Filippo Zallocco

Professor Turbé

CS610 - 71429

December 15, 2023

**Assignment 9 - Technical report: GPU-Based Parallel Computing**

1. **Evolution of GPUs and their capabilities from 1990 to today**
2. With the commercialization of video co-processors able to generate video interfaces, sounds, and reading inputs in the 1980s, chip manufacturers like Intel, Sony, Nintendo, Sega, Nvidia, and ATI (later acquired by AMD) began scaling their production of affordable high-performant graphics cards. Indeed, colored pixels were augmented by graphics cards that could map images to the screen in 65,536 possible tone variations. (wikipedia.com 2023, URL#1)
3. Besides graphics breakthroughs, the introduction of hardware acceleration APIs boosted the development of more sophisticated user interfaces and video games, including DirectDraw for 2D rendering in Windows, followed by Direct3 for 3D rendering. (wikipedia.com 2023, URL#1)
4. As the GPU market kept growing in the 1990s, the graphics card industry consolidated. The approximate number of chip manufacturers plunged from thirty-five to twelve, leaving giants like Nvidia, AMD, and Intel to carry the industry forward. As video cards grew in popularity, improved interfaces and games continued to be rolled out. Consequently, consumers and enterprises continued to seek more advanced chips to accommodate their needs and operations.
5. Not all manufacturers could afford consistent investments in research and development. In fact, most of them filed for bankruptcy or were acquired by Nvidia or AMD. Most notably, ATI, which introduced 2D & 3D acceleration for Windows PCs in the early 1990s, was acquired by AMD in 2006 following a sequence of acquisitions and mergers with key industry payers. ATI had acquired underperforming companies, namely Tseng Labs and ArtX, which pioneered competitive products for PC and Nintendo in the early 2000s. Hence, by acquiring ATI, AMD also secured markets that later supported its rise as Nvidia’s top competitor. (wikipedia.com 2023, URL#2 & URL#3)

**2) Analysis of current top GPU products (Nvidia, AMD, and Intel)**

1. Nvidia sold 30.34 million graphics chips in 2022, accounting for 58.95% of its total revenue for the fiscal year –approximately $27B–. By comparison, AMD generated $23.6B in total revenue –with the gaming segment making up 28.81% of it–, selling 6.76 million GPU units in 2022. Intel recorded $63.1B in net revenue at the end of the 2022 fiscal year, and its graphics cards segment contributed with approximately 760,000 units sold. (tomshardware.com 2023, URL#4, SEC.com 2023, URL#5 & URL#8, AMD.com 2023, URL#7, statista.com 2023, URL#6)
2. Nvidia GPUs have consistently met user expectations. They have dominated the graphics card market for decades because they are at the forefront of the latest innovations. Both Nvidia’s RTX 40 series and AMD’s RX 7000 cards, starting from $300 and $270, respectively, are equipped with ray tracing technology, but AMD cards consistently trail behind Nvidia cards when tested on demanding tasks, such as running video games like Cyberpunk 2077. The root of the problem, in this case, is not hardware; it is, in fact, software. Nvidia leads not only in capable hardware but also in embedded AI systems that adapt the software running on its cards to make the most out of the resources available. Intel is relatively new in the commercial-grade GPU market. It released its first high-performance graphics chip, Arc A750, in the third quarter of 2023. The graphics card has 28 cores, ray tracing, 2.05 GHz clock speed, and 8 GB of GDDR6 memory. While this card can compete with the likes of AMD’s RX 6700, it still lacks support from past rendering APIs, such as DX9, making it a poor-performing solution with early 2000s hit video games. Nonetheless, Intel’s first generation of graphics chips proved to be a fine alternative for AI training and ray tracing technology. (businessinsider.com 2023, URL#9, nvidia.com 2023, URL#10, amd.com 2023, URL#11, intel.com 2023, URL#13)
3. GeForce RTX 4090 comes with 16384 CUDA cores, 2.52 GHz clock speed, and 24 GB of RAM of type GDDR6X. On the other hand, AMD’s RX 7900 XTX offers 96 compute units, a clock speed between 2.3 and 2.5 GHz, and 24 GB of RAM of type GDDR6. What sets the RTX 4090 apart from AMD’s alternative is the fact that this unit supports the CUDA API, which enables one of the fastest parallel computing performances to date. It follows that game enthusiasts and AI researchers prefer Nvidia to AMD. (tutorialspoint.com 2023, URL#14)
4. Gaming is not the only market where Nvidia and AMD compete, the two manufacturers challenge each other in data centers–where AMD supplies the most capable CPUs for AI, and Nvidia supports cutting-edge GPU acceleration for its customers. AMD does not only sell CPU power to enterprises through data centers. It is a prominent challenger to Intel in the consumer-grade PC market as well, partnering with Acer, Asus, Gigabyte, and MSI. Furthermore, AMD enjoys a close relationship with gaming console manufacturers, such as Sony, Microsoft, and Steam. Differently, Nvidia's partners are largely big data enterprises, including HP Enterprise, AWS, Google, VMware, IBM, and Dell. Hence, the two chip manufacturers may not compete at all in three out of the four aforementioned markets. (thestreet.com 2023, URL#12)
5. A GPU-based computer consists of a large number of nodes placed next to each other, and they all execute instructions concurrently. Its use case is often with processing complex data structures like 2D and 3D arrays designed for enhanced graphics such as images and videos, hence the name Graphics Processing Unity or video card. A GPU cluster is an example of a super-computer made up of numerous compute nodes, each containing several processors working concurrently, that communicate with each other via cables. This type of infrastructure provides game-changing computing power, but unlike traditional PCs or single GPUs, it requires accurate instructions to divide up the data among the nodes and keep those nodes working to mitigate communication delays. The US Air Forces’s research lab built one of the fastest super-computers by using a cluster of 1760 PlayStation 3s, each featuring a single hybrid chip with a dual-core CPU with an 8-core GPU. Because of the chip’s ability to run Linux, researchers succeeded in programming the cluster to process satellite images and AI training, reducing the capital expenditure to one-tenth of the cost of building an industry-grade supercomputer. (quora.com 2020, URL#16, youtube.com 2023, URL#17 )

