

# Molecular Dynamics

Alexander B. Pacheco

User Services Consultant  
LSU HPC & LONI  
[sys-help@loni.org](mailto:sys-help@loni.org)

LSU HPC Training Series  
Louisiana State University  
July 6, 2011



# Outline

## 1 Introduction

## 2 Fundamentals of Molecular Dynamics

## 3 *Ab Initio* Molecular Dynamics Theory

## 4 Computational Chemistry Programs

## 5 LAMMPS on LONI Systems

## 6 Molecular Dynamics on GPU



# What is Computational Chemistry

- **Computational Chemistry** is a branch of chemistry that uses principles of computer science to assist in solving chemical problems.
- Uses the results of theoretical chemistry, incorporated into efficient computer programs.
- Application to single molecule, groups of molecules, liquids or solids.
- Calculates the structure and properties such as relative energies, charge distributions, dipole and multipole moments, spectroscopy, reactivity, etc.
- Computational Chemistry Methods range from
  - 1 Highly accurate (*Ab-initio*,DFT) feasible for small systems
  - 2 Less accurate (semi-empirical)
  - 3 Very Approximate (Molecular Mechanics, Classical Mechanics) large systems



## Theoretical Chemistry: broadly can be divided into two main categories

- 1 Static Methods  $\Rightarrow$  Time-Independent Schrödinger Equation
  - ◆ Quantum Chemical/*Ab Initio* /Electronic Structure Methods
  - ◆ Molecular Mechanics
- 2 Dynamical Methods  $\Rightarrow$  Time-Dependent Schrödinger Equation
  - ◆ Classical Molecular Dynamics
  - ◆ Semi-classical and *Ab-Initio* Molecular Dynamics



## Why Molecular Dynamics?

- Electronic Structure Methods are applicable to systems in gas phase under low pressure (vacuum).
- Majority of chemical reactions take place in solution at some temperature with biological reactions usually at specific pH's.
- Calculating molecular properties taking into account such environmental effects which can be dynamical in nature are not adequately described by electronic structure methods.

## Molecular Dynamics

- Generate a series of time-correlated points in phase-space (a trajectory).
- Propagate the initial conditions, position and velocities in accordance with Newtonian Mechanics.  $\mathbf{F} = m\mathbf{a} = -\nabla V$
- Fundamental Basis is the **Ergodic Hypothesis**: the average obtained by following a small number of particles over a long time is equivalent to averaging over a large number of particles for a short time.



## Applications of Molecular Dynamics

- Liquids, transport phenomena like viscosity and heat flow
- Crystal structure, defects
- Solids: Fracture, Friction between two solids
- Surface Phenomena, adsorption, diffusion
- Clusters are a bridge between molecular systems and solids, catalysis
- Biomolecules
- Electronic Properties and Dynamics

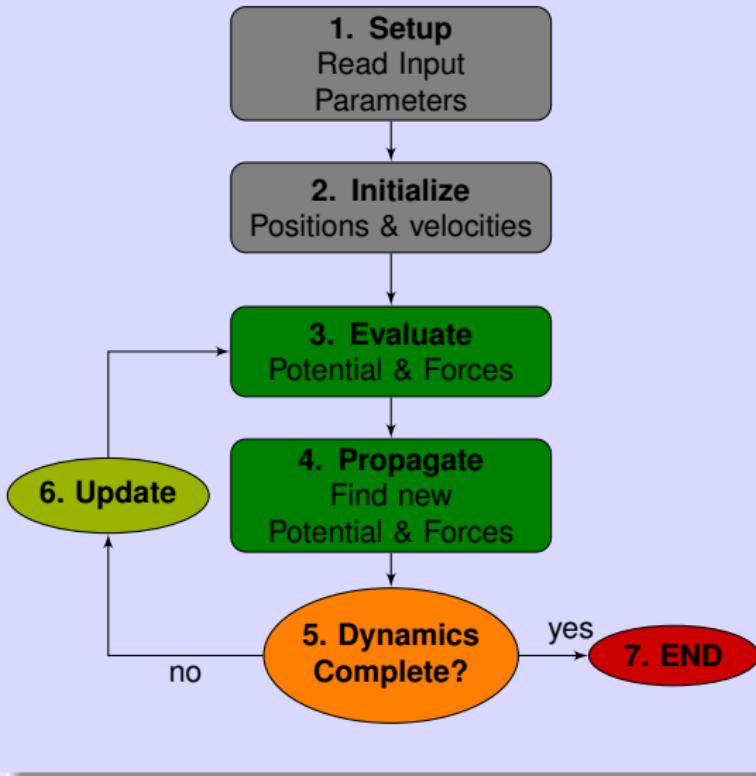
## Limitations of Molecular Dynamics

- Use of classical forces: Nuclear quantum effects become important for lighter nuclei (e.g. H, He), low temperatures.
- How realistic are the forces?
- How long should the simulation run?
- Size of system?



# General Schematic for Molecular Dynamics Program

- 1 Setup: Read input parameters
- 2 Initialize: Obtain initial positions and velocities
- 3 Evaluate: Potential Energy and Forces on nuclei
- 4 Propagate nuclei using an appropriate time integration algorithm
- 5 Check if Dynamics is complete.
- 6 If incomplete update variables and goto Step 3.
- 7 If complete end dynamics or carry out all required analysis.



## Models for Physical System

- Model for the physical system being simulated
- Choose the potential which is a function  $\mathcal{V}(\mathbf{r}_1 \cdots \mathbf{r}_N)$  of the positions of the nuclei, representing the potential energy of the system when the atoms are arranged in that specific configuration.
- Forces are obtained as gradients of the potential

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} \mathcal{V}(\mathbf{r}_1 \cdots \mathbf{r}_i \cdots \mathbf{r}_N)$$

- Simplest choice for potential is a sum of pairwise interactions

$$\mathcal{V}(\mathbf{r}_1 \cdots \mathbf{r}_N) = \sum_i \sum_{j>i} V(|\mathbf{r}_i - \mathbf{r}_j|)$$



# Potential Energy Functions

- Pair Potentials:
  - 1 Attraction
    - Long range
    - Dispersive forces, instantaneous dipole interactions
  - 2 Repulsion
    - Short range
    - Exchange forces, overlap of electron cloud.
- Lennard-Jones (LJ) Potential

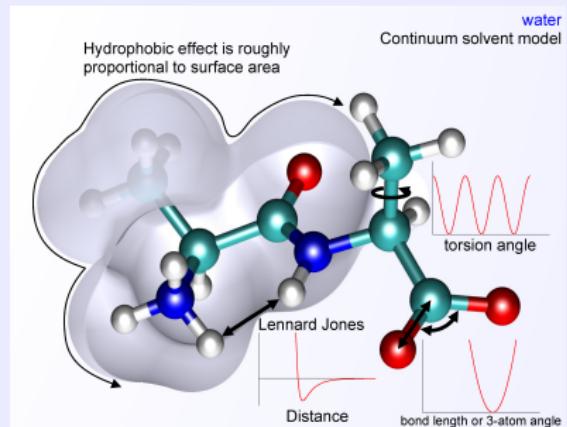
$$U(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]$$

- 7 The LJ potential decays rapidly, significant computation time can be saved by neglecting pair interactions beyond a cut-off.  
[for  $r_c > 2.5\sigma$ ,  $U(r) = 0$ ]
- Potential from Molecular Mechanics Force fields like AMBER, CHARMM, Drieding etc or from electronic structure calculations.



