Using BlueGene to characterize protein ligand interactions with DOCK and NAMD

Trent E. Balius and Sudipto Mukherjee

Rizzo Research Group

Dept. Applied Mathematics and Statistics,

Stony Brook University

e-mail: tbalius@ams.sunysb.edu

Setup and Test Using All Atom Molecular Dynamics on NY Blue

Overview

- Scientific Problem
  - Protein-Ligand binding and interaction (molecular recognition)
  - Molecular Dynamics
- Code Specifics
  - NAMD
  - AMBER
- Example Calculations
  - Scaling and Benchmarks
    - Number of processors
      - VN
      - CO
    - Number of atoms

Scientific Problem

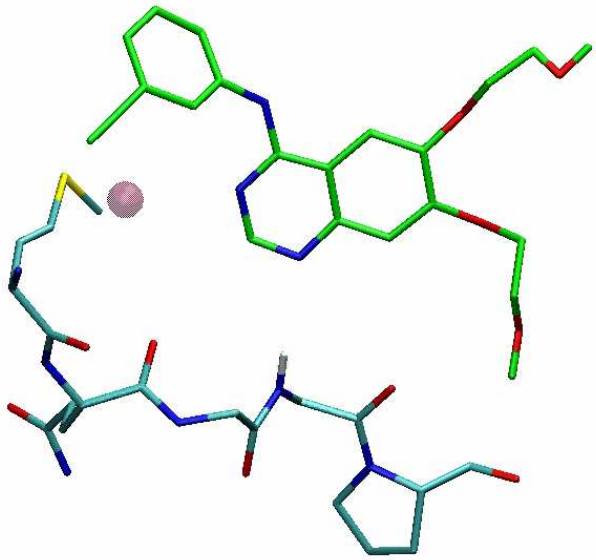
Molecular Dynamics

- Newton Equations

$$E\left(\mathrm{X}_{\text{posion}}\right)$$

$$F = -\nabla E$$

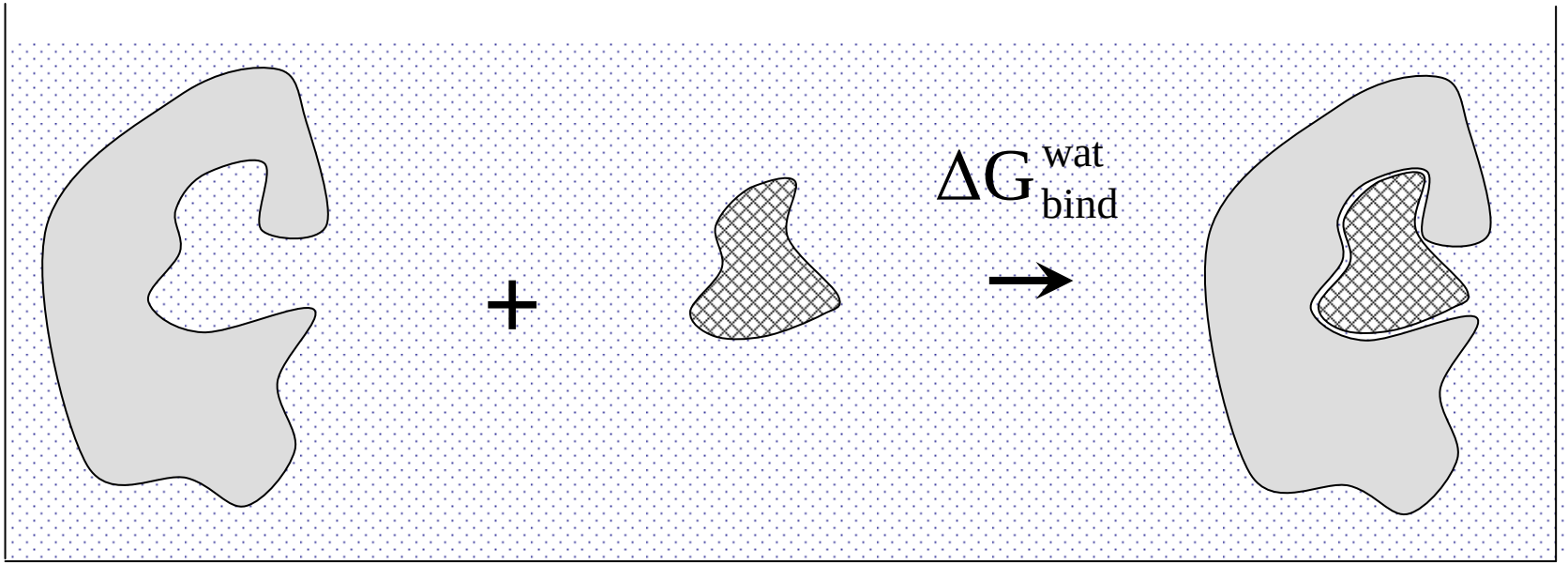
$$\mathrm{X}_{\text{posion}}^{\text{new}} = \frac{\partial^2}{\partial t^2}\left(\frac{F}{m}\right)$$



