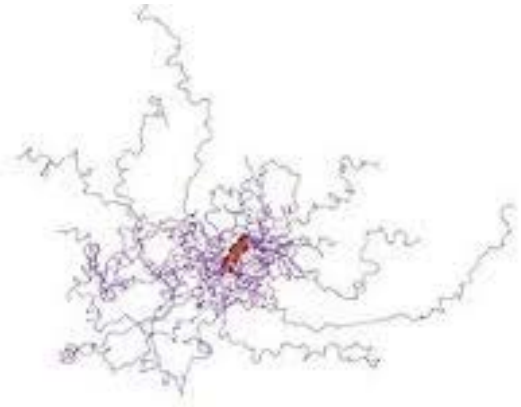
Parallelization of Atom-Based Molecular Properties using MPI

Annie Bao, University of Maryland, College Park
Joseph E. Curtis, NIST Center for Neutron Research

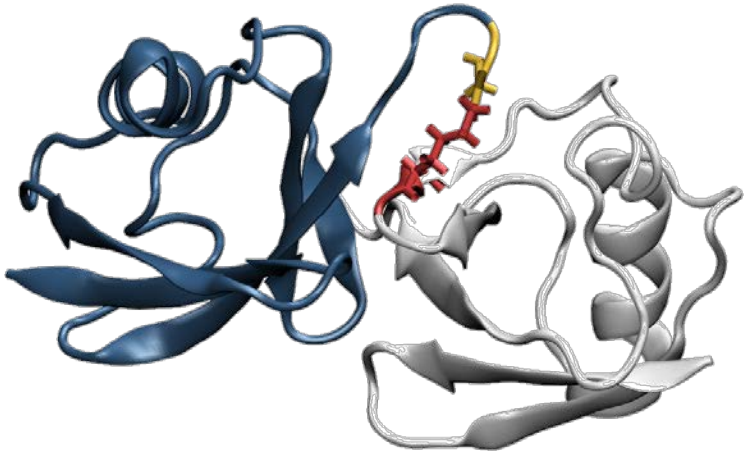
Modeling Molecular Systems

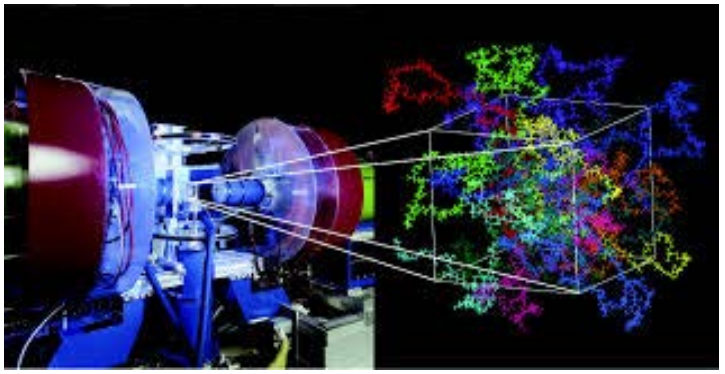
- Modeling biological molecules
  - Nucleic acids
  - Proteins
  - Lipids
- Characterizing structure
  - Better understand their role in cell signaling and diseases
  - Future applications such as drug development



Modeling Molecular Systems

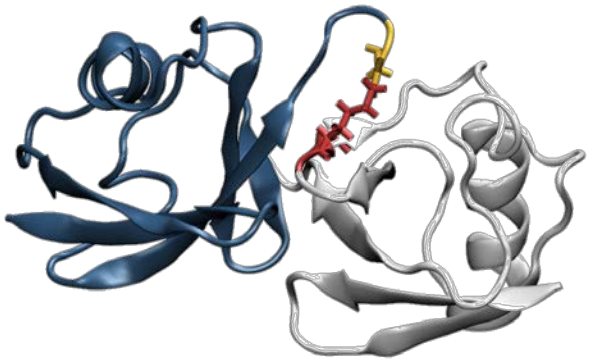
- Small angle scattering (SAS) techniques
  - Used to generate low resolution molecular shapes
- Small angle scattering data traditionally modeled using analytical equations and geometric shapes (NCNR Igor Macros, SasView, …)
- Moving towards using atomistic coordinates, “trajectories”, to model SAS data





