Should We Defy Amdahl's Law

(or DAL's motivations)

André Seznec

INRIA/IRISA

DAL: Defying Amdahl's Law

- ERC advanced grant to A. Seznec (2011-2016)

DAL objective:

« Given that Amdahl's Law is Forever propose *(impact)* the microarchitecture of the 2020 General Purpose manycore »

10 years in the multicore era and what ?

- Multicores are everywhere

- Parallel (mainstream) apps do not materialize

Multicores are everywhere

- Multicores in servers, desktop, laptops
  - 2-4-8-12 O-O-O cores
- Multicores in smart phones, tablets
  - 2-4-(not that simple) cores
- Manycores for niche markets
  - 48-80-100 simple cores
    - Tilera, Intel MIC

Multicore/multithread for everyone

- End-user : improved usage comfort
  - Can read e-mail and hear MP3

- Parallel performance for the masses?
  - Very few (scalable) mainstream // apps
    - Graphics
    - Niche market segments

No parallel software bonanza in the near future

- Inheritage of sequential legacy codes
- Parallelism is not cost-effective for most apps
- Sequential programming will remain dominant

Inheritage of sequential legacy codes

- Software is more resilient than hardware
  - Apps are surviving/evolving for years, often decades
    - Very few parallel apps now
- Unlikely redevelopment of parallel apps from scratch
- Computing intensive sections will be parallelized
  - But significant code sections will remain sequential

Parallelism is not cost-effective for most apps

- Why parallelism ?
  - Only for performance

- But costly:
  - Difficult, man-time consuming, error prone
  - Poorly portable: *functionality and performance*

Sequential programming will remain dominant

- Just easier
  - The « Joe » programmer
  - Portability, maintenance, debug

- \+ compiler to parallelize
- \+ parallel libraries
- \+ software components (developped by experts)

Looking backwards

2002: The End of the Uniprocessor Road

- Power and temperature walls:
  - Stopped the frequency increase
- 2x transistors: 5 %? 10 % ? perf. (if any)

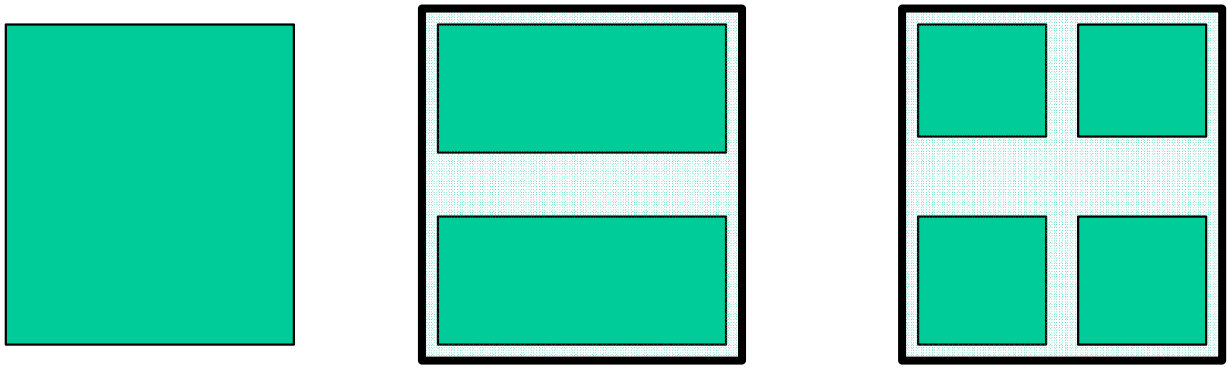
economical logic : buy smaller chips !

