Extending the QUDA library for Twisted Mass Fermions

Alexei Strelchenko
Fermi National Accelerator Laboratory
Batavia, Illinois, USA

1. Introduction

Lattice QCD (LQCD) uses large scale numerical simulations to study the strong interactions between quarks mediated by gluons (quantum chromodynamics, or QCD). Major improvements in the calculation and prediction of hadronic observables require large amounts of computer resources, of the order of hundreds of Tflop/s of sustained performance. The main objective of this project was to develop the necessary tools, so that the calculation of some key hadronic observables, which up to now where too demanding to be computed, will be made feasible. One example of this kind is disconnected diagrams, or fermion vacuum loops, that have typically been omitted from LQCD calculations due to their large computational cost, and the systematic uncertainties introduced by this omission is still an open issue. This computational intensive task could be harnessed, however, using GPUs which make it possible to investigate various numerical techniques in LQCD.

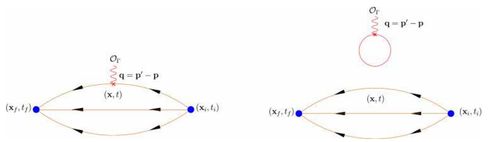

Figure 1: *Disconnected diagram contribution to the nucleon 3-point function.*

2. Discretization of the Dirac operator

Advanced LQCD calculations often require the evaluation of diagrams with disconnected quark lines. For this aim, one has to deal with inversions of very large sparse matrices that describe strong interactions that couples quarks with qluon to form hadrons (neutrons, protons, pions etc.). In the simplest case such a matrix is represented in the form (Wilson-Dirac matrix):

$$M_W = \kappa \sum_\mu \left[P^-_\mu u_\mu(x) \delta_{x,y+\hat{\mu}} + P^+_\mu u^\dagger_\mu(x-\hat{\mu}) \delta_{x,y-\hat{\mu}} \right] - \delta_{x,y}$$

where u_μ are 3x3 SU(3) gauge matrices, one link per space-time direction, and $P^\pm_\mu$ are 4x4 projection matrices in spinor subspace. This matrix can be also visualized by the following picture

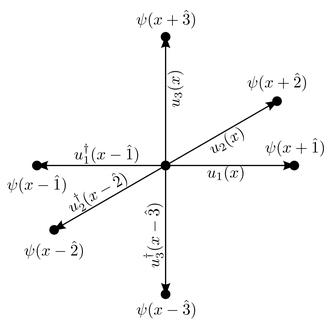

Figure 2: *Dslash operator.*

in which fermions are living in the sites of the lattice while gluon field is depicted as links connected the sites. Thus, the total matrix size is $N = (4 \times 3 \times V)$ where V is the lattice volume. Due to its local nature, the Wilson-Dirac matrix is suitable for computations on massively parallel architectures.

3. The QUDA library overview

QUDA is a software package for carrying out the time- consuming components of an LQCD application on NVIDIA CUDA platform. The QUDA library provides with optimized implementations of a number of different discretizations of the continuum QCD fermion operator as well as a range of iterative solvers for these fermion actions such as CGNE (for normalized equations), BiCGStab and recently the domain decomposition solver.
The main kernel operation of QUDA is sparse matrix-vector multiplication. The most essential optimizations are related to corresponding memory operations to diminish effects of relatively low arithmetic intensity of sparse matrix-vector multiplications on overall performance. These include several schemes:

- GPU memory coalescing data layout (using double2, float4 etc. built-in structures)
- Usage of the texture cache to reduce effects on non-coalescing access
- Data compression techniques exploiting the model symmetries
- Memory traffic reduction due to appropriate choice of the basis for projection matrices

To maximize performance gains for the iterative solvers on GPUs, QUDA exploits communication-avoiding algorithms and novel techniques such as using mixed precision (half-, single- and double precision):

- Improving performance with mixed precision solvers:
  - Use low precision (e.g., 32bit or 16bit) for bulk computations,
  - Use high precision (i.e., 64bit) for residual correction
- Improving scaling (with the domain decomposition solver by M.Luescher):

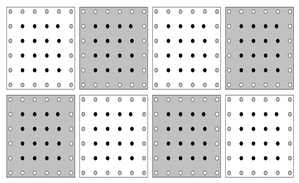

Figure 3: *Lattice domain decomposition.*

  - Separate contribution coming from interior of a sub-domain from that one coming from neighbor faces
  - Consider Wilson-Dirac operator on each sub-domain with Dirichlet BC
  - Use (locally) Schwartz procedure as a preconditioner combined with Krylov solver (as a global accelerator)
  - Most computational time spent on sub-domain → reduced communication overhead

The package is an open-source project, and freely availably from github.

