CS194-2: HW2a in-class grading rubric

Brian Kazian and Mark Hoemmen

5. Oct. 2007

1. Experimental setup (10 points)
   (a) Description of machine(s) used
      i. Described CPU type and other relevant hardware information, in as much detail as possible (e.g., "x86" is not enough)
      ii. Used `/proc/cpuinfo` in the PBS script or the benchmark executable (in order to distinguish compute nodes from front-end nodes), or specified the type of compute node in the PBS script
   (b) Description of timer
      i. Type of timer (e.g., `gettimeofday()`, x86-64 cycle counter instruction, IA64 cycle counter instruction?)
      ii. Granularity of timer ("worst resolution over n trials")
   (c) Software information
      i. Operating system(s) (name and version)
      ii. Compiler(s) (name and version)
      iii. Compiler flags
      iv. If performance depends on libraries, what libraries were used, and what versions of each?
   (d) Benchmark environment
      i. Batch scheduler used?
      ii. If not, was the load on the machine measured or described in some way? What other processes were running?
   (e) Other
      i. List of changed files
      ii. Special instructions for running benchmark (if applicable)
2. Optimization approaches (40 points)
   (a) Adequate description of each optimization approach tried
   (b) Written justification for each approach, using either heuristic reasoning or quantitative performance models

(c) Tried more than one optimization approach. Examples (these aren't necessarily good or bad; we just provide some possible choices):

- Transposed A, and used recursion for cache blocking
- Transposed A and used register blocking
- Combined register blocking and cache blocking
- Rearranged loop indices (in the triply nested loop), and unrolled the inner loop
- Transposed A, and implemented Strassen's algorithm

(d) When possible, tried combining different approaches, but also tried each one separately, so as to quantify their interaction

(e) Experimental methodology

i. Sufficient variety of input sizes (including both powers of two and other kinds of numbers)

ii. If optimization techniques depended on parameters, were a sufficient variety of parameters tried?

3. Results (40 points)

(a) Visualization of results

i. Included chart(s) or plot(s) (preferred) illustrating benchmark results

ii. Compared results against other implementations, both given and optimized (such as the Intel MKL BLAS) if possible (some students may have chosen a hardware platform for which an existing optimized implementation is hard to find)

iii. Called the Intel MKL BLAS by its real name instead of "the BLAS" (the BLAS is an interface standard, *not* an implementation)

iv. Charts / plots are labeled correctly (axes, title, units)

v. Charts / plots not too cluttered, so that you can easily extract the relevant information

(b) Discussion of results

i. Explanation of the effect each optimization approach had on performance

ii. Discussion of possible reasons for observed performance

4. Conclusion (10 points)

(a) Summary of results

(b) Illustrates any possible bottlenecks in the code (if applicable)?

(c) Describes any further optimizations that could be made?