The 30th International Conference on Parallel Computational Fluid Dynamics
May 14-17, 2018
Indianapolis, Indiana, USA

THREE-DIMENSIONAL FLOW SIMULATIONS OF THE FLYING SNAKE USING MICROSOFT AZURE

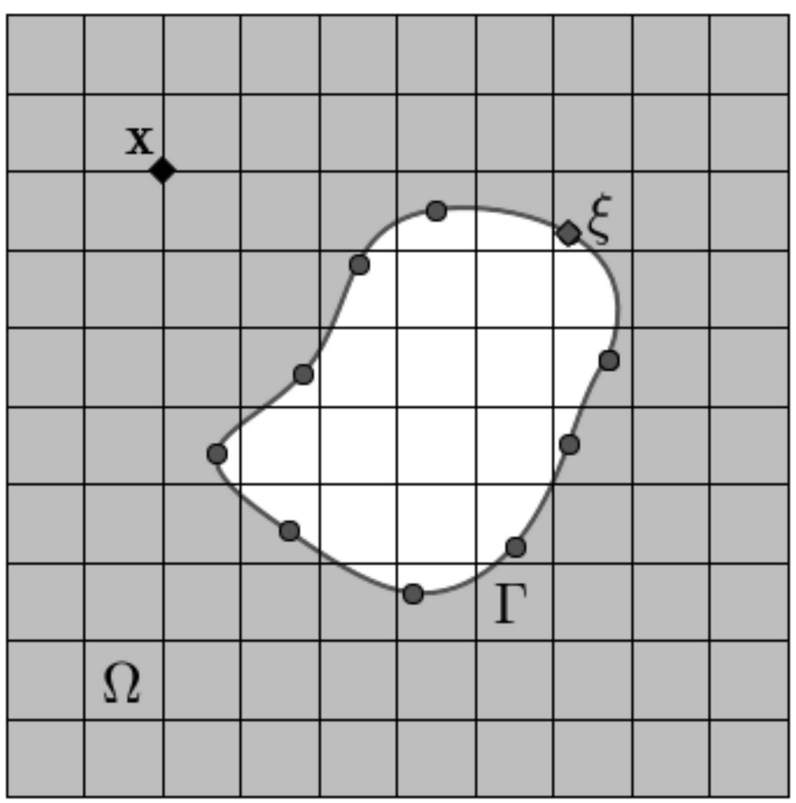
Olivier Mesnard & Lorena A. Barba

Mechanical and Aerospace Engineering Department
The George Washington University

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

PetIBM

- 2D & 3D incompressible Navier-Stokes equations (DNS)
- Projection method: block-LU decomposition (Perot, 1993)
- PETSc (Portable Extensible Toolkit for Scientific Computation)
- NVIDIA AmgX library (interfaced with AmgXWrapper)
- Immersed-boundary method



$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \int_{\mathbf{s}} \mathbf{f}\left(\xi\left(s, t\right)\right) \delta\left(\xi - \mathbf{x}\right) \mathbf{d}s \\ \nabla \cdot \mathbf{u} = \mathbf{0} \\ \mathbf{u}\left(\xi\left(s, t\right)\right) = \int_{\mathbf{x}} \mathbf{u}\left(\mathbf{x}\right) \delta\left(\mathbf{x} - \xi\right) \mathbf{d}\mathbf{x} \end{cases}$$

https://github.com/barbagroup/PetIBM

PetIBM: toolbox and applications of the immersed-boundary method on distributed-memory architectures (under review)

Immersed-boundary projection method

$$\begin{bmatrix} A & G & E^T \\ G^T & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \phi \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ u_B^{n+1} \end{pmatrix} + \begin{pmatrix} bc_1 \\ -bc_2 \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} \bar{A} & \bar{E}^T \\ \bar{E} & 0 \end{bmatrix} \begin{pmatrix} \gamma^{n+1} \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} \bar{r_1} \\ \bar{r_2} \end{pmatrix}$$

with $\bar{A} \equiv \begin{bmatrix} \bar{A} & \bar{E}^T \\ \bar{E} & 0 \end{bmatrix}; \; \gamma^{n+1} \equiv \begin{pmatrix} q^{n+1} \\ \phi \end{pmatrix}$

First block-LU decomposition:

$$\begin{bmatrix} \bar{A} & 0 \\ \bar{E} & -\bar{E}\bar{A}^{-1}\bar{E}^T \end{bmatrix} \begin{bmatrix} I & \bar{A}^{-1}\bar{E}^T \\ 0 & I \end{bmatrix} \begin{pmatrix} \gamma^{n+1} \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} \bar{r_1} \\ \bar{r_2} \end{pmatrix}$$

Second block-LU decomposition:

$$\begin{bmatrix} A & 0 \\ G^T & -G^T A^{-1} G \end{bmatrix} \begin{bmatrix} I & A^{-1}G \\ 0 & I \end{bmatrix} \begin{pmatrix} q^* \\ \phi \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

