ATPESC Track 4: 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/vmd/
Argonne Training Program on Exascale Computing (ATPESC)
3:45pm-4:30pm, St. Charles Amphitheater, Q Center,
St. Charles, IL, Monday August 5th, 2019
VMD – “Visual Molecular Dynamics”

- 100,000 active users worldwide
- Visualization and analysis of:
  - Molecular dynamics simulations
  - Lattice cell simulations
  - Quantum chemistry calculations
  - Cryo-EM densities, volumetric data
- User extensible scripting and plugins
- [http://www.ks.uiuc.edu/Research/vmd/](http://www.ks.uiuc.edu/Research/vmd/)
VMD Hands-On Tutorials

- [http://www.ks.uiuc.edu/Training/Tutorials/#vmd](http://www.ks.uiuc.edu/Training/Tutorials/#vmd)
  - 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
Goal: A Computational Microscope
Study the molecular machines in living cells

Ribosome: target for antibiotics

Poliovirus
VMD: Building A Next Generation Modeling Platform

- Provide tools for simulation preparation, visualization, and analysis
  - Reach cell-scale modeling w/ all-atom MD, coarse grained, Lattice Microbes
  - Improved performance, visual fidelity, exploit advanced technologies (GPUs, VR HMDs)
- Enable hybrid modeling and computational electron microscopy
  - Load, filter, process, interpret, visualize multi-modal structural information
- Connect key software tools to enable state-of-the-art simulations
  - Support new data types, file formats, software interfaces
- Openness, extensibility, and interoperability are VMD hallmarks
  - Reusable algorithms made available in NAMD, for other tools
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

Smoothes initial learning curve (non-expert users)

Speed up tedious simulation preparation tasks (expert users)

Reproducibility: detailed log of all steps

Interactive preparation, simulation, and analysis

Parametrization with the Force Field Toolkit (FFTK)

A plugin enabling the simulation of drug-like small molecules by lowering force field development barriers

A graphical interface organized to follow the standardized CGenFF workflow

- Find Missing Parameters
- Geometry Optimization (QM)
- Water Interaction Energy (QM)
- Charge Optimization
- Hessian Calculation (QM)
- Bond & Angle Optimization
- Torsion Scan (QM)
- Dihedral Optimization

Leverages VMD’s powerful graphics capabilities to visualize parameter-related data within the context of molecular structure

Import data from the CGenFF Program webserver to assign initial parameters by analogy

Interactively explore energy profiles

Future Plans:
- Support free-for-academics QM software, ORCA (Neese, Max Planck)
- Parametrization tools for additional MM force fields
  - AMBER force field (Gumbart, Georgia Tech.)
  - Drude polarizable force fields (Roux, U. Chicago; MacKerell, U. Maryland)

Extensive training material available from the Center’s website

- Documentation Website
- Workshop Tutorial
- Screencast Demos
<table>
<thead>
<tr>
<th>Analysis</th>
<th>Modeling</th>
<th>Visualization</th>
<th>Collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>APBSRun</td>
<td>AutoIonize</td>
<td>Clipping Plane Tool</td>
<td>Remote Control</td>
</tr>
<tr>
<td>CatDCD</td>
<td>AutoPSF</td>
<td>Clone Rep</td>
<td>Data Import</td>
</tr>
<tr>
<td>Contact Map</td>
<td>Chirality</td>
<td>DemoMaster</td>
<td>Multiplot</td>
</tr>
<tr>
<td>GofRGUI</td>
<td>Chirality</td>
<td>Dipole Watcher</td>
<td>PDBTool</td>
</tr>
<tr>
<td>HeatMapper</td>
<td>Cionize</td>
<td>Intersurf</td>
<td>MultiText</td>
</tr>
<tr>
<td>ILSTools</td>
<td>Cispeptide</td>
<td>Navigate</td>
<td></td>
</tr>
<tr>
<td>IRSpecGUI</td>
<td>CGTools</td>
<td>NavFly</td>
<td></td>
</tr>
<tr>
<td>MultiSeq</td>
<td>Dowser</td>
<td>MultiMolAnim</td>
<td></td>
</tr>
<tr>
<td>NAMD Energy</td>
<td>ftk</td>
<td>Color Scale Bar</td>
<td></td>
</tr>
<tr>
<td>NAMD Plot</td>
<td>Inorganic Builder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetworkView</td>
<td>MDFFF</td>
<td>Remote</td>
<td></td>
</tr>
<tr>
<td>NMWiz</td>
<td>Membrane</td>
<td>Palette Tool</td>
<td></td>
</tr>
<tr>
<td>ParseFEp</td>
<td>Merge Structs</td>
<td>ViewChangeRender</td>
<td></td>
</tr>
<tr>
<td>PBCTools</td>
<td>Molefacture</td>
<td>ViewMaster</td>
<td></td>
</tr>
<tr>
<td>PMEpot</td>
<td>Mutator</td>
<td>Virtual DNA Viewer</td>
<td></td>
</tr>
<tr>
<td>PropKa GUI</td>
<td>Nanotube</td>
<td>VMD Movie Maker</td>
<td></td>
</tr>
<tr>
<td>RamaPlot</td>
<td>Psfgen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSD Tool</td>
<td>RESPTool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSD Trajectory Tool</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSD Visualizer Tool</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salt Bridges</td>
<td>RNAView</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence Viewer</td>
<td>Solvate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symmetry Tool</td>
<td>SSRestraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timeline</td>
<td>TopoTools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TorsionPlot</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VolMap</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Selected VMD Plugins: Center Developed, and User Developed**

