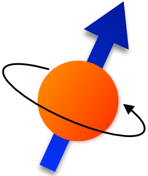
The muon anomalous magnetic moment: how supercomputers can help us find new physics

Christoph Lehner (Regensburg & BNL)

October 8, 2021 - CCS International Symposium 2021

What is a muon?



- Elementary point-like particle
- Same electric charge as an electron
- Approximately **200** times heavier than an electron
- Like the electron, behaves as if it was intrinsically spinning about a vector $\vec{S}$

These properties combine to give it a magnetic moment

$$\vec{\mu} = g\left(\frac{e}{2m}\right)\vec{S}$$

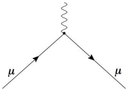
such that when put in a magnetic field, it exhibits precession similar to a spinning top.

We can measure this precession **very** precisely.

The magnetic moment and quantum corrections

The g-factor in $\vec{\mu} = g \left(\frac{e}{2m}\right) \vec{S}$ describes the strength of coupling to a magnetic field, which can be computed from theory also **very** precisely.

Dirac: $g = 2$



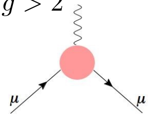
Anomalous magnetic moment

$$a = (g - 2)/2$$

$\xrightarrow{\hspace{4em}}$

quantum effects

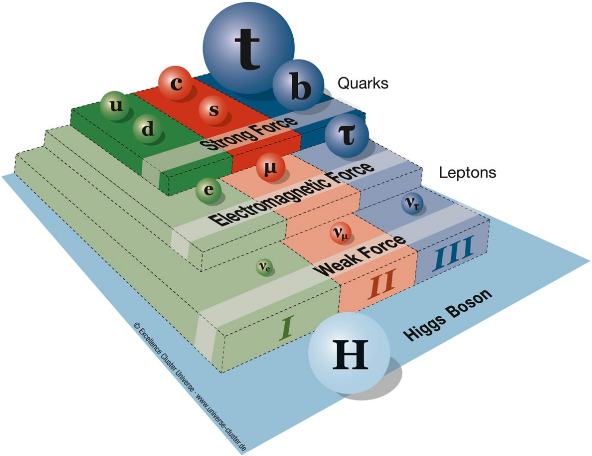
$g > 2$



The quantum effects arise from virtual particle contributions from all known **and unknown** particles.

By comparing high-precision experiments and theory, we have the potential to learn about such contributions of new particles.

Contributions from known particles: The Standard Model

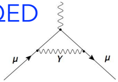


Open questions: dark matter, size of matter-antimatter asymmetry, origin of neutrino masses, ... ⇒ Standard Model is incomplete

Contributions from known particles: The Standard Model

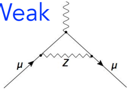
$$a_\mu(\mathrm{SM}) = a_\mu(\mathrm{QED}) + a_\mu(\mathrm{Weak}) + a_\mu(\mathrm{Hadronic})$$

QED



$+\ldots$ $\quad 116\,584\,718.9\,(1) \times 10^{-11}$ $\quad 0.001$ ppm

Weak

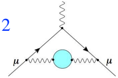


$+\ldots$ $\quad 153.6\,(1.0) \times 10^{-11}$ $\quad 0.01$ ppm

Hadronic…

…Vacuum Polarization (HVP)

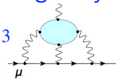
α^2



$+\ldots$ $\quad 6845\,(40) \times 10^{-11}$ [0.6%] $\quad 0.37$ ppm

…Light-by-Light (HLbL)

