The Challenge of Many Cores

Erik Hagersten
Uppsala University

UPMARC

UPPSALA UNIVERSITET

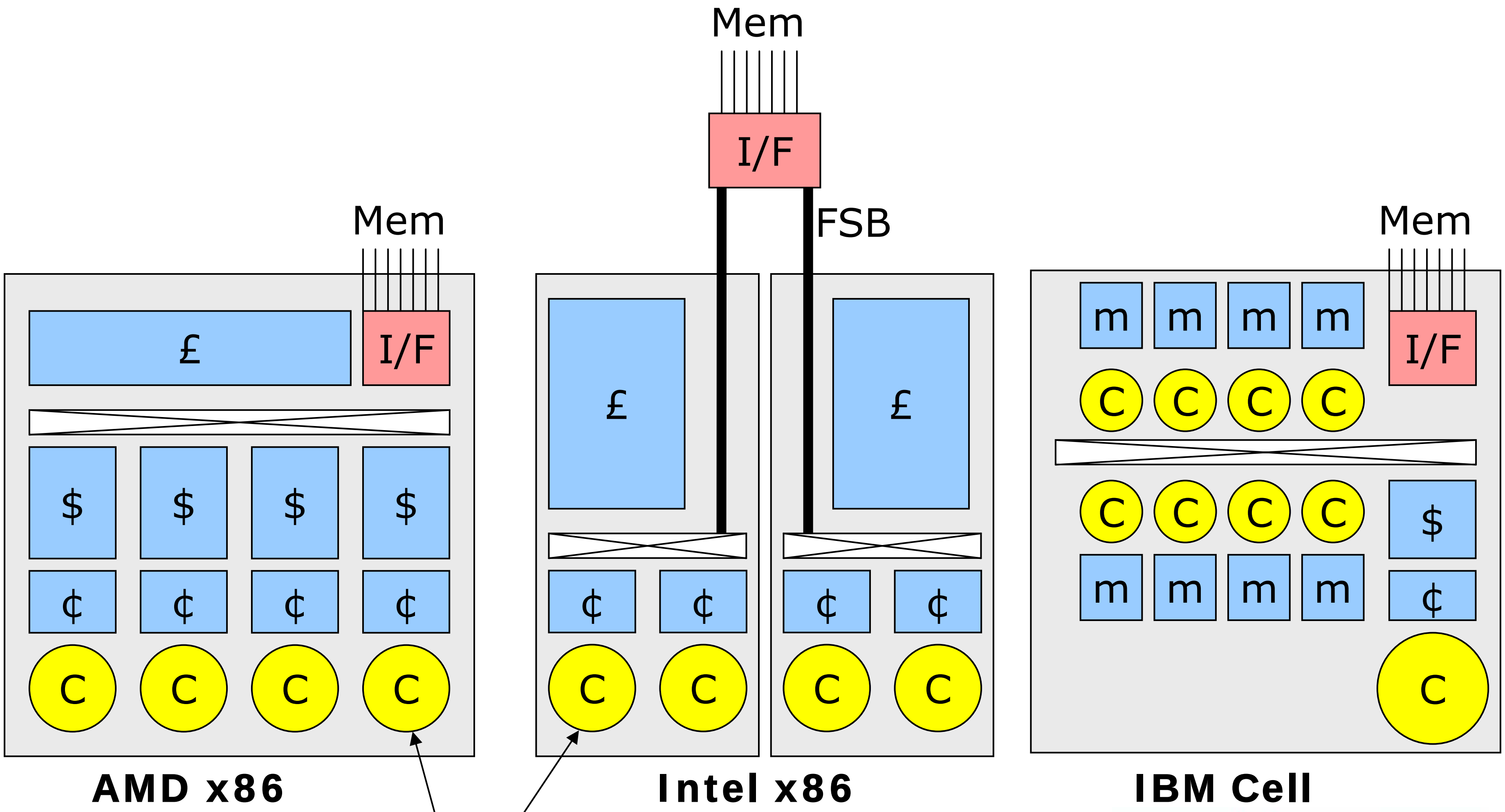
Multi-core CPUs:

- Ageia PhysX, a multi-core physics processing unit.
- Ambric Am2045, a 336-core Massively Parallel Processor Array (MPPA)
- **AMD**
  - Athlon 64, Athlon 64 FX and Athlon 64 X2 family, dual-core desktop processors.
  - Opteron, dual- and quad-core server/workstation processors.
  - Phenom, triple- and quad-core desktop processors.
  - Sempron X2, dual-core entry level processors.
  - Turion 64 X2, dual-core laptop processors.
  - Radeon and FireStream multi-core GPU/GPGPU (10 cores, 16 5-issue wide superscalar stream processors per core)
- ARM MPCore is a fully synthesizable multicore container for ARM9 and ARM11 processor cores, intended for high-performance embedded and entertainment applications.
- Azul Systems Vega 2, a 48-core processor.
- Broadcom SiByte SB1250, SB1255 and SB1455.
- Cradle Technologies CT3400 and CT3600, both multi-core DSPs.
- Cavium Networks Octeon, a 16-core MIPS MPU.
- HP PA-8800 and PA-8900, dual core PA-RISC processors.
- **IBM**
  - POWER4, the world's first dual-core processor, released in 2001.
  - POWER5, a dual-core processor, released in 2004.
  - POWER6, a dual-core processor, released in 2007.
  - PowerPC 970MP, a dual-core processor, used in the Apple Power Mac G5.
  - Xenon, a triple-core, SMT-capable, PowerPC microprocessor used in the Microsoft Xbox 360 game console.
- IBM, Sony, and Toshiba Cell processor, a nine-core processor with one general purpose PowerPC core and eight specialized SPUs (Synergystic Processing Unit) optimized for vector operations used in the Sony PlayStation 3.
- Infineon Danube, a dual-core, MIPS-based, home gateway processor.
- **Intel**
    - Celeron Dual Core, the first dual-core processor for the budget/entry-level market.
  - Core Duo, a dual-core processor.
  - Core 2 Duo, a dual-core processor.
  - Core 2 Quad, a quad-core processor.
  - Core i7, a quad-core processor, the successor of the Core 2 Duo and the Core 2 Quad.
  - Itanium 2, a dual-core processor.
  - Pentium D, a dual-core processor.
  - Teraflops Research Chip (Polaris), an 3.16 GHz, 80-core processor prototype, which the company says will be released within the next five years[6].
  - Xeon dual-, quad- and hexa-core processors.
- IntellaSys seaForth24, a 24-core processor.
- **Nvidia**
  - GeForce 9 multi-core GPU (8 cores, 16 scalar stream processors per core)
  - GeForce 200 multi-core GPU (10 cores, 24 scalar stream processors per core)
  - Tesla multi-core GPGPU (8 cores, 16 scalar stream processors per core)
- Parallax Propeller P8X32, an eight-core microcontroller.
- picoChip PC200 series 200-300 cores per device for DSP & wireless
- Rapport Kilocore KC256, a 257-core microcontroller with a PowerPC core and 256 8-bit "processing elements".
- Raza Microelectronics XLR, an eight-core MIPS MPU
- **Sun Microsystems**
  - UltraSPARC IV and UltraSPARC IV+, dual-core processors.
  - UltraSPARC T1, an eight-core, 32-thread processor.
  - UltraSPARC T2, an eight-core, 64-concurrent-threa
- Texas Instruments TMS320C80 MVP, a five-core multimedi
- Tilera TILE64, a 64-core processor
- XMOS Software Defined Silicon quad-core XS1-G4

[source: Wikipedia]

