HPES
High Performance
Embedded Systems

EEE4120F

High Performance Embedded Systems

Lecture 8:

Amdahl's Law for the Multicore Era & Base Core Equivalents (BCE)

Presented by
Simon Winberg

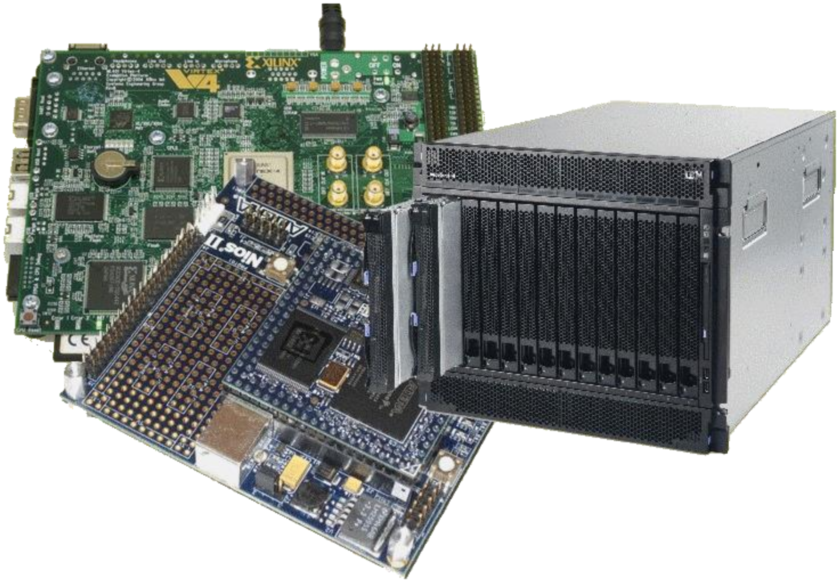
Prescribed reading: L08 Hill and Marty - Amdahl Law in Multicore Era.pdf
(see READINGS on Vula)

Planned to be double period lecture

Lecture Overview

- Amdahl’s Law for the Multicore Era
- Base Core Equivalents (BCEs)
- Calculating performance using BCEs

This presentation focuses on explaining the following prescribed reading: “L08 Hill and Marty - Amdahl Law in Multicore Era.pdf”
(see READINGS on Vula)
NB: relevant for quiz next Thursday



You have arrived….

… At a point to understand …

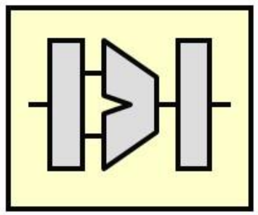
The BCE!

The Base Core Equivalent (BCE)

So let’s get into it…

Designing Multicore Chips

- Designers contend with single-core design options
  - Wakeup (of sleeping cores) – How? Economical methods?
  - Instruction fetch – From where? Latencies?
  - Instruction Decoding – CISC vs. RISC trade-offs
  - Variable clock speeds – esp. power saving modes
  - Execution unit – Pipelined? Monolithic? Shared components?
  - Execution modes (e.g. system, user, interrupt)
  - Load/Save queue(s) & data cache
- Additional degrees of freedom for design of parallel architectures
  - How many cores? Size & features of each?
  - Shared caches? Cache levels?
  - Memory interfacing? How many memory banks? Where to place them?
  - On-chip interconnects: bus structures, switching fabric, ordering and priority of cores/devices?



Base Core Equivalent (BCE)

EEE4120F

BCE Abstraction

- It all gets a bit complicated to keep track of the various umpteen types of processors that might be used
- Therefore to theorize and predict performance expectations of a conceptualized* heterogeneous system, the concept of a Base Core Equivalent is proposed…

* i.e. where you might be toying with a variety of different type of processor solutions for a platform

Calculating performance using BCEs

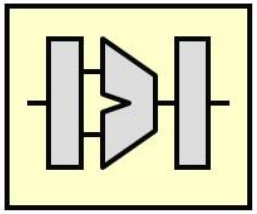
- A great many papers on the subject of Amdahl’s law
- Many mention the term BCE
- BCE = Base Core Equivalent, which is:
  - A single processing core in a multicore processor design

BCE: Simple Multicore Hardware Model

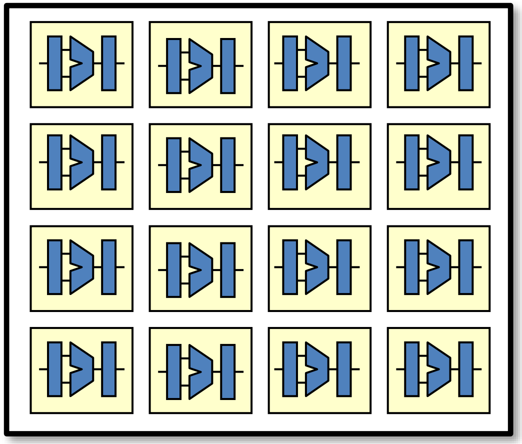
- BCE: An abstracted hardware model to complement Amdahl's software model
- The simplified model…
- Chip Hardware, abstracted as
  - Multiple Cores (with L1 caches)
  - Supporting architecture (L2/L3 cache banks, interconnect, pads, etc.)
- Resources for Multiple Cores Bounded
  - Due to area, power, cost, or multiple factors…
  - Bound of N resources per chip for cores
  - Bound in terms of Power and Area

1x BCE vs. 1x *R*-BCE vs. *N*x *R*-BCE

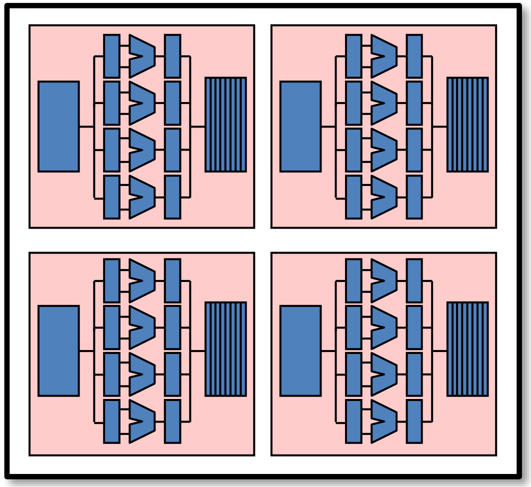
- Remember that the BCE is equivalent to a basic single core processor



= 1x 1-BCE



= 16x 1-BCE



= 4x 4-BCE

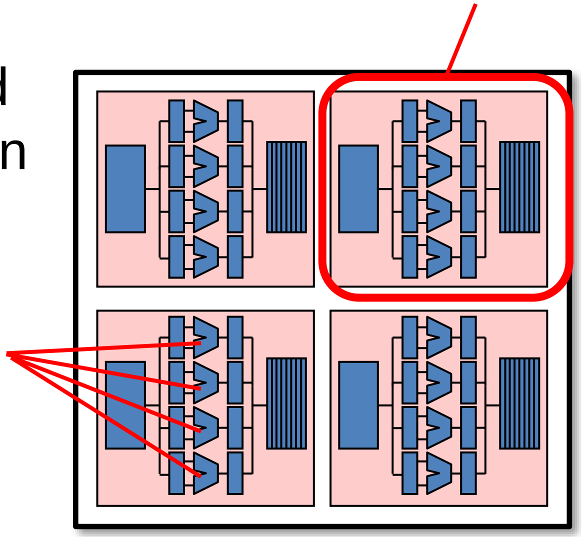
It’s an abstraction, and likely not very precise. For example a 16x4-BCE may have faster comms between cores in same block and slower comms between cores in different blocks, whereas the 16x1-BCE might have the same speed comms between all cores.

