Technische Universität München
Institut für Informatik
PD Dr. Ralf-Peter Mundani
Philipp Samfass, M.Sc.

SoSe 2018

High Performance Computing - Programming Paradigms and Scalability

Exercise Sheet 7: Repetition

1 Network Topologies

The following image shows the first levels of a recursive network topology – a pyramid network. Each node has exactly four child–nodes, all child–nodes are connected through a 2–dimensional grid on the next level.

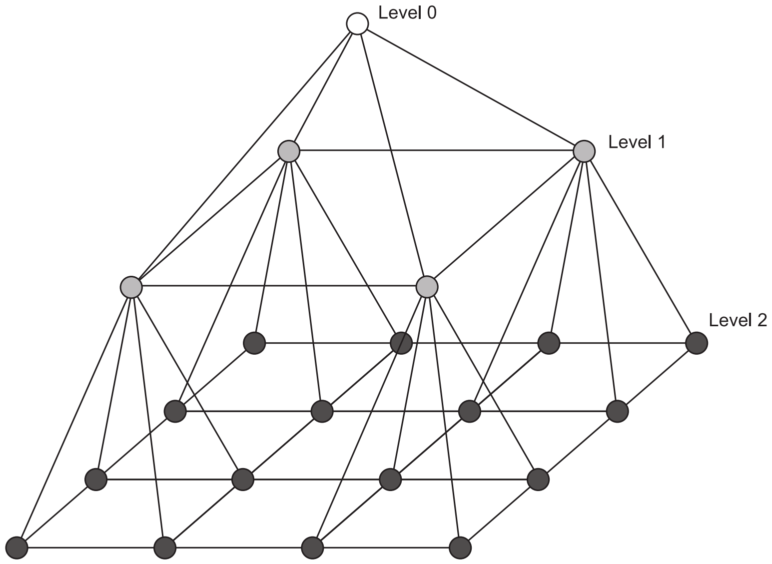

Figure 1: Example of a pyramid-network of height $H = 2$ and a total of $N = 21$.

a) Give a general formula, depending only on the height H, to compute the total number of nodes N in the Network. Sums should – if possible – be resolved.

b) Compute

   i) cost (meaning the number of edges)

   ii) diameter

   iii) bisection width

   of a general pyramid–network depending on N and H only.

2 Semaphores

A railway track between two cities contains a bridge over a canyon that can be accessed by a single train at any point in time only. Hence, this bridge is an exclusively usable resource and needs to be implemented as a critical section.

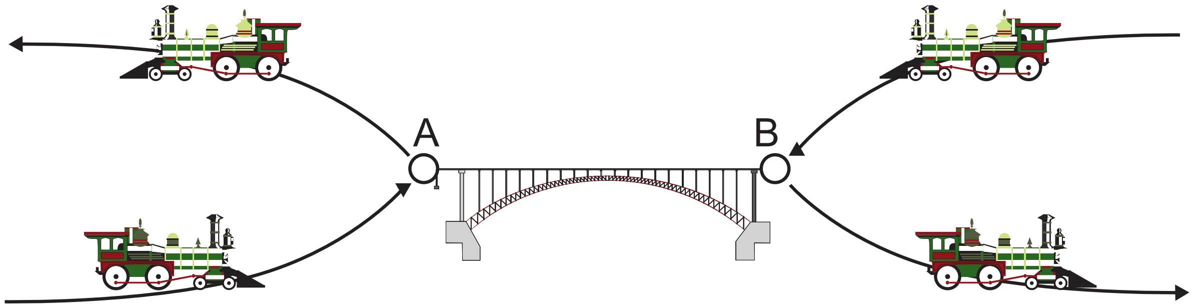

Figure 2: Visualisation of *L*- and *R-trains* and the exclusively usable bridge.

There exist two types of trains: *R-trains* that always drive from the left-hand to the right-hand side and *L-trains* that always drive from the right-hand to the left-hand side. Give a correct synchronisation (pseudo code) using as many semaphores as necessary, thus, no two trains can access the bridge at the same time **and** *R-trains* and *L-trains* access the bridge alternately, i.e. after an *R-train* always follows an *L-train* and vice versa. Also give a correct initialisation of all your semaphores!

3 OpenMP

On some quadratic meadow the grass needs to be cut. Thus, several people – each equipped with a mowing machine – have to organise themselves in order to do this work in parallel. Consider the meadow as a 2-dimensional array A of size $N \times N$ and some function `mow()` to be executed for each element α_{ij} of A. Write a parallel program using OpenMP and think about sufficient synchronisation!

4 Data Distribution and Efficiency

Given is some iterative algorithm that processes 3-dimensional data stored in a 3-dimensional matrix of side length N. Processing one element α_{ijk} of A takes $t_P = 2\text{ns}$ time. Exchanging one element between two processes (one direction!) takes $t_{ex} = 0.5t_P$ time. After one complete processing step all processes have to exchange data at their borders (treat all processes the same and do not distinguish between different amounts of neighbouring processes!).

Choose a row-wise, column-wise, or block decomposition of A and sketch the corresponding data distribution. For which sizes N of A (depending on the number of processes p only) could a parallel efficiency of at least 60% according to your decomposition be achieved?