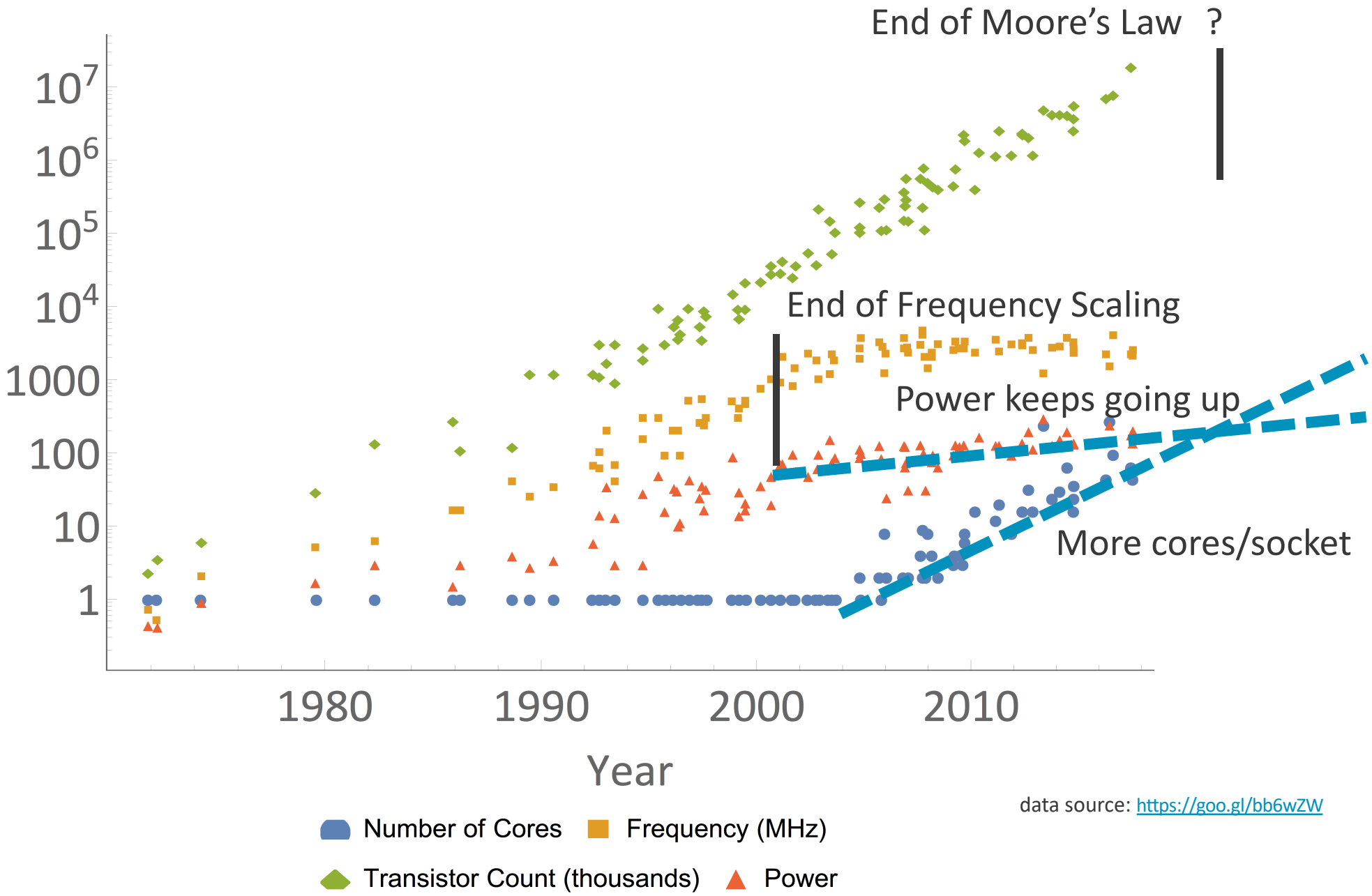
arm Research

Memory Centric High Performance Computing Panel

Jonathan Beard

11 November 2018

Trends



data source: https://goo.gl/bb6wZW

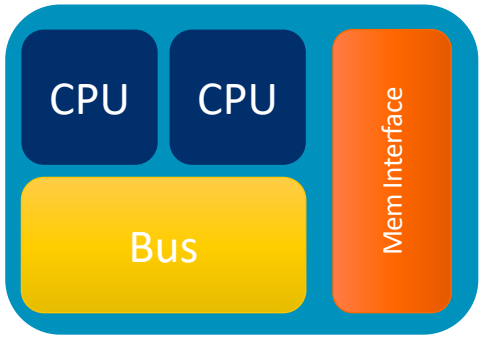
arm Research

Adapted/modified from original figure courtesy of Dilip Vasudevan (LBL)

