ATPESC Track 7: Scalable Molecular Visualization and Analysis Tools in VMD

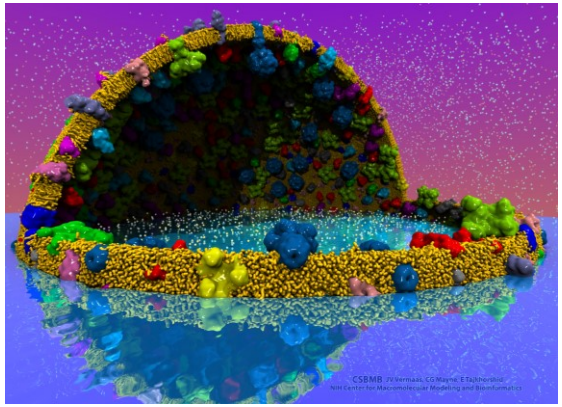
John E. Stone
Theoretical and Computational Biophysics Group
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign

http://www.ks.uiuc.edu/Research/gpu/

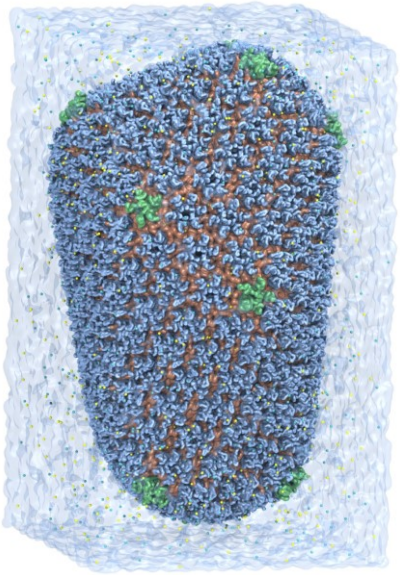
Argonne Training Program on Exascale Computing (ATPESC)
3:30pm-4:30pm, St. Charles Amphitheater, Q Center,
St. Charles, IL, Thursday August 10th, 2017

VMD – “Visual Molecular Dynamics”

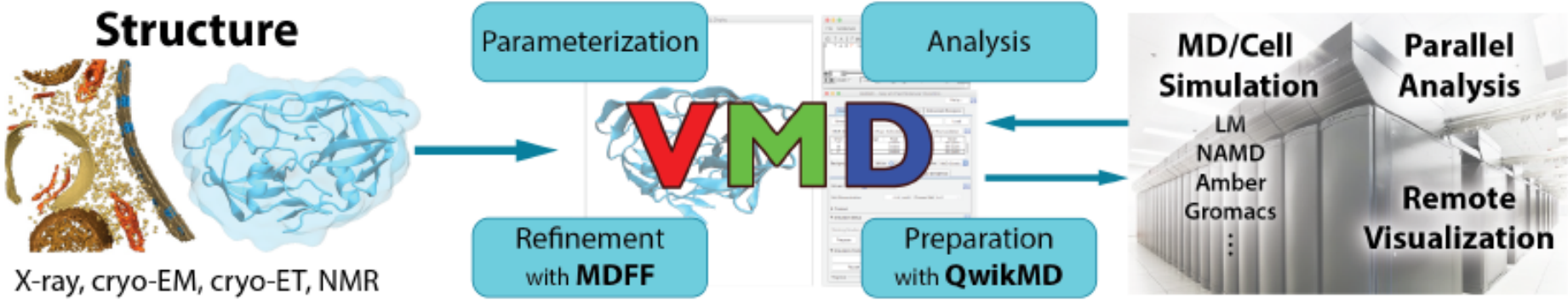
- Visualization and analysis of:
  - Molecular dynamics simulations
  - Lattice cell simulations
  - Quantum chemistry calculations
  - Sequence information
- User extensible scripting and plugins
- http://www.ks.uiuc.edu/Research/vmd/



Cell-Scale Modeling



MD Simulation

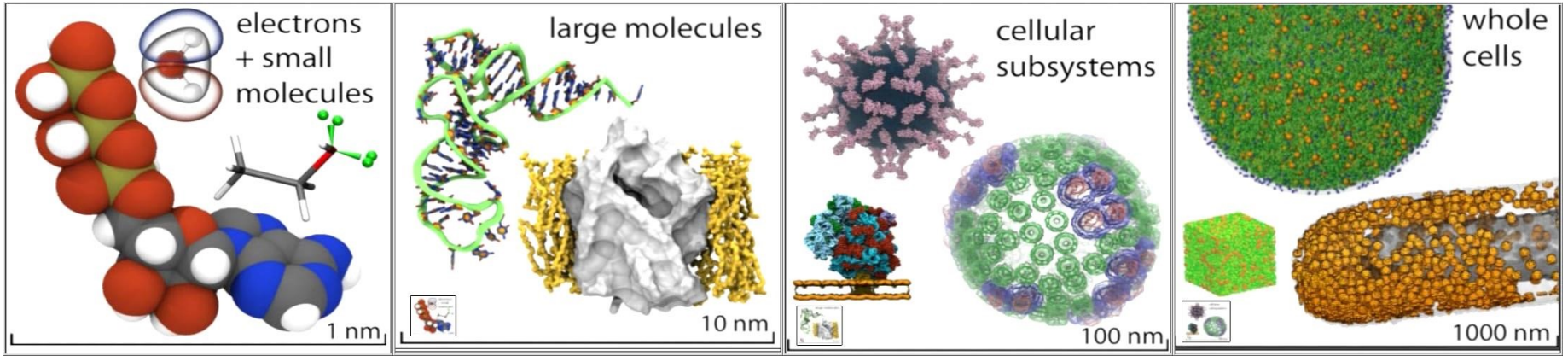


VMD Tutorial Home Page

- http://www.ks.uiuc.edu/Training/Tutorials/
  - Main VMD tutorial
  - QwikMD simulation preparation and analysis plugin
  - VMD images and movies tutorial
  - Structure check
  - VMD quantum chemistry visualization tutorial
  - Visualization and analysis of CPMD data with VMD
  - Parameterizing small molecules using ffTK

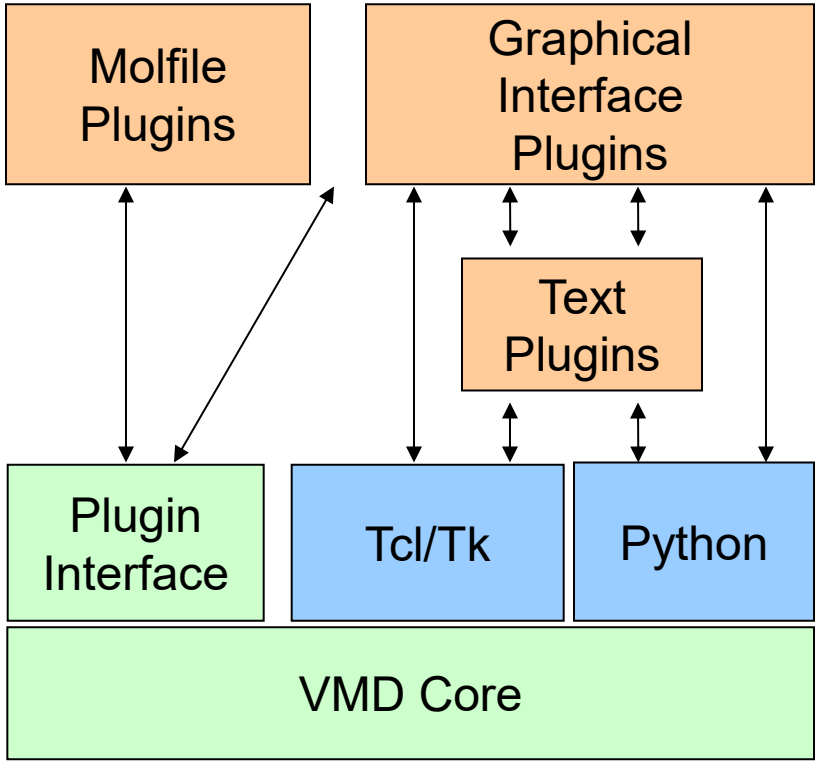
VMD Interoperability Serves Many Communities

- Uniquely interoperable with a broad range of tools:
  - AMBER, CHARMM, CPMD, DL_POLY, GAMESS, GROMACS, HOOMD, LAMMPS, NAMD, and many more …
- Supports key data types, file formats, and databases
- Incorporates tools for simulation preparation, visualization, and analysis



VMD is a Platform for Developing Research Tools
Over 110 VMD Plugins, Half Developed by Users

- VMD user-extensible scripting w/ Tcl/Tk, Python
- User-developed plugins:
  - Alanine Scanning
  - Collective Variable Analyzer
  - Clustering Tool
  - Carbon Nanostructure Builder
  - TorsionPlot
  - RMSD Trajectory Tool
  - Many others…



QwikMD: Guided MD Simulation and Training

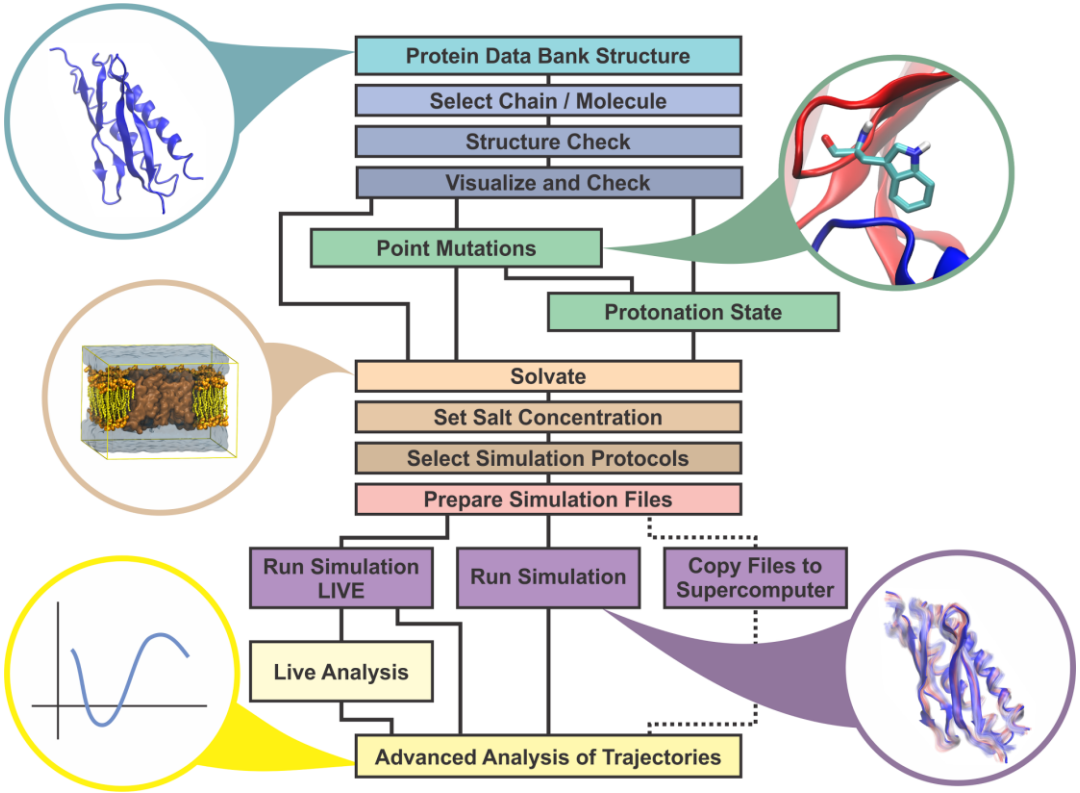
Smooths initial learning curve (non-expert users)

Training: used in 4 Center workshops to-date

Speed up tedious simulation preparation tasks (expert users)

Reproducibility: detailed log of all steps

Interactive preparation, simulation, and analysis



Ribeiro, et al. QwikMD - Integrative Molecular Dynamics Toolkit for Novices and Experts. ***Sci. Rep***. 6, 26536, 2016

Selected VMD Plugins: Center Developed, and User Developed

Analysis

- APBSRun
- CatDCD
- Contact Map
- GofRGUI
- HeatMapper
- ILSTools
- IRSpecGUI
- MultiSeq
- NAMD Energy
- NAMD Plot
- NetworkView
- NMWiz
- ParseFEP
- PBCTools
- PMEpot
- PropKa GUI
- RamaPlot
- RMSD Tool
- RMSD Trajectory Tool
- RMSD Visualizer Tool
- Salt Bridges
- Sequence Viewer
- Symmetry Tool
- Timeline
- TorsionPlot
- VolMap

Modeling

- AutoIonize
- AutoPSF
- Chirality
- Cionize
- Cispeptide
- CGTools
- Dowser
- ffTK
- Inorganic Builder
- MDFF
- Membrane
- Merge Structs
- Molefacture
- Mutator
- Nanotube
- Psfgen
- RESPTool
- RNAView
- Solvate
- SSRestraints
- Topotools

Visualization

- Clipping Plane Tool
- Clone Rep
- DemoMaster
- Dipole Watcher
- Intersurf
- Navigate
- NavFly
- MultiMolAnim
- Color Scale Bar
- Remote
- Palette Tool
- ViewChangeRender
- ViewMaster
- Virtual DNA Viewer
- VMD Movie Maker

Simulation

- AlaScan
- AutoIMD
- IMDMenu
- NAMD GUI
- NAMD Server
- QMTool

Collaboration

- Remote Control

Data Import and Plotting

- Data Import
- Multiplot
- PDBTool
- MultiText

Externally Hosted Plugins and Extensions

- Check sidechains
- MultiMSMS
- Interactive Essential Dynamics
- Mead Ionize
- Clustering Tool
- iTrajComp
- Swap RMSD
- Intervor
- SurfVol
- vmdICE

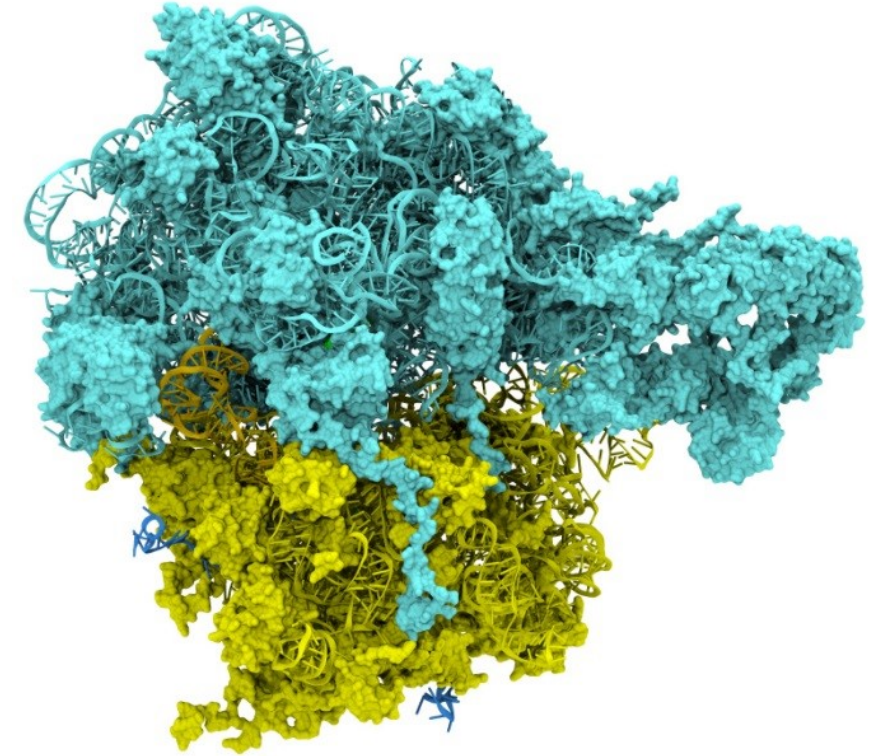
75 MolFile I/O Plugins:
structure, trajectory, sequence, and density map

http://www.ks.uiuc.edu/Research/vmd/plugins/

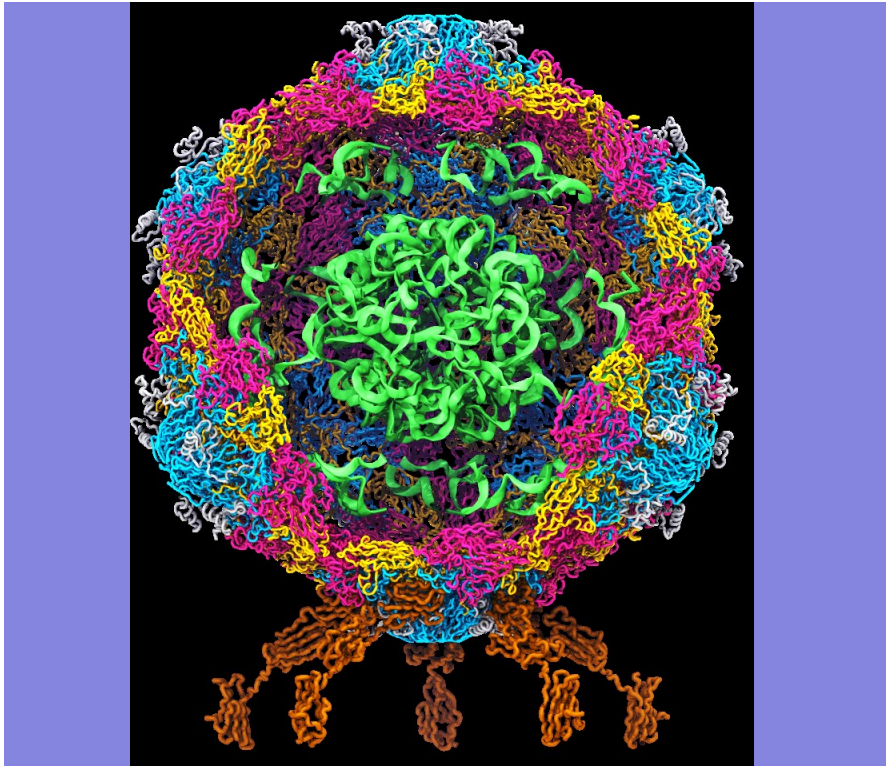
Goal: A Computational Microscope

Study the molecular machines in living cells

Ribosome: target for antibiotics



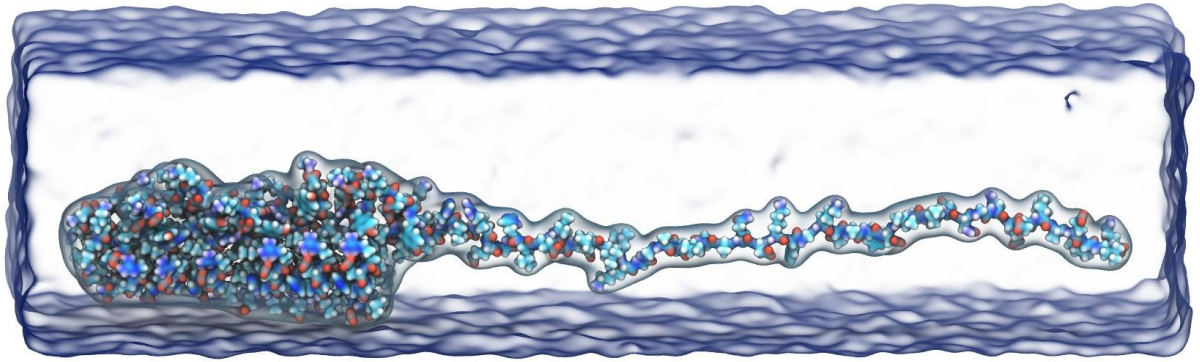
Poliovirus



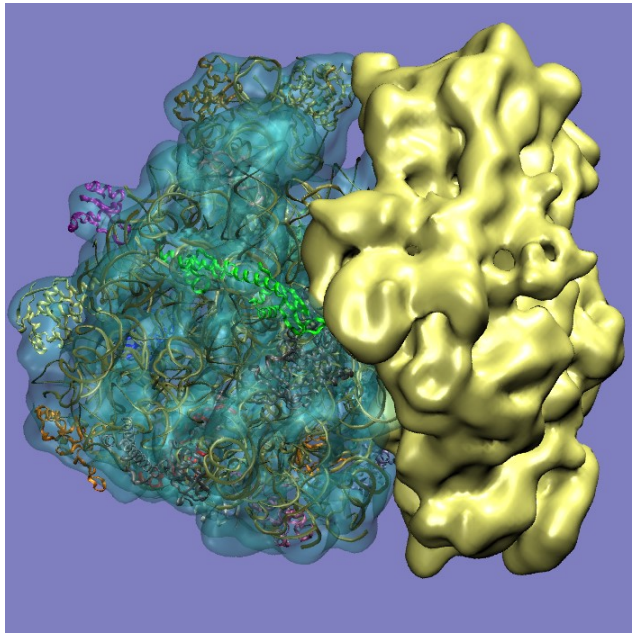
Structure Visualization

Molecular representations provide different levels of abstraction, atomic detail vs. higher level organizational information

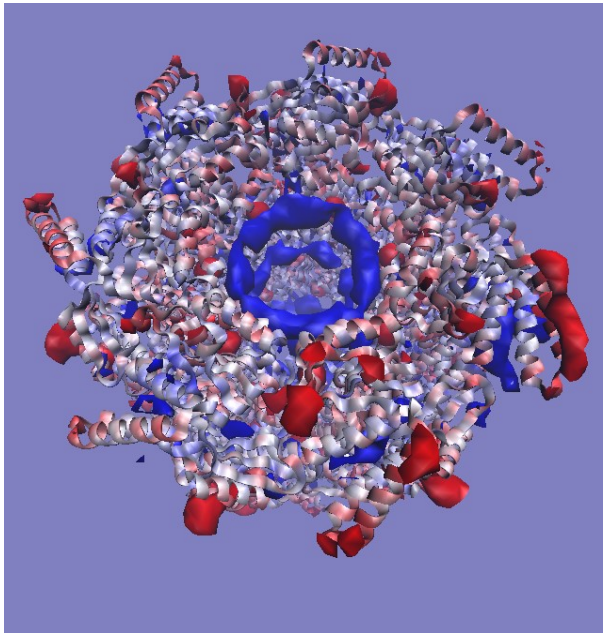
- Atoms, VdW spheres, bonds, ball-stick, …
- Molecular orbitals (quantum chemistry)
- Molecular surfaces
- Coarse-grained “beads”
- Ribbons, secondary structure, “cartoon” reps, RNA/DNA



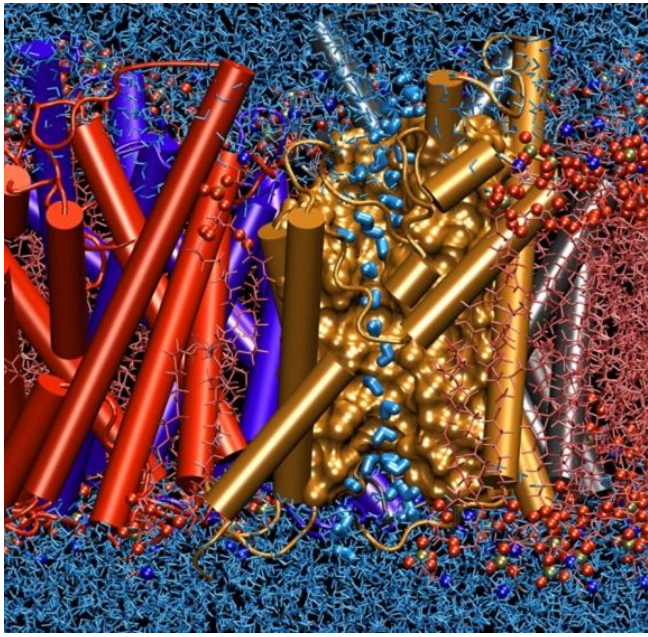
VMD Representation Examples



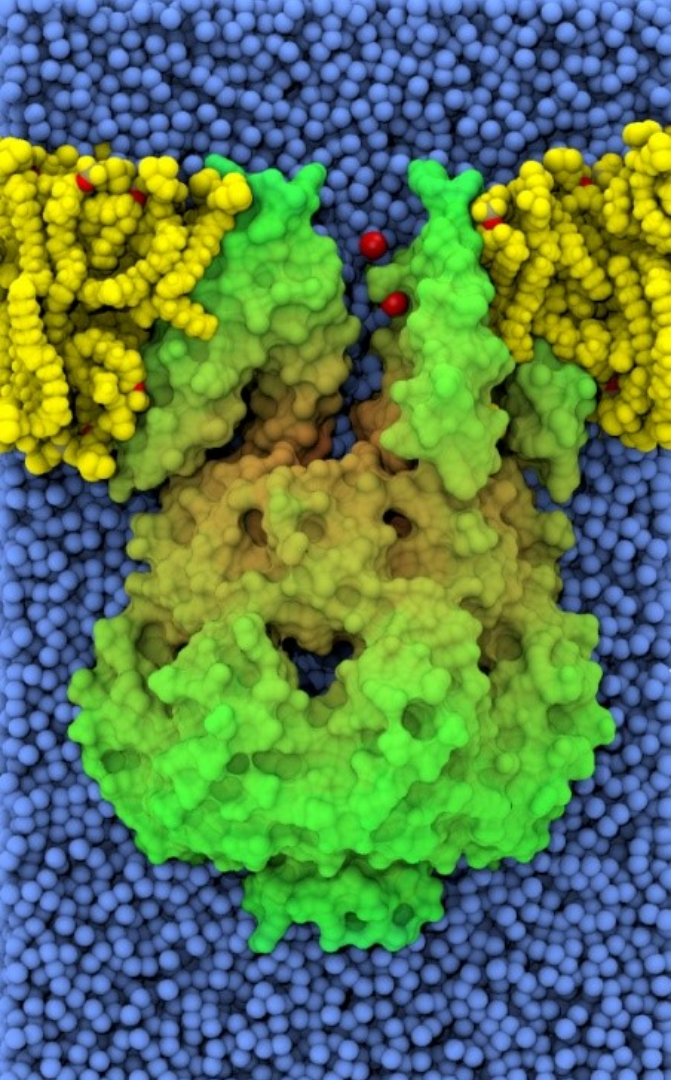
Ribosome, J. Frank

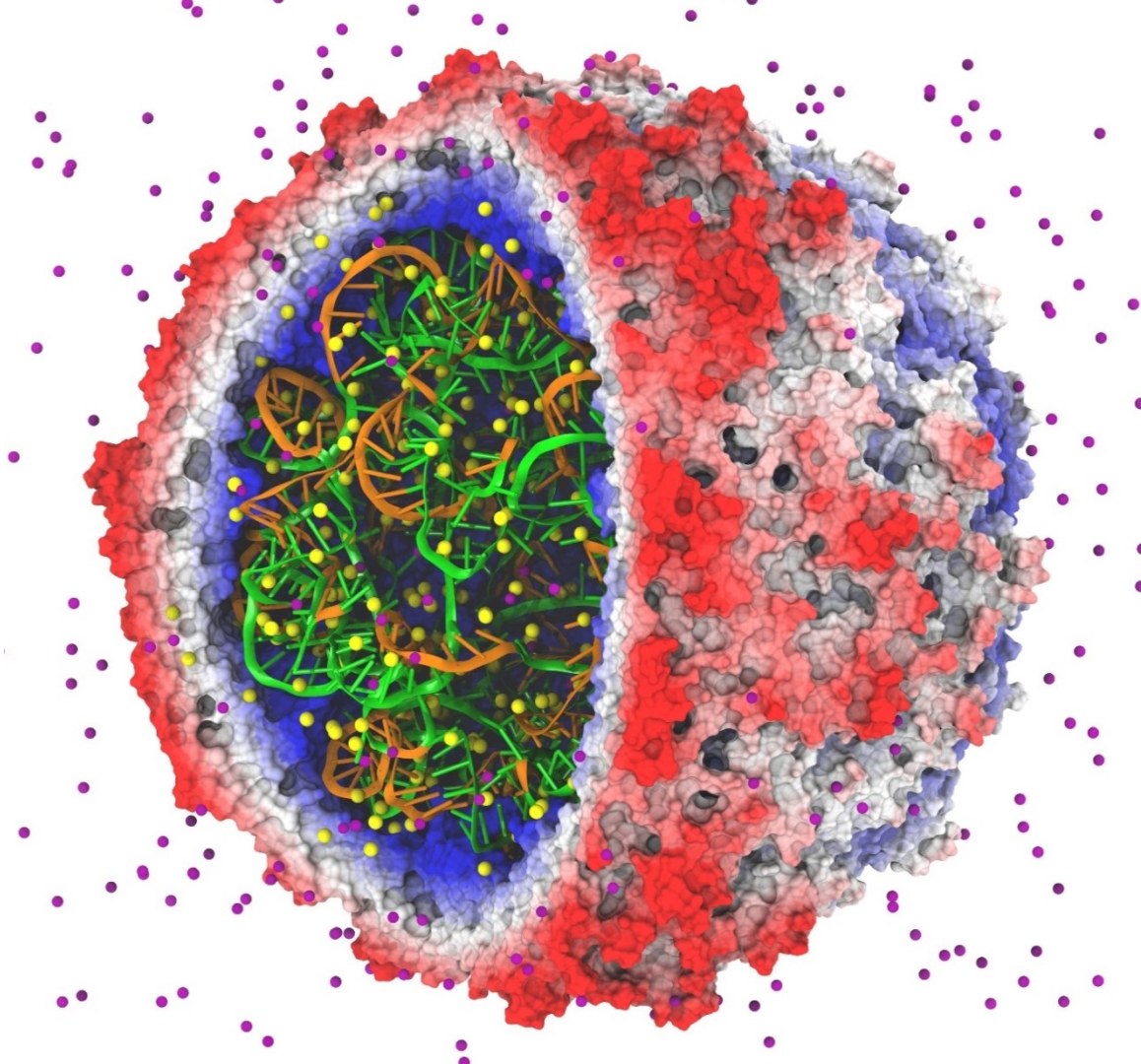


GroEL /w Situs



Aquaporin channel





Selection, Filtering

- Most viz tools allow interactive visual picking, menu-driven selections of structure components to display or operate on
- VMD also extensively uses a text-based selection language (think google):

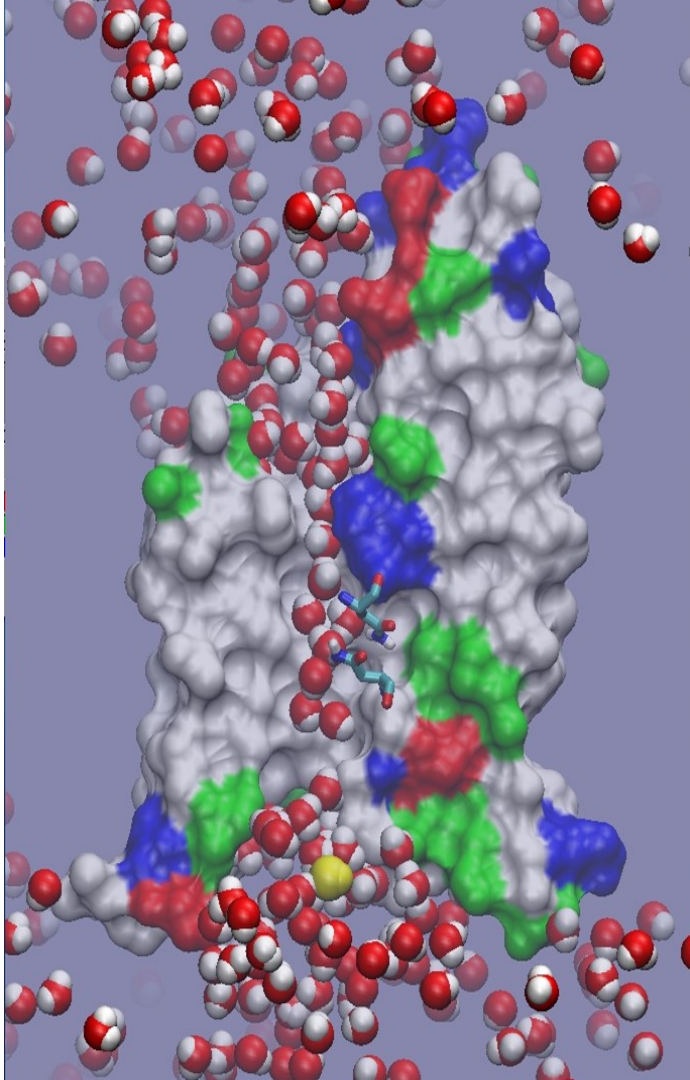
  water within 10 of protein and z > 0

  nucleic or protein or ions

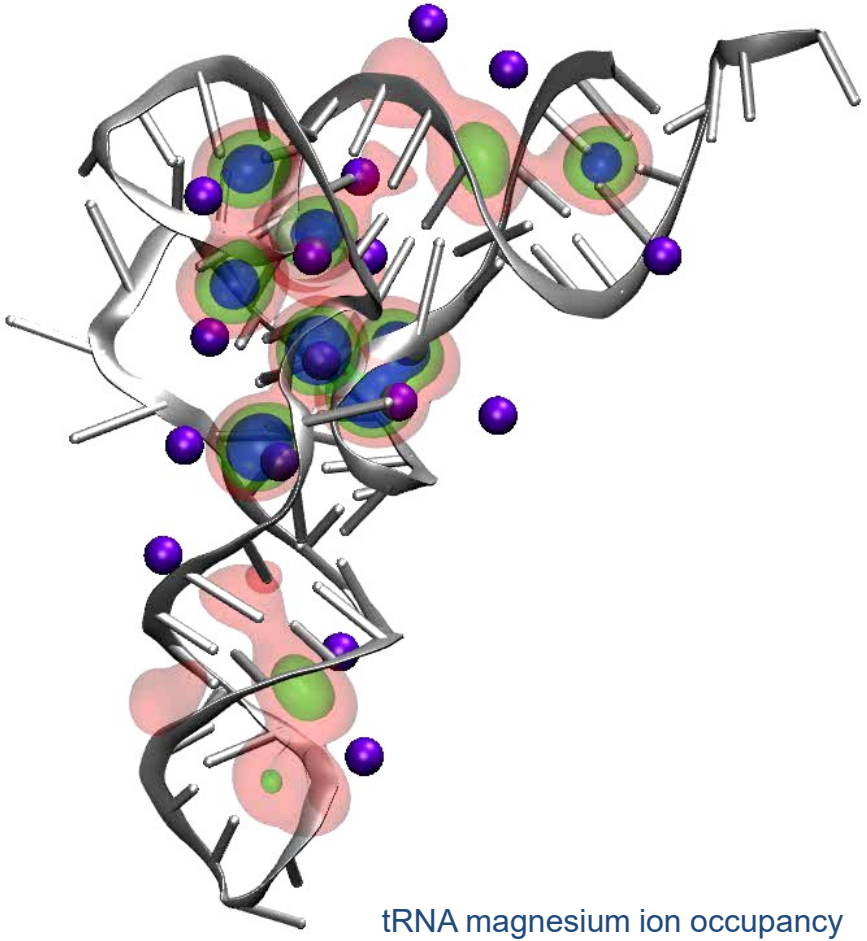
  segname BR

  name “C.*”

  – Allows selection on user-defined data fields
  – **Promotes synergy between interactive and scripting interfaces, visualization and quantitative analysis tasks**
  – Works well with huge time-varying structures



Computing Molecular Properties



tRNA magnesium ion occupancy

Compute properties, e.g., density, distance, occupancy, electrostatic potential over thousands of MD simulation trajectory frames

Example: display binding sites for diffusively bound ions as probability density isosurfaces

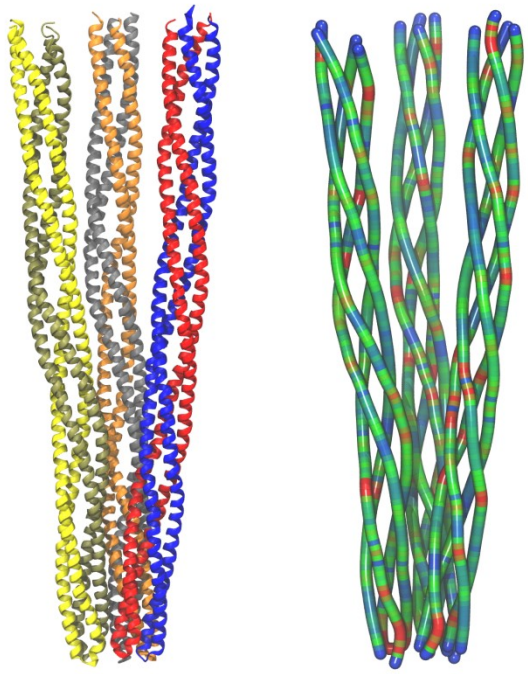
Run calculations in parallel on clusters, supercomputers with thousands of nodes, **Blue Waters, Titan, etc …**

Reduce 5 years of runtime to 20 minutes on 2048 GPU-accelerated nodes!

