Energy Efficient System Architectures

Principles and Examples

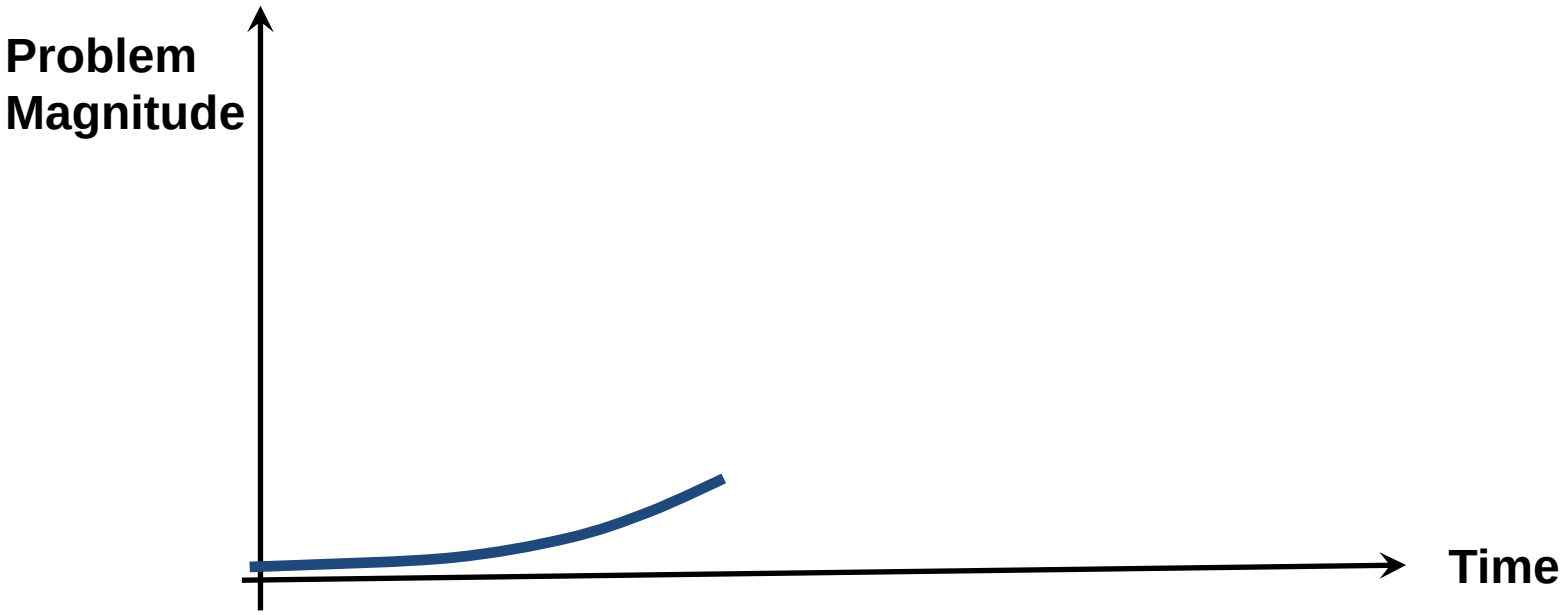
Avinoam Kolodny

Electrical Engineering Department
Technion – Israel Institute of Technology

Green Photonics – TU Berlin 2015

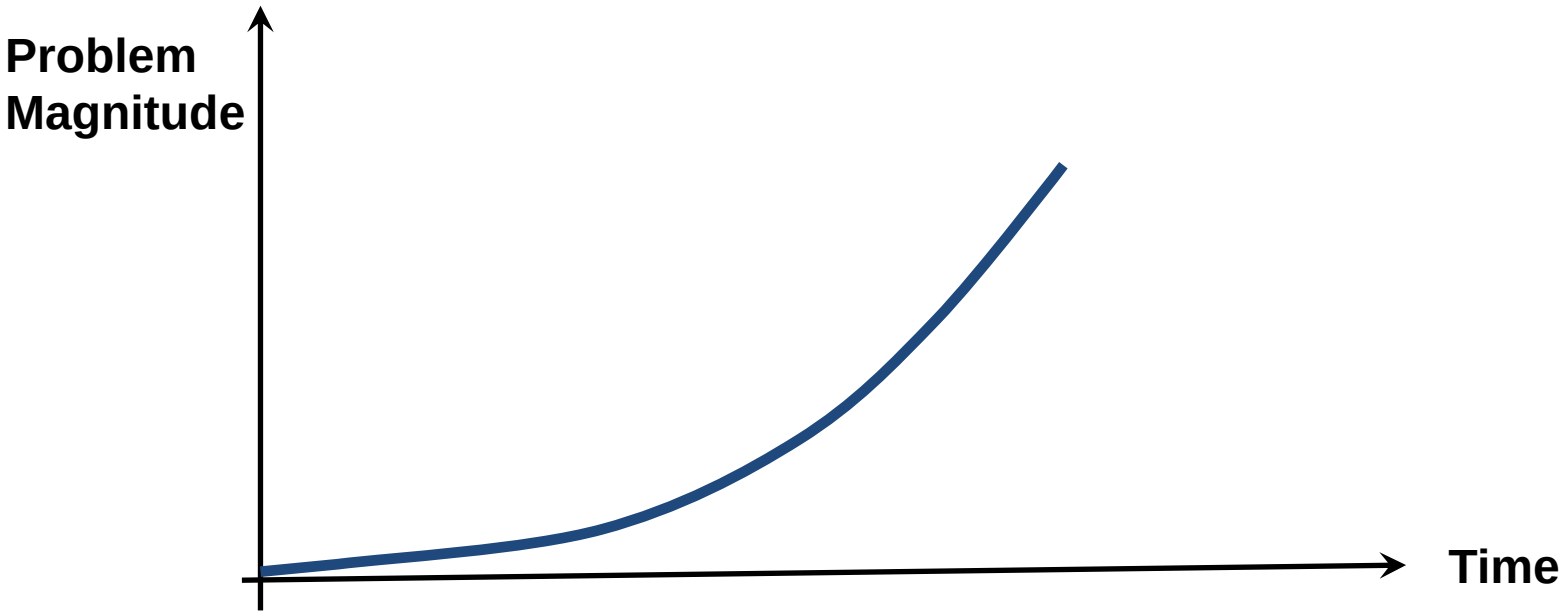
The Disruption Principle

- A problem grows quietly…..



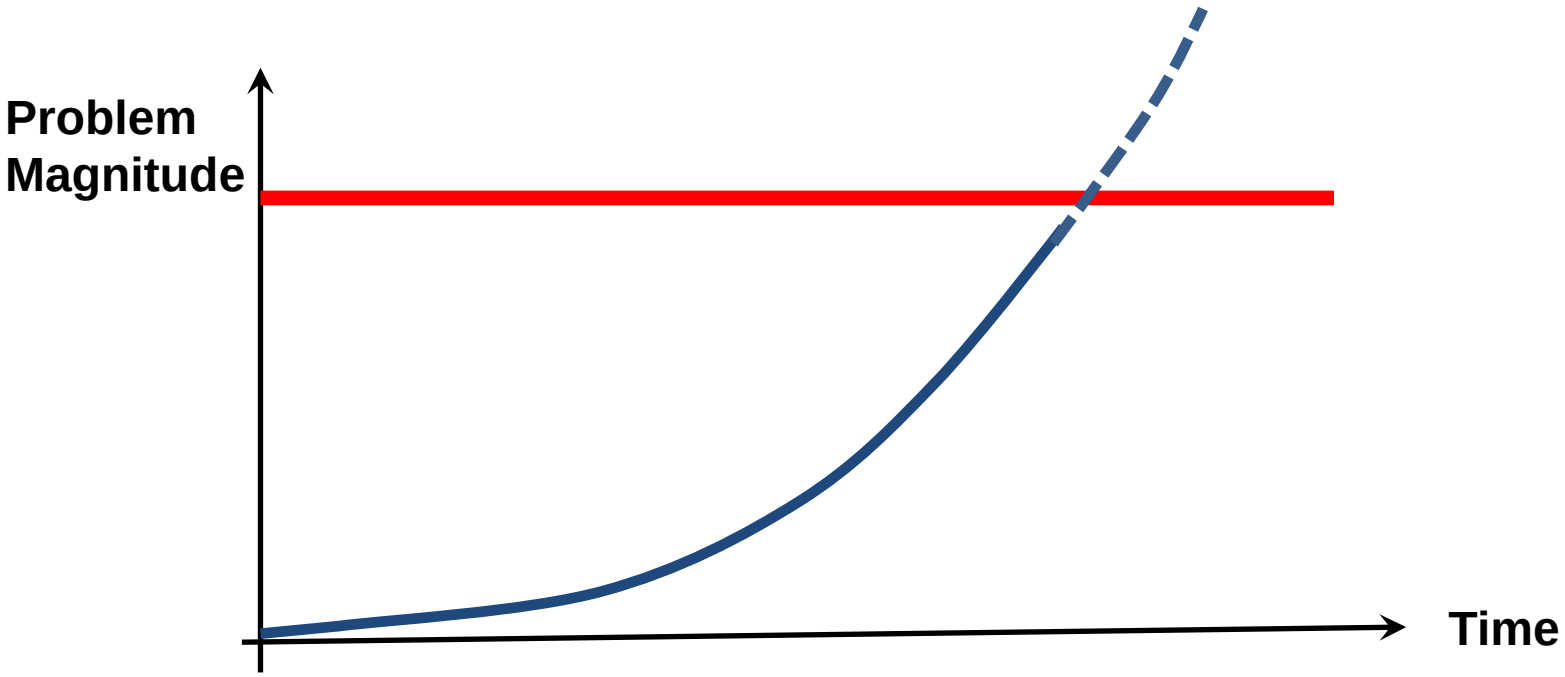
The Disruption Principle

- A problem grows quietly…..



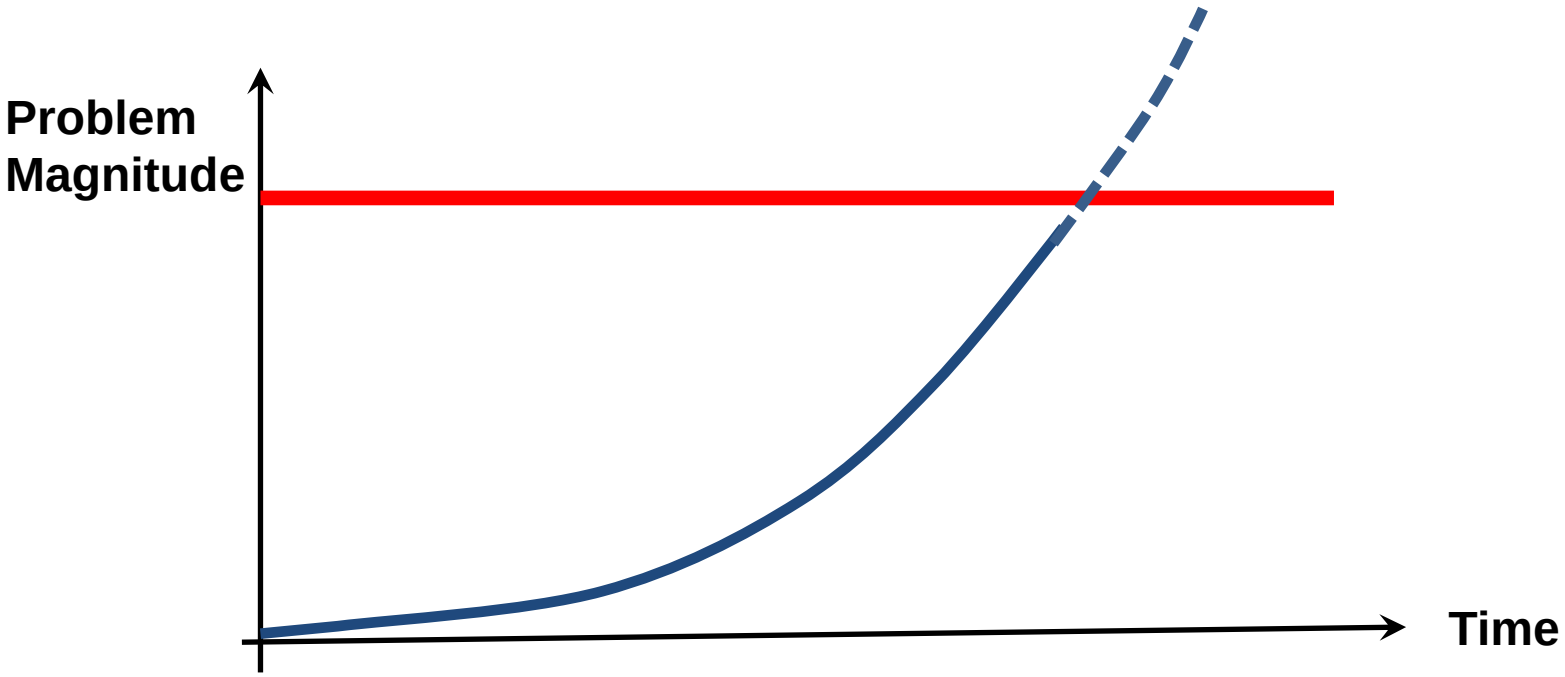
The Disruption Principle

- A problem grows quietly…..
- …. Until it becomes **critical**.



The Disruption Principle

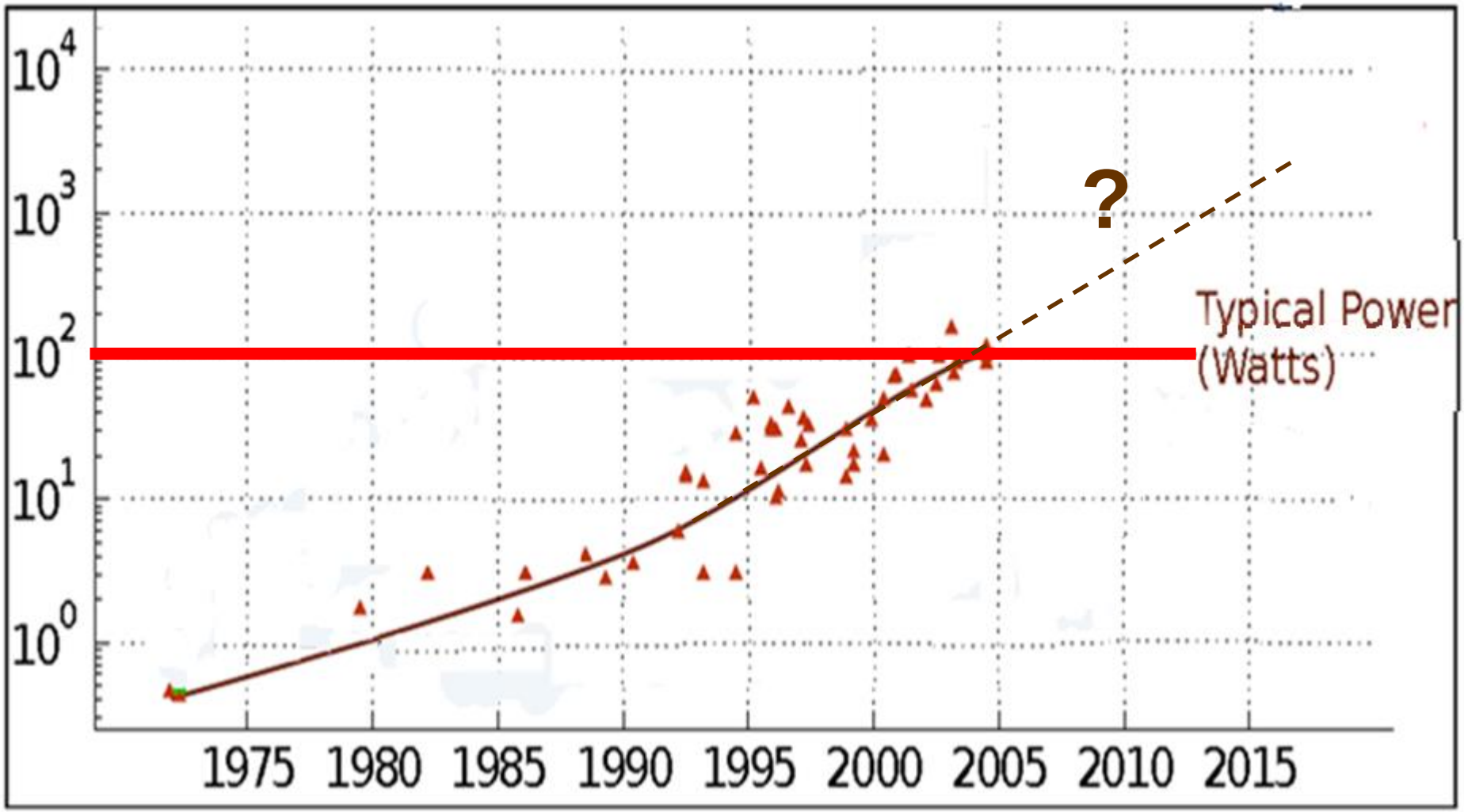
- A problem grows quietly…..
- …. Until it becomes **critical**.
- **At this point something must be done!**



Example:
Power Dissipation in Processor Chips

Power Dissipation in Processor Chips

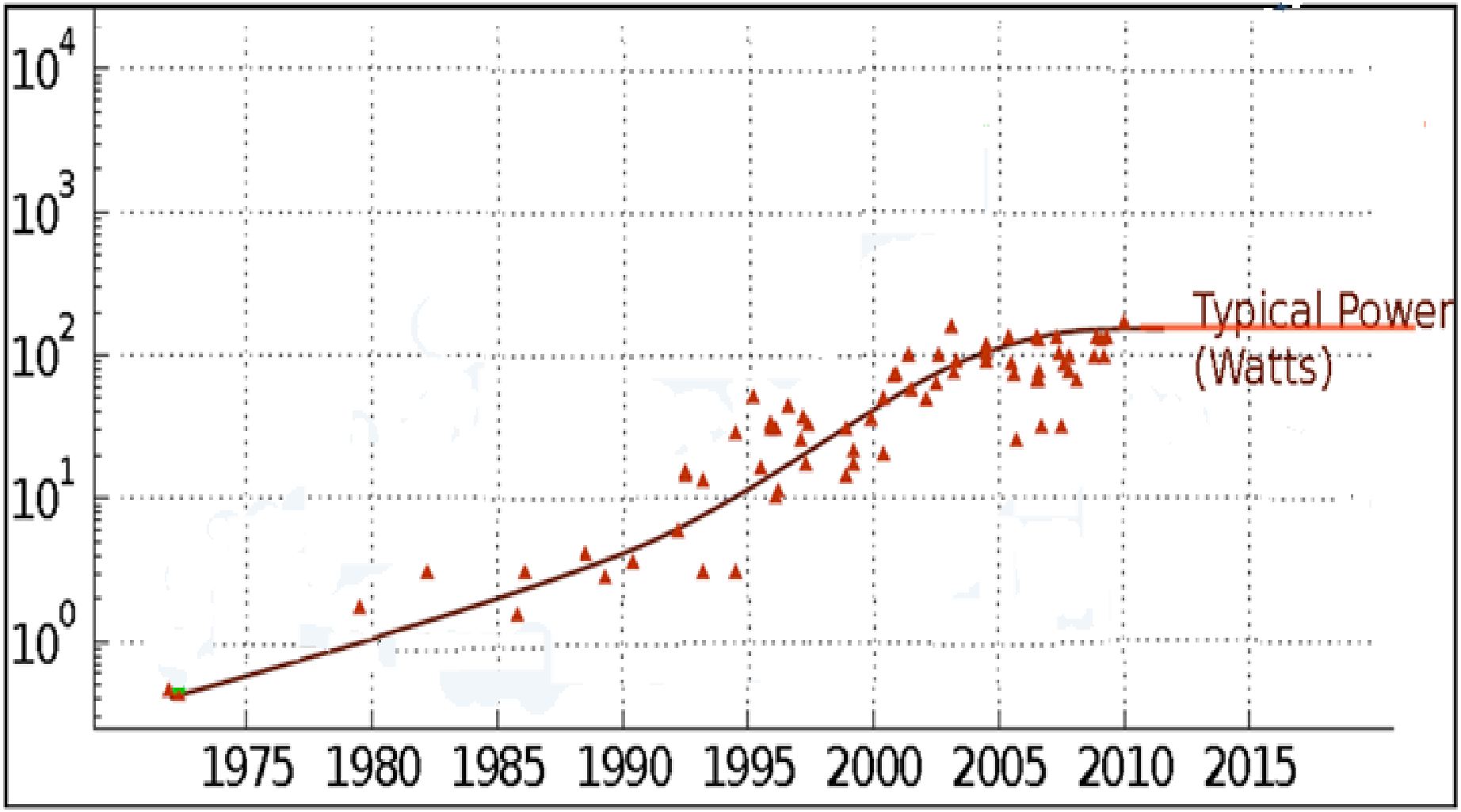
Until 2005



http://cpudb.stanford.edu/

Power Dissipation in Processor Chips

After 2005 – flat!



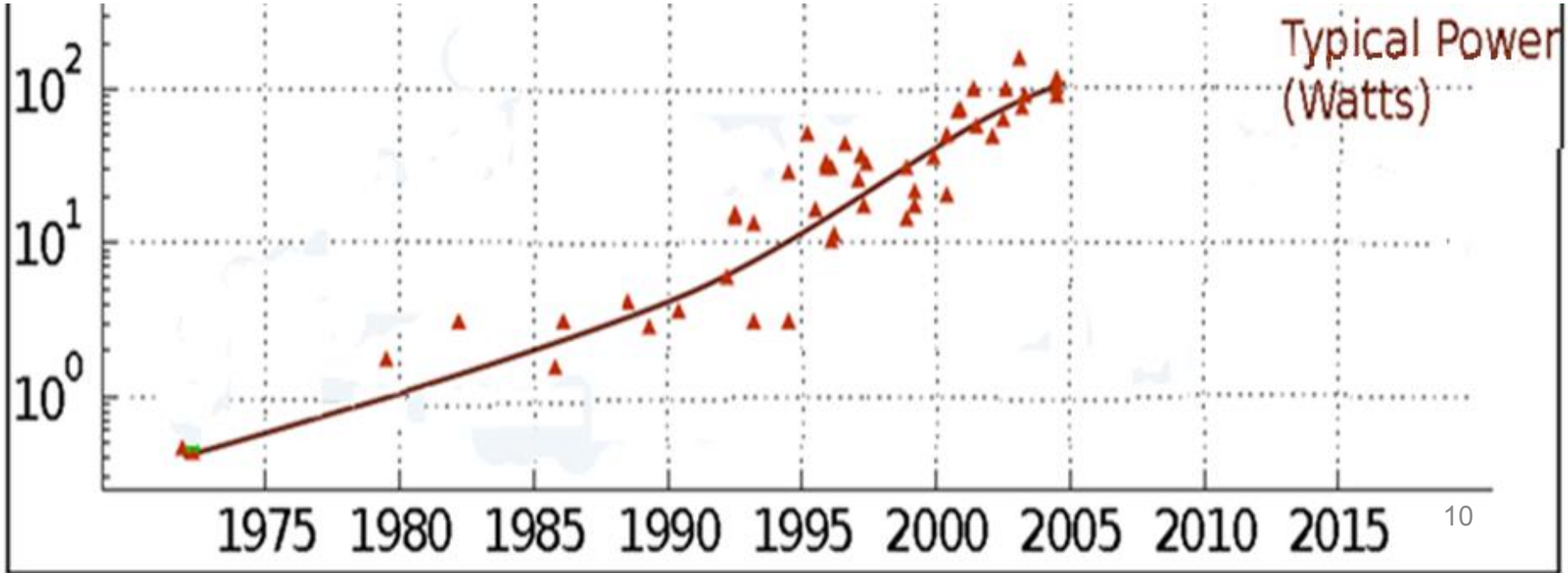
http://cpudb.stanford.edu/

What Caused the Disruption?

- **Power limit (thermal)**
  Too much heat for air-cooled computers

- **Energy limit**
  Battery life issue in mobile devices

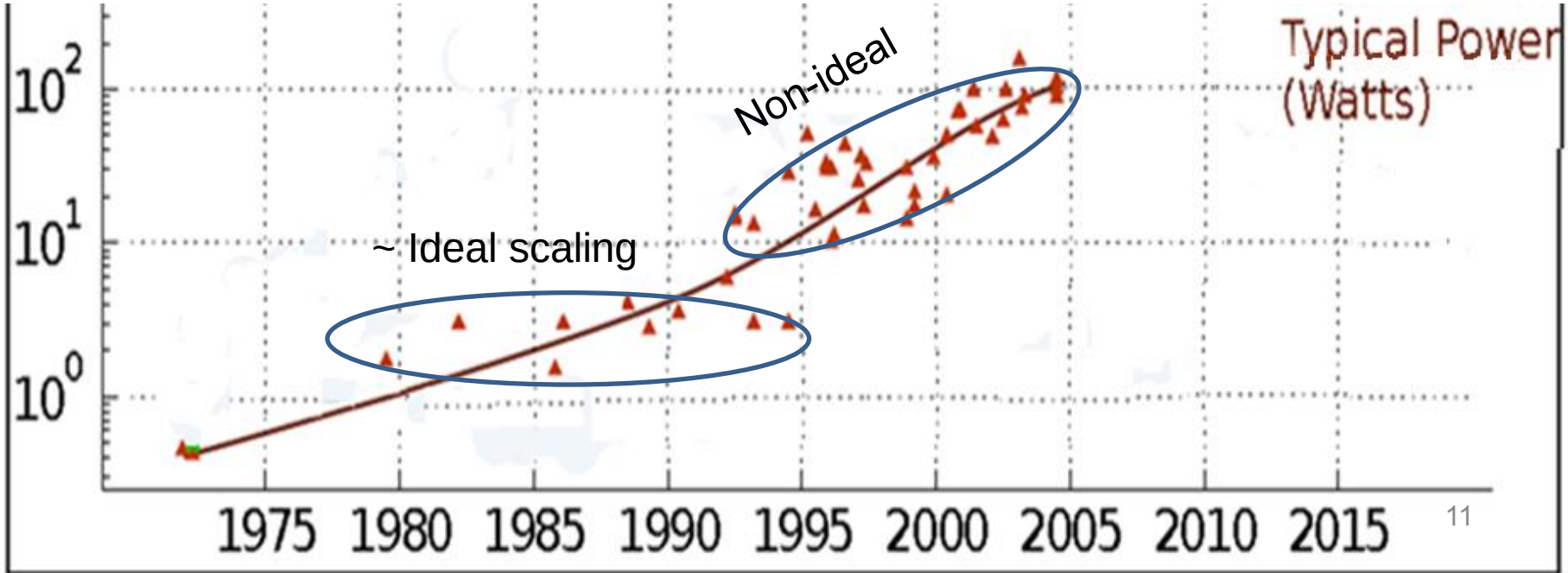
What were the reasons for growth in power dissipation?



