CS-E4690 – Programming parallel supercomputers D

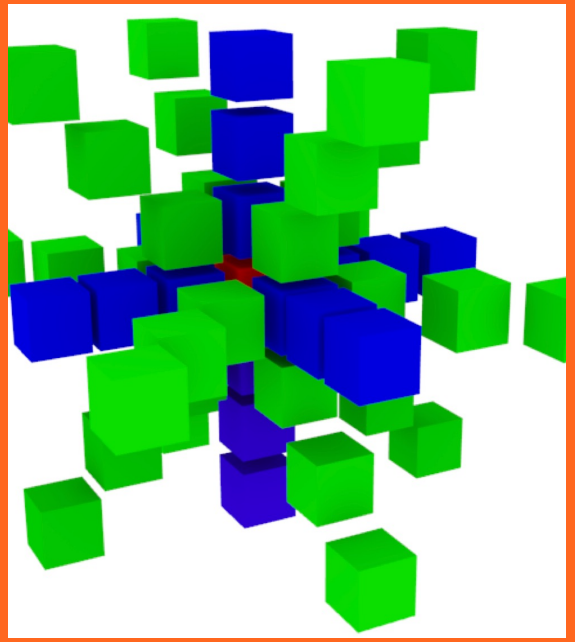
5th lecture

Maarit Korpi-Lagg

maarit.korpi-lagg@aalto.fi

21.11.2023

Aalto University
School of Science



Lecture 5

Combining distributed memory and shared memory programming concepts

- **Course updates:** 5 min
- **Quicksort stability card game:** 10 min
- **Key concepts recap - old and new** (30 min)
- **Break** (max 15 mins)
- **Example codes cntd.** (10 mins)
- **Exercise Sheet 5 tasks and tips** (30 mins)
- **Wrap up** (Feedback; 5 mins)

Break-down of learning objectives

Lecture1

Introduction to the current HPC landscape

Understanding how this course fits into that

Establishing understanding of the learning outcomes, specifically answering the question: "What are programming during this course?"

Lecture2

Learning basic definitions and taxonomies

Understanding the importance of the "network"

Learning basic performance models

Understanding the concept of a well-performing software in large-scale computing.

Lecture3

Becoming knowledgeable of the modern landscape of distributed memory programming

Understanding why in this course we will concentrate on low-level programming models

Getting acquainted with MPI: basics and synchronous and asynchronous point-to-point communication

Break-down of learning objectives

Lecture4

Learning more about MPI:

One-sided point-to-point communications

Collective communications

Lecture5

Programming MP hybrid architectures

Becoming knowledgeable of the spectrum of options

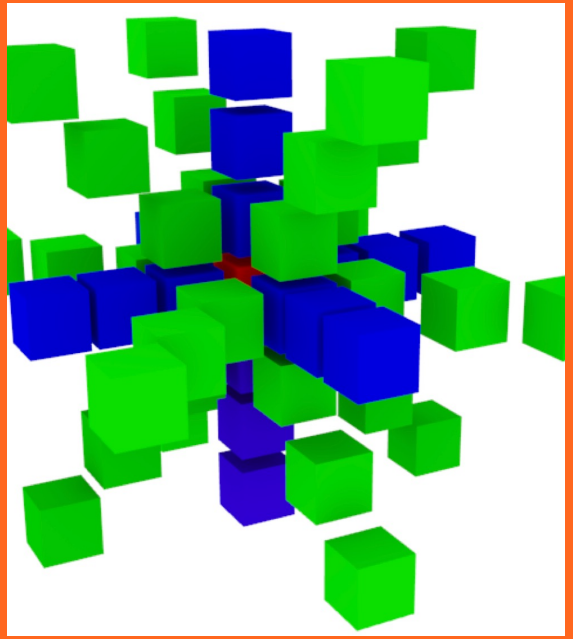
Understanding efficiency issues

Lecture6

Programming hybrid architectures with accelerators

Acquiring knowledge of CUDA-MPI programming model

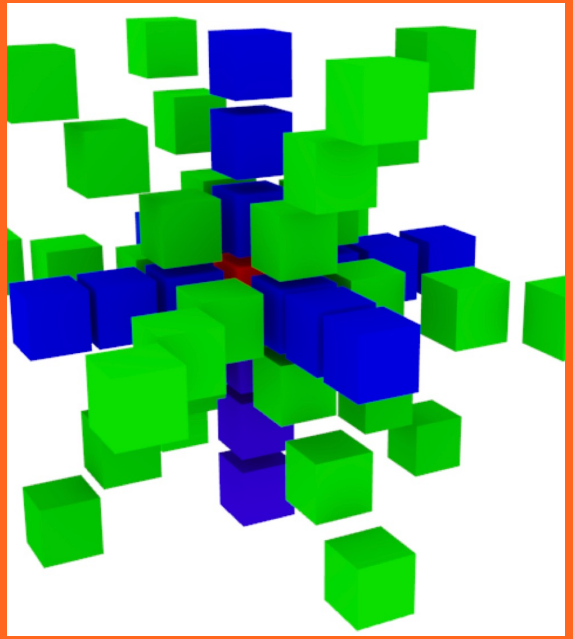
Course updates



Aalto University
School of Science

Card game

Task: Demonstrate that Quicksort is unstable



Aalto University
School of Science

Quicksort stability card game;

Recurse (until all subarrays are sorted)

- Choose pivot (P) to be the leftmost index of (sub)array
- When pivot is on the left, compare it to the rightmost element; if P>R, swap and increment L index; otherwise decrement right index
- When pivot is on the right, compare it to the leftmost element; if P<L, swap and decrement R index; otherwise increment left index
- When P, L, R are pointing to the same index, divide into subarrays based on P, which value is now considered sorted (left out).

Try out at home! Make small pack of cards with at least two same numbers and record your findings about the position of the two

Repetition of key concepts - old and new

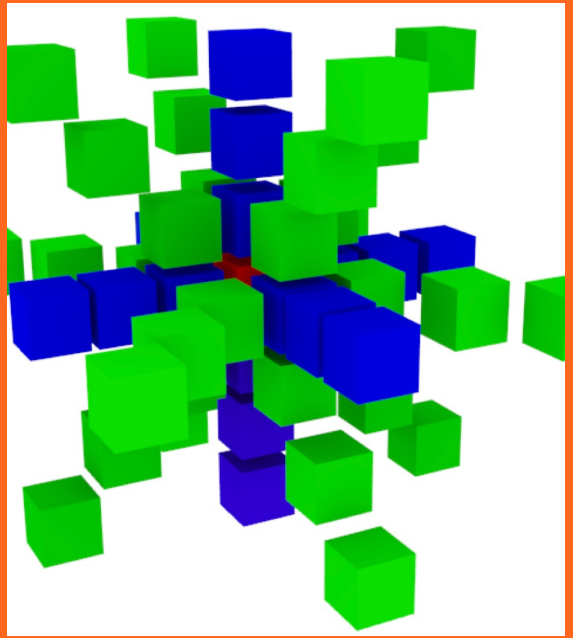
Aim: to explain the key concept in short

Discuss 1-5 mins in row-wise groups

Lecturer selects groups to answer and comment.

Model answer

Feedback with RED POST-ITS ONLY



Aalto University
School of Science

What is this?

Many correct answers.