**3) GPU Developers’ Tools: OpenMPI vs CUDA**

j) Open MPI is an open-source message-passing API introduced and supported by researchers from various universities whose goal is to deliver a comprehensive library with the latest resources for high-performance computing. Some Open MPI’s capabilities include thread safety and concurrency, dynamic process spawning, network and process fault tolerance, support network heterogeneity, multi-network single library support, run-time instrumentation, and multi-job scheduling support. Nvidia-patented CUDA, Compute Unified Device Architecture, is a parallel programming framework, built on a shared memory paradigm, that allows for direct control of the CPU bypassing the assembly language. CUDA also takes advantage of the many Arithmetic Logic Units in both the CPU and GPU to moderate the program’s performance between sequential and parallel execution. (open-mpi.org 2023, URL#18, tutorialspoint.com 2023, URL#14)

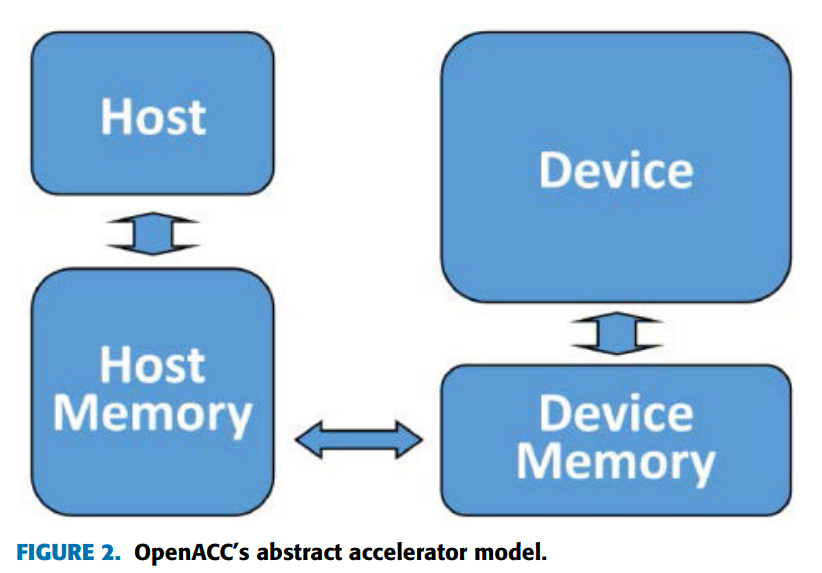
k) Being a particular implementation of MPI, Open MPI offers high scalability for big clusters of computing nodes. Considering Open MPI’s message-passing system, it is recommended to employ the API only when large volumes of data are to be assigned to nodes with local memory. Differently, CUDA excels at running small parallel algorithms as it provides simple instructions in languages like C and C++ to run algorithms without excess overhead code, dramatically reducing the time taken to implement them. In addition, CUDA supports both CPU and GPU computations, intelligently assigning sequential code to the CPU and parallel code to the GPU. (quora.com 2019, URL#19)

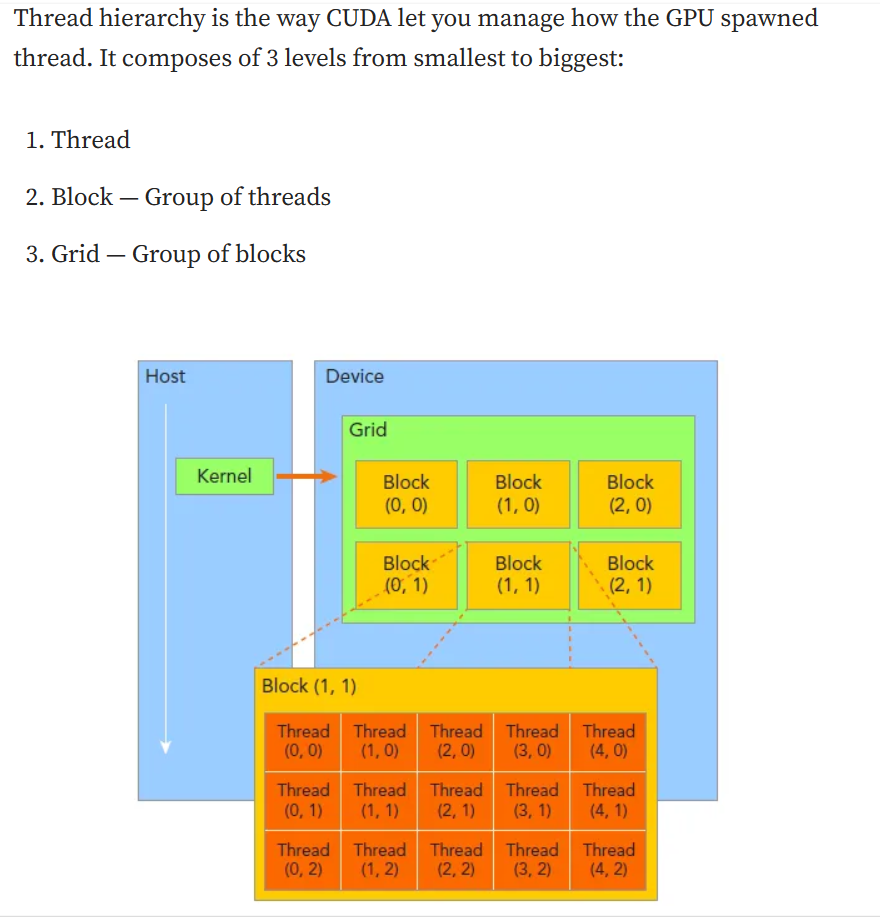
l) Open MPI implicitly renders the implementation of parallel programs challenging because it requires accuracy in assigning data to parallel compute nodes and implementing message-passing systems with little delay. Considering its architecture, the use of Open MPI is only justified when working with large batches of data. By contrast, CUDA underperforms at running parallel algorithms that deal with large volumes of data due to the size of its shared memory. Hence, for this kind of operation, it is advised to resort to Open MPI and mitigate communication delays among the nodes with more workload per node. (quora.com 2019, URL#19)

