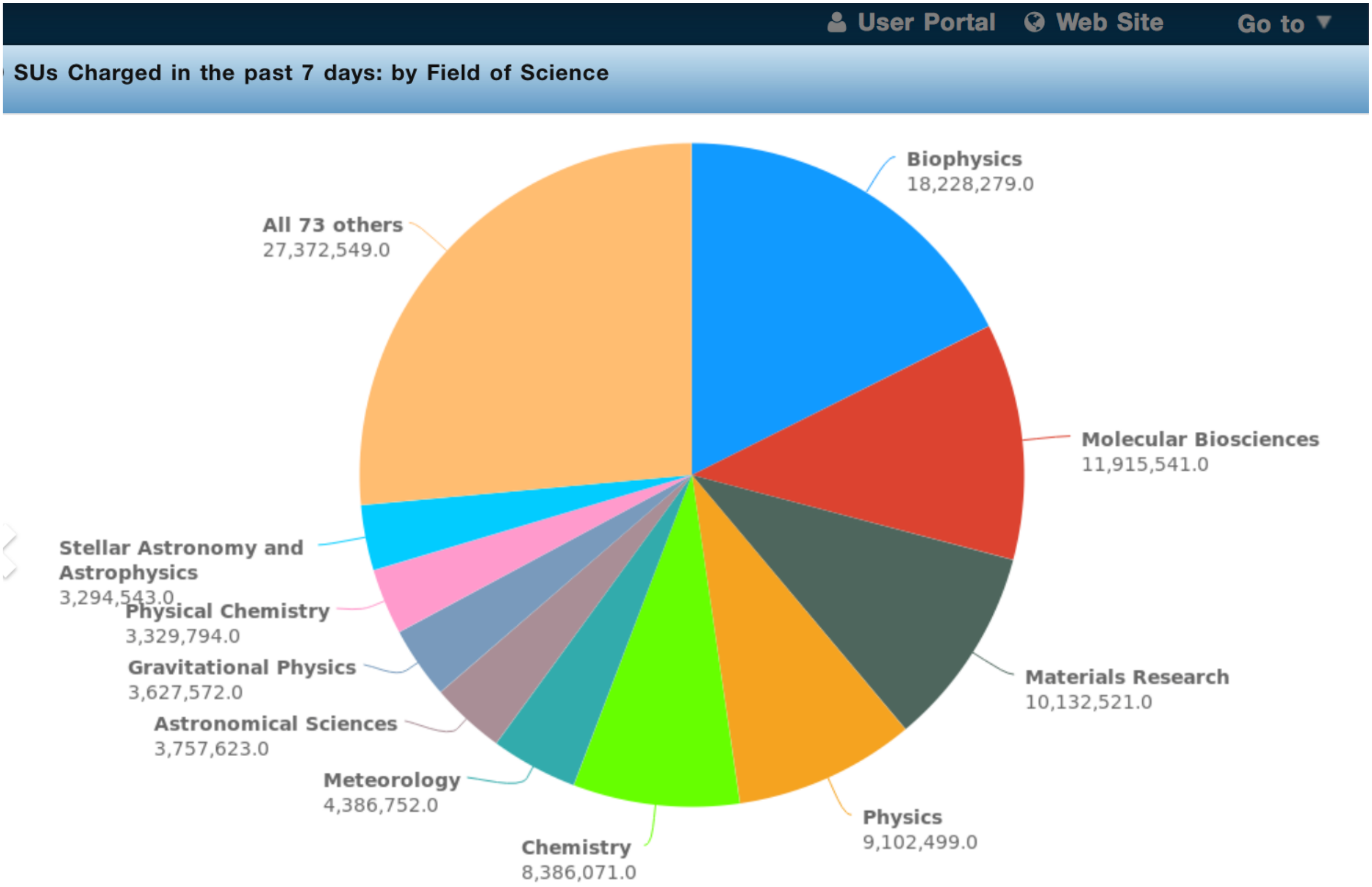
LIFE SCIENCE: USING HPC FOR INSIGHT INTO BIOMOLECULAR FUNCTION

Thomas E Cheatham III, University of Utah, Salt Lake City, UT
Erik Lindahl, Stockholm University, Stockholm, Sweden
tec3@utah.edu
erik.lindahl@scilifelab.se

Why multiple life sciences lectures?

XSEDE usage over the past 7 days (prior to 2015 iHPC-SS)

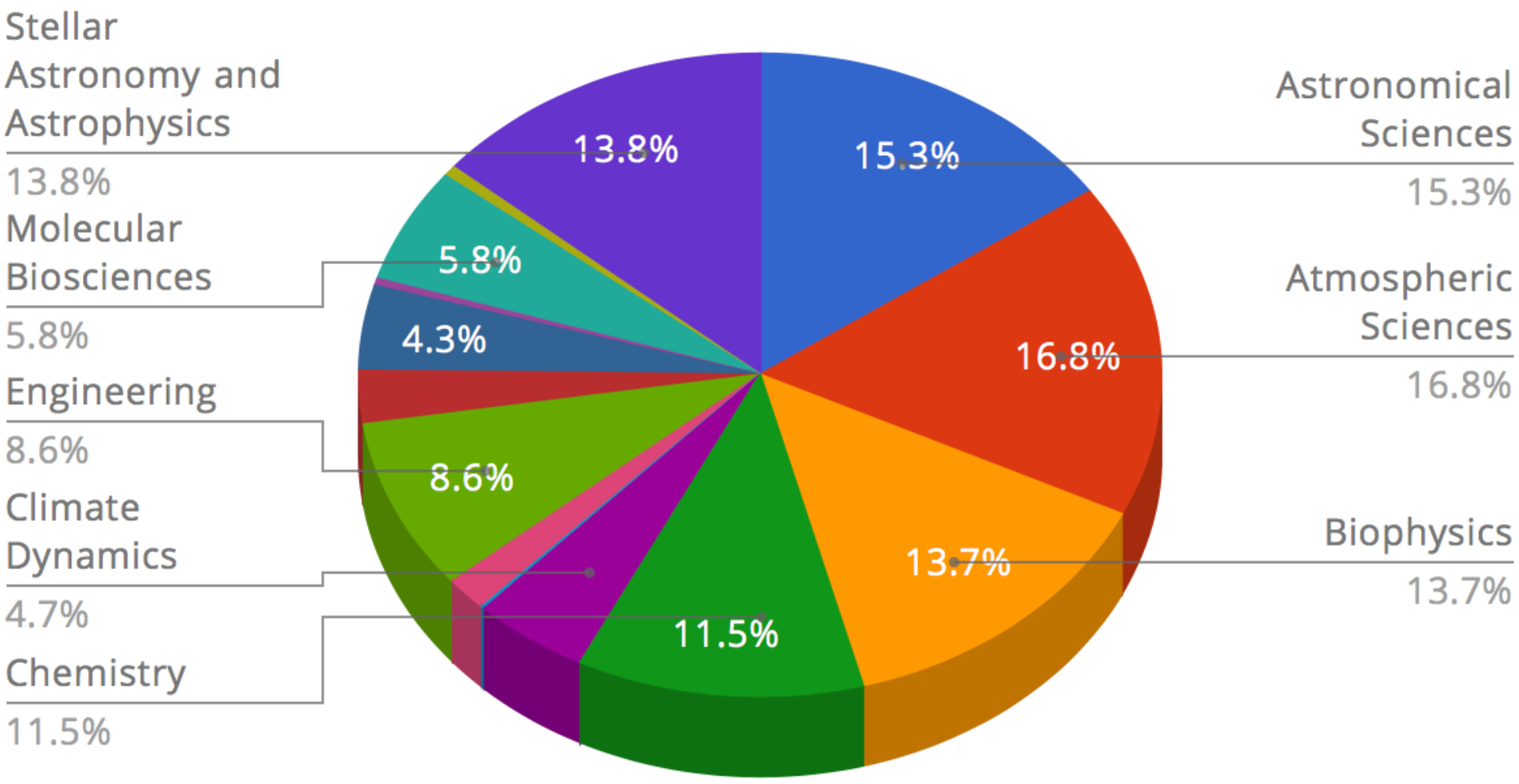


BLUE WATERS
SUSTAINED PETASCALE COMPUTING

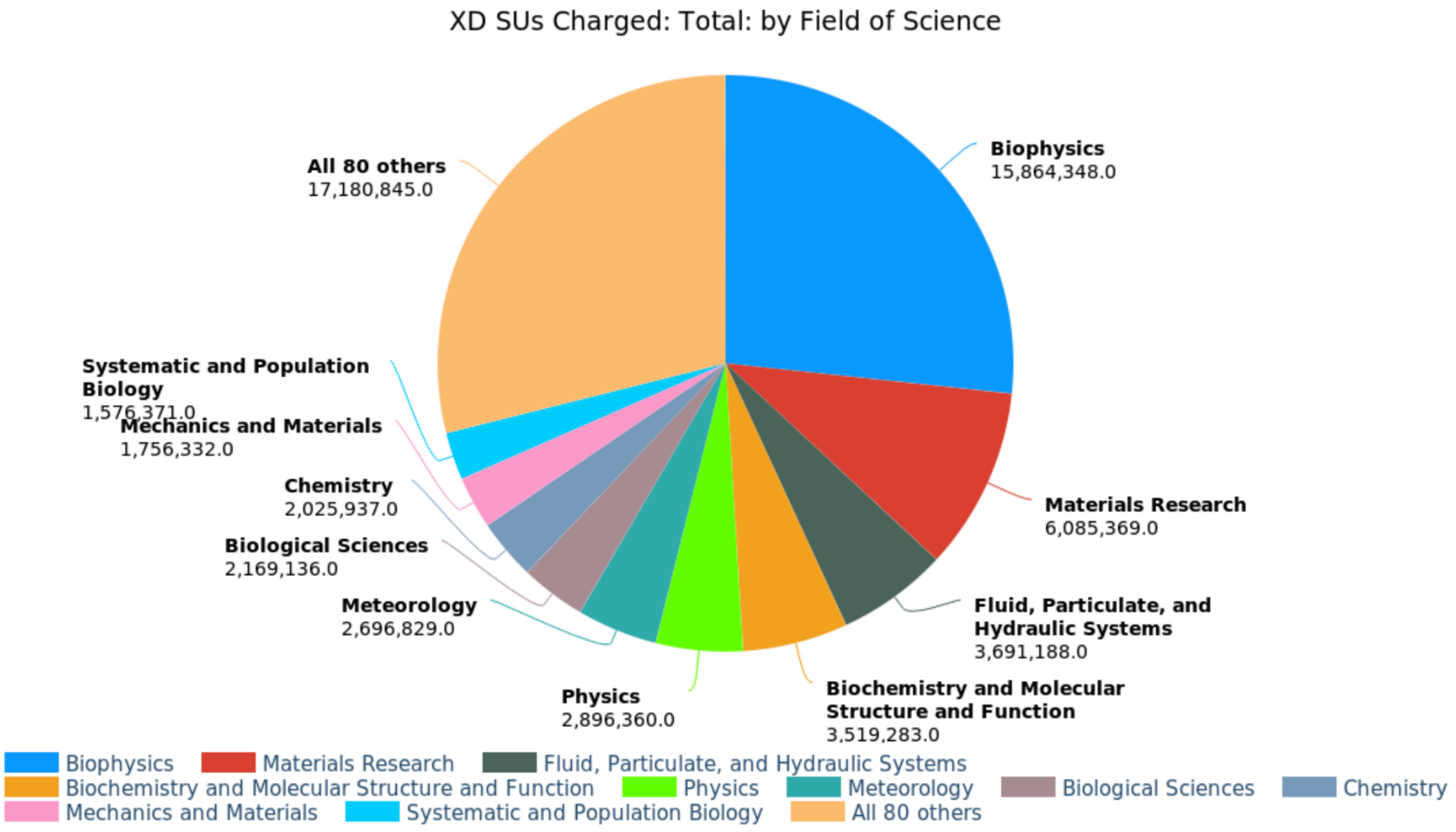
GPU nodes

Ensembles of molecular dynamics engines for assessing force fields, conformational change, and free energies of proteins and nucleic acids (**Jobs:**38) **PI:** Thomas Cheatham, University of Utah	XK	28400	310,293.43
Predicting protein structures with physical petascale molecular simulations (**Jobs:**21) **PI:** Ken Dill, SUNY at Stony Brook	XK	10240	276,988.80

