Computer Architecture

Instrumentation and Simulation

Debadatta Mishra, CSE, IITK

Recap (Performance Analysis)

- Performance analysis
  - Throughput, Response time
  - Response time vs. throughput
  - Processor performance (IPC, CPI)
  - Summarizing performance (AM, GM, HM)

Agenda: Amdahl's law, Instrumentation (Intel PIN), Simulation (champsim)

Is an optimization good?

On a machine, instruction **X** consumes 5 cycles and instruction **Y** consumes 100 cycles. A possible optimization reduces the execution time of **Y** by 10x while increases the execution time of **X** by 2x. As a computer architect, will you prefer this optimization?

(A) Yes
(B) No
(C) Can not say

Amdahl's law in action

On a machine, instruction **X** consumes 5 cycles and instruction **Y** consumes 100 cycles. A possible optimization reduces the execution time of **Y** by 10x while increases the execution time of **X** by 2x. As a computer architect, will you prefer this optimization?

(A) Yes
(B) No
(C) Can not say

Amdahl's law in action

On a machine, instruction **X** consumes 5 cycles and instruction **Y** consumes 100 cycles. A possible optimization reduces the execution time of **Y** by 10x while increases the execution time of **X** by 2x. As a computer architect, will you prefer this optimization?

(A) Yes
(B) No
(C) Can not say

Depends on the usage ratios of **X** and **Y**. If **X**:**Y** = 99:1, then NO. If **X**:**Y** = 90:10, then YES.

Amdahl's law

Consider that a fraction **E** of the overall execution, is reduced by a factor **X**

$$ExecTime_{new} = ExecTime_{old} * (1 - E) + ExecTime_{old} * \frac{E}{X}$$

$$OverallSpeedup = \frac{ExecTime_{old}}{ExecTime_{new}}$$

$$= \frac{1}{(1-E)+\frac{E}{X}}$$

- Law of diminishing returns for the optimizations to the same subsystem
- Upper bound on overall speed up?

Amdahl's law

Consider that a fraction **E** of the overall execution, is reduced by a factor **X**

$$ExecTime_{new} = ExecTime_{old} * (1 - E) + ExecTime_{old} * \frac{E}{X}$$

$$OverallSpeedup = \frac{ExecTime_{old}}{ExecTime_{new}}$$

$$= \frac{1}{(1-E)+\frac{E}{X}}$$

- Law of diminishing returns for the optimizations to the same subsystem
- Upper bound on overall speed up? $\frac{1}{1-E}$

Amdahl's law in multicore

In this case, **E = Fraction of parallelizable code** and **X = # of cores**

$$OverallSpeedup = \frac{ExecTime_{old}}{ExecTime_{new}}$$

$$= \frac{1}{(1-E)+\frac{E}{X}}$$

Q1. If in a program, 80% of the code is parallelizable, what will be the speedup if a 8-core system is used?

Q2. In the above example, what is the maximum speedup assuming no limitation to number of cores?

