2: Introduction to Scalable Computing

Seminars in *Scalable Computing*

Vittorio Scarano

Università di Salerno

ISISLab

Dottorato di Ricerca in Informatica

PLAN

1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: “how-to”
2. COMPUTER SCIENCE LAWS
   - Amdahl’s Law
   - Gustafson-Barsis’s Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl’s law and Multicore Processors
4. CONCLUSIONS

PLAN

1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: “how-to”
2. COMPUTER SCIENCE LAWS
   - Amdahl’s Law
   - Gustafson-Barsis’s Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl’s law and Multicore Processors
4. CONCLUSIONS

SCALABILITY

- **Scalability**: the property of a solution to a problem to mantain its *efficiency* as the *dimension* grows
- Some keywords to be addressed in the context of parallel programming:
  - Efficiency: speedup over the “corresponding” sequential solution
  - Dimension: processors number, type or interconnection; problem size (memory)
- Big-Oh notation for algorithms: scalability, but only in principle
  - what happens when you fill the current level of the memory hierarchy you are using
  - what happens when number of processors grows to infinity
  - ...

PLAN

1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: “how-to”
2. COMPUTER SCIENCE LAWS
   - Amdahl’s Law
   - Gustafson-Barsis’s Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl’s law and Multicore Processors
4. CONCLUSIONS

THE MOST INFLUENTIAL CONCEPTS

- Panel in 2000 IPDPS: find the most influential concepts in Parallel and Distributed Processing field, in the past millennium
- Participants: M. Theys, S. Ali, H.J. Siegel, M. Chandy, K. Hwang, K. Kennedy, L. Sha, K. Shin, M. Snir, L. Snyder, T. Sterling
- The process:
  - Proposal of candidates (≤ 10 per panelist)
  - Formulation of a list of candidates
  - Vote (panelists and audience)

OUT OF THE RESULT TOP 10 LIST

All very important and interesting concepts

- Cellular automata
- Client-server
- PRAM
- Priority inversion
- Neural networks
- RPC
- Zero ☺
- Quantum computing
- ...

THE RESULT

- 10. Multithreaded (lightweight) program execution
- 9. Cluster computing
- 8. Message passing and packet switching
- 7. Load balancing
- 6. Synchronization (including semaphores)
- 5. Multiprogramming
- 4. Divide and conquer
- 3. Pipelining
- 2. Arpanet and Internet
- **1. Amdahl’s law and scalability**

HOW TO CONJUGATE "SCALABILITY"

- Wide area network:
  - cloud computing
- Local area network:
  - cluster computing
- Personal area:
  - desktop/mobile multicore processors
- Is scalability a hopeless battle? (Ask Amdahl. . .)

SCALABILITY: IMPORTANT RESEARCH GOALS

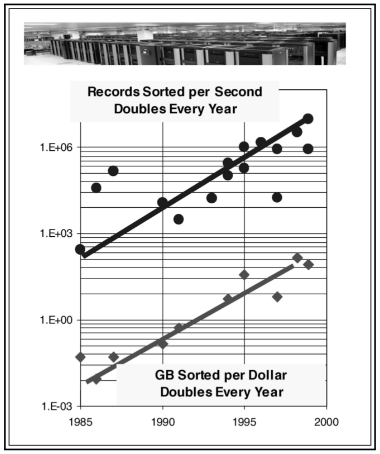
- Jim Gray (Microsoft): "A dozen Information-Technology Research Goals" (Journal of ACM 2003)
- Idenfication of *long-range* Research Goals, i.e., that are:
  - Understandable
  - Challenging
  - Useful
  - Testable
  - Incremental

THE GOAL NUMBER 1: SCALABILITY

- Devise a software and hardware architecture that scales up buy a factor of 10^6.
  - storage and processing that can grow by a factor of a million, either by doing job faster or by doing larger jobs in the same time
- . . . *simply* by adding more resource
- Among the major issues:
  - automatic management
  - automatic parallelism
  - fault-tolerance
  - load-distribution

THE PROBLEM

- Despite the improvements obtained by Internet, DBs, etc.
- When it comes to running a big monolithic on highly parallel hardware, there has been modest progress



AMONG THE OTHER GOALS

- AI's: Turing test, Hearing, Speaking, Seeing
- Personal and World Memex
- Telepresence
- Trouble-free systems (simple administration): secure and "always up"
- Automatic programming

