lattice QCD software development for heterogeneous supercomputers

Bartosz Kostrzewa
Marco Garofalo, Simone Romiti, Aniket Sen, Carsten Urbach (HISKP, University of Bonn)
Simone Bacchio, Ferenc Pittler (Cyprus Institute)
Jacob Finkenrath (University of Wuppertal)
Bálint Joó (Oak Ridge National Laboratory)
Dean Howarth (California Institute of Technology)
Kate Clark, Mathias Wagner, Evan Weinberg (NVIDIA)
Damian Alvarez, Ahmed Fahmy, Andreas Herten (Juelich Supercomputing Center)

deRSE24
Würzburg, Germany

Why this talk here?

- I could tell several different kinds of stories here:
  - Describe science and how this motivates investment in the underlying research software.
  - HPC aspects of lattice field theory research.
  - Talk about the structure of software in lattice field theory.
- I will, but focus on interactions between different people and stakeholders:
  - Researchers with physics goals.
  - Hardware vendors and their corporate strategy.
  - HPC centers providing a service to the scientific community.

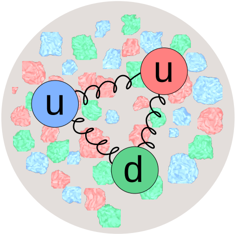
Thesis of this talk: ad-hoc interactions between the above groups have enabled much of our recent research.

- I spend a large part of my time in these interactions and they have become more important over the past decade.
- This looks very much like many open source projects.
- Unexpected and interesting things happen as a result.

What is Lattice QCD?

The Strong Nuclear Force

- **Question: Where does the mass of a proton, m_P, come from?**
  - Two *up* quarks and one *down* quark.
  - Quarks interact through *gluons*.
  - Gluons are massless, quarks are very light, $m_q \approx m_P/300$.



- When quarks are pulled apart, the force between them increases.
- **Answer: most of the mass of a proton comes from this binding energy.**

Quantum Chromodynamics (QCD)

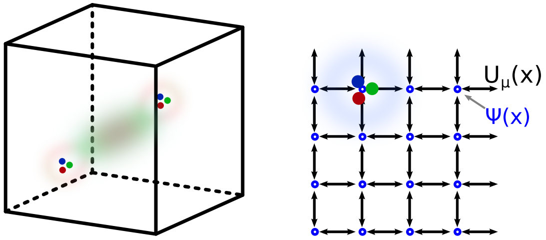
Strong force can be described by a quantum field theory called *Quantum Chromodynamics*.

$$\mathcal{L}_{\mathrm{QCD}} = \bar{\psi}_i (i\gamma^\mu (D_\mu)_{ij} - m\,\delta_{ij})\psi_j - \frac{1}{4} G^a_{\mu\nu} G^{\mu\nu}_a$$

At low energies **QCD cannot be solved analytically.**

Lattice QCD

- Solution:
  - Confine QCD to finite box.
  - Discretize theory onto a 4D space-time lattice.
  - Quark fields on vertices, gluon fields as "links".



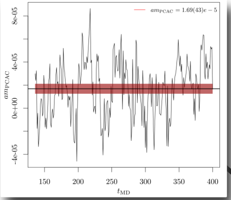
- Simulate stochastically like a statistical system with Boltzmann weight $e^{-\int \mathrm{d}x\, \mathcal{L}_{\mathrm{QCD}}}$.
  - Generate ensembles of gluon configurations $\{U\}$.
  - Observables as averages over these configurations.

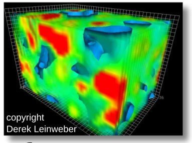
Turns out to be a numerical grand challenge!

