Project 5

Report Due: December 1 (Monday), 2025 at 11:59 pm
Presentation (PPT or else) Due: December 2 (Tuesday), 2025 at 11:59 am

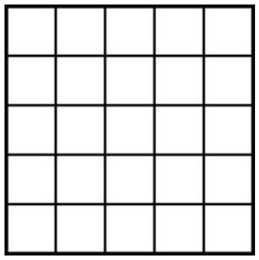
Please read the Common Projects Instructions before doing any problems.

The following is parallel processing:

Y. Deng (1992): "To make parallel processing efficient is to make all participants work smart not just hard…"

Picture source: Web (found in 2022).

Problem 5.1 (Earn 14 points each for the report and the presentation):
You are given $N = 15{,}000$ particles in a 50×50 2D box divided into a 5×5 grid of sub-boxes.



Any pair of Particles i and j interact via the Lenard-Jones potential

$$V_{ij} = \frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^{6}}$$

where $r_{ij} = |\boldsymbol{x_i} - \boldsymbol{x_j}|$ is the distance between Particles i and j. This potential is made to truncate at $r_c = 3$, i.e., $V_{ij}(r_{ij} > r_c) = 0$. The total energy for particle i is

$$E_i = \frac{1}{2}\sum_{j \neq i}^{N} V_{ij}$$

We establish a Cartesian system with origin at the lower-left vertex of the lower left-corner sub-box labeled as $(\alpha, \beta) = (1,1)$. And the upper right corner sub-box is labeled as $(\alpha, \beta) = (5,5)$ and, in each case, particles are randomly and uniformly distributed within each sub-box based on their labels (α, β).

Compute $E_i\ \forall\ i = 1, 2, \ldots, N$ for the following three case with $P = 25$ cores. For each case, report two sets of results: (a) a 2D heatmap visualizing the total energy E_i across all particles, and (b) the execution time recorded by each of the cores for completing the calculations.

When computing E_i, achieve a quasi-load balance. You may use particle decomposition, spatial decomposition, or both. With spatial decomposition, internal sub-box vertices can be freely adjusted, while boundary vertices of the original domain must remain fixed.

Case 1: Sub-box (α, β) is assigned

$$n(\alpha, \beta) = N\frac{\alpha + \beta}{\sum_{\alpha,\beta=1}^{5}(\alpha + \beta)} = \frac{N}{150}(\alpha + \beta) = 100(\alpha + \beta)$$

For example, sub-box $(1,1)$ has $n(1,1) = 200$ and sub-box $(5,5)$ has $n(5,5) = 1{,}000$ etc.

Case 2: Sub-box (α, β) is assigned

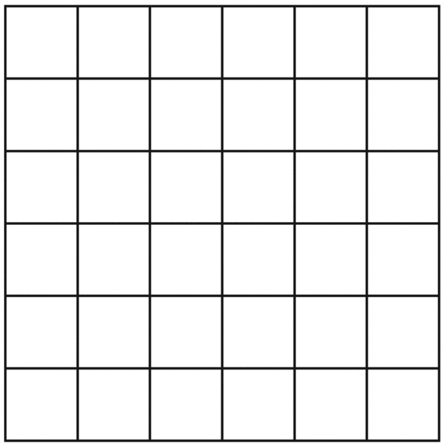
$$n(\alpha, \beta) = N\frac{|\alpha - \beta|}{\sum_{\alpha,\beta=1}^{5}|\alpha - \beta|} = \frac{N}{40}|\alpha - \beta|$$

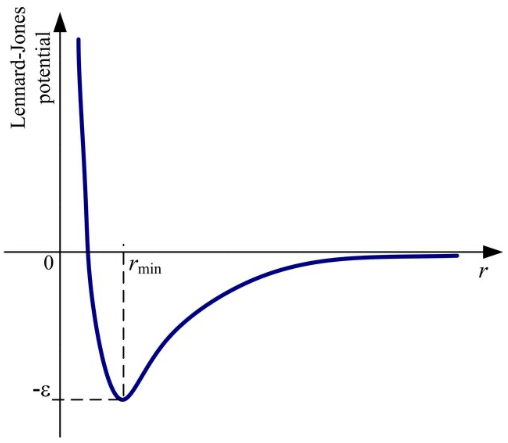
For example, sub-box $(1,1)$ has $n(1,1) = 100|1 - 1| = 0$, i. e., diagonal sub-boxes are empty.

