Big Data & Dark Silicon:

Taming Two IT Trends on a Collision Course

Babak Falsafi
Director, EcoCloud
ecocloud.ch

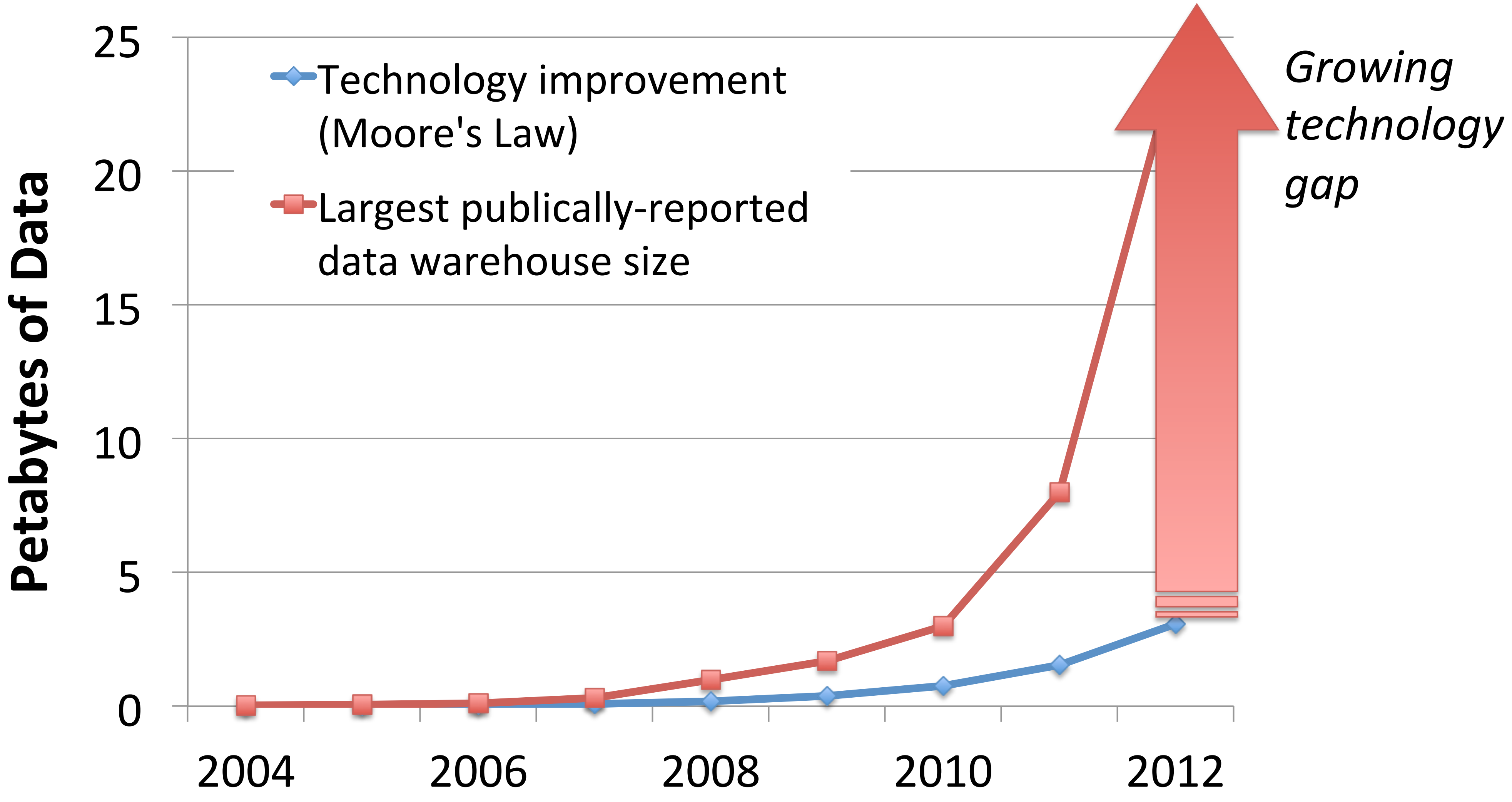
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Inflection Point #1: IT is all about Data

[source: Economist]

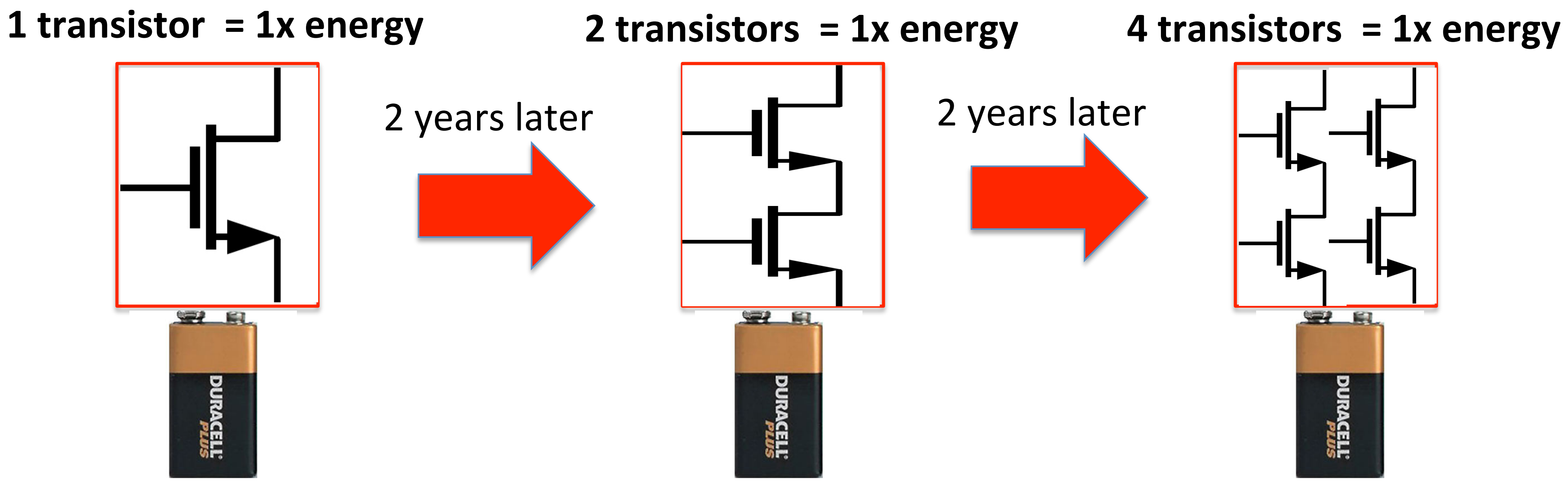
- Data growth (by 2015) = 100x in ten years [IDC 2012]
  - Population growth = 10% in ten years
- Monetizing data for commerce, health, science, services, ….
- Big Data is shaping IT & pretty much whatever we do!

Data Growing Faster than Technology



WinterCorp Survey, www.wintercorp.com

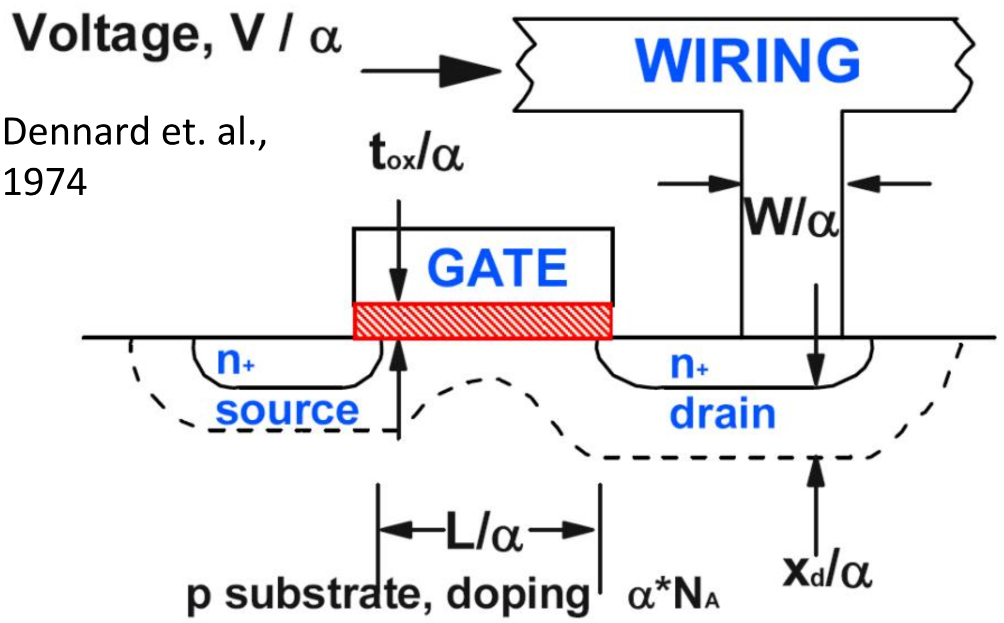
Inflection Point #2: Energy used to be "Free"



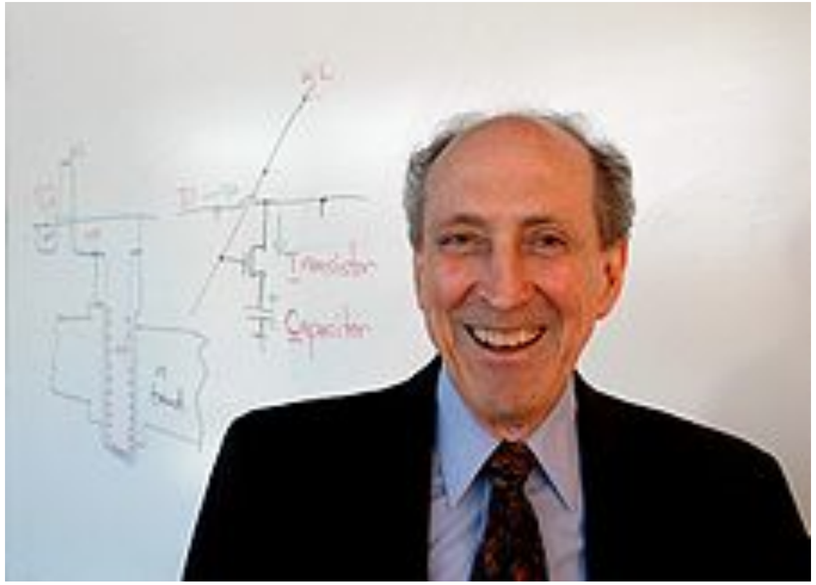
Before (1970~2005):

- Used to make transistors smaller
- Smaller transistors less electricity to operate
- Chip energy consumption remained ~ same

Where did "Free" Energy Go?



Robert H. Dennard, picture from Wikipedia

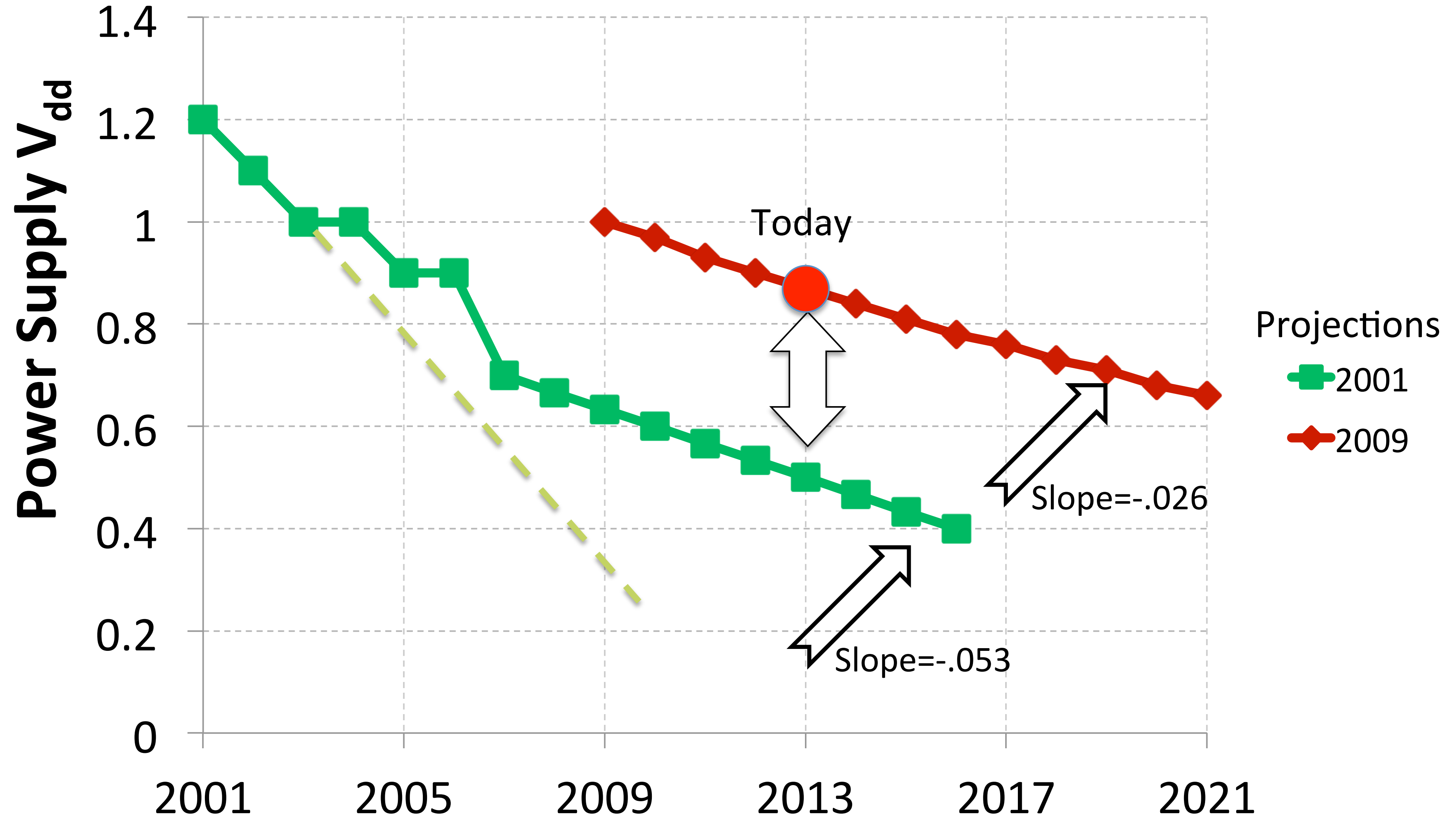


Four decades of Dennard Scaling (1970~2005):

- $P = CV^2 f$
- More transistors
- Lower voltages

➜ Constant power/chip

End of Dennard Scaling



[source: ITRS]

The fundamental energy silver bullet is gone!

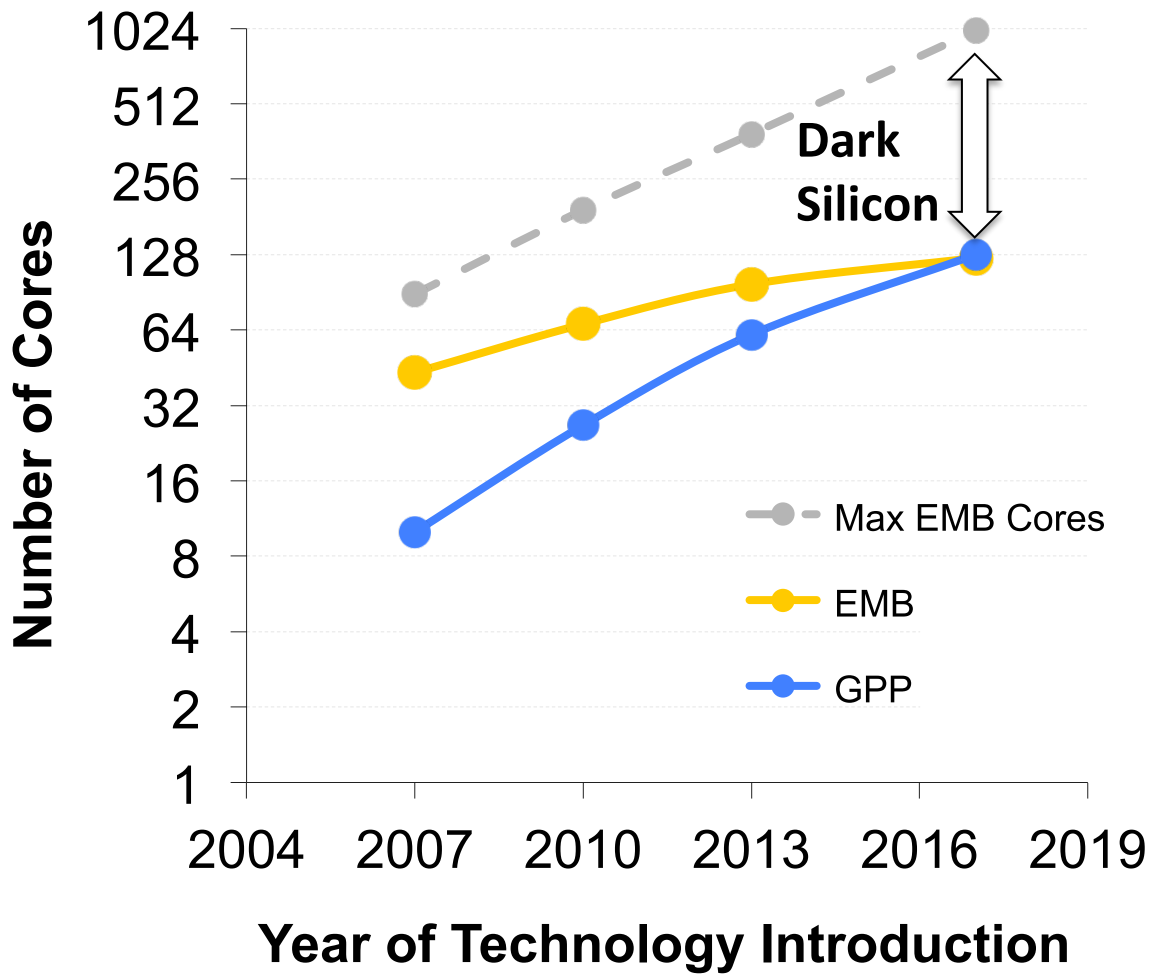
Dark Silicon: End of Multicore Scaling

Parallel computing is emerging as a popular solution for efficiency

But parallelism alone can not offset leveling voltages!