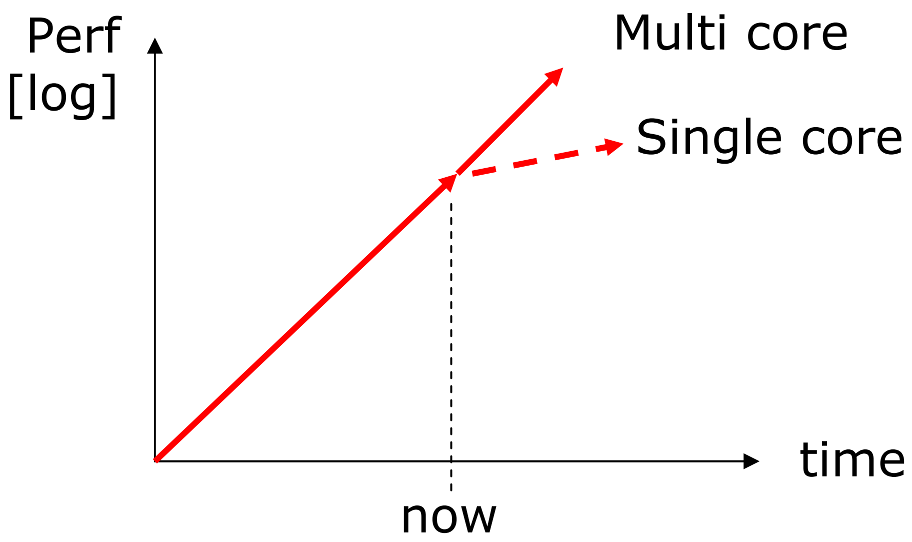
High-performance single-core CPUs:

Many shapes and forms…



May run more than one thread…

Everybody is doing is! But, why now?



- Not enough ILP to get return from using more transistors
- Power consumption $P_{dyn} \sim C \bullet f \bullet V^2$
- Signal propagation delay > transistor delay
- Memory stalls dominate

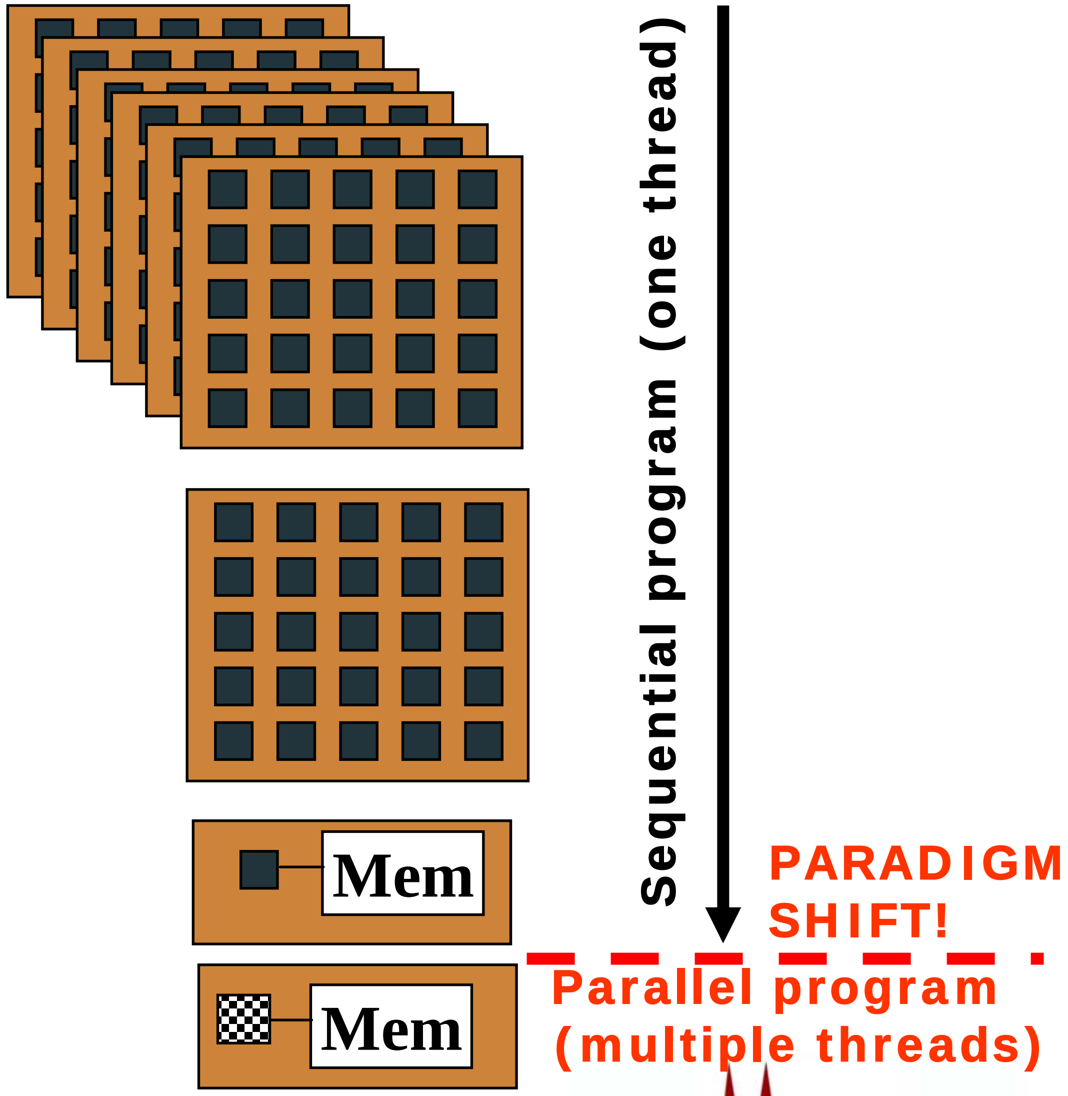
Darling, I shrunk the computer

Mainframes

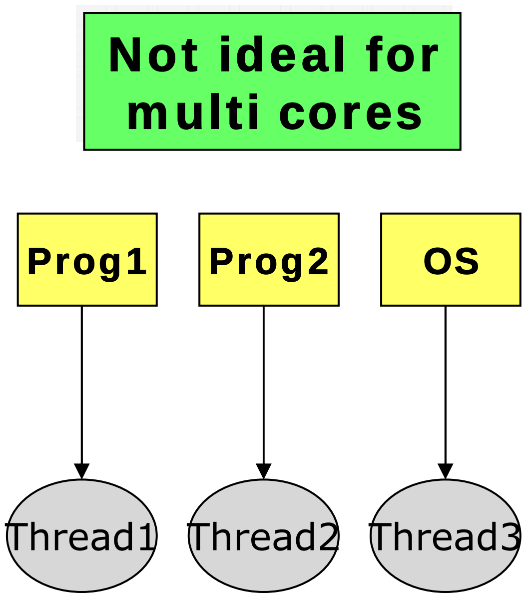
Super Minis:

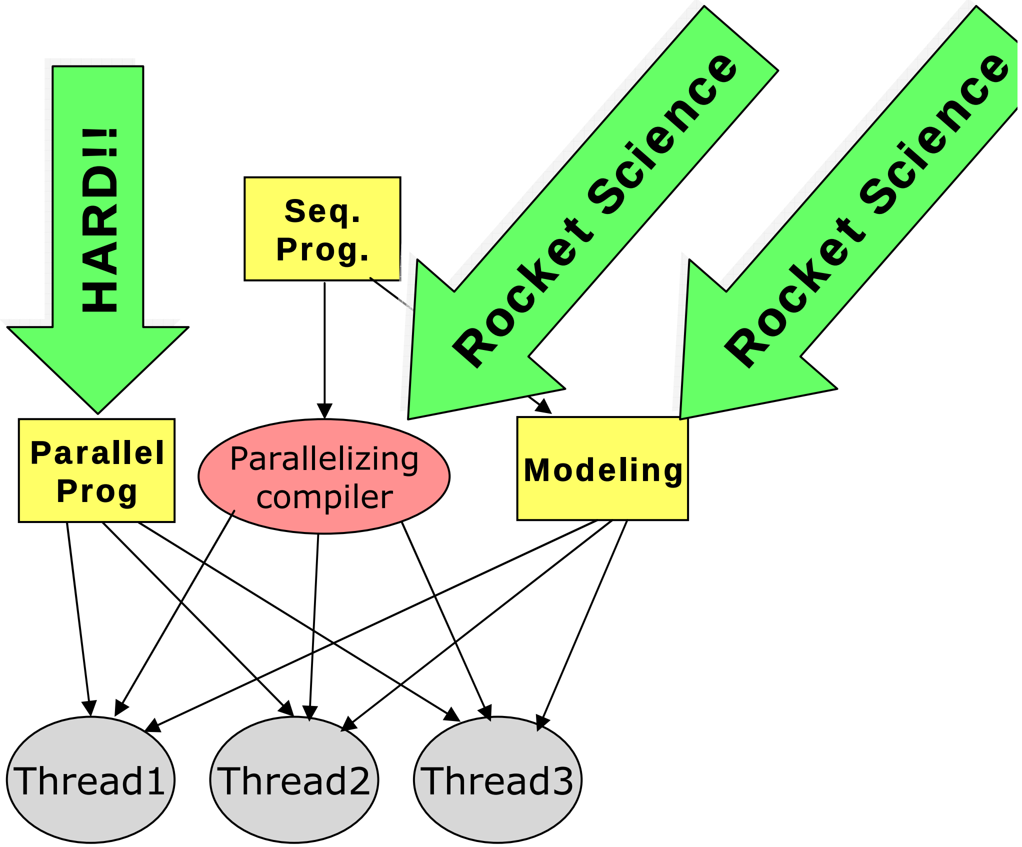
Microprocessor:

Multi core:
A multiprocessor on a chip!



How to create threads?





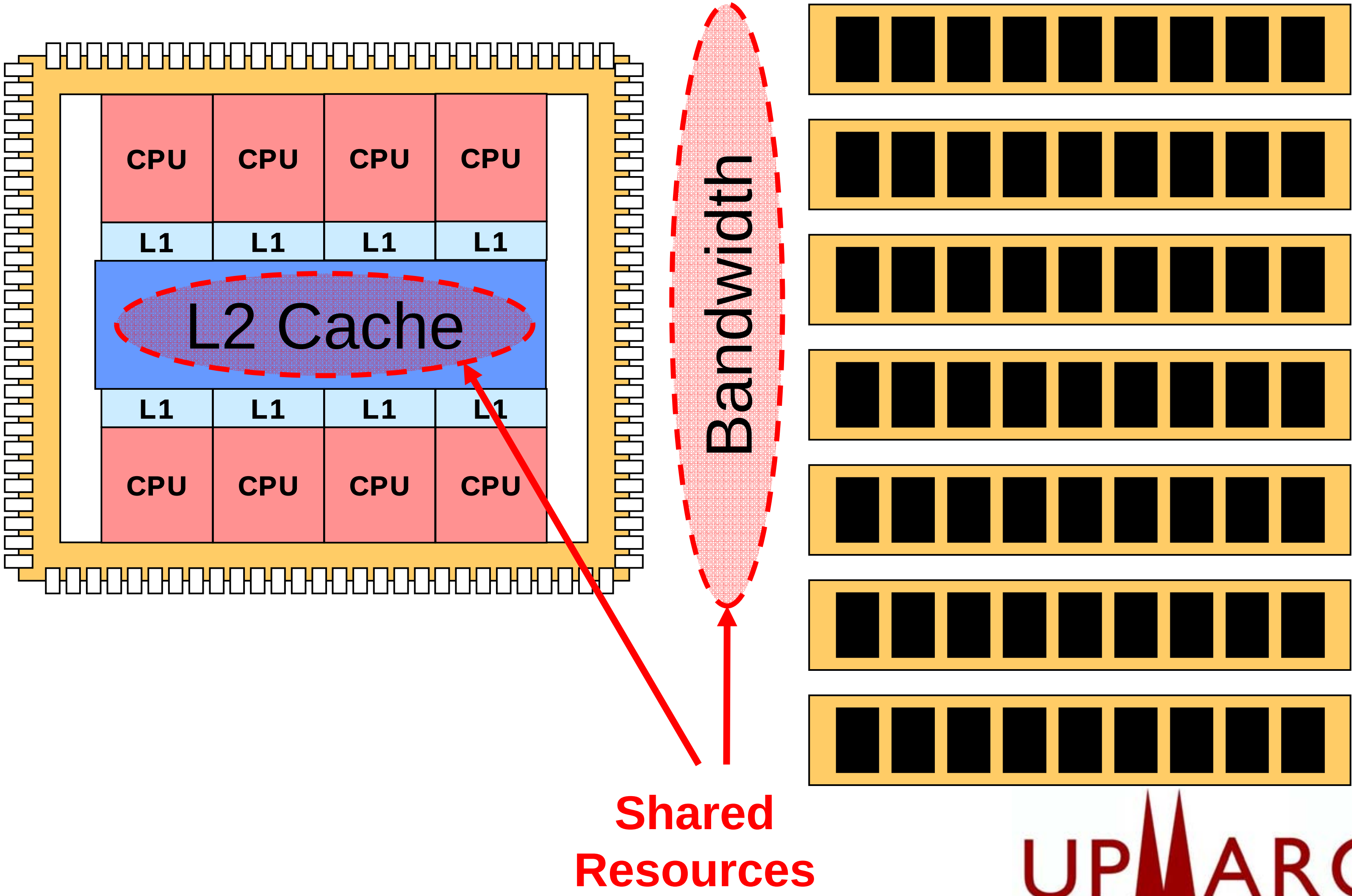
Throughput Computing

- **"Multitasking"**
- **Virtualization**
- **Concurrency**
- **...**

Capability Computing

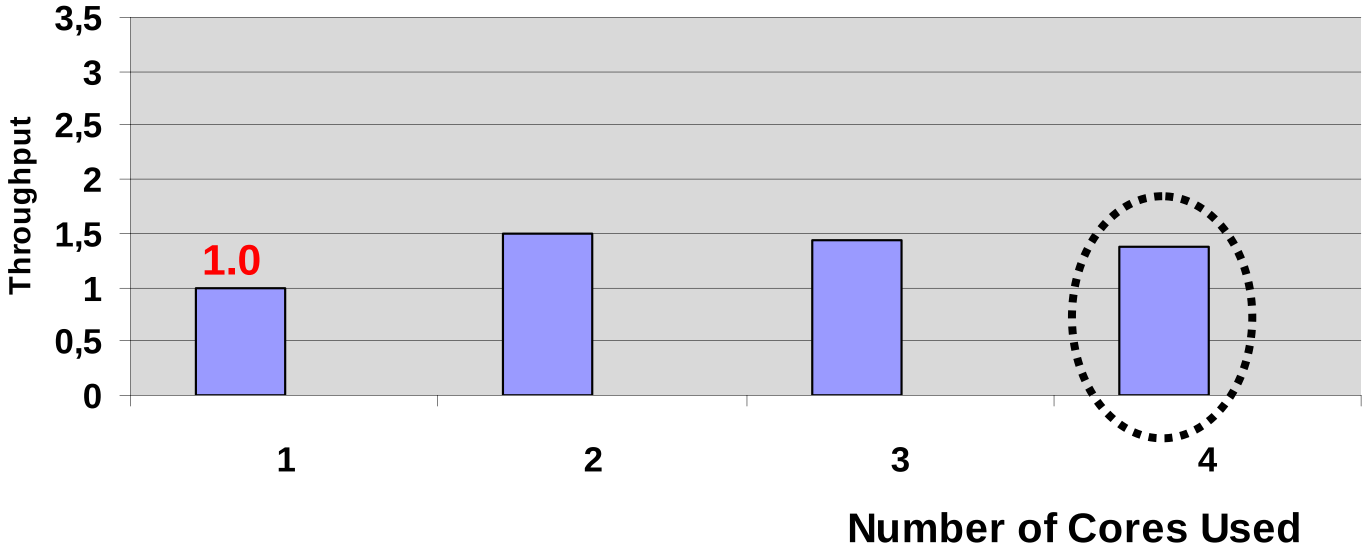
- **Weather simulation**
- **Computer games**
- **...**

Shared Bottlenecks



Example: Poor Throughput Scaling!

