A Numerical QCD "Hello World"

Bálint Joó

Thomas Jefferson National Accelerator Facility
Newport News, VA, USA

INT Summer School On Lattice QCD, 2007

Lattice Calculation Basics

What is involved in a Lattice Calculation

What is a lattice simulation/calculation ?

- Goal: evaluate path integral

$$\langle \mathcal{O} \rangle = \int \mathcal{D}U \, \mathcal{O}(U) \, P_{\rm eq}(U)$$

  - $\mathcal{O}$ is/are the observable(s) of interest
  - $\mathcal{D}U$ is the measure over the gauge fields
  - $P_{\rm eq}$ is the path integral propability distribution

$$P_{\rm eq} = \frac{1}{\mathcal{Z}} e^{-S(U)}$$

  - $S(U)$ is the *action* of the theory

"Theoretical" Lattice Recipe

- Move to a Lattice with lattice spacing a
  - coordinates x become discrete i.e: $x = (n_1, n_2, n_3, n_4)$ in 4D.
  - Gauge fields get bound to lattice *links*
    - Denote as $U_{x,\mu}$, μ specifies link direction.
  - Latticize Measure: $\mathcal{D}U \rightarrow \prod_{x,\mu} dU_{x,\mu}$
  - Latticize Action: $S \rightarrow S_{\text{latt}}$
  - Latticize Observables: $\mathcal{O} \rightarrow \mathcal{O}_{\text{latt}}$

  So

$$\langle \mathcal{O}_{\text{latt}} \rangle_a = \int \prod_{x,\mu} dU_{x,\mu} \; \mathcal{O}_{\text{latt}}(U_{x,\mu}) \; P^{\text{latt}}_{\text{eq}}(U_{x,\mu})$$

  with

$$P^{\text{latt}}_{\text{eq}}(U_{x,\mu}) = \frac{1}{\mathcal{Z}_{\text{latt}}} e^{-S_{\text{latt}}(U_{x,\mu})} \qquad \mathcal{Z}_{\text{latt}} = \int dU_{x,\mu} \prod_{x,\mu} e^{-S_{\text{latt}}(U_{x,\mu})}$$

- *For $\langle \mathcal{O} \rangle$ we need limit as $a \rightarrow 0$: continuum extrapolation*

Practialities to the Recipe

- First:

$$\langle \mathcal{O}_{\rm latt} \rangle_a = \int \prod_{x,\mu} dU_{x,\mu}\ \mathcal{O}_{\rm latt}(U_{x,\mu})\ P^{\rm latt}_{\rm eq}(U_{x,\mu})$$

is still infinite dimensional (infinite lattice). Move to finite volume to fit on a computer:

$$\langle \mathcal{O}_{\rm latt} \rangle_{a,V} = \int \prod_{x,\mu}^{V} dU_{x,\mu}\ \mathcal{O}_{\rm latt}(U_{x,\mu})\ P^{\rm latt}_{\rm eq}(U_{x,\mu})$$

  - *Need infinite volume limit*
  - *Need to beware of finite volume effects*

Practialities to the Recipe

- Secondly: $\langle \mathcal{O} \rangle_{a,V}$ is still very high dimensional
- Turn to "Monte Carlo" methods:

$$\int \prod_{x,\mu}^{V} dU_{x,\mu}\, \mathcal{O}_{\rm latt}(U_{x,\mu})\, P_{\rm eq}^{\rm latt}(U_{x,\mu}) \rightarrow \sum_{\{U^i\}} \mathcal{O}_{\rm latt}(U^i_{x,\mu})\, P_{\rm eq}^{\rm latt}(U^i_{x,\mu})$$

  - U^i is called a *configuration*
  - $\{U^i\}$ is called an *ensemble*
  - Monte Carlo integral has a *statistical error*
  - The *statistical* error typically decreases as:

$$\epsilon \approx \frac{1}{\sqrt{N_U}}$$

where N_U is the number of *independent* configurations in an ensemble

Annoyances to the Recipe

- Some lattice formulations don't preserve desired symmetries
  - e.g: Chiral Symmetry in Wilson like fermions
- It is often not possible to work at the desired physical parameters: eg: at the physical quark masses
- Thus we may need to evaluate $\mathcal{O}_{\mathrm{latt}}$ in
  - *several ensembles* at various physical couplings
  - Take appropriate limits (e.g: chiral limit)

