The Software of the Future on the Hardware of the Future

Jeff Hammond, William Scullin and Ray Bair

Leadership Computing Facility
Argonne National Laboratory

15 July 2010

Facilities?

Facility: the place with the supercomputer.

Facility: where there is unlimited disk storage, the magically computer runs 24-365 and the support team waits patiently for your request to install Matlab.

Facility: something that makes possible an action or process.

Facility: machine and software that creates the opportunity for scientific discovery.

Software as a facility

Is it unreasonable that NWChem was built with EMSL construction funds?

Do we build supercomputers to run LINPACK or do science?

Do we need to walk (petascale) before we can run (exascale)?

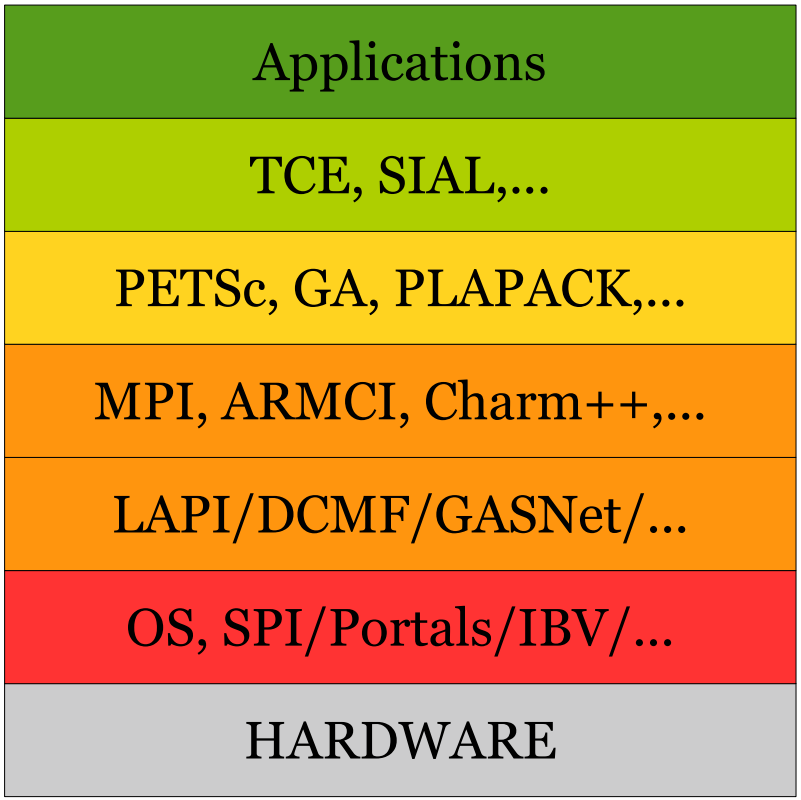
Bottlenecks to capability simulation

Unused hardware capability is a travesty:

Is it only the users problem that their code doesn't run well?

Software as a facility

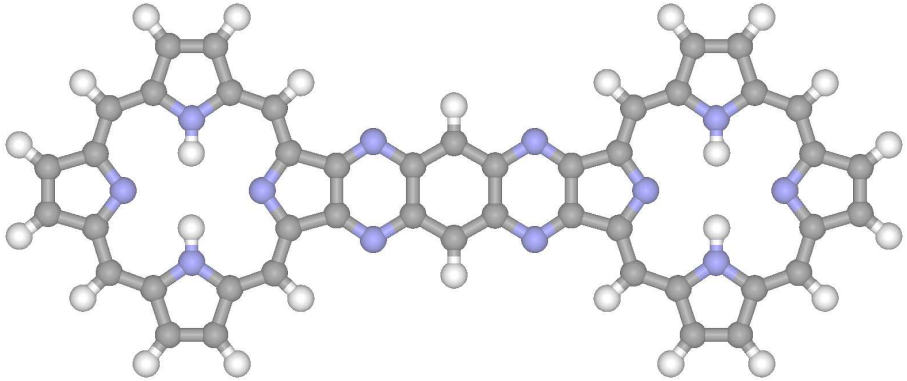
How far up the software stack do we need to go to enable exascale simulation?