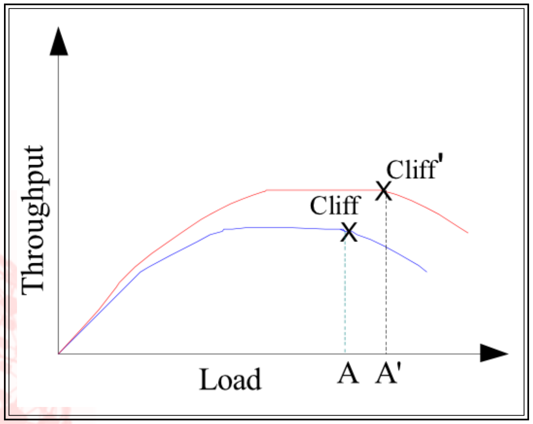
PLAN

HOW TO DEFINE IT

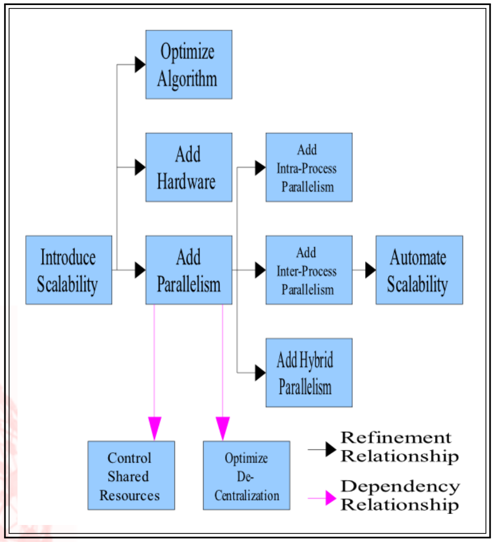
- The property of a solution to a problem to mantain its *efficiency* as the *dimension* grows
- In different contexts/domains, different problems and approaches for solution
    - Scientific computing different than transactional systems
- Triggered by the technological drive ...
- ...but also a "motivating" factor to the technology

"HOW-TO" MEASURE IT

- Different metrics
- Examples: throughput (# transactions per second) with respect to # users



"HOW-TO" REALIZE IT



PLAN

1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: "how-to"
2. COMPUTER SCIENCE LAWS
   - Amdahl's Law
   - Gustafson-Barsis's Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl's law and Multicore Processors
4. CONCLUSIONS

LAWS IN COMPUTER SCIENCE

- Computer science often proposes "Laws"
- Statements, as much simple and sharp as they are:
  - supported only by (sometime limited) empirical evidence
  - proven in very particular settings
- In the last 40 years, several laws are widely cited and "believed":
  - Moore's law, that dictates how fast the number of transistors on a chip increase with time
  - Amdahl's law, that dictates how much performance can be obtained by a program, by executing it in parallel on (arbitrarily) many resources

---

PLAN

1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: "how-to"
2. COMPUTER SCIENCE LAWS
   - Amdahl's Law
   - Gustafson-Barsis's Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl's law and Multicore Processors
4. CONCLUSIONS

A VERY "OLD" LAW: 1967

- Described (only informally!) by Gene Amdahl (IBM) in a 3 page papers of 1967 "*Validity of the single processor approach to achieving large scale computing capabilities*"
- The "validity of single processor" is described against the supportes of the parallel organization of computers (with parallel memories, connected by a bus or point-to-point, with parallel execution streams)
- The basic idea is that:
  1. the "*data management housekeeping*", that is inherently sequential, cannot be parallelized (and therefore improved) on a parallel computer (no matter how many resources it has)
  2. any "geometrically related" problem, given the irregularity of shapes/regions/etc. cannot be mapped onto a regular geometry of components

---

SPEEDUP ACCORDING TO AMDAHL

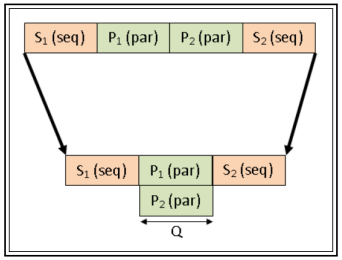
- The*speedup* S of X is the ratio between the sequential time to execute X and the parallel time (n processors) to execute X
- Let P be the part of X that can be parallelized
  - with n processors the parallel part takes time $\frac{P}{n}$ while the sequential takes time S
- Then the speedup is:

Amdahl's law

By executing X on n processors, the speedup is:

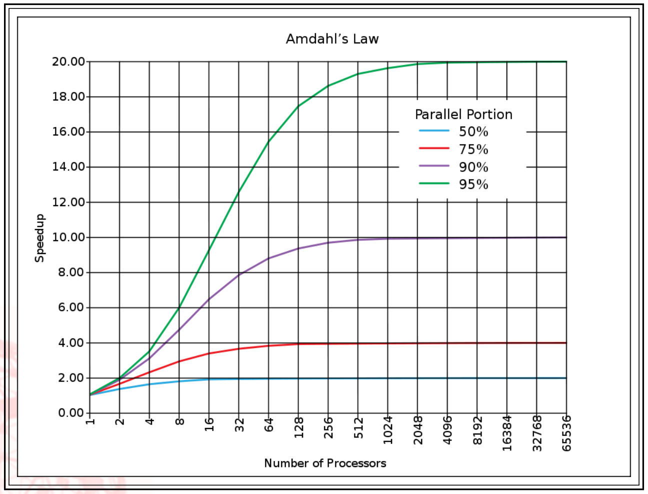
$$\text{Speedup}_A = \frac{S+P}{S+\frac{P}{n}}$$

AN EXAMPLE



- Assume a *fixed program* X with a sequential S and a parallelizable part P
- $n = 2$, $S = S_1 + S_2$, $P = P_1 + P_2$, $S_i = P_i$ fixed
- $\text{Speedup}_A(X) = \frac{(S+P)}{S+P/2} = 4/3$

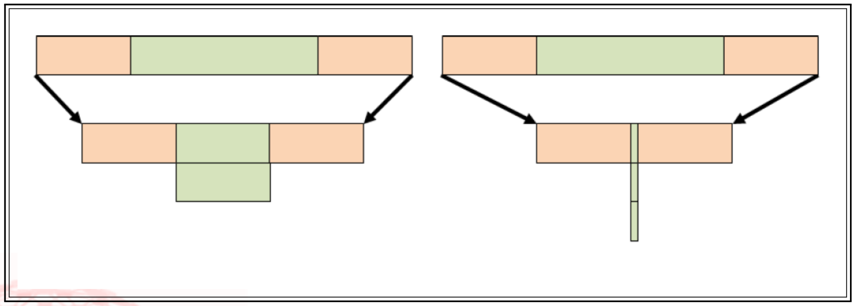
A DIAGRAM OF THE LAW



INTERPRETATION AND COMMENTS

- Amdahl's law indicates that the sequential part of a program will slow down any speedup that we can hope to get from parallelization
- $\lim_{n\to\infty} \text{Speedup}_A = \lim_{n\to\infty} \frac{S+P}{S+\frac{P}{n}} = \frac{S+P}{S}$
- If we set the Amdahl's coefficient $\alpha = S/(S+P)$, speedup is bounded by $1/\alpha$
- *Law of diminishing return* on investment
- $\Rightarrow$ it is not enough to buy/invest in new hardware, but the sequential part must be negligible with respect to the parallel part (*good for us, Computer Scientists!*)

THE SITUATION AT THE LIMIT



PLAN

1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: "how-to"
2. COMPUTER SCIENCE LAWS
   - Amdahl's Law
   - Gustafson-Barsis's Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl's law and Multicore Processors
4. CONCLUSIONS

LAWS AND REALITY

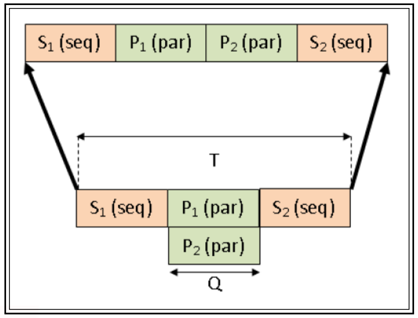
- 1988: Gustafson writes his (and his team) experience:

"Reevaluating Amdahl's Law" (Comm. of ACM, 1988)

The steepness of Amdahl's law graph when $S \to 0$ *for* $N = 1024$ *implies that very few problems will experience even a 100-fold speedup. Yet, for 3 applications (*$S = 0.4 - 0.8$ *percent) we experience speedups between 1016 and 1021.*

- The criticism: "One does not take a *fixed-size* problem and run it on various numbers of processors (except in academic research)" 🙂
- You should assume run time constant and not the problem size

