



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Heterogeneous Memory Systems

# Before we start ...

- ⌘ This course is for you ...
- ⌘ I will go fairly fast over the material
  - Be free to interrupt me anytime. This course is for you, remember? ;-)
  - I am in a full-screen mode: I do not see chat → Speak-up
- ⌘ Each a couple of slides I will ask for your questions and comments
  - Silence means
    - Everything is super clear <OR> You could not care less about the topic
    - Either way I will keep moving
- ⌘ I can provide you the slides
  - If you are interested in the topic, I would recommend you to check the pointers

# Heterogeneous memory systems

« Why?

« What is a problem that you want to solve?

- Memory wall
- Memory latency
- Memory capacity
- Memory bandwidth
- Memory cost
- Memory power/energy
- End of DRAM scaling
- DRAM refresh
- DRAM volatility (non-persistency)
- Resilience
- ...

« Does the problem exist?

- Can we quantify it?

# Problem: Memory cost

⌋ Cost of the HPC server

⌋ Cost of the HPC DIMM

Lenovo Data Center Solution Configurator Quote				
Prepared for:		Prepared by:		
Your final configuration may contain hardware, software, and services; therefore, accounting implications need to be taken into consideration. A bottom line price for the package/bundle should only be presented with accounting approval.		Price Date:		19-Feb-19
		Quote date:		Quote Expiration Date:
Part number	Product Description	Qty	Price (per unit) US Dollar	Total Part Price (quantity x unit price) US Dollar
7X21A012NA	Node : SD530, Xeon Gold 6140 18C 2.3GHz, 1x16GB (2Rx8 1.2V) RDIMM, 1xSAS/SATA/NVMe 3x2 Bay <b>You will have to manually add:</b>	1	\$ 9,799.00	\$ 9,799.00
7XG7A06228	ThinkSystem SD530 Intel Xeon Gold 6140 18C 140W 2.3GHz Processor Option Kit	1	\$ 3,819.00	\$ 3,819.00
7X77A01301	ThinkSystem 8GB TruDDR4 2666 MHz (1Rx8 1.2V) RDIMM	12	\$ 379.00	\$ 4,548.00
00WE027	Intel OPA 100 Series Single-port PCIe 3.0 x16 HFA	1	\$ 959.00	\$ 959.00
5AS7A02047	Hardware Installation Server (Business Hours)	1	\$ 539.00	\$ 539.00
5PS7A06901	Premier with Essential - 3Yr 24x7 4Hr Response + YourDrive YourData <b>You will have to manually remove:</b>	1	\$ 1,166.00	\$ 1,166.00
7X77A01303	ThinkSystem 16GB TruDDR4 2666 MHz (2Rx8 1.2V) RDIMM(Standard)	1	\$ -	\$ -
			<b>Total</b>	<b>\$ 20,830.00</b>

⌋  $\Sigma \text{ DIMMs@Server [€] / Server[€] = ?$

– 17% in our calculations (an example)

⌋  $\Sigma \text{ Server[€] / Total system [€] = 40 / 50 / 60 / 70\% ?$

– Building, interconnect, storage, cooling, power supply, sys admins, etc.

# Solving the memory cost problem

## 1 INTRODUCTION

The memory system is a major contributor to the deployment and operational costs of a large-scale high-performance computing (HPC) cluster [25, 35, 38], and in terms of system performance it is one of the most critical aspects of the system's design [21, 41]. For decades, most server and HPC cluster memory systems have been based on

☞ Your favorite proposal:

*“... decrease the cost of the memory system by **20%** ...”*

*“.. while the performance loss is between **1%** and **2%**...”*

☞ Decrease of the total cost of ownership

20% *“... decrease the cost of the memory system by **20%** ...”*

\* 17% DIMMs/Server cost

\* 50% Servers/Rest of the system

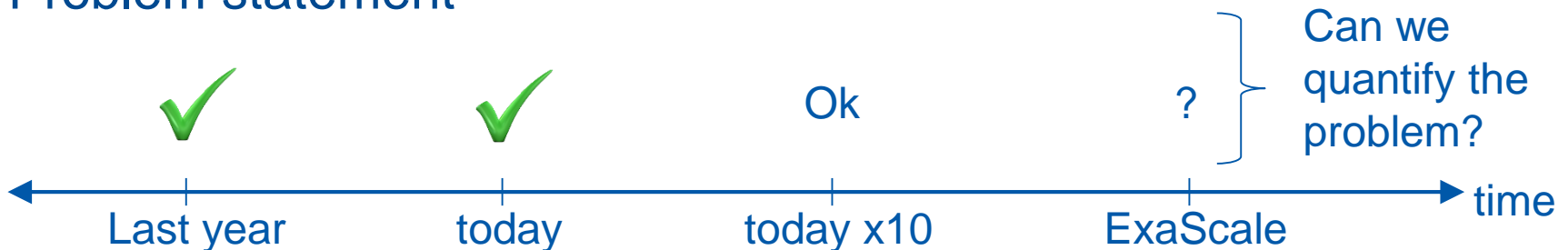
→ **1.7%**

☞ And the investment to reprogram all the workloads, OS, runtime?

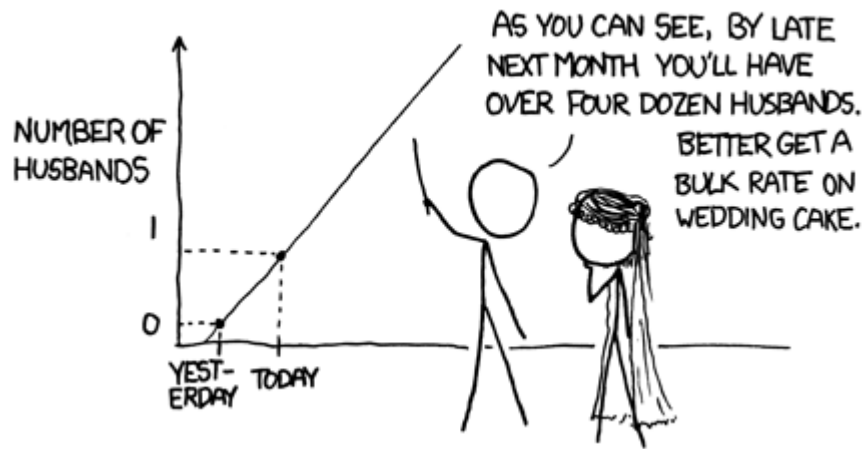
# Memory cost @ ExaScale

- ⌘ “... 50% of the overall system cost ...”
- ⌘ How did they compute this?
- ⌘ Assumptions?
- ⌘ What is “ExaScale”?
  - 1EFLOP (Running what? HPL?)
  - Performance of current systems x1000
  - “Democratization” of the HPC
    - HPC @ Cloud
    - HPC @ SMEs
    - HPC @ Hospitals

