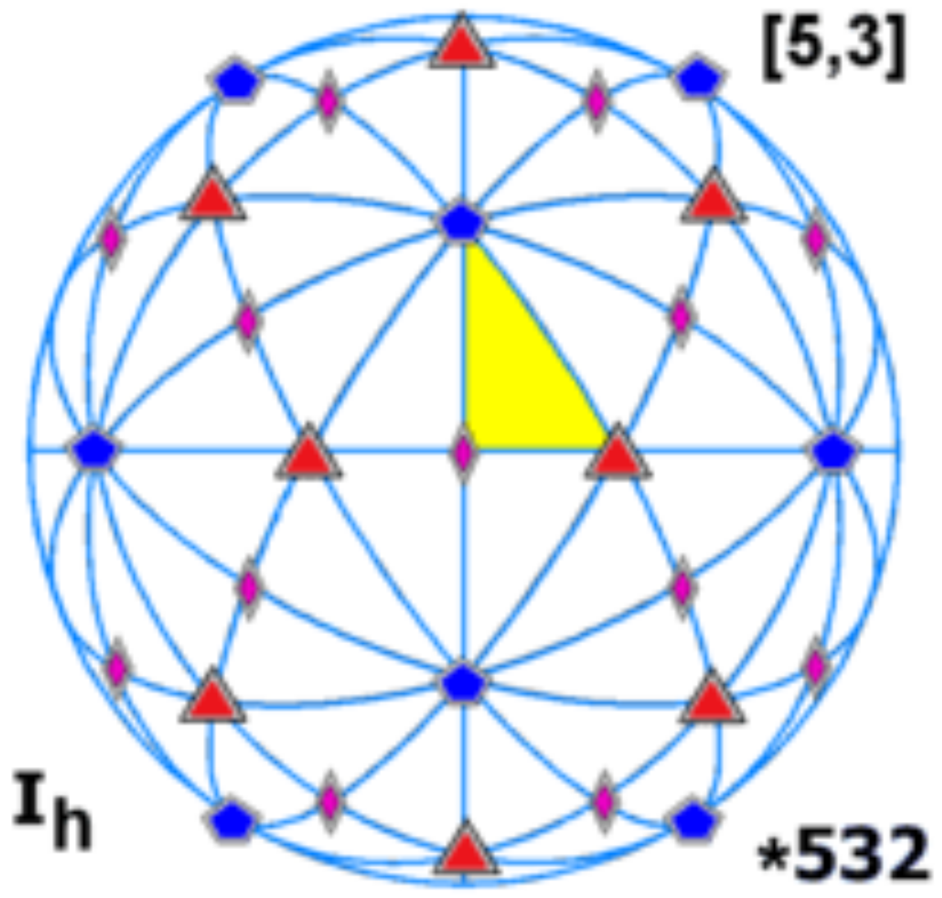
Local Gaue Operators $\quad \Omega_{ij}(x) = a_i^\dagger(x) a_j(x) + \cdots$

For SU(3) QCD have SU(6) per Link Lie Algebra $\begin{bmatrix} a_i a_j^\dagger & a_i b_j^\dagger \\ b_i a_j^\dagger & b_i b_j^\dagger \end{bmatrix}$