CURRENT RUNNING JOBS BY SCIENCE AREA

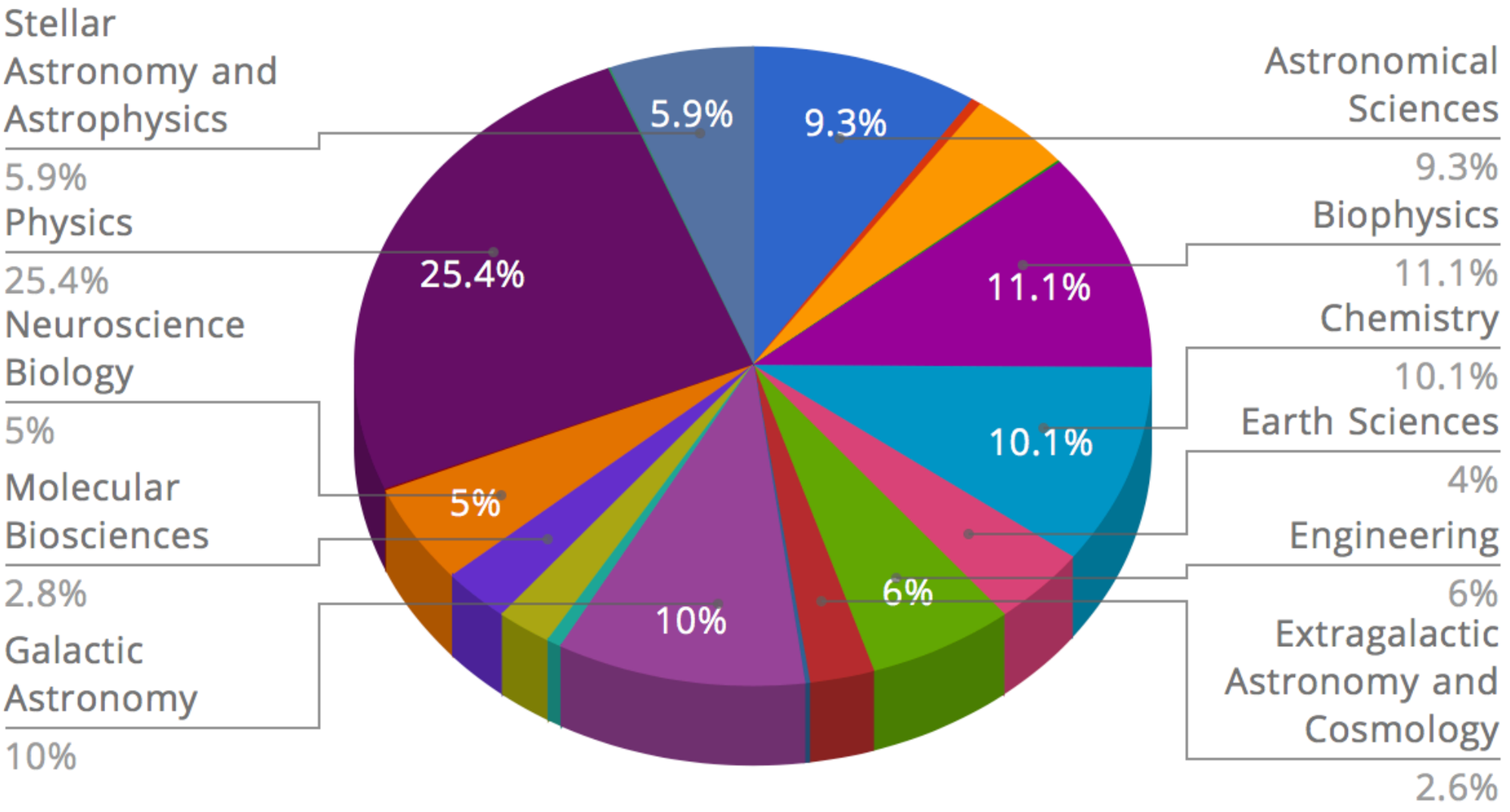


Last 7 days on XSEDE resources – May 22, 2017



BLUE WATERS
SUSTAINED PETASCALE COMPUTING

CURRENT RUNNING JOBS BY SCIENCE AREA



…more recently…

May 18, 2015

25 Percent of Life Scientists Will Require HPC in 2015

John Russell

Life science research has long been compute-intensive but requirements have largely been satisfied with traditional workstations and simple clusters. That's changing: "Roughly 25 percent of life scientists, and this includes bench-level scientists, will require HPC capabilities in 2015, few of whom have ever used a command line," said Ari Berman, GM of Government Services, the BioTeam consulting firm.

Predictably, the flood of DNA sequence data is a major driver. NIH now generates 1.5PB of data a month, and that is only from internal work and doesn't include NIH-funded external research. "[This might be the] first real case in life science where 100Gb networking might be really needed," said Berman.

However there are many contributors to the growing data flood and computing complexity in LS including, for example, proteomic data, protein structure data, cell and organelle imaging data, pathway modeling data, and efforts to integrate all of them for analysis.

"There's a revolution in the rate at which lab platforms are being redesigned, improved, and refreshed. Instrumentation and protocols are changing far faster than we can refresh our research IT and scientific computing infrastructure," said Berman, speaking to a distinguished audience at the spring HPC User Forum.

"Bench science is changing month to month while IT infrastructure is refreshed every 2-7 years. Right now IT is not part of the conversation [with life scientists] and running to catch up," he said.

Given the diversity in data types (massive text and binary files), file sizes (spanning large 600GB+ to very many 30kb or smaller files), and applications workloads, the best approach to building HPC capabilities is to focus around specific use cases rather than simply chase general performance, said Berman, who presented a fairly detailed outline of emerging HPC requirements with LS.

Berman said common LS application characteristics today include:

- Mostly SMP/threaded apps performance bound by IO and or RAM
- Hundreds of apps, codes, and toolkits
- 1TB-2TB RAM "High Memory" applications (large graphics, genomic assembly)
- Lots of Perl/Python/R
- MPI is rare (well-written is even rarer)
- Few MPI apps actually benefit from expensive low-latency interconnects (chemistry, modeling and structure work is the exception)

AMBER (Cheatham) vs. GROMACS (Lindahl)

ambermd.org

gromacs.org

history, code development, philosophy, approach, synergies, differences, challenges, futures, lessons learned

Tom:

Thomas Cheatham, III
Professor of Medicinal Chemistry, College of Pharmacy
Director, Center for High Performance Computing, University of Utah 7/1/14-

1988-1990	1990-1997	1997-2000	2000-present
programmer/analyst	graduate school	NIH postdoc	Res Asst Prof - Professor
Harvard U (DAS/ACS)	NSF centers	NIH only	CHPC+AAB→TG→XSEDE→BW
CM-2, CM-5, MasPar	T3D, T3E, Crays	beowulf, IBM SP-2	(many || + GPUs)

DOE Summer Institute @ Los Alamos (vector)
PSC Summer Institute in parallel computing
PSC Workshop on Hetereogeneous Computing
PSC AMBER Workshop (Teacher)

Allocations committee, chair
TeraGrid Science Advisory Board, chair
XSEDE User Advisory Committee, chair
XSEDE SMT, SAB
Blue Waters SETAC
ACI-REF PI Chair
CaRC Chair
RMACC, Vice-Chair

Erik:

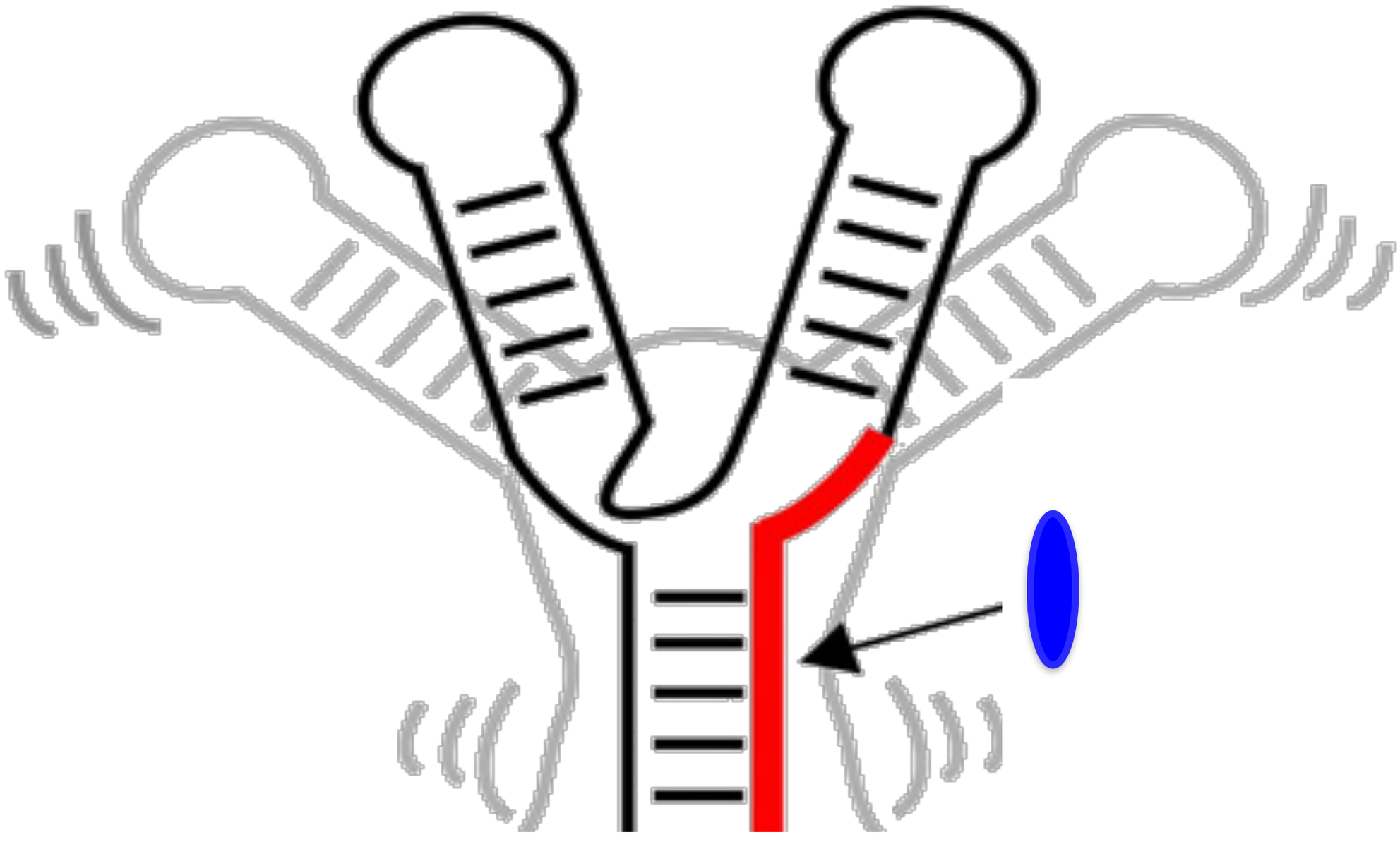
Erik Lindahl
Professor of Biophysics, Stockholm University
Professor of Theoretical Biophysics, KTH Royal Institute of Technology
Vice Director, Swedish e-Science Research Center

1996-2001	2001	2002-2003	2004	2004-present
grad. school	Groningen Univ.	Stanford Univ.	Inst. Pasteur	Faculty
KTH, Sweden	Local	Local (NIH)	Local+SNIC(SE)	SNIC+PRACE
IBM SP2, PPC	Beowulf	Linux/x86 clusters		Everything+CUDA+OCL

Board of Directors, Swedish National Infrastructure for Computing
Chair, PRACE Scientific Steering committee
Platform director, Bioinformatics, SciLifeLab
Lead scientist, BioExcel Center of Excellence for biomolecular computation
Vice director, Swedish e-Science Research Center
Swedish Research Council

What do we want to do? **Accurately model the structure and dynamics of molecules in their native environment**

(i.e. follow the motions of the atoms subject to an energetic potential or "force field" as a function of time…)



Accurate modeling of molecules requires:

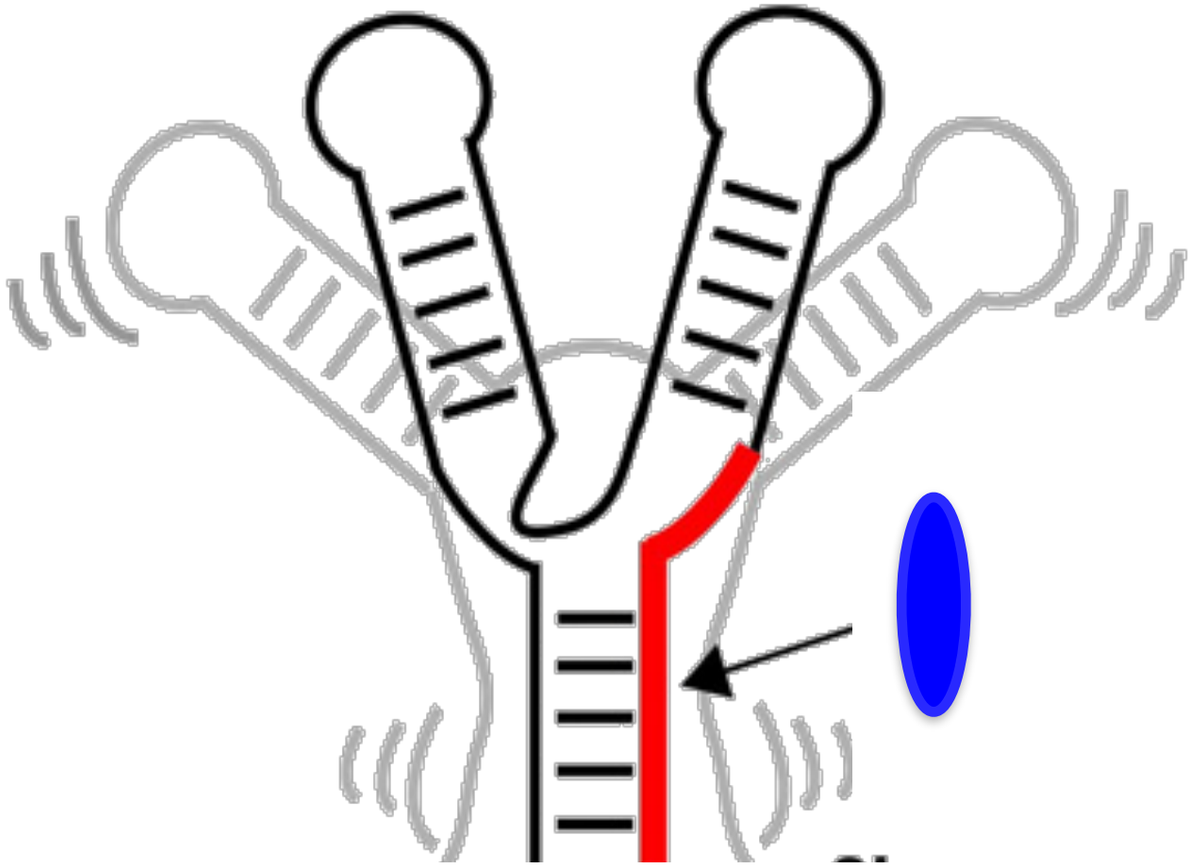
accurate and fast simulation methods

validated RNA, protein, water, ion, and ligand "force fields"

"good" experiments to assess results

dynamics and complete sampling: (convergence, reproducibility)

Question: Is the movement real or artifact?