Example: 470.lbm

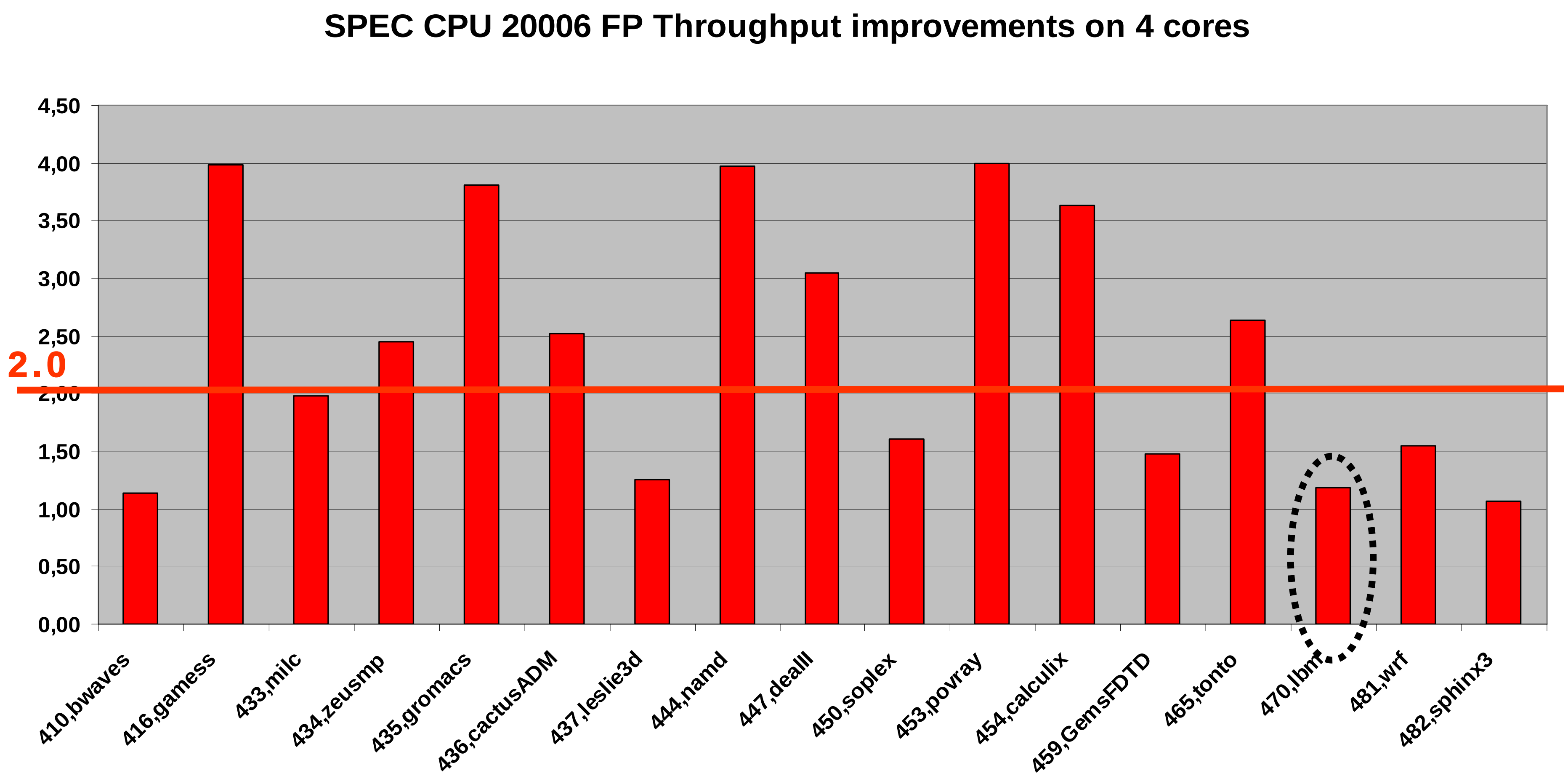


Throughput (as defined by SPEC):

Amount of work performed per time unit when <u>several instances</u> of the application is executed simultaneously.

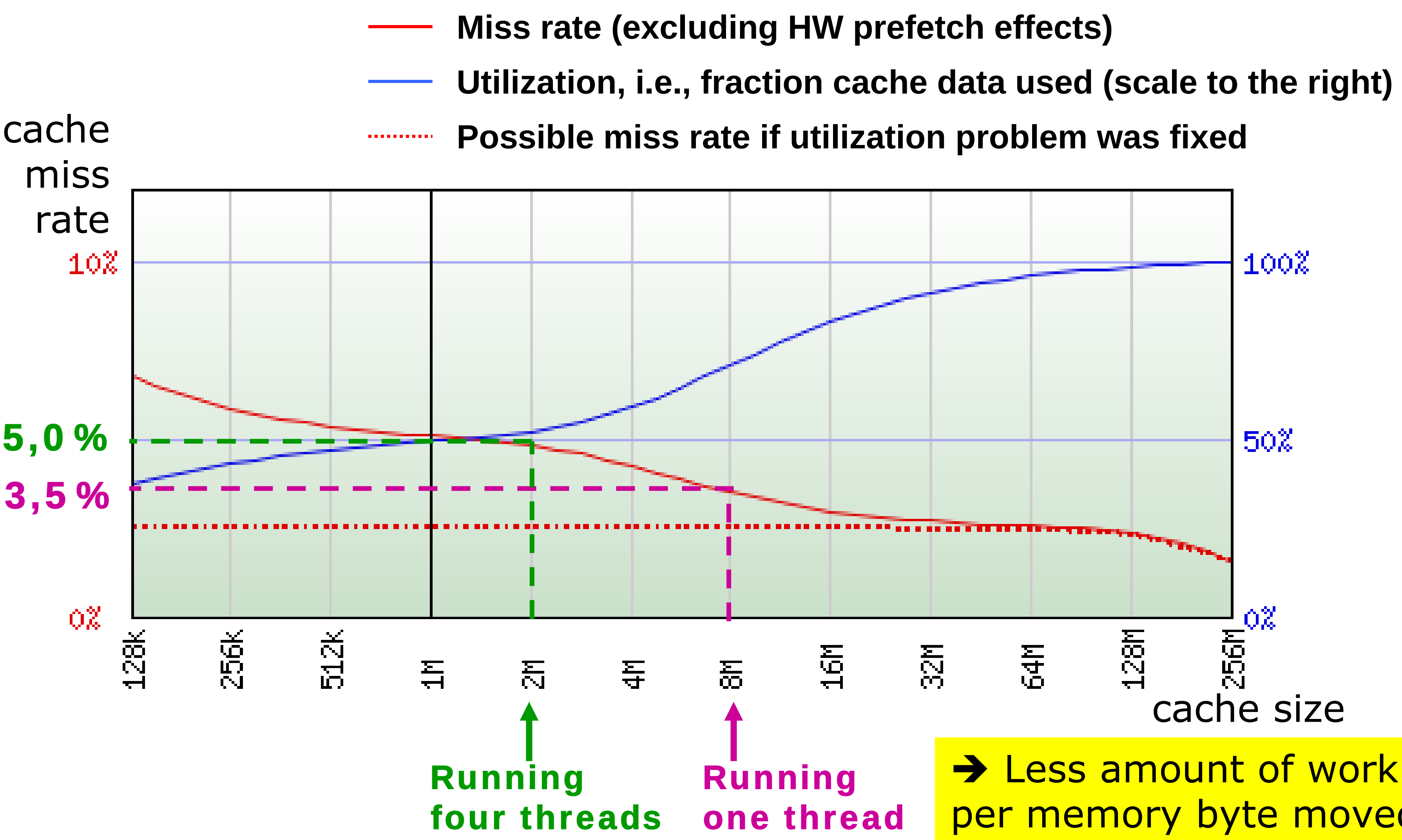
<u>Our TP study</u>: compare TP improvement when you go from 1 core to 4 cores

Throughput Scaling, More Apps



Intel X5365 3GHz, 1333 MHz FSB, 8MB L2.
(Based on data from the SPEC web)