Parallel I/O: 275 GB/sec on 8192 nodes

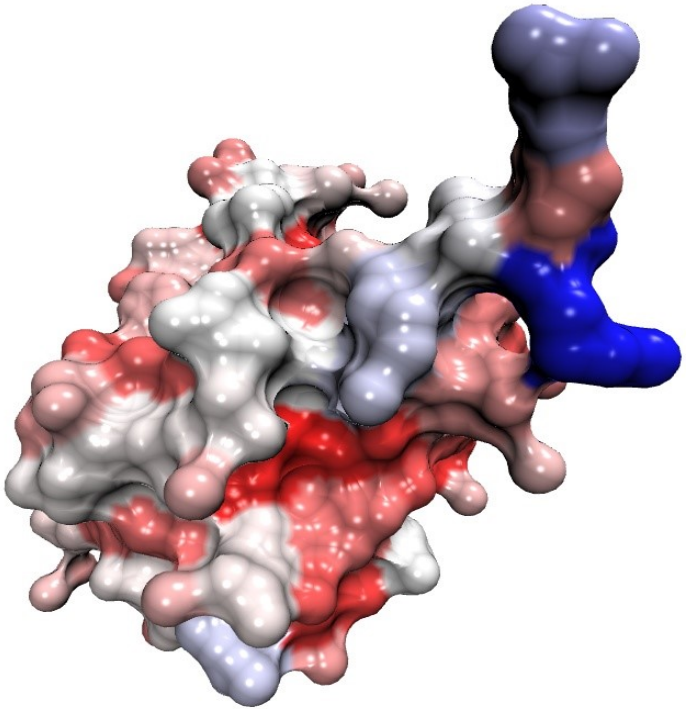
Computed Properties

- Smoothing of thermal noise
- Secondary structure
- Hydrogen bonds, salt bridges
- Forces, energies, stress, strain
- Time averaging of electrostatic fields, occupancy maps
- Quality-of-fit cross correlation with cryo-EM density maps
- Normal modes, principal component analysis, essential dynamics
- Cluster simulation trajectory timesteps by structural similarity

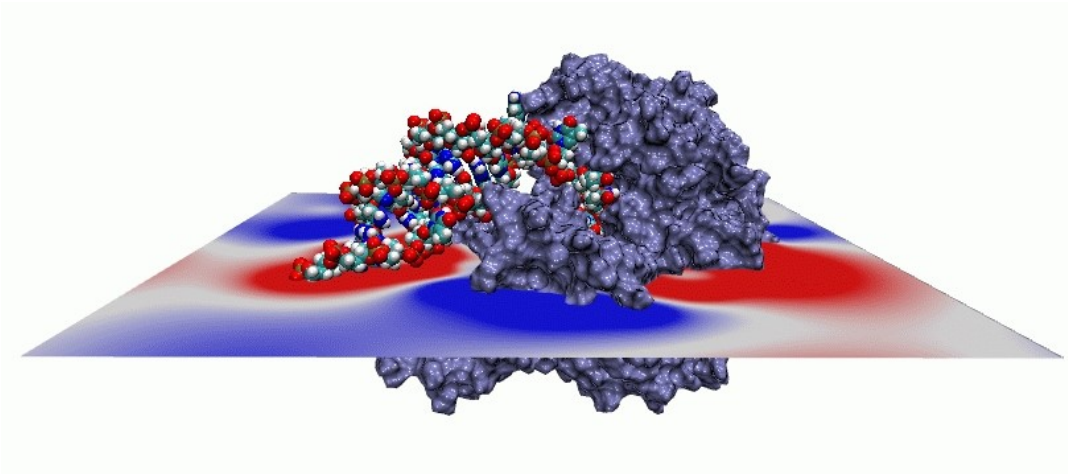


Chemoreceptor trimer-of-dimers analysis with Bendix plugin in VMD

Display of Computed Properties on Structures

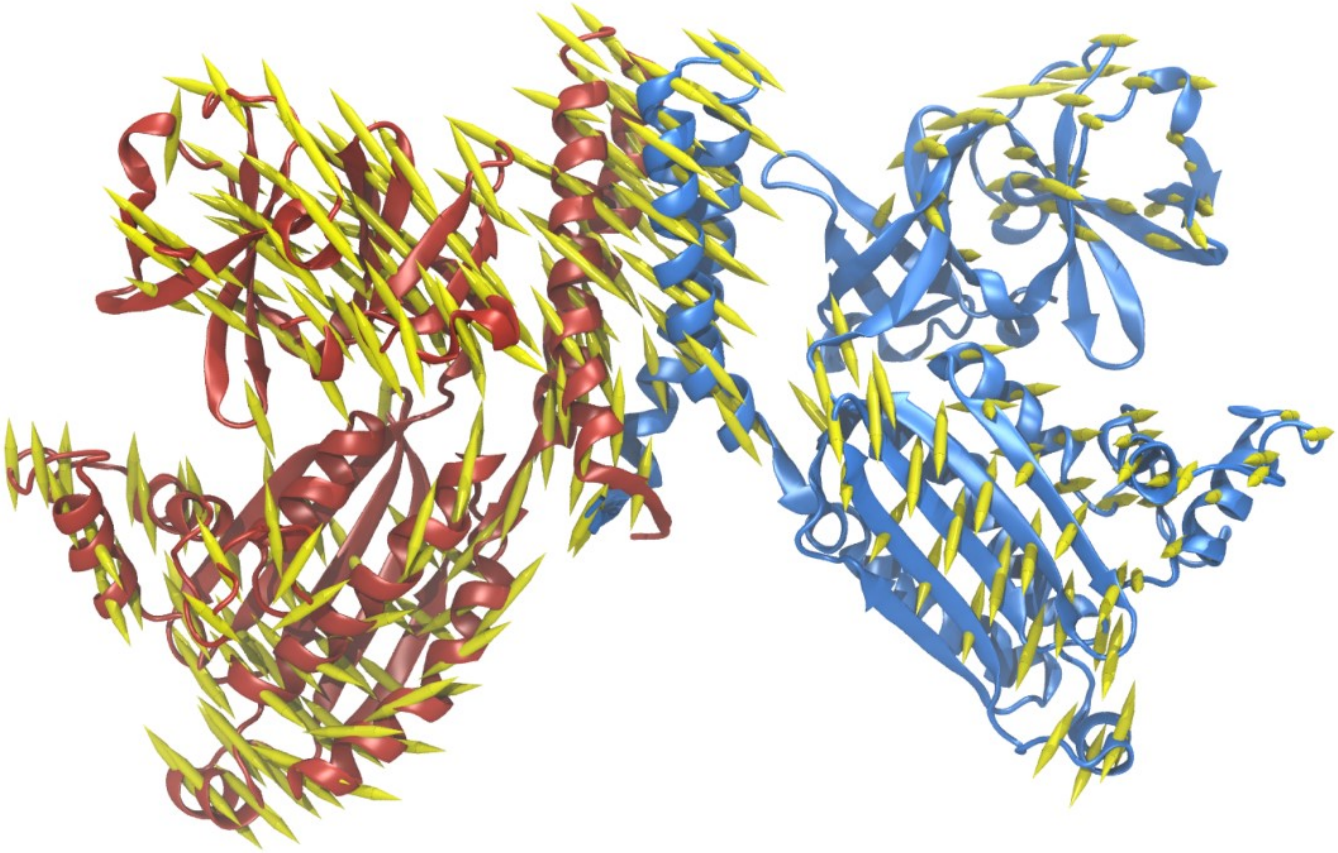


Per-residue solvent-accessible surface area of Ubiquitin

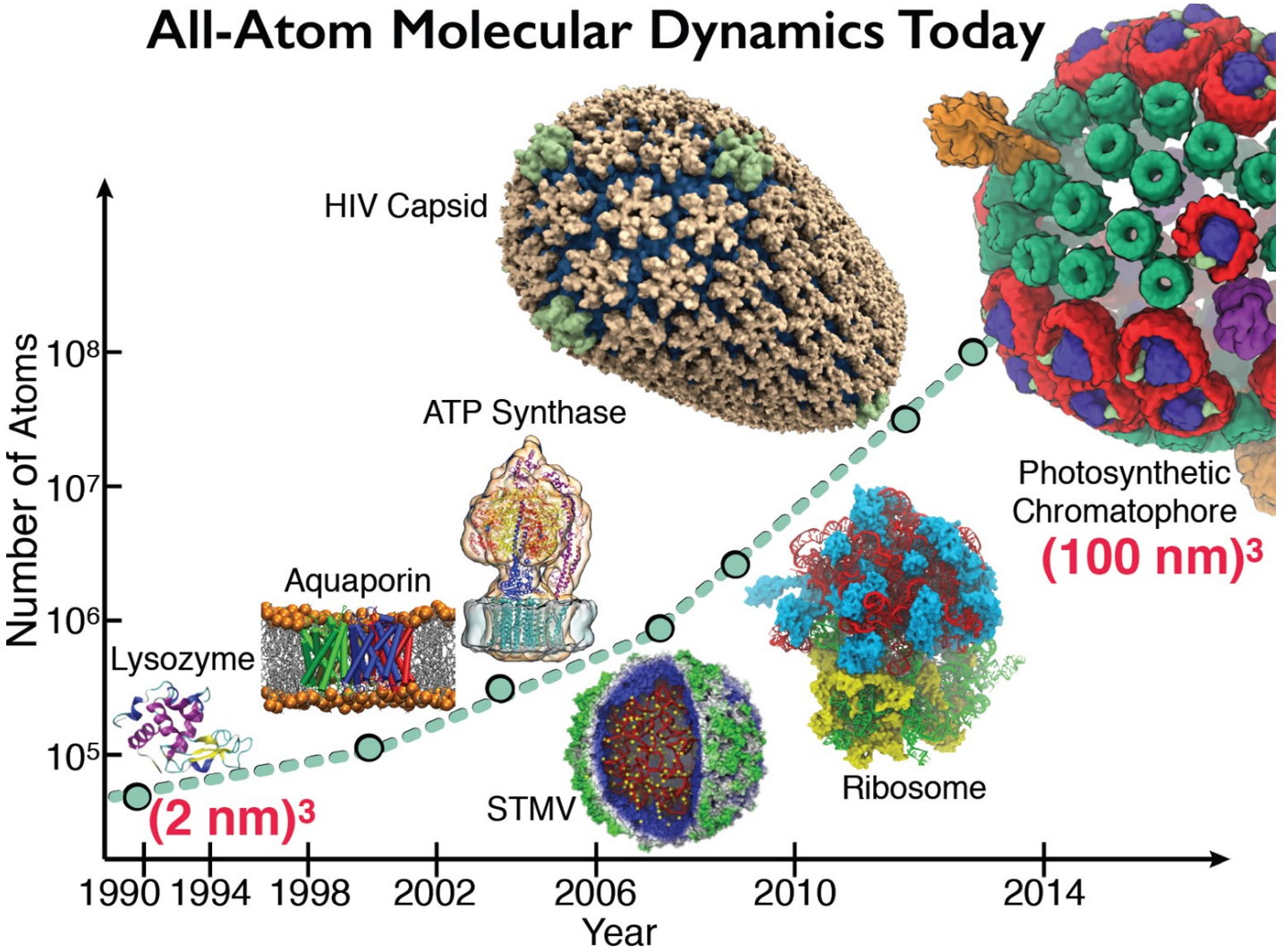


PME electrostatic potential contour for a helicase on a volumetric slice plane

CheA kinase PCA: first principal component porcupine plot



All-Atom Molecular Dynamics Today



VMD Petascale Visualization and Analysis

- Analyze/visualize large trajectories too large to transfer off-site:
  - User-defined parallel analysis operations, data types
  - Parallel rendering, movie making
- Supports GPU-accelerated Cray XK7 nodes for both visualization and analysis:
  - **GPU accelerated trajectory analysis w/ CUDA**
  - **OpenGL and GPU ray tracing for visualization and movie rendering**
- Parallel I/O rates up to **275 GB/sec** on 8192 Cray XE6 nodes – can read in **231 TB in 15 minutes!**

Parallel VMD currently available on:

ORNL Titan, NCSA Blue Waters, Indiana Big Red II, CSCS Piz Daint, and similar systems

NCSA Blue Waters Hybrid Cray XE6 / XK7
22,640 XE6 dual-Opteron CPU nodes
4,224 XK7 nodes w/ Telsa K20X GPUs

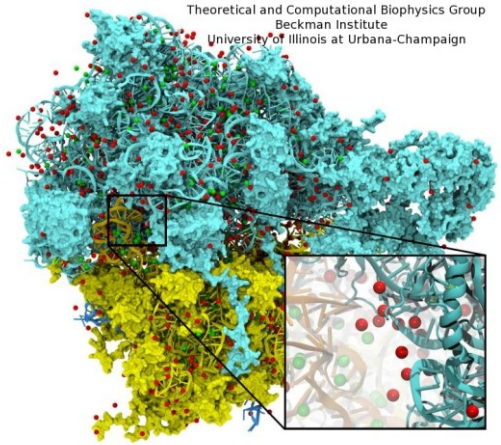
10 Years of GPU Computing in VMD

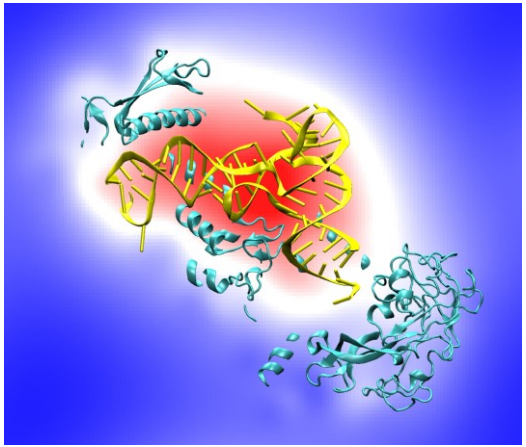
- Has stood the test of time
- Modeling, Visualization, Rendering, and Analysis

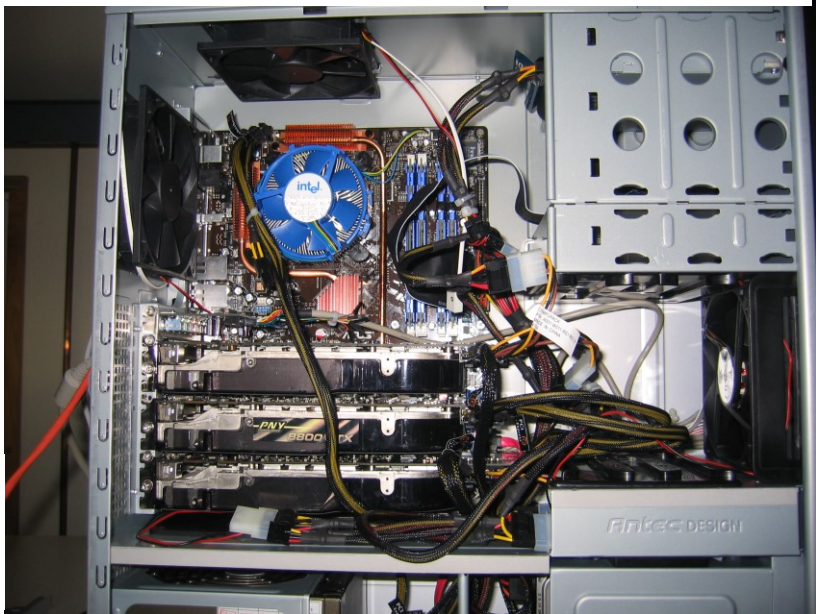
Blast from the past:

CUDA starting with version 0.7 !!!

Quad core Intel QX6700, three NVIDIA GeForce 8800GTX GPUs, RHEL4 Linux



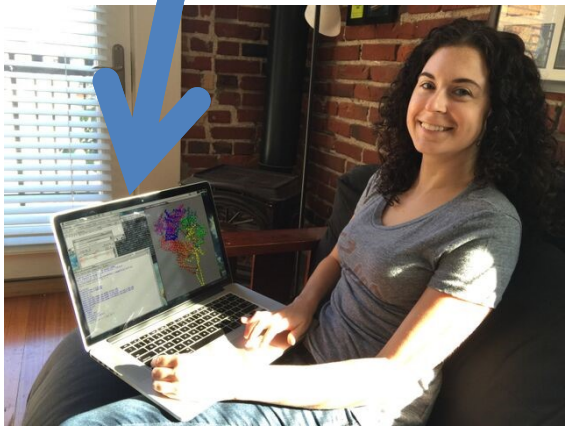




Accelerating molecular modeling applications with graphics processors. J. Stone, J. Phillips, P. Freddolino, D. Hardy, L. Trabuco, K. Schulten. *J. Comp. Chem.*, 28:2618-2640, 2007.

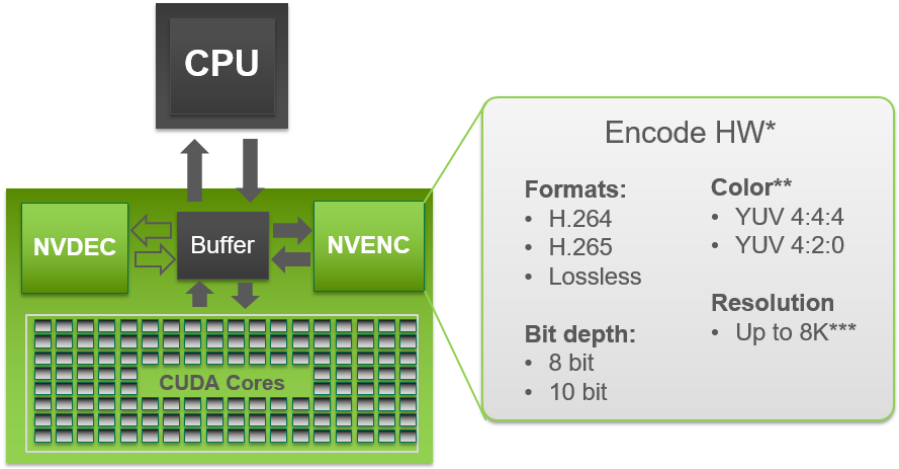
Interactive Remote Visualization and Analysis

- Enabled by hardware H.264/H.265 video encode/decode
- Enable visualization and analyses not possible with conventional workstations
- Access data located anywhere in the world
  - Same VMD session available to any device
- Linux prototype in-development using NVIDIA Video Codec SDK, easy-to-use ***NvPipe*** wrapper library



NVIDIA Video CODEC SDK and NvPipe

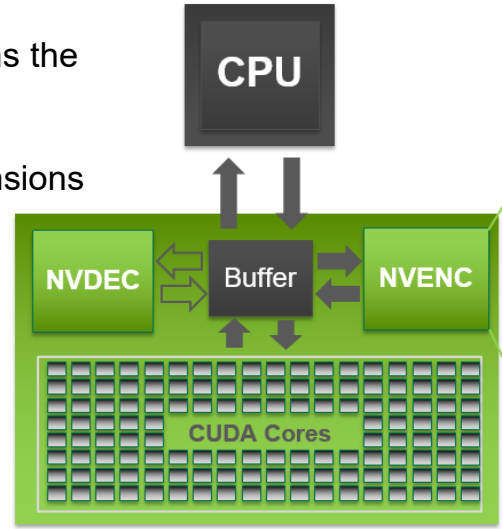
- GPUs (Kepler-on) include NVENC and NVDEC video codec acceleration hardware
- Independent of GPU compute hardware
- Hardware-accelerated codecs can ***overlap*** with interactive **rendering**, and **computation**
- ***NvPipe*** provides an easy to use API for interactive video streaming, abstracting many low level codec details, ideal for basic remote visualization implementations: **https://github.com/NVIDIA/NvPipe**



NvPipe

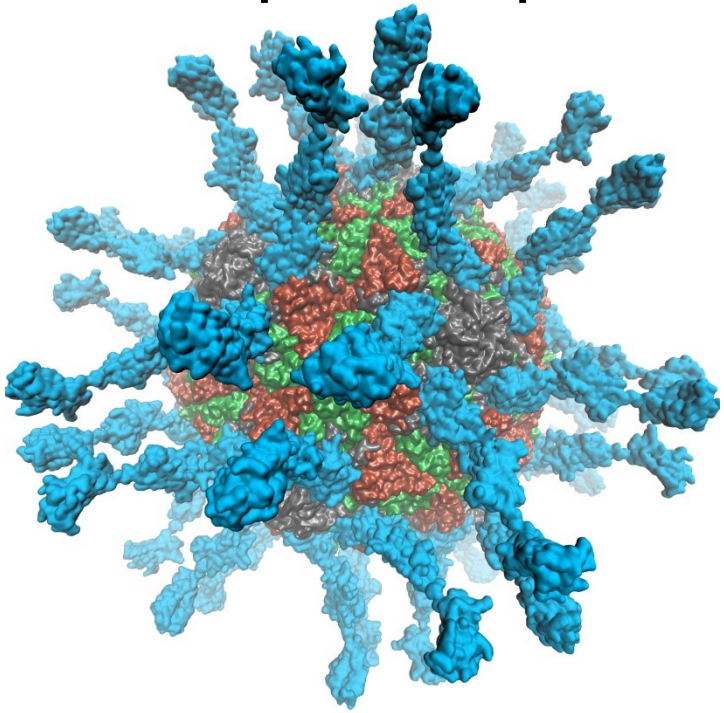
https://github.com/NVIDIA/NvPipe

- Simplified API for producing a basic encoder/decoder system.
- **Roughly 100 lines of code for basic encode/decode “Hello World” loops with minimal error handling logic**
- Encode/decode ends up being **simpler than your networking code** ☺
- Encode loop structure:
  - User selects encoder type***, e.g. NVPIPE_H264_NV***, and target encoder bitrate parameter
  - User provides uncompressed **RGB** or **RGBA** image buffer, image dimensions, and size of the output memory buffer
  - NvPipe compresses the frame using the NVENC hardware encoder, and returns the number of bytes of output written to the output buffer
- Symmetric decode loop structure:
  - Provide decoder with compressed buffer, buffer size in bytes, and image dimensions as input
  - Decoder produces uncompressed output image
- Optionally supports FFMPEG back-ends (but I haven’t tried those yet)



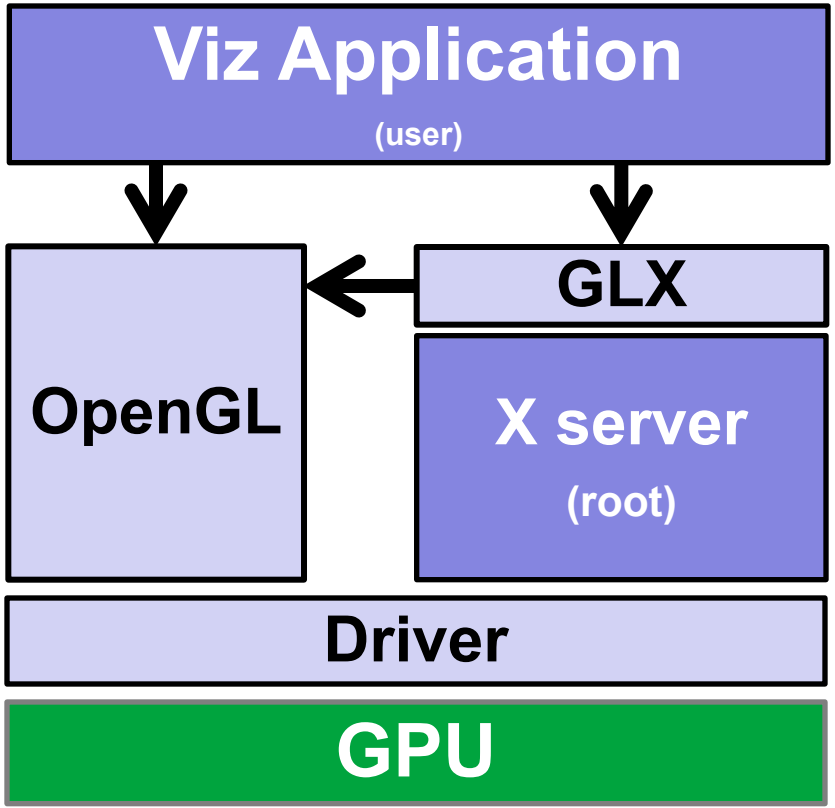
VMD 1.9.3 supports EGL for in-situ and parallel rendering on clouds, clusters, and supercomputers

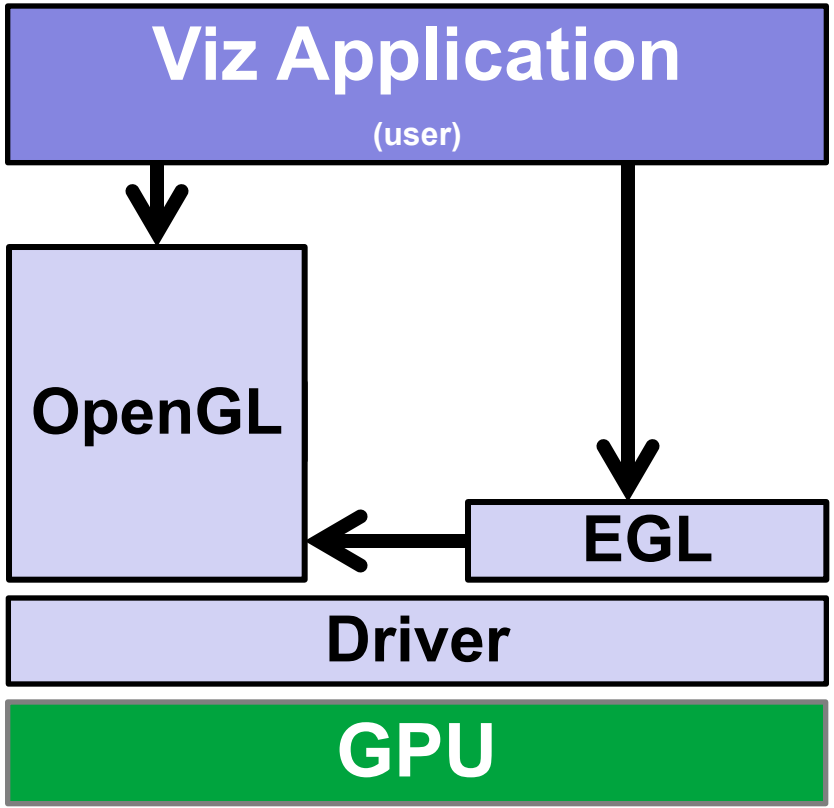
- Eliminate dependency on windowing systems
- Simplified deployment of parallel VMD builds supporting off-screen rendering
- Maintains 100% of VMD OpenGL shaders and rendering features
- Support high-quality vendor-supported commercial OpenGL implementations in HPC systems that were previously limited to Mesa



Poliovirus

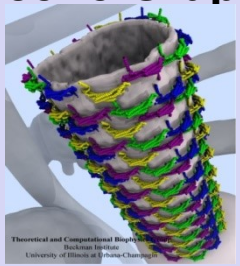
OpenGL: GLX vs. EGL





Molecular Structure Data and Global VMD State

Scene Graph



Graphical Representations

- DrawMolecule
- Non-Molecular Geometry

User Interface Subsystem

- Tcl/Python Scripting
- Mouse + Windows
- 6DoF Input “Tools”

Display Subsystem

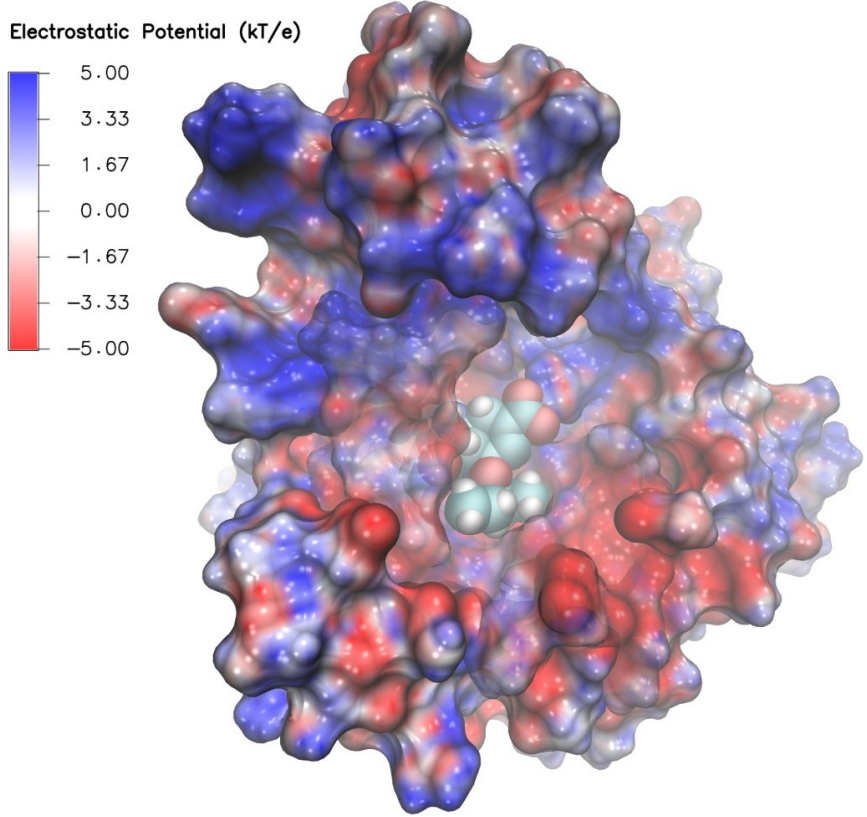
- VMDDisplayList
- DisplayDevice
- OpenGLRenderer

GLX+X11+Drv

- OpenGL Pbuffer/FBO
- Windowed OpenGL

EGL+Drv

- OpenGL Pbuffer/FBO

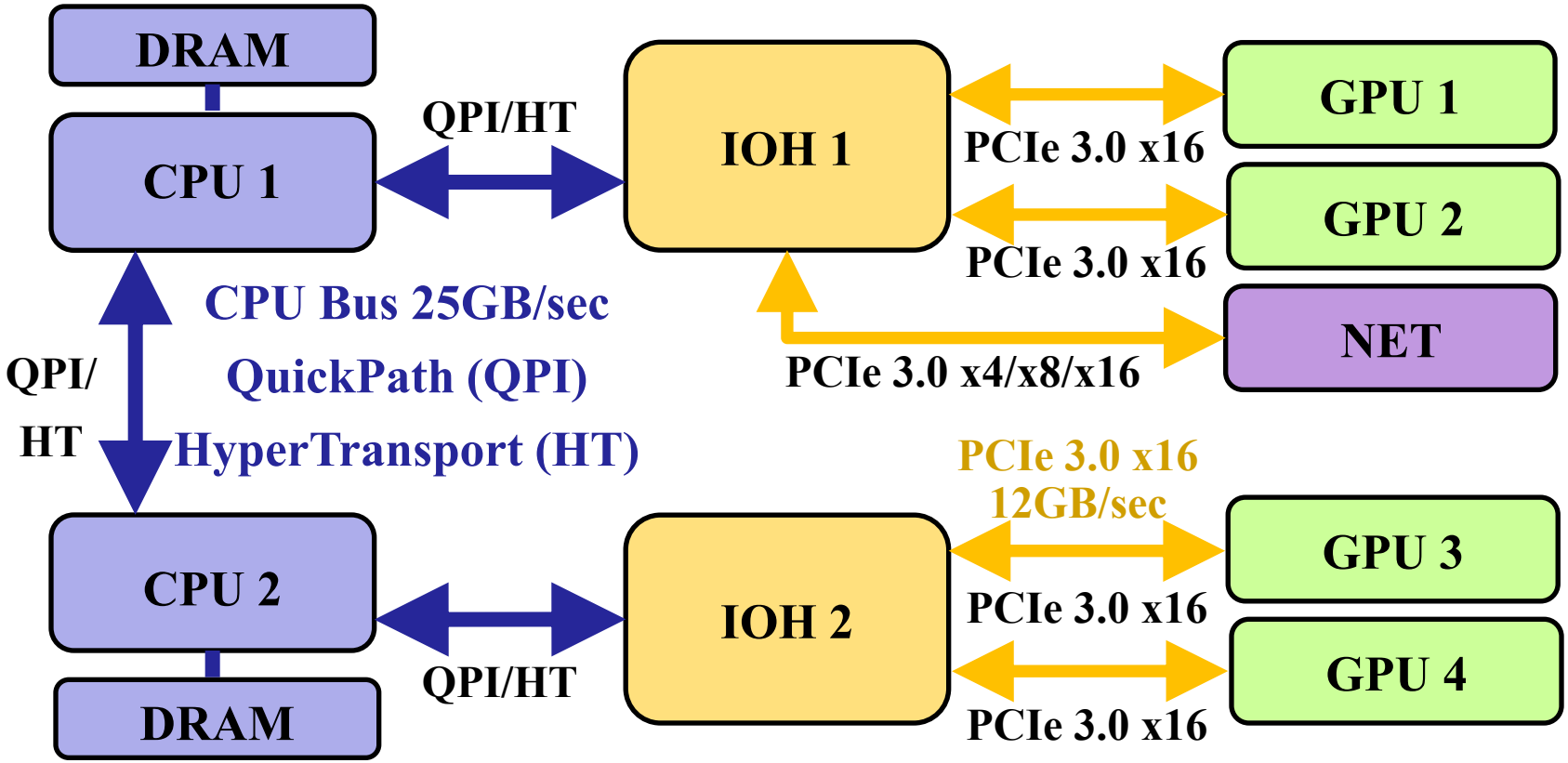


Swine Flu A/H1N1 neuraminidase bound to Tamiflu: VMD EGL rendering demonstrating full support for all VMD shaders and OpenGL features, multisample antialiasing, ray cast spheres, 3-D texture mapping, ...