$$Aq^{**} = r_1$$

$$G^T A^{-1} G \lambda = G^T q^{**} + bc_2$$

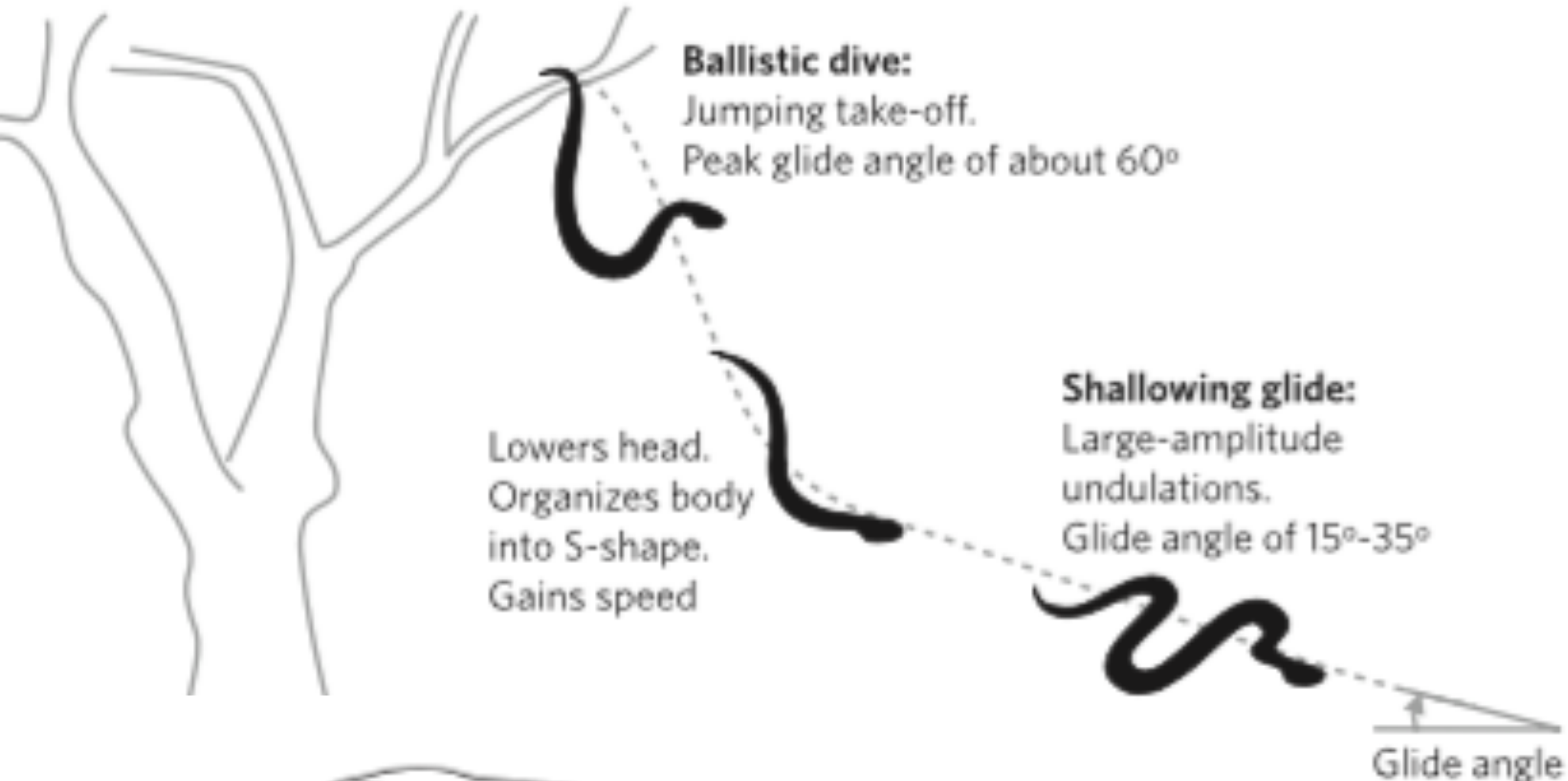
$$q^* = q^{**} - A^{-1} G \phi$$

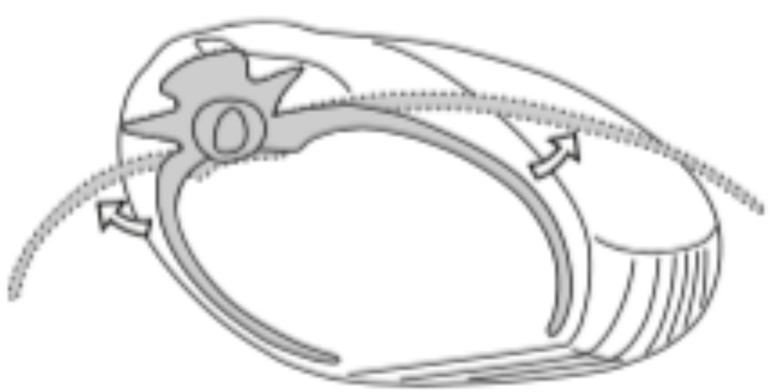
$$E A^{-1} E^T \tilde{f} = E q^* - u_B^{n+1}$$

$$q^{n+1} = q^* - A^{-1} E^T \tilde{f}$$

- Taira & Colonius (2007). The immersed boundary method: a projection approach. *Journal of Computational Physics*, *225*(2), 2118-2137.
- Li et al. (2016). An efficient immersed boundary projection method for flow over complex/moving boundaries. *Computers & Fluids*, *140*, 122-135.