Amdahl's law in multicore

In this case, **E = Fraction of parallelizable code** and **X = # of cores**

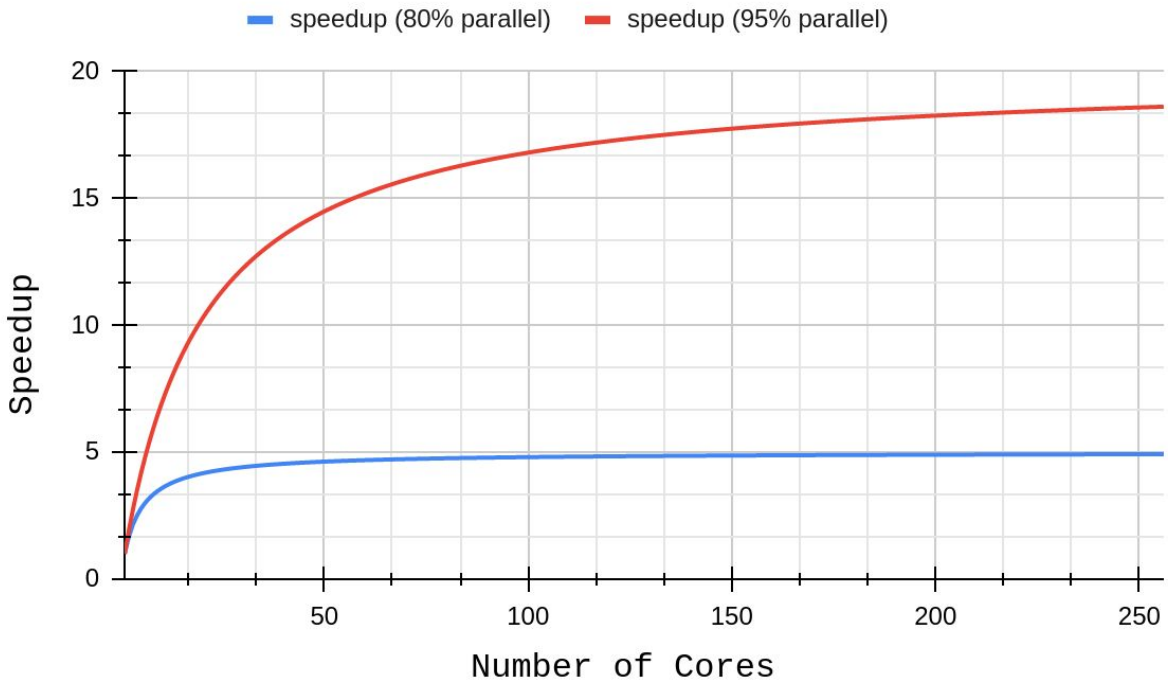
$$OverallSpeedup = \frac{ExecTime_{old}}{ExecTime_{new}}$$

$$= \frac{1}{(1-E)+\frac{E}{X}}$$

Q1. If in a program, 80% of the code is parallelizable, what will be the speedup if a 8-core system is used? **(~3.3x)**

Q2. In the above example, what is the maximum speedup assuming no limitation to number of cores? **(5x)**

Example: Diminishing return with multicore



Takeaway: Adding more cores, does not help improve the performance after a certain point.

In other words, the cost per unit speedup increases significantly

Performance Analysis: why simulation?

- Benchmarks on real system is not very common for architecture performance analysis. Why?

