Final presentation - 7 Dec (Thu)

GPU Accelerated ~~Pressure Solver~~ ~~for~~ TIGER-F code

臺科大
TAIWAN TECH

Team Members

- Fandi D. Suprianto
- Desta G. Tewolde
- 范瀞元
- 周岱佑
- 關名芳

Advisor: Prof. Ming-Jyh Chern

(NTUST CFD Lab)

Mentors: Bharat, Shijie, Kuan-Ting

OPEN HACKATHONS
OpenACC

TIGER-F

- Problem trying to be solved

  3D-CFD using the Direct Forcing Immersed Boundary (DFIB) method, LES turbulence model.

- Scientific driver for the chosen algorithm

  The flexibility to solve large scale FSI problems.

- What’s the algorithmic motif?

  The use of simple and fixed Cartesian grids eliminates the need for remeshing, even as the structure is in motion.

- What parts are you focusing on?

  Initially: Computation intensive parts (pressure solver).
  But in the end, we need to offload all processes in the main iteration.

Evolution and Strategy

- What was your goal coming here?

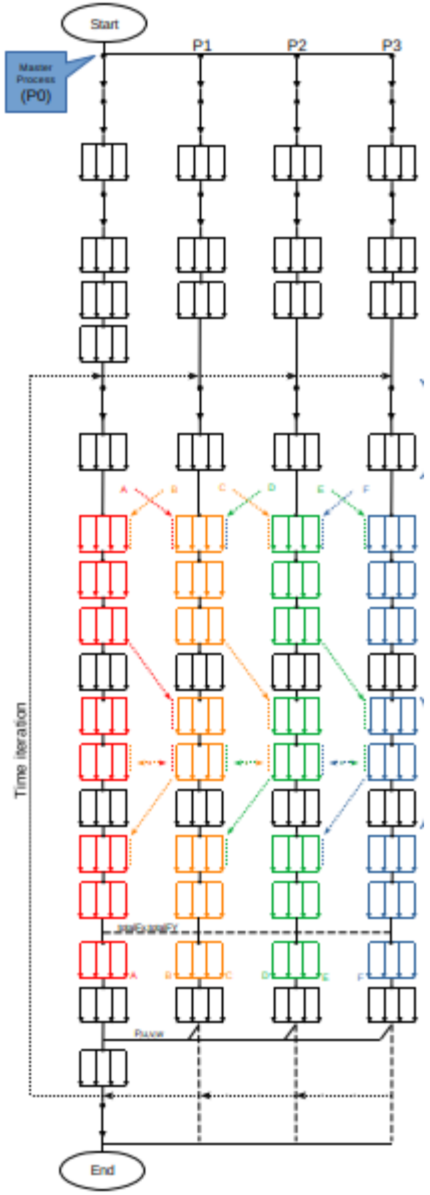
  to **speedup** our in-house solver using openACC by offloading and optimizing the computation intensive processes (e.g. pressure solver) into GPU

- What was your initial strategy?
  - To optimize the GPU-accelerated pressure solver and to use the Nsight systems for profiling.
  - to offload more subroutines into GPU.

- How did this strategy change?