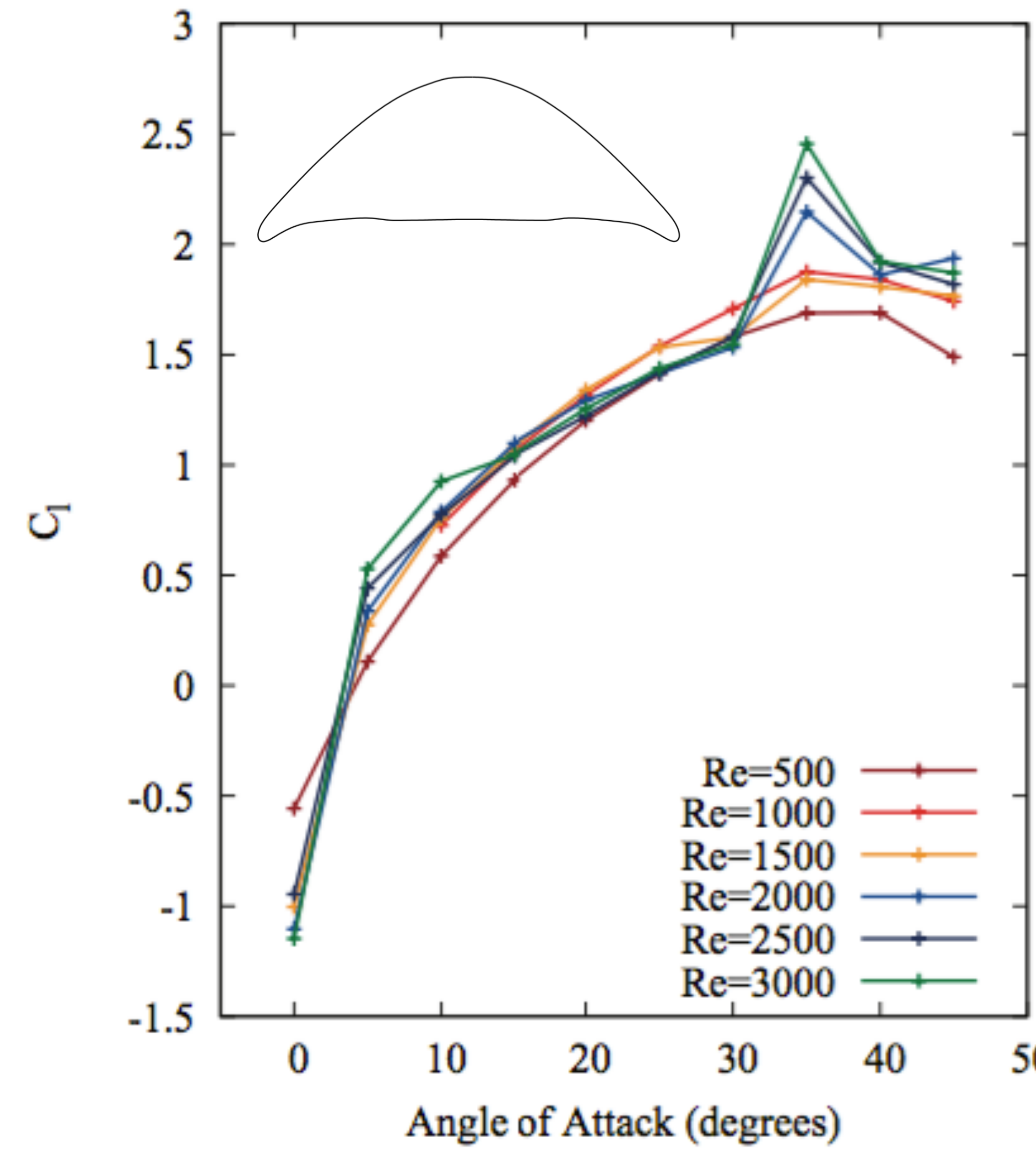
Flying snake





Normal configuration

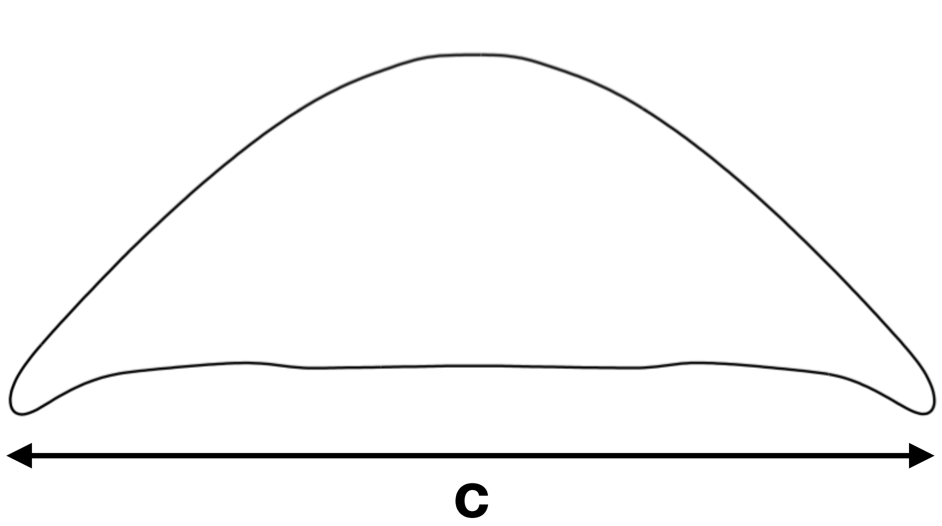
Configuration during glide

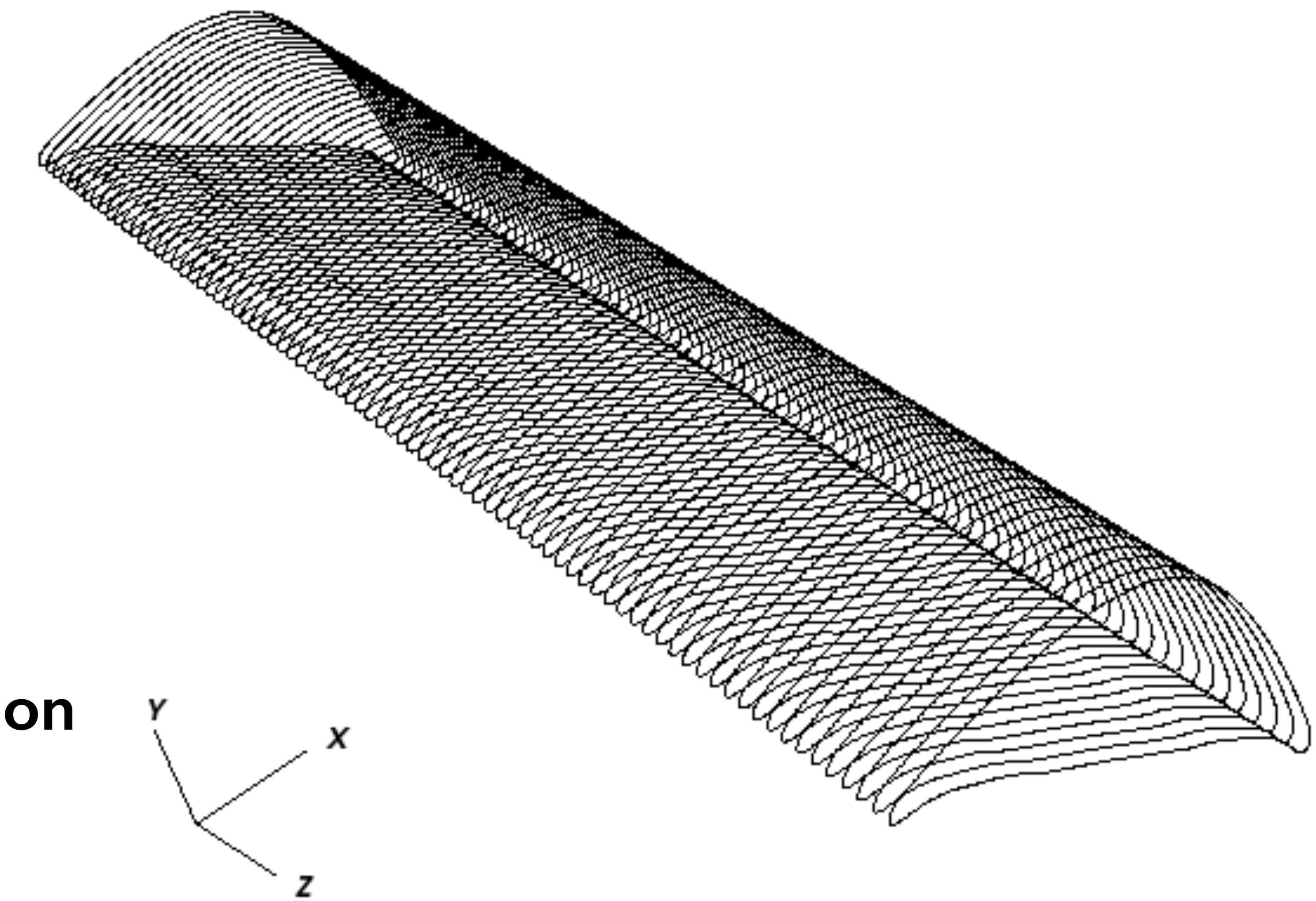


- Enhanced lift at Re ≥ 2000 at AoA 35 deg
- AoA matches previous experimental data
- Can we observe lift-enhancement with more realistic 3D simulations?

- Krishnan, A., Socha, J. J., Vlachos, P. P., & Barba, L. A. (2014). Lift and wakes of flying snakes. *Physics of Fluids*, 26(3), 031901.

3D Simulations of the Flying snake





- domain: 30c x 30c x 3.2c
- section extruded along z-direction
- cylinder centered in x/y plane

	number of cells	smallest cell width	uniform region	stretching ratio
2D	1704 x 1706 (2.9M)	0.004 x 0.004	[-0.52, 3.48] x [-2, 2]	1.01
3D fine	1704 x 1706 x 80 (233M)	0.004 x 0.004 x 0.04	[-0.52, 3.48] x [-2, 2] x [0, 3.2]	1.01
3D coarse	1071 x 1072 x 40 (46M)	0.008 x 0.008 x 0.08	[-0.52, 3.48] x [-2, 2] x [0, 3.2]	1.01

Simulation parameters

- Re = 2000
- Boundary conditions
  - Inlet/bottom/top: freestream velocity
  - Outlet: convective
  - Periodic in spanwise direction
- Linear solvers
  - Velocity system: (Jacobi) BCGS
    - PETSc KSP on CPUs
  - Poisson system: (classical AMG) CG
    - NVIDIA AmgX on GPUs (with AmgXWrapper)
  - Lagrangian forces system: (Jacobi) GMRES
    - PETSc KSP on CPUs

Moving to the Cloud

- University-managed HPC cluster busier and busier
- Cloud computing (infinite capacity and continuous availability)
- Microsoft Azure Sponsorship ($20k credit in resources)
- PetIBM application requirements:
  - GPU devices to solve the Poisson system
  - fast communication network (InfiniBand)
- Adopt a reproducible workflow to run jobs on the cloud

- Microsoft Azure for Research: https://www.microsoft.com/en-us/research/academic-program/microsoft-azure-for-research/

NC series on Azure

Instance	cores	RAM	disk sizes	GPU	pay-as-you-go	1 year reserved	3 year reserved
NC6	6	56GiB	340GiB	1 x K80	$0.90/hr	$0.574/hr	$0.40/hr
NC12	12	112GiB	680GiB	2 x K80	$1.80/hr	$1.147/hr	$0.80/hr
NC24	24	224GiB	1,440GiB	4 x K80	$3.60/hr	$2.294/hr	$1.599/hr
NC24r	24	224GiB	1,440GiB	4 x K80	$3.96/hr	$2.523/hr	$1.758/hr

- **NC24r: RDMA capable with InfiniBand network**
- **Now available:**
  - **NC series v2 (NVIDIA Tesla P100 GPUs)**
  - **NC series v3 (NVIDIA Tesla V100 GPUs)**

- Prices as of May 13, 2018 (https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/)

Workflow

- **NVIDIA Docker**:
  build PetIBM image locally and push to Docker Hub

```
> nvidia-docker build \
    --tag=mesnardo/petibm:0.2-GPU-IntelMPI \
    --file Dockerfile .
> docker push mesnardo/petibm:0.2-GPU-IntelMPI-ubuntu
```

- **Azure CLI 2.0**:
  create Azure Batch account and upload files to Azure Storage

