Tertimuss

Authors: Gaddiel Desirena López, Lorena Rubio Anguiano, Antonio Ramírez Treviño, José Luis Briz
Creation date: October 22, 2018
Web page: Thermal-aware Energy-efficient Real Time Multiprocessor Scheduling Simulator
Description: Tertimuss (Thermal-aware Energy-efficient Real Time Multiprocessor Scheduling Simulator) is a simulation environment for designing and testing Real Time multiprocessor schedulers subject to thermal constraints. It consists of four modules. The first module allows defining the system (processors, tasks) and their parameters. The second module automatically builds a TCPN model, generating the state and thermal equations. The third module is for selecting, modifying or adding and parameterizing the scheduling algorithm. The fourth module allows the user to perform simulations and collect, process and plot results. The parameters of the tasks can be defined either manually or automatically (by means of the integrated UUnifast algorithm).

Pipelined architecture for sparse DNNs

Authors: Adrián Alcolea, Javier Olivito, Javier Resano
Creation date: February 04, 2019
Repository: Pipelined architecture for sparse DNNs
Description: VHDL code of an accelerator for Convolutional Neural Networks that takes advantage of sparsity (large number of zeros) to work with compressed filters and avoid operations with a zero in at least one of the operands.

bvSFM (bit-vector sampled FM-index): a tool for sequence alignment

Authors: José Manuel Herruzo, Jesús Alastruey Benedé, Pablo Ibáñez Marín
Creation date: May 20, 2019
Repository: bvSFM: a tool for sequence alignment
Description: bvSFM (bit-vector sampled FM-index) is a tool for sequence alignment. Specifically, it implements an exact search algorithm that counts the number of matches of arbitrary length reads on a reference genome. bvSFM indexes a genome with an FM Index (based on the Burrows-Wheeler Transform or BWT). FM Index is a compact data structure suitable for fast matches of short reads to large reference genomes. For the human genome, its memory footprint is typically around 3.2 gigabytes of RAM. bvSFM uses an optimized FM-index data structure layout and codification that packs all relevant data needed in a query step within a single cache block, minimizing the memory bandwidth demand. bvSFM achieves best results when executed on multicore systems integrating high bandwidth memory, for instance an Intel Xeon Phi processor KNLa (codenamed Knights Landing, or KNL).