

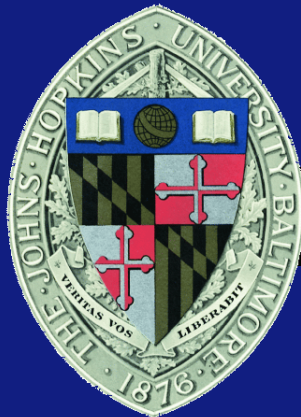
Lecture 5.2

OpenMP: Block Parallelism

EN 600.320/420

Instructor: Randal Burns

13 February 2017



Department of Computer Science, *Johns Hopkins University*

A First Program

- Parallel block
 - `block.c`
 - Runs as many times as there are threads
- How many parallel executors? Which am I?
 - Common parallel programming idiom
 - Used to divide work
- Number of times block runs is determined by the environment
 - Not the program



(Aside) Compiler Optimization

- gcc/g++ has optimization flags
 - Must set to get good performance
- Optimization level $-O^*$
 - $-O0$ (default) = Reduce compilation time and make debugging produce the expected results.
 - $-O1$ = simple optimizations that don't take a lot of compile time
 - $-O2$ = rewrite loops, follow jump pointers, inline small functions, no time/space tradeoffs
 - $-O3$ = vectorize, inline functions, branch prediction
- When debugging, you want to use $-O0$ so the code makes sense
- For performance, $-O3$ to vectorize code to processors



Blocks (braces in C/C++)

- A block is a group of statements, separated by semicolons, enclosed in braces
- Big, important concept:
 - Scoping context for variables (see `noscope.c`)
 - Single entry/exit point
 - Statements in a block run sequentially
- The scoping is super-valuable in OpenMP
 - Variables from outside the block are shared among threads
 - Variables inside block are private to each block and, thus, thread



What's an OpenMP Program

- Parallelization directive
 - `#pragma omp parallel`
 - that executes a block of C/C++/Fortran code in parallel
 - Not a language: all code written in C/C++
- Runtime libraries and routines
 - To access information about parallelization and state
 - Number of threads, thread identifier, etc.
 - Compiler directives turned into library calls for parallel execution
- Environmental variables
 - To configure parallelism
- See `block.c`



The Toolchain

- No toolchain
 - Add `-fopenmp` to gcc command line. This activates linker to libopenmp
 - `#include "omp.h"` to import symbols into your source code
- *Warning to MAC users*
 - clang has open mp support (which apparently just works)
 - May need to install Intel open-mp runtime
 - Or can install g++ through ports/brew