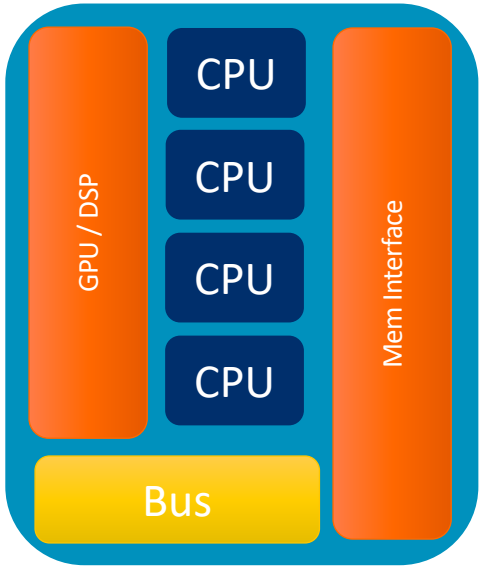
Specialization drives performance…

But at a cost

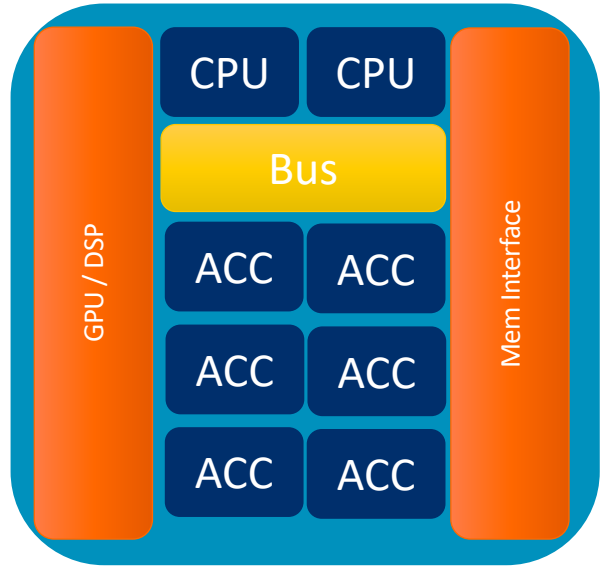
5 years ago



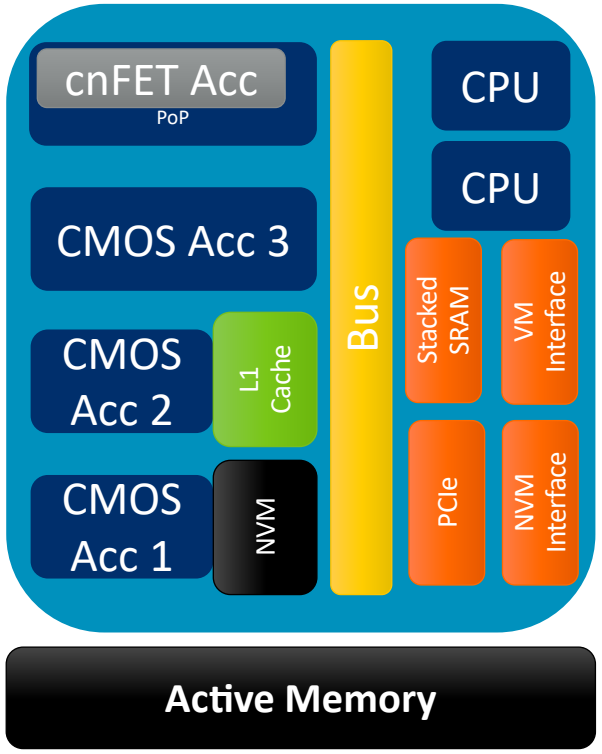
Today



Today – Mobile/Client
0-3 years



3-10 years



Towards Extreme Heterogeneity

Low **Cost to build/adopt/run** **Extreme**

arm Research

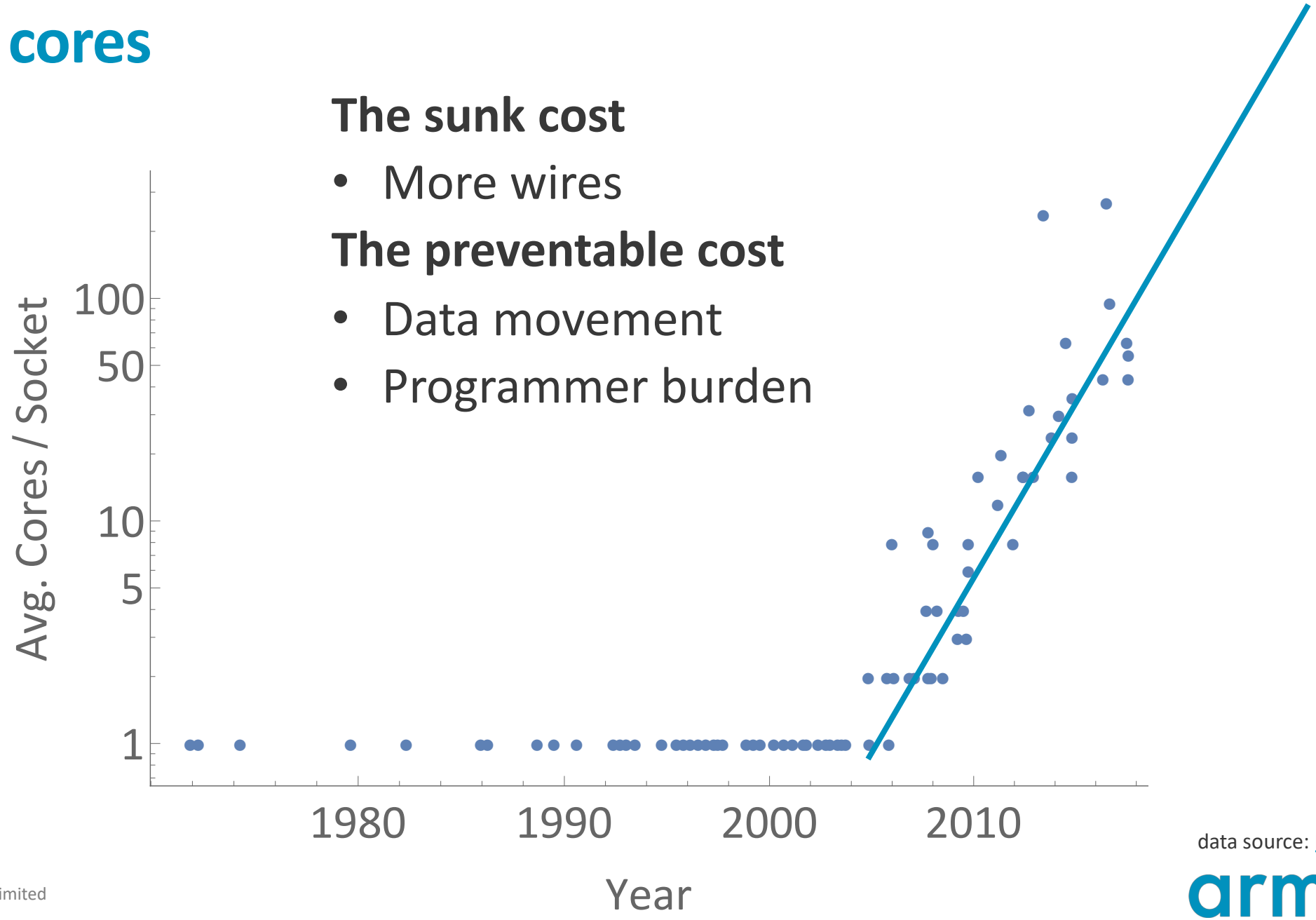
More cores

The sunk cost

- More wires

The preventable cost

- Data movement
- Programmer burden

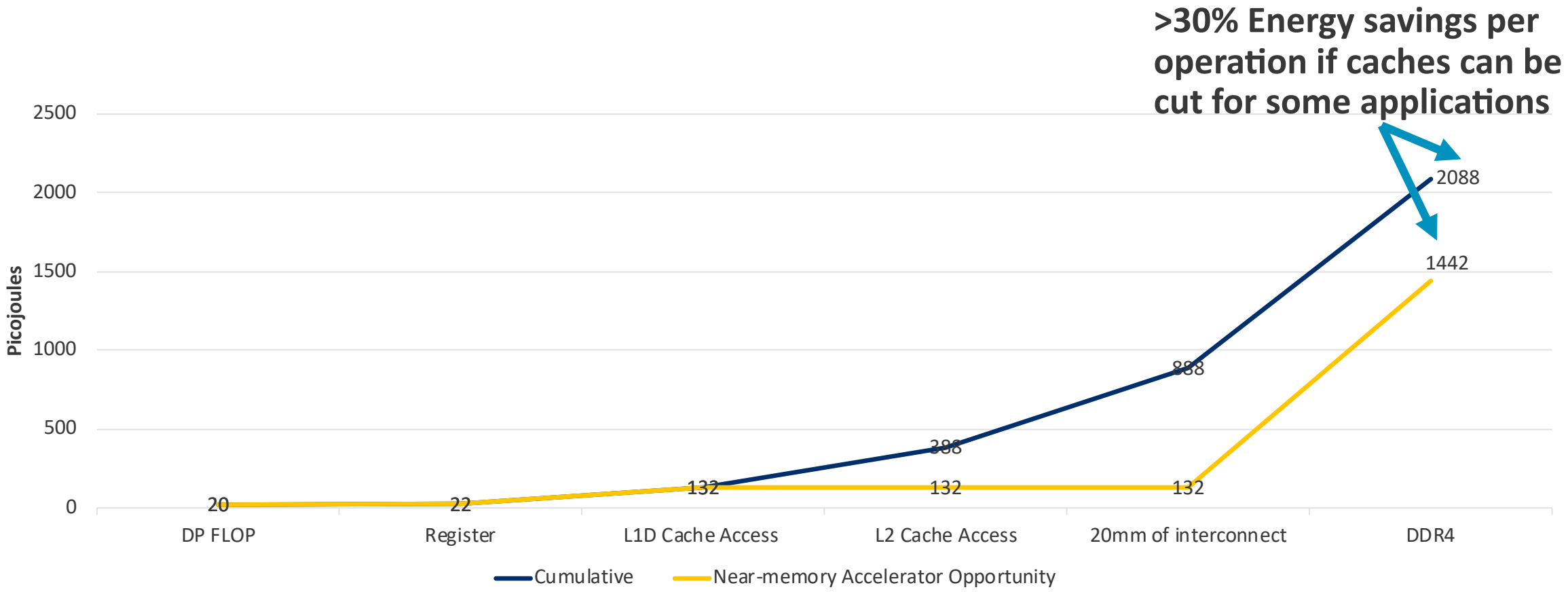


data source: https://goo.gl/bb6wZW

arm Research

