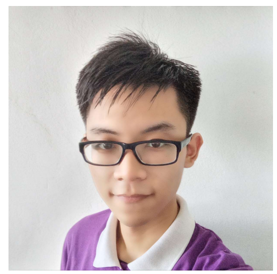
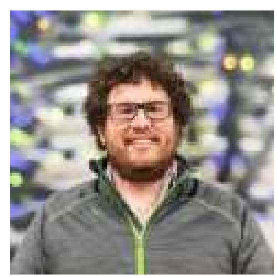
Monofonia

Team member:
Jonathan Ting (ANU)



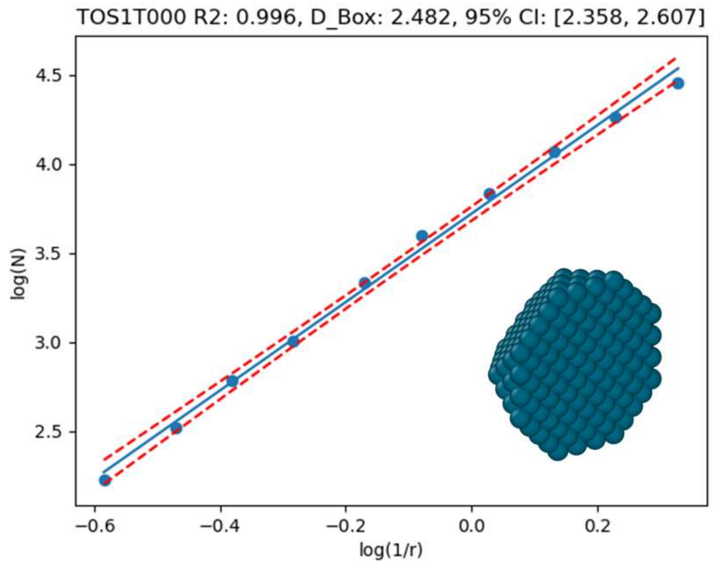
Mentors:
Wei Fang (NVIDIA)
Ben Menadue (NCI)

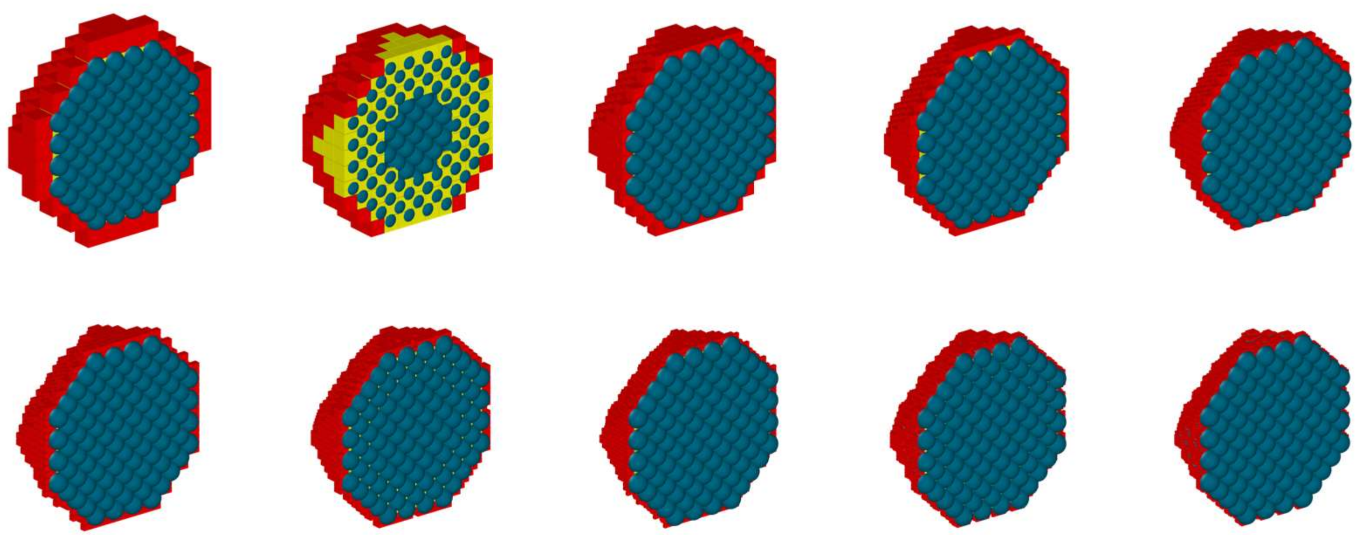


Sphractal

- Problem: Quantify roughness/complexity of spherical surfaces through box-counting algorithm
- Algorithm driver: Very common technique for shape complexity quantification
- Motif: Materials or even molecular design
- Focusing on: 3 embarrassingly parallel for loops

