µC-STATES: FINE-GRAINED GPU DATAPATH POWER MANAGEMENT

ONUR KAYIRAN, ADWAIT JOG, ASHUTOSH PATTNAIK, RACHATA AUSAVARUNGNIRUN,
XULONG TANG, MAHMUT T. KANDEMIR, GABRIEL H. LOH, ONUR MUTLU, CHITA R. DAS

EXECUTIVE SUMMARY

AMD

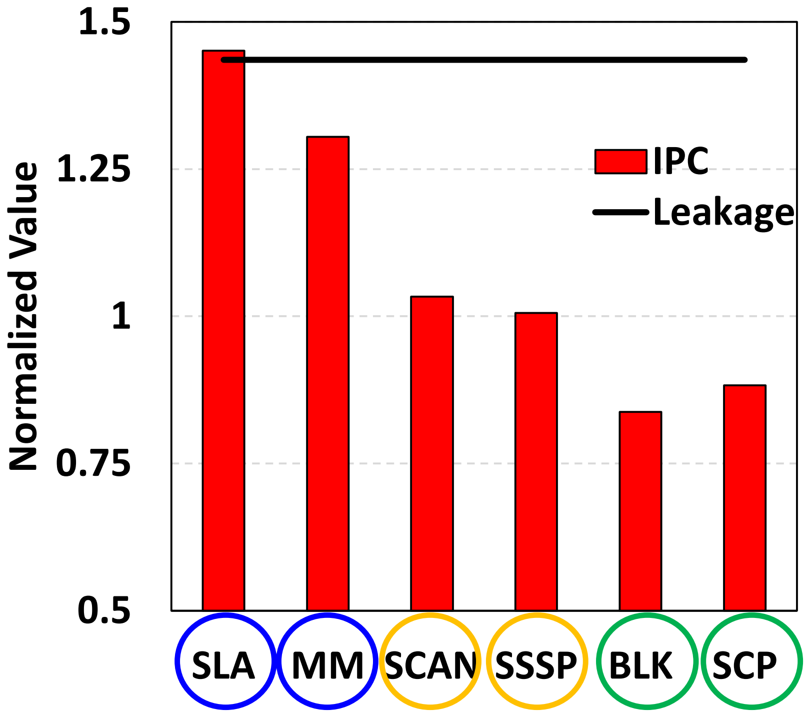
- The peak throughput and individual capabilities of the GPU cores are increasing
  - Lower and imbalanced utilization of datapath components

- We identify two key problems:
  - Wastage of datapath resources and increased static power consumption
  - Performance degradation due to contention in memory hierarchy

Our Proposal - µC-States:

- A fine-grained dynamic power- and clock-gating mechanism for the entire datapath based on queuing theory principles
- Reduces static and dynamic power, improves performance

BIG CORES VS. SMALL CORES



- SLA & MM
  - Performance ↑
  - Leakage power ↑
- SCAN & SSSP
  - Performance ↔
  - Leakage power ↑
- BLK & SCP
  - Performance ↓
  - Leakage power ↑

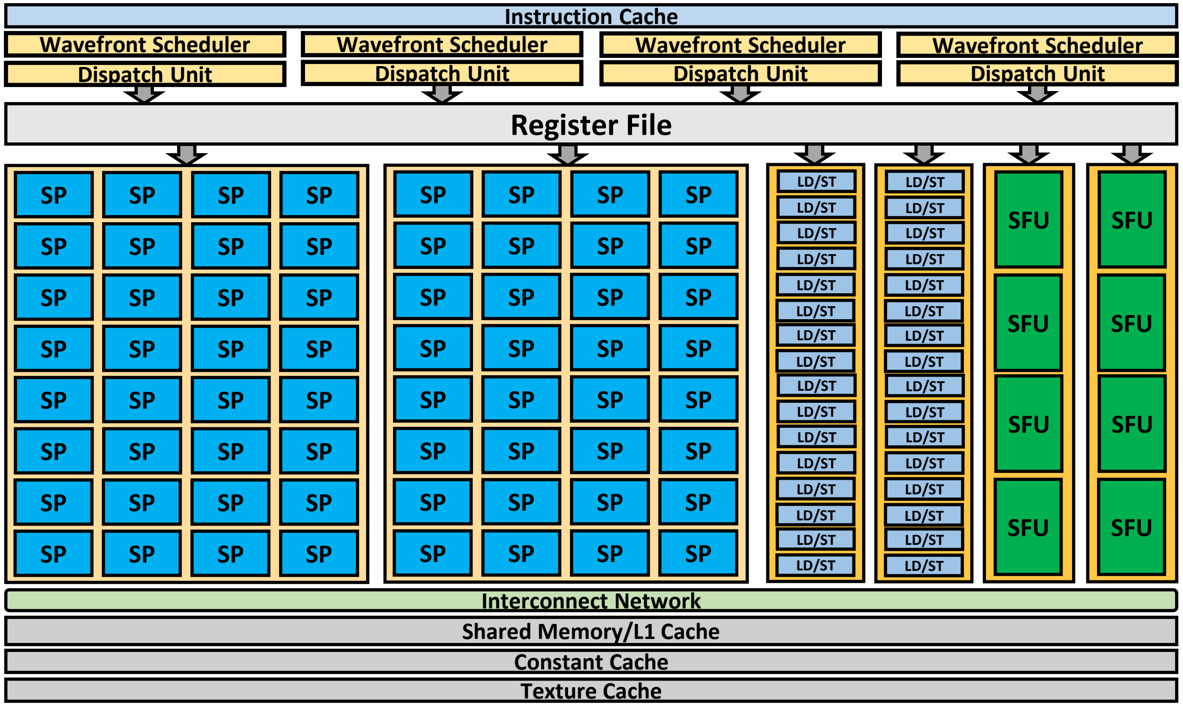
OUTLINE

- Summary
- Background
- Motivation and Analysis
- Our Proposal
- Evaluation
- Conclusions

BACKGROUND

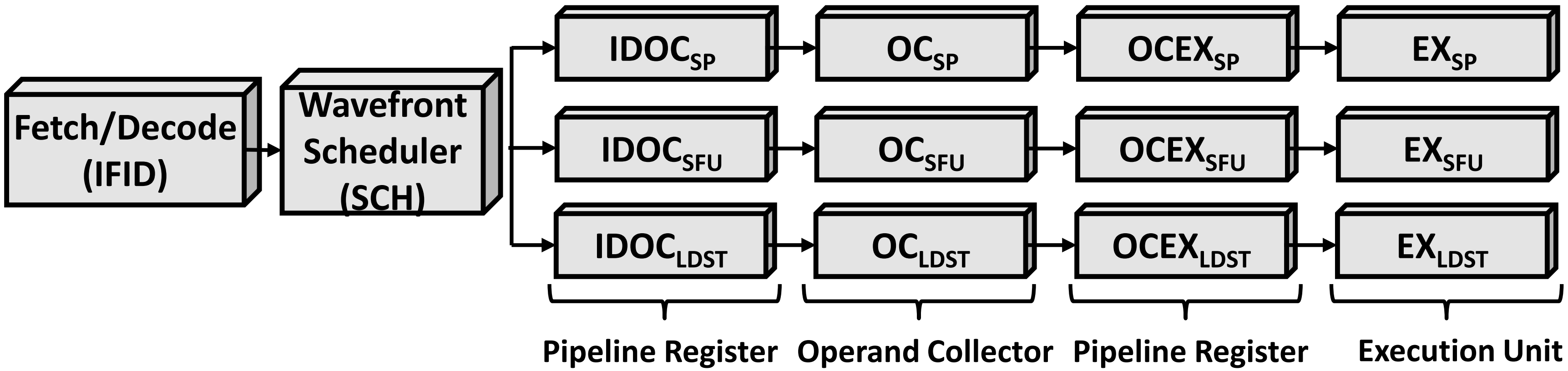
A HIGH-END GPU DATAPATH

- Per GPU core:
  - 4 wavefront schedulers
  - 64 shader processors
  - 32 LD/ST units
- Evaluation of larger GPU cores



BACKGROUND

ANALYZING CORE BOTTLENECKS



- The datapath can be modeled as a simple queuing system
  - Component with the highest utilization is the bottleneck
- Utilization Law [Jain, 1991]:
  - Utilization = Service time * Throughput
  - SP and SFU units have deterministic service times
  - LD/ST unit waits for response from the memory system
  - Used to calculate the component with highest utilization
- Little’s Law [Little, OR 1961]:
  - Number of jobs in the system = Arrival rate * Response time
  - Response time includes queuing delays
  - Used to estimate Response Time of memory instructions in LD/ST unit

