



Journey towards Exascale Computing

Wileam Yonatan Phan
Lawrence Berkeley National Laboratory
October 4, 2021



Abstract

The rapid advances in computing in the past 30 years have transformed the world in countless ways. I will discuss how computing technology has influenced my life in different ways: my childhood upbringing, throughout high school and college, up to graduate school and now. I will also discuss recent trends in supercomputing hardware, notably accelerated computing with Graphics Processing Units (GPU). Finally, I will discuss the current outlook of high performance computing from my point of view and what it means for pre-exascale systems such as Perlmutter at NERSC.

About me



Early childhood



Image credit:
Wikimedia

- Born and raised in Jakarta, Indonesia
- Dad is an electrical engineer by training
- Was fortunate enough to receive an early exposure to computers and early internet (w/ dial-up connection)
- First computer:
used Toshiba Satellite (Japanese import)
Pentium I 100MHz running Windows 95
- Favorite activity:
Windows malware disinfection

Early childhood

Exposed to the following technologies:



Image credit: Wikimedia



Image credit: Logonoid



Image credit: Popolony2k.com.br



Image credit: WorldVectorLogo

Educational background



2006-2009
High School Diploma
Tunas Bangsa Christian School
Serpong, Banten, Indonesia



2009-2014
BS in Physics
University of Indonesia
Depok, West Java, Indonesia



2016-2021
MS in Physics
University of Tennessee
Knoxville, TN, USA

First exposure to High Performance Computing (HPC)

2013 Computer Cluster Workshop
Indonesian Institute of Sciences
Bandung, West Java, Indonesia
December 11-13, 2013

<http://situs.opi.lipi.go.id/wkk2013/>
(in Indonesian)



Thus my journey into the HPC world begins...




Theoretical/Computational Condensed Matter Physics (TCMP) research group
Department of Physics, University of Indonesia, Depok, West Java, Indonesia
c. 2014



The TCMP research cluster

- Designed, assembled, installed, and maintained a set of 6 workstations for shared usage among TCMP research group users
- Commodity hardware (Intel Haswell) running Ubuntu Linux 12.04 LTS
- Loosely coupled cluster
 - No job scheduler
 - Manual MPI configuration (using hostfiles)
 - 1st phase: 4 workstations; 2nd phase: 2 workstations
 - No discrete GPUs
- Has now grown to ~30 compute nodes



Modeling the graphene-substrate interface (2013 - 2015)

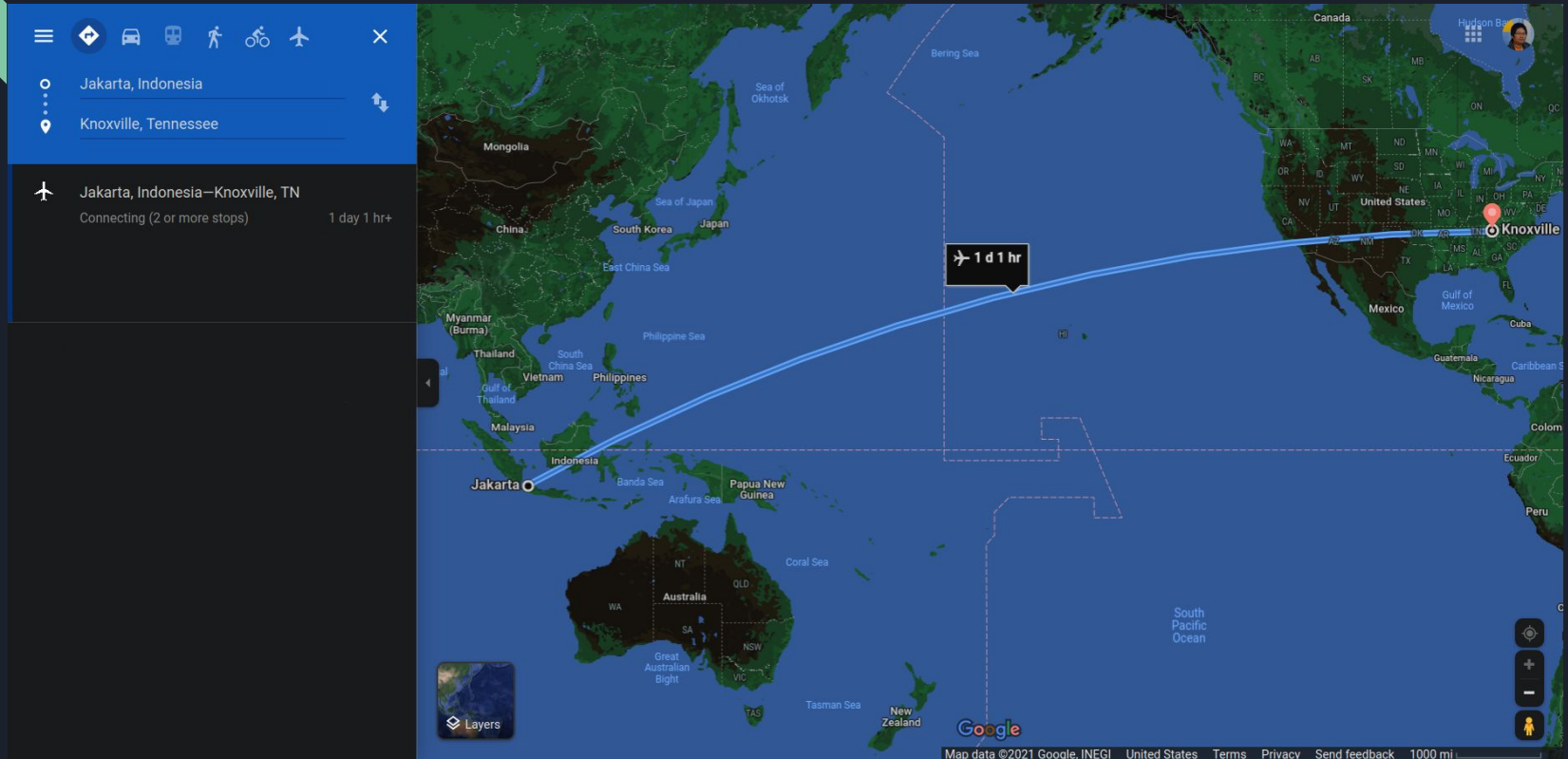
- Undergraduate research thesis
- TCMP research group, Dept. of Physics, University of Indonesia
(<https://physics.ui.ac.id/portfolio/tcmp>)
- Co-advisors: Dr. MA Majidi (University of Indonesia) and Prof. A Rusydi (National University of Singapore)
- Simple phenomenological model based on tight-binding Hamiltonian
- Hybrid MPI + OpenMP code in Fortran 90/95
- Results presented at 1st International Symposium on Current Progress in Mathematics and Sciences (ISCPMS), November 3-4, 2015, Depok, West Java, Indonesia.
- Conference proceedings: <https://doi.org/10.1063/1.4946919>

