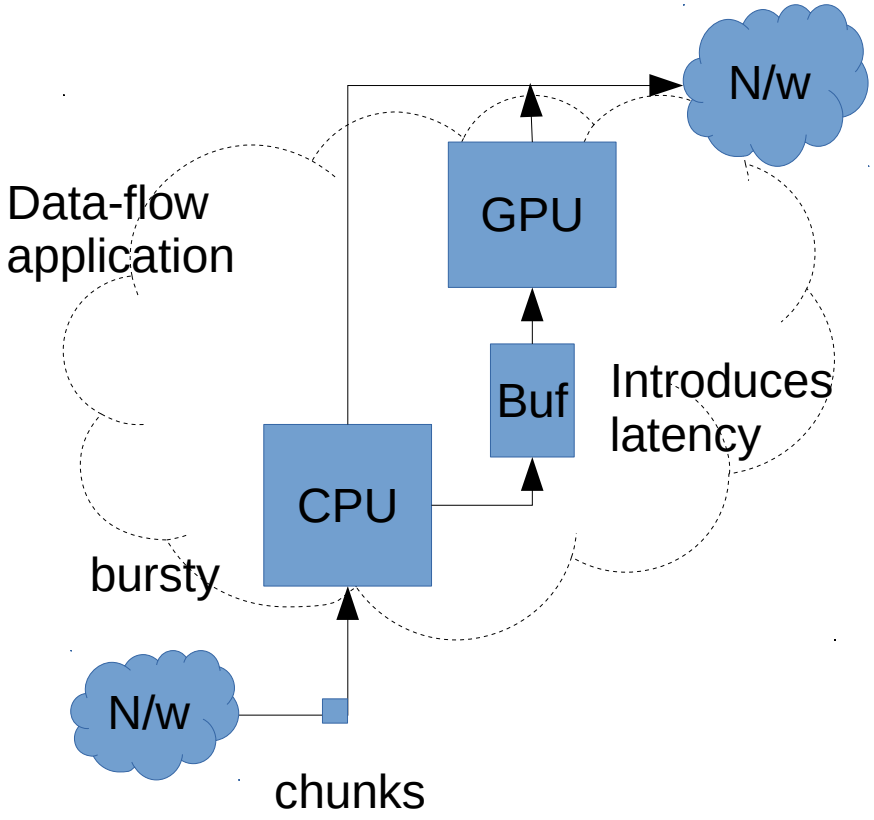
Åbo Akademi University

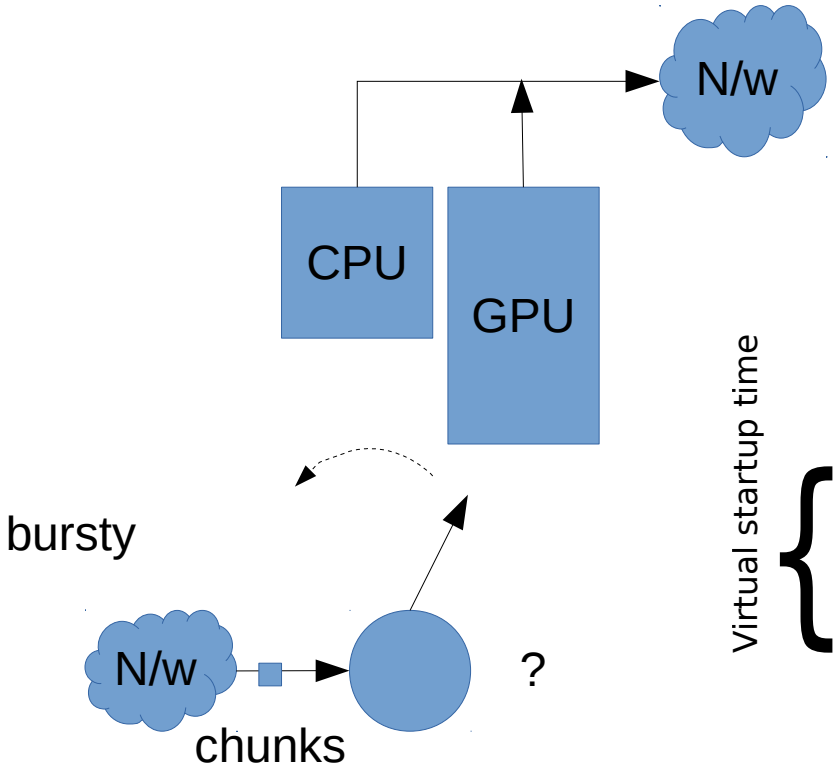
Determining a Device Crossover Point in CPU/GPU Systems for Streaming Applications