Case 3: Sub-box (α, β) is assigned

$$n(\alpha, \beta) = N\frac{\alpha * \beta}{\sum_{\alpha,\beta=1}^{5} \alpha * \beta} = \frac{N}{225}\alpha * \beta$$

Problem 5.2 (Earn 14 points each for the report and the presentation):
Place $N = 36{,}000$ particles in a 60×60 2D box divided into a 6×6 grid of sub-boxes. Assign 100 and 1,900 particles to alternating adjacent sub-boxes in a checkerboard pattern, with particles distributed uniformly at random within each sub-box.





You generate the initial coordinates and velocities (floating-point numbers) of these particles with $\boldsymbol{x_i} = (x_i, y_i) \in (0, 60)^2$; and $\boldsymbol{v_i}\ \forall\ i = 1, 2, \dots, N$. For the initial $\boldsymbol{v_i}$, we assume $|\boldsymbol{v_i}| \sim U(0,1)$ and its angles $\alpha_i \sim U(0, 2\pi)$ where $U(a, b)$ indicates uniform random numbers in (a, b). Further, we assume the particles i and j interact with the Lenard-Jones potential (ignoring constants)

$$V_{ij} = \frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^{6}}$$

where $r_{ij} = |\boldsymbol{x_i} - \boldsymbol{x_j}|$ is the distance between Particles i and j. This potential is made to truncate at $r_c = 3$, i.e., $V_{ij}(r_{ij} > r_c) = 0$. The total energy for particle i is

$$V_i = \frac{1}{2}\sum_{j \neq i}^{N} V_{ij}$$

and the force is $F_i = -\nabla V_i$ and particles' motions follow Newton's law.

Please complete:

1. Use the given decomposition without any load balancing, employ P=36 cores to advance these N particles for 100 time steps with the given step size $h = 10^{-6}$. In this case, assign one core to each sub-box.
2. Create a scatter plot displaying the final positions of the particles after 100 time steps.
3. Record the execution time on the 36 cores for completing the 100 time steps (present results in a table or bar chart).
4. Use any method of your choice to balance the load quai-optimally and then repeat the above Steps 1, 2, 3. In Step 1, of course, you will have a new distribution of the load.

Problem 5.3 (Earn 14 points each for the report and the presentation):
This problem is the sequel to Problems 3.3 and 4.3. This problem focuses on optimizing the distributions of tasks in a 3-level Strassen matrix multiplication (MM) for matrices $A_{N\times N}$ and $B_{N\times N}$, where N is a large power of 2, e.g., $N = 2^{10} = 1024$.

3-Level Strassen MM: submatrix is further decomposed recursively into a grid of sub-sub-sub-matrices (s^3-mats) which results in 64 s^3-mats per input matrix (128 total for A and B). The 3-level Strassen algorithm requires $\mathbf{7^3 = 343}$ s3-mats MMs instead of the $\mathbf{8^3 = 512}$ needed in naïve MM.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{18} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{81} & a_{82} & \dots & a_{88} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & a_{23} & b_{24} \\ b_{31} & b_{32} & a_{33} & b_{34} \\ b_{41} & b_{42} & a_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{18} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ b_{81} & b_{82} & \dots & b_{88} \end{bmatrix}$$