Benefits of EGL Platform Interfaces

- Enumerate and select among available platforms, potentially supporting multiple vendors in the same host/node
  - Allows specific target implementation to be bound, e.g. GPU, CPU-integrated GPU, software rasterizer
- **EGL interfaces make it EASY to bind a GPU** to a thread with optimal CPU affinity **with respect to NUMA topology, NVLink GPU topology**
  - High-perf. multi-GPU image compositing, video streaming
  - EGL plays nicely with MPI, CUDA/OpenCL, OptiX, NVENC, etc
  - NVIDIA EGL supports multiple GPU indexing schemes, e.g. **PCIe ordering**
  - **Exploit NVLink interconnect topology on IBM OpenPOWER platforms, e.g. "Minsky", upcoming DOE/ORNL "Summit" system**

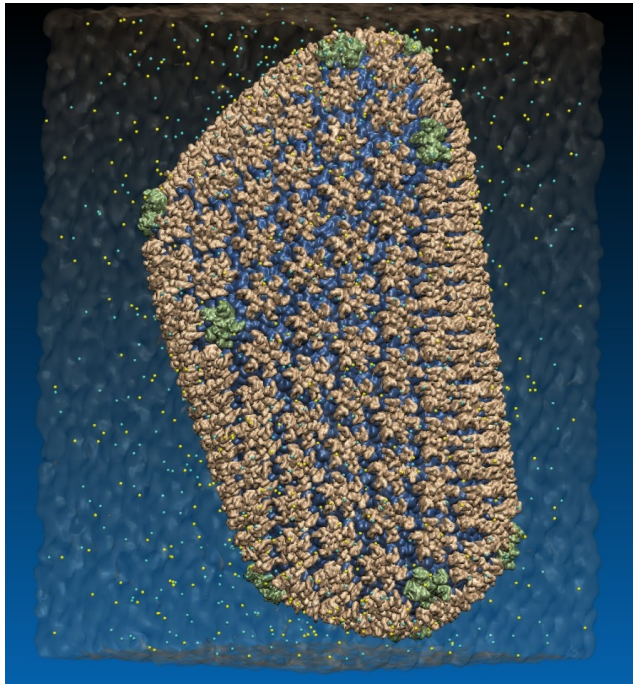
Example Node NUMA Topology



VMD EGL Performance on Amazon EC2 Cloud

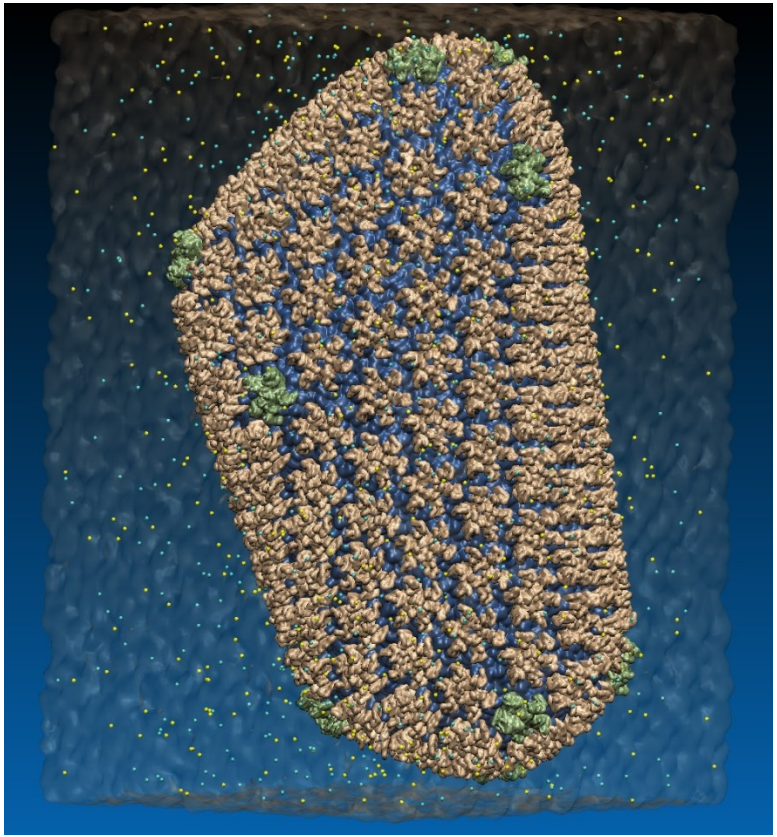
MPI Ranks	EC2 "G2.8xlarge" GPU Instances	HIV-1 movie rendering time (sec), (I/O %) 3840x2160 resolution
1	1	626s (10% I/O)
2	1	347s (19% I/O)
4	1	221s (31% I/O)
8	2	141s (46% I/O)
16	4	107s (64% I/O)
32	8	90s (76% I/O)

Performance at 32 nodes reaches ~48 frames per second

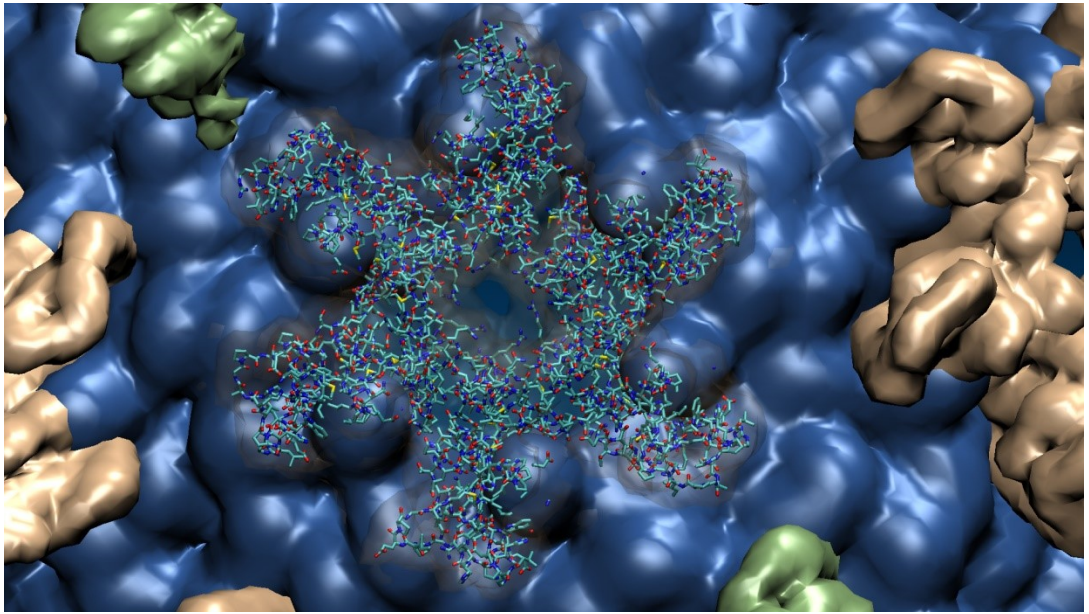


64M atom HIV-1 capsid simulation rendered via EGL

High Performance Molecular Visualization: In-Situ and Parallel Rendering with EGL. J. E. Stone, P. Messmer, R. Sisneros, and K. Schulten. High Performance Data Analysis and Visualization Workshop, IEEE IPDPSW, pp. 1014-1023, 2016.



64M atom HIV-1 capsid simulation rendered via EGL



Close-up view of HIV-1 hexamer rendered via EGL

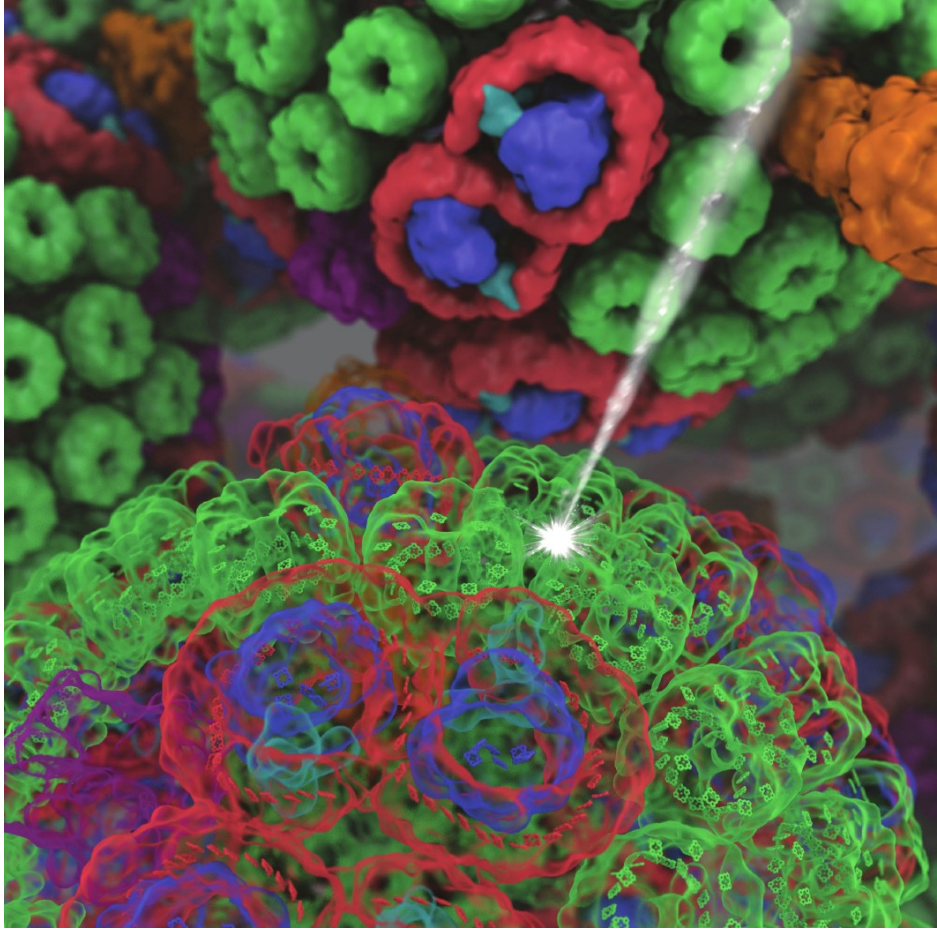
EGL Is Supported Now!

- Cloud+Workstations with most recent NVIDIA drivers
- VMD on HPC systems w/ latest Tesla P100 GPUs:
  - Cray XC50, CSCS Piz Daint, driver 375.39
  - IBM OpenPOWER, drivers 375.66 and later support both GLX and EGL

High Fidelity Ray Tracing

- **Interactive RT** on laptops, desk, cloud
- Large-scale parallel rendering:
  in situ or post hoc visualization tasks
- AO, DoF, instancing, ….
- Stereoscopic panorama and full-dome projections
- Omnidirectional VR: YouTube, HMDs

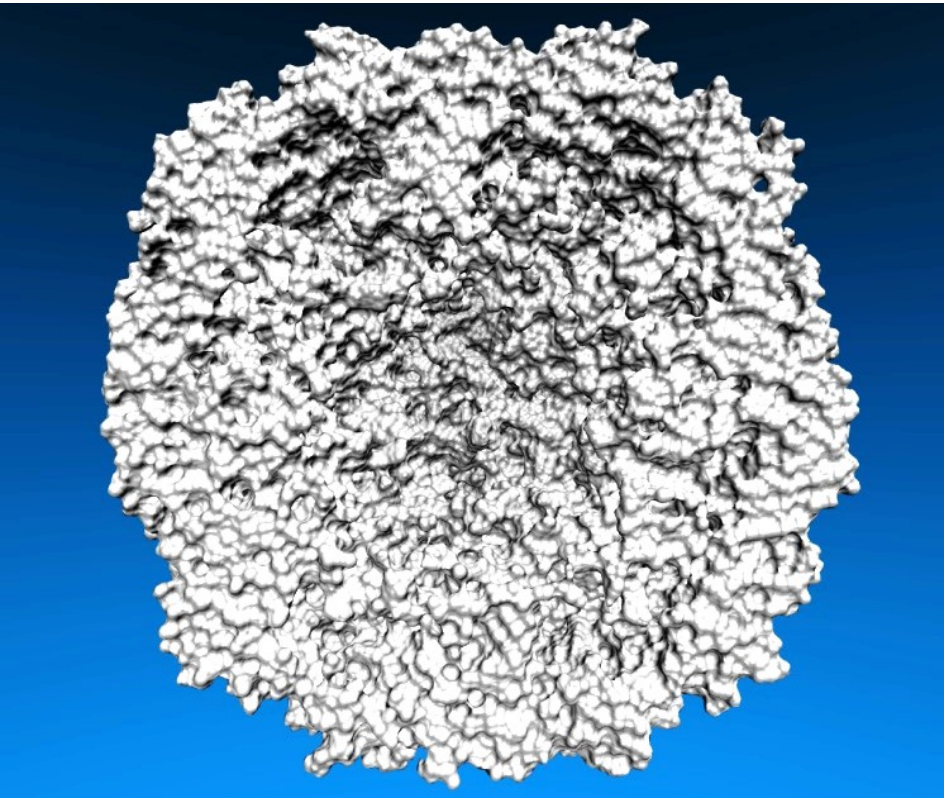
- Built-in ray tracing engines:
  - **Tachyon:** cross-platform RT
  - **NVIDIA OptiX:** GPU-accelerated and remote RT on VCA clusters
  - **Intel OSPRay:** CPU x86/Phi-optimized parallel rendering w/ MPI



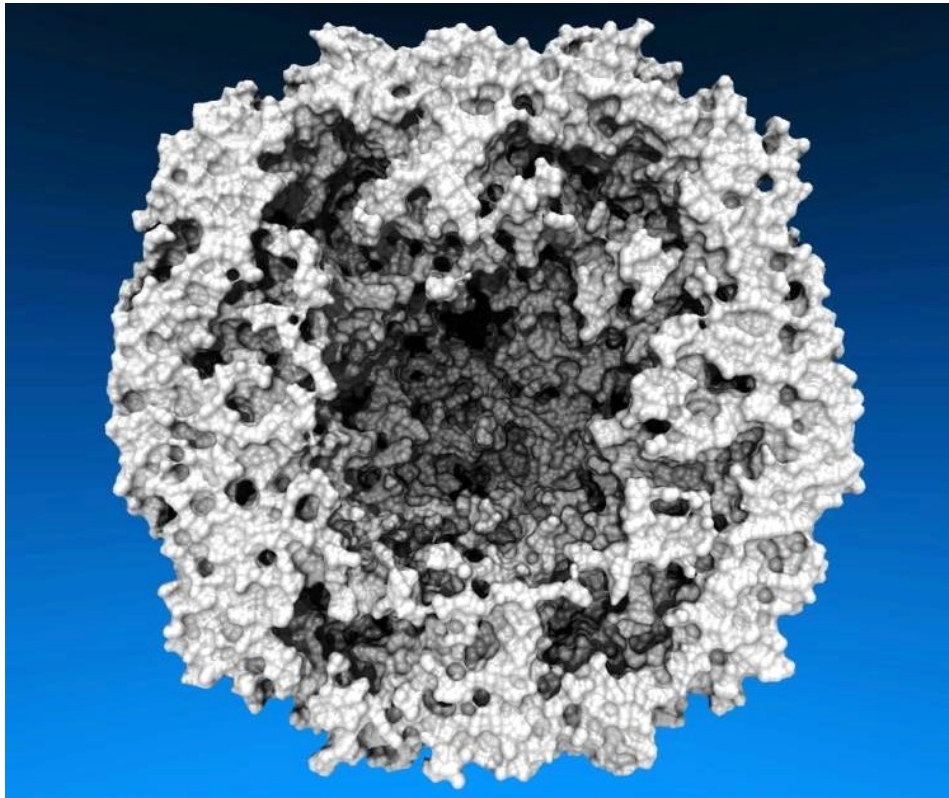
VMD/OptiX all-atom Chromatophore

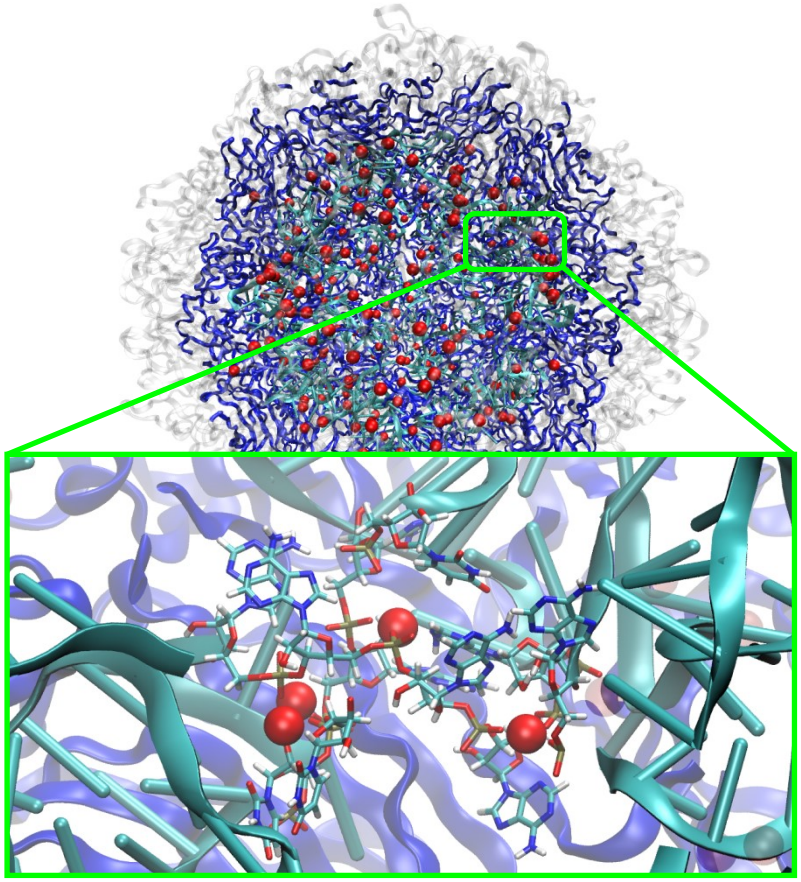
Lighting Comparison, STMV Capsid

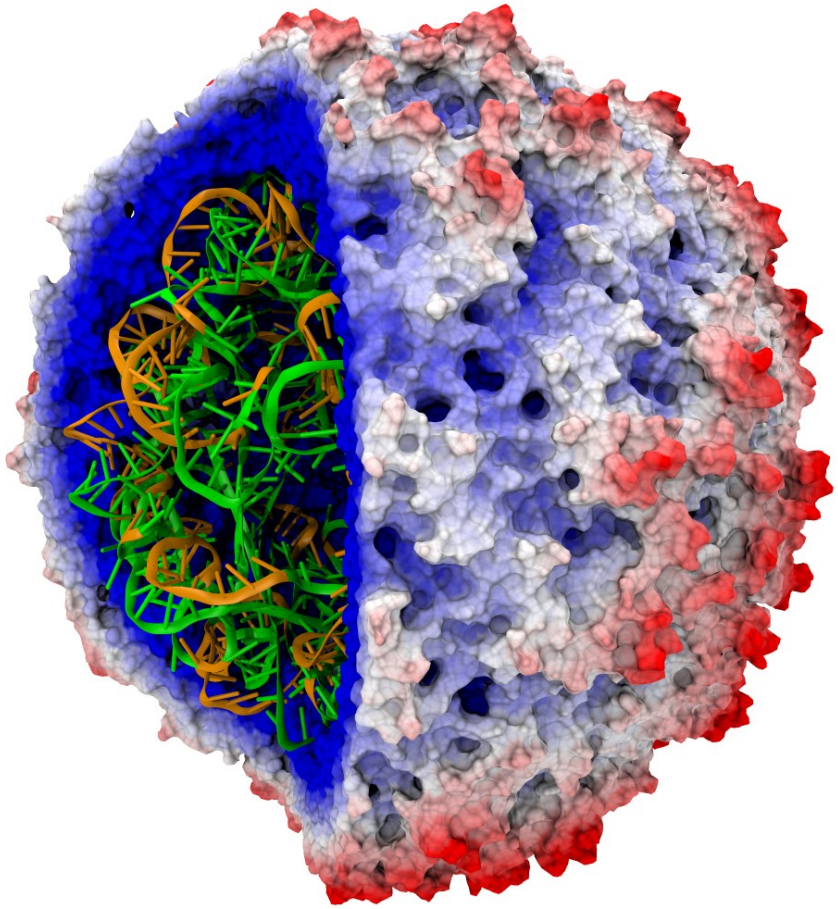
Two lights, no shadows



Ambient occlusion + two lights, 144 AO rays/hit



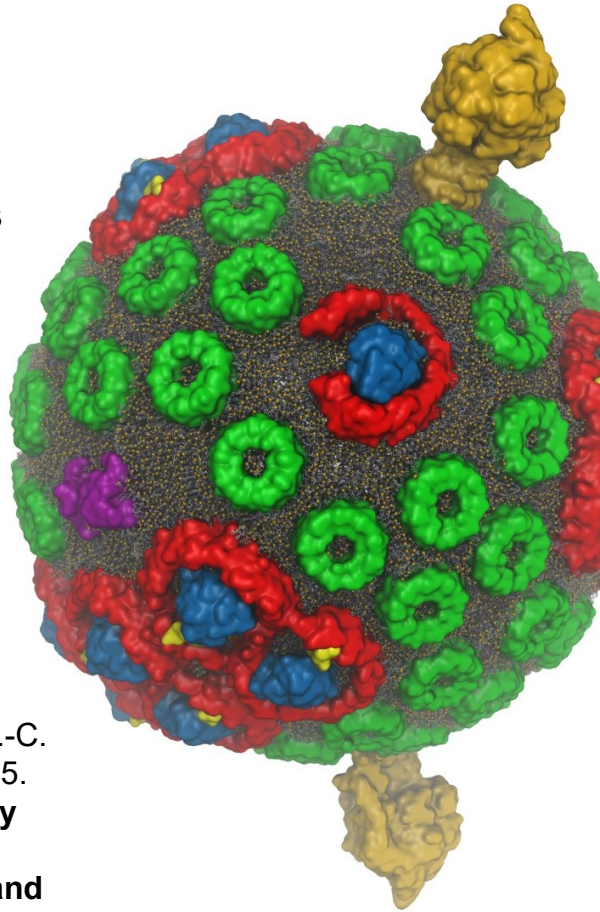




Satellite Tobacco Mosaic Virus

VMD w/ OptiX 4.1

- Interactive RT on laptops, desktops, and cloud
- Large-scale parallel rendering: in situ or post hoc visualization tasks
- Remote RT on NVIDIA VCA clusters
- Stereoscopic panoramic and full-dome projections
- Omnidirectional VR for YouTube, VR HMDs
- **Top-end Pascal Tesla GPUs roughly 2x faster than Kepler**
- **GPU memory sharing via NVLink on Quadro GP100, Tesla P100**



VMD/OptiX GPU Ray Tracing of all-atom Chromatophore w/ lipids.

GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms. J. E. Stone, K. L. Vandivort, and K. Schulten. UltraVis'13, pp. 6:1-6:8, 2013.

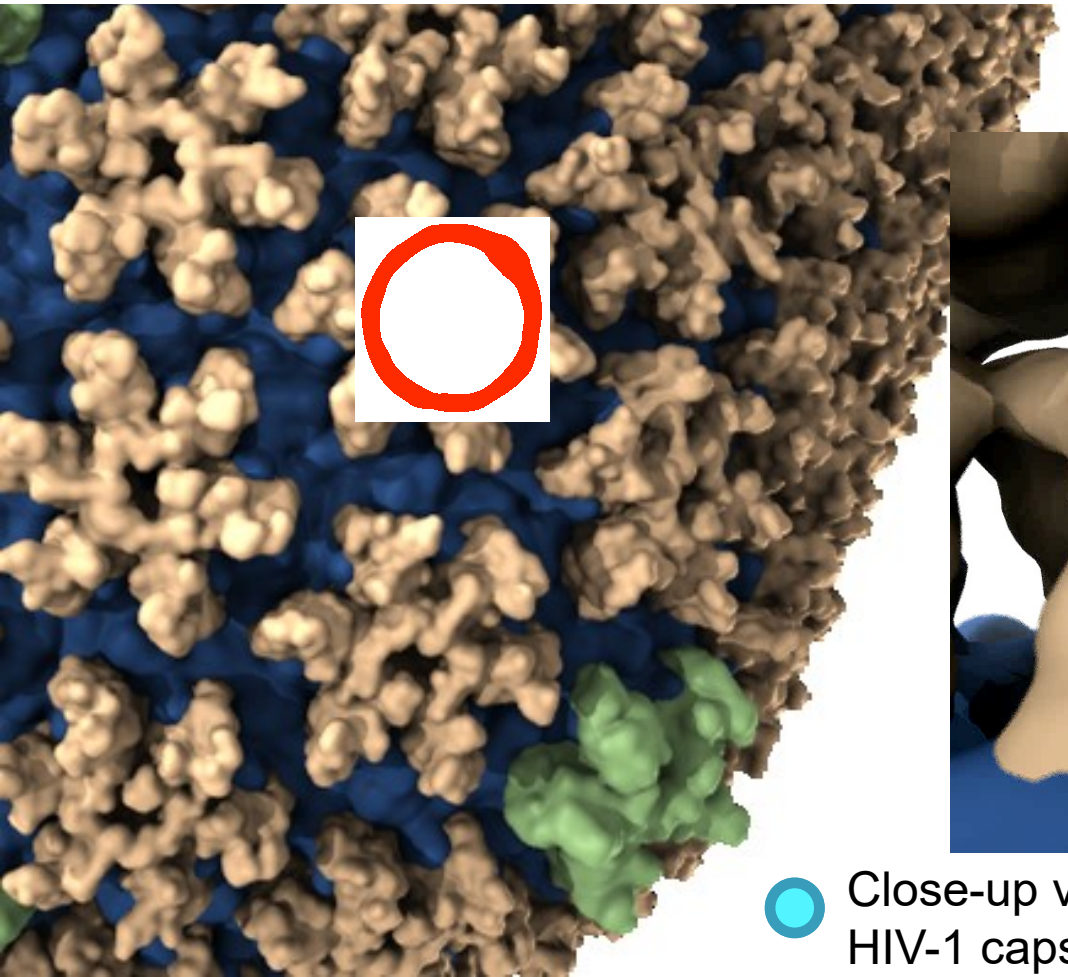
Visualization of Energy Conversion Processes in a Light Harvesting Organelle at Atomic Detail. M. Sener, et al. SC'14 Visualization and Data Analytics Showcase, 2014.

Chemical Visualization of Human Pathogens: the Retroviral Capsids. J. R. Perilla, B.-C. Goh, J. E. Stone, and K. Schulten. SC'15 Visualization and Data Analytics Showcase, 2015.

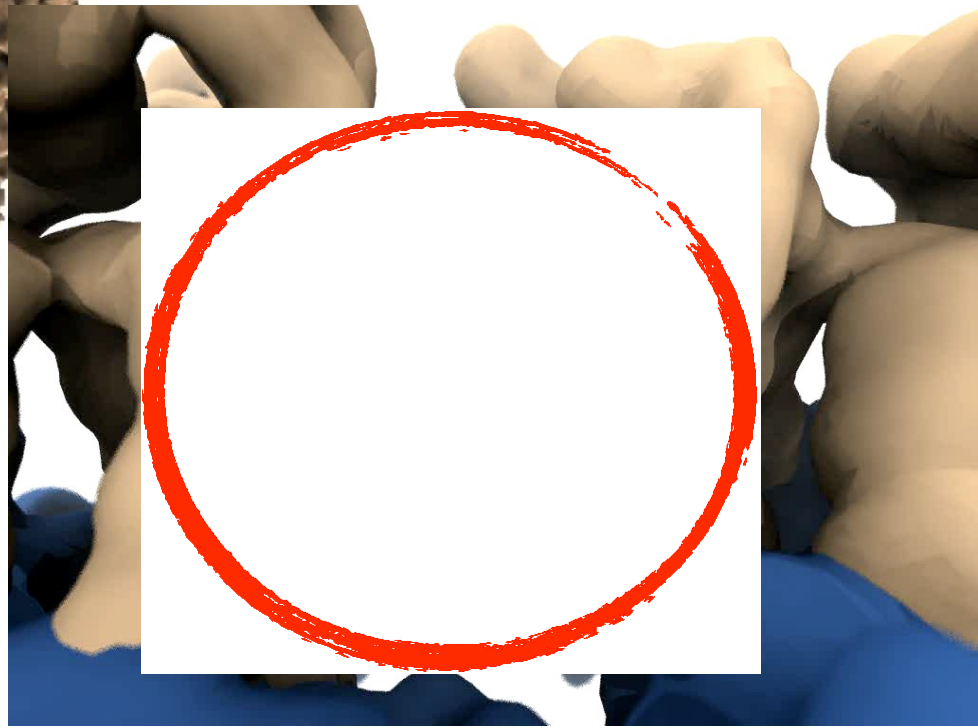
Atomic Detail Visualization of Photosynthetic Membranes with GPU-Accelerated Ray Tracing. J. E. Stone et al., J. Parallel Computing, 55:17-27, 2016.