What were the reasons for growth in power dissipation?

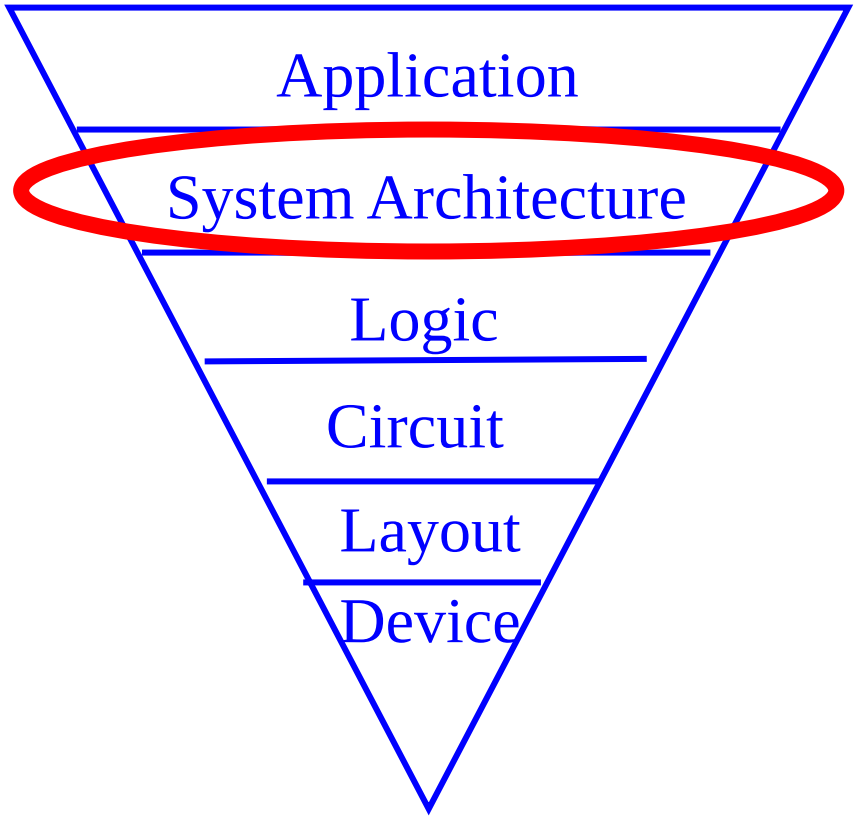
1. **Non-ideal scaling**
   - Classical Dennard scaling = Fixed power for a fixed chip area.
   - Many parameters were not scaled ideally:
     Supply voltage, Frequency, Leakage, Interconnect, ……

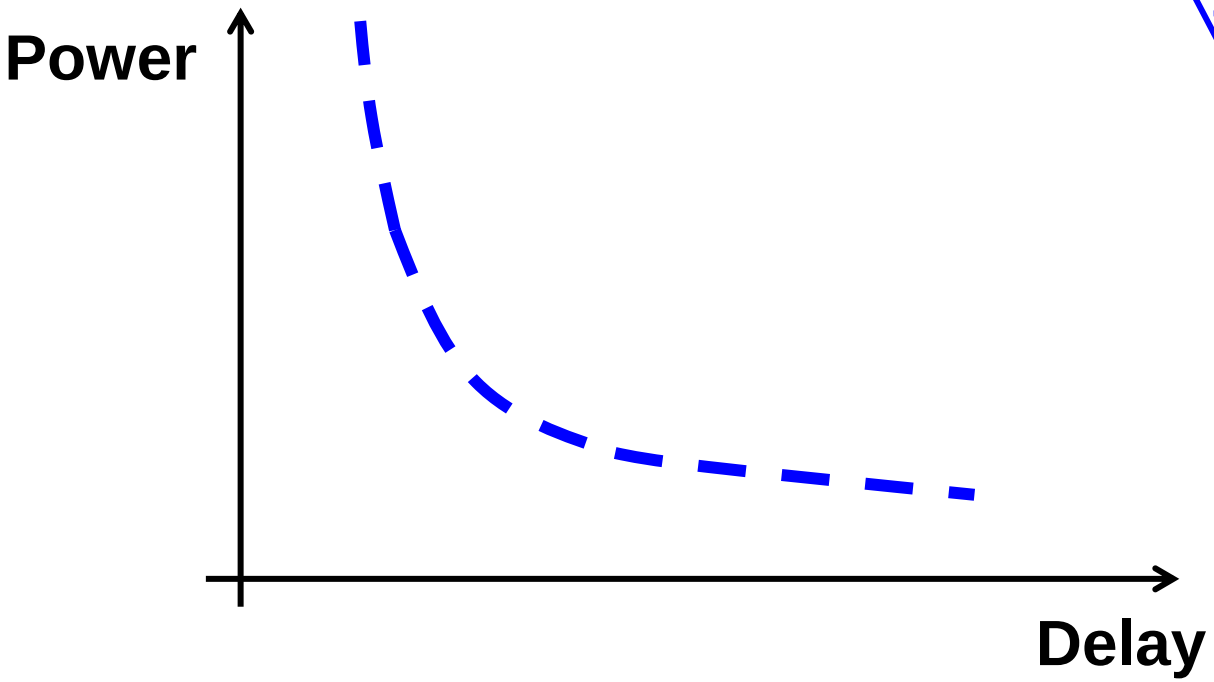
2. **Wasteful spending of power**
   while optimizing for performance



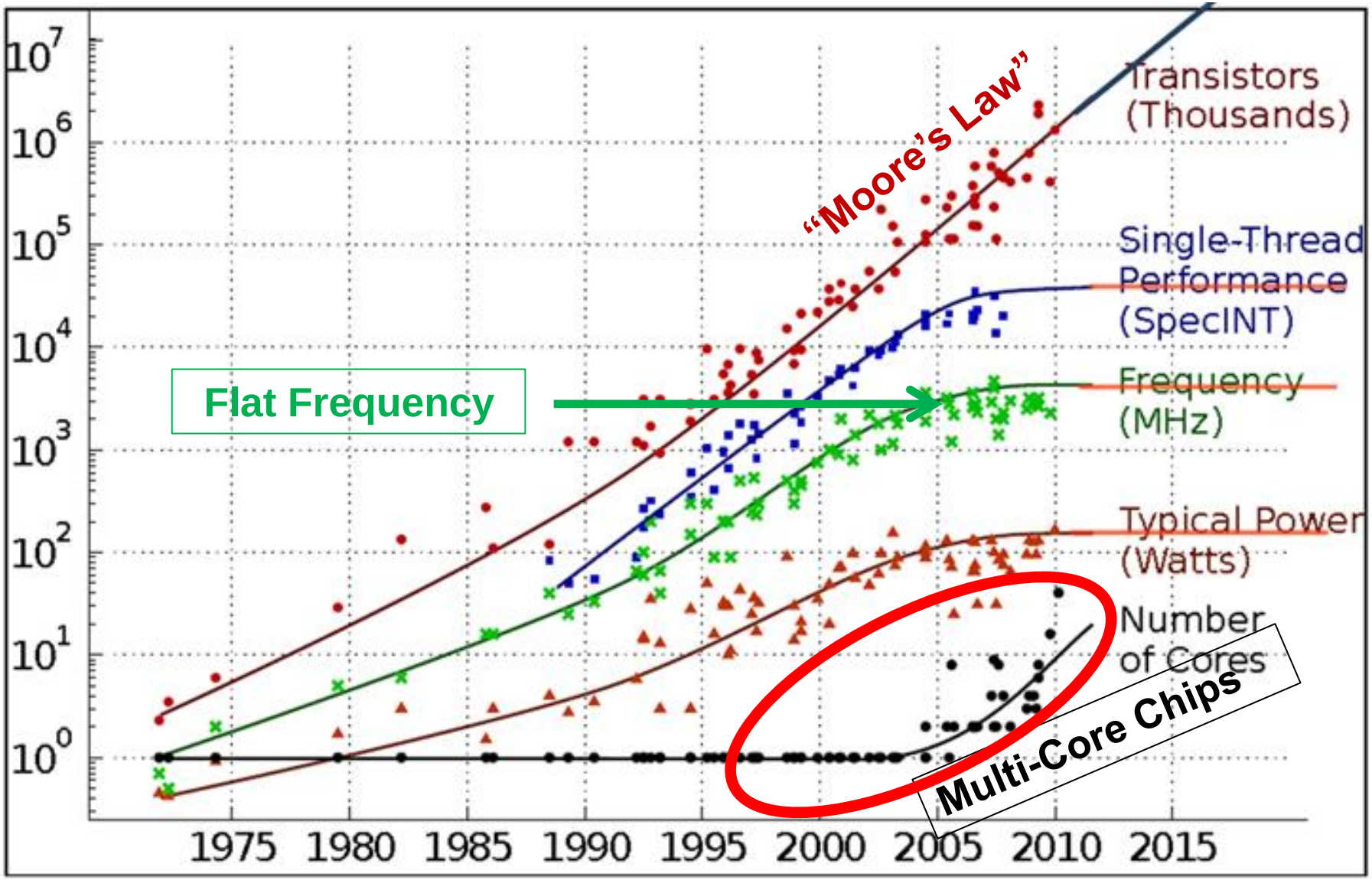
What has been done to stop the growth of power?

- **Savings at all levels**
  - Power is cumulative
- **Reduction of waste**
- **Compromise on speed**



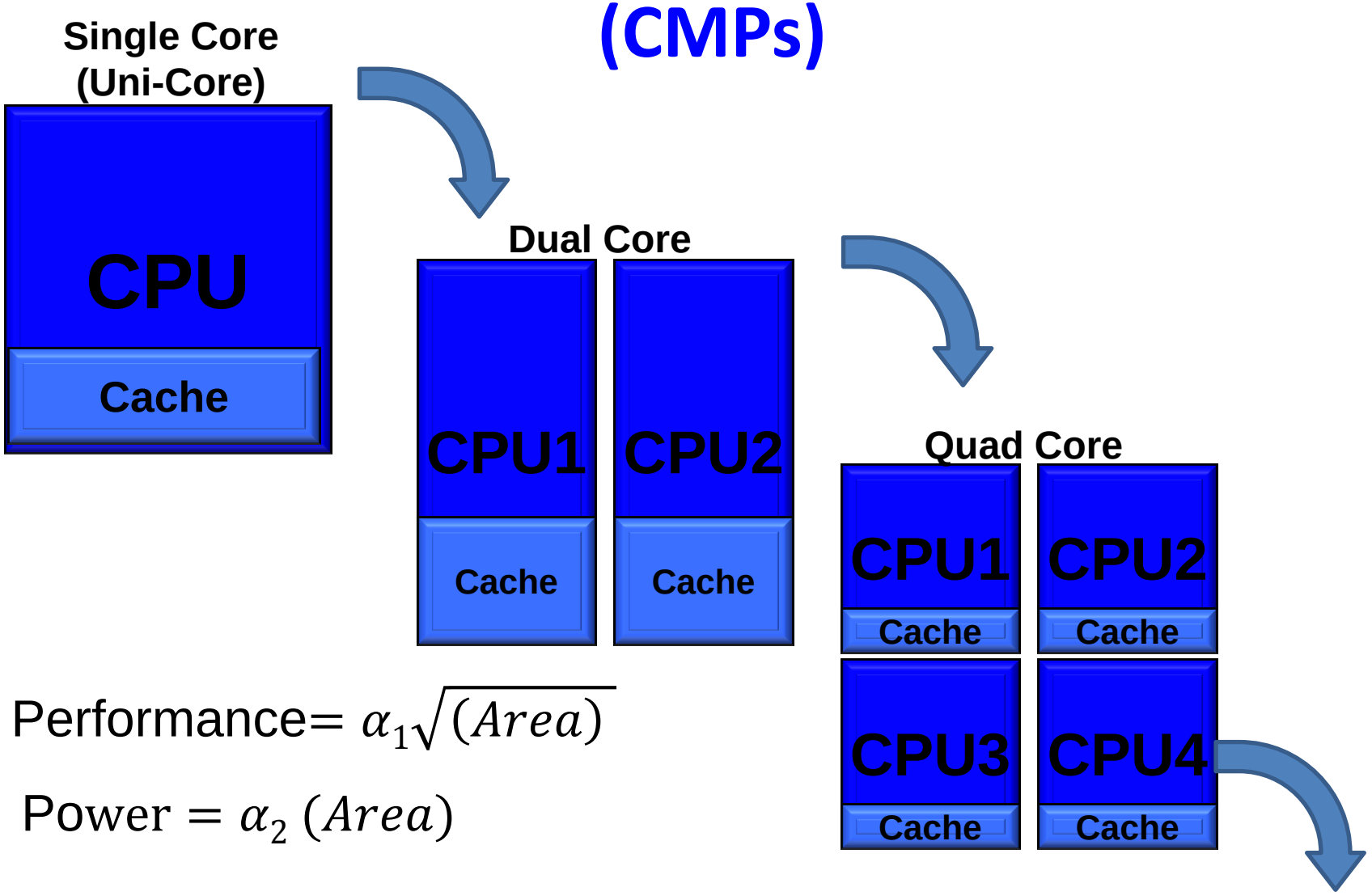


Main Architectural Changes in Microprocessors Since 2005



http://cpudb.stanford.edu/

Evolution of Multi-Core Chips (CMPs)



Performance= $\alpha_1\sqrt{(Area)}$

Power = $\alpha_2\,(Area)$

* F. Pollack, MICRO 1999.
* S. Borkar, DAC 2007.

Why Big Uni-Cores are Inefficient?

The Interconnect Bottleneck

Interconnect Delay
is dominant

* H. Bakoglu and J. Meindl, TVLSI 1985
* M. Bohr, IEDM 1995

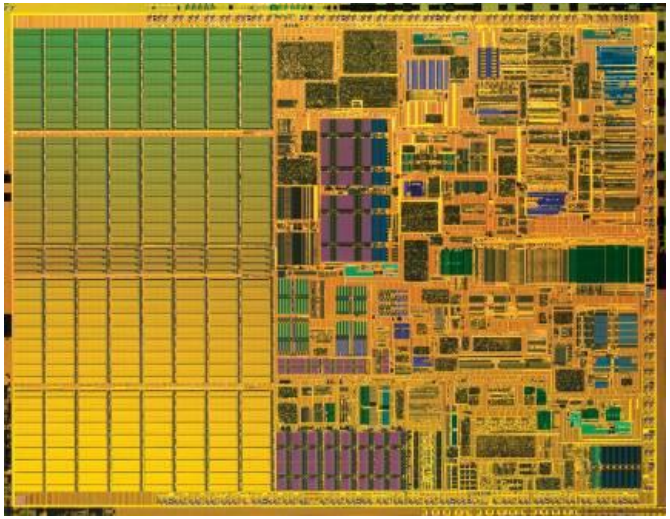
Why Big Uni-Cores are Inefficient?

The Interconnect Bottleneck

Interconnect Delay
is dominant

Interconnect Power
is dominant

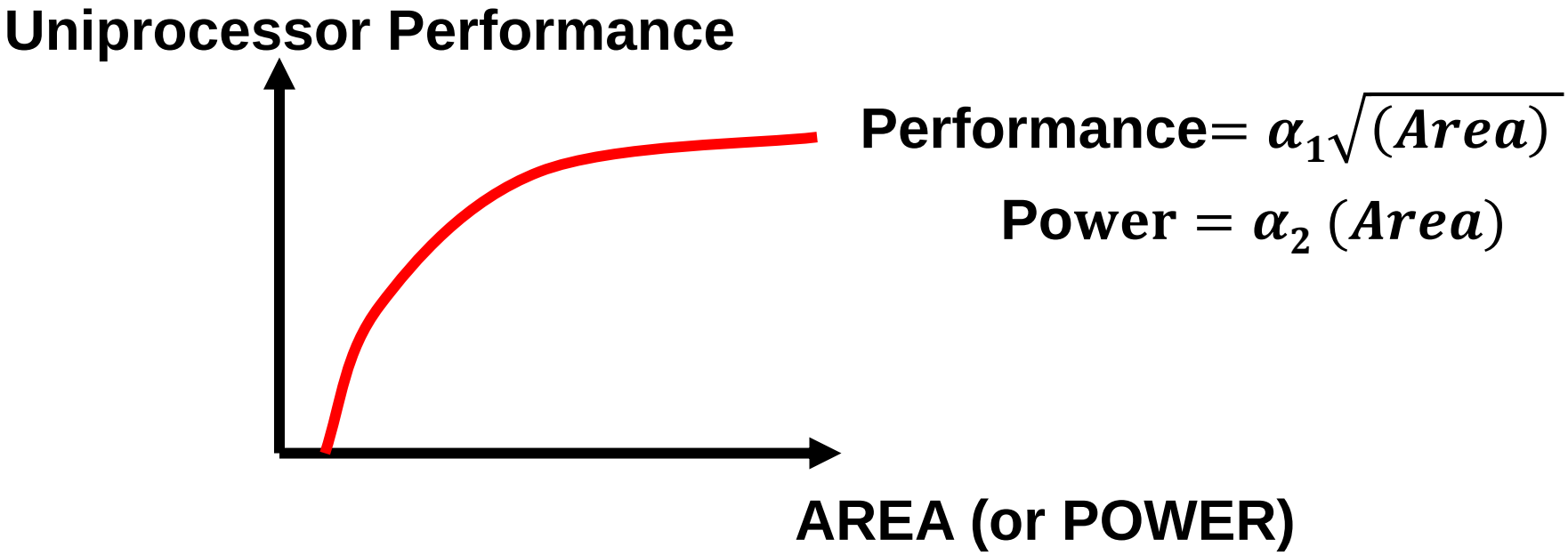
Bit-Transportation energy
is larger than
computation energy!!!



* H. Bakoglu and J. Meindl, TVLSI 1985
* M. Bohr, IEDM 1995

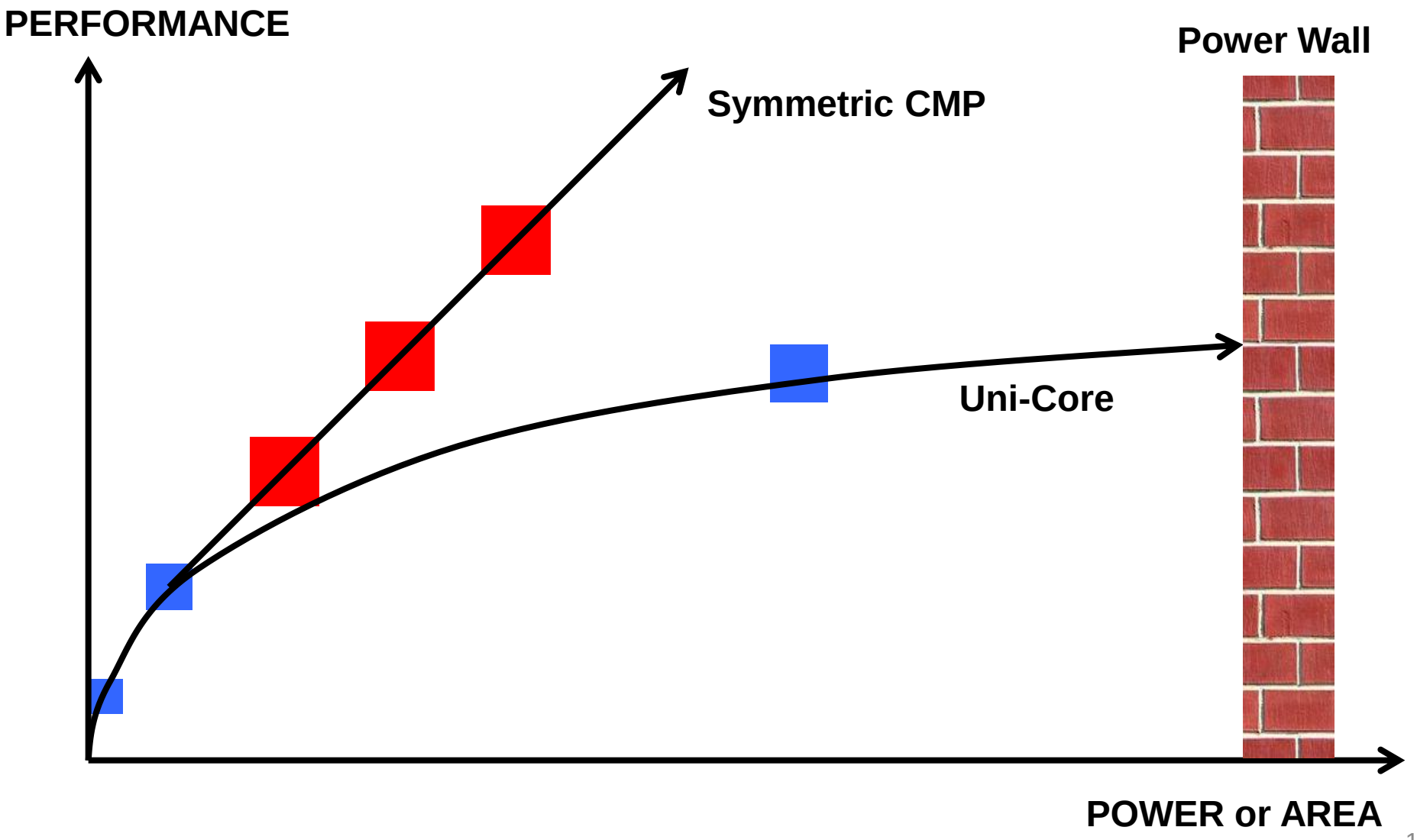
* N. Magen, A. Kolodny, U. Weiser and N. Shamir, "Interconnect-Related Energy dissipation in a Low-Power Microprocessor", Proc. SLIP, 2004.

Pollack's Empirical Rule



* F. Pollack, MICRO 1999.
* S. Borkar, DAC 2007.

