CL$^2$QCD
Lattice QCD based on OpenCL

Christopher Pinke
with M. Bach, O. Philipsen & A. Sciarra

Institute for Theoretical Physics
Goethe-University Frankfurt

GPU Computing in High Energy Physics
University of Pisa, September 10-12, 2014
Motivation

Lattice QCD

CL\(^2\)QCD: Code Structure

CL\(^2\)QCD: Performance

Summary & Perspectives
Strong Interactions: Quantum Chromodynamics (QCD)

\[ S_{\text{QCD}} = \int d^4x \bar{\psi}(x) D[A_\mu] \psi(x) - \frac{1}{4} F_{\mu\nu}^a [A_\mu](x) F_{\alpha,\mu\nu}^a [A_\alpha](x) \]

Confinement

Quarks carry 3 colours transmitted by Gluons. Confinement of quarks and gluons into colourless objects.

Running coupling

Politzer, Wilczek, Gross (NP 2004)

Betheke (2007)

QCD non-perturbative/strong at low energies

Studies from first principles \(\rightarrow\) Lattice QCD

C. Pinke (GU Frankfurt)
Exploring the QCD phase diagram

Critical Endpoint (CEP): Does it exist?
Exploring the QCD phase diagram

Sign-Problem of LQCD:

$$\det(D + \mu)^* = \det(D - \mu^*) \Rightarrow \text{Importance sampling ill-defined}$$
Exploring the QCD phase diagram

Sign-Problem of LQCD:
Solutions: Taylor expansion, Reweighting, \textit{Imaginary }\mu
Lattice QCD (I)

Discretize spacetime:

\[ S_{\text{LQCD}} = S_{\text{QCD}} + aS_1 + a^2S_2 + \ldots \]

Continuum limit: \( a \to 0 \)

Fermions on the lattice

Different discretizations:

- **Kogut-Susskind/Staggered**
  - Remnant chiral symmetry, cheap, Rooting

- **Wilson**
  - Theoretically sound, no chiral symmetry

- **Ginsparg-Wilson**
  - Chiral symmetry, very expensive

Consistency with experiment

BMW Collaboration, 2008
Lattice QCD (II)

Apply Importance Sampling: Hybrid Monte Carlo (HMC)  

\[ \langle \mathcal{O} \rangle = \frac{1}{Z} \int D[U] \mathcal{O}[U] e^{-S_{\text{eff}}[U]} \approx \frac{1}{N} \sum_m \mathcal{O}[U_m] \]

- Propability \( \rho \sim e^{-S_{\text{eff}}[U]} \), \( S_{\text{eff}}[U] \sim S_{\text{gauge}} + \ln \det D \)
- Fermion determinant: \( \det D \sim \phi^\dagger D^{-1} \phi \)
- Sign-Problem of Lattice QCD:
  \[ \det(D + \mu)^* = \det(D - \mu^*) \]
  \( \Rightarrow \) Importance sampling ill-defined at finite, real \( \mu \)
Inversion of the Fermion Matrix

\[ D \psi = \phi \iff \psi = D^{-1} \phi \]

- Dominant part of typical LQCD simulation (for small \( m_{\text{quark}} \))
- Use Krylov-subspace based solver (iterative)

Most expensive ingredient: Derivative part \( \partial \phi \):

- Wilson: read/write: 2880 Bytes
- \( \iff \) perform 1632 FLOP
- \( \rightarrow \) Low numerical density (\( \sim 0.5 \) FLOP/Byte)
- \( \Rightarrow \) LQCD always memory bandwidth limited
Lattice QCD at Finite Temperature

Lattice Volume: \( V = N_\sigma^3 \times N_\tau \)  
\( N_\sigma \): spatial extent, \( N_\tau \): temporal extent

Finite Temperature on the lattice:

\[
T = \frac{1}{a(\beta) N_\tau} \quad \text{and} \quad N_\tau \ll N_\sigma
\]

\( \left( \text{Compare} \ T = 0 : N_\tau \gg N_\sigma \right) \)

Simulation strategy

- Find critical temperature  
  \( \Rightarrow \) Need multiple \( \beta \) values (Temperature scan)

- Check volume scaling (Phase transitions \( \sim \) non-analyticities)  
  \( \Rightarrow \) Need multiple spatial volumes \( (N_\sigma) \) (Finite Size Analysis)

\[ \Rightarrow \]

- Small/Moderate problem sizes
- Studies inherently parallel (trivially)
Motivation

LQCD

CL$^2$QCD

Performance

Summary & Perspectives

Graphics Processing Units (GPUs)

LQCD always memory bandwidth limited $\Rightarrow$ GPUs well-suited

Performance

<table>
<thead>
<tr>
<th>CHIP</th>
<th>Peak SP {GFLOPS}</th>
<th>Peak DP {GFLOPS}</th>
<th>Peak BW {GB/s}</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD Radeon HD 5870 Cypress</td>
<td>2720</td>
<td>544</td>
<td>154</td>
</tr>
<tr>
<td>AMD Radeon HD 7970 Tahiti</td>
<td>3789</td>
<td>947</td>
<td>264</td>
</tr>
<tr>
<td>AMD FirePro S10000 Tahiti</td>
<td>2×3410</td>
<td>2×850</td>
<td>2×240</td>
</tr>
<tr>
<td>NVIDIA GeForce GTX 680 Kepler</td>
<td>3090</td>
<td>258</td>
<td>192</td>
</tr>
<tr>
<td>NVIDIA Tesla K40 Kepler</td>
<td>4290</td>
<td>1430</td>
<td>288</td>
</tr>
<tr>
<td>AMD Opteron 6172 Magny-Cours</td>
<td>202</td>
<td>101</td>
<td>43</td>
</tr>
<tr>
<td>Intel Xeon E5-2690 Sandy Bridge EP</td>
<td>371</td>
<td>186</td>
<td>51</td>
</tr>
</tbody>
</table>

LQCD & GPUs

Virtually all applications based on vendor-specific NVIDIA’s CUDA
See the QUDA library: https://github.com/lattice/quda
GPU clusters at Frankfurt

<table>
<thead>
<tr>
<th></th>
<th>LOEWE-CSC</th>
<th>SANAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU nodes</td>
<td>786</td>
<td>304</td>
</tr>
<tr>
<td>GPUs/node</td>
<td>$1 \times$ AMD 5870</td>
<td>$2 \times$ AMD S10000</td>
</tr>
<tr>
<td>CPUs/node</td>
<td>$2 \times$ Opteron 6172</td>
<td>$2 \times$ Xeon E5-2650</td>
</tr>
</tbody>
</table>

Future: LOEWE-CSC-Update & new LQCD-Cluster at GSI

Alternative to CUDA: OpenCL

https://www.khronos.org/opencl/

Open standard for heterogeneous computing platforms

⇒ GPUs and CPUs can be used together within same framework
CL²QCD (I)

http://code.compeng.uni-frankfurt.de/projects/clhmc

- New LQCD code based on OpenCL
- Successfully applied in physics studies on GPU clusters
  Loewe-CSC & Sanam

Features

- First OpenCL application for Wilson fermions focusing on Twisted Mass Wilson fermions Frezzotti & Rossi 2003
- Staggered fermions in standard formulation Kogut & Susskind 1975
- Improved gauge actions
- Standard inversion and integration algorithms
- ILDG-compatible IO
- RANLUX Pseudo-Random Number Generator (PRNG) Lüscher 1994
**CL$^2$QCD (II)**

**Executables**

- **HMC**: Generation of gauge field configurations for $N_f = 2$ (Twisted Mass) Wilson type fermions using HMC algorithm;
- **RHMC**: Generation of gauge field configurations for staggered type fermions using Rational HMC algorithm Clark & Kennedy (2007)
- **SU3HEATBATH**: Gen. of gauge field conf. for $SU(3)$ Pure Gauge Theory using heatbath algorithm e.g. Cabibbo & Marinari (1982)
- **INVERTER/GAUGE OBSERVABLES**: Measurements of fermionic/gauge observables on given gauge field conf.
Unit Tests

OpenCL hardware/platform independent

Computing architecture known at runtime only
⇒ Kernel compilation at runtime

- Varying computing platforms ⇒ Need regression tests
- LQCD fcts. local ⇒ well testable
CL$^2$QCD - Code Structure (I)

- **Host program** of CL$^2$QCD set up in C++
  - \( \Rightarrow \) Independent program parts & extension capabilities

- Cross-platform compilation using **CMake** http://www.cmake.org

- Two main components:
  - **physics** package: High-level functionality
  - **hardware** package: Low-level functionality
  - **meta** package: Control of program execution and I/O

- **All parts** of simulation code carried out using OpenCL kernels (double precision)

- Kernels in a certain way detached from host part (host can continue independently of kernel execution status)
  - \( \Rightarrow \) Clear separation into administrative part (host) and performance-critical calculations (kernels)

- **OpenCL kernels source files:**
  - OpenCL language based on C99
  - Compilation and execution handled within **hardware** package
CL$^2$QCD - Code Structure (II)
Memory access

Different memory access patterns:

*Array of Structures (AoS) ↔ Structure of Arrays (SoA)*

Example: Fermion field $\psi$:

<table>
<thead>
<tr>
<th>AoS</th>
<th>$\psi(0).e0$</th>
<th>$\psi(0).e1$</th>
<th>$\psi(0).e2$</th>
<th>$\psi(0).e11$</th>
<th>$\psi(0).e12$</th>
<th>$\psi(1).e0$</th>
<th>$\psi(2).e0$</th>
<th>$\psi(N).e0$</th>
<th>$\psi(0).e1$</th>
<th>$\psi(1).e1$</th>
<th>$\psi(2).e1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoA</td>
<td>$\psi(0).e0$</td>
<td>$\psi(1).e0$</td>
<td>$\psi(2).e0$</td>
<td>$\psi(0).e1$</td>
<td>$\psi(1).e1$</td>
<td>$\psi(2).e1$</td>
<td>$\psi(0).e1$</td>
<td>$\psi(1).e1$</td>
<td>$\psi(2).e1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\psi$: 12-comp. complex vector

GPU: SoA preferable for optimal memory access
Motivation

QCD Performance Summary & Perspectives

Wilson Performance

<table>
<thead>
<tr>
<th>Lattice Size</th>
<th>GB/s AMD Radeon HD 7970</th>
<th>GB/s AMD Radeon HD 5870</th>
<th>GFLOPS NVIDIA Tesla K40</th>
<th>GB/s AMD FirePro S10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>16³ × 8</td>
<td>16³ × 16</td>
<td>16³ × 24</td>
<td>16³ × 32</td>
<td>16³ × 16</td>
</tr>
<tr>
<td>24³ × 12</td>
<td>24³ × 16</td>
<td>24³ × 24</td>
<td>24³ × 32</td>
<td>24³ × 16</td>
</tr>
<tr>
<td>32³ × 8</td>
<td>32³ × 12</td>
<td>32³ × 16</td>
<td>32³ × 24</td>
<td>32³ × 12</td>
</tr>
<tr>
<td>24³ × 48</td>
<td>24³ × 32</td>
<td>32³ × 24</td>
<td>32³ × 16</td>
<td>24³ × 48</td>
</tr>
<tr>
<td>48³ × 8</td>
<td>48³ × 16</td>
<td>48³ × 24</td>
<td>48³ × 32</td>
<td>48³ × 16</td>
</tr>
</tbody>
</table>

Peak Performances:

- HD 7970: 264 GB/s
- HD 5870: 154 GB/s
- S10000: 240 GB/s
- K40: 288 GB/s

- AMD GPUs: ~ 75% of peak BW
- NVIDIA: ~ 60% (No spec. tuning)
HMC Performance

Speedup, compared to 2 LOEWE-CSC CPUs (= 1 node)

- AMD Radeon HD 5870 (LOEWE-CSC): \(\sim 2\)
- AMD FirePro S10000 (SANAM): \(\sim 4\)
Scaling of the CG solver on SANAM (4 GPUs per node):

- **Strong scaling:**
  - Time direction splittable (within node)
  - Nice scaling for large enough lattices
  - Fin. $T$ studies: Spatial splitting preferable

- **Weak scaling:**
Motivation LQCD CL²QCD Performance Summary & Perspectives

$D_{\text{staggered}}$ Performance

Peak Performances:
- HD 7970: 264 GB/s
- HD 5870: 154 GB/s
- K40: 288 GB/s

- Staggered working set much smaller than Wilson
- No optimizations carried out (yet)
- AMD HD 5870: \sim 75\% of peak BW
- Newer hardware requires more investigation
Summary

- New LQCD software $\text{CL}^2\text{QCD}$ based on OpenCL
- Shows very good performance
- Available at:

  http://code.compeng.uni-frankfurt.de/projects/clhmc

- Successfully applied in physics studies at finite temperature

Perspectives

- Implementation of new features as required by physics studies
- Tuning for specific hardware (esp. staggered code)