Let me clarify that expression...

- NxR-BCE means:
  - *N* = Number of megacores (or equivalent 'core groups') on a chip
  - *R* = Number of cores equivalent to a megacore (or num. cores in a core group)

A megacore may have – or considered to have – its own local cache (or considered as a group of cores that shares a common cache)

num megacore (N)



number of basic BCE cores (R) equivalent to this megacore

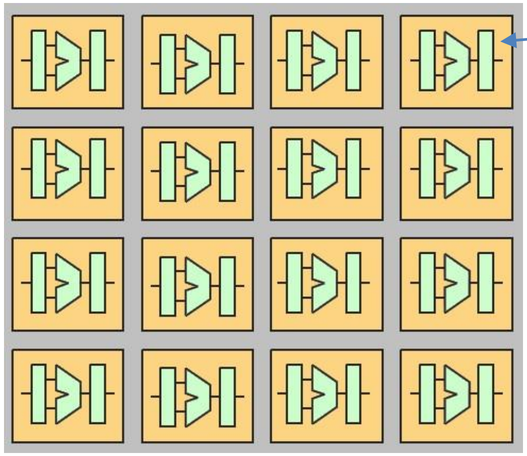
4x4-BCE

Take Note…

A *R*-BCE megacore is not necessarily physically having *R* x BCE smaller cores implemented and working together …

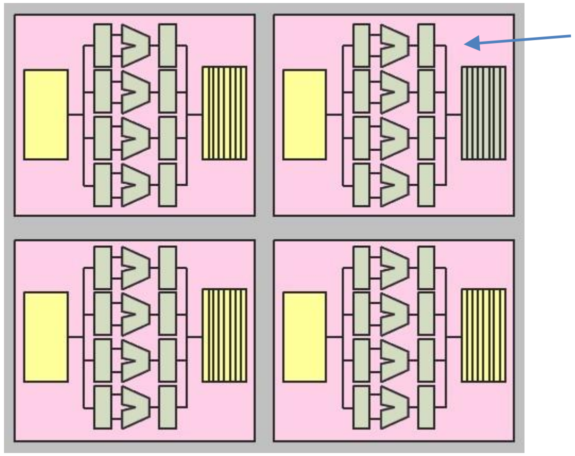
It is rather a core that has a different design and works *sort-of* equivalently to *R* x BCE cores.

Example processor structures Symmetric vs. Asymmetric multicores



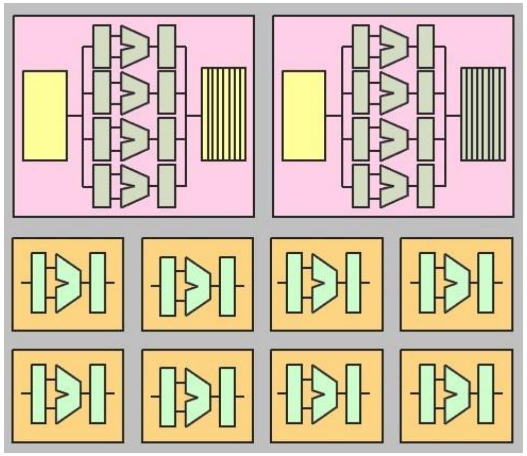
a BCE

Symmetric multicore 16 one-BCE cores



core with 4-BCE (I call them megacores to emphasize it equates to multiple BCEs)

Symmetric multicore 4 x 4-BCE megacores



Asymmetric multicore comprising 2 x 4-BCE cores + 8 x 1-BCE cores

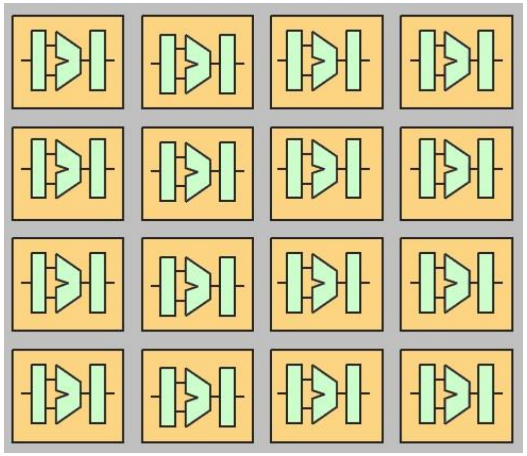
Clearly can become increasingly more difficult to achieve very accurate equivalences due to the architecture structures and things like memory latencies and inter-core comms.

Symmetric core model

Variables:

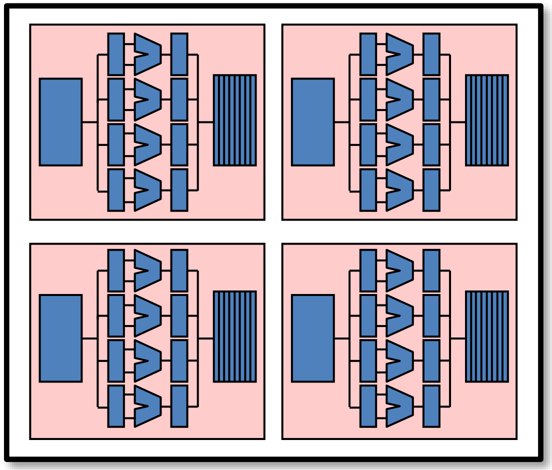
N implemented cores per chip, **n** = number BCEs available

each ***N*** *core consumes* ***R*** BCEs resources



16 x 1-BCE

N = 16, R=1 (n=16)



Symmetric four 4-BCE megacores

4 x 4-BCE, R=4, N=4 (n=16)

perf(*R*) = performance (speedup) of 1x *R*-BCE over 1x 1-BCE

Always perf(*R*) <= *R* (i.e. diminishing returns on increased parallelism)

Good heuristic: perf(R)=sqrt(R)

