Spectral Navier Stokes with UPC++ and OpenMP

Samuel Olivier

Background

- Spectrally solve the Navier Stokes Equations:

$$\frac{\partial \vec{V}}{\partial t} = \vec{V} \times \vec{\omega} - \nabla \Pi + \nu \nabla^2 \vec{V}$$

$$\nabla \cdot \vec{V} = 0$$

where

$$\vec{\omega} = \nabla \times \vec{V}, \Pi = \frac{1}{2}V^2 + \frac{P}{\rho}$$

in a triply-periodic box

- Transform $\vec{V}$, Π, and $\vec{\omega}$ to Fourier space using a 3D Fast Fourier Transform (FFT)

$$\vec{V} = \sum_m \sum_n \sum_p \tilde{V}(m,n,p) e^{imx} e^{iny} e^{ipz}$$

- Laplacian example:

$$\nabla^2 V_i = \sum_m \sum_n \sum_p -\left(m^2+n^2+p^2\right) \tilde{V}_i(m,n,p) e^{imx} e^{iny} e^{ipz}$$

- To avoid expense of convolution in Fourier space, cross product is done in physical space
- Orthoganlity of Fourier modes $\Rightarrow$ equations decouple in m, n, p.
- Adams Bashforth/Crank Nicholson Fractional Step Method

$$\nabla^2 \tilde{G} = \nabla \cdot \left(\tilde{V}^n + \frac{\Delta t}{2}\left(3\tilde{V}^n \times \tilde{\omega}^n - \tilde{V}^{n-1} \times \tilde{\omega}^{n-1}\right) + \frac{\nu \Delta t}{2} \nabla^2 \tilde{V}^n \right)$$

$$\tilde{V}^{n+1/2} = \tilde{V}^n + \frac{\Delta t}{2}\left(3\tilde{V}^n \times \tilde{\omega}^n - \tilde{V}^{n-1} \times \tilde{\omega}^{n-1}\right) - \nabla \tilde{G}$$

$$\tilde{V}^{n+1} = \tilde{V}^{n+1/2} + \frac{\nu \Delta t}{2}\left(\nabla^2 \tilde{V}^{n+1} + \nabla^2 \tilde{V}^n\right)$$

where

$$\tilde{G} = \frac{\Delta t}{2}\left(3\tilde{\Pi}^n - \tilde{\Pi}^{n-1}\right)$$

- Parallelize by distributing across UPC++ ranks
  - Array operations (gradient, divergence, addition, subtraction) are all element-wise $\Rightarrow$ embarassingly parallel
  - Communication only required in global transpose of 3D FFT

Parallel FFT

- Store variables as distributed arrays (distributed in z)
- Apply serial FFT in each direction using 1D FFTW
- x, y are locally owned but z is distributed across UPC++ ranks
- Need all data in z direction to be local $\Rightarrow$ Global Transpose (switch to data being local in x, z and distributed in y)
  - Do all FFT's in y, send rows while doing FFT's in x
- Pencils: send a row at a time after finishing each serial FFT in x
- Slabs: transform a group of rows and send an x-y plane
- Use UPC++ one-sided communication (`rput`) to avoid synchronization
- Operate on local data in parallel with OpenMP
  - Do multiple 1D FFT's in parallel
  - Pencils sends messages from OpenMP parallel region (caught issue with UPC++ on Cori)