## General form of Molecular Mechanics equations

$$E = E_{\text{bond}} + E_{\text{angle}} + E_{\text{torsion}} + E_{\text{vdW}} + E_{\text{elec}}$$
$$= \frac{1}{2} \sum_{\text{bonds}} K_b (b - b_0)^2 \quad \text{Bond}$$
$$+ \frac{1}{2} \sum_{\text{angles}} K_\theta (\theta - \theta_0)^2 \quad \text{Angle}$$
$$+ \frac{1}{2} \sum_{\text{dihedrals}} K_\phi [1 + \cos(n\phi)]^2 \quad \text{Torsion}$$
$$+ \sum_{\text{nonbonds}} \left\{ \begin{array}{l} \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] \\ + \frac{q_1 q_2}{Dr} \end{array} \right\} \quad \begin{array}{l} \text{van der Waals} \\ \text{Electrostatics} \end{array}$$



Picture taken from  
[http://en.wikipedia.org/wiki/Molecular\\_mechanics](http://en.wikipedia.org/wiki/Molecular_mechanics)



## Initial Positions

- 1 From experimental results
- 2 Assign to lattice positions

## Initial Velocity

- Trajectories from previous simulation
- Set to desired temperature of simulation
  - 1 Assign a Maxwell -Boltzmann distribution

$$f_v(v_x) = \sqrt{\frac{m}{2\pi kT}} \exp\left[\frac{-mv_x^2}{2kT}\right]$$

- 2 Random Uniform distribution: Random velocities scaled to desired temperature

$$v_{desired} = \sqrt{\frac{T_{desired}}{T_{current}}} v_{current}$$

- Total linear momentum is zero

$$\sum_i m_i v_i = 0 \quad \text{or} \quad v_j^{desired} = v_j^{current} - \frac{\sum_i m_i v_i^{current}}{m_j}$$



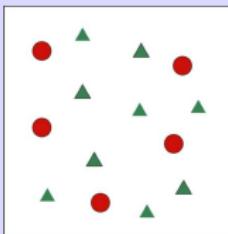
## Boundary Conditions

- What do we do at the boundaries of the simulated system?
  - 1 Nothing: Not particularly useful for infinite systems but good for single isolated or finite systems.
  - 2 Use Periodic Boundary Conditions (PBC) for simulating liquids, solids and clusters.



## Periodic Boundary Conditions

- Consider a small subset or representative size of the infinite or large system and replicate to infinity (or your desired system size).



- For each particle in our simulation box, there are infinite particles located at positions

$$\mathbf{r} + l\mathbf{a} + m\mathbf{b} + n\mathbf{c}, \quad (l, m, n = -\infty \dots \infty)$$

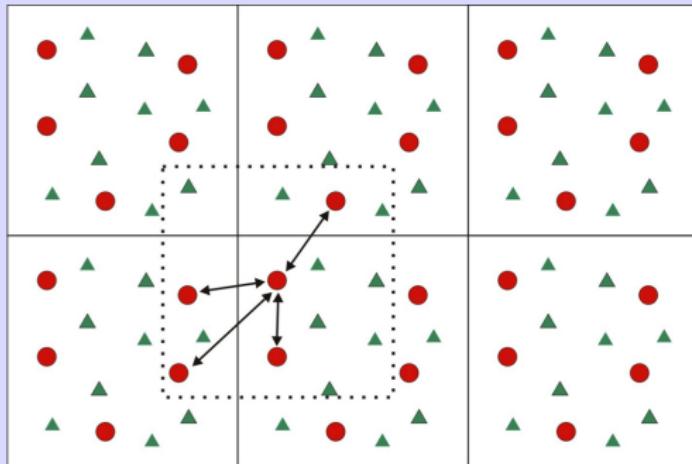
where **a**, **b** and **c** are the lattice dimensions of the box.

- Thus each particle interacts not only with the other particles in the simulation box but also with their images in every other box.
- The simulated box and its images have the same size, shape and number of atoms with the same position, velocity and acceleration.
- How do calculate these infinite interactions which will undoubtedly increase the simulation cost?



## Minimum Image Criterion

- If the potential has a finite range, say  $R_c$ , then two particles at a distance greater than  $R_c$  do not interact with each other.
- If the size of the simulation box is large than  $2R_c$ , then each particle  $i$  will at most interact with one particle  $j$  in the simulation box or the neighboring box.
- Thus interaction between particle  $i$  with particle  $j$  and its images in the replicated boxes will only consist of one interaction between the closest particle.



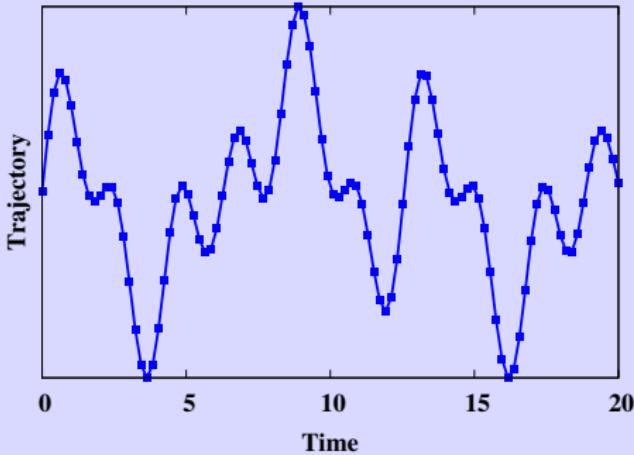
UNIVERSITY  
OF TECHNOLOGY

- Periodic Boundary Conditions are well suited for modeling infinite systems such as liquids and solids.
- What about surfaces? How do you treat surface effects?
- For surface simulation, a model of a slab is used i.e. a thick slice of the material delimited with two free surfaces.
- This is done by removing the PBC from one direction, say  $z$  direction while maintaining PBC in the  $xy$  plane.
- Thus the system is infinite in the  $xy$  plane and finite in the  $z$  direction.
- Removing PBC in two directions gives rise to a wire model.
- Removing PBC's completely gives rise to a cluster model.



## Time Integration

- The main ingredient of Molecular Dynamics is the time integration algorithm: integrate equation of motions of particles to follow trajectories.
- Based on finite difference methods: time is discretized onto a finite grid with the time step  $\Delta t$  being the distance between consecutive points on the grid.



