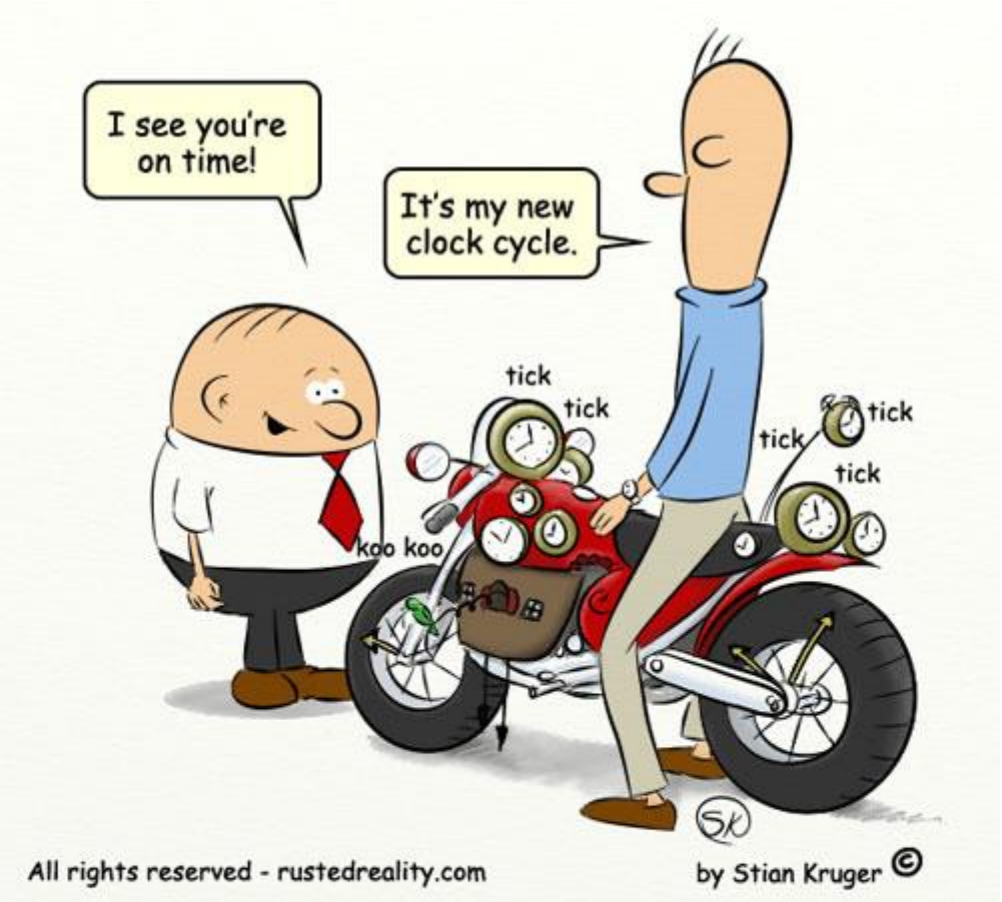
Lecture-2 (Performance Evaluation)
CS422-Spring 2020

Biswa@CSE-IITK

Let’s start with performance



- Single core
- Multi core
- Evaluation
- Benchmarks
- Metrics
- Simulators
- Latency and bandwidth

Performance

- ***Latency*** (execution/response time): time to finish one task. It is additive (Performance = 1/latency)

- ***Throughput*** (bandwidth): number of tasks/unit time. It is not additive

- Example: move people from A to B, **10** miles
  - Car: capacity = 5, speed = 60 miles/hour
  - Bus: capacity = 60, speed = 20 miles/hour
  - Latency: car = 10 min, bus = 30 min
  - Throughput: car = 15 PPH (w/ return trip), bus = 60 PPH

- *Latency lags bandwidth, Bandwidth hurts latency,*

Read: https://cacm.acm.org/magazines/2004/10/6401-latency-lags-bandwith/fulltext

Latency vs Bandwidth

Latency vs Bandwidth, How they affect each other?

Latency helps bandwidth but not vice versa.