Modeling the graphene-substrate interface (2013 - 2015)



Undergraduate thesis defense, Depok, West Java, Indonesia, May 30, 2014
From left to right: Dr. D Triyono, Dr. MA Majidi, me, Prof. A Rusydi, Prof R Saleh

The journey continues...





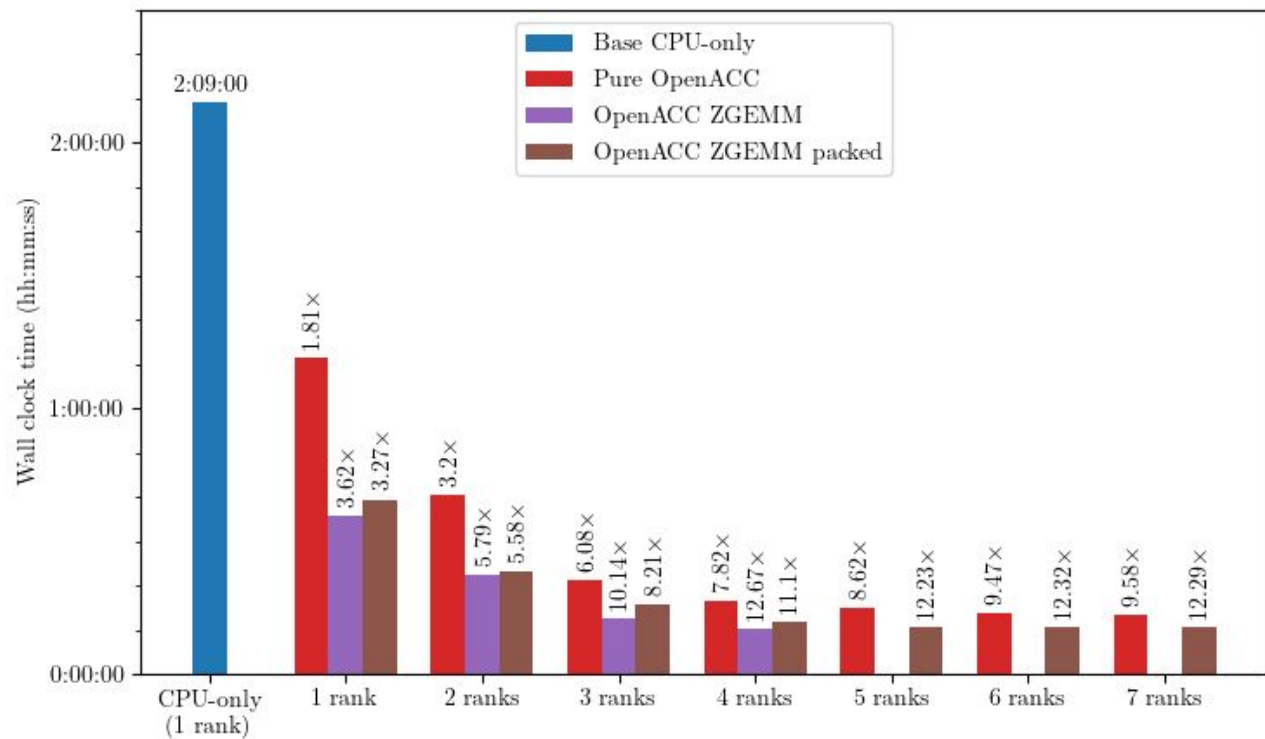
The EXCITING-PLUS project (2019 - 2021)

- Master's thesis
- Advisor: Prof. AG Eguiluz (University of Tennessee, Knoxville)
- Collaboration with Dr. Ed D'Azevedo (ORNL, now retired)
- Sophisticated density response function computational platform
- Based on ELK (<https://elk.sourceforge.io/>) FP-LAPW DFT package
- Hybrid MPI + OpenMP (+ OpenACC) code in modern Fortran
- CPU-only version was awarded the 2010 ACM Gordon Bell prize (Honorary Mention - Performance)
- Previous attempt targeting Titan (c. 2016-2018) was unsuccessful
- Successful port targets Summit NVIDIA V100 GPUs; uses OpenACC (data mgmt) and MAGMA (batched matrix multiply)
- Up to 12 \times wall clock speed-up compared to CPU-only version
- GitHub: <https://github.com/wyphan/exciting-plus-gpu>