Immersive Molecular Visualization with Omnidirectional Stereoscopic Ray Tracing and Remote Rendering J. E. Stone, W. R. Sherman, and K. Schulten. HPDAV, IPDPSW, pp. 1048-1057, 2016.

Goal: Intuitive interactive viz. in crowded molecular complexes

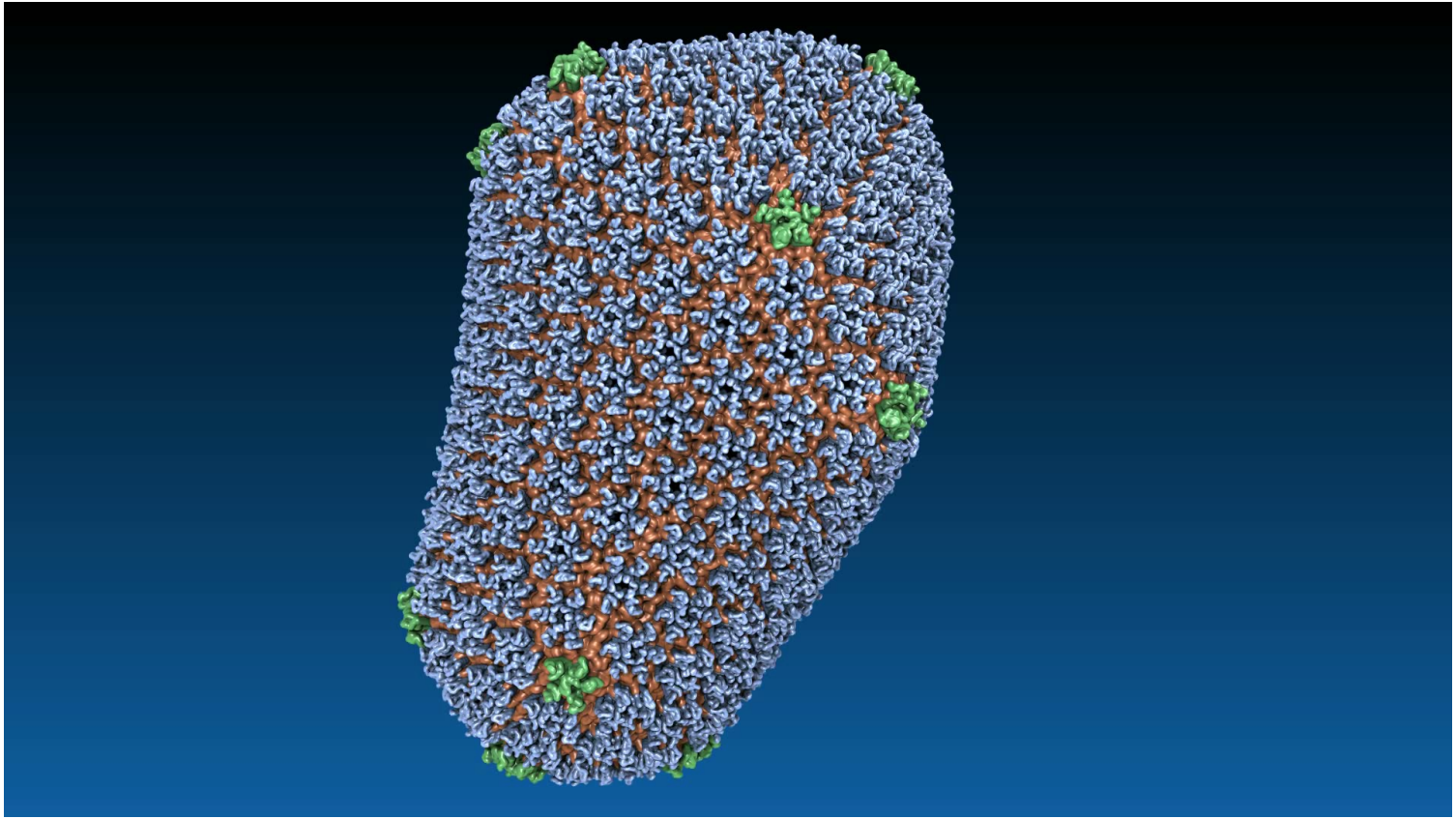


Results from 64 M atom, 1 μs sim!

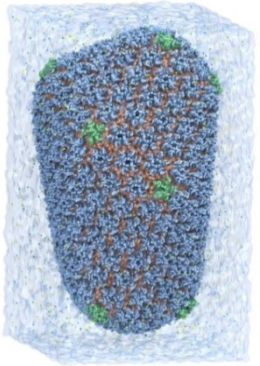


- Close-up view of chloride ions permeating through HIV-1 capsid hexameric centers

HIV-1 Capsid



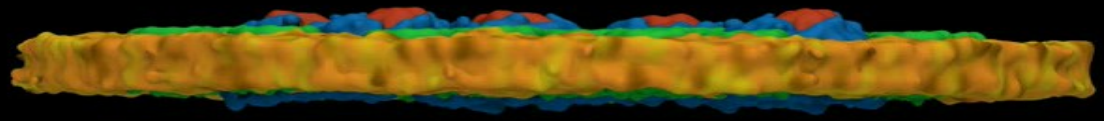
HIV-1 Parallel HD Movie Rendering on Blue Waters Cray XE6/XK7



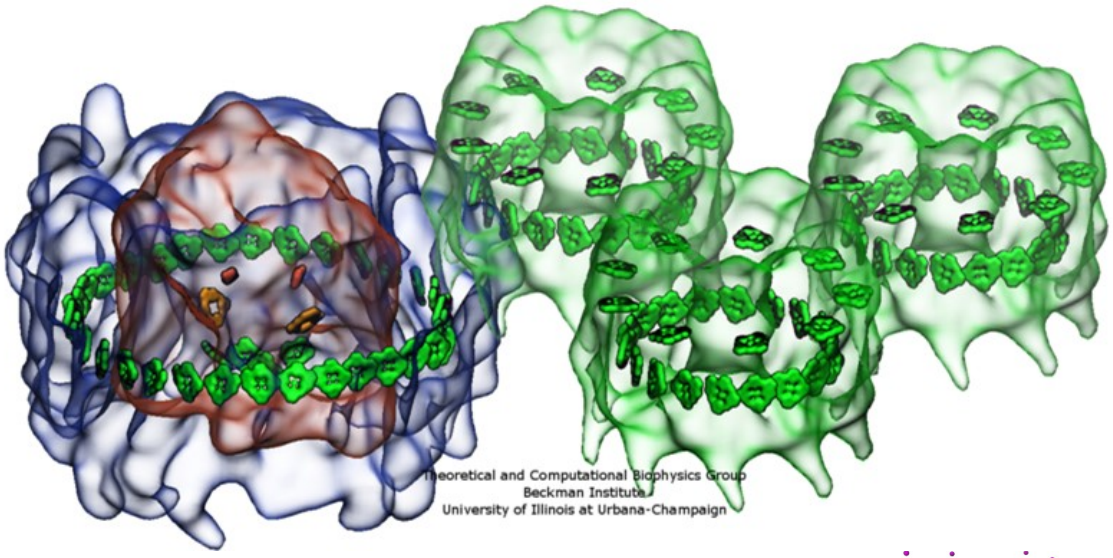
New VMD TachyonL-OptiX on XK7 vs. Tachyon on XE6:
K20X GPUs yield **up to twelve times** geom+ray tracing speedup

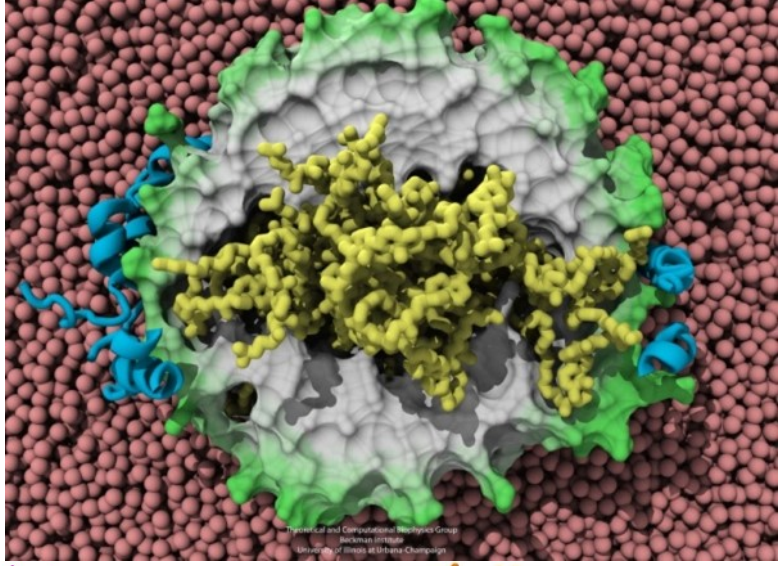
Ray Tracer Version	Node Type and Count	Script Load	State Load	Geometry + Ray Tracing	Total Time
New TachyonL-OptiX	64 XK7 Tesla K20X GPUs	2 s	39 s	435 s	476 s
New TachyonL-OptiX	**128 XK7 Tesla K20X GPUs**	**3 s**	**62 s**	**230 s**	**295 s**
TachyonL-OptiX [1]	64 XK7 Tesla K20X GPUs	2 s	38 s	655 s	695 s
TachyonL-OptiX [1]	128 XK7 Tesla K20X GPUs	4 s	74 s	331 s	410 s
TachyonL-OptiX [1]	**256 XK7 Tesla K20X GPUs**	**7 s**	**110 s**	**171 s**	**288 s**
Tachyon [1]	**256 XE6 CPUs**	7 s	160 s	**1,374 s**	**1,541 s**
Tachyon [1]	512 XE6 CPUs	13 s	211 s	808 s	1,032 s

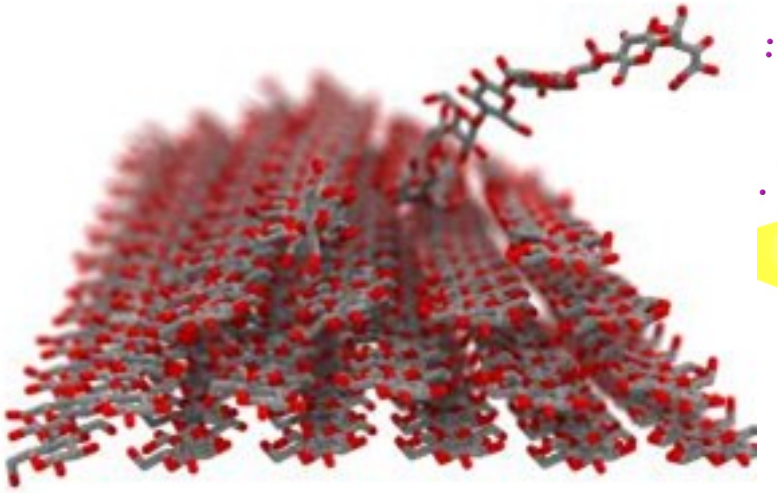
[1] GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms.
J. E. Stone, K. L. Vandivort, and K. Schulten. UltraVis'13: Proceedings of the 8th International Workshop on Ultrascale Visualization, pp. 6:1-6:8, 2013.

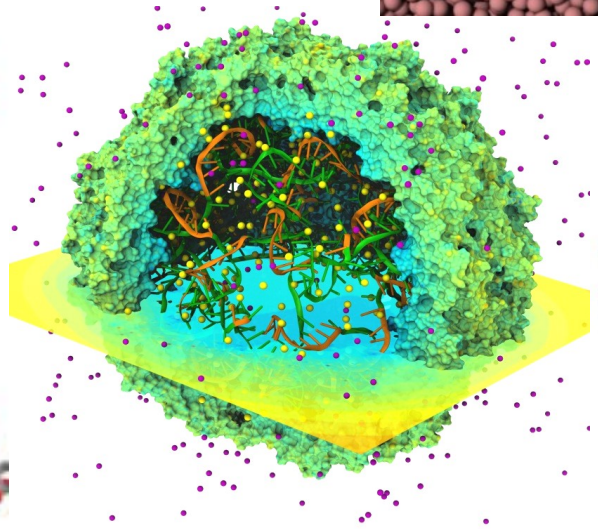


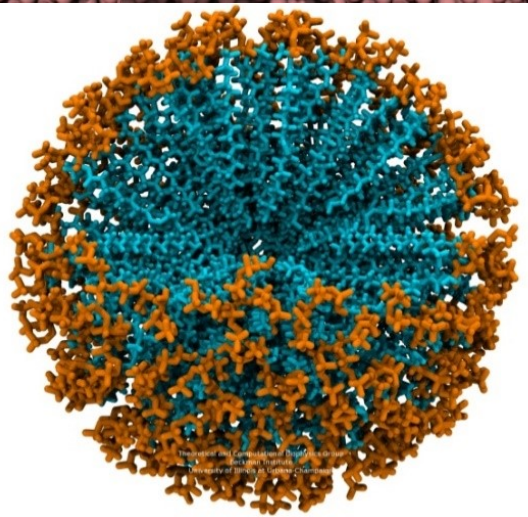
20 M atom chromatophore patch


Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign


Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign

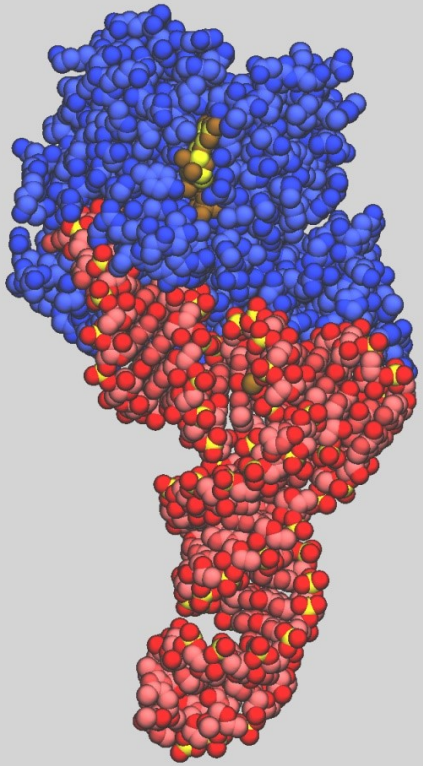




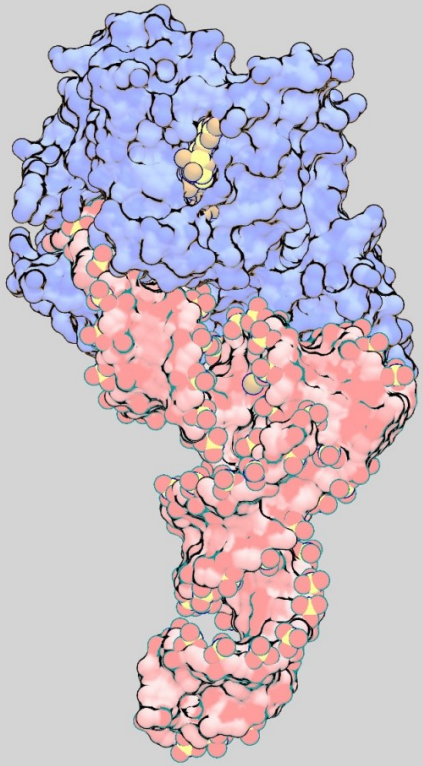


VMD Shading Comparison: EF-Tu

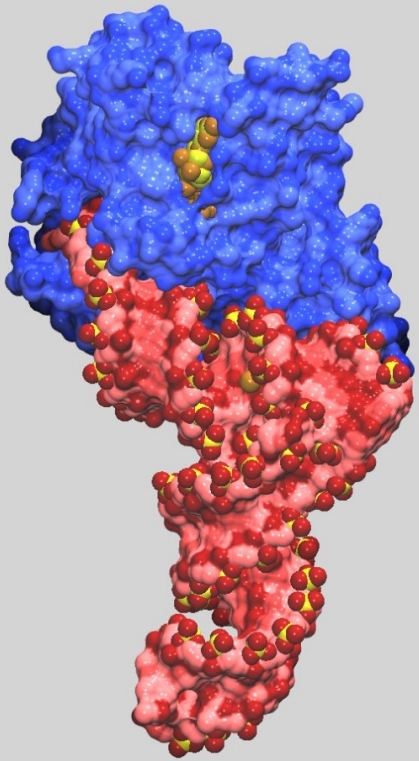
Outline Shader



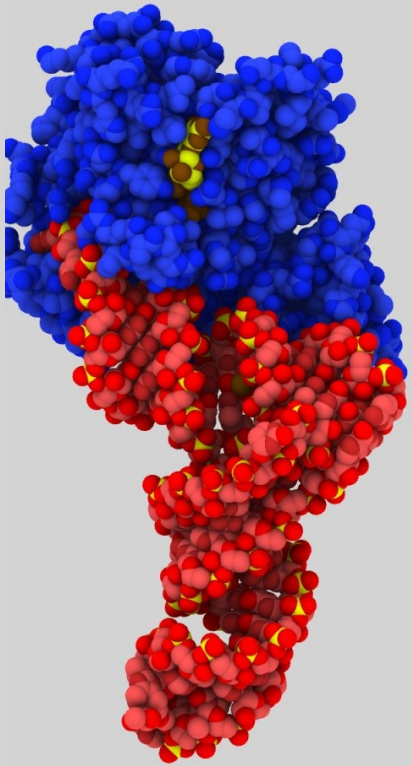
“Goodsell” Shader

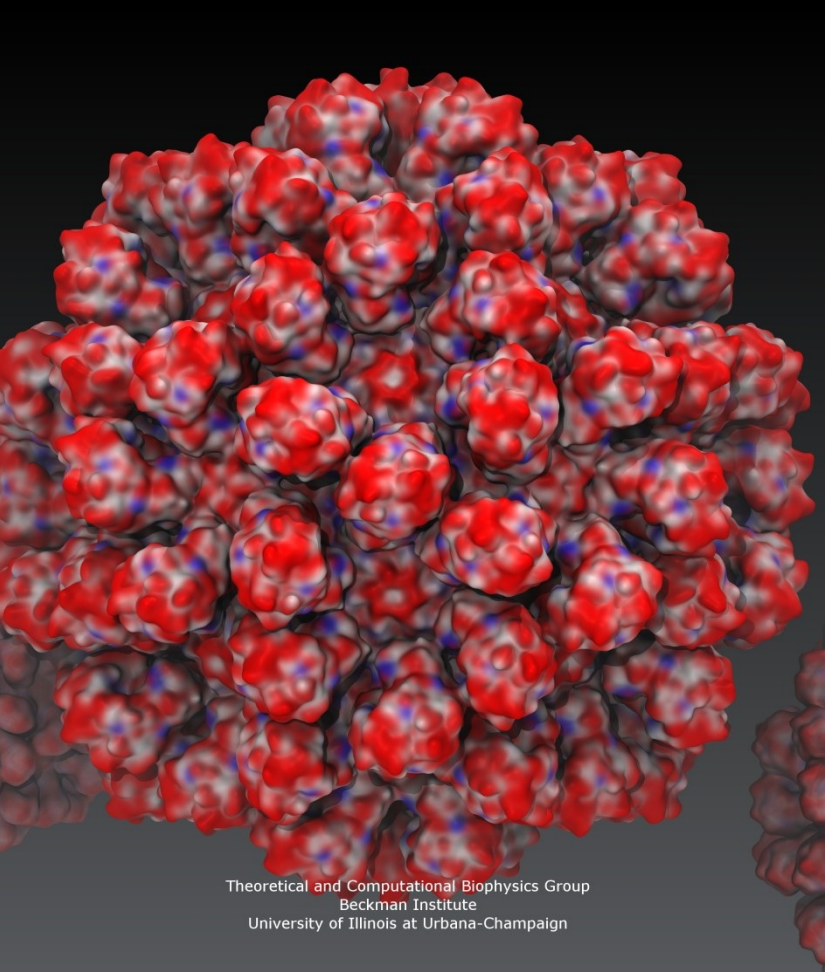


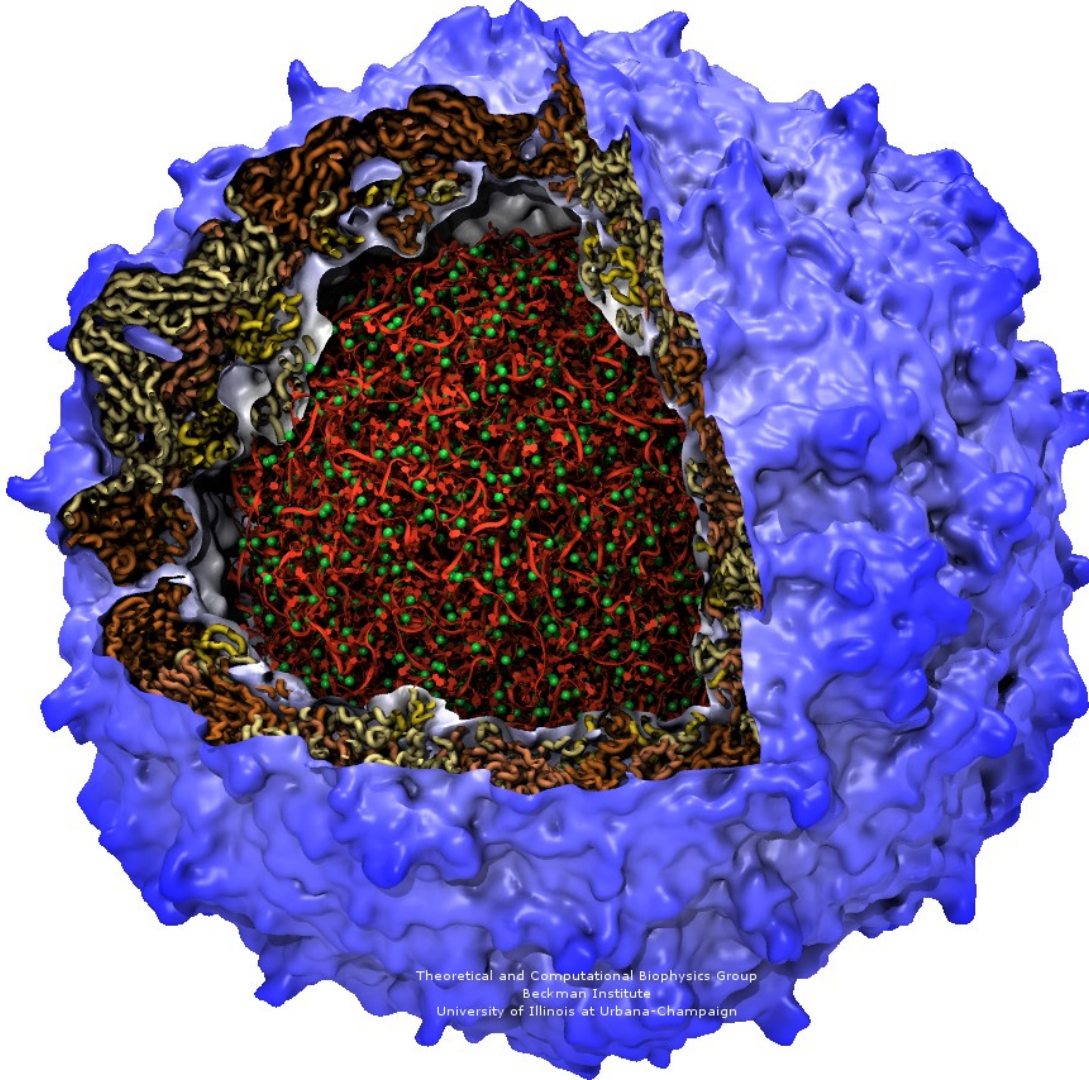
Glossy Shader



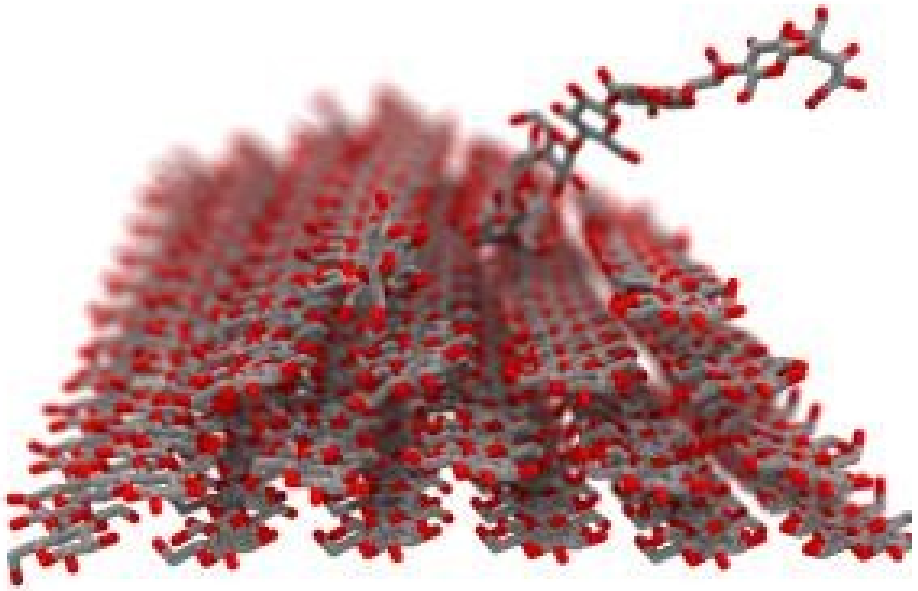
Ambient Occlusion, Shadowing



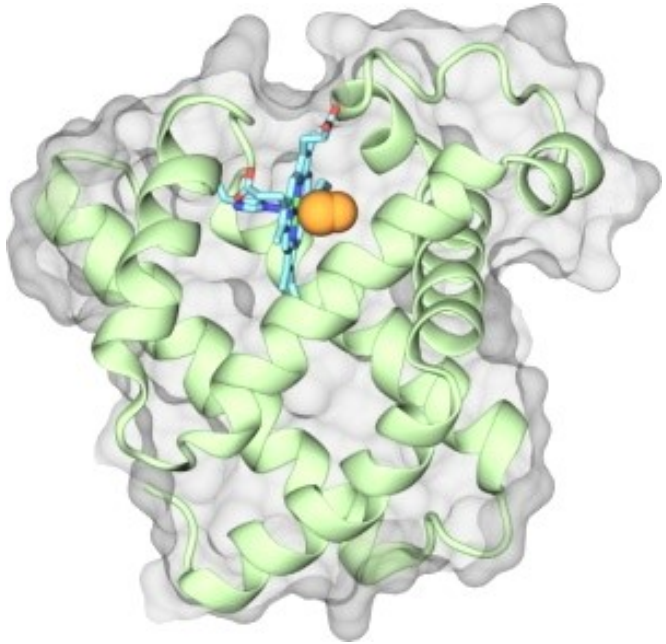

Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign


Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign

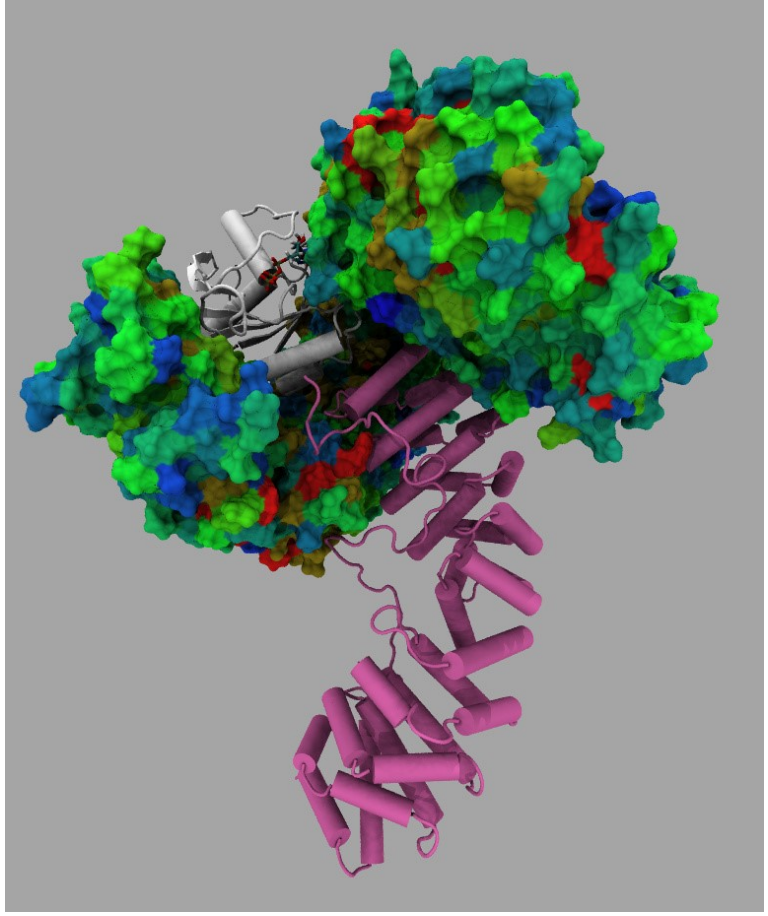
Diverse Shading and Lighting Approaches



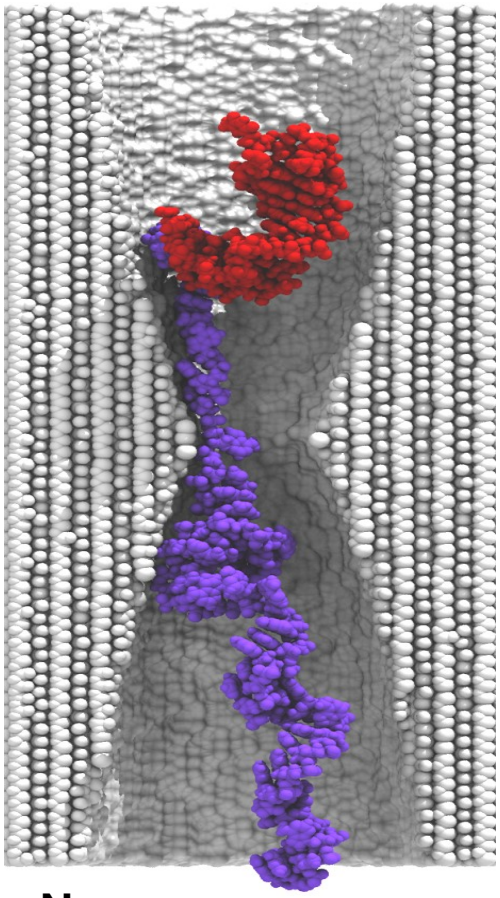
Decrystallization: Interactive Ray Tracing w/ Ambient Occlusion Lighting, Depth of Field Focal Blur



Myoglobin



Exportin Cse1p

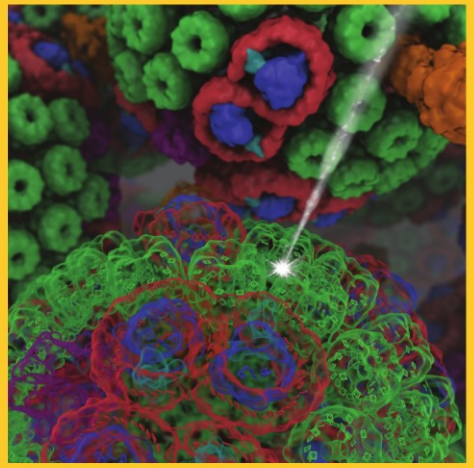


Nanopore

JPCBFK

APRIL 20, 2017
VOLUME 121
NUMBER 15
pubs.acs.org/JPCB

THE JOURNAL OF PHYSICAL CHEMISTRY B



KLAUS SCHULTEN MEMORIAL ISSUE

ACS Publications
Most Trusted. Most Cited. Most Read.

www.acs.org

Faraday Discussions

Volume: 169

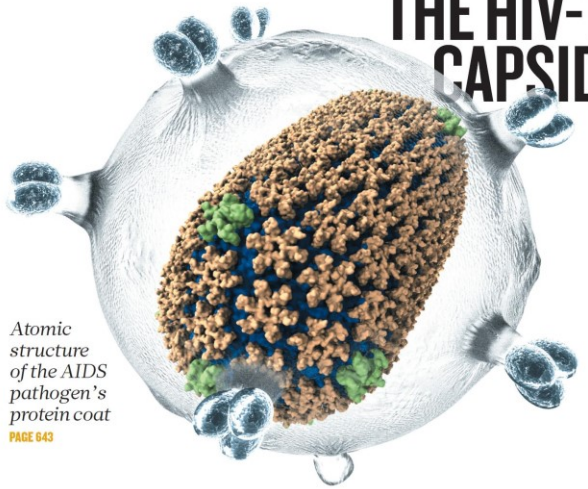
Molecular Simulations and Visualization

ROYAL SOCIETY OF CHEMISTRY

nature

THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

THE HIV-1 CAPSID



Atomic structure of the AIDS pathogen's protein coat

PAGE 643

COSMOLOGY
THE FIRST LIGHT
In pursuit of the most distant galaxies
PAGE 554

CITATION
CROSSING THE BORDERS
International collaborations make the most impact
PAGE 557

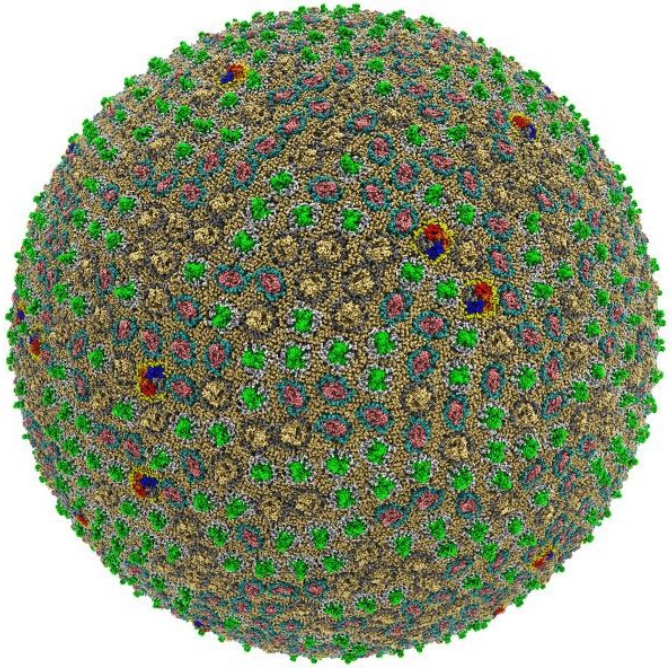
ANTICANCER DRUGS
A SITTING TARGET
An indirect hit on 'undruggable' KRAS protein
PAGES 577 & 638

NATURE.COM/NATURE
30 May 2013

Preparation, Visualization, Analysis of All-Atom Cell-Scale Simulations

- Interactive rasterization w/ OpenGL/EGL now, Vulkan in future releases of VMD
- **Interactive ray tracing on CPUs and GPUs**
- Support for large host memory (TB), up to **2 billion atoms per "molecule" now**
- Parallel analysis, visualization w/ MPI

- 200 nm spherical envelope
- Membrane with ~50% occupancy by proteins
- 63M atoms in envelope model

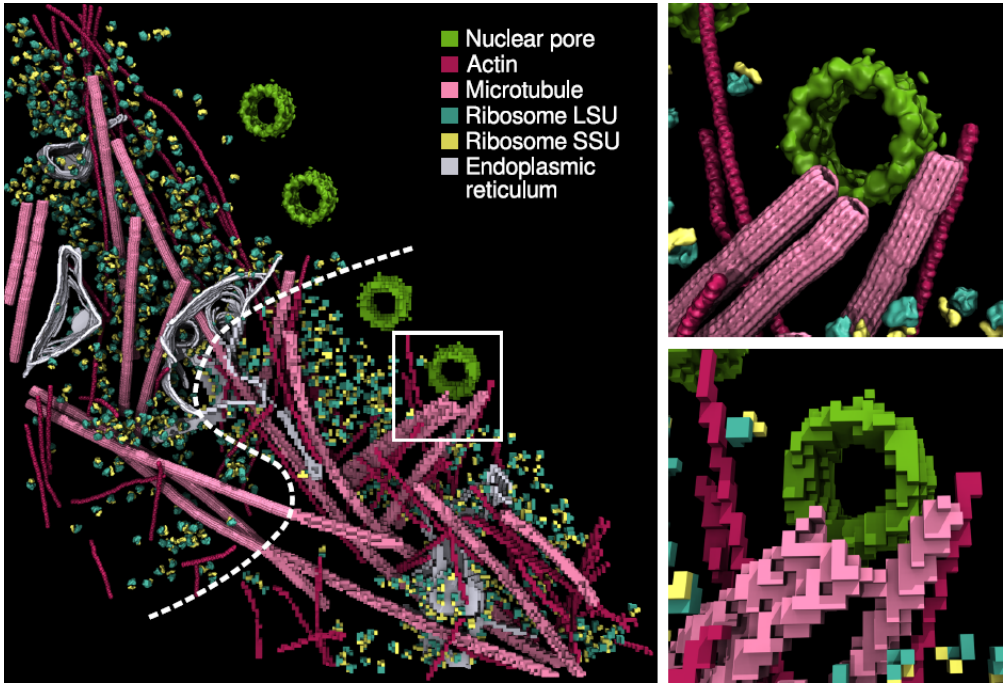


Atomic Detail Visualization of Photosynthetic Membranes with GPU-Accelerated Ray Tracing. J.E. Stone, …, K. Schulten, J. Parallel Computing, 55:17-27, 2016.

