Lattice QCD simulations on big cats, sea monsters and clock towers

Bálint Joó on behalf of the USQCD Collaboration and the Hadron Spectrum Collaboration
Jefferson Lab, 12000 Jefferson Avenue, Newport News, VA 23606
E-mail: bjoo@jlab.org

Abstract. We present details of lattice QCD computations we are performing on the Cray XT series of computers, from BigBen – an XT3 hosted at the Pittsburgh Supercomputing Center (PSC) – through Jaguar (XT4) and Kraken (XT5) – which are hosted at the National Center for Computational Science (NCCS) and the National Institute of Computational Science (NICS), respectively, at Oak Ridge National Laboratory (ORNL). We discuss algorithmic tuning to make the computation more efficient and present some recent results.

1. Introduction
Lattice QCD is currently the only known model-independent, non-perturbative method for carrying out calculations of Quantum Chromodynamics (QCD), which are relevant to both Nuclear and High Energy physics. In this poster we report on progress on calculations which will enable the calculation of the excited state baryon spectrum, the photo-couplings of Hybrid Mesons and the investigation of multi-particle states. These calculations form the basis of DOE milestones in Hadronic Physics, and are of great relevance to the experimental program at the Jefferson Lab.

In this contribution, we focus on the initial phase of the calculation: gauge configuration generation. The extraction of physical results requires that these configurations be subjected to sophisticated analysis campaigns. We will not discuss the analysis and results further here, but will rather refer the reader to other contributions in this volume, in particular those of Edwards [1], Richards [2] and Luu [3].

2. Hybrid Monte Carlo for gauge generation
Discretizing QCD onto a lattice proceeds by ascribing fermionic degrees of freedom to the sites of a space-time lattice, and the gluonic degrees of freedom to the lattice links. The gluon fields are represented by $SU(3)$ matrices on the lattice links, which correspond to parallel transport operators between points connected by the links. It is common to abuse notation and denote as $U$ the collection link variables $U_\mu(x)$ for all the links on a given lattice.

Lattice QCD calculations generically involve evaluating the path integral

$$\langle O \rangle = \frac{1}{Z} \int \mathcal{D}U \ O(U) \prod_{i=1}^{N_f} \det(M_i(U)) \ e^{-S(U)},$$

(1)
where $\mathcal{O}$ is the observable of interest and $S(U)$ is the gauge action. The path integral has partition function $Z$ and the measure $D\mathcal{U}$ is shorthand for integrating over every configuration $U$. The fermionic degrees of freedom are integrated out, and their effect amounts to a weight in the path integral of $\det(M_i)$ where $M_i$ is the fermionic kernel for the $i$-th flavor of quark. We note that the work described in this article was carried out using anisotropic clover fermions.

An efficient way to evaluate the path integral is by a Monte Carlo method using importance sampling. This method produces lattice configurations $U$ with a probability $P(U)$ so that

$$P(U) \propto \prod_{i=1}^{N_f} \det(M_i(U)) e^{-S(U)}$$

and the expectation value reduces to a simple average:

$$\bar{O} = \frac{1}{N} \sum \{ U \} O(U).$$

The workhorse of the gauge generation process is the Hybrid Monte Carlo [4] algorithm and its variants [5]. Treating the $U$ link variables as canonical coordinates, these methods extend the configuration space to a phase space by assigning to each link a canonically conjugate momentum $\pi_{\mu}(x)$. One can then define a Hamiltonian $H = \frac{1}{2} \sum_{x,\mu} \pi(x,\mu)^2 + S(U)$. The algorithms iterate the following scheme: start from some initial gauge configuration $U$, generate fresh canonical momenta $\pi'$ and perform Hamiltonian Molecular Dynamics (MD) from this phase space state to a trial state $(U', \pi')$. The new state is subjected to a Metropolis acceptance test

$$P_{\text{acc}} = \min \left( 1, e^{-H(U',\pi')+H(U,\pi)} \right).$$

If the state is accepted the new starting state will be $(U', \pi')$, otherwise it will be the old state $(U, \pi)$. We note that the MD algorithm should be reversible and area preserving, so that the Monte Carlo process satisfies the condition of detailed balance, which is sufficient for the generated Markov Chain to converge to the desired equilibrium distribution. To satisfy this condition symmetric, symplectic integration schemes are used. The acceptance rate of the algorithm depends on the energy change $\delta H = H(U',\pi') - H(U,\pi)$ along an MD trajectory. This can be controlled by varying the step-size(s) of the underlying integration scheme. Finally, we note that the primary cost of the algorithm comes from the fact that in order to evaluate the contribution from the quarks one must solve the system

$$M_i^TMx = b$$

at least once per MD step ($x$ and $b$ are vectors). This system has a dimension of $O(V)$ where $V$ is the lattice volume, and has condition number that varies inversely with the quark mass. Dimensions and condition numbers, each of $O(10^7)$, are common. Hence, typical avenues of optimization include: developing faster implementations of $M$, developing solver algorithms which converge fast and finding integration schemes which reduce $\delta H$ on average so that one may take fewer, longer steps in the MD, thus requiring fewer solves.

