The EXALAT Project

Lattice Field Theory towards the Exascale

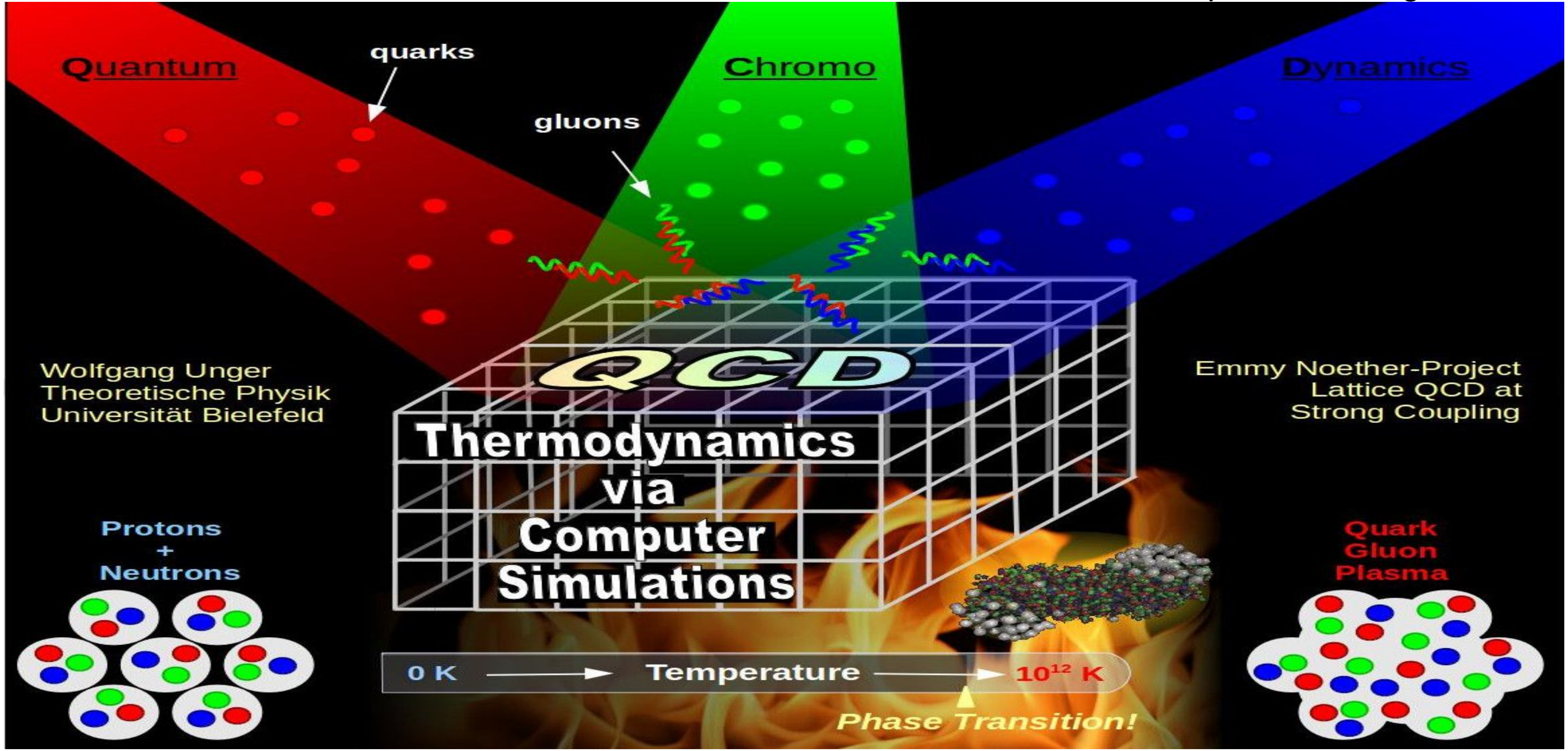
Content

- Physics challenges
- Computational challenges
- An overview of Lattice QCD Code
- EXALAT benchmarks

