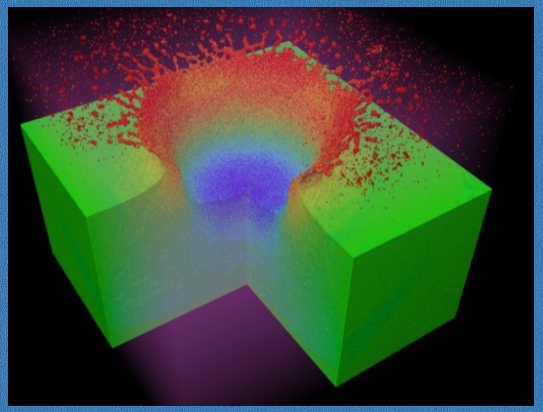
Introduction to Molecular Dynamics (MD) Simulations

Part I: Basics (b)



Eduardo M. Bringa

ebringa@yahoo.com

CONICET

FING, Universidad de Mendoza, Argentina

https://sites.google.com/site/simafweb

Funding:

Agencia CyT, Argentina

FING, U Mendoza

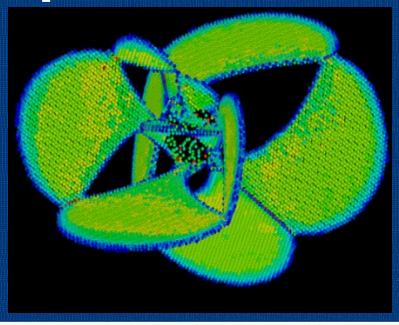
SIIP, UN Cuyo

Aplicaciones de simulaciones y programación CUDA: estudios a escala atómica, fenómenos climáticos y ópticos

Centro de Investigaciones Hidráulicas e Hidrotécnicas (CIHH)

UTP, Panamá

September 2019



COLABORATIONS: C. Ruestes, E. Millan (UN Cuyo), **B. Planes, J. Grossi, A. Araguna, E. Aparicio, G. Mora,** E. Miranda **(U. Mendoza)**, N. Gunkelmann, C. Anders, H. Urbassek (TU Kaiserslautern), R. Gonzalez-Valdez (U. Mayor, Chile), F. Valencia, M. Kiwi, F. Munoz, E. Figueroa, S. Davis, G. Gutierrez (U. Chile), M.A. Meyers, Y. Tang, E. Hahn, S. Zhao, K. Olney, D. Benson (UCSD), B. Remington, J. H G. Moraawreliak, R. Rudd (LLNL), M. Ruda, G. Bertolino (Instituto Balseiro, Argentina), A. Stukowski (TU Darmstadt, Germany), P. Erhart (Chalmers U., Sweden), A. Higginbotham, J. Wark (University of Oxford, UK), R. Gonzales, A. Rivera, A. Prada (UP Madrid), S. Ramos, E. Crespo (U. Comahue), J. Kohanoff (QUB, N. Ireland).

SiMAF

Why classical MD?

- Atoms move!
  - We may be interested in studying time dependent phenomena, such as molecular vibrations, phonons, diffusion, etc.
  - We may be interested in studying temperature dependant phenomena, such as free energies, anharmonic effects,
  - Ergodic Hypothesis: Time average over trajectory is equivalent to an ensemble average ➔ Allows the use of MD for statistical mechanics studies.
  - Mechanical properties, radiation damage, “simple” chemistry.
- New hardware and software allow direct comparison with recent experiments

Atomistic simulations are extremely helpful but … still have multiple limitations

With MD you can obtain….

"Real" time evolution of your system.

Thermodynamic properties, including T(r,t) temperature profiles that can be used in rate equations.

Mechanical properties, including elastic and plastic behavior.

Surface/bulk/cluster growth and modification.

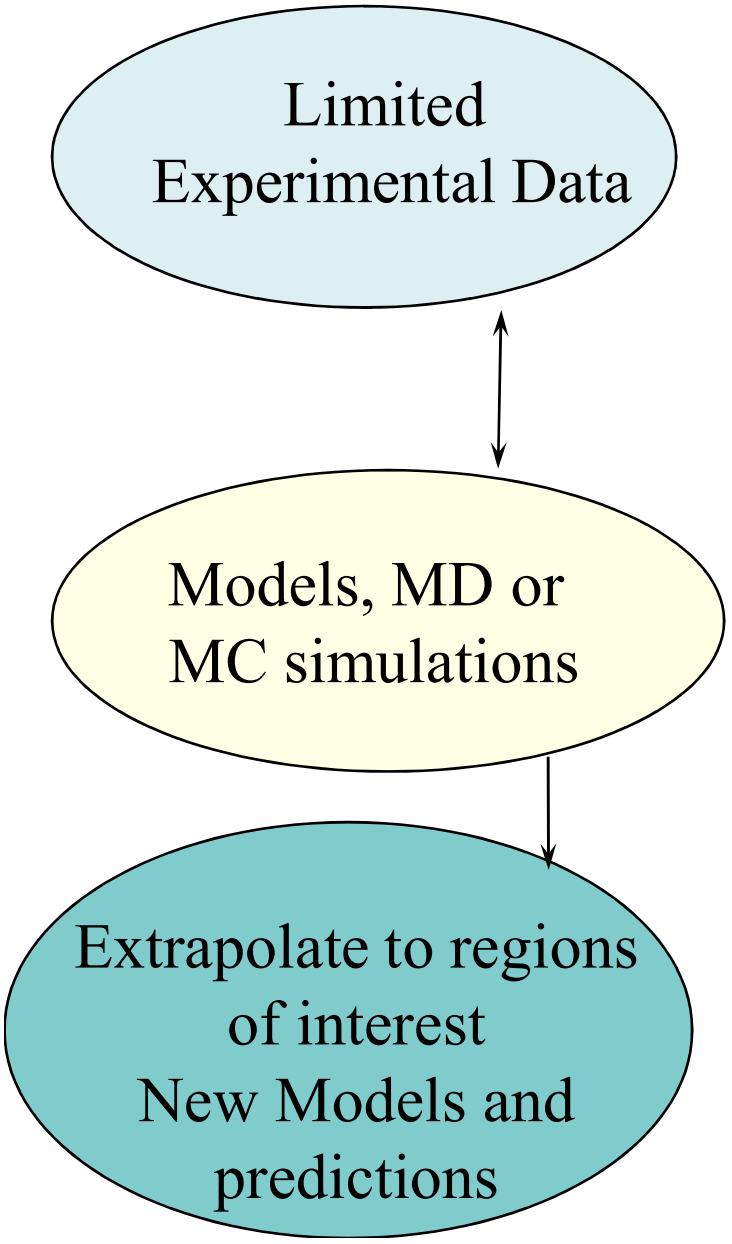
X-ray and "IR" spectra

Etcetera …

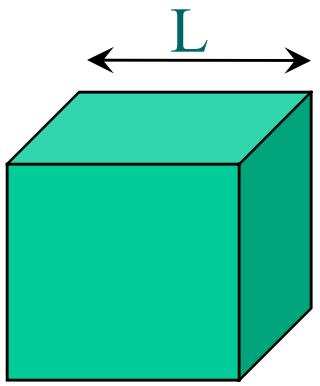
Limitations of MD

- Can simulate only small samples ($L<1$ μm, up to ~10^9 atoms).
- Can simulate only short times ($t<1$ μs, because $\Delta t \sim 1$ fs).
- Computationally expensive (weeks).
- Potential's golden rule: trash in → trash out.
- Interaction potentials for alloys, molecular solids, and excited species not well know.
- Despite its limitations **MD** is a very **powerful tool** to study nanosystems.

The cost of running atomistic simulations



But MD is very costly …



fcc lattice, L~30 monolayers $\Rightarrow 10^5$ atoms

Time step~ 10^{-15} s $\Rightarrow 10^{-11}$ s= 10^4 steps

Speed of typical MD code (short range force field) is ~5 10^{-6} s/(atom*time step)

1 iteration:

50 10^{-6} *10^5*10^4 = 5 10^4 s ~ 14 hours

20 iterations:

Need statistics ….

Total time ~ **12 days (in single core)**

MD limitations in materials sciences

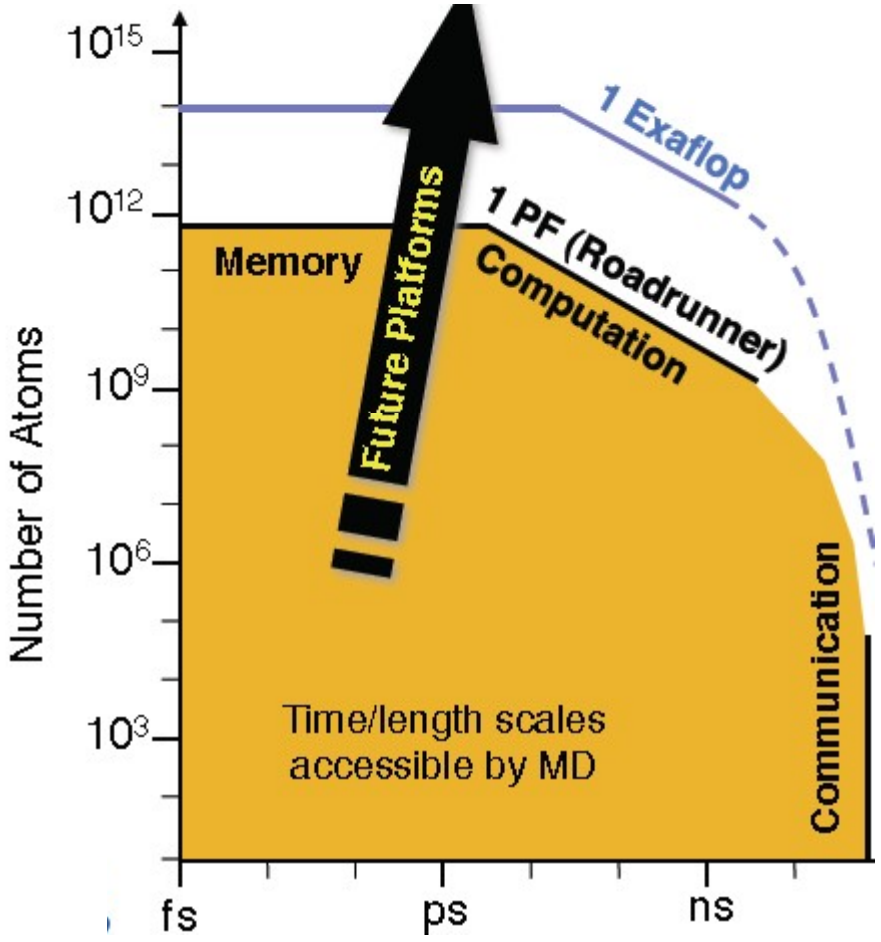


Figure by T. Germann for SPaSM (LANL)

Main Challenges:
Memory limitations +
Communication limitations

Additional problems:
Short range vs. long range potentials (how to find neighbors?), increasing complexity of potentials, I/O (including checkpointing), on the fly analysis, etc.

