Accelerating PETSc-Based CFD Codes with Multi-GPU Computing

Pi-Yueh Chuang and Lorena A. Barba

Mechanical and Aerospace Engineering, the George Washington University, Washington, DC, USA

Motivation

To enable multi-GPU computing for solving linear systems in PETSc-based applications.

- Run PETSc applications efficiently: exploit all available CPU and GPU (through NVIDIA's AmgX library) resources.
- PETSc and AmgX have their own data structures, requiring coding effort to couple the two libraries – we wrote a wrapper code, so you don't have to.
- The challenge is handling the situation where the PETSc-based code launches more MPI processes than the number of GPUs available, achieving heterogeneous speed-up.

AmgX and the wrapper

AmgX

A library developed by NVIDIA for multi-GPU iterative linear solvers, including Krylov methods and multigrid.

- Currently the only publicly available multigrid implementation exploiting multiple GPUs
- Free for non-commercial use! But closed source.

AmgX wrapper

With our wrapper, you only need two methods. All MPI communications, sub-system merging and data conversion are taken care of.

Simple usage

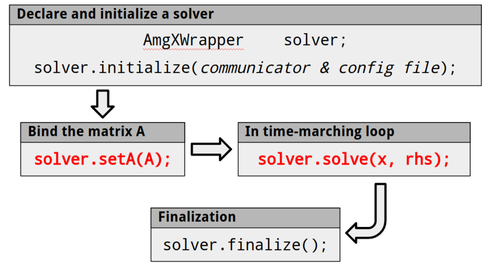

Fig. 1 With our wrapper, calling AmgX solvers in PETSc applications is simple: only `setA(A)` and `solve(x, rhs)` are required. `A`, `x`, and `rhs` are PETSc parallel matrix and vectors.

Sub-system merging on host

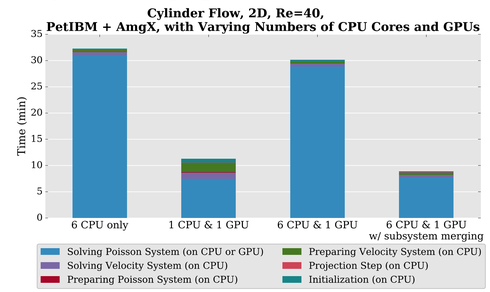

Fig. 2 Feature of sub-system merging in our wrapper: launching PetIBM on a system with more CPU cores than GPUs, there will be no significant run time penalty when using GPUs (compare the 3rd and 4th bars from left).

Benchmark: Poisson system

Assuming good scalability on CPU clusters, we extrapolate to estimate how many CPU cores we would need to achieve the same run times as on the GPU clusters.

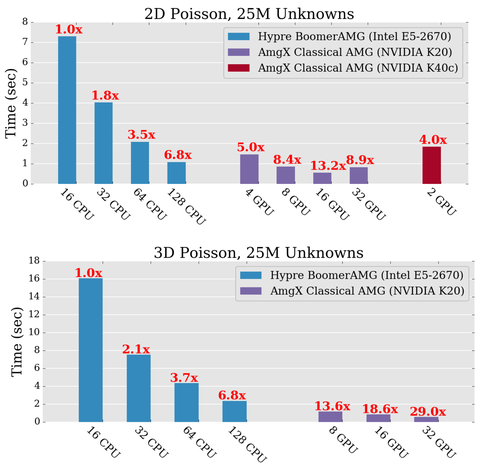

Fig. 3 For the 2D case, four K20s can compete with about 100 CPU cores. For the 3D case, eight K20s can compete with about 256 CPU cores (all tests with 25M unknowns).

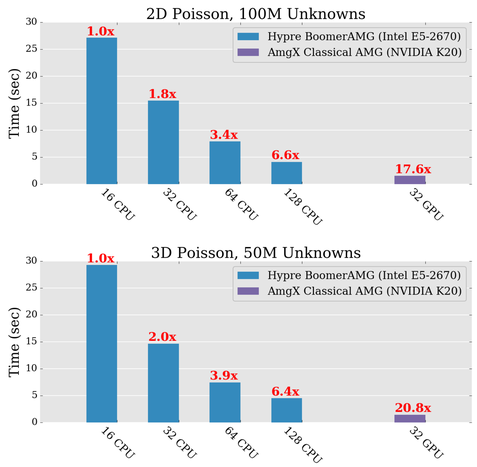

Fig. 4 Application speed-up for solving a Poisson system: 32 K20s can compete with about 400 CPU cores.

Benchmark: flying snakes–applications of PetIBM

With multi-GPU computing, we need smaller clusters or just a workstation for run times that used to require large CPU clusters.

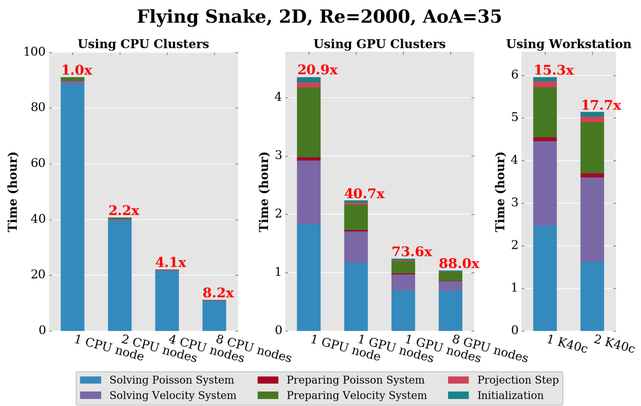

Fig. 5 Application speed-up on GPU: a 20-node CPU cluster would be needed to achieve the same 21x as 1 GPU node. Each node has 12 CPU cores/node; each GPU node has two NVIDIA K20s. The workstation has 6 CPU cores and up to two K40c GPUs. Also, a 6-CPU-core workstation with two K40c GPUs can compete with a 16-node CPU cluster.

Time is money !

Multi-GPU computing saves you time and also money, as shown by this test on cloud resources.

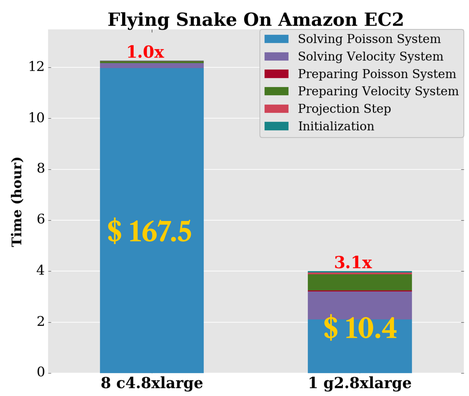

Fig. 6 Speed-up and cost savings on Amazon EC2, between 8 CPU nodes (c4.8xlarge, with 36 virtual CPU cores each) and 1 GPU node (g2.8xlarge, with 32 virtual CPU cores and 4 GPUs each). **The cost saving is 16x.**

Warning !

Speed-up here refers to **application speed-up**: replacing PETSc linear solvers with AmgX solvers in an application and experiencing faster run times.

PetIBM

A parallel Navier-Stokes solver using PETSc. Scan the QR code for the code repository.

Full report

Get a Jupyter notebook with a full report of our results.

Contact us

- http://lorenabarba.com/
- https://github.com/barbagroup/