Lab 1 Bonus: Improving Cache Performance

Due Feb 11 by 12:01pm **Points** 1 **Available** after Feb 1 at 12am

*Note: To pass this bonus task you must first pass the mandatory part of **Lab 1**.*

In this assignment you should try to improve the cache performance of a given program example. The program is a simple matrix multiplication. Try to rewrite the program so that the cache miss ratio is decreased. Use some of the methods discussed in the course material to reduce the miss ratio of the program.

You may not change the size of the matrix or use less precision, but otherwise you are free to experiment with loop-indices, matrix-transponation, blocking etc. The multiplication should be correct, though. Also, you must use `gcc` when compiling it.

To play with the optimizations, change the matrix parameter `SIZE` so that the unoptimized version runs for about 20 seconds. Then try to minimize the miss rate without changing the `SIZE` parameter.

The original program is located in `~/avdark/lab1/multiply`.

Copy `multiply.c` and `Makefile` to a working directory and hack away!

The original code has built to support for verifying the results. You may activate the verification code using the `-v` option to the binary. It is a good idea to run the verifications on the host machine instead of the simulated machine to save some time.

Note: When running in Pin, the pintool instrumentation affects the execution time of the application, which can be misleading. Performance (time) measurements should be made when running on the host machine. Instead of using Pin, you can also use the Linux `perf` tool to measure the cache misses of the application while it is running on the real cache of the system:

```
host$ perf stat -e cache-misses,cache-references,L1-dcache-load-misses,L1-dcache-loads -B ./multiply
```

Evaluation

- Show the optimizations done to the multiplication program and explain them.
- Verify all your numbers by running simulations and the different versions of the multiplication program.
- Show that your implementation is an improvement.