Try to find at least two!

$$
\begin{aligned}
&I\left(x_i, y_j, t^n\right) \\
&\approx \left(I\left(x_{i-1}, y_j, t^n\right) + I\left(x_i, y_{j-1}, t^n\right)\right. \\
&+ I\left(x_{i-1}, y_{i-1}, t^n\right) + I\left(x_i, y_j, t^n\right) + I\left(x_{i+1}, y_j, t^n\right) \\
&+ I\left(x_{i+1}, y_{j-1}, t^n\right) + I\left(x_{i+1}, y_{j+1}, t^n\right) \\
&\left. + I\left(x_{i-1}, y_{j+1}, t^n\right) + I\left(x_i, y_{j+1}, t^n\right)\right)/9
\end{aligned}
$$

Iterative stencil loop (ISL)

Very basic image blurring operation

Nine-point stencil

Second order Moore stencil

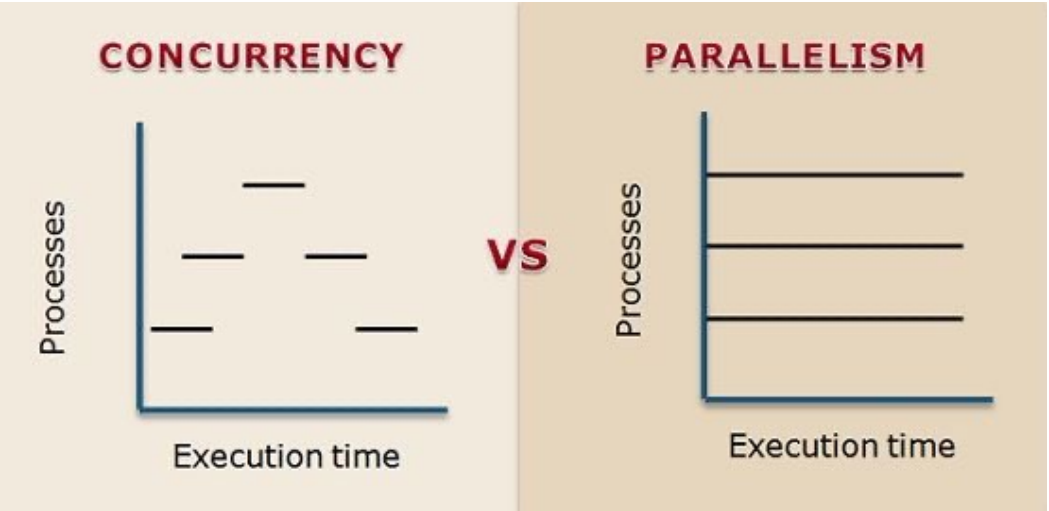
Symmetric stencil

Static stencil

Discuss concurrency versus parallelism with MPI; when and what?

Parallelism: executing (nearly) similar computing tasks simultaneously.

Concurrency: executing different kinds of tasks overlapping with each other.



Parallelism: Share (nearly) identical tasks for many workers (SPMD model)

Concurrency: Overlap communication and memory transactions with computation (non-blocking communication, RMA operations)

What else than MPI could and should be used to optimize for parallelism and concurrency?

Multithreading, using accelerators, various types of co-processors, instruction-level parallelism, MPI RMA programming model within the nodes

Blue means topics that are dicussed and rehearsed in practise during the remaining of the course

Look at Hybrid/hello_class.c

From which line of the code does the parallel execution start/end with MPI?

