lighting talk:
Lattice QCD in the exascale computing era

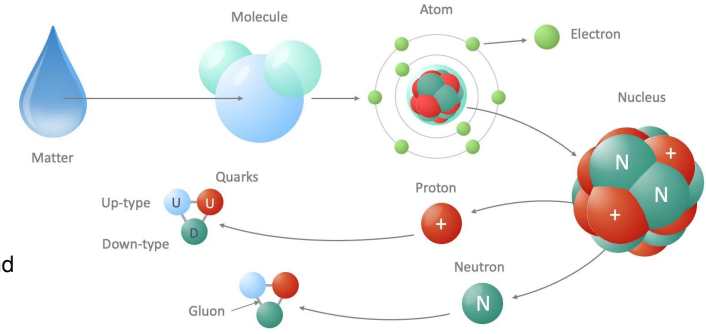
Marco Cè

ICSC and Spoke 2 - Where Are We Now?
Catania, 11 dicembre 2024

Quantum Chromodynamics

is the theory of strong interactions

- part of the Standard Model of particle physics
- explains how protons and neutrons are made from their constituents
  how they get their mass
- and how protons and neutrons are bound in atomic nuclei



femtoscale: typical length scale ≈ 1 fm $= 10^{-15}$ m (hydrogen atom is 10^{-10} m)

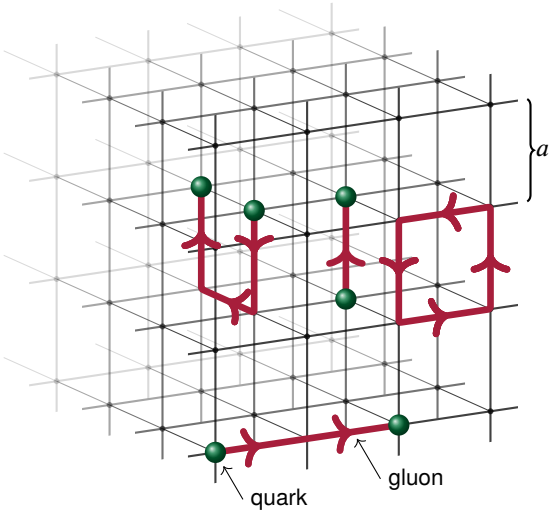
$\Rightarrow$ typical energy scale ≈ 200 MeV (or 20 million times the energy to ionize an atom)

QCD is a quantum field theory

- non-abelian gauge theory of quarks and gluons [Fritzsch, Gell-Mann 1973]
- renormalizable, asymptotically free [Gross, Wilczek 1973, Politzer 1973]

$\Rightarrow$ at low energies non-perturbative approach is needed

lattice QCD



a discretization of QCD on a four-dimentional lattice

[Wilson 1974]

- Euclidean space-time is discretized
- a is the lattice spacing
  usually, $a \approx 0.02\,\text{fm}$–$0.2\,\text{fm}$
- gluons live on links between sites
- quarks live on sites

$\Rightarrow$ enables numerical solution of QCD on supercomputers

- *ab initio* — only experimental input to set the scale
- continuum limit is recovered for $a \to 0$
  (implies $L \to \infty$)

lattice QCD fundamentals

the QCD path integral regularized on the lattice can be rigorously defined

$$\langle P(x)P(0)\rangle = \int \mathrm{d}U_\mu(x)\, \left|D[U]^{-1}(x,0)\right|^2 \exp\left\{-S_g[U]\right\} \det D[U]^2$$

- 32 integrals per lattice site ⇒ millions of integrals!!
- has the form of a high-dimensional probability distribution

curse of dimensionality ⇒ Monte Carlo integration with importance sampling

typical Monte Carlo workflow

1. generation: sample $\{U\}$ according to $p[U] \propto \det\{D[U]\} \propto \exp\{-\left|D[U]^{-1}\phi\right|^2\} > 0$
2. measurement: compute $\hat{O} = (1/n)\sum_i O[U_i]$ with $O[U] = \left|D[U]^{-1}(x,0)\right|^2$

the Dirac operator $D[U]$ is a sparse matrix of size $[3\times 4\times L^4]^2 \Rightarrow [10^7\text{–}10^9]^2$

$$D[U]\cdot\psi = \eta$$

solving the Dirac equation is the most computationally expensive part, both in the generation and measurement