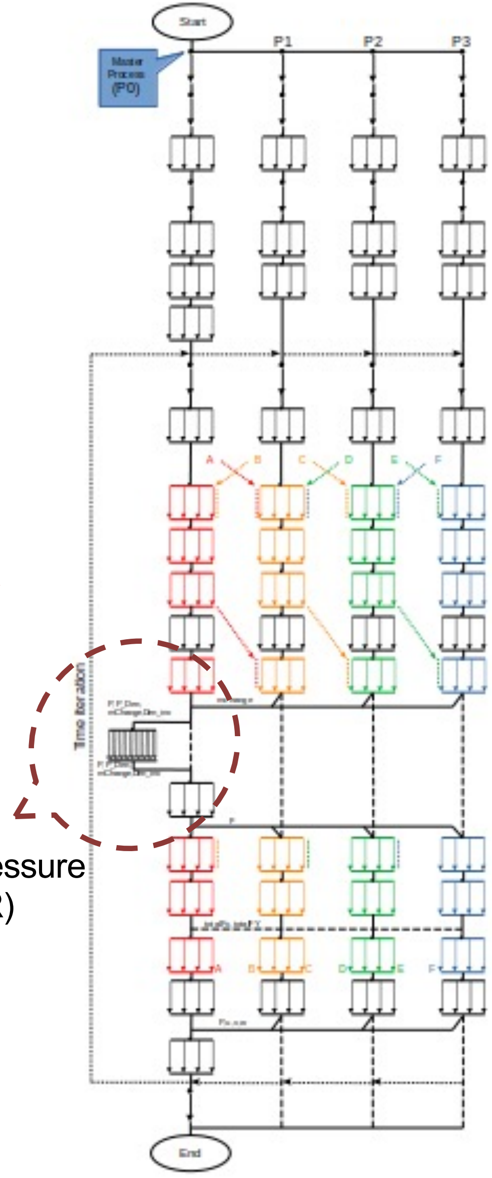
  The majority of computational time is consumed by data transfer overhead, which could be diminished by relocating **all** processes into GPU.

open MPI + openMP



Offload Pressure solver

open MPI + openMP + openACC

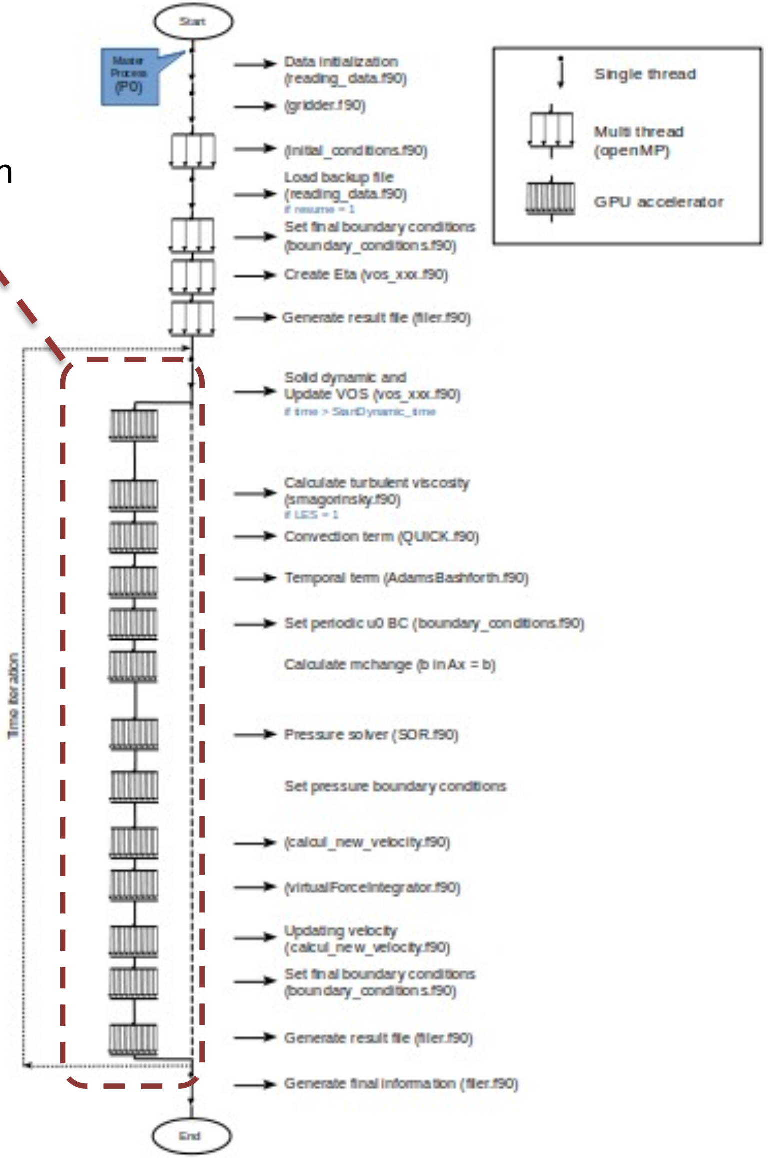


GPU for Pressure solver (SOR)