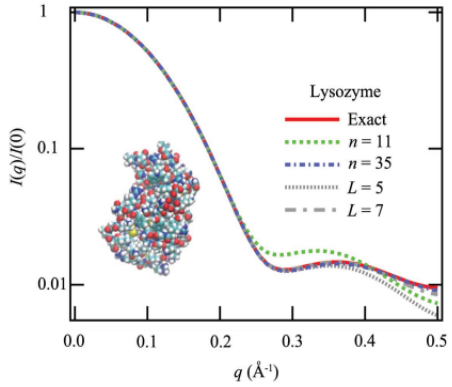
Analyzing Neutron Scattering Data

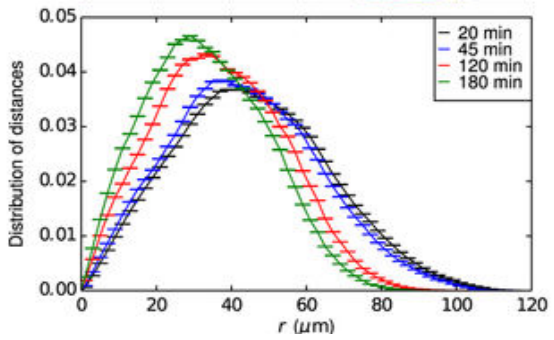
- SASSIE
  - Software to generate trajectories using molecular simulations
  - Calculate scattering profiles from atomistic coordinates
  - Compare calculated scattering to experimental data
  - Over 20 Python, C++ based modules
  - Open source



Atom-Based Molecular Properties (ABMP)

- Calculations used to determine shape of a molecule
- I(q) – scattering intensity at a given wave vector q
  - Amount of radiation scattered at any given angle
- p(r) - pair distribution function
  - Distribution of all pair distances within an object