BACKGROUND
POWER- AND CLOCK-GATING

- Power-gating reduces static power
- Clock-gating reduces dynamic power

- Power-gating leads to loss of data
  - Employ clock-gating for:
    - Instruction buffer, pipeline registers, register file banks, and LD/ST queue

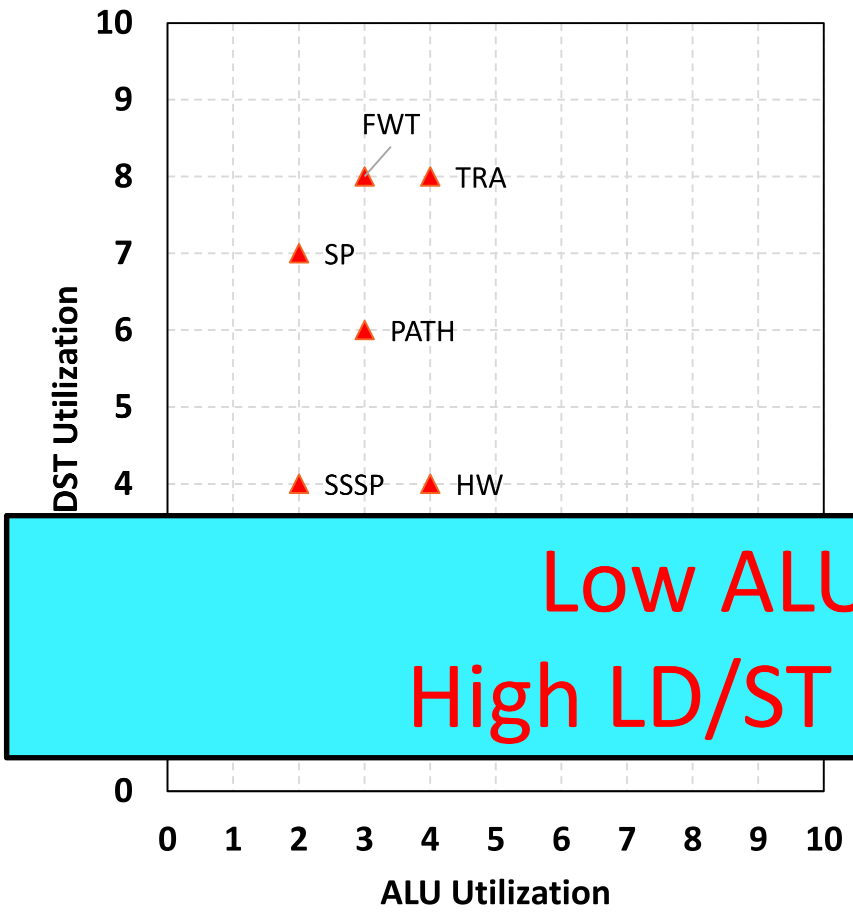
- Power-gating overheads
  - Wake-up delay: Time to power on a component
  - Break-even time: Shortest time to power-gate to compensate for the energy overhead

OUTLINE

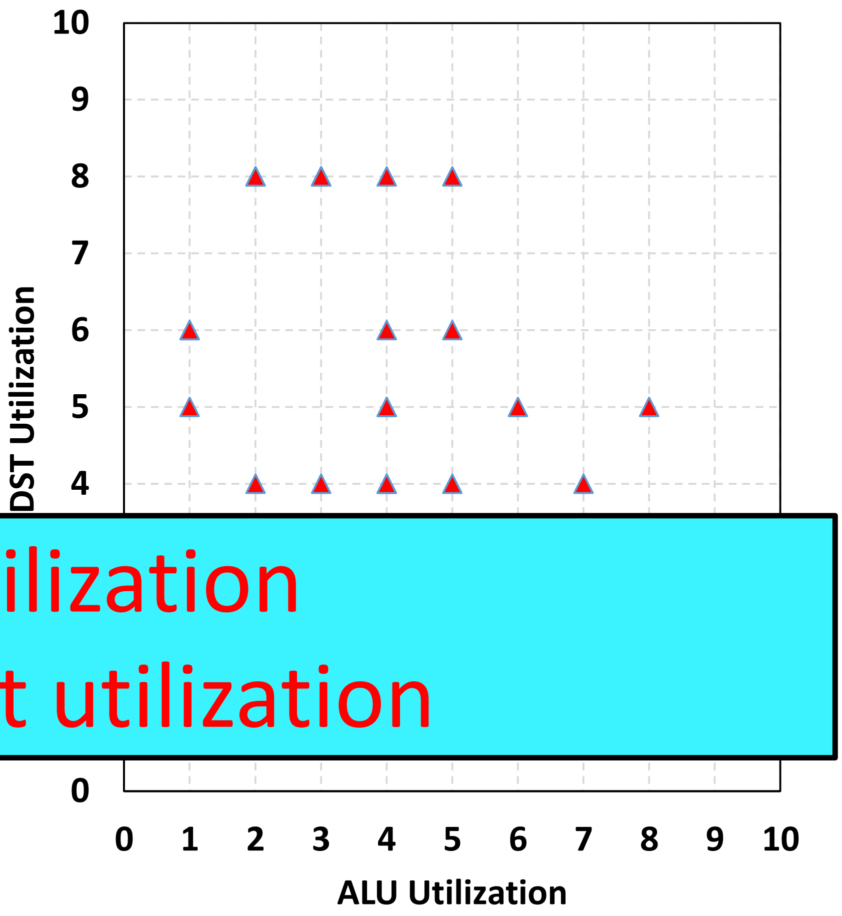
- Summary
- Background
- Motivation and Analysis
- Our Proposal
- Evaluation
- Conclusions

