**Introduction:** Efficiently managing threads is a well-known problem, but still, it is one of the bottlenecks to speed up the performance. Both in control-driven as well as in a dataflow driven programming model, thread management is complex. The complexity has been extended further by the underlying memory models (such as shared memory for multi-node (especially the old grid/peer-to-peer computing), distributed shared memory for many-/multi-cores). No matter what the application domain is, to execute the job the application will be abstracted as the thread in both the von Neumann or non-von Neumann architecture. In the state of the art, there is a huge number of solutions but either they do not scale well for a large number of cores or the solution is very specific to the underlying hardware.

To counter such issues in the thesis, we are aiming to follow the hardware-software co-design approach. Here the scheduling is further decomposed into mapping and allocating. The mapping is the novelty of the work while relying on the Kernel's CPU scheduler to allocate the threads to the cores. The primary goal of this work is to develop a scheduler that distributes threads to reduce data movement and access (read and/or write) in a multi-core (1000+ cores) architectural environment. Ideally the "same solution” should work both in many cores as well as in the multi-node environment.

**Required Theory:** Good knowledge of application threads, Linux scheduler, processor architecture.  
**Required skills:** C/C++/JAVA, JSON, Verilog/VHDL, scripting (bash), python (optional).  
**Required Tools:** Student has the option to select either simulator such as Gem5 (C/C++/JAVA) or FPGA (Verilog/VHDL) based simulation.

**Duration:** 6 months  
**Workload:** 70% coding, 30% research (approx. values)  
**Expected outcome:** Publication/ Master thesis

**Mode:** Remotely via Zoom  
**Supervision:** 1 hr/week for 6 months

**Contact:** If you are interested then send your CV (max 2 pages) and a GitHub link of your code repos. Send email to Somnath Mazumdar (sma.digi@cbs.dk).