EURO-PAR
CONFERENCE 2022

A hybrid piece-wise slowdown model for concurrent kernel execution on GPU

B. López-Albelda, F. M. Castro, J. M. González-Linares and N. Guil

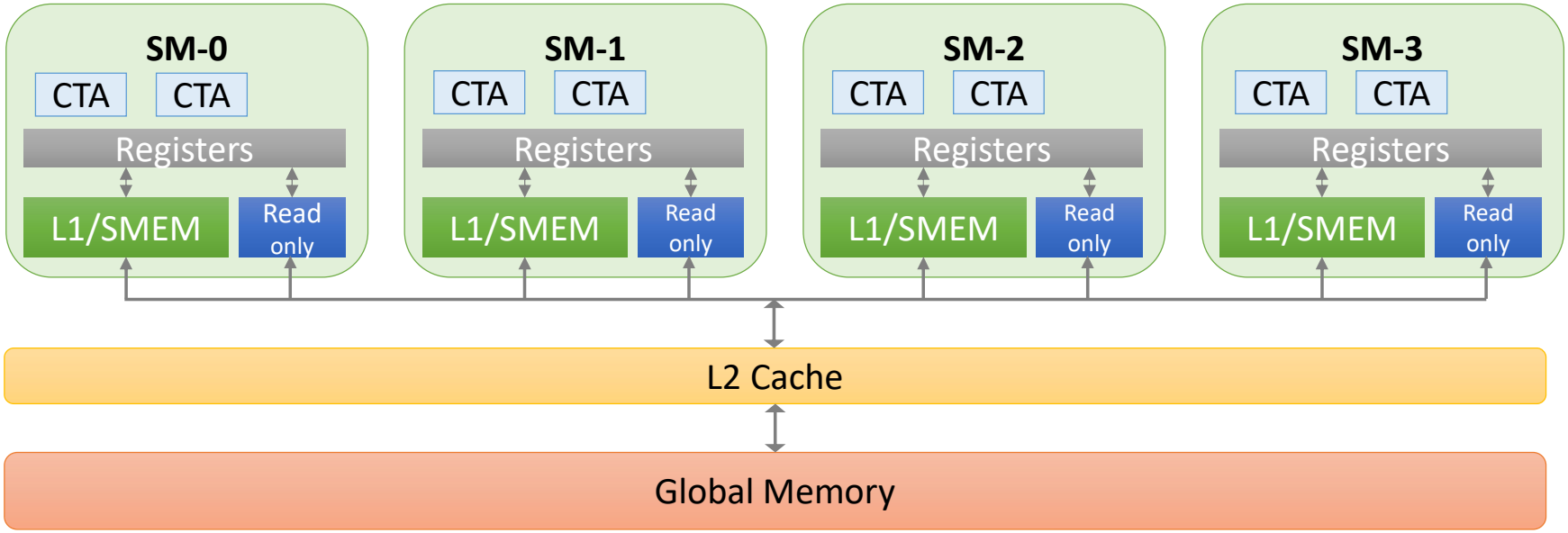
UNIVERSIDAD
DE MÁLAGA

Contents

Motivation

EURO-PAR
CONFERENCE 2022

Kernel execution on a GPU may not scale well due to architecture constraints and saturation of computational resources like pipeline stalls, L1-cache trashing, global memory bandwidth, etc.

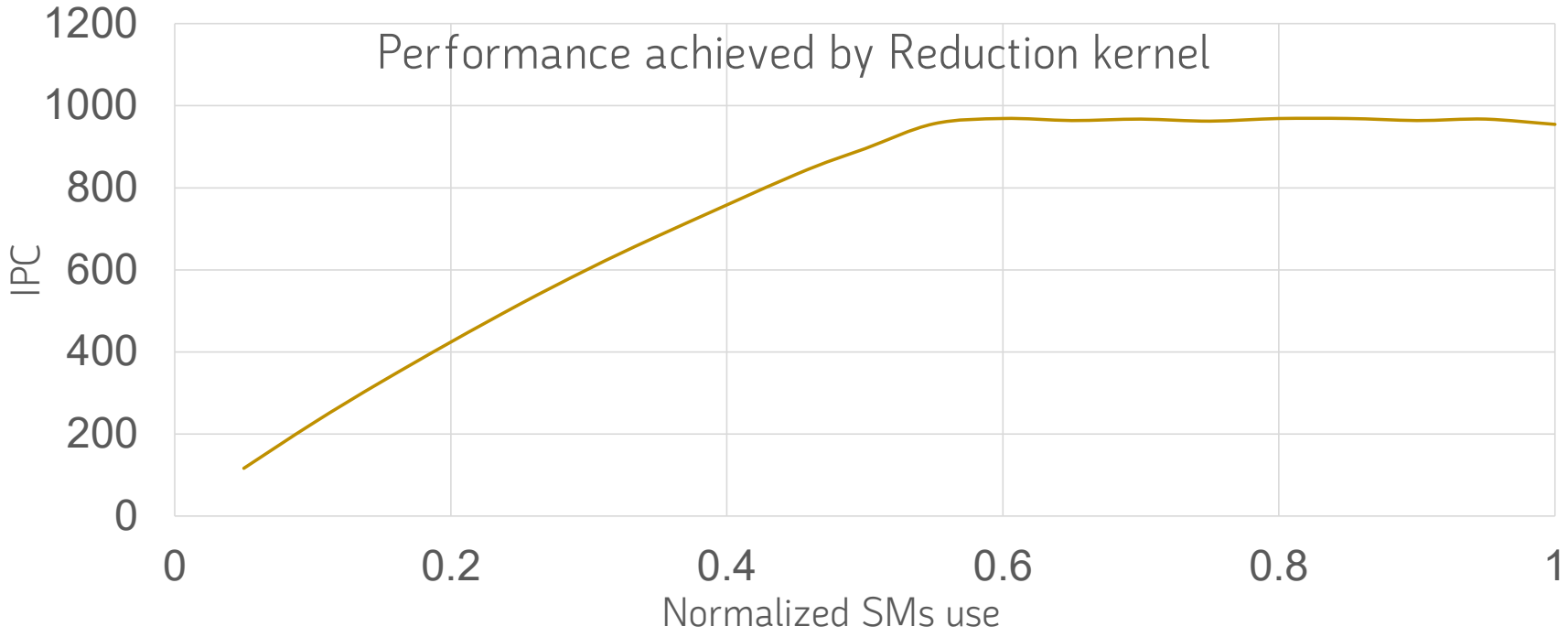


SM: Streaming Multiprocessor
CTA: Cooperative Threads Array

Waste of computational power

EURO-PAR CONFERENCE 2022

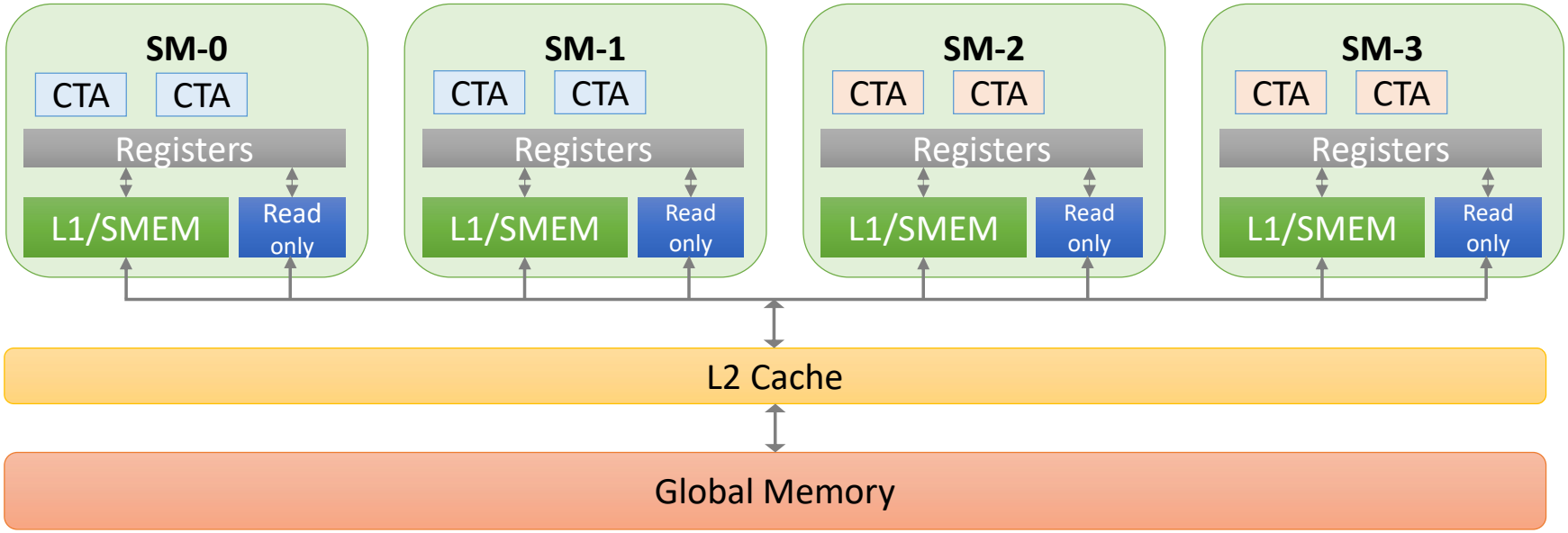
Kernels like Reduction do not scale well



