ECE 574 – Cluster Computing
Lecture 11

Vince Weaver

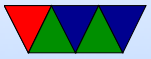
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

6 October 2015

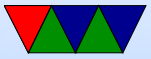
Announcements

- Homework #4 was posted
- Midterm on the 20th?



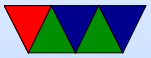
HW#4 Notes

- Pthread programming
- Provided a "solution" that's not great, but it does a lot of the crazy pointer math. For example how do you pass a 3x3 integer array in C? Lots of hassle.
- Most of it is just to get 2-way parallelism by doing sobel_x and sobel_y in parallel
- Adds use of `PAPI_get_real_usec()`
- The "something cool" is to do more fine-grained parallelism. This will involve more low-level coding.



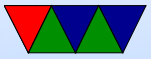
HW#3 Notes

- c/x/y loop (inside unrolled)
  1.407 seconds / 10,413,408
  y/x/c
  (loop ordering) 1.1886 / 4,510,357
- y/x/c/fy/fx
  0.932 / 83,405
  (O3) 0.8196 78,015
  (linear)
  (loop ordering) 0.5814 70,971



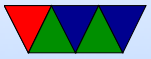
(byte-by-byte) 0.579 66,278

- y/x/c
  1.059s / 45,540
  (-msse3) 0m1.068s 46,068
- x/y/fx/fy
  0m1.178s / 939,198
  (loop unrolling) 0m1.096s 933,907
- y/x/c/fx/fy 0m0.936s / 78,605 L3 cache misses
  (loop unrolling) 0m0.717s / 71,754 L3 cache misses
- x/y/c/fx/fy
  real 0m1.425s / 3,256,597



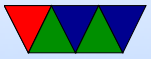
(loop unrolling/ordering) 0m0.842s / 170,851

- y/x/color/i/j
  0.634 / 76,888
  (loop unrolling) 0.404 / 68,637
- x/y
  0m1.604s / 1,802,394
  (loop order) 0m0.781s / 31,538
- y/x/color
  0.649s / 19,125
  (unrolled) 0.620s / 17,769

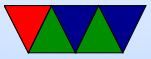


OpenMP Examples

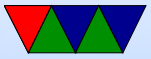
See the course website for a link to a tarball with all the examples.



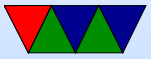
for



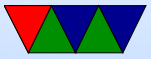
static schedule



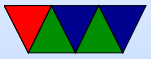
dynamic schedule



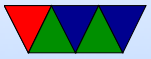
critical



section



reduction

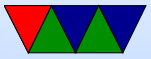


simd reduction

https://software.intel.com/en-us/articles/enabling-simd-in-program-using-openmp40

Actually works with gcc 5 (5.2 on my laptop, haven't tested on the Haswell machine yet).

Look at assembly code to verify it is making SIMD code.
You can use:
`objdump --disassemble-all openmp_simd_reduction`
Also you can use `gcc -S` to generate assembly.

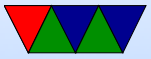


offload

Can offload to GPU or MIC.

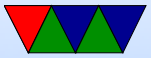
`https://gcc.gnu.org/wiki/Offloading`

Need separate compiler for component. Support really isn't there yet.



Alternate compilers

- gcc isn't the only compiler out there
- clang/llvm
- many commercial compilers over the years
- icc – from intel. can download for free if a student
- sun studio
- portland group



- old SGI compilers
- IBM has own too
- Microsoft Visual Studio