- **DRAM latency** ↓
- **More # Accesses ~DRAM Bandwidth** ↑

Bandwidth usually hurts latency

- **Queues - Bandwidth** ↑
- **Increases latency** ↓

Some More on Latency and Bandwidth

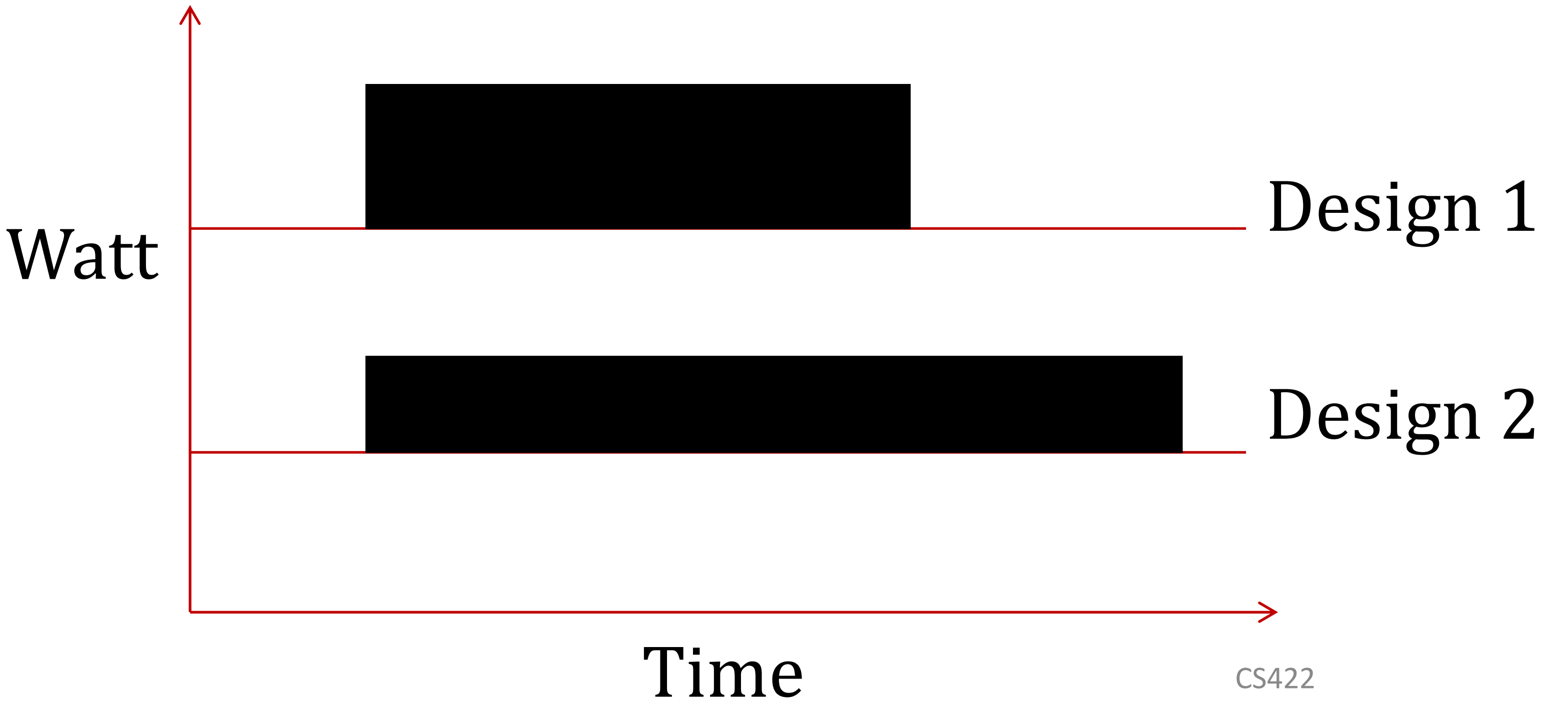
Bandwidth problems can be cured with money.
Latency problems are harder because the speed of light is fixed – you can't bribe God

https://www.youtube.com/watch?list=PL2LuePcZTMh_MzNHqZWNdvWdAnAThHCKK&v=lfqgpuH10uc&feature=emb_logo
https://www.youtube.com/watch?v=GNK-67JUH7M&list=PL2LuePcZTMh_MzNHqZWNdvWdAnAThHCKK&index=2
https://www.youtube.com/watch?v=5CxpoGwCxKU&list=PL2LuePcZTMh_MzNHqZWNdvWdAnAThHCKK&index=3