GPU resources allocation to kernels: SMT

Spatial Multitask (**SMT**): CTAs are assigned to a subset of SMs

(CTAs of one kernel in light blue, CTAs of other kernels in light orange)

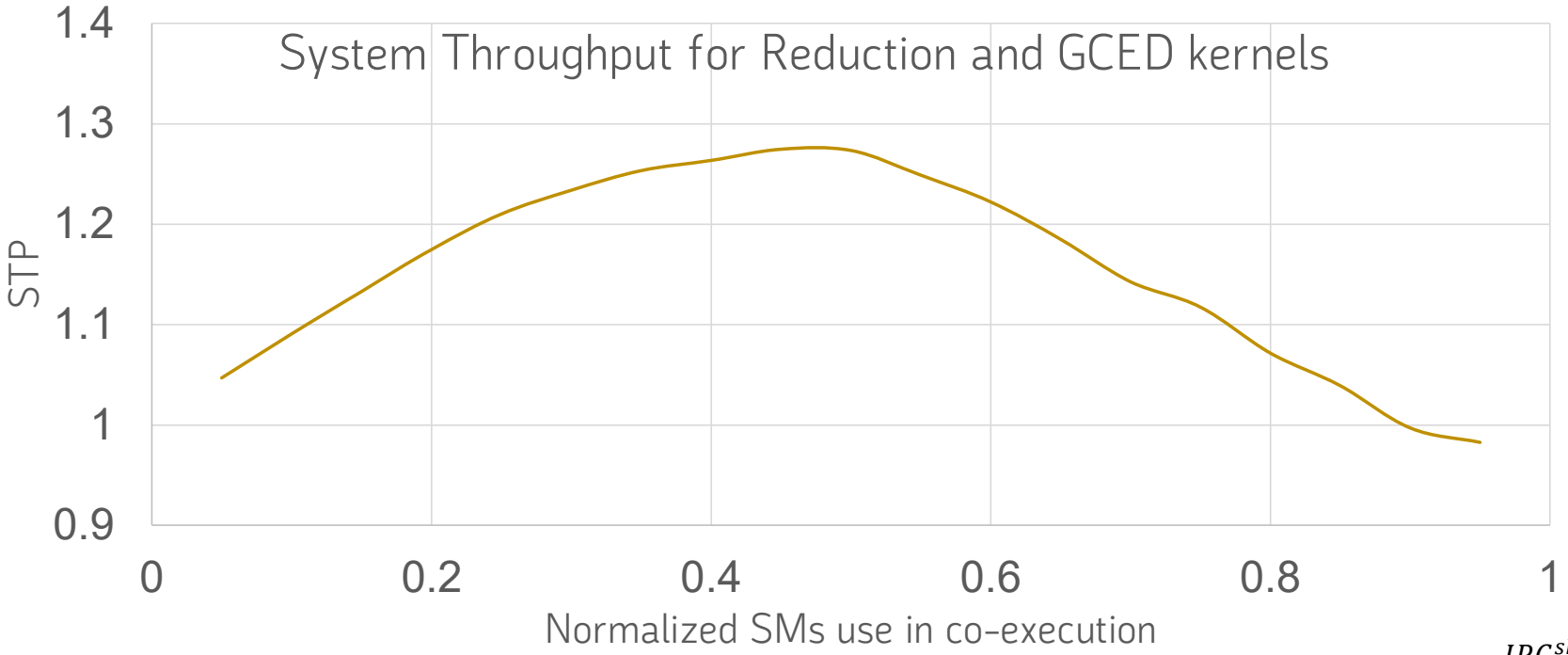


SM: Streaming Multiprocessor

CTA: Cooperative Threads Array

Concurrent Kernel Execution (CKE)

A kernel can be co-executed with another kernel with complimentary requirements to improve performance



$$STP = \frac{IPC_{RED}^{shared}}{IPC_{RED}^{alone}} + \frac{IPC_{GCEDD}^{shared}}{IPC_{GCEDD}^{alone}}$$

Slowdown Model

Best results obtained by co-executing compute bound (CB) kernels together with memory bound (MB) kernels

A slowdown model can be used to obtain the best mapping:

- Predict Normalized Progress, NP, for each SM allocation

$$NP = \frac{IPC^{shared}}{IPC^{alone}}$$

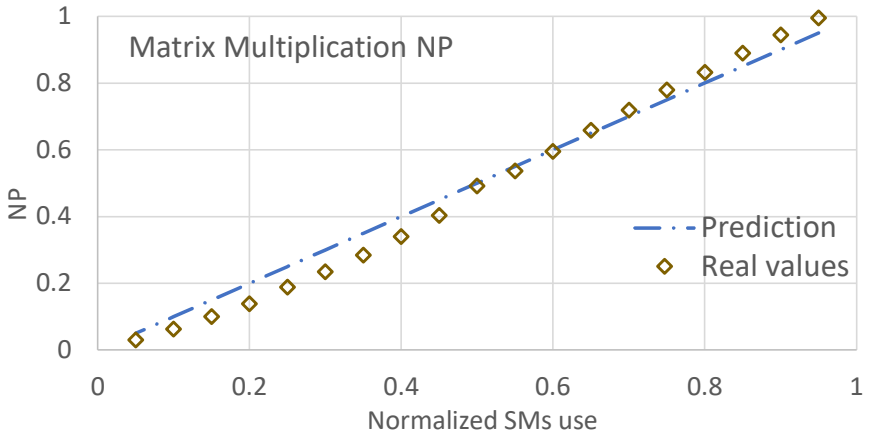
- Use some criteria like fairness, STP or ANTT to choose the best mapping

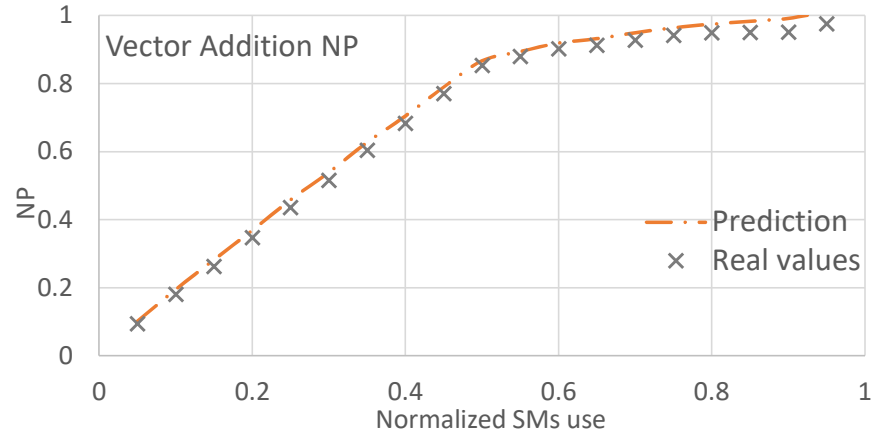
$$fairness = \min_{i,j} \frac{NP_i}{NP_j}$$

HSM: A Hybrid Slowdown Model for Multitasking GPUs, Zhao, Jahre and Eeckhout, ASPLOS'20

- NP of compute-bound kernels is predicted as $NP = \frac{R_{cb}^{shared}}{R_{cb}^{alone}}$ where R_{cb}^{shared} is the total number of SMs allocated and R_{cb}^{alone} is the total number of SMs in the GPU.