Offload more into GPU

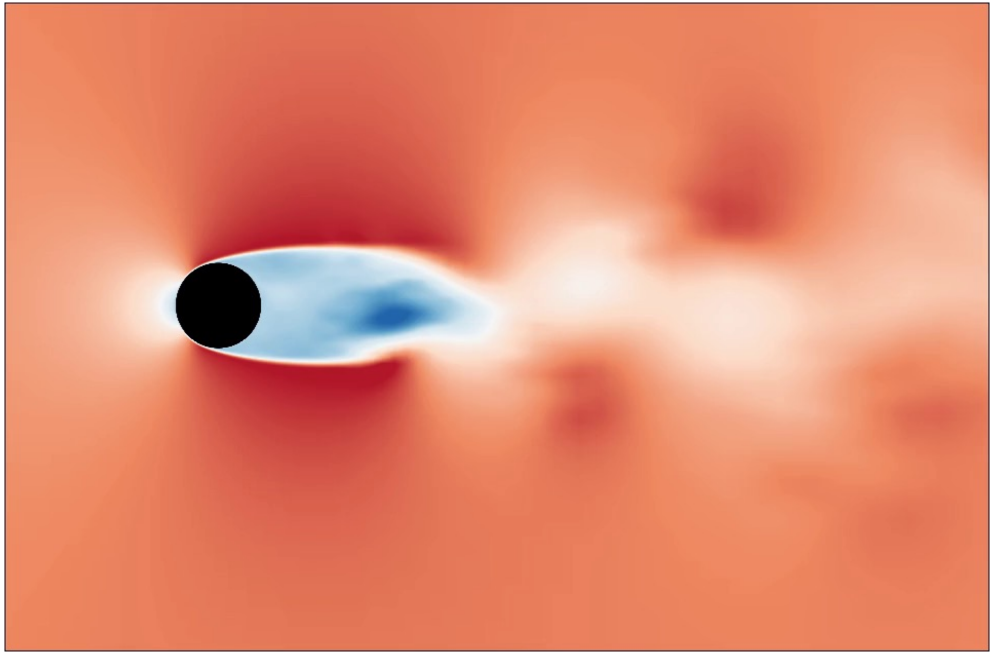
GPU for main iteration

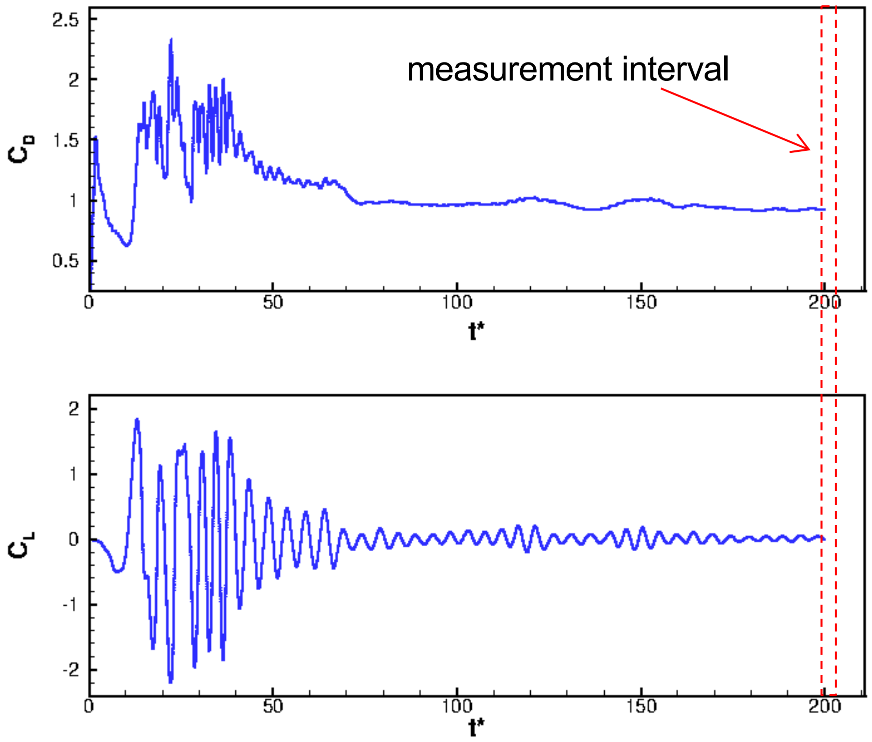
openMP + openACC



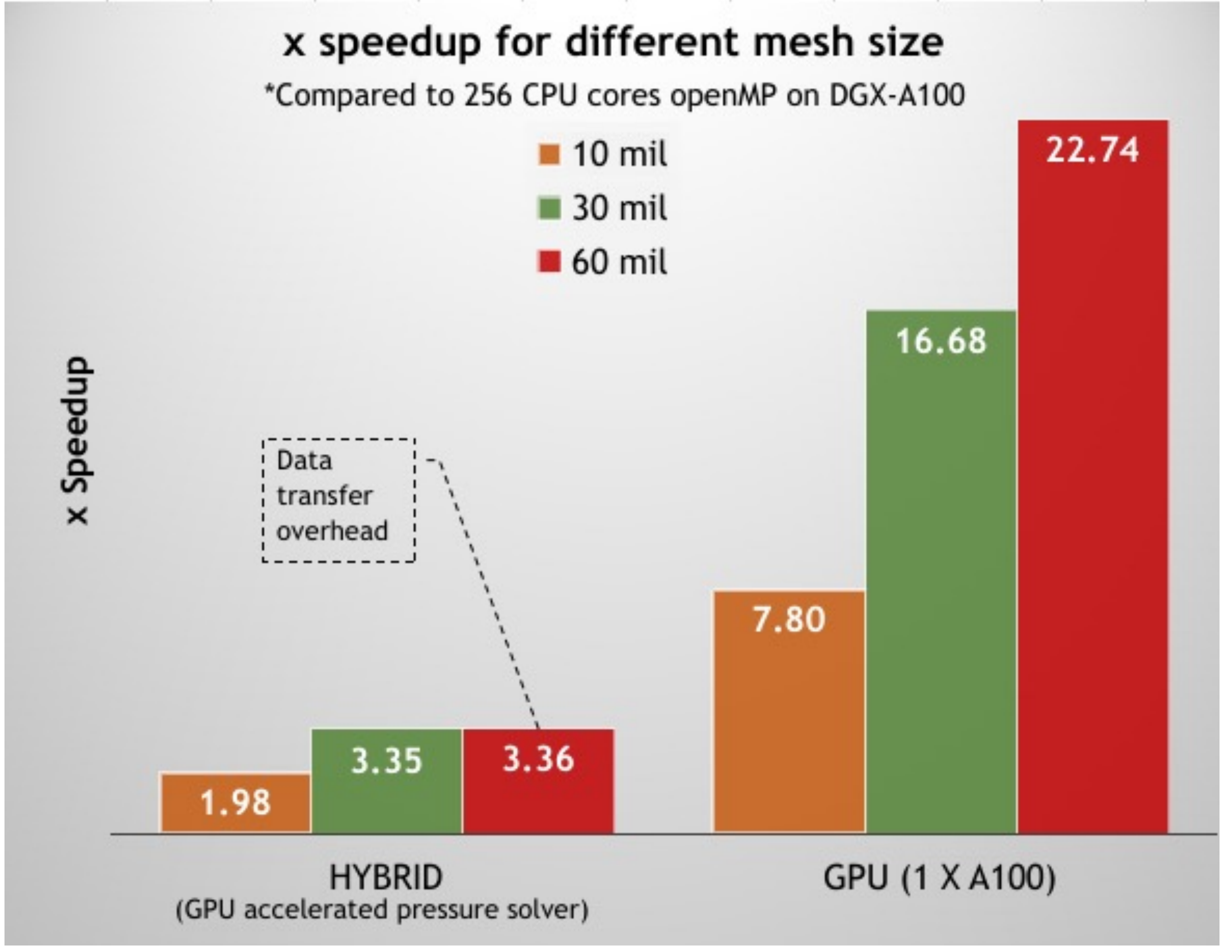
Test Case