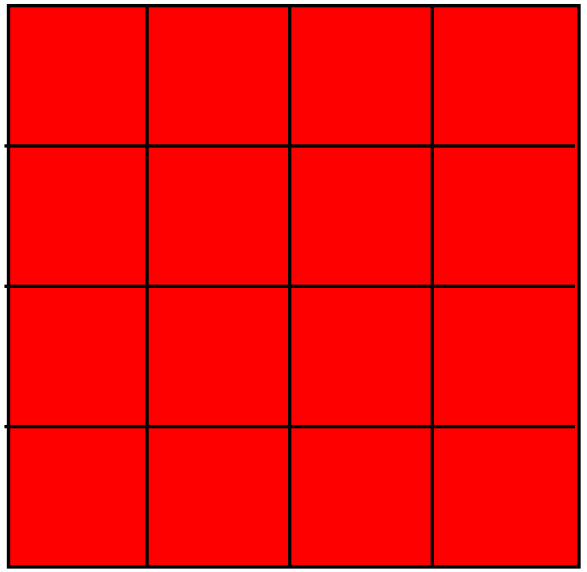
Processor Architectures:
Uni-Core, Symmetric multicore



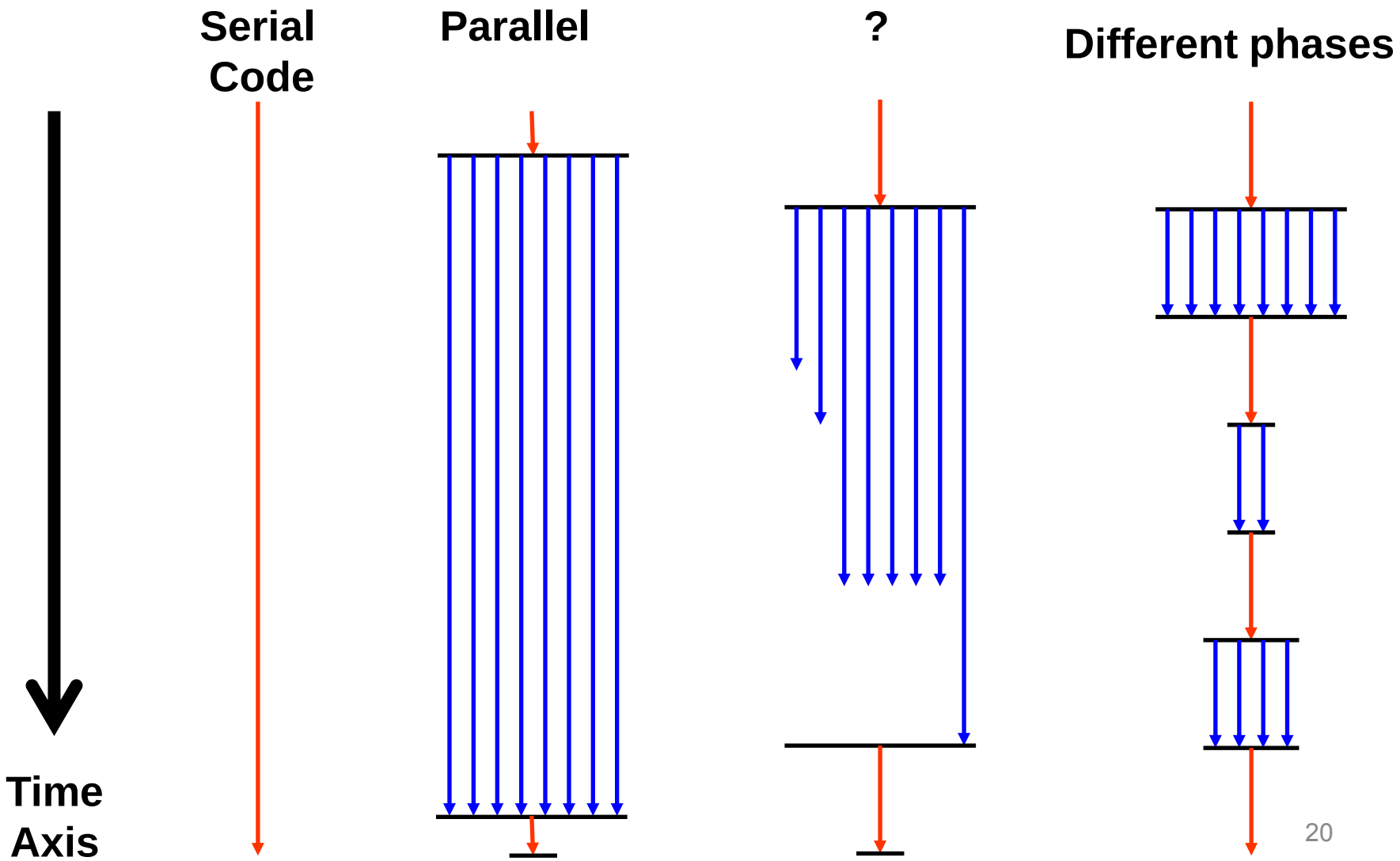
Locality Principle

- Power-efficiency requires many parallel **local computations**

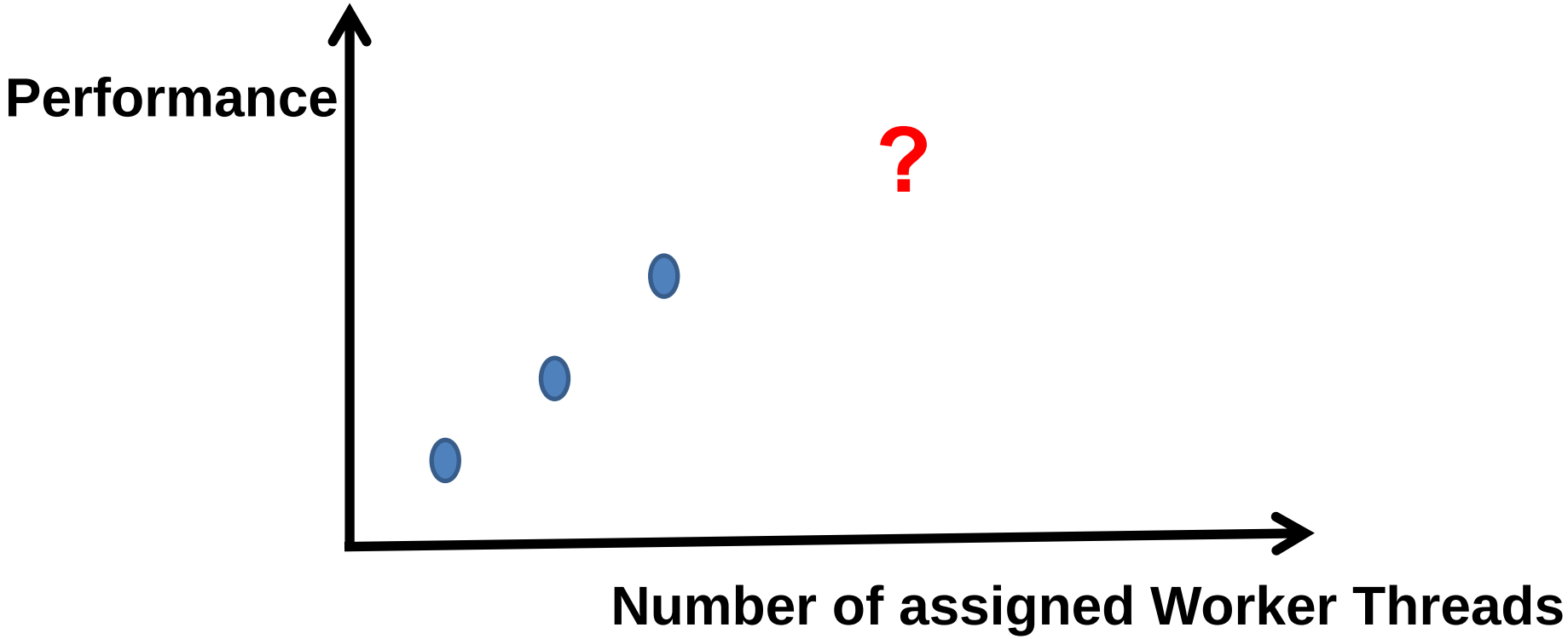
- To utilize multiple cores, software needs to run in **parallel threads**



How Much Parallelism Exists in Real Programs?



Characterizing a Multi-Threaded Software Program



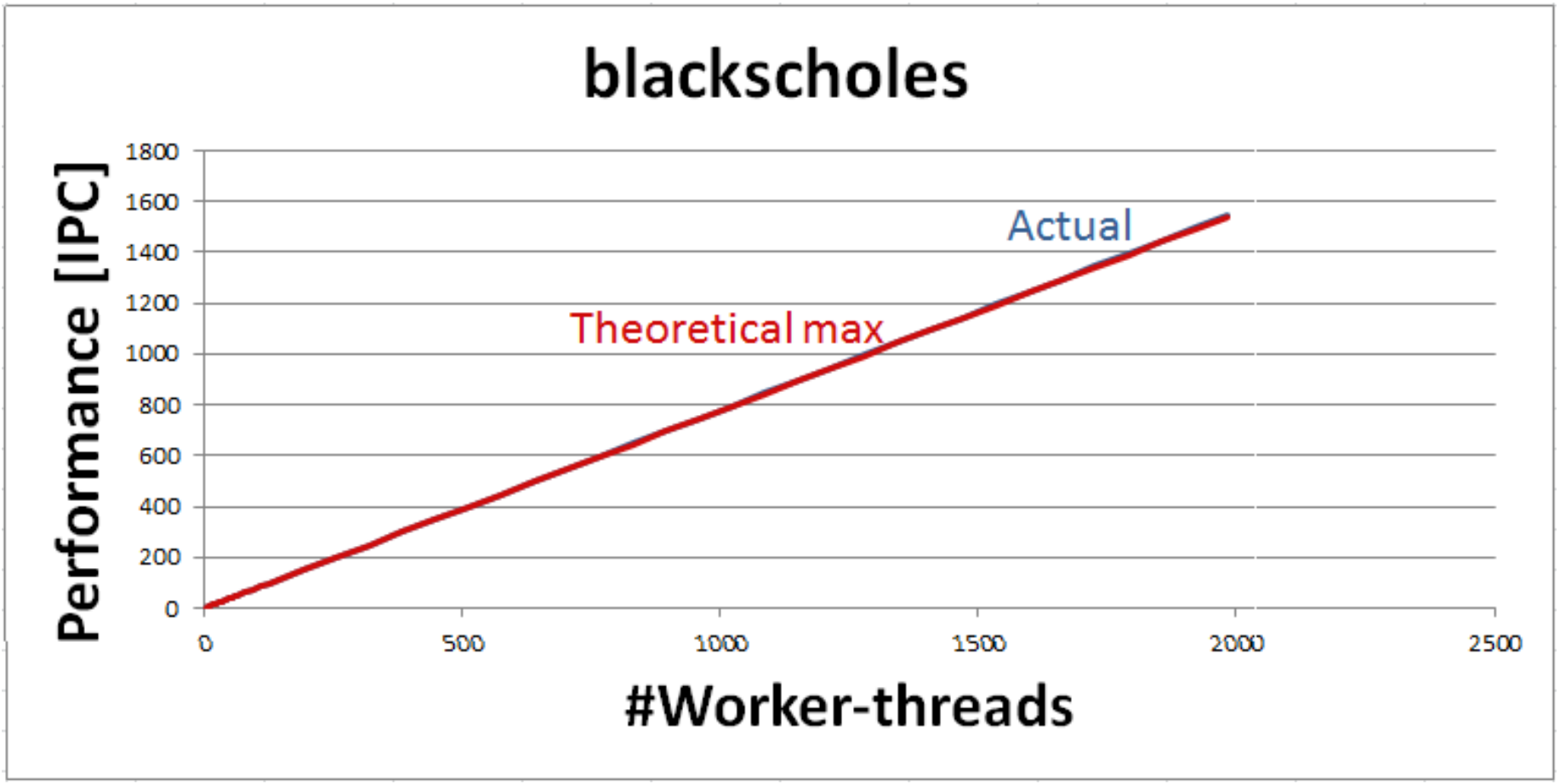
Study of Parallelism in Real Software

- Use **PARSEC** benchmark suite
- Capture the parallelism inherent in the **algorithm**
  - Limited by **inter-thread synchronization**
- Assume an **"ideal machine"**
  - Architecture model with no constraints
  - No shared resources
  - Perfect memory system – 1 cycle latency

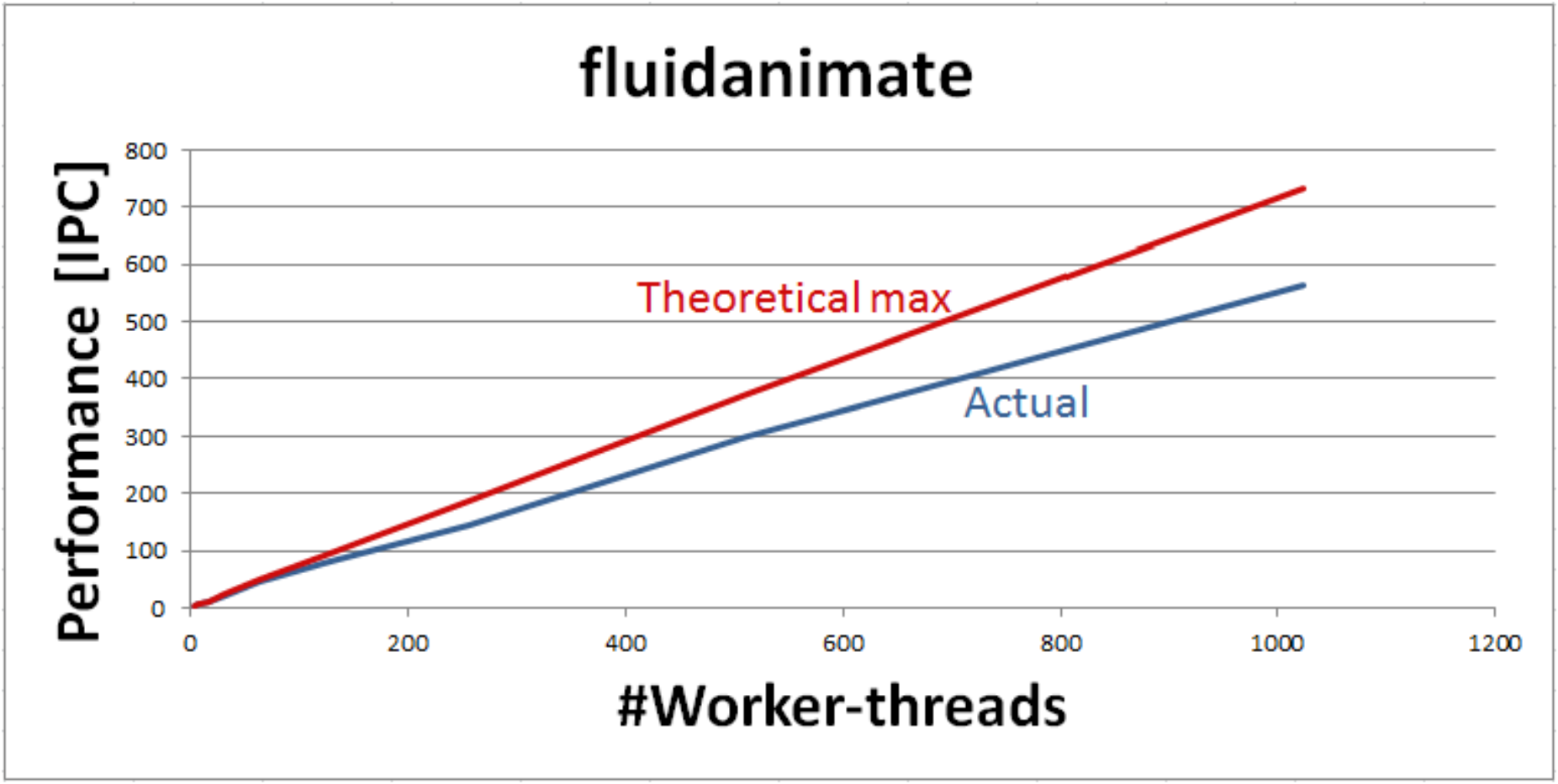
* O. Itzhak, I. Keidar, A. Kolodny and U. Weiser, "Performance scalability and dynamic behavior of Parsec benchmarks on many-core processors," *SFMA* 2014

Perfect parallelism scalability: blackscholes

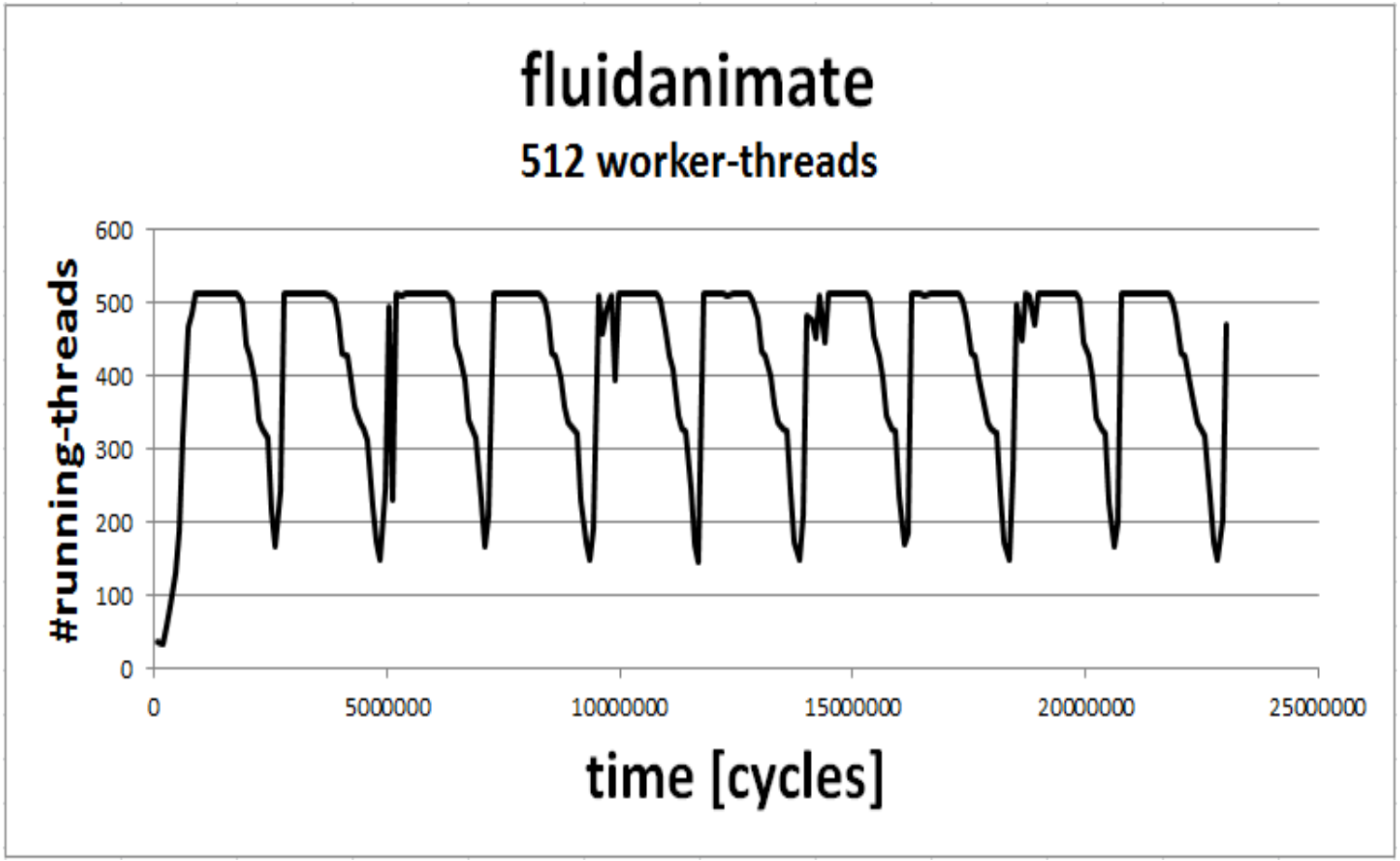
("Embarrassingly Parallel")

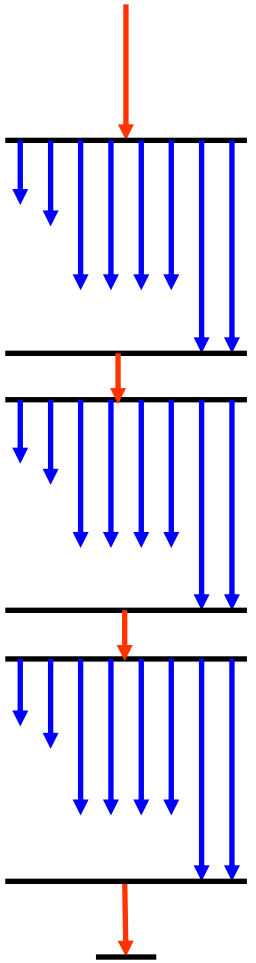


Good parallelism scalability: fluidanimate

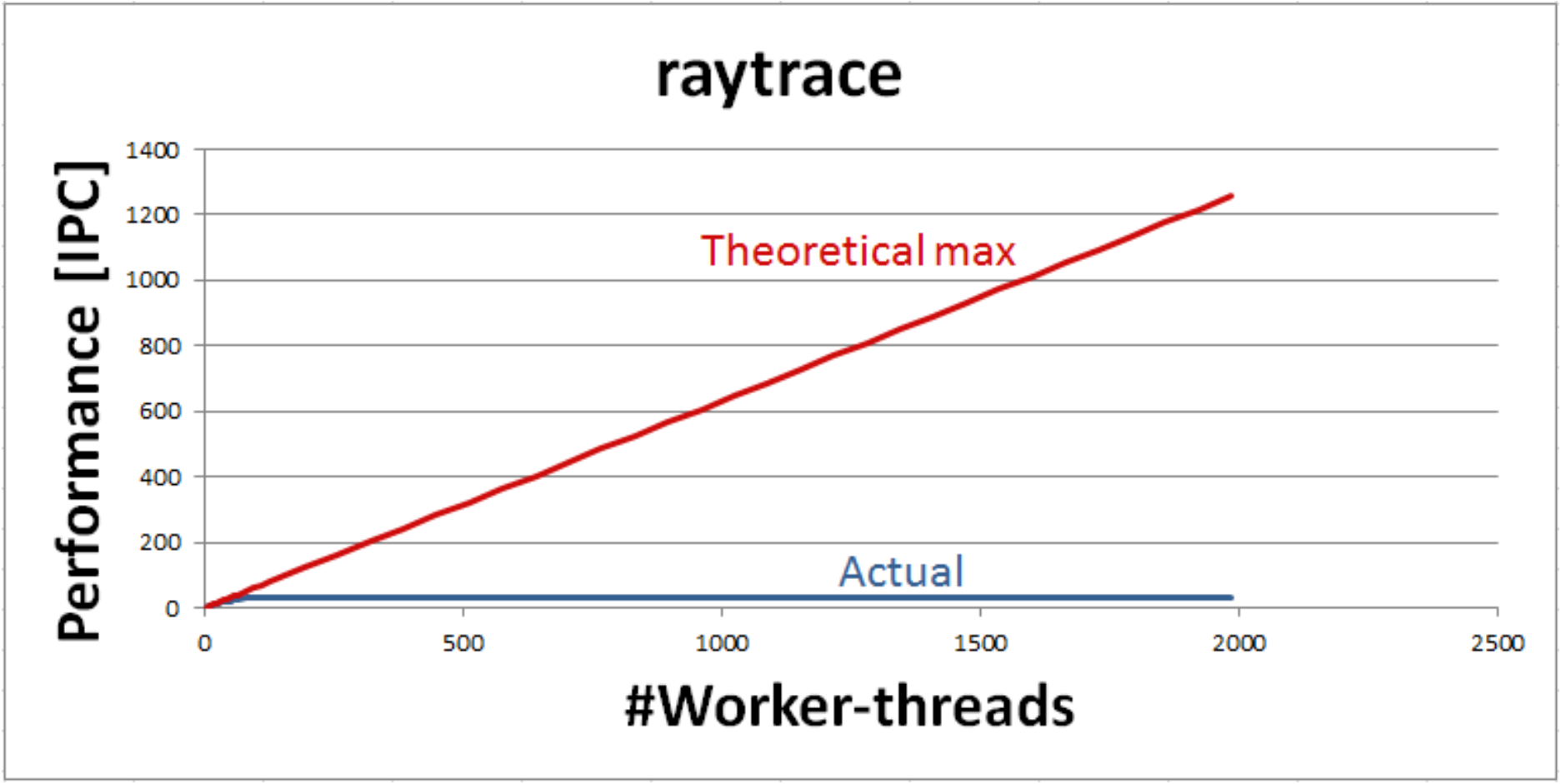


Explaining the Difference:
Some threads work part-time only!