- NP of memory-bound kernels is predicted using the effective bandwidth utilization, which can be obtained from the Row Buffer Hit rate (RBH)

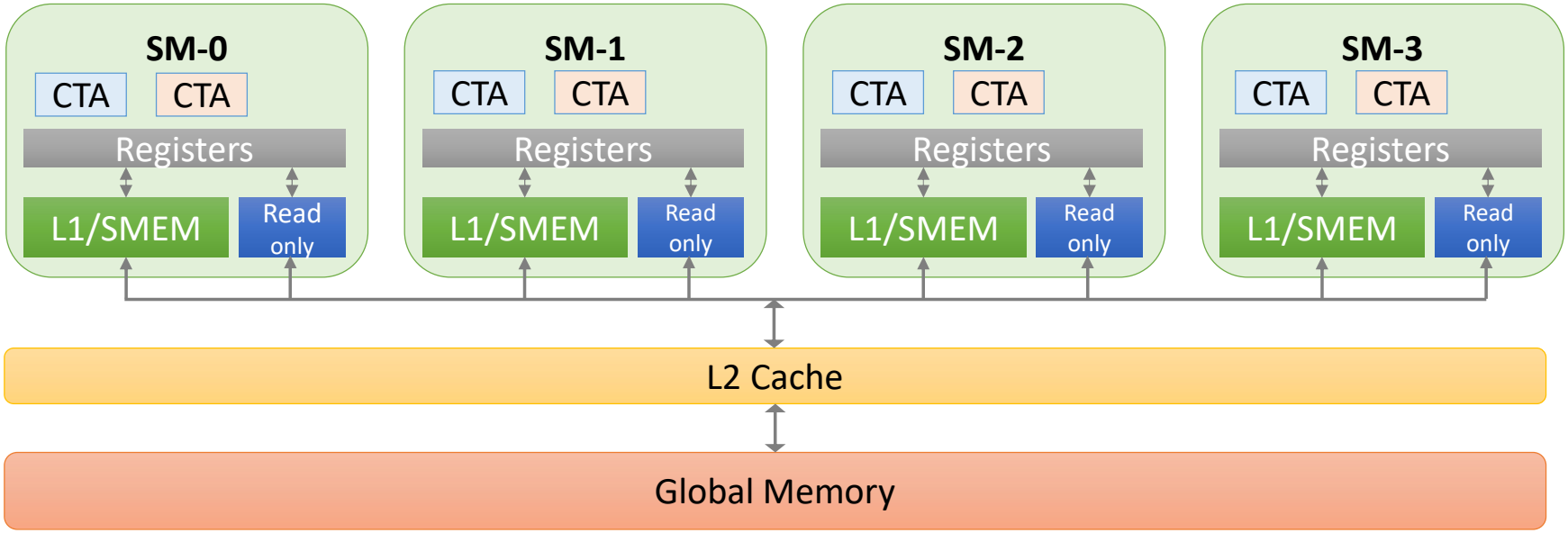




GPU resources allocation to kernels: SMK

Simultaneous Multikernel (**SMK**): CTAs are assigned to all SMs

(CTAs of one kernel in light blue, CTAs of other kernels in light orange)

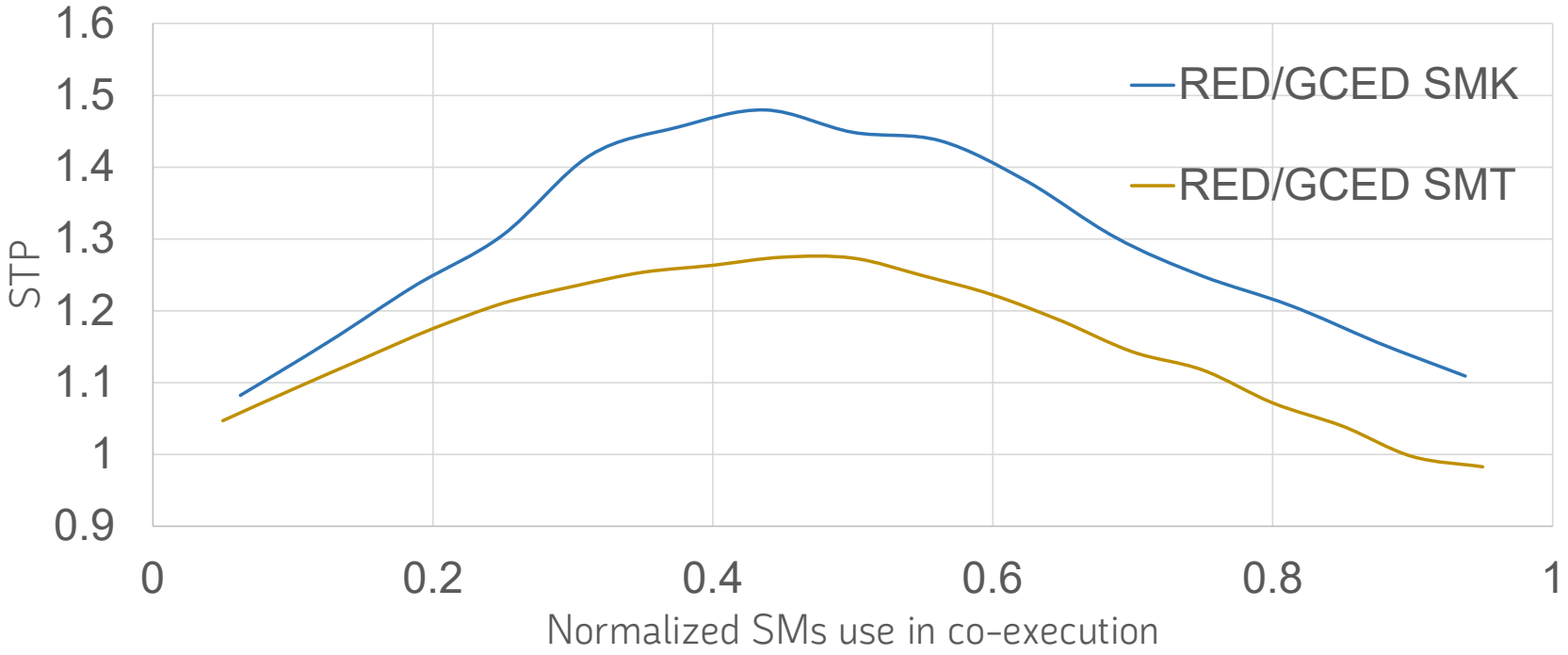


SM: Streaming Multiprocessor

CTA: Cooperative Threads Array

CKE using SMK

Performance under SMK allocation can improve significatively



Objectives

EURO-PAR
CONFERENCE 2022

Compare	Build	Implement
Compare SMK with respect to SMT in terms of performance using • Average Normalized Turn-around Time (ANTT) • System Throughput (STP)	Build a slowdown model for SMK	Implement a fairness-based CTA allocation policy

Testbed

GPGPU-sim v4.0 modified to support SMT and SMK

Simulated system: Volta Titan V, with 80 SMs and HBM memory with 3 stacks and 24 channels

