Lattice Quantum Chromodynamics on the MIC architectures

Piotr Korcyl

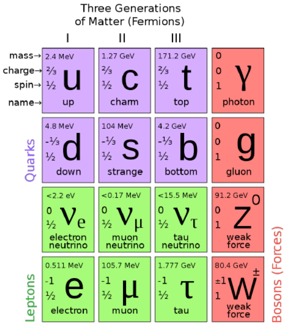
Universität Regensburg

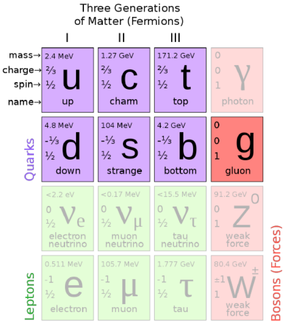
Intel MIC Programming Workshop @ LRZ
28 June 2017

Regensburg's Lattice QCD group

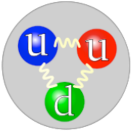
Software/Hardware	Physics
Tilo Wettig	Andreas Schäfer
Simon Heybrock	Gunnar Bali
Jacques Bloch	Sara Collins
Stefan Solbrig	Enno Scholz
Nils Meyer	Wolfgang Söldner
Robert Lohmayer	postdocs + PhD students
Peter Georg	
Daniel Richtmann	
postdocs + Phd students	

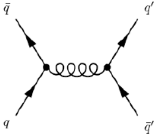
Standard Model of elementary particles and Quantum Chromodynamics





QFT describes the strong interaction at the fundamental level.





Quantum Chromodynamics: hadrons and their properties

Distinctive features

- asymptotic freedom: interation becomes weaker at shorter distances;
- quarks and gluons confined into colourless bound states (hadrons);
- spontaneous symmetry breaking determines low-energy dynamics.

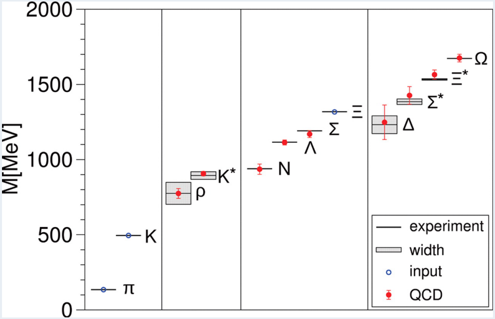


Figure: Particle spectrum measured by experiment and calculated by BMW Collaboration.

Lattice Quantum Chromodynamics

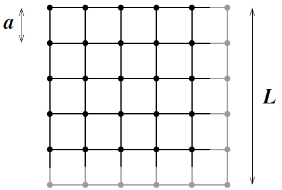
Continuum Quantum Chromodynamics

$$\mathscr{S}_{\text{QCD}} = \int d^4x \text{Tr}\left\{ \frac{1}{2{g_0}^2} F_{\mu\nu}F_{\mu\nu} + \sum_{i,j} \bar{\psi}_i \left[i(\gamma^\mu D_\mu)_{ij} - m_{0,i}\delta_{ij} \right] \psi_j \right\}$$

$$Z(g_0^2, m_{0,i}) = \int [A_\mu(x)][\bar{\psi}(x)][\psi(x)] \exp\{i\mathscr{S}_{\text{QCD}}(g_0^2, m_{0,i})\}$$

$$\langle \mathscr{O} \rangle = \frac{1}{Z(g_0^2, m_{0,i})} \int [A_\mu(x)][\bar{\psi}(x)][\psi(x)] \mathscr{O} \exp\{i\mathscr{S}_{\text{QCD}}(g_0^2, m_{0,i})\}$$

Lattice Quantum Chromodynamics



$x = a\,(n_0, n_1, n_2, n_3)\,, \qquad n_\mu \in \mathbb{Z}$

$a \equiv$ lattice spacing

$L/a \equiv$ lattice size

$L \leq \infty$, possibly $T \neq L$

Lattice Quantum Chromodynamics

Continuum Quantum Chromodynamics

$$\mathscr{S}_{\text{QCD}} = \int d^4x \text{Tr}\left\{ \frac{1}{2{g_0}^2} F_{\mu\nu}F_{\mu\nu} + \sum_{i,j} \bar{\psi}_i \left[i(\gamma^\mu D_\mu)_{ij} - m_{0,i}\delta_{ij} \right] \psi_j \right\}$$

$$Z(g_0^2, m_{0,i}) = \int [A_\mu(x)][\bar{\psi}(x)][\psi(x)] \exp\{i\mathscr{S}_{\text{QCD}}(g_0^2, m_{0,i})\}$$

$$\langle \mathscr{O} \rangle = \frac{1}{Z(g_0^2, m_{0,i})} \int [A_\mu(x)][\bar{\psi}(x)][\psi(x)] \mathscr{O} \exp\{i\mathscr{S}_{\text{QCD}}(g_0^2, m_{0,i})\}$$

