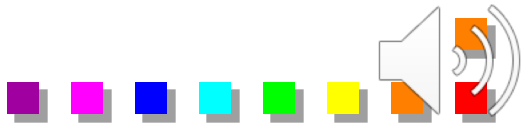
Amdahl's Law

by J. Wunderlich, Ph.D.

Amdahl's Law

- The Computer Engineering version of the "law of diminishing returns" or "law of diminishing marginal utility"
- Simply put, you can "**speed-up**" computer performance in a measured way; however you get less and less benifit for your effort as you increase your effort.



Amdahl's Law

- "Speed-up" = T_{old} / T_{new}
- ***WHERE*** T_{old} = time to execute code **prior to** the implimentation of a "new feature" to speed-up machine performance
- ***AND*** T_{new} = time to execute code **after** the implimentation of an "new feature" to speed-up machine performance
  - ***ALSO,*** $T_{new} = T_{benifit} + T_{other}$
    - ***WHERE*** $T_{benifit}$ = the **new** time to execute the part of the code that benifits from the "new feature"
    - ***AND*** T_{other} = the time to execute the part of the code that does not benifit from the "new feature"

Amdahl's Law (example):

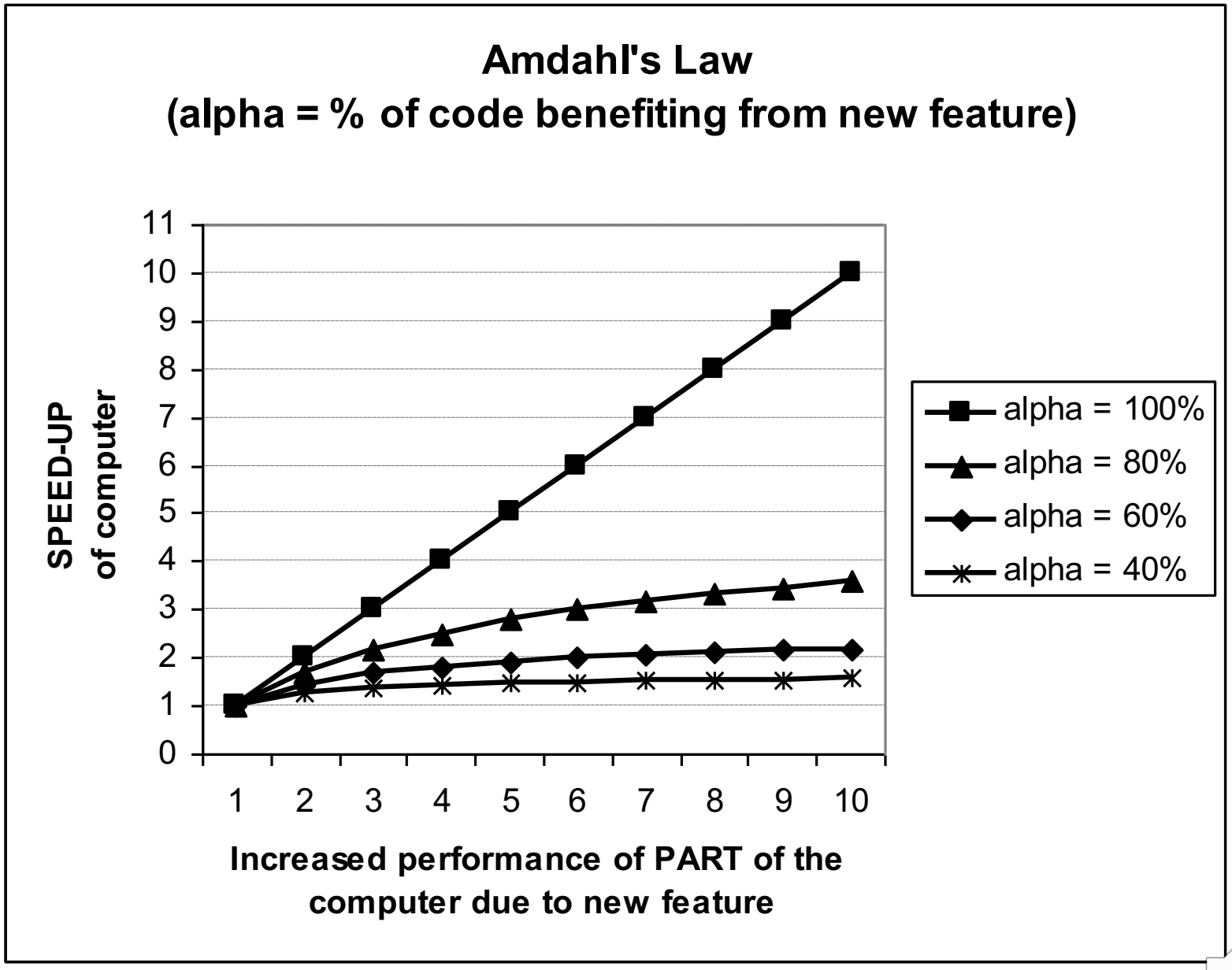
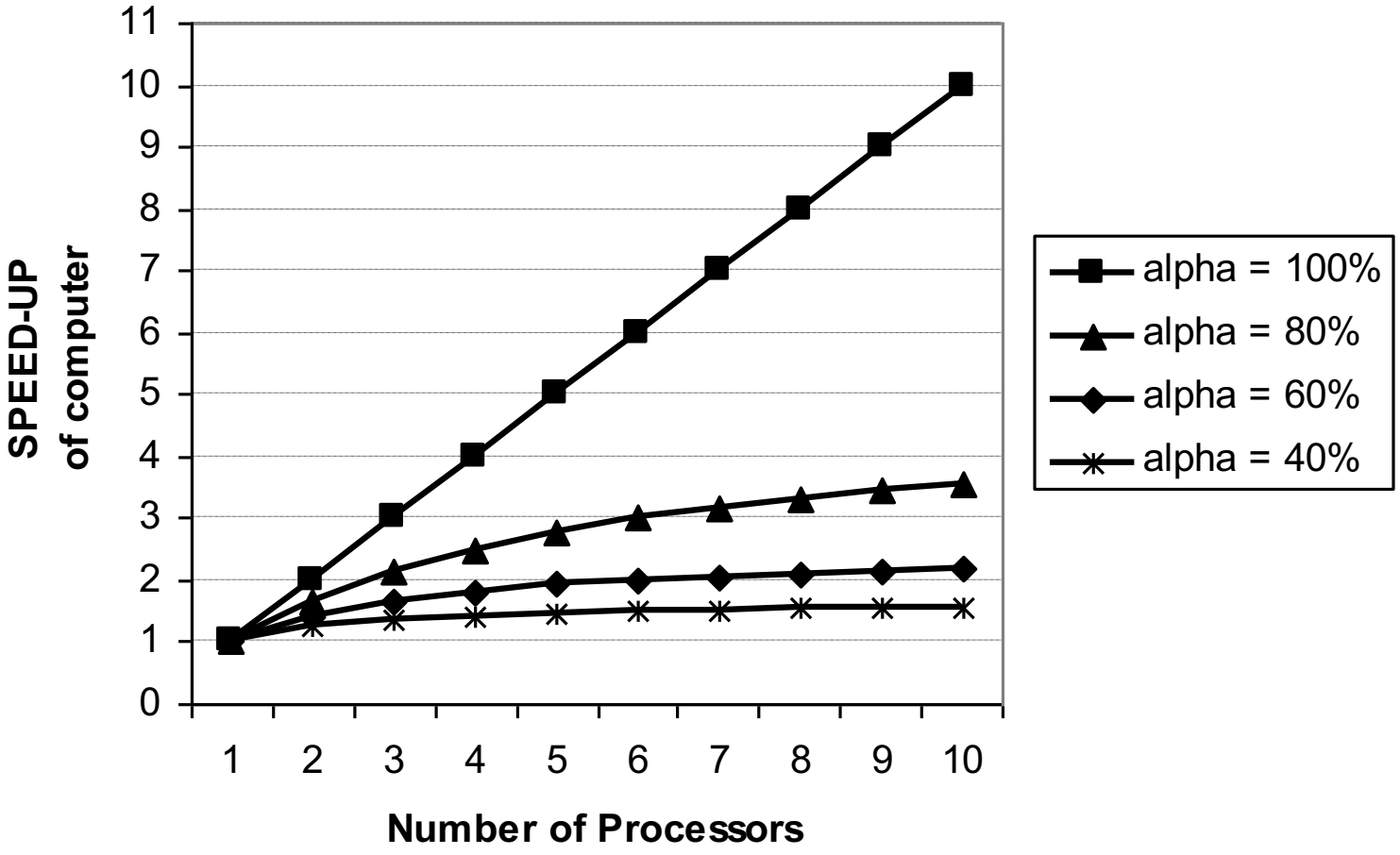
Suppose a computer has a code segment that takes 100 msec to execute. And a proposed new ALU could increase the performance of 40% of that code by 10 times. What is the potential speed-up of the entire computer?

- T_{old} = 100msec
- $T_{benifit}$ = ((40% * (100msec) / 10) = 4msec
- T_{other} = (60% * (100msec)) = 60msec
  - ***therefore*** $T_{new} = T_{benifit} + T_{other}$

    $= 4msec + 60msec$

    $= 64msec$
- "Speed-up" = T_{old} / T_{new}

  $= 100msec / 64msec = 1.56$


Amdahl's Law
(alpha = % of code benefiting from new feature)
SPEED-UP of computer
0
1
2
3
4
5
6
7
8
9
10
11
1
2
3
4
5
6
7
8
9
10
Increased performance of PART of the computer due to new feature
alpha = 100%
alpha = 80%
alpha = 60%
alpha = 40%

Amdahl's Law (applied to parallel processing)
(alpha = % of code"parallizable")