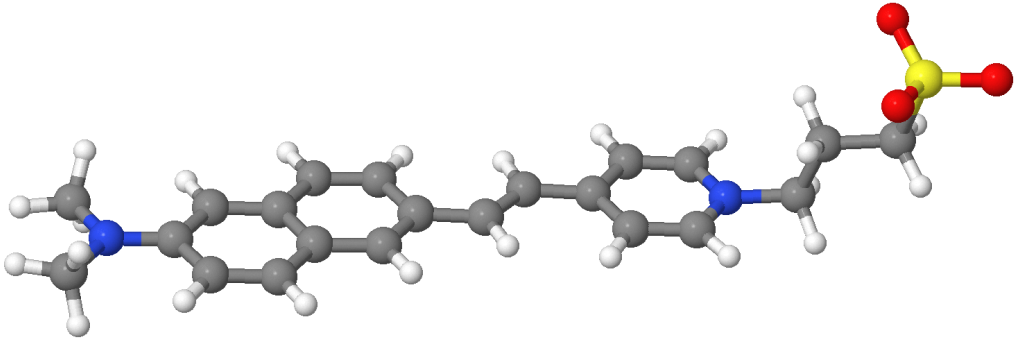


NWChem on Blue Gene/P

Motivation — Capability

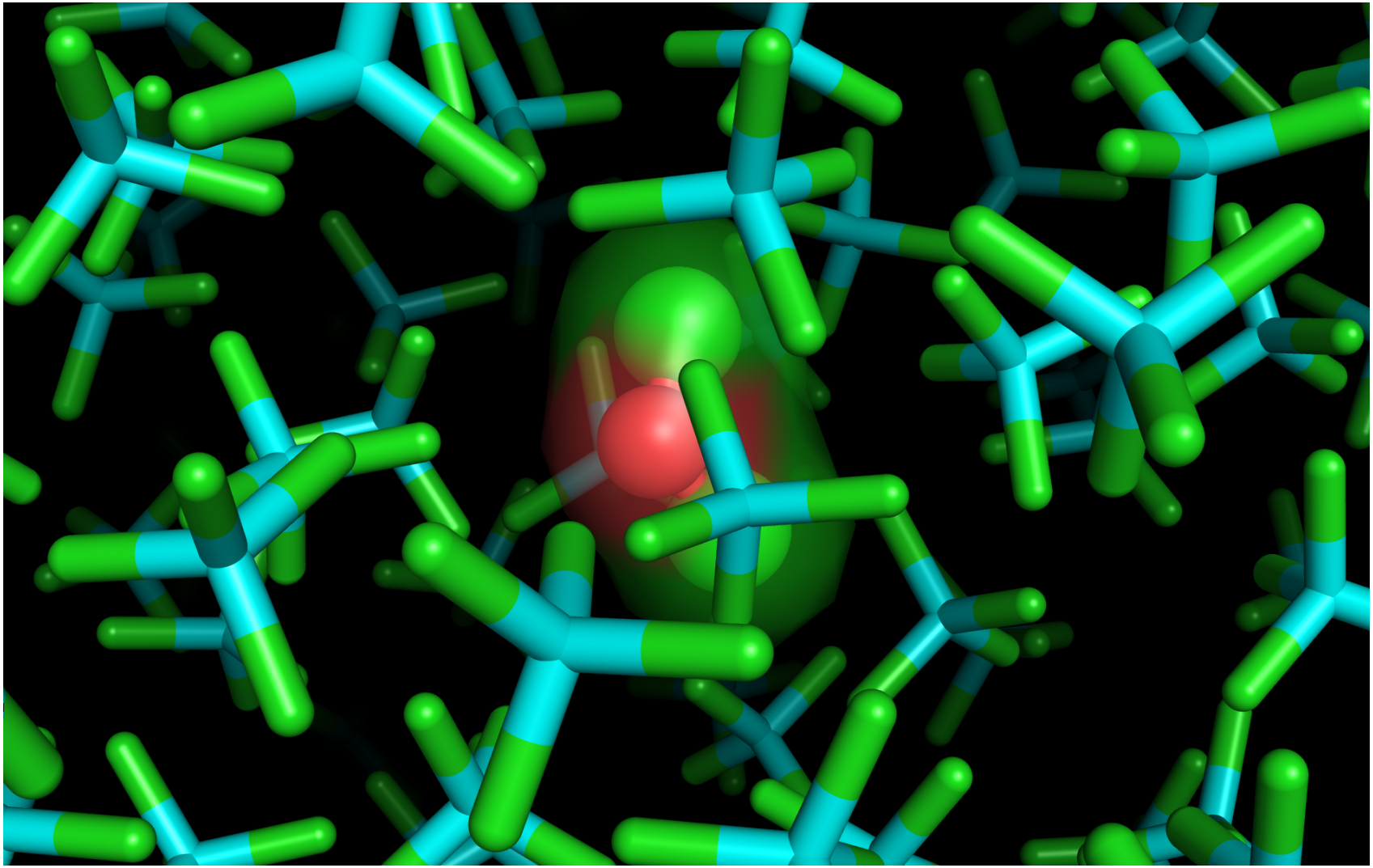
Only code which can treat 50+ atoms with quantitative accuracy, especially for excited-states and properties:





2005 CCSD(T) - 2K
2008 CCSD α - 1K
2008 CCSDT α - 1K
2009 CCSD β - 1K
2009 CCSD(T) - 96K+
2010 EOMCC - 34K

Motivation — Multiscale



Motivation — Coverage

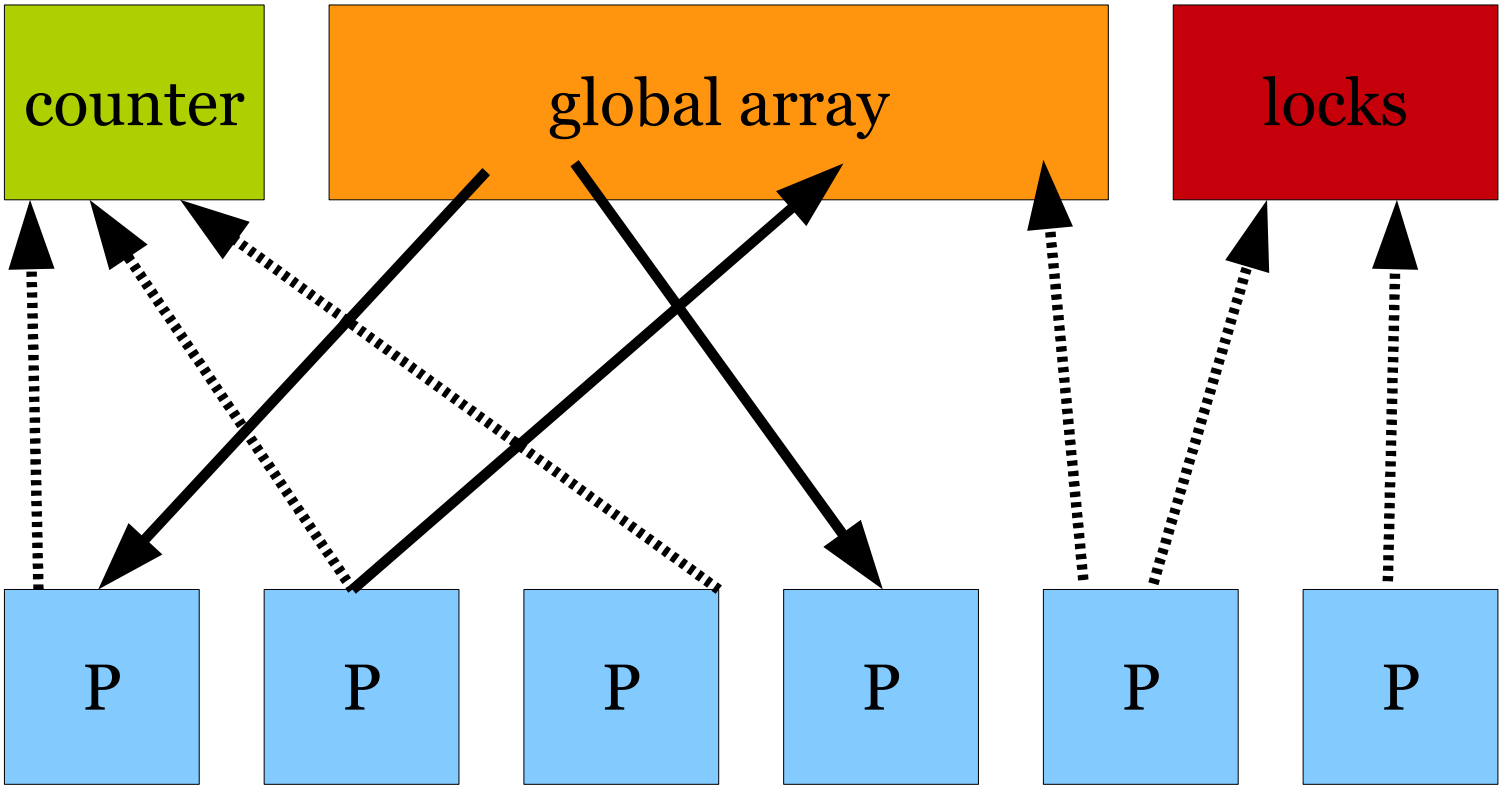
- Many-body alphabet soup
- DFT alphabet soup
- Relativistic
- QM/MM
- AIMD
- community contributions
- ...

Ported to all HPC platforms (except SiCortex).

NWChem challenges on Blue Gene/P

- vendors don't support ARMCI, can't fake it
- machine-specific optimization (esp. IA64)
- configuration assumptions (e.g. local disk)
- 82 TB and 32-bit integers???
- cannot hide behind Moore's Law

Global Arrays



RMW, local/remote completion, efficient locks N/A in MPI

Faking Global Arrays

We can always create asynchronity by splitting incoming/outgoing traffic, right?

Use threads

- No over-subscription on BG/P: lose core(s)
- Cray MPI support for `MPI_THREAD_MULTIPLE` lacking

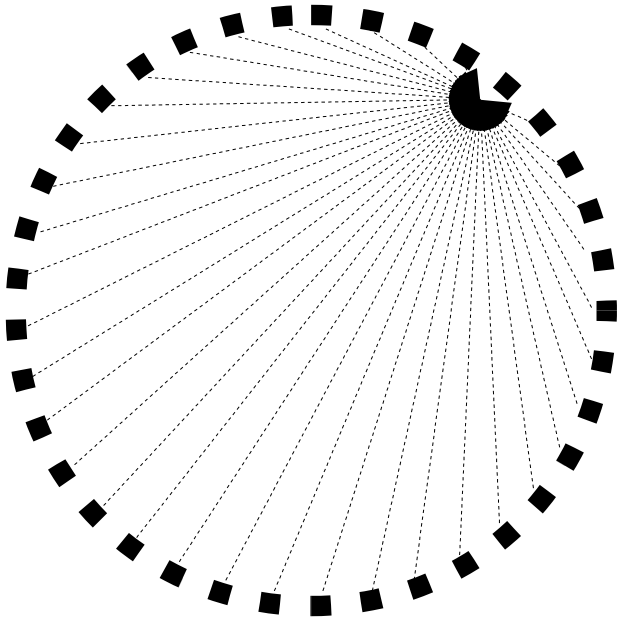
Use processes

- BG-MPI doesn't have `MPI_Comm_spawn`
- No over-subscription on BG/P: lose a core
- No UNIX IPC or SysV on BG/P

