

# LQCD Computing Project Hardware Selection Strategy

*V1.0, March 17, 2009*

## I. USQCD Computing Requirements

Two classes of computing are done on lattice QCD machines. In the first class, a simulation of the QCD vacuum is carried out, and a time series of gauge configurations, which are representative samples of the vacuum, are generated and archived. Ensembles with varying lattice spacing and quark masses are generated, with each ensemble containing on order one thousand configurations. This class of calculation is entirely compute capability bound, with each new gauge configuration in an ensemble calculated from the previous. To minimize the actual calendar time required to generate the ensembles, configuration generation streams run on the largest capability compute systems available. The second class, the analysis phase, uses the archived configurations from each ensemble to calculate quantities of physical interest. A wide variety of different quantities can be calculated from each ensemble. These analysis computations also require large floating-point capabilities; however, the calculations performed on individual configurations are independent of each other. Therefore, analysis computations have orders of magnitude more job parallelism than do configuration generation computations.

Current calculations by the USQCD community use three principal actions for the generation of gauge configurations: improved staggered, or *asqtad*, domain wall fermion, or *DWF*, and anisotropic clover. The analysis of these types of gauge configurations requires similar computations to those performed during the configuration generation, such as the inversion of the Dirac operator. Thus the performance of the *asqtad*, *DWF*, and *clover* inverters is a key metric for judging the capability of hardware for lattice QCD.

Since 2000 the number of flops devoted to analysis computing for lattice QCD has been roughly equivalent to those required for configuration generation. This has shifted recently, however, with analysis computations predicted in the future to take up as many as 80% of the flops devoted to lattice QCD.

Because gauge configuration generation requires the largest capability machines possible, these computations are generally performed on the flagship NSF and DOE supercomputers, such as the current leadership machines at Argonne and Oak Ridge National Laboratories. By allocation policy, these machines are constrained to run only large jobs. Because analysis computation tends to use a greater number of smaller jobs, these calculations have typically not been performed on these large supercomputers. National funding for the various NSF and DOE supercomputer centers in recent years has been quite strong, and USQCD has enjoyed both large allocations of time as well as extensive access during the “early science” periods on new machines.

## II. User Behavior on USQCD Resources

Computing allocations on USQCD resources are provided in units of processor hours, and users are free to optimize their resources for their proposed science in any way that they like. Theirs is

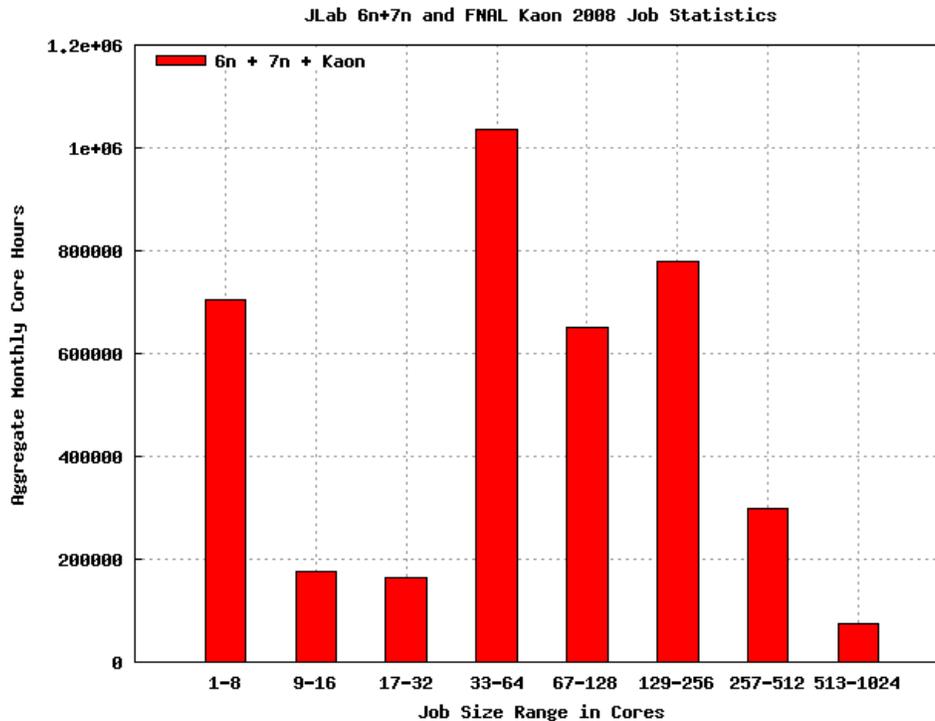
fundamentally a capacity computing problem consisting of running thousands of jobs against the thousands of configurations generated on the national and international supercomputers.