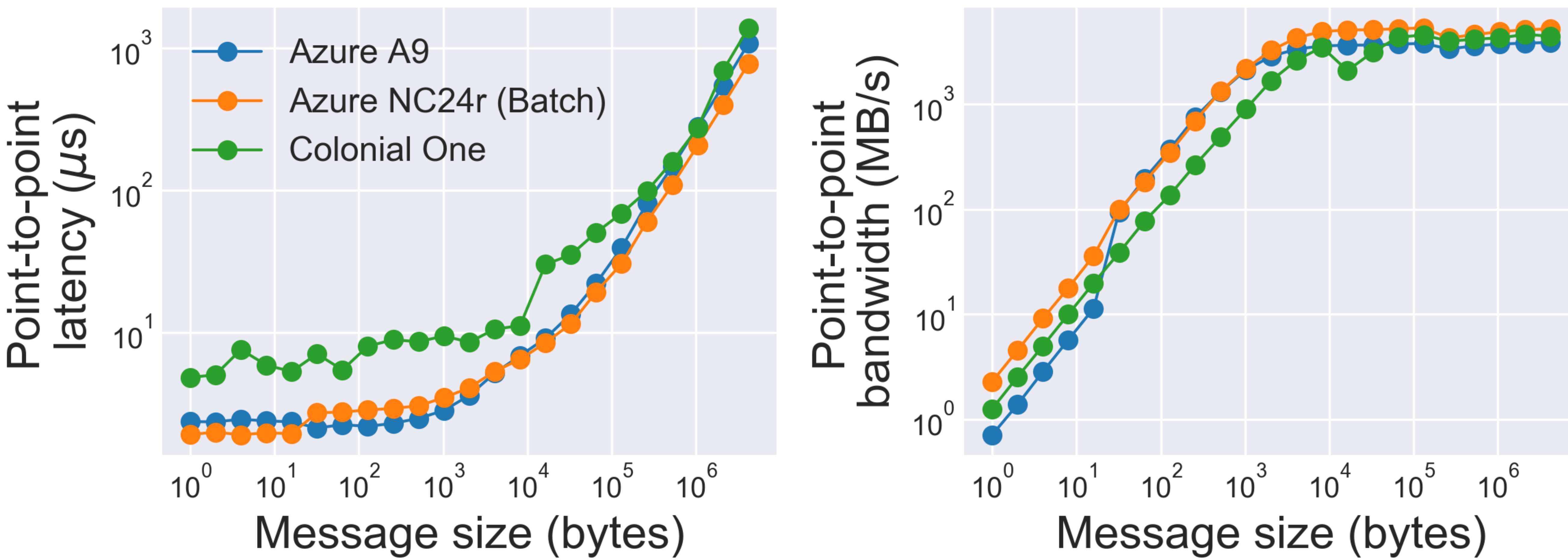
```
> az storage directory create \
    --name snake --share-name myfileshare \
    --account-name mesnardo2storage
> az storage file upload-batch \
    --source snake_inputs --destination myfileshare/snake \
    --account-name mesnardo2storage
```

- **Batch Shipyard**:
  provision compute pools and submit multi-tasking jobs to Azure Batch service

```
> shipyard pool add --configdir config_dir
> shipyard jobs add --configdir config_dir
```

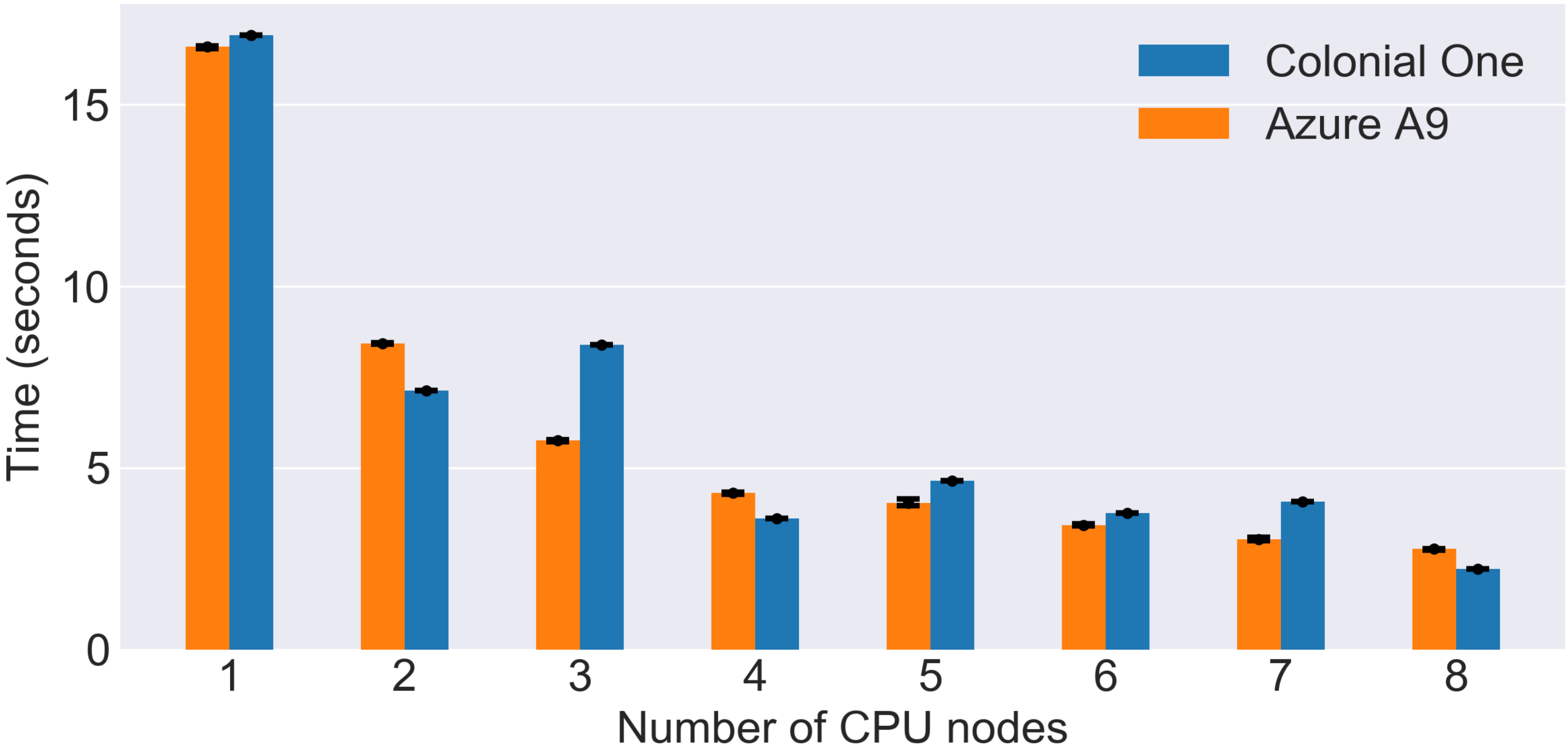
OSU Point-to-Point Benchmark

- CPU node on Colonial One
  - Dual 8-Core 2.6GHz Intel Xeon E5-2670 CPUs
  - InfiniBand
- Microsoft Azure A9 and NC24r
  - Dual 8-Core 2.6GHz Intel Xeon E5-2670 CPUs
  - InfiniBand



Poisson Benchmark

- Poisson system with 46M unknowns
- Hypre BoomerAMG classical preconditioner
- PETSc CG
- 16 CPU cores per node