Numerical Grand Challenge Problem

C A P A B I L I T Y

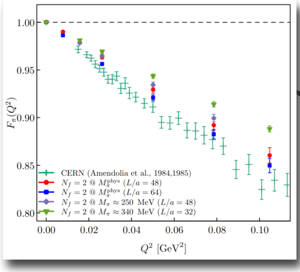
"sample vacuum"
generate ensemble





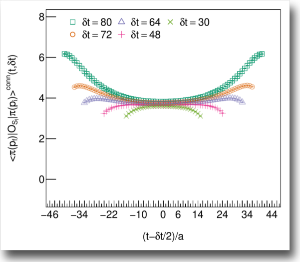
$Z = \int dU \exp(-S_{eff}[U])$

A N A L Y S I S

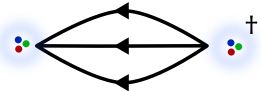


extract physical
observables

compute correlations



"excite vacuum"



$\langle\Omega \mid N(t)N(0)^\dagger \mid \Omega\rangle$

$S_q(x-y) \cong D[U]^{-1}(x,y)\psi(y)$

compute
quark propagators

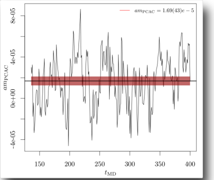
C A P A C I T Y

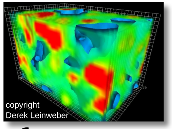
In this talk concentrate on software for first and second stages.

The first two stages of Lattice QCD Calculations

C A P A B I L I T Y

"sample vacuum"
generate ensemble





$Z = \int dU \exp(-S_{eff}[U])$

Ensemble Generation

- Markov Chain Monte Carlo run on largest supercomputers in the world
- *Capability class* or *Strong scaling* problem:
  - run on as many CPU cores or GPUs as is still efficient

Quark Propagators and Correlation Functions

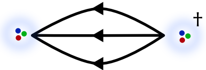
- Physics contained in correlation functions.
  - Mathematical objects which quantify interactions between different particles *created* and *annihilated* at different times
- Run on supercomputers and HPC clusters using as few resources as possible.
  - *capacity* problem, increasingly also needs *capability* resources.

- **Billions of core-hours / tens of millions of GPU-hours.**
- **Petabytes of long-term storage.**

$S_q(x-y) \cong D[U]^{-1}(x,y)\psi(y)$

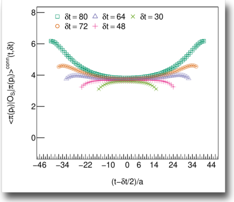
compute
quark propagators

"excite vacuum"



$\langle \Omega \mid N(t)N(0)^\dagger \mid \Omega \rangle$

compute correlations



C A P A C I T Y

The Lattice Field Theory / Lattice QCD Community

State of RSE in LQCD

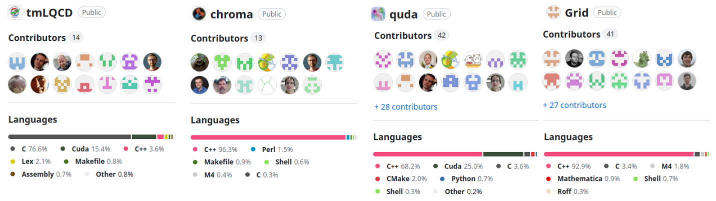
- LQCD codes traditionally written by small number of "*user-developers*".
- Algorithms historically simple compared to, e.g. multi-physics or theoretical chemistry.

$\Rightarrow$ Good: often among first groups on new architectures.
$\Rightarrow$ Bad: ad-hoc solutions $\rightarrow$ production code.

- Since about 2014: growing algorithmic complexity.
- HPC heterogeneity has become an issue. Would like to:
  - Keep pace with hardware diversification.
  - Keep production stack running.
  - Integrate short-term postdocs and PhD students.
  - Do continuous integration on GPU hardware:
    - ★ expensive (cloud) or
    - ★ mostly unsupported (HPC systems).
- RSE culture difficult to establish.

