14 April 2020 13:31

2.3 Quantitative Principles of Computer Design

① Take advantage of parallelism
- CLA - Carry Lookahead Adder
- ILP - Pipelining
- HoC, multiprocessors

② Principle of locality

Programs tend to reuse data and instructions they have used recently

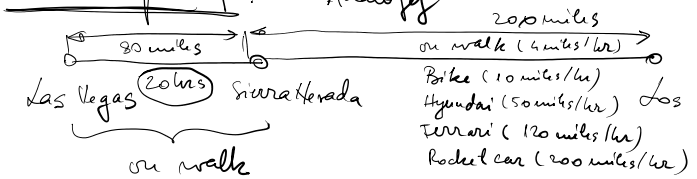
Rule of thumb: a program spends 90% of its execution time with only 10% of the code

This principle works for instructions! also applies to data
- temporal locality
- spatial locality

③ Focus on the common case ✓
- pervasive & important
- trade off
- Amdahl's law (quantifies improvement)

What is speedup?

Analogy



Vehicle for 2nd part	Hrs for 2nd part	Speedup in the desert	Hrs for the entire trip	Speedup for the entire trip
Feet	50	1	70	1
Bike	20	2.5	40	1.8
Hyundai	4	12.5	24	2.9
Ferrari	1.67	30	21.67	3.2
Rocket Car	1	50	21	3.3

$$\text{Speedup}_{\text{enhanced}} = \frac{\text{Performance when enhancement is possible}}{\text{Performance without enhancement}} =$$

$$= \frac{\text{Execution time without enhancement}}{\text{Execution time when enhancement is possible}}$$

$$\text{Fraction enhanced} = \frac{80 \text{ miles}}{280 \text{ miles}} = 0.285$$

$$\text{Execution time when enhancement is possible} = \text{Execution time without enhancement} \left((1 - \text{Fraction enhanced}) + \frac{\text{Fraction enhanced}}{\text{Speedup}} \right)$$

$$\text{Speedup}_{\text{enhanced}} = \frac{\text{Execution time without enhancement}}{\text{Execution time without enhancement} \left((1 - \text{Fraction enhanced}) + \frac{\text{Fraction enhanced}}{\text{Speedup}} \right)} = \frac{1}{(1 - \text{Fraction enhanced}) + \frac{\text{Fraction enhanced}}{\text{Speedup}}}$$

$$\lim_{\text{speedup} \to \infty} \text{Speedup}_{\text{enhanced}} = \frac{1}{1 - \text{Fraction enhanced}}$$