Energy of Data Movement

Cost of obtaining 8B from DRAM for a single DP FLOP

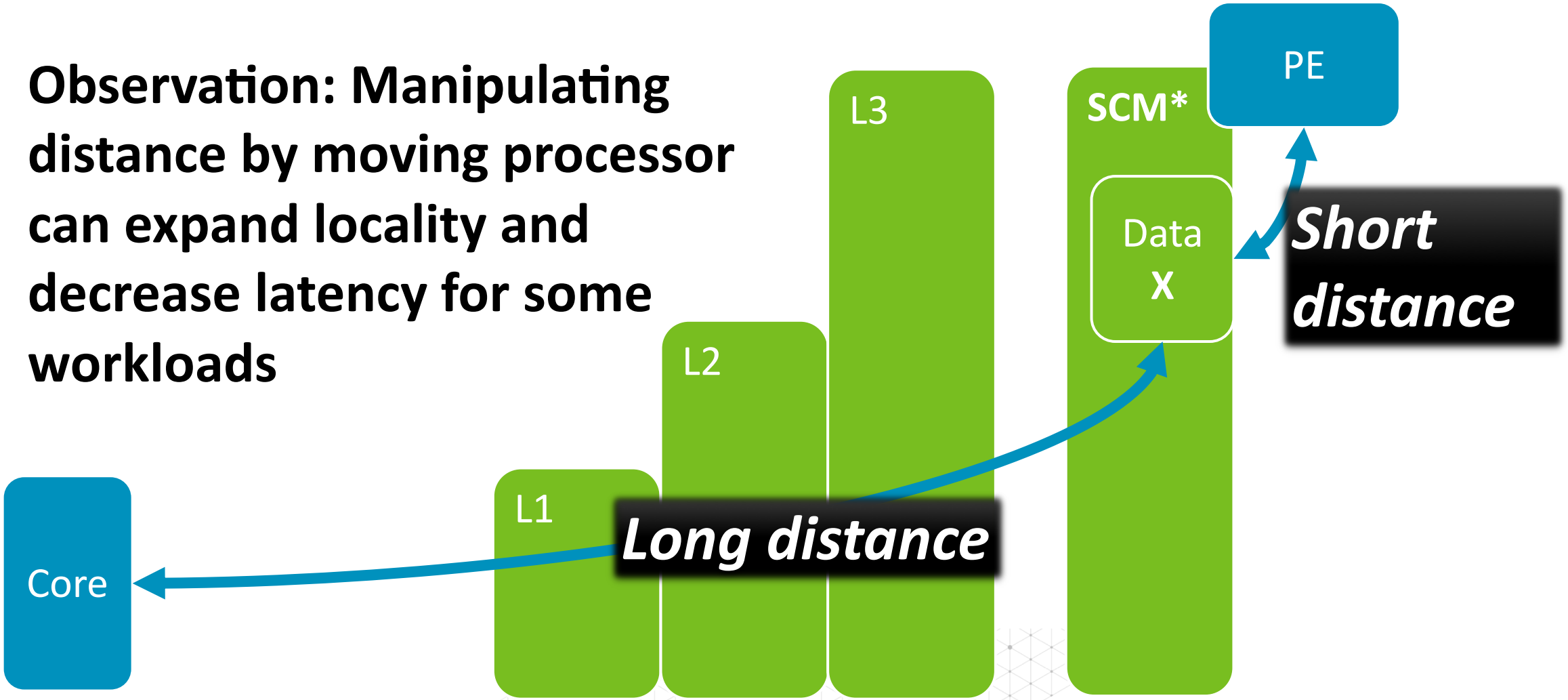


arm Research

Relativity

Locality is, from a certain point of view.

Observation: Manipulating distance by moving processor can expand locality and decrease latency for some workloads



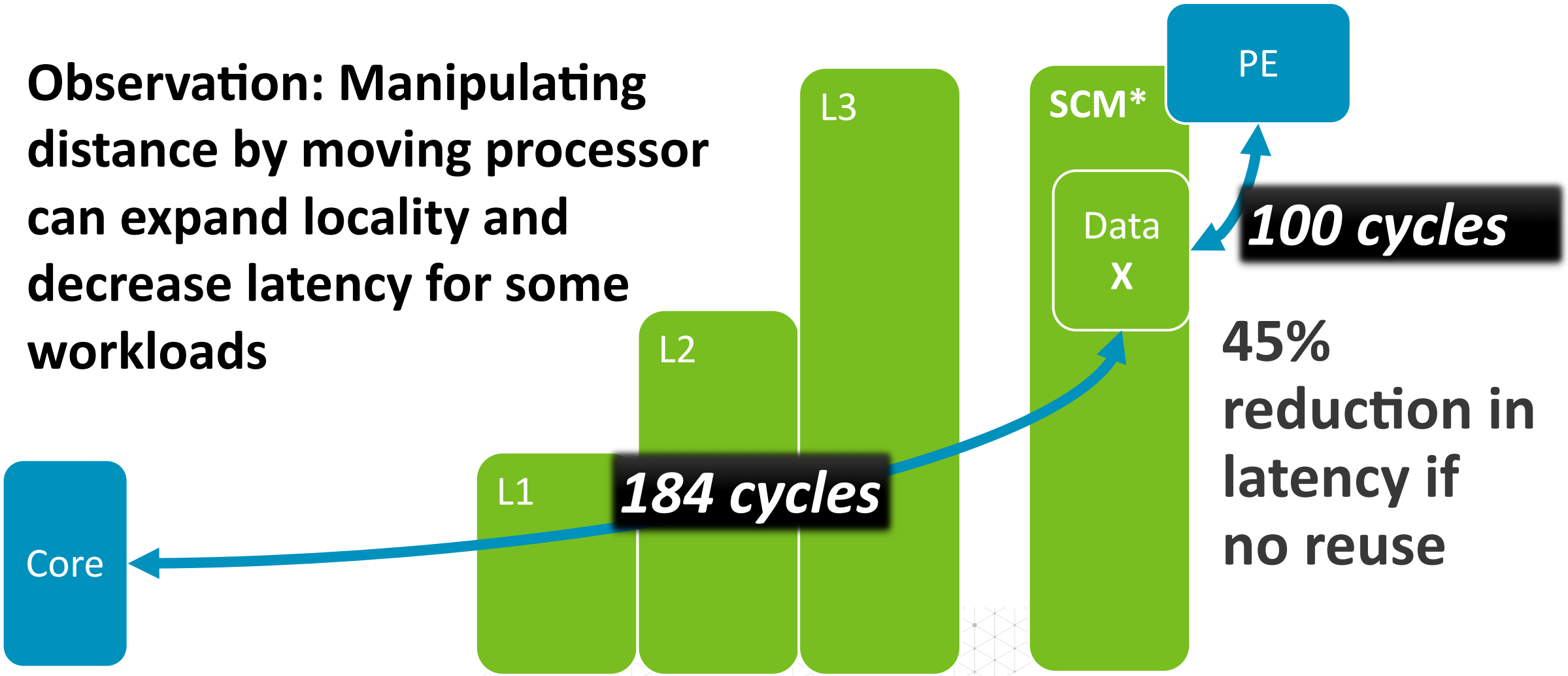
*Storage Class Memory - NVM

arm Research

Relativity

Locality is, from a certain point of view.

Observation: Manipulating distance by moving processor can expand locality and decrease latency for some workloads



