Interactive Molecular Dynamics for Nanomechanical and Nanochemical Experiments

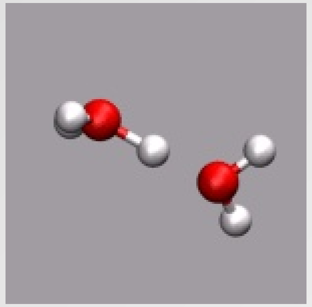
Axel Kohlmeyer

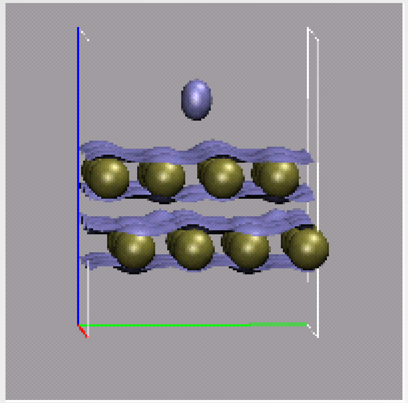
ICMS

Institute for Computational Molecular Science

Molecular Dynamics

- Simulate motion of atoms and molecules according to physical models: classical (empirical) or quantum (electrons and/or core)
- Microscopic look into the atomic scale; simulated experiment with perfect control.

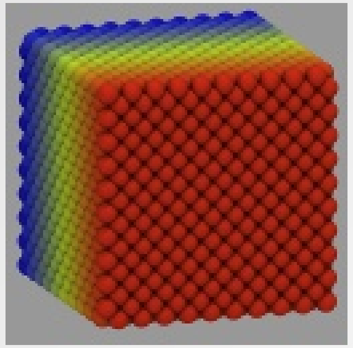




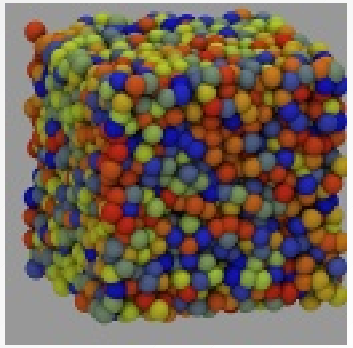
- Connection to macroscopic world through statistical mechanics or thermodynamics: => molecular level interpretation of thermodynamic quantities
- Beyond two particles chaotic => coupled differential equations, solved numerically => model and CPU determine time scales

Typical Molecular Dynamics Work Flow

- Setup: construct initial geometry, idealized or assembled pieces



- Equilibration: relax and propagate until the desired thermodynamical state is reached (or close enough)
- Production: propagate atoms and record statistically relevant data of system evolution while equilibrium is maintained
- Analysis & Visualization: done “off-line” (i.e. after production)
  - Statistical analysis to derive structural or thermodynamical properties to confirm, guide or predict experiment(s).
  - Visual inspection of structural changes or “special” events



ICMS Institute for Computational Molecular Science

Studying 'Rare Events' in Molecular Dynamics

- Time scales in MD simulations are limited by the fastest motion => total time that can be studied is restricted.
- Size of system is finite => large (local) energy fluctuations rare => events that have (free) energy barriers are often 'impossible'
- Various 'biasing methods' exist to make those events possible => biasing needs to be programmed, cannot “just play around”
- Programmed biasing or steering works best for simple moves: 'collective variables' => well defined for statistical analysis
- Difficult to study “What would happen if?”-scenarios on the fly