Results

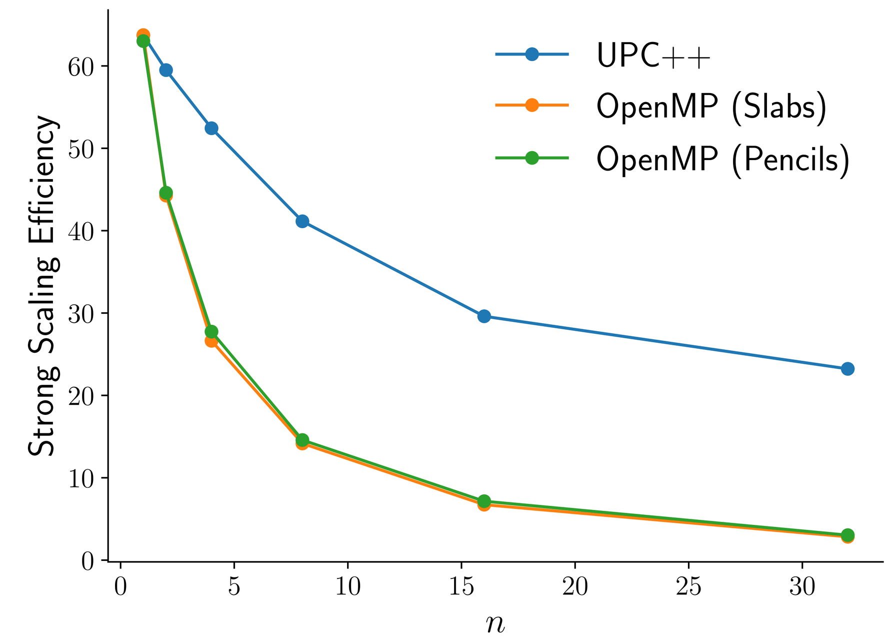

Fig. 1: Isolated strong scaling for UPC++ and OpenMP.

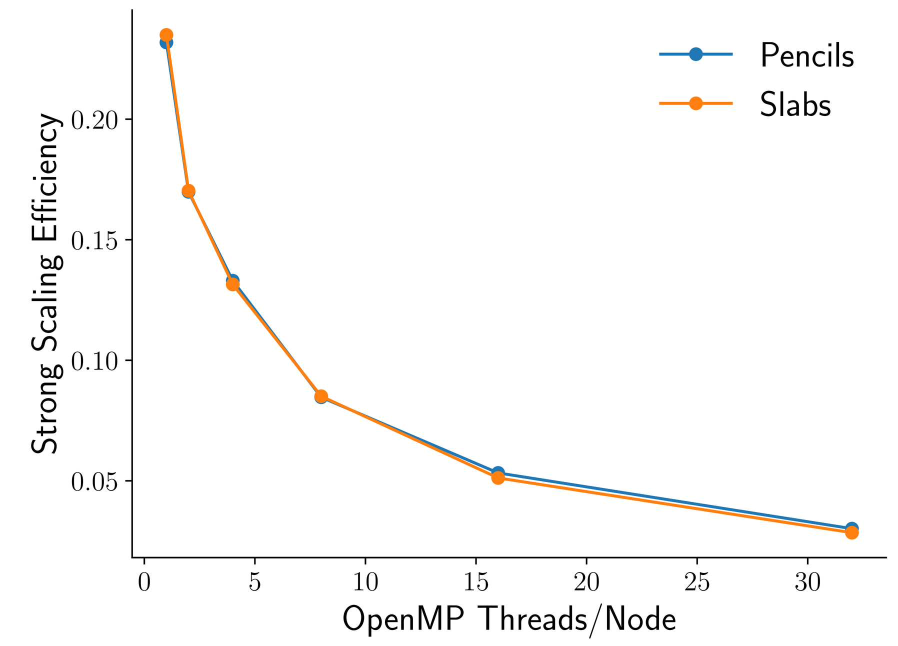

Fig. 2: Combined strong scaling UPC++ and OpenMP.