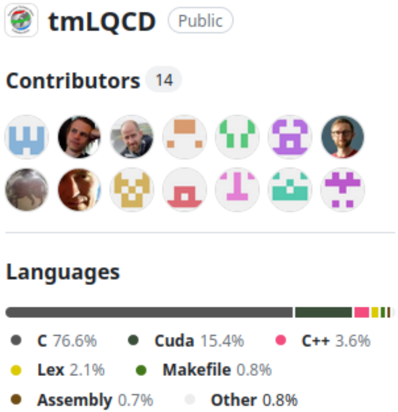
- Theoretical physicists, related to high energy, particle and hadron/nuclear physics.
- Yearly LATTICE conference attracts 500-800 participants.
  - Small compared to others with similarly sized computational requirements.

Examples of LQCD Codes and Libraries



This talk: tmLQCD & QUDA

gh.com/etmc/tmLQCD



Software suite with 20 year history started by Carsten Urbach $\sim$ 140k LOC (C, C++).

- Hybrid Monte Carlo (HMC) algorithm for Wilson fermions.
  (C. Urbach and K. Jansen, Comput.Phys.Commun. 180 (2009) 12)
- OpenMP and MPI parallelisation, support for various architectures.
  (PoS LATTICE2013 (2014) 414), (PoS LATTICE2013 (2014) 416)
- **Leverage various libraries for features and architecture support:**
  - MPI-I/O through the LEMON library.
    (A. Deuzeman et al., Comput.Phys.Commun. 183 (2012))
  - AVX512 support through the QPhiX library.
    (Joó et al., ISC (2016)),(PoS LATTICE2015 (2016) 030)
  - Advanced multigrid-preconditioned solver through DDαAMG library.
    (Frommer et al., SIAM J.Sci.Comput. 36 (2014) 4),(Alexandrou et al., Phys.Rev.D 94 (2016) 11, 114509)
  - GPU support through the QUDA library by NVIDIA.
    (M.A. Clark et al., Comput.Phys.Commun. 181 (2010) 9), (R. Babich et al., SC'11 (2011) 70), (M.A. Clark, SC'16 (2016) 68), (PoS LATTICE2022(2023) 340)

Extended Twisted Mass Collaboration

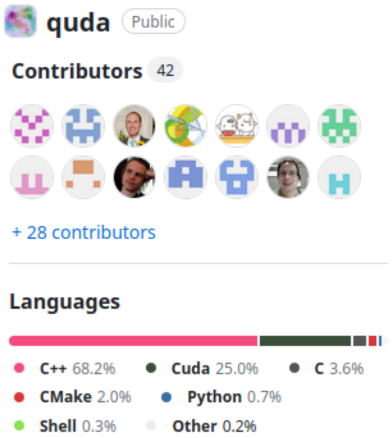
The QUDA library

Started in 2008 by Kate Clark at Boston University, now in wide use as the GPU backend for many LQCD codes $\sim$ 200k LOC (C++, CUDA).

- Key aspects which enable our science:
  - Backends for hardware from other vendors even though QUDA is an NVIDIA project.
  - Fine-tuned for highest performance.
  - Completely open development model:
    - ★ contributions welcome, lots of support
    - ★ can follow entire evolution on github
  - High test coverage, contributions must follow coding standards and provide tests.
  - High level C interface for most functionality.
    - ★ many LQCD codes are written in C, including tmLQCD

The QUDA library and the interactions with its developers have been *essential* for us over the past years.