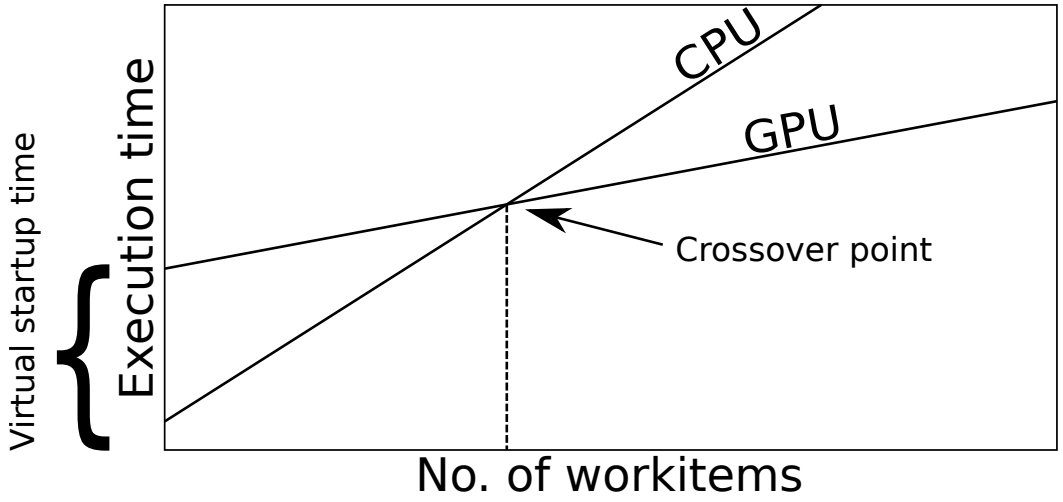
Sudeep Kanur, Wictor Lund, Leonidas Tsipoulos, Johan Lilius
Embedded Systems Lab, Åbo Akademi University
{skanur, wlund, ltsiopou, jolilus}@abo.fi