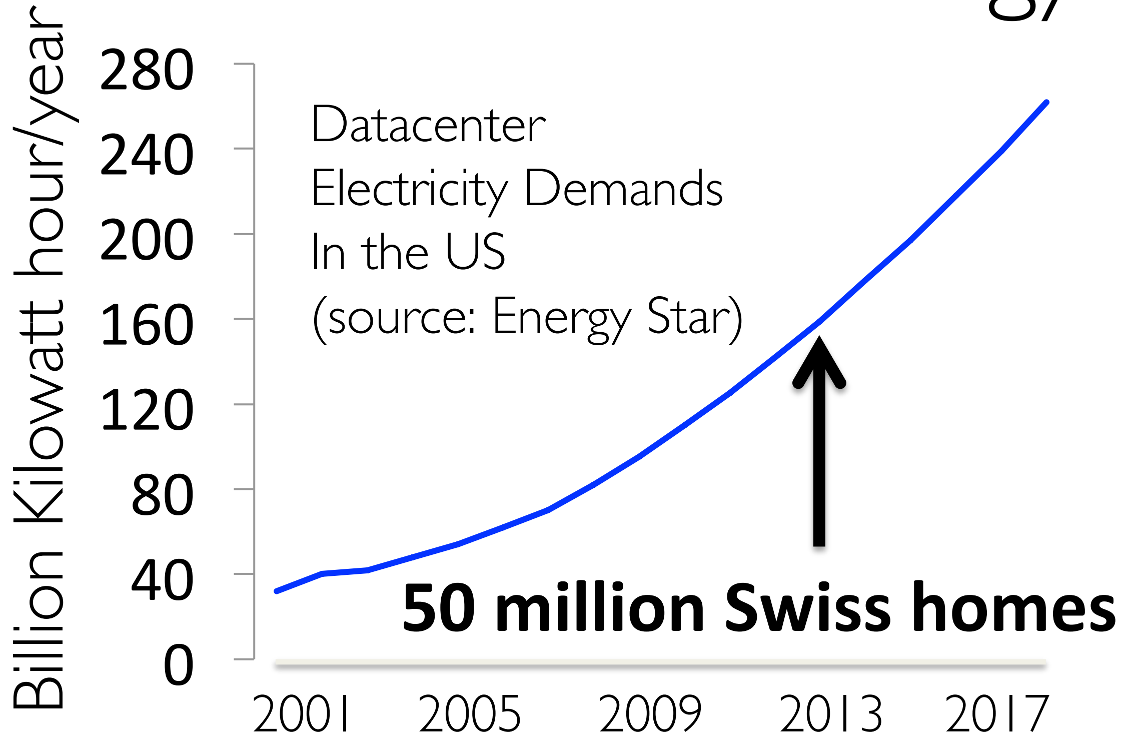
Even in servers:

- With abundant parallelism
- Programmable cores are at efficiency limits
- Soon, cannot power all chip



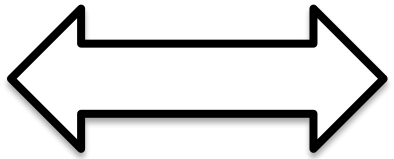
Hardavellas et. al., "Toward Dark Silicon in Servers", IEEE Micro, 2011

Higher Demand + Lower Efficiency: Datacenter Energy Not Sustainable!



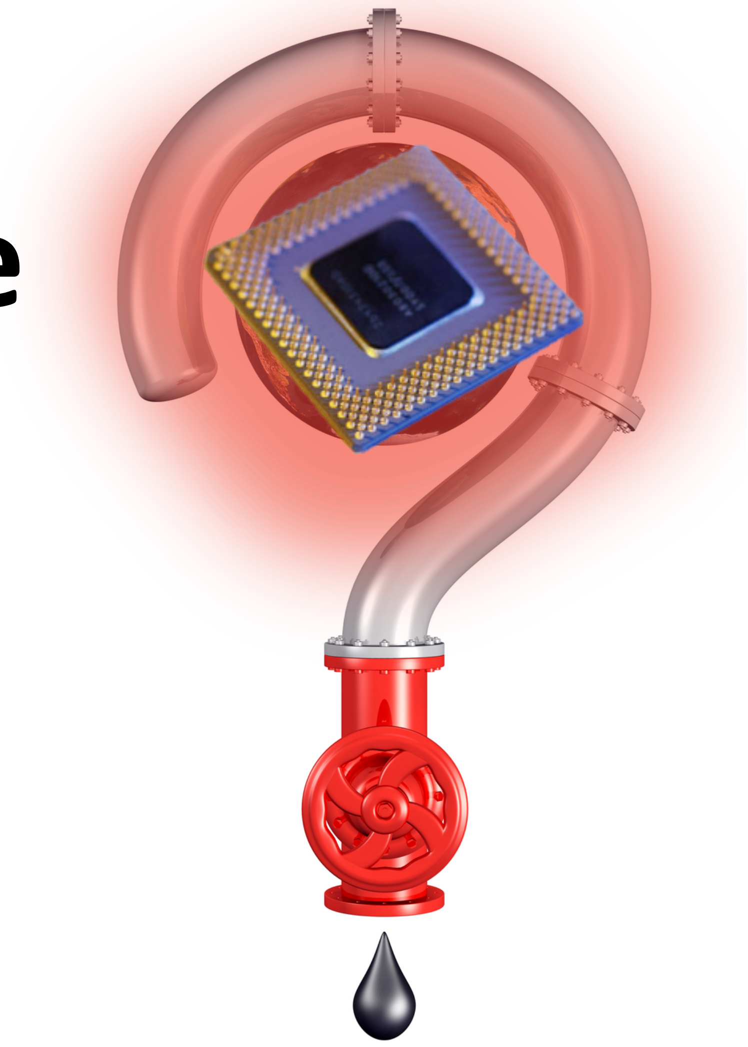
- Modern datacenters ➔ 20 MW!
- In modern world, 6% of all electricity, growing at >20%!

IT's Future



Bridging

Technologies



ecocloud

Center to bring efficiency to data

- 16 faculty, 50 researchers
- Around $3M/year budget

Mission:

- Energy-efficient data-centric IT
- From algorithms to machine infrastructure
- Maximizing Performance/TCO for Big Data

Microsoft

EATON Powering Business Worldwide

hp

intel

swisscom

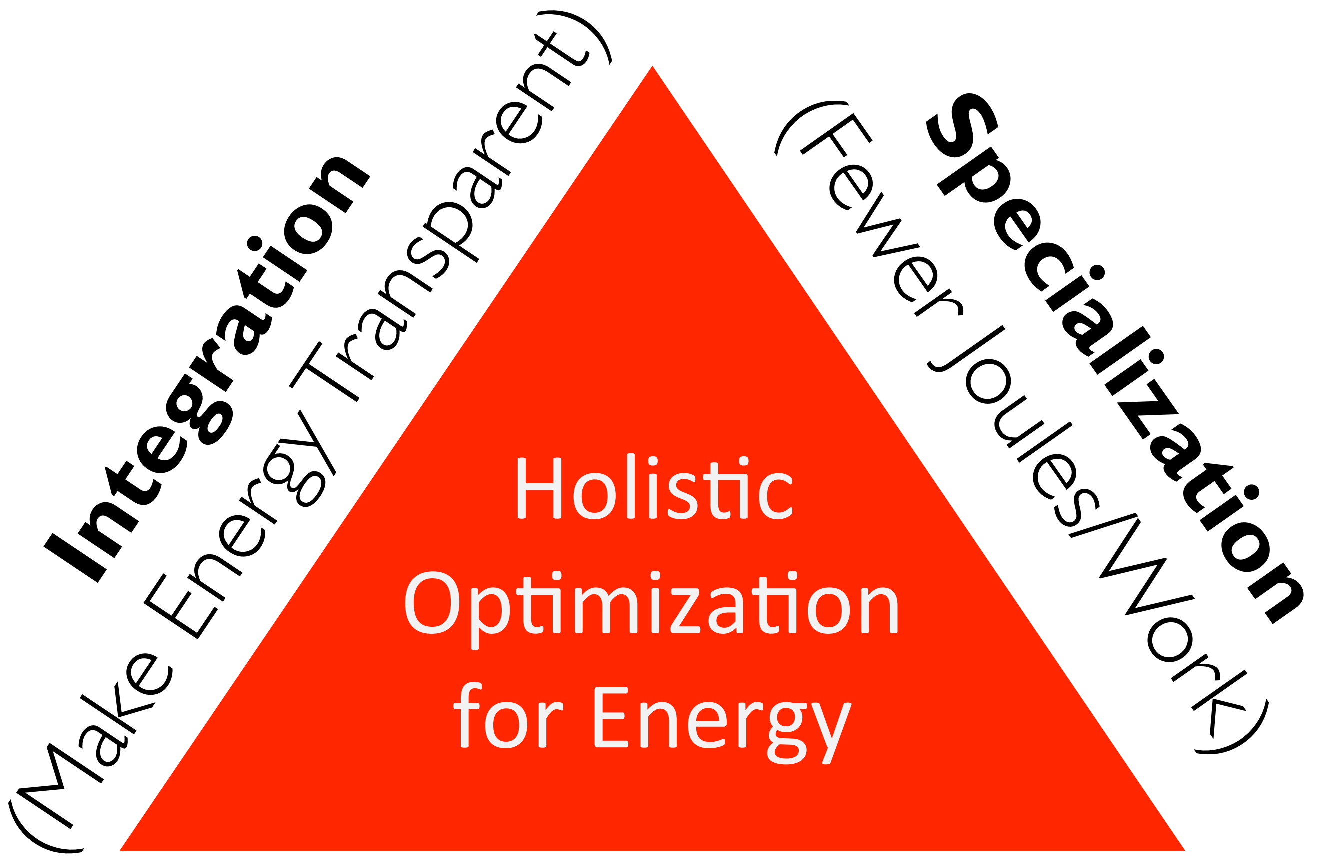
CREDIT SUISSE

ORACLE

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ecocloud.ch

Our Vision:
The ISA Triangle of Efficiency



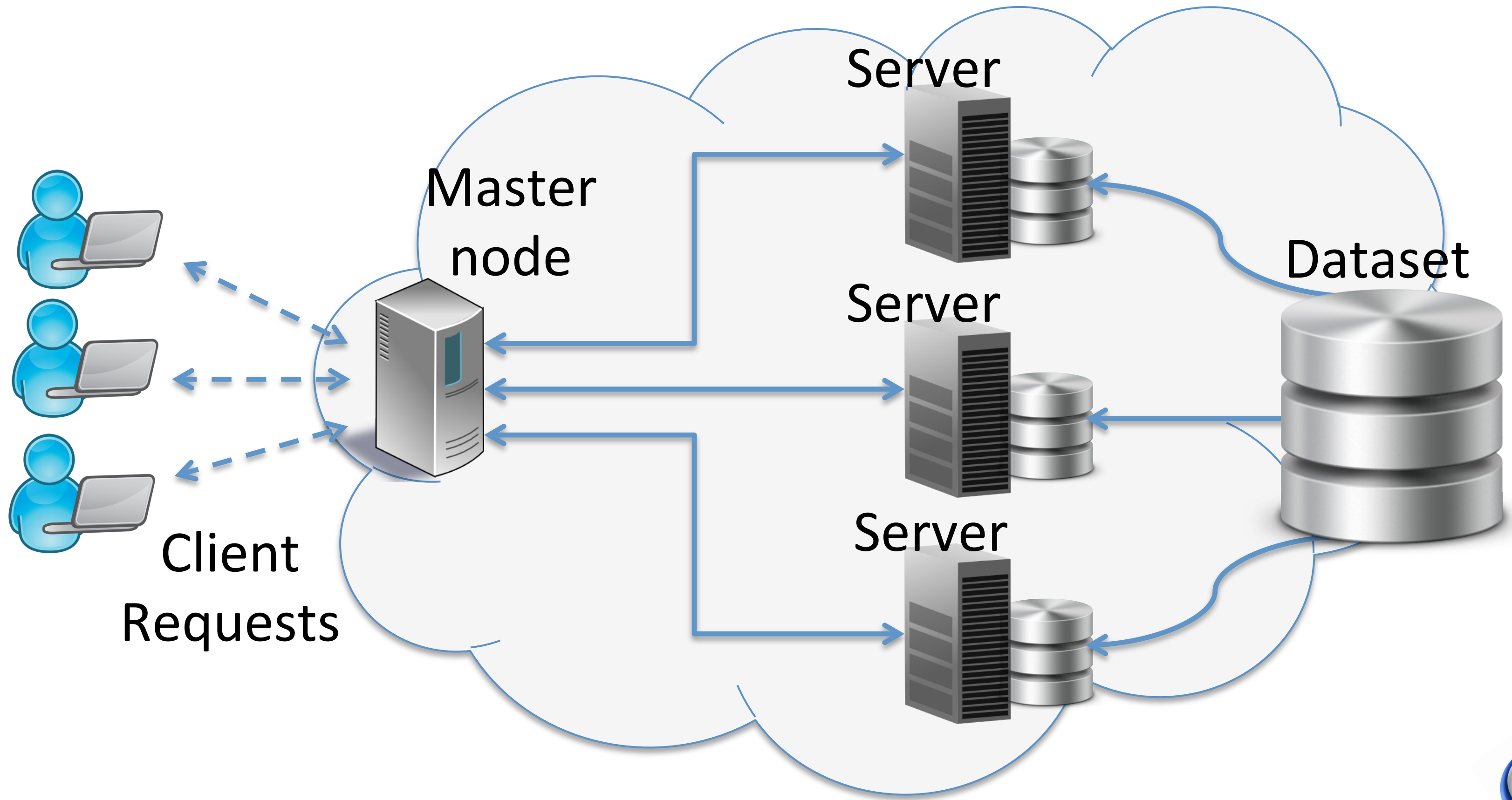
Approximation

(Trade off Accuracy for Energy)

Outline

1. Two infliction points colliding
2. How bad are today's servers?
   - Scale-Out Workloads
   - CloudSuite 2.0
   - Microarchitectural footprints
3. How do we build efficient servers?

Data-Intensive Online Services



- Many independent requests/tasks
- Huge dataset split into shards
- Minimal communication among servers

Scale-Out Datacenters

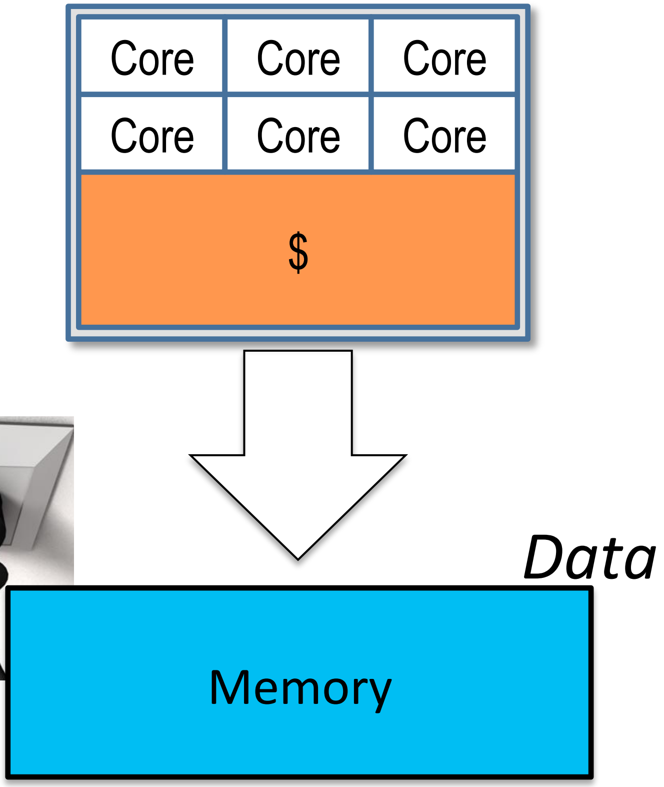
Vast data sharded across servers

Memory-resident workloads

- Necessary for performance
- Major TCO burden

Processors access data in memory

- Abundant request-level parallelism
- Performance scales with core count

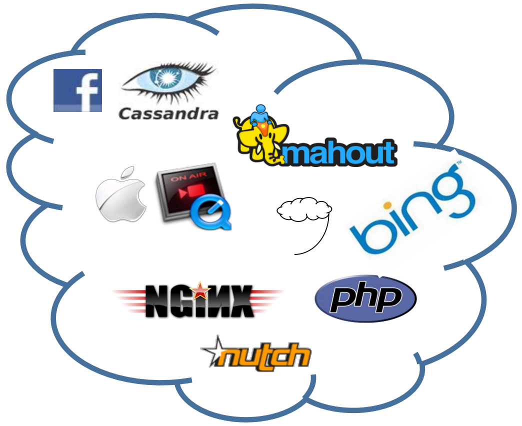


Maximize performance for better TCO

How Efficient are Today's Servers?