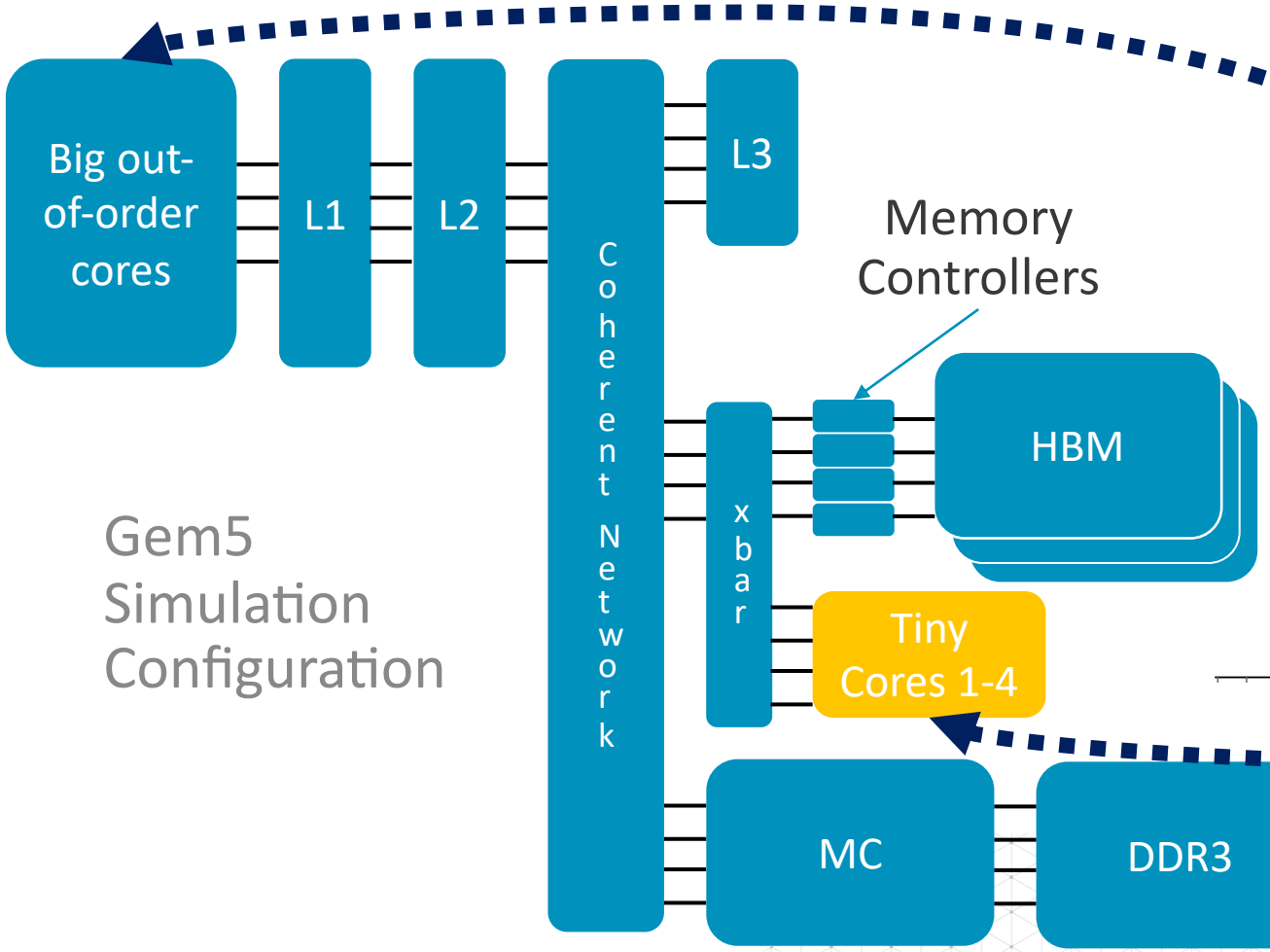
45% reduction in latency if no reuse

*Storage Class Memory - NVM

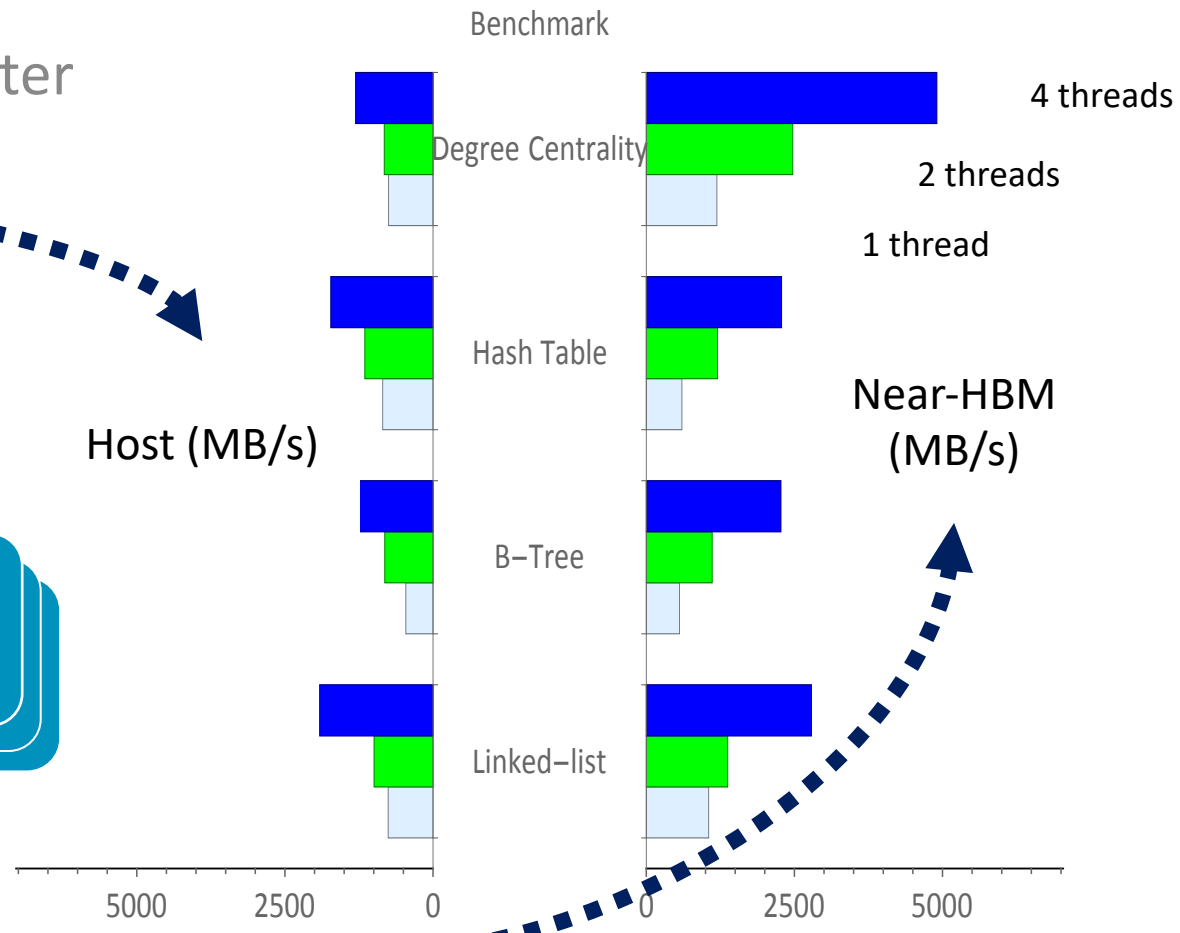
arm Research

Relativity

Application examples where reuse doesn't matter



Gem5 Simulation Configuration

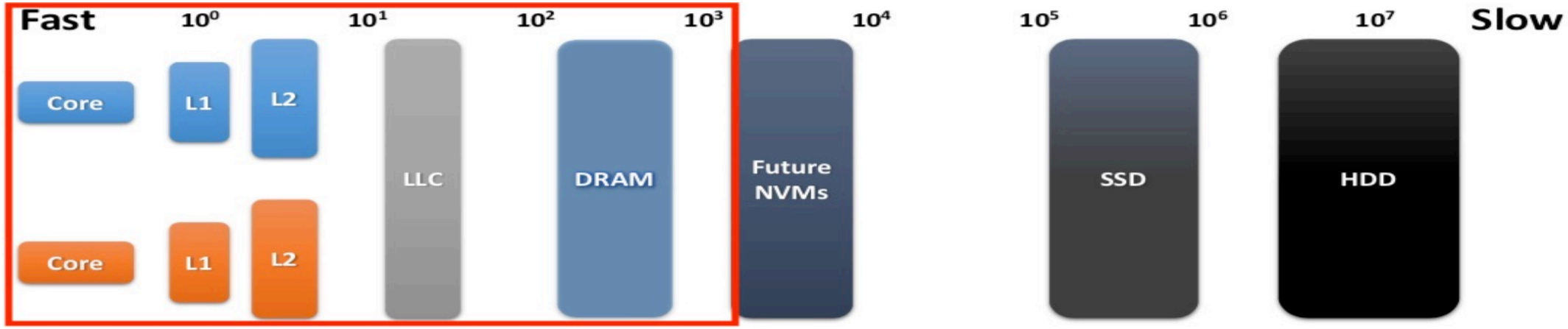


arm Research

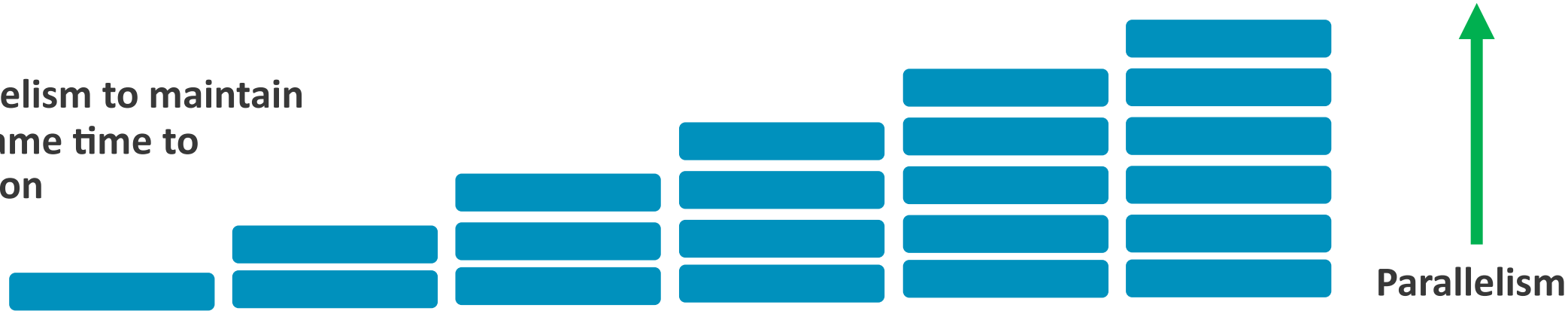
Memory Access Latency

Processor to Memory speed (clock cycles) ratio