Modelling Performance of Symmetric Multicore Chips

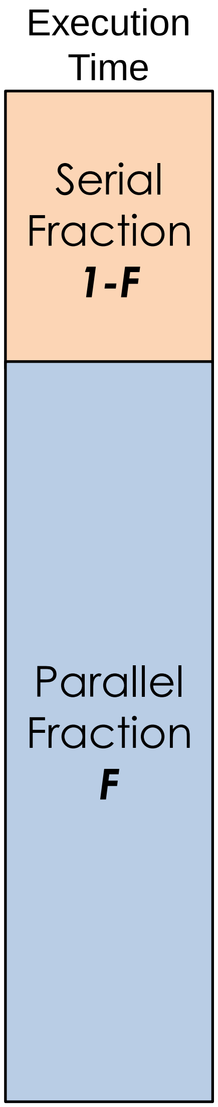
- ***N*** implemented cores per chip, ***n*** = resources in 1-BCE
- ***R*** BCEs per megacore
- Parallel Fraction of execution is ***F***
- Assume **symmetric** → all cores identical
- Serial Fraction is **1-*F*** so uses 1 megacore at rate perf(R)
- Therefore: Serial time = $(1 - F) / \text{perf}(R)$
- Parallel Fraction uses n/R megacores at rate perf(R) each
- Parallel time = $F / (\text{perf}(R) * (n/R)) = F * R / \text{perf}(R) * n$
- Therefore, w.r.t. one base core:

$$\text{Symmetric Speedup} = \frac{1}{\dfrac{1 - F}{\text{perf}(R)} + \dfrac{F * R}{\text{perf}(R) * n}}$$

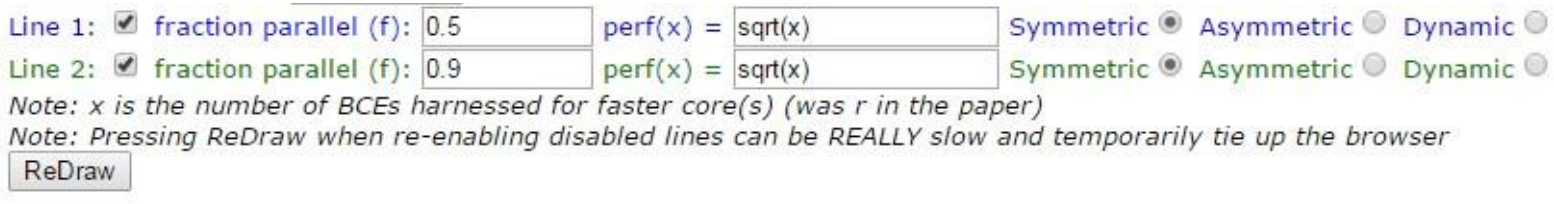
- **Implications?..**

Pretty cool don't you think! ☺

Enhanced Cores speed Serial & Parallel



Calculate performance…

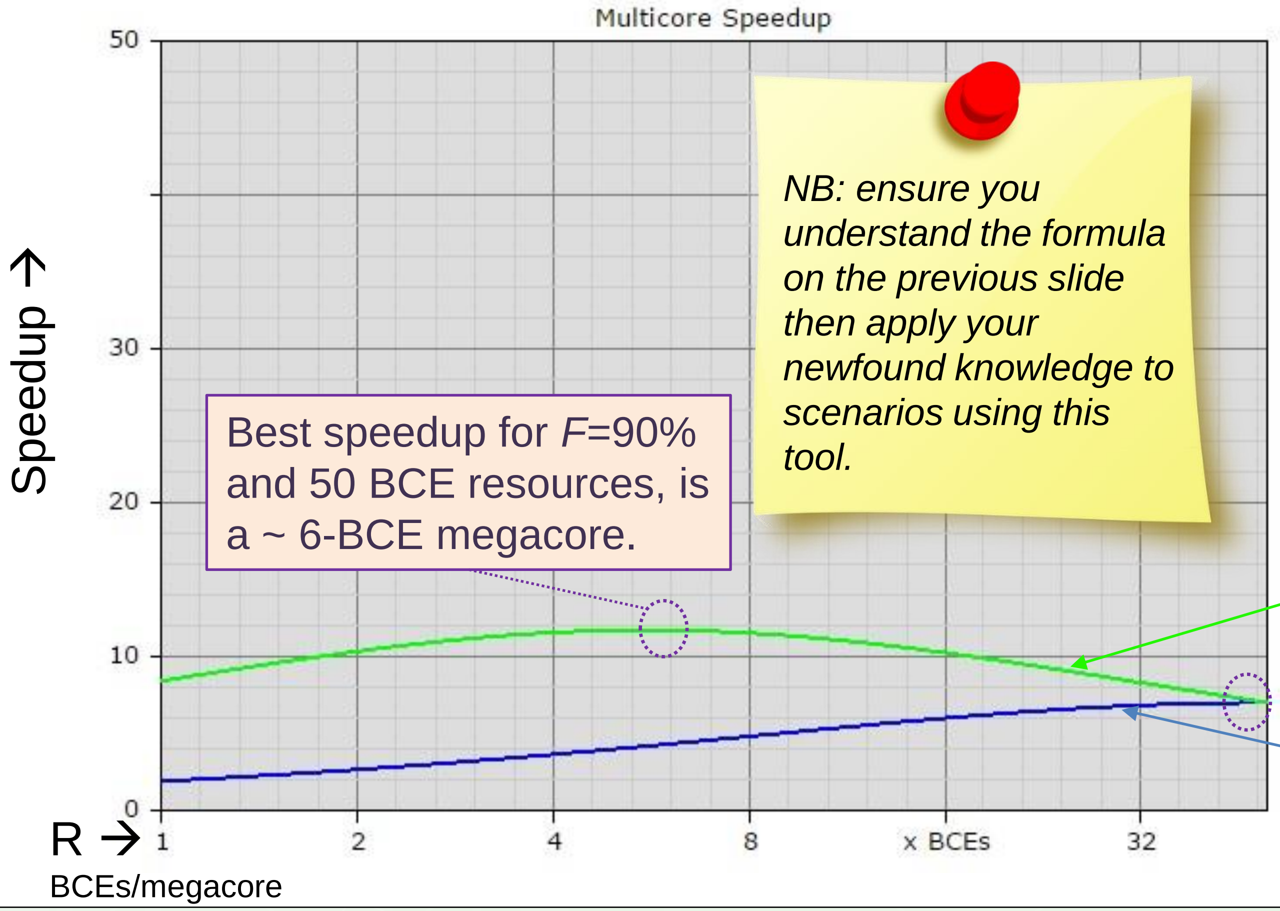


Line 1: ☑ fraction parallel (f): 0.5 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

Line 2: ☑ fraction parallel (f): 0.9 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

Note: x is the number of BCEs harnessed for faster core(s) (was r in the paper)

Note: Pressing ReDraw when re-enabling disabled lines can be REALLY slow and temporarily tie up the browser



Best speedup for *F*=90% and 50 BCE resources, is a ~ 6-BCE megacore.