Benchmarks: 14 kernels from CUDA SDK, Rodinia, Parboil and Chai

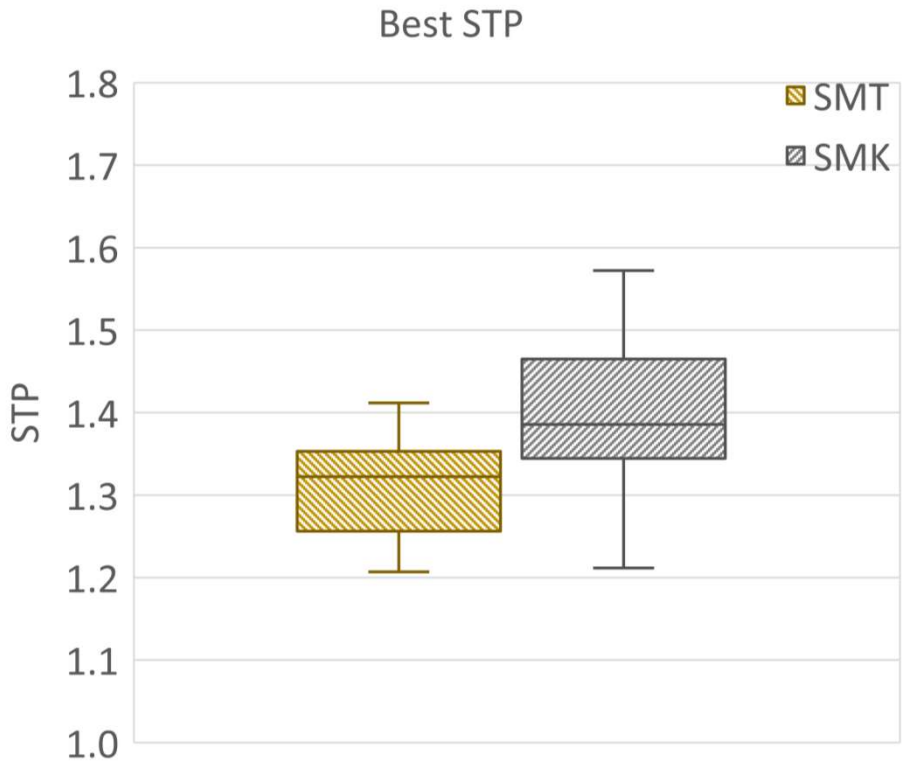
- 8 MB and 6 CB

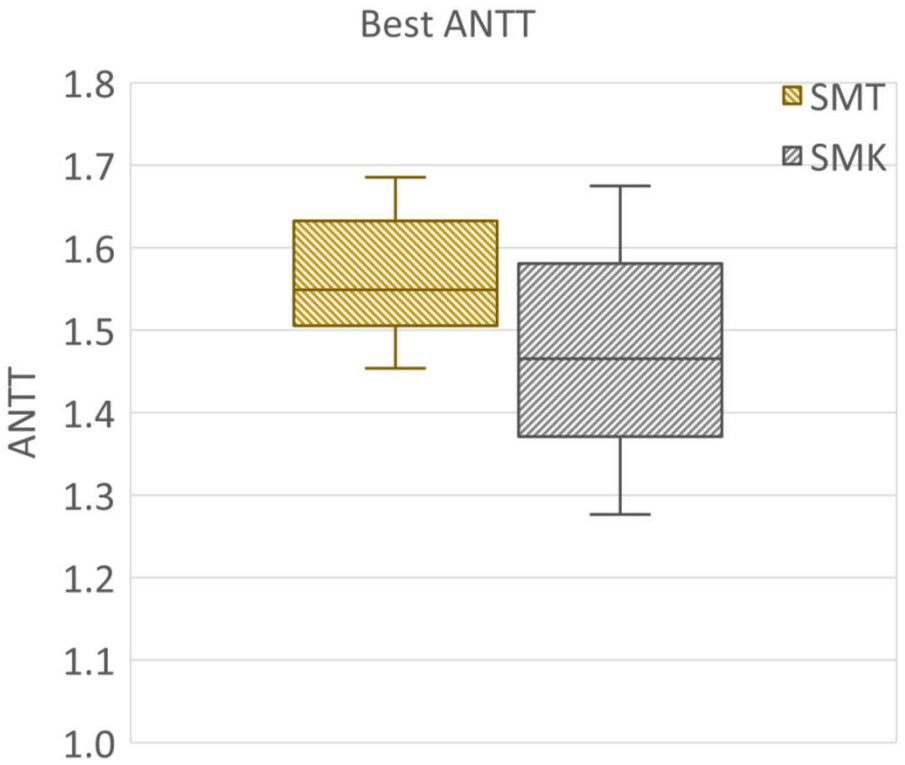
Concurrent kernel execution test: both kernels are launched at the same time, and stopped when one of them has no new CTAs left to allocate

Performance metrics:

$$NP = \frac{IPC^{shared}}{IPC^{alone}} \quad STP = NP_i + NP_j \quad ANTT = \frac{1}{2}\left(\frac{1}{NP_i} + \frac{1}{NP_j}\right) \quad fairness = \min_{i,j} \frac{NP_i}{NP_j}$$