["Clearing the Clouds", ASPLOS '12]

- Created benchmark suite
  - Diverse set of cloud workloads
  - Quantified high-level behavior



- Studied off-the-shelf hardware
  - Used performance counters
  - Identified needs of cloud apps

Modern CPUs don't match needs of cloud apps

CloudSuite 2.0

(released @ parsa.epfl.ch/cloudsuite)

Data Analytics
Machine learning

Data Caching
Memcached

Data Serving
Cassandra NoSQL

Graph Analytics
TunkRank

Media Streaming
Apple Quicktime Server

SW Testing as a Service
Symbolic constraint solver

Web Search
Apache Nutch

Web Serving
Nginx, PHP server

Covers popular scale-out services

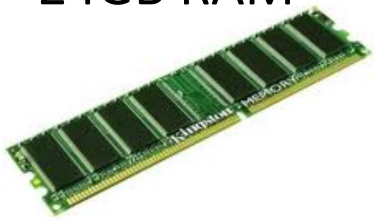
Hardware

Dell PowerEdge
M1000e

Dell Blades
M610

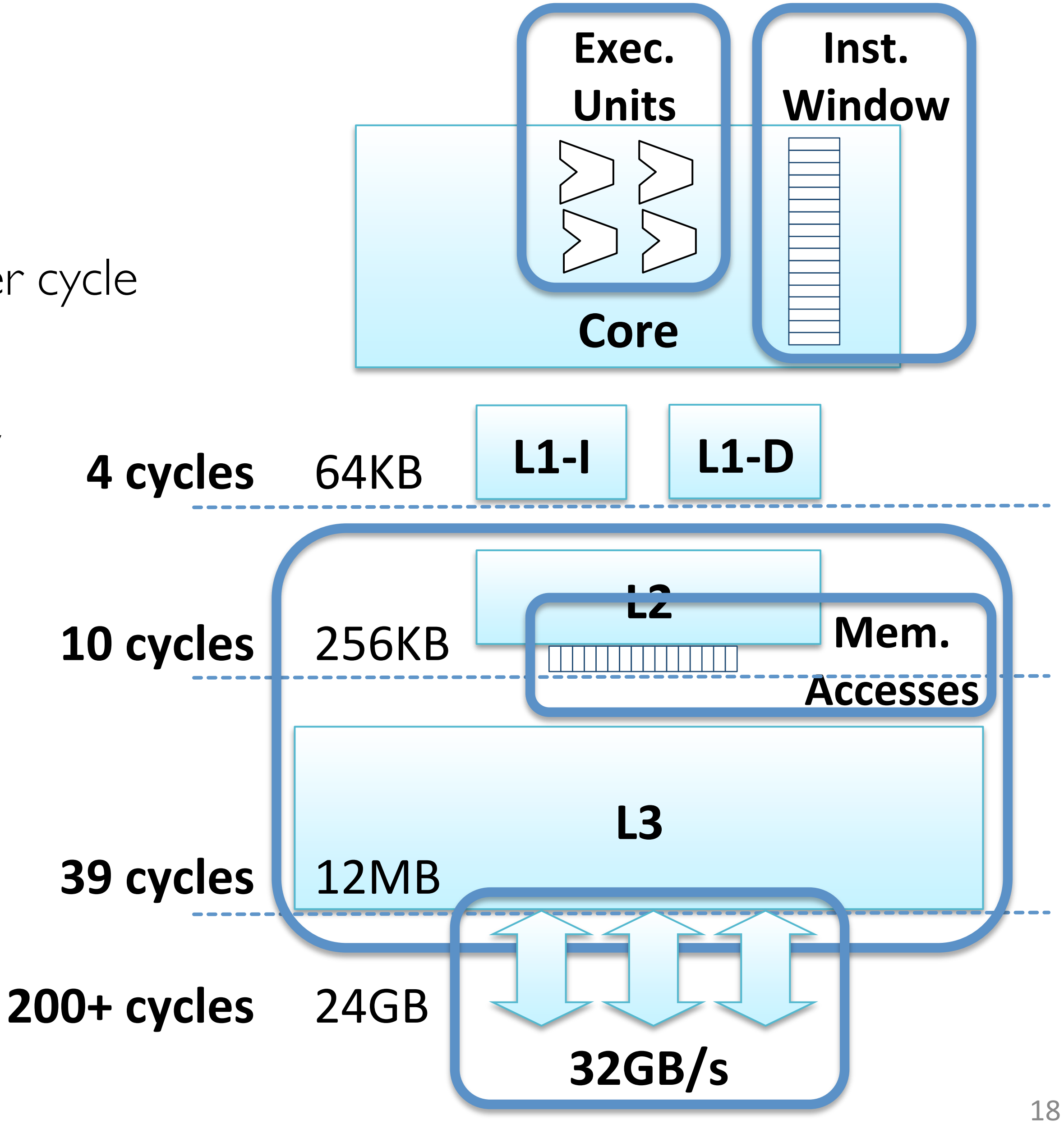
Two Intel x5670
2.9GHz
6-core, 12MB LLC

24GB RAM

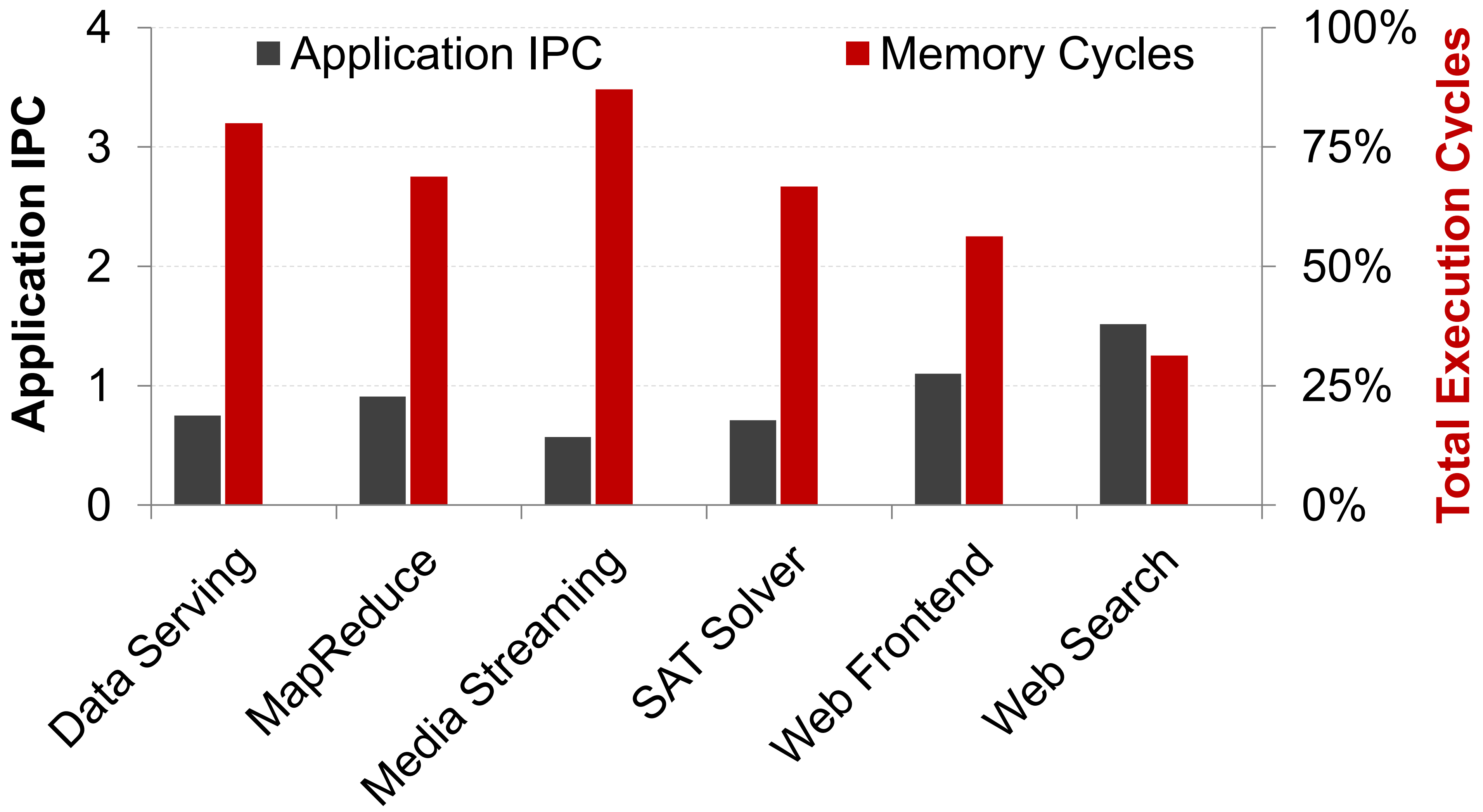


Methodology: "Server-grade" CPU

- Aggressive OoO cores
  - Run up to 4 instructions per cycle
- Large instruction window
  - 128 instructions in flight
  - 48 loads in flight
- L2 and large L3 caches
- Vast off-chip b/w



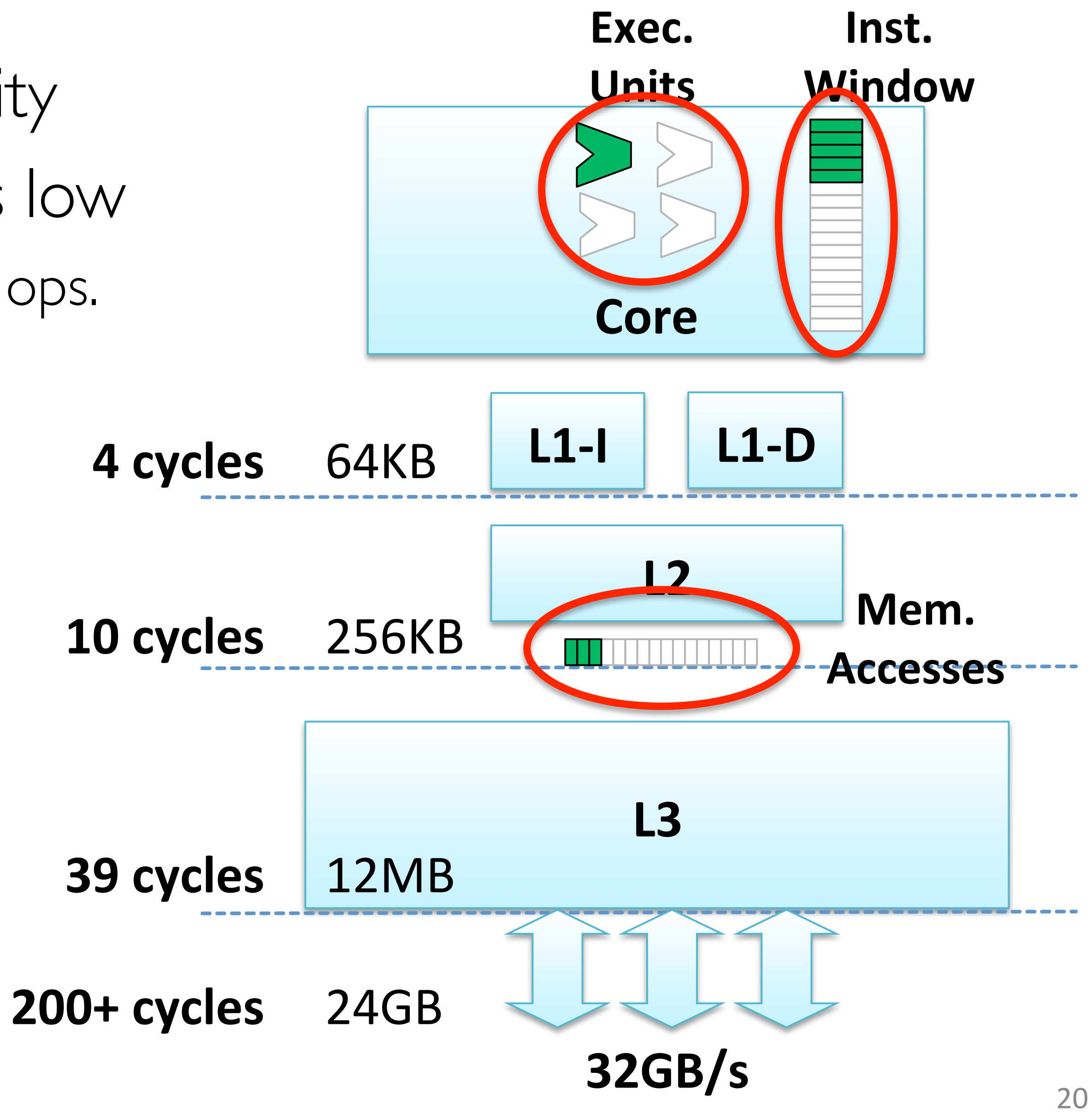
Core Inefficiencies



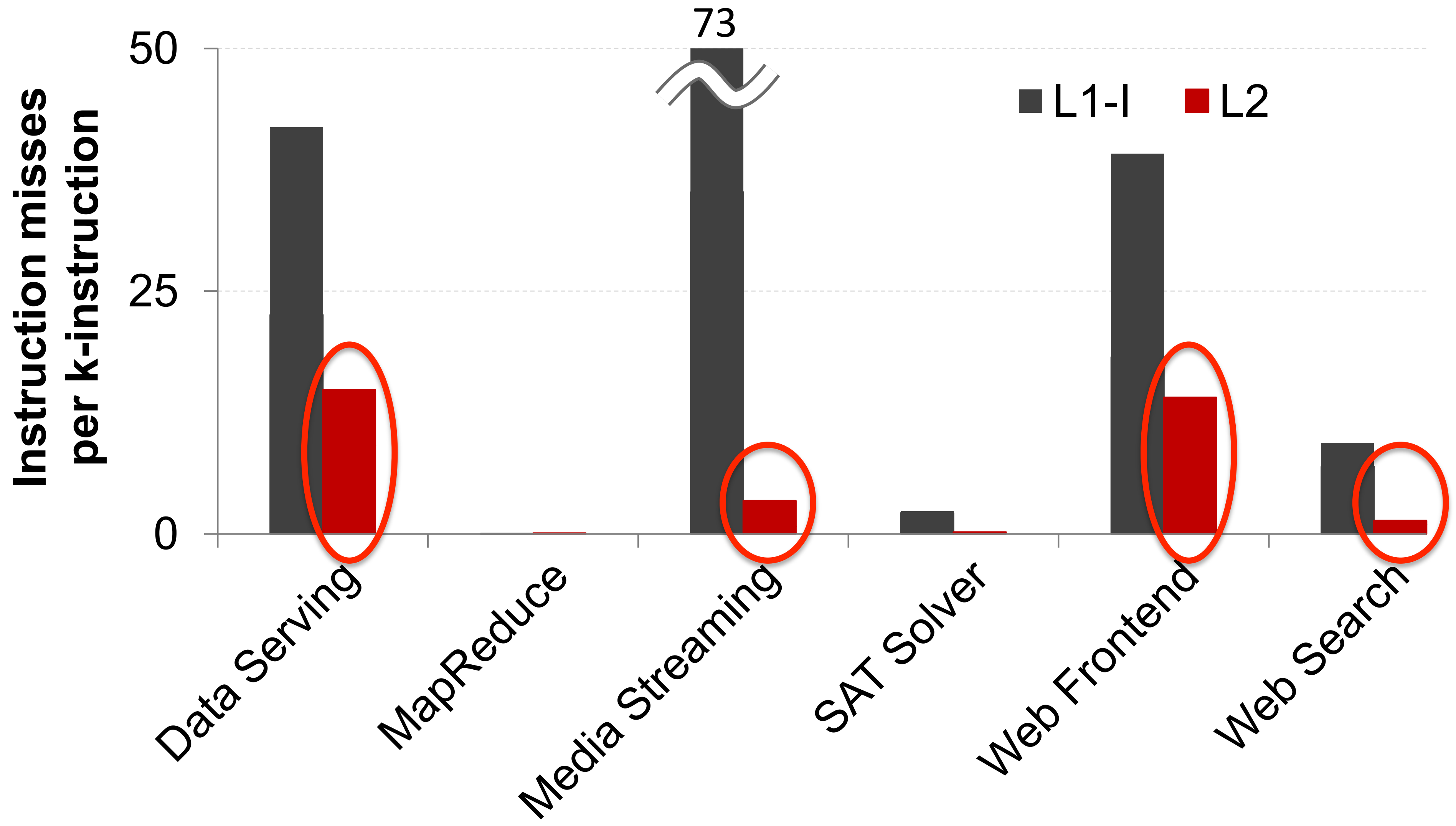
Execute ~1 instruction per cycle

Core Inefficiencies

- Underutilized complexity
- Scale-out requirements low
  - couple parallel memory ops.
  - one execution unit



