High Performance Molecular Visualization: In-Situ and Parallel Rendering with EGL

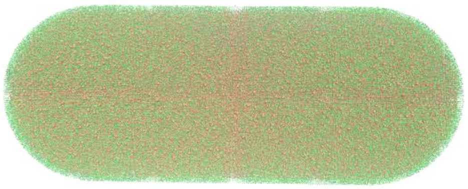
John E. Stone, Peter Messmer, Robert Sisneros, Klaus Schulten

Theoretical and Computational Biophysics Group
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
http://www.ks.uiuc.edu/

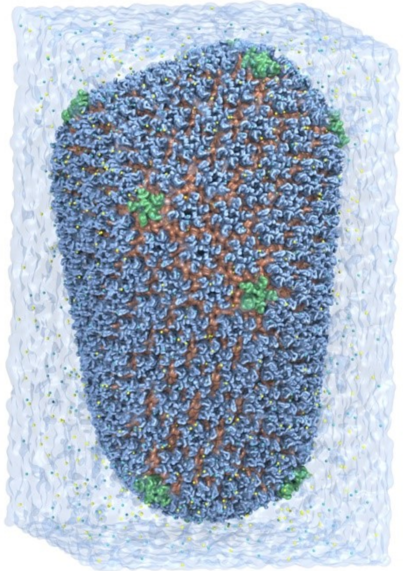
High Performace Data Analysis and Visualization Workshop
IEEE International Symposium on Parallel and Distributed Processing
Chicago, IL, May 23, 2016

VMD – “Visual Molecular Dynamics”

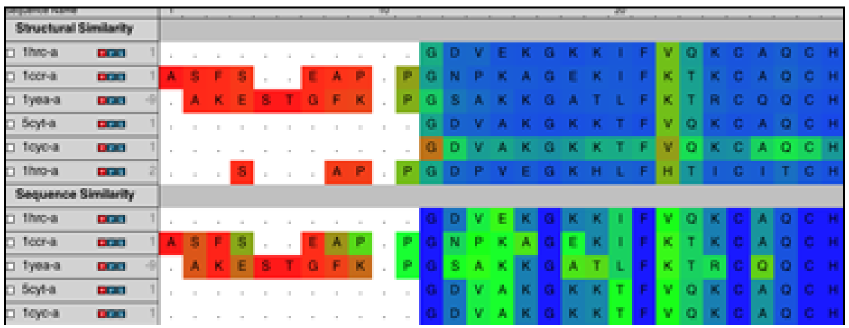
- Visualization and analysis of:
  - molecular dynamics simulations
  - particle systems and whole cells
  - cryoEM densities, volumetric data
  - quantum chemistry calculations
  - sequence information
- User extensible w/ scripting, plugins
- http://www.ks.uiuc.edu/Research/vmd/