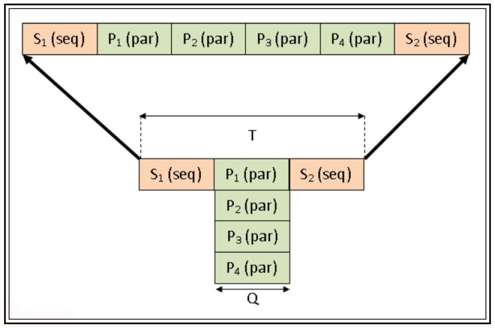
THE IDEA - 1



- Consider programs with a sequential part S, fixed, and a fixed time frame, $T = S + Q$.
- The speedup obtained by X is

$$\text{Speedup}_G(X) = \frac{S + 2Q}{S + Q} = 4/3$$

BUT, YOU CAN GET MORE "MILEAGE"...



- The speedup obtained by X is

$$\text{Speedup}_G(X) = \frac{S + 4Q}{S + Q} = 6/3$$

THE FORMULA

- Consider programs, with a sequential part S, fixed, and a fixed time frame, $T = S + Q$.
- Then the speedup by using n processors according to Gustafson (and Barsis) is:

Gustafson-Barsis' law

By executing X on n processors, the speedup is:

$$\text{Speedup}_G = \frac{S + nQ}{S + Q}$$

- With $n \to \infty$ the speedup is unbounded!

PLAN

1. PARALLELISM AND SCALABILITY
    - A hot topic
    - Scalability: "how-to"
2. COMPUTER SCIENCE LAWS
    - Amdahl's Law
    - Gustafson-Barsis's Law
3. SCALABILITY OF MULTICORE PROCESSORS
    - Technological motivations
    - Amdahl's law and Multicore Processors
4. CONCLUSIONS

PLAN

1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: "how-to"
2. COMPUTER SCIENCE LAWS
   - Amdahl's Law
   - Gustafson-Barsis's Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl's law and Multicore Processors
4. CONCLUSIONS

MOORE'S LAW

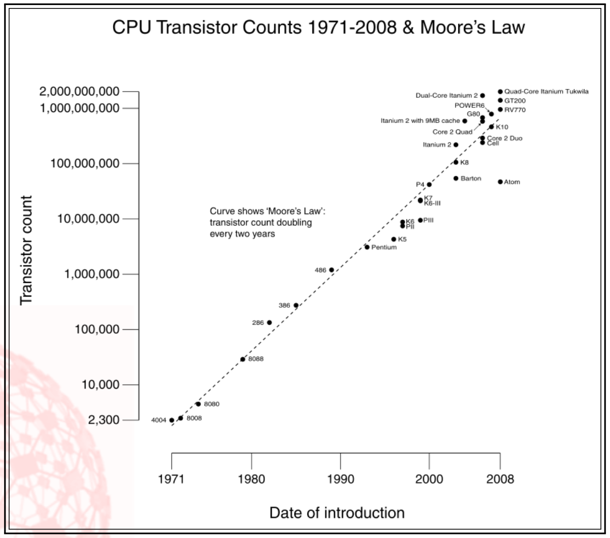
- Another, frequently cited, law in CS:

Moore's law

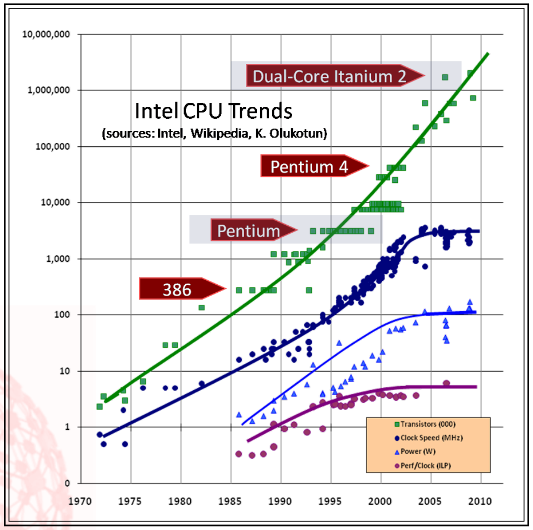
The number of transistors per chip tends to double every 2 years.

- The technological drive of our times: our desktop costs few hundreds euros and is as powerful as supercomputers of few years ago.

MOORE'S LAW



EVERYTHING GROWS (OR NOT!?)