- From position and time derivative at time  $t$ , obtain positions and time derivatives at time  $t + \Delta t$



## Verlet Algorithm

- Taylor expansion around  $\mathbf{r}(t)$

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) + \mathbf{r}(t - \Delta t) + \mathbf{a}(t)(\Delta t)^2 + \mathcal{O}(\Delta t)^4$$

$$\mathbf{a}(t) = (-1/m)\nabla\mathcal{V}(\mathbf{r}(t))$$

$$\mathbf{v}(t) = \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t - \Delta t)}{2\Delta t}$$

## Velocity Verlet Algorithm

- Obtain velocity at half step and position at full step.

$$\mathbf{v}(t + \Delta t/2) = \mathbf{v}(t) + 1/2\mathbf{a}(t)\Delta t$$

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t + \Delta t/2)\Delta t$$

$$\mathbf{a}(t + \Delta t) = (-1/m)\nabla\mathcal{V}(\mathbf{r}(t + \Delta t))$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t + \Delta t/2) + 1/2\mathbf{a}(t + \Delta t)\Delta t$$



## Predictor Corrector Algorithm

- Predictor Step: Predict position and time derivatives time  $t + \Delta t$  by Taylor expansion at time  $t$ .
- Compute Forces as gradient of potentials at the predicted positions. The difference between the predicted acceleration and the calculated acceleration is the "error signal"

$$\Delta a(t + \Delta t) = a^C(t + \Delta t) - a^P(t + \Delta t)$$

- Corrector Step: Use the "error signal" to correct the positions and its time derivatives.

$$r^C(t + \Delta t) = r^P(t + \Delta t) + c_0 \Delta a(t + \Delta t)$$

$$v^C(t + \Delta t) = v^P(t + \Delta t) + c_1 \Delta a(t + \Delta t)$$

$$a^C(t + \Delta t) = a^P(t + \Delta t) + c_2 \Delta a(t + \Delta t)$$

$$b^C(t + \Delta t) = b^P(t + \Delta t) + c_3 \Delta a(t + \Delta t)$$

- The coefficients maximizes stability and are dependent on specific algorithm
- **Gear Predictor Corrector Algorithm:**  
 $c_0 = 1/6, c_1 = 5/6, c_2 = 1$  and  $c_3 = 1/3$ .



# Analysis of Trajectory I

- Kinetic Energy

$$K = \frac{1}{2} \sum_i m_i v_i^2$$

- Temperature: from average kinetic energy using equipartition theorem

$$T = \frac{2K}{3k_B}$$

- Pressure: from virial theorem

$$P = \frac{Nk_B T}{V} - \frac{1}{N_{dim}} \left\langle \sum_i^N \mathbf{r}_i \cdot \mathbf{F}_i \right\rangle$$

where  $N$  is number of particles and  $N_{dim}$  is dimensionality of the system.

- Diffusion Coefficient: related to mean square displacement

$$D = \frac{1}{2N_{dim}} \lim_{t \rightarrow \infty} \frac{\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle}{t}$$



- Spectral analysis
  - 1 FT-VAC: Fourier Transform of Velocity Auto-Correlation function

$$V(\omega) = \frac{1}{2\pi} \int \exp(-i\omega t) \langle \mathbf{v}(t) \cdot \mathbf{v}(0) \rangle$$

if mass weighted velocities  $\mathcal{V}_i = \sqrt{m_i} \mathbf{v}_i$  are used, then  $V(\omega)$  is the kinetic energy spectra.

- 2 FT-DAC: Fourier Transform of Dipole Auto-Correlation function, related to IR spectra

$$S(\omega) = \frac{1}{2\pi} \int \exp(-i\omega t) \langle \boldsymbol{\mu}(t) \cdot \boldsymbol{\mu}(0) \rangle$$

- 3 Short-Time Fourier Transform (STFT): 2D time-frequency spectra to simulate pump-probe experiments

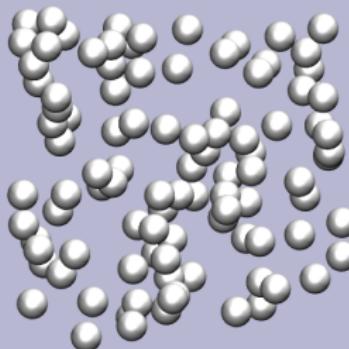
$$\mathcal{K}(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dt' \langle \mathcal{V}(t) \cdot \mathcal{V}(t') \rangle H(t, t') \exp(-i\omega t')$$

where  $H(t, t')$  is a window function



## Example code for MD

- On LONI and LSU HPC Linux systems:  
`/home/apacheco/CompChem/MD_Prog2Prod`
- Equilibration of liquid Hydrogen.
- Courtesy: Matt McKenzie, formerly LSU HPC now at NICS.
- Input File: `fort.40`
- Output File: `fort.44` (energy data) and `fort.77` (xyz file of dynamics)
- In directory `crystal`, `crystal.f90` to generate lattice structure, courtesy Furio Ercolessi <http://www.fisica.uniud.it/~ercolessi/md>



# *Ab Initio* Molecular Dynamics: Theory

- Solve the time-dependent Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{R}, \mathbf{r}, t) = \hat{H} \Psi(\mathbf{R}, \mathbf{r}, t)$$

with

$$\Psi(\mathbf{R}, \mathbf{r}, t) = \chi(\mathbf{R}, t) \Phi(\mathbf{r}, t)$$

and

$$\hat{H} = - \sum_I \frac{\hbar^2}{2M_I} \nabla_I^2 + \underbrace{\frac{-\hbar^2}{2m_e} \nabla_i^2 + V_{n-e}(\mathbf{r}, \mathbf{R})}_{H_e(\mathbf{r}, \mathbf{R})}$$

- Obtain coupled equations of motion for electrons and nuclei:  
Time-Dependent Self-Consistent Field (TD-SCF) approach.

$$i\hbar \frac{\partial \Phi}{\partial t} = \left[ - \sum_i \frac{\hbar^2}{2m_e} \nabla_i^2 + \langle \chi | V_{n-e} | \chi \rangle \right] \Phi$$

$$i\hbar \frac{\partial \chi}{\partial t} = \left[ - \sum_I \frac{\hbar^2}{2M_I} \nabla_I^2 + \langle \Phi | H_e | \Phi \rangle \right] \chi$$



- Define nuclear wavefunction as

$$\chi(\mathbf{R}, t) = A(\mathbf{R}, t) \exp [iS(\mathbf{R}, t)/\hbar]$$

where  $A$  and  $S$  are real.