High Performance Molecular Visualization: In-Situ and Parallel Rendering with EGL. J.E. Stone, ..., K. Schulten. IEEE High Performance Data Analysis and Visualization, IPDPSW, pp. 1014-1023, 2016.

Interactive Ray Tracing of Cells

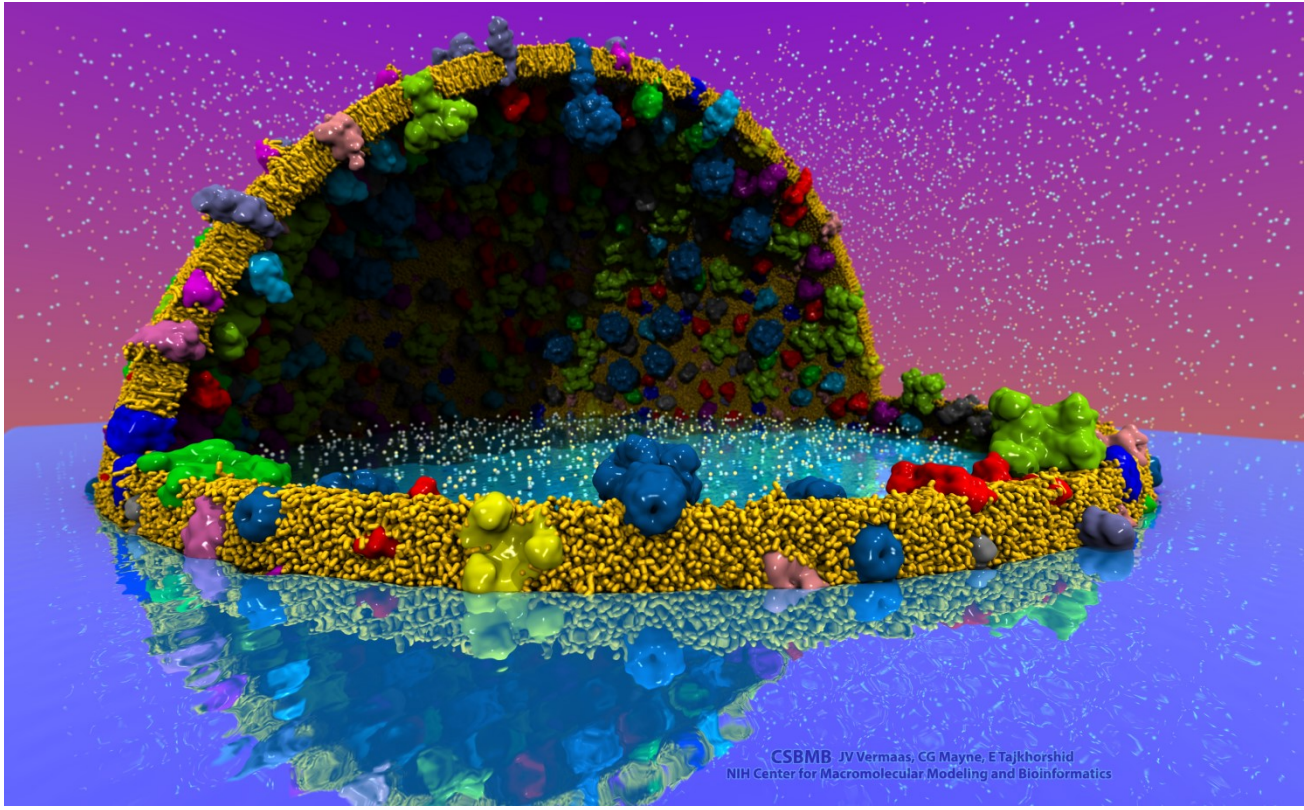
- High resolution cellular tomograms
- Multi-billion voxels
- Even isosurface or lattice site graphical representations involve ~100M geometric primitives
- 24GB Quadro M6000s used for interactive RT of cellular tomograms of this size
- **Latest Quadro GP100 GPUs benefit from OptiX 4.1 support for NVLink and distribution of scene data across multiple GPUs**



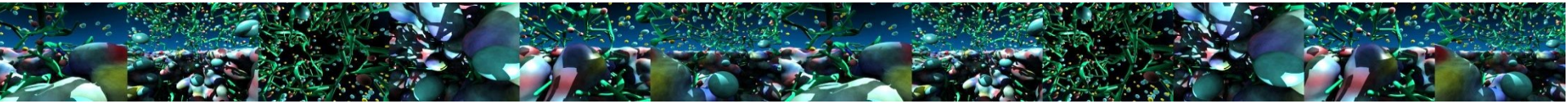
Earnest, et al. J. Physical Chemistry B, 121(15): 3871-3881, 2017.

Proto-Cell Rendered with VMD+OptiX

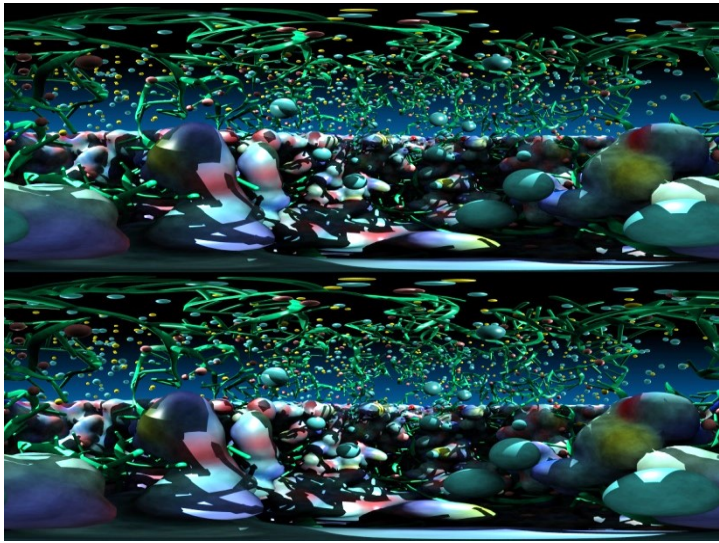
- 113M particles
- 1,397 copies of 14 different membrane proteins
- Preparing for simulations on pre-exascale computers



Stereoscopic Panorama Ray Tracing



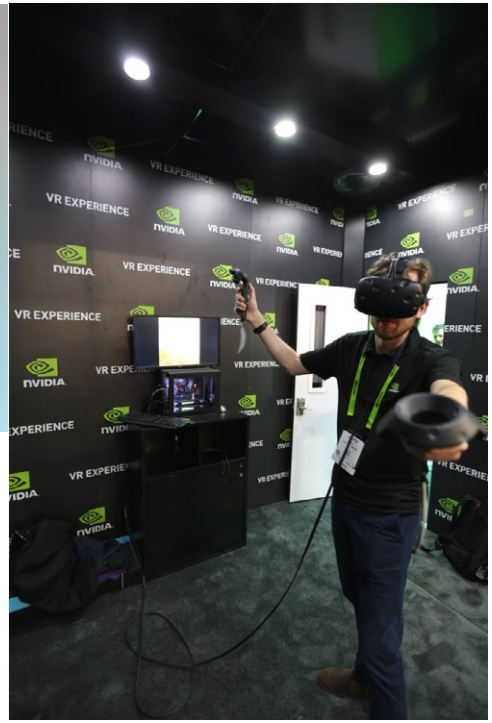
- **Render 360° images and movies for VR headsets such as Oculus Rift, Google Cardboard**
- Ray trace panoramic stereo spheremaps or cubemaps for very high-frame-rate display via OpenGL texturing onto simple geometry
- Stereo requires spherical camera projections **poorly suited to rasterization**
- Benefits from OptiX multi-GPU rendering and load balancing, **remote visualization**



VMD VR Demos

VMD VR ray tracing:
Google Cardboard [1]
Demo w/ Indiana U., SC'15 [2]

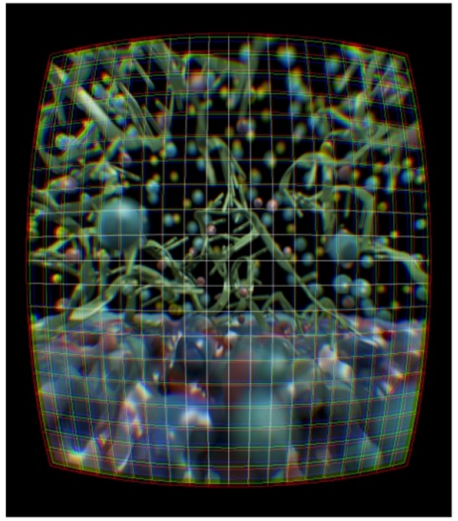
Prototype of VR user interaction with VMD models in **room-scale VR** with NVIDIA @ SC'16

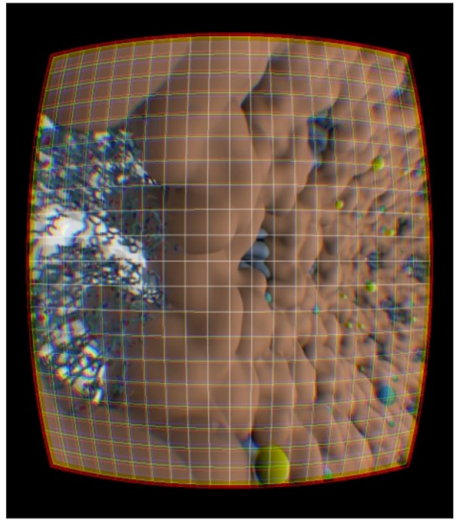


VMD Chromatophore Demo, NVIDIA VR Room at SC'16

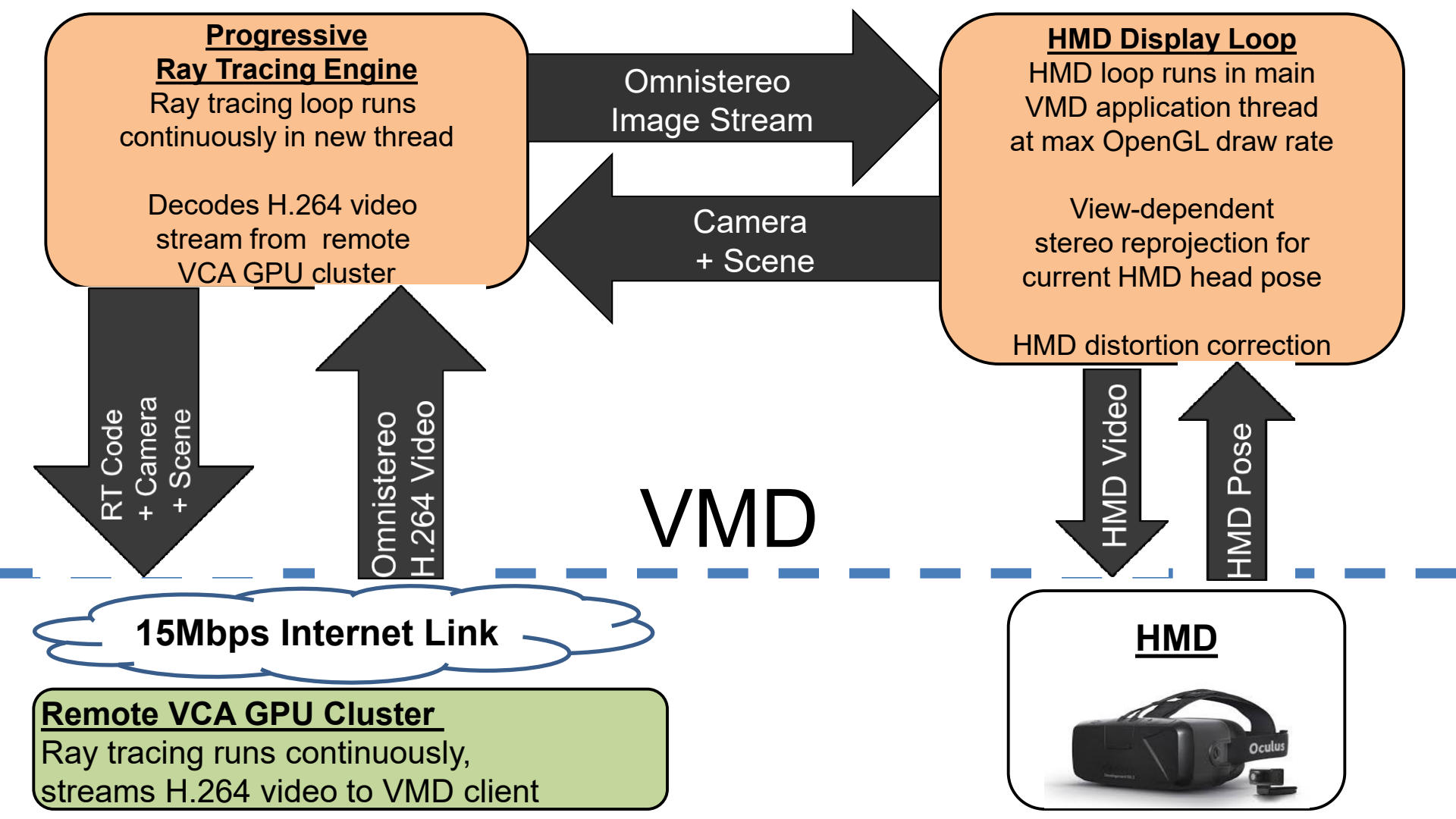
[1] Atomic Detail Visualization of Photosynthetic Membranes with GPU-Accelerated Ray Tracing. Stone et al., J. Parallel Computing, 55:17-27, 2016.

[2] Immersive Molecular Visualization with Omnidirectional Stereoscopic Ray Tracing and Remote Rendering. J.E. Stone, W.R. Sherman, K. Schulten. IEEE HPDAV (IPDPSW), pp. 1048-1057, 2016.

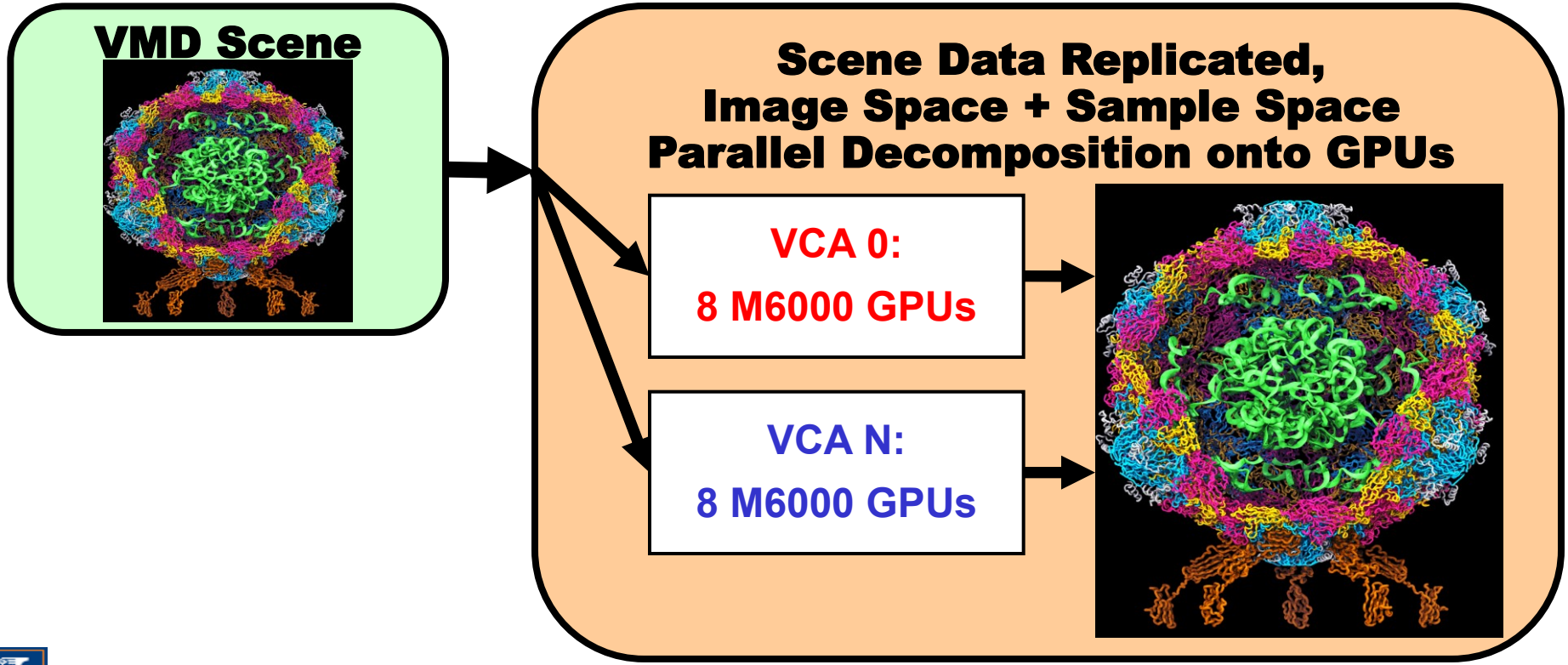




Immersive Molecular Visualization with Omnidirectional Stereoscopic Ray Tracing and Remote Rendering J. E. Stone, W. R. Sherman, and K Schulten. HPDAV, IPDPSW, pp. 1048-1057, 2016.


Progressive
Ray Tracing Engine
Ray tracing loop runs continuously in new thread
Decodes H.264 video stream from remote VCA GPU cluster
Omnistereo Image Stream
Camera + Scene
HMD Display Loop
HMD loop runs in main VMD application thread at max OpenGL draw rate
View-dependent stereo reprojection for current HMD head pose
HMD distortion correction
RT Code + Camera + Scene
Omnistereo H.264 Video
HMD Video
HMD Pose
VMD
15Mbps Internet Link
Remote VCA GPU Cluster
Ray tracing runs continuously, streams H.264 video to VMD client
HMD
Oculus

VMD TachyonL-OptiX: Multi-GPU on NVIDIA VCA Cluster

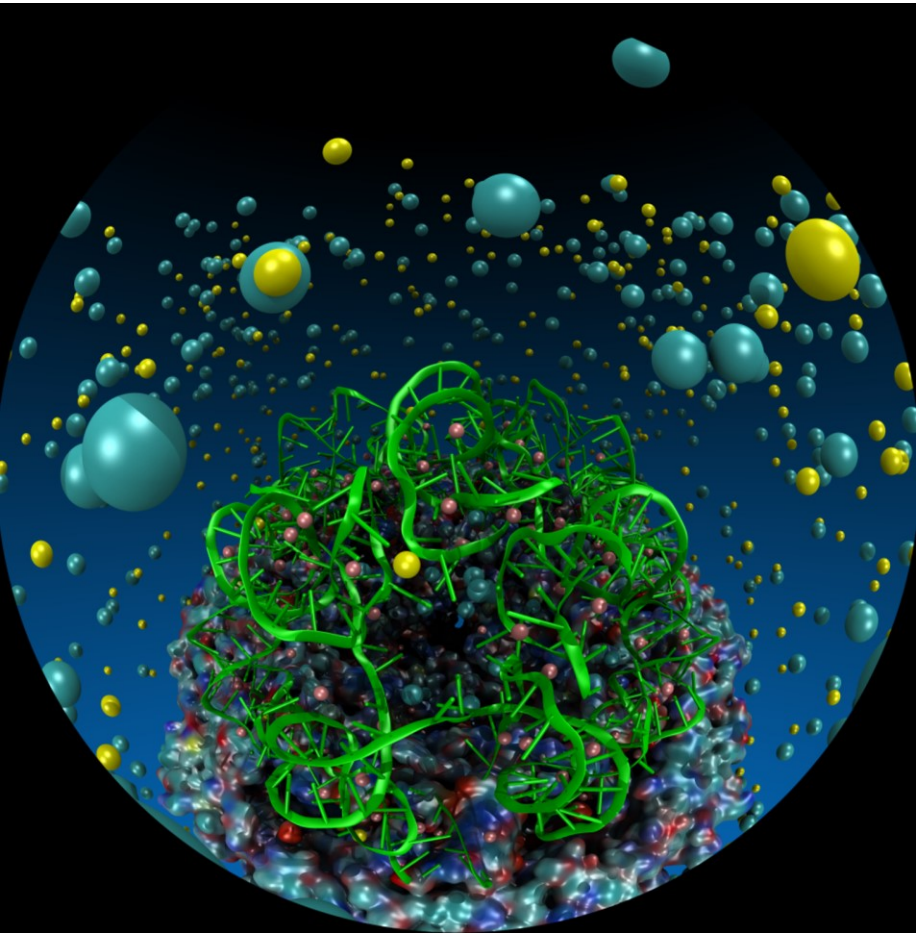


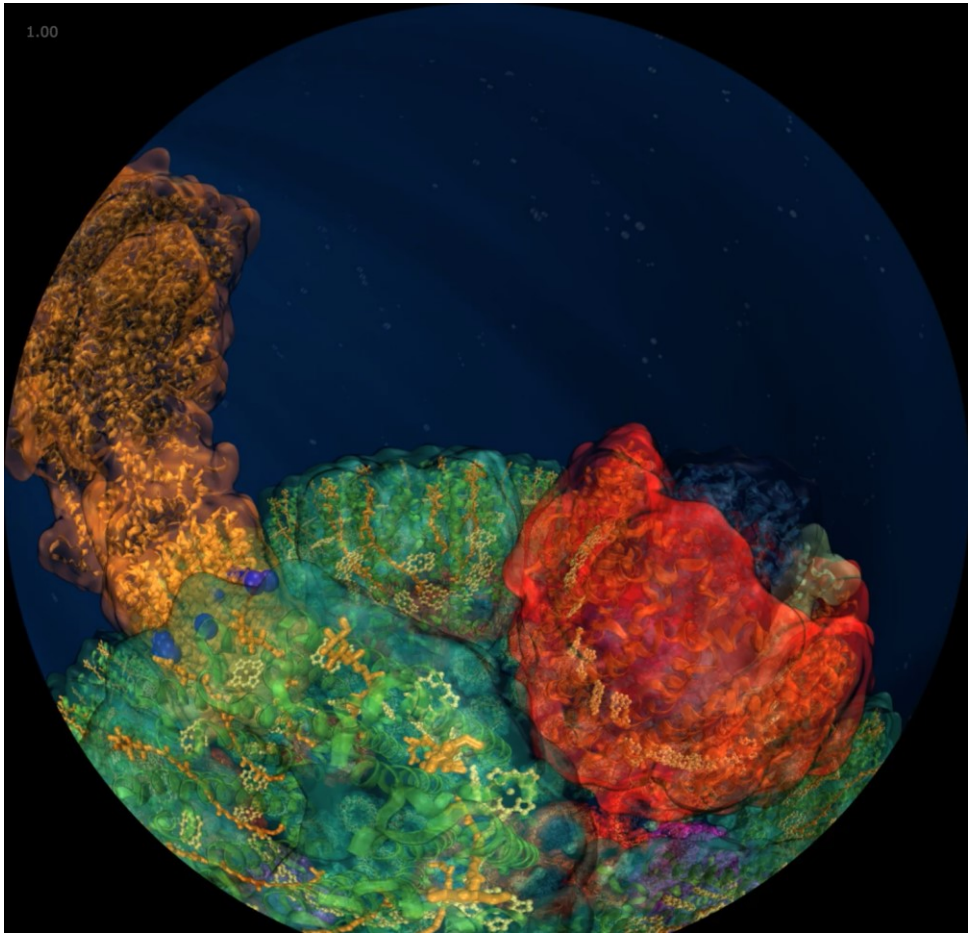
Remote Omnidirectional Stereoscopic RT Performance @ 3072x1536 w/ 2-subframes

Scene	Per-subframe samples AA : AO (AO per-hit)	RT update rate (FPS)
STMV shadows	1:0 2:0 4:0	22.2 18.1 10.3
STMV Shadows+AO	1:1 1:2 1:4	18.2 16.1 12.4
STMV Shadows+AO+DoF	1:1 2:1 2:2	16.1 11.1 8.5
HIV-1 Shadows	1:0 2:0 4:0	20.1 18.1 10.2
HIV-1 Shadows+AO	1:1 1:2 1:4	17.4 12.2 8.1