Nerd Curve: 470.lbm

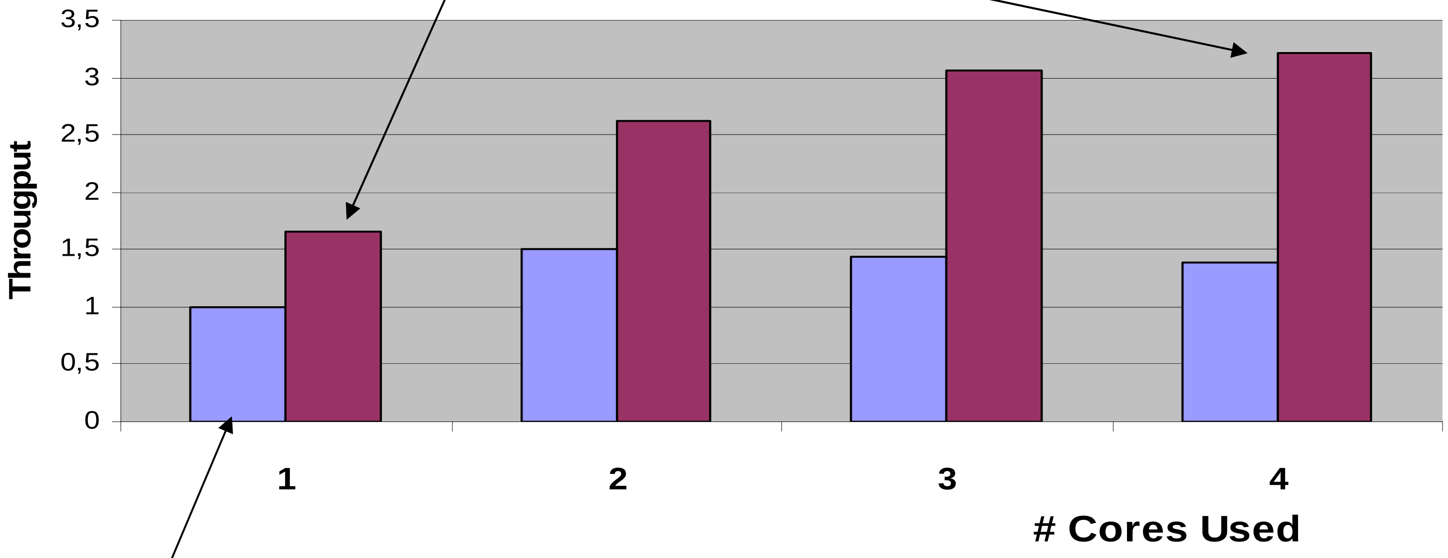


➔ Less amount of work per memory byte moved @ four threads

➔ Better Memory Usage!

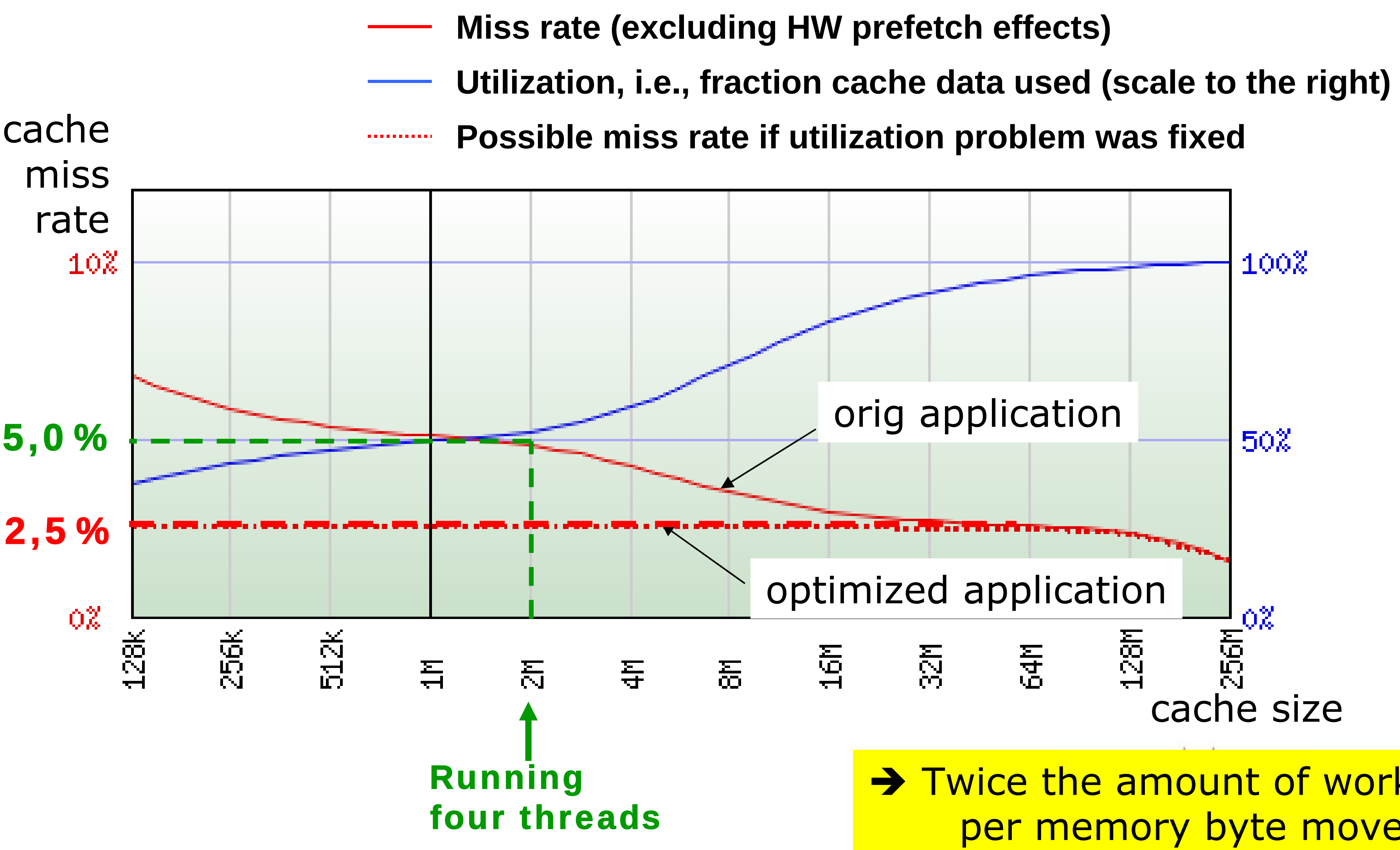
Example: 470.lbm

Modified to promote better cache utilization



Original code

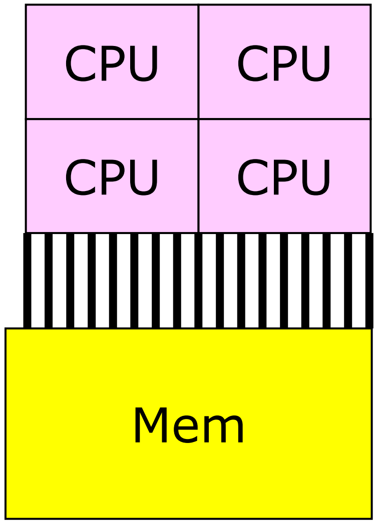
Nerd Curve (again)



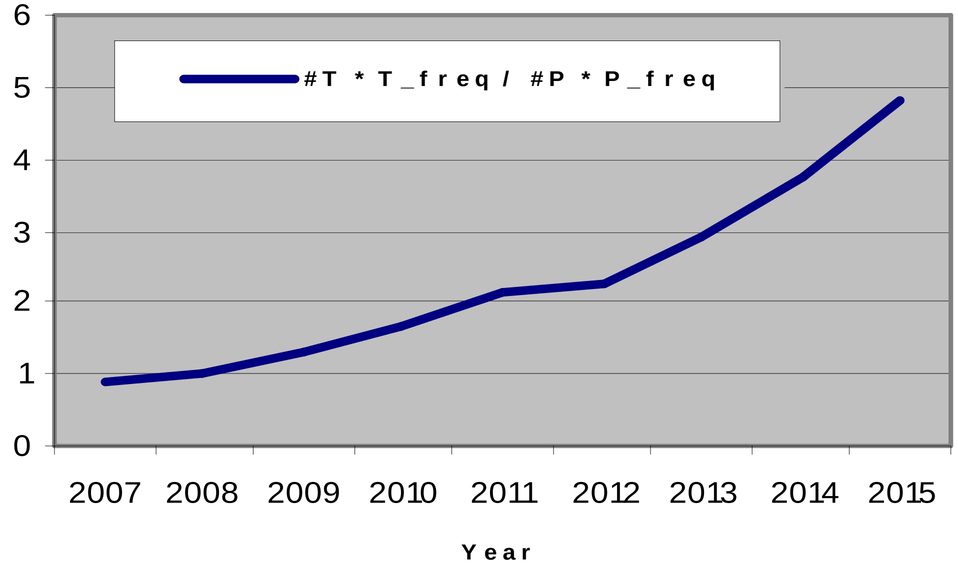
➔ Twice the amount of work per memory byte moved

BW in the Future?