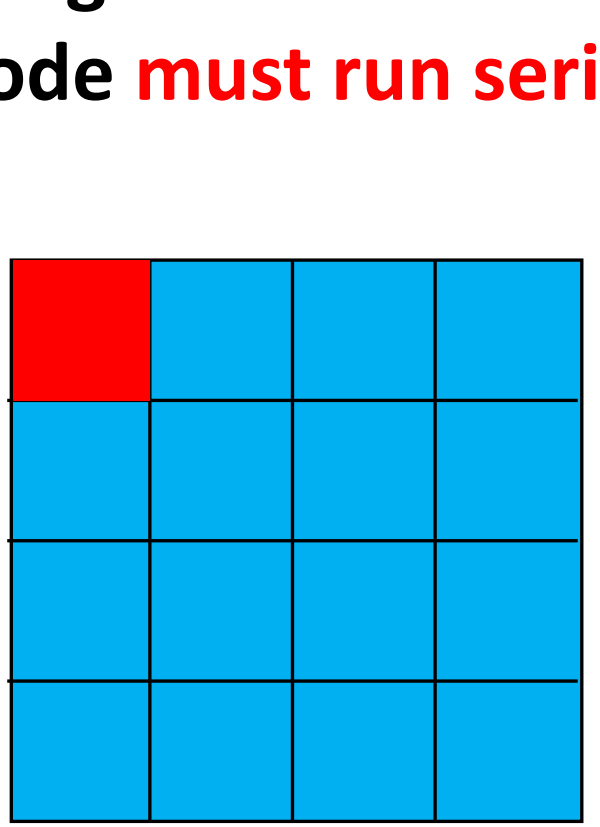


Poor parallelism scalability: raytrace



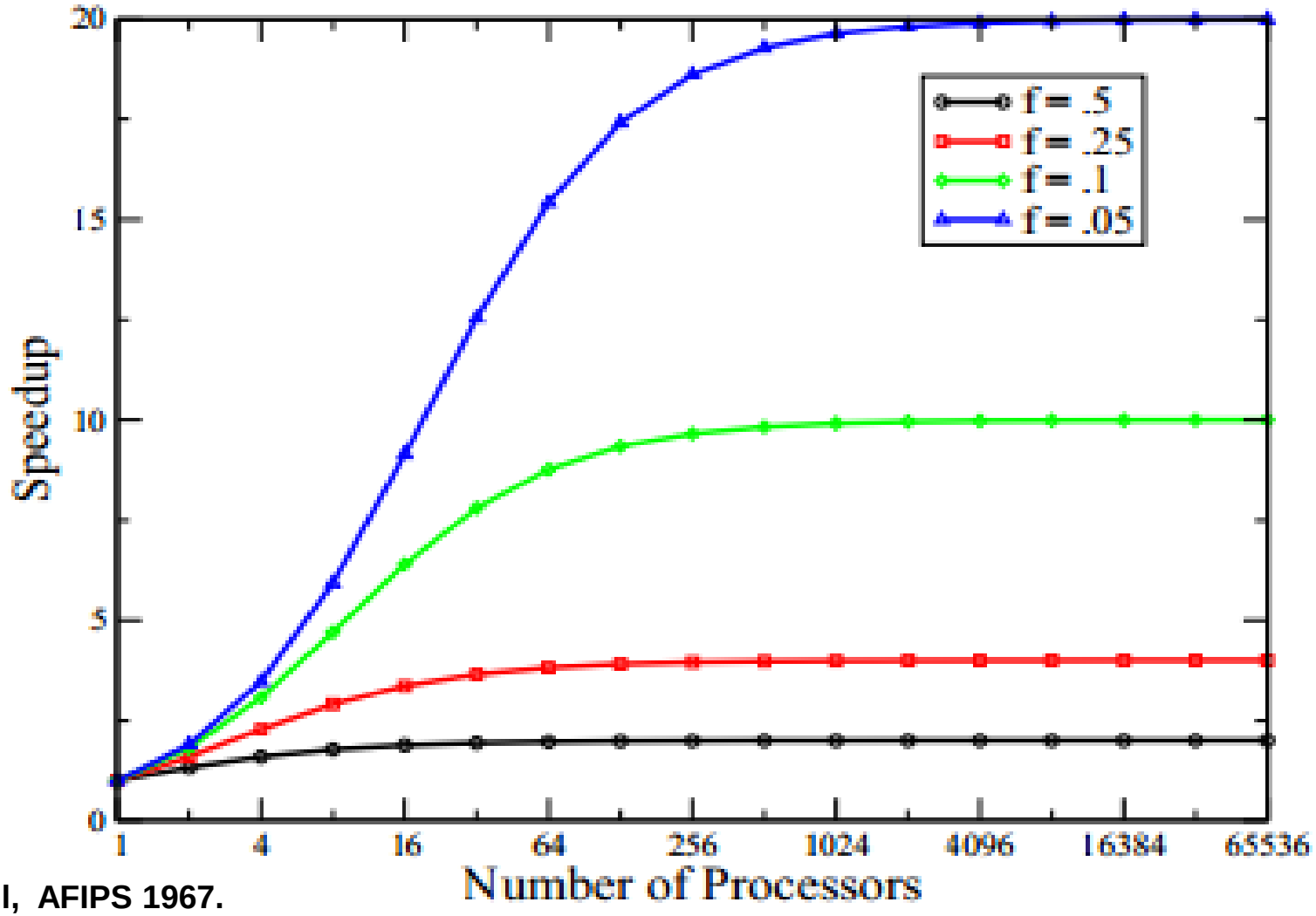
Challenge of Parallel Programming

- Many multi-threaded applications are **not scalable**: **They cannot exploit a large number of cores because some of the code must run serially.**



Amdahl's Principle

If a fraction *f* of the code must run serially, parallel speedup is limited to *1/f* when all cores are identical



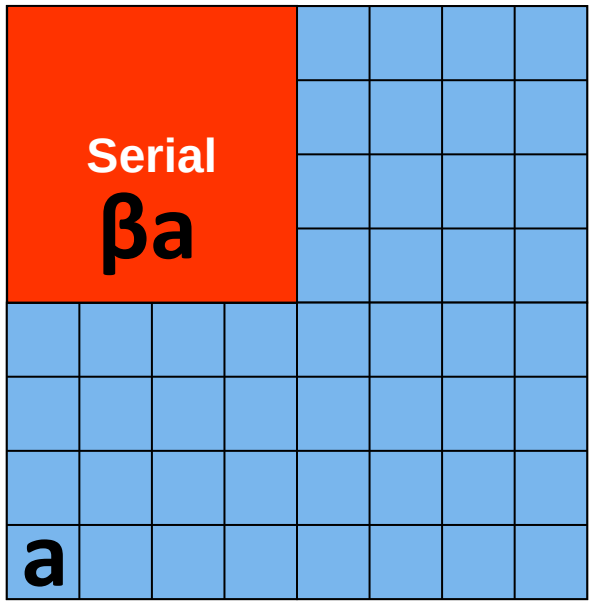
*** G. Amdahl, AFIPS 1967.**

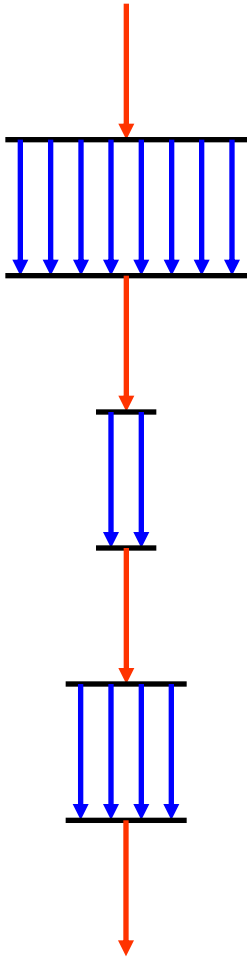
"Hurry UP" Principle: Waiting Wastes Power

- Idle cores waste power (clock, leakage,….)
- Other system components consume power
- It makes sense to hurry up, finish all work and shut down

Asymmetric (=Heterogeneous) **Multi-Core**

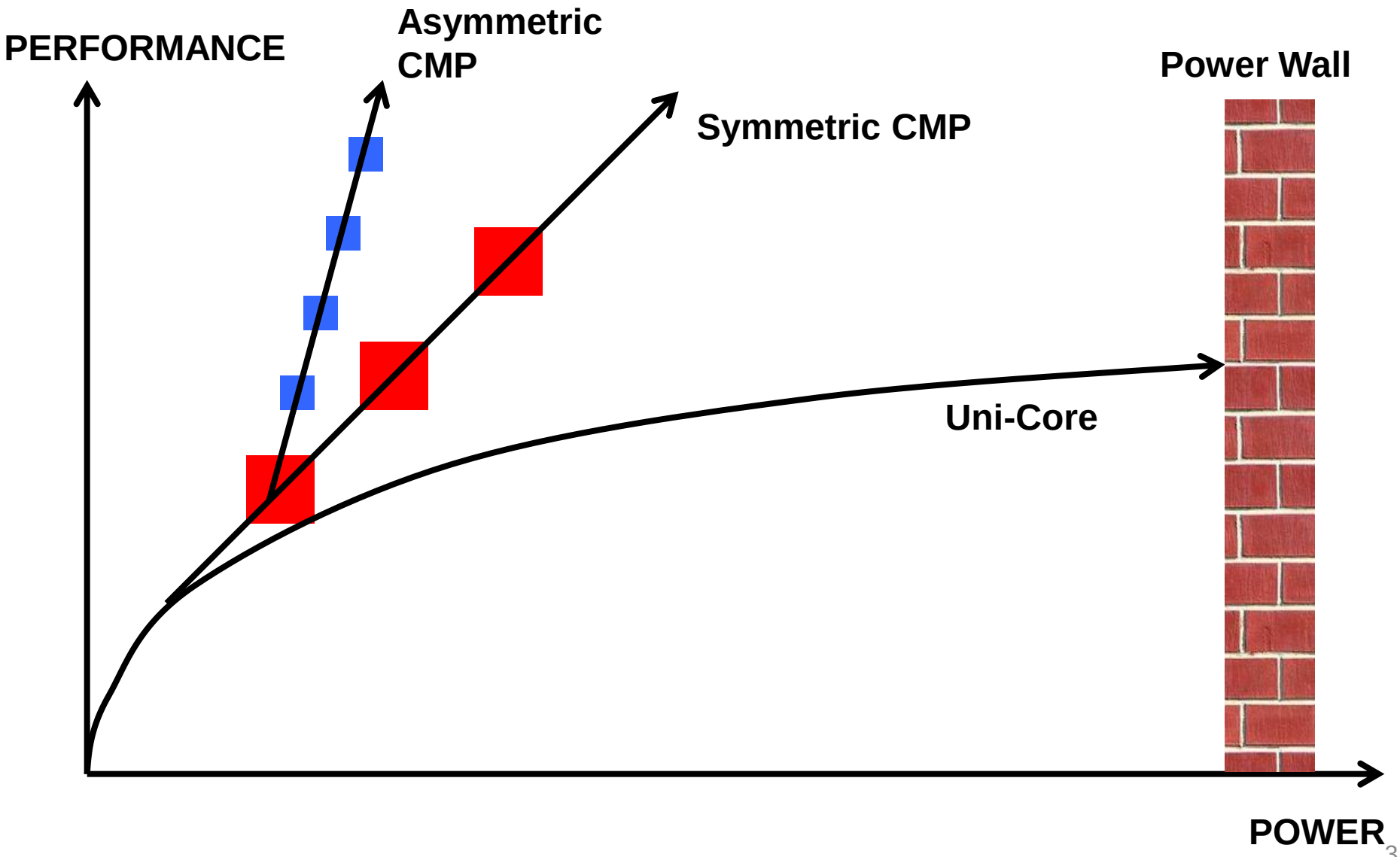
- Small cores of area: a
- Large core area: βa – used for serial code
- Parallel phases execute on all cores





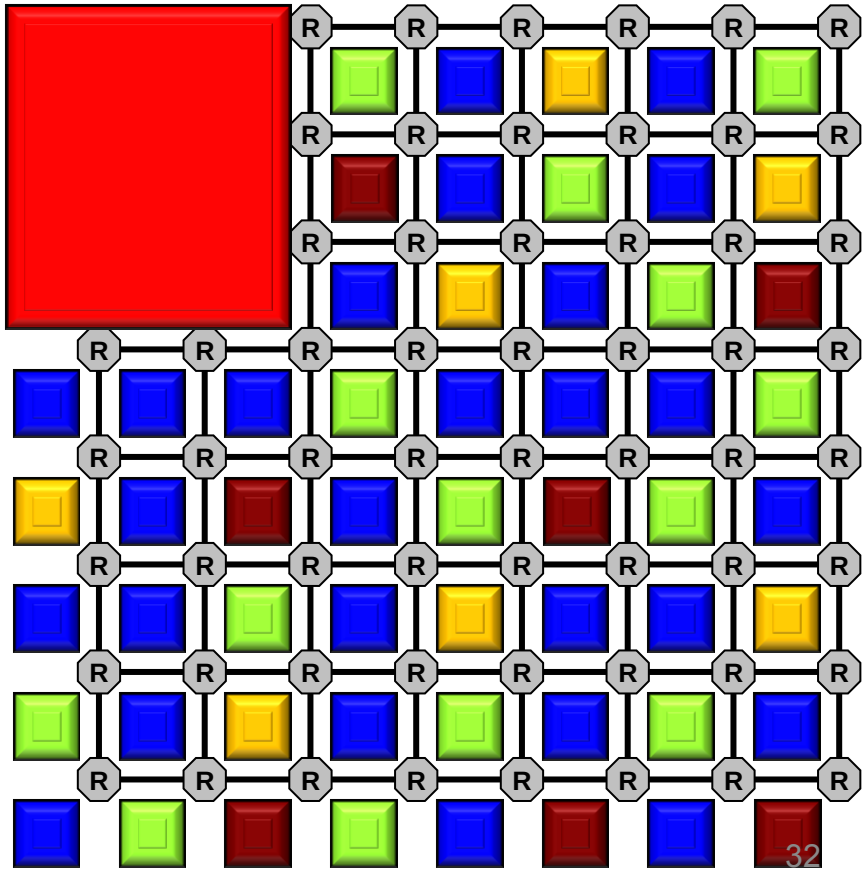
* T. Morad, U. Weiser, A. Kolodny, M. Valero and E. Ayguade,
"Performance, Power Efficiency and Scalability of Asymmetric Cluster Chip Multiprocessors,"
IEEE Computer Architecture Letters, vol. 4, 2005.

Processor Architectures:
Uni-Core, symmetric multicore, Asymmetric

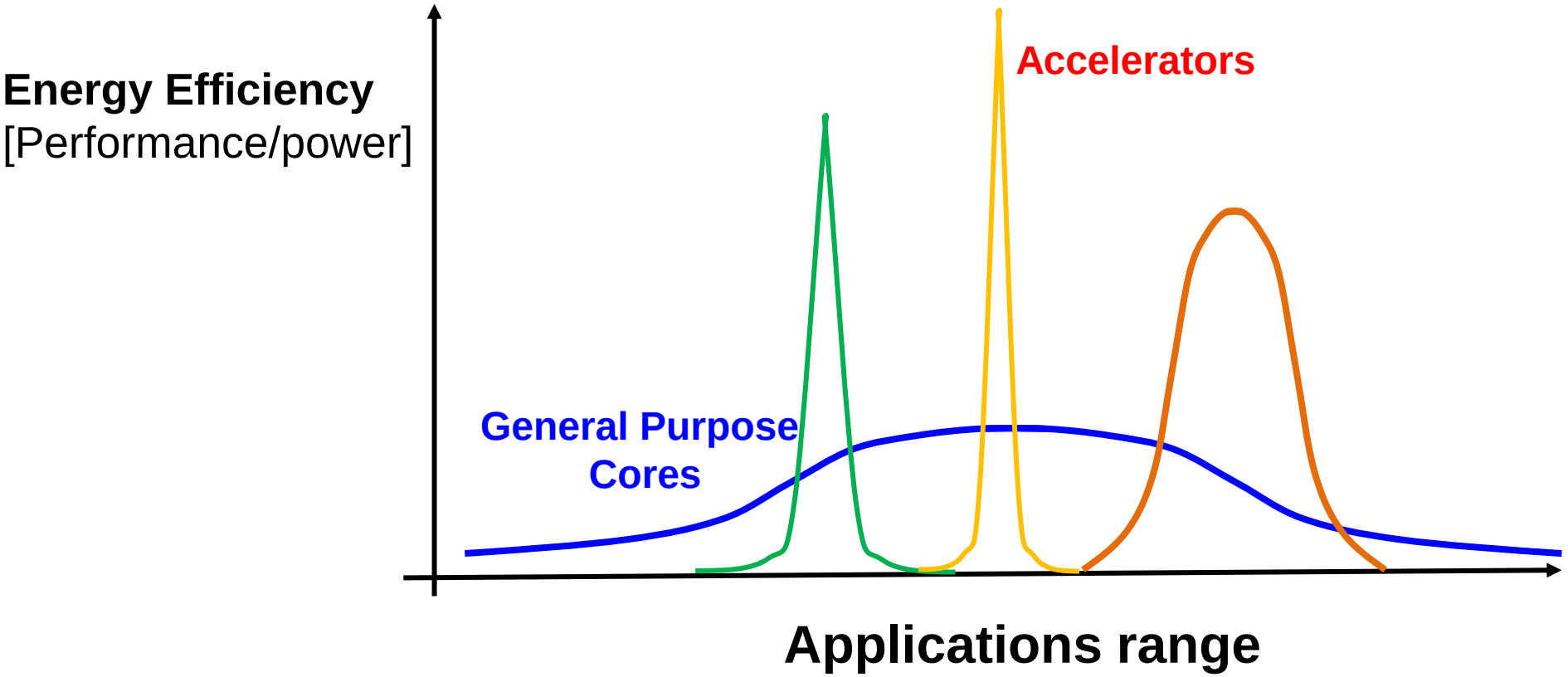


Future VLSI systems

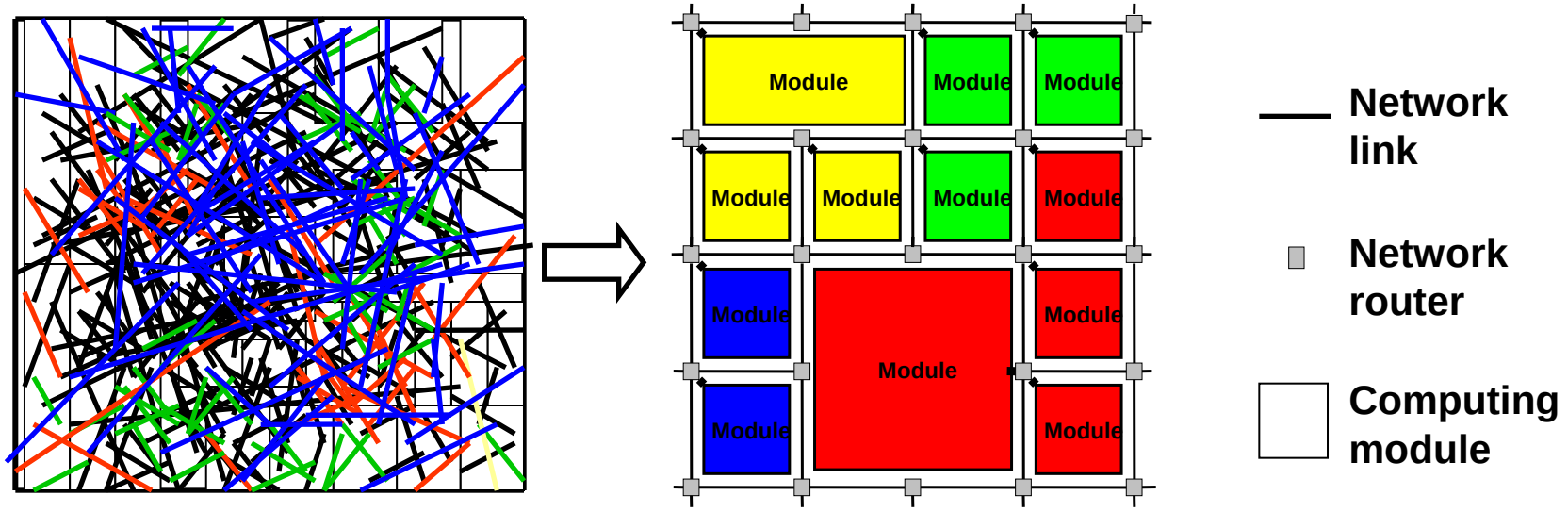
- Multiple computing cores + cache memory banks
- Special-purpose Accelerators
- Power Management:
  - Dynamic voltage and frequency scaling (DFVS)
  - "Dark silicon"
- Network on Chip (NoC)



Heterogeneous Computing: Application Specific Accelerators



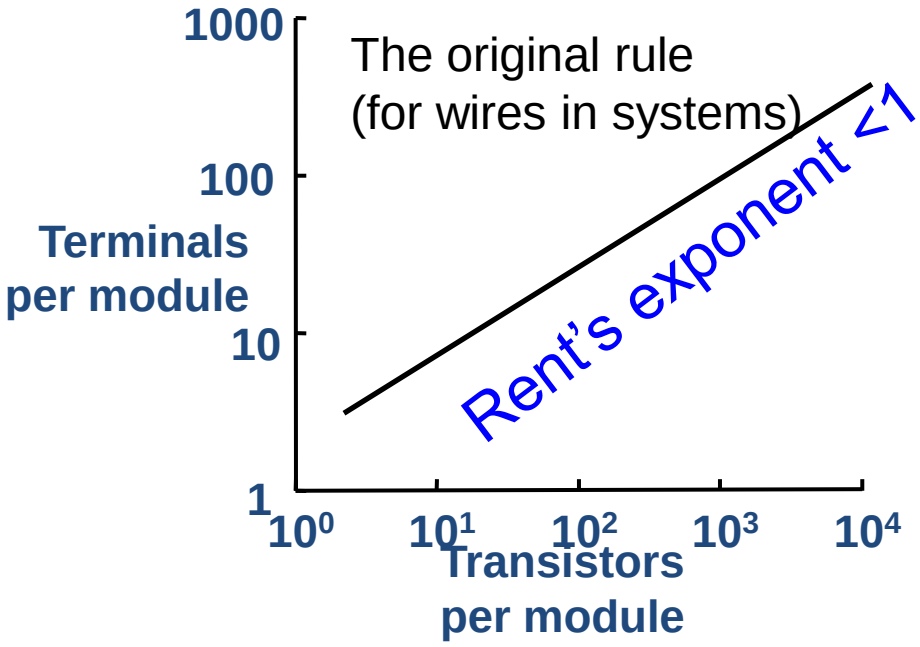
Packet-Switched Network-on-Chip (NoC)



- **Network** instead of **dedicated wires** and **buses**
  - **Inherently parallel**
  - **Efficient sharing of wires**
- **<u>Advantages:</u> scalable, cost-effective throughput**

* E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny - Journal of systems architecture, 2004

Rent's Principle (locality revisited)



Rent's Principle (locality revisited)

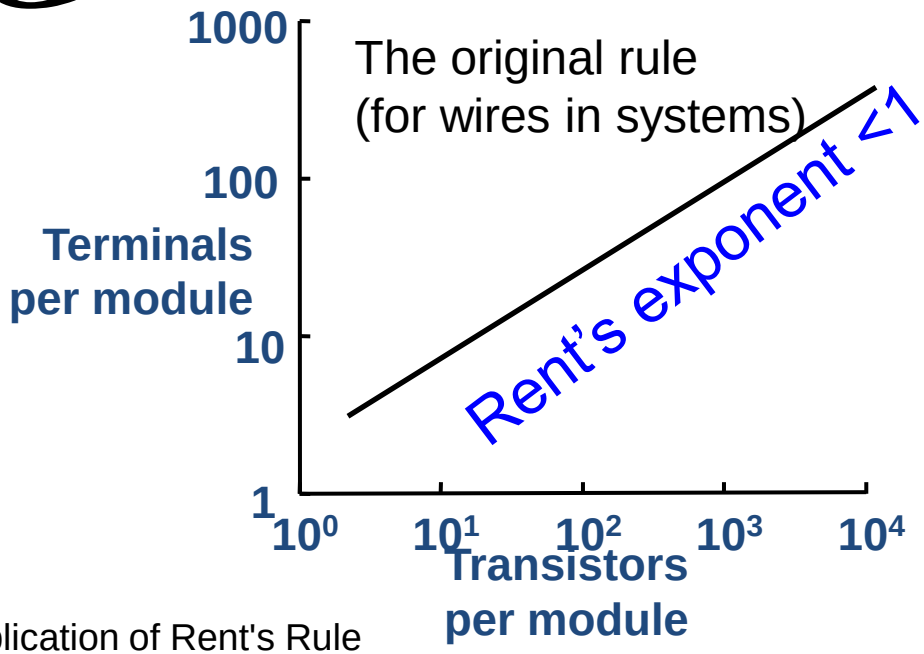
- **Rent's Rule for NOCs – Bandwidth version**

Average bandwidth per tile

Number of tiles

Bandwidth → $B = kG^r$ ← Rent's coefficient

- Low r – local traffic
- High r – global traffic



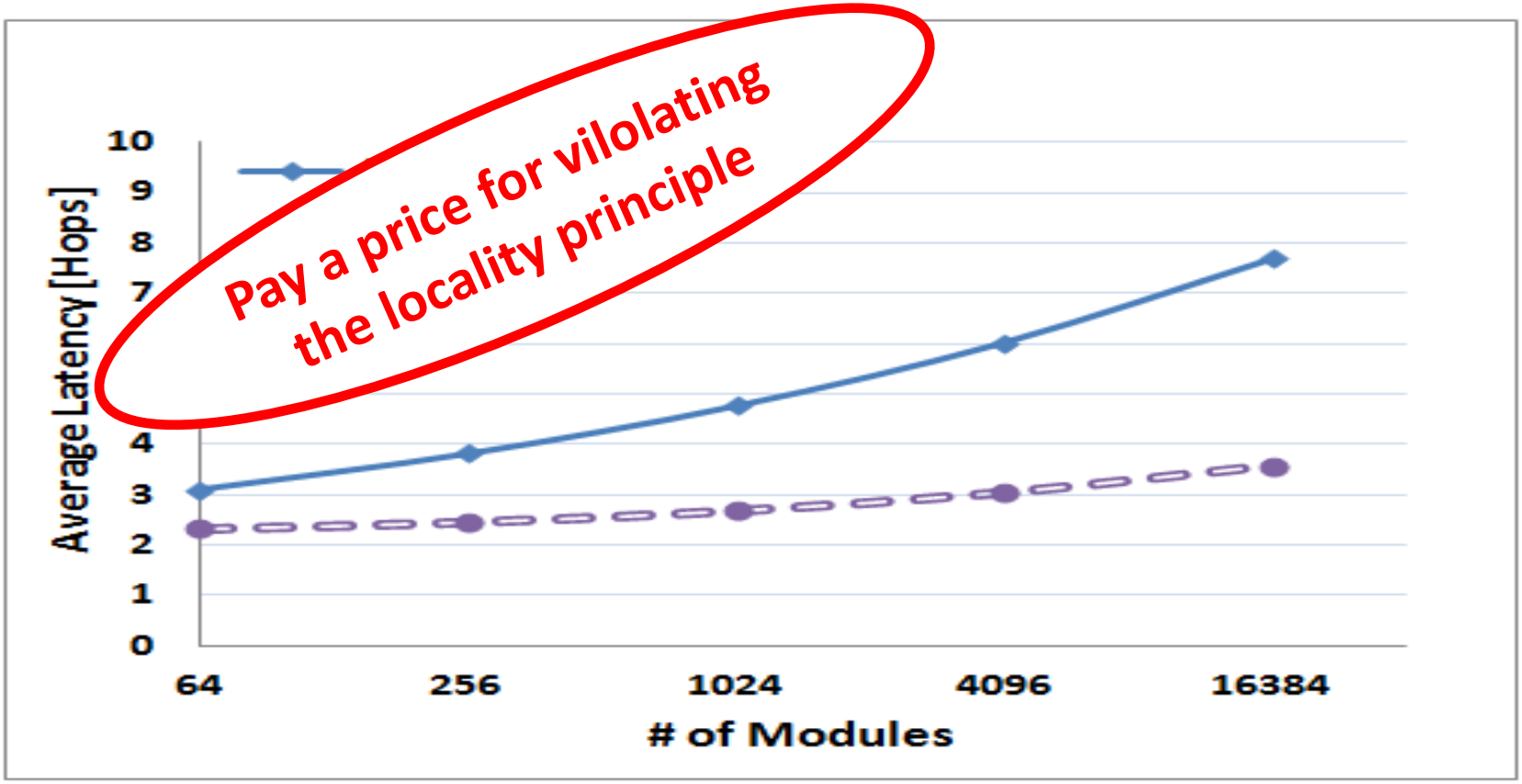
* D. Greenfield, A. Banerjee, J.G. Lee, S. Moore, "Implication of Rent's Rule for NoC Design and Its Fault-Tolerance," *NOCS'07*

***** Heirman et al., "Rent's Rule and Parallel Programs: Characterizing Network Traffic Behaviour", SLIP 2008

Basic Disadvantage of NoCs: Latency

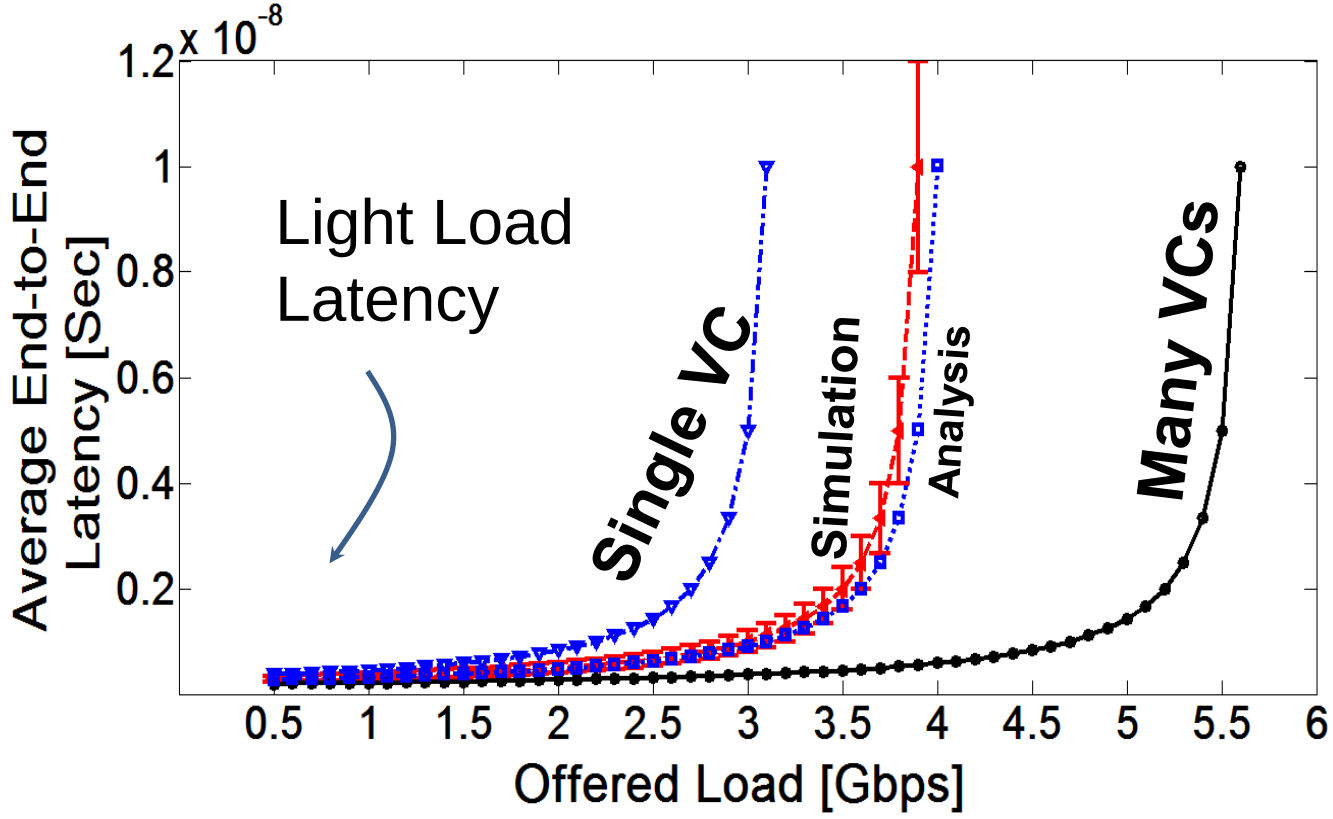
Few packets, which go long distances across the chip:

- Consume **most** of the network's bandwidth
- Increase the **average latency** at light load



* R. Manevich, I. Cidon, and A. Kolodny, "Handling Global Traffic in Future CMP NoCs", *SLIP* 2012

Latency is even worse when NoC is Loaded!