Pushing boundaries has led to many Gordon-Bell awards

Many MD codes are available
Often used as black-boxes without understanding limitations

LAMMPS (*Large-scale Atomic/Molecular Massively Parallel Simulator*):
http://lammps.sandia.gov/ . MPI ofr several GPUs/cores (LJ: 1.2 ~10^7 atoms max Tesla C2070)

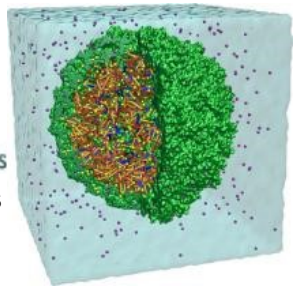
HOOMD-Blue (*Highly Optimized Object-oriented Many-particle Dynamics***)**:
http://codeblue.umich.edu/hoomd-blue/index.html OMP for several GPUs in single board.

DL_POLY:
http://www.cse.scitech.ac.uk/ccg/software/DL_POLY/ F90+MPI, CUDA+OpenMP port.

GROMACS : http://www.gromacs.org/Downloads/Installation_Instructions/Gromacs_on_GPUs
Uses OpenMM libs (https://simtk.org/home/openmm). No paralelization. ~10^6 atoms max.

AMBER (*Assisted Model Building with Energy Refinement*): http://ambermd.org/gpus/
Ross Walker (keynote). MPI for several GPUs/cores. TIP3P, PME, ~10^6 atoms max Tesla C2070)

NAMD ("*Not another" MD*): http://www.ks.uiuc.edu/Research/namd/
GPU/CPU clusters.
VMD (Visual MD): http://www.ks.uiuc.edu/Research/vmd/



1,000,000+ atom Satellite Tobacco Mosaic Virus
Freddolino *et al.*, *Structure*, 14:437-449, 2006.

Many more!!!!

http://en.wikipedia.org/wiki/Molecular_dynamics

GTC 2010 Archive: videos and pdf's: http://www.nvidia.com/object/gtc2010-presentation-archive.html#md

Examples of MD performance in HPC systems

Code	Machine	Cores	Potential	# Atoms	Cutoff (Å)	Atom-Updates/sec	Flops/atom	Atoms in cutoff	TFlop/s
LAMMPS[4]	BG/L	65536	LJ	4.0e10		6.86e9	6.28e2	55	4.3
SPaSM[21]	BG/L	131072	LJ	3.2e11	5.85	11.24e9	2.42e3	71	27.2
SPaSM[21]	BG/L	131072	LJ	3.2e11	11.69	2.50e9	1.92e4	565	48.1
SPaSM[12]	BG/L	65536	EAM	6.4e10	4.68	1.65e9	a)	36	a)
ddcMD	BG/L	65536	EAM	2.9e09	4.50	6.61e9	5.61e3	31	37.0
ddcMD	BG/L	212878	EAM	5.9e09	4.50	20.23e9	5.69e3	31	115.1
ddcMD[37]	BG/L	131072	MGPT-U	5.2e08	7.24	1.18e8	9.09e5	77	107.6
MDGRAPE[27]	MDGRAPE-3	2304	AMBER	1.4e07	30.00	4.03e7	1.37e6	11,414	55.0
MDGRAPE[14]	MDGRAPE-3	4300	AMBER	1.7e07	44.50	4.14e7	b)4.47e6	37,252	185.0
LAMMPS[4]	Red Storm	512	CHARMM	1.6e07	c)	1.20e7	a)	c)	a)
LAMMPS[4]	Red Storm	10000	CHARMM	3.2e08	c)	2.19e8	a)	c)	a)

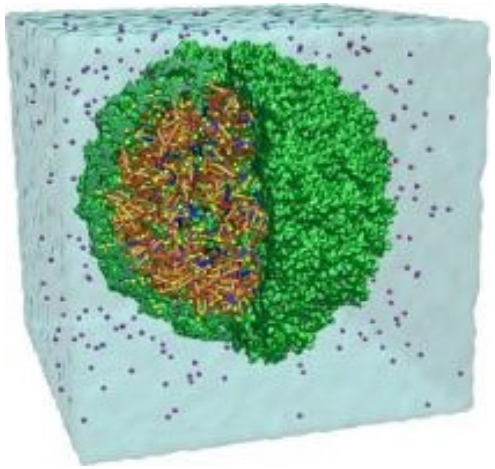
Table 1: A collection of large scale MD simulations comparing performance measures. Note a) No Flop information is provided in the reference. Note b) Neither update rates or Flops per atom provide in the reference. Flops per atom was inferred from the Flops per atoms for the 1.4×10^7 atom MDGRAPE run by scaling with $(44.5/30.0)^8$. This number and Flop rate were used to infer the update rate. Note c) LAMMPS has no cutoff for the Coulomb field. The short range part of the interaction is cutoff at 10 Å.

Table from: "Extending Stability Beyond CPU Millennium: A Micron-Scale Atomistic Simulation of Kelvin-Helmholtz Instability", J. Glosli et al (2007) Supercomputing '07 Nov. 2007, Reno, NV UCRL-CONF-230679

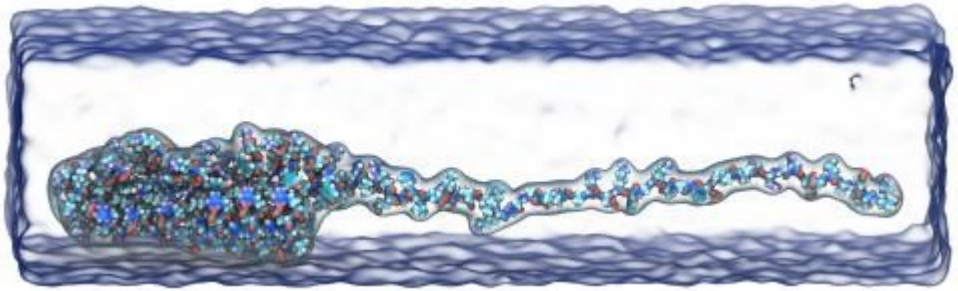
NAMD + VMD (MD+Viz for massive CPU-GPU clusters)

NAMD: http://www.ks.uiuc.edu/Research/namd/
VMD (Visual MD): http://www.ks.uiuc.edu/Research/vmd/

Freddolino *et al.*, *Structure* **14**, 437 (2006)



1,000,000+ atom simulation
Satellite Tobacco Mosaic Virus



Zhao *et al.*, Nature **497**, 643 (2013)

64,000,000 atom simulation
HIV-1 CAPSID

2002 Gordon Bell Award

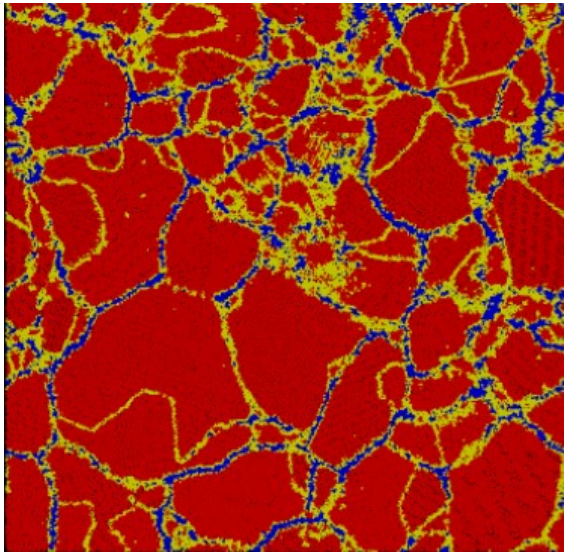
GPU surface rendering http://www.ks.uiuc.edu/Research/gpu/

ddcMD
(domain decomposition + Molecular Dynamics)

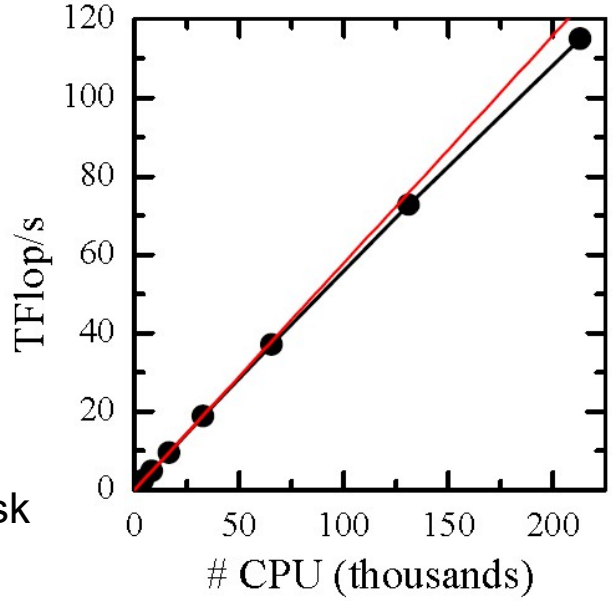
– Particle-based, particle domain decomposition
– 2005, 2007 Gordon Bell Prizes (2009 finalist)

2005: “100+ Tflop/s Solidification Simulations on BlueGene/L.”

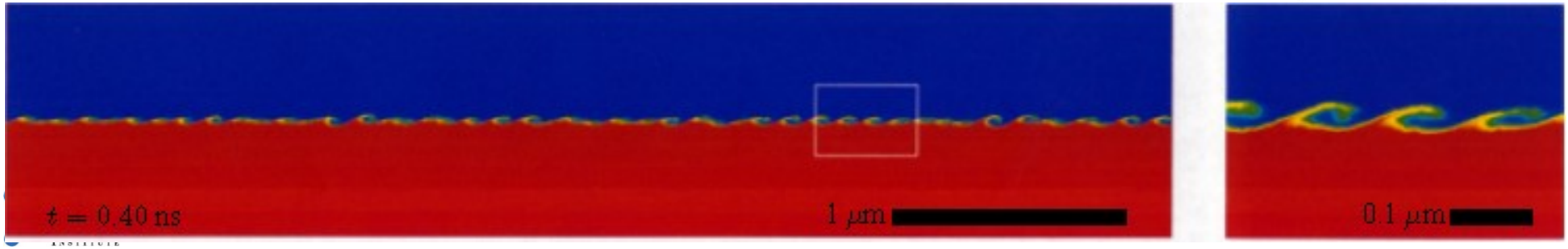
2007: “Extending Stability Beyond CPU Millennium: A Micron-Scale Atomistic Simulation of Kelvin-Helmholtz Instability.” Soft error recovery demonstrated.