MOTIVATION AND ANALYSIS

ALU AND LDST UTILIZATION W/ REAL EXPERIMENTS



NVIDIA K20 GPU



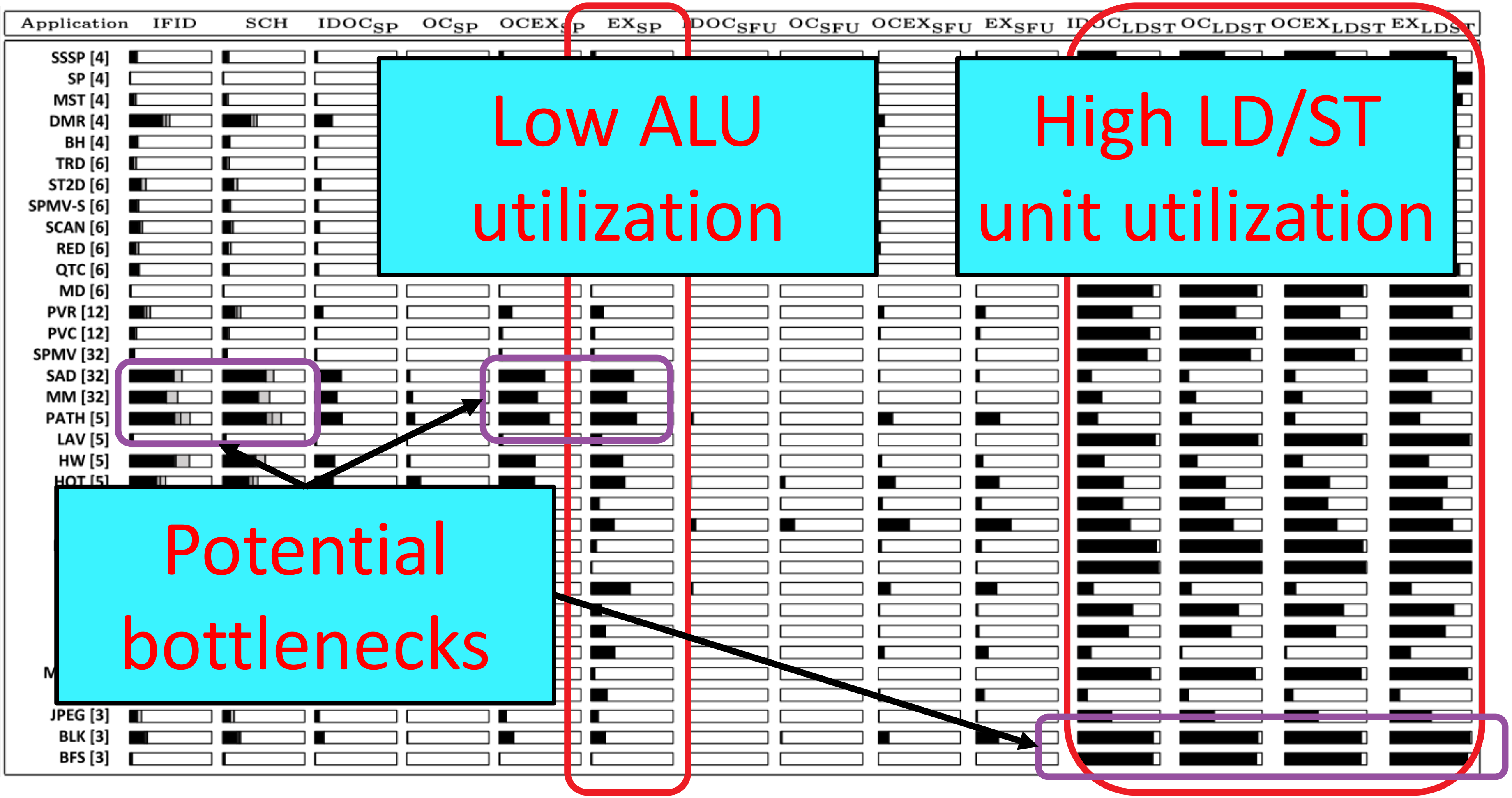
NVIDIA GTX 660 GPU

Low ALU utilization

High LD/ST unit utilization

MOTIVATION AND ANALYSIS

PER-COMPONENT UTILIZATION W/ SIMULATION



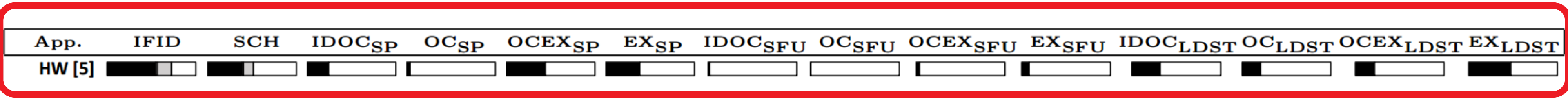
Low ALU utilization

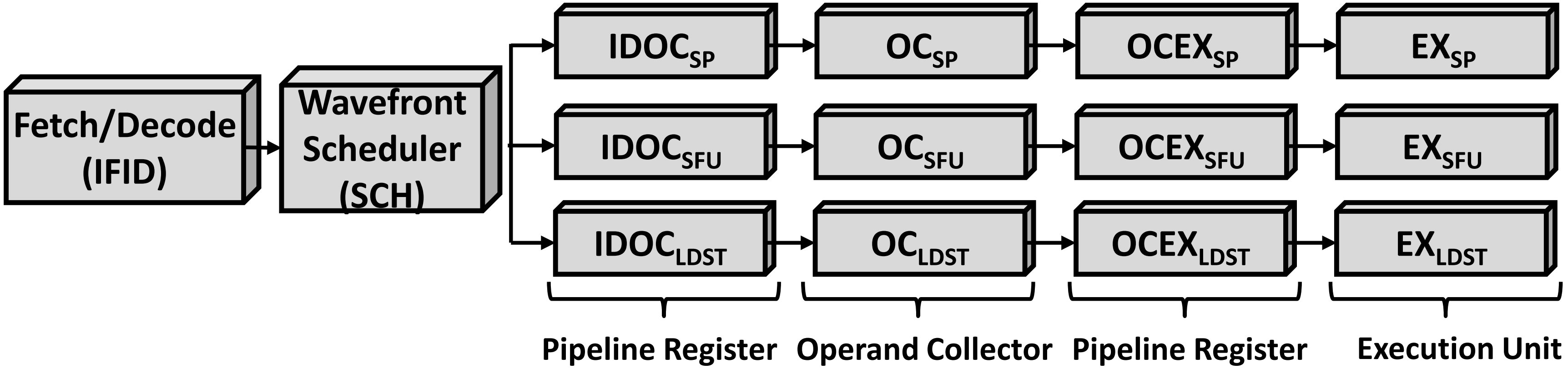
High LD/ST unit utilization

Potential bottlenecks

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS

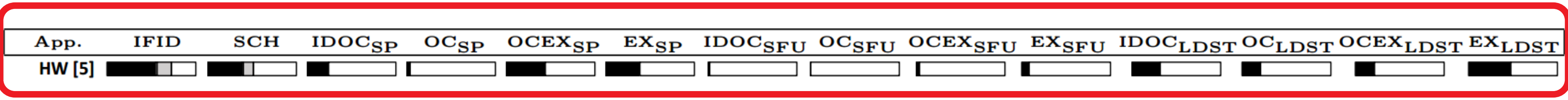


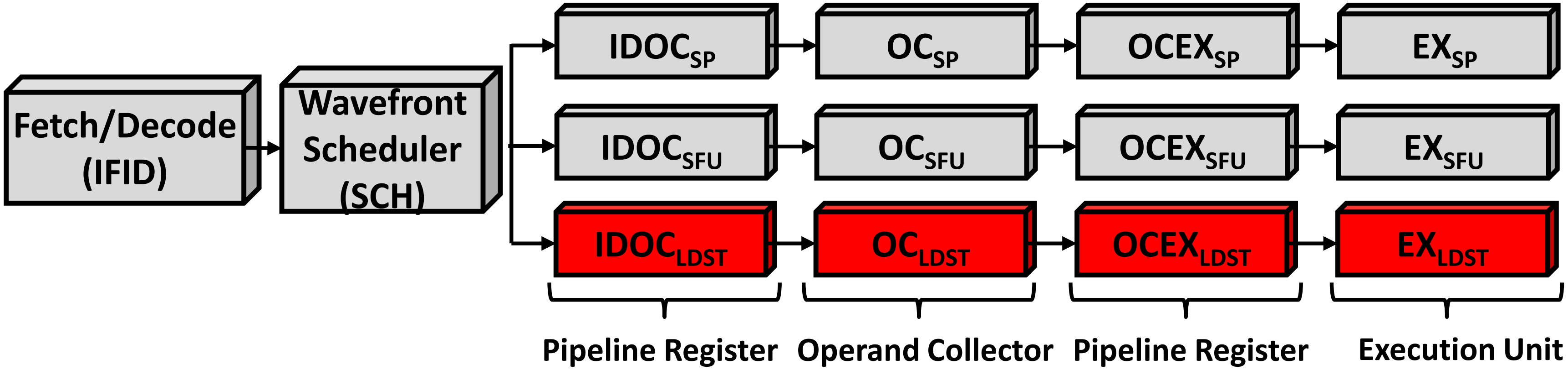


- Compute-intensive application

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS



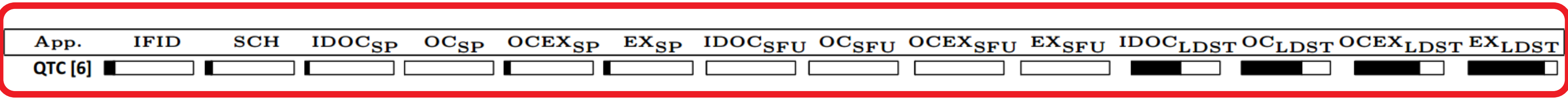


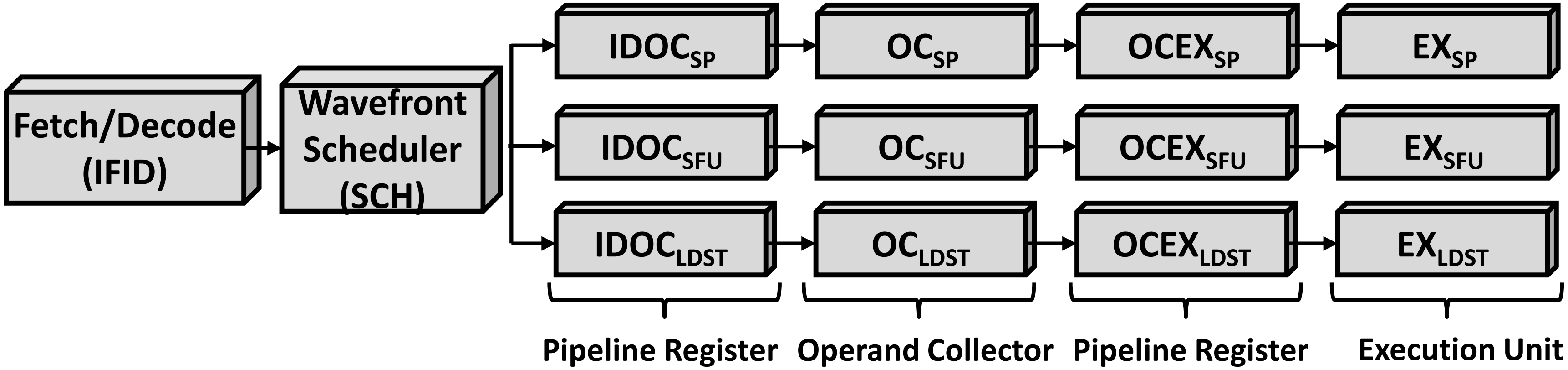
- Compute-intensive application
  - Halving the width of the red components -> No performance impact
  - Halving the width of all components -> 30% lower performance