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

**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

**Modeling**
- AutoIonize
- AutoPSF
- Chirality
- Chirality
- Cionize
- Cispeptide
- CGTools
- Dowser
- ftk
- Inorganic Builder
- MDFFF
- Membrane
- Merge Structs
- Molefacture
- Mutator
- Nanotube
- Psfgen
- RESPTool
- RNAView
- Solvate
- SSRestraints
- TopoTools

**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

**Data Import and Plotting**
- Remote Control
- Data Import
- Multiplot
- PDBTool
- MultiText

**Collaboration**
- Remote
- Palette Tool
- ViewMaster
- Virtual DNA Viewer
- VMD Movie Maker

**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/
Example VMD Visualization and Analysis Plugins

**Bendix**
Dahl ACE, Chavent M and Sansom MSP  

**Normal Mode Wizard**
Bakan A, Meireles LM, Bahar I  
VMD “Coming Soon”: VMD 1.9.4 and VMD-Next
VMD 1.9.4, and VMD-Next

- Python 3.x support
- New “molefacture” structure editor plugin
- Improved structure building and analysis tools
- High performance GPU structure+data clustering
- Density map and volume processing features: high performance GPU image segmentation, density map simulation, masking, visualization
- Many new and updated user-contributed plugins
- Deeper integration of interactive ray tracing
  - Seamless interactive RT in main VMD display window
  - Support trajectory playback in interactive RT
  - Enable multi-node interactive RT on HPC systems
- Built-in (basic) interactive remote visualization on HPC clusters and supercomputers

GPU Ray Tracing of HIV-1 Capsid Detail
VMD Data Model
VMD “Molecule”

• Collection of self-consistent molecular information
  – Atomic structure, time-varying atomic coordinates, velocities, …
  – Volumetric data
  – Graphics objects

• Key missing information required for visualization is auto-generated heuristically – “guessed”, if not provided in input files
  – Bond topology

• Limitations:
  – Fixed atom count per-timestep, requires use of “dummy atoms” for so-called “open system” simulations
  – VMD doesn’t store or use MD force field parameters, various other data that are typically specific to particular simulation tools, force fields, etc.
Atomic Structure

• Per-Atom attributes:
  – Name
  – Type
  – Residue ID, Unique residue ID, Residue Name, Residue Type
  – Bond list
  – Atomic number
  – Alternate location identifier
  – Insertion code
  – Chain
  – Segment name
  – Atom/Residue classification (protein, nucleic, hydrogen, water, “other”)
Other, Optional Per-Atom Fields

• Optional per-atom attributes:
  – Mass
  – Charge
  – Radius
  – Occupancy
  – B-factor
  – Fields primarily used by simulation preparation tools:
    • Bond Orders
    • Angles, Dihedrals, Impropers, Cross-Term Maps
Time-Varying Trajectory Data

- Atomic coordinates
- Atomic velocities
- PBC unit cell
- QM, QM/MM molecular orbital data
- User-defined time-varying scalar quantities:
  - user, user2, user3, user4
VMD Development Efforts
Supporting Integrative Hybrid Modeling

- Extending mmCIF PDBx parser to encompass new IHM-specific records, data types
- Revising VMD “molfile plugin” APIs to communicate IHM data to VMD and represent it natively
- New atom selection keywords that encompass IHM structure data
- New graphical interfaces to query and interact with IHM data both quantitatively and visually
Coarse-Grained IHM Data

- Coarse grained sphere/bead models
- Restraint information from experiments
- Multi-modal structure alignments, comparisons
- Linkage to underlying experimental images, statistics, etc.

Nuclear Pore Complex, PDB-DEV IHM #12
Display of Uncertainty, Error in IHM Models

- Query, visualize uncertainty, error, variance, in EM density maps, tomograms, atomic structure
- Requires IHM models to specify these statistics in the files
- Modeling tools, graphical interfaces can use this to guide user modeling tasks, analyses

tRNA magnesium ion occupancy probability density surfaces, VMD volmap plugin
Computing Challenges Posed by Large Hybrid Models

• Techniques like coarse graining allow modeling to reach the cell scale, but data sizes and interactivity remain a tremendous challenge

• Next-generation parallel- and GPU-accelerated computing techniques can make powerful analytical and visualization tools interactive for the first time:
  – Clustering analyses (structure RMSD, quality-of-fit, docking scores, etc)
  – Image segmentation, docking, alignment, fitting, coarse-graining…

VMD supports analysis and visualization of multi-gigavoxel EM tomograms, density maps
Interoperability Challenges Posed by IHM Data

• Reliable interoperability between modeling and simulation tools hinges upon deep understanding of IHM data semantics

• When creating multi-modal hybrid models, quantitative and visual representation of “uncertainty” and “error” becomes important, so that the best information modalities are used to make analyses, decisions, and interpretations

• What are the physical properties assigned to coarse-grained “beads”, Gaussians?