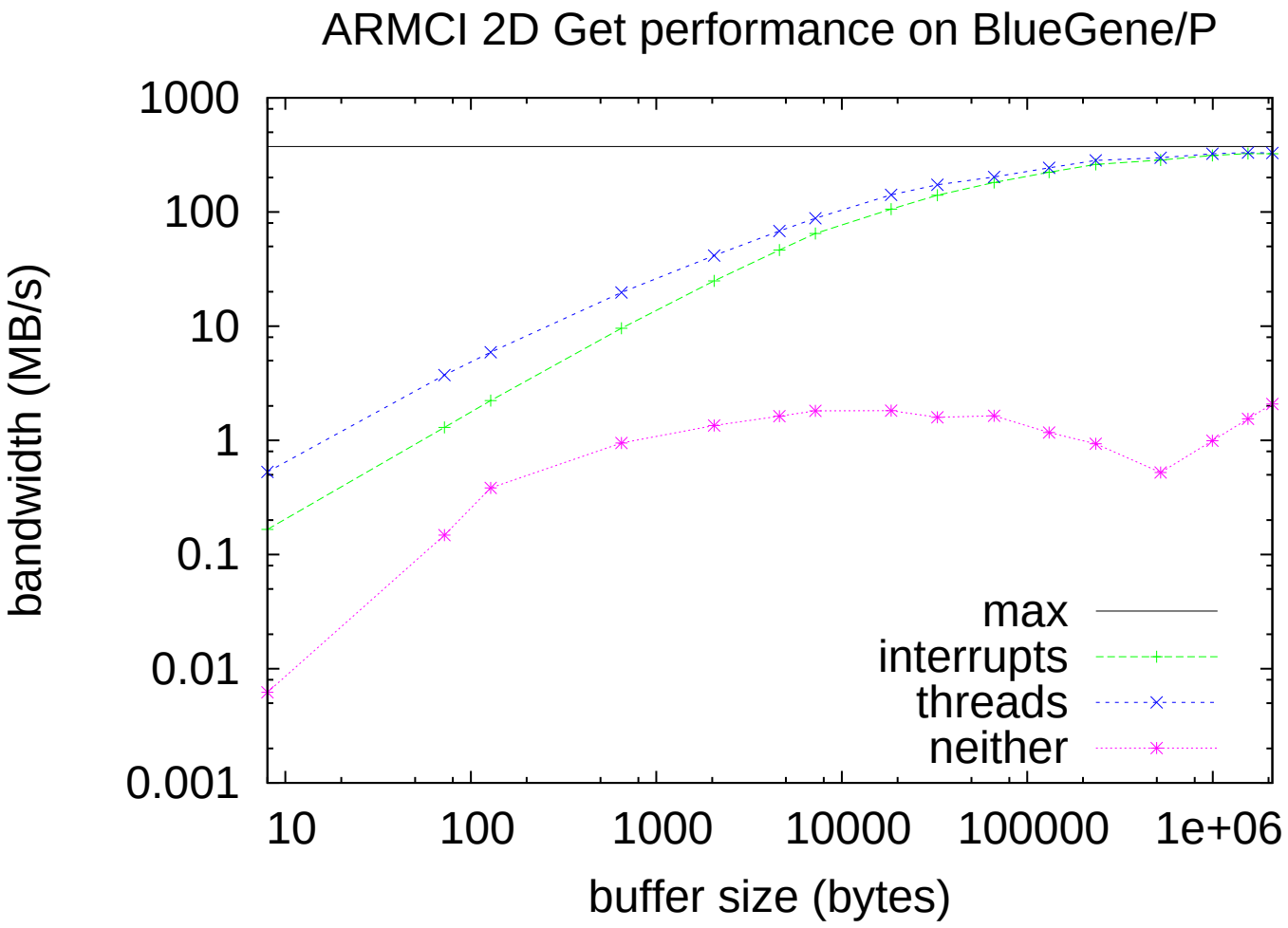
IBM makes implementing ARMCI relatively easy...

