

QLUA

Andrew Pochinsky

MIT

The overarching goal is to develop a platform-independent computational platform for lattice QCD that makes efficient use of both computational scientist's time and evolving hardware on the path to exascale

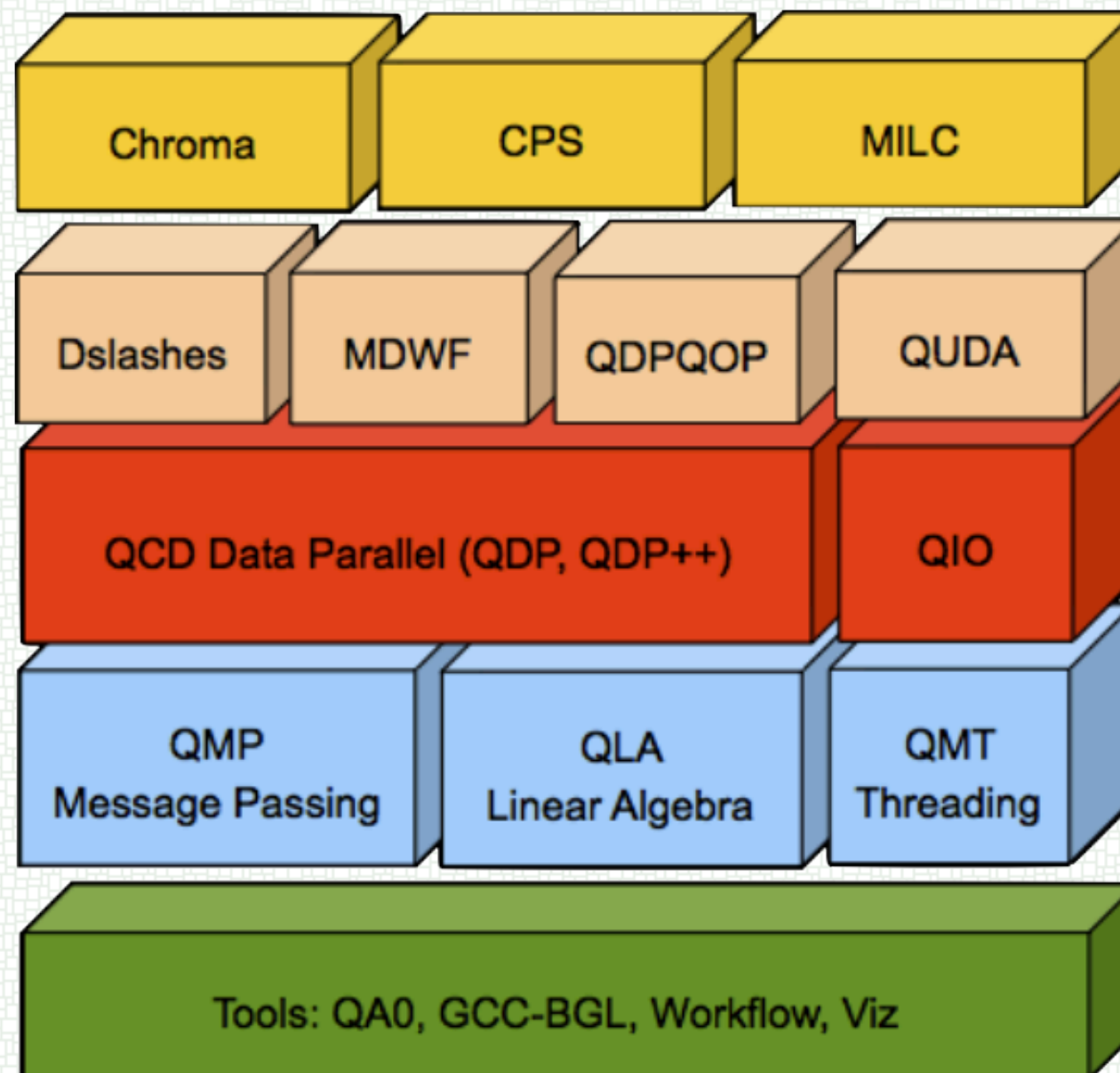
LATTICE QCD

$$\langle \mathcal{O} \rangle = \int [\mathcal{D}U] \det M[U] \exp(-S_F[U]) \mathcal{O}[U]$$