m) Academic use cases for both Open MPI and CUDA libraries were found in geology courses at select UK universities, namely, Northumbria University, University of Edinburgh, and University of Aberdeen. High-Performance Computers, or HPCs, are capable of running more complex 3D models; however, carrying out several simulations for scientific projects like ground-depth mapping can be time-consuming and expensive. For this reason, researchers employ Open MPI to take full advantage of distributed memory machines and CUDA to optimize parallel computations on Nvidia GPUs. Finally, to run tests effectively, researchers resort to the Python library Pi CUDA to write parallel code for Nvidia GPUs. (youtube.com 2020, URL#20)

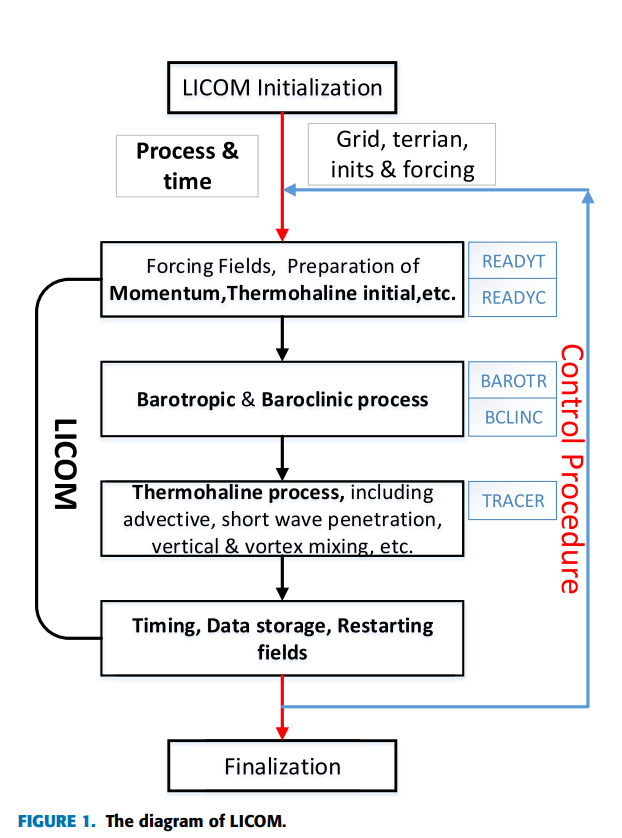
**4) Analysis of modern-day GPU-based parallel programming for global climate modeling simulations**

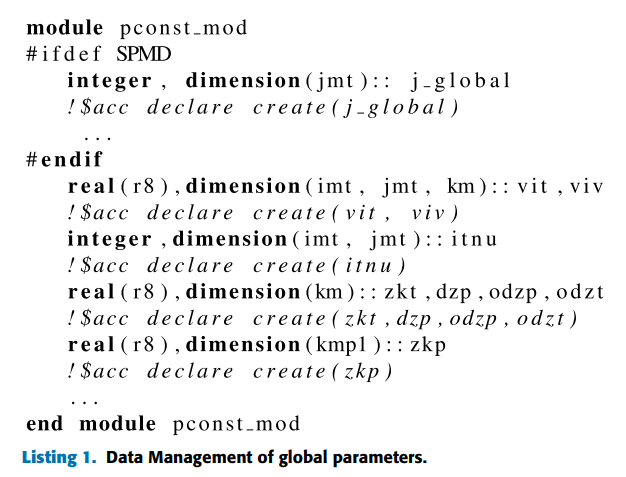
Nvidia’s CUDA library provided the groundwork for GPU acceleration in parallel computing, enabling fast simulations for predictive models. While CUDA yielded notable speedups in rendering the output of certain advanced A.I. models, its low-level implementation and difficulty in porting CPU code to the accelerator caused engineers to adopt a new parallel programming interface: OpenACC –the result of a joined effort from Nvidia, PGI, and Cray–. OpenACC supports data locality, loop, and interprocess optimization, thus making it a compelling alternative to CUDA for complicated predictive models. A real-world example of the application of OpenACC in GPU acceleration is in climate models. These algorithms often have enormous source code, considering the types of data to process, and are written in Fortran programming language by multiple engineers. Therefore, researchers prefer to use an easier interface than CUDA to port their code to the GPU. One such example of a climate model written in OpenACC to achieve fast GPU acceleration is the LICOM2 ocean model optimized for data locality, loops, and interprocess communication. (bostonuniversity.edu 2023, URL#21)

CUDA OpenACC

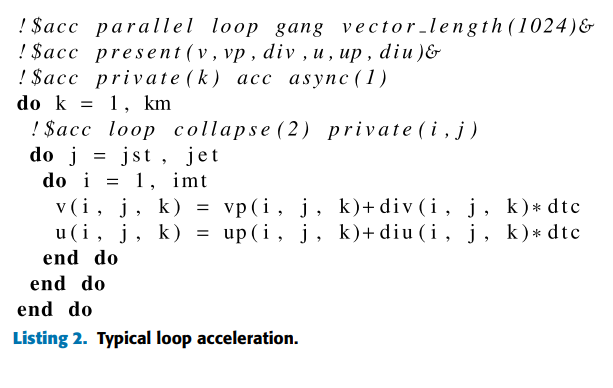


n) Below is the following algorithm written in Fortran and OpenACC:

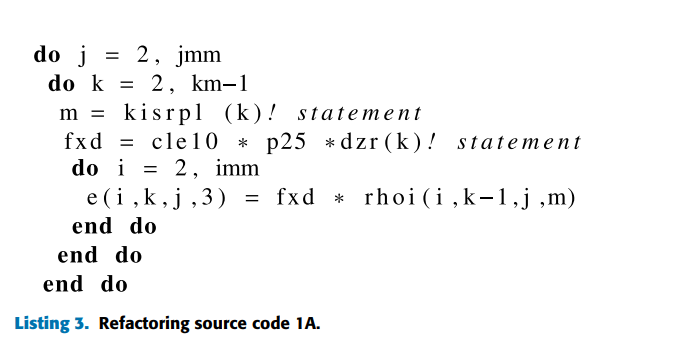




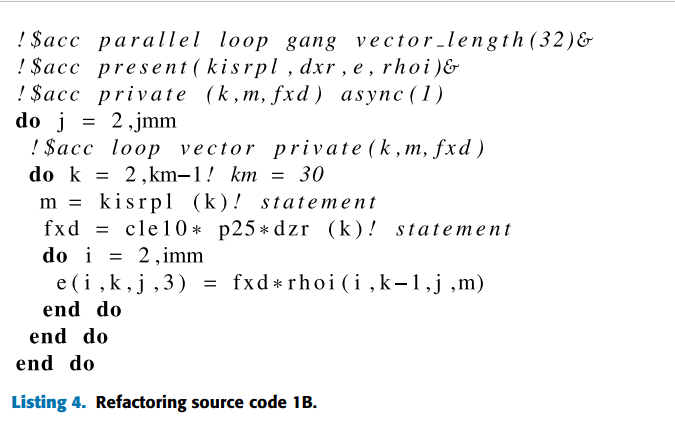
Declaring global parameters and data on the GPU to minimize data exchange between it and the CPU.



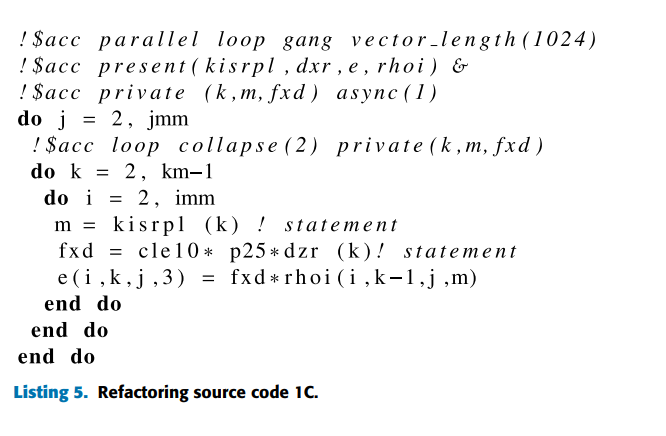
The inner loop generates an n-number of blocks based on the number of outer loop iterations.



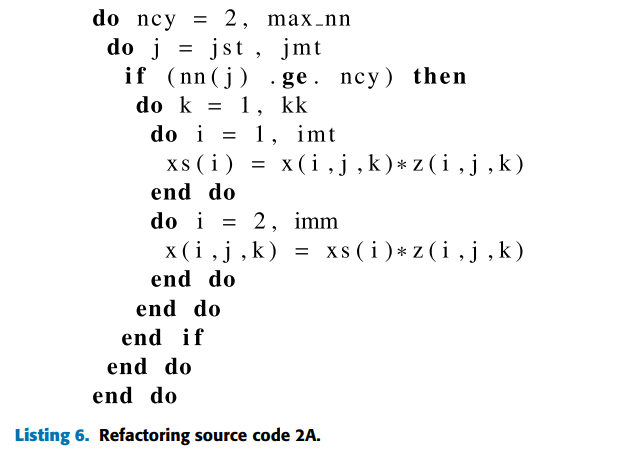
The code is refactored by moving the instructions from the outer loop into the innermost loop.



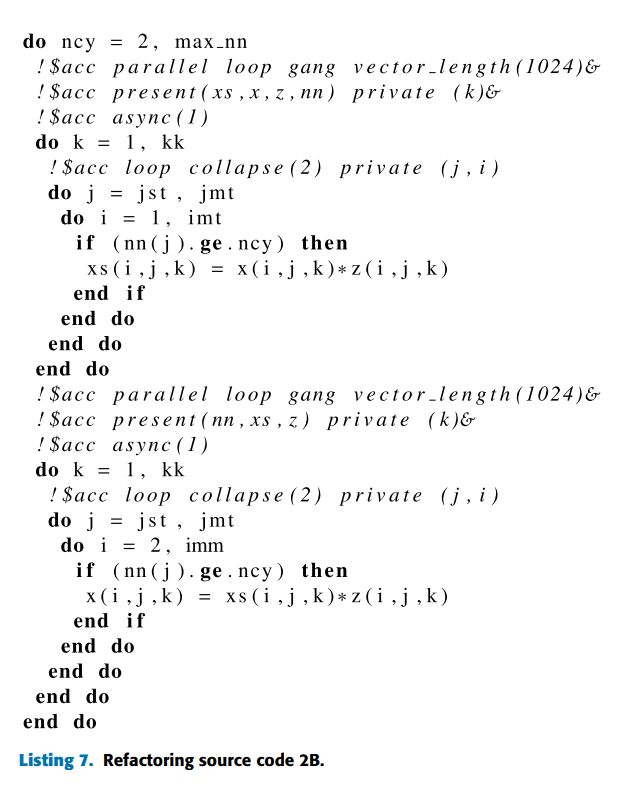
The number of cycles of the inner loop will be greater than the number of iterations performed by the outer loop