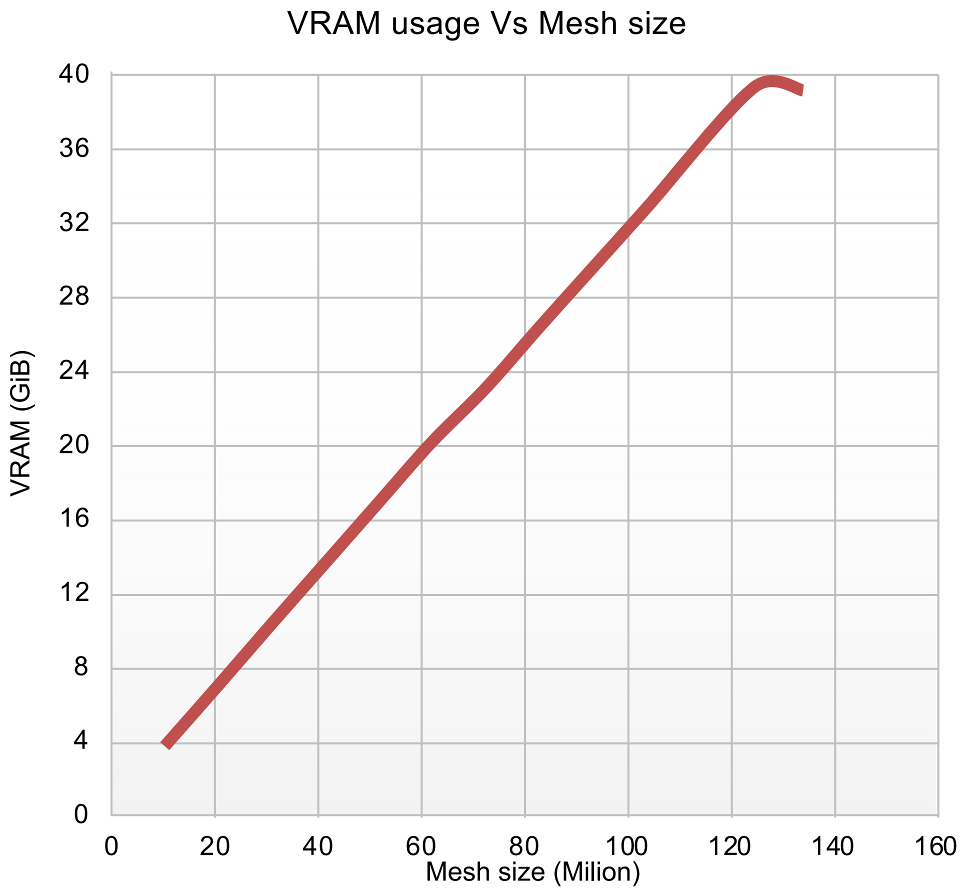
The test case was the flow past a stationary cylinder at Re = 5000, running for 200 sec of dimensionless time to give enough time to reach a **steady