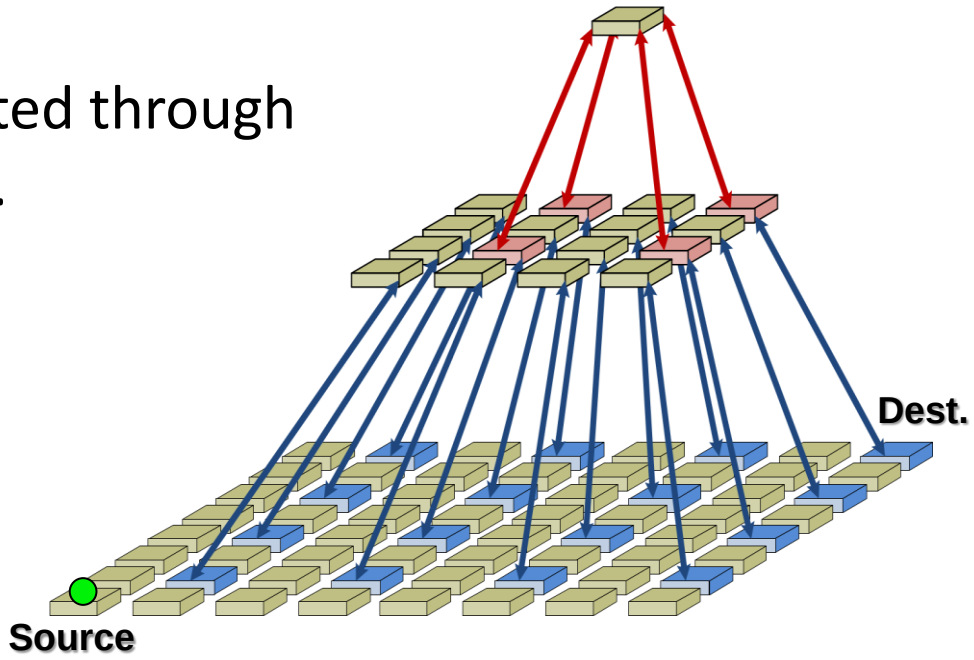


A loaded network quickly reaches a saturation point!

* Y. Ben-Itzhak, et al. "Delay analysis of wormhole based heterogeneous NoC." *NOCS* 2011.

Reducing Hop-Count by Express Links Example: PyraMesh Topology

- Hierarchical 2D mesh.
- Global packets are routed through higher hierarchy levels.



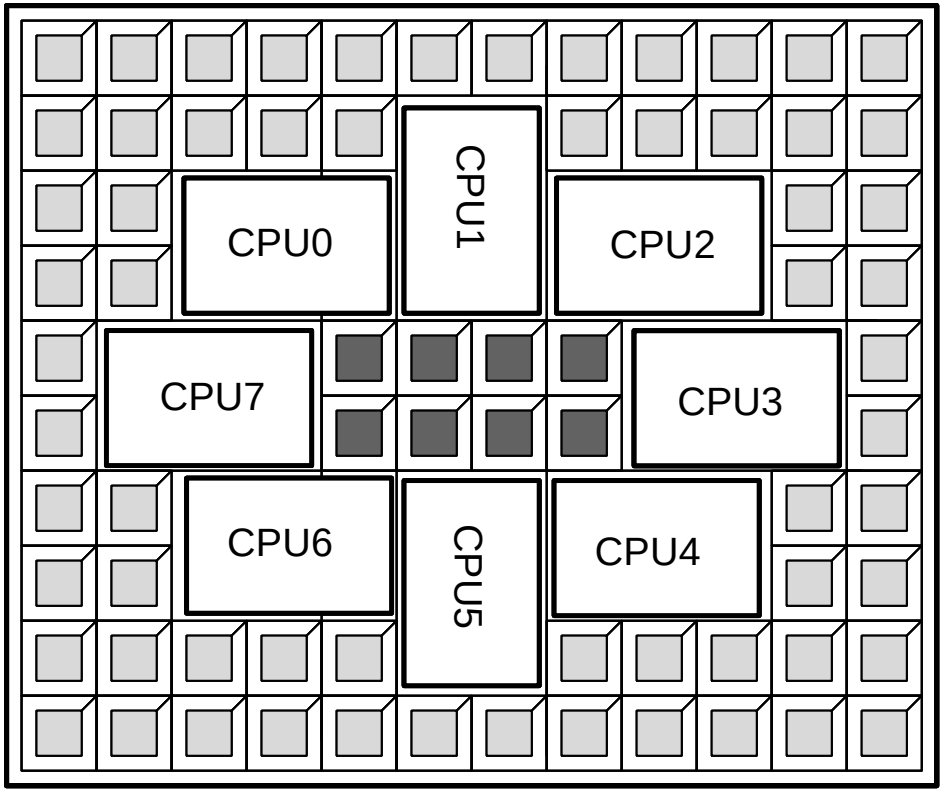
- ✓ Overall hops-count is reduced.
- ✓ Average latency is reduced.
- ✓ Average BW per router is reduced.

*** R. Manevich, I. Cidon, and A. Kolodny, "Handling Global Traffic in Future CMP NoCs", SLIP 2012**

NoC-Aware Chip Layout

Example: “Nahalal Architecture”

- **Partitioning of cache lines by “shared” vs. “private”**
- **Keep shared data in the center cache-banks**
- **Use outer cache-banks for private data**

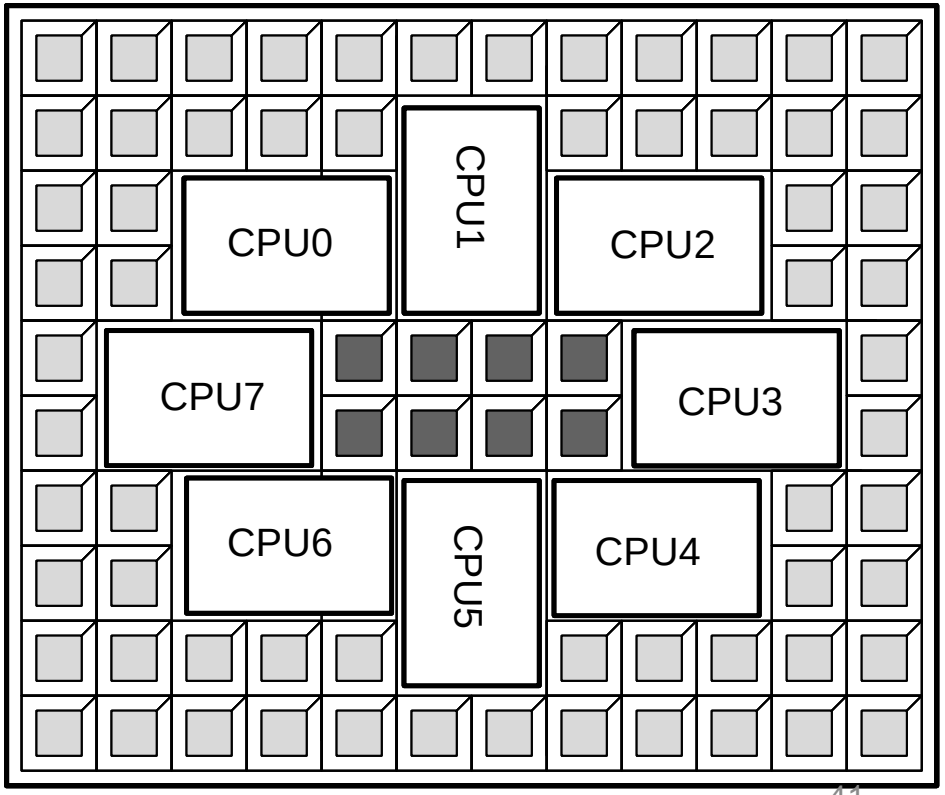


* Z. Guz et al., [SPAA-2008] , [CA-Letters’07]

NoC-Aware Chip Layout

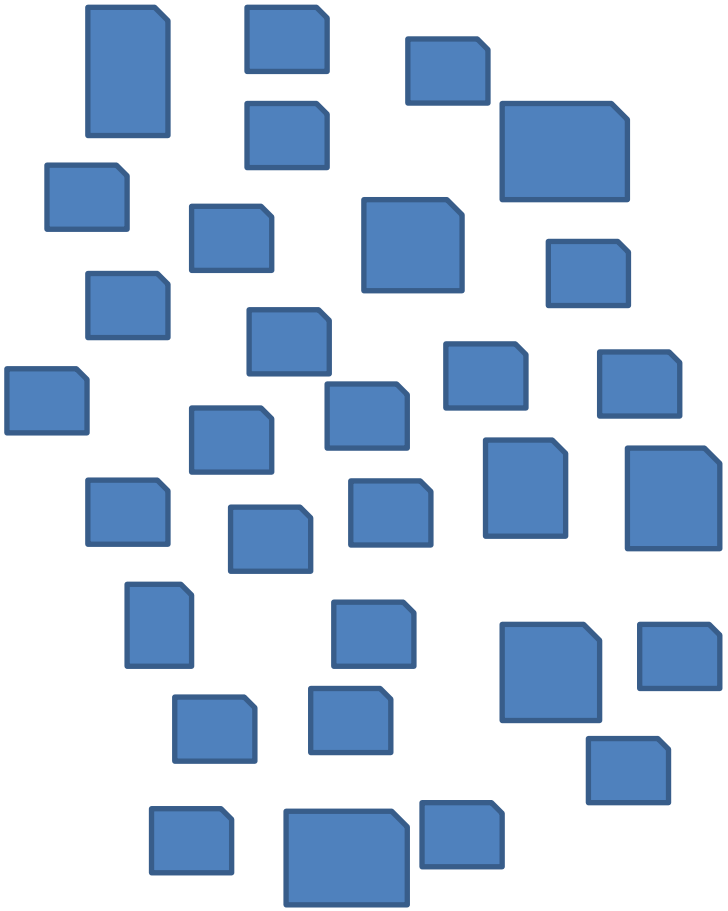
Example: "Nahalal Architecture"

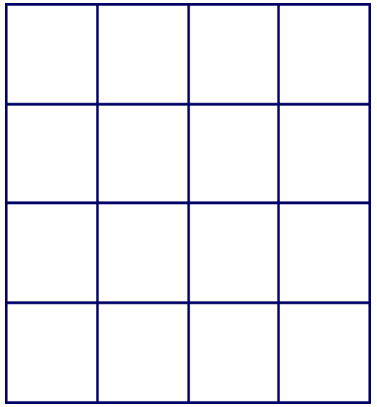
- **Partitioning of cache lines by "shared" vs. "private"**
- **Keep shared data in the center cache-banks**
- **Use outer cache-banks for private data**



* Z. Guz et al., [SPAA-2008] , [CA-Letters'07]

Bypassing The Challenge of Parallel Programming:
Run UNRELATED Programs on multi-core computers

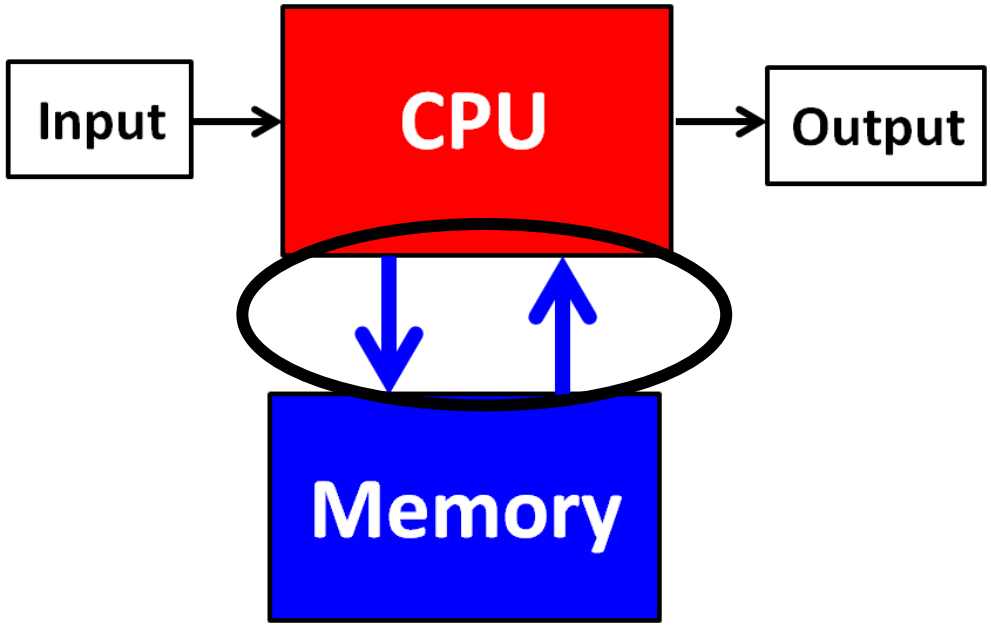


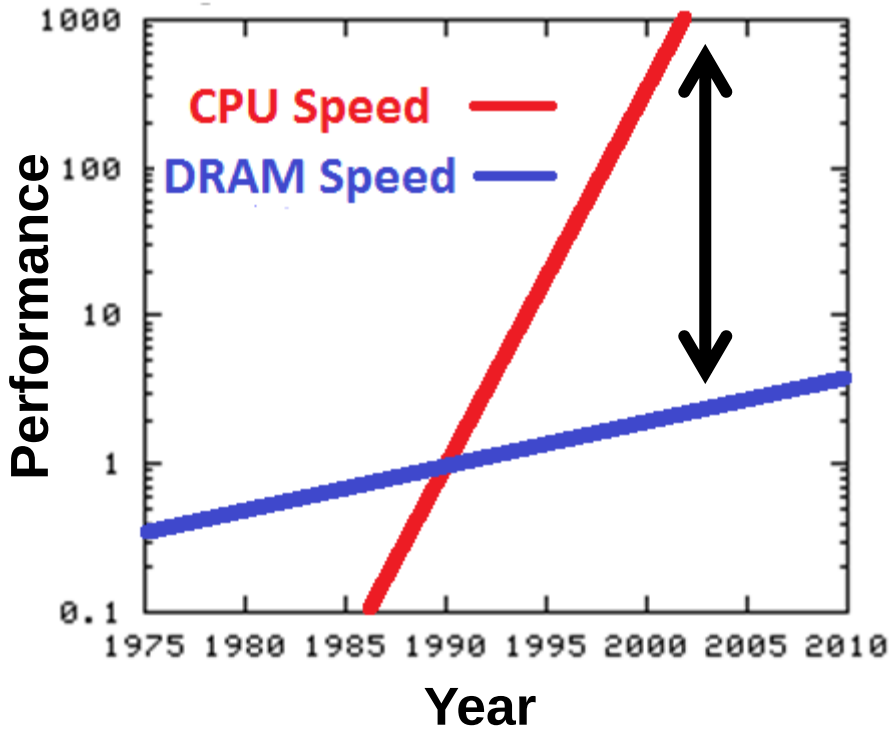


Beyond a Single Chip

Beyond a Single Chip: The External Memory Wall

von Neumann (Architecture) Bottleneck





A bottleneck of both throughput and power!

Math is Cheap, Data Movement is Expensive!

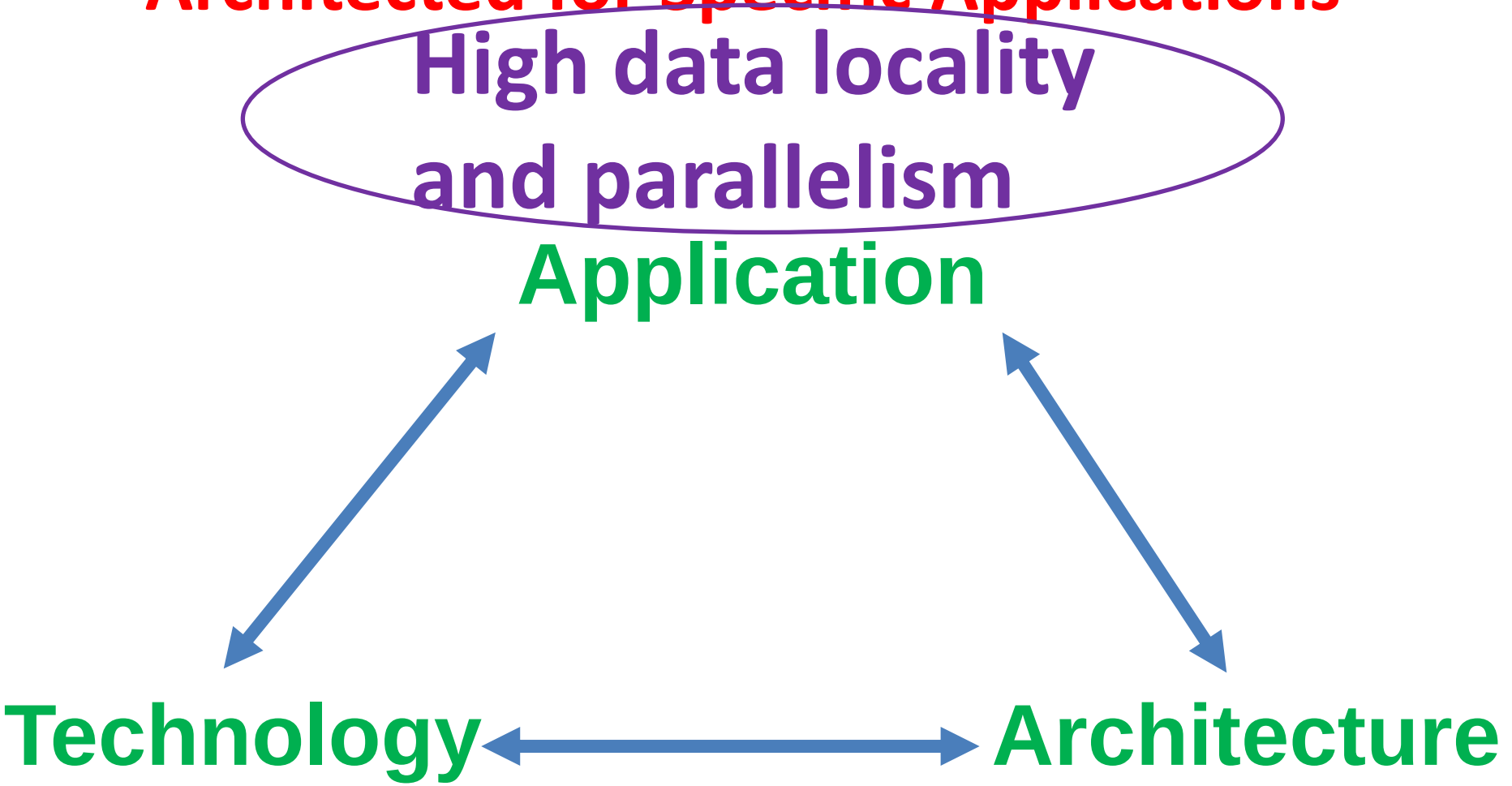
Operation (8-bit operand)	Energy/Op (45 nm)	Cost (vs. ALU)
ALU operation	0.05 pJ	1X
Move 10 nm on-chip	2.4 pJ	50X
Load from on-chip SRAM	2.5 pJ	50X
Send to off-chip DRAM	**320 pJ**	**6,400X**

Bit-Transportation energy is larger than computation energy!!!