Many components are critical for performance

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS




Fetch/Decode (IFID)

Wavefront Scheduler (SCH)

IDOC$_{SP}$, OC$_{SP}$, OCEX$_{SP}$, EX$_{SP}$

IDOC$_{SFU}$, OC$_{SFU}$, OCEX$_{SFU}$, EX$_{SFU}$

IDOC$_{LDST}$, OC$_{LDST}$, OCEX$_{LDST}$, EX$_{LDST}$

Pipeline Register

Operand Collector

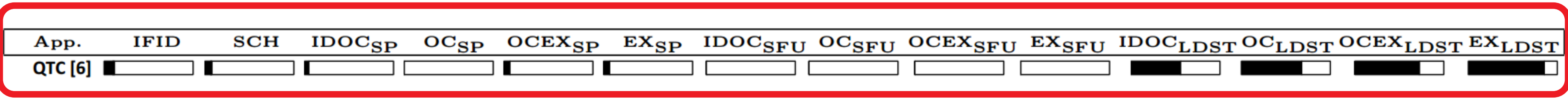
Pipeline Register

Execution Unit

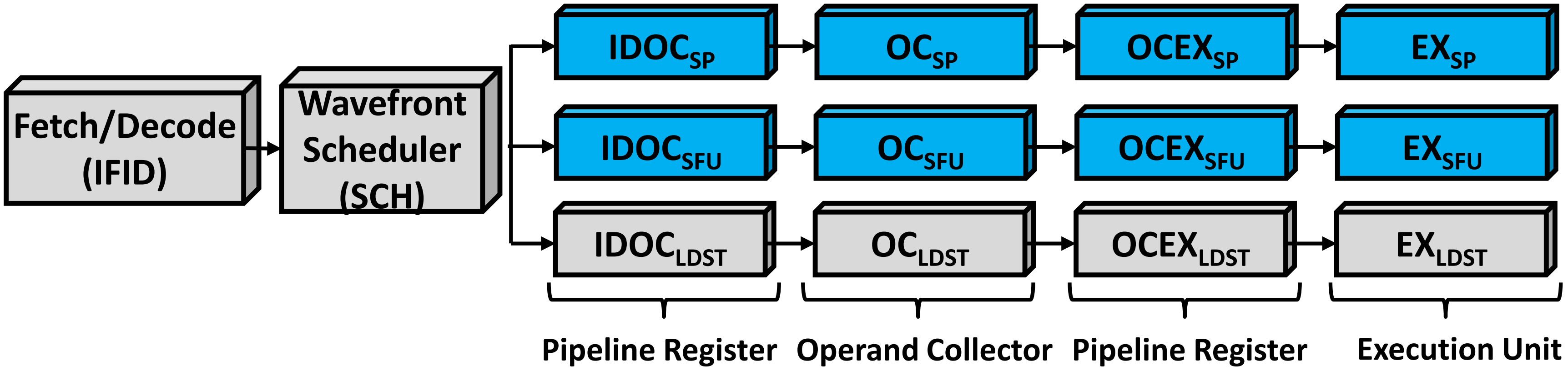
- Application with LD/ST unit bottleneck

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS



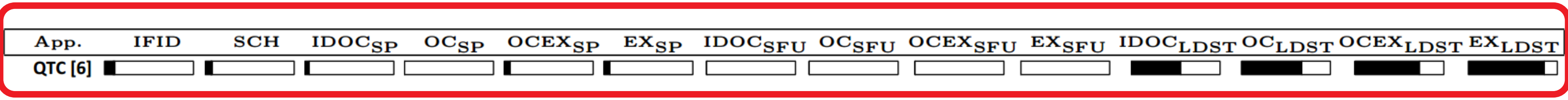
App.	IFID	SCH	$IDOC_{SP}$	OC_{SP}	$OCEX_{SP}$	EX_{SP}	$IDOC_{SFU}$	OC_{SFU}	$OCEX_{SFU}$	EX_{SFU}	$IDOC_{LDST}$	OC_{LDST}	$OCEX_{LDST}$	EX_{LDST}
QTC [6]														

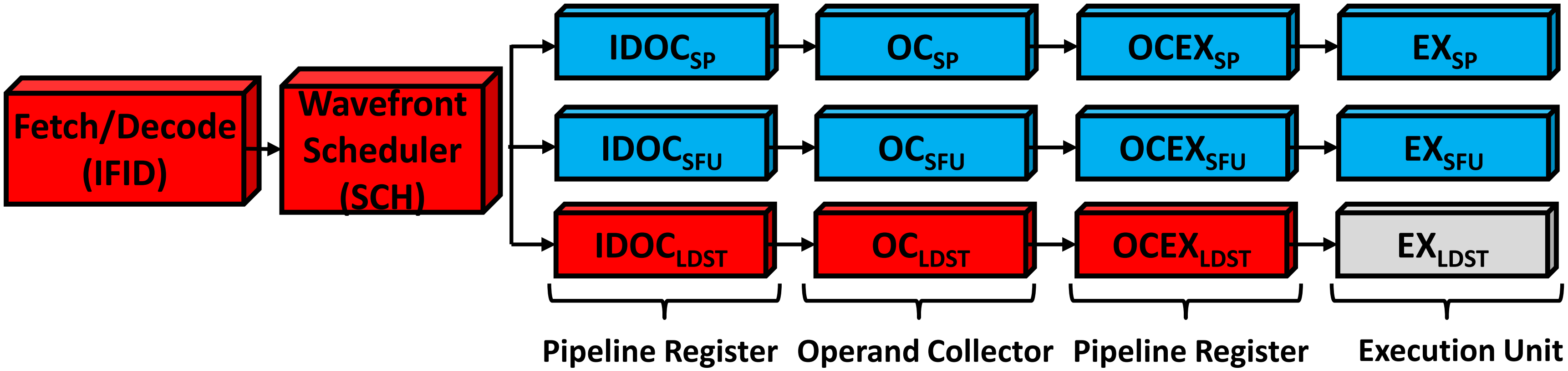


- Application with LD/ST unit bottleneck
  - Halving the width of the blue components -> No performance impact

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS

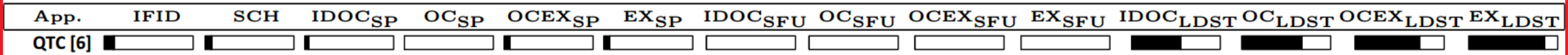


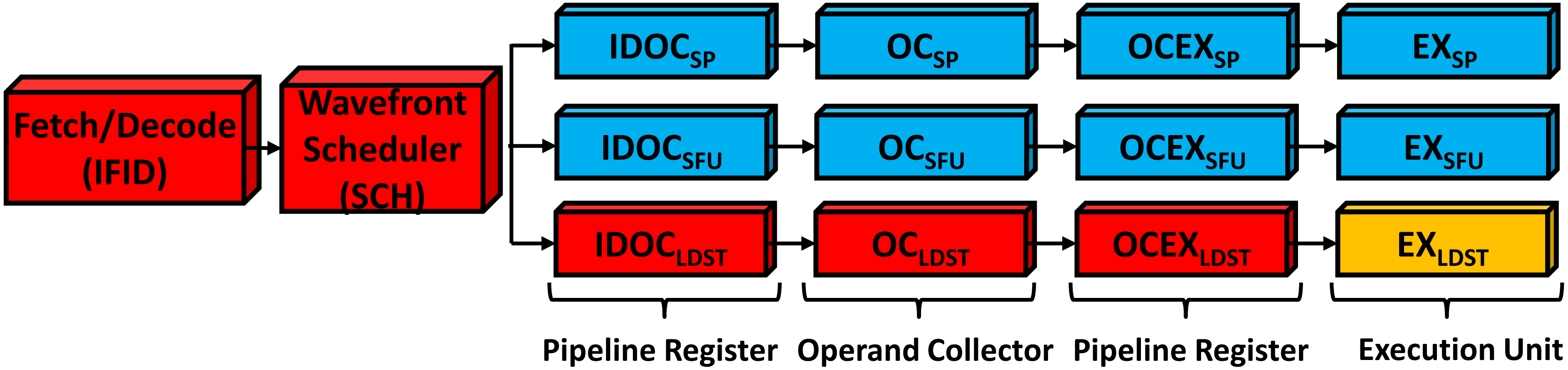


- Application with LD/ST unit bottleneck
  - Halving the width of the blue components -> No performance impact
  - Halving the width of the blue + red components -> 4% performance loss