Complete Programme

- Generate ensembles of configurations $\left\{U^i\right\}_{a_i, V_i, c_i}$
  - various physical couplings c_i, volumes V_i, latt. spacings a_i
  - This step is numerically most costly and needs *supercomputers*
- Compute $\mathcal{O}_{\text{latt}}$ on the configurations in the ensembles.
  - Typically this phase involves computing correlation functions.
  - Depending on what $\mathcal{O}_{\text{latt}}$ is, this can be moderately numerically costly to numerically cheap. This step needs *supercomputers or clusters*
- Analysis I: Evaluate the path integrals:
  - This involves fitting $\mathcal{O}$ to phenomenological forms.
  - Typically this step needs *workstations* but times are changing...
- Analysis II: Take all the appropriate limits, quantify all errors.

Errors

- *Statistical* : from the evaluation of path integrals
- *Systematic* : from the method
  - *Discretization* : from the finite lattice spacing a
  - *Finite Volume* : from the finite box
  - *Numerical* : Precision of code, subjectivity of fit
- Try to control / quantify these. Eg:
  - Use a formulation which reduces discretization error
  - Work in a big enough box
  - Have lots of configurations
  - Try to get same answer with different methods

Where is the Physics

The physics goes into 3 main places:

- How we construct the lattice action
- How we construct the observables (probes)
- How we extract the result (phenomenological forms)

Each one has computational ramifications.

Example: The Wilson Gauge action

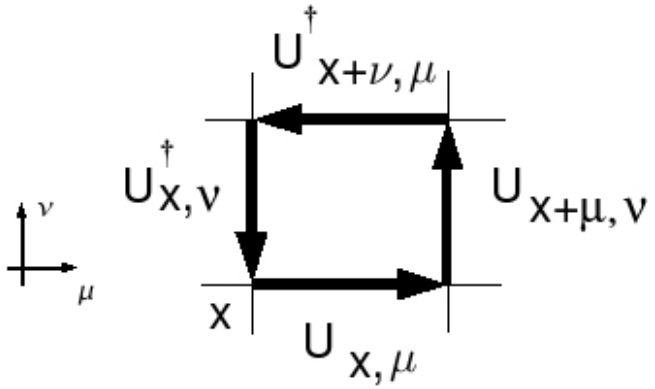
- Our continuum action with a bare coupling g_0

$$S_{\text{gauge}} = \frac{1}{4g_0^2} F_{\mu,\nu} F_{\mu,\nu}$$

- This can be expressed through Wilson loops
- One the lattice Wilson Loops can be constructed by taking the trace of the products of gauge fields over closed paths
- In particular, the *Wilson Gauge Action* is:

$$S^{\text{latt}} = \beta \sum_x \sum_{\mu \neq \nu} \frac{1}{2N_c} \left(\text{Tr} U_{\mu\nu}(x) - U^{\dagger}_{\mu\nu}(x) \right)$$

- $U_{\mu\nu}(x)$ is the product around an elementary "plaquette" at site x in the $\mu\nu$ plane



- The plaquette is:

$$\begin{aligned} U_{\mu\nu}(x) &= U_{x,\mu} U_{x+\hat{\mu},\nu} U_{x+\hat{\mu}+\hat{\nu},-\mu} U_{x+\hat{\nu},-\nu} \\ &= U_{x,\mu} U_{x+\hat{\mu},\nu} U^{\dagger}_{x+\hat{\nu},\mu} U^{\dagger}_{x,\nu}(x) \end{aligned}$$