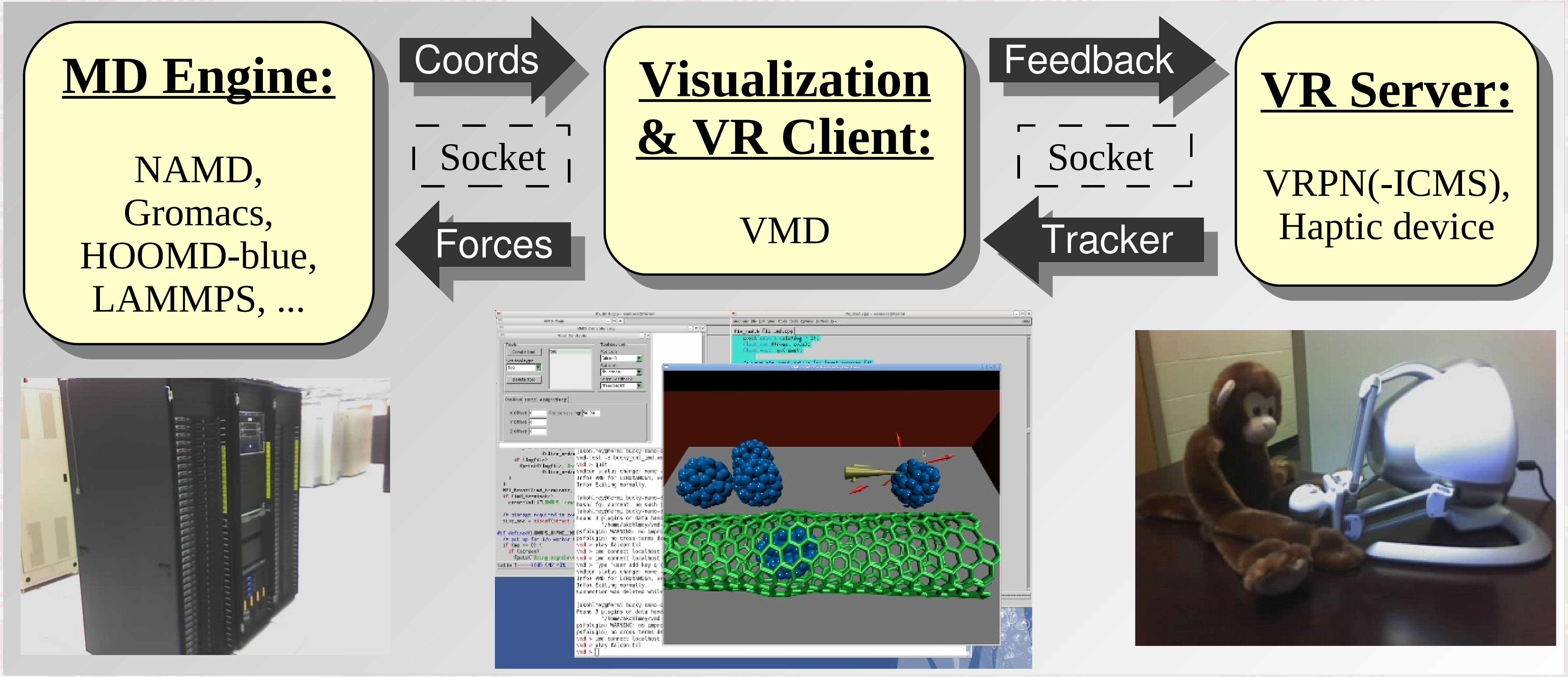
Evolution of Interactive MD (IMD)

- Origin in steered molecular dynamics (SMD) by adding run-time visualization to monitor the progress of steering
- Next step: Interactive determination of steering forces through a pointer device (2d: mouse, 3d: 3d-joystick, 3d-mouse, WiiMote)
- Then: better visual feedback with stereo display
  even better with 'immersive visualization', e.g. CAVE
- Full IMD framework with support for haptic devices
  adding force feedback to 3d tracking => VMD
- Limited adoption due to cost and disruptive nature (VR facility)

Interactive MD Applications Examples

- Education and Outreach:
  Unique experience through immersive visualization
  and force feedback allows students to “grasp” MD simulations
- Simulation Monitoring:
  Visualization can be connected to ongoing production run
- Simulation Preparation:
  Components of an MD simulation system can be interactively
  rearranged (“sculpting”) as needed while close to equilibrium
- Nano-mechanical or -chemical experiments

The IMD Infrastructure in VMD



MD Engine:

NAMD, Gromacs, HOOMD-blue, LAMMPS, ...

Coords

Socket

Forces

Visualization & VR Client:

VMD

Feedback

Socket

Tracker

VR Server:

VRPN(-ICMS), Haptic device

Time Scale Issues with IMD

Time and length scales of simulation & visualization are coupled
Typical parameters for a smooth IMD configuration:

Simulated System:

- Atom velocity: ~100 m/s
- MD Time step: ~10^{-15} s

Tangible System:

- Compute time: ~1 ms / step
- Atom movement: ~10^{-10} m / s

Newton's second law: F = *m* a
=> faster running MD
=> less IMD force needed
=> objects appear to be lighter

