README

December 6, 2023

1 Exercise 1 - HPC landscape

The first thing that we have to establish is **what are we going to program** computer-architecture-wise during this course. For this learning outcome, we need to map out the HPC landscape.

We undertake this exercise during the first lecture session in the format of a gallery walk; everyone participating will get full points. If you cannot make it, you can also write a learning diary and return it to MyCourses. The diary will be graded by the course personnel. The posters produced during the first exercise session will be the distributed as the correct solution to this exercise sheet.

If you decide to submit this exercise as a learning diary: the excercises have equal weight in the grading. The optimum length of the learning diary does not exceed one page.

Note! Completing this exercise on time (Tue 31st of Oct, 16:00) is required for obtaining Triton account and accessing the course computing resources.

2 Your tasks

Inspect the list of top 500 computers in the world.

2.0.1 1. What kind of trends between HPC computing paradigms can be seen?

Hints: Go to "Statistics" -> "Development over time", and build graphs using variable criteria for "Performance share". You can extend the time bar with the slider on top of the graph, and select/de-select data by clicking on the squares below the graph. Use the widest possible time range.

Guiding questions:

1. What can you observe if you use "Architecture" as a search keyword? Hints: SMP refers to shared memory multiprocessors and MPP to massively parallel processors without shared memory, constellations to grid computing, and clusters to architectures that use both SMP and MPP concepts.
2. What can you observe if you use "Accelerator/CP family" as a keyword?
3. What can you observe if you use "Interconnect family" as a keyword?
4. What can you observe if you use "Cores per socket" as a keyword?

2.0.2 2. How is the "power wall" phenomenon seen in the list?

Hints: see the hints and guiding questions in task 1, and go through the steps to interpret your observations in the light of the power wall phenomenon.

You are also free to use other keywords, if you think they would be more indicative to show the power wall phenomenon.

Inspect the technical specifications of the Triton cluster

2.0.3 3. What kind of processor(s), accelerators, and interconnect does it have?

Hints: from the Triton specs you see that it has many generations of computing cores, spanning over roughly 5 years or so. This gives us some opportunity to inspect how fast important components (clock frequency, co-proc. capabilities, core count, memory, memory bandwidth, interconnect speed) improve over time. In the memory bandwidth specs (DDRx-yyyy) the important number is yyyy, which tells you the number of transactions [unit of million] per second. Multiply by 8 to get the bandwidth in MB/s. Clocking frequenciess you need to search from internet by architecture.

Guiding questions:

1. How has the clock frequency of CPUs changed over time in the hardware options provided by Triton?
2. How has the memory bandwidth changed over time?
3. How has the number of co-proc. capabilities changed over time?
4. How has the ratio of compute capabilities over memory transactions changed?

Choose any of the mobile devices that you carry. Find out its technical specifications from the internet.

2.0.4 4. What kind of processor(s), accelerators, ..., does your mobile device have, how many cores and threads they run, what memory type and what caches they use.

Hints for the group work: please collect specs of different devices, and compare them to the Triton specs.

Hints for the learning diary: you can select only one device, and compare it to the Triton specs.

Guiding question: per each new device, discuss what are the differences to a supercomputer with respect to processor clock frequency, memory, anything relevant you can think of, and write down your observations.

2.0.5 5. Synthesis: What are we going to program during this course?

Guiding questions: based on the material you went through in this exercise

1. what are the programming paradigms we need to use in making efficient parallel code in Triton?
2. what are the main bottlenecks that we need to design our codes to cope with?