GA/ARMCI challenges

- Passive-target progress is challenging, even with DMA
- `NXTVAL` is pathological at scale $\longrightarrow$
- Not good if everyone flushes at the same time (Superbowl phenomena)
- Data-server wrong design for AM HW



ARMCI performance



ARMCI implementation

Passive-target progress:

- interrupts bad for computation intensive sections
- communication thread (CHT) burns 1/4 cores
  data-server on IB burns 2/8 cores
- CHT just makes sense for communication-intensive codes
- burning core better than massive NIC

BSP synchronization:

- `ARMCI_AllFence` granularization
- reorder flush targets

Hybridizing NWChem

CCSD(T) costs:

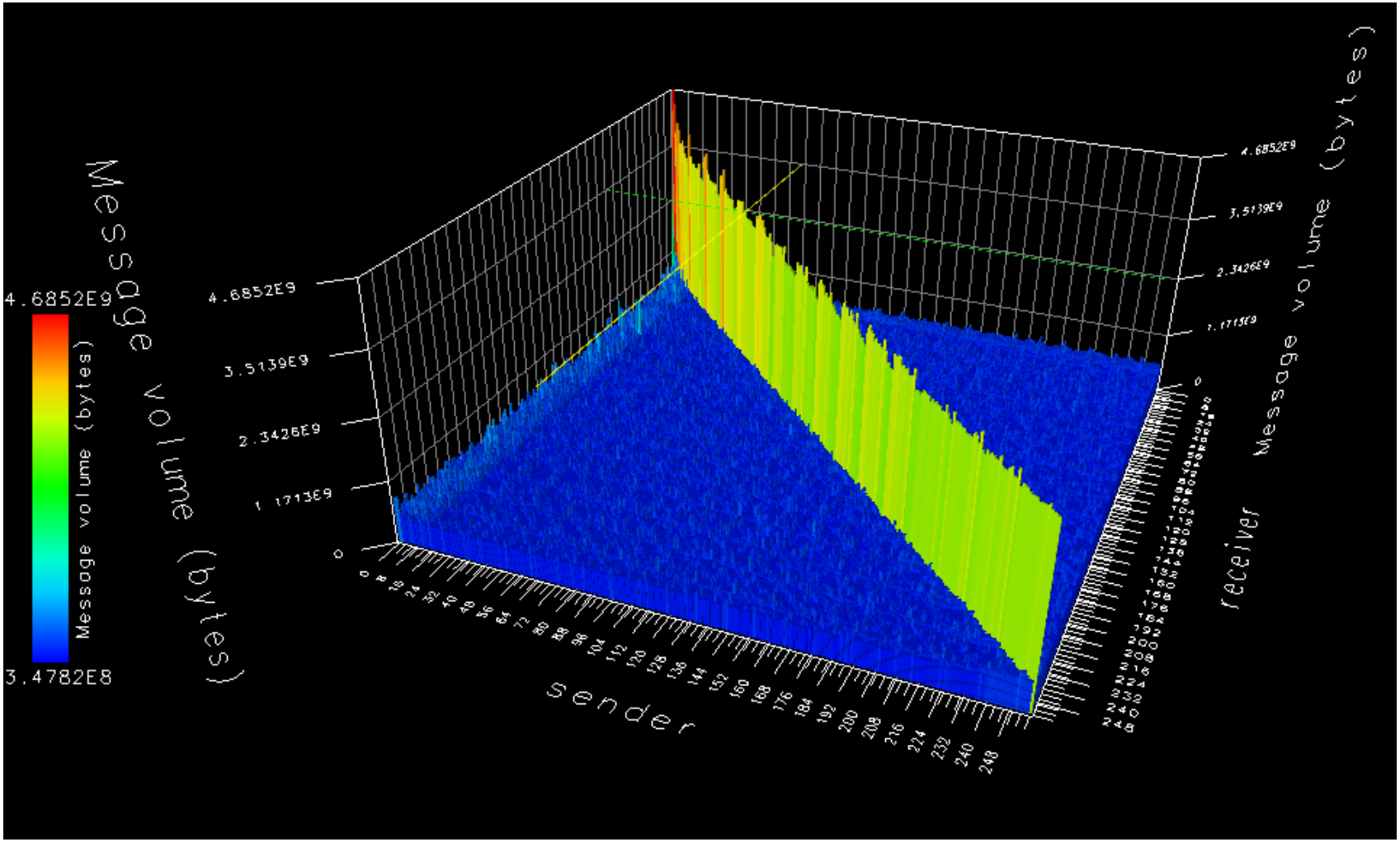
- GA_Get
- DGEMM
- TENGY

version	timing	improvement
original	923.17	-
1 thread	876.50	5.1%
2 thread	783.11	15.2%
3 thread	750.16	18.7%

Improvement in TENGY is 3.2/4 for threading (3.9 w.r.t. original) and within 20% of the theoretical peak as determined by mops alone.

Hybridization is not an option going forward...

NWChem communication matrix



GA/ARMCI lessons

Software

- GAS can create huge contention on torus
- Load-balancing strategies:
  ADLB (distributed masters)
  Charm++ (local, iterative)
- OS features versus RDMA

Hardware

- BG, Gemini don't need as much help from DS
- Intra-node memory contention
- Memory-consistency?