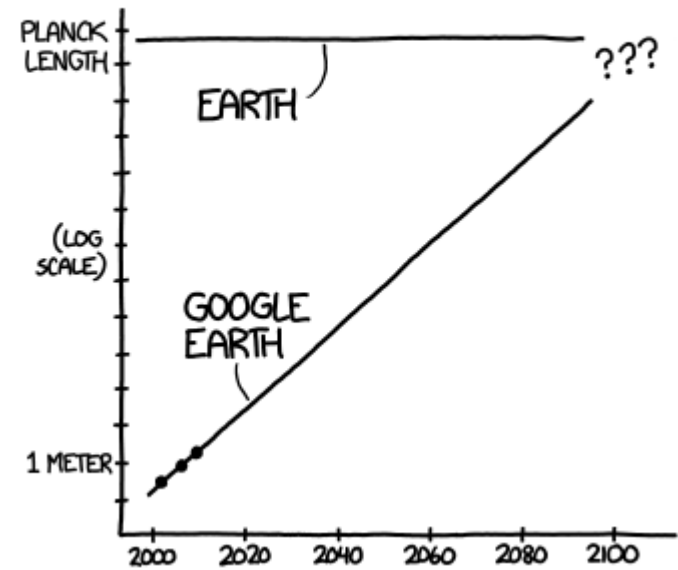
## ⌘ Problem statement



# Beware of extrapolation



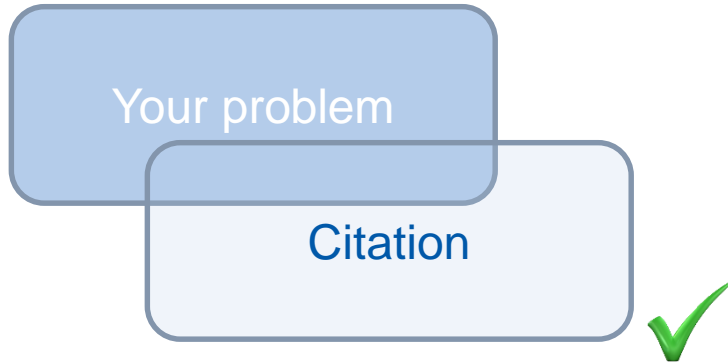
MY NEIGHBORHOOD'S RESOLUTION IN:



Your problem

## Problem statement: Take care!

- Supporting a statement with a citation is not difficult
  - Think about the context



?

- Looking into future: Beware of ...
  - Assumption
  - Extrapolations

Citation

# Problem: Memory power/energy

## ⌘ Analogous to the cost

- $\Sigma \text{ DIMMs@Server [Watts]} / \text{Server[Watts]} = ?$
- $\Sigma \text{ Server[Watts]} / \text{Total system [Watts]} = 40 / 50 / 60 / 70\% ?$ 
  - Building, interconnect, storage, cooling, power supply, sys admins, etc.
  - ...

# Memory wall

## Memory wall refers to:

- Memory latency
- Memory capacity
- Memory bandwidth
- Memory cost
- Memory power/energy
- End of DRAM scaling
- DRAM refresh
- DRAM volatility (non-persistency)
- Resilience
- ...
- All the above?

# Memory wall

## Hitting the Memory Wall: Implications of the Obvious

Wm. A. Wulf  
Sally A. McKee

Department of Computer Science  
University of Virginia  
{wulf|mckee}@virginia.edu

December 1994

This brief note points out something obvious — something the authors “knew” without really understanding. With apologies to those who did understand, we offer it to those others who, like us, missed the point.

We all know that the rate of improvement in microprocessor speed exceeds the rate of improvement in DRAM memory speed — each is improving exponentially, but the exponent for microprocessors is substantially larger than that for DRAMs. The difference between diverging exponentials also grows exponentially; so, although the disparity between processor and memory speed is already an issue, downstream someplace it will be a much bigger one. How big and how soon? The answers to these questions are what the authors had failed to appreciate.

To get a handle on the answers, consider an old friend — the equation for the average time to access memory, where  $t_c$  and  $t_m$  are the cache and DRAM access times and  $p$  is the probability of a cache hit:

$$t_{avg} = p \times t_c + (1 - p) \times t_m$$

We want to look at how the average access time changes with technology, so we'll make some conservative assumptions; as you'll see, the specific values won't change the basic conclusion of this note, namely that we are going to hit a wall in the improvement of system performance unless something *basic* changes.

First let's assume that the cache speed matches that of the processor, and specifically that it scales with the processor speed. This is certainly true for on-chip cache, and allows us to easily normalize all our results in terms of instruction cycle times (essentially saying  $t_c = 1$  cpu cycle). Second, assume that the cache is perfect. That is, the cache never has a conflict or capacity miss; the only misses are the compulsory ones. Thus  $(1 - p)$  is just the probability of accessing a location that has never been referenced before (one can quibble and adjust this for line size, but this won't affect the conclusion, so we won't make the argument more complicated than necessary).

Now, although  $(1 - p)$  is small, it isn't zero. Therefore as  $t_c$  and  $t_m$  diverge,  $t_{avg}$  will grow and system performance will degrade. In fact, it will hit a wall.

— 20 —

In 1995, Wulf and McKee published a four-page note entitled “Hitting the Memory Wall: Implications of the Obvious” in the (unrefereed) ACM SIGARCH *Computing Architecture News* [27]. The motivation was simple: at the time, researchers were so focused on improving cache designs and developing other latency-tolerance techniques that the computer architecture community largely ignored main memory systems. The article projected the performance impact of the increasing speed gap between processors and memory. The study predicted that if the trends held, even with cache hit rates above 99%, relative memory latencies would soon be so large that the processor would essentially always be waiting for memory — which amounts to “hitting the wall”.

Assignment:  
Read this paper.  
**READ IT CAREFULLY**

# Do high-bandwidth memories (HBM, MCDRAM, HMC) break through the memory wall?

logic and DRAM layers into one optimized 3D package that leverages through-silicon via (TSV) technology. The result is a category of memory unlike anything on the market today.

HMC performance levels break through the memory wall, delivering a high-bandwidth, energy-efficient, high-density memory system that will enrich next-generation networking and drive significant reductions in data center and supercomputing power consumption.

**INCREASED BANDWIDTH** Provides up to 15X the bandwidth of a DDR3 module

**REDUCED LATENCY** Enables significantly lower system latency as a result of HMC's massive parallelism

**SMALLER SIZE** Reduces the memory footprint by nearly 90% compared to today's RDIMMs due to HMC's stacked architecture

**SCALABLE** Includes logic layer flexibility, which enables HMC to be tailored to multiple platforms and applications

⌋ Plug & Play → All your problems solved

# Puzzle

## ⌘ Memory wall:

- Perf =  $f$ (memory latency)

## ⌘ HBMs will increase memory bandwidth

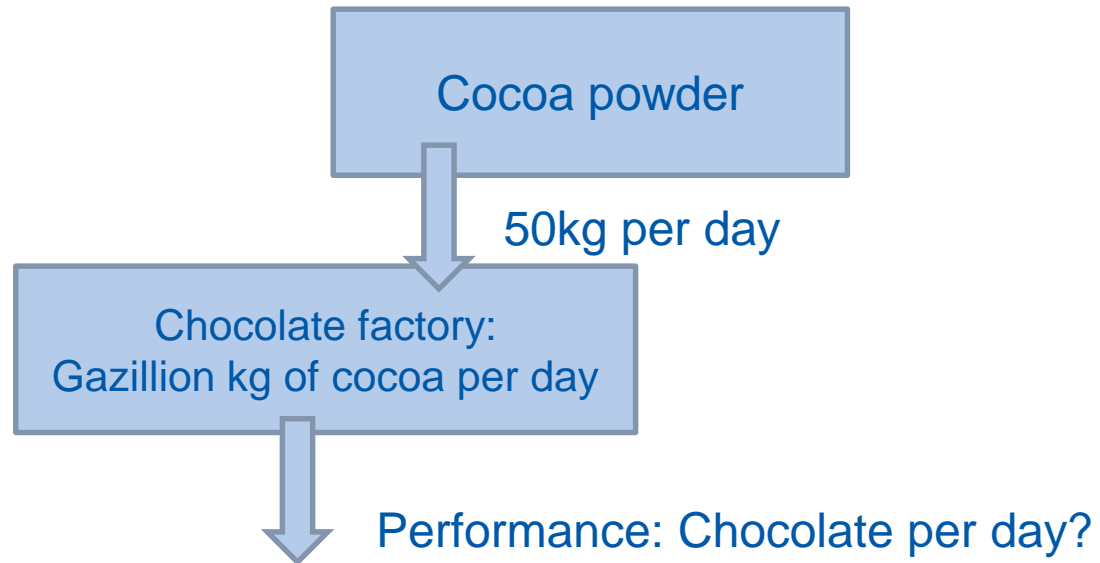
## ⌘ Something is missing

- Performance =  $f$ (memory bandwidth, memory latency)

