CRAY EMEA RESEARCH LAB

EXPERTISE

EXPERTISE kickoff training Memory Hierarchy

Adrian Tate Director, Cray EMEA Research Lab

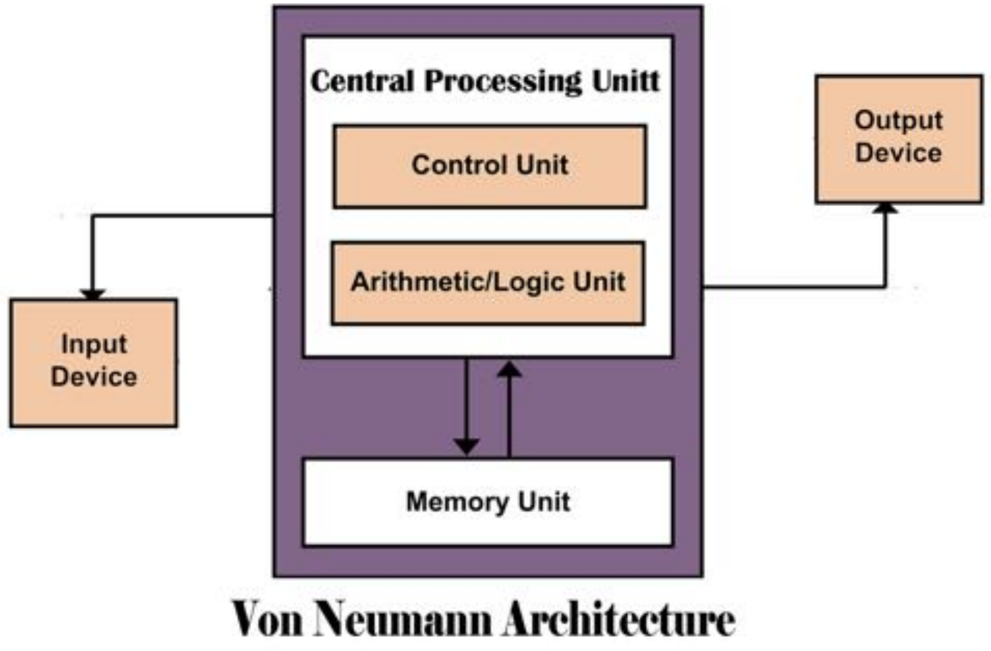
This talk

1. Hardware trends and design constraints
2. Traditional memory hierarchy
3. Future memory hierarchy

- Whirlwind tour of memory hierarchies and future memories
- No detail, but pointers to further reading for the HPC students
- slides will be made available

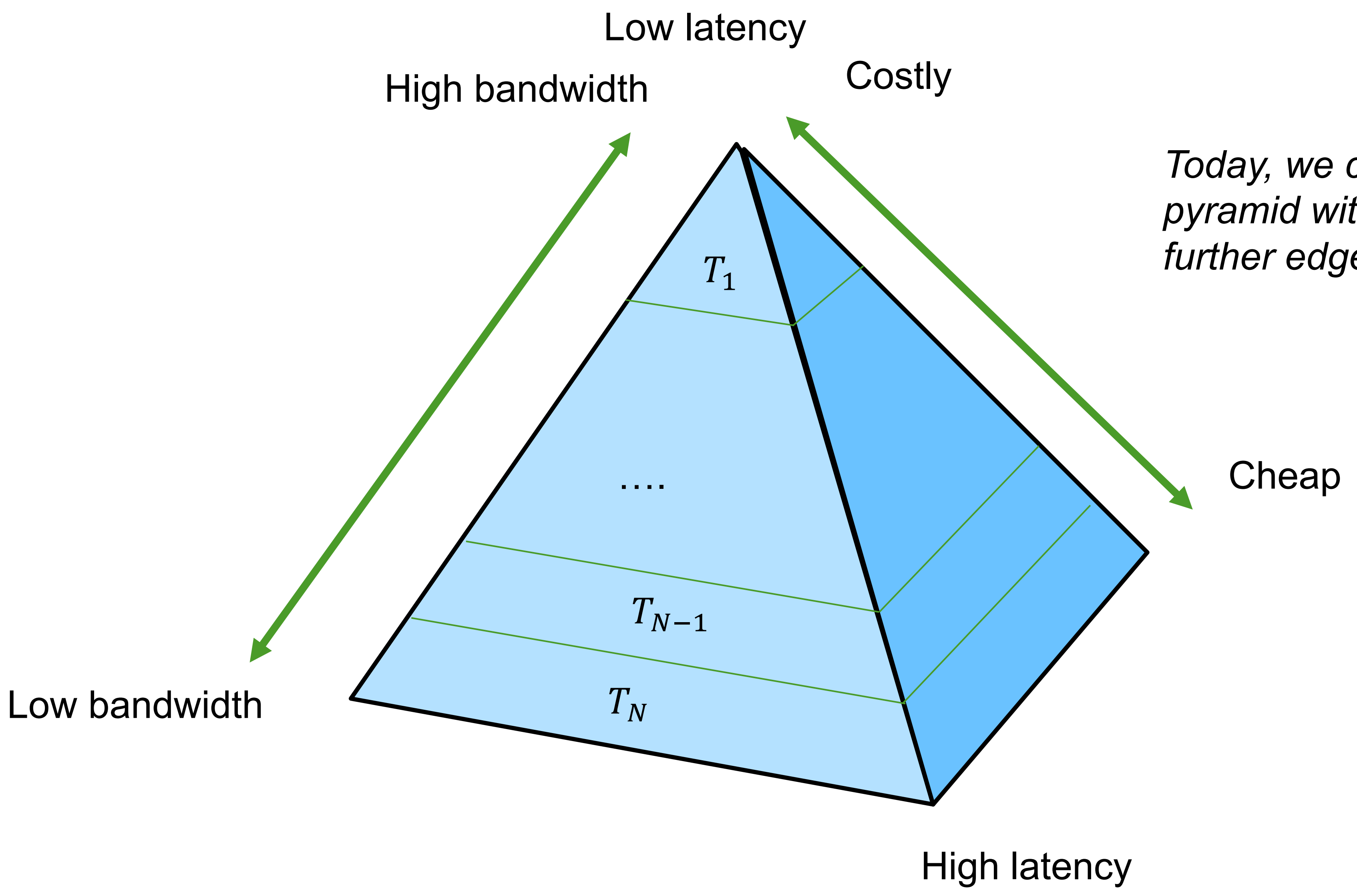
Why do we need to do something with memory?

- Memory is boring
- Why not just a single “flat” memory system?
- Fundamental issue with chosen model of computation
- Von Neumann Architecture (1945)



The von Neumann bottleneck :
computer system throughput is limited due to the relative ability of processors compared to top rates of data transfer

What is a memory hierarchy?



Today, we could also use a pentagonal pyramid with power consumption as a further edge/consideration

Von Neumann bottleneck

- Avoiding the Von Neumann bottleneck has historically been easy
- One or two-level cached memory
- Cache : A higher tier in the pyramid.
  - Fast (low latency, high bandwidth)
  - Expensive
  - Used often
- In modern times, this is not the case.
- Let's look at some trends and physics constraints in computer design

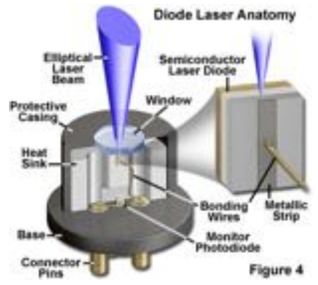
A supercomputer as scientific instrument

microscopes

Scanning Electron

CT Scan

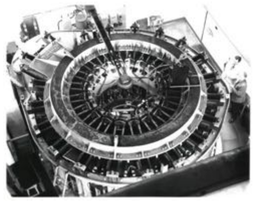
Scanning tunnelling



Laser diode

synchrotron

DESY


Glasgow

SRS

ESRF

DLS

supercomputers

CDC6600

CRAY-1

NEC SX

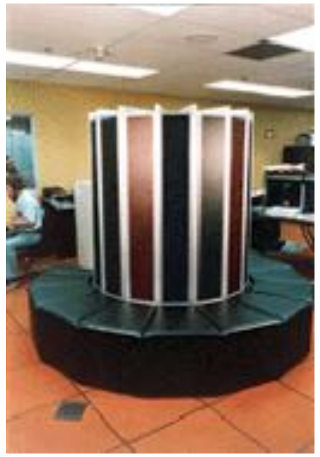
ASCI RED

Earth Simulator

Titan

1960 *Fortran* 1970 *BLAS* 1980 *LAPACK* 1990 *MPI* *openMP* 2000 *C++* *Python* 2010

Then and now(-ish)



System Performance	160 MFLOPS	27 PFLOPS	1054687 x
Perf/node	160 MFLOPS	1.4 TFLOPS	8750 x
Memory capacity /node	8 MB	37.5 GB	4687 x
Memory bandwidth /node	640 MB/s	275 GB/s	430 x
Memory bandwidth / flop	4	0.196	1/20 x

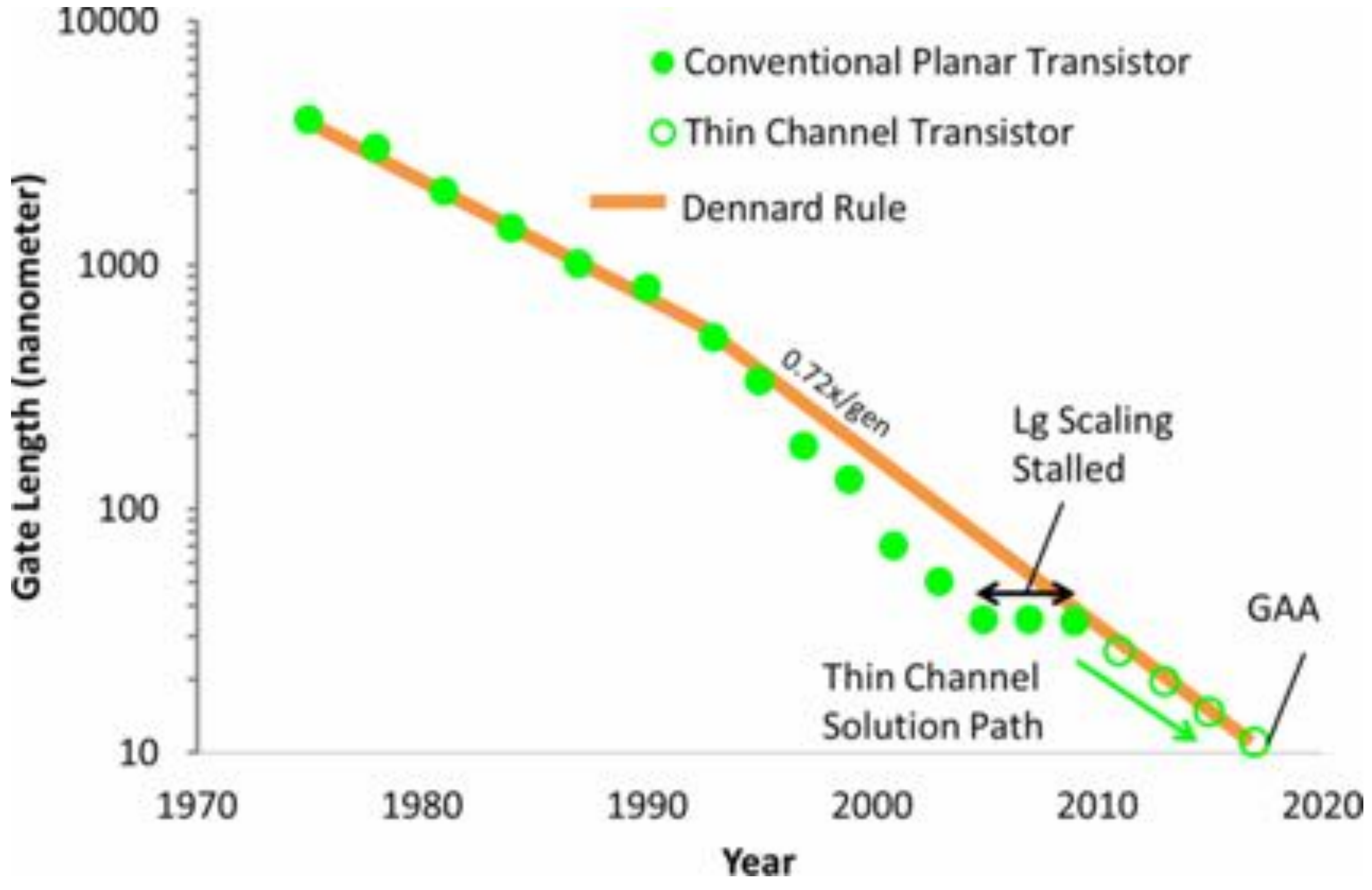
- **Is the last point an anomaly?**

Laws (trends)

- **Moore's Law : transistor count double roughly every 18 months**
  - Remains true but the performance associated with it has tailed off

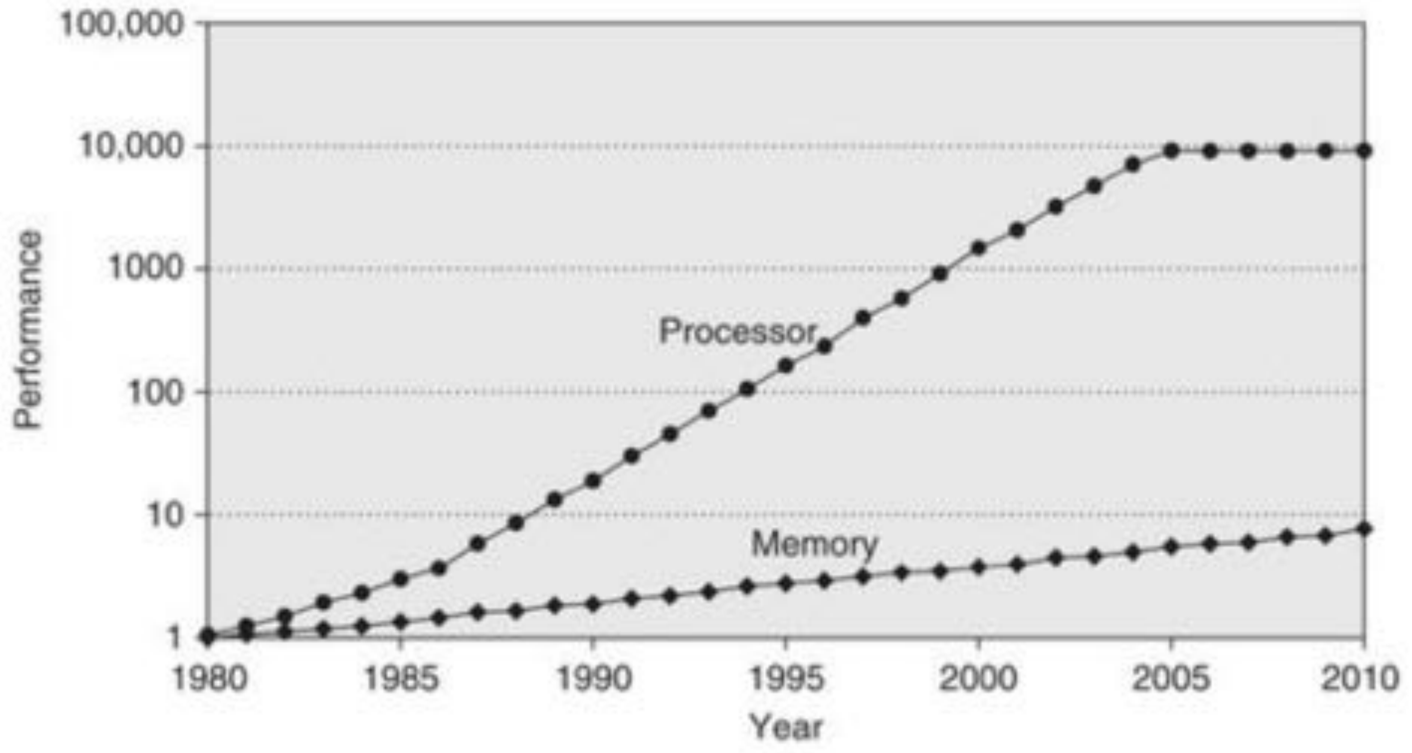
- **Dennard Scaling :** as transistors get smaller their power density stays constant, so that the power use stays in proportion with area
  - Ended in 2006
  - Response – multicore processors

Dennard Scaling



Memory vs CPU Speed trend

- **Processors are getting wider, not faster**
  - The bandwidth to memory has not widened at the same rate
  - In fact, this trend has been visible for decades :



Trends summarised (4 “walls”)

- **memory wall**. The processor asks for data much faster than memory can give it to the processor
- **ILP wall**: We've squeezed out all the parallelism at this level
- **power wall**: Power requirements increase exponentially with processor speed. Beyond a certain point, speed improvements not practical.
- **Thermal wall** : Processors can no longer sink the heat generated during computation

Thoughts

- **So the question often asked**
  “why are CPUs not getting any faster”
  is actually irrelevant

- **Although chips are not getting faster, they are still *way* too fast for the memory system to cope with**
  - Recall the bytes/flop issue we unearthed on slide 1

- **Additionally, we have two modern constraints**
  - Power concerns
  - Explosion of data (covered elsewhere)

- **(Much) faster Data processing is what is needed**
  - This summarises “Exascale Computing”
  - Innovation in memory hierarchy is the response to this