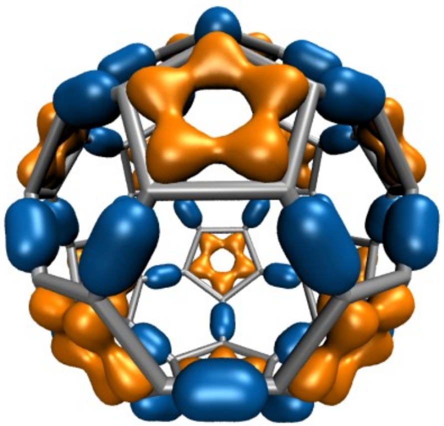
Whole Cell Simulation



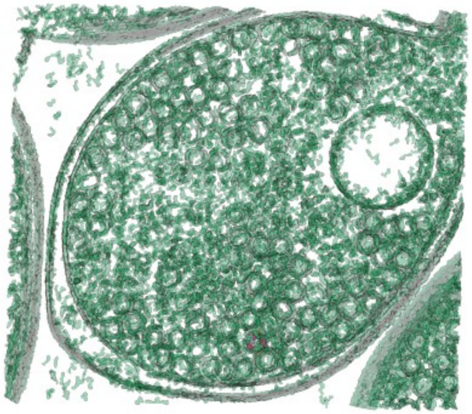
MD Simulations



Sequence Data



Quantum Chemistry

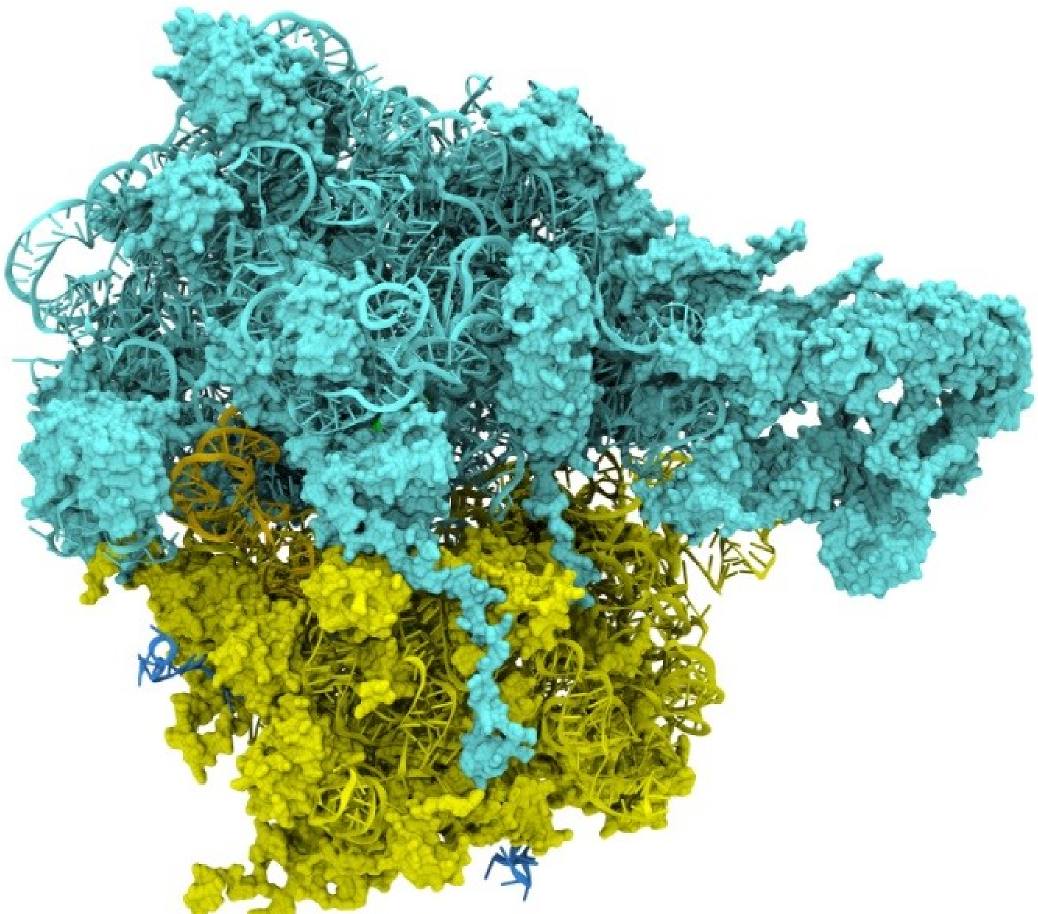


CryoEM, Cellular Tomography

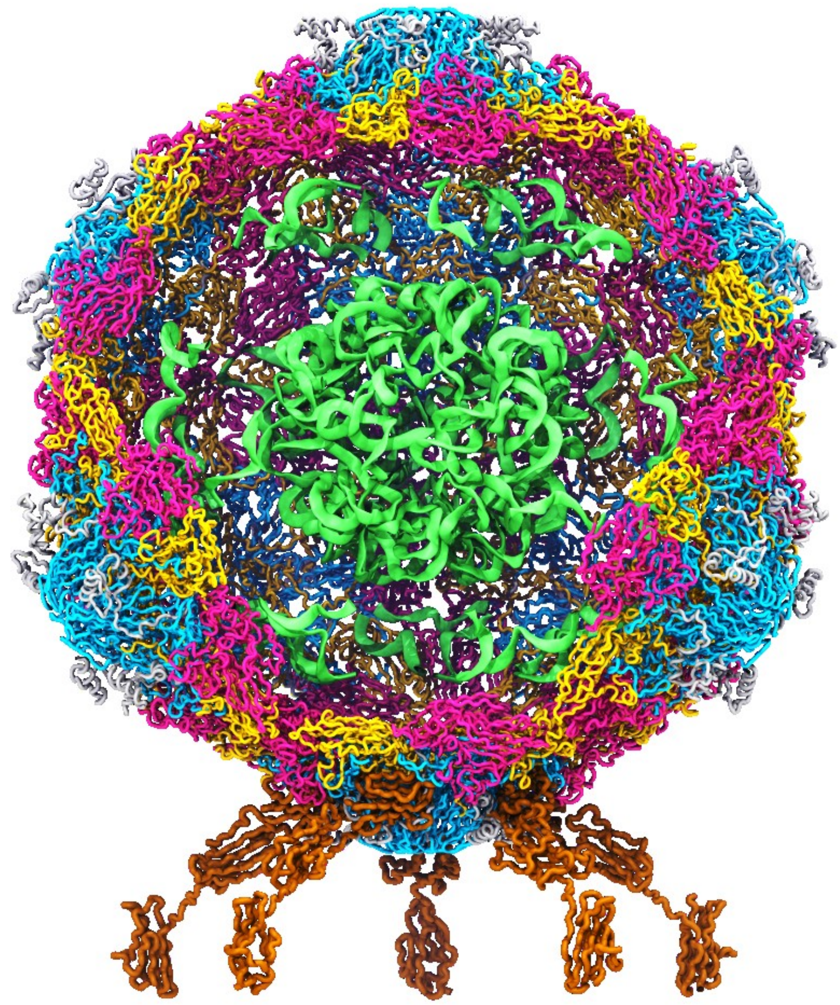
Goal: A Computational Microscope

Study the molecular machines in living cells

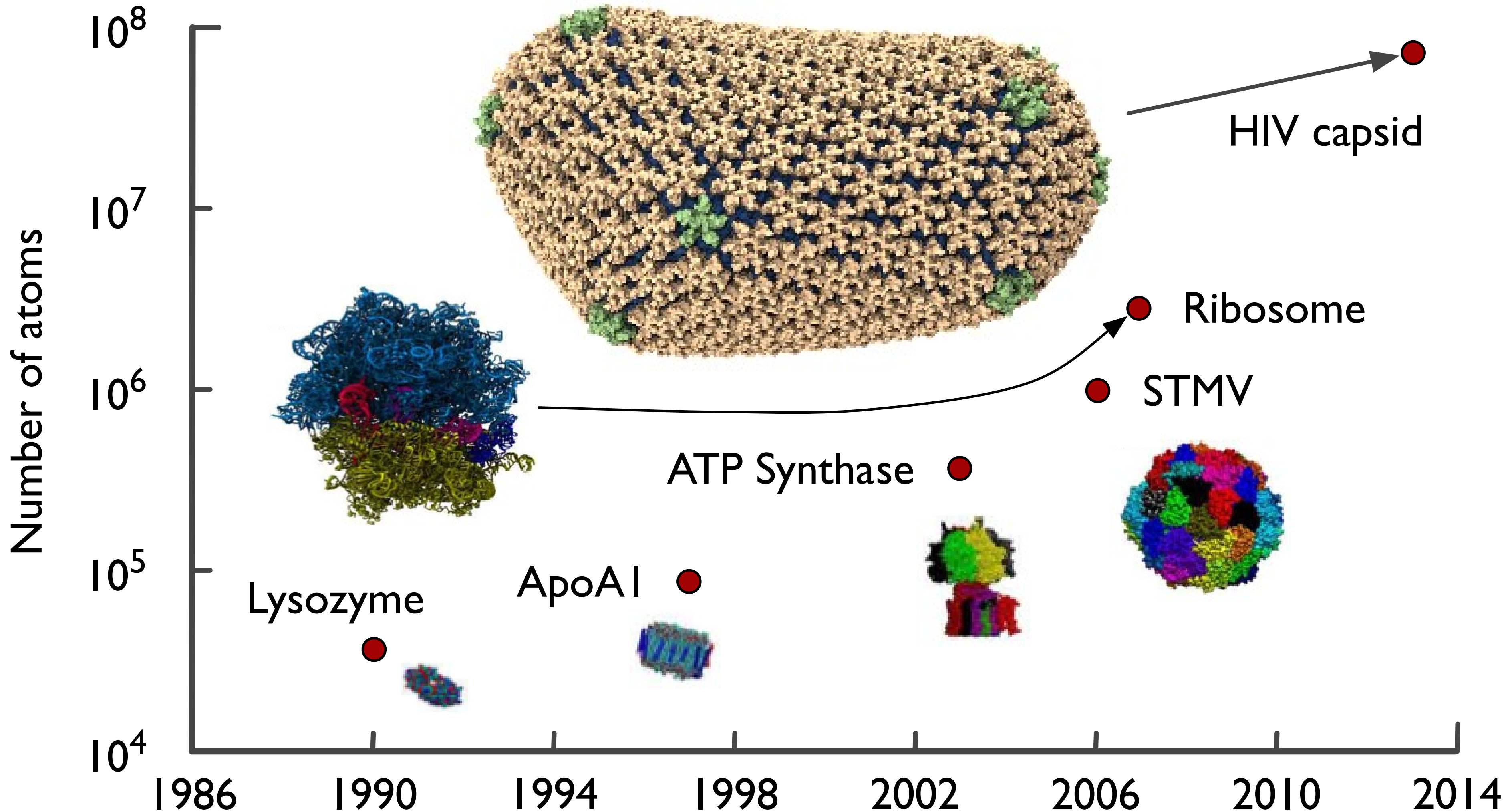
Ribosome: target for antibiotics



Poliovirus

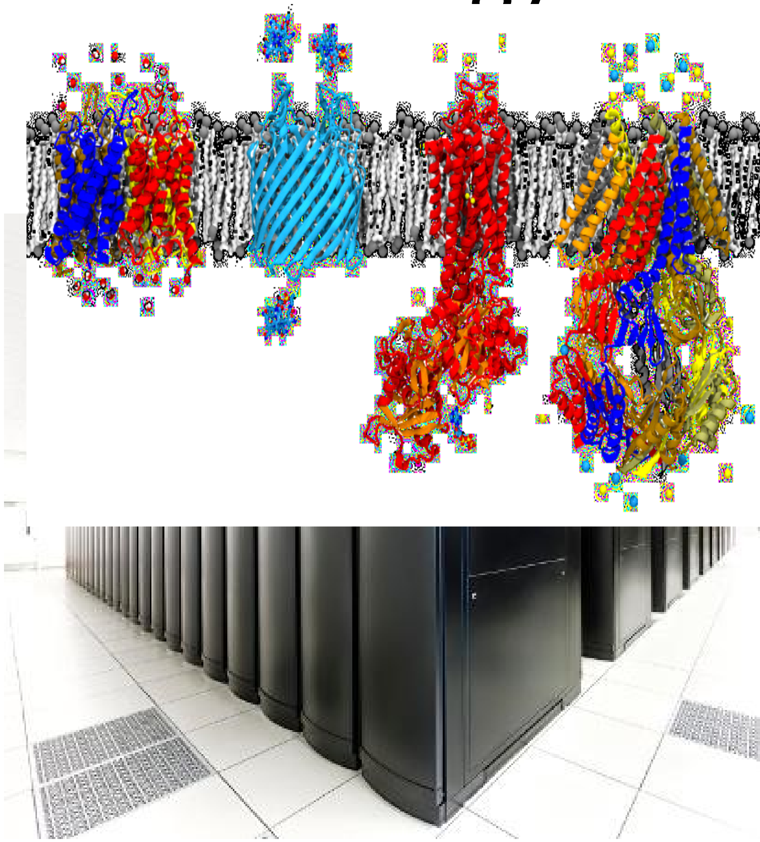


Computational Biology's Insatiable Demand for Processing Power



VMD Supports Petascale Biology

- Where to analyze the data?
  - Trajectories too large to download
  - **Analyze 231 TB trajectory set in 15 min**, parallel I/O @ **275 GB/sec** on 8,192 nodes
- **Use petascale system compute nodes for all simulation, visualization, and analysis tasks**
- GLX requirement for a running windowing system has been an impediment for use of OpenGL rasterization on large supercomputers
- Few petascale systems support windowing systems on compute nodes, **not supported by system vendors**, so sites must implement themselves
- Blue Waters has mix of node types, what about viz and analysis jobs that span node types?
- **EGL helps solve these problems**



NCSA Blue Waters
Cray XE6 / XK7 Supercomputer
22,640 XE6 CPU nodes
4,224 XK7 nodes w/ GPUs

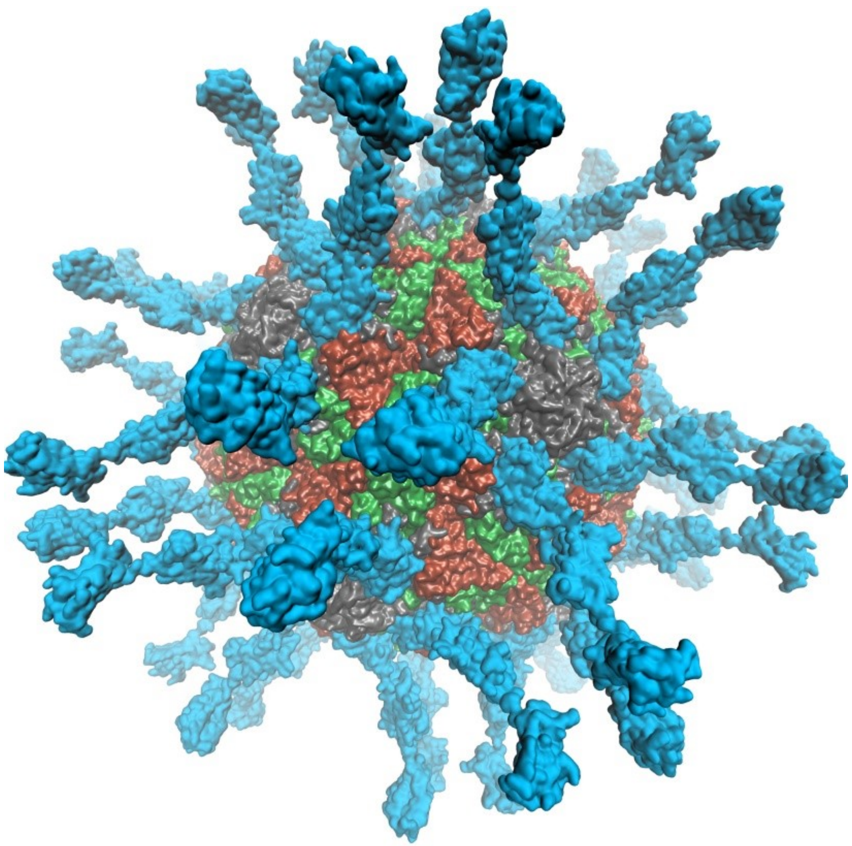