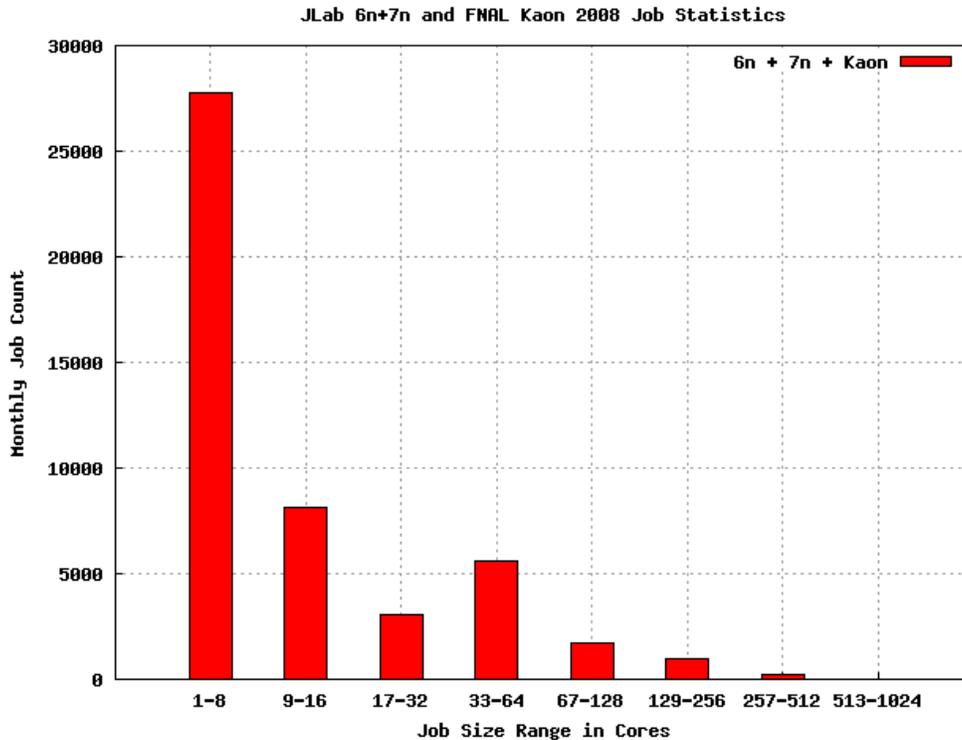
All production codes are parallel codes, and one of the important choices to be made is how parallel to run a particular calculation. A user can fill 1024 cores by running 128 8-way parallel jobs, by running 4 256-way parallel jobs, or by running a single 1024-way job. Strong scaling effects push the decision towards more jobs of smaller sizes. Memory constraints on our memory lean resources put a floor on how small the job can get. Work flow considerations push towards fewer, larger jobs in that in some cases it is easier to manage fewer job streams.

### *Bi-modal Distribution*

An analysis of the last couple of years of running at Fermilab and Jefferson Lab shows an interesting trend in the users' choices: the function of core hours consumed versus job size shows two peaks: (1) single node running (8 cores on the latest clusters), and (2) a broader peak from 32 cores up to 256 cores. There is also a certain amount of running at larger job sizes. These jobs correspond to configuration generation for the coarser and smaller lattices, i.e. jobs which are a bit small to run on the supercomputers but are otherwise similar to the other supercomputer center jobs. Total integrated processor hours used by these jobs is modest.

