BOOTCAMP-2018@DAY-5: Writing Cache Efficient Code

(Courtesy: CAR3S@CSE)

Welcome to day-5 of the bootcamp. Today, you will be writing cache-efficient code to improve your code performance.

You will be using following tools or utilities to see the effects of your code at the different levels of a typical cache hierarchy.

(i) Cachegrind: Cachegrind simulates how your program interacts with a machine's cache hierarchy and (optionally) branch predictor.

To install valgrind along with cachegrind use the following commands

sudo apt-get install valgrind

To run cachegrind on a program say a “toy-prog” run the following command:

valgrind --tool=cachegrind toyprog

Cachegrind also writes more detailed profiling information to a file. By default this file is named cachegrind.out.<pid>. To get a function-by-function summary of your code, run: ***cg_annotate <filename> // where filename is the name of output file generated by valgrind***

To get a *line by line counts* use cg_annotate with --auto=yes

cg_annotate cachegrind.out.<pid> --auto=yes

(ii) ***Perf tool:*** It is a profiler for your program running on an OS

To install perf tool use the following command.

sudo apt-get install linux-tools

To profile your program using perf tool use the following command

sudo perf stat -e <events> toyprog

Following are the events of interest that can be used to see the cache behavior:

L1-dcache-loads
L1-dcache-load-misses
L1-dcache-stores

L1-dcache-store-misses
L1-dcache-prefetches
L1-dcache-prefetch-misses
L1-icache-loads
L1-icache-load-misses
L1-icache-prefetches
L1-icache-prefetch-misses
LLC-loads
LLC-load-misses
LLC-stores
LLC-store-misses
LLC-prefetch-misses

Example: ***sudo perf stat -e LLC-loads-misses,LLC-load ./toyprog***

(iii) make your Makefile (ignore it if you know how to use make)

The Makefile is the key to the build process. In its simplest form, a Makefile is a script for compiling or building the "binaries", the executable portions of a package. The Makefile can also provide a means of updating a software package without having to recompile every single source file in it, but that is a different story (or a different article).

Steps to follow if you have not followed before:

1. If make is not installed in your system, then install it by typing **sudo apt-get install make**
2. You have been provided with ***Makefile*** that compiles all the source codes provided. (No need to understand what's written inside). Just type **make** in terminal for compiling your code.