Unlocking the Full Potential of the Cray XK7 Accelerator. M. D. Klein and J. E. Stone. Cray Users Group, Lugano Switzerland, May 2014.

Weaknesses of GLX for HPC

- Requires a windowing system
- Cedes all hardware-specific interactions to window sys., leaving apps with no information about physical devices
- No real support for NUMA or multi-GPU:
  - GLX and OpenGL predate NUMA hw
  - No easy way to observe proper NUMA CPU thread affinity due to excessive abstraction
  - Difficult to exploit multiple GPUs in parallel

Adaptation of VMD to EGL for in-situ and parallel rendering on clouds, clusters, and supercomputers

- Eliminate dependency on windowing systems
  - Avoid complex failure modes, e.g. X server dies but app continues
  - Eliminate job launch overhead associated with window sys
  - Use fewer system resources
- Simplified deployment of parallel VMD builds supporting off-screen rendering on both GPUs and software rasterization
- Maintains 100% of VMD OpenGL shaders and rendering features



Poliovirus

Déjà Vu

- Sun Microsystems proposed “glP” extension to solve these problems
- When Sun exited high performance graphics relatively soon thereafter, glP died too…
- Aside from projects at TACC (a large Sun site at the time), glP didn’t get very far
- Concepts good, still largely relevant, but we had to wait (**12 years!!****) for a new multi-vendor API to arise that could support this usage cleanly: **Enter EGL**

