Argonne NATIONAL LABORATORY

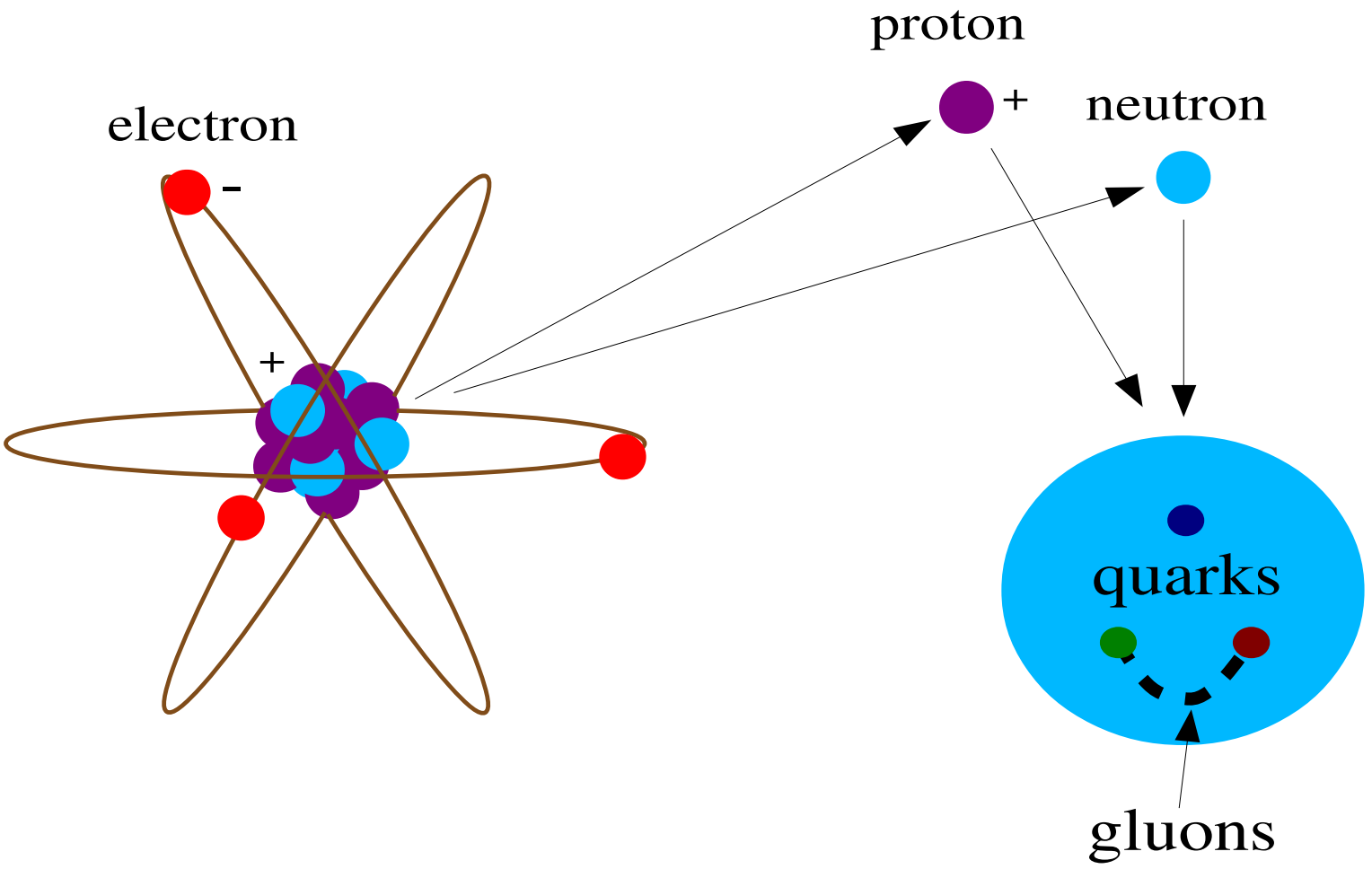
QPX for Lattice QCD

James C. Osborn
Argonne Leadership Computing Facility

Mira Boot Camp
May 20, 2015