• Can those properties be understood among many different modeling tools?
Overcoming the Biomolecular Orientation of VMD for Materials Science Work

• VMD molecular data structures are optimized for biomolecules
  – Expects atom/residue names that (roughly) follow RCSB Protein Databank conventions
  – Significant up-front structure analysis enables increased interactivity later
  – Cost of this is decreased generality in various cases:
    • Fixed atom count per timestep
    • Unrecognized atom naming conventions can inhibit use of some of the high-level atom selections

• Nanodevice and materials science data can be displayed in VMD
  – Assign per-atom name/type fields after loading
  – Enable VMD to better recognize structural elements
  – Example: xyz or LAMMPS dump files lack some expected fields, so it helps to use special features of LAMMPS plugin to assign them
VMD LAMMPS Molfile Plugin
https://sites.google.com/site/akohlmey/software/lammps-plugin

- Reads LAMMPS text mode trajectory “dump” files
- Map LAMMPS per-atom fields to VMD molfile API:
  - LAMMPSREMAPFIELDS environment variable allows user to control mapping of per-atom fields to VMD molfile API
  - List of mappings of LAMMPS fields into VMD fields, e.g. “VMD=LAMMPS”
  - Example to map LAMMPS forces into VMD velocity fields:
    set env(LAMMPSREMAPFIELDS) "vx=fx, vy=fy, vz=fz"
- Adapt LAMMPS trajectories with varying atom counts to VMD
  - LAMMPSMAXATOMS environment variable sets peak atom count for VMD, adding “dummy” atoms as placeholders:
    Dummy atoms initialized with properties: name = @, type = X, resname = @@@, segid = @@@, chain = @, mass = 0, charge = 0, radius = 0, element = X.
  - LAMMPSDUMMYPOS sets default atomic coords for dummy atoms that don’t exist for a given frame:
    set env(LAMMPSDUMMYPOS) {0.0 0.0 -10.0}
Exporting Molecular Data

- VMD can be used to prepare MD simulations in conjunction with plugins and user scripting
- Need to watch out for assumptions made by heuristics, structure building tools
- Example: bond topology may need to be modified or deleted for some force fields
- Use the most appropriate structure manipulation tools for the MD package in use, e.g.:
  - NAMD: psfgen plugin
  - LAMMPS: topotools plugin
VMD Visualization Concepts
VMD Approach to Visualization

- Molecular scene is composed of “graphical representations”
- Each representation encapsulates a group of selected atoms, a drawing style, coloring style, and other parameters
- Representations are independent of each other, can be toggled on/off easily, allowing molecular scenes to be built-up incrementally
- VMD atom selection language is shared with its analytical and scripting interfaces
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
Structure Visualization

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

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

• Geometrically complex scenes
• Often show a mix of structural and spatial properties
• Time varying!
• Spatial relationships important to see clearly: depth cueing, shadows, AO helpful
Satellite Tobacco Mosaic Virus
Exportin Cse1p

Nanopore
VMD Shading Comparison: EF-Tu

Outline Shader

“Goodsell” Shader

Glossy Shader

Ray Tracing: Ambient Occlusion, Shadowing
VMD Scripting
VMD Scripting

• Built-in Tcl interpreter is standard in all builds
• Custom builds can contain Python interpreter too
• Collection of simple VMD scripts:
  http://www.ks.uiuc.edu/Research/vmd/script_library/
Atom Selections

set ionsel [atomselect top "ions"]
set watsel [atomselect top "water"]
set solute [atomselect top "not water and not ions"]

set numions [$ionsel num]
set coords [$solute get {x y z}]
Built-in “Measure” Analysis Routines

• Provide high-performance routines for common analysis functions
• Operate on atom selections, sometimes over a frame range

set com [measure center $sel1 weight mass]
set avpos [measure avpos $sel1 first 0 last 999 step 1]
set c [measure contacts 1.0 $sel1 $sel2]

set txmat [measure fit $sel1 $sel2]
$sel2 move $txmat
# Print RMSD between two selections between the first timestep and each
# later timestep for the given molecule id (default: top)
proc print_rmsd_trajectory {{mol top}} {
    set reference [atomselect $mol "protein" frame 0]  # use frame 0 for the reference
    set compare [atomselect $mol "protein"]        # the frame being compared
    set numsteps [molinfo $mol get numframes]    # query number of trajectory frames
    for {set frame 0} {$frame < $numsteps} {incr frame} {
        $compare frame $frame  # set the correct comparison frame
        set transmat [measure fit $compare $reference]  # compute the transformation
        $compare move $transmat  # align comparison to reference
        set rmsd [measure rmsd $compare $reference]  # compute the RMSD
        puts "RMSD of $frame is $rmsd"  # print the RMSD
    }
}
Trajectory Analysis and Visualization
Molecular Dynamics Trajectory Analysis

- MD simulations sample femtosecond timescales
- Millions of timesteps stored per trajectory
- Dynamics of biomolecular complexes are main interest, but solvent often accounts for half or more of the simulation content

Skip I/O for regions of bulk solvent where possible [1]

- Modern MD tools, e.g., VMD, NAMD, LAMMPS, HOOMD, employ extensive embedded scripting (Python, Tcl, etc) to permit simulation preparation, custom simulation protocols, analysis, and visualization

- Unified collective variables module allows identical analytical computations to be performed within LAMMPS, NAMD, and VMD, during pre-simulation modeling, in-situ, and post-hoc [2]


Visualization of MD Trajectories

- Allow researchers study trajectories by analyzing force profiles, energies, structural changes, etc.
- **Visualization selections, graphics, structure properties, colors can all be recomputed for each trajectory timestep!**
Folding Dynamics of Villin Headpiece Unveiled