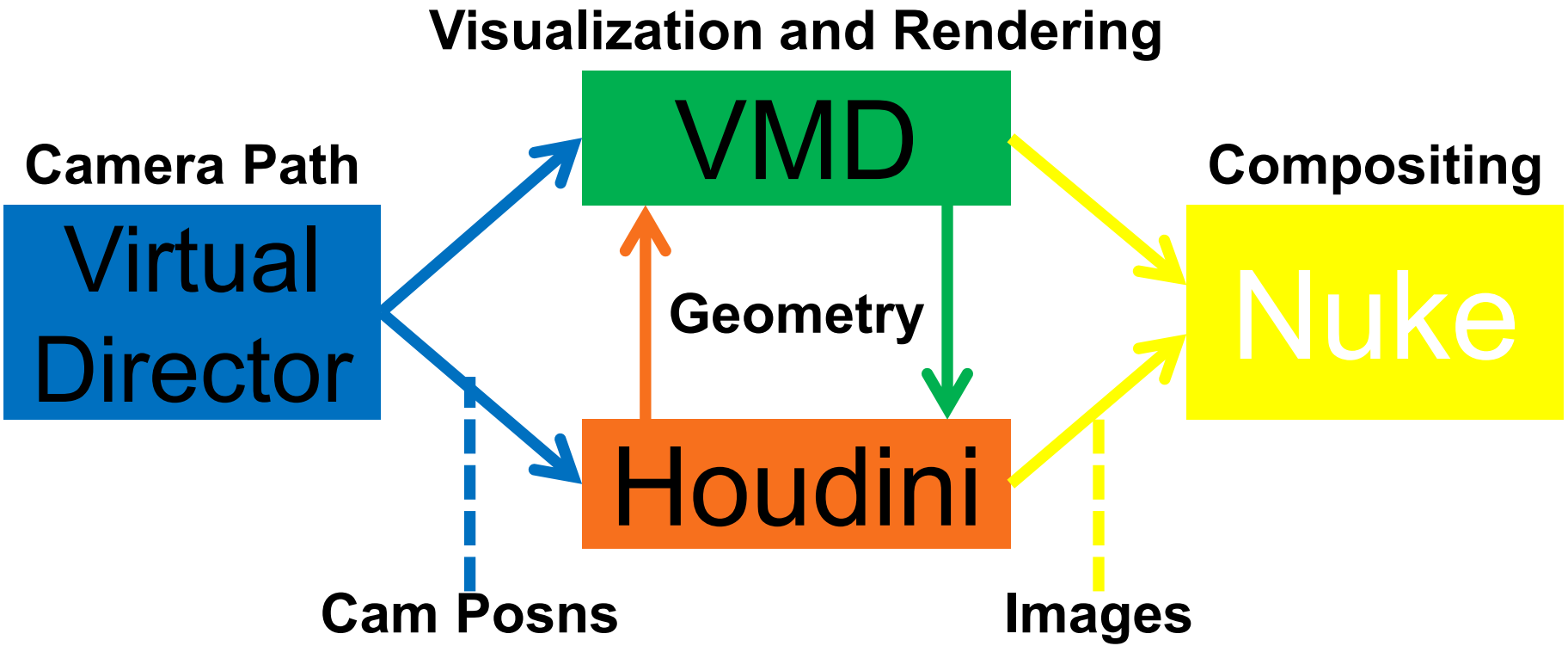
NIH

Planetarium Dome Master Projections

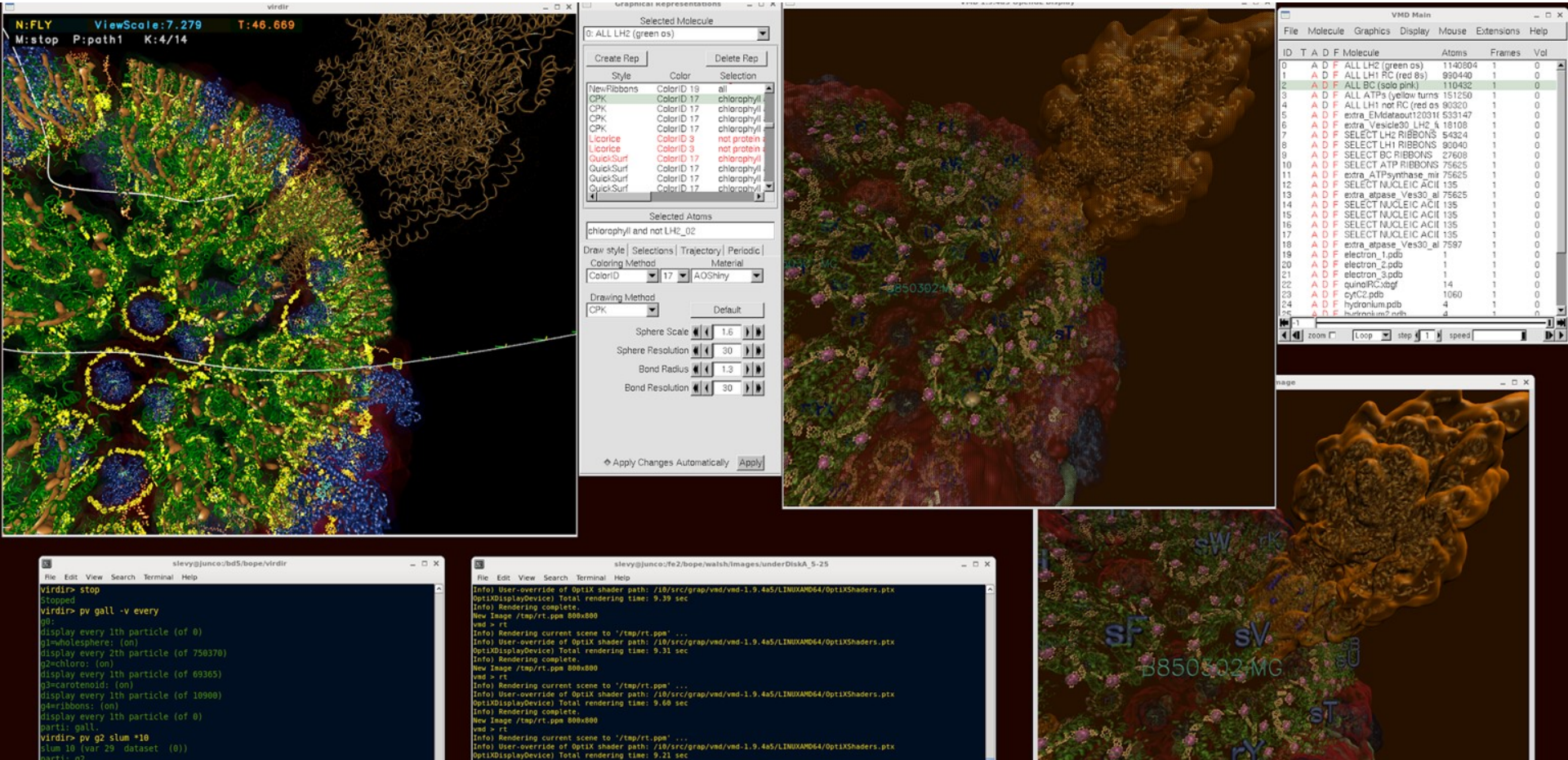


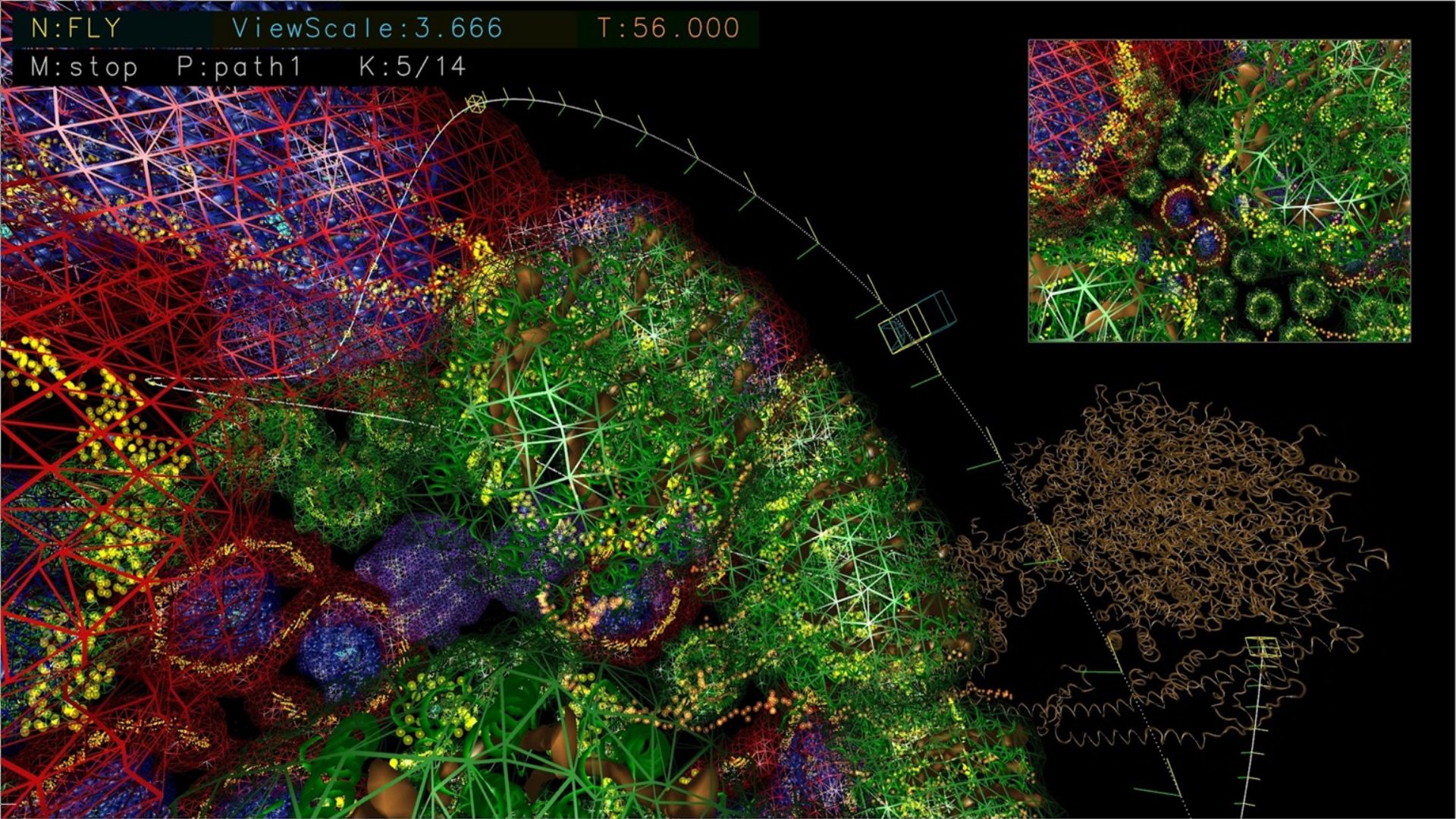


NSF CADENS "Birth of Planet Earth" Cinematic Pipeline

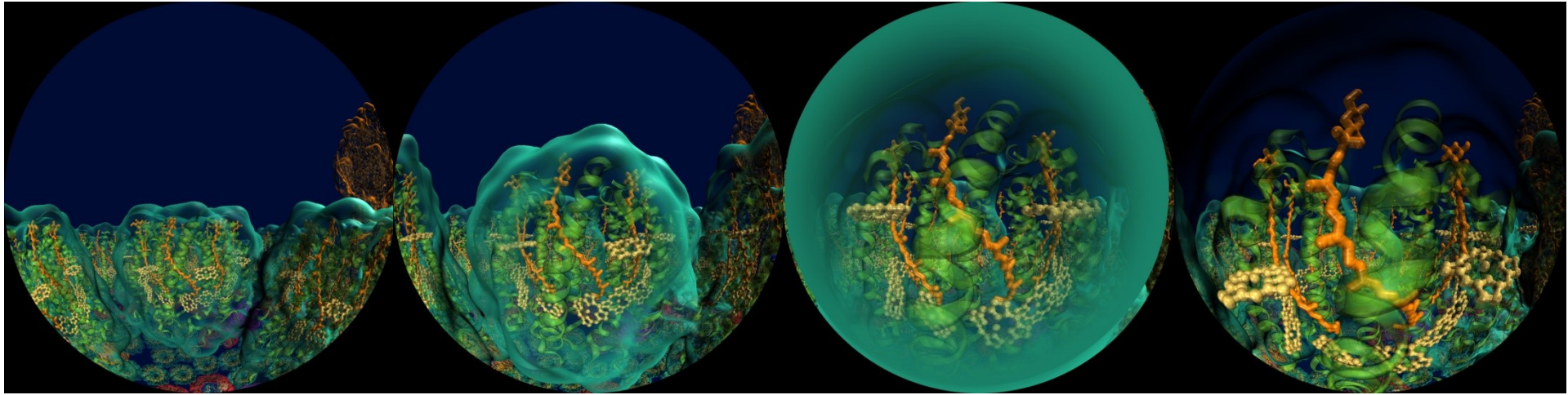


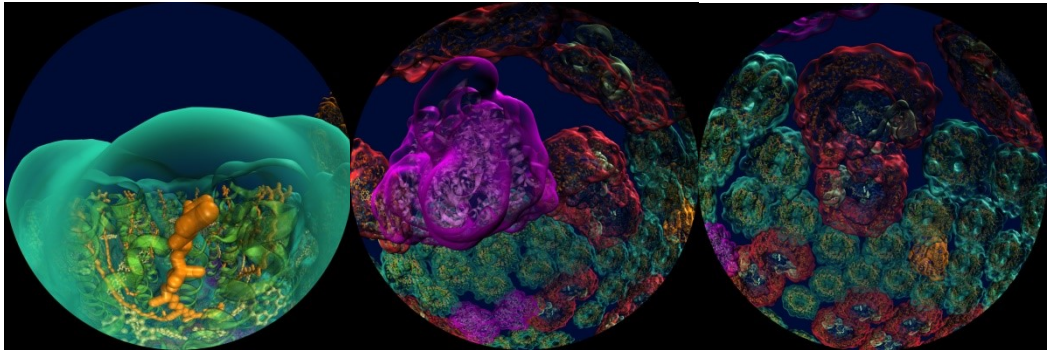
Integration of VMD with Virtual Director




N:FLY
ViewScale:3.666
T:56.000
M:stop P:path1 K:5/14

NSF CADENS “Birth of Planet Earth” Test Frames

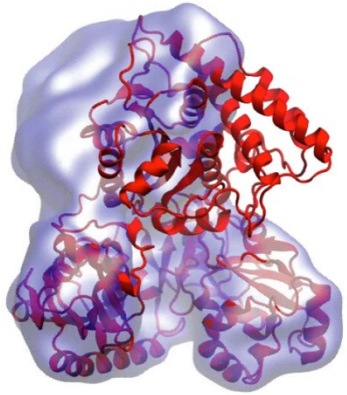




Molecular Dynamics Flexible Fitting (MDFF)

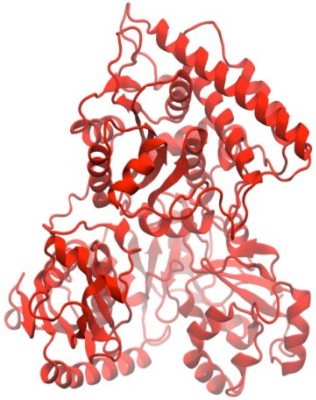
X-ray crystallography → MDFF ← Electron microscopy

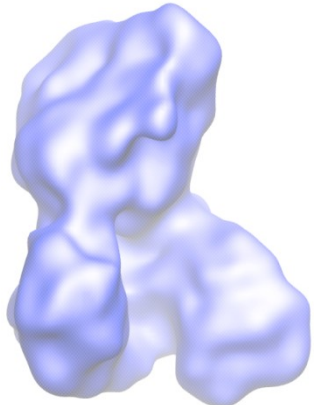
APS at Argonne



ORNL Titan

FEI microscope





Molecular dynamics-based refinement and validation for sub-5Å cryo-electron microscopy maps. A. Singharoy, I. Teo, R. McGreevy, J. E. Stone, J. Zhao, and K. Schulten. eLife 2016;10.7554/eLife.16105

Molecular Dynamics Flexible Fitting - Theory

Two terms are added to the MD potential

$$U_{total} = U_{MD} + U_{EM} + U_{SS}$$

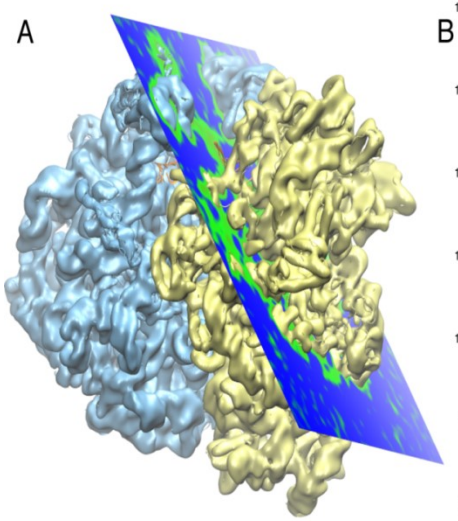
An external potential derived from the EM map is defined on a grid as

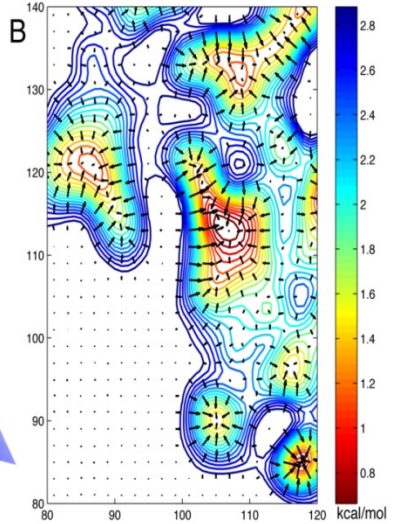
$$U_{EM}(\mathbf{R}) = \sum_j w_j V_{EM}(\mathbf{r}_j)$$

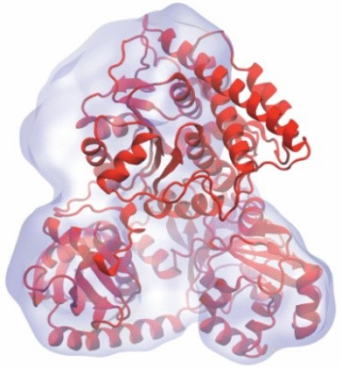
$$V_{EM}(\mathbf{r}) = \begin{cases} \xi \left(1 - \frac{\Phi(\mathbf{r}) - \Phi_{thr}}{\Phi_{max} - \Phi_{thr}}\right) & \text{if } \Phi(\mathbf{r}) \geq \Phi_{thr}, \\ \xi & \text{if } \Phi(\mathbf{r}) < \Phi_{thr}. \end{cases}$$

A mass-weighted force is then applied to each atom

$$\mathbf{f}_i^{EM} = -\nabla U_{EM}(\mathbf{R}) = -w_i \partial V_{EM}(\mathbf{r}_i)/\partial r_i$$

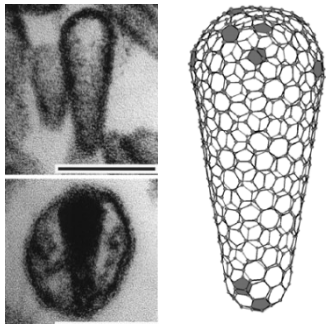






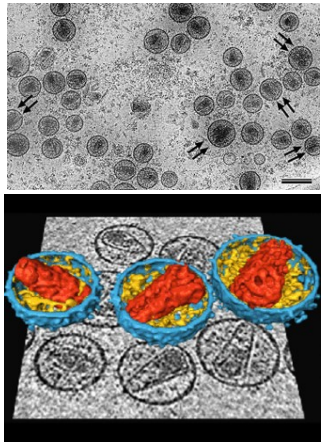
Structural Route to the all-atom HIV-1 Capsid

1st TEM (1999)



Ganser et al. *Science*, 1999

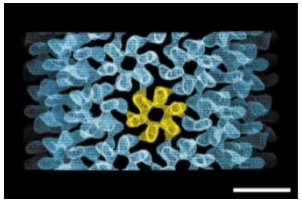
1st tomography (2003)



Briggs et al. *EMBO J*, 2003

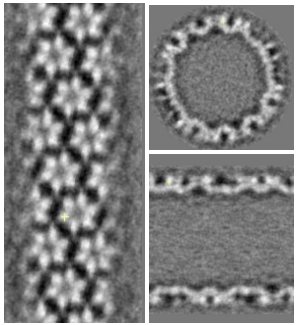
Briggs et al. *Structure*, 2006

hexameric tubule



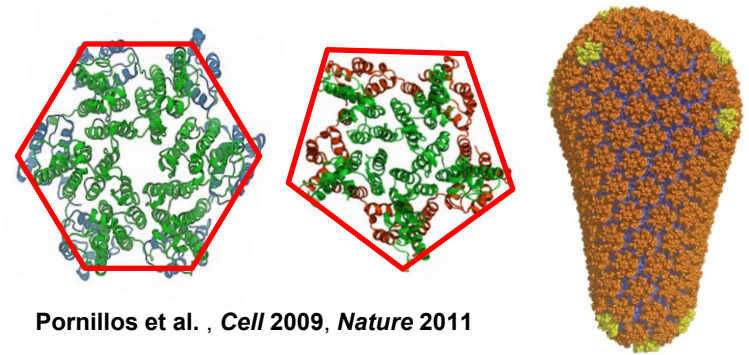
Li et al., *Nature*, 2000

cryo-ET (2006)



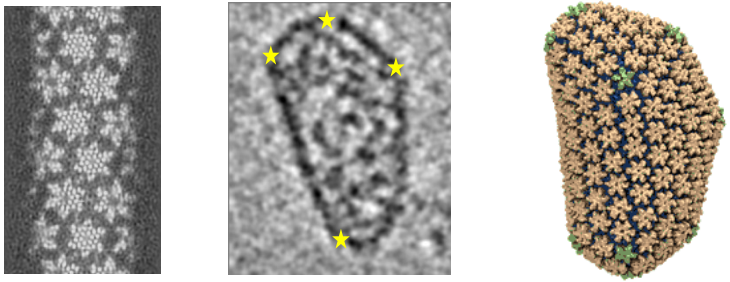
Byeon et al., *Cell* 2009

Crystal structures of separated hexamer and pentamer



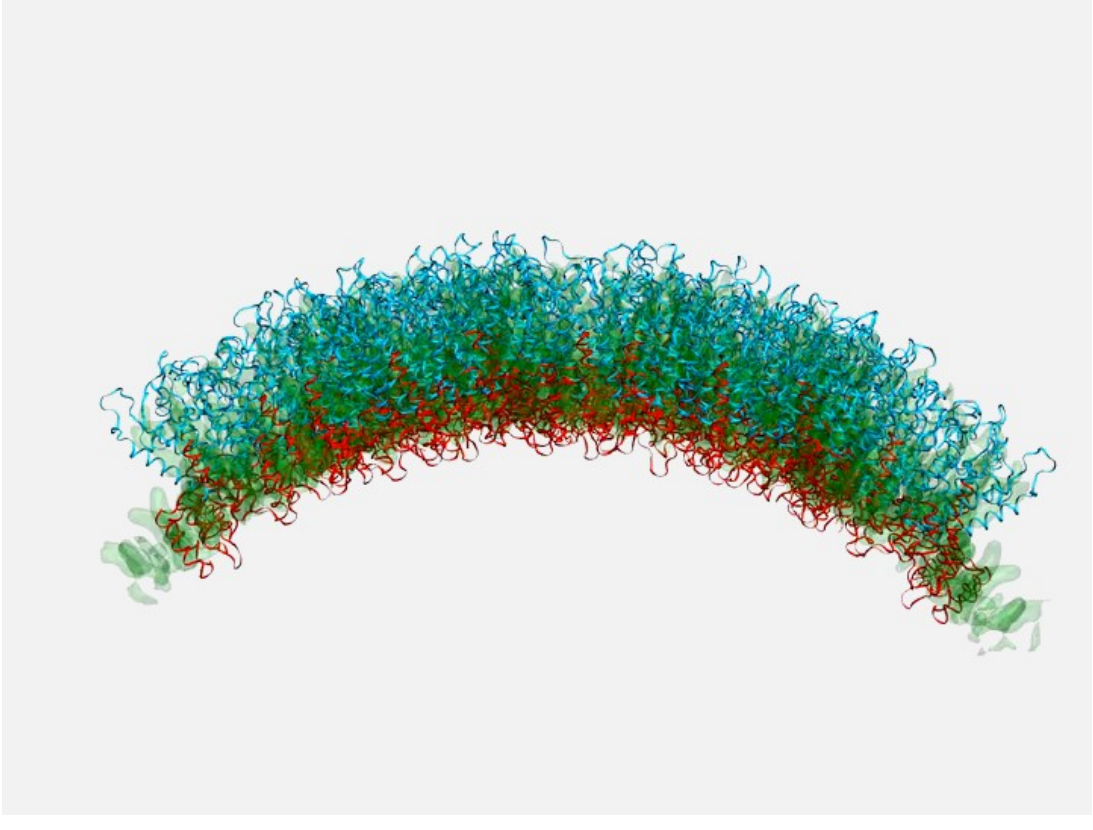
Pornillos et al. , *Cell* 2009, *Nature* 2011

High res. EM of hexameric tubule, tomography of capsid,
all-atom model of capsid by MDFF w/ NAMD & VMD,
NSF/NCSA Blue Waters computer at Illinois



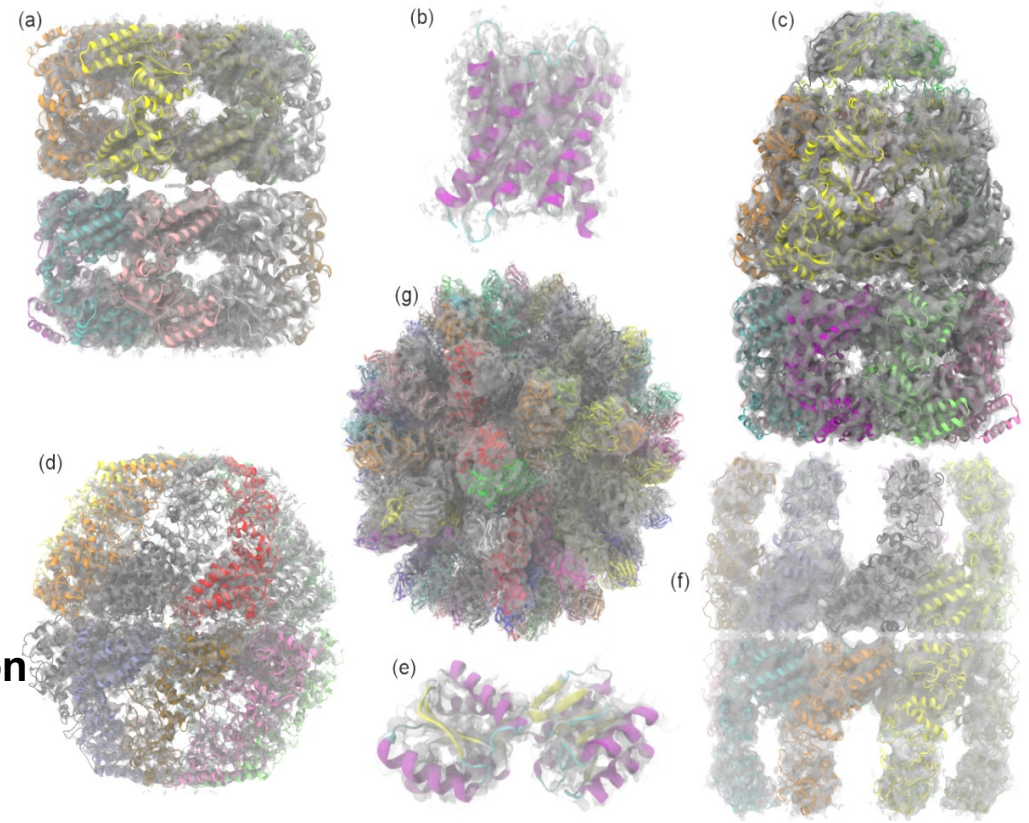
Zhao et al. , *Nature* 497: 643-646 (2013)

Hexamer of hexamers HIV capsid substructure Molecular Dynamics Flexible Fitting (MDFF) simulation. All-atom structure fitting into cryo-EM density map.



Evaluating Quality-of-Fit for Structures Solved by Hybrid Fitting Methods

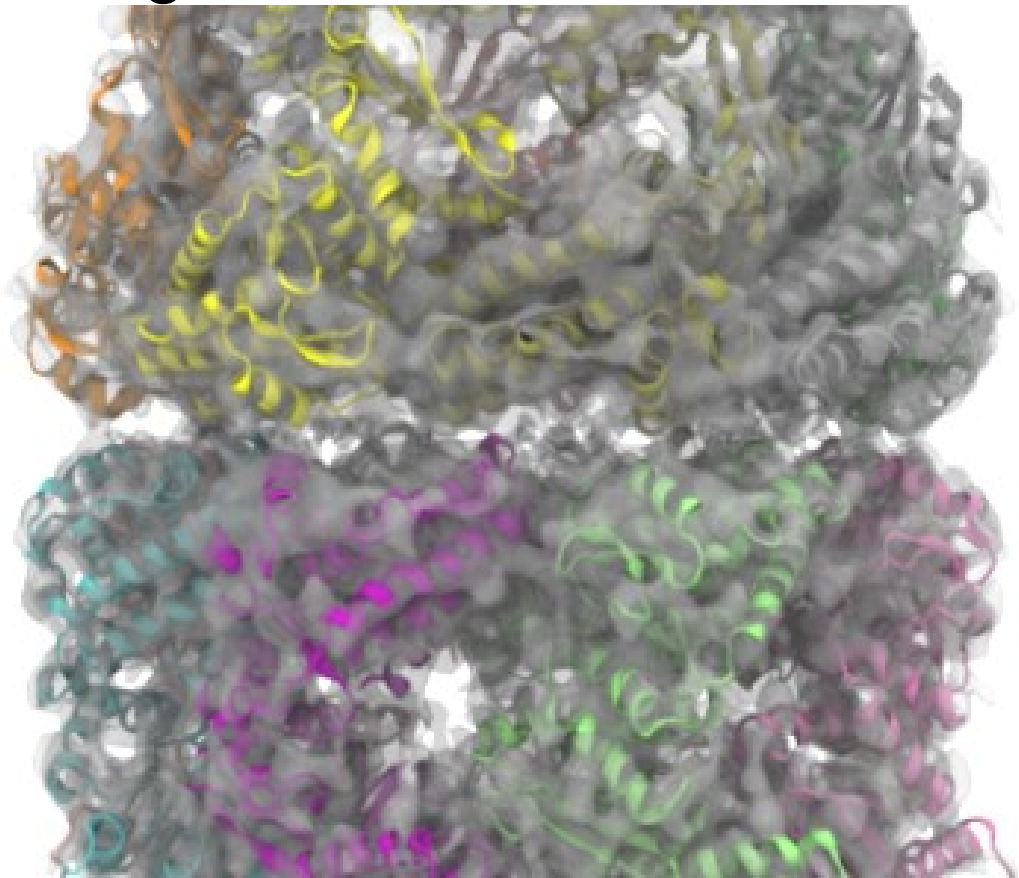
Compute Pearson correlation to evaluate the fit of a reference **cryo-EM density map** with a **simulated density map** produced from an **all-atom structure**.



GPU-Accelerated Analysis and Visualization of Large Structures Solved by Molecular Dynamics Flexible Fitting. Stone et al.. Faraday Discussions 169:265-283, 2014.

Evaluating Quality-of-Fit for Structures Solved by Hybrid Fitting Methods

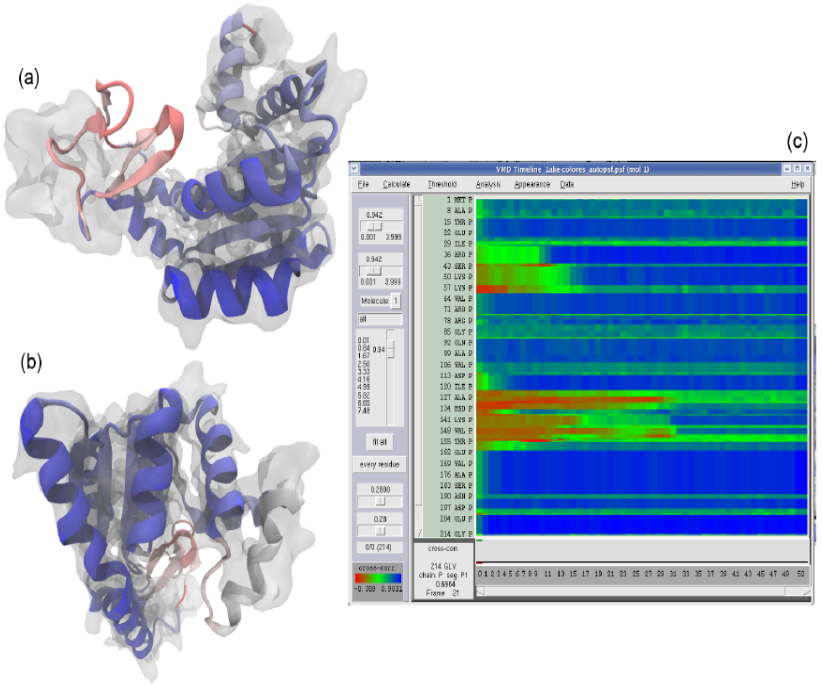
Compute Pearson correlation to evaluate quality-of-fit between a reference **cryo-EM density map** and a **simulated density map** produced from an **all-atom structure**.



GPUs Can Reduce MDFF Trajectory Analysis Runtimes from Hours to Minutes

GPUs enable laptops and desktop workstations to handle tasks that would have previously required a cluster, or a *very long wait*…

GPU-accelerated petascale supercomputers enable analyses that were previously impractical, allowing detailed study of very large structures such as viruses



GPU-accelerated MDFF Cross Correlation Timeline

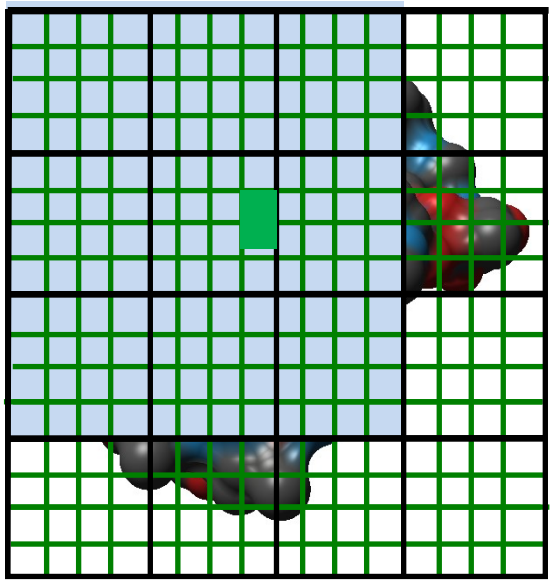
Regions with poor fit **Regions with good fit**

MDFF Density Map Algorithm

- Build spatial acceleration data structures, optimize data for GPU
- Compute 3-D density map:

$$\rho(\vec{r}; \vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N) = \sum_{i=1}^{N} e^{\frac{-|\vec{r}-\vec{r}_i|^2}{2\alpha^2}}$$

- Truncated Gaussian and spatial acceleration grid ensure linear time-complexity



3-D density map lattice point and the neighboring spatial acceleration cells it references

Single-Pass MDFF GPU Cross-Correlation

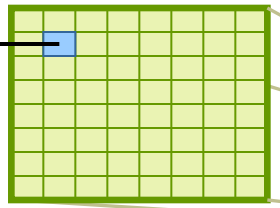
3-D density map decomposes into 3-D grid of 8x8x8 tiles containing CC partial sums and local CC values

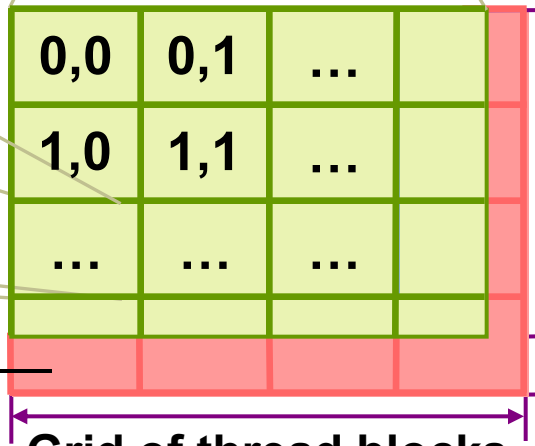
Fusion of density and CC calculations into a single CUDA kernel!!!