U.S. DEPARTMENT OF ENERGY

Subatomic particles

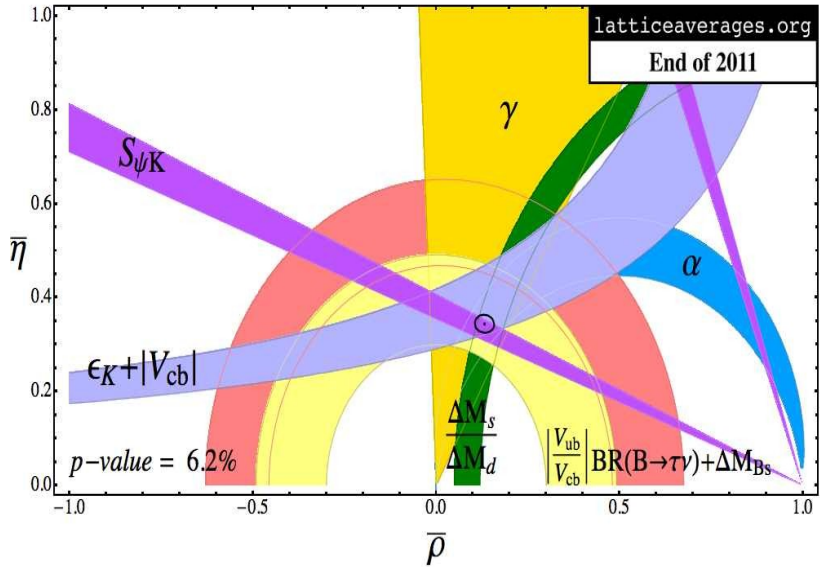


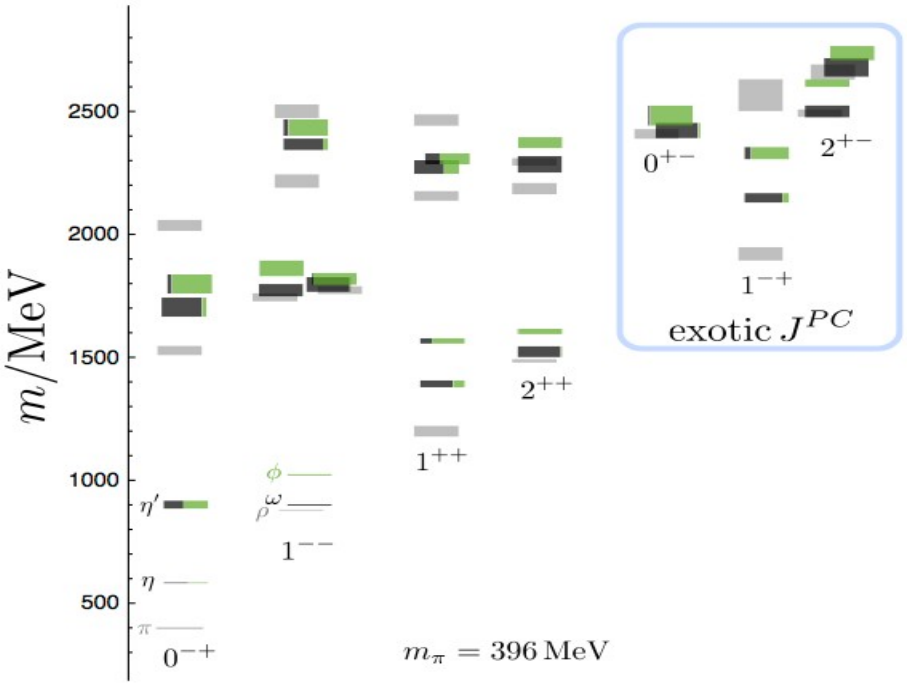
Quantum Chromodynamics (QCD)

