



Chapel in Astronomy

Nikhil Padmanabhan (Yale University)



In cooperation with:





Parallel Applications

Workshop

Alternatives to MPI+X
November 19th, 2021

Introduction

- Observational cosmologist
 - Use large galaxy surveys to constrain the underlying physics of the Universe - its initial conditions, and the nature of dark matter and dark energy
 - Inputs include both observational data as well as mock observations based on simulations.
 - Cosmologists are Bayesians - want to explore the posteriors, but likelihoods can often be expensive
- Not a traditional HPC use case, but analyses can quickly become computation-limited.
- Python has been the language of choice, normally supplemented with Cython/Numba/etc (and more recently Tensorflow/PyTorch etc).
 - Scaling : memory, multiple nodes
- Been using Chapel more and more in daily research, getting students more involved.
 - MPI bindings, initial prototype of c2chapel



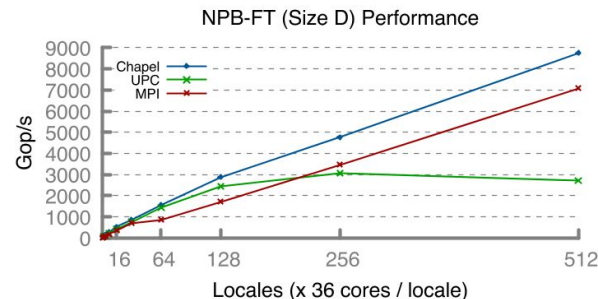
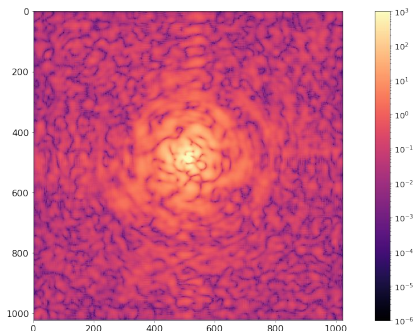
Parallel Applications

Workshop

Alternatives to MPI+X
November 19th, 2021

An example : chplUltra

- Explore the dynamics of ultralight dark matter.
- Solve a nonlinear Schrodinger equation
 - Pseudo-spectral solver
- History
 - Started as a port of a Python code
 - Initial shared memory version (directly interfaced to FFTW) easy, better performance compared to Python.
 - Multinode version required small changes to get working, reworked parallel FFT for performance.



NB: slab decompositions



Parallel Applications

Workshop

Alternatives to MPI+X
November 19th, 2021

Some thoughts

- Short-term research projects can be a fruitful ground for exploring alternatives.
 - Do not have the inertia of large legacy applications
- Big single multicore nodes now quite normal, nice to be able to scale out to multiple nodes/accelerators with small changes to code
- Lots of time/code spent doing “other” steps, important for the language to be expressive enough to handle these.
- Impedance matching with OpenMP+MPI.
- Interfacing with Python and friends.
- Community matters!