song of hardware & databases

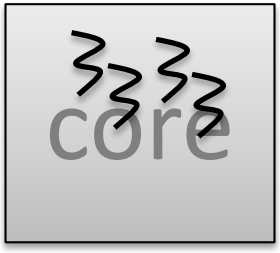
Pınar Tözün
IBM Research – Almaden

has nothing to do with what I do at IBM, this is a hobby talk!

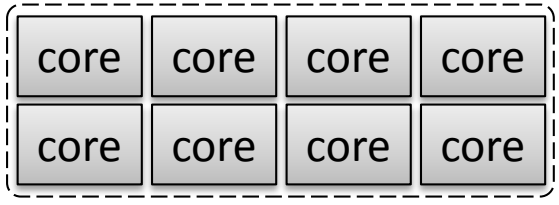
summer

2005 ← previous winter

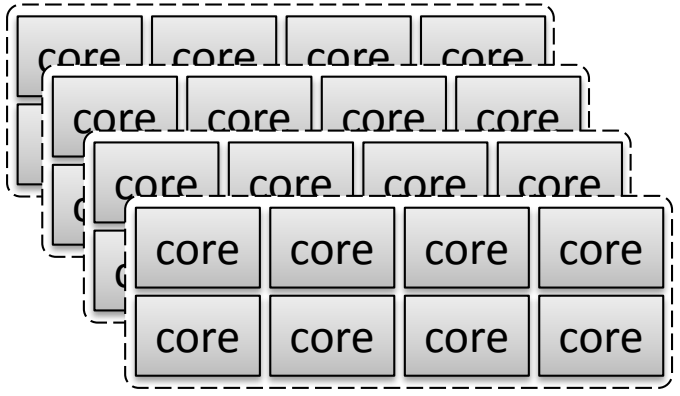
implicit parallelism | **explicit parallelism**



pipelining
ILP
multithreading



multicores
(CMP)



multisocket
multicores

implicit parallelism → free lunch

explicit parallelism → have to work hard to exploit it

wording courtesy of Wikipedia

prophecies

Moore's law

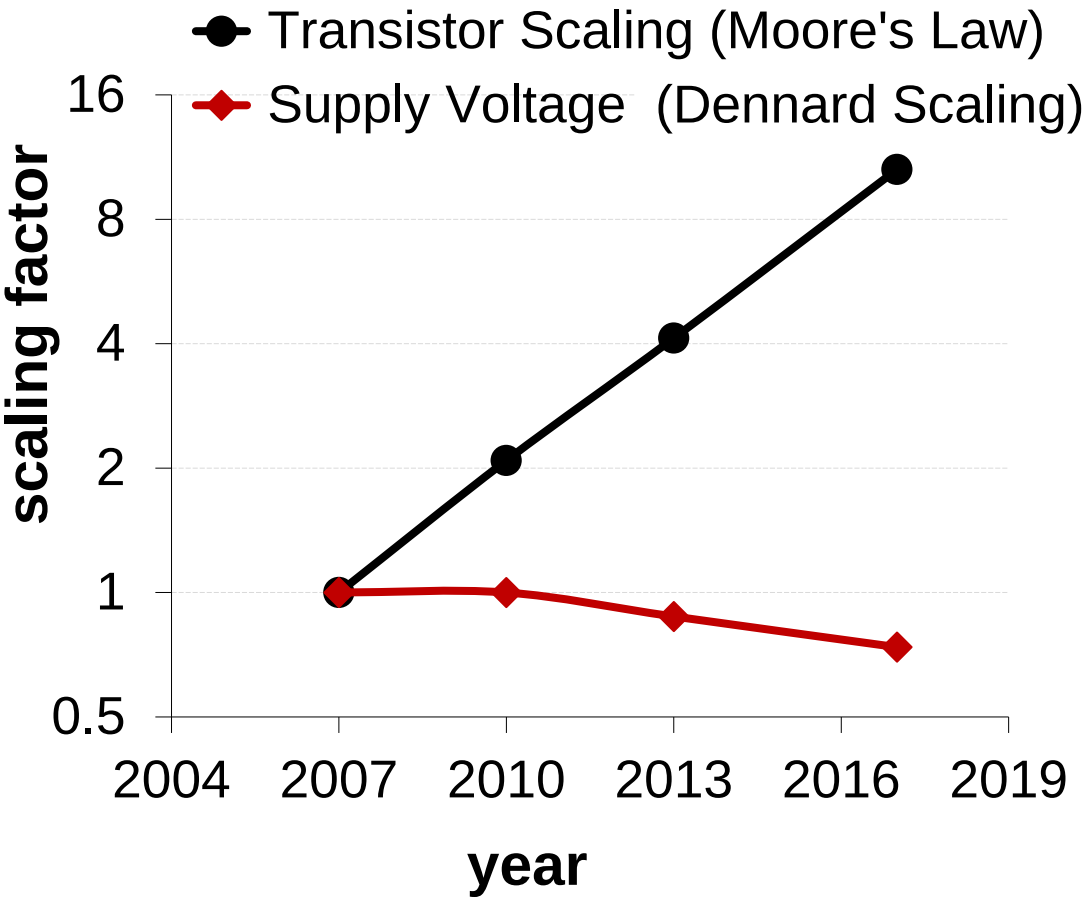
"… the observation that the number of transistors in a dense integrated circuit doubles approximately every two years."