α^3

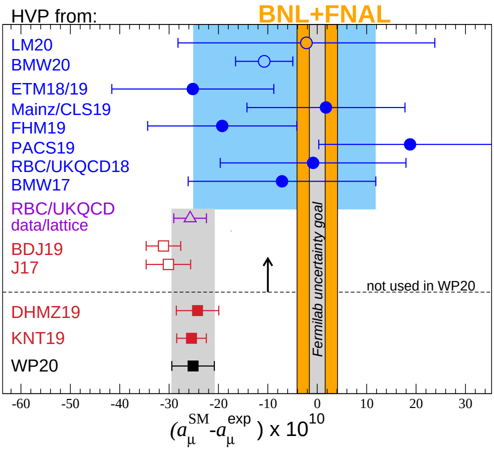


$+\ldots$ $\quad 92\,(18) \times 10^{-11}$ [20%] $\quad 0.15$ ppm

Numbers from Theory Initiative Whitepaper

Uncertainty dominated by hadronic contributions

Status and impact of hadronic vacuum polarization contribution



Ab-initio lattice QCD(+QED) calculations

Hybrid window method restricts scales that enter from lattice/dispersive data

Dispersive, $e^+e^- \to$ hadrons (20+ years of experiments)

Lattice QCD computation of the hadronic vacuum polarization

We can express

$$a_\mu^{\mathrm{HVP}} = \sum_t w_t C(t)$$

with analytically calculable w_t and

$$C(t) = \sum_{\vec{x}} \langle \mathrm{Tr}[D(U)^{-1}_{\vec{x},t;\vec{0},0} \Gamma D(U)^{-1}_{\vec{0},0;\vec{x},t} \Gamma] \rangle \,,$$

where $\langle \cdot \rangle$ denotes the expectation value over a certain ensemble of $\mathrm{SU}(3)^{4V}$ matrices U with V being a four-dimensional space-time volume. Γ are matrices in an internal 12-dimensional space.

(V can be 10^9)

The Wilson Dirac operator

A substantial part of the numerical challenge lies in inverting the operator

$$
\begin{aligned}
D(U)_{x,y} = \frac{1}{2}\sum_{\mu=0}^{3} \delta_{x+\hat{\mu},y}(\gamma_\mu - \mathbb{1})U_\mu(x) \\
- \frac{1}{2}\sum_{\mu=0}^{3} \delta_{x-\hat{\mu},y}(\gamma_\mu + \mathbb{1})U_\mu^\dagger(y) \\
+ \frac{1}{2}\kappa\delta_{x,y}
\end{aligned}
$$

with 4×4 matrices γ_μ, real number κ, and unit vectors $\hat{\mu}$.

High-performance computing

Grid Python Toolkit (GPT)

https://github.com/lehner/gpt

- A toolkit for lattice QCD and related theories as well as QIS (a parallel digital quantum computing simulator) and Machine Learning
- Python frontend, C++ backend
- Built on Grid data parallelism (MPI, OpenMP, SIMD, and SIMT)

Initial commit Feb. 2020, 47k lines of C++/Python, >1400 commits so far, 12 contributors

Guiding principles:

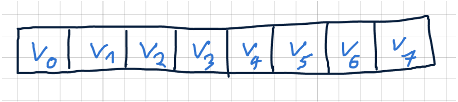
- Performance Portability
  common Grid-based framework for current and future (exascale) architectures

- Modularity / Composability
  build up from modular high-performance components, several layers of composability, "composition over parametrization"

The Grid data parallelism paradigm

https://github.com/paboyle/Grid

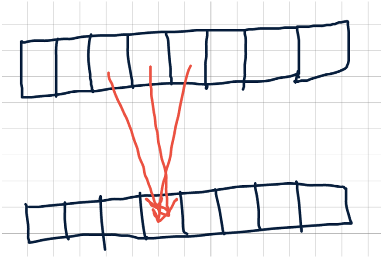
Start with a vector $v_x \in O$ with $x \in L$ and a d-dimensional Cartesian lattice L. Examples below have $d = 1$ and $L = \{0, \ldots, 7\}$.



In lattice QCD, L makes up a space-time grid and v will be fermionic/bosonic fields.