#Cores ~ #Transistors



Computation vs Bandwidth



Source: Internatronal Technology Roadmap for Semiconductors (ITRS)

See also: Karlsson et al. *Conserving Memory Bandwidth in Chip Multiprocessors with Runahead Execution*. IPDPS March 2007.

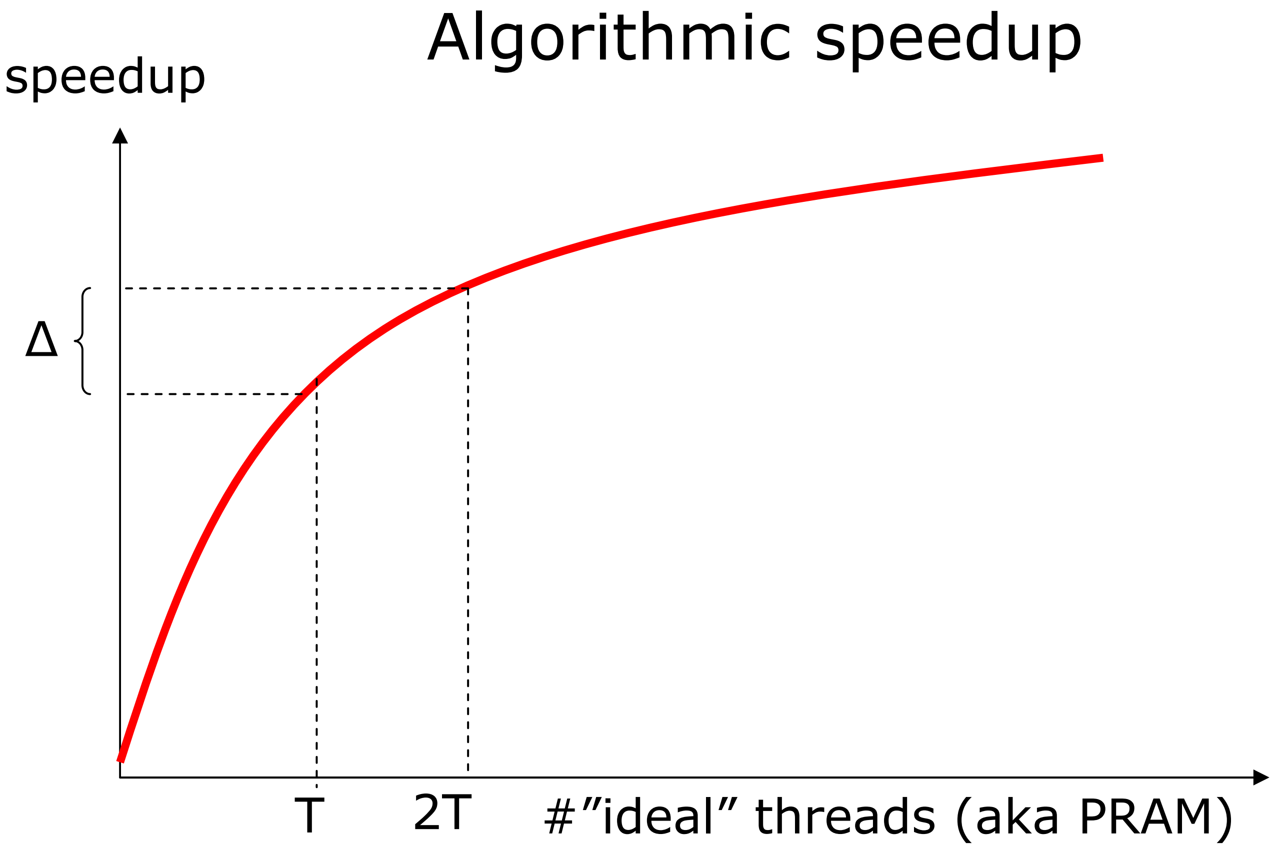
See also IDC presentation from ISC June 2007

Throughput Computing

- Does not expore the shared caches well
- Requires N times more memory

- If limited by memory capacity or memory bandwidth
  ➔go parallel

Algorithmic Speedup?

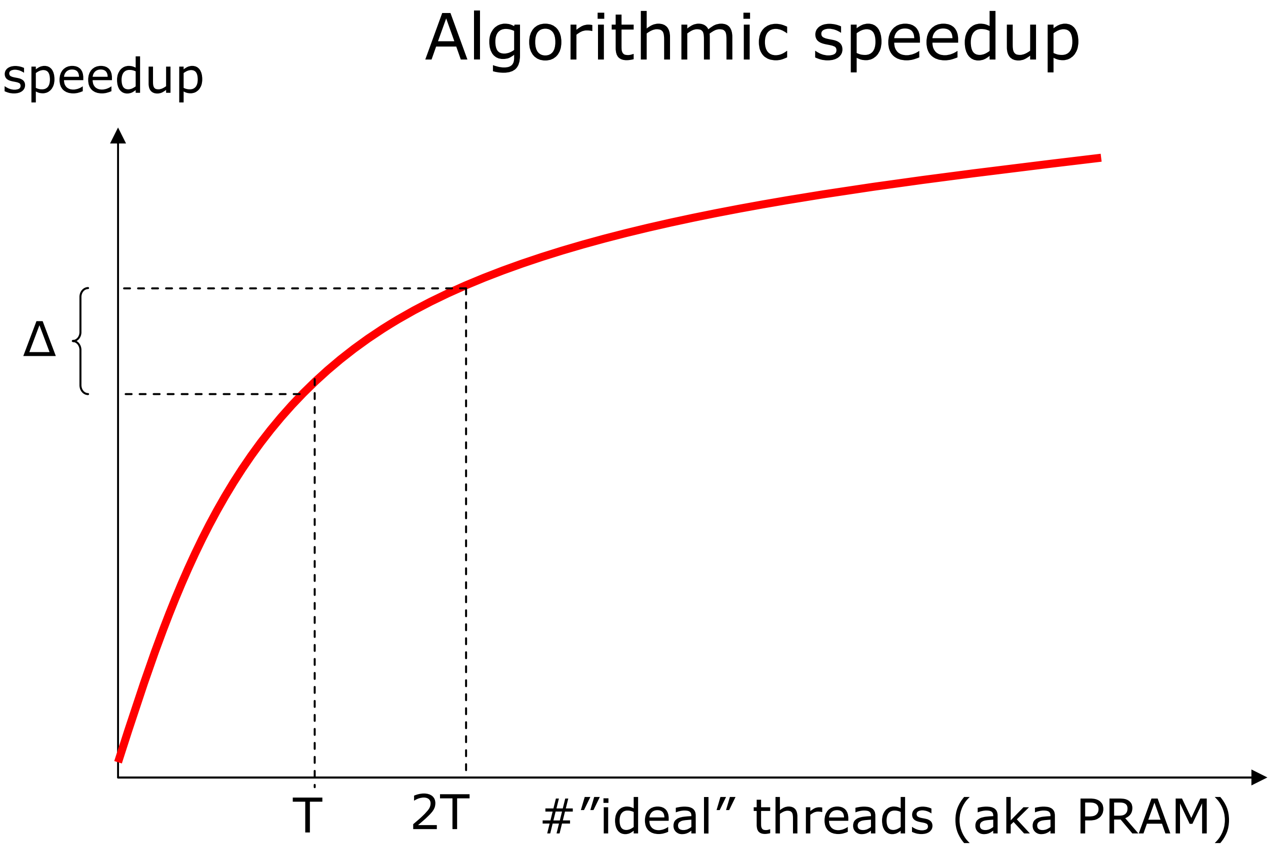


SMT (aka Hyperthreading)