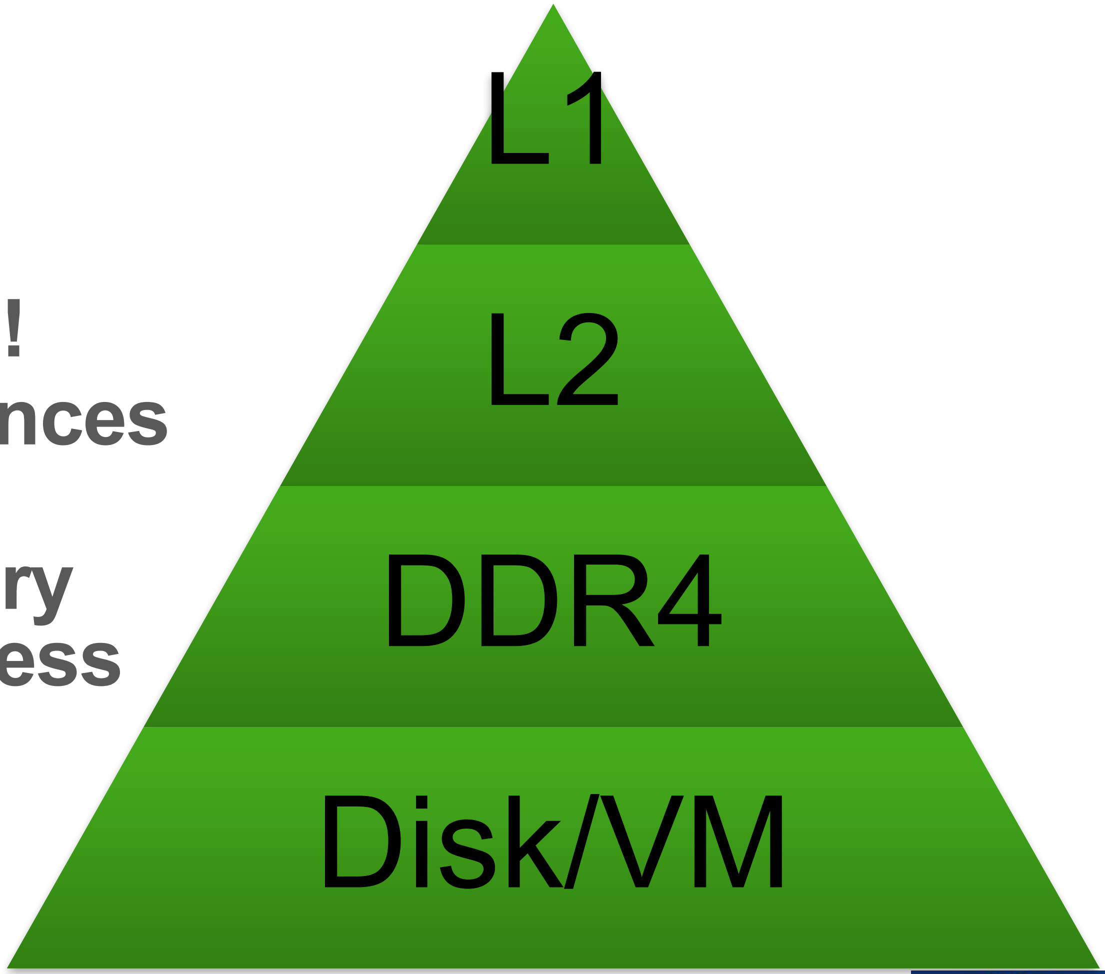
Further Reading

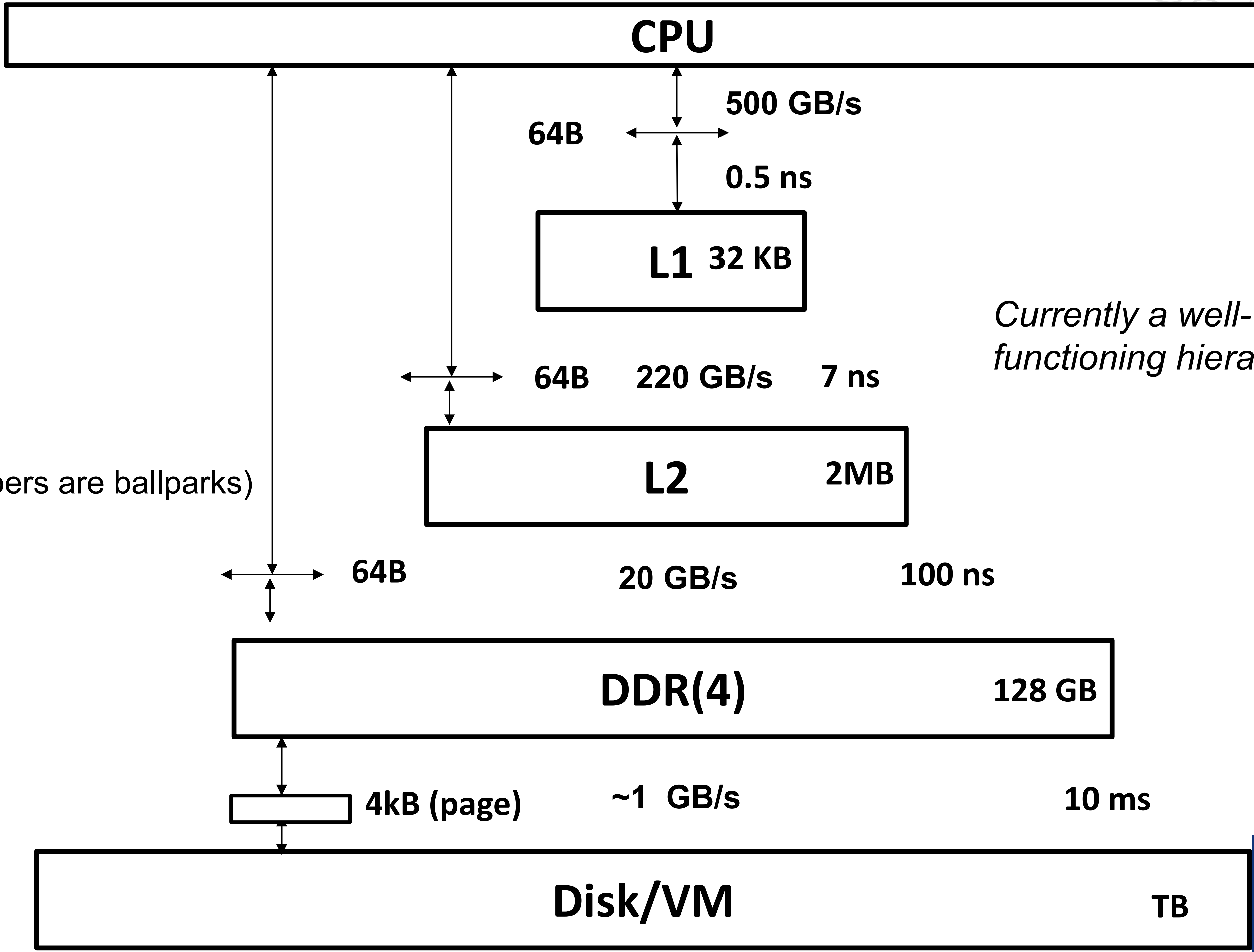
- **History of Computing**
  - ***Turing's Cathedral***, George Dyson (book)
- **Trends**
  - ***Computer Architecture: A Quantitative Approach (5th Addition),*** David Patterson and John Hennessy
    - Gets out of date quickly, so Google for modern trends
  - Georgia Tech Memory Hierarchy course: [ECE 3056 course notes]
- **Cache and memory design**
  - ***Computer Architecture: A Quantitative Approach (5th Addition),*** David Patterson and John Hennessy
- **Data-intensive science**
  - ***Synergistic challenges in data-intensive science and Exascale Computing*** https://science.energy.gov/~/media/40749FD92B58438594256267425C4AD1.ashx

EXPERTISE

Memory hierarchy (traditional)

- **Want memory with**
  - Fast access
  - Large capacity
  - Inexpensive
- **Incompatible requirements!**
- **Fortunately memory references are not random!**
- **Simulate the perfect memory by understanding data access**
- **Exploit “Locality”**
  - Temporal locality
  - Spatial locality

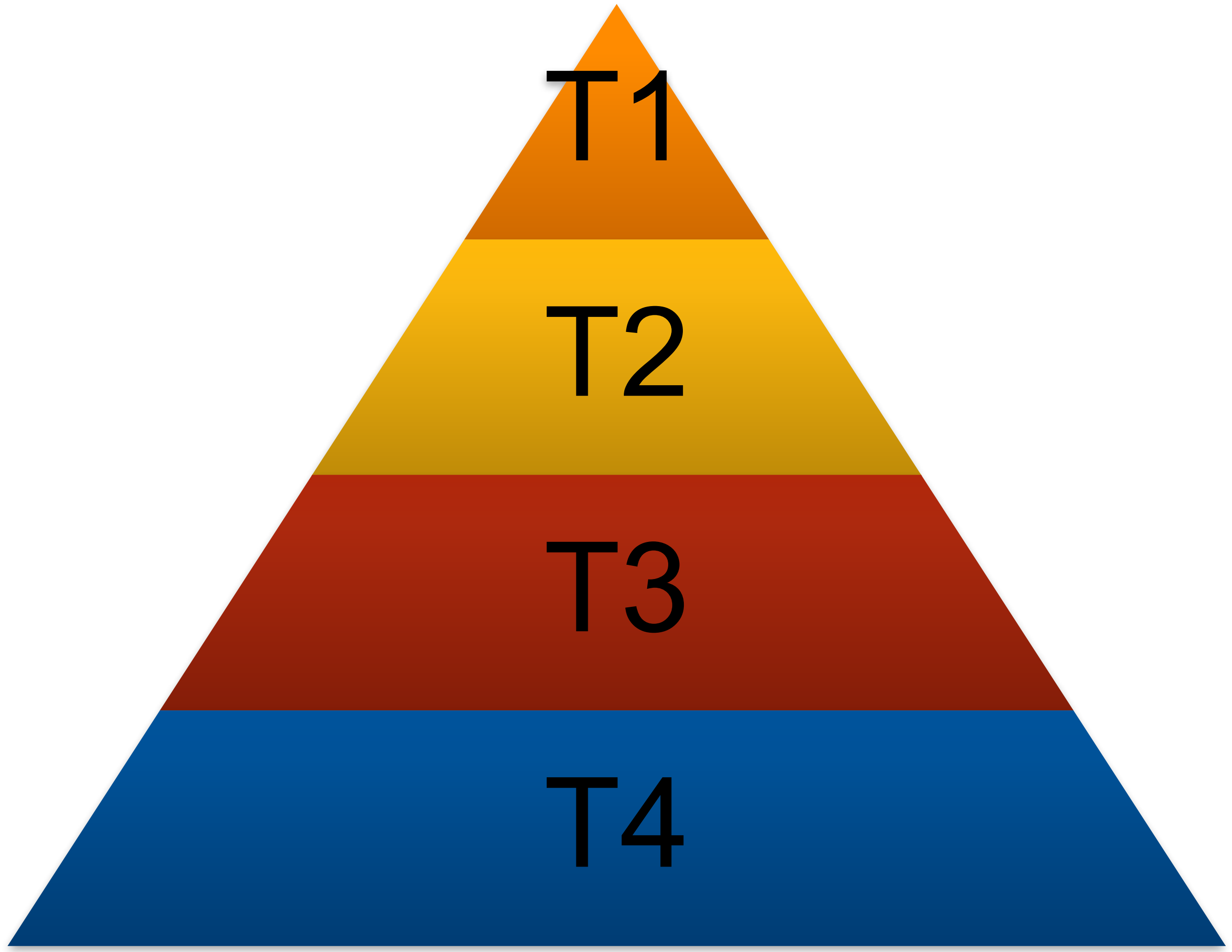


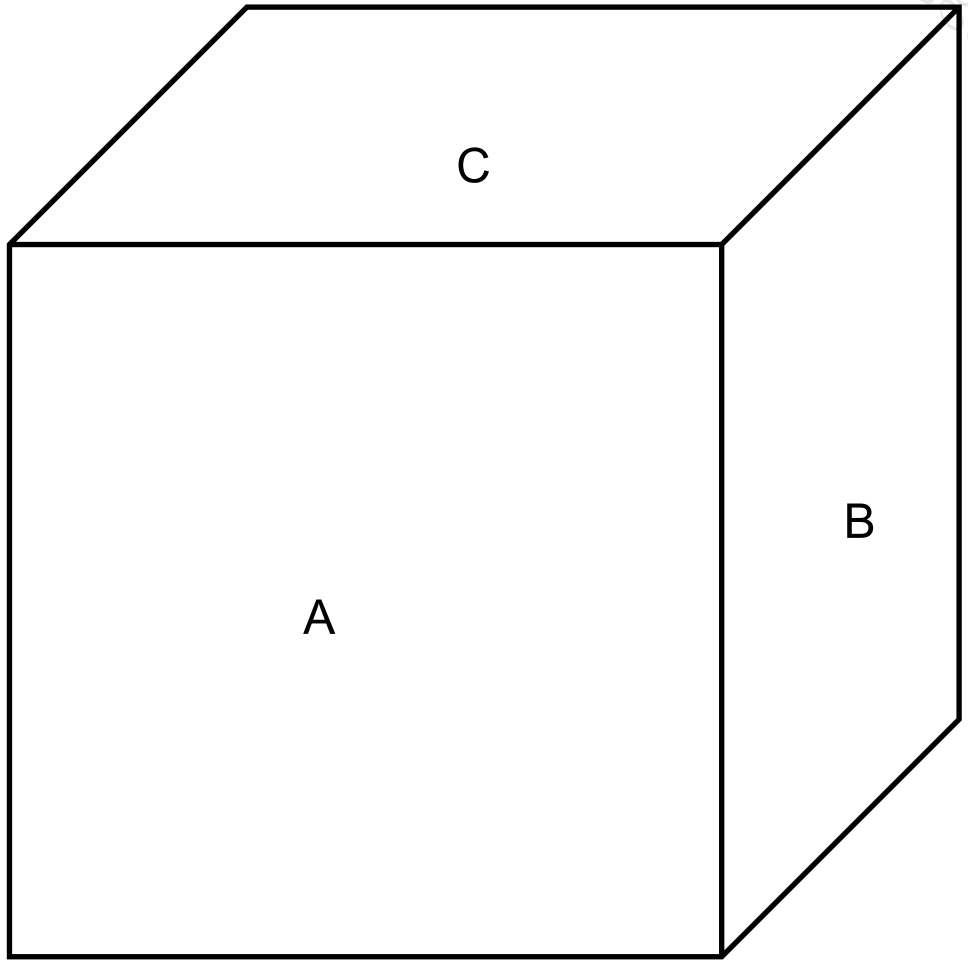


Currently a well-functioning hierarchy

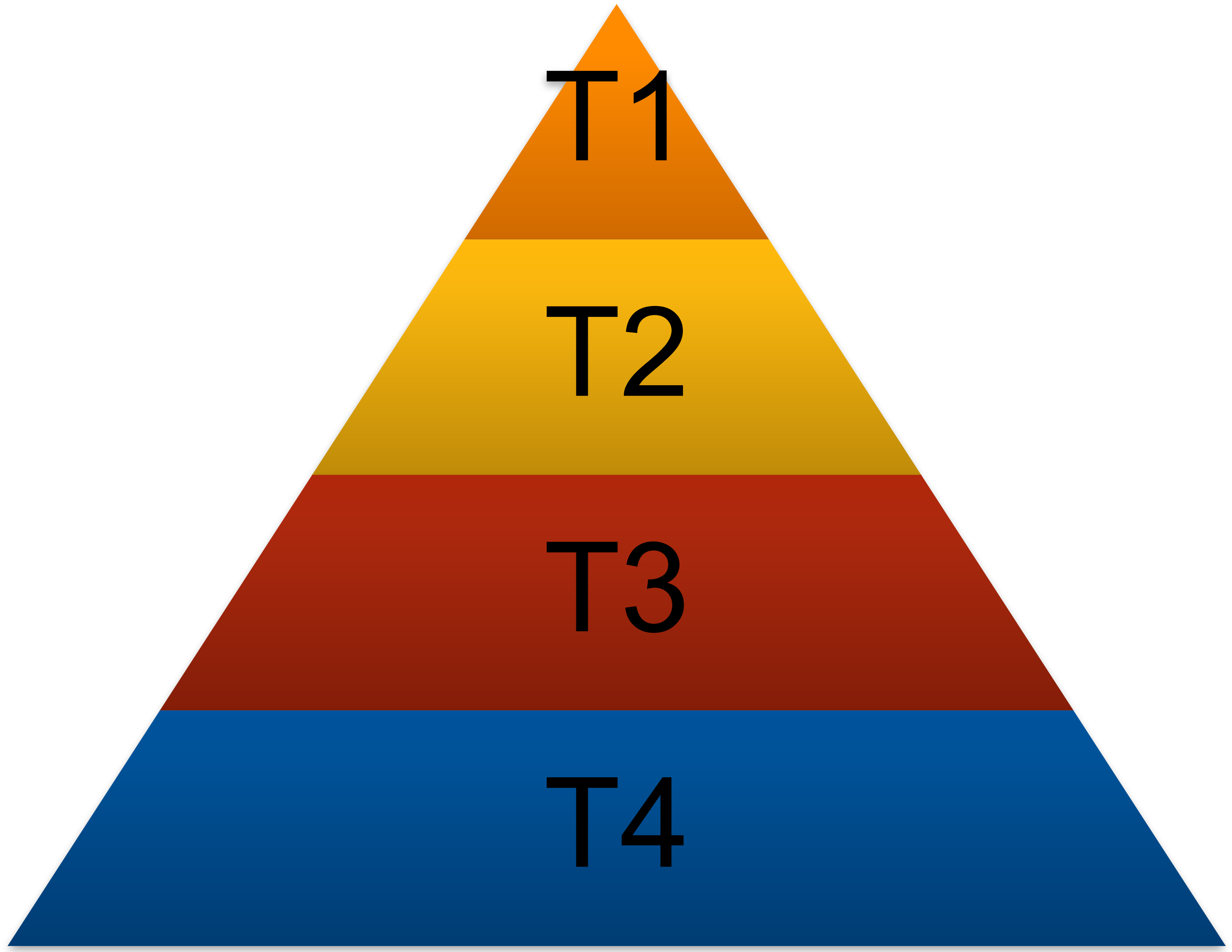
(All numbers are ballparks)