6.9 $\mu$s folding simulation of 30K atoms: 380GB trajectory


Key Folding Intermediates
Time-Averaged Volumetric Properties

• Compute density, distance, occupancy, potential maps for a frame or averaged over a trajectory
• Example: display binding sites for diffusively bound ions as probability density isosurfaces

![tRNA magnesium ion occupancy: VMD volmap plugin](image.png)
Large System Analysis and Visualization
All-Atom Molecular Dynamics Today

- HIV Capsid
- ATP Synthase
- Photosynthetic Chromatophore
- Aquaporin
- Lysozyme
- STMV
- Ribosome

Number of Atoms:

- $(2 \text{ nm})^3$
- $(100 \text{ nm})^3$

Year:

VMD Petascale Visualization and Analysis

- Combination of growing system sizes and timescales of simulation trajectories poses a major data size challenge for molecular visualization and analysis.

- Parallel I/O rates up to 275 GB/sec on 8192 Cray XE6 nodes – can read in 231 TB in 15 minutes!

- Analyze/visualize large trajectories too large to transfer off-site:
  - User-defined parallel analysis operations, data types
  - Parallel rendering, movie making

- Supports GPU-accelerated compute nodes for both visualization and analysis tasks:
  - GPU accelerated trajectory analysis w/ CUDA
  - OpenGL and GPU ray tracing for visualization and movie rendering

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

Parallel VMD currently available on:
ORNL Summit and Titan, NCSA Blue Waters, IU Big Red II, CSCS Piz Daint, many similar systems
HIV-1 Data: ~1.2 TB/day @ 4096 XK7 nodes
Petascale Molecular Dynamics I/O and Storage Challenges

- NAMD simulations can produce up to 10TB/day @ 1024 nodes (~20%) of ORNL Summit, more as optimizations raise NAMD performance further.
- Petascale science campaigns require months of simulation runs.
- Long-term storage of large-fractional petabytes impractical.
- Historical “download output files for analysis and visualization” approach is a non-starter at this scale.
- Demands visualization and analysis operate on the data in-place on the HPC system, whether post-hoc, in-transit, or in-situ.
- Analyses must identify salient features of structure, dynamics, cull data that don’t contribute to biomolecular processes of interest.
NAMD simulations can generate up to 10TB of output per day on 20% of Summit
Next Generation: Simulating a Proto-Cell

- Emulate aspects of the *Mycoplasma mycoides* bacterium
- 200nm diameter
- ~1 billion atoms w/ solvent
- ~1400 proteins in membrane

Cryo-ET image of ultra-small bacteria (scale bar 100nm)
Proto-Cell Data Challenges

- 1B-atom proto-cell requires nodes with more than TB RAM to build complete model...
- 1B-atom proto-cell binary structure file: 63GB
- Trajectory frame atomic coordinates: 12GB, 1.2TB/ns of simulation (1 frame per 10ps)
- Routine modeling and visualization tasks are a big challenge at this scale
  - Models contain thousands of atomic-detail components that must work together in harmony
  - Exploit persistent memory technologies to enable “instant on” operation on massive cell-scale models – eliminate several minutes of startup during analysis/visualization of known structure
  - Sparse output of results at multiple timescales will help ameliorate visualization and analysis I/O
  - Data quantization, compression, APIs like ZFP
IBM AC922 w/ 6 GPUs

- **Power Supplies (2x)**
  - 2200W
  - 200VAC, 277VAC, 400VDC input

- **Nvidia Volta GPU**
  - 3 per socket
  - SXM2 form factor
  - 300W
  - NVLink 2.0
  - Air/Water Cooled

- **Memory DIMM's (16x)**
  - 8 DDR4 IS DIMMs per socket
  - 8, 16, 32, 64, 128GB DIMMs

- **PCIe slot (4x)**
  - Gen4 PCIe
  - 2 x16 HHHL Adapter
  - 1 Shared Slot
  - 1 x8 HHHL Adapter

- **BMC Card**
  - IFMI
  - 1 Gb Ethernet
  - VGA
  - 1 USB 3.0

- **Power 9 Processor (2x)**
  - 16, 22C water cooled
  - 16, 20C air cooled
IBM AC922 Summit Node

- **3 GPUs Per CPU Socket**
  - Tesla V100 GPU
  - Tesla V100 GPU
  - Tesla V100 GPU

- **X-Bus**
  - 64GBps

- **DDR4 DRAM**
  - 120GBps

- **POWER9 CPU**

- **InfiniBand**
  - 12GBps

- **1.6TB SSD**
  - "Burst Buffer"

- **Nvlink 2.0**
  - 2x 50GBps:
    - 100GBps
VMD Off-Screen Rendering w/ EGL

- Containers+Cloud+Workstations with recent NVIDIA drivers
- VMD on HPC systems w/ latest GPUs:
  - Cray XC50, CSCS Piz Daint
  - **ORNL Summit in progress now**
  - IBM OpenPOWER, drivers 375.66 and later support both GLX and EGL
Swine Flu A/H1N1 neuraminidase bound to Tamiflu

VMD EGL Rendering: Supports full VMD GLSL shading features
Vulkan support coming soon...