- ODE (velocity Verlet algorithm)
  - propagate to get motion

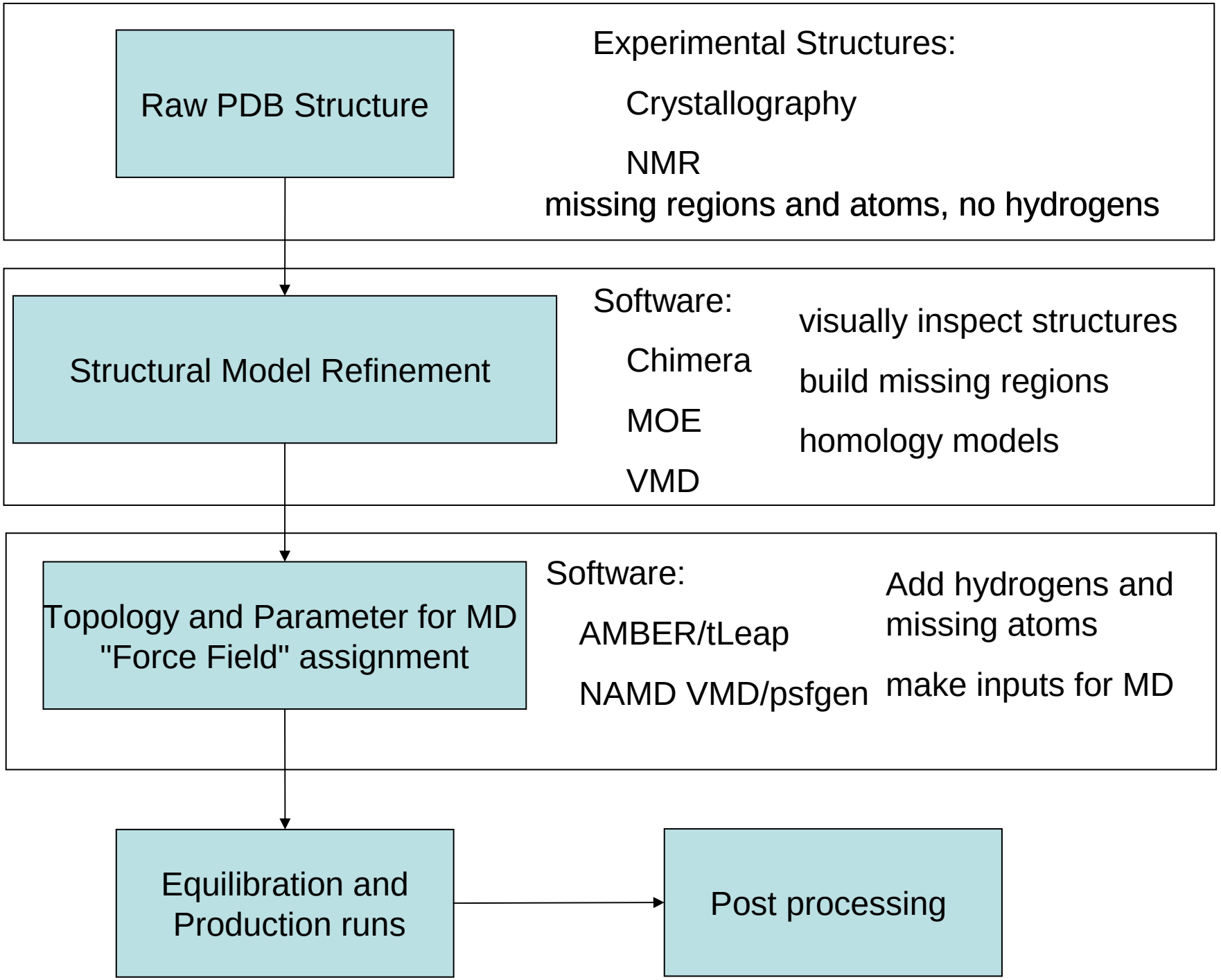
- We use MD to calculate binding energy and molecular interactions
  - sample conformations and binding modes
  - post process simulations

