4: Scalability and Energy in Manycore Processors

Seminars in *Scalable Computing*

Vittorio Scarano

Università di Salerno

Dottorato di Ricerca in Informatica

PLAN

1. PARALLELISM AND SCALABILITY
    - Amdahl's Law
    - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
    - Motivations
    - Classification
    - The models
    - Evaluation of models
    - Power-equivalent models
    - Conclusions

PLAN

1. PARALLELISM AND SCALABILITY
    - Amdahl's Law
    - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
    - Motivations
    - Classification
    - The models
    - Evaluation of models
    - Power-equivalent models
    - 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
    - Amdahl's Law
    - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
    - Motivations
    - Classification
    - The models
    - Evaluation of models
    - Power-equivalent models
    - 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

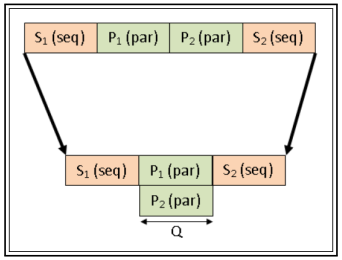
- The*speedup* S of X is the ratio between the sequential time to execute X and the parallel time (n processors) to execute X
- Let P be the part of X that can be parallelized
    - with n processors the parallel part takes time $\frac{P}{n}$ while the sequential takes time S
- Then the speedup is:

Amdahl's law

By executing X on n processors, the speedup is:

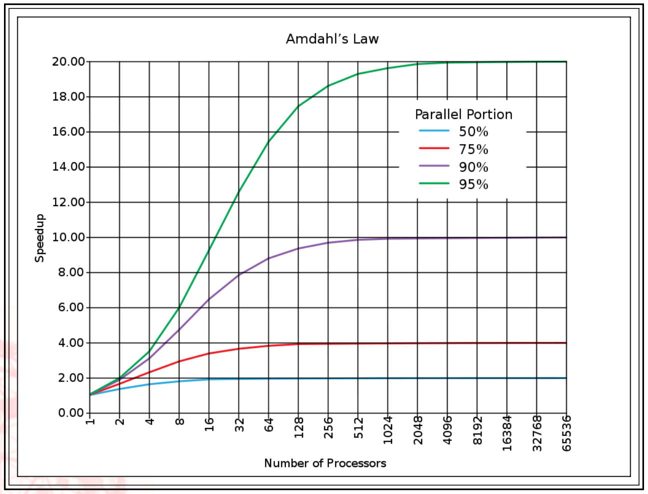
$$\text{Speedup}_A = \frac{S+P}{S+\frac{P}{n}}$$

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
- $\text{Speedup}_A(X) = \frac{(S+P)}{S+P/2} = 4/3$

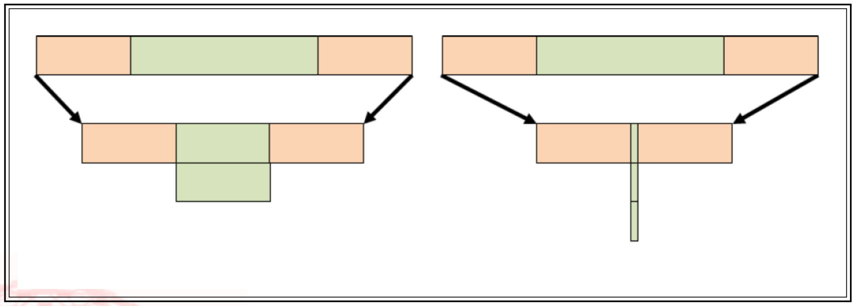
A DIAGRAM OF THE LAW



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
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

Laws and reality

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

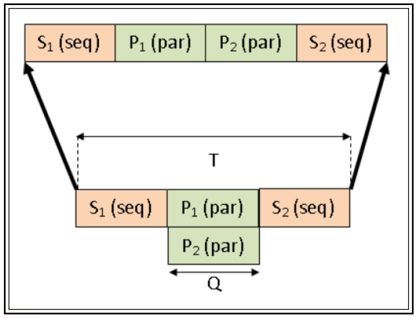
"Reevaluating Amdahl's Law" (Comm. of ACM, 1988)