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS



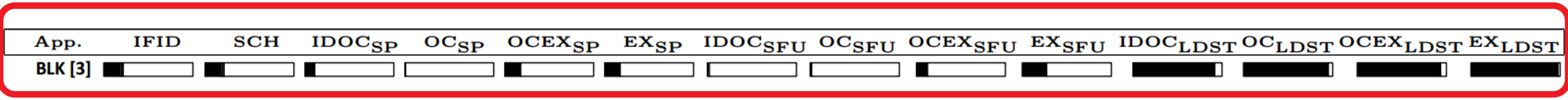


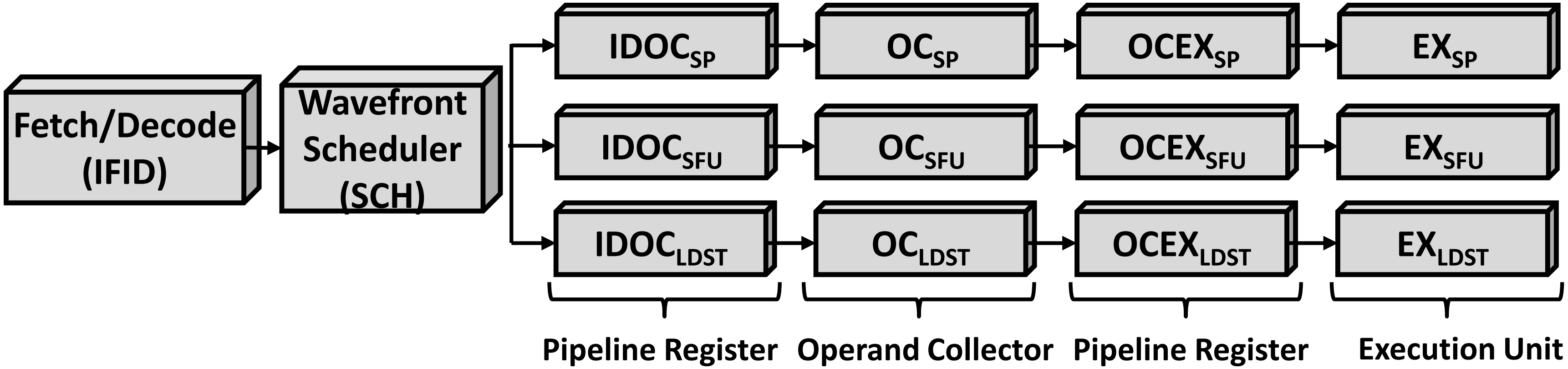
- Application with LD/ST unit bottleneck
  - Halving the width of the blue components -> No performance impact
  - Halving the width of the blue + red components -> 4% performance loss

LD/ST unit is the bottleneck

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS

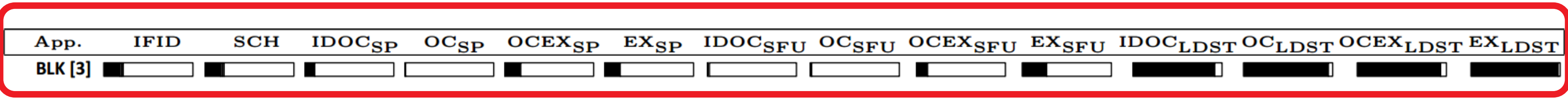


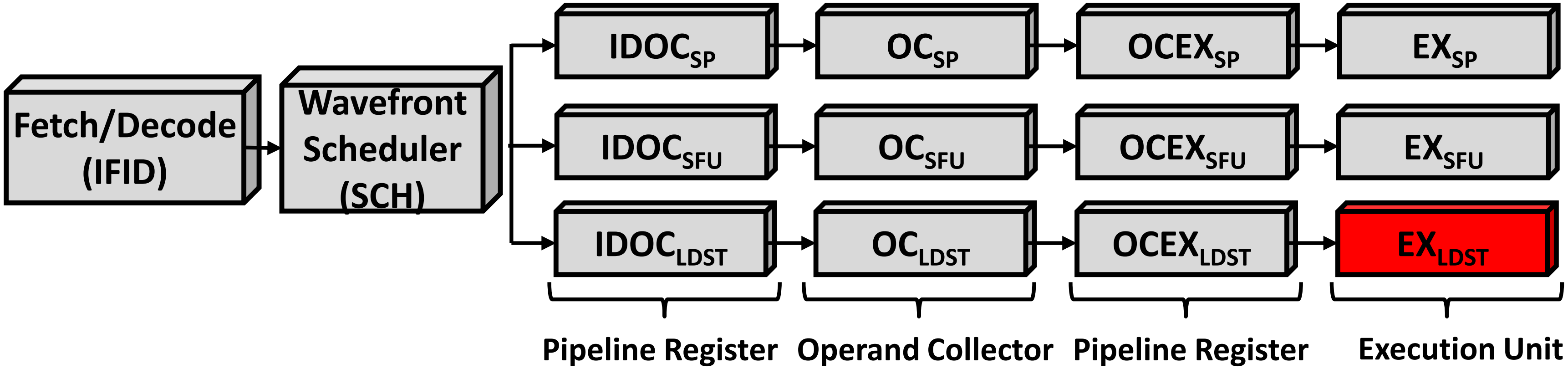


- Application with memory system bottleneck
  - Similar to QTC, but it has very high memory response time

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS

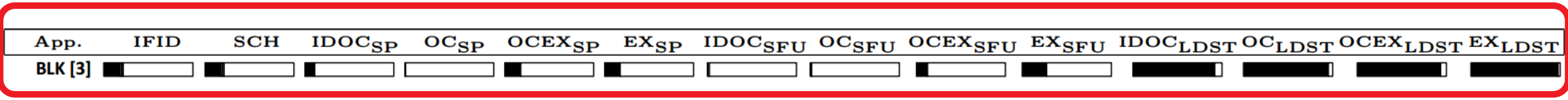


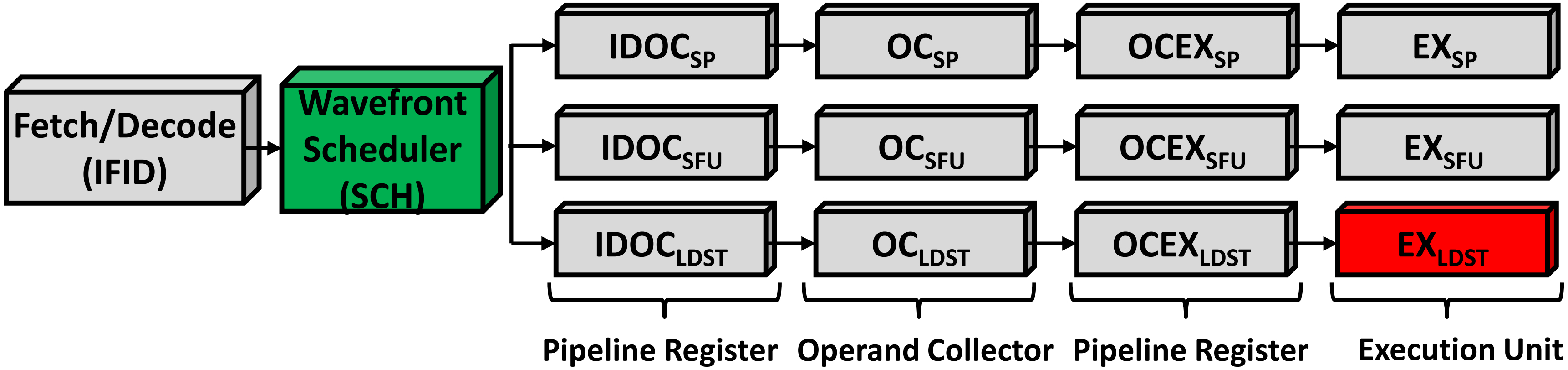


- Application with memory system bottleneck
  - Similar to QTC, but it has very high memory response time
  - Halving the width of LD/ST unit does not degrade performance