Binding Energy Calculation

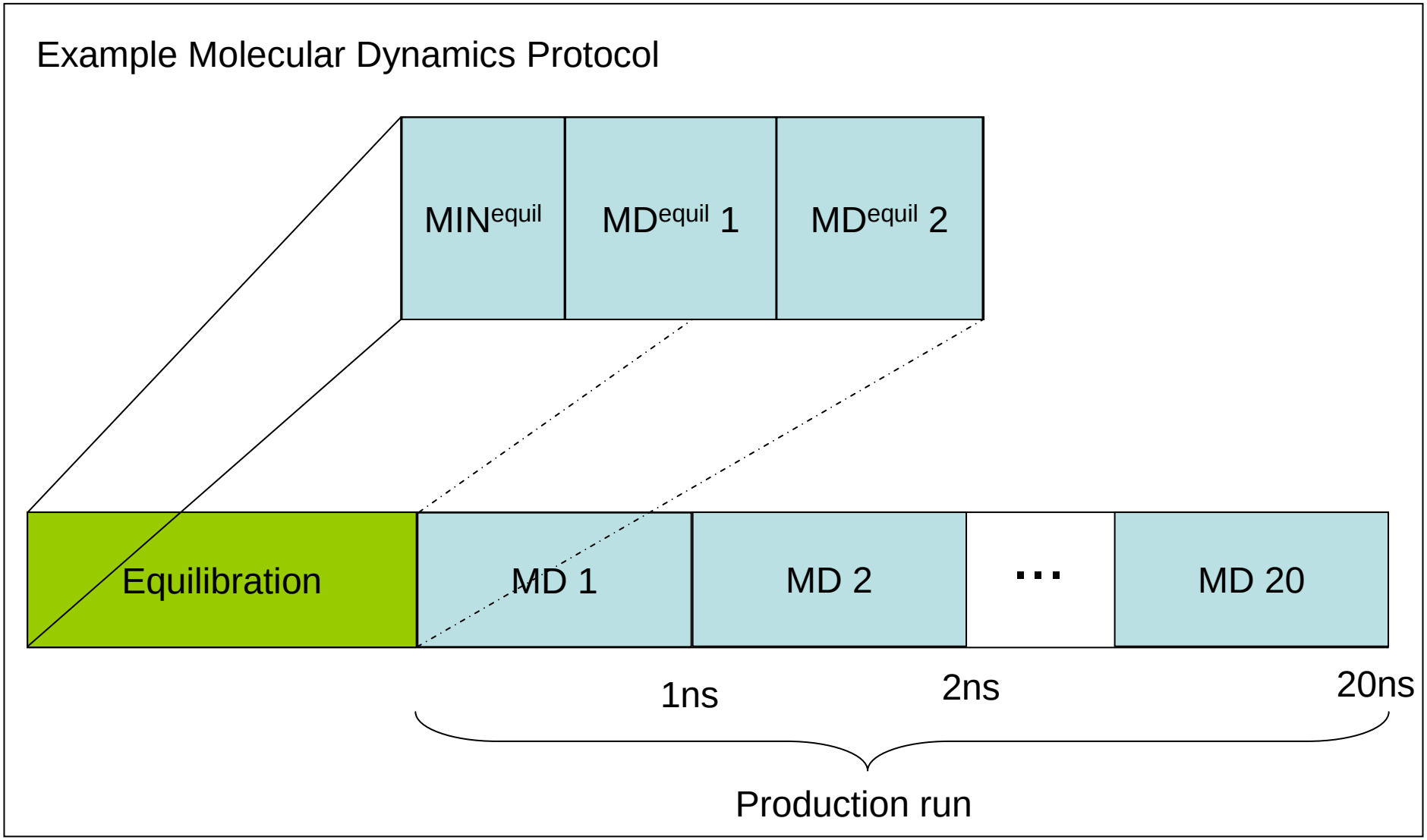


$$\Delta G^{wat}_{bind} = G^{wat}_{complex} - \left(G^{wat}_{ligand} + G^{wat}_{receptor} \right) \approx \Delta G^{exptl.}_{bind}$$

- simulate only the complex
  - post process
    - the energy of ligand, receptor and complex
    - MM-GBSA is used


Raw PDB Structure
Experimental Structures:
Crystallography
NMR
missing regions and atoms, no hydrogens
Structural Model Refinement
Software:
Chimera
MOE
VMD
visually inspect structures
build missing regions
homology models
Topology and Parameter for MD
"Force Field" assignment
Software:
AMBER/tLeap
NAMD VMD/psfgen
Add hydrogens and missing atoms
make inputs for MD
Equilibration and
Production runs
Post processing

MD running scheme



Code Specifics

Molecular Dynamics Codes

- Amber
  - Assisted Model Building with Energy Refinement
  - force field
  - suite of molecular simulation programs
  - http://amber.scripps.edu/
- Namd
  - NAnoscale Molecular Dynamics
  - molecular simulation program
  - http://www.ks.uiuc.edu/Research/namd/
- There are many other packages for MD (e.g. Gromacs)

Molecular Dynamics Codes (continued)

- Amber

  Pros

  – many functions

  – small molecule force field (gaff)

  Cons

  – poor scaling

  – currently only pmemd from amber 9 is available on NYBlue

    - pmemd is only pure md (no restraints or other functions)

- Namd

  Pros

  – scales well

  – Charmm force fields

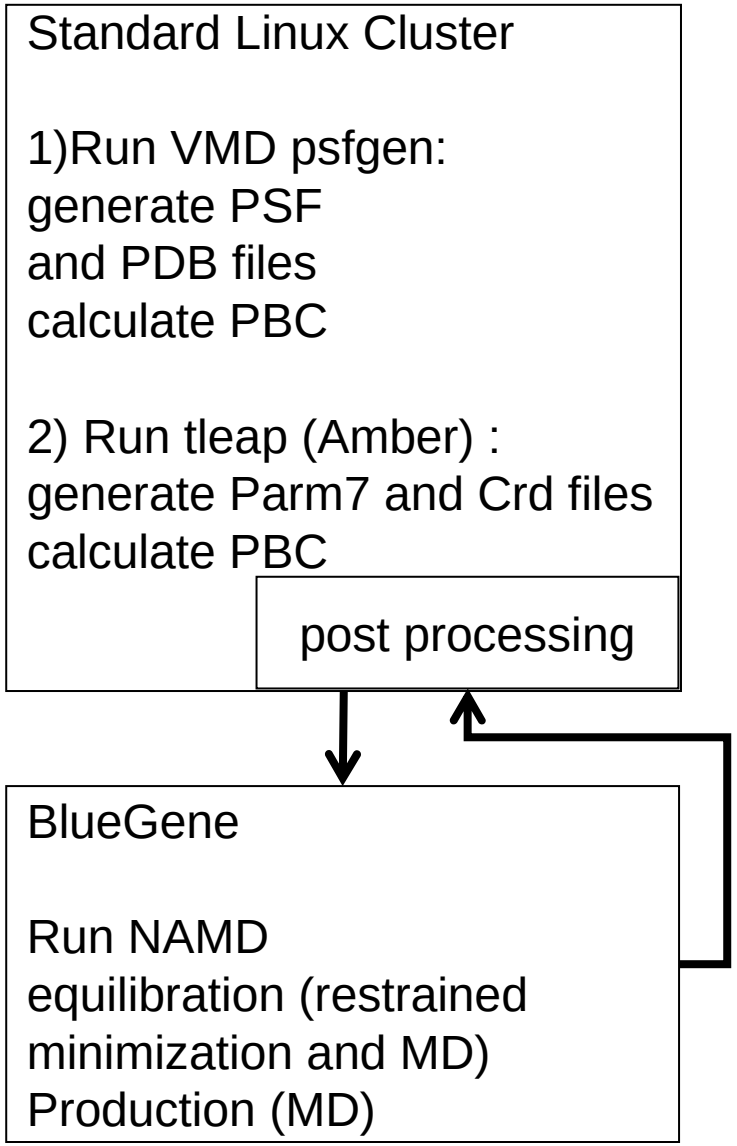
  – reads in Amber and Gromacs inputs

  Cons

  – no Charmm force field for small molecules

MD setup

- setup on other machine
  - seawulf.stonybrook.edu
  - cluster.bnl.gov
  - Amber setup
    - see namd manual
    - small molecules
  - Namd setup
- copy files to Bluegene
- Run equilibration and production on Bluegene
- post processing ?
  - viz. cluster ? (vis1-4)



MD running scheme

- long simulations (20 ns)
  - large files (file limit)
  - limit on wall clock

- divide the runs into 1ns or .5ns units
  - must finish in 48 hours
  - submit jobs in sequence

- can only have 5 jobs running or 9 jobs idling
  - e.g. 3 jobs running + 6 jobs idle
  - submit first in sequence
  - submit the rest