The two plots below show the aggregate data from 2008 for the Jefferson Lab 6n and 7n, and Fermilab Kaon, clusters. The first plot shows a histogram of consumed node hours, binned by the number of cores used. The second plot shows a histogram of executed jobs, binned by the number of cores used in each job.





### *Single Node Jobs*

Single node jobs yield higher flop/s per core than multi-node jobs for a number of reasons. First, all the communications overhead is avoided, including both the library overhead and the overhead to prepare a communications buffer. Second, communications latency is eliminated. Third, data can stay in cache without being flushed out to memory for communications, allowing all memory bandwidth to be used to support the calculation itself. These effects add up to several tens of percent gain in performance relative to multi-node jobs.

For 2 core cluster nodes (dual single core, or single dual core), this first peak is at 2 cores. For dual duals it is at 4 cores, and for dual quads (the largest fraction of USQCD resources today) it is at 8 cores. With the release of the Nehalem Intel server chips, this peak will move up to 16 cores (dual quads with two way hyper threading), and next year as the core counts go up again, this number will again increase.

### *Multi Node Jobs*

USQCD resources are typically memory lean (for price/performance reasons), and some jobs do not fit onto a single node. Once the penalty has been paid to turn on communications, surface to volume optimizations encourage running in a 2D or 3D virtual machine, and so job sizes jump by a factor of 4 or 8. This is readily seen in the graph above as the low end of the second peak is at 32 and 64 cores. As the job is made to extend over more and more cores, the total cache size increases, improving code efficiency for the single node even as communications overhead rises. Eventually the surface to volume ratio becomes too unfavorable, and thereafter strong scaling (performance per core on a fixed problem size) shows a strong downward trend.

Domain Wall, and to a lesser extent Anisotropic Clover, are able to run on larger numbers of cores simply because the problem sizes are larger (larger local volumes). Most users will give

up a small fraction of their performance, perhaps 10%, to get higher performance on a single job, and this user preference plus the variability in the actual work being performed yields job sizes in the multi-node peak from 32 to 256 cores, with the bulk of the work being done at 64 or 128 cores.

In the future, as larger and larger lattices are processed, this second broad peak will shift upward. Constrained by configuration generation, it will tend to grow with Moore's Law.

Analysis jobs generally have significantly higher I/O requirements than do configuration generation jobs. Propagators are larger than their parent gauge configurations and require fewer flops to calculate. "Tie-ups" require multiple propagators and do far fewer flops than either gauge configuration or propagator generation. Many of the current analysis jobs run on USQCD clusters rely on serialized I/O; often, the files required for a job are staged onto local disk from external storage at the beginning of a job, and output data files are similarly staged out to external storage from local disk at the end of a job, or at strategic intervals within larger jobs. Further, many parallel executables read or write data only through rank 0, again serializing I/O. Because users want to use their allocations efficiently, they will often use smaller jobs to minimize the idle time spent by assigned cores during these periods of serial I/O. Thus I/O, along with strong scaling, causes downward pressure on the sizes of analysis jobs. It is important that we take I/O considerations into account when analyzing the designs of the annual procurements and when assessing the performance and costs of vendor proposals.