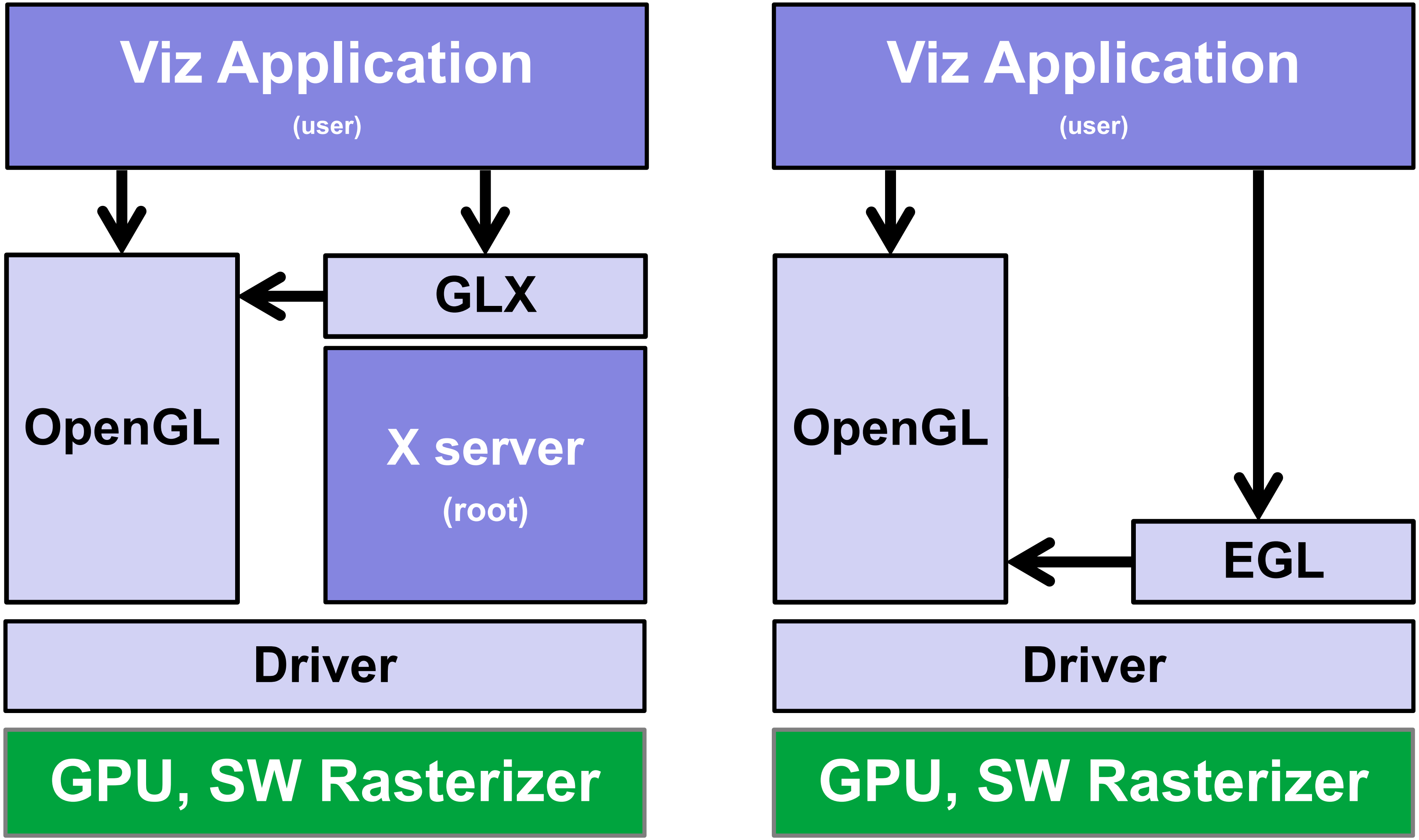
Presentation to OpenGL ARB, Sep. 21-22, 2004

****I have been pestering vendors for this for most of the 12 years!**

What is EGL?

- Interface between OpenGL rasterization APIs and underlying platform
- Manages GL context, sw/hw resources, interacts with window sys. or underlying hardware
- Originated with OpenGL for embedded sys., phones
- EGL supports OpenGL ES, full OpenGL, and other Khronos APIs, e.g. OpenVG
- To make use of EGL with full OpenGL, apps are linked against **libOpenGL.so**, EGL is used for context creation, and extensions are found via **eglGetProcAddress()**
- Current GL drivers from leading hardware vendors support EGL with full OpenGL

OpenGL: GLX vs. EGL

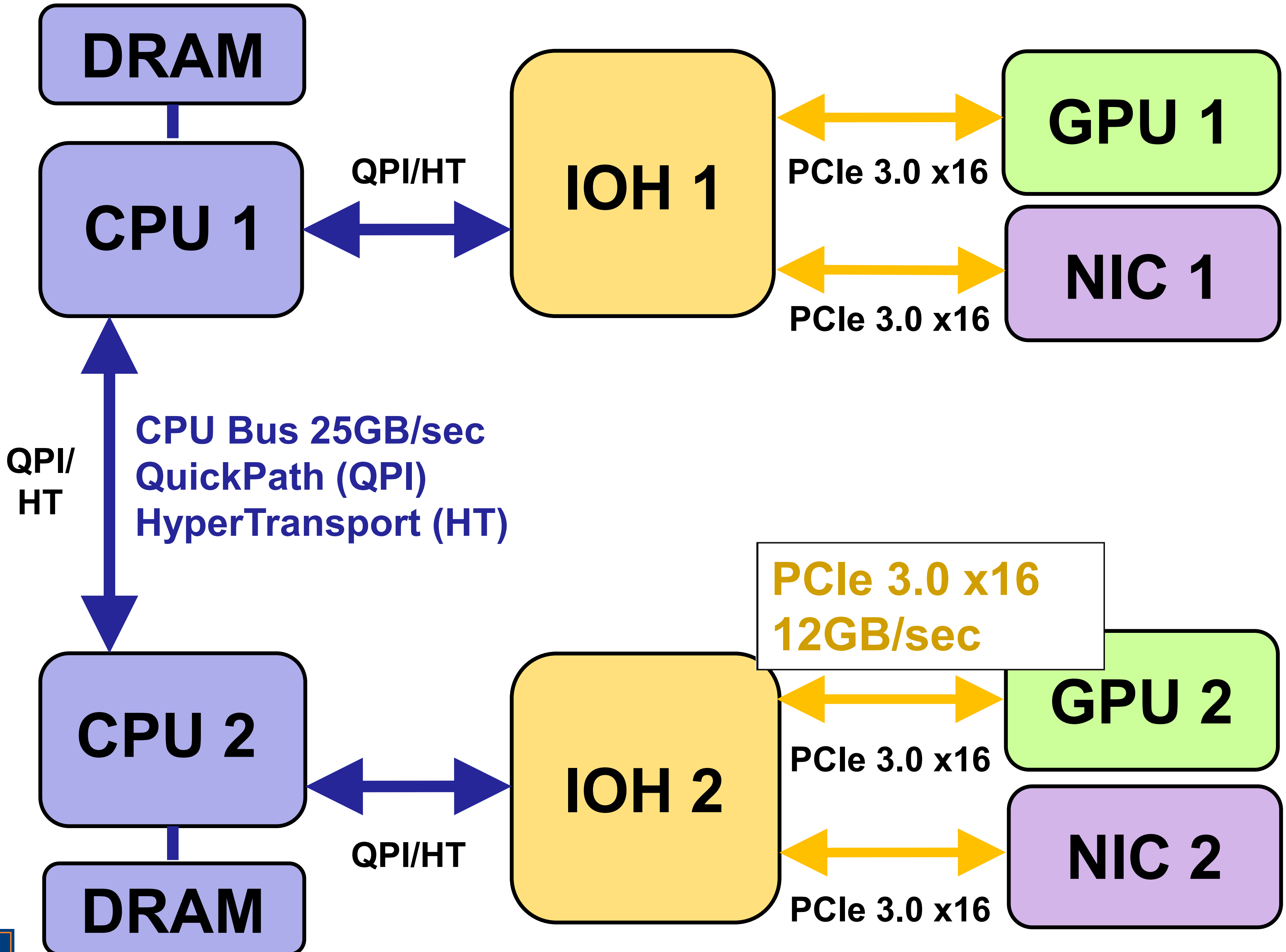


Eliminating the Windowing System

- Improves parallel job startup, simplicity
- Reduce OS threads, ctx switches, resource usage
- Gotchas:
  - All library dependencies have to be window-system-free, or “tricked” with the use of stub functions
  - App must find alternate sources for fonts and other resources typically provided by the window sys.