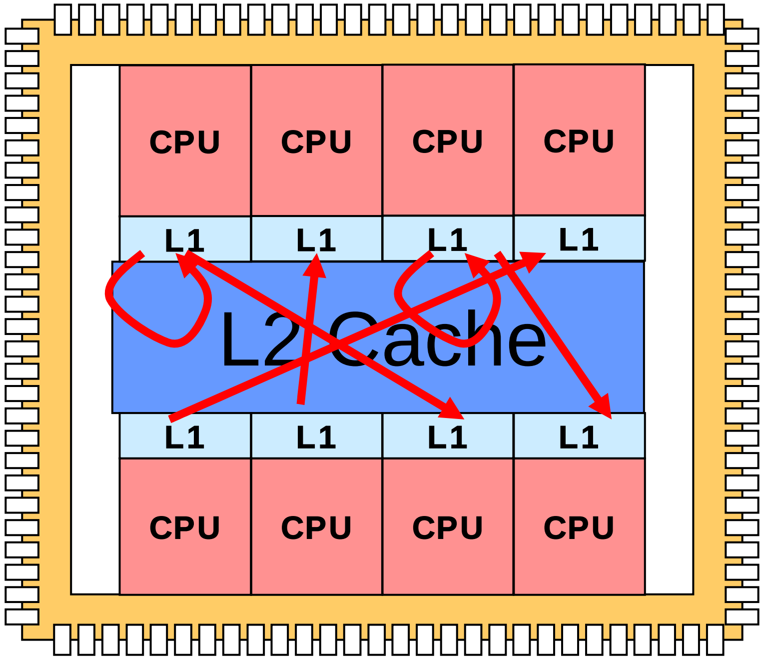
Running many threads per core!

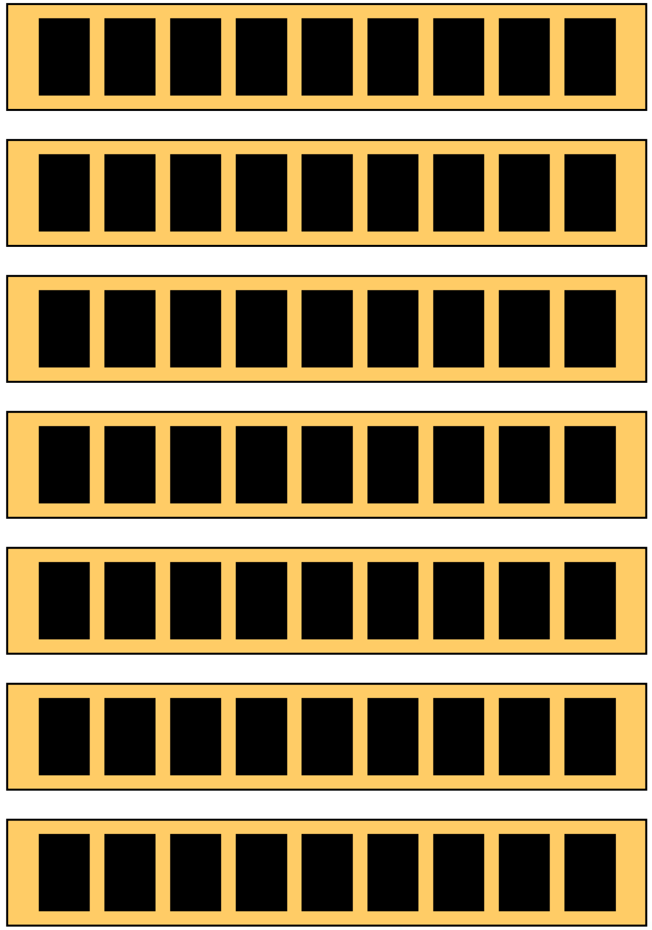
+ hides memory latency

- Teases Amdahl!!



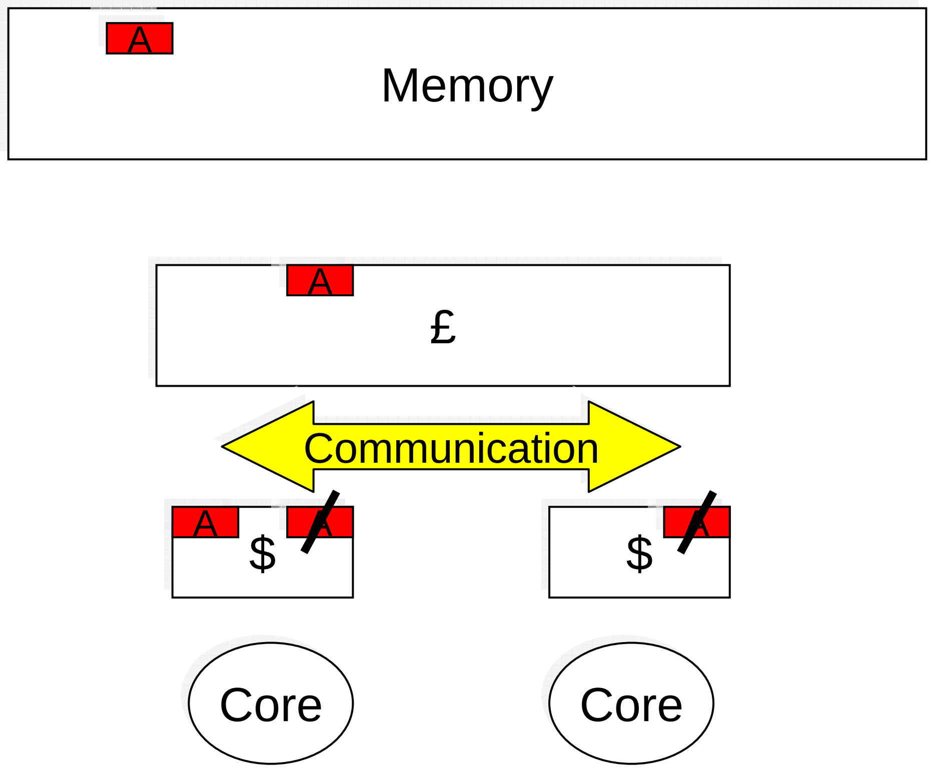
Thread Interaction





- Coherence traffic
- Communication utilization
- Load imbalance
- Synchronization
- False sharing
- ...