$$\mathcal{O}[U] = \sum P(U) (M^{-1})_{ij} \dots$$

$$p[U] = \det M[U] \exp(-S_F[U])$$

USQCD SOFTWARE



HPC HARDWARE

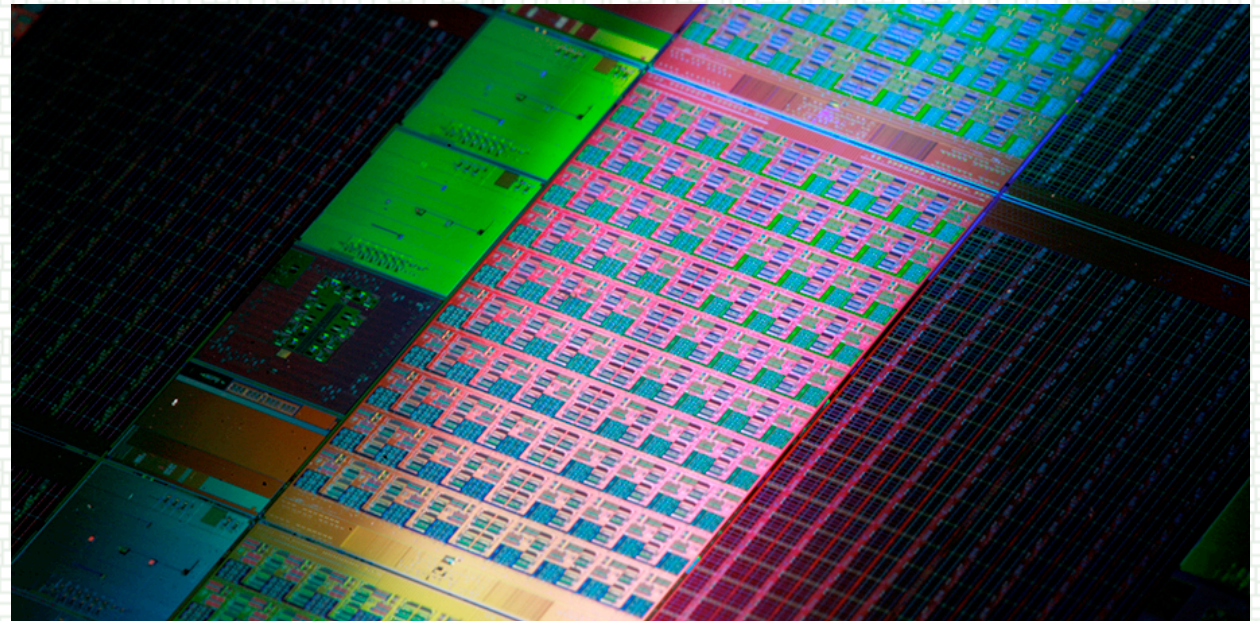
IBM: BLUE GENE

- ☐ /L, /P, /Q, ...
- ☐ short Vector FPU
- ☐ multiple cores
- ☐ torus network



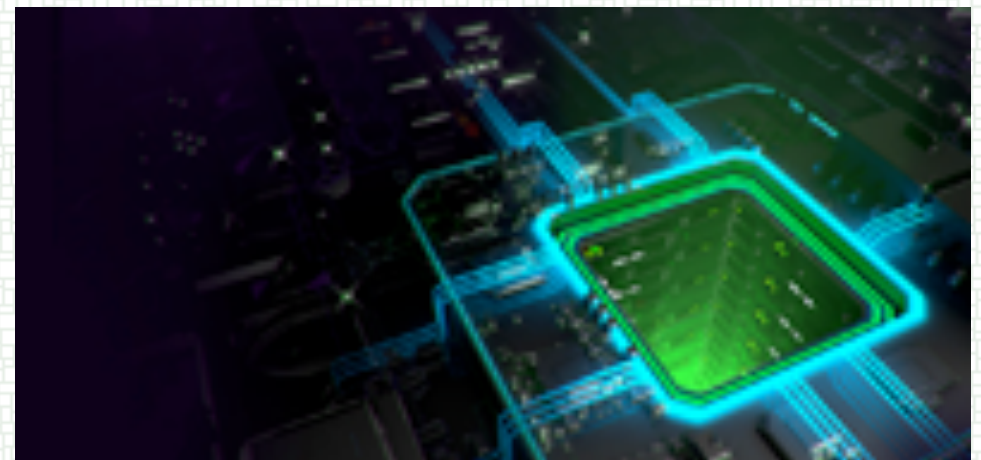
INTEL: MIC ARCHITECTURE

- ☐ x86
- ☐ AVX
- ☐ multiple cores
- ☐ 3rd party network



NVIDIA: GPU

- ☐ Many cores
- ☐ accelerator unit
- ☐ required a host

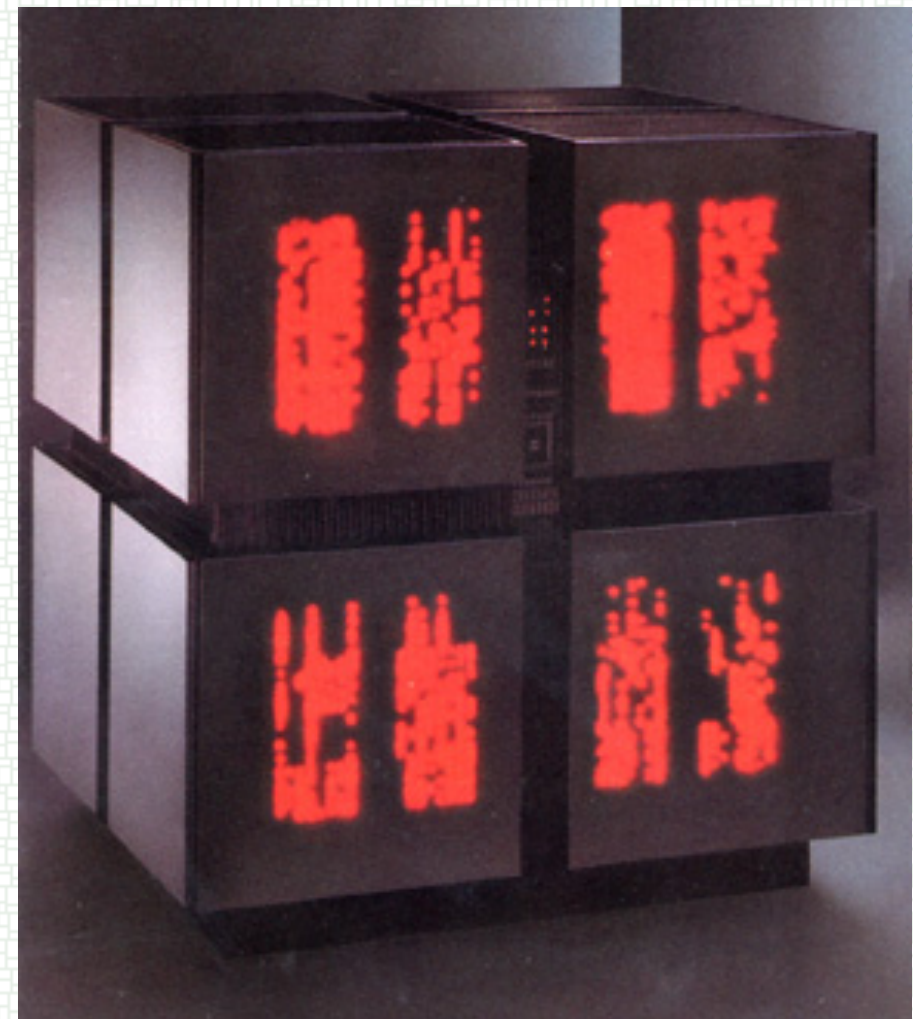


HPC HARDWARE

- ☐ network of nodes
- ☐ Many cores per node
- ☐ vector/accelerator FPGAs
- ☐ hybrid architecture

HPC HARDWARE

- ☐ network of nodes
- ☐ Many cores per node
- ☐ vector/accelerator FPGAs
- ☐ hybrid architecture



QLUA

The overarching goal is to develop a platform-independent computational platform for lattice QCD that makes efficient use of both computational scientist's time and evolving hardware on the path to exascale

$$\text{LUA} + \text{QCD} = \text{QLUA}$$

- ☐ Lua substrate
- ☐ ideas from Thinking machines days (C* and *Lisp)
- ☐ QCD-centric ops and datatypes
- ☐ Level III integration
- ☐ access to other software packages

LUA BASICS

- ☐ Scripting language
- ☐ light-weight
- ☐ dynamically typed
- ☐ managed memory
- ☐ extendable with C
- ☐ embeddable

DATA PARALLEL

- ☐ SPMD paradigm
- ☐ parallel data = serial type + geometry