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

---

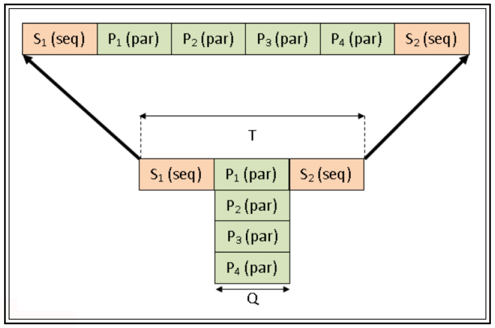
The idea - 1



- 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:

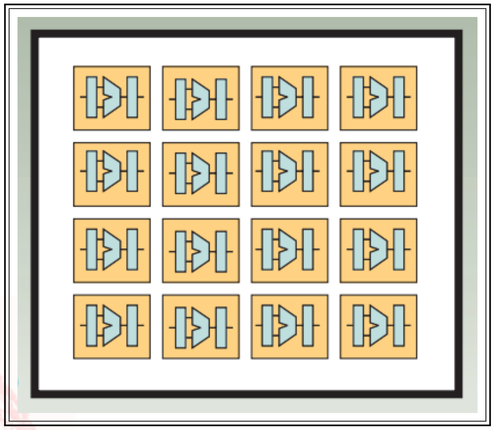
$$\text{Speedup}_G = \frac{S + nQ}{S + Q}$$

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

DESIGNER DILEMMA

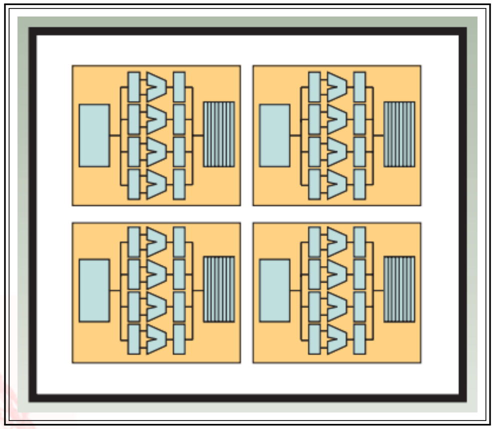
- Assume that technological constraints bound the number of transistors on a multicore chip
- The dilemma: how to organize them? Many cores of small capacity or fewer cores of large capacity?
- The model: assume that on a chip you can place at most n *Base Core Equivalents* (BCE) of computational power 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



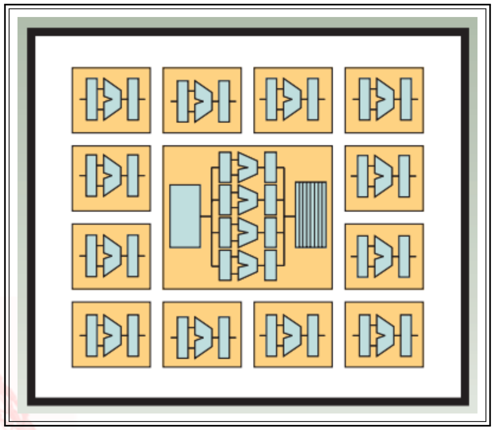
PicoChip, Connex Machine, Tilera (TILE64)