High-performance building block: small stencil operators

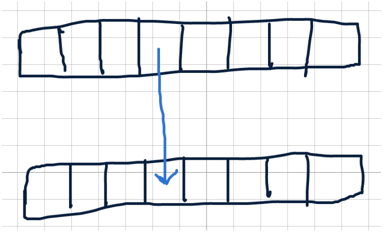
Common in lattice QCD: local operators with a small stencil (examples: Dirac matrix, Δ operator)



For such transformations, only knowledge of a few neighbors is needed.

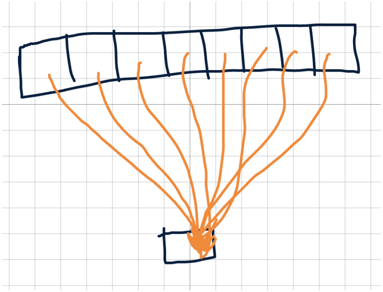
High-performance building block: site-local operators

Examples: (bi-)linear combinations of vectors

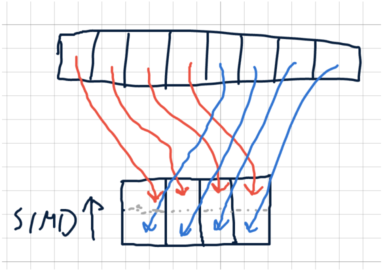


High-performance building block: reductions

Examples: inner product in lattice QCD, probability of measurement



For all these operations, the following data grouping preserves locality:



Such a group can be combined to a single SIMD word or mapped on a (fastest moving) thread index for coalesced memory access in SIMT architectures (Grid's SIMD/SIMT paradigm):

$$s_0 \equiv \begin{pmatrix} v_0 \\ v_4 \end{pmatrix}, \quad s_1 \equiv \begin{pmatrix} v_1 \\ v_5 \end{pmatrix}, \quad s_2 \equiv \begin{pmatrix} v_2 \\ v_6 \end{pmatrix}, \quad s_3 \equiv \begin{pmatrix} v_3 \\ v_7 \end{pmatrix} \tag{1}$$

Size of lattice of s reduces depending on SIMD word size.

Example: derivative on periodic lattice

The 8 operations

$$v_i' = v_{i+1 \mod 8} - v_i \tag{2}$$

with $i \in \{0, 1, \dots, 7\}$ turn into 4 operations on SIMD words

$$s_j' = s_{j+1} - s_j \tag{3}$$

with $j \in \{0, 1, 2, 3\}$ and border permutation

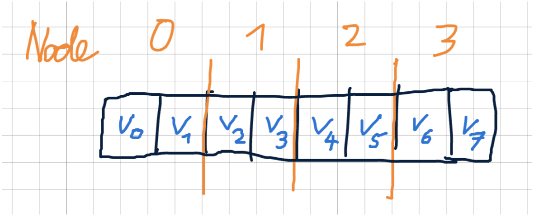
$$s_4 \equiv \begin{pmatrix} v_4 \\ v_0 \end{pmatrix} . \tag{4}$$

Check:

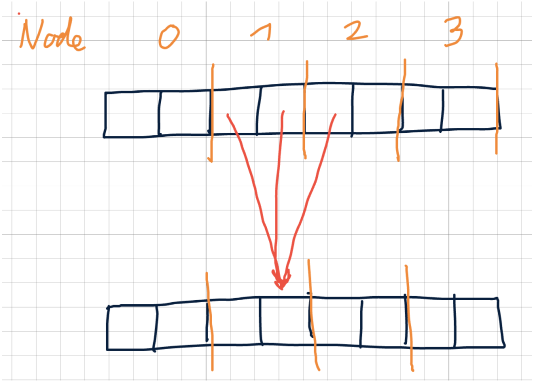
$$s_0 \equiv \begin{pmatrix} v_0 \\ v_4 \end{pmatrix} , \qquad s_1 \equiv \begin{pmatrix} v_1 \\ v_5 \end{pmatrix} , \qquad s_2 \equiv \begin{pmatrix} v_2 \\ v_6 \end{pmatrix} , \qquad s_3 \equiv \begin{pmatrix} v_3 \\ v_7 \end{pmatrix}$$

MPI parallelism

Here we allow for a d-dimensional Cartesian partition of the lattice L:



Challenge for Lattice QCD: small stencil operations



Only communication between neighboring nodes needed. Communication burden generally suppressed by surface to volume ratio.