16M Ta atoms



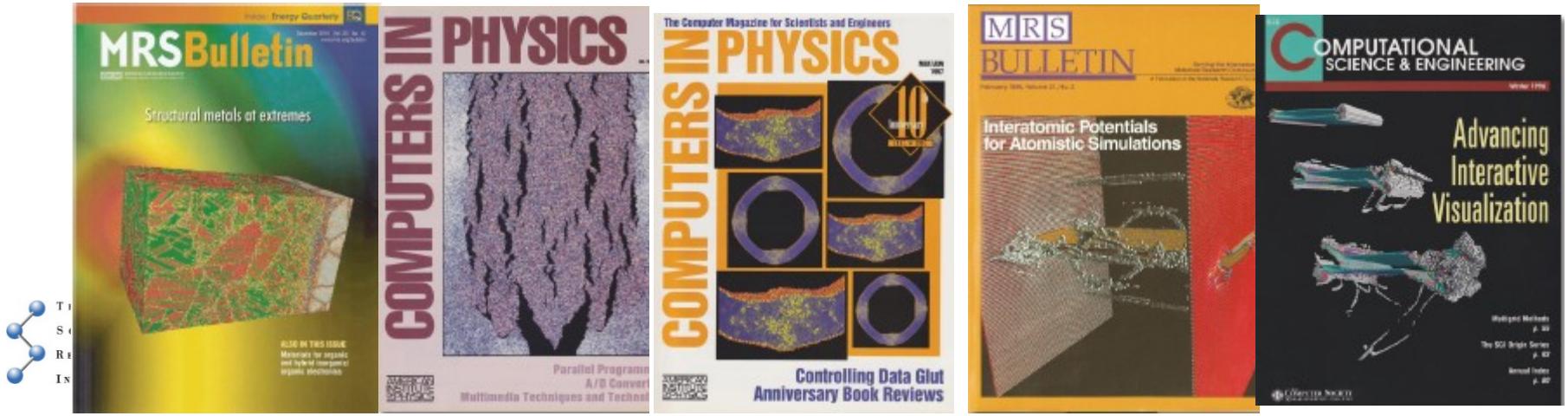
Weak scaling
45,000 atoms/task
on BGL.



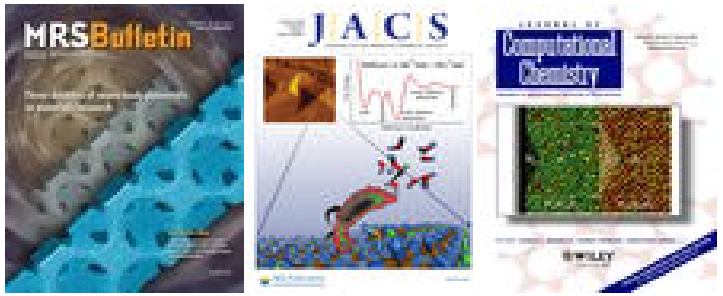
Scalable Parallel Short-range Molecular dynamics (SPaSM)

-Finite-range interactions ➔ O(N) scaling
-Particle-based, spatial (linked-cell) domain decomposition
-In situ visualization demonstrated to 1 trillion atoms on BlueGene/L
-Runtime steering.
-1993, 1998 Gordon Bell Prizes (2005, 2008 finalist)

1993, Performance, Honorable Mention: “Simulating the micro-structure of grain boundaries in solids.” 50 Gflops on a 1,024 processor CM-5.
1998: Price/Performance (2nd price): “Simulation of a shock wave propagating through a structure of 61 million atoms”, 64.9 Gflops/ $ 1 M using a 70 PE system of DEC Alpha’s (533 Mhz.) .
2005: GB Prize finalist: 48 TFlops on BGL
2008: GB Prize finalist: 369 TFlops on RoadRunner



LAMMPS (http://lammps.sandia.gov/)



Some of my personal reasons to use LAMMPS:

1) Free, open source (GNU license).

2) Easy to learn and use:

(a) extensive docs :http://lammps.sandia.gov/doc/Section_commands.html#3_5
(b) mailing list in sourceforge.

(c) responsive developers and user community.

3) It runs efficiently in my laptop (2 cores) and in BlueGeneL (100 K cores), including parallel I/O, with the same input script. Also efficient for GPUs.

4) Very efficient parallel energy minimization, including cg & FIRE.

5) Includes many-body, bond order, & reactive potentials. Can simulate inorganic & bio systems, granular and CG systems.

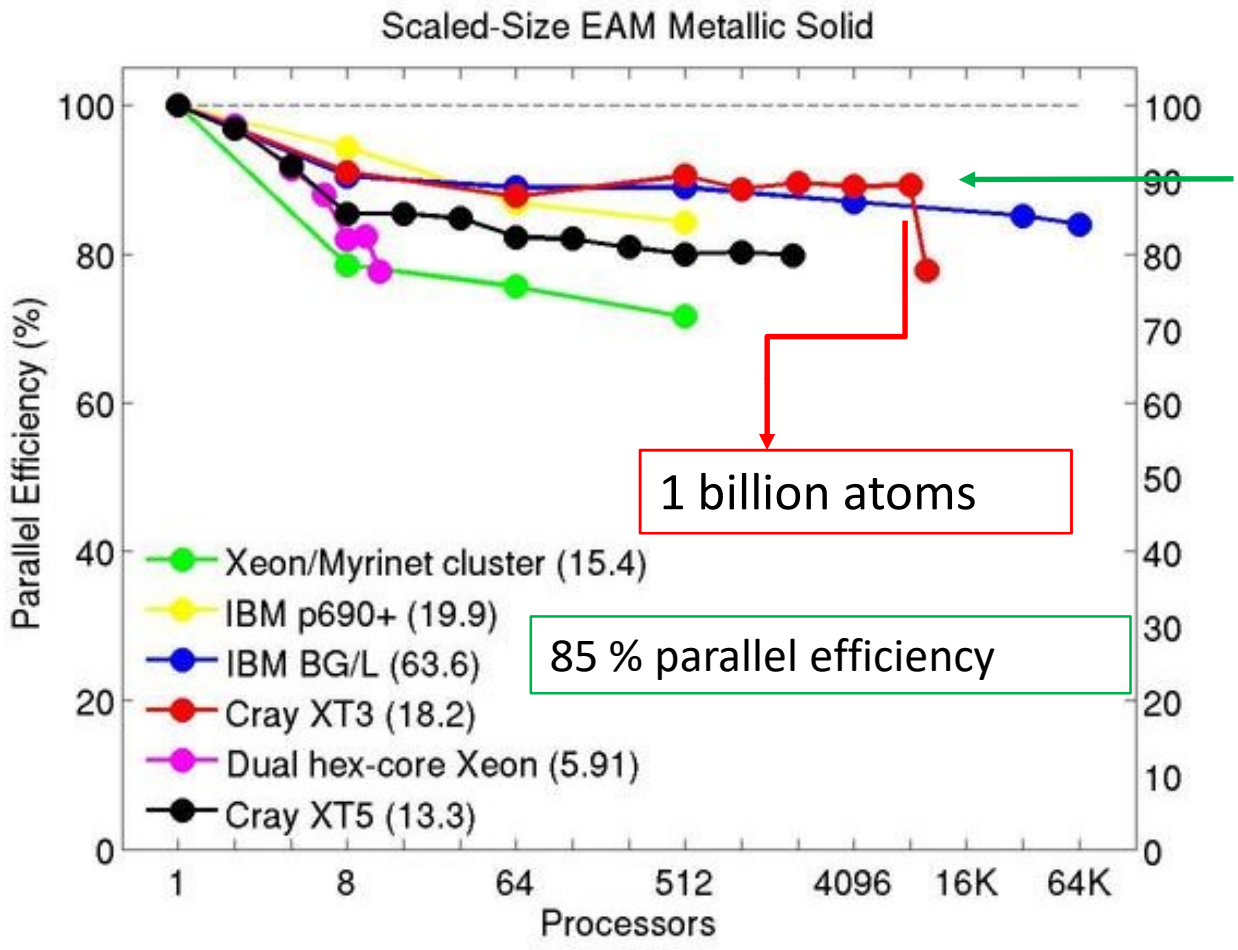
6) Can do extras like DSMC, TAD, NEB, TTM, semi-classical methods, etc.

7) Extensive set of analysis routines: coordination, centro, cna, etc.

8) Easy to write analysis inside input, using something similar to pseudo-code.

