CSE306 Software Quality in Practice

callgrind overview

POST expectations

Piazza post @214

Valgrind

"val-grinned" - the gate to Valhalla

A suite of tools (see http://valgrind.org/info/tools.html)

Memcheck "detects memory-management problems"

Cachegrind "is a cache profiler"

Callgrind "is an extension to Cachegrind. It provides all the information that Cachegrind does, plus extra information about callgraphs."

Massif "is a heap profiler"

Helgrind "is a thread debugger which finds data races in multithreaded programs"

DRD "is a tool for detecting errors in multithreaded C and C++ programs"

callgrind

valgrind --tool=callgrind [opts] prog [opts]

[opts] callgrind options

[opts] program options

callgrind_annotate

Helps to make sense of callgrind data.

callgrind_annotate --inclusive=yes --tree=both --auto=yes callgrind.out.<PID>

https://web.stanford.edu/class/archive/cs/cs107/cs107.1196/resources/callgrind

To redirect output to a file

You can redirect the output of a command to a file using '>'. For example, to redirect the output of 'ls' to a file named 'abc':

```
ls > abc
```

Note: '>' creates the file if it does not already exist, and overwrites it if it does (without warning).

Remember this code from CSE220

Why Memory Performance Matters

```
void copyij(int src[2048][2048],
            int dst[2048][2048]) {
    for (int i = 0; i < 2048; i++) {
        for (int j = 0; j < 2048; j++)
            {
             dst[i][j] = src[i][j];
        }
    }
}
```

3.8 ms

```
void copyji(int src[2048][2048],
            int dst[2048][2048]) {
    for (int j = 0; j < 2048; j++) {
        for (int i = 0; i < 2048; i++)
            {
             dst[i][j] = src[i][j];
        }
    }
}
```

72.2 ms

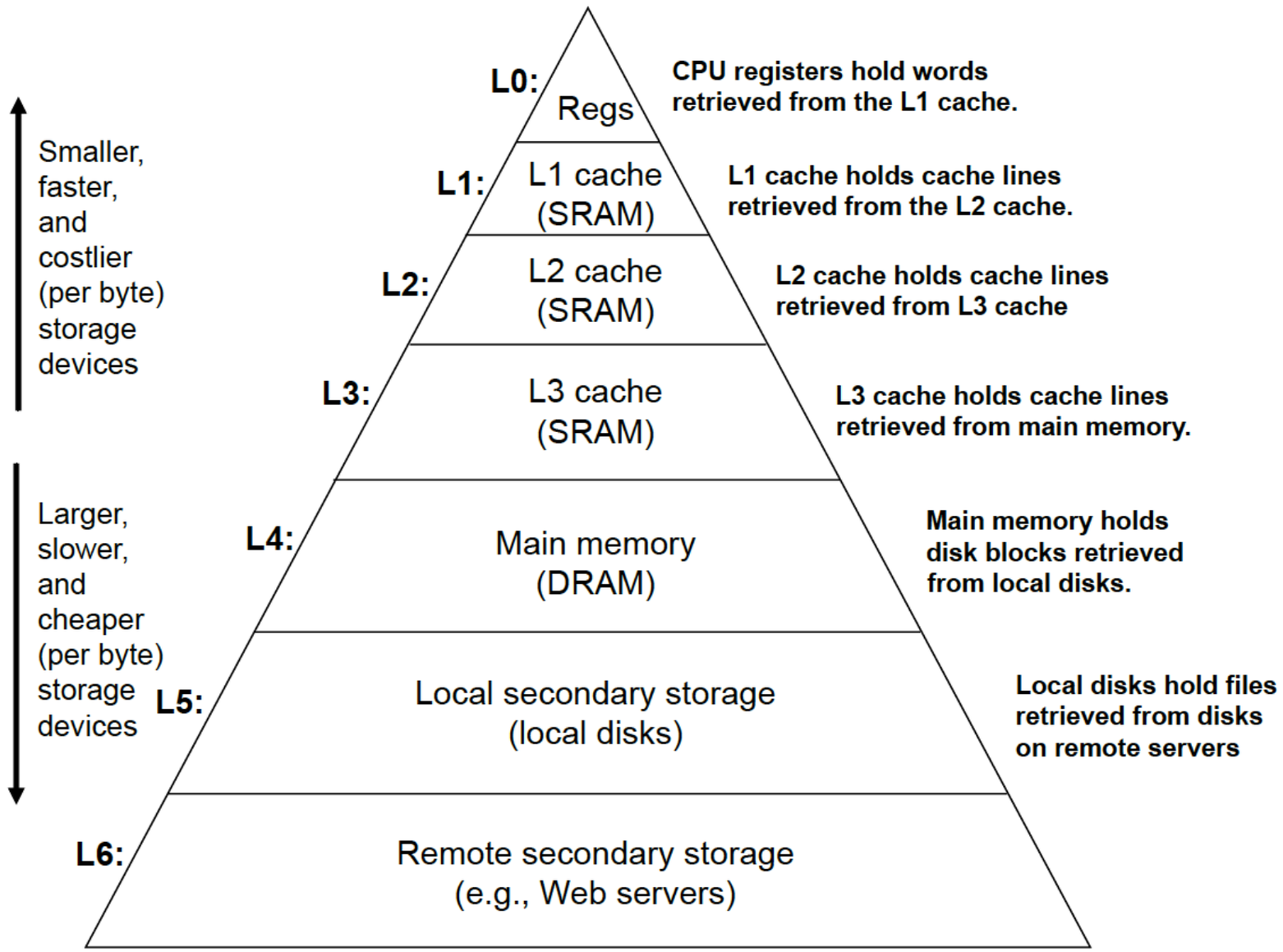
All that changed is the order of the loops!

Demo

Use callgrind to explore CSE220 code performance

- Helpful resource:
- https://accu.org/journals/overload/20/111/floyd_1886/

The Memory Hierarchy



EXERCISE

- Work through the gprof exercise from last Monday (April 8) but now using callgrind instead (or in addition to).
- Use your existing repo from earlier exercise

valgrind options

--tool=callgrind

--dump-lines=yes

--dump-instr=yes

--trace-jumps=yes

--simulate-cache=yes

Callgrind_annotate options

--auto = yes

--tree = both

--include = absolute path to source code

callgrind resources

official manual : valgrind.org → look at documentation.

accu.org/journals/overload/20/111/floyd_1886

web.stanford.edu/class/archive/cs/cs107/cs107.1196/resources/callgrind