Molecular Dynamics Codes

- Information
  - How to be put on the Namd and Amber user lists
    - Leonard (Len) Slatest email: slatest@bnl.gov telephone: (631) 344 - 4102
- mpirun executable location

  /bgl/BlueLight/ppcfloor/bglsys/bin/mpirun32
- Namd executable location

  /apps/namd2.6-optimized/NAMD_2.6_BlueGeneL/namd2
- Amber executable location (amber 10 with sander coming soon)

  /apps/pmemd9_uses_massv/amber9/exe/pmemd

  or

  /apps/pmemd9_does_not_use_massv/amber9/exe/pmemd

NYBlue BlueGene/L

- loadleveler
  - Queuing system on NYBlue
  - go to BNL tutorial on job submission
    http://bluegene.bnl.gov/comp/running2.html
  - important commands

    /opt/ibmll/LoadL/full/bin/llsubmit
    - to submit jobs

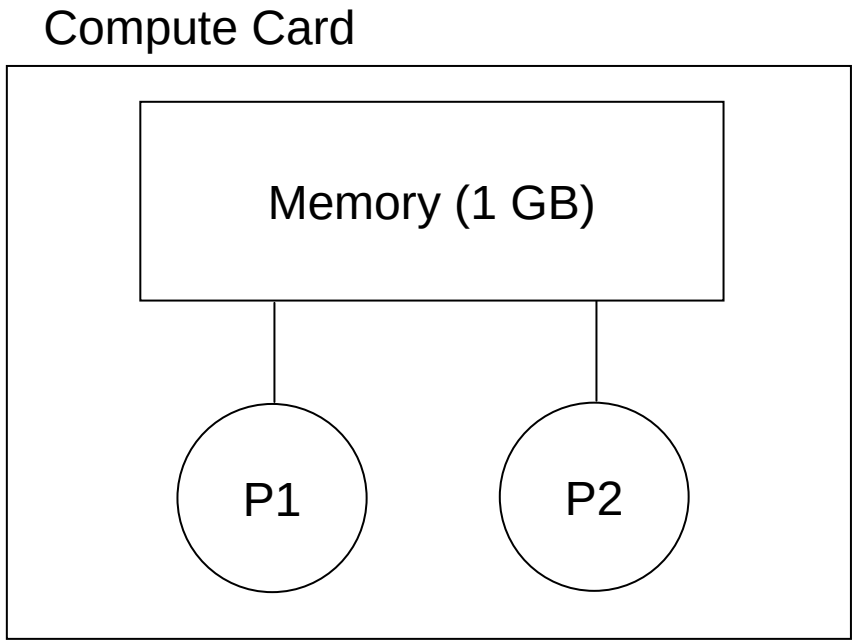
    llq
    - to see what is running

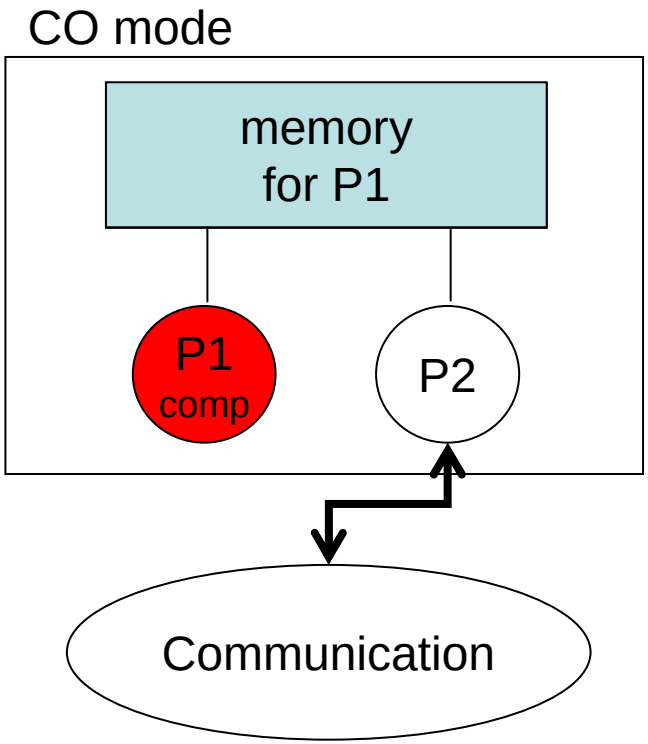
    llcancel
    - to delete jobs from the queue

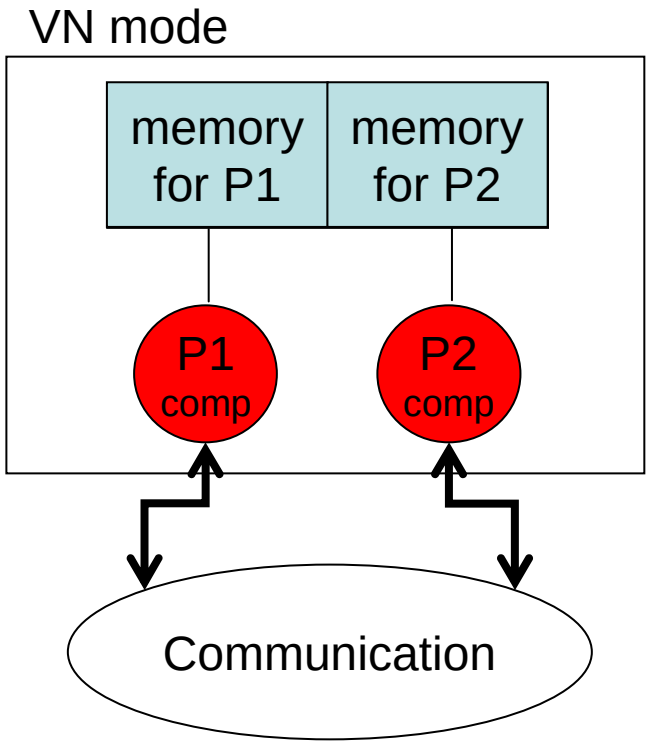
    readyblocks.pl
    - list free partitions