Lattice Quantum Chromodynamics: Monte Carlo importance sampling

$$\mathscr{S}_{\text{QCD}} = a^4 \sum_x \text{Tr}\left\{ \frac{1}{2{g_0}^2} \sum_{\mu<\nu} \text{Re}\left(1 - \text{U}_{\mu\nu}\right) - \text{tr} \log \det D[U] \right\}$$

$$\langle \mathscr{O} \rangle = \frac{1}{N} \sum_{\text{configuration}} [\text{set of configurations } (g_0^2, m_{0,i})] \mathscr{O}(\text{configuration})$$

distributed with Boltzmann probability weigth $\propto \exp\{-\mathscr{S}_{\text{QCD}}(g_0^2, m_{0,i})\}$.

Algorithms: Markov chain

Markov chain

We need $[\text{set of configurations } (g_0^2, m_{0,i})]$ generated with a probability $\propto \exp\{-\mathscr{S}_{\mathrm{QCD}}(g_0^2, m_{0,i})\}$ where

$$\mathscr{S}_{\mathrm{QCD}}(g_0^2, m_{0,i}) = a^4 \sum_x \mathrm{Tr}\Big\{ \frac{1}{2{g_0}^2} \sum_{\mu<\nu} \mathrm{Re}\big(1 - \mathrm{U}_{\mu\nu}\big) + \phi^\dagger \big(D^\dagger D\big)^{-1} \phi \Big\}$$

where D is the Wilson-Dirac operator $D_{AB}^{\alpha\beta}(x|y)$:

$$\text{sparse matrix of size } (12TL^3) \times (12TL^3), \text{ typically } 10^9 \times 10^9.$$

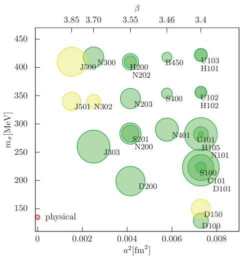
Hybrid Monte Carlo algorithm

- introduce associated momenta with a gaussian probability distribution
- integrate Hamilton's EoM for MC-time interval $\tau = 2.0$:
  - advance gauge and pseudofermion variables: $\dot{Q} = \frac{\partial H}{\partial P} = P$
  - advance momenta: $\dot{P} = -\frac{\partial H}{\partial Q} = -\frac{\partial S}{\partial Q}$
- perform accept/reject step with a probability $\exp(-\Delta S)$

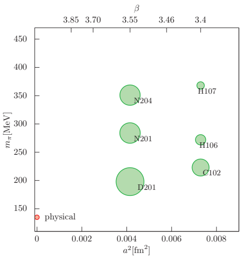
Gauge field configurations

Coordinated Lattice Simulations (CLS) ensembles with 2+1 dynamical fermions

Cost (so far)



$\mathrm{tr}\, m = \mathrm{const}$



$\widetilde{m}_s = \mathrm{const}$

area $\propto$ cost
(1 superMUC core-h $\approx$ 3.7 Juqueen core-h)

$\Rightarrow$ total of 165 Mcore-h superMUC-equivalent, 350 TB of data

Machines

Some of the computer time allocations

Dedicated machines:

- QPACE - PowerXCell 8i multi-core processors
- QPACE 2 - Xeon Phi 7120X accelerators (KNC), 64 nodes, 4 KNC each
- QPACE 3 - Xeon Phi 7210 accelerators (KNL), 320 nodes, 1 KNL each

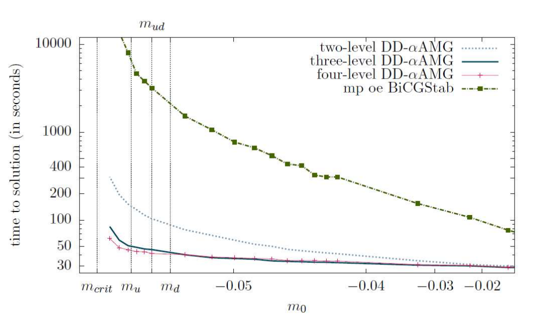
General purpose machines:

- LRZ superMIC
- CINECA Marconi KNL cluster
- LRZ superMUC
- JUQUEEN @ Juelich
- Cray XC40 at Warsaw University

Solving the Dirac equation

State-of-the-art linear solver

Invertions of the Dirac operator contribute significantly to the total numerical cost. Typically an iterative solver with a suitable preconditioner to control the condition number of the Dirac operator is used.