IC industry needs to sell new (expensive) chips:

Marketing:

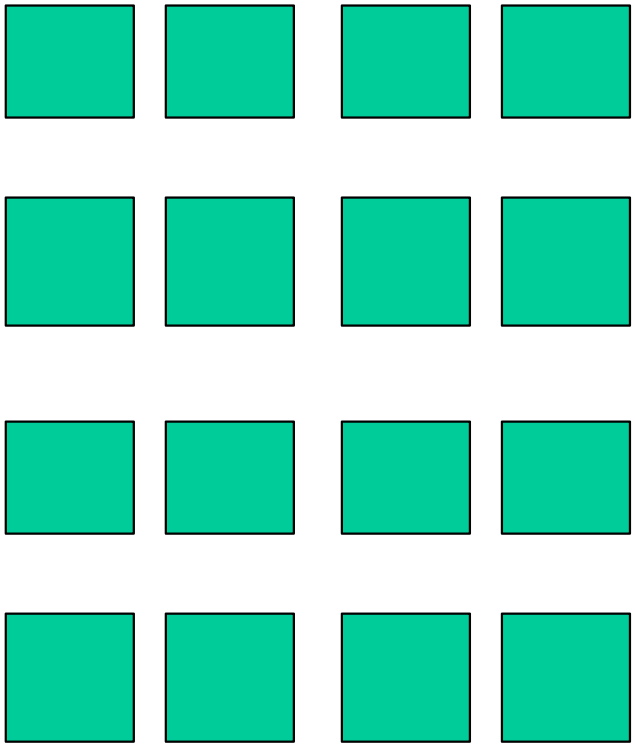
« You need 2 (4, 8) cores »

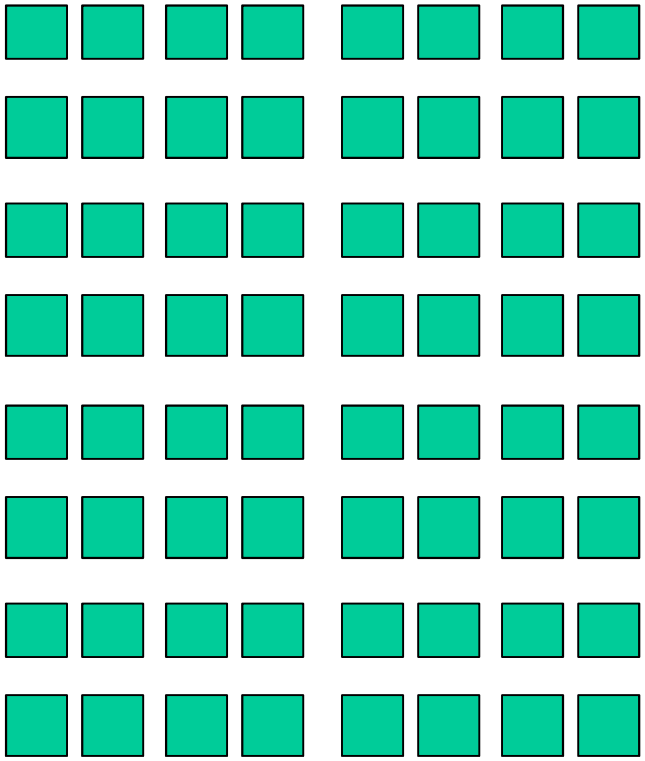
Marketing multicores to the masses 2002- ..



GREAT !!

And now ?





The end user is not such a fool ..