GPT - layout and dependencies

Python script / Jupyter notebook

gpt (Python)

- Defines data types and objects (group structures etc.)
- Expression engine (linear algebra)
- Algorithms (Solver, Eigensystem, ...)
- File formats
- Stencils / global data transfers
- QCD, QIS, ML subsystems

cgpt (Python library written in C++)

- Global data transfer system (gpt creates pattern, cgpt optimizes data movement plan)
- Virtual lattices (tensors built from multiple Grid tensors)
- Optimized blocking, linear algebra, and Dirac operators
- Vectorized ranlux-like pRNG (parallel seed through 3xSHA256)

Grid | Eigen | FFTW

Example: solvers are modular and can be mixed

General design principle: use modularity of python code instead of large number of parameters to configure solvers/algorithms;
Python can also be used in configuration files

```
# Create an coarse-grid deflated, even-odd preconditioned CG inverter
# (eig is a previously loaded multi-grid eigensystem)
sloppy_light_inverter = g.algorithms.inverter.preconditioned(
    g.qcd.fermion.preconditioner.eo1_ne(parity=g.odd),
    g.algorithms.inverter.sequence(
        g.algorithms.inverter.coarse_deflate(
            eig[1],
            eig[0],
            eig[2],
            block=200,
        ),
        g.algorithms.inverter.split(
            g.algorithms.inverter.cg({"eps": 1e-8, "maxiter": 200}),
            mpi_split=[1,1,1,1],
        ),
    ),
)
```

All algorithms implemented in Python – Example: Euler-Langevin stochastig DGL integrator

```python

class langevin_euler:
    @g.params_convention(epsilon=0.01)
    def __init__(self, rng, params):
        self.rng = rng
        self.eps = params["epsilon"]

    def __call__(self, fields, action):
        gr = action.gradient(fields, fields)
        for d, f in zip(gr, fields):
            f @= g.group.compose(
                -d * self.eps
                + self.rng.normal_element(g.lattice(d)) * (self.eps * 2.0) ** 0.5,
                f,
            )

```

Implemented algorithms:

- BiCGSTAB, CG, CAGCR, FGCR, FGMRES, MR solvers
- Multi-grid, split-grid, mixed-precision, and defect-correcting solver combinations
- Coarse and fine-grid deflation
- Arnoldi, implicitly restarted Lanczos, power iteration
- Chebyshev polynomials
- All-to-all vector generation
- SAP and even-odd preconditioners
- Gradient descent and non-linear CG optimizers
- Runge-Kutta integrators, Wilson flow
- Fourier acceleration
- Coulomb and Landau gauge fixing
- Domain-wall–overlap transformation and MADWF
- Symplectic integrators (leapfrog, OMF2, and OMF4)
- Markov: Metropolis, heatbath, Langevin, HMC in progress

Performance

Benchmark results committed to github

https://github.com/lehner/gpt/tree/master/benchmarks/reference

master | gpt / benchmarks / reference /

lehner supermuc-ng timing ··· on Apr 1 History

..	
bnl_knl	8 months ago
juron	8 months ago
juwels_booster	2 months ago
lrz_supermuc_ng	last month
qpace3	8 months ago
qpace4	4 months ago
stampede2_knl	6 months ago
summit	8 months ago

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**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 United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**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
5	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646
6	**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
7	**JUWELS Booster Module** - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, Atos Forschungszentrum Juelich (FZJ) Germany	449,280	44,120.0	70,980.0	1,764

Results available for GPU and CPU architectures. In the following, focus on Juwels booster (NVIDIA A100) and QPace4 (A64FX, same as Fugaku).