MD using "short-range" interactions has nearly perfect parallel scaling (fast connectivity, homogeneous clusters)

http://lammps.sandia.gov/bench.html#eam



LAMMPS

Large-scale Atomic/Molecular Massively Parallel Simulator

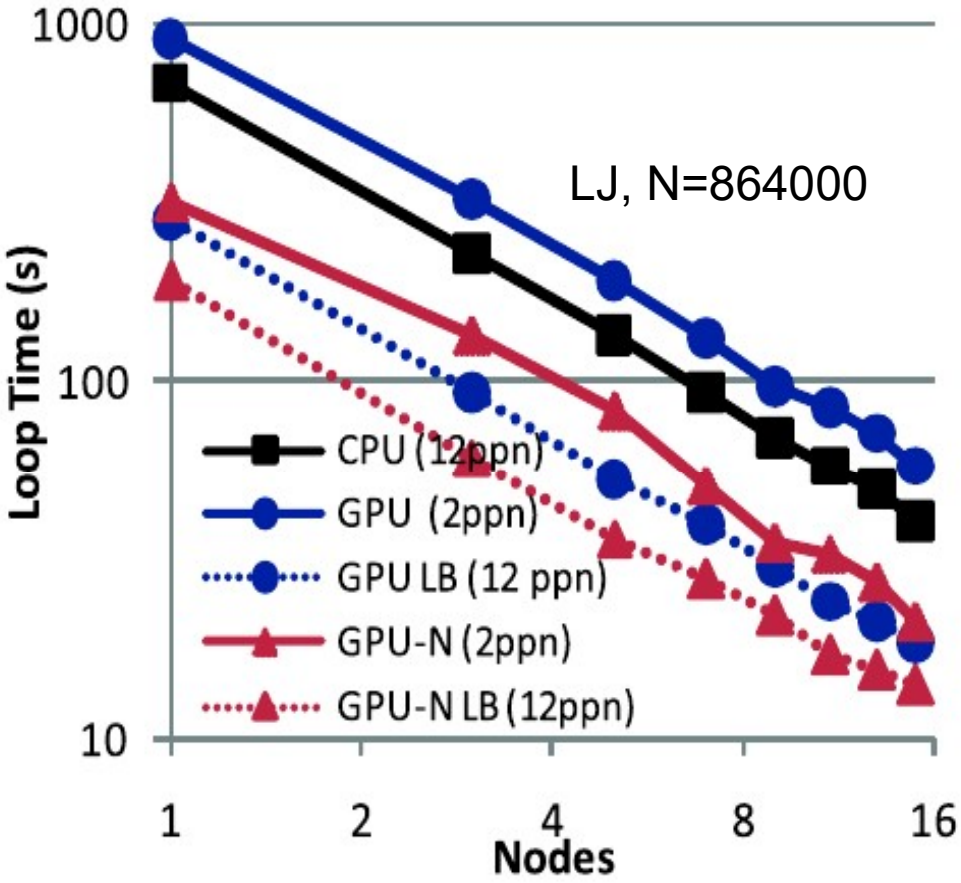
Freeware, open source. CPU-GPU

Use also other software as needed.

Big-data challenges!

Benchmarks for load balancing

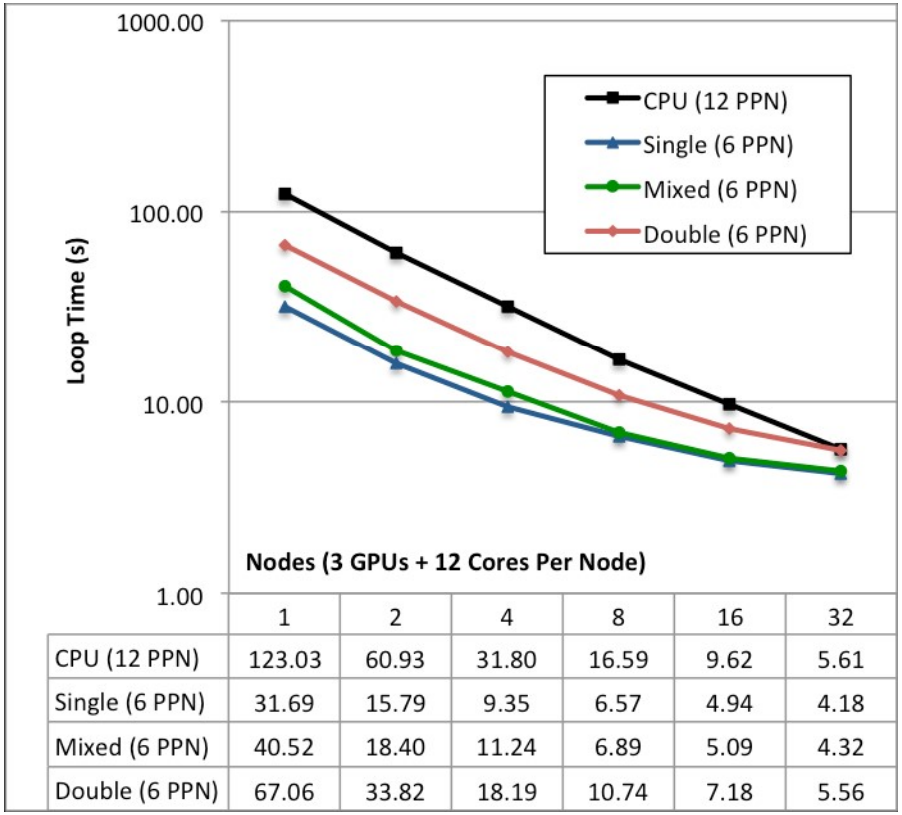
- **The performance impact resulting from splitting the force calculation between the host and device will depend on the CPU core to device ratio and the relative rates of force calculation on the host and device.**
- **Processes per node (ppn).**
- **Dynamic Load Balancing (LB).**
- **Neighboring performed on the GPU (GPU-N).**



Implementing molecular dynamics on hybrid high performance computers – short range forces
W. Michael Brown, Peng Wang, Steven J. Plimpton and Arnold N. Tharrington. Comp. Phys. Comm. **182** (2011) 898–911

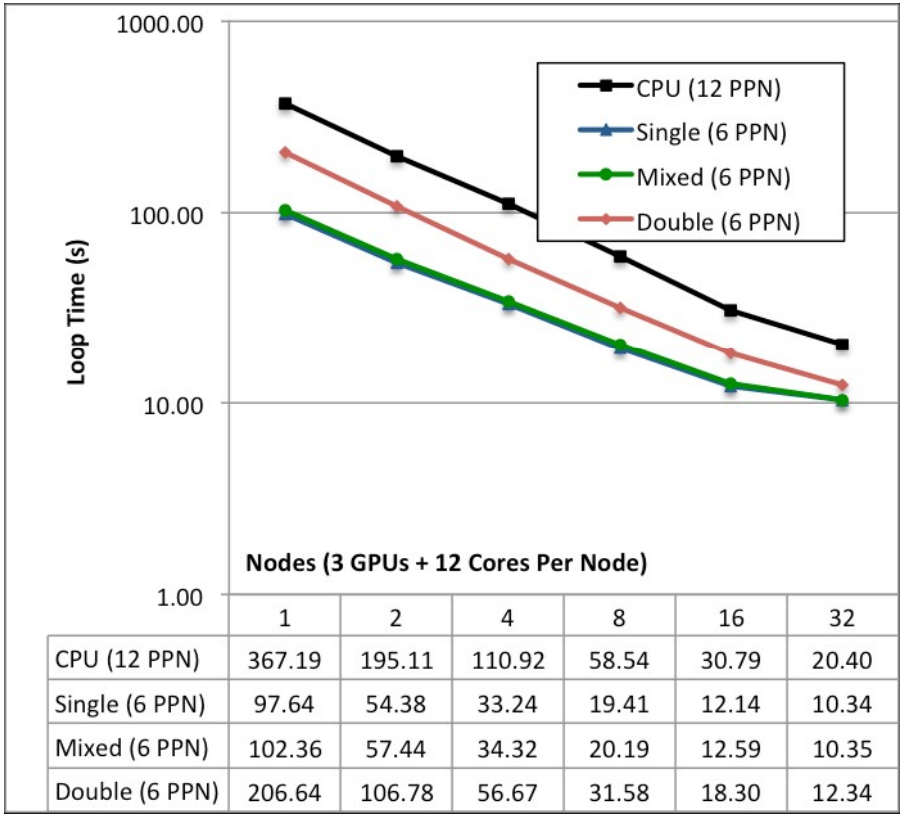
Benchmarks for different precision modes

- Single precision OK for many runs, but use at your peril! Colberg & *Höfling,* Comp. Phys. Comm. **182** (2011) 1120–1129.
- Mixed precision (single for positions and double for forces) nearly as fast as single precision!
- Double precision still cheaper than CPU.



	1	2	4	8	16	32
CPU (12 PPN)	123.03	60.93	31.80	16.59	9.62	5.61
Single (6 PPN)	31.69	15.79	9.35	6.57	4.94	4.18
Mixed (6 PPN)	40.52	18.40	11.24	6.89	5.09	4.32
Double (6 PPN)	67.06	33.82	18.19	10.74	7.18	5.56

256000 atoms LJ liquid, reduced density=0.8442. NVE, r_{cut}= 2.5σ, 5000 steps.



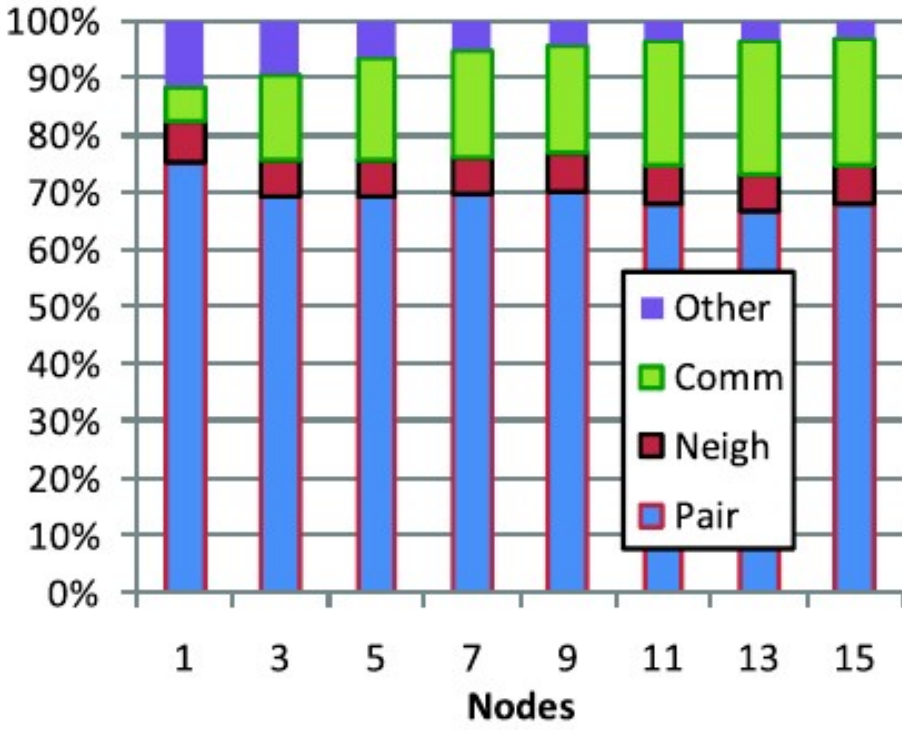
	1	2	4	8	16	32
CPU (12 PPN)	367.19	195.11	110.92	58.54	30.79	20.40
Single (6 PPN)	97.64	54.38	33.24	19.41	12.14	10.34
Mixed (6 PPN)	102.36	57.44	34.32	20.19	12.59	10.35
Double (6 PPN)	206.64	106.78	56.67	31.58	18.30	12.34

Rhodopsin protein in solvated lipid bilayer. CHARMM force field, long-range Coulombics via PPPM,SHAKE constraints.
Counter-ions and reduced amount of water to make a 32K atom system, replicated 2x2x2 to create box.
256,000 atoms, 1000 timesteps, LJ r_{cut}= 1 nm, neighbor skin of 1.0 σ, NPT.