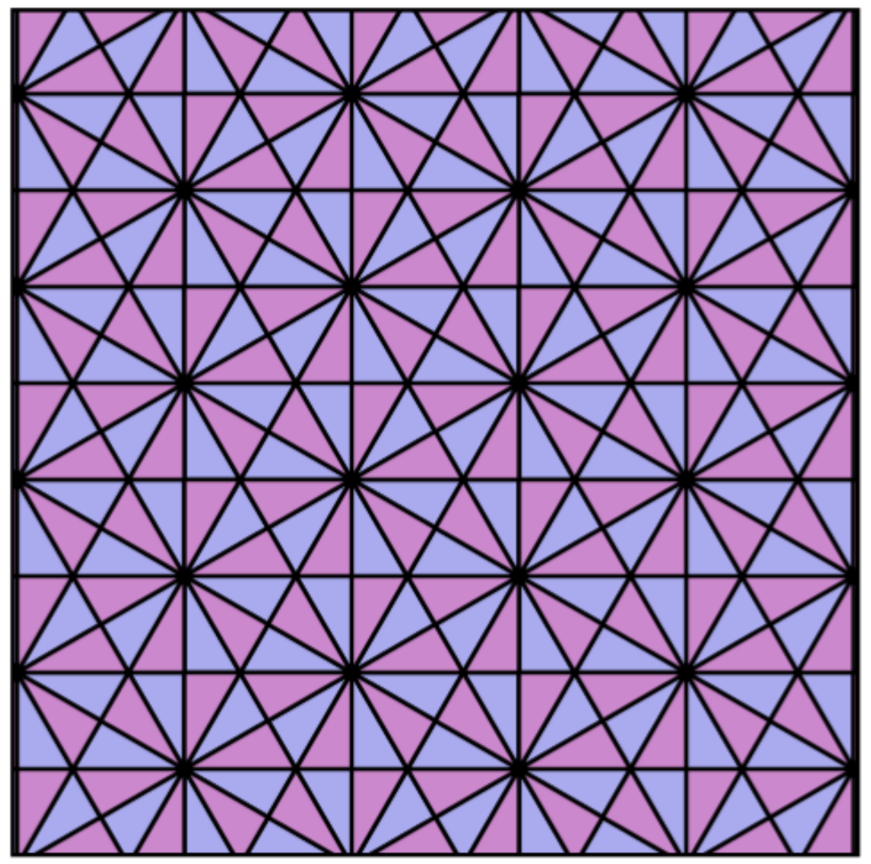
Hilbert Space is a large qubit array of color vectors in 4d space-time

DISCRETE ISOMETRIES & THE TRIANGLE GROUP

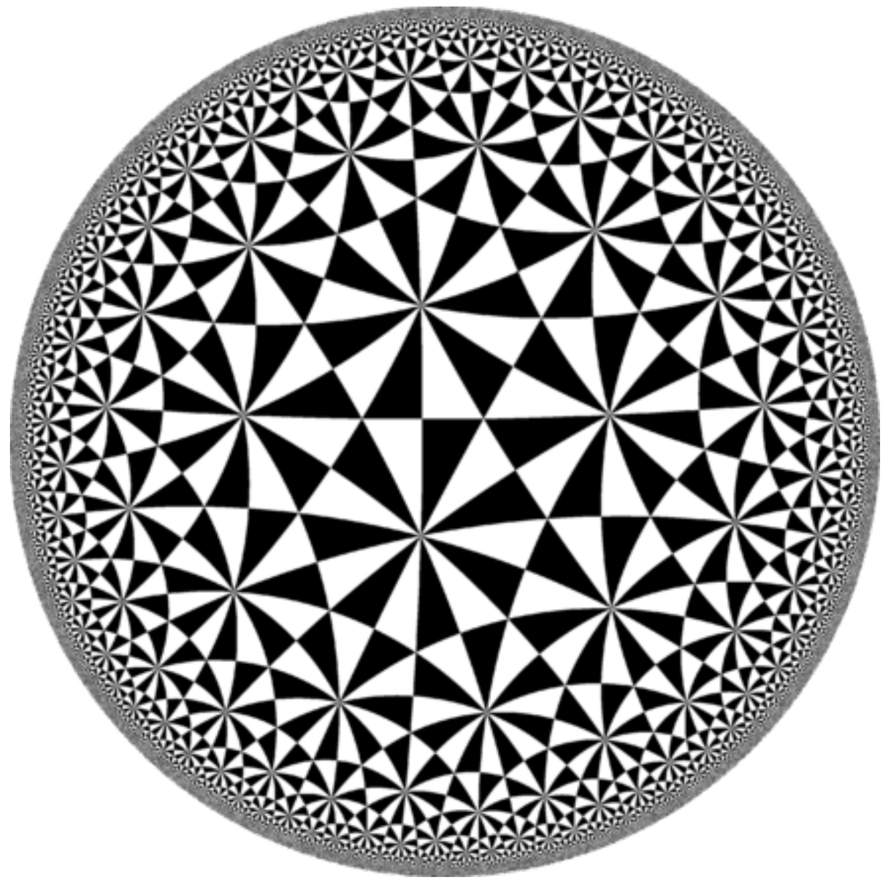
$$\frac{\pi}{p}+\frac{\pi}{q}+\frac{\pi}{r}\quad\begin{cases} >\pi & \text{Postive curvature}\\ =\pi & \text{Zero curvature}\\ <\pi & \text{Negative Curvature}\end{cases}$$