Coherence Traffic

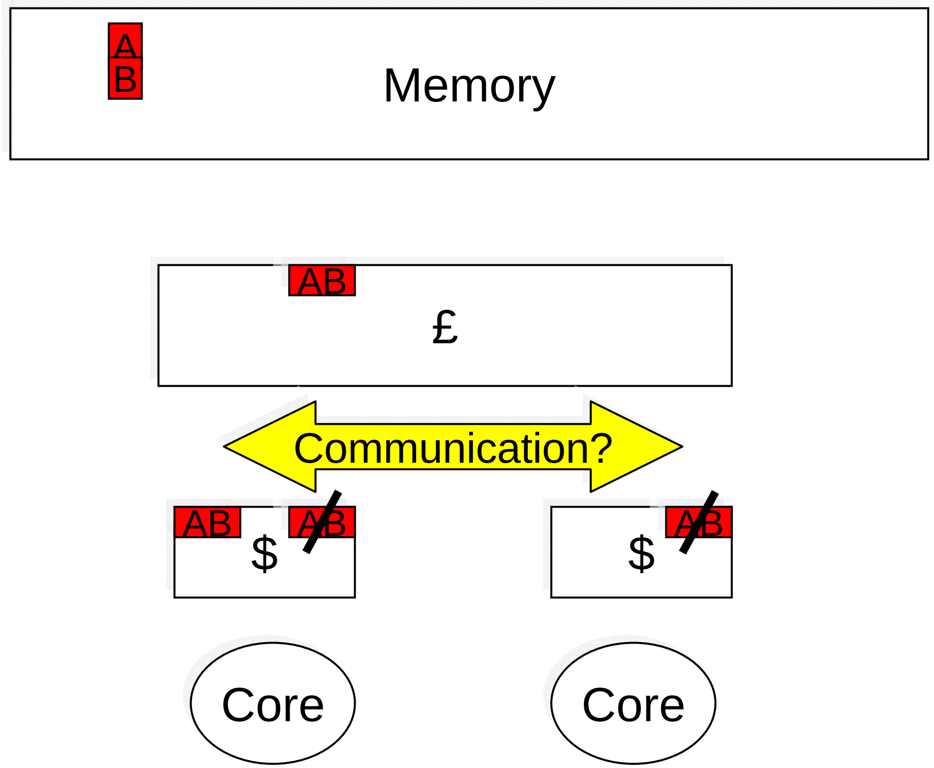


A:= 1

A := 2

A:= 3

False sharing



A:= 1

B := 1

A:= 2

UPPSALA UNIVERSITET

More SW Issues

- Writing the parallel program is the easy part!
- Making sure it is correct is hard!

- Testing?
- Determinism?
- Performance stability?
- Portability?

UPPSALA UNIVERSITET

Do We Have the Skills?

Today's applications a poor fit for multi core

➔ Lots of work to be done

➔ Few experts to do the job

Need to:

... increase the productivity of the experts

... enable non-experts to write efficient multi-core applications

➔ **Need education and tools!**

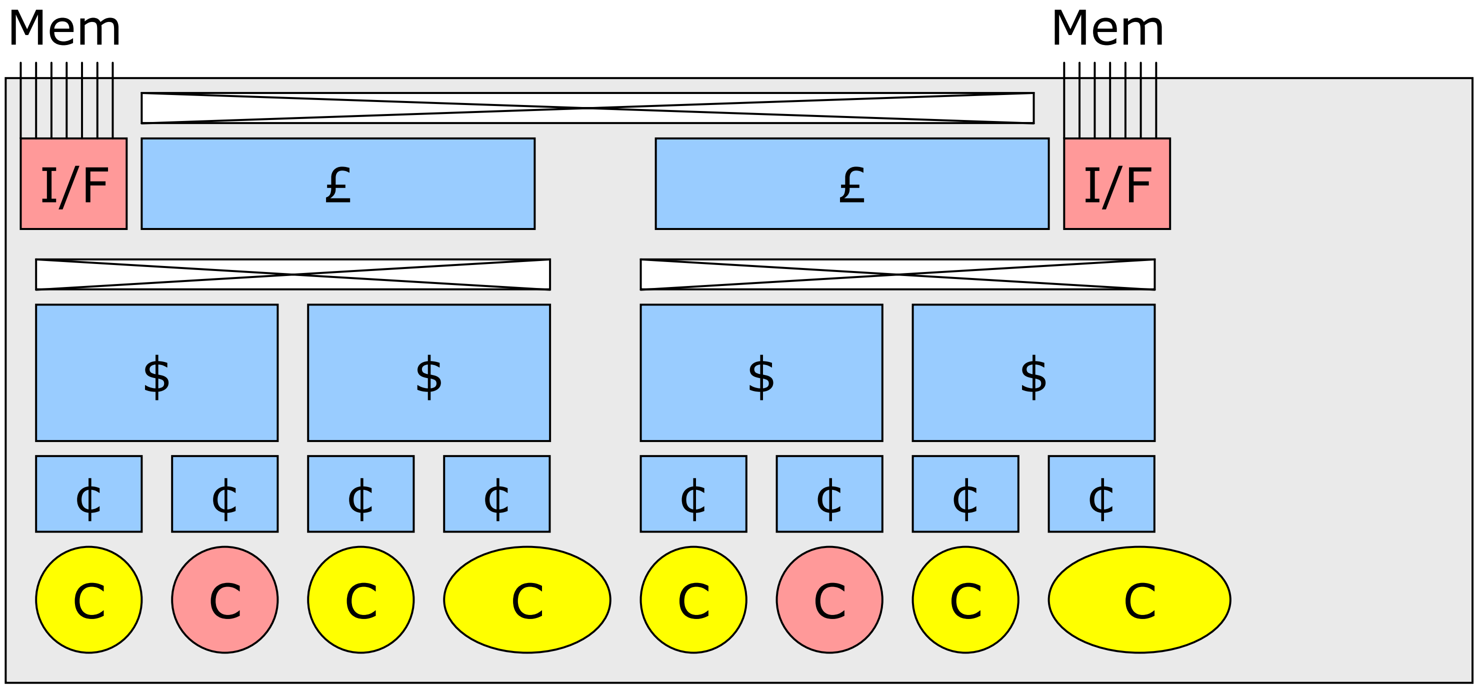
UPMARC

Moving Ahead: Many Cores

- #cores doubling every 18mo (??)
- NUMA rears its ugly head again!
- Hetergeneous cores
- ➔ The same problems, but worse!

- Will the bandwidth scale?
- Will Amdahl bite us?

Heterogeneous NUMA / NUCA



New challenges...

- Programming
- Compilers
- Tuning
- Runtime scheduling
- ...

What matters for performance?

- Are we buying…
  - CPU frequency
  - Number of cores?
  - MIPSs and FLOPs?
  - Performance/Watt?
  - Memory bandwidth?
  - Cache capacity?
  - Memory capacity?

Bad Cop Says:

- This is tough stuff!
- We do not yet understand...
  - ...what HW support is needed
  - ...what languages to use
  - ...how to get it efficient
  - ...how to get it right
  - ...what education is needed

...to use many core in a general setting

- I will take many years until we are ready
- There is not a ”one solution fits all”

- Uppsala Programming for Multicore Architecture Center
- Research Council (VR) Linneus Grant 2008
- 62 MSEK ($10M) / 10 years
- Research areas:
  - Performance modeling
  - New parallel algorithms
  - Scheduling of threads and resources
  - Testing & verification
  - Language technology
  - MC in wireless and sensores

UPPSALA UNIVERSITET

Also…

- Swedish Multicore Initiative!

- Acumem AB
  - Tools to enable multi cores to the masses
  - Research started eight years ago

UPMARC

Questions for the Future

- What applications?
- How to get parallelism?
- How to get data locality?
- How to debug?
- A case for new languages?
- A case for automatic parallelization?
- Are we buying:
  - compute power,
  - memory capacity, or
  - memory bandwidth?
- Will 100 cores be mainstream in 5 years?
- Will the CPU market diverge into desktop/capacity/capability/special-purpose CPUs again?
- **A non-question: will it happen?**

Summing up: More "Performance"

- Thread-level parallelism on the chip
- New walls: Bandwidth, locality and parallelism
- Important R&D areas (again)
  - Algorithms
  - Parallelization
  - Verification
  - Modeling/Simulation/Tools
  - Managing data locality
  - Bandwidth optimizations
  - …

12.00 Lunchbreak (lunch not included)

13.00 Prof. Kunle Olukotun, Stanford: Keynote address
14.00 Prof. Anant Agarwal, MIT and Tilera: "Tile Processor, GP Many Core"
15.00 Coffee

Parallel tracks
Track 1
15.30 Tutorial on Intel TBB by Alexey Kukanov, Intel

Track 2
15.30 Prof. Erik Hagersten, Acumem: "Analysis Technology for MC"
16.00 Prof. Per Stenström, Nema Labs: "Accelerate Reliable Threading"

Track 3
15.30 Prof. Bengt Johnsson, Uppsala University: "UPMARC..."
16.00 Dr. Joe Armstrong, Ericsson: "Erlang: Concurrent programming MC"

16.30 James Reinders, Intel: "Parallelism for Multicore and Manycore"