Sphractal


TOS1T000 R2: 0.996, D_Box: 2.482, 95% CI: [2.358, 2.607]
log(N)
log(1/r)
4.5
4.0
3.5
3.0
2.5
−0.6
−0.4
−0.2
0.0
0.2



Evolution and Strategy

- What was your goal coming here?
  - Speed up the code so that it could handle files with larger input size within reasonable time
- What was your initial strategy?
  - Distribute the lowest level function tasks to different threads on GPU
- How did this strategy change?
  - Realised that the lowest level functions that was called all the time doesn’t take up too much compute, just accumulative
  - So might be better to speed up that part instead

Results and Final Profile

- What were you able to accomplish (speedup)?
  - 1 CPU: 103.67 s to 23.09 s (~4.5 times)
  - 12 CPUs: 18.40 s to 10.10 s (~2 times)
  - 48 CPUs: 7.24 s
  - 192 CPUs: 7.23 s

- What did you learn?
  - Numba from scratch (mainly jit and guvectorize)
  - Code profiling using NVTX & Nsight Systems
  - CUDA programming hierarchy
  - Lesson: Think about potential future optimization when designing data structure

Energy Efficiency

The calculator will compare energy consumption of a number of CPU only nodes with dual CPUs required to perform the same amount of work as 1 GPU node with 2 CPUs and 8 GPUs.

INPUTS	
# CPU Cores	12
# GPUs (A100)	1
Application Speedup	4.5x

Node Replacement	3.4x

GPU NODE POWER SAVINGS			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (W)	3,713	6,500	-2,788
Networking Power (W)	157	93	64
Total Power (W)	**3,869**	**6,593**	-2,724

Node Power efficiency	**0.6x**

ANNUAL ENERGY SAVINGS PER GPU NODE			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (kWh/year)	32,522	56,940	(24,419)
Networking Power (kWh/year)	1,373	814	559
Total Power (kWh/year)	**33,894**	**57,754**	**(23,859)**

$/kWh	$	0.34
Annual Cost Savings	$	(8,112.12)
3-year Cost Savings	$	(24,336.35)

Metric Tons of CO2	(17)
Gasoline Cars Driven for 1 year	(4)
Seedlings Trees grown for 10 years	(280)

(source: Link)

1. Use this calculator for your report
2. Add your acceleration numbers in the INPUTS section
3. Modify $/kwh number if necessary
4. Paste a screenshot similar to the one on the right in this slide to report energy efficiency of your project

What problems have you encountered?

- My innermost for loop wasn’t doing too much computation in each iteration, balancing between data-moving to GPU and speedup gained via parallelism is tricky
- Numba doesn’t support a lot of Python functionalities I was relying on
- Numba error messages not too helpful at times
- @nvtx.annotate cannot be used to decorate @numba.jit functions that are called by other @numba.jit functions

Wishlist

- What do you wish existed to make your life easier?
  - Better error messages
  - More tutorials on Numba

Was it worth it?

- Was this worth it?
  - Absolutely!
- Will you continue development?
  - For sure.
  - Fix the little discrepancy.
  - Implement GPU code using Numba + CUDA
  - Publish to PyPI and Conda
- What sustained resources/support will be critical for your work after the event?
  - The Slack channel (and a few people in it)

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

During the event, I had the privilege of being assigned a mentor from NVIDIA. Starting as a complete beginner in Numba, I quickly grasped the intricacies of the framework and am now able to avoid common low-level mistakes. I managed to accelerate my code by 4.5 times via its just-in-time compilation functionalities. Moreover, I acquired valuable skills in profiling and debugging through the use of NVTX annotations and NSight Systems. I have also gained a better understanding of the hierarchical structures involved in CUDA programming. This newfound knowledge, coupled with my understanding of Numba's GPU programming capabilities, instilled me with confidence to tackle GPU programming, paving the way for future implementations in my code.

PROMOTING YOUR WORK: AVAILABLE OPPORTUNITIES

- **Papers and Talks:** Please acknowledge the Open Hackathons program and OpenACC Organization in any planned or upcoming papers, presentations, or talks.

  “This work was completed in part at the [Event name], part of the Open Hackathons program. The authors would like to acknowledge OpenACC-Standard.org for their support.”
- **Social Media Support:** Please feel free to promote your participation across your social media channels. Tag **@OpenACCorg** and **#OpenHackathons** and we are happy to amplify.
- **Blogs and Technical Write-ups**: Create a blog post or technical article that highlights the work being done and results achieved.
- **Quotes and Testimonials:** Highlight your quote or feedback on our channels (i.e. social, website, etc.).

***Please reach out to Izumi Barker (ibarker@nvidia.com) to discuss marketing options and opportunities.**