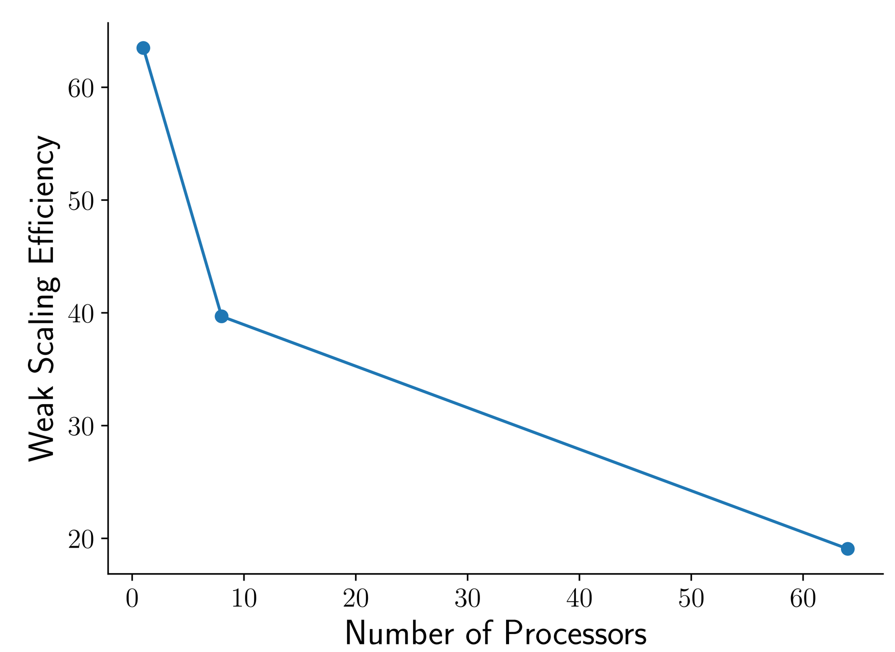

Fig. 3: Weak scaling for 1, 8, 64 processors.

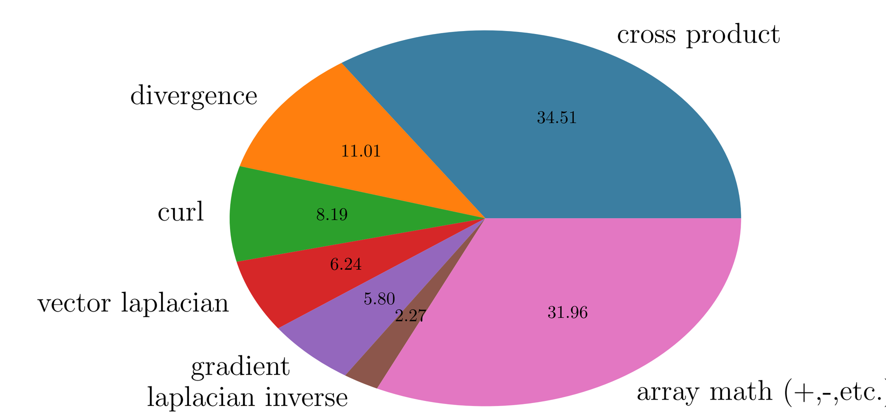

Fig. 4: Timing breakdown of kernels.

