# Lattice QCD Application Development within the US DOE Exascale Computing Project

*Richard* Brower[1], *Norman* Christ[2], *Carleton* DeTar[3,⋆], *Robert* Edwards[4], and *Paul* Mackenzie[5]

[1]*Physics Department, Boston University, Boston, Massachusetts 02215, USA*
[2]*Physics Department, Columbia University, New York, New York 10027, USA*
[3]*Department of Physics and Astronomy, University of Utah, Salt Lake City, Utah 84112, USA*
[4]*Thomas Jefferson National Accelerator Facility, Newport News, Virginia 23529, USA*
[5]*Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA*

**Abstract.** In October, 2016, the US Department of Energy launched the Exascale Computing Project, which aims to deploy exascale computing resources for science and engineering in the early 2020's. The project brings together application teams, software developers, and hardware vendors in order to realize this goal. Lattice QCD is one of the applications. Members of the US lattice gauge theory community with significant collaborators abroad are developing algorithms and software for exascale lattice QCD calculations. We give a short description of the project, our activities, and our plans.

## 1 Introduction

The Exascale Computing Project (ECP) is a joint undertaking by the US Department of Energy Office of Science (DOE-SC) and the US National Nuclear Security Administration (NNSA) [1]. Launched in October, 2016, its goal is to develop computers, software, and algorithms that will provide fifty times the power of today's fastest machines. The initial "advanced architecture" is planned for the year 2021. At least one, possibly two computers are envisioned. A "capable exascale" machine, based on ECP research and development, is planned for 2023. The procurement process is not part of the ECP.

The US is not alone in advancing to the exascale. There are similar initiatives in Europe [2], Japan [3], and China [4], some of them already well underway.

Exascale computing presents a host of technical hardware challenges. Here are some of the most important ones:

1. Increased parallelism. We expect a thousandfold increase in parallelism over that of today's systems.

2. Memory, network, and storage efficiencies. These must be improved in a manner consistent with increased computational rates and data-movement requirements.

3. Reliability. As systems grow in size and complexity, the ability to adapt and recover from faults will become crucial.

⋆Speaker, e-mail: detar@physics.utah.edu

4. Energy consumption. Electrical power consumption must be significantly reduced from today's standards.

To meet these challenges will require considerable industrial innovation.

How is exascale capability defined? It is not defined in terms of a peak flop rate. It must be meaningful to the mix of applications that are expected to need computing at that scale. It is defined as follows:

1. Fifty-fold increase in application performance. With a facility on the scale of a "leadership computing facility" (such as the Argonne and Oak Ridge Leadership Computing Facilities) it should deliver 50 times the performance of today's 20 Petaflop systems for a wide range of applications. This criterion is spelled out further below.
2. Power consumption. The targeted range is 20–30 MW.
3. Resilience. The perceived fault rate should be less than one per week.
4. Software. It should includes a software stack that supports a broad spectrum of applications and workloads.

## 2 National organization of the ECP

The ECP is headed by Paul Messina (Argonne Leadership Computing Facility)[1] and managed through Oak Ridge National Laboratory. The ECP partners with six major US DOE national laboratories, all with strong high-performance computing expertise. They are Argonne, Lawrence Livermore, Lawrence Berkeley, Los Alamos, Oak Ridge, and Sandia National Laboratories. Universities throughout the US are also involved, as are industrial partners.

The ECP development effort follows a "holistic co-design" principle that integrates the efforts of application developers and the developers of software and hardware technology. Figure 1 gives an idea of the mix. There are some thirty application development projects involved. They were selected to represent a diverse mix of the HPC scientific and "mission" programs supported by the DOE. They include applications ranging from cosmology to electrical power grid design. The software technology component includes some 80 teams with subjects ranging from MPI to solvers. Some thirty commercial vendors are involved in developing hardware technology.

In the past year, the DOE announced its selection of six major vendors under the "PathForward" program. This is a US$430 M joint industry/DOE undertaking with US$258 M being provided by the DOE in stages and the balance provided by the industrial partners. The partners are AMD, Cray, Hewlett Packard Enterprise (HPE), IBM, Intel, and NVIDIA. As they proceed with hardware development, they will be testing systems on testbeds being set up at the national laboratories.

