Programming for Performance **2013**

Programming Contest (Optimization) on Stampede

Download, extract and inspect the code (http://progforperf.github.com/test.tar). Your task is to optimize the function called superslow in the file **comp.c**. The function runs over an n x n matrix and performs some computation on each element. In its current implementation, *superslow* involves several optimization blockers. Your task is to optimize the code.

Edit the Makefile if needed (architecture flags specifying your processor). Running *make* and then the generated executable verifies the code and outputs the performance (the flop count is underestimated, since the trigonometric functions are ignored) of *superslow*. Proceed as follows:

(a) Identify optimization blockers and remove them.

(b) For every major optimization you perform, create a new function in comp.c that has the same signature and register it to the timing framework through the *register_function* procedure in *comp.c*. Let it run and, if it verifies, determine the performance.

(c) In the end, the innermost loop should be free of any procedure calls and operations other than adds and mults.

(d) When done, rerun all code versions also with optimization flags turned off (-O0 in the Makefile).

(e) Create a table with the performance numbers. Two rows (optimization flags, no optimization flags) and as many columns as versions of superslow. Briefly discuss the table.

(f) Submit your comp.c file along with the brief report mentioned in (e) on Github.

Mention what speedup do you achieve?