NB: ensure you understand the formula on the previous slide then apply your newfound knowledge to scenarios using this tool.

Model up to *n*=50 BCEs

Parallel fraction *F*=90%

Parallel fraction *F*=50%

Asymmetric (Heterogeneous) Multicore Chips

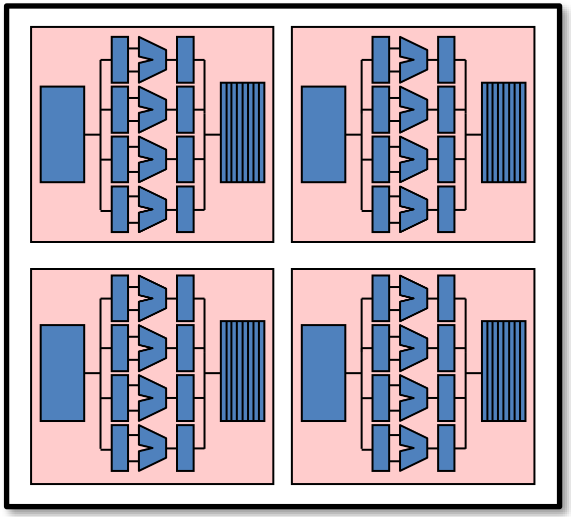
- Symmetric Multicore required all cores equal
- Let's enhance some (not all) of the cores in the processor chip…

- For Amdahl's simplified assumptions
  - One Enhanced Core
  - Others are base cores

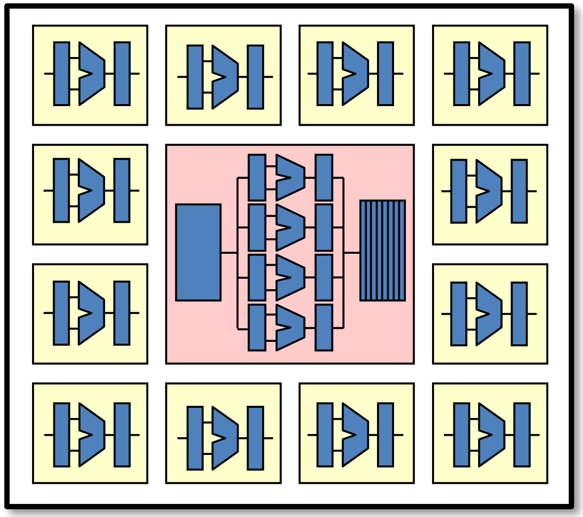
- How does this effect our hardware model?

Asymmetric Multicore Chips

- Chip bounded to total of *n* BCEs
- One *R*-BCE megacore leaves (*n-R)* BCEs



Symmetric case: four *4*-BCE megacores (i.e. here *R*=4 and *n*=16)



Asymmetric case: one *R*-BCE megacore with (*n* - *R*) 1-BCEs (i.e. one 4-BCE and 12 1-BCEs, *R*=4, *n*=16)

PS: I know it's irritating having *N* and *n*… n = amount of resources where 1 resource = 1 BCE… and then N = number R-BCEs

Modelling Performance of Asymmetric Multicore Chips

- ***n*** 1-BCE core resources per chip
- 1x *R*-BCEs , (*n*-*R*)x 1-BCEs
- Parallel Fraction of execution is ***F***
- Serial Fraction still **1-*F*** and uses 1 core at rate perf(*R*)
- Therefore: Serial time = (1 – F) / perf(*R*)

- Parallel Fraction *F*
  uses *1* cores at rate perf(*R*) and *n* -*R* cores at rate 1
- Parallel time = *F* / (perf(*R*) + *n* – *R*)

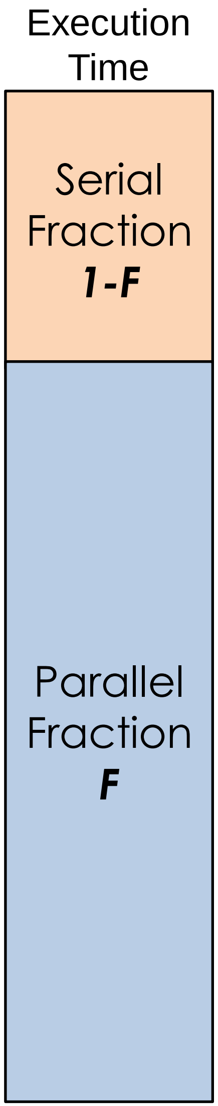
- Therefore, w.r.t. one base core:

$$\text{Asymmetric Speedup} = \frac{1}{\dfrac{1 - F}{\text{perf(R)}} + \dfrac{F}{\text{perf}(R) + (n - R)}}$$

serial part runs on enhanced core

parallel part runs concurrently on one enhanced core & other standard cores

- **Implications?..**

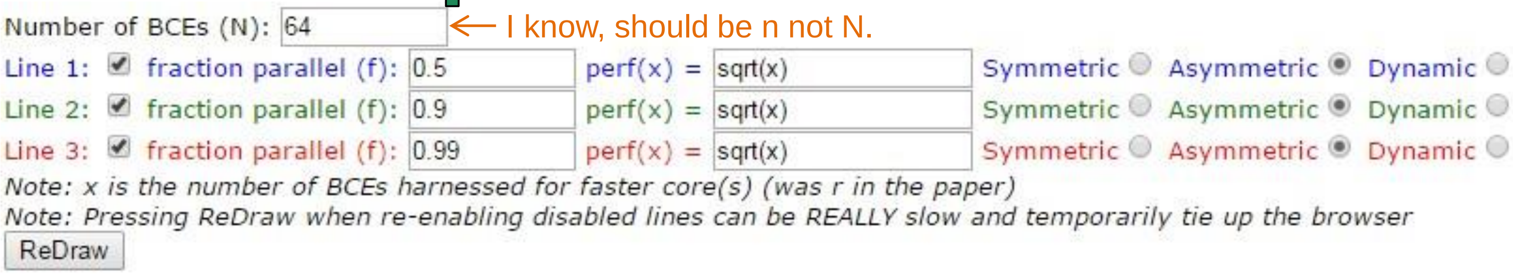


Execution Time

Serial Fraction
1-F

Parallel Fraction
F

Calculate performance…



Number of BCEs (N): 64 ← I know, should be n not N.