- benchmark for Lattice QCD algorithms and software
- we use iterative methods for sparse linear systems: Krylov-subspace solvers, *e.g.* CG, BiCGSTAB, …
- but the best algorithms are inspired by the physics behind $D[U]$, *e.g.* deflated GCR, multigrid solvers, …

lattice QCD fundamentals

the QCD path integral regularized on the lattice can be rigorously defined

$$\langle O \rangle = \int \mathrm{d}U \, O[U] p[U]$$

- 32 integrals per lattice site ⇒ millions of integrals!!
- has the form of a high-dimensional probability distribution

curse of dimensionality ⇒ Monte Carlo integration with importance sampling

typical Monte Carlo workflow

1. generation: sample $\{U\}$ according to $p[U] \propto \det\{D[U]\} \propto \exp\{-\left|D[U]^{-1}\phi\right|^2\} > 0$
2. measurement: compute $\hat{O} = (1/n)\sum_i O[U_i]$ with $O[U] = \left|D[U]^{-1}(x,0)\right|^2$

the Dirac operator $D[U]$ is a sparse matrix of size $[3 \times 4 \times L^4]^2 \Rightarrow [10^7\text{–}10^9]^2$

$$D[U] \cdot \psi = \eta$$

solving the Dirac equation is the most computationally expensive part, both in the generation and measurement

- benchmark for Lattice QCD algorithms and software
- we use iterative methods for sparse linear systems: Krylov-subspace solvers, *e.g.* CG, BiCGSTAB, …
- but the best algorithms are inspired by the physics behind $D[U]$, *e.g.* deflated GCR, multigrid solvers, …

lattice QCD fundamentals

the QCD path integral regularized on the lattice can be rigorously defined

$$\langle O \rangle = \int \mathrm{d}U \, O[U] p[U]$$

- 32 integrals per lattice site ⇒ millions of integrals!!
- has the form of a high-dimensional probability distribution

curse of dimensionality ⇒ Monte Carlo integration with importance sampling

typical Monte Carlo workflow

1. generation: sample $\{U\}$ according to $p[U] \propto \det\{D[U]\} \propto \exp\{-\left|D[U]^{-1}\phi\right|^2\} > 0$
2. measurement: compute $\hat{O} = (1/n)\sum_i O[U_i]$ with $O[U] = \left|D[U]^{-1}(x,0)\right|^2$

the Dirac operator $D[U]$ is a sparse matrix of size $[3 \times 4 \times L^4]^2 \Rightarrow [10^7\text{–}10^9]^2$

$$D[U] \cdot \psi = \eta$$

solving the Dirac equation is the most computationally expensive part, both in the generation and measurement

- benchmark for Lattice QCD algorithms and software
- we use iterative methods for sparse linear systems: Krylov-subspace solvers, *e.g.* CG, BiCGSTAB, …
- but the best algorithms are inspired by the physics behind $D[U]$, *e.g.* deflated GCR, multigrid solvers, …

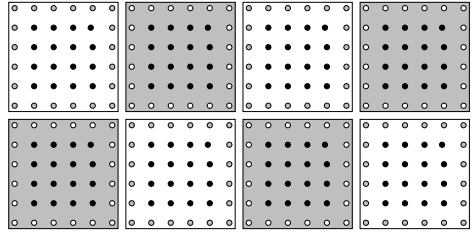
the software that we use

mutual feedback loop: physics ⇔ algorithms ⇔ software

⇒ no commercial software packages, we write our own code

openQCD [Lüscher, Schaefer 2013, https://luscher.web.cern.ch/luscher/openQCD]

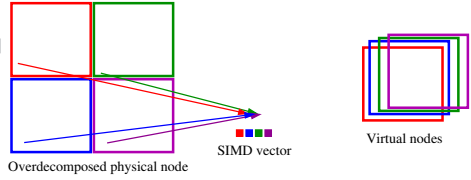
- written in C89, open source, GPL license
- lattice is decomposed in 4d grid of *local lattices*, one process per CPU core, halo communication using MPI
- since version 2.4 (1st May 2022), also OpenMP parallelization
- handwritten AVX+FMA3 inline-assembly optimizations
- implements HMC and SMD algorithms, deflated SAP-accelerated GCR solver, …



