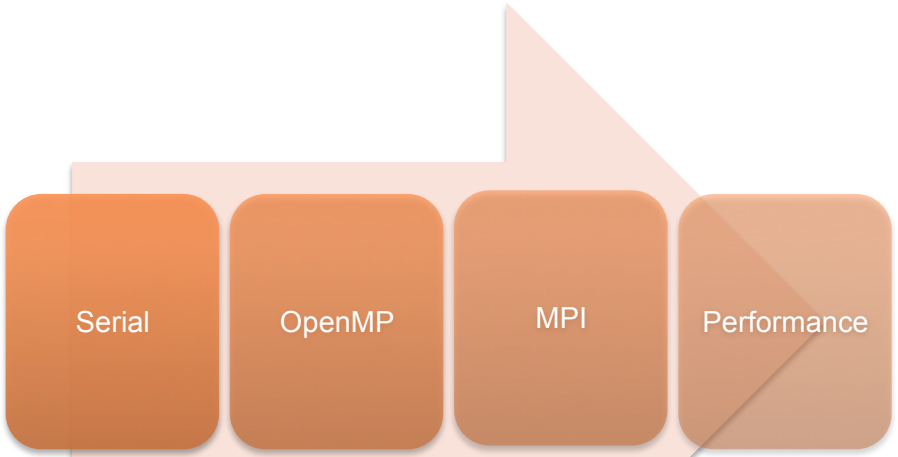
Summer Petascale School Exercises

Applications

Shawn Brown
Pittsburgh Supercomputing Center

Concepts

- General exercises bringing together concepts that have been lectured about.
- Mimicking the experience of a parallel developer
- These examples are meant to challenge you.



PSC
PITTSBURGH SUPERCOMPUTING CENTER

Solving Laplace's Equation via Jacobi Iteration

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0$$

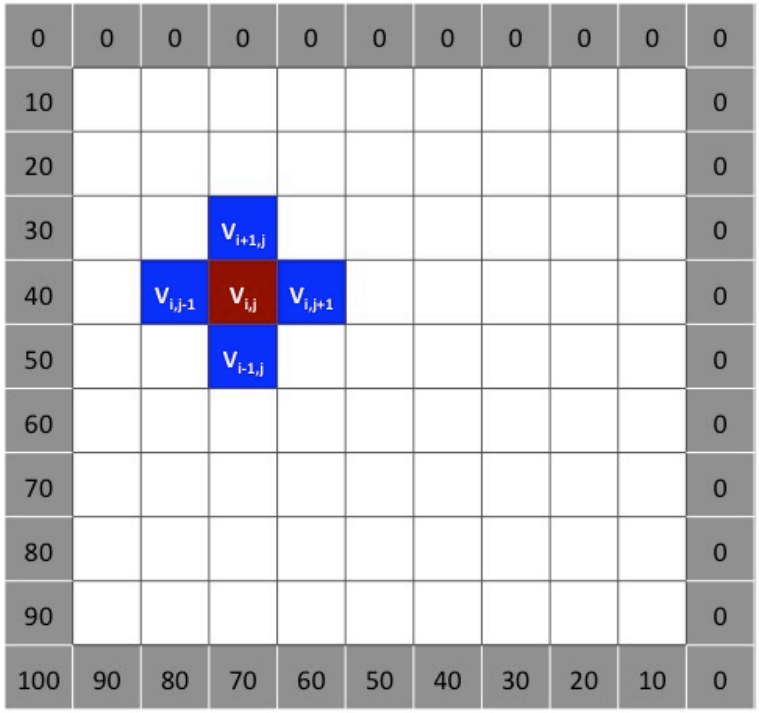


Figure 1: A diagram of the Jacobi Relaxation for Solving the Laplace's Equation on an evenly spaced 9x9 grid with the boundary conditions outlined in the text above.

$$V_{i,j}^{new} = 0.25(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1})$$

