

B_K and related matrix elements with unquenched, improved staggered fermions

Chulwoo Jung (Brookhaven National Lab)
Weonjong Lee (Seoul National University)
Stephen Sharpe (University of Washington)

November 12, 2010

Abstract

This is a request for an additional allocation for the GPU part of our 2010-11 class A proposal. We are requesting GPU time to continue our calculation of kaon matrix elements using HYP-smearred improved staggered quarks. (The project web site is <http://www.phys.washington.edu/users/sharpe/qcdoc/index.html>.) Our major focus will be on a calculation of B_K , which is of particular interest for constraining elements of the CKM matrix. Using MILC coarse, fine and superfine lattices, we have a preliminary result, using SU(2) staggered chiral perturbation theory to do the chiral extrapolation and remove some lattice artifacts. The dominant sources of errors are the matching factors (calculated at one-loop) and statistics. The present proposal aims to reduce the statistical error on the superfine lattices, and add a second ensemble with smaller light sea-quark mass.

Graduate students and postdoctoral associates at both the University of Washington and Seoul National University will contribute to this project.

We are requesting 88,000 GPU-hours on the GPU cluster with Infiniband, and 540,000 Jpsi-equivalent core hours for storage.

Following the instructions in the 10/14/2010 call, we do not repeat the project description, but only discuss the justification for our additional request.

In our original proposal, we requested 120K GPU hours, and were allocated 75K GPU hours. As noted in an email from Chip Watson (10/15/10) this allocation has, like all others, been reduced, because of the initial overallocation of GPU resources. In fact, we were able to do little running before the reduction, so our effective allocation is $75K \times 0.55 \approx 40K$ GPU hours. We hope to use most of this during the rest of the calendar year.

The present request is for the balance of the original request of 120K GPU hours, rescaled by an improved estimate of our code performance to 128K GPU hours. Thus we are requesting $128 - 40 = 88K$ GPU hours.

In the following we provide an update on our code performance and our proposed running.

1 GPU performance

Our GPU code, written in CUDA, has been developed on a cluster at SNU having two GPUs attached to each node. At present we are using the GTX-480 GPUs (Fermi gaming card), but have previously used the GTX-295 GPUs. The nodes on this cluster are connected by DDR Infiniband, so the set-up is similar to that at JLab.

So far we only have converted the CG inverter to CUDA. We have been running the inverter on $48^3 \times 144$ MILC superfine lattices, spread over 16 and 32 GPU nodes. We use double precision. We have overcome difficulties related to the parallel I/O, and to the “packet-size” on the Infiniband. Our present performance is 55 GFlop/s/GPU without communication, and 13 GFlop/s/GPU including communication.

In our original proposal, we based our timing request on what we claimed was a conservative estimate of the performance that we would obtain (8 GFlop/s/GPU), as well as on a more aggressive estimate (15 GFlop/s/GPU). At the time we had obtained 1.9 GFlop/s/GPU. Thus we have almost achieved the more aggressive estimate.

We hope that during the next few months we will further speed up our code, but do not assume so in this proposal.

2 Proposed running on GPUs

Our first goal to substantially reduce the statistical error on the superfine lattices (ensemble S1 in our original proposal—743 lattices of size $48^3 \times 144$) by doing an additional 7 measurements on each lattice. Our second goal is to repeat the calculation on a second superfine lattice with a lighter sea-quark mass (ensemble S2—826 lattices of size $64^3 \times 144$) with 2 measurements per lattice.

Using the calculations of Floppage needed in our original proposal, but changing the speed to 13 GFlop/s/GPU, we find that the S1 and S2 calculations require 73K and 55K GPU-hours, respectively. The total is thus 128 K GPU-hours. Subtracting the 40 K GPU-hours from this calendar year, we arrive at our request for **88K** GPU-hours.

The main bottleneck in our code is the MPI communication. Hence, we prefer using the GPU clusters with QDR infiniband network to those with SDR networks. We also prefer FERMI

GPUs to GTX285 GPUs. Hence, the following table is the list of our priorities for running our production jobs.

priority rank	location	# of nodes	GPU	Communication
1	Jlab	32	NVIDIA C2050 (Fermi Tesla)	QDR infiniband
2	Jlab	50	NVIDIA GTX285	QDR infiniband
3	Jlab	46	NVIDIA GTX480	SDR infiniband
4	Jlab	32	NVIDIA GTX285	SDR infiniband

Table 1: Priority rankings.

Our storage needs are unchanged from our original proposal, and convert to **5.4×10^5 Jpsi-equivalent core-hours for disk storage.**