MOTIVATION AND ANALYSIS

APPLICATION SENSITIVITY TO DATAPATH COMPONENTS

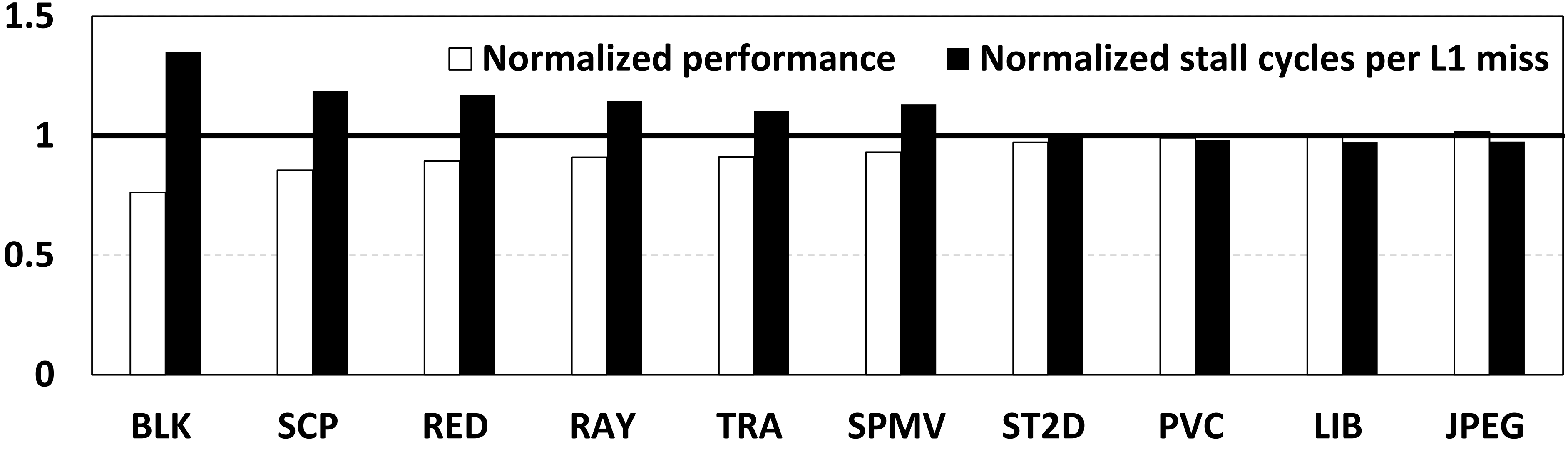




Memory system is the bottleneck,
not the LD/ST unit.
Higher issue width degrades performance!

MOTIVATION AND ANALYSIS

APPLICATIONS WITH MEMORY SYSTEM BOTTLENECK

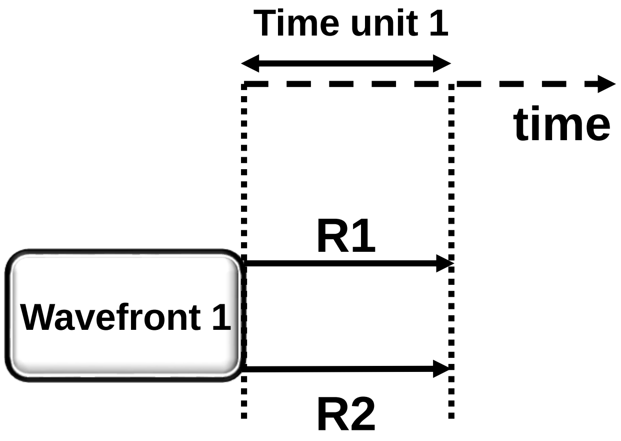


- In memory-bound applications, performance degrades with the increase in L1 stalls

MOTIVATION AND ANALYSIS

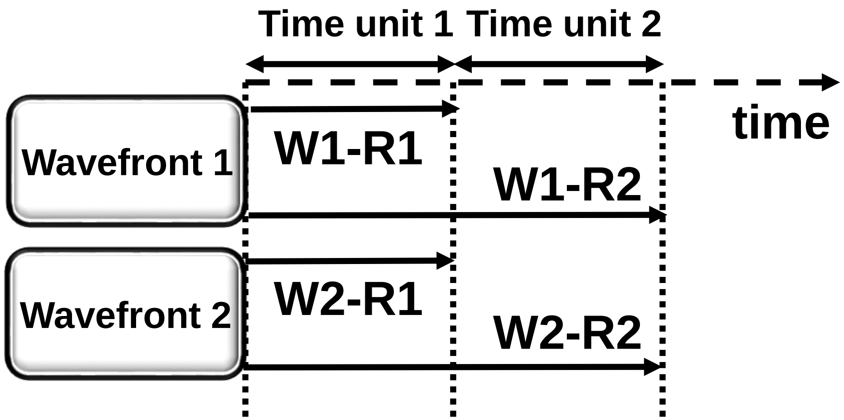
APPLICATIONS WITH MEMORY SYSTEM BOTTLENECK

Single issue width



- 2 outstanding requests / unit time

Double issue width



- 3 outstanding requests / unit time

When memory system is the bottleneck, higher issue width might degrade performance!

MOTIVATION AND ANALYSIS

KEY INSIGHTS

- Observation: Low ALU utilization, high LD/ST unit utilization

- Compute-intensive applications: Bottleneck can be fetch/decode units, wavefronts schedulers, or execution units

- Memory-intensive applications: Bottleneck can be the LD/ST unit, or the memory system

- Applications with memory system bottleneck: Divergent applications can lose performance with high issue width

OUTLINE

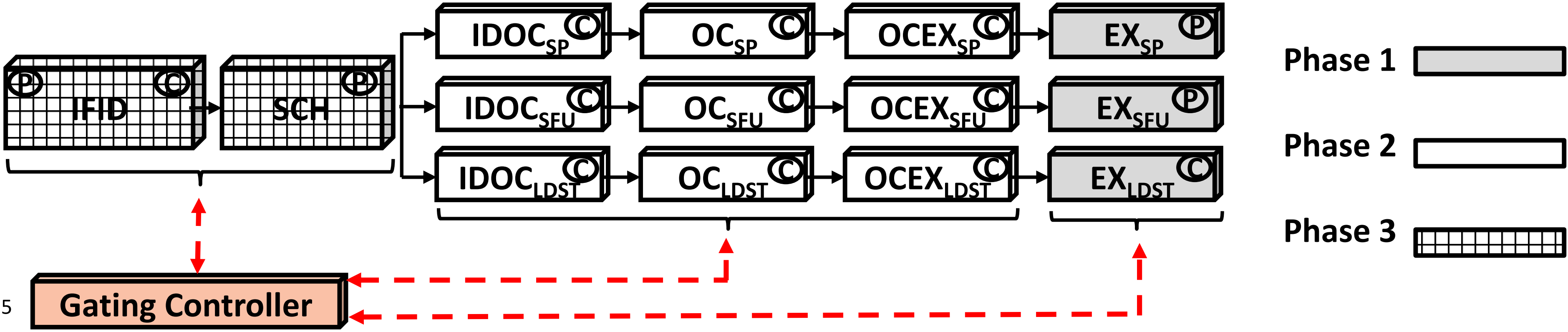
- Summary
- Background
- Motivation and Analysis
- Our Proposal
- Evaluation
- Conclusions

µC-STATES
KEY IDEAS

- Goal:
  - To reduce the static and dynamic power of the GPU core pipeline
  - To maintain, and when possible improve performance
- Power benefits:
  - Based on bottleneck analysis
  - Power- or clock-gates components that are not critical for performance
  - Employs clock-gating for components that hold execution state, or hold data for long periods
- Performance benefits:
  - Reducing issue width when memory system is the bottleneck improves performance
  - Only half the width of each component is gated

μC-STATES
ALGORITHM DETAILS

- Periodically goes through three phases
- First phase: Execution units and LD/ST unit
  - Power-gates execution units with low utilization
  - Clock-gates LD/ST units when memory response time (estimated by Little’s Law) is high
- Second phase: Register file banks and pipeline registers
  - Compares the utilization of each component with its corresponding execute stage unit
  - If lower, they are not bottleneck, and can be gated-off
- Third phase: Wavefront scheduler and fetch/decode units
  - Compares scheduler utilization to cumulative executive stage utilization
  - If lower, issue width is halved
  - If fetch/decode utilization is lower than scheduler’s, fetch/decode width is halved



µC-STATES

MORE IN THE PAPER

- Employed at coarse time granularity
- Not sensitive to overheads related to entering or exiting power-gating states
- Independent of the underlying wavefront scheduler
- Issue width sizing is fundamentally different than thread-level parallelism management
  - Comparison to CCWS [Rogers+, MICRO 2012]

OUTLINE

- Summary
- Background
- Motivation and Analysis
- Our Proposal
- Evaluation
- Conclusions