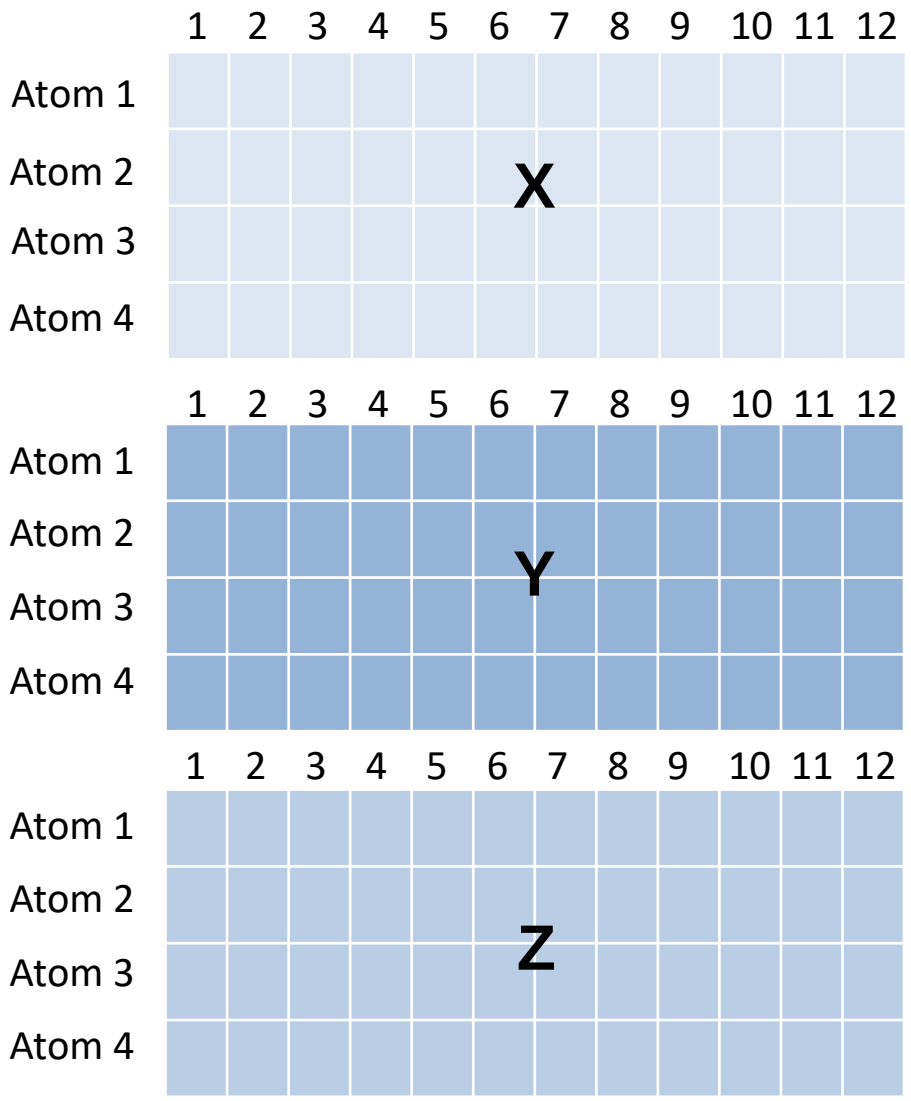


$$I(\mathbf{q}) = \left[\sum_j^N b_j \cos(\mathbf{q}\cdot\mathbf{r}_j)\right]^2 + \left[\sum_j^N b_j \sin(\mathbf{q}\cdot\mathbf{r}_j)\right]^2$$

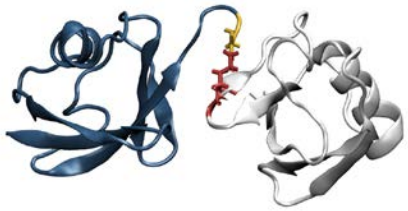
$$P(r) = \sum_{i,j=1,i\neq j}^{N} \|\vec{r_j} - \vec{r_i}\|$$

Using Atomistic Coordinates

- Using trajectory of atomistic coordinates
- Multiple frames of the molecular system over time
- Each frame contains 3D coordinate data for all atoms in the system



Using Atomistic Coordinates



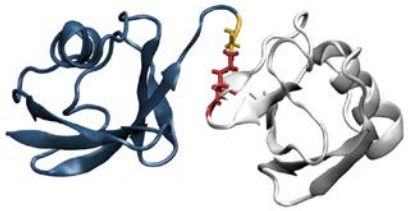
Frame 1

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.1											
Atom 2	2.2					X						
Atom 3	3.3											
Atom 4	4.4											

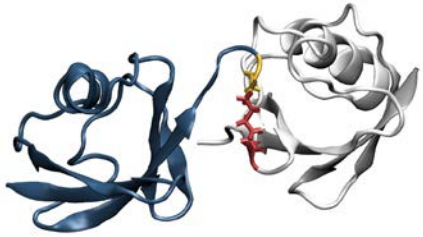
	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.5											
Atom 2	2.5					Y						
Atom 3	3.5											
Atom 4	4.5											

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.7											
Atom 2	2.7					Z						
Atom 3	3.7											
Atom 4	4.7											

Using Atomistic Coordinates



Frame 1



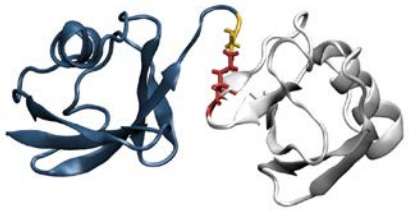
Frame 2

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.1	1.7										
Atom 2	2.2	3.5				X						
Atom 3	3.3	6.1										
Atom 4	4.4	8.1										

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.5	1.1										
Atom 2	2.5	2.1					Y					
Atom 3	3.5	7.1										
Atom 4	4.5	1.2										

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.7	2.2										
Atom 2	2.7	1.4					Z					
Atom 3	3.7	7.4										
Atom 4	4.7	2.5										

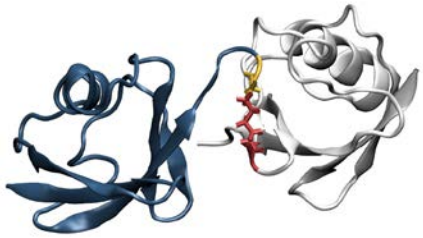
Using Atomistic Coordinates



Frame 1

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.1	1.7	.	.	.	.	.	.	.	.	.	.
Atom 2	2.2	3.5	.	.	.	.	.	.	.	.	.	.
Atom 3	3.3	6.1	.	.	.	.	.	.	.	.	.	.
Atom 4	4.4	8.1	.	.	.	.	.	.	.	.	.	.