molecular simulation / molecular dynamics

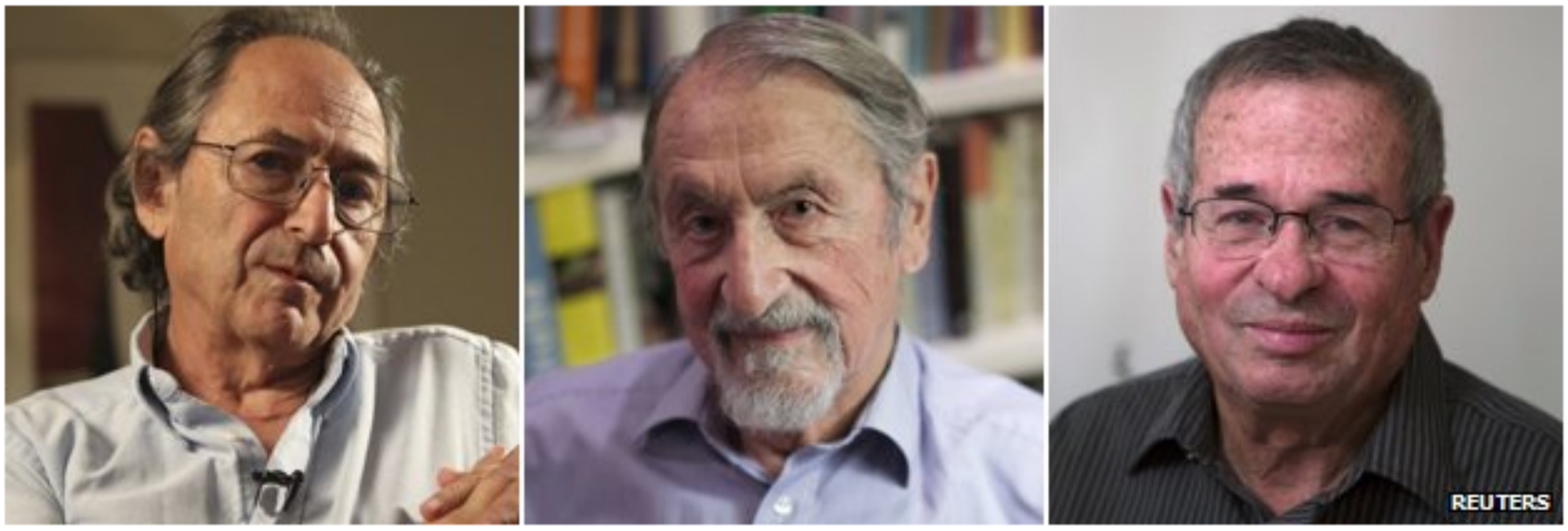
…is not new,
has a rich history,
and is largely solved (?)

Computer chemists win Nobel prize

9 October 2013

By James Morgan and Jonathan Amos
Science reporters, BBC News

"for the development of multiscale models for complex chemical systems

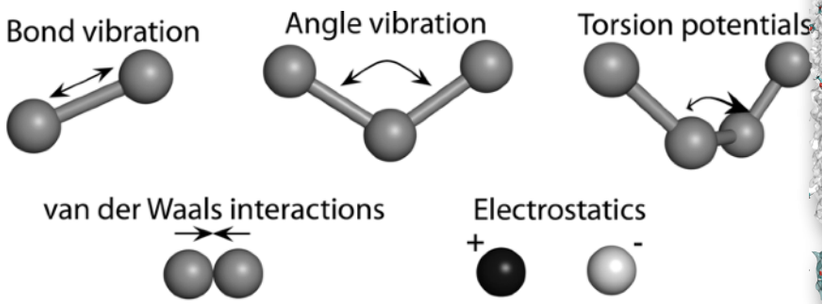


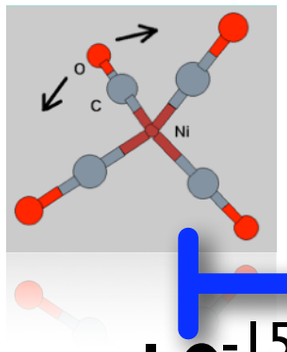
The work of Levitt, Karplus and Warshel has spawned a worldwide industry

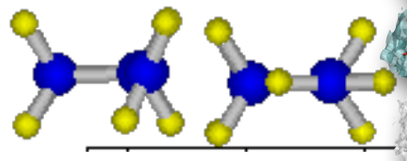
The Nobel Prize in chemistry has gone to three scientists who "took the chemical experiment into cyberspace".

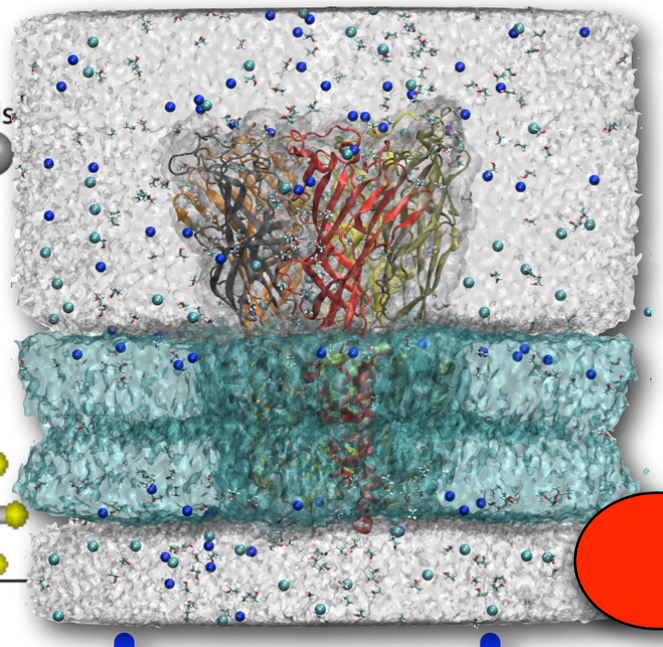
Michael Levitt: ""It's sort of nice in more general terms to see that computational science, computational biology is being recognized," he added. "It's become a very large field and it's always in some ways been the poor sister, or the ugly sister, to experimental biology."

Molecular simulations









Experiments

Efficient averaging

Less detail

Where we need to be

10^{-15} Physics 10^{-12} 10^{-9} s 10^{-6} s Chemistry 10^{-3} s 10^{0} s Biology 10^{3} s

Where we are

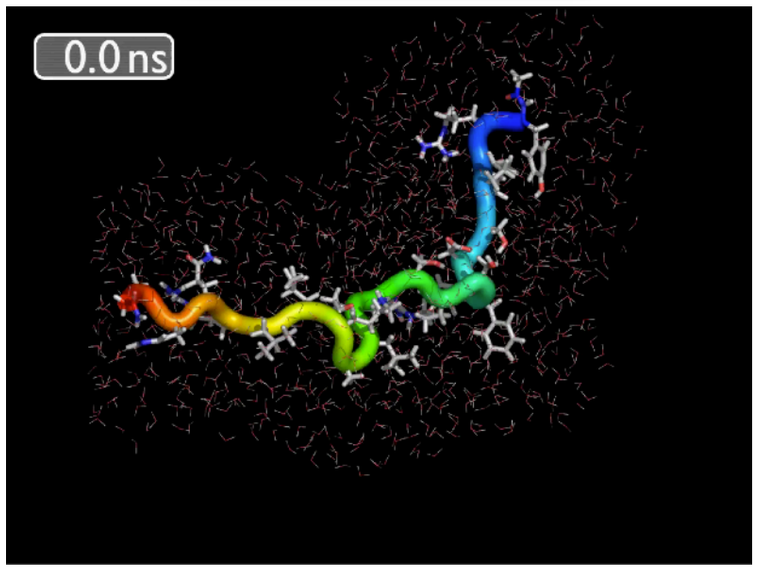
Where we want to be

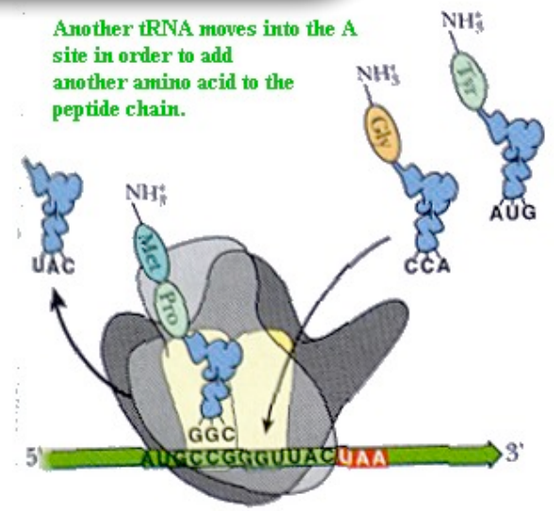
Simulations

Extreme detail

Sampling issues?

Parameter quality?





Larger machines have mostly enabled larger systems, not longer simulations

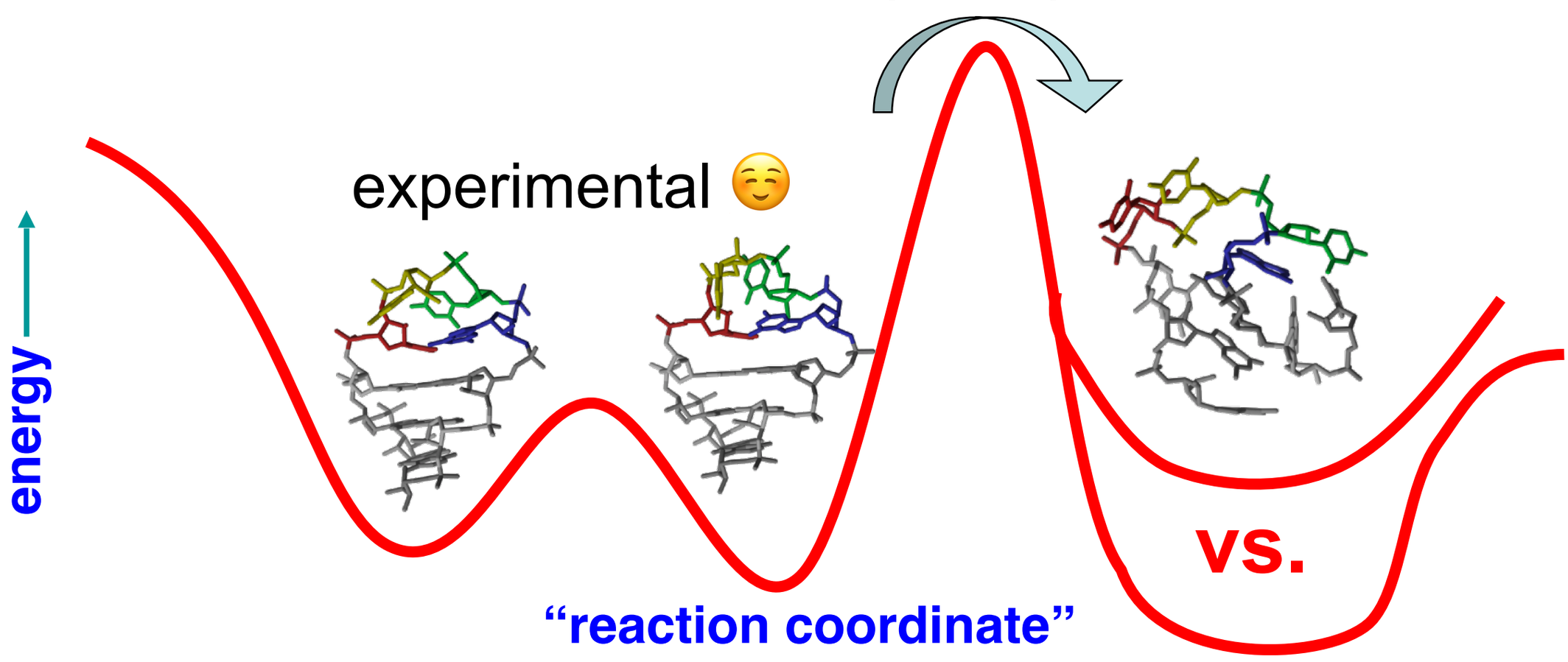
When we started, programs could use O(10) cores

How do we develop these codes and problems to use O(10,000) cores?

are the force fields reliable?
(free energetics, sampling, dynamics)

Short simulations stay near experimental structure; longer simulations invariably move away and often to unrealistic lower energy structures…

Computer power?



~1978 - present

amber

Assisted Model Building with Energy Refinement

~1978 - present

amber

Assisted Model Building with Energy Refinement

code vs. force field

the setup and calculation engines

the parameters and potentials

~1978 - present

amber

Assisted Model Building with Energy Refinement

code vs. force field

the setup and calculation engines

the parameters and potentials

- not really a professional code (some experts, some beginners)
- not really software engineered (parts were, like GPU code, optimizations)
- it is continually evolving; one of the first “community codes”…
- development efforts are not directly funded (except maybe GPU)

~1978 - present

amber

Assisted Model Building with Energy Refinement

code vs. force field

late 60's: CFF (consistent force field) + early code
{Warshel, Levitt, Lifson}

first protein simulation ~1975

1978: Bruce Gelin thesis @ Harvard {Karplus}

GROMOS

GROMACS

Amber 1.1, **1981**
(minimization only, f'')

Amber 2, **1984**
(+ dynamics)

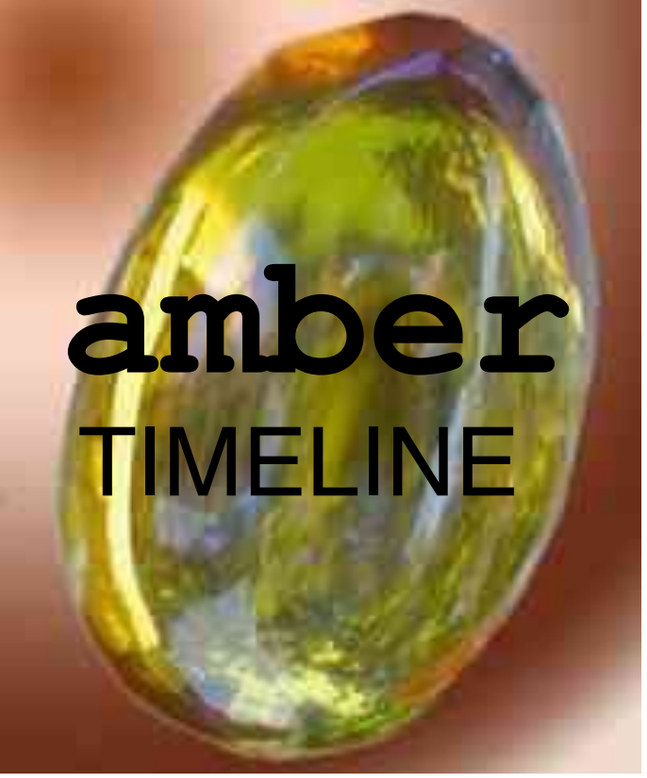
CHARMM

NAMD

ENCAD

Discover

first nucleic acid simulation in H_2O ~1985



1986: amber3

ΔG, QM/MM, non-additivity

1989: amber3a

code cleanup, bug fixes
increased performance, portability
vectorization, || on hypercube,
shared memory
Intel Paragon 1/3 speed of Y-MP

1990-1994: ~~SPASMS~~

?