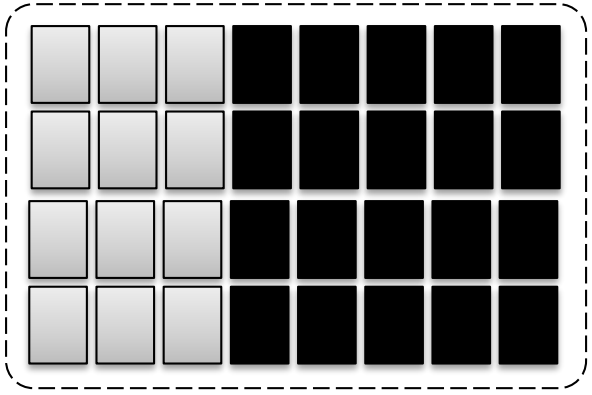
Dennard scaling

" … as transistors get smaller their power density stays constant, so that the power use stays in proportion with area: both voltage and current scale (downward) with length."

graph courtesy of Hardavellas et al.

another winter has come





Dennard scaling once again doesn't hold anymore

age of *dark silicon* is upon us

surviving the winter

- we don't need no innovation

  everyone loves 2-socket intel multicores

- be more energy-conscious

  reduce total number of instructions
  minimize/amortize cost per instruction

- evolve for/with new hardware

  commoditize heterogeneity & specialized hardware

being more energy-conscious

- amortize cost per instruction
  - e.g., SIMD – not always easy to exploit
- minimize cost per instruction
  - e.g., minimize number of cache misses
- reduce total number of instructions
  - e.g., leaner designs, compilation optimizations

writing more efficient code helps

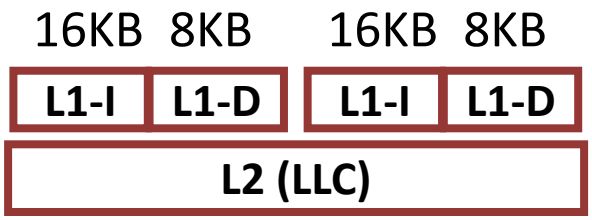
energy-efficiency as a goal while building systems

