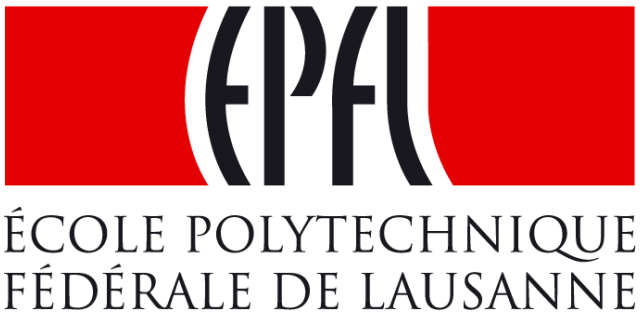
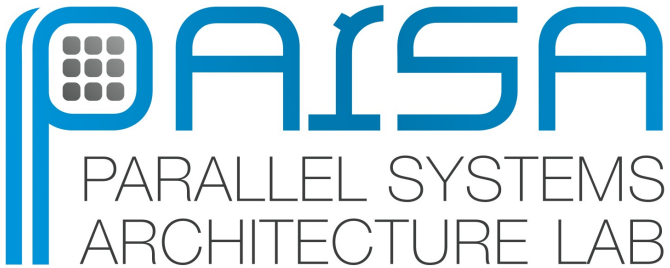
Dark Silicon & its Implication on Server Design

Babak Falsafi

Parallel Systems Architecture Lab
EPFL
parsa.epfl.ch
ecocloud.ch





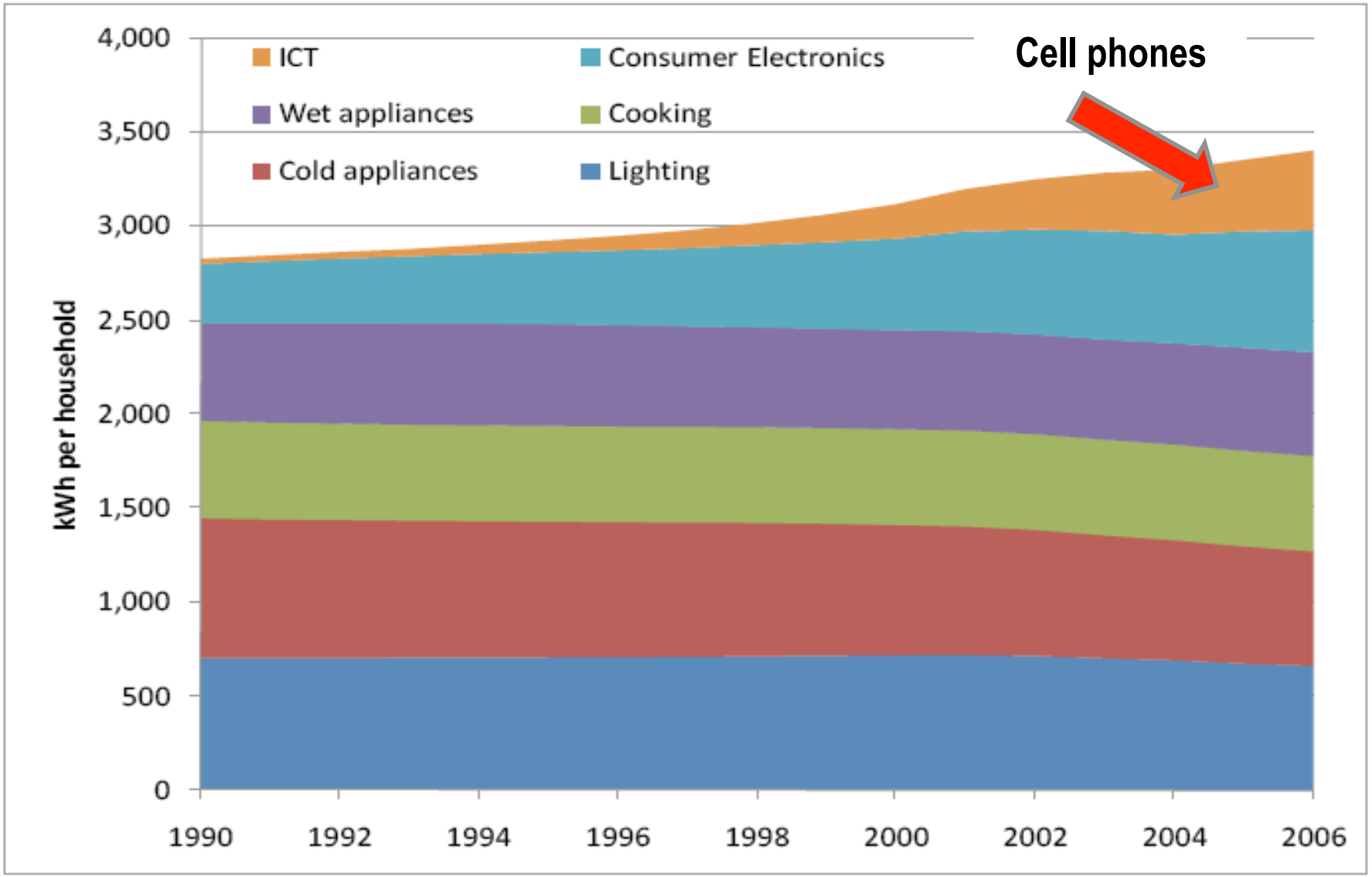
Energy: Shaping IT's Future

- 40 years of energy scalability
  - Doubling transistors every two years
  - Quadratic reduction in energy from voltages

- But, while Moore's law continues
  - Voltages have started to level off
  - ITRS projections in 2000 for voltage levels in 2009 are way off!

An exponential increase in energy usage every generation?

Household IT Energy Usage (from Sun)



Source: BERR (2008) *Energy consumption in the UK*

Shift towards Cloud Computing Helps

- Ubiquitous connectivity & access to data
- Consolidate servers ➔ Amortize energy costs

But, the Cloud has hit a wall!

Trends:

- Moore's law continues
- Server density is increasing
- But, voltage scaling has slowed

➔It's too expensive to buy/cool servers

A 1,000m^2 datacenter is 1.5MW!
(carbon footprint of airlines in 2012)

Enterprise IT Energy Usage

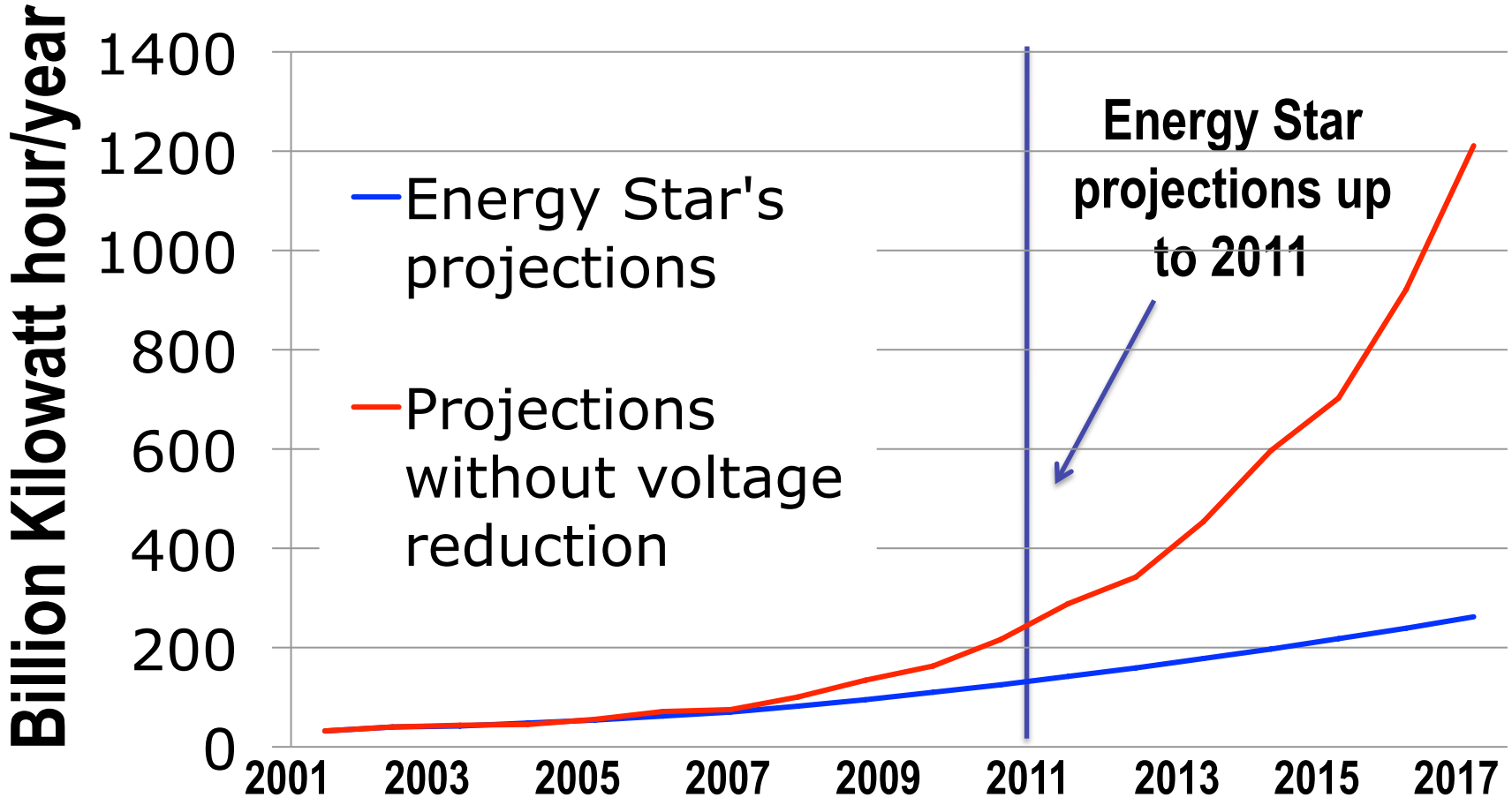
Kenneth Brill (Uptime Institute)

- "Economic Meltdown of Moore's Law"
- In 2012: Energy/server lifetime 50% more than price/server
  - And 2% of all Carbon footprint in the US

Energy Star report to Congress:

- Datacenter energy 2x from 2000 to 2006
- Roughly 2% of all electricity & growing

Example Projections for Datacenters



- Projections for 2011 are already off
- Exponential increase in usage

What lies ahead?

For the next ten years:

- CMOS is still the cheapest technology

But,

- need ~100x reduction in energy just to keep up with Moore's Law

Chip design recommendations:

- Short-term, lean chips (squeeze all fat)
  - Cores, caches, NoC
- Long-term, cannot power up all of chip
  - Live with "dark silicon", specialize

ecocloud.ch

10 faculty, CSEM & industrial affiliates

- HP, Intel, IBM, Microsoft, …

Research:

- Energy-minimal cloud computing
- Elastic data bricks and storage
- Scalable cloud applications & services

Making tomorrow's clouds green & sustainable

Outline

- Where are we?
- ➔Energy scalability for servers
- Where do we go from here?
- Future on-chip caches
- Future NoC's
- Summary

Where does server energy go?