(blue matter)
~2004

Special purpose?
MD-GRAPE
SFE, Tera, …

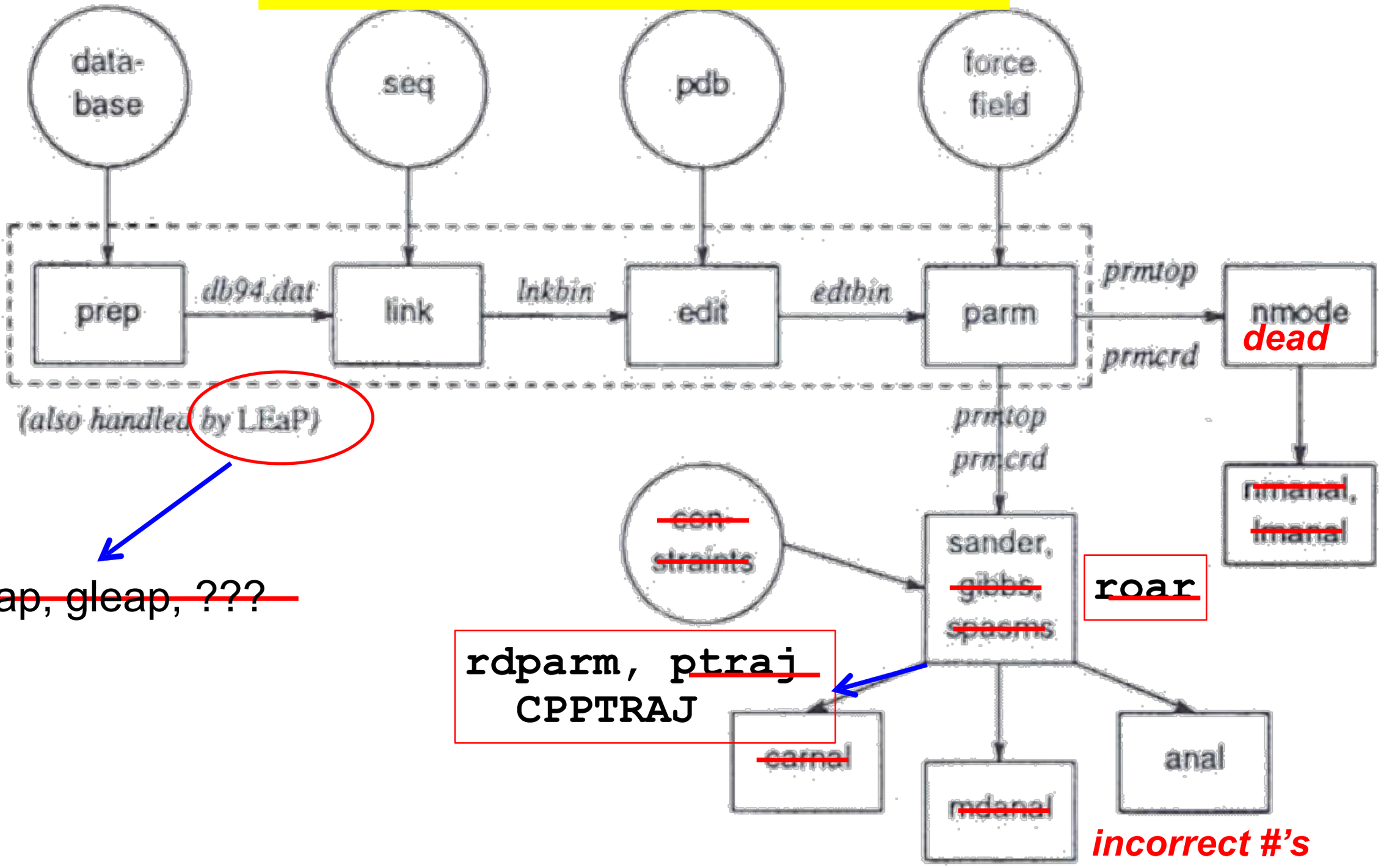
1991: amber4.0

NMR refinement, normal modes, ΔG
serious code bifurcation
|| message passing
(TCGMSG, PVM, MPI, …)

1994: amber4.1

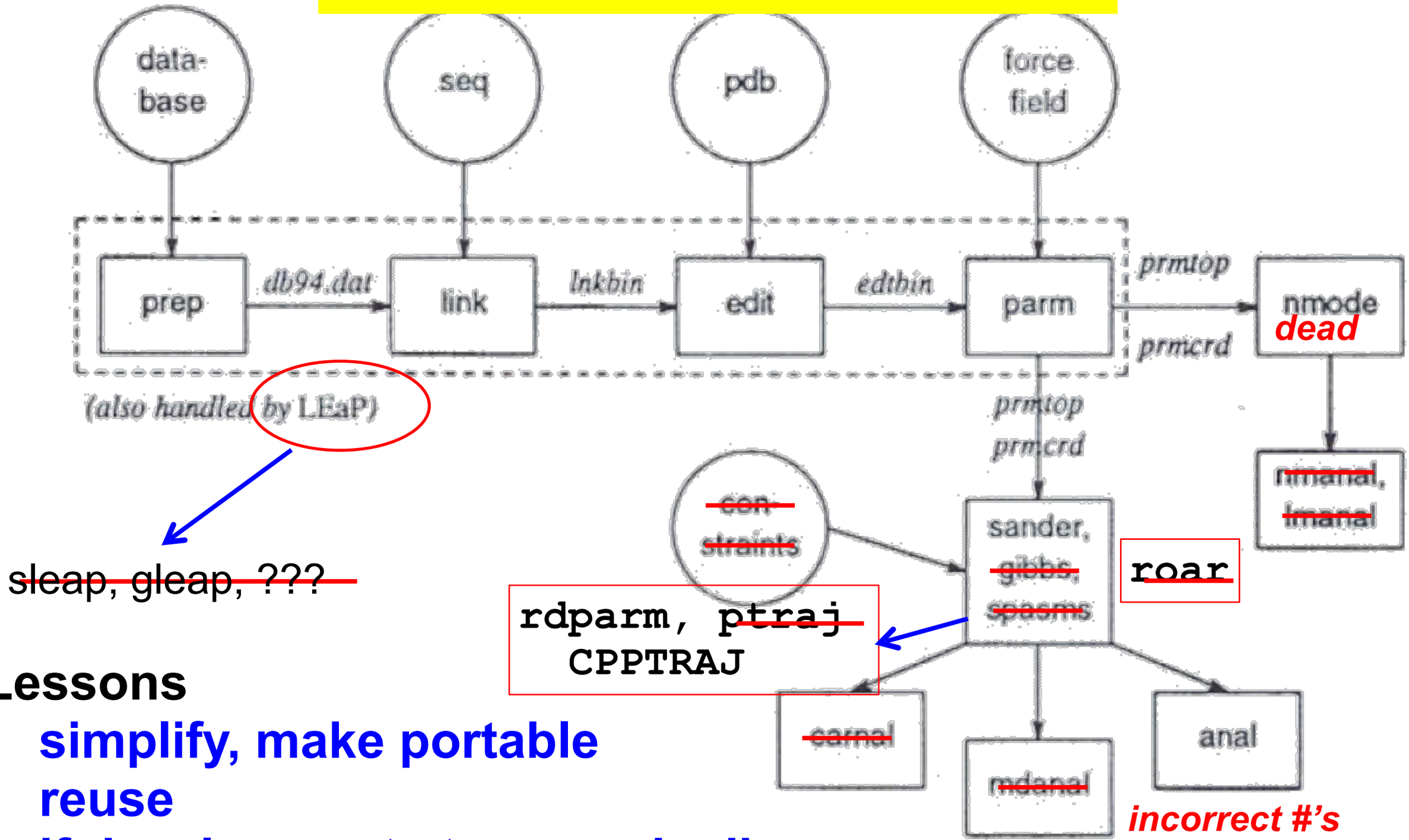
particle mesh Ewald ☺️
more shared memory, MPI only
`#ifdef MPI`

…evolution of AMBER 4.1 codes



D.A. Pearlman et. al. / Computer Physics Communications 91 (1995) 1–41

…evolution of AMBER 4.1 codes



Lessons

- simplify, make portable
- reuse
- if development stops, code dies
- replace functioning code with new code most often fails

early days: **ftp repository, makefiles (many), MACHINEFILE**

4.1-7.0: **CVS, C memory allocation move to F90, makefiles**
compile script recognizing MACHINEFILE
(fight w/ compiler for giganet vs. myrinet vs. …)

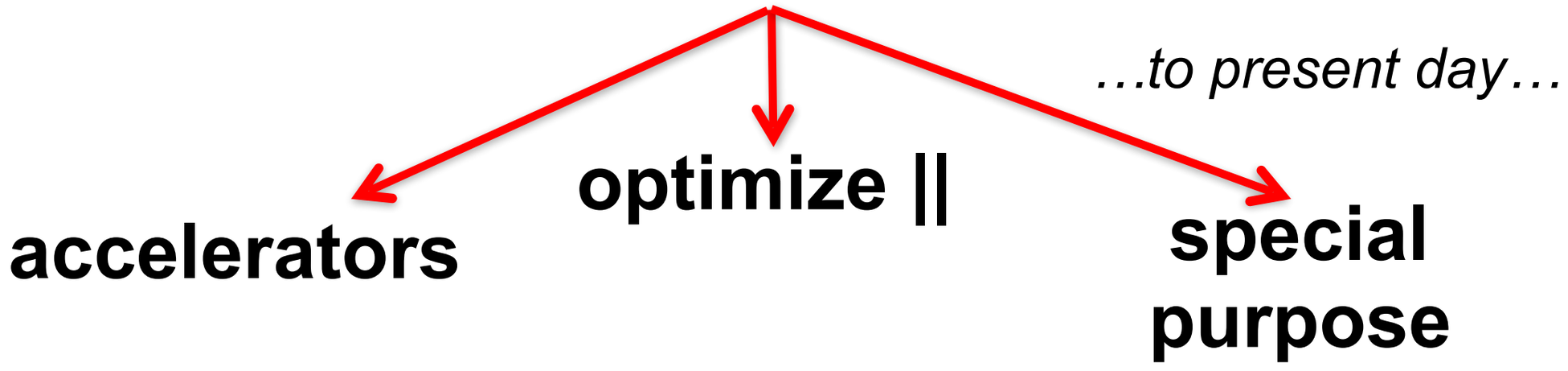
simplify, unify (as machines are becoming homogeneous)
drop vectorization, drop shared memory, drop machine specific opts

8.0: (2004) **introduce fast engine pmemd, configure scripts**

focus on fewer compilers: gnu, intel, pgi, pathscale
minimize #ifdefs to infrequently used code paths

~1995-2005: homogeneous hardware, standard MPI ||

…to present day…



floating point precision: single vs. double vs. mixed/fixed

early days: **ftp repository, makefiles (many), MACHINEFILE**

4.1-7.0: **CVS, C memory allocation move to F90, makefiles**
compile script recognizing MACHINEFILE
(fight w/ compiler for giganet vs. myrinet vs. …)

simplify, unify (as machines are becoming homogeneous)
drop vectorization, drop shared memory, drop machine specific opts

8.0: (2004) **introduce fast engine pmemd, configure scripts**

focus on fewer compilers: gnu, intel, pgi, pathscale
minimize #ifdefs to infrequently used code paths

10.0: (2008) **AmberTools (open source), OpenMP**
separate configure for AmberTools, sander, pmemd