gh.com/lattice/quda

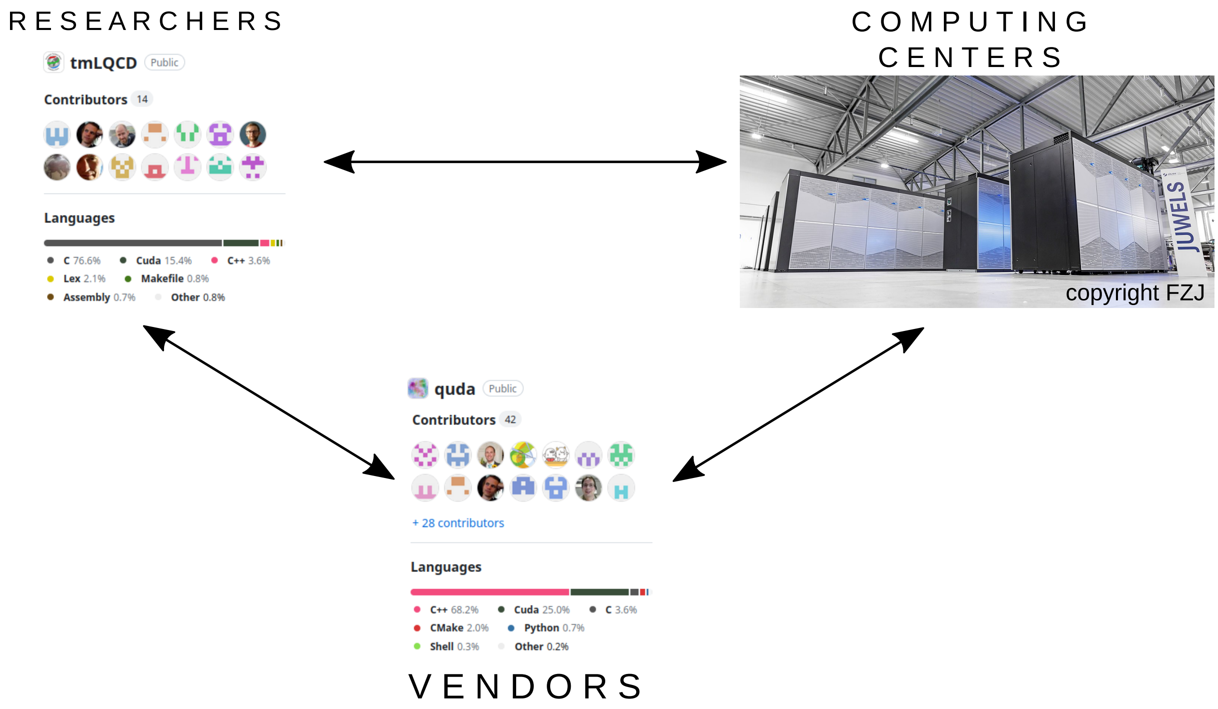


Important technical features

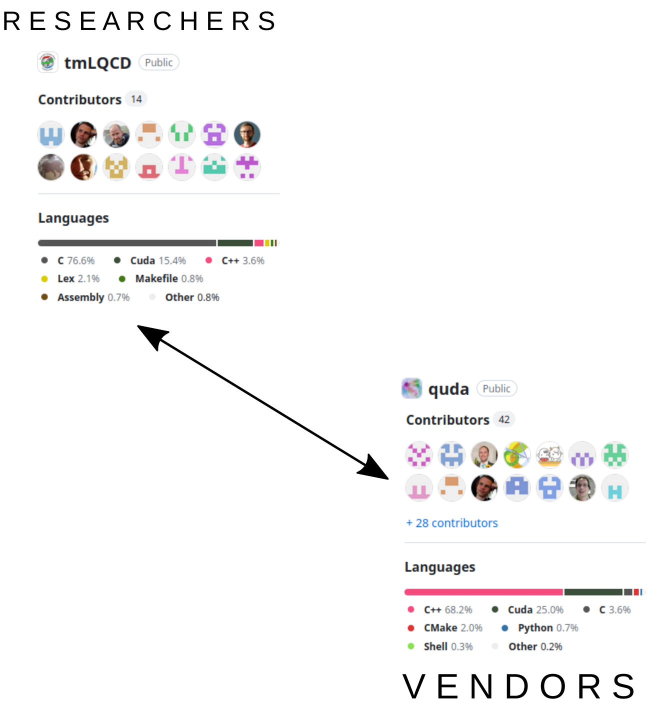
- Provides solvers for most fermionic discretisations & gauge evolution algorithms.
- Mixed-precision methods & autotuning of kernel launch parameters and communication policies.
- Highly efficient multigrid solver for problems with large condition number.
- NVSHMEM for improved strong scaling.
- Major performance-portability effort: HIP (merged), SYCL (in review), OpenMP (in development)

(M.A. Clark et al., Comput.Phys.Commun. 181 (2010) 9), (R. Babich et al., SC'11 (2011) 70), (M.A. Clark, SC'16 (2016) 68)