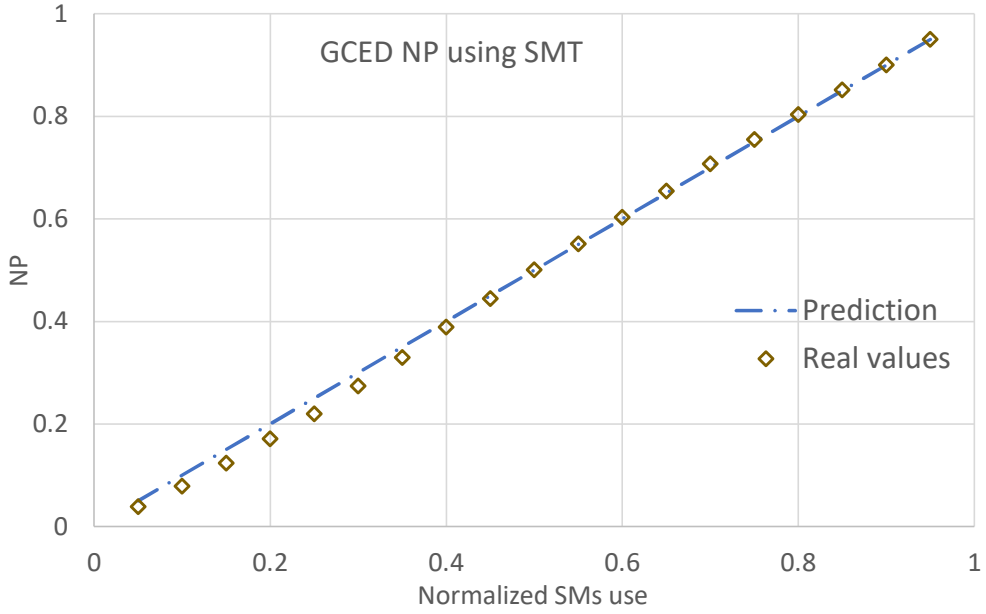
SMK vs SMT

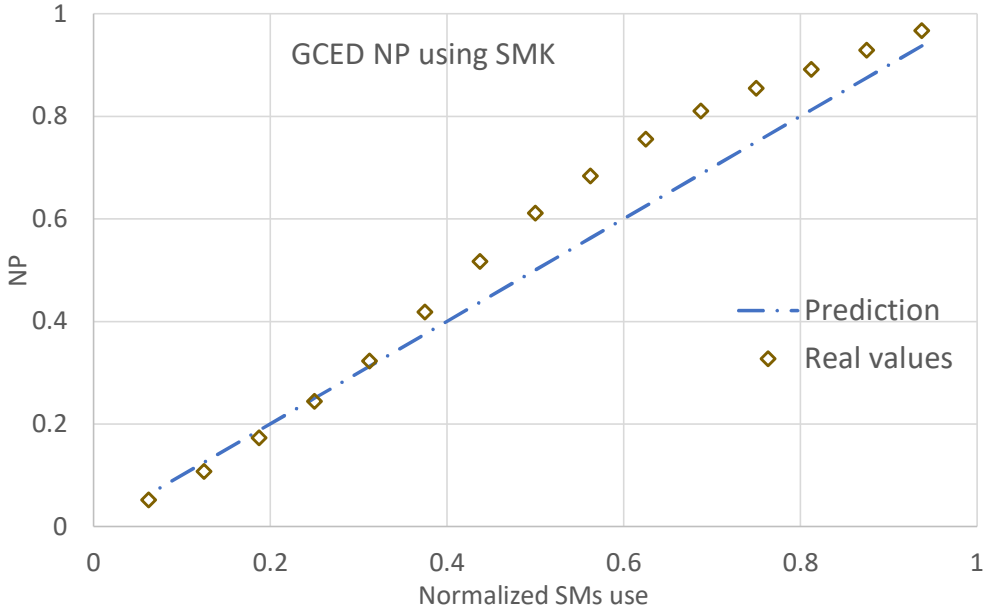

Best STP
SMT
SMK
STP
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0


Best ANTT
SMT
SMK
ANTT
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0

Using HSM with SMK

NP prediction for CB kernels under SMK allocation is not very good

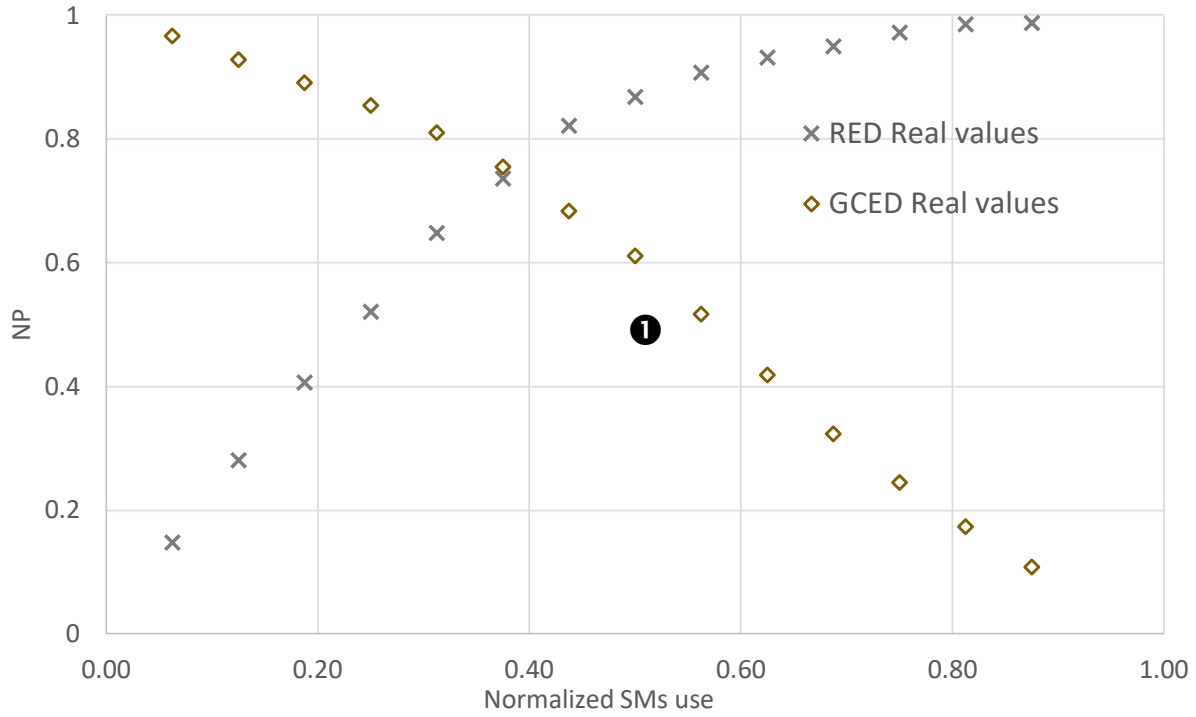




HSM slowdown prediction with SMK

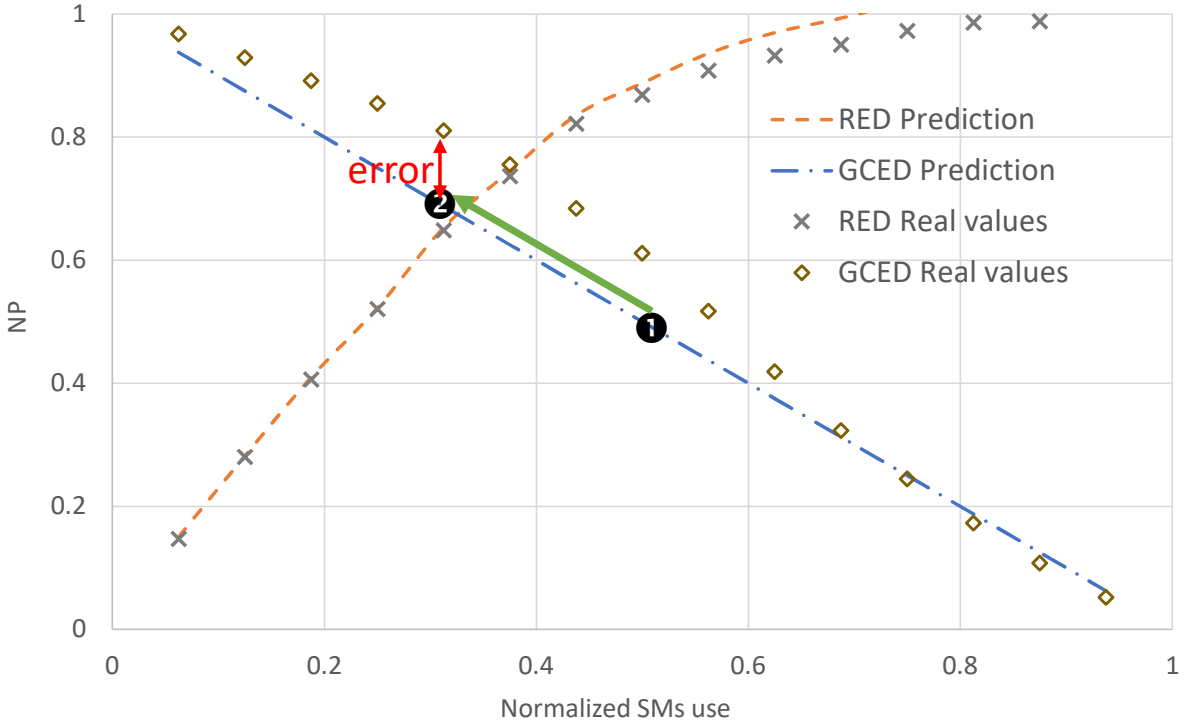
EURO-PAR CONFERENCE 2022