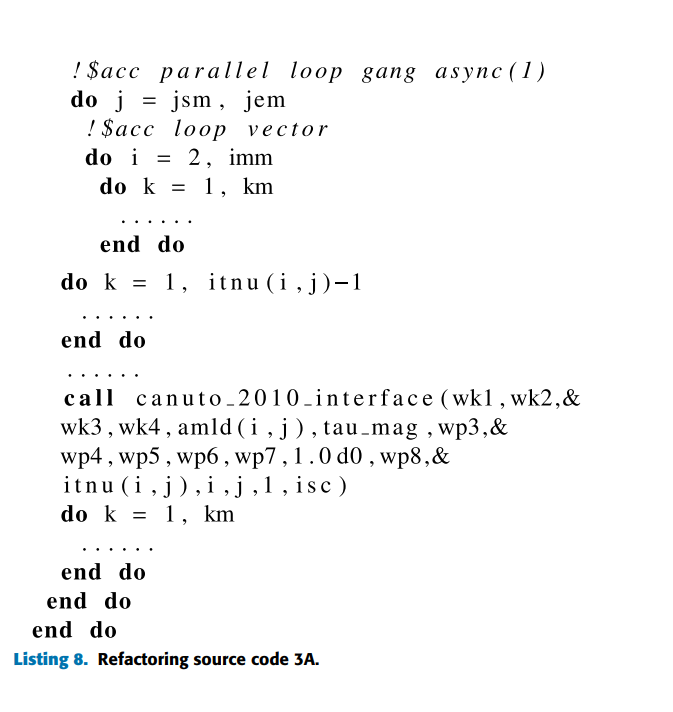
We must merge the two outer loops, k and i, to assign 1024 processes (threads) to each block created



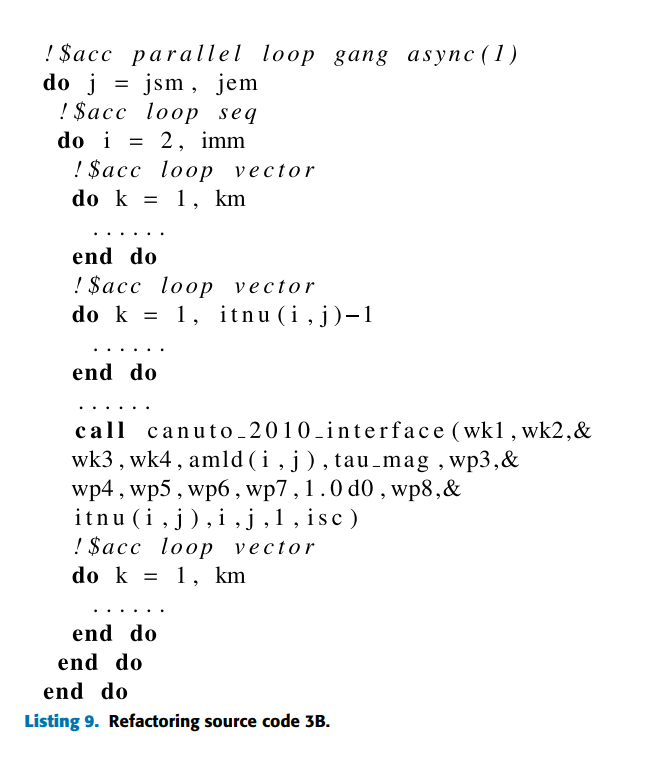
Here, the number of inner iterations is larger than that of the outer loops.

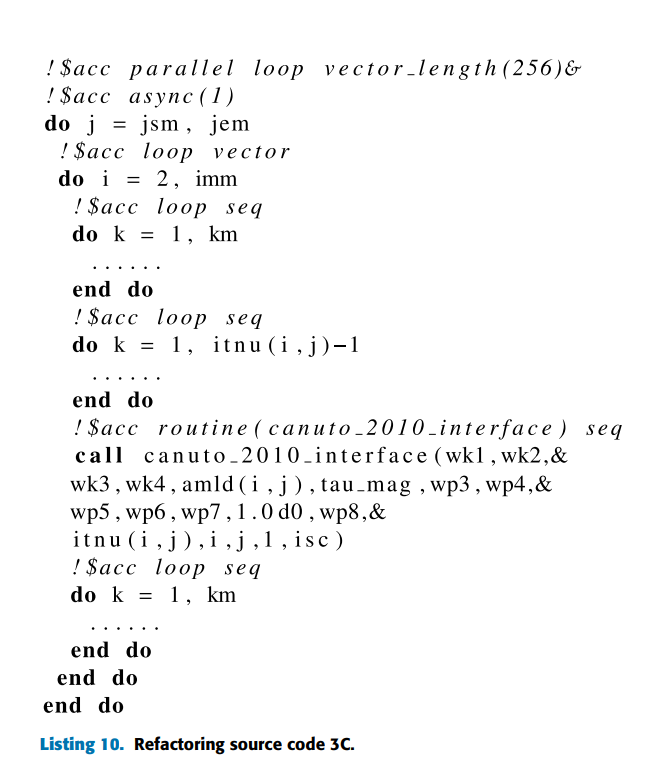


To split the two inner i-loops, we must update the size of array-x, updating the variables to i, j, k.

