

Writing fast code for Linux!

...

Vancouver Linux Users Group

April 6, 2024

Vladimir Nikolic

whoami

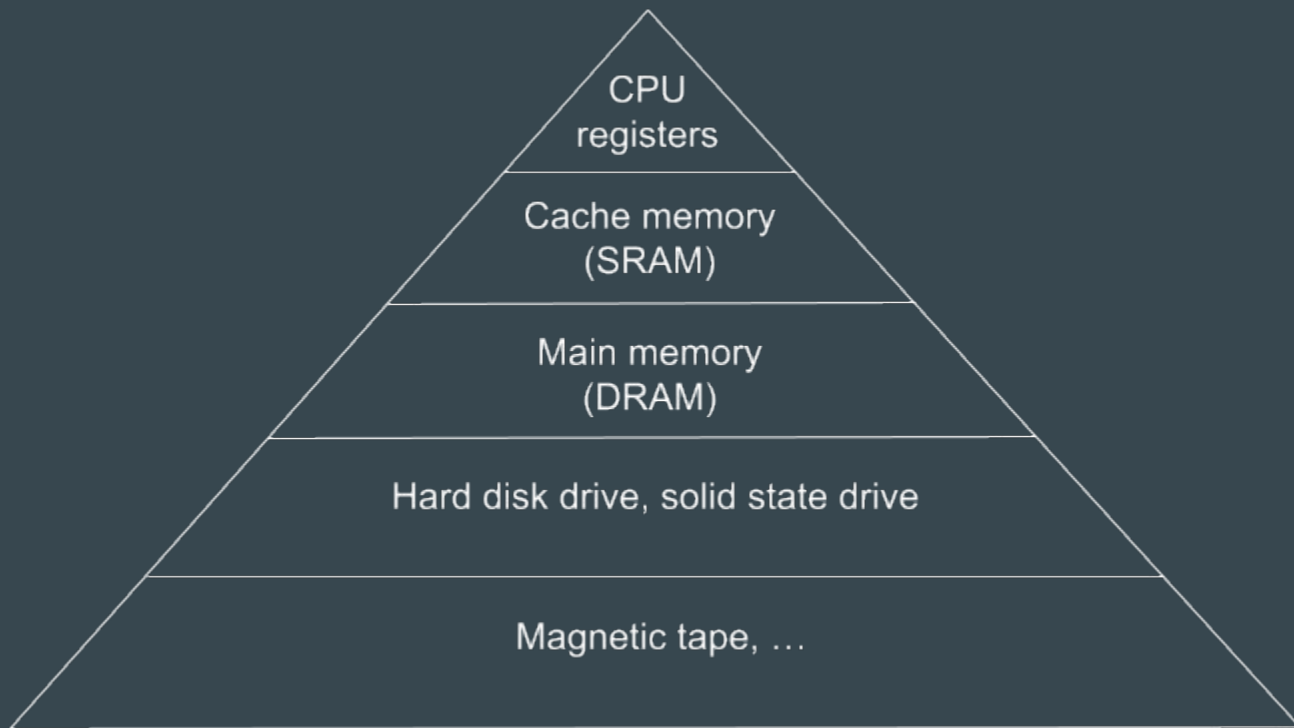
- Bioinformatician and Software Engineer
- Professionally in scientific computing for 6 years
 - Institute of Physics Belgrade
 - Institute for Systems Biology, Seattle
 - Canada's Michael Smith Genome Sciences Centre, Vancouver
 - HTuO Biosciences, Vancouver
- I nerd about software performance and science!
- Computational sciences are almost exclusively done on Linux. A great place for Linux nerds