SYMMETRIC MULTICORE LOWLY PARALLEL



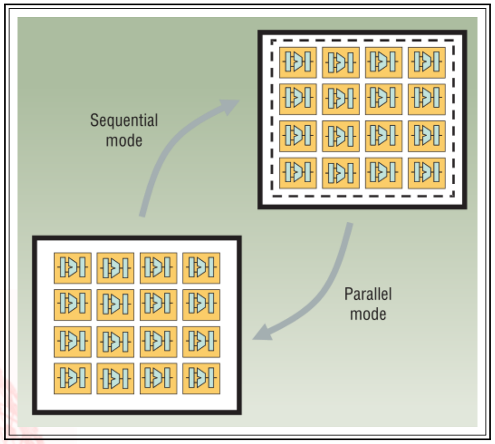
Intel, AMD

ASYMMETRIC MULTICORE



IBM/Sony Cell, Intel IXP

DYNAMIC MULTICORE



PLAN

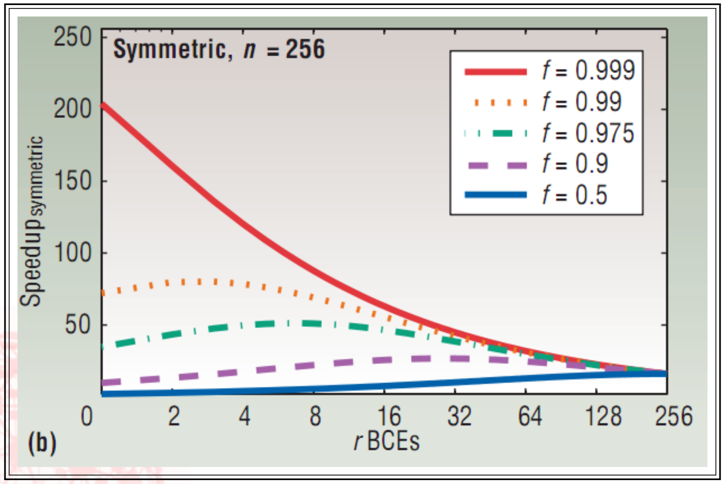
1. PARALLELISM AND SCALABILITY
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

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

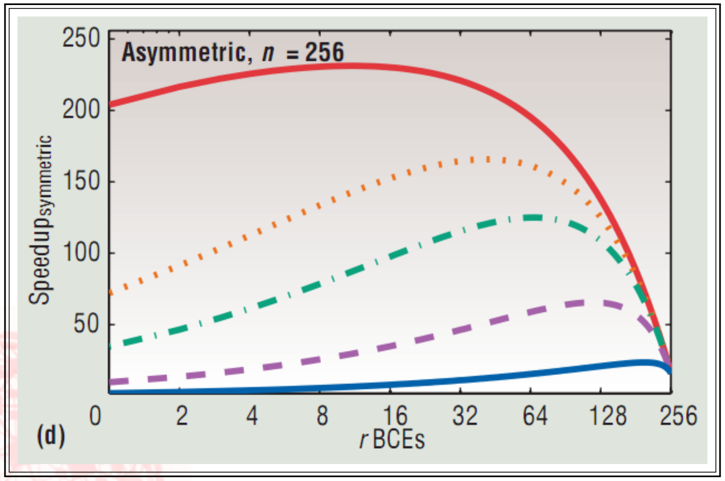


AMDAHL'S LAW FOR ASYMMETRIC MULTICORE

- 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

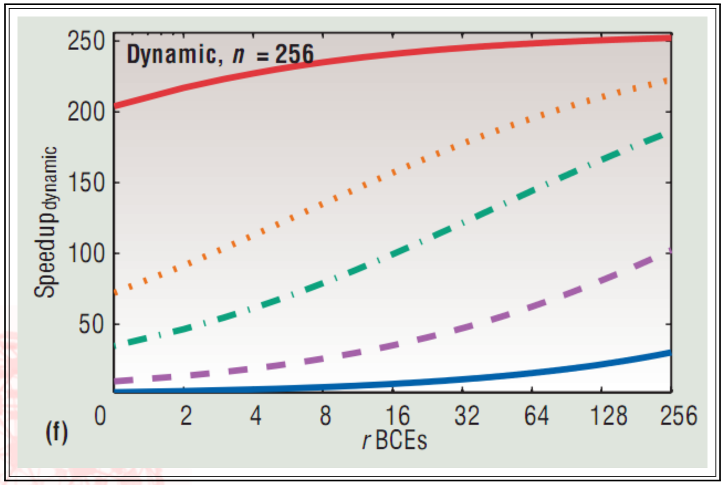


AMDAHL'S LAW IN THE DYNAMIC CASE

- 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

---

PLAN

1. PARALLELISM AND SCALABILITY
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

PLAN

1. PARALLELISM AND SCALABILITY
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

THE CHALLENGE

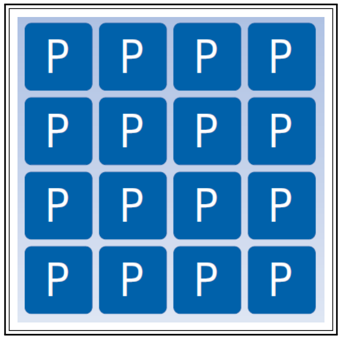
- Beyond performances, the architects face a Grand Challenge: energy efficiency
- Future many-core processors should not exceed their power budget
- The amount of power *each core* consumes will dictate the number of cores that will be placed on a die
- *Power scalability* according to Amdahl's law has to be studied to provide architectural guidelines and design insights