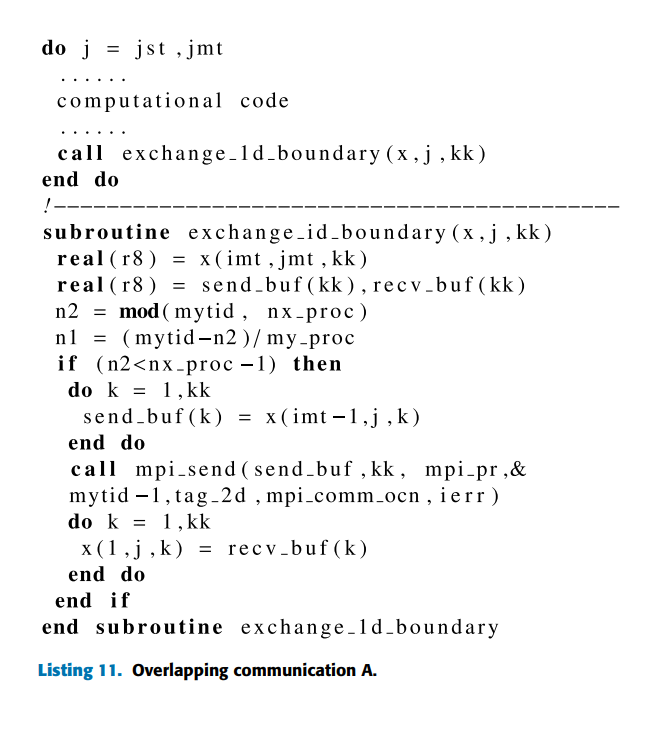


Here, loop i will be ignored because the compile does not know whether the callable function is sequential or parallel. Hence, it will create threads from the k-iterator.

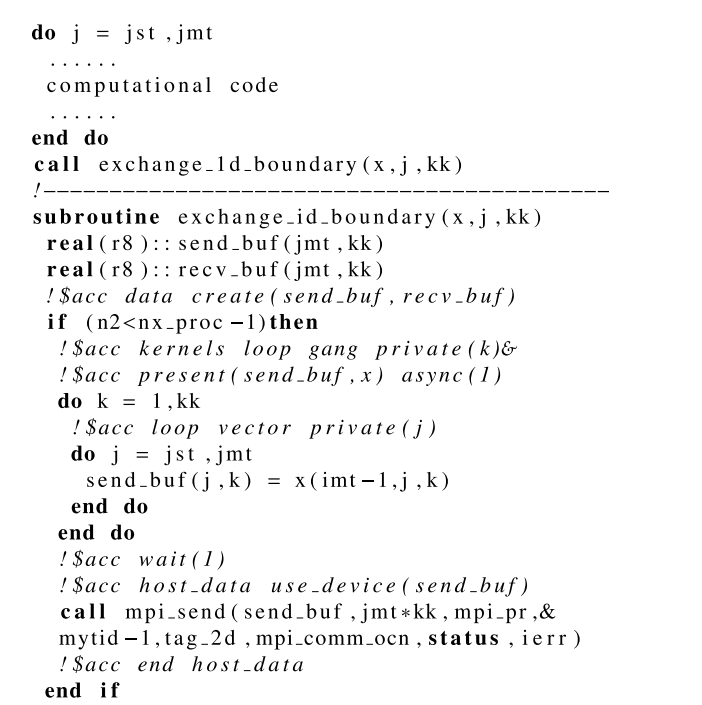


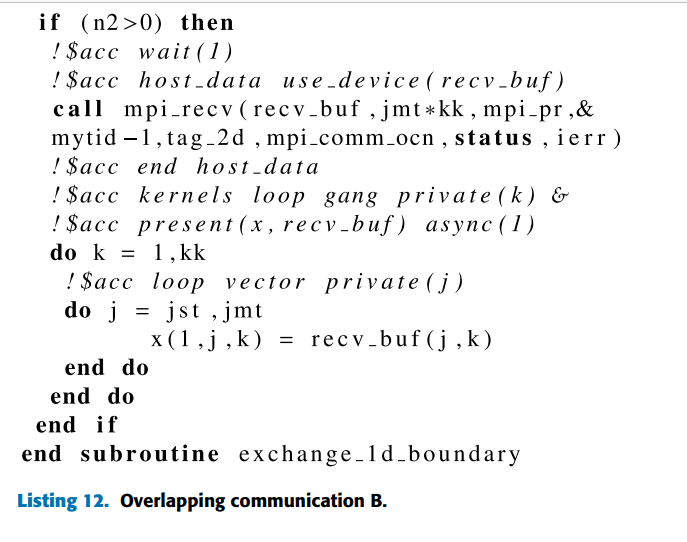


Snapshot 10 shows the correct implementation of the code as the number of k sequential iterations is specified before the loop and the canuto\_2010\_interface, and vector\_length is also defined.



GPU performance is greatly impacted by communication delays. The program, here, uses mpi\_send and recv\_but() to exchange data that will not be used by j.

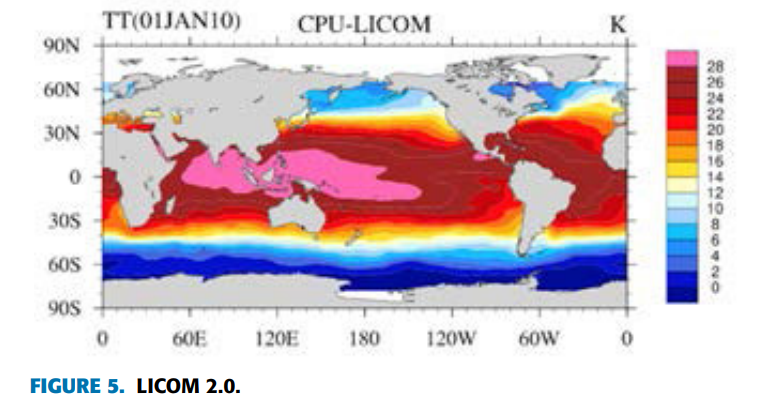
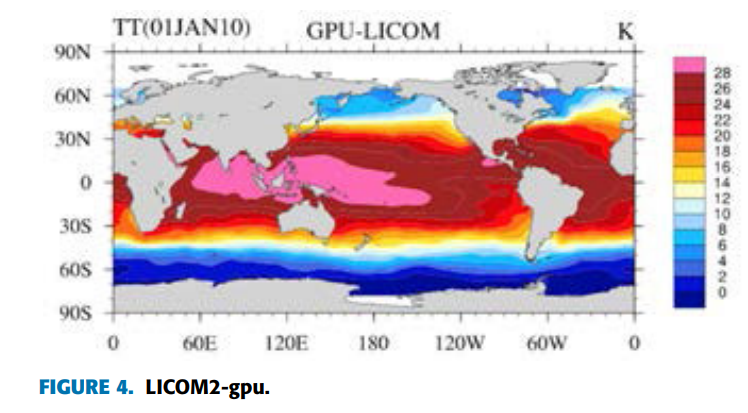