The EXCITING-PLUS project (2019 - 2021)

Figure 4.5 (b)
Wall clock time and
speed-up for the c-RPA
calculation on La_2CuO_4
paramagnetic system,
varying number of
MPI ranks per GPU

WY Phan (2021),
master's thesis
(pending publication)





The EXCITING-PLUS project (2019 - 2021)

- Porting process in 3 steps (roughly chronological order):
 - Pure OpenACC implementation
 - OpenACC + MAGMA implementation
 - OpenACC + MAGMA + memory optimization
- Participated in 2020 OLCF GPU Hackathon (team EECM);
mentor: Prof. P Luszczek (University of Tennessee, Knoxville)
- Performed calculations with the ported code on Summit (OLCF) and Cori-GPU (NERSC)
- Designed, assembled, installed, and maintained a developer workstation (AMD Ryzen 5, NVIDIA GTX 1060, AMD Radeon RX Vega 64)
- Implemented continuous integration (CI) using GitHub Actions and self-hosted runners

The EXCITING-PLUS project (2019 - 2021)



In front of Summit supercomputer room (OLCF), Oak Ridge, TN, USA
February 19, 2020



The AMReX framework

- Adaptive Mesh Refinement for eXascale
- Developed at Center for Computational Sciences and Engineering (CCSE), Lawrence Berkeley National Laboratory
- Part of Exascale Computing Project (ECP)
- <https://amrex-codes.github.io/>
- Joined in July 2021 as Scientific Computing Software Engineer (CSE-2)
- Ported several code components from Fortran to C++

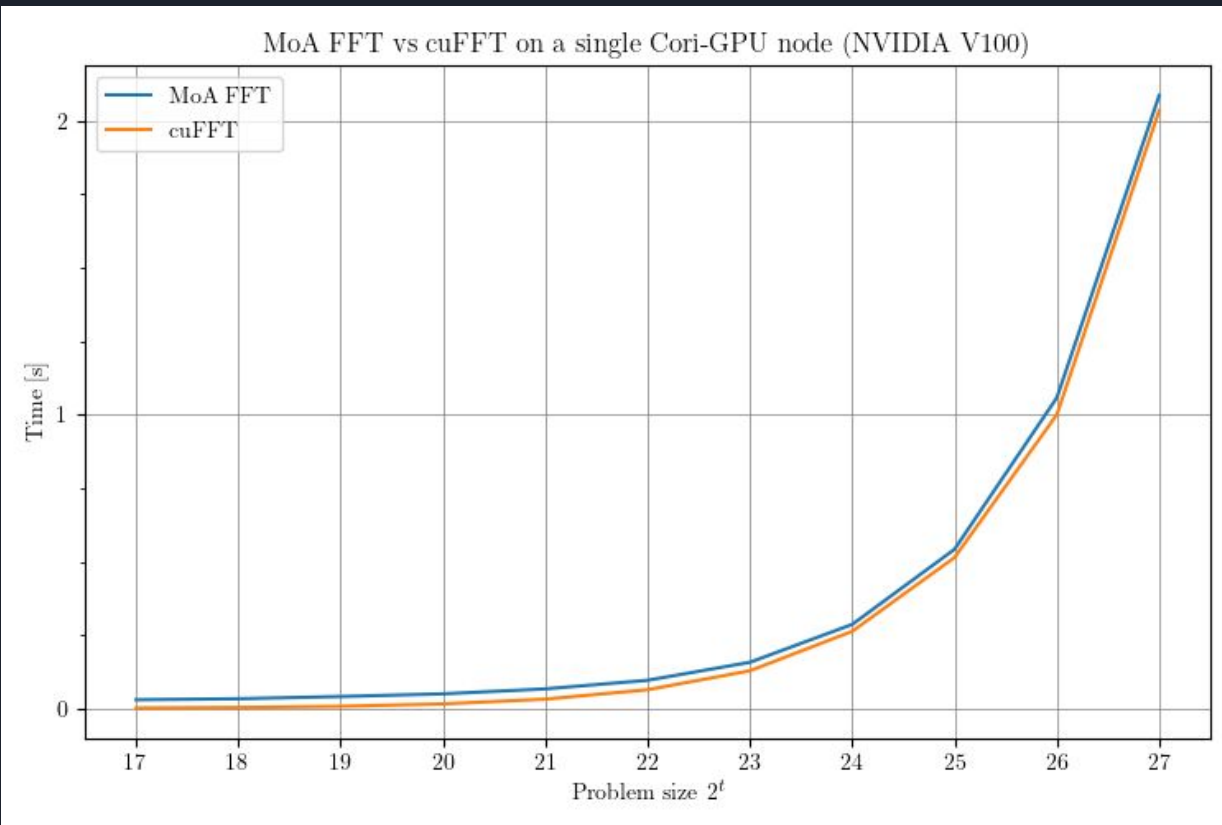


Mathematics of Arrays

- Elegant mathematical theory of n -dimensional arrays by Prof. L Mullin (State University of New York at Albany)
- Everything expressed in terms of array shapes and indexing function
- Syntax heavily inspired by Ken Iverson's APL programming language
- Closure relation added; calculus anomalies from APL removed
- Currently porting Fast Fourier Transform algorithm based on the theory (<https://arxiv.org/abs/0811.2535>) from OpenMP to OpenACC
- Ongoing set of online lectures on the theory and usage
- Presented a talk (together with Prof Mullin) at OpenACC Summit 2021 (<https://www.openacc.org/events/openacc-summit-2021>)