PLAN

1. PARALLELISM AND SCALABILITY
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

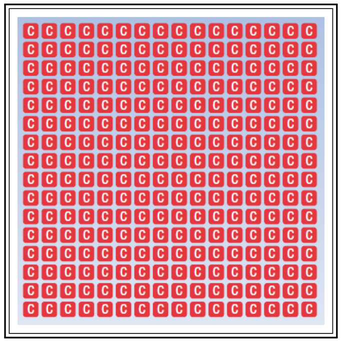
SYMMETRIC MANY-CORE P^*

- Replication of state-of-the-art superscalar processor
- Typical choice of mass-market processor vendors (Intel, AMD, etc.)
- Flexible
- General
- Cheap ☺



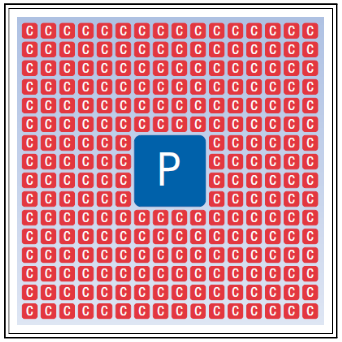
SYMMETRIC MANY-CORE (WITH SMALLER PROCESSORS) c^*

- Replication of smaller (and more power efficient) cores on a die
- Typical for embedded machines
- Performance of a single core are not as high as the state-of-the-art processor



ASYMMETRIC MANY-CORE $P + c^*$

- Mixing one state-of-the-art superscalar processor and many energy-efficient smaller processors
- Some examples: Sony-Toshiba-IBM Cell Broadband engine
- High parallelism but also high efficiency during the sequential part of the program



PLAN

1. PARALLELISM AND SCALABILITY
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

MODEL FOR P^* (1)

- The Amdahl's law for the maximum speedup is:

$$\text{Perf} = \frac{1}{(1-f)+\frac{f}{n}}$$

- Denote by $0 \leq k \leq 1$ the fraction of power consumed by a single core in idle step
- One (superscalar) processor consumes 1 if busy during a sequential computation, while the remaining $n-1$ consume $k(n-1)$
- The average power consumption is

$$W = \frac{(1-f)\times[1+(n-1)k]+\frac{f}{n}\times n}{(1-f)+\frac{f}{n}} = \frac{1+(n-1)k(1-f)}{(1-f)+\frac{f}{n}}$$

MODEL FOR P^* (2)

- Recall that the average power consumpion is

$$W = \frac{1+(n-1)k(1-f)}{(1-f)+\frac{f}{n}}$$

- We can evaluate *Performance per watt* i.e. the performance obtainable with the same cooling capacity

$$\frac{\text{Perf}}{W} = \frac{\frac{1}{(1-f)+\frac{f}{n}}}{\frac{1+(n-1)k(1-f)}{(1-f)+\frac{f}{n}}}$$

$$\frac{\text{Perf}}{W} = \frac{1}{1+(n-1)k(1-f)}$$

MODEL FOR P^* (3)

- We can also evaluate *Performance per Joule* as a measure of the performances achievable in the same battery lifecycle
  - 1 W = 1 J/s
- Considering that the time to run a parallel application (with respect to the normalized sequential time 1) is $(1-f)+\frac{f}{n}$ then ...
- ...the Performance per Joule is:

$$\frac{\text{Perf}}{J} = \frac{1}{\left((1-f)+\frac{f}{n}\right)\times(1+(n-1)k(1-f))}$$

MODEL FOR c^* (1)

- To accomodate arbitrarily sized cores, the variable s_c is introduced to measure the core performance (with respect to the state-of-the-art processor P)
- The Amdahl's law for the maximum speedup is now:

$$\text{Perf} = \frac{s_c}{(1-f) + \frac{f}{n}}$$

- $0 \leq w_c \leq 1$ is the *active* power consumption of a single (small) core with respect to an active state-of-the-art processor
- $0 \leq k_c \leq 1$ is the fraction of power consumed by a single (small) core in idle step with respect to the power consumed by the same core (that is w_c times that of a full processor)