Collaboration between Researchers, Computing Centers and Hardware Vendors



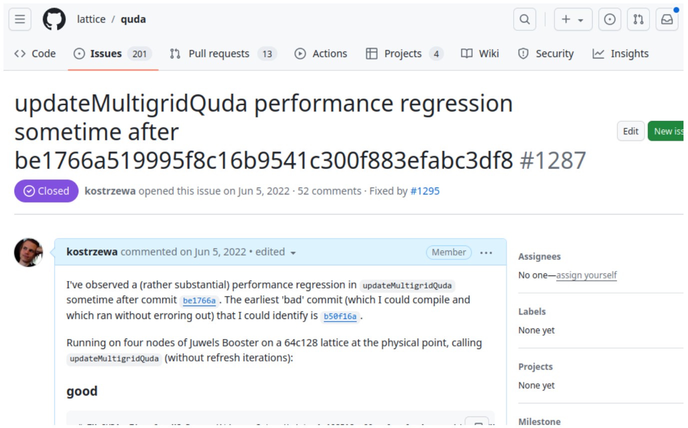
Working on the QUDA library



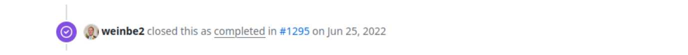
Working on the QUDA library

- We are a small collaboration → reaching performance offered by QUDA would be hard with a self-developed library *and* would need many more people to maintain.
- Very fruitful interaction through github, e-mail and video calls.
- **Examples:**

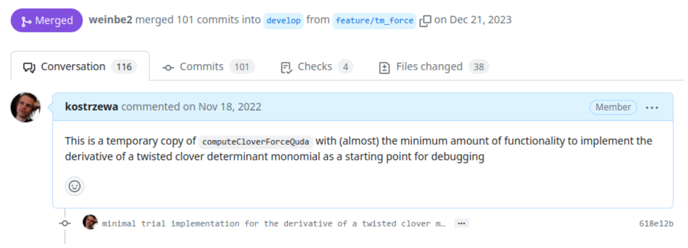
Tackling performance regressions



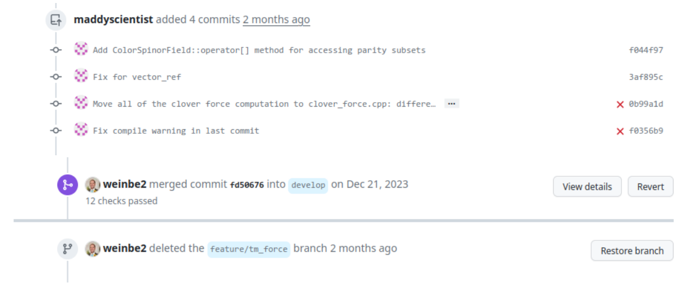
Resolution



Introduction of new features

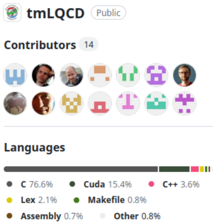


Polishing and Merging



Interaction with Juelich Supercomputing Centre (JSC)

RESEARCHERS



COMPUTING CENTERS

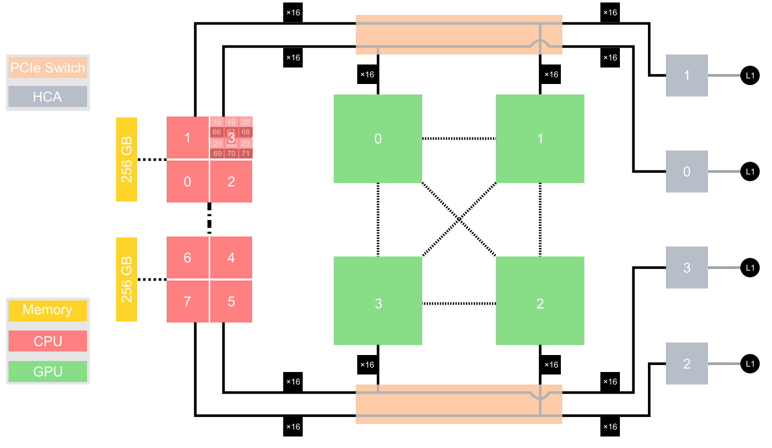
- JUWELS Booster Early Access Programme
- Long-term support with a very interesting issue