1. Allocate half SMs use to each kernel



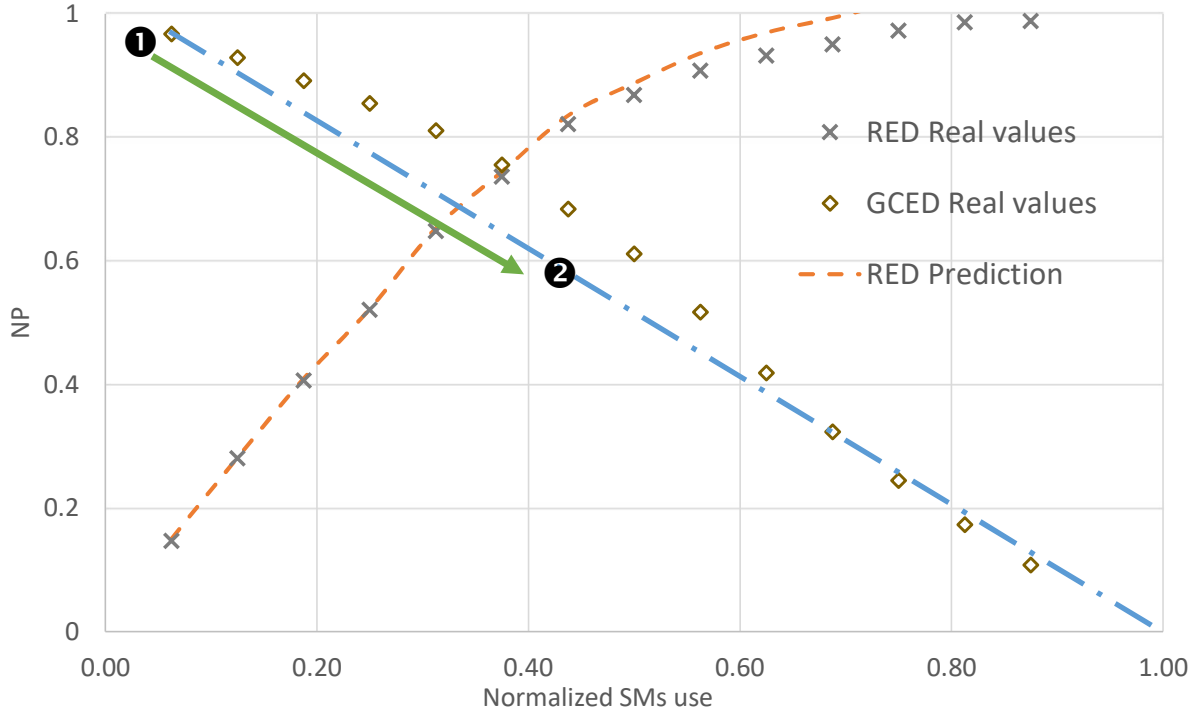
HSM slowdown prediction with SMK

1. Allocate half SMs use to each kernel. Predict NP of both kernels
2. Give (or take) SMs use to the CB kernel



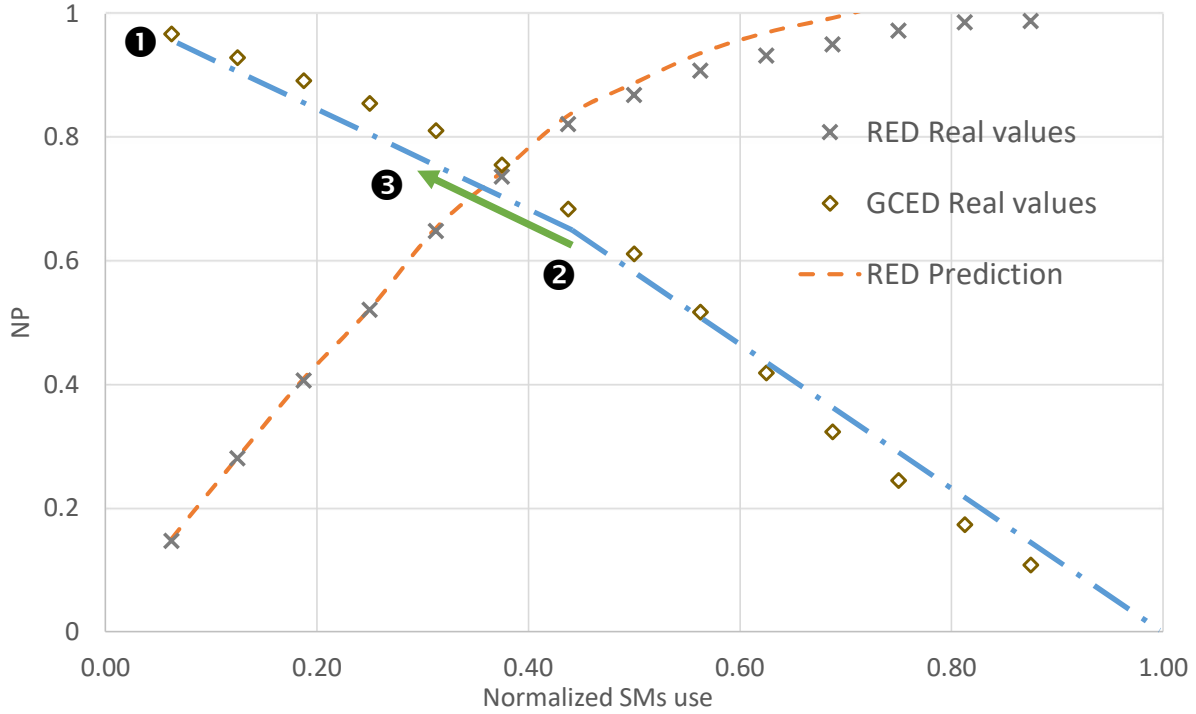
A hybrid piece-wise slowdown model

1. Start at one extrema: detect whether kernel is CB or MB
2. Give (or take) SMs use to the CB kernel



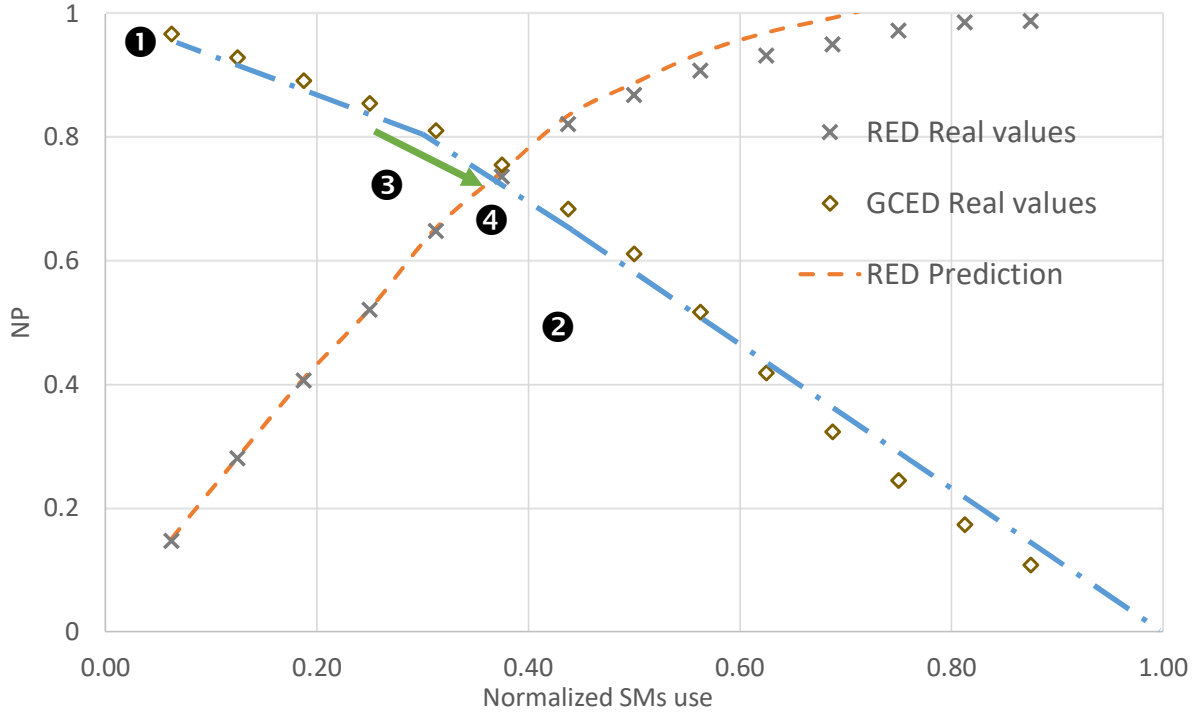
A hybrid piece-wise slowdown model