# Performance = $f$ (bandwidth)

## Chocolate factory

- Chocolate = cocoa (100% dark chocolate)
- The machines can process Gazillion kg of cocoa/chocolate per day

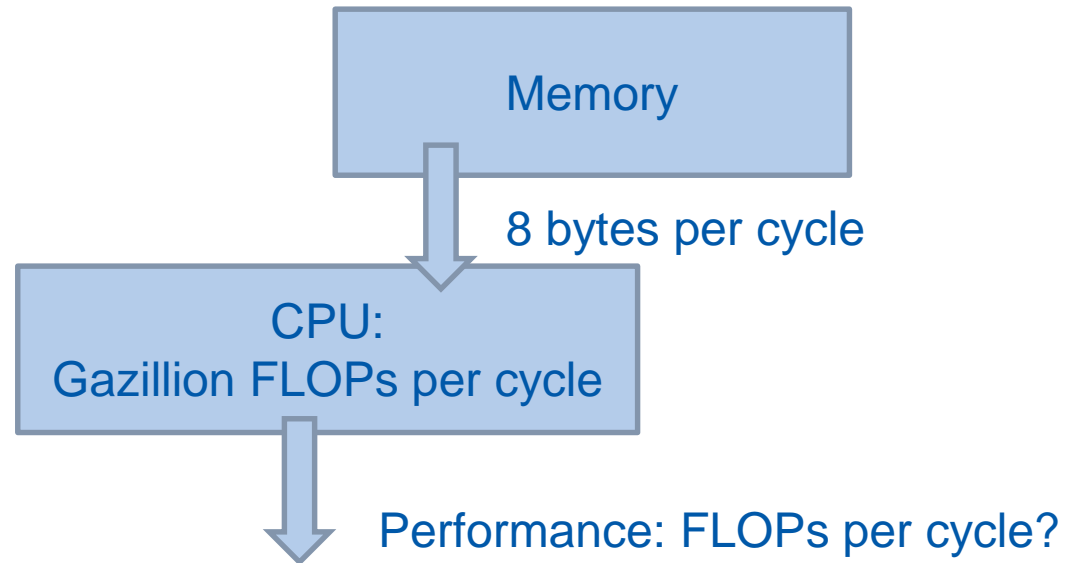


- The cocoa powder bandwidth limits your performance

# Performance = $f$ (bandwidth)

## ⌘ Back to the computer science

- CPU can perform Gazillion FP operations per cycle (CPU cycle)
- Memory bandwidth: 8 bytes per CPU cycle



- Memory bandwidth may limit your performance

# Roofline model

## Characterization of applications

### ⌘ Operational intensity

- Ratio between application's stress to CPU and stress to memory

$$\text{Operational intensity} \left[ \frac{\text{Flops}}{\text{Byte}} \right] = \frac{\text{Performance} \left[ \frac{\text{Flops}}{\text{Second}} \right]}{\text{Memory bandwidth} \left[ \frac{\text{Bytes}}{\text{Second}} \right]}$$

### ⌘ Intuition behind operational intensity

- Low Operational Intensity
  - Small computation per amount of data → Stress to memory
- High Operational Intensity
  - Significant computation per amount of data → Stress to CPU

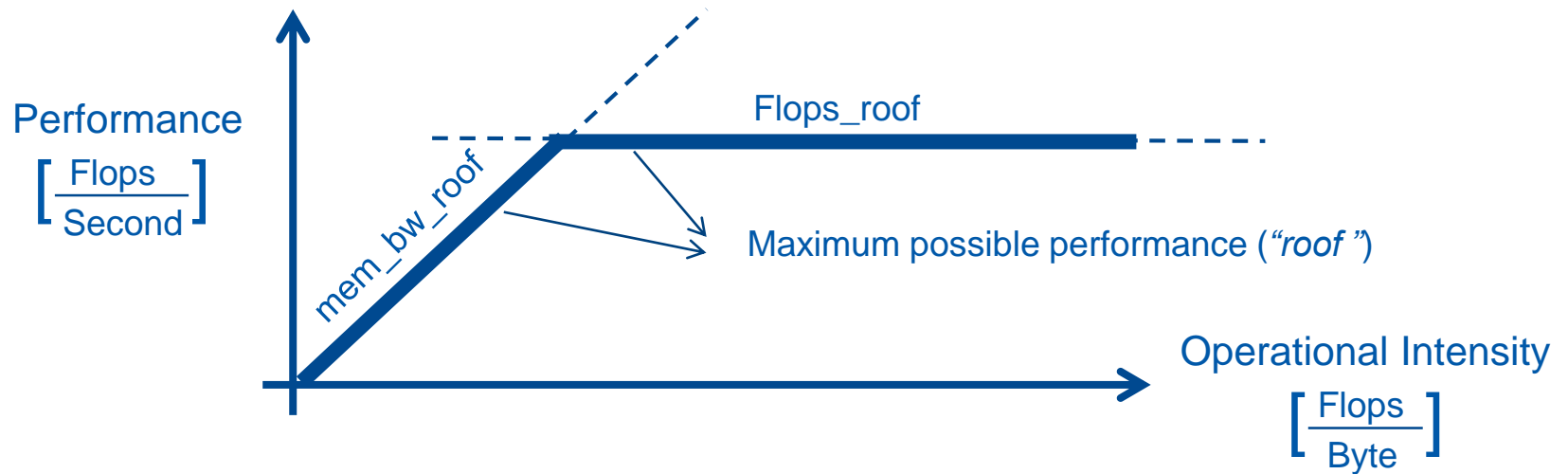
\* Flops: Floating point operations

# Roofline model

## Characterization of architecture (HW)

### Roofline model plots

Max possible performance =  $f$  (Operational intensity)