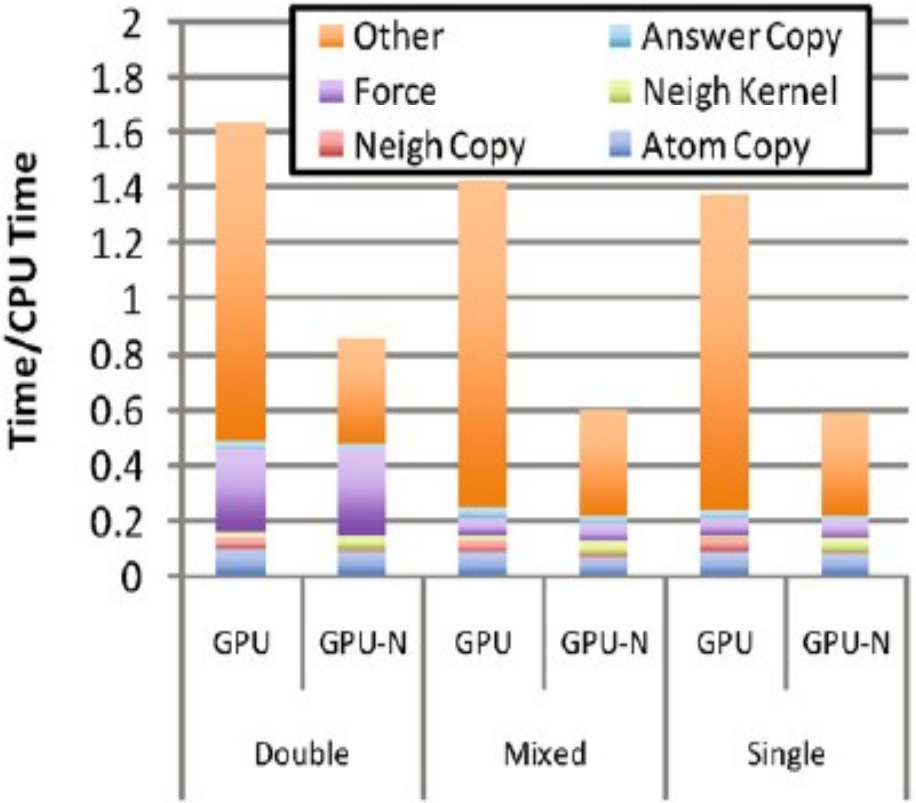
More benchmarks …

Implementing molecular dynamics on hybrid high performance computers – short range forces
W. Michael Brown, Peng Wang, Steven J. Plimpton and Arnold N. Tharrington. Comp. Phys. Comm. **182** (2011) 898–911

Strong scaling benchmark using LJ. cutoff of 2.5 and N=864 K.



LJ. Single node.



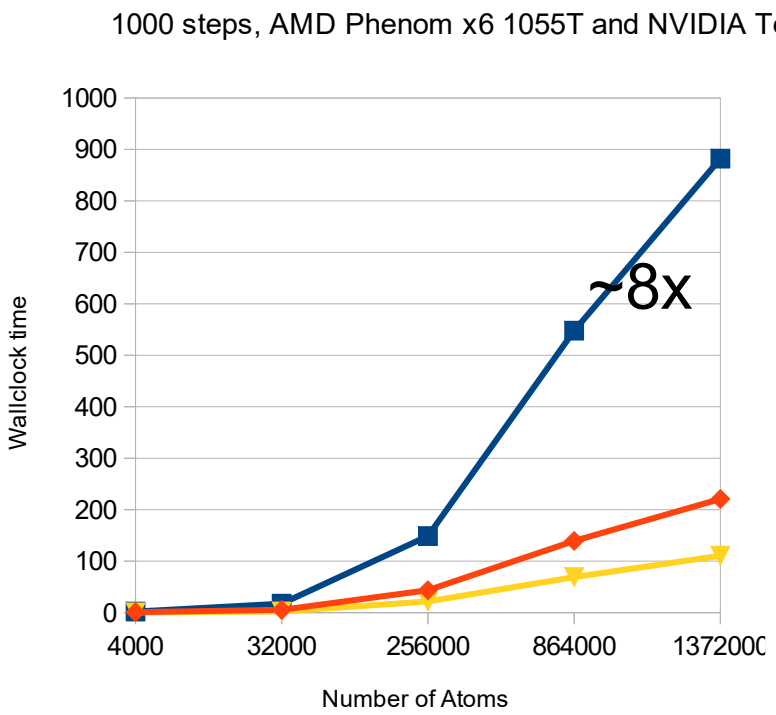
CPU versus CPU-GPU Speedups

Implementing molecular dynamics on hybrid high performance computers – short range forces
W. Michael Brown, Peng Wang, Steven J. Plimpton and Arnold N. Tharrington. Comp. Phys. Comm. **182** (2011) 898–911

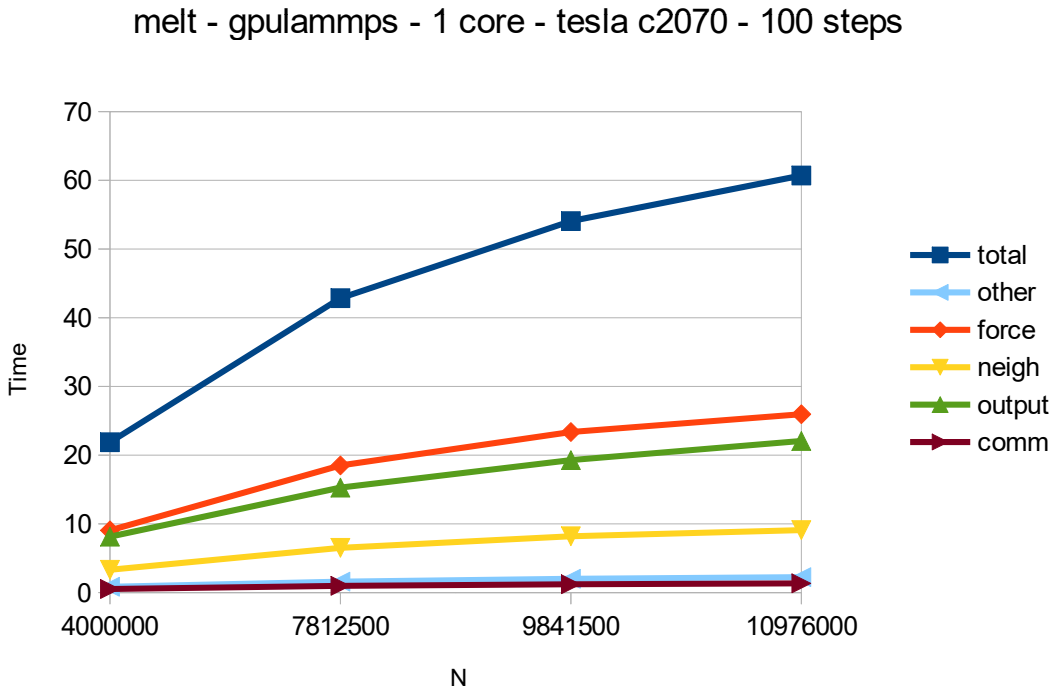
Test case	1 node		15 nodes	
	Cores	Speedup	Cores	Speedup
LJ CPU	12	9.6	180	162.5
LJ GPU single	12	23.4	180	356.4
LJ GPU-N single	12	34.4	180	467.1
LJ GPU double	12	16.0	180	224.1
LJ GPU-N double	12	20.4	180	172.6
GB CPU	12	12.8	180	182.5
GB GPU single	12	146	30	1747.4
GB GPU-N single	12	144.1	30	1541.7
GB GPU double	12	37.2	30	511.4
GB GPU-N double	12	40.9	30	503.7

LAMMPS Scaling with GPU

LJ Melt example



GPU: NVIDIA Tesla c2050, 3 GB RAM, 448 processing units
CPU: AMD Phenom x6 1055t 2.8GHz, 12 GB of RAM.



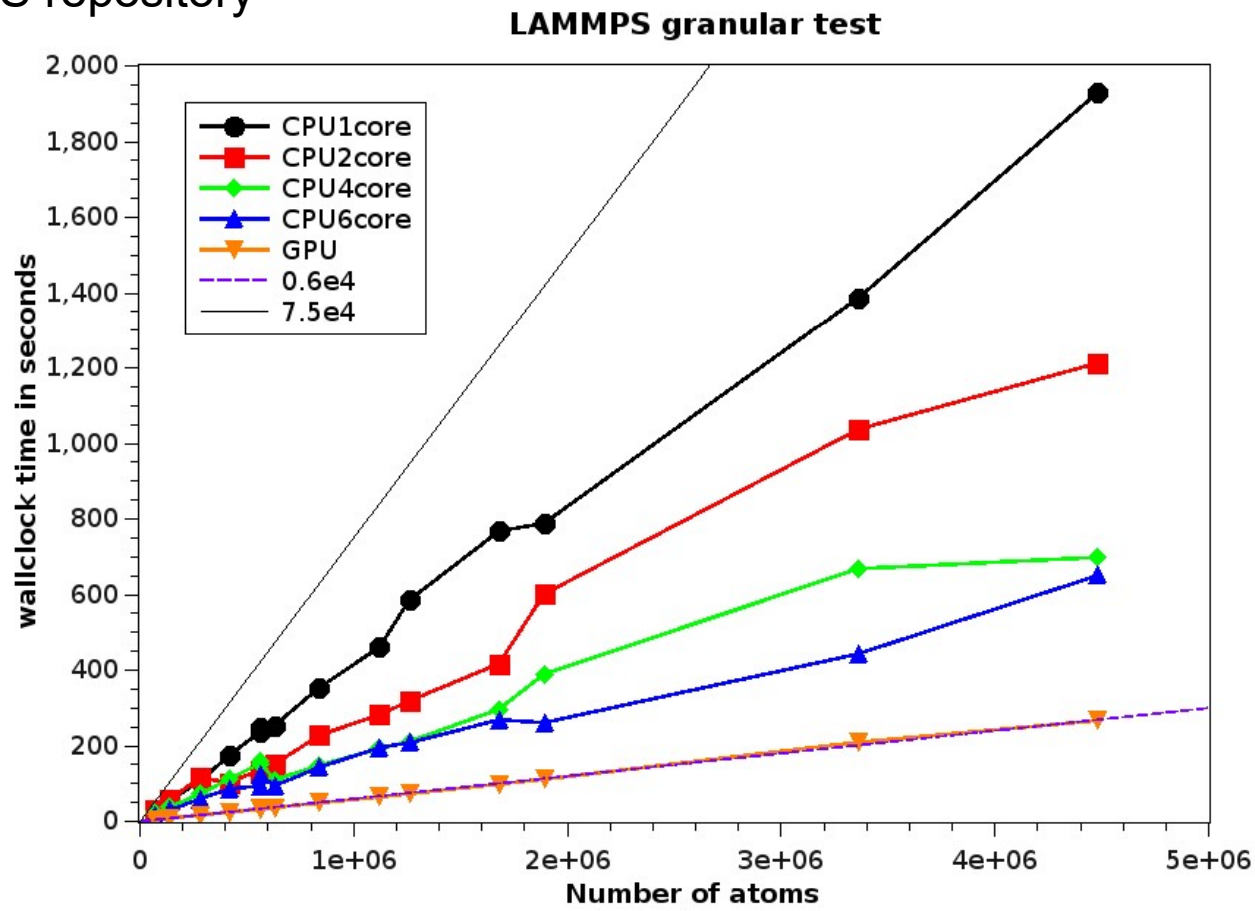
GRANULAR simulations Benchmarks in GPU (extension of USER-CUDA)