Domain Decomposition Adaptive Algebraic Multigrid Solver (DD-αAMG) is the state-of-the-art solver developed and implemented by mathematicians and physicists from Univ. of Wuppertal and Regensburg.

Solving the Dirac equation

Typical setups

lattices: $32^3 \times 96, \ldots, 64^3 \times 192 \Rightarrow$ MPI-ranks: $512, \ldots, 12288$

State-of-the-art linear solver: overview of implementation details

- global lattice divided into local lattice via MPI (or dedicated library pMR)
- even-odd decomposition allows to work simultaneously on each local lattice
- use preconditioning: choose a matrix $M^{-1} \approx D^{-1}$ and solve $DM^{-1}Mu = DM^{-1}v = f$
- domain decomposition for the preconditioner reduces communication
  - inversions done on each block separately, ideally from cache
  - boundary exchanges not so frequent
- work on local lattices done by threads (persistent openMP threads)

Details

'Lattice QCD with Domain Decomposition on Intel Xeon Phi Co-Processors'
S. Heybrock *at al.*, arXiv:1412.2629

Solving the Dirac equation

KNC/KNL architectures

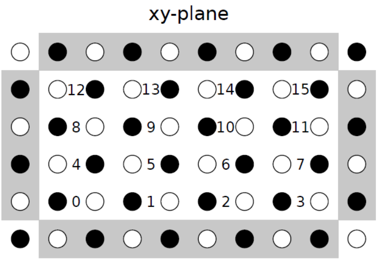
- fuse identical components of fields from different sites (site-fusing) allows to exploit large vector units
- overlap computations and communication to hide communication latency
- half-precision for the preconditioner to reduce memory bandwidth requirements and memory footprint
- multiple right-hand-sides
- real and imaginary parts kept separated in registers

Three examples

- implementation of the domain decomposition preconditioner and the changes in the data layouts $\rightarrow$ SIMD registers
- sophisticated boundary exchanges $\rightarrow$ hide communication latency
- implementation of the three-point functions measurement code and the changes in parallelization $\rightarrow$ load balance

Code developement: three examples

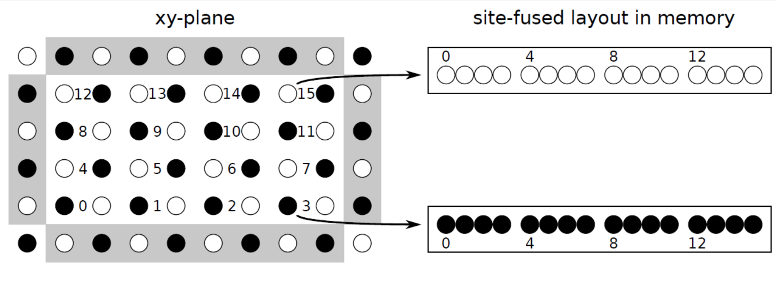
Domain-decomposition preconditioner: SIMD layouts, SOA and AOS



Simon Heybrock's figure

Code developement: three examples

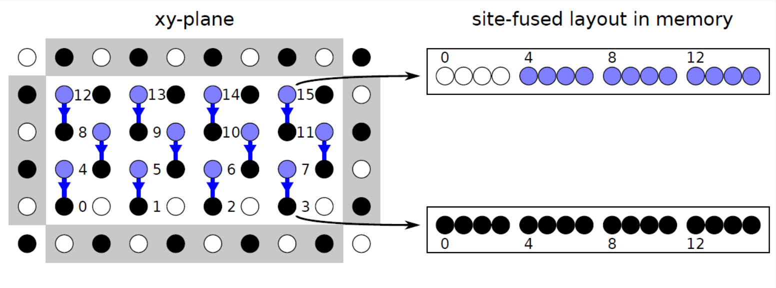
Domain-decomposition preconditioner: SIMD layouts, SOA and AOS