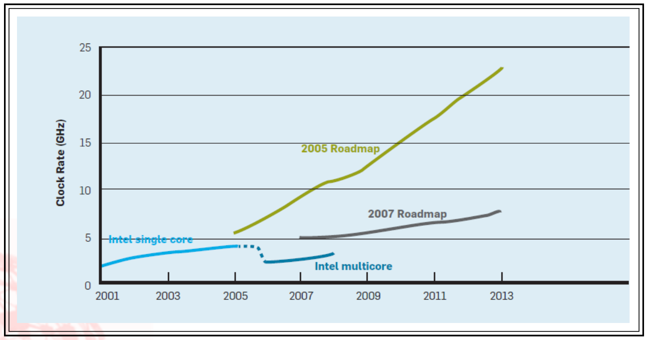


The problem: thermal noise

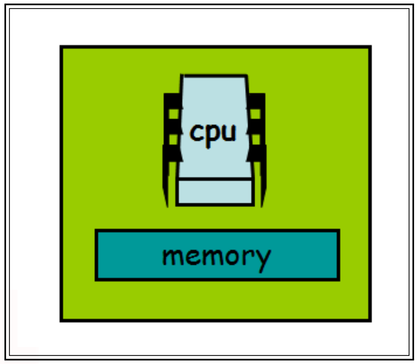
- Thermodynamics affect that perturbates Moore's law
- Critical if connected to:
    - low woltage power (to bound cooling problems)
    - increase frequency clock (to improve performances)
- Effects below the technology with 40 nm:
    - we hit it: Core i7 Intel is 45 nm technologies and prototypes reach 32 nm
- Practically: you can't have "*many*" transistors on a processor, that are "*easy to cool*" and "*quick*": you must give up one choice.

How changes are reflected

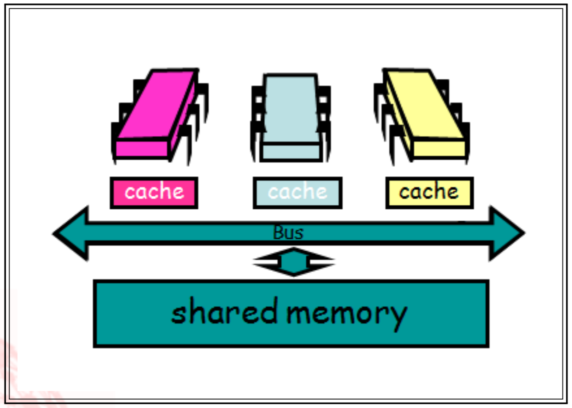
Between 2005 and 2007, here is how the "vision" of International Roadmap for Semiconductors changed



From single processor desktop...

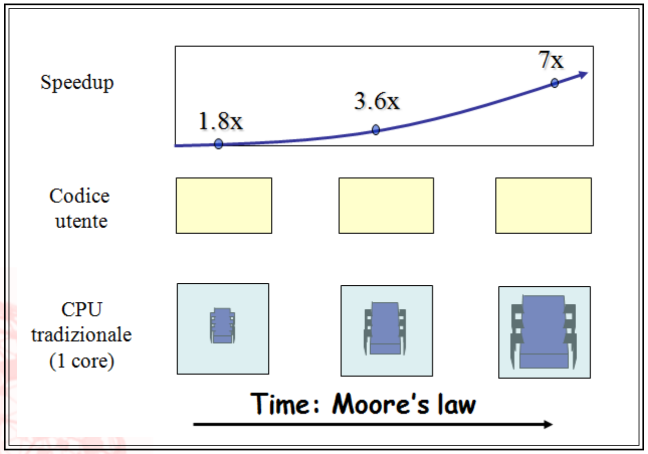


...to desktop with more processors



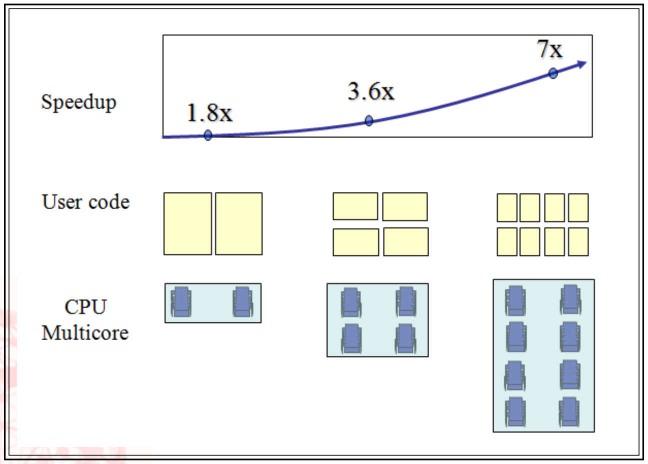
The current problem - 1

The situation with a single processor (core)