3. Tuning the molecular dynamics integration

In our simulations, we employ a three level nested integration scheme where on each level we use the MD integrator due to Omelyan [6]. As described in [7], the leading order error term on a given level is given by

$$\left( \frac{6\lambda^2 - 6\lambda + 1}{12} A + \frac{1 - 6\lambda}{24} B \right) \delta\tau^2$$

where $\lambda$ is a tunable constant, $\delta\tau$ is the integration step-size, and $A$ and $B$ are Poisson Brackets (PB) of coordinate (gauge) and momentum update operators. In [7], minimising the variance of
Figure 1. Tuning a 3 level MD integration scheme. We show the parameter $\lambda$ from eq. (5) on the horizontal axis, and the variance of $\delta H$ on the vertical axis. The three lines correspond to the innermost (circles), middle (squares) and outer (diamonds) integrators.

Figure 2. The effect of improvements

Table 1. Our target large volume data sets. We show the number of trajectories generated at each set of parameters at the time of writing.

| $V = 24^3 \times 128$ | $V = 32^3 \times 256$ |
|----------------------|----------------------|
| $m_\pi = 230 MeV$   | 5380                 |
| $m_\pi = 380 MeV$   | 280                  |
|                      | 12985                |
|                      | 978                  |

$\delta H$ over an ensemble is advocated as the best way to tune an integration scheme. The optimal $\lambda$ on each level can be determined in an experimental fashion, by varying it and measuring the $\delta H$ on several trajectories to get an estimate of its variance. We have carried out the latter procedure on Kraken-XT at NICS and show our results in fig. 1, where filled symbols show our final interpolated minima. Our tuning was very rough: we carried out only 5 trajectories for each value of $\lambda$ for each level of the integrator. Thus, in terms of trajectories the cost of the tuning is about 80 trajectories in total. While this is a small percentage of a full run which comprises several 1000 trajectories, it is a noticeable amount of computer time. It can be seen that we have reduced the variance of $\delta H$ from $\approx 0.3$ to $\approx 0.05$, or by a factor of 6. This could in principle correspond to a factor of 6 increase in step-size, however in practice this is not attained as for such large outer steps, the integration would go unstable.

Apart from tuning our step-size, we have made several other improvements to our code. Whereas we used to solve eq. (4) with the Conjugate Gradients (CG) method, we have now switched over to a two step stabilized Bi-Conjugate Gradients (BiCGStab) procedure. Further, we have added threading to our linear operator $M$, our BLAS-like operations and most lattice site loops. The overall effect of these improvements is shown in fig. 2. Switching to BiCGStab and tuning have gained us a factor of 2 in wallclock time, with a small additional gain ($\approx 30\%$) from threading.
4. Cray experiences

Since 2007, we have used a variety of Cray XT platforms, starting with Jaguar at NCCS and BigBen at PSC. Both machines were then Cray XT3-s running the Catamount operating system. We have followed Jaguar through its various upgrades to dual, then quad cores; from XT3 to XT4 interconnects, and from Catamount to CNL operating systems. Currently we are running on Jaguar XT4, Jaguar XT5, Kraken XT5 at NICS and we are still active on BigBen at PSC through INCITE, NCCS discretionary and NSF Teragrid awards. Our primary code is the Chroma code \[8\] which was developed as part of the QCD Infrastructure project of SciDAC \[9\].

Our XT experiences are summarized in figs. 3 and 4. On BigBen we run jobs of size 2K cores, in single precision. On the XT3 platform we experienced good weak scaling as can be seen in fig 3 where we show the scaling of our code to 8192 cores on Jaguar after its dual core, but prior to its CNL upgrade. Indeed, the scaling is excellent. However, it can be seen that there is a distinct sweet spot in terms of local problem size (red line). Much larger local volumes fall out of cache (blue line) while much smaller problems are hit by surface to volume effects (black line).