2. Give (or take) SMs use to the CB kernel. Adjust NP with the real value
3. Give (or take) SMs use to the CB kernel.

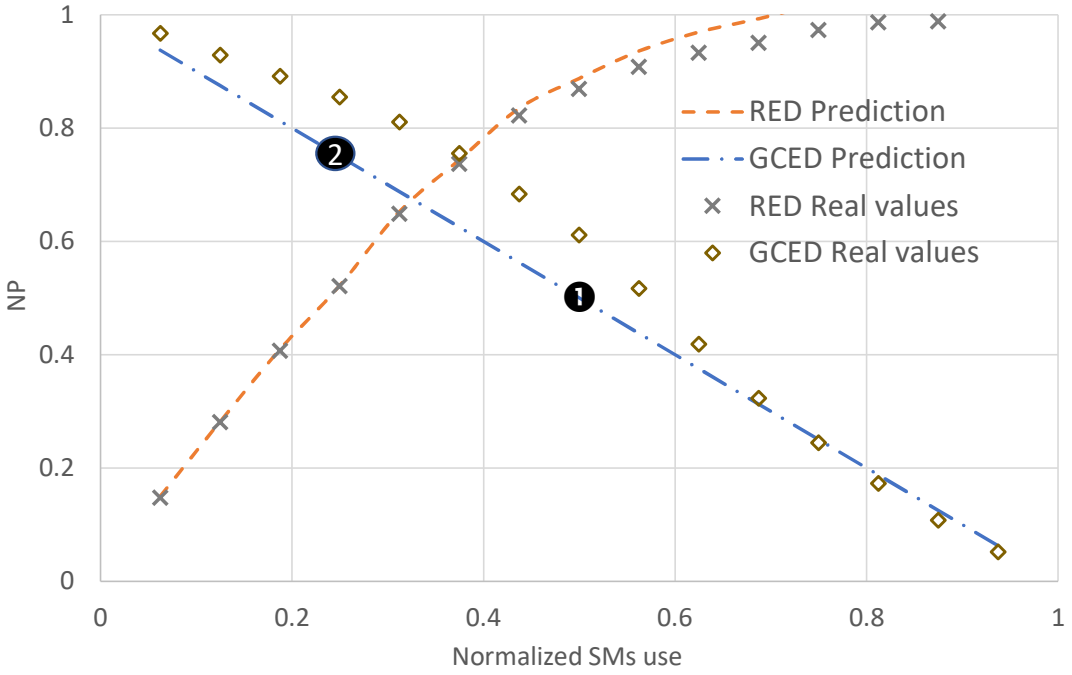


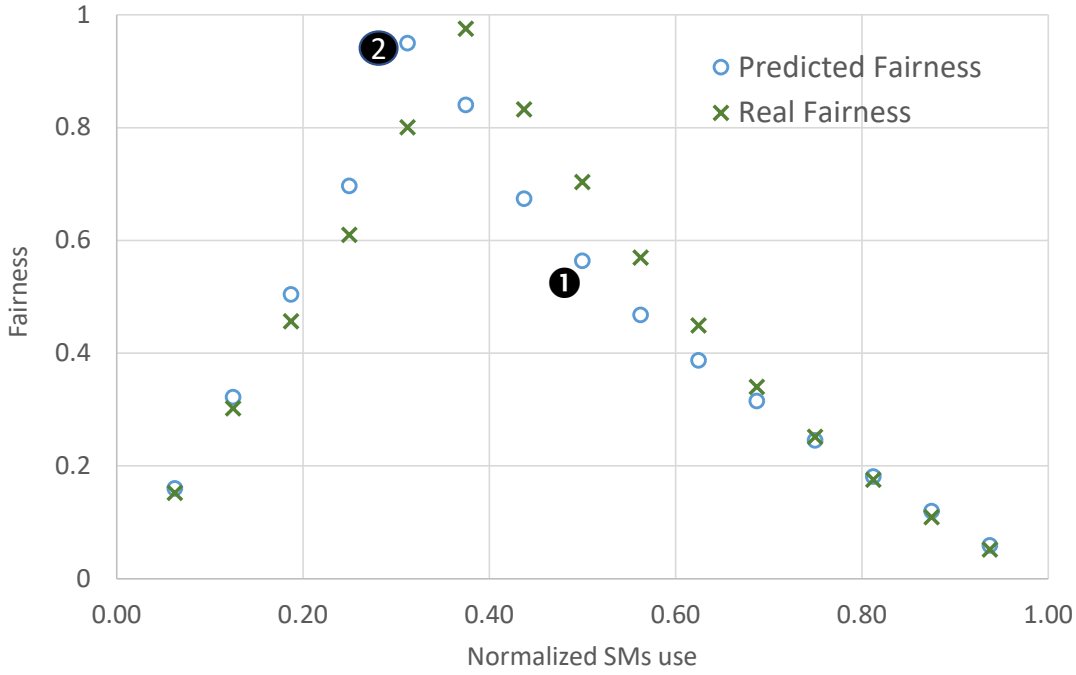
A hybrid piece-wise slowdown model

3. Give (or take) SMs use to the CB kernel. Adjust NP with the real value
4. Give (or take) SMs use to the CB kernel…

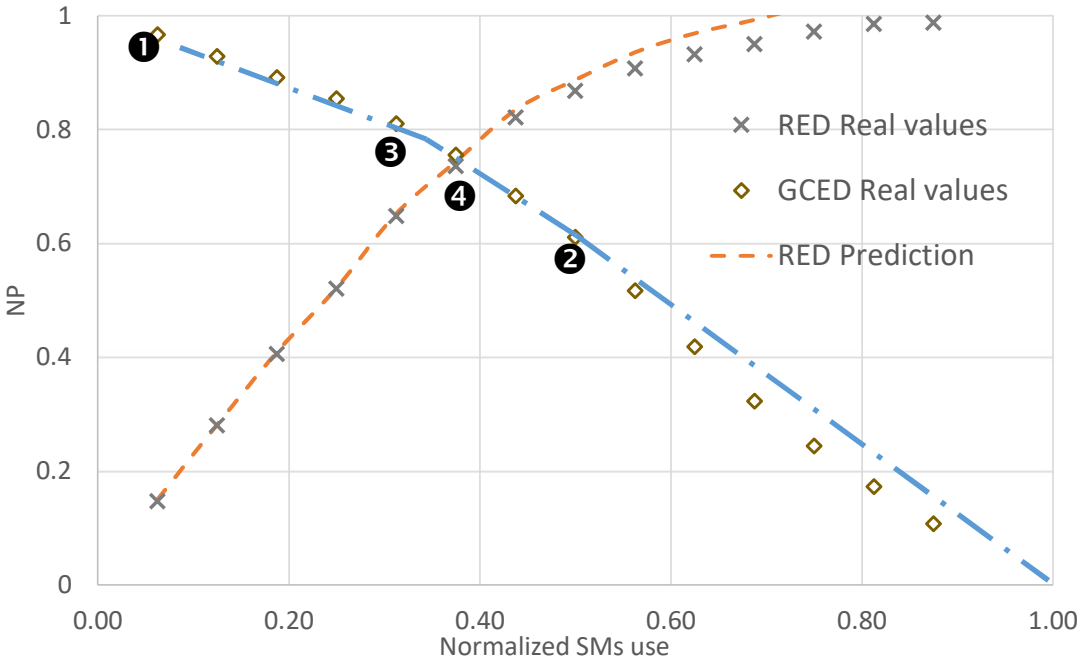


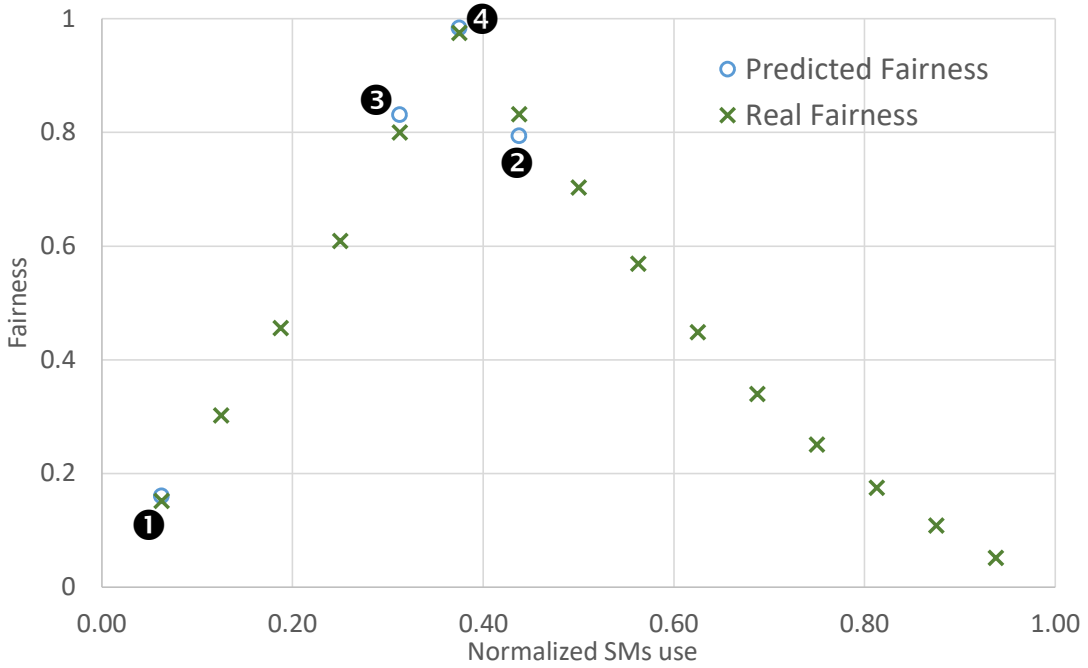
HSM-Fair


NP
Normalized SMs use
RED Prediction
GCED Prediction
RED Real values
GCED Real values
1
2