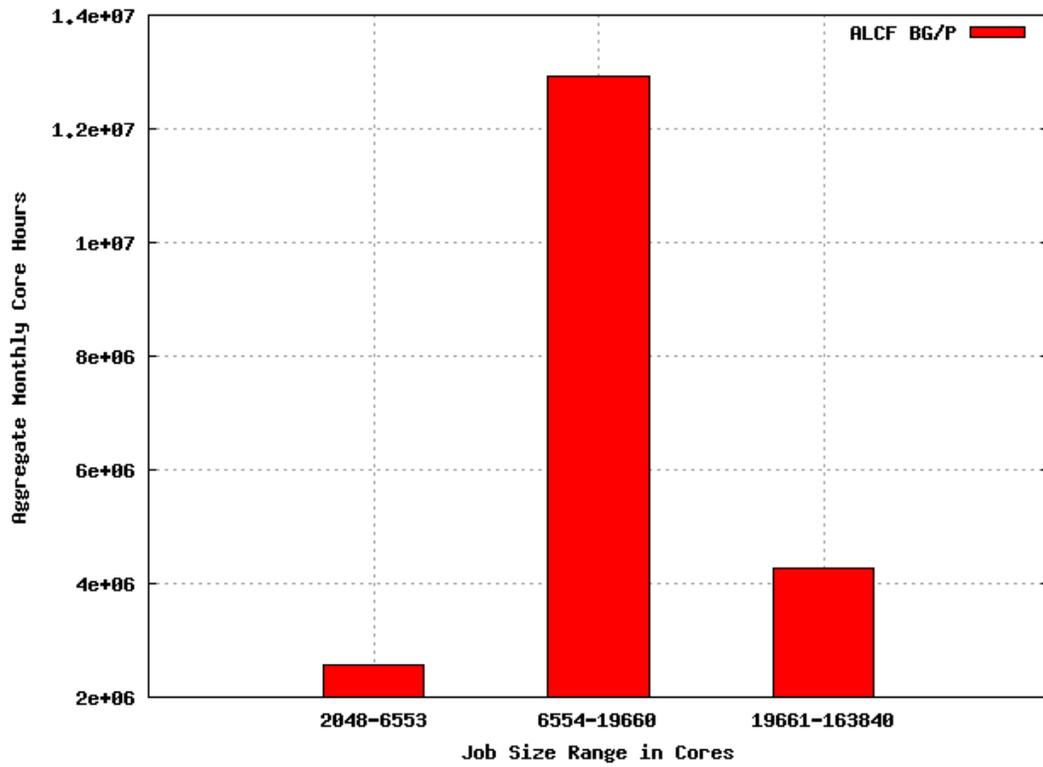
From these observations we propose that the appropriate benchmarking for understanding analysis computations will use job sizes with performance corresponding to roughly 1% of the performance of configuration generating jobs. In other words, propagator generation and other analysis tasks can and will be performed at (or below) 64-128 GFlop/s, while configuration generation will likely be performed at 5-10 Teraflop/s (with these ranges of performance increasing in time). Our assessments of hardware configurations should reflect this.