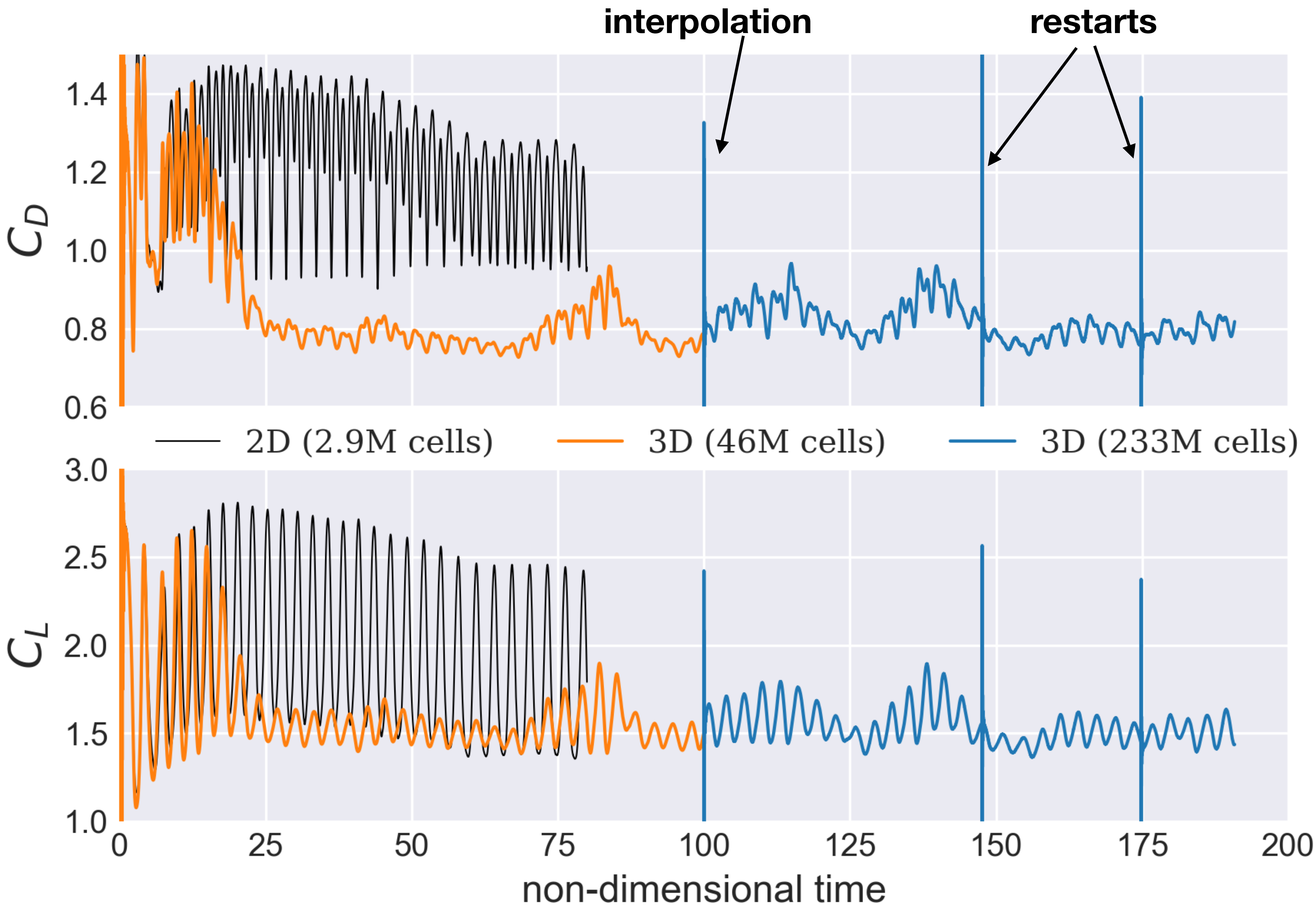


Flying Snake to the Cloud

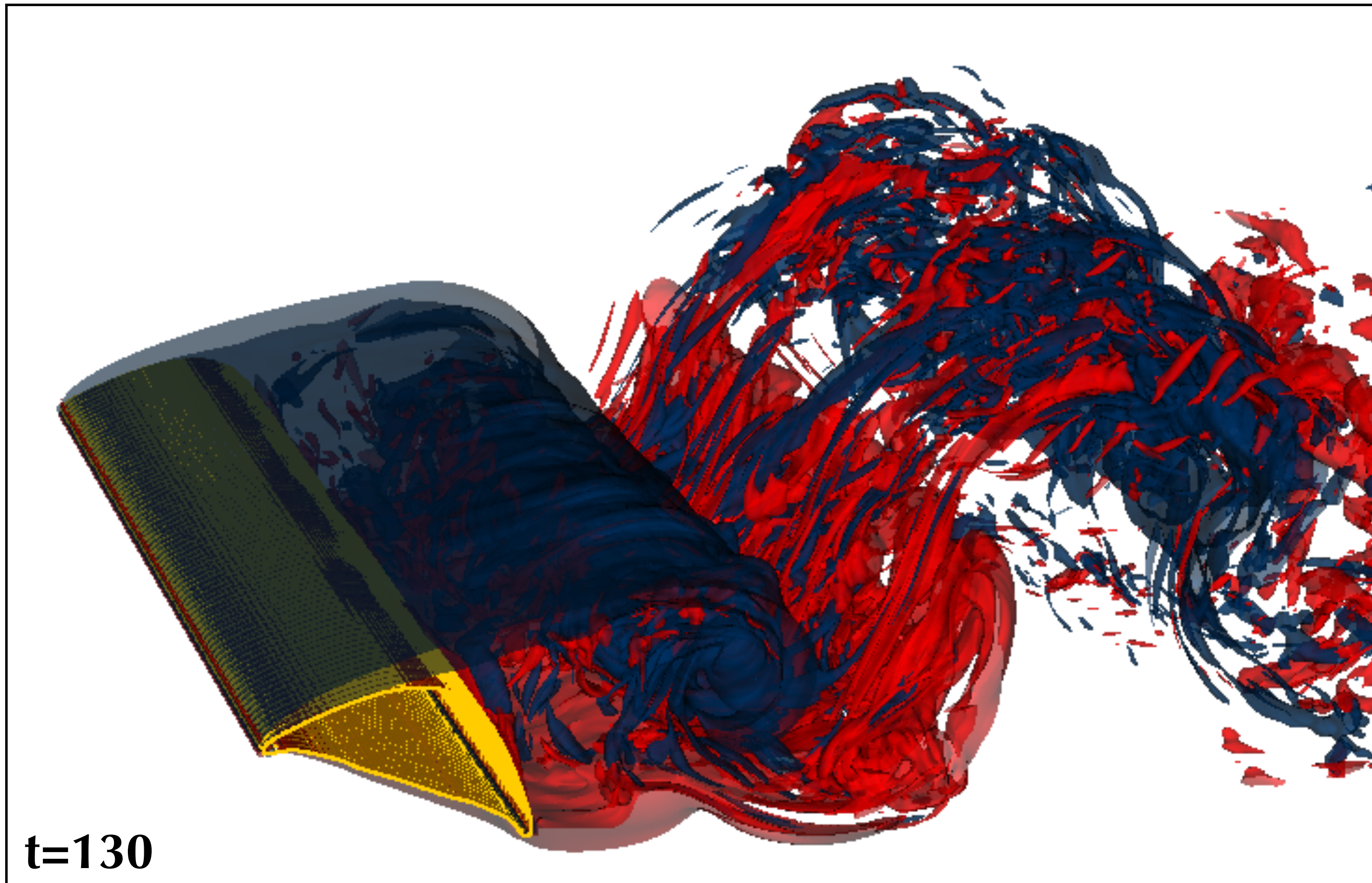
- Re=2,000 -- AoA=35°
- Coarse grid (46M cells)
  - 1 NC24 node (24 CPU cores, 4 K80 GPUs)
  - 148 hours for 100k time steps
- Fine grid (233M cells)
  - 6 NC24r nodes (24 CPU cores, 4 K80 GPUs per node)
  - container-based multi-tasks jobs (Batch Shipyard)
  - 387 hours for 186k time steps