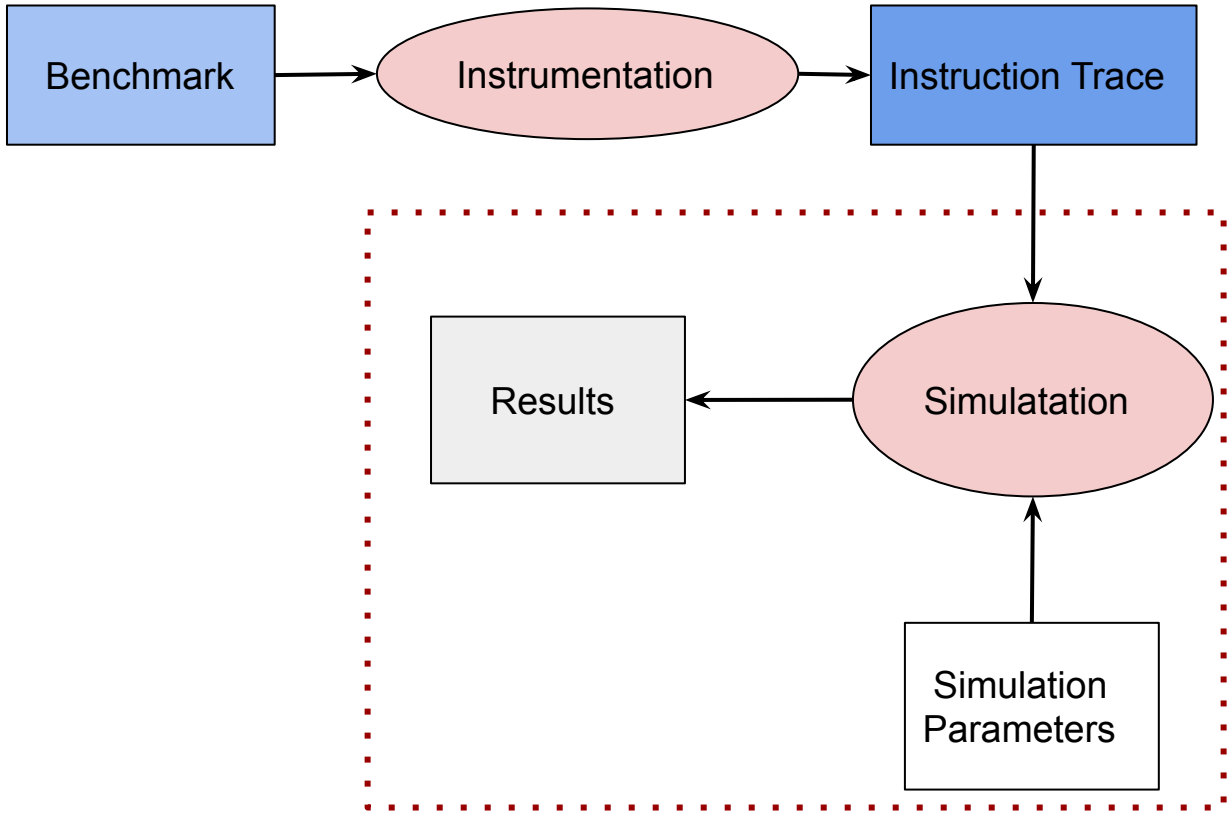
Performance Analysis: why simulation?

- Benchmarks on real system is not very common for architecture performance analysis. Why?
- Issues with real systems
  - Opaqueness: no clue on what is underneath, can not change design
  - Noise: difficult to control the environment
  - Reproducibility: Real systems have insanely large # of configurations!

Performance Analysis: why simulation?

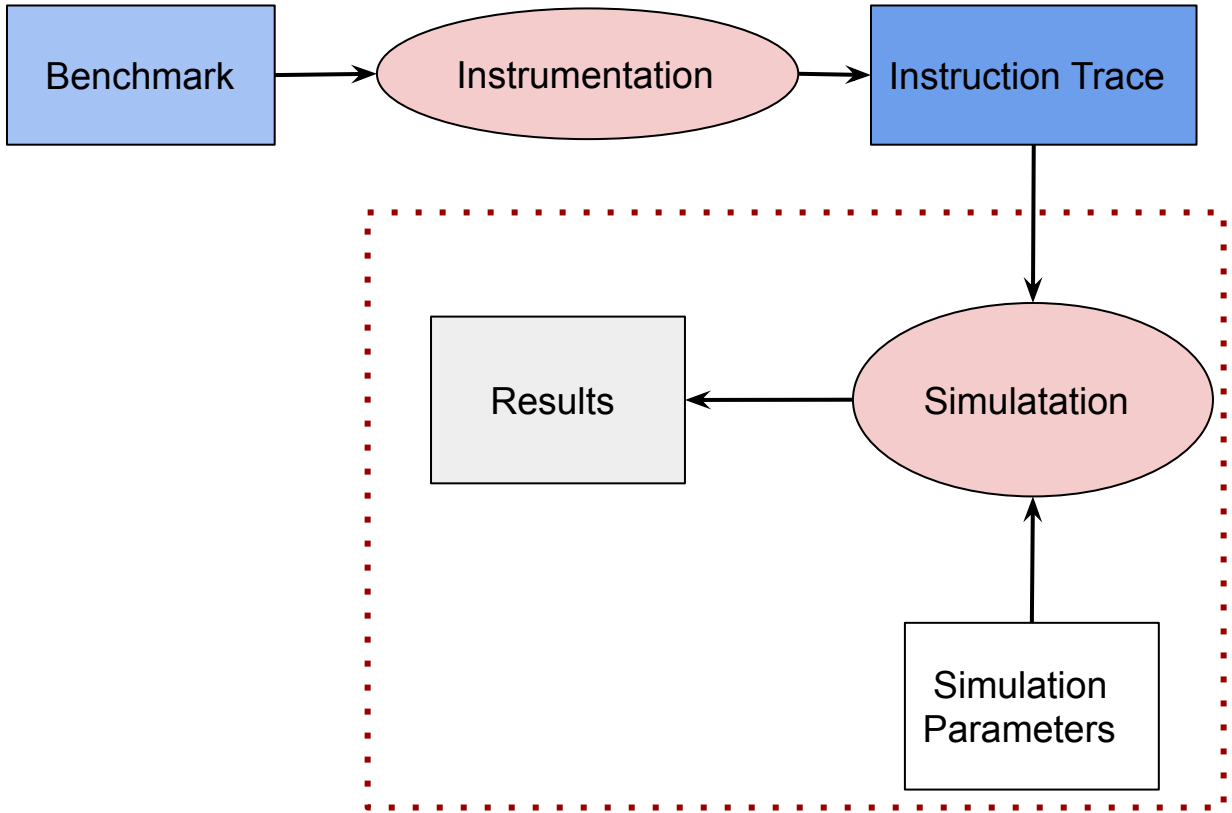
- Benchmarks on real system is not very common for architecture performance analysis. Why?
- Issues with real systems
  - Opaqueness: no clue on what is underneath, can not change design
  - Noise: difficult to control the environment
  - Reproducibility: Real systems have insanely large # of configurations!
- Architectural simulators
  - Discrete event simulators, examples: Champsim, Gem5, Multi2sim
  - More detailed simulation ⇒ Accurate modeling of the real system