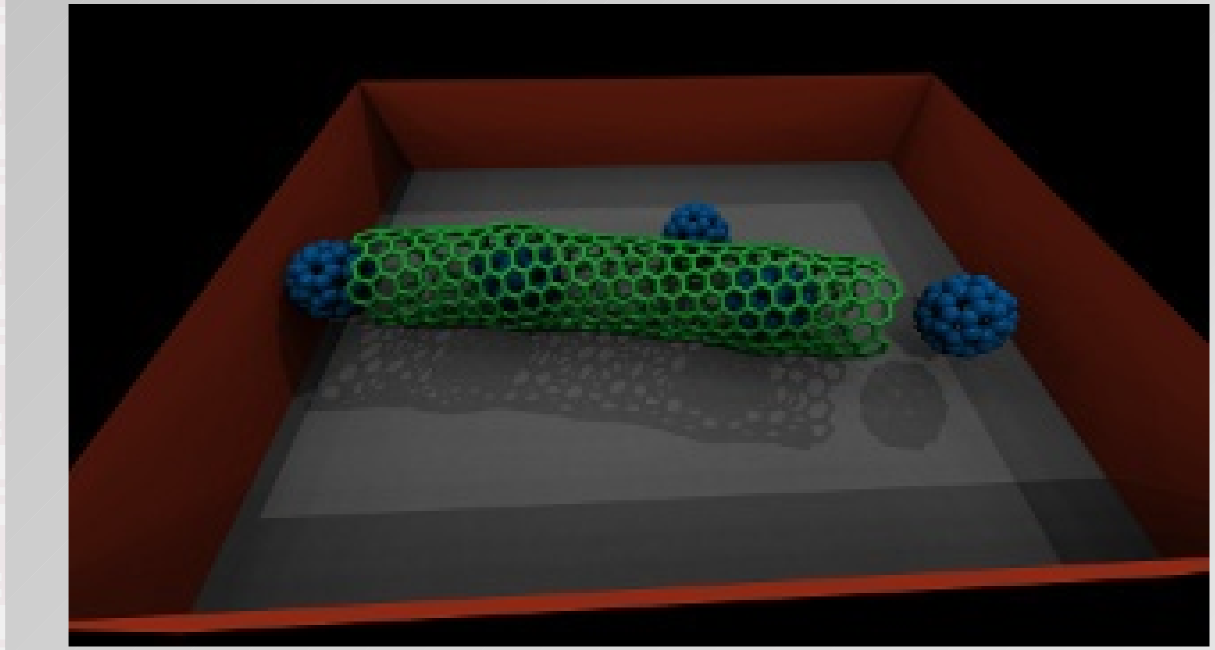
To move large object:
=> run MD faster (parallel,GPU)
=> or scale applied IMD force
=> or change particle mass

More Time Scale Issues with IMD

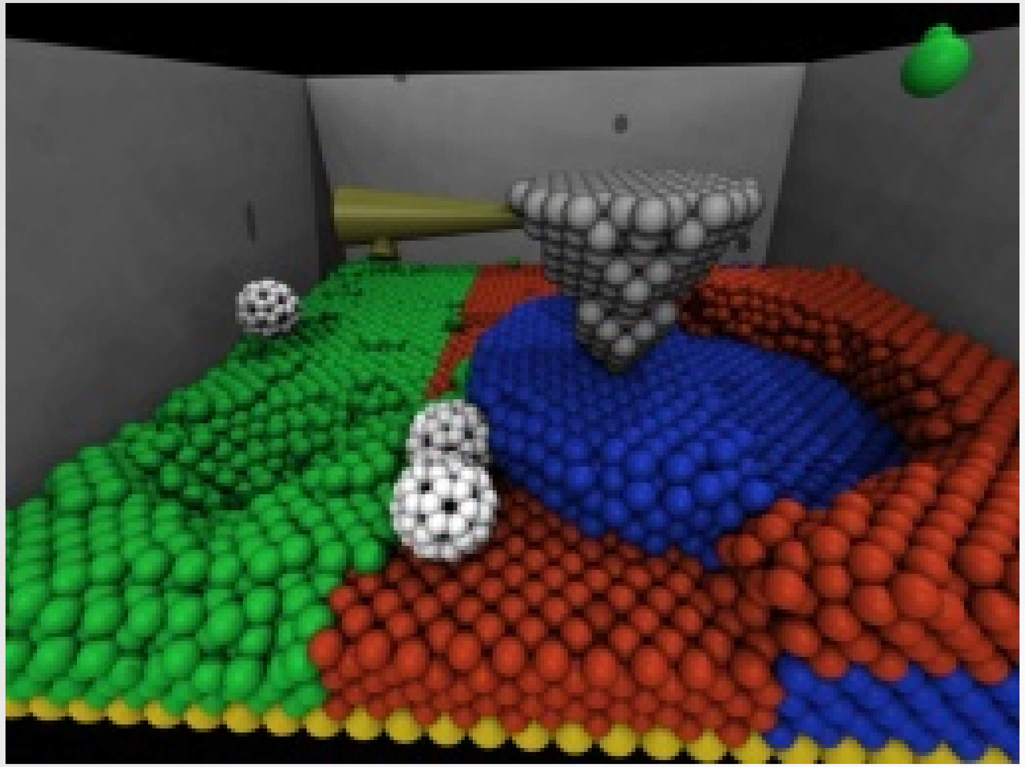
- How realistic should an IMD simulation/visualization be?
  If too large IMD force, too small particle mass => unphysical may be tolerated for educational use, unacceptable for research
- Local movements (solvent) limit length of time step
- Compute capability limits speed of MD code
- Visualization update rate limits position update frequency
  => At higher MD speed, less frequent IMD position updates
  => position data becomes more “noisy”, need denoising filter
- What if I want to look at “slow” processes? Move large objects?