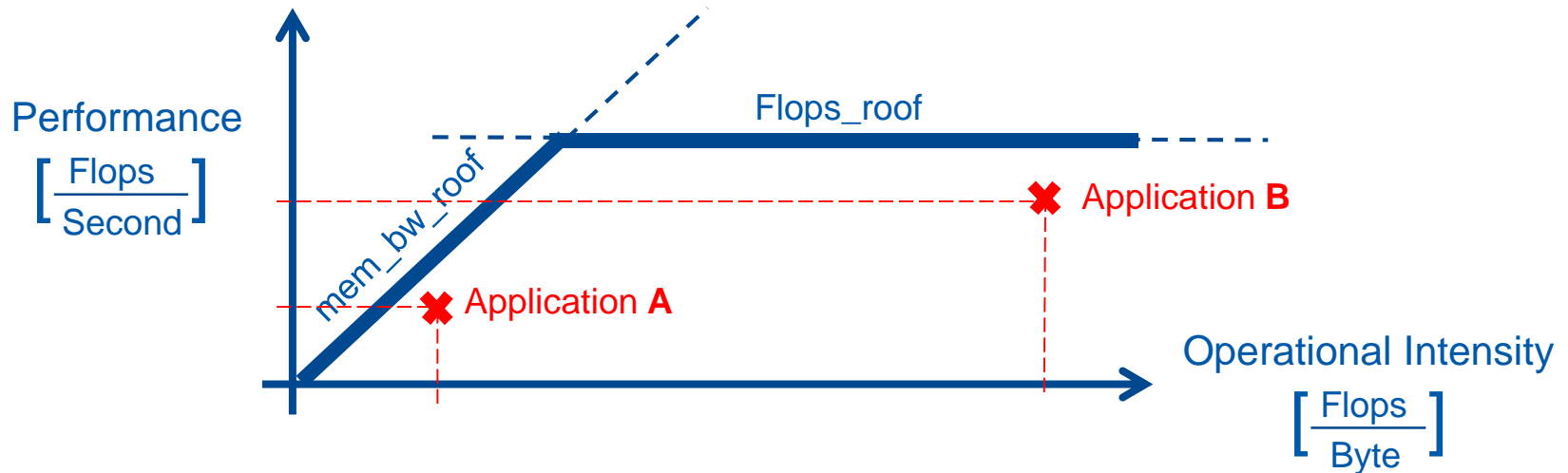


### Performance of a given architecture (HW) limited by:

- Compute units: Flops\_roof
- Memory bandwidth: mem\_bw\_roof

# Roofline model

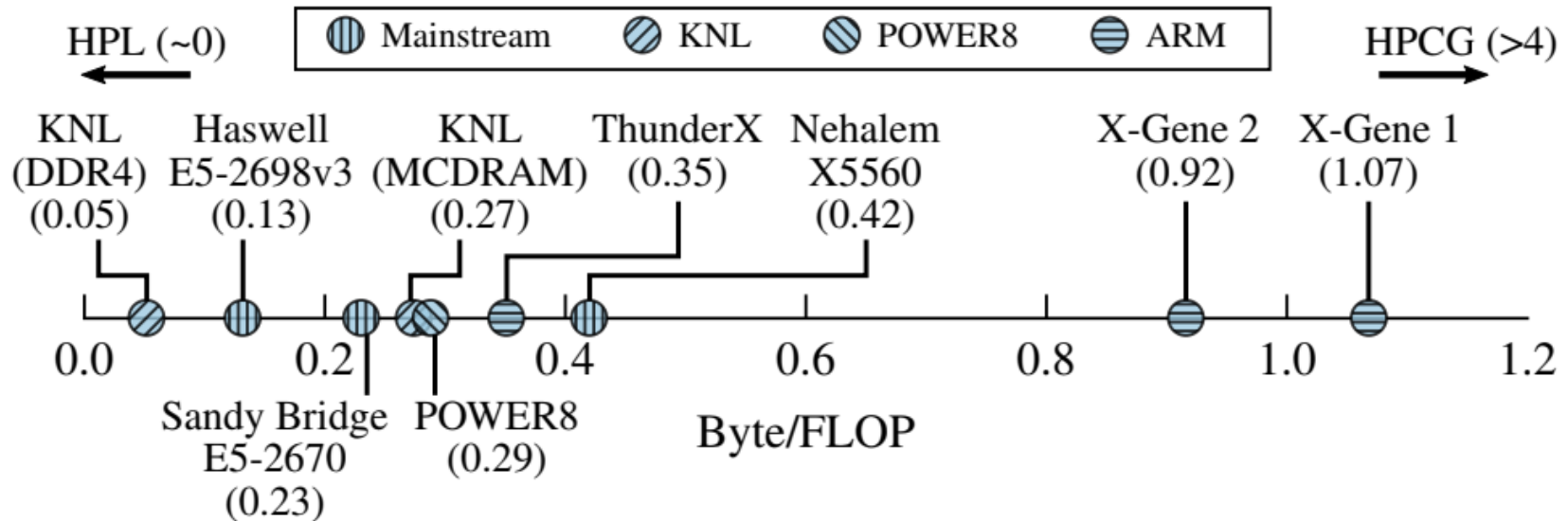
## Characterization of application and architecture



- Application A
  - Low operational intensity: Small amount of computation per amount of data
  - Performance limited by mem\_bw\_roof (memory bandwidth and latency)
- Application B
  - High operational intensity: Significant computation per amount of data
  - Performance limited by Flops\_roof (CPU, execution units)
- Summary: Roofline model provides more insights about
  - Architecture: mem\_bw\_roof and Flops\_roof
  - Applications: Performance is limited by mem\_bw\_roof or Flops\_roof

# Flops and Bytes and vice versa

⌘ HPC applications and platforms today (yesterday?)



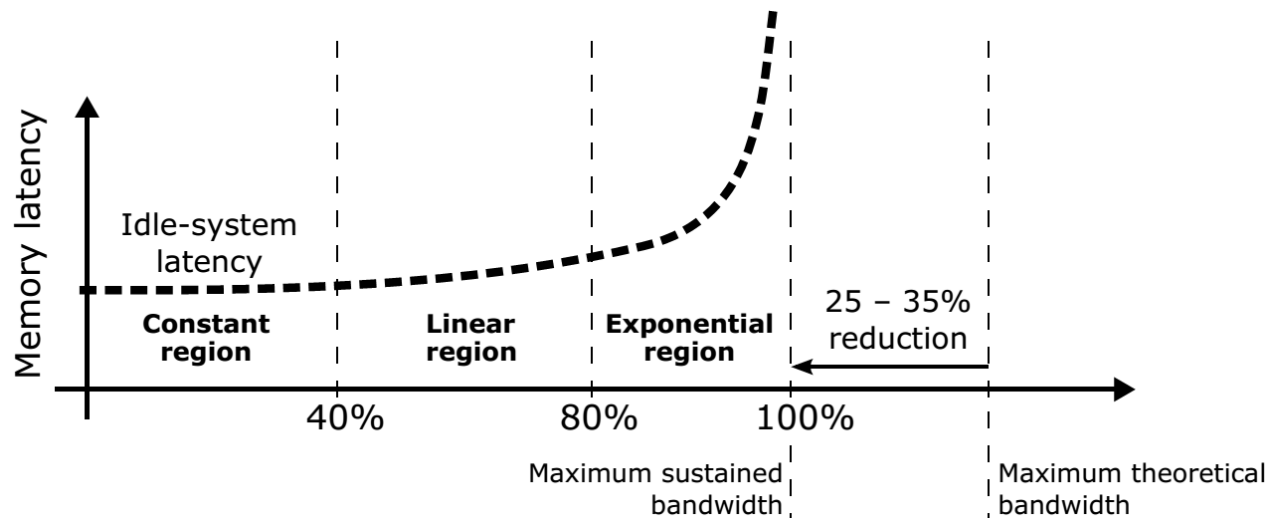
# Performance = $f(\text{bandwidth})$

## Reminder

- Memory wall:
  - Perf =  $f(\text{memory latency})$
- HBMs will increase memory bandwidth