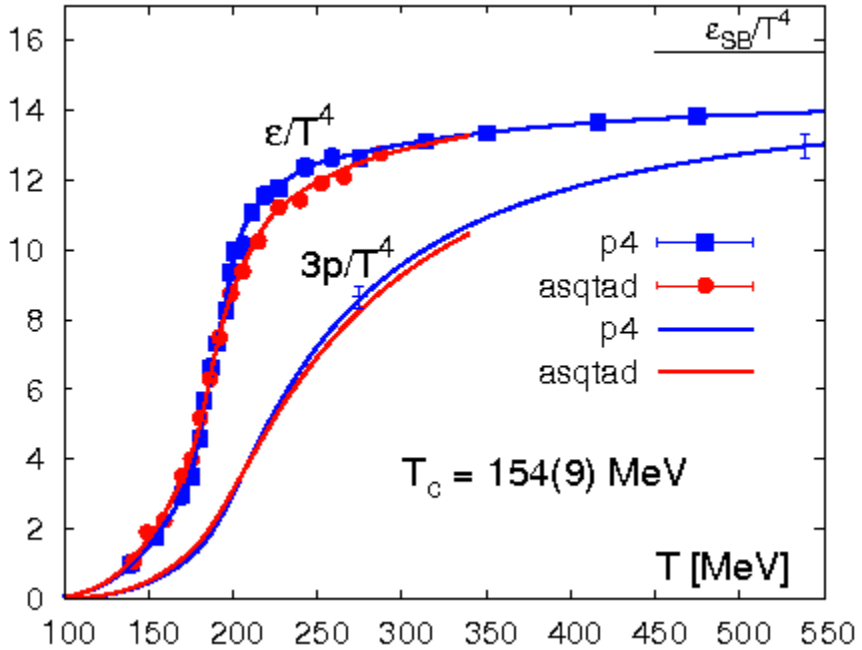
- fundamental theory of strong nuclear force
- part of the Standard Model of particle physics (with electromagnetism and weak nuclear force)
- strongly interacting (at low energies, i.e. nuclei)
  - can't reliably calculate quantities analytically
  - must resort to computer simulations
- requires large scale resources to perform calculations to desired accuracy

Major goals

- Precision tests of Standard Model and searches for new physics
- Understand matter at high temperatures and densities
- Determine properties of particles (structure of proton, etc.)