64M atom HIV-1 capsid simulation

High Performance Molecular Visualization: In-Situ and Parallel Rendering with EGL.
Running VMD on Distributed Memory Clouds, Clusters, and Supercomputers with MPI
Using VMD MPI Builds

• See “Running VMD on Supercomputers” section in the VMD tutorial
• Run one MPI rank (VMD process) per compute node
  – Each rank uses all CPU cores and all GPU accelerators they find
  – GPUs are shared between graphics/visualization and computing tasks
• Parallel VMD Scripting APIs:
  – “parallel” script subcommands, e.g., “parallel for ...”:
    • “nodename”, “noderank”, “nodecount”, “barrier”, “allgather”, “allreduce”, “for”
  – Available even in non-MPI builds, for single-node script compatibility
• Work scheduling:
  – Intra-node work scheduled via multithreading, CUDA, etc.
  – Inter-node distributed memory work scheduling of user analysis/viz scripts is provided by built-in load balancing implementation for VMD
Example MPI Session, CSCS Piz Daint

stonej@daint103> srun -C gpu -n 256 --ntasks-per-node=1 /users/stonej/local/bin/vmd193 -dispdev text -e rendermovie.tcl

Info) VMD for CRAY_XC, version 1.9.3 (December 15, 2016)
Info) http://www.ks.uiuc.edu/Research/vmd/
Info) Email questions and bug reports to vmd@ks.uiuc.edu
Info) Please include this reference in published work using VMD:
Info)-------------------------------------------------------------
Info) Creating CUDA device pool and initializing hardware...
Info) Initializing parallel VMD instances via MPI...
Info) Found 256 VMD MPI nodes containing a total of 6144 CPUs and 256 GPUs:
Info) 0: 24 CPUs, 60.8GB (96%) free mem, 1 GPUs, Name: nid03072
Info) 1: 24 CPUs, 60.8GB (96%) free mem, 1 GPUs, Name: nid03073
[...example output omitted...]
Info) 254: 24 CPUs, 60.9GB (96%) free mem, 1 GPUs, Name: nid03376
Info) 255: 24 CPUs, 60.9GB (96%) free mem, 1 GPUs, Name: nid03377
Parallel MDFF Cross Correlation Analysis on Cray XK7

Rabbit Hemorrhagic Disease Virus (RHDV)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traj. frames</td>
<td>10,000</td>
</tr>
<tr>
<td>Structure component selections</td>
<td>720</td>
</tr>
<tr>
<td>Single-node XK7 (projected)</td>
<td>336 hours (14 days)</td>
</tr>
<tr>
<td>128-node XK7</td>
<td>3.2 hours (105x speedup)</td>
</tr>
<tr>
<td>2048-node XK7</td>
<td>19.5 minutes (1035x speedup)</td>
</tr>
</tbody>
</table>

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

VMD EGL Performance on Amazon EC2 Cloud

<table>
<thead>
<tr>
<th>MPI Ranks</th>
<th>EC2 “G2.8xlarge” GPU Instances</th>
<th>HIV-1 movie rendering time (sec), (I/O %) 3840x2160 resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>626s (10% I/O)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>347s (19% I/O)</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>221s (31% I/O)</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>141s (46% I/O)</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>107s (64% I/O)</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>90s (76% I/O)</td>
</tr>
</tbody>
</table>

Performance at 32 nodes reaches ~48 FPS


64M atom HIV-1 capsid simulation rendered via EGL.
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

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

High Fidelity Ray Tracing for Interactive and Cinematic Visualization
High Fidelity Ray Tracing

- Advanced rendering techniques save scientists time, produce images that are easier to interpret
- Ambient Occlusion, Depth of Field, high quality transparency, instancing, ....
- **Interactive RT** on laptops, desk, cloud, and remote supercomputers
- Large-scale MPI parallel rendering: in situ or post hoc visualization tasks
- Stereoscopic panorama and full-dome projections
- **Omnidirectional VR:** YouTube, HMDs
- Built-in ray tracing engines:
  - **Tachyon:** cross-platform RT
  - **NVIDIA OptiX:** GPU RTX-accelerated
  - **Intel OSPRay:** CPU x86-optimized
Lighting Comparison, STMV Capsid

Two lights, no shadows

Ambient occlusion + two lights, 144 AO rays/hit
Geometrically Complex Scenes

Ray tracing techniques well matched to molecular viz. needs:

- Curved geometry, e.g. spheres, cylinders, toroidal patches, easily supported
- Greatly reduced memory footprint vs. polygonalization
- Runtime scales only moderately with increasing geometric complexity
- Occlusion culling is “free”, RT acceleration algorithms do this and much more
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
VMD w/ OptiX 6

- Interactive RT on laptops, desktops, and cloud
- Large-scale parallel rendering: in situ or post hoc visualization
- Remote ray tracing with NvPipe video streaming
- Stereoscopic panoramic and full-dome projections
- Omnidirectional VR for YouTube, VR HMDs
- VMD+OptiX NGC container: https://ngc.nvidia.com/registry/
- GPU memory sharing via NVLink
- In-progress:
  - Denoising: faster turnaround w/ AO, DoF, etc


VMD/OptiX GPU Ray Tracing of all-atom Chromatophore w/ lipids.
VMD/OpiX RTX Acceleration
What does RTX Acceleration do for VMD?