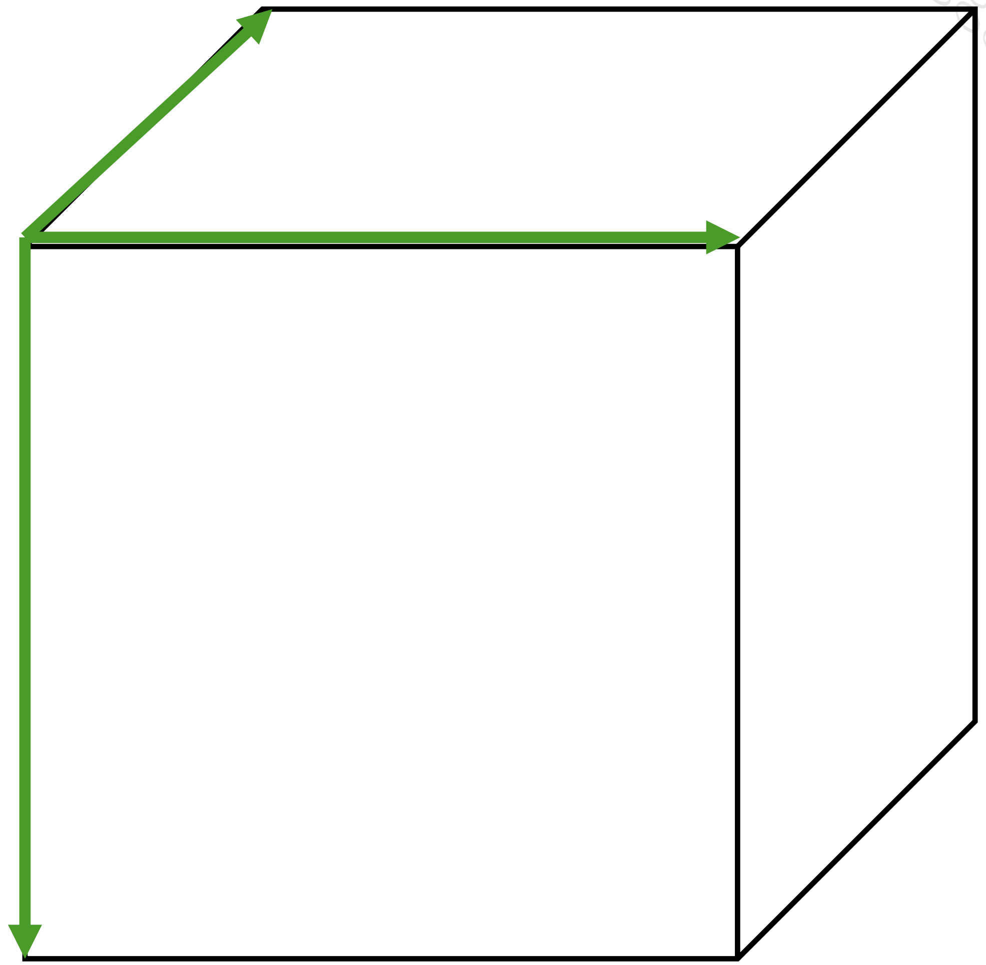
Iteration Space Tiling



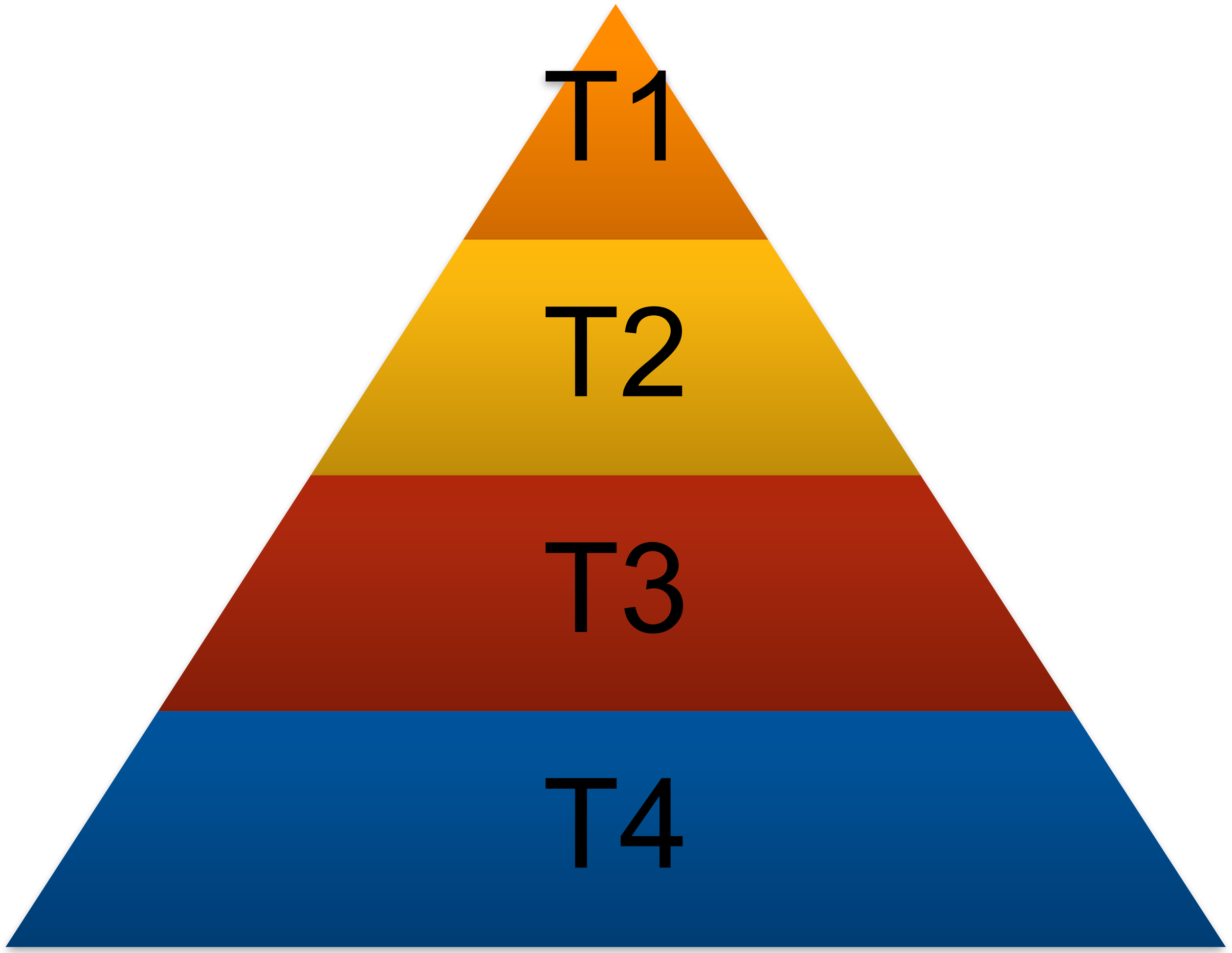


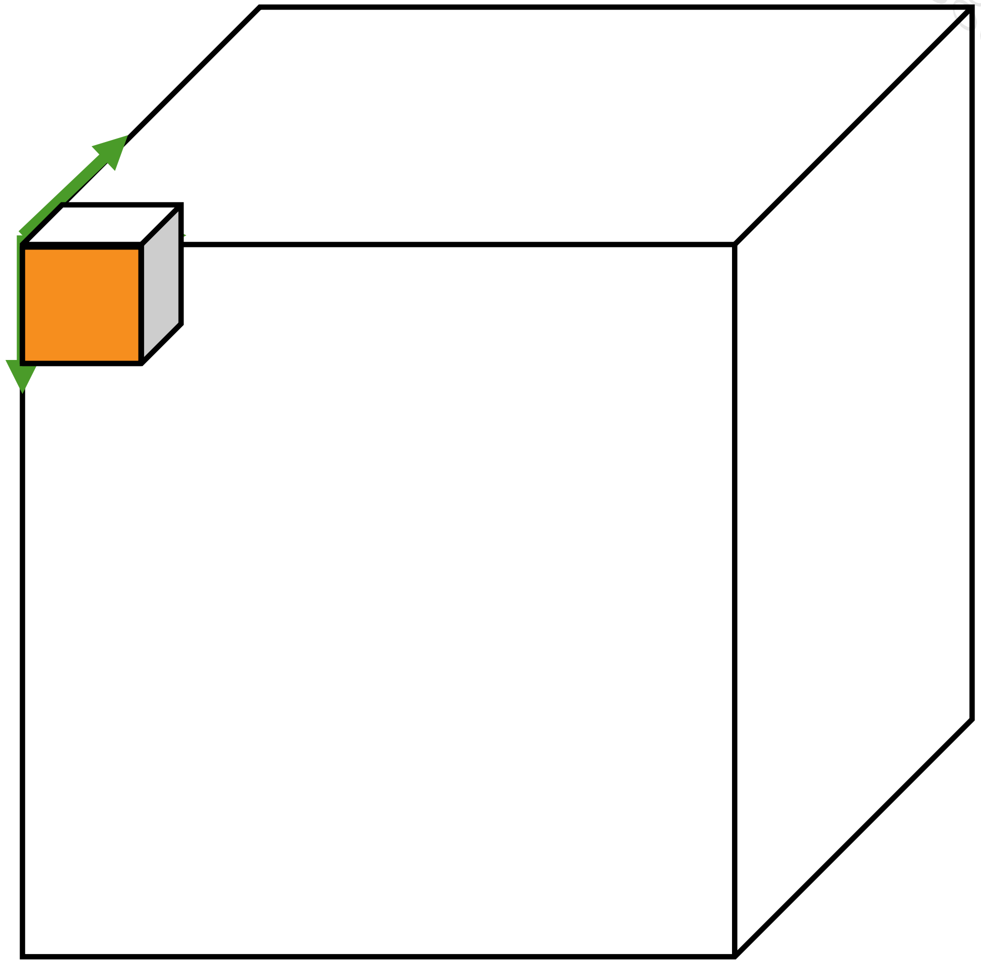
Iteration Space Tiling



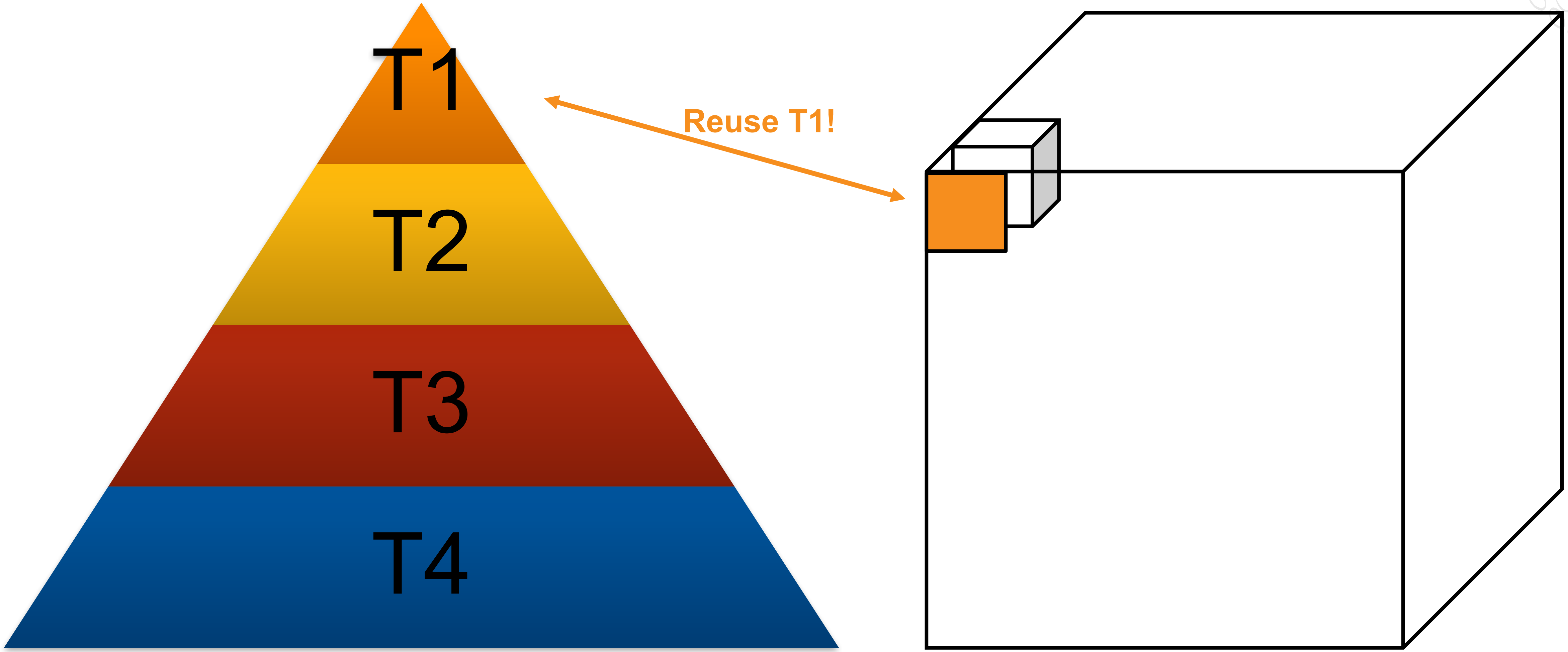


Iteration Space Tiling

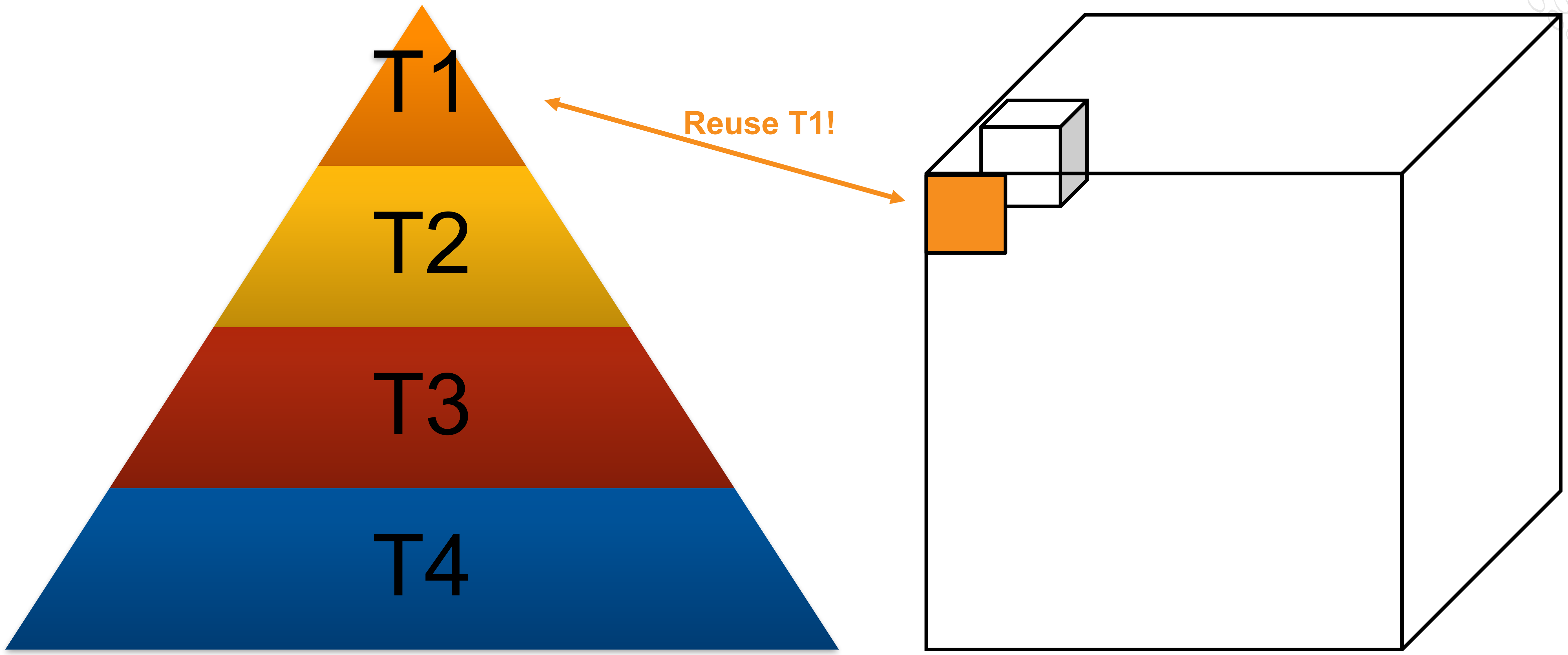




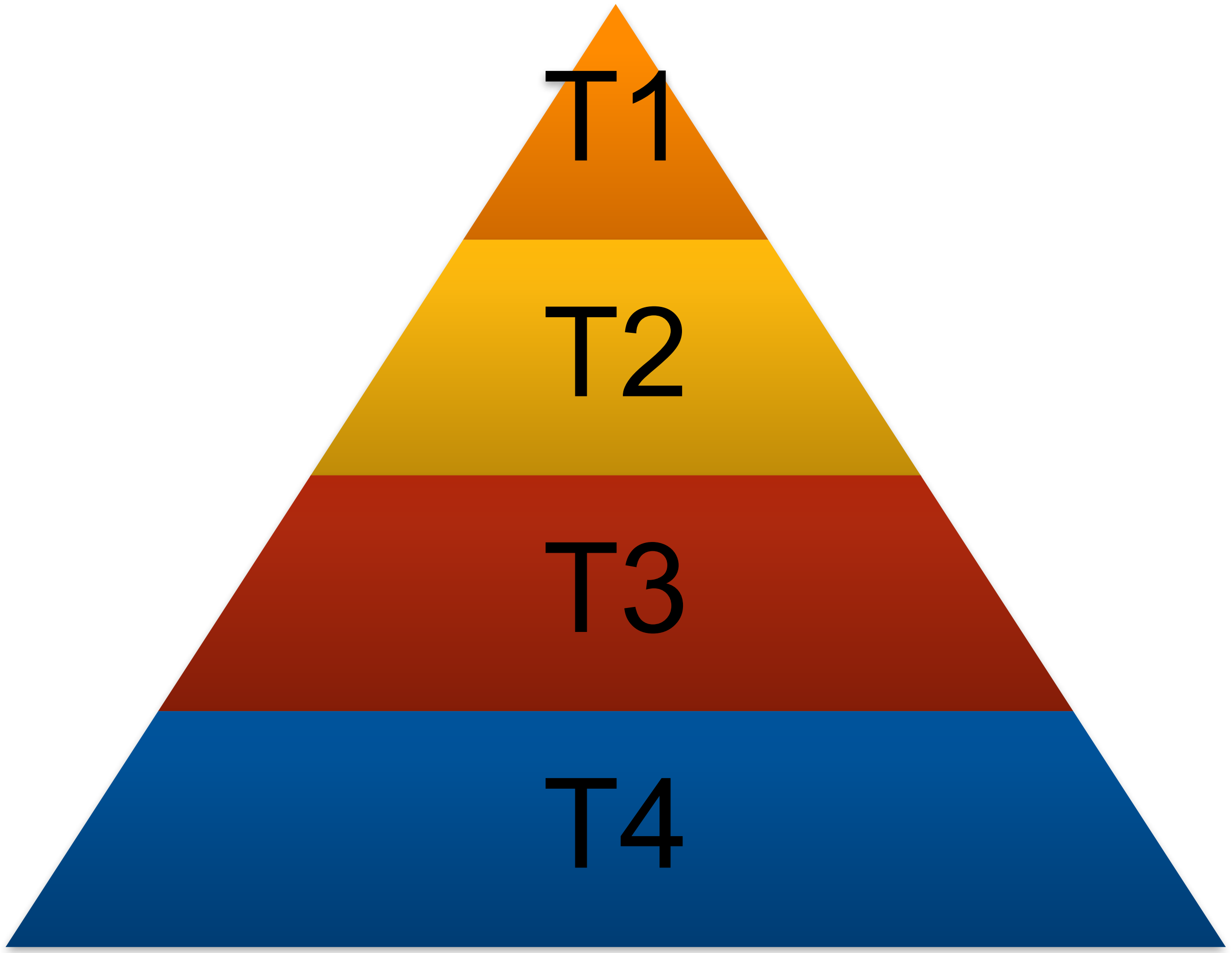
Iteration Space Tiling

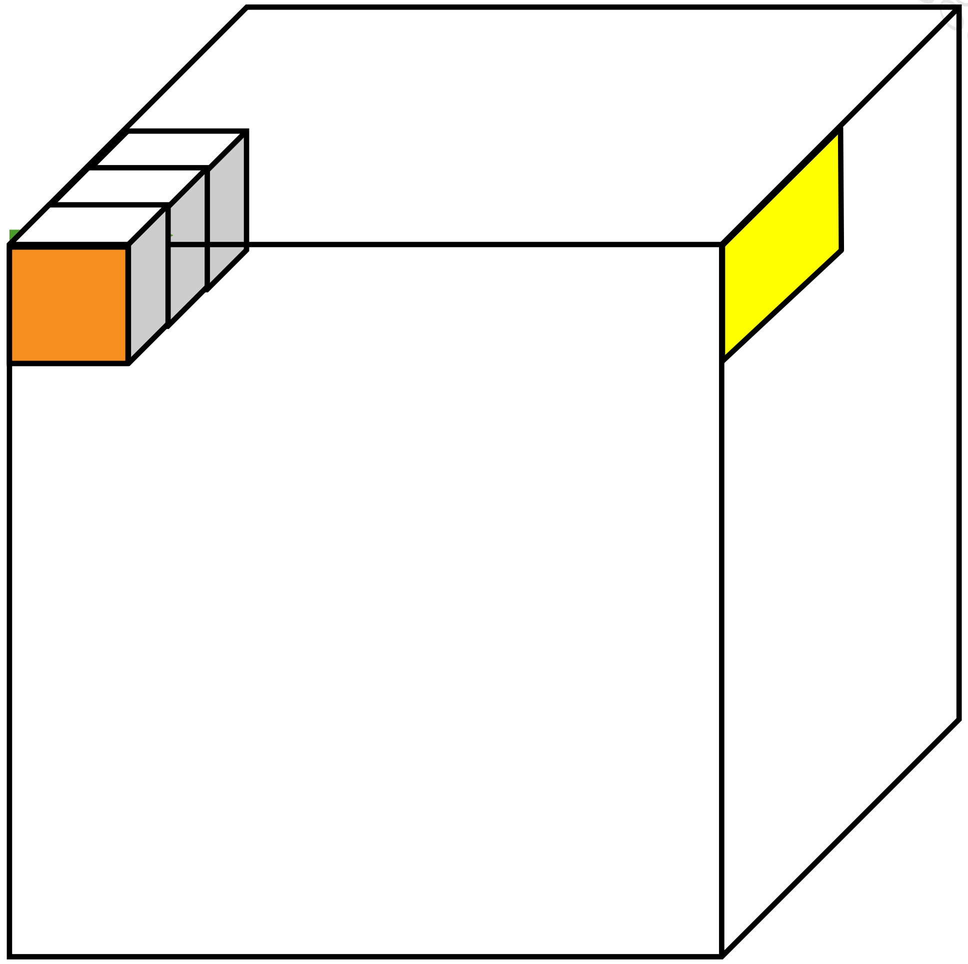


Iteration Space Tiling

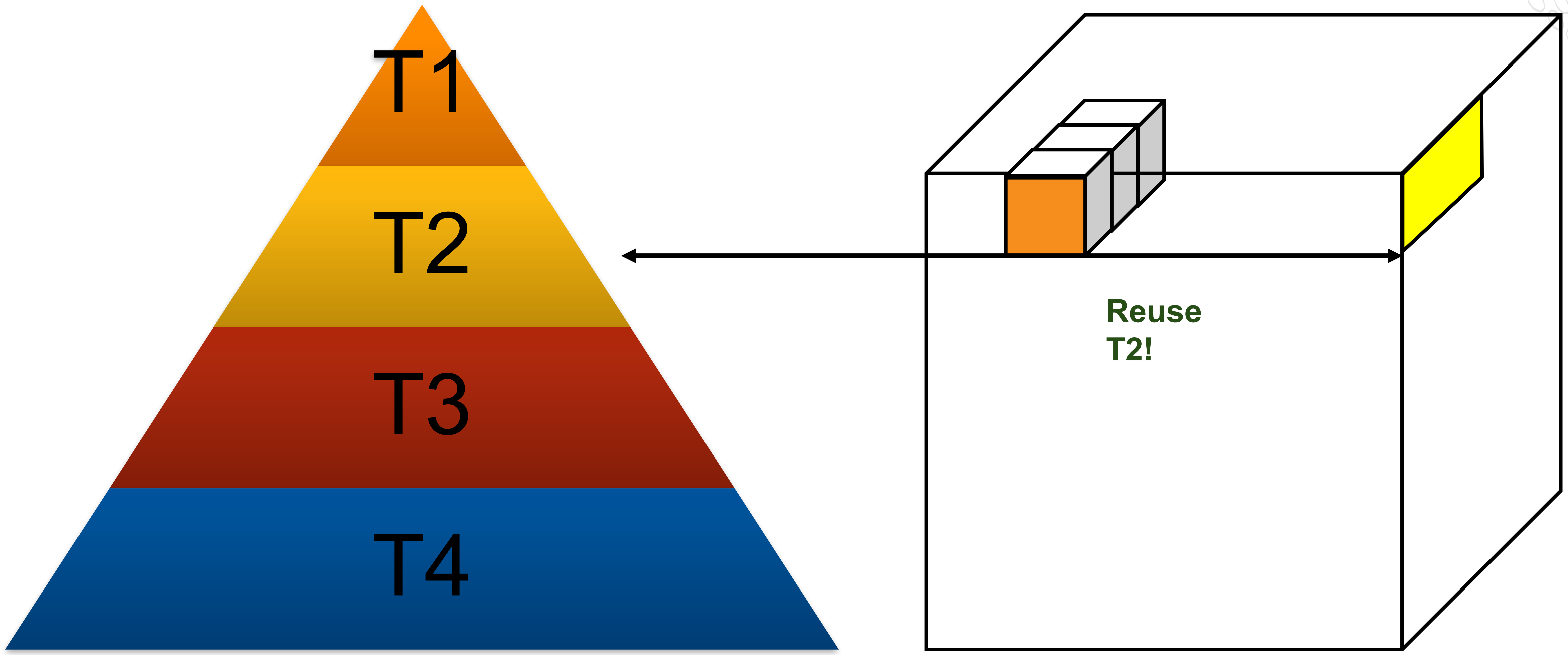


Iteration Space Tiling

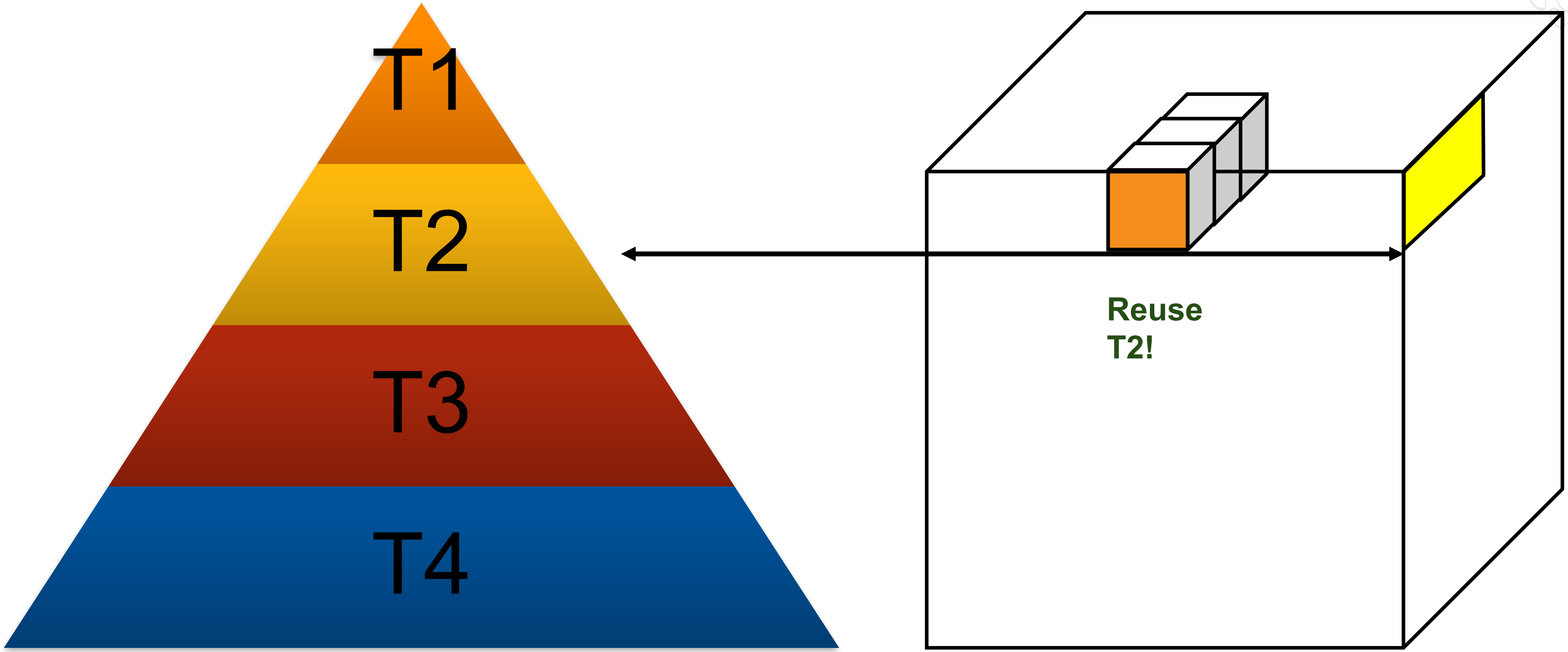




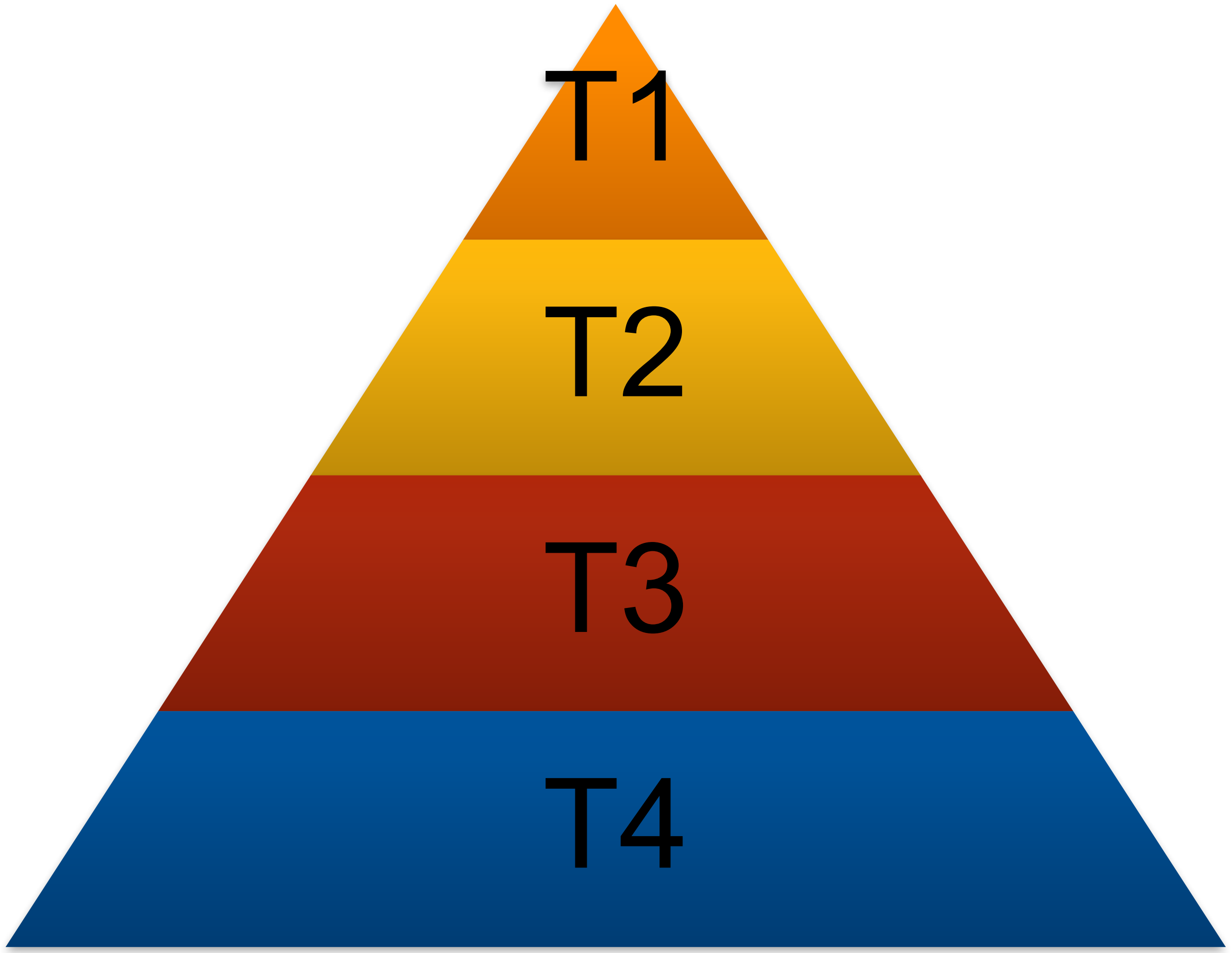
Iteration Space Tiling