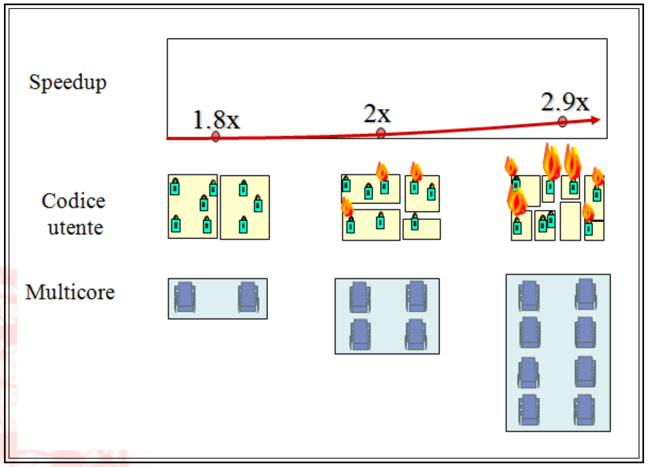
The current problem - 2

What we would like to be true



The current problem - 3

The harsh reality: balancing load is hard and you can create hot-spots



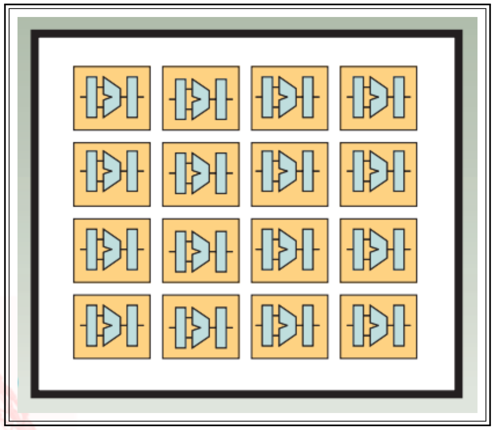
"Free lunch is over"

- Until now, technology improvements were automatically reflected on software performances
  - CPU with faster clock would execute the same code quicker
- Now, technology packs more and more transistors but organized in cores
- ...that, to be used efficiently, need parallel software

DESIGNER DILEMMA

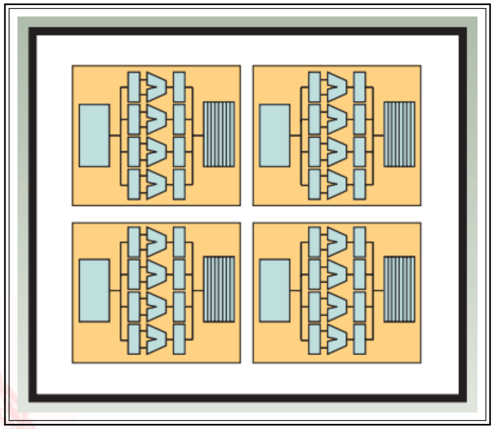
- Assume that technological constraints bound the number of transistors on a multicore chip
- The dilemma: how to organize them? Many cores of small capacity or fewer cores of large capacity?
- The model: assume that on a chip you can place at most n *Base Core Equivalents* (BCE) of computational power 1
- Area of r BCEs can be used to obtain a processor with performances $perf(r)$
- In general, $perf(r)$ is sublinear; usually $perf(r) = \sqrt{r}$ (Pollack rule)
- "*Amdahl's Law in the Multicore Era*, M.D. Hill, M.R. Marty, IEEE Computer 41(7), Nov. 2008.