History of Force Coefficients

Re=2,000 -- AoA=35°

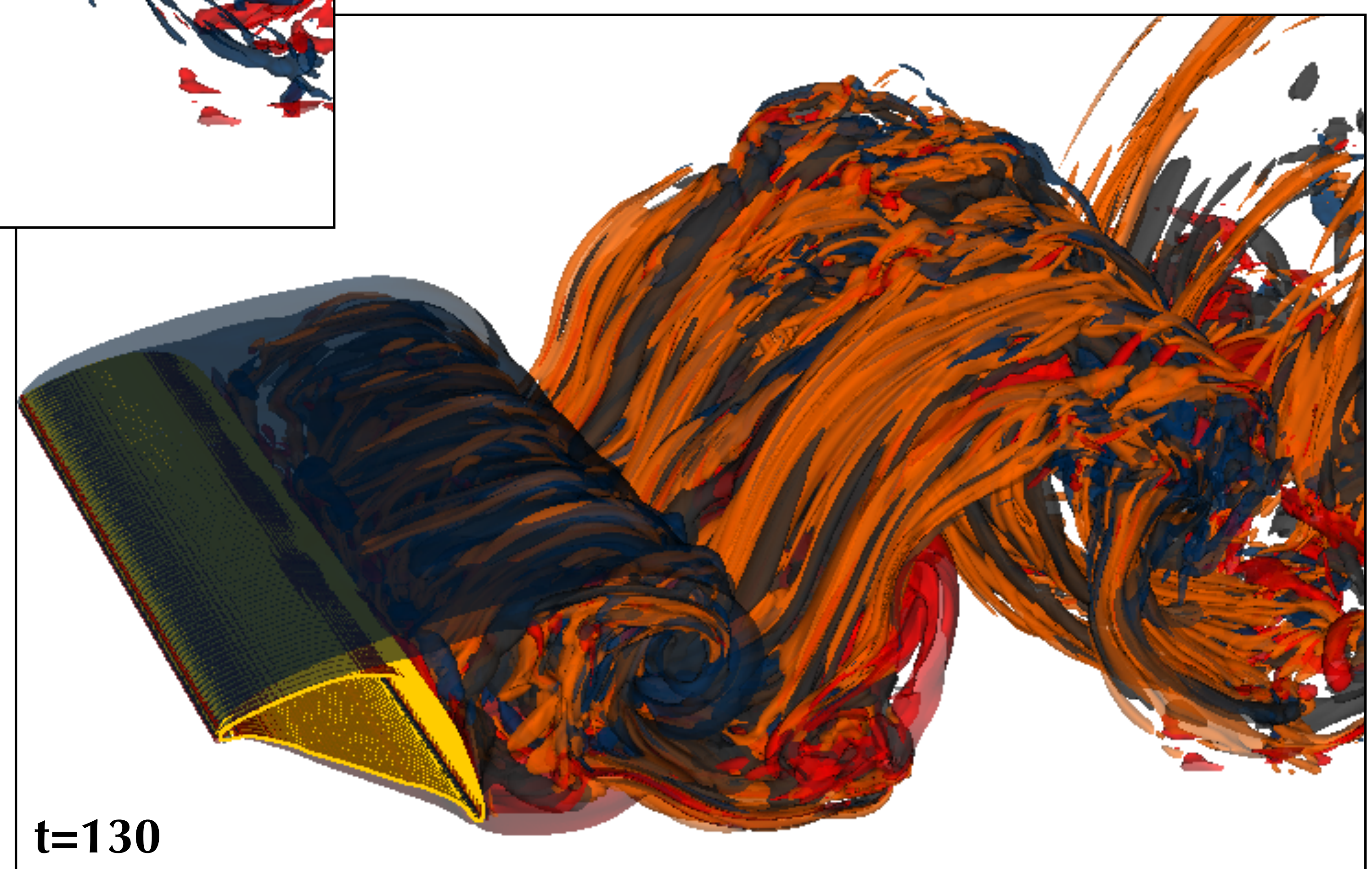


Vorticity (Re=2,000 -- AoA=35°)



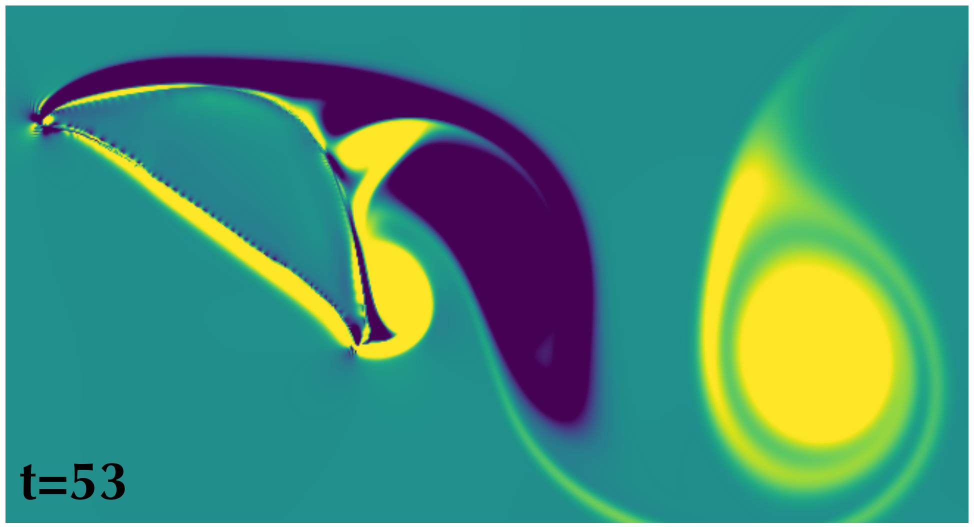
◀ Contours of instantaneous spanwise vorticity (red: -5, blue: +5)

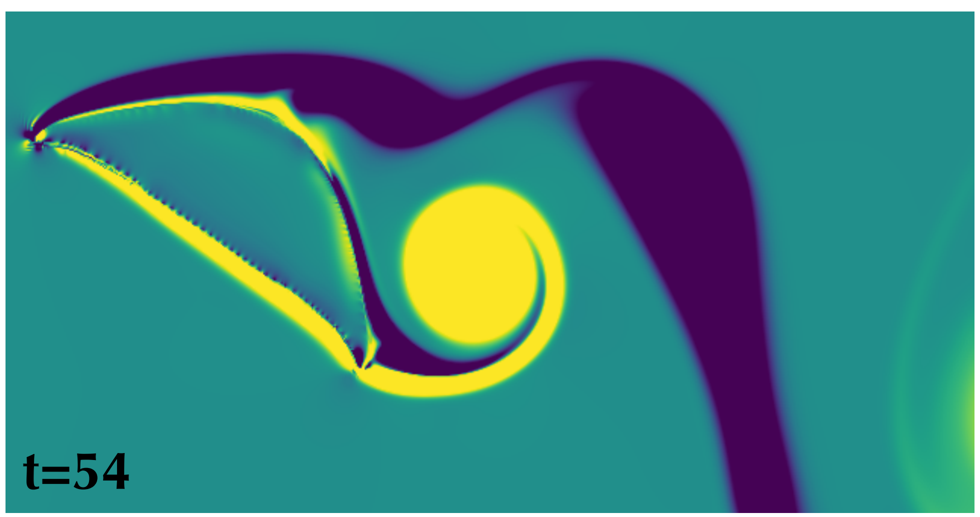
Contours of instantaneous ▶ spanwise (red: -5, blue: +5) and streamwise (grey: -20, orange: +20) vorticity