Mathematics of Arrays

- Code written in Modern Fortran + OpenACC
- Calculations performed on NVIDIA V100 GPU at Cori-GPU
- Wall clock timings are competitive compared to cuFFT



Recent trends in HPC

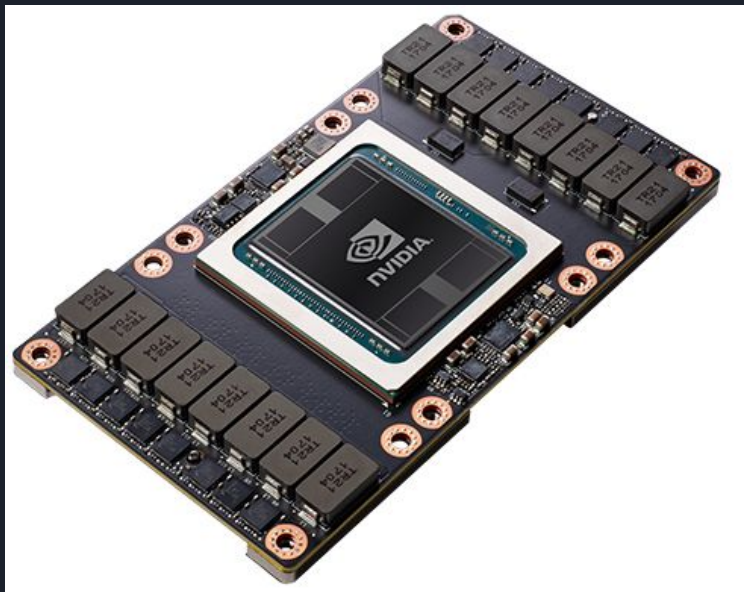
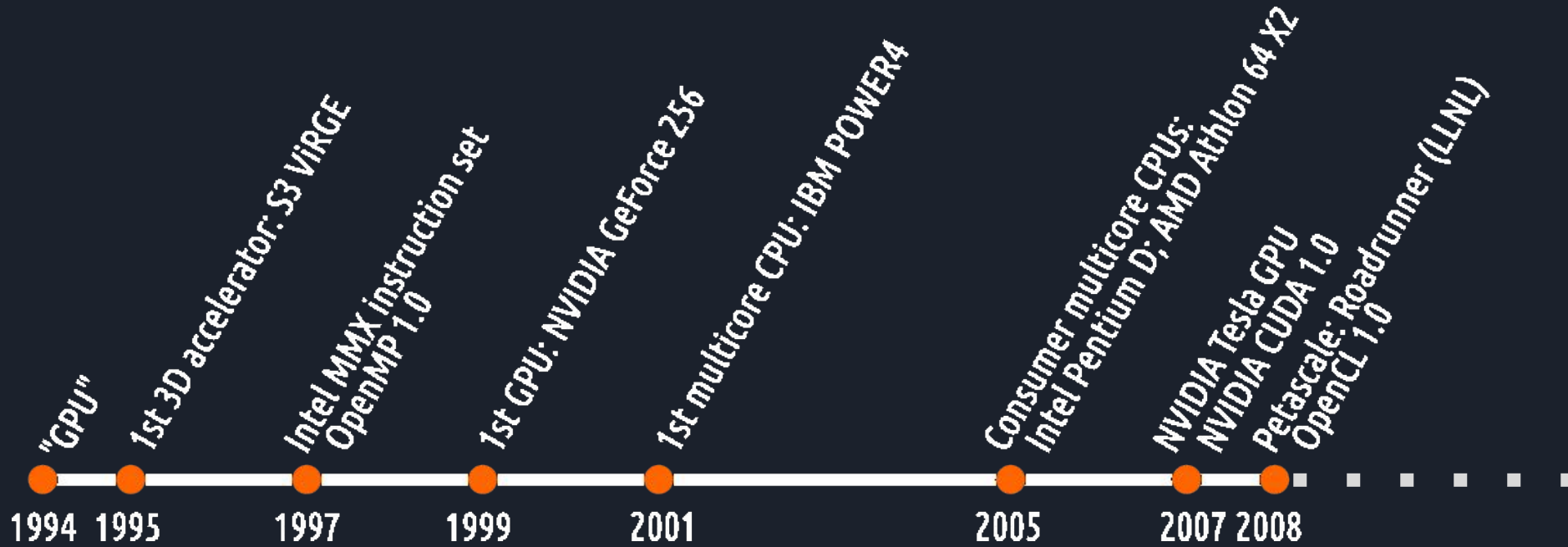
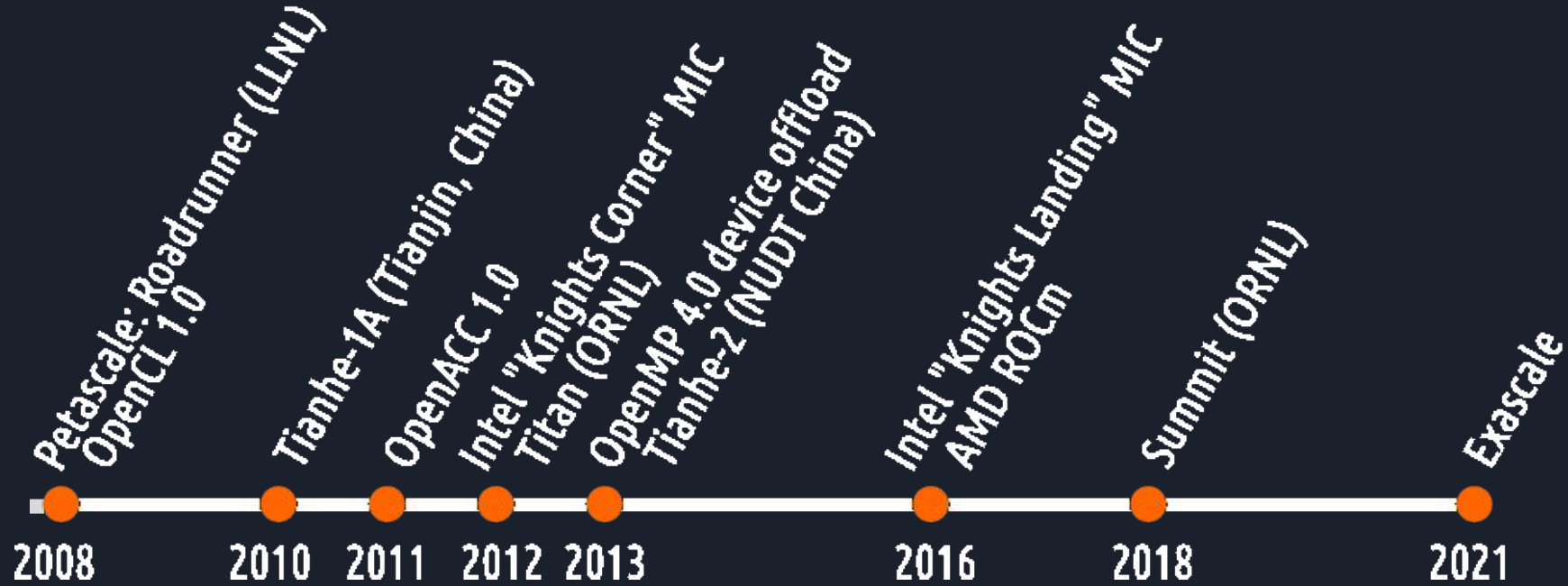