Context

The computational problem

Graphics: W. Unger



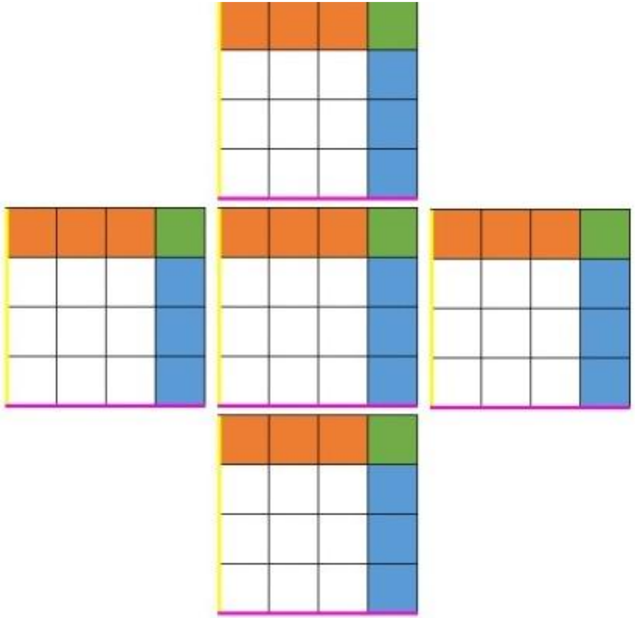
Algorithms

Algorithms

- Work flow can be broken in two main (distinct) components: generation of gauge configurations via MCMC and measurements of observables
- Generally the computational cost generation:measurements is between 80:20 and 90:10
- For the generation, the bulk of the time is spent on the inversion of a large sparse matrix
- Details are different, as there are a few community codes taking a different angle in order to optimally target different Physics applications
- Broadly speaking, the main algorithm is conjugated gradient (and improvements), which is accelerated with preconditioners

Computational challenges

- $O(10^9)$ SU(3)/SU(N)/Sp(2N) matrices at each step
- Need to invert sparse matrix of ~ this linear size
- Hierarchic programming: low-level vs accelerator vs shmem vs distmem
- Need to interlace communication and computation
- Equivalent different formulations of the physical problem implemented in community codes



Software suites

Grid

https://github.com/paboyle/Grid

- Purpose: Lattice QFT (QCD, QCD+QED, BSM, ...)
- Parallelism:
  - CPU: SIMD, Multi-thread, Multi-processing (MPI).
  - GPU: SIMT (Cuda, HIP, Sycl), Multi-processing (MPI).
  - Expresion template engine abstracts site wise operations (automatically parallel).
  - High level cshift and stencil interfaces.
- Multiplatform: vectorisation for many instruction sets. (SSE,AVX,AVX2,...)
- Implements popular lattice QCD fermion actions (Wilson, DWF, Staggered,...)
- Variety of solver algorithms already implemented (CG, Multi-grid, Lanczos,...)
- Full HMC/RHMC interface included.
- Workflow management: Hadrons [https://github.com/aportelli/Hadrons]

HiRep

https://github.com/claudiopica/HiRep

- Purpose of the code: explore novel strong interactions
- Main physics motivations: fundamental mechanism of electroweak symmetry breaking in the standard model and dark matter
- Lack of clear experimental guidance suggests to use safe methods
- Different theories are implemented at compilation time, through a PERL preprocessor
- Uses arrays of structures, enrolling and inlining of mathematical operations
- The code is high-level, developed in C, and uses MPI with latence masking
- First implementations of OpenMP and CUDA available, but needs significantly more work

openQCD
http://luscher.web.cern.ch/luscher/openQCD/

- Purpose: Lattice QFT (QCD, and derivations for QCD+QED)
- The code is highly optimized for machines with current x86-64 processors.
- The performance of the programs is mainly limited by data movement hence by the memory-to-processor bandwidth and the network latencies.
- The code is structured so as to ensure a very good data locality.
- The machine-specific optimizations include inline AVX/SSE3 assembly kernels

Algorithms:

- Nested hierarchical integrators for the molecular-dynamics equations, (leapfrog, 2nd and 4th order Omelyan-Mryglod-Folk (OMF) elementary integrators).
- Deflation acceleration and chronological solver along the molecular-dynamics trajectories.
- Solvers (CGNE, MSCG, SAP+GCR, deflated SAP+GCR) for the Dirac operator.