$(2,3,5)$
120 element
Icosahedral in $O(3)$



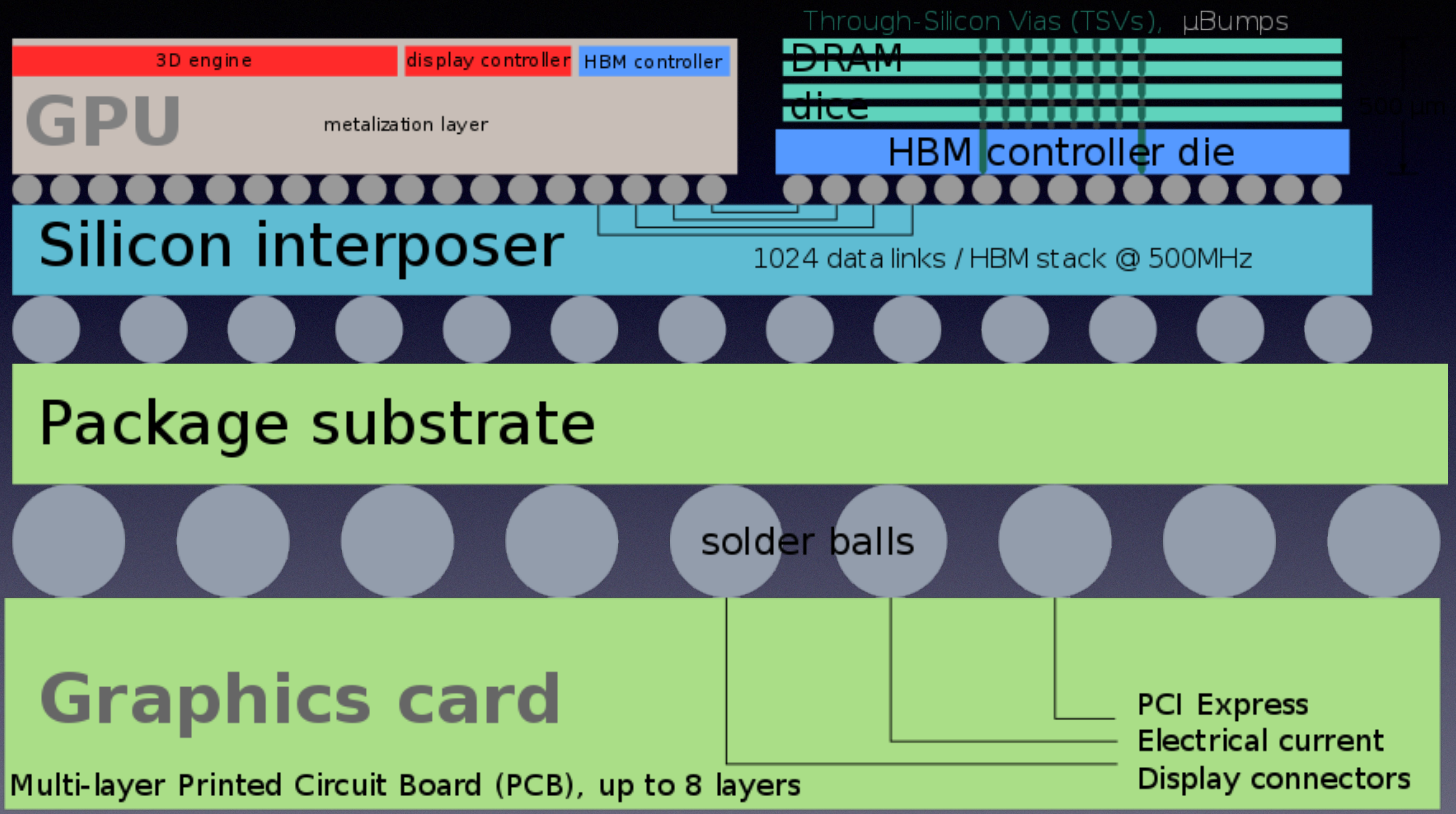
$(2,3,6)$
Triangle Lattice
on Euclidean $\mathbb{R}^2$



$(2,3,7)$
Subgroup of Modular
Group on $\mathbb{H}^2$

https://en.wikipedia.org/wiki/(2,3,7)_triangle_group

High Bandwidth Memory (HBM)



https://en.wikipedia.org/wiki/High_Bandwidth_Memory