

# Kokkos/C++ training

Measuring memory bandwidth

Pierre Kestener

December 9-11th, 2024, Cerfacs, Toulouse



Funded by  
the European Union



CERFACS

CCFR CENTRE  
DE COMPÉTENCE  
HPC.HPDA.IA

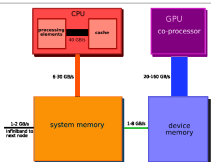


- ▶ There are two main metrics for measuring numerical computing power of an algorithm implementation :
  - ▶ **FLOPS** : Floating point Operations per seconds
  - ▶ **BW** : memory bandwidth (amount of Read and Write operations, to/from main memory) in Gbytes per seconds.
- ▶ **Memory bandwidth** is often considered as more important, because it is often the main bottleneck ; or at least it drives the software optimization refactoring for performance optimization
- ▶ What we want to do here is twofold :
  - ▶ How to determine the maximum/peak **memory bandwidth** of a given hardware architecture ?
  - ▶ Use a very simple example (saxpy) to **measure bandwidth** on a CPU system and a GPU system.

Example system (2022) :

- ▶ **CPU-GPU link**, Pci-express bus x16, Gen4 :  
 $BW = 16 * 2 * 2\text{GBytes/s} = \mathbf{64\text{ GBytes/s}}$
- ▶ **CPU-local** : DDR memory, 64-bit memory bus  
 $BW = 3200\text{ (M transferts/cycle)} * 8\text{ (channel)} * 8\text{ (bytes)} = \mathbf{205\text{ GBytes/s}}$
- ▶ **GPU-local (Nvidia A100)** : 5120-bit memory bus,  $\text{DDR}@f = 1215\text{MHz}$ ,  
 $BW = 2 * 1215\text{ (M transferts/s)} * 5120/8\text{ (bytes/transfert)} = \mathbf{1555\text{ GBytes/s}}$
- ▶ GPU memory bandwidth is often  $\times 5$  to  $\times 8$  faster than CPU memory bandwidth
- ▶  $\Rightarrow$  a given software application, memory bound, well optimized on CPU then ported to GPU should run faster on GPU by factor of  $\times 5$  to  $\times 8$

Bandwidth in a CPU-GPU System



How to determine the **peak** hardware memory bandwidth of your compute platform ?

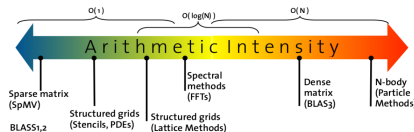
- ▶ **Multicore CPU** (e.g. Intel Skylake) :
  - ▶ Memory type ? e.g. DDR4-2666
  - ▶ Number of channels ? e.g. 6
  - ▶ Max  $BW = \# \text{NbOfChannel} \times \text{Frequency(GHz)} \times \text{BusWidth}/8 \text{ (Bytes)} \times \# \text{NbOfSockets}$
  - ▶ e.g. on TGCC/IRENE,  $BW = 6 \times 2.6 \times 64/8 \times 2 = 256 \text{ GBytes per node}$
- ▶ **Manycore CPU** (e.g. Intel KNL) :
  - ▶ depends on HBM configuration (CACHE, FLAT, HYBRID)
  - ▶ e.g. KNL on TGCC/IRENE configured in CACHE mode,  $BW \geq 400 \text{ GBytes/s}$
- ▶ **NVIDIA GPU** (e.g. Pascal P100) :
  - ▶ Use CUDA SDK deviceQuery to retrieve hardware spec (**TODO as an exercise**).
  - ▶ # Memory Clock rate : 715 Mhz
  - ▶ # Memory Bus Width : 4096-bit
  - ▶  $BW = 732.1 \text{ Gbytes/s}$
- ▶ **NVIDIA GPU** (e.g. Pascal V100) :
  - ▶  $BW = 898.0 \text{ Gbytes/s}$



- ▶ As an exercise, we will measure bandwidth on a computing node of `kraken`
- ▶ just follow the instructions in the readme
- ▶ let see how performant the `saxpy` implementation can be on CPU and on the GPU.