Benefits of EGL Platform Interfaces

- **Eliminate the need for brittle HACKS** to support both GPU rasterization and sw rasterization in the same binary!
- Enumerate and select among available implementations, **potentially supports multiple vendors in the same node**
- Allows **specific target implementation** to be bound, e.g., GPU, CPU-integrated GPU, software rasterizer
- EGL interfaces make it **EASY** to bind a CPU thread to a particular GPU or sw rasterizer context:
  - **Bind CPU threads to NUMA-optimal GPUs, MPI ranks**
  - **High-perf. multi-GPU image compositing**
- Minor similarity to OpenCL platform interfaces

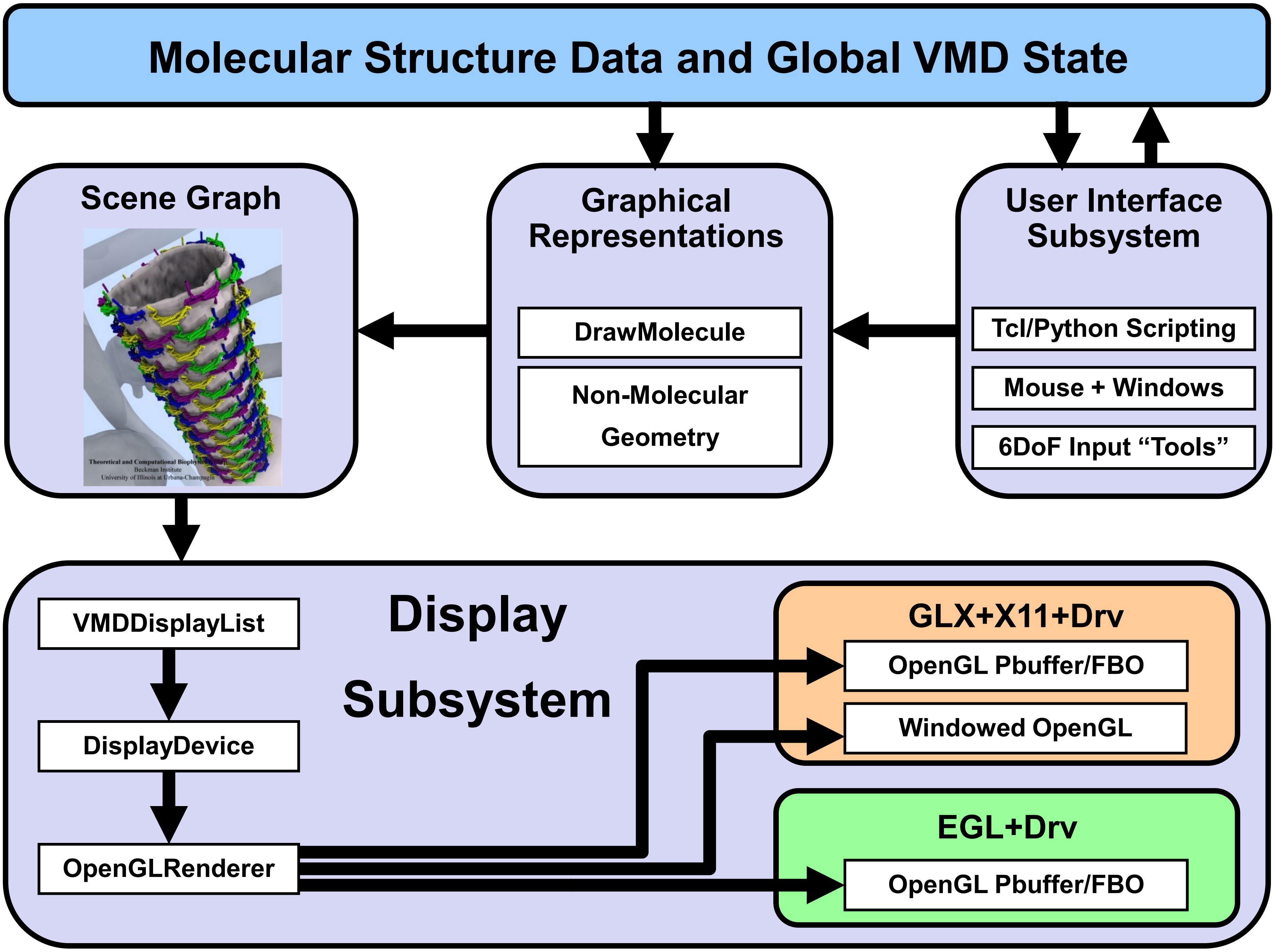

DRAM
CPU 1
QPI/HT
IOH 1
GPU 1
PCIe 3.0 x16
NIC 1
PCIe 3.0 x16
QPI/
HT
CPU Bus 25GB/sec
QuickPath (QPI)
HyperTransport (HT)
PCIe 3.0 x16
12GB/sec
GPU 2
CPU 2
PCIe 3.0 x16
IOH 2
QPI/HT
NIC 2
DRAM
PCIe 3.0 x16

