BQCD – Berlin quantum chromodynamics program

Yoshifumi Nakamura
Institut für Theoretische Physik, Universität Regensburg, 93040 Regensburg, Germany
Center for Computational Sciences, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan
E-mail: yoshi@ccs.tsukuba.ac.jp

Hinnerk Stüben*
Konrad-Zuse-Zentrum für Informationstechnik Berlin, 14195 Berlin, Germany
E-mail: stueben@zib.de

We publish BQCD as free software under the GNU General Public License. BQCD is a Hybrid Monte-Carlo program that simulates lattice QCD with dynamical Wilson fermions. It is one of the main production programs of the QCDSF collaboration. The program can simulate 2 and $2+1$ fermion flavours with pure, clover improved, and stout smeared fat link Wilson fermions as well as standard plaquette, and an improved (rectangle) gauge action. The single flavour is simulated with the Rational Hybrid Monte-Carlo algorithm.

The XXVIII International Symposium on Lattice Field Theory, Lattice2010
June 14-19, 2010
Villasimius, Italy

---

*Speaker.

1. Introduction

Berlin quantum chromodynamics program (BQCD) is a Hybrid Monte-Carlo [1] program for simulating lattice QCD with dynamical Wilson fermions. The development of BQCD started in 1998 for the two flavour case and the standard Wilson action. It was written for a study of parallel tempering [2]. At that time the whole parallelisation framework was completed. Soon the program was extended in two different directions. The first direction was the implementation of clover $O(a)$ improvement of the fermion action. With the availability of clover improvement BQCD became one of the main production codes of the QCDSF collaboration [3]. The second direction was the addition of an external field to the standard Wilson action in order to study the Aoki phase [4]. The next milestone was the implementation of the Hasenbusch trick [5, 6]. Starting in 2006 the code has been largely extended to enable simulations including a third fermion flavour [7, 8, 9, 10]. This extension includes the implementation of Rational Hybrid Monte-Carlo (RHMC) [11] for the simulation of the third quark flavour as well as many algorithmic and performance improvements.

The code is also being used by the DIK Collaboration for simulations at finite temperature [12, 13]. Several people took BQCD as a starting point for adding their own code for measurements. The plan of the QPACE project [14] to port BQCD, in particular the fermion matrix multiplication and solvers [15], to their machine has triggered the publication of the code as free software under the *GNU General Public License* on the occasion of this Lattice conference. The source and a manual can be downloaded from [16]. A description on building and testing binaries can be found in the manual.

2. Actions

The program can simulate the QCD with the following actions. The gauge action can be the Wilson action

$$S = S_G^{\text{Wilson}} = \sum_{\text{plaquette}} \frac{1}{3} \operatorname{Re} \operatorname{Tr} \left(1 - U_{\text{plaquette}}\right) \tag{2.1}$$

or a Symanzik improved gauge action

$$S_G = \frac{6}{g^2} \left[c_0 \sum_{\text{plaquette}} \frac{1}{3} \operatorname{Re} \operatorname{Tr} \left(1 - U_{\text{plaquette}}\right) + c_1 \sum_{\text{rectangle}} \frac{1}{3} \operatorname{Re} \operatorname{Tr} \left(1 - U_{\text{rectangle}}\right) \right], \tag{2.2}$$

with $c_0 + 8c_1 = 1$. The fermion action can be the Wilson action

$$S_F^{\text{Wilson}} = \sum_x \left\{ \bar{\psi}(x)\psi(x) - \kappa \left[\bar{\psi}(x)U_\mu^\dagger(x-\hat{\mu})(1+\gamma_\mu)\psi(x-\hat{\mu}) + \bar{\psi}(x)U_\mu(x)(1-\gamma_\mu)\psi(x+\hat{\mu}) \right] \right\}, \tag{2.3}$$

the Wilson action plus an explicitly parity-flavour symmetry breaking source term, where τ^3 is the third Pauli matrix

$$S_F = S_F^{\text{Wilson}} + h \sum_x \bar{\psi}(x) i \gamma_5 \tau^3 \psi(x), \tag{2.4}$$

the clover action

$$S_F = S_F^{\text{Wilson}} - \frac{i}{2} \kappa c_{\text{SW}} \sum_x \bar{\psi}(x) \sigma_{\mu\nu} F_{\mu\nu}(x) \psi(x), \tag{2.5}$$

the clover action plus a CP breaking term

$$S_F = S_F^{\text{Wilson}} - \frac{i}{2}\kappa c_{\text{SW}} \sum_x \bar{\psi}(x)\sigma_{\mu\nu}F_{\mu\nu}(x)\psi(x) + \theta\,\bar{\psi}(x)\gamma_5\psi(x) \tag{2.6}$$

or a stout smeared fat link action (any term containing gauge links can be smeared), in particular the SLiNC fermion action [9]

$$S_F = \sum_x \Big\{ \bar{\psi}(x)\psi(x) - \kappa\,\bar{\psi}(x)U_\mu^\dagger(x-\hat{\mu})[1+\gamma_\mu]\psi(x-\hat{\mu})$$
$$- \kappa\,\bar{\psi}(x)U_\mu(x)[1-\gamma_\mu]\psi(x+\hat{\mu}) + \frac{i}{2}\kappa c_{\text{SW}}\,\bar{\psi}(x)\sigma_{\mu\nu}F_{\mu\nu}(x)\psi(x)\Big\}, \tag{2.7}$$

where the gauge links U_μ are replaced by stout links [17]

$$U_\mu \to \tilde{U}_\mu(x) = e^{iQ_\mu(x)}U_\mu(x), \tag{2.8}$$

with

$$Q_\mu(x) = \frac{\alpha}{2i}\left[V_\mu(x)U_\mu^\dagger(x) - U_\mu(x)V_\mu^\dagger(x) - \frac{1}{3}\text{Tr}\left(V_\mu(x)U_\mu^\dagger(x) - U_\mu(x)V_\mu^\dagger(x)\right)\right], \tag{2.9}$$

where $V_\mu(x)$ is the sum over all staples associated with the link. Boundary condition for the gauge field are periodic in all directions. For the fermions boundary conditions can be chosen to be anti-periodic or periodic for each dimension.

3. Observables

The following gluonic observables can be measured: the average plaquette and average rectangular plaquette, the topological charge (the topological charge is measured with the field theoretic method after cooling the gauge field configuration), the Polyakov loop. In addition some fermionic bulk quantities can be measured (from stochastic estimators):

$$\langle\bar{\psi}\psi\rangle = \frac{1}{12V}\langle\text{Tr}(M^{-1})\rangle \qquad \text{('chiral condensate')}$$
$$\langle\bar{\psi}\gamma_5\psi\rangle = \frac{1}{12V}\langle\text{Tr}(\gamma_5 M^{-1})\rangle$$
$$\langle\Pi^2\rangle = \frac{1}{12V}\langle\text{Tr}(M^\dagger M)^{-1}\rangle \qquad \text{('pion norm')}$$

4. Algorithmic improvements

4.1 Integrators

HMC trajectories can be integrated with leapfrog or Omelyan [18] integrators. Multi timescale integration is possible with up to six time scales. In the following we explain a multi timescale setup that is used in production for $N_f = 2+1$ improved Wilson fermions. Starting point is the partition function

$$Z = \int DU D\bar{\psi}D\psi\, e^{-S} \tag{4.1}$$
$$S = S_g(\beta) + S_l(\kappa_l, c_{\text{SW}}) + S_s(\kappa_s, c_{\text{SW}}) \tag{4.2}$$

3

where S_g is a gluon action, S_l is an action for the degenerate *u*- and *d*-quarks and S_s is an action for the strange quark. After integrating out fermions

$$S = S_g(\beta) - \ln[\det M_l^\dagger M_l][\det M_s^\dagger M_s]^{\frac{1}{2}}. \tag{4.3}$$

First even-odd preconditioning is applied

$$\det M_l^\dagger M_l \propto \det(1 + T_{oo}^l)^2 \det Q_l^\dagger Q_l \tag{4.4}$$

$$[\det M_s^\dagger M_s]^{\frac{1}{2}} \propto \det(1 + T_{oo}^s)[\det Q_s^\dagger Q_s]^{\frac{1}{2}} \tag{4.5}$$

where

$$Q = (1 + T)_{ee} - M_{eo}(1 + T)_{oo}^{-1} M_{oe} \tag{4.6}$$

$$T = \frac{i}{2} c_{SW} \kappa \sigma_{\mu\nu} F_{\mu\nu}. \tag{4.7}$$

Then $\det Q_l^\dagger Q_l$ is separated following Hasenbusch [5]

$$\det Q_l^\dagger Q_l = \det W_l^\dagger W_l \det \frac{Q_l^\dagger Q_l}{W_l W_l^\dagger}, \qquad W = Q + \rho. \tag{4.8}$$

Finally the standard action is modified to

$$S = S_g + S_{\det}^l + S_{\det}^s + S_{f1}^l + S_{f2}^l + S_{fr}^s, \tag{4.9}$$

where

$$S_{\det}^l = -2 \operatorname{Tr} \log[1 + T_{oo}(\kappa^l)] \tag{4.10}$$

$$S_{\det}^s = - \operatorname{Tr} \log[1 + T_{oo}(\kappa^s)] \tag{4.11}$$

$$S_{f1}^l = \phi_1^\dagger [W(\kappa^l)^\dagger W(\kappa^l)]^{-1} \phi_1 \tag{4.12}$$

$$S_{f2}^l = \phi_2^\dagger W(\kappa^l) [Q(\kappa^l)^\dagger Q(\kappa^l)]^{-1} W(\kappa^l)^\dagger \phi_2 \tag{4.13}$$

$$S_{fr}^s = \sum_{i=1}^n \phi_{2+i}^\dagger [Q(\kappa^s)^\dagger Q(\kappa^s)]^{-\frac{1}{2n}} \phi_{2+i} \tag{4.14}$$

We calculate S_{fr} using the RHMC algorithm [11] with optimised values for *n* and the number of fractions. Each term of the action is split into one ultraviolet and two infrared parts,

$$S_{UV} = S_g \tag{4.15}$$

$$S_{IR-1} = S_{\det}^l + S_{\det}^s + S_{f1}^l \tag{4.16}$$

$$S_{IR-2} = S_{f2}^l + S_{fr}^s. \tag{4.17}$$

In the leap-frog integrator S_{UV}, S_{IR-1} and S_{IR-2} are put on *three separate* time scales,

$$V(\tau) = \left[V_{IR-2}\left(\frac{\delta\tau}{2}\right) A^{m_1} V_{IR-2}\left(\frac{\delta\tau}{2}\right) \right]^{n_\tau} \tag{4.18}$$

$$A = V_{IR-1}\left(\frac{\delta\tau}{2m_1}\right) B^{m_2} V_{IR-1}\left(\frac{\delta\tau}{2m_1}\right) \tag{4.19}$$

$$B = V_{UV}\left(\frac{\delta\tau}{2m_1 m_2}\right) V_Q\left(\frac{\delta\tau}{m_1 m_2}\right) V_{UV}\left(\frac{\delta\tau}{2m_1 m_2}\right) \tag{4.20}$$

where $n_\tau = \tau/(\delta\tau)$ and the *V*s are evolution operators of the Hamiltonian.

4.2 Solvers

Besides the standard conjugate gradient *(cg)* solver BiCGstab and GMRES were implemented. Variants with mixed precision arithmetics are available for *cg* and BiCGstab. In order to reduce time spent in the solver chronological inversion [19] is employed and even-odd preconditioning as well as Schwarz preconditioning [20] are used.

5. Implementation details

The code is mostly written Fortran. The C preprocessor is used for preprocessing in general and the *m4* macro processor for a few files. A simple mechanism is employed to automatically generate multi precision versions from the same source. BQCD is parallelised with MPI and OpenMP. The first version of the program was parallelised for a Cray T3E with the *shmem* library. *shmem* can still be used in the hopping matrix multiplication.

Random numbers are generated with *ranlux* [21, 22]. Binary data (SU(3) configurations) can either be stored in a native BQCD format or in the International Lattice DataGrid (ILDG) [23] format. The input parameter file and the log file are simple text files that have a *keyword value(s)* structure. Important parts of the program are instrumented for time profiling and performance measurements.

		hopping matrix multiplication (Fortran)			*cg* solver		
		per core	overall	fraction	per core	overall	fraction
#racks	#cores	Mflop/s	Tflop/s	of peak	Mflop/s	Tflop/s	of peak
1/2	2048	344	0.70	10.1 %	385	0.79	11.3 %
1	4096	429	1.76	12.6 %	461	1.89	13.6 %
2	8192	415	3.40	12.2 %	444	3.64	13.1 %
4	16384	407	6.67	12.0 %	423	6.93	12.4 %

Table 1: Performance figures for a $48^3 \times 96$ lattice obtained with the pure Fortran implementation on a Blue Gene/P.

		hopping matrix multiplication (assembler)			*cg* solver		
		per core	overall	fraction	per core	overall	fraction
#racks	#cores	Mflop/s	Tflop/s	of peak	Mflop/s	Tflop/s	of peak
1/2	2048	1057	2.16	31.1 %	821	1.68	24.1 %
1	4096	1061	4.35	31.2 %	802	3.28	23.6 %
2	8192	1019	8.35	30.0 %	763	6.25	22.5 %
4	16384	923	15.11	27.1 %	684	11.21	20.1 %

Table 2: Performance figures for a $48^3 \times 96$ lattice obtained with an assembler implementation of the hopping matrix multiplication on a Blue Gene/P.
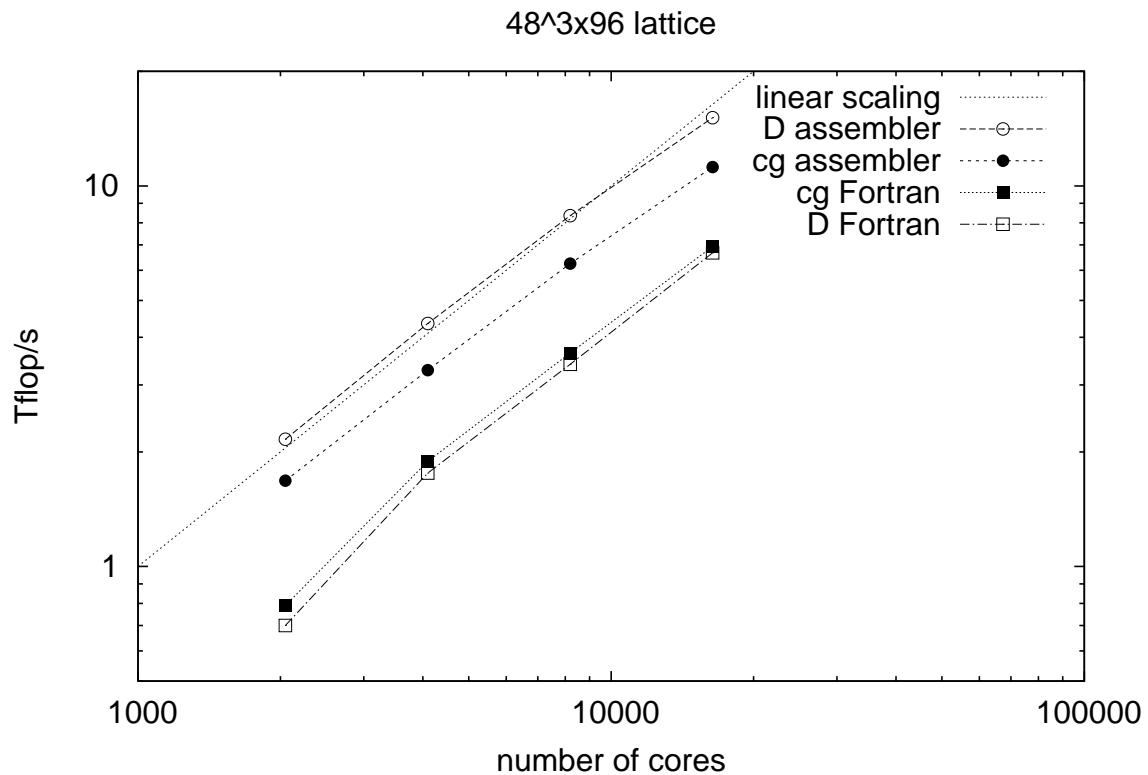
48^3x96 lattice

Figure 1: Scaling plot of performance data from a Blue Gene/P given in Tables 1 and 2. The dotted line indicates linear scaling. Any linear scaling runs parallel to this line.

6. Performance

The program scales very well to large numbers of cores. The pure Fortran case even displays some super-linear speedup (see Figure 1). For Blue Gene and Itanium2 assembler implementations of the hopping matrix multiplication were provided by Th. Streuer. Performance figures for a $48^3 \times 96$ lattice obtained on a Blue Gene/P are given in Tables 1 and 2. The assembler implementations makes it possible to overlap communication with computation. This boosts the performance of the hopping multiplication of up to a factor of 3.1 and the whole conjugate gradient solver by a factor of 1.6 to 2.1 compared with the pure Fortran version. With this code it is possible to run simulations at a sustained overall speed of 11.2 Tflop/s

7. Acknowledgements

We would like to thank Gerrit Schierholz, Roger Horsley, Dirk Pleiter, Paul Rakow and James Zanotti for support, stimulating discussions and bug reports, Thomas Streuer for providing assembler code and Andrea Nobile for discussions on Schwarz preconditioning. The computations were performed on the Blue Gene/P at Jülich Supercomputer Centre, Jülich, Germany.

References

[1] S. Duane, A. D. Kennedy, B. J. Pendleton and D. Roweth, Phys. Lett. B **195** (1987) 216.

[2] E. M. Ilgenfritz, W. Kerler, M. Müller-Preussker and H. Stüben, Phys. Rev. D **65** (2002) 094506 [arXiv:hep-lat/0111038].

[3] H. Stüben [QCDSF-UKQCD Collaboration], Nucl. Phys. Proc. Suppl. **94** (2001) 273, [arXiv:hep-lat/0011045].

[4] E. M. Ilgenfritz, W. Kerler, M. Müller-Preussker, A. Sternbeck and H. Stüben, Phys. Rev. D **69** (2004) 074511, [arXiv:hep-lat/0309057].

[5] M. Hasenbusch, Phys. Lett. B **519** (2001) 177 [arXiv:hep-lat/0107019].

[6] A. Ali Khan, T. Bakeyev, M. Göckeler, R. Horsley, D. Pleiter, P.E.L. Rakow, A. Schäfer, G. Schierholz, H. Stüben [QCDSF Collaboration], Nucl. Phys. Proc. Suppl. **129** (2004) 853 [arXiv:hep-lat/0309078].

[7] M. Göckeler *et al.* [QCDSF Collaboration], PoS **LAT2007** (2007) 041 [arXiv:0712.3525 [hep-lat]].

[8] N. Cundy *et al.* [QCDSF-UKQCD Collaborations], PoS **LAT2008** (2008) 132 [arXiv:0811.2355 [hep-lat]].

[9] N. Cundy *et al.*, Phys. Rev. D **79** (2009) 094507 [arXiv:0901.3302 [hep-lat]].

[10] W. Bietenholz *et al.* [QCDSF-UKQCD Collaborations], PoS **LAT2009** (2009) 102 [arXiv:0910.2963 [hep-lat]].

[11] M. A. Clark and A. D. Kennedy, Nucl. Phys. Proc. Suppl. **129** (2004) 850 [arXiv:hep-lat/0309084].

[12] Y. Nakamura *et al.*, AIP Conf. Proc. **756** (2005) 242 [Nucl. Phys. Proc. Suppl. **140** (2005) 535] [arXiv:hep-lat/0409153].

[13] V. G. Bornyakov *et al.*, arXiv:0910.2392 [hep-lat].

[14] H. Baier *et al.*, arXiv:0911.2174 [hep-lat].

[15] A. Nobile, PoS(LATTICE 2010)034.

[16] http://www.zib.de/stueben/bqcd

[17] C. Morningstar and M. J. Peardon, Phys. Rev. D **69** (2004) 054501 [arXiv:hep-lat/0311018].

[18] T. Takaishi and P. de Forcrand, Phys. Rev. E **73** (2006) 036706 [arXiv:hep-lat/0505020].

[19] R. C. Brower, T. Ivanenko, A. R. Levi and K. N. Orginos, Nucl. Phys. B **484** (1997) 353 [arXiv:hep-lat/9509012].

[20] M. Lüscher, Comput. Phys. Commun. **165** (2005) 199 [arXiv:hep-lat/0409106].

[21] M. Lüscher, Comput. Phys. Commun. **79** (1994) 100 [arXiv:hep-lat/9309020].

[22] http://luscher.web.cern.ch/luscher/ranlux

[23] M. G. Beckett, B. Joo, C. M. Maynard, D. Pleiter, O. Tatebe and T. Yoshie, arXiv:0910.1692 [hep-lat].