MPQC and MADNESS on Blue Gene/P

Overview

- Pthreads intranode, MPI internode
- CHT for accumulate, asynchronous messaging
- C++ (OO and template), configure, other modern methods
- emphasis on advanced numerics:
  MADNESS: multiresolution
  MPQC: explicitly-correlated

Porting

Both:

- IBM compiler fun with templates
- configure cross-compile

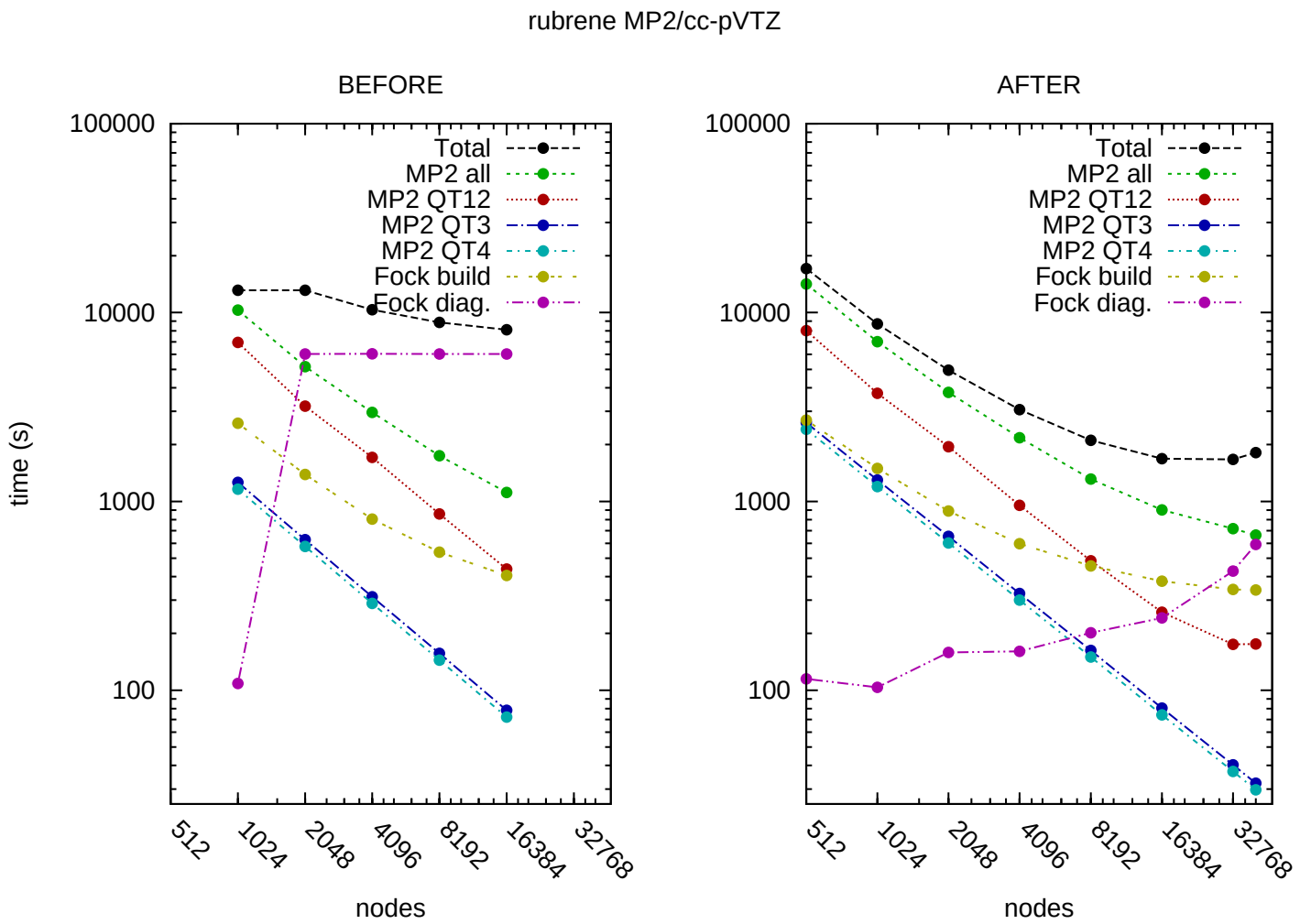
MADNESS:

- PPC atomics
- small matrix kernels in ASM
- 3 days to port and run on 1 rack

MPQC:

- 2 weeks from first execution to 40 racks.