EGL Loose Ends

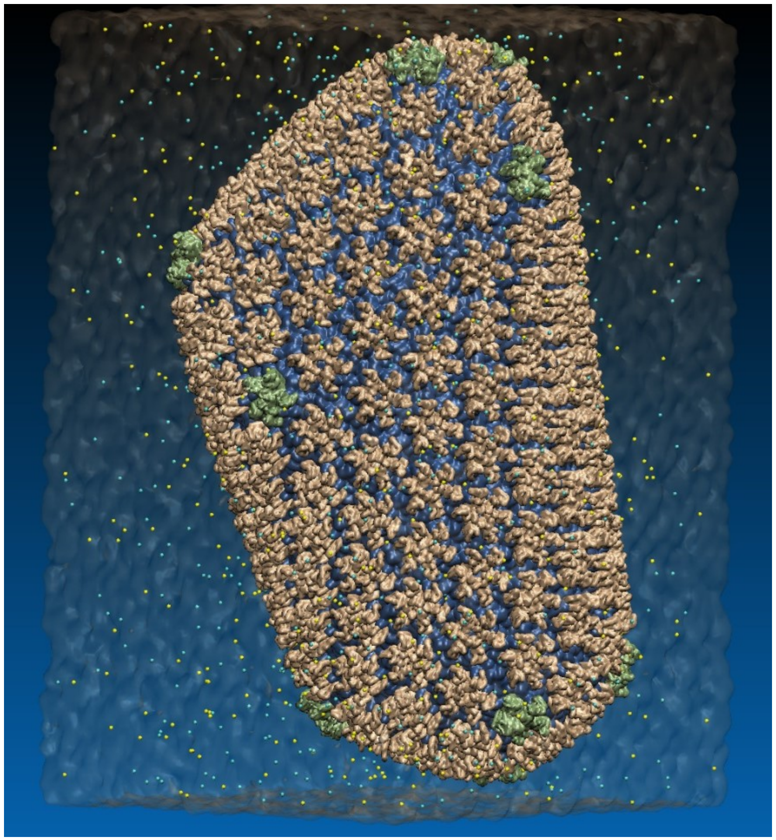
- The order of GPU or sw renderer “device” indices is determined by the vendor’s EGL implementation
- EGL can interoperate with MPI, CUDA, OpenCL, OptiX, NVENC, etc.
- Potential issue for device index order with interop APIs:
  – Proprietary APIs for graphics, compute, and video encode/decode may use arbitrary device indexing schemes
  – NVIDIA EGL implementation supports multiple indexing schemes, e.g**. PCIe-based index ordering**, which can be used to help tame the entropy among multiple APIs
- Interop APIs may support zero-copy for only certain types of EGL objects, e.g. FBOs vs. Pbuffer
- Use of FBOs vs. Pbuffers may require new shaders for features like MSAA that are “free” with a Pbuffer

Modifying VMD for EGL

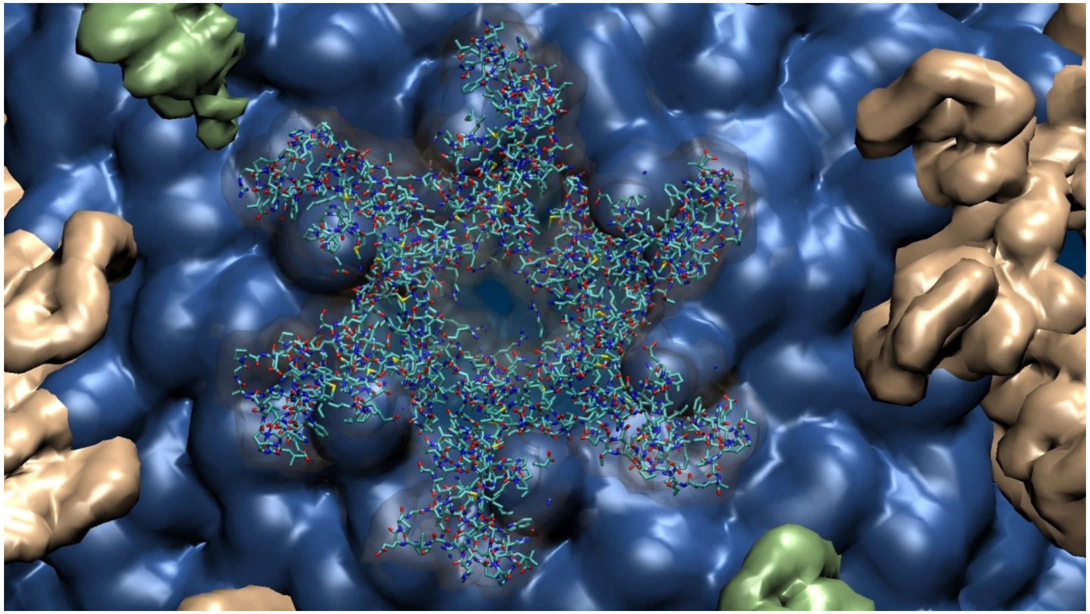
- New DisplayDevice subclass implementation for EGL
- Updates to GL extension enumeration to use EGL APIs when appropriate
- Use compiled-in Hershey font libs
- Eliminated last use of libGLU
- Provide stubs for X11 fctns referenced by other libs, e.g. old revs of OptiX


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

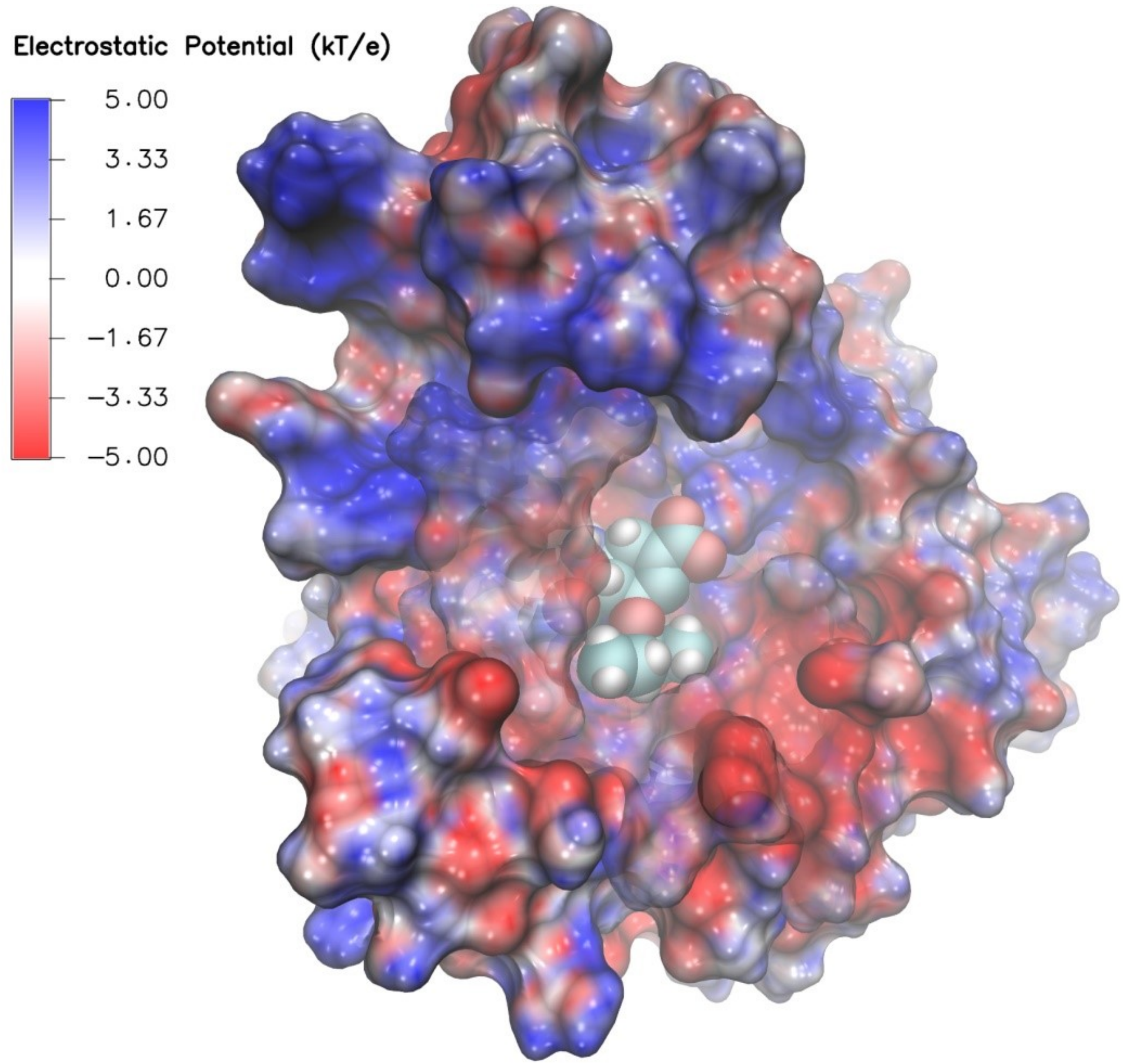
VMD EGL Rendering of HIV-1 Movie on Amazon AWS EC2 g2.8xlarge