Grid: Data parallel C++ mathematical object library

[Boyle *et al.* 2015, https://github.com/paboyle/Grid]

- portable (runs on GPUs), open source, GPL license
- automatic SIMD vectorization
- hybrid OpenMP and MPI parallelization

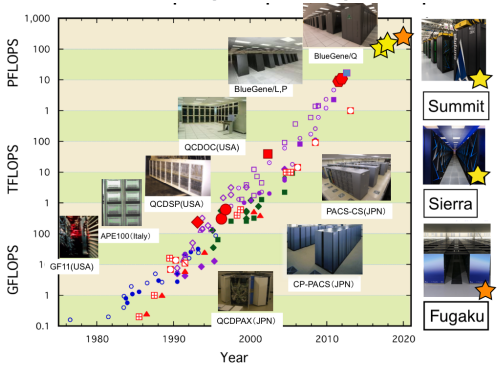


lattice QCD at the HPC frontier

TOP500 - November 2024

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
	[…]				
8	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107
9	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA Italy	1,824,768	241.20	306.31	7,494
10	**Tuolumne** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	1,161,216	208.10	288.88	3,387
11	**MareNostrum 5 ACC** - BullSequana XH3000, Xeon Platinum 8460Y+ 32C 2.3GHz, NVIDIA H100 64GB, Infiniband NDR, EVIDEN EuroHPC/BSC Spain	663,040	175.30	249.44	4,159

[Akira Ukawa 2014]

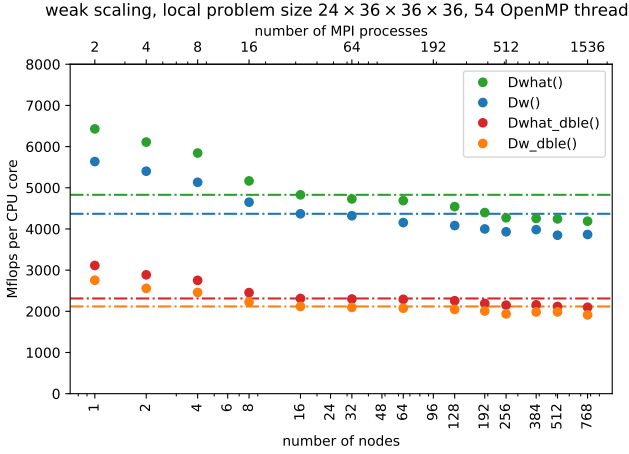


HPC resources used within Lattice QCD group @ U. Milano-Bicocca

- EuroHPC Extreme Scale Access on LUMI
  ⇒ No. 8 in Top 500 - Nov. 2024
- INFN-Cineca agreement and ICSC RAC on Leonardo
  ⇒ No. 9 in Top 500 - Nov. 2024
- EuroHPC Extreme Scale Access on MareNostrum 5
  ⇒ No. 11 in Top 500 - Nov. 2024
- many smaller others used for r&d

weak scaling example I

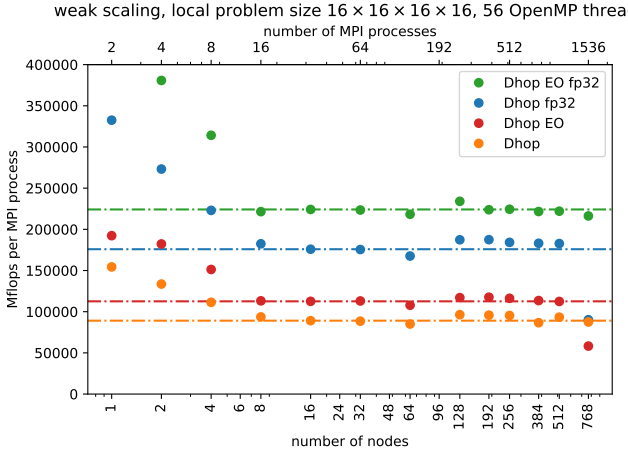
of openQCD's `Dw` and `Dw_dble`
on MareNostrum5 GPP (each node: $2\times$ Intel Xeon Platinum 8480+ 56C@2 GHz + 100 Gbit/s network)