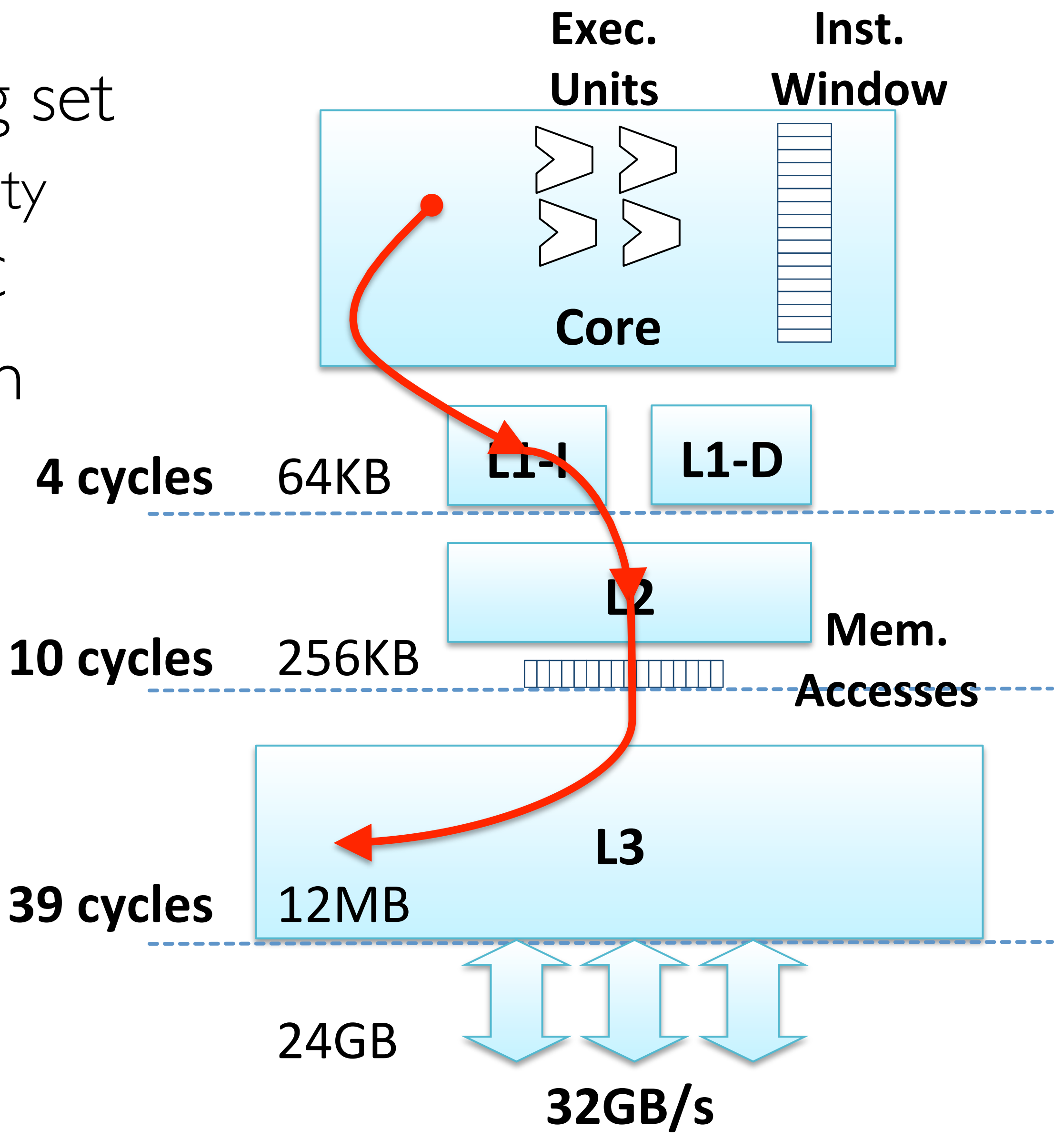
Instruction-Fetch Misses



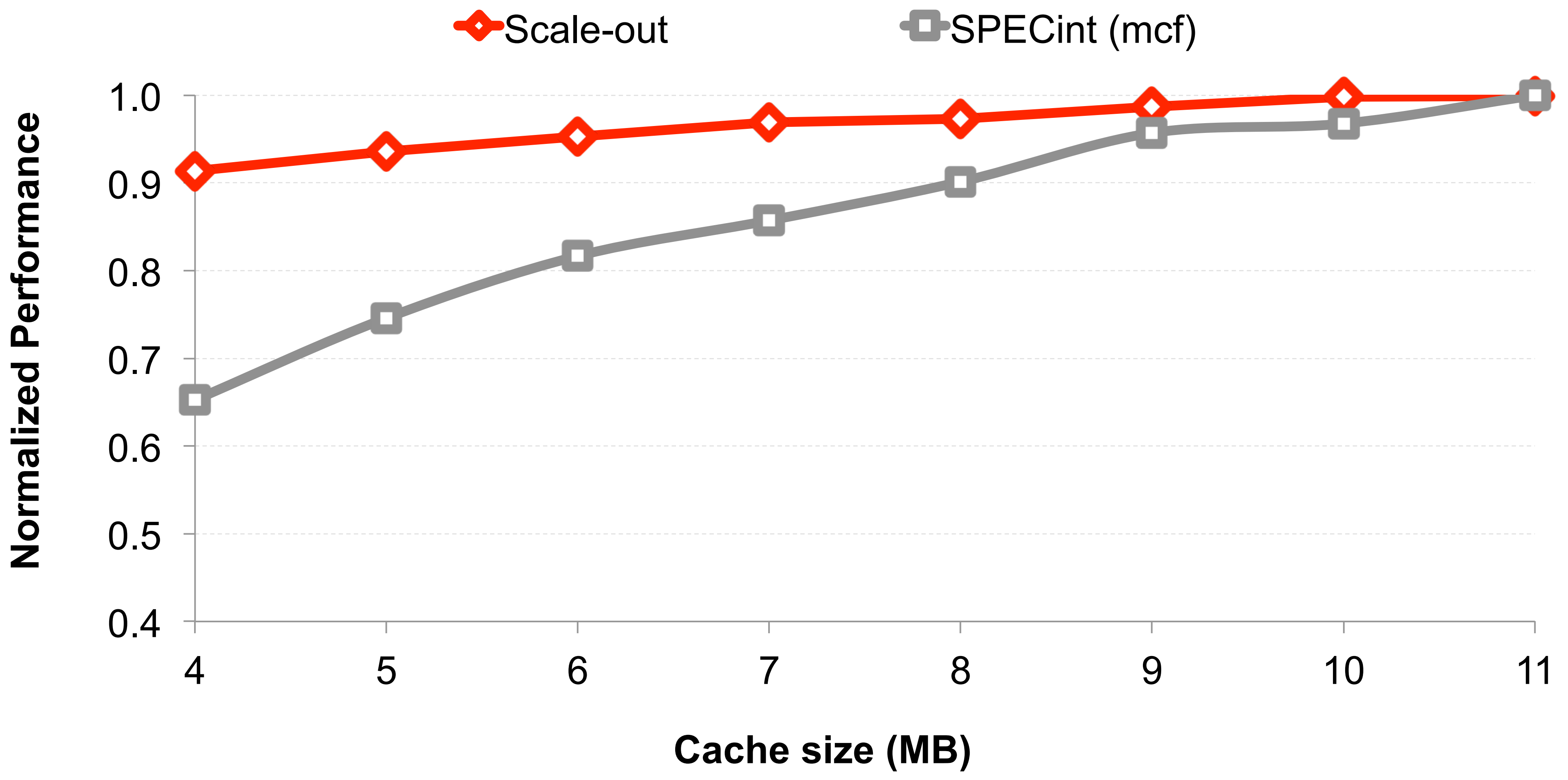
Suffer severe i-cache miss penalties

Instruction-Fetch Inefficiencies

- Large instruction working set
  - Larger than L1 & L2 capacity
  - Instructions read from LLC
- Core stalled during i-fetch

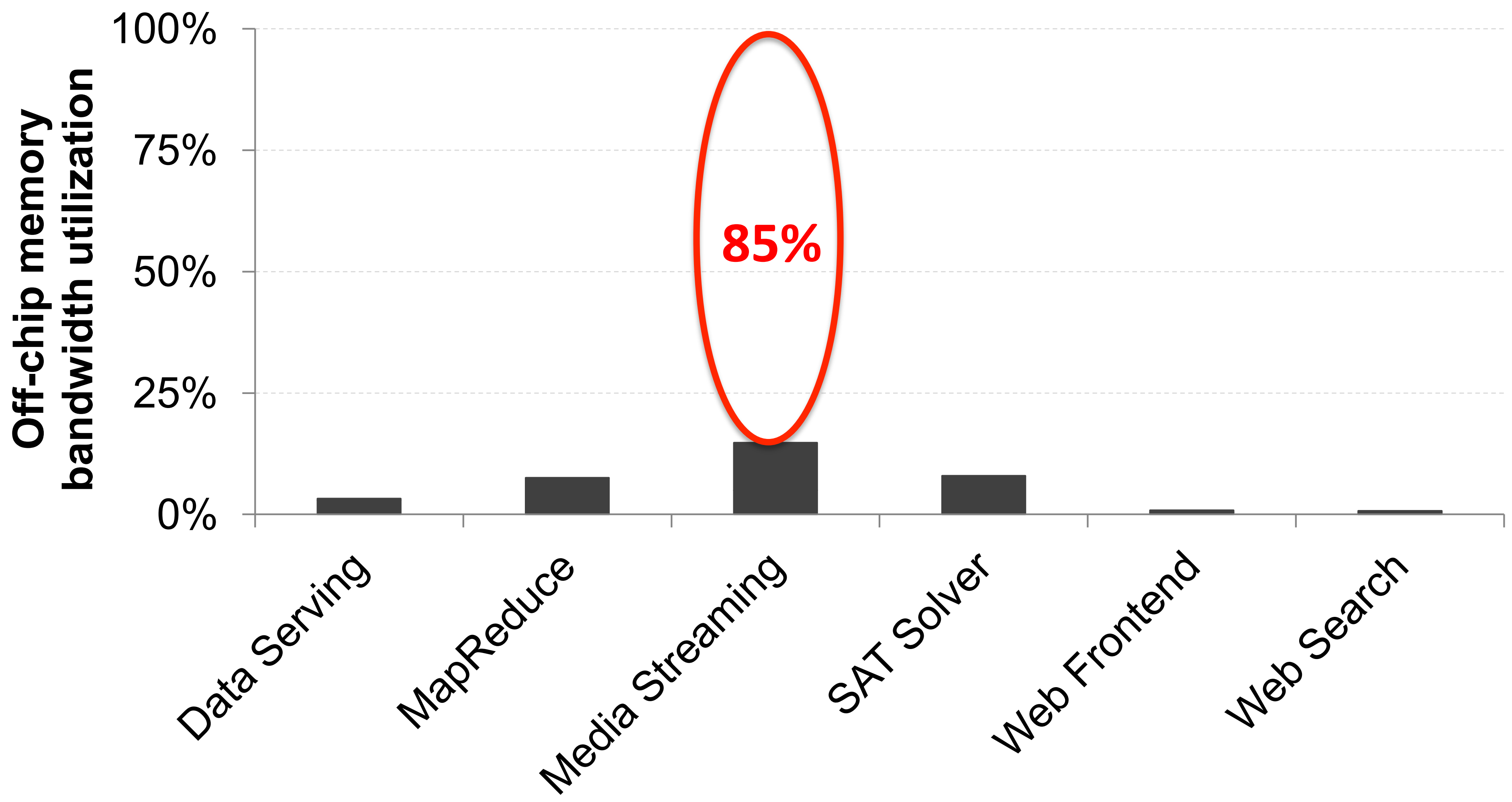


LLC Sensitivity



Minimal performance from large LLC

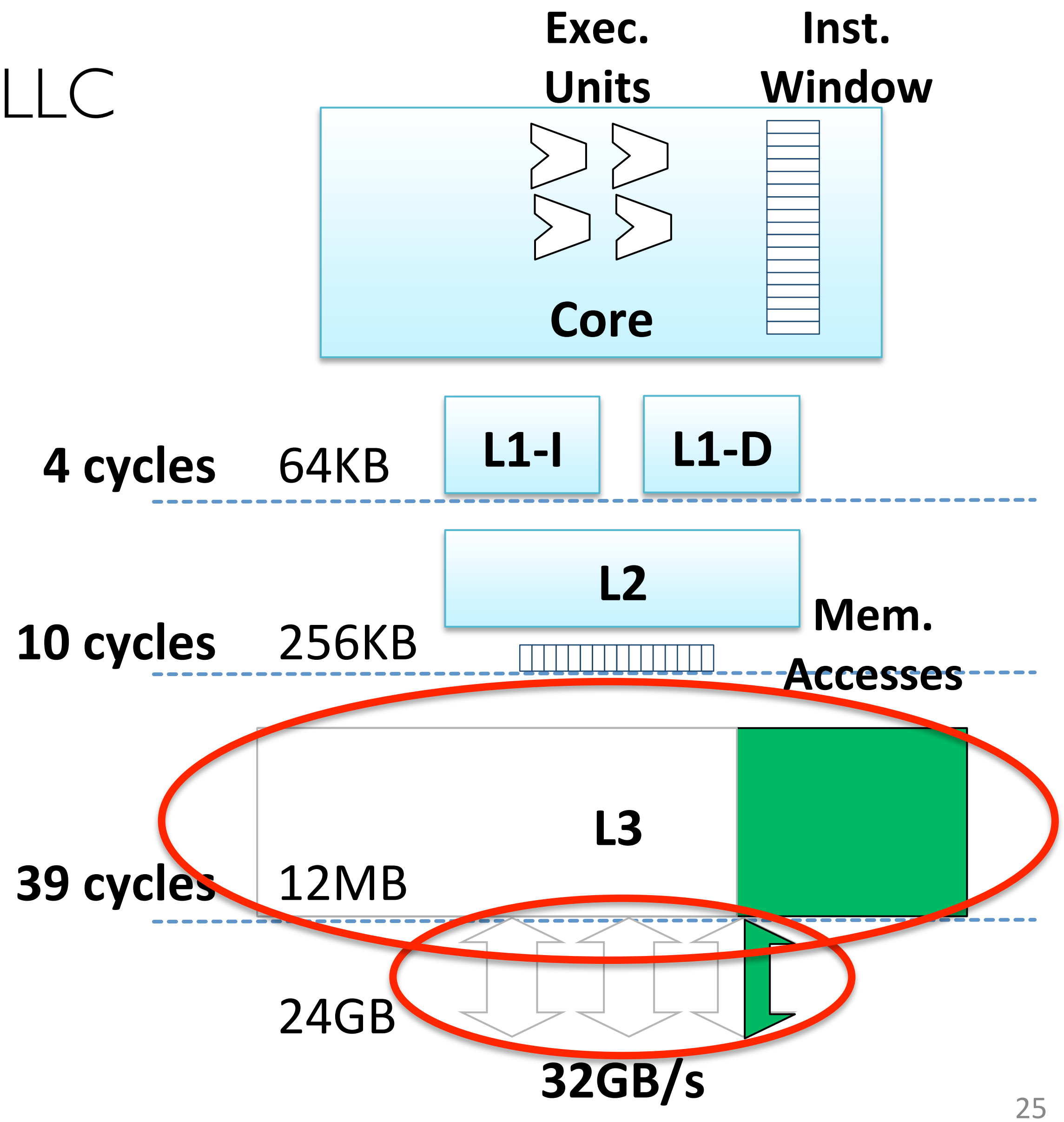
Off-chip Memory Bandwidth



Off-chip BW severely underutilized

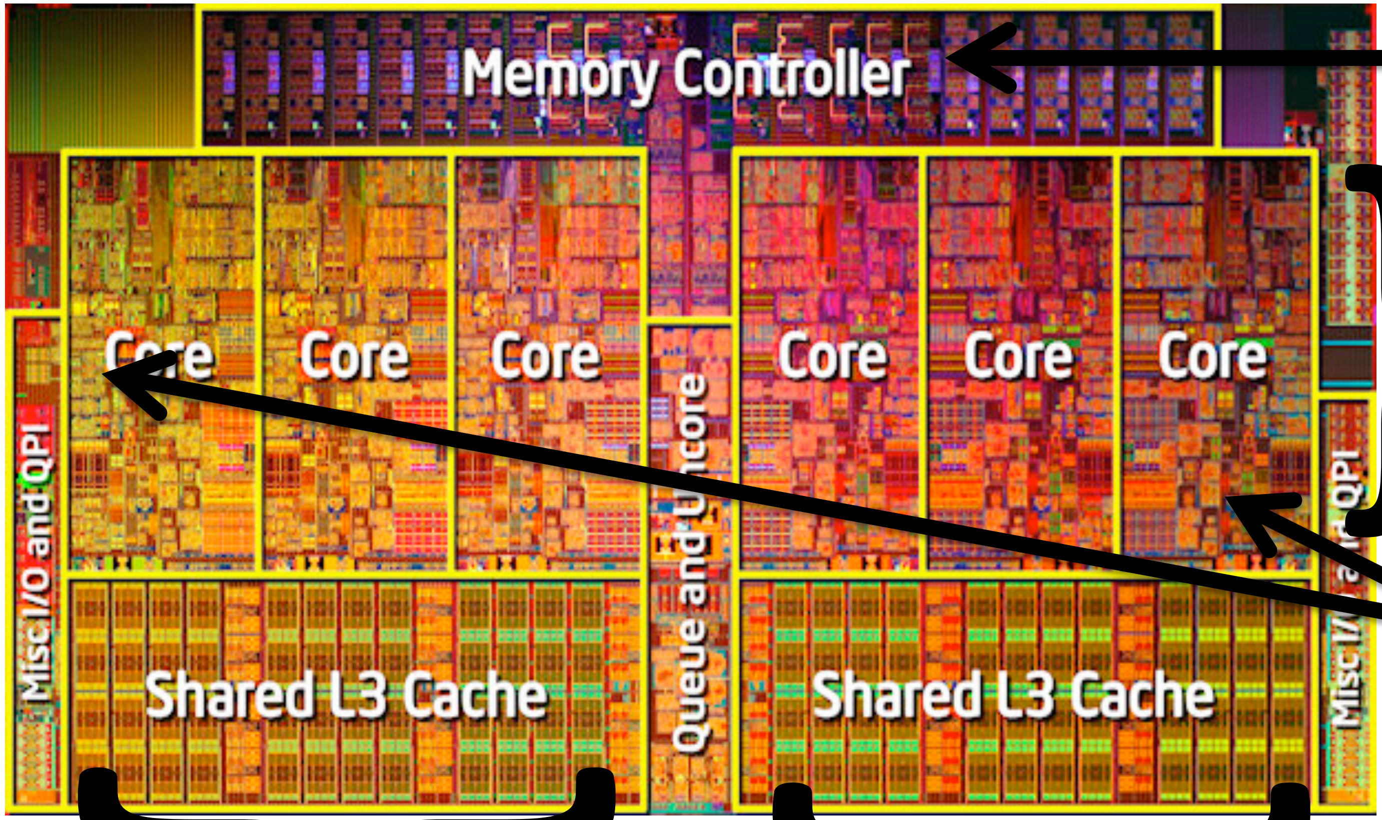
LLC and Bandwidth Inefficiencies

- Scale-out needs modest LLC
  - Beyond 3-4MB useless
  - Area & latency w/o payoff
- Low per-core BW needs
  - <15% utilization
  - Too many channels
  - Too high frequency



Clearing the Clouds in a nutshell [ASPLOS 2012]

Workload/Server Mismatch



B/W unused!