4. Implementing Twisted Mass operator

The twisted mass formulation introduces an additional mass term to the Wilson fermion action with non-trivial isospin structure, so in the most general case the dslash operator can be written in the form:

$$M_{TM} = M_W 1_f + M_f, \quad M_f = i\mu\gamma_5\tau_3 + \epsilon\tau_1$$

where μ, ϵ are the twisted mass parameters, $\tau_{1,3}$ are the *Pauli* matrices.
The twisted mass QCD:

- allows simulations with 4 quarks (i.e., u, d plus heavy quark s, c)
- provides $O(a)$ improvement for hadron masses, form factors, decay constants
- provides protection against small eigenvalues $det[M_{TM}] = det[M_W^2 + \mu_{deg}^2 + ...]$
- while computational cost is comparable to the Wilson case

We extend the QUDA library to include kernels for non-degenerate twisted mass operator with optimizations specific for this type of operators. In particular, the straightforward approach in this case is to re-use the gauge field to avoid an extra memory transaction while computing contribution from each spinor flavor . Code development included:

- New GPU kernels with optimizations specific for the non-degenerate Twisted Mass operator
- Modifications in host interface to wrap matrix-vector multiplications in Krylov solvers

5. Code tests

The kernels we adapted are available for double, single and half precision in order to exploit a mixed precision technique, which allows one to obtain the solution in full double precision accuracy while using only single or half precision arithmetics for the bulk of the computation. Additionally, even- odd (or red-black) preconditioning is used according to the problem at hand. Therefore our work in including the non-degenerate twisted-mass fermion operator consisted of implementing it in all three arithmetic precisions and for even-odd ordering. Below we present strong scaling results for the CG solver using double/single and double/half precisions for the matrix-vector multiplication for $32^3 \times 64$ lattice.

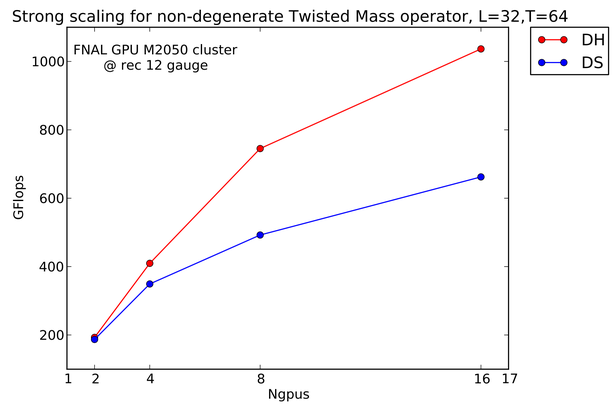

Figure 4: *Strong scaling for the mixed precision CG solver. Blue and red plots correspond to double-single and double-half mixed precision, respectively.*

For the code testing we employed FNAL GPU clusters is currently used for large GPU-count problems. Hardware details of this cluster:

- Vendor: Hewlett-Packard, using SL390s G7 blade server hosts
  - 48 GBytes memory and 2 GPUs per host, 76 hosts, 152 GPUs in total
- GPUs:
  - NVIDIA Tesla M2050, 1 TFlop/sec peak single precision, 448 cores
  - 3 GBytes memory per GPU, ECC-capable, hardware double precision

6. Conclusion and future work

The QUDA library was extended to implement an extra fermion operator thus extending the potential user base of this software package. Implementation of the non-degenerate twisted-mass operator will allow utilizing NVIDIA accelerators for a wider set of problems, in particular, problems which are relevant to the European Twisted Mass collaboration, one of the largest collaboration in Europe. The code will be further optimized by utilizing cache-blocking approach already adopted in the QUDA library for other fermion operators.