NYBlue BlueGene/L

- two modes CO or VN
- CO - Co-processor mode
- VN - Virtual node mode







load leveler inputs

```
# @ job_type = bluegene
# @ class = normal
# @ executable = mpirun32
# @ bg_partition = B01KTB01
# @ arguments =  -exe namd2 \
-cwd /gpfs/home2/username/workdir \
-args "/gpfs/home2/username/workdir/10md.in" \
-mode CO
# @ initialdir = /gpfs/home2/username/workdir
# @ input = /dev/null
# @ output = 10md.out
# @ error = 10md.err.$(jobid)
# @ wall_clock_limit = 4:00:00
# @ notification = complete
# @ queue
```

mode can be CO or VN

Namd input

```
#input files (section 3.2.1)
coordinates 09md.coor                          #name of coordinate (pdb) file
structure  GP41.WILD_t20.new.solvated.psf      #psf file
parameters par_all27_prot_lipid.prm            #parameter file
paratypecharmm          on                     #specifies if this is a charmm
                                               #force field (on or off)
velocities 09md.vel                            #velocity file for a restart note
                                               #that in a restart delete the temp


#output file (section 3.2.2)
outputname 10md                          #specifies the prefix for the output files

#basic dynamics (section 5.3.3)
exclude                 scaled1-4        #which bonded atom pairs are excluded
                                         #from non-bonded calculations
delete on restart
1-4scaling              1.0              #1.0 for Charmm, 0.833333 for Amber
rigidbonds              water            #controls how shake is used
rigidTolerance          0.00001          #allowable bond length error for shake
```

Namd with Amber inputs

```
#input files (section 3.2.1)
amber                   on        #specifies we are using AMBER Prm and CRD files
parmfile amber.parm               #specifies we are using AMBER Prm and CRD files
coordinates 09md.coor             #specifies we are using AMBER Prm and CRD files
velocities 09md.vel               #velocity file for a restart note that in a
                                  #restart delete the temp

#output file (section 3.2.2)
outputname 10md                   #specifies the prefix for the output files


#basic dynamics (section 5.3.3)
exclude                 scaled1-4 # which bonded atom pairs are excluded from
                                  # non-bonded calculations
1-4scaling              0.833333  #1.0 for Charmm, 0.833333 for Amber
scnb                    2         #This is default for both Amber and Charmm
rigidbonds              water     #controls how shake is used
rigidTolerance          0.00001   #allowable bond length error for shake
```

Namd inputs

```
#PME parameters (section 5.3.5)
PME                     yes             #turns PME on or off (yes=on no=off)
PMEGridSizeX            60              #number of grid points in X dimension
PMEGridSizeY            60              #number of grid points in Y dimension
PMEGridSizeZ            200             #number of grid points in Z dimension

#constraints (section 6.1)
constraints             on              #on or off
consref GP41.WILD_t20.new.solvated.pdb  #pdb file with restraint reference
positions
conskfile GP41.WILD_t20.restraint09.pdb #pdb file with force constant values
conskcol                B

#periodic boundry conditions (section 6.4.3)
cellbasisvector1 59.9879989624 0 0      #defines the first periodic boundary
cellbasisvector2 0 59.9939994812 0      #defines the second periodic boundary
cellbasisvector3 0 0 199.9529953001     #defines the third periodic boundary
cellorigin 0.212269335985 0.265642940998 -69.3950576782
#defines the xyz location of the center of the box
extendedSystem 09md.xsc                 #defines file which contains the PBC
                                        #info from previous runs

#equilibration
run 10000
#minimize Z-NSTEPS
```