GPU version developed by Emmanuel N. Millán
CPU version developed by Christian Ringl (Comp. Phys. 183, 2012)
Code submitted to LAMMPS repository

The 7.5e4 curve represents the results obtained in C. Ringl, Comp. Phys. 183, 2012.

CPU: AMD Phenom x6 1055t 2.8GHz
GPU: NVIDIA Tesla c2050

AVG speedup
GPU vs **1** CPU core = 7x
GPU vs **6** CPU core = 2.95x



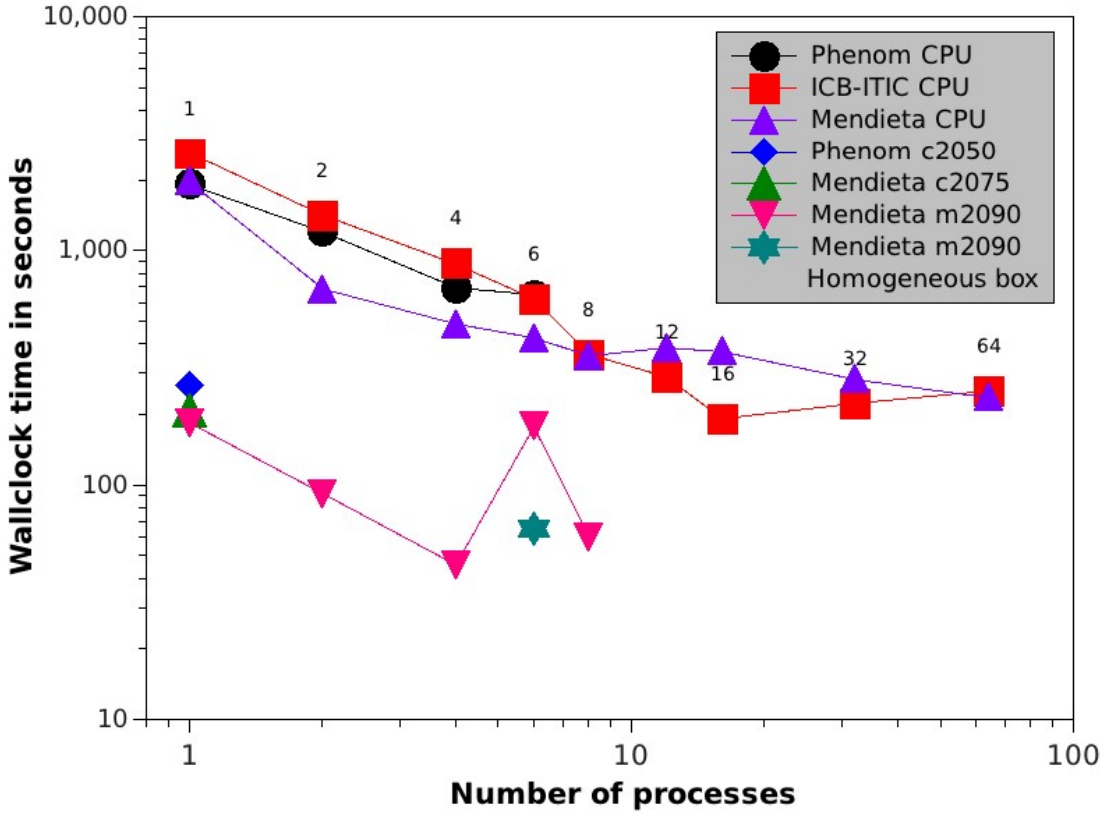
Millan et al. A GPU implementation for improved granular simulations with LAMMPS. HPCLatAm 2013, pp. 89-100 (full paper). Session: GPU Architecture and Applications. C. Garcia Garino and M. Printista (Eds.) Mendoza, Argentina, July 29-30, 2013. http://hpc2013.hpclatam.org/papers/HPCLatAm2013-paper-10.pdf

Benchmarks: Clusters

Granular simulation with the GranularEasy pair style, with 4.48e6 grains and 1000 steps, for 1 through 64 processes, in Mendieta and ICB-ITIC clusters. Various GPUs are tested: C2050, C2075 and M2090.

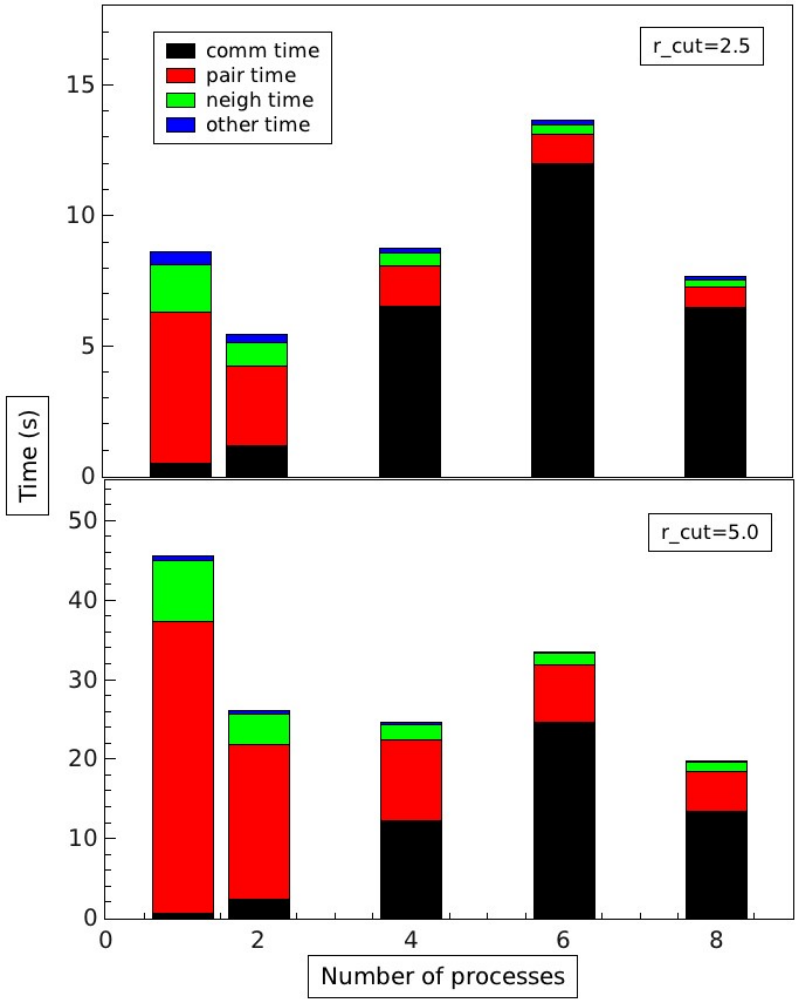
Tesla c2050 GPU ~= 16 CPU cores ICB-ITIC cluster.

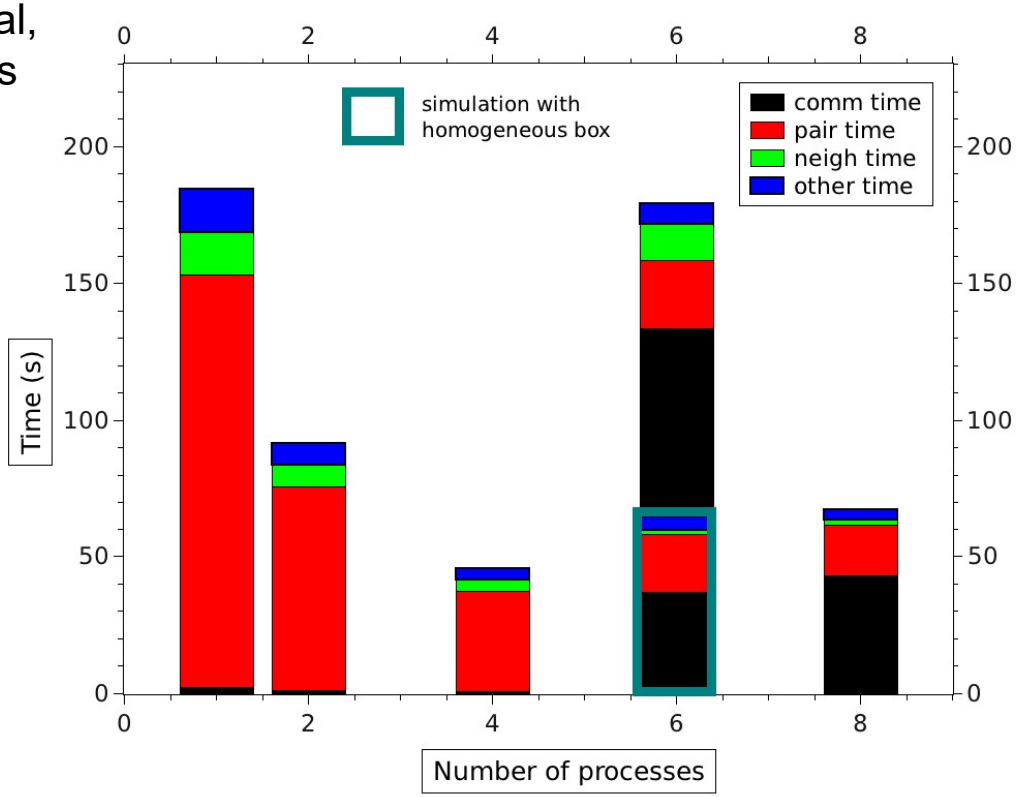
Mendieta Tesla M2090 GPUs best performance using 4 GPUs in two cluster nodes speedup of ~4.2 x against the best CPU result (ICB-ITIC cluster with 16 CPU cores).