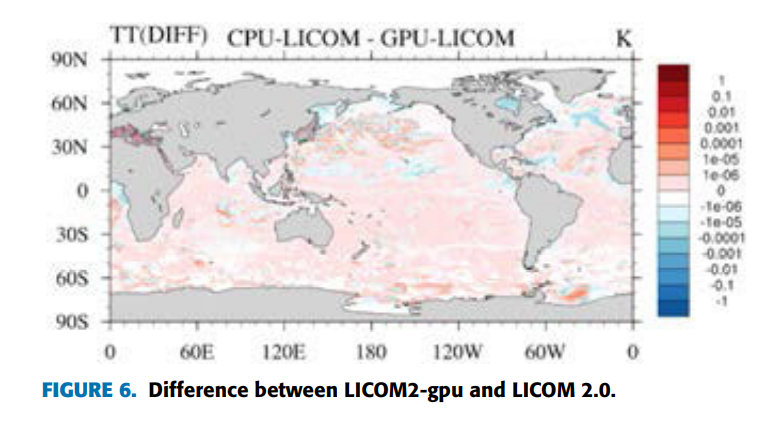


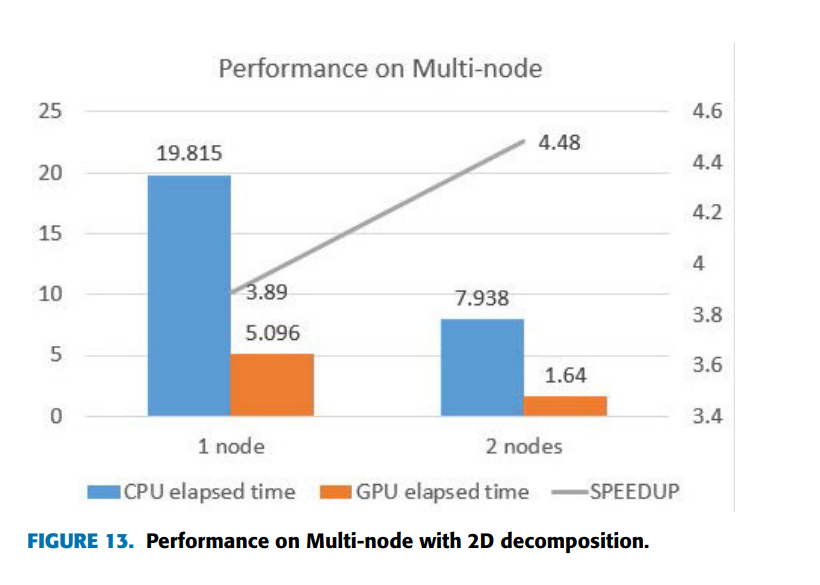
Here, Snapshot 12 explains that we have to pass more memory as parameters inside the send and receive functions. This way, we instruct the GPU to communicate its shared memory in parallel instead of repeating the same actions after calculating new values. (IEEE.org 2019, URL#24)

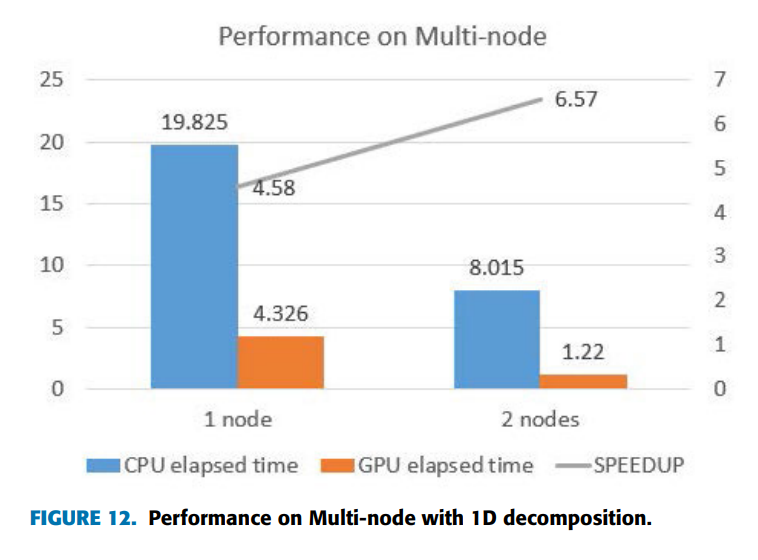
o) Three datasets, such as temperature, downward shortwave radiation, and precipitation, were used in the testing of the GPU-accelerated LICOM algorithm. The temperature dataset was saved inside the variable 2-m temperature and contains T62 input images recorded with 6-hour frequencies from 1948 through 2007. Next, downward shortwave radiation was stored in a homonymous variable, which also contains daily frequencies of T62 input images going from 1948 until 2007. Subsequently, Precipitation - Runoff, the merger of three datasets, stores inputs with T62 resolution, and each input varies by monthly intervals recorded from 1948 till 2007. (IEEE.org 2019, URL#24)

