Profiling & Performance Tuning CPU Programs

Paul Preney, OCT, M.Sc., B.Ed., B.Sc.
preney@sharcnet.ca

SHARCNET
University of Windsor
Windsor, Ontario, Canada

Nov. 2, 2022

Table of Contents

- Overview
- Profiling

Overview: What is Profiling?

What is profiling?

- Is a measuring metrics about how program run(s).
  - e.g., RAM use
  - e.g., execution time
  - e.g., instruction use
  - e.g., to aid program optimization

NOTE: Profiling does not analyse source code —it collects data based on program runs, so:

- every program run can produce different metrics
  - i.e., it is important to ensure that program runs sufficiently run all desired-to-be-measured code paths
- if the number of metrics collected is too small, the results will be of lesser/little value

Table of Contents

- Overview
- Profiling

Profiling: Prepping Your Code

In order to obtain useful metrics about your program's source code compile your program to have debug information.

Often one needs to profile optimized code: to keep generated code efficient, one needs to separate the debug information from the compiled code.

Profiling: Prepping Your Code (con't)

Commands to run to separate debug code from the compiled executable:

- Compile your code producing an executable with debug information, e.g.,
  - `gcc -O3 -march=broadwell -ggdb3 -o prog.exe prog.c`
    - GCC debug options: `-g{1,2,3}` or `-ggdb{1,2,3}`
    - Other compilers: `-g{1,2,3}`
- Extract the debug information from the executable, e.g.,
  - `objcopy --only-keep-debug ./prog.exe ./prog.exe.debuginfo`
- Strip the debug information from the executable, e.g.,
  - `strip --strip-debug --strip-unneeded ./prog.exe`
- Set executable's debug information location to be the debug information file, e.g.,
  - `objcopy --add-gnu-debuglink=./prog.exe.debuginfo ./prog.exe`

Profiling: Prepping Your Code (con't)

Debug Information

Realise:

- leaving debug information in the executable can negatively affect performance
- if the "3" level results in too large an executable, decrease the number
  - The lower the number the less debug information is stored. Sometimes this matters, sometimes it does not.
  - If disk space is not an issue, prefer using higher-numbered values for the debug option.

Profiling: Using `perf stat`

Given:

- a program executable `program.exe` in the current directory
- the program needs two arguments `arg1 arg2`

Some `perf stat` commands:

- `perf stat ./program.exe arg1 arg2`
- `perf stat -d -d -d ./program.exe arg1 arg2`
- `perf stat -d -d -d -r 3 ./program.exe arg1 arg2`

Profiling: Live Demonstration

Live Demonstration.
(Due to time, further slides have been removed.)