- ☐ communications
 - ☐ shifts
 - ☐ reductions
 - ☐ collectives: gather & scatter

COLORED DATATYPES

- ☐ QCD-specific types
- ☐ complex vectors $[N]$
- ☐ complex matrices $[n,n]$
- ☐ fermions, propagators, gamma matrices
- ☐ arithmetic operations

QLUA GATHER

0	0	1	1	2	3	3
0	1	1	2	2	2	3
0	0	0	0	1	1	0

I1

0	0	0	0	0	0	1
0	0	0	0	1	1	1
1	1	1	1	1	1	1

I2

1	2	3	4	5	6	7
11	12	13	14	15	16	17
10	21	22	23	24	25	16

A



14	32	19	6
92	49	31	24

B

```
gr = qcd.gather(L1, L2, {I1, I2})  
B = gr:add(A)
```


QLUA SCATTER

0	0	1	1	2	3	3
0	1	1	2	2	2	3
0	0	0	0	1	1	0

I1

0	0	0	0	0	0	1
0	0	0	0	1	1	1
1	1	1	1	1	1	1

I2

1	2	3	4
5	6	7	8

A



1	1	2	2	3	4	8
1	2	2	3	7	7	8
5	5	5	5	6	6	5

B

```
sr = qcd.scatter(L1, L2, {I1, I2})  
B = sr(A)
```

CURRENT STATUS

- ☐ qlua used in production for DWF and BMW projects
- ☐ ΔN and $\Delta\Delta$ transitions
- ☐ Algorithm development

DEVELOPMENT PLANS

SHORT TERM PLANS

- ☐ Production uses
 - ☐ Nucleon form factors
 - ☐ Nucleon-Delta and Delta form factors
 - ☐ Multigrid inverters development
 - ☐ Evolution: Beyond the Standard Model
 - ☐ Disconnected Diagrams
- ☐ GPU support

LONG TERM PLANS

- ☐ Beyond qlua
 - ☐ Scripting and compiling
 - ☐ C, LLVM and PTX

REFERENCES

- ☐ *Open source*
- ☐ <https://lattice.lns.mit.edu/trac/downloads>
- ☐ <https://usqcd.lns.mit.edu/w/index.php/QLUA>