specialized memory hierarchy

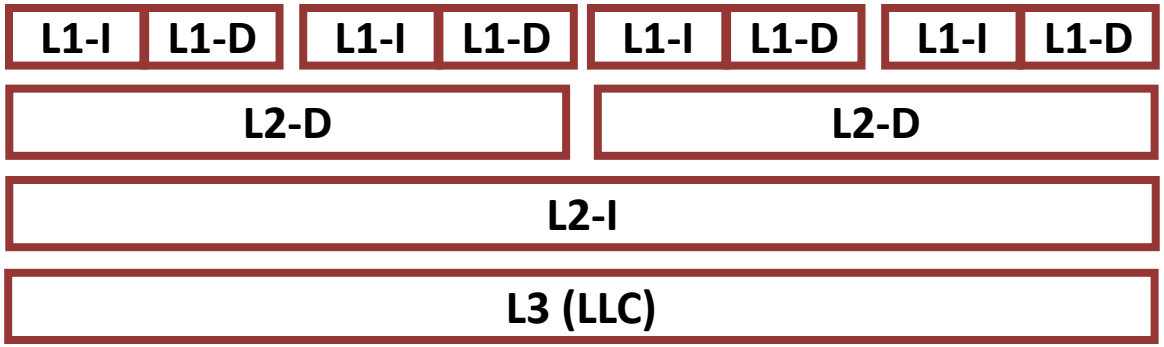
data-intensive workloads have

- large instruction footprint
- low temporal data locality

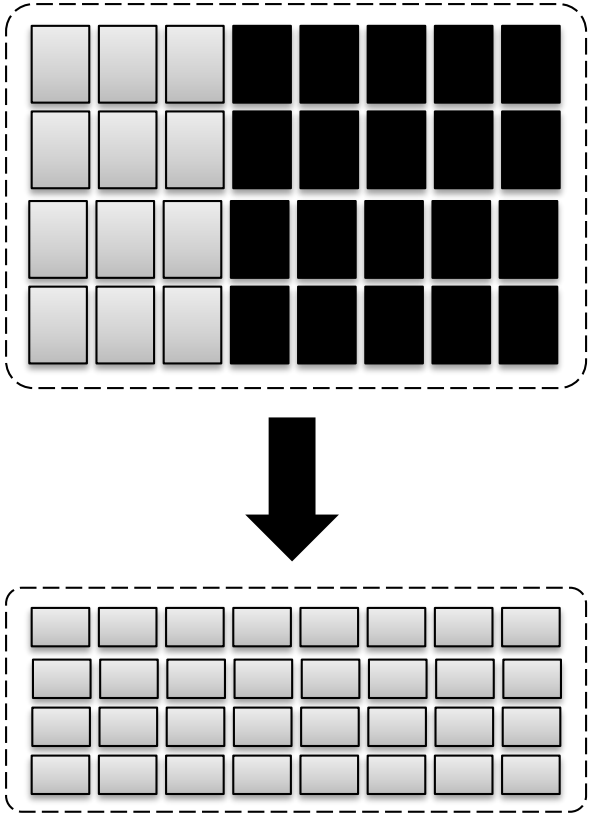
UltraSPARC T2 (Niagara 2)