#### *USQCD Computing at the Leadership Centers*

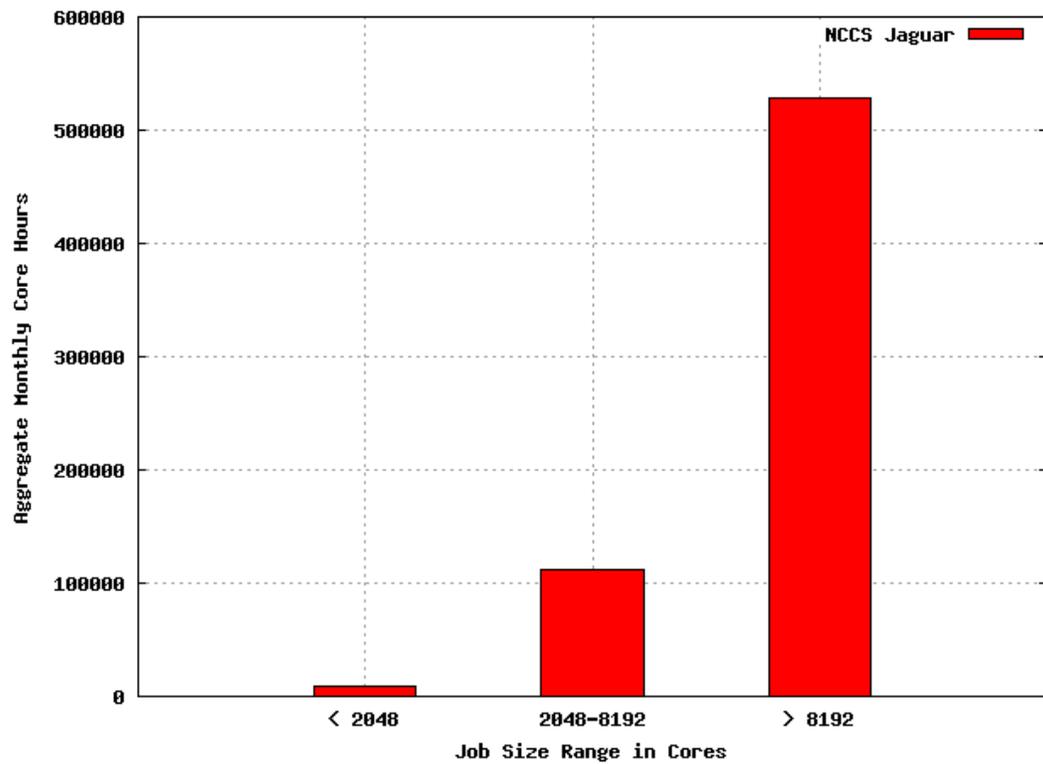
For completeness, histograms for 2008 for USQCD running on the ALCF BlueGene/P and the NCCS Cray XT4 (ORNL) are shown below. By policy, allocations are preferentially awarded to proposals whose computations can gainfully use large fractions of these supercomputers. Further, job scheduling policies give higher priority to larger jobs. Because of this scheduling policy, USQCD users on the NCCS Cray XT4 often run subjobs within larger jobs.

On the ALCF BlueGene/P, the approximate sustained LQCD performance per core is 645 MFlops, using an average of the *clover*, *asqtad*, and *DWF* actions. A 12288-core job, typical of the center bin in the plot above, therefore sustains approximately 7.9 TFlops in the inverter. On the NCCS Cray XT4 ("Jaguar"), production running in 2008 typically used 16K-core jobs. These jobs consisted of 4 to 5 smaller subjob streams of 3-4K cores each. All running on the XT4 used the *clover* action. Each 4K subjob sustained approximately 3.5 TFlops in the *clover* inverter.

ALCF 2008 Job Statistics



NCCS (ORNL) 2008 Job Statistics



### III. Benchmarking for USQCD

The LQCD Computing Project has as its primary objective the deployment of operation of the greatest amount of computing power for LQCD that can be achieved within the given budget. The decision about what computing platform to buy is thus tied to the performance of LQCD codes on that platform. In some years it is clear what the most cost effective platform will be, and it is just a question of which vendor will provide a suitable product at the lowest cost. In other years there are multiple viable platforms, and a “best value” procurement strategy is used. A key part of this best value strategy is a well defined LQCD benchmark suite.

#### *Actions*

In the first phase of our computing project, an average of the inverter performance for two actions was used: DWF and Asqtad. The size of the problem to be run was not well defined, but it was generally agreed that we would use local volumes corresponding to problem sizes likely to be run in the coming year(s).

For the second phase starting in 2010, it is worth raising the question of what kernels should be benchmarked. For at least the next several years, anisotropic clover is an important action for the USQCD program, and should be included. Weighting of the three actions need not be equal. At some point, perhaps even now, mixed precision or double precision should be added to the mix.

#### *Job Sizes*

Since configuration generation produces a thousand or more configurations, each of which is the starting point for another stream of computing, it is reasonable to target job sizes that are roughly 0.5% to 1% of the size in GFlop/s of configuration generation jobs. The computational demands devoted to the calculation of the physics observables are also increasing, exceeding 25% of the cost of the computation of quark propagators for some observables. These calculations involve predominantly local operations, with some global sums, performed on small numbers of nodes.