Spatial CC map and overall CC value computed in a single pass

Small 8x8x2 CUDA thread blocks afford large per-thread register count, shared memory

Each thread computes 4 z-axis density map lattice points and associated CC partial sums

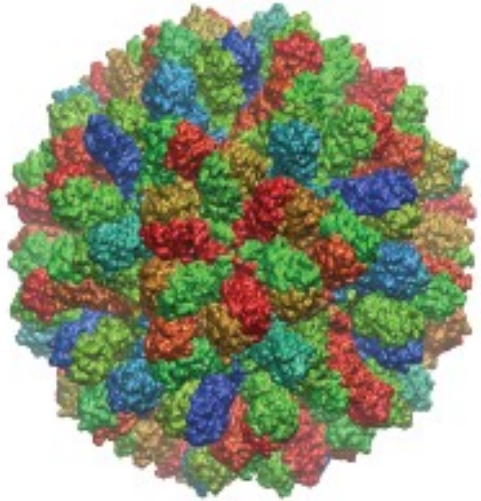




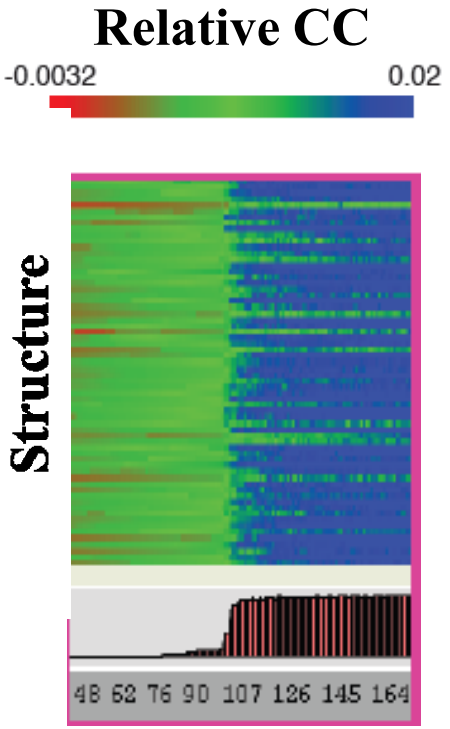
Padding optimizes global memory performance, guaranteeing coalesced global memory accesses

Parallel MDFF Cross Correlation Analysis on Cray XK7

Rabbit Hemorrhagic Disease Virus (RHDV)	
Traj. frames	10,000
Structure component selections	720
Single-node XK7 (projected)	336 hours (14 days)
128-node XK7	3.2 hours 105x speedup
2048-node XK7	19.5 minutes 1035x speedup



RHDV colored by relative CC



Calculation of 7M CCs would take **5 years** using serial CPU algorithm!

Stone et al., Faraday Discuss., 169:265-283, 2014.

VMD Tesla P100 Cross Correlation Performance

Rabbit Hemorrhagic Disease Virus: 702K atoms, 6.5Å resolution

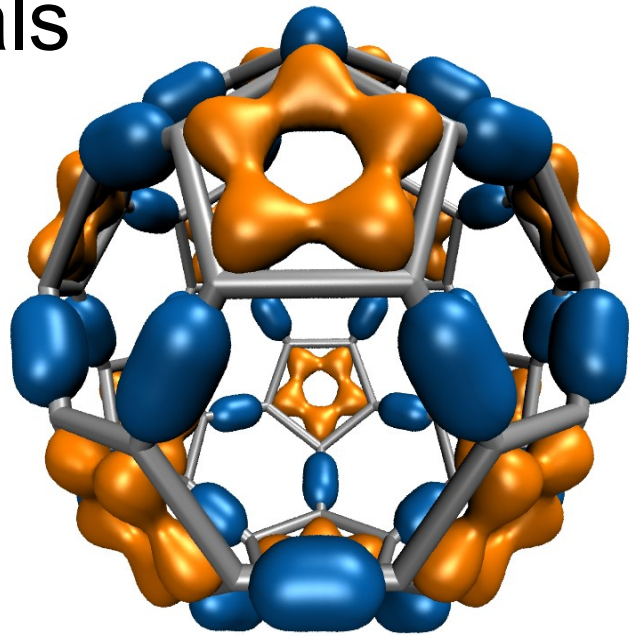
Latest results on Pascal P100, first runs on Volta V100

Hardware platform	Runtime, Speedup vs. Chimera,	VMD+GPU
Chimera Xeon E5-2687W (2 socket) [1]	15.860s, 1x	
VMD-CUDA IBM Power8 + 1x Tesla K40 [2]	0.488s, 32x	**0.9x**
VMD-CUDA Intel Xeon E5-2687W + 1x Quadro K6000 [1,2]	0.458s, 35x	**1.0x**
VMD-CUDA Intel Xeon E5-2698v3 + 1x Tesla P100	**0.090s,** 176x	**5.1x**
VMD-CUDA IBM Power8 "Minsky" + 1x Tesla P100	**0.080s,** 198x	**5.7x**
VMD-CUDA Intel Xeon E5-2697Av4 + 1x Tesla V100	**0.050s,** 317x	**9.2x**

[1] GPU-Accelerated Analysis and Visualization of Large Structures Solved by Molecular Dynamics Flexible Fitting. J. E. Stone, R. McGreevy, B. Isralewitz, and K. Schulten. Faraday Discussions 169:265-283, 2014.
[2] Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms. J. E. Stone, A.-P. Hynninen, J. C. Phillips, K. Schulten. International Workshop on OpenPOWER for HPC (IWOPH'16), LNCS 9945, pp. 188-206, 2016.

Visalization of Molecular Orbitals

- Visualization of MOs aids in understanding the chemistry of molecular system
- MO spatial distribution is correlated with probability density for an electron(s)
- **Animation** of (classical mechanics) molecular dynamics trajectories provides insight into simulation results
  - To do the same for QM or QM/MM simulations MOs must be computed at **10 FPS** or more
- **Run-time code generation** (JIT) and compilation via **CUDA NVRTC** enable further optimizations and the **highest performance to date: 1.8x faster than fully-general data-driven loops**



C_{60}

High Performance Computation and Interactive Display of Molecular Orbitals on GPUs and Multi-core CPUs. J. E. Stone, J. Saam, D. Hardy, K. Vandivort, W. Hwu, K. Schulten, *2nd Workshop on General-Purpose Computation on Graphics Processing Units (GPGPU-2), ACM International Conference Proceeding Series*, volume 383, pp. 9-18, 2009.

MO Kernel for One Grid Point (Naive C)

```
…
for (at=0; at<numatoms; at++) {
   int prim_counter = atom_basis[at];
   calc_distances_to_atom(&atompos[at], &xdist, &ydist, &zdist, &dist2, &xdiv);
   for (contracted_gto=0.0f, shell=0; shell < num_shells_per_atom[at]; shell++) {
      int shell_type = shell_symmetry[shell_counter];
      for (prim=0; prim < num_prim_per_shell[shell_counter];  prim++) {
         float exponent         = basis_array[prim_counter      ];
         float contract_coeff = basis_array[prim_counter + 1];
         contracted_gto += contract_coeff * expf(-exponent*dist2);
         prim_counter += 2;
      }
      for (tmpshell=0.0f, j=0, zdp=1.0f; j<=shell_type; j++, zdp*=zdist) {
         int imax = shell_type - j;
         for (i=0, ydp=1.0f, xdp=pow(xdist, imax); i<=imax; i++, ydp*=ydist, xdp*=xdiv)
            tmpshell += wave_f[ifunc++] * xdp * ydp * zdp;
      }
      value += tmpshell * contracted_gto;
      shell_counter++;
   }
} …..
```

Loop over atoms

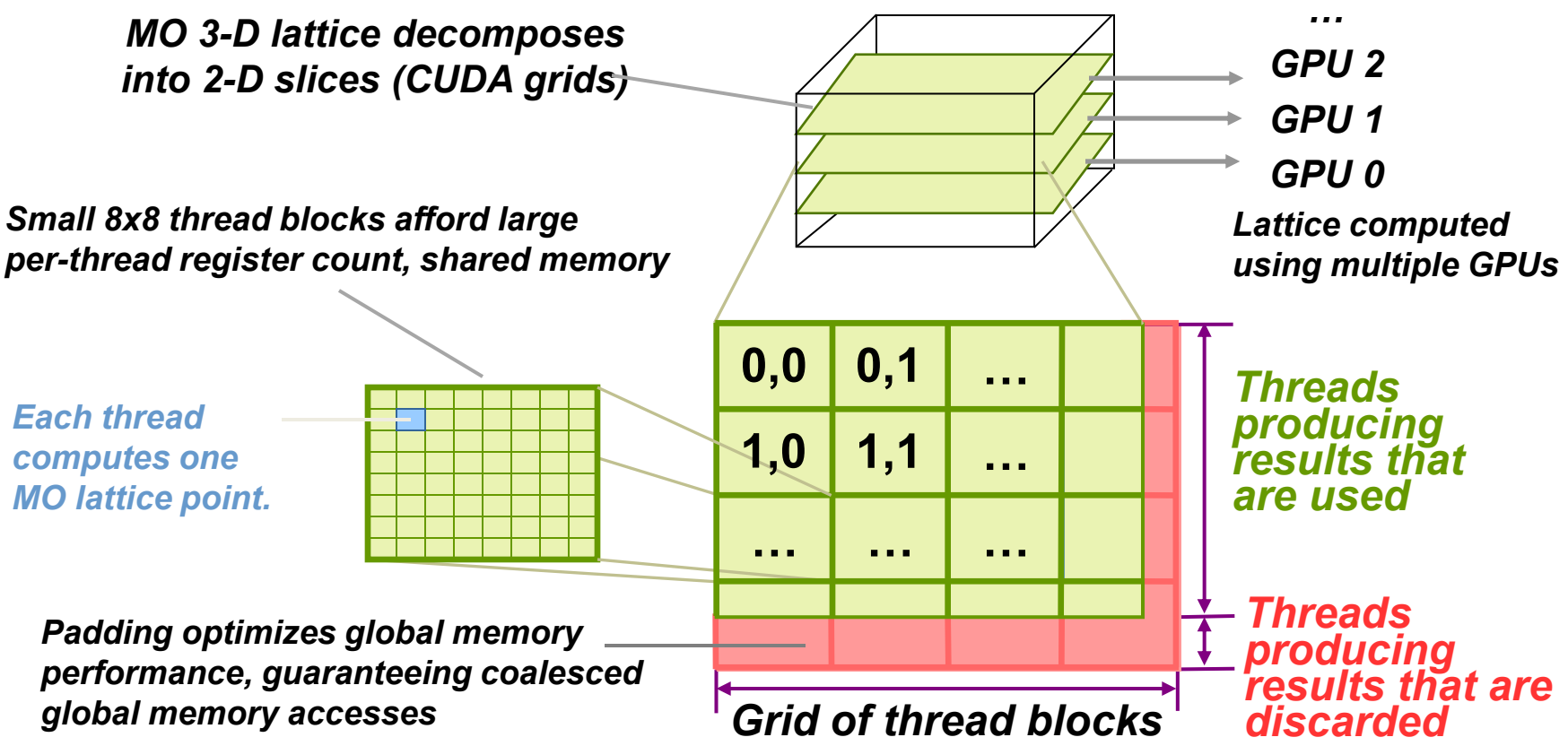
Loop over shells

Loop over primitives: largest component of runtime, due to **expf()**

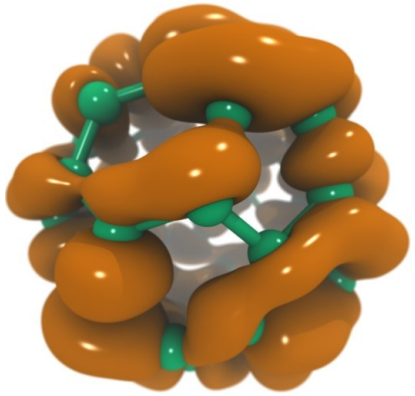
Loop over angular momenta

(unrolled in real code)

MO GPU Parallel Decomposition



VMD Tesla P100 Performance for C_{60} Molecular Orbitals, 516x519x507 grid



Hardware platform	Runtime,	Speedup
IBM Power8 (ORNL 'crest') + 1x Tesla K40 [1]	3.49s,	**1.0x**
Intel Xeon E5-2698v3 + 1x Tesla P100	**1.35s,**	**2.5x**
IBM Power8 "Minsky" + 1x Tesla P100	**1.09s,**	**3.3x**
IBM Power8 (ORNL 'crest') + 4x Tesla K40 [1]	0.91s,	3.8x
Intel Xeon E5-2698v3 + 4x Tesla P100	**0.37s,**	**9.4x**
IBM Power8 "Minsky" + 4x Tesla P100	**0.30s,**	**11.6x**
Intel Xeon E5-2697Av4 + 1x Tesla V100	**0.610s,**	**5.7x**
Intel Xeon E5-2697Av4 + 2x Tesla V100	**0.294s,**	**11.8x**
Intel Xeon E5-2697Av4 + 3x Tesla V100	**0.220s,**	**15.9x**

NVLink perf. boost w/ no code tuning (YET)

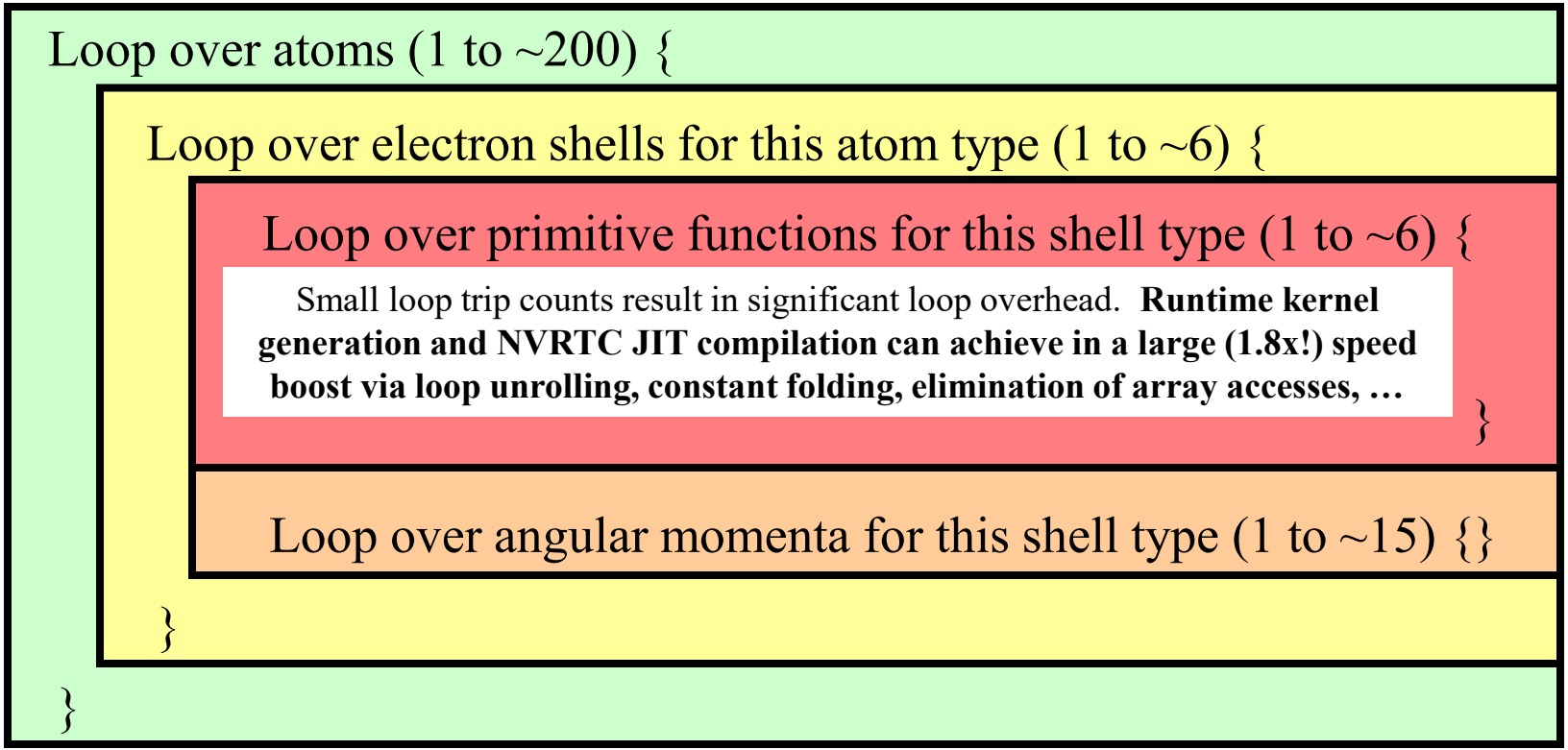
Early Volta V100 PCIe Result!

[1] Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms. J. E. Stone, A.-P. Hynninen, J. C. Phillips, K. Schulten. International Workshop on OpenPOWER for HPC (IWOPH'16), LNCS 9945, pp. 188-206, 2016.

NIH

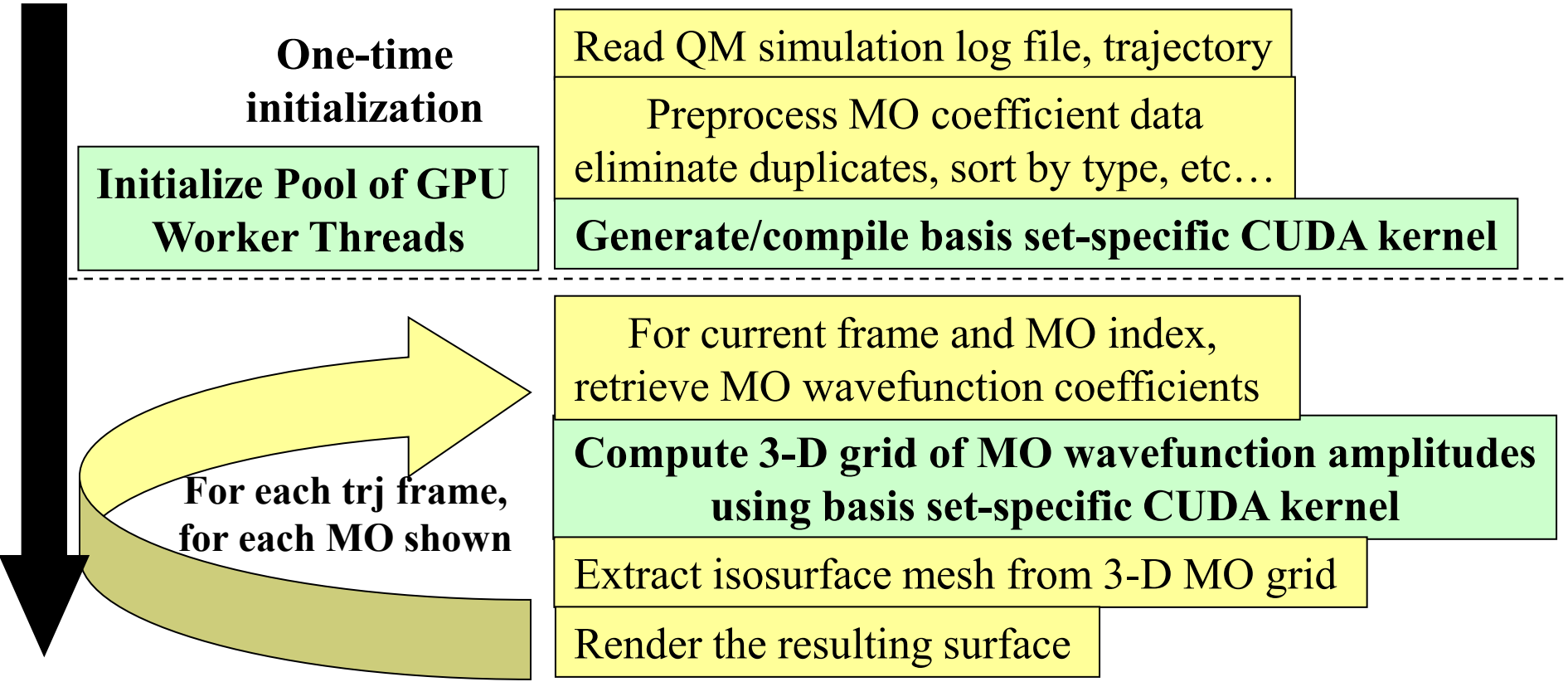
MO Kernel Structure, Opportunity for NRTC JIT…

Data-driven execution, but representative loop trip counts in (…)



Molecular Orbital Computation and Display Process

Runtime Kernel Generation, NVRTC Just-In-Time (JIT) Compilation



```
for (shell=0; shell < maxshell; shell++)  {
  float contracted_gto = 0.0f;

  // Loop over the Gaussian primitives of CGTO
  int maxprim = const_num_prim_per_shell[shell_counter];
  int shell_type = const_shell_symmetry[shell_counter];
  for (prim=0; prim < maxprim;  prim++) {
    float exponent         = const_basis_array[prim_counter     ];
    float contract_coeff = const_basis_array[prim_counter + 1];
    contracted_gto += contract_coeff  * expf(-exponent*dist2);
    prim_counter += 2;
  }
```

General loop-based data-dependent MO CUDA kernel

Runtime-generated data-specific MO CUDA kernel compiled via **CUDA NVRTC** JIT…

```
contracted_gto = 1.832937 * expf(-7.868272*dist2);
contracted_gto += 1.405380 * expf(-1.881289*dist2);
contracted_gto += 0.701383 * expf(-0.544249*dist2);
```

1.8x Faster

```
for (shell=0; shell < maxshell; shell++)  {
 float contracted_gto = 0.0f;

 // Loop over the Gaussian primitives of CGTO
 int maxprim = const_num_prim_per_shell[shell_counter];
 int shell_type = const_shell_symmetry[shell_counter];
 for (prim=0; prim < maxprim;  prim++) {
  float exponent         = const_basis_array[prim_counter     ];
  float contract_coeff = const_basis_array[prim_counter + 1];
  contracted_gto += contract_coeff  * expf(-exponent*dist2);
  prim_counter += 2;
 }

 float tmpshell=0;
 switch (shell_type) {
  case S_SHELL:
   value += const_wave_f[ifunc++] * contracted_gto;
   break;
[…..]
  case D_SHELL:
   tmpshell += const_wave_f[ifunc++] * xdist2;
   tmpshell += const_wave_f[ifunc++] * ydist2;
   tmpshell += const_wave_f[ifunc++] * zdist2;
   tmpshell += const_wave_f[ifunc++] * xdist * ydist;
```

General loop-based data-dependent MO CUDA kernel

Runtime-generated data-specific MO CUDA kernel compiled via **CUDA NVRTC** JIT…

1.8x Faster

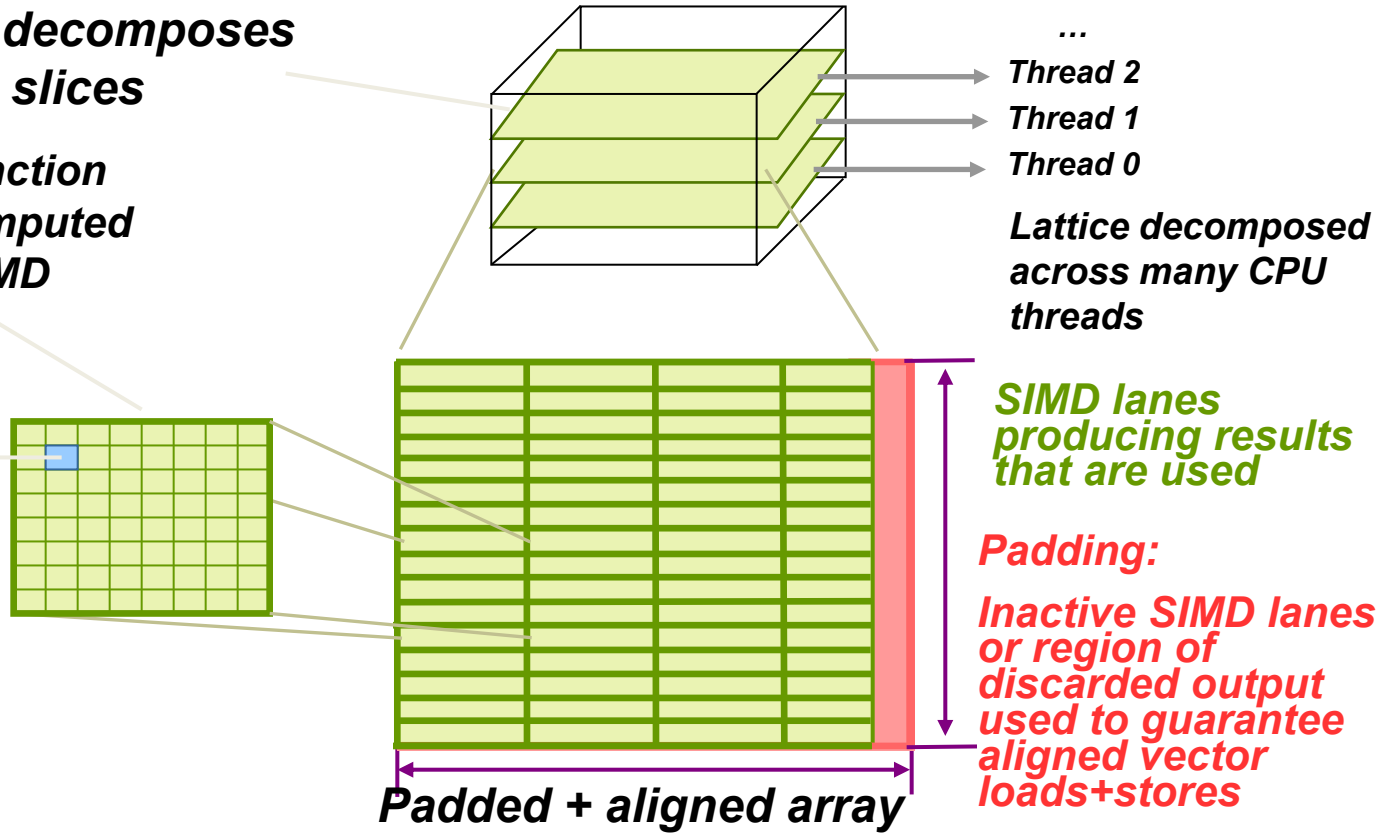
```
contracted_gto = 1.832937 * expf(-7.868272*dist2);
contracted_gto += 1.405380 * expf(-1.881289*dist2);
contracted_gto += 0.701383 * expf(-0.544249*dist2);
// P_SHELL
tmpshell = const_wave_f[ifunc++] * xdist;
tmpshell += const_wave_f[ifunc++] * ydist;
tmpshell += const_wave_f[ifunc++] * zdist;
value += tmpshell * contracted_gto;

contracted_gto = 0.187618 * expf(-0.168714*dist2);
// S_SHELL
value += const_wave_f[ifunc++] * contracted_gto;

contracted_gto = 0.217969 * expf(-0.168714*dist2);
// P_SHELL
tmpshell = const_wave_f[ifunc++] * xdist;
tmpshell += const_wave_f[ifunc++] * ydist;
tmpshell += const_wave_f[ifunc++] * zdist;
value += tmpshell * contracted_gto;

contracted_gto = 3.858403 * expf(-0.800000*dist2);
// D_SHELL
tmpshell = const_wave_f[ifunc++] * xdist2;
tmpshell += const_wave_f[ifunc++] * ydist2;
```

MO CPU Parallel Decomposition
Early AVX-512 Kernels on KNL

MO 3-D lattice decomposes into 2-D slices

Vectors of wavefunction amplitudes are computed using hardware SIMD instructions

Each CPU thread computes 1, 4, 8, 16 MO lattice points per loop iteration: C, SSE, AVX2 or AVX-512ER



AVX-512ER MO CGTO Loop

```
int maxprim = num_prim_per_shell[shell_counter];
int shelltype = shell_types[shell_counter];
for (prim=0; prim<maxprim; prim++) {
    float exponent       = basis_array[prim_counter    ];
    float contract_coeff = basis_array[prim_counter + 1];

    // contracted_gto += contract_coeff * exp(exponent*dist2);
    __m512 expval = _mm512_mul_ps(_mm512_set1_ps(exponent * MLOG2EF), dist2);

    // expf() approximation required, use (base-2) AVX-512ER instructions…
    __m512 retval = _mm512_exp2a23_ps(expval);
```

AVX-512ER MO Wavefunction Loop

```
 /* multiply with the appropriate wavefunction coefficient */
__m512 ts = _mm512_set1_ps(0.0f);
switch (shelltype) {
  case S_SHELL:
    value = _mm512_add_ps(value,  _mm512_mul_ps(_mm512_set1_ps(wave_f[ifunc++]), cgto));
        break;

  case P_SHELL:
    ts = _mm512_add_ps(ts, _mm512_mul_ps(_mm512_set1_ps(wave_f[ifunc++]),  xdist));
    ts = _mm512_add_ps(ts, _mm512_mul_ps(_mm512_set1_ps(wave_f[ifunc++]), ydist));
    ts = _mm512_add_ps(ts, _mm512_mul_ps(_mm512_set1_ps(wave_f[ifunc++]), zdist));
    value = _mm512_add_ps(value, _mm512_mul_ps(ts, cgto));
    break;

  case D_SHELL:
     ….
```

AVX-512ER+FMA MO Wavefunction Loop

```
/* multiply with the appropriate wavefunction coefficient */
__m512 ts = _mm512_set1_ps(0.0f);
switch (shelltype) {
  // use FMADD instructions
  case S_SHELL:
    value = _mm512_fmadd_ps(_mm512_set1_ps(wave_f[ifunc++]), cgto, value);
    break;

  case P_SHELL:
    ts = _mm512_fmadd_ps(_mm512_set1_ps(wave_f[ifunc++]), xdist, ts);
    ts = _mm512_fmadd_ps(_mm512_set1_ps(wave_f[ifunc++]), ydist, ts);
    ts = _mm512_fmadd_ps(_mm512_set1_ps(wave_f[ifunc++]), zdist, ts);
    value = _mm512_fmadd_ps(ts, cgto, value);
    break;
```

Performance of AVX-512ER Instrinsics vs. Autovectorization on KNL: Small 172x173x169 Grid

- Intel C++ ‘15 autovectorization **(fail)**: **220+ sec**
- Hand-coded SSE2 w/ existing thread scheme: **48.5 sec**
- Hand-coded AVX-512ER w/ existing thread scheme: **6.3 sec**
- Hand-coded AVX-512ER, refactoring thread pool: **0.2 sec**
- **Hand-coded AVX-512ER tuned thread pool: 0.131 sec**
- **Hand-coded AVX-512ER+FMA tweaks: 0.107 sec**

Further improvement will require attention to details of cache behaviour and further tuning of low-level threading constructs for Xeon Phi/KNL

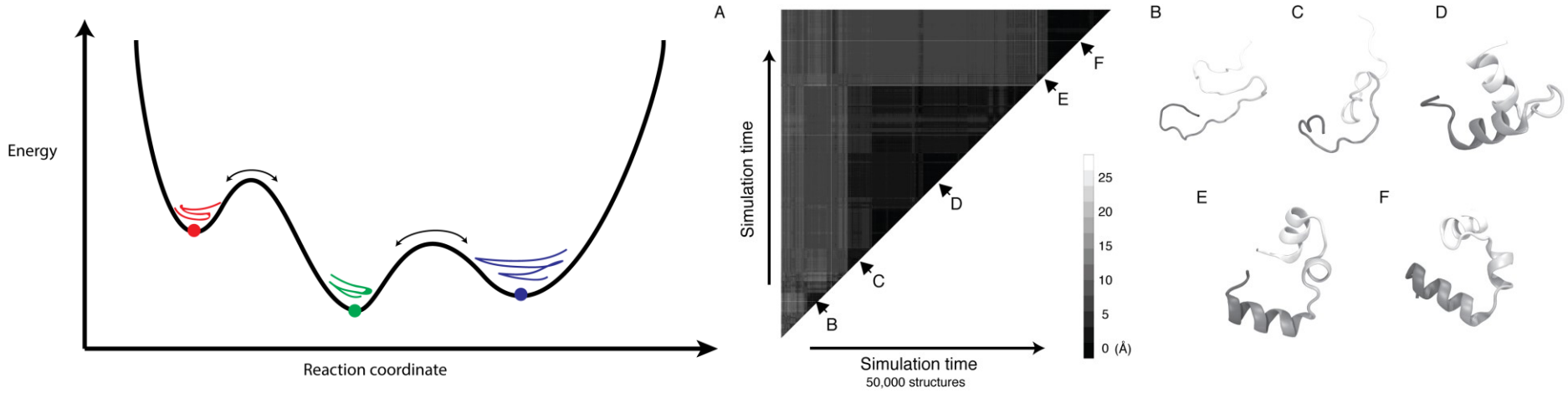
Challenges Adapting Large Software Systems for State-of-the-Art Hardware Platforms

- Initial focus on key computational kernels eventually gives way to the need to optimize an **ocean of less critical routines**, due to observance of Amdahl's Law
- Even though these less critical routines might be easily ported to CUDA or similar, the sheer number of routines often poses a challenge
- Need a low-cost approach for **getting "some" speedup** out of these second-tier routines
- In many cases, it is completely **sufficient to achieve memory-bandwidth-bound GPU performance with an existing algorithm**

Directive-Based Parallel Programming with OpenACC

- Annotate loop nests in existing code with #pragma compiler directives:
  - Annotate opportunities for parallelism
  - Annotate points where host-GPU memory transfers are best performed, indicate propagation of data
- Evolve original code structure to improve efficacy of parallelization
  - Eliminate false dependencies between loop iterations
  - Revise algorithms or constructs that create excess data movement

Clustering Analysis of Molecular Dynamics Trajectories



GPU-Accelerated Molecular Dynamics Clustering Analysis with OpenACC. J.E. Stone, J.R. Perilla, C. K. Cassidy, and K. Schulten. In, Robert Farber, ed., Parallel Programming with OpenACC, Morgan Kaufmann, Chapter 11, pp. 215-240, 2016.