JUWELS Booster early access programme

Early Access to JUWELS Booster

- **Late 2020**: Installation of JUWELS Booster.
- From **September 2020 to January 2021** EA programme.
  - **Slack channel** for quick exchange.
  - **Documentation** evolved as part of the programme.
  - **Step by step opening**, real time feedback possible.
- Valuable interaction with JSC and other groups:
  - Compilation recipes.
  - Pinning configuration to account for node topology.
  - Resolution of issues with PCIe firmware.
- As machine stabilized, early access production:
  - Large scale test of the machine with quasi-production job mix.
  - Allowed a massive number of calculations to run which would otherwise not have been possible.
- **January 2021**: online symposium to present test results, exchange experiences.

JUWELS Booster Node Topology



EA programme enabled many publications:
(Phys. Rev. D 107, 054504)
(Phys. Rev. D 107, 074506)
(Phys. Rev. D 108, 094514)
(Eur.Phys.J.C 81 (2021) 5, 436)
(Phys. Rev. Lett. 130, 241901)
& further computing time applications and papers which build on these.

Very valuable experience and an important part of working on research software targeting HPC systems.

Long-term support with vexing JUWELS Booster node crashes

This situation and the way it was handled by the team at JSC convinced me to prepare this talk.

Timeline of a node failure analysis

- **May 22nd, 2022**: a particular problem size on a particular number of nodes leads to node failures in around 25% of cases.
  - Same executable with different problem size on various node numbers does not show this issue…
- **June 15th, 2022**: Using a reproducer, Ahmed Fahmy confirms problem, code triggers BERT CPU error.
- **June 22nd, 2022**: Damian Alvarez reaches out to Atos, NVIDIA, AMD for support, suspecting a weird hardware problem.
- **Dec 12th, 2022**: It has become clear that it's a hard crash, *"something is sending a package in the PCI bus that the CPU does not know how to handle and as a result it crashes"*

On the way to a solution

- **Feb 9th, 2023**: JSC reproduces issues on JURECA DC GPU nodes.
- **May-Sep, 2023**: Lots of internal investigation and contact to Atos, exchange with Meluxina team (similar hardware, no crashes there).
- **Oct, 2023**: JSC provides dedicated reservation to test on while monitoring hardware sensors.
  - *"We noticed earlier that the crashing node experiences a low voltage value (Almost idling voltage value), for one or more very short time intervals, on one of the 2 CPUs of the crashing node, and then at a later point the node crashes. The low voltage readings do not show on any of the nodes allocated for the job, except for the crashing node."*
- **Nov, 2023**: JSC provides workaround $\rightarrow$ GPUs in lower power mode, ondemand CPU governor.
- Now waiting for AMD for permanent fix.

Didn't expect to find a CPU bug :)

Close interaction with support teams essential with these complicated HPC systems!

Thanks to everyone involved!


Simone Romiti

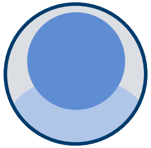

Aniket Sen


Marco Garofalo


Carsten Urbach


Bartosz Kostrzewa

UNIVERSITÄT BONN


Ahmed Fahmy


Damian Alvarez


Andreas Herten

JÜLICH Forschungszentrum


Bálint Joó

OAK RIDGE National Laboratory


Dean Howarth


Jacob Finkenrath

BERGISCHE UNIVERSITÄT WUPPERTAL


Simone Bacchio


Ferenc Pittler

THE CYPRUS INSTITUTE
RESEARCH·TECHNOLOGY·INNOVATION


Kate Clark


Evan Weinberg


Mathias Wagner

NVIDIA.

...and many others who have contributed explicitly or implictly!