p) The LICOM2 parallel program’s modules Canuto\_2010\_interface and convadj cause the compiler to fail to fetch blocks and threads, dramatically reducing the capabilities of shared memory. As a consequence, the authors of the journal revisited the program’s logical dependencies Canuto\_2010\_interface and convadj. Upon studying the difference between the CPU-trained and the GPU-trained models, they concluded that the discrepancy between the two is negligible. However, subsequent tests of CPU-intensive LICOM2.0 and GPU-intensive ones revealed that the CPU carried out the same instructions with 16 compute processes, or threads. whereas the GPU only worked with 8. All previous tests were conducted on a 1-node and 2-GPU system, and each test provided average speedups and minor performance gains. Hypothesizing that LICOM had exceeded the cache consistently in the former runs, the researchers observed that the acceleration of 2 GPUs was higher than that of one GPU. Most importantly, reducing the MPI\_communication by dividing the data into 1D decompositions instead of 2D decompositions offered even greater performance. In conclusion, the researchers found that a 1D MPI\_communication parallel system with OpenACC maximizes the performance of graphics acceleration AI systems for climate model rendering. (IEEE.org 2019, URL#24)







q) The commercial and ecological applications of parallelizing one-sided MPI communication of a parallelized OpenACC system are numerous. OpenACC allows for compute and message-passing optimization while maintaining code integrity. Thereby, leveraging a GPU cluster, researchers can develop more accurate climate, ocean, and geological predictive models, thus providing actionable results to governments, non-profits, and other organizations to promptly avert natural disasters.

**Works Cited**

Part 1:

URL#1: [Graphics processing unit - Wikipedia](https://en.wikipedia.org/wiki/Graphics_processing_unit)

URL#2: [Tseng Labs - Wikipedia](https://en.wikipedia.org/wiki/Tseng_Labs)

URL#3: [ArtX - Wikipedia](https://en.wikipedia.org/wiki/ArtX)

Part 2:

URL#4: [Nvidia Maintains Dominance as Sales of Graphics Cards Hit All-Time Low in 2022 |](https://www.tomshardware.com/news/nvidia-maintains-lead-as-sales-of-graphics-cards-hit-all-time-low-in-2022-jpr#:~:text=With%2030.34%20million%20desktop%20discrete,market%20share%20away%20from%20AMD.)

URL#5: [Document (sec.gov)](https://www.sec.gov/Archives/edgar/data/1045810/000104581022000008/q4fy22pr.htm)

URL#6: [Nvidia revenue by segment 2023 | Statista](https://www.statista.com/statistics/988034/nvidia-revenue-by-segment/)

URL#7: [0001193125-23-088137.pdf (amd.com)](https://ir.amd.com/sec-filings/filter/annual-filings/content/0001193125-23-088137/0001193125-23-088137.pdf)

URL#8: [Document (sec.gov)](https://www.sec.gov/Archives/edgar/data/50863/000005086323000005/q422_earningsrelease.htm)

URL#9: [AMD Vs. Nvidia: How to Choose the Right Graphics Card for Your PC (businessinsider.com)](https://www.businessinsider.com/guides/tech/amd-vs-nvidia-graphics-cards#:~:text=Target%20resolution%20and%20refresh%20rate*,-RX%207600&text=However%2C%20Nvidia%20cards%20tend%20to,series%20with%20slower%20frame%20rates.)

URL#10: [GeForce RTX 40 Series Graphics Cards | NVIDIA](https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/)

URL#11: [AMD Radeon™ RX 7900 XTX | AMD](https://www.amd.com/en/products/graphics/amd-radeon-rx-7900xtx)

URL#12: [Nvidia Stock: Could AMD Surpass It in Gaming? - Meme Stock Maven (thestreet.com)](https://www.thestreet.com/memestocks/others/nvidia-stock-could-amd-surpass-it-in-gaming#:~:text=Like%20Nvidia%2C%20AMD%20positions%20itself,5%2C%20and%20the%20Steam%20Deck.)

URL#13: [Intel® Arc™ A750 Graphics](https://www.intel.com/content/www/us/en/products/sku/227954/intel-arc-a750-graphics/specifications.html)

URL#14: [CUDA (tutorialspoint.com)](https://www.tutorialspoint.com/cuda/cuda_tutorial.pdf)

URL#15: [Nvidia GPU Architecture (youtube.com)](https://www.youtube.com/watch?v=lGefnd7Fmmo)

URL#16: [What's the difference between GPU and cluster computation? - Quora](https://www.quora.com/Whats-the-difference-between-GPU-and-cluster-computation)

URL#17: [Why did the PS3 Supercomputer exist? (youtube.com)](https://www.youtube.com/watch?v=Zv5Gj3mKblI&list=PLtr3JKh9YHSuaOqDn3vdqvDLlxmg29DWs&index=262)

Part 3:

URL#18: [Open MPI: Open Source High Performance Computing (open-mpi.org)](https://www.open-mpi.org/)

URL#19: [Is CUDA a better method for writing parallel code than openMPI? - Quora](https://www.quora.com/Is-CUDA-a-better-method-for-writing-parallel-code-than-openMPI)

URL#20: [Day 3 - Speed up your simulations with OpenMP, MPI, & CUDA - Dr Craig Warren (youtube.com)](https://www.youtube.com/watch?v=19gkYlRozL0)

Part 4:

URL#21: [Introduction to OpenACC (bu.edu)](https://www.bu.edu/tech/files/2017/04/OpenACC-2017Spring.pdf)

URL#22: [OpenACC Use (hpe.com)](https://www.hpe.com/psnow/resources/ebooks/a00113892en_us_v2/OpenACC_Use.html#:~:text=There%20is%20a%20hierarchy%20of,to%20the%20same%20global%20memory.)

URL#23: [CUDA vs OpenACC: Performance Case Studies with Kernel Benchmarks and a Memory-Bound](https://ieeexplore.ieee.org/document/6546071)

URL#24: [Porting LASG/ IAP Climate System Ocean Model to Gpus Using OpenAcc | IEEE Journals &](https://ieeexplore.ieee.org/document/8784044)