Juwels Booster (node has 4× A100-40GB): Single-node domain-wall fermion $\not{D}$ operator

```
============================================
              Initialized GPT
     Copyright (C) 2020 Christoph Lehner
============================================
GPT :       1.543473 s :
                       : DWF Dslash Benchmark with
                       :     fdimensions  : [64, 32, 32, 32]
                       :     precision    : single
                       :     Ls           : 12
                       :
GPT :       7.958636 s : 1000 applications of Dhop
                       :     Time to complete            : 2.93 s
                       :     Total performance           : 11325.46 GFlops/s
                       :     Effective memory bandwidth  : 7824.86 GB/s
GPT :       7.959499 s :
                       : DWF Dslash Benchmark with
                       :     fdimensions  : [64, 32, 32, 32]
                       :     precision    : double
                       :     Ls           : 12
                       :
GPT :      17.420620 s : 1000 applications of Dhop
                       :     Time to complete            : 5.78 s
                       :     Total performance           : 5749.77 GFlops/s
                       :     Effective memory bandwidth  : 7945.14 GB/s
============================================
               Finalized GPT
============================================
```

Compare to HBM bandwidth of 1,555 GB/s per GPU

QPace4 (node has one A64FX): Single-node domain-wall fermion $\not{D}$ operator

```
============================================
              Initialized GPT
    Copyright (C) 2020 Christoph Lehner
============================================
GPT :       0.265714 s :
                       : DWF Dslash Benchmark with
                       :     fdimensions  : [24, 24, 24, 24]
                       :     precision    : single
                       :     Ls           : 8
                       :
GPT :      20.218240 s : 1000 applications of Dhop
                       :     Time to complete            : 3.67 s
                       :     Total performance           : 954.90 GFlops/s
                       :     Effective memory bandwidth  : 677.11 GB/s
GPT :      20.218842 s :
                       : DWF Dslash Benchmark with
                       :     fdimensions  : [24, 24, 24, 24]
                       :     precision    : double
                       :     Ls           : 8
                       :
GPT :      45.245379 s : 1000 applications of Dhop
                       :     Time to complete            : 7.36 s
                       :     Total performance           : 475.80 GFlops/s
                       :     Effective memory bandwidth  : 674.77 GB/s
============================================
               Finalized GPT
============================================
```

Compare to HBM bandwidth of 1,000 GB/s per A64FX

Juwels Booster (node has 4$\times$ A100-40GB): Single-node site-local matrix products

```
============================================
          Initialized GPT
     Copyright (C) 2020 Christoph Lehner
============================================
GPT :       1.589357 s :
                       : Matrix Multiply Benchmark with
                       :     fdimensions  : [48, 48, 48, 128]
                       :     precision    : single
                       :
GPT :      10.985099 s : 10 matrix_multiply
                       :     Object type                 : ot_matrix_color(3)
                       :     Time to complete            : 0.0058 s
                       :     Effective memory bandwidth  : 5271.36 GB/s
                       :
GPT :      16.689329 s : 10 matrix_multiply
                       :     Object type                 : ot_matrix_spin(4)
                       :     Time to complete            : 0.01 s
                       :     Effective memory bandwidth  : 5333.21 GB/s
                       :
GPT :      62.092583 s : 10 matrix_multiply
                       :     Object type                 : ot_matrix_spin_color(4,3)
                       :     Time to complete            : 0.097 s
                       :     Effective memory bandwidth  : 5057.37 GB/s
                       :
```

```
GPT :      62.262581 s :
                       : Matrix Multiply Benchmark with
                       :     fdimensions  : [48, 48, 48, 128]
                       :     precision    : double
                       :
GPT :      72.003471 s : 10 matrix_multiply
                       :     Object type                 : ot_matrix_color(3)
                       :     Time to complete            : 0.012 s
                       :     Effective memory bandwidth  : 5264.01 GB/s
                       :
GPT :      78.174681 s : 10 matrix_multiply
                       :     Object type                 : ot_matrix_spin(4)
                       :     Time to complete            : 0.02 s
                       :     Effective memory bandwidth  : 5439.91 GB/s
                       :
GPT :     128.232979 s : 10 matrix_multiply
                       :     Object type                 : ot_matrix_spin_color(4,3)
                       :     Time to complete            : 0.22 s
                       :     Effective memory bandwidth  : 4416.45 GB/s
                       :
============================================
          Finalized GPT
============================================
```