Benchmarks: Communication

GPU Melt simulation, Lennard Jones potential, 1000 steps with 256e3 atoms, each atom has ~70 neighbors with 2.5 cutoff and ~500 neighbors with 5.0 cutoff





GPU Granular simulation, 1000 steps with 4.48e6 grains. For six processes, two simulations are shown, one has an elongated, half empty heterogeneous box, and the second has a cubic homogeneous box filled with grains. Each grain has 2-5 neighbors.

Future (?) of MD

- **Sample size: in 10 years, ~tens of μm, but most simulations still sub-μm.**
- **More/better hybrid codes to extend time and length scales: MD+MC, MD+kMC, MD+DD, MD+continuum, MD+BCA, MD+TB, MD+CPMD, …**
- **Time scale problem: new algorithms to extend time scale and simulate thermal evolution.**
- **Better description of electronic effects by:**

  I) Physics + Chemistry + Biology ➔ "reactive" potentials that are accurate and efficient for full periodic table.

  II) coupling to CPMD, tight-binding, etc. (TDDFT?)

  III) TTM, Ehrenfest dynamics, inclusion of magnetic effects, etc.

Major roadblocks:

- **Computers are becoming faster and larger, but algorithms for long range potentials (biology & oxides), ab-initio and continuum simulations typically do not scale well beyond couple thousand CPUs ➔ expect better results within the next 10 years.**
- **No set recipes to build better potentials, specially if chemistry (reactive potentials) or electronic effects (charge transfer, potentials for excited states, etc.) are involved.**
- **Nobody knows yet what to do to solve the time scale problem beyond some simple model problems.**

Coupling TIME and length scales ….

• Choose set of parameters from MD, save those parameters and "pass" them to a "higher" level code. Example: calculate defect concentrations as the initial configuration for a kinetic Monte Carlo code.

• Use some accelerated technique, which boost the time step, for instance "TAD" by A. Voter (LANL). Very expensive computationally, practical only for "2D" simulations or small 3D simulations. Difficult to model "rare events" efficiently.

• Several people are currently working on improving this situation … Keep tuned!

Summary: there are many opportunities for MD

- **Petascale➔Exascale! (USA & EU initiatives).** Science **335**, 394 (2012).
- **New software: novel algorithms and preferably open source** (Nature 482, 485 (2012)**]. Still need significant advances in visualization (**Visual Strategies: A Practical Guide to Graphics for Scientists and Engineers, F. Frankel & A. Depace, Yale University Press, 2012**), dataset analysis [**Science **334**, 1518 (2011)**], self-recovery & fault tolerance, etc.**
- **New hardware :** better (faster/greener/cheaper) processors, connectivity, memory and disk access; MD-tailored machines (MD-GRAPE-4, Anton, etc.); GPUs, MICs, hybrid architectures (GPU/CPU); cloud computing, etc.
- **Experiments going micro-nano/ns-ps ➔ same as MD**
- **Can go micron-size, but still have to connect to mm-m scale ➔** novel approaches needed, including smart sampling, concurrent coupling, dynamic/ adaptive **load balancing/refining** for heterogeneous systems, asynchronous simulations, etc.
- **Need human resources with mix of hardware, soft & science expertise.**

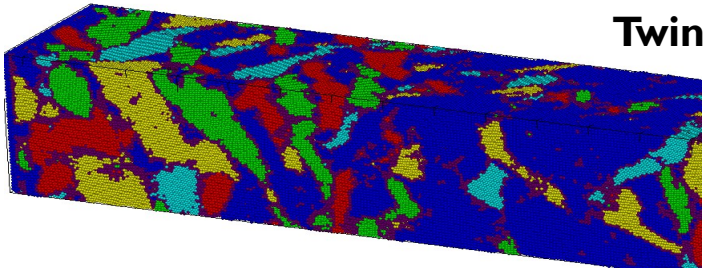
Simulation areas which require urgent new developments

- **Multiscale techniques: multi-scale techniques, fictitious dynamics, "rare events", etc.**
- **Técnicas multi-escala espaciales: problemas de frontera y acoplamiento entre escalas, incluyendo problemas "estáticos" y dinámicos.**
- **Inestabilidades y fragmentación: RT, RM, Euler, parámetros de orden, etc.**
- **Medios desordenados: estructura, plasticidad y viscosidad en vidrios y medios porosos.**
- **Propagación de ondas en medios no homogéneos, con propiedades no-lineales y posibles cambios de fase.**
- **Data mining en archivos de TBs: cómo encontrar la aguja en el pajar.**
- **Como graficar en paralelo y con interfaces "amigables".**
- **Nuevos algoritmos eficientes en paralelos para problemas mucho mas complejos que los que se resuelven muy bien en sistemas pequeños en serie: Monte Carlo, métodos de minimización, interacciones de largo alcance, códigos CFD, etc.**

Algún voluntario?

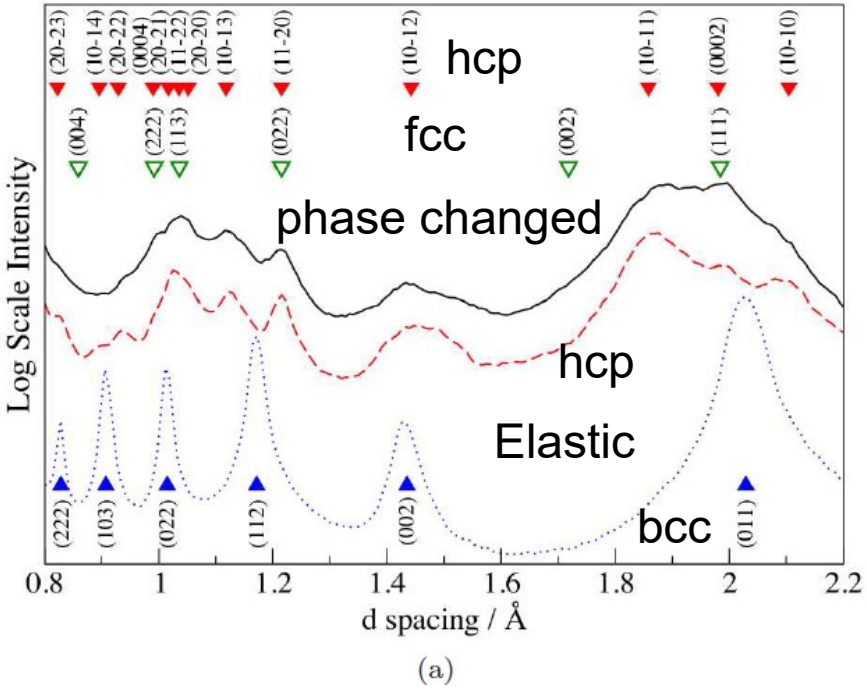
Simulated X-Ray diffraction (use cufftw)

A. Higginbotham, M. Suggit, J.S. Wark (U. Oxford).



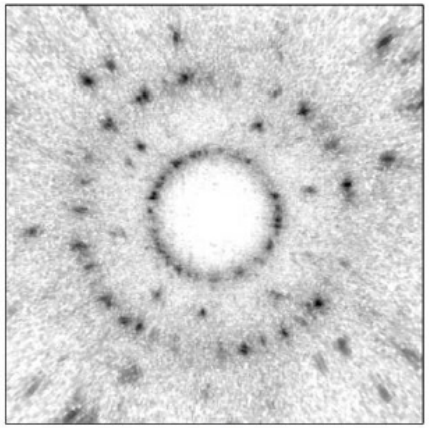
Twin detection in bcc metals: Suggit *et al*, Phys. Rev. B (2013)

Fe phase change: Gunkelmann *et al*, Phys. Rev. B (2014)

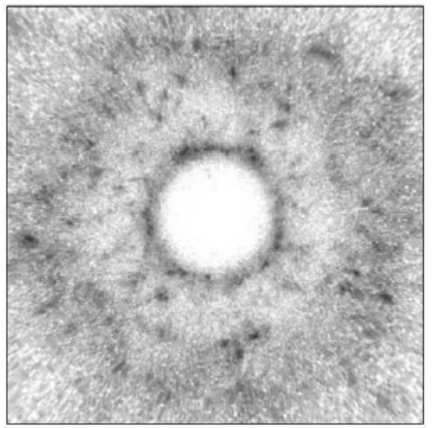


(a)

Experimental geometry: 50 × 50 mm film, placed 30 mm in transmission, 8.05 keV (Cu Kα) X-rays, perpendicular to the film.