64M atom HIV-1 capsid simulation rendered via EGL



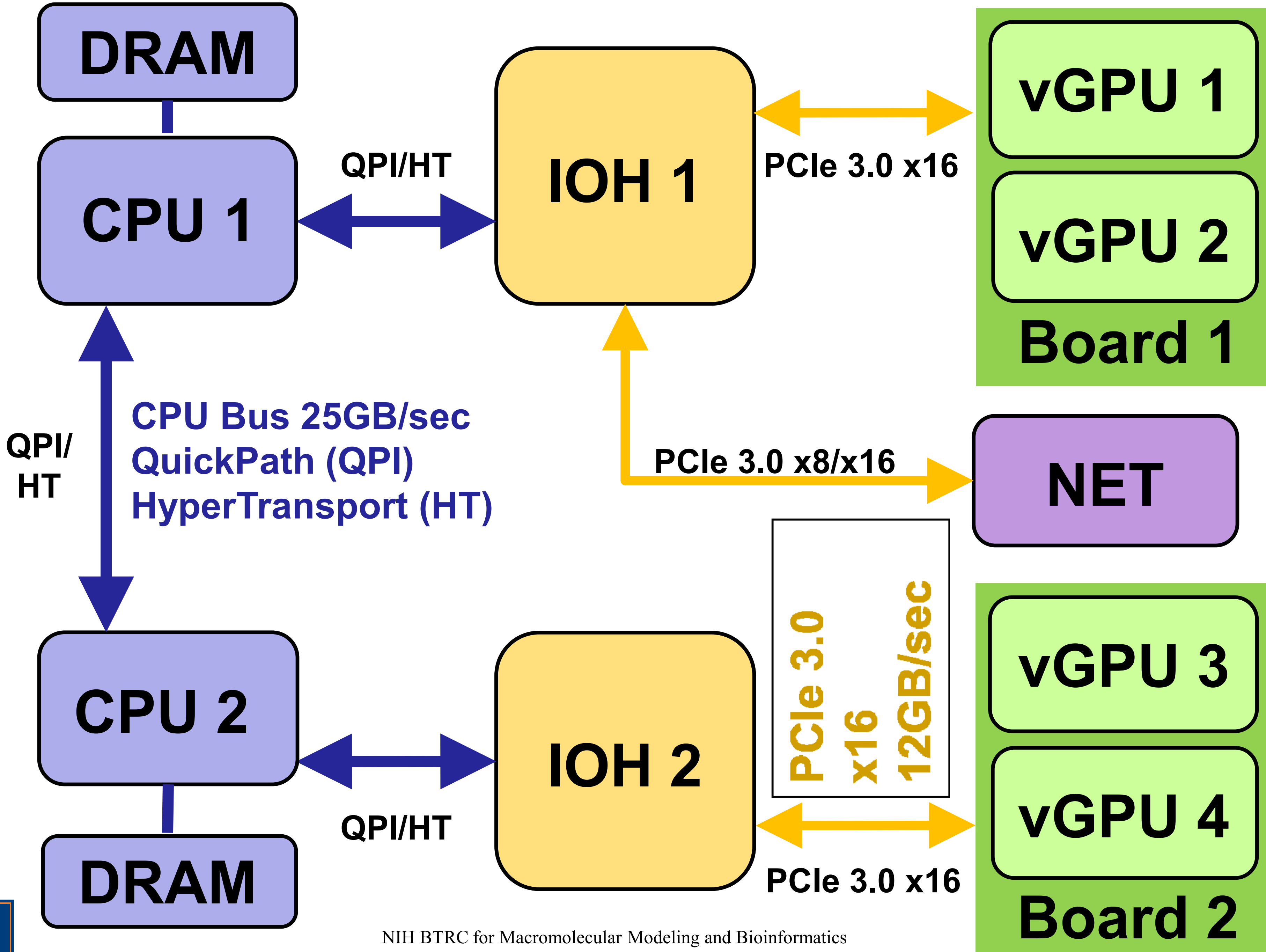
Close-up view of HIV-1 hexamer rendered via EGL



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, ...

VMD EGL Performance Tests

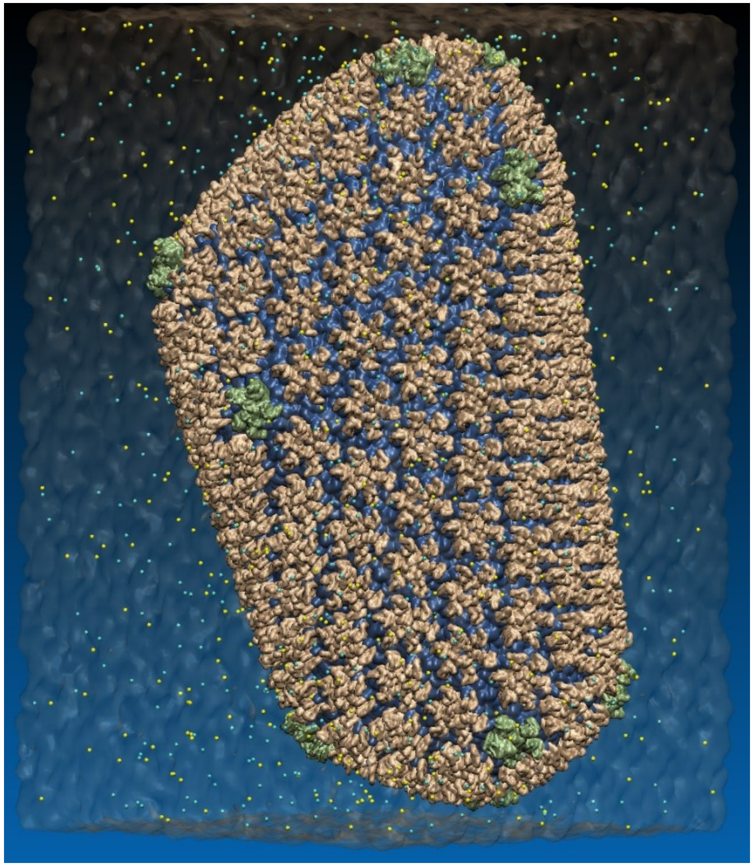
- Initial development performed on workstations
- Parallel MPI tests all performed on Amazon AWS EC2 “g2.8xlarge” instance types:
  - **Easy test deployment, no disruption to production clusters**
  - Installed devel EGL drivers on Amazon G2 GPU instance
  - **Added MPI-rank to EGL index GPU affinity rules for Amazon EC2 G2 instances**
- Performance tests were based on a movie rendering for HIV-1 capsid visualization
  - AWS EBS not comparable to supercomputer storage sys!
  - To mitigate I/O impact, tests used instance-local SSDs for output images, input becomes bottleneck at 32 nodes…