number** of pressure iterations. Speedup was measured for the next 1 second.



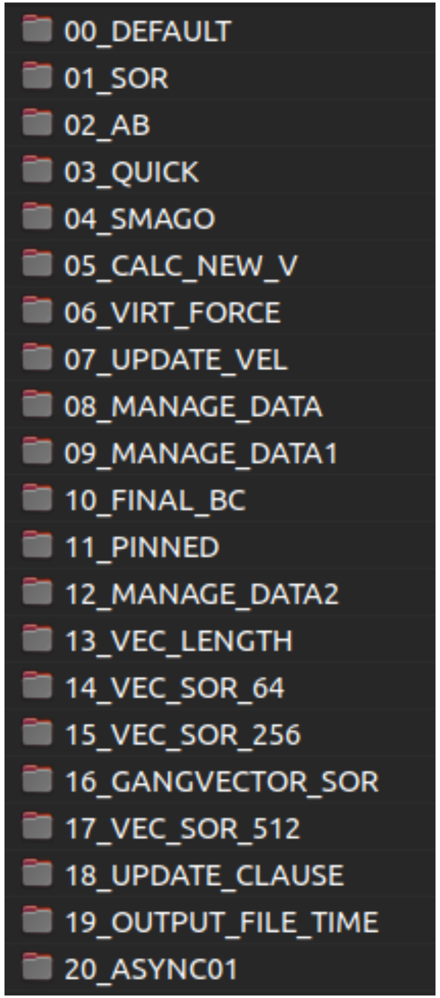


Results and Final Profile





Results and Final Profile



What did you learn?