Compare to HBM bandwidth of 1,555 GB/s per GPU

Juwels Booster (node has 4$\times$ A100-40GB): Inner product (reduction)

```
GPT :      28.406798 s : 100 rank_inner_product
                       :     Object type                : ot_vector_singlet(12)
                       :     Block                      : 4 x 4
                       :     Data resides in            : accelerator
                       :     Performed on               : accelerator
                       :     Time to complete           : 0.13 s
                       :     Effective memory bandwidth  : 4827.16 GB/s
                       :
                       :     rip: timing: unprofiled          = 0.000000e+00 s (=   0.00 %)
                       : rip: timing: rip: view           = 9.706020e-04 s (=   0.70 %)
                       : rip: timing: rip: loop           = 1.369879e-01 s (=  99.30 %)
                       : rip: timing: total               = 1.379585e-01 s (= 100.00 %)
                       :
```

Compare to HBM bandwidth of 1,555 GB/s per GPU

Performance summary

Machine	Operation	Performance	Bandwidth
Booster	$\not{D}$	12 TF/s	7.8 TB/s
Booster	ColorMatrix $\times$		5.2 TB/s
Booster	SpinColorMatrix $\times$		5.1 TB/s
Booster	SpinColorVector $\langle\cdot,\cdot\rangle$		4.8 TB/s
QPace4	$\not{D}$	0.95 TF/s	0.68 TB/s
SuperMUC-NG	$\not{D}$	0.72 TF/s	0.51 TB/s

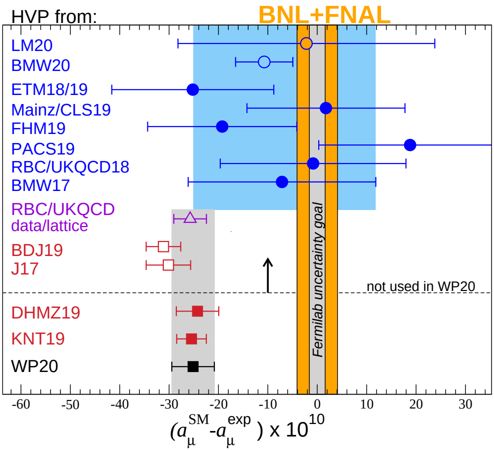
Single-node SP performance of Wilson $\not{D}$ and linear algebra on Juwels Booster (4×A100, HBM BW 1.6 TB/s per A100), Qpace4 (A64FX, HBM BW of 1 TB/s per node), and the SuperMUC-NG (Skylake 8174). The $\not{D}$ performance is inherited from Grid, the linear algebra performance is based on cgpt.

Total cost of a high-precision calculation of a_μ^{HVP}

- Need the equivalent of several 100,000 inversions of $D(U)$ on lattices of size $96 \times 96 \times 96 \times 192$.

- This corresponds to several hundred million core hours on leadership class supercomputers.

Status and impact of hadronic vacuum polarization contribution



Ab-initio lattice QCD(+QED) calculations

Hybrid window method restricts scales that enter from lattice/dispersive data

Dispersive, $e^+e^- \to$ hadrons (20+ years of experiments)

Uncertainties of lattice QCD results expected to be reduced by an order-of-magnitude in coming years. Clarify: New Physics needed to explain tension?