where we use that $U_{x,\mu}$ are unitary so

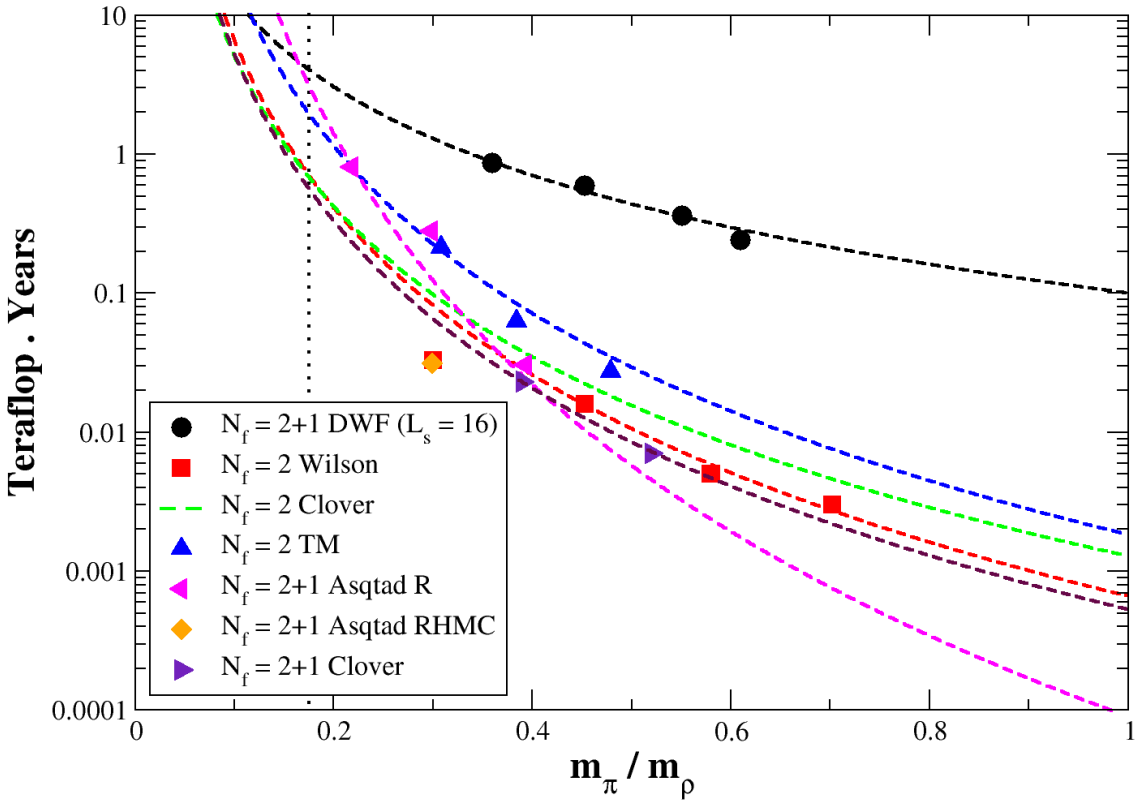
$$U_{x+\hat{mu},-\mu} = U^{-1}_{x,\mu} = U^{\dagger}_{x,\mu}$$

- $\beta = \frac{2N_c}{g^2}$ is lattice version of the coupling
- This action is has discretisation errors of $O(a^2)$
- More elaborate formulations involving bigger loops have smaller discretisation errors

Fermions

- A subject of its own. Different formulations sacrifice different properties:
  - Wilsonesque Fermions (Wilson, Clover, Twisted Mass)
    - sacrifice chiral symmetry, possible flavor symmetry (TM)
    - $O(a)$ (Wilson), $O(a^2)$ (Clover, TM) errors
  - AsqTAD Fermions (and other Improved Staggered)
    - Sacrifice flavour symmetry, retain $U(1)$ symmetry
    - $O(a^2)$ errors
  - Chiral Fermions (eg: Domain Wall, Overlap)
    - maintain chiral symmetry arbitrarily accurately
    - Sacrifice 4D transfer matrix
    - $O(a^2)$ discretisation errors
- Common Feature:
  - Computational cost explodes as quark mass approaches physical value

Computational Cost of Fermions



Cost to generate 1000 independent gauge configurations in Teraflop Years

(from Mike Clark, Lattice 2006 proceedings, arXiv:hep-lat/0610048)

Currently only parallel computers can deliver Teraflop scale computing

A Parallel Computer: The (currently) 110Tflop Cray XT3 at Oak Ridge National Laboratory, Oak Ridge, Tennesee

Complete Big Picture

A credible lattice calculation is a *formidable* undertaking

- Requires:
  - Large amount (Teraflops) of computer time (Politics)
  - Effective collaboration at the various levels (Management)
  - Technical Know How at various levels (Physics, Algorithms, Code Development and Porting, Engineering, Analysis)
  - Infrastructure: Hardware, Software, Grids, Tapes, etc
- Tendencies:
  - Large Collaborations (LHPC, MILC, UKQCD, ETMC etc)
  - Multi-year planned data production runs
  - Inter Collaboration Collaborations are now appearing e.g: USQCD, USQCD-UKQCD collaborations
  - Emergence of “Infrastructure Groups”
    - provide software/hardware for you (eg: USQCD Nat. Fac.)
    - provide services/data for you (eg: ILDG: LDG, DiGS, LDG, CSSM, JLDG)

Basics of Parallel Computing

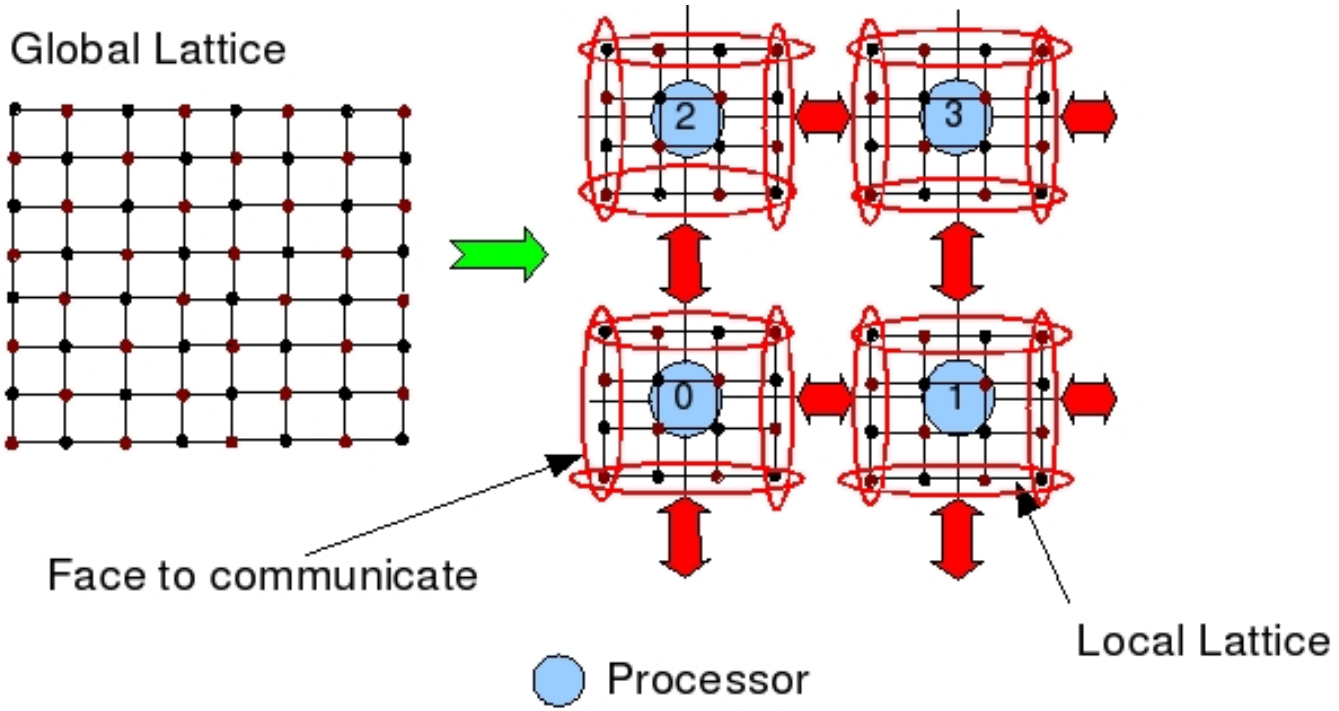
- Tasks that don't depend on each other can be done simultaneously
- Types of parallelism in problems:
  - Embarassing/Comfortable: Tasks completely independent
    - Can make effective use of a collection of independent PCs
  - Closely coupled: tasks exchange information frequently (eg: share data)
    - Efficient information exchange needed: Shared memory / Network
    - eg: PC Cluster machines (with network), Supercomputers
  - Loosely coupled: tasks exchange information infrequently
    - Speed of information exchange not critical, use internet etc.
    - eg: managing a large collection of jobs on a Grid.
- Lattice QCD is closely coupled

Recent Trends in Hardware

- Massively Parallel Systems (MPP): Contain O(10000) processing elements (PEs)
- Message Passing between PEs
  - Fast Custom Networks (BG/L, QCDOC, Cray XT3/4, APE)
  - Commodity Networks on Clusters (eg infiniband)
- Multi-Socket/Multi-Core PEs
  - QCDOC and BG/L have 2 processors per node card
  - Cray and Clusters employ multi-core chips (Intel, AMD)
- Some amount of vectorization on PEs
  - BG/L has “double hummer” FPU - 2 FPUs in one
  - Cray and Clusters have SSE, SSE2, SSE3 instructions