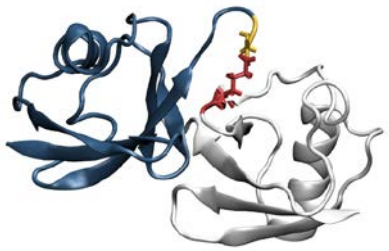
X



Frame 2

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.5	1.1	.	.	.	.	.	.	.	.	.	.
Atom 2	2.5	2.1	.	.	.	.	.	.	.	.	.	.
Atom 3	3.5	7.1	.	.	.	.	.	.	.	.	.	.
Atom 4	4.5	1.2	.	.	.	.	.	.	.	.	.	.

Y



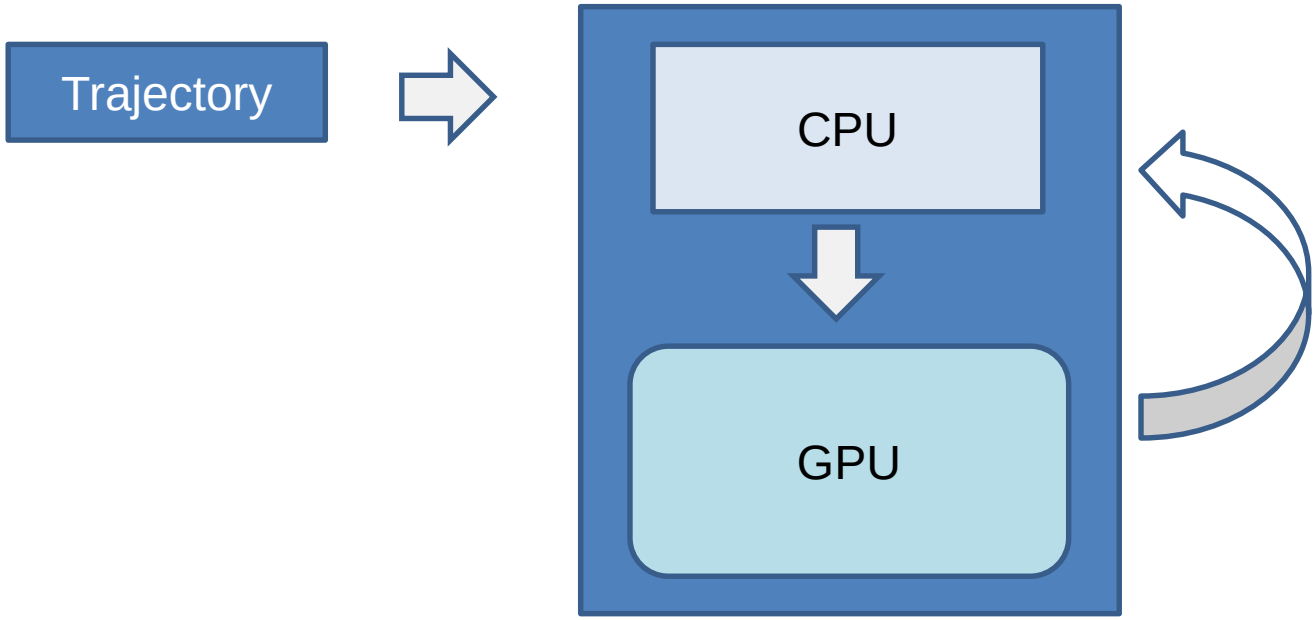
Frame 3

	1	2	3	4	5	6	7	8	9	10	11	12
Atom 1	1.7	2.2	.	.	.	.	.	.	.	.	.	.
Atom 2	2.7	1.4	.	.	.	.	.	.	.	.	.	.
Atom 3	3.7	7.4	.	.	.	.	.	.	.	.	.	.
Atom 4	4.7	2.5	.	.	.	.	.	.	.	.	.	.