Many sources of power consumption:

- Server only [Fan, ISCA'07]
  - Processors chips (37%)
  - Memory (17%)
  - Peripherals (29%)
  - …
- Infrastructure (another 50%)
  - Cooling
  - Power distribution

How did we get here? Leakage killed the supply voltage

Historically,

$$Power \propto V^2 f$$

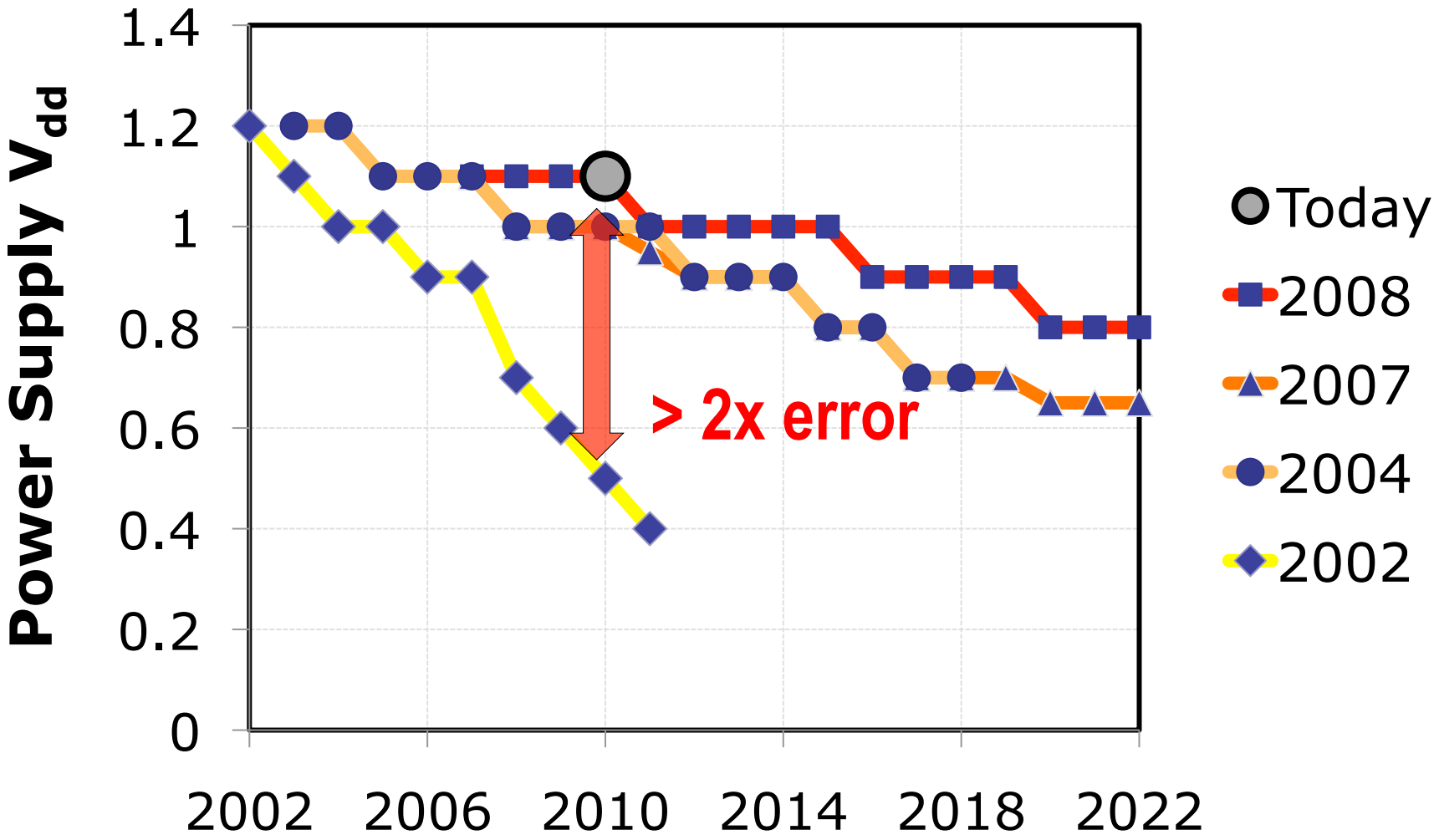
Voltage Frequency

Four decades of reducing V to keep up

But, can no longer reduce V due to leakage!!

- Exponential in area
- Exponential in temperature

Voltages have already leveled off



ITRS estimates for today were off by > 2x

A Study of Server Chip Scalability

Actual server workloads today

- Easily parallelizable (performance-scalable)

Actual physical char. of processors/memory

ITRS projections for technology nodes

Modeled power/performance across nodes

For server chips

- Bandwidth is near-term limiter

→**Energy is the ultimate limiter**

A few words about our model

Physical char. modeled after Niagara

Area: cores/caches (72% die)

- scaled across tech. nodes

Power:

- Active: projected V_{dd}/ITRS
  - Core=scaled, cache=f(miss), crossbar=f(hops)
- Leakage: projected V_{th}/ITRS, f(area), 62C

Performance:

- Parameters from real server workloads (DB2, Oracle, Apache, Zeus)
- Cache miss rate model (validated)
- CPI model based on miss rate

Caveat: Simple Parallelizable Workloads

Workloads are assumed parallel

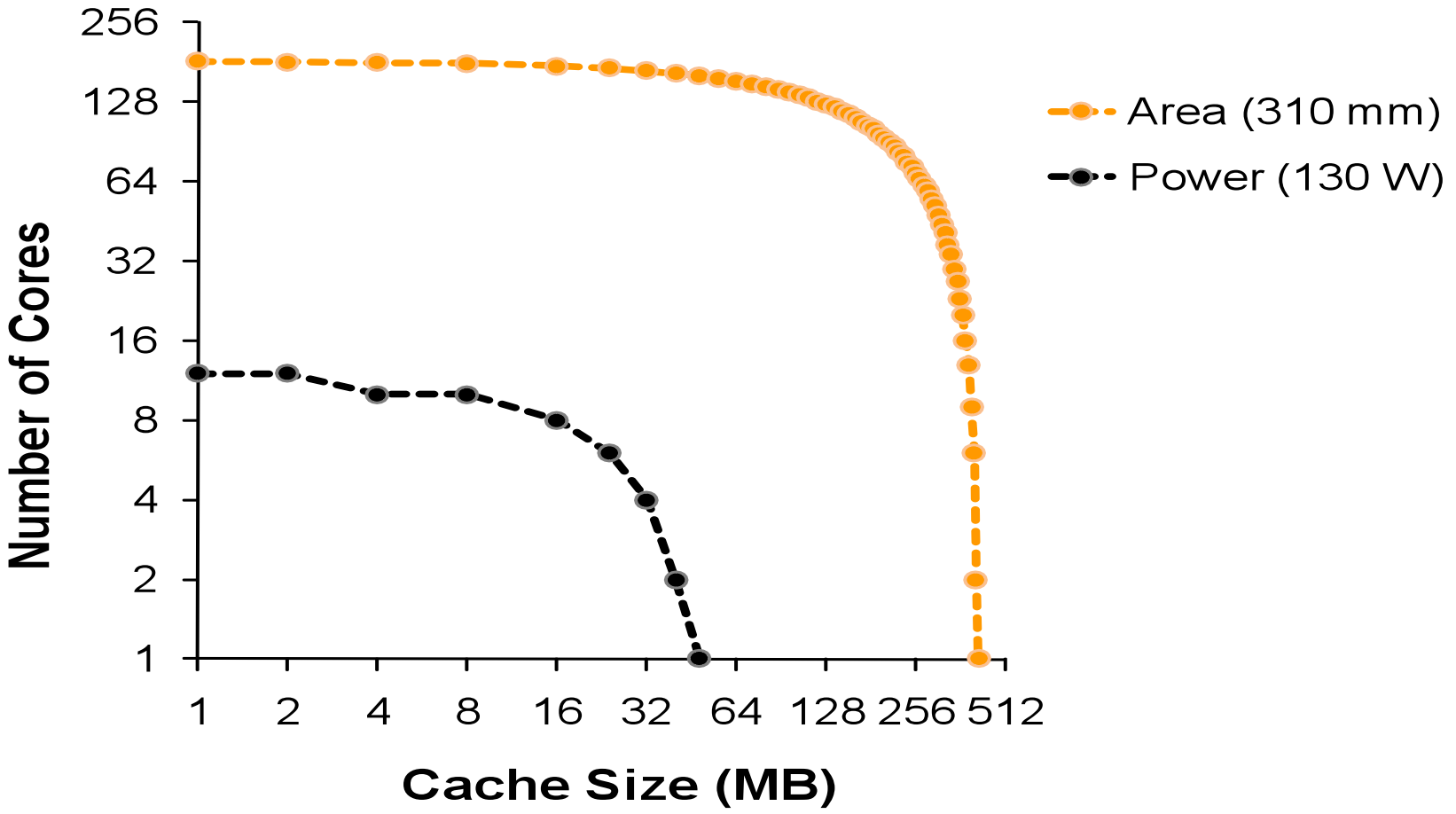
- Scaling server workloads is reasonable

CPI model:

- Works well for workloads with low MLP
- OLTP, Web & DSS are mostly memory-latency dependent