EVALUATION METHODOLOGY

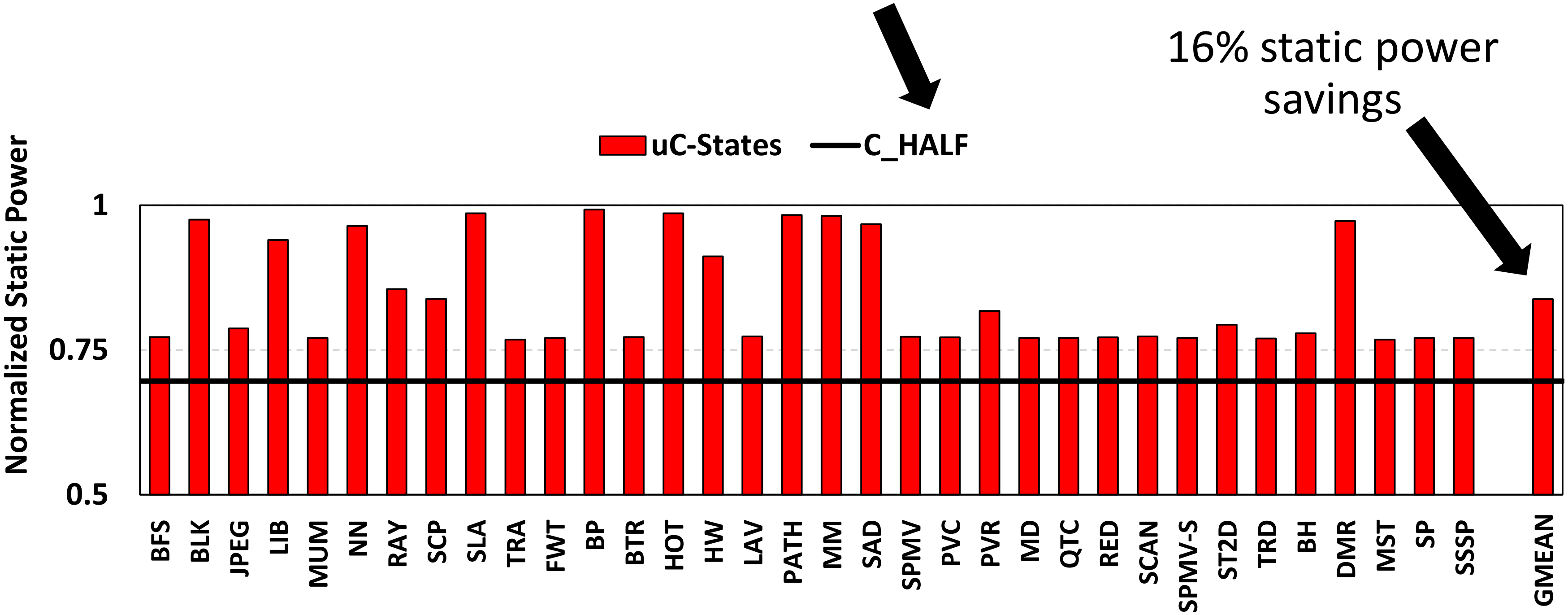
- We simulate the baseline architecture using a modified version of GPGPU-Sim v3.2.2 that allows larger GPU cores

- GPU-Wattch
  - Reports dynamic power
  - Area calculations for static power
  - Conservative assumption of non-core components, such as the memory subsystem and DRAM, to contribute to 40% of static power

- Baseline architecture
  - 16 Shader Cores, SIMT Width = 32 × 4
  - 36K Registers, 16kB L1 cache, 48kB shared memory
  - GTO wavefront scheduler
  - 6 shared GDDR5 MCs

RESULTS SUMMARY

POWER SAVINGS



All components are half-width

16% static power savings

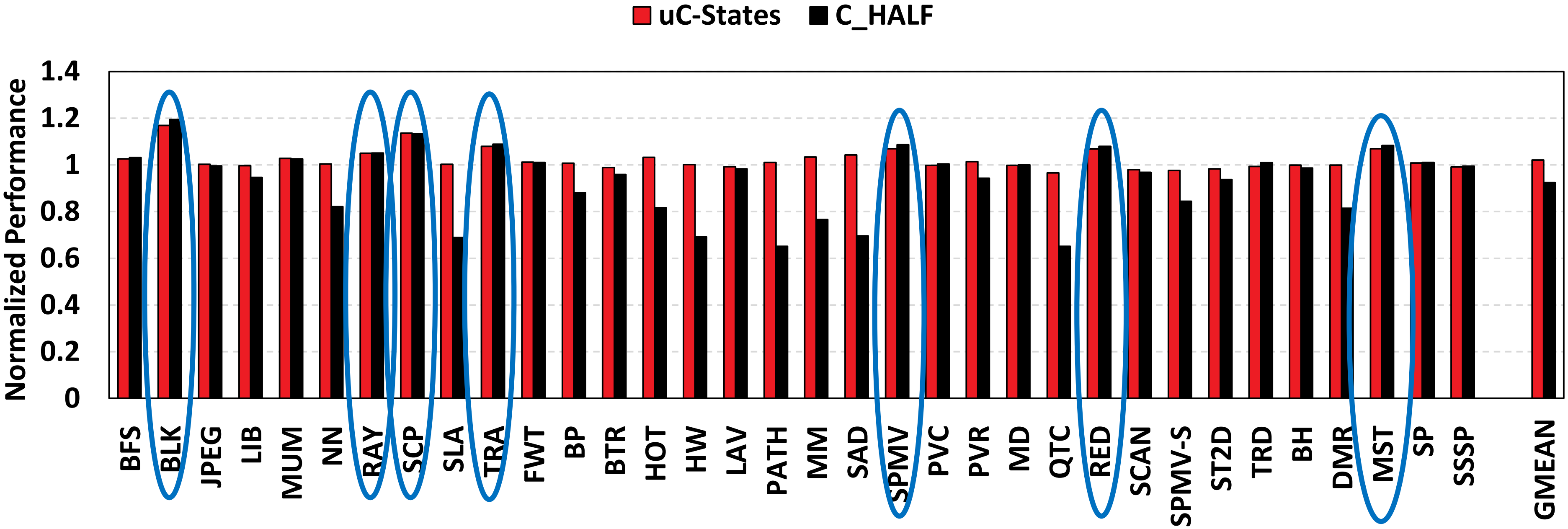
7% dynamic power savings

11% total power savings for the chip

RESULTS SUMMARY

PERFORMANCE

All components are half-width



10% performance improvement over C_HALF

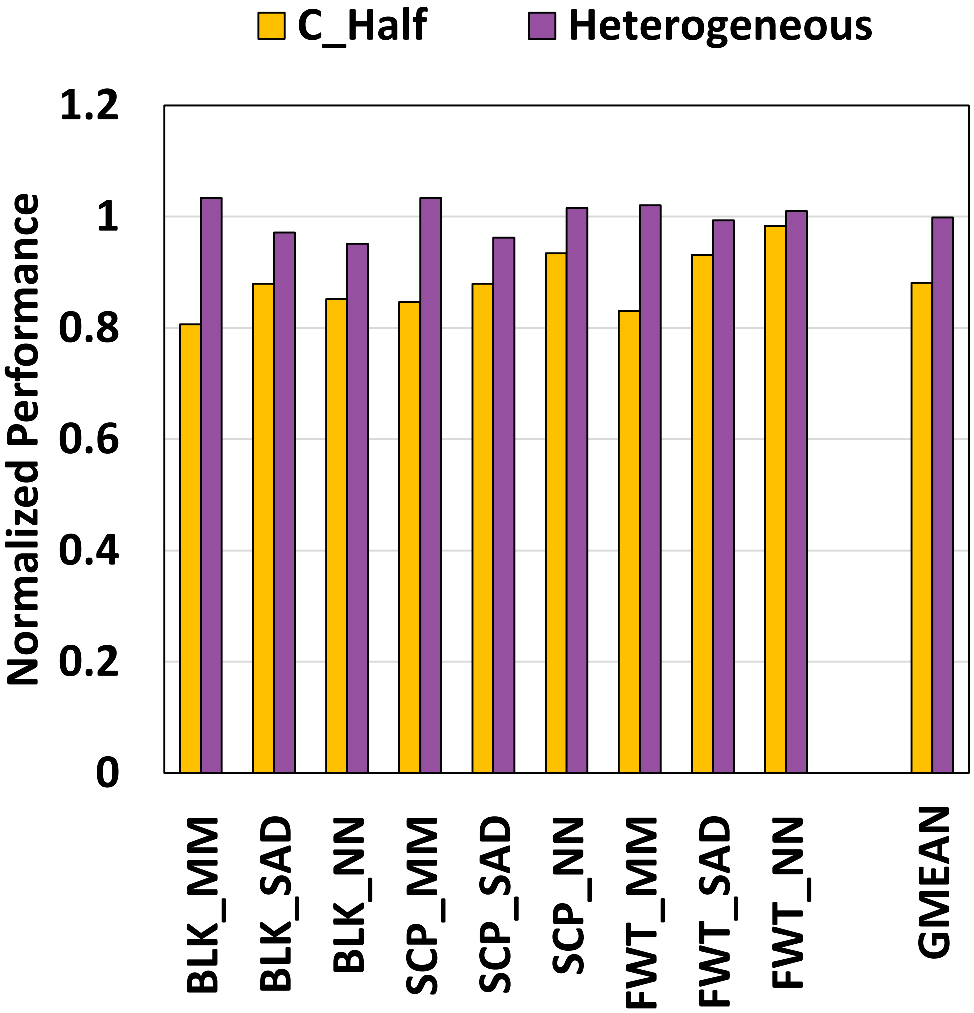
2% performance improvement over the baseline