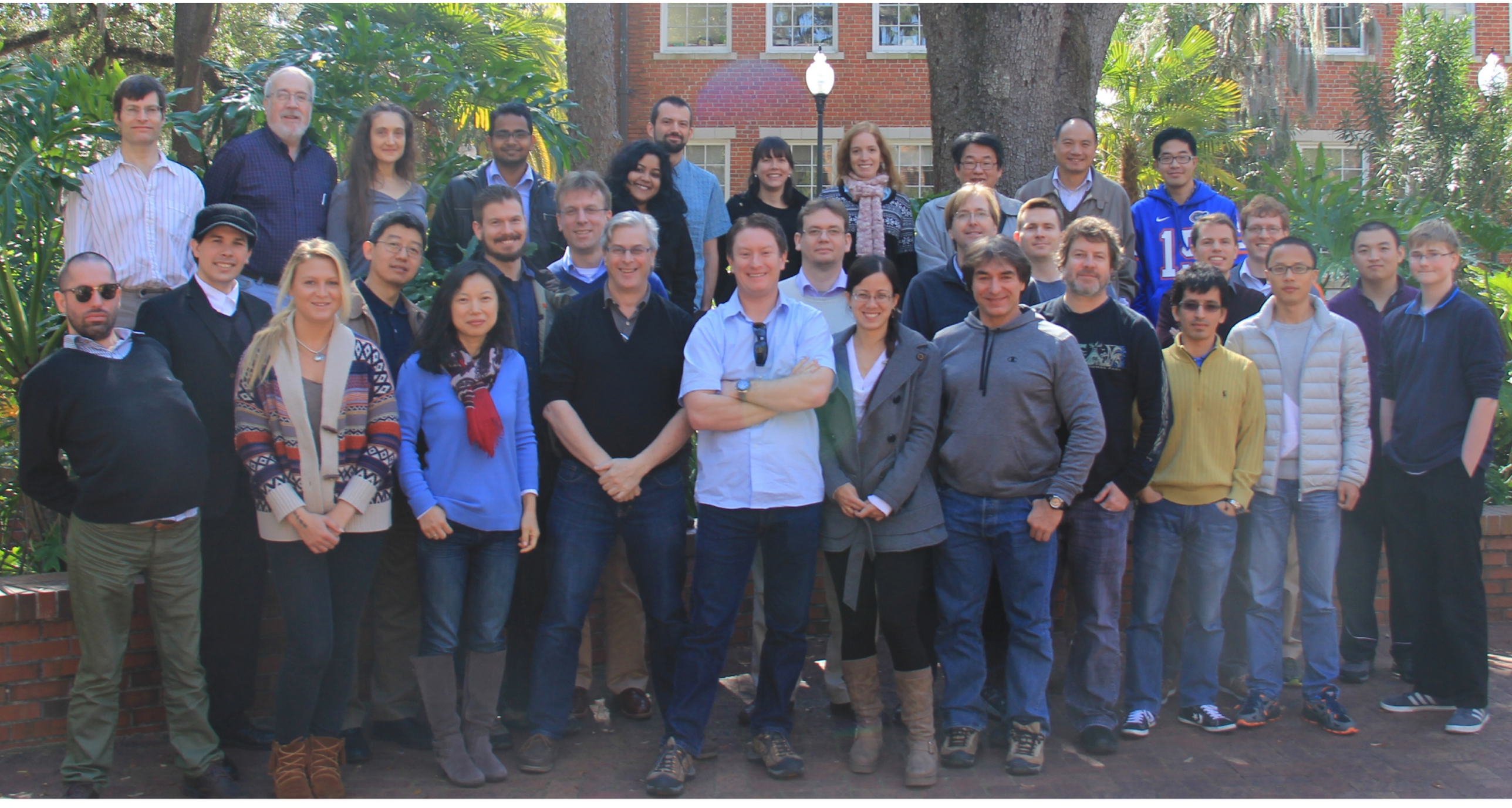
11:0: (2010) **git tree, full F90, make depend**

Challenge: building/patching the code!

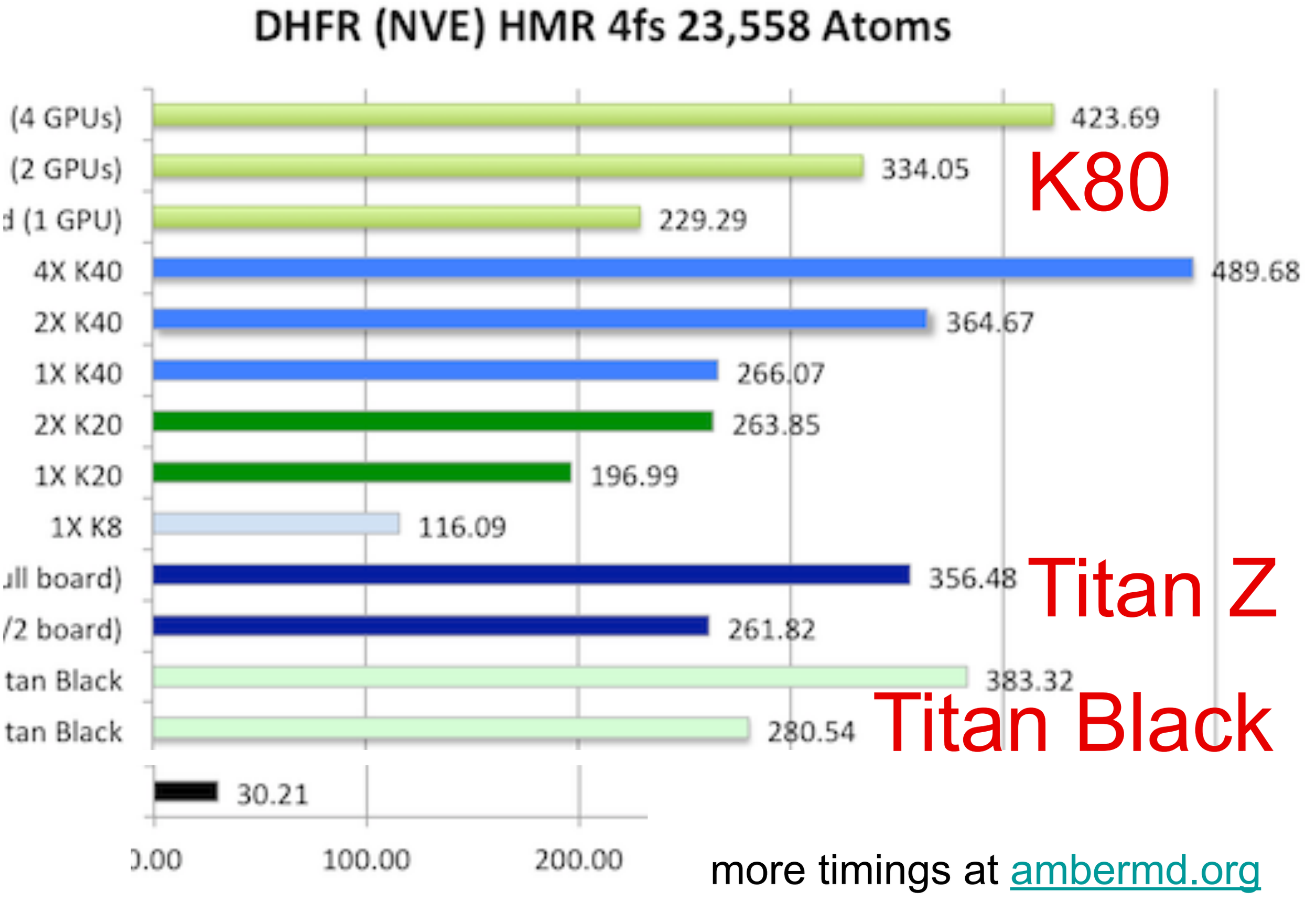
12.0: (2012) **Unified "configure" script, easy compile, …**
!!! automatic bug patching !!!

14.0: (2012) **GPU 1.23x, multi-GPU on node ||, …**

AMBER 18 (released ~May 2018)



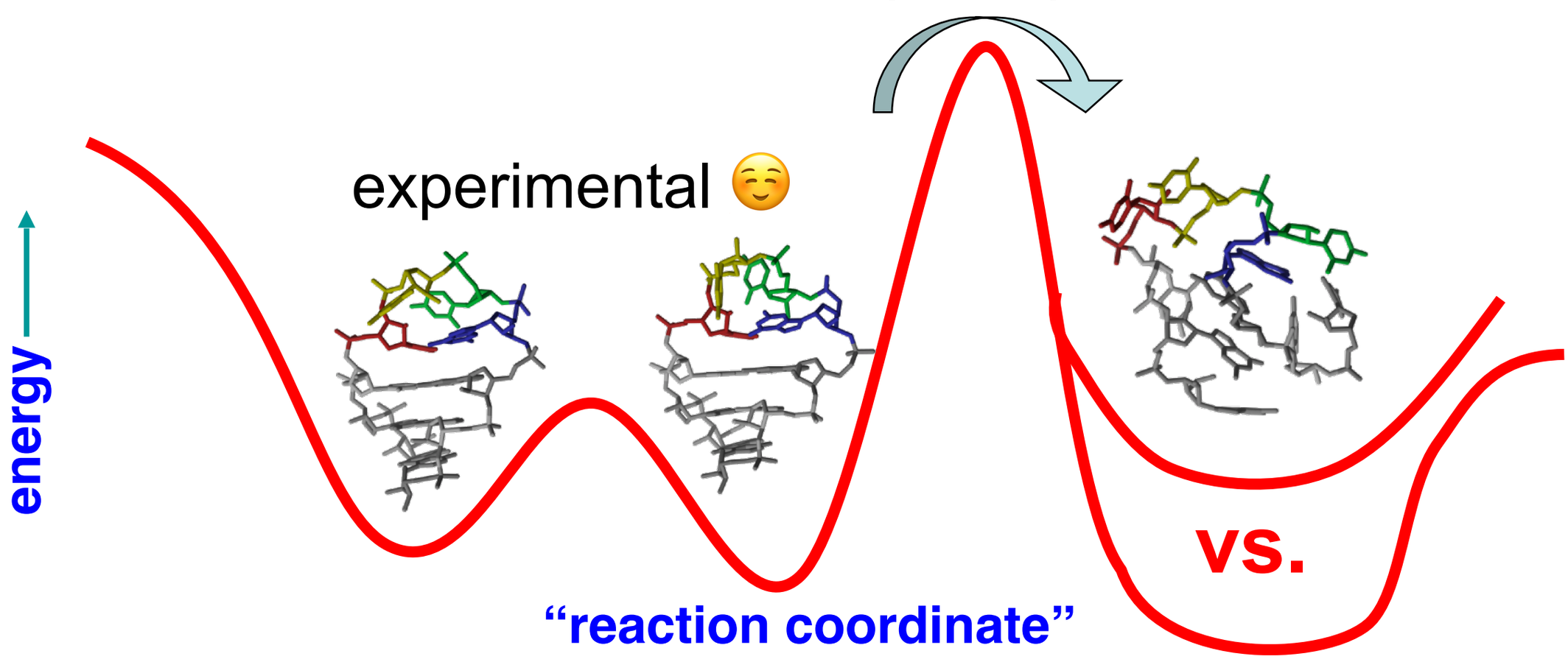
TitanXP 646 ns/day



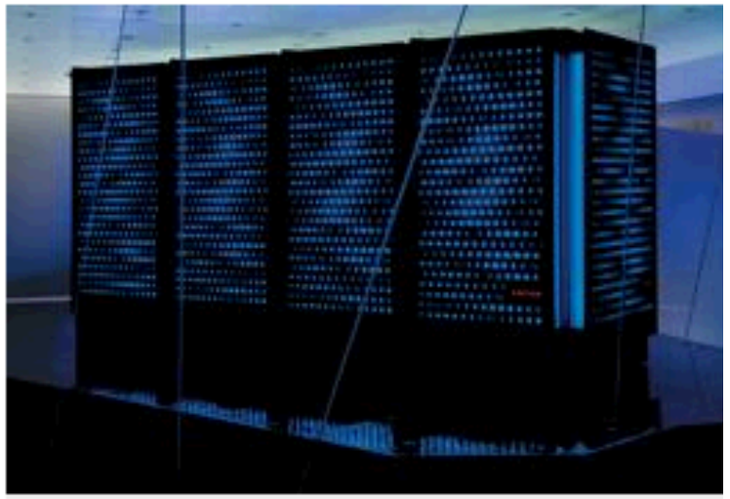
more timings at ambermd.org

are the force fields reliable?
(structure, dynamics, free energies)
can we fully sample the conformational ensemble?
(convergence, reproducibility)

Computer power?



How to fully sample conformational ensemble?



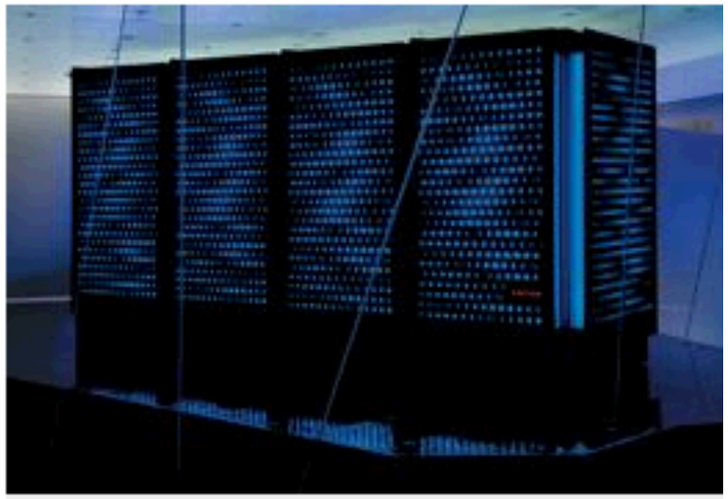
brute force – long contiguous in time MD
requires: special purpose / unique hardware

D.E. Shaw's Anton machine

Simulating protein movements using Anton could aid drug design.

SCIENCE/AAAS

How to fully sample conformational ensemble?

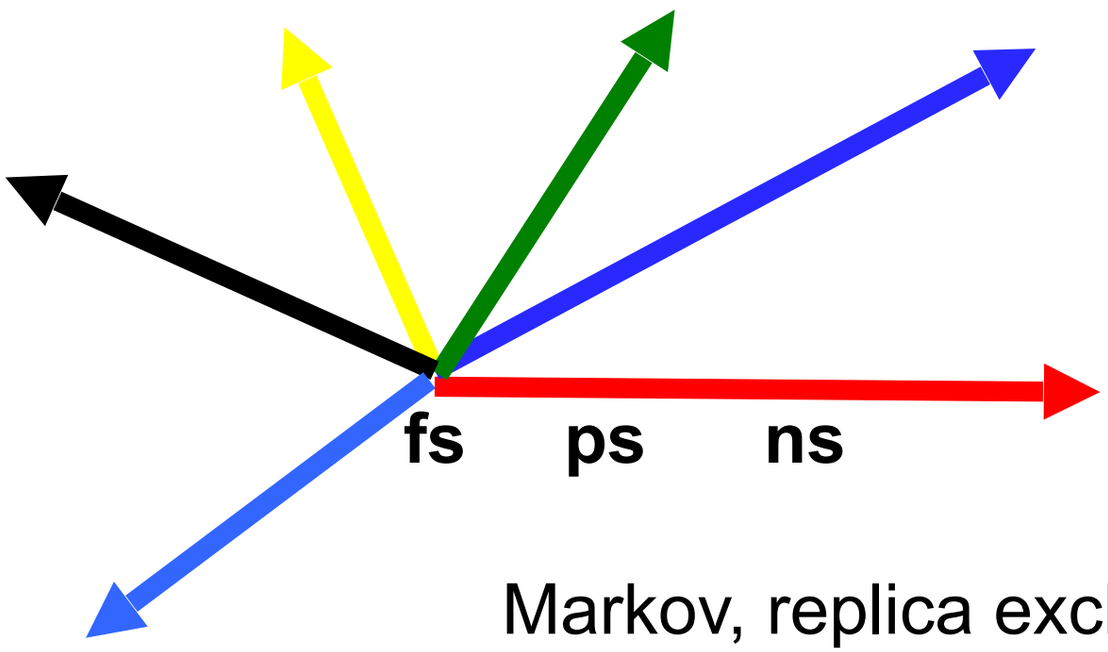


brute force – long contiguous in time MD
requires: special purpose / unique hardware

D.E. Shaw's Anton machine

Simulating protein movements using Anton could aid drug design.