Future servers will run a mix of workloads

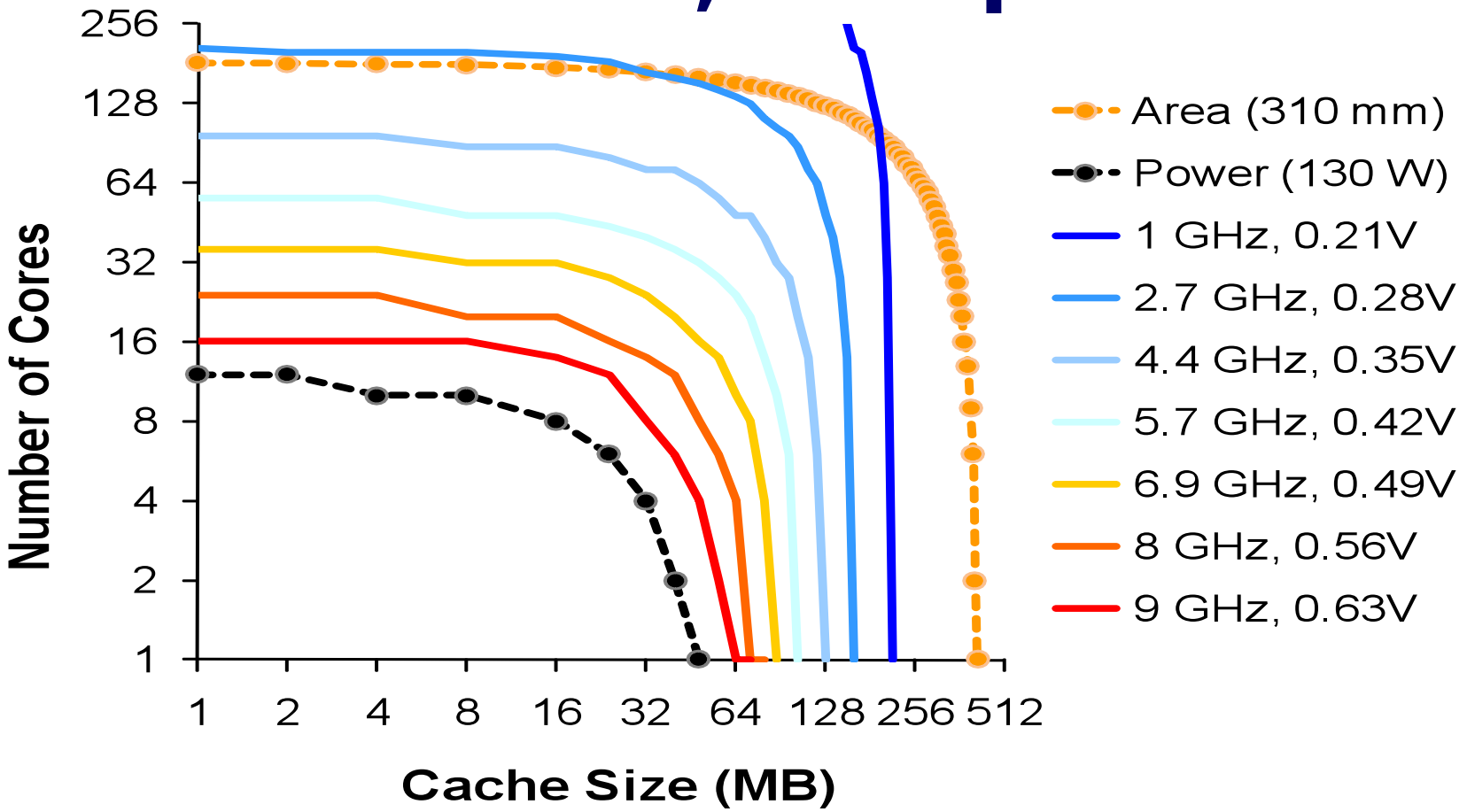
Area vs. Power Envelope (22nm)



✓ Good news: can fit hundreds of cores

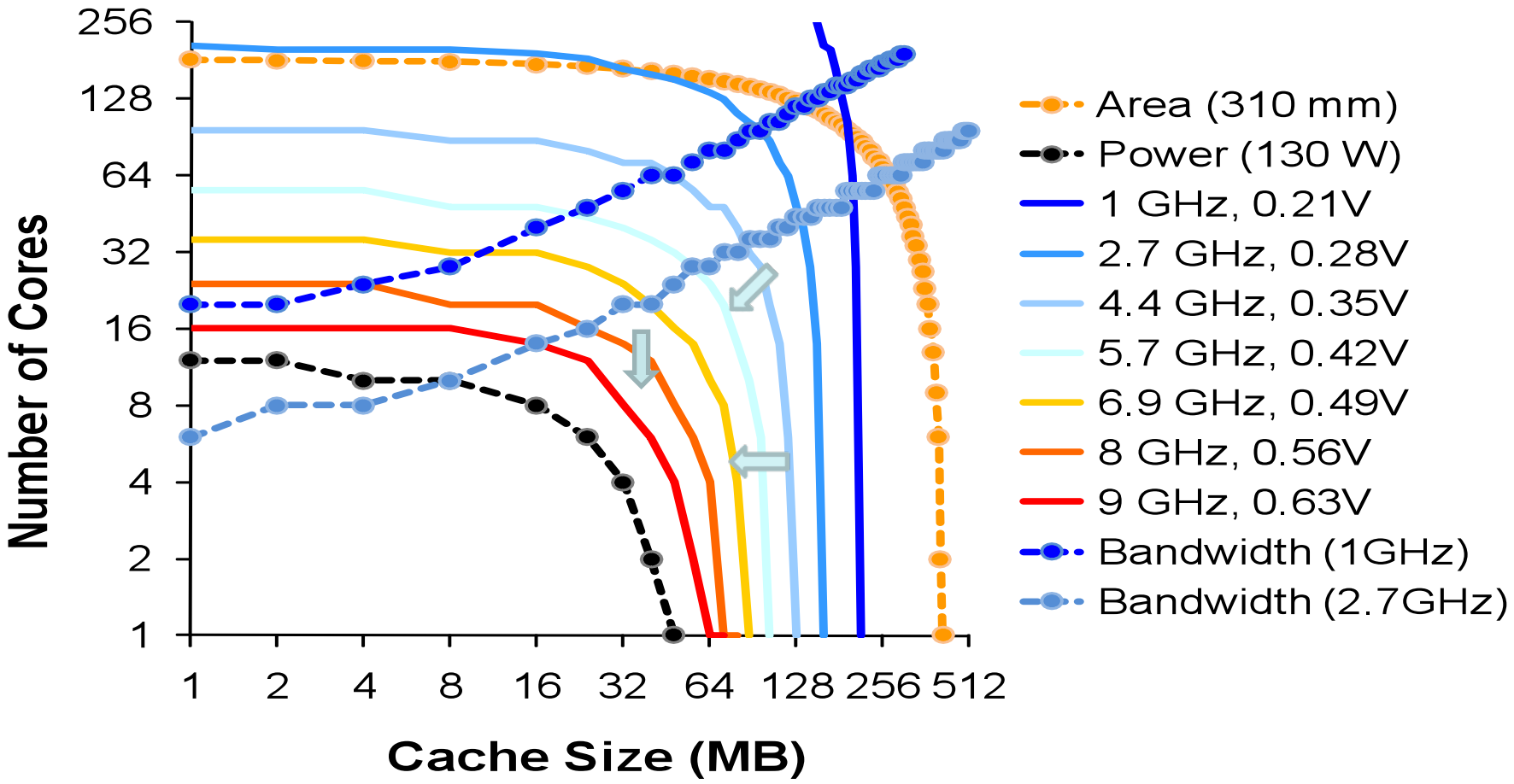
× Can not use them all at highest speed

Of course one could pack more slower cores, cheaper cache



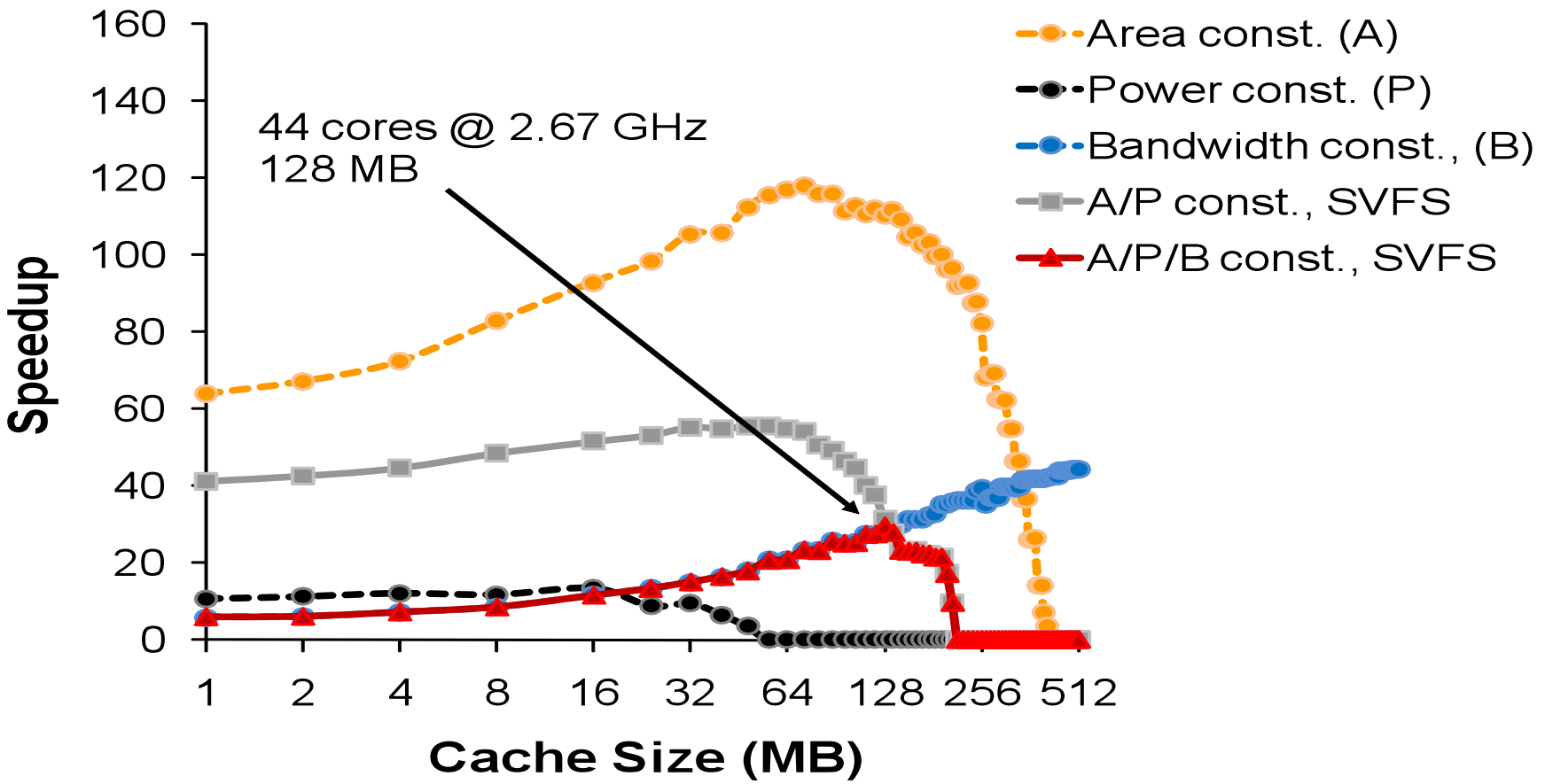
- Result: a performance/power trade-off
- Assuming bandwidth is unlimited

But, limited pin b/w favors fewer cores + more cache



- For clarity, only showing two bandwidth lines
- Where would the best performance be?