Latency relative to core

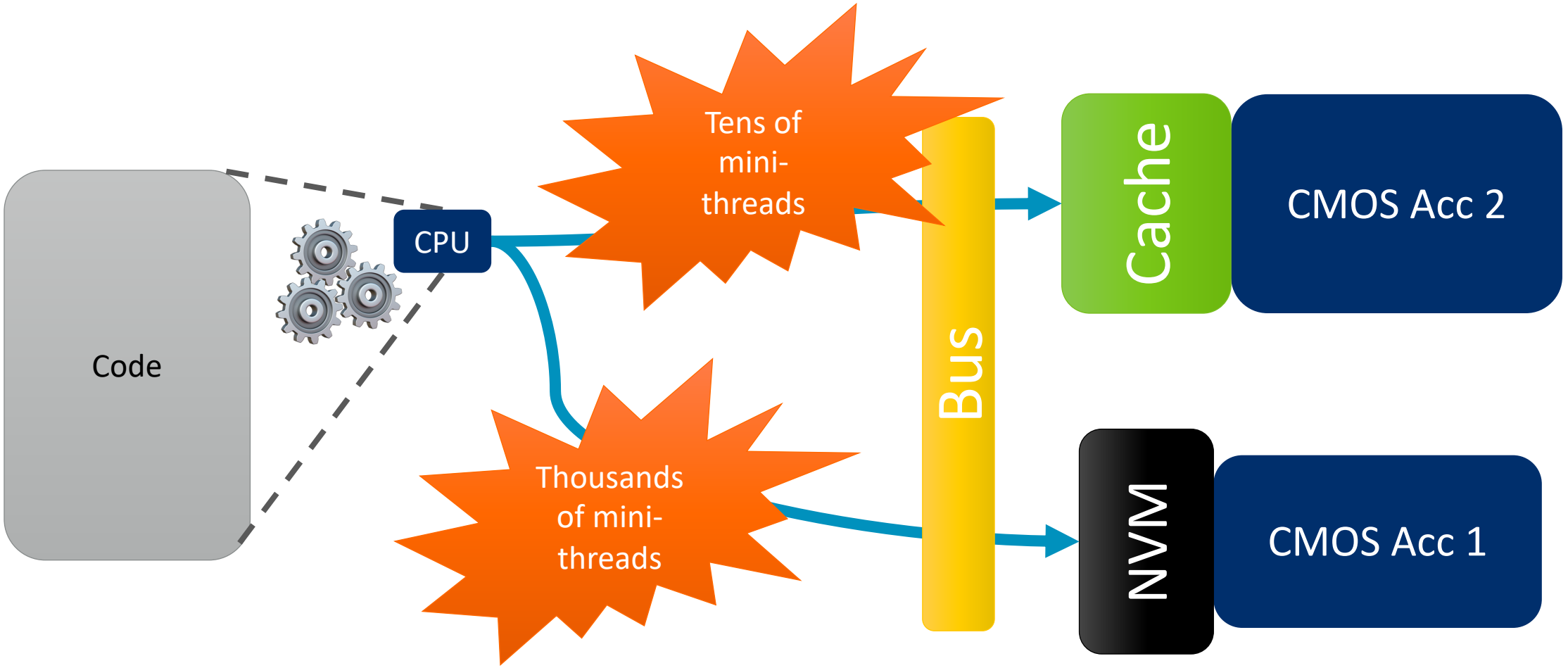


Parallelism to maintain the same time to solution



arm Research

Scalable parallelism



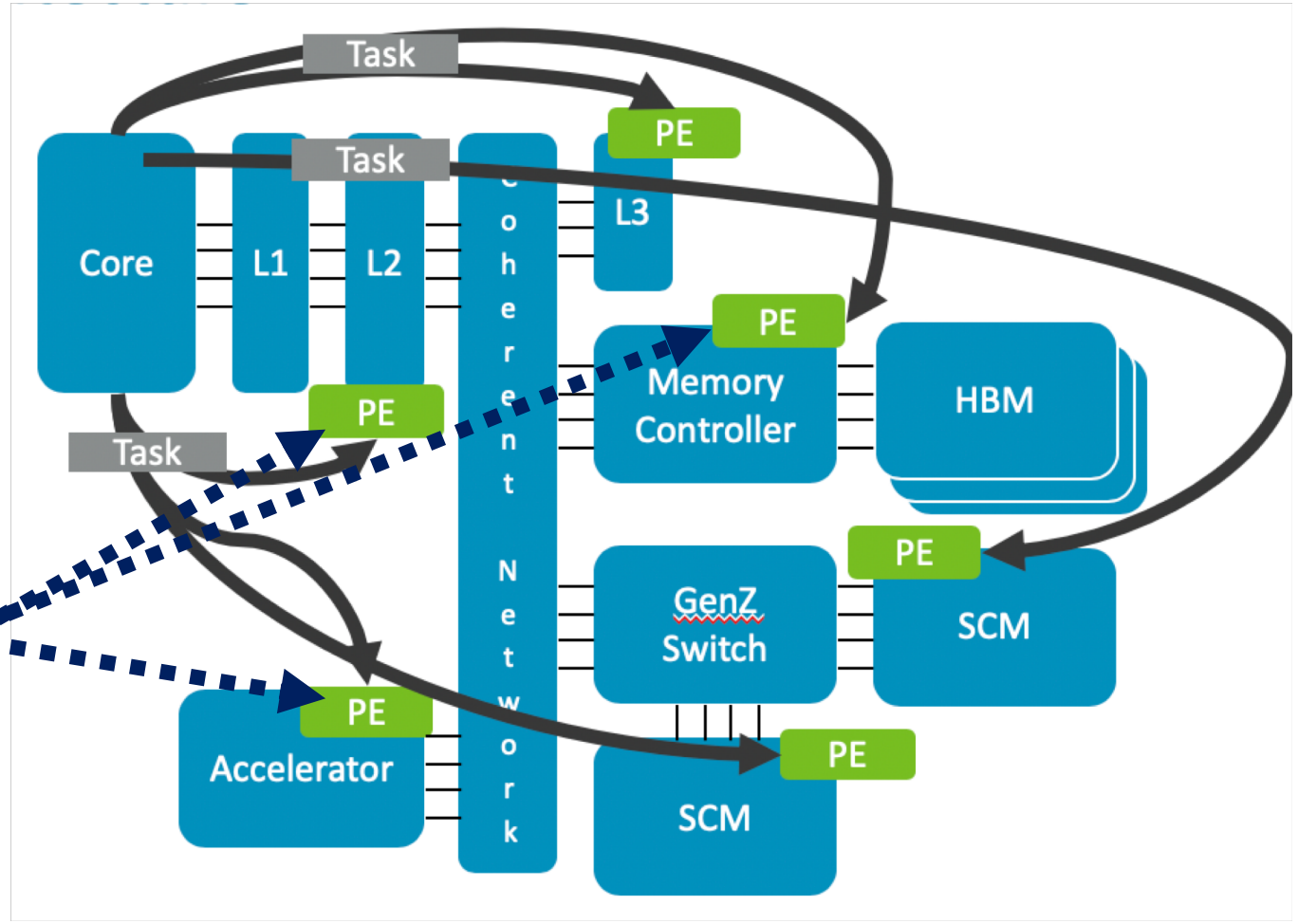
arm Research

There’s no such thing as processing in memory

Challenge: breaking the semantic barrier

Biggest problem with processing in memory is that we call it processing in memory.