Before we start

- Short questions fine during presentation, keep long ones for after
- We cover optimizations for utilizing the hardware, the kernel, and the compiler capabilities better; data structures and algorithms are not covered
- How many people have used C/C++ professionally, or used them at all?
- Feel free to follow along, demo files at:

github.com/vlad0x00/vanlug-2024

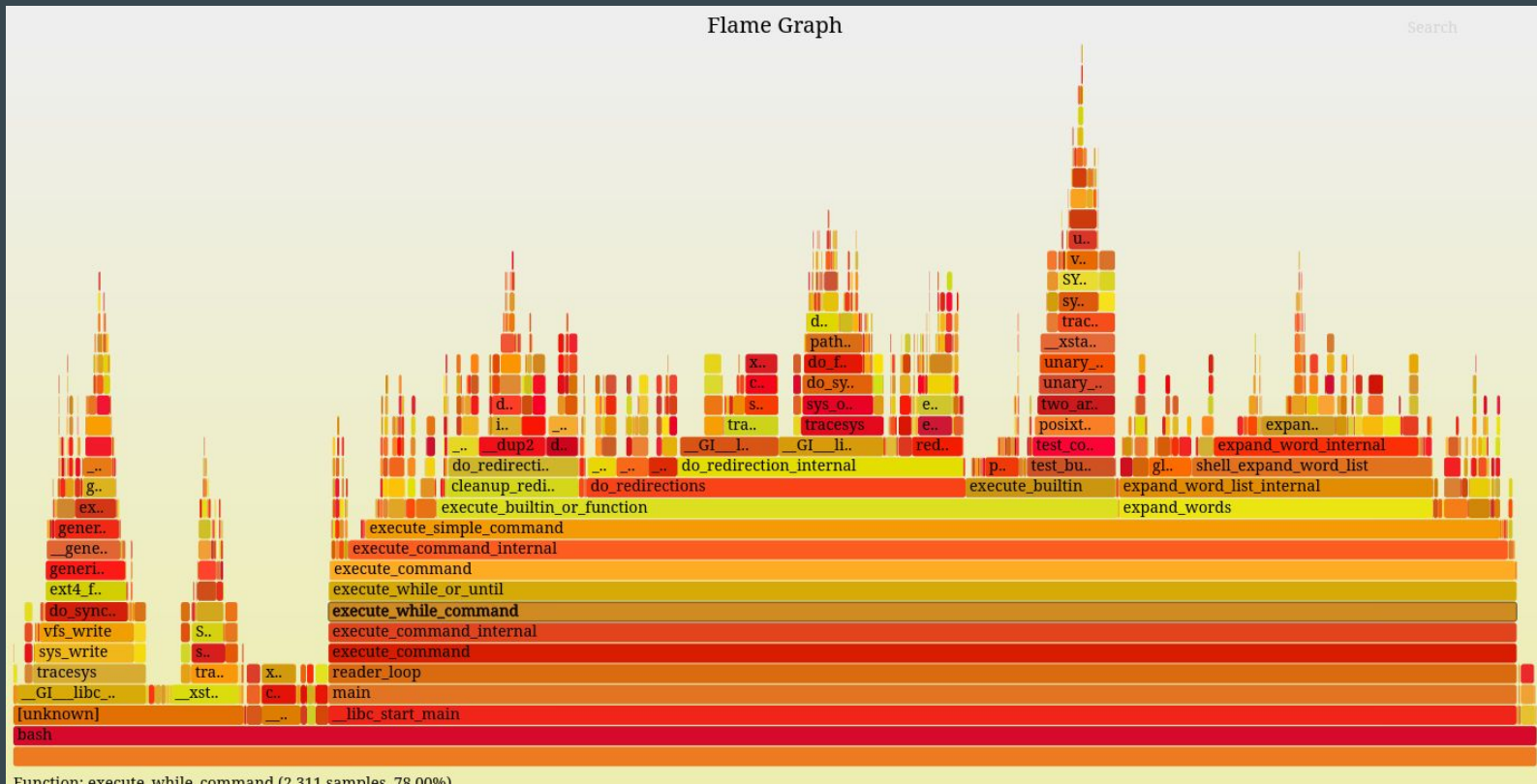
Memory hierarchy



Example 1: Profilers

- Flame graphs: <https://brendangregg.com/flamegraphs.html>
- gperftools: <https://github.com/gperftools/gperftools>
- Score-P: <https://www.vi-hps.org/projects/score-p/>