- 4d communication not needed below 16 nodes
- near-perfect efficiency ≥ 16 nodes

weak scaling example II

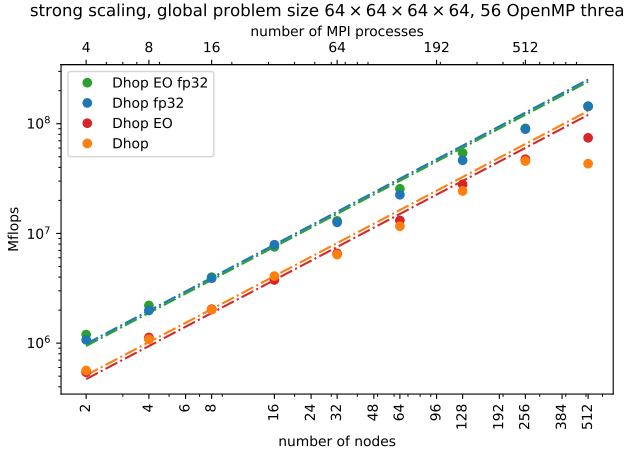
of Grid's `DomainWallFermion::Dhop`
on MareNostrum5 GPP



- 4d communication not needed below 16 nodes
- near-perfect efficiency $\geq$ 16 nodes

strong scaling example

of Grid's `DomainWallFermion::Dhop`
on MareNostrum5 GPP



- near-perfect speedup from parallelizing the same problem over 500 nodes / 50 thousand cores

performance and bottlenecks

a Dirac operator application has low computational intensity

- favour architectures with higher memory bandwidth per core
- using multiple RHS improves things

ensemble generation with Markov-chain Monte Carlo is intrinsically serial
$\Rightarrow$ strong scaling to many nodes (thousands of cores) is crucial

- network communication must be fast and low latency, *e.g.* InfiniBand
- each rank has 8 neighbours in a $4d$ torus, network topology also plays a role

measurement runs may be trivially parallelizable

- less scaling, but still tens of nodes/hundreds of cores
- some workflows require a lot of RAM

conclusions and outlook

QCD can be solved on the lattice through highly-parallelized HPC simulations

- crucial role of the algorithmic developments inspired by physics
  ⇒ enables huge lattices, *e.g.* master-field approach
- future architectures will have more (lower-precision) flops per memory and network bandwidth
  ⇒ multiple RHS, matrix operations required to saturate the machine
- hierarchy and locality must be considered ⇒ domain decomposition, multi-level integration stategies

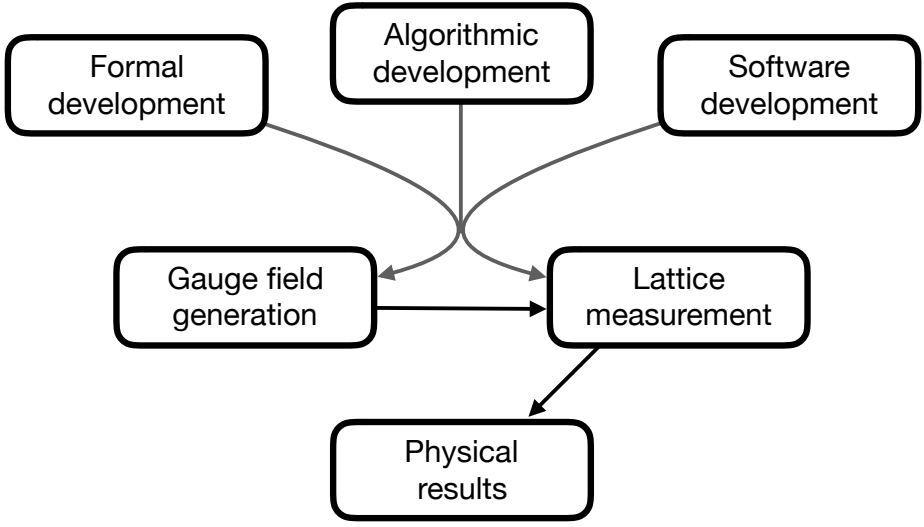
HPC enables progress towards answering open questions in particle physics