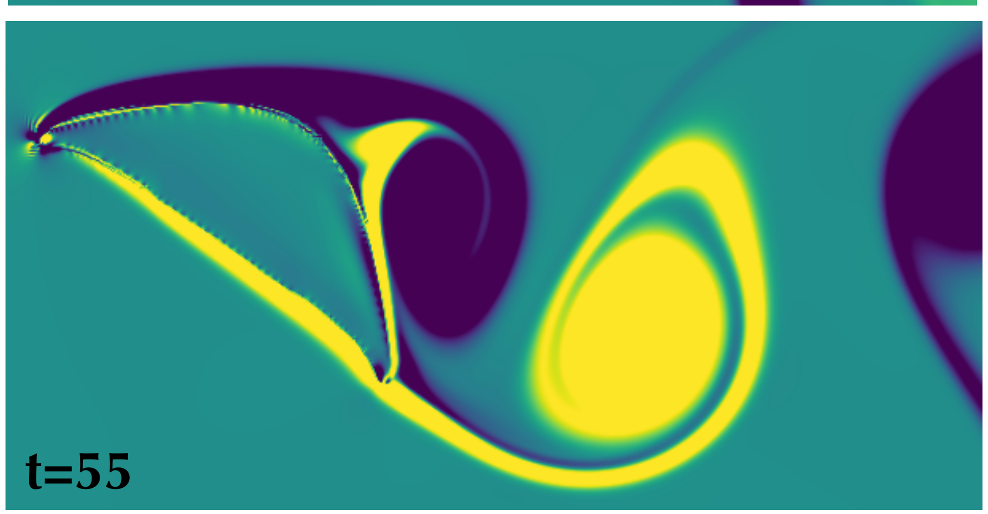


Vorticity (Re=2,000 -- AoA=35°)

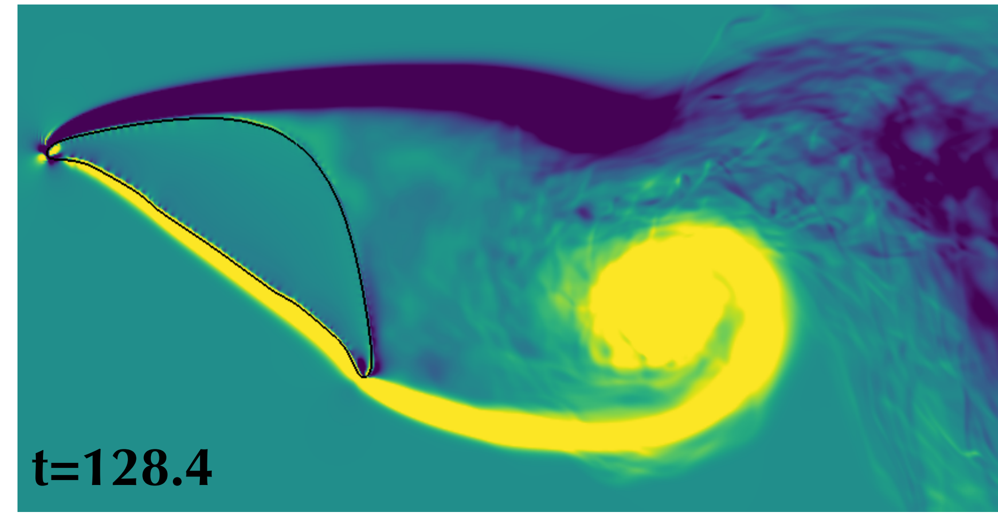
2D (2.9M cells)

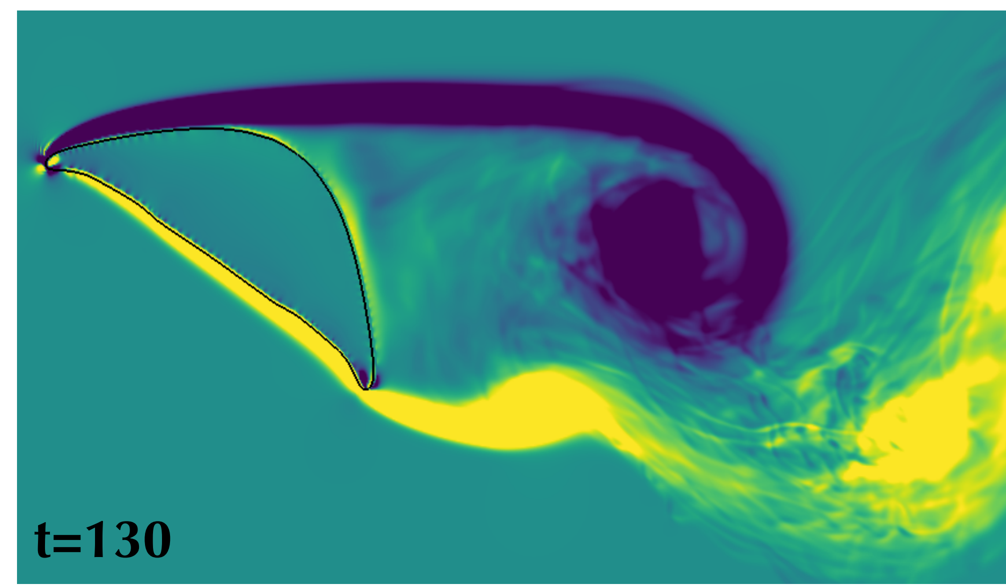


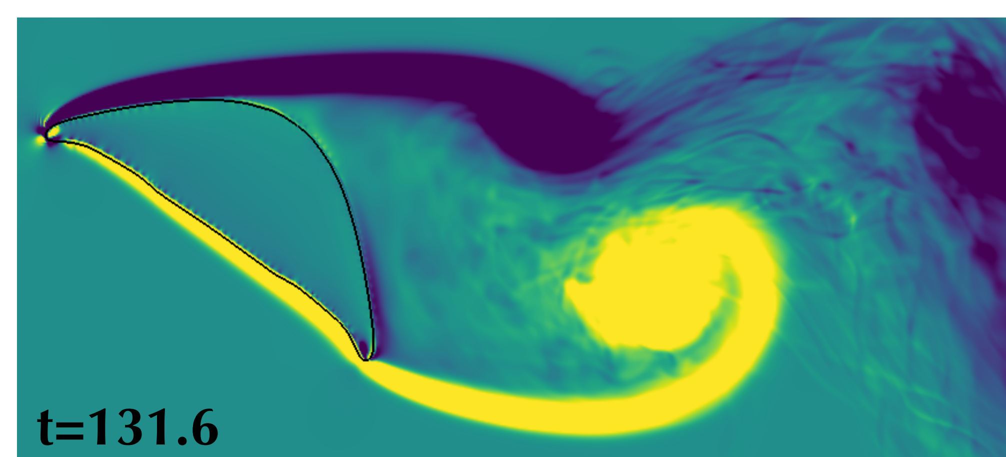




3D (233M cells)



