

Parallel Computing for Science and Engineering (SSC 374C/394C)

02/22/2018

Instructors:

Lars Koesterke & Charlie Dey



THE UNIVERSITY OF TEXAS AT AUSTIN
Texas Advanced Computing Center

General Home Work Guidelines

What you have to do!

- In your work directory:
- Create a directory pcse with subdirectories: **hw0**, **hw1**, ..., **ex1**, ...
 - Place your “final products” in these directories
 - Expect us to do:
 - `cd hw1; make clean; make hw1` or
 - A readme file that explains 'how to compile', and 'how to run'
- Please print the results of your home work out!

General Remarks

- Resources (Stampede and Maverick)
 - may be in preventive maintenance, typically Tuesdays (but not every Tuesday)
- Resource may face an emergency and may be inaccessible at any time.
- Also consider that you may have to wait in the queue
- Do not wait with your home work to the last minute!
- Let us know if you cannot make the deadline

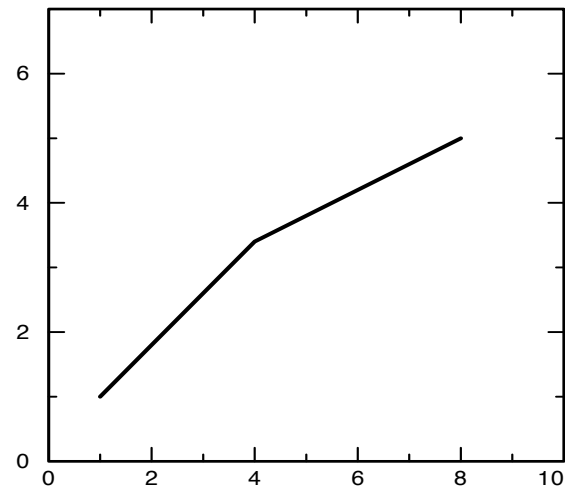
The most difficult part!

- A “home work” in the grading sense may comprise **several** assignments (tasks) or example tasks
- An example is something you execute following the instructions. No (or very little) coding effort
- An actual Home Work is a “coding” assignment
- Months ago I started creating material, now we are bundling material up
- Assignment #1 (for the first grade) will contain ex1 and hw1

What's wrong with this?

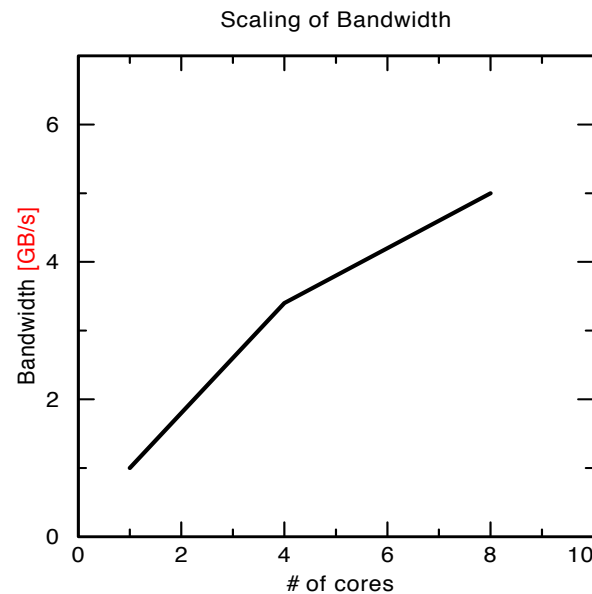
Let's get a bit organized

What's wrong with this?



What's wrong with this?

- A figure (or table) needs a caption
- The x and y-axis have to be labeled
- The labels need units, if applicable



What's wrong with this?

- The experiment ran 3 times. The results and the average are:

	#1	#2	#3
Run-time	17.612345	17.167890	18.112345
Average	17.63086		

What's wrong with this?

- Units were missing
- The scatter is about 0.5 seconds
- Timings below 0.01 seconds are apparently meaningless
- Apply appropriate rounding!

	#1	#2	#3
Run-time [s]	17.61	17.17	18.11
Average [s]	17.63		

How to get around in Linux

- Editors
 - vi (vim) or emacs
- Basic commands
 - cd, pwd, ls, rm, mv, cat, more, mkdir, rmdir, hostname, top, echo
 - ps, kill (later to identify and kill run-away processes)
 - icc and ifort for compiling
- Man pages
 - Example: man ls
- Logon to a remote resource with ssh (see Kent's intro)
- Google it!
 - Example: google for 'linux explain ls'
 - Look for tutorials
- Ask a friend, a co-worker, a fellow student
- Ask us



Multidimensional arrays in C

- There are at least three different ways of creating a 2d array in C

Multidimensional arrays in C

- There are different ways of creating a 2d array in C
 1. Row-by-row: multiple malloc calls
 2. Whole matrix + a vector that points to the beginning of each row
 3. Whole matrix with manual index calculation
 - Square matrix a with (n x n) elements
 - $a_{i,j} :: a[i*n + j]$
- Array has to be allocated contiguously
 - one malloc call → variants 2 and 3
- In your C/C++ code please use variant 3
- In your Fortran code you will 'automatically' use variant 3