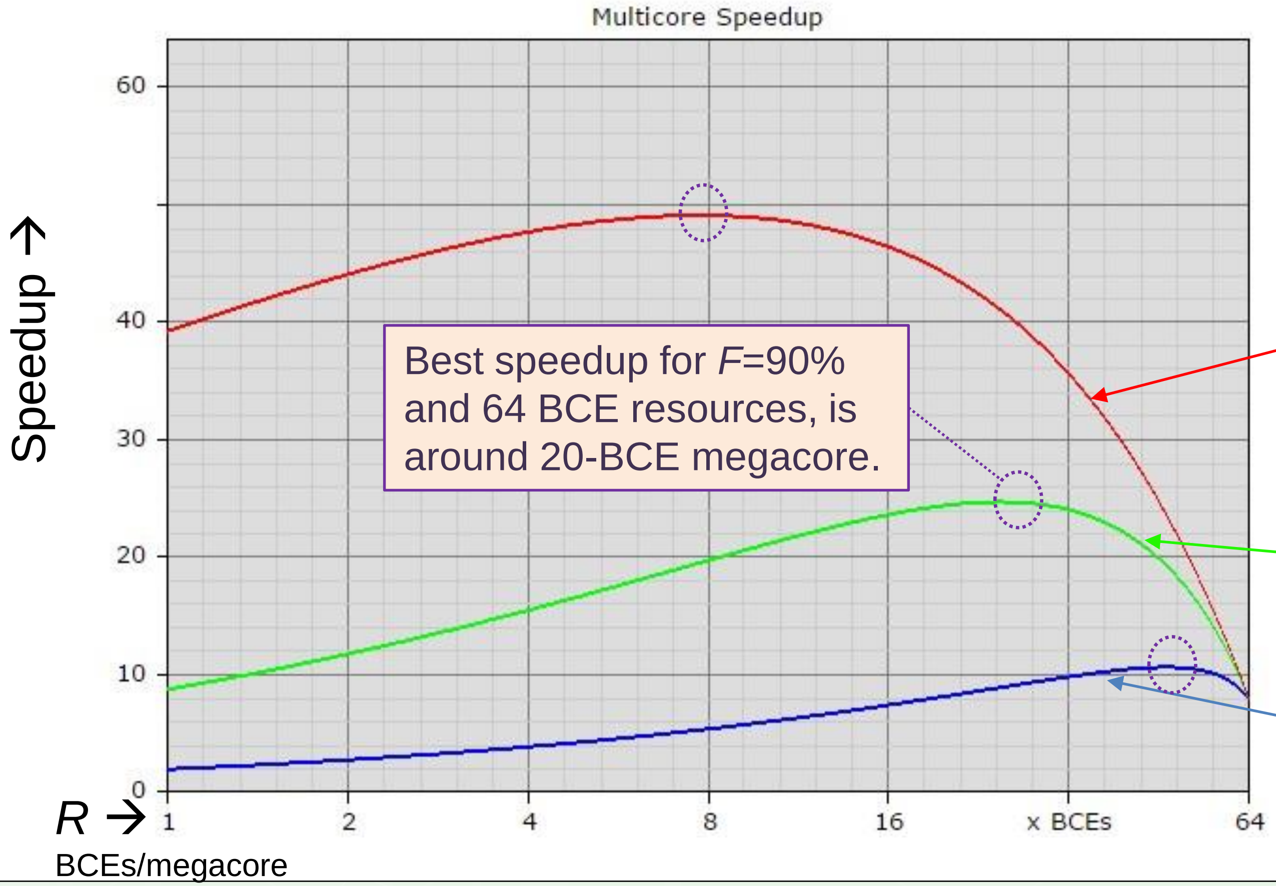
Line 1: ☑ fraction parallel (f): 0.5 perf(x) = sqrt(x) Symmetric ○ Asymmetric ● Dynamic ○

Line 2: ☑ fraction parallel (f): 0.9 perf(x) = sqrt(x) Symmetric ○ Asymmetric ● Dynamic ○

Line 3: ☑ fraction parallel (f): 0.99 perf(x) = sqrt(x) Symmetric ○ Asymmetric ● Dynamic ○

Note: x is the number of BCEs harnessed for faster core(s) (was r in the paper)

Note: Pressing ReDraw when re-enabling disabled lines can be REALLY slow and temporarily tie up the browser



Best speedup for *F*=90% and 64 BCE resources, is around 20-BCE megacore.

Model up to *n*=64 BCEs

Parallel fraction *F*=99%

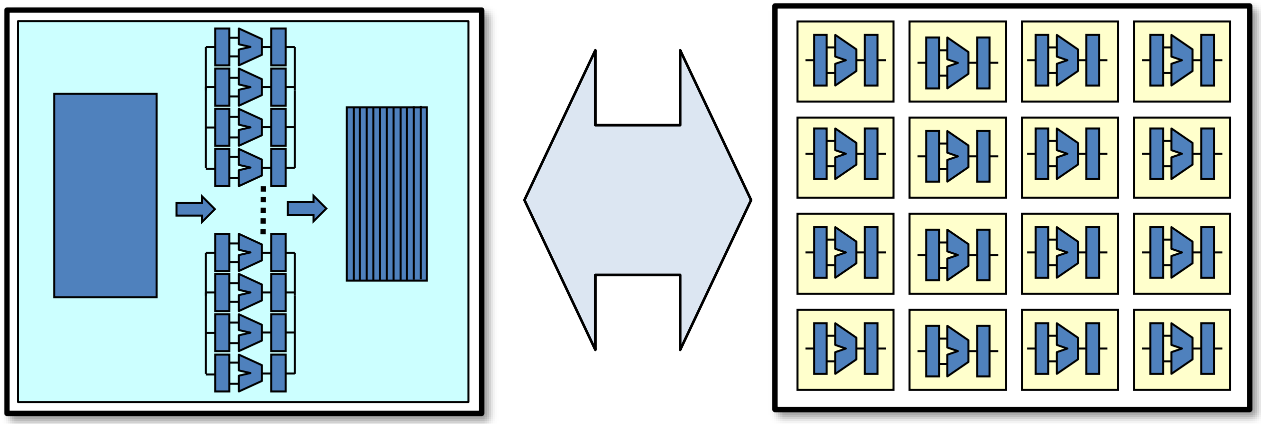
Parallel fraction *F*=90%

Parallel fraction *F*=50%

Dynamic Multicore Chips

Prescribed reading

- *n* base cores for best parallel performance
- Harness *R* cores together for best serial performance *(a bit of an optimistic dream perhaps)*



Serial Case
1x *n*-BCE megacore

Parallel Case
n x *1*-BCE cores

Modelling Performance of Dynamic Multicore Chips

Prescribed reading

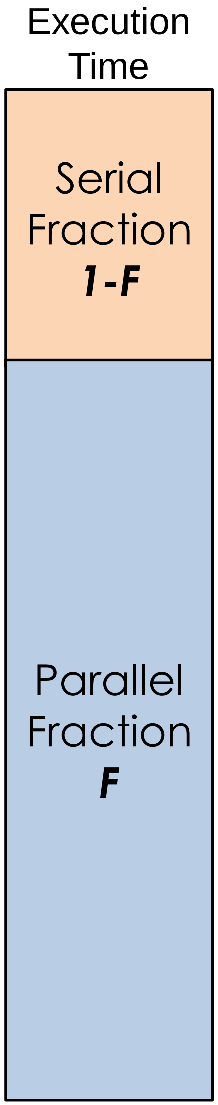
- ***n*** base cores per chip where ***R*** can be harnessed
- 1x *R*-BCEs or (*n* - *R*)x 1-BCEs
- Parallel Fraction of execution is *F*
- Serial Fraction 1-*F* uses *R* BCEs at rate perf(*R*)
- Therefore: Serial time = (1 – *F*) / perf(*R*)

