GROMACS Imagine Programming System

Ankit Garg & Yanan Zhao

Eric Darve

Abhishek Das

Mattan Erez

GROMACS: Molecular Dynamics

- Simulate the dynamics of large molecules (like proteins,DNA) by solving Newton's equation of motion for the atoms.

$$\frac{d^2 r_i}{dt^2} = \frac{F_i}{m_i}$$

- Equations of motion integrated in time to get position of all particles at later times.

Calculating the forces

- Several contributions to the force
  - Bond vibrations between atoms
  - Angle vibrations
  - Torsional/dihedral angles
  - Nonbonded electrostatics & Lennard-Jones between all atoms that are close in space

Nonbonded forces

- Accounts for 90-95% of the runtime in C/Fortran code
- Most common form:

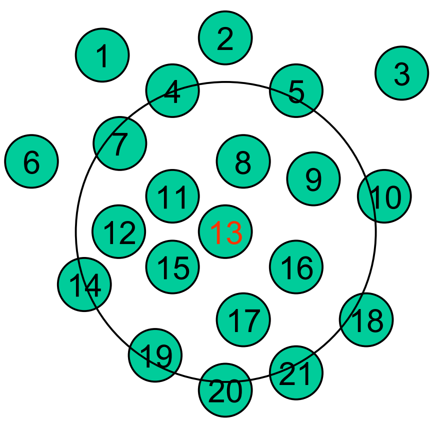
$$V_{nb} = \sum_{i,j} \left[\frac{1}{4\pi\epsilon_0} \frac{q_i \, q_j}{r_{ij}} + \left(\frac{C_{12}}{r_{ij}^{12}} - \frac{C_6}{r_{ij}^{6}} \right) \right]$$

Electrostatics

Lennard-Jones

Neighbor lists

- Neighbor list constructed every 10 steps.
- In practice: 10,000-100,000 atoms, with 100-200 neighbors in each list



Neighbor list for 'center atom' 13 = { 8, 9, 11, 12, 15, 16, 17}

- In our problem, we have totally 900 water molecules, the 3 atoms in each molecule are indexed in the same order: O, H1, H2, so instead of atom-atom interaction, we use molecule-molecule interaction.

- Every center molecule have 16 neighbors:
  - If: molecule has < 16 real neighbors
    Then: dummy neighbors(#901) are added
    If: molecule has > 16 real neighbors
    Then: molecule is repeated in the center molecule list (iinr)

- original molecule list:

0	1	2	3	…	…

- original position list-pos:

pos(0)	pos(1)	pos(2)	pos(3)	…	…

- center molecule list-iinr: (for convenience, from this slide, assume each center molecule has 4 neighbor molecules instead of 16 as in real computation)

0	0	1	1	1	2	…	…	…

- neighbor list-jjnr:

5	3	2	7	8	4	900	900	….	….	….	6	5	10	900	…	…	…

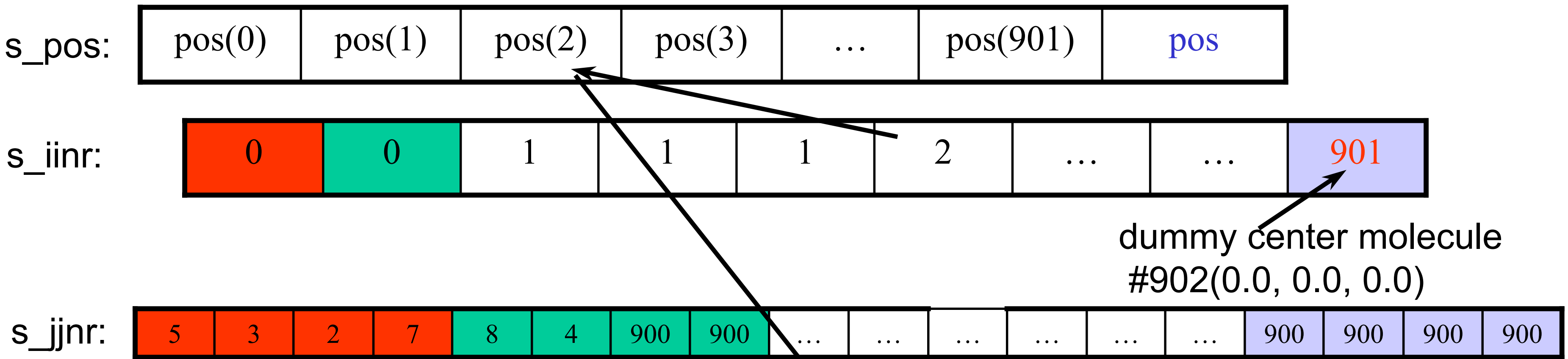
faraway dummy molecule #901

The SSS Programming System

- Borrowed from Imagine Programming System---
  a stream processor
- Operates on sequences of data records called streams
- Two Parts:
  - kernels: operations within individual records of a stream (KernelC)
  - stream program: (stream C)
    - streams definition
    - high-level control
    - data flow
- When Brook works, computation can be realized with the stream functions and operations in Brook, i.e. streamGather(), streamScatter(), streamGroup(), etc.

The Algorithm

1. Load the molecule and position arrays into streams (all streams must have a multiple of 16 records)



2. **Gather** positions to new streams indexed by iinr and jjnr

3. **Group** the position stream indexed by jjnr (s_pos_j) into a 2D stream

s_iinr:

0	0	1	1	1	2	…	901	901

s_jjnr_N:

5	8	.				…	.	900
3	4	.				…	.	900
2	900	.				…	.	900
7	900	.				…	.	900

4. Call the kernel

Returns a stream of forces (stream_forces_i) indexed by iinr

s_f_i:

s_f_i(0)	s_f_i(0)	s_f_i(1)	s_f_i(1)	s_f_i(1)	s_f_i(2)	…	s_f_i(901)	s_f_i(901)

5. Accumulate forces for molecules repeated in iinr and **scatter** these final forces back according to the original indexing scheme.

s_force:

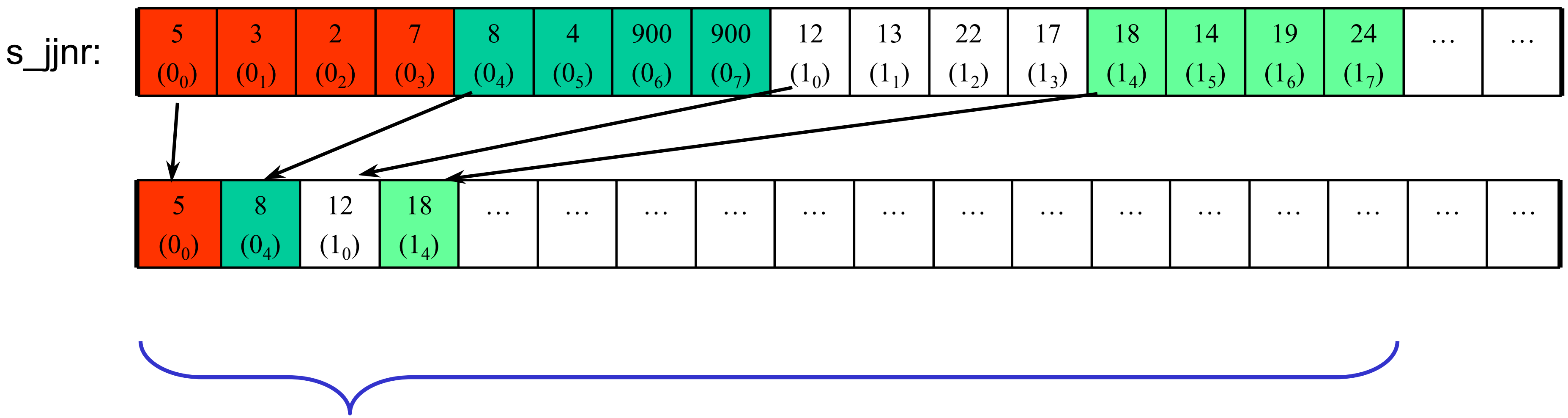
force(0)	force(1)	force(2)	force(3)	…	force(901)

What do we do in the kernel?