Recorded IMD Demos with “Falcon” Controller

- “Stick the buckyballs into the nanotube demo



- Virtual vacuum AFM demo with 3 types of LJ particles



Revived Interest in Interactive MD

- A smooth IMD visualization needs about 20-30 frames/s
  - Significant compute power for fast MD on all but the smallest systems
  - A powerful graphics workstation with stereo capability is required

    => a dedicated and expensive facility was needed that few locations could afford and that would require to schedule access ahead of time
- 3d screens affordable (3d-TVs, Scanline polarized LCD)
- High performance graphics with 3d capability available (games)
- Multi-core CPUs and GPUs turn workstations into clusters
- Affordable controllers (Falcon (gaming), smartphones (6DOF))

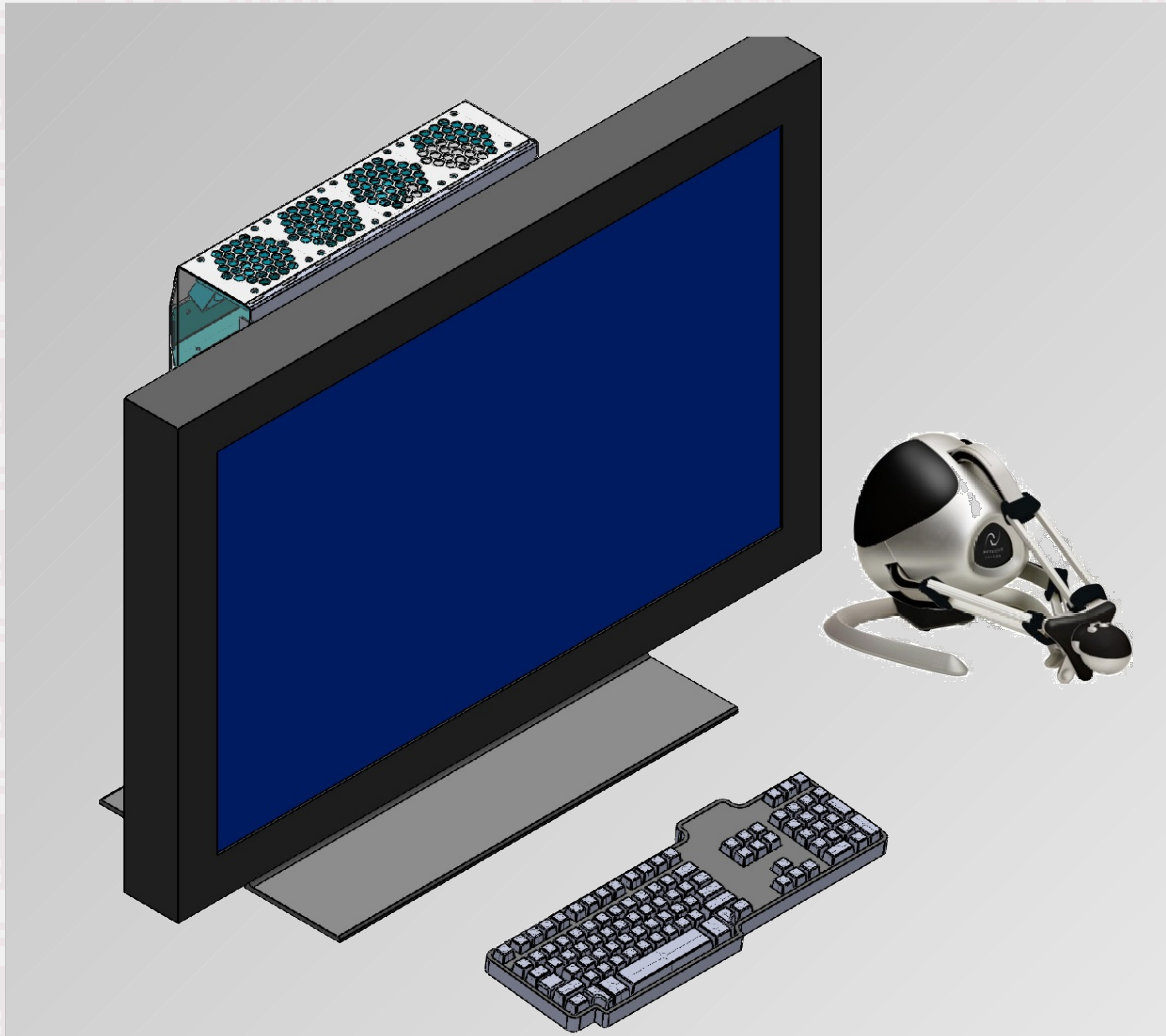
VRPN and VMD Enhancements

- Support in VRPN for Novint Falcon as haptic 3DOF device
  - Implement Tracker and Button classes as sending devices
  - Implement ForceDevice class as receiving device
  - Use libnifalcon (http://libnifalcon.nonpolynomial.org) to access Falcon
  - Implement damping scheme for smooth force constraint updates (force update in device at 1000Hz, update from VMD less frequent)
- Enhancements in VMD
  - Support for enforced TCP only communication with VRPN server for using remote visualization facility via VirtualGL (LRZ Munich)
  - Support for whole residue mode with “tug” tool

LAMMPS Enhancements

- OpenMP (LAMMPS-ICMS) and GPU (GPULAMMPS) acceleration for non-bonded interactions
  => faster MD for smaller (OpenMP) or larger (GPU) systems
- Improvements in IMD module (fix imd):
  - Listening for IMD force input in separate thread.
    No more need to “drain” all incoming IMD communication data
  - Sending of coordinate data in separate thread