Energy and Power

Energy: Measure of using power for some time

Power: Instantaneous rate of energy transfer



Power: Height of the curve

Energy: Area under the curve

Power efficiency = Performance/watt

Energy efficiency = Performance/Joule

Execution Time

$$\textbf{Execution Time} = \frac{\textbf{Time}}{\textbf{Program}}$$

Executed Instructions not static ones: Driven by: Algorithm, ISA, and Compilers

Instruction ??

$$= \frac{\textbf{Instructions}}{\textbf{Program}} \times$$

Driven by: ISA and Processor Organization

Cycle ??

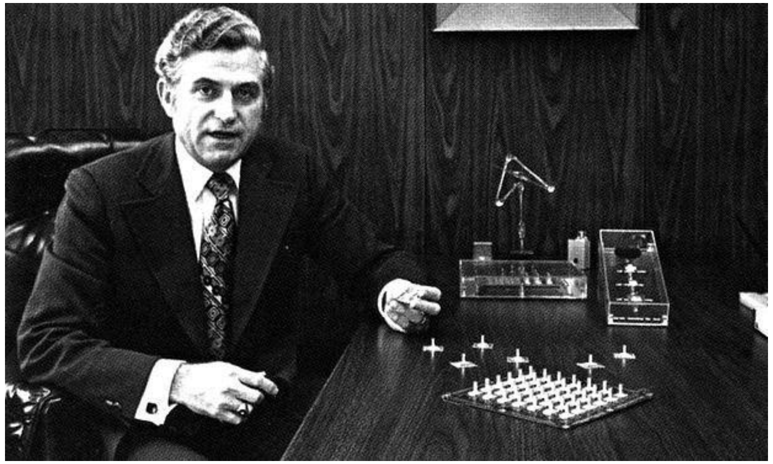
$$= \frac{\textbf{Instructions}}{\textbf{Program}} \times \frac{\textbf{Cycles}}{\textbf{Instruction}} \times$$

Driven by Technology

$$= \frac{\textbf{Instructions}}{\textbf{Program}} \times \frac{\textbf{Cycles}}{\textbf{Instruction}} \times \frac{\textbf{Time}}{\textbf{Cycle}}$$

(Iron law)

Amdahl's Law



Source: The Guardian

$$\mathrm{ExTime}_{new} = \mathrm{ExTime}_{old} \times \left[\left(1 - \mathrm{Fraction}_{enhanced}\right) + \frac{\mathrm{Fraction}_{enhanced}}{\mathrm{Speedup}_{enhanced}} \right]$$

$$\mathrm{Speedup}_{overall} = \frac{\mathrm{ExTime}_{old}}{\mathrm{ExTime}_{new}} = \frac{1}{\left(1 - \mathrm{Fraction}_{enhanced}\right) + \frac{\mathrm{Fraction}_{enhanced}}{\mathrm{Speedup}_{enhanced}}}$$

Amdahl's Law

Which one will provide better overall speedup?

A. Small speedup on the large fraction of execution time.
B. Large speedup on the small fraction of execution time.
C. Does not matter.

Depends on the difference between small and large. Mostly it is A.

Amdahl's law for parallel processing

Evaluation

- *To Compare Processor A with Processor X by running programs*
- *How many programs?* 
- *The programs that you care.* 
- *What if I want to build a new one (processor, caches, DRAM) ?* 

World of Benchmarks

- SPEC CPU 2017 (https://www.spec.org/cpu2017/)

The **SPEC CPU® 2017** benchmark package contains SPEC's next-generation, industry-standardized, CPU intensive suites for measuring and comparing compute intensive performance, stressing a system's processor, memory subsystem and compiler.

SPECspeed: used for comparing time for a computer to complete single tasks

SPECrate: measure the throughput or work per unit of time.

World of Benchmarks

CloudSuite (https://www.cloudsuite.ch/)

CloudSuite is a benchmark suite for cloud services. The benchmarks are based on real-world software stacks and represent real-world setups.