## 3 US Lattice QCD Exascale Development

The scientific goals for US exascale computing are shared by many in the world community. They include

- Assisting in the search for physics beyond the Standard Model by comparing precise calculations with precise experimental results.
  - $B$ physics at lattice spacing smaller than $1/m_b$.

[1]Douglas Kothe assumed this leadership position on October 1, 2017.

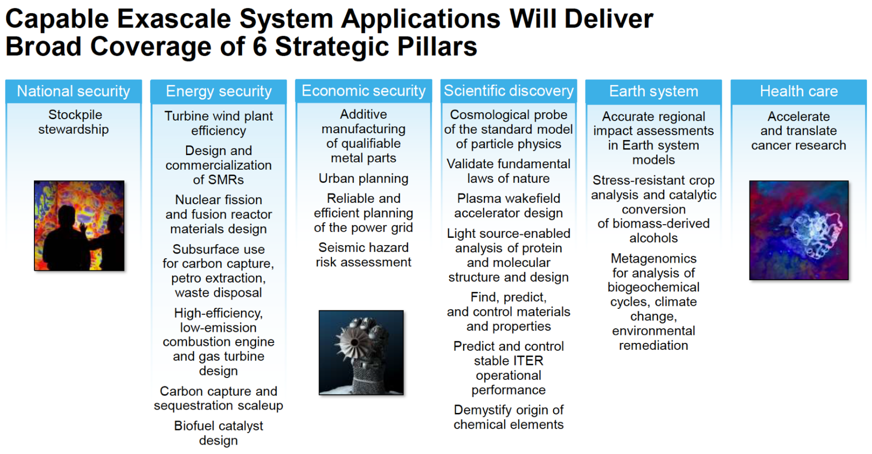
Capable Exascale System Applications Will Deliver Broad Coverage of 6 Strategic Pillars
National security
Stockpile stewardship
Energy security
Turbine wind plant efficiency
Design and commercialization of SMRs
Nuclear fission and fusion reactor materials design
Subsurface use for carbon capture, petro extraction, waste disposal
High-efficiency, low-emission combustion engine and gas turbine design
Carbon capture and sequestration scaleup
Biofuel catalyst design
Economic security
Additive manufacturing of qualifiable metal parts
Urban planning
Reliable and efficient planning of the power grid
Seismic hazard risk assessment
Scientific discovery
Cosmological probe of the standard model of particle physics
Validate fundamental laws of nature
Plasma wakefield accelerator design
Light source-enabled analysis of protein and molecular structure and design
Find, predict, and control materials and properties
Predict and control stable ITER operational performance
Demystify origin of chemical elements
Earth system
Accurate regional impact assessments in Earth system models
Stress-resistant crop analysis and catalytic conversion of biomass-derived alcohols
Metagenomics for analysis of biogeochemical cycles, climate change, environmental remediation
Health care
Accelerate and translate cancer research

**Figure 1.** Illustration of the range of applications targeted by the ECP from [5].

  - Understanding CP violation in the K system
  - Simulating "non-standard" weak K decays.
  - A ten-fold reduction in uncertainties in weak matrix elements.

- Calculations at physical quark masses with QED.
- Precision study of the structure and interactions of light nuclei

Some fifty lattice physicists participate in the lattice QCD application development project. Several postdocs and lab staff receive some salary support, but the rest do not receive salary support. In addition to US lattice gauge theorists from labs and universities, this effort includes significant participants at the University of Edinburgh and NVIDIA corporation, and it includes applied mathematicians and computer scientists. The lattice effort is coordinated through a steering committee with Paul Mackenzie as PI, and four working groups:

- Solver research (Richard Brower)
- Critical slowing down research (Norman Christ)
- Algorithms for Wick constractions (Robert Edwards)
- Software development (Carleton DeTar)

*Solver research*

The research program investigates algorithms that will improve the efficiency of the solvers used in exascale calculations. An important activity is to select or develop robust algorithms that reduce or avoid internode communication. Another is the application of multigrid algorithms to the staggered fermion formulation [6]. Other topics incude combining multigrid algorithms with deflation, domain decomposition solvers, and all-mode-averaging with eigenvector compression [7].

*Critical slowing down*

Exascale will make it possible to do calculations at a fine enough lattice spacing that critical slowing down of gauge-field generation algorithms becomes an increasing concern. While the most dramatic consequence of this critical slowing down is the freezing of topology, one expects the autocorrelation times of physical observables to grow as the inverse square of the lattice spacing, $1/a^2$. Even with open boundary conditions, the local diffusion of topology from the boundaries into the bulk is expected to slow as $1/a^2$

*Algorithms for Wick contractions*

For lattice QCD calculations of hadronic matrix elements of even light nuclei, dealing with all the quark-line tensor (Wick) contractions presents a considerable combinatoric problem. So it is important to develop efficient strategies that reduce the cost of calculation. These strategies involve finding a unique set of "edge reductions" that maximize reuse of building blocks, and to rearrange the graph order to lower the number of temporaries.

*Software development*

For many years the USQCD community has been developing lattice-QCD-specific community codes, including the DOE SciDAC suite: QDP/C, QDP++, and QLUA [10]. Specifically for GPUs, QUDA provides excellent performance[11]. The several US lattice collaborations have also developed code bases more specific to the scientific goals of the respective collaborations, such as Chroma [12, 13], the CPS code [14], and the MILC code [15].

Drawing from this experience, we are formulating a new data parallel applications programming interface (API) that supports the algorithms under development for lattice QCD. These include support for multigrid and domain-decomposed solvers. The API design should be extensible and allow for rapid prototyping of new algorithms.

We are also looking for implementations that port readily between many core and GPU architectures and yet deliver excellent performance on both. We are particularly interested in the Grid software system being developed by Peter Boyle and collaborators at Edinburgh [16, 17]. It was originally designed to perform especially well on the vectorized many-core Intel KNL architecture. However, its design is flexible, so we have begun experimenting with strategies for porting it to GPUs [18].

In preparation for formulating a data-parallel API, the software group has laid out the required functionality and assessed semantic alternatives for expressing it. We are looking at both C++-11 and Nim [19]. We have been using the Grid software system as an example of C++ semantics and the QEX framework of James Osborn and collaborators for Nim [20, 21]. We give a few examples of each.

Parallel transport is obviously important for QCD calculations. In Grid the syntax is

```
tmp = CovShiftForward(U[mu], mu, src);
```

where both `tmp` and `src` are lattice fields with a color index. In QEX the syntax is

```
let T = newTransporters(U, src, 1)
tmp = T[mu] ^* src
```

For another example, for mulitgrid algorithms we start from a fine grid, subdivide it into blocks, and project to a coarse grid, using a set of $n$ basis vectors as block projectors. In Grid the syntax for this operation is

```
blockProject( coarseData, fineData, Basis );
```

and in QEX it is

```
let R = newRestrictor(coarseLattice, Basis)
coarseData := R ^* fineData
```

Data-parallel operations hide loops over all sites. However, it is often desirable to fuse multiple loops into a single loop to avoid creating temporary intermediate fields and wasting time with avoidable memory traffic. One way loop fusion can occur in data-parallel operation is through the use of expression templates. Consider the operation

```
  q = a*x + y
  r = y + b*q;
```

where q, r, x, and y are lattice fields and a and b are scalar constants. With expression templates, the first operation can be compiled into a single, threaded site loop. But the second operation gets a separate loop, an y must be kept as a temporary intermediate. If we are willing to expose the site index, we can fuse the loops. Here is an example from Grid:

```
PARALLEL_FOR_LOOP
for(s = 0; s < nsites; s++){
  q[s] = a*x[s] + y[s];
  r[s] = y[s] + b*q[s];
}
```

where the PARALLEL_FOR_LOOP generates an OpneMP pragma, requesting the the fused loop be threaed. With QEX it is proposed to do it this way:

```
fuse:
  q = a*x + y
  r = y + b*q
```

So with QEX, an explicit directive is provided to force fusion, but the site index is not exposed.

The US ECP project could make an interesting case study for a sociologist. The management regimen is quite a bit stricter and more corporation-like than most of us outside industry are used to. We are required to use sophisticated project management systems, namely, Atlassian's Confluence and Jira to manage our work, defining and setting intermediate goals and deliverables to keep the work on track. Clearly, too much such management risks hampering creativity, while too little risks failing to reach goals.

Integrating the work of many projects is also presents a sociological challenge. "Match making" was the central purpose of the first annual plenary ECP meeting, which took place in Knoxville, Tennessee, January, 2017. For Lattice QCD, the strongest interest thus far is in the Kokkos [22] and Trilinos [23] software efforts, OpenMP, OpenACC, and MPI design, HDF5, and checkpointing schemes.

Success of the entire ECP project will be measured by the ability of most of the included applications to reach the goal of a fifty-fold increase in computing capability. All project improvements are to be included in determining success, namely, hardware, systems design, software, and algorithms. To measure improvement in performance, each application is required to specify a set of application-specific "Key Performance Parameters (KPP)" and "Figures of Merit (FOM)". For Lattice QCD, we have defined a set of lattice-generation and physics-analysis tasks and measured the time it takes to complete those tasks on a specified fraction of a leadership-class facility, such as those at Argonne or Oak Ridge National Laboratories. We are then aiming for a fifty-fold improvement in the time required to complete the same tasks on the eventual exascale replacement of the same facility.

## 4 Conclusion and Outlook

The US DOE Exascale Computing Project is helping to accelerate the arrival of wide-spread exascale computing. Coordination among hardware vendors, systems development teams, and application developers is crucial to its success. We hope our efforts contribute to the success of the lattice QCD program in the US and to the international effort as well.

## 5 Acknowledgment

This project is supported by the US Department of Energy, Exascale Computing Project.

## References

[1] US Department of Energy, *Exascale computing project*, https://exascaleproject.org

[2] European Commission, *European exascale projects*, http://exascale-projects.eu

[3] Riken Advanced Institute for Computational Science, *Post-K project*, http://www.aics.riken.jp/en/postk/project

[4] Xinhua Net, http://www.scmp.com/news/china/society/article/2107796/worlds-next-fastest-supercomputer-will-help-boost-chinas-growing

[5] P. Messina, *Update on the Exascale Computing Project (ECP)*, https://exascaleproject.org/wp-content/uploads/2017/04/Messina-ECP-Presentation-HPC-User-Forum-2017-04-18.pdf

[6] E. Weinberg, *Update on a Staggered Multigrid Algorithm in Four Dimensions*, in *Proceedings, 35th International Symposium on Lattice Field Theory (Lattice2017): Granada, Spain*, to appear in EPJ Web Conf., 1710.01000

[7] C. Lehner, *Multi-grid Lanczos*, in *Proceedings, 35th International Symposium on Lattice Field Theory (Lattice2017): Granada, Spain*, to appear in EPJ Web Conf., 1710.01000

[8] G. McGlynn, R.D. Mawhinney, Phys. Rev. **D90**, 074502 (2014), 1406.4551

[9] C. Bernard, D. Toussaint (MILC) (2017), 1707.05430

[10] USQCD Collaboration, *SciDAC software suite*, https://usqcd-software.github.io

[11] R. Babbich, K. Clark, A. Strelchenko, A. Vaquero, M. Wagner et al., *QUDA library*, https://github.com/lattice/quda

[12] R.G. Edwards, B. Joo, *Chroma software system*, https://jeffersonlab.github.io/chroma/

[13] R.G. Edwards, B. Joo (SciDAC, LHPC, UKQCD), Nucl. Phys. Proc. Suppl. **140**, 832 (2005), hep-lat/0409003

[14] RBC and UKQCD collaborations, *CPS software system*, https://usqcd-software.github.io/CPS.html

[15] MILC collaboration, *MILC code*, https://github.com/milc-qcd/milc_qcd

[16] P.A. Boyle, G. Cossu, A. Yamaguchi, A. Portelli, PoS **LATTICE2015**, 023 (2016)

[17] P. Boyle et al., *Grid: Data parallel C++ mathematical object library*, https://github.com/paboyle/Grid

[18] M. Lin, P. Boyle, K. Clark, C. DeTar, A. Vaquero, *Performance Portability Strategies for Grid C++ Expression Template*, in *Proceedings, 35th International Symposium on Lattice Field Theory (Lattice2017): Granada, Spain*, to appear in EPJ Web Conf., 1710.01000

[19] A. Rumpf, *NIM Programming Language*, https://nim-lang.org

[20] X.Y. Jin, J.C. Osborn, PoS **ICHEP2016**, 187 (2016), 1612.02750

[21] X.Y. Jin, J. Osborn, *QEX software*, https://github.com/jcosborn/qex

[22] C. Trott et al., *Kokkos Project*, https://github.com/kokkos

[23] M. Heroux et al., *Trilinos Project*, https://trilinos.org