Program Testing and Analysis: Performance Profiling

Prof. Dr. Michael Pradel

Software Lab, TU Darmstadt

Outline

1. **Introduction**
2. **CPU Time Profiling**
3. **Empirical Complexity**

Partially based on these papers:

- *Evaluating the accuracy of Java profilers*, Mytkowicz et al., PLDI 2010
- *Measuring empirical computational complexity*, Goldsmith et al., ESEC/FSE 2007

Motivation

- **Performance: Non-functional property**
- **Important because:**
  - Users dislike slow applications
  - Related to monetary cost (e.g., in data centers or automated trading)
  - Related to energy consumption
- **Simple changes may yield huge improvements**

Performance Profiling

- **Profiling: Dynamic analysis to measure performance**
- **Observe runtime behavior to**
  - Measure performance of code
  - Understand performance bottlenecks
- **Ultimate goal: Provide insights that help developer address bottlenecks**

Performance

- **Various quantities to measure:**
  - Time (focus of this lecture), memory usage, network bandwidth
- **Absolute performance**
  - E.g., milliseconds (time) or megabyte (memory usage)
- **Relative performance**
  - Compare versions of same program
  - Compare different programs
  - Compare ways to execute the same program

Speedup vs. Improvement

Two representations of relative performance:

Speedup

$$s = \frac{t_{baseline}}{t_{measured}}$$

Improvement

$$i = \frac{t_{baseline} - t_{measured}}{t_{baseline}}$$

Example: A takes 10 seconds, B takes 15 seconds

- Speedup of A over B is $15/10 = 1.5$
  (A is 1.5x faster than B)
- A improves over B by $\frac{15 - 10}{15} = 33\%$

Scalability

- **Related to performance, but not the same**
- **Scalability: How does performance change w.r.t. to some parameter**
- **Typical parameters:**
  - Input size
  - Number of CPU cores
  - Available memory

Execution Time

What is ”time”?

Elapsed time

- **CPU time**
  - **User CPU time**
  - **System CPU time**
- **Other time**
  - **Disk I/O**
  - **Network I/O**

(See Unix ”time” command)

What do we want to measure the execution time of?

- Entire program
- Code segment of interest

Quiz

Program	Elapsed time	CPU time
A	4s	3s
B	7s	4s

Which of the following is true?

- A is 1.43x faster than B
- B has a speedup of 0.57x over A
- A has a speedup of 1.75x over B
- A improves the CPU-time consumption by 25%
- A improves the CPU-time consumption by 33%

Quiz

Program	Elapsed time	CPU time
A	4s	3s
B	7s	4s

Which of the following is true?

- A is 1.43x faster than B **1.75x**
- B has a speedup of 0.57x over A ✓
- A has a speedup of 1.75x over B ✓
- A improves the CPU-time consumption by 25% ✓
- A improves the CPU-time consumption by 33% **25%**

Outline

1. **Introduction**
2. **CPU Time Profiling** ←
3. **Empirical Complexity**

Partially based on these papers:

- *Evaluating the accuracy of Java profilers*, Mytkowicz et al., PLDI 2010
- *Measuring empirical computational complexity*, Goldsmith et al., ESEC/FSE 2007