MODEL FOR c^* (2)

- One processor consumes w_c if busy during a sequential computation, while the remaining $n-1$ consume $(n-1) \cdot k_c(w_c)$
- The time of sequential computation is $\frac{1-f}{s_c}$ while the parallel time is $\frac{f}{n \cdot s_c}$
- The average power consumption is, then,

$$W = \frac{\frac{(1-f)}{s_c} \times [w_c + (n-1)k_c w_c] + \frac{f}{n \cdot s_c} \times n w_c}{\frac{(1-f)}{s_c} + \frac{f}{n \cdot s_c}}$$

$$W = \frac{w_c + (n-1)k_c w_c (1-f)}{(1-f) + \frac{f}{n}}$$

MODEL FOR c^* (3)

- The average power consumpion is

$$W = \frac{w_c + (n-1)k_c w_c (1-f)}{(1-f) + \frac{f}{n}}$$

- We can evaluate *Performance per watt*

$$\frac{\text{Perf}}{W} = \frac{s_c}{w_c + (n-1)k_c w_c (1-f)}$$

- ...and the *Performance per Joule*

$$\frac{\text{Perf}}{J} = \frac{s_c}{(1-f) + \frac{f}{n}} \times \frac{s_c}{w_c + (n-1)k_c w_c (1-f)}$$

MODEL FOR $P + c^*$ (1)

- One single state-of-the-art core P and $n-1$ small, energy-efficient cores

- The Amdahl's law for the maximum speedup is now:

$$\text{Perf} = \frac{1}{(1-f) + \frac{f}{(n-1)s_c}}$$

- Notice that the P core is inactive during the parallel phase
  - different assembler code, different HW, complex synchronization
- 1 is the *active* power consumption of P
- $0 \leq k_c \leq 1$ is the fraction of power consumed by a single (small) core in idle step with respect to the power consumed by the same core (that is w_c times that of a full processor)

MODEL FOR $P + c^*$ (2)

- One (superscalar) processor consumes 1 if busy during a sequential computation, while the remaining $n-1$ consume $(n-1) \cdot k_c(w_c)$
- During parallel computation, processors consume $k + (n-1)w_c$
- The time of sequential computation is $1-f$ while the parallel time is $\frac{f}{(n-1) \cdot s_c}$
- The average power consumpion is, then,

$$W = \frac{(1-f) \times \left(1 + (n-1)k_c w_c\right) + \frac{f}{s_c}\left(\frac{k}{n-1} + w_c\right)}{(1-f) + \frac{f}{(n-1)s_c}}$$

MODEL FOR $P + c^*$ (3)

- The average power consumpion is, then,

$$W = \frac{(1-f) \times \left(1 + (n-1)k_c w_c\right) + \frac{f}{s_c}\left(\frac{k}{n-1} + w_c\right)}{(1-f) + \frac{f}{(n-1)s_c}}$$

- We can evaluate *Performance per watt*

$$\frac{\text{Perf}}{W} = \frac{1}{(1-f) \times \left(1 + (n-1)k_c w_c\right) + \frac{f}{s_c}\left(\frac{k}{n-1} + w_c\right)}$$

- ...and the *Performance per Joule*

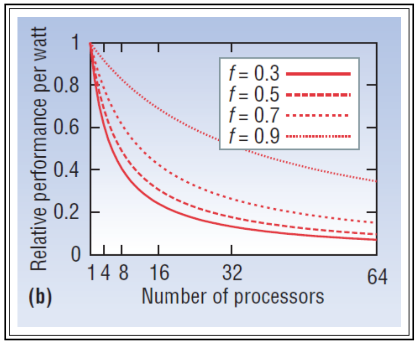
$$\frac{\text{Perf}}{J} = \frac{1}{\left((1-f) + \frac{f}{(n-1)s_c}\right)} \times \frac{1}{(1-f) \times \left(1 + (n-1)k_c w_c\right) + \frac{f}{s_c}\left(\frac{k}{n-1} + w_c\right)}$$