Z

Current Workflow

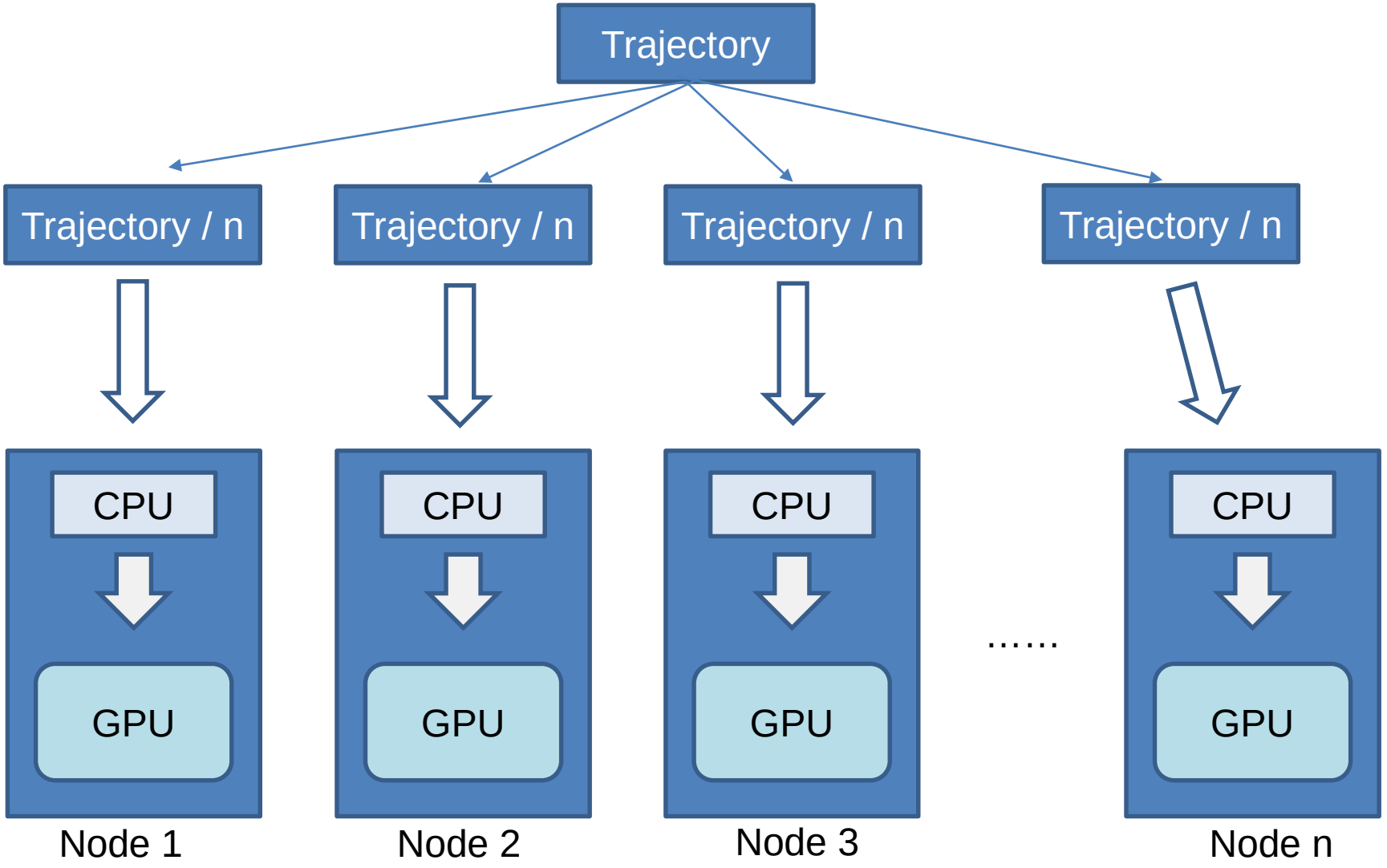
- Single node GPU programming
  - Program sent to one CPU
  - Trajectory is split into batches of frames
  - Each batch of frames is run on the GPU



Parallel Processing

- Calculations of large molecular systems can be slow
  - 10^4 – 10^6 atoms and 10^4 – 10^6 frames
- Further parallelization of the algorithms introduces speedup
- MPI
  - Library called by C++ and Fortran programs
  - Standardized means of exchanging messages between multiple computers running a parallel program across distributed memory