These are just accelerators….



arm Research

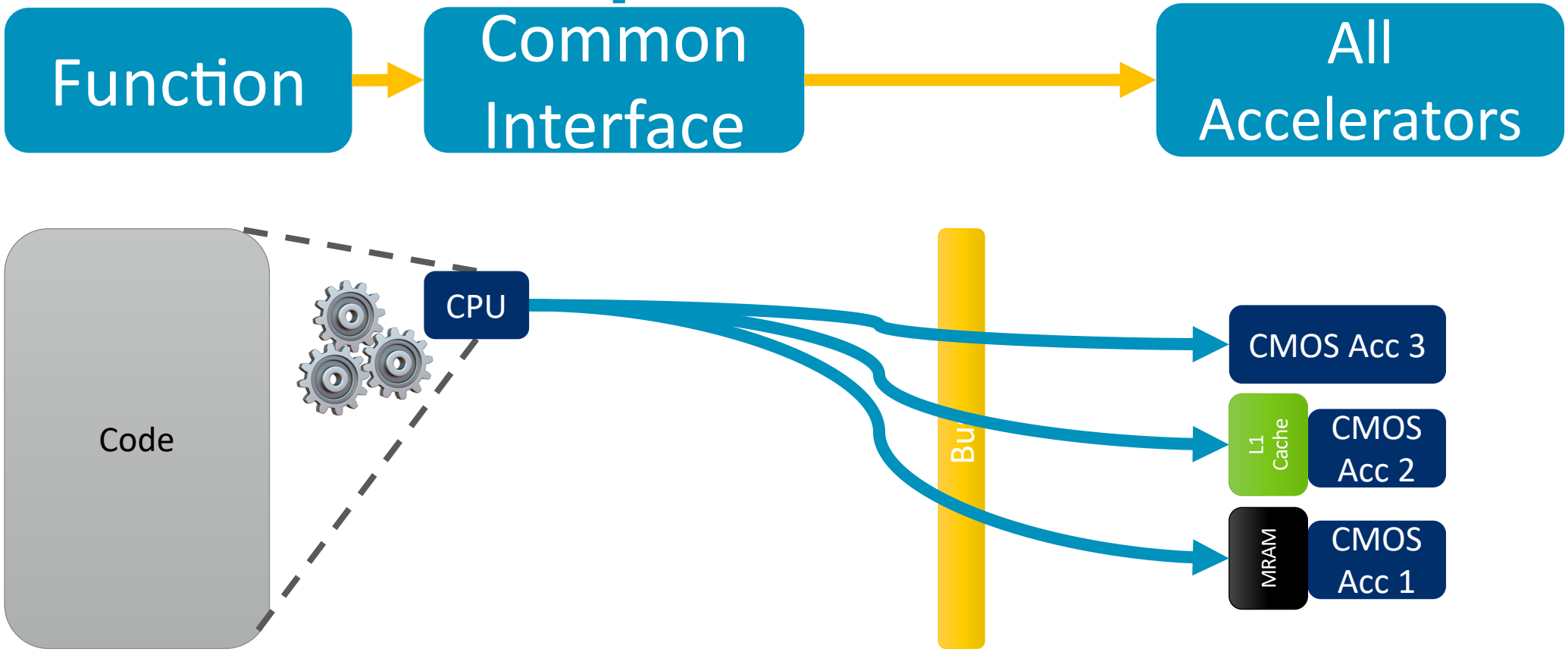
Interface standards

Explosion of interfaces…can we make them boring

Challenge

- Make it easy to build accelerators while making it easy to program / debug them

Unlock innovation on both sides of interface! – Minimize software disruption, maximize innovation pace



arm Research

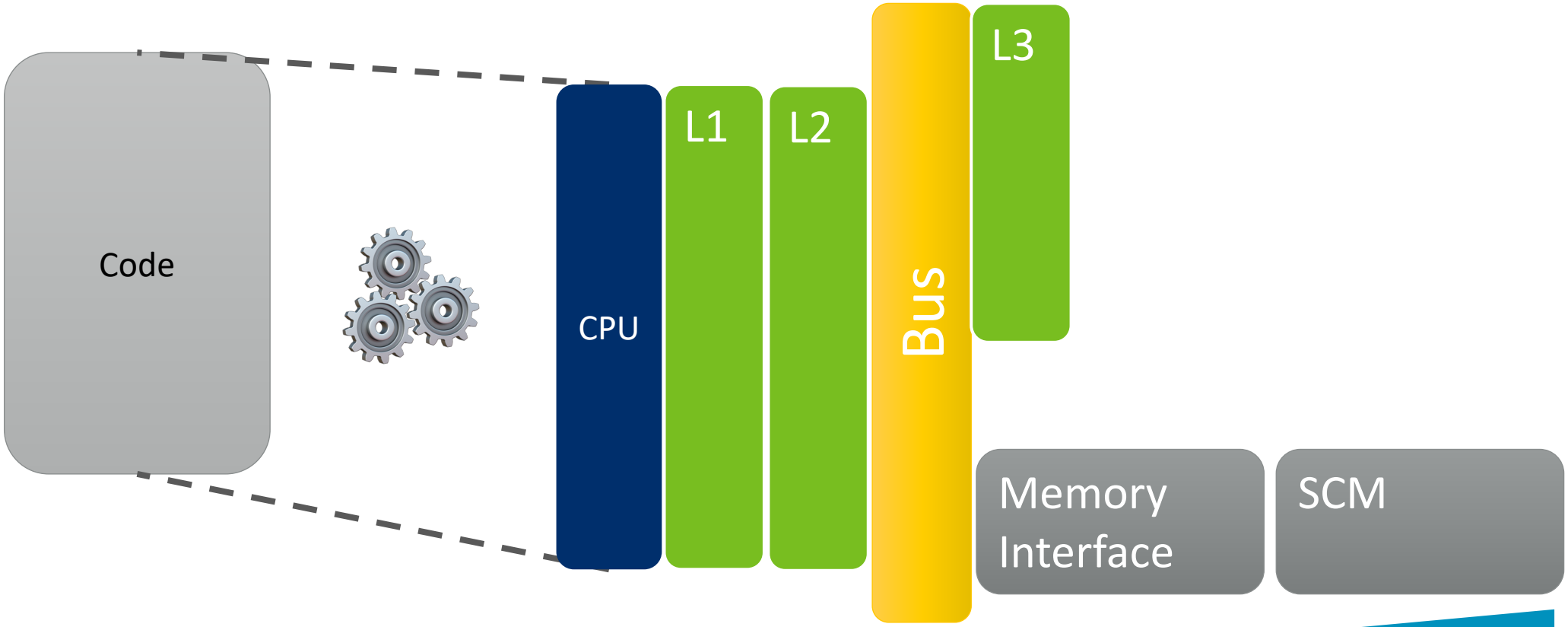
Interface standards

Data never rests

Challenge

- Locality-based targeting

Time to offload



How long will the data stay put

arm Research

Lost in translation

We need virtualization!

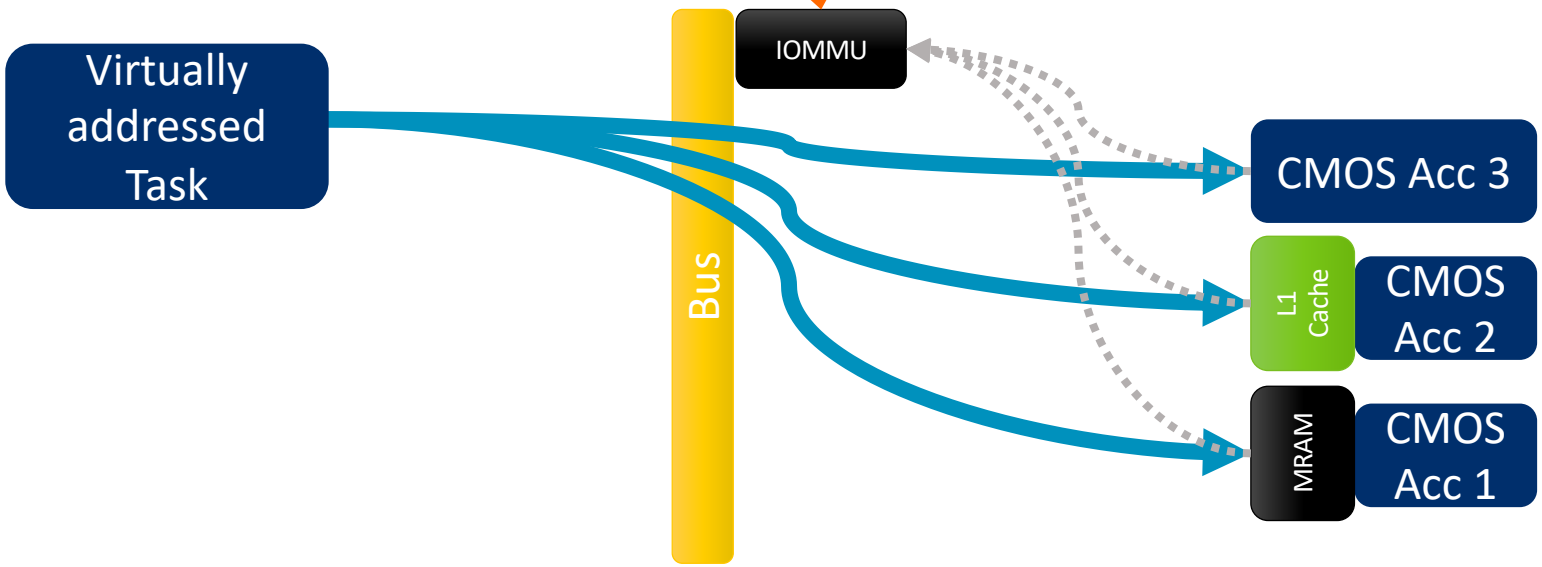
Imagine your processor of 64 cores sharing only a few of these