- Solve the time-dependent equation for nuclear wavefunction and take classical limit ( $\hbar \rightarrow 0$ ) to obtain

$$\frac{\partial S}{\partial t} + \sum_I \frac{\hbar^2}{2M_I} (\nabla_I S)^2 + \langle \Phi | H_e | \Phi \rangle = 0$$

an equation that is isomorphic with the Hamilton-Jacobi equation with the classical Hamilton function given by

$$\mathcal{H}(\{\mathbf{R}_I\}, \{\mathbf{P}_I\}) = \sum_I \frac{\hbar^2}{2M_I} \mathbf{P}_I^2 + V(\{\mathbf{R}_I\})$$

where

$$\mathbf{P}_I \equiv \nabla_I S \quad \text{and} \quad V(\{\mathbf{R}_I\}) = \langle \Phi | H_e | \Phi \rangle$$

- Obtain equations of nuclear motion from Hamilton's equation

$$\frac{d\mathbf{P}_I}{dt} = -\frac{d\mathcal{H}}{d\mathbf{R}_I} \Rightarrow M \ddot{\mathbf{R}}_I = -\nabla_I V$$

$$\frac{d\mathbf{R}_I}{dt} = \frac{d\mathcal{H}}{d\mathbf{P}_I}$$



- Replace nuclear wavefunction by delta functions centered on nuclear position to obtain

$$i\hbar \frac{\partial \Phi}{\partial t} = H_e(\mathbf{r}, \{\mathbf{R}_I\})\Phi(\mathbf{r}; \{\mathbf{R}_I\}, t)$$

- This approach of simultaneously solving the electronic and nuclear degrees of freedom by incorporating feedback in both directions is known as **Ehrenfest Molecular Dynamics**.
- Expand  $\Phi$  in terms of many electron wavefunctions or determinants

$$\Phi(\mathbf{r}; \{\mathbf{R}_I\}, t) = \sum_i c_i(t)\Phi_i(\mathbf{r}; \{\mathbf{R}_I\})$$

with matrix elements

$$H_{ij} = \langle \Phi_i | H_e | \Phi_j \rangle$$

- Inserting  $\Phi$  in the TDSE above, we get

$$i\hbar \dot{c}_i(t) = c_i(t)H_{ii} - i\hbar \sum_{I,i} \dot{\mathbf{R}}_I \mathbf{d}_I^{ij}$$

with non-adiabatic coupling elements given by

$$\mathbf{d}_I^{ij}(\mathbf{R}_I) = \langle \Phi_i | \nabla_I | \Phi_j \rangle$$



- Up to this point, no restriction on the nature of  $\Phi_i$  i.e. adiabatic or diabatic basis has been made.
- Ehrenfest method rigorously includes non-adiabatic transitions between electronic states within the framework of classical nuclear motion and mean field (TD-SCF) approximation to the electronic structure.
- Now suppose, we define  $\{\Phi_i\}$  to be the adiabatic basis obtained from solving the time-independent Schrödinger equation,

$$H_e(\mathbf{r}, \{\mathbf{R}_I\})\Phi_i(\mathbf{r}; \{\mathbf{R}_I\}) = E_i(\{\mathbf{R}_I\})\Phi_i(\mathbf{r}; \{\mathbf{R}_I\})$$

- The classical nuclei now move along the adiabatic or Born-Oppenheimer potential surface. Such dynamics are commonly known as **Born-Oppenheimer Molecular Dynamics** or BOMD.
- If we restrict the dynamics to only the ground electronic state, then we obtain ground state BOMD.
- If the Ehrenfest potential  $V(\{\mathbf{R}_I\})$  is approximated to a global potential surface in terms of many-body contributions  $\{v_n\}$ .

$$V(\{\mathbf{R}_I\}) \approx V_e^{approx}(\mathbf{R}) = \sum_{I=1}^N v_1(\mathbf{R}_I) + \sum_{I>J}^N v_2(\mathbf{R}_I, \mathbf{R}_J) + \sum_{I>J>K}^N v_3(\mathbf{R}_I, \mathbf{R}_J, \mathbf{R}_K) + \dots$$



- Thus the problem is reduced to purely classical mechanics once the  $\{v_n\}$  are determined usually Molecular Mechanics Force Fields. This class of dynamics is most commonly known as **Classical Molecular Dynamics**.
- Another approach to obtain equations of motion for ab-initio molecular dynamics is to apply the Born-Oppenheimer approximation to the full wavefunction  $\Psi(\mathbf{r}, \mathbf{R}, t)$

$$\Psi(\mathbf{r}, \mathbf{R}, t) = \sum_k \chi_k(\mathbf{R}, t) \Phi_k(\mathbf{r}; \mathbf{R}(t))$$

where

$$H_e \Phi_k(\mathbf{r}; \mathbf{R}(t)) = E_k(\mathbf{R}(t)) \Phi_k(\mathbf{r}; \mathbf{R}(t))$$

- Assuming that the nuclear dynamics does not change the electronic state, we arrive at the equation of motion for nuclear wavefunction

$$i\hbar \frac{\partial}{\partial t} \chi_k(\mathbf{R}, t) = \left[ \sum_I -\frac{\hbar^2}{2M_I} \nabla_I^2 + E_k(\mathbf{R}) \right] \chi_k(\mathbf{R}, t)$$



- The Lagrangian for this system is given by.

$$\mathcal{L} = \hat{T} - \hat{V}$$

- Corresponding Newton's equation of motion are then obtained from the associated Euler-Lagrange equations,

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{R}}_I} = \frac{\partial \mathcal{L}}{\partial \mathbf{R}_I}$$

- The Lagrangian for ground state BOMD is

$$\mathcal{L}_{\text{BOMD}} = \sum_I \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 - \min_{\Phi_0} \langle \Phi | H_e | \Phi \rangle$$

and equations of motions

$$M_I \ddot{\mathbf{R}}_I = \frac{d}{dt} \frac{\partial \mathcal{L}_{\text{BOMD}}}{\partial \dot{\mathbf{R}}_I} = \frac{\partial \mathcal{L}_{\text{BOMD}}}{\partial \mathbf{R}_I} = -\nabla_I \min_{\Phi_0} \langle \Phi | H_e | \Phi \rangle$$

### Extended Lagrangian Molecular Dynamics (ELMD)

Extend the Lagrangian by adding kinetic energy of fictitious particles and obtain their equation of motions from Euler-Lagrange equations.



Molecular Orbitals:  $\{\phi_i\}$   
 Density Matrix:  $P_{\mu\nu} = \sum_i c_{\mu i}^* c_{\nu i}$

## Car-Parrinello Molecular Dynamics (CPMD)

### CPMD and NWChem

$$\mathcal{L}_{\text{CPMD}} = \sum_I \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 + \sum_i \frac{1}{2} \mu_i \langle \dot{\phi}_i | \dot{\phi}_i \rangle - \langle \Phi_0 | H_e | \Phi_0 \rangle + \text{constraints}$$

---

R. Car and M. Parrinello, Phys. Rev. Lett. 55 (22), 2471 (1985)

## Atom centered Density Matrix Propagation (ADMP)

### Gaussian 03/09

$$\mathcal{L}_{\text{ADMP}} = \frac{1}{2} \text{Tr}(\mathbf{V}^T \mathbf{M} \mathbf{V}) + \frac{1}{2} \mu \text{Tr}(\dot{\mathbf{P}} \dot{\mathbf{P}}) - E(\mathbf{R}, \mathbf{P}) - \text{Tr}[\boldsymbol{\Lambda}(\mathbf{P} \mathbf{P} - \mathbf{P})]$$

---

H. B. Schlegel, J. M. Millam, S. S. Iyengar, G. A. Voth, A. D. Daniels, G. E. Scuseria, M. J. Frisch, J. Chem. Phys. 114, 9758 (2001)