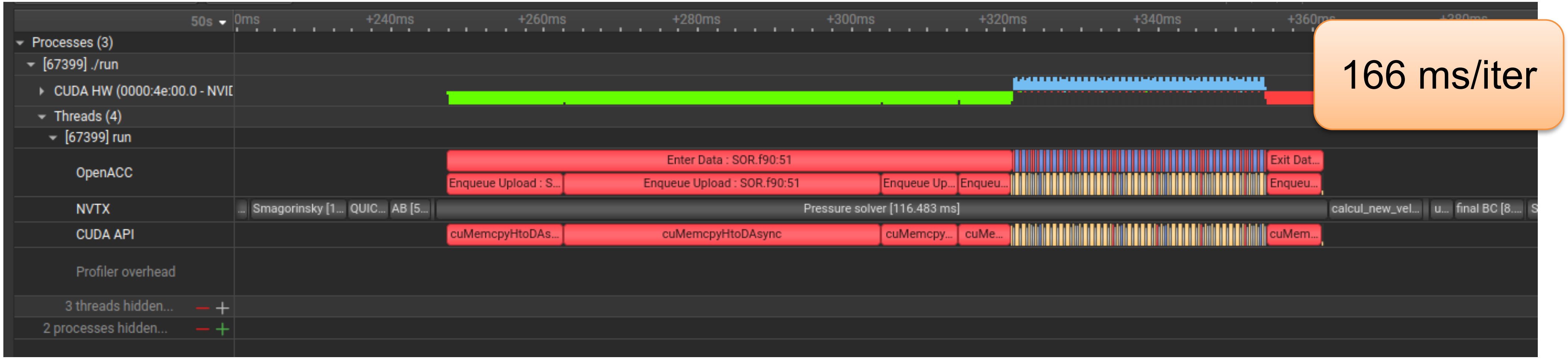
- Data transfer management.
- OpenACC’s three levels of parallelism (gang, worker, vector).
- Profiling tools: Nsight systems, NVTX.

Achieved new scientific goals?

- Apply the red-black SOR algorithm for 3D domain, and run in GPU.
- 100 Million of mesh is possible to be done (very) much faster --> one step closer to simulate real world cases.

Results and Final Profile

HYBRID 10mil



GPU 10mil

Energy Efficiency

INPUTS	
# CPU Cores	256
# GPUs (A100)	1
Application Speedup	16.7x

Node Replacement	266.9x

GPU NODE POWER SAVINGS			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (W)	293,568	6,500	287,068
Networking Power (W)	12,393	93	12,300
Total Power (W)	**305,961**	**6,593**	299,368

