

# Supercomputing in Molecular Dynamics

Anthony Trubiano

Molecular dynamics (MD) simulations of many particle systems are increasingly being used to study the form and function of biological molecules and proteins. These systems are typically too small or evolve too quickly to be studied experimentally, so MD simulations have become the main tool in studying them. Studying such systems provides insight into physical phenomena such as phase transitions and nucleation, and can also be used to guide drug design.

The need for large computational resources comes in due to the size of the systems and the time scales that need to be resolved. Simulations involve computing the forces and then integrating the equations of motion in time for each particle, of which there may be thousands. For stability purposes, the time step is typically in the range of femtoseconds to picoseconds ( $10^{15} - 10^{-12}$  seconds), while the events of interest take place on the nanosecond, microsecond, or even millisecond time scale.

Parallelization plays a role in two distinct ways here. The first is splitting the computational domain among many processors. Each processor is responsible for computing forces between atoms in its domain. There is then some communication between nearby domains to update the forces.

An alternative method is to use Monte Carlo type methods. Here, instead of integrating equations of motion, we propose a random move and accept it with probability proportional to the Boltzmann distribution. This is given by

$$P(x) = \exp(-\beta\Delta U(x)),$$

where  $x$  is the proposed state,  $\Delta U$  is the potential energy difference between initial and final states, and  $\beta$  is an inverse temperature. Here we see that as  $\beta \rightarrow \infty$  (i.e. temperature goes to 0), the acceptance probability goes to zero. Thus the simulation can get stuck in a region for a long time. A remedy for this is called parallel tempering, in which  $M$  simulations are run on different processors, each at a different temperature. Then, every  $k$  steps, we propose a configuration swap across processors, that is accepted with some probability. This helps the low temperature simulations avoid getting stuck for long periods of time and can drastically reduce the simulation time needed to achieve a desired order of accuracy.

One of the popular MD simulation programs is called NAMD, developed at Argonne National Lab. It is run on their supercomputer, Mira, which is number 21 on the Top 500 list. It uses an BlueGene/Q Power BQC 16C 1.6Ghz processor, and has 786432 cores. They state their largest simulations run on

up to 500,000 cores, so it seems quite scalable. Using this software, they have simulated sub-atomic scales, membrane proteins, molecular motors, and HIV.