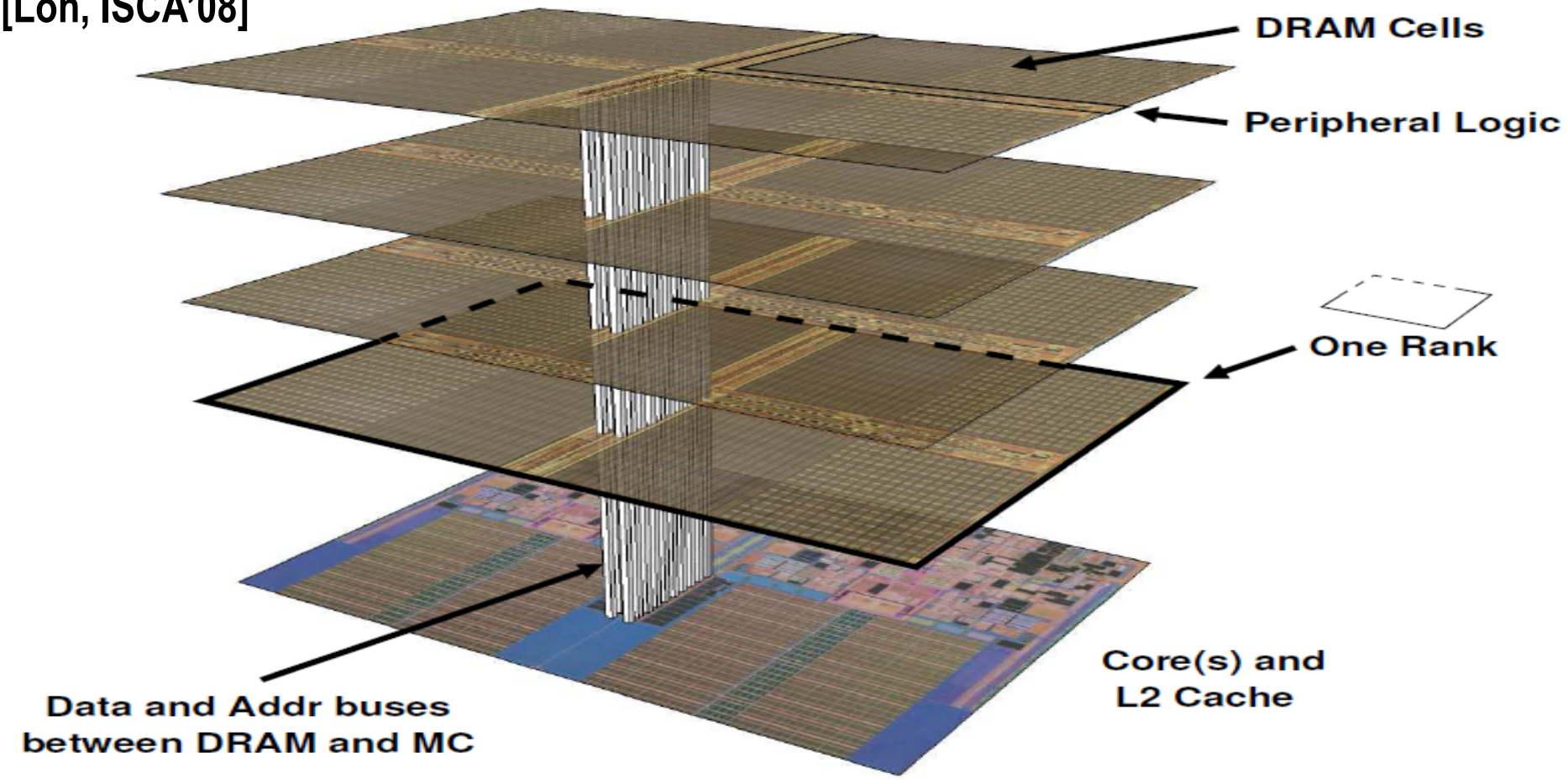
Peak Performing with Conventional Memory



- B/W constrained, then power constrained
- Fewer slower cores, lots of cache

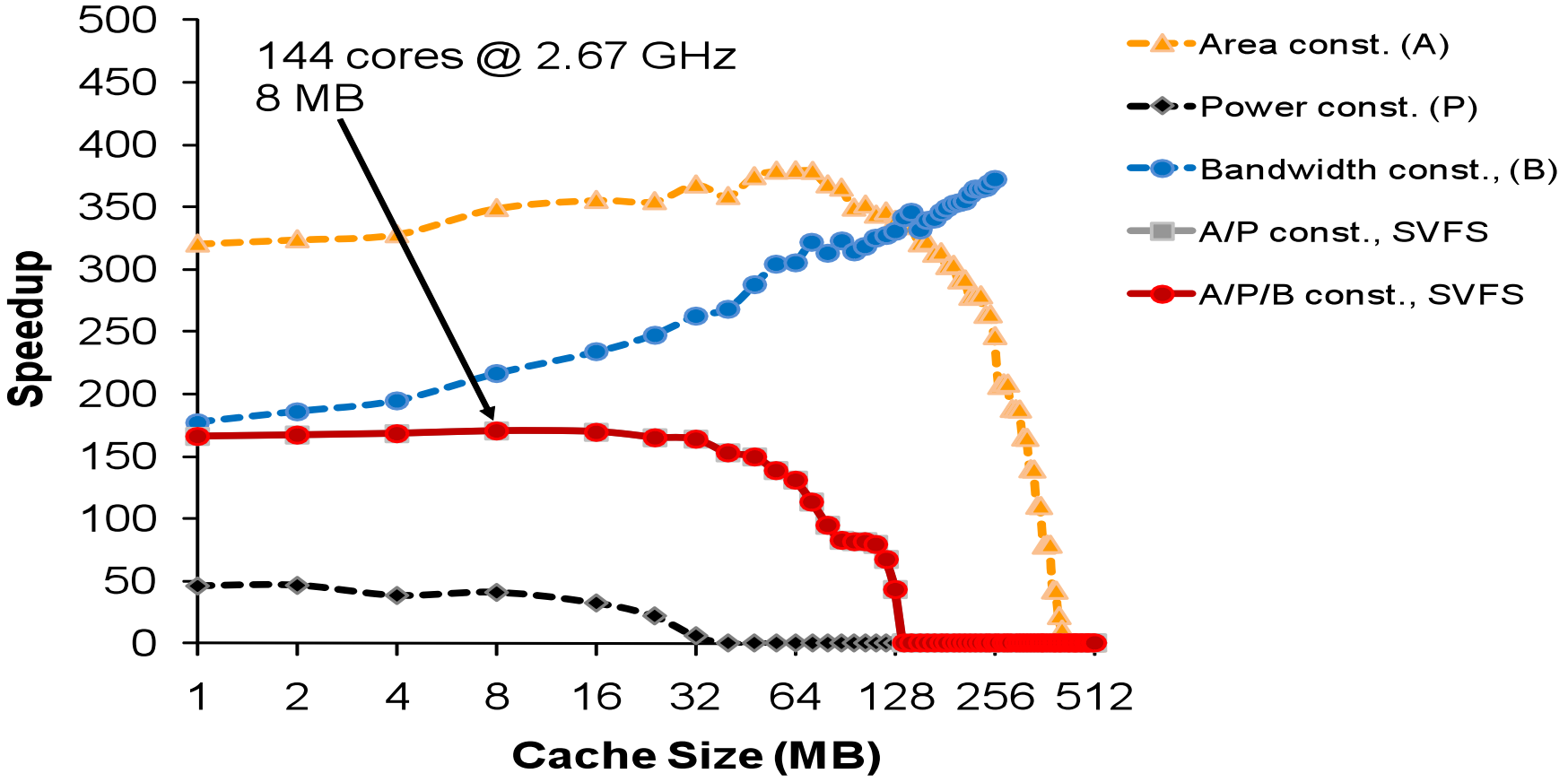
Mitigating B/W Limitations: 3D-stacked Memory

[Loh, ISCA'08]