Cores too fat!

Too few cores!

8 MB (60%) waste of space (no reuse)!

Outline

1. Two infliction points colliding
2. How bad are today's servers?
3. How do we build efficient servers?
   - Scale-Out Processors [ISCA'12, IEEE Micro'12]
   - Scale-Out NoCs [MICRO'12]
   - Die-Stacked Caches [ISCA'13]

What do Scale-Out Apps Need?

Cores share instructions

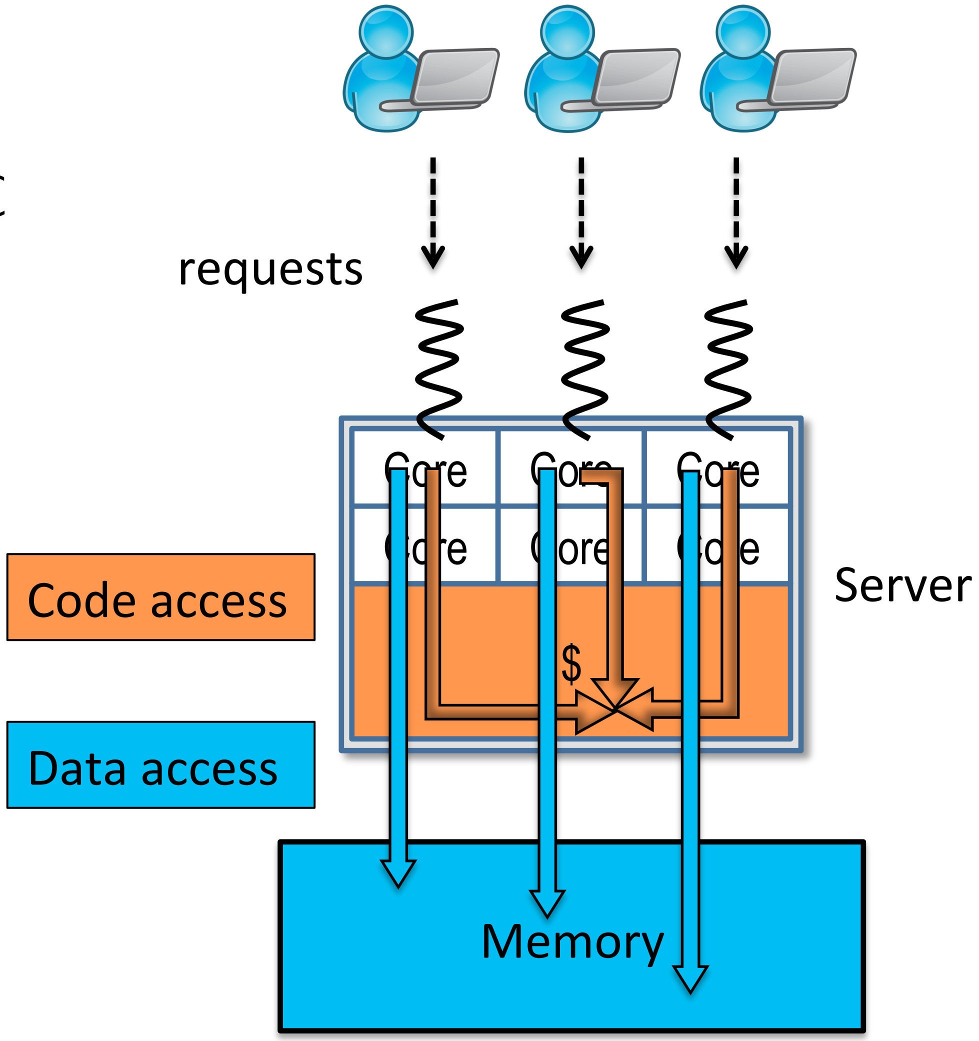
– Large code footprint fits in LLC

Data is in memory

– Data footprint dwarfs LLC

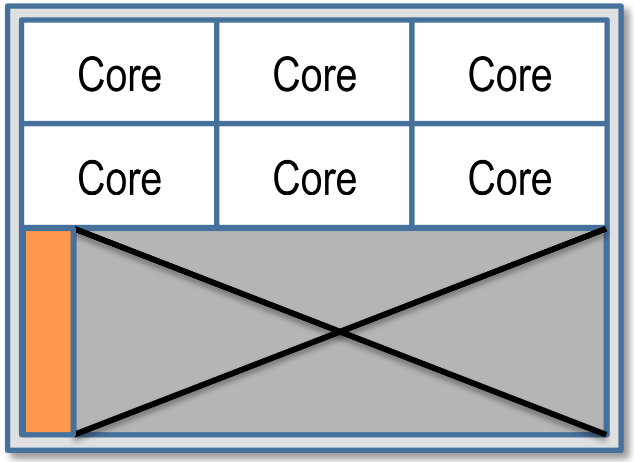
Cores don't communicate

– Independent requests
– Mostly Read-only accesses

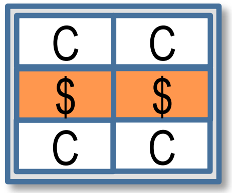


Common traits across applications

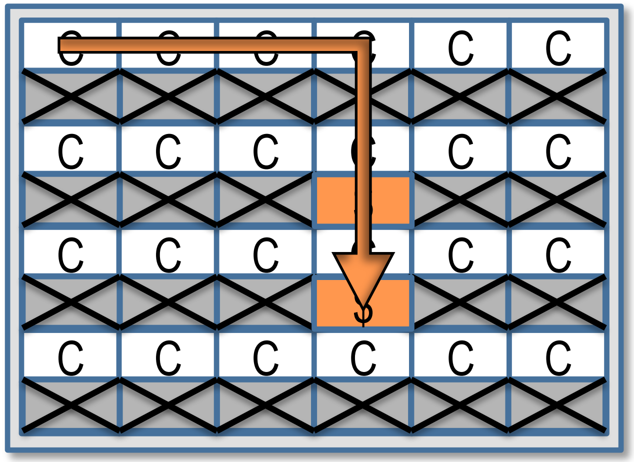
What do Existing Processors Offer?



Intel Xeon (~100 W)



Calxeda (~5W)



Tilera (~30W)

- ✘ Few fat cores
- ✘ Large LLC

- ✘ Few lean cores
- ✓ Compact LLC

- ✓ Many lean cores
- ✘ Large LLC
- ✘ Large distance

Mismatch with workload demands!

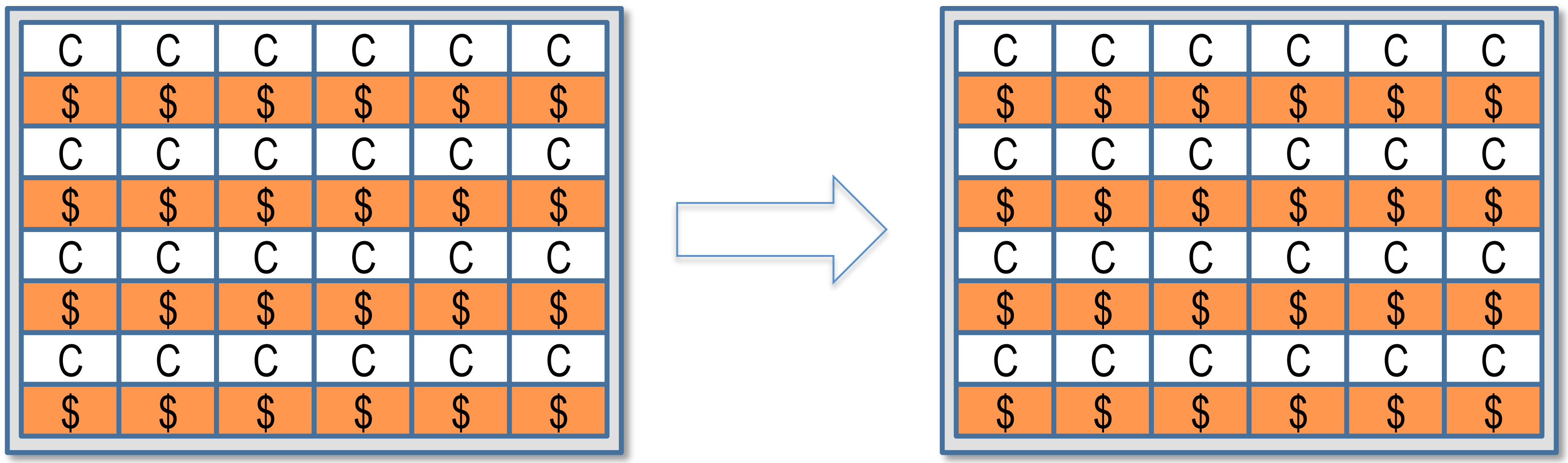
Roadmap for Specialization: Scale-Out Processors

[ISCA ‘12]
[IEEE Micro ‘12]

1. Eliminate wasteful capacity in the LLC
   – Shrink the cache, add cores in freed space
2. Reduce delay to LLC
   – Partition the die into independent multi-core *pods*
3. Enable technology scalability
   – Do not interconnect the pods

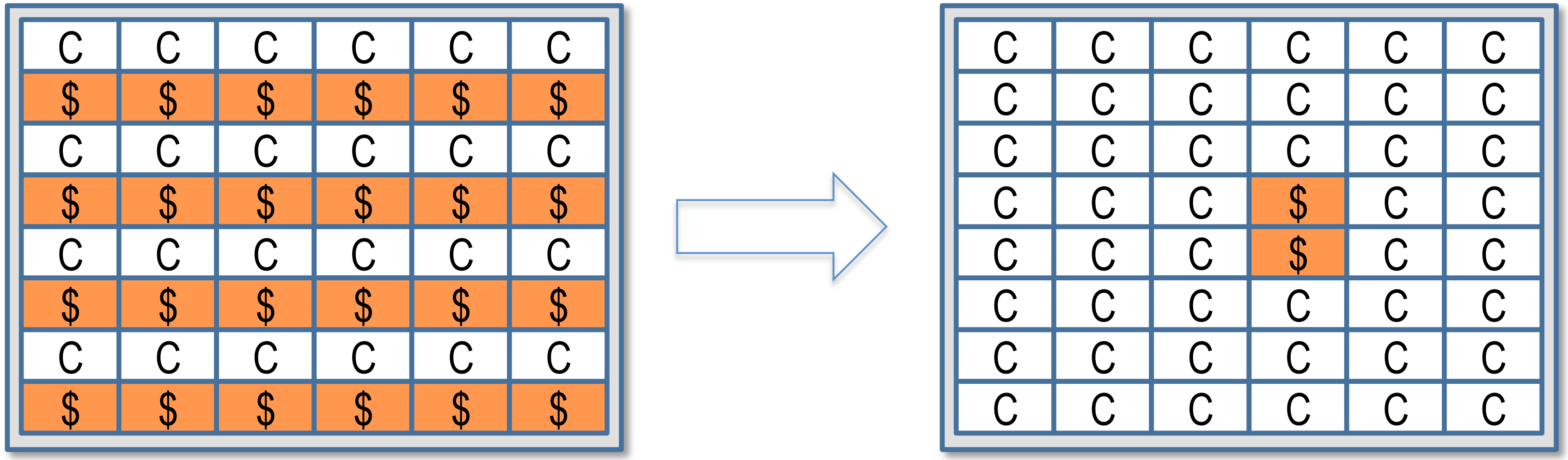
Step 1: Less Cache, More Cores

✓ Small LLC
- Capture instructions
- More area for the cores



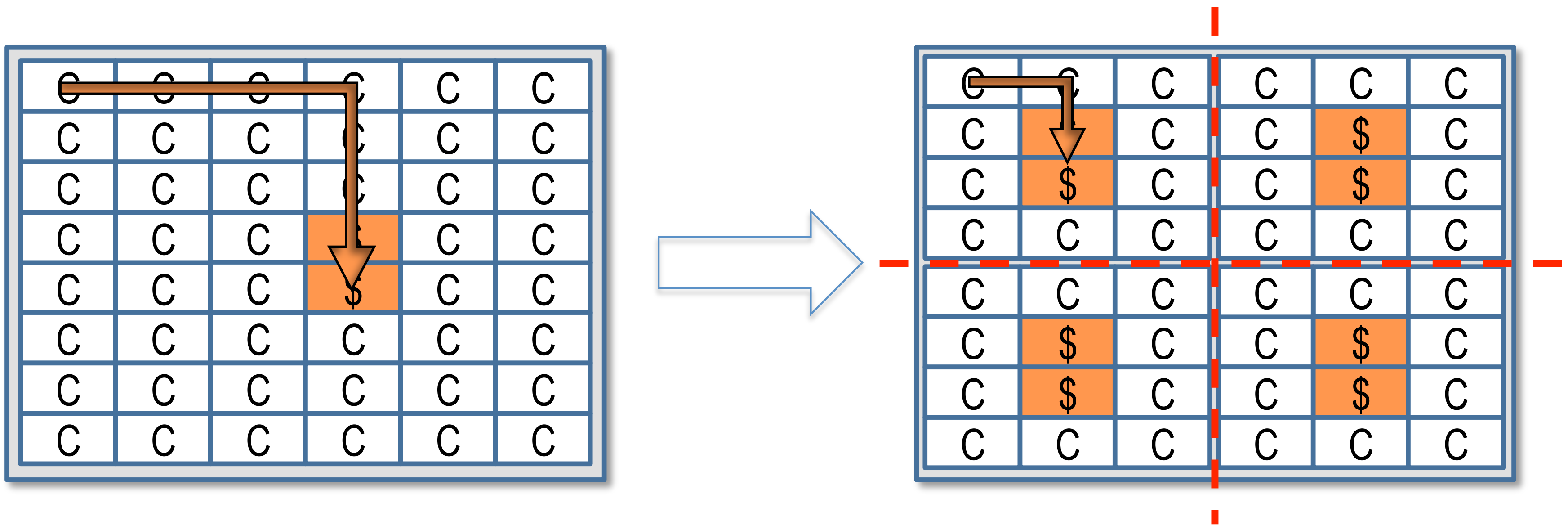
Step 1: Less Cache, More Cores

- ✓ Small LLC
  - Capture instructions
  - More area for the cores
- ✓ More cores for higher throughput



Step 2: Form "Pods" to Reduce Distance

- ✓ Small LLC
- ✓ More cores for higher throughput
- ✘ Fast path to LLC for instructions



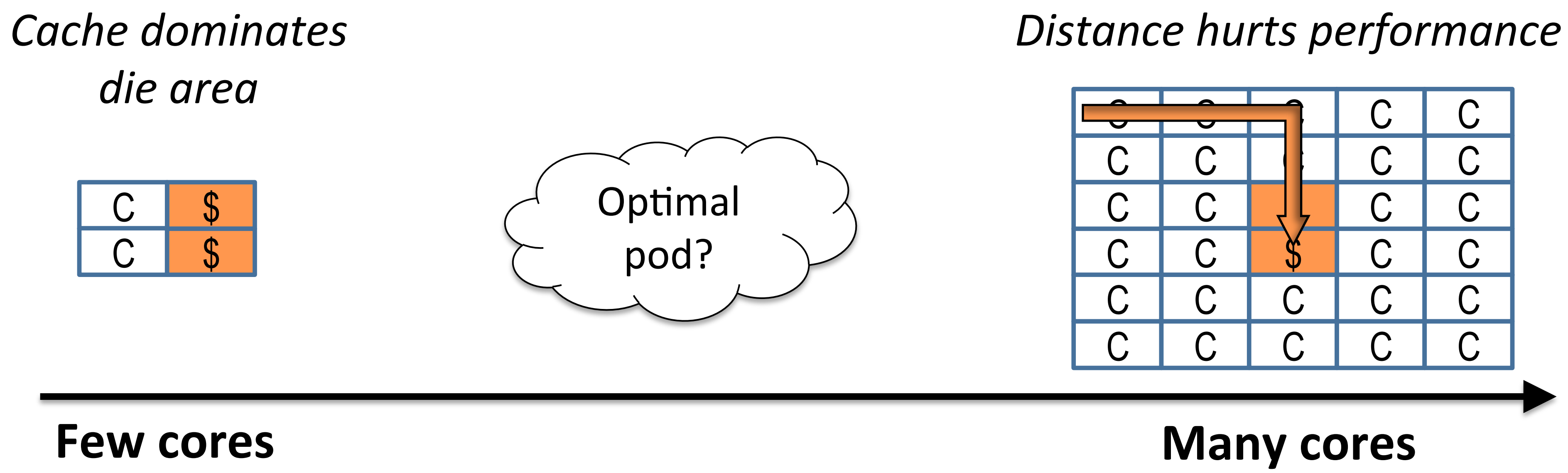
How to choose the optimal pod?

