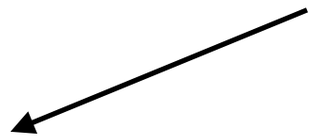
Message-Passing Interface Standard (MPI)

Material from Big Data Workshop organized by XSEDE
along with the Pittsburgh Supercomputing Center. Permission to reuse exercise
and slides from John Urbanic at PSC.

See upcoming events here:
https://www.psc.edu/index.php/xsede-hpc-series-all-workshops

Hello world example

Generates a.out output file



```
mpicc hello.c
mpirun -n 8 a.out

  Hello from 5.
  Hello from 1.
  Hello from 6.
  Hello from 0.
  Hello from 4.
  Hello from 7.
  Hello from 2.
  Hello from 3.
```

Hello world example

and now in Python…

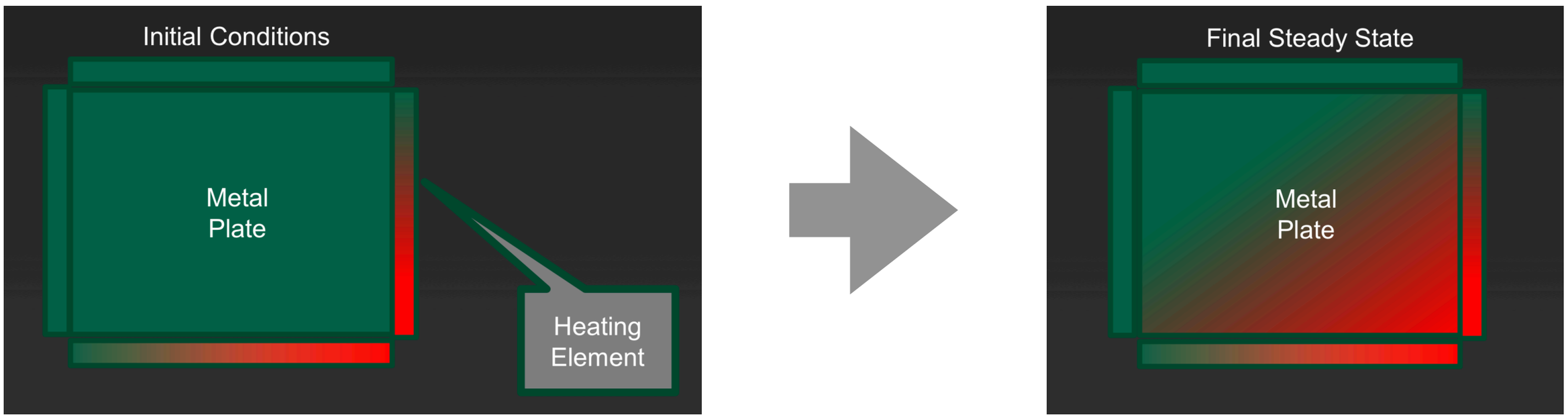
No output file generated

```
mpiexec -n python 4 ./hello.py

   Hello from 5.
   Hello from 1.
   Hello from 6.
   Hello from 0.
   Hello from 4.
   Hello from 7.
   Hello from 2.
   Hello from 3.
```

Exercise

Solving Laplace equation on grid



From “Introduction to OpenMP” slides by John Urbanic (Parallel Computing Scientist @ Pittsburgh Supercomputing Center)

Exercise

Solving Laplace equation on grid

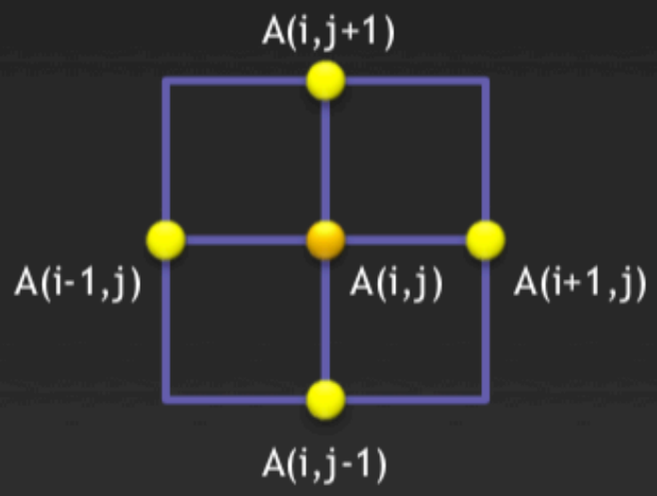
$$\nabla^2 \Phi(x,y) = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \qquad \text{(2D)}$$

Steady state heat equation for temperature!

http://www.dzre.com/alex/P441/lectures/lec_20.pdf

Exercise

Solving Laplace equation on grid



$$A_{k+1}(i,j) = \frac{A_k(i-1,j) + A_k(i+1,j) + A_k(i,j-1) + A_k(i,j+1)}{4}$$

"Relaxation method"

From "Introduction to OpenMP" slides by John Urbanic (Parallel Computing Scientist @ Pittsburgh Supercomputing Center)

Exercise

Solving Laplace equation on grid

C code

```
/************************************************************
 * Laplace MPI C Version
 *
 * T is initially 0.0
 * Boundaries are as follows
 *
 *                T                        4 sub-grids
 *   0  +--------------------+  0    +--------------------+
 *      |                    |       |                    |
 *      |                    |       |--------------------|
 *      |                    |       |                    |
 *   T  |                    |  T    |--------------------|
 *      |                    |       |                    |
 *      |                    |       |--------------------|
 *      |                    |       |                    |
 *   0  +--------------------+ 100    +--------------------+
 *      0         T       100
 *
 * Each PE only has a local subgrid.
 * Each PE works on a sub grid and then sends
 * its boundaries to neighbors.
 *
 *  John Urbanic, PSC 2014
 *
 ************************************************************/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <sys/time.h>
#include <mpi.h>

#define COLUMNS      1000
#define ROWS_GLOBAL  1000        // this is a "global" row count
#define NPES            4        // number of processors
#define ROWS (ROWS_GLOBAL/NPES)  // number of real local rows

// communication tags
#define DOWN     100
#define UP       101

#define MAX_TEMP_ERROR 0.01

double Temperature[ROWS+2][COLUMNS+2];
double Temperature_last[ROWS+2][COLUMNS+2];
```

Python version?

From “Introduction to OpenMP” slides by John Urbanic (Parallel Computing Scientist @ Pittsburgh Supercomputing Center)

Getting mpiexec executable

1. Download Open MPI software from https://www.open-mpi.org/software/ompi/v3.1/
   (I used these instructions for Mac:
   https://wiki.helsinki.fi/display/HUGG/Open+MPI+install+on+Mac+OS+X)
2. Install mpi4py with e.g. **`pip install mpi4py`**
   (I followed this tutorial to get started:
   https://mpi4py.scipy.org/docs/usrman/tutorial.html)

Good look-up sites:

MPI module: https://mpi4py.scipy.org/docs/apiref/mpi4py.MPI-module.html

Comm module: https://mpi4py.scipy.org/docs/apiref/mpi4py.MPI.Comm-class.html

From “Introduction to OpenMP” slides by John Urbanic (Parallel Computing Scientist @ Pittsburgh Supercomputing Center)

C -> Python

Startup MPI routines

In C:

```
#include <mpi.h>
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
```

In python:

```
from mpi4py import MPI
comm        =   MPI.COMM_WORLD
rank        =   comm.Get_rank()
size        =   comm.Get_size()
```

The following slides are an introduction to MPI
by John Urbanic
from the Big Data Workshop held on June 4-7 2018…