PARSEC (https://parsec.cs.princeton.edu/)

Benchmark suite composed of multithreaded programs. The suite focuses on emerging workloads and was designed to be representative of next-generation shared-memory programs for chip-multiprocessors.

World of Benchmarks

- MobileBench (https://mobilebench.engineering.asu.edu/)

comprising a selection of representative smart phone applications.

Pitfalls of Benchmarks

- Benchmark not representative
  - Your workload is I/O bound → SPECCPU is useless
- Benchmark is too old
  - Benchmarks age poorly (SPEC CPU 2006 and then CPU 2017)
  - Benchmarketing pressure causes vendors to optimize compiler/hardware/software to benchmarks

  → Need to be periodically refreshed

Non-Benchmarks

- Application kernels: A small code fragment or part of the program
- Synthetic benchmark : Not part of any real program!!
- Micro-benchmark
- *OK! So, I will create a chip and then evaluate these benchmarks*

World of Simulators

- Functional Simulator: Used to verify the correct execution of the program. Can not be used for performance evaluation.

- Performance simulators:

(i) Trace-driven: ChampSim (https://github.com/ChampSim/ChampSim)

(ii) Execution-driven: gem5, Multi2sim

Functional simulator is part of the performance simulators.

Evaluation Contd..

Pick a *relevant* benchmark suite

Measure IPC of each program

Summarize the performance using:

- Arithmetic Mean (AM)
- Geometric Mean (GM)
- Harmonic Mean (HM)

Which one to choose?

Example

	IMTEL	ABM	AND
App. one	10	20	30
App. two	20	30	40
App. three	30	40	10

Which machine performs better over IMTEL and why?

Contd.

	ABM	AND
App. one	2	3
App. two	1.5	2
App. three	1.3	0.3
A.M.	1.60	1.76
G.M.	1.57	1.21
H.M.	1.54	0.72

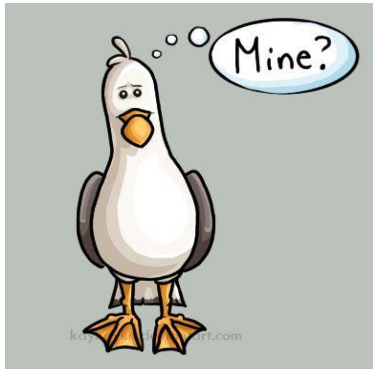
AM on Ratios

	X	Y
App. 1	1	100
App. 2	1000	10

Normalized to X	X	Y
App. 1	1	100
App. 2	1	0.01
AM	1	50.005

Normalized to Y	X	Y
App. 1	0.01	1
App. 2	100	1
AM	50.005	1

Y is 50 times faster than X



X is 50 times faster than Y

When to use what?

Edgar H. Sibley
Panel Editor

Using the arithmetic mean to summarize normalized benchmark results leads to mistaken conclusions that can be avoided by using the preferred method: the geometric mean.

Do not use A.M. on normalized numbers

HOW NOT TO LIE WITH STATISTICS: THE CORRECT WAY TO SUMMARIZE BENCHMARK RESULTS

Use G.M. for normalized numbers

PHILIP J. FLEMING and JOHN J. WALLACE

What about Multi-core Systems?

Application *i* running on an N-core system

Throughput = ∑ IPC (i)

Individual Slowdown (i) = CPI-together (i) / CPI-alone (i)

Weighted Speedup = ∑ (IPC-together(i) / IPC-alone (i))

Harmonic Mean of Speedups = N/∑ (IPC-alone(i)/IPC-together (i))

Unfairness =
Max-Slowdown/Min-Slowdown =
max(Individual slowdowns)/min(individual slowdowns)

Todos

- Clock cycle, machine cycle, tick, ……….
- AM and GM on ratios
- Reading assignment: Latency lags bandwidth and YouTube links

From Lecture-1 (Deadline: January 12):

Compare and contrast KD vs RM building [Put your perspective on Piazza]

Assignment 0:
https://docs.google.com/forms/d/e/1FAIpQLSeofKOH0a9oktzGdf7zW-9GpMmcVfnxqsbgLuspFGeJ-MfRlg/viewform