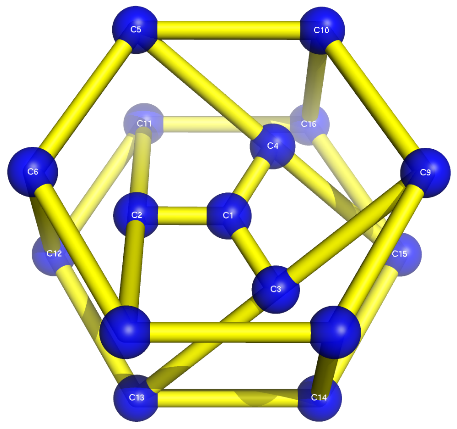
Three-level Strassen → 64 s^3-matrices per input matrix → 343 s^3-MMs (vs 512 naïve).

Tasks:
1. Distribute the 343 s^3-MMs quasi-optimally across the 7 cores and justify (without formal proof) why your distribution is near-optimal.
2. After distribution, designate 4 s^3-matrices as permanent locals in the 7 cores; design a communication protocol to fetch the remaining s^3-matrices needed to complete each core's 49 local s^3-MMs.
3. Implement the algorithm and report per-core times for communication T_{Comm} and computation T_{Comp}.
4. Repeat the above three steps except changing to designate 8 s^3-matrices as permanent locals

Results:

Permanent locals	4 s^3-matrices							8 s^3-matrices						
Core	1	2	3	4	5	6	7	1	2	3	4	5	6	7
T_{Comm}														
T_{Comp}														
T_{Total}														

Problem 5.4 (Earn 14 points each for the report and the presentation):



In the given 3D graph, the 16 blue vertices are identical, and the edges are undirected. This graph defines the network topology of a fictitious supercomputer.

Tasks: You generate a matrix $A_{N\times N} \sim U(-1, 1)$ and a vector $b_{N\times 1} = (1,1, \dots 1)^T$ and carry out the following tasks for three cases: $N = 16 \times 8$, 16×16 and 16×32

(1) Design a method to map your operations onto a fictitious supercomputer with the given network topology.
(2) Implement your method on a real supercomputer, such as Seawulf, with 16 cores to emulate the topology with reasonable fidelity.
(3) Report the timing results from the actual implementation (2) where you need to report the communication T_{Comm} and computation T_{Comp} separately.

Results:

N	16×8				16×16				16×32			
at Core	1	2	…	16	1	2	…	16	1	2	…	16
T_{Comm}												
T_{Comp}												
T_{Total}												

Note, in the unlikely case of your using iterative method, perform sufficient iterations to achieve reasonable accuracy. In all cases, you must verify that the solution is correct.

Problem 5.5* (Earn 14 points each for the report and the presentation):

In this project, you will design and implement a parallel inverse Fast Fourier Transform (IFFT) and make two applications:

- Reconstruction of a dense 2D MRI image from complex k-space data.
- Denoising and recovery of a 1D time-varying noisy signal.

A comprehensive analysis of the computational and communication performance for multiple decompositions and reveal the scaling of your parallelization.

Problem Description:
You are given (or may generate synthetically if needed) a complex-valued MRI k-space dataset

$$K \in C^{N\times N} \text{ where } N = 2^{11}$$

representing a fully sampled 2D Cartesian acquisition. The corresponding image is obtained by the 2D inverse Fourier transform.

Tasks:
(1) Complete the inverse FFT on $P = 2^6$ cores and report the computational and communication times for each of P cores.
(2) Design a quasi-optimal load balancing strategy so that tasks are more evenly mapped to the cores and report, again, the computational and communication times for each of $P = 2^6$ cores. Make a comparison with the timing results obtained in Task-1.
(3) Repeat the above Task-2 experiments on $P = 2^0, 2^1, \dots, 2^6$ cores and perform scaling analysis.
(4) Perform a 1D FFT (and relevant timing analysis) on noisy time-series data

$$s(t) \in R^M \text{ where } M = 2^{21}$$

with additive white Gaussian noise such that SNR (signal-to-noise ration) ≈10 dB.

Note: This project is adapted from B. Malloy's proposal.