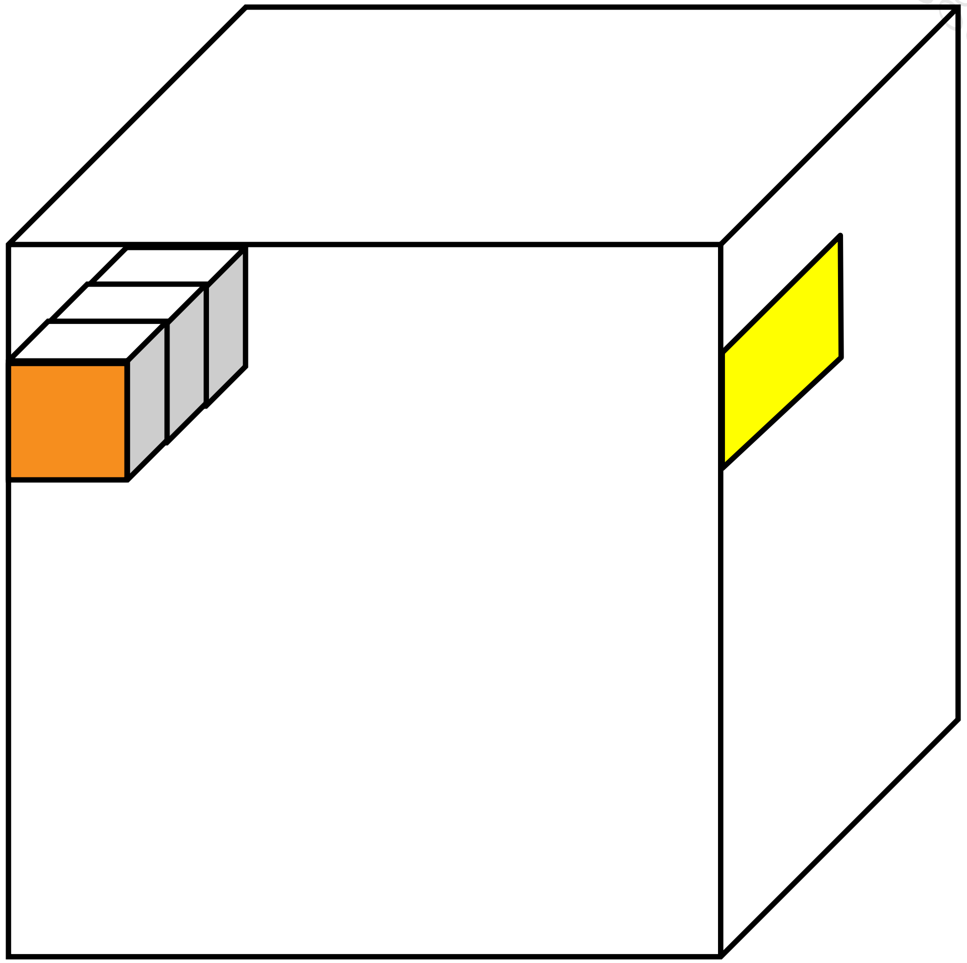


Iteration Space Tiling

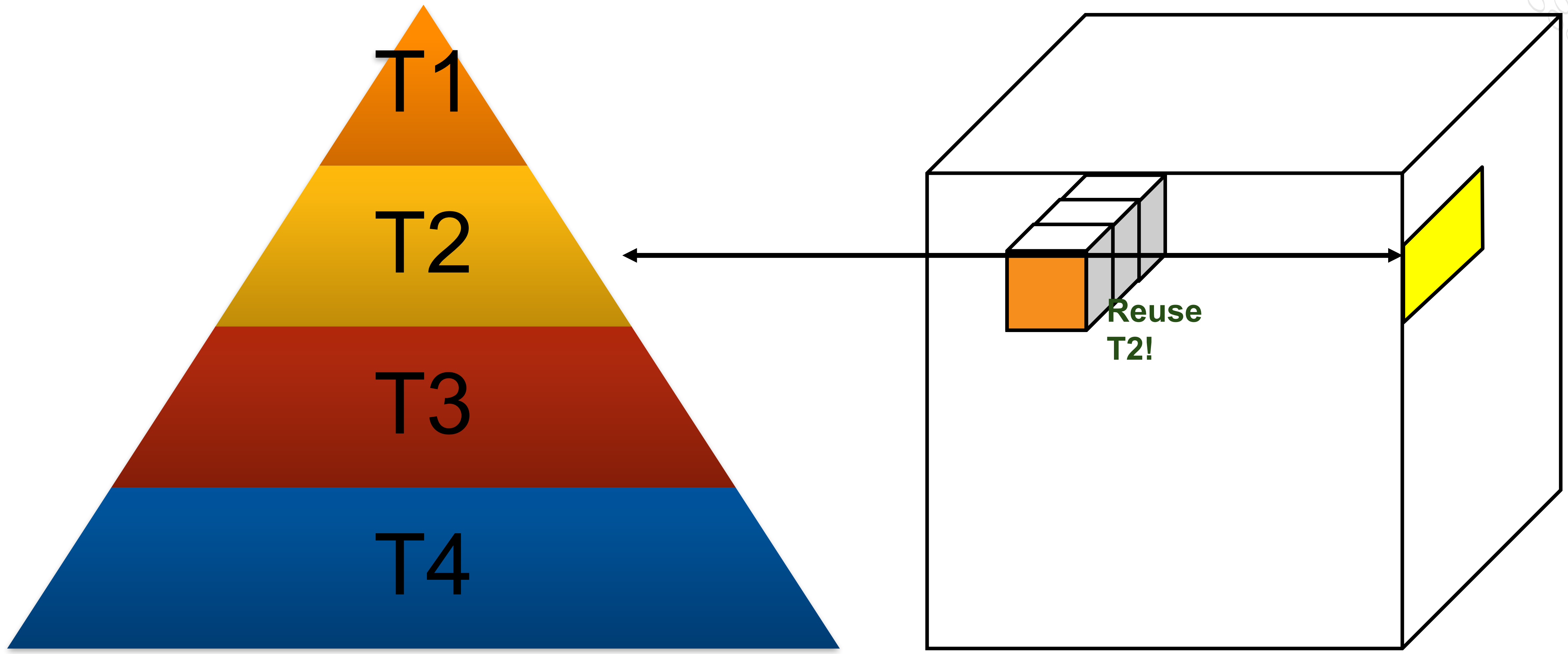


Iteration Space Tiling

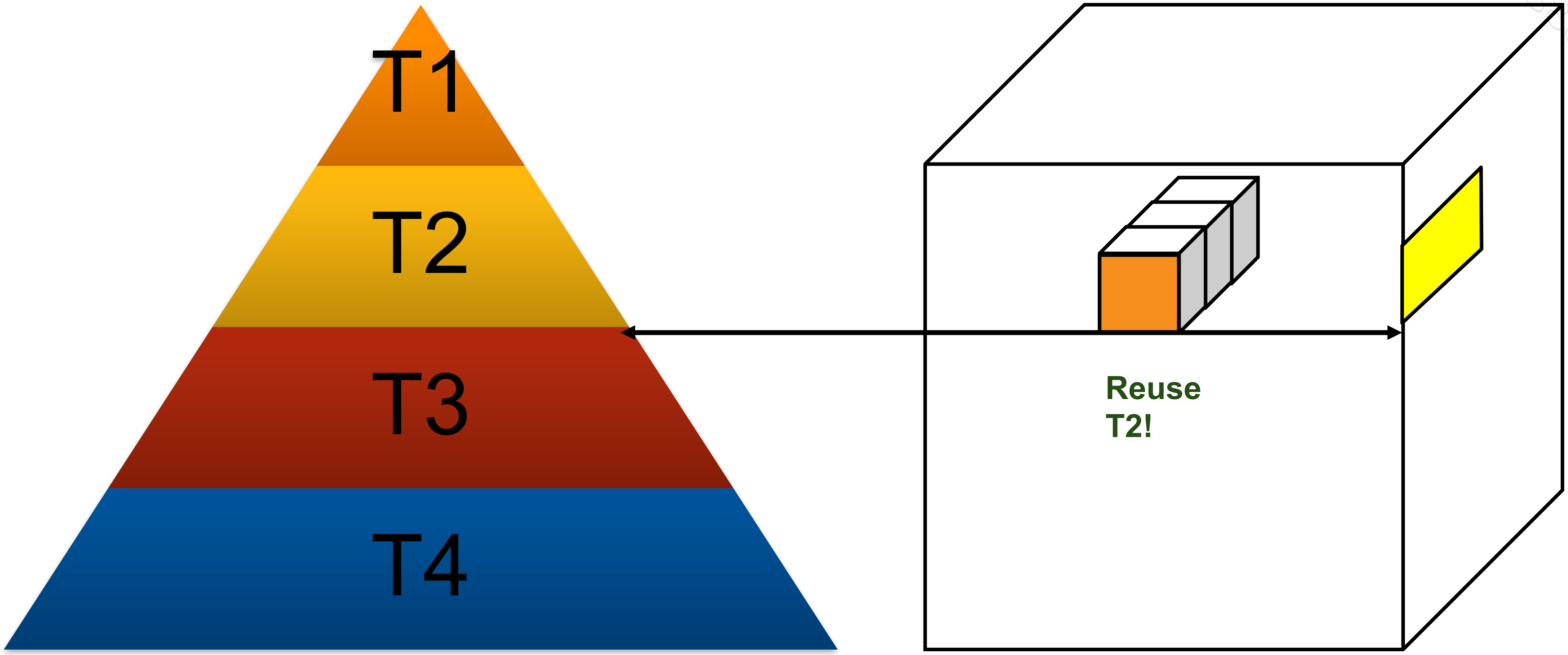




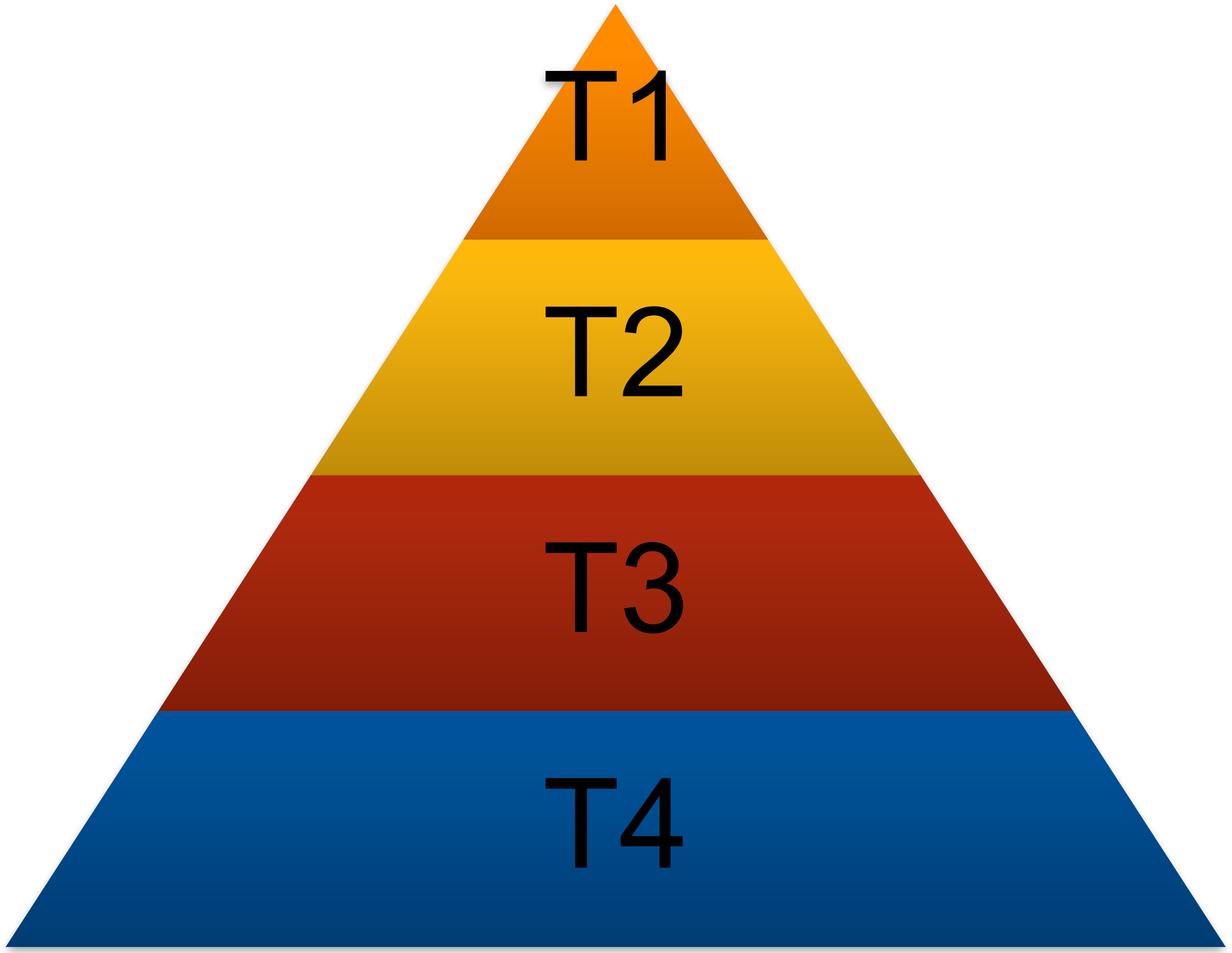
Iteration Space Tiling

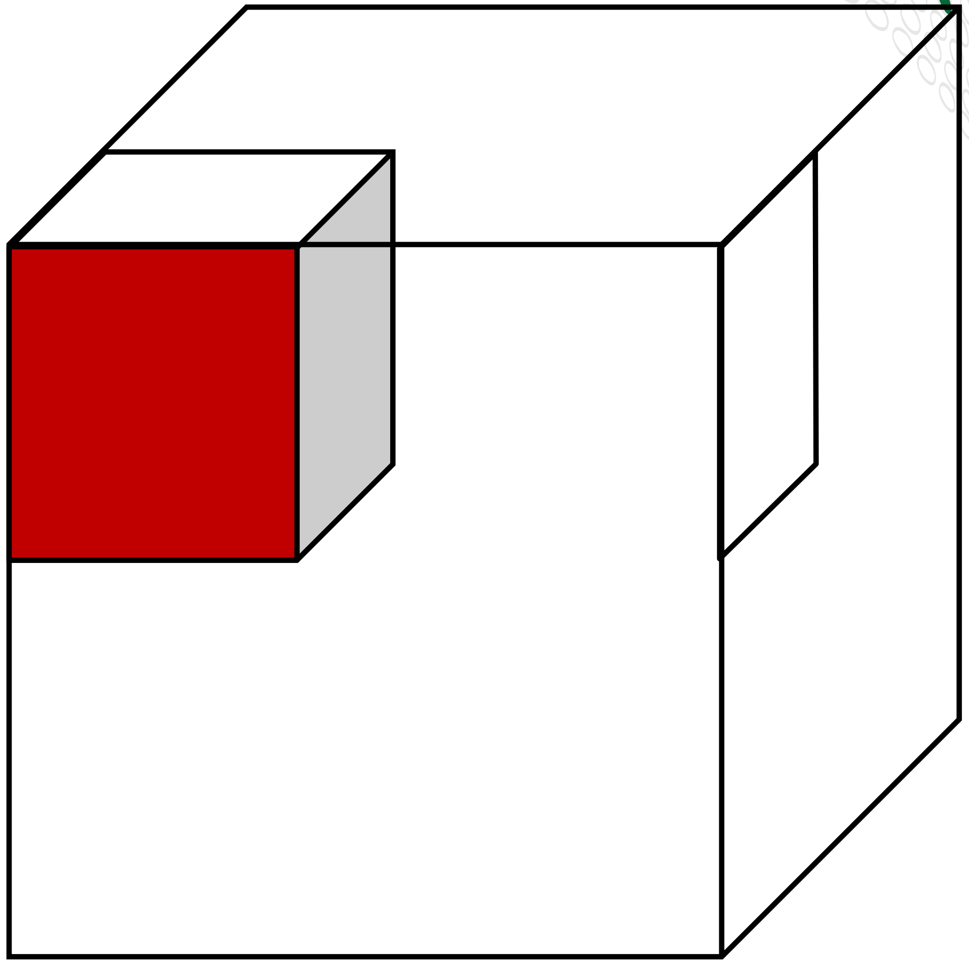


Iteration Space Tiling

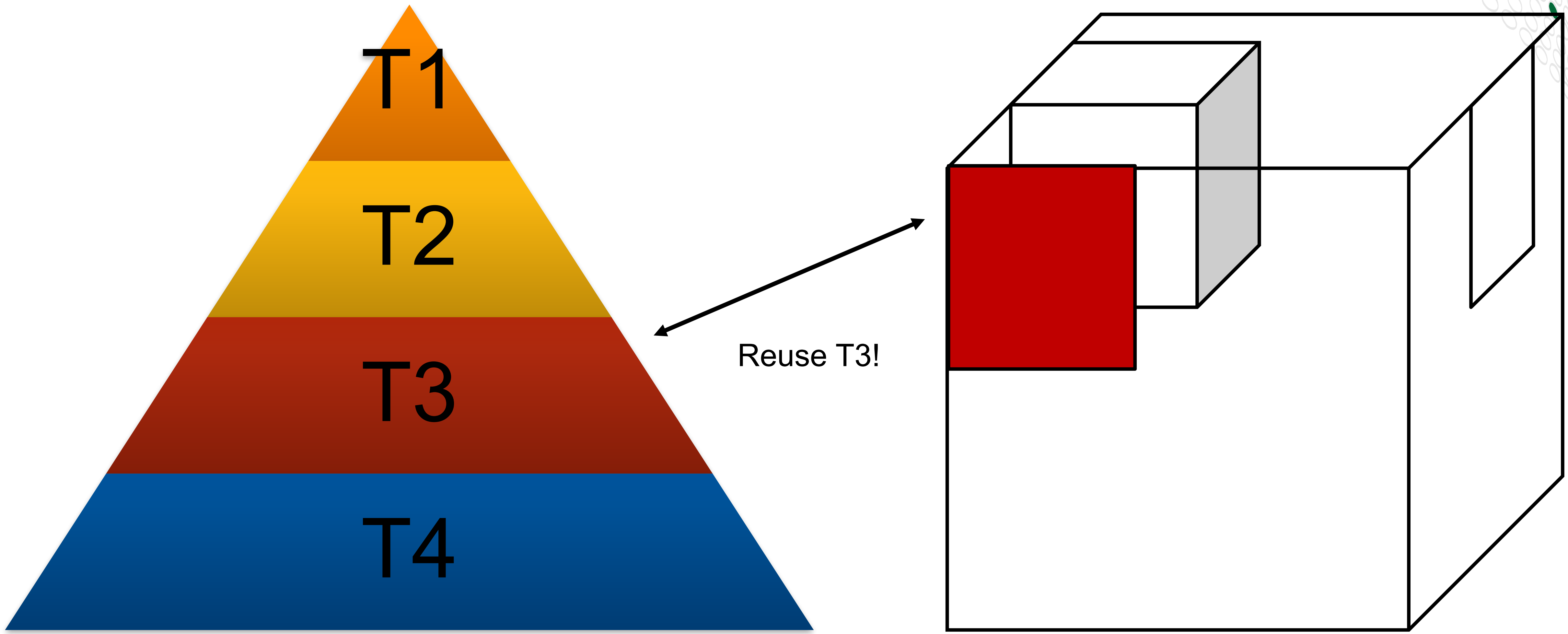


Iteration Space Tiling

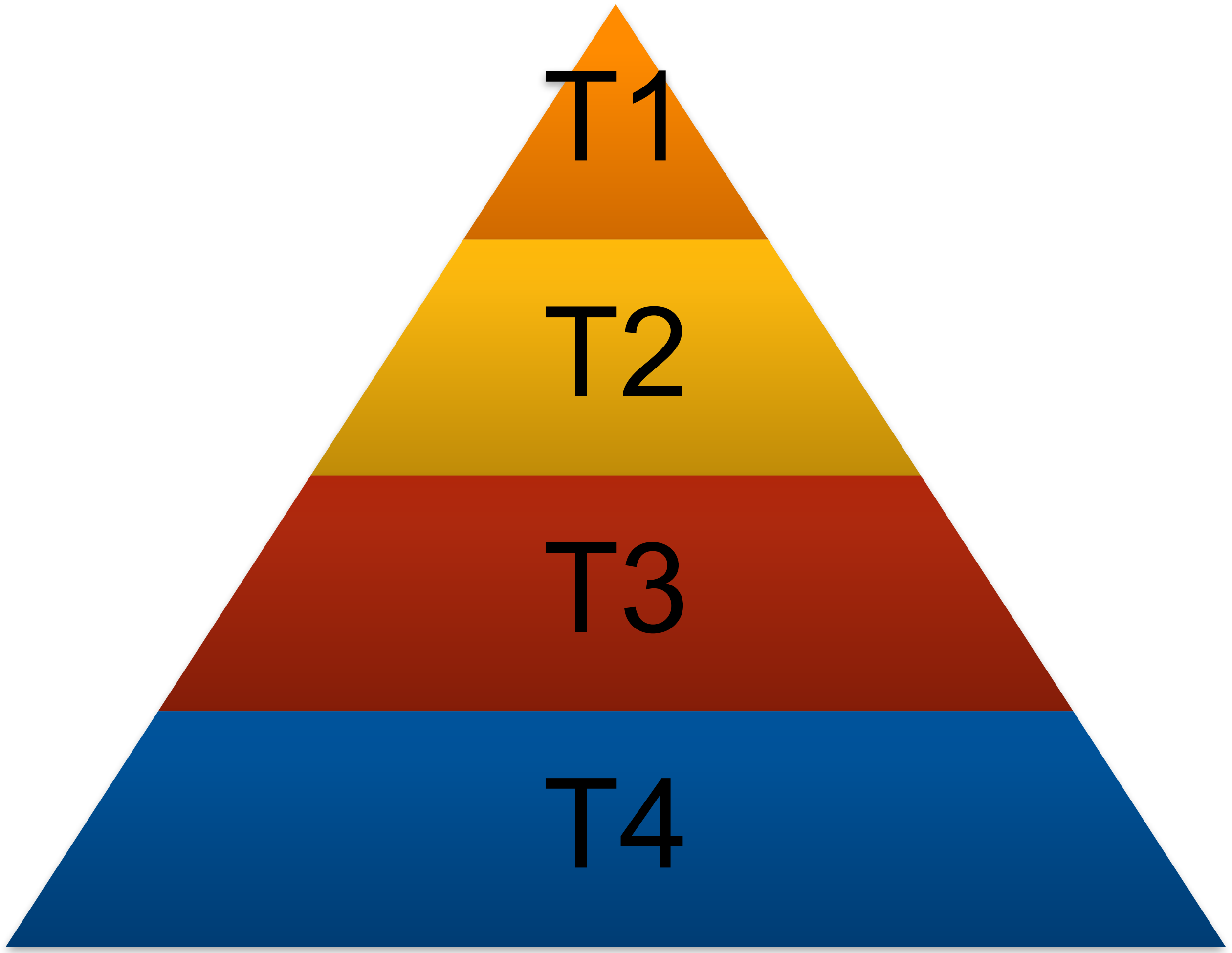


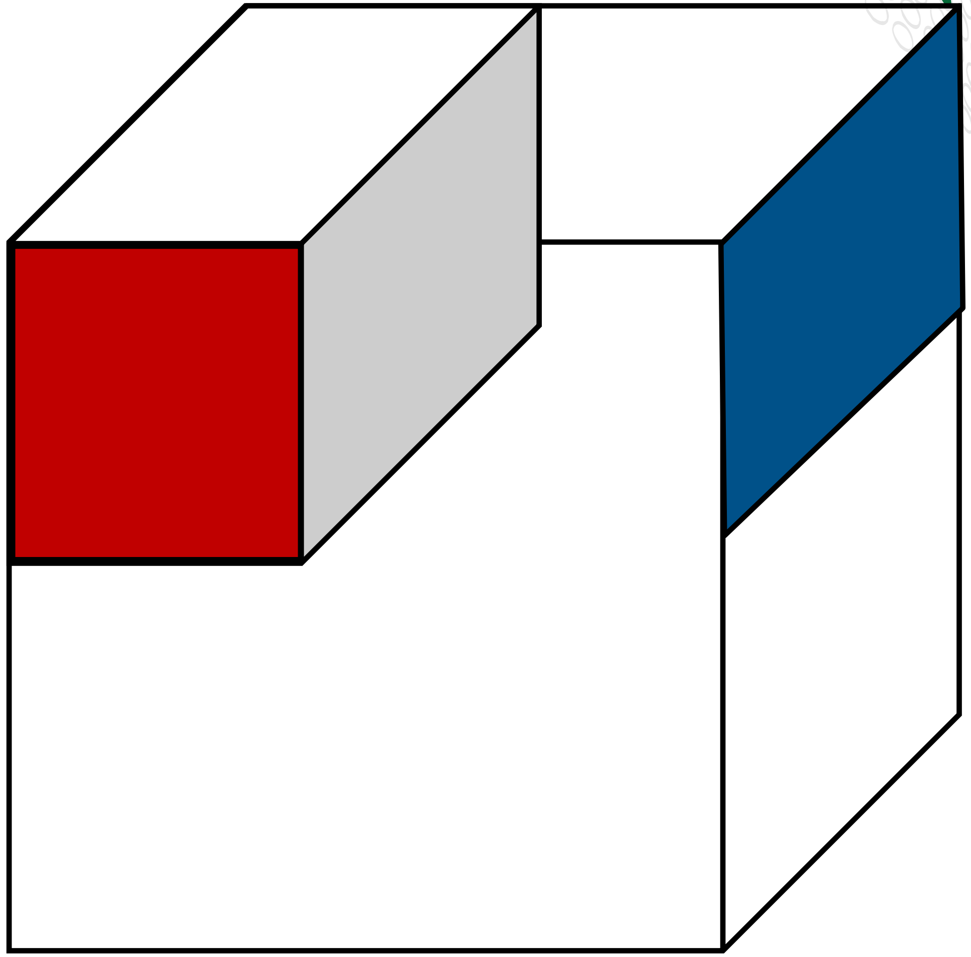


Iteration Space Tiling

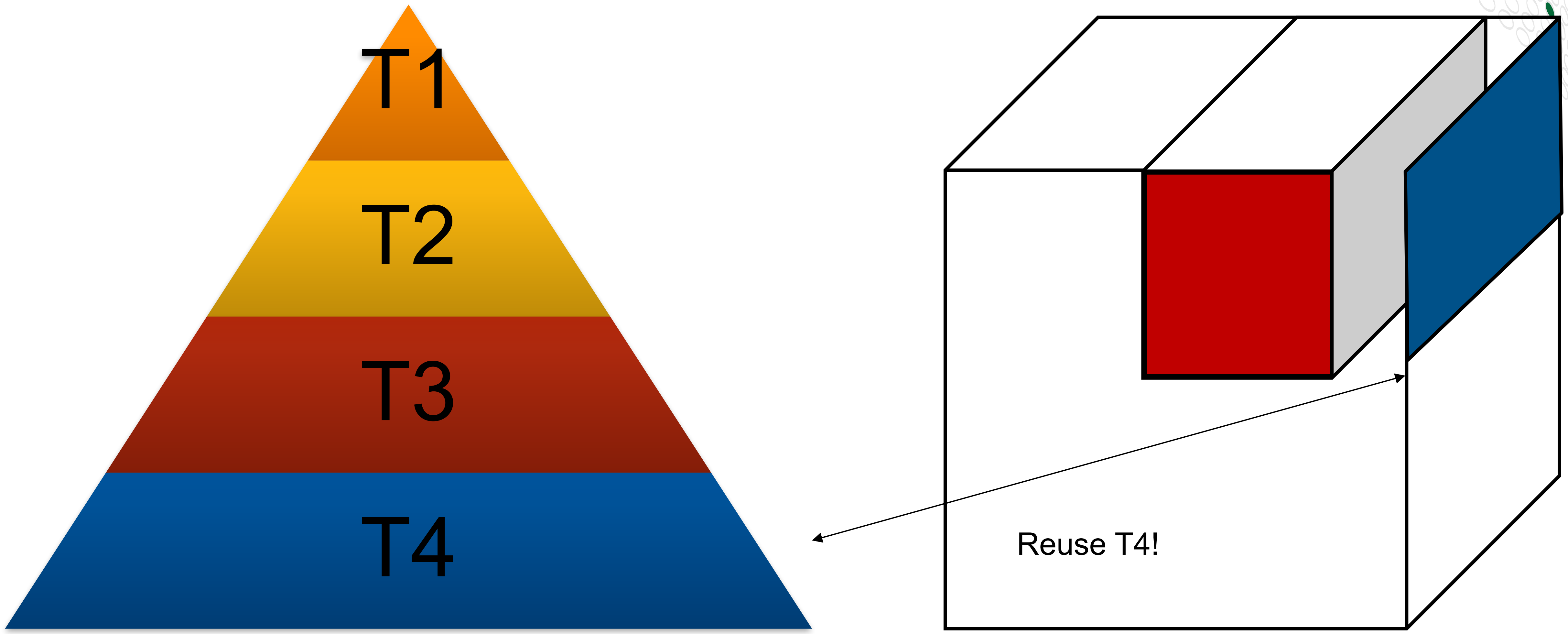


Iteration Space Tiling





Iteration Space Tiling



Iteration Space tiling

