Title: Using GPUs to accelerate molecular dynamics simulations

Date: Jun 23, 2011 04:50 PM

URL: http://pirsa.org/11060039

Abstract: GPUs can offer a less costly solution to large-scale calculations of astrophysical systems. I will outline the basics of the CUDA libraries and also compare with various metrics our in-development GPU code for molecular dynamics versus our hybrid OpenMP/MPI version.

Molecular Dynamics

- We use a screened Yukawa potential, $V(r) \propto \frac{e^{-r/\lambda}}{r}$
- λ is the Thomas-Fermi screening length, $\lambda^{-1} = 2\alpha^{1/2} k_F / \pi^{1/2}$
- Atoms are completely pressure ionized so $n_e = Zn$ and the electrons form a degenerate Fermi gas
- Simulations are characterized by the Coulomb Parameter

$$\Gamma = \frac{Z^2 e^2}{aT} \qquad (1$$

Molecular Dynamics

- Use Velocity Verlet algorithm
  - $\vec{v}(t) \Rightarrow \vec{v}(t + \frac{1}{2}\Delta t)$ - Update velocity half-step
  - $\vec{x}(t) \Rightarrow \vec{x}(t + \Delta t)$ - Update positions
  - $\vec{a}(t) \Rightarrow \vec{a}(t + \Delta t)$ - Force calculation
  - $\vec{v}(t + \frac{1}{2}\Delta t) \Rightarrow \vec{v}(t + \Delta t)$ - Update velocity half-step
- Global Error of order Δt^2

Simulations

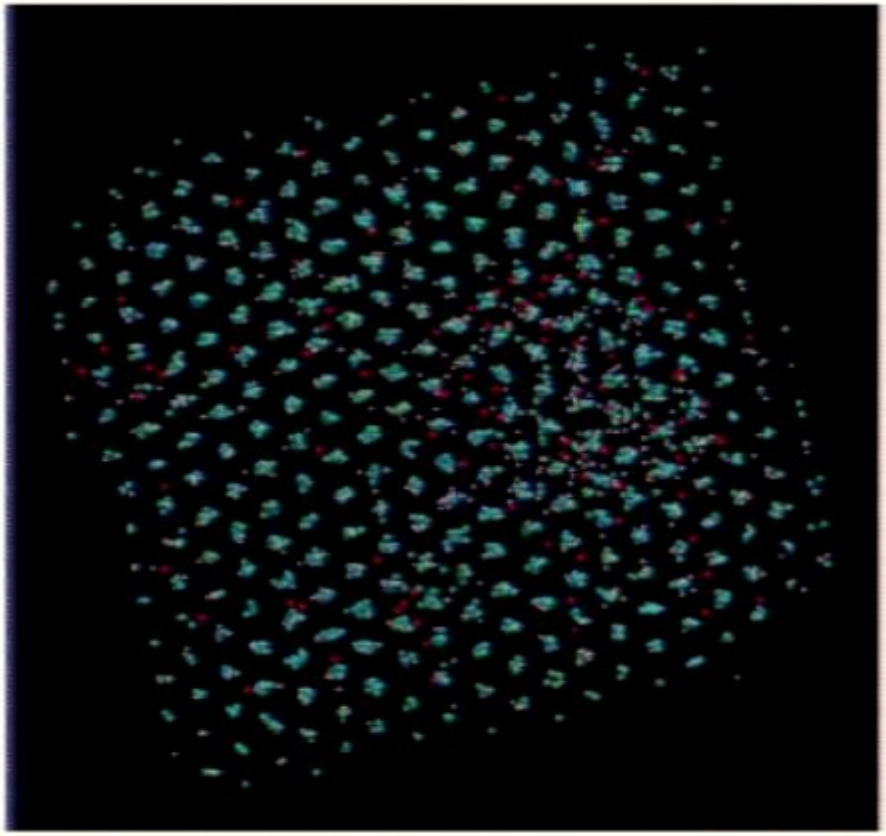


Figure: Horowitz, Caballero, Berry (2009)

- MD simulations can be used to model a variety of microphysical processes
  - Thermal conductivity
  - Shear modulus/breaking strain
  - Phase separation
  - Diffusion in liquid and solid
  - Liquid/solid phase diagram
  - ...