Namd with Amber inputs

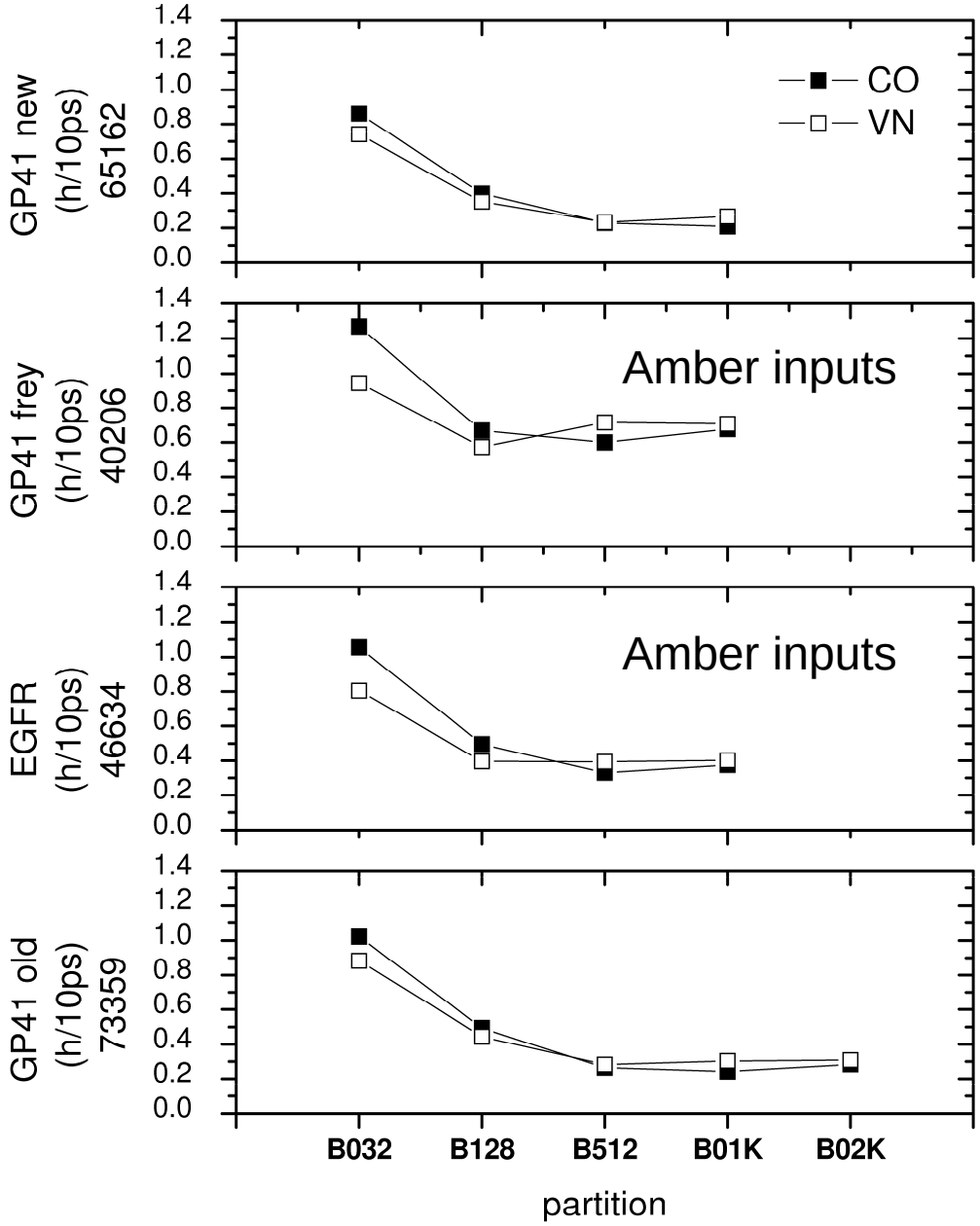
```
---AMBER----      ---NAMD---

 TITLE
 &cntrl
  ntc=2, ntf=2,   # SHAKE to the bond between each hydrogen and it mother atom
                  rigidBonds     all
  tol=0.0005,     rigidTolerance 0.0005  # Default is  0.00001
  nstlim=500,     numsteps       500  # Num of total steps
  ntpr=50,        outputEnergies 50  # Energy output frequency
  ntwr=100,       restartfreq    100  # Restart file frequency
  ntwx=100,       DCDfreq        100  # Trajectory file frequency
  dt=0.001,       timestep       1  # in unit of fs (This is default)
  tempi=300.,     temperature    300  # Initial temp for velocity assignment
  cut=9.,         cutoff         9
                  switching      off  # Turn off the switching functions
 &end
 &ewald           PME            on  # Use PME for electrostatic calculation
                  # Orthogonal periodic box size
  a=62.23,        cellBasisVector1   62.23  0  0
  b=62.23,        cellBasisVector2   0  62.23  0
  c=62.23,        cellBasisVector3   0  0  62.23
  nfft1=64,       PMEGridSizeX   64
  nfft2=64,       PMEGridSizeY   64
  nfft3=64,       PMEGridSizeZ   64
  ischrgd=1,      # NAMD doesn't force neutralization of charge
 &end
                  amber          on  # Specify this is AMBER force field
                  parmfile       FILENAME  # Input PARM file
                  ambercoor      FILENAME  # Input coordinate file
                  outputname     PREFIX  # Prefix of output files
                  exclude        scaled1-4
                  1-4scaling     0.833333  # =1/1.2, default is 1.0
```

NAMD User's Guide Version 2.6 page 26
http://www.ks.uiuc.edu/Research/namd/2.6/ug.pdf

Example Calculation: Benchmarks and Scaling

Namd Benchmarks



hours/10 ps

we run jobs in 1ns segments

1ns = 1000ps

we need to run ~5 and ~20 ns to get converged data

for this system "best bang for the buck" is B128

B512 is less efficient

note that this bench mark has 1 fs time step

Conclusions

- Molecular Dynamics on NYBlue
  - Namd
  - Amber
- Namd has good scaling

Issues

- best way to do post processing?

Resources

- http://ringo.ams.sunysb.edu/index.php/Rizzo_Lab_Information_and_Tutorials
- http://bluegene.bnl.gov/comp/running2.html
- http://bluegene.bnl.gov/comp/userGuide.shtml