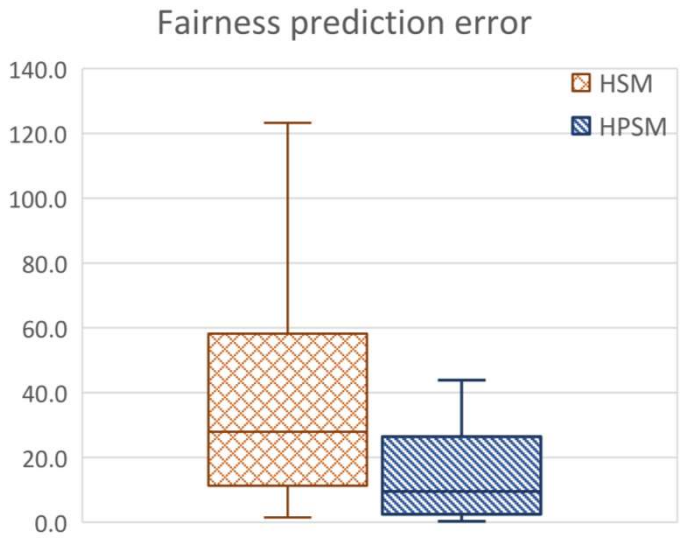

Fairness
Normalized SMs use
Predicted Fairness
Real Fairness
1
2

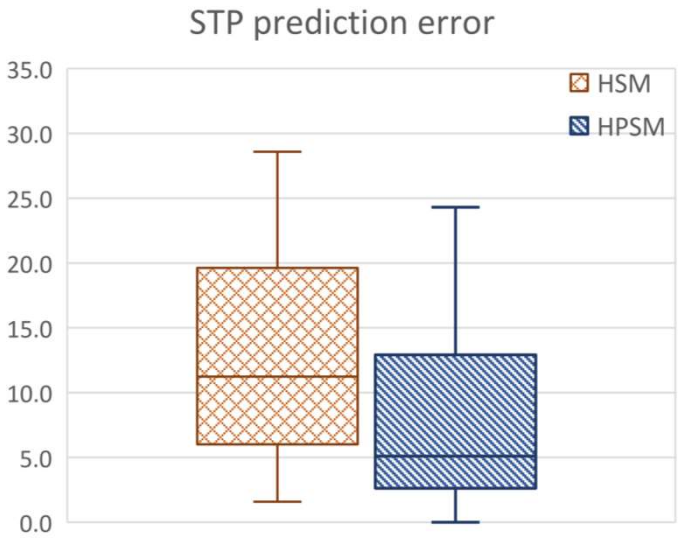
HPSM-Fair

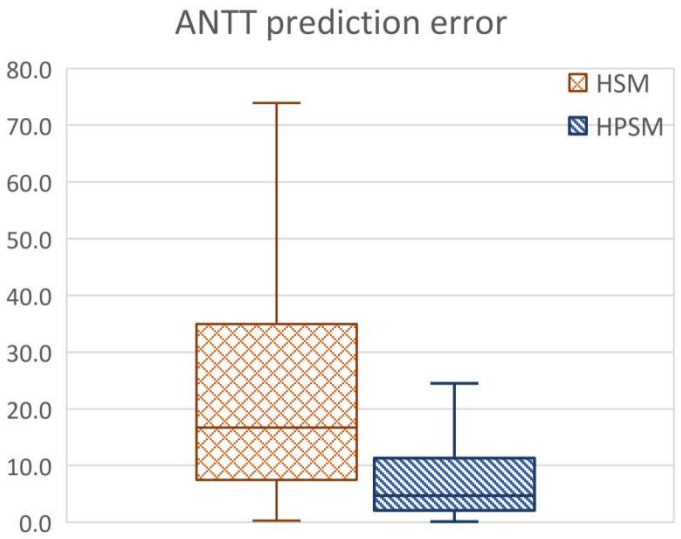

1
0.8
0.6
0.4
0.2
0
NP
0.00
0.20
0.40
0.60
0.80
1.00
Normalized SMs use
RED Real values
GCED Real values
RED Prediction
1
2
3
4


1
0.8
0.6
0.4
0.2
0
Fairness
0.00
0.20
0.40
0.60
0.80
1.00
Normalized SMs use
Predicted Fairness
Real Fairness
1
2
3
4

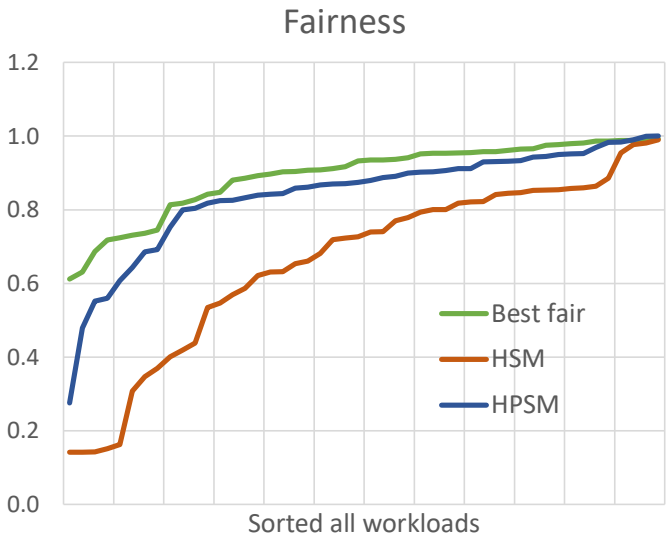
Prediction error

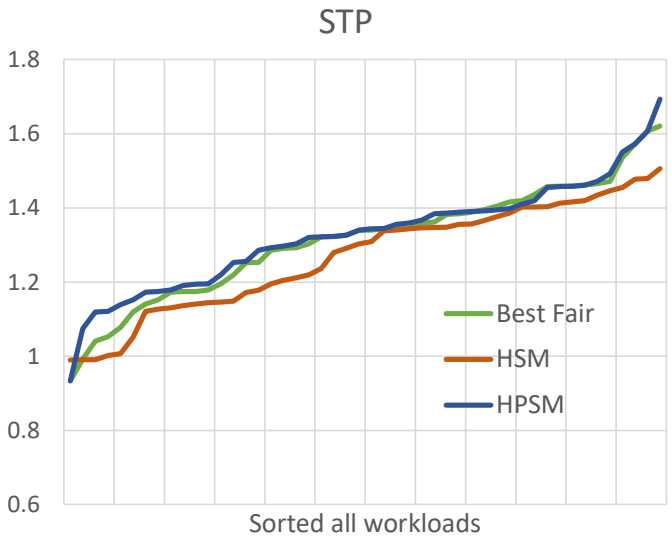


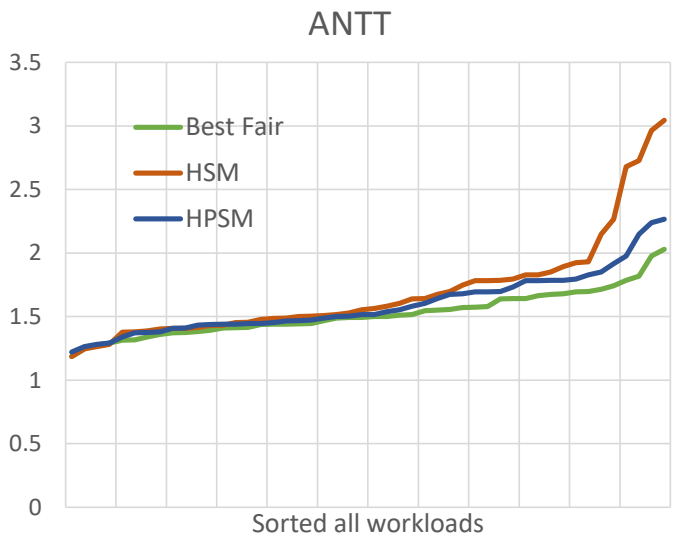




Performance evaluation







Conclusions and future work

EURO-PAR
CONFERENCE 2022

We have compared two CTA distribution strategies: SMT and SMK

We have built a Hybrid Piece-wise Slowdown Model that predicts accurately the normalized progress of co-executing kernels

We have implemented a fairness-based CTA allocation policy

In the future, we plan to develop a more sophisticated model for kernel performance to reduce errors and number of approximations

THANK YOU FOR YOUR ATTENTION!

ANY QUESTIONS?