SPARC M7 & M8

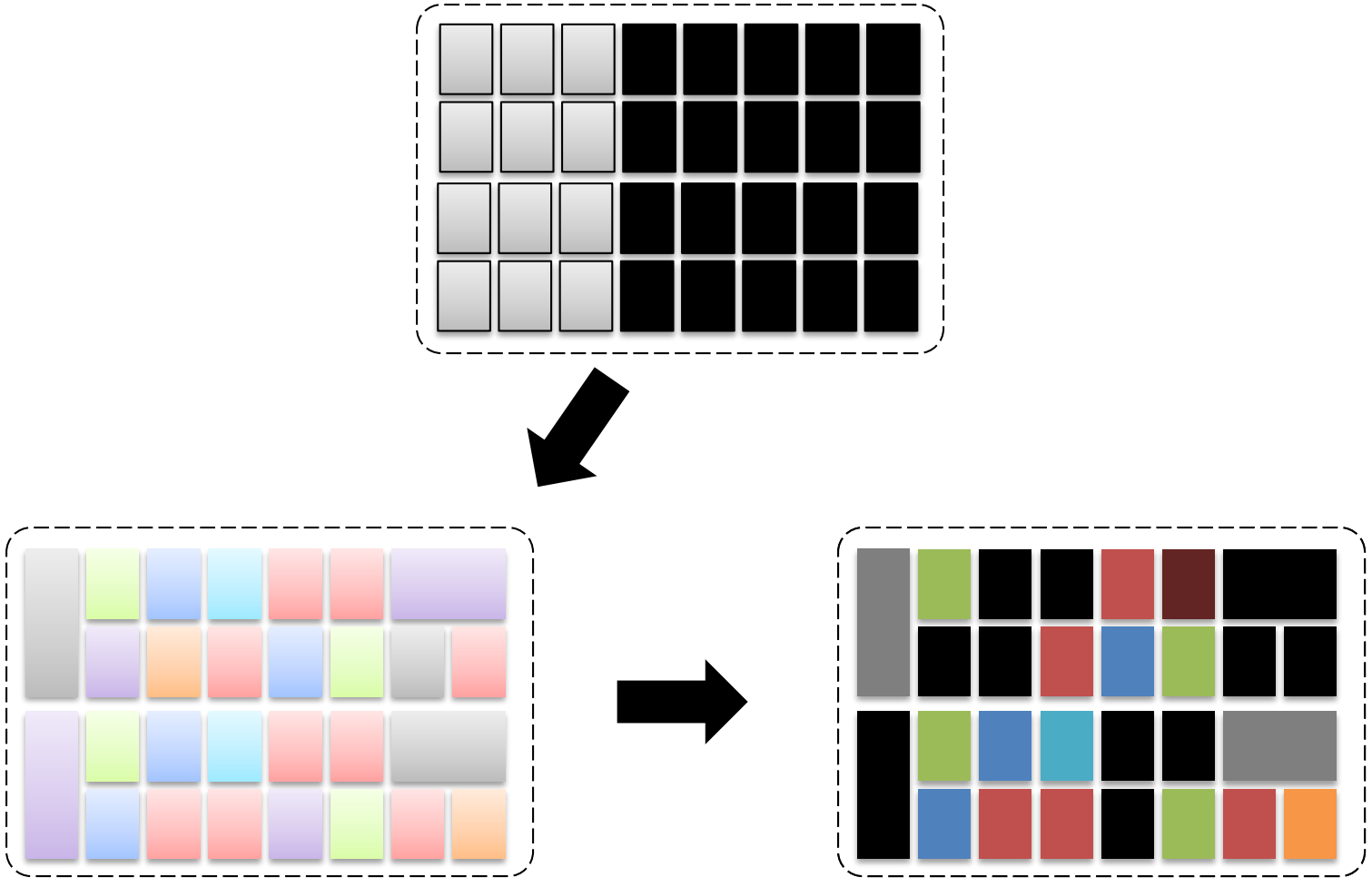


lighter cores



not feasible for latency critical stuff

colorful cores



better long-term solution, but creates mess

how to deal with the mess?

- compilers
- scheduling
- finding the right use case
- scale

haven’t we tried already?

- database machines — DeWitt et al. [TOC79]
- reprogrammable accelerators — Mueller et al. [VLDB09]
- meet the walkers — Kocberber et al. [MICRO13]
- bionic databases — academic bros [CIDR13]
- database processing unit — Wu et al. [ASPLOS14]
- RAPID — Oracle Labs [MICRO17]

economic challenges

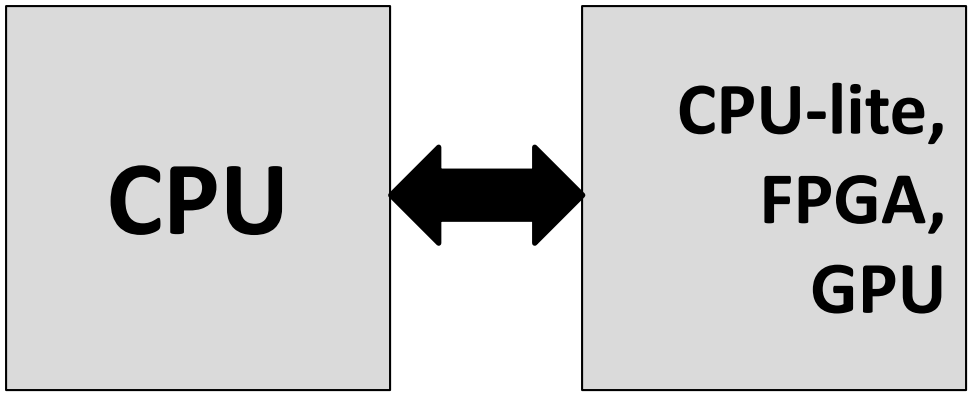
some light

tpu

catapult

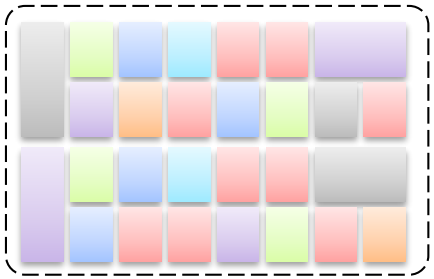
dax

machine learning on FPGAs & GPUs



conclusions

- Dennard scaling is broken again



- be more energy-conscious

  reduce total number of instructions
  minimize/amortize cost per instruction

- evolve for/with new hardware

  commoditize heterogeneity & specialized hardware

thank you!