Using DOCK to characterize protein ligand interactions

Sudipto Mukherjee

Robert C. Rizzo Lab

Acknowledgements

The Rizzo Lab

- Dr. Robert C. Rizzo
- Brian McGillick
- Rashi Goyal
- Yulin Huang

IBM Rochester

- Carlos P. Sosa
- Amanda Peters

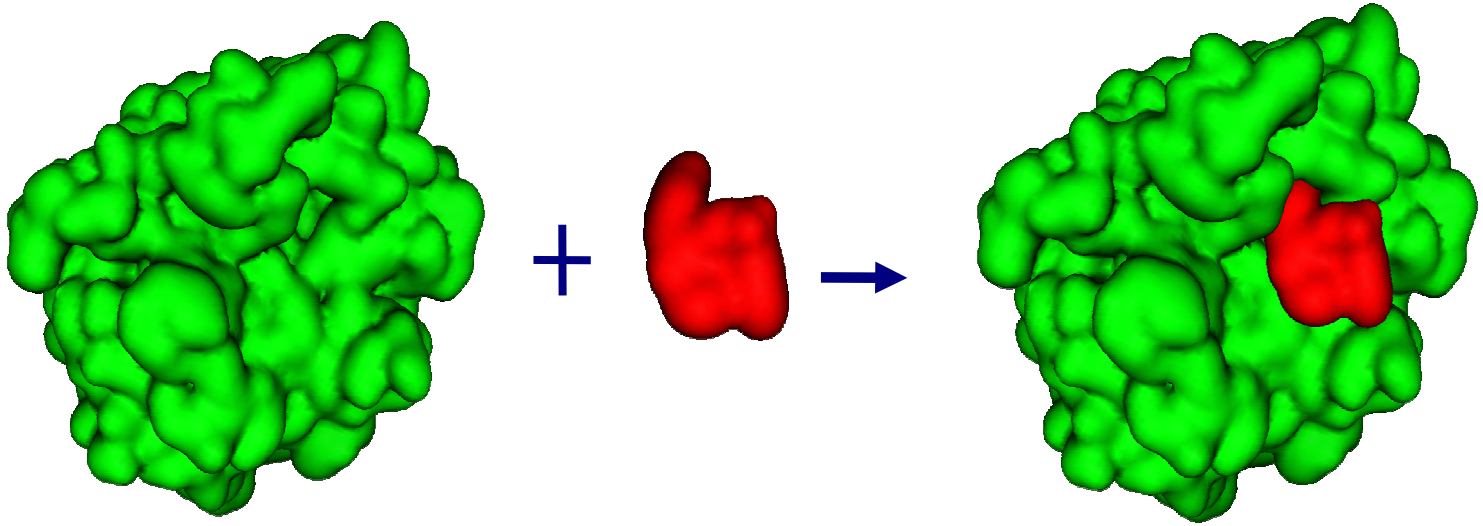
Support

- Stony Brook Department of Applied Mathematics and Statistics
- New York State Office of Science, Technology & Academic Research
- Computational Science Center at Brookhaven National Laboratory
- National Institutes of Health (NIGMS)
- NIH National Research Service Award.
  Grant Number:1F31CA134201-01. (Trent E. Balius)

Introduction

- What is Docking?
- Compilation of DOCK on BG
- Scaling Benchmarks

Docking as a Drug Discovery Tool



Docking : Computational Search for energetically favorable binding poses of a ligand with a receptor. Find origins of ligand binding which drive molecular recognition.

Finding the correct pose, given a ligand and a receptor.
Finding the best molecule, given a database and a receptor.

- Conformer Generation
- Shape Fitting
- Scoring Functions
- Pose Ranking

Docking Resources

- Small Molecule Databases
  - NCI (National Cancer Institute)
  - UCSF ZINC zinc.docking.org
- Protein receptor structure
  - Protein Data Bank www.rcsb.org/
- Docking Tutorials
  - Rizzo Lab Wiki http://ringo.ams.sunysb.edu/index.php/DOCK_tutorial_with_1LAH
  - UCSF Tutorials dock.compbio.ucsf.edu/DOCK_6/index.htm
  - AMS535-536 Comp Bio Course Sequence
- Modeling Tools
  - Chimera (UCSF)

Compiling DOCK6 on BlueGene

IBM XL Compiler Optimizations

- O5 Level Optimization
  - qhot Loop analysis optimization
  - qipa Enable interprocedural analysis
- PowerPC Double Hummer (2 FPU)
  - qtune=440 qarch=440d
- MASSV Mathematical Acceleration Subsystem
  - -lmassv

DOCK Accessory programs not ported

- Energy Grid files must be computed on FEN, not on regular Linux cluster because of endian issues

High Throughput Computing Validation for Drug Discovery Using the DOCK Program on a Massively Parallel System

Thanks to Amanda Peters, Carlos P. Sosa (IBM) for compilation help

Compiling Dock on BG/L

Cross-compile on Front End Node with Makefile parameters for IBM XL Compilers

```
CC =      /opt/ibmcmp/vac/bg/8.0/bin/blrts_xlc
CXX =     /opt/ibmcmp/vacpp/bg/8.0/bin/blrts_xlC

BGL_SYS= /bgl/BlueLight/ppcfloor/bglsys

CFLAGS = -qcheck=all -DBUILD_DOCK_WITH_MPI -DMPICH_IGNORE_CXX_SEEK
         -I$(BGL_SYS)/include -lmassv -qarch=440d -qtune=440
         -qignprag=omp -qinline -qflag=w:w -O5 -qlist -qsource -qhot

FC =      /opt/ibmcmp/xlf/bg/10.1/bin/blrts_xlf90
FFLAGS = -fno-automatic -fno-second-underscore
LOAD =    /opt/ibmcmp/vacpp/bg/8.0/bin/blrts_xlC

LIBS =    -lm -L$(BGL_SYS)/lib -lmpich.rts -lmsglayer.rts
          -lrts.rts -ldevices.rts
```

Note that library files and compiler binaries are located in different paths on BG/L and BG/P

Compiling Dock on BG/P

```
CC=             /opt/ibmcmp/vac/bg/9.0/bin/bgxlc
CXX=            /opt/ibmcmp/vacpp/bg/9.0/bin/bgxlC

BGP_SYS=        /bgsys/drivers/ppcfloor

CFLAGS=         -L/opt/ibmcmp/xlmass/bg/4.4/bglib -lmassv -L-qcheck=all
                $(XLC_TRACE_LIB) -qarch=440d -qtune=440 -qignprag=omp
                -qinline  -qflag=w:w  -DBUILD_DOCK_WITH_MPI
                -DMPICH_IGNORE_CXX_SEEK -I$(BGP_SYS)/comm/include
                -O5  -qlist -qsource  -qhot

FC=             /opt/ibmcmp/xlf/bg/11.1/bin/bgxlf90
FFLAGS=         $(XLC_TRACE_LIB) -O3 -qlist -qsource  -qhot
                -fno-automatic -fno-second-underscore -qarch-440d
                -O3  -qlist -qsource -qhot -qlist -fno-automatic
                -fno-second-underscore

LOAD=           /opt/ibmcmp/vacpp/bg/9.0/bin/bgxlC
LIBS=           -lm -L$(BGP_SYS)/comm/lib   -lmpich.cnk -ldcmfcoll.cnk
                -ldcmf.cnk -L$(BGP_SYS)/runtime/SPI -lSPI.cna -lrt
                -lpthread -lmass
```

Dock scaling background

- Embarrassingly parallel simulation
  - No comm required between MPI processes
  - Each molecule can be docked independently as a serial process
  - VN mode should always be better
- Scaling bottlenecks
  - Disk I/O (need to read and write molecules and output file)
  - MPI master node is a compute node

- Scaling benchmarks were done with a database of 100,000 molecules with 48 hour time limit.
  - # of molecules docked is used to determine performance

- Typical virtual screening run uses ca. 5 million molecules.

Virtual Node mode

Mode (#of CPU's)	Molecules Docked
CO (128)	13363
VN (256)	24657

This is a check to verify that VN mode is about twice as fast as CO mode.

