Computation for Beyond Standard Model Physics

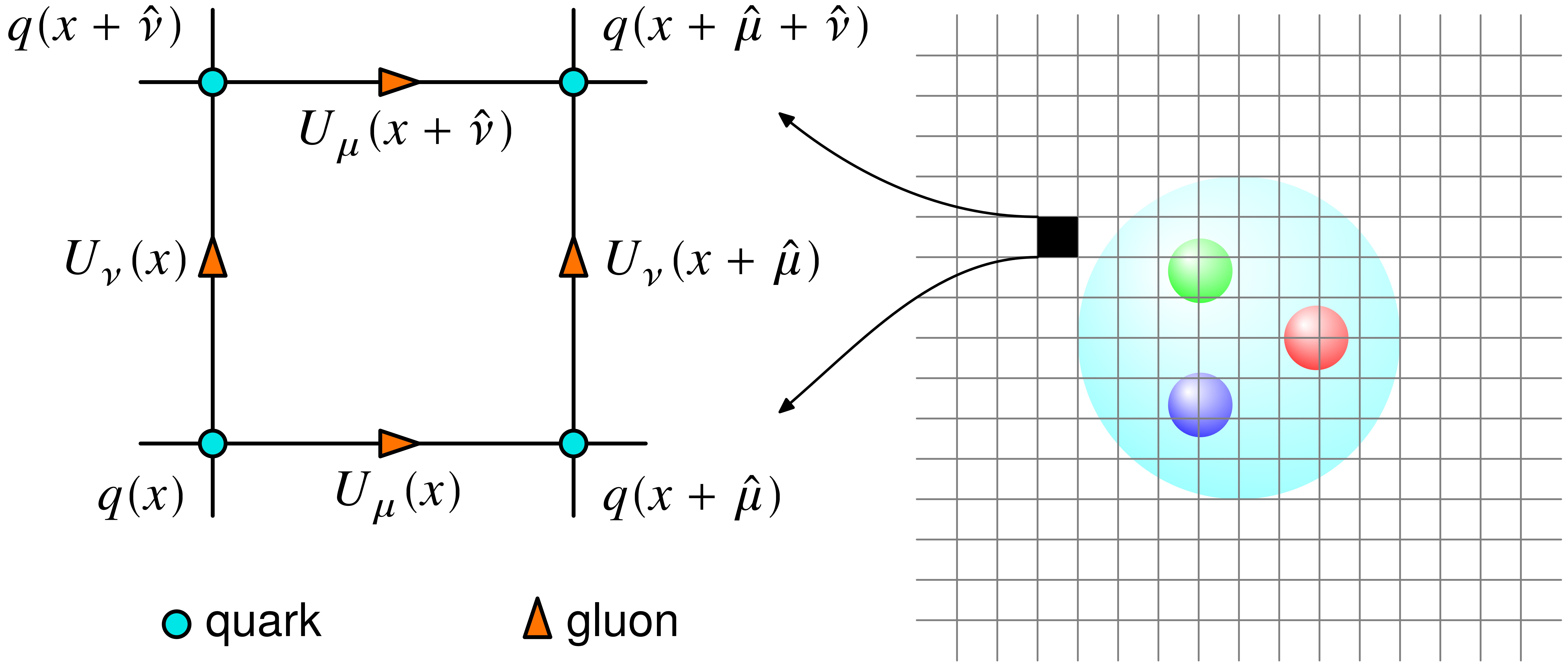
Xiao-Yong Jin
Argonne National Laboratory