PSC
PITTSBURGH SUPERCOMPUTING CENTER

1-D and 2-D Decomposition

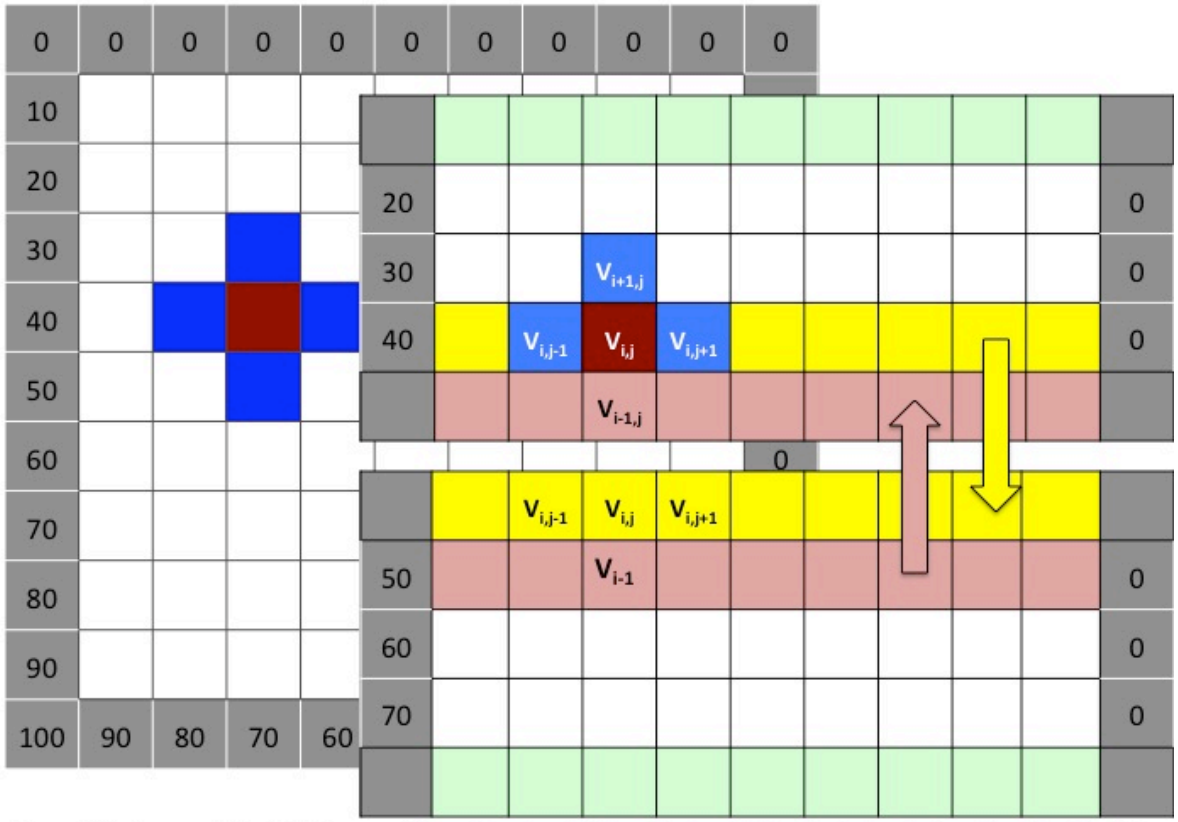
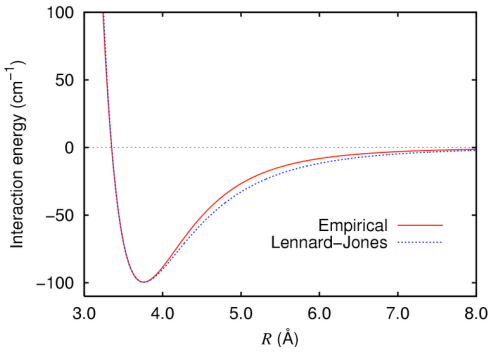