## curvy-steps ELMD (csELMD)

### Q-Chem

$$\mathcal{L}_{\text{csELMD}} = \sum_I \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 + \frac{1}{2} \mu \sum_{i < j} \dot{\Delta}_{ij} - E(\mathbf{R}, \mathbf{P}); \quad \mathbf{P}(\lambda) = e^{\lambda \Delta} \mathbf{P}(0) e^{-\lambda \Delta}$$

---

J.M. Herbert and M. Head-Gordon, J. Chem. Phys. 121, 11542 (2004)



CENTER FOR COMPUTATION  
& TECHNOLOGY

# Molecular Dynamics: Methods and Programs

- Electronic energy obtained from
  - Molecular Mechanics ⇒ Classical Molecular Dynamics
    - 1 LAMMPS
    - 2 NAMD
    - 3 Amber
    - 4 Gromacs
  - Ab-Initio Methods ⇒ Quantum or Ab-Initio Molecular Dynamics
    - 1 Born-Oppenheimer Molecular Dynamics: Gaussian, GAMESS
    - 2 Extended Lagrangian Molecular Dynamics: CPMD, Gaussian (ADMP), NWChem(CPMD), VASP, QChem (curvy-steps ELMD)
    - 3 Time Dependent Hartree-Fock and Time Dependent Density Functional Theory: Gaussian, GAMESS, NWChem, QChem
    - 4 Multiconfiguration Time Dependent Hartree(-Fock), MCTDH(F)
    - 5 Non-Adiabatic and Ehrenfest Molecular Dynamics, Multiple Spawning, Trajectory Surface Hopping
    - 6 Quantum Nuclei: QWAIMD(Gaussian), NEO(GAMESS)



## Classical Molecular Dynamics

- Advantages
  - ① Large Biological Systems
  - ② Long time dynamics
- Disadvantages
  - ③ Cannot describe Quantum Nuclear Effects

## *Ab Initio* and Quantum Dynamics

- Advantages
  - ④ Quantum Nuclear Effects
- Disadvantages
  - ⑤ ~ 100 atoms
  - ⑥ Full Quantum Dynamics ie treating nuclei quantum mechanically:  
less than 10 atoms
  - ⑦ Picosecond dynamics at best



# Software Installed on LONI & LSU HPC Systems

| Software | QB | Eric | Louie | Oliver | Painter | Poseidon | Philip | Tezpur |
|----------|----|------|-------|--------|---------|----------|--------|--------|
| Amber    | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        | ✓      | ✓      |
| Desmond  | ✓  |      |       |        |         |          |        |        |
| DL_Poly  | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        | ✓      | ✓      |
| Gromacs  | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        | ✓      | ✓      |
| LAMMPS   | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        | ✓      | ✓      |
| NAMD     | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        | ✓      | ✓      |
| CPMD     | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        |        | ✓      |
| GAMESS   | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        | ✓      | ✓      |
| Gaussian |    | ✓    | ✓     | ✓      | ✓       |          | ✓      | ✓      |
| NWCHEM   | ✓  | ✓    | ✓     | ✓      | ✓       | ✓        |        | ✓      |

| Software | Bluedawg | Ducky | Lacumba | Neptune | Zeke | Pelican | Pandora |
|----------|----------|-------|---------|---------|------|---------|---------|
| Amber    |          | ✓     | ✓       |         |      | ✓       | ✓       |
| Gromacs  | ✓        | ✓     | ✓       | ✓       | ✓    | ✓       | ✓       |
| LAMMPS   | ✓        | ✓     | ✓       | ✓       | ✓    | ✓       | ✓       |
| NAMD     | ✓        | ✓     | ✓       | ✓       | ✓    | ✓       | ✓       |
| CPMD     | ✓        | ✓     | ✓       | ✓       | ✓    |         | ✓       |
| GAMESS   |          |       |         |         |      |         | ✓       |
| Gaussian | ✓        | ✓     | ✓       |         | ✓    | ✓       | ✓       |
| NWCHEM   | ✓        | ✓     | ✓       | ✓       | ✓    | ✓       | ✓       |



**LSU**

CENTER FOR COMPUTATION  
& TECHNOLOGY

## Computational Chemistry Programs

- Other Software: Q-Chem, SIESTA ,CHARMM, VASP, Quantum Expresso, Octopus
- [http://en.wikipedia.org/wiki/Quantum\\_chemistry\\_computer\\_programs](http://en.wikipedia.org/wiki/Quantum_chemistry_computer_programs)
- <http://www.ccl.net/chemistry/links/software/index.shtml>
- <http://www.redbrick.dcu.ie/~noel/linux4chemistry/>