Context: Accelerating streaming data-flow application

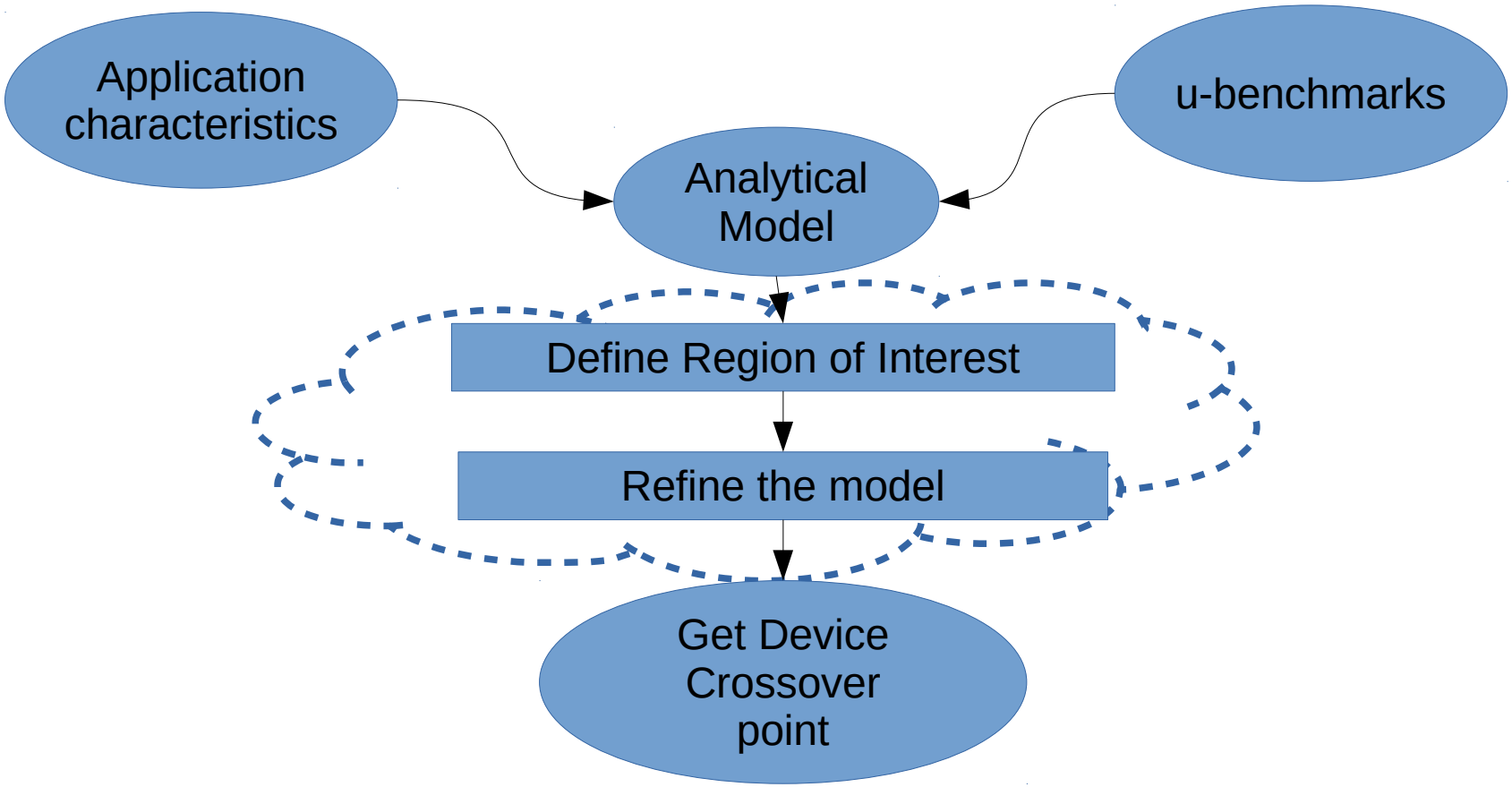


Efficient processing of streaming data





To find Device Crossover Point



Åbo Akademi University

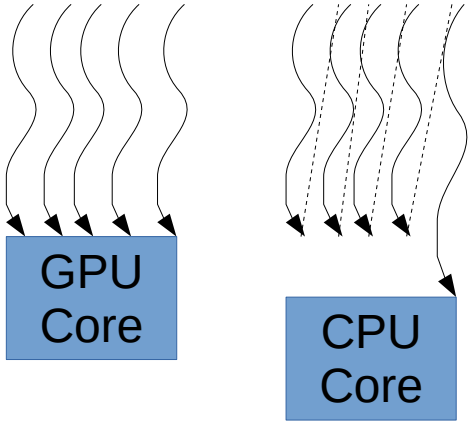
Analytical Model

- Refinement of model proposed by Hong and Kim [1]
  - MWP: Number of warps a core can concurrently access during one memory access request
  - CWP: Number of warps a core can compute during memory request of warp is being serviced
- Two major modification in the model and then applied to both CPUs and GPUs

[1]Sunpyo Hong and Hyesoon Kim. An analytical model for a GPU architecture with memory-level and thread-level parallelism awaremess. In ACM SIGARCH *Computer Architecture News,* volume 37, pages 152-163. ACM, 2009