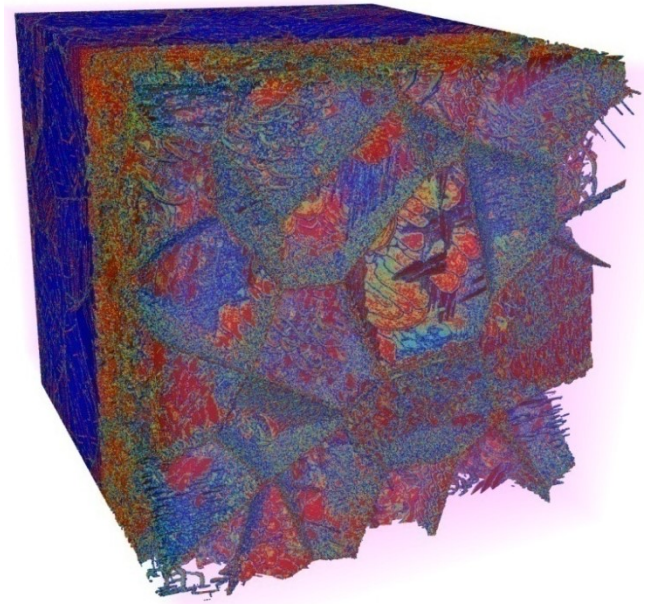
(b) unshocked

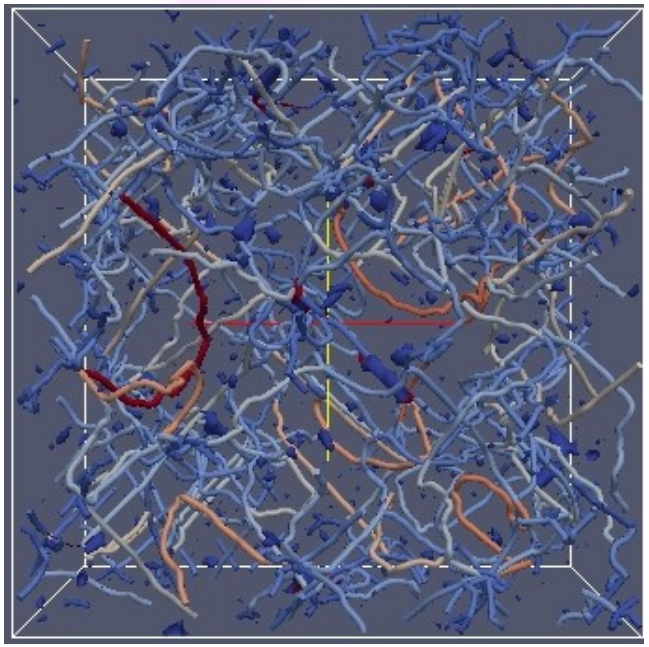


(c) phase changed

Summary and future perspectives

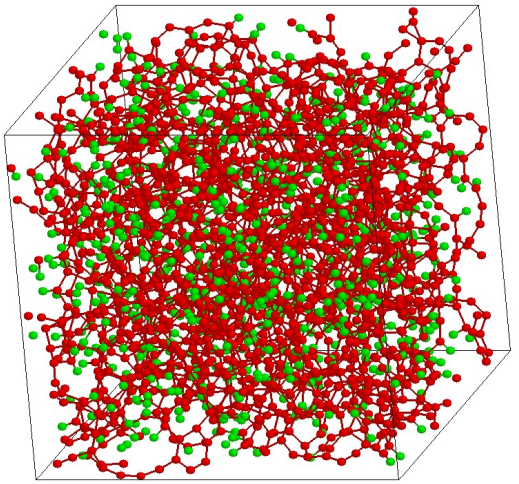
- Thermodynamics and stat mech are “robust”, even for non-steady, few-particle scenarios.
- Novel experimental ultra-fast diagnostic techniques already allow direct comparison with MD simulations.
- New computers, together with new programs and models, allow direct comparison of MD and experiments.
- GPUs and other commodity novel hardware allows unprecedented performance for small systems.

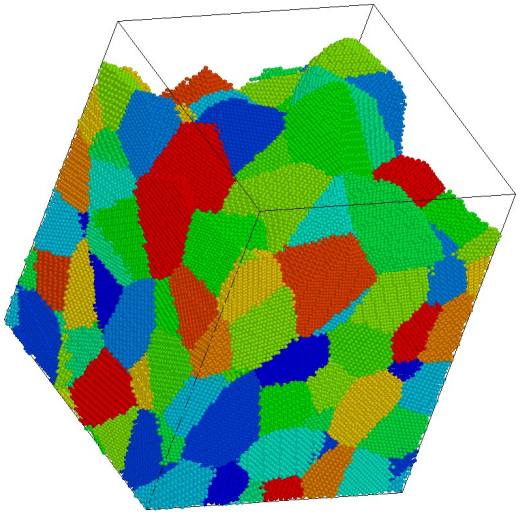




More future perspectives

- Multicore +GPU clusters: challenges on load balancing, communication and synchronization.
- MD often requires more CPU/GPU time for analysis than for time evolution (months versus days) ➔ need parallel processing.
- Need smart algorithms which scale well in parallel archs taking advantage of link-cells.
- Need parallel viz tools for samples with +few million atoms (generally not the case in chemistry/biology).
- GPU processing has bright future!





Next: Lab + Advanced MD

Introduction to Linux. Emmanuel Millan (UNCuyo)

REPASO para el laboratorio computacional

Utilidades: grep y tar

- grep busca una determinada cadena en un archivo o en la salida de un comando.
  - ps ax | grep openmpi (busca en los procesos que están ejecutandose el proceso openmpi)
  - grep “run” in.melt (busca la palabra “run” en el archivo in.melt)
- tar en conjunto con gzip o bzip2 sirven para comprimir/descomprimir archivos.
  - tar zcvf comp.tar.gz archivo1 archivo2 ...
  - tar zcvf comp2.tar.gz ./*
  - tar xvf comp3.tar.gz (descomprimir)

Utilidades: awk

- La función básica de awk es buscar lineas en archivos que contienen un cierto patrón. Awk realiza una tarea específica sobre las lineas encontradas.
- Ejemplos:
  - Ordena las lineas de un archivo:
    - $ awk -F: '{ print $1 }' /etc/passwd | sort
  - Imprime solo las lineas pares:
    - $ awk 'NR % 2 == 0' data
  - Suma la columna número 5 del comando ls -l
    - ls -l | awk '{ SUM += $5}
      END { print SUM }'

Referencia: http://www.gnu.org/software/gawk/manual/gawk.html

Utilidades: gnuplot

- Utilidad de linea de comandos para graficar. Se ejecuta con el comando gnuplot.

Ejemplo:

gnuplot> set terminal "png"

gnuplot> set output "salida.png"

gnuplot> splot "dump.col.0" using 2:3:4

Guarda en un archivo de imagen png el gráfico 3D utilizando las columnas 2, 3 y 4 del archivo de entrada.

Documentación:

http://www.gnuplot.info/docs_4.4/gnuplot.pdf

http://physicspmb.ukzn.ac.za/index.php/Gnuplot_tutorial

http://www.duke.edu/~hpgavin/gnuplot.html

Utilidades: ssh y scp

La forma de acceder de forma remota a una máquina mas comúnmente utilizada es a través del comando ssh. Sintaxis:

$ ssh [username]@[ip | hostname]

Ejemplo

$ ssh emmanuel@www.miservidor.com.ar

Para copiar archivos hacia o desde una máquina remota se puede utilizar el comando scp:

$ scp -r emmanuel@www.miservidor.com.ar:/home/emmanuel/lammps* /tmp

$ scp -r /tmp/lammps/examples emmanuel@www.miservidor.com.ar:/home/emmanuel/tmp

Los comandos ssh y scp están incluidos por defecto en la mayoría de las distribuciones Linux, en el caso de windows se puede utilizar Putty para acceder a través de ssh a un servidor remoto.

Cálculo paralelo

Existen varias alternativas para realizar cálculo en paralelo, dependiendo del hardware disponible y los requerimientos del software. Algunas opciones son: MPI (OpenMPI o MPICH), OpenMP, GP-GPU (CUDA/OpenCL).

El código tiene que estar desarrollado para ejecutarse en paralelo. No es posible ejecutar una aplicación desarrollada para un procesador en un cluster de varios procesadores a menos que la aplicación sea reescrita.

En las siguientes slides se presenta cada unas de estas opciones.

MPI

MPI ("Message Passing Interface", Interfaz de Paso de Mensajes) es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.

En otras palabras, es una librería que permite el pasaje de mensajes entre computadoras para ejecutar un mismo programa, dividiendo las tareas a realizar entre los distintos procesadores.

Debe quedar claro que el programa tiene que estar desarrollado utilizando esta librería, de no ser así, no se puede ejecutar en paralelo una aplicación que no ha sido desarrollada teniendo en cuenta esto.

Las implementaciones de esta librería mas utilizadas son OpenMPI y MPICH.

Ref: http://es.wikipedia.org/wiki/Interfaz_de_Paso_de_Mensajes
http://www.open-mpi.org/

MPI

- Aplicaciones como LAMMPS, Gromacs, namd, etc, utilizan MPI para dividir calculo en procesadores de un cluster o de una workstation.
- Por ejemplo, para ejecutar LAMMPS en 10 procesadores de un cluster se ejecuta la siguiente linea:
  - *$ mpirun -np 10 -machinefile machines /home/emmanuel/lammps/src/lmp_openmpi < in.melt > salida.log*

Este comando lee el archivo *machines* que contiene el listado de computadoras en donde se debe ejecutar LAMMPS (por número de IP o nombre de máquina, una por linea), luego el ejecutable que tiene que llamar, que archivo de entrada debe leer el ejecutable (in.melt) y para finalizar en que archivo debe guardar la salida del comando.

OpenMP

OpenMP es una interfaz de programación de aplicaciones (API) para la programación multiproceso de memoria compartida en múltiples plataformas. Permite añadir concurrencia a los programas escritos en C, C++ y Fortran sobre la base del modelo de ejecución fork-join.

Esto permite que en una máquina con múltiples núcleos, nuestro programa se ejecute de forma paralela en todos los núcleos disponibles y compartan la memoria del sistema. El compilador GCC incluye la librería.

Ejemplo:

```
#pragma omp for
for(int n=0; n<10; ++n)  {
  c[n]= a[n] + b[n];
}
```

Tutoriales:
http://bisqwit.iki.fi/story/howto/openmp/
https://computing.llnl.gov/tutorials/openMP/

Referencia: http://es.wikipedia.org/wiki/OpenMP - http://openmp.org/wp/

GP-GPU: CUDA / OpenCL

Un nuevo paradigma de programación se ha hecho popular en los últimos 5 años aprox. Las tarjetas gráficas tiene potentes procesadores utilizados para renderizar imágenes de juegos, la comunidad científica supo aprovechar esta potencia de cálculo y comenzó a utilizar OpenGL (una librería para generar gráficos) para realizar cálculos matemáticos.

Luego NVIDIA desarrollo CUDA con el fin específico de realizar cálculos de propósito general sobre sus tarjetas gráficas. La primera versión del SDK para desarrollar en CUDA salió en el año 2007.

Por otro lado Apple propuso desarrollar una API abierta, llamada OpenCL, en el 2008 el grupo Khronos tomo la API y un consorcio de empresas administra su desarrollo. AMD / ATI apoya este estándar en lugar de su API que prácticamente se ha dejado de utilizar.