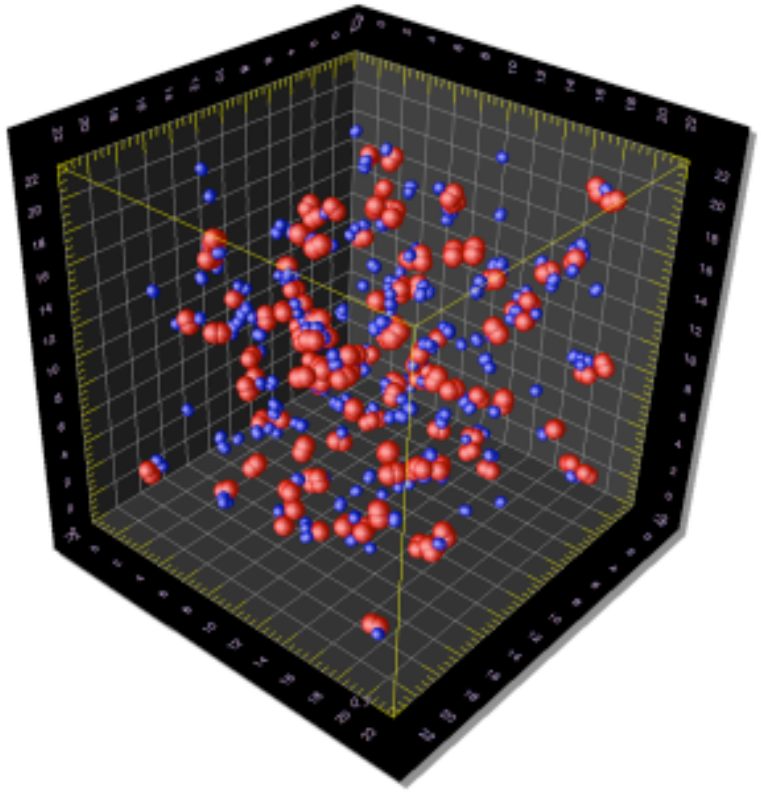
Figure 2: A diagram of the 1-D decomposition of the Jacobi Relaxation for Solving the Laplace's Equation, showing that the boundary elements that need to be communicated between processors.

PSC
PITTSBURGH SUPERCOMPUTING CENTER

Molecular Dynamics

$$V(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$





PSC
PITTSBURGH SUPERCOMPUTING CENTER

Domain Decomposition

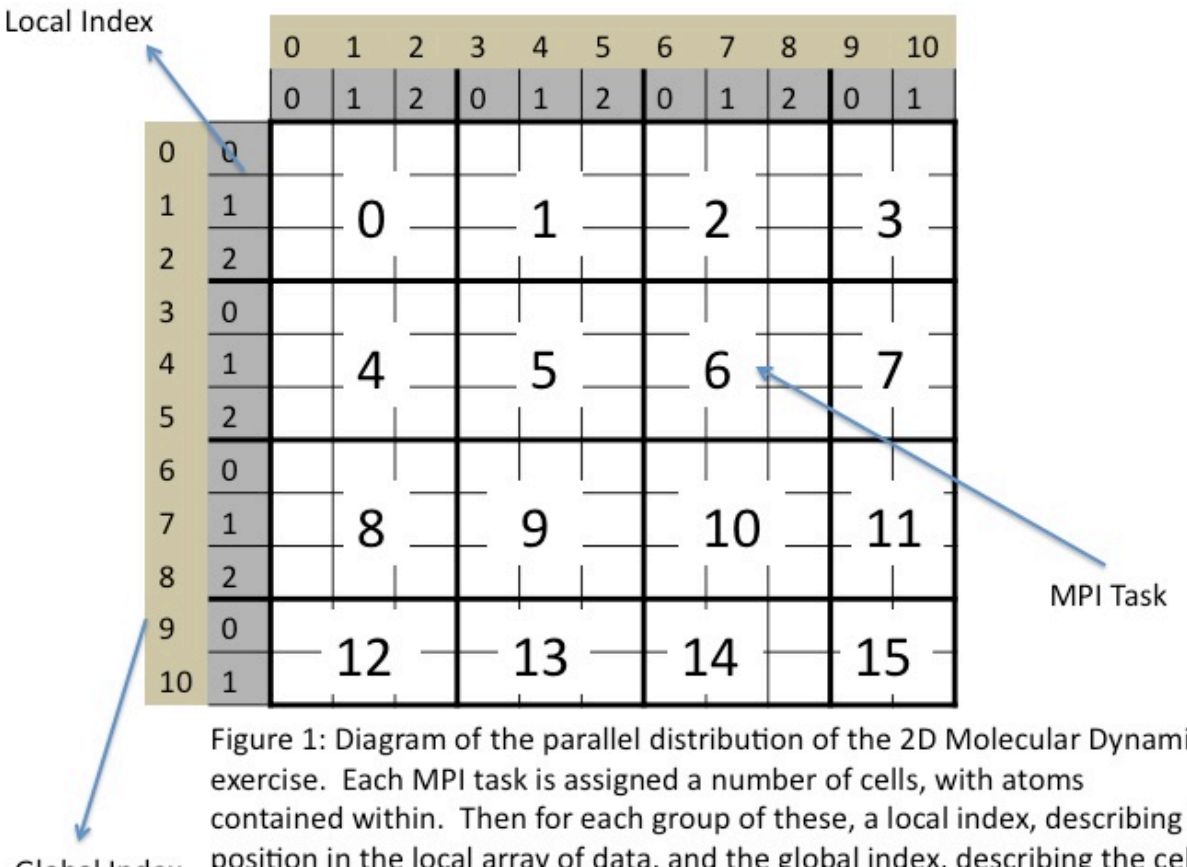


Figure 1: Diagram of the parallel distribution of the 2D Molecular Dynamics exercise. Each MPI task is assigned a number of cells, with atoms contained within. Then for each group of these, a local index, describing its position in the local array of data, and the global index, describing the cells overall position in the full matrix, are determined. If one can define these three sets of coordinates for each cell, then no matter where you are at in the algorithm, you will be able to find the data.

5 send its coordinates to 0,1,2,4,6,8,9,and 10

It also receives from them.

PSC
PITTSBURGH SUPERCOMPUTING CENTER

General Tips

- This is a petascale workshop…
  - You want to write a code that will scale to 1000s or cores, not 16.
  - No synchronous communication!
- Design before you write
  - These examples have some tricky indexing, thinking about it before you write code will make a big difference
- Work on small examples
  - Then scale up!

```
#include "mpi.h"
#include <stdio.h>
int main(int argc, char *argv[])
{
    int myid, numprocs, left, right;
    int buffer[10], buffer2[10];
    MPI_Request request, request2;
    MPI_Status status;

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    right = (myid + 1) % numprocs;
    left = myid - 1;
    if (left < 0)
        left = numprocs - 1;
    MPI_Irecv(buffer, 10, MPI_INT, left, 123, MPI_COMM_WORLD, &request);
    MPI_Isend(buffer2, 10, MPI_INT, right, 123, MPI_COMM_WORLD, &request2);
    MPI_Wait(&request, &status);
    MPI_Wait(&request2, &status);
    MPI_Finalize();
    return 0;
}
```

Development

- You don’t need the supercomputer to program MPI.
  - Most of you have a parallel machine on your desktop.
  - If you have compilers and development tools on your machines, you can write and test at small task counts.
- Move to supercomputing to test the large jobs.

Development

- To develop on the supercomputer, you can use interactive job submission for compiling and debugging.
  - For Athena: qsub –I
  - For Ranger: I am working on it, I will post.