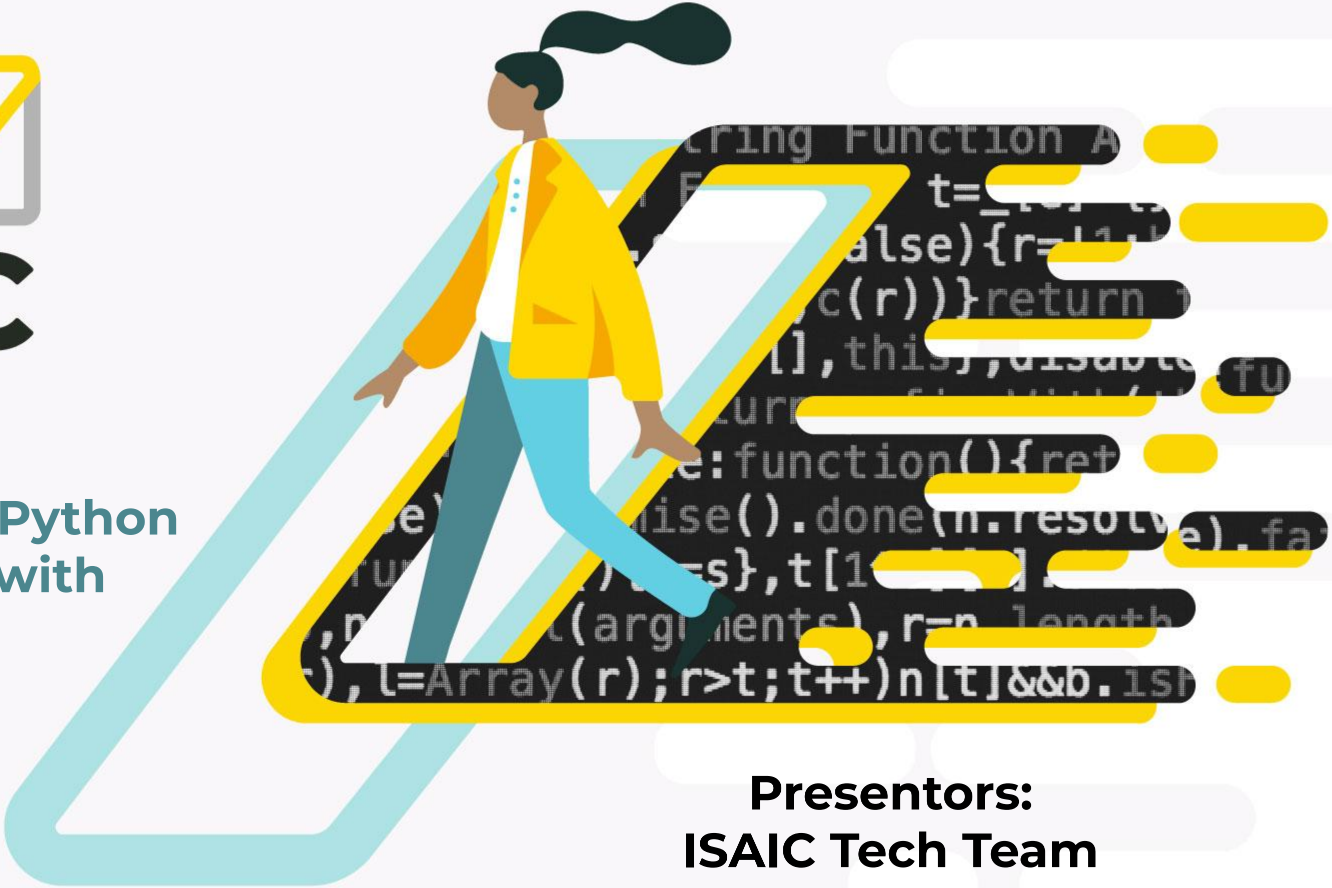
ISAIC

Industry Sandbox
& AI Computing

GPU-Accelerated Python Programming with Numba



Presentors:
ISAIC Tech Team

DATE:

Today’s Discussion

- Who we are and more about ISAIC?
- CPU vs. GPU computations
- GPU and Parallel Computing
- Introduction to Numba
- Tutorial

ISAIC is powering the A.I.mbition in Western Canada

- **Small to medium size start ups**
- **Accelerate AI adoption and commercialization**
- **By abstracting away hardware management**

