Software Stewardship in SciDAC

Robert Edwards
Jefferson Lab

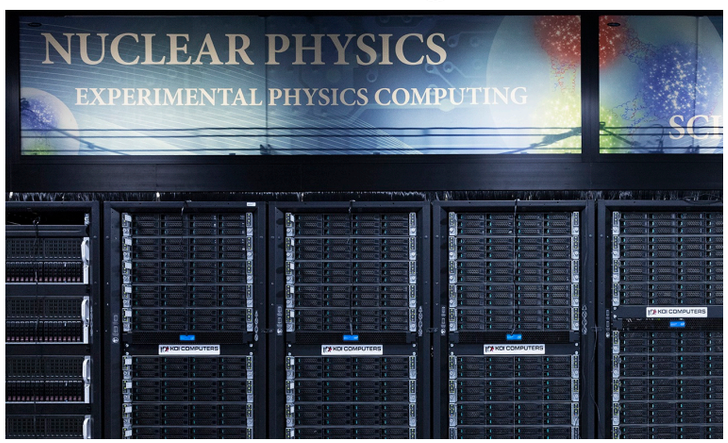
Who is Robert Edwards?

Principal Staff Scientist @ Jefferson Lab (since 1999)
Center for Theoretical and Computational Physics
PhD Physics @ New York U.
1998 Gordon Bell Prize (Price/performance) - QCDSP

PI: Nuclear & Particle Physics Lattice Computing Initiative
Currently, Chair of USQCD Collaboration (Lattice QCD)
PI: NP/ASCR SciDAC - "Fundamental nuclear physics at the exascale and beyond"

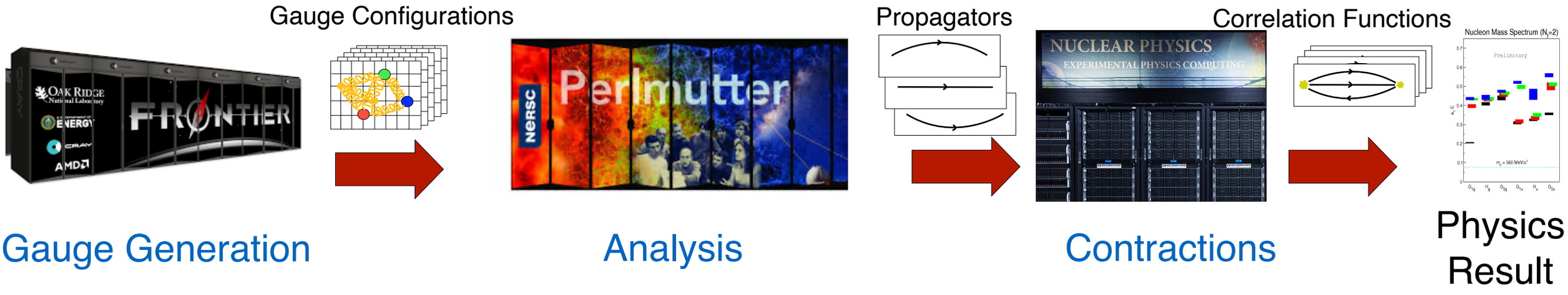
Quantum Chromo-Dynamics - theory of quarks/gluons/nuclei
Architect of "Chroma" - software system for Lattice QCD

HPC clusters @ JLab



Jefferson Lab

Monte Carlo methods for LQCD Workflow



- Gauge Generation: Few chains. Strong scaling challenge
  - Carried out on LCFs

- Analysis: Propagator Calculations
  - Many independent solves, throughput challenge