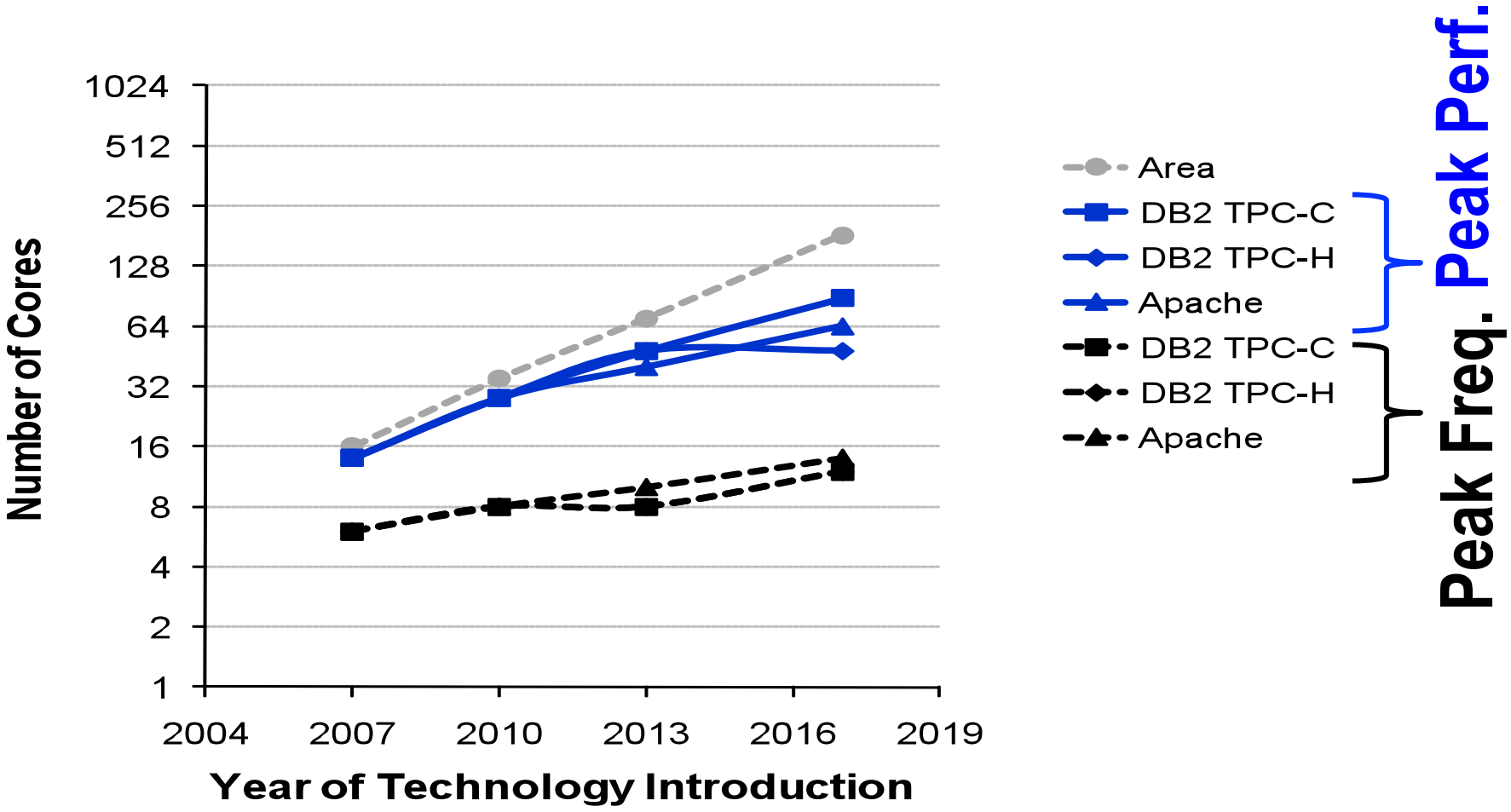
- Delivers TB/sec of bandwidth

Peak Performing w/ 3D-stacked Memory



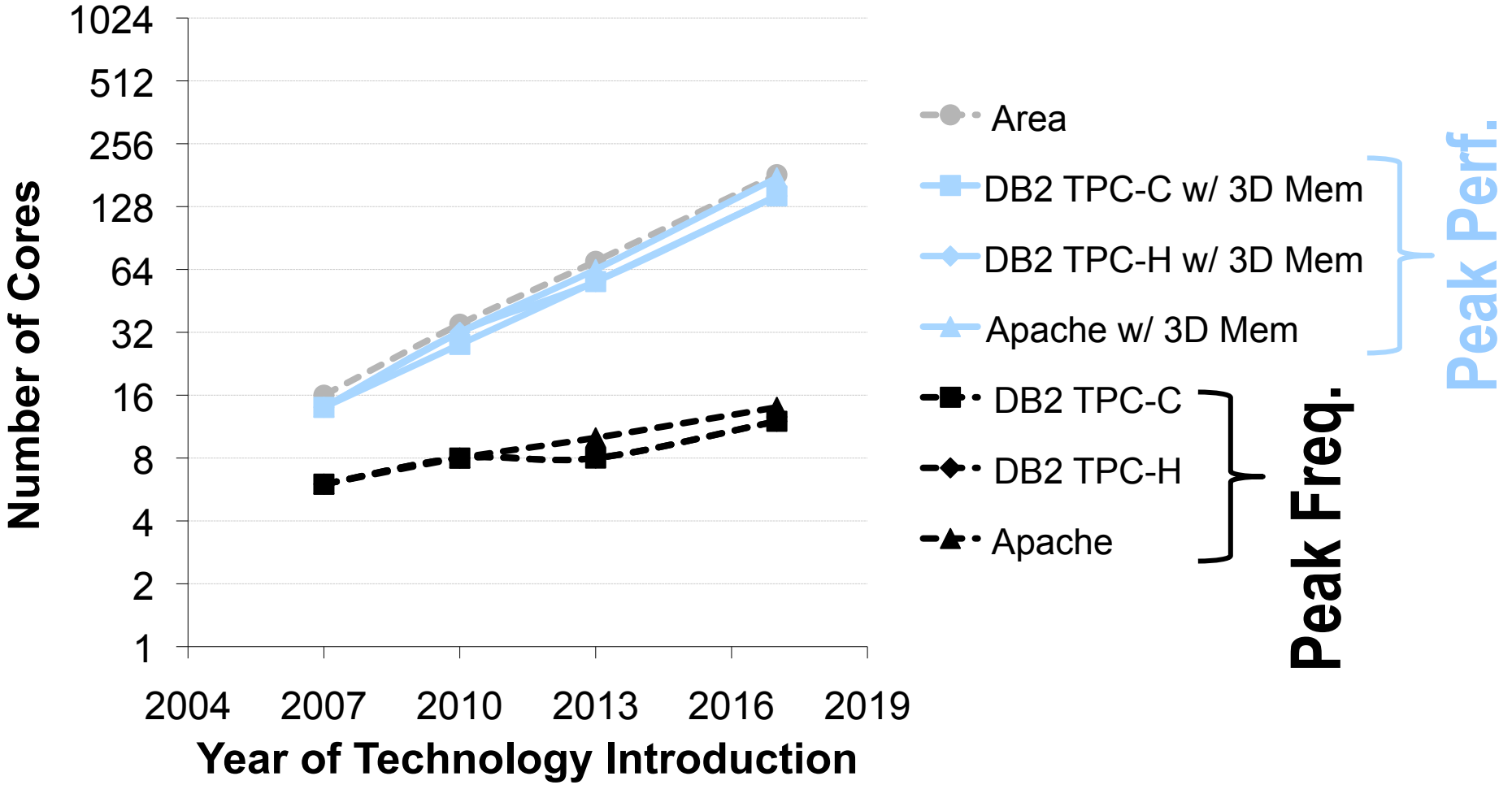
- Only power-constrained
- Virtually eliminates on-chip cache

Core Scaling across Technologies



- Assumes a 130-Watt chip envelope
- Pin b/w keeps Niagara from scaling

Niagara + 3D-stacked Memory



- Power limits Niagara to 75% area!

But, even Niagara is an overkill!

Servers mostly access memory

Benefit little from core complexity

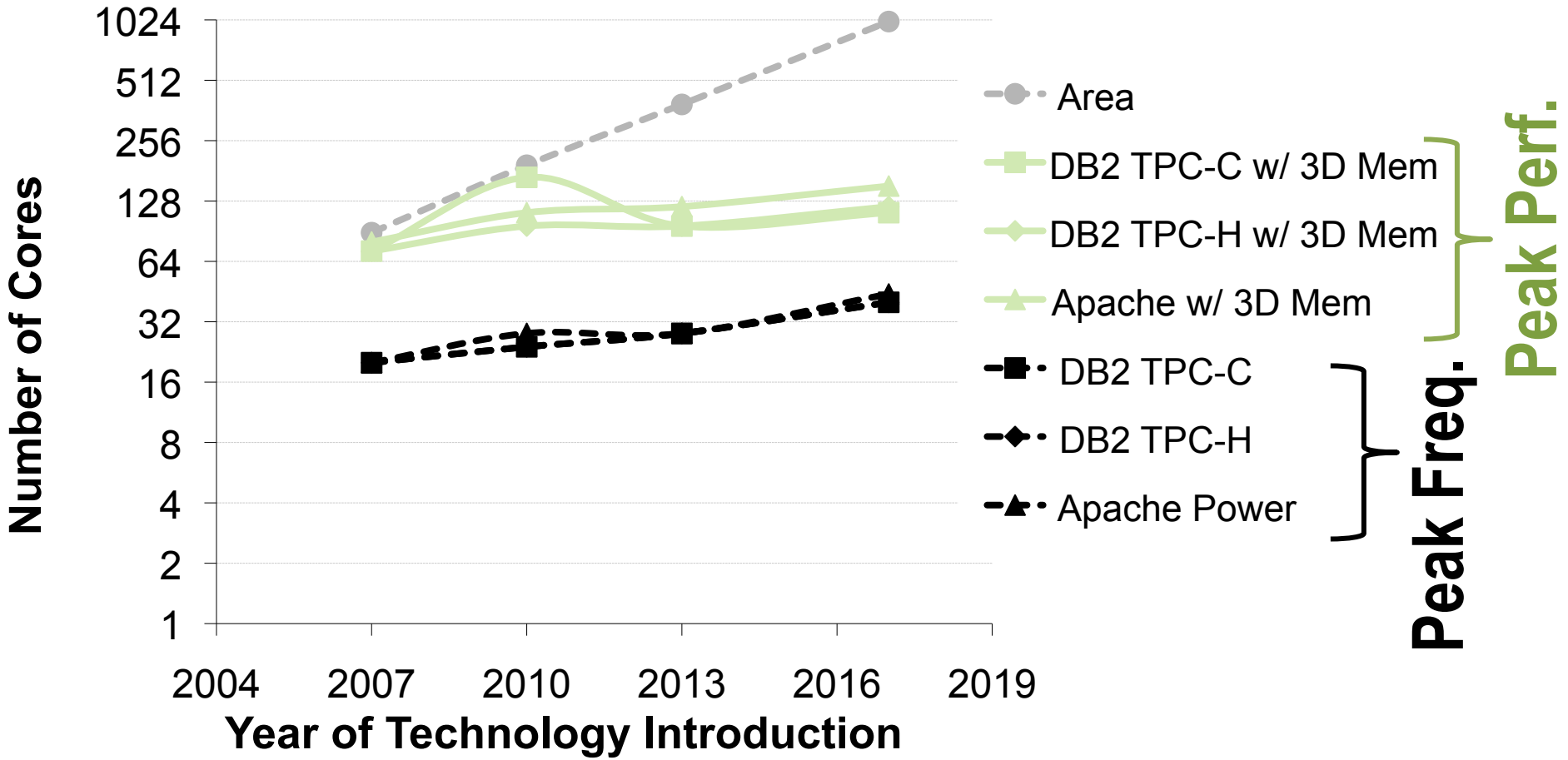
Niagara cores are too big!

E.g., Kgil et al., ASPLOS06:

- Servers on embedded cores + 3D

Can we run servers with embedded cores?

ARM9 + 3D-stacked Memory



- Can not scale with a 130-Watt envelope!!!
- On-chip hierarchy + interconnect not scalable

Long-term: Where to go from here?

1. Redo SW stack
   - Minimize joules/work (algo. down to HW)
   - Program for locality + heterogeneity

2. Pray for technology
   - Energy-scalable silicon devices
   - Emerging nanoscale technologies?

3. Infrastructure technology
   - Renewable/carbon-neutral energy
   - Scalable cooling + power delivery

Short-term Scaling Implications

- Caches are getting huge
  - Need cache architectures to deal with >> MB
  - E.g., Reactive NUCA [ISCA'09]
- Interconnect + cache hierarchy power
  - Need lean on-chip communication/storage
  - Eurocloud chip: ARM+3D [ACLD'10]
- Dark Silicon
  - Specialized processors
  - Use only parts of the chip at a time