Lattice for BSM Physics 2018
Boulder, Colorado
April 6, 2018

My PhD years at Columbia

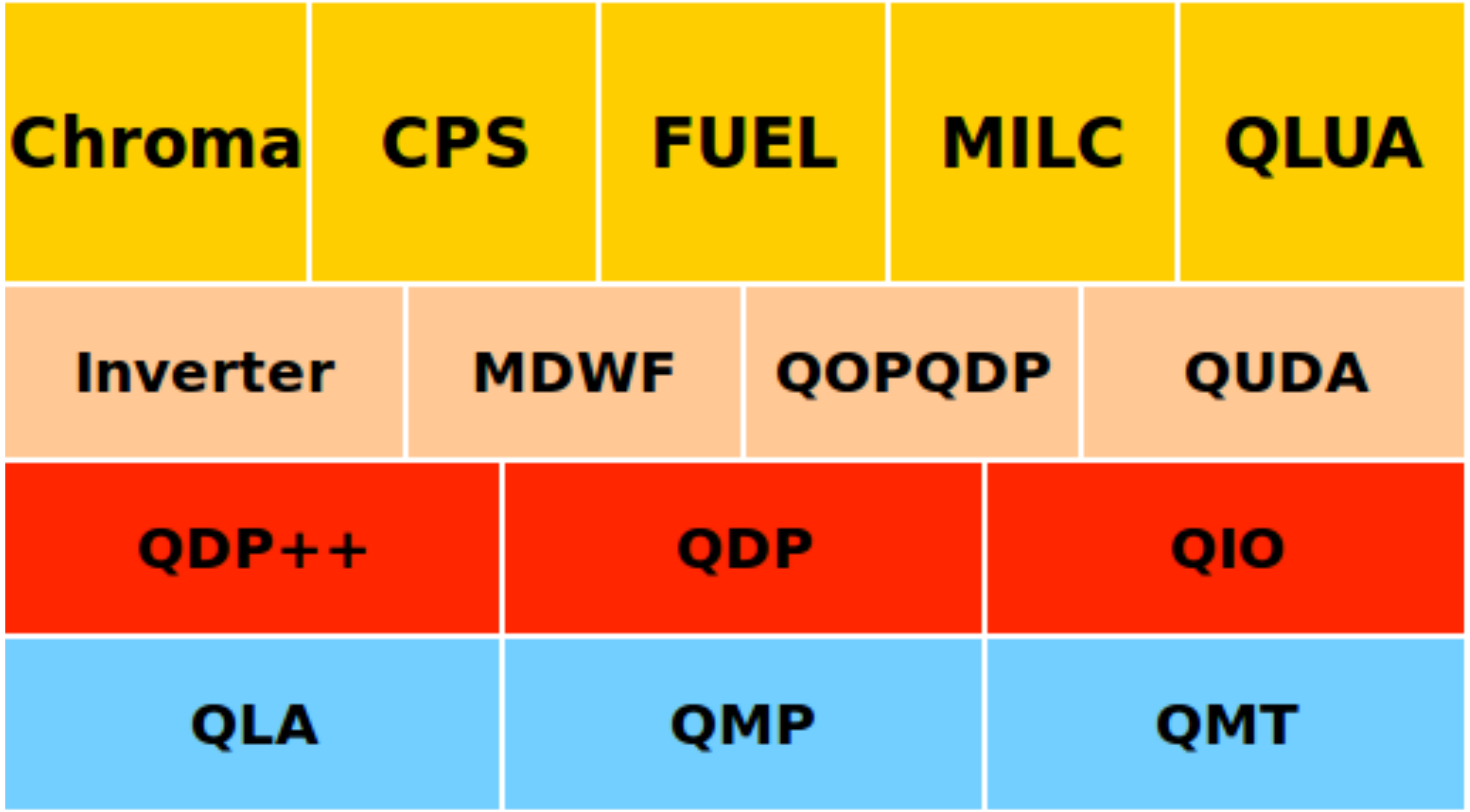
Lattice gauge theory

- Large 4D (5D) grid of small vectors/matrices with homogeneous stencil operations — large sparse linear algebra



USQCD software framework developed under DOE SciDAC

- Level 1: message passing and linear algebra
- Level 2: data parallel and IO
- Level 3: algorithms
- Top: applications
- Design tech circa 2001
  - Limited support: SIMD / threading
  - Multiple lattice sizes added later
  - Limited support for BSM



Software support for lattice BSM

- Superset of lattice QCD
  - Various gauge groups
  - Various fermion representations
  - Various dimensions
- Lattice SUSY
- Lattice gauge theory on curved space-time