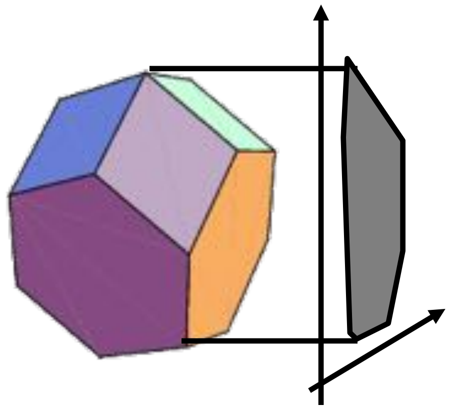
- **For single level of cache is already hard**
  - Tile-size selection is NP-hard
  - Even for one memory tier!
- **Associative caches make tiling practically difficult**
- **Tiling is**
  - Theoretically hard
  - Computationally hard
  - Practically hard
- **N-tier tiling as described here is essentially impossible**
- **Compilers use heuristic-based approaches**

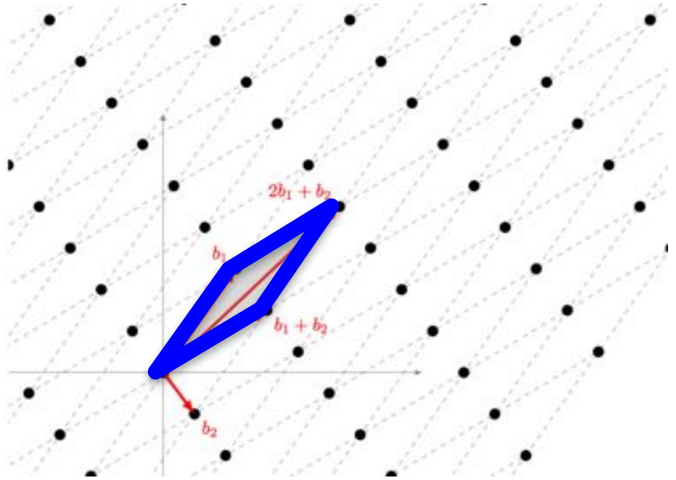
"Eiger" (A research project at Cray)

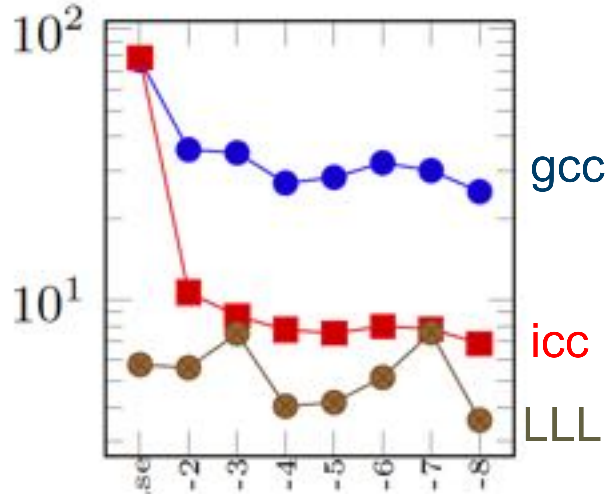
Data transformation is a hard problem. Tile decomposition is NP-hard