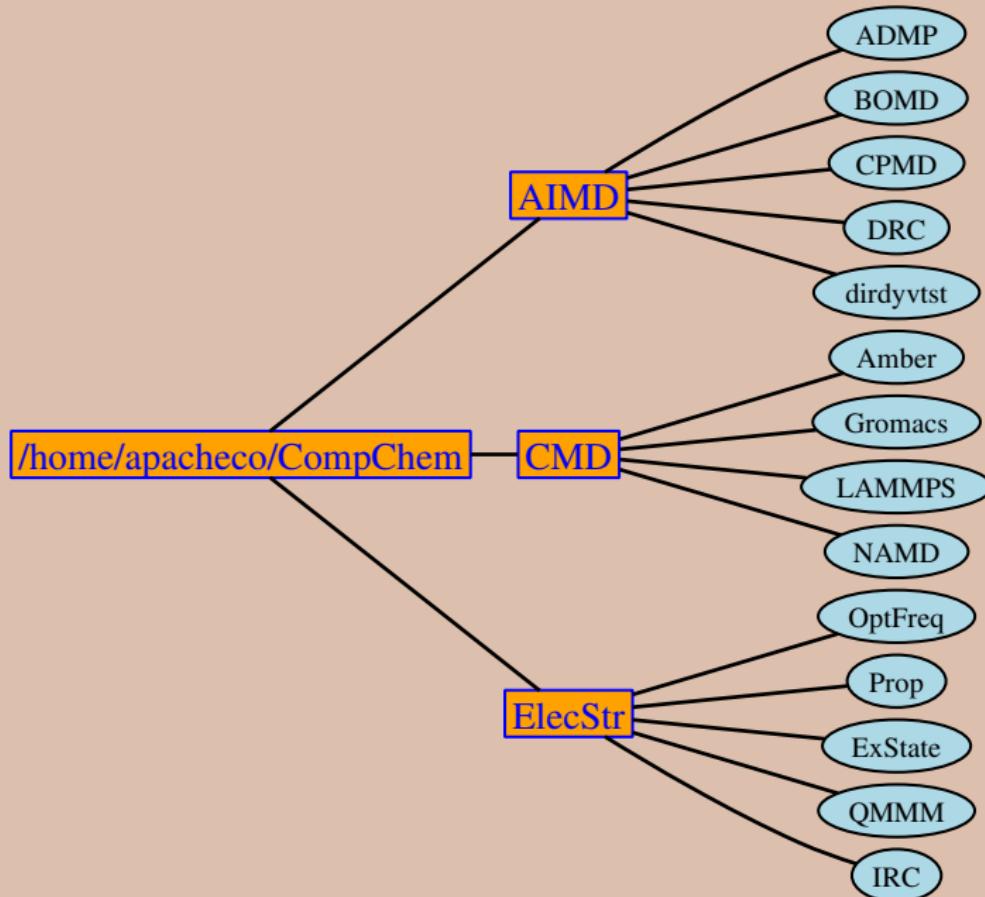
## Molecular Dynamics Calculations

- CPMD:
  - CP: Car-Parrinello Molecular Dynamics
  - BO: Born-Oppenheimer Molecular Dynamics
- Gaussian:
  - BOMD: Born-Oppenheimer Molecular Dynamics
  - ADMP: Atom centered Density Matrix Propagation and ground state BOMD
- GAMESS:
  - DRC: Direct Dynamics, a classical trajectory method based on "on-the-fly" ab-initio or semi-empirical potential energy surfaces
- NWChem:
  - Car-Parrinello: Car Parrinello Molecular Dynamics (CPMD)
  - DIRDYVTST: Direct Dynamics Calculations using POLYRATE with electronic structure from NWChem



CENTER FOR COMPUTATION  
& TECHNOLOGY

# On LONI/LSU HPC Linux Systems



## Using LAMMPS on LONI systems

- LAMMPS stands for Large-scale Atomic/Molecular Parallel Simulator.
- LAMMPS is a classical molecular dynamics code that models an ensemble of particles in a liquid, solid, or gaseous state designed to run efficiently on parallel computers.
- It can model atomic, polymeric, biological, metallic, granular, and coarse-grained systems using a variety of force fields and boundary conditions.
- LAMMPS can model systems with only a few particles up to millions or billions.
- LAMMPS is designed to be easy to modify or extend with new capabilities, such as new force fields, atom types, boundary conditions, or diagnostics.
- LAMMPS runs efficiently on single-processor desktop or laptop machines, but is designed for parallel computers.
- It is an open-source code, distributed freely under the terms of the GNU Public License (GPL).



- LAMMPS doesn't
  - ① Build molecular systems
  - ② Assign force-field coefficients auto-magically
  - ③ Compute lots of diagnostics on-the-fly
  - ④ Visualize your output
- LAMMPS version "4 May 2011" is installed on all LONI Dell Linux Clusters and LSU HPC machines, Tezpur (Linux) and Pandora (AIX).
- Add the appropriate soft keys to your .soft file
  - ① Linux: +lammps-4May11-intel-11.1-mvapich-1.1
  - ② Pandora: +lammps-4May11
- Running LAMMPS
  - ◆ Linux: mpirun -np \${no of procs} {LAMMPS directory}/bin/lmp\_linux < inputfile
  - ◆ Pandora: poe {LAMMPS directory}/bin/lmp\_power7 < inputfile
- Command line options
  - ◆ -in inputfile: specify input file
  - ◆ -log logfile: specify log file
  - ◆ -partition MxN L: Run on (MxN)+L processors with M partitions on N processors each and 1 partition with L processors
  - ◆ -screen file: Specify a file to write screen information



- Reads an input script in ASCII format one line at a time.
- Input script consists of 4 parts
  - 1 Initialization: Set parameters that need to be defined before atoms are created or read-in from a file.  
units, dimension, newton, processors, boundary,  
atom\_style, atom\_modify
  - 2 Atom definition: read\_data, read\_restart, lattice, region,  
create\_box, create\_atoms, replicate
  - 3 Settings: Once atoms and molecular topology are defined, a variety of  
settings can be specified: force field coefficients, simulation parameters,  
output options, etc.  
pair\_coeff, bond\_coeff, angle\_coeff, dihedral\_coeff,  
improper\_coeff, kspace\_style, dielectric,  
special\_bonds, neighbor, neigh\_modify, group,  
timestep, reset\_timestep, run\_style, min\_style,  
min\_modify, fix, compute, compute\_modify, variable
  - 4 Run a simulation: A MD is run using the run command. Energy  
minimization (molecular statics) is performed using the minimize  
command. A parallel tempering (replica-exchange) simulation can be run  
using the temper command.

• [http://lammps.sandia.gov/doc/Section\\_commands.html](http://lammps.sandia.gov/doc/Section_commands.html)

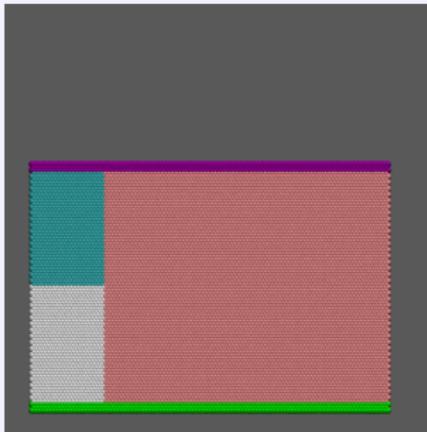


- "dump" command outputs snapshots of atom properties
  - default format is simple : id, type, x, y, z
  - other supported formats : XYZ, DCD, XTC
- Visualization using VMD

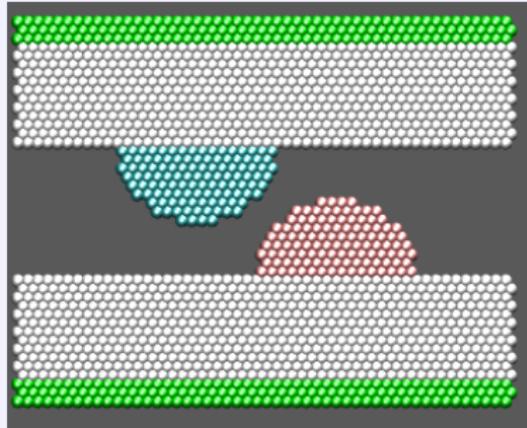
```
vmd -lammpstrj dumpfile
```



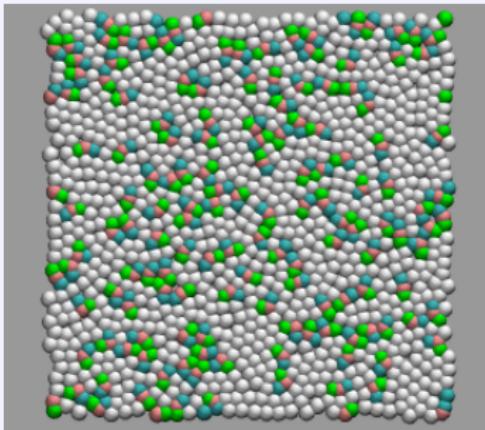
- The LAMMPS distribution includes an examples sub-directory with several sample problems.
- crack: crack propagation in a 2d solid



- The LAMMPS distribution includes an examples sub-directory with several sample problems.
- friction: frictional contact of spherical asperities between 2d surfaces