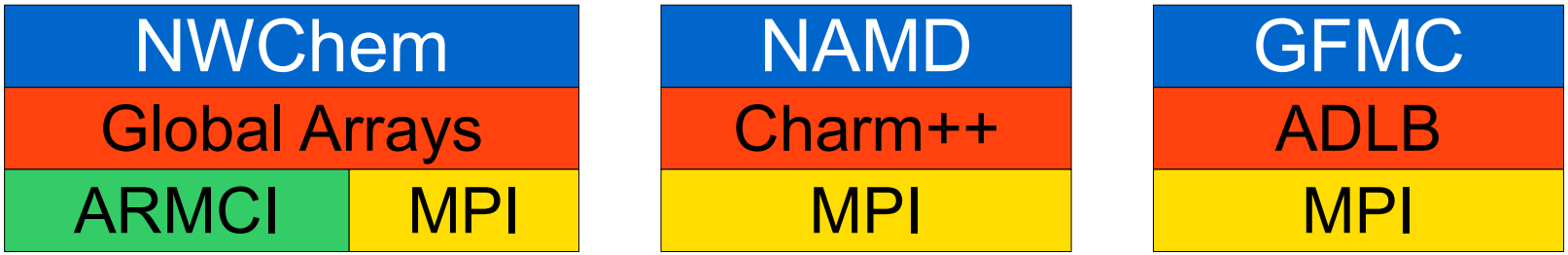
Scalability



Building applications for petascale and beyond

Runtime system co-design

Codes which require a more advanced programming model roll their own runtime systems (with help):



Can we unify runtime systems and move abstraction up a level?

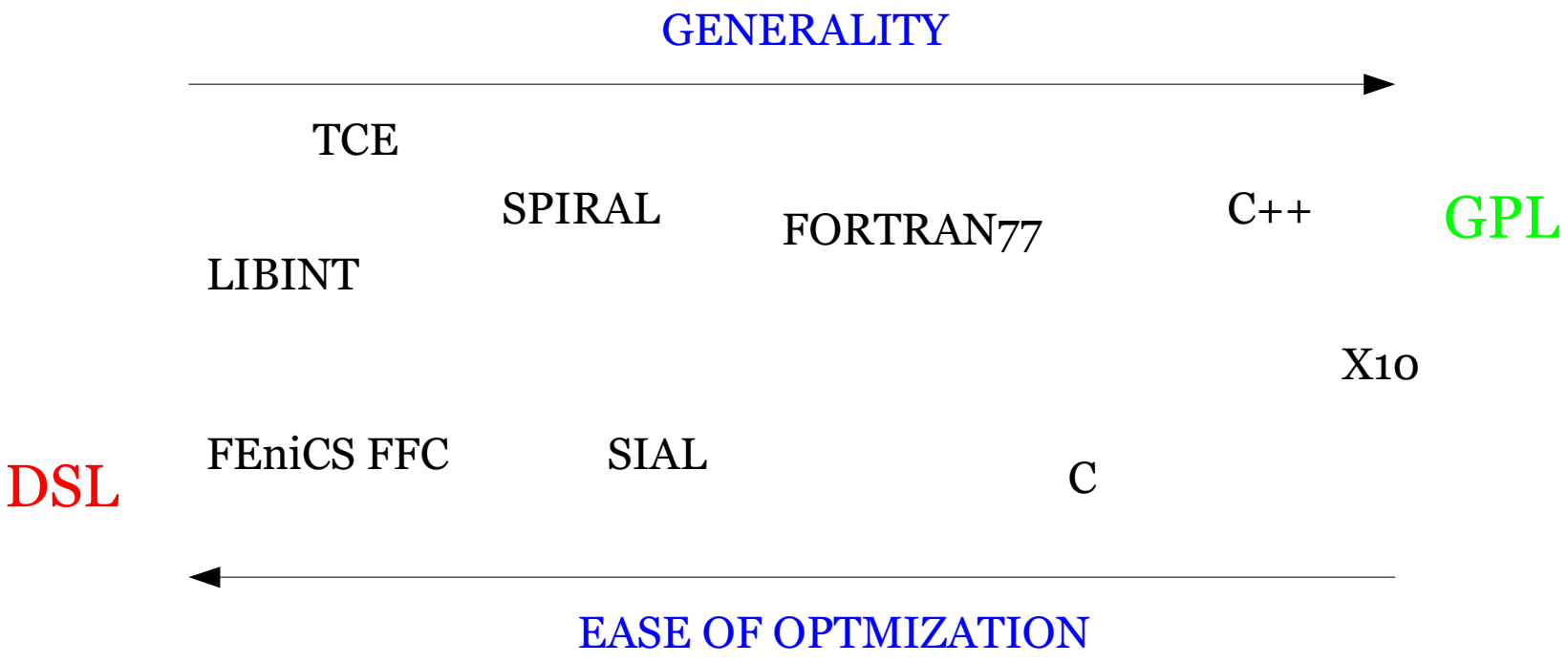
Runtime system co-design

MPI-3 *might* solve the communication primitive issue, but can we agree on programming model implementation conventions?

Despite intimate co-design, GA doesn't support every data structure required for chemistry.

TCE demonstrates that domain-specific compiler can replace static data structure approach.

Language development



DSL versus GPL

DSL success

SPIRAL: DSP generation in numerous contexts.

TCE and SIAL: QMBT massively-parallel code generation; years-to-hours compression of development time.

LIBINT: Fully-general AO generation.

Parallelizing compilers?

Should we be adding parallelization to the list of compiler duties when it cannot optimize matrix multiplication?

DSL versus GPL

DSL success

SPIRAL: DSP generation in numerous contexts.

TCE and SIAL: QMBT massively-parallel code generation; years-to-hours compression of development time.