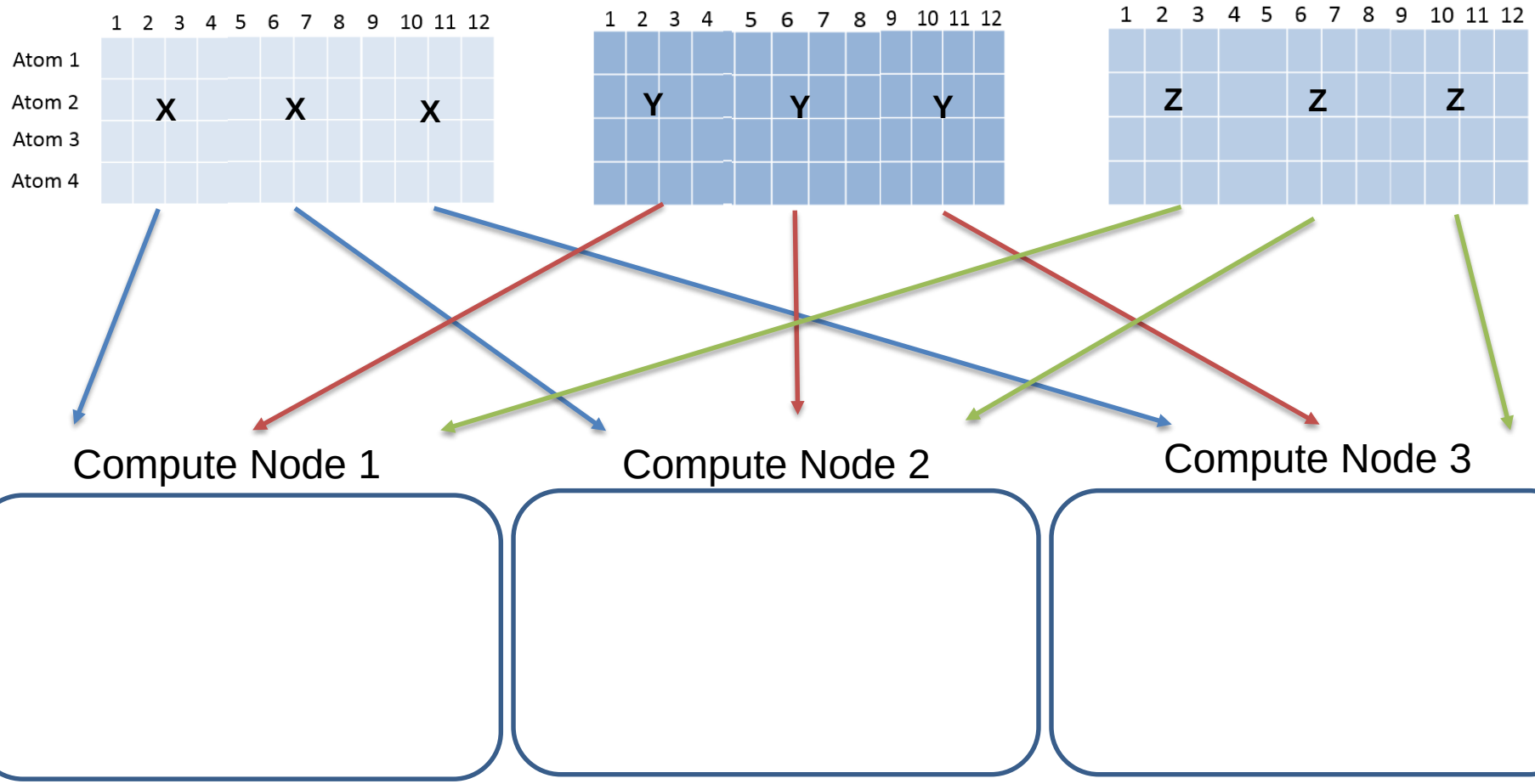
Parallel Processing



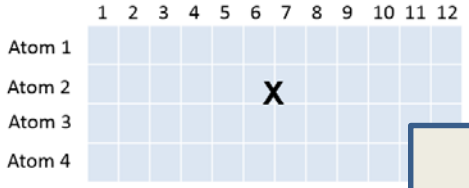
High Performance Computing

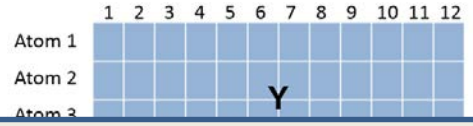
- Titan Cray XK7 Oak Ridge National Laboratory
  - 18,688 AMD Opteron compute nodes
  - 299,008 2GB Opteron Cores – 16 per node
  - 50,233,344 K20X Kepler GPU cores – 2,688 cores (1 GPU card) per node
- Dividing up trajectory evenly into N pieces to run on N nodes simultaneously