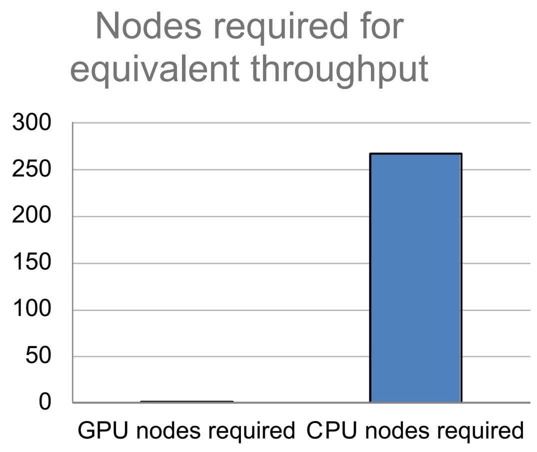
Node Power efficiency	**46.4x**

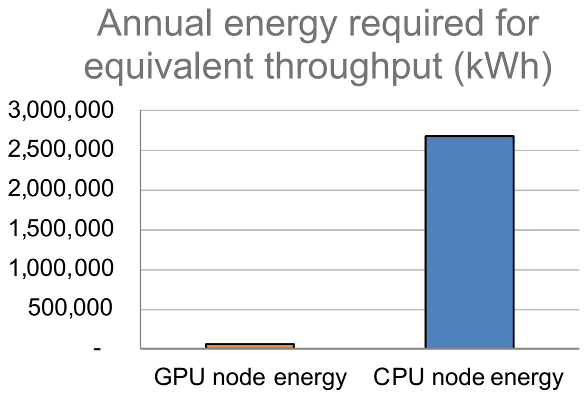
ANNUAL ENERGY SAVINGS PER GPU NODE			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (kWh/year)	2,571,656	56,940	2,514,716
Networking Power (kWh/year)	108,563	814	107,749
Total Power (kWh/year)	**2,680,219**	**57,754**	**2,622,465**

$/kWh	$ 0.34
Annual Cost Savings	$ 891,638.13
3-year Cost Savings	$ 2,674,914.39

Metric Tons of CO_2	1,859
Gasoline Cars Driven for 1 year	401
Seedlings Trees grown for 10 years	30,735

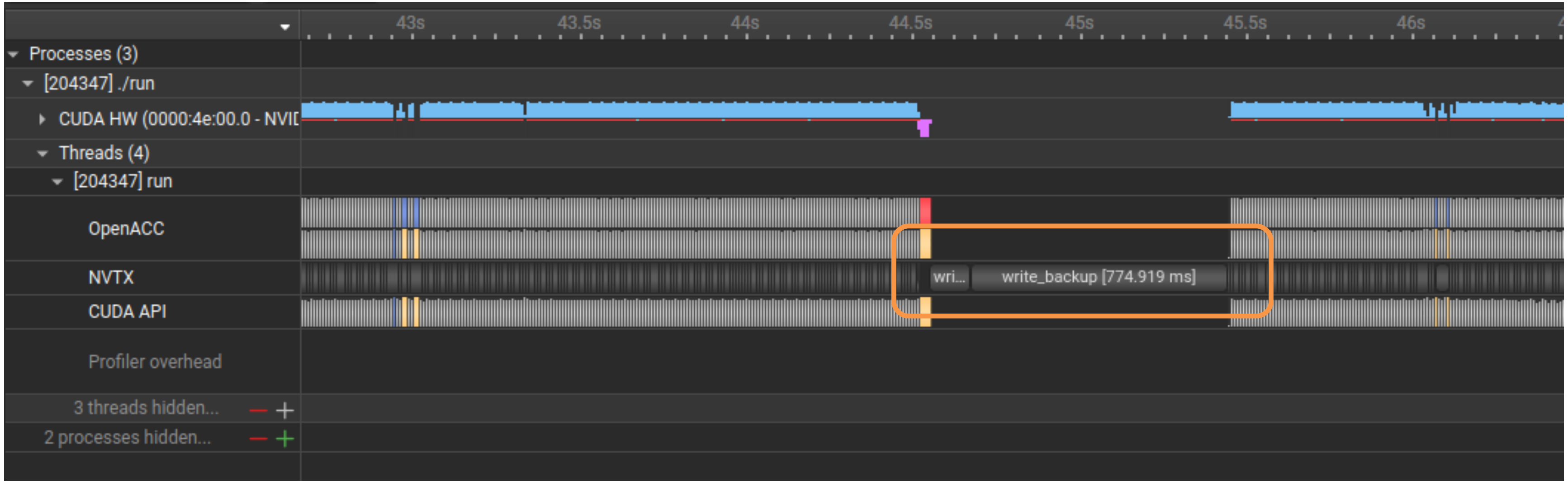
(source: Link)





What problems have you encountered?

- Unsuccessful Async between CPU file writing and GPU processes
- Unable to use “cache” clause
- Nsight compute could not be used because CAP_SYS_ADMIN capability was disabled.