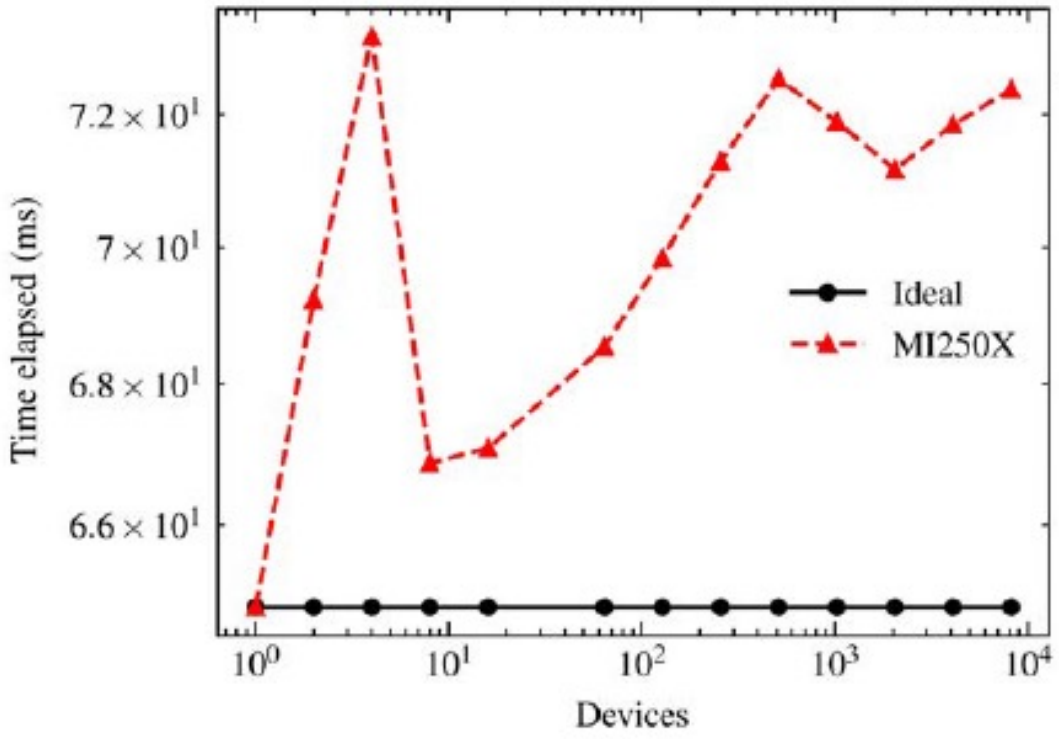
From which line of the code does the parallel execution start/end with openMP?

Look at Hybrid/hello_class.c

First/Last line.

Line 38/40.

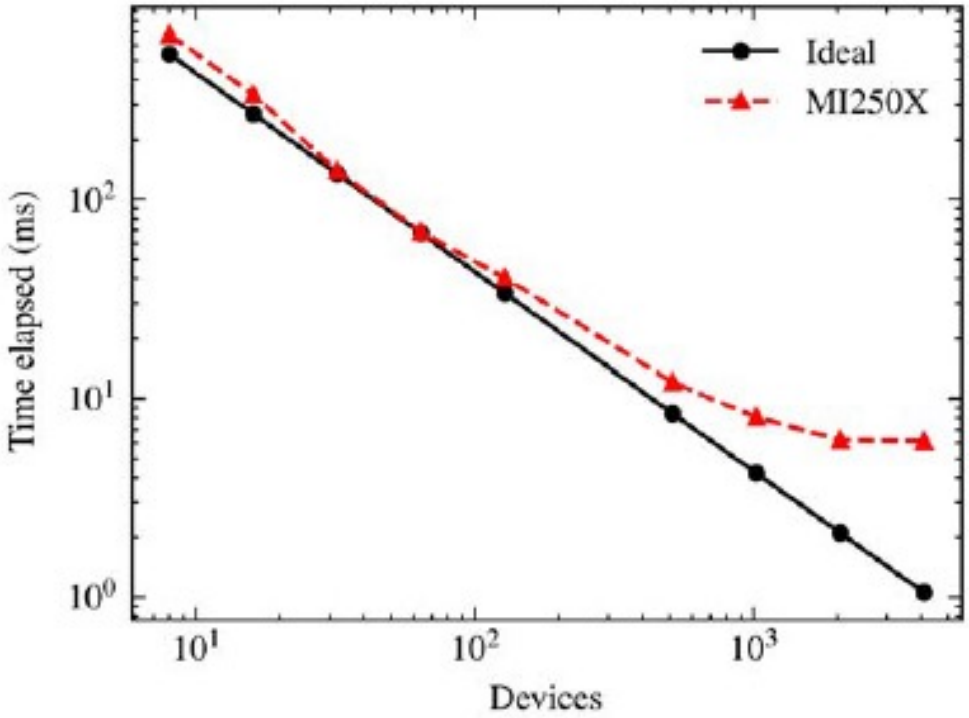
What does the data and graph illustrate?



