Code Profiling

Performance Analysis

- Measure the behavior of a program as it runs
- Measurement through a wide variety of techniques, such as hardware interrupts, code instrumentation, operating system hooks, and performance counters
- Analyze the frequency and duration of function calls and instructions
- The analysis can be in the form of
    1. Trace
        - Stream of recorded events
        - Size of trace is proportional to program execution time
        - More suitable to analyze parallel programs
            - Gives information on events such as message wait or synchronization
        - Tools such as TotalView and vTunes Analyzer
    2. Profile
        - Statistical summary of events
        - Size of summary is proportional to the code size
            - More practical than trace
        - More suitable for sequential programs
        - Tools such as `gprof`

Profiling goals

- Investigate run-time behavior of code at different points
    - Check the time taken by different instructions from machine language to high-level functions
    - The time could be in actual time or the number of calls to the instruction

Profiler Types

- Flat profiler
    - Compute the average call times, from the calls
    - Do not break down call times based on called function or context
- Call-graph profiles
    - Show the call times and frequencies of functions
    - Show call chains based on the called functions
    - Do not show context
    - Dynamic call graph
        - Record of an execution of the program, as output by profiler
        - Describes only one run of the program
    - Static call graph
        - Intended to profile every possible run of the program

    - Computationally undecideable; so static call graph algorithms are just overapproximations

gprof

- Produces an execution profile of a program
  - Effect of called functions is incorporated in the profile of caller
  - Data is taken from a call graph profile file generated by programs that are compiled with the option `-pg` with `gcc` or `g++`
  - Default name for the profile data file is `gmon.out`
- Reads and correlates symbol table from the executable image (`a.out`) with the graph profile file