Wishlist

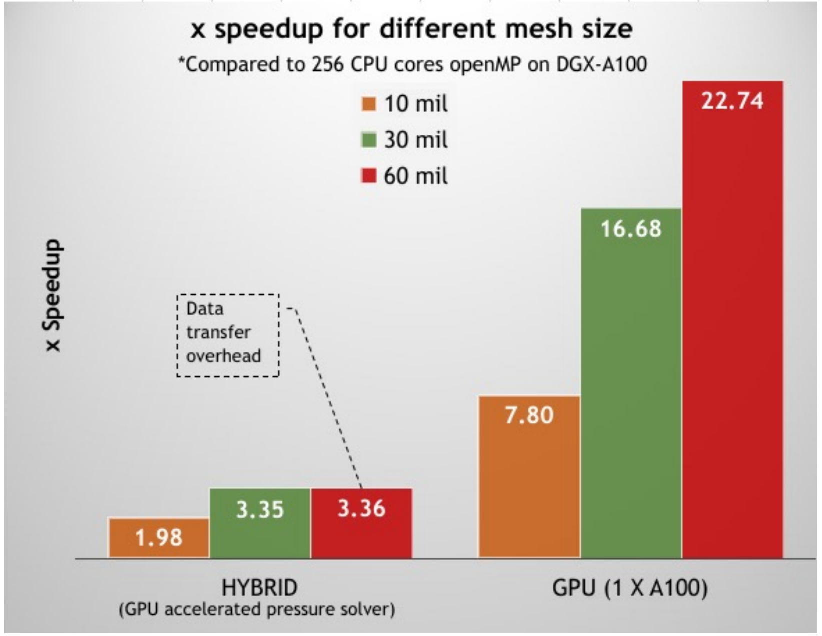
- What do you wish existed to make your life easier?
  - Multi GPU resource availability.
  - Updated training especialy on multi GPU utilization.
  - An integrated cluster with a balanced CPU+GPU performance.

Was it worth it?

- Was this worth it? Definitely YES!
- Will you continue development? YES
  - problems encountered on this event (async file writing, cache)
  - Use Nsight compute to optimize kernels
  - Multi-GPU process
  - Employing a readily available linear system solver with multi-GPU support. (AmgX)
- What sustained resources/support will be critical for your work after the event?
  - Multi-GPU resource and support.

Application Background

- A 3D-CFD in-house code using the DFIB method, LES turbulence model.
- Pressure solver is the most computational intensive subroutine.



Hackathon Objectives and Approach

- Offloading into GPU
- Reducing data transfer overhead
- Profiling / hot spots
- Performance tuning

Technical Accomplishments and Impact

- Learn more openACC directives & clauses
- Learn how to use Nsight systems
- Offload more processes into GPU
- Reduce data transfer overhead
- Speedup

Please use 100 words to summarize your team's achievements during this Hackathon

We began considering GPU offloading around 7 months ago and successfully accelerated our pressure solver onto the GPU. This hackathon not only boosted our learning curve but also enhanced the performance of our code. From this event, we are able to:

- Utilize openACC directives and their clauses
- Manage host-device data transfer better
- Optimize parallelism using openACC's three levels of parallelism (gang, worker, vector)
- Utilize Nsight systems with NVTX to profile the code
- Offload all processes in the main iteration to the GPU.
- **Achieve a speedup of 16.7x** (and even higher with increased computational load)

This marks another milestone for our in-house code as large-scale computations are required to simulate real-world cases. The significant speedup achieved is a substantial leap forward for us in reaching that goal.

Thank you!

Our mentors:

Bharat, Shijie, Kuan-Ting

Host	Jay	**Floating Mentor**	Bharat, Aswin
Host	CK	**Marketing**	Jinny
Infra Setup	Kuan-Ting	**Event logistics**	Apoorva
Account Manager	Vincent	**NCHC Contact Window**	Zhoujin Wu

All mentors and all participants in this hackathon

NARLabs 國家實驗研究院
國家高速網路與計算中心
National Center for High-performance Computing

OPEN HACKATHONS
OpenACC