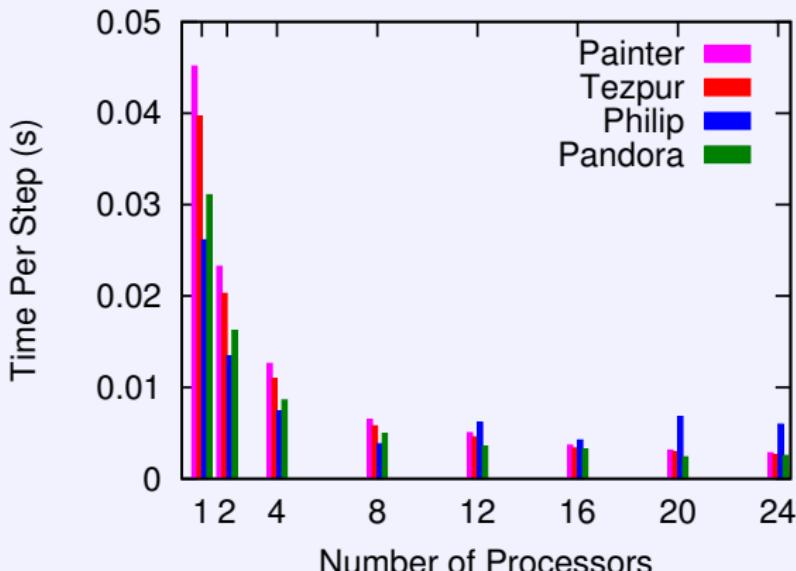


- The LAMMPS distribution includes an examples sub-directory with several sample problems.
- micelle: self-assembly of small lipid-like molecules into 2d bilayers



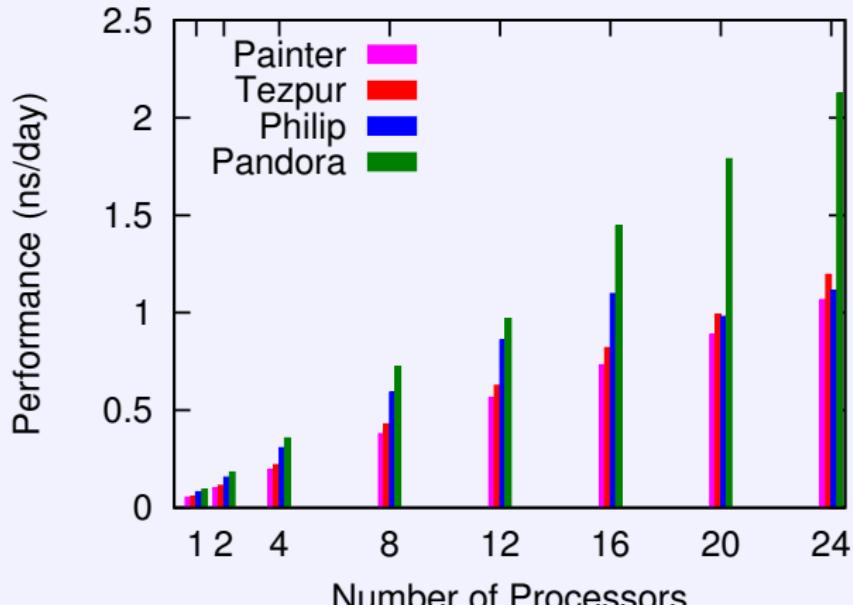
# Benchmarks

- LAMMPS on LONI & LSU HPC machines
  - ◆ Atomic fluid with Lennard-Jones Potential
  - ◆ 32,000 atoms for 100,000 steps
  - ◆ force cutoff =  $2.5\sigma$ , neighbor skin =  $0.3\sigma$ , neighbor/atom = 55
  - ◆ NVE time integration



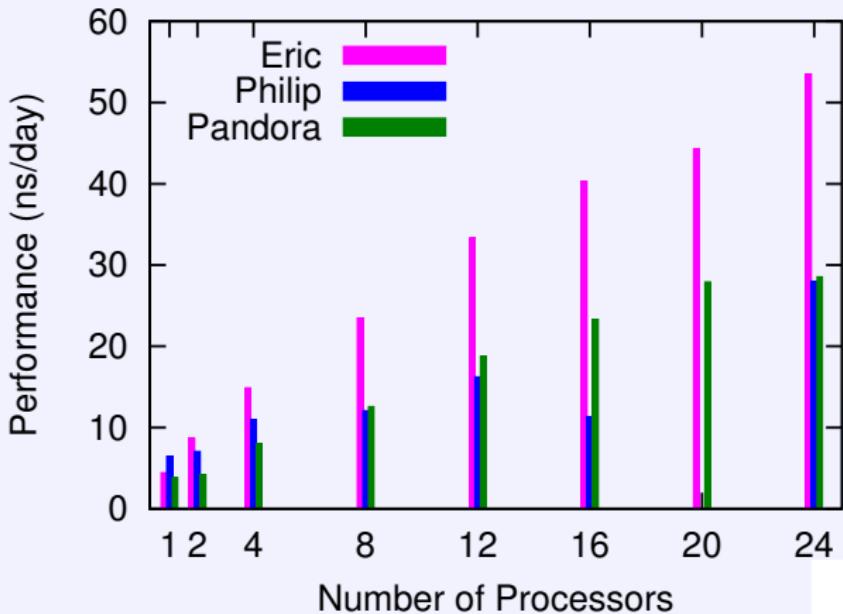
# Benchmarks

- NAMD on LONI & LSU HPC machines
  - ◆ Apoa1 Benchmark
  - ◆ 92,224 atoms, 12Å cutoff + PME every 4 steps, periodic



# Benchmarks

- Gromacs on LONI & LSU HPC machines
  - ◆ 159 residue protein dihydrofolate reductase (dhfr) with implicit solvent
  - ◆ 2489 atoms, 1nm cutoff



# Molecular Dynamics on GPU

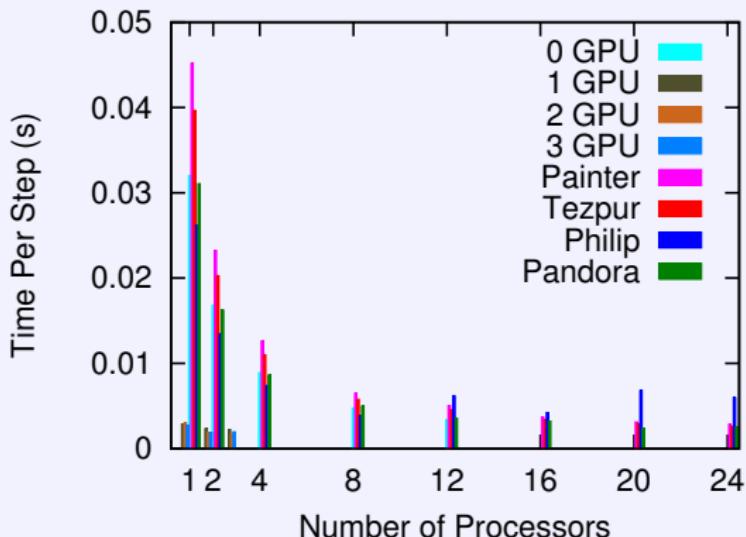
- Using a graphics processing unit (GPU) for molecular simulations.
- MD programs capable of GPU computing
  - 1 LAMMPS
  - 2 NAMD
  - 3 GROMACS
  - 4 Amber/PMEMD
- Not all features of the MD programs are capable of GPU computing.