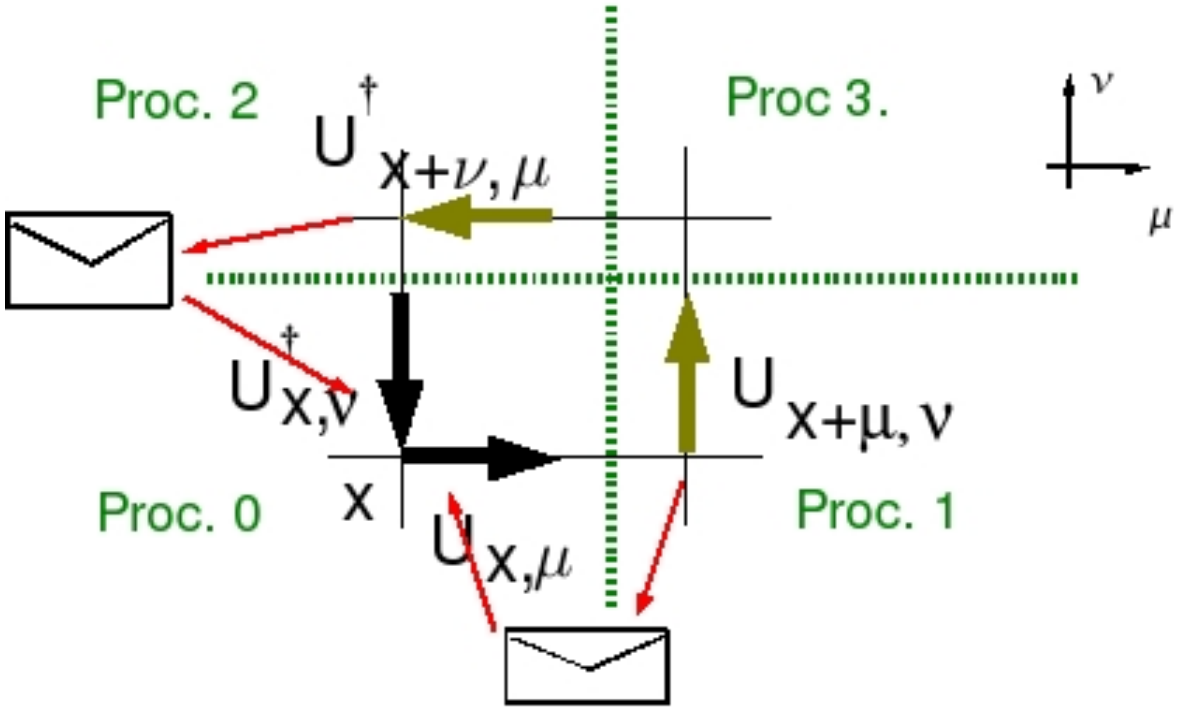
A Useful Model Computer

- The processing elements form a grid
- Each processor can communicate with neighbours



- Some machines are built like this (QCDOC, BG/L)
- Can be implemented “virtually” on machines with richer connectivity or shared memory.

Message Passing



- For plaquette:
  - $U_{x+\hat{\mu}}$ is put in message (proc 1 to 0)
  - $U^{\dagger}_{x+\hat{\nu}}$ is put in message (proc 2 to 0)

