The Thread Building Blocks
Lab Time Hands-on

SPD Course

2017-18

Massimo Coppola

TBB Installation

- Download latest TBB either as binary package (for your OS) or as source package
  - unpack in your working directory
  - If starting from source, check out the readme and compile it
- Be sure to have a suitable tool chain (e.g. recent C++ compiler and linker version)
  - https://www.threadingbuildingblocks.org/system-requirements
- ***EITHER*** Set up your environment for command line usage
  - Look for the script …/bin/tbbvars.* (sh / csh versions)
  - Add your install directory inside the script by editing it
  - Call it from a shell every time you need TBB (or add to .profile)
- ***OR*** follow the instructions in TBB readme files to use GUI-based coding environments
  - Check your own : Eclipse, MS Visual Studio, Xcode
  - Note that many configurations work besides those officially listed - Eclipse with CDT plugins is known to support TBB on Linux and OS X

Details

- Set up the install dir of TBB inside the tbbvars script
- Call the script in your bash/csh profile passing arguments <architecture> <os>
  - On ottavinareale: intel64   linux

Exercise 1

- Write a parallel for with 1d and 2 ranges in steps
  - Without any actual computation (leave operator() empty)
  - Without an actual value passed, just the indexes
  - Passing an initialized array of given type: have the operator() perform some computation
  - Add another array to store the results
- Take the 2d version, add a mandelbrot function and compute over a 2D range that spans a rectangle in the complex plane
  - result is an integer = number of iterations performed before detecting divergence

Exercise 1

- Some code snippets are found here http://didawiki.cli.di.unipi.it/doku.php/magistraleinformaticanetworking/spd/2018/mandel
  - Code for the Mandelbrot function
  - Code for saving array data as .ppm files you can view
- Other ideas:
  - compute any long, iterative function and write the result somewhere, so that the compiler does not optimize it away

Exercise 2

- Starting from the Mandelbrot example do the following
  - Set the investigated area to cover at least part of the mandelbrot set
  - Raise the number of pixels and the number of iterations of a couple of order of magniture
    - Set TBB to only use **one** thread – see example there: https://software.intel.com/en-us/node/506296 where explicit task scheduler init allows to control the thread number
    - Experimentally find values that cause a sequential running time in the range in between 15 and 60 s
    - Allow TBB to create TN>1 threads, measure the difference in execution time with varying TN