Serial QCP RMSD Inner Product Loop

- Simple example where directive based parallelism can be applied easily and effectively
- Such a loop is inherently a memory-bandwidth-bound algorithm, so that's the goal for acceleration

```
for (int l=0; l<cnt; l++) {
  double x1, x2, y1, y2, z1, z2;
  x1 = crdx1[l];
  y1 = crdy1[l];
  z1 = crdz1[l];

  G1 += x1*x1 + y1*y1 + z1*z1;

  x2 = crdx2[l];
  y2 = crdy2[l];
  z2 = crdz2[l];

  G2 += x2*x2 + y2*y2 + z2*z2;

  a0 += x1 * x2;
  a1 += x1 * y2;
  a2 += x1 * z2;

  a3 += y1 * x2;
  a4 += y1 * y2;
  a5 += y1 * z2;

  a6 += z1 * x2;
  a7 += z1 * y2;
  a8 += z1 * z2;
}
```

OpenACC QCP RMSD Inner Product Loop

- Simple example where directive based parallelism can be applied easily and effectively
- Such a loop is inherently a memory-bandwidth-bound algorithm, so that's the goal for acceleration

```
// excerpted code that has been abridged for brevity…
void rmsdmat_qcp_acc(int cnt, int padcnt, int framecrdsz,
                                int framecount, const float * restrict crds,
long i, j, k;
#pragma acc kernels copyin(crds[0:tsz]), copy(rmsdmat[0:msz])
  for (k=0; k<(framecount*(framecount-1))/2; k++) {
    // compute triangular matrix index 'k' in a helper function
    // to ensure that the compiler doesn't think that we have
    // conflicts or dependencies between loop iterations
    acc_idx2sub_tril(long(framecount-1), k, &i, &j);
    long x1addr = j * 3L * framecrdsz;
    long x2addr = i * 3L * framecrdsz;

#pragma acc loop vector(256)
    for (long l=0; l<cnt; l++) {
    // abridged for brevity ...

    rmsdmat[k]=rmsd; // store linearized triangular matrix
  }
}
```

OpenACC QCP RMSD Inner Product Loop Performance Results

- Xeon 2867W v3, w/ hand-coded AVX and FMA intrinsics: 20.7s
- Tesla K80 w/ OpenACC: **6.5s (3.2x speedup)**
- OpenACC on K80 achieved 65% of theoretical peak memory bandwidth, with 2016 compiler and just a few lines of #pragma directives. Excellent speedup for minimal changes to code.
- Future OpenACC compiler revs should provide higher performance yet

VMD on IBM OpenPOWER

- VMD has been running on various POWER hardware since 1998!
- Now runs on POWER8 w/ Linux in little-endian mode:
  - VMD 1.9.3 release on Nov 2016 includes first OpenPOWER release
  - Src supports CUDA 7.x [1], and **CUDA 8.x w/ P100 and NVLink**
  - Uses P8 VSX instructions for hand-coded and vectorized kernels [1]
- In-progress VMD 1.9.4 development:
  - **VMD supports full OpenGL via GLX and EGL on POWER now!**
    - Latest public NVIDIA driver version 375.66
  - Ongoing improvements to CPU/GPU work scheduling, **NUMA** optimality, and use of new **NVLink** interconnect on latest **IBM "Minsky"** hardware

[1] Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms. J. E. Stone, A.-P. Hynninen, J. C. Phillips, K. Schulten. International Workshop on OpenPOWER for HPC (IWOPH'16), LNCS 9945, pp. 188-206, 2016.

Ongoing VMD CUDA Work on POWER8

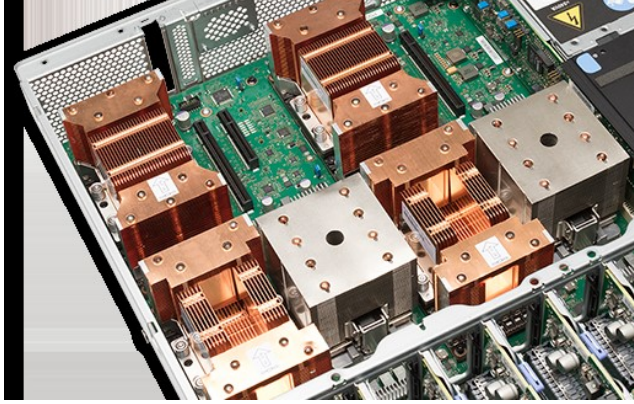
- CUDA kernels all run correctly on previous-gen PCIe Tesla K40s, and new Tesla P100 w/ NVLink
- Early observations about P8+CUDA+NVLink so far:
  - P8 single-thread perf more of an issue than on x86 for small untuned parts of existing code
  - **P8+CUDA NUMA-correctness w/ NVLink much more important** than PCIe (e.g. x86) due to **larger benefits/penalties** when NVLink is used effectively vs. not
  - **P8 "Minsky"** systems get extra benefits for algorithms that have lots of host-GPU DMA transfers, where the NVLink interconnect speeds greatly outpeform PCIe

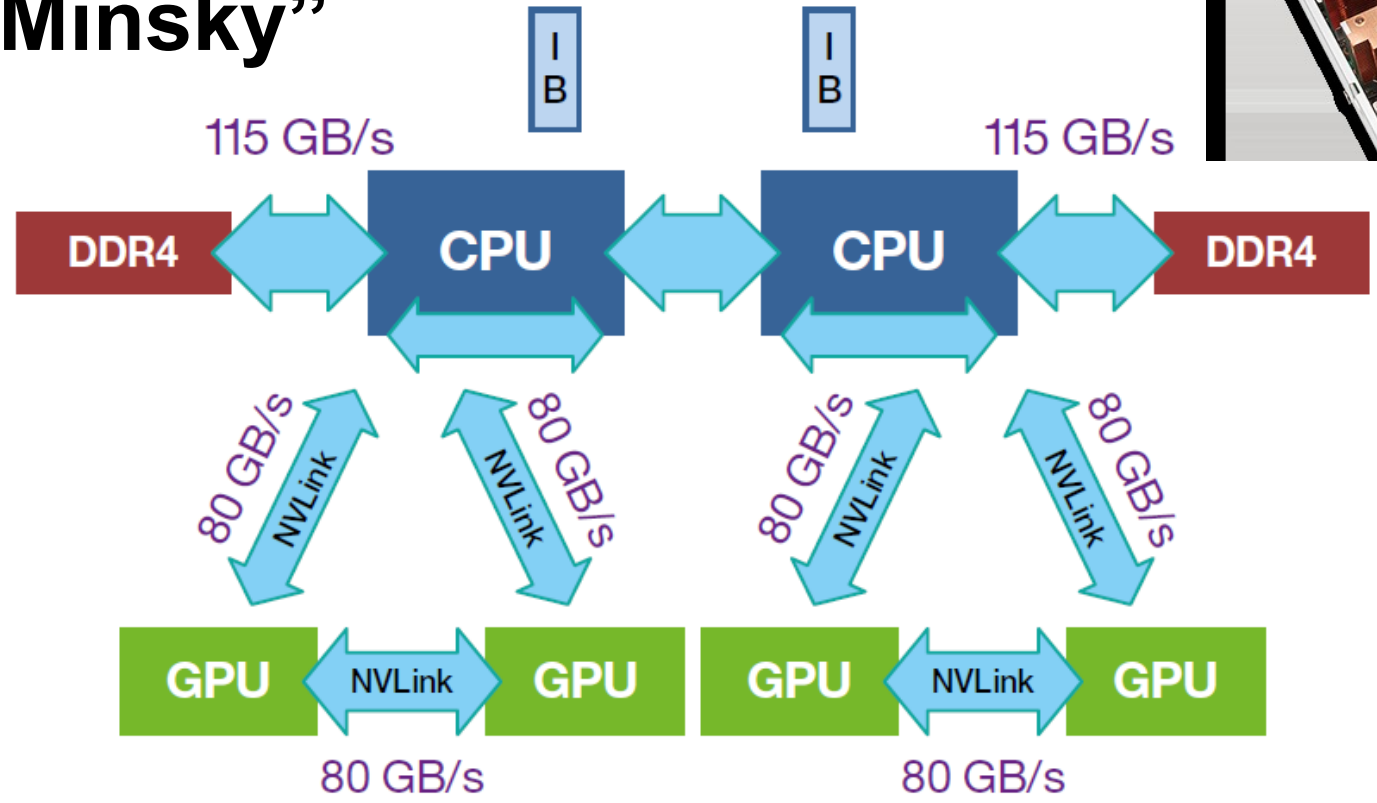
Benefits of P8+NVLink for VMD

- Rapid access to host-side data too large to fit entirely in P100 GPU memory
  - Many existing VMD CUDA kernels already used this strategy w/ PCIe, performance gains from NVLink are large and immediate
- Rapid peer-to-peer GPU data transfers:
  - Bypass host whenever possible, perform nearest-neighbor exchanges for pairwise calculations, e.g. those that arise in algorithms for simulation trajectory clustering
  - Use aggregate GPU memory to collectively store/cache large data – well suited for high-fidelity ray tracing of scenes containing massive amounts of geometry

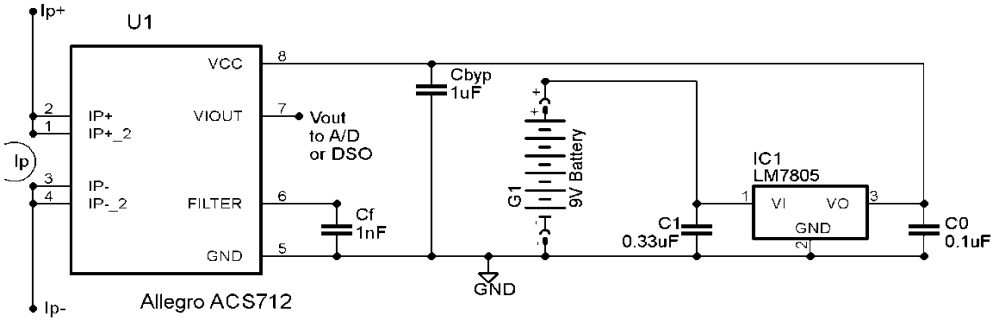
IBM S822LC w/ NVLink

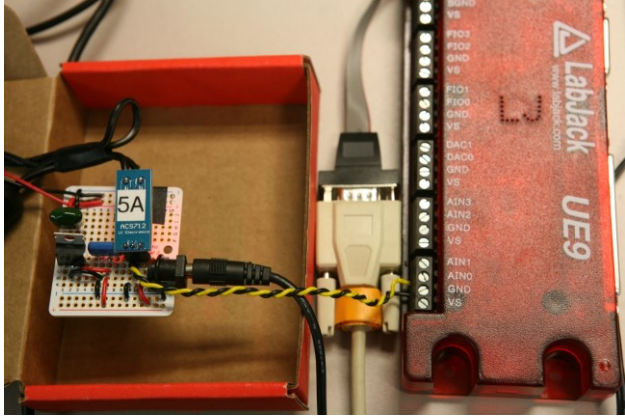
“Minsky”

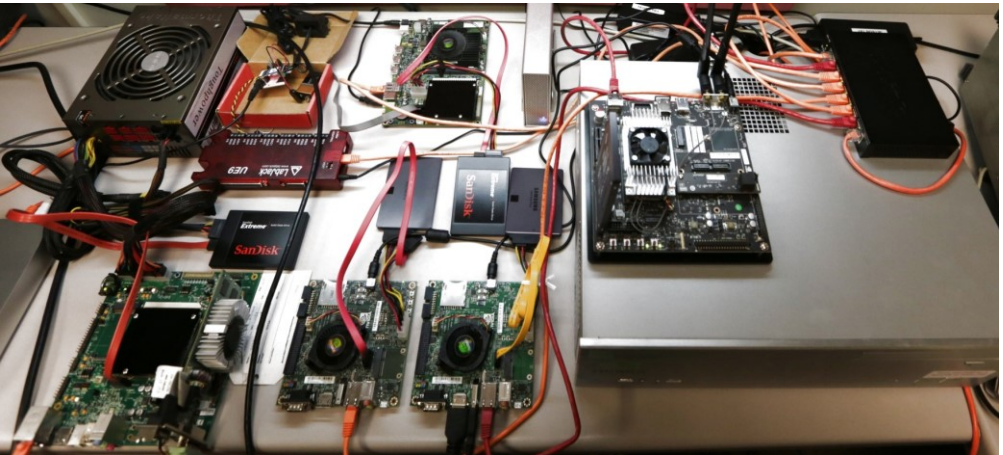


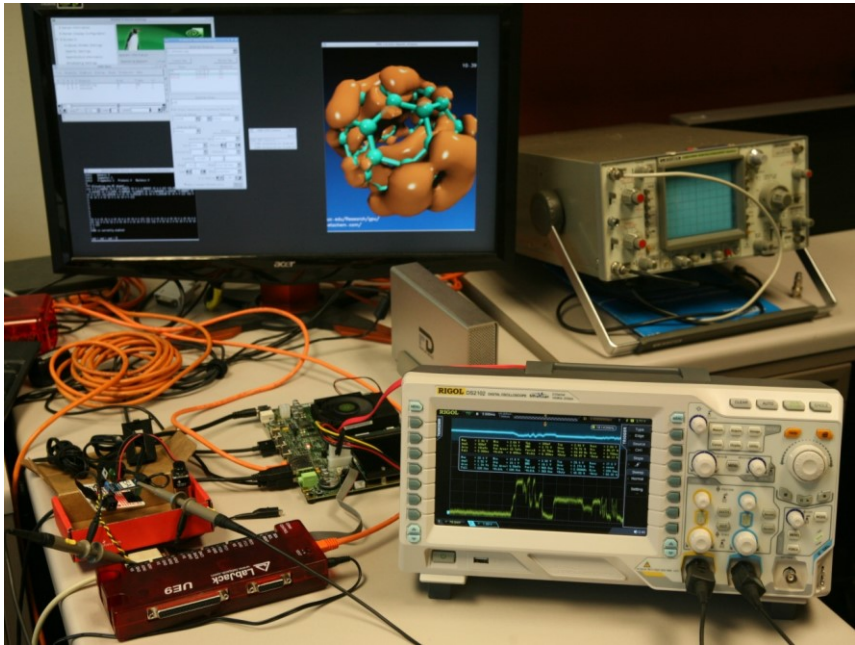


Evaluation of Emerging Energy-Efficient Heterogeneous Computing Platforms for Biomolecular and Cellular Simulation Workloads. Stone et al., 25th International Heterogeneity in Computing Workshop, pp. 89-100, 2016.





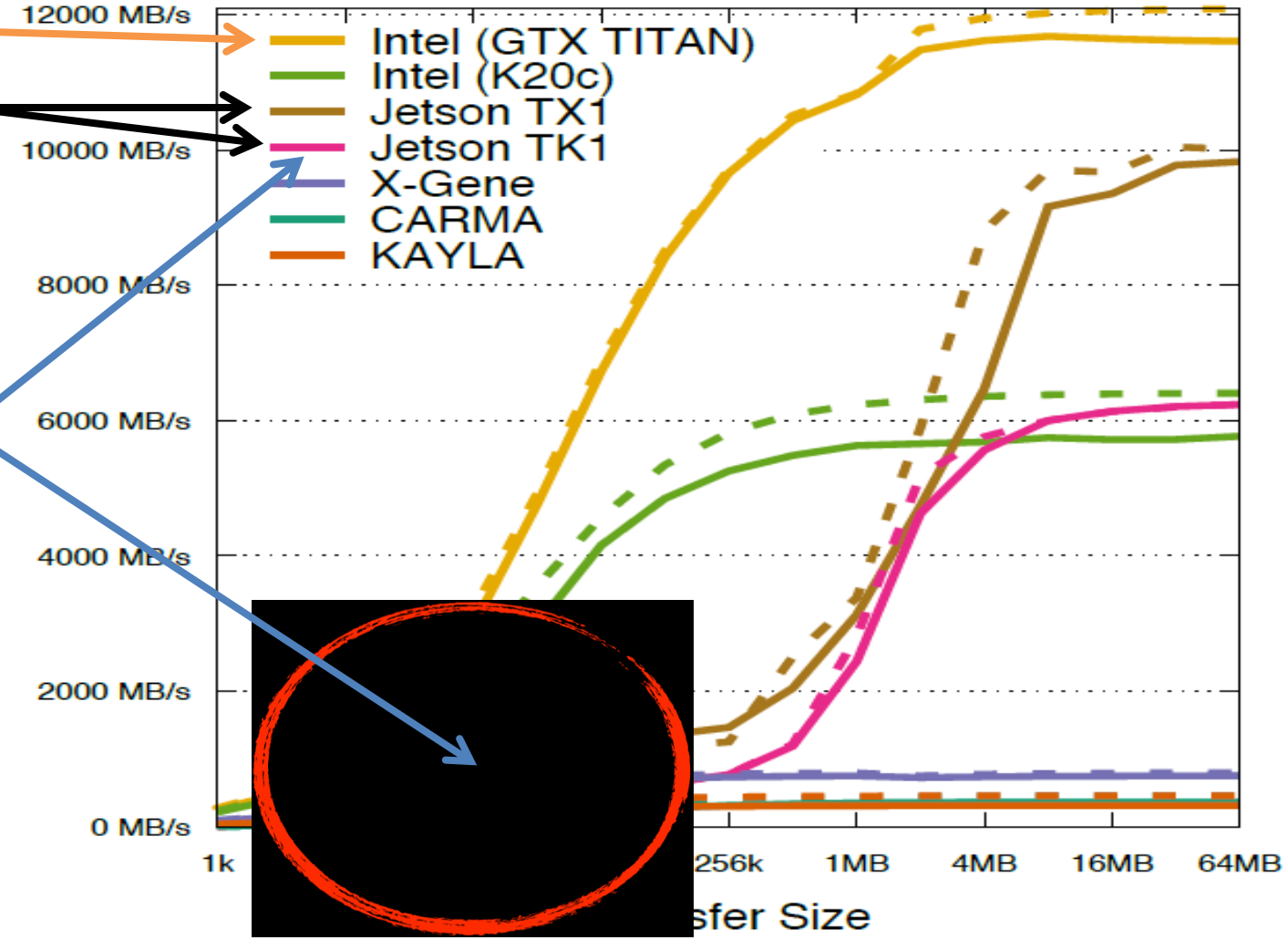




i7-3960X+Titan
fastest x86 platform
ARM +
integrated GPU

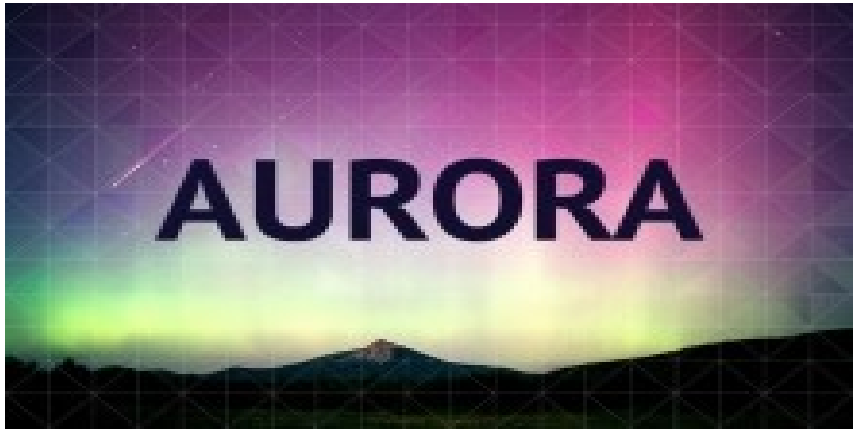
X-Gene+GPU:
fastest ARM
platform

Note comparative
performance for
transfers ranging
from 2KB to 16KB



Future Work

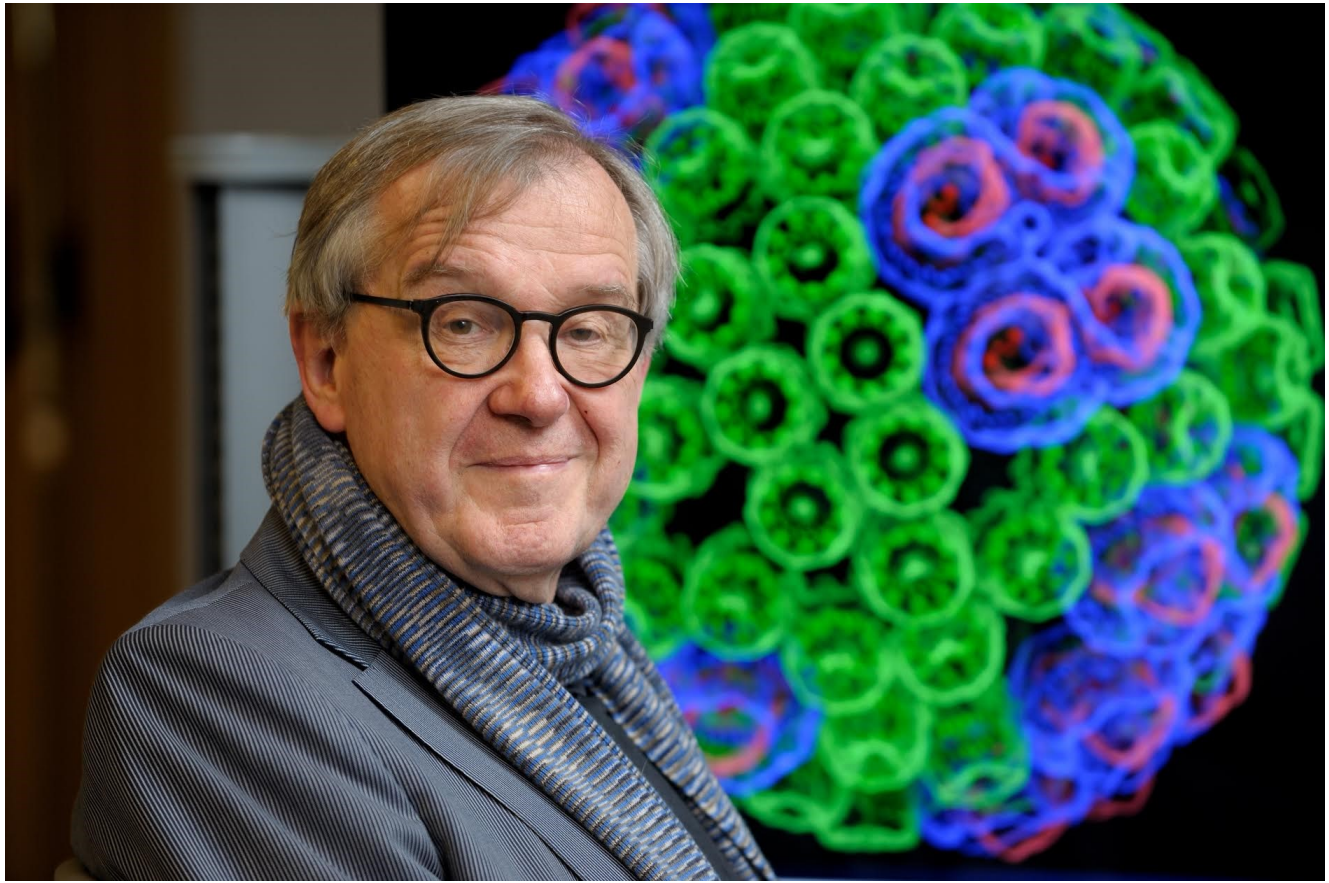
- New analysis features
- Support emerging platforms
- Many more CUDA GPU and AVX-512 CPU kernels…
- Interactive ray tracing of time-varying molecular geometry
- Vulkan graphics API



https://www.khronos.org/vulkan/

Acknowledgements

- Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign
- CUDA Center of Excellence, University of Illinois at Urbana-Champaign
- NVIDIA CUDA and OptiX teams
- Intel OSPRay team
- Funding:
  - DOE INCITE, ORNL Titan: DE-AC05-00OR22725
  - NSF Blue Waters:
    NSF OCI 07-25070, PRAC “The Computational Microscope”,
    ACI-1238993, ACI-1440026
  - NIH support: 9P41GM104601, 5R01GM098243-02



“When I was a young man, my goal was to look with mathematical and computational means at the inside of cells, one atom at a time, to decipher how living systems work. That is what I strived for and I never deflected from this goal.” – Klaus Schulten

Related Publications

http://www.ks.uiuc.edu/Research/gpu/

- **Challenges of Integrating Stochastic Dynamics and Cryo-electron Tomograms in Whole-cell Simulations.** T. M. Earnest, R. Watanabe, J. E. Stone, J. Mahamid, W. Baumeister, E. Villa, and Z. Luthey-Schulten. J. Physical Chemistry B, 121(15): 3871-3881, 2017.
- **Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms.** J. E. Stone, A.-P. Hynninen, J. C. Phillips, and K. Schulten. International Workshop on OpenPOWER for HPC (IWOPH'16), LNCS 9945, pp. 188-206, 2016.
- **Immersive Molecular Visualization with Omnidirectional Stereoscopic Ray Tracing and Remote Rendering.** J. E. Stone, W. R. Sherman, and K. Schulten. High Performance Data Analysis and Visualization Workshop, IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW), pp. 1048-1057, 2016.
- **High Performance Molecular Visualization: In-Situ and Parallel Rendering with EGL.** J. E. Stone, P. Messmer, R. Sisneros, and K. Schulten. High Performance Data Analysis and Visualization Workshop, IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW), pp. 1014-1023, 2016.
- **Evaluation of Emerging Energy-Efficient Heterogeneous Computing Platforms for Biomolecular and Cellular Simulation Workloads.** J. E. Stone, M. J. Hallock, J. C. Phillips, J. R. Peterson, Z. Luthey-Schulten, and K. Schulten.25th International Heterogeneity in Computing Workshop, IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW), pp. 89-100, 2016.
- **Atomic Detail Visualization of Photosynthetic Membranes with GPU-Accelerated Ray Tracing.** J. E. Stone, M. Sener, K. L. Vandivort, A. Barragan, A. Singharoy, I. Teo, J. V. Ribeiro, B. Isralewitz, B. Liu, B.-C. Goh, J. C. Phillips, C. MacGregor-Chatwin, M. P. Johnson, L. F. Kourkoutis, C. Neil Hunter, and K. Schulten. J. Parallel Computing, 55:17-27, 2016.

Related Publications

http://www.ks.uiuc.edu/Research/gpu/

- **Chemical Visualization of Human Pathogens: the Retroviral Capsids.** Juan R. Perilla, Boon Chong Goh, John E. Stone, and Klaus Schulten. SC'15 Visualization and Data Analytics Showcase, 2015.
- **Visualization of Energy Conversion Processes in a Light Harvesting Organelle at Atomic Detail.** M. Sener, J. E. Stone, A. Barragan, A. Singharoy, I. Teo, K. L. Vandivort, B. Isralewitz, B. Liu, B. Goh, J. C. Phillips, L. F. Kourkoutis, C. N. Hunter, and K. Schulten. SC'14 Visualization and Data Analytics Showcase, 2014.
  ***Winner of the SC'14 Visualization and Data Analytics Showcase**
- **Runtime and Architecture Support for Efficient Data Exchange in Multi-Accelerator Applications.** J. Cabezas, I. Gelado, J. E. Stone, N. Navarro, D. B. Kirk, and W. Hwu. IEEE Transactions on Parallel and Distributed Systems, 26(5):1405-1418, 2015.
- **Unlocking the Full Potential of the Cray XK7 Accelerator.** M. D. Klein and J. E. Stone. Cray Users Group, Lugano Switzerland, May 2014.
- **GPU-Accelerated Analysis and Visualization of Large Structures Solved by Molecular Dynamics Flexible Fitting.** J. E. Stone, R. McGreevy, B. Isralewitz, and K. Schulten. Faraday Discussions, 169:265-283, 2014.
- **Simulation of reaction diffusion processes over biologically relevant size and time scales using multi-GPU workstations**. M. J. Hallock, J. E. Stone, E. Roberts, C. Fry, and Z. Luthey-Schulten. Journal of Parallel Computing, 40:86-99, 2014.

Related Publications

http://www.ks.uiuc.edu/Research/gpu/

- **GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms.** J. Stone, K. L. Vandivort, and K. Schulten. UltraVis'13: Proceedings of the 8th International Workshop on Ultrascale Visualization, pp. 6:1-6:8, 2013.
- **Early Experiences Scaling VMD Molecular Visualization and Analysis Jobs on Blue Waters.** J. Stone, B. Isralewitz, and K. Schulten. In proceedings, Extreme Scaling Workshop, 2013.
- **Lattice Microbes: High-performance stochastic simulation method for the reaction-diffusion master equation.** E. Roberts, J. Stone, and Z. Luthey-Schulten. J. Computational Chemistry 34 (3), 245-255, 2013**.**
- **Fast Visualization of Gaussian Density Surfaces for Molecular Dynamics and Particle System Trajectories.** M. Krone, J. Stone, T. Ertl, and K. Schulten. *EuroVis Short Papers,* pp. 67-71, 2012.
- **Immersive Out-of-Core Visualization of Large-Size and Long-Timescale Molecular Dynamics Trajectories.** J. Stone, K. L. Vandivort, and K. Schulten. G. Bebis et al. (Eds.): *7th International Symposium on Visual Computing (ISVC 2011)*, LNCS 6939, pp. 1-12, 2011.
- **Fast Analysis of Molecular Dynamics Trajectories with Graphics Processing Units – Radial Distribution Functions.** B. Levine, J. Stone, and A. Kohlmeyer. *J. Comp. Physics*, 230(9):3556-3569, 2011.

Related Publications

http://www.ks.uiuc.edu/Research/gpu/

- **Quantifying the Impact of GPUs on Performance and Energy Efficiency in HPC Clusters.** J. Enos, C. Steffen, J. Fullop, M. Showerman, G. Shi, K. Esler, V. Kindratenko, J. Stone, J Phillips. *International Conference on Green Computing,* pp. 317-324, 2010.
- **GPU-accelerated molecular modeling coming of age.** J. Stone, D. Hardy, I. Ufimtsev, K. Schulten. *J. Molecular Graphics and Modeling,* 29:116-125, 2010.
- **OpenCL: A Parallel Programming Standard for Heterogeneous Computing.** J. Stone, D. Gohara, G. Shi. *Computing in Science and Engineering,* 12(3):66-73, 2010.
- **An Asymmetric Distributed Shared Memory Model for Heterogeneous Computing Systems**. I. Gelado, J. Stone, J. Cabezas, S. Patel, N. Navarro, W. Hwu. *ASPLOS '10: Proceedings of the 15th International Conference on Architectural Support for Programming Languages and Operating Systems,* pp. 347-358, 2010.

Related Publications

http://www.ks.uiuc.edu/Research/gpu/

- **GPU Clusters for High Performance Computing**. V. Kindratenko, J. Enos, G. Shi, M. Showerman, G. Arnold, J. Stone, J. Phillips, W. Hwu. *Workshop on Parallel Programming on Accelerator Clusters (PPAC),* In Proceedings IEEE Cluster 2009, pp. 1-8, Aug. 2009.
- **Long time-scale simulations of in vivo diffusion using GPU hardware**. E. Roberts, J. Stone, L. Sepulveda, W. Hwu, Z. Luthey-Schulten. In *IPDPS'09: Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Computing*, pp. 1-8, 2009.
- **High Performance Computation and Interactive Display of Molecular Orbitals on GPUs and Multi-core CPUs**. J. E. Stone, J. Saam, D. Hardy, K. Vandivort, W. Hwu, K. Schulten, *2nd Workshop on General-Purpose Computation on Graphics Pricessing Units (GPGPU-2), ACM International Conference Proceeding Series*, volume 383, pp. 9-18, 2009.
- **Probing Biomolecular Machines with Graphics Processors**. J. Phillips, J. Stone. *Communications of the ACM,* 52(10):34-41, 2009.
- **Multilevel summation of electrostatic potentials using graphics processing units**. D. Hardy, J. Stone, K. Schulten. *J. Parallel Computing*, 35:164-177, 2009.

Related Publications

http://www.ks.uiuc.edu/Research/gpu/

- **Adapting a message-driven parallel application to GPU-accelerated clusters**. J. Phillips, J. Stone, K. Schulten. *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, IEEE Press, 2008.
- **GPU acceleration of cutoff pair potentials for molecular modeling applications**. C. Rodrigues, D. Hardy, J. Stone, K. Schulten, and W. Hwu. *Proceedings of the 2008 Conference On Computing Frontiers*, pp. 273-282, 2008.
- **GPU computing**. J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, J. Phillips. *Proceedings of the IEEE*, 96:879-899, 2008.
- **Accelerating molecular modeling applications with graphics processors**. J. Stone, J. Phillips, P. Freddolino, D. Hardy, L. Trabuco, K. Schulten. *J. Comp. Chem.*, 28:2618-2640, 2007.
- **Continuous fluorescence microphotolysis and correlation spectroscopy**. A. Arkhipov, J. Hüve, M. Kahms, R. Peters, K. Schulten. *Biophysical Journal*, 93:4006-4017, 2007.