In each cluster #1 to #16

- read the next center molecule from iinr
- read all of its neighboring molecules from jjnr
- calculate force due to molecular interaction between the center molecule and all of its neighbors
- Accumulate the total force on the center molecule
- Output total force to the force stream

How the Kernel Works



cluster	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	…	3	4
s_iinr	0	0	1	1	1	2	3	3	3	3	4	4	5	5	5	…	901	901
s_jjnr	0_0	0_4	1_0	1_4	1_8	2_0	3_0	3_4	.	.	.	.	.	.	.	…	.	.
	0_1	0_5	1_1	1_5	1_9	2_1	3_1	3_5	.	.	.	.	.	.	.	…	.	.
	0_2	0_6	1_2	1_6	1_{10}	2_2	3_2	3_6	.	.	.	.	.	.	.	…	.	.
	0_3	0_7	1_3	1_7	1_{11}	2_3	3_3	3_7		.	.	.	.	.	.	…	.	.

Recap – KernelC and StreamC

KernelC

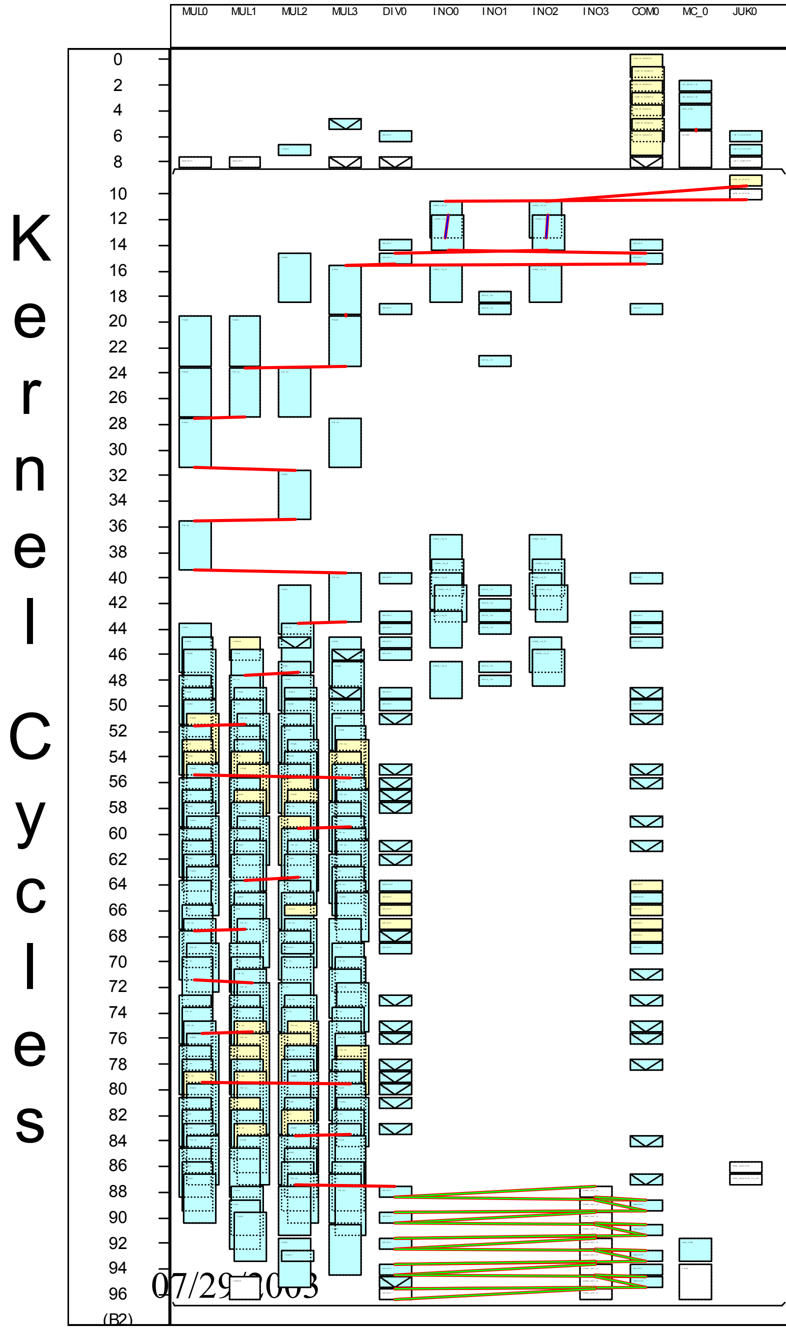
- Each cluster gets data for a center molecule from **iinr** along with all of its neighbors.
- Independently, clusters calculate the forces due to interactions between the molecules with their neighbors, update the stream of forces.