Sizing Pods: Must Optimize for Efficiency

Too few cores ➔ limited parallelism

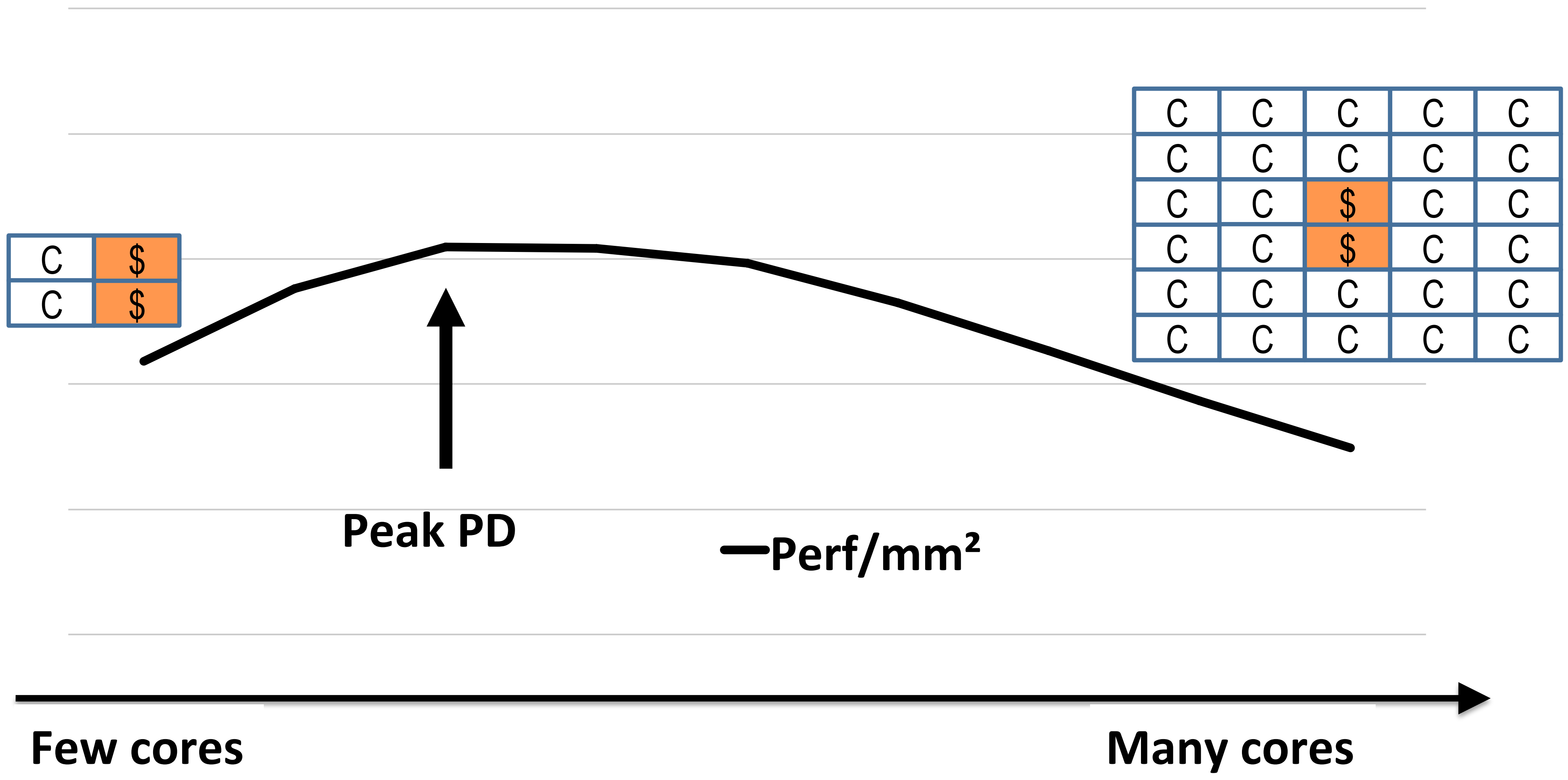
Too many cores ➔ long distance to LLC

- Slower instruction fetch



<span>How do we characterize optimality?</span>

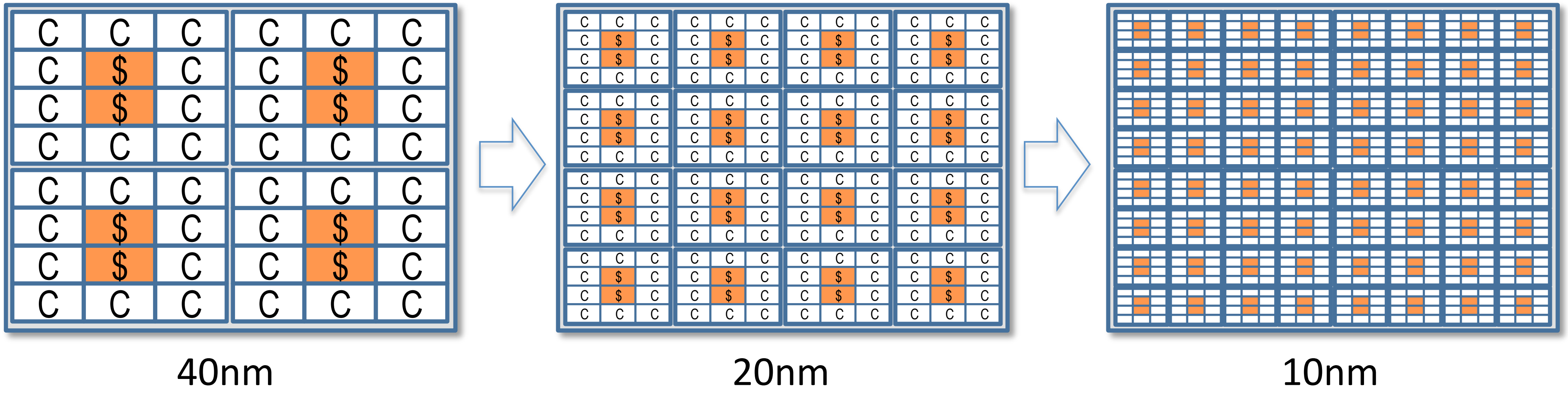
Our Optimization Criterion: Performance Density (PD)



PD pinpoints optimal use of silicon

Step 3: Scale-Out Processors

- One or more pods
- Each pod is a standalone server
  - Runs a full software stack
- No inter-pod connectivity or coherence
  - Scalability and optimality across generations



Inherently optimal & scalable

Methodology

Evaluated designs: Conventional, Small-Chip, Tiled, SOP

Flexus sim'n infrastructure [Wenisch '06]

TCO model [Hardy '11]

Chip components

- Technology node: 20 nm
- Core: 1.1 mm^2
- 1MB cache: 1.2 mm^2
- Mem channel: 12 mm^2