In figure 4 we show normalized histograms of performances reported by the CG solvers on both BigBen, and on Jaguar XT4 post the quad core and CNL upgrade. While see some structure in the histogram for BigBen, essentially performances fall between 9.35-1.13 Gflops per core. On Jaguar XT4 the performances are substantially lower, around 450-700 Mflops/core. This may be due to the fact that on Jaguar we are working with a different global problem size, and that we are using 8192 cores, whereas on BigBen the partition size is 2048 cores. Nonetheless, the Jaguar histogram appears wider than the BigBen one with a shoulder extending toward lower performance numbers, with some as low or lower than 250Mflops/core. To ensure that this is not just a feature of our code or the large partitions, we show some data from running the MILC code on 2048 cores of Kraken XT4. This histogram is narrower than that of the Chroma code on Jaguar, but also exhibits a shoulder towards lower performances and is not immune from the occasional very low outlier.
5. Progress
We have made substantial progress in generating ensembles. Some initial results are presented in
[10, 11]. We show our progress in table 1. We are concentrating on two volumes: \( V = 24^3 \times 128 \) and \( V = 32^3 \times 256 \) lattice sites respectively; and at two values of the quark mass which correspond
to pion masses of \( m_\pi = 230 \) MeV and \( m_\pi = 380 \) MeV. The two volumes are required to ensure control over finite volume effects, and also because investigation of decays requires a finite size scaling analysis. With the INCITE 2008 resources, we have completed the run with \( V = 24^3 \times 128 \) and \( m_\pi = 380 \) MeV and with our 2008 NSF TeraGrid allocation we have begun the run at \( V = 24^3 \times 128 \) with a quark mass of \( m_\pi = 230 \) MeV which we are continuing on BigBen this year.

Using our INCITE 2009 and NSF Teragrid allocation, we have targeted the larger volume lattices using Kraken XT5 and Jaguar. We have not made as much progress as we intended for several reasons: Due to the large volumes used we found that single precision was no longer sufficient for the MD part of our simulations, so we moved to double precision. Secondly, we have scaled our running up to partitions of size 16384 cores and found a reduction in the performance of our code at this partition size. We have managed to compensate for some of this by the improvements in algorithm and software discussed earlier but progress remains slower than we’d like. We are currently in the process of investigating this issue.

6. Summary
We have discussed the program of anisotropic lattice gauge configuration generation and our
experiences on the Cray XT platforms to date. In particular we have presented some algorithmic
tuning results and touched on additional software improvements to our code. We discussed the
performance of our code on the Cray XT architecture through its various incarnations.

7. Acknowledgements
We wish to thank Mike Clark for many helpful discussions on the tuning Molecular Dynamics,
and Steve Gottlieb for supplying performance data from the MILC code. This research used
resources of the National Center for Computational Sciences at Oak Ridge National Laboratory,
which is supported by the Office of Science of the U.S. Department of Energy Under contract
DE-AC05-000R22725; and was supported in part by the National Science Foundation, through
TeraGrid allocations at the Kraken Cray XT5 facility at the National Institute for Computational
Sciences, The University of Tennessee and Oak Ridge National Laboratory and on the BigBen
Cray XT3 facility at the Pittsburgh Supercomputing Center. Authored by Jefferson Science
Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177. The U.S. Government
retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this
manuscript for U.S. Government purposes.

8. References
[1] Edwards R G these proceedings
[2] Richards D G these proceedings
[3] Luu T C these proceedings
[4] Duane S, Kennedy A D, Pendleton B J and Roweth D 1987 Phys. Lett. B195 216–222
[5] Clark M A, Kennedy A D and Sroczynski Z 2005 Nucl. Phys. Proc. Suppl. 140 835–837 (Preprint
hep-lat/0409133)
[6] I P Omelyan I M M and Folk R 2003 Comput. Phys. Commun. 151 272
[7] Clark M A, Kennedy A D and Silva P J 2008 Tuning HMC using Poisson brackets (Preprint 0810.1315)
[8] Edwards R G and Joo B (SciDAC) 2005 Nucl. Phys. Proc. Suppl. 140 832 (Preprint hep-lat/0409003)
[9] Joo B 2007 J. Phys. Conf. Ser. 78 012034
[10] Bulava J M et al. 2009 Phys. Rev. D79 034505 (Preprint 0901.0027)
[11] Lin H W et al. (Hadron Spectrum) 2009 Phys. Rev. D79 034502 (Preprint 0810.3588)