We offer High-performance Computing Virtual Machines

- **At ISAIC, we offer different flavours of high-performance VMs that come preconfigured and specifically tailored to their needs**
- **Our services come with 1 to 8 GPUs and up to 64 CPU cores with 512GB RAM**
- **Our offerings come ready with AI tools including newest libraries from TensorFlow, Torch, & Keras**
- **We offer in-person expert consultation to our clients and help them through their AI journey**

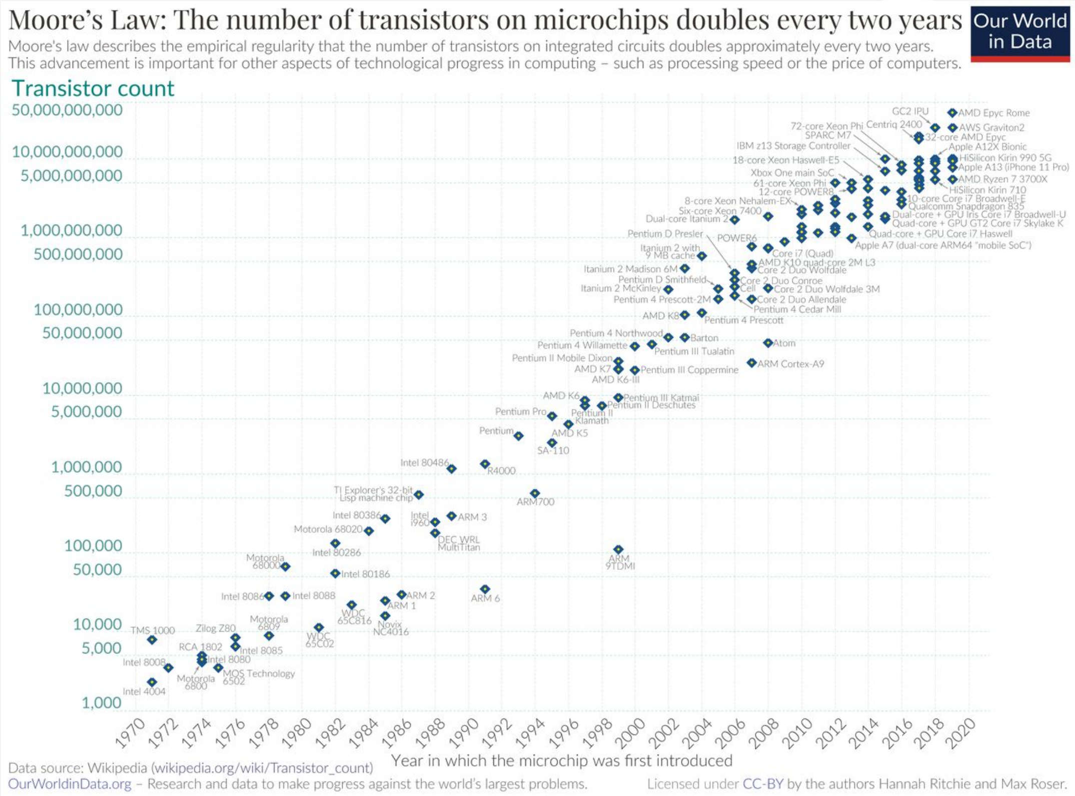
Today, we will see how ISAIC creates and uses VMs for our clients

- **What is a Virtual Machine?**
  - **Through virtualization we can divide existing hardware resources into multiple machines and create virtual hardware that our Operating Systems run on .....**

- **Let’s set aside the technical terminology and definition and take a look at virtualization from an operational point of view**

Why do we need GPUs?

- **Transistors today are of size ~10nm**
- **In recent years, shrinkage of transistor has slowed down**
- **Moore's law hitting the physics limitation**
- **However we need higher computation power than ever before to process exponentially growing data**
- **GPUs come to the rescue by using the distributed parallel computing**



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

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.

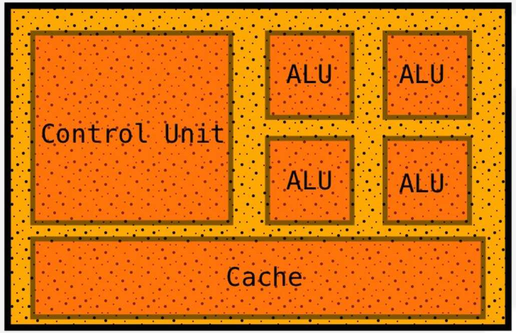
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

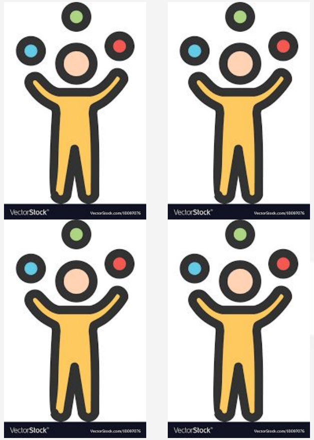
OurWorldinData.org – Research and data to make progress against the world's largest problems.

CPU Vs. GPU

- **Serial vs. parallel computing**
- **Several cores vs. thousands of cores**
- **Fast and versatile vs. high throughput**
- **System memory vs. graphics card memory**
- **Cache memory management**

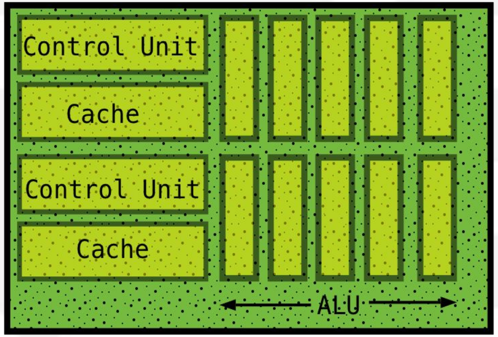
CPU

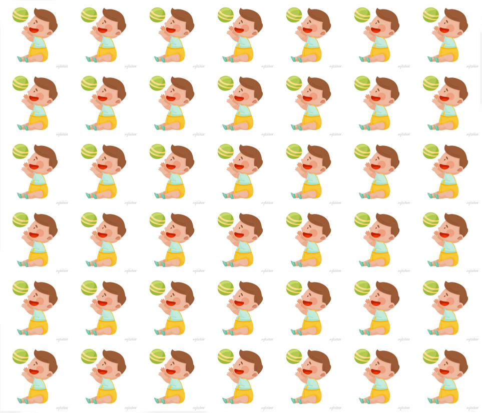




https://www.vectorstock.com/royalty-free-vector/juggling-balls-vector-10097076

GPU

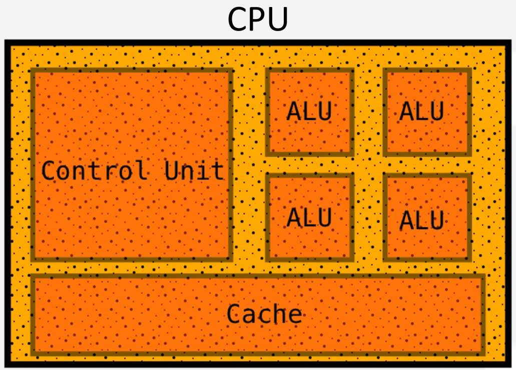


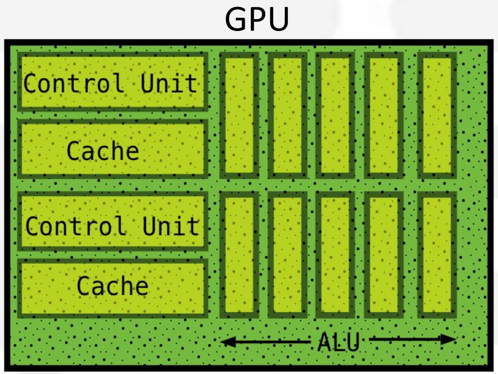


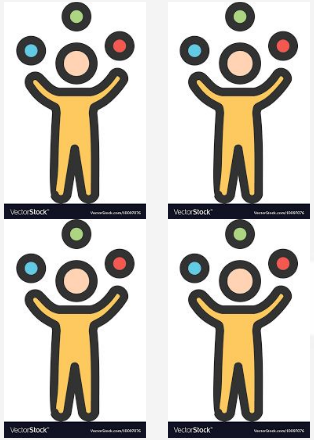
https://myloview.com/sticker-baby-boy-sitting-on-the-floor-and-throwing-ball-in-the-air-vector-no-C7D1DC2

Advantages of GPU over CPU

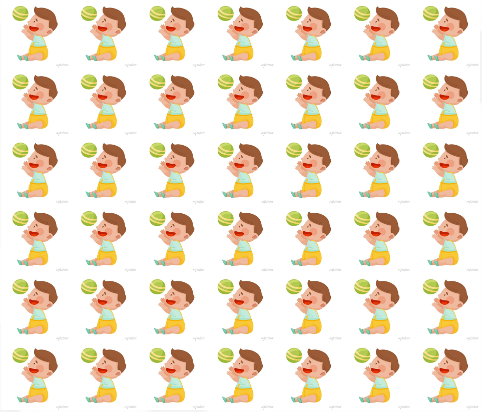
- **Memory Bandwidth**
- **Large Datasets**
- **Parallelism**
- **Cost Effective**







https://www.vectorstock.com/royalty-free-vector/juggling-balls-vector-10097076

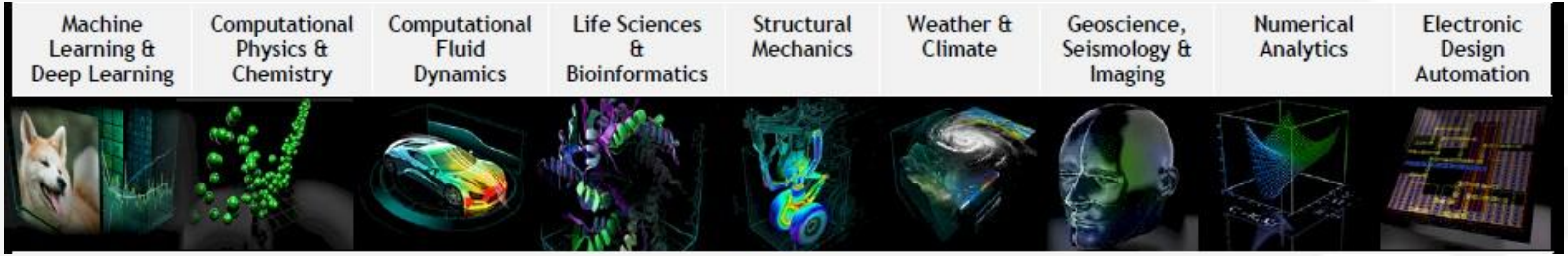


https://myloview.com/sticker-baby-boy-sitting-on-the-floor-and-throwing-ball-in-the-air-vector-no-C7D1DC2

GPU at a Glance

- **First developed for the purpose of graphics, 3D image/video rendering**
- **Used as special purpose graphics unit mainly for game developments**
- **Quickly transformed into general purpose computing device and sparked AI boom, became integrated part of modern supercomputers**
- **GPU is complementary to CPU, NOT a replacement**

https://inspirationfeed.com/game-worlds/


Machine Learning & Deep Learning
Computational Physics & Chemistry
Computational Fluid Dynamics
Life Sciences & Bioinformatics
Structural Mechanics
Weather & Climate
Geoscience, Seismology & Imaging
Numerical Analytics
Electronic Design Automation

https://digital.hbs.edu/platform-digit/submission/nvidias-winning-platform-strategy-with-cuda/

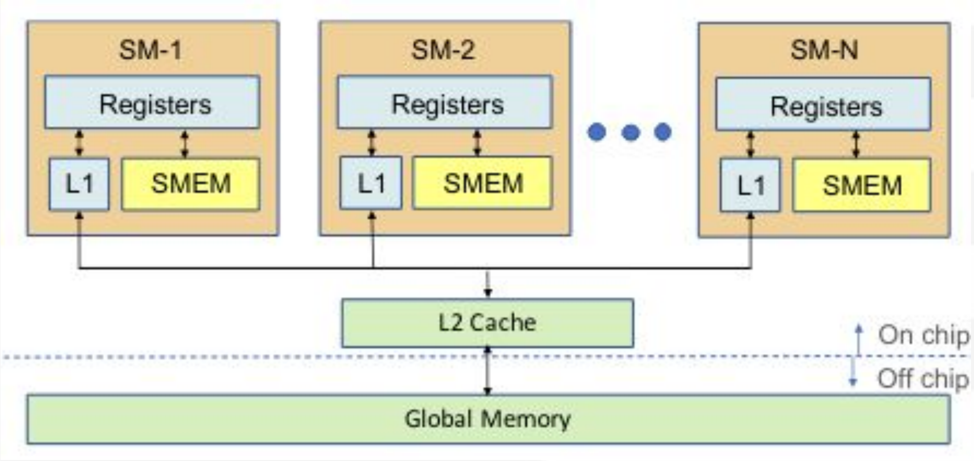
Diving Deep into GPU Architecture

- **Compute units called Streaming Multiprocessors (SMs)**

- **Each SM has a number of cores, registers and shared memory (SMEM) and L1 cache local to each SM**