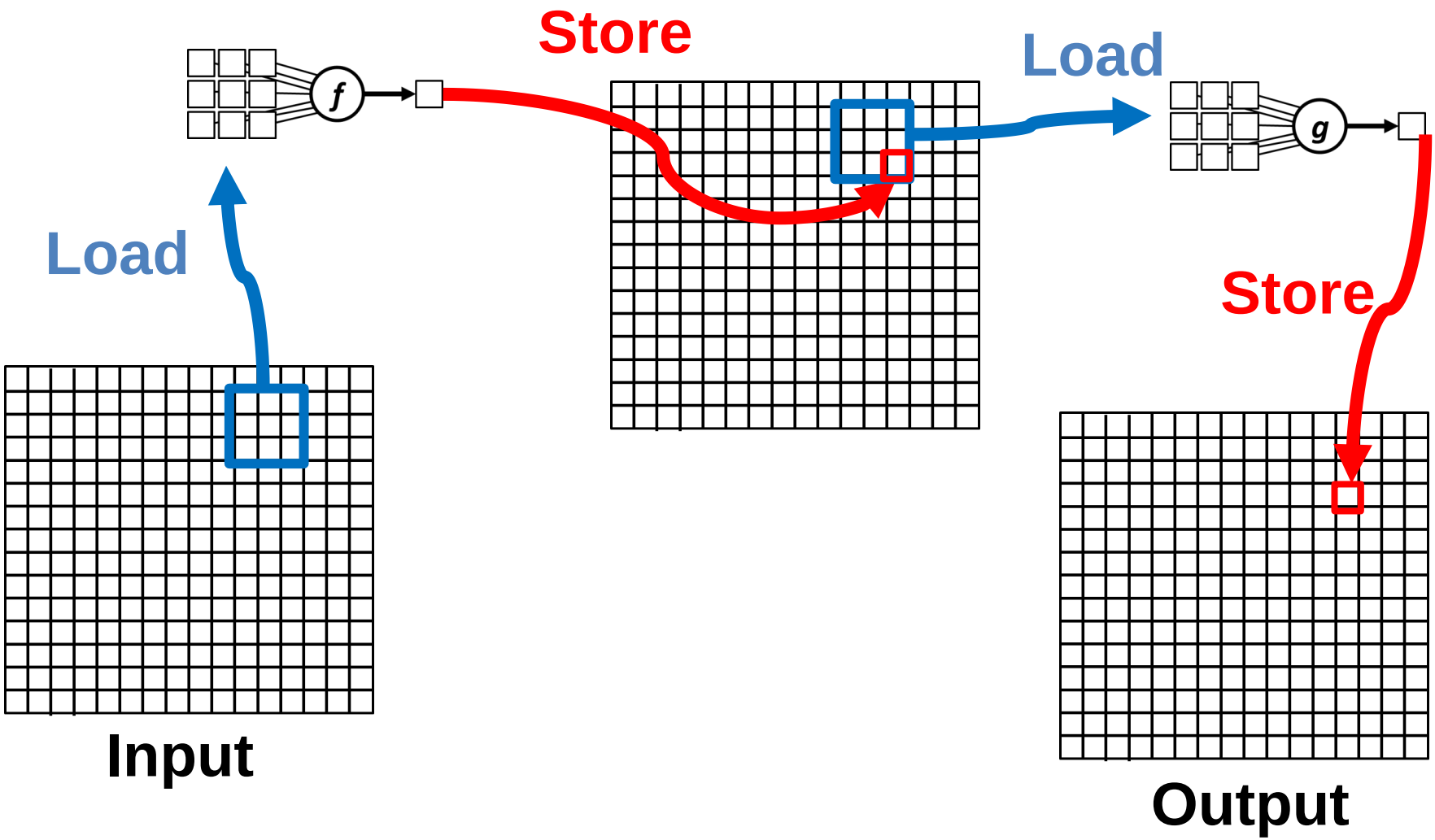
* Data from J. Brunhaver, W. Dally, M. Horowitz, S. Kvatinsky, Stanford University

Why accelerators are efficient

Architected for Specific Applications

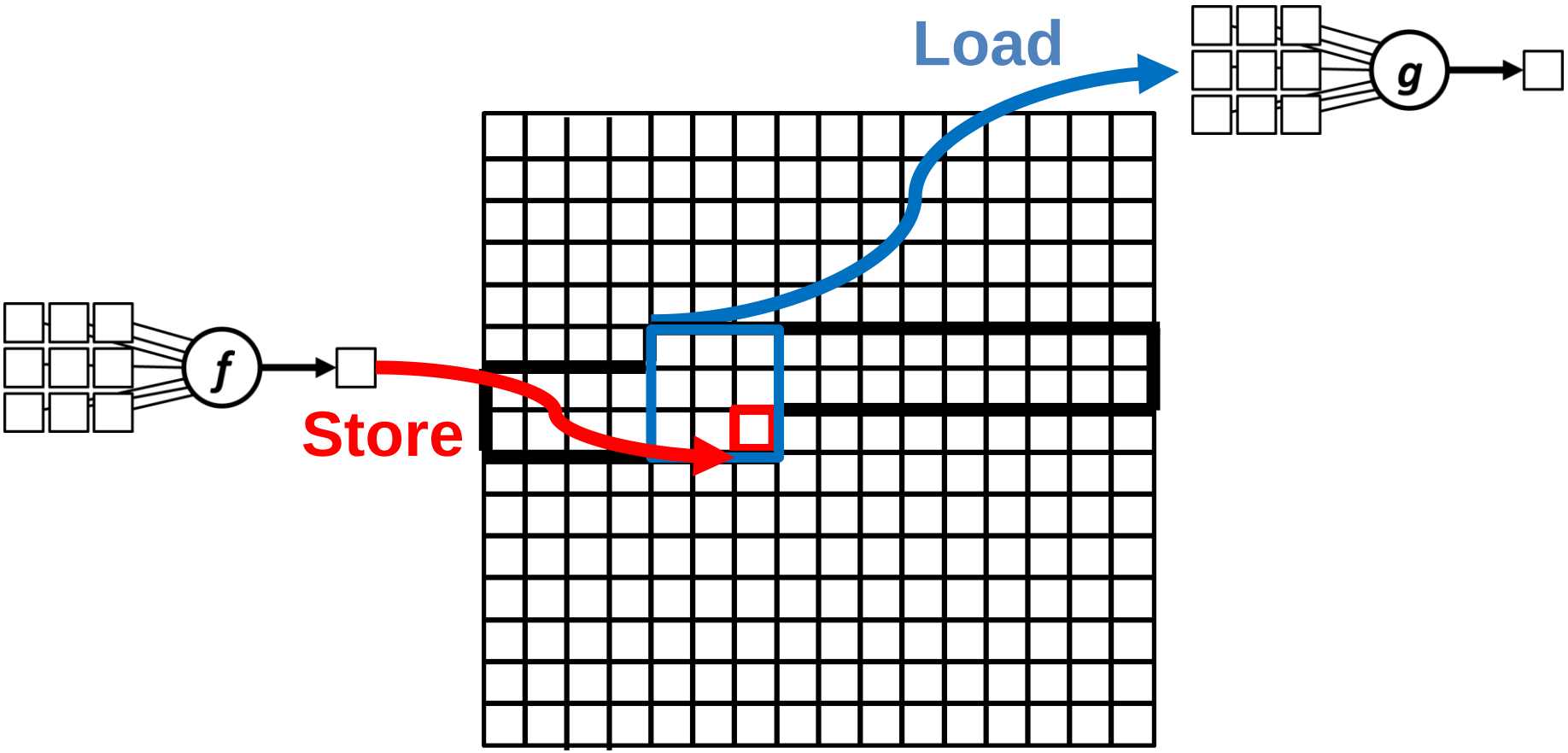


Example: Camera Signal Processing Pipeline



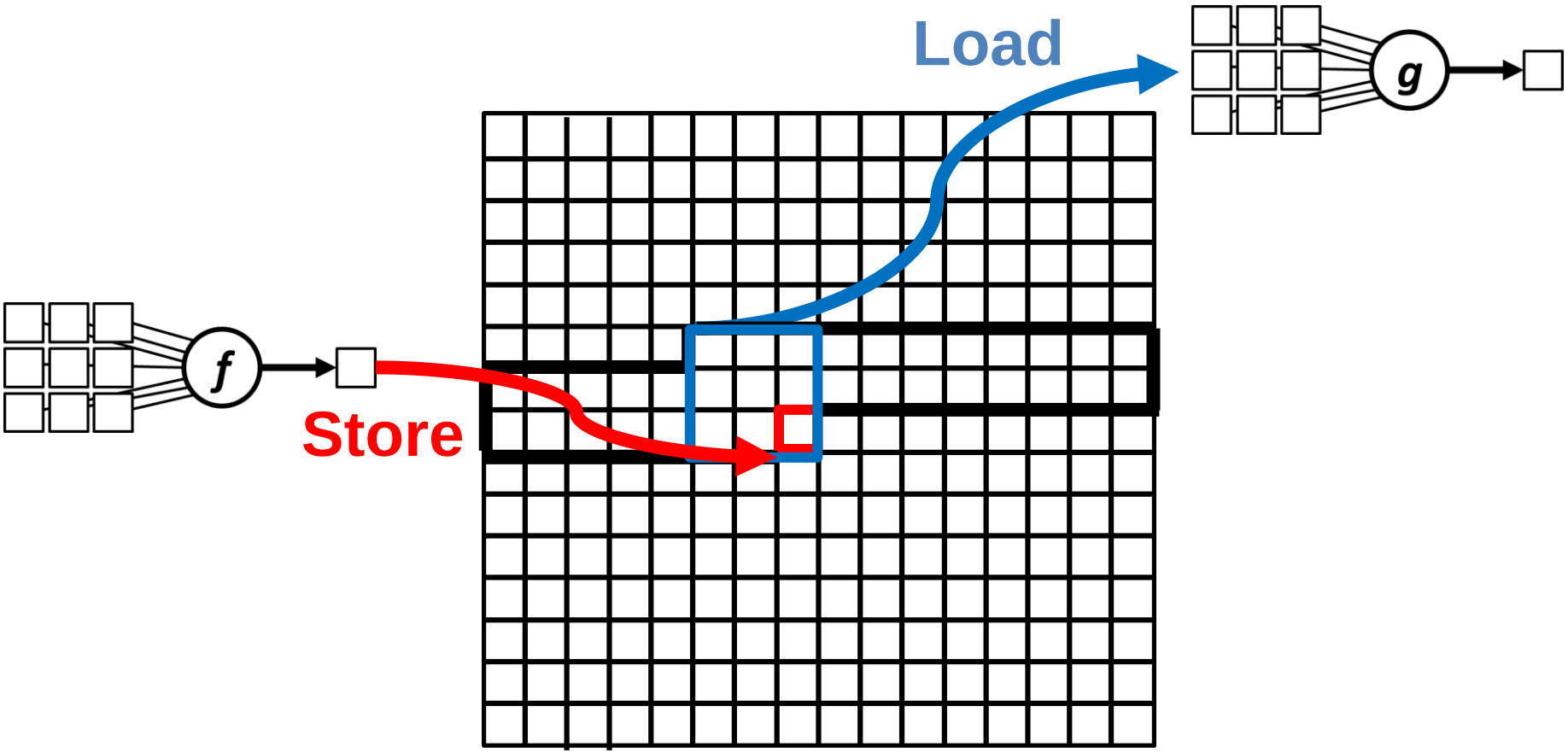
* Slide from S. Kvatinsky, Stanford University

Hold Intermediate in Line Buffers



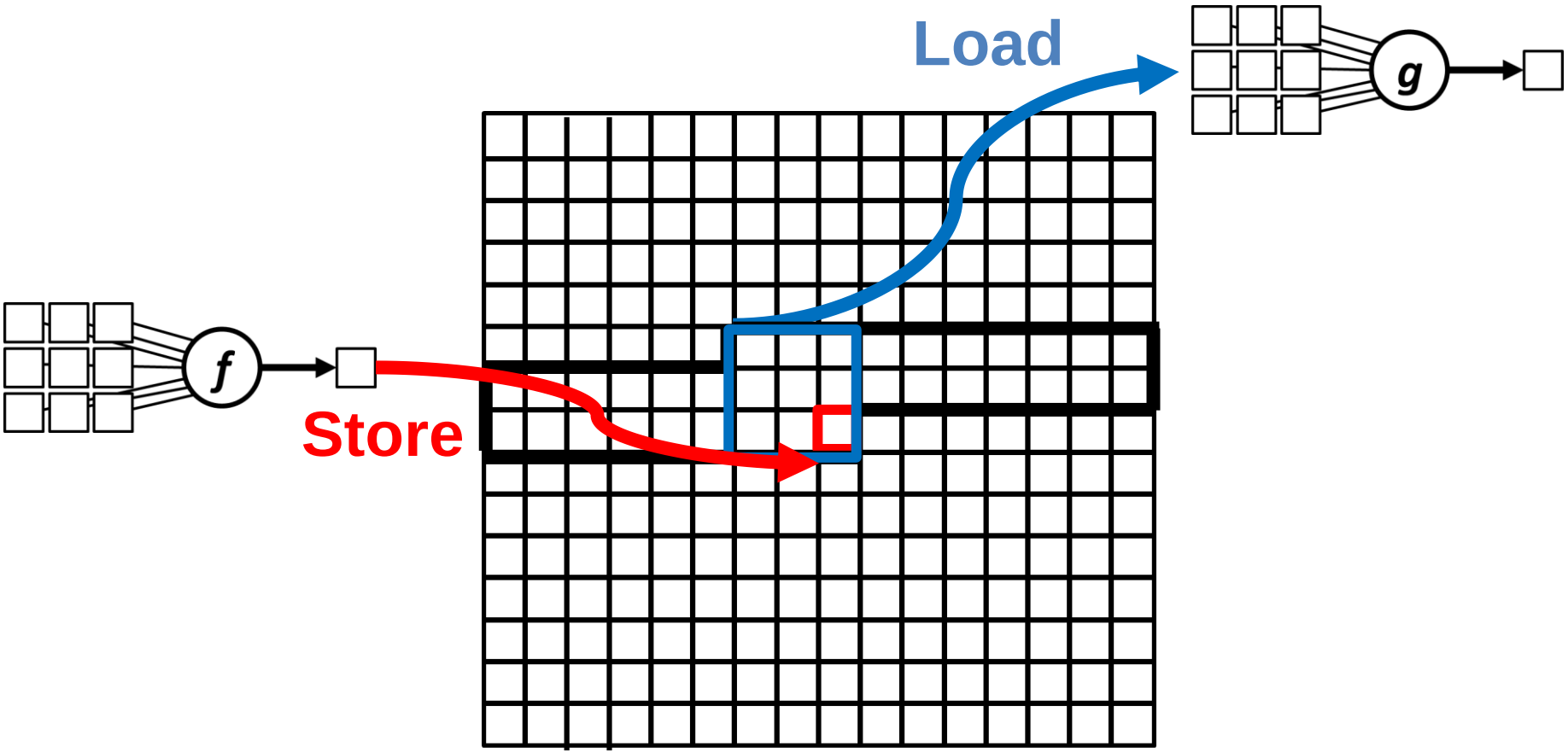
* Slide from S. Kvatinsky, Stanford University

Hold Intermediate in Line Buffers



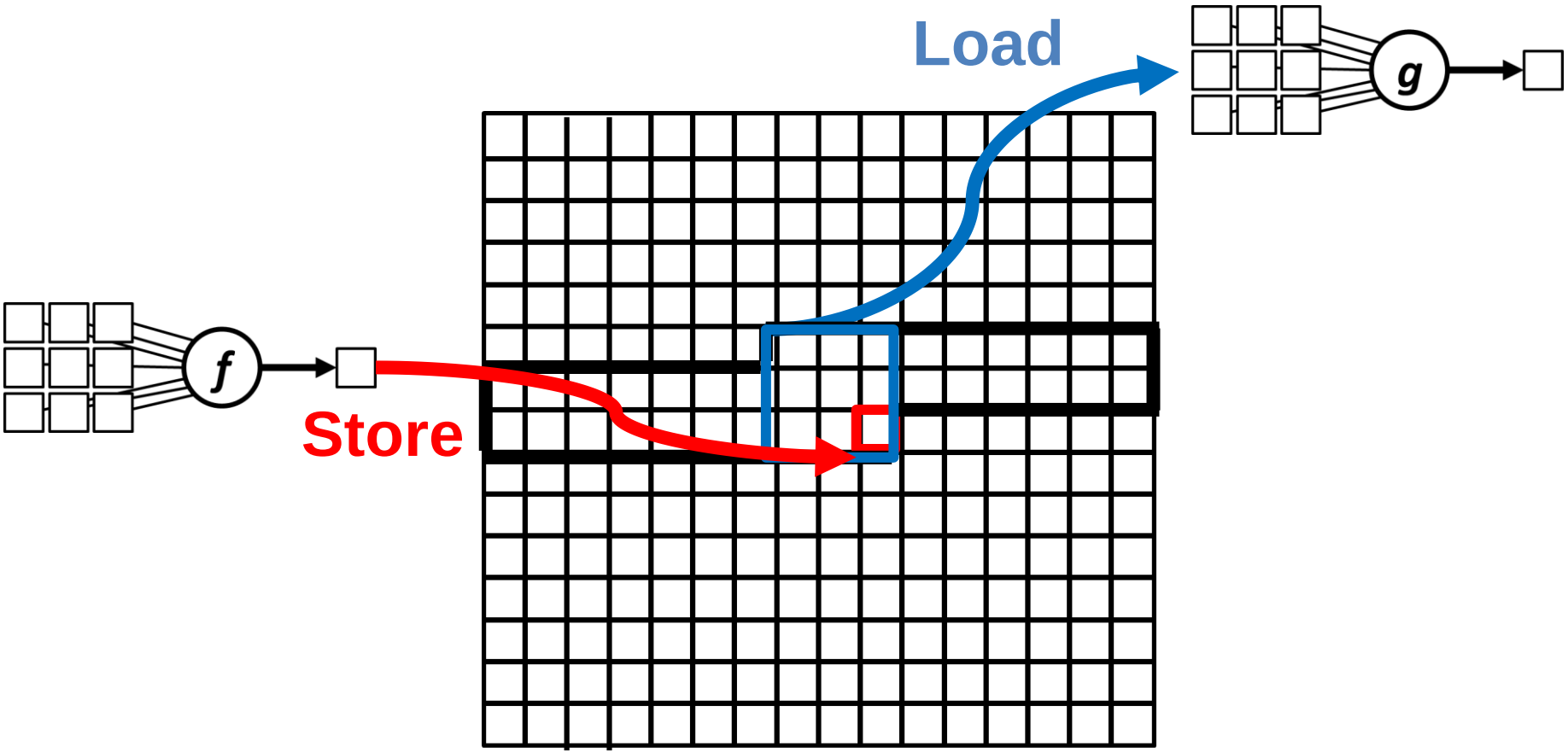
* Slide from S. Kvatinsky, Stanford University

Hold Intermediate in Line Buffers



* Slide from S. Kvatinsky, Stanford University

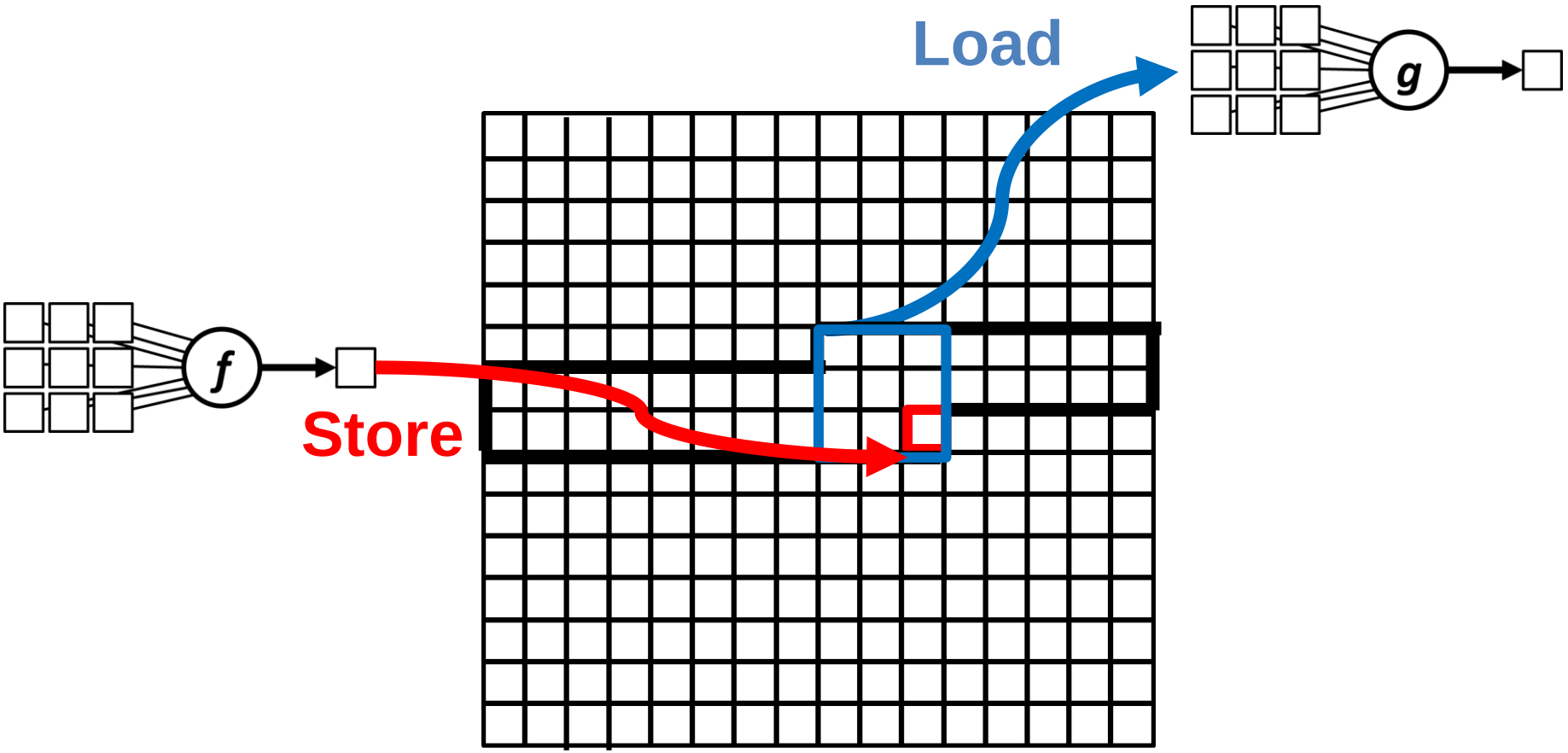
Hold Intermediate in Line Buffers



Line buffers are SRAM

* Slide from S. Kvatinsky, Stanford University

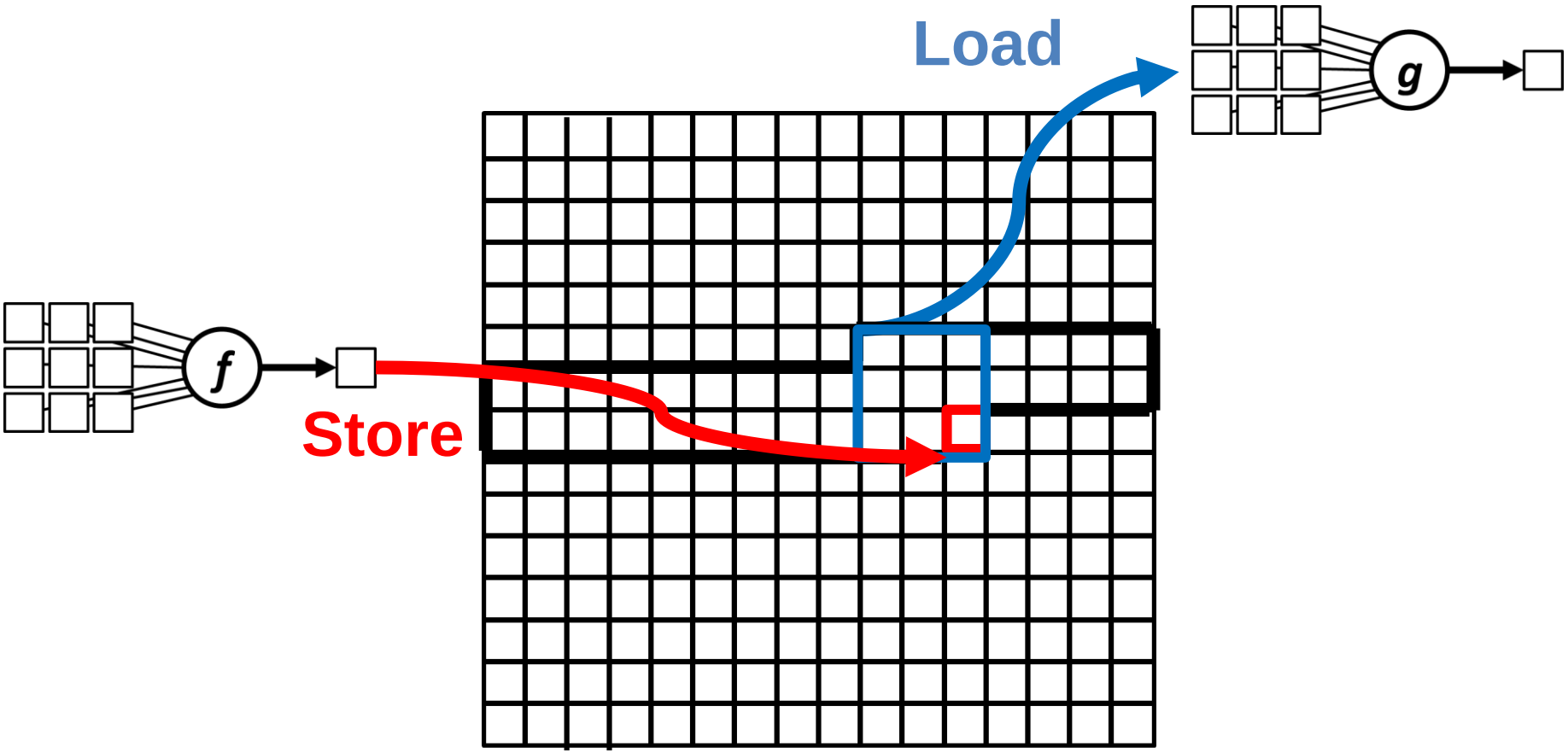
Hold Intermediate in Line Buffers



Line buffers are sliding windows

* Slide from S. Kvatinsky, Stanford University

Hold Intermediate in Line Buffers

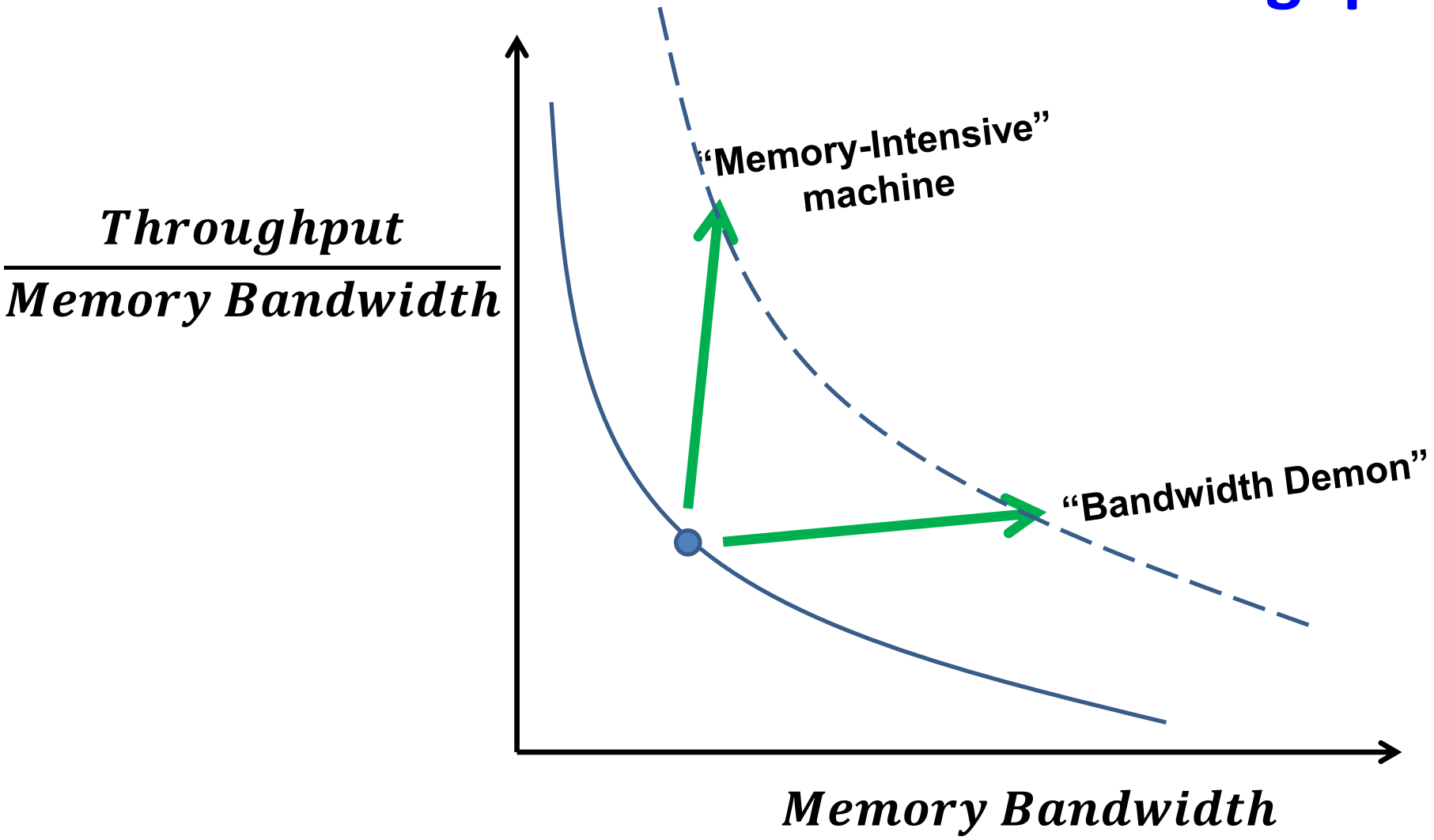


Line buffers are small and constant in size

* Slide from S. Kvatinsky, Stanford University

- **In Image processing , it is easy to exploit locality and parallelism.**
- **Can we do something similar for general-purpose programs?**