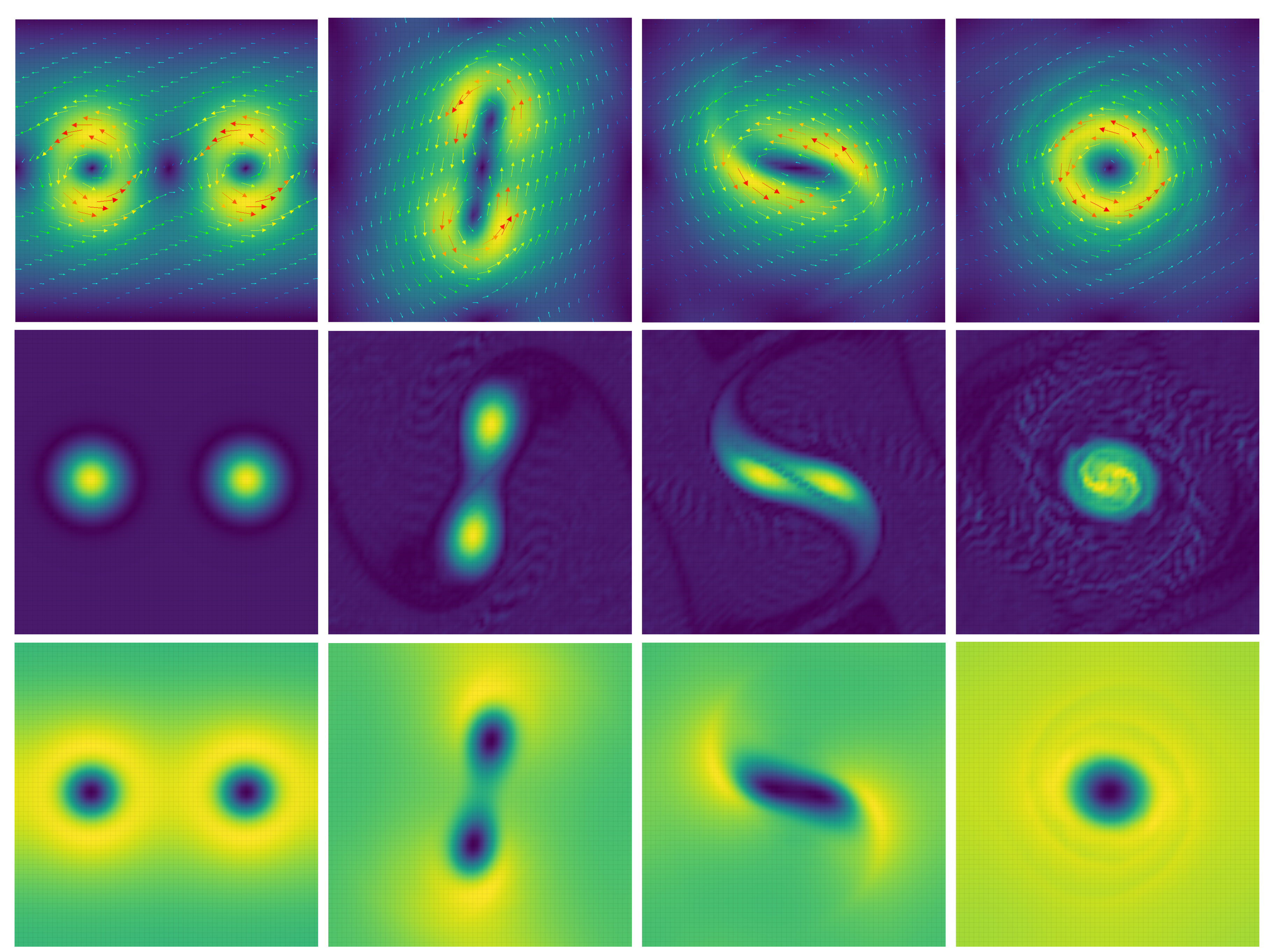

Fig. 5: Velocity (top), vorticity (middle), and pressure (bottom) at four time slices. Mesh size 64^3.

Conclusions

- Inter opting with OpenMP degrades performance
  - Fastest with all UPC++ ranks instead of balance between OpenMP and UPC++
  - OpenMP strong scaled poorly
    - May be from data contention between threads operating on same data in 1D FFT's
    - Unknown slow performance in embarrassingly parallel array operations
- Pencils and slabs performed nearly identically
- Poor weak scaling expected from all-to-all messages in FFT global transpose
- Additional transpose to make data contiguous in z did not improve performance

Future Work

- Test on more nodes
  - Slower messages across the network may make inter opting with OpenMP more favorable
  - Might expose a larger difference between pencils and slabs
- Investigate numerical issue with high resolution meshes ($N > 128^3$)
- Implement planes in global transpose
- Investigate OpenMP scaling issue

References

[1] M. Frigo, "A fast fourier transform compiler," *SIGPLAN Not.*, vol. 34, pp. 169–180, May 1999.

[2] C. Bell, D. Bonachea, R. Nishtala, and K. Yelick, "Optimizing bandwidth limited problems using one-sided communication and overlap," in *Proceedings of the 20th International Conference on Parallel and Distributed Processing*, IPDPS'06, (Washington, DC, USA), pp. 84–84, IEEE Computer Society, 2006.

[3] A. Chan, P. Balaji, W. Gropp, and R. Thakur, "Communication analysis of parallel 3d fft for flat cartesian meshes on large blue gene systems," in *Proceedings of the 15th International Conference on High Performance Computing*, HiPC'08, (Berlin, Heidelberg), pp. 350–364, Springer-Verlag, 2008.

[4] Bachan J, Baden S, Bonachea D, and Hargrove P, "UPC++ specification v1.0, draft 6," *Lawrence Berkeley National Laboratory Tech Report*, 2018.

[5] L. Dagum and R. Menon, "Openmp: An industry-standard api for shared-memory programming," *IEEE Comput. Sci. Eng.*, vol. 5, pp. 46–55, Jan. 1998.

[6] C. Canuto, *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1988.