- Hardware acceleration (Turing RT cores) of:
  - BVH AS traversal
  - Ray-triangle intersection
- BVH AS can embed triangle geometry
  - Triangle geometry buffer can be ephemeral
- Often gives VMD a \( \sim 6 \times \) performance gain
VMD Scene w/ OptiX RTX APIs

Scene Graph

OptiX RTX Scene Construction

Incoming VMD Geom

General Geometry

Allocate Geometry Buffers

Assign Closest-Hit Material Shaders

Triangle Meshes

Allocate Triangle Buffers

Assign Closest-Hit Material Shaders

...
Secondary Benefits and RTX Performance Observations

- Traversal and intersection work performed by RT cores vacates GPU SMs and makes them available for other RT work
- RTX hardware traversal performance approaches GPU memory bandwidth limits
- Future RTX hardware could end up being bandwidth bound in some cases
- Start adapting geometric data representations for minimum footprint, e.g. by using compressed or quantized data representations such as Octahedron Normal Vector encoding (replace 3x 32-bit floats with a single 32-bit int)
VMD OptiX RT performance on Quadro RTX 6000

<table>
<thead>
<tr>
<th>Speedup Factor X</th>
<th>Chromatophore @ 4Kx4K</th>
<th>Chrom Cell, 512x DoF @ 1080p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Quadro GV100 1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1</td>
<td>2x Quadro GV100 1.97</td>
<td>1.95</td>
</tr>
<tr>
<td>2</td>
<td>Quadro RTX 6000 8.02</td>
<td>8.18</td>
</tr>
</tbody>
</table>
VMD Application Examples and Implementation Details
Cinematic Rendering:
“Birth of Planet Earth” Fulldome Show

- Joint project with:
  - NCSA Advanced Visualization Lab
  - Thomas Lucas Productions, Inc.
  - Spitz Creative Media
  - Tellus Science Museum

- NSF Support: CADENS award ACI-1445176

- Premiered March 28, Zeiss Großplanetarium, Berlin, Germany
Why Use VMD for Cinematic Rendering

- Use all science materials where possible
- Maintain peak efficiency for calculating molecular geometries and representations
- OptiX-based RT engine and data structures designed for memory efficiency and performance specifically for molecular models
- Built-in scripting engines enabled control and incorporation within cinematic pipeline
- Avoid I/O bottlenecks
VMD Planetarium Dome Master Camera

- Fully interactive RT with ambient occlusion, shadows, depth of field, reflections, …
- Both mono and stereoscopic
- No post-processing required

Ray Tracing Gems Ch. 4
“Birth of Planet Earth” Cinematic Pipeline

Visualization and Rendering

VMD

Nuke

Virtual Director

Houdini

Camera Path

Geometry

Cam Posns

Images
Integration of VMD with Virtual Director
Camera Path in Virtual Director and VMD
Stereoscopic Panorama Ray Tracing

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

• Ray trace 360° images and movies for Desk and VR HMDs: Oculus, Vive, Cardboard
• Stereo spheremaps or cubemaps allow very high-frame-rate interactive OpenGL display
• AO lighting, depth of field, shadows, transparency, curved geometry, …
• Summit 6x Tesla V100 GPU nodes:
  – Render many omni-stereo viewpoints, no acceleration structure rebuilds, tens of frames/sec per-node!
  – OptiX multi-GPU rendering, NVLink compositing and data distribution, etc…
  – Future: AI for warping between views


Ray Tracing Gems

- Ch. 4, “A Planetarium Dome Master Camera”
- Ch. 27, “Interactive Ray Tracing Techniques for High-Fidelity Scientific Visualization”
- Tons of great material and code samples!
Molecular Dynamics Flexible Fitting (MDFF)

X-ray crystallography → MDFF ← Electron microscopy

APS at Argonne

ORNL Titan

FEI microscope

Density Map Segmentation

VMD GPU-accelerated density map segmentation of GroEL

Cryo-EM / Cryo-ET Density Map Segmentation

Evaluate 3-D volumetric electron density maps and segment them, to identify key structural components.

Index/label components so they can be referred to, colored, analyzed, and simulated…
Structural Route to the all-atom HIV-1 Capsid


Crystal structures of separated hexamer and pentamer

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

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 from an all-atom structure.

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
Padding optimizes global memory performance, guaranteeing coalesced global memory accesses.

Grid of thread blocks

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

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

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

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

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

Threads producing results that are used.

Inactive threads, region of discarded output.

Single-Pass MDFF GPU Cross-Correlation.
Parallel MDFF Cross Correlation Analysis on Cray XK7

Rabbit Hemorrhagic Disease Virus (RHDV)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Traj. frames</td>
<td>10,000</td>
</tr>
<tr>
<td>Structure component selections</td>
<td>720</td>
</tr>
<tr>
<td>Single-node XK7 (projected)</td>
<td>336 hours (14 days)</td>
</tr>
<tr>
<td>128-node XK7</td>
<td>3.2 hours 105x speedup</td>
</tr>
<tr>
<td>2048-node XK7</td>
<td>19.5 minutes 1035x speedup</td>
</tr>
</tbody>
</table>

RHDV colored by relative CC

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

VMD Tesla V100 Cross Correlation Performance

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

Volta GPU architecture almost 2x faster than previous gen Pascal:

<table>
<thead>
<tr>
<th>Application and Hardware platform</th>
<th>Runtime, Speedup vs. Chimera, VMD+GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chimera Xeon E5-2687W (2 socket) [1]</td>
<td>15.860s, 1x</td>
</tr>
<tr>
<td>VMD-CUDA IBM Power8 + 1x Tesla K40 [2]</td>
<td>0.488s, 32x 0.9x</td>
</tr>
<tr>
<td>VMD-CUDA Intel Xeon E5-2687W + 1x Quadro K6000 [1,2]</td>
<td>0.458s, 35x 1.0x</td>
</tr>
<tr>
<td>VMD-CUDA Intel Xeon E5-2698v3 + 1x Tesla P100</td>
<td>0.090s, 176x 5.1x</td>
</tr>
<tr>
<td>VMD-CUDA IBM Power8 “Minsky” + 1x Tesla P100</td>
<td>0.080s, 198x 5.7x</td>
</tr>
<tr>
<td>VMD-CUDA Intel Xeon E5-2697Av4 + 1x Tesla V100</td>
<td>0.050s, 317x 9.2x</td>
</tr>
<tr>
<td>VMD-CUDA IBM Power9 “Newell” + 1x Tesla V100</td>
<td>0.049s, 323x 9.3x</td>
</tr>
</tbody>
</table>


New VMD MDFF Density Map Tools

- **New** Map Tools tab of MDFF GUI provides wide array of density map manipulation tools including:
  - **New** Rigid Body Fitting
  - **New** Interactive Histogram
  - Trim, Crop, Clamp, Smooth…
  - Easy Masking routine

- **New** Density Segmentation
- Add, subtract, multiply maps
- Cross correlation and potential calculations for MDFF
New Density Map Tools - Masking

Easily select and mask density map regions with VMD selection language

TRPV1 structure (3J5P) and cryo-EM density (emd-5778) colored by local resolution obtained by ResMap

High Res (~3 Å)
Med Res (~4 Å)
Low Res (~5 Å)

Use of Node-Local Burst Buffers and Non-Volatile Memory DIMMs

- Perform viz+analysis in-transit in node-local SSDs, persistent memory NVDIMMs
- ORNL Summit I/O:
  - Parallel FS: 2.5 TB/s
  - Node-local PCIe “burst buffer” SSDs: 10+ TB/sec, 7PB capacity
- Plenty of capacity for full-detail MD trajectories, could enable ~100x increase in temporal resolution in cases where it would be valuable to the science
- Enable all-pairs trajectory clustering analyses and resulting visualizations
- Future systems with NVDIMMs (3D Xpoint, phase change memory) could eventually provide bandwidths approaching DRAM
- Use NVDIMMs w/ mmap(), APIs like PMDK to perform formerly-out-of-core calculations using persistent memory:
  - https://github.com/pmem/pmdk
- Imagine future Summit-like machines w/ NVLink-connected GPUs w/ access to high-bandwidth persistent memory on each node
Trade FLOPS for Reduced I/O

ORNL Summit compute node:
- 6x Tesla V100 GPUs, 2x POWER9 CPUs
- GPUs Peak: ~46 DP TFLOPS, ~96 SP TFLOPS
- Peak IB rate per node: ~23GB/sec
- Ratio of FLOPS vs. I/O:
  ~2,000 DP FLOPS/byte, ~4000 SP FLOPS/byte
  ~16K FLOPS per FP word

Unconventional approach: Recompute to avoid I/O
Computing+Visualizing Molecular Orbitals

- Movies of simulation trajectories provides insight into results
- QM, and hybrid (QM/MM) MO visualizations historically done from huge “cube” files, impractical
- Store QM wavefunctions + Gaussian basis set, only 10s of KB per stored timestep compared to 100s of MB
- Recompute MO grid on-the-fly from QM basis set, huge decrease in RAM+I/O in exchange for heavy FP arithmetic

http://dx.doi.org/10.1038/nmeth.4638

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++;

.....
MO GPU Parallel Decomposition

MO 3-D lattice decomposes into 2-D slices (CUDA grids)

Small 8x8 thread blocks afford large per-thread register count, shared memory

Each thread computes one MO lattice point.

Padding optimizes global memory performance, guaranteeing coalesced global memory accesses

Grid of thread blocks

Threads producing results that are used

Threads producing results that are discarded

Lattice computed using multiple GPUs

GPU 0

GPU 1

GPU 2
VMD C\textsubscript{60} MO Viz. Perf, 516x519x507 Grid: @ .13s/frame, avoids 3.8GB/s I/O per-node

<table>
<thead>
<tr>
<th>Hardware platform</th>
<th>Runtime,</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Power8 (ORNL ‘crest’) + 1x Tesla K40 [1]</td>
<td>3.49s,</td>
<td>1.0x</td>
</tr>
<tr>
<td>Intel Xeon E5-2697Av4 + 1x Tesla V100</td>
<td>0.610s,</td>
<td>5.7x</td>
</tr>
<tr>
<td>Intel Xeon E5-2697Av4 + 2x Tesla V100</td>
<td>0.294s,</td>
<td>11.8x</td>
</tr>
<tr>
<td>Intel Xeon E5-2697Av4 + 3x Tesla V100</td>
<td>0.220s,</td>
<td>15.9x</td>
</tr>
<tr>
<td>IBM Power9 “Newell” + 1x Tesla V100</td>
<td>0.394s,</td>
<td>8.8x</td>
</tr>
<tr>
<td>IBM Power9 “Newell” + 2x Tesla V100</td>
<td>0.207s,</td>
<td>16.8x</td>
</tr>
<tr>
<td>IBM Power9 “Newell” + 3x Tesla V100</td>
<td>0.151s,</td>
<td>23.1x</td>
</tr>
<tr>
<td>IBM Power9 “Newell” + 4x Tesla V100</td>
<td>0.130s,</td>
<td>26.8x</td>
</tr>
</tbody>
</table>