What structure does the hardware impose on the problem to help solve it?

C/Fortran
Loop Nest







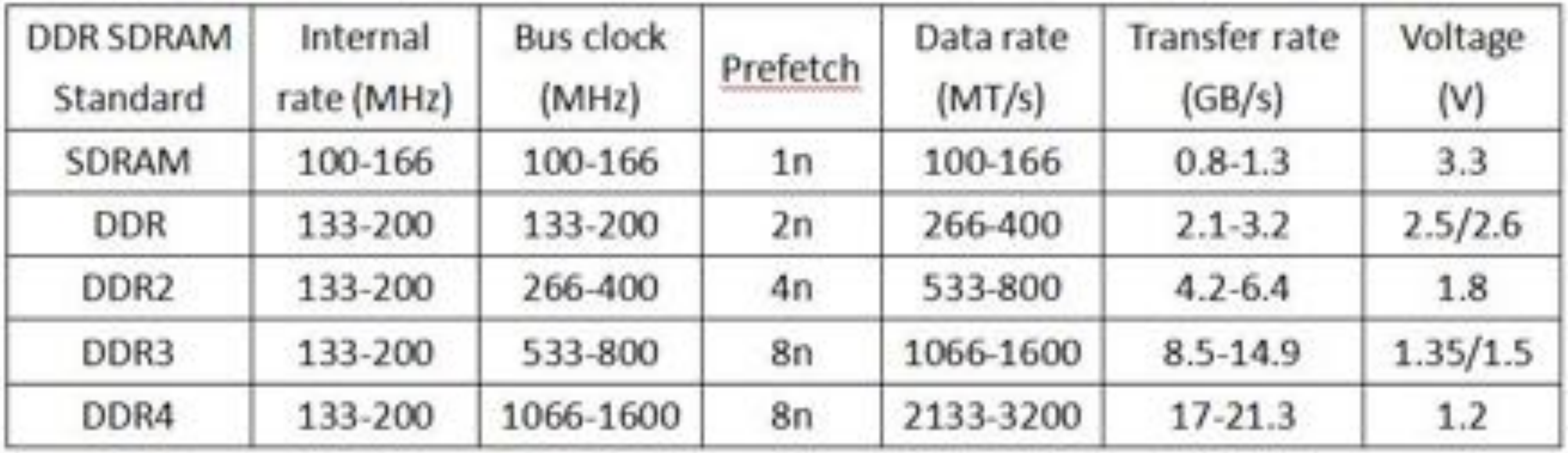
1. Eiger + LLVM for general purpose tile optimiser

2. Per-task Eiger Analysis in Octopus workflow optimisation

DRAM (main memory)

- **DRAM is dominant main memory technology**
  - Stores data as charge in a capacitor
  - Needs constant refresh
- **As opposed to SRAM which stores information in pairs of gates**
  - Does not need to be refreshed
  - Takes less space than DRAM
  - Much slower than DRAM
  - Old SRAM designs were power hungry
- **DDR is the standard implementation of DRAM**