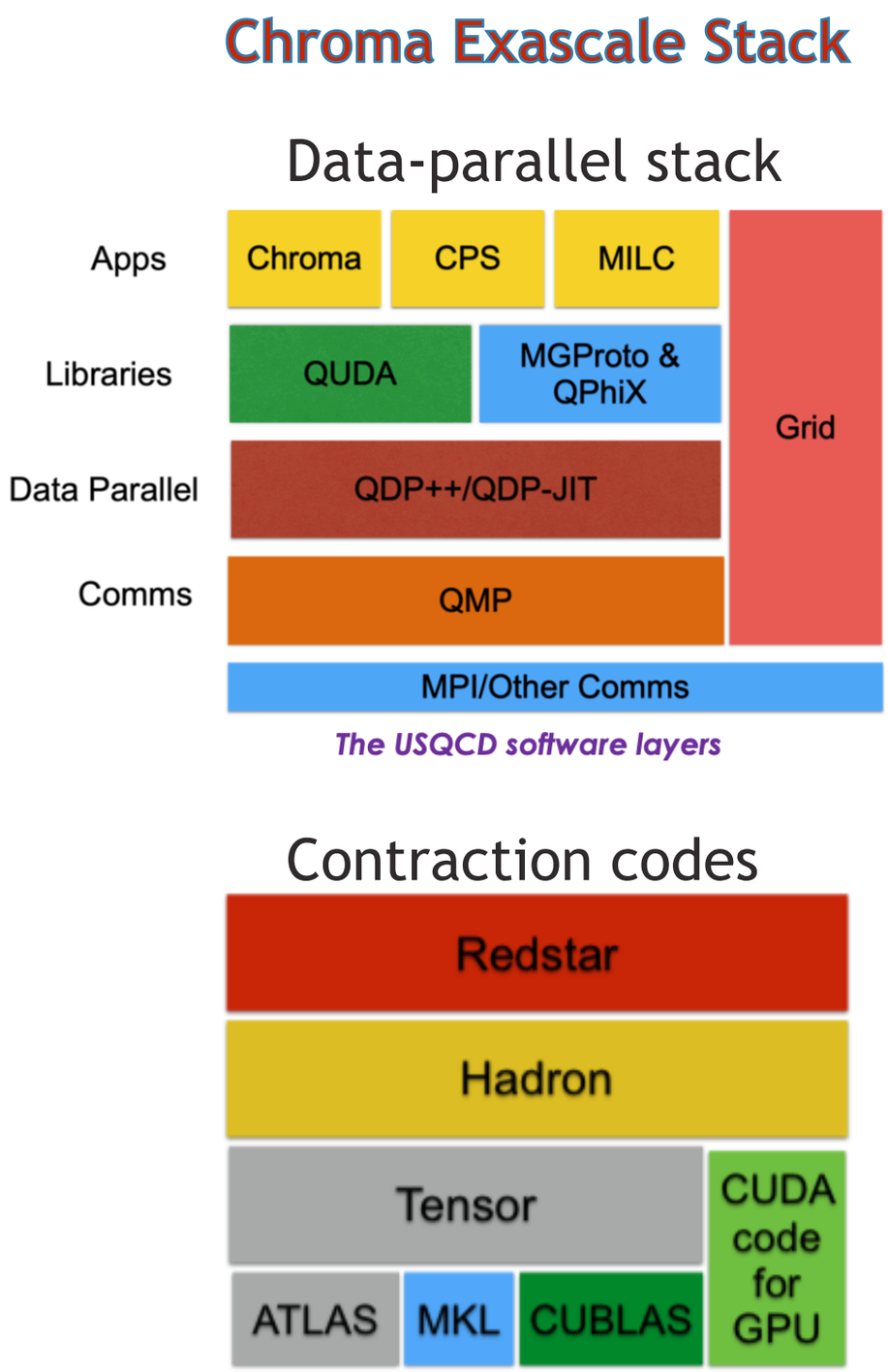
- Contractions: Correlation Function Construction
  - Capacity/midrange systems, tho LCFs can also be used - large memory footprint

Software stewardship challenges

- Chroma built on layered software infrastructure
  - QMP: message passing abstraction
  - QDP++: QCD data-parallel abstraction
    - Using Just-In-Time compilation framework
  - External solver libs - QUDA, QPhiX
  - Chroma - general application framework
    - Most physicist work at this level

- Need high performance at each stage of workflow
  - leadership/strong-scaling + high-throughput + single-node/graph optimization