9% performance improvement for applications with memory system bottleneck

RESULTS SUMMARY

HETEROGENEOUS-CORE GPUS

- A system with 8 small and 8 big cores
- Performs better than 16 small cores
- Performs as good as 16 big cores
- Has smaller power consumption and area than the 16-core system



OUTLINE

- Summary
- Background
- Motivation and Analysis
- Our Proposal
- Evaluation
- Conclusions

CONCLUSIONS

- Many GPU datapath components are heavily underutilized

- More resources in a GPU core can sometimes degrade performance because of contention in the memory system

- µC-States minimizes power consumption by turning off datapath components that are not performance bottlenecks, and improves performance for applications with memory system bottleneck

- Our analysis could be useful in guiding scheduling and design decisions in a heterogeneous-core GPU with both small and big cores

- Our analysis and proposal can be useful for developing other new analyses and optimization techniques for more efficient GPU and heterogeneous architectures

Thanks!
Questions?

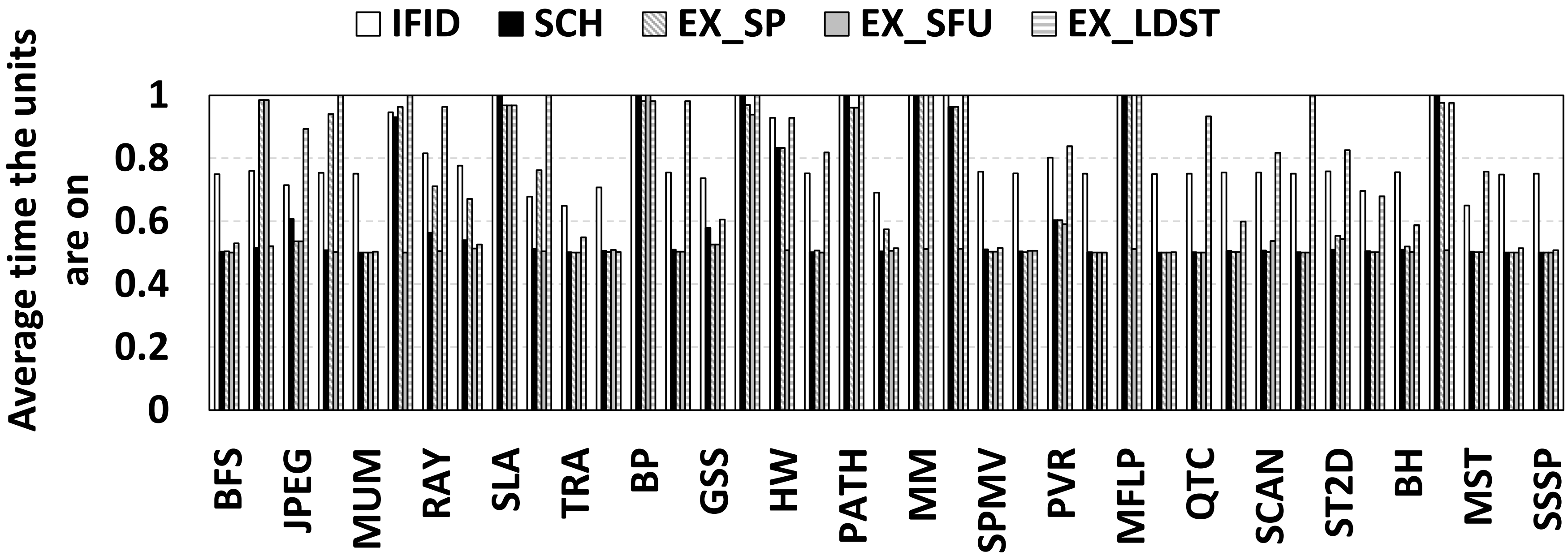
μC-STATES: FINE-GRAINED GPU DATAPATH POWER MANAGEMENT

ONUR KAYIRAN, ADWAIT JOG, ASHUTOSH PATTNAIK, RACHATA AUSAVARUNGNIRUN, XULONG TANG, MAHMUT T. KANDEMIR, GABRIEL H. LOH, ONUR MUTLU, CHITA R. DAS

Backup

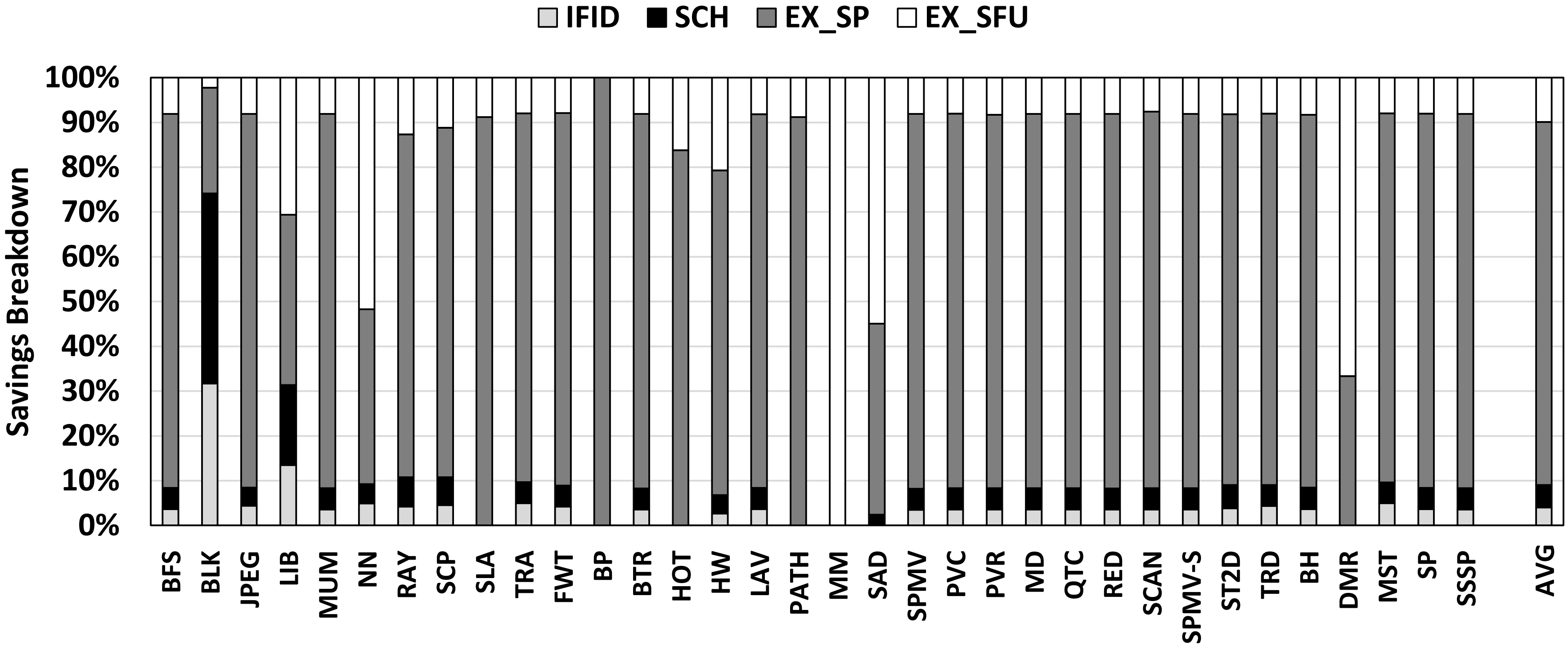
ADDITIONAL RESULTS

AVERAGE TIME THE UNITS ARE ON



ADDITIONAL RESULTS

SAVINGS BREAKDOWN



DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION