PROCEEDINGS OF SCIENCE

# Accelerating Twisted Mass LQCD with QPhiX

**Mario Schröck**[*]
*INFN - Sezione Roma Tre, Rome (Italy)*
*E-mail:* mario.schroeck@roma3.infn.it

**Silvano Simula**
*INFN - Sezione Roma Tre, Rome (Italy)*
*E-mail:* silvano.simula@roma3.infn.it

**Alexei Strelchenko**
*Fermi National Accelerator Laboratory*
*E-mail:* astrel@fnal.gov

We present the implementation of twisted mass fermion operators for the QPhiX library.[†] We analyze the performance on the Intel Xeon Phi (Knights Corner) coprocessor as well as on Intel Xeon Haswell CPUs. In particular, we demonstrate that on the Xeon Phi 7120P the Dslash kernel is able to reach 80% of the theoretical peak bandwidth, while on a Xeon Haswell E5-2630 CPU our generated code for the Dslash operator with AVX2 instructions outperforms the corresponding implementation in the tmLQCD library by a factor of $\sim 5\times$ in single precision. We strong scale the code up to 6.8 (14.1) Tflops in single (half) precision on 64 Xeon Haswell CPUs.

*The 33rd International Symposium on Lattice Field Theory*
*14–18 July 2015*
*Kobe International Conference Center, Kobe, Japan*

[*] Speaker.

[†] https://github.com/JeffersonLab/qphix

PoS(LATTICE 2015)030

## 1. Introduction

The *QPhiX* library [1] offers a collection of highly optimized kernels and inverters to perform Lattice QCD computations on recent Intel x86 systems. QPhiX is a C++ library and consists of two components: the lower level component of the library provides with the code generator which abstracts away vector intrinsics (supporting IMCI, AVX, AVX2, AVX512, SSE and QPX instructions). The high level part is concerned with parallelizing over threads via OpenMP and multi-processing via MPI; it is also performing the loop structure for the cache-blocking strategy. Currently, the library supports Wilson (Clover) fermions and (mixed precicion) CG, BiCGstab inverters. There is also an implementation of the staggered Dslash operator [2]. For details we refer to the original work [1]. In the present study we extend QPhiX for initial support of degenerate twisted mass fermions [3, 4].

## 2. QPhiX and twisted mass fermions

The degenerate twisted mass Dslash operator $\not{D}_{\mathrm{TM}}$ is defined as

$$\not{D}_{\mathrm{TM}} = \not{D}_{\mathrm{W}} 1_f + i\mu\gamma_5\tau_3 \,, \tag{2.1}$$

where $\not{D}_{\mathrm{W}}$ is the standard Wilson Dslash operator which is diagonal in flavour space. The second term is the *twisted mass term* which is nontrivial in flavour space ($\tau_3$ being the third Pauli matrix in flavour space), $\mu$ is the twisted mass.

QPhiX relies on even-odd preconditioning and therefore the operator (2.1) requires two kernels (plus the corresponding daggered versions) which act on the even (odd) sites separately, in QPhiX parlance, *Dslash* and *AChiMBDPsi*:

1. *Dslash*:
$$\chi = R^{-1}\, \not{D}_{\mathrm{W}} \psi \tag{2.2}$$
2. *AChiMBDPsi*:
$$\phi = R\chi - b\, \not{D}_{\mathrm{W}} \psi \tag{2.3}$$
with $R = 1 + i2\kappa\mu\gamma_5$ being the twist operator.

The successive application of these kernels corresponds to the Schur decomposed fermion matrix

$$\tilde{M}_{oo} = R_{oo} - \frac{1}{2}\kappa(\not{D}_{\mathrm{W}})_{oe} R_{ee}^{-1}\, (\not{D}_{\mathrm{W}})_{eo} \tag{2.4}$$

which then enters, e.g., the Conjugate Gradient algorithm. We implement these kernels on the level of the code-generator. The high-level part has to be modified to include the twisted mass parameter $\mu$ and to call the corresponding twisted mass low-level kernels, apart from that it is equivalent to the pure Wilson case.

The QPhiX routine `dslash_plain_body()` generates the (Wilson) Dslash kernel routines, within that, after the call of `dslash_body()`, we call the (inverse) twisted term. This generates files with vector intrinsics of the form, e.g.,

```
tmf_dslash_plus_body_float_float_v8_s4_12
```

for the different floating point precisions, SIMD vector lengths, structure of arrays lengths and gauge field compression types (12 or 18). Next, the generated files are wrapped up as template specializations of the form

```
template<>
inline void
dslash_plus_vec<FPTYPE,VEC,SOA,COMPRESS12>(...)
{
    #include INCLUDE_FILE_VAR(qphix/avx2/generated/dslash_plus_body_,
        FPTYPE,VEC,SOA,COMPRESS_SUFFIX)
}
```

The procedure for the AChiMBDPsi kernels is done in a close analogy with the plain Dslash ones.

Lastly, we have to consider the unpacking routines for the MPI communication. The spinors are projected to halfspinors before being exchanged at the boundaries of each domain. These packing routines are identical to the Wilson case. However, when unpacking, while accumulating the different directions and before streaming to memory, we still have to apply the twisted mass term and therefore we need twisted mass specific unpacking routines.

## 3. Performance

### 3.1 Single node

In Fig. 1 we summarize the performance results of our twisted mass Dslash kernels as well as of the full Conjugate Gradient (CG) inversion algorithm on a single Intel Xeon-Phi 7120P. The results are distinguished by double (DP), single (SP) and half precision (HP) and moreover whether 12 compression[1] has been used or not (otherwise 18 “compression”). Note that the DP and SP 12 compression Dslash kernels reach around 80% of the theoretical peak bandwidth of the Xeon-Phi.

In Fig. 2 we present an equivalent performance plot but on a Dual Socket Xeon Haswell CPU (E5-2630 at 2.4GHz) with kernels that we produced with the QPhiX code generator for AVX2 instructions. The performance gives around 60% of the corresponding Xeon-Phi performance. With the tmLQCD package [5, 6], we reach 28 Gflops in double and 38 Gflops in single precision, respectively, for the twisted mass Dslash operation. (Note that tmLQCD supports neither 12 compression nor half precision arithmetics). Therefore our QPhiX kernels account for speedup factors of 3× in DP, 4.9× in SP and 6.7× in HP (the latter in relation to the tmLQCD SP result), respectively.

### 3.2 Weak scaling

Fig. 3 and Fig. 4 show the weak scaling behaviour of the Dslash kernel on up to 64 Xeon-Phi 7120P devices and up to 64 Dual socket Haswell E5-2630 CPUs, respectively. These tests have been performed on the “Galileo” cluster[2] at Cineca, Italy. On both architectures we compare the

[1] To overcome the bandwidth bottleneck to some extent, one may transfer only two of the three SU(3) matrix rows (thus 12 parameters instead of 18) and recalculate the third one on-the-fly.

[2] `http://www.hpc.cineca.it/hardware/galileo`

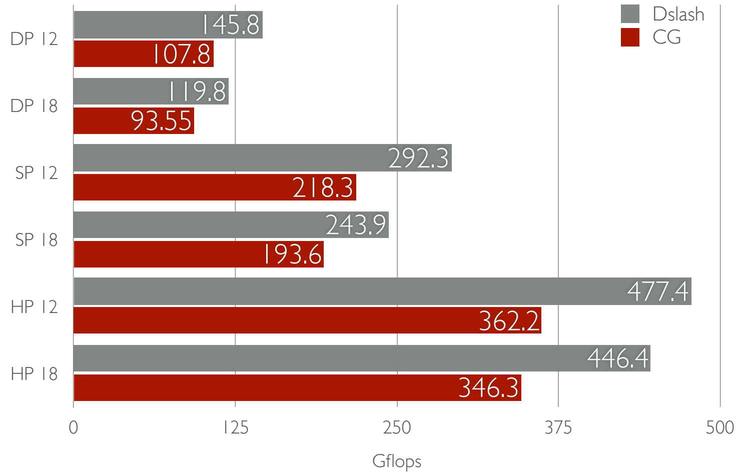

**Figure 1:** Twisted Mass $32^3 \times 64$ Xeon-Phi 7120P.

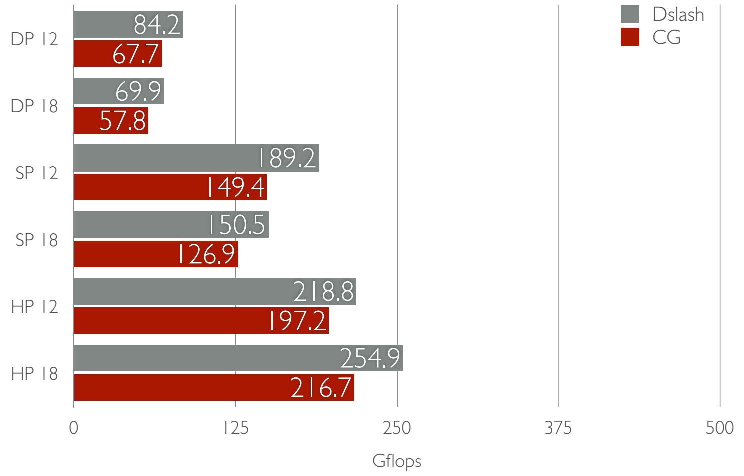

**Figure 2:** Twisted Mass $32^3 \times 64$ Dual Socket Xeon Haswell E5-2630 2.4GHz AVX2.

performance to a run with a proxy [1] that has been made available to us by Parallel Computing Lab, Intel Corporation. When running the code with the proxy, the latter will fully occupy one of the compute cores. Loosing one of the 16 cores of the Dual Socket Xeon Haswell is too costly to profit from optimized MPI communication (Fig. 4). On the Xeon-Phi (Fig. 3), on the other hand, dedicating one of the 61 cores for communication pays off and weak scaling is better than without proxy. The kernel reaches 18.2 Tflops on 64 devices.

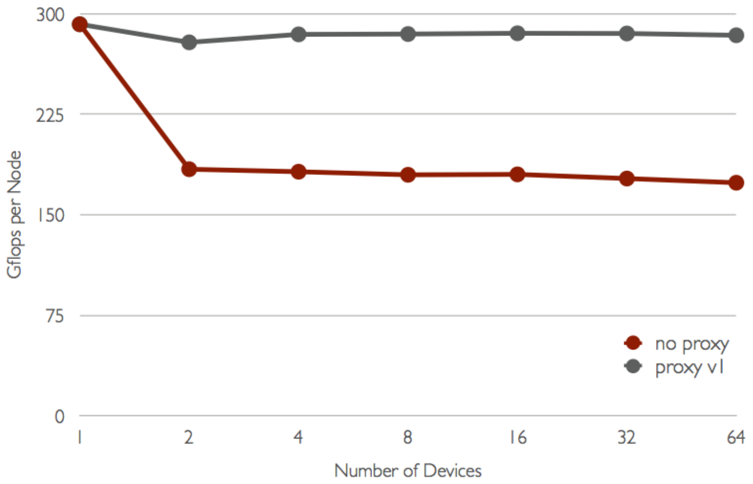

**Figure 3:** Weak Scaling Twisted Mass Dslash SP $48^3 \times 96$ per device Xeon-Phi 7120P.

### 3.3 Strong scaling

In Figs. 5 and 6 we present the strong scaling behaviour of the Dslash kernel on Dual Socket Xeon Haswell CPUs in single (SP) and half precision (HP), respectively, for a range of lattice sizes ($24^3 \times 48$, $32^3 \times 64$ and $48^3 \times 96$). While the strong scaling behaviour on the Xeon CPU is resonable, the Xeon-Phis require a too large local volume (compared to today's state of the art lattice sizes) to reach high performance. On the largest lattice we reach 6.8 (14.1) Tflops in single (half) precision on 64 nodes.

## 4. Summary

We have implemented *Dslash* and *AChiMBDPsi* kernels of the twisted mass fermion formulation for the QPhiX library. The code passes the unit tests of the kernels, the full fermion matrix (2.4) and the Conjugate Gradient algorithm. On the Xeon-Phi 7120P the Dslash kernel reaches 80% of the theoretical peak bandwidth and on a Dual Socket Xeon Haswell CPU with AVX2 generated code our QPhiX kernel outperforms the tmLQCD library by a factor of 4.9× in single precision (6.7× when making use of half precision arithmetics). Thanks to Intel's MPI proxy the weak scaling behaviour is good not only on the Xeon CPUs but also on Xeon-Phis. While strong scaling is good on the Xeon Haswell CPUs on our test machine, the Xeon-Phis require too large local volumes for practical purposes at this stage.

## Acknowledgments

We are very greatful to Bálint Joó, Dhiraj D. Kalamkar and Karthikeyan Vaidyanathan for sharing the QPhiX library and moreover for helpful discussions and assistance. Support by INFN and SUMA is acknowledged.

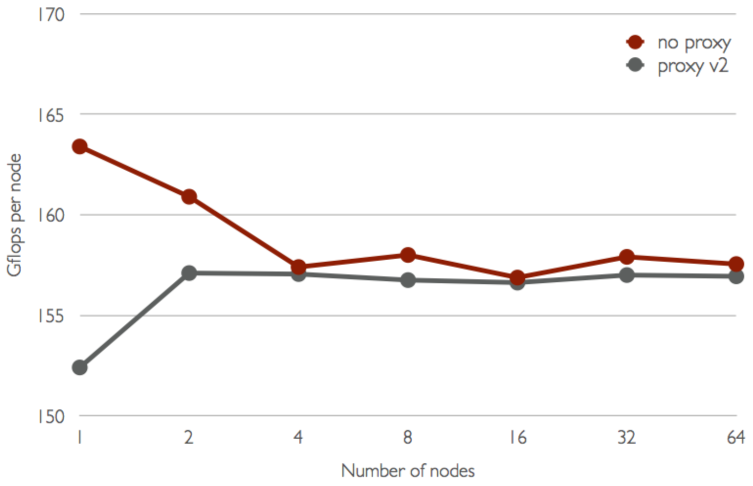

**Figure 4:** Weak Scaling Twisted Mass Dslash SP $32^3 \times 64$ per node Dual Socket Xeon Haswell E5-2630 2.4GHz AVX2.