Trace based simulation



- Instrumentation is a one-time activity for a given benchmark
- Simulation is repeated for different configs and h/w designs

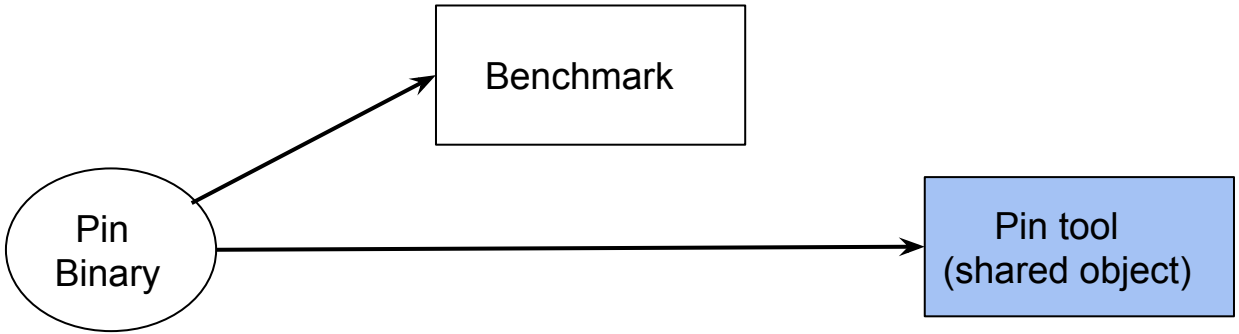
Trace based simulation



For this course:

- Instrumentation using Intel Pin
- Simulation using Champsim

Instrumentation using Intel Pin



Perform dynamic instrumentation of the benchmark binary like a JIT compiler.

Specify the points of instrumentation and handlers for those points.

Simulation using Champsim

- Champsim is a trace-based simulator
- Features: x86 traces, OoO CPU, cache/memory
- Gives a specialized pintool for tracing
- Example:
  - Change L1 DCache size for a simple benchmark
  - L1 DCache hit improves with increased cache size, but no improvement in IPC!
  - Changing OoO parameters makes the effects visible…


???