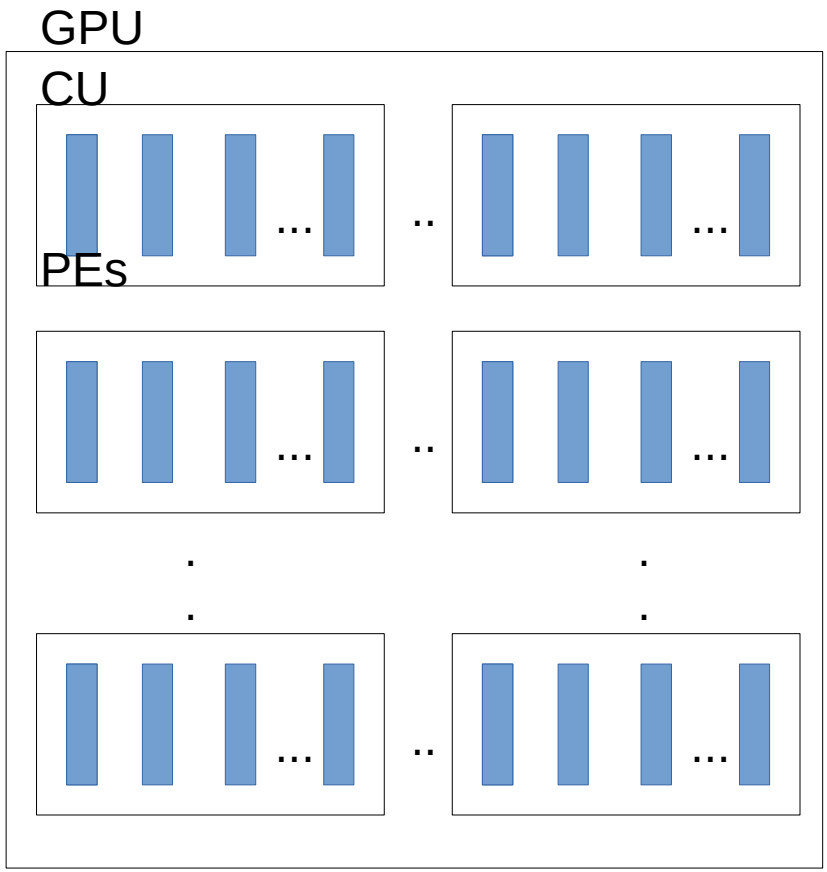
Åbo Akademi University

Modifications



- Differences in the way threads are executed in devices
- GPUs group threads and run them in parallel
- Active threads on GPU increases with a step of parallel elements
- Active threads on CPU increases linearly in CPU
- Also, a typical CPU core has a max limit of 1-2 based on SMT capabilities
- Memory access patterns simplified for AMD devices

Refinement



- Model doesn't take hardware optimizations into account → Overestimation of execution time
- We define a region of interest for a chunk size: nCU x N_{eff} x (group size)
- Find correction factor at this point

Application Characteristics

- Validation on kernels of H.264/MPEG-4 AVC and Motion-JPEG
- Applications written in RVC-CAL dataflow language.
- OpenCL code generation through ORCC compiler
  - SDF actor that are stateless are translated to OpenCL code [2]
- OpenCL code generated uses global memory only and has no synchronization primitives

Wictor Lund, Sudeep Kanur, Johan Ersfolk, Leonidas Tsipoulos, Johan Lilius, Joakim Haldin and Ulf Falk. Execution of dataflow process networks on opencl platforms. In *Parallel, Distributed and Network-Based Processing (PDP), 2015 23rd Euromicro International Conference on,* pages 618-625, March 2015