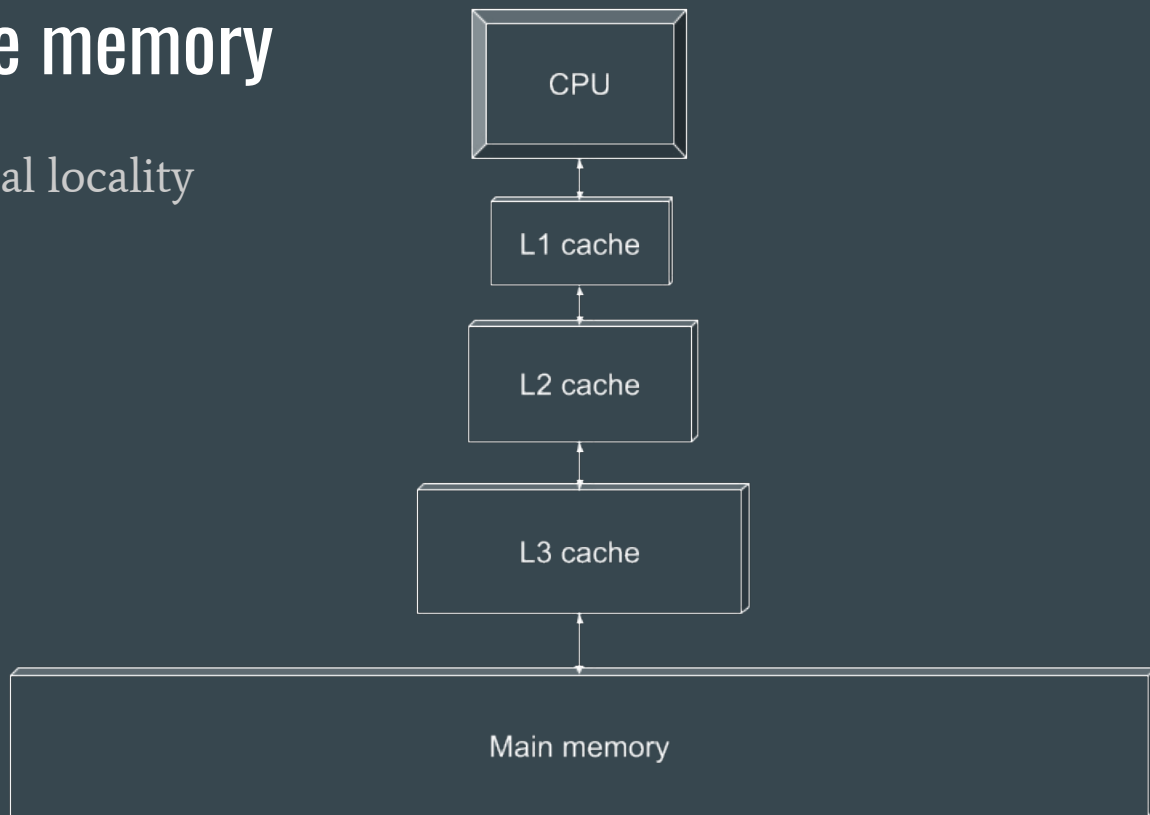
Example 1: Profilers



<https://github.com/brendangregg/FlameGraph>

Example 2: Cache memory

- Temporal and spatial locality



Example 3: fadvise

- https://linux.die.net/man/2/posix_fadvise
- Hints the kernel about the file access patterns

Example 4: Cache bypass

- Bypass the cache and write data directly to RAM

Example 5: perf

- A very elaborate Linux for instrumenting CPU performance counters (hardware level)
- `perf list`

Example 6: /dev/shm

- File storage/operations in RAM; shared between all processes
- Great for temporary file storage, e.g. pipelines

Example 7: Compiler flags

By default, GCC and Clang leave a lot of optimization on the table (even with -O3)

<https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html>

Meta flags:

- march

- ffast-math

- * -funsafe-math-optimizations

- flto

- fprofile-generate, -fprofile-use

Example 8: godbolt.org

Example 9: OpenMP

- High level compile directives for multithreading: let the compiler know that it can parallelize parts of your code

Non-Uniform Memory Access (NUMA)

- lstopo

Q&A

- Also feel free to talk to me about performance questions you have not covered in this presentation

Contact

- vlich.net
- x.com/ersteregen (mostly sh*tposting)
- linkedin.com/in/vladimir-nik