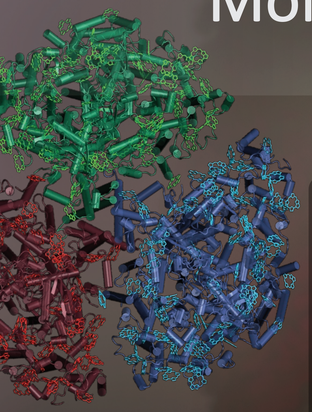
Molecular Dynamics Simulations on Supercomputers Performing 10^{18} flop/s

Abhinav Bhatele, William D. Gropp and Laxmikant V. Kale



Trends in High Performance Computing

A flop/s refers to one floating point operation per second. 1 Teraflop/s refers to 10^{12} flop/s and an Exaflop/s is 10^{6} Teraflop/s or 10^{18} flop/s.

The first Teraflop/s computer: ASCI Red in 1997
The first Petaflop/s computer: Roadrunner in 2008
The first Exaflop/s computer: 2018 ?

Jaguar, a Cray XT5 - the fastest supercomputer in US

Strong scaling of parallel codes to exascale machines will be a challenge and it is important to study the scaling behavior of existing algorithms and their implementations.

A Hypothetical Exascale Machine

It is diffcult to predict what an exascale machine will look like. However, we can attempt to project the performance of parallel applications with some minimalistic assumptions about the machine. This can help in influencing architectural decisions for building them.

Assumptions for this study:
- 1 GHz processing cores, compute time per flop, t_c = 0.1 ns
- Each node will have 1024 such cores
- Number of nodes, $P_n = 2^{20}$, number of cores, $P_c = 2^{30}$

Performance Model for Analysis

The amount of computation is described in terms of the problem size, N, and the number of processing cores, P_c. For each computation, we calculate the number of floating point operatons, n and multiply that by the time for computing a flop.

$$T_{comp} = \frac{1}{\eta} \times f(N, P_c) \times n \times t_c \quad \ldots (1)$$

Communication on parallel machines can be described in terms of:
- Start-up time (t_s): time required for handling the message at the sender and receiver
- Per-hop time (t_h): time spent at every switch/router on the network that the message goes through
- Per-word time (t_w): if the size of each word is 4 bytes and the link bandwidth is B_w, then time spent by a word to traverse a link = $4/B_w$.

Hence, time for sending a message on the network is given by,

$$t_s + l \times t_h + m \times t_w$$

where l is the number of hops, and
m is the size of the message.

If an application sends M = $g(N, P_c)$ messages and each message is of size $h(N, P_c)$, then time for communication is given by:

$$T_{comm} = M \times (t_s + h(N, P_c) \times t_w) \quad \ldots (2)$$

Molecular Dynamics

MD simulations form an important class of parallel applications used for studying the life of biomolecules. NAMD, AMBER, Gromacs and Desmond are examples of some popular parallel MD codes. For this study, we focus on "short-range" MD and use a spatial decomposition of the simulation box for parallelization.

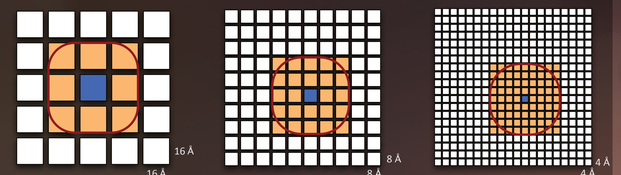
For short-range MD, if we have > 100 atoms per core, the performance follows a universal curve depending only on the number of atoms. Hence, total size of the system to achieve 1 Exaflop/s (10% of the peak performance) = $2^{30} \times 100 \approx 107$ billion atoms. Number of floating point operations for a molecular system of N atoms = $33547 \times N$. Since we want the number of flops to be greater than 10^{18} flop/s,

$$\frac{flops}{T} > 10^{18} \quad \text{or} \quad \frac{33547 \times N}{10^{18}} > T$$

Putting the value of $N = 2^{30} \times 100$, $T < 3.6 \times 10^{-3}$ seconds

Weak Scaling

To estimate the communication involved, we need to decide the size of the simulation box and its decomposition into cubiodal cells.



The standard size of each cell containing 400 atoms is 16 Å in each dimension. If each cell has 100 atoms, two of the dimensions are split into half. In this scenario, each cell communicates with 5 × 5 × 3 = 75 other cells to calculate the forces on its atoms.

Since there are 1024 cores on each node, inter-node messages will be sent only for cells on the surface. Assuming 8 x 8 x 16 cells are assigned to each node, # external messages = 12 x 10 x 20 − 8 x 8 x 16 = 1376.

