

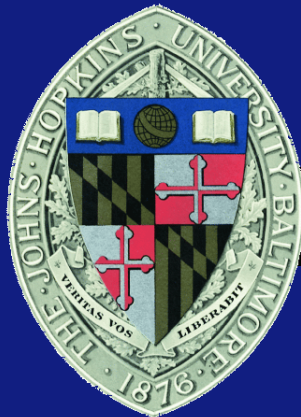
Lecture 2.2

Amdahl's Law

EN 600.320/420

Instructor: Randal Burns

1 February 2017



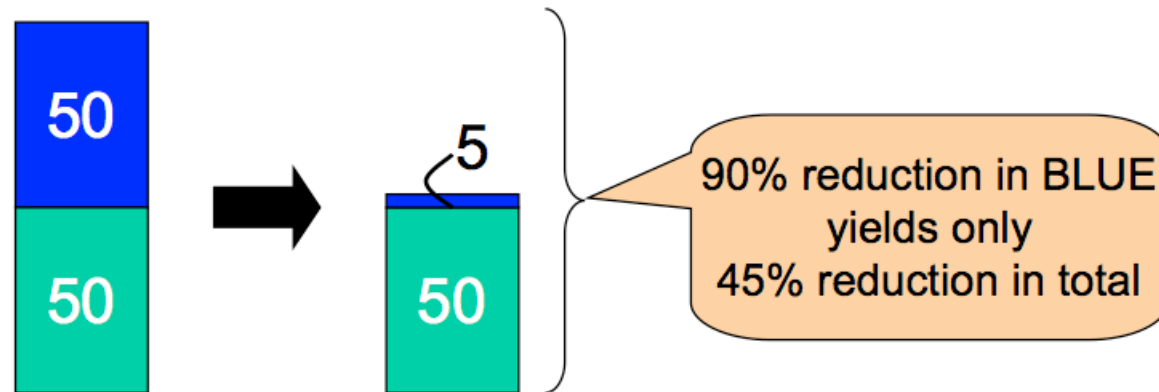
Department of Computer Science, *Johns Hopkins University*

Amdahl's Law

- Improving a portion P of a computation by factor s results in an overall speedup of

$$S_{\text{latency}}(s) = \frac{1}{(1 - p) + \frac{p}{s}}$$

- Paraphrased: speedup limited to fraction improved
 - Obvious but fundamental observation



Amdahl's Law

$$S_{\text{latency}}(s) = \frac{1}{(1 - p) + \frac{p}{s}}$$

- p is the proportion of execution time that benefits from improved resources, i.e. the parallel part
 - $(1-p)$ is the portion that does not benefit; i.e. the serial part
- s is the speedup of the optimized part
- $S_{\text{latency}}(s)$ theoretical speedup of the whole task
- In practice (for analyzing parallel codes)
 - $S_{\text{latency}}(s)$ observed speedup
 - s number of resources (cores or processors)
 - p is the fraction that runs in parallel
- We use the theory of parallelism (Amdahl's) law to infer parameters/properties of a



Amdahl's Law and Parallelism

- *This is my old slide for your reference*
- Consider a parallel computation to consist of parallel steps and of serial (non-parallelizable) steps
- Translation:
 - s (speedup) is the number of processors
 - p (portion) is the optimized/parallelized portion
 - Best-case reasoning about how Amdahl's law governs parallel computing

Paraphrased from Hennesey and Patterson, *Computer Architecture, 2nd Ed.*



Amdahl's Law and Parallelism

- Supposing you want to achieve a speedup of 80 with 100 processors, what fraction of the original computation can be sequential?



Paraphrased from Hennesey and Patterson, *Computer Architecture, 2nd Ed.*

Amdahl's Law and Parallelism

- Supposing you want to achieve a speedup of 80 with 100 processors, what fraction of the original computation can be sequential?

$$\text{Speedup} = \frac{1}{1 - P + \frac{P}{S}}$$

$$80 = \frac{1}{1 - P + \frac{P}{100}}$$

$$P = 0.9975$$

Paraphrased from Hennesey and Patterson, *Computer Architecture, 2nd Ed.*



What is sequential?

- An abstraction to reason about parallelism.
- Sometimes literal:
 - Outer/control thread in OpenMP
 - File system I/O before launching a program
- Sometimes metaphorical:
 - When one/few processes are running and other complete
 - When parallelism introduces additional computation
- The unoptimized fraction of the code
 - When not all resources are doing useful work at full capacity