Simon Heybrock's figure

Code developement: three examples

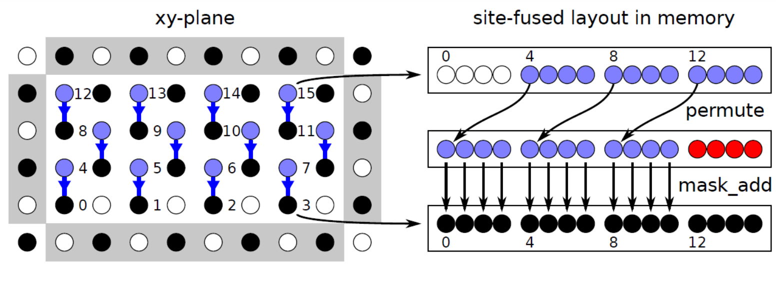
Domain-decomposition preconditioner: SIMD layouts, SOA and AOS



Simon Heybrock's figure

Code developement: three examples

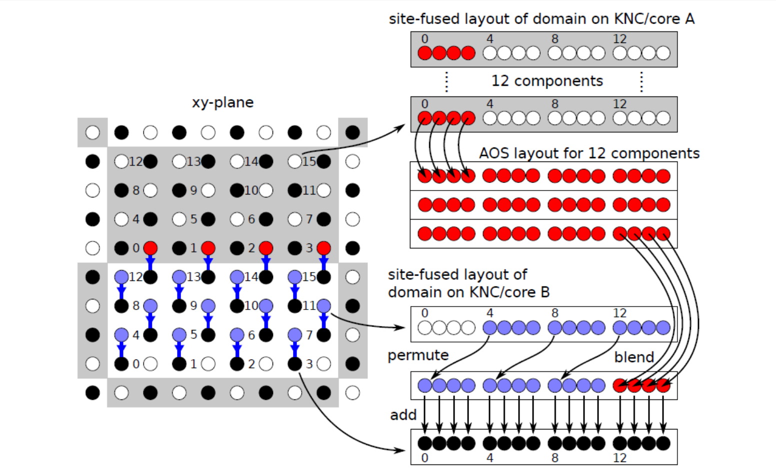
Domain-decomposition preconditioner: SIMD layouts, SOA and AOS



Simon Heybrock's figure

Code developement: three examples

Domain-decomposition preconditioner: SIMD layouts, SOA and AOS

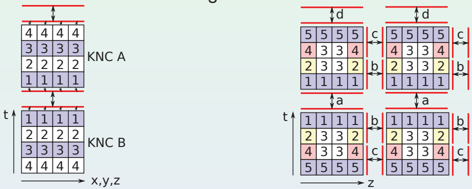


Simon Heybrock's figure

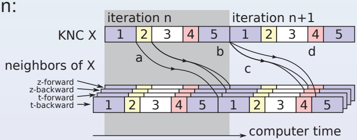
Code developement: three examples

Communication latency hiding for DD

- boxes represent domains, numbers represent order of execution, small letters represent order of communication
- bad: good:



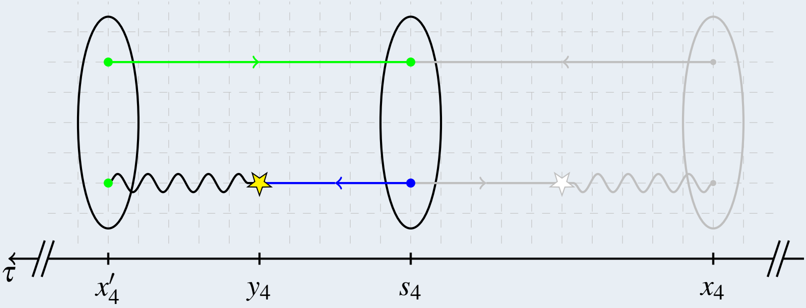
- linear representation:



Tilo Wettig's slide from QCDNA16 workshop.

Code developement: three examples

Three-point correlation functions: load balancing

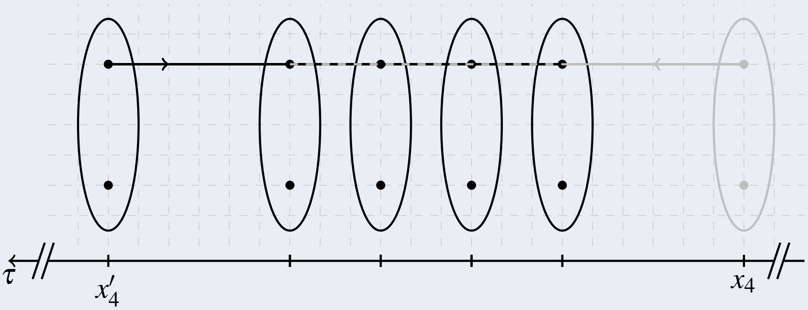


Description:

- each elipse corresponds to $N_x N_y N_z$ 12×12 matrix multiplications
- each star corresponds to $N_x N_y N_z$ 12×12 matrix multiplications
- we need to repeat for different positions of the central elipse and all stars in between

Code developement: three examples

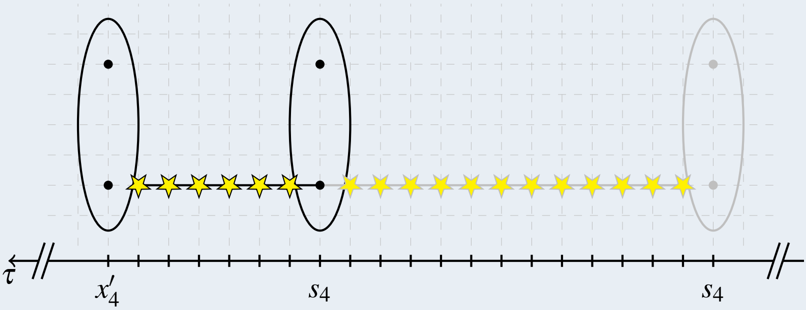
Three-point correlation functions: load balancing



Data redistribution:

- redistribute data along the t direction and compute in parallel

Three-point correlation functions: load balancing



Yet another data redistribution:

- redistribute data along the t direction and compute in parallel

Practical details

KNC/KNL

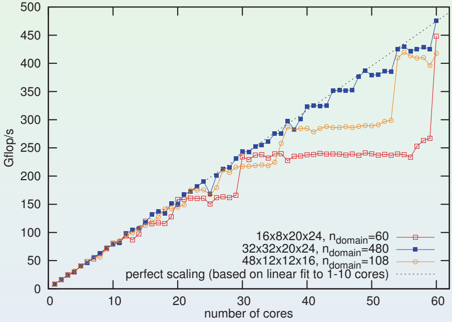
- KNC: we run with 2 or 4 threads/core
  - 2 threads: cache misses
  - 4 threads: cache eviction
- KNL/OmniPath: we run with > 4 MPI ranks / KNL

Example wall-clock timings on QPACE 2 (in seconds)

Lattice size	MDA	BDA	Bar.spec.	H-Bar.spec.	
$32^3 \times 96$	308	777	33	90	Reference
	32	**76**	**13**	**27**	LHA
$32^3 \times 128$	379	1011	51	127	Reference
	53	**112**	**26**	**37**	LHA
$48^3 \times 128$	370	910	47	101	Reference
	49	**102**	**19**	**34**	LHA

Solver single-node scaling

Results: Single KNC



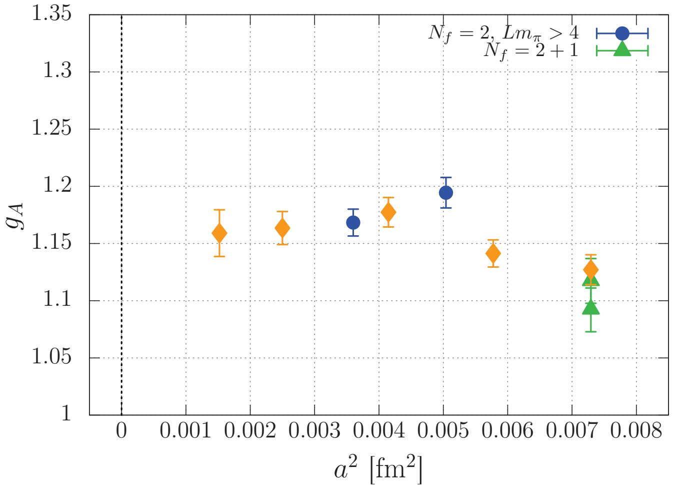
- almost perfect scaling (except for load imbalance):
  - cores can work independently during MR inversion
  - almost no competition for memory access since MR runs from cache

Tilo Wettig's slide from QCDNA16 workshop.

Physical results

Spin content of the nucleon presented at Lattice2017 in Granada

g_A: continuum limit at $M_\pi \approx 420$ MeV



Conclusions

Quantum Chromodynamics and HPC

Lattice QFT & High Performance Computing combined with high precision experiments is the only way to check the correctness of our understanding of the Standard Model and therefore to discover New Physics behind it.

QCD and new architectures

- adapt the application to both processor and interconnect
- choose an appropriate algorithm
- vectorization: data layout!
- load balancing: make sure all cores are busy
- many other ways of enforcing vectorization
- avoiding intrinsics - code portability between KNC and KNL

Thank you for your attention!