SCIENCE/AAAS



ensembles of independent simulations

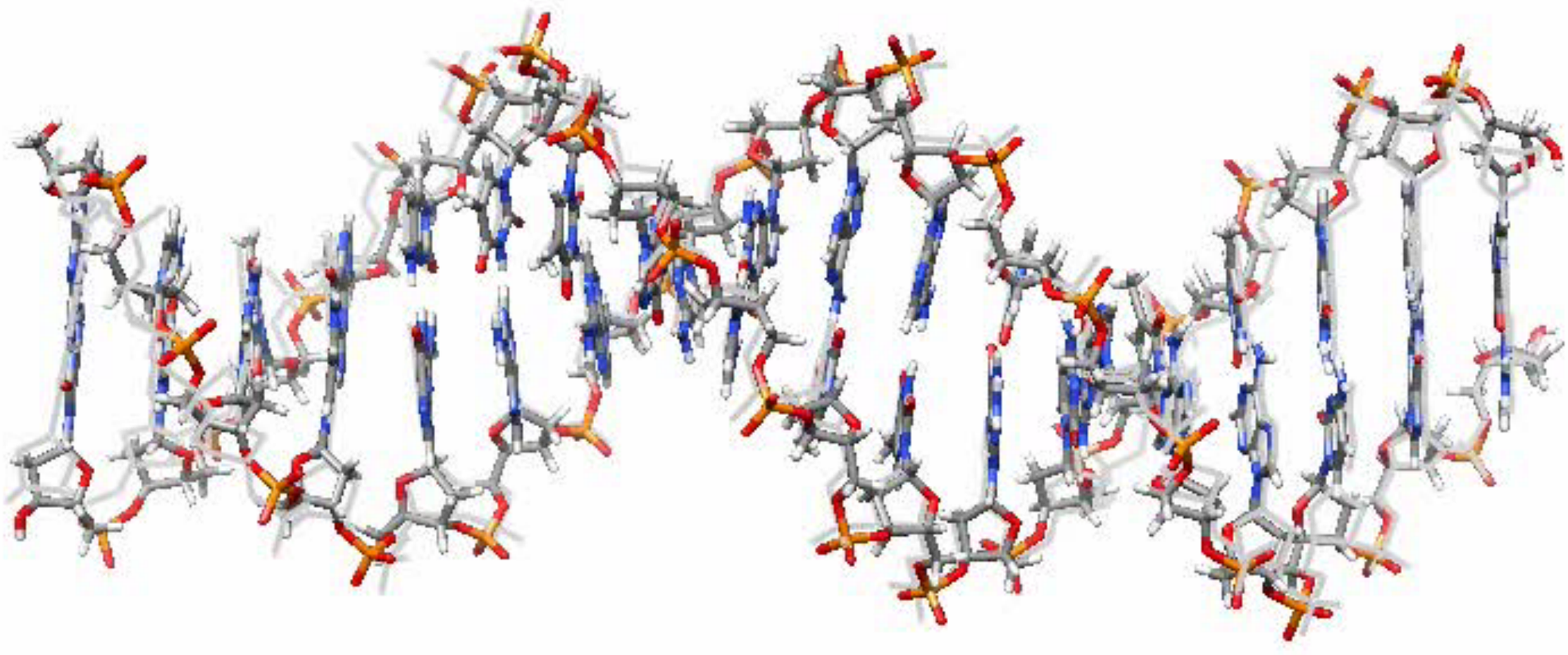
Markov, replica exchange, swarms, …

AMBER on GPUs

646 ns/day!
TitanXP

Convergence, force field and salt dependence in simulations of nucleic acids

d(GCACGAACGAACGAACGC) – Anton vs. GPUs



2 ns intervals (10 ns running average), render every 5th frame: ~10 us total time

How to test for convergence between two simulations?

- Aggregate independent runs into a single trajectory
- Calculate principal components and/or clustering
- Project principal components independently on each separate run, compare cluster populations between individual runs
- Visualize results

2013

PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data

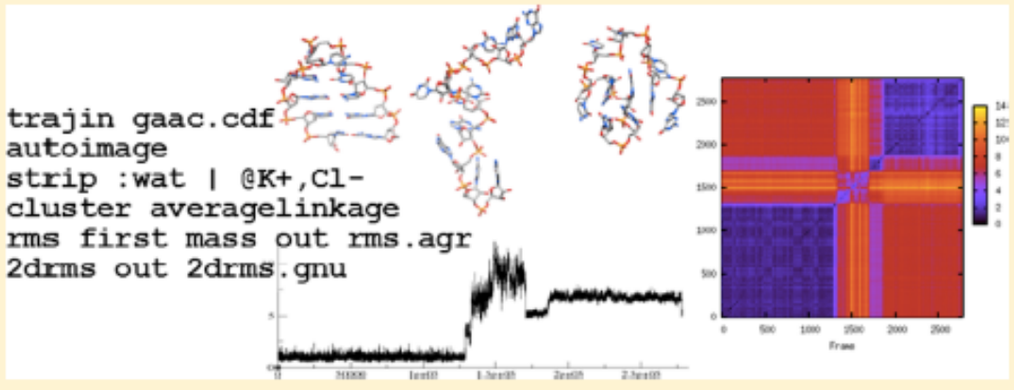
Daniel R. Roe* and Thomas E. Cheatham, III*

Department of Medicinal Chemistry, College of Pharmacy, 2000 South 30 East Room 105, University of Utah, Salt Lake City, Utah 84112, United States

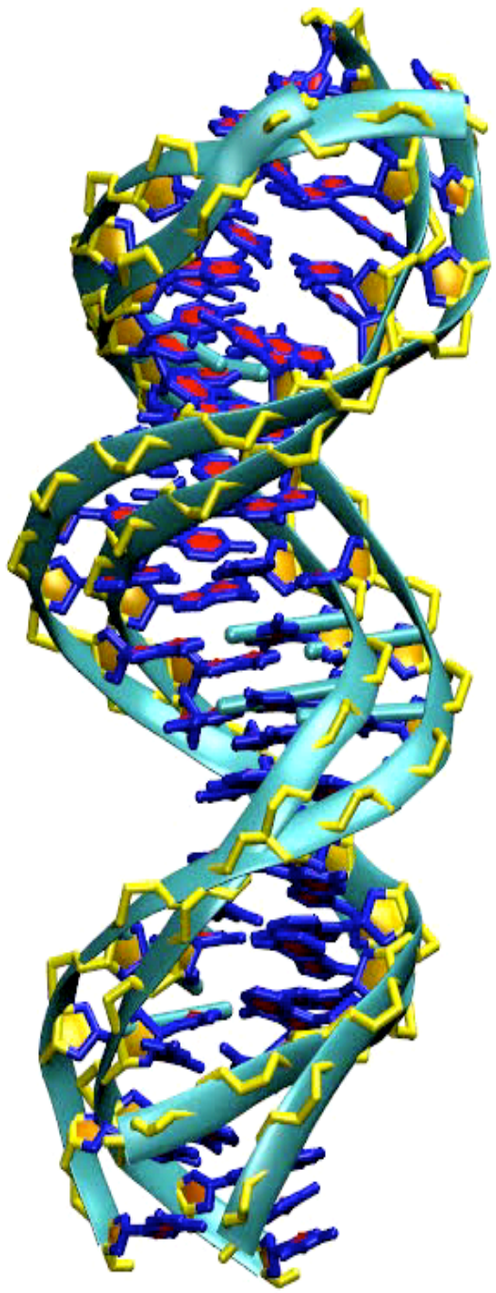
JCTC Journal of Chemical Theory and Computation

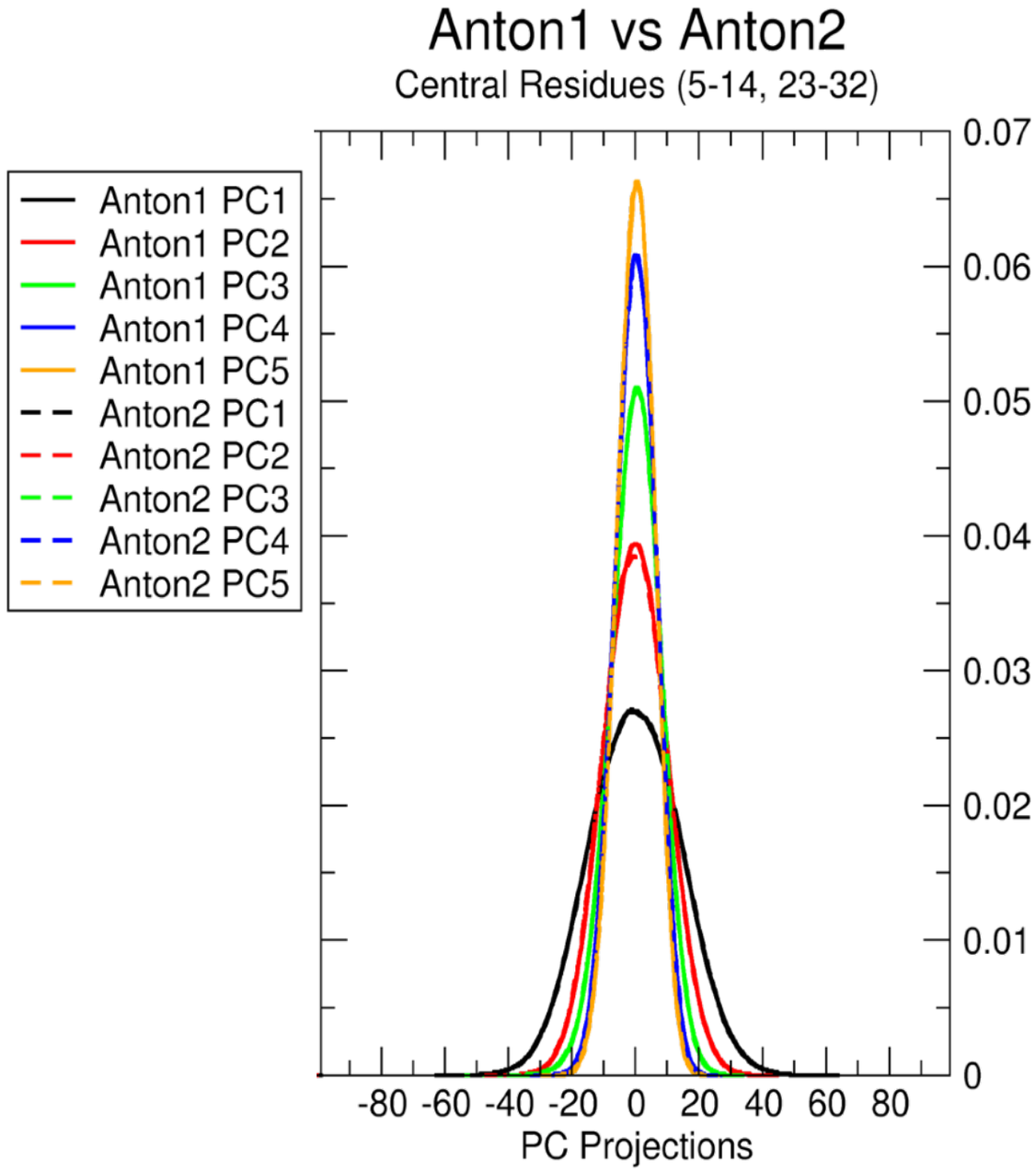
Supporting Information

ABSTRACT: We describe PTRAJ and its successor CPPTRAJ, two complementary, portable, and freely available computer programs for the analysis and processing of time series of three-dimensional atomic positions (i.e., coordinate trajectories) and the data therein derived. Common tools include the ability to manipulate the data to convert among trajectory formats, process groups of trajectories generated with ensemble methods (e.g., replica exchange molecular dynamics), image with periodic boundary conditions, create average structures, strip subsets of the system, and perform calculations such as RMS fitting, measuring distances, B-factors, radii of gyration, radial distribution functions, and time correlations, among other actions and analyses. Both the PTRAJ and CPPTRAJ programs and source code are freely available under the GNU General Public License version 3 and are currently distributed within the AmberTools 12 suite of support programs that make up part of the Amber package of computer programs (see http://ambermd.org). This overview describes the general design, features, and history of these two programs, as well as algorithmic improvements and new features available in CPPTRAJ.