Image credit: NVIDIA

A brief timeline of accelerated supercomputing



A brief timeline of accelerated supercomputing



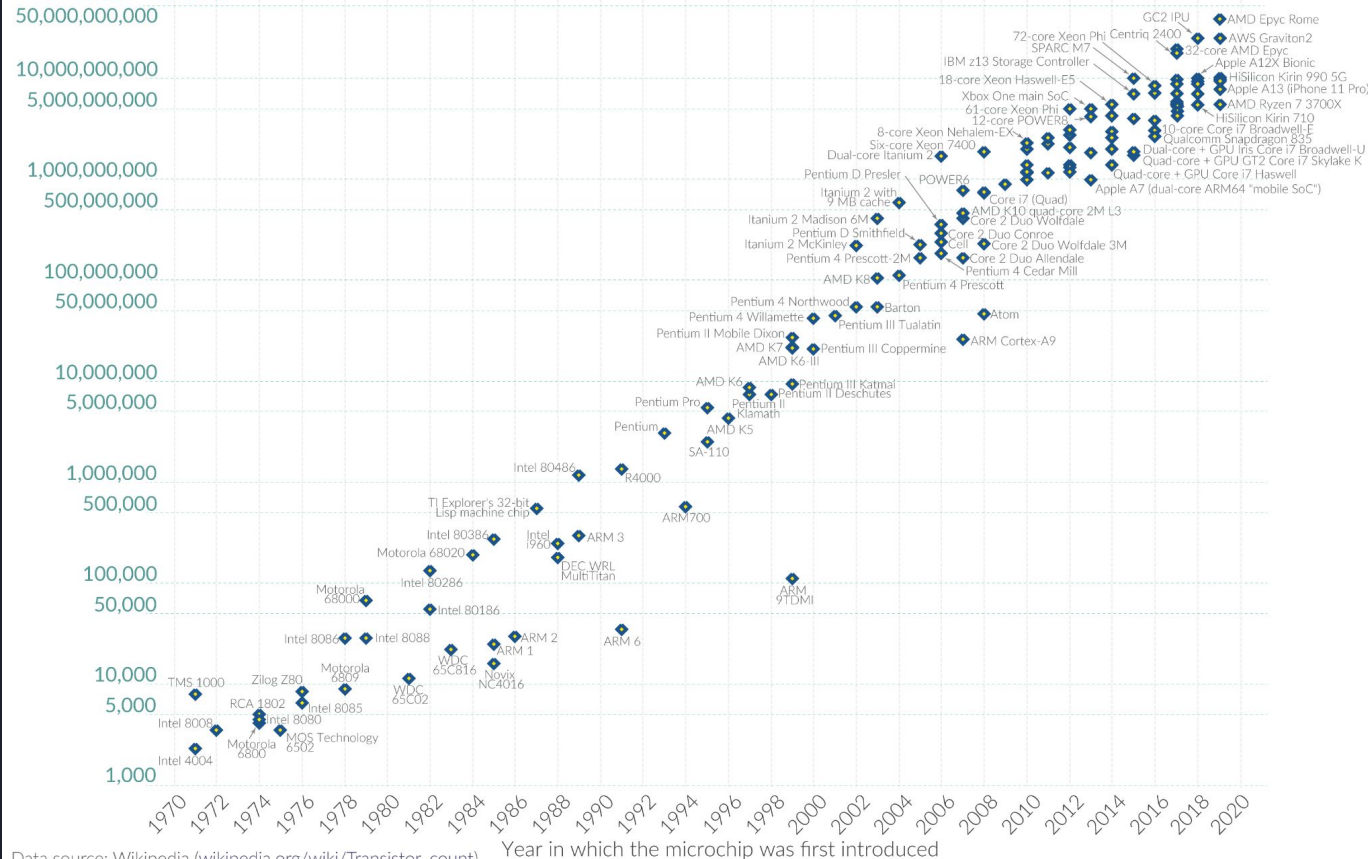
Does Moore's Law still hold?