## One more piece of the puzzle

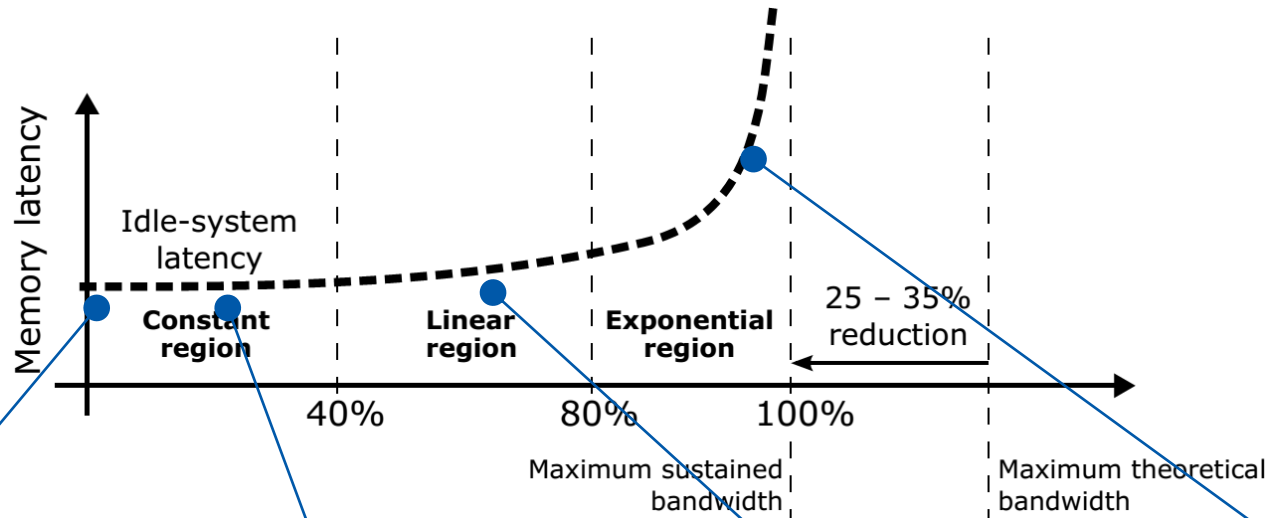
- Memory latency =  $f(\text{used bandwidth})$



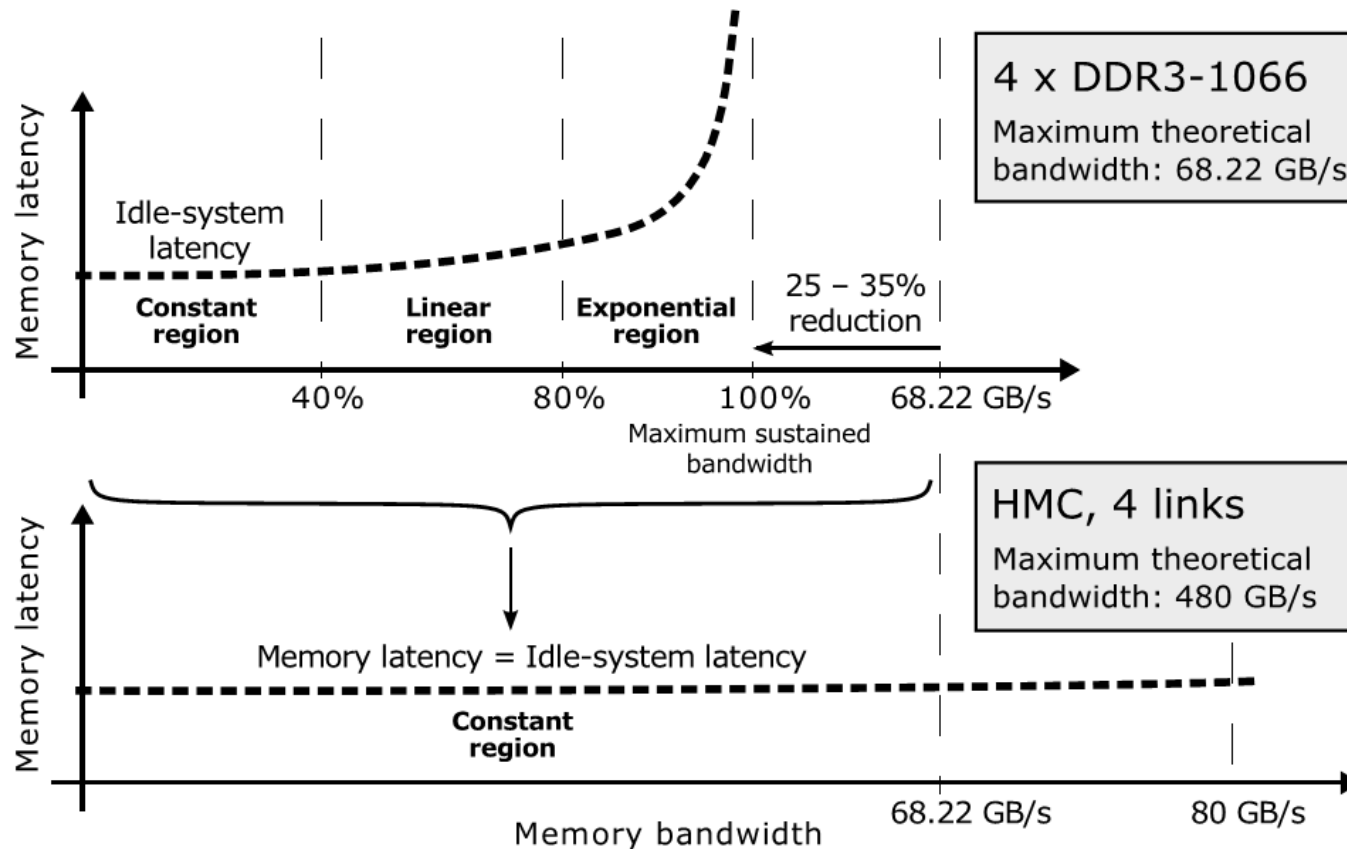
# Performance = $f(\text{bandwidth})$

## « One more piece of the puzzle

- Memory latency =  $f(\text{used bandwidth})$



# What will change from DDRx to HBMs?

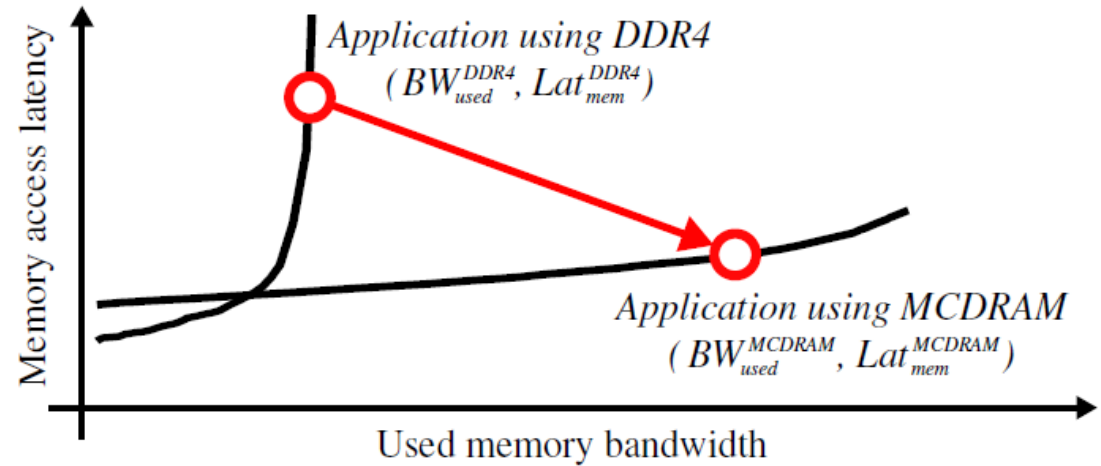


« Significant latency (performance) improvements only if the workloads use significant portion of memory bandwidth