Let's look at a curve of Constant Throughput

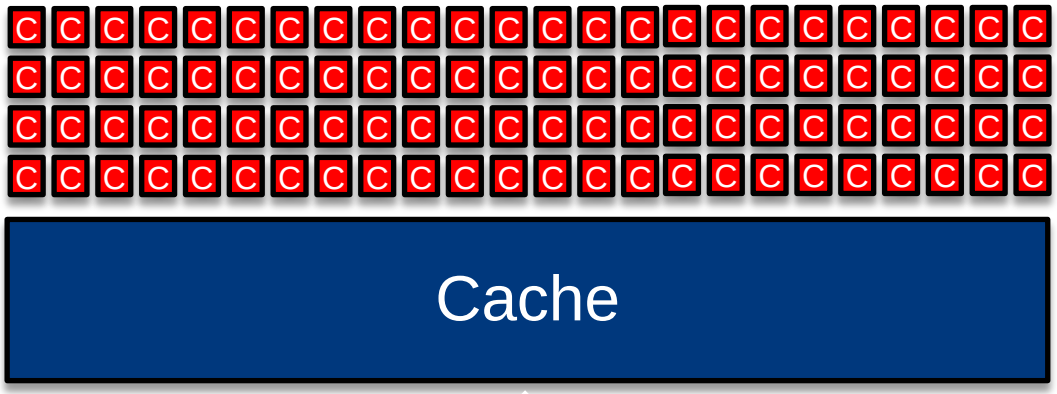


- Large bandwidth utilization ➔ Large power
- Increase on-die-memory
  (e.g. innovative cache? new ideas: Memory Intensive Architectures)

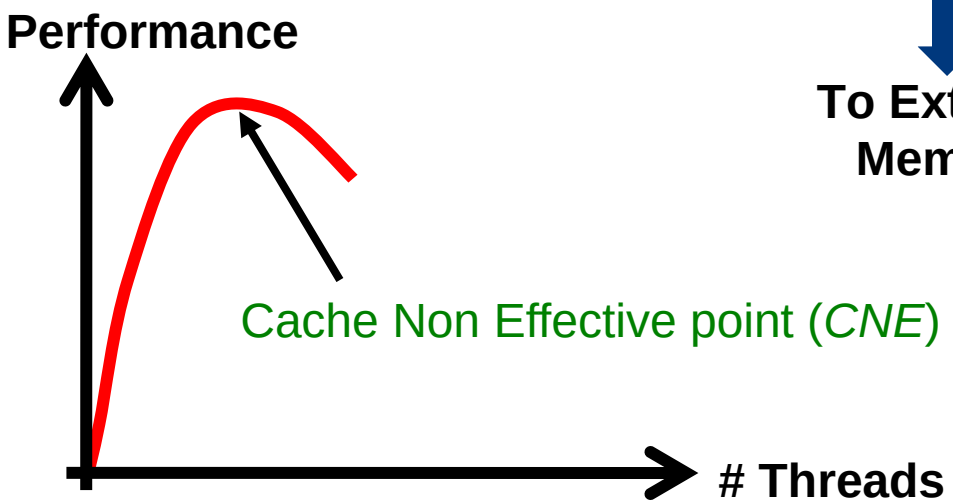
- **In the past, memory was added primarily for enhancing performance**
- **Now, we need memory-intensive systems for energy-efficiency**

Cache Machines

- Many cores behind a shared cache

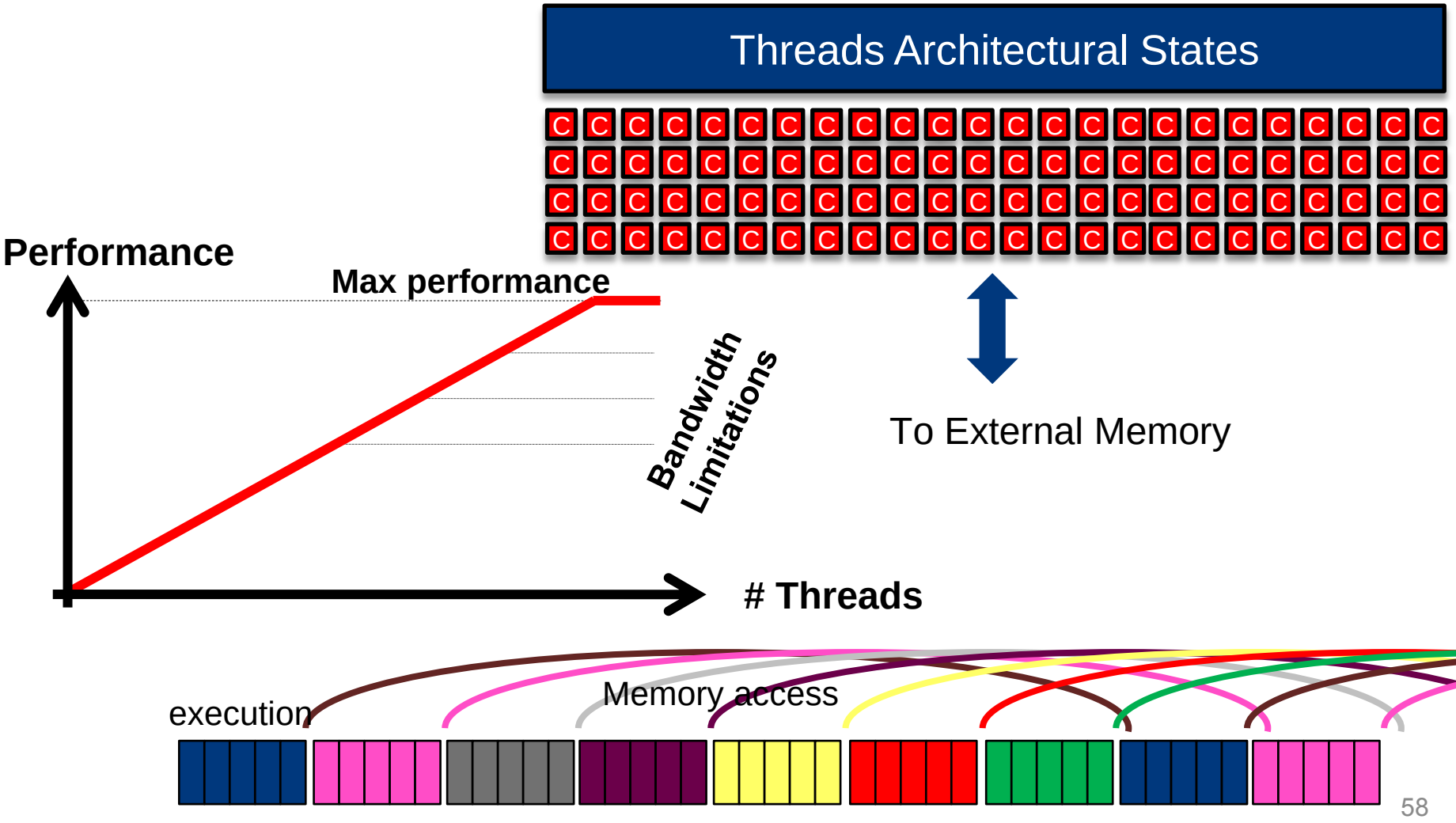


To External Memory



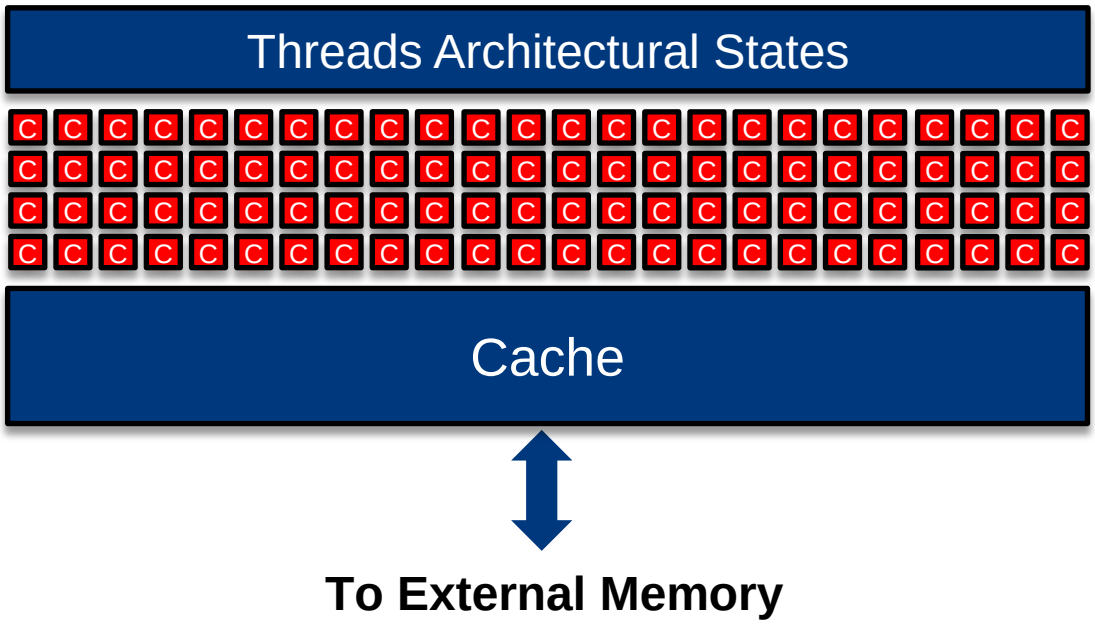
Multi-Thread Machines (e.g. GPGPU)

- Memory latency shielded by multiple thread execution



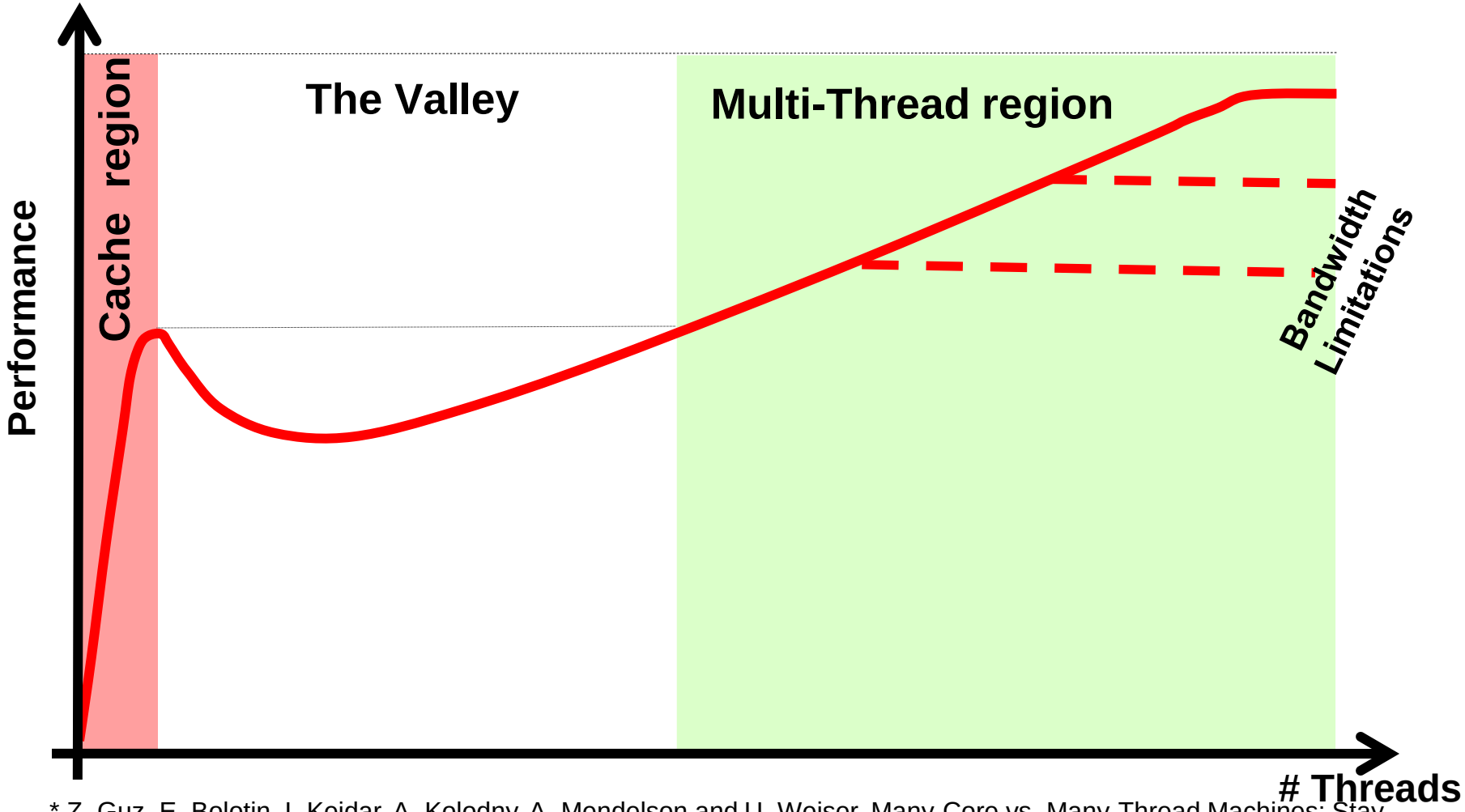
A Unified Machine Model

- Use both cache and many threads to shield memory access
  - The uniform framework renders the comparison meaningful
  - We derive simple, parameterized equations for performance, power, BW,..



Unified Machine Performance

- 3 regions: Cache efficiency region, The Valley, MT efficiency region



* Z. Guz, E. Bolotin, I. Keidar, A. Kolodny, A. Mendelson and U. Weiser, Many-Core vs. Many-Thread Machines: Stay Away From the Valley", IEEE Computer Architecture Letters, Volume 8, Issue 1, Jan. 2009

Wasteful Effects of Shared Resources (⬆ Energy, ⬇ Throughput)

- Threads might disturb each other as they need a shared cache, memory bus, network bandwidth, disk, ….

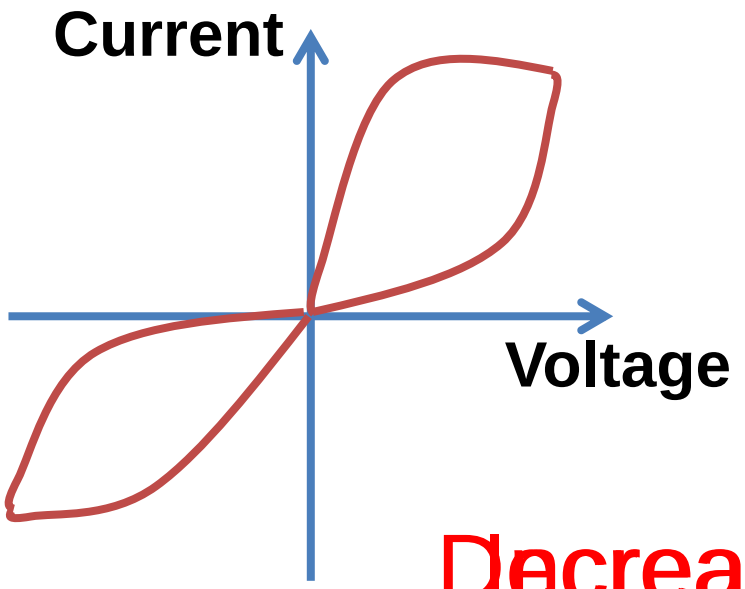
- **Destructive Interference:** Causing unnecessary work
  Example: threads pollute each others' cache

- **Collisions:** Requesting the same resource
  The waiting cores waste energy.

New Research Direction: New Memory-Intensive Architectures

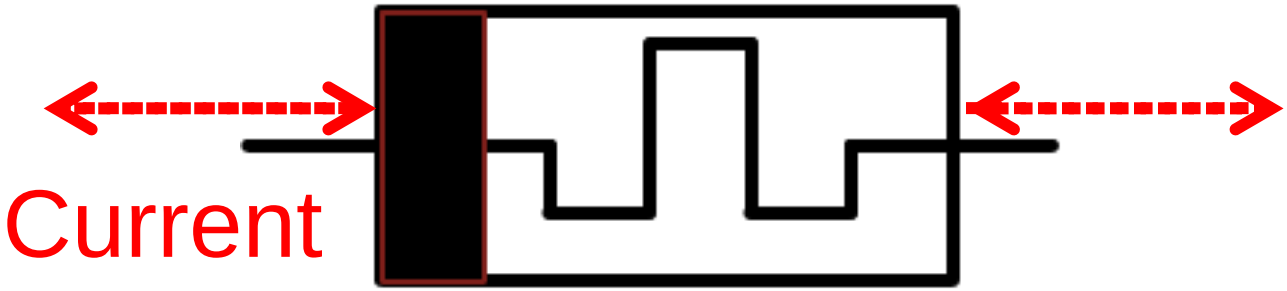
- Embed memory in execution units?
- Add computational capabilities within memory?

New memory devices: Memristors



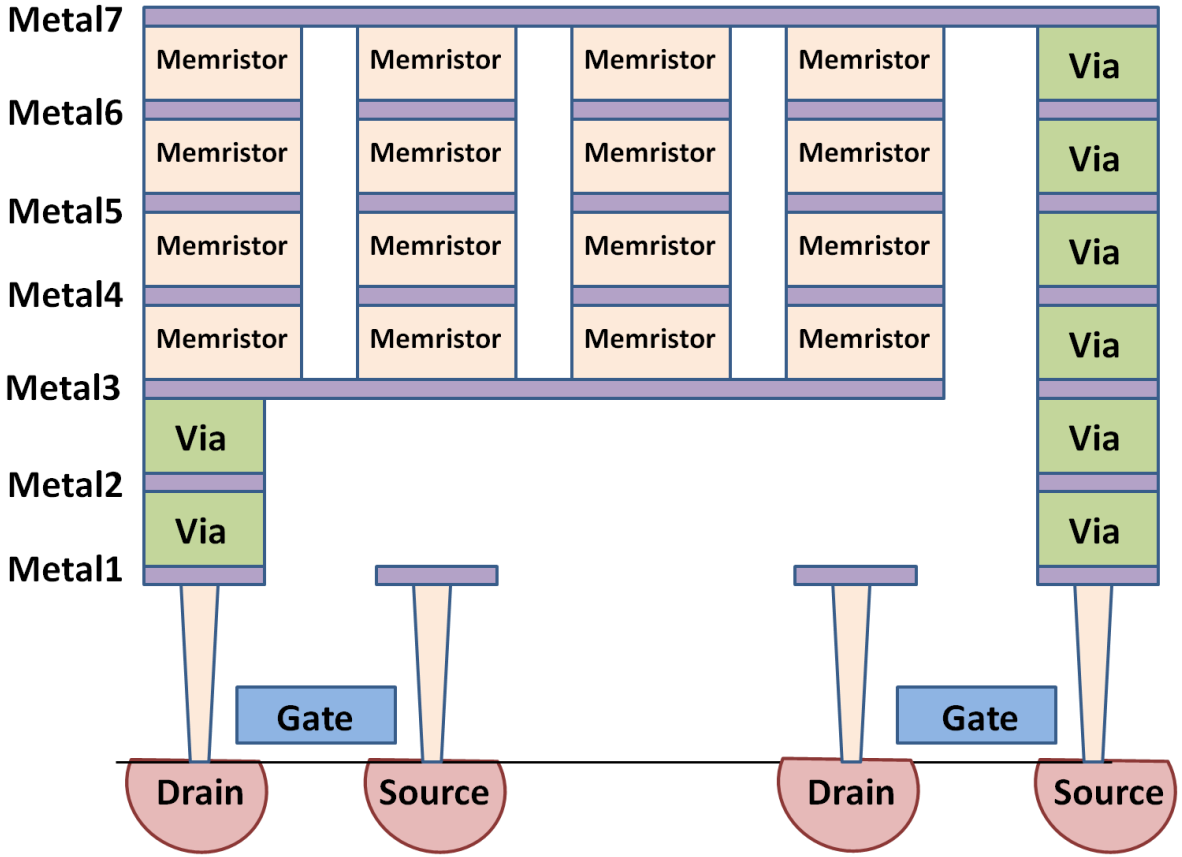
"0" = High Resistance
"1" = Low Resistance

Decrease resistance



Memristors Enable New Structures

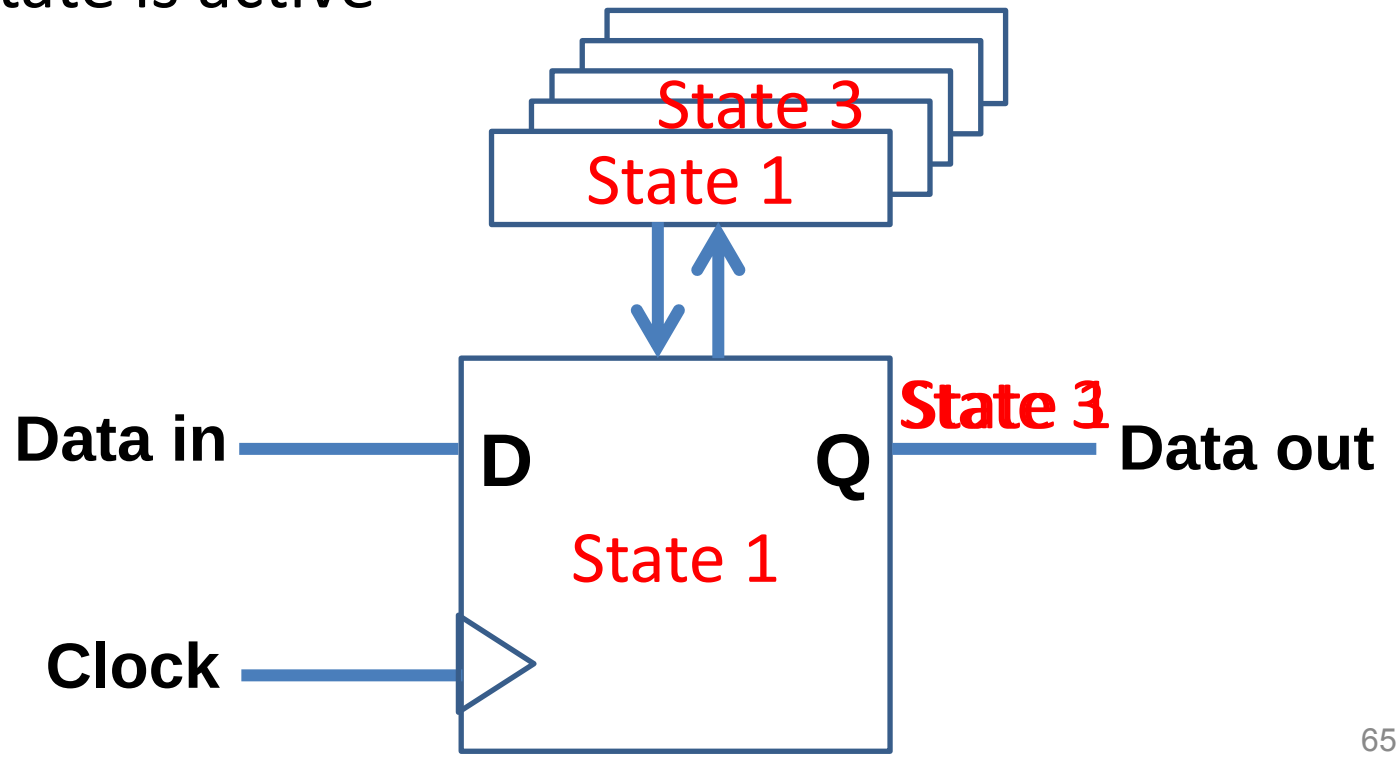
- Memory on-top of logic – high density
- Store data locally above circuit which needs it



