

CS 21 - Computer Organization and Assembly Language Programming

Lecture 16 Amdahl's Law

University of the Philippines - Diliman
College of Engineering
Department of Computer Science

Outline

Amdahl's Law

Examples

Insights

Outline

Amdahl's Law

Examples

Insights

The problem

Design improvement costs time, money, and manpower. So realistically, you could only improve on certain areas of your processor design at any one time.

The problem

Design improvement costs time, money, and manpower. So realistically, you could only improve on certain areas of your processor design at any one time.

Which area should you work on? Where would you invest?

Amdahl's Law

You improved a segment of your program execution by s . The said segment is p percent of your total program. The speedup resulting from your improvement effort would be

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

Amdahl's Law

Amdahl's law, also known as Amdahl's argument, is named after computer architect Gene Amdahl, and is used to find the **maximum expected improvement** to an overall system when *only part* of the system is improved.

Outline

Amdahl's Law

Examples

Insights

Example

I was able to halve the execution time for my jump instructions. Jump instructions make up 20% of my program. What should my expected speedup be?

Example

I was able to halve the execution time for my jump instructions. Jump instructions make up 20% of my program. What should my expected speedup be?

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

Example

I was able to halve the execution time for my jump instructions. Jump instructions make up 20% of my program. What should my expected speedup be?

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

$$Speedup = \frac{1}{(1 - 0.2) + \frac{0.2}{2}}$$

Example

I was able to halve the execution time for my jump instructions. Jump instructions make up 20% of my program. What should my expected speedup be?

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

$$Speedup = \frac{1}{(1 - 0.2) + \frac{0.2}{2}}$$

$$Speedup = 1.11$$

Example

I have two improvement options:

1. Make load instructions finish 10x faster. Load instructions make up 10% of total instructions.
2. Make register-register instructions 1.5x faster. Register-register instructions make up 40% of total instructions.

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

Which should I choose?

Example

I have two improvement options:

1. Make load instructions finish 10x faster. Load instructions make up 10% of total instructions.
2. Make register-register instructions 1.5x faster. Register-register instructions make up 40% of total instructions.

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

Which should I choose?

IMPROVEMENT OPTION 2

Outline

Amdahl's Law

Examples

Insights

Insights

Insights from Amdahl's Law:

1. Big improvements in components/instructions that are rarely used = small improvements
2. Small improvements in components/instructions that are frequently used = big improvements

BOTTOMLINE:

Insights

Insights from Amdahl's Law:

1. Big improvements in components/instructions that are rarely used = small improvements
2. Small improvements in components/instructions that are frequently used = big improvements

BOTTOMLINE:

Make the common case fast

Example

A certain instruction class makes up 20% of the program. I want to achieve 1.16 speedup. By how much should I improve the said instruction class to achieve this?

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

Example

A certain instruction class makes up 20% of the program. I want to achieve 1.16 speedup. By how much should I improve the said instruction class to achieve this?

$$Speedup = \frac{1}{(1 - p) + \frac{p}{s}}$$

Answer: 3.222