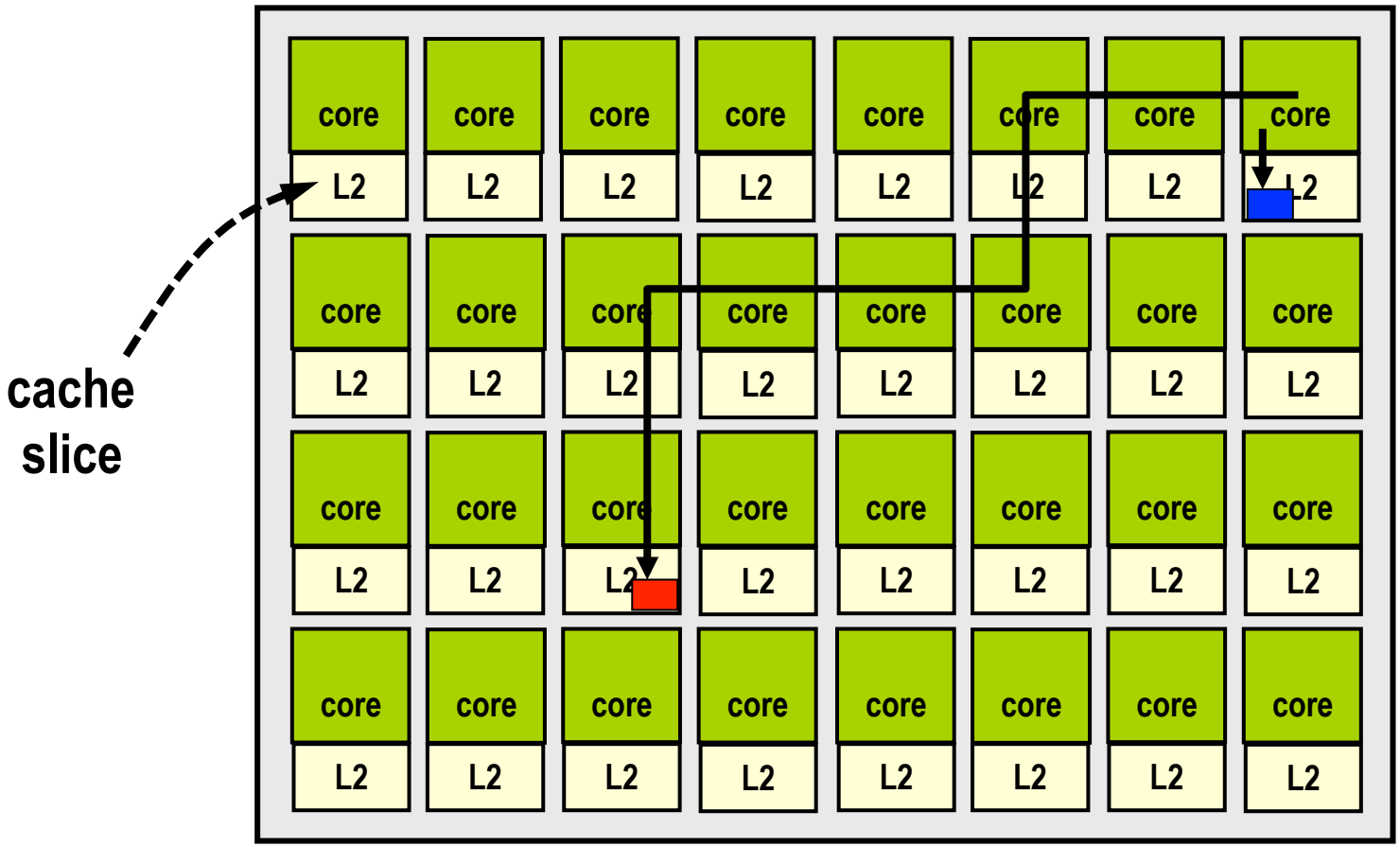
Outline

- Where are we?
- Energy scalability for servers
- Where do we go from here

→Future on-chip caches

- Future NoC's
- Summary

Optimal Data Placement in Large On-chip Caches



- Data placement determines performance
- Goal: place data on chip close to where they are used

Prior Work

- Several proposals for CMP cache management
  - ASR, cooperative caching, victim replication, CMP-NuRapid, D-NUCA

- ...but suffer from shortcomings
  - complex, high-latency lookup/coherence
  - don't scale
  - lower effective cache capacity
  - optimize only for subset of accesses

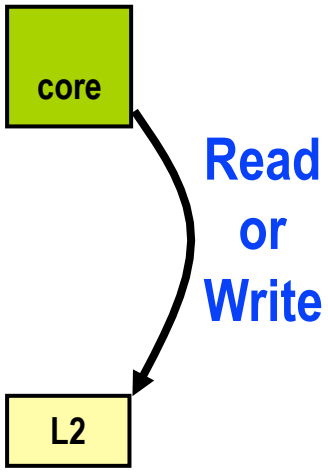
We need:

➡ Simple, scalable mechanism for fast access to all data

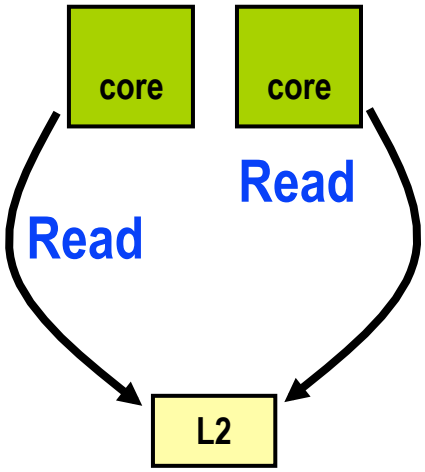
Our Proposal: Reactive NUCA
[ISCA'09, IEEE Micro Top Picks '10]

- Cache accesses can be classified at run-time
  - Each class amenable to different placement
- Per-class block placement
  - Simple, scalable, transparent
  - No need for HW coherence mechanisms at LLC
- Speedup
  - Up to 32% speedup
  - -5% on avg. from ideal cache organization

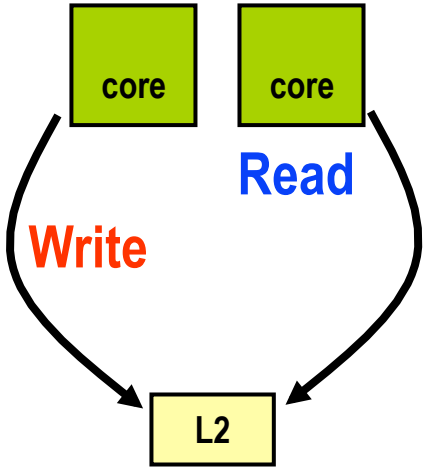
Terminology: Data Types



Private



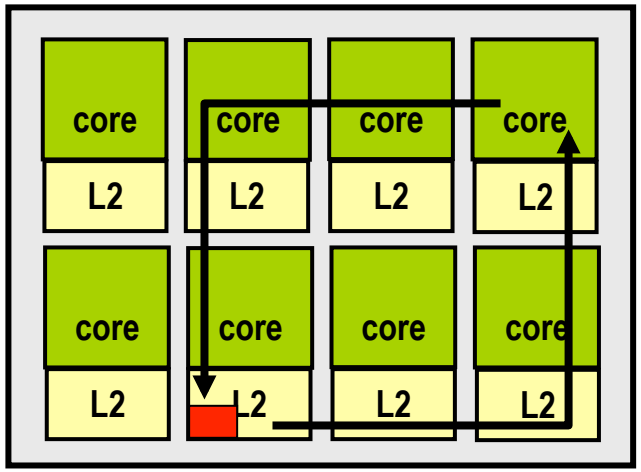
Shared
Read-Only



Shared
Read-Write

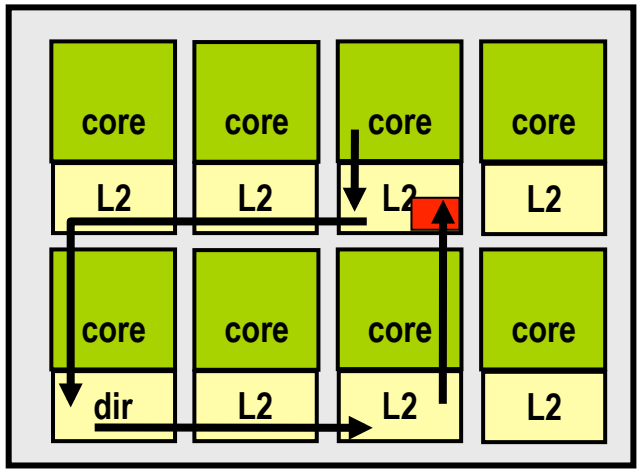
Conventional Multicore Caches

Shared



Addr-interleave blocks
+ High effective capacity
– Slow access

Private

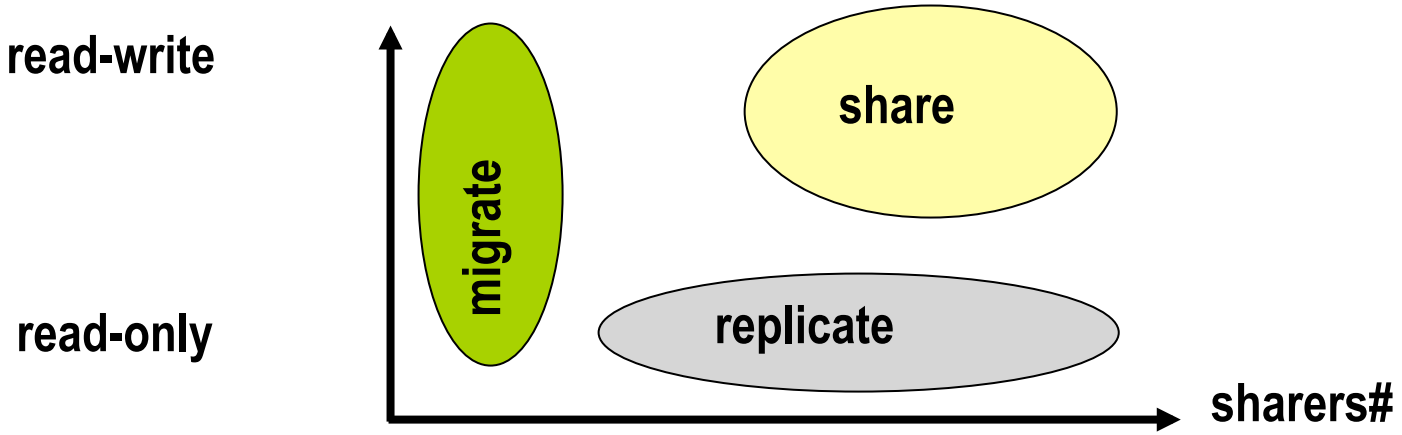


Each block cached locally
+ Fast access (local)
– Low capacity (replicas)
– Coherence: via indirection (distributed directory)

➡ We want: high capacity (shared) + fast access (priv.)

Where to Place the Data?

- Close to where they are used!
- Accessed by single core: migrate locally
- Accessed by many cores: replicate (?)
  - If read-only, replication is OK
  - If read-write, coherence a problem
    - Low reuse: evenly distribute across sharers



