TDT4200 Problem set 2

Graded. Counts 10% of final grade.
Deadline: 13/09, 20:00
Answers should be submitted on It's Learning

Task 1 - Miscellaneous theory (10 %)

a) Briefly explain SISD, SIMD, MISD and MIMD and come up with example uses of these techniques.

b) When does it make sense to use `MPI_Pack` and `MPI_Unpack`?

c) What are the advantages of using `MPI_Sendrecv` instead of `MPI_Send` and `MPI_Recv` separately?

d) Show an example of a communication patter in MPI that can lead to deadlock, and explain how the deadlock might occur. Show and explain how the communication can be changed to avoid the deadlock.

Task 2 - Amdahl's law and multicore (10 %)

Background information for this task can be found in: Hill, M.D.; Marty, M.R.;, "Amdahl's Law in the Multicore Era," Computer, vol.41, no.7, pp.33-38, July 2008

For this task, we will assume a simple cost model for the design of multicore chips. We assume that a chip of a given cost can support a certain number n of *base core equivalents* (BCEs). Each BCE can be used to construct a single simple core, or they can be combined into faster cores. We will here assume that when r BCEs are combined to form a single, larger core the performance of the large core will be $\text{perf}(r)$ times as fast as a single BCE core.

We will look at a heterogeneous system, where r BCEs are used for a single large core, and the remaining $n - r$ BCEs are used as simple parallel processors. The speedup is then given by the formula

$$\text{Speedup}_{\text{asymmetric}}(f, n, r) = \frac{1}{\frac{1-f}{\text{perf}(r)} + \frac{f}{\text{perf}(r)+n-r}},$$

where f is the parallel fraction of our program, n is the total number of BCEs, and r is the number of BCEs used for the large core.

Assume that we have 1024 BCEs. Find out how many BCEs we should allocate for the single large core, and report the best possible speedup in the following scenarios:

a) $\text{perf}(r) = \sqrt{r}$, $f = 0.8$

b) $\text{perf}(r) = \sqrt{r}$, $f = 0.5$

c) $\text{perf}(r) = \sqrt[3]{r}$, $f = 0.8$

d) $\text{perf}(r) = \sqrt[3]{r}$, $f = 0.5$

Task 3 - Communication in a Cartesian grid (10 %)

You are given a periodic 1024×1024 grid of bytes and 16 processors which can communicate at 1000 bytes per second with a startup time of 50 ms. Each processor can perform full-duplex communication with at most one processor at a time. Multiple pairs of processors can communicate at the same time.

You are to perform a border exchange with a border of thickness 1. Calculate the total communication time in each of the following cases:

a) The grid is divided using strip partitioning (partitioning in one dimension only).

b) The grid is divided using block partitioning with a grid size of 8×2.

c) The grid is divided using block partitioning with a grid size of 4×4.

Show your work.

Task 4 - Image processing (70 %)

In this task, you should modify inv_filter.c form the supplied archive to implement iterative inverse filtering.

In particular, you should implement or extend the following functions: `create_types()` `initialize_guess()` `exchange_borders()` `perform_convolution()` `gather_image()`.

Futher details can be found in the recitation slides.

Your code should compile and run on clustis3. Do not run your program on the login node.