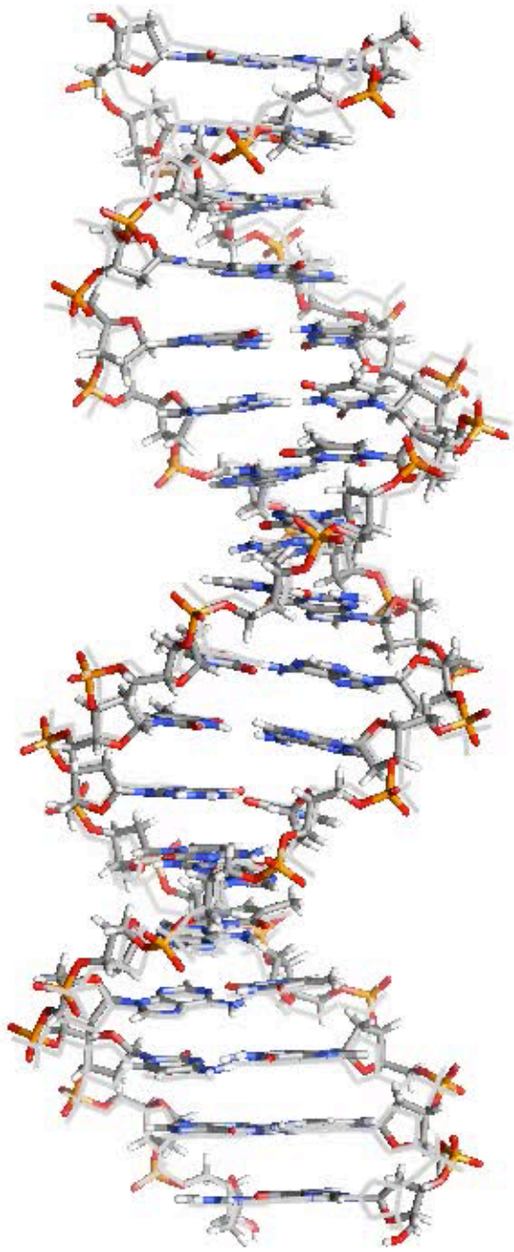


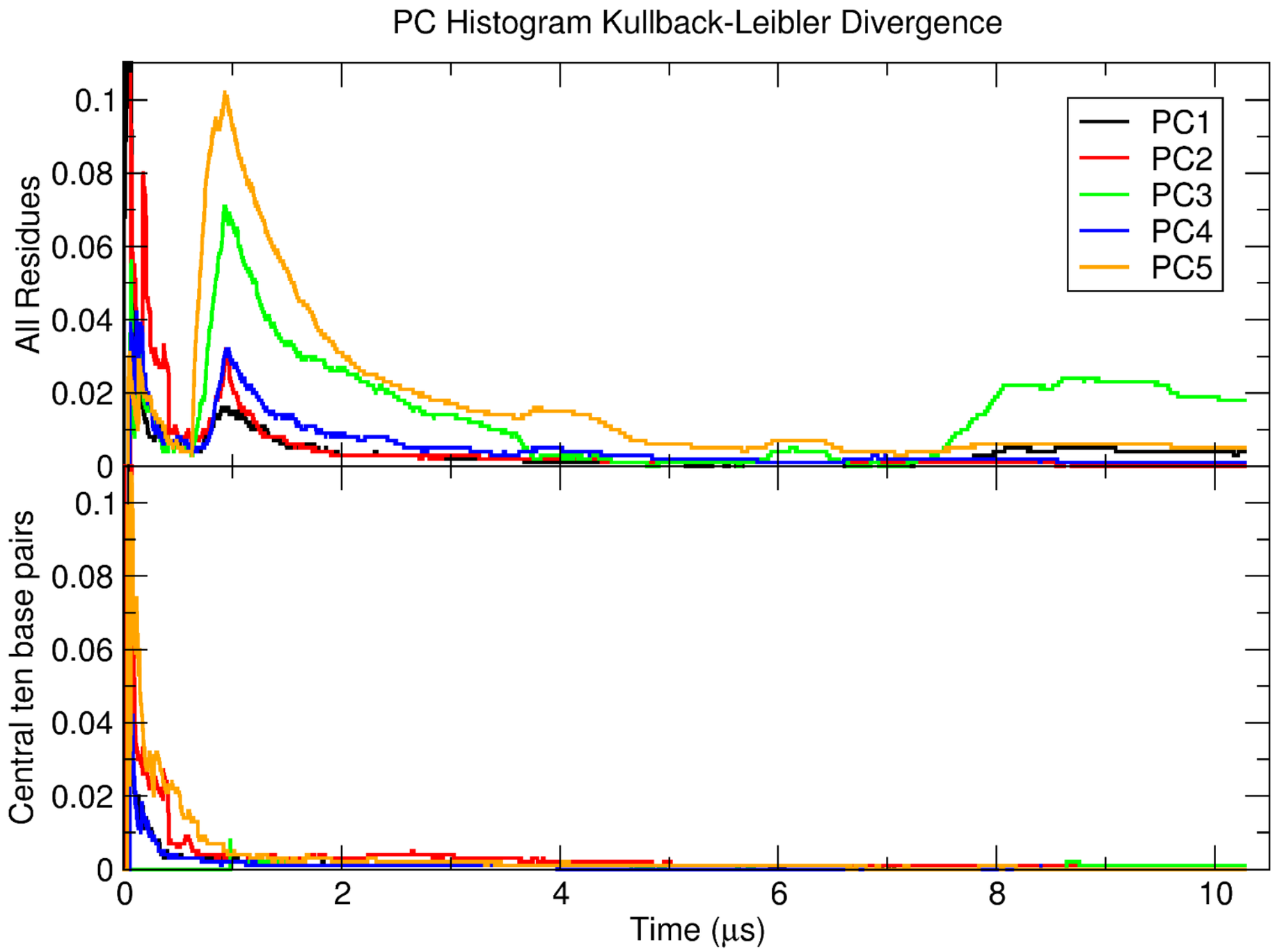
Test for convergence within and between simulations…





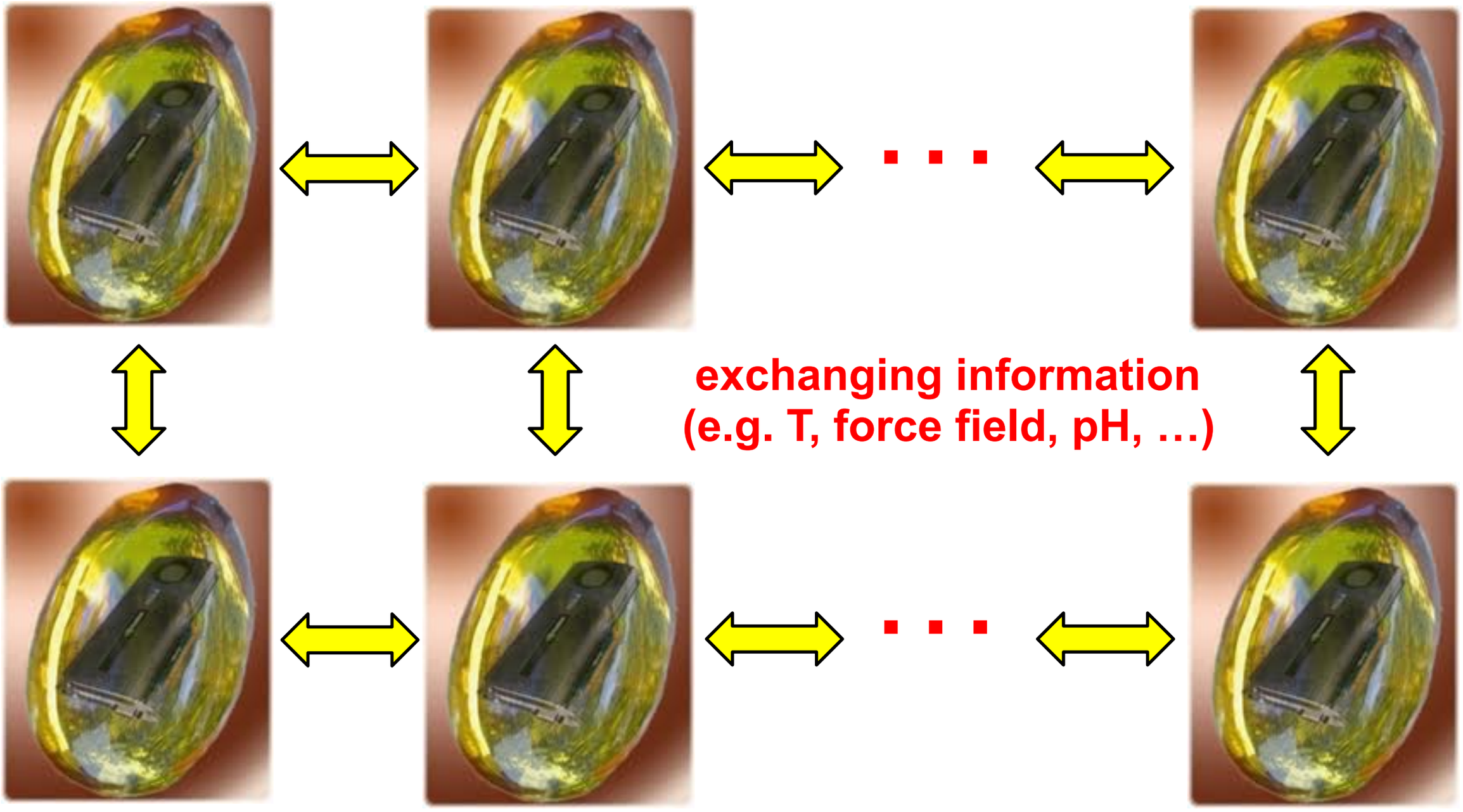
Test for convergence within and between simulations…





If we cannot scale to larger machine (more cores), couple independent MD simulations: i.e., use ensembles (replica exchange, | | tempering, Markov State modeling, ...)

independent ||
MD engines

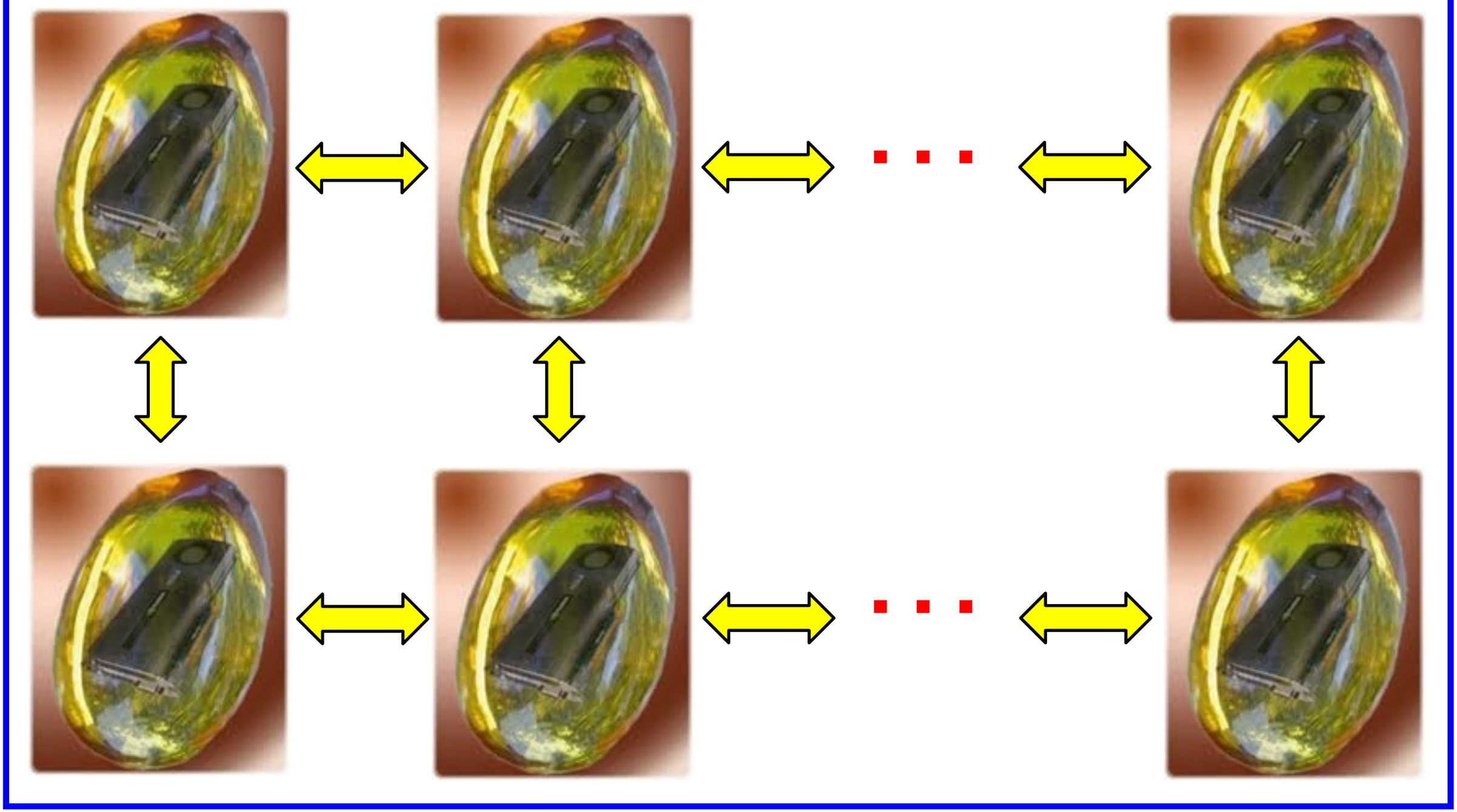


exchanging information
(e.g. T, force field, pH, …)