DRAM
CPU 1
QPI/HT
IOH 1
PCIe 3.0 x16
vGPU 1
vGPU 2
Board 1
QPI/
HT
CPU Bus 25GB/sec
QuickPath (QPI)
HyperTransport (HT)
PCIe 3.0 x8/x16
NET
PCIe 3.0
x16
12GB/sec
vGPU 3
vGPU 4
Board 2
CPU 2
IOH 2
QPI/HT
PCIe 3.0 x16
DRAM

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 FPS

64M atom HIV-1 capsid simulation rendered via EGL

HPC Viz. Apps Supporting EGL Today

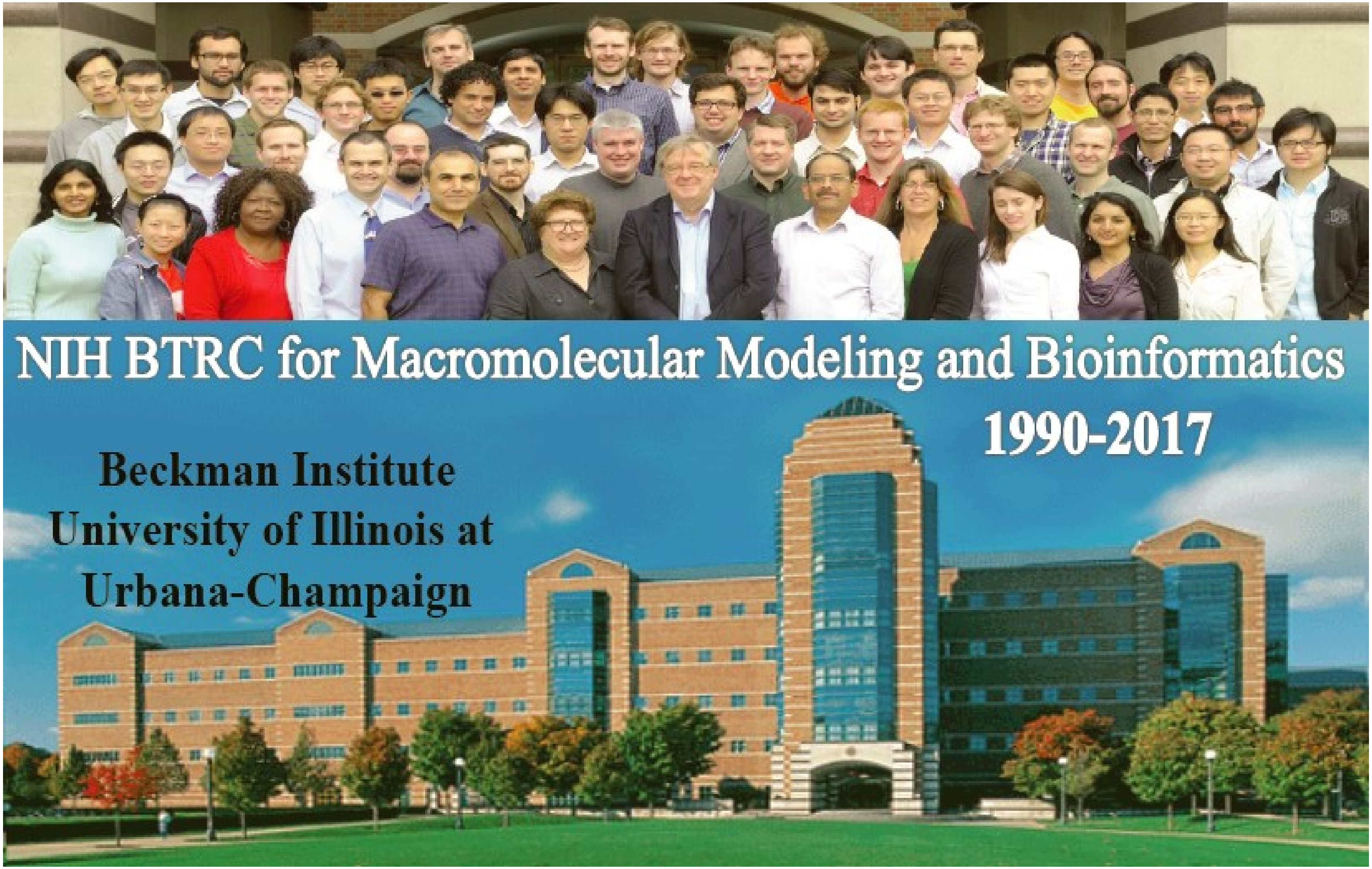
- ParaView
- VMD
- VTK

Future Work

- VMD support for EGL w/ NVENC and other hardware video encoding APIs for remote visualization
- Production EGL support on CSCS Piz Daint (in test now), NCSA Blue Waters, ORNL Titan, etc.
- EGL-based compositing for multi-GPU node architectures similar to the upcoming ORNL and LLNL "Summit" and "Sierra" systems
- Testing EGL with latest Mesa and OpenSWR, e.g. for ANL "Theta", "Aurora"

Acknowledgements

- Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign
- Amazon AWS EC2 team
- NCSA
- 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


NIH BTRC for Macromolecular Modeling and Bioinformatics
1990-2017
Beckman Institute
University of Illinois at
Urbana-Champaign

Related Publications

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

- **Immersive Molecular Visualization with Omnidirectional Stereoscopic Ray Tracing and Remote Rendering.** John E. Stone, William R. Sherman, and Klaus Schulten.High Performance Data Analysis and Visualization Workshop, IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2016. **(In-press)**
- **High Performance Molecular Visualization: In-Situ and Parallel Rendering with EGL.** John E. Stone, Peter Messmer, Robert Sisneros, and Klaus Schulten.High Performance Data Analysis and Visualization Workshop, IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2016. **(In-press)**
- **Evaluation of Emerging Energy-Efficient Heterogeneous Computing Platforms for Biomolecular and Cellular Simulation Workloads.** John E. Stone, Michael J. Hallock, James C. Phillips, Joseph R. Peterson, Zaida Luthey-Schulten, and Klaus Schulten.25th International Heterogeneity in Computing Workshop, IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2016. **(In-press)**
- **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, 2016. **(In-press)**
- **Chemical Visualization of Human Pathogens: the Retroviral Capsids.** Juan R. Perilla, Boon Chong Goh, John E. Stone, and Klaus SchultenSC'15 Visualization and Data Analytics Showcase, 2015.

Related Publications

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

- **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, 2014. **(In press)**
- **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. 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.