PLAN

1. PARALLELISM AND SCALABILITY
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

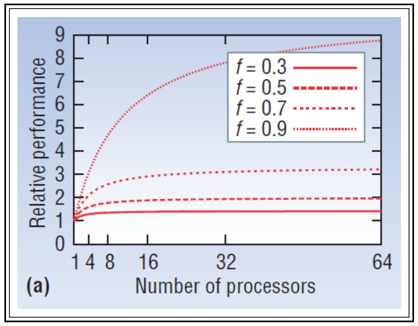
EVALUATION OF P^* (1): PERF/WATT

- Parallel execution on P^* consumes more energy than the sequential execution
- In case of $f = 1$ (maximum parallelism) we can get the max Perf/Watt



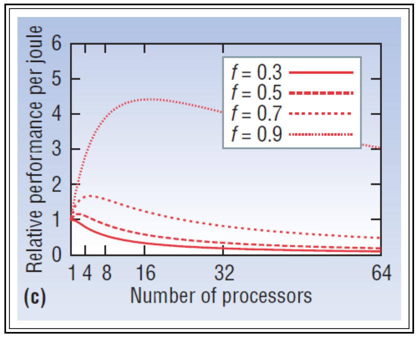
EVALUATION OF P^* (2): PERFORMANCES

- This happens because performance does not scale linearly . . .
- . . . while the amount of power consumption while idle, scales linearly



EVALUATION OF P^* (3): PERF/JOULE

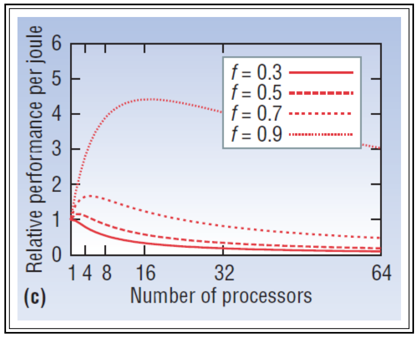
- Spanning threads on different cores is more power efficient than multi-threading
- Maximizing parallelization and balancing workload is crucial *also* for power-supply efficiency and extended battery life (not only for performances)
- When $f = 0.9$ a 16 core P^* can achieve four-fold speedup than a single-core, with the same energy



$k = 0.3$

EVALUATION OF P^* (4): PERF/JOULE

- When an application is only half-parallel ($f = 0.5$) then with 8 processors we consume more than the correspondent single processor
- Optimal number of cores: if possible, with a 32-cores architectures, with an application with $f = 0.9$ it is efficient to shut down the last 15 cores



$k = 0.3$

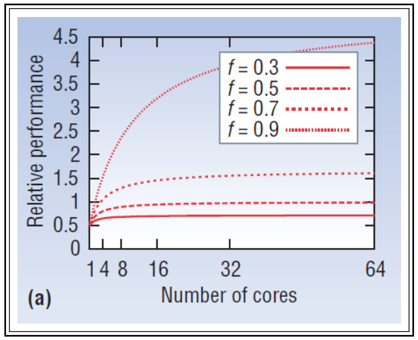
EVALUATION OF c^* (1): SETTINGS

- We need to model the relationship between each core's performances and its size
- Pollack's rule: given the same technology, a processor provides:
  - 1.5 to 1.7 times higher performance
  - 2 to 3 times the die area

  compared to its previous generation counterpart
- If transistors grow by a factor of T, then the performances grow by a factor of $\sqrt{T}$
- Power consumption is proportional to the number of transistors
- Technological parameters:
  - each core c has 1/4 the transistors of $P \Rightarrow w_c = 0.25$
  - each core has efficient 1/2 as $P \Rightarrow s_c = 0.5$
  - fraction of power is at 20% $\Rightarrow k_c = 0.2$

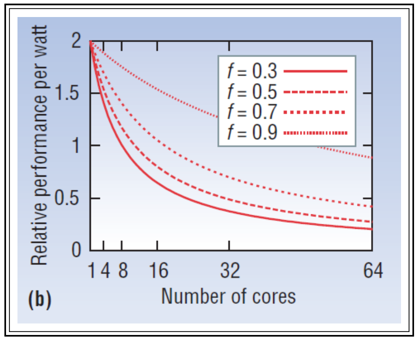
EVALUATION OF c^* (2): PERFORMANCES

- Maximal speedup of c^* is not as high as P^*'s (that could reach upto 8-9 times)
- Efficiency of each core is lower, which limits efficiency into the sequential part