MILC
https://github.com/milc-qcd/milc_qcd

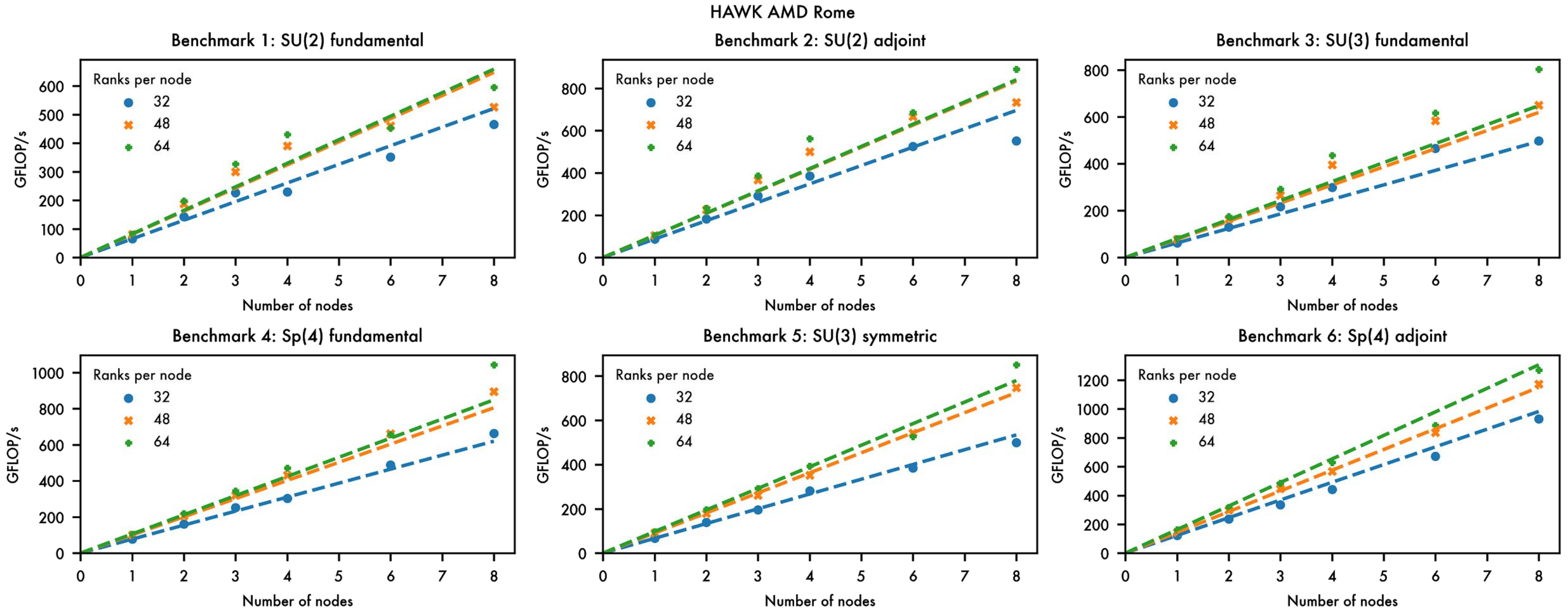
- Main code of the homonymous collaboration and of the HotQCD collaboration
- Main speciality: light quark masses with lesser computational cost (down to the fermion formulation, which however is not trouble-free)
- Main applications: spectrum of QCD and the deconfinement phase transition
- In the UK, mostly used for measurements at zero temperature
- Written in pure C, use libraries common to other US project
- Parallelism through MPI and OpenMP
- CUDA available

Benchmark status

SOMBRERO

- Its previous incarnation, BSMBench, has been widely used by Intel (for OPA development), Mellanox (now NVIDIA Networking) and the HPC-AI Advisory Council
- Mini-app based on stripped-down version of HiRep
  - Initially a manual process
  - Now automated
  - 13,000 lines of code
- Tests speed of conjugate gradient Dirac operator inversion
  - Includes FLOP counts
- Benchmarks for six theories
  - QCD is one
  - Different FLOPs/byte from memory
  - Different FLOPs/byte across interconnect