Moore's Law: The number of transistors on microchips doubles every two years

Our World
in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Transistor count



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Image credit: Wikimedia



Hardware evolution: CPUs

- 1990's:
 - Rising popularity of x86 architecture
 - x86 vector extensions: Streaming SIMD extensions (SSE)
 - Birth of industry standards for parallelization: Message Parsing Interface (MPI) and OpenMultiProcessing (OpenMP)
- 2000's:
 - Power wall problem: can't keep increasing clock speed!
 - Industry answer: multicore processors
IBM POWER4 (2001); Intel Pentium D, AMD Athlon X2 (2004)
- 2010's:
 - Less performance improvements on newer CPU generations
 - Rise of the smartphone



Hardware evolution: accelerators

- 1990's:
 - Video game industry boom
 - 3D graphics accelerator cards enter the market
 - Rising popularity of Graphical User Interfaces (GUIs)
- 2000's:
 - NVIDIA introduces Compute Unified Device Architecture (CUDA)
 - Khronos Group introduces OpenCL
- 2010's:
 - Open ACCelerators (OpenACC) introduced by industry consortium (NVIDIA, Cray, CAPS, PGI)
 - AMD introduces Radeon Open Compute (ROCm) platform
 - GPUs become the *de facto* standard for accelerators



Hardware evolution: what about today?

- 2020's:
 - Resurgence of alternative system architectures (ARM64, RISC-V, ...)
 - Creative 3D transistor design (FinFET, HBM, ...)
 - Quantum effects at nanoscale (tunneling)
 - Can't keep miniaturizing components!
 - Intel wafer yield problems with 5 nm process
 - Silicon chip supply chain disruption and shortage
 - Innovations in machine learning drive hardware demand

Systems ranked #1 on Top500.org, 2008-2021

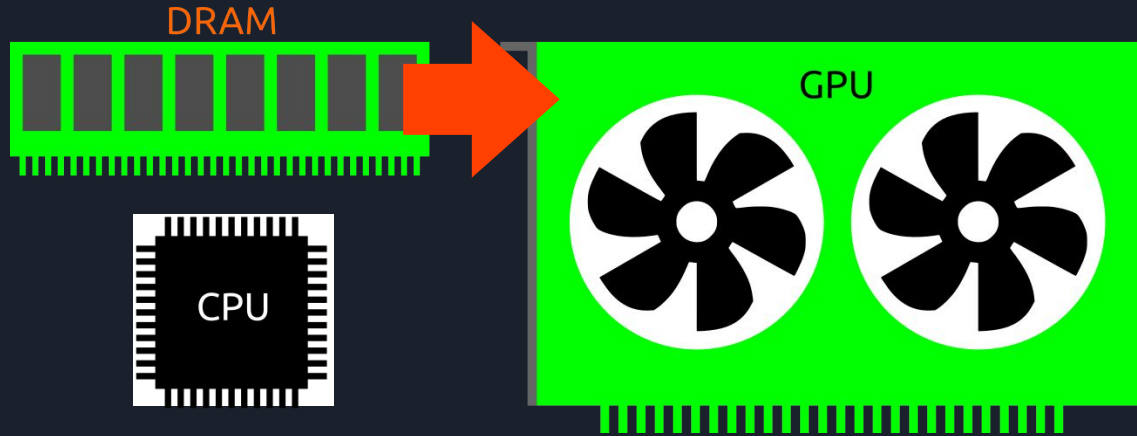
Time	System	Location	CPU	Accelerator
6/08 - 6/09	Roadrunner	LANL (NM, USA)	AMD Santa Rosa (2-core)	IBM PowerXCell
11/09 - 6/10	Jaguar	ORNL (TN, USA)	AMD Istanbul (6-core)	-
11/10	Tianhe-1A	China	Intel Westmere-EP (6-core)	NVIDIA Fermi
6/11 - 11/11	K computer	Japan	Fujitsu SPARC64 (8-core)	-
6/12	Sequoia	LLNL (CA, USA)	IBM BlueGene/Q (16-core)	-
11/12	Titan	ORNL (TN, USA)	AMD Interlagos (16-core)	NVIDIA Kepler
6/13 - 11/15	Tianhe-2	China	Intel Ivy Bridge (12-core)	Intel Knights Corner
6/16 - 11/17	TaihuLight	China	Sunway (260-core)	-
6/18 - 11/19	Summit	ORNL (TN, USA)	2× IBM POWER9 (42-core)	6× NVIDIA Volta
6/20 - 6/21	Fugaku	Japan	Fujitsu ARM64 (48-core)	-