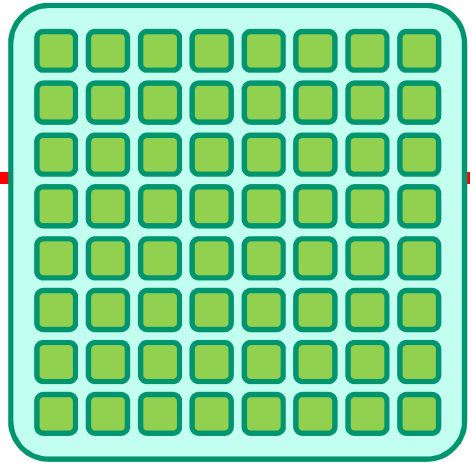
Following the trend: 2020

- Silicon area, power envelope
  - for 100 Nehalem class cores

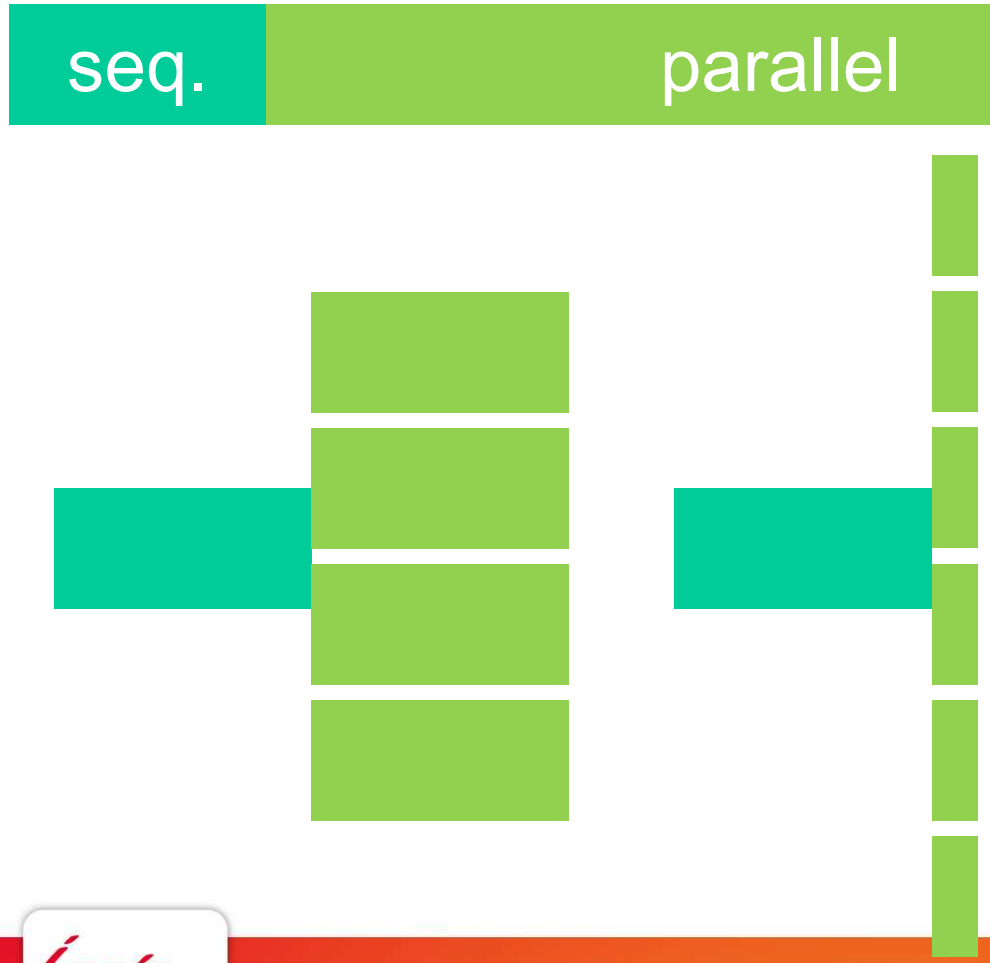
or

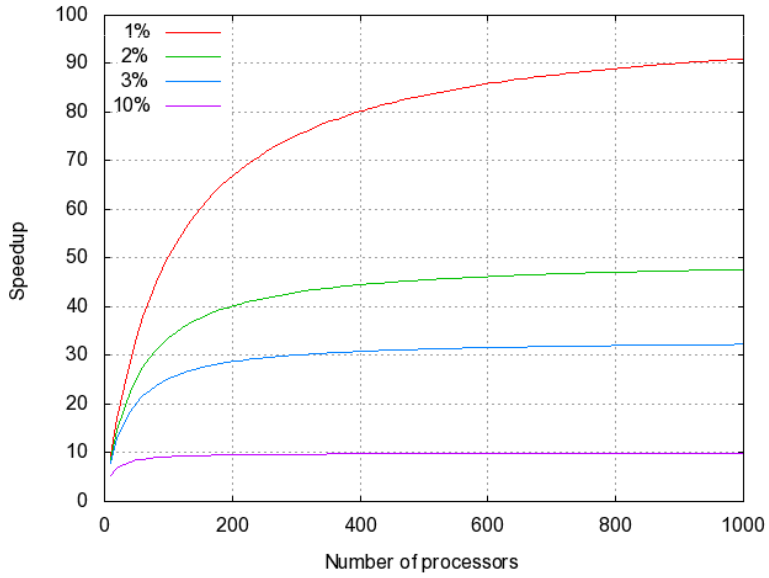
  - for 1,000 simple cores (VLIW, in-order superscalar)

Amdahl's Law



"Cannot run faster than sequential part"





Naive model

- A parallel application:
  - Parallel section: can use 1000 processors
  - Sequential section: run on a single processor

SEQ: fraction of code in sequential section

Complex cores against simple cores

- CC: 100 complex vs SC :1000 simple cores

  with complex 2X faster than simple

if SEQ > 0.8 % then CC > SC

And if ..