In light of this, and the observations in section II above on user preferences, for 2010 two job sizes are likely to be most important: ~20 GFlop/s (achievable by a single node Nehalem cluster node) and ~150 GFlop/s (8 nodes). Of course any platform which can support these job sizes with adequate resources (memory, disk server bandwidth, etc.) can be considered. In future years, these job sizes will roughly grow with Moore’s Law, with the larger job size remaining about 1% of the job sizes run on the supercomputers.

In summary, the benchmark suite should be composed of as many as a dozen benchmarks:

(3 actions, 2 precisions, 2 job sizes)

#### *Other Factors*

While application performance is the most important factor in the procurement, other factors must be taken into account at the 5% to 10% level. These factors include:

1. Platform scalability – ability to run jobs at higher performance if desired
2. User friendliness – productivity of the users
3. Disk bandwidth – especially important for analysis jobs
4. Tape storage – access to archival storage (bandwidth, cost, volume)
5. Physical size – more important at some sites than others
6. Power and cooling requirements – impact on lifecycle costs

### III. Evaluations

As described in the Project Execution Plan for the LQCD Computing Project, procurement decisions will be primarily based upon careful evaluations of LQCD benchmarks, with the intent to procure the greatest science capability for the fixed budget. Two different activities related to platform selection are defined for the project: (1) a tentative budget and procurement plan presented each spring at the annual project review and included in the annual OMB-300 submission, and (2) a just-in-time evaluation conducted as part of the procurement activity. This second evaluation activity captures the latest performance and market data, and can either be done ahead of a low-bid call for proposals if the winning architecture is clear (true for almost half of our procurements to date), or during a best-value call for proposals, used when the winning platform is not clearly known ahead of the call.

The project also maintains an Alternatives Analysis document, which lists the specific alternative architectures being considered, and gives an estimate of price / performance for these platforms. As explained in that document, project staff track the market and attend non-disclosure meetings with various vendors to keep these analyses as correct as possible. The selected, preferred alternative in this document is of course not necessarily the alternative that will win a competitive bid process, and the project plan and OMB-300 submission clearly states that the final decision will be made as close to procurement as possible. The following is a condensed version of such an alternatives analysis, but without as detailed cost information. It gives our best understanding of the near-term prospects for the FY2010 procurement.

#### *Clusters and Cluster Nodes*

The commodity market currently includes only two families of microprocessors that are cost effective for large-scale LQCD clusters: AMD Opteron, and Intel Xeon, which both support the x86 instruction set. From 2006 through the first quarter of 2009, only AMD Opteron-based systems had sufficient memory bandwidth to allow the use of two or more processor sockets in a node in order to drive down the cost of the Infiniband interconnect used for communications on multi-node jobs. The Intel “Nehalem” architecture, available in late 2008 on single processor socket machines, and in mid-2009 on dual processor socket machines, greatly increases memory bandwidth. We expect Nehalem to be the preferred x86-compatible processor component for any commodity cluster purchases for LQCD in 2010.

The table below shows the measured performance of recent Intel and Opteron processor clusters on the three LQCD actions, as measured on USQCD clusters at Jefferson Lab and Fermilab. Also shown in italics are the estimated performances of Shanghai- (latest AMD Opteron generation) and Nehalem-based clusters, based on single-node benchmarks. For Nehalem we assume an 80% scaling factor from single-node to multiple-node runs of typical size (64 to 128 cores). This is a conservative scaling estimate; observed scaling on the existing USQCD quad-core Opteron systems is 90%, while a measurement of scaling on an early Nehalem test cluster was 80%. However, the increased bandwidth and lower latency of new Infiniband quad data rate hardware (the prior scaling test used double data rate) may improve this scaling factor. We assume Shanghai scaling to match the scaling observed on the USQCD Opteron clusters.