- Must allow for high flexibility - algorithm development - to address new physics problems
  - Teaming with RAPIDS & FastMath



Most important aspects of software stewardship

- Performance
  - Adapting/optimizing for new varieties of GPUs and their API-s
  - GPU details only for the heartiest of soles (e.g., nowhere is CUDA/ROCm/SYCL exposed to users)

- Performance-Portability
  - Our SciDAC-5 have three main algorithm thrusts - each with challenges
  - **AI/ML** - all applications have required new development/reimplementations
  - **Multi-scale methods** - architecture disparities - e.g., a lot more CPU codes than physical GPUs

Third party software

it seemed like a good idea at the time…

- Third party software - “QUDA” and “LLVM” (compilers - not LLM-s…)

- Chose QUDA for high linear system solver performance on NVIDIA GPUs
  - Under ECP, GPU interface abstracted - now supports NVIDIA, AMD, Intel

- Has been well supported - however, all main developers now work for NVIDIA

- New codes supporting evolving algorithm requirements have provided a risk-mitigation strategy

Impact on SciDAC partnerships?

- SciDAC partnerships have been great!
  - Initial software stacks developed are used throughout the world. Lots of citations

- SciDAC partnerships have focused on algorithm research - important!
  - However, software maintenance/portability required

- Ways to increase partnerships? Bring in operations/facilities staff

Legacy of ECP

- Exascale Computing Project: **hugely impactful on our science (& other) programs**
- See many more science-domain applications on leadership systems (competition!)

- ECP: very computing facility focused. And there are many facilities!
  - Brought in new standards to improve software portability and sustainability
  - Programing methods, package management, scaling, rigorous metrics

- Co-design efforts - involved applications (and ours benefitted greatly)
  - Application porting/development benefitted both the science/facility/hardware teams
  - My personal view - these interactions were the single largest impact on our program

Recent NP/ASCR workshop - June 2024

- Workshop on Software Infrastructure for Advanced Nuclear Physics Computing
  - Brought together nuclear theorists, experimentalists, computing and ASCR communities
- White paper will be forthcoming. High level summary of observations:

- SciDAC, CSSI and ECP - successfully fostered software innovation. Support of these & similar programs should be continued and expanded.

- **New initiatives in software stewardship** are needed to take advantage of hardware innovations, industry trends and best practices for the development of a sustainable software infrastructure for nuclear physics.

NP/ASCR workshop - full-text summary

- SciDAC, CSSI and ECP have successfully fostered software innovation. Support of these and similar programs should be continued and expanded. New programs in emerging areas, such as AI/ML and QIST are needed to build on their success to foster collaborations between NP scientists with others, including data and computational scientists and statisticians, to further advance the goals of the NSAC LRP.
- To extend the scientific impact of these investments, new initiatives in software stewardship are needed to take advantage of hardware innovations, industry trends and best practices for the development of a sustainable software infrastructure for nuclear physics.
- Experiments require innovation to exploit their capabilities, which often needs R&D beyond project and operations scope; support for such effort is needed. Experiments have been drivers for common software and common software ecosystems. This should be fostered and strengthened.

Blank

Questions

- Please introduce yourself and say a few words about the software you represent, if there is any.
- What are some of the challenges you face with regard to software stewardship?
- What aspects of software stewardship are most important to you?
- Are you using any third party software? If so, what criteria led you to choose this particular software product? Did software sustainability play an important role?
- What is the impact of the scientific software stack on SciDAC partnerships? What are ways to assess and demonstrate this impact? What are ways to increase it?
- What is the impact of the scientific software stack software on portability and performance of the application software?

More questions

- ECP had a strong emphasis on software integration and software adoption. As a result, many applications were restructured or rewritten to leverage the ECP software stack. Is SciDAC the right venue to maintain and move forward these integrations? How do we establish new ones? What is your vision for the future of application - scientific software integration?
- What do you see as the most significant hindrances for software sustainability in the next few years?
- What are some issues in the HPC software community that are not being sufficiently addressed right now that need to be considered to better address your important requirements?