- Use a huge amount of resource for a single core:

➔10X the area of the complex core

➔10X the power of the complex core

➔ Use all the uniprocessor techniques

  - Very wide issue (8 – 16 ?)
  - Ultimate frequency (« heat and run »)
  - Helper threads
  - Value prediction
  - ..

And if ..

- UC ultra complex cores (but only 10)
  - 10X more resources than complex cores
    - but only 10 of them
  - 2X faster

➔ If SEQ > 3.3 % then UC > SC

➔ If SEQ > 8 % then UC > CC

So what ?

- Embarassingly parallel

  ➔ SC simple cores

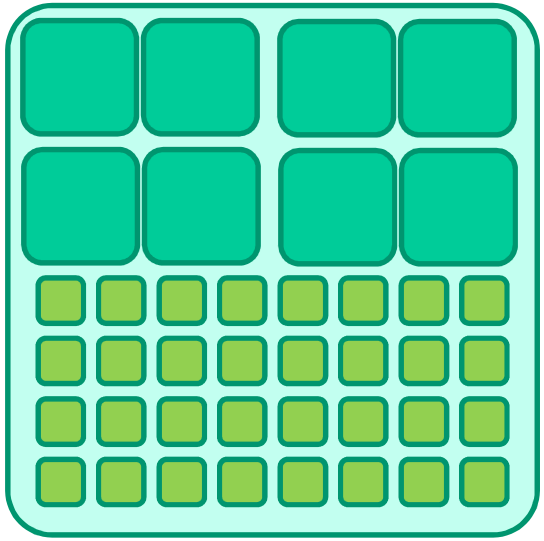
- Some parallel + some sequential

  ➔ CC complex cores

- Sequential+ poor parallel + multiprogrammed

  ➔ UC ultra complex cores

And hybrid SC + CC ?



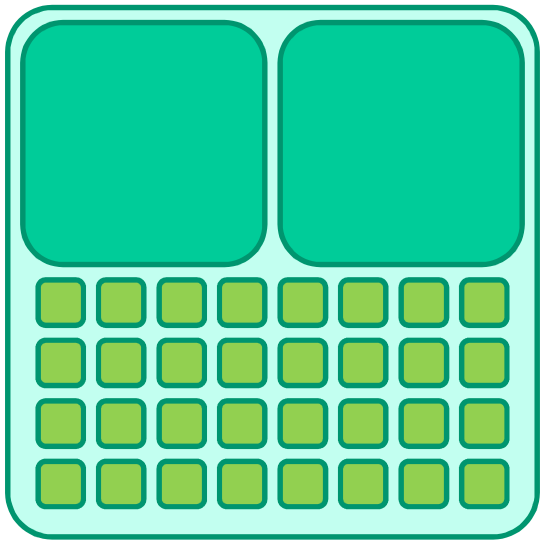
CC_SC:

- 50 complex
- 500 simple

if SEQ> 0.2% then

CC_SC > SC

DAL architecture proposition



- Heterogeneous architecture:
  - A few ultra complex cores
    - to enable performance on sequential codes and/or critical sections
  - A « sea » of simple cores
    - for parallel sections

For our simple model

« DAL » : UC_SC

5 ultra complex cores + 500 simple cores

- If SEQ > 0.13 % then « DAL » > SC
- « DAL » always better than UC, CC, CC_SC

DAL

➔Many groups targetting architecture for parallel performance

➔Many groups targetting energy efficiency

Let us concentrate on performance on sequential apps or code sections

DAL research directions

- Focus on the sequential performance
  - The sequential accelerator
    - Heat and run
  - Microarchitecture of O-O-O execution cores
    - Revisit all the « old » concepts
      - but with quasi-unlimited resources
- Manycores and sequential codes
  - Can we use (adapt) the plurality of (simple) cores ?