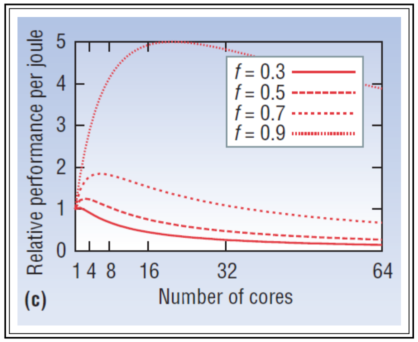
EVALUATION OF c^* (3): PERF/WATT

- With fewer cores, c^* consumes less than a single-core processor: perf-to-power ratio of a small core c is better than P's
- As the number of cores increases, the energy consumption is higher that that of a single core



EVALUATION OF c^* (4): PERF/JOULE

- Performances are not very good
- Unless the application is embarassingly parallel ($f = 0.9$)



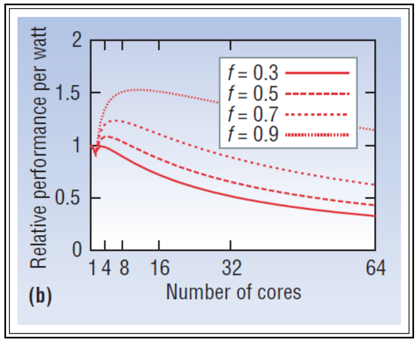
$k = 0.3$

EVALUATION OF $P + c^*$ (1): SETTINGS

- Same setting as the c^* model
- Technological parameters:
  - each core c has 1/4 the transistors of $P \Rightarrow w_c = 0.25$
  - each core has efficient 1/2 as $P \Rightarrow s_c = 0.5$
  - fraction of power is at 20% $\Rightarrow k_c = 0.2$

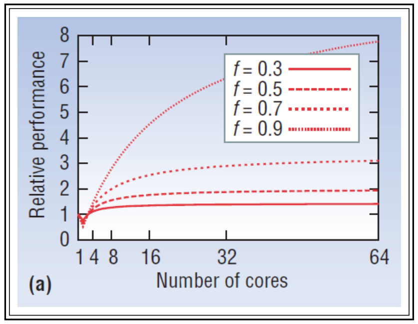
EVALUATION OF $P + c^*$ (2): PERF/WATT

- Unlike other models, an optimal number of cores exists, that consumes the least amount of energy
- But after the peak, it becomes worse than the single core: with few additional cores, the performance increase dominates over additional power overhead (see performance)



EVALUATION OF $P + c^*$ (3): PERFORMANCES

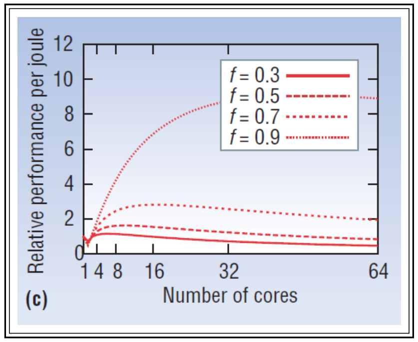
- After a certain value, the performance increase is saturated (does not grow anymore) and can't balance the energy overhead



EVALUATION OF $P + c^*$ (4): PERF/JOULE

Good results, because of:

- low-latency sequential execution (state-of-the-art processor P)
- energy efficient execution (many energy-efficient small cores, c^*)



$k = 0.3$

PLAN

1. PARALLELISM AND SCALABILITY
   - Amdahl's Law
   - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
   - Motivations
   - Classification
   - The models
   - Evaluation of models
   - Power-equivalent models
   - Conclusions