Methodology

Flexus: Full-system cycle-accurate timing simulation

Workloads

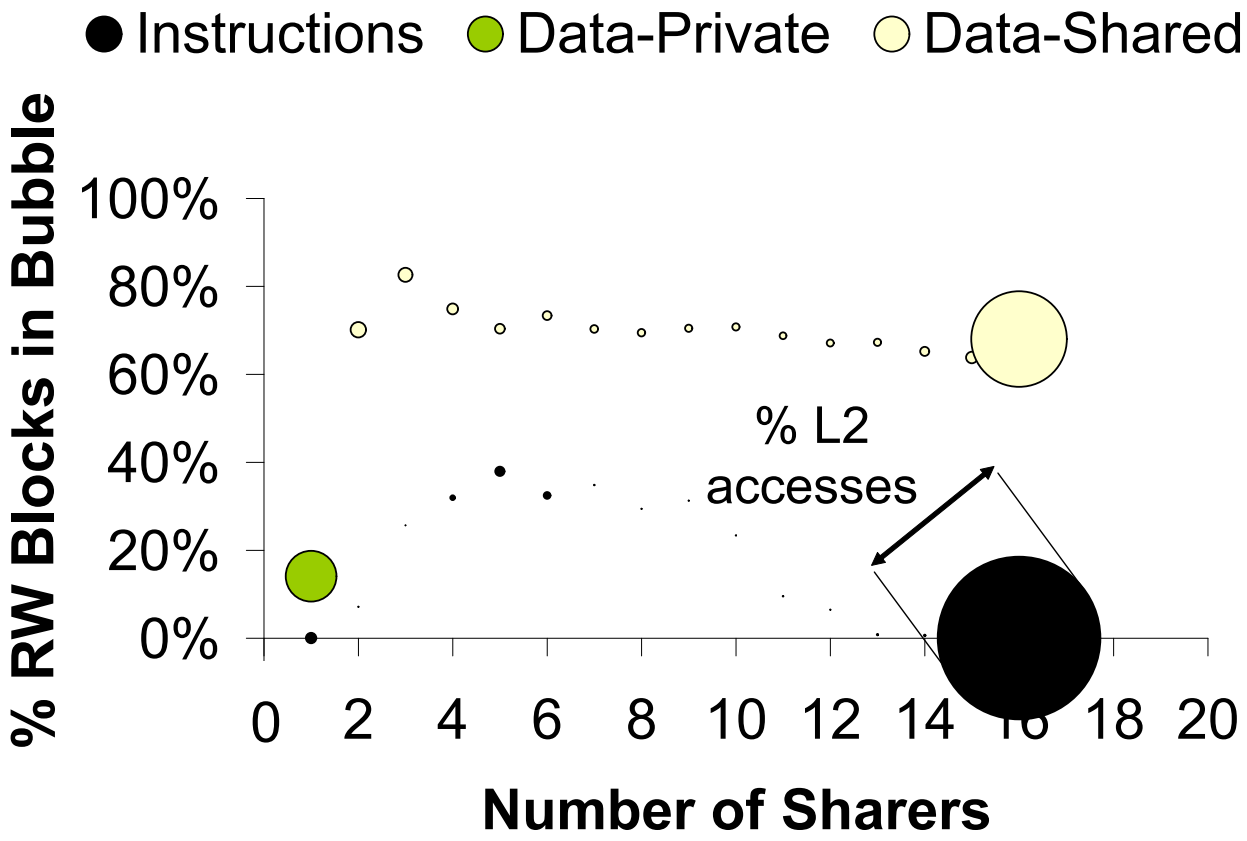
- OLTP: TPC-C 3.0 100 WH
  - IBM DB2 v8
  - Oracle 10g
- DSS: TPC-H Qry 6, 8, 13
  - IBM DB2 v8
- SPECweb99 on Apache 2.0
- Multiprogammed: SPEC2K
- Scientific: em3d

Model Parameters

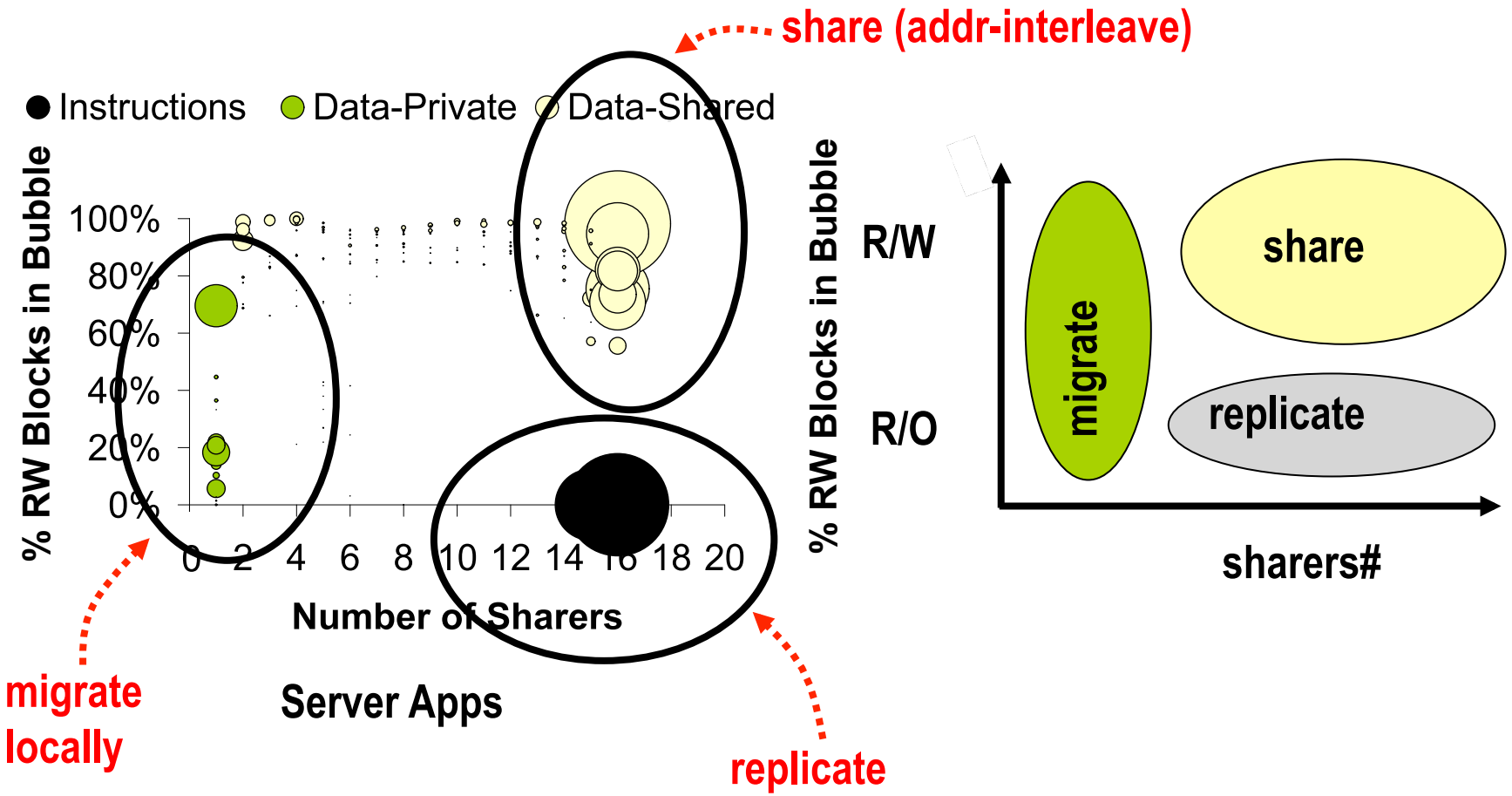
- Tiled, LLC = L2
- 16-cores, 1MB/core
- OoO, 2GHz, 96-entry ROB
- Folded 2D-torus
  - 2-cycle router
  - 1-cycle link
- 45ns memory

Cache Access Classification

- Each bubble: cache blocks shared by x cores
- Size of bubble proportional to % L2 accesses
- y axis: % blocks in bubble that are read-write



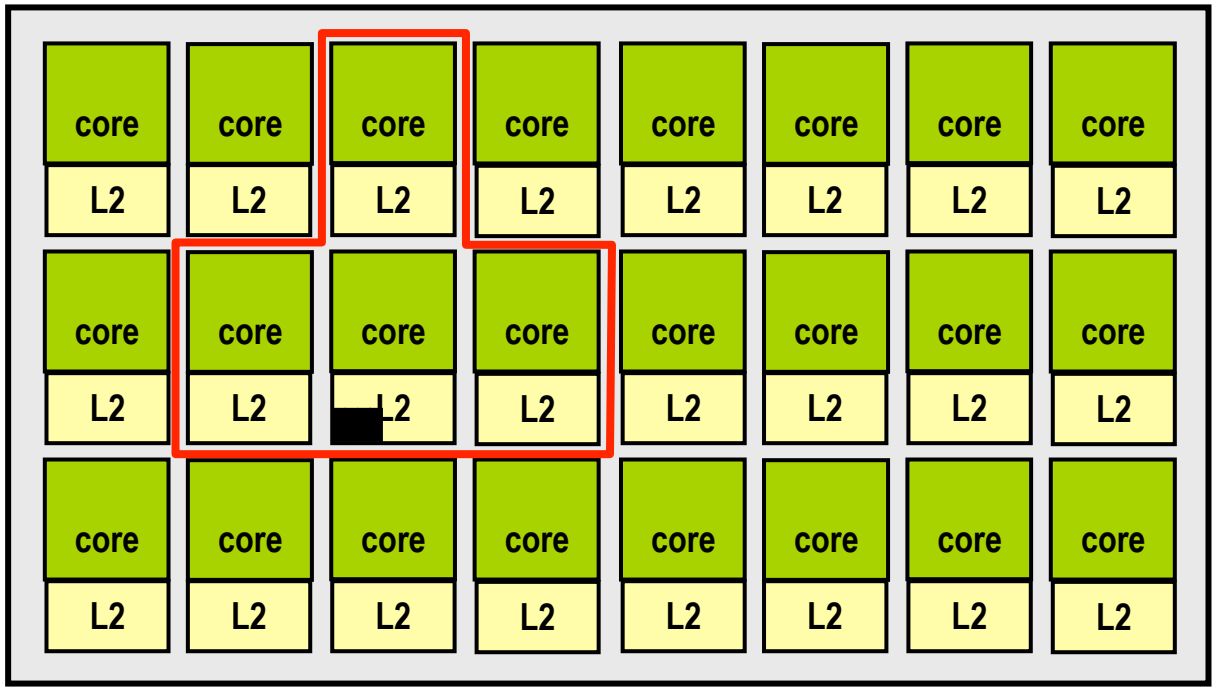
Cache Access Clustering



- Accesses naturally form 3 clusters

Instruction Replication

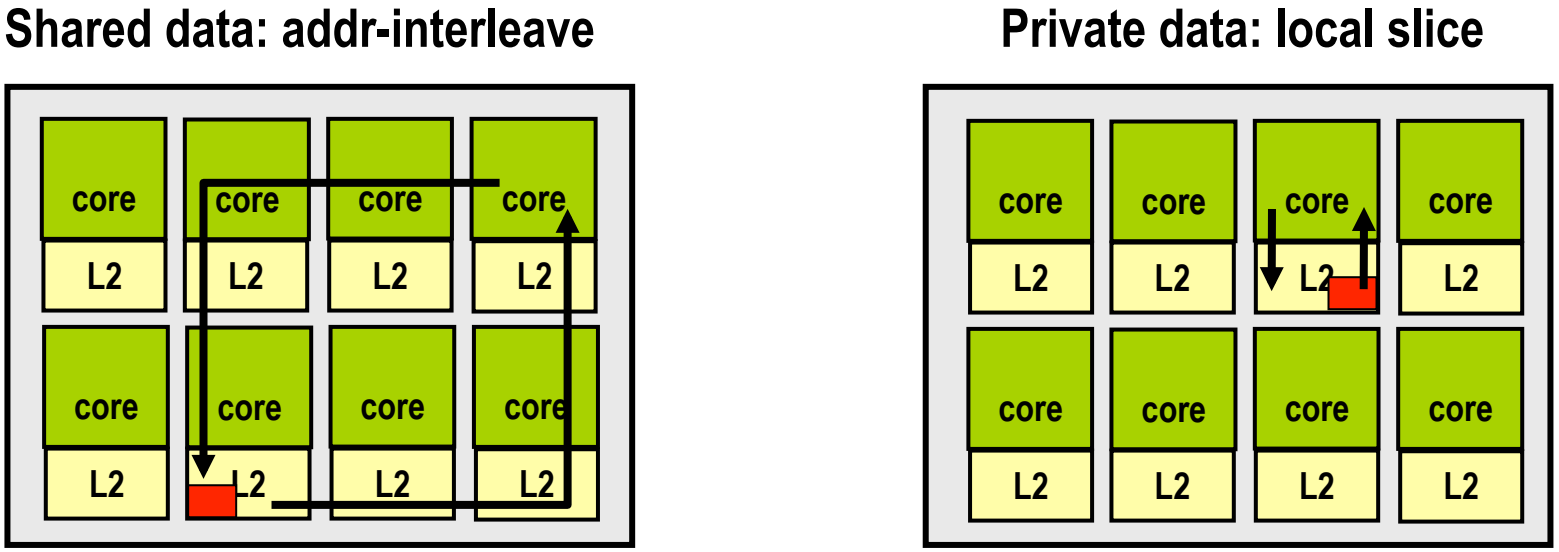
- Instruction working set too large for one cache slice