This works for a few course grained accelerators

IOMMU is effectively shared TLB across many high throughput cores. Is this a problem?

Challenge

- Increase translation reach for all cores
- Give all cores access to robust virtualization infrastructure



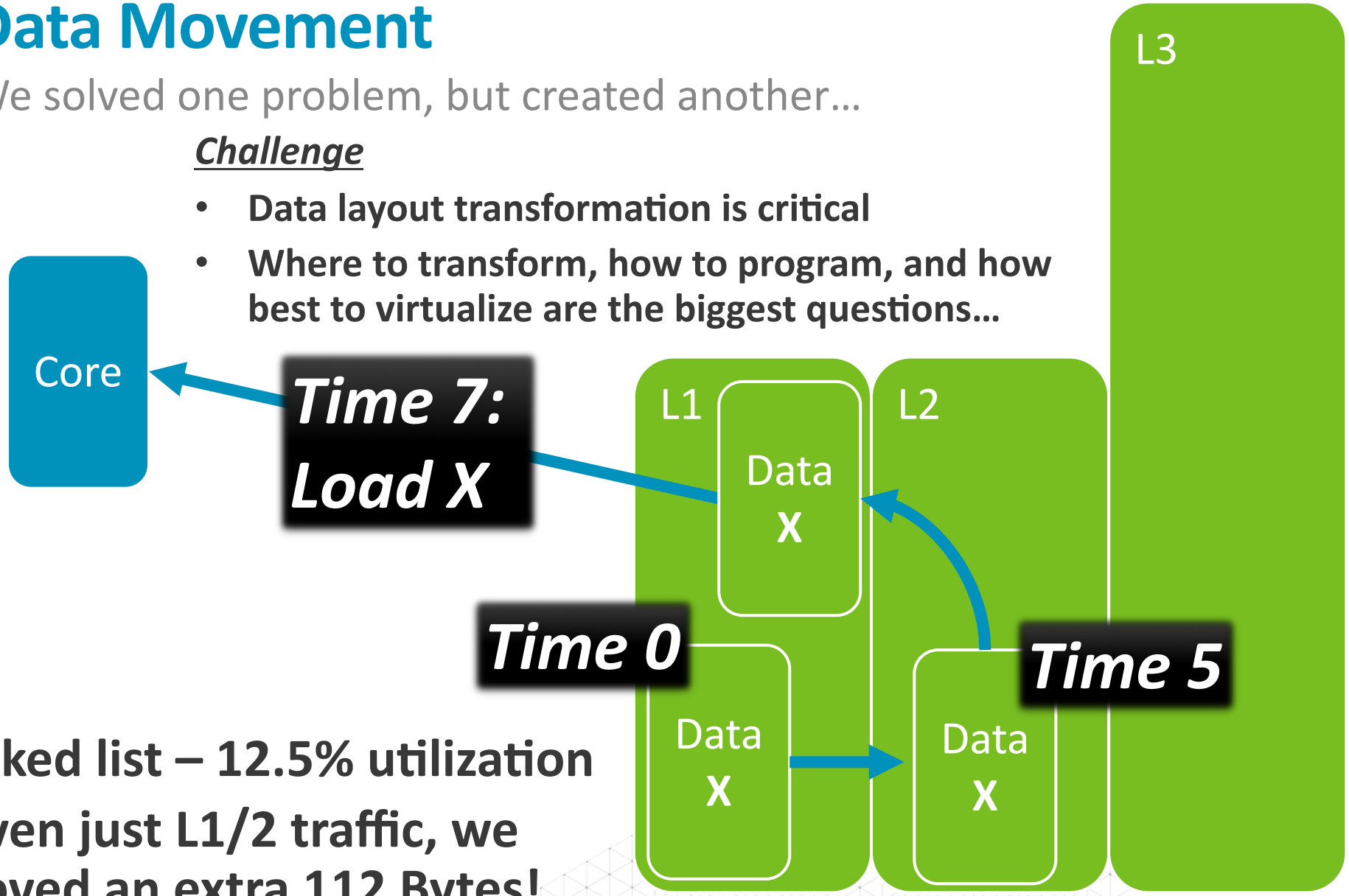
arm Research

Data Movement

We solved one problem, but created another…

Challenge

- **Data layout transformation is critical**
- **Where to transform, how to program, and how best to virtualize are the biggest questions…**



L1D Cache Line Utilized versus Cache Line Wasted

GUPS: 80%

CoMD: 50%

mcb: 40%

LULESH: 20%

DGEMM: 10%

Linked list – 12.5% utilization

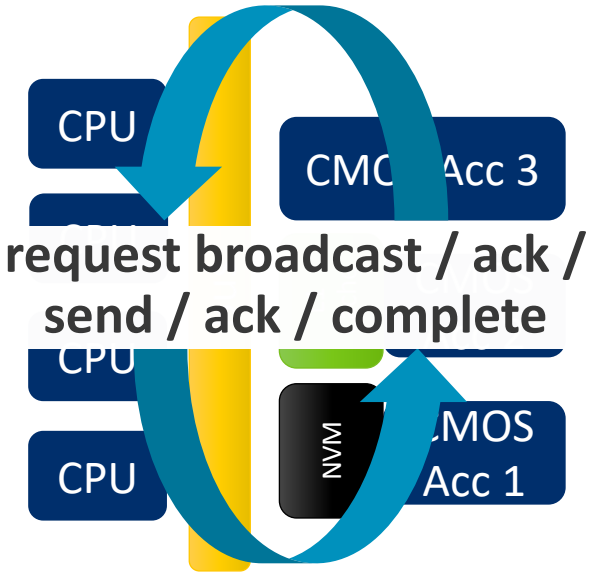
Given just L1/2 traffic, we moved an extra 112 Bytes!

arm Research

Keeping dark bandwidth coherent

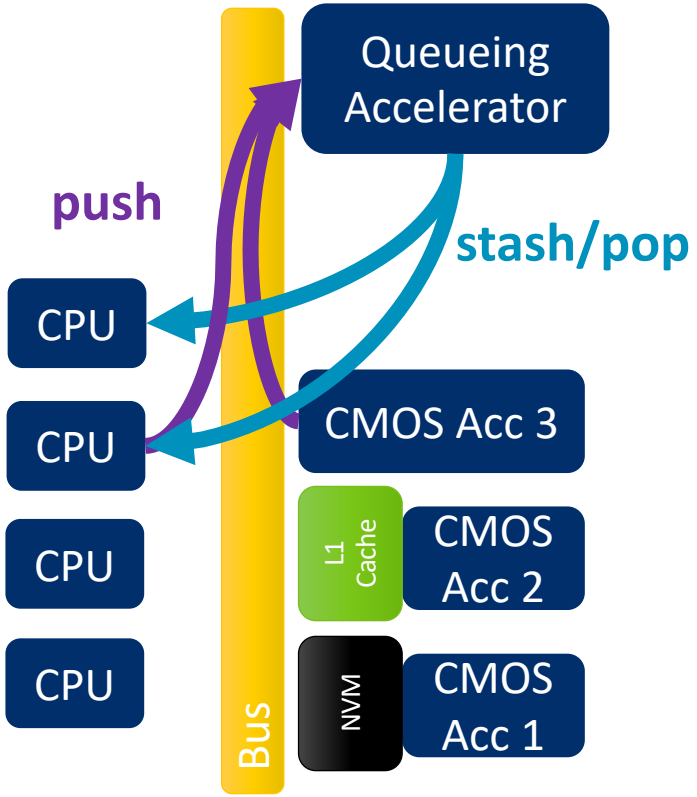
Hmmm….maybe there's a better way.