- sub-percent determination of the hadronic contribution to the muon anomalous magnetic moment $(g-2)$
- study of the nuclear matter under extreme conditions, *e.g.* Ts up to $O(100\,\text{GeV})$
- determination of the strength of the strong interactions ⇒ coupling α_s
- … and much more

thanks for your attention!

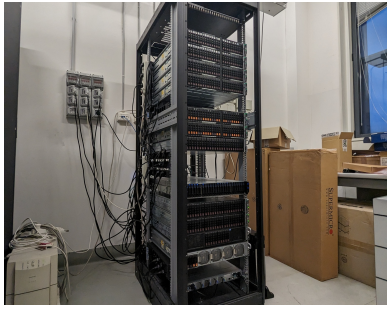
backup slides

lattice QCD workflow



three levels of HPC usage

small



[knuth @ dip. Fisica, U. Bicocca]

dedicated small clusters,

- algorithms r&d
- new software testing

medium

[MOGON II @ JGU Mainz, July '17]

shared access to
university / tier-1 clusters,

- scaling testing
- r&d at scale
- project production

large

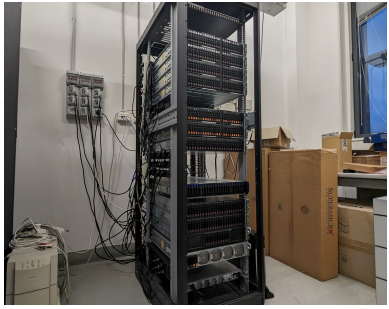
[Leonardo @ Cineca]

tier-0 supercomputer access
through large-scale applications,

- large project production

three levels of HPC usage

small



[knuth @ dip. Fisica, U. Bicocca]

dedicated small clusters,

e.g. knuth:
36 dual 16-core Epyc 7302

\+ similar for GPU clusters

medium

[MOGON II @ JGU Mainz, July '17]

shared access to
university / tier-1 clusters,

e.g. MOGON II @ JGU Mainz:
822 dual 10-core Xeon E5-2630v4
1136 dual 16-core Xeon Gold 6130
top500 66th on 11/2017

large

[Leonardo @ Cineca]

tier-0 supercomputer access
through large-scale applications,

e.g. through EuroHPC:

- 120 Mcore-h on LUMI-C @ CSC
- 50 Mcore-h on Leonardo @ Cineca

typical Monte Carlo work flow

1. gauge field configurations generation

using Markov-chain Monte Carlo methods, *e.g.* Metropolis-Hastings algorithm

- local update methods are fast without quarks, but $p[U] \propto \det\{D[U]\}$ is non-local
- state-of-the-art hybrid Monte Carlo (HMC): Molecular Dynamics update proposal + Metropolis accept-reject step

$\Rightarrow$ intrinsically serial task $\Rightarrow$ strong scaling is crucial!

2. lattice measurements

key building block: correlation functions

- some trivial parallelization is possible, but large problems still require large partitions
  e.g. correlator measurements on 192^4 lattice on 256 nodes of dual EPYC 7763 @ LUMI-C
- observable measurements is usually (but not always!) less expensive
  $\Rightarrow$ configurations are saved and libraries are shared in large collaborations
- signal-to-noise of correlators is often poor
  $\Rightarrow$ standard Monte Carlo scales only $\propto 1/\sqrt{n}$

in both cases, the Dirac operator $D[U]$ plays a crucial role

lattice QCD on GPUs

lattice theory at the early GPU frontier: pioneering work in 2007

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

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.