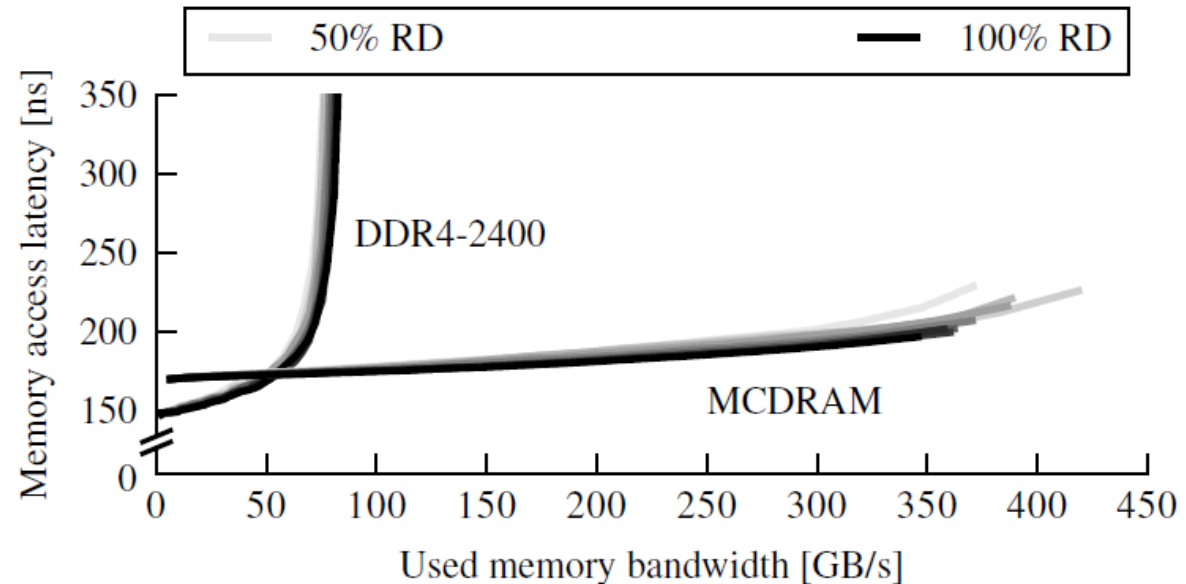
« DDRx vs. HBMs idle-system latency ?!

# Knights Landing: DDR4 vs. MCDRAM

## « The idea

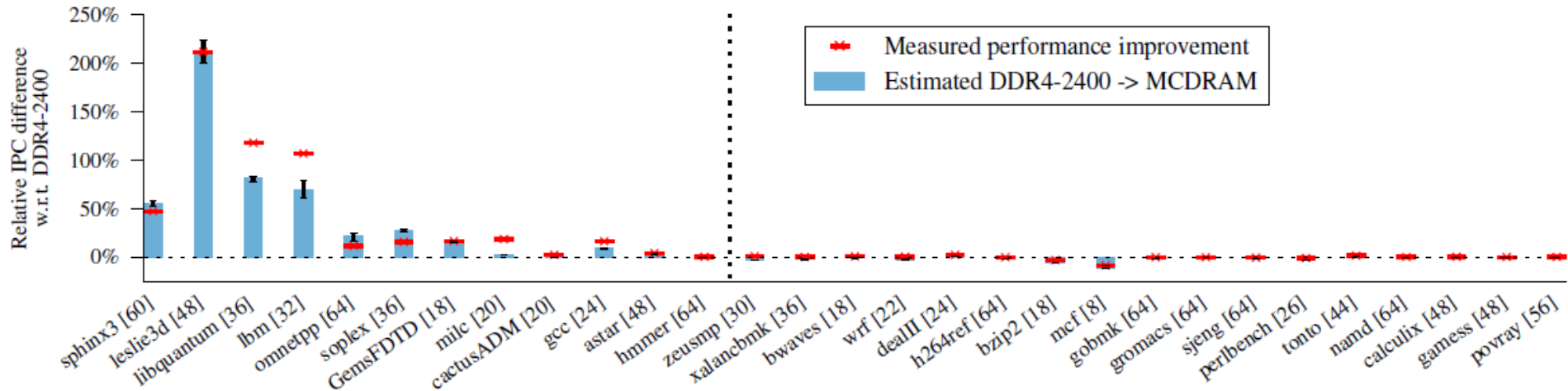
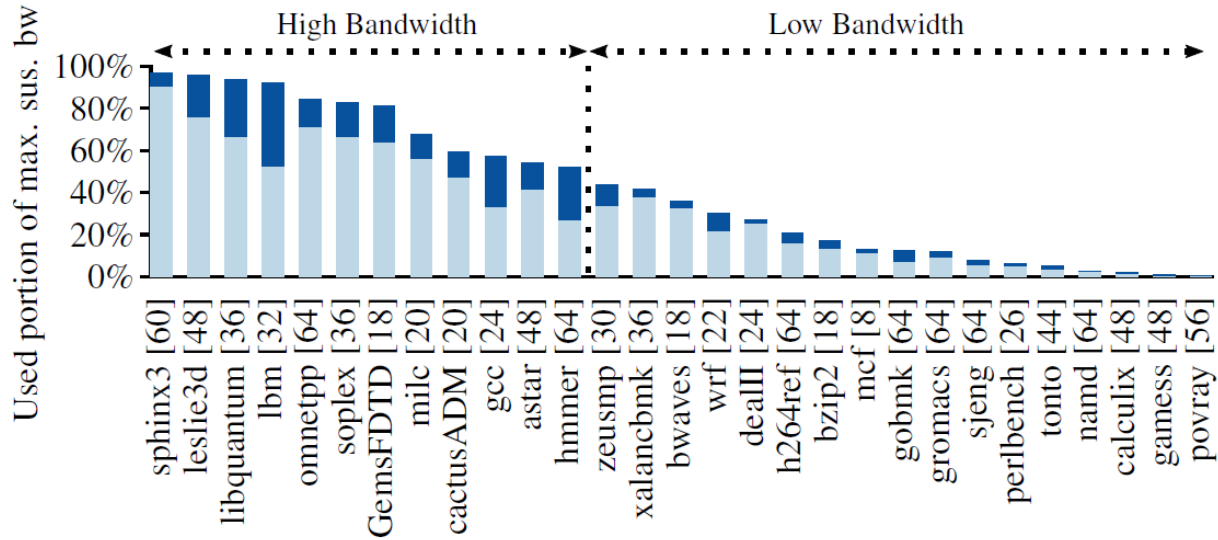