DRAM speeds



DDR SDRAM Standard	Internal rate (MHz)	Bus clock (MHz)	Prefetch	Data rate (MT/s)	Transfer rate (GB/s)	Voltage (V)
SDRAM	100-166	100-166	1n	100-166	0.8-1.3	3.3
DDR	133-200	133-200	2n	266-400	2.1-3.2	2.5/2.6
DDR2	133-200	266-400	4n	533-800	4.2-6.4	1.8
DDR3	133-200	533-800	8n	1066-1600	8.5-14.9	1.35/1.5
DDR4	133-200	1066-1600	8n	2133-3200	17-21.3	1.2

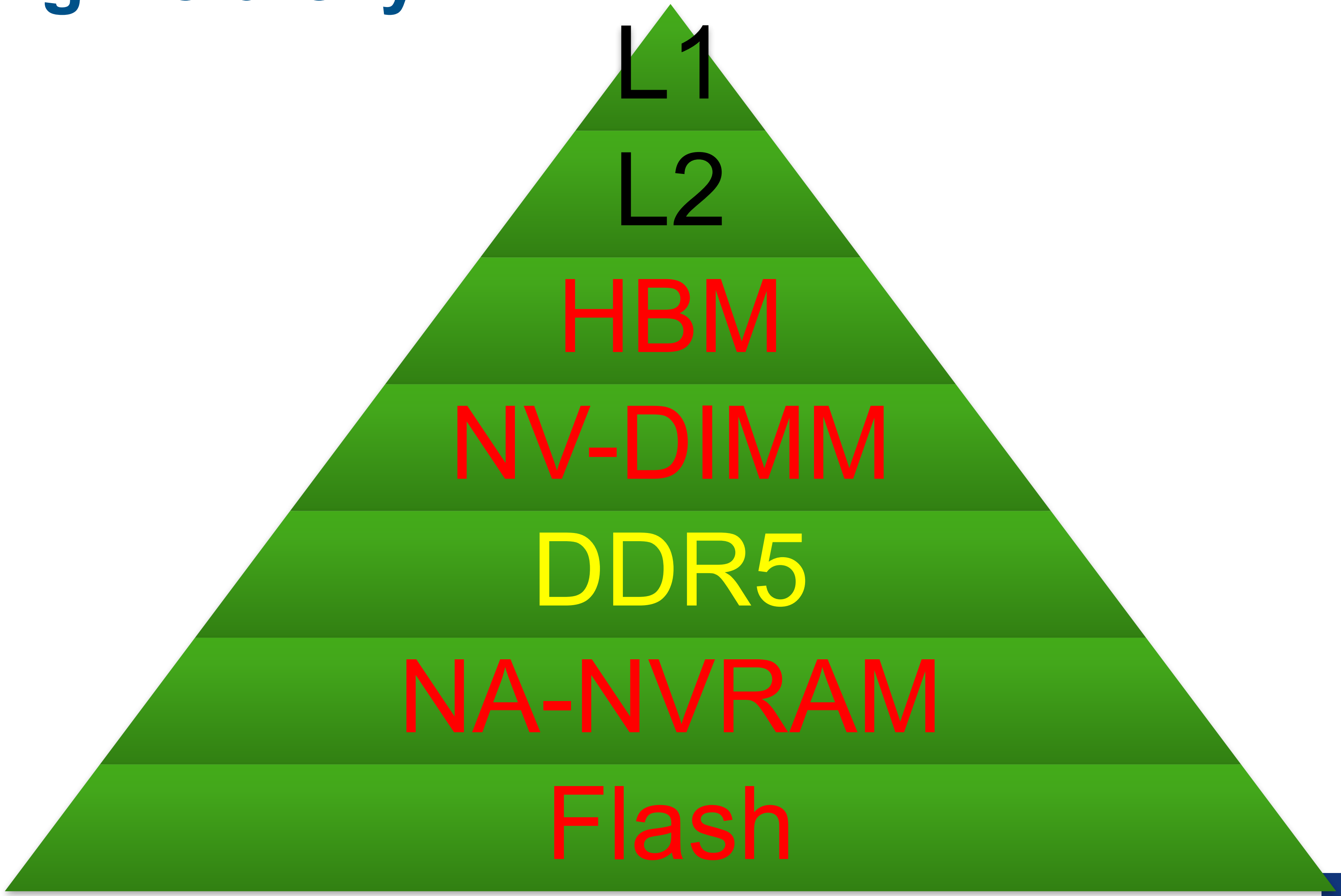
DDR5 (2019) **40 GB/s** **0.6**

This evolutionary development of DDR5 is not sufficient to close the memory wall
Hence, innovation has come above and below main memory

More info

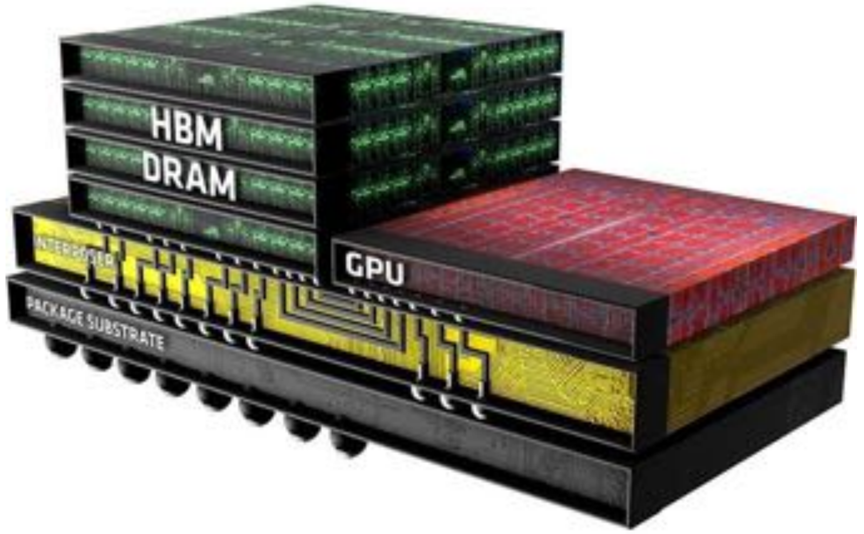
- **Cache details**
  - ***Chapter 2 of Computer Architecture: A Quantitative Approach (5th Addition),*** David Patterson and John Hennessy
- **Iteration Space Tiling**
  - ***D. Lam, E. E. Rothberg, and M. E. Wolf. The cache performance and optimizations of block algorithms.*** ACM SIGOPS Operating Systems Review 25(Special Issue)
  - ***Optimizing Compilers for Modern Architectures: A Dependence-based Approach*** 1st Edition, Randy Allen and Ken Kennedy (book)
- **Eiger**
  - ***Adjiashvili, U.-U. Haus, and A. Tate. Model-driven, automatic tiling with cache associativity lattices***, arXiv preprint arXiv:1511.05585, 2015
- **DRAM designs**
  - ***Chapter 2 of Computer Architecture***

Emerging hierarchy



High-Bandwidth Memory (HBM)

- **Graphics companies spearheaded innovation in fast memory**
- **Core concept**
  - DRAM packaged alongside processor on single substrate
  - Stack up to 4 DRAM dies and memory controller together
  - Connect them using through-silicon-via (TSV)

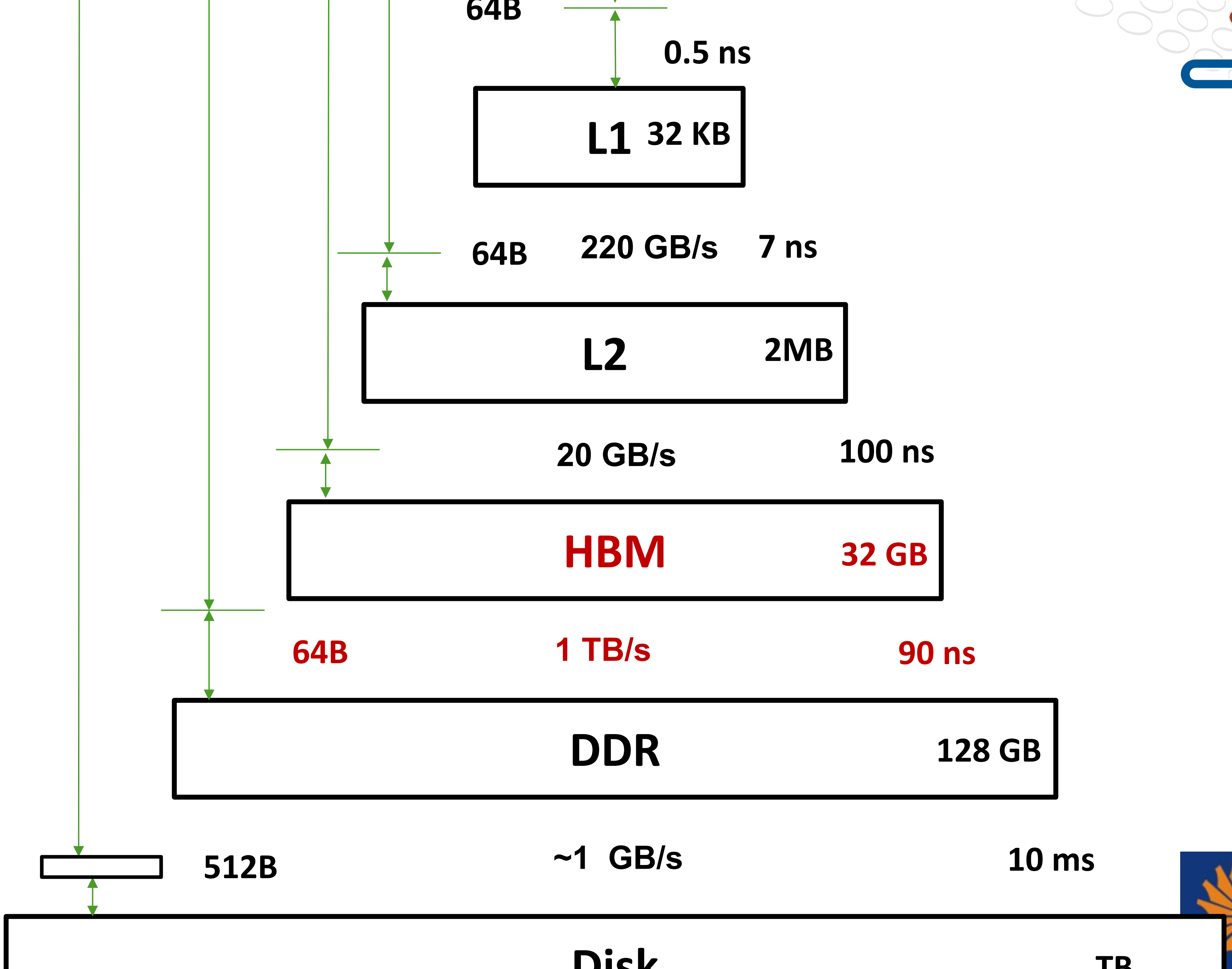


- **Result:**
  - **higher bandwidth, less power, smaller form-factor than DRAM !**

HBM (cont)

- Intel's KNL has a poor-man's HBM (MCDRAM or HMC)
  - AMD Fiji and NVIDIA Pascal have the first good HBM
- Limit on the number of stacks that can be supported (4 = 32GB, possibly 64GB)
- 6-7 pJ / bit - compared to 30 pJ/bit for DDR
- Prices for HBM are currently high
- HBM should be a game-changer in HPC, no question
- Can be used in caching mode or accessed directly

EXPERTISE

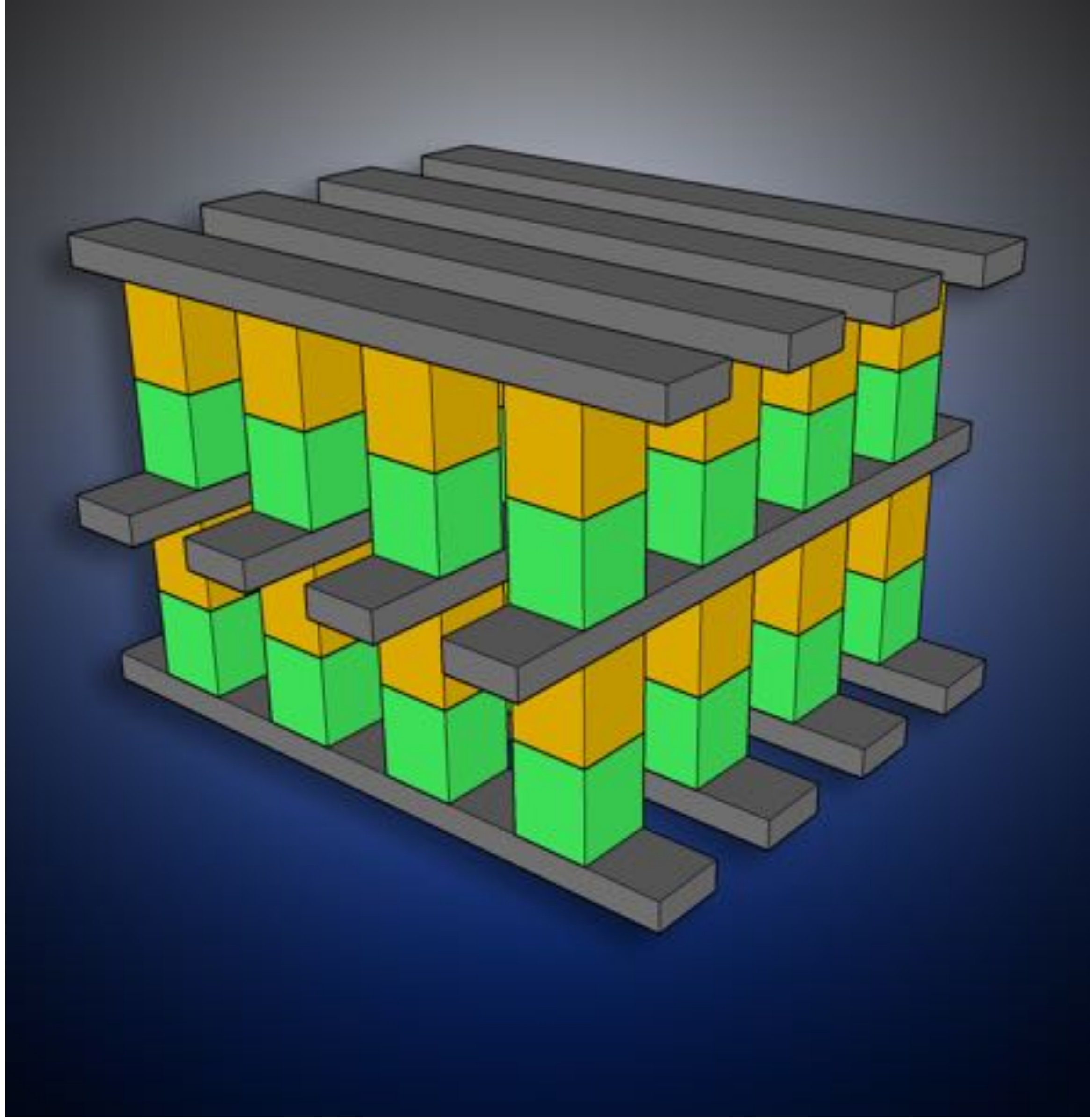

64B
0.5 ns
L1 32 KB
64B
220 GB/s
7 ns
L2
2MB
20 GB/s
100 ns
HBM
32 GB
64B
1 TB/s
90 ns
DDR
128 GB
512B
~1 GB/s
10 ms
Disk
TB

Non-volatile memories

- **Recall that DRAM requires constant refresh**
  - Energy costs of storing and moving data becoming too high
  - Data lost after power cycle
- **Non-volatile memories are the class of memories that store information in ways that do not need refresh**
- **Not a new general concept (SSDs)**
- **However, various new technical innovations:**
  - Intel’s 3dXPoint
  - Phase Change Memory (PCM)
  - Magnetoresistive memories (MRAM)
  - Resistive RAM (ReRAM)
  - Flash
- **Most of these are a way from being viable HPC Components**
- **NV programming model – SNIA / pmem.io**