Chip budget

- Die area: 280 mm^2
- Power: 95W
- BW: 6 x 3.2 GT/s DDR4

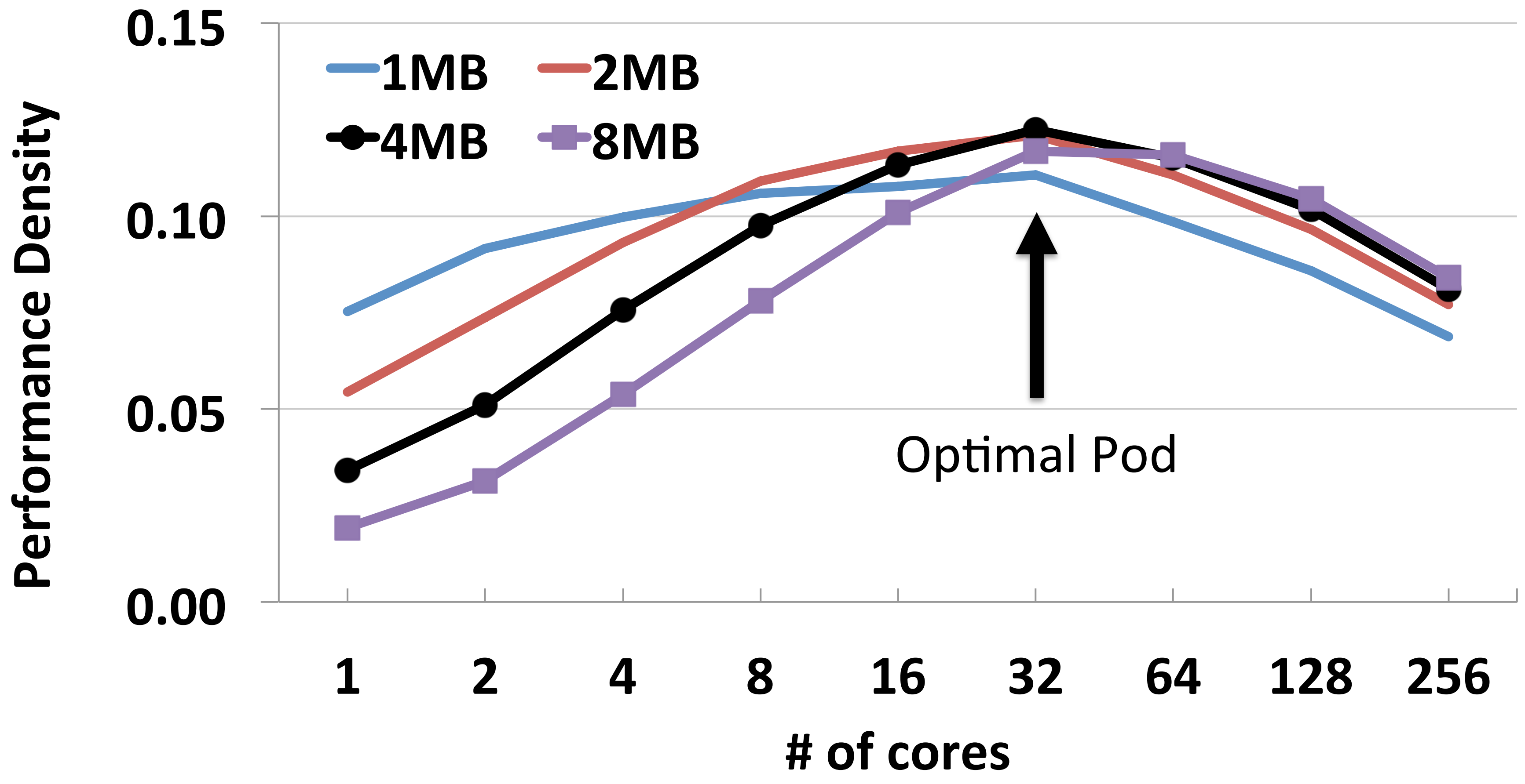
Core Parameters

- Out-of-order, 3-way, 2 GHz
- L1 (I & D): 32KB, 2-way

Datacenter

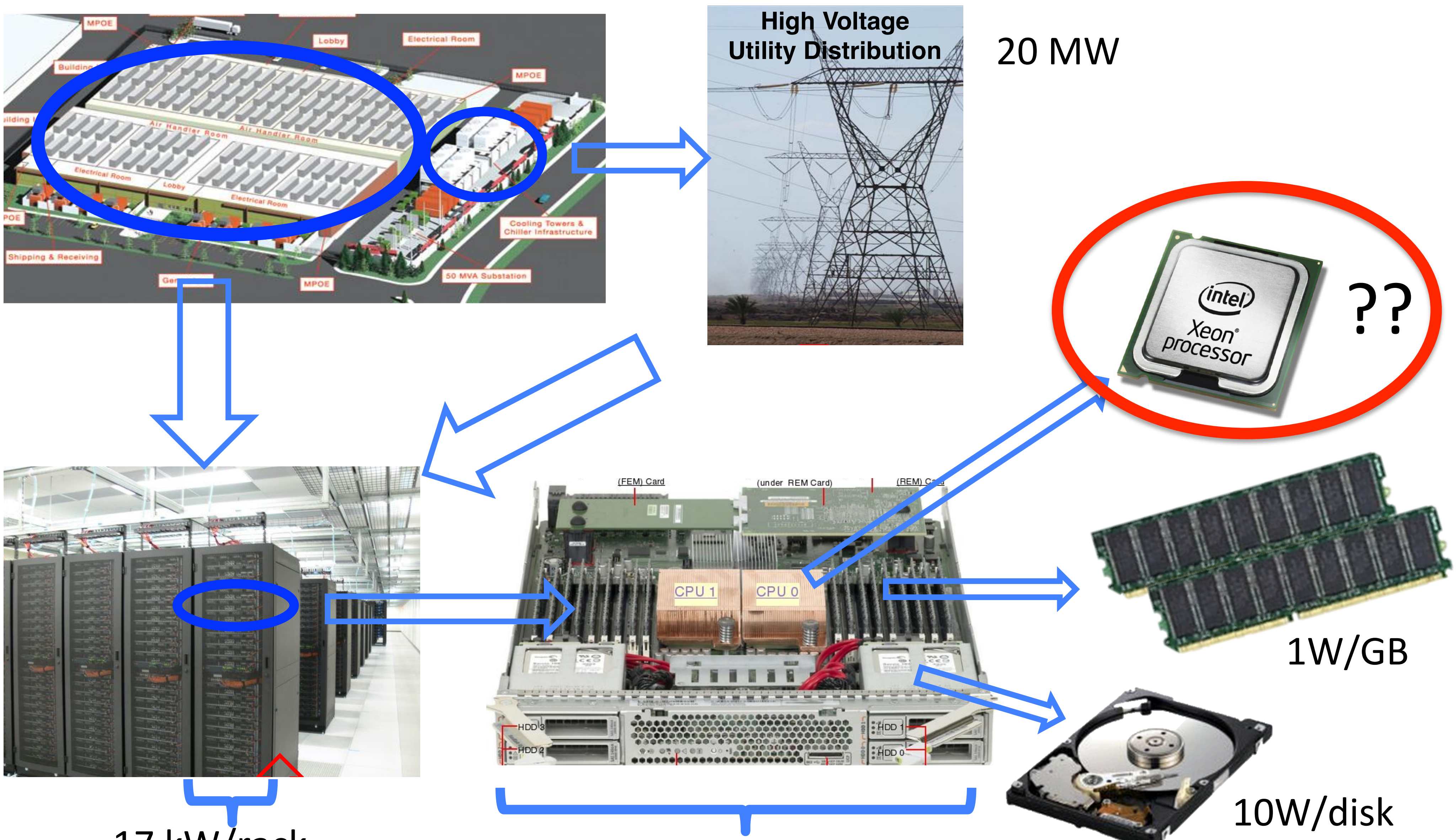
- Power: 20 MW
- Rack: 42 U, 17 kW

Pod Design Space Exploration



Optimal Pod: 32 cores & 4MB of cache

Datacenter-level Evaluation Methodology



20 MW

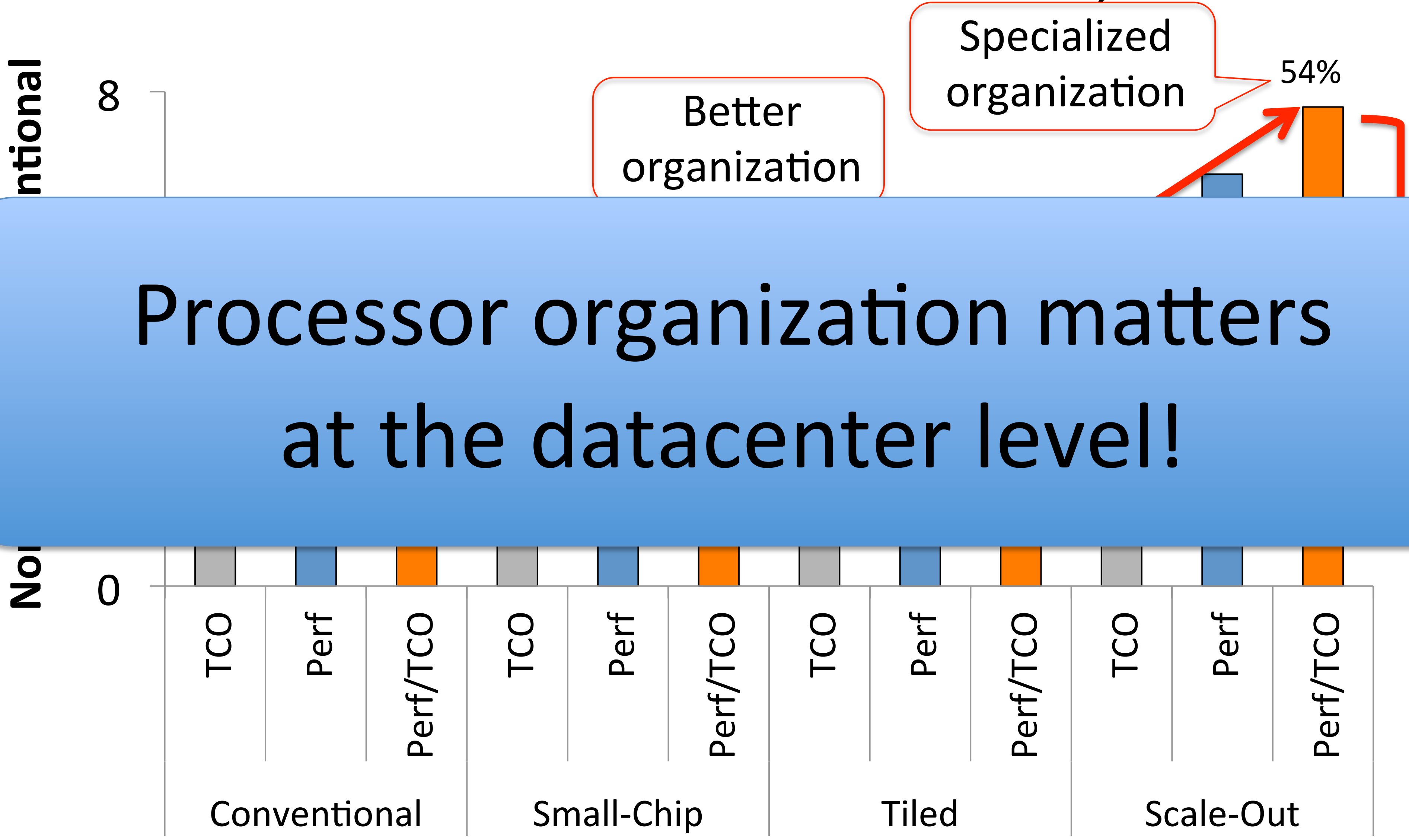
1W/GB

10W/disk

17 kW/rack

~400W per blade

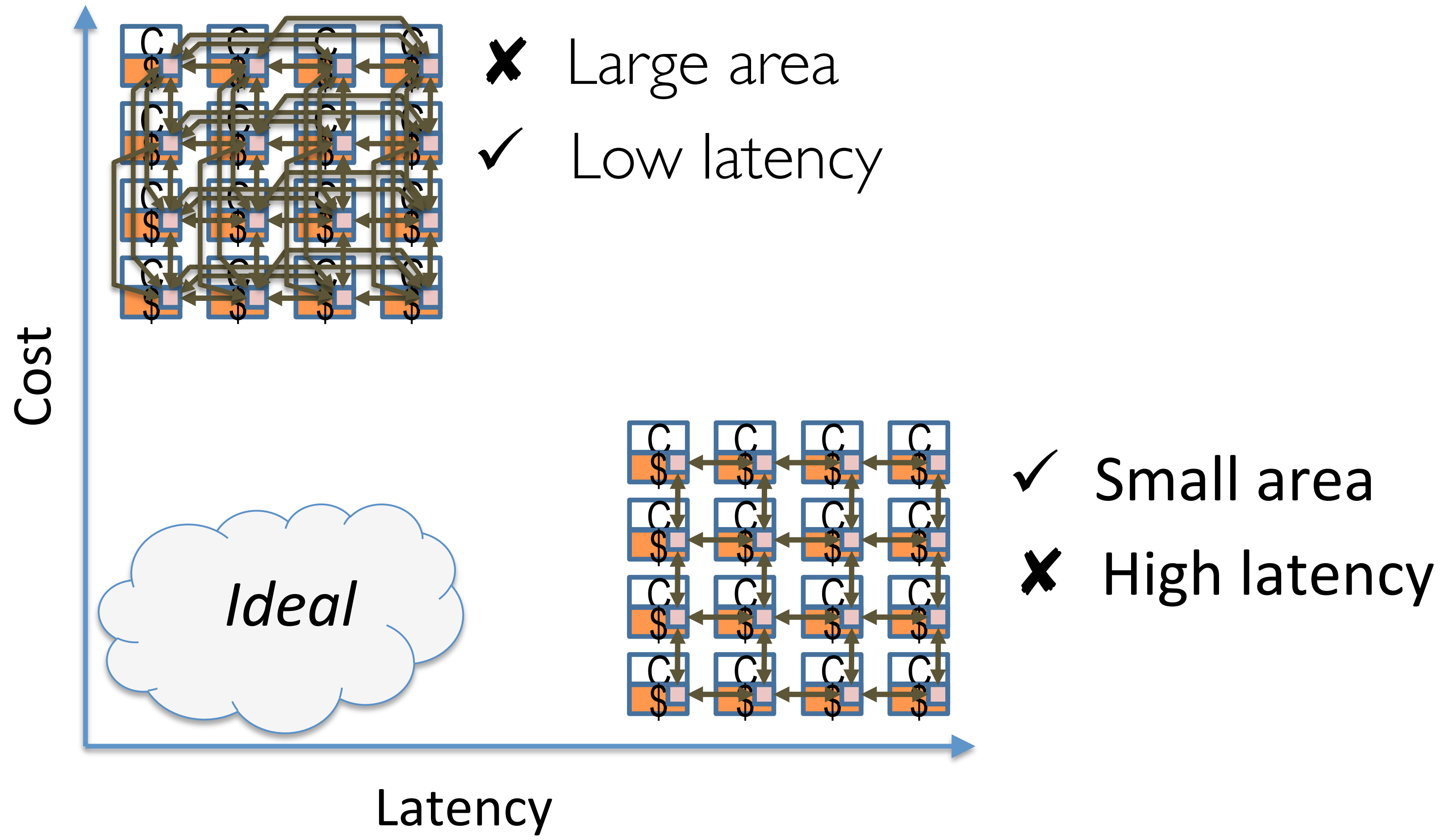
Datacenter Efficiency



Processor organization matters at the datacenter level!

Scale-Out: highest performance/TCO

Network-on-Chip for Pods: Off-the-shelf Designs Not Ideal



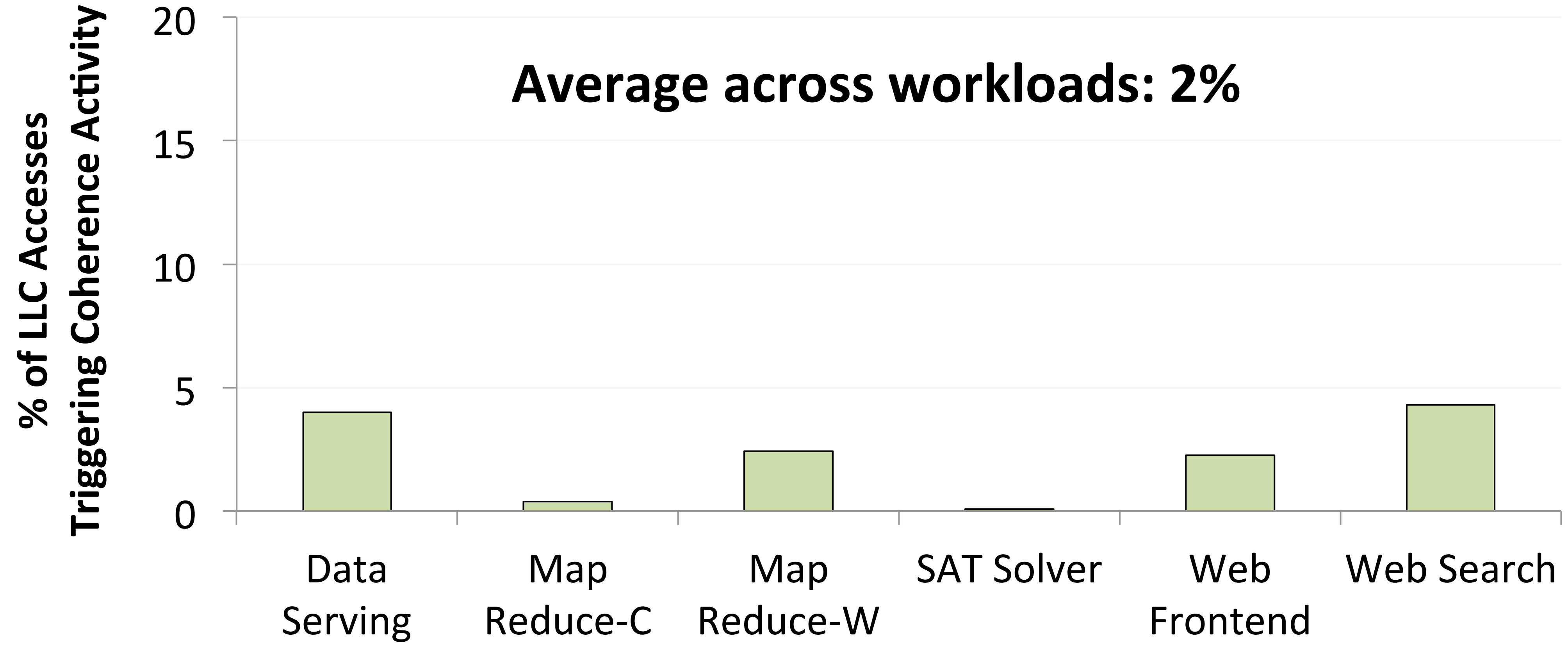
Exploit workload characteristics for efficiency

How do Cores Access Data?

Where are the instructions and data?

- Instructions: in LLC
- Data: in memory

} Nothing useful in remote L1 caches!



Roadmap for an “Ideal” NoC

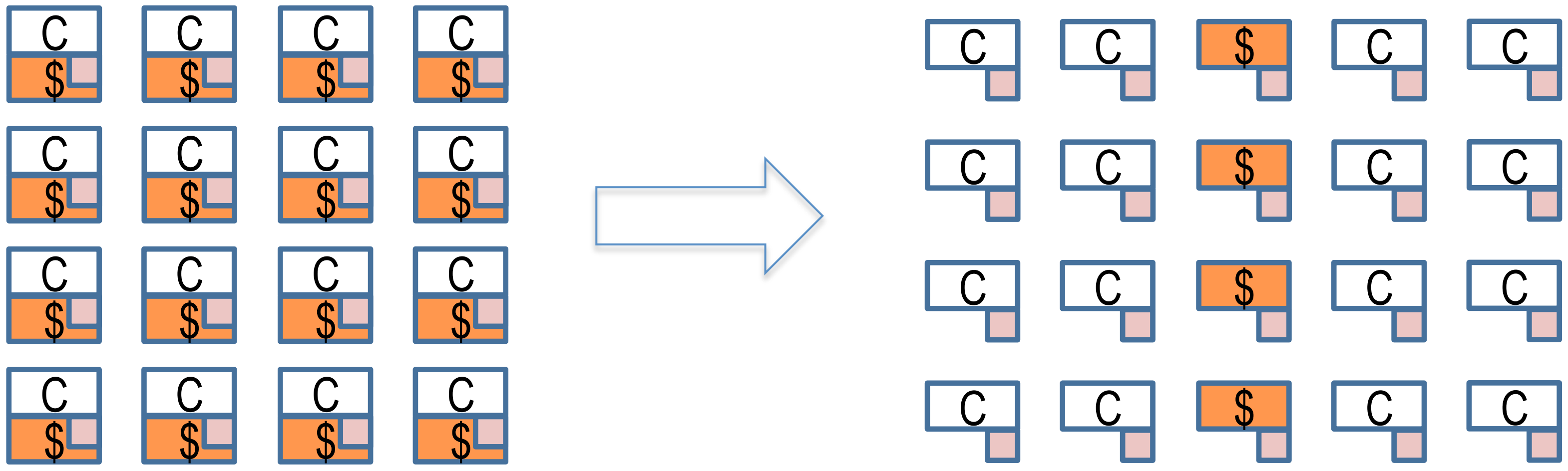
1. LLC in the center
   - Decouple cores and LLC banks

2. Eliminate core-to-core links
   - Leverage the *bilateral* traffic pattern to lower cost

3. Specialize the NOC
   - Maximize performance-to-cost

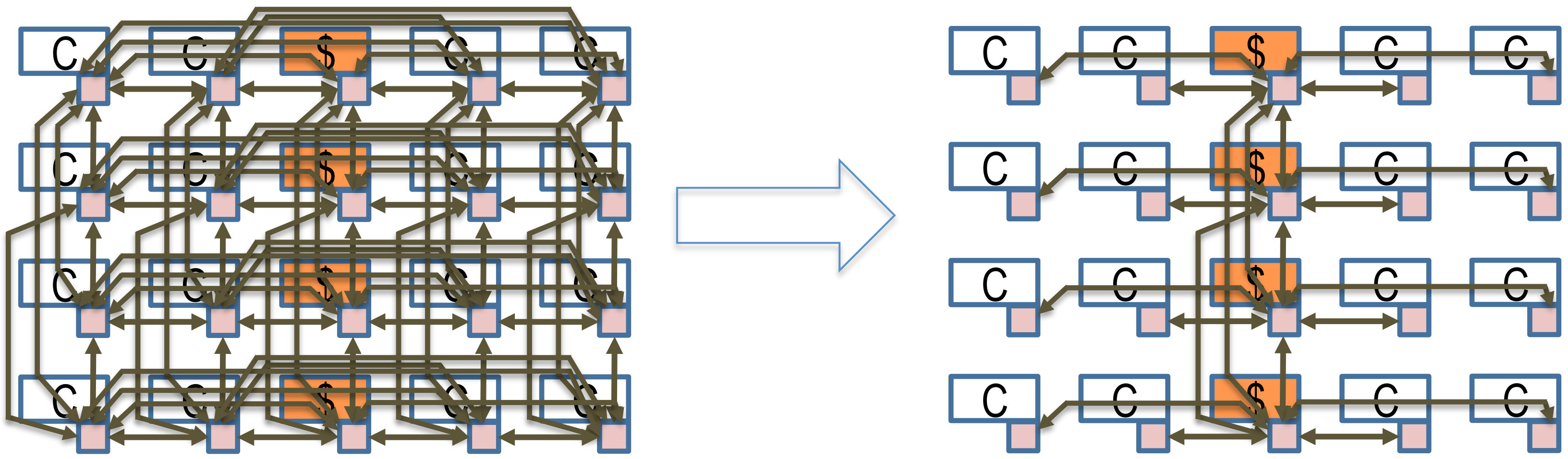
Step 1: LLC in the Center

- ✓ Decouple core and LLC tiles
  - – Natural fit for the bilateral traffic pattern



Step 2: Limit Connectivity

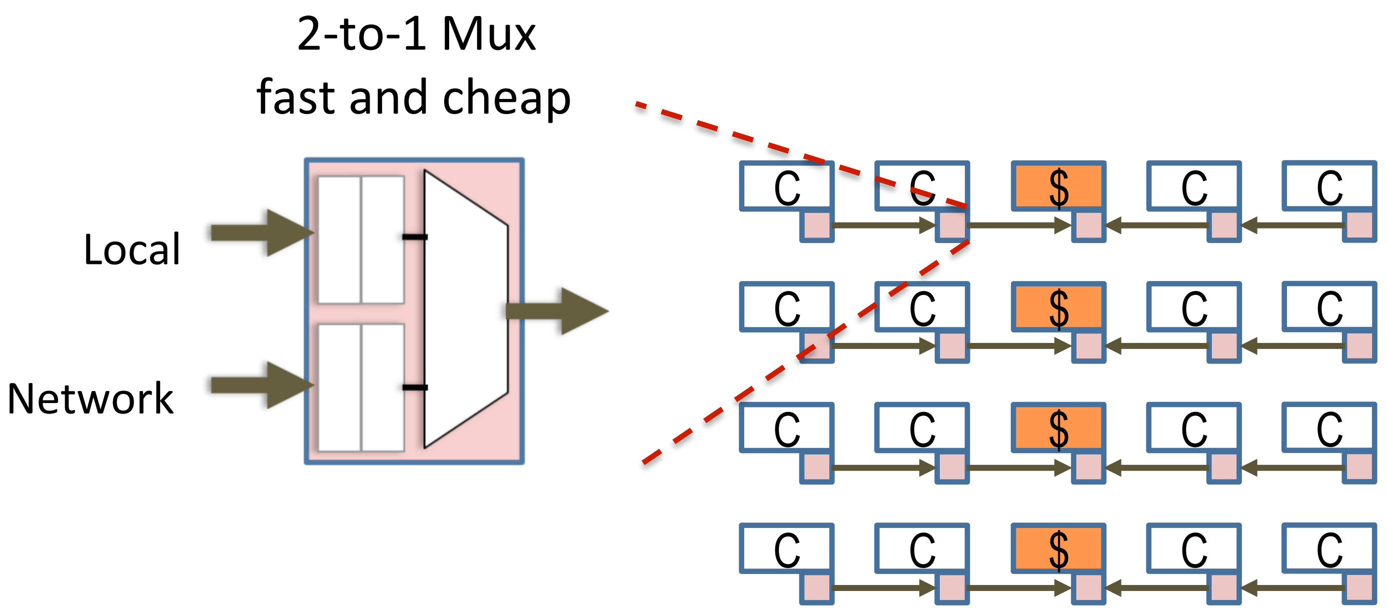
✓ Core-to-core connectivity not needed
- – Lower cost & complexity
- – Unperturbed performance



