Data-aware algorithms for matrix and tensor computations

Suraj Kumar

Inria & ENS Lyon
Email:*suraj.kumar@inria.fr*

CR12: September 2023

https://surakuma.github.io/courses/daamtc.html

Table of Contents

Course details

Topics:

- Communication costs on sequential and parallel machines
- Matrix computations
- Tensor computations
- Popular ways to work with large matrix and tensor data

Class hours: Tuesday 08:00 AM + Thursday 03:45 PM

Location: Room B1

Contact: suraj.kumar@inria.fr

Course details

Evaluation:

- 5 homework assignments (best 3 would be counted) (40%)
- Project (60%): 1/2 students, paper(s) reading + report + presentation

Website:

- `https://surakuma.github.io/courses/daamtc.html`
- Course material available online (right after class)
- Link to assignments and other resources

General stuff:

- Stop me with questions, ask for re-explanation
- Will have catch-up sessions after each 2 weeks
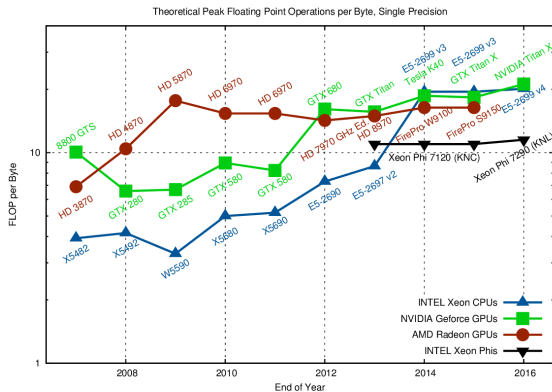- Feel free to email your doubts

Table of Contents

Motivation

- (Fast) Memory: place to store data to compute

- Always been a limited resource (4KB in Appollo 11 computer)

- 64KB L1 cache in recent iphone

- Problem size is always getting bigger ...

- Annual improvements:
  - Time per flop (computation): 59%
  - Data movement (communication):

	Bandwidth	Latency
Network	26%	15%
DRAM	23%	5%

Data from *Getting up to speed: The future of supercomputing*, 2005, National
Academies Press (2004 figure based on data on the period 1988-2002)

Flop per byte moved ratio

Theoretical Peak Floating Point Operations per Byte, Single Precision

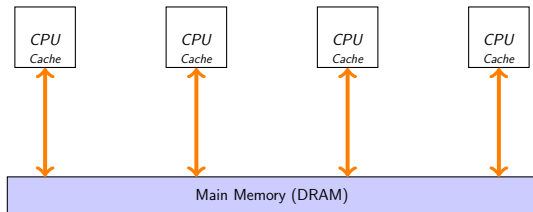From http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

- Number of flops performed in the time needed to move a byte
- $\dfrac{\text{computing speed}}{\text{communication speed}}$
- Avoid communication to save time (and energy)

Simplified model for a sequential CPU core
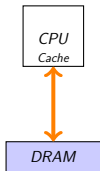

Cache hierarchy model

Simplified model
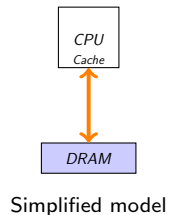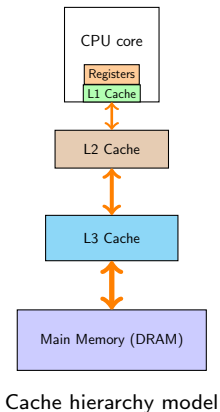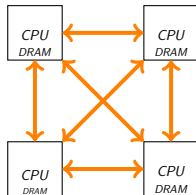
Shared memory machine model

Shared memory parallel machine model

- We can view all CPUs as a single large processing unit

Data transfer models in this course

Sequential machine

Distributed memory homogeneous machine

- α: per-message latency cost, β: per-word bandwidth cost, γ: per-operation arithmetic cost
- An algorithm's total cost $= \alpha \cdot$ (#messages) $+ \beta \cdot$ (#data transfers) $+ \gamma \cdot$ (#computations)
- For a parallel machine, we consider cost along the critical path
- Concentrate mainly on bandwidth costs throughout the course

Sometimes computations can be overlapped with data transfers. However to keep the models simple we will not take that into account through out the course.
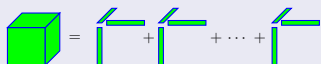
Tensors and their uses

Vector   Matrix   3-dimensional tensor   4-dimensional tensor

- **Neuroscience**: Neuron $\times$ Time $\times$ Trial
- **Media**: User x Movie x Time
- **Ecommerce**: User x Product x Time
- **Social**-**Network**: Person x Person x Time x Type

Tensors and their uses

- High dimensional tensors: Neural network, Molecular simulation, Quantum computing

- People work with low dimensional structure (decomposition) of the tensors