- Parallel Fraction *F* uses *n* BCEs at rate 1 each
- Parallel time = *F* / *n*

- Therefore, w.r.t. one base core:

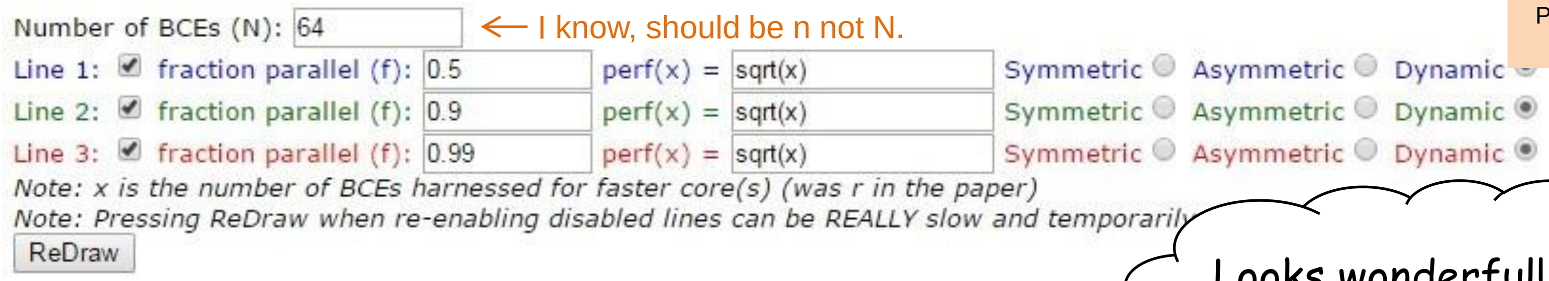
$$\text{Dynamic Speedup} = \frac{1}{\frac{1 - F}{\text{perf}(R)} + \frac{F}{n}}$$

- **Implications?..**



Calculate performance...

Prescribed reading



Number of BCEs (N): 64

← I know, should be n not N.

Line 1: ☑ fraction parallel (f): 0.5 perf(x) = sqrt(x) Symmetric ○ Asymmetric ○ Dynamic

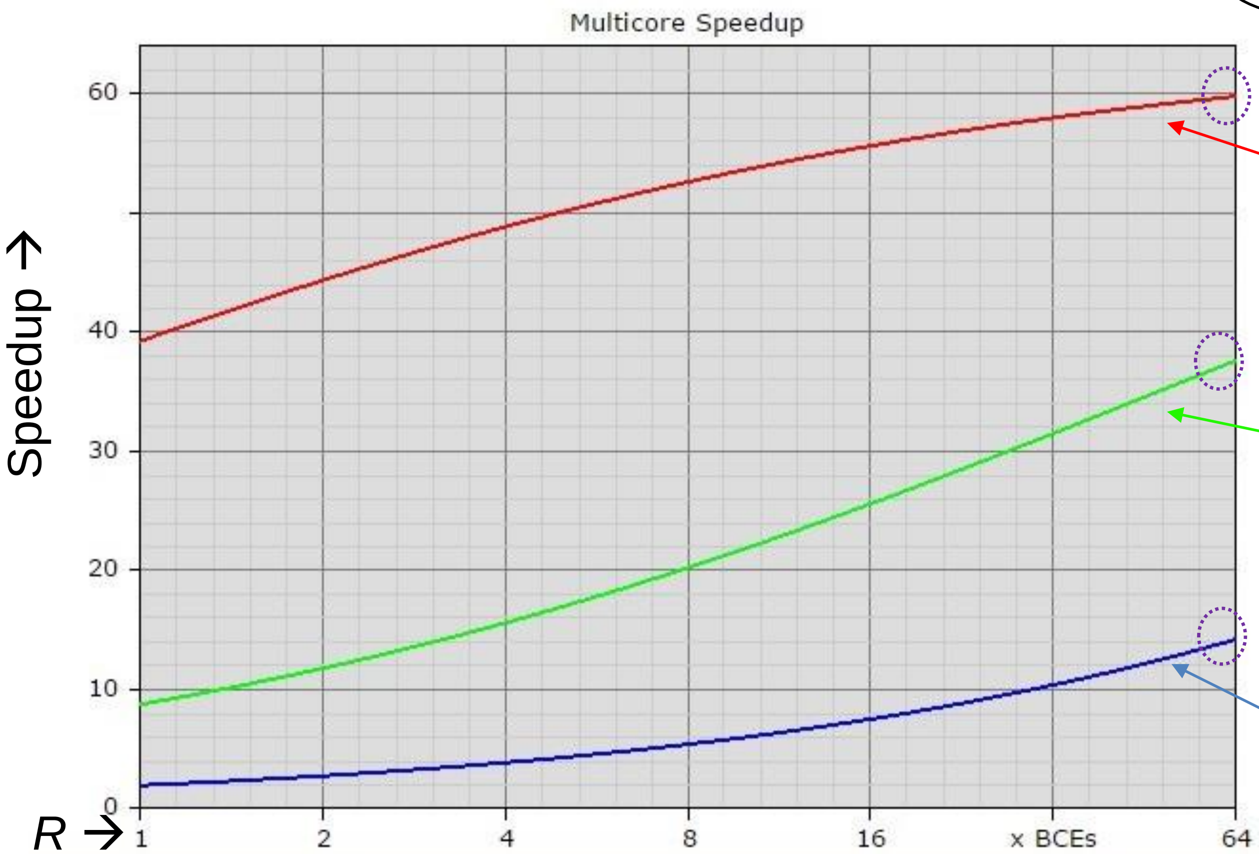
Line 2: ☑ fraction parallel (f): 0.9 perf(x) = sqrt(x) Symmetric ○ Asymmetric ○ Dynamic ●

Line 3: ☑ fraction parallel (f): 0.99 perf(x) = sqrt(x) Symmetric ○ Asymmetric ○ Dynamic ●

Note: x is the number of BCEs harnessed for faster core(s) (was r in the paper)

Note: Pressing ReDraw when re-enabling disabled lines can be REALLY slow and temporarily

Looks wonderful!
But it is realistic?