Thermal Conductivity

- Thermal conductivity, κ, is dominated by heat carried by electrons
- Heat is transferred mainly by inelastic, e.g. ion-electron, collisions
- We only need to compute the structure factor $S(q)$ for the system
- $S(q)$ includes both Bragg scattering contributions and inelastic excitation contributions, $S(q) = S'(q) + S_{Bragg}(q)$

Thermal Conductivity

- This high thermal conductivity explains NS cooling curves, e.g. KS1731-260
- κ is low in amorphous solids
- NS crusts are most likely crystalline and not amorphous

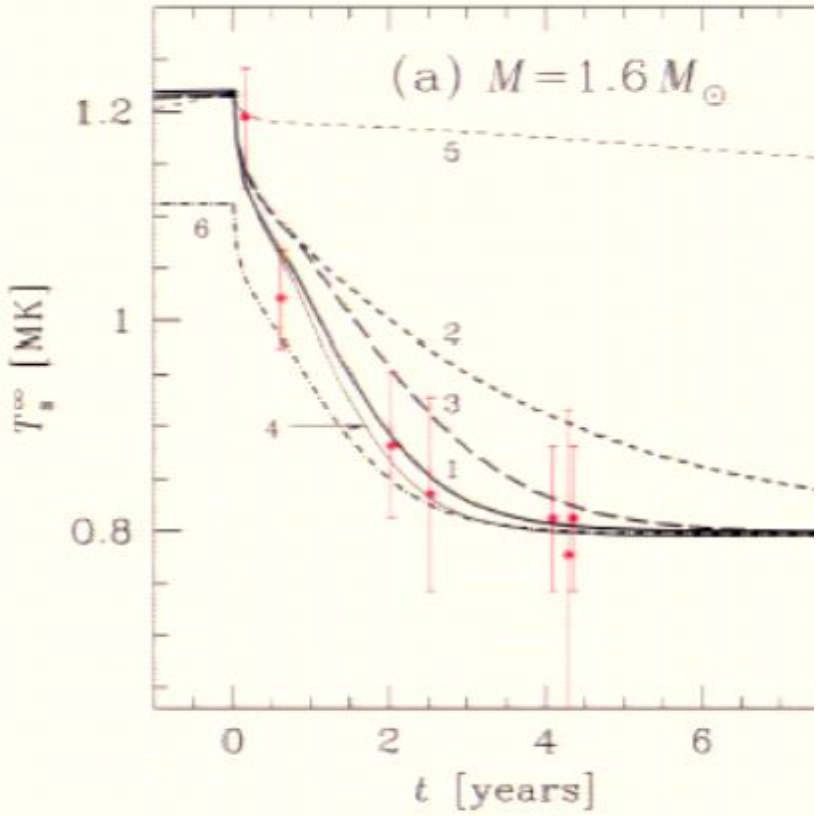


Figure: Shternin et al. (2007)

Shear Modulus

- Measuring the system response to deformations leads to shear modulus
- We use the prescription of Ogata et al. (1990) which deformed the system and calculated elastic constants
- Shear modulus is sensitive to long tails of our potential
- These calculations have only recently become accessible

Hybrid OpenMP/MPI

- OpenMP and MPI meet our needs
- MPI
  - One MPI rank per node
  - Copies data between nodes
- OpenMP
  - Four (Big Red) to twelve (Kraken) OpenMP threads per nod
  - Computes forces and updates positions and velocities

Hybrid OpenMP/MPI

- Hybrid scheme lets us reduce the number of messages by a factor equal to the number of cores
- Lower communication overhead allows for nice scalability
- 27,648 ion systems run above 85% efficiency on 768 cores

GPU Computing

- Much of a CPU die is filled with circuits that do not perform FLOPs
- CPUs are very flexible and save time with advanced logic, e.g. algorithm prediction
- Multicore era has begun but clock speeds are remaining constant (or decreasing)
- Designed to run a single thread very quickly