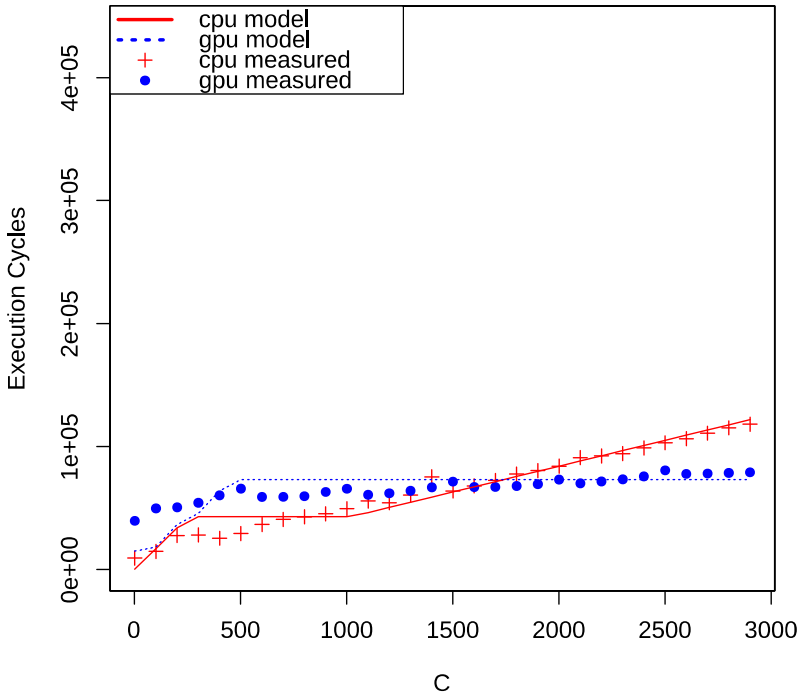
Åbo Akademi University

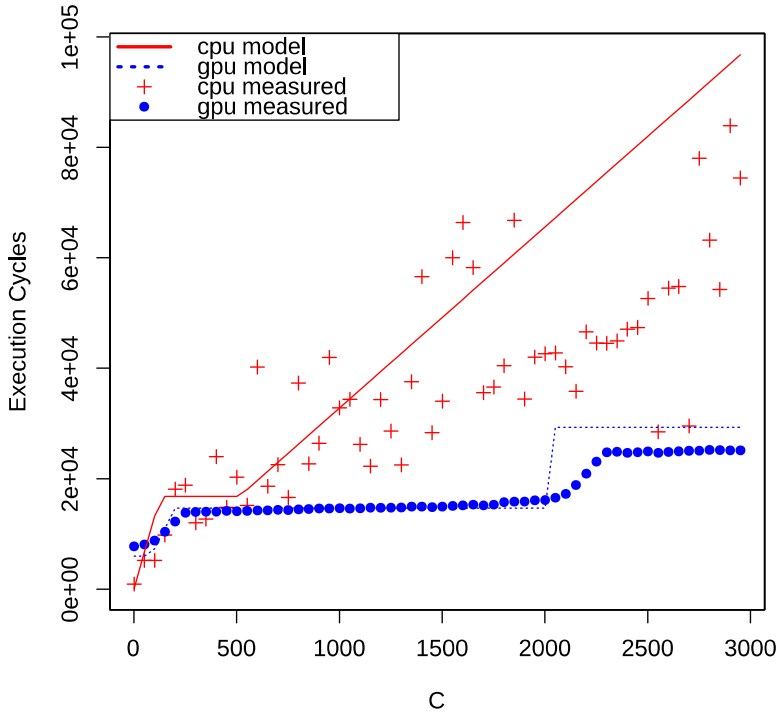
Experiments

Parameter	CPU	GPU	Units
WarpSize	128	64	-
nCU	4	8	-
Freq	2.4	1.2	GHz
Mem rate	3.4	29	GBps
DRAM lat	5	70	cycles
Inst lat	72	84	cycles

- CPU - AMD A10-7870K Accelerated Processing Unit
- GPU – R7 integrated GPU
- OpenCL 1.2
- Kernels are run for varying chunk sizes and execution times are noted.

Results





Kernel Name	Model	Measured
2D IDCT	1600	1600
Zigzag	100	200-250

Åbo Akademi University

Discussion

- Large deviation from predicted execution time from the model to the measured
- Only execution times are considered, no startup or transfer times
- Despite shortcomings, prediction of device crossover point is fairly accurate
- Decision is simple once the model is tuned initially around the region of interest

Åbo Akademi University

Future Work

- Better model to predict CPU execution time
  - Analyze CFG of the kernel
  - Analysis for arbitrary OpenCL kernel
- Better way to obtain region of interest
- Validation against more devices/kernel pair.

Åbo Akademi University

Thank you

Sudeep Kanur, Wictor Lund, Leonidas Tsipoulos, Johan Lilius
Embedded Systems Lab, Åbo Akademi University
{skanur, wlund, ltsiopou, jolilus}@abo.fi