Protein = 2PK4, B128 BG/L block

BG/P has three modes with 1,2 or 4 processors available.

Protein = 2PK4, B064 BG/P block

BG/P B064 is almost twice as fast as BG/L B128 even though both have same # of CPU's

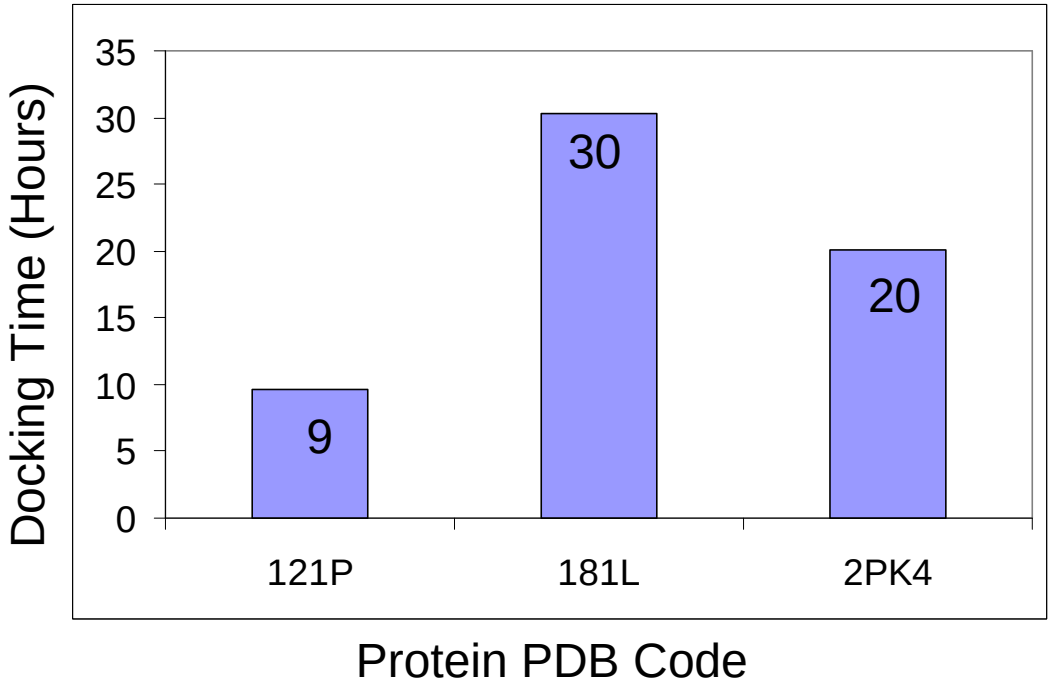
Mode	Molecules Docked
SMP (64)	15287
DUAL (128)	26152
VN (256)	40316

All simulations were allowed to run for the limit of 48 hours and benchmarked on the # of molecules docked within that time.

BG/P VN mode provides best scaling

Mode	# CPUs	121P	181L	1F8B	1VRT	2PK4
SMP	64	17366	10528	16036	18329	15287
DUAL	128	26449	19525	24729	25625	26152
VN	256	40412	30666	38002	40681	40316

Same simulation with 5 different system shows that BG/P in VN mode is best suited for virtual screening simulations. [B064 BG/P block]

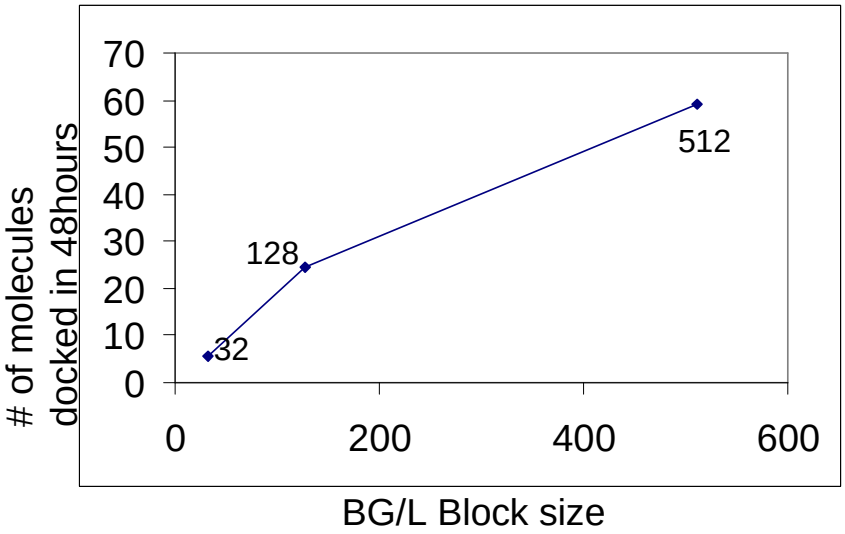


BG/P B512 block
VN mode = 2048 cpus

Timing varies widely with type of protein target

Timing in hours for Production Run of 100,000 molecules docked

Scaling Benchmark on BG/L



Blocksize (VN mode)	Docked molecules
32	5733
128	24657
512	59086

Virtual Screening was performed with the protein target 2PK4 (PDB code) with a database of 100,000 molecules run for the limit of 48 hours.

For 5 million molecule screen, assuming 48 hr jobs

512 BG/L blocks, VN mode 50,000 molecule chunks = 100 jobs
128 BG/L blocks, VN mode 20,000 molecule chunks = 250 jobs
i.e about 2 million node hours for a virtual screen

On BG/P 512 block VN mode, 100,000 molecules docked in 20 hours
i.e. we can use 200,000 molecule chunks = 25 jobs!

TODO: Future Plans for Optimization

- Streamline I/O operations to use fewer disk writes
- The HTC mode (High Throughput Computing) available on BG/P provides better scaling for embarrassingly parallel simulations.
- Implement multi-threading using OpenMP to take advantage of BG/P
- Sorting small molecules by # of rotatable bonds leads to better load balancing (Suggestion by IBM researchers)