BCEs/megacore

Model up to *r*=64 BCEs

Parallel fraction *F*=99%

Parallel fraction *F*=90%

Parallel fraction *F*=50%

BCE Performance Calculator – M Hill & M Marty (*summary*)

Prescribed reading

- To apply Amdahl's law to a multicore chip, we need
  - A cost model for the number and performance of cores that the chip can support.
  - 1) assume a multicore chip can contain at most ***n*** base core equivalents (based on resources a chip designer devotes to cores & portion of code is parallelized)
  - 2) Assume (micro-)architects have techniques for using the resources of multiple BCEs to create a core with greater sequential performance.
    - Let the performance of a single-BCE core be **1**.
    - Assume architects can expend the resources of ***R*** BCEs to create a more powerful processor (i.e. a *R*-BCE core) with sequential performance **perf(*R*)**.
    - *Example* assume perf(*R*) = sqrt(*R*) … In other words, we assume efforts that devotes ***R*** BCE resources will result in sequential performance ***R***. Thus, architectures can double performance at a cost of four BCEs, triple it for nine BCEs, etc. illustrated by graph results …

Amdahl's Law Milticore Performance Calculator

Supplementary reading

Check out Hill and Marty's Amdahl's interactive grapher at:

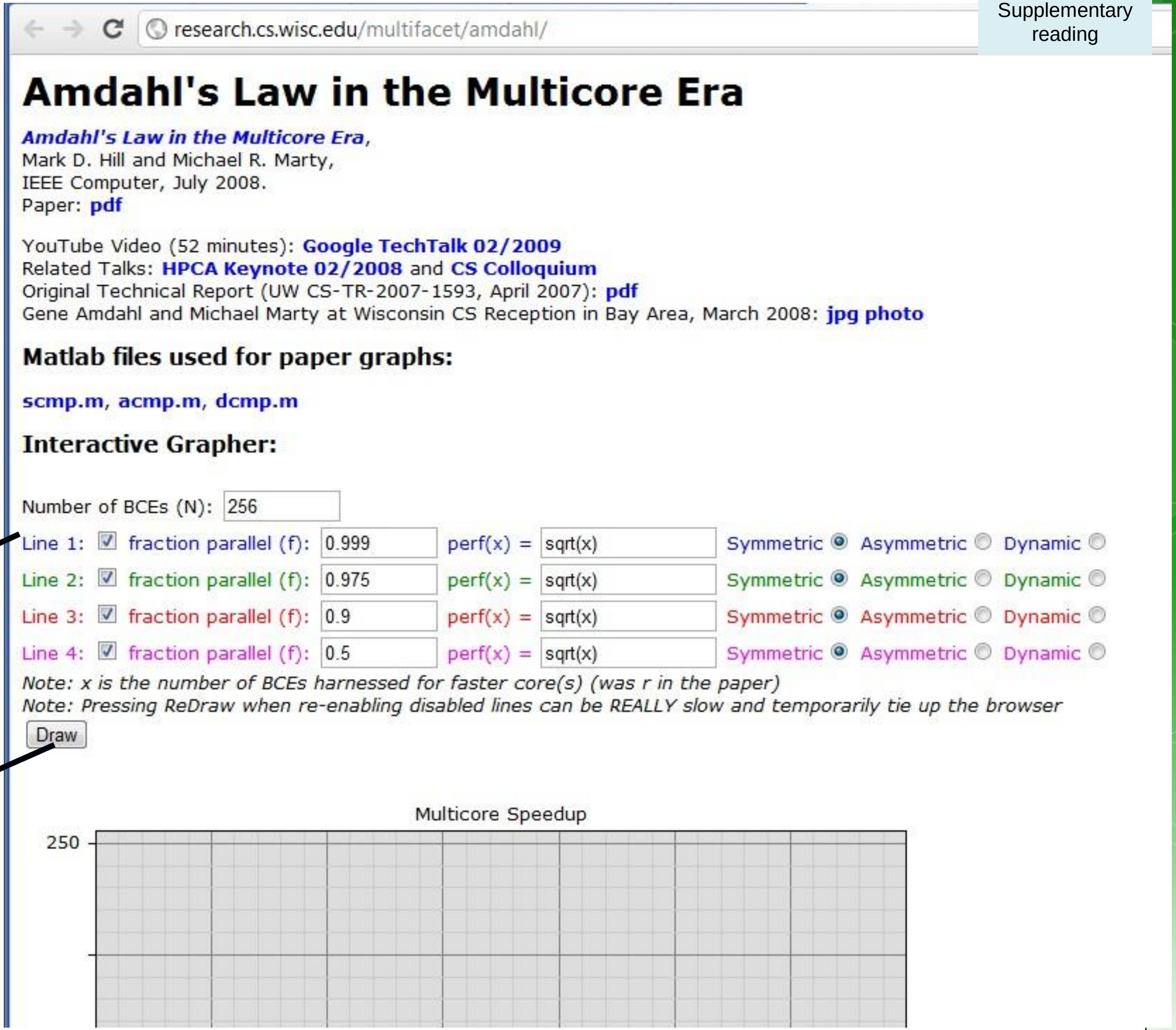
http://research.cs.wisc.edu/multifacet/amdahl/

You can also download the OCTAVE / Matlab code to do these graphs!

How it works:

Specify #cores (*n*), the portion parallel code (*F*), type of BCE cores to use

generate example of resultant graph…



research.cs.wisc.edu/multifacet/amdahl/

Amdahl's Law in the Multicore Era

Amdahl's Law in the Multicore Era,
Mark D. Hill and Michael R. Marty,
IEEE Computer, July 2008.
Paper: pdf

YouTube Video (52 minutes): Google TechTalk 02/2009
Related Talks: HPCA Keynote 02/2008 and CS Colloquium
Original Technical Report (UW CS-TR-2007-1593, April 2007): pdf
Gene Amdahl and Michael Marty at Wisconsin CS Reception in Bay Area, March 2008: jpg photo

Matlab files used for paper graphs:

scmp.m, acmp.m, dcmp.m

Interactive Grapher:

Number of BCEs (N): 256

Line 1: ☑ fraction parallel (f): 0.999 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

Line 2: ☑ fraction parallel (f): 0.975 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

Line 3: ☑ fraction parallel (f): 0.9 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