Programming CPUs vs accelerators

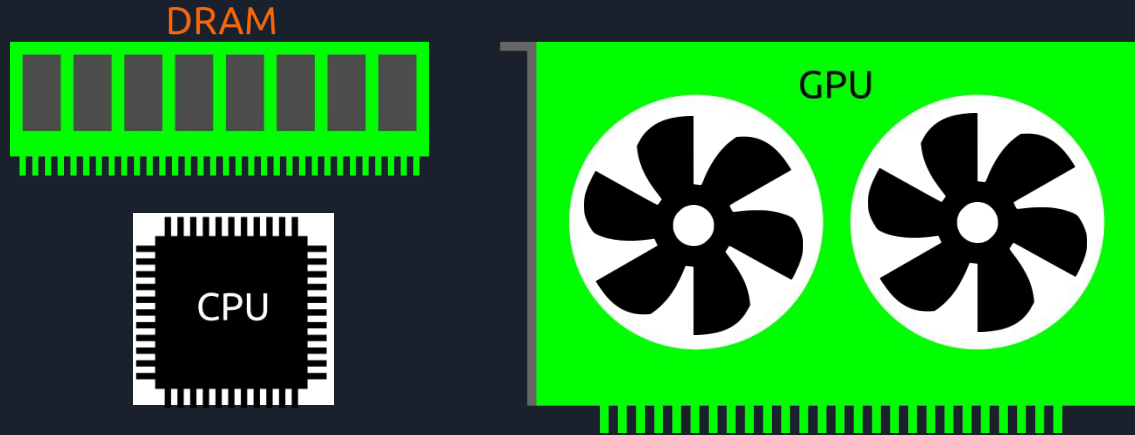
CPUs	Accelerators
“Heavy” cores	“Light” cores
General purpose	Specialized
Complex instruction set	Simpler instruction set
Tens of cores	Thousands of cores
Mature compiler infrastructure	Evolving compiler infrastructure
Relatively easy to program	Somewhat harder to program
“Quality”	“Quantity”

The 3-step paradigm of accelerated computing



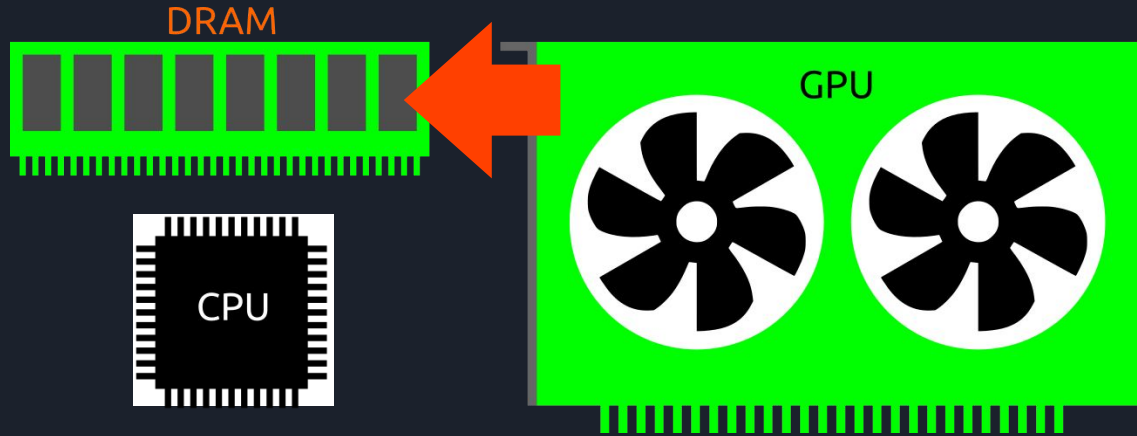
1. Transfer data to device

The 3-step paradigm of accelerated computing



2. Compute on device

The 3-step paradigm of accelerated computing



3. Transfer results to host



Programming GPUs: frameworks

- Vendor-provided programming models and libraries
 - NVIDIA: CUDA (<https://developer.nvidia.com/cuda-toolkit>)
 - AMD: HIP, ROCm (<https://developer.amd.com/resources/rocm-learning-center/>)
 - Intel: oneAPI, DPC++ (<https://software.intel.com/content/www/us/en/develop/tools/oneapi.html>)
- Industry consortium
 - OpenCL (<https://www.khronos.org/opencl/>)
 - OpenACC (<https://www.openacc.org/>)
 - OpenMP **target offload** (<https://www.openmp.org/>)
- Emerging research software
 - Kokkos (Sandia, <https://kokkos.org/>)
 - RAJA (LLNL, <https://github.com/LLNL/RAJA>)

Programming GPUs: considerations



- Programming language support (esp. Fortran, Python)
- Interoperability between different programming models
- Portability vs hardware-specific performance
- Compiler/toolchain availability and compatibility
- Learning curve and user friendliness/ease of use
- Documentation, support groups, and user community

Towards exascale



Image credit: OLCF

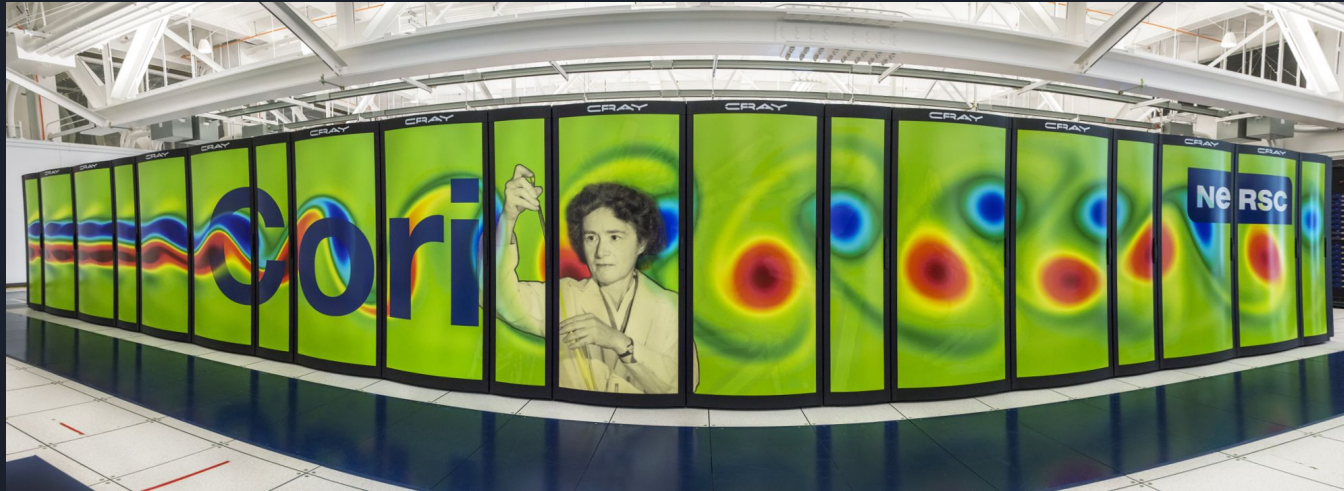


Exascale computing?