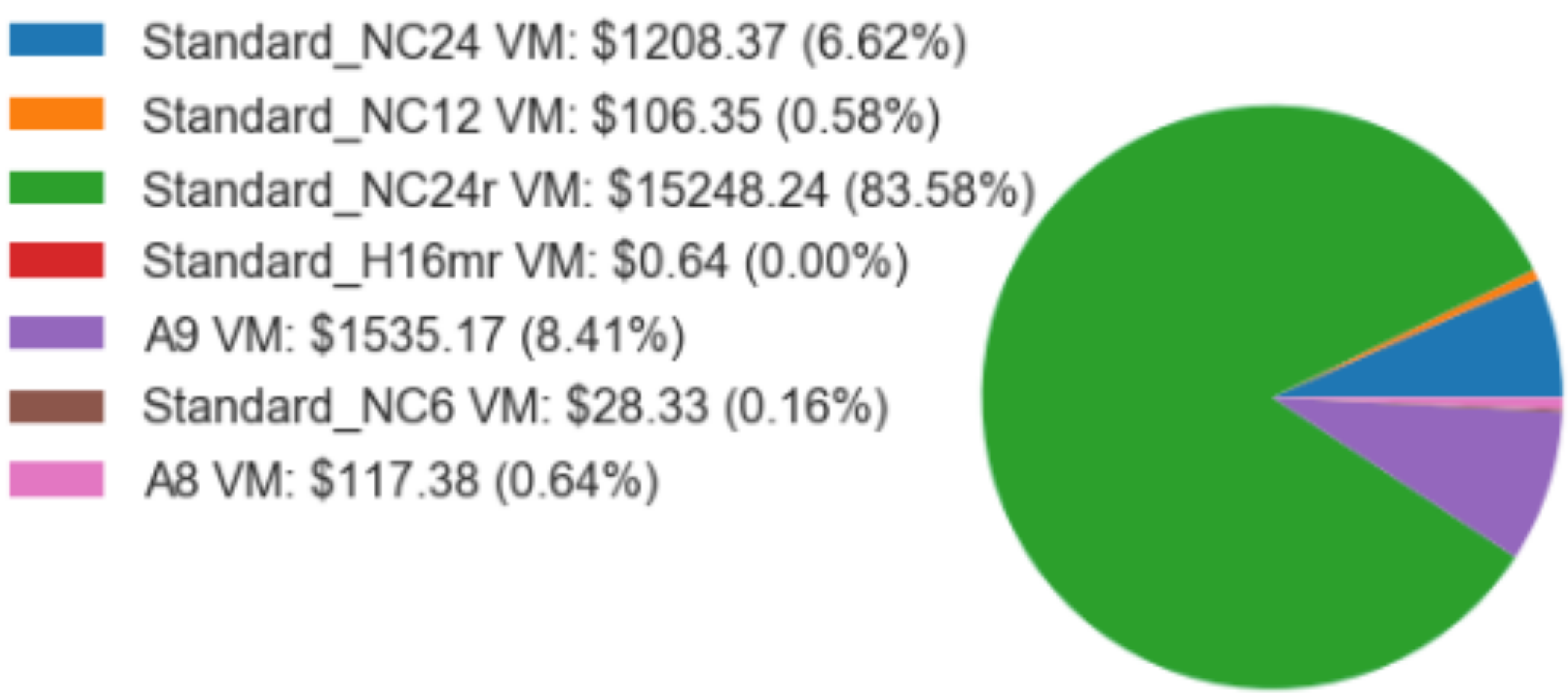




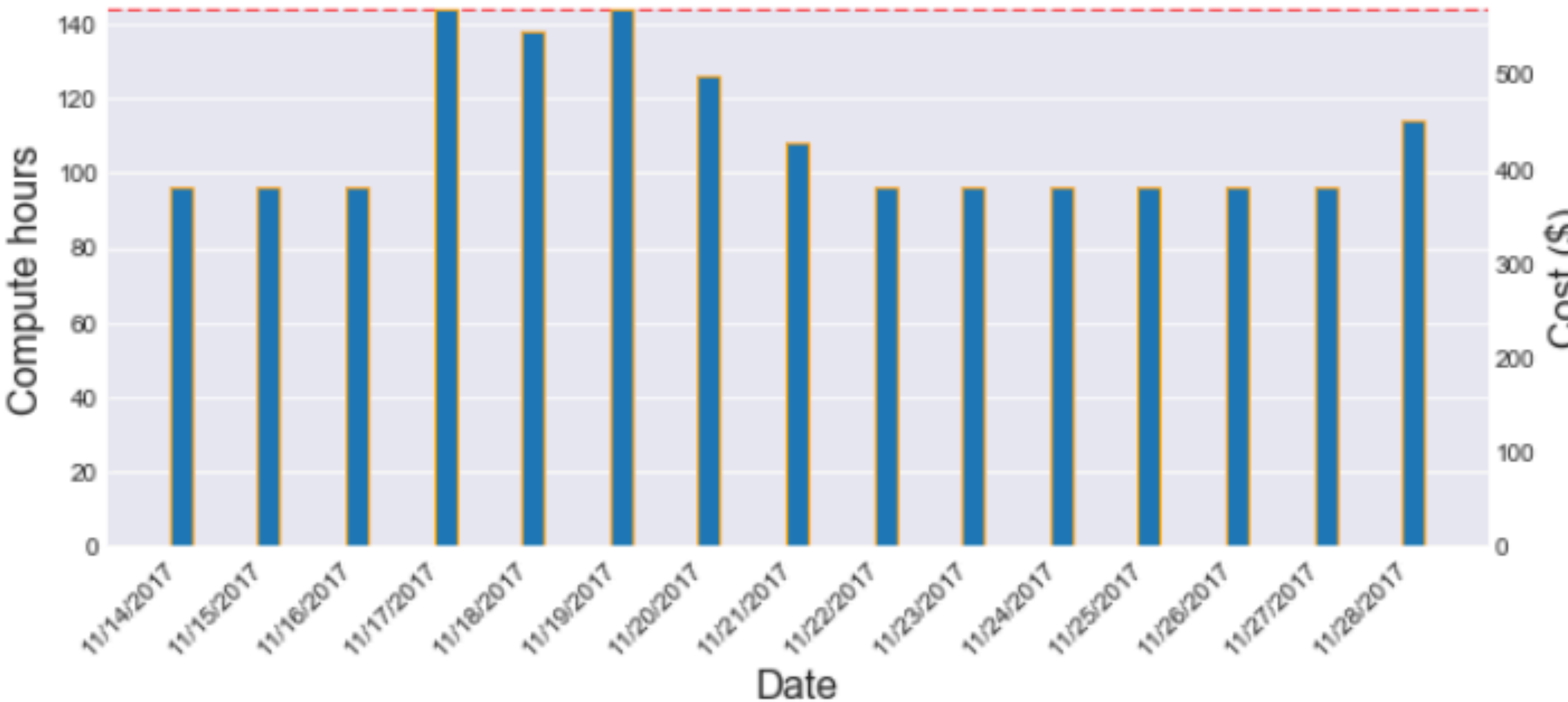
Charges incurred on the cloud

- Microsoft Azure Sponsorship: $20k
- Total spent: $18,436.6 (over 10 months)

Service	Cost	%
Data Management	$11.7	0.06
Networking	$57.6	0.31
Storage	$122.8	0.67
Virtual Machines	$18,244.5	98.96



Bank Error in your Favor



"Earlier this month Engineering did identify a bug related to our Sponsorship billing system and broader Azure usage system, in which customers were being underbilled for services leveraged. This has been addressed, but the past usage is not retroactively billed against affected accounts."
-- Customer Support Service (March 28, 2018)

Cost of running PetIBM on the Cloud

run	grid size	time steps	pool	cost ($)†	under-charge ($)*	estimated cost/cell/time-step ($\times 10^{-10}$)
2k35 (coarse)	46M	100,000	1 NC24	586.3*	-	1.27
2k35 (fine)	233M	98,614	6 NC24r	6,486.5	1,368.2	3.42
2k35 (fine, 1st restart)	233M	55,200	6 NC24r	2,496.2	927.9	2.66
2k35 (fine, 2nd restart)	233M	32,000	6 NC24r	1,576.6	346.6	2.58
2k30 & 1k35 (coarse)	46M	2x 100,000	2x 2 NC24r	1,719.8	488.8	2.40
Total				12865.4	3131.5	

† Cost as reported on our Azure billing statement
* Cost based on the runtime of the run

- Unexpectedly saved 20% due to bug in billing system
- Running PetIBM for 10k time steps on a grid with 1 million points:
  - using NC24r instances: ~$2.6
  - using a single NC24 instance: $1.27

TOOLBOX

- **PetIBM:** https://github.com/barbagroup/PetIBM
- **NVIDIA AmgX:** https://github.com/NVIDIA/AMGX
- **AmgXWrapper:** https://github.com/barbagroup/AmgXWrapper
- **Microsoft Azure for Research:** https://www.microsoft.com/en-us/research/academic-program/microsoft-azure-for-research/
- **NVIDIA Docker:** https://github.com/NVIDIA/nvidia-docker
- **Azure CLI 2.0:** https://github.com/Azure/azure-cli
- **Batch Shipyard:** https://github.com/Azure/batch-shipyard
- **FlyingSnake2Cloud:** https://github.com/mesnardo/FlyingSnake2Cloud