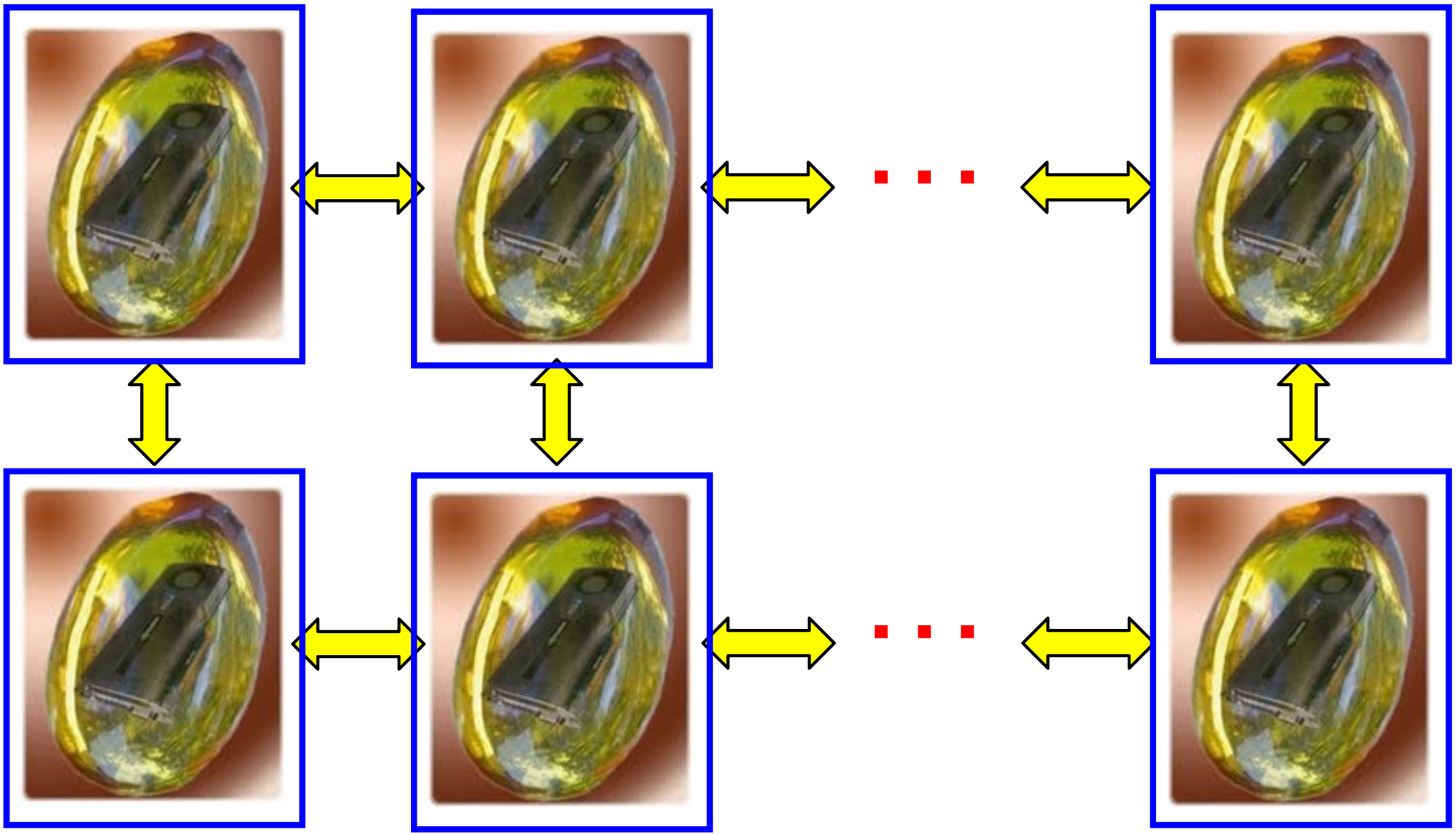
```
! o  All MPI communications should be done using the new
!    communicators rather than MPI_COMM_WORLD.  A number
!    of new communicators are defined:
!     CommSander  -- communications within a given sander job
!                    (replaces MPI_COMM_WORLD)
!     CommWorld   -- communications to ALL processors across
!                    multiple sander jobs
!     CommMaster  -- communications to the master node of each
!                    separate sander job each has corresponding
!                    size and rank,
!                    i.e. MasterRank, MasterSize
```

```
CommWorld = MPI_COMM_WORLD
call mpi_comm_rank( CommWorld, worldrank,  ierror )
call mpi_comm_size( CommWorld, worldsize,  ierror )
call mpi_barrier( CommWorld, ierror )
```

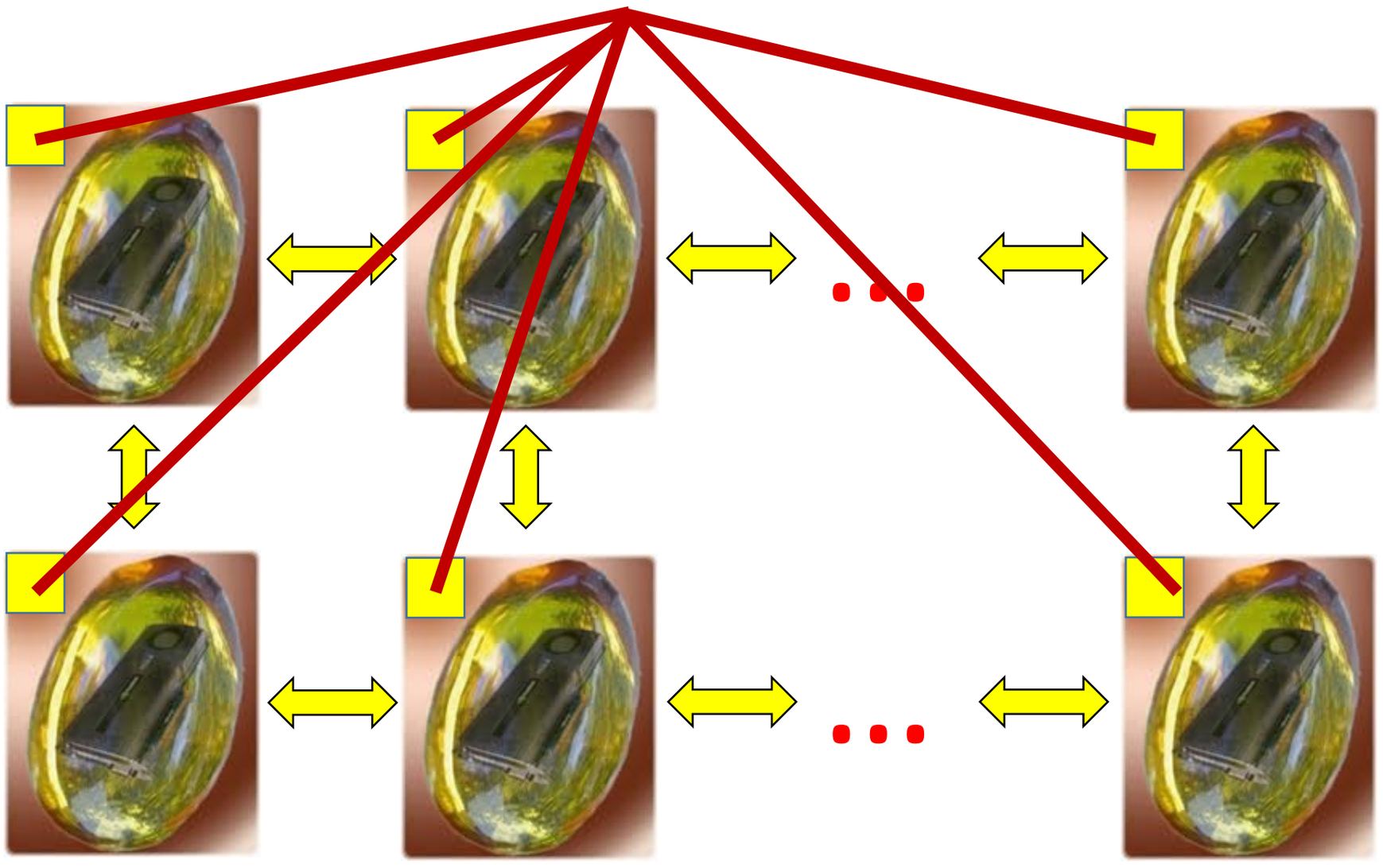
CommWorld



CommSander



CommMaster



```
! Create a communicator for each group of -ng NumGroup processors

   commsander = mpi_comm_world
   sandersize = worldsize
   sanderrank = worldrank
   nodeid = mod(worldrank, numgroup)
   if (numgroup > 1) then
      commsander = mpi_comm_null
      call mpi_comm_split(commworld, nodeid, worldrank, &
            commsander, ierror)
      if (commsander == mpi_comm_null) then
         if (worldrank == 0) then
            write(6,'(a,i5,a,i5)') 'Error: NULL Communicator', &
                  ' on PE', worldrank, ' from group ', nodeid
         end if
         call mexit(6,1)
      end if
      call mpi_comm_size(commsander, sandersize, ierror)
      call mpi_comm_rank(commsander, sanderrank, ierror)
end if
```

```
!  Define a communicator (CommMaster) that only talks between the local
!  "master" in each group.  This is equivalent to a SanderRank .eq. 0

  masterid = 0
  masterrank = MPI_UNDEFINED
  mastersize = 0
  if (numgroup > 1) then
     commmaster = mpi_comm_null
     if(sanderrank /= 0) then
        masterid = MPI_UNDEFINED
     end if

     call mpi_comm_split(commworld, masterid, worldrank, &
           commmaster, ierror)
     ! will this be emitted when using the default MPI error handler ?
     if (ierror /= MPI_SUCCESS) then
        write(6,*) 'Error: MPI_COMM_SPLIT error ', ierror, &
              ' on PE ', worldrank
     end if

     if(commmaster /= mpi_comm_null) then
        call mpi_comm_size(commmaster, mastersize, ierror)
        call mpi_comm_rank(commmaster, masterrank, ierror)
     end if
  end if
```

Can converge r(GAAC) in 1 day, a tetraloop in ~1-2 weeks!!

Production MD is no longer rate limiting step in workflow!

Setup, analysis, data management, …

Needs:

- ensemble management tools
- workflow tools
- data management solutions
- means to compare and share research results and codes

BW PAID & CPPTRAJ developments

levels of parallelism

(1) Ensemble processing (in || with MPI) – M-REMD

- convergence, reproducibility

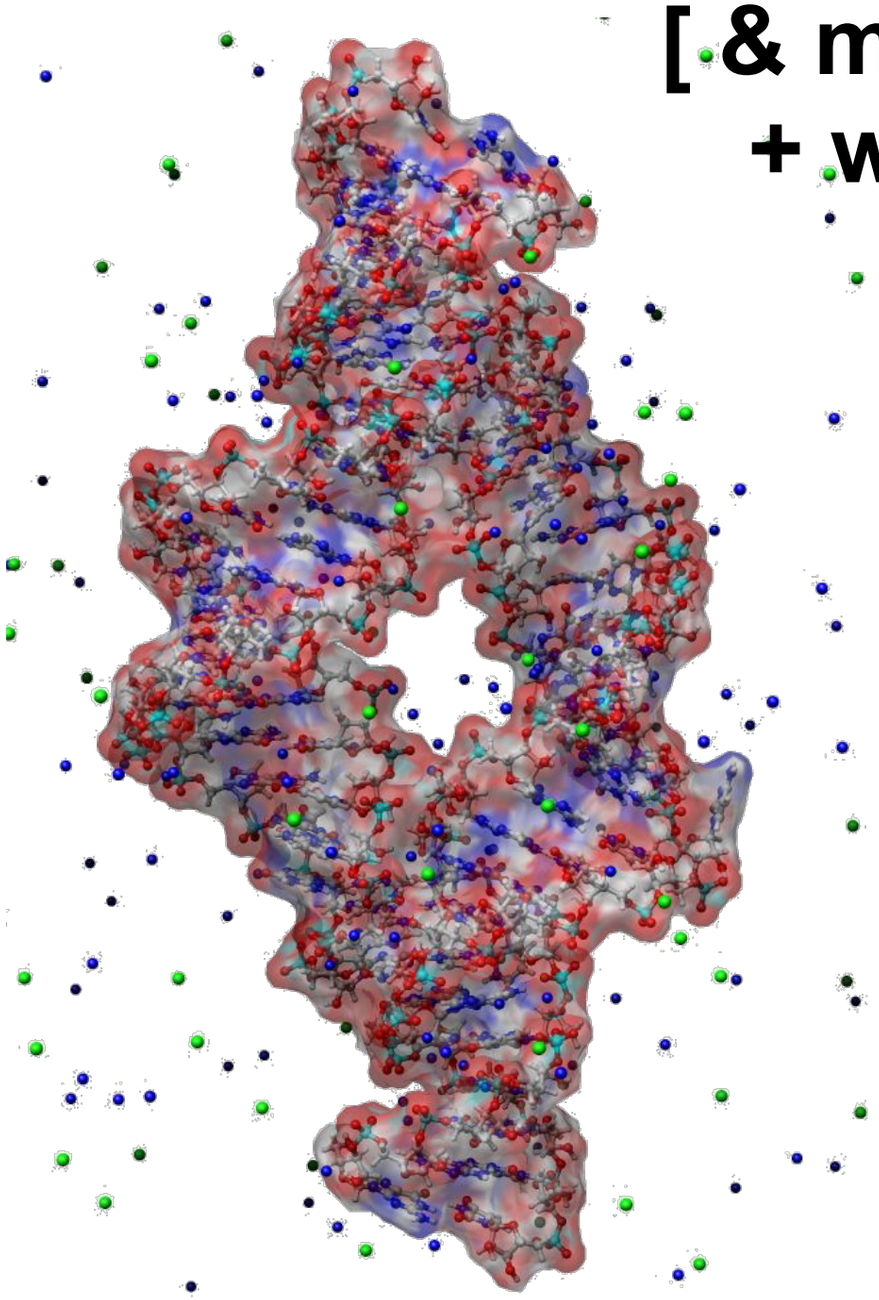
(2) MPI over file / intra-file level parallelization

(3) OpenMP for computational intensive analyses

(4) CUDA for time-consuming distance calculations

- Supports general datasets: 1D, 2D, …
- Interactive analysis on large memory resources
- [energetic analyses]
- support for more file formats (should add Desmond)

Peta- or exa- scale science: the problem will only get worse!



Solutions? Analysis "on the fly…"
[& more coarse-grained sampling]
+ workflow tools for ensembles

- **Do not move the data (?)**
- **Tiered resources**
- **Persistent storage**
- **Re-running the simulations**

…what will we miss? Can we only get low hanging fruit?

final thoughts (tec3)

- code development / optimization never stops
- practice good code / software engineering practices
- being able to code increases potential career opportunities
- GPUs (openACC vs. CUDA vs. multi-node)
- convergence, reproducibility, reliability, validation
- collaboration
- source code (and parameters) should be publicly available
- machines do not evolve to improve our performance, we need to evolve algorithms to the machine given to us…