QUDA with a Q

[https://lattice.github.io/quda/]

a library for QCD on GPUs [Clark *et al.* 2010]

- open source, BSD license
- based on CUDA, developed in collaboration with Nvidia
- heavily optimized to hide communication behind computation
- ~~Nvidia-specific~~
  not anymore: HIP (merged), SYCL (in review) and OpenMP (in development) support [Clark Lattice 2023]

but

- algorithm development on GPUs is hard!
- not all workflows are easily portable

also: Grid now supports GPUs

- SIMD lanes on CPUs map to GPU threads
- CUDA, HIP, SYCL, OpenMP offloading

storage

state-of-the-art gauge field ensemble: a few hundred 192×96^3 gauge field configurations

- each configuration is $18\,\texttt{float64} \times 4 \times L_t \times L_s^3 = 91$ GiB
- MC produces $\lesssim 10$ every day
- (part of) checkpoints, but also input of measurement calculations
- configurations are expensive and saved for reuse

$\Rightarrow$ sizeable storage needs included in HPC resources applications

- measurement runs are more scalable, but configurations need to be moved
- Dirac equation solves for measurements are also expensive: alternative workflows save the solutions to disk

2001: the Berlin wall

cost of simulating QCD with Wilson fermions
estimated at the Lattice 2001 conference in Berlin [Ukawa, Nucl. Phys. B Proc. Suppl. 2002]

$$\text{cost} \approx 2.8 \left[\frac{\#\text{conf.}}{1000}\right] \left[\frac{M_\pi/M_\rho}{0.6}\right]^{-6} \left[\frac{L}{3\,\text{fm}}\right]^{5} \left[\frac{a^{-1}}{2\,\text{GeV}}\right]^{7} \text{Tflops} \cdot \text{year}$$

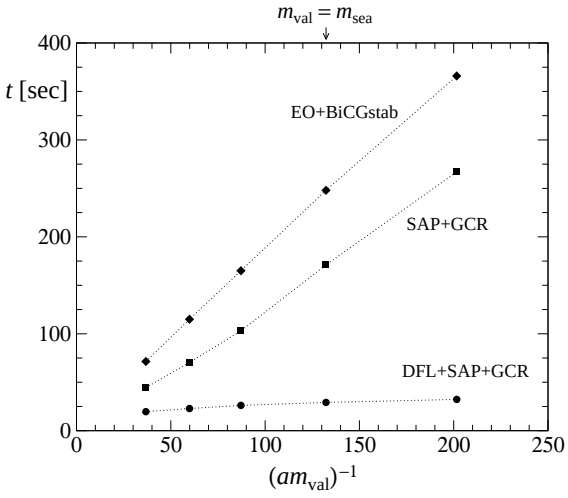
with year 2000 algorithms, 100 conf. of 192×96^3, $a = 0.064$ fm would cost 280 Pflops · year
$\Rightarrow \approx 640$ billion core hours!!

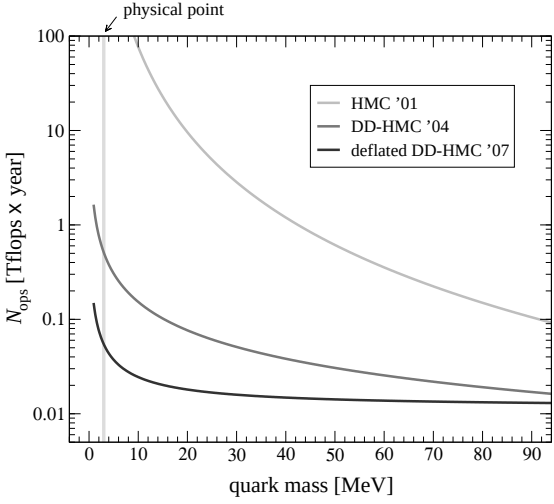
today, this calculation can be done in about 20 million core hours, 32 thousand times faster!!

tearing down the wall

- Sexton–Weingarten multiple time-step integration [1992]
- Hasenbusch factorization [2001]
  and mass preconditioning [Urbach *et al.* 2006]
- domain decomposition [Lüscher 2004]
  and low-modes deflation [Lüscher 2007]
- the rational HMC algorithm [Clark, Kennedy 2007]
- multigrid solvers [Brannick *et al.* 2008]
- open boundary conditions against topology freezing [Lüscher, Schaefer 2011]

tearing down the wall





[Lüscher, Les Houches 2009]