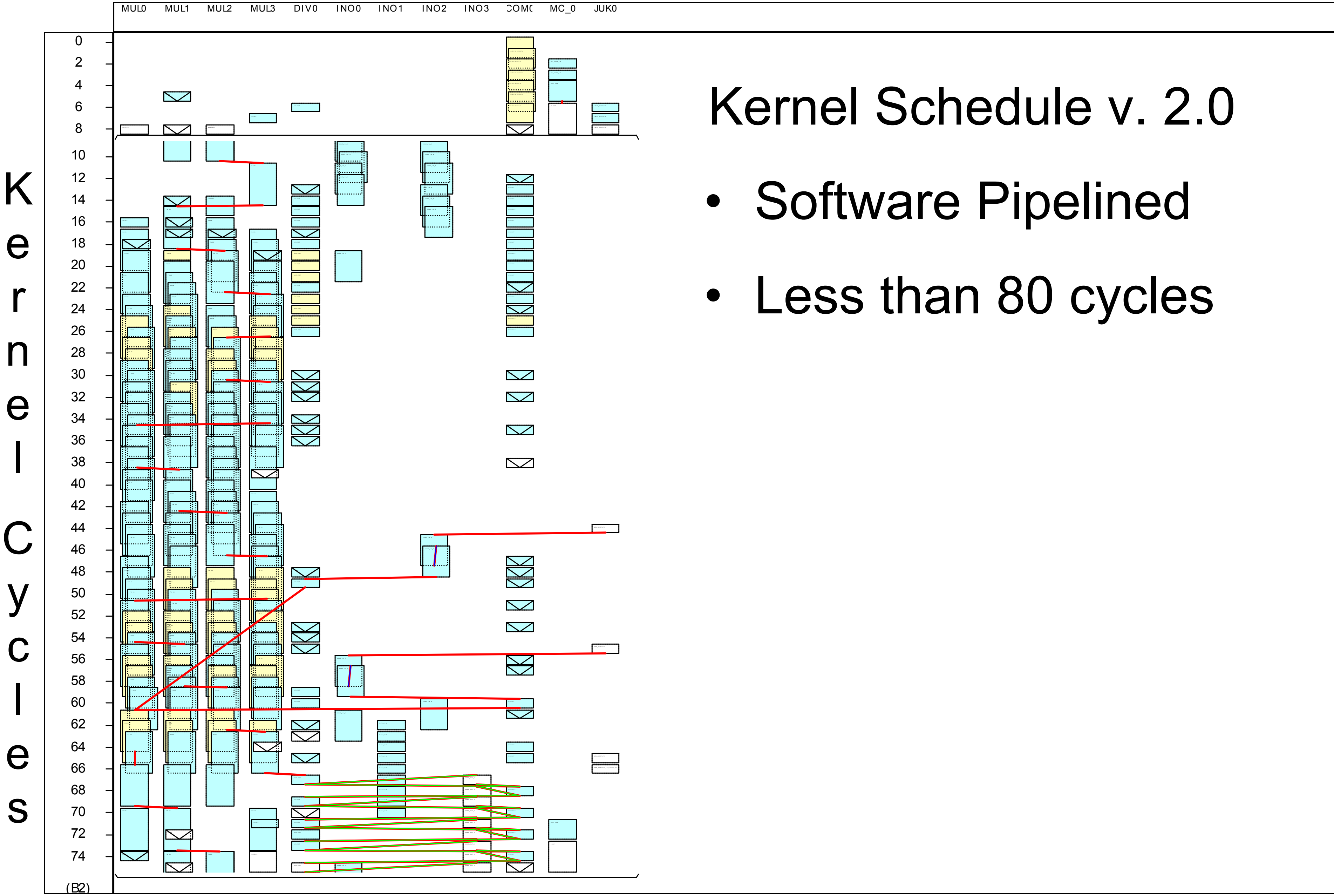
StreamC

- Read in GROMACS data (**iinr**, **jjnr**, original molecule positions, etc).
- Convert data into streams
- Gather and Group
- Call kernel.
- Scatter and Add
- Output the forces

Kernel Schedule v. 1.0

- No optimizations
- Around 100 cycles





Kernel Schedule v. 2.0

- Software Pipelined
- Less than 80 cycles

Next step:

- Further optimizations
  - Unrolling: combine loop iterations to give scheduler more flexibility in optimization.
- Multi-step calculations
  - Optimizations for stream program.