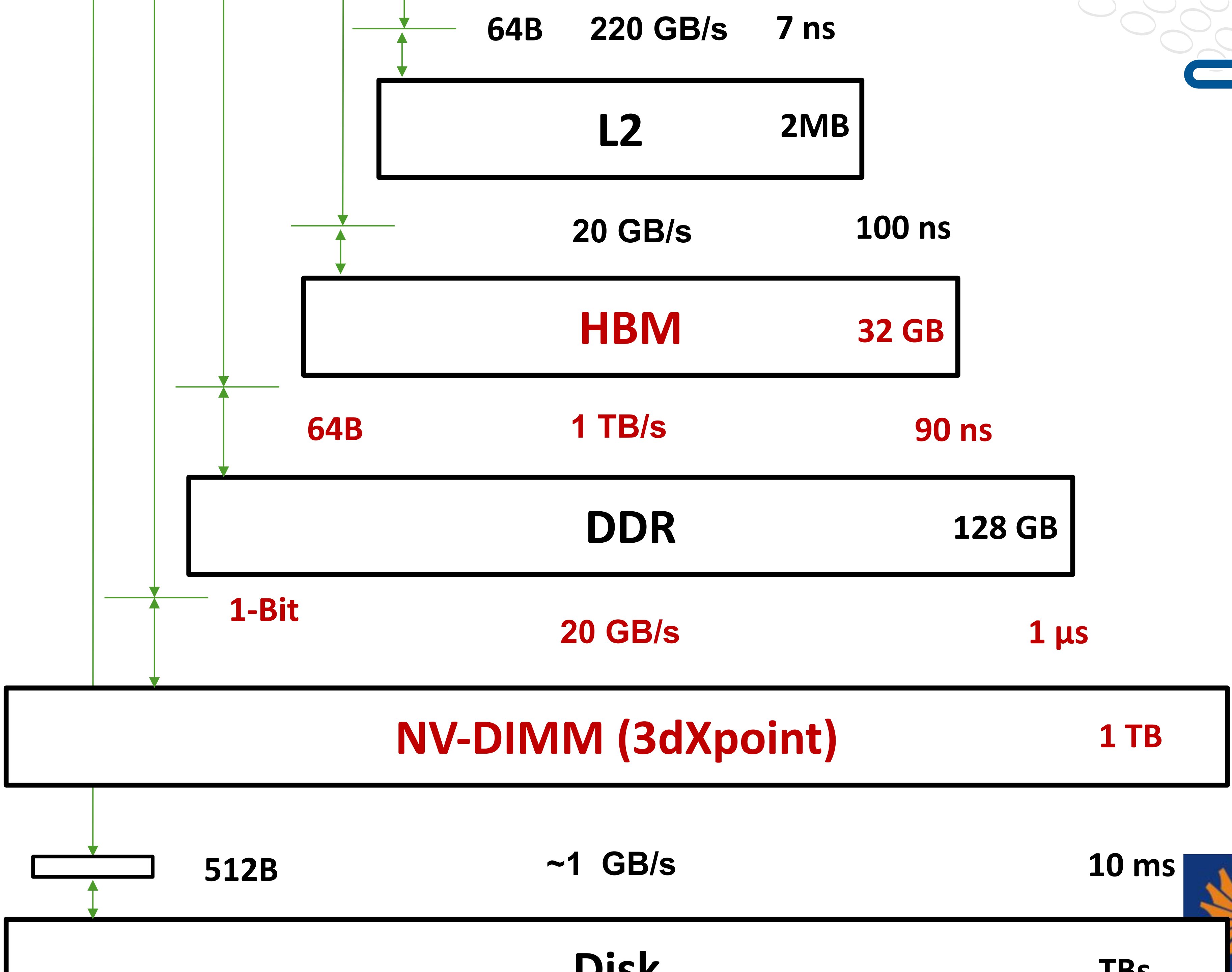
3dXpoint (Intel and Micron)

- **(secret) microscopic material is packed into columns**
  - Contains a memory cell and a selector
- **Cells are connected with a cross-structure**
- **To select any individual cell, any wire above and below is selected.**
- **Selection happens without transistors – these are voltage regulated.**
- **Most interestingly : can be used as either memory or storage depending on the use-case**
- **Specs for 3dXpoint are secret, let's assume**
  - Similar bandwidth to DRAM
  - Latency only 10x DRAM
  - Much higher capacity than DRAM
  - Lower power requirements than DRAM & SSD
  - Higher price (now)
  - Cheaper price (later)



Disclaimer: The specs are taken from public information, may or not be correct

EXPERTISE


64B
220 GB/s
7 ns
L2
2MB
20 GB/s
100 ns
HBM
32 GB
64B
1 TB/s
90 ns
DDR
128 GB
1-Bit
20 GB/s
1 µs
NV-DIMM (3dXpoint)
1 TB
512B
~1 GB/s
10 ms
Disk
TBs

Use-cases

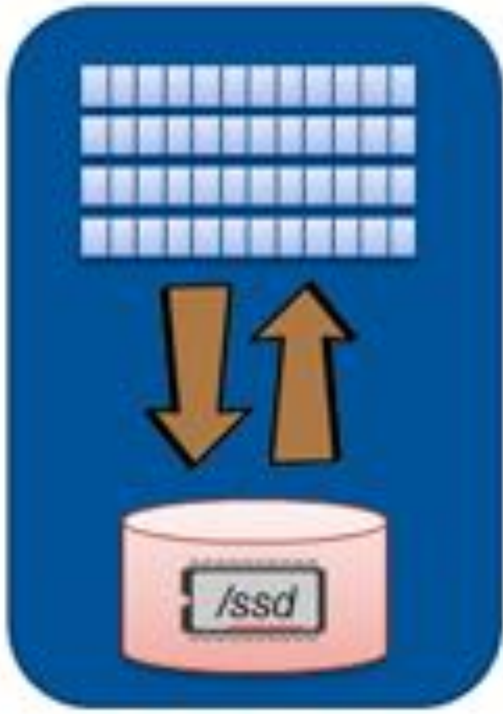
- **Persistent storage for closely coupled applications – write and persist data**
  - E.g. in-situ analysis
  - E.g. Coupled models (e.g. climate models)
- **What about within one application? If :**
  - Need more capacity than DRAM
  - Need more bandwidth or lower latency than SSDs/Flash
  - Can afford it
- **Currently a niche, later may be everybody**
- **Today's use case: certain types of in-memory databases**
- **If price becomes lower than DRAM, then it's a win**

Flash Memory

- Solid-state memory invented in the 1980s by Toshiba
- Two types (NAND and NOR).
- Block addressable, like disk (not suited to memory)
- Any particular cell can be accessed O(1000) times only!
- Prices versus disk have plummeted recently
- Flash gaining traction in HPC as primary storage or as faster storage tier
- Expect to see no spinning disks in HPC systems very soon

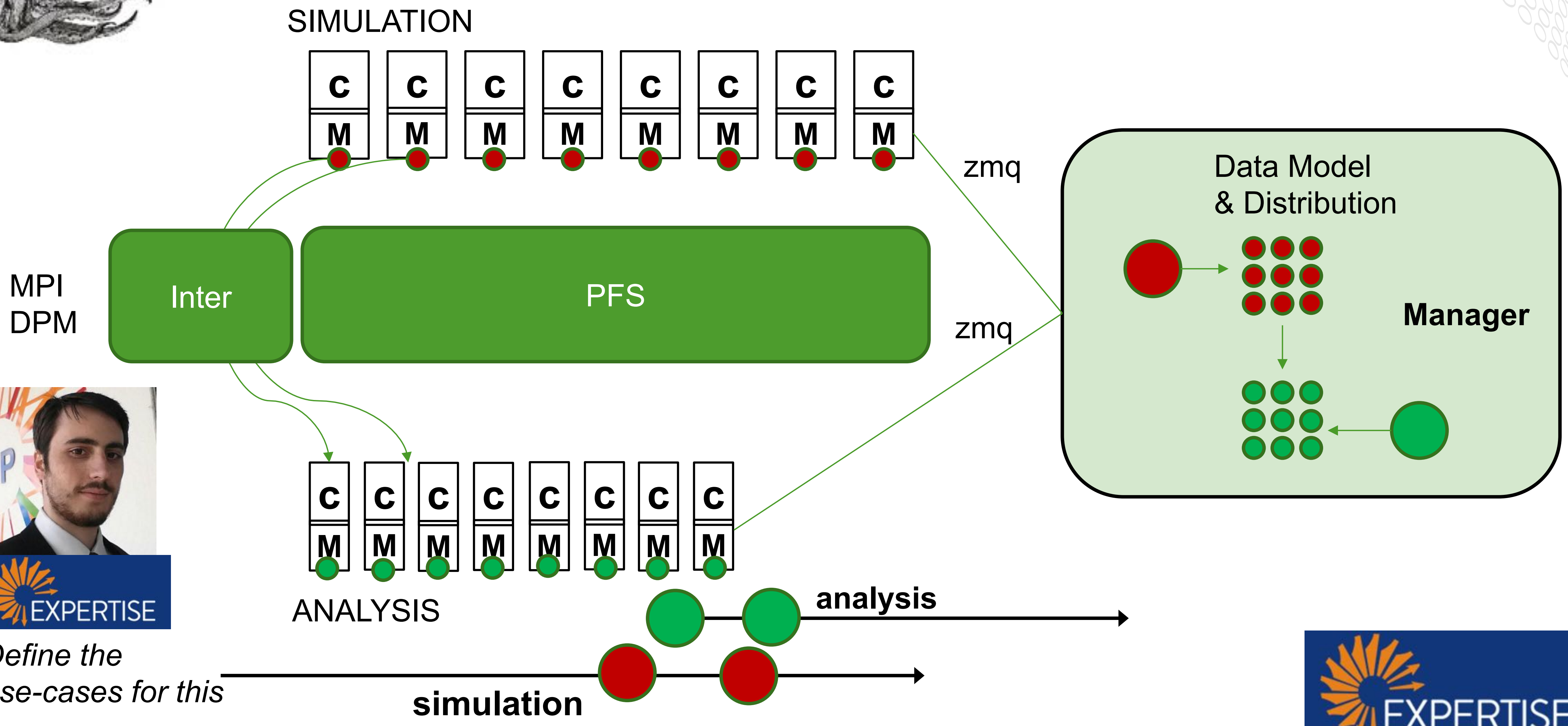
Network attached NVRAM/SSD

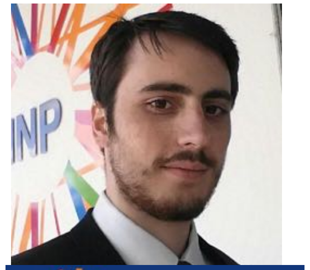
- Accessing local NV-RAM or SSD as storage cache is v. hard
  - Have consistency and coherence costs
- Emerging concept is to use same technologies as fast storage cache
- First demonstrated as burst-buffer
- Now being used for more general I/O forms
- Cray DataWarp



CRAY DATAWARP

Octopus (research project at Cray)





Define the use-cases for this

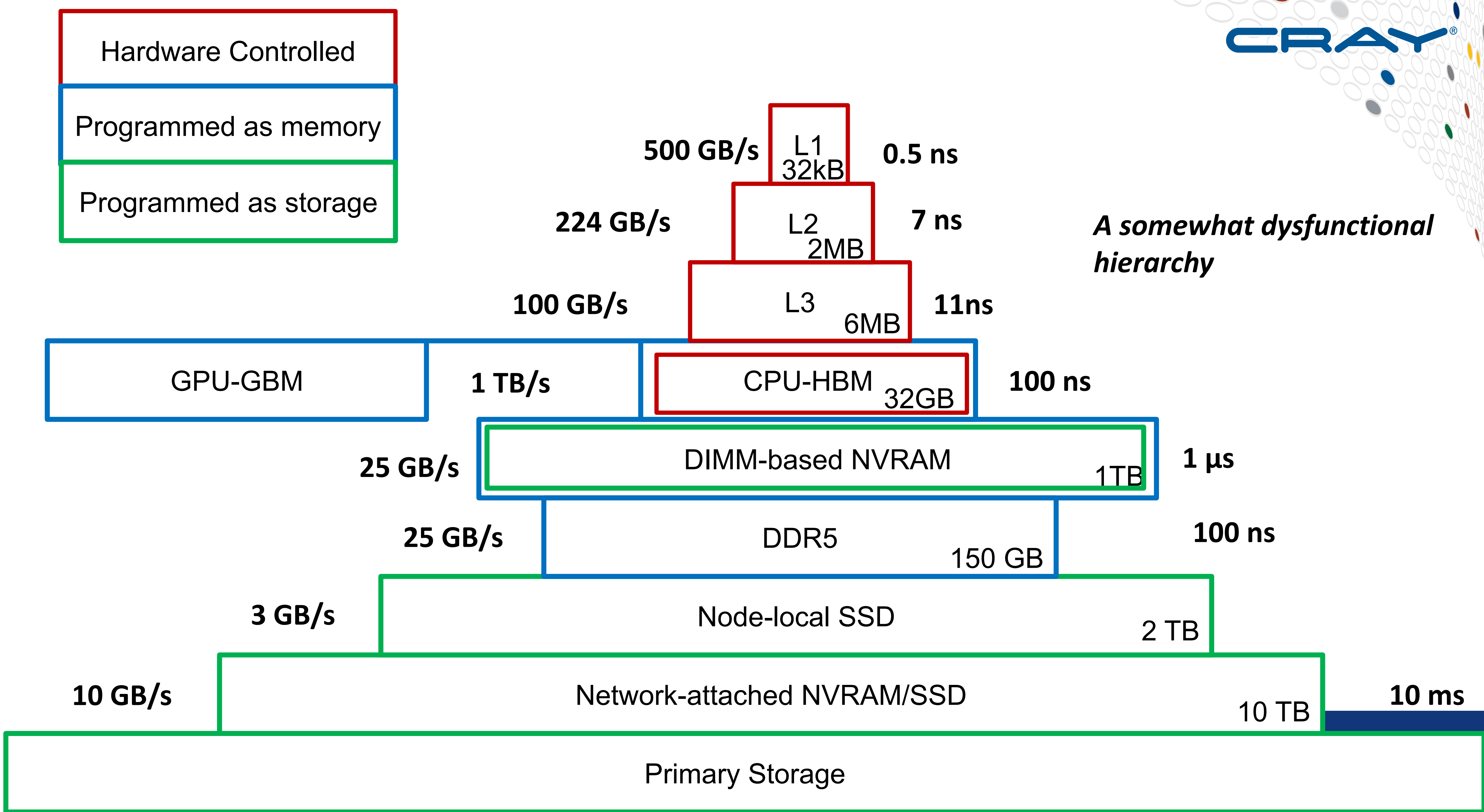
Is this a still hierarchy?

- Latency, bandwidth, capacity numbers of the new memories are

- Also we must consider GPU memory (not covered here)
  - Separate address space

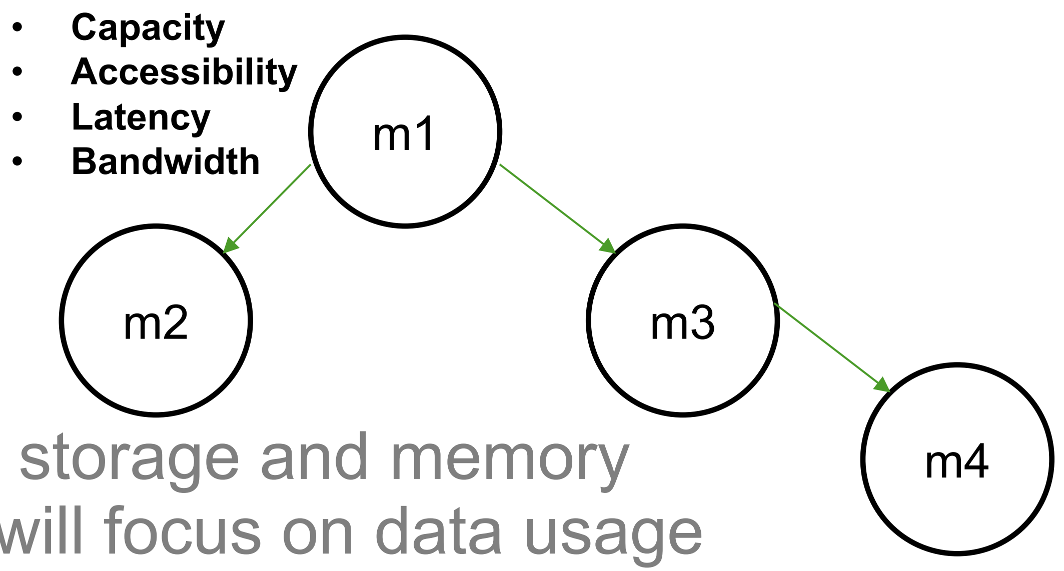
- Does the hierarchy still hold up?

A somewhat dysfunctional hierarchy



The future

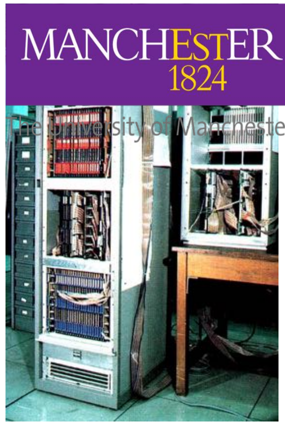
- Memory hierarchy as a concept is not useful anymore
- Need a graph-based set of connected memories + attribute
- NUMA effects compound this



- Future Memory programming:
  - There will be no distinction between storage and memory
  - A new wave of algorithmic analysis will focus on data usage
  - Programming models will allow much more data expression
- Future processors will avoid the von Neumann Bottleneck altogether
  - Dataflow processors will be resurrected!

More info

- **3dXpoint**
  - https://www.intel.com/content/www/us/en/architecture-and-technology/intel-optane-technology.html (warning: vile marketing)
  - **3D XPoint memory – NAND flash killer or DRAM replacement?** https://www.computerworld.com/article/3194147/data-storage/faq-3d-xpoint-memory-nand-flash-killer-or-dram-replacement.html
  - ***SNIA programming language*** https://www.snia.org/tech_activities/standards/curr_standards/npm
  - ***Libpmem*** http://pmem.io/pmdk/
- **HBM**
  - https://www.amd.com/en/technologies/hbm
  - http://meseec.ce.rit.edu/551-projects/fall2016/1-4.pdf Ketan Reddy and Tyler Krupicka
  - ***JEDEC HBM standard*** https://www.jedec.org/standards-documents/docs/jesd235a
- **Cray Octopus**
  - See upcoming paper at Cray User Group 2018
- **Dataflow processors**
  - ***The Manchester Dataflow Computer*** https://pdfs.semanticscholar.org/a81e/ccf02dc85e08fb34212ae56842366fe38678.pdf



EXPERTISE

- Please feel free to contact me for more in-depth discussion
- adrian@cray.com
- www.cray.com/cerl