Cluster	Processor	DWF Performance per Node	Clover Performance per Node	Asqtad Performance per Node	6n equiv.
6n	3.0 GHz Single CPU Dual Core Pentium	2900 MFlops	1408 MFlops	1960 MFlops	1.0
kaon	2.0 GHz Dual CPU Dual Core Opteron	4696 MFlops	3180 MFlops	3832 MFlops	1.757
7n	1.9 GHz Dual CPU Quad Core Opteron	8800 MFlops	5148 MFlops	6300 MFlops	3.10
J/Psi	2.1 GHz Dual CPU Quad Core Opteron	10061 MFlops	7423 MFlops	9563 MFlops	4.04
<i>Shanghai</i>	<i>2.4 GHz Dual CPU Quad Core Opteron</i>	<i>12530 MFlops</i>	<i>Not measured</i>	<i>10370 MFlops</i>	<i>4.71</i>
<i>Nehalem 1066 MHz FSB</i>	<i>2.26 GHz Dual CPU Quad Core Xeon</i>	<i>22200 MFlops</i>	<i>12460 MFlops</i>	<i>15940 MFlops</i>	<i>7.8</i>
<i>Nehalem 1333 MHz FSB</i>	<i>2.93 GHz Dual CPU Quad Core Xeon</i>	<i>27720 MFlops</i>	<i>15260 MFlops</i>	<i>19390 MFlops</i>	<i>9.7</i>

In late 2010, according to roadmaps the next generation of Intel processor, code-named “Westmere”, will be released. This generation will use a 32nm process, rather than Nehalem’s 45nm process, and it will have 6 cores per socket. This core increase, combined with a boost in memory bandwidth from an increase in memory clock to 1600 MHz, should maintain the Intel family as the most cost-effective commodity processor.

USQCD clusters since 2005 have utilized Infiniband as the interconnect fabric. The latest cluster, Fermilab’s J/Psi, used double-data rate (20 gb/sec signal rate) Infiniband. Quad-data rate (QDR) components are currently available, with better bandwidth, reduced latency (particularly on newer motherboards with second-generation PCI Express), and improved switching (adaptive routine, congestion control). QDR Infiniband is the assumed network fabric of choice for 2010.

The FY08/FY09 combined USQCD purchase of the J/Psi cluster at Fermilab had a cost of \$1.929M. The aggregate performance of this 856-node, 6848-core cluster, based on the per-node DWF-asqtad average (see <http://lqcd.fnal.gov/performance.html>) of 9.812 GF/node is 8.399 TF. Thus the price/performance of J/Psi for USQCD analysis production is \$0.23/MFlop. Costs are not yet available for Nehalem clusters. However, our best guess is that a cluster based on the slower Nehalem chip in the table above will have a similar cost per node in 2010 to the J/Psi cluster. Thus we estimate the price/performance of a 2010 Nehalem cluster to be \$0.12/MFlop.

### *Computations Based on GPU Hardware*

Graphics processing units (GPUs) in recent years have started to be used for numerical computations, a practice known as GPGPU (General Purpose computations on Graphics Processing Units). N6grádi *et al* at the Lattice’06 conference reported on using OpenGL codes on NVidia GPUs for the Wilson action. Since then, the CUDA programming environment on newer NVidia GPU hardware has considerably eased the programming difficulty. Rebbi *et al* at

the Lattice'08 conference reported on using CUDA codes on NVidia Tesla C870 and GTX 280 GPUs, achieving 92 and 80 GFlops, respectively, on the latter GPU on the Wilson-Dirac operator and on the Wilson conjugate gradient inverter.

Although the utility of GPUs is currently limited to jobs that fit within a single processor, and to the actions that have been coded to date, they are nevertheless extremely cost effective for those parts of the USQCD physics program for which they are appropriate. NVidia packages four Tesla GPUs in a 1U chassis that connects to a pair of Unix hosts via external PCI Express cables. The cost, as of early 2009, for one of these quad-GPU units is \$6400, with computational throughput of approximately 300 GFlops aggregate on the existing codes (\$0.021/MFlop); note that this does not include the cost of the required host nodes. If host node costs are included, the price/performance increases to roughly \$0.037/MFlop.