- Exa- = 10^{18} = quintillion, measured in 64-bit (double precision) floating point operations per second (FLOP/s)
- Fugaku (RIKEN, Kobe, Japan): Fujitsu A64FX CPU, 442 petaFLOP/s
- Collaboration of Oak Ridge, Argonne and Livermore (CORAL):
 - Summit (ORNL): IBM POWER9 CPU + NVIDIA Volta GPU, 148 petaFLOP/s
 - Aurora (ANL): Intel Sapphire Rapids CPU + Intel Ponte Vecchio GPU, currently under deployment
 - Sierra (LLNL): IBM POWER9 CPU + NVIDIA Volta GPU, 94 petaFLOP/s
- Upcoming systems
 - Frontier (ORNL): AMD CPU + AMD Aldebaran GPU, currently under deployment
 - El Capitan (LLNL): AMD CPU + AMD GPU

What about **NERSC** ?

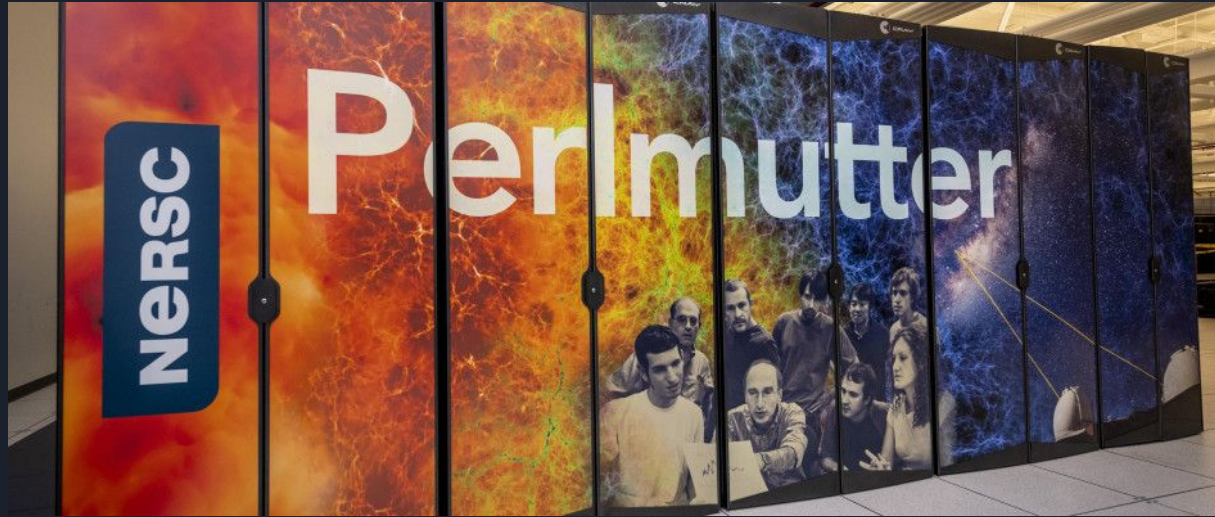
The Cori supercomputer



- #30 on Top500.org (6/2021), 14 petaFLOP/s LINPACK benchmark
- Named after American biochemist & Nobel laureate Gerty Cori
- 3 partitions: **haswell** (Intel Haswell CPU), **kn1** (Intel Knights Landing MIC), **gpu** (Intel Skylake CPU + NVIDIA Volta GPU)

What about **NERSC**?

The Perlmutter supercomputer



- #5 on Top500.org (6/2021), 65 petaFLOP/s LINPACK benchmark
- Named after American astrophysicist & Nobel laureate Saul Perlmutter
- AMD Milan CPU + NVIDIA Ampere GPU
- First phase deployment complete, currently testing with limited number of users



Moving forward

- Scientific users seem to show reluctance / relatively slow adoption of accelerated supercomputing
 - NSF systems still *mostly* CPU-only
 - On Cori, **haswell** queue *significantly* more crowded than **kn1** or **gpu**
 - “If it works, don’t break it”
 - High cost of software development / porting
 - Limited developer resources
- Hardware vendors keep pushing the latest and greatest tech
 - Competing vendors, each with different solutions
 - High complexity in software/toolchain deployment
 - Evolving infrastructure means things break often
- **There is a large gap between available hardware and user software!**



Proposed strategies for Perlmutter

- Hold office hours for easing transition to AMD Milan and NVIDIA Ampere
- Write extensive, easy-to-follow documentation and tutorials
- Provide userland utility scripts for ease of use (e.g. update the job script generator)
- Provide robust environment for cross-compiling applications on login node
- Provide hardware resources for code development
- Host training and tutorial re-runs; GPU bootcamp for new users
- Work closely with other leadership computing facilities for interoperability purposes
- Use stricter **cgroups** configuration / “patrol” login nodes for abuse

Discussion: Questions & Answers