QOPQDP & FUEL

- QOPQDP (and QDP/QLA)
  - Optimized for few-core, short-vector
  - Set of Perl code generators
- FUEL
  - Wraps QOPQDP/QDP in high level scripting language (Lua)
  - Very convenient for writing new code, experimenting with algorithms, etc.
  - Lose efficiency and flexibility of writing to lower level

- Optimized staggered fermion action
- Support different Nc
- Optimized on Blue Gene
- Calls QUDA for GPU acceleration
- Needs significant overhaul to make efficient use of future architectures while keeping high level, easy to use, scripting (-like) interface

Nim & QEX

- Nim offers extremely useful set of features
  - Extensive metaprogramming
  - Integrated build system (modules)
  - Simple, high-level “script-like” syntax
  - Seamless integration with C/C++ code, intrinsics, pragmas, etc.
  - Same performance as hand written C
- Developing Nim wrappers for external libraries (PRIMME, QUDA, QMP, QIO, MPI, OpenMP, SIMD intrinsics, LAPACK, BLAS, CUDA)
- Planning more wrappers/packages (Chroma, MILC, ...)

- QEX framework for lattice field theory
  - Data parallel library for tensor objects on a lattice including shifts, reductions
  - Mostly in Nim, with USQCD SciDAC C libraries
  - High level interface in development
  - Currently calls QUDA on GPU
  - https://github.com/jcosborn/qex
- Performance portability study: cudanim
  - Supports arrays on both CPU and GPU
  - https://github.com/jcosborn/cudanim

Janos (courtesy of Ricky Wong)

- Optimized for the Blue Gene Q
- Only depends on the XLC compiler, the mass_simd and SPI libraries
- Main contributors: Szabolcs Borsanyi and Chik Him (Ricky) Wong
- BSM applications: nearly conformal gauge theories, potential candidates for composite Higgs
  - Nf = 10, 12 in fundamental representation and Nf = 2 in two-index symmetric (sextet) representation
    - Beta functions of renormalized coupling in an attempt to identify the boundary of conformal window quantitatively and nonperturbatively
    - Hadron spectroscopy, particularly the emergent light 0++ scalar boson which may become Higgs boson impostor and heavier resonances in other channels which may become dark matter candidates and may be reachable by LHC

Incomplete list of other softwares developed/used for BSM

- QUDA
- Grid
- Chroma
- IroIro++
- Derivatives of MILC

The US Exascale Computing Project

EXASCALE COMPUTING PROJECT

- Focus on accelerating the delivery of a capable exascale computing ecosystem that delivers 50 times more computational science and data analytic application power than possible with DOE HPC systems such as Titan (ORNL) and Sequoia (LLNL)
- Goal: launch a US exascale ecosystem by 2021
- Entire exascale ecosystem: applications, system software, hardware technologies and architectures, along with critical workforce development.
- Collaborative effort under DOE – the Office of Science (DOE-SC) and the National Nuclear Security Administration (NNSA)

U.S. DEPARTMENT OF ENERGY | Office of Science

NNSA National Nuclear Security Administration

ECP focus areas

1. Application development

- Chemistry and Materials Applications
- Energy Applications
- Earth and Space Science Applications
- Data Analytics and Optimization Applications
- Co-Design

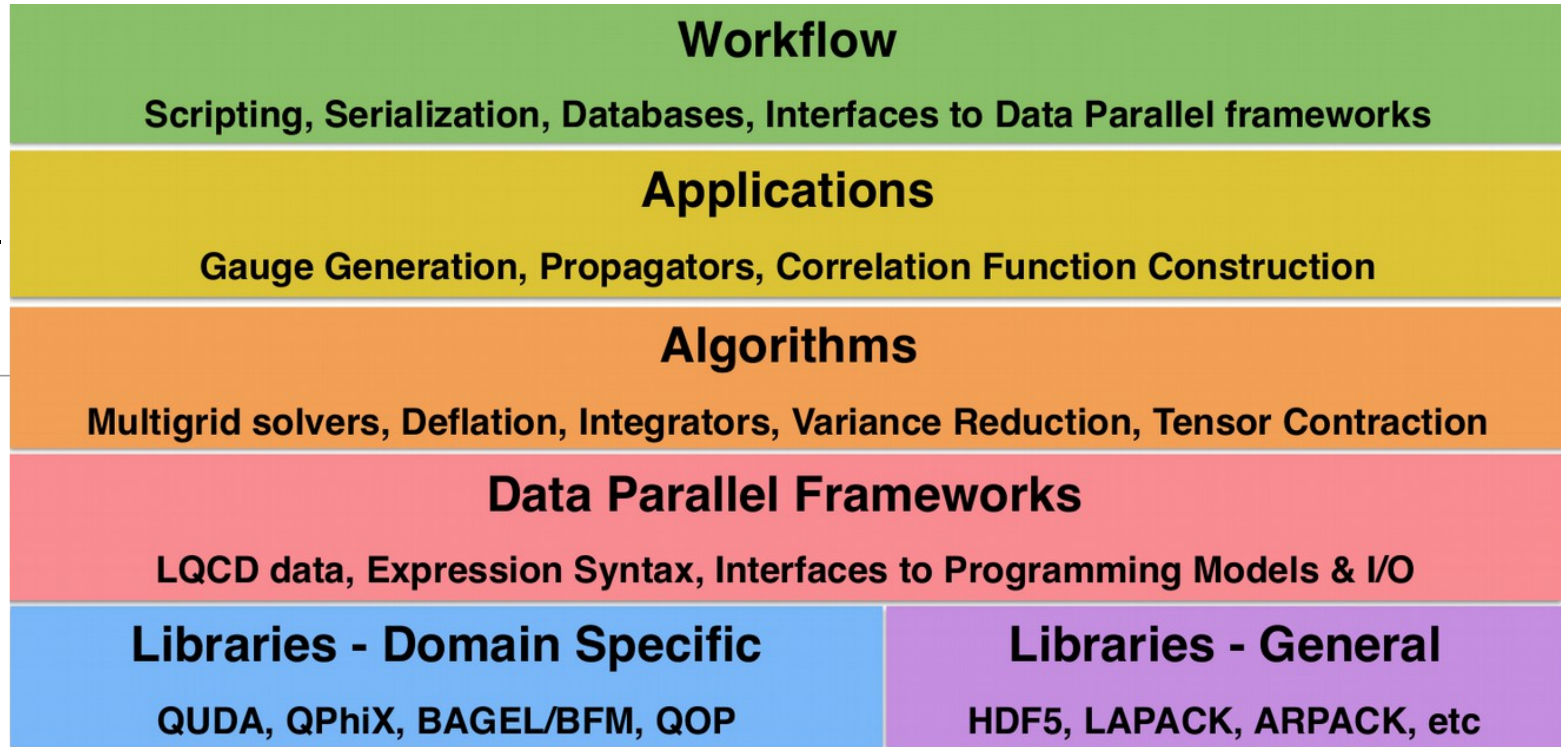
Lattice QCD

2. Software technology

- Programming Models and Runtimes
- Development Tools
- Mathematical Libraries
- Data and Visualization
- Software Ecosystem and Delivery

3. Hardware and integration

USQCD Exascale Computing Project



- Collaborative efforts
  - 4 DOE labs: ANL, BNL, Fermilab, Jefferson Lab
  - 7 university partners: Boston University, Columbia University, University of Illinois, Indiana University, Stony Brook University, University of Utah, William and Mary
- 4 Working Groups targeting different areas:
  - Workflow/Contractions (Lead:Robert Edwards, Jefferson Lab)
  - Critical Slowing Down (Lead: Norman Christ, Columbia University)
  - Linear Solvers (Lead: Richard Brower, Boston University)
  - Data-Parallel API (Lead: Carleton DeTar, University of Utah)

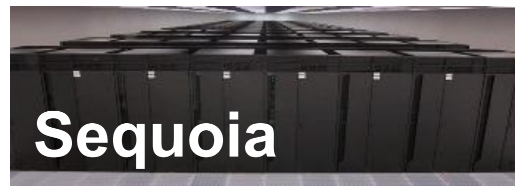
Current and future supercomputer systems

Pre-Exascale Systems

2013

Argonne
IBM BG/Q
Open

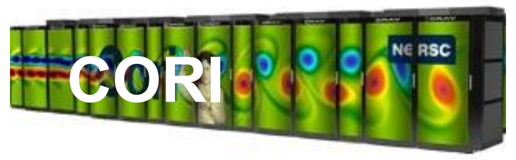
ORNL
Cray/NVidia K20
Open



LLNL
IBM BG/Q
Secure

2016

Argonne
Intel/Cray KNL
Open



LBNL
Cray/Intel Xeon/KNL
Unclassified

LANL/SNL
Cray/Intel Xeon/KNL
Secure

2018

ORNL
IBM/NVidia
P9/Volta
Open

LLNL
IBM/NVidia
P9/Volta
Secure

2020

NERSC-9

LBNL
TBD
Open

Crossroads

LANL/SNL
TBD
Secure

Exascale Systems

2021-2022

Argonne
Intel/Cray TBD
Open

Frontier

ORNL
TBD
Open

El Capitan

LLNL
TBD
Secure


AURORA
COMING SOON

AURORA 2021 ESP TIMELINE (NOTIONAL)

AURORA circa 2021 (1 of 3) - LOOPS

- Nodes will have both high single thread core performance and the ability to get exceptional performance when there is concurrency of modest scale in the code.

- The architecture is optimized to support codes with sections of fine grain concurrency (~100 lines of code in a FOR loop for example) separated by serial section of code. The degree of fine grain concurrency (number of iterations of loop for example) that will be needed to fully exploit the performance opportunities is moderate. In the ~1000 range for most applications.

- Independence of these loops is ideal but not required for correctness although dependencies that restrict the number of things that can be done in parallel will likely impact performance.

- There is no limit on the number of such loops and the overhead of starting and ending loops is very low.

AURORA circa 2021 (2 of 3)

- Serial code (within an MPI rank) will execute very efficiently and the ratio of the performance of the serial to parallel capabilities is a moderate ratio of around 10X, allowing for code that has not been entirely reworked to still perform well.

- OpenMP 5 will likely contain the constructs necessary to guide the compiler to get optimal performance.

- The compute performance of the nodes will rise in a manner similar to the memory bandwidth so the ratio of memory BW to compute performance will not be significantly different than systems were a few years ago. A bit better in fact than they have been recently.

AURORA circa 2021 (3 of 3) - HOMOGENEITY

- The memory capacity will not grow as fast as the compute performance so getting more performance through concurrency from the same capacity will be a key strategy to exploit the future architectures. While this capacity is not growing fast compared to current machines it will have the characteristic that the memory will all be high performance alleviating some of the concerns of managing multiple levels of memory and data movement explicitly.

- The memory in a node will be coherent and all compute will be first class citizens and will have equal access to all resources, memory and fabric etc.

- The fabric BW will be increasing similar to the compute performance for local communication patterns although global communication bandwidth will likely not increase as fast as compute performance.

Emergent technologies

- FPGA
  - Growing interest in using as HPC accelerators
  - Work being done at ANL (LCF & MCS)
    - Prototyping OpenMP frontend
    - Investigating reducing compile times
    - Exploring reduced precision
- Neuromorphic
  - Modeled on brain neurons
  - Mostly positioned for pattern matching, data processing could make way into HPC eventually
- Quantum Computing

Conclusions

- Non-perturbative BSM phenomenology needs lattice simulations
- Lattice BSM has aspects much more challenging than QCD
- Redesign software infrastructure for future machines and algorithms
- The US Exascale Computing Project provides opportunities for evolving our software ecosystem toward future architectures
- Lattice BSM community need to be prepared and take part in pushing simulations beyond standard model