Step 3: Specialize the NOC

Request network

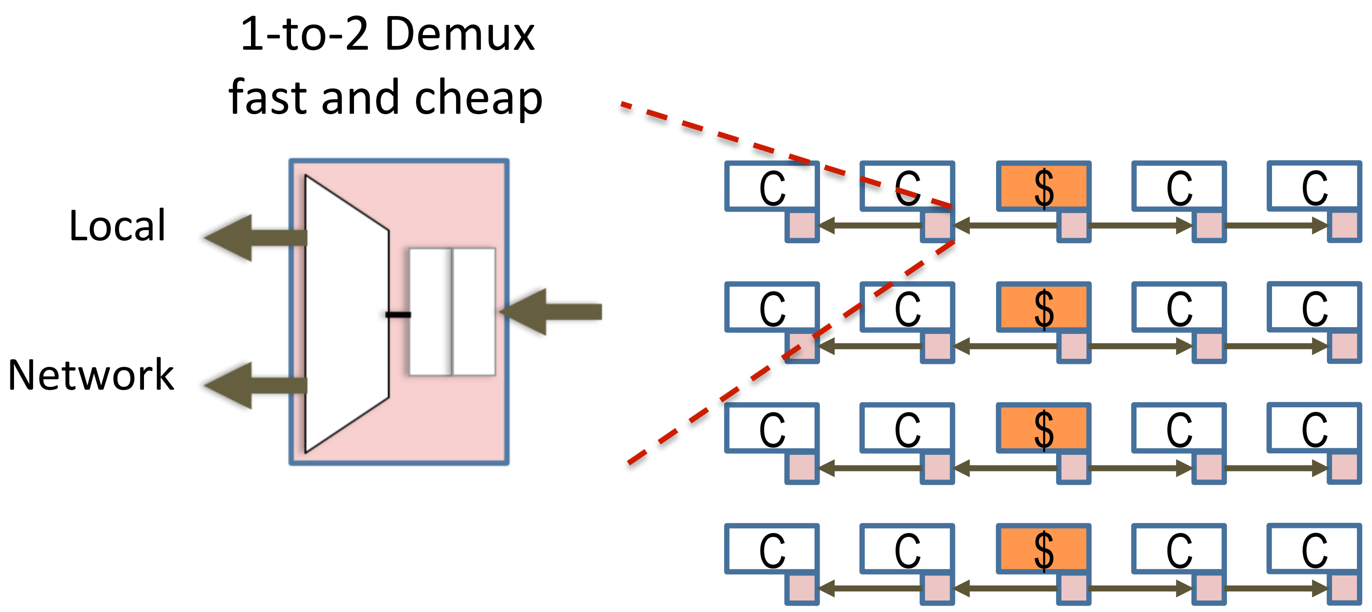
- Tree topology
- Each node: 2-to-1 flow-controlled mux



Step 3: Specialize the NOC

Reply network

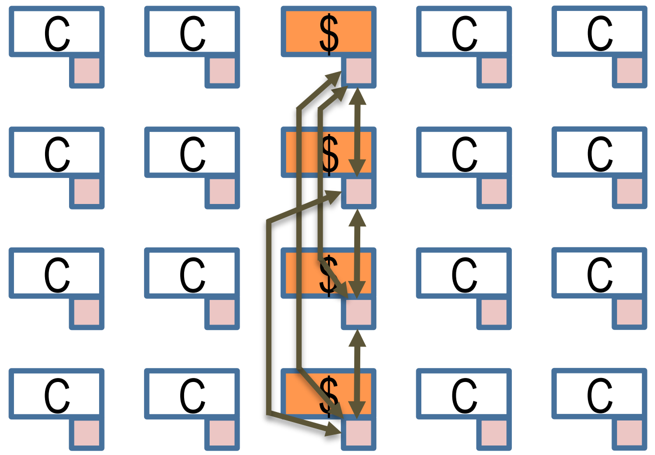
- Tree topology
- Each node: 1-to-2 flow-controlled demux



Step 3: Specialize the NOC

LLC network

- Flattened butterfly topology
- High BW & low latency for convergent traffic
- Expense limited to a fraction of the die



NOC-Out: [MICRO'12]
Near-Ideal Pod Design

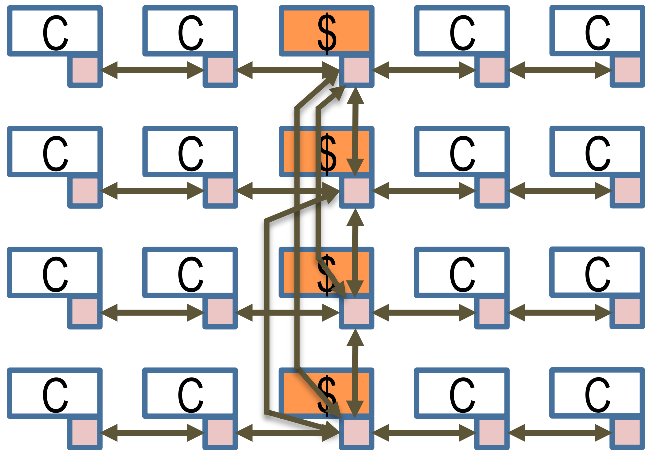
Request & Reply networks: tree topology

– Limited connectivity for efficiency

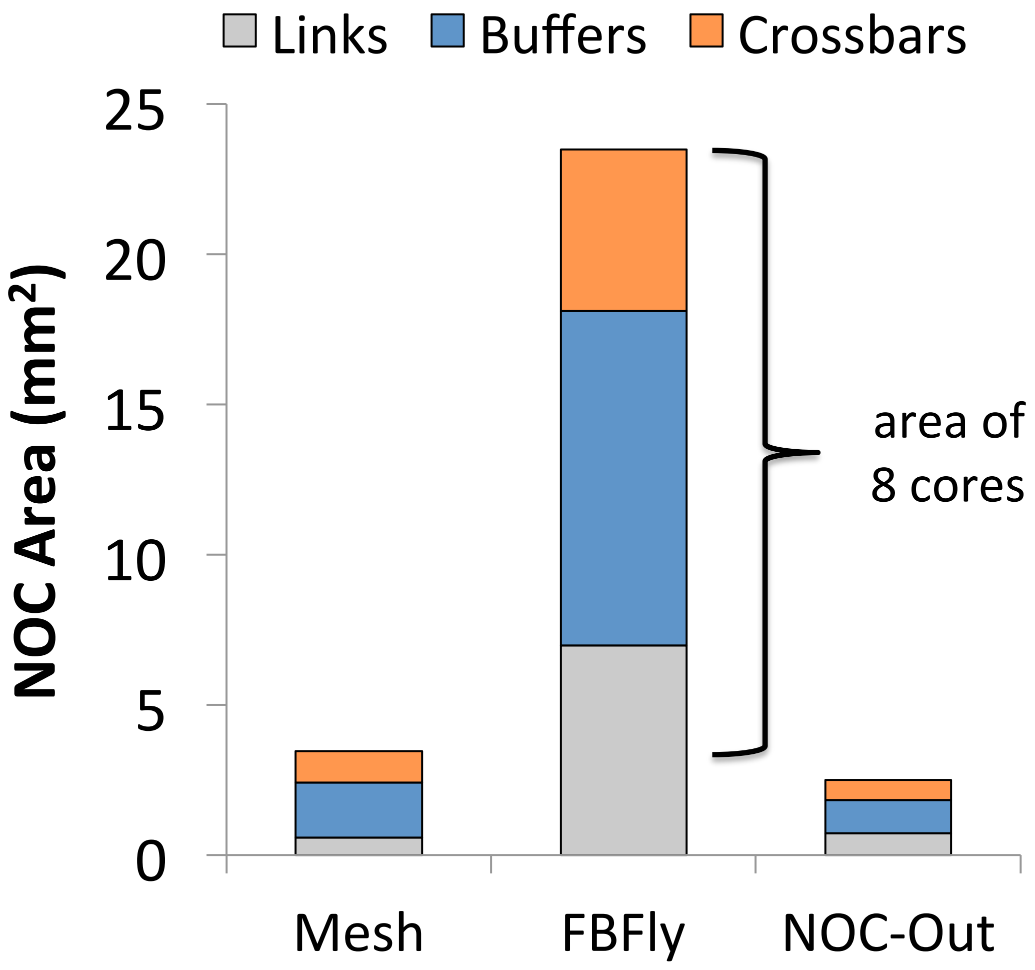
– SW stack unaffected

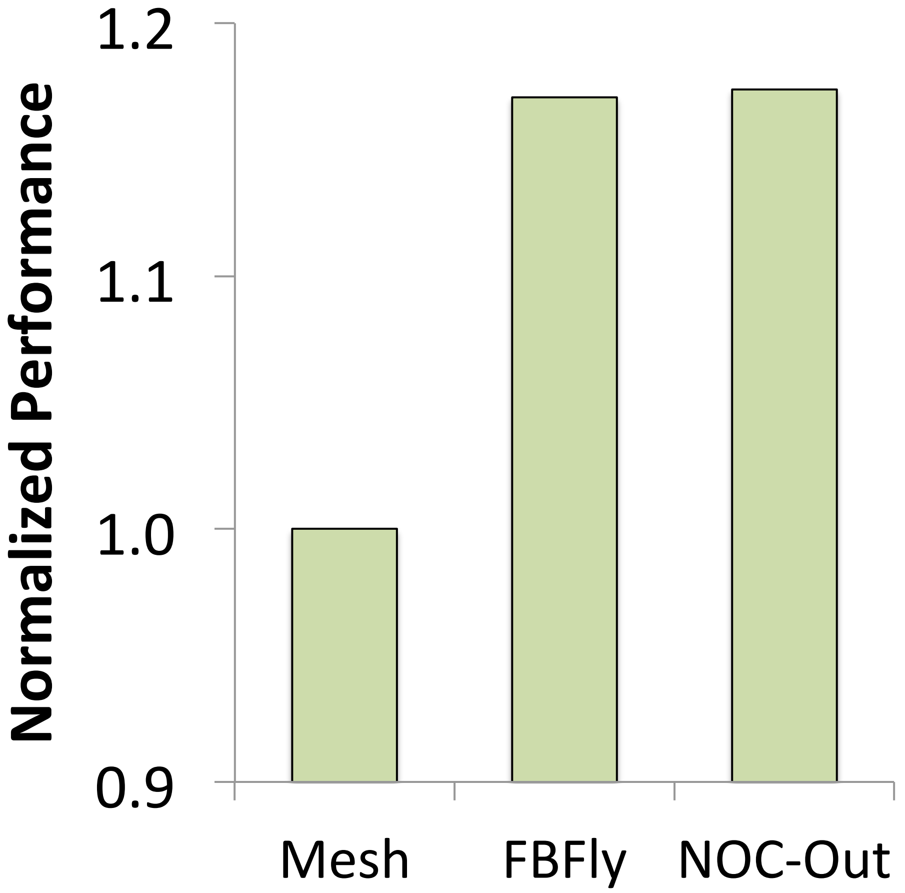
LLC network: flattened butterfly topology

– Expense limited to a fraction of the die



NOC-Out Evaluation Highlights



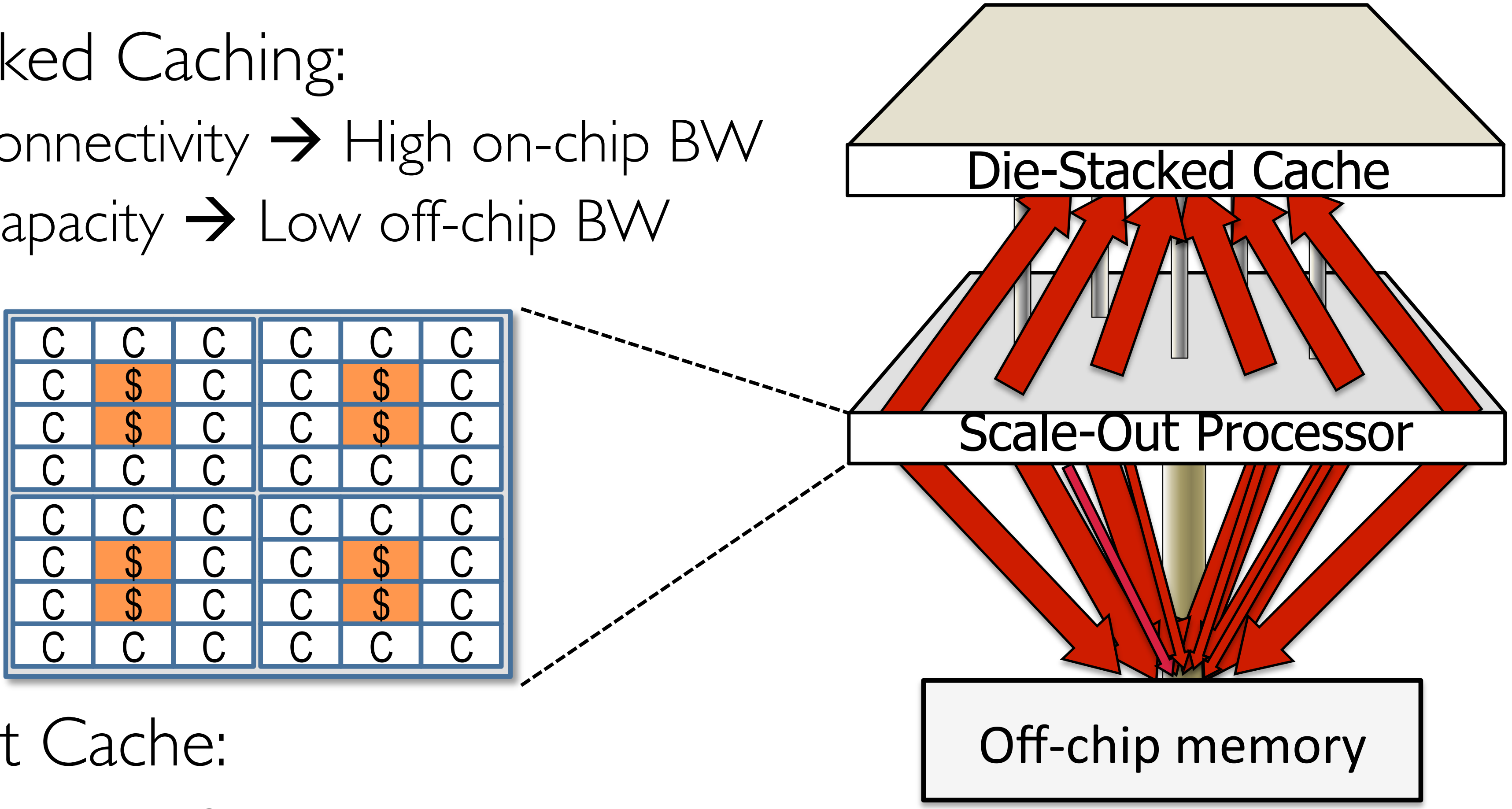


NOC-Out: FBfly's performance at 1/10th cost

Footprint Cache: [ISCA'13]
Effective Die-Stacked Caching for Servers

Die-Stacked Caching:

- Rich connectivity → High on-chip BW
- High capacity → Low off-chip BW



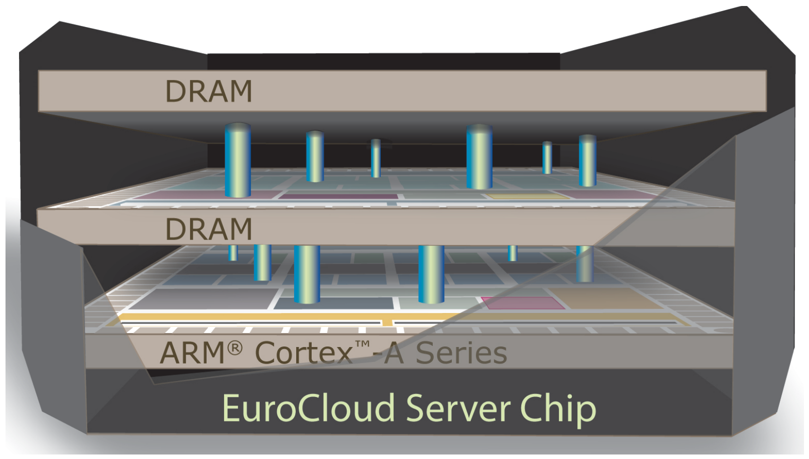
Footprint Cache:

- Allocate tags for pages
- Predict & fetch page's footprint

3D Scale-Out Chip: An Architecture for Big Data (www.eurocloudserver.com)

Mobile efficiency in servers

– Swarms of ARM cores
– Die-stacked cache
– 10x more efficiency
– Runs Linux LAMP stack



Coming soon to Datacenter near you!

Demoed to EC Parliament!

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE · ARM · SEVENTH FRAMEWORK PROGRAMME · NOKIA Connecting People · imec · UCyprus

Summary

Two IT trends on a collision course:

- Data growing at ~10x/year
- Nearing end of Dennard & Multicore Scaling
- Need technologies to bring efficiency to data

Scale-out Processors: near-term efficiency gains

- Disconnected Architecture/NoC-optimized Pods
- Die-stacked Caches to alleviate bandwidth

Long term:

Integrate + Specialize + Approximate (ISA for Big Data)

Thank You!

For more information please visit us at

ecocloud.ch

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE