Parallel programming in Chapel

JUAN ZUNIGA
juan.zuniga@usask.ca

ALEX RAZOUMOV
alex.razoumov@westgrid.ca

slides and code examples at at ...

- Simple 2D heat (diffusion) equation

$$\frac{\partial T(x,y,t)}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

- Discretize the solution $T(x,y,t) \approx T_{i,j}^{(n)}$ with $i = 1, ..., n$ and $j = 1, ..., n$
- Imagine a metallic plate being constantly heated in one corner, for example $T_{1,1}^{(n)} = 1$
- Everywhere else the initial solution is $T_{i,j}^{(0)} = 0$
- Discretize the equation

$$\frac{T_{i,j}^{(n+1)} - T_{i,j}^{(n)}}{\Delta t} = \frac{T_{i+1,j}^{(n)} - 2T_{i,j}^{(n)} + T_{i-1,j}^{(n)}}{(\Delta x)^2} + \frac{T_{i,j+1}^{(n)} - 2T_{i,j}^{(n)} + T_{i,j-1}^{(n)}}{(\Delta y)^2}$$

- For simplicity assume $\Delta x = \Delta y = 1$
- Use $\Delta t = 1/4$ which is the upper limit of numerical stability
- The finite difference equation becomes

$$T_{i,j}^{(n+1)} = \frac{1}{4}\left[T_{i+1,j}^{(n)} + T_{i-1,j}^{(n)} + T_{i,j+1}^{(n)} + T_{i,j-1}^{(n)}\right]$$

- The objective is to find $T_{i,j}$ after a certain number of iterations, or when the system is in steady state)
- Once done, also try increasing the number of points in the grid to illustrate the advantage of parallelism

Chapel base language

Task parallelism

Data parallelism

- one
- two

Advanced language features

- one
- two

- one
- two