SYMMETRIC MULTICORE HIGHLY PARALLEL



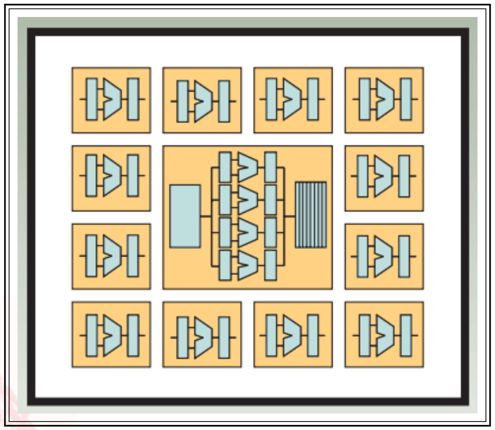
PicoChip, Connex Machine, Tilera (TILE64)

SYMMETRIC MULTICORE LOWLY PARALLEL



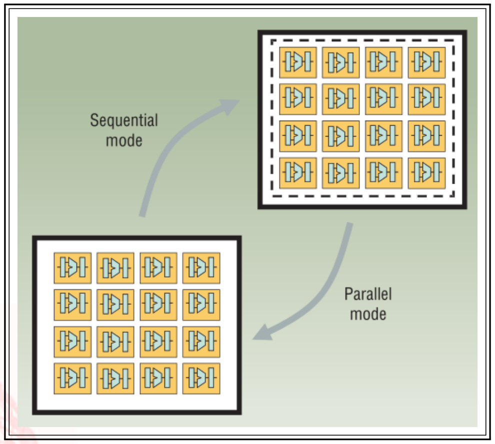
Intel, AMD

ASYMMETRIC MULTICORE



IBM/Sony Cell, Intel IXP

DYNAMIC MULTICORE



IBM/Sony Cell, Intel IXP

PLAN

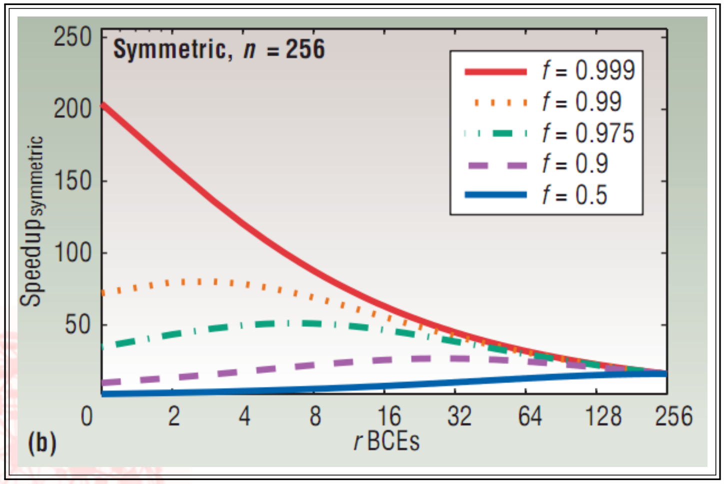
1. PARALLELISM AND SCALABILITY
   - A hot topic
   - Scalability: "how-to"
2. COMPUTER SCIENCE LAWS
   - Amdahl's Law
   - Gustafson-Barsis's Law
3. SCALABILITY OF MULTICORE PROCESSORS
   - Technological motivations
   - Amdahl's law and Multicore Processors
4. CONCLUSIONS

AMDAHL'S LAW FOR SYMMETRIC MULTICORE

- Speedup depends on the parallelizable part of the program, f, by resources n on chip (in BCEs) and by resources dedicated to each core (r BCE)
- There are n/r core, each with performance $perf(r) = \sqrt{r}$
- Amdahl's law, in this case, is

$$\text{Speedup}_{symm}(f, n, r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{perf(r) \cdot n/r}}$$