devices	Ideal	MI250X
1	64.85	64.85
2	64.85	69.22
4	64.85	73.19
8	64.85	66.87
16	64.85	67.08
64	64.85	68.54
128	64.85	69.84
256	64.85	71.28
512	64.85	72.55
1024	64.85	71.89
2048	64.85	71.16
4096	64.85	71.84
8192	64.85	72.38

Weak scaling

What does the data and graph illustrate?



devices	Ideal	MI250X
8	539.50	677.23
16	269.75	336.76
32	134.87	139.31
64	67.44	68.54
128	33.72	40.20
512	8.43	12.05
1024	4.21	8.07
2048	2.11	6.20
4096	1.05	6.12

Strong scaling

Which machine could this be?

Triton

LUMI

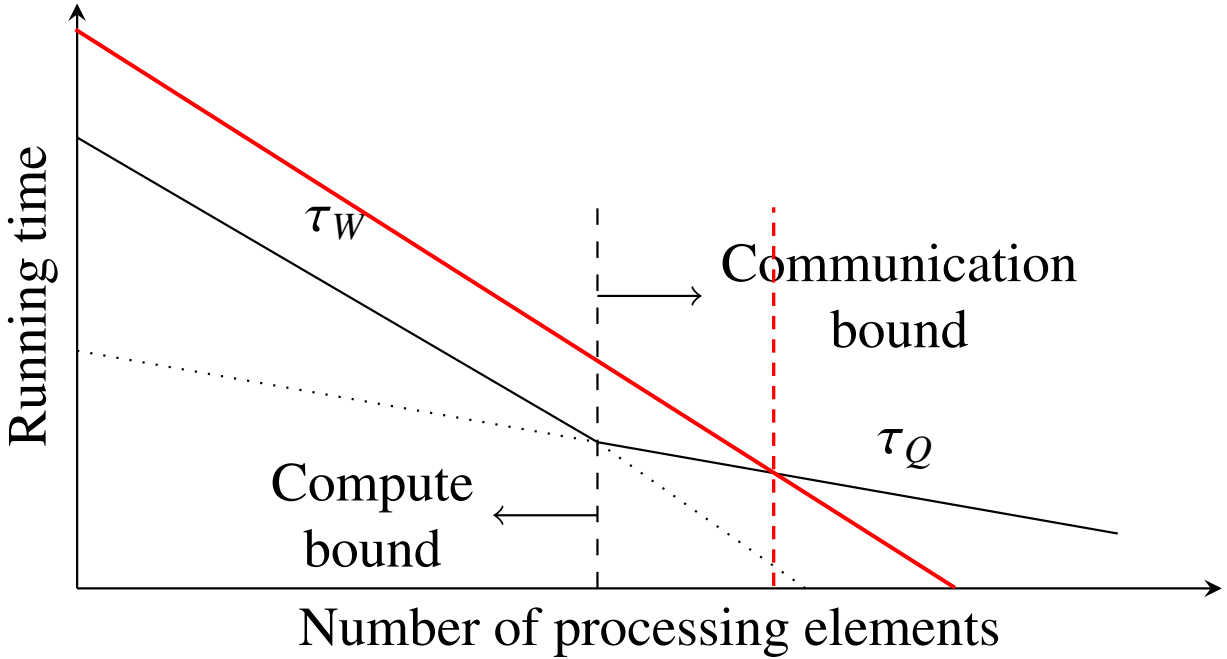
Mahti

Frontier

Puhti

LUMI or Frontier

What does the graph demonstrate?



If speed of computations is increased, then concurrency will be limiting the scale-up even for smaller number of processes.

Remember Sheet 2 trends!

Wohoo!

Next week SCITEC teaching evaluation committee will visit us.

You can give feedback already now!