LIBINT: Fully-general AO generation.

Parallelizing compilers?

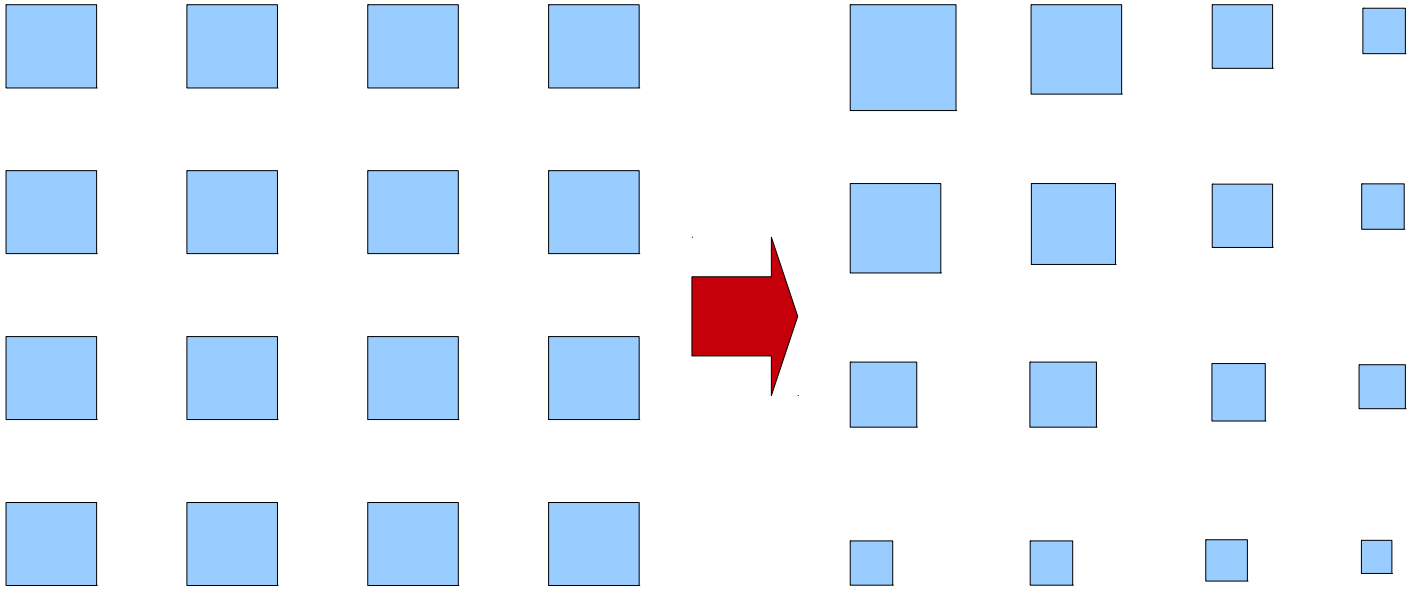
Should we be adding parallelization to the list of compiler duties when it cannot optimize matrix multiplication?

Libraries versus languages

GA versus UPC:

- Do we gain anything by replacing `GA_Put` with `UPC_Put`?
- Can a compiler recognize `GA_PutS` and `GA_PutV` semantics?
- Will GA be able to leverage next-generation networks designed for PGAS?
- If ARMCI is implemented with "#define", how is this different from what the compiler does?

Network perspective



NUMA gone wild

- hierarchical networks
- contention issues
- intranode only gets worse
- accelerators programming models?

Magical languages?

What does PGAS/HPCS know *at compile time*?

- Size of job?
- Mapping of ranks to hardware?
- Is Edo running NWChem?
  (Is FLASH checkpointing?)

Libraries and applications are better suited to address the topology problem (e.g. GPAW).

NUMA across the entire machine is significantly harder than cache-blocking, which we know compilers can't do.

Exascale

Exascale hardware architectures

How many ways to get to 10^{18}?

- billion-way thread concurrency
- hundreds to many thousands of threads per node
- 5-, 10-, 20-level memory hierarchy?

Two hardware tracks emerge, but what if your preferred track loses? Who wants to do CFD on a GPU?

Motif-driven architectures?

- Molecular dynamics (NAMD) requires MB per node, alltoall limited
- Quantum chemistry (NWChem) requires GB per node, bandwidth limited
- CFD maps to 3D torus, low-memory
- Quantum Monte Carlo insensitive to network
- 1984 injection-limited

What is the power and financial cost for each additional application area we support on a given platform?

Motif-driven architecture examples

N-body motif and Lattice QCD well-suited for specialization:

- QCDOC
- D.E.Shaw Anton
- RIKEN MD-GRAPE

Addresses power and cost efficiency requirements but is this approach extensible?

What are the practical barriers to FPGA and ASIC adoption?

Acknowledgments

ANL: Vitali Morozov, Pavan Balaji

PNNL: Karol Kowalski, Sriram Krishnamoorthy

ORNL: Robert Harrison, Vinod Tipparaju

OSU: P. Sadayappan, Sreeram Potluri

IBM: Brian Smith, Michael Blocksome