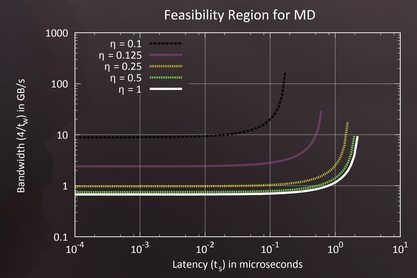
Using equations (1) and (2), time for each MD iteration,

$$T = \frac{1}{\eta} \times \frac{N}{P_c} \times 33547 \times t_c + 1376 \times \left(t_s + \frac{N}{P_c} 4t_w\right) \quad \ldots (3)$$

Putting in the values of N/P_c = 100 and $t_c = 10^{-10}$ seconds,

$$\frac{1}{\eta} \times 100 \times 33547 \times 10^{-10} + 1376 \times (t_s + 4 \times 100 \times t_w) < 3.6 \text{ X } 10^{-3}$$

$$t_s + 400t_w < 2.62 \times 10^{-6} - \frac{1}{\eta} \times 2.44 \times 10^{-7}$$



Smaller Problem Sizes

Simulating a 107 billion atom molecular system will be a challenge for biophysicists and will require doing very long simulations (milliseconds to seconds). Hence, we also consider smaller problem sizes as shown below (time per step calculated based on the # atoms):

# Atoms	Atoms/core	Cell Dimensions (Å)	Time (*ms*)
107 billion	100	8 × 8 × 16	3.602
53.6 billion	50	8 × 8 × 8	1.801
21.5 billion	20	5.33 × 5.33 × 8	0.720
5.4 billion	5	4 × 4 × 4	0.180

Similar to the weak scaling case, we can write equations for the time per step for these smaller problems:

$$T_{50} = \frac{1}{\eta} \times \frac{N_{50}}{P_c} \times 33547 \times t_c + 1856 \times \left(t_s + \frac{N_{50}}{P_c} 4t_w\right)$$

$$T_{20} = \frac{1}{\eta} \times \frac{N_{20}}{P_c} \times 33547 \times t_c + 2672 \times \left(t_s + \frac{N_{20}}{P_c} 4t_w\right)$$

$$T_5 = \frac{1}{\eta} \times \frac{N_5}{P_c} \times 33547 \times t_c + 5120 \times \left(t_s + \frac{N_5}{P_c} 4t_w\right)$$

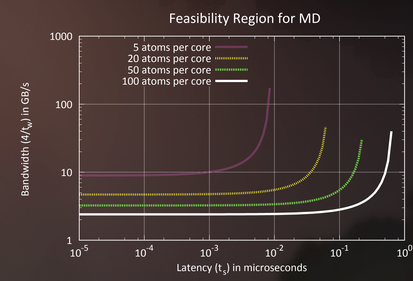
Putting the values of $N_{50}/P_c = 50$, $N_{20}/P_c = 20$, $N_5/P_c = 5$ and $t_c = 10^{-10}$ s,

$$t_s + 200t_w < 9.7 \times 10^{-7} - \frac{1}{\eta} \times 9.04 \times 10^{-8}$$

$$t_s + 80t_w < 2.69 \times 10^{-7} - \frac{1}{\eta} \times 2.51 \times 10^{-8}$$

$$t_s + 20t_w < 3.52 \times 10^{-8} - \frac{1}{\eta} \times 3.28 \times 10^{-9}$$

Using these equations we can plot and compare the latency and bandwidth requirements for different problem sizes:



Conclusion and Future Work

In summary, MD codes will have reasonable network requirements on exascale machines for weak scaling. Smaller problem sizes will require significantly lower latencies and higher bandwidth to achieve 1 Exaflop/s.

We plan to extend this work by modeling long-range elcectrostatic force calculations, load imbalance in MD simulations and hybrid decomposition strategies. Future analysis will also consider overlap of computation and communication.

Acknowledgements:
Jaguar: Downloaded from the Media Center on the National Center for Computational Sciences website: http://www.nccs.gov/media-center/image-gallery/
Trimeric photosystem I and Chromatophore vesicle: Generated using VMD, thanks to Melih Sener and Danielle Chandler (Theoretical and Computational Biophysics Group)
Poster background: designed by David Michael Kunzman (Parallel Programming Laboratory)

PARALLEL PROGRAMMING LAB
DEPT OF COMPUTER SCIENCE, UNIVERSITY OF ILLINOIS

NAMD@EXASCALE