- Coming Soon: 2 Philip nodes with 3 GPU's each.
  - Intel Xeon X5650
    - ◆ 2.66GHz Dual Hexa-core with hyperthreading: 12 cores, 24 threads
    - ◆ Memory : 48GB
    - ◆ L3 Cache: 12MB
  - Tesla M2070
    - ◆ 448 CUDA cores (14 Multiprocessor  $\times$  32 CUDA Cores/MP )
    - ◆ GPU Clock Speed: 1.15GHz
    - ◆ Total Memory: 5.25GB
- Compilers: CUDA 4.0, PGI Accelerator
- Infiniband Interconnect between the two nodes.
- For optimum scaling, ratio for CPU core:GPU should be 1.
- LAMMPS and NAMD can bind GPU to multiple cores but performance degrades.



- Need to use `newton off`
- `fix_gpu` should be used in order to initialize and configure the GPUs for use
  - fix ID group-ID gpu mode first last split  
mode can be force or force/neigh
- currently limited to a few pair\_styles and PPPM



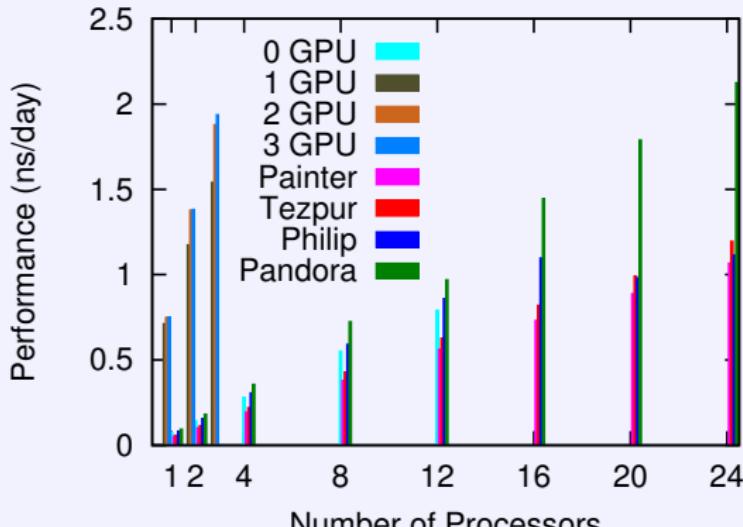
- command line options to add

```
+idlepoll +devices 0,1
```

If `+devices` option is not added then all available GPU's will be used

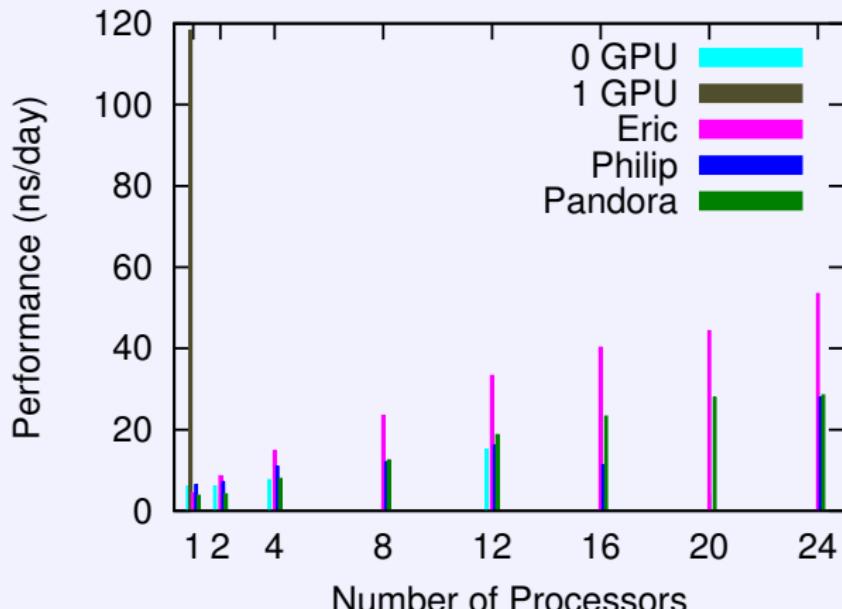
Each CPU will be assigned a GPU cyclically

i.e. `mpirun -np 4 namd2 +idlepoll +devices 0,1` will bind  
CPU 0 and 2 to GPU device 0 and CPU 1 to GPU device 1



## Gromacs on GPU

- Use mdrun compiled with cuda: mdrun-gpu
- mdrun-gpu can only run on 1 GPU, multiple GPUs functionality not available yet
- Usage: same as serial or parallel mdrun



# Useful Links

- Amber:<http://ambermd.org>
- Desmond:[http://www.deshawresearch.com/resources\\_desmond.html](http://www.deshawresearch.com/resources_desmond.html)
- DL\_POLY:[http://www.cse.scitech.ac.uk/ccg/software/DL\\_POLY](http://www.cse.scitech.ac.uk/ccg/software/DL_POLY)
- Gromacs:<http://www.gromacs.org>
- LAMMPS:<http://lammps.sandia.gov>
- NAMD:<http://www.ks.uiuc.edu/Research/namd>
- CPMD:<http://www.cpmd.org>
- GAMESS:<http://www.msg.chem.iastate.edu/gamess>
- Gaussian:<http://www.gaussian.com>
- NWChem:<http://www.nwchem-sw.org>
- PINY\_MD:[http://homepages.nyu.edu/~mt33/PINY\\_MD/PINY.html](http://homepages.nyu.edu/~mt33/PINY_MD/PINY.html)
- Basis Set:<https://bse.pnl.gov/bse/portal>



## Further Reading

- A Molecular Dynamics Primer by Furio Ercolelli  
<http://www.fisica.uniud.it/~ercolelli/md>
- Molecular Modeling - Principles and Applications, A. R. Leach.
- Computer Simulation of Liquids, M. P. Allen and D. J. Tildesley.
- Mark Tuckerman's Notes at NYU:  
<http://www.nyu.edu/classes/tuckerman/quant.mech/index.html>
- Ab Initio Molecular Dynamics: Theory and Implementation, D. Marx and J. Hutter  
<http://www.theochem.ruhr-uni-bochum.de/research/marx/marx.pdf>
- Quantum Dynamics with Trajectories: Introduction to Quantum Hydrodynamics, R. E. Wyatt.
- Quantum Dynamics of Complex Molecular Systems, Editors: D. A. Micha and I. Burghardt
- Energy Transfer Dynamics in Biomaterial Systems. Editors: I. Burghardt, V. May, D. A. Micha and E. R. Bittner .