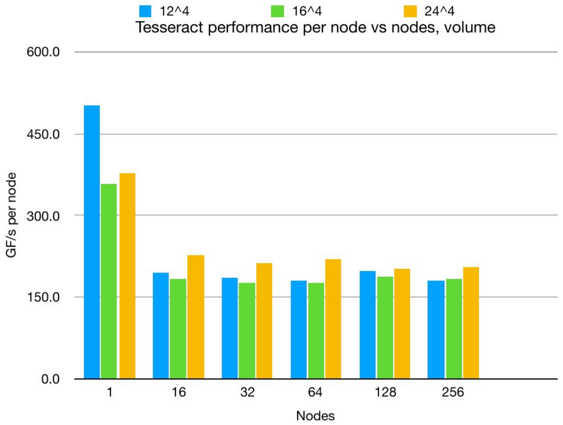
SOMBRERO results



Grid benchmarks

- Grid benchmarks have been used for Dirac 2.5 and Dirac 3 extreme scaling procurement.
- Benchmark_ITT:
  - Scans a range of local volumes (8, 16, 24, 32)
  - Packages a number of benchmarks to give a view on the system as a whole.
  - Memory Bandwidth: a*x+y and matrix multiply
  - Communication: Bi-directional bandwidth
  - **D-slash** kernels (Wilson, DWF, Staggered)
- Benchmark_IO:
  - Standard Read/Write
  - Lime Read/Write
  - Robustness check

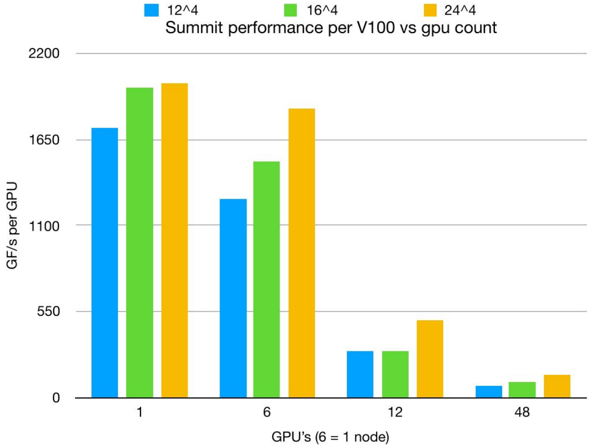
Grid benchmark results

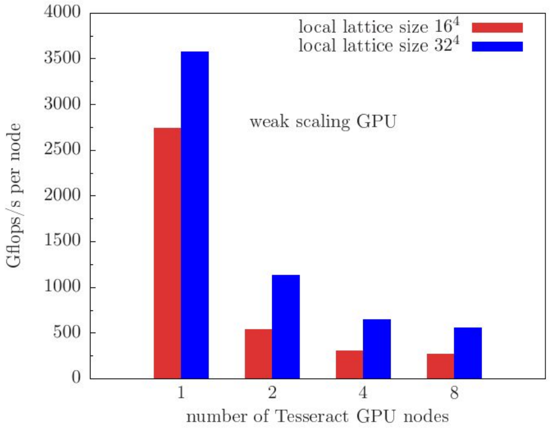


Tesseract CPU performance.

Grid single precision D-slash

Summit GPU performance.





Tesseract GPU performance.

Peter Boyle

USQCD All-Hands Collaboration Meeting 2019

Vera Guelpers

Conclusions

Summary

- Lattice QCD (or better, Lattice Field Theory) is a mature computational branch of theoretical particle physics
- Computational demands of the science questions keep being a challenge also at the exascale
- The flip side is that LFT can drive exascale development, both in hardware and software, that are transferable across disciplines
- A set of community codes are being ported to future architectures
- Benchmarks that are derived from those codes can provide measures of performance that usefully inform other applications (and the vendors!)