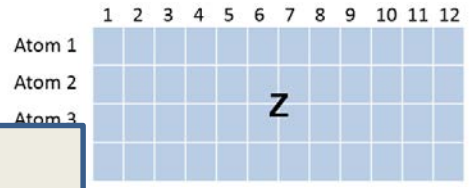
Implementation of MPI



Implementation of MPI

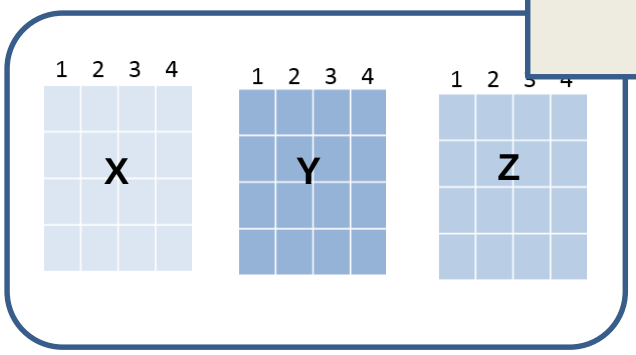


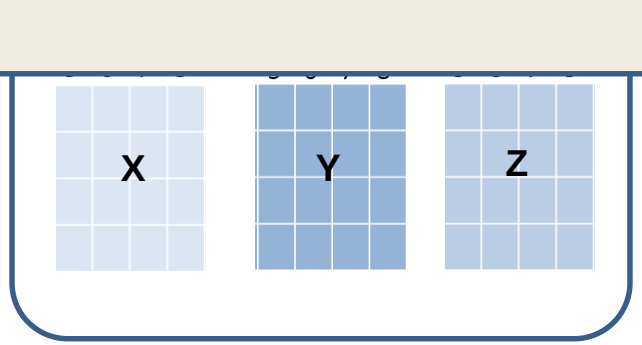


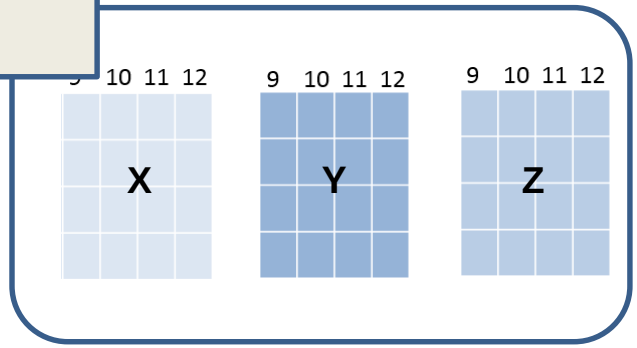


MPI_Send – sends batches of frames from node 0

MPI_Receive – each node accepts the batch of frames







Results

- Addition of the MPI layer shows linear speed increase in throughput
- Titan calculation of I(q) with 100,000 frames and 20,000 atoms
  - Single node GPU ~48 hours
  - 50 nodes w/ 50 GPU ~1 hour
- Over 400 registered users on Sassie
  - ½ of all compute time is used for I(q)
  - Most users require ~100,000 frames
- Using more nodes will take more time to run
- Optimization curve based on # nodes, # atoms, and # frames

Future Work

- Profiling the program to find bottlenecks, improve efficiency
- Porting the methodology into SASSIE (I(q) and P(r))
- Any property that can be calculated from the coordinates in a trajectory

Search:

Acknowledgments

- Joseph Curtis, NIST Center for Neutron Research
- Julie Borchers and Joe Dura, NCNR SURF Directors
- National Science Foundation, CCP-SAS (CHE-1265821) grant
- Engineering and Physical Sciences Research Council, grant EP/K039121/1
- Center for High Resolution Neutron Scattering (CHRNS)