**Goal** : making sense of performance measurement, assessing the need for refactoring for optimization

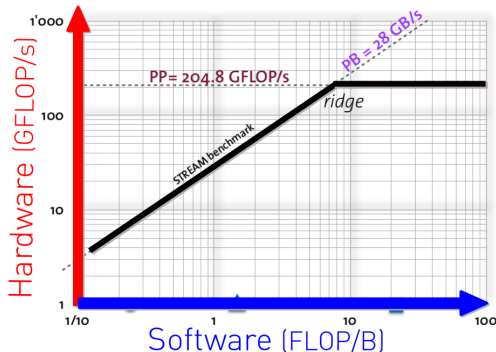
- ▶ an increasingly large diversity of architectures
- ▶ software challenges to use new architectures :
  - ▶ **refactoring** (when ? where ?) ; avoid sub-optimal use of hardware/software
  - ▶ **optimization strategies**
- ▶ **Roofline model** : a simple way of characterizing hardware performances  
Each algorithm implementation is characterized by
  - ▶ **arithmetic intensity (FLOPS/Bytes)** : number of FLOP per bytes read/write from external memory
  - ▶ **effective memory bandwidth (GB/s)** : data moved to/from external RAM



reference : [http://www.nvidiacodesignlab.ethz.ch/news/CoDesignLabWorkshop2013\\_Rossinelli\\_Roofline.pdf](http://www.nvidiacodesignlab.ethz.ch/news/CoDesignLabWorkshop2013_Rossinelli_Roofline.pdf)

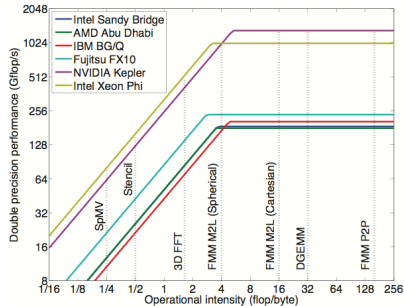
Goal : making sense of performance measurement, assessing the need for refactoring for optimization

## The roofline model



- ▶ it visually relates hardware with software
- ▶ Performance =  $\min(\text{Peak Bandwidth} * \text{Arith Intensity}, \text{Peak Flops})$

reference : [http://www.nvidiacodesignlab.ethz.ch/news/CoDesignLabWorkshop2013\\_Rossinelli\\_Roofline.pdf](http://www.nvidiacodesignlab.ethz.ch/news/CoDesignLabWorkshop2013_Rossinelli_Roofline.pdf)



- Understand inherent hardware limitations
- Show priority of optimization

references :

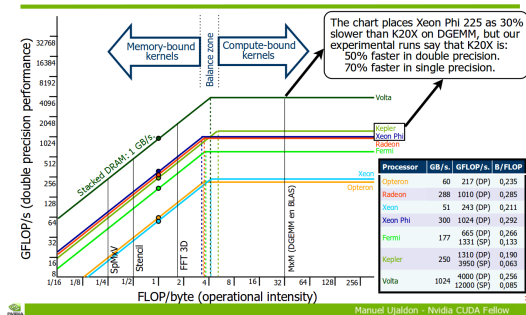
<http://lorenabarba.com/news/fast-multipole-method-in-the-exascale-era/>

<http://icpp2013.ens-lyon.fr/GPUs-ICPP.pdf>

Roofline model, HLRS Training, Nodel-Level Performance Engineering, June 2023



## The Roofline model: Hardware vs. Software



- Understand inherent hardware limitations
- Show priority of optimization

references :

<http://lorenabarba.com/news/fast-multipole-method-in-the-exascale-era/>

<http://icpp2013.ens-lyon.fr/GPUs-ICPP.pdf>

Roofline model, HLRS Training, Nodel-Level Performance Engineering, June 2023