## « Actual curves



# Knights Landing: DDR4 vs. MCDRAM

## Some results



# What about the Optane?

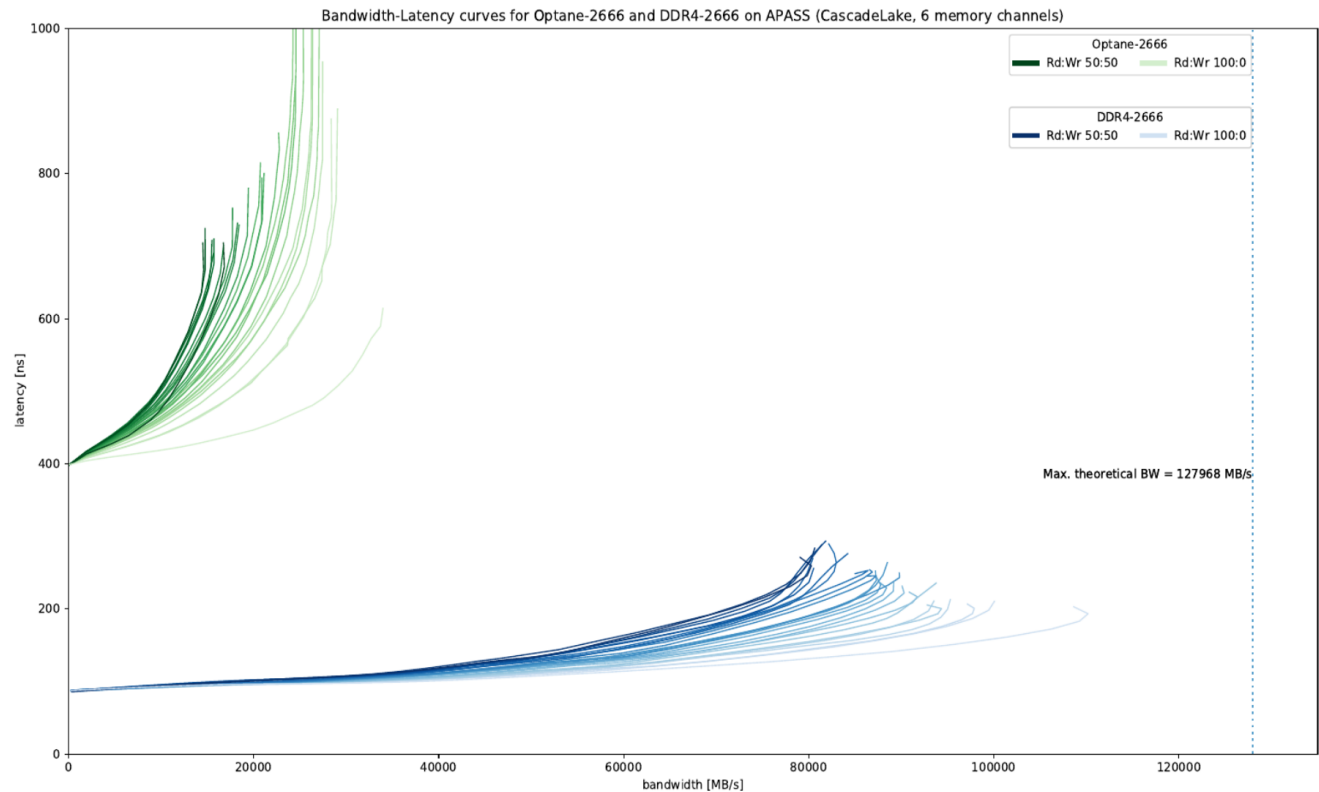
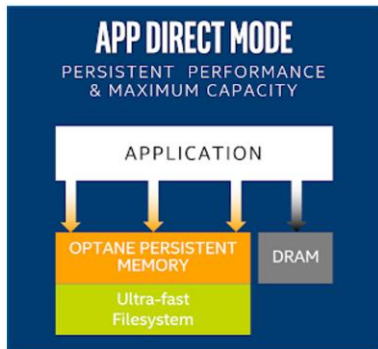
## Experimental platform

### – APASS server:

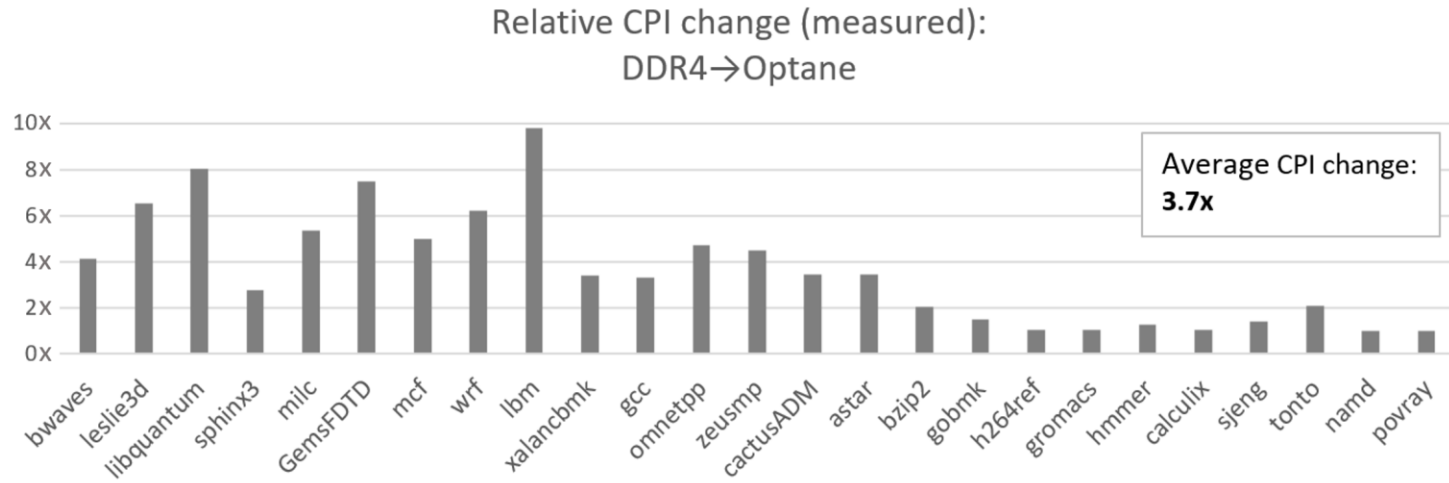
- Motherboard: Intel S2600WFD (dual socket)
- CPU: 2x Intel Xeon 8260 (Cascade Lake).
  - 24 cores. Hyperthreading disabled.
  - Base frequency: 2.4 GHz
  - 6 memory channels
  - Memory:
    - » DDR4 Kingston 8GB 9965690-002.A00G @ 2666 MT/s
    - » Optane Intel 128GB NMA1XXD128GPS @ 2666 MT/s

# Bandwidth-latency curves: DDR4-2666 & Optane

App direct mode (user decides where the data is)

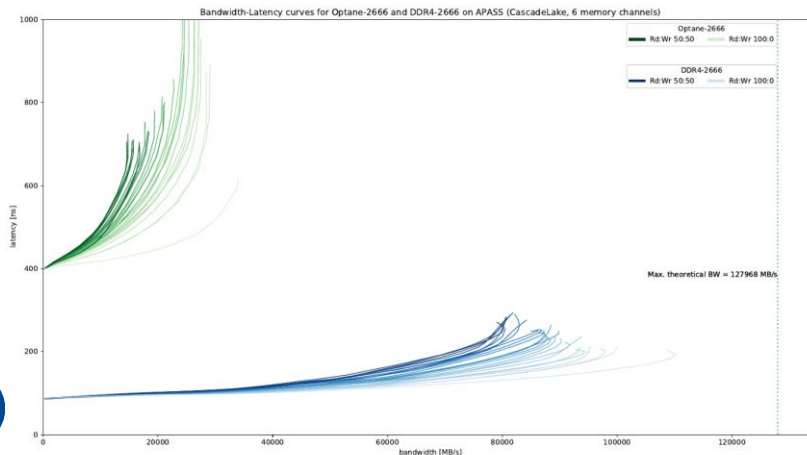
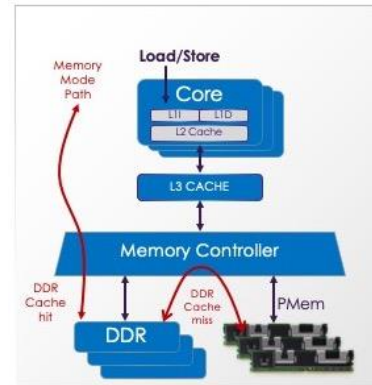
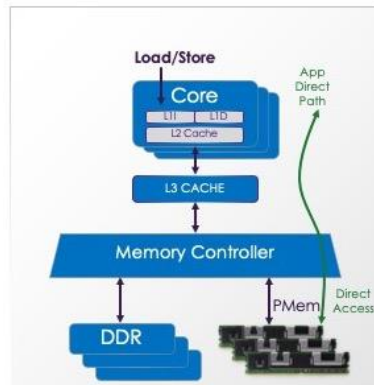
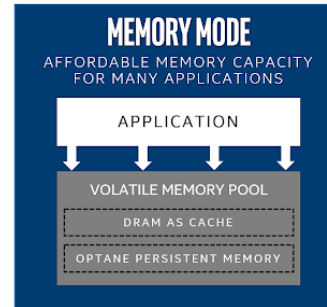
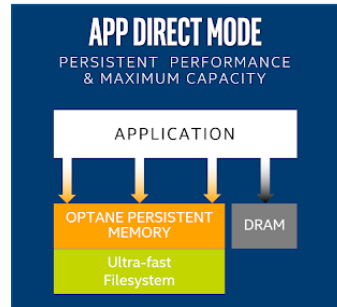