Collective Operations

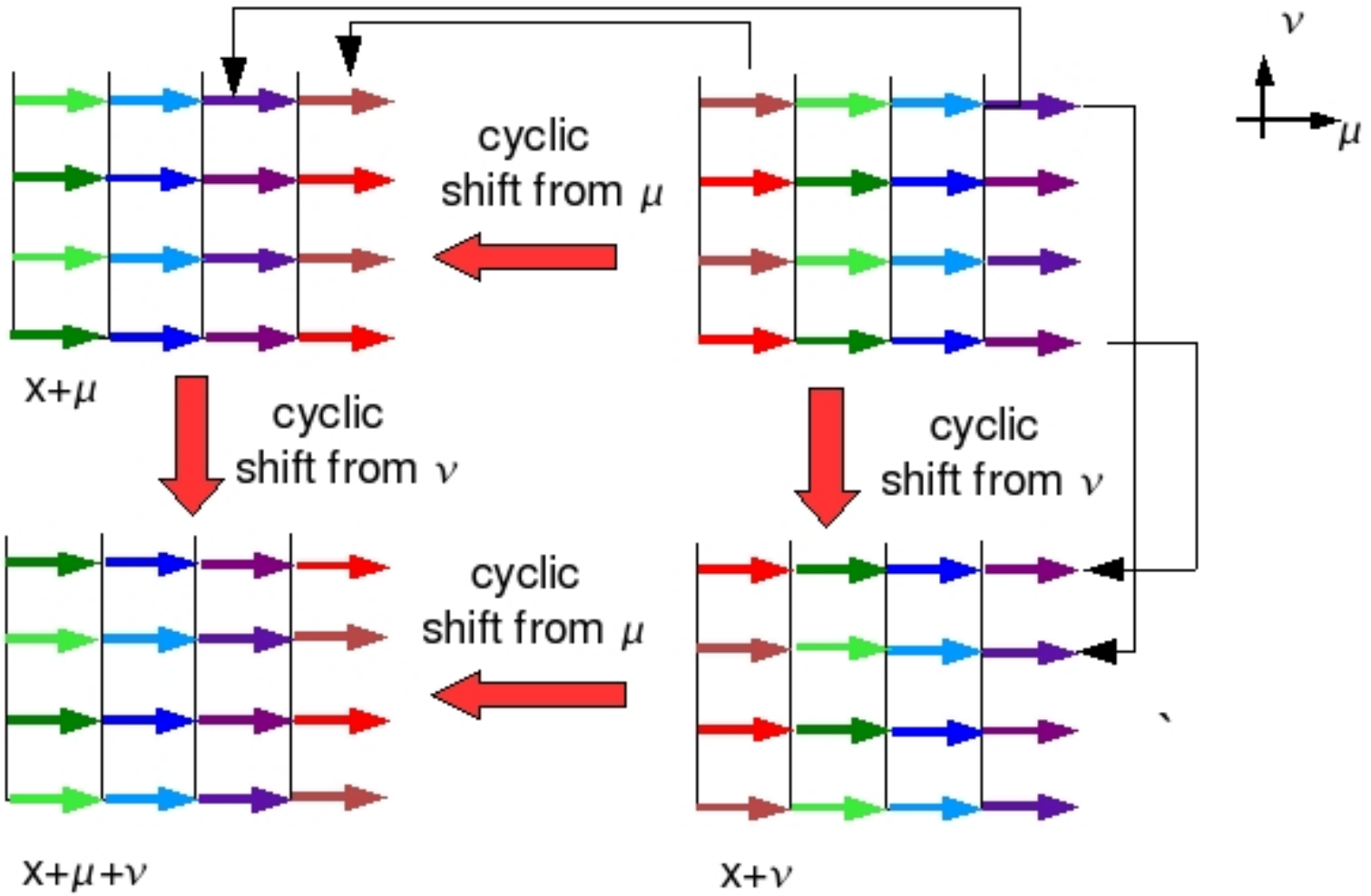
- Collective operations are called by all PEs
- There are the following kinds:
  - Local collectives: each node gets own answer
  - Gathers: one (some) gets answer from many
  - Scatters: many gets answer from one (some)
    - Broadcasts: one node sends to all
  - All to all: all get answers from all
    - Reductions: eg global sum, min, max

- The Message Passing Interface (MPI)
  - *The* International Message Passing Standard
  - Rich Data Model
  - Many different ways to pass messages
  - Quite complex
  - http://www-unix.mcs.anl.gov/mpi
- The QCD Message Passing (QMP) Interface
  - Designed by USQCD SciDAC software committee
  - Simple data model
  - Asynchronous Sends Only
  - Relatively easy to implement/use:
    - over MPI
    - over custom networks (QCDOC, GigE mesh)
    - http://usqcd.jlab.org/usqcd-docs/qmp