- *Dark Bandwidth* blows up transfers, moving one cache line actually moves far more than that!
- With every move comes coherence traffic, often lots.
- Synchronization also takes time…



Challenge

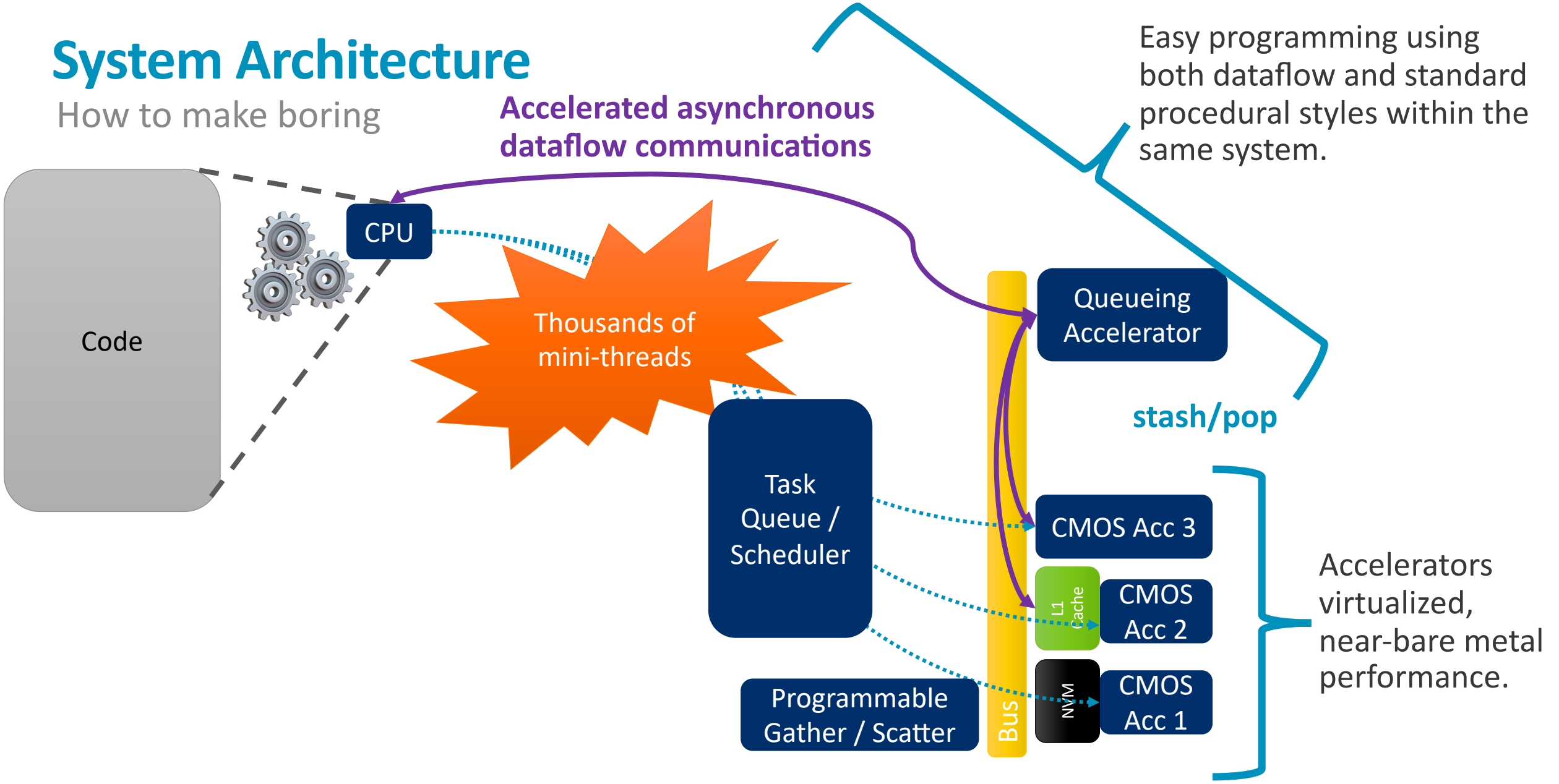
- **Improve performance / transparency of communications between all processing elements.**
- **Do we really need coherence? (likely not always)**



arm Research

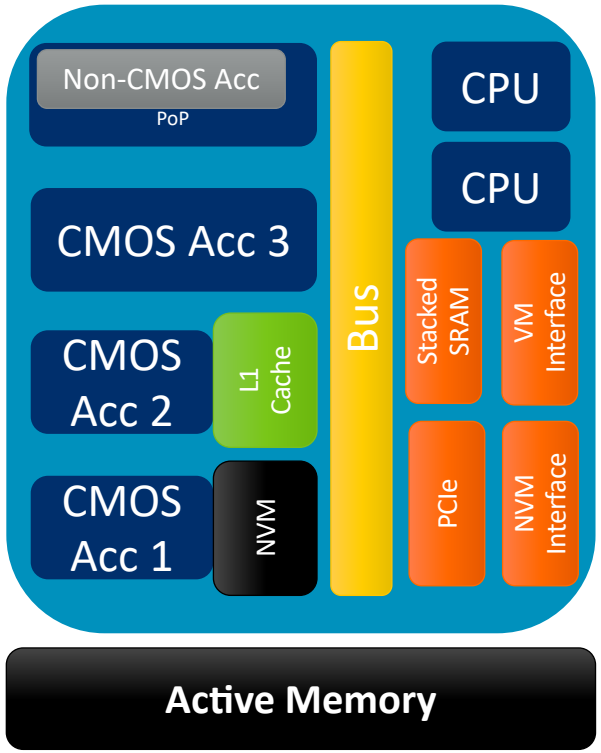
System Architecture

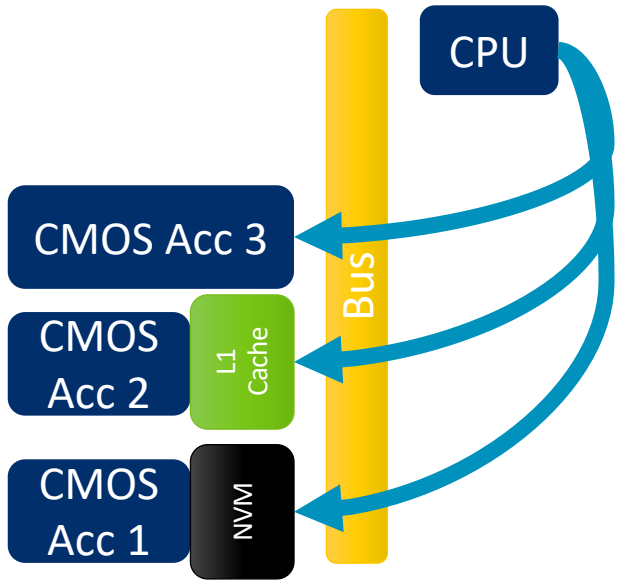
How to make boring



arm Research

Grand Challenges





1: Efficiency of data movement, logic is cheap, movement is expensive – future systems must capitalize on both data with reuse and streaming data, *Dark Bandwidth* must be avoided

2: Multiple drivers / compilers / software stacks are multi-million-dollar efforts for each vendor – will developers even adopt? – Reducing cost is a huge disruptor!

3: Communications / scalability of cores is not good with current coherence methods – but specialization and more cores are the future, Post-Moore.

4: Virtualization and translation for accelerators is an afterthought at the moment, extending the virtual memory model eases programming – but can we do more?

arm Research

Thank You!
Danke!
Merci!
谢谢!
ありがとう!
Gracias!
Kiitos!

arm Research