For GPUs, or any other accelerator, additional considerations are needed, specifically the effects of Amdahl's Law. If 99% of the run time is within the GPU, then the above analysis is valid. But if only 60% of the code were to be in the inverter, and if that were the only optimized portion of code, then if the inverter were running at 20x the speed of the host, then the actual acceleration is only  $100/(40 + 60/20) = 2.32$ , not 20. This effect would make the combined host+GPU less cost effective on full applications than just the host. Therefore, in evaluating an accelerator like the GPU, it will be necessary to assign a number which is the fraction of the code which, on average, will be accelerated by that hardware.

In 2009 USQCD is adding four of these quad GPU systems to the Fermilab J/Psi cluster, to be used for one of the allocated physics projects. JLab is deploying a Nehalem+GPU test system. These R&D systems will serve as software development platforms to ascertain how much code can usefully be moved onto the GPU, and with what total effect. In 2010 forward, a fraction of the annual hardware procurement for USQCD should go towards GPU hardware if it is shown to be cost-effective and appropriate for anticipated allocations.

The NVidia hardware includes DMA engines that in principal should allow implementation of parallel codes running over multiple GPUs, initially within a single system, and perhaps eventually using Infiniband to run over GPUs in multiple boxes. Specifications for access to this DMA engine are not yet available, but access is promised in a future version of the CUDA environment. With suitable software development to exploit this capability, GPU based hardware could take on a greater fraction of LQCD computations than is allowed by current limitations.

Both AMD, with their ATI GPUs, and Intel, with their forthcoming "Larrabee" GPUs, provide additional GPU-class hardware that could potentially be exploited for LQCD. Competition among these vendors is clearly healthy for USQCD in terms of downwards pressure on pricing and upwards pressure on performance. AMD and NVidia are also both participating in the standardization of the OpenCL language, an alternative to CUDA that could allow LQCD code to run unmodified on different multicore processors.

### *Blue Gene /\* Supercomputers*

The BlueGene/\* line of commercial supercomputers is well matched to the requirements of LQCD. While the BlueGene/P is not cost competitive with clusters for the non-configuration-generation jobs addressed by the LQCD Computing project, the BlueGene/Q is expected to be

much more cost effective, perhaps by a factor of 4 or more. Release date is somewhat uncertain, but late 2010 or early 2011 is a good estimate. Thus, this machine will arrive too late for the first years of the LQCD project extension, but should be a candidate for FY2012. Evolution of this future machine will be tracked each year just ahead of procurements to evaluate its suitability for inclusion in a benchmarking process.

#### **IV. Discussion**

For the LQCD II project, it is critical that USQCD establish metrics for assessing the performance of computing hardware for LQCD calculations. The metrics will not only be used in the decisions of the type of hardware to buy in each year, but they should also be used as the standard measurements for reporting performance to the DOE and OMB (deployed TFlops, delivered TFlops-yrs). In the first LQCD project, USQCD used the average of the performance of the DWF and asqtad actions, measured using the optimal running conditions for each type of hardware. On commodity clusters, optimal running (*i.e.* the configuration that gave the highest performance numbers) uses large local volumes and, when possible, fewer directions of communications (for asqtad, 4-D communications were used, whereas for DWF either 2-D or 3-D communications were used). Job sizes were picked to be large enough to involve a non-trivial number of nodes. On early clusters with one core per node, we used 64-process jobs, and on later clusters with four or eight cores per node, we used 128-process jobs.

For LQCD II, in light of the discussions above, a reasonable metric likely involves a weighted average of the performance of the three actions, using single and double precisions, at multiple characteristic jobs sizes (single node, multiple node at the likely size of common analysis jobs, and perhaps multiple node at the size appropriate for configuration generation).

For the purpose of awarding purchases, the price/performance obtained using this metric will have to be adjusted to take into account all other relevant production and cost factors.