THE MOTIVATIONS

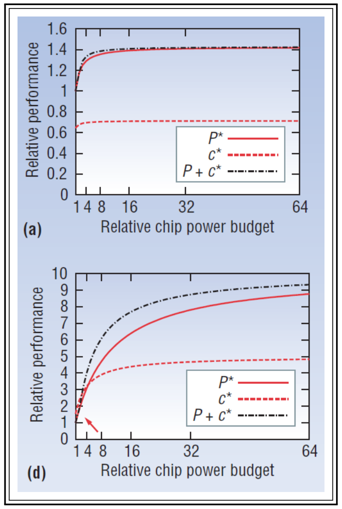
- Because the limited power budget is the most critical design consideration
  - or at least one of the topmost important criteria
- Two main limiting factors:
  - power supply: proportional to the energy cost for sustaining machines in data center
  - power density: proportional to the mechanisms for thermal control (extra complexity and cost)
- Example: given 160W max power budget, one can use it to interate 8 full processors (requiring 20W each) or 32 efficient cores (requiring 5W each) on a single die
- ⇒ the need of a power-equivalent model

THE POWER-EQUIVALENT MODEL

- Let W_{budget} be the power budget assigned to a single die
- Let n_{P^*} be the number of full processors, each one consuming 1
- Let n_{c^*} be the number of small efficient cores, each one consuming w_c
- Let n_{P+c^*} be the number of 1 processor + small efficient cores, each one consuming w_c
- On a die, you can consume:
  - at most n_{P^*} i.e. $n_{P^*} = W_{\text{budget}}$
  - at most $n_{c^*} \cdot w_c$ i.e. $n_{c^*} = W_{\text{budget}}/w_c$
  - at most $1 + (n_{P+c^*} - 1)w_c$ i.e. $n_{P+c^*} = 1 + \frac{W_{\text{budget}}{}^{-1}}{w_c}$

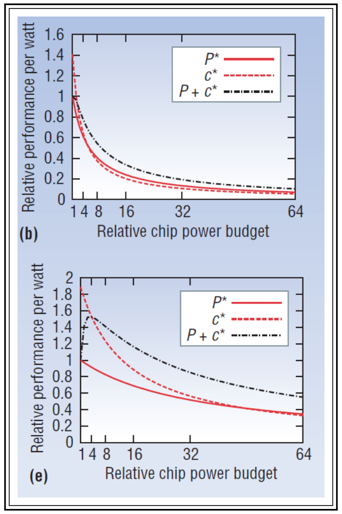
EVALUATION OF THE MODEL: PERFORMANCES

- The performance of $P + c^*$ and P^* are the highest for small parallelism ($f = 0.3$) but $P + c^*$ is more efficient, since it can have more core, when parallelism is higher ($f = 0.9$)
- If very small energy budget is available and a a lot of parallelism is available, c^* is better (see the arrow)



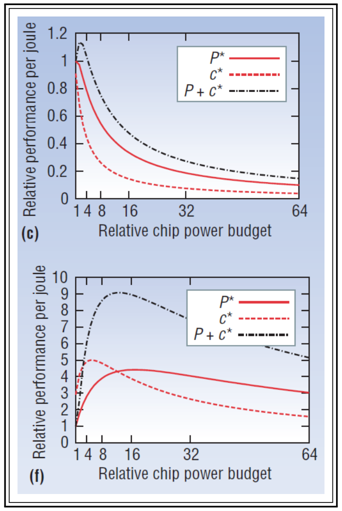
EVALUATION OF THE MODEL: PERF/WATT

- With small power budget, c^* is better
- With large budget, $P + c^*$ is the most efficient: with small budget, c^* outperforms because of its parallel performances, when more budget allows a sequential P, then $P + c^*$ wins



EVALUATION OF THE MODEL: PERF/JOULE

- The performances of $P + c^*$ is the highest for small parallelism ($f = 0.3$) and large parallelism ($f = 0.9$)
- Similar trends are confirmed by other simulations with different values



PLAN

1. PARALLELISM AND SCALABILITY
    - Amdahl's Law
    - Gustafson-Barsis's Law
2. AMDAHL'S LAW AND MULTICORE PROCESSORS
3. ENERGY EFFICIENCY
    - Motivations
    - Classification
    - The models
    - Evaluation of models
    - Power-equivalent models
    - Conclusions

WHAT DOES ENERGY EFFICIENCY SUGGEST?

- The analysis demonstrates that symmetric many-core processors lose energy efficiency as the number of cores increases.
- An alternative is the many-core alternative, integrated with a full-blown processor.
- If the amount of parallelism available in an application prior to execution is known
    - it is possible to find the optimal number of active cores for maximizing performance
    - for a given cooling capacity and energy in a system