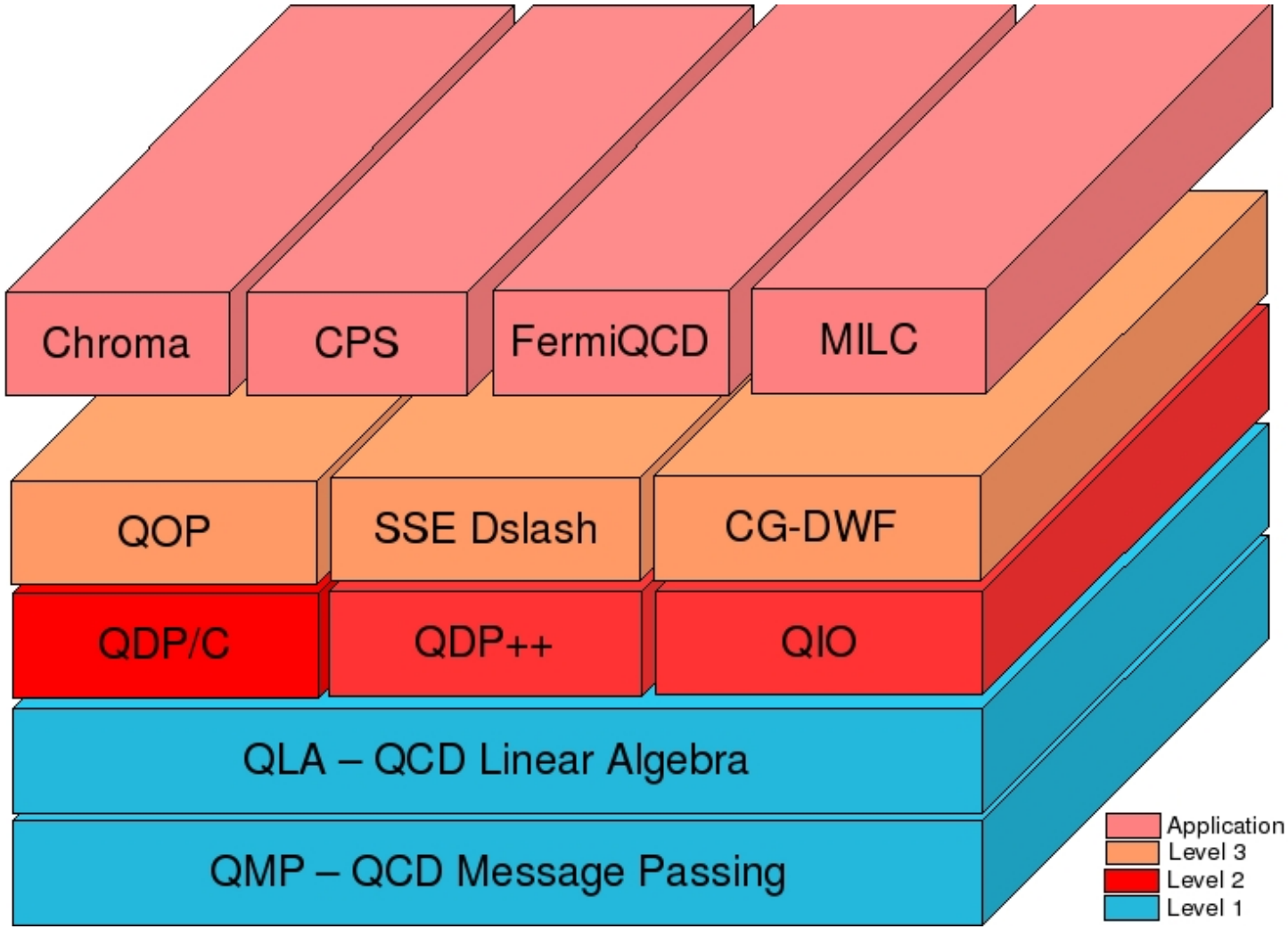
Data Parallelism

- Convenient programming model
- Everything is collective
- “Shift Lattice” to get at neighbours
- “Global” fill operations
- Limited by “masks”
- Try not to refer to an individual site
- Doesn’t feel really parallel at all (Good!)
- Similar to CM-Fortran, HPF, F90

How Shifts Work



The SciDAC software stack for Lattice QCD



Rest of tutorial

- For the HPC sections
  - We will work with a data parallel framework
  - We will use a freely available library: QDP++
- For the last lecture (Analysis)
  - We will use some *real* and *recent* data
- Sadly I don't have time to cover Chroma
  - But most of the QDP++ examples are taken from chroma
  - After the tutorial you should find chroma code straightforward

Summary Of Lecture

- I discussed the gross details of a lattice calculation
- I discussed aspects of parallelism
- Now: Let's write some code