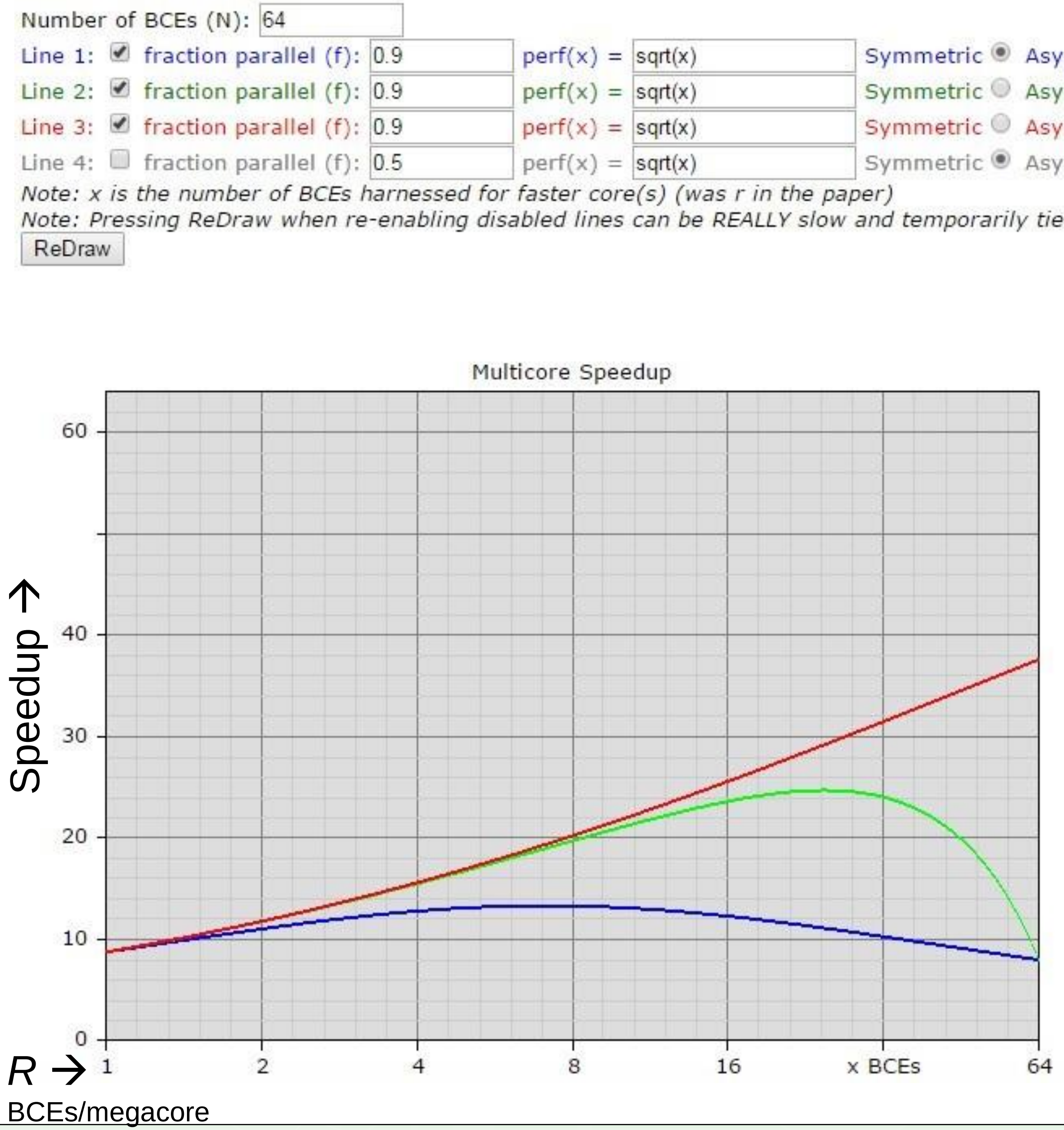
Line 4: ☑ fraction parallel (f): 0.5 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

Note: x is the number of BCEs harnessed for faster core(s) (was r in the paper)
Note: Pressing ReDraw when re-enabling disabled lines can be REALLY slow and temporarily tie up the browser

http://research.cs.wisc.edu/multifacet/amdahl/

Example run of Hill and Marty’s Amdahl’s interactive grapher:

Supplementary reading



Number of BCEs (N): 64

Line 1: ☑ fraction parallel (f): 0.9 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

Line 2: ☑ fraction parallel (f): 0.9 perf(x) = sqrt(x) Symmetric ○ Asymmetric ● Dynamic ○

Line 3: ☑ fraction parallel (f): 0.9 perf(x) = sqrt(x) Symmetric ○ Asymmetric ○ Dynamic ●

Line 4: □ fraction parallel (f): 0.5 perf(x) = sqrt(x) Symmetric ● Asymmetric ○ Dynamic ○

Note: x is the number of BCEs harnessed for faster core(s) (was r in the paper)

Note: Pressing ReDraw when re-enabling disabled lines can be REALLY slow and temporarily tie up the browser

Gives a comparison of the different architecture models. Clearly Dynamic is probably always going to win.

Recommended Reading

Supplementary reading

NB: do try to read this one! *

- Hill and Marty 2008: “Amdahl’s Law in the Multicore Era” Available: http://research.cs.wisc.edu/multifacet/papers/ieeecomputer08_amdahl_multicore.pdf

  See Vula: L08 Hill and Marty - Amdahl Law in Multicore Era.pdf

- Good article on Wikipedia look over: http://en.wikipedia.org/wiki/Amdahl's_law

* Questions on BCEs likely to to appear in test or exam

Further Reading / Refs

Supplementary reading

Suggestions for further reading, slides partially based / general principles elaborated in:

- https://computing.llnl.gov/tutorials/parallel_comp/
- http://www2.physics.uiowa.edu/~ghowes/teach/ihpc12/lec/ihpc12Lec_DesignHPC12.pdf

Some resources related to systems thinking:

- https://www.youtube.com/watch?v=lhbLNBqhQkc
- https://www.youtube.com/watch?v=AP7hMdnNrH4

Some resources related to critical analysis and critical thinking:

An easy introduction to critical analysis:

- http://www.deakin.edu.au/current-students/study-support/study-skills/handouts/critical-analysis.php

An online quiz and learning tool for understanding critical thinking:

- http://unilearning.uow.edu.au/critical/1a.html

Assigned Reading

For Next Thursday …

R8 Hill and Marty 2008: “Amdahl’s Law in the Multicore Era”

Find it on: Vula Resources/Readings
Thursday 5 Mar 2pm:
There will be a short quiz, and I will follow that with solutions

(end of lecture)

Disclaimers and copyright/licensing details

I have tried to follow the correct practices concerning copyright and licensing of material, particularly image sources that have been used in this presentation. I have put much effort into trying to make this material open access so that it can be of benefit to others in their teaching and learning practice. Any mistakes or omissions with regards to these issues I will correct when notified. To the best of my understanding the material in these slides can be shared according to the Creative Commons "Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)" license, and that is why I selected that license to apply to this presentation (it's not because I particulate want my slides referenced but more to acknowledge the sources and generosity of others who have provided free material such as the images I have used).

Image sources:

Clipart sources – public domain CC0 (http://pixabay.com/)
Pixabay
PxFuel – CC0 (https://www.pxfuel.com/)
commons.wikimedia.org
Images from flickr