- **Shared memory utilization is programmable**

- **Data flow: SM ->L1 cache -> L2 cache -> global memory**



https://arxiv.org/pdf/2003.01178.pdf

Diving Deep into GPU Processing

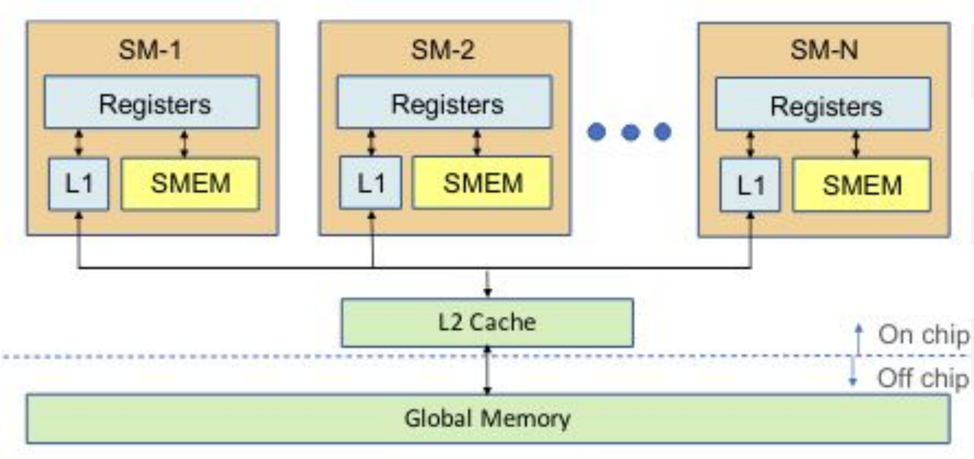
- GPU processes large number of threads (tasks) organized into `thread blocks`

- Each `thread block` is run by one SM

- Thread blocks are further divided into subgroups of threads called warps (usually consisting of 32 threads)

- Threads of a warp execute a *Single Instruction Multiple Threads* (SIMT) model

- SIMT model is the execution mechanism that gives GPU the power of parallel computing



https://arxiv.org/pdf/2003.01178.pdf

APIs to implement SIMT: Introduction to CUDA

- CUDA is a software platform that implements parallel computing

- Developed by NVIDIA with first release in 2007

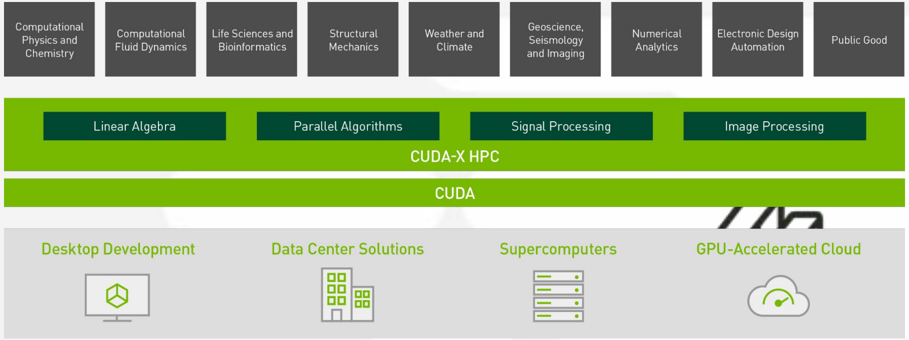
- Designed to work with programming language C, C++, FORTRAN

- Like CUDA, there are other parallel programming models e.g. ROCm, OpenCL

- These APIs and libraries built on top of them provide GPU the massive power to build high performance computing (HPC) and cutting edge AI solutions

- In our tutorial, we will learn how to convert traditional python code into GPU-accelerated code that runs on CUDA supported devices

https://en.wikipedia.org/wiki/CUDA



https://www.nvidia.com/en-us/technologies/cuda-x/

Accelerated Python Programming: Introduction to Numba

- **Numba is a “just-in-time” (jit) compiler for python**
- **Works best on code that uses NumPy, functions and loops**
- **Requires very little modification to existing python code**
- **Optimizes python functions for both CPU and CUDA based GPU**
- **pyCUDA is an alternative option for Numba, but requires writing C code in python and a lot of code modification**
- **In our tutorial, we will first learn how to use Numba to compile python functions for CPU, then switch to GPU acceleration using Numba compiler**