MO Kernel Structure, Opportunity for NRTC JIT…
Data-driven execution, but representative loop trip counts in (…)

Loop over atoms (1 to ~200) {
  Loop over electron shells for this atom type (1 to ~6) {
    Loop over primitive functions for this shell type (1 to ~6) {
      Small loop trip counts result in significant loop overhead. Runtime kernel generation and NVRTC JIT compilation can achieve in a large (1.8x!) speed boost via loop unrolling, constant folding, elimination of array accesses, …
    }
  }
}

Loop over angular momenta for this shell type (1 to ~15) {}}
Molecular Orbital Computation and Display Process
Runtime Kernel Generation, NVRTC Just-In-Time (JIT) Compilation

One-time initialization

- Initialize Pool of GPU Worker Threads

Read QM simulation log file, trajectory

- Preprocess MO coefficient data
  - eliminate duplicates, sort by type, etc...

Generate/compile basis set-specific CUDA kernel

For each trj frame, for each MO shown

- Compute 3-D grid of MO wavefunction amplitudes
  - using basis set-specific CUDA kernel

Extract isosurface mesh from 3-D MO grid

Render the resulting surface

For current frame and MO index, retrieve MO wavefunction coefficients
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;
    }
}

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

General loop-based data-dependent MO CUDA kernel

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

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;
        tmpshell += const_wave_f[ifunc++] * xdist * zdist;
        tmpshell += const_wave_f[ifunc++] * ydist * zdist;
        value += tmpshell * contracted_gto;
        break;
    }
}

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;
    tmpshell += const_wave_f[ifunc++] * zdist2;
    tmpshell += const_wave_f[ifunc++] * xdist * ydist;
    tmpshell += const_wave_f[ifunc++] * xdist * zdist;
    tmpshell += const_wave_f[ifunc++] * ydist * zdist;
    value += tmpshell * contracted_gto;

1.8x Faster
Vectors of wavefunction amplitudes are computed using hardware SIMD instructions.

MO 3-D lattice decomposes into 2-D slices.

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

Padded + aligned array

SIMD lanes producing results that are used

Padding:
Inactive SIMD lanes or region of discarded output used to guarantee aligned vector loads+stores
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];

    __m512 expval = _mm512_mul_ps(_mm512_set1_ps(exponent * MLOG2EF), dist2);

    __m512 retval = _mm512_exp2a23_ps(expval);

    __m512 ctmp = _mm512_mul_ps(_mm512_set1_ps(contract_coeff), retval);
    contracted_gto = _mm512_add_ps(contracted_gto, ctmp);
    prim_counter += 2;
}

// expf() approximation required, use (base-2) AVX-512ER instructions…
__m512 retval = _mm512_exp2a23_ps(expval);
Performance of AVX-512ER Intrinsics 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
Convenient Packaging of HPC-Optimized Versions of VMD and NAMD
Making Our Research Tools Easily Accessible

• Docker “container” images available in NVIDIA NGC registry
  – Users obtain Docker images via registry, download and run on the laptop, workstation, cloud, or supercomputer of their choosing
  – https://ngc.nvidia.com/registry/

• Cloud based deployment
  – Full virtual machines (known as “AMI” in Amazon terminology)
  – Amazon AWS EC2 GPU-accelerated instances:
    http://www.ks.uiuc.edu/Research/cloud/


VMD / NAMD / LM, NGC Containers

VMD

VMD is designed for modeling, visualization, and analysis of biomolecular systems such as proteins, nucleic acids, lipid membranes, carbohydrate structures, etc. VMD provides a wide variety of graphical representations for visualizing and coloring molecular structures: molecular surfaces, space-filling CPK spheres and cylinders, licorice bonds, backbone tubes and ribbons, secondary structure cartoons, and others.

VMD can be used to animate and analyze the trajectory of a molecular dynamics (MD) simulation. In particular, VMD can act as a graphical front end for an external MD program by transferring the trajectory to VMD and then using VMD's animation and analysis tools.
VMD OptiX/EGL NGC Container

- https://ngc.nvidia.com/registry/
- CUDA-accelerated viz+analysis
- EGL off-screen rendering – no windowing system needed
- OptiX high-fidelity GPU ray tracing engine built in
- All dependencies included
- Easy to deploy on a wide range of GPU accelerated platforms

Technology Opportunities and Collaborations

- Supercomputer Centers, Cray, IBM
  - Remote visualization
  - Performance, power profiling and optimization
- NVIDIA
  - GPU computing
  - Ray tracing
  - Remote visualization
  - ARM, Tablets, power profiling and optimization
- Intel
  - x86, Xeon Phi optimization, ray tracing
- Amazon
  - Cloud deployment of VMD/NAMD, related tools
  - Remote visualization
- Universities:
  - G. Fiorin, J. Henin, Toni Giorgino, collective variables
  - T. Ertl, U. Stuttgart: visualization algorithms
  - M. Kuttel, U. Cape Town: visualization and analysis
  - W. Sherman, Indiana U.: VR HMDs, visualization

Energy efficiency: ARM+GPU
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/


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


- **Winner of the SC'14 Visualization and Data Analytics Showcase**


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


Related Publications

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


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


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


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


Related Publications

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


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