## References

[1] B. Joó, D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnany, V. Lee, P. Dubey, and I. Watson, William, *Lattice QCD on Intel Xeon Phi Coprocessors*, in *Supercomputing* (J. Kunkel, T. Ludwig, and H. Meuer, eds.), vol. 7905 of *Lecture Notes in Computer Science*, pp. 40–54. Springer Berlin Heidelberg, 2013.

[2] R. Li and S. Gottlieb, *Staggered Dslash Performance on Intel Xeon Phi Architecture*, PoS LATTICE **2014**, 034 (2015) [arXiv:1411.2087 [hep-lat]].

[3] **ALPHA** Collaboration, R. Frezzotti, P. A. Grassi, S. Sint, and P. Weisz, *Lattice QCD with a chirally twisted mass term*, *JHEP* **08** (2001) 058, [hep-lat/0101001].

[4] R. Frezzotti and G. C. Rossi, *Twisted-mass lattice QCD with mass non-degenerate quarks*, *Nucl. Phys. Proc. Suppl.* **128** (2004) 193–202, [hep-lat/0311008].

[5] K. Jansen and C. Urbach, *tmLQCD: A Program suite to simulate Wilson Twisted mass Lattice QCD*, *Comput. Phys. Commun.* **180** (2009) 2717–2738, [arXiv:0905.3331].

[6] A. Abdel-Rehim, F. Burger, A. Deuzeman, K. Jansen, B. Kostrzewa, L. Scorzato, and C. Urbach, *Recent developments in the tmLQCD software suite*, *PoS* **LATTICE2013** (2014) 414, [arXiv:1311.5495].

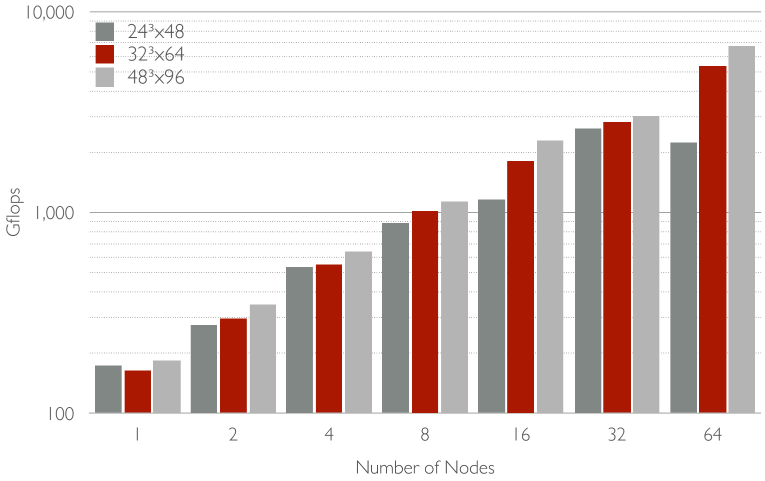

**Figure 5:** Strong Scaling Twisted Mass Dslash SP Compression12 Dual Socket Xeon Haswell E5-2630 2.4GHz AVX2.

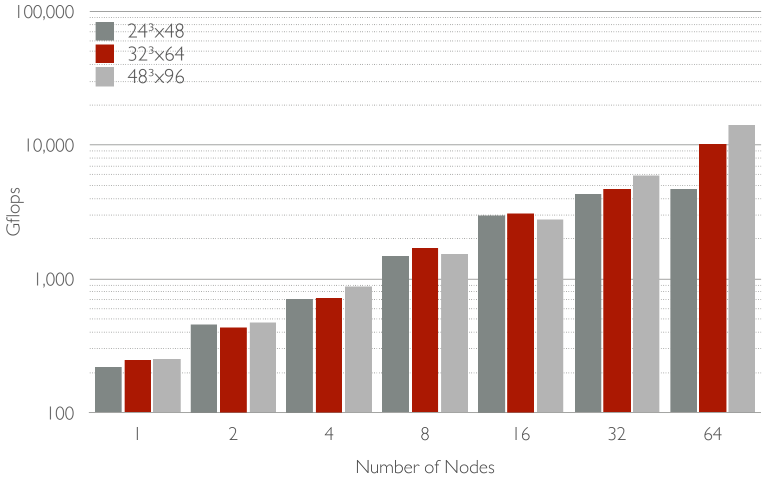

**Figure 6:** Strong Scaling Twisted Mass Dslash HP Compression18 Dual Socket Xeon Haswell E5-2630 2.4GHz AVX2.