# SPEC benchmarks: DDR4→Optane



- Significant performance difference DDR4 vs. Optane
  - 3.7x on average
  - Up to 9.8x (lbm benchmark)

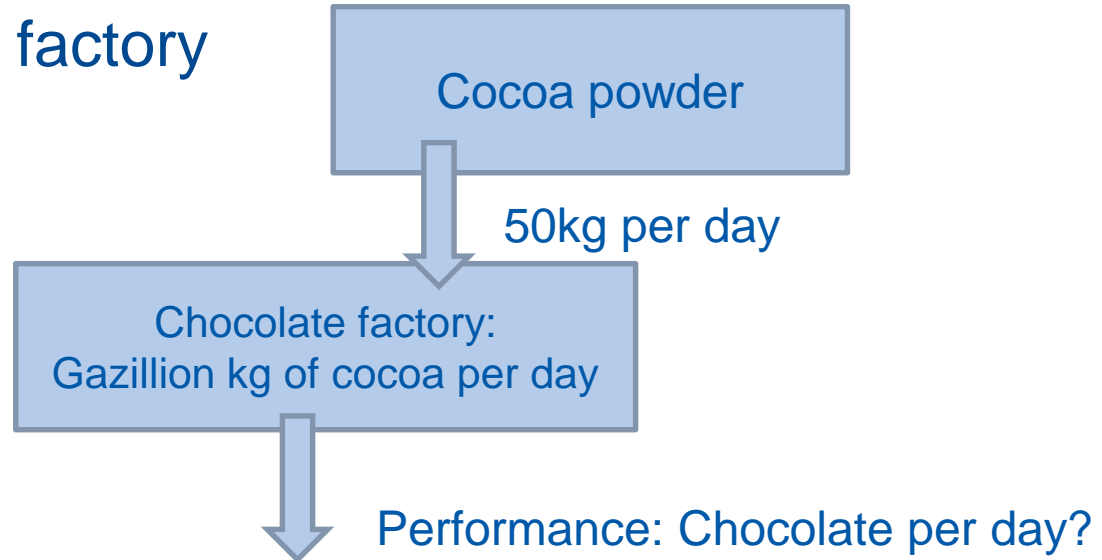
# What will change in the Memory mode?



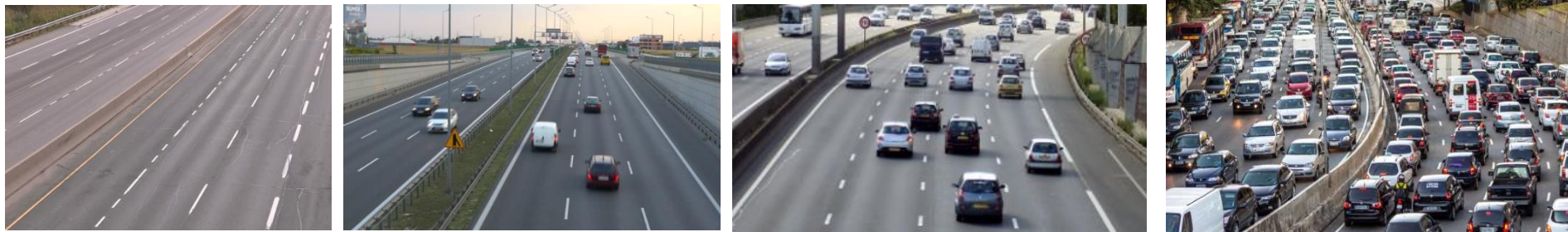
# Take out

⌘ Performance =  $f(\text{memory bandwidth, memory latency})$ ?

⌘ Case 1: Chocolate factory



⌘ Case 2: Driving to work



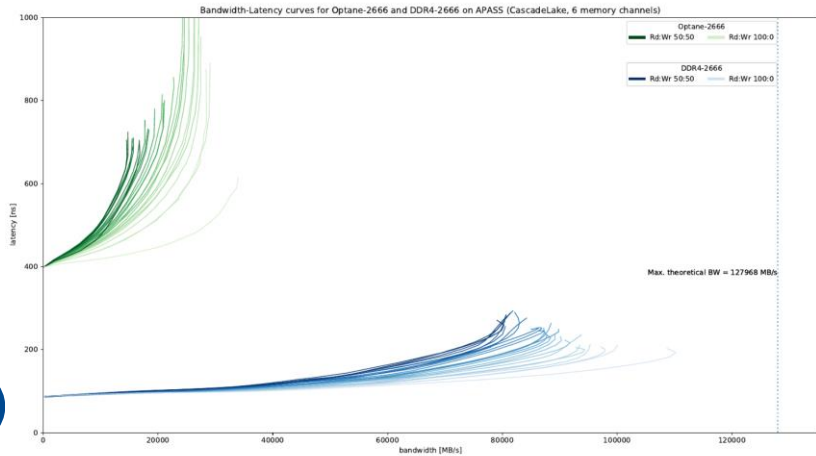
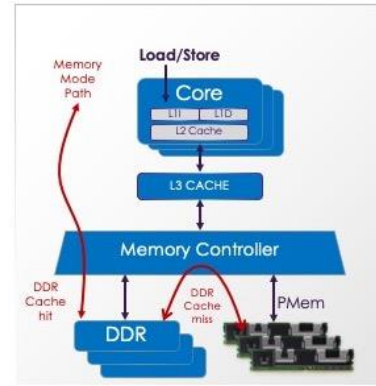
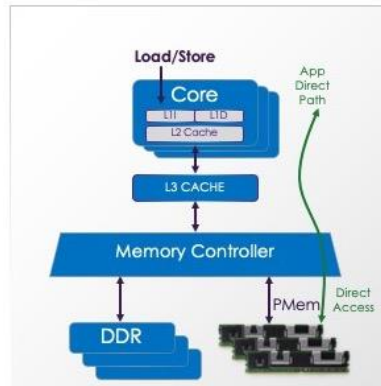
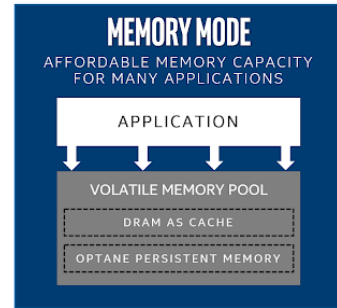
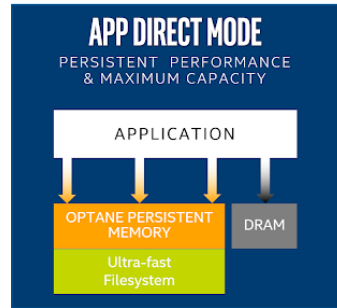
## ⌘ Heterogeneous memory systems

- Why?
- Can you quantify the problem (e.g., cost)?

## ⌘ Puzzle

- Memory wall:
  - Performance =  $f(\text{memory latency})$
- HBMs increases the available memory bandwidth
  - Something is missing
    - Performance =  $f(\text{memory bandwidth, memory latency})$ ?
- Optane
  - Higher latency, lower bandwidth, higher capacity
    - Any performance gain has to come from the higher memory capacity
    - But this is another story:  
Darko Zivanovic et al., *Large-Memory Nodes for Energy Efficient High-Performance Computing*. MEMSYS, 2016. The best paper award.

# Hands-on DDR4 vs. Optane & App direct mode vs. Memory mode





**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

Thank you!

For further information please contact  
[petar.radojkovic@bsc.es](mailto:petar.radojkovic@bsc.es)