    No more need to wait when large IMD data is sent over slow link
  - Addition of Savitzky-Golay filtering of coordinate data
    Denoises coordinate updates with large Δt with minimal distortion

IMD Appliance Concept



- Combines:
  - Multi-core/CPU/GPU compute
  - Stereo capable visualization
  - 3d display
  - Haptic device
  - Software
- No special facility needed
- Commodity components
- Kiosk mode for education

ICMS Institute for Computational Molecular Science

Perspectives

- Advances in GPU acceleration will expand applicability
- GPU acceleration more effective in compute intense models
  => nano-mechanics (Tersoff, Stillinger-Weber, AIREBO)
  => nano-chemistry (Reaxx)
- More approximate models for large changes
  (temporary coarse graining)
- More experiments with example applications or demos needed
- IMD protocol expansions and optimizations
- VRPN-ICMS improvements (multi-Falcon support, alternate grip)

References

- VMD: http://www.ks.uiuc.edu/Research/vmd
- LAMMPS: http://lammps.sandia.gov/
LAMMPS-ICMS (code gets merged to LAMMPS when stable):
http://sites.google.com/site/akohlmey/software/lammps-icms
- VRPN: http://www.cs.unc.edu/Research/vrpn/
VRPN-ICMS (code will be merged into VRPN when stable):
http://sites.google.com/site/akohlmey/software/vrpn-icms
- GPULAMMPS (GPU acceleration with CUDA for LAMMPS):
http://code.google.com/p/gpulammps/

Acknowledgements

- Michael L. Klein (freedom and funding, NSF CHE 09-46358)
- John Stone, Klaus Schulten (VMD and **much** more)
- Steve J. Plimpton, Paul S. Crozier and many others (LAMMPS)
- Russ Taylor (VRPN), Kyle Machulis (libnifalcon)
- Tom Anderson, Brandon Williams, Novint, (free Falcons) ...and...
- Greg & Gary Scantlen, Creative Consultants, (contacts, encouragement, perseverance, 3D-buckyballs) Make sure you try out the “Nano Dome”, Booth 29/30.