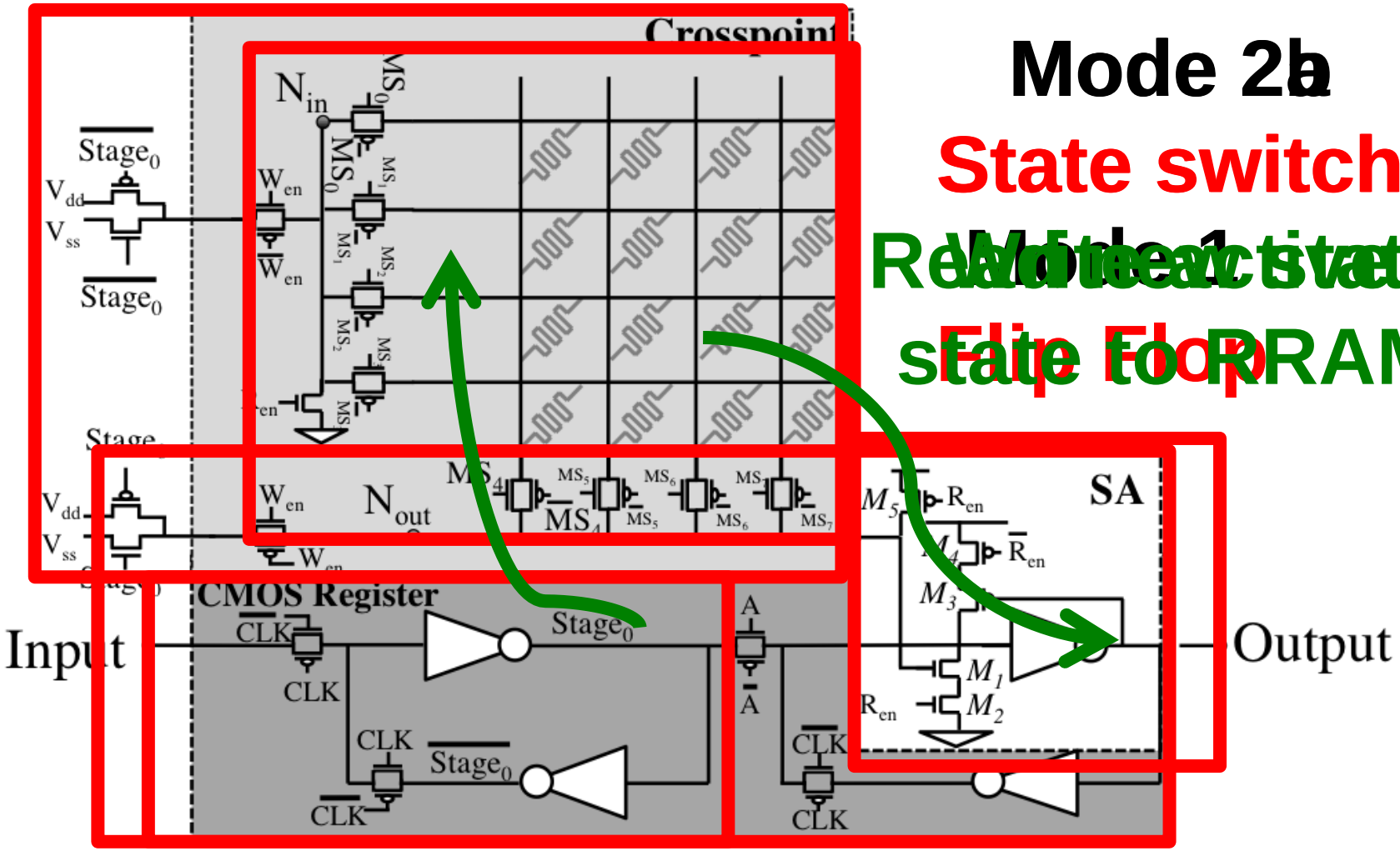
Example of Embedding Memory in a Processor Pipeline: Multistate Register Concept

- CMOS register – a single state
- Multistate register – multiple states,

  a single state is active



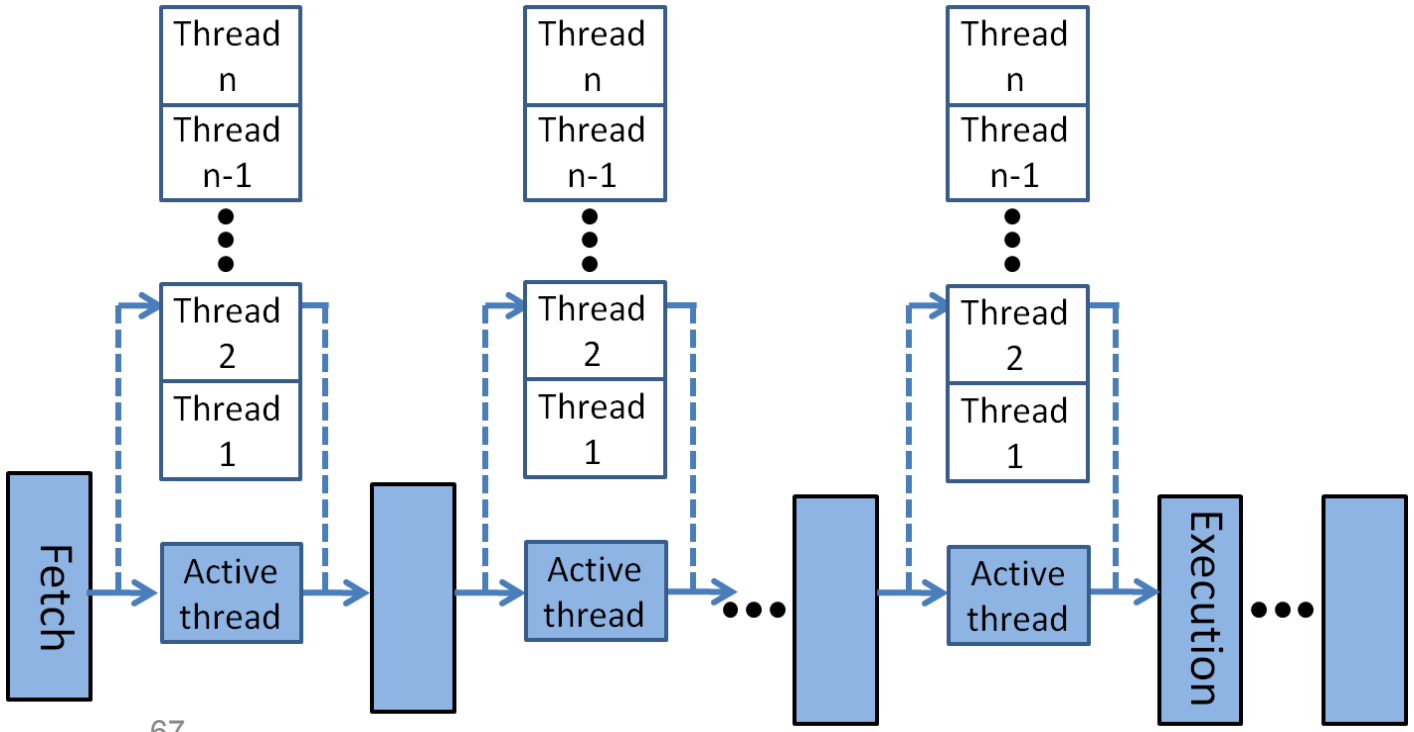
Operation of Multistate Register



R. Patel, S. Kvatinsky, E. G. Friedman, and A. Kolodny, "Multistate Register Based on Resistive RAM," IEEE Transactions on Very Large Scale Integration (VLSI)

Continuous Flow Multithreading

- Using multistate registers
- Enhanced performance to energy ratio
  (44% improvement over SoE, 310% over SMT)

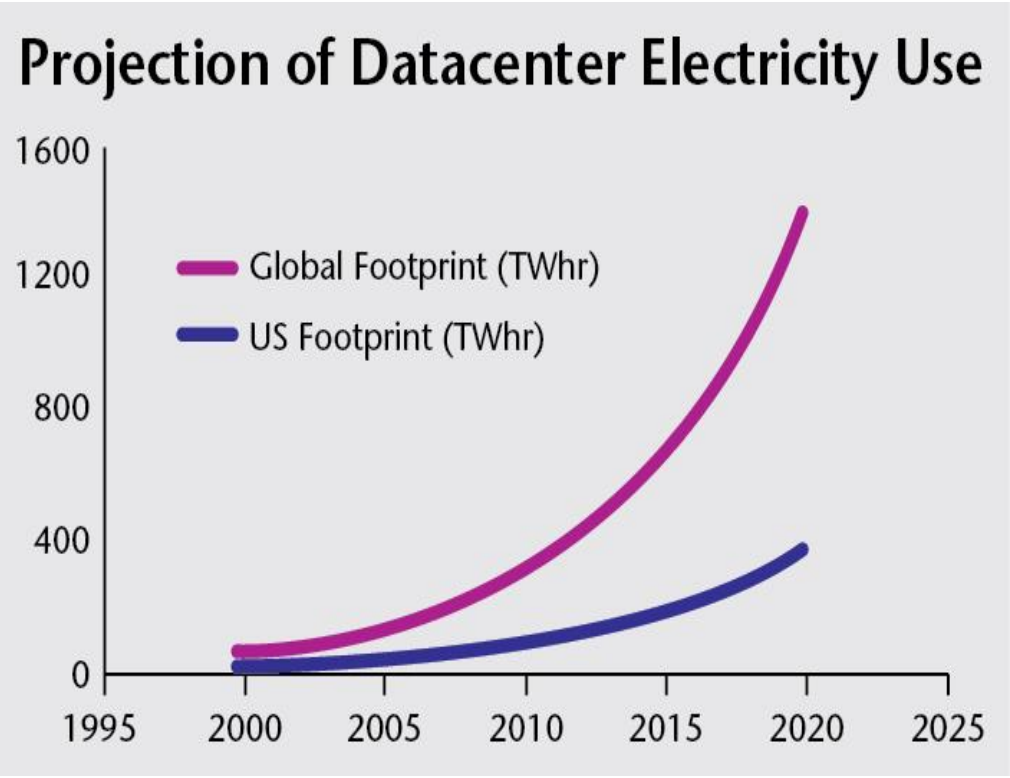


S. Kvatinsky, Y. Nacson, Y. Etsion, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based Multithreading," IEEE Computer Architecture Letters, 2013
S. Kvatinsky, Y. Nacson, R. Patel, Y. Etsion, E. G. Friedman, A. Kolodny, and U. C. Weiser, "On the In-Die 3D Integration of Memory in CMOS Metal Layers and Its Implications on Processor Microarchitecture," submitted to ISCA 2015

From Small Systems to Data Centers

Data Centers

- Size: 1 million m^2
- Power: 100 MegaWatts (only ~30% of it for computing!)
- Total power cost in U.S. 2014 - $10B[1]



(1) http://www.computerworld.com/article/2598562/data-center/data-centers-are-the-new-polluters.html

Datacenter Power Consumed by IT Equipment

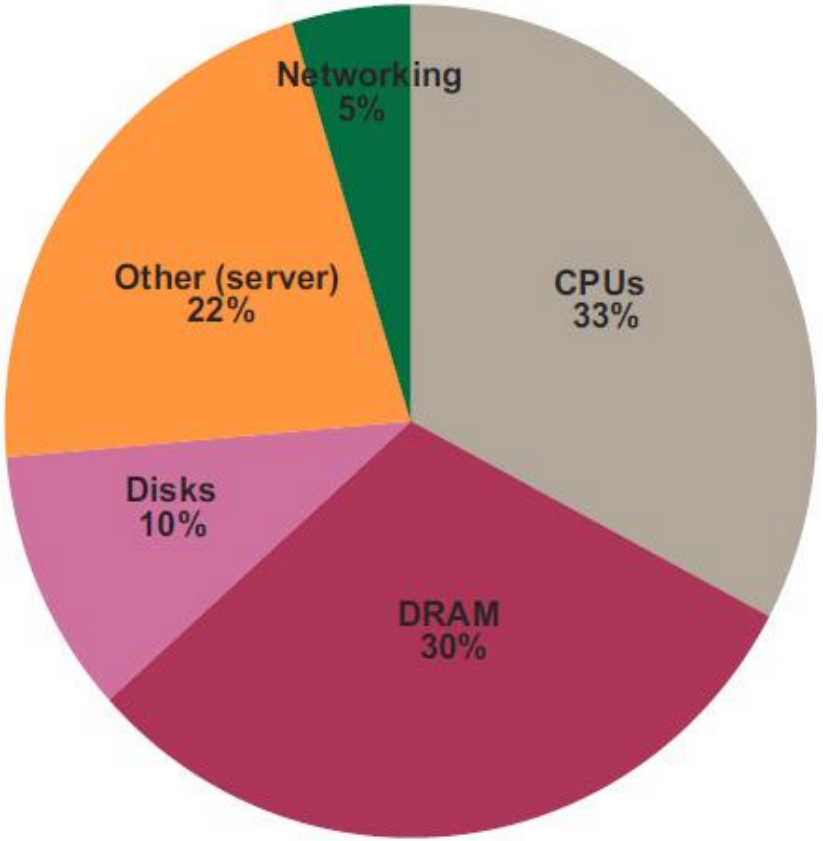
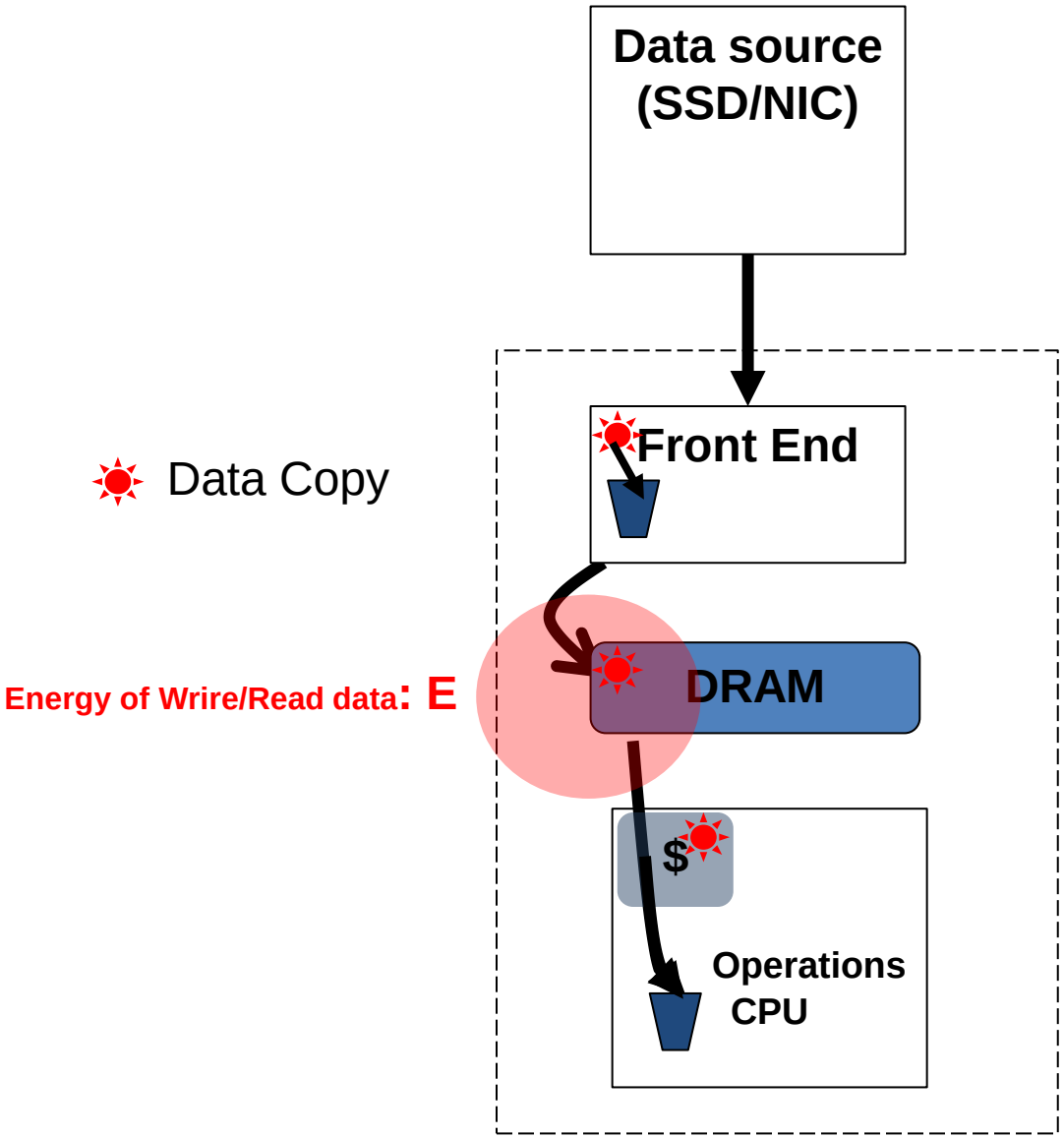


FIGURE 1.4: Approximate distribution of peak power usage by hardware subsystem in one of Google's datacenters (circa 2007).

Lious Borrosso talk http://www.cs.berkeley.edu/~rxin/db-papers/WarehouseScaleComputing.pdf

Let's Use the Same Principles to Save Power in Data Centers!

Example: Avoid Unnecessary DRAM usage?



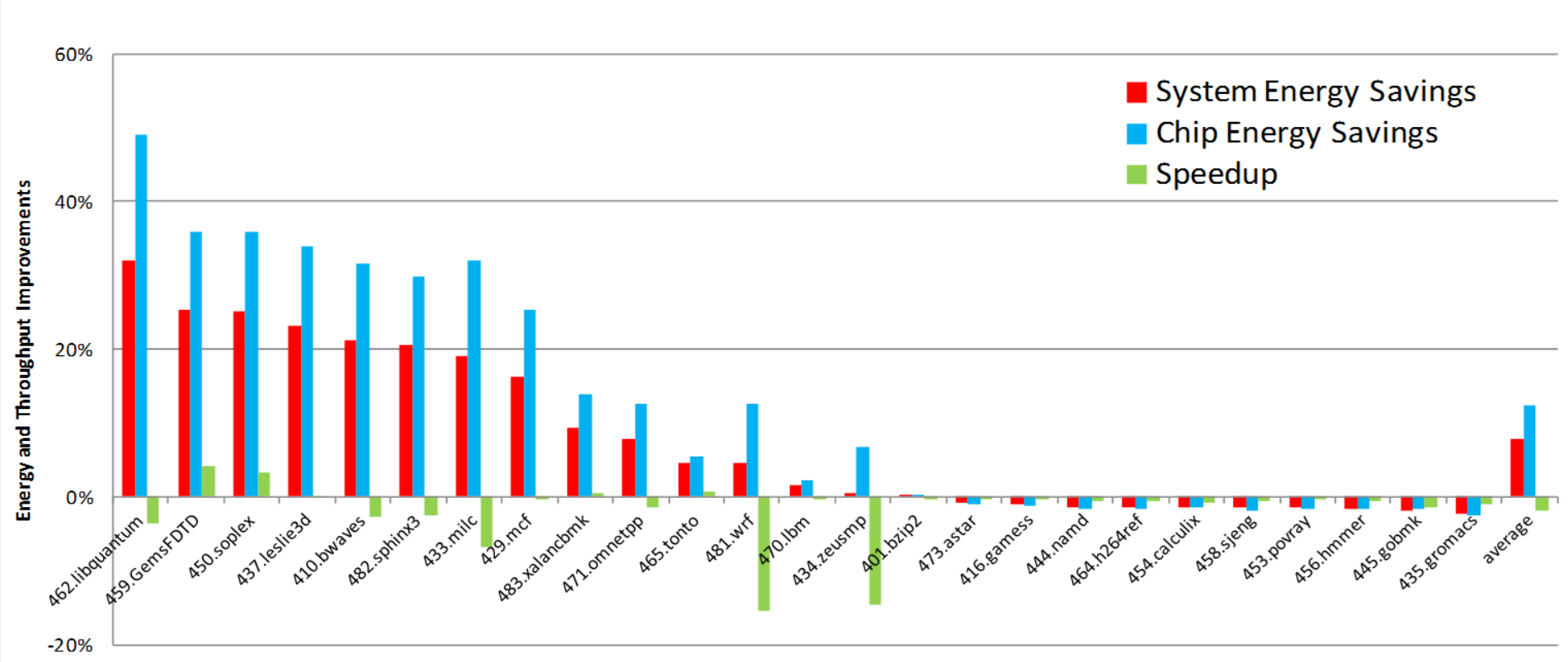
Cache/Memory are not effective if:

Memory related:
Reuse distance: >1G access

Cache related:
Reuse distance: >1M access

* T. Morad et al., unpublished

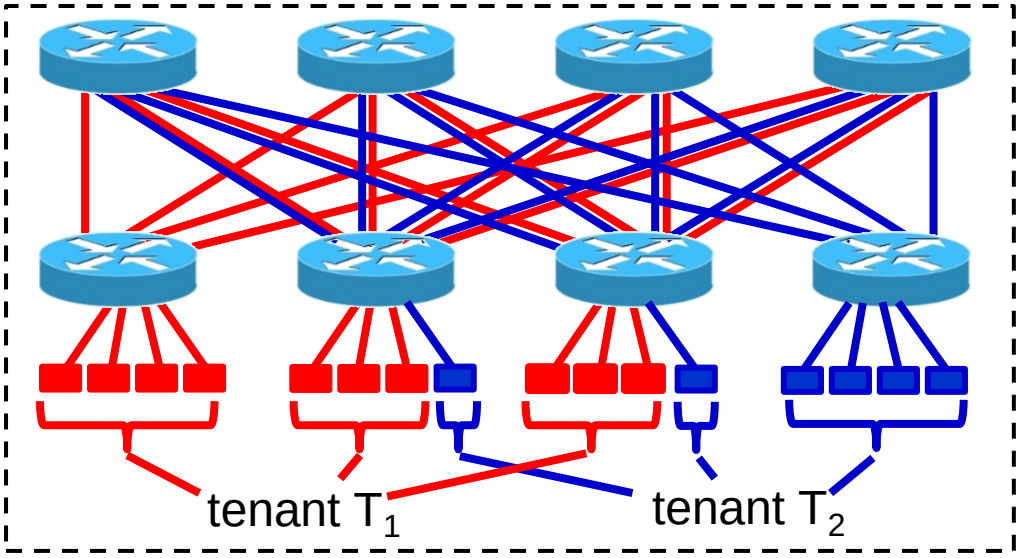
Example: O/S Scheduling of Threads for Energy Efficiency



* T. Morad et al., unpublished

Example: Avoid Interference on Links in Data-Center Network?

Although each tenant gets his own servers, tenant 1 performance depends on Tenant 2 traffic



Tenants 1 and 2 sharing many links

Possible consequences:

- Performance degradation >50%

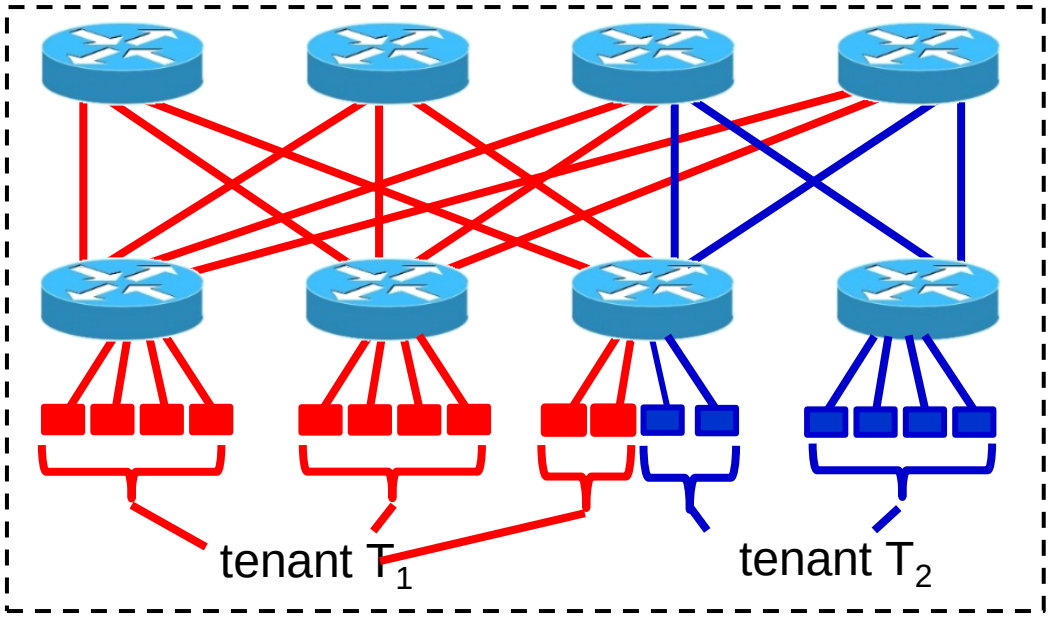
- Unpredictable performance or response time

* E. Zahavi et al., unpublished

Solution Example:

Links as a Service, to Isolate Tenants in the Cloud

- Dedicating Links to Tenants – No Shared Links!
- Guarantee the resulting tenant network:
  - Support any admissible traffic pattern
  - Rearrangeable non-blocking routing



* E. Zahavi et al., unpublished

Summary

- Many opportunities to **improve energy efficiency:**
  - **Energy-efficient architectures**
  - **Power-aware system management**

Thanks

This talk is based on joint work with many students and collaborators:

Eitan Zahavi

Evgeny Bolotin

Nir Magen

Oved Itzhak

Shahar Kvatinsky

Tomer Morad

Yaniv Ben-Itzhak

Yoni Aizik

Yuval Nacson

Zigi Walter

Zvika Guz

Avi Mendelson

Eby Friedman

Idit Keidar

Isaac Keslassy

Israel Cidon

Uri Weiser

Yoav Etzion