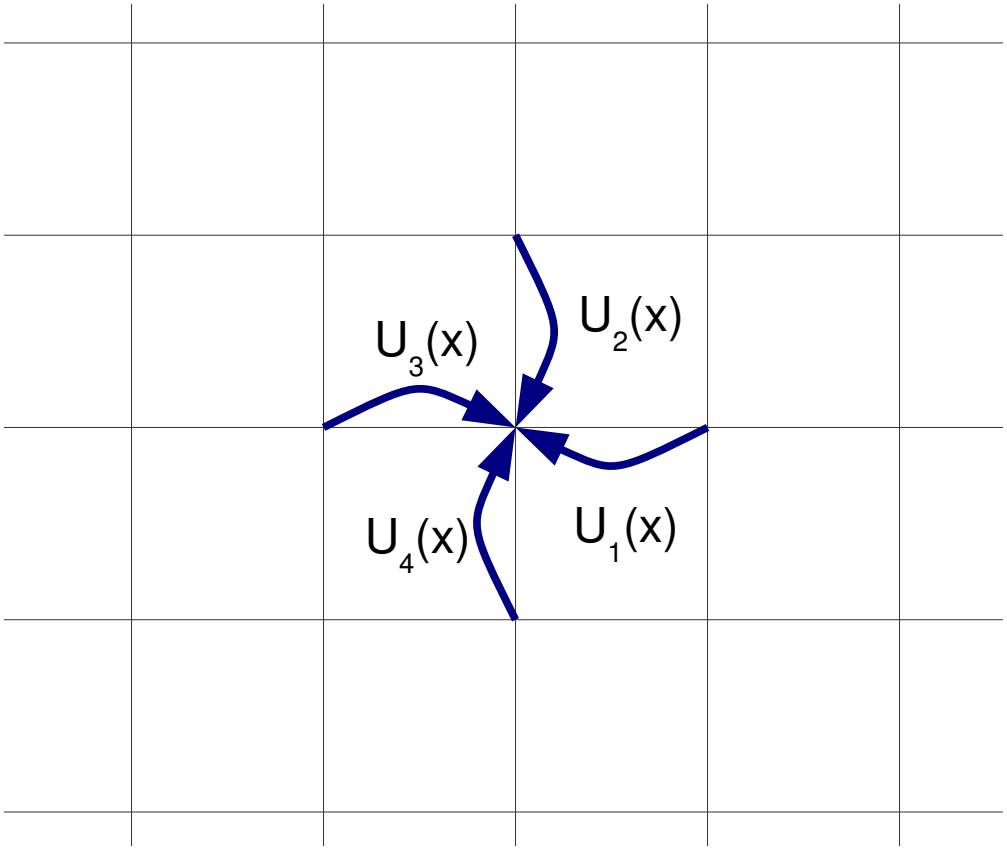






Dirac equation

- Main expense is solving Dirac equation (relativistic Schrödinger equation)
- First order PDE in 4 dimensions
- Quarks represented by length 3 complex vectors on sites
- Gluon fields are 3x3 complex matrices on links
- Simplest discretization: gather quark fields from neighbors, multiply by gluons along links



QCD kernel

- Calculate

```
for(i=0; i<N; i++) x[i] += u[i] * (*y[i]);
```

- u[i] 3x3 complex matrix (18 reals)
- x[i], *y[i] length 3 complex vector (6 reals)

- QPX length 4 vector
  - Unroll loop by 2

QCD kernel unrolled

- Keep results contiguous in vector register
- Start with

  {x[i](0),x[i](1)} += {u[i](0,0),u[i](1,0)} * {y[i](0),y[i](0)}

  vec_ld() vec_ld()+vec_perm() vec_ld2()

Complex fused multiply add

- QPX has good support for complex arithmetic

  z=a*b+c: z=vec_xmadd(a,b,c); z=vec_xxnpmadd(b,a,z)
  z = c + real(a)*b z += i*imag(a)*b

- Second operation has dependency on first
- Can remove with one extra add at end of dependency chain

```
z1 = vec_xmul(a1,b1);  z2 = vec_xxnpmadd(b1,a1,c)
z1 = vec_xmadd(a2,b2,z1);  z2 = vec_xxnpmadd(b2,a2,z2)
z1 = vec_xmadd(a3,b3,z1);  z2 = vec_xxnpmadd(b3,a3,z2)
z = vec_add(z1,z2)
```

- Not necessary here if running with 2 or more threads/ranks per core

Other issues

- Ensure data is properly aligned: 16 (float) or 32 (double) bytes
  - vec_ld assumes data is aligned, will round address down if not
  - Use vec_lda if unsure (will generate SIGBUS if not aligned)
- Declare absence of aliasing (if true)
  - C99 restrict keyword
  - Xlc: “#pragma disjoint(a,b)”
- Round results before storing in single precision (if needed)
  - Storing double precision vector registers to float array truncates
  - Need to explicitly round to single (vec_rsp)

Summary

- Good news
  - QPX has rich set of operations (load,store,permute,(x)madd)
  - Can do even complicated arithmetic patterns with relatively low overhead for additional rearrangement of data
  - Can get good performance without need to reorder data
- Bad news
  - Requires care with alignment, aliasing, dependencies (not too bad, but requires work)
  - Vector units getting bigger (8-16 wide)
  - Array of Structures layouts will become more difficult to deal with in future
- BG/Q makes a good platform for transitioning code from past styles (AoS, MPI-only) to future coding styles (SoA, MPI+OpenMP)
  - Legacy code still works fine (possibly with some optimization effort) can still see benefit from redesigning code

Performance on BG/Q
full Dirac solver
512 nodes, single precision