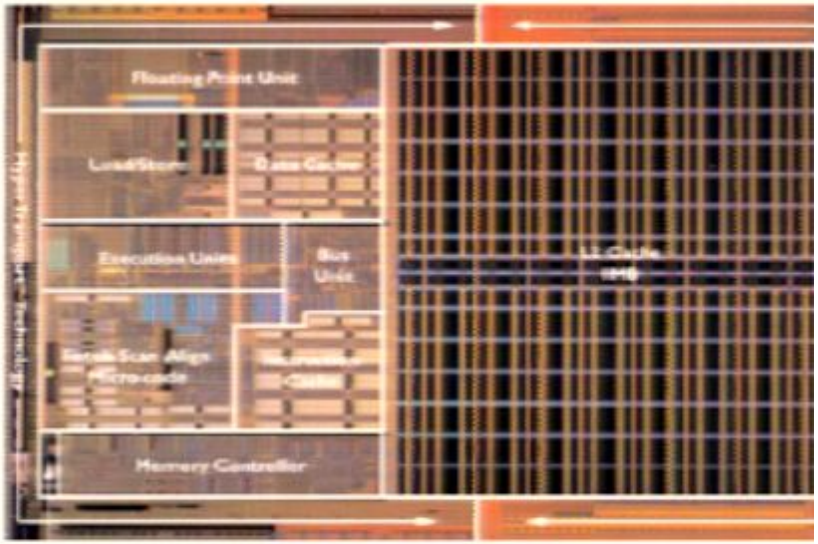
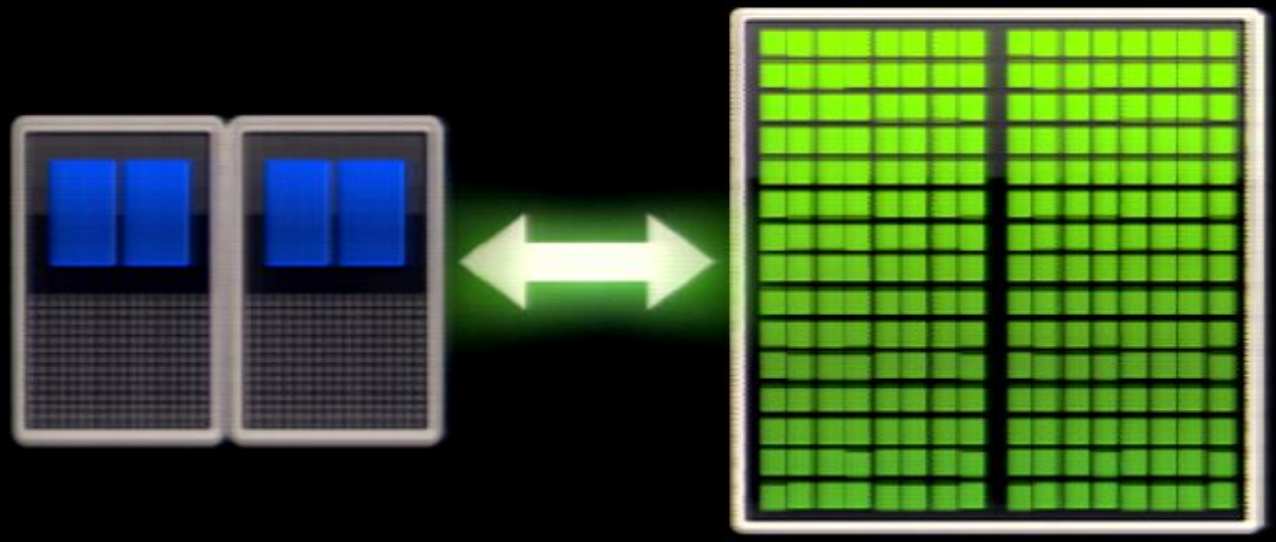


Figure: Standard CPU layout

GPU Computing

- GPUs have more die space allocated to computational circuitry
- More parallel 'multiprocessors' with (small) on-chip memory
- (large) Shared global memory space
- Basic idea: create/run many concurrent threads that can hide latency
- Thousands of threads are required to keep GPU fully utilize

Heterogeneous Parallel Computing



Multicore CPU

Fast Serial Processing

Manycore GPU

Scalable Parallel Processing

CUDA

- NVIDIA provides CUDA (Compute Unified Device Architecture) libraries for C
- Fortran support is available (for the moment) using PGI compilers
- CUDA does not work for devices from other vendors, i.e. AMD/ATI

Standard CUDA Workflow

- Copy data from CPU to device
  - GPU is connected over PCI bus
  - Data copy can be quite slow
- CPU starts kernel on GPU
  - Once kernel is launched, CPU can be active again
  - Allows another level of parallism
- GPU divvies up task to run
  - Kernel is called with dimensionality of task
  - CUDA takes care of any further system calls
- Copy data from GPU to CPU

CUDA Inner Workings

- Threads executed in 'thread blocks' which share a local memory space (16kB-48kB)
- Synchronization protocols exist to allow threads to modify local shared memory
- Data fetch from local/shared memory takes $\sim$1 clock cycle
- Fetch from global DRAM takes $\sim$100 clock cycles

Porting Legacy Code

- Different approach means a different way of thinking
- No communication between thread blocks
- Data structures must be reworked to work well with GPUs
- Only algorithms that could benefit from GPU style of parallelism need to ported

Should Your Code Run on GPUs?

- 'Embarrassingly parallel' routines fit well on GPUs
- Algorithms that require a high level of inter-communication need to be modified
- Routines somewhere in the middle can run some on the GPU, some on the CPU
- Local working data sets of $\sim$10 kB can really benefit from fast memory accesses
- I/O bound problems should be handled by convential CPUs

Summary - GPU Computing

- Many multiprocessors allow for high level of parallelization
- Large codes will need to implement CUDA routines along with MPI calls
- New large hybrid CPU/GPU systems coming online
- 3 of 5 systems in Top 500 include GPUs

Summary - Molecular Dynamics

- Molecular dynamics simulations can be used to study microphysics of the crusts of neutron stars and the interior o white dwarfs
- To increase system size and simulation times, large parallel codes are needed
- Hybrid OpenMP/MPI approach works well, but has a high cost (i.e. SUs)
- GPUs offer cheaper solution, but with high initial time investment

Future Work

- Magnetic fields
  - Method involves putting the B field into the equation of motion
  - Can now compute shear modulus, breaking strain, diffusion, etc.
  - *Preliminary:* Magnetic fields modify the trajectories, but no closed orbits
- Shear modulus/breaking strain for pasta phases
- Magnetothermal evolution of NS crust w/ José Pons