PERFORMANCES: SYMMETRIC MULTICORE

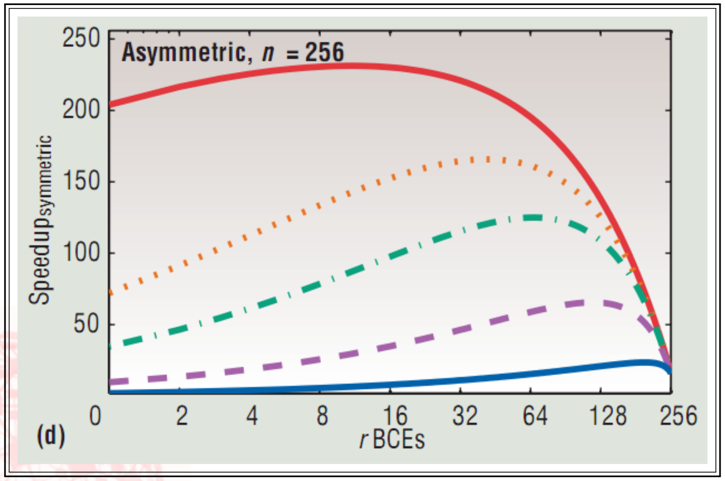


Amdahl’s law for asymmetric multicore

- Speedup depends on the parallelizable part of the program, f, by resources n on chip (in BCEs) and by resources dedicated to each core (r BCE)
- For the asymmetric multicore, a processor has more resources (r) and there are $n - r$ core with 1 BCE each
- In total $1 + n - r$ core, with different performances
- In the sequential part of the program, we can use the largest (r BCEs) core.
- In the parallel part, we can use all the cores (each with its performance)
- Amdahl’s law, in this case, is:

$$\text{Speedup}_{asymm}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{perf(r)+n-r}}$$

Performances: Asymmetric Multicore

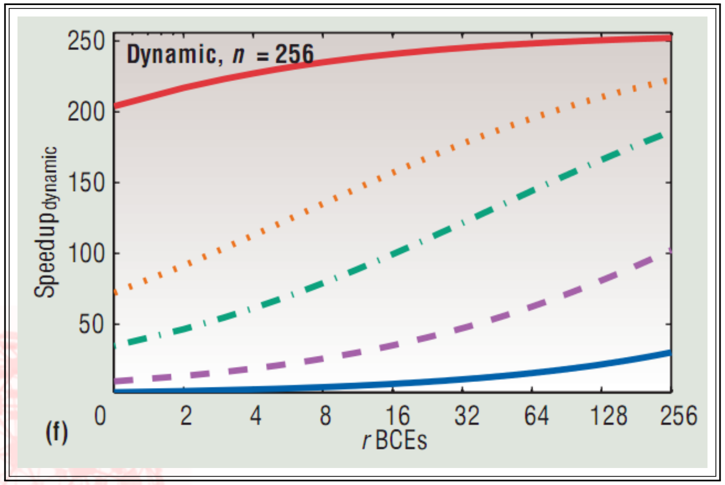


Amdahl’s law in the dynamic case

- Speedup depends on the parallelizable part of the program, f, by resources n on chip (in BCEs) and by resources dedicated to each core (r BCE)
- If it is possible to exploit each core (with multithread, for example) then each processor can be both a single processor (with r BCEs), in sequential, and n processors with 1 BCE of processing power
- In the sequential part, we can use the “largest” core (r BCE)
- In the parallel part, we can use all the n cores
- Amdahl’s law, in this case, is:

$$\text{Speedup}_{asymm}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{n}}$$

Performances: Dynamic Multicore

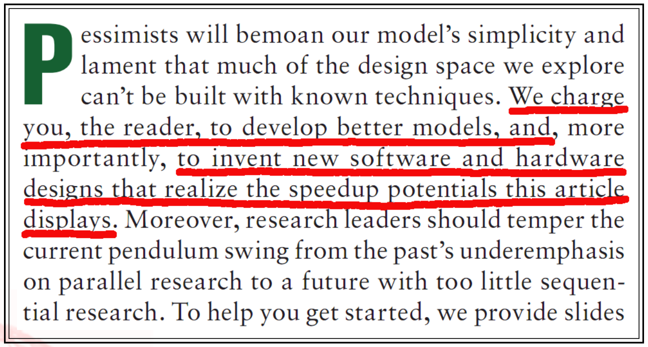


Comments

- Software is not infinitely parallel/sequential
- Data movements and tasks add overhead
- Scheduling on asymmetric/dynamic can be more costly than on symmetric
- "Learning curve" for programmers
  - More costly to develop parallel software than sequential software
  - With asymmetric, double (at least) the number of platform to develop software on

---

The conclusions of the paper



Pessimists will bemoan our model's simplicity and lament that much of the design space we explore can't be built with known techniques. We charge you, the reader, to develop better models, and, more importantly, to invent new software and hardware designs that realize the speedup potentials this article displays. Moreover, research leaders should temper the current pendulum swing from the past's underemphasis on parallel research to a future with too little sequential research. To help you get started, we provide slides

---

Plan

What lies ahead

- Insights into the Amdahl's law for multicore and relative performances
- Energy efficiency to be considered, as well as the architecture of multicore
- The "challenge" of Murray-Hill taken
- "Laws" are often oversimplifying: can we take into account also other costs (such as synchronization) ?