➧ Distribute in cluster of neighbors, replicate across

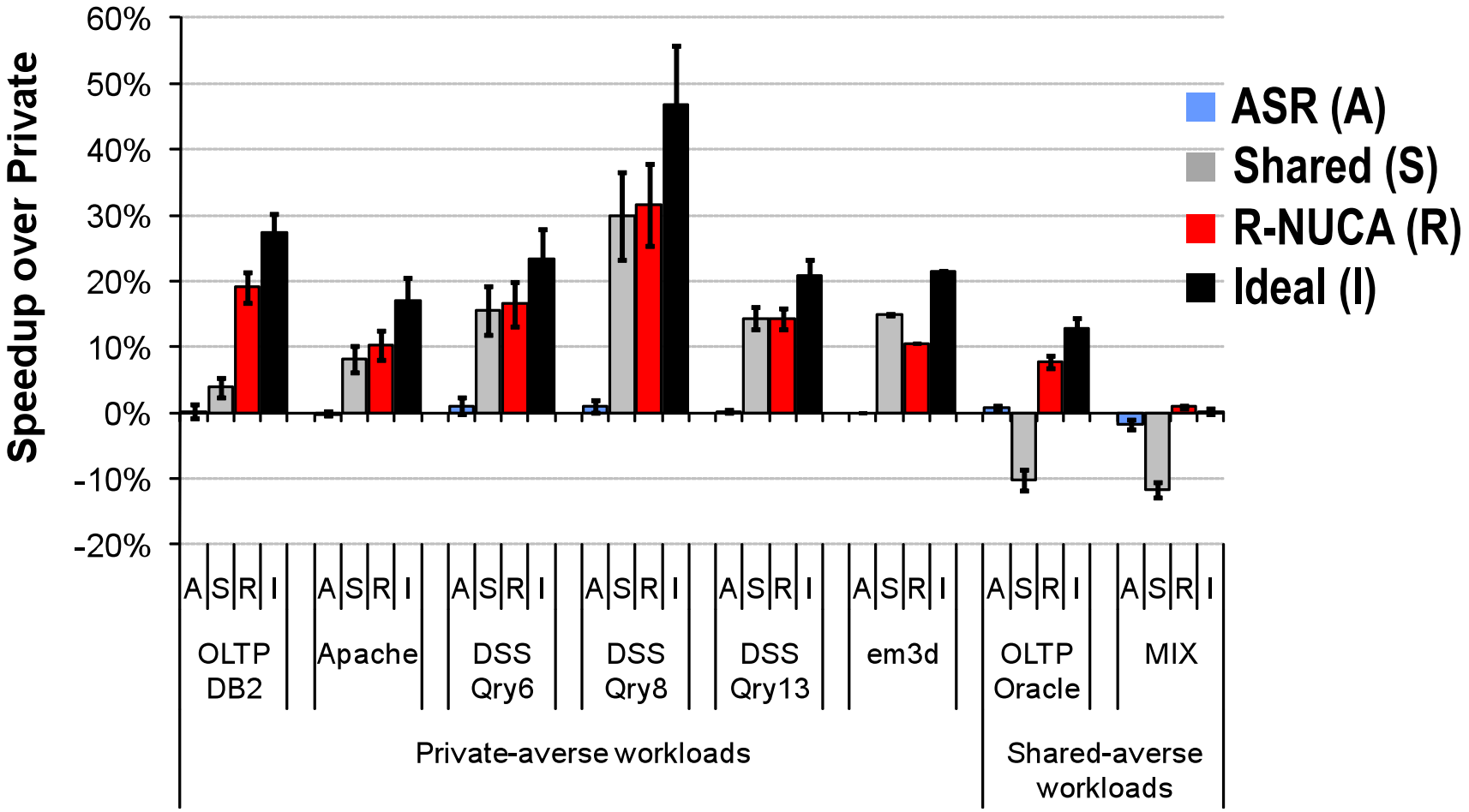
Coherence: No Need for HW Mechanisms at LLC

- Reactive NUCA placement guarantee
  - Each R/W datum in unique & known location



➡ Fast access, eliminates HW overhead

Evaluation



- Delivers robust performance across workloads
  - Shared: same for Web, DSS; 17% for OLTP, MIX
  - Private: 17% for OLTP, Web, DSS; same for MIX

R-NUCA Conclusions

Near-optimal block placement and replication in distributed caches

- Cache accesses can be classified at run-time
  - Each class amenable to different placement
- Reactive NUCA: placement of each class
  - Simple, scalable, low-overhead, transparent
  - Obviates HW coherence mechanisms for LLC
- Robust performance across server workloads
  - Near-optimal placement (-5% avg. from ideal)

Outline

- Overview
- Where are we?
- Energy scalability for servers
- Where do we from here?
- Future on-chip caches
- ➔Future NoC's
- Summary

Optimal Interconnect

On-chip interconnect is an energy hog!

- Over 30% of chip energy (e.g., SCC, RAW)

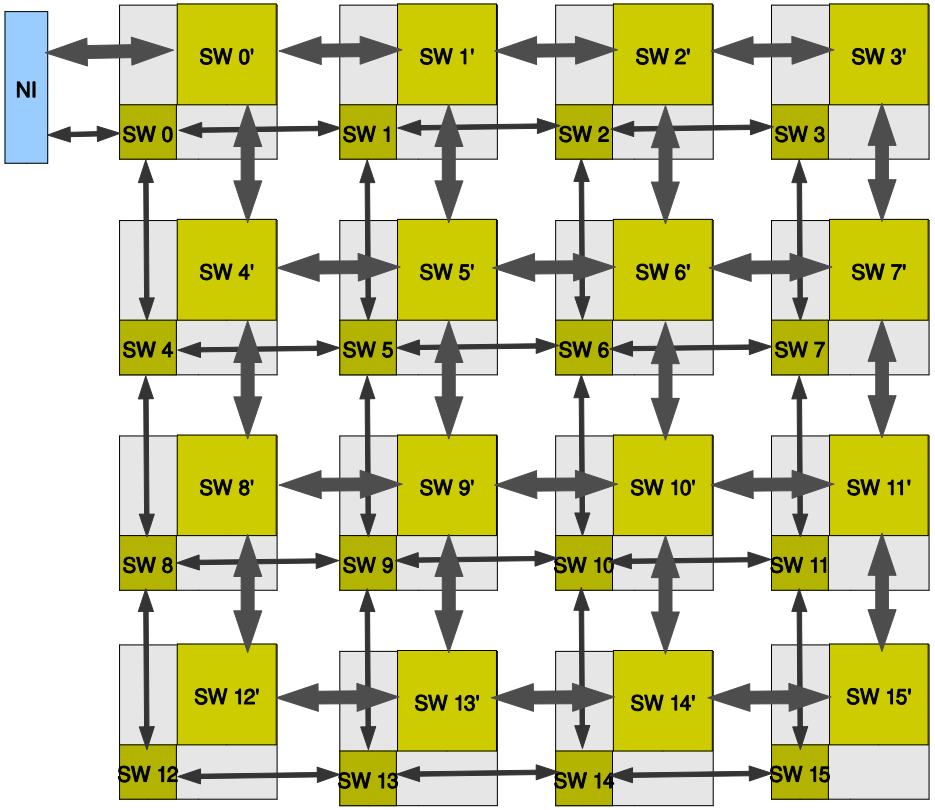
Modern NoC's:

- Optimize for worst-case traffic
- Virtualize req/rep to avoid protocol deadlock

But,

- Cache-coherent chips have bimodal traffic
  - Requests are control, replies are blocks
  - Typically just load cache blocks

CCNoC: Optimal for Bimodal Traffic



- Narrow request plane
- Full-width response plane
- No need for virtual channels

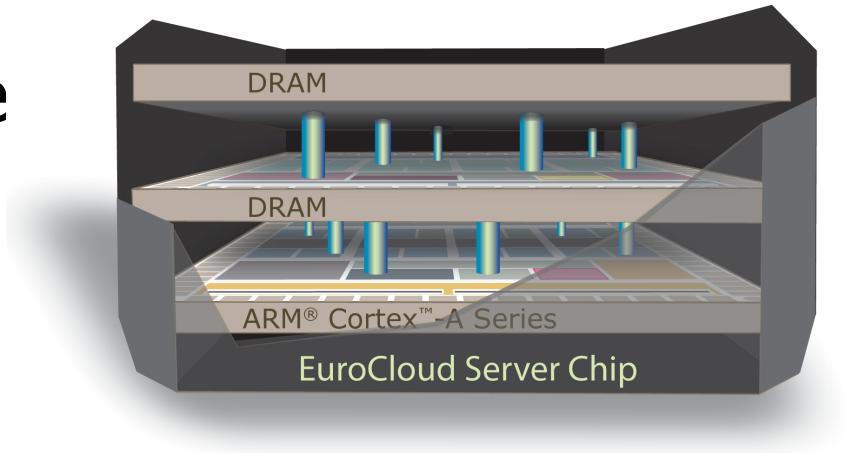
→ **30%-40% power improvement**

Bringing it all together: The EuroCloud Chip

(www.eurocloudserver.com)

Datacenters with mobile processors

- ARM cores
  - Will likely have to be multithreaded!
- 3D-stacked memory
- Nokia's Ovi Cloud applications



Your 1-Watt Future Datacenter Chip
[ACLD'10]

Design for Dark Silicon
Long-term: Vertically Integrate

Can not power up entire chip?

➠ Specialize!

Vertically-integrated server architecture (VISA)

- Identify services which are energy hogs
- Integrate SW/HW to minimize energy/service
- Provide service API not ISA
- E.g., Intel's TCP/IP processor @ 1W

Good places to start:

- OS, DBMS, machine learning

Summary

- Moore's law continues (for another decade)
- CMOS is still cheap
- But, energy scaling has slowed down

Recommendation: Energy-Centric Computing

- Can't get there with parallelism alone
- Holistic approach to energy

Time to put the "embedded" into all of computing!

For more information

please visit us online at parsa.epfl.ch, ecocloud.ch